智能数字信号处理器及其解决方案提供商

# BY320RV0025硬件参考手册

版本V1.0

青岛本原微电子有限公司

# 目 录

**1 RV0025 处理器概述 ..... 10**

1.1 内核介绍 ..... 10

1.2 主要特性 ..... 11

1.3 存储器 ..... 11

**2 系统控制和中断 ..... 16**

2.1 介绍 ..... 16

2.2 电源管理 ..... 17

2.3 器件标识与配置寄存器 ..... 17

2.4 复位 ..... 17

2.5 外设中断 ..... 21

2.6 异常和不可屏蔽中断 ..... 23

2.7 时钟 ..... 26

2.8 定时器 0/1/2 ..... 40

2.9 看门狗定时器 ..... 40

2.10 低功耗模式 ..... 45

2.11 存储器控制器模块 ..... 49

2.12 JTAG ..... 55

2.13 双代码安全模块 ..... 55

2.14 系统控制寄存器配置限制 ..... 65

2.15 寄存器描述 ..... 66

**3 ROM 和启动引导 ..... 249**

3.1 简介 ..... 249

3.2 器件启动引导序列 ..... 249

3.3 器件启动引导模式 250
3.4 器件启动引导配置 250
3.5 器件启动流程图 256
3.6 器件复位与异常处理 259
3.7 引导程序 ROM 描述 259
**4 闪存控制器（FLASH 存储控制器） 277**
4.1 FLASH 和 OTP 存储器简介 277
4.2 FLASH BANK、OTP 和电荷泵 279
4.3 FLASH 模块控制器 (FMC) 279
4.4 FLASH 和 OTP 存储器掉电模式与唤醒 279
4.5 有效缓冲周期 280
4.6 FLASH 和 OTP 存储器性能 281
4.7 FLASH 读接口 282
4.8 FLASH 擦除和编程 285
4.9 错误校正码 (ECC) 保护 286
4.10 FLASH 和 OTP 存储器内的保留位置 292
4.11 将应用从 RAM 迁移到 FLASH 292
4.12 更改 FLASH 控制寄存器的步骤 293
4.13 软件 294
4.14 寄存器描述 294
**5 双时钟比较器（DCC） 326**
5.1 简介 326
5.2 模块操作 327
5.3 中断 334

5.4 软件 ..... 334
5.5 寄存器描述 ..... 338
**6 后台循环冗余校验模块（BGCRC） ..... 349**
6.1 简介 ..... 349
6.2 主要特性 ..... 349
6.3 功能描述 ..... 349
6.4 寄存器描述 ..... 354
**7 通用输入/输出接口（GPIO） ..... 381**
7.1 简介 ..... 381
7.2 配置概述 ..... 383
7.3 ADC 引脚上的数字输入（AIOs） ..... 385
7.4 ADC 引脚的数字输入和输出（AGPIOs） ..... 385
7.5 数字通用 I/O 控制 ..... 386
7.6 输入限定 ..... 388
7.7 GPIO 和外设复用 ..... 393
7.8 内部上拉配置要求 ..... 400
7.9 寄存器描述 ..... 400
**8 交叉开关（X-BAR) ..... 486**
8.1 简介 ..... 486
8.2 输入交叉开关和 CLB 输入交叉开关 ..... 486
8.3 EPWM、CLB 和 GPIO 输出交叉开关 ..... 489
8.4 寄存器描述 ..... 497
**9 直接存储器访问（DMA） ..... 795**
9.1 简介 ..... 795

9.2 主要特性 ..... 795
9.3 功能描述 ..... 796
9.4 寄存器描述 ..... 814
**10 可配置逻辑块（CLB） ..... 840**
10.1 简介 ..... 840
10.2 主要特性 ..... 840
10.3 功能描述 ..... 840
10.4 寄存器描述 ..... 870
**11 主机接口控制器（HIC） ..... 953**
11.1 简介 ..... 953
11.2 主要特性 ..... 954
11.3 功能描述 ..... 959
11.4 寄存器描述 ..... 976
**12 模拟子系统 ..... 1036**
12.1 简介 ..... 1036
12.2 优化上电时间 ..... 1039
12.3 ADC 引脚（AIO）的数字输入 ..... 1039
12.4 模数转换器引脚上的数字输入输出（AGPIOs） ..... 1039
12.5 模拟引脚与内部连接 ..... 1040
12.6 寄存器描述 ..... 1045
**13 模数转换器 （ADC） ..... 1060**
13.1 简介 ..... 1060
13.2 主要特性 ..... 1060
13.3 功能描述 ..... 1061

13.4 寄存器描述 ..... 1103

**14 比较器子系统（CMPSS） ..... 1228**

14.1 简介 ..... 1228

14.2 主要特性 ..... 1229

14.3 功能描述 ..... 1230

14.4 寄存器描述 ..... 1240

**15 增强型脉宽调制器（EPWM） ..... 1257**

15.1 简介 ..... 1257

15.2 主要特性 ..... 1265

15.3 功能描述 ..... 1266

15.4 时基（TB）子模块 ..... 1267

15.5 计数器比较（CC）子模块 ..... 1281

15.6 动作限定器（AQ）子模块 ..... 1287

15.7 死区生成器（DB）子模块 ..... 1300

15.8 PWM 斩波器（PC）子模块 ..... 1307

15.9 跳闸区域（TZ）子模块 ..... 1311

15.10 事件触发（ET）子模块 ..... 1319

15.11 数字比较（DC）子模块 ..... 1325

15.12 EPWM 交叉开关（X-BAR） ..... 1336

15.13 功率拓扑结构的应用 ..... 1337

15.14 寄存器锁定保护 ..... 1353

15.15 高分辨率脉冲宽度调制器（HRPWM） ..... 1354

15.16 寄存器描述 ..... 1386

**16 增强型捕获模块（ECAP） ..... 1508**

16.1 简介 ..... 1508
16.2 描述 ..... 1509
16.3 ECAP 设备引脚配置 ..... 1510
16.4 捕获和 APWM 操作模式 ..... 1513
16.5 捕获方式描述 ..... 1514
16.6 ECAP 模块的应用 ..... 1524
16.7 APWM 模式的应用 ..... 1528
16.8 软件 ..... 1529
16.9 寄存器描述 ..... 1531
**17 高分辨率捕捉（HRCAP） ..... 1551**
17.1 简介 ..... 1551
17.2 操作细节 ..... 1552
17.3 已知例外 ..... 1556
17.4 软件 ..... 1557
17.5 寄存器描述 ..... 1557
**18 增强型正交编码器脉冲（EQEP） ..... 1569**
18.1 简介 ..... 1569
18.2 主要特性 ..... 1575
18.3 功能描述 ..... 1576
18.4 寄存器描述 ..... 1598
**19 控制器局域网（CAN） ..... 1626**
19.1 概述 ..... 1626
19.2 功能描述 ..... 1629
19.3 工作模式 ..... 1630

19.4 多种时钟源 ..... 1636
19.5 中断功能 ..... 1636
19.6 DMA 功能 ..... 1639
19.7 奇偶校验机制 ..... 1639
19.8 调试模式 ..... 1640
19.9 模块初始化 ..... 1640
19.10 消息对象配置 ..... 1641
19.11 消息处理 ..... 1644
19.12 位定时与同步 ..... 1651
19.13 接口寄存器组（IFx） ..... 1659
19.14 消息 RAM 组织 ..... 1662
19.15 寄存器描述 ..... 1667
**20 快速串行接口（FSI） ..... 1726**
20.1 FSI 特性 ..... 1726
20.2 FSI 结构 ..... 1727
20.3 功能描述 ..... 1735
20.4 编程指南 ..... 1770
20.5 寄存器描述 ..... 1773
**21 内部集成电路总线接口（I2C） ..... 1826**
21.1 简介 ..... 1826
21.2 主要特性 ..... 1826
21.3 功能描述 ..... 1827
21.4 模块详细操作 ..... 1831
21.5 I2C 的中断请求生成 ..... 1839

21.6 寄存器描述 ..... 1842

**22 本地互联网络（LIN） ..... 1865**

22.1 简介 ..... 1865

22.2 主要特性 ..... 1865

22.3 功能描述 ..... 1868

22.4 寄存器描述 ..... 1922

**23 电源管理总线（PMBUS） ..... 1971**

23.1 简介 ..... 1971

23.2 从机模式运行 ..... 1973

23.3 主机模式运行 ..... 1984

23.4 寄存器描述 ..... 1994

**24 串行通信接口（SCI） ..... 2013**

24.1 简介 ..... 2013

24.2 架构 ..... 2014

24.3 SCI 模块信号汇总 ..... 2016

24.4 多处理器和异步通信模式 ..... 2017

24.5 SCI 可编程数据格式 ..... 2017

24.6 SCI 多处理器通信 ..... 2018

24.7 空闲线多处理器模式 ..... 2020

24.8 地址位多处理器模式 ..... 2023

24.9 SCI 通信格式 ..... 2024

24.10 SCI 端口中断 ..... 2026

24.11 SCI 波特率计算 ..... 2030

24.12 SCI 增强特性 ..... 2030

24.13 寄存器描述 ..... 2034

**25 串行外设接口（SPI） ..... 2056**

25.1 概述 ..... 2056

25.2 系统级集成 ..... 2058

25.3 SPI 工作原理 ..... 2062

25.4 编程步骤 ..... 2076

25.5 寄存器描述 ..... 2081

**26 嵌入式实时分析和诊断寄存器（ERAD） ..... 2098**

26.1 ERAD 特性 ..... 2098

26.2 EBC 单元（ENHANCED BUS COMPARATOR） ..... 2099

26.3 SEC 单元（SYSTEM EVENT COUNTER） ..... 2101

26.4 ERAD 编程顺序 ..... 2108

26.5 寄存器描述 ..... 2111

**27 修订历史记录 ..... 2149**

# 1 RV0025 处理器概述

## 1.1 内核介绍

BY320RV0025 处理器核是嵌入式领域的 RISC-V 兼容的高能效、低成本 32 位浮点处理器。基于 RISC-V 架构的 32 位浮点 DSP 内核 SummerCore，采用 RV32IMAFCB-Ext 指令集架构，面向实时控制应用而优化设计，主要应用于工业控制、新能源、电动汽车和电力电子等领域。

SummerCore 采用增强型静态双发射八级流水线微架构、高精度分支预测、零开销硬件循环、指令融合和专用计算加速器等关键技术。专用加速器包括三角函数、FFT 和 CRC 加速器。其架构图如图所示。

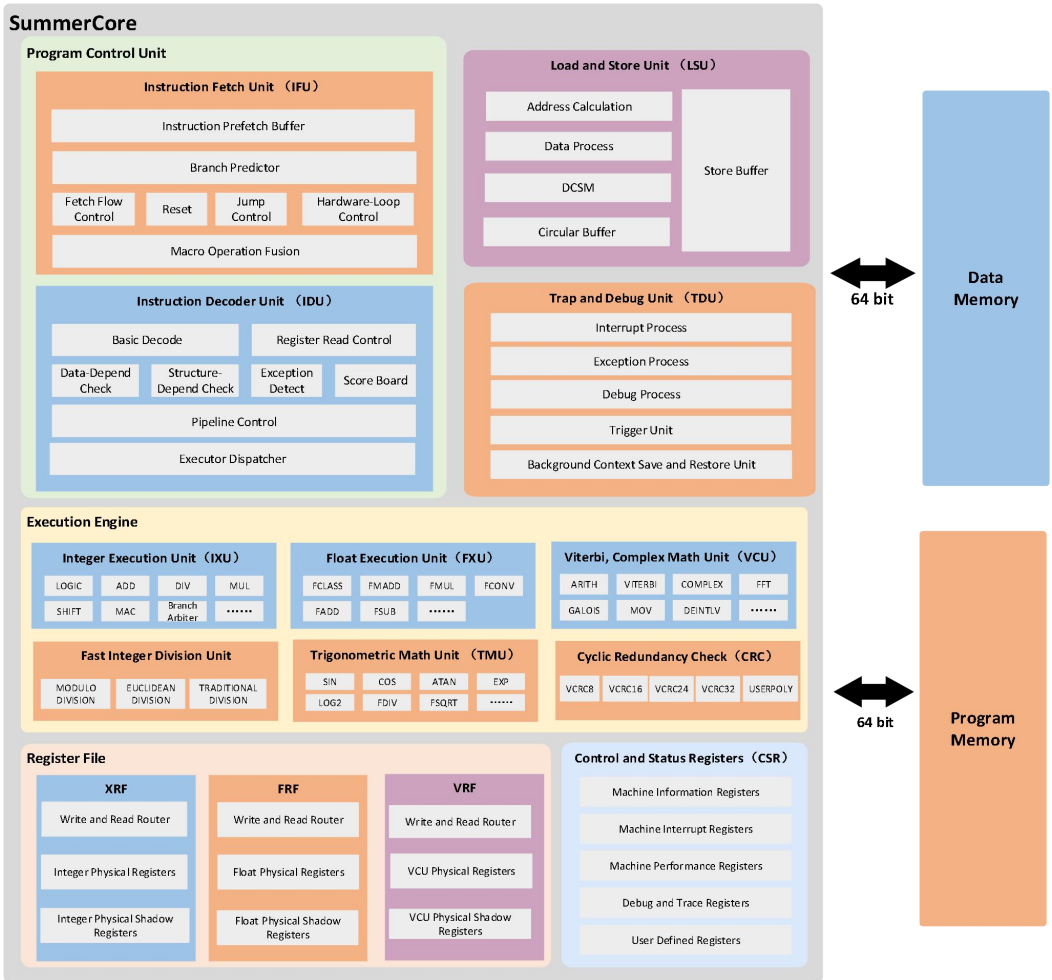



图 1-1 SummerCore 结构图

## 1.2 主要特性

（1）微架构设计

SummerCore 采用增强型静态双发射八级流水线微架构，支持高精度分支预测、零开销硬件循环，采用多种常用模式的指令融合，同时支持访存-运算双发、不同功能部件双发执行。

(2) 专用加速器集成

集成实时控制领域专用加速器，支持三角函数、快速傅里叶变换、循环冗余校验、维特比等专用硬件指令，通过硬件化指令执行替代软件模拟，降低算法延迟。

（3）存储与中断系统

片上集成 ROM、RAM、FLASH 等多种存储资源，采用 64bit 宽总线与多访存口并发设计，保障高带宽数据传输；构建软硬件协同的高效中断处理系统，满足实时控制场景的低延迟需求。

（4）调试与安全体系

为方便对内核状态分析与问题定位，SummerCore 调试系统在兼容 RISC-V Debug 0.13.2 规范的调试机制的基础上，采用四线 JTAG 调试，自定义了 Trace 功能和实时调试功能。

为保护芯片功能和数据安全，SummerCore 的功能安全系统采用双区域代码安全模块，支持 ECC/parity 保护的存储系统，BGCRC，ERAD 等可靠性增强保护组件。数据安全系统构建全场景权限管理与数据保护机制，覆盖芯片使用全生命周期的安全需求。

## 1.3 存储器

### 1.3.1 存储器映射

表 1-1 列出了各存储区域的起止地址、容量及访问属性。

表 1-1 存储器映射

| 名称 | 起始地址 | 结束地址 | 长度 | DMA | HIC | ECC/PARITY | 安全 |
|---|---|---|---|---|---|---|---|
| M0 | 0000_0000 | 0000_0FFF | 1000 | | | ECC | |
| M1 | 0000_1000 | 0000_1FFF | 1000 | | | ECC | |
| LS0 | 0001_0000 | 0001_1FFF | 2000 | | | ECC | 是 |
| LS1 | 0001_2000 | 0001_3FFF | 2000 | | | ECC | 是 |
| LS2 | 0001_4000 | 0001_5FFF | 2000 | | | ECC | 是 |
| LS3 | 0001_6000 | 0001_7FFF | 2000 | | | ECC | 是 |
| GS0 | 000E_0000 | 000E_1FFF | 2000 | 是 | 是 | Parity | |
| FLASH | 0010_0000 | 0013_FFFF | 40000 | | | ECC | 是 |
| FDM OTP | 0030_0000 | 0030_17FF | 1800 | | | | |
| USER OTP | 0030_2000 | 0030_37FF | 1800 | | | ECC | USER OTP |
| SECURE ROM | 0038_0000 | 0038_FFFF | 10000 | | | Parity | 是 |
| BOOT ROM | 0040_0000 | 0041_FFFF | 20000 | | | Parity | |

### 1.3.2 闪存存储器映射

在 BY320RV0025 器件上，提供了一个闪存组。

表 1-2 闪存扇区地址

| 序号 | 区域 | 数据空间 | | | ECC空间 | | |
|---|---|---|---|---|---|---|---|
| | | 起始地址 | 结束地址 | 大小(字节) | 起始地址 | 结束地址 | 大小(字节) |
| 1 | sector0 | 0x0010 0000 | 0x0010 3FFF | 0x4000(16KB) | 0x0031 0000 | 0x0031 07FF | 0x800(2KB) |
| 2 | sector1 | 0x0010 4000 | 0x0010 7FFF | 0x4000(16KB) | 0x0031 0800 | 0x0031 0FFF | 0x800(2KB) |
| 3 | sector2 | 0x0010 8000 | 0x0010 BFFF | 0x4000(16KB) | 0x0031 1000 | 0x0031 17FF | 0x800(2KB) |
| 4 | sector3 | 0x0010 C000 | 0x0010 FFFF | 0x4000(16KB) | 0x0031 1800 | 0x0031 1FFF | 0x800(2KB) |
| 5 | sector4 | 0x0011 0000 | 0x0011 3FFF | 0x4000(16KB) | 0x0031 2000 | 0x0031 27FF | 0x800(2KB) |
| 6 | sector5 | 0x0011 4000 | 0x0011 7FFF | 0x4000(16KB) | 0x0031 2800 | 0x0031 2FFF | 0x800(2KB) |
| 7 | sector6 | 0x0011 8000 | 0x0011 BFFF | 0x4000(16KB) | 0x0031 3000 | 0x0031 37FF | 0x800(2KB) |
| 8 | sector7 | 0x0011 C000 | 0x0011 FFFF | 0x4000(16KB) | 0x0031 3800 | 0x0031 3FFF | 0x800(2KB) |
| 9 | sector8 | 0x0012 0000 | 0x0012 3FFF | 0x4000(16KB) | 0x0031 4000 | 0x0031 47FF | 0x800(2KB) |

| 序号 | 区域 | 数据空间 | | | ECC空间 | | |
|---|---|---|---|---|---|---|---|
| | | 起始地址 | 结束地址 | 大小(字节) | 起始地址 | 结束地址 | 大小(字节) |
| 10 | sector9 | 0x0012 4000 | 0x0012 7FFF | 0x4000(16KB) | 0x0031 4800 | 0x0031 4FFF | 0x800(2KB) |
| 11 | sector10 | 0x0012 8000 | 0x0012 BFFF | 0x4000(16KB) | 0x0031 5000 | 0x0031 57FF | 0x800(2KB) |
| 12 | sector11 | 0x0012 C000 | 0x0012 FFFF | 0x4000(16KB) | 0x0031 5800 | 0x0031 5FFF | 0x800(2KB) |
| 13 | sector12 | 0x0013 0000 | 0x0013 3FFF | 0x4000(16KB) | 0x0031 6000 | 0x0031 67FF | 0x800(2KB) |
| 14 | sector13 | 0x0013 4000 | 0x0013 7FFF | 0x4000(16KB) | 0x0031 6800 | 0x0031 6FFF | 0x800(2KB) |
| 15 | sector14 | 0x0013 8000 | 0x0013 BFFF | 0x4000(16KB) | 0x0031 7000 | 0x0031 77FF | 0x800(2KB) |
| 16 | sector15 | 0x0013 C000 | 0x0013 FFFF | 0x4000(16KB) | 0x0031 7800 | 0x0031 7FFF | 0x800(2KB) |
| 17 | FDM OTP | 0x0030 0000 | 0x0030 17FF | 0x1800(6KB) | 0x0031 8000 | 0x0031 82FF | 0x300(768) |
| 18 | User OTP | 0x0030 2000 | 0x0030 37FF | 0x1800(6KB) | 0x0031 A000 | 0x0031 A2FF | 0x300(768) |

### 1.3.3 外设寄存器内存映射

“外设寄存器存储器映射”表列出了外设寄存器。

表 1-3 外设寄存器存储器映射

| 外设帧 | 外设 | 起始地址 | 结束地址 | 大小 | DMA | HIC | ECC/PARITY |
|---|---|---|---|---|---|---|---|
| PF060_0000 | AdcaResultRegs | 60_0000 | 60_007F | 80 | 是 | 是 | |
| | AdccResultRegs | 60_0080 | 60_00FF | 80 | 是 | 是 | |
| | CpuTimer0Regs | 60_1000 | 60_100F | 10 | | | |
| | CpuTimer1Regs | 60_1010 | 60_101F | 10 | | | |
| | CpuTimer2Regs | 60_1020 | 60_102F | 10 | | | |
| | DmaRegs | 60_2000 | 60_203F | 40 | | | |
| | Dmach1Regs | 60_2040 | 60_207F | 40 | | | |
| | Dmach2Regs | 60_2080 | 60_20BF | 40 | | | |
| | Dmach3Regs | 60_20C0 | 60_20FF | 40 | | | |
| | Dmach4Regs | 60_2100 | 60_213F | 40 | | | |
| | Dmach5Regs | 60_2140 | 60_217F | 40 | | | |
| | Dmach6Regs | 60_2180 | 60_21BF | 40 | | | |
| PF161_0000 | Clb1LogicCfgRegs | 61_0000 | 61_01FF | 200 | 是 | 是 | |
| | Clb1LogicCtrlRegs | 61_0200 | 61_03FF | 200 | 是 | 是 | |
| | Clb1DataExchRegs | 61_0400 | 61_07FF | 400 | 是 | 是 | |

| 外设帧 | 外设 | 起始地址 | 结束地址 | 大小 | DMA | HIC | ECC/PARITY |
|---|---|---|---|---|---|---|---|
| | Clb2LogicCfgRegs | 61_0800 | 61_09FF | 200 | 是 | 是 | |
| | Clb2LogicCtrlRegs | 61_0A00 | 61_0BFF | 200 | 是 | 是 | |
| | Clb2DataExchRegs | 61_0C00 | 61_0FFF | 400 | 是 | 是 | |
| | EPwm1Regs | 61_4000 | 61_41FF | 200 | 是 | 是 | |
| | EPwm2Regs | 61_4200 | 61_43FF | 200 | 是 | 是 | |
| | EPwm3Regs | 61_4400 | 61_45FF | 200 | 是 | 是 | |
| | EPwm4Regs | 61_4600 | 61_47FF | 200 | 是 | 是 | |
| | EPwm5Regs | 61_4800 | 61_49FF | 200 | 是 | 是 | |
| | EPwm6Regs | 61_4A00 | 61_4BFF | 200 | 是 | 是 | |
| | EPwm7Regs | 61_4C00 | 61_4DFF | 200 | 是 | 是 | |
| | EQep1Regs | 61_8000 | 61_807F | 80 | 是 | 是 | |
| | EQep2Regs | 61_8080 | 61_80FF | 80 | 是 | 是 | |
| | ECap1Regs | 61_A000 | 61_A07F | 80 | 是 | 是 | |
| | ECap2Regs | 61_A080 | 61_A0FF | 80 | 是 | 是 | |
| | ECap3Regs | 61_A100 | 61_A13F | 40 | 是 | 是 | |
| | Cmpss1Regs | 61_C000 | 61_C03F | 40 | 是 | 是 | |
| | Cmpss2Regs | 61_C040 | 61_C07F | 40 | 是 | 是 | |
| | Cmpss3Regs | 61_C080 | 61_C0BF | 40 | 是 | 是 | |
| | Cmpss4Regs | 61_C0C0 | 61_C0FF | 40 | 是 | 是 | |
| PF262_0000 | SpiaRegs | 62_0000 | 62_001F | 20 | 是 | 是 | |
| | SpibRegs | 62_0020 | 62_003F | 20 | 是 | 是 | |
| | PmbusaRegs | 62_1000 | 62_103F | 40 | 是 | 是 | |
| | HicRegs | 62_2000 | 62_20FF | 100 | 是 | 是 | |
| | FsiTxaRegs | 62_3000 | 62_30FF | 100 | 是 | 是 | |
| | FsiRxaRegs | 62_3100 | 62_31FF | 100 | 是 | 是 | |
| PF363_0000 | AdcaRegs | 63_0000 | 63_01FF | 200 | | | |
| | AdccRegs | 63_0200 | 63_03FF | 200 | | | |
| PF464_0000 | InputXbarRegs | 64_0000 | 64_003F | | | | |
| | XbarRegs | 64_0040 | 64_005F | | | | |
| | SyncSocRegs | 64_0080 | 64_008F | | | | |
| | CLBInputXbarRegs | 64_00C0 | 64_00FF | | | | |
| | DmaClaSrcSelRegs | 64_0100 | 64_01FF | | | | |
| | EPwmXbarRegs | 64_0200 | 64_027F | | | | |
| | ClbXbarRegs | 64_0280 | 64_02FF | | | | |
| | OutputXbarRegs | 64_0300 | 64_03FF | | | | |
| | CLBOutputXbarRegs | 64_0580 | 64_05FF | | | | |
| | GpioCtrlRegs | 64_0800 | 64_0BFF | 400 | | | |

| 外设帧 | 外设 | 起始地址 | 结束地址 | 大小 | DMA | HIC | ECC/PARITY |
|---|---|---|---|---|---|---|---|
| | GpioDataRegs | 64_0C00 | 64_0CFF | 100 | | 是/仅读 | |
| | GpioDataReadRegs | 64_0D00 | 64_0DFF | 100 | | 是 | |
| PF565_0000 | DEVCFG | 65_0000 | 65_03FF | 400 | | | |
| | CLKCFG | 65_0400 | 65_05FF | 200 | | | |
| | CpuSysRegs | 65_0600 | 65_07FF | 200 | | | |
| | PeriphAcRegs | 65_0800 | 65_0BFF | 400 | | | |
| | AnalogSubsysRegs | 65_1000 | 65_127F | 280 | | | |
| PF767_0000 | FDCAN | 67_0000 | 67_1FFF | 2000 | 是 | 是 | |
| | CanaRegs | 67_2000 | 67_3FFF | 2000 | 是 | 是 | Parity(CAN RAM) |
| | CanaMboxRegs | 67_4000 | 67_4FFF | 1000 | 是 | 是 | |
| | DCC1 | 67_8000 | 67_807F | 80 | | | |
| | DCC2 | 67_8100 | 67_817F | 80 | | | |
| PF868_0000 | LinaRegs | 68_0000 | 68_01FF | 200 | 是 | 是 | |
| | LinbRegs | 68_0200 | 68_03FF | 200 | 是 | 是 | |
| PF969_0000 | I2caRegs | 69_0000 | 69_007F | 80 | | 是 | |
| | I2cbRegs | 69_0080 | 69_00FF | 80 | | 是 | |
| | SciaRegs | 69_1000 | 69_101F | 20 | | 是 | |
| | WDRegs | 69_2000 | 69_207F | 80 | | 是 | |
| | XintRegs | 69_2200 | 69_221F | 20 | | 是 | |
| | NMIIntruptRegs | 69_2400 | 69_241F | 20 | | 是 | |

# 2 系统控制和中断

该微控制器的系统级功能可配置 CPU 与外设的时钟、复位和中断，还可配置片上存储器、定时器及安全功能的运行模式。

## 2.1 介绍

系统级配置由一组子模块控制，这些子模块统称为系统控制模块（system control module）。该系统控制模块具备以下功能：

- 系统级复位，包括上电复位（power-on reset）和掉电复位（brownout reset）
- 时钟源选择与锁相环（PLL）配置
- 时钟丢失检测（missing clock detection）
- 时钟门控低功耗模式（clock-gating low-power modes）
- 外设中断处理（peripheral interrupt handling）
- 针对特定故障情况的不可屏蔽中断（non-maskable interrupts）
- 三个 32 位定时器（32-bit timers）
- 窗口式看门狗定时器（windowed watchdog timer），可生成中断或复位信号
- RAM 初始化、写保护及主控权控制（mastership control）
- 闪存（Flash memory）的纠错码（ECC）、等待状态（wait state）及缓存（cache）配置
- 双区域代码安全模块（dual-zone code security module）

### 2.1.1 系统配置寄存器的 LOCK（锁定）保护

若干系统配置寄存器通过“LOCK（锁定）”寄存器防止受到意外的 CPU 写操作干扰。一旦这些相关联的 LOCK 寄存器位被置位，相应的被锁定寄存器便无法再通过软件进行修改。

## 2.2 电源管理

BY320RV0025 微控制器（MCU）的 1.2V 内核通过低压差稳压器（LDO，即 VREG）实现内部供电。由于 VREG 的电源取自 3.3V 电源轨，因此该器件可支持单 3.3V 电源轨运行。

## 2.3 器件标识与配置寄存器

器件标识寄存器与配置寄存器可提供关于该器件的零件号、产品系列、版本、引脚数量、认证状态及功能可用性的信息。

所有器件信息均属于 DEV_CFG_REGS 寄存器空间（注：DEV_CFG_REGS 为 “Device Configuration Registers” 的缩写，即 “器件配置寄存器”，此处保留原始标识以符合技术文档规范）。其中，标识寄存器包括 PARTIDL（低字节零件标识寄存器）、PARTIDH（高字节零件标识寄存器）和 REVID（版本标识寄存器）。

UID_REGS 寄存器空间（注：UID_REGS 为 “Unique ID Registers” 的缩写，即 “唯一标识寄存器”）中提供了一个 256 位的唯一标识（UID，Unique ID）。这 256 位数据分布在以下寄存器中：

UID_PSRAND0-5：192 位伪随机数据（注：“PS” 为 “Pseudo-Random” 的缩写，指 “伪随机”）。

UID_UNIQUE：32 位唯一数据；该寄存器中的数值在所有具有相同 PARTIDH（高字节零件标识）的器件中均不重复

UID_CHECKSUM：UID_PSRAND0-5 与 UID_UNIQUE 的 32 位弗莱彻校验和（注：Fletcher checksum 为一种差错检测算法），在工厂测试阶段按小端序（little-endian）或大端序（big-endian）计算得出。

## 2.4 复位

本节介绍该器件上不同复位（功能 / 信号）的类型及其作用

### 2.4.1 复位信号

下表汇总了各类复位信号及其对器件的影响。

表 2-1 复位信号

| 复位源 | CPU 内核复位(CPU、FPU、CRC) | 外设复位 | JTAG/调试逻辑复位 | 输入/输出引脚 | $\overline{\text{XRS}}$ 输出 |
|---|---|---|---|---|---|
| POR | Yes | Yes | Yes | Hi-Z | Yes |
| BOR | Yes | Yes | Yes | Hi-Z | Yes |
| $\overline{\text{XRS}}$ Pin | Yes | Yes | No | Hi-Z | - |
| $\overline{\text{WDRS}}$ | Yes | Yes | No | Hi-Z | Yes |
| $\overline{\text{NMIWDRS}}$ | Yes | Yes | No | Hi-Z | Yes |
| SYSRS (Debug Reset) | Yes | Yes | No | Hi-Z | No |
| SIMRESET. XRS | Yes | Yes | No | Hi-Z | Yes |
| SIMRESET. CPU1RS | Yes | Yes | No | Hi-Z | No |

复位可分为两类：

芯片级复位（包括 XRS、POR、BOR、WDRS 和 NMIWDRS）：此类复位会重置器件的全部或几乎全部组件。

系统级复位（包括 SYSRS）：此类复位会重置器件的大部分组件，但会保留部分系统级配置。

复位后，复位原因寄存器（RESC）会更新为相应的复位原因。该寄存器中的各位状态会在多次复位过程中保持不变，且这些位只能通过上电复位（POR）或向复位原因清除寄存器（RESCCLR）写入 1 来清除。其中部分位会由引导只读存储器（boot ROM）在启动程序执行过程中自动清除。

许多外设模块拥有独立的复位功能，可通过软件复位寄存器（SOFTPRESx）进行控制。

无论何种复位，中央处理器（CPU）都会从引导只读存储器（boot ROM）中的地址 0x3FFFC0（即复位向量）开始执行程序。在运行完引导只读存储器（boot ROM）中的代码后，CPU 通常会跳转到闪存（Flash memory）的起始地址 0x100000 继续执行。如需了解引导过程的控制方法。

### 2.4.2 外部复位（XRS）

外部复位（XRS）是该器件的主芯片级复位。XRS 会复位中央处理器（CPU）、所有外设及输入 / 输出（I/O）引脚配置，以及大部分系统控制寄存器。XRS 拥有一个专用的开漏输出引脚，该引脚可用于驱动应用中其他集成电路（IC）的复位引脚，也可由外部信号源驱动。在看门狗复位、不可屏蔽中断（NMI）复位和上电复位（POR）期间，XRS 会由器件内部驱动。

无论因何种原因导致 XRS 被拉低，复位原因寄存器（RESC）中的 XRSn 位都会被置位，随后该位会由引导只读存储器（boot ROM）清除。

### 2.4.3 模拟外部复位

用户可通过软件模拟外部复位（XRS）。具体操作方式为：通过软件将 SIMRESET 寄存器中的 XRSn 位置 1。此操作会触发 XRS 引脚电平翻转，并对整个器件执行复位（效果与外部复位完全一致）。

此次复位后，复位原因寄存器（RESC）中的 SIMRESET_XRSn 位和 XRSn 位会被置位。软件可读取这两个位以确认复位原因，并通过向复位原因清除寄存器（RESCCLR）中对应的位写入 1 来清除该复位状态。

### 2.4.4 上电复位（POR）

上电复位（POR）电路会在器件上电过程中为整个器件生成稳定的复位信号，同时抑制通用输入输出接口（GPIO）上的干扰脉冲。在整个上电复位（POR）期间，外部复位（XRS）引脚会保持低电平。在大多数应用场景中，XRS 引脚保持低电平的时长足以复位系统中的其他集成电路（IC），但部分应用可能需要更长的复位脉冲。这种情况下，可通过外部驱动使 XRS 引脚保持低电平，以确保复位时长符合需求。

上电复位（POR）不仅能复位外部复位（XRS）可复位的所有组件，还会复位另外几个寄存器，包括复位原因寄存器（RESC）、不可屏蔽中断影子标志寄存器

（NMISHDFLG）以及 X1 时钟计数器寄存器（X1CNT）。此外，上电复位（POR）还会复位联合测试行动小组（JTAG）接口所使用的调试逻辑。

上电复位（POR）完成后， RESC 中的 POR 位和 XRSn 位会被置位，随后这两个位将由引导只读存储器（boot ROM）清除。

### 2.4.5 模拟 CPU 复位

用户可通过软件模拟中央处理器（CPU）复位（即系统级复位 SYSRS）。具体操作方式为：通过 CPU1 的软件将 SIMRESET 寄存器中的 CPU1RSn 位置 1。此操作会触发 CPU1 的 SYSRS 信号电平翻转，并对 CPU 执行复位（效果与调试器复位完全一致）。

此次复位后，复位原因寄存器（RESC）中的 SIMRESET_CPU1RSn 位会被置位。软件可读取该位以确认复位原因，并通过向复位原因清除寄存器（RESCCLR）中对应的位写入 1 来清除该复位状态。

### 2.4.6 看门狗复位（WDRS）

该器件配备了一个看门狗定时器。若中央处理器（CPU）未在用户指定的时间内为看门狗定时器提供服务（即 “喂狗”），该定时器可选择触发一次复位。此次看门狗复位（WDRS）会产生一个外部复位（XRS）信号，该信号的持续时间为 512 个内部振荡器 1（INTOSC1）周期。

看门狗复位后，复位原因寄存器（RESC）中的 WDRSn 位和 XRSn 位会被置位。

### 2.4.7 不可屏蔽中断看门狗复位（NMIWDRS）

该器件配备了一个不可屏蔽中断（NMI）模块，用于检测系统中的硬件错误。该 NMI 模块内置一个看门狗定时器，若中央处理器（CPU）未在用户指定的时间内对错误做出响应，该定时器将触发一次复位。此次不可屏蔽中断看门狗复位

（NMIWDRS）会产生一个外部复位（XRS）信号，该信号的持续时间为 512 个内部振荡器 1（INTOSC1）周期。

不可屏蔽中断看门狗复位后，复位原因寄存器（RESC）中的 NMIWDRSn 位和 XRSn 位会被置位。

## 2.5 外设中断

### 2.5.1 中断概念

中断是一种信号，它会使 CPU 暂停当前的执行过程，并跳转到一段被称为中断服务程序（ISR）的不同代码。这是一种处理外设事件的有效机制，相比寄存器轮询方式，它占用的 CPU 开销更少，程序复杂度也更低。然而，由于中断与程序流是异步的，必须小心避免中断和主程序代码访问同一资源时产生冲突。

中断通过一系列标志寄存器和使能寄存器传递到 CPU。标志寄存器会保存中断信号，直到该中断被处理。使能寄存器则可阻止中断的传递。当中断信号到达 CPU 时，CPU 会从一个名为向量表的列表中获取相应的中断服务程序（ISR）地址。

### 2.5.2 中断架构

RISC-V 架构定义的中断类型有 4 种：外部中断（external interrupt）、定时器中断（timer interrupt）、软件中断（software interrupt）和调试中断（debug interrupt）。其中内核中断控制器（CLIC, Core-local Interrupt Controller）是 RISC-V 架构标准定义的系统中断控制器，主要用于多个外部中断源的优先级仲裁和派发，理论上最多可以支持 4096 外部中断源，根据内核的状态和中断的优先级实现中断仲裁，使得内核服务于最紧急的中断事件，同时内核通过总线配置 CLIC 的配置寄存器产生定时器中断和软件中断。

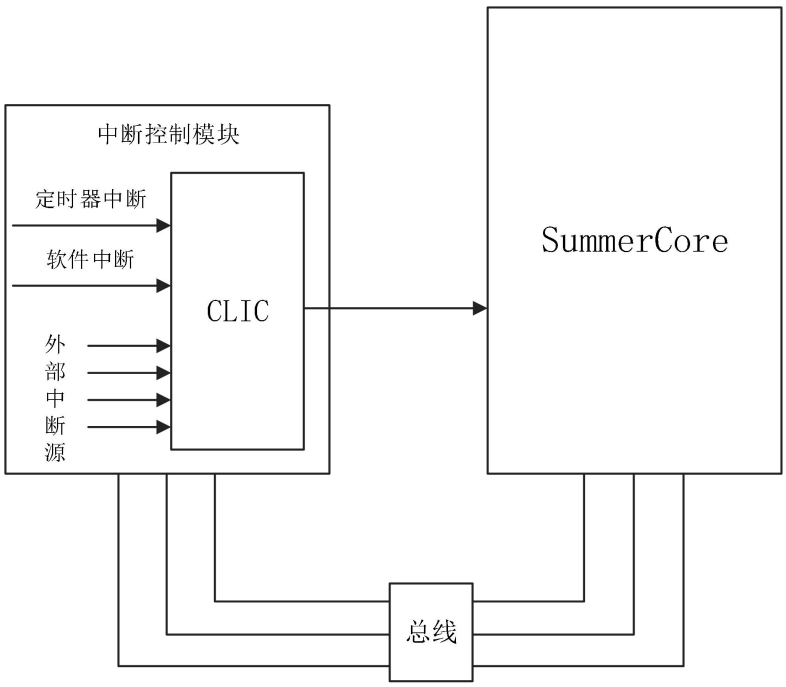



图 2-1 CLIC 框图

### 2.5.3 技术特性

1) 中断仲裁机制

- CLIC 采用多级仲裁机制，中断级别（Interrupt Level）作为第一仲裁条件，中断优先级（Interrupt Priority）作为第二仲裁条件，中断 ID（Interrupt ID）作为第三仲裁条件。当多个中断请求同时满足响应条件时，优先选择中断级别更高的请求；若中断级别相同，则比较中断优先级；若二者均相同，则根据中断 ID 进行仲裁。

2) “水平”中断抢占

- 在支持 CLIC 的特权模式下，中断级别用于表示中断抢占优先级。CLIC 最多支持 16 个中断级别（level 0~15，实际支持数量由配置参数决定）。其中 level 0 表示最低中断级别，level 1~15 表示不同的中断服务级别。当当前正在执行低级别 ISR 时，如果产生更高中断级别的请求，并满足中断使能条件，则高级别中断可以抢占当前 ISR，实现中断嵌套。

3) 选择性中断硬件向量化

- 对实时性要求高的外设中断，可以采取硬件向量化；
- 其余可采取非硬件向量化的方式，通过统一入口软件完成中断分发。

4) 支持电平触发、边沿触发的中断源信号检测

- 对于电平触发中断，当中断输入保持有效电平时，中断控制器保持该中断处于 pending 状态。进入 ISR 后，软件需要清除外设内部中断状态，使中断输入恢复无效电平；若清除后输入再次变为有效电平，则该中断会再次进入 pending 状态并触发新的 ISR。
- 对于边沿触发中断，中断控制器检测到有效边沿后产生一次中断请求，并置位 pending 状态。在进入 ISR 前产生的多个有效边沿通常会被合并为一次 pending 请求，ISR 执行一次；若 ISR 执行期间再次检测到有效边沿，则根据中断控制器实现方式重新置位 pending，并在 ISR 返回后再次响应。

## 2.6 异常和不可屏蔽中断

本节描述可能触发不可屏蔽中断（NMI）的系统级错误条件。该中断使应用程序能够响应错误，不可屏蔽中断（NMI）分为两类可恢复不可屏蔽中断(RNMI)和不可恢复不可屏蔽中断(UNMI)。

### 2.6.1 配置和使用 NMI

一个传入的 NMI 会在 NMIFLG 寄存器中设置一个状态位，并启动 NMI 看门狗计数器。这个计数器由 SYSCLK 时钟驱动，如果计数器达到 NMIWDPRD 寄存器中的值，计数器会触发 NMI 看门狗复位（NMIWDRS）。为了防止这种情况，NMI 处理程序必须使用 NMIFLGCLR 寄存器清除标志位。一旦所有标志位都被清除， NMIFLG 寄存器中的 NMIINT 位也可以被清除，以便允许未来的 NMI 请求。NMI 模块在启动过程中由引导 ROM 启用。为了响应 NMI，必须将 NMI 处理程序向量写入 NMI 处理函数入口地址。

### 2.6.2 仿真注意事项

在调试（debug）条件下，NMI 看门狗计数器的行为如下：

1）CPU 被挂起 / HALT

当 CPU 被挂起（Suspend）时，NMI 看门狗计数器也会被挂起（停止计数）。

2）自由运行模式

当 CPU 被设置为 Run-Free 模式时，NMI 看门狗计数器恢复正常运行。

3）实时单步模式

当 CPU 处于实时单步模式时，NMI 看门狗计数器会被挂起。

4）实时自由运行模式

当 CPU 处于实时自由运行模式时，NMI 看门狗计数器正常运行。

### 2.6.3 NMI 触发源

有多种类型的硬件错误都可以触发 NMI。关于具体错误的更多信息，通常可以从检测到该错误的模块中获取。

#### 2.6.3.1 时钟丢失检测

时钟丢失检测逻辑用于监控 OSCCLK 是否发生故障。如果 OSCCLK 时钟源停止工作，PLL 将被旁路，OSCCLK 会连接到 INTOSC1，并向 CPU 触发一个 RNMI。

#### 2.6.3.2 RAM 不可纠正错误

在 RAM 读取过程中，如果发生单比特奇偶校验错误、双比特 ECC 数据错误，将触发 RNMI，对应寄存器NMIFLG[2]，也可以选择配置NMITYPE寄存器，若触发会进入 UNMI 中断函数，程序无法返回。此规则适用于 CPU 、DMA 和 HIC 读取。单比特 ECC 数据错误不会触发 NMI，但可以选择触发一个普通外设中断。

#### 2.6.3.3 Flash 不可纠正的 ECC 错误

在 Flash 读取过程中，如果发生双比特 ECC 数据错误，将触发 RNMI。单比特 ECC 数据错误不会触发 NMI，但可以选择触发一个普通外设中断。

#### 2.6.3.4 软件强制错误

存在一个特殊的 NMI 源，只能通过向 NMIFLGFRC 寄存器中的 SWERR 位写入来触发。由于 SWERR 标志永远不会由真实的硬件故障置位，因此可以用于实现 RNMI 子系统的自测试模式。

#### 2.6.3.5 BGCRC 校验失败

BGCRC 模块的 CRC 失败结果可以生成 RNMI。默认情况下，该 NMI 是启用的。若要禁用此功能，必须在 BGCRC_CTRL1 寄存器的 NMIDIS 配置字段中写入 “1010”。

#### 2.6.3.6 ERAD 触发的 NMI

ERAD 模块可以基于不同事件生成 RNMI。该功能可 ERAD 寄存器中进行配置。

#### 2.6.3.7 CLB 触发的 NMI

CLB 模块可以基于不同事件生成 RNMI。该功能可在 CLB 寄存器中进行配置。

#### 2.6.3.8 错误引脚（Error Pin）

一个名为 ERRORSTS 的信号可以输出到 GPIO24、GPIO28 或 GPIO29。

当 NMI 影子标志寄存器（NMISHDFLG）中的任意一位被置位时，该信号会变为低电平。

它可用于向外部系统发出微控制器内部出现问题的告警信号。

由于 ERRORSTS 的状态基于影子标志位，因此在 CPU 清除这些标志位之前，或者发生上电复位之前，ERRORSTS 会一直保持低电平。

所有 GPIO 引脚在上电时默认配置为输入模式。如果在上电期间所选 ERRORSTS 引脚的电平状态很重要，则应在该引脚上外接一个下拉电阻。

## 2.7 时钟

本节介绍本器件的时钟源和时钟域，以及如何对其进行配置以供应用使用。下图概述了该器件的时钟系统。

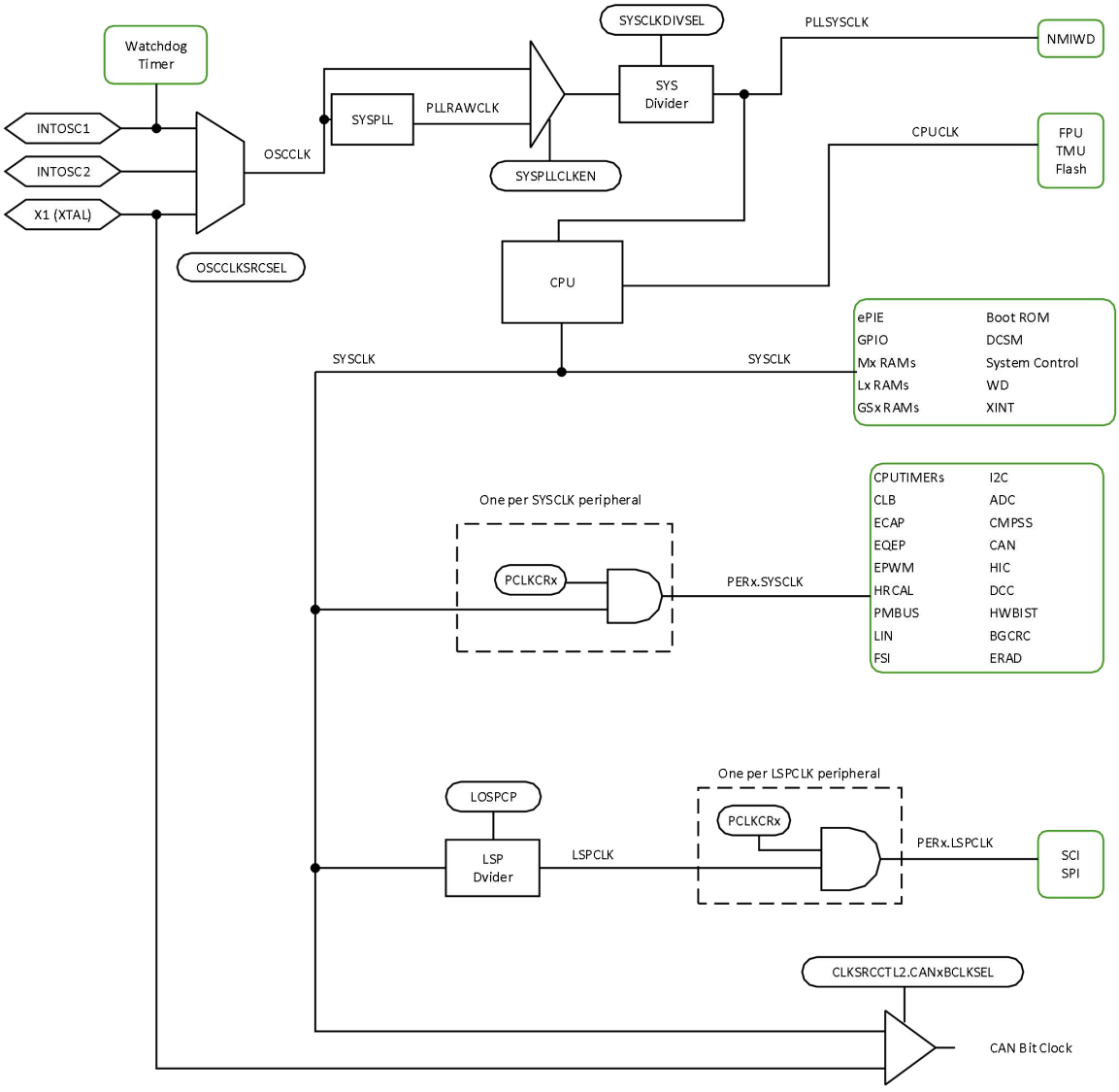
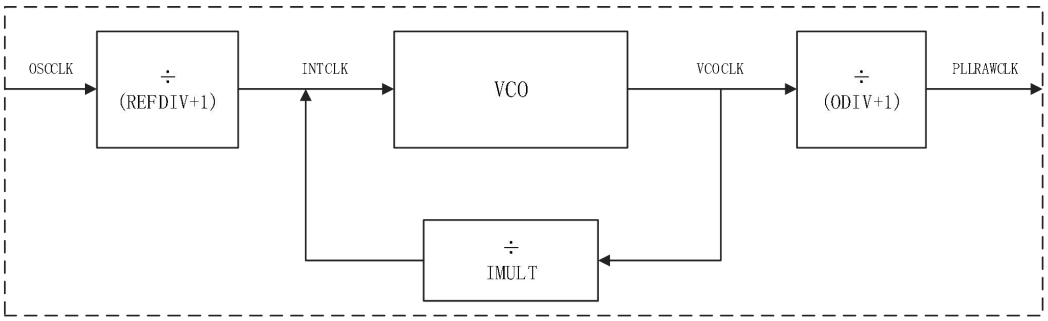



图 2-2 时钟系统

SYSPLL




fPLLRAWCLK = fOSCCLK ÷ (REFDIV + 1) × IMULT ÷ (ODIV + 1)

图 2-3 PLL 系统

### 2.7.1 时钟源（Clock Sources）

本器件中的所有时钟都来源于以下四种时钟源之一.

#### 2.7.1.1 主内部振荡器（INTOSC2）

在上电时，器件由片上 10 MHz 振荡器（INTOSC2）提供时钟。

INTOSC2 是主要的内部时钟源，并且在复位时作为默认的系统时钟。INTOSC2 用于运行 Boot ROM，也可以作为应用程序的系统时钟源。

需要注意的是，INTOSC2 的频率容差较大，无法满足 CAN 的时序要求。因此，使用 CAN 模块时必须使用外部振荡器。

当 INTOSC2 作为系统时钟源时，GPIO19（X1）和 GPIO18（X2）可以作为普通 GPIO 引脚使用。

#### 2.7.1.2 备用内部振荡器（INTOSC1）

器件还包含一个冗余的片上 10 MHz 振荡器（INTOSC1）。

INTOSC1 是备用时钟源，通常仅用于为看门狗定时器和时钟丢失检测电路（MCD）提供时钟。

如果启用了 MCD 且检测到系统时钟丢失：

系统 PLL 会被旁路

所有系统时钟会自动切换连接到 INTOSC1

此外，INTOSC1 也可以出于调试目的被手动选择为系统时钟源。

### 2.7.1.3 外部振荡器（XTAL）

器件支持外部时钟源（XTAL），可用作：

主系统时钟源

CAN 位时钟源

外部时钟源使用 X1/GPIO19 和 X2/GPIO18 引脚。

支持三种外部时钟源。
单端 3.3 V 外部时钟：时钟信号必须连接到 X1。

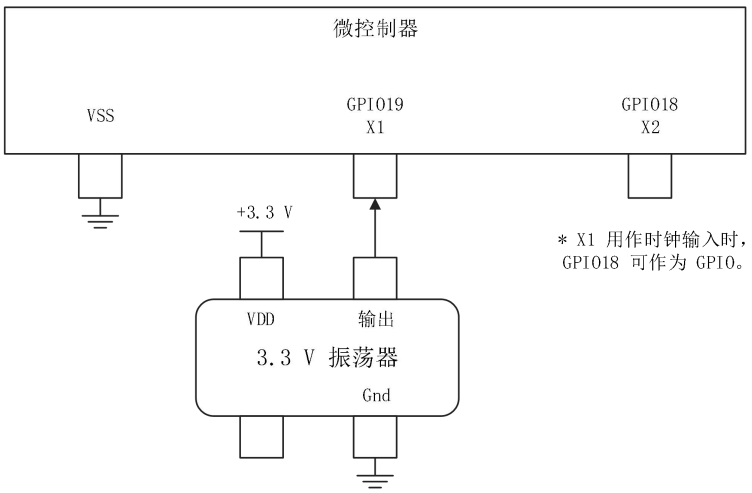



图 2-4 外部振荡器（XTAL）示意图

外部晶体：晶体必须跨接在 X1 和 X2 之间，
负载电容连接到 VSS。

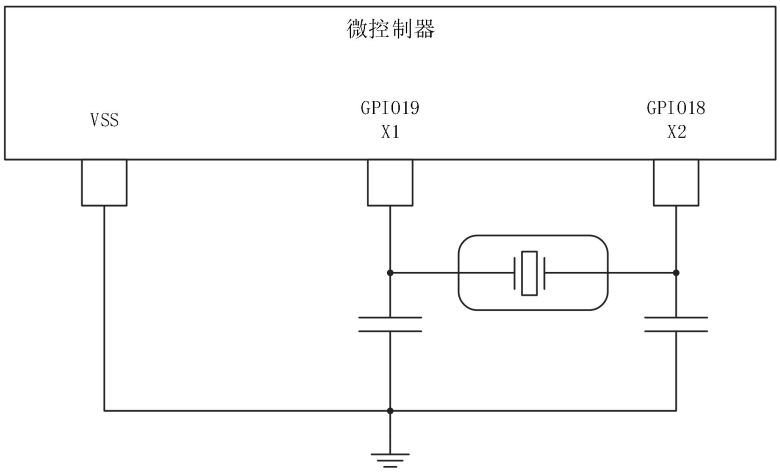



图 2-5 外部时钟

外部谐振器：谐振器必须跨接在 X1 和 X2 之间，
接地端连接到 VSS。

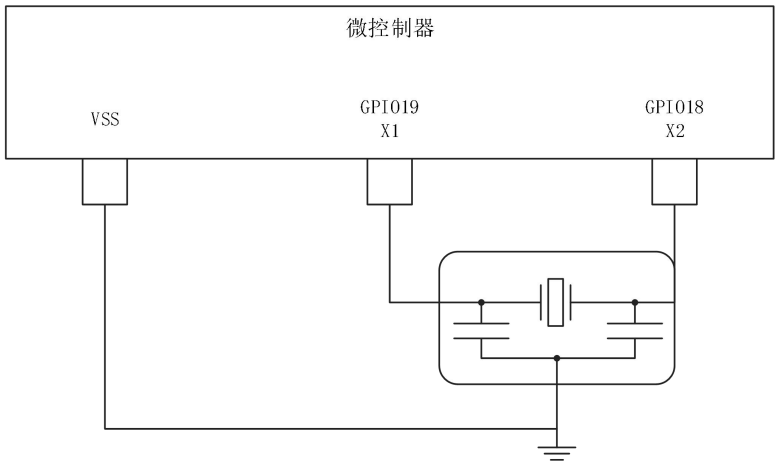



图 2-6 外部系统

表 2-2 配置

| XTALCR位 | | 工作模式 | GPIO19在X1上是否可用 | GPIO18在X2上是否可用 |
|---|---|---|---|---|
| OSCOFF | SE | | | |
| 0 | 0 | 晶体模式：石英晶体连接X1和X2 | 否 | 否 |
| 0 | 1 | 单端模式：外部晶振连接X1 | 否 | 是 |
| 1 | 0 | 振荡器关闭 | 是 | 是 |
| 1 | 1 | 单端模式：外部晶振连接X1 | 否 | 是 |

说明：

(1) OSCOFF 和 SE 位决定 GPIO18 与 GPIO19 的 ALT（复用）模式。

(2) 在该模式下，X1 引脚上存在约 1 kΩ 下拉电阻，因此外部单端时钟源必须能够驱动该负载。

## 2.7.2 派生时钟（Derived Clocks）

上一节介绍的时钟源可以通过 PLL 倍频 或 分频 来生成应用所需的时钟频率。该过程会产生一组派生时钟，本节对此进行说明。

### 2.7.2.1 振荡器时钟（OSCCLK）

必须从 INTOSC2、XTAL 或 INTOSC1 中选择一个作为 CPU 和大多数外设的参考控制时钟（OSCCLK）。

OSCCLK 可以：直接使用或经过系统 PLL 提高频率后使用。

在复位时，OSCCLK 默认作为系统时钟，并连接到 INTOSC2。

#### 2.7.2.2 系统 PLL 输出时钟（PLLRAWCLK）

系统 PLL 允许设备以最大额定工作频率运行，并且在大多数应用生成主系统时钟。该 PLL 使用 OSCCLK 作为参考。PLLRAWCLK 是 PLL 的电压控制振荡器(VCO)的输出。

### 2.7.3 器件时钟域（Device Clock Domains）

器件中的各个模块通过时钟域获得时钟输入。这些时钟域直接或通过额外分频器连接到派生时钟。

#### 2.7.3.1 系统时钟（PLLSYSCLK）

NMI 看门狗定时器有自己的时钟域 PLLSYSCLK。尽管名字如此，PLLSYSCLK 可以连接到系统 PLL(PLLRAWCLK)或 OSCCLK(系统振荡器时钟)。选择的时钟源会通过一个分频器运行，该分频器的配置由 SYSCLKDIVSEL 寄存器控制。

在 HALT 模式下，PLLSYSCLK 会被门控关闭。

#### 2.7.3.2 CPU 时钟（CPUCLK）

CPU 时钟(CPUCLK)用于为 CPU、FLASH 提供时钟信号，该时钟和 PLLSYSCLK 相同。

当 CPU 进入 IDLE 或 STANDBY 或 HALT 模式时，CPUCLK 会被关断。

#### 2.7.3.3 CPU 子系统时钟（SYSCLK 与 PERx.SYSCLK）

CPU 子系统层和外设时钟。

PERx.SYSCLK 等同于 SYSCLK 时钟，每个外设时钟都有一个独立的时钟门控，由 PCLKCRx 寄存器控制。该时钟控制子系统内部 ERAD、BGCRC 模块和 SOC 其余外设。

当 CPU 进入 HALT 或 STANDBY 模式时，SYSCLK 会被禁用。

注意:使用 PCLKCRx 时，应用程序在使能外设的时钟后需要等待 5 个 SYSCLK 周期。

### 2.7.3.4 低速外设时钟（LSPCLK 与 PERx.LSPCLK）

SCI 和 SPI 模块的通信速率远低于 CPU 频率。这些模块连接到一个共享的时钟分频器该分频器从 SYSCLK 生成一个低速外设时钟(LSPCLK)。LSPCLK 默认使用/4 分频器(即，SYSCLK 的频率除以 4)，但可以通过 LOSPCP 寄存器更改分频比。

每个 SCl 和 SPI 模块的时钟(PERX.LSPCLK)可以通过 PCLKCRx 寄存器独立地进行门控.

### 2.7.3.5 CAN 位时钟

CAN 位时钟所需的频率值差异取决于位时序设置和网络配置，可以差值小至 0.1%。由于主系统时钟(以 PERx.SYSCLK 的形式)可能不够精确，CAN 位时钟也可以通过 CLKSRCCTL2 寄存器连接到 XTAL。每个 CAN 模块都有独立的选择。

为保证正确操作，CAN 位时钟的频率必须小于或等于 SYSCLK 频率。

### 2.7.3.6 CPU Timer2 时钟（TIMER2CLK）

CPU 定时器 0 和 1 连接到 PERx.SYSCLK 。Timer2 默认连接到 PERx.SYSCLK，但也可以使用 TMR2CLKCTL 寄存器连接到 INTOSC1、INTOSC2 或 XTAL。该寄存器还为定时器 2 提供了一个单独的预分频器。

如果使用非 SYSCLK 源，则必须将源分频到不超过 SYSCLK 频率的一半。使用非 SYSCLK 源的主要原因是为了内部频率测量。在大多数应用程序中，定时器 2 从 SYSCLK 运行。

### 2.7.4 外部时钟输出

外部时钟输出(XCLKOUT)功能通过将时钟连接到外部引脚(可以是 GPIO16 或 GPIO18),以便于支持调试和测试。可用的时钟源包括 PLLSYSCLK、PLLRAWCLK、SYSCLK 、 INTOSC1 、 INTOSC2 和 XTAL 。 使 用 XCLKOUT ， 首 先 使 用 CLKSRCCTL3 寄存器选择时钟源。接下来，使用 XCLKOUTDIVSEL 寄存器选择所需的输出分频器。最后，使用 GPIO 配置寄存器将 GPIO16 或 GPIO18 连接到多路复用器通道 11。

### 2.7.5 时钟连接

表 2-3 时钟连接

| 时钟 | 模块名称 |
|---|---|
| CPUCLK | CPU |
| | Flash |
| SYSCLK | CLIC |
| | DCSM |
| | M0-M1 RAM |
| | LS0-LS3 RAM |
| | GS0 RAM |
| | Boot ROM |
| | Secure ROM |
| | Memory Control |
| | Debug Module |
| PLLSYSCLK | NMIWD |
| PERx.SYSCLKw | CLB1-2 |
| | CPUTimer0-2 |
| | DCC0-1 |
| | FSI |
| | ePWM1-7 |
| | eCAP1-3 |
| | eQEP1-2 |
| | ADCA,C |
| | CMPSS1-4 |
| | FDCAN |
| | CANA |
| | I2CA-B |
| | PMBUSA |
| | LINA-B |
| | HICA |
| | BGCRC |
| | ERAD |
| PERx.LSPCLK | SCIA |
| | SPIA-B |

| 时钟 | 模块名称 |
| --- | --- |
| WDCLK(INTOSC1) | Watchdog Timer |

### 2.7.6 时钟源与 PLL 配置

应用需求最终决定时钟配置方式。诸如应用性能、功耗、系统总体成本以及EMC（电磁兼容性）等具体因素超出了本文档的讨论范围，但这些因素可以帮助回答以下问题：

1.所需的 CPU 工作频率是多少？

2.是否需要 CAN 功能？

3.可使用哪些类型的外部振荡器或时钟源？

如果需要 CAN 功能，则必须使用具有高精度频率的外部时钟源作为参考时钟。否则，可以仅使用 INTOSC2，从而避免增加额外的外部器件。

### 2.7.7 使用外部晶体或谐振器

X1 和 X2 引脚同时复用为 GPIO19 和 GPIO18。

上电时，这些引脚处于 GPIO 模式，片上晶体振荡器处于关闭状态。

可通过以下步骤将引脚切换为 X1/X2 模式并使能振荡器：

1.清除 XTALCR.OSCOFF 位。

2.等待晶体振荡器上电稳定。典型等待时间为 1 ms，但具体取决于所使用的晶体。

3.通过向 X1CNT.CLR 先写入 1 再写入 0 来清除 X1 计数器，并持续清除，直到 X1CNT 寄存器中的计数值不再饱和为 2047 (0x7FF)。

4.等待 X1CNT 寄存器中的计数值达到 2047 (0x7FF)。

5.将步骤 3–4 再重复执行三次。

6.通过向 CLKSRCCTL1.OSCCLKSRCSEL 写入 1，选择 XTAL 作为 OSCCLK 时钟源。

7.检查 MCDCR 寄存器中的 MCLKSTS 位。

8.如果该位被置位，说明振荡器尚未完成启动，需要更多时间：

a. 向 MCDCR.MCLKCLR 写入 1 以清除缺失时钟状态；

b. 重复步骤 2–7。不要复位器件，否则振荡器会被关闭，需要从步骤 1 重新开始；

c. 如果振荡器在 10 ms 内仍未完成启动，则说明存在真实的时钟故障。

9.如果 MCDCR.MCLKSTS 为 0，则表示振荡器启动成功，此时系统时钟由 XTAL 提供。

#### 2.7.7.1 X1/X2 预处理电路（X1/X2 Precondition Circuit）

X1/X2 上 GPIO19/18 的复用功能可用于在需要时将晶体启动时间最多缩短 30%。该功能通过在使能 XTAL 之前，将负载电容 CL1 和 CL2 预充至已知状态来实现。

在开启 XTAL 之前，对 X1/X2 进行预处理的步骤如下：

```
ClkCfgRegs.XTALCR2.bit.XIF = 1;// 将 X1 预置为高电平
ClkCfgRegs.XTALCR2.bit.XOF = 1;// 将 X2 预置为高电平
ClkCfgRegs.XTALCR2.bit.FEN = 1;// 使能 X1/X2 预处理功能
DEVICE_DELAY_US(1);
ClkCfgRegs.XTALCR.bit.OSCOFF = 0;// 取消预处理并开启 XTAL
ClkCfgRegs.XTALCR2.bit.FEN = 0;// 禁用 X1/X2 预处理功能。
```

### 2.7.8 使用外部振荡器（Using an External Oscillator）

使用连接到 X1 引脚的外部振荡器，其配置流程与使用晶体或谐振器的过程类似：

1.清除 XTALCR.OSCOFF 位。

2.设置 XTALCR.SE 位以使能单端模式（Single-Ended Mode）。

3.通过向 X1CNT.CLR 先写入 1 再写入 0 来清除 X1 计数器，并持续清除，直到 X1CNT 寄存器中的计数值不再饱和为 2047 (0x7FF)。

4.等待 X1CNT 寄存器中的计数值达到 2047 (0x7FF)。

5.将步骤 3–4 再重复执行三次。

6.通过向 CLKSRCCTL1.OSCCLKSRCSEL 写入 1，选择 XTAL 作为 OSCCLK 时钟源。

7.检查 MCDCR 寄存器中的 MCLKSTS 位。如果该位被置位，则表示外部振荡器或器件发生故障。

8.如果 MCLKSTS 为 0，则说明成功切换到外部时钟，此时系统时钟由 XTAL 提供。

### 2.7.9 选择 PLL 参数（Choosing PLL Settings）

IMULT：倍频器的整数倍数值

REFDIV：OSCCLK 的参考分频器

ODIV：PLLRAWCLK 的输出分频器

PLLSYSCLKDIV：系统时钟分频器

多种倍频与分频组合可以产生相同的输出频率。然而，参考时钟频率与倍频系数的乘积（称为 VCO 频率）必须位于数据手册规定的范围内。

### 2.7.10 系统时钟配置（System Clock Setup）

在明确应用需求后，可以确定具体的时钟配置方案。默认配置为使用 INTOSC2 作为系统时钟（PLLSYSCLK），分频系数为 1。

必须按照以下步骤完成所需的应用时钟配置：

可参考 Ware 安装目录中的 SysCtl_setClock() 函数示例。

推荐的系统 PLL 配置流程

1.通过清除 SYSPLLCTL1[PLLCLKEN] 旁路 PLL。至少等待 60 条 NOP 指令 使其生效。

2.通过设置 SYSPLLCTL1.PLLEN = 0 关闭 PLL 电源。至少等待 60 条 NOP 指令 使其生效。

3.通过设置 CLKSRCCTL1.OSCCLKSRCSEL 选择参考时钟源（OSCCLK）。至少等待 300 条 NOP 指令 使其生效。

4.通过清除 SYSCLKDIVSEL[PLLSYSCLKDIV]，将系统时钟分频设置为 /1，以确保 PLL 处于最高性能配置。

5.一次性向 SYSPLLMULT 写入 32 位数值，同时设置 IMULT、REFDIV 和 ODIV。此操作会自动使能 PLL。

6.通过轮询锁定状态位等待 PLL 锁定，即：SYSPLLSTS.LOCKS = 1。

7.使用 DCC 模块进行配置：

参考时钟选择 OSCCLK

被测时钟选择 PLLRAWCLK

验证 PLL 的输出频率。如果频率超出范围，则不要将 PLLRAWCLK 作为 SYSCLK，应在此停止并进行问题排查。

8.通过设置 SYSPLLCTL1[PLLCLKEN]，将 PLL 切换为系统时钟源。

注意事项：

1.在改变 OSCCLK 时钟源之前，必须手动旁路并关闭 SYSPLL。

2.在旁路 PLL（SYSPLLCTL1.PLLCLKEN = 0）后，至少需要 60 个 CPU 时钟周期延时。

3.在关闭 PLL（SYSPLLCTL1.PLLEN = 0）后，至少需要 60 个 CPU 时钟周期延时。

4.在改变 OSCCLK 时钟源后，至少需要 300 个 CPU 时钟周期延时。

5.不支持 PLL SLIP 位，必须使用 DCC 来检测 PLL 时钟的有效性。该功能已包含在 Ware 的 SysCtl_setClock() 函数中。

### 2.7.11 SYS PLL 旁路（Bypass）

如果应用需要将 PLL 时钟从系统中旁路，应将 SYSPLLCTL1.PLLCLKEN = 0 进行配置。从设置该位开始，到旁路真正生效，最长可能需要 60 个 CPU 时钟周期。

在此期间，如果将 PLLSYSCLKDIV 调整为更小的分频值（例如从 /2 改为 /1，或从 /4 改为 /2），器件的工作时钟频率可能会超过其最大额定频率，从而导致不可预测的器件行为。

因此，当 PLL 从使能状态进行旁路（即 PLLCLKEN 从 1 变为 0）后，必须等待 60 个 CPU 时钟周期的延时。

### 2.7.12 时钟（OSCCLK）故障检测

为了实现安全诊断功能，可以使用 缺失时钟检测（Missing Clock Detection，MCD）。

表 2-4 时钟检测

| 时钟故障检测电路 | 被检测的时钟 | 检测所需时间（周期数） | 限制 |
|---|---|---|---|
| 缺失时钟检测（MCD） | INTOSC2、XTAL/X1 | 8192 个 INTOSC1 周期 | 无法检测 INTOSC1 时钟故障 |

#### 2.7.12.1 缺失时钟检测（Missing Clock Detection）

缺失时钟检测（MCD）逻辑使用 INTOSC1 作为参考时钟源，用于检测 OSCCLK 故障。该电路只能检测 OSCCLK 完全丢失 的情况，不能检测 OSCCLK 的频率漂移。

该电路使用由 INTOSC1（次级时钟）提供的 10 MHz 时钟 作为备用时钟，对 OSCCLK（主时钟） 进行监测，其工作过程如下：

1）主时钟（OSCCLK）驱动一个 7 位计数器（MCDPCNT） 计数。该计数器在 XRSn 信号作用下进行异步复位。

2）次级时钟（INTOSC1）驱动一个 13 位计数器（MCDSCNT） 计数。该计数器同样在 XRSn 信号作用下进行异步复位。

3）每当 MCDPCNT 发生溢出时，MCDSCNT 计数器会被复位。因此，只要 OSCCLK 存在，或其频率不低于 INTOSC1 的 1/64，MCDSCNT 就不会发生溢出。

4）如果 OSCCLK 因某种原因停止，或其频率至少比 INTOSC1 慢 64 倍，则 MCDSCNT 会发生溢出，从而检测到 OSCCLK 缺失。

5）除非通过 MCDCR 寄存器（将 MCLKOFF 位置 1）禁用 MCD，否则上述检测会持续运行。

6）一旦检测到 OSCCLK 缺失，将发生以下情况：

- MCDSTS 标志位置位
- MCDSCNT 计数器被冻结，以防止进一步的缺失时钟检测
- CLOCKFAIL 信号拉高，从而：向 PWM 模块产生 TRIP 事件，向 CPU1.NMIWD 触发 NMI 中断
- PLL 被强制旁路，OSCCLK 时钟源切换为 INTOSC1（新的系统时钟频率 = INTOSC1 频率 10 MHz / SYSDIV）。在切换至 INTOSC1 的过程中，系统运行在 PLL limp clock
- SYSPLLMULT.IMULT 自动被清零
- 当 MCDSTS 位为 1 时，OSCCLKSRCSEL 位无效，OSCCLK 被强制连接到 INTOSC1
- 送往系统的 PLLRAWCLK 自动切换为 INTOSC1

7）若向 MCLKCLR 位写 1（W=1 位）：清除 MCDSTS 标志位，OSCCLK 时钟源重新由 OSCCLKSRCSEL 位决定。同时清除 MCDPCNT 和 MCDSCNT

计数器，使电路重新进行缺失时钟检测。如果用户希望在缺失时钟检测后重新锁定 PLL，应：先通过 OSCCLKSRCSEL 将时钟源切换到 INTOSC1，执行 MCLKCLR，然后重新锁定 PLL 。

8）MCD 在上电时默认启用。若 INTOSC2 在器件上电时失效，则不支持缺失时钟检测。

图中展示了缺失时钟检测逻辑的功能流程。

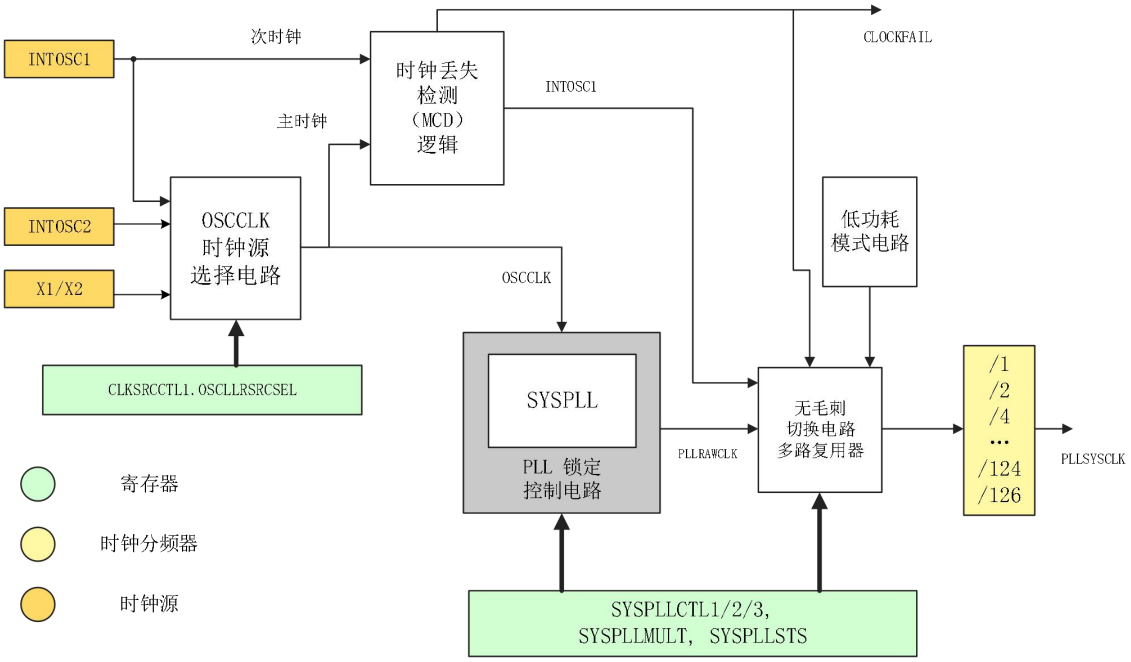



图 2-7 时钟检测逻辑的功能流程

注意：

当发生完全时钟故障（即 OSCCLK 停止）时，在 CLOCKFAIL 信号变为高电平之前，最长可能需要 8192 个 INTOSC1 时钟周期（即 0.8192 ms）。在此之后，将发生以下情况：

- 产生 NMI 中断
- OSCCLK 切换到 INTOSC1
- 触发 PWM Trip（保护事件）

## 2.8 定时器 0/1/2

本章节介绍 3 个 32 位 CPU 定时器 TIMER0/1/2，如图所示

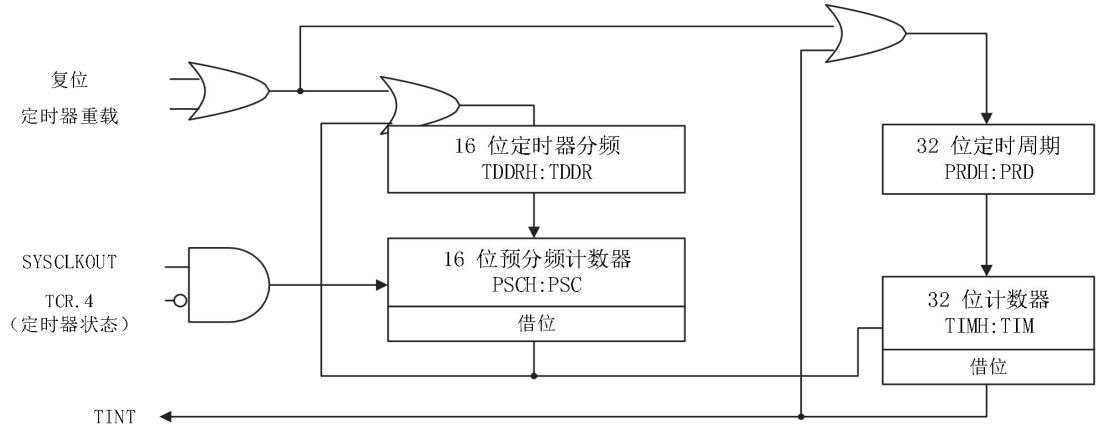



图 2-8 CPU 定时器

TIMER0 和 TIMER1 可以用于用户应用程序。TIMER2 预留给实时操作系统使用。如果应用程序没有使用此定时器的操作系统，则可以在应用程序中使用 TIMER2。

CPU 定时器的一般操作如下：

- 32 位计数器寄存器 TIMH:TIM，加载周期寄存器 PRDH:PRD
- 计数器每（TPR[TDDRH:TDDR]+1) SYSCLK 周期递减一次，其中 TDDRH:TDDR 是定时器的分频器。
- 当计数器达到 0 时，定时器中断输出信号产生中断脉冲。

## 2.9 看门狗定时器

看门狗模块由一个 8 位计数器组成，该计数器由一个经过预分频的时钟（WDCLK，连接到 INTOSC1）驱动。当计数器达到其最大值时，模块会产生一个宽度为 512 个 WDCLK 周期的输出脉冲。该脉冲可以用于产生中断或复位。

CPU 必须周期性地向看门狗密钥寄存器写入 0x55 + 0xAA 的序列，以复位看门狗计数器。该计数器也可以被禁用。

计数器时钟由 WDCLK 经过两个分频器分频得到：

预分频器（Prescaler） 可在 /1 到 /64 之间按 2 的幂进行调整；

前级分频器（Pre-divider） 默认为 /512（用于向后兼容），但也可以在 /2 到 /4096 之间按 2 的幂进行调整。

这种结构为安全关键应用提供了范围广泛的超时时间配置能力。

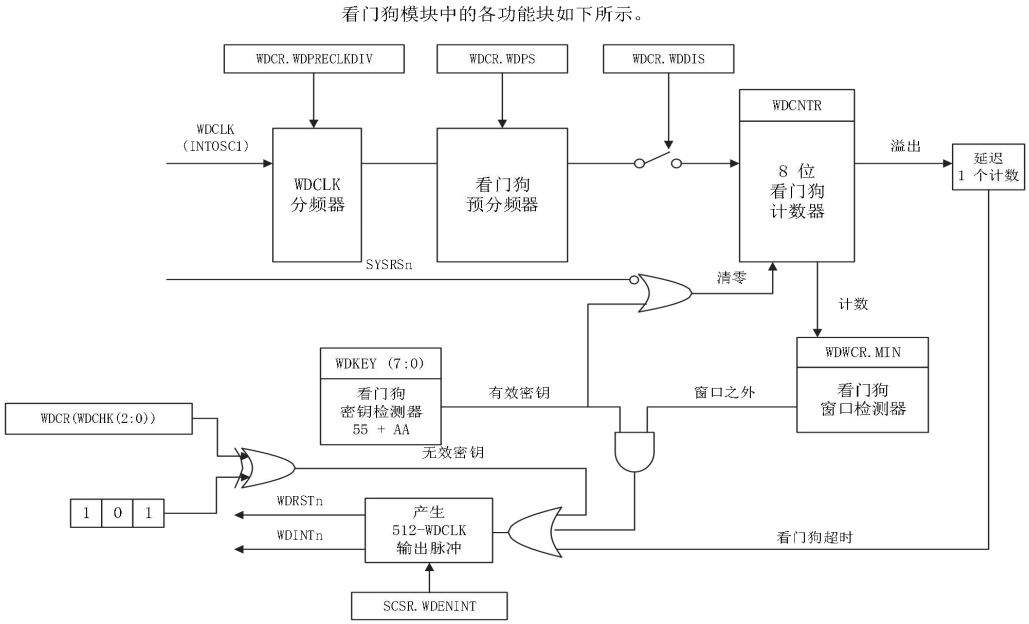



图 2-9 看门狗时钟

## 2.9.1 看门狗定时器的服务

当在 8 位看门狗计数器溢出之前，向 WDKEY 寄存器写入正确的序列时，看门狗计数器（WDCNTR）会被复位。

当向 WDKEY 写入数值 0x55 时，WDCNTR 被允许复位。

当随后向 WDKEY 写入 0xAA 时，WDCNTR 才会真正被复位。

向 WDKEY 写入除 0x55 或 0xAA 之外的任何数值都不会产生任何动作。

可以向 WDKEY 写入任意顺序的 0x55 和 0xAA，而不会导致系统复位；只有先写入 0x55，再写入 0xAA，才会使 WDCNTR 被复位。

使 WDCNTR 允许复位的第一次操作如表中的步骤 3 所示。

WDCNTR 实际上直到步骤 6 才被复位。

步骤 8 再次使 WDCNTR 允许复位，步骤 9 对 WDCNTR 进行复位。

步骤 10 再次使 WDCNTR 允许复位。

在步骤 11 中向 WDKEY 写入错误的 key 值不会产生任何动作，但此时 WDCNTR 已不再处于允许复位状态，因此步骤 12 中写入的 0xAA 将不会产生效果。

如果看门狗被配置为对器件执行复位，那么：

WDCR 发生溢出，或

向 WDCR[WDCHK] 位写入错误值

都会导致器件复位，并在复位原因寄存器（RESC）中置位看门狗标志位 WDRSn。

复位之后，程序可以读取该标志位状态，以判断此次复位是否由看门狗引起。完成判断后，程序可以清除 WDRSn，以便后续能够检测新的看门狗复位事件。

当该标志位被置位时，并不会阻止后续的看门狗复位发生。

表 2-5 看门狗关键流程

| 步骤 | 写入WDKEY的值 | 结果 |
|---|---|---|
| 1 | 0xAA | 无操作。 |
| 2 | 0xAA | 无操作。 |
| 3 | 0x55 | 允许复位 WDCNTR；若下一个写入值为 0xAA，则执行复位。 |
| 4 | 0x55 | 允许复位 WDCNTR；若下一个写入值为 0xAA，则执行复位。 |
| 5 | 0x55 | 允许复位 WDCNTR；若下一个写入值为 0xAA，则执行复位。 |
| 6 | 0xAA | WDCNTR 被复位。 |
| 7 | 0xAA | 无操作。 |
| 8 | 0x55 | 允许复位 WDCNTR；若下一个写入值为 0xAA，则执行复位。 |
| 9 | 0xAA | WDCNTR 被复位。 |
| 10 | 0x55 | 允许复位 WDCNTR；若下一个写入值为 0xAA，则执行复位。 |
| 11 | 0x32 | 写入 WDKEY 的值无效。不执行操作，且下一次写入 0xAA 将无法复位 WDCNTR。 |
| 12 | 0xAA | 由于前一次写入值无效，不执行操作。 |
| 13 | 0x55 | 允许复位 WDCNTR；若下一个写入值为 0xAA，则执行复位。 |
| 14 | 0xAA | WDCNTR 被复位。 |

### 2.9.2 最小窗口检查

为了补充超时机制，看门狗还包含一个可选的“窗口（windowing）”功能，该功能要求两次计数器复位之间必须满足一个最小时间间隔。这有助于防止某些错误情况——例如程序流程的大部分被绕过，但仍然执行了看门狗服务操作。

要设置窗口最小值，需要向 WDWCR 寄存器写入期望的最小看门狗计数值。该数值将在下一次 WDKEY 序列执行后生效。

此后，当 WDCNTR 小于 WDWCR 时，如果尝试对看门狗进行服务操作，将会触发看门狗中断或复位。

当 WDCNTR 大于或等于 WDWCR 时，看门狗可以被正常服务。

在复位时，窗口最小值默认为 0，这将禁用窗口功能。

### 2.9.3 看门狗复位模式或看门狗中断模式

当看门狗计数器达到最大值时，可以通过 SCSR 寄存器将看门狗配置为复位器件（WDRST）或产生中断（WDINT）。各模式行为如下：

- 复位模式（Reset mode）：

如果看门狗被配置为复位器件，当看门狗计数器达到最大值时，WDRST 信号会将器件复位引脚（XRS）拉低 512 个 INTOSC1 时钟周期。

- 中断模式（Interrupt mode）：

当看门狗计数器超时时，计数器通过将 WDINT 信号拉低 512 个 INTOSC1 时钟周期来产生中断。

WDINT 的下降沿在 CLIC 中触发 WAKEINT 中断（如果该中断已使能）。由于 CLIC 是边沿触发的，因此在 WDINT 仍然有效期间重新使能 WAKEINT，不会产生重复中断。为了避免不可预期的行为，在 WDINT 处于有效状态时，软件不得修改看门狗配置。

例如：

当 WDINT 有效时，如果将模式从中断模式切换为复位模式，器件会立即发生复位。

当 WDINT 有效时，如果禁用看门狗，则在之后重新启用看门狗时会产生重复中断。

如果在 WDINT 有效期间发生调试复位（debug reset），则复位原因寄存器（RESC）会显示为看门狗复位。可以通过读取 SCSR 寄存器中的 WDINTS 位来确定 WDINT 的当前状态。

### 2.9.4 低功耗模式下的看门狗运行

在 IDLE 模式下，看门狗中断（WDINT）信号可以产生一个 CPU 中断，使 CPU 退出 IDLE 模式。与其他外设相同，在 IDLE 模式期间，看门狗中断会在 CLIC 中触发一个 WAKE 中断。用户软件必须自行判断是哪一个外设引发了该中断。

注意：如果使用看门狗中断从 IDLE 低功耗模式中唤醒系统，软件必须确保在再次进入 IDLE 模式之前，WDINT 信号已经恢复为高电平。

当产生看门狗中断时，WDINT 信号会被保持为低电平 512 个 INTOSC1 时钟周期。

WDINT 的当前状态可以通过读取 SCSR 寄存器中的看门狗中断状态位（WDINTS）来确定。WDINTS 相对于 WDINT 信号状态存在 2 个 SYSCLKOUT 周期的延迟。

在 HALT 模式下，如果用户将 CLKSRCCTL1.WDHALTI = 1，则内部振荡器和看门狗定时器仍然保持工作状态。

在 HALT 模式中：看门狗复位可以唤醒系统；但看门狗中断不能唤醒系统。

### 2.9.5 仿真注意事项

在不同的调试条件下，看门狗模块的行为如下：

- CPU 挂起（CPU Suspended）

  当 CPU 被挂起时，看门狗时钟（WDCLK）也会被挂起。

- 自由运行模式（Run-Free Mode）

  当 CPU 被设置为自由运行模式时，看门狗模块恢复正常运行。

- 实时单步模式（Real-Time Single-Step Mode）

  当 CPU 处于实时单步模式时，看门狗时钟（WDCLK）被挂起。

  即使在实时中断期间，看门狗仍然保持挂起状态。

- 实时自由运行模式（Real-Time Run-Free Mode）

  当 CPU 处于实时自由运行模式时，看门狗按正常方式运行。

## 2.10 低功耗模式

该设备有 HALT, IDLE 和 STANDBY 作为时钟门控低功耗模式。所有的低功耗模式都是通过设置 LPMCR 寄存器和执行 WFI 指令进入的。当 Flash 编程或擦除操作正在进行时，不应进入低功耗模式。进入 HALT 将停止所有 CPU 和外设设备活动。这包括主动传输和控制算法。当准备进入 HALT 模式时，应用程序应确保系统准备进入一段不活动时间。

在进入 HALT 模式之前，检查为 HALT 唤醒（GPIOLPMSEL0/1）选择的引脚的 GPIODAT 寄存器的值，在进入低功耗模式之前，以确保唤醒事件未被声明。

### 2.10.1 时钟门控低功耗模式

下表描述了当任何时钟门控低功耗模式被输入时对系统的影响。

表 2-6 时钟门控低功率模式对设备的影响

| 模块/时钟域 | 空闲（IDLE） | 待机（STANDBY） | 停机（HALT） |
|---|---|---|---|
| SYSCLK | Active | Gated | Gated |
| CPUCLK | Gated | Gated | Gated |
| PERx.SYSCLK连接的模块时钟 | Active | Gated | Gated |

| 模块/时钟域 | 空闲（IDLE） | 待机（STANDBY） | 停机（HALT） |
|---|---|---|---|
| WDCLK | Active | Active | Gated if CLKSRCCTL1.WDHALTI = 0 |
| PLL | Powered | Powered | 软件必须在进入 HALT 之前关闭锁相环 |
| INTOSC1 | Powered | Powered | 若 CLKSRCCTL1.WDHALTI = 0 则关机 |
| INTOSC2 | Powered | Powered | 若 CLKSRCCTL1.WDHALTI = 0 则关机 |
| Flash （1） | Powered | Powered | Powered |
| XTAL （2） | Powered | Powered | Powered |

### 2.10.2 IDLE 模式

IDLE 是 CPU 的标准功能。在这种模式下，CPU 时钟处于门控状态，而所有外设时钟都处于运行状态。

1、进入 IDLE 模式，将 LPMCR.LPM 设置为 0x0;

2、执行 Fence 指令，确保对寄存器的访存操作完成;

3、再执行 WFI 指令;

4、低功耗模块会进入 IDLE 低功耗模式，只关断 CPU 时钟；

5、在任何启用的中断事件发生时，CPU 都会恢复正常运行。

### 2.10.3 STANDBY 模式

STANDBY 是一种更激进的低功耗模式，可同时关闭 CPU 时钟和 SYSCLK 的外设时钟。但看门狗仍处于激活状态。STANDBY 模式可由 GPIO 引脚事件或看门狗中断唤醒，用户在例程中需使能看门狗中断。

GPIO0-46 中的每个 GPIO 均可配置为低电平有效时唤醒 CPU。唤醒时，CPU 会收到 WAKEINT 中断。

1、进入 STANDBY 模式，设置 LPMCR.LPM 为 0x1；

2、对于看门狗中断唤醒，将 LPMCR.WDINTE 设置为 1 并配置看门狗以产生中断；

3、对于 GPIO 唤醒,设置 GPIOLPMSELO 和 GPIOLPMSEL1 以将所选 GPIO 连接到 LPM 模块，并设置 LPMCR.QUALSTDBY 以选择输入鉴定的 OSCCLK 周期数。

4、执行 Fence 指令，确保对寄存器的访存操作完成;

5、执行 WFI 指令进入 STANDBY 状态;

6、STANDBY 模式唤醒，配置所需的 GPIO 以触发唤醒。

7、将所选 GPIO 信号驱动为低电平:必须在 LPMCR 寄存器中 QUALSTDBY 位指定的 OSCCLK 周期数内保持低电平。如果在此期间信号采样为高电平，则计数重新开始。在鉴定期结束时，PLL 将启用 CPU 的 CLKIN，并锁存 WAKEINT 中断。此时，CPU 已脱离 STANDBY（待机）模式，可以恢复正常运行。

### 2.10.4 HALT 模式

HALT 是一种全局低功耗模式，可对几乎所有系统时钟进行门控，并允许对振荡器和模拟模块进行断电。进入 HALT 模式后，XTAL 不会自动断电。此外，如果 XTAL 未接通电源,从 HALT 模式唤醒时也不会自动接通 XTAL 的电源。我们增加了 XTALCR.OSCOFF 位，以便在不需要时通过应用软件打开或关闭 XTAL 电路。

对于要求在 HALT 模式期间功耗最小的应用，应用软件可以在进入 HALT 之前关闭 XTAL。如果 OSCCLK 源被配置为 XTAL，应用软件可在设置 XTALCR.OSCOFF 之前，先将 OSSCLK 源切换为 INTOSC1 或 INTOSC2。

GPIO0-63 可配置为从 HALT 唤醒系统。唤醒时，CPU 会收到一个 WAKEINT 中断。没有其他唤醒选项。不过，看门狗定时器仍可进行时钟设置，并可在需要超时机制时配置为产生看门狗复位系统。用户在例程中需使能看门狗复位。

1、进入 HALT 模式。

2、将 LPMCR.LPM 设置为 0x2。设置 GPIOLPMSELO 和 GPIOLPMSEL1,将所选 GPIO 连接到 LPM 模块。

3、将 CLKSRCCTL1.WDHALTI 设置为 1,以保持看门狗定时器处于激活状态,INTOSC1 和 INTOSC2 处于 HALT 上电状态。

4、将 CLKSRCCTL1.WDHALTI 设置为 0，禁用看门狗定时器，并在 HALT 状态下关闭 INTOSC1 和 INTOSC2 的电源。

5、执行 Fence 指令，确保对寄存器的访存操作完成;

6、执行 WFI 指令进入 HALT

7、进入 HALT 低功耗模式，默认情况下会关断除 XTAL 和 FLASH 电源外其余时钟，此时 FLASH、CPU、CLIC、DCSM、RAM、RVDM、外设等均关断;

8、从 HALT 模式唤醒,将选定的 GPIO 驱动为低电平至少 5us。这将激活 WAKEINT CLIC 中断。

9、再次将唤醒 GPIO 驱动为高电平，以启动 SYSPLL 的上电。

10、等待 16us 加上 1024 个 OSCLK 周期，以允许 PLLs 锁定并锁存 WAKEINT ISR。

11、执行 WAKEINTISR。设备现在已退出 HALT 模式，可以恢复正常执行。

注意:如果 WFI 指令在流水线中时收到中断或 NMI，系统将不再执行 WFI 指令。

从低功耗模式唤醒后，程序从原来位置继续执行，或执行中断服务函数的内容

### 2.10.5 Flash 断电

此设备的 Flash 模块在应用程序期间随时可进行断电。一些 Flash 断电需要考虑的情况:应确保将 Flash 断电使其进入睡眠模式时，对应的软件应用程序应在 RAM 中执行。并且注意当为了降低功耗而使 Flash 进入睡眠后，不能再访问 Flash。如果在睡眠后访问 Flash,将激活 Flash 的唤醒程序(激活时间取决于 PSLEEP 和 RWAIT)，则用户程序不能达到降低功耗的目的。例如，在 Flash 进入

睡眠状态后，且设备还未进入低功耗模式时，如果用户程序需要执行任何代码，都必须从 RAM 中执行，不能从 Flash 执行。

可根据给定的系统时钟频率调整 PSLEEP 和 RWAIT 从而减少唤醒 Flash 所需的时间。BootROM 为 150MHz 操作配置最佳的 PSLEEP 值，当 SYSCLK 低于 150MHz 时用户程序可通过降低 PSLEEP 来减少 Flash 唤醒时间。这适用于应用程序进入 Halt 模式的情况，因为在进入 Halt 模式之前必须禁用 PLL。在这种情况下，可以降低 PSLEEP 值以在退出 LPM 时更快地唤醒 Flash。

Flash 存储体本身支持低功耗模式，进入低功耗及允许时钟关断的流程:当经过激活宽限期后无 Flash 访问，Flash 模块会进入由回退功耗模式寄存器中设定的回退功耗模式;Flash 的功耗状态会在功耗状态寄存器中显示,CPU 查询该寄存器中当前功耗状态已处于低功耗状态，可以允许进行时钟关断时，将 Flash 时钟关闭。

## 2.11 存储器控制器模块

在该器件中，各个 RAM 具有不同的特性，具体如下：

- 专用于 CPU 的 RAM：M0、M1 和 LSx RAM
- 由 CPU、DMA 和 HIC 共享的 RAM：GSx RAM

所有这些 RAM 都具有高度可配置性，可用于控制来自不同主设备（master）的写访问和取指访问权限。

所有专用 RAM （M0、M1 和 LSx）都启用了 ECC 功能（同时覆盖数据）。

共享 RAM（GSx RAM）则启用了 奇偶校验（Parity）功能（同样覆盖数据）。

部分专用存储器同时也是安全存储器（secure memory）。

每个 RAM 都具有其各自独立的控制器，该控制器负责处理该 RAM 的访问保护 / 安全相关检查以及 ECC 或奇偶校验功能。

下图展示了这些 RAM 的配置情况。

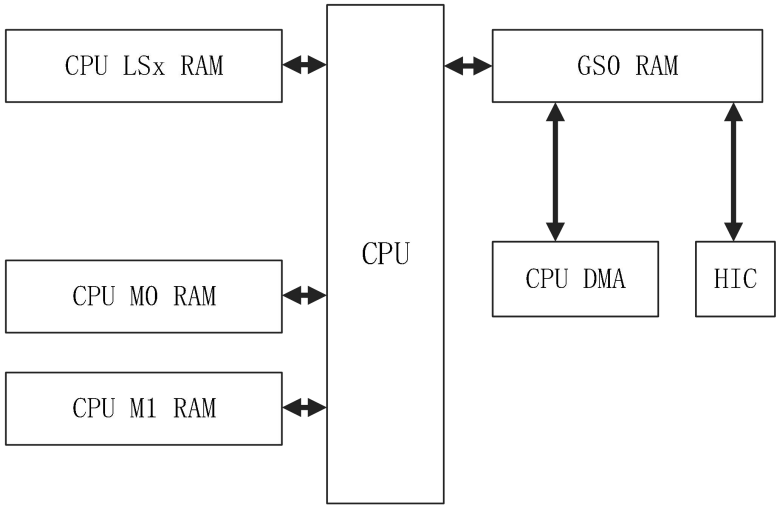



图 2-10 存储架构

### 2.11.1 功能描述

本节将进一步定义并讨论该器件中的专用 RAM 和共享 RAM。

#### 2.11.1.1 专用 RAM（Mx RAM）

该器件包含两个专用 RAM 块：M0 和 M1。

M0 和 M1 存储器是与 CPU 紧密耦合（tightly coupled）的小容量存储块。

只有 CPU 可以访问这些存储器，其他主设备（DMA 和 HIC）均无法访问。

所有专用 RAM 均支持 ECC 功能。

#### 2.11.1.2 本地共享 RAM（LSx RAM）

本地共享 RAM（LSx RAM）属于安全存储器（secure memory），并支持 ECC。

这些存储器专用于 CPU。

#### 2.11.1.3 全局共享 RAM（GSx RAM）

可被 CPU、DMA 和 HIC 访问的 RAM 块称为全局共享 RAM（GSx RAM）。

用户可以对 GSx RAM 块的访问保护配置进行锁定，以防止后续对这些位字段的修改。

用户还可以通过在 GSxCOMMIT 寄存器中设置特定位字段，将配置永久锁定（更多信息请参考寄存器说明）。

一旦某个 GSx RAM 块的配置被提交（commit），在执行 CPU.SYSRS（系统复位）之前，该配置将无法再被修改。

表 2-7 全局共享存储

| CPU（取指） | CPU（读） | CPU（写） | CPU.DMA（读） | CPU.DMA（写） | HIC（读） | HIC（写） |
|---|---|---|---|---|---|---|
| 是 | 是 | 是 | 是 | 是 | 是 | 是 |

#### 2.11.1.4 访问仲裁（Access Arbitration）

对于 GSxRAM，有 CPU（IFU/LSU）、HIC、DMA 参与轮询仲裁，默认优先级为 CPU>HIC>DMA，对于 GSx 的仲裁逻辑是，当寄存器未配置时，此时寄存器阈值为 0，所以三者的仲裁使能是一直拉高的，只要有 CPU/HIC/DMA 其中之一访问就会使得优先级开始轮询，这里只能保证优先级是轮询的，不能保证轮询完优先级的顺序一定是想要的最低的。

若此时设置阈值的话，三者的访问请求会增加其各自的计数器，若计数器等于阈值的话，优先级轮询一次，同时计数器清零，之后重新计数；当计数器值再次达到阈值时，再轮转一次优先级。

对于 DxRAM、LSxRAM、BootROM（其余寄存器 BGCRC_REG、DCSM_REG、MEM_CONTROL_REG、ERAD_REG、FLASH_NVR_CFG、RVDM），只有 CPU 区分 IFU 和 LSU 访问，这里与 GSxRAM 不同，只有当 IFU 和 LSU 发生冲突的时候才会参与仲裁，默认优先级为 IFU>LSU；

当寄存器未配置时，此时寄存器阈值为 0，IFU 和 LSU 同时发生请求冲突时，这时会产生仲裁使能，优先级开始轮询一次，若继续有请求冲突，则继续轮询。

若此时设置阈值的话，只有发生请求冲突并完成仲裁后，才递增相应主设备的计数器，若计数器等于阈值的话，优先级轮询一次，同时计数器清零，之后重新计数；当计数器值再次达到阈值时，再轮转一次优先级。

#### 2.11.1.5 存储器错误检测、纠正与错误处理

这些器件具备存储器错误检测与纠正功能，以满足安全标准的要求。这些安全要求需要增加检测机制，以应对有限的危险性故障。

在该器件中：所有专用 RAM 支持 ECC（错误纠正码）保护。共享 RAM 采用 奇偶校验（Parity）保护。所使用的 ECC 方案为 单比特纠错、双比特检测（SECDED）。所使用的奇偶校验方案为 偶校验（even parity）。ECC/Parity 同时覆盖：存储在存储器中的数据位。

ECC/Parity 的计算在存储器控制器模块内部完成，并与数据一起写入存储器。ECC/Parity 基于 32 位数据进行计算，因此对于每 32 位数据：共有 两个 7 位 ECC 码（或 2 位奇偶校验位），其中两个用于数据。

##### 2.11.1.5.1 错误检测与纠正（Error Detection and Correction）

错误检测在从存储器读取数据时执行。对于不同类型存储器：Parity 存储器：只能检测单比特错误。ECC 存储器：既可检测单比特错误，也可检测双比特错误。这些错误被分为可纠正错误和不可纠正错误，其特性如下：

- 奇偶校验错误始终属于不可纠正错误；
- 单比特 ECC 错误属于可纠正错误；
- 双比特 ECC 错误属于不可纠正错误；

对于可纠正错误：存储器控制器模块会自动进行纠正；然后将正确的数据返回给主设备作为读数据；同时将纠正后的数据重新写回存储器，以防止同一地址再次发生单比特错误而演变为双比特错误。

#### 2.11.1.5.2 错误处理（Error Handling）

对于每一次可纠正错误（correctable error），可纠正错误计数寄存器中的计数值会增加 1。当该计数寄存器中的数值达到可纠正错误阈值寄存器中配置的数值时，如果在可纠正错误中断使能寄存器中已使能中断，则会向 CPU 产生一个中断。用户需要根据系统需求配置可纠正错误阈值寄存器。同时：发生错误的地址会被锁存到相应寄存器中；状态寄存器中的对应标志位也会被置位。

如果发生不可纠正错误（uncorrectable error），则会向 CPU 产生 NMI（不可屏蔽中断）。在这种情况下：发生错误的地址同样会被锁存到寄存器中；状态寄存器中的标志位会被置位。

下表总结了可能出现的不同错误情况。软件需要利用提供的状态信息和中断指示，对这些错误进行适当处理。

表 2-8 错误处理方式

| 访问类型 | 错误来源 | 错误类型 | 状态指示 | 错误通知 |
|---|---|---|---|---|
| 读访问 | 从存储器读取的数据 | 不可纠正错误（Parity RAM 的单比特错误或 ECC RAM 的双比特错误） | 是 — CPU/CPU.DMA/CPU/DMA 读错误地址寄存器 | 返回给 CPU/CPU.DMA 的数据错误；CPU 访问产生 NMI；CPU.DMA 访问产生 NMI |
| 读访问 | 从存储器读取的数据 | ECC RAM 单比特错误 | 是 — CPU/CPU.DMA CPU/DMA 读错误地址寄存器 | 单比特错误计数器递增；当计数达到用户可编程阈值时产生中断 |

注意：在 CPU 执行取指（fetch）过程中，如果发生不可纠正错误，由于在 NMI 产生之前错误指令可能已经进入 CPU 流水线，因此可能会先触发 ITRAP，随后才产生 NMI 异常。

### 2.11.1.6 用于错误检测与纠正的应用测试钩子

由于错误检测与纠正逻辑属于安全关键逻辑的一部分，安全类应用可能需要确保该逻辑始终正常工作（包括运行期间）。为此，器件提供了一种测试模式（test

mode），在该模式下，用户可以修改数据位（而不修改 ECC/Parity 位），或直接修改 ECC/Parity 位。通过该功能，可以向数据中注入 ECC 或 Parity 错误。

注意：ECC/Parity 位与数据位使用相同的存储器映射地址。用户必须选择不同的测试模式来访问 ECC/Parity 位。

在测试模式下，对存储器（包括数据和 ECC/Parity）的所有访问能以 32 位访问方式进行，最高支持到 64 位。通过相应地址读取时 ECC/奇偶校验位的位映射。ECC 代码位映射显示了数据的低 32 位和高 32 位的 ECC 代码分别位于读取数据的 6:0 位和 38:32 位，其余位不使用。

下表展示了在 RAMTEST 模式下，通过对应地址读取时 ECC/Parity 位的位映射关系。

表 2-9 位映射关系

| 位域 | 内容（ECC 存储器） |
|---|---|
| 6:0 | ECC Code for lower 32bits of data |
| 31:7 | 保留 |
| 38:32 | ECC Code for higher 32bits of data |
| 63:39 | 保留 |

表 2-10 位映射关系

| 位域 | 内容（Parity存储器） |
|---|---|
| 0 | Parity for lower 32 bits of data |
| 31:1 | 保留 |
| 32 | Parity for upper 32 bits of data |
| 63:33 | 保留 |

### 2.11.1.7 RAM 初始化（RAM Initialization）

为了确保从未初始化的 RAM 地址进行读或取指操作时不会产生 ECC 或 Parity 错误，每个存储器块都提供了 RAM_INIT 功能。通过该功能，可以将任意 RAM 块初始化为：数据值 0x0，并同时生成对应的 ECC/Parity 位。初始化过程可通过在 INIT 寄存器中，将对应 RAM 块的 INIT 位设置为 1 来启动。软件可以通过轮询 INITDONE 寄存器中对应 RAM 块的 INITDONE 位来检查初始化状态。在该位被置位之前，不允许访问该 RAM 存储块。

注意：在初始化进行期间，任何主设备都无法访问该存储器。如果在 RAMINITDONE 置位之前访问存储器，则存储器读写操作以及初始化过程都无法正确完成。

## 2.12 JTAG

TAP_STATUS 寄存器可实时反映 JTAG TAP 的状态。正常情况下，器件未连接 JTAG 时，状态应为 IDLE（空闲）。在某些 PCB 噪声过大的场景中，会出现不必要的 TMS、TCK 信号跳变，使 JTAG 退出空闲状态。若该现象持续存在，最终可能导致 JTAG 边界扫描或其他 JTAG 模式被意外激活，从而干扰应用程序正常运行。为避免此类问题，需在板卡上放置足够强度的上拉 / 下拉电阻，防止噪声误触发 JTAG。

作为调试手段，应用程序可轮询 TAP_STATUS 寄存器，判断器件异常是否由此原因导致。也可通过 SOFTPRES40 [JTAG_nTRST] 寄存器 由软件复位 JTAG TAP。使用该复位寄存器时需谨慎：一旦操作，将无法连接调试器，除非代码中结合其他 GPIO 状态或其他判断机制，区分是噪声干扰还是调试器访问，再对该寄存器进行写操作。

## 2.13 双代码安全模块

双代码安全模块（DCSM）是该设备中内置的安全模块。DCSM 阻止未授权人员访问和查看片上安全内存（和其他安全资源）。DCSM 还防止专有代码的复制和逆向工程。术语“安全”意味着 DCSM 会阻止通过调试器，访存模块，系统总线访问片上安全内存和资源。术语“不安全”意味着允许访问（内存的内容可以通过任何手段读取）。

### 2.13.1 功能描述

DCSM 具有双重区域安全，分别为区域 1（zone1）和区域 2（zone2），用户可以根据需求配置片上的存储区域哪些属于 zone1，哪些属于 zone2。两个区域的

安全机制是相同的，每个区域都包括一个专用的安全资源和一些可供分配的安全资源，下面是此设备上几种不同的安全资源：

OTP：OTP（One Time Programed）是一块负责保存 DCSM 配置信息的存储区域。每个 zone 都有自己的专用安全 OTP（USER OTP），其中存储对应 zone 的安全配置信息。如果某个 zone 是安全的，则只有使用密码匹配流（PMF）解锁该 zone 时，才可以读取其 USER OTP 内容（包括 CSM 密码），该部分信息不可以被执行(不可被取指模块访问)。

RAM：所有 LSx RAM 都是可供分配给 zone1/zone2 的安全 RAM。通过在 Bank0 USER OTP 中配置相应的 GRABRAM 位置，可以将这些 RAM 分配给任一 zone 进行保护，见表中。

Flash Sectors 及对应的 ECC 地址区域： 片上 Flash 包括 16 个扇区（Sector），都是可供分配的安全资源。每个 sector 都可以通过配置所在 Bank0 的 USER OTP 中的 GRABSECT 来分配给任一 zone 进行保护，具体规则和配置 RAM 的安全区规则相同。每个扇区对应的 ECC 地址安全状态和扇区的安全状态保持一致。

表 2-11 RAM 安全区分配规则

| 输入寄存器配置信息 | | | 输出结果 | |
|---|---|---|---|---|
| Z1_GRABRAMx（1） | Z2_GRABRAMx | Zone状态 | RAMSTAT（2） | RAM所属关系和可访问性 |
| 01 | 10 | —— | 01 | RAM属于Zone1 |
| 01 | 11 | Zone2非安全 | 01 | RAM属于Zone1 |
| | | Zone2安全 | 00 | 不可访问（3） |
| 10 | 01 | —— | 10 | RAM属于Zone2 |
| 11 | 01 | Zone1非安全 | 10 | RAM属于Zone2 |
| | | Zone1安全 | 00 | 不可访问 |
| 10 | 10 | —— | 11 | 非安全 |
| 11 | 11 | Zone1非安全且Zone2非安全 | 11 | 非安全 |
| | | 其他 | 00 | 不可访问 |
| 00（复位后的状态，默认为不可访问状态） | | —— | 00 | 不可访问 |

注：（1）OTP 分为 Z1_OTP 和 Z2_OTP，上述 Z1_GRABRAMx 存于 Z1_OTP 里，Z2_GRABRAMx 存于 Z2_OTP 里，二者称为配置信息。

（2）RAMSTAT 是保存在 DCSM 内部的只读状态寄存器，用于记录每一块 RAM 的安全区分配信息，它是根据 Z1_OTP 和 Z2_OTP 的配置由硬件自动计算得到，RAMSTAT 值为 11 表示该存储不属于任何 zone 管理，即该区域是非安全的。01 表示某块存储属于 zone1 安全管理，10 表示某块存储属于 zone2 安全管理。00 表示该区域不属于任何一个安全区，但是处于不可被访存模块，调试器，系统总线访问的状态。

（3）当 Zx_GRABRAMx 配置为“11”属性时，需要对应安全区解锁后，才可进一步查看 RAM/Sector 的分配信息，否则 RAM/Sector 为不可访问状态。不可访问指的是访存模块和系统总线不可访问，取指模块的访问在任何时刻都是允许的。即这条指令可以被执行。如果该指令是访存指令，它可以访问非安全地址上的数据。

注意：不要在 OTP 的该区域烧录其他没有列举到的值，否则会导致该区域不可访问。

### 2.13.2 访问方式

安全区域 zone1/zone2 可以配置为加密状态和非加密状态，对于加密状态的 zone，称之为安全内存区域。安全区和非安全区都可以存放代码和数据，因此，程序对存储器的访问规则如下：

普通访存请求：普通访存包括调试器，系统总线，内核的访存模块的访存请求。普通访存对安全内存的访问总是被阻止的，除非程序的地址（PC）和被访问的地址属于同一个 zone。对非安全区的读写则总是被允许的。

Debug 模式访问：依据目标地址所属 Zone 的解锁情况以及 HW_DPC 的安全性判断（HW_DPC 不可被软件修改）。

取指单元访问：总是允许。

### 2.13.3 DCSM 密码

每个安全 zone 都有一个 128 位的密码。

如果某 zone 的所有 128 位密码都被编程为零(“全零密码”)，则无论 CSMKEYx 寄存器的内容如何，该 zone 都将变为永久安全状态（禁止 flash 烧录，禁止用户输入密码解锁，但该 zone 存储的程序可以被取指模块正常访问，该 zone 的数据仅可以被属于同一安全区的代码访问)。因此，用户编程过程中应避免使用“全零密码”，因为全零密码可防止调试安全代码和重新编程 Flash 扇区。通常情况下，用户可以选择在完全编程结束后使用全 0 密码对该 zone 进行永久保护。

CSMKEYx 寄存器是用户可访问的寄存器，用于解除区域安全。当 4 个 CSMKEY 寄存器中的值和 OTP 中的正确密码一致时，该安全区会被自动解锁。

注意：密码为全 F 时，对应 Zone 为解锁状态芯片。出厂时，未编程 OTP 的密码数据及其 ECC 位均保持全 F 状态。这种全 F 组合并不满足正常的 ECC 校验规则，可能会被误判为 ECC 错误。为保证用户在出厂后仍能够对 OTP 进行烧录，Flash 控制器会在识别到 PSW 和 ECC 均为全 F 时跳过 ECC 校验。当用户进行烧录时，即使密码数据本身仍为全 F，Flash 控制器也会正常进行 ECC 校验，因为 ECC 校验值已经由用户根据全 F 的写数据完成了计算，从而能触发安全初始化操作。

### 2.13.4 DCSM 安全初始化

在此设备上，GSx，MSx 内存不安全。为了避免在设备处于默认状态（重置后）时发生任何潜在的黑客攻击，所有可以被保护的安全存储区域（LSRAM，Flash USER OTP，Flash Sector）的所有类型的访问都将被禁用，直到完成正确的安全初始化。这意味着在复位后，除了 BOOT_ROM 和厂家 OTP 存储器之外，用户无法访问安全存储器资源，包括 LSRAM0~3，FLASH Sector0~15，Flash USER OTP，可以访问非安全资源 GSx，MSx。复位（任何类型的复位）后需要执行以下步骤，以初始化每个 CPU 子系统的安全性。

安全初始化未完成时，由 BOOTROM 软件按特定顺序访问 USEROTP，将每个区域对应的配置信息被硬件自动搬移至 DCSM 的控制寄存器中（该访问为假读取，访存模块发起对应地址的读请求，但是会得到结果 0，真实的数据被硬件直接发送

给 DCSM 寄存器）；安全初始化完成后，对 OTP 的访问请求根据正常的访存安全判断逻辑访问。具体的安全初始化过程如下（该程序应在 BOOTROM 上，每次复位后自动执行），该过程不可以被调试。

安全初始化读取的序列顺序如下：

- B0_Z1OTP_LINKPOINTER1
- B0_Z1OTP_LINKPOINTER2
- B0_Z1OTP_LINKPOINTER3
- B0_Z2OTP_LINKPOINTER1
- B0_Z2OTP_LINKPOINTER2
- B0_Z2OTP_LINKPOINTER3
- Z1OTP_PSWDLOCK
- Z1OTP_JTAGLOCK
- Z1OTP_BOOTPIN_CONFIG
- Z1OTP_GPREG2
- Z1OTP_BOOTDEF_LOW
- Z1OTP_BOOTDEF_HIGH
- Z2OTP_PSWDLOCK
- Z2OTP_JTAGLOCK
- Z2OTP_BOOTPIN_CONFIG
- Z2OTP_GPREG2
- Z2OTP_BOOTDEF_LOW
- Z2OTP_BOOTDEF_HIGH

读取 DCSM 寄存器 Z1_LINKPOINTER 计算安全区 1 的区选择块的起始地址

- Z1OTP_GRABRAM
- B0_Z1OTP_GRABSECT

- Z1OTP_CSMPSWD0
- Z1OTP_CSMPSWD1
- Z1OTP_CSMPSWD2
- Z1OTP_CSMPSWD3

读取 DCSM 寄存器 Z2_LINKPOINTER 计算安全区 2 的区选择块的起始地址

- Z2OTP_GRABRAM
- B0_Z2OTP_GRABSECT
- Z2OTP_CSMPSWD0
- Z2OTP_CSMPSWD1
- Z2OTP_CSMPSWD2
- Z2OTP_CSMPSWD3

BOOTROM 两种解锁方式，先使用 linkpointer 计算出默认密码尝试解锁，如果发现没有解锁（软件可读解锁状态），User OTP 被擦除为全 F，使用全 F 密码尝试解锁。

### 2.13.5 仿真代码安全逻辑（ECSL）

每个 Zone 使用低 64 位密码(CSMKEY (0/1) )实现了仿真代码安全逻辑（Emulation Code Secure Logic），以防止未经授权的用户通过调试访问安全代码。

若低 64 位密码(CSMKEY (0/1) 不正确（与 OTP 中保存的低 64 位密码对比），则当代码从非安全区运行至安全区时，JTAG 连接断开，仿真暂停，cpu 内部也停止运行，CPU 不受调试器控制；若低 64 位密码(CSMKEY (0/1) )正确，且此刻代码从非安全区运行至安全区时，则 JTAG 连接仍然暂时断开，仿真对用户暂停，但 cpu 仍在全速运行，直到跳出安全区后，自动恢复，此过程调试不可见。此机制允许系统在安全区域进行仿真，同时保持 DCSM 机制对安全内存的保护。JTAG 连接断开后，上位机无法调试。

### 2.13.6 密码锁定功能

通过在 USEROTP 中的 PSWDLOCK 位置将区域的 PSWDLOCK 字段烧录为 1111(0xF)以外的任何值，可以锁定每个 Zone USEROTP 中的密码位置。在锁定区域的密码之前，密码位置是非安全的，并且可以从来自调试器或 unsecure 内存中运行的代码读取。在软件开发阶段对闪存扇区进行编程时，用户可以使用此功能来避免意外锁定区域。在新设备上，所有 Zone OTP 中 PSWDLOCK 位置的值为 1111，代表所有 Zone 的密码均未锁定。

注意：密码解锁仅使密码位置为非安全状态，所有其他安全存储器和闪存扇区的其他位置根据安全设置保持安全。但是由于密码是不安全的，任何人都可以读取正确密码并使区域解锁。

代码安全性通常无需在项目开发阶段实施。但当为特定区域开发出稳健代码后，便需要进行安全防护。在将代码写入闪存前，必须设置 CSM 密码以保护该区域。完成 CSM 密码设置后，该区域即处于安全状态（通过在指定位置编程密码，并执行设备复位或设置 forcesec 位（Zx_CR.15）即可实现设备锁定）。此后，无论采用何种方式（如使用 JTAG 、运行外部/片上存储器代码等）调试安全内存内容均需输入有效密码。从安全内存中运行代码时（如典型终端用户使用场景）无需密码验证，但出于调试目的访问安全内存内容仍需密码授权。

### 2.13.7 JTAG 锁定功能

JTAG 锁定功能可用于永久禁用使用 JTAG 端口对设备的任何访问。例如，调试器或编程工具的访问被阻止。通过将 USEROTP 中的 Z1OTP_JTAGLOCK 位置中的值编程为 1111(0xF)以外的任何值，即可使能此功能。

### 2.13.8 链接指针和区选择块

每个安全区域均设有专用一次性可编程（OTP）存储块，用于存储与该区域安全配置相关的参数。以下为用户可编程配置项：

- B0_ZxOTP_LINKPOINTER1
- B0_ZxOTP_LINKPOINTER2
- B0_ZxOTP_LINKPOINTER3
- ZxOTP_BOOTPIN_CONFIG
- ZxOTP_GPREG2
- ZxOTP_BOOTDEF_LOW
- ZxOTP_PSWDLOCK
- ZxOTP_BOOTDEF_HIGH
- ZxOTP_GRABRAM
- B0_ZxOTP_GRABSECT
- ZxOTP_CSMPSWD0
- ZxOTP_CSMPSWD1
- ZxOTP_CSMPSWD2
- ZxOTP_CSMPSWD3

由于一次性密码本（OTP）不可擦除，因此以下配置均被置于两个存储库中各区域 OTP 闪存的区域选择块内。

- ZxOTP_GRABRAM
- B0_ZxOTP_GRABSECT
- ZxOTP_CSMPSWD0
- ZxOTP_CSMPSWD1
- ZxOTP_CSMPSWD2
- ZxOTP_CSMPSWD3

在 OTP 中，区域选择块的位置由三个 29 位链接指针（Bx-Zx-LINKPOINTERx）的数值决定，这些指针均被编程存储在各区域的 OTP 中。除链接指针位置外，所有 OTP 位置均采用 ECC 校验进行保护。由于链接指针位置未受 ECC 保护，系统提

供了三个需要编程相同数值的链接指针。当通过硬件对所有链接指针进行虚拟读取时，系统会通过位表决逻辑比较三个数值来确定最终链接指针值。由于 OTP 中 1 可被用户翻转为 0，但 0 不可被翻转为 1（OTP 不支持擦除操作），因此在解析后的链接指针中最高有效位为 0 时，即确定了区域选择块的有效基地址。在生成最终链接指针值时，若其位模式不符合表中所示配置，则最终链接指针值将变为 All_1（0xFFFF_FFFF），用于选择区域选择块 1（即默认区域选择块）。

表 2-12 链接指针和区选择块地址

| Zx_LINKPOINTER | 对应区选择块的地址偏移（相对于zonex_otp的基地址） |
|---|---|
| 32bxxx11111111111111111111111111111 1 | 0x40 |
| 32bxxx11111111111111111111111111111 0 | 0x60 |
| 32bxxx11111111111111111111111111111 0x | 0x80 |
| 32bxxx11111111111111111111111111111 0xx | 0xA0 |
| 32bxxx11111111111111111111111111111 0xxx | 0xC0 |
| 32bxxx11111111111111111111111111111 0xxxx | 0xE0 |
| 32bxxx11111111111111111111111111111 0xxxxx | 0x100 |
| 32bxxx11111111111111111111111111111 0xxxxxx | 0x120 |
| 32bxxx11111111111111111111111111111 0xxxxxxx | 0x140 |
| 32bxxx11111111111111111111111111111 0xxxxxxxx | 0x160 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxx | 0x180 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxx | 0x1A0 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxx | 0x1C0 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxx | 0x1E0 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxx | 0x200 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxx | 0x220 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxx | 0x240 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxx | 0x260 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxx | 0x280 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxxx | 0x2A0 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxxxx | 0x2C0 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxxxxx | 0x2E0 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxxxxxx | 0x300 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxxxxxxx | 0x320 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxxxxxxxx | 0x340 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxxxxxxxxx | 0x360 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxxxxxxxxxx | 0x380 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxxxxxxxxxxx | 0x3A0 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxxxxxxxxxxxx | 0x3C0 |
| 32bxxx11111111111111111111111111111 0xxxxxxxxxxxxxxxxxxxxxxxxxxxx | 0x3E0 |

表 2-13 区选择块块内地址

| 区选择块（ZSB） | |
|---|---|
| 地址偏移（相对于ZSB的基地址） | 32位存储内容 |

| 区选择块（ZSB） | |
|---|---|
| 0x0 | 保留 |
| 0x4 | 保留 |
| 0x8 | ZxOTP_GRABRAM |
| 0xC | B0_ZxOTP_GRABSECT |
| 0x10 | ZxOTP_CSMPSWD0 |
| 0x14 | ZxOTP_CSMPSWD1 |
| 0x18 | ZxOTP_CSMPSWD2 |
| 0x1C | ZxOTP_CSMPSWD3 |

表 2-14 基于链路指针的 Bank0 区选块位置确定方法

| 区域 1 OTP | | 区域 2 OTP | |
|---|---|---|---|
| 0x302000 | B0_Z1OTP_LINKPOINTER1 | 0x302400 | Bc1 |
| 0x302008 | B0_Z1OTP_LINKPOINTER2 | 0x302408 | B0_Z2OTP_LINKPOINTER2 |
| 0x302010 | B0_Z1OTP_LINKPOINTER3 | 0x302410 | B0_Z2OTP_LINKPOINTER3 |
| 0x302018 | Z1OTP_BOOTPIN_CONFIG | 0x302418 | Z2OTP_BOOTPIN_CONFIG |
| 0x30201C | Z1OTP_GPREG2 | 0x30241C | Z2OTP_GPREG2 |
| 0x302020 | Z1OTP_PSWDLOCK | 0x302420 | Z2OTP_PSWDLOCK |
| 0x302030 | Z1OTP_JTAGLOCK | 0x302430 | Z2OTP_JTAGLOCK |
| 0X302038 | Z1OTP_BOOTDEF_LOW | 0X302438 | Z2OTP_BOOTDEF_LOW |
| 0x30203C | Z1OTP_BOOTDEF_HIGH | 0x30243C | Z2OTP_BOOTDEF_HIGH |
| 0x302040 | Zone1区选择块1 | 0x302440 | Zone2区选择块1 |
| 0x302060 | Zone1区选择块2 | 0x302460 | Zone2区选择块2 |
| ... | ... | ... | ... |
| 0x3023E0 | zone1区选择块30 | 0x3027E0 | zone2区选择块30 |

注意 ：上表中未提到的地址，为保留地址，不要向其烧录任何值。OTP 除 Linkpointer 外，皆采用 ECC 保护，烧录对应值时应正确烧录 ECC 值，烧录错误的 ECC 值会导致芯片锁死。每 64 位数据与 8 位 ECC 校验位对应，烧录时，同一个 64 位应一次性烧录，多次烧录可能导致 ECC 从 0 到 1 的翻转从而无法完成烧录。

### 2.13.9 Flash 烧录保护

烧录保护目的是为了防止未经授权的指令对 Flash 内部值进行随意修改。当 Flash 接收到来自 CPU 的烧录指令时，Flash 会将此烧录指令的 PC 值和目标 地址值给 DCSM 进行权限判断，若权限允许则开始烧录过程。权限允许的烧录场景如下：

指令 PC 与指令目标烧录地址属于同一 Zone。

指令 PC 与指令目标烧录地址不属于同一 Zone，且目标烧录地址位于非安全区或解锁的安全区。

FLASH 模块内置快速擦除功能，当且仅当 FLASH 的所有扇区都为非安全状态时，可以快速对 FLASH 的所有的扇区进行全部擦除。该烧录的地址为一个不属于 FLASH 可访问的虚拟地址（0x80100000），用以标记该次擦除操作为 FLASH 快速擦除。

### 2.13.10 Trace 安全

DCSM 接收来自 TRACE 模块提供的 PC，DCSM 判断其 PC 值是否位于安全区。PC 判断结果若是属于非安全区，则 TRACE 会记录该 PC 至暂存区；判断结果若是属于安全区，则用特殊值替换真实的 PC 值。

### 2.13.11 Trigger 安全

当 TRIGGER 模块使能时，DCSM 接收来自 TRIGGER 模块提供的 PC 值，若该 PC 值位于安全区，则不触发该 TRIGGER。PC 判断结果若是属于非安全区，则正常触发 TRIGGER。

## 2.14 系统控制寄存器配置限制

系统控制模块中的存储器映射寄存器工作在 INTOSC1 时钟域。因此，CPU 对这些寄存器进行连续写操作时，两次写入之间必须有一定的延迟，否则第二次写入可能会丢失。应用程序需要考虑这一要求，并在每次写入表中所列寄存器后，通过执行一定数量的 NOP（空操作）指令来实现延迟。连续写操作之间的延迟计算公式如下：

$$\textbf{延迟（以 SYSCLK 周期数计）} = 3 \times (F_{SYSCLK} \div F_{INTOSC1}) + 9 \quad (2\text{-}1)$$

例如，当 SYSCLK = 100 MHz 时：

$$\textbf{延迟（以 SYSCLK 周期数计）} = 3 \times (100\,MHz \div 10\,MHz) + 9 = 39 \text{ 个 SYSCLK 周期} \quad (2\text{-}2)$$

表 2-15 系统控制寄存器配置限制数据表

| 每次写入后需要延迟的寄存器 |
| --- |
| CLBCLKCTL |
| PERCLKDIVSEL |
| SYSCLKDIVSEL |
| SYSPLLCTL1 |
| SYSPLLMULT |
| WDCR |
| XCLKOUTDIVSEL |
| XTALCR |
| CLKSRCCTL1 |
| CLKSRCCTL2 |
| CLKSRCCTL3 |
| CPU1TMR2CTL (TMR2CLKCTL) |

# 2.15 寄存器描述

## 2.15.1 SYSCTRL 基地址表

表 2-16 SYSCTRL 基地址表

| 位域名 | | DriverLib 名称 | 基地址 | DMA | HIC | 流水线 |
| --- | --- | --- | --- | --- | --- | --- |
| 实例 | 结构 | | | | | |
| CpuTimer0Regs | CPUTIMER_REGS | CPUTIMER0_BASE | 0x0060_1000 | - | - | - |
| CpuTimer1Regs | CPUTIMER_REGS | CPUTIMER1_BASE | 0x0060_1010 | - | - | - |
| CpuTimer2Regs | CPUTIMER_REGS | CPUTIMER2_BASE | 0x0060_1020 | - | - | - |
| WdRegs | WD_REGS | WD_BASE | 0x0069_2000 | - | YES | YES |
| NmiIntruptRegs | NMI_INTRUPT_REGS | NMI_BASE | 0x0069_2400 | - | YES | YES |
| XintRegs | XINT_REGS | XINT_BASE | 0x0069_2200 | - | YES | YES |
| SyncSocRegs | SYNC_SOC_REGS | SYNCSOC_BASE | 0x0064_0080 | - | - | YES |
| DmaClaSrcSelRegs | DMA_CLA_SRC_SEL_REGS | DMACLASRCSEL_BASE | 0x0064_0100 | - | - | YES |
| DevCfgRegs | DEV_CFG_REGS | DEVCFG_BASE | 0x0065_0000 | - | - | YES |
| ClkCfgRegs | CLK_CFG_REGS | CLKCFG_BASE | 0x0065_0400 | - | - | YES |
| CpuSysRegs | CPU_SYS_REGS | CPUSYS_BASE | 0x0065_0600 | - | - | YES |
| DcsmBank0Z1Regs | DCSM_Bank0_Z1_REGS | DCSMBank0_Z1_BASE | 0x0000_5800 | - | - | YES |
| DcsmBank0Z2Regs | DCSM_Bank0_Z2_REGS | DCSMBank0_Z2_BASE | 0x0000_5880 | - | - | YES |

| 位域名 | | DriverLib 名称 | 基地址 | DMA | HIC | 流水线 |
|---|---|---|---|---|---|---|
| 实例 | 结构 | | | | | |
| DcsmCommonRegs | DCSM_COMMON_REGS | DCSMCOMMON_BASE | 0x0000_58e0 | - | - | YES |
| MemCfgRegs | MEM_CFG_REGS | MEMCFG_BASE | 0x0000_6000 | - | - | YES |
| AccessProtectionRegs | ACCESS_PROTECTION_REGS | ACCESSPROTECTION_BASE | 0x0000_6180 | - | - | YES |
| MemoryErrorRegs | MEMORY_ERROR_REGS | MEMORYERROR_BASE | 0x0000_6200 | - | - | YES |
| TestErrorRegs | TEST_ERROR_REGS | TESTERROR_BASE | 0x0000_6300 | - | - | YES |
| BusPollArbitrationRegs | BUS_POLL_ARB_REGS | - | 0x0000_6400 | - | - | - |
| DcsmBank0Z1OtpRegs | DCSM_Bank0_Z1_OTP | DCSMBank0_Z1OTP_BASE | 0x0030_2000 | - | - | - |
| DcsmBank0Z2OtpRegs | DCSM_Bank0_Z2_OTP | DCSMBank0_Z2OTP_BASE | 0x0030_2400 | - | - | - |

## 2.15.2 CPUTIMER_REGS 寄存器

表列出了 CPU 定时器寄存器（CPUTIMER_REGS）的内存映射寄存器。表中未列出的所有寄存器偏移地址均应视为保留地址，且其寄存器内容不得修改。

表 2-17 CPUTIMER_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | TIM | CPU-Timer, 计数器寄存器 |
| 4h | PRD | CPU-Timer, 周期寄存器 |
| 8h | TCR | CPU-Timer, 控制寄存器 |
| Ch | TPR | CPU-Timer, 预分频寄存器 |
| Eh | TPRH | CPU-Timer, 预分频寄存器高位 |

本文档对复杂的位访问类型进行了编码，以适应小的单元格。

表 2-18 用于访问类型的代码

| 访问类型 | 代码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| 写类型 | | |
| W | W | 写 |
| W1C | W<br>1C | 写<br>1设置 |
| 复位或默认值 | | |

| 访问类型 | 代码 | 说明 |
| --- | --- | --- |
| - n | | 复位后的值或默认值 |

### 2.15.2.1 TIM 寄存器

偏移地址：0h　复位值：0000FFFFh

表 2-19 TIM 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MSW | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MSW | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LSW | | | | | | | |
| R/W-FFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LSW | | | | | | | |
| R/W-FFFFh | | | | | | | |

表 2-20 TIM 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 31-16 | MSW | R/W | 0h | CPU 定时器计数寄存器 TIMH 寄存器保存当前 32 位定时器计数值的高 16 位。TIMH:TIM 每 (TDDRH:TDDR+1) 个时钟周期减一，其中 TDDRH:TDDR 是定时器预分频分频值。当 TIMH:TIM 减到零时，TIMH:TIM 寄存器会重新加载 PRDH:PRD 寄存器中的周期值。此时产生定时器中断 (TINT) 信号。复位类型：SYSRSn |
| 15-0 | LSW | R/W | FFFFh | CPU 定时器计数寄存器 TIM 寄存器保存当前 32 位定时器计数值的低 16 位。TIMH:TIM 每 (TDDRH:TDDR+1) 个时钟周期减一，其中 TDDRH:TDDR 是定时器预分频分频值。当 TIMH:TIM 减到零时，TIMH:TIM 寄存器会重新加载 PRDH:PRD 寄存器中的周期值。此时产生定时器中断 (TINT) 信号。复位类型：SYSRSn |

### 2.15.2.2 PRD 寄存器

偏移地址：4h　复位值：0000FFFFh

表 2-21 PRD 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MSW | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MSW | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LSW | | | | | | | |
| R/W-FFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LSW | | | | | | | |
| R/W-FFFFh | | | | | | | |

表 2-22 PRD 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | MSW | R/W | 0h | CPU 定时器周期寄存器 PRDH 寄存器保存 32 位周期值的高 16 位。当 TIMH:TIM 减到零时，在下一个定时器输入时钟周期（预分频器输出）开始时，TIMH:TIM 寄存器会重新加载 PRDH:PRD 寄存器中的周期值。当您在定时器控制寄存器 (TCR) 中设置定时器重载位 (TRB) 时，PRDH:PRD 的内容也会加载到 TIMH:TIM 中。复位类型：SYSRSn |
| 15-0 | LSW | R/W | FFFFh | CPU 定时器周期寄存器 PRD 寄存器保存 32 位周期值的低 16 位。当 TIMH:TIM 减到零时，在下一个定时器输入时钟周期（预分频器输出）开始时，TIMH:TIM 寄存器会重新加载 PRDH:PRD 寄存器中的周期值。当您在定时器控制寄存器 (TCR) 中设置定时器重载位 (TRB) 时，PRDH:PRD 的内容也会加载到 TIMH:TIM 中。复位类型：SYSRSn |

### 2.15.2.3 TCR 寄存器

偏移地址：8h　复位值：0001h

表 2-23 TCR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| TIF | TIE | 保留 | | FREE | SOFT | 保留 | |
| R/W1C-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | TRB | TSS | 保留 | | | |
| R-0h | | R/W-0h | R/W-0h | R-1h | | | |

表 2-24 TCR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15 | TIF | R/W1C | 0h | CPU 定时器溢出标志。TIF 指示自上次清除 TIF 以来是否发生了定时器溢出。TIF 不会自动清除，也不需要清除即可使能下一次定时器中断。<br>复位类型：SYSRSn<br>0h (R/W) = CPU 定时器尚未减到零。写入 0 被忽略。<br>1h (R/W) = 当 CPU 定时器减到零时此标志置位。向此位写 1 可清除该标志。 |
| 14 | TIE | R/W | 0h | CPU 定时器中断使能。<br>复位类型：SYSRSn<br>0h (R/W) = CPU 定时器中断禁用。<br>1h (R/W) = CPU 定时器中断使能。如果定时器减到零且 TIE 置位，则定时器会断言其中断请求。 |
| 13-12 | RESERVED | R | 0h | 保留 |
| 11 | FREE | R/W | 0h | 如果 FREE 位置 1，则在软件断点发生时定时器继续运行。如果 FREE 为 0，则 SOFT 位控制仿真行为。<br>复位类型：SYSRSn<br>0h (R/W) = 在 TIMH:TIM 下一次减一后停止（硬停止）（SOFT 位控制仿真行为）<br>1h (R/W) = 自由运行（SOFT 位无关，计数器自由运行） |
| 10 | SOFT | R/W | 0h | 如果 FREE 位置 1，则在软件断点发生时定时器继续运行（即自由运行）。此时 SOFT 位无关。但如果 FREE 为 0，则 SOFT 位生效。<br>复位类型：SYSRSn<br>0h (R/W) = 在 TIMH:TIM 下一次减一后停止（硬停止）。（仅在 FREE=0 时有效，如果 FREE=1 则此位无关）<br>1h (R/W) = 在 TIMH:TIM 减到 0 后停止（软停止）。在软停止模式下，定时器在关闭前会产生中断（因为达到 0 是触发中断的条件）。（仅在 FREE=0 时有效，如果 FREE=1 则此位无关） |
| 9-6 | RESERVED | R | 0h | 保留 |
| 5 | TRB | R/W | 0h | 定时器重载。<br>复位类型：SYSRSn<br>0h (R/W) = TRB 位始终读为零。写入 0 被忽略。<br>1h (R/W) = 当向 TRB 写 1 时，TIMH:TIM 加载 PRDH:PRD 中的值，预分频计数器 (PSCH:PSC) 加载定时器分频寄存器 (TDDRH:TDDR) 中的值。 |
| 4 | TSS | R/W | 0h | CPU 定时器停止状态位。TSS 是停止或启动 CPU 定时器的 1 位标志。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 0h (R/W) = 读为 0 表示 CPU 定时器正在运行。要启动或重新启动 CPU 定时器，将 TSS 设为 0。复位时 TSS 清零为 0，CPU 定时器立即启动。<br>1h (R/W) = 读为 1 表示 CPU 定时器已停止。要停止 CPU 定时器，将 TSS 设为 1。 |
| 3-0 | RESERVED | R | 1h | 保留 |

### 2.15.2.4 TPR 寄存器

偏移地址：Ch　复位值：0000h

表 2-25 TPR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| PSC | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TDDR | | | | | | | |
| R/W-0h | | | | | | | |

表 2-26 TPR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-8 | PSC | R | 0h | CPU 定时器预分频计数器。这些位保存定时器当前的预分频计数值。每当定时器时钟源周期中 PSCH:PSC 值大于 0 时，PSCH:PSC 减一。在 PSCH:PSC 达到 0 后的一个定时器时钟（定时器预分频器输出）周期，PSCH:PSC 会加载 TDDRH:TDDR 的内容，定时器计数寄存器 (TIMH:TIM) 减一。每当软件设置定时器重载位 (TRB) 时，PSCH:PSC 也会重新加载。可以通过读取寄存器检查 PSCH:PSC，但不能直接设置。它必须从定时器分频寄存器 (TDDRH:TDDR) 获取其值。复位时 PSCH:PSC 设为 0。复位类型：SYSRSn |
| 7-0 | TDDR | R/W | 0h | CPU 定时器分频。每 (TDDRH:TDDR + 1) 个定时器时钟源周期，定时器计数寄存器 (TIMH:TIM) 减一。复位时 TDDRH:TDDR 位清零为 0。要将整个定时器计数增加整数倍，请将该倍数减一写入 TDDRH:TDDR 位。当预分频计数器 (PSCH:PSC) 值为 0 时，一个定时器时钟源周期后，TDDRH:TDDR 的内容会重新加载 PSCH:PSC，TIMH:TIM 减一。每当软件设置定时器重载位 (TRB) 时，TDDRH:TDDR 也会重新加载 PSCH:PSC。复位类型：SYSRSn |

### 2.15.2.5 TPRH 寄存器

偏移地址：Eh　复位值：0000h

表 2-27 TPRH 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| PSCH | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TDDRH | | | | | | | |
| R/W-0h | | | | | | | |

表 2-28 TPRH 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-8 | PSCH | R | 0h | 参见 TIMERxTPR 描述。复位类型：SYSRSn |
| 7-0 | TDDRH | R/W | 0h | 参见 TIMERxTPR 描述。复位类型：SYSRSn |

## 2.15.3 WD_REGS 寄存器

表中列出了 WD_REGS 寄存器的存储器映射寄存器。所有未在表中列出的寄存器偏移地址应视为保留位置，不应修改寄存器内容。

表 2-29 WD_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 44h | SCSR | 系统控制和状态寄存器 |
| 46h | WDCNTR | 看门狗计数寄存器 |
| 4Ah | WDKEY | 看门狗复位密钥寄存器 |
| 52h | WDCR | 看门狗控制寄存器 |
| 54h | WDWCR | 看门狗窗口控制寄存器 |

复杂位访问类型经过编码以适应较小的表格单元格。表中显示了本节中使用的访问类型代码。

表 2-30 WD_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | 读取类型 | 读取类型 |
| R | R | 读取 |
| R-0 | R-0 | 读取返回 0 |
| 写入类型 | 写入类型 | 写入类型 |
| W | W | 写入 |
| W1C | W1C | 写 1 清零 |
| W1S | W1S | 写 1 置位 |
| 复位或默认值 | 复位或默认值 | 复位或默认值 |

| 访问类型 | 代码 | 描述 |
|---|---|---|
| - n | | 复位后或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | 寄存器数组变量 |
| i,j,k,l,m,n | | 当这些变量出现在寄存器名称、偏移或地址中时，它们表示寄存器数组的值，其中寄存器是重复寄存器组的一部分。寄存器组形成层次结构，数组以公式表示。 |
| y | | 当此变量出现在寄存器名称、偏移或地址中时，它表示寄存器数组的值。 |

### 2.15.3.1 SCSR 寄存器

偏移地址：44h　复位值：5h

系统控制和状态寄存器

建议仅使用 16 位访问来写此寄存器。使用读-修改-写指令可能会无意中清除其他位。

表 2-31 SCSR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | WDINTS | WDENINT | WDOVERRIDE |
| R-0-0h | | | | | R-1h | R/W-0h | R/W1C-1h |

表 2-32 SCSR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-3 | RESERVED | R-0 | 0h | 保留 |
| 2 | WDINTS | R | 1h | 看门狗中断状态。此位指示低电平有效的看门狗中断信号的状态（与 SYSCLK 同步）。如果看门狗中断用于将系统从低功耗模式唤醒，则只有在此位为高时才应进入该模式。同样，在看门狗可以安全地禁用和重新使能之前，此位必须先变为高。<br>复位类型：SYSRSn<br>0h (R/W) = 看门狗中断信号有效。<br>1h (R/W) = 看门狗中断信号无效。 |
| 1 | WDENINT | R/W | 0h | 看门狗中断使能/复位禁用。此位决定计数器到期时看门狗触发中断 (WAKE/WDOG) 还是复位 (WDRS)。<br>复位类型：SYSRSn<br>0h (R/W) = 计数器到期触发复位。这是上电和任何系统复位后的默认状态。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 1h (R/W) = 计数器到期触发中断。 |
| 0 | WDOVERRIDE | R/W1C | 1h | 如果此位设为 1，则允许用户更改看门狗控制 (WDCR) 寄存器中看门狗禁用 (WDDIS) 位的状态。如果 WDOVERRIDE 位通过写 1 被清除，则用户无法修改 WDDIS 位。写 0 无效。如果此位被清除，它将保持此状态直到发生复位。用户可以读取此位的当前状态。复位类型：SYSRSn |

### 2.15.3.2 WDCNTR 寄存器

偏移地址：46h　复位值：0h

看门狗计数寄存器

表 2-33 WDCNTR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WDCNTR | | | | | | | |
| R-0h | | | | | | | |

表 2-34 WDCNTR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |
| 7-0 | WDCNTR | R | 0h | 看门狗计数器。这些位包含看门狗计数器的当前值。此计数器随每个 WDCLK (INTOSC1) 周期递增。如果计数器溢出，则根据 SCSR 寄存器中 WDINTEN 位的值产生中断或复位。如果向 WDKEY 寄存器写入正确的值，此计数器会复位为零。复位类型：IORSn |

### 2.15.3.3 WDKEY 寄存器

偏移地址：4Ah　复位值：0h

看门狗复位密钥寄存器

表 2-35 WDKEY 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WDKEY | | | | | | | |
| R/W-0h | | | | | | | |

表 2-36 WDKEY 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |
| 7-0 | WDKEY | R/W | 0h | 看门狗计数器复位。写入 0x55 后接着写入 0xAA 会使看门狗计数器复位为零，防止溢出。写入其他值无效。读取此寄存器返回 WDCR 寄存器的值。<br>复位类型：IORSn |

### 2.15.3.4 WDCR 寄存器

偏移地址：52h　复位值：0h

看门狗控制寄存器

此存储器映射寄存器要求后续写入之间有延迟，否则第二次写入可能丢失。对于 200 MHz 器件，所需延迟为 69 个 SYSCLK 周期；对于 120 MHz 器件为 45 个 SYSCLK 周期；对于 100 MHz 器件为 39 个 SYSCLK 周期。此延迟可以通过添加与所需延迟周期数对应的 NOP 指令来实现。

表 2-37 WDCR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | WDPRECLKDIV | | | |
| R-0-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WDFLG | WDDIS | WDCHK | | | WDPS | | |
| R/W1S-0h | R/W-0h | R-0/W-0h | | | R/W-0h | | |

表 2-38 WDCR 寄存器字段描述

| 位 | 字 段 | 类 型 | 复位 | 描　　述 |
|---|---|---|---|---|
| 15-12 | RESERVED | R-0 | 0h | 保留 |
| 11-8 | WDPRECLKDIV | R/W | 0h | 看门狗时钟预分频器。这些位决定看门狗时钟预分频器，它是 INTOSC1 和看门狗计数器时钟 (WDCLK) 之间的两个分频器中的第一个。WDCLK 的频率由以下公式给出：PREDIVCLK = INTOSC1 / |

| 位 | 字 段 | 类 型 | 复位 | 描　　述 |
|---|---|---|---|---|
| | | | | 预分频器 WDCLK = PREDIVCLK / 预分频器 复位类型：IORSn<br>0h (R/W) = PREDIVCLK = INTOSC1 / 512<br>1h (R/W) = PREDIVCLK = INTOSC1 / 1024<br>2h (R/W) = PREDIVCLK = INTOSC1 / 2048<br>3h (R/W) = PREDIVCLK = INTOSC1 / 4096<br>4h (R/W) = 保留<br>5h (R/W) = 保留<br>6h (R/W) = 保留<br>7h (R/W) = 保留<br>8h (R/W) = PREDIVCLK = INTOSC1 / 2<br>9h (R/W) = PREDIVCLK = INTOSC1 / 4<br>Ah (R/W) = PREDIVCLK = INTOSC1 / 8<br>Bh (R/W) = PREDIVCLK = INTOSC1 / 16<br>Ch (R/W) = PREDIVCLK = INTOSC1 / 32<br>Dh (R/W) = PREDIVCLK = INTOSC1 / 64<br>Eh (R/W) = PREDIVCLK = INTOSC1 / 128<br>Fh (R/W) = PREDIVCLK = INTOSC1 / 256 |
| 7 | WDFLG | R/W1S | 0h | 看门狗复位状态标志位。如果此位被置位，则指示看门狗复位 (WDRSTn) 产生了复位条件。如果为 0，则是外部器件或上电复位条件。此位保持锁存，直到用户写 1 清除该条件。写 0 被忽略。复位类型：IORSn |
| 6 | WDDIS | R/W | 0h | 看门狗禁用。将此位置位可禁用看门狗模块。将此位清零可使能看门狗模块。此位可被 SCSR 寄存器中的 WDOVERRIDE 位锁定。复位时看门狗使能。复位类型：IORSn |
| 5-3 | WDCHK | R-0/W | 0h | 看门狗检查位。在对本寄存器的任何写入期间，这些位必须写入值 101（二进制）。写入任何其他值都会立即触发看门狗复位或中断。复位类型：IORSn |
| 2-0 | WDPS | R/W | 0h | 看门狗时钟预分频器。这些位决定看门狗时钟预分频器，它是 INTOSC1 和看门狗计数器时钟 (WDCLK) 之间的两个分频器中的第二个。WDCLK 的频率由以下公式给出：PREDIVCLK = INTOSC1 / 预分频器 WDCLK = PREDIVCLK / 预分频器 看门狗复位或中断脉冲宽度为 512 个 INTOSC1 周期，因此计数器周期必须更长。为保证这一点，预分频器和预分频器的乘积必须大于或等于四。默认预分频器值为 1。<br>复位类型：IORSn<br>0h (R/W) = WDCLK = PREDIVCLK / 1<br>1h (R/W) = WDCLK = PREDIVCLK / 1<br>2h (R/W) = WDCLK = PREDIVCLK / 2<br>3h (R/W) = WDCLK = PREDIVCLK / 4<br>4h (R/W) = WDCLK = PREDIVCLK / 8<br>5h (R/W) = WDCLK = PREDIVCLK / 16<br>6h (R/W) = WDCLK = PREDIVCLK / 32<br>7h (R/W) = WDCLK = PREDIVCLK / 64 |

### 2.15.3.5 WDWCR 寄存器

偏移地址：54h　复位值：0h

看门狗窗口控制寄存器

表 2-39 WDWCR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | FIRSTKEY |
| R-0-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MIN | | | | | | | |
| R/W-0h | | | | | | | |

表 2-40 WDWCR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-9 | RESERVED | R-0 | 0h | 保留 |
| 8 | FIRSTKEY | R | 0h | 此位指示在 MIN 配置为非零值后是否检测到第 1 个有效 WDKEY (0x55 + 0xAA)。<br>0：非零 MIN 配置后的第一个有效密钥尚未发生<br>1：非零 MIN 配置后的第一个有效密钥已检测到<br>注：[1] 如果 MIN = 0，此位永不被置位 [2] 如果 MIN 从非零值改回 0x0，此位自动清除 [3] 此位仅用于调试目的<br>复位类型：IORSn |
| 7-0 | MIN | R/W | 0h | 看门狗窗口阈值。这些位指定看门狗计数器复位窗口的下限。如果在计数器值达到此寄存器中的值之前通过 WDKEY 寄存器复位计数器，看门狗会立即触发复位或中断。<br>复位类型：IORSn |

## 2.15.4 NMI_INTRUPT_REGS 寄存器

表中列出了 NMI_INTRUPT_REGS 寄存器的存储器映射寄存器。所有未在表中列出的寄存器偏移地址应视为保留位置，不应修改寄存器内容。

表 2-41 NMI_INTRUPT_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | NMICFG | NMI 配置寄存器 |
| 2h | NMIFLG | NMI 标志寄存器（SYSRsn 清零） |
| 4h | NMIFLGCLR | NMI 标志清零寄存器 |
| 6h | NMIFLGFRC | NMI 标志强制寄存器 |
| 8h | NMIWDCNT | NMI 看门狗计数寄存器 |
| Ah | NMIWDPRD | NMI 看门狗周期寄存器 |

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| Ch | NMISHDFLG | NMI 影子标志寄存器 |
| Eh | ERRORSTS | 错误引脚状态 |
| 10h | ERRORSTSCLR | ERRORSTS 清零寄存器 |
| 12h | ERRORSTSFRC | ERRORSTS 强制寄存器 |
| 14h | ERRORCTL | 错误引脚控制寄存器 |
| 16h | ERRORLOCK | 错误引脚寄存器锁定寄存器。 |

复杂位访问类型经过编码以适应较小的表格单元格。表中显示了本节中使用的访问类型代码。

表 2-42 NMI_INTRUPT_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | | |
| R | R | 读取 |
| R-0 | R-0 | 读取返回 0 |
| 写入类型 | | |
| W | W | 写入 |
| W1S | W1S | 写 1 置位 |
| WSonce | WSonce | 写设置一次 |
| 复位或默认值 | | |
| - n | | 复位后或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | |
| i,j,k,l,m,n | | 当这些变量出现在寄存器名称、偏移或地址中时，它们表示寄存器数组的值，其中寄存器是重复寄存器组的一部分。寄存器组形成层次结构，数组以公式表示。 |
| y | | 当此变量出现在寄存器名称、偏移或地址中时，它表示寄存器数组的值。 |

### 2.15.4.1 NMICFG 寄存器

偏移地址：0h 复位值：0000h

NMI 配置寄存器

表 2-43 NMICFG 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | NMIE |
| R-0-0h | | | | | | | R/W1S-0h |

表 2-44 NMICFG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | NMIE | R/W1S | 0h | 当设置为 1 时，任何条件都会向 CPU 生成 NMI 中断，并启动 NMI 看门狗计数器。作为引导序列的一部分，应在器件安全相关初始化完成后设置此位。0：NMI 禁用；1：NMI 使能。复位类型：SYSRSn |

### 2.15.4.2 NMIFLG 寄存器

偏移地址：2h　复位值：0000h

NMI 标志寄存器（由 SYSRSn 清零）

表 2-45 NMIFLG 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | CRC_FAIL | SWERR | 保留 | 保留 | 保留 | 保留 | CLBNMI |
| R-0-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERADNMI | 保留 | RAM_UNMI | 保留 | FLUNCERR | RAMUNCERR | CLOCKFAIL | NMIINT |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 2-46 NMIFLG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15 | RESERVED | R-0 | 0h | 保留 |
| 14 | CRC_FAIL | R | 0h | 0：存储器后台 CRC 校验未失败。1：存储器后台 CRC 校验已失败。复位类型：SYSRSn |
| 13 | SWERR | R | 0h | 软件错误强制 NMI 标志：此位指示是否通过 NMIFLGFRC 寄存器强制产生了 NMI。此位只能由用户向 NMIFLGCLR 寄存器中的相应位写入或通过 SYSRSn 复位来清零：0：未产生软件错误强制；1：NMI 被软件强制产生。在用户清零此标志之前，不再生成进一步的 NMI 脉冲。复位类型：SYSRSn |
| 12 | RESERVED | R | 0h | 保留 |
| 11 | RESERVED | R | 0h | 保留 |
| 10 | RESERVED | R | 0h | 保留 |
| 9 | RESERVED | R | 0h | 保留 |
| 8 | CLBNMI | R | 0h | 可重构逻辑 NMI 标志：此位指示 NMI 是否由可重构逻辑产生。此位只能由用户向 NMIFLGCLR 寄存器中的相应清零位写入或通过 XRSn 复位来清零：0：无可重构逻辑 NMI 挂起；1：已产生可重构逻辑 NMI。复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 7 | ERADNMI | R | 0h | ERAD模块 NMI 标志：此位指示 NMI 是否由 ERAD模块产生。此位只能由用户向 NMIFLGCLR 寄存器中的相应清零位写入或通过 XRSn 复位来清零：0：无系统调试 NMI 挂起；1：已产生系统调试 NMI。复位类型：SYSRSn |
| 6 | RESERVED | R | 0h | 保留 |
| 5 | RAM_UNMI | R | 0h | RAM 不可纠正错误 NMI 标志：此位指示在任何主设备对 RAM 的访问中是否发生了不可纠正错误且该条件被锁存。此位只能由用户向 NMIFLGCLR 寄存器中的相应清零位写入或通过 XRSn 复位来清零：0：无 RAM 不可纠正错误条件挂起；1：已产生 RAM 不可纠正错误条件。复位类型：SYSRSn |
| 4 | RESERVED | R | 0h | 保留 |
| 3 | FLUNCERR | R | 0h | Flash 不可纠正错误 NMI 标志：此位指示在 Flash 访问时是否发生了不可纠正错误且该条件被锁存。此位只能由用户向 NMIFLGCLR 寄存器中的相应清零位写入或通过 XRSn 复位来清零：0：无 Flash 不可纠正错误条件挂起；1：已产生 Flash 不可纠正错误条件。复位类型：SYSRSn |
| 2 | RAMUNCERR | R | 0h | RAM 不可纠正错误 NMI 标志：此位指示在任何主设备对 RAM 的访问中是否发生了不可纠正错误且该条件被锁存。此位只能由用户向 NMIFLGCLR 寄存器中的相应清零位写入或通过 XRSn 复位来清零：0：无 RAM 不可纠正错误条件挂起；1：已产生 RAM 不可纠正错误条件。复位类型：SYSRSn |
| 1 | CLOCKFAIL | R | 0h | 时钟失败中断标志：这些位指示 CLOCKFAIL 条件是否被锁存。这些位只能由用户向 NMIFLGCLR 寄存器中的相应位写入或通过 XRSn 复位来清零：0：无 CLOCKFAIL 条件挂起；1：已产生 CLOCKFAIL 条件。复位类型：SYSRSn |
| 0 | NMIINT | R | 0h | NMI 中断标志：此位指示是否产生了 NMI 中断。此位只能由用户向 NMIFLGCLR 寄存器中的相应位写入或通过 XRSn 复位来清零：0：未产生 NMI 中断；1：已产生 NMI 中断。在用户清零此标志之前，不再生成进一步的 NMI 中断脉冲。复位类型：SYSRSn |

### 2.15.4.3 NMIFLGCLR 寄存器

偏移地址：4h　复位值：0000h

NMI 标志清零寄存器

表 2-47 NMIFLGCLR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | CRC_FAIL | SWERR | 保留 | 保留 | 保留 | 保留 | CLBNMI |
| R-0-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERADNMI | 保留 | RAM_UNMI | 保留 | FLUNCERR | RAMUNCERR | CLOCKFAIL | NMIINT |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 2-48 NMIFLGCLR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15 | RESERVED | R-0 | 0h | 保留 |
| 14 | CRC_FAIL | R-0/W1S | 0h | 向相应位写入 1 会清零 NMIFLG 和 NMISHDFLG 寄存器中对应的标志位。写入 0 被忽略。始终回读为 0。注：[1] 如果硬件在同一周期内试图将某位置 1，而软件试图将其清零，则硬件优先。[2] 用户应先清除挂起的 FAIL 标志，然后再清零 NMIINT 标志。复位类型：SYSRSn |
| 13 | SWERR | R-0/W1S | 0h | 向相应位写入 1 会清零 NMIFLG 和 NMISHDFLG 寄存器中对应的标志位。写入 0 被忽略。始终回读为 0。注：[1] 如果硬件在同一周期内试图将某位置 1，而软件试图将其清零，则硬件优先。[2] 用户应先清除挂起的 FAIL 标志，然后再清零 NMIINT 标志。复位类型：SYSRSn |
| 12 | RESERVED | R-0/W1S | 0h | 保留 |
| 11 | RESERVED | R-0/W1S | 0h | 保留 |
| 10 | RESERVED | R-0/W1S | 0h | 保留 |
| 9 | RESERVED | R-0/W1S | 0h | 保留 |
| 8 | CLBNMI | R-0/W1S | 0h | 向相应位写入 1 会清零 NMIFLG 和 NMISHDFLG 寄存器中对应的标志位。写入 0 被忽略。始终回读为 0。注：[1] 如果硬件在同一周期内试图将某位置 1，而软件试图将其清零，则硬件优先。[2] 用户应先清除挂起的 FAIL 标志，然后再清零 NMIINT 标志。复位类型：SYSRSn |
| 7 | ERADNMI | R-0/W1S | 0h | 向相应位写入 1 会清零 NMIFLG 和 NMISHDFLG 寄存器中对应的标志位。写入 0 被忽略。始终回读为 0。注：[1] 如果硬件在同一周期内试图将某位置 1，而软件试图将其清零，则硬件优先。[2] 用户应先清除挂起的 FAIL 标志，然后再清零 NMIINT 标志。复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 6 | RESERVED | R-0/W1S | 0h | 保留 |
| 5 | RAM_UNMI | R-0/W1S | 0h | 向相应位写入 1 会清零 NMIFLG 和 NMISHDFLG 寄存器中对应的标志位。写入 0 被忽略。始终回读为 0。注：[1] 如果硬件在同一周期内试图将某位置 1，而软件试图将其清零，则硬件优先。[2] 用户应先清除挂起的 FAIL 标志，然后再清零 NMIINT 标志。复位类型：SYSRSn |
| 4 | RESERVED | R-0/W1S | 0h | 保留 |
| 3 | FLUNCERR | R-0/W1S | 0h | 向相应位写入 1 会清零 NMIFLG 和 NMISHDFLG 寄存器中对应的标志位。写入 0 被忽略。始终回读为 0。注：[1] 如果硬件在同一周期内试图将某位置 1，而软件试图将其清零，则硬件优先。[2] 用户应先清除挂起的 FAIL 标志，然后再清零 NMIINT 标志。复位类型：SYSRSn |
| 2 | RAMUNCERR | R-0/W1S | 0h | 向相应位写入 1 会清零 NMIFLG 和 NMISHDFLG 寄存器中对应的标志位。写入 0 被忽略。始终回读为 0。注：[1] 如果硬件在同一周期内试图将某位置 1，而软件试图将其清零，则硬件优先。[2] 用户应先清除挂起的 FAIL 标志，然后再清零 NMIINT 标志。复位类型：SYSRSn |
| 1 | CLOCKFAIL | R-0/W1S | 0h | 向相应位写入 1 会清零 NMIFLG 和 NMISHDFLG 寄存器中对应的标志位。写入 0 被忽略。始终回读为 0。注：[1] 如果硬件在同一周期内试图将某位置 1，而软件试图将其清零，则硬件优先。[2] 用户应先清除挂起的 FAIL 标志，然后再清零 NMIINT 标志。复位类型：SYSRSn |
| 0 | NMIINT | R-0/W1S | 0h | 向相应位写入 1 会清零 NMIFLG 和 NMISHDFLG 寄存器中对应的标志位。写入 0 被忽略。始终回读为 0。注：[1] 如果硬件在同一周期内试图将某位置 1，而软件试图将其清零，则硬件优先。[2] 用户应先清除挂起的 FAIL 标志，然后再清零 NMIINT 标志。复位类型：SYSRSn |

### 2.15.4.4 NMIFLGFRC 寄存器

偏移地址：6h　复位值：0000h

NMI 标志强制寄存器

表 2-49 NMIFLGFRC 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | CRC_FAIL | SWERR | 保留 | 保留 | 保留 | 保留 | CLBNMI |
| R-0-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERADNMI | 保留 | RAM_UNMI | 保留 | FLUNCERR | RAMUNCERR | CLOCKFAIL | 保留 |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0-0h |

表 2-50 NMIFLGFRC 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15 | RESERVED | R-0 | 0h | 保留 |
| 14 | CRC_FAIL | R-0/W1S | 0h | 向这些位写入 1 会将 NMIFLG 和 NMISHDFLG 寄存器中相应的 FAIL 标志置位。写入 0 被忽略。始终回读为 0。这可以用作测试 NMI 机制的一种手段。复位类型：SYSRSn |
| 13 | SWERR | R-0/W1S | 0h | 向这些位写入 1 会将 NMIFLG 和 NMISHDFLG 寄存器中相应的 FAIL 标志置位。写入 0 被忽略。始终回读为 0。这可以用作测试 NMI 机制的一种手段。复位类型：SYSRSn |
| 12 | RESERVED | R-0/W1S | 0h | 保留 |
| 11 | RESERVED | R-0/W1S | 0h | 保留 |
| 10 | RESERVED | R-0/W1S | 0h | 保留 |
| 9 | RESERVED | R-0/W1S | 0h | 保留 |
| 8 | CLBNMI | R-0/W1S | 0h | 向这些位写入 1 会将 NMIFLG 和 NMISHDFLG 寄存器中相应的 FAIL 标志置位。写入 0 被忽略。始终回读为 0。这可以用作测试 NMI 机制的一种手段。复位类型：SYSRSn |
| 7 | ERADNMI | R-0/W1S | 0h | 向这些位写入 1 会将 NMIFLG 和 NMISHDFLG 寄存器中相应的 FAIL 标志置位。写入 0 被忽略。始终回读为 0。这可以用作测试 NMI 机制的一种手段。复位类型：SYSRSn |
| 6 | RESERVED | R-0/W1S | 0h | 保留 |
| 5 | RAM_UNMI | R-0/W1S | 0h | 向这些位写入 1 会将 NMIFLG 和 NMISHDFLG 寄存器中相应的 FAIL 标志置位。写入 0 被忽略。始终回读为 0。这可以用作测试 NMI 机制的一种手段。复位类型：SYSRSn |
| 4 | RESERVED | R-0/W1S | 0h | 保留 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 3 | FLUNCERR | R-0/W1S | 0h | 向这些位写入 1 会将 NMIFLG 和 NMISHDFLG 寄存器中相应的 FAIL 标志置位。写入 0 被忽略。始终回读为 0。这可以用作测试 NMI 机制的一种手段。复位类型：SYSRSn |
| 2 | RAMUNCERR | R-0/W1S | 0h | 向这些位写入 1 会将 NMIFLG 和 NMISHDFLG 寄存器中相应的 FAIL 标志置位。写入 0 被忽略。始终回读为 0。这可以用作测试 NMI 机制的一种手段。复位类型：SYSRSn |
| 1 | CLOCKFAIL | R-0/W1S | 0h | 向这些位写入 1 会将 NMIFLG 和 NMISHDFLG 寄存器中相应的 FAIL 标志置位。写入 0 被忽略。始终回读为 0。这可以用作测试 NMI 机制的一种手段。复位类型：SYSRSn |
| 0 | RESERVED | R-0 | 0h | 保留 |

### 2.15.4.5 NMIWDCNT 寄存器

偏移地址：8h　复位值：0000h

NMI 看门狗计数器寄存器

表 2-51 NMIWDCNT 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NMIWDCNT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NMIWDCNT | | | | | | | |
| R-0h | | | | | | | |

表 2-52 NMIWDCNT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-0 | NMIWDCNT | R | 0h | NMI 看门狗计数器：当任何一个已使能的 FAIL 标志被置位时，此 16 位递增计数器开始递增。如果计数器达到周期值，则会触发 NMIRSn 信号，进而复位系统。计数器在达到周期值时会复位为零，并且如果任何已使能的 FAIL 标志仍被置位，则会重新开始计数。如果没有已使能的 FAIL 标志被置位，则计数器将复位为零，并保持为零直到已使能的 FAIL 标志被置位。通常，软件会响应生成的 NMI 中断，并在 NMI 看门狗触发放大器复位之前清除有问题的标志。在某些情况下，软件可能决定仍允许看门狗复位器件。计数器以 SYSCLKOUT 速率计时。复位类型：SYSRSn |

### 2.15.4.6 NMIWDPRD 寄存器

偏移地址：Ah　复位值：FFFFh

NMI 看门狗周期寄存器

表 2-53 NMIWDPRD 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NMIWDPRD | | | | | | | |
| R/W-FFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NMIWDPRD | | | | | | | |
| R/W-FFFFh | | | | | | | |

表 2-54 NMIWDPRD 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-0 | NMIWDPRD | R/W | FFFFh | NMI 看门狗周期：此 16 位值包含当看门狗计数器匹配时产生复位的周期值。复位时此值设置为最大值。软件可以在初始化时减小周期值。写入一个小于当前计数器值的 PERIOD 值会自动强制产生 NMIRSn，从而复位看门狗计数器。复位类型：SYSRSn |

### 2.15.4.7 NMISHDFLG 寄存器

偏移地址：Ch　复位值：0000h

NMI 影子标志寄存器

表 2-55 NMISHDFLG 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | CRC_FAIL | SWERR | 保留 | 保留 | 保留 | 保留 | CLBNMI |
| R-0-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERADNMI | 保留 | RAM_UNMI | 保留 | FLUNCERR | RAMUNCERR | CLOCKFAIL | 保留 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0-0h |

表 2-56 NMISHDFLG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15 | RESERVED | R-0 | 0h | 保留 |
| 14 | CRC_FAIL | R | 0h | 影子 NMI 标志：当器件中任何可能的 NMI 源导致 NMIFLG 位置位时，此寄存器中的相应位也会置位。注意，NMIFLGFRC 和 NMIFLGCLR 寄存器对此寄存器位的影响与对 NMIFLG 寄存器相 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 同。此寄存器仅由 PORESETn 复位。注：[1] 添加此寄存器是为了保持系统控制复位原因寄存器的定义清晰。 |
| 13 | SWERR | R | 0h | 影子 NMI 标志：当器件中任何可能的 NMI 源导致 NMIFLG 位置位时，此寄存器中的相应位也会置位。注意，NMIFLGFRC 和 NMIFLGCLR 寄存器对此寄存器位的影响与对 NMIFLG 寄存器相同。此寄存器仅由 PORESETn 复位。注：[1] 添加此寄存器是为了保持系统控制复位原因寄存器的定义清晰。复位类型：PORESETn |
| 12 | RESERVED | R | 0h | 保留 |
| 11 | RESERVED | R | 0h | 保留 |
| 10 | RESERVED | R | 0h | 保留 |
| 9 | RESERVED | R | 0h | 保留 |
| 8 | CLBNMI | R | 0h | 影子 NMI 标志：当器件中任何可能的 NMI 源导致 NMIFLG 位置位时，此寄存器中的相应位也会置位。注意，NMIFLGFRC 和 NMIFLGCLR 寄存器对此寄存器位的影响与对 NMIFLG 寄存器相同。此寄存器仅由 PORESETn 复位。注：[1] 添加此寄存器是为了保持系统控制复位原因寄存器的定义清晰。复位类型：PORESETn |
| 7 | ERADNMI | R | 0h | 影子 NMI 标志：当器件中任何可能的 NMI 源导致 NMIFLG 位置位时，此寄存器中的相应位也会置位。注意，NMIFLGFRC 和 NMIFLGCLR 寄存器对此寄存器位的影响与对 NMIFLG 寄存器相同。此寄存器仅由 PORESETn 复位。注：[1] 添加此寄存器是为了保持系统控制复位原因寄存器的定义清晰。复位类型：PORESETn |
| 6 | RESERVED | R | 0h | 保留 |
| 5 | RAM_UNMI | R | 0h | 影子 NMI 标志：当器件中任何可能的 NMI 源导致 NMIFLG 位置位时，此寄存器中的相应位也会置位。注意，NMIFLGFRC 和 NMIFLGCLR 寄存器对此寄存器位的影响与对 NMIFLG 寄存器相同。此寄存器仅由 PORESETn 复位。注：[1] 添加此寄存器是为了保持系统控制复位原因寄存器的定义清晰。复位类型：PORESETn |
| 4 | RESERVED | R | 0h | 保留 |
| 3 | FLUNCERR | R | 0h | 影子 NMI 标志：当器件中任何可能的 NMI 源导致 NMIFLG 位置位时，此寄存器中的相应位也会置位。注意，NMIFLGFRC 和 NMIFLGCLR 寄存器对此寄存器位的影响与对 NMIFLG 寄存器相同。此寄存器仅由 PORESETn 复位。注：[1] 添加此寄存器是为了保持系统控制复位原因寄存器的定义清晰。复位类型：PORESETn |
| 2 | RAMUNCERR | R | 0h | 影子 NMI 标志：当器件中任何可能的 NMI 源导致 NMIFLG 位置位时，此寄存器中的相应位也会 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 置位。注意，NMIFLGFRC 和 NMIFLGCLR 寄存器对此寄存器位的影响与对 NMIFLG 寄存器相同。此寄存器仅由 PORESETn 复位。注：[1] 添加此寄存器是为了保持系统控制复位原因寄存器的定义清晰。复位类型：PORESETn |
| 1 | CLOCKFAIL | R | 0h | 影子 NMI 标志：当器件中任何可能的 NMI 源导致 NMIFLG 位置位时，此寄存器中的相应位也会置位。注意，NMIFLGFRC 和 NMIFLGCLR 寄存器对此寄存器位的影响与对 NMIFLG 寄存器相同。此寄存器仅由 PORESETn 复位。注：[1] 添加此寄存器是为了保持系统控制复位原因寄存器的定义清晰。复位类型：PORESETn |
| 0 | RESERVED | R-0 | 0h | 保留 |

## 2.15.4.8 ERRORSTS 寄存器

偏移地址：Eh　复位值：0000h

错误引脚状态

表 2-57 ERRORSTS 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | PINSTS | ERROR |
| R-0-0h | | | | | | R-0h | R-0h |

表 2-58 ERRORSTS 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | PINSTS | R | 0h | 0：错误引脚为 0；1：错误引脚为 1。复位类型：PORESETn |
| 0 | ERROR | R | 0h | 0：没有错误源被触发。1：一个或多个错误源被触发，或者通过向 ERRORSTSFRC.ERROR 位写入 1 设置了 ERRORSTS.ERROR。一旦置位，可通过向 ERRORSTSCLR.ERROR 位写入 1 来清零 ERROR 标志。以下事件/触发器可置位此位：1. CPU 上的 NMI 中断 2. 看门狗复位 3. CLIC 向量取指错误 4. eFuse 错误。读取此位时，将返回错误引脚的状态。复位类型：PORESETn |

### 2.15.4.9 ERRORSTSCLR 寄存器

偏移地址：10h　复位值：0000h

ERRORSTS 清零寄存器

表 2-59 ERRORSTSCLR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | ERROR |
| R-0-0h | | | | | | | R-0/W1S-0h |

表 2-60 ERRORSTSCLR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | ERROR | R-0/W1S | 0h | 0：无影响；1：ERRORSTS.ERROR 被清零为 0。<br>复位类型：PORESETn |

### 2.15.4.10 ERRORSTSFRC 寄存器

偏移地址：12h　复位值：0000h

表 2-61 ERRORSTSFRC 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | ERROR |
| R-0-0h | | | | | | | R-0/W1S-0h |

表 2-62 ERRORSTSFRC 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | ERROR | R-0/W1S | 0h | 0：无影响；1：ERRORSTS.ERROR 被置 1。复位类型：PORESETn |

### 2.15.4.11 ERRORCTL 寄存器

偏移地址：14h　复位值：0000h

错误引脚控制寄存器

表 2-63 ERRORCTL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | ERRORPOLSEL |
| R-0-0h | | | | | | | R/W-0h |

表 2-64 ERRORCTL 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | ERRORPOLSEL | R/W | 0h | 0：如果 ERRORSTS.ERROR 为 1，错误引脚将被驱动为 0，否则为 1。1：如果 ERRORSTS.ERROR 为 1，错误引脚将被驱动为 1，否则为 0。复位类型：PORESETn |

## 2.15.4.12 ERRORLOCK 寄存器

偏移地址：16h 复位值：0000h

错误引脚寄存器锁定寄存器

表 2-65 ERRORLOCK 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | ERRORCTL |
| R-0-0h | | | | | | | R/WSonce-0h |

表 2-66 ERRORLOCK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | ERRORCTL | R/WSonce | 0h | 0：允许写入 ERRORCTL 寄存器。1：阻止写入 ERRORCTL 寄存器。向此位写入 0 无效。写入 1 将置位此位，只能通过 SYSRSn 清零。复位类型：SYSRSn |

## 2.15.5 XINT_REGS 寄存器

表中列出了 XINT_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，且不应修改寄存器内容。

表 2-67 XINT_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | XINT1CR | XINT1 配置寄存器 |
| 1h | XINT2CR | XINT2 配置寄存器 |
| 2h | XINT3CR | XINT3 配置寄存器 |
| 3h | XINT4CR | XINT4 配置寄存器 |
| 4h | XINT5CR | XINT5 配置寄存器 |
| 8h | XINT1CTR | XINT1 计数器寄存器 |
| 9h | XINT2CTR | XINT2 计数器寄存器 |
| Ah | XINT3CTR | XINT3 计数器寄存器 |

复杂位访问类型被编码以适应小表格单元格。表中显示了本节中使用的访问类型代码。

表 2-68 XINT_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读类型 | 读类型 | 读类型 |
| R | R | 读取 |
| R-0 | R-0 | 读取返回 0 |
| 写类型 | 写类型 | 写类型 |
| W | W | 写入 |
| 复位或默认值 | 复位或默认值 | 复位或默认值 |
| - n | - n | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | 寄存器数组变量 |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中该寄存器是重复寄存器组的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。 |

### 2.15.5.1 XINT1CR 寄存器

偏移地址：0h　复位值：0000h

XINT1 配置寄存器

表 2-69 XINT1CR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | POLARITY | | 保留 | ENABLE |
| R-0-0h | | | | R/W-0h | | R-0-0h | R/W-0h |

表 2-70 XINT1CR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-4 | RESERVED | R-0 | 0h | 保留 |
| 3-2 | POLARITY | R/W | 0h | 00：中断选择为负边沿触发；01：中断选择为正边沿触发；10：中断选择为负边沿触发；11：中断选择为正或负边沿触发。复位类型：SYSRSn |
| 1 | RESERVED | R-0 | 0h | 保留 |
| 0 | ENABLE | R/W | 0h | 0：中断禁用；1：中断使能。复位类型：SYSRSn |

### 2.15.5.2 XINT2CR 寄存器

偏移地址：1h　复位值：0000h

XINT2 配置寄存器

表 2-71 XINT2CR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | POLARITY | | 保留 | ENABLE |
| R-0-0h | | | | R/W-0h | | R-0-0h | R/W-0h |

表 2-72 XINT2CR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-4 | RESERVED | R-0 | 0h | 保留 |
| 3-2 | POLARITY | R/W | 0h | 00：中断选择为负边沿触发；01：中断选择为正边沿触发；10：中断选择为负边沿触发；11：中断选择为正或负边沿触发。复位类型：SYSRSn |
| 1 | RESERVED | R-0 | 0h | 保留 |
| 0 | ENABLE | R/W | 0h | 0：中断禁用；1：中断使能。复位类型：SYSRSn |

### 2.15.5.3 XINT3CR 寄存器

偏移地址：2h　复位值：0000h

XINT3 配置寄存器

表 2-73 XINT3CR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | POLARITY | | 保留 | ENABLE |
| R-0-0h | | | | R/W-0h | | R-0-0h | R/W-0h |

表 2-74 XINT3CR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-4 | RESERVED | R-0 | 0h | 保留 |
| 3-2 | POLARITY | R/W | 0h | 00：中断选择为负边沿触发；01：中断选择为正边沿触发；10：中断选择为负边沿触发；11：中断选择为正或负边沿触发。复位类型：SYSRSn |
| 1 | RESERVED | R-0 | 0h | 保留 |
| 0 | ENABLE | R/W | 0h | 0：中断禁用；1：中断使能。复位类型：SYSRSn |

## 2.15.5.4 XINT4CR 寄存器

偏移地址：3h　复位值：0000h

XINT4 配置寄存器

表 2-75 XINT4CR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | POLARITY | | 保留 | ENABLE |
| R-0-0h | | | | R/W-0h | | R-0-0h | R/W-0h |

表 2-76 XINT4CR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-4 | RESERVED | R-0 | 0h | 保留 |
| 3-2 | POLARITY | R/W | 0h | 00：中断选择为负边沿触发；01：中断选择为正边沿触发；10：中断选择为负边沿触发；11：中断选择为正或负边沿触发。复位类型：SYSRSn |
| 1 | RESERVED | R-0 | 0h | 保留 |
| 0 | ENABLE | R/W | 0h | 0：中断禁用；1：中断使能。复位类型：SYSRSn |

### 2.15.5.5 XINT5CR 寄存器

偏移地址：4h　复位值：0000h

XINT5 配置寄存器

表 2-77 XINT5CR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | POLARITY | | 保留 | ENABLE |
| R-0-0h | | | | R/W-0h | | R-0-0h | R/W-0h |

表 2-78 XINT5CR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-4 | RESERVED | R-0 | 0h | 保留 |
| 3-2 | POLARITY | R/W | 0h | 00：中断选择为负边沿触发；01：中断选择为正边沿触发；10：中断选择为负边沿触发；11：中断选择为正或负边沿触发。复位类型：SYSRSn |
| 1 | RESERVED | R-0 | 0h | 保留 |
| 0 | ENABLE | R/W | 0h | 0：中断禁用；1：中断使能。复位类型：SYSRSn |

### 2.15.5.6 XINT1CTR 寄存器

偏移地址：8h　复位值：0000h

XINT1 计数器寄存器

表 2-79 XINT1CTR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| INTCTR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTCTR | | | | | | | |
| R-0h | | | | | | | |

表 2-80 XINT1CTR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-0 | INTCTR | R | 0h | 这是一个自由运行的 16 位递增计数器，以 SYSCLKOUT 速率计时。当检测到有效中断边沿时，计数器值复位为 0x0000，然后继续计数，直到检测到下一个有效中断边沿。计数器只能由相应中断控制寄存器中所选的 POLARITY 边沿复位。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 当中断被禁用时，计数器将停止。计数器是自由运行计数器，达到最大值时会回绕到零。计数器是只读寄存器，只能通过有效中断边沿或复位来清零。<br>复位类型：SYSRSn |

### 2.15.5.7 XINT2CTR 寄存器

偏移地址：9h　复位值：0000h

XINT2 计数器寄存器

表 2-81 XINT2CTR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| INTCTR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTCTR | | | | | | | |
| R-0h | | | | | | | |

表 2-82 XINT2CTR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-0 | INTCTR | R | 0h | 这是一个自由运行的 16 位递增计数器，以 SYSCLKOUT 速率计时。当检测到有效中断边沿时，计数器值复位为 0x0000，然后继续计数，直到检测到下一个有效中断边沿。计数器只能由相应中断控制寄存器中所选的 POLARITY 边沿复位。当中断被禁用时，计数器将停止。计数器是自由运行计数器，达到最大值时会回绕到零。计数器是只读寄存器，只能通过有效中断边沿或复位来清零。<br>复位类型：SYSRSn |

### 2.15.5.8 XINT3CTR 寄存器

偏移地址：Ah　复位值：0000h

XINT3 计数器寄存器

表 2-83 XINT3CTR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| INTCTR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTCTR | | | | | | | |
| R-0h | | | | | | | |

表 2-84 XINT3CTR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-0 | INTCTR | R | 0h | 这是一个自由运行的 16 位递增计数器，以 SYSCLKOUT 速率计时。当检测到有效中断边沿时，计数器值复位为 0x0000，然后继续计数，直到检测到下一个有效中断边沿。计数器只能由相应中断控制寄存器中所选的 POLARITY 边沿复位。当中断被禁用时，计数器将停止。计数器是自由运行计数器，达到最大值时会回绕到零。计数器是只读寄存器，只能通过有效中断边沿或复位来清零。复位类型：SYSRSn |

## 2.15.6 SYNC_SOC_REGS 寄存器

表中列出了 SYNC_SOC_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，且不应修改寄存器内容。

表 2-85 SYNC_SOC_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | SYNCSELECT | 同步输入和输出选择寄存器 |
| 2h | ADCSOCOUTSELECT | 外部 ADCSOC 选择寄存器 |
| 4h | SYNCSOCLOCK | SYNCSEL 和 EXTADCSOC 选择锁定寄存器 |

复杂位访问类型被编码以适应小表格单元格。表中显示了本节中使用的访问类型代码。

表 2-86 SYNC_SOC_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读类型 | | |
| R R-0 | R R-0 | 读取；读取返回 0 |
| 写类型 | | |
| W | W | 写入 |
| WSonce | WSonce | 写 1 置位一次 |
| 复位或默认值 | | |
| - n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中该寄存器是重复寄存器组的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。 |

### 2.15.6.1 SYNCSELECT 寄存器

偏移地址：0h　复位值：E003FFFFh

同步输入和输出选择寄存器

表 2-87 SYNCSELECT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | SYNCOUT | | | | |
| R/W-7h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | 保留 | |
| R-0-0h | | | | | | R/W-7h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | 保留 | | | 保留 | | | 保留 |
| R/W-7h | R/W-7h | | | R/W-7h | | | R/W-7h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | 保留 | | 保留 | | | |
| R/W-7h | | R/W-7h | | R/W-7h | | | |

表 2-88 SYNCSELECT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-29 | RESERVED | R/W | 7h | 保留 |
| 28-24 | SYNCOUT | R/W | 0h | 选择 SYNCOUT 源：00000：选择 EPWM1SYNCOUT 驱动 SYNCOUT 引脚。00001：选择 EPWM2SYNCOUT 驱动 SYNCOUT 引脚。00010：选择 EPWM3SYNCOUT 驱动 SYNCOUT 引脚。00011：选择 EPWM4SYNCOUT 驱动 SYNCOUT 引脚。00100：选择 EPWM5SYNCOUT 驱动 SYNCOUT 引脚。00101：选择 EPWM6SYNCOUT 驱动 SYNCOUT 引脚。00110：选择 EPWM7SYNCOUT 驱动 SYNCOUT 引脚。00111：保留 01000：保留 01001：保留 01010：保留 01011：保留 01100：保留 01101：保留 01110：保留 01111：保留 10000：保留 10001：保留 10010：保留 10011：保留 10100：保留 10101：保留 10110：保留 10111：保留 11000：选择 ECAP1SYNCOUT 驱动 SYNCOUT 引脚。11001：选择 ECAP2SYNCOUT 驱动 SYNCOUT 引脚。11010：选择 ECAP3SYNCOUT 驱动 SYNCOUT 引脚。11011：保留 11100：保留 11101：保留 11110：保留 11111：保留 注：[1] 保留位置默认为 00 选择。复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 23-18 | RESERVED | R-0 | 0h | 保留 |
| 17-15 | RESERVED | R/W | 7h | 保留 |
| 14-12 | RESERVED | R/W | 7h | 保留 |
| 11-9 | RESERVED | R/W | 7h | 保留 |
| 8-6 | RESERVED | R/W | 7h | 保留 |
| 5-3 | RESERVED | R/W | 7h | 保留 |
| 2-0 | RESERVED | R/W | 7h | 保留 |

### 2.15.6.2 ADCSOCOUTSELECT 寄存器

偏移地址：2h　复位值：00000000h

外部 ADCSOC 选择寄存器

表 2-89 ADCSOCOUTSELECT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | 保留 | 保留 | 保留 | 保留 |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | PWM7SOCBEN | PWM6SOCBEN | PWM5SOCBEN | PWM4SOCBEN | PWM3SOCBEN | PWM2SOCBEN | PWM1SOCBEN |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | 保留 | 保留 | 保留 | 保留 |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | PWM7SOCAEN | PWM6SOCAEN | PWM5SOCAEN | PWM4SOCAEN | PWM3SOCAEN | PWM2SOCAEN | PWM1SOCAEN |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-90 ADCSOCOUTSELECT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-28 | RESERVED | R-0 | 0h | 保留 |
| 27 | RESERVED | R/W | 0h | 保留 |
| 26 | RESERVED | R/W | 0h | 保留 |
| 25 | RESERVED | R/W | 0h | 保留 |
| 24 | RESERVED | R/W | 0h | 保留 |
| 23 | RESERVED | R/W | 0h | 保留 |
| 22 | PWM7SOCBEN | R/W | 0h | ADCSOCBOn 源选择：0：不选择相应 EPWM SOCB 输出；1：选择相应 EPWM SOCB 输出。<br>复位类型：SYSRSn |
| 21 | PWM6SOCBEN | R/W | 0h | ADCSOCBOn 源选择：0：不选择相应 EPWM SOCB 输出；1：选择相应 EPWM SOCB 输出。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 20 | PWM5SOCBEN | R/W | 0h | ADCSOCBOn 源选择：0：不选择相应 EPWM SOCB 输出；1：选择相应 EPWM SOCB 输出。<br>复位类型：SYSRSn |
| 19 | PWM4SOCBEN | R/W | 0h | ADCSOCBOn 源选择：0：不选择相应 EPWM SOCB 输出；1：选择相应 EPWM SOCB 输出。<br>复位类型：SYSRSn |
| 18 | PWM3SOCBEN | R/W | 0h | ADCSOCBOn 源选择：0：不选择相应 EPWM SOCB 输出；1：选择相应 EPWM SOCB 输出。<br>复位类型：SYSRSn |
| 17 | PWM2SOCBEN | R/W | 0h | ADCSOCBOn 源选择：0：不选择相应 EPWM SOCB 输出；1：选择相应 EPWM SOCB 输出。<br>复位类型：SYSRSn |
| 16 | PWM1SOCBEN | R/W | 0h | ADCSOCBOn 源选择：0：不选择相应 EPWM SOCB 输出；1：选择相应 EPWM SOCB 输出。<br>复位类型：SYSRSn |
| 15-12 | RESERVED | R-0 | 0h | 保留 |
| 11 | RESERVED | R/W | 0h | 保留 |
| 10 | RESERVED | R/W | 0h | 保留 |
| 9 | RESERVED | R/W | 0h | 保留 |
| 8 | RESERVED | R/W | 0h | 保留 |
| 7 | RESERVED | R/W | 0h | 保留 |
| 6 | PWM7SOCAEN | R/W | 0h | ADCSOCAOn 源选择：0：不选择相应 EPWM SOCA 输出；1：选择相应 EPWM SOCA 输出。<br>复位类型：SYSRSn |
| 5 | PWM6SOCAEN | R/W | 0h | ADCSOCAOn 源选择：0：不选择相应 EPWM SOCA 输出；1：选择相应 EPWM SOCA 输出。<br>复位类型：SYSRSn |
| 4 | PWM5SOCAEN | R/W | 0h | ADCSOCAOn 源选择：0：不选择相应 EPWM SOCA 输出；1：选择相应 EPWM SOCA 输出。<br>复位类型：SYSRSn |
| 3 | PWM4SOCAEN | R/W | 0h | ADCSOCAOn 源选择：0：不选择相应 EPWM SOCA 输出；1：选择相应 EPWM SOCA 输出。<br>复位类型：SYSRSn |
| 2 | PWM3SOCAEN | R/W | 0h | ADCSOCAOn 源选择：0：不选择相应 EPWM SOCA 输出；1：选择相应 EPWM SOCA 输出。<br>复位类型：SYSRSn |
| 1 | PWM2SOCAEN | R/W | 0h | ADCSOCAOn 源选择：0：不选择相应 EPWM SOCA 输出；1：选择相应 EPWM SOCA 输出。<br>复位类型：SYSRSn |
| 0 | PWM1SOCAEN | R/W | 0h | ADCSOCAOn 源选择：0：不选择相应 EPWM SOCA 输出；1：选择相应 EPWM SOCA 输出。<br>复位类型：SYSRSn |

### 2.15.6.3 SYNCSOCLOCK 寄存器

偏移地址：4h　复位值：00000000h

SYNCSEL 和 EXTADCSOC 选择锁定寄存器

表 2-91 SYNCSOCLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | ADCSOCOUTSELECT | SYNCSELECT |
| R-0-0h | | | | | | R/WSonce-0h | R/WSonce-0h |

表 2-92 SYNCSOCLOCK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | ADCSOCOUTSELECT | R/WSonce | 0h | ADCSOCOUTSELECT 寄存器锁定位：0：相应寄存器未锁定；1：相应寄存器已锁定。注：[1] 此寄存器中的任一位一旦置位，只能通过 SYSRSn 清零。向此寄存器的任何位写入 0 无效。[2] 锁定机制仅适用于写入。对具有 LOCK 保护的寄存器的读取始终允许。复位类型：SYSRSn |
| 0 | SYNCSELECT | R/WSonce | 0h | SYNCSELECT 寄存器锁定位：0：相应寄存器未锁定；1：相应寄存器已锁定。注：[1] 此寄存器中的任一位一旦置位，只能通过 SYSRSn 清零。向此寄存器的任何位写入 0 无效。[2] 锁定机制仅适用于写入。对具有 LOCK 保护的寄存器的读取始终允许。复位类型：SYSRSn |

## 2.15.7 DMA_CLA_SRC_SEL_REGS 寄存器

表中列出了 DMA_CLA_SRC_SEL_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，且不应修改寄存器内容。

表 2-93 DMA_CLA_SRC_SEL_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 4h | DMACHSRCSELLOCK | DMA 通道触发源选择锁定寄存器 |
| 16h | DMACHSRCSEL1 | DMA 通道触发源选择寄存器 1 |
| 18h | DMACHSRCSEL2 | DMA 通道触发源选择寄存器 2 |

复杂位访问类型被编码以适应小表格单元格。表中显示了本节中使用的访问类型代码。

表 2-94 DMA_CLA_SRC_SEL_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读类型 | | |
| R R-0 | R R-0 | 读取；读取返回 0 |
| 写类型 | | |
| W | W | 写入 |
| WSonce | WSonce | 写 1 置位一次 |
| 复位或默认值 | | |
| - n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中该寄存器是重复寄存器组的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。 |

### 2.15.7.1 DMACHSRCSELLOCK 寄存器

偏移地址：4h 复位值：00000000h

DMA 通道触发源选择锁定寄存器

表 2-95 DMACHSRCSELLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | DMACHSRCSEL2 | DMACHSRCSEL1 |
| R-0-0h | | | | | | R/WSonce-0h | R/WSonce-0h |

表 2-96 DMACHSRCSELLOCK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | DMACHSRCSEL2 | R/WSonce | 0h | DMACHSRCSEL2 寄存器锁定位：0：相应寄存器未锁定；1：相应寄存器已锁定。注：[1] 此寄存器中的任一位一旦置位，只能通过 SYSRSn 清零。向此寄存器的任何位写入 0 无效。[2] 锁定机制仅适用于写入。对具有 LOCK 保护的寄存器的读取始终允许。复位类型：SYSRSn |
| 0 | DMACHSRCSEL1 | R/WSonce | 0h | DMACHSRCSEL1 寄存器锁定位：0：相应寄存器未锁定；1：相应寄存器已锁定。注：[1] 此寄存器中的任一位一旦置位，只能通过 SYSRSn 清零。向此寄存器的任何位写入 0 无效。[2] 锁定机制仅适用于写入。对具有 LOCK 保护的寄存器的读取始终允许。复位类型：SYSRSn |

## 2.15.7.2 DMACHSRCSEL1 寄存器

偏移地址：16h　复位值：00000000h

DMA 通道触发源选择寄存器 1

表 2-97 DMACHSRCSEL1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CH4 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CH3 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CH2 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CH1 | | | | | | | |
| R/W-0h | | | | | | | |

表 2-98 DMACHSRCSEL1 寄存器字段描述

| 位 | 字 段 | 类 型 | 复 位 | 描 述 |
|---|---|---|---|---|
| 31-24 | CH4 | R/W | 0h | 选择 DMA 的触发和同步源 CH4。复位类型：SYSRSn |
| 23-16 | CH3 | R/W | 0h | 选择 DMA 的触发和同步源 CH3。复位类型：SYSRSn |
| 15-8 | CH2 | R/W | 0h | 选择 DMA 的触发和同步源 CH2。复位类型：SYSRSn |
| 7-0 | CH1 | R/W | 0h | 选择 DMA 的触发和同步源 CH1。复位类型：SYSRSn |

### 2.15.7.3 DMACHSRCSEL2 寄存器

偏移地址：18h　复位值：00000000h

DMA 通道触发源选择寄存器 2

表 2-99 DMACHSRCSEL2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CH6 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CH5 | | | | | | | |
| R/W-0h | | | | | | | |

表 2-100 DMACHSRCSEL2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R-0 | 0h | 保留 |
| 15-8 | CH6 | R/W | 0h | 选择 DMA 的触发和同步源 CH6。复位类型：SYSRSn |
| 7-0 | CH5 | R/W | 0h | 选择 DMA 的触发和同步源 CH5。复位类型：SYSRSn |

## 2.15.8 DEV_CFG_REGS 寄存器

表中列出了 DEV_CFG_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，且不应修改寄存器内容。

表 2-101 DEV_CFG_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 10h | PARTIDL | 器件部件标识号的低 32 位 |
| 14h | PARTIDH | 器件部件标识号的高 32 位 |
| 18h | REVID | 器件修订号 |
| 74h | DC21 | 器件能力：CLB |
| 104h | SOFTPRES0 | 处理块软件复位寄存器 |
| 10Ch | SOFTPRES2 | 外设软件复位寄存器 |
| 110h | SOFTPRES3 | 外设软件复位寄存器 |
| 114h | SOFTPRES4 | 外设软件复位寄存器 |
| 120h | SOFTPRES7 | 外设软件复位寄存器 |
| 124h | SOFTPRES8 | 外设软件复位寄存器 |
| 128h | SOFTPRES9 | 外设软件复位寄存器 |
| 12Ch | SOFTPRES10 | 外设软件复位寄存器 |
| 138h | SOFTPRES13 | 外设软件复位寄存器 |
| 13Ch | SOFTPRES14 | 外设软件复位寄存器 |
| 148h | SOFTPRES17 | 外设软件复位寄存器 |
| 14Ch | SOFTPRES18 | 外设软件复位寄存器 |
| 150h | SOFTPRES19 | 外设软件复位寄存器 |
| 154h | SOFTPRES20 | 外设软件复位寄存器 |
| 158h | SOFTPRES21 | 外设软件复位寄存器 |
| 168h | SOFTPRES25 | 外设软件复位寄存器 |
| 260h | TAP_STATUS | JTAG 状态机和调试器连接状态 |
| 336h | ECAPTYPE | 配置器件的 ECAP 类型 |
| 340h | CANTYPE | 配置器件的 CAN 类型 |

复杂位访问类型被编码以适应小表格单元格。表中显示了本节中使用的访问类型代码。

表 2-102 DEV_CFG_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读类型 | 读类型 | 读类型 |
| R | R | 读取 |
| R-0 | R-0 | 读取返回 0 |
| 写类型 | 写类型 | 写类型 |
| W | W | 写入 |
| W1C | W1C | 写 1 清零 |
| W1S | W1S | 写 1 置位 |
| WOnce | WOnce | 写入一次 |
| WSonce | WSonce | 写 1 置位一次 |
| 复位或默认值 | 复位或默认值 | 复位或默认值 |
| - n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | 寄存器数组变量 |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中该寄存器是重复寄存器组的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。 |

### 2.15.8.1 PARTIDL 寄存器

偏移地址：10h 复位值：00XXXXX0h

器件部件标识号的低 32 位

表 2-103 PARTIDL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | 保留 | | | |
| R-0h | | | | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FLASH_SIZE | | | | | | | |
| R-X | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | INSTASPIN | | 保留 | 保留 | PIN_COUNT | | |
| R-0h | R-X | | R-0h | R-X | R-X | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QUAL | | 保留 | 保留 | | 保留 | | |
| R-X | | R-0h | R-0h | | R-0h | | |

表 2-104 PARTIDL 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-28 | RESERVED | R | 0h | 保留 |
| 27-24 | RESERVED | R | 0h | 保留 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 23-16 | FLASH_SIZE | R | X | 5 = 128KB；4 = 64KB；3 = 32KB。复位类型：PORESETn |
| 15 | RESERVED | R | 0h | 保留 |
| 14-13 | INSTASPIN | R | X | 1 = InstaSPIN-FOC；2 = 无；3 = 无。复位类型：PORESETn |
| 12 | RESERVED | R | 0h | 保留 |
| 11 | RESERVED | R | X | 保留 |
| 10-8 | PIN_COUNT | R | X | 0 = 56 引脚（保留）；1 = 64 引脚（Q100）；2 = 64 引脚；3 = 80 引脚；4 = 48 引脚；5 = 100 引脚（保留）；6 = 保留；7 = 保留。复位类型：PORESETn |
| 7-6 | QUAL | R | X | 0 = 工程样品（TMX）；1 = 先期生产（TMP）；2 = 完全合格（TMS）。复位类型：PORESETn |
| 5 | RESERVED | R | 0h | 保留 |
| 4-3 | RESERVED | R | 0h | 保留 |
| 2-0 | RESERVED | R | 0h | 保留 |

### 2.15.8.2 PARTIDH 寄存器

偏移地址：14h　复位值：04XX0500h

器件部件标识号的高 32 位

表 2-105 PARTIDH 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DEVICE_CLASS_ID | | | | | | | |
| R-4h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PARTNO | | | | | | | |
| R-X | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FAMILY | | | | | | | |
| R-5h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | | | |
| R-0h | | | | R-0h | | | |

表 2-106 PARTIDH 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | DEVICE_CLASS_ID | R | 4h | 器件类别 ID。复位类型：PORESETn |
| 23-16 | PARTNO | R | X | 器件部件号请参阅数据手册。复位类型：PORESETn |
| 15-8 | FAMILY | R | 5h | 器件系列。复位类型：PORESETn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 7-4 | RESERVED | R | 0h | 保留 |
| 3-0 | RESERVED | R | 0h | 保留 |

### 2.15.8.3 REVID 寄存器

偏移地址：18h　复位值：00000000h

器件修订号

表 2-107 REVID 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| REVID | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| REVID | | | | | | | |
| R-0h | | | | | | | |

表 2-108 REVID 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R-0 | 0h | 保留 |
| 15-0 | REVID | R | 0h | 这 16 位指定硅修订版本。详细信息请参见器件专用数据手册。复位类型：不适用 |

### 2.15.8.4 DC21 寄存器

偏移地址：74h　复位值：00000003h

器件能力：CLB

表 2-109 DC21 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | CLB2 | CLB1 |
| R-0h | | | | | | R-1h | R-1h |

注：位 3,2,1,0 未在原表中列出。

表 2-110 DC21 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | CLB2 | R | 1h | CLB2：<br>0：该功能在设备上不存在<br>1：该功能在设备上存在<br>复位类型：PORESETn |
| 0 | CLB1 | R | 1h | CLB1：<br>0：该功能在设备上不存在<br>1：该功能在设备上存在<br>复位类型：PORESETn |

### 2.15.8.5 SOFTPRES0 寄存器

偏移地址：104h　复位值：00000000h

处理模块软件复位寄存器

当该寄存器中的位被置位时，相应的模块处于复位状态。所有设计数据都会丢失，模块寄存器返回其复位状态。位在置位后必须手动清零。

表 2-111 SOFTPRES0 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | CPU1_ERAD |
| R-0-0h | | | | | | | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | CPU1_CPUBGCRC | 保留 | | | | |
| R-0-0h | | R/W-0h | R-0-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | 保留 | 保留 | 保留 |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-112 SOFTPRES0 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-25 | RESERVED | R-0 | 0h | 保留 |
| 24 | CPU1_ERAD | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 23-16 | RESERVED | R-0 | 0h | 保留 |
| 15-14 | RESERVED | R-0 | 0h | 保留 |
| 13 | CPU1_CPUBGCRC | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 12-4 | RESERVED | R-0 | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | RESERVED | R/W | 0h | 保留 |
| 1 | RESERVED | R/W | 0h | 保留 |
| 0 | RESERVED | R/W | 0h | 保留 |

### 2.15.8.6 SOFTPRES2 寄存器

偏移地址：10Ch　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-113 SOFTPRES2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | EPWM7 | EPWM6 | EPWM5 | EPWM4 | EPWM3 | EPWM2 | EPWM1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-114 SOFTPRES2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R-0 | 0h | 保留 |
| 15 | RESERVED | R/W | 0h | 保留 |
| 14 | RESERVED | R/W | 0h | 保留 |
| 13 | RESERVED | R/W | 0h | 保留 |
| 12 | RESERVED | R/W | 0h | 保留 |
| 11 | RESERVED | R/W | 0h | 保留 |
| 10 | RESERVED | R/W | 0h | 保留 |
| 9 | RESERVED | R/W | 0h | 保留 |
| 8 | RESERVED | R/W | 0h | 保留 |
| 7 | RESERVED | R/W | 0h | 保留 |
| 6 | EPWM7 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 5 | EPWM6 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 4 | EPWM5 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 3 | EPWM4 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 2 | EPWM3 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 1 | EPWM2 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 0 | EPWM1 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

### 2.15.8.7 SOFTPRES3 寄存器

偏移地址：110h　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-115 SOFTPRES3 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | 保留 | 保留 | 保留 | ECAP3 | ECAP2 | ECAP1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-116 SOFTPRES3 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-8 | RESERVED | R-0 | 0h | 保留 |
| 7 | RESERVED | R/W | 0h | 保留 |
| 6 | RESERVED | R/W | 0h | 保留 |
| 5 | RESERVED | R/W | 0h | 保留 |
| 4 | RESERVED | R/W | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | ECAP3 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 1 | ECAP2 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 0 | ECAP1 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

### 2.15.8.8 SOFTPRES4 寄存器

偏移地址：114h　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-117 SOFTPRES4 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | 保留 | EQEP2 | EQEP1 |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-118 SOFTPRES4 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | RESERVED | R/W | 0h | 保留 |
| 1 | EQEP2 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 0 | EQEP1 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

### 2.15.8.9 SOFTPRES7 寄存器

偏移地址：120h　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-119 SOFTPRES7 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | 保留 | 保留 | SCI_A |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-120 SOFTPRES7 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | RESERVED | R/W | 0h | 保留 |
| 1 | RESERVED | R/W | 0h | 保留 |
| 0 | SCI_A | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

## 2.15.8.10 SOFTPRES8 寄存器

偏移地址：124h　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-121 SOFTPRES8 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | 保留 | 保留 |
| R-0-0h | | | | | | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | 保留 | SPI_B | SPI_A |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-122 SOFTPRES8 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-18 | RESERVED | R-0 | 0h | 保留 |
| 17 | RESERVED | R/W | 0h | 保留 |
| 16 | RESERVED | R/W | 0h | 保留 |
| 15-4 | RESERVED | R-0 | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | RESERVED | R/W | 0h | 保留 |
| 1 | SPI_B | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 0 | SPI_A | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

### 2.15.8.11 SOFTPRES9 寄存器

偏移地址：128h　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-123 SOFTPRES9 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | I2C_B | I2C_A |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-124 SOFTPRES9 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | I2C_B | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 0 | I2C_A | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

## 2.15.8.12 SOFTPRES10 寄存器

偏移地址：12Ch　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-125 SOFTPRES10 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | 保留 | 保留 | CAN_A |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-126 SOFTPRES10 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | RESERVED | R/W | 0h | 保留 |
| 1 | RESERVED | R/W | 0h | 保留 |
| 0 | CAN_A | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

### 2.15.8.13 SOFTPRES13 寄存器

偏移地址：138h　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-127 SOFTPRES13 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | ADC_C | 保留 | ADC_A |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-128 SOFTPRES13 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | ADC_C | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 1 | RESERVED | R/W | 0h | 保留 |
| 0 | ADC_A | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

### 2.15.8.14 SOFTPRES14 寄存器

偏移地址：13Ch　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-129 SOFTPRES14 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | 保留 | 保留 | CMPSS4 | CMPSS3 | CMPSS2 | CMPSS1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-130 SOFTPRES14 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-8 | RESERVED | R-0 | 0h | 保留 |
| 7 | RESERVED | R/W | 0h | 保留 |
| 6 | RESERVED | R/W | 0h | 保留 |
| 5 | RESERVED | R/W | 0h | 保留 |
| 4 | RESERVED | R/W | 0h | 保留 |
| 3 | CMPSS4 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 2 | CMPSS3 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 1 | CMPSS2 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 0 | CMPSS1 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

## 2.15.8.15 SOFTPRES17 寄存器

偏移地址：148h　复位值：0000000Xh

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-131 SOFTPRES17 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | 保留 | CLB2 | CLB1 |
| R-0h | | | | R/W-0h | R/W-0h | R-X | R-X |

表 2-132 SOFTPRES17 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | RESERVED | R/W | 0h | 保留 |
| 1 | CLB2 | R | X | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 0 | CLB1 | R | X | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

### 2.15.8.16 SOFTPRES18 寄存器

偏移地址：14Ch　复位值：0000000Xh

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-133 SOFTPRES18 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | 保留 | FSIRX_A | FSITX_A |
| R-0h | | | | R/W-0h | R/W-0h | R-X | R-X |

表 2-134 SOFTPRES18 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | RESERVED | R/W | 0h | 保留 |
| 1 | FSIRX_A | R | X | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 0 | FSITX_A | R | X | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

### 2.15.8.17 SOFTPRES19 寄存器

偏移地址：150h　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-135 SOFTPRES19 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | 保留 | LIN_B | LIN_A |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-136 SOFTPRES19 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | RESERVED | R/W | 0h | 保留 |
| 1 | LIN_B | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 0 | LIN_A | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

### 2.15.8.18 SOFTPRES20 寄存器

偏移地址：154h　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-137 SOFTPRES20 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | 保留 | PMBUS_A |
| R-0h | | | | | | R-0h | R-0h |

表 2-138 SOFTPRES20 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | RESERVED | R | 0h | 保留 |
| 0 | PMBUS_A | R | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

## 2.15.8.19 SOFTPRES21 寄存器

偏移地址：158h　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-139 SOFTPRES21 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | DCC1 | DCC0 |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-140 SOFTPRES21 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | DCC1 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |
| 0 | DCC0 | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

## 2.15.8.20 SOFTPRES25 寄存器

偏移地址：168h　复位值：00000000h

外设软件复位寄存器

当该寄存器中的位被置位时，相应的外设处于复位状态。所有数据都会丢失，外设寄存器返回其复位状态。位在置位后必须手动清零。

表 2-141 SOFTPRES25 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | HIC_A |
| R-0-0h | | | | | | | R/W-0h |

表 2-142 SOFTPRES25 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | HIC_A | R/W | 0h | 1：模块处于复位状态。0：模块复位由正常器件复位结构决定。复位类型：SYSRSn |

## 2.15.8.21 TAP_STATUS 寄存器

偏移地址：260h　复位值：00000000h

JTAG 状态机和调试器连接状态

表 2-143 TAP_STATUS 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DCON | 保留 | | | | | | |
| R-0h | R-0-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TAP_STATE | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TAP_STATE | | | | | | | |
| R-0h | | | | | | | |

表 2-144 TAP_STATUS 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | DCON | R | 0h | 来自 IcePick 的调试连接指示。复位类型：PORESETn |
| 30-16 | RESERVED | R-0 | 0h | 保留 |
| 15-0 | TAP_STATE | R | 0h | TAP 状态向量。这些位表示连接对应的 POTAP* 输出到：0：TLR，1：IDLE，2：SELECTDR，3：CAPDR，4：SHIFTDR，5：EXIT1DR，6：PAUSEDR，7：EXIT2DR，8：UPDTDR，9：SLECTIR，10：CAPIR，11：SHIFTIR，12：EXIT1IR，13：PAUSEIR，14：EXIT2IR，15：UPDTIR。复位类型：PORESETn |

## 2.15.8.22 ECAPTYPE 寄存器

偏移地址：336h　复位值：0000h

根据配置使能或禁用与 ECAP 类型相关的功能。

表 2-145 ECAPTYPE 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| LOCK | 保留 | | | | | | |
| R/WSonce-0h | R-0-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | TYPE | |
| R-0-0h | | | | | | R/W-0h | |

表 2-146 ECAPTYPE 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15 | LOCK | R/WSonce | 0h | 1：不允许写入该寄存器。0：允许写入该寄存器。复位类型：SYSRSn |
| 14-2 | RESERVED | R-0 | 0h | 保留 |
| 1-0 | TYPE | R/W | 0h | ‘00，10，11’：1. ECAP 寄存器不受 EALLOW 保护。‘01’：1. ECAP 寄存器受 EALLOW 保护。复位类型：SYSRSn |

## 2.15.8.23 CANTYPE 寄存器

偏移地址：340h　复位值：0000h

根据配置使能或禁用与 CAN 类型相关的功能。

表 2-147 CANTYPE 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | CANTYPE |
| R-0-0h | | | | | | | R/W-0h |

表 2-148 CANTYPE 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | CANTYPE | R/W | 0h | 0：默认CAN外设；1：配置为1，启用FDCAN。复位类型：SYSRSn |

## 2.15.9 CLK_CFG_REGS 寄存器

表中列出了 CLK_CFG_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址应视为保留位置，不应修改寄存器内容。

表 2-149 CLK_CFG_REGS 寄存器

| 偏移 | 简称 | 寄存器名称 |
|---|---|---|
| 4h | CLKCFGLOCK1 | CLKCFG 寄存器锁定位 |
| 10h | CLKSRCCTL1 | 时钟源控制寄存器 1 |
| 14h | CLKSRCCTL2 | 时钟源控制寄存器 2 |
| 18h | CLKSRCCTL3 | 时钟源控制寄存器 3 |
| 1Ch | SYSPLLCTL1 | SYSPLL 控制寄存器 1 |
| 28h | SYSPLLMULT | SYSPLL 倍频器寄存器 |
| 2Ch | SYSPLLSTS | SYSPLL 状态寄存器 |
| 44h | SYSCLKDIVSEL | 系统时钟分频器选择寄存器 |
| 50h | XCLKOUTDIVSEL | XCLKOUT 分频器选择寄存器 |
| 58h | LOSPCP | 低速时钟源预分频器 |
| 5Ch | MCDCR | 时钟丢失检测控制寄存器 |
| 60h | X1CNT | X1 时钟上的 10 位计数器 |
| 78h | CLKFAILCFG | 时钟失效原因配置 |

复杂位访问类型编码为适合小表格单元格。表中显示了本节中使用的访问类型代码。

表 2-150 CLK_CFG_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读类型 | | |
| R | R | 读取 |
| R-0 | R -0 | 读取返回 0 |
| 写类型 | | |

| 访问类型 | 代码 | 描述 |
| --- | --- | --- |
| W | W | 写入 |
| W1C | W 1C | 写 1 清零 |
| W1S | W 1S | 写 1 置位 |
| WSonce | W Sonce | 写入设置一次 |
| 复位或默认值 | | |
| - n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | |
| i，j，k，l，m，n | | 当这些变量用于寄存器名称、偏移量或地址时，它们指的是寄存器数组的值，其中该寄存器是重复寄存器组的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | 当此变量用于寄存器名称、偏移量或地址时，它指的是寄存器数组的值。 | |

### 2.15.9.1 CLKCFGLOCK1 寄存器

偏移地址：4h　复位值：00000000h

CLKCFG 寄存器锁定位

注意：

- [1] 该寄存器中的任何位，一旦置位，只能通过 CPU1.SYSRSn 清零。对该寄存器任何位写入 0 无效。
- [2] 锁定机制仅适用于写入。对具有 LOCK 保护的寄存器的读取始终是允许的。

表 2-151 CLKCFGLOCK1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | XTALCR |
| R-0-0h | | | | | | | R/WSonce-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LOSPCP | CLBCLKCTL | PERCLKDIVSEL | 保留 | SYSCLKDIVSEL | 保留 | 保留 | 保留 |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R-0-0h | R-0-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | SYSPLLMULT | 保留 | 保留 | SYSPLLCTL1 | CLKSRCCTL3 | CLKSRCCTL2 | CLKSRCCTL1 |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |

表 2-152 CLKCFGLOCK1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-17 | RESERVED | R-0 | 0h | 保留 |
| 16 | XTALCR | R/WSOnce | 0h | XTALCR 和 XTALCR2 寄存器的公共锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 15 | LOSPCP | R/WSOnce | 0h | LOSPCP 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 14 | CLBCLKCTL | R/WSOnce | 0h | CLBCLKCTL 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 13 | PERCLKDIVSEL | R/WSOnce | 0h | PERCLKDIVSEL 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 12 | RESERVED | R/WSOnce | 0h | 保留 |
| 11 | SYSCLKDIVSEL | R/WSOnce | 0h | SYSCLKDIVSEL 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 10 | RESERVED | R/WSOnce | 0h | 保留 |
| 9 | RESERVED | R-0 | 0h | 保留 |
| 8 | RESERVED | R-0 | 0h | 保留 |
| 7 | RESERVED | R/WSOnce | 0h | 保留 |
| 6 | SYSPLLMULT | R/WSOnce | 0h | SYSPLLMULT 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 5 | RESERVED | R/WSOnce | 0h | 保留 |
| 4 | RESERVED | R/WSOnce | 0h | 保留 |
| 3 | SYSPLLCTL1 | R/WSOnce | 0h | SYSPLLCTL1 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 2 | CLKSRCCTL3 | R/WSOnce | 0h | CLKSRCCTL3 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 1 | CLKSRCCTL2 | R/WSOnce | 0h | CLKSRCCTL2 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 0 | CLKSRCCTL1 | R/WSOnce | 0h | CLKSRCCTL1 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |

## 2.15.9.2 CLKSRCCTL1 寄存器

偏移地址：10h　复位值：00000000h

时钟源控制寄存器 1

该存储器映射寄存器要求在连续两次写入寄存器之间等待 39 个 SYSCLK 周期，否则第二次写入可能会丢失。可以通过添加 39 条 NOP 指令来实现此延迟。

表 2-153 CLKSRCCTL1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | WDHALTI | 保留 | INTOSC2OFF_NOTSUPPORTED | 保留 | OSCCLKSRCSEL | |
| R-0-0h | | R/W-0h | R/W-0h | R/W-0h | R-0-0h | R/W-0h | |

表 2-154 CLKSRCCTL1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-6 | RESERVED | R-0 | 0h | 保留 |
| 5 | WDHALTI | R/W | 0h | 看门狗 HALT 模式忽略位：此位决定 WD 在 HALT 模式下是否工作。0 = WD 在 HALT 模式下不工作。当系统进入 HALT 模式时，WD 的时钟被门控。此外，当系统进入 HALT 模式时，INTOSC1 和 INTOSC2 断电。1 = WD 在 HALT 模式下工作。当系统进入 HALT 模式时，WD 的时钟不被门控，INTOSC1/2 不断电。复位类型：XRSn |
| 4 | RESERVED | R/W | 0h | 保留 |
| 3 | INTOSC2OFF_NOTSUPPORTED | R/W | 0h | 保留：此位不再受支持，不应设置为 1。注意：如果将此位设置为 1，将关闭 INTOSC2 并导致 PLL 失效。复位类型：XRSn |
| 2 | RESERVED | R-0 | 0h | 保留 |
| 1-0 | OSCCLKSRCSEL | R/W | 0h | 振荡器时钟源选择位：此位选择 OSCCLK 的时钟源。00 = INTOSC2（复位时默认）01 = 外部振荡器（XTAL）10 = INTOSC1 11 = 保留（默认为 INTOSC1）。上电或 XRSn 复位后，默认选择 INTOSC2。每当用户使用这些位更改时钟源时，SYSPLLMULT 寄存器将被强制清零，PLL 将被旁路并断电。这可以防止潜在的 PLL 过冲。然后用户必须写入 SYSPLLMULT 寄存器以配置适当 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 的倍频器。注意：[1] 保留选择默认为 00 配置。[2] 建议仅在检测到时钟丢失后才使用 INTOSC1。如果用户希望在检测到时钟丢失后使用 INTOSC1（备用时钟源）重新锁定 PLL，可以执行 MCLKCLR 并锁定 PLL。[3] 对这些位的任何写入后必须至少等待 300 个 CPU 周期，方法为添加至少 300 条 NOP 指令。[4] 当 PLL 正在运行并被系统使用时（即 PLLCLKEN=1），更改 OSCCLKSRC 可能导致系统时钟死锁。用户需要首先通过 PLLCLKEN=0 将 PLL 时钟从系统旁路，然后再更改 OSCCLK 源。复位类型：XRSn |

### 2.15.9.3 CLKSRCCTL2 寄存器

偏移地址：14h　复位值：00000000h

时钟源控制寄存器 2

该存储器映射寄存器要求在连续两次写入寄存器之间等待 39 个 SYSCLK 周期，否则第二次写入可能会丢失。可以通过添加 39 条 NOP 指令来实现此延迟。

表 2-155 CLKSRCCTL2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | 保留 | |
| R-0-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | 保留 | | CANABCLKSEL | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 2-156 CLKSRCCTL2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-10 | RESERVED | R-0 | 0h | 保留 |
| 9-8 | RESERVED | R/W | 0h | 保留 |
| 7-6 | RESERVED | R/W | 0h | 保留 |
| 5-4 | RESERVED | R/W | 0h | 保留 |
| 3-2 | CANABCLKSEL | R/W | 0h | CANA 位时钟源选择位：00 = PERx.SYSCLK（复位时默认）01 = 外部振荡器（XTAL）10 = 保留 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 11 = 保留。时钟丢失检测电路对这些位没有影响。复位类型：XRSn |
| 1-0 | RESERVED | R/W | 0h | 保留 |

### 2.15.9.4 CLKSRCCTL3 寄存器

偏移地址：18h 复位值：00000000h

时钟源控制寄存器 3

该存储器映射寄存器要求在连续两次写入寄存器之间等待 39 个 SYSCLK 周期，否则第二次写入可能会丢失。可以通过添加 39 条 NOP 指令来实现此延迟。

表 2-157 CLKSRCCTL3 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | XCLKOUTSEL | | | |
| R-0-0h | | | | R/W-0h | | | |

表 2-158 CLKSRCCTL3 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留 |
| 3-0 | XCLKOUTSEL | R/W | 0h | XCLKOUT 源选择位：此位选择 XCLKOUT 的时钟源：0000 = PLLSYSCLK（复位时默认）0001 = PLLRAWCLK 0010 = SYSCLK 0101 = INTOSC1 0110 = INTOSC2 0111 = XTAL OSC 输出时钟 1001 = PUMPOSC 1010 = SYSAPLL.CLK_AUX 1100 = SYSPLL.CLKOUT 其余 = 保留。复位类型：SYSRSn |

### 2.15.9.5 SYSPLLCTL1 寄存器

偏移地址：1Ch 复位值：00000000h

SYSPLL 控制寄存器 1

该存储器映射寄存器要求在连续两次写入寄存器之间等待 39 个 SYSCLK 周期，否则第二次写入可能会丢失。可以通过添加 39 条 NOP 指令来实现此延迟。

表 2-159 SYSPLLCTL1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | 保留 | 保留 | 保留 | 保留 | PLLCLKEN | PLLEN |
| R-0-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-160 SYSPLLCTL1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-6 | RESERVED | R-0 | 0h | 保留 |
| 5 | RESERVED | R/W | 0h | 保留 |
| 4 | RESERVED | R/W | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | RESERVED | R/W | 0h | 保留 |
| 1 | PLLCLKEN | R/W | 0h | SYSPLL 旁路或包含在 PLLSYSCLK 路径中：此位决定生成 PLLSYSCLK 时是否旁路 SYSPLL。1 = PLLSYSCLK 由 SYSPLL 时钟输出提供。用户需要确保在将此时钟使能到系统之前 PLL 已锁定。0 = SYSPLL 被旁路。系统时钟直接来自 OSCCLK。注意：对该位的任何写入后必须至少等待 60 个 CPU 周期，方法为添加至少 120 条 NOP 指令。复位类型：XRSn |
| 0 | PLLEN | R/W | 0h | SYSPLL 使能或禁用：此位决定 SYSPLL 是否使能。1 = SYSPLL 使能。0 = SYSPLL 断电。系统时钟直接来自 OSCCLK。注意：对该位的任何写入后必须至少等待 60 个 CPU 周期，方法为添加至少 60 条 NOP 指令。复位类型：XRSn |

## 2.15.9.6 SYSPLLMULT 寄存器

偏移地址：28h　复位值：00000000h

SYSPLL 倍频器寄存器

注意：FMULT 和 IMULT 字段必须同时写入，以确保 PLL 正确运行。

该存储器映射寄存器要求在连续两次写入寄存器之间等待 39 个 SYSCLK 周期，否则第二次写入可能会丢失。可以通过添加 39 条 NOP 指令来实现此延迟。

表 2-161 SYSPLLMULT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | REFDIV | | | | |
| R-0-0h | | | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | ODIV | | | | |
| R-0-0h | | | R/W-0h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | 保留 | | 保留 | | 保留 | |
| R-0-0h | | R/W-0h | | R-0-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IMULT | | | | | | | |
| R/W-0h | | | | | | | |

表 2-162 SYSPLLMULT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-29 | RESERVED | R-0 | 0h | 保留 |
| 28-24 | REFDIV | R/W | 0h | SYSPLL 参考时钟分频器。PLL 参考分频器 = REFDIV + 1。复位类型：XRSn |
| 23-21 | RESERVED | R-0 | 0h | 保留 |
| 20-16 | ODIV | R/W | 0h | SYSPLL 输出时钟分频器。PLL 输出分频器 = ODIV + 1。ODIV 应设置为 1 或更大，以确保 PLL 输出满足占空比要求。复位类型：XRSn |
| 15-14 | RESERVED | R-0 | 0h | 保留 |
| 13-12 | RESERVED | R/W | 0h | 保留 |
| 11-10 | RESERVED | R-0 | 0h | 保留 |
| 9-8 | RESERVED | R/W | 0h | 保留 |
| 7-0 | IMULT | R/W | 0h | SYSPLL 整数倍频器：0000000 时，Fout = Fref（PLLBYPASS），整数倍频器 = 1；0000001 整数倍频器 = 1；0000010 整数倍频器 = 2；0000011 整数倍频器 = 3；……. 1111111 整数倍频器 = 127。注意：对于 APLL，倍频器值 0-3 无效，内部会将它们视为 4。复位类型：XRSn |

### 2.15.9.7 SYSPLLSTS 寄存器

偏移地址：2Ch　复位值：00000030h

SYSPLL 状态寄存器

表 2-163 SYSPLLSTS 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | 保留 | 保留 | REF_LOSTS | 保留 | SLIPS_NOTSUPPORTED | LOCKS |
| R-0-0h | | R-1h | R-1h | W1C-0h | R-0h | R-0h | R-0h |

表 2-164 SYSPLLSTS 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-6 | RESERVED | R-0 | 0h | 保留 |
| 5 | RESERVED | R | 1h | 保留 |
| 4 | RESERVED | R | 1h | 保留 |
| 3 | REF_LOSTS | W1C | 0h | SYSPLL“参考时钟丢失”状态位：此位指示 SYSPLL 是否超出锁定范围。0 = 未发生“参考时钟丢失”事件。1 = 已发生“参考时钟丢失”事件。复位类型：XRSn |
| 2 | RESERVED | R | 0h | 保留 |
| 1 | SLIPS_NOTSUPPORTED | R | 0h | 保留：此位保留，读取的值应忽略。建议使用 DCC 来评估 SYSPLL 滑差状态。有关使用 DCC 检查 SYSPLL 滑差状态，请参阅 Ware 中最新示例软件中的 InitSysPll() 或 SysCtl_setClock() 函数。复位类型：XRSn |
| 0 | LOCKS | R | 0h | SYSPLL 锁定状态位：此位指示 SYSPLL 是否锁定。0 = SYSPLL 尚未锁定。1 = SYSPLL 已锁定。复位类型：XRSn |

### 2.15.9.8 SYSCLKDIVSEL 寄存器

偏移地址：44h　复位值：00000000h

系统时钟分频器选择寄存器

该存储器映射寄存器要求在连续两次写入寄存器之间等待 39 个 SYSCLK 周期，否则第二次写入可能会丢失。可以通过添加 39 条 NOP 指令来实现此延迟。

表 2-165 SYSCLKDIVSEL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | PLLSYSCLKDIV_LSB |
| R-0-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | PLLSYSCLKDIV | | | | | |
| R-0-0h | | R/W-0h | | | | | |

表 2-166 SYSCLKDIVSEL 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-9 | RESERVED | R-0 | 0h | 保留 |
| 8 | PLLSYSCLKDIV_LSB | R/W | 0h | 此位是分频器的 LSB，置位时允许奇数分频，分频值为 {PLLSYSCLKDIV，PLLSYSCLKDIV_LSB}。例如，如果 PLLSYSCLKDIV=0x1，且 PLLSYSCLKDIV_LSB=0，则使用 2 分频；如果 PLLSYSCLKDIV_LSB=1，则分频值为 3。复位类型：XRSn |
| 7-6 | RESERVED | R-0 | 0h | 保留 |
| 5-0 | PLLSYSCLKDIV | R/W | 0h | PLLSYSCLK 分频选择：此位选择 PLLSYSCLK 的分频器设置。000000 = /1 000001 = /2 000010 = /4（复位时默认）000011 = /6 000100 = /8 …… 111111 = /126。复位类型：XRSn |

## 2.15.9.9 XCLKOUTDIVSEL 寄存器

偏移地址：50h　复位值：00000003h

XCLKOUT 分频器选择寄存器

该存储器映射寄存器要求在连续两次写入寄存器之间等待 39 个 SYSCLK 周期，否则第二次写入可能会丢失。可以通过添加 39 条 NOP 指令来实现此延迟。

表 2-167 XCLKOUTDIVSEL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | XCLKOUTDIV | |
| R-0-0h | | | | | | R/W-3h | |

表 2-168 XCLKOUTDIVSEL 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1-0 | XCLKOUTDIV | R/W | 3h | XCLKOUT 分频选择：此位选择 XCLKOUT 的分频器设置。00 = /1 01 = /2 10 = /4 11 = /8（复位时默认）。复位类型：SYSRSn |

## 2.15.9.10 LOSPCP 寄存器

偏移地址：58h　复位值：00000002h

低速时钟源预分频器

表 2-169 LOSPCP 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | LSPCLKDIV | | |
| R-0-0h | | | | | R/W-2h | | |

表 2-170 LOSPCP 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-3 | RESERVED | R-0 | 0h | 保留 |
| 2-0 | LSPCLKDIV | R/W | 2h | 这些位配置低速外设时钟（LSPCLK）速率：000，LSPCLK = /1；001，LSPCLK = /2；010，LSPCLK = /4（复位时默认）；011，LSPCLK = /6；100，LSPCLK = /8；101，LSPCLK = /10；110，LSPCLK = /12；111，LSPCLK = /14。注意：[1] 此时钟用作 SCI 和 SPI 模块的选通信号。复位类型：SYSRSn |

## 2.15.9.11 MCDCR 寄存器

偏移地址：5Ch　复位值：00000000h

时钟丢失检测控制寄存器

表 2-171 MCDCR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | 保留 | 保留 |
| R-0-0h | | | | | | R/W-0h | R-0/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | 保留 | 保留 | OSCOFF | MCLKOFF | MCLKCLR | MCLKSTS |
| R-0h | R-0h | R-0h | R-0h | R/W-0h | R/W-0h | R-0/W1S-0h | R-0h |

表 2-172 MCDCR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-10 | RESERVED | R-0 | 0h | 保留 |
| 9 | RESERVED | R/W | 0h | 保留 |
| 8 | RESERVED | R-0/W1S | 0h | 保留 |
| 7 | RESERVED | R | 0h | 保留 |
| 6 | RESERVED | R | 0h | 保留 |
| 5 | RESERVED | R | 0h | 保留 |
| 4 | RESERVED | R | 0h | 保留 |
| 3 | OSCOFF | R/W | 0h | 振荡器时钟与 MCD 断开位：0 = OSCCLK 连接到 MCD 模块中的 OSCCLK 计数器。1 = OSCCLK 与 MCD 模块中的 OSCCLK 计数器断开。复位类型：XRSn |
| 2 | MCLKOFF | R/W | 0h | 时钟丢失检测关闭位：0 = 时钟丢失检测电路使能。1 = 时钟丢失检测电路禁用。复位类型：XRSn |
| 1 | MCLKCLR | R-0/W1S | 0h | 时钟丢失清零位：向此位写入“1”以清零 MCLKSTS 位并复位时钟丢失检测电路。复位类型：XRSn |
| 0 | MCLKSTS | R | 0h | 时钟丢失状态位：0 = OSCCLK 正常。1 = 检测到 OSCCLK 丢失，已生成 CLOCKFAILn。复位类型：XRSn |

## 2.15.9.12 X1CNT 寄存器

偏移地址：60h　复位值：00000000h

X1 时钟上的 10 位计数器

表 2-173 X1CNT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | CLR |
| R-0-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | X1CNT | | |
| R-0-0h | | | | | R-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| X1CNT | | | | | | | |
| R-0h | | | | | | | |

表 2-174 X1CNT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-17 | RESERVED | R-0 | 0h | 保留 |
| 16 | CLR | R/W | 0h | X1 计数器清零：向该字段写入“1”会清零 X1CNT，并使其从 0x0 重新开始计数（前提是 X1 时钟在运行）。向该字段写入“0”被忽略。复位类型：XRSn |
| 15-11 | RESERVED | R-0 | 0h | 保留 |
| 10-0 | X1CNT | R | 0h | X1 计数器：- 该计数器在 X1 时钟的每个上升沿递增。- 一旦达到 0x7ff 的值，它就会冻结。- 在从 INTOSC2 切换到 X1 之前，应用程序必须检查该计数器并确保其已饱和。这将确保连接到 X1/X2 的晶体正在振荡。复位类型：XRSn |

## 2.15.9.13 CLKFAILCFG 寄存器

偏移地址：78h　复位值：00000000h

时钟失效原因配置

表 2-175 CLKFAILCFG 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | DCC1_ERROR_EN | DCC0_ERROR_EN |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-176 CLKFAILCFG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | DCC1_ERROR_EN | R/W | 0h | 此字段使能 DCC1 错误以触发时钟失效 NMI 断言。0：DCC1 错误不影响时钟失效 NMI。1：发生 DCC1 错误时触发时钟失效 NMI 断言和 ERROR 引脚断言。复位类型：XRSn |
| 0 | DCC0_ERROR_EN | R/W | 0h | 此字段使能 DCC0 错误以触发时钟失效 NMI 断言。0：DCC0 错误不影响时钟失效 NMI。1：发生 DCC0 错误时触发时钟失效 NMI 断言和 ERROR 引脚断言。复位类型：XRSn |

## 2.15.10 CPU_SYS_REGS 寄存器

表中列出了 CPU_SYS_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址应视为保留位置，不应修改寄存器内容。

表 2-177 CPU_SYS_REGS 寄存器

| 偏移 | 简称 | 寄存器名称 |
|---|---|---|
| 0h | CPUSYSLOCK1 | CPUSYS 寄存器锁定位 |
| 4h | CPUSYSLOCK2 | CPUSYS 寄存器锁定位 |
| 44h | PCLKCR0 | 外设时钟门控寄存器 |
| 4Ch | PCLKCR2 | 外设时钟门控寄存器 - ETPWM |
| 50h | PCLKCR3 | 外设时钟门控寄存器 - ECAP |
| 54h | PCLKCR4 | 外设时钟门控寄存器 - EQEP |
| 60h | PCLKCR7 | 外设时钟门控寄存器 - SCI |
| 64h | PCLKCR8 | 外设时钟门控寄存器 - SPI |
| 68h | PCLKCR9 | 外设时钟门控寄存器 - I2C |

| 偏移 | 简称 | 寄存器名称 |
|---|---|---|
| 6Ch | PCLKCR10 | 外设时钟门控寄存器 - CAN |
| 78h | PCLKCR13 | 外设时钟门控寄存器 - ADC |
| 7Ch | PCLKCR14 | 外设时钟门控寄存器 - CMPSS |
| 88h | PCLKCR17 | 外设时钟门控寄存器 - CLB |
| 8Ch | PCLKCR18 | 外设时钟门控寄存器 - FSI |
| 90h | PCLKCR19 | 外设时钟门控寄存器 - LIN |
| 94h | PCLKCR20 | 外设时钟门控寄存器 - PMBUS |
| 98h | PCLKCR21 | 外设时钟门控寄存器 - DCC |
| 9Ch | PCLKCR22 | 外设时钟门控寄存器 - MPOST |
| A8h | PCLKCR25 | 外设时钟门控寄存器 - HIC |
| E0h | SIMRESET | 模拟复位寄存器 |
| ECh | LPMCR | LPM 控制寄存器 |
| F0h | GPIOLPMSEL0 | GPIO LPM 唤醒选择寄存器 |
| F4h | GPIOLPMSEL1 | GPIO LPM 唤醒选择寄存器 |
| F8h | TMR2CLKCTL | 定时器 2 时钟测量功能控制寄存器 |
| FCh | RESCCLR | 复位原因清零寄存器 |
| 100h | RESC | 复位原因寄存器 |

复杂位访问类型编码为适合小表格单元格。表中显示了本节中使用的访问类型代码。

表 2-178 CPU_SYS_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读类型 | 读类型 | 读类型 |
| R | R | 读取 |
| R-0 | R -0 | 读取返回 0s |
| 写类型 | 写类型 | 写类型 |
| W | W | 写入 |
| W1C | W 1C | 写 1 清零 |
| W1S | W 1S | 写 1 置位 |
| WSonce | W Sonce | 写入设置一次 |

## 2.15.10.1 CPUSYSLOCK1 寄存器

偏移地址：0h 复位值：00000000h

CPUSYS 寄存器锁定位

注意：

- [1] 该寄存器中的任何位，一旦置位，只能通过 CPU1.SYSRSn 清零。对该寄存器任何位写入 0 无效。
- [2] 锁定机制仅适用于写入。对具有 LOCK 保护的寄存器的读取始终是允许的。

表 2-179 CPUSYSLOCK1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | PCLKCR0 | SIMRESET | LPMCR | GPIOLPMSEL1 |
| R-0-0h | | | | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIOLPMSEL0 | TMR2CLKCTL | RESCCLR | RESC | 保留 | | | |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R-0-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |

表 2-180 CPUSYSLOCK1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-20 | RESERVED | R-0 | 0h | 保留 |
| 19 | PCLKCR0 | R/WSonce | 0h | PCLKCR0 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 18 | SIMRESET | R/WSonce | 0h | SIMRESET 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 17 | LPMCR | R/WSonce | 0h | LPMCR 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 16 | GPIOLPMSEL1 | R/WSonce | 0h | GPIOLPMSEL1 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 15 | GPIOLPMSEL0 | R/WSonce | 0h | GPIOLPMSEL0 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 14 | TMR2CLKCTL | R/WSonce | 0h | TMR2CLKCTL 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 13 | RESCCLR | R/WSonce | 0h | RESCCLR 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 12 | RESC | R/WSonce | 0h | RESC 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 11-0 | RESERVED | R-0 | 0h | 保留 |

## 2.15.10.2 CPUSYSLOCK2 寄存器

偏移地址：4h　复位值：00000000h

CPUSYS 寄存器锁定位

注意：

- [1] 该寄存器中的任何位，一旦置位，只能通过 CPU1.SYSRSn 清零。对该寄存器任何位写入 0 无效。
- [2] 锁定机制仅适用于写入。对具有 LOCK 保护的寄存器的读取始终是允许的。

表 2-181 CPUSYSLOCK2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | PCLKCR22 | PCLKCR21 | PCLKCR20 | PCLKCR19 | PCLKCR18 | PCLKCR17 |
| R-0-0h | | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | PCLKCR14 | PCLKCR13 | 保留 | PCLKCR10 | PCLKCR9 | PCLKCR8 | PCLKCR7 |
| R-0-0h | R/WSonce-0h | R/WSonce-0h | R-0-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | PCLKCR4 | PCLKCR3 | PCLKCR2 | 保留 | | |
| R-0-0h | | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R-0-0h | | |

表 2-182 CPUSYSLOCK2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-22 | RESERVED | R-0 | 0h | 保留 |
| 21 | PCLKCR22 | R/WSonce | 0h | PCLKCR22 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 20 | PCLKCR21 | R/WSonce | 0h | PCLKCR21 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 19 | PCLKCR20 | R/WSonce | 0h | PCLKCR20 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 18 | PCLKCR19 | R/WSonce | 0h | PCLKCR19 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 17 | PCLKCR18 | R/WSonce | 0h | PCLKCR18 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 16 | PCLKCR17 | R/WSonce | 0h | PCLKCR17 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 15 | RESERVED | R-0 | 0h | 保留 |
| 14 | PCLKCR14 | R/WSonce | 0h | PCLKCR14 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 13 | PCLKCR13 | R/WSonce | 0h | PCLKCR13 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 12 | RESERVED | R-0 | 0h | 保留 |
| 11 | PCLKCR10 | R/WSonce | 0h | PCLKCR10 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 10 | PCLKCR9 | R/WSonce | 0h | PCLKCR9 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 9 | PCLKCR8 | R/WSonce | 0h | PCLKCR8 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 8 | PCLKCR7 | R/WSonce | 0h | PCLKCR7 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 7-6 | RESERVED | R-0 | 0h | 保留 |
| 5 | PCLKCR4 | R/WSonce | 0h | PCLKCR4 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 4 | PCLKCR3 | R/WSonce | 0h | PCLKCR3 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 3 | PCLKCR2 | R/WSonce | 0h | PCLKCR2 寄存器锁定位：0：相应寄存器未锁定。1：相应寄存器已锁定。复位类型：SYSRSn |
| 2-0 | RESERVED | R-0 | 0h | 保留 |

### 2.15.10.3 PCLKCR0 寄存器

偏移地址：44h　复位值：00000000h

外设时钟门控寄存器

表 2-183 PCLKCR0 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | CLA1 |
| R-0-0h | | | | | | | R/W-0h |

表 2-184 PCLKCR0 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | CLA1 | R/W | 0h | 0：CLA1 时钟关闭。1：CLA1 时钟开启。复位类型：SYSRSn |

### 2.15.10.4 PCLKCR2 寄存器

偏移地址：4Ch　复位值：00000000h

外设时钟门控寄存器 - ETPWM

表 2-185 PCLKCR2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | EPWM7 | EPWM6 | EPWM5 | EPWM4 | EPWM3 | EPWM2 | EPWM1 |
| R-0-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-186 PCLKCR2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-7 | RESERVED | R-0 | 0h | 保留 |
| 6 | EPWM7 | R/W | 0h | 0：EPWM7 时钟关闭。1：EPWM7 时钟开启。复位类型：SYSRSn |
| 5 | EPWM6 | R/W | 0h | 0：EPWM6 时钟关闭。1：EPWM6 时钟开启。复位类型：SYSRSn |
| 4 | EPWM5 | R/W | 0h | 0：EPWM5 时钟关闭。1：EPWM5 时钟开启。复位类型：SYSRSn |
| 3 | EPWM4 | R/W | 0h | 0：EPWM4 时钟关闭。1：EPWM4 时钟开启。复位类型：SYSRSn |
| 2 | EPWM3 | R/W | 0h | 0：EPWM3 时钟关闭。1：EPWM3 时钟开启。复位类型：SYSRSn |
| 1 | EPWM2 | R/W | 0h | 0：EPWM2 时钟关闭。1：EPWM2 时钟开启。复位类型：SYSRSn |
| 0 | EPWM1 | R/W | 0h | 0：EPWM1 时钟关闭。1：EPWM1 时钟开启。复位类型：SYSRSn |

### 2.15.10.5 PCLKCR3 寄存器

偏移地址：50h　复位值：00000000h

外设时钟门控寄存器 - ECAP

表 2-187 PCLKCR3 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | ECAP3 | ECAP2 | ECAP1 |
| R-0-0h | | | | | R/W-0h | R/W-0h | R/W-0h |

表 2-188 PCLKCR3 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-3 | RESERVED | R-0 | 0h | 保留 |
| 2 | ECAP3 | R/W | 0h | 0：ECAP3 时钟关闭。1：ECAP3 时钟开启。复位类型：SYSRSn |
| 1 | ECAP2 | R/W | 0h | 0：ECAP2 时钟关闭。1：ECAP2 时钟开启。复位类型：SYSRSn |
| 0 | ECAP1 | R/W | 0h | 0：ECAP1 时钟关闭。1：ECAP1 时钟开启。复位类型：SYSRSn |

### 2.15.10.6 PCLKCR4 寄存器

偏移地址：54h　复位值：00000000h

外设时钟门控寄存器 - EQEP

表 2-189 PCLKCR4 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | EQEP2 | EQEP1 |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-190 PCLKCR4 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | EQEP2 | R/W | 0h | 0：EQEP2 时钟关闭。1：EQEP2 时钟开启。复位类型：SYSRSn |
| 0 | EQEP1 | R/W | 0h | 0：EQEP1 时钟关闭。1：EQEP1 时钟开启。复位类型：SYSRSn |

## 2.15.10.7 PCLKCR7 寄存器

偏移地址：60h　复位值：00000000h

外设时钟门控寄存器 - SCI

表 2-191 PCLKCR7 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | SCI_A |
| R-0-0h | | | | | | | R/W-0h |

表 2-192 PCLKCR7 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | SCI_A | R/W | 0h | 0：SCI_A 时钟关闭。1：SCI_A 时钟开启。复位类型：SYSRSn |

## 2.15.10.8 PCLKCR8 寄存器

偏移地址：64h　复位值：00000000h

外设时钟门控寄存器 - SPI

表 2-193 PCLKCR8 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | SPI_B | SPI_A |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-194 PCLKCR8 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | SPI_B | R/W | 0h | 0：SPI_B 时钟关闭。1：SPI_B 时钟开启。复位类型：SYSRSn |
| 0 | SPI_A | R/W | 0h | 0：SPI_A 时钟关闭。1：SPI_A 时钟开启。复位类型：SYSRSn |

## 2.15.10.9 PCLKCR9 寄存器

偏移地址：68h　复位值：00000000h

外设时钟门控寄存器 - I2C

表 2-195 PCLKCR9 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | I2C_B | I2C_A |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-196 PCLKCR9 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | I2C_B | R/W | 0h | 0：I2C_B 时钟关闭。1：I2C_B 时钟开启。复位类型：SYSRSn |
| 0 | I2C_A | R/W | 0h | 0：I2C_A 时钟关闭。1：I2C_A 时钟开启。复位类型：SYSRSn |

## 2.15.10.10 PCLKCR10 寄存器

偏移地址：6Ch　复位值：00000000h

外设时钟门控寄存器 - CAN

表 2-197 PCLKCR10 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | CAN_A |
| R-0-0h | | | | | | | R/W-0h |

表 2-198 PCLKCR10 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | CAN_A | R/W | 0h | 0：CAN_A 时钟关闭。1：CAN_A 时钟开启。复位类型：SYSRSn |

## 2.15.10.11 PCLKCR13 寄存器

偏移地址：78h　复位值：00000000h

外设时钟门控寄存器 - ADC

表 2-199 PCLKCR13 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | ADC_C | 保留 | ADC_A |
| R-0-0h | | | | | R/W-0h | R-0h | R/W-0h |

表 2-200 PCLKCR13 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-3 | RESERVED | R-0 | 0h | 保留 |
| 2 | ADC_C | R/W | 0h | 0：ADC_C 时钟关闭。1：ADC_C 时钟开启。复位类型：SYSRSn |
| 1 | RESERVED | R | 0h | 保留 |
| 0 | ADC_A | R/W | 0h | 0：ADC_A 时钟关闭。1：ADC_A 时钟开启。复位类型：SYSRSn |

## 2.15.10.12 PCLKCR14 寄存器

偏移地址：7Ch　复位值：00000000h

外设时钟门控寄存器 - CMPSS

表 2-201 PCLKCR14 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | CMPSS4 | CMPSS3 | CMPSS2 | CMPSS1 |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 2-202 PCLKCR14 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留 |
| 3 | CMPSS4 | R/W | 0h | 0：CMPSS4 时钟关闭。1：CMPSS4 时钟开启。<br>复位类型：SYSRSn |
| 2 | CMPSS3 | R/W | 0h | 0：CMPSS3 时钟关闭。1：CMPSS3 时钟开启。<br>复位类型：SYSRSn |
| 1 | CMPSS2 | R/W | 0h | 0：CMPSS2 时钟关闭。1：CMPSS2 时钟开启。<br>复位类型：SYSRSn |
| 0 | CMPSS1 | R/W | 0h | 0：CMPSS1 时钟关闭。1：CMPSS1 时钟开启。<br>复位类型：SYSRSn |

## 2.15.10.13 PCLKCR17 寄存器

偏移地址：88h　复位值：00000000h

外设时钟门控寄存器 - CLB

表 2-203 PCLKCR17 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | CLB2 | CLB1 |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-204 PCLKCR17 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | CLB2 | R/W | 0h | 0：CLB2 时钟关闭。1：CLB2 时钟开启。复位类型：SYSRSn |
| 0 | CLB1 | R/W | 0h | 0：CLB1 时钟关闭。1：CLB1 时钟开启。复位类型：SYSRSn |

## 2.15.10.14 PCLKCR18 寄存器

偏移地址：8Ch　复位值：00000000h

外设时钟门控寄存器 - FSI

表 2-205 PCLKCR18 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | FSIRX_A | FSITX_A |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-206 PCLKCR18 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | FSIRX_A | R/W | 0h | 0：FSIRX_A 时钟关闭。1：FSIRX_A 时钟开启。<br>复位类型：SYSRSn |
| 0 | FSITX_A | R/W | 0h | 0：FSITX_A 时钟关闭。1：FSITX_A 时钟开启。<br>复位类型：SYSRSn |

## 2.15.10.15 PCLKCR19 寄存器

偏移地址：90h　复位值：00000000h

外设时钟门控寄存器 - LIN

表 2-207 PCLKCR19 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | LIN_B | LIN_A |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-208 PCLKCR19 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | LIN_B | R/W | 0h | 0：LIN_B 时钟关闭。1：LIN_B 时钟开启。复位类型：SYSRSn |
| 0 | LIN_A | R/W | 0h | 0：LIN_A 时钟关闭。1：LIN_A 时钟开启。复位类型：SYSRSn |

## 2.15.10.16 PCLKCR20 寄存器

偏移地址：94h　复位值：00000000h

外设时钟门控寄存器 - PMBUS

表 2-209 PCLKCR20 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | PMBUS_A |
| R-0-0h | | | | | | | R/W-0h |

表 2-210 PCLKCR20 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | PMBUS_A | R/W | 0h | 0：PMBUS_A 时钟关闭。1：PMBUS_A 时钟开启。复位类型：SYSRSn |

## 2.15.10.17 PCLKCR21 寄存器

偏移地址：98h　复位值：00000000h

外设时钟门控寄存器 - DCC

表 2-211 PCLKCR21 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | DCC1 | DCC0 |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-212 PCLKCR21 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | DCC1 | R/W | 0h | 0：DCC1 时钟关闭。1：DCC1 时钟开启。复位类型：SYSRSn |
| 0 | DCC0 | R/W | 0h | 0：DCC0 时钟关闭。1：DCC0 时钟开启。复位类型：SYSRSn |

## 2.15.10.18 PCLKCR22 寄存器

偏移地址：9Ch　复位值：00000000h

外设时钟门控寄存器 - MPOST

表 2-213 PCLKCR22 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | MPOST_A |
| R-0-0h | | | | | | | R/W-0h |

表 2-214 PCLKCR22 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | MPOST_A | R/W | 0h | 0：MPOST_A 时钟关闭。1：MPOST_A 时钟开启。复位类型：SYSRSn |

## 2.15.10.19 PCLKCR25 寄存器

偏移地址：A8h　复位值：00000000h

外设时钟门控寄存器 - HIC

表 2-215 PCLKCR25 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | HIC_A |
| R-0-0h | | | | | | | R/W-0h |

表 2-216 PCLKCR25 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | HIC_A | R/W | 0h | 0：HIC_A 时钟关闭。1：HIC_A 时钟开启。复位类型：SYSRSn |

## 2.15.10.20 SIMRESET 寄存器

偏移地址：E0h　复位值：00000000h

模拟复位寄存器

表 2-217 SIMRESET 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | XRSn | CPU1RSn |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-218 SIMRESET 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | XRSn | R/W | 0h | 模拟 XRSn 复位：向此位写入 1 会触发 XRSn 复位。此位为自清零。复位类型：SYSRSn |
| 0 | CPU1RSn | R/W | 0h | 模拟 CPU1 复位：向此位写入 1 会触发 CPU1 复位。此位为自清零。复位类型：SYSRSn |

## 2.15.10.21 LPMCR 寄存器

偏移地址：ECh 复位值：00000000h

LPM 控制寄存器

表 2-219 LPMCR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | LPM | |
| R-0-0h | | | | | | R/W-0h | |

表 2-220 LPMCR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1-0 | LPM | R/W | 0h | LPM 模式选择：00 = 正常运行。01 = 睡眠模式。10 = 深度睡眠模式。11 = 保留。复位类型：SYSRSn |

## 2.15.10.22 GPIOLPMSEL0 寄存器

偏移地址：F0h 复位值：00000000h

GPIO LPM 唤醒选择寄存器

表 2-221 GPIOLPMSEL0 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIOx | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIOx | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIOx | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIOx | | | | | | | |
| R/W-0h | | | | | | | |

表 2-222 GPIOLPMSEL0 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | GPIOx | R/W | 0h | GPIOx LPM 唤醒选择：0 = GPIOx 不唤醒 LPM。<br>1 = GPIOx 唤醒 LPM。复位类型：SYSRSn |

## 2.15.10.23 GPIOLPMSEL1 寄存器

偏移地址：F4h　复位值：00000000h

GPIO LPM 唤醒选择寄存器

表 2-223 GPIOLPMSEL1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIOx | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIOx | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIOx | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIOx | | | | | | | |
| R/W-0h | | | | | | | |

表 2-224 GPIOLPMSEL1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | GPIOx | R/W | 0h | GPIOx LPM 唤醒选择：0 = GPIOx 不唤醒 LPM。<br>1 = GPIOx 唤醒 LPM。复位类型：SYSRSn |

## 2.15.10.24 TMR2CLKCTL 寄存器

偏移地址：F8h　复位值：00000000h

定时器 2 时钟测量功能控制寄存器

表 2-225 TMR2CLKCTL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | CLKSRC | CLKMEAS |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 2-226 TMR2CLKCTL 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | CLKSRC | R/W | 0h | 定时器 2 时钟测量时钟源：0 = SYSCLK。1 = X1 时钟。复位类型：SYSRSn |
| 0 | CLKMEAS | R/W | 0h | 定时器 2 时钟测量使能：0 = 时钟测量禁用。1 = 时钟测量使能。复位类型：SYSRSn |

## 2.15.10.25 RESCCLR 寄存器

偏移地址：FCh　复位值：00000000h

复位原因清零寄存器

表 2-227 RESCCLR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | SIMRESET_XRSn | SIMRESET_CPU1RSn | 保留 | |
| R-0-0h | | | | W1C-0h | W1C-0h | R-0-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | NMIWDRSn | WDRSn | XRSn | POR |
| R-0-0h | | | | W1S-0h | W1S-0h | W1S-0h | W1S-0h |

表 2-228 RESCCLR 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-12 | RESERVED | R-0 | 0h | 保留。 |
| 11 | SIMRESET_XRSn | W1C | 0h | 清除 RESC 中对应状态位的控制位。读取 RESCCLR 始终返回 0。写入 1 将 RESC 中对应状态位清零，写入 0 无影响。<br>复位类型：SYSRSn。 |
| 10 | SIMRESET_CPU1RSn | W1C | 0h | 清除 RESC 中对应状态位的控制位。读取 RESCCLR 始终返回 0。写入 1 将 RESC 中对应状态位清零，写入 0 无影响。<br>复位类型：SYSRSn。 |
| 9 | RESERVED | R-0 | 0h | 保留。 |
| 8 | RESERVED | R-0 | 0h | 保留。 |
| 7 | RESERVED | R-0 | 0h | 保留。 |
| 6 | RESERVED | R-0 | 0h | 保留。 |
| 5 | RESERVED | R-0 | 0h | 保留。 |
| 4 | RESERVED | R-0 | 0h | 保留。 |
| 3 | NMIWDRSn | W1S | 0h | 清除 RESC 中对应状态位的控制位。读取 RESCCLR 始终返回 0。写入 1 将 RESC 中对应状态位清零，写入 0 无影响。<br>复位类型：SYSRSn。 |
| 2 | WDRSn | W1S | 0h | 清除 RESC 中对应状态位的控制位。读取 RESCCLR 始终返回 0。写入 1 将 RESC 中对 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 应状态位清零，写入 0 无影响。<br>复位类型：SYSRSn。 |
| 1 | XRSn | W1S | 0h | 清除 RESC 中对应状态位的控制位。读取 RESCCLR 始终返回 0。写入 1 将 RESC 中对应状态位清零，写入 0 无影响。<br>复位类型：SYSRSn。 |
| 0 | POR | W1S | 0h | 清除 RESC 中对应状态位的控制位。读取 RESCCLR 始终返回 0。写入 1 将 RESC 中对应状态位清零，写入 0 无影响。<br>复位类型：SYSRSn。 |

## 2.15.10.26 RESC 寄存器

偏移地址：100h　复位值：00000001h

复位原因寄存器

表 2-229 RESC 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DCON | XRSn_pin_status | 保留 | | | | | |
| R-0h | R-X | R-0-0h | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | SIMRESET_XRSn | SIMRESET_CPU1RSn | 保留 | |
| R-0-0h | | | | R-0h | R-0h | R-0-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | NMIWDRSn | WDRSn | XRSn | POR |
| R-0-0h | | | | R-0h | R-0h | R-1h | R-1h |

表 2-230 RESC 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | DCON | R | 0h | 读取此位可获得调试器与 CPU CPU 的连接状态。<br>0：调试器未连接至 CPU CPU。<br>1：调试器已连接至 CPU CPU。<br>注意：[1] 此位连接至 CPU CPU 的 DCON 输出信号。<br>复位类型：N/A。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 30 | XRSn_pin_status | R | X | 读取此位可获得 XRSn 引脚的当前状态。复位值反映引脚状态。<br>复位类型：N/A。 |
| 29-16 | RESERVED | R-0 | 0h | 保留。 |
| 15-12 | RESERVED | R-0 | 0h | 保留。 |
| 11 | SIMRESET_XRSn | R | 0h | 此位置位表示器件由 SIMRESET_XRSn 复位。<br>复位类型：PORESETn。 |
| 10 | SIMRESET_CPU1RSn | R | 0h | 此位置位表示器件由 SIMRESET_CPU1RSn 复位。<br>复位类型：PORESETn。 |
| 9 | RESERVED | R-0 | 0h | 保留。 |
| 8 | RESERVED | R-0 | 0h | 保留。 |
| 7 | RESERVED | R-0 | 0h | 保留。 |
| 6 | RESERVED | R-0 | 0h | 保留。 |
| 5 | RESERVED | R-0 | 0h | 保留。 |
| 4 | RESERVED | R-0 | 0h | 保留。 |
| 3 | NMIWDRSn | R | 0h | 此位置位表示器件由 NMIWDRSn 复位。<br>注意：要确定复位后 NMI 的确切原因，软件需要读取 NMISHDFLG 寄存器。<br>复位类型：PORESETn。 |
| 2 | WDRSn | R | 0h | 此位置位表示器件由 WDRSn 复位。<br>注意：[1] 看门狗模块内也有一位提供相同信息，此位用于保持一致性。软件可通过本寄存器集中查询 CPU 内核的复位原因。<br>复位类型：PORESETn。 |
| 1 | XRSn | R | 1h | 此位置位表示器件由 XRSn 复位。<br>复位类型：PORESETn。 |
| 0 | POR | R | 1h | 此位置位表示器件由 PORn/BORn 复位。<br>复位类型：PORESETn。 |

## 2.15.11 DCSM_BANK0_Z1_REGS 寄存器

表中列出了 DCSM_BANK0_Z1_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址应视为保留位置，不应修改寄存器内容。

表 2-231 DCSM_BANK0_Z1_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | B0_Z1_LINKPOINTER | Zone 1 链接指针（Flash BANK0） |
| 4h | Z1_OTPSECLOCK | Zone 1 OTP 安全锁定 |

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 8h | Z1_BOOTDEF_HIGH | 启动定义（高 32 位） |
| Ch | B0_Z1_LINKPOINTERERR | Flash BANK0 链接指针错误 |
| 10h | Z1_BOOTPIN_CONFIG | 启动引脚配置 |
| 14h | Z1_GPREG2 | Zone1 通用寄存器 2 |
| 18h | Z1_BOOTDEF_LOW | 启动定义（低 32 位） |
| 20h | Z1_CSMKEY0 | Zone 1 CSM 密钥 0 |
| 24h | Z1_CSMKEY1 | Zone 1 CSM 密钥 1 |
| 28h | Z1_CSMKEY2 | Zone 1 CSM 密钥 2 |
| 2Ch | Z1_CSMKEY3 | Zone 1 CSM 密钥 3 |
| 32h | Z1_CR | Zone 1 CSM 控制寄存器 |
| 34h | B0_Z1_GRABSECTR | Zone 1 获取 Flash BANK0 扇区寄存器 |
| 38h | Z1_GRABRAMR | Zone 1 获取 RAM 块寄存器 |

复杂的位访问类型经过编码以适应小型表格单元。表中显示了本节中使用的访问类型代码。

表 2-232 DCSM_BANK0_Z1_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读类型 | | |
| R | R | 读取 |
| R-0 | R-0 | 读返回 0 |
| 写类型 | | |
| W | W | 写入 |
| 复位或默认值 | | |
| -n | | 复位后或默认值 |
| 寄存器数组变量 | | |
| i, j, k, l, m, n | | 当这些变量用于寄存器名称、偏移或地址时，它们指寄存器数组中该寄存器所属重复寄存器组的取值。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指寄存器数组的取值。 |

## 2.15.11.1 B0_Z1_LINKPOINTER 寄存器

偏移地址：0h 复位值：E0000000h

Zone 1 链接指针（Flash BANK0）

表 2-233 B0_Z1_LINKPOINTER 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | LINKPOINTER | | | | |
| R-7h | | | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| LINKPOINTER | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LINKPOINTER | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINKPOINTER | | | | | | | |
| R-0h | | | | | | | |

表 2-234 B0_Z1_LINKPOINTER 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 7h | 保留 |
| 28-0 | LINKPOINTER | R | 0h | 这是 Flash BANK0 的 USER OTP 中 Zone1 区域选择块的已解析链接指针。该值通过使用从 Flash BANK0 的 OTP 中加载的三个物理链接指针值生成。复位类型：SYSRSn |

## 2.15.11.2 Z1_OTPSECLOCK 寄存器

偏移地址：2h　复位值：0000000Fh

Zone 1 OTP 安全锁定

表 2-235 Z1_OTPSECLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | CRCLOCK | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PSWDLOCK | | | | JTAGLOCK | | | |
| R-0h | | | | R-Fh | | | |

表 2-236 Z1_OTPSECLOCK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-12 | RESERVED | R | 0h | 保留 |
| 11-8 | CRCLOCK | R | 0h | 当对 OTP 中 Z1OTP_CRCLOCK 的地址位置发出读操作时，此字段的值从 Z1OTP_CRCLOCK[3:0] 加载。1111：VCU/VCRC 能够对安全存储器计算 CRC。其他值：VCU/VCRC 不能对安全存储器计算 CRC。复位类型：XRSn |
| 7-4 | PSWDLOCK | R | 0h | 当对 OTP 中 Z1OTP_PSWDLOCK 的地址位置发出读操作时，此字段的值从 Z1OTP_PSWDLOCK[3:0] 加载。1111：OTP 中的 CSM 密码位置不受保护，调试器以及从任意位置运行的代码均可读取。其他值：OTP 中的 CSM 密码位置受保护，不解锁 Zone1 则无法读取。复位类型：XRSn |
| 3-0 | JTAGLOCK | R | Fh | 当对 OTP 中 Z1OTP_JTAGLOCK 的地址位置发出读操作时，此字段的值从 Z1OTP_JTAGLOCK[3:0] 加载。1111：允许 JTAG/仿真访问。其他值：不允许 JTAG/仿真访问。复位类型：XRSn |

## 2.15.11.3 Z1_BOOTDEF_HIGH 寄存器

偏移地址：4h　复位值：00000000h

表 2-237 Z1_BOOTDEF_HIGH 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BOOTDEF_HIGH | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BOOTDEF_HIGH | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BOOTDEF_HIGH | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BOOTDEF_HIGH | | | | | | | |
| R-0h | | | | | | | |

表 2-238 Z1_BOOTDEF_HIGH 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | BOOTDEF_HIGH | R | 0h | 复位类型：SYSRSn |

### 2.15.11.4 B0_Z1_LINKPOINTERERR 寄存器

偏移地址：6h　复位值：00000000h

Flash BANK0 链接指针错误

表 2-239 B0_Z1_LINKPOINTERERR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | Z1_LINKPOINTERERR | | | | |
| R-0-0h | | | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1_LINKPOINTERERR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1_LINKPOINTERERR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1_LINKPOINTERERR | | | | | | | |
| R-0h | | | | | | | |

表 2-240 B0_Z1_LINKPOINTERERR 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-29 | RESERVED | R-0 | 0h | 保留 |
| 28-0 | Z1_LINKPOINTERERR | R | 0h | 这些位指示在从 Flash BANK0 的 USER OTP 加载三个物理链接指针值后形成解析链接指针值期间发生的错误。0：无错误。其他：在设置为 1 的位位置上有错误。复位类型：SYSRSn |

### 2.15.11.5 Z1_BOOTPIN_CONFIG 寄存器

偏移地址：8h　复位值：00000000h

启动引脚配置

表 2-241 Z1_BOOTPIN_CONFIG 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BOOTPIN_CONFIG | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BOOTPIN_CONFIG | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BOOTPIN_CONFIG | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BOOTPIN_CONFIG | | | | | | | |
| R-0h | | | | | | | |

表 2-242 Z1_BOOTPIN_CONFIG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | BOOTPIN_CONFIG | R | 0h | 复位类型：SYSRSn |

## 2.15.11.6 Z1_GPREG2 寄存器

偏移地址：Ah　复位值：00000000h

Zone1 通用寄存器 2

表 2-243 Z1_GPREG2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPREG2 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPREG2 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPREG2 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPREG2 | | | | | | | |
| R-0h | | | | | | | |

表 2-244 Z1_GPREG2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | GPREG2 | R | 0h | 复位类型：SYSRSn |

### 2.15.11.7 Z1_BOOTDEF_LOW 寄存器

偏移地址：Ch　复位值：00000000h

启动定义（低 32 位）

表 2-245 Z1_BOOTDEF_LOW 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BOOTDEF_LOW | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BOOTDEF_LOW | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BOOTDEF_LOW | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BOOTDEF_LOW | | | | | | | |
| R-0h | | | | | | | |

表 2-246 Z1_BOOTDEF_LOW 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | BOOTDEF_LOW | R | 0h | 复位类型：SYSRSn |

### 2.15.11.8 Z1_CSMKEY0 寄存器

偏移地址：10h　复位值：00000000h

Zone 1 CSM 密钥 0

表 2-247 Z1_CSMKEY0 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z1_CSMKEY0 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1_CSMKEY0 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1_CSMKEY0 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1_CSMKEY0 | | | | | | | |
| R-0h | | | | | | | |

表 2-248 Z1_CSMKEY0 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z1_CSMKEY0 | R | 0h | 要解锁 Zone1，用户需要向此寄存器写入与 USER OTP 中编程的 Z1_CSMPSWD0 完全相同的值（仅当 USER OTP 中的 128 位密码与写入四个 CSMKEY 寄存器的值匹配时，区域才会解锁）。<br>复位类型：SYSRSn |

## 2.15.11.9 Z1_CSMKEY1 寄存器

偏移地址：12h　复位值：00000000h

Zone 1 CSM 密钥 1

表 2-249 Z1_CSMKEY1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z1_CSMKEY1 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1_CSMKEY1 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1_CSMKEY1 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1_CSMKEY1 | | | | | | | |
| R-0h | | | | | | | |

表 2-250 Z1_CSMKEY1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z1_CSMKEY1 | R | 0h | 要解锁 Zone1，用户需要向此寄存器写入与 USER OTP 中编程的 Z1_CSMPSWD1 完全相同的值（仅当 USER OTP 中的 128 位密码与写入四个 CSMKEY 寄存器的值匹配时，区域才会解锁）。<br>复位类型：SYSRSn |

## 2.15.11.10 Z1_CSMKEY2 寄存器

偏移地址：14h　复位值：00000000h

Zone 1 CSM 密钥 2

表 2-251 Z1_CSMKEY2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z1_CSMKEY2 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1_CSMKEY2 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1_CSMKEY2 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1_CSMKEY2 | | | | | | | |
| R-0h | | | | | | | |

表 2-252 Z1_CSMKEY2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z1_CSMKEY2 | R | 0h | 要解锁 Zone1，用户需要向此寄存器写入与 USER OTP 中编程的 Z1_CSMPSWD2 完全相同的值（仅当 USER OTP 中的 128 位密码与写入四个 CSMKEY 寄存器的值匹配时，区域才会解锁）。<br>复位类型：SYSRSn |

## 2.15.11.11 Z1_CSMKEY3 寄存器

偏移地址：16h　复位值：00000000h

Zone 1 CSM 密钥 3

表 2-253 Z1_CSMKEY3 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z1_CSMKEY3 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1_CSMKEY3 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1_CSMKEY3 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1_CSMKEY3 | | | | | | | |
| R-0h | | | | | | | |

表 2-254 Z1_CSMKEY3 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z1_CSMKEY3 | R | 0h | 要解锁 Zone1，用户需要向此寄存器写入与 USER OTP 中编程的 Z1_CSMPSWD3 完全相同的值（仅当 USER OTP 中的 128 位密码与写入四个 CSMKEY 寄存器的值匹配时，区域才会解锁）。复位类型：SYSRSn |

## 2.15.11.12 Z1_CR 寄存器

偏移地址：19h　复位值：0008h

Zone 1 CSM 控制寄存器

表 2-255 Z1_CR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FORCESEC | 保留 | | | | | | |
| R-0/W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | ARMED | UNSECURE | ALLONE | ALLZERO | 保留 | | |
| R-0h | R-0h | R-0h | R-0h | R-1h | R-0h | | |

表 2-256 Z1_CR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15 | FORCESEC | R-0/W | 0h | 向此字段写入“1”会复位区域状态。如果区域已解锁，则会将区域锁定（安全），并复位此寄存器中的所有位。复位类型：SYSRSn |
| 14-8 | RESERVED | R | 0h | 保留 |
| 7 | RESERVED | R | 0h | 保留 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 6 | ARMED | R | 0h | 0：尚未对 USER OTP 中的 CSM 密码位置执行虚拟读。1：已对 USER OTP 中的 CSM 密码位置执行虚拟读。复位类型：SYSRSn |
| 5 | UNSECURE | R | 0h | 指示 Zone1 的状态。0：区域处于锁定（安全）状态。1：区域处于解锁（非安全）状态。复位类型：SYSRSn |
| 4 | ALLONE | R | 0h | 指示 CSM 密码的状态。0：区域 CSM 密码不全为 1。1：区域 CSM 密码全为 1，器件被永久阻止。复位类型：SYSRSn |
| 3 | ALLZERO | R | 1h | 指示 CSM 密码的状态。0：CSM 密码不全为 0。1：CSM 密码全为 0，器件被永久锁定。复位类型：SYSRSn |
| 2-0 | RESERVED | R | 0h | 保留 |

## 2.15.11.13 B0_Z1_GRABSECTR 寄存器

偏移地址：1Ah　复位值：00000000h

Zone 1 获取 Flash BANK0 扇区寄存器

表 2-257 B0_Z1_GRABSECTR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GRAB_SECT15 | | GRAB_SECT14 | | GRAB_SECT13 | | GRAB_SECT12 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GRAB_SECT11 | | GRAB_SECT10 | | GRAB_SECT9 | | GRAB_SECT8 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GRAB_SECT7 | | GRAB_SECT6 | | GRAB_SECT5 | | GRAB_SECT4 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GRAB_SECT3 | | GRAB_SECT2 | | GRAB_SECT1 | | GRAB_SECT0 | |
| R-0h | | R-0h | | R-0h | | R-0h | |

表 2-258 B0_Z1_GRABSECTR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-30 | GRAB_SECT15 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[31:30] 加载。00：无效。Flash 扇区 15 不可访问。01：请求将 Flash 扇区 15 分配给 Zone1。10：对 Flash 扇区 15 无请求。11：此区域解锁时对 Flash 扇区 15 无请求。否则，此区域锁定时 Flash 扇区 15 不可访问。复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 29-28 | GRAB_SECT14 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[29:28]加载。00：无效。Flash 扇区 14 不可访问。01：请求将 Flash 扇区 14 分配给 Zone1。10：对 Flash 扇区 14 无请求。11：此区域解锁时对 Flash 扇区 14 无请求。否则，此区域锁定时 Flash 扇区 14 不可访问。复位类型：SYSRSn |
| 27-26 | GRAB_SECT13 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[27:26]加载。00：无效。Flash 扇区 13 不可访问。01：请求将 Flash 扇区 13 分配给 Zone1。10：对 Flash 扇区 13 无请求。11：此区域解锁时对 Flash 扇区 13 无请求。否则，此区域锁定时 Flash 扇区 13 不可访问。复位类型：SYSRSn |
| 25-24 | GRAB_SECT12 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[25:24]加载。00：无效。Flash 扇区 12 不可访问。01：请求将 Flash 扇区 12 分配给 Zone1。10：对 Flash 扇区 12 无请求。11：此区域解锁时对 Flash 扇区 12 无请求。否则，此区域锁定时 Flash 扇区 12 不可访问。复位类型：SYSRSn |
| 23-22 | GRAB_SECT11 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[23:22]加载。00：无效。Flash 扇区 11 不可访问。01：请求将 Flash 扇区 11 分配给 Zone1。10：对 Flash 扇区 11 无请求。11：此区域解锁时对 Flash 扇区 11 无请求。否则，此区域锁定时 Flash 扇区 11 不可访问。复位类型：SYSRSn |
| 21-20 | GRAB_SECT10 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[21:20]加载。00：无效。Flash 扇区 10 不可访问。01：请求将 Flash 扇区 10 分配给 Zone1。10：对 Flash 扇区 10 无请求。11：此区域解锁时对 Flash 扇区 10 无请求。否则，此区域锁定时 Flash 扇区 10 不可访问。复位类型：SYSRSn |
| 19-18 | GRAB_SECT9 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[19:18]加载。00：无效。Flash 扇区 9 不可访问。01：请求将 Flash 扇区 9 分配给 Zone1。10：对 Flash 扇区 9 无请求。11：此区域解锁时对 Flash |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 扇区 9 无请求。否则，此区域锁定时 Flash 扇区 9 不可访问。复位类型：SYSRSn |
| 17-16 | GRAB_SECT8 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[17:16] 加载。00：无效。Flash 扇区 8 不可访问。01：请求将 Flash 扇区 8 分配给 Zone1。10：对 Flash 扇区 8 无请求。11：此区域解锁时对 Flash 扇区 8 无请求。否则，此区域锁定时 Flash 扇区 8 不可访问。复位类型：SYSRSn |
| 15-14 | GRAB_SECT7 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[15:14] 加载。00：无效。Flash 扇区 7 不可访问。01：请求将 Flash 扇区 7 分配给 Zone1。10：对 Flash 扇区 7 无请求。11：此区域解锁时对 Flash 扇区 7 无请求。否则，此区域锁定时 Flash 扇区 7 不可访问。复位类型：SYSRSn |
| 13-12 | GRAB_SECT6 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[13:12] 加载。00：无效。Flash 扇区 6 不可访问。01：请求将 Flash 扇区 6 分配给 Zone1。10：对 Flash 扇区 6 无请求。11：此区域解锁时对 Flash 扇区 6 无请求。否则，此区域锁定时 Flash 扇区 6 不可访问。复位类型：SYSRSn |
| 11-10 | GRAB_SECT5 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[11:10] 加载。00：无效。Flash 扇区 5 不可访问。01：请求将 Flash 扇区 5 分配给 Zone1。10：对 Flash 扇区 5 无请求。11：此区域解锁时对 Flash 扇区 5 无请求。否则，此区域锁定时 Flash 扇区 5 不可访问。复位类型：SYSRSn |
| 9-8 | GRAB_SECT4 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[9:8] 加载。00：无效。Flash 扇区 4 不可访问。01：请求将 Flash 扇区 4 分配给 Zone1。10：对 Flash 扇区 4 无请求。11：此区域解锁时对 Flash 扇区 4 无请求。否则，此区域锁定时 Flash 扇区 4 不可访问。复位类型：SYSRSn |
| 7-6 | GRAB_SECT3 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[7:6] 加载。00：无效。Flash 扇区 3 不可访问。01： |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 请求将 Flash 扇区 3 分配给 Zone1。10：对 Flash 扇区 3 无请求。11：此区域解锁时对 Flash 扇区 3 无请求。否则，此区域锁定时 Flash 扇区 3 不可访问。复位类型：SYSRSn |
| 5-4 | GRAB_SECT2 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[5:4] 加载。00：无效。Flash 扇区 2 不可访问。01：请求将 Flash 扇区 2 分配给 Zone1。10：对 Flash 扇区 2 无请求。11：此区域解锁时对 Flash 扇区 2 无请求。否则，此区域锁定时 Flash 扇区 2 不可访问。复位类型：SYSRSn |
| 3-2 | GRAB_SECT1 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[3:2] 加载。00：无效。Flash 扇区 1 不可访问。01：请求将 Flash 扇区 1 分配给 Zone1。10：对 Flash 扇区 1 无请求。11：此区域解锁时对 Flash 扇区 1 无请求。否则，此区域锁定时 Flash 扇区 1 不可访问。复位类型：SYSRSn |
| 1-0 | GRAB_SECT0 | R | 0h | 当对 Flash BANK0 的 USER OTP 中 B0_Z1OTP_GRABSECT 的地址位置发出读操作时，此字段的值从 B0_Z1OTP_GRABSECT[1:0] 加载。00：无效。Flash 扇区 0 不可访问。01：请求将 Flash 扇区 0 分配给 Zone1。10：对 Flash 扇区 0 无请求。11：此区域解锁时对 Flash 扇区 0 无请求。否则，此区域锁定时 Flash 扇区 0 不可访问。复位类型：SYSRSn |

## 2.15.11.14 Z1_GRABRAMR 寄存器

偏移地址：1Ch　复位值：00000000h

Zone 1 获取 RAM 块寄存器

表 2-259 Z1_GRABRAMR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GRAB_RAM7 | | GRAB_RAM6 | | GRAB_RAM5 | | GRAB_RAM4 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | 保留 | | 保留 | | 保留 | |
| R-0h | | R-0h | | R-0h | | R-0h | |

表 2-260 Z1_GRABRAMR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-14 | GRAB_RAM7 | R | 0h | 当对 USER OTP 中 Z1OTP_GRABRAM 的地址位置发出读操作时，此字段的值从 Z1OTP_GRABRAM[15:14] 加载。00：无效。LS7 RAM 不可访问。01：请求将 LS7 RAM 分配给 Zone1。10：对 LS7 RAM 无请求。11：此区域解锁时对 LS7 RAM 无请求。否则，此区域锁定时 LS7 RAM 不可访问。复位类型：SYSRSn |
| 13-12 | GRAB_RAM6 | R | 0h | 当对 USER OTP 中 Z1OTP_GRABRAM 的地址位置发出读操作时，此字段的值从 Z1OTP_GRABRAM[13:12] 加载。00：无效。LS6 RAM 不可访问。01：请求将 LS6 RAM 分配给 Zone1。10：对 LS6 RAM 无请求。11：此区域解锁时对 LS6 RAM 无请求。否则，此区域锁定时 LS6 RAM 不可访问。复位类型：SYSRSn |
| 11-10 | GRAB_RAM5 | R | 0h | 当对 USER OTP 中 Z1OTP_GRABRAM 的地址位置发出读操作时，此字段的值从 Z1OTP_GRABRAM[11:10] 加载。00：无效。LS5 RAM 不可访问。01：请求将 LS5 RAM 分配给 Zone1。10：对 LS5 RAM 无请求。11：此区域解锁时对 LS5 RAM 无请求。否则，此区域锁定时 LS5 RAM 不可访问。复位类型：SYSRSn |
| 9-8 | GRAB_RAM4 | R | 0h | 当对 USER OTP 中 Z1OTP_GRABRAM 的地址位置发出读操作时，此字段的值从 Z1OTP_GRABRAM[9:8] 加载。00：无效。LS4 RAM 不可访问。01：请求将 LS4 RAM 分配给 Zone1。10：对 LS4 RAM 无请求。11：此区域解 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 锁时对 LS4 RAM 无请求。否则，此区域锁定时 LS4 RAM 不可访问。复位类型：SYSRSn |
| 7-6 | RESERVED | R | 0h | 保留 |
| 5-4 | RESERVED | R | 0h | 保留 |
| 3-2 | RESERVED | R | 0h | 保留 |
| 1-0 | RESERVED | R | 0h | 保留 |

## 2.15.12 DCSM_BANK0_Z2_REGS 寄存器

表中列出了 DCSM_BANK0_Z2_REGS 寄存器的存储器映射寄存器。所有未在表中列出的寄存器偏移地址应视为保留位置，不应修改寄存器内容。

表 2-261 DCSM_BANK0_Z2_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | B0_Z2_LINKPOINTER | Flash BANK0 Zone 2 链接指针 |
| 4h | Z2_OTPSECLOCK | Zone 2 OTP 安全锁 |
| 8h | Z2_BOOTDEF_HIGH | 引导定义（高 32 位） |
| Ch | B0_Z2_LINKPOINTERERR | Flash BANK0 链接指针错误 |
| 10h | Z2_BOOTPIN_CONFIG | 引导引脚配置 |
| 14h | Z2_GPREG2 | Zone2 通用寄存器-2 |
| 18h | Z2_BOOTDEF_LOW | 引导定义（低 32 位） |
| 20h | Z2_CSMKEY0 | Zone 2 CSM 密钥 0 |
| 24h | Z2_CSMKEY1 | Zone 2 CSM 密钥 1 |
| 28h | Z2_CSMKEY2 | Zone 2 CSM 密钥 2 |
| 2Ch | Z2_CSMKEY3 | Zone 2 CSM 密钥 3 |
| 32h | Z2_CR | Zone 2 CSM 控制寄存器 |
| 34h | B0_Z2_GRABSECTR | Zone 2 Grab Flash BANK0 扇区寄存器 |
| 38h | Z2_GRABRAMR | Zone 2 Grab RAM 块寄存器 |

复杂位访问类型经过编码以适合小型表格单元格。表中显示了本节中用于访问类型的代码。

表 2-262 DCSM_BANK0_Z2_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | 读取类型 | 读取类型 |
| R | R | 读取 |
| R-0 | R-0 | 读取返回 0 |
| 写入类型 | 写入类型 | 写入类型 |
| W | W | 写入 |
| 复位或默认值 | 复位或默认值 | 复位或默认值 |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | 寄存器数组变量 |

| 访问类型 | 代码 | 描述 |
|---|---|---|
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。 |

### 2.15.12.1 B0_Z2_LINKPOINTER 寄存器

偏移地址：0h　复位值：E0000000h

Flash BANK0 Zone 2 链接指针

表 2-263 B0_Z2_LINKPOINTER 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | LINKPOINTER | | | | |
| R-7h | | | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| LINKPOINTER | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LINKPOINTER | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LINKPOINTER | | | | | | | |
| R-0h | | | | | | | |

表 2-264 B0_Z2_LINKPOINTER 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-29 | RESERVED | R | 7h | 保留 |
| 28-0 | LINKPOINTER | R | 0h | 这是 Zone2 区域选择块 Flash BANK0 USER OTP 的解析链接指针。它是通过使用从 Flash BANK0 的 OTP 加载的三个物理链接指针值生成的。复位类型：SYSRSn |

### 2.15.12.2 Z2_OTPSECLOCK 寄存器

偏移地址：4h　复位值：0000000Fh

Zone 2 OTP 安全锁

表 2-265 Z2_OTPSECLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | CRCLOCK | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PSWDLOCK | | | | JTAGLOCK | | | |
| R-0h | | | | R-Fh | | | |

表 2-266 Z2_OTPSECLOCK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-12 | RESERVED | R | 0h | 保留 |
| 11-8 | CRCLOCK | R | 0h | 当对 OTP 中的 Z2OTP_CRCLOCK 地址位置发出读取时，该字段的值从 Z2_CRCLOCK[3:0] 加载。1111：VCU/VCRC 有能力对安全存储器计算 CRC。其他值：VCU/VCRC 没有能力对安全存储器计算 CRC。复位类型：XRSn |
| 7-4 | PSWDLOCK | R | 0h | 当对 OTP 中的 Z2OTP_PSWDLOCK 地址位置发出读取时，该字段的值从 Z2_PSWDLOCK[3:0] 加载。1111：OTP 中的 CSM 密码位置不受保护，调试器和从任何位置运行的代码都可以读取。其他值：OTP 中的 CSM 密码位置受保护，若不解锁该区域的 CSM 则无法读取。复位类型：XRSn |
| 3-0 | JTAGLOCK | R | Fh | 当对 OTP 中的 Z2_JATGLOCK 地址位置发出读取时，该字段的值从 Z2OTP_JATGLOCK[3:0] 加载。1111：允许 JTAG/仿真访问。其他值：不允许 JTAG/仿真访问。复位类型：XRSn |

### 2.15.12.3 Z2_BOOTDEF_HIGH 寄存器

偏移地址：8h　复位值：00000000h

引导定义（高 32 位）

表 2-267 Z2_BOOTDEF_HIGH 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BOOTDEF_HIGH | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BOOTDEF_HIGH | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BOOTDEF_HIGH | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BOOTDEF_HIGH | | | | | | | |
| R-0h | | | | | | | |

表 2-268 Z2_BOOTDEF_HIGH 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | BOOTDEF_HIGH | R | 0h | 复位类型：SYSRSn |

## 2.15.12.4 B0_Z2_LINKPOINTERERR 寄存器

偏移地址：ch　复位值：00000000h

Flash BANK0 链接指针错误

表 2-269 B0_Z2_LINKPOINTERERR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | Z2_LINKPOINTERERR | | | | |
| R-0-0h | | | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2_LINKPOINTERERR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2_LINKPOINTERERR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2_LINKPOINTERERR | | | | | | | |
| R-0h | | | | | | | |

表 2-270 B0_Z2_LINKPOINTERERR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-29 | RESERVED | R-0 | 0h | 保留 |
| 28-0 | Z2_LINKPOINTERERR | R | 0h | 这些位指示在从 Flash BANK0 的 USER OTP 加载的三个物理链接指针值之后形成解析链接指针值 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 期间的错误。0：无错误。其他：设置为 1 的位位置发生错误。复位类型：SYSRSn |

## 2.15.12.5 Z2_BOOTPIN_CONFIG 寄存器

偏移地址：10h　复位值：00000000h

引导引脚配置

表 2-271 Z2_BOOTPIN_CONFIG 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BOOTPIN_CONFIG | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BOOTPIN_CONFIG | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BOOTPIN_CONFIG | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BOOTPIN_CONFIG | | | | | | | |
| R-0h | | | | | | | |

表 2-272 Z2_BOOTPIN_CONFIG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 31-0 | BOOTPIN_CONFIG | R | 0h | 复位类型：SYSRSn |

## 2.15.12.6 Z2_GPREG2 寄存器

偏移地址：14h　复位值：00000000h

Zone2 通用寄存器-2

表 2-273 Z2_GPREG2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPREG2 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPREG2 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPREG2 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPREG2 | | | | | | | |
| R-0h | | | | | | | |

表 2-274 Z2_GPREG2 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | GPREG2 | R | 0h | 复位类型：SYSRSn |

## 2.15.12.7 Z2_BOOTDEF_LOW 寄存器

偏移地址：18h 复位值：00000000h

引导定义（低 32 位）

表 2-275 Z2_BOOTDEF_LOW 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BOOTDEF_LOW | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BOOTDEF_LOW | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BOOTDEF_LOW | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BOOTDEF_LOW | | | | | | | |
| R-0h | | | | | | | |

表 2-276 Z2_BOOTDEF_LOW 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | BOOTDEF_LOW | R | 0h | 复位类型：SYSRSn |

### 2.15.12.8 Z2_CSMKEY0 寄存器

偏移地址：20h　复位值：00000000h

Zone 2 CSM 密钥 0

表 2-277 Z2_CSMKEY0 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z2_CSMKEY0 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2_CSMKEY0 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2_CSMKEY0 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2_CSMKEY0 | | | | | | | |
| R-0h | | | | | | | |

表 2-278 Z2_CSMKEY0 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z2_CSMKEY0 | R | 0h | 要解锁 Zone2，用户需要向此寄存器写入与 USER OTP 中编程的 Z2_CSMPSWD0 完全相同的值（仅当 USER OTP 中的 128 位密码与写入四个 CSMKEY 寄存器的值匹配时，区域才会解锁）。<br>复位类型：SYSRSn |

### 2.15.12.9 Z2_CSMKEY1 寄存器

偏移地址：24h　复位值：00000000h

Zone 2 CSM 密钥 1

表 2-279 Z2_CSMKEY1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z2_CSMKEY1 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2_CSMKEY1 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2_CSMKEY1 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2_CSMKEY1 | | | | | | | |
| R-0h | | | | | | | |

表 2-280 Z2_CSMKEY1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z2_CSMKEY1 | R | 0h | 要解锁 Zone2，用户需要向此寄存器写入与 USER OTP 中编程的 Z2_CSMPSWD1 完全相同的值（仅当 USER OTP 中的 128 位密码与写入四个 CSMKEY 寄存器的值匹配时，区域才会解锁）。<br>复位类型：SYSRSn |

## 2.15.12.10 Z2_CSMKEY2 寄存器

偏移地址：28h　复位值：00000000h

Zone 2 CSM 密钥 2

表 2-281 Z2_CSMKEY2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z2_CSMKEY2 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2_CSMKEY2 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2_CSMKEY2 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2_CSMKEY2 | | | | | | | |
| R-0h | | | | | | | |

表 2-282 Z2_CSMKEY2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z2_CSMKEY2 | R | 0h | 要解锁 Zone2，用户需要向此寄存器写入与 USER OTP 中编程的 Z2_CSMPSWD2 完全相同的值（仅当 USER OTP 中的 128 位密码与写入四个 CSMKEY 寄存器的值匹配时，区域才会解锁）。<br>复位类型：SYSRSn |

### 2.15.12.11 Z2_CSMKEY3 寄存器

偏移地址：2ch　复位值：00000000h

Zone 2 CSM 密钥 3

表 2-283 Z2_CSMKEY3 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z2_CSMKEY3 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2_CSMKEY3 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2_CSMKEY3 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2_CSMKEY3 | | | | | | | |
| R-0h | | | | | | | |

表 2-284 Z2_CSMKEY3 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z2_CSMKEY3 | R | 0h | 要解锁 Zone2，用户需要向此寄存器写入与 USER OTP 中编程的 Z2_CSMPSWD3 完全相同的值（仅当 USER OTP 中的 128 位密码与写入四个 CSMKEY 寄存器的值匹配时，区域才会解锁）。<br>复位类型：SYSRSn |

### 2.15.12.12 Z2_CR 寄存器

偏移地址：32h　复位值：0008h

Zone 2 CSM 控制寄存器

表 2-285 Z2_CR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FORCESEC | 保留 | | | | | | |
| R-0/W-0h | R-0-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | ARMED | UNSECURE | ALLONE | ALLZERO | 保留 | | |
| R-0h | R-0h | R-0h | R-0h | R-1h | R-0h | | |

表 2-286 Z2_CR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15 | FORCESEC | R-0/W | 0h | 向此字段写入“1”会复位区域状态。如果区域已解锁，它会锁定（保护）区域，并同时复位此寄存器中的所有位。复位类型：SYSRSn |
| 14-8 | RESERVED | R-0 | 0h | 保留 |
| 7 | RESERVED | R | 0h | 保留 |
| 6 | ARMED | R | 0h | 0：尚未对 USER OTP 中的 CSM 密码位置执行虚拟读取。1：已对 USER OTP 中的 CSM 密码位置执行虚拟读取。复位类型：SYSRSn |
| 5 | UNSECURE | R | 0h | 指示区域的状态。0：区域处于锁定（安全）状态。1：区域处于解锁（非安全）状态。复位类型：SYSRSn |
| 4 | ALLONE | R | 0h | 指示 CSM 密码的状态。0：区域 CSM 密码不全为 1。1：区域 CSM 密码全为 1，器件被永久阻止。复位类型：SYSRSn |
| 3 | ALLZERO | R | 1h | 指示 CSM 密码的状态。0：CSM 密码不全为 0。1：CSM 密码全为 0，器件被永久锁定。复位类型：SYSRSn |
| 2-0 | RESERVED | R | 0h | 保留 |

## 2.15.12.13 B0_Z2_GRABSECTR 寄存器

偏移地址：34h　复位值：00000000h

Zone 2 Grab Flash BANK0 扇区寄存器

表 2-287 B0_Z2_GRABSECTR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GRAB_SECT15 | | GRAB_SECT14 | | GRAB_SECT13 | | GRAB_SECT12 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GRAB_SECT11 | | GRAB_SECT10 | | GRAB_SECT9 | | GRAB_SECT8 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GRAB_SECT7 | | GRAB_SECT6 | | GRAB_SECT5 | | GRAB_SECT4 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GRAB_SECT3 | | GRAB_SECT2 | | GRAB_SECT1 | | GRAB_SECT0 | |
| R-0h | | R-0h | | R-0h | | R-0h | |

表 2-288 B0_Z2_GRABSECTR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-30 | GRAB_SECT15 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[31:30] 加载。00：无效。Flash 扇区 15 不可访问。01：请求将 Flash 扇区 15 分配给 Zone2。10：对 Flash 扇区 15 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 15 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 15 不可访问。复位类型：SYSRSn |
| 29-28 | GRAB_SECT14 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[29:28] 加载。00：无效。Flash 扇区 14 不可访问。01：请求将 Flash 扇区 14 分配给 Zone2。10：对 Flash 扇区 14 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 14 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 14 不可访问。复位类型：SYSRSn |
| 27-26 | GRAB_SECT13 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[27:26] 加载。00：无效。Flash 扇区 13 不可访问。01：请求将 Flash 扇区 13 分配给 Zone2。10：对 Flash 扇区 13 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 13 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 13 不可访问。复位类型：SYSRSn |
| 25-24 | GRAB_SECT12 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 字段的值从 B0_Z2OTP_GRABSECT[25:24] 加载。00：无效。Flash 扇区 12 不可访问。01：请求将 Flash 扇区 12 分配给 Zone2。10：对 Flash 扇区 12 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 12 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 12 不可访问。复位类型：SYSRSn |
| 23-22 | GRAB_SECT11 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[23:22] 加载。00：无效。Flash 扇区 11 不可访问。01：请求将 Flash 扇区 11 分配给 Zone2。10：对 Flash 扇区 11 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 11 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 11 不可访问。复位类型：SYSRSn |
| 21-20 | GRAB_SECT10 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[21:20] 加载。00：无效。Flash 扇区 10 不可访问。01：请求将 Flash 扇区 10 分配给 Zone2。10：对 Flash 扇区 10 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 10 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 10 不可访问。复位类型：SYSRSn |
| 19-18 | GRAB_SECT9 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[19:18] 加载。00：无效。Flash 扇区 9 不可访问。01：请求将 Flash 扇区 9 分配给 Zone2。10：对 Flash 扇区 9 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 9 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 9 不可访问。复位类型：SYSRSn |
| 17-16 | GRAB_SECT8 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[17:16] 加载。00：无效。Flash 扇区 8 不可访问。01：请求将 Flash 扇区 8 分配给 Zone2。10：对 Flash 扇区 8 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 8 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 8 不可访问。复位类型：SYSRSn |
| 15-14 | GRAB_SECT7 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[15:14] 加 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 载。00：无效。Flash 扇区 7 不可访问。01：请求将 Flash 扇区 7 分配给 Zone2。10：对 Flash 扇区 7 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 7 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 7 不可访问。复位类型：SYSRSn |
| 13-12 | GRAB_SECT6 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[13:12] 加载。00：无效。Flash 扇区 6 不可访问。01：请求将 Flash 扇区 6 分配给 Zone2。10：对 Flash 扇区 6 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 6 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 6 不可访问。复位类型：SYSRSn |
| 11-10 | GRAB_SECT5 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[11:10] 加载。00：无效。Flash 扇区 5 不可访问。01：请求将 Flash 扇区 5 分配给 Zone2。10：对 Flash 扇区 5 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 5 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 5 不可访问。复位类型：SYSRSn |
| 9-8 | GRAB_SECT4 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[9:8] 加载。00：无效。Flash 扇区 4 不可访问。01：请求将 Flash 扇区 4 分配给 Zone2。10：对 Flash 扇区 4 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 4 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 4 不可访问。复位类型：SYSRSn |
| 7-6 | GRAB_SECT3 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[7:6] 加载。00：无效。Flash 扇区 3 不可访问。01：请求将 Flash 扇区 3 分配给 Zone2。10：对 Flash 扇区 3 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 3 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 3 不可访问。复位类型：SYSRSn |
| 5-4 | GRAB_SECT2 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[5:4] 加载。00：无效。Flash 扇区 2 不可访问。01：请求将 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | Flash 扇区 2 分配给 Zone2。10：对 Flash 扇区 2 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 2 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 2 不可访问。复位类型：SYSRSn |
| 3-2 | GRAB_SECT1 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[3:2] 加载。00：无效。Flash 扇区 1 不可访问。01：请求将 Flash 扇区 1 分配给 Zone2。10：对 Flash 扇区 1 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 1 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 1 不可访问。复位类型：SYSRSn |
| 1-0 | GRAB_SECT0 | R | 0h | 当对 Flash BANK0 的 USER OTP 中的 B0_Z2OTP_GRABSECT 地址位置发出读取时，该字段的值从 B0_Z2OTP_GRABSECT[1:0] 加载。00：无效。Flash 扇区 0 不可访问。01：请求将 Flash 扇区 0 分配给 Zone2。10：对 Flash 扇区 0 无请求 11：当此区域处于解锁状态时，对 Flash 扇区 0 无请求。否则，如果此区域处于锁定状态，则 Flash 扇区 0 不可访问。复位类型：SYSRSn |

## 2.15.12.14 Z2_GRABRAMR 寄存器

偏移地址：38h　复位值：00000000h

Zone 2 RAM 块分配寄存器。寄存器位布局及字段说明见下表。

表 2-289 Z2_GRABRAMR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GRAB_RAM7 | | GRAB_RAM6 | | GRAB_RAM5 | | GRAB_RAM4 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | 保留 | | 保留 | | 保留 | |
| R-0h | | R-0h | | R-0h | | R-0h | |

表 2-290 Z2_GRABRAMR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-14 | GRAB_RAM7 | R | 0h | 当读取 USER OTP 中的 Z2OTP_GRABRAM 地址位置时，本字段从 Z2OTP_GRABRAM[15:14] 加载其值。<br>00：无效。LS7 RAM 不可访问。<br>01：请求将 LS7 RAM 分配给 Zone2。<br>10：不请求分配 LS7 RAM。<br>11：当本区域处于解锁状态时，不请求分配 LS7 RAM；当本区域处于锁定状态时，LS7 RAM 不可访问。<br>复位类型：SYSRSn |
| 13-12 | GRAB_RAM6 | R | 0h | 当读取 USER OTP 中的 Z2OTP_GRABRAM 地址位置时，本字段从 Z2OTP_GRABRAM[13:12] 加载其值。<br>00：无效。LS6 RAM 不可访问。<br>01：请求将 LS6 RAM 分配给 Zone2。<br>10：不请求分配 LS6 RAM。<br>11：当本区域处于解锁状态时，不请求分配 LS6 RAM；当本区域处于锁定状态时，LS6 RAM 不可访问。<br>复位类型：SYSRSn |
| 11-10 | GRAB_RAM5 | R | 0h | 当读取 USER OTP 中的 Z2OTP_GRABRAM 地址位置时，本字段从 Z2OTP_GRABRAM[11:10] 加载其值。<br>00：无效。LS5 RAM 不可访问。<br>01：请求将 LS5 RAM 分配给 Zone2。<br>10：不请求分配 LS5 RAM。<br>11：当本区域处于解锁状态时，不请求分配 LS5 RAM；当本区域处于锁定状态时，LS5 RAM 不可访问。<br>复位类型：SYSRSn |
| 9-8 | GRAB_RAM4 | R | 0h | 当读取 USER OTP 中的 Z2OTP_GRABRAM 地址位置时，本字段从 Z2OTP_GRABRAM[9:8] 加载其值。<br>00：无效。LS4 RAM 不可访问。<br>01：请求将 LS4 RAM 分配给 Zone2。<br>10：不请求分配 LS4 RAM。<br>11：当本区域处于解锁状态时，不请求分配 LS4 RAM；当本区域处于锁定状态时，LS4 RAM 不可访问。<br>复位类型：SYSRSn |
| 7-6 | RESERVED | R | 0h | 保留 |
| 5-4 | RESERVED | R | 0h | 保留 |
| 3-2 | RESERVED | R | 0h | 保留 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 1-0 | RESERVED | R | 0h | 保留 |

## 2.15.13 DCSM_COMMON_REGS 寄存器

表中列出了 DCSM_COMMON_REGS 寄存器的存储器映射寄存器。所有未在表中列出的寄存器偏移地址应视为保留位置，不应修改寄存器内容。

表 2-291 DCSM_COMMON_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 4h | B0_SECTSTAT | Flash BANK0 扇区状态寄存器 |
| 8h | RAMSTAT | RAM 状态寄存器 |

复杂位访问类型经过编码以适合小型表格单元格。表中显示了本节中用于访问类型的代码。

表 2-292 DCSM_COMMON_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | 读取类型 | 读取类型 |
| R | R | 读取 |
| R-0 | R-0 | 读取返回 0 |
| 写入类型 | 写入类型 | 写入类型 |
| W | W | 写入 |
| W1S | W1S | 写 1 置位 |
| 复位或默认值 | 复位或默认值 | 复位或默认值 |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | 寄存器数组变量 |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。 |

### 2.15.13.1 B0_SECTSTAT 寄存器

偏移地址：4h　复位值：00000000h

Flash BANK0 扇区状态寄存器

表 2-293 B0_SECTSTAT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| STATUS_SECT15 | | STATUS_SECT14 | | STATUS_SECT13 | | STATUS_SECT12 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| STATUS_SECT11 | | STATUS_SECT10 | | STATUS_SECT9 | | STATUS_SECT8 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| STATUS_SECT7 | | STATUS_SECT6 | | STATUS_SECT5 | | STATUS_SECT4 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STATUS_SECT3 | | STATUS_SECT2 | | STATUS_SECT1 | | STATUS_SECT0 | |
| R-0h | | R-0h | | R-0h | | R-0h | |

表 2-294 B0_SECTSTAT 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-30 | STATUS_SECT15 | R | 0h | 反映 Flash BANK0 扇区 15 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 29-28 | STATUS_SECT14 | R | 0h | 反映 Flash BANK0 扇区 14 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 27-26 | STATUS_SECT13 | R | 0h | 反映 Flash BANK0 扇区 13 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 25-24 | STATUS_SECT12 | R | 0h | 反映 Flash BANK0 扇区 12 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 23-22 | STATUS_SECT11 | R | 0h | 反映 Flash BANK0 扇区 11 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 21-20 | STATUS_SECT10 | R | 0h | 反映 Flash BANK0 扇区 10 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 19-18 | STATUS_SECT9 | R | 0h | 反映 Flash BANK0 扇区 9 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 17-16 | STATUS_SECT8 | R | 0h | 反映 Flash BANK0 扇区 8 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 15-14 | STATUS_SECT7 | R | 0h | 反映 Flash BANK0 扇区 7 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 13-12 | STATUS_SECT6 | R | 0h | 反映 Flash BANK0 扇区 6 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 11-10 | STATUS_SECT5 | R | 0h | 反映 Flash BANK0 扇区 5 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 9-8 | STATUS_SECT4 | R | 0h | 反映 Flash BANK0 扇区 4 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 7-6 | STATUS_SECT3 | R | 0h | 反映 Flash BANK0 扇区 3 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 5-4 | STATUS_SECT2 | R | 0h | 反映 Flash BANK0 扇区 2 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 3-2 | STATUS_SECT1 | R | 0h | 反映 Flash BANK0 扇区 1 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 1-0 | STATUS_SECT0 | R | 0h | 反映 Flash BANK0 扇区 0 的状态。00：扇区不可访问 01：扇区属于 Zone1。10：扇区属于 Zone2。11：扇区非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |

## 2.15.13.2 RAMSTAT 寄存器

偏移地址：8h　复位值：00000000h

RAM 状态寄存器

表 2-295 RAMSTAT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| STATUS_RAM7 | | STATUS_RAM6 | | STATUS_RAM5 | | STATUS_RAM4 | |
| R-0h | | R-0h | | R-0h | | R-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | 保留 | | 保留 | | 保留 | |
| R-0h | | R-0h | | R-0h | | R-0h | |

表 2-296 RAMSTAT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-14 | STATUS_RAM7 | R | 0h | 反映 LS7 RAM 的状态。00：RAM 不可访问<br>01：RAM 属于 Zone1。10：RAM 属于 Zone2。<br>11：RAM 非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 13-12 | STATUS_RAM6 | R | 0h | 反映 LS6 RAM 的状态。00：RAM 不可访问<br>01：RAM 属于 Zone1。10：RAM 属于 Zone2。<br>11：RAM 非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 11-10 | STATUS_RAM5 | R | 0h | 反映 LS5 RAM 的状态。00：RAM 不可访问<br>01：RAM 属于 Zone1。10：RAM 属于 Zone2。<br>11：RAM 非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 9-8 | STATUS_RAM4 | R | 0h | 反映 LS4 RAM 的状态。00：RAM 不可访问<br>01：RAM 属于 Zone1。10：RAM 属于 Zone2。<br>11：RAM 非安全，两个区域中运行的代码都具有完全访问权限。复位类型：SYSRSn |
| 7-6 | RESERVED | R | 0h | 保留 |
| 5-4 | RESERVED | R | 0h | 保留 |
| 3-2 | RESERVED | R | 0h | 保留 |
| 1-0 | RESERVED | R | 0h | 保留 |

## 2.15.14 MEM_CFG_REGS 寄存器

表中列出了 MEM_CFG_REGS 寄存器的存储器映射寄存器。所有未在表中列出的寄存器偏移地址应视为保留位置，不应修改寄存器内容。

表 2-297 MEM_CFG_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 20h | DxTEST | 专用 RAM 测试寄存器 |
| 24h | DxINIT | 专用 RAM 初始化寄存器 |
| 28h | DxINITDONE | 专用 RAM 初始化完成状态寄存器 |
| 60h | LSxTEST | 本地共享 RAM 测试寄存器 |
| 64h | LSxINIT | 本地共享 RAM 初始化寄存器 |
| 68h | LSxINITDONE | 本地共享 RAM 初始化完成状态寄存器 |
| A0h | GSxTEST | 全局共享 RAM 测试寄存器 |
| A4h | GSxINIT | 全局共享 RAM 初始化寄存器 |
| A8h | GSxINITDONE | 全局共享 RAM 初始化完成状态寄存器 |
| 144h | ROM_TEST | ROM 测试寄存器 |
| 148h | ROM_FORCE_ERROR | ROM 强制错误寄存器 |

复杂位访问类型经过编码以适合小型表格单元格。表中显示了本节中用于访问类型的代码。

表 2-298 MEM_CFG_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | 读取类型 | 读取类型 |
| R | R | 读取 |
| R-0 | R-0 | 读取返回 0 |
| 写入类型 | 写入类型 | 写入类型 |
| W | W | 写入 |
| W1S | W1S | 写 1 置位 |
| WSonce | WSonce | 写一次置位 |
| 复位或默认值 | 复位或默认值 | 复位或默认值 |
| -n | 复位后的值或默认值 | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | 寄存器数组变量 |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。 |

### 2.15.14.1 DxTEST 寄存器

偏移地址：20h　复位值：00000000h

专用 RAM 测试寄存器

表 2-299 DxTEST 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | TEST_M1 | | TEST_M0 | |
| R-0h | | | | R/W-0h | | R/W-0h | |

表 2-300 DxTEST 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | 保留 |
| 3-2 | TEST_M1 | R/W | 0h | 选择 M1 RAM 的不同模式：00：Mx RAM处于功能模式<br>01：Mx RAM仅允许对数据位进行写操作，不允许对ECC位进行写操作，不会进行校验。<br>10：Mx RAM仅允许对ECC位进行写操作，不允许对数据位进行写操作，不会进行校验。<br>11：Mx RAM处于测试模式，正常进行校验，但不产生中断，错误记录在TEST_ERROR_REGS中<br>复位类型：SYSRSn |
| 1-0 | TEST_M0 | R/W | 0h | 选择 M0 RAM 的不同模式：00：Mx RAM处于功能模式<br>01：Mx RAM仅允许对数据位进行写操作，不允许对ECC位进行写操作，不会进行校验。<br>10：Mx RAM仅允许对ECC位进行写操作，不允许对数据位进行写操作，不会进行校验。<br>11：Mx RAM处于测试模式，正常进行校验，但不产生中断，错误记录在TEST_ERROR_REGS中<br>复位类型：SYSRSn |

## 2.15.14.2 DxINIT 寄存器

偏移地址：24h　复位值：00000000h

专用 RAM 初始化寄存器

表 2-301 DxINIT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | INIT_M1 | INIT_M0 |
| R-0h | | | | | | R-0/W1S-0h | R-0/W1S-0h |

表 2-302 DxINIT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | INIT_M1 | R-0/W1S | 0h | M1 RAM 的 RAM 初始化控制：0：无。1：启动 RAM 初始化。复位类型：SYSRSn |
| 0 | INIT_M0 | R-0/W1S | 0h | M0 RAM 的 RAM 初始化控制：0：无。1：启动 RAM 初始化。复位类型：SYSRSn |

### 2.15.14.3 DxINITDONE 寄存器

偏移地址：28h　复位值：00000000h

专用 RAM 初始化完成状态寄存器

表 2-303 DxINITDONE 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | INITDONE_M1 | INITDONE_M0 |
| R-0h | | | | | | R-0h | R-0h |

表 2-304 DxINITDONE 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | INITDONE_M1 | R | 0h | M1 RAM 的 RAM 初始化状态：0：RAM 初始化未完成。1：RAM 初始化已完成。复位类型：SYSRSn |
| 0 | INITDONE_M0 | R | 0h | M0 RAM 的 RAM 初始化状态：0：RAM 初始化未完成。1：RAM 初始化已完成。复位类型：SYSRSn |

### 2.15.14.4 LSxTEST 寄存器

偏移地址：60h　复位值：00000000h

本地共享 RAM TEST 寄存器

表 2-305 LSxTEST 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TEST_LS3 | | TEST_LS2 | | TEST_LS1 | | TEST_LS0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 2-306 LSxTEST 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | 保留 |
| 7-6 | TEST_LS3 | R/W | 0h | 选择 LS3 RAM 的不同模式：00：LSx RAM处于功能模式<br>01：LSx RAM仅允许对数据位进行写操作，不允许对ECC位进行写操作，不会进行校验。<br>10：LSx RAM仅允许对ECC位进行写操作，不允许对数据位进行写操作，不会进行校验。<br>11：LSx RAM处于测试模式，正常进行校验，但不产生中断，错误记录在TEST_ERROR_REGS中<br>复位类型：SYSRSn |
| 5-4 | TEST_LS2 | R/W | 0h | 选择 LS2 RAM 的不同模式：00：LSx RAM处于功能模式<br>01：LSx RAM仅允许对数据位进行写操作，不允许对ECC位进行写操作，不会进行校验。<br>10：LSx RAM仅允许对ECC位进行写操作，不允许对数据位进行写操作，不会进行校验。<br>11：LSx RAM处于测试模式，正常进行校验，但不产生中断，错误记录在TEST_ERROR_REGS中<br>复位类型：SYSRSn |
| 3-2 | TEST_LS1 | R/W | 0h | 选择 LS1 RAM 的不同模式：00：LSx RAM处于功能模式<br>01：LSx RAM仅允许对数据位进行写操作，不允许对ECC位进行写操作，不会进行校验。<br>10：LSx RAM仅允许对ECC位进行写操作，不允许对数据位进行写操作，不会进行校验。<br>11：LSx RAM处于测试模式，正常进行校验，但不产生中断，错误记录在TEST_ERROR_REGS中<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 1-0 | TEST_LS0 | R/W | 0h | 选择 LS0 RAM 的不同模式：00：LSx RAM处于功能模式<br>01：LSx RAM仅允许对数据位进行写操作，不允许对ECC位进行写操作，不会进行校验。<br>10：LSx RAM仅允许对ECC位进行写操作，不允许对数据位进行写操作，不会进行校验。<br>11：LSx RAM处于测试模式，正常进行校验，但不产生中断，错误记录在TEST_ERROR_REGS中<br>复位类型：SYSRSn |

### 2.15.14.5 LSxINIT 寄存器

偏移地址：64h 复位值：00000000h

本地共享 RAM Init 寄存器

表 2-307 LSxINIT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | INIT_LS3 | INIT_LS2 | INIT_LS1 | INIT_LS0 |
| R-0h | | | | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 2-308 LSxINIT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | 保留 |
| 3 | INIT_LS3 | R-0/W1S | 0h | LS3 RAM 的 RAM 初始化控制：0：无操作。1：启动 RAM 初始化。复位类型：SYSRSn |
| 2 | INIT_LS2 | R-0/W1S | 0h | LS2 RAM 的 RAM 初始化控制：0：无操作。1：启动 RAM 初始化。复位类型：SYSRSn |
| 1 | INIT_LS1 | R-0/W1S | 0h | LS1 RAM 的 RAM 初始化控制：0：无操作。1：启动 RAM 初始化。复位类型：SYSRSn |
| 0 | INIT_LS0 | R-0/W1S | 0h | LS0 RAM 的 RAM 初始化控制：0：无操作。1：启动 RAM 初始化。复位类型：SYSRSn |

### 2.15.14.6 LSxINITDONE 寄存器

偏移地址：68h　复位值：00000000h

本地共享 RAM InitDone 状态寄存器

表 2-309 LSxINITDONE 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | INITDONE_LS3 | INITDONE_LS2 | INITDONE_LS1 | INITDONE_LS0 |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

表 2-310 LSxINITDONE 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | 保留 |
| 3 | INITDONE_LS3 | R | 0h | LS3 RAM 的 RAM 初始化状态：0：RAM 初始化未完成。1：RAM 初始化已完成。复位类型：SYSRSn |
| 2 | INITDONE_LS2 | R | 0h | LS2 RAM 的 RAM 初始化状态：0：RAM 初始化未完成。1：RAM 初始化已完成。复位类型：SYSRSn |
| 1 | INITDONE_LS1 | R | 0h | LS1 RAM 的 RAM 初始化状态：0：RAM 初始化未完成。1：RAM 初始化已完成。复位类型：SYSRSn |
| 0 | INITDONE_LS0 | R | 0h | LS0 RAM 的 RAM 初始化状态：0：RAM 初始化未完成。1：RAM 初始化已完成。复位类型：SYSRSn |

### 2.15.14.7 GSxTEST 寄存器

偏移地址：A0h　复位值：00000000h

全局共享 RAM TEST 寄存器

表 2-311 GSxTEST 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | TEST_GS0 | |
| R-0h | | | | | | R/W-0h | |

表 2-312 GSxTEST 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1-0 | TEST_GS0 | R/W | 0h | 选择 GS0 RAM 的不同模式：00：GS0RAM处于功能模式<br>01：GS0RAM仅允许对数据位进行写操作，不允许对奇偶位进行写操作，不会进行校验。<br>10：GS0RAM仅允许对奇偶位进行写操作，不允许对数据位进行写操作，不会进行校验。<br>11：GS0RAM处于测试模式，正常进行校验，但不产生中断，错误记录在TEST_ERROR_REGS中<br>复位类型：SYSRSn |

## 2.15.14.8 GSxINIT 寄存器

偏移地址：A4h　复位值：00000000h

全局共享 RAM Init 寄存器

表 2-313 GSxINIT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | INIT_GS0 |
| R-0h | | | | | | | R-0/W1S-0h |

表 2-314 GSxINIT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | 保留 |
| 0 | INIT_GS0 | R-0/W1S | 0h | GS0 RAM 的 RAM 初始化控制：0：无操作。1：启动 RAM 初始化。复位类型：SYSRSn |

## 2.15.14.9 GSxINITDONE 寄存器

偏移地址：A8h　复位值：00000000h

全局共享 RAM InitDone 状态寄存器

表 2-315 GSxINITDONE 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | INITDONE_GS0 |
| R-0h | | | | | | | R-0h |

表 2-316 GSxINITDONE 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | 保留 |
| 0 | INITDONE_GS0 | R | 0h | GS0 RAM 的 RAM 初始化状态：0：RAM 初始化未完成。1：RAM 初始化已完成。复位类型：SYSRSn |

## 2.15.14.10 ROM_TEST 寄存器

偏移地址：144h　复位值：00000000h

ROM TEST 寄存器

表 2-317 ROM_TEST 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | TEST_SECUREROM | | TEST_BOOTROM | |
| R-0h | | | | R/W-0h | | R/W-0h | |

表 2-318 ROM_TEST 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | 保留 |
| 3-2 | TEST_SECUREROM | R/W | 0h | 选择 SECUREROM 的不同模式：SECUREROM:<br>00:CPU的取指和BGCRC可以对SECUREROM的内容进行访问，会进行奇偶校验检查，出错会影响ERROR寄存器组；<br>01：CPU的取指和BGCRC可以对SECUREROM的内容进行访问，不会进行奇偶校验检查<br>10：SECUREROM奇偶校验值不可见，若CPU访问返回0，此时指令异常；BGCRC访问校验值该行为错误<br>11：CPU的取指和BGCRC可以对SECUREROM的内容进行访问，会进行奇偶校验检查，出错会影响TEST_ERROR寄存器组,但不会产生中断；<br>复位类型：SYSRSn |
| 1-0 | TEST_BOOTROM | R/W | 0h | 选择 BOOTROM 的不同模式：BOOTROM：<br>00：CPU和BGCRC可以对BOOTROM的内容进行访问，会进行奇偶校验检查，出错会影响ERROR寄存器组；<br>01：CPU和BGCRC可以对BOOTROM的内容进行访问，不会进行奇偶校验检查<br>10：CPU可正常访问BOOTROM奇偶校验值，但不会进行校验；BGCRC访问校验值该行为错误<br>11：CPU和BGCRC可以对BOOTROM的内容进行访问，会进行奇偶校验检查，出错会影响TEST_ERROR寄存器组,但不会产生中断；<br>复位类型：SYSRSn |

### 2.15.14.11 ROM_FORCE_ERROR 寄存器

偏移地址：148h　复位值：00000000h

ROM Force Error 寄存器

表 2-319 ROM_FORCE_ERROR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | FORCE_SECU REROM_ERROR | FORCE_BOOT ROM_ERROR |
| R-0h | | | | | | R/W-0h | R/W-0h |

表 2-320 ROM_FORCE_ERROR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | FORCE_SECUREROM_ERROR | R/W | 0h | Force寄存器为1后，再读ROM会触发报错。复位类型：SYSRSn |
| 0 | FORCE_BOOTROM_ERROR | R/W | 0h | Force寄存器为1后，再读ROM会触发报错。复位类型：SYSRSn |

## 2.15.15 ACCESS_PROTECTION_REGS 寄存器

表中列出了 ACCESS_PROTECTION_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，且不应修改寄存器内容。

表 2-321 ACCESS_PROTECTION_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 40h | MAVFLG | 主访问受DCSM保护RAM违例标志寄存器 |
| 44h | MAVSET | 主访问违例标志置位寄存器 |
| 48h | MAVCLR | 主访问违例标志清零寄存器 |
| 54h | MCPUAVADDR | 主 CPU 访问违例地址 |

复杂位访问类型经过编码以适应小表格单元格。表中显示了本节中用于访问类型的代码。

表 2-322 ACCESS_PROTECTION_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读类型 | 读类型 | 读类型 |
| R | R | 读 |
| R-0 | R-0 | 读返回 0 |
| 写类型 | 写类型 | 写类型 |
| W | W | 写 |
| W1S | W1S | 写 1 置位 |
| 复位或默认值 | 复位或默认值 | 复位或默认值 |
| - n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | 寄存器数组变量 |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们引用寄存器数组的值，其中该寄存器是重复寄存器组的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它引用寄存器数组的值。 |

## 2.15.15.1 MAVFLG 寄存器

偏移地址：40h　复位值：00000000h

主访问受 DCSM 保护 RAM 违例标志寄存器

表 2-323 MAVFLG 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | DCSM_FLAG | 保留 |
| R-0h | | | | | | R-0h | R-0h |

表 2-324 MAVFLG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | DCSM_FLAG | R | 0h | 若CPU的LSU访问受DCSM保护RAM时标志：0：无违例。1：发生访问违例。复位类型：SYSRSn |
| 0 | RESERVED | R | 0h | 保留 |

## 2.15.15.2 MAVSET 寄存器

偏移地址：44h　复位值：00000000h

主访问违例标志置位寄存器

表 2-325 MAVSET 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | CPUWRITE | 保留 |
| R-0h | | | | | | R-0/W1S-0h | R-0h |

表 2-326 MAVSET 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | CPUWRITE | R-0/W1S | 0h | 0：无操作。1：MAVFLG 寄存器中的 CPU 写访问违例标志将被置位。复位类型：SYSRSn |
| 0 | RESERVED | R | 0h | 保留 |

## 2.15.15.3 MAVCLR 寄存器

偏移地址：48h　复位值：00000000h

主访问违例标志清零寄存器

表 2-327 MAVCLR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | CPUWRITE | 保留 |
| R-0h | | | | | | R-0/W1S-0h | R-0h |

表 2-328 MAVCLR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | CPUWRITE | R-0/W1S | 0h | 0：无操作。1：MAVFLG 寄存器中的 CPU 写访问违例标志将被清零。复位类型：SYSRSn |
| 0 | RESERVED | R | 0h | 保留 |

## 2.15.15.4 MCPUFAVADDR 寄存器

偏移地址：54h　复位值：00000000h

主 CPU 取指访问违例地址

表 2-329 MCPUFAVADDR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MCPUFAVADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MCPUFAVADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MCPUFAVADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MCPUFAVADDR | | | | | | | |
| R-0h | | | | | | | |

表 2-330 MCPUFAVADDR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | MCPUFAVADDR | R | 0h | 该寄存器捕获发生主 CPU的 LOAD/STORE访问被DCSM判断为无效时的访问的目标地址。复位类型：SYSRSn |

## 2.15.16 MEMORY_ERROR_REGS 寄存器

表中列出了 MEMORY_ERROR_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，且不应修改寄存器内容。

表 2-331 MEMORY_ERROR_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | UCERRFLG | 不可纠正错误标志寄存器 |
| 4h | UCERRSET | 不可纠正错误标志置位寄存器 |
| 8h | UCERRCLR | 不可纠正错误标志清零寄存器 |
| Ch | UCCPUREADDR | 不可纠正 CPU 读错误地址 |
| 10h | UCDMAREADDR | 不可纠正 DMA 读错误地址 |
| 20h | UCHICAREADDR | 不可纠正 HICA 读错误地址 |
| 40h | CERRFLG | 可纠正错误标志寄存器 |
| 44h | CERRSET | 可纠正错误标志置位寄存器 |
| 48h | CERRCLR | 可纠正错误标志清零寄存器 |
| 4Ch | CCPUREADDR | 可纠正 CPU 读错误地址 |
| 5Ch | CERRCNT | 可纠正错误计数寄存器 |
| 60h | CERRTHRES | 可纠正错误阈值寄存器 |
| 64h | CEINTFLG | 可纠正错误中断标志状态寄存器 |
| 68h | CEINTCLR | 可纠正错误中断标志清零寄存器 |
| 6Ch | CEINTSET | 可纠正错误中断标志置位寄存器 |
| 70h | CEINTEN | 可纠正错误中断使能寄存器 |
| 74h | NMITYPE | RAM不可纠正错误所触发的中断类型 |

复杂位访问类型经过编码以适应小表格单元格。表中显示了本节中用于访问类型的代码。

表 2-332 MEMORY_ERROR_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读类型 | 读类型 | 读类型 |
| R | R | 读 |
| R-0 | R-0 | 读返回 0 |
| 写类型 | 写类型 | 写类型 |
| W | W | 写 |
| W1S | W1S | 写 1 置位 |
| 复位或默认值 | 复位或默认值 | 复位或默认值 |

| 访问类型 | 代码 | 描述 |
|---|---|---|
| - n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器数组变量 | 寄存器数组变量 |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们引用寄存器数组的值，其中该寄存器是重复寄存器组的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它引用寄存器数组的值。 |

### 2.15.16.1 UCERRFLG 寄存器

偏移地址：0h　复位值：00000000h

不可纠正错误标志寄存器

表 2-333 UCERRFLG 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | HICARDERR | 保留 | 保留 | 保留 | DMARDERR | CPURDERR |
| R-0h | | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 2-334 UCERRFLG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-6 | RESERVED | R | 0h | 保留 |
| 5 | HICARDERR | R | 0h | HICA 不可纠正读错误标志：0：无错误。1：HICA 读期间发生不可纠正错误。复位类型：SYSRSn |
| 4 | RESERVED | R | 0h | 保留 |
| 3 | RESERVED | R | 0h | 保留 |
| 2 | RESERVED | R | 0h | 保留 |
| 1 | DMARDERR | R | 0h | DMA 不可纠正读错误标志：0：无错误。1：DMA 读期间发生不可纠正错误。复位类型：SYSRSn |
| 0 | CPURDERR | R | 0h | CPU 不可纠正读错误标志：0：无错误。1：CPU 读期间发生不可纠正错误。复位类型：SYSRSn |

## 2.15.16.2 UCERRSET 寄存器

偏移地址：4h　复位值：00000000h

不可纠正错误标志置位寄存器

表 2-335 UCERRSET 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | HICARDERR | 保留 | 保留 | 保留 | DMARDERR | CPURDERR |
| R-0h | | R-0/W1S-0h | R-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 2-336 UCERRSET 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-6 | RESERVED | R | 0h | 保留 |
| 5 | HICARDERR | R-0/W1S | 0h | 0：无操作。1：UCERRFLG 寄存器中的 HICA 读错误标志将被置位，且如果已使能，将生成中断。复位类型：SYSRSn |
| 4 | RESERVED | R | 0h | 保留 |
| 3 | RESERVED | R-0/W1S | 0h | 保留 |
| 2 | RESERVED | R-0/W1S | 0h | 保留 |
| 1 | DMARDERR | R-0/W1S | 0h | 0：无操作。1：UCERRFLG 寄存器中的 DMA 读错误标志将被置位，且如果已使能，将生成中断。复位类型：SYSRSn |
| 0 | CPURDERR | R-0/W1S | 0h | 0：无操作。1：UCERRFLG 寄存器中的 CPU 读错误标志将被置位，且如果已使能，将生成中断。复位类型：SYSRSn |

## 2.15.16.3 UCERRCLR 寄存器

偏移地址：8h　复位值：00000000h

不可纠正错误标志清零寄存器

表 2-337 UCERRCLR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | HICARDERR | 保留 | 保留 | 保留 | DMARDERR | CPURDERR |
| R-0h | | R-0/W1S-0h | R-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 2-338 UCERRCLR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-6 | RESERVED | R | 0h | 保留 |
| 5 | HICARDERR | R-0/W1S | 0h | 0：无操作。1：UCERRFLG 寄存器中的 HICA 读错误标志将被清零。复位类型：SYSRSn |
| 4 | RESERVED | R | 0h | 保留 |
| 3 | RESERVED | R-0/W1S | 0h | 保留 |
| 2 | RESERVED | R-0/W1S | 0h | 保留 |
| 1 | DMARDERR | R-0/W1S | 0h | 0：无操作。1：UCERRFLG 寄存器中的 DMA 读错误标志将被清零。复位类型：SYSRSn |
| 0 | CPURDERR | R-0/W1S | 0h | 0：无操作。1：UCERRFLG 寄存器中的 CPU 读错误标志将被清零。复位类型：SYSRSn |

## 2.15.16.4 UCCPUREADDR 寄存器

偏移地址：Ch 复位值：00000000h

不可纠正 CPU 读取错误地址寄存器位布局（31 位至 0 位，包含 UCCPUREADDR 字段）。

表 2-339 UCCPUREADDR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| UCCPUREADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| UCCPUREADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| UCCPUREADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UCCPUREADDR | | | | | | | |
| R-0h | | | | | | | |

表 2-340 UCCPUREADDR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | UCCPUREADDR | R | 0h | 此寄存器捕获导致 CPU 读取/取指访问发生不可纠正 ECC/奇偶校验错误的地址位置。复位类型：SYSRSn |

## 2.15.16.5 UCDMAREADDR 寄存器

偏移地址：10h　复位值：00000000h

不可纠正 DMA 读取错误地址寄存器位布局（31 位至 0 位，包含 UCDMAREADDR 字段）。

表 2-341 UCDMAREADDR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| UCDMAREADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| UCDMAREADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| UCDMAREADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UCDMAREADDR | | | | | | | |
| R-0h | | | | | | | |

表 2-342 UCDMAREADDR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | UCDMAREADDR | R | 0h | 此寄存器捕获导致 DMA 读取访问发生不可纠正奇偶校验错误的地址位置。复位类型：SYSRSn |

## 2.15.16.6 UCHICAREADDR 寄存器

偏移地址：20h 复位值：00000000h

不可纠正 HICA 读取错误地址寄存器位布局（31 位至 0 位，包含 UCHICAREADDR 字段）。

表 2-343 UCHICAREADDR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| UCHICAREADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| UCHICAREADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| UCHICAREADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UCHICAREADDR | | | | | | | |
| R-0h | | | | | | | |

表 2-344 UCHICAREADDR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | UCHICAREADDR | R | 0h | 此寄存器捕获导致 HICA 读取访问发生不可纠正 ECC/奇偶校验错误的地址位置。复位类型：SYSRSn |

## 2.15.16.7 CERRFLG 寄存器

偏移地址：40h 复位值：00000000h

可纠正错误标志寄存器位布局（31 位至 0 位，包含 RESERVED 和 CPURDERR 字段）。

表 2-345 CERRFLG 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | 保留 | 保留 | CPURDERR |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

表 2-346 CERRFLG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-4 | RESERVED | R | 0h | 保留 |
| 3 | RESERVED | R | 0h | 保留 |
| 2 | RESERVED | R | 0h | 保留 |
| 1 | RESERVED | R | 0h | 保留 |
| 0 | CPURDERR | R | 0h | CPU 可纠正读取错误标志。0：无错误。1：CPU 读取期间发生可纠正错误。复位类型：SYSRSn |

### 2.15.16.8 CERRSET 寄存器

偏移地址：44h　复位值：00000000h

可纠正错误标志置位寄存器位布局（31 位至 0 位，包含 RESERVED 和 CPURDERR 字段）。

表 2-347 CERRSET 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | 保留 | 保留 | CPURDERR |
| R-0h | | | | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 2-348 CERRSET 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-4 | RESERVED | R | 0h | 保留 |
| 3 | RESERVED | R-0/W1S | 0h | 保留 |
| 2 | RESERVED | R-0/W1S | 0h | 保留 |
| 1 | RESERVED | R-0/W1S | 0h | 保留 |
| 0 | CPURDERR | R-0/W1S | 0h | 0：无操作。1：CERRFLG 寄存器中的 CPU 读取错误标志将被置位，如果已使能，将产生中断。复位类型：SYSRSn |

## 2.15.16.9 CERRCLR 寄存器

偏移地址：48h　复位值：00000000h

可纠正错误标志清零寄存器位布局（31 位至 0 位，包含 RESERVED 和 CPURDERR 字段）。

表 2-349 CERRCLR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | 保留 | 保留 | 保留 | CPURDERR |
| R-0h | | | | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 2-350 CERRCLR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-4 | RESERVED | R | 0h | 保留 |
| 3 | RESERVED | R-0/W1S | 0h | 保留 |
| 2 | RESERVED | R-0/W1S | 0h | 保留 |
| 1 | RESERVED | R-0/W1S | 0h | 保留 |
| 0 | CPURDERR | R-0/W1S | 0h | 0：无操作。1：CERRFLG 寄存器中的 CPU 读取错误标志将被清零。复位类型：SYSRSn |

## 2.15.16.10 CCPUREADDR 寄存器

偏移地址：4Ch　复位值：00000000h

可纠正 CPU 读取错误地址寄存器位布局（31 位至 0 位，包含 CCPUREADDR 字段）。

表 2-351 CCPUREADDR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CCPUREADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CCPUREADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CCPUREADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCPUREADDR | | | | | | | |
| R-0h | | | | | | | |

表 2-352 CCPUREADDR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | CCPUREADDR | R | 0h | 此寄存器捕获导致 CPU 读取/取指访问发生可纠正 ECC 错误的地址位置。复位类型：SYSRSn |

## 2.15.16.11 CERRCNT 寄存器

偏移地址：5Ch　复位值：00000000h

可纠正错误计数寄存器位布局（31 位至 0 位，包含 CERRCNT 字段）。

表 2-353 CERRCNT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CERRCNT | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CERRCNT | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CERRCNT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CERRCNT | | | | | | | |
| R-0h | | | | | | | |

表 2-354 CERRCNT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | CERRCNT | R | 0h | 此寄存器保存可纠正错误发生的次数。复位类型：SYSRSn |

### 2.15.16.12 CERRTHRES 寄存器

偏移地址：60h　复位值：00000000h

可纠正错误阈值寄存器位布局（31 位至 0 位，包含 RESERVED 和 CERRTHRES 字段）。

表 2-355 CERRTHRES 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CERRTHRES | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CERRTHRES | | | | | | | |
| R/W-0h | | | | | | | |

表 2-356 CERRTHRES 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-0 | CERRTHRES | R/W | 0h | 当 CERRCNT 寄存器中的值大于此寄存器中配置的值时，如果已使能，将产生可纠正中断。复位类型：SYSRSn |

### 2.15.16.13 CEINTFLG 寄存器

偏移地址：64h　复位值：00000000h

可纠正错误中断标志状态寄存器位布局（31 位至 0 位，包含 RESERVED 和 CEINTFLAG 字段）。

表 2-357 CEINTFLG 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | CEINTFLAG |
| R-0h | | | | | | | R-0h |

表 2-358 CEINTFLG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-1 | RESERVED | R | 0h | 保留 |
| 0 | CEINTFLAG | R | 0h | 总可纠正错误计数超过阈值标志。0：总可纠正错误数 < CERRTHRES 寄存器中配置的阈值。1：总可纠正错误数 >= CERRTHRES 寄存器中配置的阈值。复位类型：SYSRSn |

## 2.15.16.14 CEINTCLR 寄存器

偏移地址：68h　复位值：00000000h

可纠正错误中断标志清零寄存器位布局（31 位至 0 位，包含 RESERVED 和 CEINTCLR 字段）。

表 2-359 CEINTCLR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | CEINTCLR |
| R-0h | | | | | | | R-0/W1S-0h |

表 2-360 CEINTCLR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-1 | RESERVED | R | 0h | 保留 |
| 0 | CEINTCLR | R-0/W1S | 0h | 0：无操作。1：CEINTFLG 寄存器中的总可纠正错误计数超过标志将被清零。复位类型：SYSRSn |

## 2.15.16.15 CEINTSET 寄存器

偏移地址：6Ch　复位值：00000000h

可纠正错误中断标志置位寄存器位布局（31 位至 0 位，包含 RESERVED 和 CEINTSET 字段）。

表 2-361 CEINTSET 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | CEINTSET |
| R-0h | | | | | | | R-0/W1S-0h |

表 2-362 CEINTSET 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-1 | RESERVED | R | 0h | 保留 |
| 0 | CEINTSET | R-0/W1S | 0h | 0：无操作。1：CEINTFLG 寄存器中的总可纠正错误计数超过标志将被置位，如果已使能，将产生中断。复位类型：SYSRSn |

## 2.15.16.16 CEINTEN 寄存器

偏移地址：70h　复位值：00000000h

可纠正错误中断使能寄存器位布局（31 位至 0 位，包含 RESERVED 和 CEINTEN 字段）。

表 2-363 CEINTEN 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | CEINTEN |
| R-0h | | | | | | | R/W-0h |

表 2-364 CEINTEN 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-1 | RESERVED | R | 0h | 保留 |
| 0 | CEINTEN | R/W | 0h | 0：可纠正错误中断被禁止。1：可纠正错误中断被使能。复位类型：SYSRSn |

## 2.15.16.17 NMITYPE 寄存器

偏移地址：74h　复位值：00000000h

RAM 不可纠正错误所触发的中断类型（rnmi 或 unmi）。

表 2-365 NMITYPE 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | NMITYPE |
| R-0h | | | | | | | R/W-0h |

表 2-366 NMITYPE 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | 保留 |
| 0 | NMITYPE | R/W | 0h | 0：ram检测到不可纠正错误，默认触发rnmi中断。<br>1：配置为1，触发unmi中断。复位类型：SYSRSn |

## 2.15.17 TEST_ERROR_REGS 寄存器

表中列出了 TEST_ERROR_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址应视为保留位置，且不应修改寄存器内容。

表 2-367 TEST_ERROR_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | CPU_RAM_TEST_ERROR_STS | RAM 测试：错误状态寄存器 |
| 4h | CPU_RAM_TEST_ERROR_STS_CLR | RAM 测试：错误状态清零寄存器 |
| 8h | CPU_RAM_TEST_ERROR_ADDR | RAM 测试：错误地址寄存器 |

复杂位访问类型被编码以适应小型表格单元格。表中显示了本节中访问类型所使用的代码。

表 2-368 TEST_ERROR_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | Read Type | 读取类型 |
| R | R | 读取 |
| R-0 | R -0 | 读取返回 0 |
| 写入类型 | Write Type | 写入类型 |
| W1S | W 1S | 写 1 置位 |
| 复位或默认值 | 复位 or Default Value | 复位或默认值 |
| - n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器 Array Variables | 寄存器数组变量 |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。 |

### 2.15.17.1 CPU_RAM_TEST_ERROR_STS 寄存器

偏移地址：0h 复位值：00000000h

RAM 测试错误状态寄存器位布局（31 位至 0 位，包含 RESERVED、UNC_ERROR 和 COR_ERROR 字段）。

表 2-369 CPU_RAM_TEST_ERROR_STS 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | UNC_ERROR | COR_ERROR |
| R-0h | | | | | | R-0h | R-0h |

表 2-370 CPU_RAM_TEST_ERROR_STS 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | UNC_ERROR | R | 0h | 0：表示在 RAM/ROM 测试模式下未产生“不可纠正错误”。1：表示在 RAM/ROM 测试模式下产生了“不可纠正错误”。复位类型：SYSRSn |
| 0 | COR_ERROR | R | 0h | 0：表示在 RAM/ROM 测试模式下未产生“可纠正错误”。1：表示在 RAM/ROM 测试模式下产生了“可纠正错误”。复位类型：SYSRSn |

### 2.15.17.2 CPU_RAM_TEST_ERROR_STS_CLR 寄存器

偏移地址：4h　复位值：00000000h

RAM 测试错误状态清零寄存器位布局（31 位至 0 位，包含 RESERVED、UNC_ERROR 和 COR_ERROR 字段）。

表 2-371 CPU_RAM_TEST_ERROR_STS_CLR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | UNC_ERROR | COR_ERROR |
| R-0h | | | | | | R-0/W1S-0h | R-0/W1S-0h |

表 2-372 CPU_RAM_TEST_ERROR_STS_CLR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | UNC_ERROR | R-0/W1S | 0h | 0：无影响。1：清零 CPU_RAM_TEST_ERROR_STS 寄存器中的相应位。复位类型：SYSRSn |
| 0 | COR_ERROR | R-0/W1S | 0h | 0：无影响。1：清零 CPU_RAM_TEST_ERROR_STS 寄存器中的相应位。复位类型：SYSRSn |

### 2.15.17.3 CPU_RAM_TEST_ERROR_ADDR 寄存器

偏移地址：8h　复位值：00000000h

RAM 测试错误地址寄存器位布局（31 位至 0 位，包含 ADDR 字段）。

表 2-373 CPU_RAM_TEST_ERROR_ADDR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDR | | | | | | | |
| R-0h | | | | | | | |

表 2-374 CPU_RAM_TEST_ERROR_ADDR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | ADDR | R | 0h | 在 RAM/ROM 测试模式下检测到错误的地址位置。复位类型：SYSRSn |

## 2.15.18 BUS_POLL_ARB_REGS 寄存器

表中列出了 BUS_POLL_ARB_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址应视为保留位置，且不应修改寄存器内容。

表 2-375 BUS_POLL_ARB_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | CPU_IFU_ARBITER_CNT | MxRAM/LSxRAM CPU IFU轮询仲裁阈值次数 |
| 4h | CPU_LSU_ARBITER_CNT | MxRAM/LSxRAM CPU LSU轮询仲裁阈值次数 |
| 8h | CPU_ARBITER_CNT | GSxRAM CPU(IFU/LSU)轮询仲裁阈值次数 |
| Ch | HIC_ARBITER_CNT | GSxRAM HIC轮询仲裁阈值次数 |
| 10h | DMA_ARBITER_CNT | GSxRAM DMA轮询仲裁阈值次数 |

复杂位访问类型被编码以适应小型表格单元格。表中显示了本节中访问类型所使用的代码。

表 2-376 BUS_POLL_ARB_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | Read Type | 读取类型 |
| R | R | 读取 |
| R-0 | R -0 | 读取返回 0 |
| 写入类型 | Write Type | 写入类型 |
| W1S | W 1S | 写 1 置位 |

| 访问类型 | 代码 | 描述 |
| --- | --- | --- |
| 复位或默认值 | 复位 or Default Value | 复位或默认值 |
| - n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器 Array Variables | 寄存器数组变量 |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。 |

### 2.15.18.1 CPU_IFU_ARBITER_CNT 寄存器

偏移地址：0h　复位值：00000000h

MxRAM/LSxRAM CPU IFU 轮询仲裁阈值次数

表 2-377 CPU_IFU_ARBITER_CNT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IFU_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IFU_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| IFU_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IFU_CNT | | | | | | | |
| R/W-0h | | | | | | | |

表 2-378 CPU_IFU_ARBITER_CNT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 31-0 | IFU_CNT | R/W | 0h | 配置CPU IFU访问MxRAM、LSxRAM、BootROM阈值，当寄存器未配置时，此时寄存器阈值为0，IFU和LSU同时发生请求冲突时，这时会产生仲裁使能，优先级开始轮询一次，若继续有请求冲突，则继续轮询。<br>若此时设置阈值的话，只有发生请求冲突并完成仲裁后，才递增相应主设备的计数器，若计数器等于阈值的话，优先级轮询一次，同时计数器清零，之后重新计数；当计数器值再次达到阈值时，再轮转一次优先级。<br>复位类型：SYSRSn |

### 2.15.18.2 CPU_LSU_ARBITER_CNT 寄存器

偏移地址：4h　复位值：00000000h

MxRAM/LSxRAM CPU LSU 轮询仲裁阈值次数

表 2-379 CPU_LSU_ARBITER_CNT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| LSU_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| LSU_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LSU_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LSU_CNT | | | | | | | |
| R/W-0h | | | | | | | |

表 2-380 CPU_LSU_ARBITER_CNT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | LSU_CNT | R/W | 0h | 配置CPU LSU访问MxRAM、LSxRAM、BootROM阈值，当寄存器未配置时，此时寄存器阈值为0，IFU和LSU同时发生请求冲突时，这时会产生仲裁使能，优先级开始轮询一次，若继续有请求冲突，则继续轮询。<br>若此时设置阈值的话，只有发生请求冲突并完成仲裁后，才递增相应主设备的计数器，若计数器等于阈值的话，优先级轮询一次，同时计数器清零，之后重新计数；当计数器值再次达到阈值时，再轮转一次优先级。<br>复位类型：SYSRSn |

### 2.15.18.3 CPU_ARBITER_CNT 寄存器

偏移地址：8h　复位值：00000000h

GSxRAM CPU(IFU/LSU)轮询仲裁阈值次数

表 2-381 CPU_ARBITER_CNT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CPU_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CPU_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CPU_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CPU_CNT | | | | | | | |
| R/W-0h | | | | | | | |

表 2-382 CPU_ARBITER_CNT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | CPU_CNT | R/W | 0h | 对于GSxRAM，有CPU（IFU/LSU）、HIC、DMA参与轮询仲裁，默认优先级为CPU>HIC>DMA，对于GSx的仲裁逻辑是，当寄存器未配置时，此时寄存器阈值为0，所以三者的仲裁使能是一直拉高的，只要有CPU/HIC/DMA其中之一访问就会使得优先级开始轮询，这里只能保证优先级是轮询的，不能保证轮询完优先级的顺序一定是想要的最低的。<br>若此时设置阈值的话，三者的访问请求会增加其各自的计数器，若计数器等于阈值的话，优先级轮询一次，同时计数器清零，之后重新计数；当计数器值再次达到阈值时，再轮转一次优先级。<br>复位类型：SYSRSn |

## 2.15.18.4 HIC_ARBITER_CNT 寄存器

偏移地址：Ch　复位值：00000000h

GSxRAM HIC 轮询仲裁阈值次数

表 2-383 HIC_ARBITER_CNT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| HIC_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| HIC_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HIC_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HIC_CNT | | | | | | | |
| R/W-0h | | | | | | | |

表 2-384 HIC_ARBITER_CNT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | HIC_CNT | R/W | 0h | 对于GSxRAM，有CPU（IFU/LSU）、HIC、DMA参与轮询仲裁，默认优先级为CPU>HIC>DMA，对于GSx的仲裁逻辑是，当寄存器未配置时，此时寄存器阈值为0，所以三者的仲裁使能是一直拉高的，只要有CPU/HIC/DMA其中之一访问就会使得优先级开始轮询，这里只能保证优先级是轮询的，不能保证轮询完优先级的顺序一定是想要的最低的。<br>若此时设置阈值的话，三者的访问请求会增加其各自的计数器，若计数器等于阈值的话，优先级轮询一次，同时计数器清零，之后重新计数；当计数器值再次达到阈值时，再轮转一次优先级。<br>复位类型：SYSRSn |

## 2.15.18.5 DMA_ARBITER_CNT 寄存器

偏移地址：10h　复位值：00000000h

GSxRAM DMA 轮询仲裁阈值次数

表 2-385 DMA_ARBITER_CNT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DMA_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DMA_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DMA_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DMA_CNT | | | | | | | |
| R/W-0h | | | | | | | |

表 2-386 DMA_ARBITER_CNT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | DMA_CNT | R/W | 0h | 对于GSxRAM，有CPU（IFU/LSU）、HIC、DMA参与轮询仲裁，默认优先级为CPU>HIC>DMA，对于GSx的仲裁逻辑是，当寄存器未配置时，此时寄存器阈值为0，所以三者的仲裁使能是一直拉高的，只要有CPU/HIC/DMA其中之一访问就会使得优先级开始轮询，这里只能保证优先级是轮询的，不能保证轮询完优先级的顺序一定是想要的最低的。<br>若此时设置阈值的话，三者的访问请求会增加其各自的计数器，若计数器等于阈值的话，优先级轮询一次，同时计数器清零，之后重新计数；当计数器值再次达到阈值时，再轮转一次优先级。<br>复位类型：SYSRSn |

## 2.15.19 DCSM_BANK0_Z1_OTP 寄存器

表中列出了 DCSM_BANK0_Z1_OTP 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址应视为保留位置，且不应修改寄存器内容。

表 2-387 DCSM_BANK0_Z1_OTP 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | B0_Z1OTP_LINKPOINTER1 | Flash BANK0 的 Z1 OTP 中的 Zone 1 链接指针 1 |
| 4h | B0_Z1OTP_LINKPOINTER2 | Flash BANK0 的 Z1 OTP 中的 Zone 1 链接指针 2 |
| 8h | B0_Z1OTP_LINKPOINTER3 | Flash BANK0 的 Z1 OTP 中的 Zone 1 链接指针 3 |
| Ch | Z1OTP_BOOTPIN_CONFIG | 启动引脚配置 |
| Eh | Z1OTP_GPREG2 | Zone-1 通用寄存器-2 内容 |
| 10h | Z1OTP_PSWDLOCK | Z1 OTP 中的安全密码锁定 |
| 14h | Z1OTP_CRCLOCK | Z1 OTP 中的安全 CRC 锁定 |

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 18h | Z1OTP_JTAGLOCK | Z1 OTP 中的安全 JTAG 锁定 |
| 1Ch | Z1OTP_BOOTDEF_LOW | 启动定义（低 32 位） |
| 1Eh | Z1OTP_BOOTDEF_HIGH | 启动定义（高 32 位） |

复杂位访问类型被编码以适应小型表格单元格。表中显示了本节中访问类型所使用的代码。

表 2-388 DCSM_BANK0_Z1_OTP 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | Read Type | 读取类型 |
| R | R | 读取 |
| 复位或默认值 | 复位 or Default Value | 复位或默认值 |
| - n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器 Array Variables | 寄存器数组变量 |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。 |

### 2.15.19.1 B0_Z1OTP_LINKPOINTER1 寄存器

偏移地址：0h 复位值：1FFFFFFFh

Zone 1 链接指针 1 寄存器位布局（31 位至 0 位，包含 B0_Z1OTP_LINKPOINTER1 字段）。

表 2-389 B0_Z1OTP_LINKPOINTER1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| B0_Z1OTP_LINKPOINTER1 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| B0_Z1OTP_LINKPOINTER1 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| B0_Z1OTP_LINKPOINTER1 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| B0_Z1OTP_LINKPOINTER1 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |

表 2-390 B0_Z1OTP_LINKPOINTER1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | B0_Z1OTP_LINKPOINTER1 | R | 1FFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone1 链接指针 1 位置。复位类型：SYSRSn |

## 2.15.19.2 B0_Z1OTP_LINKPOINTER2 寄存器

偏移地址：4h　复位值：1FFFFFFFh

Zone 1 链接指针 2 寄存器位布局（31 位至 0 位，包含 B0_Z1OTP_LINKPOINTER2 字段）。

表 2-391 B0_Z1OTP_LINKPOINTER2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| B0_Z1OTP_LINKPOINTER2 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| B0_Z1OTP_LINKPOINTER2 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| B0_Z1OTP_LINKPOINTER2 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| B0_Z1OTP_LINKPOINTER2 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |

表 2-392 B0_Z1OTP_LINKPOINTER2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | B0_Z1OTP_LINKPOINTER2 | R | 1FFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone1 链接指针 2 位置。复位类型：SYSRSn |

## 2.15.19.3 B0_Z1OTP_LINKPOINTER3 寄存器

偏移地址：8h　复位值：1FFFFFFFh

Zone 1 链接指针 3 寄存器位布局（31 位至 0 位，包含 B0_Z1OTP_LINKPOINTER3 字段）。

表 2-393 B0_Z1OTP_LINKPOINTER3 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| B0_Z1OTP_LINKPOINTER3 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| B0_Z1OTP_LINKPOINTER3 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| B0_Z1OTP_LINKPOINTER3 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| B0_Z1OTP_LINKPOINTER3 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |

表 2-394 B0_Z1OTP_LINKPOINTER3 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | B0_Z1OTP_LINKPOINTER3 | R | 1FFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone1 链接指针 3 位置。复位类型：SYSRSn |

## 2.15.19.4 Z1OTP_BOOTPIN_CONFIG 寄存器

偏移地址：Ch　复位值：FFFFFFFFh

Zone 1 启动引脚配置寄存器位布局（31 位至 0 位，包含 Z1OTP_BOOTPIN_CONFIG 字段）。

表 2-395 Z1OTP_BOOTPIN_CONFIG 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z1OTP_BOOTPIN_CONFIG | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1OTP_BOOTPIN_CONFIG | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1OTP_BOOTPIN_CONFIG | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1OTP_BOOTPIN_CONFIG | | | | | | | |
| R-FFFFFFFFh | | | | | | | |

表 2-396 Z1OTP_BOOTPIN_CONFIG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z1OTP_BOOTPIN_CONFIG | R | FFFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone1 启动引脚配置寄存器位置。复位类型：SYSRSn |

### 2.15.19.5 Z1OTP_GPREG2 寄存器

偏移地址：Eh　复位值：FFFFFFFFh

Zone 1 通用寄存器 2 位布局（31 位至 0 位，包含 Z1OTP_GPREG2 字段）。

表 2-397 Z1OTP_GPREG2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z1OTP_GPREG2 | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1OTP_GPREG2 | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1OTP_GPREG2 | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1OTP_GPREG2 | | | | | | | |
| R-FFFFFFFFh | | | | | | | |

表 2-398 Z1OTP_GPREG2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z1OTP_GPREG2 | R | FFFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone1 通用寄存器位置。复位类型：SYSRSn |

### 2.15.19.6 Z1OTP_PSWDLOCK 寄存器

偏移地址：10h　复位值：FB7FFFFFh

Zone 1 密码锁定寄存器位布局（31 位至 0 位，包含 Z1OTP_PSWDLOCK 字段）。

表 2-399 Z1OTP_PSWDLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z1OTP_PSWDLOCK | | | | | | | |
| R-FB7FFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1OTP_PSWDLOCK | | | | | | | |
| R-FB7FFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1OTP_PSWDLOCK | | | | | | | |
| R-FB7FFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1OTP_PSWDLOCK | | | | | | | |
| R-FB7FFFFFh | | | | | | | |

表 2-400 Z1OTP_PSWDLOCK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z1OTP_PSWDLOCK | R | FB7FFFFFh | Flash BANK0 的 USER OTP 中的 Zone1 密码锁定位置。复位类型：SYSRSn |

## 2.15.19.7 Z1OTP_CRCLOCK 寄存器

偏移地址：14h　复位值：4BFFFFFFh

Zone 1 CRC 锁定寄存器位布局（31 位至 0 位，包含 Z1OTP_CRCLOCK 字段）。

表 2-401 Z1OTP_CRCLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z1OTP_CRCLOCK | | | | | | | |
| R-4BFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1OTP_CRCLOCK | | | | | | | |
| R-4BFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1OTP_CRCLOCK | | | | | | | |
| R-4BFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1OTP_CRCLOCK | | | | | | | |
| R-4BFFFFFFh | | | | | | | |

表 2-402 Z1OTP_CRCLOCK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z1OTP_CRCLOCK | R | 4BFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone1 CRC 锁定位置。复位类型：SYSRSn |

## 2.15.19.8 Z1OTP_JTAGLOCK 寄存器

偏移地址：18h　复位值：9FFFFFFFh

Zone 1 JTAG 锁定寄存器位布局（31 位至 0 位，包含 Z1OTP_JTAGLOCK 字段）。

表 2-403 Z1OTP_JTAGLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z1OTP_JTAGLOCK | | | | | | | |
| R-9FFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1OTP_JTAGLOCK | | | | | | | |
| R-9FFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1OTP_JTAGLOCK | | | | | | | |
| R-9FFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1OTP_JTAGLOCK | | | | | | | |
| R-9FFFFFFFh | | | | | | | |

表 2-404 Z1OTP_JTAGLOCK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z1OTP_JTAGLOCK | R | 9FFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone1 JTAG 锁定位置。复位类型：SYSRSn |

## 2.15.19.9 Z1OTP_BOOTDEF_LOW 寄存器

偏移地址：1Ch　复位值：FFFFFFFFh

Zone 1 启动定义低 32 位寄存器位布局（31 位至 0 位，包含 Z1OTP_BOOTDEF_LOW 字段）。

表 2-405 Z1OTP_BOOTDEF_LOW 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z1OTP_BOOTDEF_LOW | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1OTP_BOOTDEF_LOW | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1OTP_BOOTDEF_LOW | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1OTP_BOOTDEF_LOW | | | | | | | |
| R-FFFFFFFFh | | | | | | | |

表 2-406 Z1OTP_BOOTDEF_LOW 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z1OTP_BOOTDEF_LOW | R | FFFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone1 启动定义（低）寄存器位置。复位类型：SYSRSn |

## 2.15.19.10 Z1OTP_BOOTDEF_HIGH 寄存器

偏移地址：1Eh　复位值：FFFFFFFFh

Zone 1 启动定义高 32 位寄存器位布局（31 位至 0 位，包含 Z1OTP_BOOTDEF_HIGH 字段）。

表 2-407 Z1OTP_BOOTDEF_HIGH 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z1OTP_BOOTDEF_HIGH | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z1OTP_BOOTDEF_HIGH | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z1OTP_BOOTDEF_HIGH | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z1OTP_BOOTDEF_HIGH | | | | | | | |
| R-FFFFFFFFh | | | | | | | |

表 2-408 Z1OTP_BOOTDEF_HIGH 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z1OTP_BOOTDEF_HIGH | R | FFFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone1 启动定义（高）寄存器位置。复位类型：SYSRSn |

## 2.15.20 DCSM_BANK0_Z2_OTP 寄存器

表中列出了 DCSM_BANK0_Z2_OTP 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址应视为保留位置，且不应修改寄存器内容。

表 2-409 DCSM_BANK0_Z2_OTP 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | B0_Z2OTP_LINKPOINTER1 | Flash BANK0 的 Z2 OTP 中的 Zone 2 链接指针 1 |
| 4h | B0_Z2OTP_LINKPOINTER2 | Flash BANK0 的 Z2 OTP 中的 Zone 2 链接指针 2 |
| 8h | B0_Z2OTP_LINKPOINTER3 | Flash BANK0 的 Z2 OTP 中的 Zone 2 链接指针 3 |
| Ch | Z2OTP_BOOTPIN_CONFIG | 启动引脚配置 |
| Eh | Z2OTP_GPREG2 | Zone-2 通用寄存器-2 内容 |
| 10h | Z2OTP_PSWDLOCK | Z2 OTP 中的安全密码锁定 |
| 14h | Z2OTP_CRCLOCK | Z2 OTP 中的安全 CRC 锁定 |
| 18h | Z2OTP_JTAGLOCK | Z2 OTP 中的安全 JTAG 锁定 |
| 1Ch | Z2OTP_BOOTDEF_LOW | 启动定义（低 32 位） |
| 1Eh | Z2OTP_BOOTDEF_HIGH | 启动定义（高 32 位） |

复杂位访问类型被编码以适应小型表格单元格。表中显示了本节中访问类型所使用的代码。

表 2-410 DCSM_BANK0_Z2_OTP 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | Read Type | 读取类型 |
| R | R | 读取 |
| 复位或默认值 | 复位 or Default Value | 复位或默认值 |
| - n | | 复位后的值或默认值 |
| 寄存器数组变量 | 寄存器 Array Variables | 寄存器数组变量 |
| i,j,k,l,m,n | | 当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | | 当此变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。 |

### 2.15.20.1 B0_Z2OTP_LINKPOINTER1 寄存器

偏移地址：0h　复位值：1FFFFFFFh

Zone 2 链接指针 1 寄存器位布局（31 位至 0 位，包含 B0_Z2OTP_LINKPOINTER1 字段）。

表 2-411 B0_Z2OTP_LINKPOINTER1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| B0_Z2OTP_LINKPOINTER1 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| B0_Z2OTP_LINKPOINTER1 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| B0_Z2OTP_LINKPOINTER1 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| B0_Z2OTP_LINKPOINTER1 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |

表 2-412 B0_Z2OTP_LINKPOINTER1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | B0_Z2OTP_LINKPOINTER1 | R | 1FFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone2 链接指针 1 位置。复位类型：SYSRSn |

## 2.15.20.2 B0_Z2OTP_LINKPOINTER2 寄存器

偏移地址：4h 复位值：1FFFFFFFh

Zone 2 链接指针 2 寄存器位布局（31 位至 0 位，包含 B0_Z2OTP_LINKPOINTER2 字段）。

表 2-413 B0_Z2OTP_LINKPOINTER2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| B0_Z2OTP_LINKPOINTER2 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| B0_Z2OTP_LINKPOINTER2 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| B0_Z2OTP_LINKPOINTER2 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| B0_Z2OTP_LINKPOINTER2 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |

表 2-414 B0_Z2OTP_LINKPOINTER2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | B0_Z2OTP_LINKPOINTER2 | R | 1FFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone2 链接指针 2 位置。复位类型：SYSRSn |

## 2.15.20.3 B0_Z2OTP_LINKPOINTER3 寄存器

偏移地址：8h　复位值：1FFFFFFFh

Zone 2 链接指针 3 寄存器位布局（31 位至 0 位，包含 B0_Z2OTP_LINKPOINTER3 字段）。

表 2-415 B0_Z2OTP_LINKPOINTER3 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| B0_Z2OTP_LINKPOINTER3 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| B0_Z2OTP_LINKPOINTER3 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| B0_Z2OTP_LINKPOINTER3 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| B0_Z2OTP_LINKPOINTER3 | | | | | | | |
| R-1FFFFFFFh | | | | | | | |

表 2-416 B0_Z2OTP_LINKPOINTER3 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | B0_Z2OTP_LINKPOINTER3 | R | 1FFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone2 链接指针 3 位置。复位类型：SYSRSn |

## 2.15.20.4 Z2OTP_BOOTPIN_CONFIG 寄存器

偏移地址：Ch　复位值：FFFFFFFFh

Zone 2 启动引脚配置寄存器位布局（31 位至 0 位，包含 Z2OTP_BOOTPIN_CONFIG 字段）。

表 2-417 Z2OTP_BOOTPIN_CONFIG 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z2OTP_BOOTPIN_CONFIG | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2OTP_BOOTPIN_CONFIG | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2OTP_BOOTPIN_CONFIG | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2OTP_BOOTPIN_CONFIG | | | | | | | |
| R-FFFFFFFFh | | | | | | | |

表 2-418 Z2OTP_BOOTPIN_CONFIG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z2OTP_BOOTPIN_CONFIG | R | FFFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone2 启动引脚配置寄存器位置。复位类型：SYSRSn |

## 2.15.20.5 Z2OTP_GPREG2 寄存器

偏移地址：Eh　复位值：FFFFFFFFh

Zone 2 通用寄存器 2 位布局（31 位至 0 位，包含 Z2OTP_GPREG2 字段）。

表 2-419 Z2OTP_GPREG2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z2OTP_GPREG2 | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2OTP_GPREG2 | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2OTP_GPREG2 | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2OTP_GPREG2 | | | | | | | |
| R-FFFFFFFFh | | | | | | | |

表 2-420 Z2OTP_GPREG2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z2OTP_GPREG2 | R | FFFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone2 通用寄存器位置。复位类型：SYSRSn |

## 2.15.20.6 Z2OTP_PSWDLOCK 寄存器

偏移地址：10h　复位值：BF7FFFFFh

Zone 2 密码锁定寄存器位布局（31 位至 0 位，包含 Z2OTP_PSWDLOCK 字段）。

表 2-421 Z2OTP_PSWDLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z2OTP_PSWDLOCK | | | | | | | |
| R-BF7FFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2OTP_PSWDLOCK | | | | | | | |
| R-BF7FFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2OTP_PSWDLOCK | | | | | | | |
| R-BF7FFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2OTP_PSWDLOCK | | | | | | | |
| R-BF7FFFFFh | | | | | | | |

表 2-422 Z2OTP_PSWDLOCK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z2OTP_PSWDLOCK | R | BF7FFFFFh | USER OTP 中的 Zone2 密码锁定位置。复位类型：SYSRSn |

## 2.15.20.7 Z2OTP_CRCLOCK 寄存器

偏移地址：14h　复位值：0FFFFFFFh

Zone 2 CRC 锁定寄存器位布局（31 位至 0 位，包含 Z2OTP_CRCLOCK 字段）。

表 2-423 Z2OTP_CRCLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z2OTP_CRCLOCK | | | | | | | |
| R-0FFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2OTP_CRCLOCK | | | | | | | |
| R-0FFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2OTP_CRCLOCK | | | | | | | |
| R-0FFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2OTP_CRCLOCK | | | | | | | |
| R-0FFFFFFFh | | | | | | | |

表 2-424 Z2OTP_CRCLOCK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z2OTP_CRCLOCK | R | 0FFFFFFFh | USER OTP 中的 Zone2 CRC 锁定位置。复位类型：SYSRSn |

## 2.15.20.8 Z2OTP_JTAGLOCK 寄存器

偏移地址：18h　复位值：DBFFFFFFh

Zone 2 JTAG 锁定寄存器位布局（31 位至 0 位，包含 Z2OTP_JTAGLOCK 字段）。

表 2-425 Z2OTP_JTAGLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z2OTP_JTAGLOCK | | | | | | | |
| R-DBFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2OTP_JTAGLOCK | | | | | | | |
| R-DBFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2OTP_JTAGLOCK | | | | | | | |
| R-DBFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2OTP_JTAGLOCK | | | | | | | |
| R-DBFFFFFFh | | | | | | | |

表 2-426 Z2OTP_JTAGLOCK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z2OTP_JTAGLOCK | R | DBFFFFFFh | USER OTP 中的 Zone2 JTAG 锁定位置。复位类型：SYSRSn |

## 2.15.20.9 Z2OTP_BOOTDEF_LOW 寄存器

偏移地址：1Ch 复位值：FFFFFFFFh

Zone 2 启动定义低 32 位寄存器位布局（31 位至 0 位，包含 Z2OTP_BOOTDEF_LOW 字段）。

表 2-427 Z2OTP_BOOTDEF_LOW 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z2OTP_BOOTDEF_LOW | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2OTP_BOOTDEF_LOW | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2OTP_BOOTDEF_LOW | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2OTP_BOOTDEF_LOW | | | | | | | |
| R-FFFFFFFFh | | | | | | | |

表 2-428 Z2OTP_BOOTDEF_LOW 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z2OTP_BOOTDEF_LOW | R | FFFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone2 启动定义（低）寄存器位置。复位类型：SYSRSn |

## 2.15.20.10 Z2OTP_BOOTDEF_HIGH 寄存器

偏移地址：1Eh　复位值：FFFFFFFFh

Zone 2 启动定义高 32 位寄存器位布局（31 位至 0 位，包含 Z2OTP_BOOTDEF_HIGH 字段）。

表 2-429 Z2OTP_BOOTDEF_HIGH 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Z2OTP_BOOTDEF_HIGH | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Z2OTP_BOOTDEF_HIGH | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Z2OTP_BOOTDEF_HIGH | | | | | | | |
| R-FFFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Z2OTP_BOOTDEF_HIGH | | | | | | | |
| R-FFFFFFFFh | | | | | | | |

表 2-430 Z2OTP_BOOTDEF_HIGH 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | Z2OTP_BOOTDEF_HIGH | R | FFFFFFFFh | Flash BANK0 的 USER OTP 中的 Zone2 启动定义（高）寄存器位置。复位类型：SYSRSn |

# 3 ROM 和启动引导

## 3.1 简介

本章旨在解释 CPU 内核的引导只读存储器（ROM）代码功能，包括引导过程。本章还讨论了引导 ROM 代码的功能和特性，并详细介绍了 ROM 内存映射的内容。每次复位时，设备会根据复位类型和引导配置在 ROM 中执行引导序列。该序列会初始化设备以运行应用程序代码。对于 CPU 而言，引导 ROM 还包含可用于将应用程序加载到 RAM 中的外围引导加载程序。

表中展示了不同 ROM 的大小。

表 3-1 ROM 空间

| ROM | 大小 |
| --- | --- |
| Unsecure boot ROM | 128 KB |
| Secure ROM | 64 KB |

## 3.2 器件启动引导序列

引导序列描述了每次 CPU 核心复位时的通用引导 ROM 流程。在引导过程中，引导 ROM 代码会更新 RAM 中的引导状态（boot status），该状态详细记录了此过程中执行的操作。

初始化所有通用寄存器与部分 RAM 空间。

时钟配置以及 Flash 上电初始化。

从 OTP（一次性可编程存储器）中加载外设校准参数（trimming）以及设备配置寄存器。

在上电复位时，所有 RAM 都会被初始化。

使能不可屏蔽中断与异常处理机制，并执行 DCSM 的初始化。

使用 OTP 中存储的校准值对指定的外设进行参数校准（trimming）。

判断是否需要通过 GPIO 引脚来确定启动模式。如果需要，则读取启动模式相关的 GPIO 引脚状态，以确定要执行的启动模式。

根据检测到的启动模式及其选项配置，执行相应的启动流程。

## 3.3 器件启动引导模式

本节说明默认启动模式及该设备支持的所有可用启动模式。启动 ROM 通过启动模式选择通用输入/输出（GPIO）引脚来确定启动模式配置。

表中列出了默认引导模式选择引脚可选的引导模式选项。用户可对设备进行编程，以自定义引导表中可选的引导模式及所使用的引导模式选择引脚 GPIO。

该设备所有可用的启动模式均列于表中。

表 3-2 设备默认启动模式

| 启动模式 | GPIO24<br>（默认启动模式选择引脚 1） | GPIO32<br>（默认启动模式选择引脚 0） |
|---|---|---|
| Parallel IO | 0 | 0 |
| SCI / 等待boot (1) | 0 | 1 |
| CAN | 1 | 0 |
| Flash | 1 | 1 |

只要在 SCI 自动波特锁定过程中持续等待‘A’或‘a’，SCI 启动模式即可作为等待启动模式使用。

注意：本章支持的所有外设启动模式均使用外设模块的首个实例（SCIA 、SPIA 、I2CA、CANA 等）。当本章提及这些启动模式时（如 SCI 启动），实际上指的是首个模块实例，即 SCIA 端口上的 SCI 启动。其他外设启动模式亦同。

## 3.4 器件启动引导配置

本节详细说明可用的启动配置及其配置方法。该设备支持从 0 个启动模式选择引脚到 3 个启动模式选择引脚，以及从 1 个配置启动模式到 8 个配置启动模式。

如需将设备从默认设置更改为应用程序的自定义设置，请按照以下步骤操作：

（1）确定应用程序启动时支持的所有不同方式（例如：主应用程序采用闪存启动作为首选启动方式，固件更新采用 CAN 总线启动作为备用启动方式，调试时采用 SCI 总线启动作为第三启动方式等）。

（2）根据所需启动模式数量，确定需要配置多少个启动模式选择引脚（BMSPs）来区分不同启动模式（例如：需要 2 个 BMSPs 来区分 3 种启动模式选项）。

（3）将所需的BMSPs分配到物理 GPIO 引脚（例如：BMSP0分配给GPIO10，BMSP1 分配给 GPIO51，BMSP2 默认保留未启用）。

（4）将确定的启动模式定义分配到自定义启动表中与 BMSPs 解码值对应的索引（例如：BOOTDEF0=闪存启动，BOOTDEF1=CAN 总线启动，BOOTDEF2=SCI 总线启动；其他 BOOTDEFx 保留默认/空值）。

### 3.4.1 配置启动模式引脚

本节说明用户如何通过编程用户可配置双区安全模块（DCSM）的 OTP 中 bootpin-config 位置，自定义启动模式选择引脚。该位置在 DCSM OTP 中为 Z1-OTP-bootpin-config 或 Z2-OTP-bootpin-config。调试时，EMU-bootpin-config 相当于 Z1-OTP-bootpin-config/Z2-OTP-bootpin-config 的仿真配置，可编程实现不同启动模式的测试，无需写入 OTP。设备可根据需求编程使用 0、1、2 或 3 个启动模式选择引脚。

注意：使用 Z2-OTP-bootpin-config 时，该位置的配置优先于 Z1-OTP-bootpin-config 中的配置。建议先使用 Z1-OTP-bootpinconfig，若需修改 OTP 配置，则切换至 Z2-OTP-bootpinconfig。

表 3-3 bootpin-config 位字段

| 位 | 名称 | 描述 |
|---|---|---|
| 31:24 | Key | 在这 8 位中写入 0x5A，以告知引导 ROM (Boot ROM) 代码该寄存器中的位是有效的。 |
| 23:16 | BMSP2 (启动模式选择引脚 2) | 参考 BMSP0 的描述，区别在于此处对应 BMSP2。 |
| 15:08 | BMSP1 (启动模式选择引脚 1) | 参考 BMSP0 的描述，区别在于此处对应 BMSP1。 |
| 7:0 | BMSP0 (启动模式选择引脚 0) | 设置引导期间使用的 GPIO 引脚（最高支持 255）。 |

| 位 | 名称 | 描述 |
|---|---|---|
| | | 0x0 = GPIO0；0x1 = GPIO1，依此类推。 |
| | | 写入 0xFF 会禁用 BMSP0，该引脚将不再用于选择启动模式。 |

以下 GPIO 不可用作 BMSP 。若在特定 BMSP 中选中，引导 ROM 将自动采用出厂默认 GPIO（BMSP2 的出厂默认值为 0xFF，该值禁用 BMSP）。

- GPIO 18 和 GPIO 19
- GPIO 20 和 GPIO 21
- GPIO 36 和 GPIO 38
- GPIO 47 至 GPIO 60
- GPIO 63 至 GPIO 223

表 3-4 单机启动模式选择引脚解码

| BOOTPIN_CONFIG 密钥 | BMSP0 | BMSP1 | BMSP2 | 启动模式 |
|---|---|---|---|---|
| != 0x5A | 无关 | 无关 | 无关 | 按照出厂默认的 BMSP 定义进行启动。 |
| = 0x5A | 0xFF | 0xFF | 0xFF | 按照启动表中定义的启动模式 0 进行启动（禁用所有 BMSP）。 |
| | 有效 GPIO | 0xFF | 0xFF | 由 BMSP0 的值决定启动模式（禁用 BMSP1 和 BMSP2）。 |
| | 0xFF | 有效 GPIO | 0xFF | 由 BMSP1 的值决定启动模式（禁用 BMSP0 和 BMSP2）。 |
| | 0xFF | 0xFF | 有效 GPIO | 由 BMSP2 的值决定启动模式（禁用 BMSP0 和 BMSP1）。 |
| | 有效 GPIO | 有效 GPIO | 0xFF | 由 BMSP0 和 BMSP1 的组合值决定启动模式（禁用 BMSP2）。 |
| | 有效 GPIO | 0xFF | 有效 GPIO | 由 BMSP0 和 BMSP2 的组合值决定启动模式（禁用 BMSP1）。 |
| | 0xFF | 有效 GPIO | 有效 GPIO | 由 BMSP1 和 BMSP2 的组合值决定启动模式（禁用 BMSP0）。 |
| | 有效 GPIO | 有效 GPIO | 有效 GPIO | 由 BMSP0, BMSP1 和 BMSP2 的组合值决定启动模式。 |
| | 无效 GPIO | 有效 GPIO | 有效 GPIO | BMSP0 重置为出厂默认 GPIO；由 BMSP0, BMSP1 和 BMSP2 的值决定启动模式。 |
| | 有效 GPIO | 无效 GPIO | 有效 GPIO | BMSP1 重置为出厂默认 GPIO；由 BMSP0, BMSP1 和 BMSP2 的值决 |

| BOOTPIN_CONFIG 密钥 | BMSP0 | BMSP1 | BMSP2 | 启动模式 |
|---|---|---|---|---|
| | | | | 定启动模式。 |
| | 有效 GPIO | 有效 GPIO | 无效 GPIO | BMSP2 重置为出厂默认状态（即禁用）；由 BMSP0 和 BMSP1 的值决定启动模式。 |

注意：在解码启动模式时，BMSP0 是启动表索引值的最低有效位，BMSP2 是最高有效位。建议禁用 BMSP 时优先禁用 BMSP2。例如，当仅使用 BMSP2（BMSP1 和 BMSP0 被禁用）时，可选择的启动表索引仅为 0 和 4；若仅使用 BMSP0，则可选择的启动表索引为 0 和 1。

### 3.4.2 配置启动模式表选项

本节详解如何配置设备的引导定义表（bootdef）及其相关引导选项。64 位引导定义表位于用户可配置的 DCSM OTP 区域，具体位置为 Z1-OTP-bootdef-LOW 和 Z1-OTP-bootdef-HIGH。调试时，EMU-bootdef-LOW 和 EMU-bootdef-HIGH 是 Z1-OTP-bootdef-LOW 和 Z1-OTP-bootdef-HIGH 的仿真对应项，无需写入 OTP 即可通过编程测试不同引导模式选项。引导定义表的自定义范围取决于所使用的引导模式选择引脚（BMSP）数量：0 个 BMSP 对应 1 个表项，1 个 BMSP 对应 2 个表项，2 个 BMSP 对应 4 个表项，3 个 BMSP 对应 8 个表项。

注：当配置 Z2-OTP-bootpin-config 时，将使用 Z2-OTP-bootdef-LOW 和 Z2-OTP-bootdef-HIGH 位置替代 Z1-OTP-bootdef-LOW 和 Z1-OTP-bootdef-HIGH 位置。

表 3-5 引导程序位字段

| BOOTDEF名称 | 字节位置 | 名称 | 描述 |
|---|---|---|---|
| BOOT_DEF0 | 7:00 | BOOT_DEF0 模式/选项 | 为引导表的索引 0 设置启动模式。<br>不同的启动模式和选项可以包括：例如，为特定的引导加载程序（Bootloader）使用不同的 GPIO，或使用不同的 Flash 入口点地址。任何不支持的启动模式都会导致设备进入“等待引导（Wait Boot）”或从 Flash 启动。 |
| BOOT_DEF1 | 15:08 | BOOT_DEF1 模式/选项 | 参考 BOOT_DEF0 的描述。 |
| BOOT_DEF2 | 23:16 | BOOT_DEF2 模式/选项 | |

| BOOTDEF名称 | 字节位置 | 名称 | 描述 |
|---|---|---|---|
| BOOT_DEF3 | 31:24 | BOOT_DEF3 模式/选项 | |
| BOOT_DEF4 | 39:32 | BOOT_DEF4 模式/选项 | |
| BOOT_DEF5 | 47:40 | BOOT_DEF5 模式/选项 | |
| BOOT_DEF6 | 55:48 | BOOT_DEF6 模式/选项 | |
| BOOT_DEF7 | 63:56 | BOOT_DEF7 模式/选项 | |

### 3.4.3 启动模式示例用例

本节将展示配置启动模式选择引脚及启动模式的若干应用场景。

#### 3.4.3.1 零启动模式选择引脚

该用例展示了一个应用场景：应用程序不使用任何启动模式选择引脚，且设备始终通过闪存启动。

在 OTP 中对 BOOTPIN_CONFIG 寄存器进行如下编程：

- 将 BOOTPIN_CONFIG.BMSP0 设置为 0xFF
- 将 BOOTPIN_CONFIG.BMSP1 设置为 0xFF
- 将 BOOTPIN_CONFIG.BMSP2 设置为 0xFF
- 将 BOOTPIN_CONFIG KEY 设置为 0x5A，使引导 ROM 将这些寄存器位视为有效并使用自定义引导表。

配置设备的 bootdef（引导程序）位置选项。该操作实质上是建立设备专用的引导模式表。

- 将 bootdef.BOOTDEF0 设为 0x03 以实现闪存启动（入口地址选项 0），此时闪存引导将对应引导表索引 0。

表 3-6 零启动引脚启动表结果

| 启动模式表编号 | 启动模式 |
|---|---|
| 0 | Flash Boot (0x03) |

#### 3.4.3.2 一种启动模式选择引脚

该用例展示了一个应用场景：应用程序通过单一启动模式选择引脚，在启动至闪存或采用 CAN 启动之间进行选择。

在 OTP 中编程 BOOTPIN_CONFIG 位位点如下：

- 将 BOOTPIN_CONFIG.BMSP0 设置为用户指定的 GPIO ，例如 GPIO0 的 0x0
- 将 BOOTPIN_CONFIG.BMSP1 设置为 0xFF
- 将 BOOTPIN_CONFIG.BMSP2 设置为 0xFF
- 将 BOOTPIN_CONFIG KEY 设置为 0x5A，使引导 ROM 将这些寄存器位视为有效并使用自定义引导表。

配置设备的 bootdef（引导程序）位置选项。该操作实质上是建立设备专属的引导模式表。具体设置如下：

- 将 bootdef.BOOTDEF0 设为 0x02 以启用 CAN 总线引导，此时引导表索引为 0。
- 将 bootdef.BOOTDEF1 设为 0x03 以启用闪存引导（条目地址选项 0），此时引导表索引为 1。

表 3-7 一个启动引脚启动表结果

| 启动模式表编号 | 启动模式 |
|---|---|
| 0 | CAN Boot (0x02) |
| 1 | Flash Boot (0x03) |

#### 3.4.3.3 三种启动模式选择引脚

本用例展示了一个应用程序通过三个启动模式选择引脚，在自定义启动表中选择不同启动模式的场景。

按以下方式编程 OTP 中的 BOOTPIN_CONFIG 寄存器：

- 将 BOOTPIN_CONFIG.BMSP0 设置为用户指定的 GPIO，例如 GPIO0 的 0x0；
- 将 BOOTPIN_CONFIG.BMSP1 设置为用户指定的 GPIO，例如 GPIO1 的 0x1；
- 将 BOOTPIN_CONFIG.BMSP2 设置为用户指定的 GPIO，例如 GPIO2 的 0x2；

- 将 BOOTPIN_CONFIG KEY 设置为 0x5A，使启动 ROM 将这些寄存器位视为有效并使用自定义启动表。

  编程设置设备的 bootdef 位置选项。这相当于建立设备专属的引导模式表。
- 将 bootdef.BOOTDEF0 设为 0x02 以启用 CAN 引导模式，此时 CAN 引导对应引导表索引 0。
- 将 bootdef.BOOTDEF1 设为 0x03 以启用闪存引导（入口地址选项 0），此时闪存引导对应引导表索引 1。
- 将 bootdef.BOOTDEF2 设为 0x24 以启用等待引导（备用选项），此时等待引导对应引导表索引 2。
- 将 bootdef.BOOTDEF3 设为 0x66 以启用 SPI 引导（备用 GPIO 选项 3），此时 SPI 引导对应引导表索引 3。
- 将 bootdef.BOOTDEF4 设为 0x43 以启用闪存引导（入口地址选项 2），此时闪存引导对应引导表索引 4。
- 将 bootdef.BOOTDEF5 设为 0x09 以启用 USB 引导，此时 USB 引导对应引导表索引 5。

表 3-8 三种引脚的引脚表结果

| 启动模式表编号 | 启动模式 |
|---|---|
| 0 | CAN Boot (0x02) |
| 1 | Flash Boot (0x03) |
| 2 | Wait Boot - Alt (0x24) |
| 3 | SPI - Alt3 (0x66) |
| 4 | Flash Boot - Alt2 (0x43) |
| 5 | USB Boot (0x09) |

## 3.5 器件启动流程图

本节详细说明了独立启动流程与仿真启动流程的流程图。

### 3.5.1 引导流

复位后，CPU 将执行图中所示的启动流程。根据是否连接 JTAG 调试器，CPU 会继续执行仿真启动流程或独立启动流程。

注意：任何同时导致 XRS 复位位被置位的复位操作（POR/BOR 除外）将遵循 XRS 复位的流程。

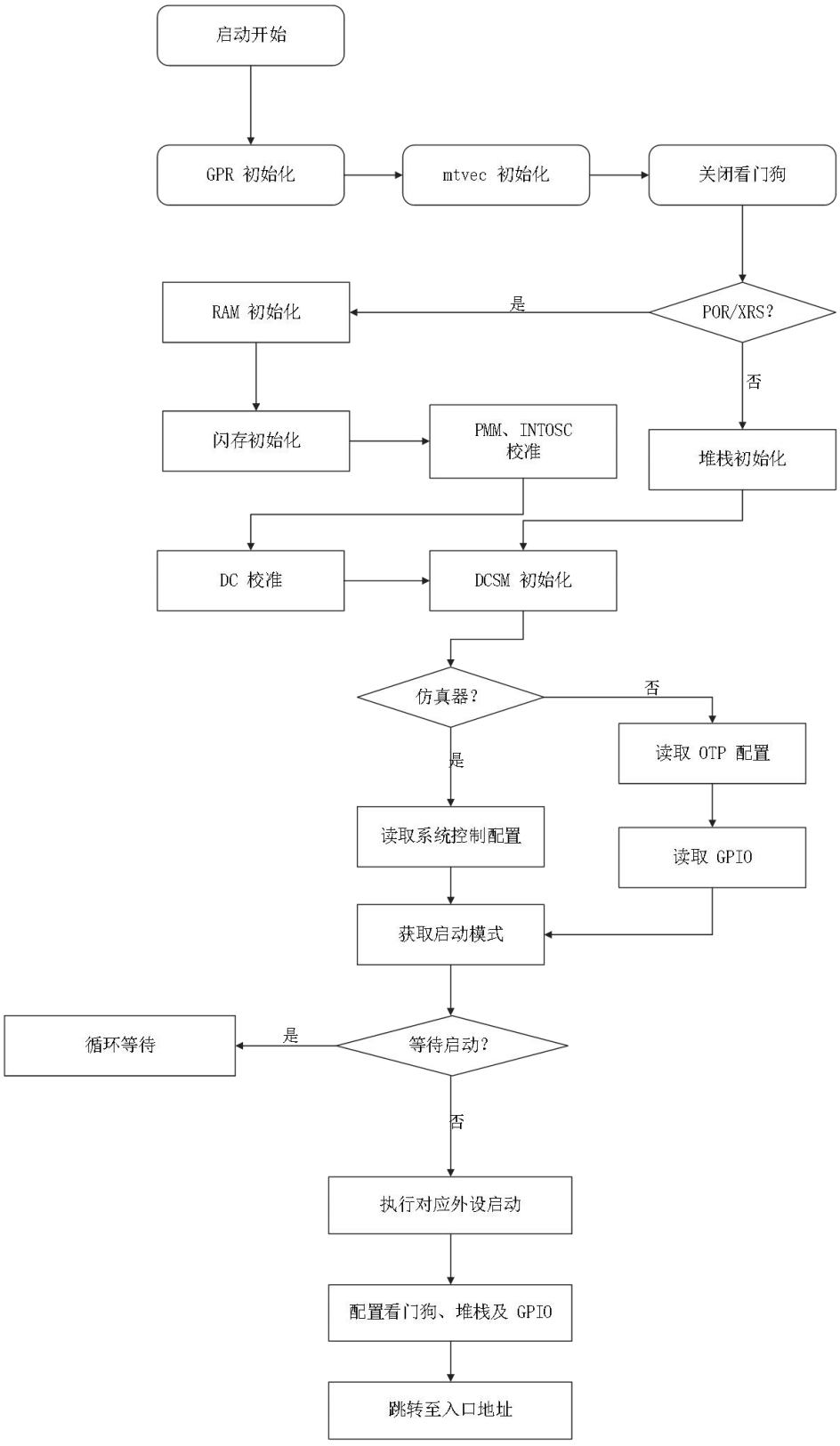

启动开始
GPR 初始化
mtvec 初始化
关闭看门狗
POR/XRS?
是
RAM 初始化
否
闪存初始化
PMM、INTOSC
校准
堆栈初始化
DC 校准
DCSM 初始化
仿真器?
否
是
读取 OTP 配置
读取系统控制配置
读取 GPIO
获取启动模式
循环等待
是
等待启动?
否
执行对应外设启动
配置看门狗、堆栈及 GPIO
跳转至入口地址

图 3-1 设备启动流

## 3.6 器件复位与异常处理

### 3.6.1 复位原因和处理

表中详细说明了不同复位原因下，各引导 ROM 在复位时执行的操作。

表 3-9 系统引导程序重置原因与处理措施

| 复位源 | 引导 ROM 操作 |
| --- | --- |
| 上电复位 (Power on Reset, POR) | 1. 配置分频器 (Clock Divider)<br>2. Flash 上电<br>3. 设备配置与修调<br>4. RAM 初始化<br>5. 继续执行默认引导流程 |
| 外部复位 (External Reset, XRS)<br>包括：<br>看门狗复位 (Watchdog Reset)<br>NMI 看门狗复位<br>SIMRESET XRS | 1. 配置分频器 (Clock Divider)<br>2. Flash 上电<br>3. 设备配置与校准<br>4. 清除用于引导堆栈 (Boot Stack) 的 RAM<br>5. 继续执行默认引导流程 |
| SIMRESET | 1. 清除用于引导堆栈 (Boot Stack) 的 RAM<br>2. 继续执行默认引导流程 |
| 调试器复位 (Debugger Reset) | 1. 清除用于引导堆栈 (Boot Stack) 的 RAM<br>2. 继续执行默认引导流程 |

### 3.6.2 异常和中断处理

引导 ROM 在启动过程中若发生异常时，多数情况下，异常处理会记录错误信息并进入死循环，用户可以通过 escap_point_9 在异常处理程序中补丁修复异常。

若 RAM 出现不可纠正的错误或 ROM 奇偶校验错误，由于系统会执行 RAM 初始化以尝试纠正错误，导致存储在 RAM 中的引导状态信息被清除。由于引导状态信息已被擦除，该异常可通过 NMIWD 复位发生且所有 RAM 均被擦除来识别。

## 3.7 引导程序 ROM 描述

本节详细说明设备启动 ROM 的相关细节。

### 3.7.1 启动程序配置寄存器

引导 ROM 代码涉及多个执行时使用的内存地址和寄存器。系统提供两种配置模式：仿真模式和独立引导流程模式。位于 RAM 中的仿真配置区域可模拟 OTP 配置，且支持多次写入。而用户可配置的 DCSM OTP 区域用于编程设备 OTP，因此仅支持单次写入。表中详细列出了这些配置区域。

此外，引导 ROM 支持 DCSM 1 区和 2 区寄存器的引导配置。2 区配置优先于 1 区配置，因此应优先使用 1 区配置，2 区配置作为备选方案。

表 3-10 启动 ROM 寄存器

| 引导流程 | 寄存器名称 | 引导 ROM 使用名称 | 寄存器地址 | 用户 OTP 地址 |
|---|---|---|---|---|
| 仿真 | - | EMU-BOOTPIN-CONFIG | 0x0065 0374 | - |
| | - | EMU-BOOTDEF-LOW | 0x0065 0378 | - |
| | - | EMU-BOOTDEF-HIGH | 0x0065 037C | - |
| 独立运行 (Using Z1) | Z1-GPREG1 | Z1-OTP-BOOTPIN-CONFIG | 0x0000 5810 | 0x0030 2018 |
| | Z1-GPREG2 | Z1-OTP-BOOT-GPREG2 | 0x0000 5814 | 0x0030 201C |
| | Z1-GPREG3 | Z1-OTP-BOOTDEF-LOW | 0x0000 5818 | 0x0030 2038 |
| | Z1-GPREG4 | Z1-OTP-BOOTDEF-HIGH | 0x0000 5808 | 0x0030 203C |
| 独立运行 (Using Z2) | Z2-GPREG1 | Z2-OTP-BOOTPIN-CONFIG | 0x0000 5890 | 0x0030 2418 |
| | Z2-GPREG2 | Z2-OTP-BOOT-GPREG2 | 0x0000 5894 | 0x0030 241C |
| | Z2-GPREG3 | Z2-OTP-BOOTDEF-LOW | 0x0000 5898 | 0x0030 2438 |
| | Z2-GPREG4 | Z2-OTP-BOOTDEF-HIGH | 0x0000 5888 | 0x0030 243C |

### 3.7.2 入口点

表中详细列出了不同启动模式的入口地址。这些地址会根据所选启动模式，指示引导 ROM 在启动结束时跳转至的目标地址。

表 3-11 入口点地址

| 入口点 | 详细 | 地址 |
|---|---|---|
| Flash 0 | Flash Sector 0 | 0x100000 |
| Flash 1 | Flash Sector 1 | 0x110000 |
| Flash 2 | Flash Sector 2 | 0x120000 |

| 入口点 | 详细 | 地址 |
| --- | --- | --- |
| Flash 3 | Flash Sector 3 | 0x130000 |
| RAM | M1 RAM | 0x1000 |

### 3.7.3 等待点

在引导 ROM 执行过程中，CPU 可能因多种原因陷入代码等待循环。若 CPU 的 PC 寄存器值处于这些状态之一，表中将详细列出对应的地址范围。

表 3-12 等待点地址

| PC 地址 | 说明 |
| --- | --- |
| 0x4028F0 | 处于 Wait Boot |
| 0x4028CE | 处于 Wait Boot |
| 0x403840 - 0x4039AE | 处于 NMI处理程序 |
| 0x403830 | 处于异常处理 |
| 0x401A3A - 0x3A3E | 处于SCI Boot 等待波特率配置 |

### 3.7.4 内存映射

#### 3.7.4.1 启动 ROM 内存映射

表中详细列出了 ROM 内存映射。

表 3-13 内存映射

| 序号 | 类别 | 起始地址(字节) | 分配大小(字节) |
| --- | --- | --- | --- |
| 1 | Boot code | 0x400000 | 0x3800 |
| 2 | Interrupt Handlers | 0x403800 | 0x200 |
| 3 | AES Tables | 0x403A00 | 0x2800 |
| 4 | IQmath Tables | 0x406200 | 0x2CE0 |
| 5 | FPU32 Fast Tables | 0x408EE0 | 0x1040 |
| 6 | FPU32 Twiddle Tables | 0x409F20 | 0x2BF0 |

| 序号 | 类别 | 起始地址(字节) | 分配大小(字节) |
|---|---|---|---|
| 7 | vcu table | 0x40CB10 | 0x1030 |
| 8 | reserved_1 | 0x40DB40 | 0x4C0 |
| 9 | Flash API Library (ROM) | 0x40E000 | 0x6000 |
| 10 | newlib functions | 0x414000 | 0xB500 |
| 11 | reserved_2 | 0x41F500 | 0xAE0 |
| 12 | bootrom version | 0x41FFE0 | 0x10 |
| 13 | bootrom checksum | 0x41FFF0 | 0x10 |

核对数据表以确认这些信息是否适用于您的设备部件编号。若不可用，则将这些部分视为预留。

### 3.7.4.2 保留的 RAM 内存映射

表中详细列出了预留给引导 ROM 使用的 RAM 内存占用情况。这些内存区域应在用户应用程序中预留。

表 3-14 保留的 RAM 内存映射

| 存储器 | 类别 | 起始地址(字节) | 分配大小(字节) |
|---|---|---|---|
| RAM | Boot Status, Flash API data | 0x0 | 0x1000 |

## 3.7.5 启动方式和加载程序

本节详细说明了该设备支持的可用启动模式及引导加载程序。

### 3.7.5.1 启动方式

本节详细说明不涉及外围引导加载程序的可用引导模式。

等待引导

等待启动模式会使 CPU 进入循环状态，不会跳转到用户应用程序代码。设备可通过配置或启动过程中出现错误进入该模式。在使用调试器时采用等待启动模式，以避免 JTAG 相关问题。

导致进入等待启动模式的操作：

- 用户将等待启动设为启动模式

- 未识别启动模式且调试器已连接设备
- 模拟器 BOOTPIN_CONFIG 键值不等于 0xA5 或 0x5A
- 模拟启动过程中发生错误，且启动模式引脚解码值无法识别为有效启动模式

Flash 引导

闪存启动模式会跳转至闪存中配置的内存地址。

RAM 引导

RAM 启动模式将分支至 RAM 中配置的内存地址。

### 3.7.5.2 Bootloaders

本节详细说明采用外围引导加载程序的可用引导模式。

SCI 启动模式

SCI 启动模式异步将代码从 SCI-A 传输至内部存储器。该启动模式仅支持传入的 8 位数据流。

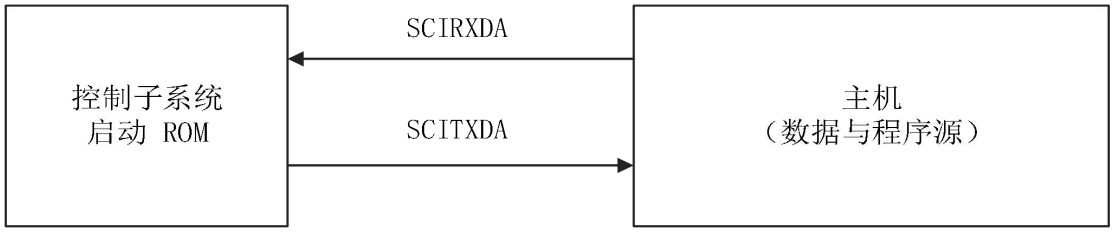



图 3-2 SCI 引导程序运行概述

该设备通过 SCI-A 外设与外部主机进行通信。SCI 端口的自动波特率功能可锁定与主机的波特率。因此，SCI 加载器具有高度灵活性，支持多种波特率与设备通信。

每次数据传输后，引导加载程序会将接收到的 8 位字符回传给主机，使主机能够确认每个字符是否被引导加载程序成功接收。

在较高波特率下，传入数据位的上升时间可能受收发器和连接器性能影响。虽然常规串行通信通常运行良好，但这种上升时间限制会导致在较高波特率（通常超过 100 kbaud）时无法可靠检测自动波特率，并可能使自动波特锁定功能失效。为避免此类问题，建议采取以下措施：

1）通过降低波特率实现主机与 SCI 引导程序之间的波特率锁定。

2）在此低波特率下加载传入的应用程序或自定义加载程序。

3）主机随后可与已加载的应用程序进行握手，将 SCI 波特率寄存器设置为所需的高波特率。

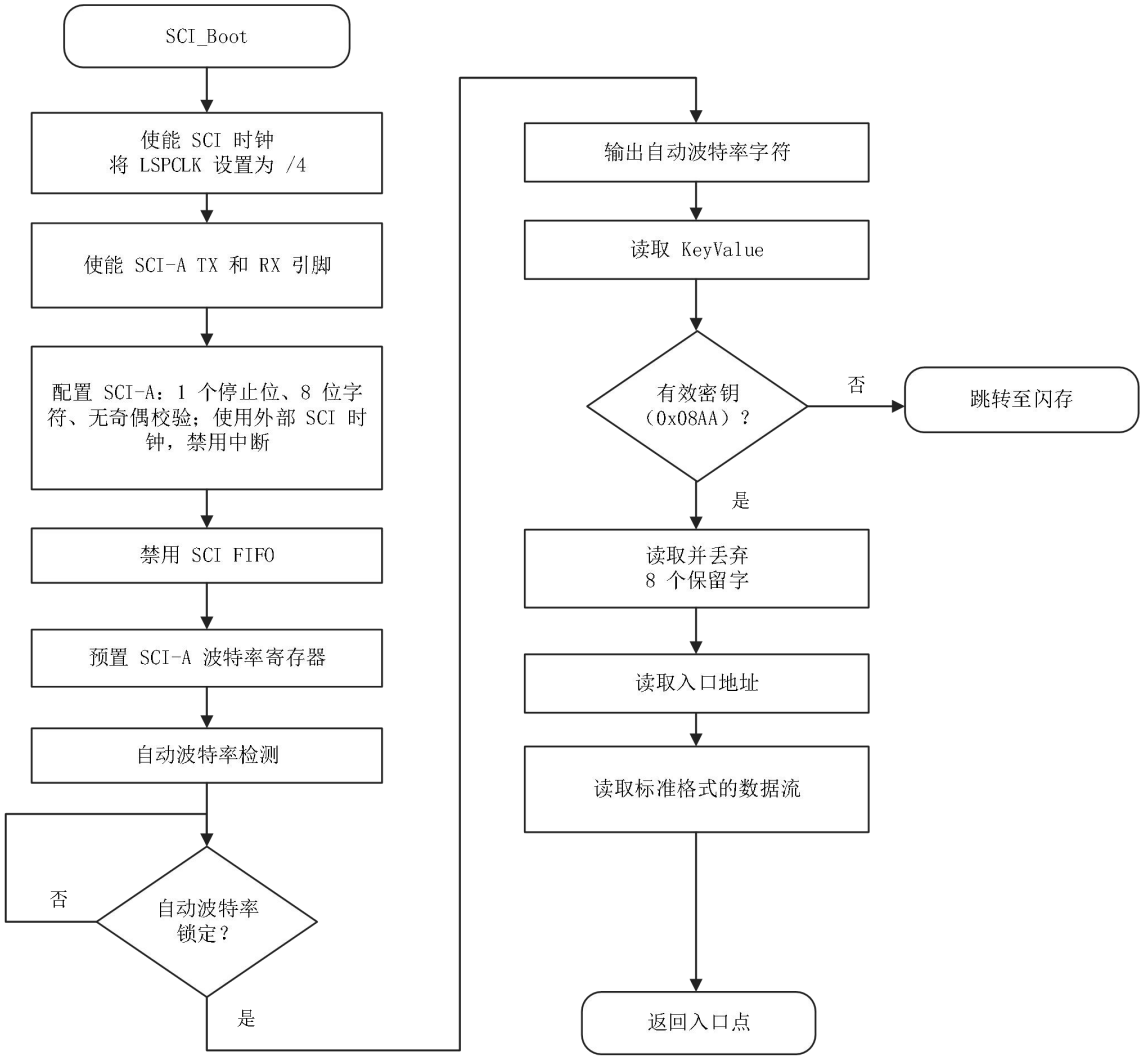



图 3-3 SCI 启动功能概述

SPI 启动模式

SPI 加载器要求 SPI 兼容的 16 位或 24 位可寻址串行 EEPROM 或串行 Flash 设备连接至 SPI-A 引脚，如图所示。SPI 引导加载程序支持 8 位数据流，但不支持 16 位数据流。

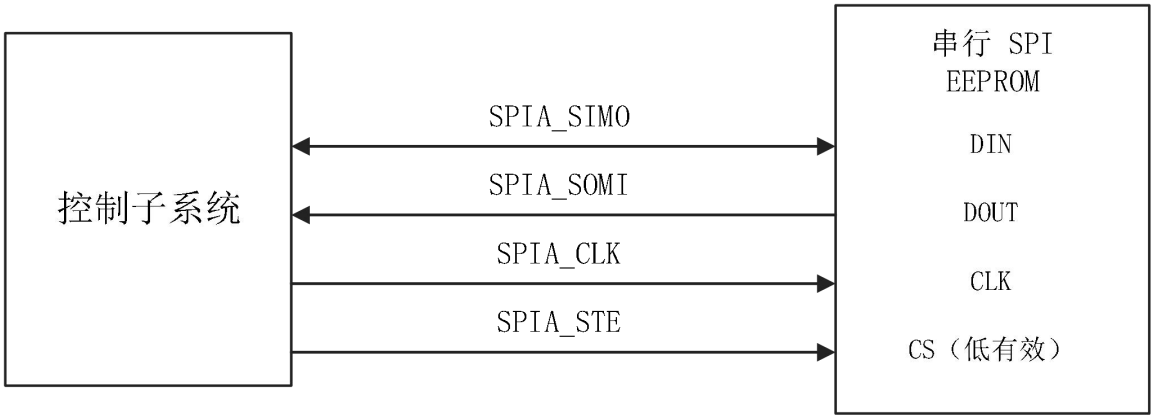



图 3-4 SPI 加载器

SPI 启动 ROM 加载器将 SPI 模块初始化为与串行 SPI EEPROM 或闪存接口。此类设备包括但不限于 Xicor X25320（4Kx8）和 Xicor X25256（32Kx8）SPI 串行 SCLICEPROM，以及 AtmelAT25F1024A 串行闪存。

SPI 启动 ROM 加载器将 SPI 初始化为以下设置：启用 FIFO、8 位字符、内部 SPICLK 主模式和 talk 模式、时钟相位=1、极性=0、使用最慢波特率。

若需通过其他设备的 SPI 端口进行下载，该设备必须设置为从设备模式并模拟串行 SPI EEPROM 。进入 SPI_Boot 功能后，SPI 引脚功能立即被设为主用模式，系统随即启动 SPI 初始化。初始化过程以最慢速度执行。完成 SPI 初始化并读取密钥值后，可指定修改波特率或低速外设时钟。表中展示了 SPI 使用的 8 位数据流。

表 3-15 SPI 8 位数据流

| 字节 | 内容 |
|---|---|
| 1 | LSB: AA (8位位宽校验低位) |
| 2 | MSB: 08h (8位位宽校验高位) |
| 3 | LSB: LOSPCP |
| 4 | MSB: SPIBRR |
| 5 | LSB: (预留将来使用) |
| 6 | MSB: reserved for future use (预留将来使用) |
| ... | ... |
| ... | Reserved (预留) |
| ... | ... |
| 17 | LSB: (预留将来使用) |
| 18 | MSB: (预留将来使用) |
| 19 | LSB: PC[23:16] (入口点 PC 高半字的低位字节) |

| 字节 | 内容 |
| --- | --- |
| 20 | MSB:  PC[31:24] (入口点 PC 高半字的高位字节) |
| 21 | LSB:  PC[7:0] (入口点 PC 低半字的低位字节) |
| 22 | MSB:  PC[15:8] (入口点 PC 低半字的高位字节) |
| ... | ... |
| ... | (本节数据) |
| ... | ... |
| ... | (数据块格式：大小/目标地址/数据) |
| ... | ... |
| ... | (本节数据) |
| n | LSB: 00h (数据流结束标志低位) |
| n+1 | MSB: 00h  (数据流结束标志高位 - 指示数据源结束) |

数据传输采用串行 SPI EEPROM 的‘突发’模式进行，整个传输过程以字节模式完成（SPI 采用 8 位/字符）。具体操作步骤如下：

1）SPI-A 端口初始化

2）根据 SPI 选项定义， GPIO 引脚作为串行 SPI EEPROM 或闪存的芯片选择引脚

3）SPI-A 向串行 SPI EEPROM 发送读取命令

4）SPI-A 向串行 SPI EEPROM 发送地址 0x0000；即主机要求 EEPROM 或闪存中必须存在从地址 0x0000 开始的可下载数据包。该加载器兼容 16 位和 24 位地址

下次获取的字必须与 8 位数据流的键值（0x08AA）匹配。该字的最低有效字节为首次读取的字节，最高有效字节为后续获取的字节。此规则适用于 SPI 上的所有字传输。若键值不匹配，则加载过程终止，引导加载程序跳转至闪存。

接下来读取的两个字节可用于修改低速外设时钟寄存器（LOSPCP）和 SPI 波特率寄存器（SPIBRR）的值。第一个读取的字节是 LOSPCP 值，第二个读取的字节是 SPIBRR 值。接下来的七个字节预留用于后续功能增强。SPI 引导加载程序会读取这七个字节并丢弃它们。

接下来的两个字节构成 32 位入口点地址，执行流程在引导加载过程完成后继续于此地址运行。这通常是通过 SPI 端口下载程序的入口点。

随后，多个代码和数据块会通过 SPI 端口从外部串行 SPI EEPROM 复制到内存中。这些代码块按照之前介绍的标准数据流结构进行组织。这一过程持续进行，直到遇到 0x0000 的块大小。此时，入口点地址会被返回给调用例程，该例程随后退出引导加载程序并在指定地址处恢复执行。

I2C 启动模式

I2C 引导程序要求 I2C-A 总线地址 0x50 处必须存在一个 8 位宽的 I2C 兼容 EEPROM 设备，如图所示。该 EEPROM 必须遵循本节所述的常规 I2C EEPROM 协议，并采用 16 位基地址架构。

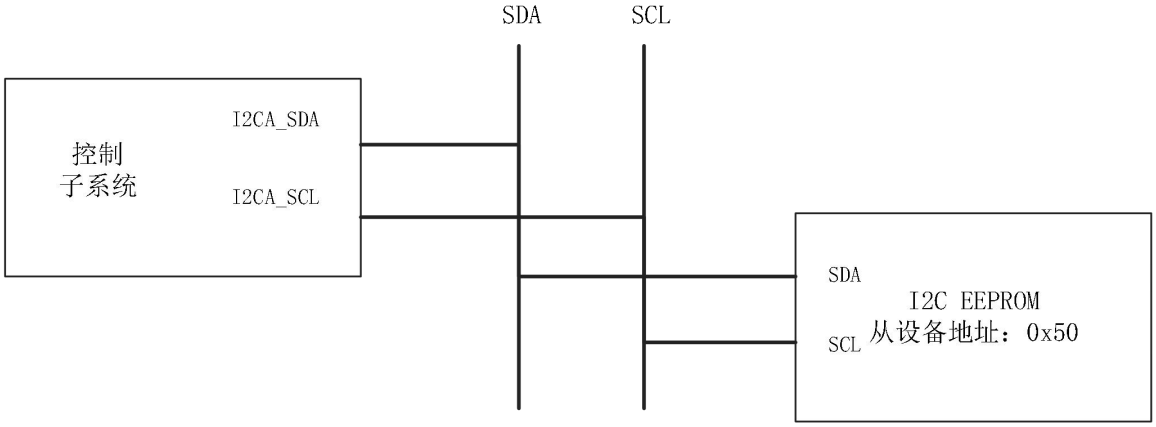



图 3-5 地址 0x50 处的 EEPROM 设备

若需从非 EEPROM 设备下载数据，则该设备必须设置为从设备模式并模拟 I2C EEPROM 。进入 I2C 启动功能后， GPIO 引脚将配置为 I2C-A 模式，同时完成 I2C 初始化。通过 I2C 模块启动时需满足以下要求：

- 设备输入频率须处于指定范围。
- EEPROM 地址必须为从设备地址 0x50。

位周期预分频器（I2CCLKH 和 I2CCLKL）由引导程序配置为在系统时钟为 10MHz 时，以 50%占空比和 100kHz 比特率（标准 I2C 模式）运行 I2C。这些寄存

器在接收来自 EEPROM 的前几个字节后可被修改。这使得在剩余的数据读取期间，通信速率可提升至 400kHz 比特率（快速 I2C 模式）。

仲裁、总线忙和从属信号不被检查。因此，在此初始化阶段不允许其他主设备控制总线。如果应用程序在 I2C 启动模式期间需要另一个主设备，则该主设备必须配置为在应用程序软件信号表明已通过引导程序初始化部分之前，暂停发送任何 I2C 消息。

非确认位仅在发送第一个初始化 EEPROM 基地址的消息时被检查。这是为了确保在继续之前地址 0x50 处存在 EEPROM 。如果地址 0x50 处不存在 EEPROM ，则在数据读取消息的地址阶段（I2C_Get Word）不检查非确认位。如果在数据读取消息期间收到非确认信号，I2C 总线将挂起。表中显示了 I2C 使用的 8 位数据流。

表 3-16 I2C 8 位数据流

| 字节 | 内容 |
|---|---|
| 1 | LSB: AA (8位位宽校验低位) |
| 2 | MSB: 08h (8位位宽校验高位) |
| 3 | LSB: ICPSC[7:0] |
| 4 | Reserved |
| 5 | LSB: I2CCLKH[7:0] |
| 6 | MSB: I2CCLKH[15:8] |
| 7 | LSB: I2CCLKL[7:0] |
| 8 | MSB: I2CCLKL[15:8] |
| ... | ... |
| ... | Reserved (预留) |
| ... | ... |
| 17 | LSB: (预留将来使用) |
| 18 | MSB: (预留将来使用) |
| 19 | LSB: PC[23:16] (入口点 PC 高半字的低位字节) |
| 20 | MSB: PC[31:24] (入口点 PC 高半字的高位字节) |
| 21 | LSB: PC[7:0] (入口点 PC 低半字的低位字节) |
| 22 | MSB: PC[15:8] (入口点 PC 低半字的高位字节) |
| ... | ... |
| ... | (本节数据) |

| 字节 | 内容 |
| --- | --- |
| ... | ... |
| ... | (数据块格式：大小/目标地址/数据) |
| ... | ... |
| ... | (本节数据) |
| n | LSB: 00h (数据流结束标志低位) |
| n+1 | MSB: 00h (数据流结束标志高位 - 指示数据源结束) |

I2C 引导程序所需的 I2C EEPROM 协议如图所示。首次通信将 EEPROM 地址指针设置为 0x0000 并读取 KeyValue（0x08AA）。所有后续读取操作每次读取两个字节。

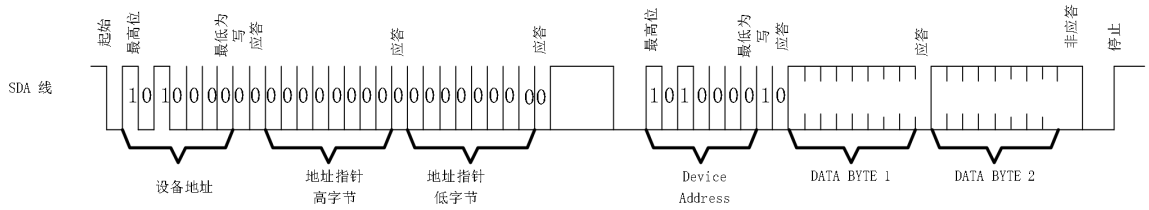



图 3-6 随机读

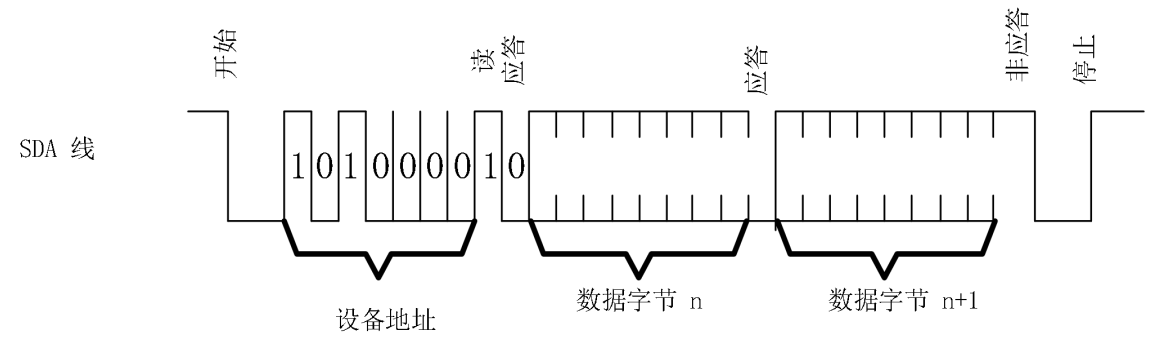



图 3-7 顺序读取

并行启动模式

并行通用输入输出（GPIO）启动模式异步将代码从主机传输至 CPU 内部存储器。每个值为 8 位，其数据流与图所示相同。




图 3-8 并行 GPIO 引导加载程序操作概述

控制子系统通过轮询/驱动主机控制线和 CPU 控制线与外部主机设备通信。如图所示，必须使用握手协议才能通过 GPIO [D0:D7]成功传输每个字。该协议具有极强的鲁棒性，允许主机与主控子系统进行慢速或快速通信。

系统会连续读取两个 8 位字以形成单个 16 位字，其中最低有效字节（LSB）优先读取，随后是最高有效字节（MSB）。

本例中数据从 GPIO [D0:D7]读取，8 位数据流。

表 3-17 并行 GPIO 启动 8 位数据流

| 字节 | | GPIO[D0:D7] (第 1 字节) | GPIO[D0:D7] (第 2 字节) | 描述 |
|---|---|---|---|---|
| 1 | 2 | AA | 8 | 0x08AA (Key，表示内存位宽为 16 位) |
| 3 | 4 | 0 | 0 | 8 个预留字 (第 2 - 9 字) |
| ... | ... | ... | ... | ... |
| 17 | 18 | 0 | 0 | 最后一个预留字 |
| 19 | 20 | BB | 0 | 入口点 PC[22:16] |
| 21 | 22 | DD | CC | 入口点 PC[15:0] (PC = 0x00BBCCDD) |
| 23 | 24 | NN | MM | 待加载的首个数据块大小 = 0xMMNN 字 |
| 25 | 26 | BB | AA | 首个数据块的目标地址 Addr[31:16] |
| 27 | 28 | DD | CC | 首个数据块的目标地址 Addr[15:0] (地址 = 0xAABBCCDD) |
| 29 | 30 | BB | AA | 数据源中待加载首个数据块的第一个字 = 0xAABB |
| ... | | | | ... 本节数据 ... |
| . | | BB | AA | 数据源中待加载首个数据块的最后一个字 = 0xAABB |
| . | | NN | MM | 待加载的第二个数据块大小 = 0xMMNN 字 |
| . | | BB | AA | 第二个数据块的目标地址 Addr[31:16] |
| . | | DD | CC | 第二个数据块的目标地址 Addr[15:0] |
| . | | BB | AA | 待加载的第二个数据块的第一个字 |
| ... | | | | ... |
| n | n+1 | BB | AA | 待加载的最后一个数据块的最后一个字 (如有需要可包含更多节) |
| n+2 | n+3 | 0 | 0 | 块大小为 0000h —— 指示源程序结束 |

该设备首先通过将 CPU 控制引脚拉低，向主机发出准备开始数据传输的信号。随后，主机加载模块通过将 DSP 控制引脚拉低，启动数据传输。完整协议如图所示。

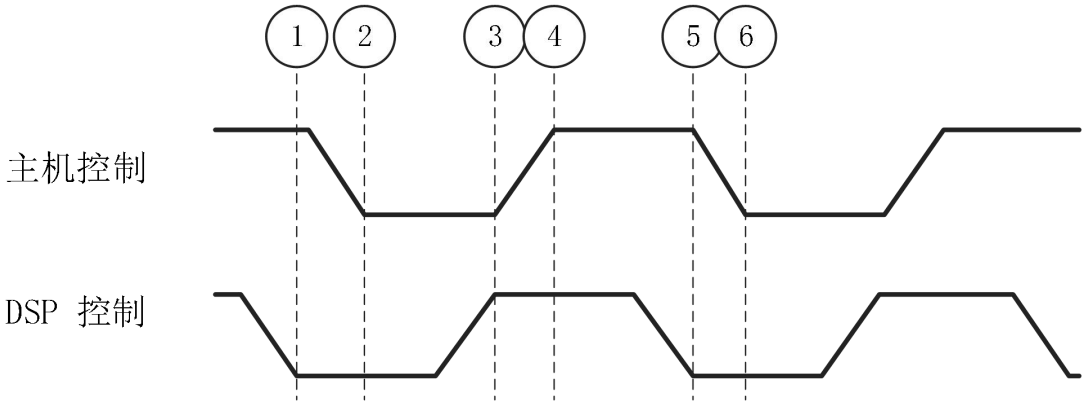



图 3-9 并行 GPIO 引导程序握手协议

1.设备通过将 CPU 控制引脚拉低，表明已准备好开始接收数据。

2.引导程序等待主机将数据写入 GPIO [D0:D7]。主机通过将主机控制引脚拉低，向设备发送数据准备就绪的信号。

3.设备读取数据后，通过将CPU控制引脚拉高，向主机发送读取完成的信号。

4.引导程序等待主机通过将主机控制引脚拉高来确认设备已响应。

5.设备再次通过将 CPU 控制引脚拉低，表明已准备好接收更多数据。

6.该流程会针对每个待发送的数据值重复执行。

7.在此模式下，CPU 与主机的运行速度并不关键，因为主机等待设备，设备等待主机。因此，该协议既适用于主机运行速度快于设备的情况，也适用于主机运行速度慢于设备的情形。

CAN 启动模式

CAN 引导程序异步将代码从 CAN-A 传输至内部存储器。主机可为任意 CAN 节点。通信首先使用 11 位标准标识符（MSGID 为 0x1），每个数据帧采用两个字节。若需要更高数据吞吐量，主机可下载内核以重新配置 CAN。

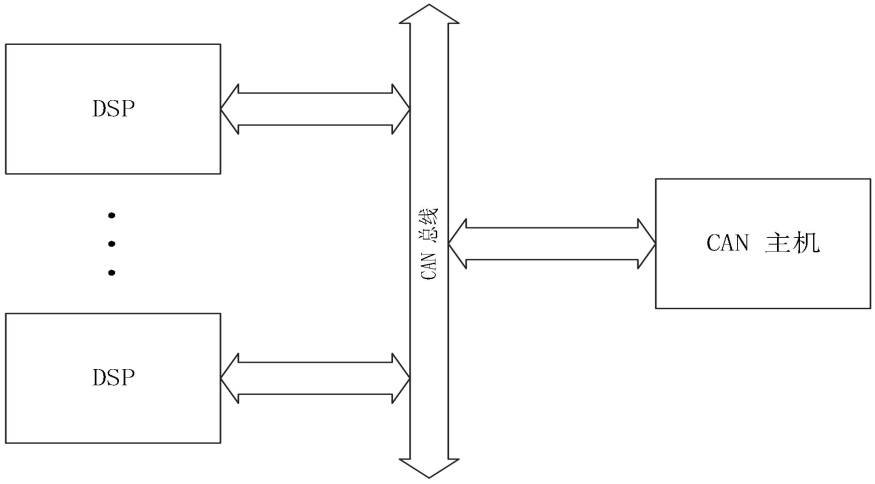



图 3-10 CAN-A 启动程序运行概述

位定时寄存器的编程方式可使 20MHz 外部振荡器实现 100kbps 的比特率，如表所示。

表 3-18 内部振荡器的比特率值

| OSCCLK | SYSCLK | 比特率 |
|---|---|---|
| 20MHz | 10MHz | 100kbps |

显示的 sysclkout 值是采用默认 PLL 设置的复位值。BRP 和位时间值分别硬编码为 10 和 20。

注意：CAN 启动加载程序采用 XTAL 作为位时钟源，INTOSC2 作为系统时钟源。

邮箱 1 的程序设定为标准 MSGID 0x1，用于引导加载程序通信。CAN 主机每次仅能传输两个字节， LSB 先发送 AA， MSB 发送 08。例如，若要向设备发送字 0x08AA，需先发送 AA，再发送 08。CAN 引导加载程序的程序流程与 SCI 引导加载程序完全相同。CAN 引导加载程序的数据序列如表所示。

表 3-19 CAN 8 位数据流

| 字节 | | 第1字节 | 第2字节 | 描述 |
|---|---|---|---|---|
| 1 | 2 | AA | 8 | 0x08AA (Key，表示内存位宽为 16 位) |
| 3 | 4 | 0 | 0 | 8 个预留字 (第 2 - 9 字) |
| ... | ... | ... | ... | ... |
| 17 | 18 | 0 | 0 | 最后一个预留字 |
| 19 | 20 | BB | 0 | 入口点 PC[22:16] |

| 字节 | | 第1字节 | 第2字节 | 描述 |
|---|---|---|---|---|
| 21 | 22 | DD | CC | 入口点 PC[15:0] (PC = 0x00BBCCDD) |
| 23 | 24 | NN | MM | 待加载的首个数据块大小 = 0xMMNN 字 |
| 25 | 26 | BB | AA | 首个数据块的目标地址 Addr[31:16] |
| 27 | 28 | DD | CC | 首个数据块的目标地址 Addr[15:0] (地址 = 0xAABBCCDD) |
| 29 | 30 | BB | AA | 数据源中待加载首个数据块的第一个字 = 0xAABB |
| ... | | | | ... 本节数据 ... |
| . | | BB | AA | 数据源中待加载首个数据块的最后一个字 = 0xAABB |
| . | | NN | MM | 待加载的第二个数据块大小 = 0xMMNN 字 |
| . | | BB | AA | 第二个数据块的目标地址 Addr[31:16] |
| . | | DD | CC | 第二个数据块的目标地址 Addr[15:0] |
| . | | BB | AA | 待加载的第二个数据块的第一个字 |
| ... | | | | ... |
| n | n+1 | BB | AA | 待加载的最后一个数据块的最后一个字 (如有需要可包含更多节) |
| n+2 | n+3 | 0 | 0 | 块大小为 0000h —— 指示源程序结束 |

### 3.7.6 GPIO 分配

本节详细说明位于 Z1-OTP-bootdef-LOW/Z2-OTP-bootdef-LOW 和 Z1-OTP-bootdef-HIGH/Z2-OTP-bootdef-HIGH 的 BOOT_DEF 存储器位置中用于设置启动模式的 GPIO 和启动选项值。在选择启动模式选项时，请确保所选设备封装的引脚多路复用器选项中具备所需引脚。

表 3-20 SCI 启动选项

| 选项 | BOOTDEF 值 | SCITXDA GPIO | SCIRXDA GPIO |
|---|---|---|---|
| 0 (默认) | 0x01 | GPIO29 | GPIO28 |
| 1 | 0x21 | GPIO16 | GPIO17 |
| 2 | 0x41 | GPIO8 | GPIO9 |
| 3 | 0x61 | GPIO2 | GPIO3 |
| 4 | 0x81 | GPIO16 | GPIO3 |

表 3-21 CAN 启动选项

| 选项 | BOOTDEF 值 | CANTXA GPIO | CANRXA GPIO |
|---|---|---|---|
| 0 (默认) | 0x02 | GPIO4 | GPIO5 |
| 1 | 0x22 | GPIO32 | GPIO33 |
| 2 | 0x42 | GPIO2 | GPIO3 |

表 3-22 I2C 启动选项

| 选项 | BOOTDEF 值 | SDAA GPIO | SCLA GPIO |
|---|---|---|---|
| 0 | 0x07 | GPIO32 | GPIO33 |
| 1 | 0x27 | GPIO0 | GPIO1 |
| 2 | 0x47 | GPIO10 | GPIO8 |

表 3-23 RAM 启动选项

| 选项 | BOOTDEF 值 | RAM 入口点地址 |
|---|---|---|
| 0 | 0x05 | 0x1000 |

表 3-24 Flash 启动选项

| 选项 | BOOTDEF 值 | Flash 入口点地址 | Flash 扇区 |
|---|---|---|---|
| 0 (默认) | 0x03 | 0x100000 | Flash Sector 0 |
| 1 | 0x23 | 0x110000 | Flash Sector 1 |
| 2 | 0x43 | 0x120000 | Flash Sector 2 |
| 3 | 0x63 | 0x130000 | Flash Sector 3 |

表 3-25 Wait 启动选项

| 选项 | BOOTDEF 值 | 看门狗 |
|---|---|---|
| 0 | 0x04 | 使能 |
| 1 | 0x24 | 禁用 |

表 3-26 Parallel 启动选项

| 选项 | BOOTDEF 值 | D0-D7 GPIO | DSP 控制 GPIO | 主机控制 GPIO |
|---|---|---|---|---|
| 0 (default) | 0x00 | D0 - GPIO28 | GPIO16 | GPIO29 |
| | | D1 - GPIO1 | | |
| | | D2 - GPIO2 | | |
| | | D3 - GPIO3 | | |
| | | D4 - GPIO4 | | |
| | | D5 - GPIO5 | | |
| | | D6 - GPIO6 | | |
| | | D7 - GPIO7 | | |
| 1 | 0x20 | D0 - GPIO0 | GPIO16 | GPIO11 |
| | | D1 - GPIO1 | | |
| | | D2 - GPIO2 | | |
| | | D3 - GPIO3 | | |
| | | D4 - GPIO4 | | |
| | | D5 - GPIO5 | | |
| | | D6 - GPIO6 | | |

| 选项 | BOOTDEF 值 | D0-D7 GPIO | DSP 控制 GPIO | 主机控制 GPIO |
|---|---|---|---|---|
| | | D7 - GPIO7 | | |

### 3.7.7 启动状态信息

引导 ROM 会记录其运行过程中发生的各类操作和事件。这是因为设备默认启用 NMI（不可屏蔽中断机制）及其他异常处理功能，必须进行相应处理。引导 ROM 将引导状态信息存储在 RAM 中，以便用户应用程序读取该状态，并根据应用需求采取必要措施来处理这些事件。

#### 3.7.7.1 启动状态

启动 ROM 的健康状态和启动状态会被写入 M0RAM 的 32 位地址。该状态会在 POR 或 XRS 复位时被清除，其他复位操作则保留原有状态。例如，在执行调试器设备复位前，应先清除该状态，以便查看状态中反映的最新启动 ROM 操作。

表 3-27 启动状态地址

| 描述 | 地址 |
|---|---|
| Boot ROM 状态 | 0xF00 |

表 3-28 启动状态位字段

| 位 | 描述 |
|---|---|
| 31 | HWBIST复位已成功处理 |
| 30 | Boot ROM 处于等待引导状态 |
| 24 | SYSPLL 已成功启用 |
| 23 | 发生了 UNMI |
| 22 | 发生了时钟丢失 NMI |
| 21 | 发生了 RAM 不可纠正错误 NMI |
| 20 | 发生了 Flash 不可纠正错误 NMI |
| 19 | 发生了 RL NMI |
| 18 | 发生了 ERAD NMI |
| 17 | Boot ROM 检测到外设中断不匹配 |
| 16 | Boot ROM 检测到非法指令中断 |
| 15 | Boot ROM 已完成运行 |
| 13 | Boot ROM 已处理上电复位 (POR) |
| 12 | Boot ROM 已处理外部复位 (XRS) |
| 11 | Boot ROM 已处理所有复位 |
| 10 | POR 内存测试已完成 |

| 位 | 描述 |
| --- | --- |
| 9 | 双代码安全模块初始化已完成 |
| 8 | RAM 初始化完成 |
| 7 | CAN 引导已开始 |
| 6 | I2C 引导已开始 |
| 5 | SPI 引导已开始 |
| 4 | SCI 引导已开始 |
| 3 | RAM 引导已开始 |
| 2 | 并行引导已开始 |
| 1 | Flash 引导已开始 |
| 0 | Boot ROM 已开始运行 |

# 4 闪存控制器（Flash 存储控制器）

## 4.1 Flash 和 OTP 存储器简介

Flash 是一种电可擦除/可编程的非易失性存储器，可多次编程和擦除，以简化代码开发。Flash 存储器主要可用作内核的程序存储器，其次可用作静态数据存储器。

本节描述了配置 Flash 等待状态和操作模式的正确顺序。还包括关于 Flash 和 OTP 电源模式的信息，如何通过启用 Flash 预取/缓存模式来提高 Flash 性能，以及 SECDED 安全特性。

### 4.1.1 特性

闪存的特性包括：

- 一个闪存库（Bank0）
- 一个闪存库 FMC 控制器
- 每次可编程 128 位（存储体位宽）并支持 ECC 校验
- 多扇区设计，可选择保留部分扇区编程而擦除特定扇区
- 用户可编程 OTP 存储单元（用户可配置 DCSM OTP，亦称 USEROTP）用于设置安全参数、OTP 启动模式及启动模式选择引脚（若用户无法使用出厂默认的启动模式选择引脚）
- 通过 FMC 的代码预取机制和数据缓存实现性能优化
- 支持可配置等待状态，根据执行速度动态调整以获得最佳性能
- 安全特性：
  1) SECDED - FMC 支持单比特错误校正和双比特错误检测
  2) ECC 内置地址位
  3) 通过测试模式检测 ECC 逻辑运行状态
- 支持闪存库和泵的低功耗模式以节省电力

- 内置电源模式控制逻辑
- FMC 集成闪存编程/擦除状态机（FSM）
  1) 简单的闪存 API 算法
  2) 快速擦除和编程时间
- 双代码安全模块（DCSM）防止未经授权访问闪存

### 4.1.2 工具

中科本原(FDMTEK)提供以下用于 Flash 的工具：

- CodeCanvas(中科本原 IDE 以下简称 IDE) - 集成了 Flash 插件的开发环境。FDMTEK 建议在使用 IDE 将代码编程到 Flash 后执行调试复位和重新启动。
- Flash API 库 - 一组用于擦除/编程 Flash 的软件外设函数。
- 用户可以使用 IDE "On-Chip Flash"工具。

### 4.1.3 默认 Flash 配置

以下是上电时 Flash 模块的配置设置：

- 专用闪存库和共享泵处于休眠模式（FBFALLBACK 寄存器中的 BNKPWR 位字段）
- ECC 启用
- 等待状态设置为最大值（0xF）
- FMC 的代码预取机制和数据缓存均被禁用
- 库和泵的活动宽限期设置（参见 FBAC 寄存器中的 BAGP 字段）

注意，引导 ROM 会将 BNPWR 位域更改为有效模式。

用户应用程序软件必须使用 FRDCNTL 寄存器初始化等待状态，并使用 FRD_INTF_CTRL 寄存器配置缓存/预取功能，以达到最佳系统性能。配置 Flash 设置（如等待状态、缓存/预取功能等）的软件必须仅从 RAM 存储器执行，而不能从 Flash 存储器执行。

注意：在初始化等待状态之前，请关闭 FRD_INTF_CTRL 寄存器中的预取和数据缓存。

## 4.2 Flash Bank、OTP 和电荷泵

CPU 子系统有一个专用的 Flash Bank。此外，还有一个一次性可编程 (OTP) 存储器，称为 USER OTP，用户只能对其编程一次且无法擦除。Flash 和 OTP 在程序和数据存储器空间中统一映射。

CPU 子系统有一个 FDM-OTP，其中包含制造信息，例如 Flash 状态机用于擦除和编程操作的设置等。用户可以读取 FDM-OTP，但不能对其进行编程或擦除。有关 Flash Banks、FDM-OTP、USER OTP 及其对应 ECC 位置的存储器映射和大小信息，请参考器件数据手册。

## 4.3 Flash 模块控制器 (FMC)

CPU 与 FMC 接口，FMC 进而与 Flash Bank 和电荷泵接口，以执行擦除或编程操作、读取数据以及从 Flash Bank 执行代码。

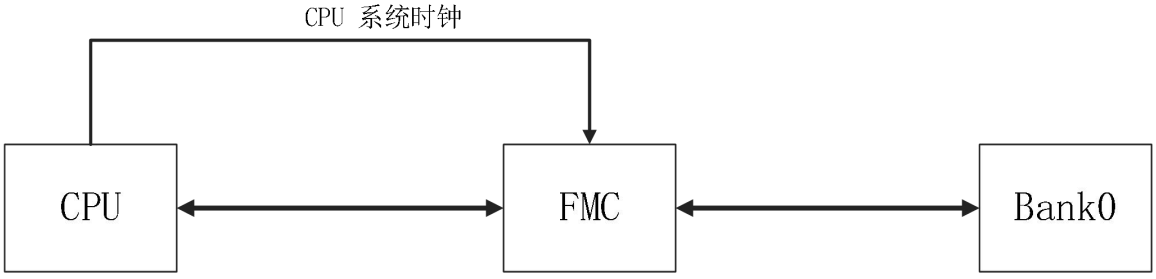



图 4-1 带核、存储库和泵的 FMC 接口

## 4.4 Flash 和 OTP 存储器掉电模式与唤醒

Flash 在工作时会消耗大量电能。闪存模块设有专门的关机机制，可自动关闭闪存库和电荷泵。特殊定时器会自动控制 CPU 闪存库的启动顺序。

闪存库与电荷泵支持三种功耗模式：睡眠模式（最低功耗）、待机模式和工作模式（最高功耗）。

- 休眠状态

这是设备复位后的初始状态。在此状态下，当 CPU 执行数据读取或操作码获取时，会自动切换电源模式：先进入待机状态，再切换至活动状态。在向活动状态过渡期间，CPU 执行会自动暂停。

- 待机状态

该状态耗电量高于休眠状态，但切换至活动或读取状态所需时间更短。此时 CPU 执行数据读取或操作码获取时，会自动切换至活动状态。过渡期间 CPU 执行同样会自动暂停。当闪存/OTP 进入活动状态后，CPU 访问即可正常完成。

- 活动/读取状态

该状态下，存储器阵列处于最高功率的活动模式。

当闪存存储器阵列或 OTP 处于休眠模式时，任何访问操作都会使 Flash 存储器进入激活模式。执行擦除或编程指令时，Flash 会激活。在 Flash 存储器休眠期间，其休眠计数器会存储用户可配置的数值（FPAC1 寄存器中的 PSLEEP 位字段）。当 Flash 存储器退出休眠电源模式时，计数器会经历从 0 到 PSLEEP 预分频 SYSCLK 时钟周期（预分频时钟为 SYSCLK /2）的延迟，随后将 Flash 存储器切换至激活电源模式。

以下是闪存库从低功耗模式唤醒所需的周期数。

1)存储器从休眠状态转为活动状态= PSLEEP ×( SYSCLK /2)个周期

2)存储器从待机状态转为活动状态=55 个闪存时钟周期

式中： 闪存时钟= SYSCLK /( RWAIT +1)

## 4.5 有效缓冲周期

有效缓冲周期 (AGP) 可用于优化 Flash 模块的功耗与访问时间。更快的访问时间与更高的功率操作模式相关联。一方面，电源控制逻辑可以尝试在每次 Flash 访问结束时立即将 Bank 和电荷泵置入低功耗模式以减少功耗。但是，如果访问之间仅间隔几个周期，这实际上可能会增加功耗，而不是让 Flash 保持上电状态，因为在 Flash 启动和访问期间，Bank 和电荷泵消耗更多功率。

有效缓冲周期允许在访问后将 Bank 和电荷泵保持在激活模式一段指定的时间。这样，在 AGP 时间内再次读取时，可获得更短的访问延迟，并减少重复上电带来的能量消耗。如果下一次访问未在 AGP 时间内发生，电源控制逻辑可以自动将 Bank 和电荷泵置入低功耗模式，以在长时间不活动期间减少功耗。

AGP 值由一组可编程计数器（FBAC）编程，这些计数器将 Flash Bank 和电荷泵保持在激活模式，直到计数器超时，此时 Bank 和电荷泵恢复到 FBFALLBACK 中定义的备用电源模式。应用程序软件可以配置备用电源模式以降低功耗，或者将电源模式配置为激活模式以使 Bank 保持激活状态，而与计数器设置无关（默认为 SLEEP）。当 Bank 处于激活状态时，电荷泵 AGP 计数器保持初始化状态，包括 Bank 的 AGP 计数器。当 Bank 变为非激活状态时，电荷泵 AGP 计数器开始计数。

应用程序软件可以通过读取 FBPDRY 寄存器来检查 Flash Bank 和电荷泵的当前电源模式。

## 4.6 Flash 和 OTP 存储器性能

Flash 读取或指令获取访问可以分类为 Flash 访问（访问 Flash 中的地址位置）或 OTP 访问（访问 OTP 中的地址位置）。

当 CPU 执行对 Flash 存储器地址的访问时，数据在 (RWAIT+1) 个 SYSCLK 周期后返回。

对于 USER-OTP 访问，数据在 （RWAIT+1）个 SYSCLK 周期后返回。

RWAIT 定义了随机访问等待状态的数量，并使用 FRDCNTL 寄存器中的 RWAIT 字段进行配置。复位时，RWAIT 默认为最坏情况的等待状态计数 (15)，因此必须根据 CPU 时钟频率和 Flash 的访问时间，将其初始化为适当的等待状态数以改善性能。当 RWAIT 设置为零且 CPU 时钟频率足够低以适应 Flash 访问时间时，Flash 支持零等待访问。

对于给定的系统时钟频率，使用以下公式配置 RWAIT：

对于 Flash Bank：RWAIT = ceiling[(SYSCLK/FCLK)-1]

其中 SYSCLK 是 CPU1 的系统工作频率，FCLK 是 Flash 的时钟频率。

FCLK 必须 ≤ FCLKmax，即在 RWAIT = 0 时允许的最大 Flash 时钟频率。

如果使用上述公式计算 RWAIT 得到小数值，则将 RWAIT 向上舍入到最接近的整数。

## 4.7 Flash 读接口

本节详细介绍了访问 Flash Bank/OTP 的数据读取模式以及控制读取接口的配置寄存器。除了标准读取模式外，FMC 还具有内置的预取和缓存机制，以允许在适用的情况下提高时钟速度和 CPU 吞吐量。

### 4.7.1 RV0025 FMC Flash 读取接口

#### 4.7.1.1 标准读取模式

标准读取模式是复位后默认的 Flash 读取模式。在此模式下，代码预取机制和数据缓存被禁用。当标准读取模式激活时，对 Flash 的每次读取访问都由 Flash 包装器解码，以从寻址位置获取数据，并在 RWAIT+1 个周期后返回数据。

在标准读取模式下，与预取机制和数据缓存关联的 Flash 数据缓冲区被旁路；因此，对 Flash/OTP 的每次访问都立即被 CPU 使用，并且每次访问都会产生一次唯一的 Flash Bank 访问。

标准读取模式是较低系统频率操作的建议模式，此时 RWAIT 可以设置为零以提供单周期访问操作。FMC 可以在更高频率下使用标准读取模式，但代价是增加等待状态。在更高系统频率下，建议启用数据缓存和预取机制以提高性能。请参考器件数据手册，以确定标准读取模式下允许的最大 Flash 频率（即 RWAIT = 0 时的最大 Flash 时钟频率，FCLKMAX）。

#### 4.7.1.2 预取模式

Flash 存储器通常用于存储应用程序代码。在代码执行期间，指令从连续的存储器地址获取，除非发生跳转。通常，驻留在连续地址位置的代码部分构成了应用程序代码的大部分，并被称为线性代码。为了提高线性代码执行的性能，已实现了 Flash 预取机制。图中说明了此模式的工作方式。

预取机制对线性地址增量执行前瞻性预取，从最后一次指令获取的地址开始。Flash 预取机制默认禁用。要启用预取模式，请设置 FRD_INTF_CTRL 寄存器中的 PREFETCH_EN 位，或调用 Flash_enablePrefetch() driverlib 函数。

每次从 Flash 或 OTP 获取指令都会读出 128 位。来自 Flash 的访问起始地址自动对齐到 128 位边界，使得指令位置位于要获取的 128 位之内。当 Flash 预取模式启用时，从指令获取中读取的 128 位存储在一个 128 位宽、2 级深的指令预取缓冲区中。然后根据需要将此预取缓冲区的内容发送给 CPU 进行处理。

最多四个 32 位或八个 16 位指令可以驻留在单个 128 位访问中。大多数 CPU 指令是 32 位的，因此对于从 Flash Bank 的每次 128 位指令获取，预取缓冲区中最多有八条指令准备好供 CPU 处理。在处理这些指令期间，Flash 预取机制自动启动对 Flash Bank 的另一次访问，以预取下 128 位。Flash 预取机制在后台工作，以尽可能保持指令预取缓冲区满。使用这种技术，从 Flash 或 OTP 执行顺序代码的整体效率得到了显著提高。

注意：如果启用了预取机制，并且 Flash Bank 边界之外没有有效的 Flash 存储器地址，则不能使用 Flash Bank 的最后两行（16 个 16 位字，256 位）。这是为了防止 Flash 预取机制尝试从无效地址位置预加载指令缓冲区，从而导致 ECC 错误。

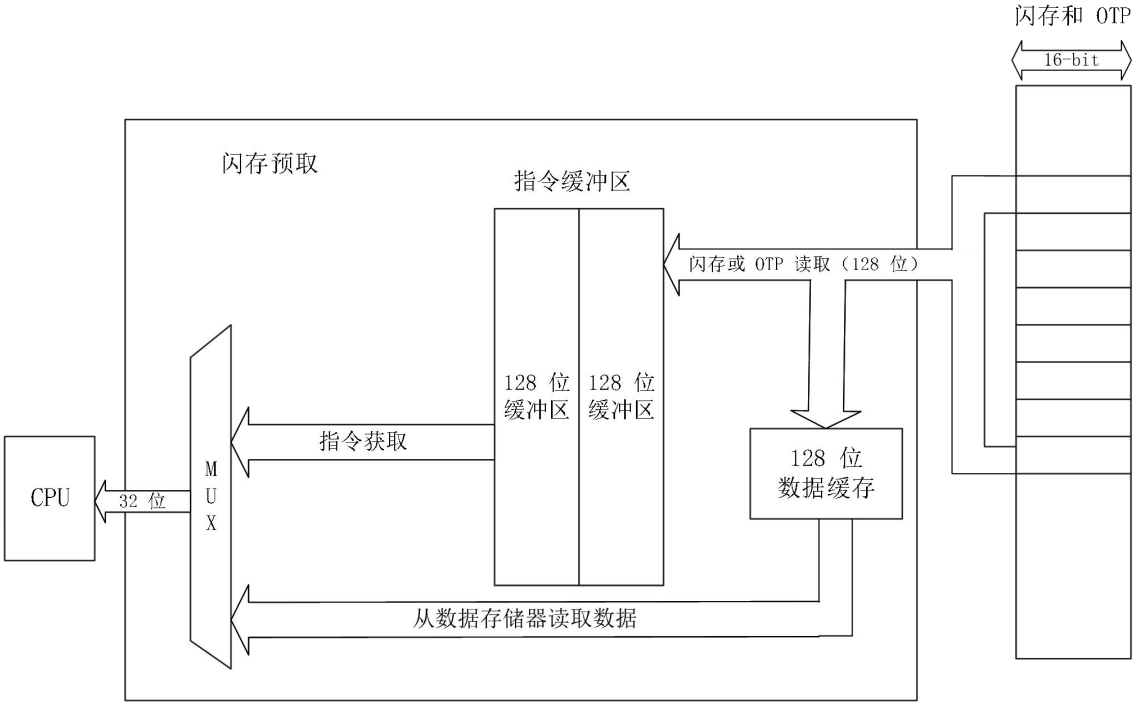



图 4-2 快闪预取方式

Flash 预取仅在执行指令（如分支、函数调用或循环）导致代码不连续时中止。当发生这种情况时，预取机制被中止，并且预取缓冲区的内容被刷新。发生这种情况时有两种可能的情况：

如果目标地址在 Flash 或 OTP 内，则预取中止，然后在目标地址处恢复。

如果目标地址在 Flash 和 OTP 之外，则预取中止，并且仅在代码分支回 Flash 或 OTP 时重新开始。Flash 预取机制仅适用于来自程序空间的指令获取。来自数据存储器和程序存储器的数据读取不使用预取机制，因此旁路预取缓冲区。例如，诸如 MAC、DMAC 和 PREAD 之类的指令从程序存储器读取数据值。当发生此类读取时，预取缓冲区被旁路，但缓冲区不会被刷新。如果在启动数据读取操作时指令预取正在进行中，则数据读取将暂停，直到预取完成。

注意，当 RWAIT 配置为零时，预取机制被旁路。

#### 4.7.1.3 数据缓存

除了预取机制外，还实现了一个 128 位宽的数据缓存，以提高数据空间读取性能。此数据缓存与指令预取缓冲区分开，仅用于数据读取。每当 CPU 对 Flash Bank 地址执行数据读取访问时，如果该地址处的数据当前未加载到数据缓存中，则 Flash 包装器从 Flash Bank 读取 128 位数据，并将数据存储在数据缓存中。该数据最终被发送给 CPU 进行处理。Flash Bank 访问的起始地址自动对齐到 128 位边界，使得请求的地址位置位于要从 Bank 读取的 128 位之内。

数据缓存默认在复位时禁用。要启用数据缓存，请设置 FRD_INTF_CTRL 寄存器中的 DATA_CACHE_EN 位，或调用 Flash_enableCache() driverlib 函数。请注意，当 RWAIT 设置为零时，数据缓存被旁路。

注意：当调试器访问或对 ECC 存储器映射区域执行读取时，数据缓存不会更新。

## 4.8 Flash 擦除和编程

Flash 存储器可以使用 IDE "On_Chip Flash" 插件进行编程。如果这些方法在应用程序中不可行，则可以使用 Flash API。Flash 存储器只能使用 Flash API 库进行编程、擦除和验证。这些函数由中科本原（FDMTEK） 编写、编译和验证。Flash 模块包含一个 Flash 状态机 (FSM) 来执行编程和擦除操作。

推荐的 Flash 编程流程是：擦除 → 编程 → 验证

### 4.8.1 擦除

当目标 Flash 被擦除时，Flash 读取为全 1。这种状态称为“空白”。擦除函数必须在编程之前执行。用户不能跳过对读取为“空白”的扇区的擦除，因为这些扇区可能由于列位边缘擦除不彻底而需要额外擦除。FSM 提供了一个擦除扇区命令来擦除目标扇区。擦除函数会一起擦除数据和 ECC。

### 4.8.2 编程

Flash 控制器提供了一个命令来编程 Flash 和用户 OTP。此命令也用于编程 ECC 校验位。

注意：主阵列 Flash 编程必须对齐到 64 位地址边界，并且每个 64 位字在每个写/擦除周期中只能编程一次。

DCSM OTP 编程必须对齐到 128 位地址边界，并且每个 128 位字只能编程一次。例外情况是：

- DCSM OTP 中的 DCSM Zx-LINKPOINTER1 和 Zx-LINKPOINTER2 值可以一起编程，并且可以根据 DCSM 操作的需要一次编程一位。
- DCSM OTP 中的 DCSM Zx-LINKPOINTER3 值可以根据 DCSM 操作的需要一次编程一位。

### 4.8.3 校验

编程后，用户必须使用 API 函数 Fapi_doVerify() 执行验证。此函数验证 Flash 内容与提供的数据是否一致。

应用程序软件通常在上电期间和运行时定期（根据需要）对 Flash 存储器内容执行 CRC 检查。除此之外，当启用 ECC 逻辑时（默认启用），只要 CPU 从 Flash 地址读取/获取，它就会捕获单比特错误、双比特错误和地址错误。

## 4.9 错误校正码 (ECC) 保护

FMC 包含一个嵌入式的单比特错误校正和双比特错误检测 (SECDED) 模块。SECDED 启用后，能够筛选出存储器故障。SECDED 可以检测和校正单比特数据错误，并检测地址错误/双比特数据错误。对于每 64 位 Flash/OTP 数据（对齐在 64 位存储器边界上），必须计算八个 ECC 校验位并将其编程到 ECC 存储器空间中。有关 Flash/OTP ECC 存储器映射，请参考器件数据手册。SECDED 使用与每个 64 位宽数据字和对应的 128 位存储器对齐地址相关联的、总共八位用户计

算的错误校正码 (ECC) 校验位。用户必须将 ECC 校验位与 Flash 数据一起编程。FDMTEK 建议使用 API 中可用的 AutoEccGeneration 选项来编程 ECC。用户可以使用 Flash API 来计算 ECC 数据并将其与 Flash 数据一起编程。Flash API 使用器件中的硬件 ECC 逻辑为给定的 Flash 数据生成 ECC 数据。与 CodeCanvas (IDE) 集成的 Flash 编程工具，使用 Flash API 生成和编程 ECC 数据。

在指令获取或数据读取操作期间，19 个最高有效地址位（地址的三个最低有效位不考虑），连同从 Flash Banks/ECC 存储器映射区域读出的 64 位数据/8 位 ECC，通过 SECDED 逻辑，并在 FMC 中产生八个校验位。然后将这八个计算出的 ECC 校验位与和地址及读取数据相关联的存储校验位（用户编程的校验位）进行异或运算。8 位输出在 SECDED 模块内部被解码，以确定三种情况之一：

- 未发生错误
- 发生可纠正错误（单比特数据错误）
- 发生不可纠正错误（双比特数据错误或地址错误）

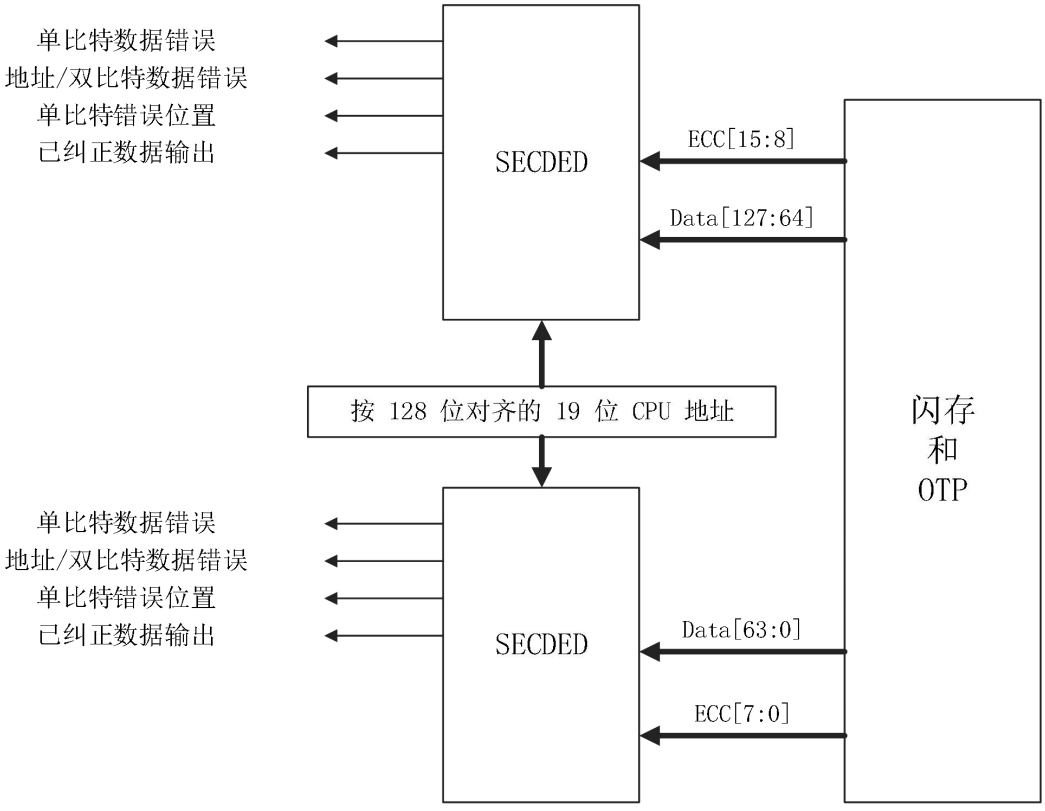



图 4-3 ECC 逻辑输入与输出

如果 SECDED 逻辑在地址字段中发现单比特错误，则该错误被视为不可校正错误。

注意：由于 ECC 是针对整个 64 位数据计算的，因此非 64 位读取（如字节读取或半字读取）仍然会强制读取和计算整个 64 位数据，但 CPU 实际只使用字节或半字。

此 ECC (SECDED) 功能在复位时启用。ECC_ENABLE 寄存器可用于配置（启用/禁用）ECC 功能。必须为应用程序代码编程 ECC。每个 FMC 中有两个 SECDED 模块。从 Bank/OTP 地址读取的 128 位数据（对齐在 128 位存储器边界）中，较低的 64 位数据和相应的 8 位 ECC（从用户可编程的 ECC 存储器区域读取）与读取数据时使用的、128 位对齐的 19 位地址一起，作为一个 SECDED 模块的输入。较高的 64 位数据和相应的 8 位 ECC 与 128 位对齐的 19 位地址一起，并行馈入另一个 SECDED 模块。每个 SECDED 模块评估输入，并确定是否存在任何单比特数据错误或双比特数据错误/地址错误。

当从 Bank 获取的 64 位数据及其关联的 ECC 位全为 1 或全为 0 时，ECC 逻辑被旁路。

### 4.9.1 单比特数据错误

本节提供有关单比特数据错误和单比特 ECC 校验位错误的信息。如果 Flash 数据或 ECC 数据中发生单比特翻转（0 到 1 或 1 到 0），则该错误被视为单比特数据错误。SECDED 模块在将读取数据提供给 CPU 之前，检测并校正（如果存在）64 位 Flash 数据或从 Flash/ECC 存储器映射读取的八个 ECC 校验位中的单比特错误。

当 SECDED 发现并校正单比特数据错误时，如果启用了 ECC 功能，以下信息将记录在 ECC 寄存器中：

- 错误发生地址：若单比特错误出现在 128 位内存对齐闪存数据字的低 64 位中，低 64 位数据字的地址将被记录在 SINGLE_ERR_ADDR_LOW 寄存器中；若单

比特错误出现在 128 位数据字的高 64 位中，则高 64 位数据字的地址将被记录在 SINGLE_ERR_ADDR_HIGH 寄存器中。

- 错误类型：ERR_POS 寄存器中的 ERR_TYPE_L 和 ERR_TYPE_H 位字段分别指示错误发生在 128 位内存对齐闪存数据字的低 64 位/低 8 位 ECC 数据位，还是高 64 位/高 8 位 ECC 数据位。
- 错误位位置：ERR_POS 寄存器中的 ERR_POS_L 和 ERR_POS_H 位字段分别指示错误发生在 128 位内存对齐闪存数据字的低 64 位/低 8 位 ECC 数据位，还是高 64 位/高 8 位 ECC 数据位。
- 校正值是否为 0（FAIL_0_L ，ERR_STATUS 寄存器中的 FAIL_0_H 标志位）。
- 校正值是否为 1（FAIL_1_L ，ERR_STATUS 寄存器中的 FAIL_1_H 标志位）。
- 单比特错误计数器（ERR_CNT 寄存器），每当发生单比特错误时递增，直至达到用户可配置的阈值（参见 ERR_THRESHOLD）。
- 当 ERR_CNT 等于 ERR_THRESHOLD 后出现一个或多个单比特错误时，该标志位会被设置（ERR_INTFLG 寄存器中的 SINGLE_ERR_INT_FLG 标志位）。

当 ERR_CNT 值等于 ERR_THRESHOLD+1 且发生单比特错误时，闪存模块会设置 SINGLE_ERR_INT 标志并触发中断信号。为使生成的中断脉冲能传递至 CPU，用户应用程序需在 RV0025 的中断控制模块 clic 中启用 Intr_Flash_correctable_error 通道。该中断信号将持续保持高电平，直至应用程序通过写入 ERR_INTCLR 寄存器中的 SINGLE_ERR_INTCLR 位来清除 SINGLE_ERR_INTFLG 标志。由于该中断属于边沿触发式中断，因此在 SINGLE_ERR_INTFLG 被清除前，闪存模块无法向 clic/CPU 生成任何后续的 Intr_Flash_correctable_error 中断信号。

当 ECC 逻辑检测到多个单比特错误时，Flash ECC 寄存器的内容会反映最近发生的 ECC 错误。若检测到多个单比特错误， FAIL_0_L 和 FAIL_1_L（或 FAIL_0_H 和 FAIL_1_H）均会被置位，表明单比特 fail0/fail1 错误发生在不同的 64 位对齐地址。

虽然 ECC 校验基于 64 位进行计算，但若在 128 位对齐的 Flash 数据字中任一地址位置读取时，若该 128 位数据字的低 64 位或高 64 位（或对应的 ECC 校验位）中存在单比特错误，则会触发单比特错误标志。

### 4.9.2 不可校正错误

不可校正错误包括地址错误以及数据或 ECC 中的双比特错误。当 ECC 逻辑发现不可校正错误时，如果启用了 ECC 功能，以下信息将记录在 ECC 寄存器中：

- 错误发生地址记录：若不可纠正错误出现在 128 位闪存对齐数据字的低 64 位地址中，系统会将该地址捕获至 UNC_ERR_ADDR_LOW 寄存器；若错误出现在高 64 位地址中，则记录于 UNC_ERR_ADDR_HIGH 寄存器。
- 错误标志位设置：ERR_STATUS 寄存器中的 UNC_ERR_L 和 UNC_ERR_H 标志位分别指示不可纠正错误发生在 128 位闪存对齐数据字的低 64 位/低 8 位 ECC 校验位，或高 64 位/高 8 位 ECC 校验位。
- 错误中断标志位设置：ERR_INTFLG 寄存器中的 UNC_ERR_INTFLG 标志位用于记录不可纠正错误中断的发生。

当发生不可纠正错误时，闪存模块会设置 UNC_ERR_INTFLG 位并触发不可纠正错误中断。若启用该中断，CPU 将生成不可屏蔽中断（NMI）。若未通过写入 ERR_INTCLR 寄存器中的 UNC_ERR_INTCLR 位清除不可纠正错误中断标志，闪存模块将无法生成新的不可纠正中断信号，因为该中断属于边沿触发式。

虽然 ECC 校验基于 64 位计算，但当读取 128 位对齐闪存字中的任意地址时，若该 128 位数据字的低 64 位或高 64 位（或对应的 ECC 校验位）中存在不可纠正错误，将触发不可纠正错误标志并产生不可纠正错误中断/NMI。

### 4.9.3 SECDED 逻辑正确性检查

由于错误检测和校正逻辑是安全关键逻辑的一部分，安全应用通常需要确保 SECDED 逻辑始终正常工作。出于这些安全考虑，为了确认 SECDED 逻辑的正

确性，提供了一种 ECC 测试模式来定期测试 ECC 逻辑的正确性。在此 ECC 测试模式下，ECC 逻辑的数据/ECC 和地址输入分别由 ECC 测试模式寄存器 FDATAH_TEST、FDATAL_TEST、FECC_TEST 和 FADDR_TEST 控制。使用此测试模式，用户可以引入单比特错误、双比特错误或地址错误，并检查 SECDED 逻辑是否捕获到这些错误。用户还可以检查在没有引入错误时 SECDED 逻辑是否报告任何虚假错误。

可以通过设置 FECC_CTRL 寄存器中的 ECC_TEST_EN 位来启用此 ECC 测试模式。当启用 ECC 测试模式时，CPU 无法从 Flash 读取数据。相反，CPU 从 ECC 测试模式寄存器（FDATAH_TEST/FDATAL_TEST）获取数据。这是因为 ECC 测试模式寄存器（FDATAH_TEST、FDATAL_TEST、FECC_TEST）与来自 Flash 的数据是多路复用的。因此，ECC 测试模式代码必须从 RAM 执行，而不能从 Flash 执行。

一次只能测试一个 SECDED 模块（两个 SECDED 模块中的一个，分别处理读取的 128 位数据的较低 64 位和较高 64 位）。用户可以通过配置 FECC_CTRL 寄存器中的 ECC_SELECT 位来选择要测试的 SECDED 模块之一。

要使用 ECC 测试模式测试 ECC 逻辑，请遵循以下步骤：

1)通过 Flash API 提供的自动 ECC 生成功能，获取指定 Flash 地址（128 位对齐）和 64 位数据的 ECC 校验码。

2)开发一个应用程序，利用上述数据测试 ECC 逻辑。该应用程序需满足以下要求：

- 将 128 位对齐的 19 位 Flash 地址写入 FADDR_TEST 寄存器。
- 将 64 位数据写入 FDATAx_TEST（FDATAL_TEST 和 FDATAH_TEST）寄存器。
- 将对应的 8 位 ECC 写入 FECC_TEST 寄存器。

- 在上述三个步骤中的任意一步，应用程序可插入错误（单比特数据错误、双比特数据错误、地址错误、单比特 ECC 错误或双比特 ECC 错误），以验证 ECC 逻辑是否能检测这些错误。
- 通过 FECC_CTRL 寄存器中的 ECC_SELECT 位选择需要测试的 ECC 逻辑块（低 64 位或高 64 位）。
- 通过 FECC_CTRL 寄存器中的 ECC_TEST_EN 位启用 ECC 测试模式。
- 在 FECC_CTRL 寄存器的 DO_ECC_CALC 位写入 1，以启用单周期 ECC 测试逻辑，对 FADDR_TEST、FDATAx_TEST 和 FECC_TEST 寄存器中的地址、数据及 ECC 进行 ECC 错误评估。

当用户完成上述 ECC 测试模式寄存器的写入操作后：

- FECC_OUTOUT 寄存器存储待测 SECDED 块的输出数据位 63:32。
- FECC_OUTOUT 寄存器存储待测 SECDED 块的输出数据位 31:0。
- FECC_STATUS 寄存器记录数据/校验位中单比特错误的发生状态、不可纠正错误的发生情况以及单比特错误的位置。

## 4.10 Flash 和 OTP 存储器内的保留位置

在为 Flash 和 OTP 存储器分配代码与数据时，请注意以下预留位置：

- OTP 整体预留了用户可配置的安全区域和启动流程区域。
- 闪存启动入口点专为闪存分支指令预留，当启用闪存启动选项时，引导 ROM 会跳转至该地址。若用户在此处编程分支指令，代码执行将被重定向至应用程序的入口点。

## 4.11 将应用从 RAM 迁移到 Flash

若要将现有应用程序从内存运行配置迁移至基于闪存的链接器配置，请按以下步骤操作：

将 RAM 链接器命令文件替换为 Flash 链接器命令文件。

修改基于 Flash 的链接器命令文件时，务必将所有初始化区域映射到 Flash 存储器区域。

确保引导模式引脚配置为 Flash 引导模式。该设置会指示引导 ROM 在完成引导代码执行后，将执行重定向至 Flash 存储器中编程的应用程序。

当设备配置为 Flash 引导模式时，引导 ROM 会在引导代码执行结束时将执行重定向至 Flash 入口点位置。请确保在 Flash 入口点处存在指向代码初始化函数的分支指令。

为实现 Flash 执行的最佳性能，需根据设备数据手册中规定的时钟频率配置 Flash 等待状态。同时启用预取模式和数据缓存模式。调用 Flash_initModule()驱动程序库函数即可完成这些设置。需注意，初始化 Flash 模块的代码必须在 RAM 地址执行，可通过将 Flash 初始化函数分配至 ram_funcs 段来实现。在链接器命令文件中，需将该段映射至 Flash 内存进行加载，并映射至 RAM 执行。

对于需要 0 或 1 等待状态性能的函数，需在链接器命令文件中将其映射至 RAM 执行，操作方式与 Flash 初始化函数类似。提供的 Flash 链接器命令文件中的 ram_funcs 段可实现此功能。

在映射至 Flash 内存时，使用链接器命令文件中的 align 指令，将所有代码和数据段对齐至 128 位地址边界。

确保为 Flash 应用镜像正确配置 ECC 位。在 CodeCanvas Flash 工具中保持 AutoEccGeneration 选项启用。

## 4.12 更改 Flash 控制寄存器的步骤

在 Flash 配置期间，不能有任何对 Flash 或 OTP 的访问正在进行。这包括仍在 CPU 流水线中的指令、数据读取和指令预取操作。为确保在配置更改期间不发生访问，对于任何修改 Flash 控制寄存器的代码，请遵循下面所示的步骤。

从 RAM/Flash/OTP 启动应用程序代码执行。

转入或调用 RAM 中的 Flash 配置代码（该代码用于写入 Flash 控制寄存器）。此操作是配置变更前清空 CPU 流水线的必要步骤。修改 Flash 配置的功能模块不能直接从 Flash 或 OTP 执行，必须驻留在 RAM 中。

执行位于 RAM 中的 Flash 配置代码，该代码会写入诸如 FRDCNTL、FRD_INTF_CTRL 等 Flash 控制寄存器。

在 Flash 配置代码执行完毕后，等待八个时钟周期使写入指令在 CPU 流水线中传播。此操作必须在执行函数返回前完成。

返回驻留在 RAM 或 Flash/OTP 中的调用函数并继续执行。

## 4.13 软件

### 4.13.1 FLASH 示例

详见例程库。

## 4.14 寄存器描述

### 4.14.1 FLASH 基地址表

表 4-1 FLASH 基地址表

| 位字段实例 | 位字段结构 | DriverLib 名称 | 基地址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| Flash0CtrlRegs | FLASH_CTRL_REGS | FLASH0CTRL_BASE | 0x0002F_F800 | - | - | 是 |
| Flash0EccRegs | FLASH_ECC_REGS | FLASH0ECC_BASE | 0x0002F_F900 | - | - | 是 |

### 4.14.2 FLASH_CTRL_REGS 寄存器

表中列出了 FLASH_CTRL_REGS 的存储器映射寄存器。表中未列出的寄存器偏移地址均应视为保留位置，不应修改这些位置的寄存器内容。

表 4-2 FLASH_CTRL_REGS 寄存器

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 20h | FRDCNTL | Flash 读取控制寄存器 |
| 24h | FBAC | Flash 存储体访问控制寄存器 |

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 28h | FBFALLBACK | Flash 存储体回退功耗寄存器 |
| 2Ch | FBPRDY | Flash 存储体与电荷泵就绪寄存器 |
| 30h | FPAC1 | Flash 电荷泵访问控制寄存器 1 |
| 38h | FMSTAT | Flash 模块状态寄存器 |
| 3Ch | FRD_INTF_CTRL | Flash 读取接口控制寄存器 |

复杂的位访问类型采用编码形式，以便放入较小的表格单元格。下表列出了本节使用的访问类型编码。

表 4-3 FLASH_CTRL_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| 写类型 | | |
| W | W | 写 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i,j,k,l,m,n | | 这些变量用于寄存器名称、偏移量或地址时，表示寄存器数组的取值；对应寄存器属于一组重复寄存器。这些寄存器组构成层次结构，并使用公式表示数组。 |
| y | | 此变量用于寄存器名称、偏移量或地址时，表示寄存器数组的取值。 |

### 4.14.2.1 FRDCNTL 寄存器

偏移地址：20h　复位值：00000F00h

Flash 读取控制寄存器。

表 4-4 FRDCNTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | RWAIT | | | |
| R-0h | | | | R/W-Fh | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |

表 4-5 FRDCNTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11-8 | RWAIT | R/W | Fh | 随机读取等待状态。这些位指示在 Flash 读取 / 取指访问中插入的等待状态数。RWAIT 可设置为 0～0xF。RWAIT≥7有效（150MHZ），Flash 访问在 RWAIT+1 个 SYSCLK 周期后返回数据。<br>注意：各 SYSCLK 频率所需的等待状态数请参阅器件数据手册。<br>复位类型：SYSRSn |
| 7-0 | RESERVED | R | 0h | 保留。 |

### 4.14.2.2 FBAC 寄存器

偏移地址：24h　复位值：0000000Fh

Flash 存储体访问控制寄存器。

表 4-6 FBAC 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BAGP | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R/W-Fh | | | | | | | |

表 4-7 FBAC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-8 | BAGP | R/W | 0h | 存储体活动宽限期。这些位保存 BAGP 递减计数器的初始计数值。对某个存储体的任何访问都会使其 BAGP 计数器重新装载该存储体的 BAGP 值。在最后一次访问该 Flash 存储体之后，递减计数器延迟 0～255 个预分频 SYSCLK 周期，再使存储体进入由 FBFALLBACK 寄存器指定的某种回退功耗模式。当回退模式不是 ACTIVE（活动）时，此值必须大于 1。<br>注意：BAGP 递减计数器使用的预分频时钟由输入 SYSCLK 经 16 分频产生。<br>复位类型：SYSRSn |
| 7-0 | RESERVED | R/W | Fh | 保留。 |

### 4.14.2.3 FBFALLBACK 寄存器

偏移地址：28h　复位值：00000003h

Flash 存储体回退功耗寄存器。

表 4-8 FBFALLBACK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | BNKPWR0 | |
| R-0h | | | | | | R/W-3h | |

表 4-9 FBFALLBACK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-2 | RESERVED | R | 0h | 保留。 |
| 1-0 | BNKPWR0 | R/W | 3h | 存储体功耗模式控制。<br>00：休眠（禁用感应放大器和感应参考）。<br>01：待机（禁用感应放大器，但启用感应参考）。<br>10：保留。<br>11：活动（感应放大器和感应参考均启用）。<br>注意：若存储体和电荷泵未处于活动模式，此时发生访问，则该寄存器值会自动更改为活动模式。<br>复位类型：SYSRSn |

### 4.14.2.4 FBPRDY 寄存器

偏移地址：2Ch　复位值：00000000h

Flash 存储体与电荷泵就绪寄存器。

表 4-10 FBPRDY 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | 保留 | BANK0RDY |
| R-0h | | | | | | R-0h | R-0h |

表 4-11 FBPRDY 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-2 | RESERVED | R | 0h | 保留。 |
| 1 | RESERVED | R | 0h | 保留。 |
| 0 | BANK0RDY | R | 0h | 存储体 0 就绪。这是一个只读寄存器，软件可在访问之前判断存储体 0 是否已准备好接受 Flash 访问。<br>注意：用户应等待电荷泵和存储体均就绪后再尝试访问。<br>0：存储体 0 未就绪。<br>1：存储体 0 处于活动功耗模式，已准备好接受访问。<br>复位类型：SYSRSn |

### 4.14.2.5 FPAC1 寄存器

偏移地址：30h　复位值：00A00000h

Flash 电荷泵访问控制寄存器 1。

表 4-12 FPAC1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | PSLEEP | | | |
| R-0h | | | | R/W-A0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PSLEEP | | | | | | | |
| R/W-A0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |

表 4-13 FPAC1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-28 | RESERVED | R | 0h | 保留。 |
| 27-16 | PSLEEP | R/W | A0h | 电荷泵休眠。这些位保存电荷泵休眠递减计数器的初始计数值。电荷泵处于休眠模式时，功耗模式管理逻辑将电荷泵休眠计数器保持在此值。电荷泵退出休眠功耗模式后，递减计数器延迟 0～PSLEEP 个预分频 SYSCLK 周期，再使电荷泵进入活动功耗模式。<br>注意：电荷泵休眠递减计数器与存储体休眠递减计数器使用相同的预分频时钟，该时钟由输入 SYSCLK 经 2 分频产生。<br>注意：BootROM 将 PSLEEP 配置为 0x755，以支持 150 MHz 工作频率。如有需要，用户可根据应用要求修改 PSLEEP 值。<br>复位类型：SYSRSn |
| 15-0 | RESERVED | R | 0h | 保留。 |

### 4.14.2.6 FMSTAT 寄存器

偏移地址：38h　复位值：00000000h

Flash 模块状态寄存器。

表 4-14 FMSTAT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | 保留 | 保留 |
| R-0h | | | | | | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | 保留 | ILA | PGV | 保留 | EV | 保留 | BUSY |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERS | PGM | INVDAT | CSTAT | VOLTSTAT | ESUSP | PSUSP | 保留 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 4-15 FMSTAT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | 保留。 |
| 17 | RESERVED | R | 0h | 保留。 |
| 16 | RESERVED | R | 0h | 保留。 |
| 15 | RESERVED | R | 0h | 保留。 |
| 14 | RESERVED | R | 0h | 保留。 |
| 13 | ILA | R | 0h | 置1，表示无权限烧录或擦除导致的烧录或擦除失败。<br>无权限场景：<br>(1) 非安全区代码擦除/烧录安全区域Flash；<br>(2) zone1代码烧录/擦除zone2的Flash区域； |
| 12 | PGV | R | 0h | 编程校验。此位置位表示：编程操作已施加允许的最大编程脉冲数，但某个字仍未成功编程。<br>复位类型：SYSRSn |
| 11 | RESERVED | R | 0h | 保留。 |
| 10 | EV | R | 0h | 擦除校验。此位置位表示：擦除操作已施加允许的最大擦除脉冲数，但某个扇区仍未成功擦除。<br>复位类型：SYSRSn |
| 9 | RESERVED | R | 0h | 保留。 |
| 8 | BUSY | R | 0h | 此位置位表示正在处理编程、擦除或挂起操作。<br>复位类型：SYSRSn |
| 7 | ERS | R | 0h | 擦除活动。此位置位表示 Flash 模块正在执行擦除操作。擦除开始时置位，擦除完成时清零；擦 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 除挂起时也会清零，擦除恢复时重新置位。<br>复位类型：SYSRSn |
| 6 | PGM | R | 0h | 编程活动。此位置位表示 Flash 模块正在执行编程操作。编程开始时置位，编程完成时清零；编程挂起时也会清零，编程恢复时重新置位。<br>复位类型：SYSRSn |
| 5 | INVDAT | R | 0h | 无效数据。此位置位表示用户试图在已有 0 的位置编程写入 1。<br>复位类型：SYSRSn |
| 4 | CSTAT | R | 0h | 命令状态。有限状态机（FSM）启动后，任何失败都会使此位置位。此位置位用于通知主机：编程、擦除或扇区验证命令失败，且该命令已停止。通过清除状态命令可清除此位。对于某些错误，由于其原因不属于其他错误位所涵盖的类型，此位将是 FSM 出错的唯一指示。<br>复位类型：SYSRSn |
| 3 | VOLTSTAT | R | 0h | 内核电压状态。此位置位表示，在编程或擦除操作期间，电荷泵电源的内核电压发生器输出曾降至允许下限以下。<br>复位类型：SYSRSn |
| 2 | ESUSP | R | 0h | 此位置位表示 Flash 模块已接收并处理擦除挂起操作。此位将保持置位，直至发出擦除恢复命令或执行 Clear_More 命令。<br>复位类型：SYSRSn |
| 1 | PSUSP | R | 0h | 此位置位表示 Flash 模块已接收并处理编程挂起操作。此位将保持置位，直至发出编程恢复命令或执行 Clear_More 命令。<br>复位类型：SYSRSn |
| 0 | RESERVED | R | 0h | 保留。 |

### 4.14.2.7 FRD_INTF_CTRL 寄存器

偏移地址：3Ch　复位值：00000000h

Flash 读取接口控制寄存器。

表 4-16 FRD_INTF_CTRL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | DATA_CACHE_EN | PREFETCH_EN |
| R-0h | | | | | | R/W-0h | R/W-0h |

表 4-17 FRD_INTF_CTRL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-2 | RESERVED | R | 0h | 保留。 |
| 1 | DATA_CACHE_EN | R/W | 0h | 数据缓存使能。<br>0：禁用数据缓存。<br>1：启用数据缓存。<br>复位类型：SYSRSn |
| 0 | PREFETCH_EN | R/W | 0h | 预取使能。<br>0：禁用预取机制。<br>1：启用预取机制。<br>复位类型：SYSRSn |

### 4.14.3 FLASH_ECC_REGS 寄存器

表中列出了 FLASH_ECC_REGS 的存储器映射寄存器。表中未列出的寄存器偏移地址均应视为保留位置，不应修改这些位置的寄存器内容。

表 4-18 FLASH_ECC_REGS 寄存器

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | ECC_ENABLE | ECC 使能 |
| 8h | SINGLE_ERR_ADDR_LOW | 低位单比特错误地址 |
| Ch | SINGLE_ERR_ADDR_HIGH | 高位单比特错误地址 |

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 10h | UNC_ERR_ADDR_LOW | 低位不可纠正错误地址 |
| 14h | UNC_ERR_ADDR_HIGH | 高位不可纠正错误地址高部 |
| 18h | ERR_STATUS | 错误状态 |
| 1Ch | ERR_POS | 错误位置 |
| 20h | ERR_STATUS_CLR | 错误状态清除 |
| 24h | ERR_CNT | 单比特错误数量 |
| 28h | ERR_THRESHOLD | 错误阈值 |
| 2Ch | ERR_INTFLG | 错误中断标志 |
| 54h | ERR_INTCLR | 错误中断标志清除 |
| 30h | FDATAH_TEST | 高位测试数据 |
| 34h | FDATAL_TEST | 低位测试数据 |
| 38h | FADDR_TEST | ECC 测试地址 |
| 40h | FECC_TEST | ECC 测试校验值 |
| 44h | FECC_CTRL | ECC 测试控制 |
| 48h | FOUTH_TEST | 高位测试数据输出 |
| 4Ch | FOUTL_TEST | 低位测试数据输出 |
| 50h | FECC_STATUS | ECC 测试状态 |

复杂的位访问类型采用编码形式，以便放入较小的表格单元格。下表列出了本节使用的访问类型编码。

表 4-19 FLASH_ECC_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1S | W<br>1S | 写<br>1置位 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |

| 访问类型 | 编码 | 说明 |
| --- | --- | --- |
| i,j,k,l,m,n | | 这些变量用于寄存器名称、偏移量或地址时，表示寄存器数组的取值；对应寄存器属于一组重复寄存器。这些寄存器组构成层次结构，并使用公式表示数组。 |
| y | | 此变量用于寄存器名称、偏移量或地址时，表示寄存器数组的取值。 |

### 4.14.3.1 ECC_ENABLE 寄存器

偏移地址：0h　复位值：0000000Ah

ECC 使能。

表 4-20 ECC_ENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | ENABLE | | | |
| R-0h | | | | R/W-Ah | | | |

表 4-21 ECC_ENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-4 | RESERVED | R | 0h | 保留。 |
| 3-0 | ENABLE | R/W | Ah | ECC 使能。值为 0xA 时启用 ECC，其他任何值均禁用 ECC。<br>复位类型：SYSRSn |

### 4.14.3.2 SINGLE_ERR_ADDR_LOW 寄存器

偏移地址：8h　复位值：00000000h

低位数据单比特错误地址。

表 4-22 SINGLE_ERR_ADDR_LOW 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ERR_ADDR_L | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ERR_ADDR_L | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ERR_ADDR_L | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERR_ADDR_L | | | | | | | |
| R/W-0h | | | | | | | |

表 4-23 SINGLE_ERR_ADDR_LOW 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | ERR_ADDR_L | R/W | 0h | 在 128 位对齐的存储区域的低 64 位中发生单比特错误时，对应的 64 位对齐地址。<br>复位类型：SYSRSn |

### 4.14.3.3 SINGLE_ERR_ADDR_HIGH 寄存器

偏移地址：Ch　复位值：00000000h

高位单比特错误地址。

表 4-24 SINGLE_ERR_ADDR_HIGH 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ERR_ADDR_H | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ERR_ADDR_H | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ERR_ADDR_H | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERR_ADDR_H | | | | | | | |
| R/W-0h | | | | | | | |

表 4-25 SINGLE_ERR_ADDR_HIGH 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | ERR_ADDR_H | R/W | 0h | 在 128 位对齐的存储区域的高 64 位中发生单比特错误时，对应的 64 位对齐地址。<br>复位类型：SYSRSn |

### 4.14.3.4 UNC_ERR_ADDR_LOW 寄存器

偏移地址：10h　复位值：00000000h

低位不可纠正错误地址。

表 4-26 UNC_ERR_ADDR_LOW 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| UNC_ERR_ADDR_L | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| UNC_ERR_ADDR_L | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| UNC_ERR_ADDR_L | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNC_ERR_ADDR_L | | | | | | | |
| R/W-0h | | | | | | | |

表 4-27 UNC_ERR_ADDR_LOW 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | UNC_ERR_ADDR_L | R/W | 0h | 在 128 位对齐的存储区域的低 64 位中发生不可纠正错误时，对应的 64 位对齐地址。<br>复位类型：SYSRSn |

### 4.14.3.5 UNC_ERR_ADDR_HIGH 寄存器

偏移地址：14h　复位值：00000000h

高位不可纠正错误地址。

表 4-28 UNC_ERR_ADDR_HIGH 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| UNC_ERR_ADDR_H | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| UNC_ERR_ADDR_H | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| UNC_ERR_ADDR_H | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UNC_ERR_ADDR_H | | | | | | | |
| R/W-0h | | | | | | | |

表 4-29 UNC_ERR_ADDR_HIGH 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | UNC_ERR_ADDR_H | R/W | 0h | 在 128 位对齐的存储区域的高 64 位中发生不可纠正错误时，对应的 64 位对齐地址。<br>复位类型：SYSRSn |

### 4.14.3.6 ERR_STATUS 寄存器

偏移地址：18h　复位值：00000000h

错误状态。

表 4-30 ERR_STATUS 寄存器表

| 3 1 | 3 0 | 2 9 | 2 8 | 2 7 | 2 6 | 2 5 | 2 4 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | UNC_ERR_H | FAIL_1_H | FAIL_0_H |
| R-0h | | | | | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | UNC_ERR_L | FAIL_1_L | FAIL_0_L |
| R-0h | | | | | R-0h | R-0h | R-0h |

表 4-31 ERR_STATUS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-19 | RESERVED | R | 0h | 保留。 |
| 18 | UNC_ERR_H | R | 0h | 不可纠正错误。值为 1 表示在 128 位对齐地址对应的高 64 位中发生了不可纠正错误。向 ERR_STATUS_CLR 寄存器的 UNC_ERR_H_CLR 位写入 1 可清除此位。<br>复位类型：SYSRSn |
| 17 | FAIL_1_H | R | 0h | 纠正后值为 1 的错误（Fail on 1）。<br>0：在 128 位对齐地址对应的高 64 位中，未发生纠正后值为 1 的单比特错误。<br>1：在 128 位对齐地址对应的高 64 位中发生了单比特错误，且纠正后的值为 1。向 ERR_STATUS_CLR 寄存器的 FAIL_1_H_CLR 位写入 1 可清除此位。<br>注意：每次 Flash 访问导致单比特错误时都会更新此位。因此，多次发生单比特错误时，其状态对应最近一次错误。<br>复位类型：SYSRSn |
| 16 | FAIL_0_H | R | 0h | 纠正后值为 0 的错误（Fail on 0）。<br>0：在 128 位对齐地址对应的高 64 位中，未发生纠正后值为 0 的单比特错误。<br>1：在 128 位对齐地址对应的高 64 位中发生了单比特错误，且纠正后的值为 0。向 ERR_STATUS_CLR 寄存器的 FAIL_0_H_CLR 位写入 1 可清除此位。<br>注意：每次 Flash 访问导致单比特错误时都会更新此位。因此，多次发生单比特错误时，其状态对应最近一次错误。<br>复位类型：SYSRSn |
| 15-3 | RESERVED | R | 0h | 保留。 |
| 2 | UNC_ERR_L | R | 0h | 不可纠正错误。值为 1 表示在 128 位对齐地址对应的低 64 位中发生了不可纠正错误。向 ERR_STATUS_CLR 寄存器的 UNC_ERR_L_CLR 位写入 1 可清除此位。<br>复位类型：SYSRSn |
| 1 | FAIL_1_L | R | 0h | 纠正后值为 1 的错误（Fail on 1）。<br>0：在 128 位对齐地址对应的低 64 位中，未发生纠正后值为 1 的单比特错误。<br>1：在 128 位对齐地址对应的低 64 位中发生了单比特错误，且纠正后的值为 1。向 ERR_STATUS_CLR 寄存器的 FAIL_1_L_CLR 位写入 1 可清除此位。<br>注意：每次 Flash 访问导致单比特错误时都会更 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 新此位。因此，多次发生单比特错误时，其状态对应最近一次错误。<br>复位类型：SYSRSn |
| 0 | FAIL_0_L | R | 0h | 纠正后值为 0 的错误（Fail on 0）。<br>0：在 128 位对齐地址对应的低 64 位中，未发生纠正后值为 0 的单比特错误。<br>1：在 128 位对齐地址对应的低 64 位中发生了单比特错误，且纠正后的值为 0。向 ERR_STATUS_CLR 寄存器的 FAIL_0_L_CLR 位写入 1 可清除此位。<br>注意：每次 Flash 访问导致单比特错误时都会更新此位。因此，多次发生单比特错误时，其状态对应最近一次错误。<br>复位类型：SYSRSn |

### 4.14.3.7 ERR_POS 寄存器

偏移地址：1Ch　复位值：00000000h

错误位置。

表 4-32 ERR_POS 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | ERR_TYPE_H |
| R-0h | | | | | | | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | ERR_POS_H | | | | | |
| R-0h | | R-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | ERR_TYPE_L |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | ERR_POS_L | | | | | |
| R-0h | | R-0h | | | | | |

表 4-33 ERR_POS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-25 | RESERVED | R | 0h | 保留。 |
| 24 | ERR_TYPE_H | R | 0h | 错误类型。<br>0：在 128 位对齐地址对应的高 64 个数据位中 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 发生了单比特错误。<br>1：在 128 位对齐地址对应的高 64 位的 ECC 校验位中发生了单比特错误。<br>复位类型：SYSRSn |
| 23-22 | RESERVED | R | 0h | 保留。 |
| 21-16 | ERR_POS_H | R | 0h | 错误位置。表示在 128 位对齐地址对应的高 64 位中发生单比特错误的位位置。应根据 ERR_TYPE 位指示的是校验位还是数据位来解释该位置。如果 ERR_TYPE 指示校验位错误，错误位置范围为 0～7；否则范围为 0～63。<br>复位类型：SYSRSn |
| 15-9 | RESERVED | R | 0h | 保留。 |
| 8 | ERR_TYPE_L | R | 0h | 错误类型。<br>0：在 128 位对齐地址对应的低 64 个数据位中发生了单比特错误。<br>1：在 128 位对齐地址对应的低 64 位的 ECC 校验位中发生了单比特错误。<br>复位类型：SYSRSn |
| 7-6 | RESERVED | R | 0h | 保留。 |
| 5-0 | ERR_POS_L | R | 0h | 错误位置。表示在 128 位对齐地址对应的低 64 位中发生单比特错误的位位置。应根据 ERR_TYPE 位指示的是校验位还是数据位来解释该位置。如果 ERR_TYPE 指示校验位错误，错误位置范围为 0～7；否则范围为 0～63。<br>复位类型：SYSRSn |

### 4.14.3.8 ERR_STATUS_CLR 寄存器

偏移地址：20h　复位值：00000000h

错误状态清除。

表 4-34 ERR_STATUS_CLR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | UNC_ERR_H_CLR | FAIL_1_H_CLR | FAIL_0_H_CLR |
| R-0h | | | | | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | UNC_ERR_L_CLR | FAIL_1_L_CLR | FAIL_0_L_CLR |
| R-0h | | | | | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 4-35 ERR_STATUS_CLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-19 | RESERVED | R | 0h | 保留。 |
| 18 | UNC_ERR_H_CLR | R-0/W1S | 0h | 不可纠正错误清除。向此位写入 1 可清除 ERR_STATUS 寄存器中的 UNC_ERR_H 位。写入 0 无效。读取返回 0。<br>复位类型：SYSRSn |
| 17 | FAIL_1_H_CLR | R-0/W1S | 0h | 纠正后值为 1 的错误清除。向此位写入 1 可清除 ERR_STATUS 寄存器中的 FAIL_1_H 位。写入 0 无效。读取返回 0。<br>复位类型：SYSRSn |
| 16 | FAIL_0_H_CLR | R-0/W1S | 0h | 纠正后值为 0 的错误清除。向此位写入 1 可清除 ERR_STATUS 寄存器中的 FAIL_0_H 位。写入 0 无效。读取返回 0。<br>复位类型：SYSRSn |
| 15-3 | RESERVED | R | 0h | 保留。 |
| 2 | UNC_ERR_L_CLR | R-0/W1S | 0h | 不可纠正错误清除。向此位写入 1 可清除 ERR_STATUS 寄存器中的 UNC_ERR_L 位。写入 0 无效。读取返回 0。<br>复位类型：SYSRSn |
| 1 | FAIL_1_L_CLR | R-0/W1S | 0h | 纠正后值为 1 的错误清除。向此位写入 1 可清除 ERR_STATUS 寄存器中的 FAIL_1_L 位。写入 0 无效。读取返回 0。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 0 | FAIL_0_L_CLR | R-0/W1S | 0h | 纠正后值为 0 的错误清除。向此位写入 1 可清除 ERR_STATUS 寄存器中的 FAIL_0_L 位。写入 0 无效。读取返回 0。<br>复位类型：SYSRSn |

### 4.14.3.9 ERR_CNT 寄存器

偏移地址：24h　复位值：00000000h

单比特错误数量。

表 4-36 ERR_CNT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ERR_CNT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERR_CNT | | | | | | | |
| R/W-0h | | | | | | | |

表 4-37 ERR_CNT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-0 | ERR_CNT | R/W | 0h | 单比特错误计数。每发生一次 ECC 单比特错误，此计数器加 1。达到阈值后，计数器停止对单比特错误计数。无论是否达到阈值，都可使用“单比特错误中断清除”（Single Err Int Clear）位清除 ERR_CNT。此功能同时适用于 ECC 逻辑测试模式和正常工作模式。<br>复位类型：SYSRSn |

### 4.14.3.10 ERR_THRESHOLD 寄存器

偏移地址：28h　复位值：00000000h

错误阈值。

表 4-38 ERR_THRESHOLD 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ERR_THRESHOLD | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERR_THRESHOLD | | | | | | | |
| R/W-0h | | | | | | | |

表 4-39 ERR_THRESHOLD 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-0 | ERR_THRESHOLD | R/W | 0h | 单比特错误阈值。设置单比特错误的阈值。当 ERR_CNT 值等于 THRESHOLD 值且再次发生单比特错误时，SINGLE_ERR_INT 标志置位，并触发中断。<br>复位类型：SYSRSn |

### 4.14.3.11 ERR_INTFLG 寄存器

偏移地址：2Ch　复位值：00000000h

错误中断标志。

表 4-40 ERR_INTFLG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | UNC_ERR_INTFLG | SINGLE_ERR_INTFLG |
| R-0h | | | | | | R-0h | R-0h |

表 4-41 ERR_INTFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-2 | RESERVED | R | 0h | 保留。 |
| 1 | UNC_ERR_INTFLG | R | 0h | 不可纠正位错误中断标志。发生不可纠正错误时，此位置位并触发 UNC_ERR_INT 中断。向 ERR_INTCLR 寄存器的 UNC_ERR_INTCLR 位写入 1 可清除此位。<br>复位类型：SYSRSn |
| 0 | SINGLE_ERR_INTFLG | R | 0h | 单比特错误中断标志。当 ERR_CNT 值等于 ERR_THRESHOLD 值且发生单比特错误时，SINGLE_ERR_INT 标志置位并触发 SINGLE_ERR_INT 中断。向 ERR_INTCLR 寄存器的 SINGLE_ERR_INTCLR 位写入 1 可清除此位。<br>复位类型：SYSRSn |

### 4.14.3.12 ERR_INTCLR 寄存器

偏移地址：54h　复位值：00000000h

错误中断标志清除。

表 4-42 ERR_INTCLR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | UNC_ERR_INTCLR | SINGLE_ERR_INTCLR |
| R-0h | | | | | | R-0/W1S-0h | R-0/W1S-0h |

表 4-43 ERR_INTCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-2 | RESERVED | R | 0h | 保留。 |
| 1 | UNC_ERR_INTCLR | R-0/W1S | 0h | 不可纠正位错误中断标志清除。向此位写入 1 可清除 UNC_ERR_INT_FLG。写入 0 无效。<br>复位类型：SYSRSn |
| 0 | SINGLE_ERR_INTCLR | R-0/W1S | 0h | 单比特错误中断标志清除。向此位写入 1 可清除 SINGLE_ERR_INT_FLG。写入 0 无效。<br>复位类型：SYSRSn |

### 4.14.3.13 FDATAH_TEST 寄存器

偏移地址：30h　复位值：00000000h

高 32 位测试数据。

表 4-44 FDATAH_TEST 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| FDATAH | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FDATAH | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FDATAH | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FDATAH | | | | | | | |
| R/W-0h | | | | | | | |

表 4-45 FDATAH_TEST 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | FDATAH | R/W | 0h | 所选 64 位数据的高32位。在 ECC 测试模式下，用户可配置所选数据块的位 63:32。<br>复位类型：SYSRSn |

## 4.14.3.14 FDATAL_TEST 寄存器

偏移地址：34h　复位值：00000000h

低 32 位测试数据。

表 4-46 FDATAL_TEST 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| FDATAL | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FDATAL | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FDATAL | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FDATAL | | | | | | | |
| R/W-0h | | | | | | | |

表 4-47 FDATAL_TEST 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | FDATAL | R/W | 0h | 所选 64 位数据的低32位。在 ECC 测试模式下，用户可配置所选数据块的位 31:0。<br>复位类型：SYSRSn |

## 4.14.3.15 FADDR_TEST 寄存器

偏移地址：38h　复位值：00000000h

ECC 测试地址。

表 4-48 FADDR_TEST 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | ADDRH | | | | | |
| R-0h | | R/W-0h | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADDRL | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDRL | | | | | 保留 | | |
| R/W-0h | | | | | R-0h | | |

表 4-49 FADDR_TEST 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | 保留。 |
| 21-16 | ADDRH | R/W | 0h | 所选 64 位数据的地址。在 ECC 测试模式下，用户可配置所选数据的地址位。将地址左移 1 位以获得字节地址，忽略地址的最低 3 位，并将其余地址中的位 21:16 写入此字段。<br>复位类型：SYSRSn |
| 15-3 | ADDRL | R/W | 0h | 所选 64 位数据的地址。在 ECC 测试模式下，用户可配置所选数据的地址位。将地址左移 1 位以获得字节地址，忽略地址的最低 3 位，并将其余地址中的位 15:3 写入此字段。<br>复位类型：SYSRSn |
| 2-0 | RESERVED | R | 0h | 保留。 |

### 4.14.3.16 FECC_TEST 寄存器

偏移地址：40h　复位值：00000000h

ECC 测试校验值。

表 4-50 FECC_TEST 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ECC | | | | | | | |
| R/W-0h | | | | | | | |

表 4-51 FECC_TEST 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | ECC | R/W | 0h | 所选 64 位数据的 8 位 ECC。在 ECC 测试模式下，用户可配置所选 64 位数据块的 ECC 位。<br>复位类型：SYSRSn |

### 4.14.3.17 FECC_CTRL 寄存器

偏移地址：44h　复位值：00000000h

ECC 控制。

表 4-52 FECC_CTRL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | DO_ECC_CALC | ECC_SELECT | ECC_TEST_EN |
| R-0h | | | | | R-0/W1S-0h | R/W-0h | R/W-0h |

表 4-53 FECC_CTRL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-3 | RESERVED | R | 0h | 保留。 |
| 2 | DO_ECC_CALC | R-0/W1S | 0h | 使能 ECC 计算。通过置位 ECC_TEST_EN 启用 ECC 测试逻辑后，ECC 逻辑将在一个周期内，对写入 ECC 测试寄存器的数据和地址计算 ECC。<br>复位类型：SYSRSn |
| 1 | ECC_SELECT | R/W | 0h | ECC 块选择。<br>0：选择针对存储体数据位 [63:0] 的 ECC 块。<br>1：选择针对存储体数据位 [127:64] 的 ECC 块。<br>复位类型：SYSRSn |
| 0 | ECC_TEST_EN | R/W | 0h | ECC 测试模式使能。<br>0：禁用 ECC 测试模式。<br>1：启用 ECC 测试模式。<br>复位类型：SYSRSn |

## 4.14.3.18 FOUTH_TEST 寄存器

偏移地址：48h　复位值：00000000h

测试数据输出高 32 位。

表 4-54 FOUTH_TEST 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DATAOUTH | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DATAOUTH | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DATAOUTH | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATAOUTH | | | | | | | |
| R-0h | | | | | | | |

表 4-55 FOUTH_TEST 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | DATAOUTH | R | 0h | 测试数据输出高32位。保存所选 ECC 块输出数据的位 63:32。<br>复位类型：SYSRSn |

## 4.14.3.19 FOUTL_TEST 寄存器

偏移地址：4Ch　复位值：00000000h

测试数据输出低 32 位。

表 4-56 FOUTL_TEST 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DATAOUTL | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DATAOUTL | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DATAOUTL | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATAOUTL | | | | | | | |
| R-0h | | | | | | | |

表 4-57 FOUTL_TEST 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | DATAOUTL | R | 0h | 测试数据输出低32位。保存所选 ECC 块输出数据的位 31:0。<br>复位类型：SYSRSn |

## 4.14.3.20 FECC_STATUS 寄存器

偏移地址：50h　复位值：00000000h

ECC 状态。

表 4-58 FECC_STATUS 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | ERR_TYPE |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA_ERR_POS | | | | | | UNC_ERR | SINGLE_ERR |
| R-0h | | | | | | R-0h | R-0h |

表 4-59 FECC_STATUS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-9 | RESERVED | R | 0h | 保留。 |
| 8 | ERR_TYPE | R | 0h | 测试模式 ECC 单比特错误指示。为 1 时，表示单比特错误位于校验位；为 0 时，表示单比特错误位于数据位（同时要求 SINGLE_ERR 字段已置位）。<br>复位类型：SYSRSn |
| 7-2 | DATA_ERR_POS | R | 0h | 测试模式单比特错误位置。保存发生单比特错误的位位置。应根据 CHK_ERR 位指示的是校验位还是数据位来解释该位置。如果 CHK_ERR 指示校验位错误，错误位置范围为 0～7；否则 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 范围为 0～63。<br>复位类型：SYSRSn |
| 1 | UNC_ERR | R | 0h | 测试模式 ECC 双比特错误。为 1 时，表示 ECC 测试产生了不可纠正的位错误。<br>复位类型：SYSRSn |
| 0 | SINGLE_ERR | R | 0h | 测试模式 ECC 单比特错误。为 1 时，表示 ECC 测试产生了单比特错误。<br>复位类型：SYSRSn |

### 4.14.4 FLASH 寄存器与 DriverLib 函数的对应关系

表 4-60 FLASH 寄存器与 DriverLib 函数的对应关系

| 文件 | DriverLib 函数 |
|---|---|
| **FRDCNTL** | |
| flash.h | Flash_setWaitstates |
| **FBAC** | |
| flash.h | Flash_setBankActiveGracePeriod |
| **FBFALLBACK** | |
| flash.h | Flash_setBankPowerMode |
| **FBPRDY** | |
| flash.h | Flash_isBankReady |
| **FPAC1** | |
| flash.h | Flash_setPumpWakeupTime |
| **FMSTAT** | |
| - | |
| **FRD_INTF_CTRL** | |
| flash.h | Flash_enablePrefetch |
| flash.h | Flash_disablePrefetch |
| flash.h | Flash_enableCache |
| flash.h | Flash_disableCache |
| **ECC_ENABLE** | |
| flash.h | Flash_enableECC |
| flash.h | Flash_disableECC |
| **SINGLE_ERR_ADDR_LOW** | |
| flash.h | Flash_getSingleBitErrorAddressLow |
| **SINGLE_ERR_ADDR_HIGH** | |
| flash.h | Flash_getSingleBitErrorAddressHigh |
| **UNC_ERR_ADDR_LOW** | |

| 文件 | DriverLib 函数 |
| --- | --- |
| flash.h | Flash_getUncorrectableErrorAddressLow |
| **UNC_ERR_ADDR_HIGH** | |
| flash.h | Flash_getUncorrectableErrorAddressHigh |
| **ERR_STATUS** | |
| flash.h | Flash_getLowErrorStatus |
| flash.h | Flash_getHighErrorStatus |
| flash.h | Flash_clearLowErrorStatus |
| flash.h | Flash_clearHighErrorStatus |
| **ERR_POS** | |
| flash.h | Flash_getLowErrorPosition |
| flash.h | Flash_getHighErrorPosition |
| flash.h | Flash_clearLowErrorPosition |
| flash.h | Flash_clearHighErrorPosition |
| flash.h | Flash_getLowErrorType |
| flash.h | Flash_getHighErrorType |
| **ERR_STATUS_CLR** | |
| flash.h | Flash_clearLowErrorStatus |
| flash.h | Flash_clearHighErrorStatus |
| **ERR_CNT** | |
| flash.h | Flash_getErrorCount |
| **ERR_THRESHOLD** | |
| flash.h | Flash_setErrorThreshold |
| **ERR_INTFLG** | |
| flash.h | Flash_getInterruptFlag |
| **ERR_INTCLR** | |
| flash.h | Flash_clearSingleErrorInterruptFlag |
| flash.h | Flash_clearUncorrectableInterruptFlag |
| **FDATAH_TEST** | |
| flash.h | Flash_setDataHighECCTest |
| **FDATAL_TEST** | |
| flash.h | Flash_setDataLowECCTest |
| **FADDR_TEST** | |
| flash.h | Flash_setECCTestAddress |
| **FECC_TEST** | |
| flash.h | Flash_setECCTestECCBits |
| **FECC_CTRL** | |

| 文件 | DriverLib 函数 |
| --- | --- |
| flash.h | Flash_enableECCTestMode |
| flash.h | Flash_disableECCTestMode |
| flash.h | Flash_selectLowECCBlock |
| flash.h | Flash_selectHighECCBlock |
| flash.h | Flash_performECCCalculation |
| **FOUTH_TEST** | |
| flash.h | Flash_getTestDataOutHigh |
| **FOUTL_TEST** | |
| flash.h | Flash_getTestDataOutLow |
| **FECC_STATUS** | |
| flash.h | Flash_getECCTestStatus |
| flash.h | Flash_getECCTestErrorPosition |
| flash.h | Flash_getECCTestSingleBitErrorType |

# 5 双时钟比较器（DCC）

## 5.1 简介

双时钟比较器模块用于基于第二时钟评估和监控时钟输入，该时钟可为更精确可靠的版本。该仪器用于检测时钟源或时钟结构中的故障，从而提升系统的安全指标。

### 5.1.1 功能

各 DCC 模块的主要功能包括：

- 使应用程序能够确保两个时钟信号频率之间保持固定比例。
- 支持通过参考时钟周期数定义可编程容差窗口。
- 支持无需应用程序干预的持续监测功能。
- 支持用于点测量的单序列模式。
- 允许为每个计数器选择不同的时钟源，从而实现多种特定应用场景。

### 5.1.2 模块示意图

图中分别展示了 DCC 如何与系统其他部分连接、DCC 模块的主要概念。

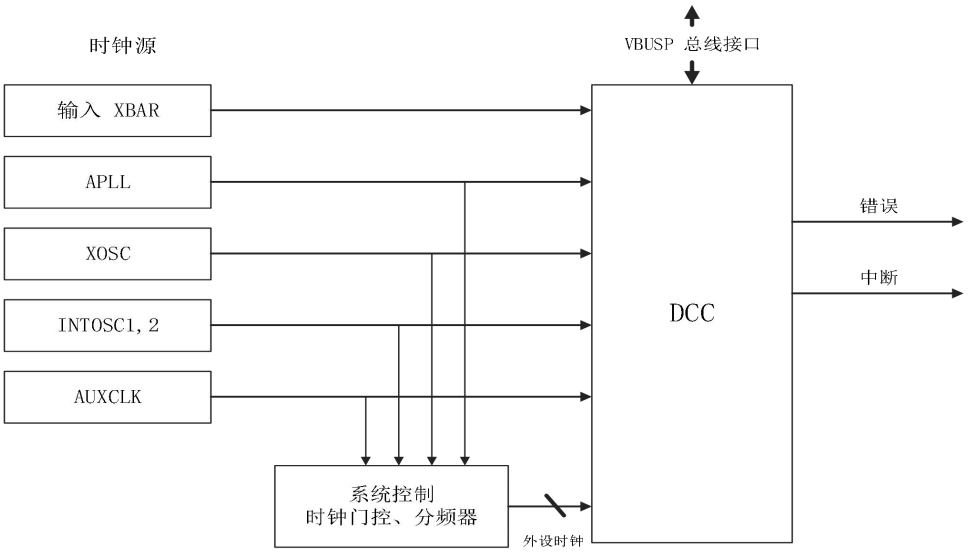



图 5-1 DCC 模块框图

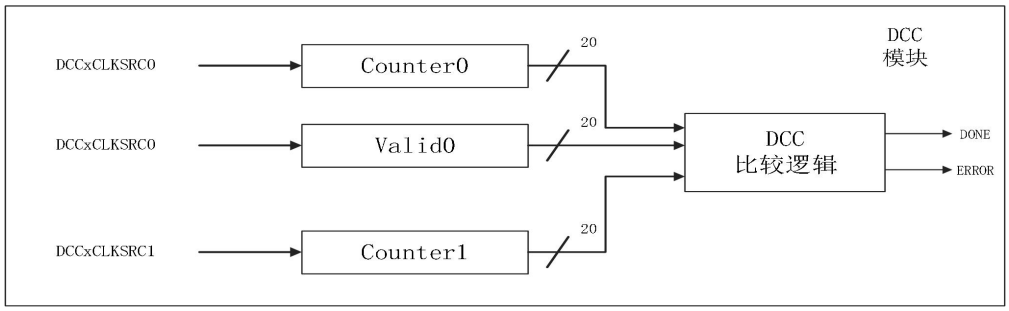



图 5-2 DCC 操作

## 5.2 模块操作

如图所示，DCC 包含三个计数器——计数器 0、有效计数器 0 和计数器 1。初始状态下，所有计数器均预装用户自定义的初始值。当启用 DCC 时，计数器 0 和计数器 1 会分别以 Clock0 和 Clock1 的频率为基准开始递减。当计数器 0 归零（到期）时，有效计数器 0 会以 Clock0 的频率继续递减。若计数器 1 在有效窗口内归零，则不会触发错误，此时 Clock1 的时钟信号即被视为符合用户设定的容差范围。

### 5.2.1 配置 DCC 计数器

计数器 0 和计数器 1 的配置基于时钟 0 与时钟 1 的频率比（Fclk1×计数器 0 = Fclk0×计数器 1）。有效计数器 0 提供容差，其配置基于 DCC 中的误差。由于时钟 0 和时钟 1 是异步的，计数器的启动和停止不会同步。因此，在配置计数器时必须考虑两种不同的误差来源：

- 由时钟 0 和时钟 1 异步定时引起的 DCC 误差：这取决于时钟 0 和时钟 1 的频率：
  - 若 Fclk1 > Fclk0，则异步误差（以时钟 0 周期计）= 2 + 2×(Fsysclk/Fclk0)
  - 若 Fclk1 < Fclk0，则异步误差（以时钟 0 周期计）= 2×(Fclk0/Fclk1) + 2×-(Fsysclk/Fclk0)
  - 若 Fclk1 未知，则异步误差（以时钟 0 周期计）= 2 + 2×(Fsysclk/Fclk0)
- 数字化误差 = 8 时钟 0 周期

  **DCC 错误（以 Clock0 周期计）=异步错误 + 数字化错误**

DCC 误差表现为被测时钟的频率误差。该误差由 DCC 引起，不代表被测时钟频率本身的误差。应用在配置计数器时需考虑此因素，并确定 DCC 误差的容许范围，以界定测量窗口。举例说明：

**窗口（以 Clock0 周期为单位）= (DCC 误差) / (0.01×容差)**

例如，若 DCC 误差为 10 且所需容差为±0.1%，则：

窗口（以 Clock0 周期计）= 10/(0.01×0.1) = 10000

根据上述窗口公式，若所需容差较低，则计数器数值较大，测量窗口也会相应扩大。这意味着 0.1%容差对应的计数器数值会比 0.2%容差的数值更大。因此，应根据应用定义的容差，以 Clock0 周期为单位确定测量窗口。

被测时钟可能存在允许的频率误差，若该误差在预期范围内，配置计数器时也可考虑误差因素。例如，当使用外部晶体作为参考时钟测量 INTOSC1/2 频率时，可将 INTOSC1/2 的允许容差（如±1%）纳入考量，并将其纳入计数器配置中。计算公式如下：

允许频率误差（以 Clock0 周期为单位）= 窗口 ×（允许频率容差（百分比）/100）总误差（以 Clock0 周期为单位）= DCC 误差 + 允许频率误差

以下方程式用于配置计数器数值：

计数器 0（DCCCNTSEED0）= 窗口 - 总误差；

有效 0（DCCVALIDSEED0）= 2 × 总误差；

计数器 1（DCCCNTSEED1）= 窗口 × (Fclk1/Fclk0)

注：计数器 1 为 20 位计数器，其最大可能值不得超过 1048575。若实际数值超过该上限，则需提高 DCC 误差的容差设定，从而缩小测量窗口。计算最小容差的公式如下：容差（%）=（100 × DCC 误差 × (Fclk1/Fclk0)) / 1048575

### 5.2.2 单次测量模式

通过启用单次触发模式，可对 DCC 模块进行一次倒计数编程。在此模式下，当倒计数器 0 和有效计数器 0 均归零时，DCC 将停止工作。

单次触发模式下完成一次倒计数序列后，DCC 会自动禁用，从而防止继续计数。该模式通常用于信号频率的点检。

示例-1：验证 PLLRAWCLK 频率

一个实际应用案例是使用 XTAL 作为参考时钟来验证 PLL 输出时钟频率。假设 XTAL 为 10MHz， PLL 输出频率为 100MHz， SYSCLK 为 100MHz，允许频率容差为 0.1%，所需 DCC 容差为 0.1%。测量流程如下：

- 将 Clock0 的信号源设置为 XTAL ，同时将 Clock1 的信号源设置为 PLL 输出时钟，用于驱动 Counter0 和 Valid0。
- 计算出计数器的初始值：Counter0=29940；Valid0=120；Counter1=300000。
- 启用 DCC 后，计数器 Counter0 和 Counter1 将从初始值开始倒计时。
- 当 Counter0 归零时，会自动触发 Valid0 计数器。
- 若 Valid0 归零而 Counter1 未归零，系统将设置错误状态标志并向 CLIC 发送“DCC 错误”信号。此时 Counter1 将被冻结，停止继续倒计时。应用程序可配置 CLIC 在检测到 DCC 错误时触发中断。
- 需要清除错误状态标志并重启 DCC 模块，使其恢复运行状态以准备进行下一次点测量。

若序列结束时未生成错误，则设置 DONE 状态标志并触发 DONE 中断。应用程序必须在重启 DCC 前清除 DONE 标志。

错误条件：

以下任一情况均可产生错误状态：

1.当计数器 0 尚未归零时，计数器 1 已开始倒计时至 0。该现象表明时钟 1 运行速度超出预期，或时钟 0 运行速度低于预期。此错误包括时钟 0 卡在 1 或 0 的情况。

2. 即使计数器 0 和有效位 0 均已归零，计数器 1 仍未归零。该现象表明时钟 1 运行速度低于预期。此错误包括时钟 1 卡在 1 或 0 的情况。

3.任何错误都会冻结计数器的计数功能。应用程序随后可读取计数器数值，以帮助确定错误原因。

示例-2：测量 AUXCLKIN 频率

单次测量模式的另一个示例是：以 INTOSC1（10MHz）作为参考时钟，SYSCLK 为 10MHz，测量未知频率的 auxclkin。测量流程如下：

- 将计数器 0 和有效位 0 的时钟源设置为 INTOSC1（10MHz），计数器 1 的时钟源设置为 auxclkin。
- 现在配置计数值。容差=±0.1%时，总误差=10 个时钟 0 周期；窗口=10000 个时钟 0 周期；计数器 0=9990；有效位 0=20。由于时钟 1 频率（Fclk1）未知，可将计数器 1 的值设为最大值 1048575（0xFFFFF）。
- 启用 DCC 后，计数器 0 和计数器 1 均从初始值开始倒计时。
- 由于计数器 1 已设为最大值 1048575，当计数器 0 和有效位 0 到期时，计数器不会失效。这会产生预期误差，但应用程序会忽略该错误并使用计数器 1 的值来计算时钟 1 频率（Fclk1）。
- 已知时钟 0 频率（INTOSC1）为 10MHz，通过式（5-1）可计算 auxclkin 频率 Fclk1：

$$\begin{aligned} \text{Fclk1} &= \frac{\text{Fclk0} \times (1048575 - \text{Meas. Counter1})}{\text{Counter0} + \text{Valid0}} \\ &= \frac{10 \times (1048575 - \text{Meas. Counter1})}{9990 + 20} \end{aligned} \quad (5\text{-}1)$$

### 5.2.3 连续监测方式

在此模式下，应用通过 DCC 确保两个时钟信号保持正确的频率比。假设应用需要确保 PLL 输出信号始终与 XTAL 保持固定的频率关系：

- 在此情况下，应用程序可将 XTAL 用作 Clock0 信号（对应 Counter0 和 Valid0），并将 PLL 输出作为 Clock1（对应 Counter1）。

- Counter0、Valid0 和 Counter1 的种子值根据第 5.2.1 节定义的公式选取，确保当 Clock0 和 Clock1 的实际频率与预期频率一致时，Valid0 计数器的倒计数过程中 Counter1 会归零。
- 若 Valid0 计数器倒计数期间 Counter1 归零，则所有计数器（Counter0、Valid0、Counter1）将被重置为初始种子值。
- 这一倒计数与校验的循环将持续进行，直至出现错误或 DCC 模块被禁用。
- 当应用程序重置并重启 DCC 模块时，所有计数器必须重新加载。

错误条件：

以下任一情况将产生错误状态：

当计数器 0 尚未归零时，计数器 1 已开始倒计时至 0。这种情况表明时钟 1 运行速度超出预期，或时钟 0 运行速度低于预期。该情形包含时钟 0 卡在 1 或 0 的情况。

即使计数器 0 和有效位 0 均已归零，计数器 1 仍未归零。这说明时钟 1 运行速度低于预期。该情形包含时钟 1 卡在 1 或 0 的情况。

任何错误都会冻结计数器的计数功能。应用程序随后可读取计数器数值，以帮助确定错误原因。

### 5.2.4 错误条件

在持续运行模式下，计数器将重新加载种子值，并在以下条件下继续进行递减计数：

- 模块由应用程序重置或重启
- 计数器 0、有效值 0 及计数器 1 均无误地归零。

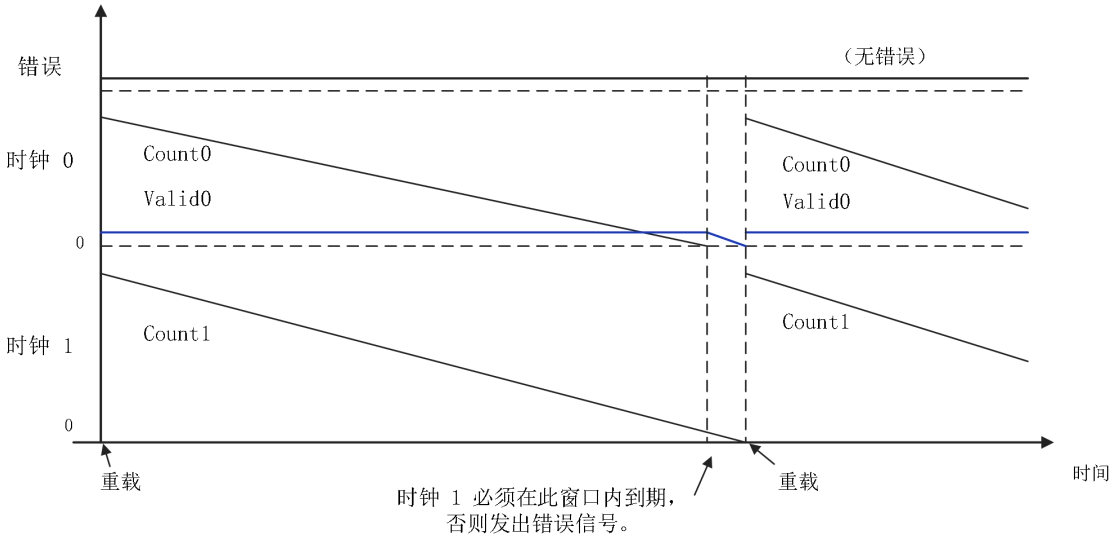



图 5-3 计数器关系

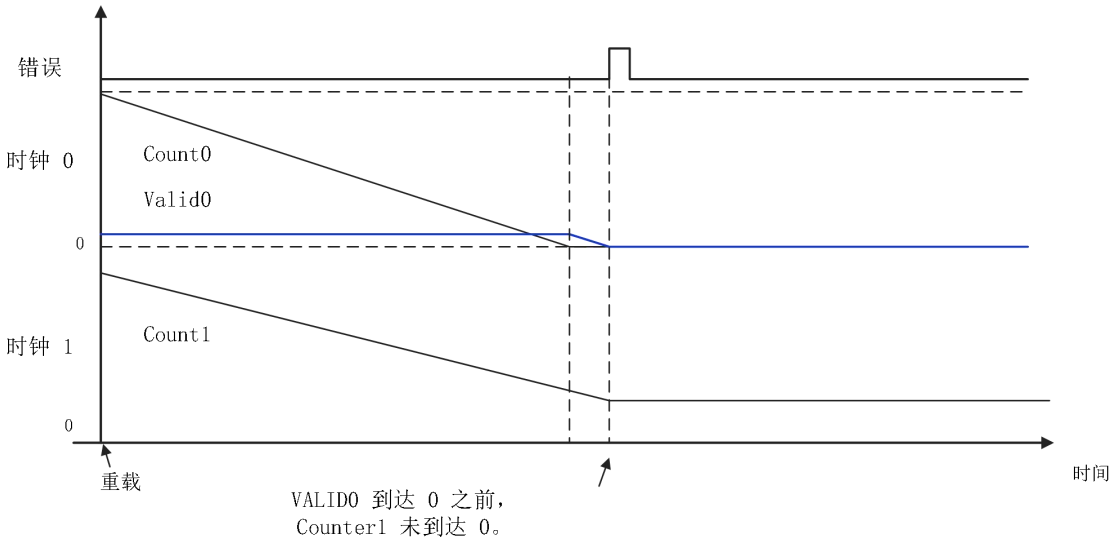



图 5-4 时钟 1 计数速度慢于时钟 0 - 导致错误并停止计数

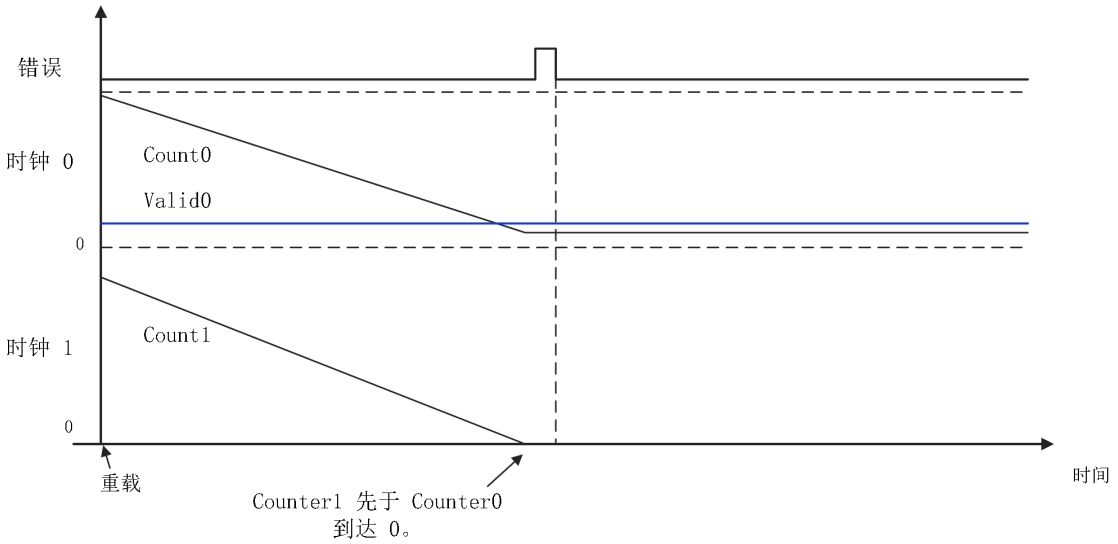



图 5-5 时钟 1 比时钟 0 快——导致错误并停止计数

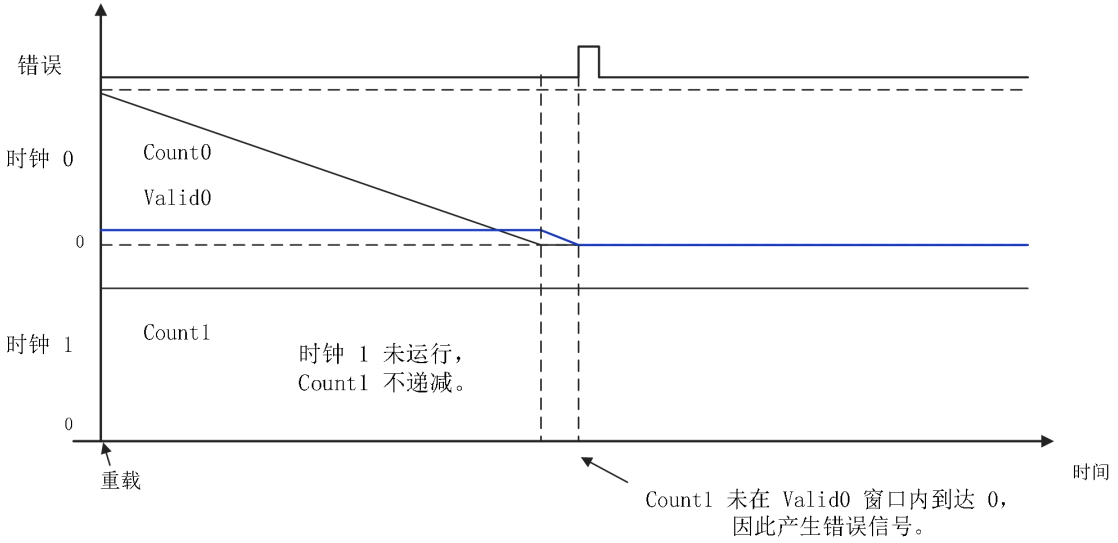



图 5-6 时钟 1 未就位 - 导致错误并停止计数

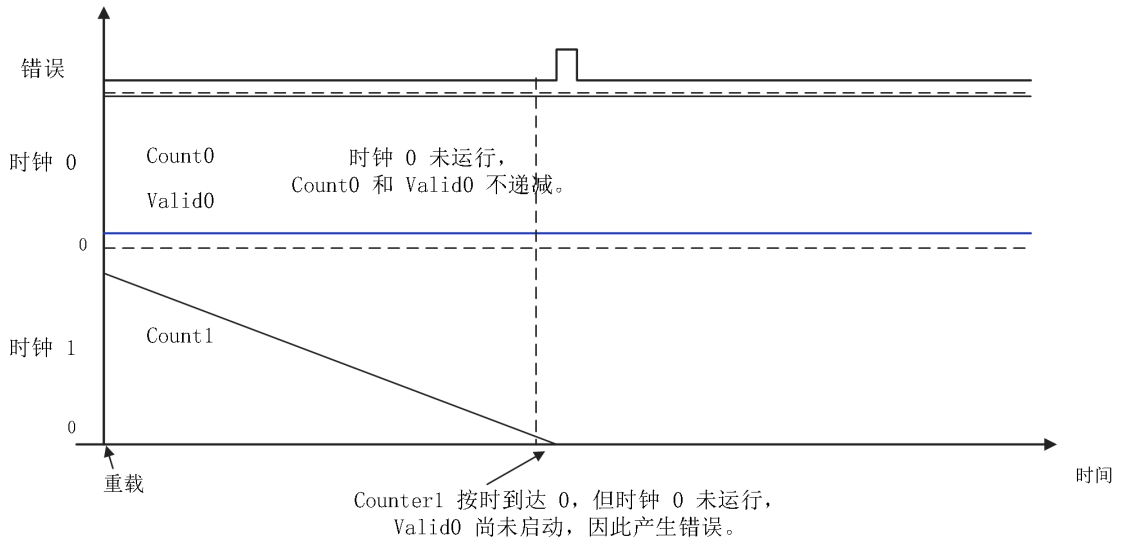



图 5-7 时钟 0 未存在 - 导致错误并停止计数

## 5.3 中断

DCC 会在以下两种情况之一触发中断：

- 当 DCC 完成计数且所有计数器在指定窗口内到期时，表示已完成操作（前提是设置为 DCCGCTRL. DONENA =1）。
- 当 DCC 计数失败且计数器未在指定窗口内到期时，表示发生错误事件，并在设置为 DCCGCTRL. ERRENA =1 时触发中断。

由 DONE 或错误事件触发的中断会进行或运算处理，并标记为 DCCx 中断。应用程序中断服务程序需通过检查 DCCSTATUS 寄存器中的状态标志，以判断该中断是由错误还是 DONE 事件引发。

通过启用 CLKFAILCFG.DCCx_ERROR_EN 标志，DCC 错误中断还可配置为不可屏蔽中断（RNMI）

## 5.4 软件

### 5.4.1 DCC 示例

注：这些示例位于 IDE 安装目录中。

### 5.4.2 DCC 单次射频时钟验证

文件： dcc_ex1_single_shot_verification .c

本程序采用 XTAL 时钟作为基准时钟，用于验证 PLLRAW 时钟的频率。

双时钟比较器模块 0 负责时钟验证工作。其中，时钟源 0 为基准时钟（Fclk0=20MHz），时钟源 1 为待验证时钟（Fclk1=100MHz）。Seed 是加载到计数器中的初始值。

外部连接：

- 无

观察变量

- 状态/结果 - PLLRAW 时钟验证状态

### 5.4.3 DCC 单次射频时钟测量

文件： dcc_ex2_single_shot_measurement .c

该程序演示了使用 XTAL 作为参考时钟，对 INTOSC2 时钟引脚校准进行单次测量。

双时钟比较器模块 0 用于时钟测量。时钟源 0 为参考时钟（Fclk0=20MHz），时钟源 1 为待测时钟（Fclk1=10MHz）。由于需要测量时钟 1 的频率，初始种子被设置为计数器的最大值。

外部连接：

- 无

观察变量：

- 结果——INTOSC2 时钟测量成功完成时的状态
- meas_freq1 ——测量的时钟频率（本例中为 INTOSC2）

### 5.4.4 DCC 连续时钟监测

文件： dcc_ex3_continuous_monitoring_of_clock .c

该程序演示了使用 INTOSC2 作为参考时钟对系统中 PLL 时钟进行持续监控。一旦发生错误，将触发中断，导致计数器的递减/重载停止。

双时钟比较器模块 0 用于时钟监控。其中时钟源 0 为基准时钟（频率 Fclk0=10MHz），时钟源 1 为待监控时钟（频率 Fclk1=100MHz）。通过设置时钟 0 和时钟 1 的种子值，可实现 300 微秒的监控窗口。种子值是加载到计数器中的数值。为便于演示，对时钟 1 的种子值进行微调，从而在持续监控中产生误差。

注意：在闪存配置下运行时，建议在加载示例后执行复位与重启操作，以清除任何陈旧的标志/状态。

外部连接：

- 无

观察变量：

- 状态/结果 - PLLRAW 时钟监控状态
- cnt0 - 当发生错误时的计数器 0 值
- cnt1 - 当发生错误时的计数器 1 值
- valid - 当发生错误时的 valid0 值

### 5.4.5 DCC 连续时钟监测

文件： dcc_ex3_continuous_monitoring_of_clock_syscfg.c

本程序演示了使用 INTOSC2 作为参考时钟对系统中 PLL 时钟进行持续监测。当出现任何错误时，该程序会触发中断，导致计数器的递减/重载操作停止。双时钟比较器模块 0 用于时钟监测，其中 clocksource0 为参考时钟（Fclk0=10MHz），clocksource1 为需要监测的时钟（Fclk1=100MHz）。时钟 0 和时钟 1 的种子值由本项目所含 sysconfig 文件中定义的容错参数自动设定。为便于演示，我们假设存在不切实际的容错阈值，从而在持续监测过程中产生错误。

注意：在闪存配置下运行时，建议在加载示例后执行复位重启操作，以清除任何陈旧标志/状态。

外部连接：

- 无

观察变量：

- status/result - PLLRAW 时钟监控状态
- cnt0 - 生成错误时的计数器 0 值
- cnt1 - 生成错误时的计数器 1 值
- valid - 生成错误时的 valid0 值。

### 5.4.6 DCC 时钟故障检测

文件： dcc_ex4_clock_fail_detect .c

本程序演示了使用 XTAL 作为振荡器时钟源时，对系统 PLL 时钟进行持续监测的时钟故障检测。一旦振荡器时钟发生故障，将触发 DCC 错误中断，导致计数器的递减/重载停止。本示例中，通过关闭 XTAL 振荡器来模拟时钟故障。当 ISR 服务完成后，将振荡器源切换为 INTOSC1，并关闭 PLL 。

双时钟比较器模块 0 用于时钟监控。时钟源 0 为参考时钟（Fclk0=20MHz），时钟源 1 为待监控时钟（Fclk1=100MHz）。种子值是加载到计数器中的数值。

在当前示例中， XTAL 应为运行在晶体模式下的谐振器，该模式随后会被关闭以模拟时钟故障。若使用 SE 晶体，需物理断开板载时钟。有关需设置的计数器种子值，请参阅技术参考手册（TRM）。注意：在闪存配置模式下运行时，建议在加载示例后执行复位重启操作，以清除任何陈旧的标志/状态。

外部连接

- 无

监控变量

- 状态/结果 - 时钟故障检测状态

## 5.5 寄存器描述

本节描述双时钟比较器寄存器。 注意：DCC 由引导 ROM 使用；因此，寄存器值可能与硬件复位值不同。用户需要确保在启用 DCC 之前，将这些寄存器的值配置为所需的值。

### 5.5.1 DCC 基地址表

表 5-1 DCC 基地址表

| 位字段实例 | 位字段结构 | DriverLib 名称 | 基地址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| Dcc0Regs | DCC_REGS | DCC0_BASE | 0x0067_8000 | - | - | 是 |
| Dcc1Regs | DCC_REGS | DCC1_BASE | 0x0067_8100 | - | - | 是 |

### 5.5.2 DCC_REGS 寄存器

表中未列出的寄存器偏移地址均为保留位置，不应修改这些位置的寄存器内容。

表 5-2 DCC_REGS 寄存器

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | DCCGCTRL | 启动 / 停止计数器，并清除错误信号 |
| 10h | DCCCNTSEED0 | 时钟源 0 所连接计数器的种子值 |
| 18h | DCCVALIDSEED0 | 时钟源 0 所连接超时计数器的种子值 |
| 20h | DCCCNTSEED1 | 时钟源 1 所连接计数器的种子值 |
| 28h | DCCSTATUS | DCC 模块状态 |
| 30h | DCCCNT0 | 时钟源 0 所连接计数器的值 |
| 38h | DCCVALID0 | 时钟源 0 所连接有效时间计数器的值 |
| 40h | DCCCNT1 | 时钟源 1 所连接计数器的值 |
| 48h | DCCCLKSRC1 | 计数器 1 的时钟源选择 |
| 50h | DCCCLKSRC0 | 计数器 0 的时钟源选择 |

复杂的位访问类型采用编码形式。下表列出了本节使用的访问类型编码。

表 5-3 DCC_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |

| 访问类型 | 编码 | 说明 |
|---|---|---|
| R-1 | R<br>-1 | 读<br>返回 1 |
| 写类型 | | |
| W | W | 写 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i、j、k、l、m、n | | 用于寄存器名称、偏移地址或地址时，表示属于某个重复组的寄存器数组值。这些组按层次组织，以公式表示数组。 |
| y | | 用于寄存器名称、偏移地址或地址时，表示寄存器数组值。 |

### 5.5.2.1 DCCGCTRL 寄存器

偏移地址：0h　复位值：00005555h

启动 / 停止计数器，并清除错误信号。

表 5-4 DCCGCTRL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DONEENA | | | | SINGLESHOT | | | |
| R/W-5h | | | | R/W-5h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERRENA | | | | DCCENA | | | |
| R/W-5h | | | | R/W-5h | | | |

表 5-5 DCCGCTRL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-12 | DONEENA | R/W | 5h | 完成信号使能。<br>使能或禁用完成中断信号，但不影响 DCCSTAT 寄存器中的完成状态标志。<br>0101：禁用完成信号。<br>其他值：使能完成信号。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 11-8 | SINGLESHOT | R/W | 5h | 单次模式使能。<br>使能或禁用 DCC 的重复运行。<br>1010：当 COUNTER0 和 VALID0 均减至零时停止计数。<br>1011：保留。<br>其他值：连续重复运行（直到发生错误）。<br>复位类型：SYSRSn。 |
| 7-4 | ERRENA | R/W | 5h | 错误信号使能。<br>使能或禁用错误信号。<br>0101：禁用错误信号。<br>其他值：使能错误信号。<br>复位类型：SYSRSn。 |
| 3-0 | DCCENA | R/W | 5h | DCC 使能。<br>启动或停止 DCC 的运行。<br>0101：计数器停止。<br>其他值：计数器运行。<br>复位类型：SYSRSn。 |

### 5.5.2.2 DCCCNTSEED0 寄存器

偏移地址：10h　复位值：00000000h

时钟源 0 所连接计数器的种子值。

表 5-6 DCCCNTSEED0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | COUNTSEED0 | | | |
| R-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| COUNTSEED0 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNTSEED0 | | | | | | | |
| R/W-0h | | | | | | | |

表 5-7 DCCCNTSEED0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 19-0 | COUNTSEED0 | R/W | 0h | 计数器 0 的种子值。<br>包含加载到计数器 0（时钟源 0）的种子值。<br>注意：COUNTSEED0 寄存器为 0 时运行 DCC，将导致未定义的行为。<br>复位类型：SYSRSn。 |

### 5.5.2.3 DCCVALIDSEED0 寄存器

偏移地址：18h　复位值：00000000h

时钟源 0 所连接超时计数器的种子值。

表 5-8 DCCVALIDSEED0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VALIDSEED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VALIDSEED | | | | | | | |
| R/W-0h | | | | | | | |

表 5-9 DCCVALIDSEED0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-0 | VALIDSEED | R/W | 0h | 有效持续时间计数器 0 的种子值。<br>包含加载到时钟源 0 的有效持续时间计数器中的种子值。<br>注意：VALIDSEED0 寄存器为 0 时运行 DCC，将导致未定义的行为。VALID0 定义了 COUNT1 计数到期的时间窗口。该窗口的宽度应至少为 4 个周期。不要向 VALID0 寄存器写入小于 4 的值。<br>复位类型：SYSRSn。 |

### 5.5.2.4 DCCCNTSEED1 寄存器

偏移地址：20h　复位值：00000000h

时钟源 1 所连接计数器的种子值。

表 5-10 DCCCNTSEED1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | COUNTSEED1 | | | |
| R-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| COUNTSEED1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNTSEED1 | | | | | | | |
| R/W-0h | | | | | | | |

表 5-11 DCCCNTSEED1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | 保留。 |
| 19-0 | COUNTSEED1 | R/W | 0h | 计数器 1 的种子值。<br>包含加载到计数器 1（时钟源 1）的种子值。<br>注意：COUNTSEED1 寄存器为 0 时运行 DCC，将导致未定义的行为。<br>复位类型：SYSRSn。 |

### 5.5.2.5 DCCSTATUS 寄存器

偏移地址：28h　复位值：00000000h

DCC 模块状态。

表 5-12 DCCSTATUS 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | DONE | ERR |
| R-0h | | | | | | R/W-0h | R/W-0h |

表 5-13 DCCSTATUS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留。 |
| 1 | DONE | R/W | 0h | 单次模式完成标志。<br>指示单次模式是否已无错误地完成。向此位写入 1 可清除此标志。<br>0：单次模式尚未完成。<br>1：单次模式已完成。<br>复位类型：SYSRSn。 |
| 0 | ERR | R/W | 0h | 错误标志。<br>指示是否已发生错误。向此位写入 1 可清除此标志。<br>0：未发生错误。<br>1：已发生错误。<br>复位类型：SYSRSn。 |

### 5.5.2.6 DCCCNT0 寄存器

偏移地址：30h　复位值：00000000h

时钟源 0 所连接计数器的值。

表 5-14 DCCCNT0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | COUNT0 | | | |
| R-0h | | | | R-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| COUNT0 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNT0 | | | | | | | |
| R-0h | | | | | | | |

表 5-15 DCCCNT0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | 保留。 |
| 19-0 | COUNT0 | R | 0h | 计数器 0 的当前值。<br>复位类型：SYSRSn。 |

### 5.5.2.7 DCCVALID0 寄存器

偏移地址：38h　复位值：00000000h

时钟源 0 所连接有效时间计数器的值。

表 5-16 DCCVALID0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| VALID0 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VALID0 | | | | | | | |
| R-0h | | | | | | | |

表 5-17 DCCVALID0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-0 | VALID0 | R | 0h | 有效持续时间计数器 0 的当前值。<br>复位类型：SYSRSn。 |

### 5.5.2.8 DCCCNT1 寄存器

偏移地址：40h　复位值：00000000h

时钟源 1 所连接计数器的值。

表 5-18 DCCCNT1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | COUNT1 | | | |
| R-0h | | | | R-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| COUNT1 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNT1 | | | | | | | |
| R-0h | | | | | | | |

表 5-19 DCCCNT1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | 保留。 |
| 19-0 | COUNT1 | R | 0h | 计数器 1 的当前值。<br>复位类型：SYSRSn。 |

### 5.5.2.9 DCCCLKSRC1 寄存器

偏移地址：48h　复位值：00000000h

计数器 1 的时钟源选择。

表 5-20 DCCCLKSRC1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| KEY | | | | 保留 | | | |
| R-0/W-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | CLKSRC1 | | | | |
| R-0h | | | R/W-0h | | | | |

表 5-21 DCCCLKSRC1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-12 | KEY | R-0/W | 0h | 使能或禁用 COUNT1 的时钟源写入。<br>1010：CLKSRC 字段选择 COUNT1 的时钟源。<br>其他值：保留原值，对寄存器字段的新写入不产生影响。<br>复位类型：SYSRSn。 |
| 11-5 | RESERVED | R | 0h | 保留。 |

## 5.5.2.10 DCCCLKSRC0 寄存器

偏移地址：50h　复位值：00000000h

计数器 0 的时钟源选择。

表 5-22 DCCCLKSRC0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| KEY | | | | 保留 | | | |
| R-0/W-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | CLKSRC0 | | | |
| R-0h | | | | R/W-0h | | | |

表 5-23 DCCCLKSRC0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15-12 | KEY | R-0/W | 0h | 使能或禁用 COUNT0 的时钟源写入。<br>1010：与密钥一同写入的 CLKSRC0 字段更新为新的 COUNT0 时钟源选择。<br>其他值：保留原值，对寄存器字段的新写入不产生影响。<br>复位类型：SYSRSn。 |
| 11-4 | RESERVED | R | 0h | 保留。 |
| 3-0 | CLKSRC0 | R/W | 0h | 计数器 0 的时钟源选择。<br>当 KEY 字段使能此功能时，指定 COUNT0 的时钟源。<br>注意：下列未明确定义的值均为保留值。<br>复位类型：SYSRSn。<br>0h (R/W) = 晶体振荡器输出<br>1h (R/W) = INTOSC1 输出<br>2h (R/W) = INTOSC2 输出<br>5h (R/W) = CPU1 系统时钟<br>Ch (R/W) = INPUTXBAR1 的输入 16 |

## 5.5.3 DCC 寄存器与 DriverLib 函数的对应关系

表 5-24 DCC 寄存器与 DriverLib 函数的对应关系

| 文件 | DriverLib 函数 |
|---|---|
| **DCCGCTRL** | |
| dcc.h | DCC_enableModule |

| 文件 | DriverLib 函数 |
| --- | --- |
| dcc.h | DCC_disableModule |
| dcc.h | DCC_enableErrorSignal |
| dcc.h | DCC_enableDoneSignal |
| dcc.h | DCC_disableErrorSignal |
| dcc.h | DCC_disableDoneSignal |
| dcc.h | DCC_enableSingleShotMode |
| dcc.h | DCC_disableSingleShotMode |
| **DCCCNTSEED0** | |
| dcc.h | DCC_setCounterSeeds |
| **DCCVALIDSEED0** | |
| dcc.h | DCC_setCounterSeeds |
| **DCCCNTSEED1** | |
| dcc.h | DCC_setCounterSeeds |
| **DCCSTATUS** | |
| dcc.h | DCC_getErrorStatus |
| dcc.h | DCC_getSingleShotStatus |
| dcc.h | DCC_clearErrorFlag |
| dcc.h | DCC_clearDoneFlag |
| sysctl.c | SysCtl_isPLLValid |
| **DCCCNT0** | |
| dcc.h | DCC_getCounter0Value |
| **DCCVALID0** | |
| dcc.h | DCC_getValidCounter0Value |
| **DCCCNT1** | |
| dcc.h | DCC_getCounter1Value |
| **DCCCLKSRC1** | |
| dcc.h | DCC_setCounter1ClkSource |
| dcc.h | DCC_getCounter1ClkSource |
| **DCCCLKSRC0** | |
| dcc.h | DCC_setCounter0ClkSource |
| dcc.h | DCC_getCounter0ClkSource |

# 6 后台循环冗余校验模块（BGCRC）

## 6.1 简介

BGCRC 模块会对可配置的内存块计算 CRC-32 校验值。其实现方式为：在空闲周期（CPU 或 DMA 未对该内存块进行访问的时段）获取指定的内存块数据，进而完成校验值计算。计算得出的 CRC-32 值会与标准 CRC-32 值进行比较，以判定结果为“通过”或“失败”。本质上，BGCRC 模块的作用是帮助识别内存故障与数据损坏问题。本设备上配备有一个 BGCRC 模块，标识为 CPU_CRC。

## 6.2 主要特性

BGCRC 模块具有以下特性：

- 对 64 位数据进行一个周期的 CRC-32 计算
- 零等待状态存储器没有 CPU 带宽影响
- 双操作模式（CRC-32 模式和清理模式）
- 用于计时 CRC-32 完成过程的看门狗定时器
- 能够暂停和恢复 CRC-32 计算

## 6.3 功能描述

BGCRC 模块的 CRC-32 计算可通过将寄存器 BGCRC_EN.START 配置为 1010 来启动。一旦启动，该模块便会在无需 CPU 干预的情况下执行后台内存检查。不过，当 CRC-32 计算完成，或出现故障（计算失败）时，会通过中断（interrupt）向 CPU 发送通知。

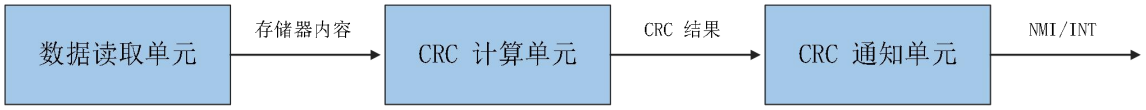



图 6-1 BGCRC 框图

### 6.3.1 数据读取单元

一旦启动 CRC-32 计算，BGCRC 模块就会作为后台进程持续从内存中读取数据。这些读取操作在空闲时间（当 CPU 或 DMA 不访问该内存块时）进行，因此不会影响功能性访问。只有在没有未完成的功能性访问时，数据读取单元才会读取数据。数据读取单元通过从地址 BGCRC_START_ADDR 读取 BGCRC_CTRL2.BLOCK_SIZE 大小的数据块来开始运行。请注意，BGCRC_START_ADDR 必须按 0x100 字节对齐。如果 BGCRC_START_ADDR 未按 0x100 字节对齐，则会将其最低有效位清零，以获得按 0x100 字节对齐的 BGCRC_START_ADDR。例如，如果编程的 BGCRC_START_ADDR = 0x1AF3，那么内部按 0x100 字节对齐的起始地址就是 0x1A00。

当数据读取单元读取数据块时，会检查 ECC 和奇偶校验。读取过程中发生的任何 ECC 或奇偶校验错误，都会通过设置相应的 NMI 来指示，并且如果进行了相关配置，还会生成中断。不过，当出现可纠正的 ECC 错误时，BGCRC 模块不会写回纠正后的内存内容。纠正后的值的写回操作由软件处理。

### 6.3.2 CRC-32 计算单元

数据读取单元读取一个数据块后，会将该数据提供给 CRC-32 计算单元。

该单元使用标准多项式 0x04C11DB7（x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7+ x5 + x4 + x2 + x + 1）计算 CRC-32。CRC-32 单元从数据块中每次提取 64 位数据来计算该多项式。此计算过程耗时 1 个周期。例如，对 1KB 数据块进行 CRC-32 计算需要 128 个周期，即 128 个 64 位数据块各需 1 个周期。CRC-32 计算的初始值可通过 BGCRC_SEED 进行配置。

当一个数据块的 CRC-32 计算完成后，最终结果会被加载到 BGCRC_RESULT 中。

请注意，BGCRC_RESULT 仅包含整个数据块的最终计算结果；中间的 64 位计算结果不会更新 BGCRC_RESULT。硬件会将 BGCRC_RESULT 中的值与 BGCRC_GOLDEN 中的值进行比较，CRC 通知单元会相应地设置 NMI/中断标志。

一旦 CRC-32 计算开始，可通过将 BGCRC_CTRL2.TEST_HALT 设置为 1010 来暂停计算。清除该位可从暂停点恢复 CRC-32 计算。

### 6.3.3 通知单元

在 CRC-32 计算单元完成对一个数据块的 CRC-32 计算后，或者在配置了相关参数的情况下，未能在 BGCRC_WD_MIN 和 BGCRC_WD_MAX 范围内完成计算时，CRC-32 计算单元会在结果通过或失败时发出一个 NMI /中断。此外，在数据读取单元读取数据的过程中，如果发生 ECC 或奇偶校验错误，CRC 通知单元可以发出一个 NMI /中断。当出现 ECC 或奇偶校验错误时，BGCRC 会停止运行，且 BGCRC_CURR_ADDR 中会包含导致该 ECC 或奇偶校验错误的内存地址。BGCRC_CURR_ADDR 的内容以及 NMI /中断标志可用于调试 BGCRC 故障。

图中分别展示了 NMI 线路和中断线路。

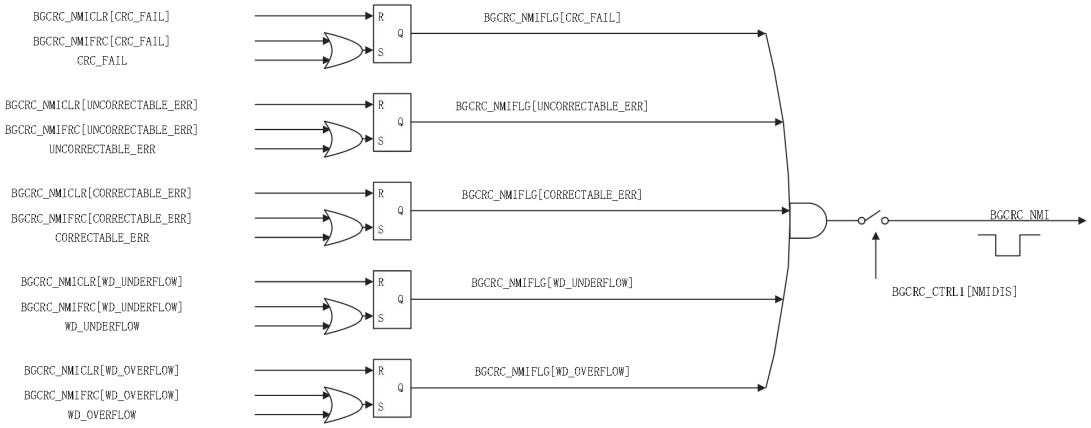



图 6-2 BGCRC 不可屏蔽中断

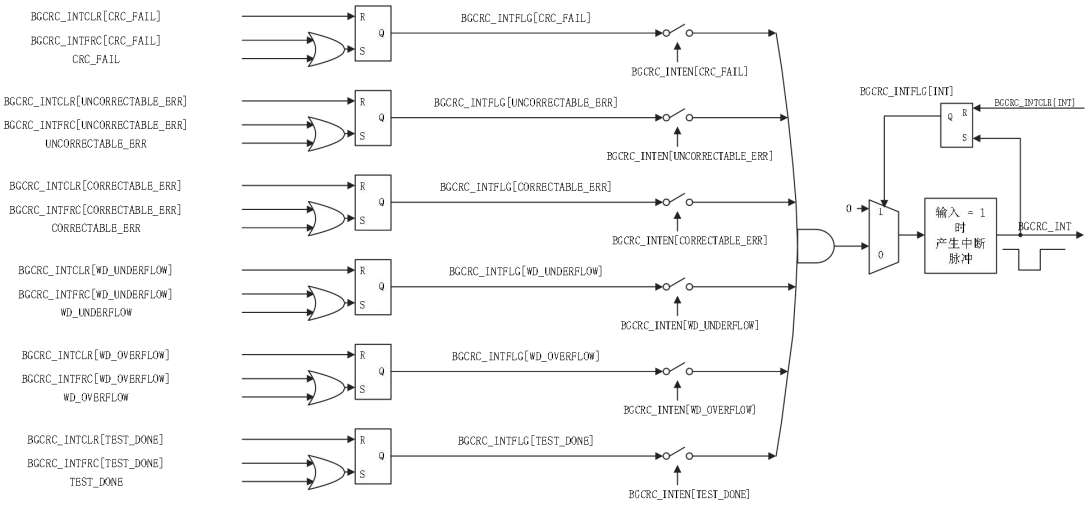



图 6-3 BGCRC 中断

### 6.3.4 操作模式

BGCRC 模块支持两种操作模式：CRC 模式和清理模式。可通过清除或设置寄存器 BGCRC_CTRL2.SCRUB_MODE 来配置操作模式。

#### 6.3.4.1 CRC 模式

对数据块执行 CRC-32 计算，并将结果与标准值进行比较。在此模式下会检查 ECC 和奇偶校验错误。结果不匹配、ECC 错误或奇偶校验错误可能会触发 NMI /中断。

#### 6.3.4.2 清理模式

在清理模式下，CRC-32 结果不会与标准值进行比较，且 BGCRC_RESULT 寄存器不会更新。此模式下也会检查 ECC 和奇偶校验错误。但与 CRC 模式不同的是，只有 ECC 或奇偶校验错误才会产生 NMI/中断。在清理模式下，内存块的奇偶校验位和 ECC 位需要由 CPU 进行初始化。

### 6.3.5 BGCRC 看门狗

BGCRC 模块内置了一个窗口式看门狗，用作诊断工具，以检查内存测试是否在预期的时间窗口内完成。这有助于防范可能导致内存检查无法在规定时间内完成

的硬件缺陷，此类缺陷无法被系统看门狗捕获，因为错误可能是由 DMA 持续访问引起的。窗口机制还有助于检测看门狗操作中的其他故障模式，例如看门狗卡死。

BGCRC 看门狗默认启用，并在 BGCRC 模块开始从内存读取数据时启动。可以通过寄存器 BGCRC_WD_CFG.WDDIS 禁用该看门狗。BGCRC 看门狗计数器是一个 32 位计数器，其值反映在 BGCRC_WD_CNT 寄存器中。下限和上限窗口设置分别通过 BGCRC_WD_MIN 和 BGCRC_WD_MAX 进行配置。BGCRC_WD_MIN 和 BGCRC_WD_MAX 需要在测试开始前配置好，且在 BGCRC 运行期间不得更改。如果进行了配置，当内存测试未能在配置的时间窗口内完成时，会触发 NMI 或中断。当 CRC-32 检查完成、CRC-32 检查失败以及出现 ECC / 奇偶校验错误时，计数器会停止计数。当下一次内存检查开始时，计数器会重置。

可以通过配置 BGCRC_WD_CFG.WDDIS 寄存器来暂停 BGCRC 看门狗。看门狗从暂停状态恢复后，计数器会从之前的计数继续开始计数，除非启动了新的内存检查操作。当 CRC-32 计算暂停时，计数器不会暂停，但默认情况下，在调试期间会暂停。可以通过配置相应的 BGCRC 寄存器来更改仿真期间看门狗的行为。此外，由于仿真期间内存内容会不断变化，不建议在仿真期间运行 BGCRC。在看门狗出现故障时，CRC-32 计算会继续进行，软件需要处理这种情况。

### 6.3.6 硬件及软件错误保护

配置寄存器通过锁定和提交配置来提供保护。每个配置寄存器都可以单独进行锁定和提交操作。寄存器一旦被锁定，在移除锁定之前就无法再进行更新。不过，如果寄存器的锁定状态被提交，那么在设备重置之前，将不允许进行任何进一步的写入操作。建议在完成配置后对寄存器进行锁定，以防止因软件故障而导致的数据损坏。此外，对于模块功能和故障检测至关重要的寄存器，采用多位字段来实现，以抵御硬件故障的影响。

## 6.4 寄存器描述

### 6.4.1 寄存器基地址表

表 6-1 基地址表

| 位域名 | | DriverLib 命名 | 基地址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| 实例 | 结构体 | | | | | |
| BgcrcCpuRegs | BGCRC_RESS | BGCRC_CPU_BASE | 0x0000_5000 | 是 | 是 | 是 |

### 6.4.2 BGCRC_REGS 寄存器

表 6-2 寄存器

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 00h | BGCRC_EN | BGCRC 使能 |
| 04h | BGCRC_CTRL1 | BGCRC 控制寄存器1 |
| 08h | BGCRC_CTRL2 | BGCRC 控制寄存器2 |
| 0Ch | BGCRC_START_ADDR | BGCRC检查的起始地址 |
| 10h | BGCRC_SEED | CRC 计算种子 |
| 1Ch | BGCRC_GOLDEN | 用于对比的黄金CRC值 |
| 20h | BGCRC_RESULT | CRC计算结果 |
| 24h | BGCRC_CURR_ADDR | 当前地址寄存器 |
| 38h | BGCRC_WD_CFG | BGCRC 窗口看门狗配置 |
| 3Ch | BGCRC_WD_MIN | BGCRC 窗口看门狗最小值 |
| 40h | BGCRC_WD_MAX | BGCRC 窗口看门狗最大值 |
| 44h | BGCRC_WD_CNT | BGCRC 窗口看门狗计数 |
| 54h | BGCRC_NMIFLG | BGCRC NMI标志寄存器 |
| 58h | BGCRC_NMICLR | BGCRC NMI 标志清除寄存器 |
| 5Ch | BGCRC_NMIFRC | BGCRC NMI 标志强制寄存器 |
| 68h | BGCRC_INTEN | 中断使能 |
| 6Ch | BGCRC_INTFLG | 中断标志 |
| 70h | BGCRC_INTCLR | 中断标志清除 |
| 74h | BGCRC_INTFRC | 中断标志强制 |
| 78h | BGCRC_LOCK | BGCRC寄存器映射上锁配置 |
| 7Ch | BGCRC_COMMIT | BGCRC 寄存器映射提交配置 |

复杂的位访问类型采用编码形式。下表列出了本节使用的访问类型编码。

表 6-3 BGCRC_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |

| 访问类型 | 编码 | 说明 |
| --- | --- | --- |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1S | W<br>1S | 写<br>1置位 |
| WSonce | W<br>Sonce | 写<br>仅可置位一次 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |

### 6.4.2.1 BGCRC_EN 寄存器

偏移地址：0h　复位值：00000000h

BGCRC 使能。

表 6-4 BGCRC_EN 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| RUN_STS | 保留 | | | | | | |
| R-0h | R-0-0h | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | START | | | |
| R-0-0h | | | | R-0/W-0h | | | |

表 6-5 BGCRC_EN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 31 | RUN_STS | R | 0h | 状态位。<br>0：CRC 模块空闲。<br>1：CRC 模块处于活动状态。<br>在 BGCRC_CTRL2.TEST_HALT = 1 期间，此位保持置位。<br>复位类型：CPUx.SYSRSn。 |
| 30-16 | RESERVED | R-0 | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R-0 | 0h | 保留。 |
| 3-0 | START | R-0/W | 0h | 启动位。<br>1010：启动 CRC 计算。<br>其他值：忽略。<br>注意：CRCEN.START = 1010 时，BGCRC_WD_CNT 寄存器复位。此配置不影响 BGCRC_INTFLG 和 BGCRC_NMIFLG。上一次 CRC 计算完成之前，不应设置此位。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.2 BGCRC_CTRL1 寄存器

偏移地址：4h　复位值：00000000h

BGCRC 控制寄存器 1。

表 6-6 BGCRC_CTRL1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | NMIDIS | | | |
| R-0-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | FREE_SOFT | 保留 | | | |
| R-0-0h | | | R/W-0h | R-0-0h | | | |

表 6-7 BGCRC_CTRL1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | RESERVED | R-0 | 0h | 保留。 |
| 19-16 | NMIDIS | R/W | 0h | 1010：禁用 NMI。<br>其他值：使能 NMI。<br>复位类型：CPUx.SYSRSn。 |
| 15-5 | RESERVED | R-0 | 0h | 保留。 |
| 4 | FREE_SOFT | R/W | 0h | 仿真控制位，控制仿真期间 CRC 计算的行为。<br>0：软停止。CRC 控制器进入 DEBUG SUSPEND 状态时，CRC 模块和 CRC 看门狗立即停止。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：自由运行。CRC 计算和 CRC 看门狗不受 CRC 控制器 DEBUG HALT 状态的影响。<br>复位类型：CPUx.SYSRSn。 |
| 3-0 | RESERVED | R-0 | 0h | 保留。 |

### 6.4.2.3 BGCRC_CTRL2 寄存器

偏移地址：8h　复位值：00000000h

BGCRC 控制寄存器 2。

表 6-8 BGCRC_CTRL2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | SCRUB_MODE | | | |
| R-0-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TEST_HALT | | | | 保留 | | BLOCK_SIZE | |
| R/W-0h | | | | R-0-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BLOCK_SIZE | | | | | | | |
| R/W-0h | | | | | | | |

表 6-9 BGCRC_CTRL2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | RESERVED | R-0 | 0h | 保留。 |
| 19-16 | SCRUB_MODE | R/W | 0h | 擦除模式配置。<br>1010：擦除模式。数据的 CRC 不与参考 CRC 比较，通过 ECC / 奇偶校验逻辑检查错误。<br>其他值：除通过 ECC / 奇偶校验逻辑检查数据正确性外，还在计算结束时将 CRC 值与参考 CRC 比较。<br>注意：1010 配置用于数据存储器的擦除模式：读取存储器数据，通过 ECC / 奇偶校验逻辑检测错误。在此配置下，不更新 BGCRC_RESULT.CRC_VALUE。<br>复位类型：CPUx.SYSRSn。 |
| 15-12 | TEST_HALT | R/W | 0h | 暂停位。<br>1010：停止模块运行。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 其他值：从暂停位置继续 / 恢复 CRC 计算。<br>注意：当 TEST_HALT = 1010 时，配置 BGCRC_EN.START = 1010 将暂停 CRC 计算。将 TEST_HALT 配置为非 1010 的值后，新的检查将恢复。<br>复位类型：CPUx.SYSRSn。 |
| 11-10 | RESERVED | R-0 | 0h | 保留。 |
| 9-0 | BLOCK_SIZE | R/W | 0h | 配置检查的数据块大小。<br>0x0：256 字节（默认）。<br>0x1：512 字节。<br>0x2：768 字节。<br>0x3：1 KB。<br>…<br>0x3FF：256 KB。<br>0xn 对应 (n+1) × 256 字节。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.4 BGCRC_START_ADDR 寄存器

偏移地址：Ch　复位值：00000000h

BGCRC 检查起始地址。

表 6-10 BGCRC_START_ADDR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| START_ADDRESS | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| START_ADDRESS | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| START_ADDRESS | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| START_ADDRESS | | | | | | | |
| R/W-0h | | | | | | | |

表 6-11 BGCRC_START_ADDR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | START_ADDRESS | R/W | 0h | START_ADDRESS 指定测试的起始地址。<br>对于 CPU_CRC，该地址为 CPU 地址；对于 CLA_CRC，该地址为存储器映射到的 CLA 地 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 址。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.5 BGCRC_SEED 寄存器

偏移地址：10h　复位值：00000000h

CRC 计算种子值。

表 6-12 BGCRC_SEED 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SEED | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SEED | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SEED | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEED | | | | | | | |
| R/W-0h | | | | | | | |

表 6-13 BGCRC_SEED 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 31-0 | SEED | R/W | 0h | CRC 初始值。通过写入 START 位触发 CRC 计算时，此值被复制到 CRC 寄存器。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.6 BGCRC_GOLDEN 寄存器

偏移地址：1Ch　复位值：00000000h

用于比较的参考 CRC。

表 6-14 BGCRC_GOLDEN 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CRC_VALUE | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CRC_VALUE | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CRC_VALUE | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRC_VALUE | | | | | | | |
| R/W-0h | | | | | | | |

表 6-15 BGCRC_GOLDEN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | CRC_VALUE | R/W | 0h | 参考 CRC 寄存器。<br>若使能 CRC 检查，则将计算得到的 CRC 值与参考 CRC 比较，并更新状态。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.7 BGCRC_RESULT 寄存器

偏移地址：20h　复位值：00000000h

计算得到的 CRC。

表 6-16 BGCRC_RESULT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CRC_VALUE | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CRC_VALUE | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CRC_VALUE | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRC_VALUE | | | | | | | |
| R-0h | | | | | | | |

表 6-17 BGCRC_RESULT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | CRC_VALUE | R | 0h | CRC 结果寄存器。<br>仅在完成对 BGCRC_CTRL2.BLOCK_SIZE 所配置数据块的 CRC 检查后，才更新此寄存器值。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.8 BGCRC_CURR_ADDR 寄存器

偏移地址：24h　复位值：00000000h

当前地址寄存器。

表 6-18 BGCRC_CURR_ADDR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CURRENT_ADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CURRENT_ADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CURRENT_ADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CURRENT_ADDR | | | | | | | |
| R-0h | | | | | | | |

表 6-19 BGCRC_CURR_ADDR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | CURRENT_ADDR | R | 0h | 当前获取数据的地址。<br>发生故障时，CURRENT_ADDR 字段指示最后一次获取数据的地址。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.9 BGCRC_WD_CFG 寄存器

偏移地址：38h　复位值：00000000h

BGCRC 窗口看门狗配置。

表 6-20 BGCRC_WD_CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | WDDIS | | | |
| R-0-0h | | | | R/W-0h | | | |

表 6-21 BGCRC_WD_CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留。 |
| 3-0 | WDDIS | R/W | 0h | 1010：禁用 CRC 看门狗计数器。<br>其他值：使能 CRC 看门狗。<br>看门狗是递增计数器，在 BGCRC_EN.START 有效时开始计数。在 TEST_HALT 状态下（BGCRC_CTRL2.TEST_HALT = 1010），看门狗也继续计数。可通过显式配置 BGCRC_WD_CFG.WDDIS = 1010，在 TEST_HALT 状态下禁用 CRC 看门狗。禁用后重新使能看门狗时，计数从先前禁用的位置恢复。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.10 BGCRC_WD_MIN 寄存器

偏移地址：3Ch　复位值：00000000h

BGCRC 窗口看门狗最小值。

表 6-22 BGCRC_WD_MIN 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MINVAL | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MINVAL | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MINVAL | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MINVAL | | | | | | | |
| R/W-0h | | | | | | | |

表 6-23 BGCRC_WD_MIN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | MINVAL | R/W | 0h | 如果 CRC 计算在达到 BGCRC_WD_MIN.MINVAL 之前完成，则置位 FAIL_STATUS.WD_UNDERFLOW 标志。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.11 BGCRC_WD_MAX 寄存器

偏移地址：40h　复位值：FFFFFFFFh

BGCRC 窗口看门狗最大值。

表 6-24 BGCRC_WD_MAX 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MAXVAL | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MAXVAL | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MAXVAL | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAXVAL | | | | | | | |
| R/W-FFFFFFFFh | | | | | | | |

表 6-25 BGCRC_WD_MAX 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | MAXVAL | R/W | FFFFFFFFh | 如果 CRC 计算未在达到 BGCRC_WD_MIN.MAXVAL 之前完成，则置位 FAIL_STATUS.WD_OVERFLOW 标志。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.12 BGCRC_WD_CNT 寄存器

偏移地址：44h　复位值：00000000h

BGCRC 窗口看门狗计数值。

表 6-26 BGCRC_WD_CNT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| WD_CNT | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| WD_CNT | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| WD_CNT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WD_CNT | | | | | | | |
| R-0h | | | | | | | |

表 6-27 BGCRC_WD_CNT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | WD_CNT | R | 0h | CRC 窗口看门狗计数器值。<br>CRC 计算结束时计数器值冻结，仅通过 BGCRC_EN.START = 1010 配置重新装载。发生故障时，BGCRC_WD_CNT 寄存器冻结。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.13 BGCRC_NMIFLG 寄存器

偏移地址：54h　复位值：00000000h

BGCRC NMI 标志寄存器。

表 6-28 BGCRC_NMIFLG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | WD_OVERFLOW | WD_UNDERFLOW | CORRECTABLE_ERR | UNCORRECTABLE_ERR | CRC_FAIL | 保留 | 保留 |
| R-0-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0-0h | R-0-0h |

表 6-29 BGCRC_NMIFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | RESERVED | R-0 | 0h | 保留。 |
| 6 | WD_OVERFLOW | R | 0h | 窗口看门狗溢出。<br>1：测试未在达到 BGCRC_WD_MAX.MAXVAL 之前完成。<br>0：无此类错误。<br>复位类型：CPUx.SYSRSn。 |
| 5 | WD_UNDERFLOW | R | 0h | 窗口看门狗下溢。<br>1：测试在达到 BGCRC_WD_MIN.MINVAL 之前完成。<br>0：无此类错误。<br>复位类型：CPUx.SYSRSn。 |
| 4 | CORRECTABLE_ERR | R | 0h | 可纠正错误指示。<br>0：存储器读取期间未发生可纠正的 ECC 错误。<br>1：存储器读取期间发生可纠正的 ECC 错误。<br>注意：每次存储器读取均执行 ECC 计算。模块不回写纠正后的值，因此不能忽略可纠正错误。错误仍保留在存储器中，CPU / CLA 必须在中断服务程序中采取相应的纠正措施。<br>复位类型：CPUx.SYSRSn。 |
| 3 | UNCORRECTABLE_ERR | R | 0h | 不可纠正错误指示。<br>0：存储器读取期间未发生不可纠正的 ECC 错误 / 奇偶校验错误。<br>1：存储器读取期间发生不可纠正的 ECC 错误 / 奇偶校验错误。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 注意：每次存储器读取均执行 ECC / 奇偶校验检查。<br>复位类型：CPUx.SYSRSn。 |
| 2 | CRC_FAIL | R | 0h | CRC 检查失败中断。<br>0：CRC 检查未失败。<br>1：CRC 检查失败。<br>注意：仅在 CRC 计算完成后才使能比较。<br>复位类型：CPUx.SYSRSn。 |
| 1 | RESERVED | R-0 | 0h | 保留。 |
| 0 | RESERVED | R-0 | 0h | 保留。 |

### 6.4.2.14 BGCRC_NMICLR 寄存器

偏移地址：58h　复位值：00000000h

BGCRC NMI 标志清除寄存器。

表 6-30 BGCRC_NMICLR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | WD_OVERFLOW | WD_UNDERFLOW | CORRECTABLE_ERR | UNCORRECTABLE_ERR | CRC_FAIL | 保留 | 保留 |
| R-0-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0-0h | R-0-0h |

表 6-31 BGCRC_NMICLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | RESERVED | R-0 | 0h | 保留。 |
| 6 | WD_OVERFLOW | R-0/W1S | 0h | 清除 WD_OVERFLOW NMI 标志。<br>0：无影响。<br>1：清除 NMI 标志。<br>复位类型：CPUx.SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 5 | WD_UNDERFLOW | R-0/W1S | 0h | 清除 WD_UNDERFLOW NMI 标志。<br>0：无影响。<br>1：清除 NMI 标志。<br>复位类型：CPUx.SYSRSn。 |
| 4 | CORRECTABLE_ERR | R-0/W1S | 0h | 清除 CORRECTABLE_ERR NMI 标志。<br>0：无影响。<br>1：清除 NMI 标志。<br>复位类型：CPUx.SYSRSn。 |
| 3 | UNCORRECTABLE_ERR | R-0/W1S | 0h | 清除 UNCORRECTABLE_ERROR NMI 标志。<br>0：无影响。<br>1：清除 NMI 标志。<br>复位类型：CPUx.SYSRSn。 |
| 2 | CRC_FAIL | R-0/W1S | 0h | 清除 CRC_FAIL NMI 标志。<br>0：无影响。<br>1：清除 NMI 标志。<br>复位类型：CPUx.SYSRSn。 |
| 1 | RESERVED | R-0 | 0h | 保留。 |
| 0 | RESERVED | R-0 | 0h | 保留。 |

### 6.4.2.15 BGCRC_NMIFRC 寄存器

偏移地址：5Ch 复位值：00000000h

BGCRC NMI 标志强制置位寄存器。

表 6-32 BGCRC_NMIFRC 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | WD_OVERFLOW | WD_UNDERFLOW | CORRECTABLE_ERR | UNCORRECTABLE_ERR | CRC_FAIL | 保留 | 保留 |
| R-0-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0-0h | R-0-0h |

表 6-33 BGCRC_NMIFRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | RESERVED | R-0 | 0h | 保留。 |
| 6 | WD_OVERFLOW | R-0/W1S | 0h | 强制置位 WD_OVERFLOW NMI 标志。<br>0：无影响。<br>1：强制置位 NMI 标志。<br>复位类型：CPUx.SYSRSn。 |
| 5 | WD_UNDERFLOW | R-0/W1S | 0h | 强制置位 WD_UNDERFLOW NMI 标志。<br>0：无影响。<br>1：强制置位 NMI 标志。<br>复位类型：CPUx.SYSRSn。 |
| 4 | CORRECTABLE_ERR | R-0/W1S | 0h | 强制置位 CORRECTABLE_ERR NMI 标志。<br>0：无影响。<br>1：强制置位 NMI 标志。<br>复位类型：CPUx.SYSRSn。 |
| 3 | UNCORRECTABLE_ERR | R-0/W1S | 0h | 强制置位 UNCORRECTABLE_ERR NMI 标志。<br>0：无影响。<br>1：强制置位 NMI 标志。<br>复位类型：CPUx.SYSRSn。 |
| 2 | CRC_FAIL | R-0/W1S | 0h | 强制置位 CRC_FAIL NMI 标志。<br>0：无影响。<br>1：强制置位 NMI 标志。<br>复位类型：CPUx.SYSRSn。 |
| 1 | RESERVED | R-0 | 0h | 保留。 |
| 0 | RESERVED | R-0 | 0h | 保留。 |

### 6.4.2.16 BGCRC_INTEN 寄存器

偏移地址：68h　复位值：00000000h

中断使能。

表 6-34 BGCRC_INTEN 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | WD_OVERFLOW | WD_UNDERFLOW | CORRECTABLE_ERR | UNCORRECTABLE_ERR | CRC_FAIL | TEST_DONE | 保留 |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0-0h |

表 6-35 BGCRC_INTEN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | RESERVED | R | 0h | 保留。 |
| 6 | WD_OVERFLOW | R/W | 0h | 0：禁用 WD_OVERFLOW 中断。<br>1：使能 WD_OVERFLOW 中断。<br>复位类型：CPUx.SYSRSn。 |
| 5 | WD_UNDERFLOW | R/W | 0h | 0：禁用 WD_UNDERFLOW 中断。<br>1：使能 WD_UNDERFLOW 中断。<br>复位类型：CPUx.SYSRSn。 |
| 4 | CORRECTABLE_ERR | R/W | 0h | 0：禁用 CORRECTABLE_ERR 中断。<br>1：使能 CORRECTABLE_ERR 中断。<br>复位类型：CPUx.SYSRSn。 |
| 3 | UNCORRECTABLE_ERR | R/W | 0h | 0：禁用 UNCORRECTABLE_ERR 中断。<br>1：使能 UNCORRECTABLE_ERR 中断。<br>复位类型：CPUx.SYSRSn。 |
| 2 | CRC_FAIL | R/W | 0h | 0：禁用 CRC_FAIL 中断。<br>1：使能 CRC_FAIL 中断。<br>复位类型：CPUx.SYSRSn。 |
| 1 | TEST_DONE | R/W | 0h | 0：禁用 TEST_DONE 中断。<br>1：使能 TEST_DONE 中断。<br>复位类型：CPUx.SYSRSn。 |
| 0 | RESERVED | R-0 | 0h | 保留。 |

### 6.4.2.17 BGCRC_INTFLG 寄存器

偏移地址：6Ch　复位值：00000000h

中断标志。

表 6-36 BGCRC_INTFLG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | WD_OVERFLOW | WD_UNDERFLOW | CORRECTABLE_ERR | UNCORRECTABLE_ERR | CRC_FAIL | TEST_DONE | INT |
| R-0-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 6-37 BGCRC_INTFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | RESERVED | R-0 | 0h | 保留。 |
| 6 | WD_OVERFLOW | R | 0h | 窗口看门狗溢出。<br>1：测试未在达到 BGCRC_WD_MAX.MAXVAL 之前完成。<br>0：无此类错误。<br>复位类型：CPUx.SYSRSn。 |
| 5 | WD_UNDERFLOW | R | 0h | 窗口看门狗下溢。<br>1：测试在达到 BGCRC_WD_MIN.MINVAL 之前完成。<br>0：无此类错误。<br>复位类型：CPUx.SYSRSn。 |
| 4 | CORRECTABLE_ERR | R | 0h | 可纠正错误指示。<br>0：存储器读取期间未发生可纠正的 ECC 错误。<br>1：存储器读取期间发生可纠正的 ECC 错误。<br>注意：每次存储器读取均执行 ECC 计算。模块不回写纠正后的值，因此不能忽略可纠正错误。错误仍保留在存储器中，CPU / CLA 必须在中 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 断服务程序中采取相应的纠正措施。<br>复位类型：CPUx.SYSRSn。 |
| 3 | UNCORRECTABLE_ERR | R | 0h | 不可纠正错误指示。<br>0：存储器读取期间未发生不可纠正的 ECC 错误 / 奇偶校验错误。<br>1：存储器读取期间发生不可纠正的 ECC 错误 / 奇偶校验错误。<br>注意：每次存储器读取均执行 ECC / 奇偶校验检查。<br>复位类型：CPUx.SYSRSn。 |
| 2 | CRC_FAIL | R | 0h | CRC 检查失败中断。<br>0：CRC 检查未失败。<br>1：CRC 检查失败。<br>注意：仅在 CRC 计算完成后才使能比较。<br>复位类型：CPUx.SYSRSn。 |
| 1 | TEST_DONE | R | 0h | 完成中断状态标志。<br>0：CRC 正在计算，或 CRC 模块空闲。<br>1：CRC 计算完成。<br>注意：即使 CRC 不匹配，CRC 计算完成时也会置位 TEST_DONE 标志。<br>复位类型：CPUx.SYSRSn。 |
| 0 | INT | R | 0h | 全局中断状态标志。<br>0：未产生中断。<br>1：已产生中断。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.18 BGCRC_INTCLR 寄存器

偏移地址：70h　复位值：00000000h

中断标志清除。

表 6-38 BGCRC_INTCLR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | WD_OVERFLOW | WD_UNDERFLOW | CORRECTABLE_ERR | UNCORRECTABLE_ERR | CRC_FAIL | TEST_DONE | INT |
| R-0-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 6-39 BGCRC_INTCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | RESERVED | R-0 | 0h | 保留。 |
| 6 | WD_OVERFLOW | R-0/W1S | 0h | 清除中断标志。<br>0：无影响。<br>1：清除中断标志。<br>复位类型：CPUx.SYSRSn。 |
| 5 | WD_UNDERFLOW | R-0/W1S | 0h | 清除中断标志。<br>0：无影响。<br>1：清除中断标志。<br>复位类型：CPUx.SYSRSn。 |
| 4 | CORRECTABLE_ERR | R-0/W1S | 0h | 清除中断标志。<br>0：无影响。<br>1：清除中断标志。<br>复位类型：CPUx.SYSRSn。 |
| 3 | UNCORRECTABLE_ERR | R-0/W1S | 0h | 清除中断标志。<br>0：无影响。<br>1：清除中断标志。<br>复位类型：CPUx.SYSRSn。 |
| 2 | CRC_FAIL | R-0/W1S | 0h | 清除中断标志。<br>0：无影响。<br>1：清除中断标志。<br>复位类型：CPUx.SYSRSn。 |
| 1 | TEST_DONE | R-0/W1S | 0h | 清除中断标志。<br>0：无影响。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：清除中断标志。<br>复位类型：CPUx.SYSRSn。 |
| 0 | INT | R-0/W1S | 0h | 清除全局中断。<br>0：无影响。<br>1：清除中断标志，并允许事件标志置 1 时继续产生中断。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.19 BGCRC_INTFRC 寄存器

偏移地址：74h　复位值：00000000h

中断标志强制置位。

表 6-40 BGCRC_INTFRC 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | WD_OVERFLOW | WD_UNDERFLOW | CORRECTABLE_ERR | UNCORRECTABLE_ERR | CRC_FAIL | TEST_DONE | 保留 |
| R-0-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0-0h |

表 6-41 BGCRC_INTFRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | RESERVED | R-0 | 0h | 保留。 |
| 6 | WD_OVERFLOW | R-0/W1S | 0h | 强制置位中断标志。<br>0：无影响。<br>1：强制置位中断标志。<br>复位类型：CPUx.SYSRSn。 |
| 5 | WD_UNDERFLOW | R-0/W1S | 0h | 强制置位中断标志。<br>0：无影响。<br>1：强制置位中断标志。<br>复位类型：CPUx.SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 4 | CORRECTABLE_ERR | R-0/W1S | 0h | 强制置位中断标志。<br>0：无影响。<br>1：强制置位中断标志。<br>复位类型：CPUx.SYSRSn。 |
| 3 | UNCORRECTABLE_ERR | R-0/W1S | 0h | 强制置位中断标志。<br>0：无影响。<br>1：强制置位中断标志。<br>复位类型：CPUx.SYSRSn。 |
| 2 | CRC_FAIL | R-0/W1S | 0h | 强制置位中断标志。<br>0：无影响。<br>1：强制置位中断标志。<br>复位类型：CPUx.SYSRSn。 |
| 1 | TEST_DONE | R-0/W1S | 0h | 强制置位中断标志。<br>0：无影响。<br>1：强制置位中断标志。<br>复位类型：CPUx.SYSRSn。 |
| 0 | RESERVED | R-0 | 0h | 保留。 |

### 6.4.2.20 BGCRC_LOCK 寄存器

偏移地址：78h　复位值：00000000h

BGCRC 寄存器锁定配置。

表 6-42 BGCRC_LOCK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | 保留 | BGCRC_INTFRC | 保留 | 保留 | BGCRC_INTEN | 保留 | 保留 |
| R-0-0h | R-0-0h | R/W-0h | R-0-0h | R-0-0h | R/W-0h | R-0-0h | R-0-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BGCRC_NMIFRC | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | BGCRC_WD_MAX |
| R/W-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BGCRC_WD_MIN | BGCRC_WD_CFG | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 |
| R/W-0h | R/W-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BGCRC_GOLDEN | 保留 | 保留 | BGCRC_SEED | BGCRC_START_ADDR | BGCRC_CTRL2 | BGCRC_CTRL1 | BGCRC_EN |
| R/W-0h | R-0-0h | R-0-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 6-43 BGCRC_LOCK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | RESERVED | R-0 | 0h | 保留。 |
| 30 | RESERVED | R-0 | 0h | 保留。 |
| 29 | BGCRC_INTFRC | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |
| 28 | RESERVED | R-0 | 0h | 保留。 |
| 27 | RESERVED | R-0 | 0h | 保留。 |
| 26 | BGCRC_INTEN | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |
| 25 | RESERVED | R-0 | 0h | 保留。 |
| 24 | RESERVED | R-0 | 0h | 保留。 |
| 23 | BGCRC_NMIFRC | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |
| 22 | RESERVED | R-0 | 0h | 保留。 |
| 21 | RESERVED | R-0 | 0h | 保留。 |
| 20 | RESERVED | R-0 | 0h | 保留。 |
| 19 | RESERVED | R-0 | 0h | 保留。 |
| 18 | RESERVED | R-0 | 0h | 保留。 |
| 17 | RESERVED | R-0 | 0h | 保留。 |
| 16 | BGCRC_WD_MAX | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |
| 15 | BGCRC_WD_MIN | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |
| 14 | BGCRC_WD_CFG | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |
| 13 | RESERVED | R-0 | 0h | 保留。 |
| 12 | RESERVED | R-0 | 0h | 保留。 |
| 11 | RESERVED | R-0 | 0h | 保留。 |
| 10 | RESERVED | R-0 | 0h | 保留。 |
| 9 | RESERVED | R-0 | 0h | 保留。 |
| 8 | RESERVED | R-0 | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 7 | BGCRC_GOLDEN | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |
| 6 | RESERVED | R-0 | 0h | 保留。 |
| 5 | RESERVED | R-0 | 0h | 保留。 |
| 4 | BGCRC_SEED | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |
| 3 | BGCRC_START_ADDR | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |
| 2 | BGCRC_CTRL2 | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |
| 1 | BGCRC_CTRL1 | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |
| 0 | BGCRC_EN | R/W | 0h | 0：寄存器配置未锁定。<br>1：寄存器配置已锁定。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.2.21 BGCRC_COMMIT 寄存器

偏移地址：7Ch　复位值：00000000h

BGCRC 寄存器配置提交。

表 6-44 BGCRC_COMMIT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | 保留 | BGCRC_INTFRC | 保留 | 保留 | BGCRC_INTEN | 保留 | 保留 |
| R-0-0h | R-0-0h | R/WSonce-0h | R-0-0h | R-0-0h | R/WSonce-0h | R-0-0h | R-0-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BGCRC_NMIFRC | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | BGCRC_WD_MAX |
| R/WSonce-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h | R/WSonce-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BGCRC_WD_MIN | BGCRC_WD_CFG | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 |
| R/WSonce-0h | R/WSonce-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h | R-0-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BGCRC_GOLDEN | 保留 | 保留 | BGCRC_SEED | BGCRC_START_ADDR | BGCRC_CTRL2 | BGCRC_CTRL1 | BGCRC_EN |
| R/WSonce-0h | R-0-0h | R-0-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |

表 6-45 BGCRC_COMMIT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | RESERVED | R-0 | 0h | 保留。 |
| 30 | RESERVED | R-0 | 0h | 保留。 |
| 29 | BGCRC_INTFRC | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。<br>配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |
| 28 | RESERVED | R-0 | 0h | 保留。 |
| 27 | RESERVED | R-0 | 0h | 保留。 |
| 26 | BGCRC_INTEN | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。<br>配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |
| 25 | RESERVED | R-0 | 0h | 保留。 |
| 24 | RESERVED | R-0 | 0h | 保留。 |
| 23 | BGCRC_NMIFRC | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。<br>配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |
| 22 | RESERVED | R-0 | 0h | 保留。 |
| 21 | RESERVED | R-0 | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 20 | RESERVED | R-0 | 0h | 保留。 |
| 19 | RESERVED | R-0 | 0h | 保留。 |
| 18 | RESERVED | R-0 | 0h | 保留。 |
| 17 | RESERVED | R-0 | 0h | 保留。 |
| 16 | BGCRC_WD_MAX | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。<br>配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |
| 15 | BGCRC_WD_MIN | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。<br>配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |
| 14 | BGCRC_WD_CFG | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。<br>配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |
| 13 | RESERVED | R-0 | 0h | 保留。 |
| 12 | RESERVED | R-0 | 0h | 保留。 |
| 11 | RESERVED | R-0 | 0h | 保留。 |
| 10 | RESERVED | R-0 | 0h | 保留。 |
| 9 | RESERVED | R-0 | 0h | 保留。 |
| 8 | RESERVED | R-0 | 0h | 保留。 |
| 7 | BGCRC_GOLDEN | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。<br>配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |
| 6 | RESERVED | R-0 | 0h | 保留。 |
| 5 | RESERVED | R-0 | 0h | 保留。 |
| 4 | BGCRC_SEED | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。<br>配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |
| 3 | BGCRC_START_ADDR | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。<br>配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |
| 2 | BGCRC_CTRL2 | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |
| 1 | BGCRC_CTRL1 | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。<br>配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |
| 0 | BGCRC_EN | R/WSonce | 0h | 0：寄存器锁定配置未提交。<br>1：寄存器配置已提交。<br>配置一旦提交，只有复位才能更改该配置。<br>复位类型：CPUx.SYSRSn。 |

### 6.4.3 BGCRC 寄存器与 DriverLib 函数的对应关系

表 6-46 BGCRC 寄存器与 DriverLib 函数的对应关系

| 文件 | DriverLib 函数 |
|---|---|
| **EN** | |
| bgcrc.h | BGCRC_start |
| bgcrc.h | BGCRC_getRunStatus |
| **CTRL1** | |
| bgcrc.h | BGCRC_setConfig |
| **CTRL2** | |
| bgcrc.h | BGCRC_setRegion |
| bgcrc.h | BGCRC_halt |
| bgcrc.h | BGCRC_resume |
| **START_ADDR** | |
| bgcrc.h | BGCRC_setRegion |
| **SEED** | |
| bgcrc.h | BGCRC_setSeedValue |
| **GOLDEN** | |
| bgcrc.h | BGCRC_setGoldenCRCValue |
| **RESULT** | |
| bgcrc.h | BGCRC_getResult |
| **CURR_ADDR** | |
| bgcrc.h | BGCRC_getCurrentAddress |
| **WD_CFG** | |
| bgcrc.h | BGCRC_enableWatchdog |
| bgcrc.h | BGCRC_disableWatchdog |
| **WD_MIN** | |

| 文件 | DriverLib 函数 |
| --- | --- |
| bgcrc.h | BGCRC_setWatchdogWindow |
| **WD_MAX** | |
| bgcrc.h | BGCRC_setWatchdogWindow |
| **WD_CNT** | |
| bgcrc.h | BGCRC_getWatchdogCounterValue |
| **NMIFLG** | |
| bgcrc.h | BGCRC_getNMIStatus |
| **NMICLR** | |
| bgcrc.h | BGCRC_clearNMIStatus |
| **NMIFRC** | |
| bgcrc.h | BGCRC_forceNMI |
| **INTEN** | |
| bgcrc.h | BGCRC_enableInterrupt |
| bgcrc.h | BGCRC_disableInterrupt |
| **INTFLG** | |
| bgcrc.h | BGCRC_getInterruptStatus |
| **INTCLR** | |
| bgcrc.h | BGCRC_clearInterruptStatus |
| **INTFRC** | |
| bgcrc.h | BGCRC_forceInterrupt |
| **LOCK** | |
| bgcrc.h | BGCRC_lockRegister |
| bgcrc.h | BGCRC_unlockRegister |
| **COMMIT** | |
| bgcrc.h | BGCRC_commitRegisterLock |

# 7 通用输入/输出接口（GPIO）

## 7.1 简介

GPIO 模块控制设备的数字和模拟 I/O 复用，它使用共享引脚来最大限度地提高应用程序的灵活性。引脚由通用 I/O 名称命名（例如，GPIO0、GPIO25、GPIO58）。这些引脚可以单独选择作为数字 I/O（也称为 GPIO 模式）运行，也可以连接到几个外围 I/O 信号之一。输入信号可以被限定以去除不需要的噪声。

除了 CPU 控制的 I/O 功能外，单个 GPIO 启用引脚上最多可复用 12 个独立的外围信号。每个引脚输出都可以由外设或 CPU 主机控制。

最多有 8 个可能的 I/O 端口：

- Port A consists of GPIO0-GPIO31
- Port B consists of GPIO32-GPIO63
- Port C consists of GPIO64-GPIO95
- Port D consists of GPIO96-GPIO127
- Port E consists of GPIO128-GPIO159
- Port F consists of GPIO160-GPIO191
- Port G consists of GPIO192-GPIO223
- Port H consists of GPIO224-GPIO255

此设备上的模拟信号与数字输入和输出复用。其中一些模拟 IO（AIO）引脚不具备数字输出能力。其他引脚则是能够实现全数字输入和输出功能的模拟引脚（AGPIO）。具有 AIO（仅支持数字输入）功能的模拟引脚，在设备数据表的“引脚属性”表中显示为“AIO”信号。而具备全输入和输出功能(AGPIO 引脚）的模拟引脚，则在数据表的“引脚属性”表中显示为“GPIO”信号。AGPIO 引脚的名称中同时包含模拟信号和 GPIO 信号。

该设备上的模拟信号与数字输入多路复用。这些模拟 IO（AIO）引脚没有数字输出能力。模拟 IO（AIO）引脚分配给单个端口：

端口 H 由 GPIO224-GPIO247 组成。

图中显示了单个引脚的 GPIO 逻辑。

图中有两个关键特征需要注意。首先，输入和输出路径完全分开，仅在引脚处连接。第二个特性是外设复用发生在远离引脚的地方。因此，CPU 可以独立于外设复用读取引脚的物理状态。同样，外部中断也可以从外设活动中产生。所有引脚选项，如输入限定和漏极开路输出，均适用于所有外设。

表 7-1 简介数据表

| |
|---|
| JTAG使用不支持反转或限定的不同信号路径。<br>GPIO18/X2和GPIO19/X1由于振荡器电路施加在它们上的负载而具有不同的定时。<br>如果具有锐边（高dv/dt）的数字信号连接到AIO，则可能会与相邻的模拟信号发生串扰。因此，如果相邻信道用于模拟功能，请限制连接到AIO的信号的边缘速率。 |
| 在开漏模式下，GPIO不会将引脚驱动为高电平，GPIO只能将引脚拉低电平。相反，使用总线电压的外部上拉来驱动高电平。启用漏极开路模式时，GPyDAT寄存器中的值仍控制引脚状态。写入值1会关闭驱动器，以允许外部上拉控制引脚；写入值0会将引脚拉到地。I2C和PMBus等外围设备会自动使用漏极开路配置（无需在本地启用漏极开路模式）。此模式也可以通过写入GPyODR寄存器手动设置，并且可以在同一网络上有多个节点时使用，以避免推挽驱动器可能导致的引脚争用。 |

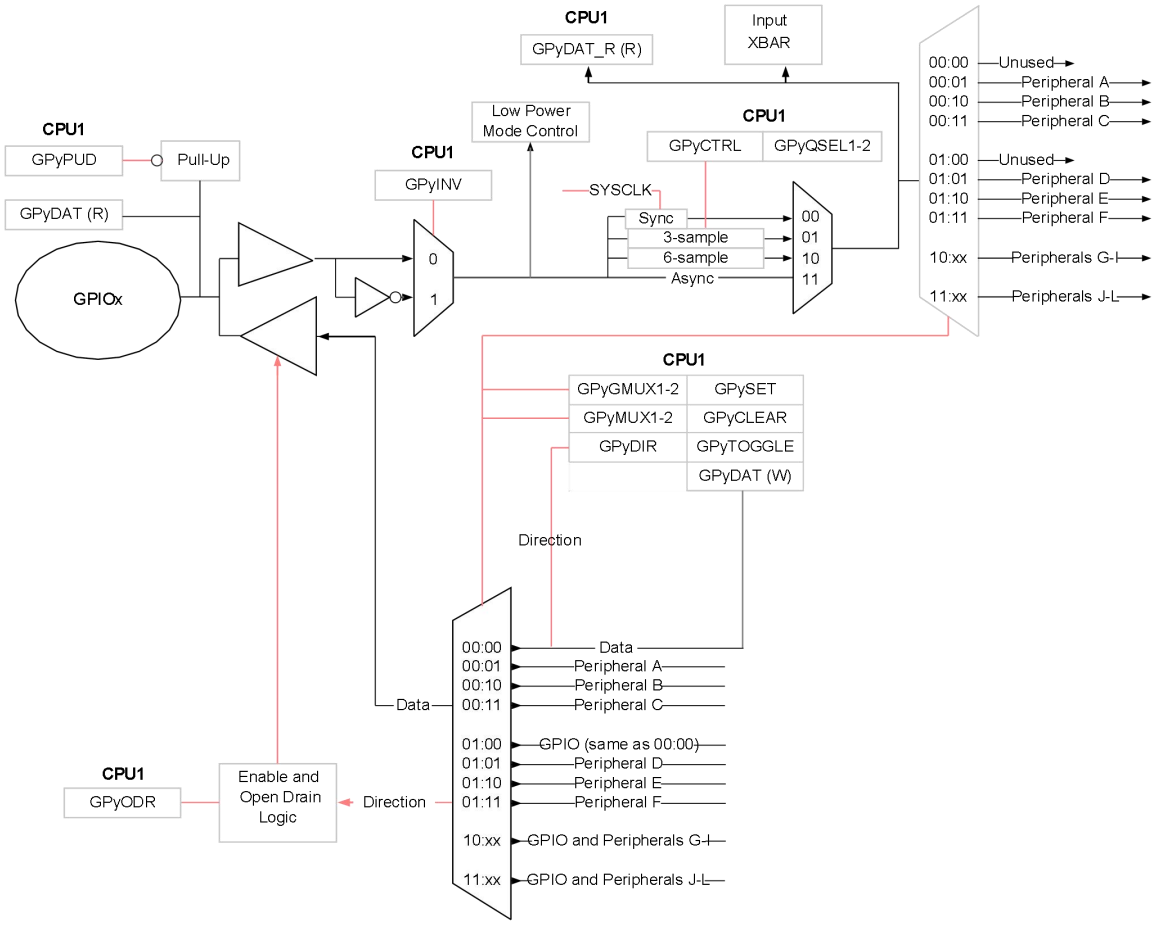



图 7-1 GPIO 逻辑图

## 7.2 配置概述

I/O 引脚配置包括几个步骤：

1. 规划设备引脚：列出应用程序所需的所有外设。使用设备数据表中的外设多路复用器信息，选择用于外设信号的 GPIO。决定将剩余的 GPIO 中的哪一个用作输入和输出。

   一旦选择了外设复用，通过将适当的值写入 GPyMUX1/2 和 GPyGMUX1/2 寄存器来实现复用。更改引脚的 GPyGMUX 值时，请始终先将相应的 GPyMUX 位设置为零，以避免多路复用器出现故障。默认情况下，所有引脚都是通用 I/O，而不是外设信号。

2. （可选）启用内部上拉电阻器：要启用或禁用上拉电阻器，请写入 GPIO 上拉禁用寄存器（GPyPUD）中的相应位。默认情况下，所有上拉都是禁用的。当没有外部信号驱动输入引脚时，可以使用上拉将输入引脚保持在已知状态。
3. 选择输入限定：如果引脚用作输入，请指定所需的输入限定（如果有的话）。在 GPyCTRL 寄存器中选择输入限定采样周期，而在 GPyQSEL1 和 GPyQSEL2 寄存器中选择限定类型。默认情况下，所有限定认证都与等于 PLLSYSCLK 的采样周期同步。
4. 选择任何通用 I/O 引脚的方向：对于配置为 GPIO 的每个引脚，使用 GPyDIR 寄存器将引脚的方向指定为输入或输出。默认情况下，所有 GPIO 引脚都是输入。在将引脚更改为输出之前，通过将要驱动的值写入 GPySET、GPyCLEAR 或 GPyDAT 寄存器来加载输出锁存器。加载锁存器后，写入 GPyDIR 以更改引脚方向。默认情况下，所有输出锁存器均为零。GPyDAT_R 寄存器可用于读取写入 GPyDAT 寄存器的值。
5. 选择低功耗模式唤醒源：GPIO 0-63 可用于从低功耗模式中唤醒系统。要选择一个或多个 GPIO 进行唤醒，请写入 GPIOLPMSEL0 和 GPIOLPMSEL1 寄存器中的相应位。这些寄存器是 CPU 系统寄存器空间的一部分。
6. 选择外部中断源：配置外部中断是一个两步过程。首先，必须启用中断本身，并且必须使用 XINTnCR 寄存器配置极性。其次，必须通过分别选择输入 X-BAR 信号 4、5、6、13 和 14 的源来设置 XINT1-5 GPIO 引脚。

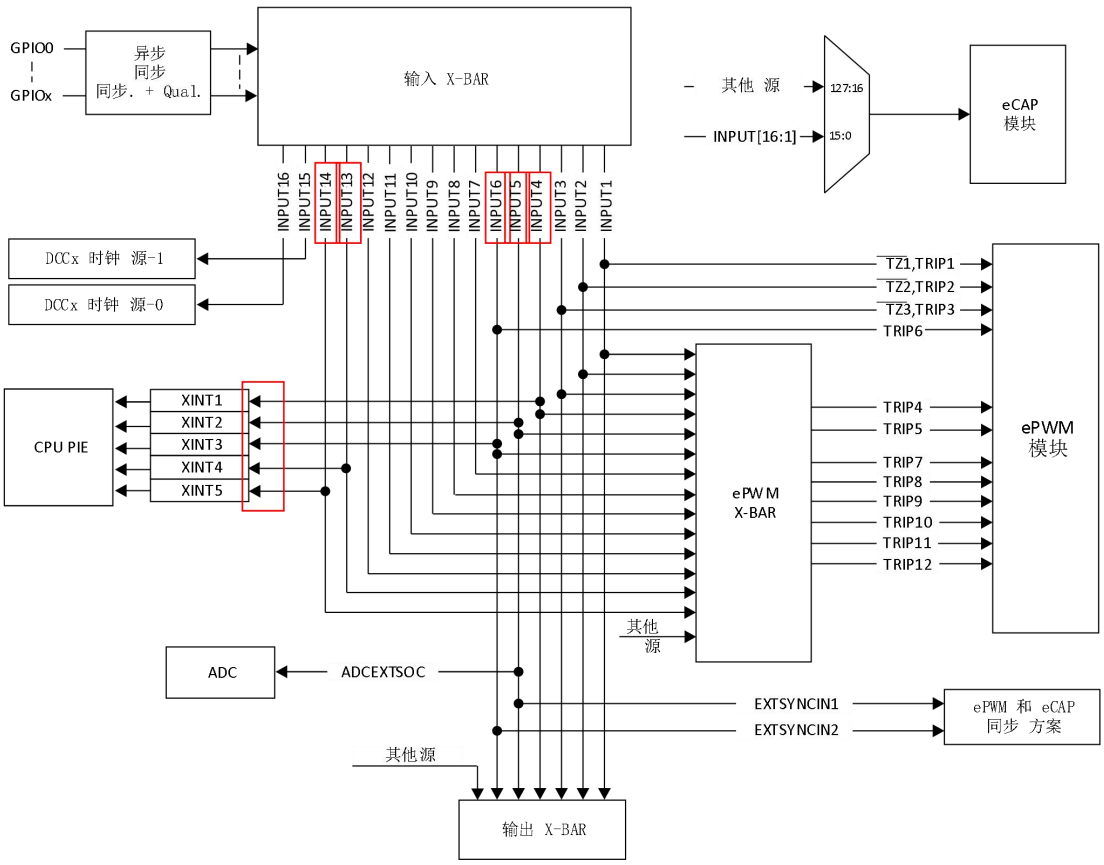



图 7-2 X-BAR 输入

## 7.3 ADC 引脚上的数字输入（AIOs）

一些 GPIO 与模拟引脚复用，仅具有数字输入功能。这些也被称为 AIO。此端口上只有 AIO 选项的引脚只能在输入模式下工作。AIO 信号列表见设备数据表。默认情况下，这些引脚用作模拟引脚，GPIO 处于高阻抗状态。GPyAMSEL 寄存器用于配置这些引脚以进行数字或模拟操作。目前只有 PortH 端口支持作为模拟信号端口。

## 7.4 ADC 引脚的数字输入和输出（AGPIOs）

一些 GPIO 与模拟引脚复用，并具有数字输入和输出功能。这些也称为 AGPIO。与 AIO 不同，AGPIO 具有完整的输入和输出能力。

默认情况下，AGPIO 未连接，必须进行配置。表中显示了如何配置 AGPIO。要启用模拟功能，请根据模拟子系统（Analog Subsystem）章节设置寄存器

AGPIOCTRLx。要启用数字功能，请根据通用输入/输出（GPIO）章节设置寄存器GPxAMSEL。

表 7-2 AGPIO 配置

| AGPIOCTRLx.GPIOy<br>（默认值 =0） | GPxAMSEL.GPIOy<br>（默认值 =1） | 引脚连接至： | |
|---|---|---|---|
| | | **ADC** | **GPIOy** |
| 0 | 0 | - | Yes |
| 0 | 1 | - | - |
| 1 | 0 | - | Yes |
| 1 | 1 | Yes | - |

从上图可以看出，只有两个寄存器的相应位域都为1时，引脚才会链接到ADC。

## 7.5 数字通用 I/O 控制

配置为 GPIO 的引脚上的值可以通过使用以下寄存器进行更改。

### 7.5.1 GPyDAT 寄存器

每个 I/O 端口都有一个数据寄存器。数据寄存器中的每个位对应一个 GPIO 引脚。数据寄存器中的每一位对应一个 GPIO 引脚。无论引脚如何配置（GPIO 或外设功能），数据寄存器中的相应位都会在经过限定后反映引脚的当前状态。写入 GPyDAT 寄存器会清除或设置相应的输出锁存器，如果该引脚被启用为通用输出（GPIO 输出），则该引脚也会被驱动为低或高。如果引脚未配置为 GPIO 输出，则值被锁存，但引脚未被驱动。只有当引脚稍后配置为 GPIO 输出时，锁存值才会被驱动到引脚上。

使用 GPyDAT 寄存器更改输出引脚的电平时，请注意不要意外更改另一个引脚的电平。例如，为了通过使用读取-修改-写入指令写入 GPADAT 寄存器位 0 来改变 GPIOA1 的输出锁存电平，如果另一个 I/O 端口 a 信号在指令的读取和写入阶段之间改变电平，则可能会出现问题。以下是对发生这种情况的原因的分析：

GPyDAT 寄存器反映的是引脚的状态，而不是锁存器的状态。这意味着寄存器反映了实际的引脚值。然而，在写入寄存器的时间与新引脚值反映回寄存器的时间之间存在滞后。当在后续程序语句中使用此寄存器来更改 GPIO 引脚的状态时，这

可能会造成问题。下面显示了一个示例，其中两个程序语句试图将两个不同的GPIO 引脚从当前的低状态驱动到高状态。

```
GpioDataRegs.GPADAT.bit.GPIO1 = 1;  // I1：对 GPADAT 执行读－修改－写
GpioDataRegs.GPADAT.bit.GPIO2 = 1;  // I2：同样对 GPADAT 执行读－修改－写
                                    // 由于延迟，GPADAT 读取到 GPIO1 的旧值
```

图 7-3 GPyDAT Registers 示意图

如果在 GPyDAT 寄存器上使用读取-修改-写入操作，由于第一条指令（I1）的输出和输入之间的延迟，第二条指令（I2）读取旧值并将其写回。

第二条指令等待第一条指令完成写入，因为该外设帧上有写入保护和读取保护。然而，在（I1）的写入和反映引脚上新值（1）的 GPyDAT 位之间存在一些滞后。在该延迟期间，第二指令读取 GPIO1（0）的旧值，并将该值与 GPIO2（1）的新值一起写回。因此，GPIO1 引脚保持低电平。

一个答案是在指令之间放置一些 NOP。更好的答案是使用 GPySET/GPyCLEAR/GPyTOGGLE 寄存器而不是 GPyDAT 寄存器。这些寄存器总是读回 0，写入 0 无效。只可以指定需要更改的位，而不会干扰当前正在更改的任何其他位。

### 7.5.2 GPyDAT_R 寄存器

GPyDAT_R 寄存器是只读寄存器，它返回写入 GPyDAT 寄存器的值，而不是引脚状态。对这些寄存器的写入无效。

### 7.5.3 GPySET 寄存器

设置寄存器用于在不干扰其他引脚的情况下将指定的 GPIO 引脚驱动为高电平。每个 I/O 端口都有一个设置寄存器，每个位对应一个 GPIO 引脚。设置（set）寄存器总是读回 0。如果相应的引脚被配置为输出，那么在置位寄存器中向该位写入 1 会将输出锁存器设置为高电平，并且相应的引脚会被驱动为高电平。如果引脚未配置为 GPIO 输出，则值被锁存，但引脚未被驱动。只有当引脚稍后配置为 GPIO 输出时，锁存值才会被驱动到引脚上。将 0 写入集合寄存器中的任何位都没有效果。

### 7.5.4 GPyCLEAR 寄存器

清除寄存器用于将指定的 GPIO 引脚驱动为低电平，而不会干扰其他引脚。每个 I/O 端口都有一个透明寄存器。清除寄存器总是读回 0。如果相应的引脚被配置为通用输出，则向清除寄存器中的相应位写入 1 会清除输出锁存器，并且引脚被驱动为低电平。如果引脚未配置为 GPIO 输出，则值被锁存，但引脚未被驱动。只有当引脚稍后配置为 GPIO 输出时，锁存值才会被驱动到引脚上。将 0 写入清除寄存器中的任何位都没有效果。

### 7.5.5 GPyTOGGLE 寄存器

切换寄存器用于将指定的 GPIO 引脚驱动到相反的电平，而不会干扰其他引脚。每个 I/O 端口都有一个切换寄存器。切换寄存器总是读回 0。如果相应的引脚被配置为输出，那么在切换寄存器中向该位写入 1 会翻转输出锁存器，并沿相反方向拉动相应的引脚。也就是说，如果输出引脚被驱动为低电平，那么向切换寄存器中的相应位写入 1 会将引脚拉高。同样，如果输出引脚为高，则向切换寄存器中的相应位写入 1 会将引脚拉低。如果引脚未配置为 GPIO 输出，则值被锁存，但引脚未被驱动。只有当引脚稍后配置为 GPIO 输出时，锁存值才会被驱动到引脚上。将 0 写入切换寄存器中的任何位都没有效果。

## 7.6 输入限定

输入限定方案的设计非常灵活。通过配置 GPyQSEL1 和 GPyQSEL2 寄存器，为每个 GPIO 引脚选择输入限定类型。在 GPIO 输入引脚的情况下，可以将限定指定为仅与 SYSCLKOUT 同步或通过采样窗口进行限定认证。对于配置为外设输入的引脚，除了与 SYSCLKOUT 同步或由采样窗口限定外，输入还可以是异步的。本节的其余部分描述了可用的选项。

### 7.6.1 非同步（异步输入）

此模式用于不需要输入同步或自己执行同步的外设。示例包括通信端口 McBSP、SCI、SPI 和 I2C。此外，ePWM 跳闸区（TZn）信号可以独立于 SYSCLKOUT 的存在而工作。

| 注意：<br>在外设执行同步时，使用输入同步可能会导致意外结果。在这种情况下，用户必须确保 GPIO引脚配置为异步。 |
|---|

### 7.6.2 仅与 SYSCLKOUT 同步

这是重置时所有引脚的默认限定模式。在此模式下，输入信号仅与系统时钟（SYSCLKOUT）同步。由于输入信号是异步的，因此需要一个 SYSCLKOUT 周期的延迟来改变设备的输入。不对信号进行进一步的限定。

### 7.6.3 使用采样窗口进行限定

在这种模式下，信号首先与系统时钟（SYSCLKOUT）同步，然后在允许输入更改之前经过指定数量的周期。图显示了如何进行输入限定以消除不需要的噪声。用户为这种类型的限定指定了两个参数：1）采样周期，即相邻两次采样之间的时间间隔，2）要采集的样本数量。

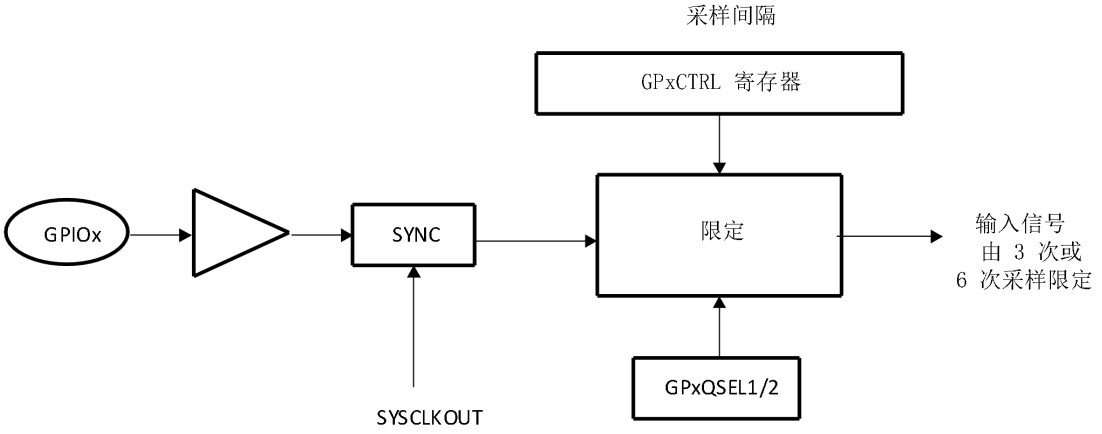



图 7-4 消除噪声

#### 7.6.3.1 采样周期

为了使信号合格，以规则的周期对输入信号进行采样。采样周期由用户指定，并决定采样之间的持续时间，或相对于 CPU 时钟（SYSCLKOUT）对信号进行采样的频率。

采样周期由 GPxCTRL 寄存器中的限定期（QUALPRDn）位指定。采样周期可配置为 8 个输入信号的组。例如，GPIO0 至 GPIO7 使用 GPACTRL[QUALPRD0]设置，GPIO8 至 GPIO15 使用 GPACTRL[QUALPRD1]设置。表显示了采样周期或采样频率与 GPxCTRL[QUALPRDn]设置之间的关系。

表 7-3 采样周期（时长）

| 项目 | 条件 | 计算公式 |
|---|---|---|
| 采样周期 | GPxCTRL[QUALPRDn] = 0 | 1 × T_SYSCLKOUT |
| 采样周期 | GPxCTRL[QUALPRDn] ≠ 0 | 2 × GPxCTRL[QUALPRDn] × T_SYSCLKOUT |

其中，T_SYSCLKOUT 为 SYSCLKOUT 的时钟周期，f_SYSCLKOUT 为 SYSCLKOUT 的时钟频率。

表 7-4 采样频率

| 项目 | 条件 | 计算公式 |
|---|---|---|
| 采样频率 | GPxCTRL[QUALPRDn] = 0 | f_SYSCLKOUT |
| 采样频率 | GPxCTRL[QUALPRDn] ≠ 0 | f_SYSCLKOUT ÷ (2 × GPxCTRL[QUALPRDn]) |

| GPxCTRL寄存器值的规律：值0当1，非零值左移1位。 |
|---|

根据这些方程，可以计算出给定 SYSCLKOUT 频率下采样之间的最小和最大时间。

表 7-5 采样周期

| 项目 | 条件与计算结果 |
|---|---|
| 示例：最大采样频率 | 若 GPxCTRL[QUALPRDn] = 0，则采样频率为 f_SYSCLKOUT。<br>例如，f_SYSCLKOUT = 60 MHz，则以 60 MHz 采样，即每 16.67 ns 采样一次。 |
| 示例：最小采样频率 | 若 GPxCTRL[QUALPRDn] = 0xFF（255），则采样频率为 f_SYSCLKOUT × 1 ÷ (2 × GPxCTRL[QUALPRDn])。<br>例如，f_SYSCLKOUT = 60 MHz，则采样频率为 60 MHz × 1 ÷ (2 × 255) = 117.647 kHz，即每 8.5 μs 采样一次。 |

### 7.6.3.2 采样数

信号采样的次数是限定选择（GPAQSEL1、GPAQSEL2、GPBQSEL1 和 GPBQSEL2）寄存器中指定的三个样本或六个样本。当连续三到六个周期相同时，输入变化会传递给设备。

### 7.6.3.3 总采样窗口宽度

采样窗口是对输入信号进行采样的时间。通过使用采样周期的方程以及要采集的样本数量，可以确定窗口的总宽度。

为了使输入限定符检测输入的变化，信号的电平必须在采样窗口宽度或更长时间内保持稳定。

窗口内的采样周期数总是比采集的样本数少一个。对于三采样窗口，采样窗口宽度为两个采样周期宽。同样，对于六采样窗口，采样窗口宽度为五个采样周期宽。表显示了用于根据 GPxCTRL[QUALPRDn]和取样数量确定总取样窗口宽度的计算。

表 7-6 三采样采样窗口宽度

| 项目 | 条件 | 计算公式 |
|---|---|---|
| 总采样窗口宽度 | GPxCTRL[QUALPRDn] = 0 | 2 × T_SYSCLKOUT |
| 总采样窗口宽度 | GPxCTRL[QUALPRDn] ≠ 0 | 2 × 2 × GPxCTRL[QUALPRDn] × T_SYSCLKOUT |

表 7-7 六采样采样窗口宽度

| 项目 | 条件 | 计算公式 |
|---|---|---|
| 总采样窗口宽度 | GPxCTRL[QUALPRDn] = 0 | 5 × T_SYSCLKOUT |
| 总采样窗口宽度 | GPxCTRL[QUALPRDn] ≠ 0 | 5 × 2 × GPxCTRL[QUALPRDn] × T_SYSCLKOUT |

其中，T_SYSCLKOUT 为 SYSCLKOUT 的时钟周期

注意：
外部信号变化相对于采样周期和SYSCLKOUT都是异步的。由于外部信号的异步特性，输入必须在大于采样窗口宽度的时间内保持稳定，以确保逻辑检测到信号的变化。所需的额外时间最多可达额外的采样周期+TSYSCLKOUT。
数据表中描述了输入信号稳定所需的持续时间，以便限定逻辑检测到变化。

限定窗口案例：

对于所示的示例，输入限定配置如下：

- GPxQSEL1/2 = 1.0。这表示六样本限定。

- GPxCTRL[QUALPRDn] = 1。采样周期为 tw(SP)=2×GPxCTRL[QUALPRDn] ×TSYSCLKOUT=2×TSYSCLKOUT

此配置得出以下结果：

- 采样窗口宽度为

tw(IQSW)=5 × tw(SP)=5 × 2 × GPxCTRL[QUALPRDn] × TSYSCLKOUT=5 × 2 × TSYSCLKOUT

- 例如，若 TSYSCLKOUT=16.67ns，则采样窗口的持续时间为：采样周期，tw(SP)=2×TSYSCLKOUT=2×16.67 ns=33.3 ns，采样窗口，tw(IQSW)=5×tw(SP)=5×33.3 ns=166.7 ns。
- 考虑到输入相对于采样周期和 SYSCLKOUT 的异步特性，检测输入信号变化最多需要额外一个采样周期和一个 SYSCLK 周期。对于此示例：

tw(IQSW)+tw(SP)+TSYSCLKOUT=166.7 ns+33.3 ns+16.67 ns=216.7 ns

- 毛刺（A）的持续时间短于限定窗口，因此被输入限定器忽略

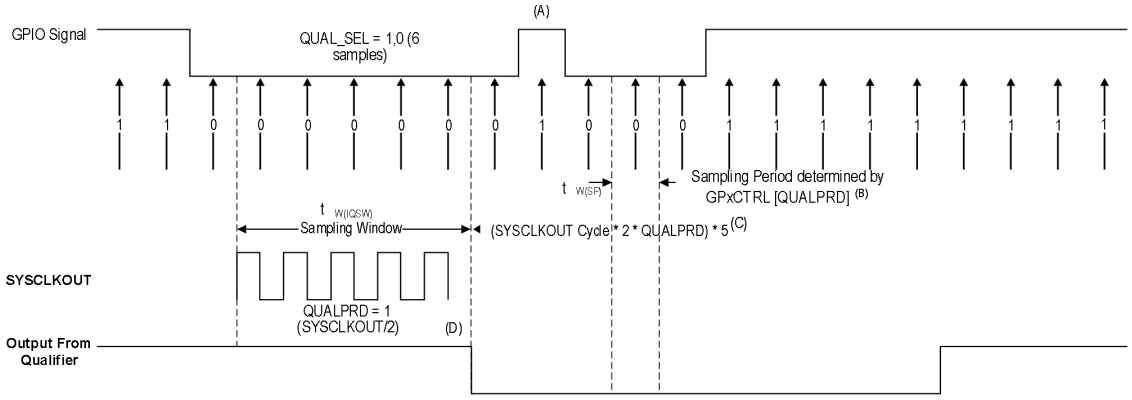



图 7-6 采样时钟

A. 该毛刺会被输入限定器忽略。QUALPRD 位域指定了限定采样周期，其取值范围为 00 到 0xFF。若 QUALPRD = 00，则采样周期为 1 个 SYSCLKOUT 周期。对于其他值 “n”，限定采样周期为 2n 个 SYSCLKOUT 周期（即每经过 2n 个 SYSCLKOUT 周期，对 GPIO 引脚采样一次）。

B. 通过 GPxCTRL 寄存器选择的限定周期适用于 8 个 GPIO 引脚为一组。

C. 限定模块可采用三个或六个样本。QUAL_SEL 寄存器用于选择使用哪种采样模式。

D. 在所示示例中，为使限定器检测到变化，输入应保持稳定达到或超过 10 个 SYSCLKOUT 周期。换言之，输入应保持稳定持续（5 × QUALPRD × 2）个 SYSCLKOUT 周期。这样可为检测提供 5 个采样周期。由于外部信号是异步驱动的，因此 13 个 SYSCLKOUT 周期宽的脉冲可确保可靠识别。

## 7.7 GPIO 和外设复用

### 7.7.1 GPIO 复用

注意：

每个引脚最多可复用十二种不同的外围功能以及通用输入/输出（GPIO）功能。这允许您选择最适合您特定应用的外围设备组合和引脚。复用组合和定义见表中。

表 7-8 GPIO 复用列表

| 0, 4, 8,12 | 1 | 2 | 3 | 5 | 6 | 7 | 9 | 10 | 11 | 13 | 14 | 15-0 | 备选功能（ALT） |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIOO | EPWM1_A | CANA_RX | OUTPUTXBAR7 | SCIA_RX | I2CA_SDA | SPIA_STE | FSIRXACLK | | CLB OUTPUT XBAR8 | EQEP1INDEX | | HIC BASE SEL1 | |
| GPIO1 | EPWM1_B | | | SCIA_TX | I2CA_SCL | SPIA_SOMI | EQEP1_STROBE | | CLB OUTPUT XBAR7 | HIC_A2 | FSITXA TDM_D1 | HIC_D10 | |
| GPIO2 | EPWM2_A | | | OUTPUT XBAR1 | PMBUSASDA | SPIA_SIMO | SCIA_TX | FSIRXA_D1 | I2CB_SDA | HIC_A1 | CANA_TX | HIC_D9 | |
| GPIO3 | EPWM2_B | OUTPUT XBAR2 | | OUTPUT XBAR2 | PMBUSASCL | SPIA_CLK | SCIA_RX | FSIRXA_D0 | I2CB_SCL | HIC_NOE | CANA_RX | HIC_D4 | |
| GPIO4 | EPWM3_A | I2CA_SCL | | OUTPUT XBAR3 | CANA_TX | SPIB_CLK | EQEP2 STROBE | FSIRXACLK | CLB OUTPUT XBAR6 | HIC BASE SEL2 | SPIA_SOMI | HIC_NWE | |
| GPIO5 | EPWM3_B | I2CA_SDA | OUTPUT XBAR3 | | CANA_RX | SPIA_STE | FSITXA_D1 | CLB OUTPUT BAR5 | SCIA_RX | HIC_A7 | HIC_D4 | HIC_D15 | |
| GPIO6 | EPWM4_A | OUTPUT BAR4 | SYNCOUT | EQEP1_A | | SPIB_SOMI | FSITXA_D0 | | FSITXA_D1 | HIC_NBE1 | CLB OUTPUT XBAR8 | HIC_D14 | |
| GPIO7 | EPWM4_B | EPWM2_A | OUTPUT XBAR5 | EQEP1_B | | SPIB_SIMO | FSITXACLK | CLB OUTPUT BAR2 | SCIA_TX | HIC_A6 | CANA_TX | HIC_D14 | |
| GPIO8 | EPWM5_A | | ADCSOCAO | EQEP1 STROBE | SCIA_TX | SPIA_SIMO | I2CA_SCL | FSITXA_D1 | CLB OUTPUT XBAR5 | HIC_A0 | FSITXA TDM_CLK | HIC_D8 | |
| GPIO9 | EPWM5_B | | OUTPUT XBAR6 | EQEP1 INDEX | SCIA_RX | SPIA_CLK | | FSITXA_D0 | LINB_RX | HIC BASE SELO | I2CB_SCL | HIC_NRDY | |
| GPIO10 | EPWM6_A | | ADCSOCBO | EQEP1_A | | SPIA_SOMI | I2CA_SDA | FSITXACLK | LINB_TX | HIC_NWE | FSITXA TDM D0 | | |
| GPIO11 | EPWM6_B | CANA_TX | OUTPUT XBAR7 | EQEP1_B | | SPIA_STE | FSIRXA_D1 | LINB_RX | EQEP2_A | SPIA_SIMO | HIC_D6 | HIC_NBE0 | |
| GPIO12 | EPWM7_A | | | EQEP1 STROBE | | PMBUSACTL | FSIRXA_D0 | LINB_TX | SPIA_CLK | CANA_RX | HIC_D13 | HIC_INT | A20,C20 |
| GPIO13 | EPWM7_B | | | EQEP1 INDEX | | PMBUSAALERT | FSIRXACLK | LINB_RX | SPIA_SOMI | CANA_TX | HIC_D11 | HIC_D5 | A19,C19 |

| 0, 4, 8,12 | 1 | 2 | 3 | 5 | 6 | 7 | 9 | 10 | 11 | 13 | 14 | 15-0 | 备选功能（ALT） |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO14 | | | | I2CB_SDA | OUTPUT XBAR3 | PMBUSA SDA | SPIB_CLK | EQEP2_A | LINB_TX | EPWM3_A | CLB OUTPUT XBAR7 | HIC_D15 | |
| GPIO15 | | | | I2CB_SCL | OUTPUT XBAR4 | PMBUSA SCL | SPIB_STE | EQEP2_B | LINB_RX | EPWM3_B | CLB OUTPUT XBAR6 | HIC_D12 | |
| GPIO16 | SPIA_SIMO | | OUTPUT XBAR7 | EPWM5_A | SCIA_TX | | EQEP1 STROBE | PMBUSA SCL | XCLKOUT | EQEP2_B | SPIB_SOMI | HIC_D1 | |
| GPIO17 | SPIA_SOMI | | OU IPU1 XBAR8 | EPWM5_B | SCIA_RX | | EQEP1 INDEX | PMBUSA SDA | CANA_TX | | EPWM6_A | HIC_D2 | |
| GPIO18_X2 | SPIA_CLK | | CANA_RX | EPWM6_A | I2CA_SCL | | EQEP2_A | PMBUSA CTL | XCLKOUT | LINB_TX | FSITXA TDM CLK | HIC_INT | X2 |
| GPIO19_X1 | SPIA_STE | | CANA_TX | EPWM6_B | I2CA_SDA | | EQEP2_B | PMBUSA ALERT | CLB OUTPUT XBAR1 | LINB_RX | FSITXA TDM_D0 | HIC_NBE0 | X1 |
| GPIO20 | EQEP1_A | | CANA_TX | | SPIA_SIMO | | | | I2CA_SCL | | | LINA_TX | A17,C17 |
| GPIO21 | EQEP1_B | | CANA_RX | | SPIA_SOMI | | | | I2CA_SDA | | | LINA_RX | A18,C18 |
| GPIO22 | EQEP1 STROBE | | | | SPIB_CLK | | LINA_TX | CLB OUTPUT XBAR1 | LINB_TX | HIC_A5 | EPWM4_A | HIC_D13 | |
| GPIO23 | EQEP1 INDEX | | | | SPIB_STE | | LINA_RX | | LINB_RX | HIC_A3 | EPWM4_B | HIC_D11 | |
| GPIO24 | OUTPUT XBAR1 | EQEP2_A | SPIA_STE | EPWM4_A | SPIB_SIMO | | LINB_TX | PMBUSA SCL | SCIA_TX | ERRORSTS | | HIC_D3 | |
| GPIO25 | OUTPUT XBAR2 | EQEP2_B | | EQEP1_A | SPIB_SOMI | | FSITXA_D1 | PMBUSA SDA | SCIA_RX | | HIC BASE SELO | | |
| GPIO26 | OUTPUT XBAR3 | EQEP2 INDEX | | OUTPUT XBAR3 | SPIB_CLK | | FSITXA_D0 | PMBUSA CTL | I2CA_SDA | | HIC_D0 | HIC_A1 | |
| GPIO27 | OUTPUT XBAR4 | EQEP2 STROBE | | OUTPUT XBAR4 | SPIB_STE | | FSITXA CLK | PMBUSA ALERT | I2CA_SCL | | HIC_D1 | HIC_A4 | |
| GPIO28 | SCIA_RX | | EPWM7_A | OUTPUT XBAR5 | EQEP1_A | | EQEP2 STROBE | LINA_TX | SPIB_CLK | ERRORSTS | I2CB_SDA | HIC_NOE | A16,C16 |

| 0, 4, 8,12 | 1 | 2 | 3 | 5 | 6 | 7 | 9 | 10 | 11 | 13 | 14 | 15-0 | 备选功能（ALT） |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO29 | SCIA_TX | | EPWM7_B | OUTPUT XBAR6 | EQEP1_B | | EQEP2 INDEX | LINA_RX | SPIB_STE | ERRORS TS | I2CB_SCL | HIC_NCS | |
| GPIO30 | CANA_RX | | SPIB_SIM O | OUTPUT XBAR7 | EQEP1 STROBE | | FSIRXA CLK | | EPWM1_A | | HIC_D8 | | |
| GPIO31 | CANA_TX | | SPIB_.SO MI | OUTPUT XBAR8 | EQEP1 INDEX | | FSIRXA _D1 | | EPWM1_B | | HIC_D10 | | |
| GPIO32 | I2CA_ SDA | EQEP1IN DEX | SPIB CLK | EPWM4_B | LINA_TX | | FSIRXA D0 | CANA TX | | ADCSOC BO | | HIC INT | |
| GPIO33 | I2CA_SCL | | SPIB_STE | OUTPUT XBAR4 | LINA_RX | | FSIRXA CLK | CANA_R X | EQEP2_B | ADCSOC AO | | HIC_D0 | |
| GPIO34 | OUTPUT XBAR1 | | | | PMBUSA SDA | | | | | HIC_NB E1 | I2CB_SDA | HIC_D9 | |
| GPIO35 | SCIA_RX | SPIA_SO MI | I2CA_SDA | CANA_RX | PMBUSA SCL | LINA_R X | EQEP1_ A | PMBUSA CTL | EPWM5_B | | HIC_NWE | TDI(**仿真器)** | |
| GPIO37 | OUTPUT XBAR2 | SPIA_ST E | I2CA_SCL | SCIA_TX | CANA_T X | LINA_TX | EQEP1_ B | PMBUSA LERT | EPWM5_A | | HIC_NRDY | TDO | |
| GPIO39 | | | | | | FSIRXA CLK | EQEP2 INDEX | | CLB OUTPUT XBAR2 | SYNCOU T | EQEP1 INDEX | HIC_D7 | |
| GPIO40 | SPIB_SIMO | | | EPWM2_B | PMBUSA SDA | FSIRXA_ D0 | | EQEP1_A | LINB_TX | | HIC_NBE1 | HIC_D5 | |
| GPIO41 | EPWM7_A | | | EPWM2_A | PMBUSA SCL | FSIRXA_ D1 | | EQEP1_B | LINB_RX | HIC_A4 | SPIB_SOMI | HIC_D12 | |
| GPIO42 | | LINA_R X | OUTPUT XBAR5 | PMBUSA CTL | I2CA_SD A | | | EQEP1 STROBE | CLB OUTPUT XBAR3 | | HIC_D2 | HIC_A6 | |
| GPIO43 | | | OUTPUT XBAR6 | PMBUSA ALERT | I2CA_SC L | | | EQEP1 INDEX | CLB OUTPUT XBAR4 | | HIC_D3 | HIC_A7 | |
| GPIO44 | | | OUTPUT XBAR7 | EQEP1_A | | FSITXA CLK | | CLB OUTPUT BAR3 | | HIC_D7 | | HIC_D5 | |
| GPIO45 | | | OUTPUT XBAR8 | | | FSITXA_ D0 | | CLB OUTPUT XBAR4 | | | | HIC_D6 | |

| 0, 4, 8,12 | 1 | 2 | 3 | 5 | 6 | 7 | 9 | 10 | 11 | 13 | 14 | 15-0 | 备选功能（ALT） |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO46 | | | LINA TX | | | FSITXAD1 | | | | | | HIC NWE | |
| GPIO61 | | | | | | | | | | | | | |
| GPIO62 | | | | | | | | | | | | | |
| GPIO63 | | | | | | | | | | | | | |
| GPIO224 | | | | OUTPUTXBAR3 | SPIA_SIMO | | EPWM1_A | CANA_TX | EQEP1_A | | LINA_TX | HIC_A3 | A2,C9 |
| GPIO225 | | | | | | | | | | | | HIC_NWE | A4,C14 |
| GPIO226 | | | | EPWM6_A | SPIA_CLK | | EPWM1_B | | EQEP1_STROBE | | LINA_RX | HIC_A1 | C6 |
| GPIO227 | I2CB_SCL | | EPWM3_A | OUTPUTXBAR1 | EPWM2_B | | | | | | | HIC_NBE0 | A9,C8 |
| GPIO228 | | | ADCSOCAO | CANA_TX | SPIA_SOMI | | EPWM2_B | | EQEP1_B | | | HIC_A0 | A6 |
| GPIO230 | I2CB_SDA | | EPWM3_B | CANA_RX | EPWM2_A | I2CA_SDA | | | | | | HIC_BASESEL2 | A10,C10 |
| AIO231 | | | | | | | | | | | | HIC_BASESEL1 | A0,C15 |
| AIO232 | | | | | | | | | | | | HIC_BASESEL0 | A1 |
| AIO233 | | | | | | | | | | | | HIC_A4 | A15,C7 |
| AIO237 | | | | | | | | | | | | HIC_A6 | A11,C0 |
| AIO238 | | | | | | | | | | | | HIC_NCS | A12,C1 |
| AIO239 | | | | | | | | | | | | HIC_A5 | A14,C4 |
| AIO241 | | | | | | | | | | | | HIC_NBE1 | A8,C11 |
| GPIO242 | | | | OUTPUTXBAR2 | SPIA_STE | | EPWM4_A | CANA_RX | EQEP1INDEX | | | HIC_A2 | A3,C5 |
| AIO244 | | | | | | | | | | | | HIC_A7 | A5,C2 |

| 0, 4, 8,12 | 1 | 2 | 3 | 5 | 6 | 7 | 9 | 10 | 11 | 13 | 14 | 15-0 | 备选功能（ALT） |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AIO245 | | | | | | | | | | | | HIC_NOE | A7,C3 |

### 7.7.2 外设复用

例如，GPIO6 引脚的多路复用是通过写入 GPAGMUX[13:12]和 GPAMUX[13:12]来控制的。通过写入这些位，GPIO6 被配置为通用数字 I/O 或几种不同外围功能之一。表中显示了单个 GPIO 的 GPyGMUX 和 GPyMUX 选择和选项的示例。

表 7-9 GPIO6 引脚复用选择

| GPAGMUX1[13:12] | GPAMUX1[13:12] | 引脚功能 |
|---|---|---|
| 00 | 00 | GPIO6 |
| 00 | 01 | Peripheral1 |
| 00 | 10 | Peripheral2 |
| 00 | 11 | Peripheral3 |
| 01 | 00 | GPIO6 |
| 01 | 01 | Peripheral4 |
| 01 | 10 | Peripheral5 |
| 01 | 11 | |
| 10 | 00 | GPIO6 |
| 10 | 01 | |
| 10 | 10 | Peripheral6 |
| 10 | 11 | Peripheral7 |
| 11 | 00 | GPIO6 |
| 11 | 01 | Peripheral8 |
| 11 | 10 | Peripheral9 |
| 11 | 11 | Peripheral10 |

这些设备具有不同的多路复用方案。如果特定设备上没有可用的外设，则该多路复用器选择将保留在该设备上，不得使用。

| 注意：<br>如果选择了未映射到外围设备或GPIO模式的保留GPIO多路复用器配置，则引脚的状态未定义，引脚被驱动。未实现的配置用于未来的扩展，不得选择。在设备多路复用表（见数据表）中，这些选项显示为保留或留空。 |
|---|

一些外围设备可以通过多路复用寄存器分配给多个引脚。例如，根据各个系统的要求，可以将 OUTPUTXBAR1 分配给 GPIO p、q 或 r（其中 p、q 和 r 是示例 GPIO 编号）。表中展示了一个示例。

| 注意：下表仅为示例。位范围不能与此设备上的OUTPUTXBAR1对应。有关此设备的GPIO和相应多路复用器选项的正确列表。 |
|---|

如果没有或多个 GPIO 引脚被配置为外围输入引脚，则该 GPIO 被设置为硬连线默认值。如表中所示。

表 7-10 外设多路复用

| GMUX 配置 | MUX 配置 | |
|---|---|---|
| choice 1:GPIOp | GpyGMUX1[5:4]=01 | GpyMUX1[5:4]=01 |
| or choice 2:GPIOq | GpyGMUX2[17:16]=00 | GpyMUX2[17:16]=01 |
| or choice 3:GPIOr | GpyGMUX1[7:6]=01 | GpyMUX1[7:6]=01 |

## 7.8 内部上拉配置要求

重置时，GPIO 处于输入模式，并禁用内部上拉。未驱动的输入可能会浮动到中轨电压，并导致输入缓冲器上的直通电流浪费。用户必须始终将每个 GPIO 置于以下配置之一：

输入模式，由另一个组件在板上驱动到高于 Vih 或低于 Vil 的水平

启用 GPIO 内部上拉的输入模式

输出模式

在引脚数较少的设备上，默认情况下启用封装中未引出的 GPIO 的上拉，以防止浮动输入。用户必须注意避免在应用程序代码中禁用这些上拉。

在具有较大引脚数封装的设备上，必须为所有内部未引出的 GPIO 启用上拉，以防止浮动输入。用户必须注意避免在应用程序代码中禁用这些上拉。

表 7-11 引脚初始上拉设置

| 引脚 | 复位(XRSn=0) | 器件启动 | 应用程序 |
|---|---|---|---|
| GPIOx | pullup disabled | pullup disabled | application defined |
| GPIO35/TDI | pullup disabled | | application defined |
| GPIO37/TDO | pullup disabled | | application defined |
| TCK | pullup active | | |
| TMS | pullup active | | |
| XRSn | pullup active | | |
| Other pins(including AIOs) | No pullup or pulldown present | | |

## 7.9 寄存器描述

### 7.9.1 GPIO 基地址表

表 7-12 GPIO 基地址表

| 比特域名称 | | 基地址 |
|---|---|---|
| 例化名 | 结构体 | |
| GpioCtrlRegs | GPIO_CTRL_REGS | 0x0064_0800 |
| GpioDataRegs | GPIO_DATA_REGS | 0x0064_0C00 |
| GpioDataReadRegs | GPIO_DATA_READ_REGS | 0x0064_0D00 |

### 7.9.2 GPIO 控制寄存器

表 7-13 GPIO 控制寄存器

| 偏移量 | 寄存器 | 描述 |
|---|---|---|
| 0h | GPACTRL | GPIO A采样周期控制(GPIO0至31) |
| 4h | GPAQSEL1 | GPIO A输入限定选择1(GPIO0至15) |
| 8h | GPAQSEL2 | GPIO A输入限定选择2(GPIO16至31) |
| Ch | GPAMUX1 | GPIO A复用选择1低2位(GPIO0至15) |
| 10h | GPAMUX2 | GPIO A复用选择2低2位(GPI16至31) |
| 14h | GPADIR | GPIO A方向选择 |
| 18h | GPAPUD | GPIO A上拉禁用 |
| 20h | GPAINV | GPIO A输入极性翻转 |
| 24h | GPAODR | GPIO A开漏输出 |
| 28h | GPAAMSEL | GPIO A模拟输入 |
| 40h | GPAGMUX1 | GPIO A复用选择高12位(GPIO0至15) |
| 44h | GPAGMUX2 | GPIO A复用选择2高2位(GPI16至31) |
| 78h | GPALOCK | GPIO A配置锁定 |
| 7Ch | GPACR | GPIO A提交锁定配置 |
| 80h | GPBCTRL | GPIO B采样周期控制(GPIO0至31) |
| 84h | GPBQSEL1 | GPIO B输入限定选择1(GPIO0至15) |
| 88h | GPBQSEL2 | GPIO B输入限定选择2(GPIO16至31) |
| 8Ch | GPBMUX1 | GPIO B复用选择1低2位(GPIO0至15) |
| 90h | GPBMUX2 | GPIO B复用选择2低2位(GPI16至31) |
| 94h | GPBDIR | GPIO B方向选择 |
| 98h | GPBPUD | GPIO B上拉禁用 |
| A0h | GPBINV | GPIO B输入极性翻转 |
| A4h | GPBODR | GPIO B开漏输出 |
| C0h | GPBGMUX1 | GPIO B复用选择高12位(GPIO0至15) |
| C4h | GPBGMUX2 | GPIO B复用选择2高2位(GPI16至31) |
| F8h | GPBLOCK | GPIO B配置锁定 |
| FCh | GPBCR | GPIO B提交锁定配置 |
| 380h | GPHCTRL | GPIO H采样周期控制(GPIO0至31) |
| 384h | GPHQSEL1 | GPIO H输入限定选择1(GPIO0至15) |
| 388h | GPHQSEL2 | GPIO H输入限定选择2(GPIO16至31) |
| 38Ch | GPHMUX1 | GPIO H复用选择1低2位(GPIO0至15) |
| 390h | GPHMUX2 | GPIO H复用选择2低2位(GPI16至31) |
| 394h | GPHDIR | GPIO H方向选择 |
| 398h | GPHPUD | GPIO H上拉禁用 |
| 3A0h | GPHINV | GPIO H输入极性翻转 |
| 3A4h | GPHODR | GPIO H开漏输出 |
| 3A8h | GPHAMSEL | GPIO H模拟输入 |
| 3C0h | GPHGMUX1 | GPIO H复用选择高12位(GPIO0至15) |
| 3C4h | GPHGMUX2 | GPIO H复用选择2高2位(GPI16至31) |
| 3F8h | GPHLOCK | GPIO H配置锁定 |

| 偏移量 | 寄存器 | 描述 |
| --- | --- | --- |
| 3FCh | GPHCR | GPIO H提交锁定配置 |

### 7.9.2.1 GPACTRL 寄存器

偏移地址：0h　复位值：00000000h

表 7-14 GPACTRL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| QUALPRD3 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QUALPRD2 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QUALPRD1 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QUALPRD0 | | | | | | | |
| R-0h | | | | | | | |

表 7-15 GPACTRL

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 31-24 | QUALPRD3 | R/W | 0h | GPIO24至GPIO31的合格采样周期：<br>0x00，QUALPRDx=PLLSYSCLK<br>0x01，QUALPRDx=PLLSYSCLK/2<br>0x02，QUALPRDx=PLLSYSCLK/4<br>...<br>0xFF，QUALPRDx=PLLSYSCLK/510<br>重置类型：SYSRSn |
| 23-16 | QUALPRD2 | R | 0h | GPIO16至GPIO23的合格采样周期：<br>0x00，QUALPRDx=PLLSYSCLK<br>0x01，QUALPRDx=PLLSYSCLK/2<br>0x02，QUALPRDx=PLLSYSCLK/4<br>...<br>0xFF，QUALPRDx=PLLSYSCLK/510<br>重置类型：SYSRSn |
| 15-8 | QUALPRD1 | R | 0h | GPIO8至GPIO15的合格采样周期：<br>0x00，QUALPRDx=PLLSYSCLK<br>0x01，QUALPRDx=PLLSYSCLK/2<br>0x02，QUALPRDx=PLLSYSCLK/4<br>...<br>0xFF，QUALPRDx=PLLSYSCLK/510<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 7-0 | QUALPRD0 | R | 0h | GPIO0至GPIO7的合格采样周期：<br>0x00，QUALPRDx=PLLSYSCLK<br>0x01，QUALPRDx=PLLSYSCLK/2<br>0x02，QUALPRDx=PLLSYSCLK/4<br>...<br>0xFF，QUALPRDx=PLLSYSCLK/510<br>重置类型：SYSRSn |

### 7.9.2.2 GPAQSEL1 寄存器

偏移地址：4h　复位值：00000000h

表 7-16 GPAQSEL1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO15 | | GPIO14 | | GPIO13 | | GPIO12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO11 | | GPIO10 | | GPIO9 | | GPIO8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO7 | | GPIO6 | | GPIO5 | | GPIO4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO3 | | GPIO2 | | GPIO1 | | GPIO0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-17 GPAQSEL1

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO15 | R/W | 0h | 为GPIO15选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 29-28 | GPIO14 | R/W | 0h | 为GPIO14选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 27-26 | GPIO13 | R/W | 0h | 为GPIO13选择输入限定类型：<br>0，0，同步 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| | | | | 0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 25-24 | GPIO12 | R/W | 0h | 为GPIO12选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 23-22 | GPIO11 | R/W | 0h | 为GPIO11选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 21-20 | GPIO10 | R/W | 0h | 为GPIO15选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 19-18 | GPIO9 | R/W | 0h | 为GPIO9选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 17-16 | GPIO8 | R/W | 0h | 为GPIO8选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br><br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 15-14 | GPIO7 | R/W | 0h | 为GPIO7选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 13-12 | GPIO6 | R/W | 0h | 为GPIO6选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样) |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| | | | | 1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 11-10 | GPIO5 | R/W | 0h | 为GPIO5选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 9-8 | GPIO4 | R/W | 0h | 为GPIO4选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 7-6 | GPIO3 | R/W | 0h | 为GPIO3选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 5-4 | GPIO2 | R/W | 0h | 为GPIO2选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 3-2 | GPIO1 | R/W | 0h | 为GPIO1选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 1-0 | GPIO0 | R/W | 0h | 为GPIO0选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |

### 7.9.2.3 GPAQSEL2 寄存器

偏移地址：8h　复位值：00000000h

表 7-18 GPAQSEL2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | | GPIO30 | | GPIO29 | | GPIO28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO27 | | GPIO26 | | GPIO25 | | GPIO24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO23 | | GPIO22 | | GPIO21 | | GPIO20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO19 | | GPIO18 | | GPIO17 | | GPIO16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-19 GPAQSEL2

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO31 | R/W | 0h | 为GPIO31选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 29-28 | GPIO30 | R/W | 0h | 为GPIO30选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 27-26 | GPIO29 | R/W | 0h | 为GPIO29选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 25-24 | GPIO28 | R/W | 0h | 为GPIO28选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 23-22 | GPIO27 | R/W | 0h | 为GPIO27选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样) |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| | | | | 1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 21-20 | GPIO26 | R/W | 0h | 为GPIO26选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 19-18 | GPIO25 | R/W | 0h | 为GPIO25选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 17-16 | GPIO24 | R/W | 0h | 为GPIO24选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 15-14 | GPIO23 | R/W | 0h | 为GPIO23选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 13-12 | GPIO22 | R/W | 0h | 为GPIO22选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 11-10 | GPIO21 | R/W | 0h | 为GPIO21选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 9-8 | GPIO20 | R/W | 0h | 为GPIO20选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 7-6 | GPIO19 | R/W | 0h | 为GPIO19选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 5-4 | GPIO18 | R/W | 0h | 为GPIO2选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 3-2 | GPIO17 | R/W | 0h | 为GPIO17选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 1-0 | GPIO16 | R/W | 0h | 为GPIO16选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |

### 7.9.2.4 GPAMUX1 寄存器

偏移地址：Ch　复位值：00000000h

表 7-20 GPAMUX1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO15 | | GPIO14 | | GPIO13 | | GPIO12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO11 | | GPIO10 | | GPIO9 | | GPIO8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO7 | | GPIO6 | | GPIO5 | | GPIO4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO3 | | GPIO2 | | GPIO1 | | GPIO0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-21 GPAMUX1

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO15 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 29-28 | GPIO14 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 27-26 | GPIO13 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 25-24 | GPIO12 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 23-22 | GPIO11 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 21-20 | GPIO10 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 19-18 | GPIO9 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn重置类型：SYSRSn |
| 17-16 | GPIO8 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 15-14 | GPIO7 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 13-12 | GPIO6 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 11-10 | GPIO5 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 9-8 | GPIO4 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 7-6 | GPIO3 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 5-4 | GPIO2 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 3-2 | GPIO1 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 1-0 | GPIO0 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

### 7.9.2.5 GPAMUX2 寄存器

偏移地址：10h　复位值：00000000h

表 7-22 GPAMUX2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | | GPIO30 | | GPIO29 | | GPIO28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO27 | | GPIO26 | | GPIO25 | | GPIO24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO23 | | GPIO22 | | GPIO21 | | GPIO20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO19 | | GPIO18 | | GPIO17 | | GPIO16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-23 GPAMUX2 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO31 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 29-28 | GPIO30 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 27-26 | GPIO29 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 25-24 | GPIO28 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 23-22 | GPIO27 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 21-20 | GPIO26 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 19-18 | GPIO25 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn重置类型：SYSRSn |
| 17-16 | GPIO24 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 15-14 | GPIO23 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 13-12 | GPIO22 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 11-10 | GPIO21 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 9-8 | GPIO20 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 7-6 | GPIO19 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 5-4 | GPIO18 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 3-2 | GPIO17 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 1-0 | GPIO16 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

### 7.9.2.6 GPADIR 寄存器

偏移地址：14h　复位值：00000000h

表 7-24 GPADIR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-25 GPADIR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO31 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 30 | GPIO30 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 29 | GPIO29 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 28 | GPIO28 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 27 | GPIO27 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 26 | GPIO26 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 25 | GPIO25 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 24 | GPIO24 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 23 | GPIO23 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 22 | GPIO22 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 21 | GPIO21 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 20 | GPIO20 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 19 | GPIO19 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 18 | GPIO18 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 17 | GPIO17 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 16 | GPIO16 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 15 | GPIO15 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 14 | GPIO14 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 13 | GPIO13 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 12 | GPIO12 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 11 | GPIO11 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 10 | GPIO10 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 9 | GPIO9 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 8 | GPIO8 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 7 | GPIO7 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 6 | GPIO6 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 5 | GPIO5 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 4 | GPIO4 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 3 | GPIO3 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 2 | GPIO2 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 1 | GPIO1 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 0 | GPIO0 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

### 7.9.2.7 GPAPUD 寄存器

偏移地址：18h　复位值：FFFFFFFFh

表 7-26 GPAPUD 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-27 GPAPUD 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO31 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 30 | GPIO30 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 29 | GPIO29 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 28 | GPIO28 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 27 | GPIO27 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 26 | GPIO26 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 25 | GPIO25 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 24 | GPIO24 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 23 | GPIO23 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 22 | GPIO22 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 21 | GPIO21 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 20 | GPIO20 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 19 | GPIO19 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 18 | GPIO18 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 17 | GPIO17 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 16 | GPIO16 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 15 | GPIO15 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 14 | GPIO14 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 13 | GPIO13 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 12 | GPIO12 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 11 | GPIO11 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 10 | GPIO10 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 9 | GPIO9 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 8 | GPIO8 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 7 | GPIO7 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 6 | GPIO6 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 5 | GPIO5 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 4 | GPIO4 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 3 | GPIO3 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 2 | GPIO2 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 1 | GPIO1 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 0 | GPIO0 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |

### 7.9.2.8 GPAINV 寄存器

偏移地址：20h　复位值：00000000h

表 7-28 GPAINV 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-29 GPAINV 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO31 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 30 | GPIO30 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 29 | GPIO29 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 28 | GPIO28 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 27 | GPIO27 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 26 | GPIO26 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 25 | GPIO25 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 24 | GPIO24 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 23 | GPIO23 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 22 | GPIO22 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 21 | GPIO21 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 20 | GPIO20 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 19 | GPIO19 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 18 | GPIO18 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 17 | GPIO17 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 16 | GPIO16 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 15 | GPIO15 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 14 | GPIO14 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 13 | GPIO13 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 12 | GPIO12 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 11 | GPIO11 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 10 | GPIO10 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 9 | GPIO9 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 8 | GPIO8 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 7 | GPIO7 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 6 | GPIO6 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 5 | GPIO5 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 4 | GPIO4 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 3 | GPIO3 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 2 | GPIO2 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 1 | GPIO1 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |
| 0 | GPIO0 | R/W | 0h | 输入反转控制<br>重置类型：SYSRSn |

### 7.9.2.9 GPAODR 寄存器

偏移地址：24h　复位值：00000000h

表 7-30 GPAODR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-31 GPAODR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO31 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 30 | GPIO30 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 29 | GPIO29 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 28 | GPIO28 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 27 | GPIO27 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 26 | GPIO26 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 25 | GPIO25 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 24 | GPIO24 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 23 | GPIO23 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 22 | GPIO22 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 21 | GPIO21 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 20 | GPIO20 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 19 | GPIO19 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 18 | GPIO18 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 17 | GPIO17 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 16 | GPIO16 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 15 | GPIO15 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 14 | GPIO14 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 13 | GPIO13 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 12 | GPIO12 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 11 | GPIO11 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 10 | GPIO10 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 9 | GPIO9 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 8 | GPIO8 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 7 | GPIO7 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 6 | GPIO6 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 5 | GPIO5 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 4 | GPIO4 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 3 | GPIO3 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 2 | GPIO2 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 1 | GPIO1 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 0 | GPIO0 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |

### 7.9.2.10 GPAAMSEL 寄存器

偏移地址：28h　复位值：FFFFFFFFh

表 7-32 GPAAMSEL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | GPIO28 | 保留 | | | |
| R/W-1h | | | R/W-0h | R/W-1h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | GPIO21 | GPIO20 | 保留 | | | |
| R/W-1h | | R/W-0h | R/W-0h | R/W-1h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | GPIO21 | GPIO20 | 保留 | | | |
| R/W-1h | | R/W-0h | R/W-0h | R/W-1h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R/W-1h | | | | | | | |

表 7-33 GPAAMSEL 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-29 | 保留位 | R/W | 1h | 保留 |
| 28 | GPIO28 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 27-22 | 保留位 | R/W | 1h | 保留 |
| 21 | GPIO21 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 20 | GPIO20 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 19-14 | 保留位 | R/W | 1h | 保留 |
| 13 | GPIO21 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 12 | GPIO20 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 11-0 | 保留位 | R/W | 1h | 保留 |

### 7.9.2.11 GPAGMUX1 寄存器

偏移地址：40h　复位值：00000000h

表 7-34 GPAGMUX1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO15 | | GPIO14 | | GPIO13 | | GPIO12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO11 | | GPIO10 | | GPIO9 | | GPIO8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO7 | | GPIO6 | | GPIO5 | | GPIO4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO3 | | GPIO2 | | GPIO1 | | GPIO0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-35 GPAGMUX1 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO15 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 29-28 | GPIO14 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 27-26 | GPIO13 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 25-24 | GPIO12 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 23-22 | GPIO11 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 21-20 | GPIO10 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 19-18 | GPIO9 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn重置类型：SYSRSn |
| 17-16 | GPIO8 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 15-14 | GPIO7 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 13-12 | GPIO6 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 11-10 | GPIO5 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 9-8 | GPIO4 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 7-6 | GPIO3 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 5-4 | GPIO2 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 3-2 | GPIO1 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 1-0 | GPIO0 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

### 7.9.2.12 GPAGMUX2 寄存器

偏移地址：44h　复位值：00000000h

表 7-36 GPAGMUX2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | | GPIO30 | | GPIO29 | | GPIO28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO27 | | GPIO26 | | GPIO25 | | GPIO24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO23 | | GPIO22 | | GPIO21 | | GPIO20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO19 | | GPIO18 | | GPIO17 | | GPIO16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-37 GPAGMUX2 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO31 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 29-28 | GPIO30 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 27-26 | GPIO29 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 25-24 | GPIO28 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 23-22 | GPIO27 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 21-20 | GPIO26 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 19-18 | GPIO25 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn重置类型：SYSRSn |
| 17-16 | GPIO24 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 15-14 | GPIO23 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 13-12 | GPIO22 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 11-10 | GPIO21 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 9-8 | GPIO20 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 7-6 | GPIO19 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 5-4 | GPIO18 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 3-2 | GPIO17 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 1-0 | GPIO16 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

### 7.9.2.13 GPALOCK 寄存器

偏移地址：78h　复位值：00000000h

表 7-38 GPALOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-39 GPALOCK 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO31 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 30 | GPIO30 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 29 | GPIO29 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 28 | GPIO28 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 27 | GPIO27 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 26 | GPIO26 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 25 | GPIO25 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 24 | GPIO24 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 23 | GPIO23 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 22 | GPIO22 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 21 | GPIO21 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 20 | GPIO20 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 19 | GPIO19 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 18 | GPIO18 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 17 | GPIO17 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 16 | GPIO16 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 15 | GPIO15 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 14 | GPIO14 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 13 | GPIO13 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 12 | GPIO12 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 11 | GPIO11 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 10 | GPIO10 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 9 | GPIO9 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 8 | GPIO8 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 7 | GPIO7 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 6 | GPIO6 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 5 | GPIO5 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 4 | GPIO4 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 3 | GPIO3 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 2 | GPIO2 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 1 | GPIO1 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 0 | GPIO0 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |

### 7.9.2.14 GPACR 寄存器

偏移地址：7Ch　复位值：00000000h

表 7-40 GPACR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |

表 7-41 GPACR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO31 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 30 | GPIO30 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 29 | GPIO29 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 28 | GPIO28 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 27 | GPIO27 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 26 | GPIO26 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 25 | GPIO25 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 24 | GPIO24 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 23 | GPIO23 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 22 | GPIO22 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 21 | GPIO21 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 20 | GPIO20 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 19 | GPIO19 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 18 | GPIO18 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 17 | GPIO17 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 16 | GPIO16 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 15 | GPIO15 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 14 | GPIO14 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 13 | GPIO13 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 12 | GPIO12 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 11 | GPIO11 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 10 | GPIO10 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 9 | GPIO9 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 8 | GPIO8 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 7 | GPIO7 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 6 | GPIO6 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 5 | GPIO5 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 4 | GPIO4 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 3 | GPIO3 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 2 | GPIO2 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 1 | GPIO1 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 0 | GPIO0 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |

### 7.9.2.15 GPBCTRL 寄存器

偏移地址：80h　复位值：00000000h

表 7-42 GPBCTRL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| QUALPRD3 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QUALPRD2 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QUALPRD1 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QUALPRD0 | | | | | | | |
| R-0h | | | | | | | |

表 7-43 GPBCTRL 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-24 | QUALPRD3 | R/W | 0h | GPIO56至GPIO63的合格采样周期：<br>0x00，QUALPRDx=PLLSYSCLK<br>0x01，QUALPRDx=PLLSYSCLK/2<br>0x02，QUALPRDx=PLLSYSCLK/4<br>...<br>0xFF，QUALPRDx=PLLSYSCLK/510<br>重置类型：SYSRSn |
| 23-16 | QUALPRD2 | R | 0h | GPIO48至GPIO55的合格采样周期：<br>0x00，QUALPRDx=PLLSYSCLK<br>0x01，QUALPRDx=PLLSYSCLK/2<br>0x02，QUALPRDx=PLLSYSCLK/4<br>...<br>0xFF，QUALPRDx=PLLSYSCLK/510<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 15-8 | QUALPRD1 | R | 0h | GPIO40至GPIO47的合格采样周期：<br>0x00，QUALPRDx=PLLSYSCLK<br>0x01，QUALPRDx=PLLSYSCLK/2<br>0x02，QUALPRDx=PLLSYSCLK/4<br>...<br>0xFF，QUALPRDx=PLLSYSCLK/510<br>重置类型：SYSRSn |
| 7-0 | QUALPRD0 | R | 0h | GPIO32至GPIO39的合格采样周期：<br>0x00，QUALPRDx=PLLSYSCLK<br>0x01，QUALPRDx=PLLSYSCLK/2<br>0x02，QUALPRDx=PLLSYSCLK/4<br>...<br>0xFF，QUALPRDx=PLLSYSCLK/510<br>重置类型：SYSRSn |

### 7.9.2.16 GPBQSEL1 寄存器

偏移地址：84h　复位值：00000CC0h

GPIO35：TDI（仿真器） --> 默认为异步输入

GPIO37：TDO（仿真器）--> 默认为异步输入

表 7-44 GPBQSEL1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | GPIO46 | | GPIO45 | | GPIO44 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO43 | | GPIO42 | | GPIO41 | | GPIO40 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO39 | | 保留 | | GPIO37 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO35 | | GPIO34 | | GPIO33 | | GPIO32 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-45 GPBQSEL1 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | 保留位 | R/W | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 29-28 | GPIO46 | R/W | 0h | 为GPIO46选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 27-26 | GPIO45 | R/W | 0h | 为GPIO45选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 25-24 | GPIO44 | R/W | 0h | 为GPIO44选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 23-22 | GPIO43 | R/W | 0h | 为GPIO43选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 21-20 | GPIO42 | R/W | 0h | 为GPIO42选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 19-18 | GPIO41 | R/W | 0h | 为GPIO41选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 17-16 | GPIO40 | R/W | 0h | 为GPIO40选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 15-14 | GPIO39 | R/W | 0h | 为GPIO39选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样) |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| | | | | 1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 13-12 | 保留位 | R/W | 0h | 保留 |
| 11-10 | GPIO37 | R/W | 0h | 为GPIO37选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 9-8 | 保留位 | R/W | 0h | 保留 |
| 7-6 | GPIO35 | R/W | 0h | 为GPIO35选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 5-4 | GPIO34 | R/W | 0h | 为GPIO34选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 3-2 | GPIO33 | R/W | 0h | 为GPIO33选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 1-0 | GPIO32 | R/W | 0h | 为GPIO32选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |

### 7.9.2.17 GPBQSEL2 寄存器

偏移地址：88h　复位值：00000000h

表 7-46 GPBQSEL2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | | GPIO62 | | GPIO61 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |

表 7-47 GPBQSEL2 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO63 | R/W | 0h | 为GPIO63选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 29-28 | GPIO62 | R/W | 0h | 为GPIO62选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 27-26 | GPIO61 | R/W | 0h | 为GPIO61选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 25-0 | 保留位 | R/W | 0h | 保留 |

### 7.9.2.18 GPBMUX1 寄存器

偏移地址：8Ch　复位值：00000CC0h

表 7-48 GPBMUX1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | GPIO46 | | GPIO45 | | GPIO44 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO43 | | GPIO42 | | GPIO41 | | GPIO40 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO39 | | 保留 | | GPIO37 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO35 | | GPIO34 | | GPIO33 | | GPIO32 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-49 GPBMUX1 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | 保留位 | R/W | 0h | 保留 |
| 29-28 | GPIO46 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 27-26 | GPIO45 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 25-24 | GPIO44 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 23-22 | GPIO43 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 21-20 | GPIO42 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 19-18 | GPIO41 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 17-16 | GPIO40 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 15-14 | GPIO39 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 13-12 | 保留位 | R/W | 0h | 保留 |
| 11-10 | GPIO37 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 9-8 | 保留位 | R/W | 0h | 保留 |
| 7-6 | GPIO35 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 5-4 | GPIO34 | R/W | 0h | GPIO引脚选择 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| | | | | 重置类型：SYSRSn |
| 3-2 | GPIO33 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 1-0 | GPIO32 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

### 7.9.2.19 GPBMUX2 寄存器

偏移地址：90h　复位值：00000000h

表 7-50 GPBMUX2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | | GPIO62 | | GPIO61 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |

表 7-51 GPBMUX2 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO63 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 29-28 | GPIO62 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 27-26 | GPIO61 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 25-0 | 保留位 | R/W | 0h | 保留 |

### 7.9.2.20 GPBDIR 寄存器

偏移地址：94h　复位值：00000000h

表 7-52 GPBDIR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | GPIO62 | GPIO61 | 保留 | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO39 | 保留 | GPIO37 | 保留 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-53 GPBDIR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO63 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 30 | GPIO62 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 29 | GPIO61 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 28-15 | 保留位 | R/W | 0h | 保留 |
| 14 | GPIO46 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 13 | GPIO45 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 12 | GPIO44 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 11 | GPIO43 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 10 | GPIO42 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 9 | GPIO41 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 8 | GPIO40 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 7 | GPIO39 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 6 | 保留位 | R/W | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 5 | GPIO37 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 4 | 保留位 | R/W | 0h | 保留 |
| 3 | GPIO35 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 2 | GPIO34 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 1 | GPIO33 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 0 | GPIO32 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |

### 7.9.2.21 GPBPUD 寄存器

偏移地址：98h　复位值：FFFFFFFFh

表 7-54 GPBPUD 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | GPIO62 | GPIO61 | 保留 | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO39 | 保留 | GPIO37 | 保留 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-55 GPBPUD 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO63 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 30 | GPIO62 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 29 | GPIO61 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 28-15 | 保留位 | R/W | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 14 | GPIO46 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 13 | GPIO45 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 12 | GPIO44 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 11 | GPIO43 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 10 | GPIO42 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 9 | GPIO41 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 8 | GPIO40 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 7 | GPIO39 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 6 | 保留位 | R/W | 0h | 保留 |
| 5 | GPIO37 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 4 | 保留位 | R/W | 0h | 保留 |
| 3 | GPIO35 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 2 | GPIO34 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 1 | GPIO33 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 0 | GPIO32 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |

### 7.9.2.22 GPBINV 寄存器

偏移地址：A0h　复位值：00000000h

表 7-56 GPBINV 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | GPIO62 | GPIO61 | 保留 | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO39 | 保留 | GPIO37 | 保留 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-57 GPBINV 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO63 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 30 | GPIO62 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 29 | GPIO61 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 28-15 | 保留位 | R/W | 0h | 保留 |
| 14 | GPIO46 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 13 | GPIO45 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 12 | GPIO44 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 11 | GPIO43 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 10 | GPIO42 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 9 | GPIO41 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 8 | GPIO40 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 7 | GPIO39 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 6 | 保留位 | R/W | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 5 | GPIO37 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 4 | 保留位 | R/W | 0h | 保留 |
| 3 | GPIO35 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 2 | GPIO34 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 1 | GPIO33 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 0 | GPIO32 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |

### 7.9.2.23 GPBODR 寄存器

偏移地址：A4h　复位值：00000000h

表 7-58 GPBODR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | GPIO62 | GPIO61 | 保留 | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO39 | 保留 | GPIO37 | 保留 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-59 GPBODR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO63 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 30 | GPIO62 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 29 | GPIO61 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 28-15 | 保留位 | R/W | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 14 | GPIO46 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 13 | GPIO45 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 12 | GPIO44 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 11 | GPIO43 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 10 | GPIO42 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 9 | GPIO41 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 8 | GPIO40 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 7 | GPIO39 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 6 | 保留位 | R/W | 0h | 保留 |
| 5 | GPIO37 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 4 | 保留位 | R/W | 0h | 保留 |
| 3 | GPIO35 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 2 | GPIO34 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 1 | GPIO33 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 0 | GPIO32 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |

### 7.9.2.24 GPBGMUX1 寄存器

偏移地址：C0h　复位值：0000CC0h

表 7-60 GPBGMUX1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | GPIO46 | | GPIO45 | | GPIO44 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO43 | | GPIO42 | | GPIO41 | | GPIO40 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO39 | | 保留 | | GPIO37 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO35 | | GPIO34 | | GPIO33 | | GPIO32 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-61 GPBGMUX1 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | 保留位 | R/W | 0h | 保留 |
| 29-28 | GPIO46 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 27-26 | GPIO45 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 25-24 | GPIO44 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 23-22 | GPIO43 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 21-20 | GPIO42 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 19-18 | GPIO41 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 17-16 | GPIO40 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 15-14 | GPIO39 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 13-12 | 保留位 | R/W | 0h | 保留 |
| 11-10 | GPIO37 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 9-8 | 保留位 | R/W | 0h | 保留 |
| 7-6 | GPIO35 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 5-4 | GPIO34 | R/W | 0h | GPIO引脚选择 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| | | | | 重置类型：SYSRSn |
| 3-2 | GPIO33 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 1-0 | GPIO32 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

### 7.9.2.25 GPBGMUX2 寄存器

偏移地址：C4h　复位值：00000000h

表 7-62 GPBGMUX2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | | GPIO62 | | GPIO61 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |

表 7-63 GPBGMUX2 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO63 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 29-28 | GPIO62 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 27-26 | GPIO61 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 25-0 | 保留位 | R/W | 0h | 保留 |

### 7.9.2.26 GPBLOCK 寄存器

偏移地址：F8h　复位值：00000000h

表 7-64 GPBLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | GPIO62 | GPIO61 | 保留 | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO39 | 保留 | GPIO37 | 保留 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-65 GPBLOCK 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO63 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 30 | GPIO62 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 29 | GPIO61 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 28-15 | 保留位 | R/W | 0h | 保留 |
| 14 | GPIO46 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 13 | GPIO45 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 12 | GPIO44 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 11 | GPIO43 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 10 | GPIO42 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 9 | GPIO41 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 8 | GPIO40 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 7 | GPIO39 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 6 | 保留位 | R/W | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 5 | GPIO37 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 4 | 保留位 | R/W | 0h | 保留 |
| 3 | GPIO35 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 2 | GPIO34 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 1 | GPIO33 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 0 | GPIO32 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |

### 7.9.2.27 GPBCR 寄存器

偏移地址：FCh　复位值：00000000h

表 7-66 GPBCR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | GPIO62 | GPIO61 | 保留 | | | | |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/WSonce-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO39 | 保留 | GPIO37 | 保留 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |

表 7-67 GPBCR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO63 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 30 | GPIO62 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 29 | GPIO61 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 28-15 | 保留位 | R/WSonce | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 14 | GPIO46 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 13 | GPIO45 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 12 | GPIO44 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 11 | GPIO43 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 10 | GPIO42 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 9 | GPIO41 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 8 | GPIO40 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 7 | GPIO39 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 6 | 保留位 | R/WSonce | 0h | 保留 |
| 5 | GPIO37 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 4 | 保留位 | R/WSonce | 0h | 保留 |
| 3 | GPIO35 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 2 | GPIO34 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 1 | GPIO33 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 0 | GPIO32 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |

### 7.9.2.28 GPHCTRL 寄存器

偏移地址：380h　复位值：00000000h

表 7-68 GPHCTRL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QUALPRD2 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QUALPRD1 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QUALPRD0 | | | | | | | |
| R-0h | | | | | | | |

表 7-69 GPHCTRL 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-24 | 保留位 | R/W | 0h | 保留 |
| 23-16 | QUALPRD2 | R | 0h | GPIO48至GPIO55的合格采样周期：<br>0x00，QUALPRDx=PLLSYSCLK<br>0x01，QUALPRDx=PLLSYSCLK/2<br>0x02，QUALPRDx=PLLSYSCLK/4<br>...<br>0xFF，QUALPRDx=PLLSYSCLK/510<br>重置类型：SYSRSn |
| 15-8 | QUALPRD1 | R | 0h | GPIO40至GPIO47的合格采样周期：<br>0x00，QUALPRDx=PLLSYSCLK<br>0x01，QUALPRDx=PLLSYSCLK/2<br>0x02，QUALPRDx=PLLSYSCLK/4<br>...<br>0xFF，QUALPRDx=PLLSYSCLK/510<br>重置类型：SYSRSn |
| 7-0 | QUALPRD0 | R | 0h | GPIO32至GPIO39的合格采样周期：<br>0x00，QUALPRDx=PLLSYSCLK<br>0x01，QUALPRDx=PLLSYSCLK/2<br>0x02，QUALPRDx=PLLSYSCLK/4<br>...<br>0xFF，QUALPRDx=PLLSYSCLK/510<br>重置类型：SYSRSn |

### 7.9.2.29 GPHQSEL1 寄存器

偏移地址：384h　复位值：00000000h

表 7-70 GPHQSEL1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO239 | | GPIO238 | | GPIO237 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | GPIO233 | | GPIO232 | |
| R/W-0h | | | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO231 | | GPIO230 | | 保留 | | GPIO228 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO227 | | GPIO226 | | GPIO225 | | GPIO224 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-71 GPHQSEL1 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO239 | R/W | 0h | 为GPIO239选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 29-28 | GPIO238 | R/W | 0h | 为GPIO238选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 27-26 | GPIO237 | R/W | 0h | 为GPIO237选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 25-20 | 保留位 | R/W | 0h | 保留 |
| 19-18 | GPIO233 | R/W | 0h | 为GPIO233选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 17-16 | GPIO232 | R/W | 0h | 为GPIO232选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 15-14 | GPIO231 | R/W | 0h | 为GPIO231选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 13-12 | GPIO230 | R/W | 0h | 为GPIO230选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 11-10 | 保留位 | R/W | 0h | 保留 |
| 9-8 | GPIO228 | R/W | 0h | 为GPIO228选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 7-6 | GPIO227 | R/W | 0h | 为GPIO227选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 5-4 | GPIO226 | R/W | 0h | 为GPIO226选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 3-2 | GPIO225 | R/W | 0h | 为GPIO225选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 1-0 | GPIO224 | R/W | 0h | 为GPIO224选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |

## 7.9.2.30 GPHQSEL2 寄存器

偏移地址：388h　复位值：00000000h

表 7-72 GPHQSEL2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | GPIO245 | | GPIO244 | |
| R/W-0h | | | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | GPIO242 | | GPIO241 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-73 GPHQSEL2 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-12 | 保留位 | R/W | 0h | 保留 |
| 11-10 | GPIO245 | R/W | 0h | 为GPIO245选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 9-8 | GPIO244 | R/W | 0h | 为GPIO244选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 7-6 | 保留位 | R/W | 0h | 保留 |
| 5-4 | GPIO242 | R/W | 0h | 为GPIO242选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 3-2 | GPIO241 | R/W | 0h | 为GPIO241选择输入限定类型：<br>0，0，同步<br>0，1，鉴定(3采样)<br>1，0，鉴定(6采样)<br>1，1，异步(无同步或限定)<br>重置类型：SYSRSn |
| 1-0 | 保留位 | R/W | 0h | 保留 |

## 7.9.2.31 GPHMUX1 寄存器

偏移地址：38Ch　复位值：00000000h

表 7-74 GPHMUX1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO239 | | GPIO238 | | GPIO237 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | GPIO233 | | GPIO232 | |
| R/W-0h | | | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO231 | | GPIO230 | | 保留 | | GPIO228 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO227 | | GPIO226 | | GPIO225 | | GPIO224 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-75 GPHMUX1 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO239 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 29-28 | GPIO238 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 27-26 | GPIO237 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 25-20 | 保留位 | R/W | 0h | 保留 |
| 19-18 | GPIO233 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 17-16 | GPIO232 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 15-14 | GPIO231 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 13-12 | GPIO230 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 11-10 | 保留位 | R/W | 0h | 保留 |
| 9-8 | GPIO228 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 7-6 | GPIO227 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 5-4 | GPIO226 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 3-2 | GPIO225 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 1-0 | GPIO224 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

### 7.9.2.32 GPHMUX2 寄存器

偏移地址：390h　复位值：00000000h

表 7-76 GPHMUX2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | GPIO245 | | GPIO244 | |
| R/W-0h | | | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | GPIO242 | | GPIO241 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-77 GPHMUX2 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-12 | 保留位 | R/W | 0h | 保留 |
| 11-10 | GPIO245 | R/W | 0h | 保留 |
| 9-8 | GPIO244 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 7-6 | 保留位 | R/W | 0h | 保留 |
| 5-4 | GPIO242 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 3-2 | GPIO241 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 1-0 | 保留位 | R/W | 0h | 保留 |

## 7.9.2.33 GPHDIR 寄存器

偏移地址：394h　复位值：00000000h

表 7-78 GPHDIR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | GPIO242 | 保留 | |
| R/W-0h | | | | | R/W-0h | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | 保留 | GPIO224 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-79 GPHDIR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-19 | 保留位 | R/W | 0h | 保留 |
| 18 | GPIO242 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 17-7 | 保留位 | R/W | 0h | 保留 |
| 6 | GPIO230 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 5 | 保留位 | R/W | 0h | 保留 |
| 4 | GPIO228 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 3 | GPIO227 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 2 | GPIO226 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |
| 1 | 保留位 | R/W | 0h | 保留 |
| 0 | GPIO224 | R/W | 0h | GPIO引脚方向<br>重置类型：SYSRSn |

### 7.9.2.34 GPHPUD 寄存器

偏移地址：398h　复位值：FFFFFFFFh

表 7-80 GPHPUD 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | GPIO245 | GPIO244 | 保留 | GPIO242 | GPIO241 | 保留 |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO239 | GPIO238 | GPIO237 | 保留 | | | GPIO233 | GPIO232 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO231 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | GPIO225 | GPIO224 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-81 GPHPUD 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-22 | 保留位 | R/W | 0h | 保留 |
| 21 | GPIO245 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 20 | GPIO244 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 19 | 保留位 | R/W | 0h | 保留 |
| 18 | GPIO242 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 17 | GPIO241 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 16 | 保留位 | R/W | 0h | 保留 |
| 15 | GPIO239 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 14 | GPIO238 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 13 | GPIO237 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 12-10 | 保留位 | R/W | 0h | 保留 |
| 9 | GPIO233 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 8 | GPIO232 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 7 | GPIO231 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 6 | GPIO230 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 5 | 保留位 | R/W | 0h | 保留 |
| 4 | GPIO228 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 3 | GPIO227 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 2 | GPIO226 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 1 | GPIO225 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |
| 0 | GPIO224 | R/W | 0h | 上拉禁用控制<br>重置类型：SYSRSn |

### 7.9.2.35 GPHINV 寄存器

偏移地址：3A0h　复位值：00000000h

表 7-82 GPHINV 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | GPIO245 | GPIO244 | 保留 | GPIO242 | GPIO241 | 保留 |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO239 | GPIO238 | GPIO237 | 保留 | | | GPIO233 | GPIO232 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO231 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | GPIO225 | GPIO224 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-83 GPHINV 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-22 | 保留位 | R/W | 0h | 保留 |
| 21 | GPIO245 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 20 | GPIO244 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 19 | 保留位 | R/W | 0h | 保留 |
| 18 | GPIO242 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 17 | GPIO241 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 16 | 保留位 | R/W | 0h | 保留 |
| 15 | GPIO239 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 14 | GPIO238 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 13 | GPIO237 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 12-10 | 保留位 | R/W | 0h | 保留 |
| 9 | GPIO233 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 8 | GPIO232 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 7 | GPIO231 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 6 | GPIO230 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 5 | 保留位 | R/W | 0h | 保留 |
| 4 | GPIO228 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 3 | GPIO227 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 2 | GPIO226 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 1 | GPIO225 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |
| 0 | GPIO224 | R/W | 0h | 输入反向控制<br>重置类型：SYSRSn |

### 7.9.2.36 GPHODR 寄存器

偏移地址：3A4h　复位值：00000000h

表 7-84 GPHODR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | GPIO242 | 保留 | |
| R/W-0h | | | | | R/W-0h | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | 保留 | GPIO224 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-85 GPHODR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-19 | 保留位 | R/W | 0h | 保留 |
| 18 | GPIO242 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 17-7 | 保留位 | R/W | 0h | 保留 |
| 6 | GPIO230 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 5 | 保留位 | R/W | 0h | 保留 |
| 4 | GPIO228 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 3 | GPIO227 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 2 | GPIO226 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |
| 1 | 保留位 | R/W | 0h | 保留 |
| 0 | GPIO224 | R/W | 0h | 开漏输出控制<br>重置类型：SYSRSn |

### 7.9.2.37 GPHAMSEL 寄存器

偏移地址：3A8h　复位值：FFFFFFFFh

表 7-86 GPHAMSEL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | GPIO245 | GPIO244 | 保留 | GPIO242 | GPIO241 | 保留 |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO239 | GPIO238 | GPIO237 | 保留 | | | GPIO233 | GPIO232 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO231 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | GPIO225 | GPIO224 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-87 GPHAMSEL 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-22 | 保留位 | R/W | 0h | 保留 |
| 21 | GPIO245 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 20 | GPIO244 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 19 | 保留位 | R/W | 0h | 保留 |
| 18 | GPIO242 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 17 | GPIO241 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 16 | 保留位 | R/W | 0h | 保留 |
| 15 | GPIO239 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 14 | GPIO238 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 13 | GPIO237 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 12-10 | 保留位 | R/W | 0h | 保留 |
| 9 | GPIO233 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 8 | GPIO232 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 7 | GPIO231 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 6 | GPIO230 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 5 | 保留位 | R/W | 0h | 保留 |
| 4 | GPIO228 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 3 | GPIO227 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 2 | GPIO226 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 1 | GPIO225 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |
| 0 | GPIO224 | R/W | 0h | 0：管脚的模拟功能被禁用，并且管脚能够实现其他GPIO配置寄存器指定的数字功能<br>1：使能管脚的模拟功能。读取本寄存器可获得当前配置值。<br>重置类型：SYSRSn |

## 7.9.2.38 GPHGMUX1 寄存器

偏移地址：3C0h　复位值：00000000h

表 7-88 GPHGMUX1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO239 | | GPIO238 | | GPIO237 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | GPIO233 | | GPIO232 | |
| R/W-0h | | | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO231 | | GPIO230 | | 保留 | | GPIO228 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO227 | | GPIO226 | | GPIO225 | | GPIO224 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-89 GPHGMUX1 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-30 | GPIO239 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 29-28 | GPIO238 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 27-26 | GPIO237 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 25-20 | 保留位 | R/W | 0h | 保留 |
| 19-18 | GPIO233 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 17-16 | GPIO232 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 15-14 | GPIO231 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 13-12 | GPIO230 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 11-10 | 保留位 | R/W | 0h | 保留 |
| 9-8 | GPIO228 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 7-6 | GPIO227 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 5-4 | GPIO226 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 3-2 | GPIO225 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 1-0 | GPIO224 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |

### 7.9.2.39 GPHGMUX2 寄存器

偏移地址：3C4h　复位值：00000000h

表 7-90 GPHGMUX2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | GPIO245 | | GPIO244 | |
| R/W-0h | | | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | GPIO242 | | GPIO241 | | 保留 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 7-91 GPHGMUX2 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-12 | 保留位 | R/W | 0h | 保留 |
| 11-10 | GPIO245 | R/W | 0h | 保留 |
| 9-8 | GPIO244 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 7-6 | 保留位 | R/W | 0h | 保留 |
| 5-4 | GPIO242 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 3-2 | GPIO241 | R/W | 0h | GPIO引脚选择<br>重置类型：SYSRSn |
| 1-0 | 保留位 | R/W | 0h | 保留 |

### 7.9.2.40 GPHLOCK 寄存器

偏移地址：3F8h　复位值：00000000h

表 7-92 GPHLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | GPIO245 | GPIO244 | 保留 | GPIO242 | GPIO241 | 保留 |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO239 | GPIO238 | GPIO237 | 保留 | | | GPIO233 | GPIO232 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO231 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | GPIO225 | GPIO224 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-93 GPHLOCK 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-22 | 保留位 | R/W | 0h | 保留 |
| 21 | GPIO245 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 20 | GPIO244 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 19 | 保留位 | R/W | 0h | 保留 |
| 18 | GPIO242 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 17 | GPIO241 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 16 | 保留位 | R/W | 0h | 保留 |
| 15 | GPIO239 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 14 | GPIO238 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 13 | GPIO237 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 12-10 | 保留位 | R/W | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 9 | GPIO233 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 8 | GPIO232 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 7 | GPIO231 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 6 | GPIO230 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 5 | 保留位 | R/W | 0h | 保留 |
| 4 | GPIO228 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 3 | GPIO227 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 2 | GPIO226 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 1 | GPIO225 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |
| 0 | GPIO224 | R/W | 0h | 配置锁定<br>重置类型：SYSRSn |

### 7.9.2.41 GPHCR 寄存器

偏移地址：3FCh　复位值：00000000h

表 7-94 GPHCR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R / W S o n c e - 0 h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | GPIO245 | GPIO244 | 保留 | GPIO242 | GPIO241 | 保留 |
| R / W S o n c e - 0 h | | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO239 | GPIO238 | GPIO237 | 保留 | | | GPIO233 | GPIO232 |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R / W S o n c e - 0 h | | | R/WSonce-0h | R/WSonce-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO231 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | GPIO225 | GPIO224 |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |

表 7-95 GPHCR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-22 | 保留位 | R/WSonce | 0h | 保留 |
| 21 | GPIO245 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 20 | GPIO244 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 19 | 保留位 | R/WSonce | 0h | 保留 |
| 18 | GPIO242 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 17 | GPIO241 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 16 | 保留位 | R/WSonce | 0h | 保留 |
| 15 | GPIO239 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 14 | GPIO238 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 13 | GPIO237 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 12-10 | 保留位 | R/WSonce | 0h | 保留 |
| 9 | GPIO233 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 8 | GPIO232 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 7 | GPIO231 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 6 | GPIO230 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 5 | 保留位 | R/WSonce | 0h | 保留 |
| 4 | GPIO228 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 3 | GPIO227 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 2 | GPIO226 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 1 | GPIO225 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |
| 0 | GPIO224 | R/WSonce | 0h | 配置锁定提交<br>重置类型：SYSRSn |

### 7.9.3 GPIO 数据寄存器

表 7-96 GPIO 数据寄存器数据表

| 偏移量 | 寄存器 | 描述 |
|---|---|---|
| 0h | GPADAT | GPIO A数据寄存器（GPIO0~GPIO31） |
| 4h | GPASET | GPIO A数据置位寄存器（GPIO0~GPIO31） |
| 8h | GPACLEAR | GPIO A数据清除寄存器（GPIO0~GPIO31） |
| Ch | GPATOGGLE | GPIO A数据翻转寄存器（GPIO0~GPIO31） |
| 10h | GPBDAT | GPIO B数据寄存器（GPIO32~GPIO63） |
| 14h | GPBSET | GPIO B数据置位寄存器（GPIO32~GPIO63） |
| 18h | GPBCLEAR | GPIO B数据清除寄存器（GPIO32~GPIO63） |
| 1Ch | GPBTOGGLE | GPIO B数据翻转寄存器（GPIO32~GPIO63） |
| 70h | GPHDAT | GPIO H数据寄存器（GPIO224~GPIO255） |
| 74h | GPHSET | GPIO H数据置位寄存器（GPIO224~GPIO255） |
| 78h | GPHCLEAR | GPIO H数据清除寄存器（GPIO224~GPIO255） |
| 7Ch | GPHTOGGLE | GPIO H数据翻转寄存器（GPIO224~GPIO255） |

#### 7.9.3.1 GPADAT 寄存器

偏移地址：0h　复位值：00000000h

读此寄存器输出输入信号输入限定后的值

表 7-97 GPADAT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-98 GPADAT 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO31 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 30 | GPIO30 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 29 | GPIO29 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 28 | GPIO28 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 27 | GPIO27 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 26 | GPIO26 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 25 | GPIO25 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 24 | GPIO24 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 23 | GPIO23 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 22 | GPIO22 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 21 | GPIO21 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 20 | GPIO20 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 19 | GPIO19 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 18 | GPIO18 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 17 | GPIO17 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 16 | GPIO16 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 15 | GPIO15 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 14 | GPIO14 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 13 | GPIO13 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 12 | GPIO12 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 11 | GPIO11 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 10 | GPIO10 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 9 | GPIO9 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 8 | GPIO8 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 7 | GPIO7 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 6 | GPIO6 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 5 | GPIO5 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 4 | GPIO4 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 3 | GPIO3 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 2 | GPIO2 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 1 | GPIO1 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 0 | GPIO0 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |

### 7.9.3.2 GPASET 寄存器

偏移地址：4h　复位值：00000000h

表 7-99 GPASET 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-100 GPASET 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO31 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 30 | GPIO30 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 29 | GPIO29 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 28 | GPIO28 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 27 | GPIO27 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 26 | GPIO26 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 25 | GPIO25 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 24 | GPIO24 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 23 | GPIO23 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 22 | GPIO22 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 21 | GPIO21 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 20 | GPIO20 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 19 | GPIO19 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 18 | GPIO18 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 17 | GPIO17 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 16 | GPIO16 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 15 | GPIO15 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 14 | GPIO14 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 13 | GPIO13 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 12 | GPIO12 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 11 | GPIO11 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 10 | GPIO10 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 9 | GPIO9 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 8 | GPIO8 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 7 | GPIO7 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 6 | GPIO6 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 5 | GPIO5 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 4 | GPIO4 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 3 | GPIO3 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 2 | GPIO2 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 1 | GPIO1 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 0 | GPIO0 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |

### 7.9.3.3 GPACLEAR 寄存器

偏移地址：8h　复位值：00000000h

表 7-101 GPACLEAR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-102 GPACLEAR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO31 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 30 | GPIO30 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 29 | GPIO29 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 28 | GPIO28 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 27 | GPIO27 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 26 | GPIO26 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 25 | GPIO25 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 24 | GPIO24 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 23 | GPIO23 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 22 | GPIO22 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 21 | GPIO21 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 20 | GPIO20 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 19 | GPIO19 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 18 | GPIO18 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 17 | GPIO17 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 16 | GPIO16 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 15 | GPIO15 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 14 | GPIO14 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 13 | GPIO13 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 12 | GPIO12 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 11 | GPIO11 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 10 | GPIO10 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 9 | GPIO9 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 8 | GPIO8 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 7 | GPIO7 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 6 | GPIO6 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 5 | GPIO5 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 4 | GPIO4 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 3 | GPIO3 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 2 | GPIO2 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 1 | GPIO1 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 0 | GPIO0 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |

### 7.9.3.4 GPATOGGLE 寄存器

偏移地址：Ch　复位值：00000000h

表 7-103 GPATOGGLE 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-104 GPATOGGLE 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO31 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
| --- | --- | --- | --- | --- |
| 30 | GPIO30 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 29 | GPIO29 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 28 | GPIO28 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 27 | GPIO27 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 26 | GPIO26 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 25 | GPIO25 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 24 | GPIO24 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 23 | GPIO23 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 22 | GPIO22 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 21 | GPIO21 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 20 | GPIO20 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 19 | GPIO19 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 18 | GPIO18 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 17 | GPIO17 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 16 | GPIO16 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 15 | GPIO15 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 14 | GPIO14 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 13 | GPIO13 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 12 | GPIO12 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 11 | GPIO11 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 10 | GPIO10 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 9 | GPIO9 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 8 | GPIO8 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 7 | GPIO7 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 6 | GPIO6 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 5 | GPIO5 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 4 | GPIO4 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 3 | GPIO3 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 2 | GPIO2 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 1 | GPIO1 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 0 | GPIO0 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |

### 7.9.3.5 GPBDAT 寄存器

偏移地址：10h　复位值：00000000h

表 7-105 GPBDAT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | GPIO62 | GPIO61 | 保留 | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO39 | 保留 | GPIO37 | 保留 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-106 GPBDAT 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO63 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 30 | GPIO62 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 29 | GPIO61 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 28-15 | 保留位 | R/W | 0h | 保留 |
| 14 | GPIO46 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 13 | GPIO45 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 12 | GPIO44 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 11 | GPIO43 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 10 | GPIO42 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 9 | GPIO41 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 8 | GPIO40 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 7 | GPIO39 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 6 | 保留位 | R/W | 0h | 保留 |
| 5 | GPIO37 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 4 | 保留位 | R/W | 0h | 保留 |
| 3 | GPIO35 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 2 | GPIO34 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 1 | GPIO33 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 0 | GPIO32 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSnn |

### 7.9.3.6 GPBSET 寄存器

偏移地址：14h　复位值：00000000h

表 7-107 GPBSET 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | GPIO62 | GPIO61 | 保留 | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO39 | 保留 | GPIO37 | 保留 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-108 GPBSET 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO63 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 30 | GPIO62 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 29 | GPIO61 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 28-15 | 保留位 | R/W | 0h | 保留 |
| 14 | GPIO46 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 13 | GPIO45 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 12 | GPIO44 | R/W | 0h | 输出置位<br>重置类型：SYSRSnn |
| 11 | GPIO43 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 10 | GPIO42 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 9 | GPIO41 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 8 | GPIO40 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 7 | GPIO39 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 6 | 保留位 | R/W | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 5 | GPIO37 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 4 | 保留位 | R/W | 0h | 保留 |
| 3 | GPIO35 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 2 | GPIO34 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 1 | GPIO33 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 0 | GPIO32 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |

### 7.9.3.7 GPBCLEAR 寄存器

偏移地址：18h　复位值：00000000h

表 7-109 GPBCLEAR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | GPIO62 | GPIO61 | 保留 | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO39 | 保留 | GPIO37 | 保留 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-110 GPBCLEAR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO63 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 30 | GPIO62 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 29 | GPIO61 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 28-15 | 保留位 | R/W | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 14 | GPIO46 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 13 | GPIO45 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 12 | GPIO44 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 11 | GPIO43 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 10 | GPIO42 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 9 | GPIO41 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 8 | GPIO40 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 7 | GPIO39 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 6 | 保留位 | R/W | 0h | 保留 |
| 5 | GPIO37 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 4 | 保留位 | R/W | 0h | 保留 |
| 3 | GPIO35 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 2 | GPIO34 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 1 | GPIO33 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 0 | GPIO32 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |

### 7.9.3.8 GPBTOGGLE 寄存器

偏移地址：1Ch　复位值：00000000h

表 7-111 GPBTOGGLE 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GPIO63 | GPIO62 | GPIO61 | 保留 | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO39 | 保留 | GPIO37 | 保留 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-112 GPBTOGGLE 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31 | GPIO63 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 30 | GPIO62 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 29 | GPIO61 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 28-15 | 保留位 | R/W | 0h | 保留 |
| 14 | GPIO46 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 13 | GPIO45 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 12 | GPIO44 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 11 | GPIO43 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 10 | GPIO42 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 9 | GPIO41 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 8 | GPIO40 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 7 | GPIO39 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 6 | 保留位 | R/W | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 5 | GPIO37 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 4 | 保留位 | R/W | 0h | 保留 |
| 3 | GPIO35 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 2 | GPIO34 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 1 | GPIO33 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 0 | GPIO32 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |

### 7.9.3.9 GPHDAT 寄存器

偏移地址：70h　复位值：00000000h

表 7-113 GPHDAT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | GPIO245 | GPIO244 | 保留 | GPIO242 | GPIO241 | 保留 |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GPIO239 | GPIO238 | GPIO237 | 保留 | | | GPIO233 | GPIO232 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GPIO231 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | GPIO225 | GPIO224 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-114 GPHDAT 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-22 | 保留位 | R/W | 0h | 保留 |
| 21 | GPIO245 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 20 | GPIO244 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 19 | 保留位 | R/W | 0h | 保留 |

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 18 | GPIO242 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 17 | GPIO241 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 16 | 保留位 | R/W | 0h | 保留 |
| 15 | GPIO239 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 14 | GPIO238 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 13 | GPIO237 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 12-10 | 保留位 | R/W | 0h | 保留 |
| 9 | GPIO233 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 8 | GPIO232 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 7 | GPIO231 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 6 | GPIO230 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 5 | 保留位 | R/W | 0h | 保留 |
| 4 | GPIO228 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 3 | GPIO227 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 2 | GPIO226 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 1 | GPIO225 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |
| 0 | GPIO224 | R/W | 0h | 此引脚的数据寄存器<br>重置类型：SYSRSn |

### 7.9.3.10 GPHSET 寄存器

偏移地址：74h　复位值：00000000h

注：AIO 可配置为数字或模拟类型；只能在输入模式下工作；GPHDAT 反应的是输出值，所以不可配置，SET 和 CLEAR 寄存器可取消 AIO 位域：225,231,232,233,237,238,239,241,244,245

表 7-115 GPHSET 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | GPIO242 | 保留 | |
| R/W-0h | | | | | R/W-0h | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | 保留 | GPIO224 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-116 GPHSET 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-19 | 保留位 | R/W | 0h | 保留 |
| 18 | GPIO242 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 17-7 | 保留位 | R/W | 0h | 保留 |
| 6 | GPIO230 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 5 | 保留位 | R/W | 0h | 保留 |
| 4 | GPIO228 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 3 | GPIO227 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 2 | GPIO226 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |
| 1 | 保留位 | R/W | 0h | 保留 |
| 0 | GPIO224 | R/W | 0h | 输出置位<br>重置类型：SYSRSn |

### 7.9.3.11 GPHCLEAR 寄存器

偏移地址：78h　复位值：00000000h

表 7-117 GPHCLEAR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | GPIO242 | 保留 | |
| R/W-0h | | | | | R/W-0h | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | 保留 | GPIO224 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-118 GPHCLEAR 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-19 | 保留位 | R/W | 0h | 保留 |
| 18 | GPIO242 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 17-7 | 保留位 | R/W | 0h | 保留 |
| 6 | GPIO230 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 5 | 保留位 | R/W | 0h | 保留 |
| 4 | GPIO228 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 3 | GPIO227 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 2 | GPIO226 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |
| 1 | 保留位 | R/W | 0h | 保留 |
| 0 | GPIO224 | R/W | 0h | 输出清除<br>重置类型：SYSRSn |

### 7.9.3.12 GPHTOGGLE 寄存器

偏移地址：7Ch　复位值：00000000h

表 7-119 GPHTOGGLE 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | GPIO242 | 保留 | |
| R/W-0h | | | | | R/W-0h | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | 保留 | GPIO224 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 7-120 GPHTOGGLE 寄存器字段说明

| 位 | 名称 | 类型 | 复位 | 功能描述 |
|---|---|---|---|---|
| 31-19 | 保留位 | R/W | 0h | 保留 |
| 18 | GPIO242 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 17-7 | 保留位 | R/W | 0h | 保留 |
| 6 | GPIO230 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 5 | 保留位 | R/W | 0h | 保留 |
| 4 | GPIO228 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 3 | GPIO227 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 2 | GPIO226 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |
| 1 | 保留位 | R/W | 0h | 保留 |
| 0 | GPIO224 | R/W | 0h | 输出翻转<br>重置类型：SYSRSn |

## 7.9.4 GPIO 读数据寄存器

表 7-121 GPIO 读数据寄存器数据表

| 偏移量 | 寄存器 | 描述 |
|---|---|---|
| 0h | GPADAT_R | GPIO A数据读取寄存器 |
| 4h | GPBDAT_R | GPIO B数据读取寄存器 |
| 1Ch | GPHDAT_R | GPIO H数据读取寄存器 |

### 7.9.4.1 GPADAT_R 寄存器

偏移地址：0h

记录寄存器 GPADAT 本身的值。

表 7-122 GPADAT_R 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 数据 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 数据 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 数据 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 数据 | | | | | | | |
| R-0h | | | | | | | |

表 7-123 GPADAT_R 字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-0 | 数据 | R | 0h | 返回GPADAT寄存器的内容，写入此寄存器无效<br>复位类型：CPU1.SYSRSn |

### 7.9.4.2 GPBDAT_R 寄存器

偏移地址：4h

表 7-124 GPBDAT_R 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 数据 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 数据 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 数据 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 数据 | | | | | | | |
| R-0h | | | | | | | |

表 7-125 GPBDAT_R 字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-0 | 数据 | R | 0h | 返回GPBDAT寄存器的内容，写入此寄存器无效<br>复位类型：CPU1.SYSRSn |

### 7.9.4.3 GPHDAT_R 寄存器

偏移地址：1Ch

表 7-126 GPHDAT_R 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 数据 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 数据 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 数据 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 数据 | | | | | | | |
| R-0h | | | | | | | |

表 7-127 GPHDAT_R 字段说明

| 位 | 名称 | 类型 | 复位 | 取值及功能描述 |
|---|---|---|---|---|
| 31-0 | 数据 | R | 0h | 返回GPHDAT寄存器的内容，写入此寄存器无效<br>复位类型：CPU1.SYSRSn |

# 8 交叉开关（X-BAR)

## 8.1 简介

该交叉开关（本章中统称为 X-BAR）可灵活连接设备的输入端、输出端及内部资源，支持多种配置方式。该设备共包含六个 X-BAR：

- input X-BAR
- CLB input X-BAR
- output X-BAR
- CLB output X-BAR
- CLB X-BAR
- ePWM X-BAR

每个 X-BAR 的命名依据其信号接收位置。例如，input X-BAR 和 CLB input X-BAR 将外部信号引入设备内部，output X-BAR 和 CLB output X-BAR 将内部信号输出至 GPIO，CLB X-BAR 和 ePWMX-BAR 则分别将信号传输至 CLB 和 ePWM 模块。

## 8.2 输入交叉开关和 CLB 输入交叉开关

在该设备中，input X-BAR 用于将 GPIO 信号路由至多个不同 IP 模块，例如模数转换器（ADC）、电子控制单元（eCAP）、电子脉宽调制器（ePWM）以及外部中断模块。每个 input X-BAR 实例（INPUTx）的输入端可连接任意 GPIO ，而各实例的输出端则连接至设备中的不同 IP 模块。AIO 的数字输入端也可作为 input X-BAR 的输入端口。这种灵活性通过允许用户将任意 GPIO 连接至指定 input X-BAR 实例的输出端，从而缓解了外围设备多路复用的某些限制。需注意的是，input X-BAR 所选 GPIO 可配置为输入或输出。input X-BAR 仅将输入缓冲器上的信号连接至所选 input X-BAR 实例的输出端。因此，用户可实现诸如将 ePWM 输出路由至 eCAP 模块进行频率测试等操作。

input X-BAR 通过 INPUTxSELECT 寄存器进行配置，各 INPUTx 的目标端口如图所示。需注意各 INPUTx 的目标端口是固定的，不可由用户配置。

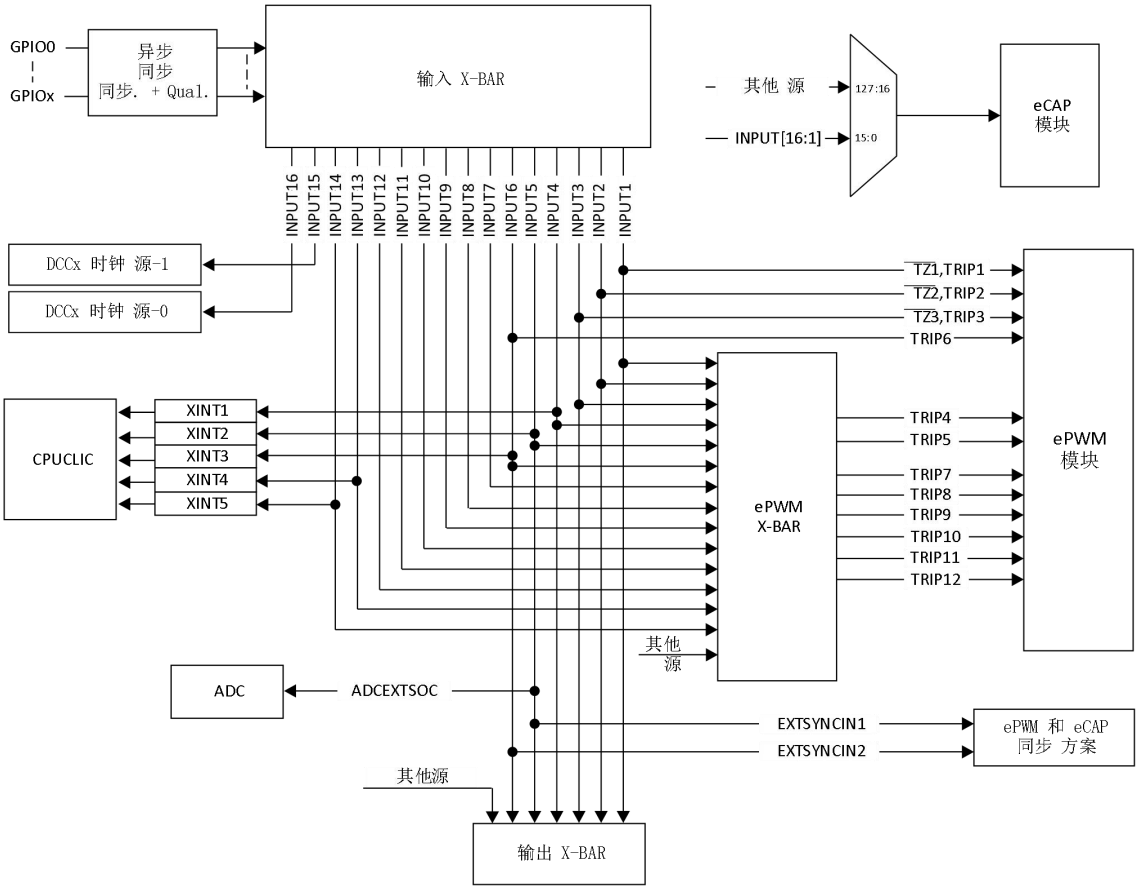



图 8-1 输入交叉开关

表 8-1 输入交叉开关终点

| 输入 | ECAP | ePWM 交叉开关 | CLB 交叉开关 | 输出交叉开关 | ePWM 跳闸输入 | CPU 外部中断 | ADC 转换启动 | ePWM/eCAP 同步 |
|---|---|---|---|---|---|---|---|---|
| INPUTXBAR1 | 支持 | 支持 | 支持 | 支持 | TZ1, TRIP1 | | | |
| INPUTXBAR2 | 支持 | 支持 | 支持 | 支持 | TZ2, TRIP2 | | | |
| INPUTXBAR3 | 支持 | 支持 | 支持 | 支持 | TZ3, TRIP3 | | | |
| INPUTXBAR4 | 支持 | 支持 | 支持 | 支持 | | XINT1 | | |
| INPUTXBAR5 | 支持 | 支持 | 支持 | 支持 | | XINT2 | ADCEXTSOC | EXTSYNCIN1 |
| INPUTXBAR6 | 支持 | 支持 | 支持 | 支持 | TRIP6 | XINT3 | | EXTSYNCIN2 |
| INPUTXBAR7 | 支持 | 支持 | 支持 | | | | | |
| INPUTXBAR8 | 支持 | 支持 | 支持 | | | | | |
| INPUTXBAR9 | 支持 | 支持 | 支持 | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| INPUTXBAR10 | 支持 | 支持 | 支持 | | | | | |
| INPUTXBAR11 | 支持 | 支持 | 支持 | | | | | |
| INPUTXBAR12 | 支持 | 支持 | 支持 | | | | | |
| INPUTXBAR13 | 支持 | 支持 | 支持 | | | XINT4 | | |
| INPUTXBAR14 | 支持 | 支持 | 支持 | | | XINT5 | | |
| INPUTXBAR15 | 支持 | | | | | | | |
| INPUTXBAR16 | 支持 | | | | | | | |

表 8-2 输入交叉开关和 CLB 输入交叉开关数据表

| 输入 | ECAP | ePWM 交叉开关 | CLB 交叉开关 | 输出交叉开关 | ePWM 跳闸输入 | CPU 外部中断 | ADC 转换启动 | ePWM/eCAP 同步 |
|---|---|---|---|---|---|---|---|---|
| INPUTXBAR1 | 支持 | 支持 | 支持 | 支持 | TZ1,TRIP1 | | | |
| INPUTXBAR2 | 支持 | 支持 | 支持 | 支持 | TZ2,TRIP2 | | | |
| INPUTXBAR3 | 支持 | 支持 | 支持 | 支持 | TZ3,TRIP3 | | | |
| INPUTXBAR4 | 支持 | 支持 | 支持 | 支持 | | XINT1 | | |
| INPUTXBAR5 | 支持 | 支持 | 支持 | 支持 | | XINT2 | ADCEXTSOC | EXTSYNCIN1 |
| INPUTXBAR6 | 支持 | 支持 | 支持 | 支持 | TRIP6 | XINT3 | | EXTSYNCIN1 |
| INPUTXBAR7 | 支持 | 支持 | 支持 | | | | | |
| INPUTXBAR8 | 支持 | 支持 | 支持 | | | | | |
| INPUTXBAR9 | 支持 | 支持 | 支持 | | | | | |
| INPUTXBAR10 | 支持 | 支持 | 支持 | | | | | |
| INPUTXBAR11 | 支持 | 支持 | 支持 | | | | | |
| INPUTXBAR12 | 支持 | 支持 | 支持 | | | | | |
| INPUTXBAR13 | 支持 | 支持 | 支持 | | | XINT4 | | |
| INPUTXBAR14 | 支持 | 支持 | 支持 | | | XINT5 | | |
| INPUTXBAR15 | 支持 | | | | | | | |
| INPUTXBAR16 | 支持 | | | | | | | |

### 8.2.1 CLB 输入交叉开关

CLB input X-BAR 在架构上与 input X-BAR 完全相同，唯一的区别在于每个 INPUTx 的接收对象。如表所示，每个 INPUTx 的接收对象仅限于 CLB Tile。这使得 CLB Tile 无需同时使用 input X-BAR 和 CLB X-BAR 即可访问 GPIO。

注意：通过 XBAR 接入 CLB 的信号必须在 CLB 内保持同步。

表 8-3 CLB 输入交叉开关终点

| 输入 | CLB |
|---|---|
| CLBINPUTXBAR1 | 支持 |
| CLBINPUTXBAR2 | 支持 |
| CLBINPUTXBAR3 | 支持 |
| CLBINPUTXBAR4 | 支持 |
| CLBINPUTXBAR5 | 支持 |

| 输入 | CLB |
| --- | --- |
| CLBINPUTXBAR6 | 支持 |
| CLBINPUTXBAR7 | 支持 |
| CLBINPUTXBAR8 | 支持 |
| CLBINPUTXBAR9 | 支持 |
| CLBINPUTXBAR10 | 支持 |
| CLBINPUTXBAR11 | 支持 |
| CLBINPUTXBAR12 | 支持 |
| CLBINPUTXBAR13 | 支持 |
| CLBINPUTXBAR14 | 支持 |
| CLBINPUTXBAR15 | 支持 |
| CLBINPUTXBAR16 | 支持 |

# 8.3 ePWM、CLB 和 GPIO 输出交叉开关

本节介绍 ePWM、CLB 及 GPIO output X-BAR。

## 8.3.1 ePWM 交叉开关

ePWM X-BAR 负责将信号传输至 ePWM 模块。具体而言，该模块与每个 ePWM 模块的数字比较（DC）子模块相连，用于实现跳闸区设置、同步等操作。图中展示了 ePWM X-BAR 的架构。需注意的是，ePWM X-BAR 的架构与 GPIO output X-BAR 完全相同（输出锁存器除外）。

### 8.3.1.1 ePWM 交叉开关架构

ePWM X-BAR 具有八个输出端口，每个输出端口均连接至对应的 ePWM 模块。图中展示了单个输出端口的架构示意图，但该输出端口的架构与其他所有输出端口完全相同。

首先，通过确定可传输至 ePWM 的信号类型。每个 TRIPx 输出端口最多可选择一个信号通过多路复用器。使用 TRIPxMUX0TO15CFG 和 TRIPxMUX16TO31CFG 寄存器选择 ePWM X-BAR 的输入信号。若需将任何信号传输至 ePWM，需通过 TRIPxMUXENABLE 寄存器启用该信号。所有启用的信号在传输至 ePWM 的对应 TRIPx 信号前，会经过逻辑或运算处理。若需选择性反转信号方向，可使用 tripoutinv 寄存器进行操作。

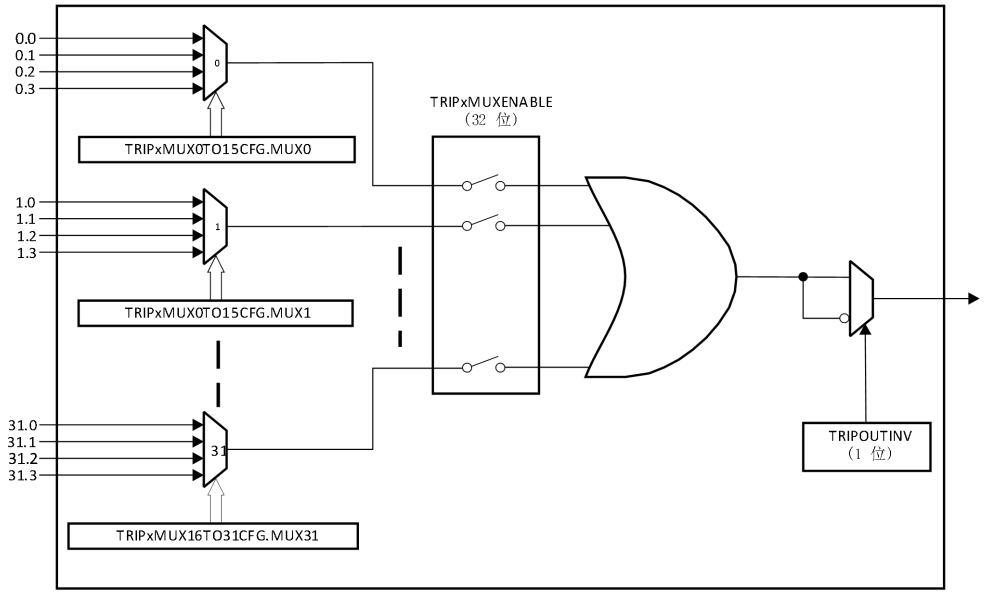



图 8-2 ePWM 交叉开关架构 - 单个输出

表 8-4 EPWM 交叉开关多路复用器配置表

| Mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| G0 | CMPSS1_CTRIPH | CMPSS1_CTRIPH_OR_CTRIPL | ADCAEVT1 | ECAP1_OUT |
| G1 | CMPSS1_CTRIPL | INPUTXBAR1 | CLB1_OUT12 | ADCCEVT1 |
| G2 | CMPSS2_CTRIPH | CMPSS2_CTRIPH_OR_CTRIPL | ADCAEVT2 | ECAP2_OUT |
| G3 | CMPSS2_CTRIPL | INPUTXBAR2 | CLB1_OUT13 | ADCCEVT2 |
| G4 | CMPSS3_CTRIPH | CMPSS3_CTRIPH_OR_CTRIPL | ADCAEVT3 | ECAP3_OUT |
| G5 | CMPSS3_CTRIPL | INPUTXBAR3 | CLB2_OUT12 | ADCCEVT3 |
| G6 | CMPSS4_CTRIPH | CMPSS4_CTRIPH_OR_CTRIPL | ADCAEVT4 | 保留 |
| G7 | CMPSS4_CTRIPL | INPUTXBAR4 | CLB2_OUT13 | ADCCEVT4 |
| G8 | 保留 | 保留 | 保留 | 保留 |
| G9 | 保留 | INPUTXBAR5 | 保留 | 保留 |
| G10 | 保留 | 保留 | 保留 | 保留 |
| G11 | 保留 | INPUTXBAR6 | 保留 | 保留 |
| G12 | 保留 | 保留 | 保留 | 保留 |
| G13 | 保留 | ADCSOCAO | 保留 | 保留 |
| G14 | 保留 | 保留 | 保留 | EXTSYNCOUT |
| G15 | 保留 | ADCSOCBO | 保留 | 保留 |
| G16 | 保留 | 保留 | 保留 | 保留 |
| G17 | 保留 | INPUTXBAR7 | 保留 | 保留 |
| G18 | 保留 | 保留 | 保留 | 保留 |
| G19 | 保留 | INPUTXBAR8 | 保留 | 保留 |
| G20 | 保留 | 保留 | 保留 | 保留 |
| G21 | 保留 | INPUTXBAR9 | 保留 | 保留 |
| G22 | 保留 | 保留 | 保留 | 保留 |
| G23 | 保留 | INPUTXBAR10 | 保留 | 保留 |
| G24 | 保留 | 保留 | 保留 | 保留 |
| G25 | 保留 | INPUTXBAR11 | 保留 | 保留 |
| G26 | 保留 | 保留 | 保留 | 保留 |

| Mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| G27 | 保留 | INPUTXBAR12 | 保留 | 保留 |
| G28 | 保留 | 保留 | 保留 | 保留 |
| G29 | 保留 | INPUTXBAR13 | 保留 | 保留 |
| G30 | 保留 | 保留 | 保留 | 保留 |
| G31 | 保留 | INPUTXBAR14 | 保留 | 保留 |

## 8.3.2 CLB 交叉开关

CLB X-BAR 将信号传输至 CLB 模块。图中展示了 CLB X-BAR 的架构。需注意， CLB X-BAR 的架构与 GPIO output X-BAR 完全相同（输出锁存器除外）。

### 8.3.2.1 CLB 交叉开关架构

CLB X-BAR 配备八个输出端口，每个 CLB 模块均连接一个输出端口。图中展示了单个输出端口的架构，但该输出端口的架构与其他所有输出端口完全相同。

首先，通过确定可传输至 CLB 的信号。每个 AUXSIGx 输出端口最多可选择一个信号（共 31 个多路复用器）。使用 AUXSIGxMUX0TO15CFG 和 AUXSIGxMUX16TO31CFG 寄存器选择各多路复用器的输入信号。若需将任何信号传输至 CLB ，需在 AUXSIGxMUXENABLE 寄存器中启用对应多路复用器。所有启用的多路复用器在信号传输至 CLB 的 AUXSIGx 信号前，会进行逻辑或运算。若需选择性反转信号，可使用 auxsigoutinv 寄存器。

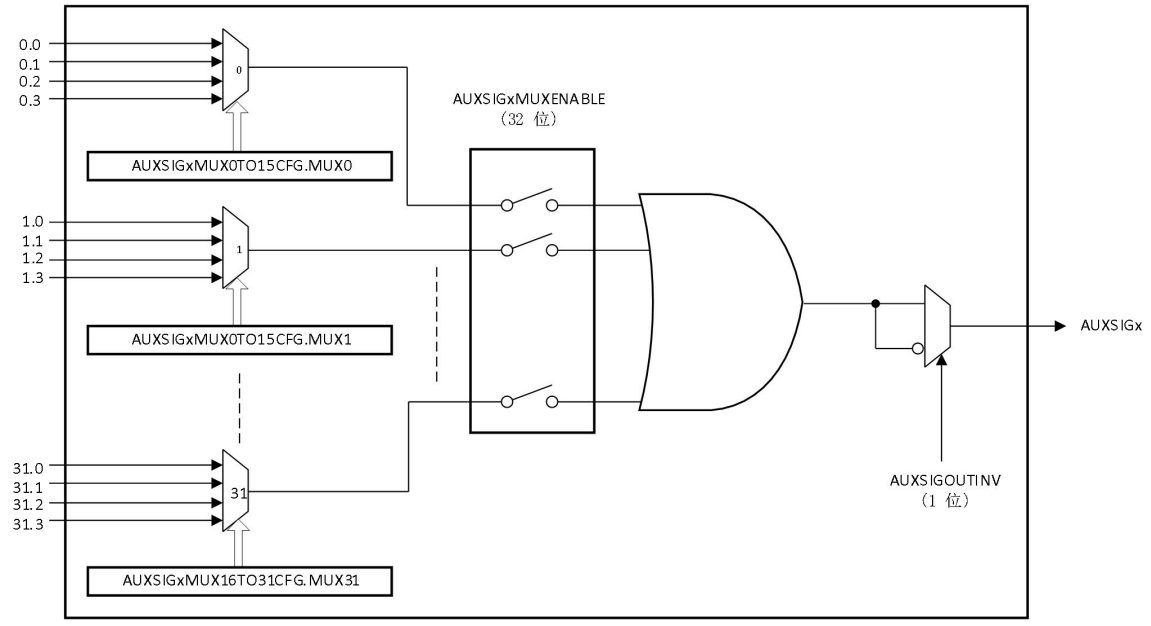


图 8-3 CLB 交叉开关架构 - 单输出

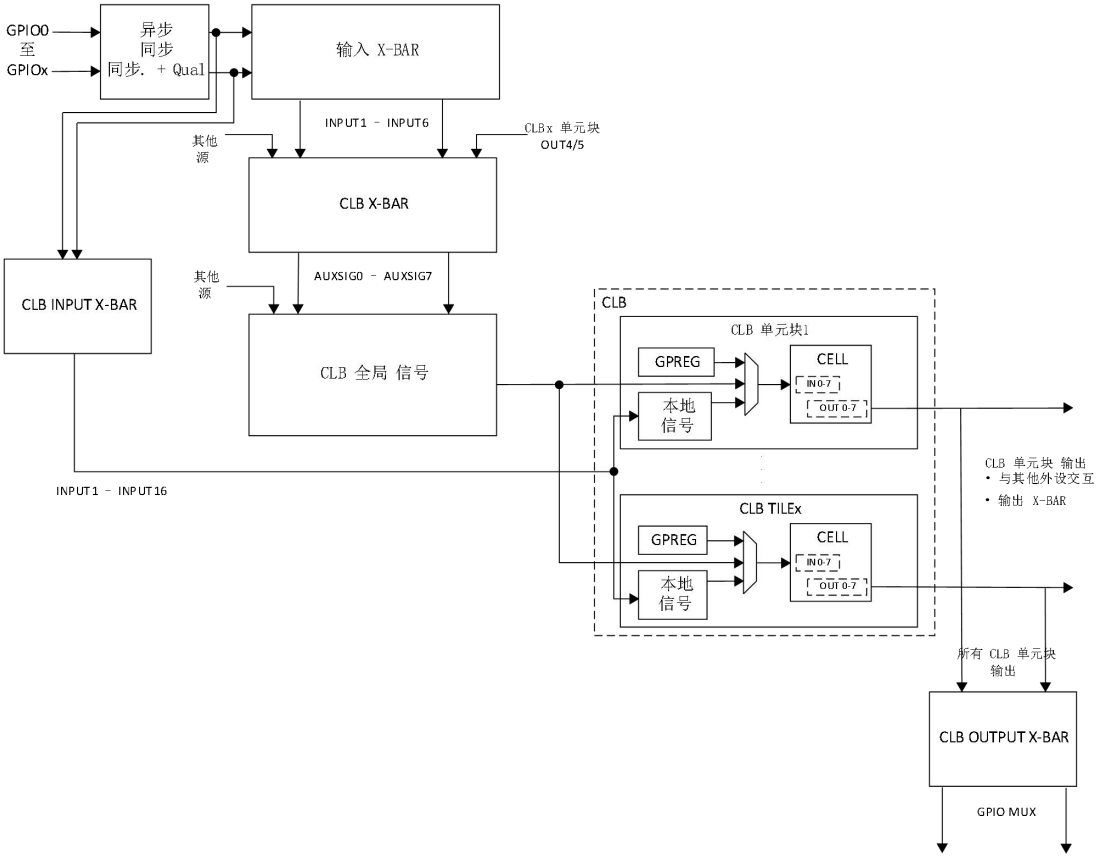



图 8-4 GPIO 到 CLB 逻辑块连接

表 8-5 CLB 交叉开关多路复用器配置表

| Mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| G0 | CMPSS1_CTRIPH | CMPSS1_CTRIPH_OR_CTRIPL | ADCAEVT1 | ECAP1_OUT |
| G1 | CMPSS1_CTRIPL | INPUTXBAR1 | CLB1_OUT12 | ADCCEVT1 |
| G2 | CMPSS2_CTRIPH | CMPSS2_CTRIPH_OR_CTRIPL | ADCAEVT2 | ECAP2_OUT |
| G3 | CMPSS2_CTRIPL | INPUTXBAR2 | CLB1_OUT13 | ADCCEVT2 |
| G4 | CMPSS3_CTRIPH | CMPSS3_CTRIPH_OR_CTRIPL | ADCAEVT3 | ECAP3_OUT |
| G5 | CMPSS3_CTRIPL | INPUTXBAR3 | CLB2_OUT12 | ADCCEVT3 |
| G6 | CMPSS4_CTRIPH | CMPSS4_CTRIPH_OR_CTRIPL | ADCAEVT4 | 保留 |
| G7 | CMPSS4_CTRIPL | INPUTXBAR4 | CLB2_OUT13 | ADCCEVT4 |
| G8 | 保留 | 保留 | 保留 | 保留 |
| G9 | 保留 | INPUTXBAR5 | 保留 | 保留 |
| G10 | 保留 | 保留 | 保留 | 保留 |
| G11 | 保留 | INPUTXBAR6 | 保留 | 保留 |
| G12 | 保留 | 保留 | 保留 | 保留 |
| G13 | 保留 | ADCSOCAO | 保留 | 保留 |
| G14 | 保留 | 保留 | 保留 | EXTSYNCOUT |
| G15 | 保留 | ADCSOCBO | 保留 | 保留 |
| G16 | 保留 | 保留 | 保留 | 保留 |
| G17 | 保留 | INPUTXBAR7 | 保留 | 保留 |

| Mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| G18 | 保留 | 保留 | 保留 | 保留 |
| G19 | 保留 | INPUTXBAR8 | 保留 | 保留 |
| G20 | 保留 | 保留 | 保留 | 保留 |
| G21 | 保留 | INPUTXBAR9 | 保留 | 保留 |
| G22 | 保留 | 保留 | 保留 | 保留 |
| G23 | 保留 | INPUTXBAR10 | 保留 | 保留 |
| G24 | 保留 | 保留 | 保留 | 保留 |
| G25 | 保留 | INPUTXBAR11 | 保留 | 保留 |
| G26 | 保留 | 保留 | 保留 | 保留 |
| G27 | 保留 | INPUTXBAR12 | 保留 | 保留 |
| G28 | 保留 | 保留 | 保留 | 保留 |
| G29 | 保留 | INPUTXBAR13 | 保留 | 保留 |
| G30 | 保留 | 保留 | 保留 | 保留 |
| G31 | 保留 | INPUTXBAR14 | 保留 | 保留 |

### 8.3.3 GPIO 输出交叉开关

GPIO output X-BAR 将设备内部的信号传输至 GPIO 。图中展示了 GPIO output X-BAR 的架构。该 X-BAR 包含八个输出端口，每个端口至少对应 GPIO 多路复用器上的一个位置，标记为 OUTPUTXBARx。X-BAR 支持选择单个信号或最多 32 个信号的逻辑或运算。

#### 8.3.3.1 GPIO 输出交叉开关架构

GPIO output X-BAR 模块共有八个输出端口，这些信号均接入 GPIO 模块。图中. GPIO output X-BAR 架构图中展示了单个输出的架构示意图，但该输出与其他所有输出的架构完全相同。需注意，除输出锁存器外，X-BAR 的架构与 ePWM X-BAR 模块基本一致。

首先，通过参考表确定可传输至 GPIO 的信号类型。每个 OUTPUTXBARx 输出最多可选择一个信号（共 32 个多路复用器）。使用 OUTPUTxMUX0TO15CFG 和 OUTPUTxMUX16TO31CFG 寄存器配置各多路复用器的输入端口。若需将信号传输至 GPIO ，需启用 OUTPUTxMUXENABLE 寄存器中的对应多路复用器。所有启用的多路复用器在信号传输至 GPIO 模块的 OUTPUTx 信号前，会进行逻辑或运算处理。若需对信号进行反相处理，可使用 outputinv 寄存器。只有当通过 GpioCtrlRegs、

GPxMUX 和 GpioCtrlRegs、GPxGMUX 寄存器正确配置 OUTPUTx 多路复用选项时，GPIO 模块才能识别该信号。

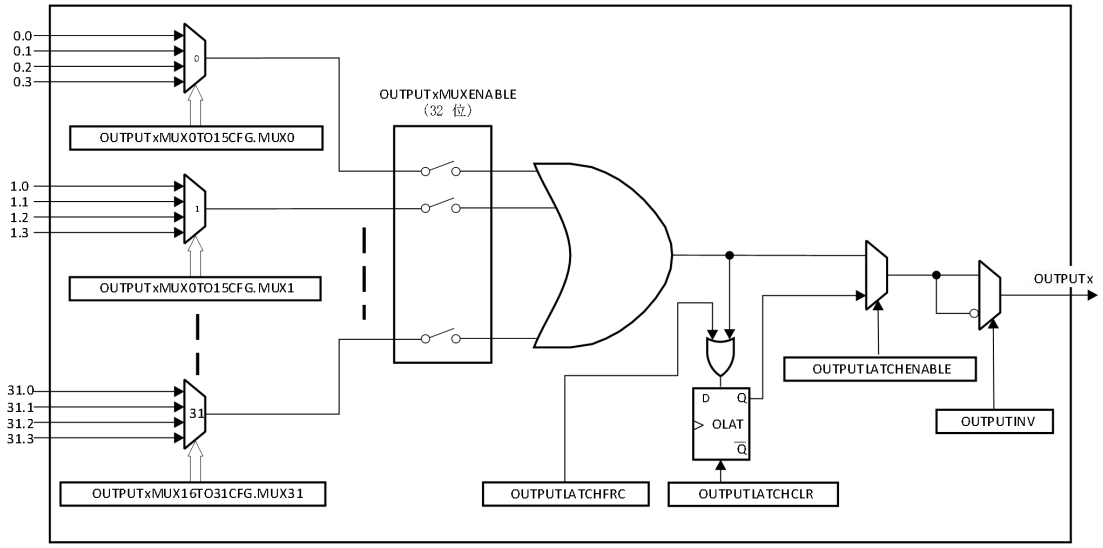



图 8-5 GPIO 输出交叉开关架构

表 8-6 输出交叉开关多路复用器配置表

| Mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| G0 | CMPSS1_CTRIPH | CMPSS1_CTRIPH_OR_CTRIPL | ADCAEVT1 | ECAP1_OUT |
| G1 | CMPSS1_CTRIPL | INPUTXBAR1 | CLB1_OUT12 | ADCCEVT1 |
| G2 | CMPSS2_CTRIPH | CMPSS2_CTRIPH_OR_CTRIPL | ADCAEVT2 | ECAP2_OUT |
| G3 | CMPSS2_CTRIPL | INPUTXBAR2 | CLB1_OUT13 | ADCCEVT2 |
| G4 | CMPSS3_CTRIPH | CMPSS3_CTRIPH_OR_CTRIPL | ADCAEVT3 | ECAP3_OUT |
| G5 | CMPSS3_CTRIPL | INPUTXBAR3 | CLB2_OUT12 | ADCCEVT3 |
| G6 | CMPSS4_CTRIPH | CMPSS4_CTRIPH_OR_CTRIPL | ADCAEVT4 | 保留 |
| G7 | CMPSS4_CTRIPL | INPUTXBAR4 | CLB2_OUT13 | ADCCEVT4 |
| G8 | 保留 | 保留 | 保留 | 保留 |
| G9 | 保留 | INPUTXBAR5 | 保留 | 保留 |
| G10 | 保留 | 保留 | 保留 | 保留 |
| G11 | 保留 | INPUTXBAR6 | 保留 | 保留 |
| G12 | 保留 | 保留 | 保留 | 保留 |
| G13 | 保留 | ADCSOCAO | 保留 | 保留 |
| G14 | 保留 | 保留 | 保留 | EXTSYNCOUT |
| G15 | 保留 | ADCSOCBO | 保留 | 保留 |
| G16 | 保留 | 保留 | 保留 | 保留 |
| G17 | 保留 | 保留 | 保留 | 保留 |
| G18 | 保留 | 保留 | 保留 | 保留 |
| G19 | 保留 | 保留 | 保留 | 保留 |
| G20 | 保留 | 保留 | 保留 | 保留 |
| G21 | 保留 | 保留 | 保留 | 保留 |
| G22 | 保留 | 保留 | 保留 | 保留 |
| G23 | 保留 | 保留 | 保留 | 保留 |

| Mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| G24 | 保留 | 保留 | 保留 | 保留 |
| G25 | 保留 | 保留 | 保留 | 保留 |
| G26 | 保留 | 保留 | 保留 | 保留 |
| G27 | 保留 | 保留 | 保留 | 保留 |
| G28 | 保留 | 保留 | 保留 | 保留 |
| G29 | 保留 | 保留 | 保留 | 保留 |
| G30 | 保留 | 保留 | 保留 | 保留 |
| G31 | 保留 | 保留 | 保留 | 保留 |

### 8.3.4 CLB 输出交叉开关

CLB output X-BAR 将 CLB 内核的信号提取并输出至 GPIO 。

#### 8.3.4.1 CLB 输出交叉开关架构

CLB output X-BAR 有八个输出端口，均连接至 GPIO 模块。 CLB output X-BAR 的架构与 GPIO output X-BAR 完全相同。首先，参考表确定可传输至 GPIO 的信号。

表 8-7 CLB 输出交叉开关多路复用器配置表

| Mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| G0 | CLB1_OUT0 | 保留 | 保留 | 保留 |
| G1 | CLB1_OUT1 | 保留 | 保留 | 保留 |
| G2 | CLB1_OUT2 | 保留 | 保留 | 保留 |
| G3 | CLB1_OUT3 | 保留 | 保留 | 保留 |
| G4 | CLB1_OUT4 | 保留 | 保留 | 保留 |
| G5 | CLB1_OUT5 | 保留 | 保留 | 保留 |
| G6 | CLB1_OUT6 | 保留 | 保留 | 保留 |
| G7 | CLB1_OUT7 | 保留 | 保留 | 保留 |
| G8 | CLB2_OUT0 | 保留 | 保留 | 保留 |
| G9 | CLB2_OUT1 | 保留 | 保留 | 保留 |
| G10 | CLB2_OUT2 | 保留 | 保留 | 保留 |
| G11 | CLB2_OUT3 | 保留 | 保留 | 保留 |
| G12 | CLB2_OUT4 | 保留 | 保留 | 保留 |
| G13 | CLB2_OUT5 | 保留 | 保留 | 保留 |
| G14 | CLB2_OUT6 | 保留 | 保留 | 保留 |
| G15 | CLB2_OUT7 | 保留 | 保留 | 保留 |
| G16 | 保留 | 保留 | 保留 | 保留 |
| G17 | 保留 | 保留 | 保留 | 保留 |
| G18 | 保留 | 保留 | 保留 | 保留 |
| G19 | 保留 | 保留 | 保留 | 保留 |
| G20 | 保留 | 保留 | 保留 | 保留 |
| G21 | 保留 | 保留 | 保留 | 保留 |

| Mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| G22 | 保留 | 保留 | 保留 | 保留 |
| G23 | 保留 | 保留 | 保留 | 保留 |
| G24 | 保留 | 保留 | 保留 | 保留 |
| G25 | 保留 | 保留 | 保留 | 保留 |
| G26 | 保留 | 保留 | 保留 | 保留 |
| G27 | 保留 | 保留 | 保留 | 保留 |
| G28 | 保留 | 保留 | 保留 | 保留 |
| G29 | 保留 | 保留 | 保留 | 保留 |
| G30 | 保留 | 保留 | 保留 | 保留 |
| G31 | 保留 | 保留 | 保留 | 保留 |

### 8.3.5 交叉开关标志

除 CMPSS 信号外，ePWM X-BAR 与 output X-BAR 共享全部输入信号。由于 ePWM X-BAR、 CLB X-BAR 和 output X-BAR 的输入信号结构相同，所有 X-BAR 模块均采用统一的输入标志位组来标识触发信号。当事件发生时，软件可直接检查这些标志位。每个输入信号在 XBARFLGx 寄存器中对应一个独立位，该标志位会持续保持激活状态，直至通过对应的 XBARCLRx 寄存器进行清除。

注意：并非所有输入源都会被路由到所有 X-BAR 模块。

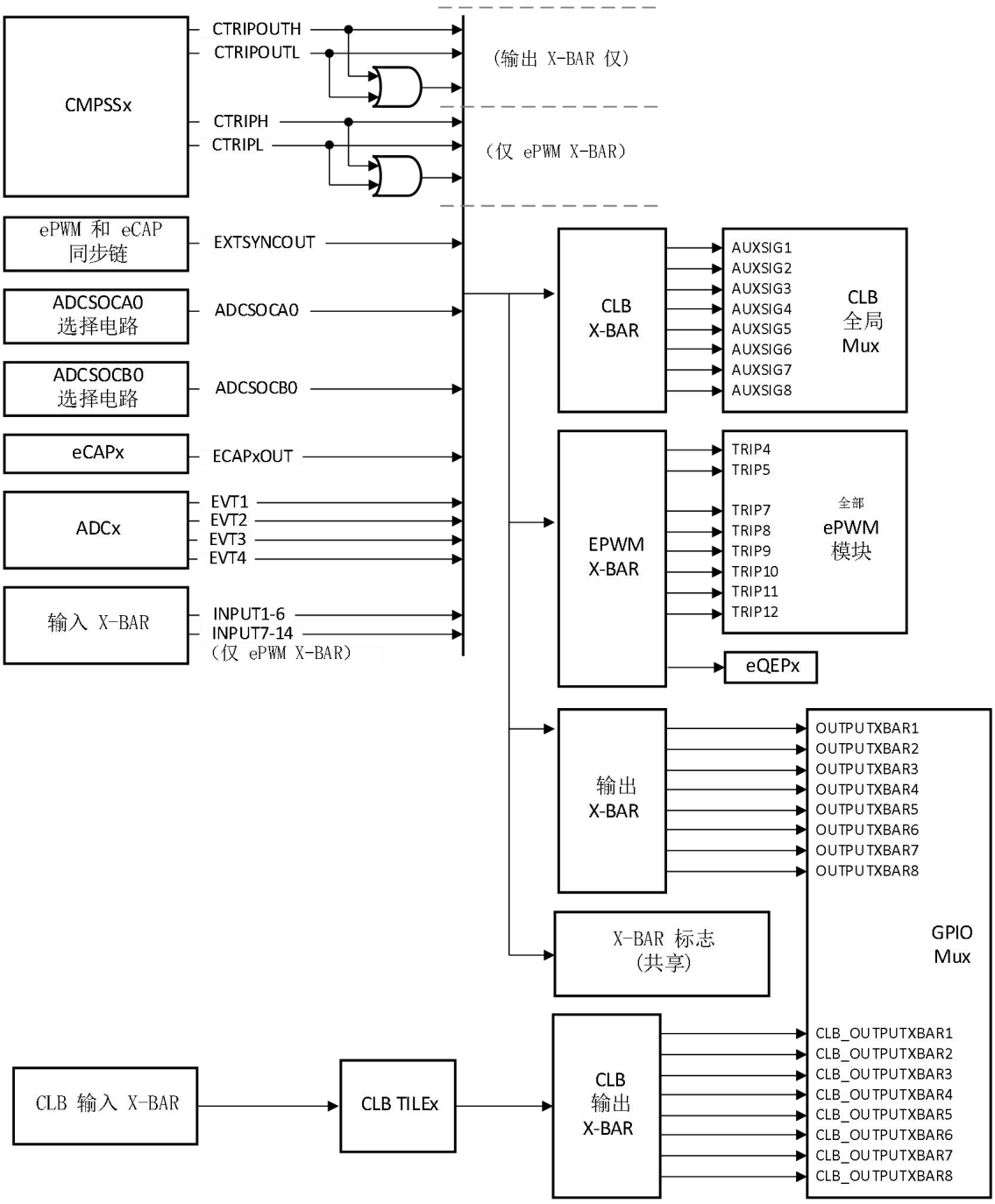



图 8-6 交叉开关输入源

## 8.4 寄存器描述

本节阐述交叉开关寄存器的功能。

### 8.4.1 交叉开关基地址表

表 8-8 交叉开关基地址表

| 比特域名称 | | 基地址 |
|---|---|---|
| 例化名 | 结构体 | |
| InputXbarRegs | INPUT_XBAR_REGS | 0x0064_0000 |
| XbarRegs | XBAR_REGS | 0x0064_0040 |
| ClbInputXbarRegs | INPUT_XBAR_REGS | 0x0064_00C0 |
| EPwmXbarRegs | EPWM_XBAR_REGS | 0x0064_0200 |
| ClbXbarRegs | CLB_XBAR_REGS | 0x0064_0280 |
| OutputXbarRegs | OUTPUT_XBAR_REGS | 0x0064_0300 |
| ClbOutputXbarRegs | OUTPUT_XBAR_REGS | 0x0064_0580 |

### 8.4.2 INPUT_XBAR_REGS 寄存器

表中列出了 INPUT_XBARRegs 寄存器的内存映射寄存器。表中未列出的所有寄存器偏移地址均应视为保留位置，且不得修改寄存器内容。

表 8-9 INPUT_XBAR_REGS 寄存器

| 偏移量 | 寄存器 | 描述 |
|---|---|---|
| 0h | INPUT1SELECT | INPUT1 输入选择寄存器 |
| 2h | INPUT2SELECT | INPUT2 输入选择寄存器 |
| 4h | INPUT3SELECT | INPUT3 输入选择寄存器 |
| 6h | INPUT4SELECT | INPUT4 输入选择寄存器 |
| 8h | INPUT5SELECT | INPUT5 输入选择寄存器 |
| Ah | INPUT6SELECT | INPUT6 输入选择寄存器 |
| Ch | INPUT7SELECT | INPUT7 输入选择寄存器 |
| Eh | INPUT8SELECT | INPUT8 输入选择寄存器 |
| 10h | INPUT9SELECT | INPUT9 输入选择寄存器 |
| 12h | INPUT10SELECT | INPUT10 输入选择寄存器 |
| 14h | INPUT11SELECT | INPUT11 输入选择寄存器 |
| 16h | INPUT12SELECT | INPUT12 输入选择寄存器 |
| 18h | INPUT13SELECT | INPUT13 输入选择寄存器 |
| 1Ah | INPUT14SELECT | INPUT14 输入选择寄存器 |
| 1Ch | INPUT15SELECT | INPUT15 输入选择寄存器 |
| 1Eh | INPUT16SELECT | INPUT16 输入选择寄存器 |
| 3Ch | INPUTSELECTLOCK | 输入选择锁定寄存器 |

复杂位访问类型经编码后可适配于小型表格单元格。表中展示了本节中用于访问类型的编码方案。

表 8-10 INPUT_XBAR_REGS 访问类型代码

| 访问类型 | 代码 | 说明 |
|---|---|---|
| 读类型 | | |

| 访问类型 | 代码 | 说明 |
| --- | --- | --- |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回0 |
| 写类型 | | |
| W | W | 写 |
| WSonce | W<br>Sonce | 写<br>仅设置1次 |
| 复位或默认值 | | |
| -n | | 复位后的值或默认值 |

INPUT1SELECT-INPUT16SELECT 分别对应 INPUT1-INPUT16 的输入选择，所用寄存器结构一致。

### 8.4.2.1 INPUT1SELECT 寄存器

偏移地址：0h　复位值：FFFEh

INPUT1 输入选择寄存器（GPIO0 至 x）。

表 8-11 INPUT1SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-12 INPUT1SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 15～0 | 选择 | R/W | FFFEh | 为INPUT1信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.2 INPUT2SELECT 寄存器

偏移地址：2h　复位值：FFFEh

INPUT2 输入选择寄存器（GPIO0 至 x）。

表 8-13 INPUT2SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-14 INPUT2SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT2信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.3 INPUT3SELECT 寄存器

偏移地址：4h　复位值：FFFEh

INPUT3 输入选择寄存器（GPIO0 至 x）。

表 8-15 INPUT3SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-16 INPUT3SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT3信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.4 INPUT4SELECT 寄存器

偏移地址：6h　复位值：FFFEh

INPUT4 输入选择寄存器（GPIO0 至 x）。

表 8-17 INPUT4SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-18 INPUT4SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT4信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.5 INPUT5SELECT 寄存器

偏移地址：8h　复位值：FFFEh

INPUT5 输入选择寄存器（GPIO0 至 x）。

表 8-19 INPUT5SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-20 INPUT5SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT5信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.6 INPUT6SELECT 寄存器

偏移地址：ah　复位值：FFFEh

INPUT6 输入选择寄存器（GPIO0 至 x）。

表 8-21 INPUT6SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-22 INPUT6SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT1信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

## 8.4.2.7 INPUT7SELECT 寄存器

偏移地址：ch　复位值：FFFEh

INPUT7 输入选择寄存器（GPIO0 至 x）。

表 8-23 INPUT7SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-24 INPUT7SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT7信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.8 INPUT8SELECT 寄存器

偏移地址：eh　复位值：FFFEh

INPUT8 输入选择寄存器（GPIO0 至 x）。

表 8-25 INPUT8SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-26 INPUT8SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT8信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.9 INPUT9SELECT 寄存器

偏移地址：10h　复位值：FFFEh

INPUT9 输入选择寄存器（GPIO0 至 x）。

表 8-27 INPUT9SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-28 INPUT9SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT9信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

## 8.4.2.10 INPUT10SELECT 寄存器

偏移地址：12h　复位值：FFFEh

INPUT10 输入选择寄存器（GPIO0 至 x）。

表 8-29 INPUT10SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-30 INPUT10SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT10信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.11 INPUT11SELECT 寄存器

偏移地址：14h　复位值：FFFEh

INPUT11 输入选择寄存器（GPIO0 至 x）。

表 8-31 INPUT11SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-32 INPUT11SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT11信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.12 INPUT12SELECT 寄存器

偏移地址：16h　复位值：FFFEh

INPUT12 输入选择寄存器（GPIO0 至 x）。

表 8-33 INPUT12SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-34 INPUT12SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT12信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

## 8.4.2.13 INPUT13SELECT 寄存器

偏移地址：18h　复位值：FFFEh

INPUT13 输入选择寄存器（GPIO0 至 x）。

表 8-35 INPUT13SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-36 INPUT13SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT13信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.14 INPUT14SELECT 寄存器

偏移地址：1ah　复位值：FFFEh

INPUT14 输入选择寄存器（GPIO0 至 x）。

表 8-37 INPUT14SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-38 INPUT14SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT14信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.15 INPUT15SELECT 寄存器

偏移地址：1ch　复位值：FFFEh

INPUT15 输入选择寄存器（GPIO0 至 x）。

表 8-39 INPUT15SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-40 INPUT15SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT15信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

## 8.4.2.16 INPUT16SELECT 寄存器

偏移地址：1eh　复位值：FFFEh

INPUT16 输入选择寄存器（GPIO0 至 x）。

表 8-41 INPUT16SELECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 选择 | | | | | | | |
| R/W-FFFEh | | | | | | | |

表 8-42 INPUT16SELECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15～0 | 选择 | R/W | FFFEh | 为INPUT16信号选择GPIO：<br>0x0：选择GPIO0<br>0x1：选择GPIO1<br>0x2：选择GPIO2<br>...<br>0xFFFD：‘1’将驱动目标引脚<br>0xFFFE：‘1’将驱动目标引脚<br>0xFFFF：‘0’将驱动目标引脚<br>注意：当选择值超过设备上可用GPIO管脚数量时(0xFFFF除外)，驱动值为‘1’。<br>重置类型：CPU1.SYSRSN |

### 8.4.2.17 INPUTSELECTLOCK 寄存器

偏移地址：3ch　复位值：00000000h

输入选择锁定寄存器。

表 8-43 INPUTSELECTLOCK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INPUT16SELECT | INPUT15SELECT | INPUT14SELECT | INPUT13SELECT | INPUT12SELECT | INPUT11SELECT | INPUT10SELECT | INPUT9SELECT |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INPUT8SELECT | INPUT7SELECT | INPUT6SELECT | INPUT5SELECT | INPUT4SELECT | INPUT3SELECT | INPUT2SELECT | INPUT1SELECT |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |

表 8-44 INPUTSELECTLOCK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31～16 | 保留位 | R-0 | 0h | 保留 |
| 15 | INPUT16SELECT | R/WSonce | 0h | INPUT16选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 14 | INPUT15SELECT | R/WSonce | 0h | INPUT15选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 13 | INPUT14SELECT | R/WSonce | 0h | INPUT14选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 12 | INPUT13SELECT | R/WSonce | 0h | INPUT13选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 11 | INPUT12SELECT | R/WSonce | 0h | INPUT12选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 10 | INPUT11SELECT | R/WSonce | 0h | INPUT11选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 9 | INPUT10SELECT | R/WSonce | 0h | INPUT10选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 8 | INPUT9SELECT | R/WSonce | 0h | INPUT9选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 7 | INPUT8SELECT | R/WSonce | 0h | INPUT8选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 6 | INPUT7SELECT | R/WSonce | 0h | INPUT7选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 5 | INPUT6SELECT | R/WSonce | 0h | INPUT6选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 4 | INPUT5SELECT | R/WSonce | 0h | INPUT5选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 3 | INPUT4SELECT | R/WSonce | 0h | INPUT4选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 2 | INPUT3SELECT | R/WSonce | 0h | INPUT3选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 1 | INPUT2SELECT | R/WSonce | 0h | INPUT2选择寄存器的锁定位。<br>0:寄存器不锁定 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1:寄存器锁定<br>重置类型：CPU1.SYSRSN |
| 0 | INPUT1SELECT | R/WSonce | 0h | INPUT1选择寄存器的锁定位。<br>0:寄存器不锁定<br>1:寄存器锁定<br>重置类型：CPU1.SYSRSN |

### 8.4.3 XBAR_REGS 寄存器

表中未列出的寄存器偏移地址均为保留位置，不应修改这些位置的寄存器内容。

表 8-45 XBAR_REGS 寄存器

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | XBARFLG1 | X-Bar输入状态寄存器1 |
| 4h | XBARFLG2 | X-Bar输入状态寄存器2 |
| 8h | XBARFLG3 | X-Bar输入状态寄存器3 |
| Ch | XBARFLG4 | X-Bar输入状态寄存器4 |
| 10h | XBARCLR1 | X-Bar输入状态清除寄存器1 |
| 14h | XBARCLR2 | X-Bar输入状态清除寄存器2 |
| 18h | XBARCLR3 | X-Bar输入状态清除寄存器3 |
| 1Ch | XBARCLR4 | X-Bar输入状态清除寄存器4 |

复杂的位访问类型采用编码形式。下表列出了本节使用的访问类型编码。

表 8-46 XBAR_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W1S | W<br>1S | 写<br>1置位 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i、j、k、l、m、n | | 用于寄存器名称、偏移地址或地址时，表示属于重复寄存器组的寄存器数组值。寄存器组按层次组织，数组用公式表示。 |

| 访问类型 | 编码 | 说明 |
| --- | --- | --- |
| y | | 用于寄存器名称、偏移地址或地址时，表示寄存器数组值。 |

### 8.4.3.1 XBARFLG1 寄存器

偏移地址：0h　复位值：00000000h

X-BAR 输入标志寄存器 1。

表 8-47 XBARFLG1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CMPSS4_CTRIPOUTH | CMPSS4_CTRIPOUTL | CMPSS3_CTRIPOUTH | CMPSS3_CTRIPOUTL | CMPSS2_CTRIPOUTH | CMPSS2_CTRIPOUTL | CMPSS1_CTRIPOUTH | CMPSS1_CTRIPOUTL |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMPSS4_CTRIPH | CMPSS4_CTRIPL | CMPSS3_CTRIPH | CMPSS3_CTRIPL | CMPSS2_CTRIPH | CMPSS2_CTRIPL | CMPSS1_CTRIPH | CMPSS1_CTRIPL |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 8-48 XBARFLG1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 31-24 | 保留位 | R | 0h | 保留 |
| 23 | CMPSS4_CTRIPOUTH | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 22 | CMPSS4_CTRIPOUTL | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 21 | CMPSS3_CTRIPOUTH | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 20 | CMPSS3_CTRIPOUTL | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 19 | CMPSS2_CTRIPOUTH | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 18 | CMPSS2_CTRIPOUTL | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 17 | CMPSS1_CTRIPOUTH | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 16 | CMPSS1_CTRIPOUTL | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 15-8 | 保留位 | R | 0h | 保留 |
| 7 | CMPSS4_CTRIPH | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 6 | CMPSS4_CTRIPL | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 5 | CMPSS3_CTRIPH | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 4 | CMPSS3_CTRIPL | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 3 | CMPSS2_CTRIPH | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 2 | CMPSS2_CTRIPL | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 1 | CMPSS1_CTRIPH | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 0 | CMPSS1_CTRIPL | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |

### 8.4.3.2 XBARFLG2 寄存器

偏移地址：4h　复位值：00000000h

X-BAR 输入标志寄存器 2。

表 8-49 XBARFLG2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ADCCEVT1 | 保留 | | | | ADCAEVT4 | ADCAEVT3 | ADCAEVT2 |
| R-0h | R-0h | | | | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADCAEVT1 | EXTSYNCOUT | 保留 | | | ECAP3_OUT | ECAP2_OUT | ECAP1_OUT |
| R-0h | R-0h | R-0h | | | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INPUT14 | INPUT13 | INPUT12 | INPUT11 | INPUT10 | INPUT9 | INPUT8 | INPUT7 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADCSOCB | ADCSOCA | INPUT6 | INPUT5 | INPUT4 | INPUT3 | INPUT2 | INPUT1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 8-50 XBARFLG2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | ADCCEVT1 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 30-27 | 保留位 | R | 0h | 保留 |
| 26 | ADCAEVT4 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 25 | ADCAEVT3 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 24 | ADCAEVT2 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 23 | ADCAEVT1 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 22 | EXTSYNCOUT | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 21-19 | 保留位 | R | 0h | 保留 |
| 18 | ECAP3_OUT | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 17 | ECAP2_OUT | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 16 | ECAP1_OUT | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 15 | INPUT14 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 14 | INPUT13 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 13 | INPUT12 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 12 | INPUT11 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 11 | INPUT10 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 10 | INPUT9 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 9 | INPUT8 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 8 | INPUT7 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 7 | ADCSOCB | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 6 | ADCSOCA | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 5 | INPUT6 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 4 | INPUT5 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 3 | INPUT4 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 2 | INPUT3 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 1 | INPUT2 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 0 | INPUT1 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |

### 8.4.3.3 XBARFLG3 寄存器

偏移地址：8h　复位值：00000000h

X-BAR 输入标志寄存器 3。

表 8-51 XBARFLG3 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | ADCCEVT4 | ADCCEVT3 | ADCCEVT2 |
| R-0h | | | | | R-0h | R-0h | R-0h |

表 8-52 XBARFLG3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-3 | 保留位 | R | 0h | 保留 |
| 2 | ADCCEVT4 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 1 | ADCCEVT3 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 重置类型：CPU1.SYSRSN |
| 0 | ADCCEVT2 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |

### 8.4.3.4 XBARFLG4 寄存器

偏移地址：ch　复位值：00000000h

X-BAR 输入标志寄存器 4。

表 8-53 XBARFLG4 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | CLB2_5_1 | CLB2_4_1 | CLB1_5_1 | CLB1_4_1 |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |

表 8-54 XBARFLG4 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | 保留位 | R | 0h | 保留 |
| 19 | CLB2_5_1 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 18 | CLB2_4_1 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 17 | CLB1_5_1 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 16 | CLB1_4_1 | R | 0h | 该寄存器用于标记XBAR的输入，以便让软件知道哪些输入源被触发。<br>1：对应输入被触发<br>0：对应输入未被触发<br>注意：此位的设置优先于软件清除<br>重置类型：CPU1.SYSRSN |
| 15-0 | 保留位 | R | 0h | 保留 |

### 8.4.3.5 XBARCLR1 寄存器

偏移地址：10h　复位值：00000000h

X-BAR 输入标志清除寄存器 1。

表 8-55 XBARCLR1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0/W1S-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CMPSS4_CTRIPOUTH | CMPSS4_CTRIPOUTL | CMPSS3_CTRIPOUTH | CMPSS3_CTRIPOUTL | CMPSS2_CTRIPOUTH | CMPSS2_CTRIPOUTL | CMPSS1_CTRIPOUTH | CMPSS1_CTRIPOUTL |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0/W1S-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMPSS4_CTRIPH | CMPSS4_CTRIPL | CMPSS3_CTRIPH | CMPSS3_CTRIPL | CMPSS2_CTRIPH | CMPSS2_CTRIPL | CMPSS1_CTRIPH | CMPSS1_CTRIPL |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 8-56 XBARCLR1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-24 | 保留位 | R-0/W1S | 0h | 保留 |
| 23 | CMPSS4_CTRIPOUTH | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 22 | CMPSS4_CTRIPOUTL | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 21 | CMPSS3_CTRIPOUTH | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 20 | CMPSS3_CTRIPOUTL | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 19 | CMPSS2_CTRIPOUTH | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 18 | CMPSS2_CTRIPOUTL | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 17 | CMPSS1_CTRIPOUTH | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 16 | CMPSS1_CTRIPOUTL | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 15-8 | 保留位 | R-0/W1S | 0h | 保留 |
| 7 | CMPSS4_CTRIPH | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 6 | CMPSS4_CTRIPL | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 5 | CMPSS3_CTRIPH | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 4 | CMPSS3_CTRIPL | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 3 | CMPSS2_CTRIPH | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 2 | CMPSS2_CTRIPL | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 1 | CMPSS1_CTRIPH | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 0 | CMPSS1_CTRIPL | R-0/W1S | 0h | 置1可清除XBARFLG1寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |

### 8.4.3.6 XBARCLR2 寄存器

偏移地址：14h　复位值：00000000h

X-BAR 输入标志清除寄存器 2。

表 8-57 XBARCLR2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ADCCEVT1 | 保留 | | | | ADCAEVT4 | ADCAEVT3 | ADCAEVT2 |
| R-0/W1S-0h | R-0/W1S-0h | | | | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADCAEVT1 | EXTSYNCOUT | 保留 | | | ECAP3_OUT | ECAP2_OUT | ECAP1_OUT |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | | | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INPUT14 | INPUT13 | INPUT12 | INPUT11 | INPUT10 | INPUT9 | INPUT8 | INPUT7 |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADCSOCB | ADCSOCA | INPUT6 | INPUT5 | INPUT4 | INPUT3 | INPUT2 | INPUT1 |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 8-58 XBARCLR2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | ADCCEVT1 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 30-27 | 保留位 | R-0/W1S | 0h | 保留 |
| 26 | ADCAEVT4 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 25 | ADCAEVT3 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 24 | ADCAEVT2 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 23 | ADCAEVT1 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 22 | EXTSYNCOUT | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 21-19 | 保留位 | R-0/W1S | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 18 | ECAP3_OUT | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 17 | ECAP2_OUT | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 16 | ECAP1_OUT | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 15 | INPUT14 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 14 | INPUT13 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 13 | INPUT12 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 12 | INPUT11 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 11 | INPUT10 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 10 | INPUT9 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 9 | INPUT8 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 8 | INPUT7 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 7 | ADCSOCB | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 6 | ADCSOCA | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 5 | INPUT6 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 4 | INPUT5 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 3 | INPUT4 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 2 | INPUT3 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 1 | INPUT2 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 0 | INPUT1 | R-0/W1S | 0h | 置1可清除XBARFLG2寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |

### 8.4.3.7 XBARCLR3 寄存器

偏移地址：18h　复位值：00000000h

X-BAR 输入标志清除寄存器 3。

表 8-59 XBARCLR3 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0/W1S-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0/W1S-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0/W1S-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | ADCCEVT4 | ADCCEVT3 | ADCCEVT2 |
| R-0/W1S-0h | | | | | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 8-60 XBARCLR3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-3 | 保留位 | R-0/W1S | 0h | 保留 |
| 2 | ADCCEVT4 | R-0/W1S | 0h | 置1可清除XBARFLG3寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 1 | ADCCEVT3 | R-0/W1S | 0h | 置1可清除XBARFLG3寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 0 | ADCCEVT2 | R-0/W1S | 0h | 置1可清除XBARFLG3寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |

### 8.4.3.8 XBARCLR4 寄存器

偏移地址：1ch　复位值：00000000h

X-BAR 输入标志清除寄存器 4。

表 8-61 XBARCLR4 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0/W1S-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | CLB2_5_1 | CLB2_4_1 | CLB1_5_1 | CLB1_4_1 |
| R-0/W1S-0h | | | | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0/W1S-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R-0/W1S-0h | | | | | | | |

表 8-62 XBARCLR4 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | 保留位 | R-0/W1S | 0h | 保留 |
| 19 | CLB2_5_1 | R-0/W1S | 0h | 置1可清除XBARFLG4寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 18 | CLB2_4_1 | R-0/W1S | 0h | 置1可清除XBARFLG4寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 17 | CLB1_5_1 | R-0/W1S | 0h | 置1可清除XBARFLG4寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 16 | CLB1_4_1 | R-0/W1S | 0h | 置1可清除XBARFLG4寄存器中的对应位。<br>重置类型：CPU1.SYSRSN |
| 15-0 | 保留位 | R-0/W1S | 0h | 保留 |

## 8.4.4 EPWM_XBAR_REGS 寄存器

表中未列出的寄存器偏移地址均为保留位置，不应修改这些位置的寄存器内容。

表 8-63 EPWM_XBAR_REGS 寄存器

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | TRIP4MUX0TO15CFG | TRIP4 的ePWM 交叉开关多路选择器配置 |
| 4h | TRIP4MUX16TO31CFG | TRIP4 的ePWM 交叉开关多路选择器配置 |
| 8h | TRIP5MUX0TO15CFG | TRIP5 的ePWM 交叉开关多路选择器配置 |
| Ch | TRIP5MUX16TO31CFG | TRIP5 的ePWM 交叉开关多路选择器配置 |
| 10h | TRIP7MUX0TO15CFG | TRIP7 的ePWM 交叉开关多路选择器配置 |

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 14h | TRIP7MUX16TO31CFG | TRIP7 的ePWM 交叉开关多路选择器配置 |
| 18h | TRIP8MUX0TO15CFG | TRIP8 的ePWM 交叉开关多路选择器配置 |
| 1Ch | TRIP8MUX16TO31CFG | TRIP8 的ePWM 交叉开关多路选择器配置 |
| 20h | TRIP9MUX0TO15CFG | TRIP9 的ePWM 交叉开关多路选择器配置 |
| 24h | TRIP9MUX16TO31CFG | TRIP9 的ePWM 交叉开关多路选择器配置 |
| 28h | TRIP10MUX0TO15CFG | TRIP10 的ePWM 交叉开关多路选择器配置 |
| 2Ch | TRIP10MUX16TO31CFG | TRIP10 的ePWM 交叉开关多路选择器配置 |
| 30h | TRIP11MUX0TO15CFG | TRIP11 的ePWM 交叉开关多路选择器配置 |
| 34h | TRIP11MUX16TO31CFG | TRIP11 的ePWM 交叉开关多路选择器配置 |
| 38h | TRIP12MUX0TO15CFG | TRIP12 的ePWM 交叉开关多路选择器配置 |
| 3Ch | TRIP12MUX16TO31CFG | TRIP12 的ePWM 交叉开关多路选择器配置 |
| 40h | TRIP4MUXENABLE | TRIP4 的ePWM 交叉开关多路选择器使能 |
| 44h | TRIP5MUXENABLE | TRIP5 的ePWM 交叉开关多路选择器使能 |
| 48h | TRIP7MUXENABLE | TRIP7 的ePWM 交叉开关多路选择器使能 |
| 4Ch | TRIP8MUXENABLE | TRIP8 的ePWM 交叉开关多路选择器使能 |
| 50h | TRIP9MUXENABLE | TRIP9 的ePWM 交叉开关多路选择器使能 |
| 54h | TRIP10MUXENABLE | TRIP10 的ePWM 交叉开关多路选择器使能 |
| 58h | TRIP11MUXENABLE | TRIP11 的ePWM 交叉开关多路选择器使能 |
| 5Ch | TRIP12MUXENABLE | TRIP12 的ePWM 交叉开关多路选择器使能 |
| 70h | TRIPOUTINV | ePWM 交叉开关输出反相 |
| 7Ch | TRIPLOCK | ePWM 交叉开关配置锁定 |

复杂的位访问类型采用编码形式。下表列出了本节使用的访问类型编码。

表 8-64 EPWM_XBAR_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |
| WSonce | W<br>Sonce | 写<br>仅可置位一次 |
| 复位值或默认值 | | |

| 访问类型 | 编码 | 说明 |
| --- | --- | --- |
| -n | | 复位后的值或默认值 |

### 8.4.4.1 TRIP4MUX0TO15CFG 寄存器

偏移地址：0h　复位值：00000000h

ePWM X-BAR 的 TRIP4 多路选择器配置。

表 8-65 TRIP4MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-66 TRIP4MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 31-30 | MUX15 | R/W | 0h | EPWM-XBAR TRIP4 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | EPWM-XBAR TRIP4 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | EPWM-XBAR TRIP4 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 25-24 | MUX12 | R/W | 0h | EPWM-XBAR TRIP4 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | EPWM-XBAR TRIP4 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入<br>10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | EPWM-XBAR TRIP4 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | EPWM-XBAR TRIP4 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | EPWM-XBAR TRIP4 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | EPWM-XBAR TRIP4 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | EPWM-XBAR TRIP4 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | EPWM-XBAR TRIP4 MUX5的选择位： |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | EPWM-XBAR TRIP4 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入<br>11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | EPWM-XBAR TRIP4 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | EPWM-XBAR TRIP4 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | EPWM-XBAR TRIP4 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | EPWM-XBAR TRIP4 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.2 TRIP4MUX16TO31CFG 寄存器

偏移地址：4h　复位值：00000000h

ePWM X-BAR 的 TRIP4 多路选择器配置。

表 8-67 TRIP4MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-68 TRIP4MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | EPWM-XBAR TRIP4 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | EPWM-XBAR TRIP4 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | EPWM-XBAR TRIP4 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | EPWM-XBAR TRIP4 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | EPWM-XBAR TRIP4 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | EPWM-XBAR TRIP4 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | EPWM-XBAR TRIP4 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | EPWM-XBAR TRIP4 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | EPWM-XBAR TRIP4 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | EPWM-XBAR TRIP4 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | EPWM-XBAR TRIP4 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | EPWM-XBAR TRIP4 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | EPWM-XBAR TRIP4 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | EPWM-XBAR TRIP4 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | EPWM-XBAR TRIP4 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | EPWM-XBAR TRIP4 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.3 TRIP5MUX0TO15CFG 寄存器

偏移地址：8h　复位值：00000000h

ePWM X-BAR 的 TRIP5 多路选择器配置。

表 8-69 TRIP5MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-70 TRIP5MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | EPWM-XBAR TRIP5 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | EPWM-XBAR TRIP5 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | EPWM-XBAR TRIP5 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | EPWM-XBAR TRIP5 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | EPWM-XBAR TRIP5 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | EPWM-XBAR TRIP5 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | EPWM-XBAR TRIP5 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | EPWM-XBAR TRIP5 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | EPWM-XBAR TRIP5 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | EPWM-XBAR TRIP5 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | EPWM-XBAR TRIP5 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | EPWM-XBAR TRIP5 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | EPWM-XBAR TRIP5 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | EPWM-XBAR TRIP5 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | EPWM-XBAR TRIP5 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | EPWM-XBAR TRIP5 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.4 TRIP5MUX16TO31CFG 寄存器

偏移地址：ch　复位值：00000000h

ePWM X-BAR 的 TRIP5 多路选择器配置。

表 8-71 TRIP5MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-72 TRIP5MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | EPWM-XBAR TRIP5 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | EPWM-XBAR TRIP5 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | EPWM-XBAR TRIP5 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | EPWM-XBAR TRIP5 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | EPWM-XBAR TRIP5 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | EPWM-XBAR TRIP5 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | EPWM-XBAR TRIP5 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | EPWM-XBAR TRIP5 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | EPWM-XBAR TRIP5 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | EPWM-XBAR TRIP5 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | EPWM-XBAR TRIP5 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | EPWM-XBAR TRIP5 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | EPWM-XBAR TRIP5 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | EPWM-XBAR TRIP5 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | EPWM-XBAR TRIP5 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | EPWM-XBAR TRIP5 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.5 TRIP7MUX0TO15CFG 寄存器

偏移地址：10h　复位值：00000000h

ePWM X-BAR 的 TRIP7 多路选择器配置。

表 8-73 TRIP7MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-74 TRIP7MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | EPWM-XBAR TRIP7 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | EPWM-XBAR TRIP7 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | EPWM-XBAR TRIP7 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | EPWM-XBAR TRIP7 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | EPWM-XBAR TRIP7 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | EPWM-XBAR TRIP7 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | EPWM-XBAR TRIP7 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | EPWM-XBAR TRIP7 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | EPWM-XBAR TRIP7 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | EPWM-XBAR TRIP7 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | EPWM-XBAR TRIP7 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | EPWM-XBAR TRIP7 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | EPWM-XBAR TRIP7 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | EPWM-XBAR TRIP7 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | EPWM-XBAR TRIP7 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | EPWM-XBAR TRIP7 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.6 TRIP7MUX16TO31CFG 寄存器

偏移地址：14h　复位值：00000000h

ePWM X-BAR 的 TRIP7 多路选择器配置。

表 8-75 TRIP7MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-76 TRIP7MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | EPWM-XBAR TRIP7 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | EPWM-XBAR TRIP7 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | EPWM-XBAR TRIP7 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | EPWM-XBAR TRIP7 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | EPWM-XBAR TRIP7 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | EPWM-XBAR TRIP7 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | EPWM-XBAR TRIP7 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | EPWM-XBAR TRIP7 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | EPWM-XBAR TRIP7 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | EPWM-XBAR TRIP7 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | EPWM-XBAR TRIP7 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | EPWM-XBAR TRIP7 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | EPWM-XBAR TRIP7 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | EPWM-XBAR TRIP7 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | EPWM-XBAR TRIP7 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | EPWM-XBAR TRIP7 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.7 TRIP8MUX0TO15CFG 寄存器

偏移地址：18h　复位值：00000000h

ePWM X-BAR 的 TRIP8 多路选择器配置。

表 8-77 TRIP8MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-78 TRIP8MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | EPWM-XBAR TRIP8 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | EPWM-XBAR TRIP8 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | EPWM-XBAR TRIP8 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | EPWM-XBAR TRIP8 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | EPWM-XBAR TRIP8 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | EPWM-XBAR TRIP8 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | EPWM-XBAR TRIP8 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | EPWM-XBAR TRIP8 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | EPWM-XBAR TRIP8 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | EPWM-XBAR TRIP8 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | EPWM-XBAR TRIP8 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | EPWM-XBAR TRIP8 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | EPWM-XBAR TRIP8 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | EPWM-XBAR TRIP8 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | EPWM-XBAR TRIP8 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | EPWM-XBAR TRIP8 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.8 TRIP8MUX16TO31CFG 寄存器

偏移地址：1ch　复位值：00000000h

ePWM X-BAR 的 TRIP8 多路选择器配置。

表 8-79 TRIP8MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-80 TRIP8MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | EPWM-XBAR TRIP8 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | EPWM-XBAR TRIP8 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | EPWM-XBAR TRIP8 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | EPWM-XBAR TRIP8 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | EPWM-XBAR TRIP8 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | EPWM-XBAR TRIP8 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | EPWM-XBAR TRIP8 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | EPWM-XBAR TRIP8 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | EPWM-XBAR TRIP8 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | EPWM-XBAR TRIP8 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | EPWM-XBAR TRIP8 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | EPWM-XBAR TRIP8 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | EPWM-XBAR TRIP8 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | EPWM-XBAR TRIP8 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | EPWM-XBAR TRIP8 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | EPWM-XBAR TRIP8 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.9 TRIP9MUX0TO15CFG 寄存器

偏移地址：20h　复位值：00000000h

ePWM X-BAR 的 TRIP9 多路选择器配置。

表 8-81 TRIP9MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-82 TRIP9MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | EPWM-XBAR TRIP9 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | EPWM-XBAR TRIP9 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | EPWM-XBAR TRIP9 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | EPWM-XBAR TRIP9 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | EPWM-XBAR TRIP9 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | EPWM-XBAR TRIP9 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | EPWM-XBAR TRIP9 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | EPWM-XBAR TRIP9 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | EPWM-XBAR TRIP9 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | EPWM-XBAR TRIP9 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | EPWM-XBAR TRIP9 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | EPWM-XBAR TRIP9 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | EPWM-XBAR TRIP9 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | EPWM-XBAR TRIP9 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | EPWM-XBAR TRIP9 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | EPWM-XBAR TRIP9 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.10 TRIP9MUX16TO31CFG 寄存器

偏移地址：24h　复位值：00000000h

ePWM X-BAR 的 TRIP9 多路选择器配置。

表 8-83 TRIP9MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-84 TRIP9MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | EPWM-XBAR TRIP9 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | EPWM-XBAR TRIP9 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | EPWM-XBAR TRIP9 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | EPWM-XBAR TRIP9 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | EPWM-XBAR TRIP9 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | EPWM-XBAR TRIP9 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | EPWM-XBAR TRIP9 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | EPWM-XBAR TRIP9 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | EPWM-XBAR TRIP9 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | EPWM-XBAR TRIP9 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | EPWM-XBAR TRIP9 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | EPWM-XBAR TRIP9 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | EPWM-XBAR TRIP9 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | EPWM-XBAR TRIP9 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | EPWM-XBAR TRIP9 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | EPWM-XBAR TRIP9 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.11 TRIP10MUX0TO15CFG 寄存器

偏移地址：28h　复位值：00000000h

ePWM X-BAR 的 TRIP10 多路选择器配置。

表 8-85 TRIP10MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-86 TRIP10MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | EPWM-XBAR TRIP10 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | EPWM-XBAR TRIP10 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | EPWM-XBAR TRIP10 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | EPWM-XBAR TRIP10 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | EPWM-XBAR TRIP10 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | EPWM-XBAR TRIP10 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | EPWM-XBAR TRIP10 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | EPWM-XBAR TRIP10 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | EPWM-XBAR TRIP10 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | EPWM-XBAR TRIP10 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | EPWM-XBAR TRIP10 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | EPWM-XBAR TRIP10 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | EPWM-XBAR TRIP10 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | EPWM-XBAR TRIP10 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | EPWM-XBAR TRIP10 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | EPWM-XBAR TRIP10 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.12 TRIP10MUX16TO31CFG 寄存器

偏移地址：2ch　复位值：00000000h

ePWM X-BAR 的 TRIP10 多路选择器配置。

表 8-87 TRIP10MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-88 TRIP10MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | EPWM-XBAR TRIP10 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | EPWM-XBAR TRIP10 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | EPWM-XBAR TRIP10 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | EPWM-XBAR TRIP10 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | EPWM-XBAR TRIP10 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | EPWM-XBAR TRIP10 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | EPWM-XBAR TRIP10 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | EPWM-XBAR TRIP10 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | EPWM-XBAR TRIP10 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | EPWM-XBAR TRIP10 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | EPWM-XBAR TRIP10 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | EPWM-XBAR TRIP10 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | EPWM-XBAR TRIP10 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | EPWM-XBAR TRIP10 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | EPWM-XBAR TRIP10 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | EPWM-XBAR TRIP10 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.13 TRIP11MUX0TO15CFG 寄存器

偏移地址：30h　复位值：00000000h

ePWM X-BAR 的 TRIP11 多路选择器配置。

表 8-89 TRIP11MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-90 TRIP11MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | EPWM-XBAR TRIP11 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | EPWM-XBAR TRIP11 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | EPWM-XBAR TRIP11 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | EPWM-XBAR TRIP11 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | EPWM-XBAR TRIP11 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | EPWM-XBAR TRIP11 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | EPWM-XBAR TRIP11 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | EPWM-XBAR TRIP11 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | EPWM-XBAR TRIP11 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | EPWM-XBAR TRIP11 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | EPWM-XBAR TRIP11 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | EPWM-XBAR TRIP11 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | EPWM-XBAR TRIP11 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | EPWM-XBAR TRIP11 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | EPWM-XBAR TRIP11 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | EPWM-XBAR TRIP11 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.14 TRIP11MUX16TO31CFG 寄存器

偏移地址：34h　复位值：00000000h

ePWM X-BAR 的 TRIP11 多路选择器配置。

表 8-91 TRIP11MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-92 TRIP11MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | EPWM-XBAR TRIP11 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | EPWM-XBAR TRIP11 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | EPWM-XBAR TRIP11 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | EPWM-XBAR TRIP11 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | EPWM-XBAR TRIP11 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | EPWM-XBAR TRIP11 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | EPWM-XBAR TRIP11 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | EPWM-XBAR TRIP11 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | EPWM-XBAR TRIP11 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | EPWM-XBAR TRIP11 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | EPWM-XBAR TRIP11 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | EPWM-XBAR TRIP11 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | EPWM-XBAR TRIP11 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | EPWM-XBAR TRIP11 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | EPWM-XBAR TRIP11 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | EPWM-XBAR TRIP11 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.15 TRIP12MUX0TO15CFG 寄存器

偏移地址：38h　复位值：00000000h

ePWM X-BAR 的 TRIP12 多路选择器配置。

表 8-93 TRIP12MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-94 TRIP12MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | EPWM-XBAR TRIP12 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | EPWM-XBAR TRIP12 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | EPWM-XBAR TRIP12 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | EPWM-XBAR TRIP12 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | EPWM-XBAR TRIP12 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | EPWM-XBAR TRIP12 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | EPWM-XBAR TRIP12 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | EPWM-XBAR TRIP12 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | EPWM-XBAR TRIP12 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | EPWM-XBAR TRIP12 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | EPWM-XBAR TRIP12 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | EPWM-XBAR TRIP12 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | EPWM-XBAR TRIP12 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | EPWM-XBAR TRIP12 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | EPWM-XBAR TRIP12 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | EPWM-XBAR TRIP12 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.16 TRIP12MUX16TO31CFG 寄存器

偏移地址：3ch　复位值：00000000h

ePWM X-BAR 的 TRIP12 多路选择器配置。

表 8-95 TRIP12MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-96 TRIP12MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | EPWM-XBAR TRIP12 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | EPWM-XBAR TRIP12 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | EPWM-XBAR TRIP12 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | EPWM-XBAR TRIP12 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | EPWM-XBAR TRIP12 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | EPWM-XBAR TRIP12 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | EPWM-XBAR TRIP12 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | EPWM-XBAR TRIP12 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | EPWM-XBAR TRIP12 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | EPWM-XBAR TRIP12 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | EPWM-XBAR TRIP12 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | EPWM-XBAR TRIP12 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | EPWM-XBAR TRIP12 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | EPWM-XBAR TRIP12 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | EPWM-XBAR TRIP12 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | EPWM-XBAR TRIP12 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.4.17 TRIP4MUXENABLE 寄存器

偏移地址：40h　复位值：00000000h

ePWM X-BAR 的 TRIP4 多路选择器使能。

表 8-97 TRIP4MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-98 TRIP4MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动EPWM-XBAR的TRIP4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.4.18 TRIP5MUXENABLE 寄存器

偏移地址：44h　复位值：00000000h

ePWM X-BAR 的 TRIP5 多路选择器使能。

表 8-99 TRIP5MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-100 TRIP5MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动EPWM-XBAR的TRIP5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.4.19 TRIP7MUXENABLE 寄存器

偏移地址：48h　复位值：00000000h

ePWM X-BAR 的 TRIP7 多路选择器使能。

表 8-101 TRIP7MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-102 TRIP7MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动EPWM-XBAR的TRIP7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.4.20 TRIP8MUXENABLE 寄存器

偏移地址：4ch　复位值：00000000h

ePWM X-BAR 的 TRIP8 多路选择器使能。

表 8-103 TRIP8MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-104 TRIP8MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动EPWM-XBAR的TRIP8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.4.21 TRIP9MUXENABLE 寄存器

偏移地址：50h　复位值：00000000h

ePWM X-BAR 的 TRIP9 多路选择器使能。

表 8-105 TRIP9MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-106 TRIP9MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动EPWM-XBAR的TRIP9<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.4.22 TRIP10MUXENABLE 寄存器

偏移地址：54h　复位值：00000000h

ePWM X-BAR 的 TRIP10 多路选择器使能。

表 8-107 TRIP10MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-108 TRIP10MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动EPWM-XBAR的TRIP10<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.4.23 TRIP11MUXENABLE 寄存器

偏移地址：58h　复位值：00000000h

ePWM X-BAR 的 TRIP11 多路选择器使能。

表 8-109 TRIP11MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-110 TRIP11MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动EPWM-XBAR的TRIP11<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.4.24 TRIP12MUXENABLE 寄存器

偏移地址：5ch　复位值：00000000h

ePWM X-BAR 的 TRIP12 多路选择器使能。

表 8-111 TRIP12MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-112 TRIP12MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动EPWM-XBAR的TRIP12<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.4.25 TRIPOUTINV 寄存器

偏移地址：70h　复位值：00000000h

ePWM X-BAR 输出反相寄存器。

表 8-113 TRIPOUTINV 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TRIP12 | TRIP11 | TRIP10 | TRIP9 | TRIP8 | TRIP7 | TRIP5 | TRIP4 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-114 TRIPOUTINV 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-8 | 保留位 | R-0 | 0h | 保留 |
| 7 | TRIP12 | R/W | 0h | 为EPWM-XBAR的TRIP12选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 6 | TRIP11 | R/W | 0h | 为EPWM-XBAR的TRIP11选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 5 | TRIP10 | R/W | 0h | 为EPWM-XBAR的TRIP10选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 4 | TRIP9 | R/W | 0h | 为EPWM-XBAR的TRIP9选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 3 | TRIP8 | R/W | 0h | 为EPWM-XBAR的TRIP8选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 2 | TRIP7 | R/W | 0h | 为EPWM-XBAR的TRIP7选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 1 | TRIP5 | R/W | 0h | 为EPWM-XBAR的TRIP5选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 0 | TRIP4 | R/W | 0h | 为EPWM-XBAR的TRIP4选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |

### 8.4.4.26 TRIPLOCK 寄存器

偏移地址：64h　复位值：00000000h

ePWM X-BAR 配置锁定寄存器。

表 8-115 TRIPLOCK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| KEY | | | | | | | |
| R-0W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | LOCK |
| R-0-0h | | | | | | | R/WSonce-0h |

表 8-116 TRIPLOCK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | KEY | R-0W | 0h | KEY=0x5a5a时才能设置零比特位 |
| 15-1 | 保留位 | R-0 | 0h | 保留<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 0 | LOCK | R/WSonce | 0h | 锁定EPWM-XBAR的配置。一旦配置被锁定，对EPWM-XBAR的以下寄存器的写入被阻止。受锁定机制影响的寄存器：EPWM-XBAROUTyMUX0TO15CFG<br>EPWM-XBAROUTyMUX16TO31CFG<br>EPWM-XBAROUTyMUXENABLE<br>EPWM-XBAROUTLATEN<br>EPWM-XBAROUTINV<br>0：允许写入上述寄存器<br>1：不允许写入上述寄存器<br>重置类型：CPU1.SYSRSN |

## 8.4.5 CLB_XBAR_REGS 寄存器

表中未列出的寄存器偏移地址均为保留位置，不应修改这些位置的寄存器内容。

表 8-117 CLB_XBAR_REGS 寄存器

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | AUXSIG0MUX0TO15CFG | 输出 0 的CLB 交叉开关多路选择器配置 |
| 4h | AUXSIG0MUX16TO31CFG | 输出 0 的CLB 交叉开关多路选择器配置 |
| 8h | AUXSIG1MUX0TO15CFG | 输出 1 的CLB 交叉开关多路选择器配置 |

| 偏移地址 | 缩写 | 寄存器名称 |
| --- | --- | --- |
| Ch | AUXSIG1MUX16TO31CFG | 输出 1 的CLB 交叉开关多路选择器配置 |
| 10h | AUXSIG2MUX0TO15CFG | 输出 2 的CLB 交叉开关多路选择器配置 |
| 14h | AUXSIG2MUX16TO31CFG | 输出 2 的CLB 交叉开关多路选择器配置 |
| 18h | AUXSIG3MUX0TO15CFG | 输出 3 的CLB 交叉开关多路选择器配置 |
| 1Ch | AUXSIG3MUX16TO31CFG | 输出 3 的CLB 交叉开关多路选择器配置 |
| 20h | AUXSIG4MUX0TO15CFG | 输出 4 的CLB 交叉开关多路选择器配置 |
| 24h | AUXSIG4MUX16TO31CFG | 输出 4 的CLB 交叉开关多路选择器配置 |
| 28h | AUXSIG5MUX0TO15CFG | 输出 5 的CLB 交叉开关多路选择器配置 |
| 2Ch | AUXSIG5MUX16TO31CFG | 输出 5 的CLB 交叉开关多路选择器配置 |
| 30h | AUXSIG6MUX0TO15CFG | 输出 6 的CLB 交叉开关多路选择器配置 |
| 34h | AUXSIG6MUX16TO31CFG | 输出 6 的CLB 交叉开关多路选择器配置 |
| 38h | AUXSIG7MUX0TO15CFG | 输出 7 的CLB 交叉开关多路选择器配置 |
| 3Ch | AUXSIG7MUX16TO31CFG | 输出 7 的CLB 交叉开关多路选择器配置 |
| 40h | AUXSIG0MUXENABLE | 输出 0 的CLB 交叉开关多路选择器使能 |
| 44h | AUXSIG1MUXENABLE | 输出 1 的CLB 交叉开关多路选择器使能 |
| 48h | AUXSIG2MUXENABLE | 输出 2 的CLB 交叉开关多路选择器使能 |
| 4Ch | AUXSIG3MUXENABLE | 输出 3 的CLB 交叉开关多路选择器使能 |
| 50h | AUXSIG4MUXENABLE | 输出 4 的CLB 交叉开关多路选择器使能 |
| 54h | AUXSIG5MUXENABLE | 输出 5 的CLB 交叉开关多路选择器使能 |
| 58h | AUXSIG6MUXENABLE | 输出 6 的CLB 交叉开关多路选择器使能 |
| 5Ch | AUXSIG7MUXENABLE | 输出 7 的CLB 交叉开关多路选择器使能 |
| 70h | AUXSIGOUTINV | CLB 交叉开关输出反相 |
| 7Ch | AUXSIGLOCK | CLB 交叉开关配置锁定 |

复杂的位访问类型采用编码形式。下表列出了本节使用的访问类型编码。

表 8-118 CLB_XBAR_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
| --- | --- | --- |
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |

| 访问类型 | 编码 | 说明 |
| --- | --- | --- |
| WSonce | W<br>Sonce | 写<br>仅可置位一次 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |

### 8.4.5.1 AUXSIG0MUX0TO15CFG 寄存器

偏移地址：0h　复位值：00000000h

CLB X-BAR 的 AUXSIG0 多路选择器配置。

表 8-119 AUXSIG0MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-120 AUXSIG0MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 31-30 | MUX15 | R/W | 0h | CLB-XBAR AUXSIG0 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | CLB-XBAR AUXSIG0 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | CLB-XBAR AUXSIG0 MUX13的选择位：<br>00：为Mux13选择0输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | CLB-XBAR AUXSIG0 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | CLB-XBAR AUXSIG0 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入<br>10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | CLB-XBAR AUXSIG0 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | CLB-XBAR AUXSIG0 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | CLB-XBAR AUXSIG0 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | CLB-XBAR AUXSIG0 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | CLB-XBAR AUXSIG0 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | CLB-XBAR AUXSIG0 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | CLB-XBAR AUXSIG0 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入<br>11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | CLB-XBAR AUXSIG0 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | CLB-XBAR AUXSIG0 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | CLB-XBAR AUXSIG0 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | CLB-XBAR AUXSIG0 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.2 AUXSIG0MUX16TO31CFG 寄存器

偏移地址：4h　复位值：00000000h

CLB X-BAR 的 AUXSIG0 多路选择器配置。

表 8-121 AUXSIG0MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-122 AUXSIG0MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | CLB-XBAR AUXSIG0 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | CLB-XBAR AUXSIG0 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | CLB-XBAR AUXSIG0 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | CLB-XBAR AUXSIG0 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | CLB-XBAR AUXSIG0 MUX27的选择位： |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 00：为Mux27选择0输入<br>01：为Mux27选择1输入<br>10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | CLB-XBAR AUXSIG0 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | CLB-XBAR AUXSIG0 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | CLB-XBAR AUXSIG0 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | CLB-XBAR AUXSIG0 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | CLB-XBAR AUXSIG0 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | CLB-XBAR AUXSIG0 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | CLB-XBAR AUXSIG0 MUX20的选择位：<br>00：为Mux20选择0输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 01：为Mux20选择1输入<br>10：为Mux20选择2输入<br>11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | CLB-XBAR AUXSIG0 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | CLB-XBAR AUXSIG0 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | CLB-XBAR AUXSIG0 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | CLB-XBAR AUXSIG0 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.3 AUXSIG1MUX0TO15CFG 寄存器

偏移地址：8h　复位值：00000000h

CLB X-BAR 的 AUXSIG1 多路选择器配置。

表 8-123 AUXSIG1MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-124 AUXSIG1MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | CLB-XBAR AUXSIG1 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | CLB-XBAR AUXSIG1 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | CLB-XBAR AUXSIG1 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | CLB-XBAR AUXSIG1 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | CLB-XBAR AUXSIG1 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | CLB-XBAR AUXSIG1 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | CLB-XBAR AUXSIG1 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | CLB-XBAR AUXSIG1 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | CLB-XBAR AUXSIG1 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | CLB-XBAR AUXSIG1 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | CLB-XBAR AUXSIG1 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | CLB-XBAR AUXSIG1 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | CLB-XBAR AUXSIG1 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | CLB-XBAR AUXSIG1 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | CLB-XBAR AUXSIG1 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | CLB-XBAR AUXSIG1 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.4 AUXSIG1MUX16TO31CFG 寄存器

偏移地址：ch　复位值：00000000h

CLB X-BAR 的 AUXSIG1 多路选择器配置。

表 8-125 AUXSIG1MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-126 AUXSIG1MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | CLB-XBAR AUXSIG1 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | CLB-XBAR AUXSIG1 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | CLB-XBAR AUXSIG1 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | CLB-XBAR AUXSIG1 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | CLB-XBAR AUXSIG1 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | CLB-XBAR AUXSIG1 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | CLB-XBAR AUXSIG1 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | CLB-XBAR AUXSIG1 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | CLB-XBAR AUXSIG1 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | CLB-XBAR AUXSIG1 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | CLB-XBAR AUXSIG1 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | CLB-XBAR AUXSIG1 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | CLB-XBAR AUXSIG1 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | CLB-XBAR AUXSIG1 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | CLB-XBAR AUXSIG1 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | CLB-XBAR AUXSIG1 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.5 AUXSIG2MUX0TO15CFG 寄存器

偏移地址：10h　复位值：00000000h

CLB X-BAR 的 AUXSIG2 多路选择器配置。

表 8-127 AUXSIG2MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-128 AUXSIG2MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | CLB-XBAR AUXSIG2 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | CLB-XBAR AUXSIG2 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | CLB-XBAR AUXSIG2 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | CLB-XBAR AUXSIG2 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | CLB-XBAR AUXSIG2 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | CLB-XBAR AUXSIG2 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | CLB-XBAR AUXSIG2 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | CLB-XBAR AUXSIG2 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | CLB-XBAR AUXSIG2 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | CLB-XBAR AUXSIG2 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | CLB-XBAR AUXSIG2 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | CLB-XBAR AUXSIG2 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | CLB-XBAR AUXSIG2 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | CLB-XBAR AUXSIG2 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | CLB-XBAR AUXSIG2 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | CLB-XBAR AUXSIG2 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.6 AUXSIG2MUX16TO31CFG 寄存器

偏移地址：14h　复位值：00000000h

CLB X-BAR 的 AUXSIG2 多路选择器配置。

表 8-129 AUXSIG2MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-130 AUXSIG2MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | CLB-XBAR AUXSIG2 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | CLB-XBAR AUXSIG2 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | CLB-XBAR AUXSIG2 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | CLB-XBAR AUXSIG2 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | CLB-XBAR AUXSIG2 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | CLB-XBAR AUXSIG2 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | CLB-XBAR AUXSIG2 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | CLB-XBAR AUXSIG2 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | CLB-XBAR AUXSIG2 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | CLB-XBAR AUXSIG2 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | CLB-XBAR AUXSIG2 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | CLB-XBAR AUXSIG2 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | CLB-XBAR AUXSIG2 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | CLB-XBAR AUXSIG2 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | CLB-XBAR AUXSIG2 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | CLB-XBAR AUXSIG2 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.7 AUXSIG3MUX0TO15CFG 寄存器

偏移地址：18h　复位值：00000000h

CLB X-BAR 的 AUXSIG3 多路选择器配置。

表 8-131 AUXSIG3MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-132 AUXSIG3MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | CLB-XBAR AUXSIG3 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | CLB-XBAR AUXSIG3 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | CLB-XBAR AUXSIG3 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | CLB-XBAR AUXSIG3 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | CLB-XBAR AUXSIG3 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | CLB-XBAR AUXSIG3 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | CLB-XBAR AUXSIG3 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | CLB-XBAR AUXSIG3 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | CLB-XBAR AUXSIG3 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | CLB-XBAR AUXSIG3 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | CLB-XBAR AUXSIG3 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | CLB-XBAR AUXSIG3 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | CLB-XBAR AUXSIG3 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | CLB-XBAR AUXSIG3 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | CLB-XBAR AUXSIG3 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | CLB-XBAR AUXSIG3 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.8 AUXSIG3MUX16TO31CFG 寄存器

偏移地址：1ch　复位值：00000000h

CLB X-BAR 的 AUXSIG3 多路选择器配置。

表 8-133 AUXSIG3MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-134 AUXSIG3MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | CLB-XBAR AUXSIG3 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | CLB-XBAR AUXSIG3 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | CLB-XBAR AUXSIG3 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | CLB-XBAR AUXSIG3 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | CLB-XBAR AUXSIG3 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | CLB-XBAR AUXSIG3 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | CLB-XBAR AUXSIG3 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | CLB-XBAR AUXSIG3 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | CLB-XBAR AUXSIG3 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | CLB-XBAR AUXSIG3 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | CLB-XBAR AUXSIG3 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | CLB-XBAR AUXSIG3 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | CLB-XBAR AUXSIG3 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | CLB-XBAR AUXSIG3 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | CLB-XBAR AUXSIG3 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | CLB-XBAR AUXSIG3 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.9 AUXSIG4MUX0TO15CFG 寄存器

偏移地址：20h　复位值：00000000h

CLB X-BAR 的 AUXSIG4 多路选择器配置。

表 8-135 AUXSIG4MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-136 AUXSIG4MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | CLB-XBAR AUXSIG4 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | CLB-XBAR AUXSIG4 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | CLB-XBAR AUXSIG4 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | CLB-XBAR AUXSIG4 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | CLB-XBAR AUXSIG4 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | CLB-XBAR AUXSIG4 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | CLB-XBAR AUXSIG4 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | CLB-XBAR AUXSIG4 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | CLB-XBAR AUXSIG4 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | CLB-XBAR AUXSIG4 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | CLB-XBAR AUXSIG4 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | CLB-XBAR AUXSIG4 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | CLB-XBAR AUXSIG4 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | CLB-XBAR AUXSIG4 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | CLB-XBAR AUXSIG4 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | CLB-XBAR AUXSIG4 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.10 AUXSIG4MUX16TO31CFG 寄存器

偏移地址：24h　复位值：00000000h

CLB X-BAR 的 AUXSIG4 多路选择器配置。

表 8-137 AUXSIG4MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-138 AUXSIG4MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | CLB-XBAR AUXSIG4 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | CLB-XBAR AUXSIG4 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | CLB-XBAR AUXSIG4 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | CLB-XBAR AUXSIG4 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | CLB-XBAR AUXSIG4 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | CLB-XBAR AUXSIG4 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | CLB-XBAR AUXSIG4 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | CLB-XBAR AUXSIG4 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | CLB-XBAR AUXSIG4 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | CLB-XBAR AUXSIG4 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | CLB-XBAR AUXSIG4 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | CLB-XBAR AUXSIG4 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | CLB-XBAR AUXSIG4 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | CLB-XBAR AUXSIG4 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | CLB-XBAR AUXSIG4 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | CLB-XBAR AUXSIG4 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.11 AUXSIG5MUX0TO15CFG 寄存器

偏移地址：28h　复位值：00000000h

CLB X-BAR 的 AUXSIG5 多路选择器配置。

表 8-139 AUXSIG5MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-140 AUXSIG5MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | CLB-XBAR AUXSIG5 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | CLB-XBAR AUXSIG5 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | CLB-XBAR AUXSIG5 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | CLB-XBAR AUXSIG5 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | CLB-XBAR AUXSIG5 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | CLB-XBAR AUXSIG5 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | CLB-XBAR AUXSIG5 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | CLB-XBAR AUXSIG5 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | CLB-XBAR AUXSIG5 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | CLB-XBAR AUXSIG5 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | CLB-XBAR AUXSIG5 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | CLB-XBAR AUXSIG5 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | CLB-XBAR AUXSIG5 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | CLB-XBAR AUXSIG5 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | CLB-XBAR AUXSIG5 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | CLB-XBAR AUXSIG5 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.12 AUXSIG5MUX16TO31CFG 寄存器

偏移地址：2ch　复位值：00000000h

CLB X-BAR 的 AUXSIG5 多路选择器配置。

表 8-141 AUXSIG5MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-142 AUXSIG5MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | CLB-XBAR AUXSIG5 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | CLB-XBAR AUXSIG5 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | CLB-XBAR AUXSIG5 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | CLB-XBAR AUXSIG5 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | CLB-XBAR AUXSIG5 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | CLB-XBAR AUXSIG5 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | CLB-XBAR AUXSIG5 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | CLB-XBAR AUXSIG5 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | CLB-XBAR AUXSIG5 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | CLB-XBAR AUXSIG5 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | CLB-XBAR AUXSIG5 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | CLB-XBAR AUXSIG5 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | CLB-XBAR AUXSIG5 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | CLB-XBAR AUXSIG5 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | CLB-XBAR AUXSIG5 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | CLB-XBAR AUXSIG5 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.13 AUXSIG6MUX0TO15CFG 寄存器

偏移地址：30h　复位值：00000000h

CLB X-BAR 的 AUXSIG6 多路选择器配置。

表 8-143 AUXSIG6MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-144 AUXSIG6MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | CLB-XBAR AUXSIG6 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | CLB-XBAR AUXSIG6 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | CLB-XBAR AUXSIG6 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | CLB-XBAR AUXSIG6 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | CLB-XBAR AUXSIG6 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | CLB-XBAR AUXSIG6 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | CLB-XBAR AUXSIG6 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | CLB-XBAR AUXSIG6 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | CLB-XBAR AUXSIG6 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | CLB-XBAR AUXSIG6 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | CLB-XBAR AUXSIG6 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | CLB-XBAR AUXSIG6 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | CLB-XBAR AUXSIG6 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | CLB-XBAR AUXSIG6 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | CLB-XBAR AUXSIG6 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | CLB-XBAR AUXSIG6 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.14 AUXSIG6MUX16TO31CFG 寄存器

偏移地址：34h　复位值：00000000h

CLB X-BAR 的 AUXSIG6 多路选择器配置。

表 8-145 AUXSIG6MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-146 AUXSIG6MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | CLB-XBAR AUXSIG6 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | CLB-XBAR AUXSIG6 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | CLB-XBAR AUXSIG6 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | CLB-XBAR AUXSIG6 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | CLB-XBAR AUXSIG6 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | CLB-XBAR AUXSIG6 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | CLB-XBAR AUXSIG6 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | CLB-XBAR AUXSIG6 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | CLB-XBAR AUXSIG6 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | CLB-XBAR AUXSIG6 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | CLB-XBAR AUXSIG6 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | CLB-XBAR AUXSIG6 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | CLB-XBAR AUXSIG6 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | CLB-XBAR AUXSIG6 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | CLB-XBAR AUXSIG6 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | CLB-XBAR AUXSIG6 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.15 AUXSIG7MUX0TO15CFG 寄存器

偏移地址：38h　复位值：00000000h

CLB X-BAR 的 AUXSIG7 多路选择器配置。

表 8-147 AUXSIG7MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-148 AUXSIG7MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | CLB-XBAR AUXSIG7 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | CLB-XBAR AUXSIG7 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | CLB-XBAR AUXSIG7 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | CLB-XBAR AUXSIG7 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | CLB-XBAR AUXSIG7 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | CLB-XBAR AUXSIG7 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | CLB-XBAR AUXSIG7 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | CLB-XBAR AUXSIG7 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | CLB-XBAR AUXSIG7 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | CLB-XBAR AUXSIG7 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | CLB-XBAR AUXSIG7 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | CLB-XBAR AUXSIG7 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | CLB-XBAR AUXSIG7 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | CLB-XBAR AUXSIG7 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | CLB-XBAR AUXSIG7 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | CLB-XBAR AUXSIG7 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.16 AUXSIG7MUX16TO31CFG 寄存器

偏移地址：3ch　复位值：00000000h

CLB X-BAR 的 AUXSIG7 多路选择器配置。

表 8-149 AUXSIG7MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-150 AUXSIG7MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | CLB-XBAR AUXSIG7 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | CLB-XBAR AUXSIG7 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | CLB-XBAR AUXSIG7 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | CLB-XBAR AUXSIG7 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | CLB-XBAR AUXSIG7 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | CLB-XBAR AUXSIG7 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | CLB-XBAR AUXSIG7 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | CLB-XBAR AUXSIG7 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | CLB-XBAR AUXSIG7 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | CLB-XBAR AUXSIG7 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | CLB-XBAR AUXSIG7 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | CLB-XBAR AUXSIG7 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | CLB-XBAR AUXSIG7 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | CLB-XBAR AUXSIG7 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | CLB-XBAR AUXSIG7 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | CLB-XBAR AUXSIG7 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.5.17 AUXSIG0MUXENABLE 寄存器

偏移地址：40h　复位值：00000000h

CLB X-BAR 的 AUXSIG0 多路选择器使能。

表 8-151 AUXSIG0MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-152 AUXSIG0MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动CLB-XBAR的AUXSIG0 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动CLB-XBAR的AUXSIG0<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.5.18 AUXSIG1MUXENABLE 寄存器

偏移地址：44h　复位值：00000000h

CLB X-BAR 的 AUXSIG1 多路选择器使能。

表 8-153 AUXSIG1MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-154 AUXSIG1MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动CLB-XBAR的AUXSIG1 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动CLB-XBAR的AUXSIG1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.5.19 AUXSIG2MUXENABLE 寄存器

偏移地址：48h　复位值：00000000h

CLB X-BAR 的 AUXSIG2 多路选择器使能。

表 8-155 AUXSIG2MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-156 AUXSIG2MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动CLB-XBAR的AUXSIG2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动CLB-XBAR的AUXSIG2 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.5.20 AUXSIG3MUXENABLE 寄存器

偏移地址：4ch　复位值：00000000h

CLB X-BAR 的 AUXSIG3 多路选择器使能。

表 8-157 AUXSIG3MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-158 AUXSIG3MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动CLB-XBAR的AUXSIG3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.5.21 AUXSIG4MUXENABLE 寄存器

偏移地址：50h　复位值：00000000h

CLB X-BAR 的 AUXSIG4 多路选择器使能。

表 8-159 AUXSIG4MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-160 AUXSIG4MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动CLB-XBAR的AUXSIG4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.5.22 AUXSIG5MUXENABLE 寄存器

偏移地址：54h　复位值：00000000h

CLB X-BAR 的 AUXSIG5 多路选择器使能。

表 8-161 AUXSIG5MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-162 AUXSIG5MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动CLB-XBAR的AUXSIG5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.5.23 AUXSIG6MUXENABLE 寄存器

偏移地址：58h　复位值：00000000h

CLB X-BAR 的 AUXSIG6 多路选择器使能。

表 8-163 AUXSIG6MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-164 AUXSIG6MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动CLB-XBAR的AUXSIG6 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动CLB-XBAR的AUXSIG6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.5.24 AUXSIG7MUXENABLE 寄存器

偏移地址：5ch　复位值：00000000h

CLB X-BAR 的 AUXSIG7 多路选择器使能。

表 8-165 AUXSIG7MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-166 AUXSIG7MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动CLB-XBAR的AUXSIG7 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动CLB-XBAR的AUXSIG7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.5.25 AUXSIGOUTINV 寄存器

偏移地址：70h　复位值：00000000h

CLB X-BAR 输出反相寄存器。

表 8-167 AUXSIGOUTINV 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUT7 | OUT6 | OUT5 | OUT4 | OUT3 | OUT2 | OUT1 | OUT0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-168 AUXSIGOUTINV 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-8 | 保留位 | R-0 | 0h | 保留 |
| 7 | OUT7 | R/W | 0h | 为CLB-XBAR的AUXSIG7选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 6 | OUT6 | R/W | 0h | 为CLB-XBAR的AUXSIG6选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 5 | OUT5 | R/W | 0h | 为CLB-XBAR的AUXSIG5选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 4 | OUT4 | R/W | 0h | 为CLB-XBAR的AUXSIG4选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 3 | OUT3 | R/W | 0h | 为CLB-XBAR的AUXSIG3选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 2 | OUT2 | R/W | 0h | 为CLB-XBAR的AUXSIG2选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 重置类型：CPU1.SYSRSN |
| 1 | OUT1 | R/W | 0h | 为CLB-XBAR的AUXSIG1选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 0 | OUT0 | R/W | 0h | 为CLB-XBAR的AUXSIG0选择极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |

## 8.4.5.26 AUXSIGLOCK 寄存器

偏移地址：7ch　复位值：00000000h

CLB X-BAR 配置锁定寄存器。

表 8-169 AUXSIGLOCK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| KEY | | | | | | | |
| R-0W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | LOCK |
| R-0-0h | | | | | | | R/WSonce-0h |

表 8-170 AUXSIGLOCK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | KEY | R-0W | 0h | KEY=0x5a5a时才能设置零比特位 |
| 15-1 | 保留位 | R-0 | 0h | 保留<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 0 | LOCK | R/WSonce | 0h | 锁定EPWM-XBAR的配置。一旦配置被锁定，对EPWM-XBAR的以下寄存器的写入被阻止。受锁定机制影响的寄存器：CLB-XBAROUTyMUX0TO15CFG |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | CLB-XBAROUTyMUX16TO31CFG<br>CLB-XBAROUTyMUXENABLE<br>CLB-XBAROUTLATEN<br>CLB-XBAROUTINV<br>0：允许写入上述寄存器<br>1：不允许写入上述寄存器<br>重置类型：CPU1.SYSRSN |

## 8.4.6 OUTPUT_XBAR_REGS 寄存器

表中未列出的寄存器偏移地址均为保留位置，不应修改这些位置的寄存器内容。

表 8-171 OUTPUT_XBAR_REGS 寄存器

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | OUTPUT1MUX0TO15CFG | 输出 1 的输出交叉开关多路选择器配置 |
| 4h | OUTPUT1MUX16TO31CFG | 输出 1 的输出交叉开关多路选择器配置 |
| 8h | OUTPUT2MUX0TO15CFG | 输出 2 的输出交叉开关多路选择器配置 |
| Ch | OUTPUT2MUX16TO31CFG | 输出 2 的输出交叉开关多路选择器配置 |
| 10h | OUTPUT3MUX0TO15CFG | 输出 3 的输出交叉开关多路选择器配置 |
| 14h | OUTPUT3MUX16TO31CFG | 输出 3 的输出交叉开关多路选择器配置 |
| 18h | OUTPUT4MUX0TO15CFG | 输出 4 的输出交叉开关多路选择器配置 |
| 1Ch | OUTPUT4MUX16TO31CFG | 输出 4 的输出交叉开关多路选择器配置 |
| 20h | OUTPUT5MUX0TO15CFG | 输出 5 的输出交叉开关多路选择器配置 |
| 24h | OUTPUT5MUX16TO31CFG | 输出 5 的输出交叉开关多路选择器配置 |
| 28h | OUTPUT6MUX0TO15CFG | 输出 6 的输出交叉开关多路选择器配置 |
| 2Ch | OUTPUT6MUX16TO31CFG | 输出 6 的输出交叉开关多路选择器配置 |
| 30h | OUTPUT7MUX0TO15CFG | 输出 7 的输出交叉开关多路选择器配置 |
| 34h | OUTPUT7MUX16TO31CFG | 输出 7 的输出交叉开关多路选择器配置 |
| 38h | OUTPUT8MUX0TO15CFG | 输出 8 的输出交叉开关多路选择器配置 |
| 3Ch | OUTPUT8MUX16TO31CFG | 输出 8 的输出交叉开关多路选择器配置 |
| 40h | OUTPUT1MUXENABLE | 输出 1 的输出交叉开关多路选择器使能 |
| 44h | OUTPUT2MUXENABLE | 输出 2 的输出交叉开关多路选择器使能 |
| 48h | OUTPUT3MUXENABLE | 输出 3 的输出交叉开关多路选择器使能 |
| 4Ch | OUTPUT4MUXENABLE | 输出 4 的输出交叉开关多路选择器使能 |
| 50h | OUTPUT5MUXENABLE | 输出 5 的输出交叉开关多路选择器使能 |
| 54h | OUTPUT6MUXENABLE | 输出 6 的输出交叉开关多路选择器使能 |
| 58h | OUTPUT7MUXENABLE | 输出 7 的输出交叉开关多路选择器使能 |

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 5Ch | OUTPUT8MUXENABLE | 输出 8 的输出交叉开关多路选择器使能 |
| 60h | OUTPUTLATCH | 输出交叉开关输出锁存 |
| 64h | OUTPUTLATCHCLR | 输出交叉开关输出锁存清除 |
| 68h | OUTPUTLATCHFRC | 输出交叉开关输出锁存强制 |
| 6Ch | OUTPUTLATCHENABLE | 输出交叉开关输出锁存使能 |
| 70h | OUTPUTINV | 输出交叉开关输出反相 |
| 7Ch | OUTPUTLOCK | 输出交叉开关配置锁定 |

复杂的位访问类型采用编码形式。下表列出了本节使用的访问类型编码。

表 8-172 OUTPUT_XBAR_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1S | W<br>1S | 写<br>写 1 置位 |
| WSonce | W<br>Sonce | 写<br>仅可置位一次 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i、j、k、l、m、n | | 用于寄存器名称、偏移地址或地址时，表示属于重复寄存器组的寄存器数组值。寄存器组按层次组织，数组用公式表示。 |
| y | | 用于寄存器名称、偏移地址或地址时，表示寄存器数组值。 |

### 8.4.6.1 OUTPUT1MUX0TO15CFG 寄存器

偏移地址：0h　复位值：00000000h

输出 X-BAR 的 OUTPUT1 多路选择器配置。

表 8-173 OUTPUT1MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-174 OUTPUT1MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | OUTPUT1 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | OUTPUT1 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | OUTPUT1 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | OUTPUT1 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | OUTPUT1 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | OUTPUT1 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | OUTPUT1 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | OUTPUT1 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | OUTPUT1 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | OUTPUT1 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | OUTPUT1 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | OUTPUT1 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | OUTPUT1 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | OUTPUT1 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | OUTPUT1 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | OUTPUT1 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.2 OUTPUT1MUX16TO31CFG 寄存器

偏移地址：4h　复位值：00000000h

输出 X-BAR 的 OUTPUT1 多路选择器配置。

表 8-175 OUTPUT1MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-176 OUTPUT1MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | OUTPUT1 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | OUTPUT1 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | OUTPUT1 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | OUTPUT1 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | OUTPUT1 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | OUTPUT1 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | OUTPUT1 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | OUTPUT1 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | OUTPUT1 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | OUTPUT1 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | OUTPUT1 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | OUTPUT1 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | OUTPUT1 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | OUTPUT1 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | OUTPUT1 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | OUTPUT1 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.3 OUTPUT2MUX0TO15CFG 寄存器

偏移地址：8h　复位值：00000000h

输出 X-BAR 的 OUTPUT2 多路选择器配置。

表 8-177 OUTPUT2MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-178 OUTPUT2MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | OUTPUT2 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | OUTPUT2 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | OUTPUT2 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | OUTPUT2 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | OUTPUT2 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | OUTPUT2 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | OUTPUT2 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | OUTPUT2 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | OUTPUT2 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | OUTPUT2 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | OUTPUT2 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | OUTPUT2 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | OUTPUT2 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | OUTPUT2 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | OUTPUT2 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | OUTPUT2 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.4 OUTPUT2MUX16TO31CFG 寄存器

偏移地址：ch　复位值：00000000h

输出 X-BAR 的 OUTPUT2 多路选择器配置。

表 8-179 OUTPUT2MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-180 OUTPUT2MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | OUTPUT2 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | OUTPUT2 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | OUTPUT2 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | OUTPUT2 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | OUTPUT2 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | OUTPUT2 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | OUTPUT2 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | OUTPUT2 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | OUTPUT2 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | OUTPUT2 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | OUTPUT2 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | OUTPUT2 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | OUTPUT2 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | OUTPUT2 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | OUTPUT2 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | OUTPUT2 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.5 OUTPUT3MUX0TO15CFG 寄存器

偏移地址：10h　复位值：00000000h

输出 X-BAR 的 OUTPUT3 多路选择器配置。

表 8-181 OUTPUT3MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-182 OUTPUT3MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | OUTPUT3 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | OUTPUT3 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | OUTPUT3 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | OUTPUT3 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | OUTPUT3 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | OUTPUT3 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | OUTPUT3 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | OUTPUT3 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | OUTPUT3 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | OUTPUT3 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | OUTPUT3 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | OUTPUT3 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | OUTPUT3 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | OUTPUT3 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | OUTPUT3 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | OUTPUT3 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.6 OUTPUT3MUX16TO31CFG 寄存器

偏移地址：14h　复位值：00000000h

输出 X-BAR 的 OUTPUT3 多路选择器配置。

表 8-183 OUTPUT3MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-184 OUTPUT3MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | OUTPUT3 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | OUTPUT3 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | OUTPUT3 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | OUTPUT3 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | OUTPUT3 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | OUTPUT3 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | OUTPUT3 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | OUTPUT3 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | OUTPUT3 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | OUTPUT3 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | OUTPUT3 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | OUTPUT3 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | OUTPUT3 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | OUTPUT3 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | OUTPUT3 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | OUTPUT3 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.7 OUTPUT4MUX0TO15CFG 寄存器

偏移地址：18h　复位值：00000000h

输出 X-BAR 的 OUTPUT4 多路选择器配置。

表 8-185 OUTPUT4MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-186 OUTPUT4MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | OUTPUT4 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | OUTPUT4 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | OUTPUT4 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | OUTPUT4 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | OUTPUT4 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | OUTPUT4 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | OUTPUT4 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | OUTPUT4 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | OUTPUT4 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | OUTPUT4 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | OUTPUT4 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | OUTPUT4 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | OUTPUT4 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | OUTPUT4 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | OUTPUT4 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | OUTPUT4 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.8 OUTPUT4MUX16TO31CFG 寄存器

偏移地址：1ch　复位值：00000000h

输出 X-BAR 的 OUTPUT4 多路选择器配置。

表 8-187 OUTPUT4MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-188 OUTPUT4MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | OUTPUT4 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | OUTPUT4 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | OUTPUT4 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | OUTPUT4 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | OUTPUT4 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | OUTPUT4 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | OUTPUT4 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | OUTPUT4 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | OUTPUT4 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | OUTPUT4 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | OUTPUT4 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | OUTPUT4 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | OUTPUT4 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | OUTPUT4 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | OUTPUT4 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | OUTPUT4 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.9 OUTPUT5MUX0TO15CFG 寄存器

偏移地址：20h　复位值：00000000h

输出 X-BAR 的 OUTPUT5 多路选择器配置。

表 8-189 OUTPUT5MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-190 OUTPUT5MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | OUTPUT5 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | OUTPUT5 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | OUTPUT5 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | OUTPUT5 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | OUTPUT5 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | OUTPUT5 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | OUTPUT5 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | OUTPUT5 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | OUTPUT5 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | OUTPUT5 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | OUTPUT5 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | OUTPUT5 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | OUTPUT5 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | OUTPUT5 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | OUTPUT5 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | OUTPUT5 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.10 OUTPUT5MUX16TO31CFG 寄存器

偏移地址：24h　复位值：00000000h

输出 X-BAR 的 OUTPUT5 多路选择器配置。

表 8-191 OUTPUT5MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-192 OUTPUT5MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | OUTPUT5 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | OUTPUT5 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | OUTPUT5 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | OUTPUT5 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | OUTPUT5 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | OUTPUT5 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | OUTPUT5 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | OUTPUT5 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | OUTPUT5 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | OUTPUT5 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | OUTPUT5 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | OUTPUT5 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | OUTPUT5 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | OUTPUT5 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | OUTPUT5 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | OUTPUT5 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.11 OUTPUT6MUX0TO15CFG 寄存器

偏移地址：28h　复位值：00000000h

输出 X-BAR 的 OUTPUT6 多路选择器配置。

表 8-193 OUTPUT6MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-194 OUTPUT6MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | OUTPUT6 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | OUTPUT6 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | OUTPUT6 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | OUTPUT6 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | OUTPUT6 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | OUTPUT6 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | OUTPUT6 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | OUTPUT6 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | OUTPUT6 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | OUTPUT6 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | OUTPUT6 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | OUTPUT6 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | OUTPUT6 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | OUTPUT6 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | OUTPUT6 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | OUTPUT6 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.12 OUTPUT6MUX16TO31CFG 寄存器

偏移地址：2ch　复位值：00000000h

输出 X-BAR 的 OUTPUT6 多路选择器配置。

表 8-195 OUTPUT6MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-196 OUTPUT6MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | OUTPUT6 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | OUTPUT6 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | OUTPUT6 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | OUTPUT6 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | OUTPUT6 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | OUTPUT6 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | OUTPUT6 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | OUTPUT6 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | OUTPUT6 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | OUTPUT6 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | OUTPUT6 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | OUTPUT6 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | OUTPUT6 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | OUTPUT6 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | OUTPUT6 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | OUTPUT6 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.13 OUTPUT7MUX0TO15CFG 寄存器

偏移地址：30h　复位值：00000000h

输出 X-BAR 的 OUTPUT7 多路选择器配置。

表 8-197 OUTPUT7MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-198 OUTPUT7MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | OUTPUT7 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | OUTPUT7 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | OUTPUT7 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | OUTPUT7 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | OUTPUT7 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | OUTPUT7 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | OUTPUT7 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | OUTPUT7 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | OUTPUT7 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | OUTPUT7 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | OUTPUT7 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | OUTPUT7 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | OUTPUT7 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | OUTPUT7 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | OUTPUT7 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | OUTPUT7 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.14 OUTPUT7MUX16TO31CFG 寄存器

偏移地址：34h　复位值：00000000h

输出 X-BAR 的 OUTPUT7 多路选择器配置。

表 8-199 OUTPUT7MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-200 OUTPUT7MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | OUTPUT7 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | OUTPUT7 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | OUTPUT7 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | OUTPUT7 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | OUTPUT7 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | OUTPUT7 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | OUTPUT7 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | OUTPUT7 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | OUTPUT7 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | OUTPUT7 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | OUTPUT7 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | OUTPUT7 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | OUTPUT7 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | OUTPUT7 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | OUTPUT7 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | OUTPUT7 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.15 OUTPUT8MUX0TO15CFG 寄存器

偏移地址：38h　复位值：00000000h

输出 X-BAR 的 OUTPUT8 多路选择器配置。

表 8-201 OUTPUT8MUX0TO15CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX15 | | MUX14 | | MUX13 | | MUX12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX11 | | MUX10 | | MUX9 | | MUX8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX7 | | MUX6 | | MUX5 | | MUX4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX3 | | MUX2 | | MUX1 | | MUX0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-202 OUTPUT8MUX0TO15CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX15 | R/W | 0h | OUTPUT8 MUX15的选择位：<br>00：为Mux15选择0输入<br>01：为Mux15选择1输入<br>10：为Mux15选择2输入<br>11：为Mux15选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX14 | R/W | 0h | OUTPUT8 MUX14的选择位：<br>00：为Mux14选择0输入<br>01：为Mux14选择1输入<br>10：为Mux14选择2输入<br>11：为Mux14选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX13 | R/W | 0h | OUTPUT8 MUX13的选择位：<br>00：为Mux13选择0输入<br>01：为Mux13选择1输入<br>10：为Mux13选择2输入<br>11：为Mux13选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX12 | R/W | 0h | OUTPUT8 MUX12的选择位：<br>00：为Mux12选择0输入<br>01：为Mux12选择1输入<br>10：为Mux12选择2输入<br>11：为Mux12选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX11 | R/W | 0h | OUTPUT8 MUX11的选择位：<br>00：为Mux11选择0输入<br>01：为Mux11选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux11选择2输入<br>11：为Mux11选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX10 | R/W | 0h | OUTPUT8 MUX10的选择位：<br>00：为Mux10选择0输入<br>01：为Mux10选择1输入<br>10：为Mux10选择2输入<br>11：为Mux10选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX9 | R/W | 0h | OUTPUT8 MUX9的选择位：<br>00：为Mux9选择0输入<br>01：为Mux9选择1输入<br>10：为Mux9选择2输入<br>11：为Mux9选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX8 | R/W | 0h | OUTPUT8 MUX8的选择位：<br>00：为Mux8选择0输入<br>01：为Mux8选择1输入<br>10：为Mux8选择2输入<br>11：为Mux8选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX7 | R/W | 0h | OUTPUT8 MUX7的选择位：<br>00：为Mux7选择0输入<br>01：为Mux7选择1输入<br>10：为Mux7选择2输入<br>11：为Mux7选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX6 | R/W | 0h | OUTPUT8 MUX6的选择位：<br>00：为Mux6选择0输入<br>01：为Mux6选择1输入<br>10：为Mux6选择2输入<br>11：为Mux6选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX5 | R/W | 0h | OUTPUT8 MUX5的选择位：<br>00：为Mux5选择0输入<br>01：为Mux5选择1输入<br>10：为Mux5选择2输入<br>11：为Mux5选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX4 | R/W | 0h | OUTPUT8 MUX4的选择位：<br>00：为Mux4选择0输入<br>01：为Mux4选择1输入<br>10：为Mux4选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux4选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX3 | R/W | 0h | OUTPUT8 MUX3的选择位：<br>00：为Mux3选择0输入<br>01：为Mux3选择1输入<br>10：为Mux3选择2输入<br>11：为Mux3选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX2 | R/W | 0h | OUTPUT8 MUX2的选择位：<br>00：为Mux2选择0输入<br>01：为Mux2选择1输入<br>10：为Mux2选择2输入<br>11：为Mux2选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX1 | R/W | 0h | OUTPUT8 MUX1的选择位：<br>00：为Mux1选择0输入<br>01：为Mux1选择1输入<br>10：为Mux1选择2输入<br>11：为Mux1选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX0 | R/W | 0h | OUTPUT8 MUX0的选择位：<br>00：为Mux0选择0输入<br>01：为Mux0选择1输入<br>10：为Mux0选择2输入<br>11：为Mux0选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.16 OUTPUT8MUX16TO31CFG 寄存器

偏移地址：3ch　复位值：00000000h

输出 X-BAR 的 OUTPUT8 多路选择器配置。

表 8-203 OUTPUT8MUX16TO31CFG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | | MUX30 | | MUX29 | | MUX28 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX27 | | MUX26 | | MUX25 | | MUX24 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX23 | | MUX22 | | MUX21 | | MUX20 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX19 | | MUX18 | | MUX17 | | MUX16 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 8-204 OUTPUT8MUX16TO31CFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | MUX31 | R/W | 0h | OUTPUT8 MUX31的选择位：<br>00：为Mux31选择0输入<br>01：为Mux31选择1输入<br>10：为Mux31选择2输入<br>11：为Mux31选择3输入<br>重置类型：CPU1.SYSRSN |
| 29-28 | MUX30 | R/W | 0h | OUTPUT8 MUX30的选择位：<br>00：为Mux30选择0输入<br>01：为Mux30选择1输入<br>10：为Mux30选择2输入<br>11：为Mux30选择3输入<br>重置类型：CPU1.SYSRSN |
| 27-26 | MUX29 | R/W | 0h | OUTPUT8 MUX29的选择位：<br>00：为Mux29选择0输入<br>01：为Mux29选择1输入<br>10：为Mux29选择2输入<br>11：为Mux29选择3输入<br>重置类型：CPU1.SYSRSN |
| 25-24 | MUX28 | R/W | 0h | OUTPUT8 MUX28的选择位：<br>00：为Mux28选择0输入<br>01：为Mux28选择1输入<br>10：为Mux28选择2输入<br>11：为Mux28选择3输入<br>重置类型：CPU1.SYSRSN |
| 23-22 | MUX27 | R/W | 0h | OUTPUT8 MUX27的选择位：<br>00：为Mux27选择0输入<br>01：为Mux27选择1输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 10：为Mux27选择2输入<br>11：为Mux27选择3输入<br>重置类型：CPU1.SYSRSN |
| 21-20 | MUX26 | R/W | 0h | OUTPUT8 MUX26的选择位：<br>00：为Mux26选择0输入<br>01：为Mux26选择1输入<br>10：为Mux26选择2输入<br>11：为Mux26选择3输入<br>重置类型：CPU1.SYSRSN |
| 19-18 | MUX25 | R/W | 0h | OUTPUT8 MUX25的选择位：<br>00：为Mux25选择0输入<br>01：为Mux25选择1输入<br>10：为Mux25选择2输入<br>11：为Mux25选择3输入<br>重置类型：CPU1.SYSRSN |
| 17-16 | MUX24 | R/W | 0h | OUTPUT8 MUX24的选择位：<br>00：为Mux24选择0输入<br>01：为Mux24选择1输入<br>10：为Mux24选择2输入<br>11：为Mux24选择3输入<br>重置类型：CPU1.SYSRSN |
| 15-14 | MUX23 | R/W | 0h | OUTPUT8 MUX23的选择位：<br>00：为Mux23选择0输入<br>01：为Mux23选择1输入<br>10：为Mux23选择2输入<br>11：为Mux23选择3输入<br>重置类型：CPU1.SYSRSN |
| 13-12 | MUX22 | R/W | 0h | OUTPUT8 MUX22的选择位：<br>00：为Mux22选择0输入<br>01：为Mux22选择1输入<br>10：为Mux22选择2输入<br>11：为Mux22选择3输入<br>重置类型：CPU1.SYSRSN |
| 11-10 | MUX21 | R/W | 0h | OUTPUT8 MUX21的选择位：<br>00：为Mux21选择0输入<br>01：为Mux21选择1输入<br>10：为Mux21选择2输入<br>11：为Mux21选择3输入<br>重置类型：CPU1.SYSRSN |
| 9-8 | MUX20 | R/W | 0h | OUTPUT8 MUX20的选择位：<br>00：为Mux20选择0输入<br>01：为Mux20选择1输入<br>10：为Mux20选择2输入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：为Mux20选择3输入<br>重置类型：CPU1.SYSRSN |
| 7-6 | MUX19 | R/W | 0h | OUTPUT8 MUX19的选择位：<br>00：为Mux19选择0输入<br>01：为Mux19选择1输入<br>10：为Mux19选择2输入<br>11：为Mux19选择3输入<br>重置类型：CPU1.SYSRSN |
| 5-4 | MUX18 | R/W | 0h | OUTPUT8 MUX18的选择位：<br>00：为Mux18选择0输入<br>01：为Mux18选择1输入<br>10：为Mux18选择2输入<br>11：为Mux18选择3输入<br>重置类型：CPU1.SYSRSN |
| 3-2 | MUX17 | R/W | 0h | OUTPUT8 MUX17的选择位：<br>00：为Mux17选择0输入<br>01：为Mux17选择1输入<br>10：为Mux17选择2输入<br>11：为Mux17选择3输入<br>重置类型：CPU1.SYSRSN |
| 1-0 | MUX16 | R/W | 0h | OUTPUT8 MUX16的选择位：<br>00：为Mux16选择0输入<br>01：为Mux16选择1输入<br>10：为Mux16选择2输入<br>11：为Mux16选择3输入<br>重置类型：CPU1.SYSRSN |

### 8.4.6.17 OUTPUT1MUXENABLE 寄存器

偏移地址：40h　复位值：00000000h

输出 X-BAR 的 OUTPUT1 多路选择器使能。

表 8-205 OUTPUT1MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-206 OUTPUT1MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动OUTPUT-XBAR的OUTPUT1 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.6.18 OUTPUT2MUXENABLE 寄存器

偏移地址：44h　复位值：00000000h

输出 X-BAR 的 OUTPUT2 多路选择器使能。

表 8-207 OUTPUT2MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-208 OUTPUT2MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动OUTPUT-XBAR的OUTPUT1<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动OUTPUT-XBAR的OUTPUT2 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动OUTPUT-XBAR的OUTPUT2<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.6.19 OUTPUT3MUXENABLE 寄存器

偏移地址：48h　复位值：00000000h

输出 X-BAR 的 OUTPUT3 多路选择器使能。

表 8-209 OUTPUT3MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-210 OUTPUT3MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动OUTPUT-XBAR的OUTPUT3 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动OUTPUT-XBAR的OUTPUT3<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.6.20 OUTPUT4MUXENABLE 寄存器

偏移地址：4ch　复位值：00000000h

输出 X-BAR 的 OUTPUT4 多路选择器使能。

表 8-211 OUTPUT4MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-212 OUTPUT4MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动OUTPUT-XBAR的OUTPUT4 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动OUTPUT-XBAR的OUTPUT4<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.6.21 OUTPUT5MUXENABLE 寄存器

偏移地址：50h　复位值：00000000h

输出 X-BAR 的 OUTPUT5 多路选择器使能。

表 8-213 OUTPUT5MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-214 OUTPUT5MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动OUTPUT-XBAR的OUTPUT5 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动OUTPUT-XBAR的OUTPUT5<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.6.22 OUTPUT6MUXENABLE 寄存器

偏移地址：54h　复位值：00000000h

输出 X-BAR 的 OUTPUT6 多路选择器使能。

表 8-215 OUTPUT6MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-216 OUTPUT6MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动OUTPUT-XBAR的OUTPUT6 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动OUTPUT-XBAR的OUTPUT6<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.6.23 OUTPUT7MUXENABLE 寄存器

偏移地址：58h　复位值：00000000h

输出 X-BAR 的 OUTPUT7 多路选择器使能。

表 8-217 OUTPUT7MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-218 OUTPUT7MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动OUTPUT-XBAR的OUTPUT7 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动OUTPUT-XBAR的OUTPUT7<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.6.24 OUTPUT8MUXENABLE 寄存器

偏移地址：5ch　复位值：00000000h

输出 X-BAR 的 OUTPUT8 多路选择器使能。

表 8-219 OUTPUT8MUXENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MUX31 | MUX30 | MUX29 | MUX28 | MUX27 | MUX26 | MUX25 | MUX24 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MUX23 | MUX22 | MUX21 | MUX20 | MUX19 | MUX18 | MUX17 | MUX16 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MUX15 | MUX14 | MUX13 | MUX12 | MUX11 | MUX10 | MUX9 | MUX8 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MUX7 | MUX6 | MUX5 | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 8-220 OUTPUT8MUXENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MUX31 | R/W | 0h | 选择MUX31的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 30 | MUX30 | R/W | 0h | 选择MUX30的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 29 | MUX29 | R/W | 0h | 选择MUX29的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 重置类型：CPU1.SYSRSN |
| 28 | MUX28 | R/W | 0h | 选择MUX28的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 27 | MUX27 | R/W | 0h | 选择MUX27的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 26 | MUX26 | R/W | 0h | 选择MUX26的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 25 | MUX25 | R/W | 0h | 选择MUX25的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 24 | MUX24 | R/W | 0h | 选择MUX24的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 23 | MUX23 | R/W | 0h | 选择MUX23的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 22 | MUX22 | R/W | 0h | 选择MUX22的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 21 | MUX21 | R/W | 0h | 选择MUX21的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 20 | MUX20 | R/W | 0h | 选择MUX20的输出以驱动OUTPUT-XBAR的OUTPUT8 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 19 | MUX19 | R/W | 0h | 选择MUX19的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 18 | MUX18 | R/W | 0h | 选择MUX18的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 17 | MUX17 | R/W | 0h | 选择MUX17的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 16 | MUX16 | R/W | 0h | 选择MUX16的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 15 | MUX15 | R/W | 0h | 选择MUX15的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 14 | MUX14 | R/W | 0h | 选择MUX14的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 13 | MUX13 | R/W | 0h | 选择MUX13的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 12 | MUX12 | R/W | 0h | 选择MUX12的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 11 | MUX11 | R/W | 0h | 选择MUX11的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 10 | MUX10 | R/W | 0h | 选择MUX10的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 9 | MUX9 | R/W | 0h | 选择MUX9的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 8 | MUX8 | R/W | 0h | 选择MUX8的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 7 | MUX7 | R/W | 0h | 选择MUX7的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 6 | MUX6 | R/W | 0h | 选择MUX6的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 5 | MUX5 | R/W | 0h | 选择MUX5的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 4 | MUX4 | R/W | 0h | 选择MUX4的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 3 | MUX3 | R/W | 0h | 选择MUX3的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 2 | MUX2 | R/W | 0h | 选择MUX2的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 1 | MUX1 | R/W | 0h | 选择MUX1的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |
| 0 | MUX0 | R/W | 0h | 选择MUX0的输出以驱动OUTPUT-XBAR的OUTPUT8<br>0：禁用此输出驱动<br>1：使能此输出驱动<br>重置类型：CPU1.SYSRSN |

### 8.4.6.25 OUTPUTLATCH 寄存器

偏移地址：60h　复位值：00000000h

输出 X-BAR 输出锁存。

表 8-221 OUTPUTLATCH 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUTPUT8 | OUTPUT7 | OUTPUT6 | OUTPUT5 | OUTPUT4 | OUTPUT3 | OUTPUT2 | OUTPUT1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 8-222 OUTPUTLATCH 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-8 | 保留位 | R-0 | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 7 | OUTPUT8 | R | 0h | 记录OUTPUT-XBAR的OUTPUT8<br>0：输出未翻转<br>1：输出翻转<br>重置类型：CPU1.SYSRSN |
| 6 | OUTPUT7 | R | 0h | 选记录OUTPUT-XBAR的OUTPUT7<br>0：输出未翻转<br>1：输出翻转<br>重置类型：CPU1.SYSRSN |
| 5 | OUTPUT6 | R | 0h | 记录OUTPUT-XBAR的OUTPUT6<br>0：输出未翻转<br>1：输出翻转<br>重置类型：CPU1.SYSRSN |
| 4 | OUTPUT5 | R | 0h | 记录OUTPUT-XBAR的OUTPUT5<br>0：输出未翻转<br>1：输出翻转<br>重置类型：CPU1.SYSRSN |
| 3 | OUTPUT4 | R | 0h | 记录OUTPUT-XBAR的OUTPUT4<br>0：输出未翻转<br>1：输出翻转<br>重置类型：CPU1.SYSRSN |
| 2 | OUTPUT3 | R | 0h | 记录OUTPUT-XBAR的OUTPUT3<br>0：输出未翻转<br>1：输出翻转<br>重置类型：CPU1.SYSRSN |
| 1 | OUTPUT2 | R | 0h | 记录OUTPUT-XBAR的OUTPUT2<br>0：输出未翻转<br>1：输出翻转<br>重置类型：CPU1.SYSRSN |
| 0 | OUTPUT1 | R | 0h | 记录OUTPUT-XBAR的OUTPUT1<br>0：输出未翻转<br>1：输出翻转<br>重置类型：CPU1.SYSRSN |

## 8.4.6.26 OUTPUTLATCHCLR 寄存器

偏移地址：64h　复位值：00000000h

输出 X-BAR 输出锁存清除。

表 8-223 OUTPUTLATCHCLR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUTPUT8 | OUTPUT7 | OUTPUT6 | OUTPUT5 | OUTPUT4 | OUTPUT3 | OUTPUT2 | OUTPUT1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 8-224 OUTPUTLATCHCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-8 | 保留位 | R-0 | 0h | 保留 |
| 7 | OUTPUT8 | R | 0h | 清除OUTPUT-XBAR的OUTPUT8输出锁存<br>写入1清除OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 6 | OUTPUT7 | R | 0h | 清除OUTPUT-XBAR的OUTPUT7输出锁存<br>写入1清除OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 5 | OUTPUT6 | R | 0h | 清除OUTPUT-XBAR的OUTPUT6输出锁存<br>写入1清除OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 4 | OUTPUT5 | R | 0h | 清除OUTPUT-XBAR的OUTPUT5输出锁存<br>写入1清除OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 3 | OUTPUT4 | R | 0h | 清除OUTPUT-XBAR的OUTPUT4输出锁存<br>写入1清除OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 2 | OUTPUT3 | R | 0h | 清除OUTPUT-XBAR的OUTPUT3输出锁存<br>写入1清除OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | OUTPUT2 | R | 0h | 清除OUTPUT-XBAR的OUTPUT2输出锁存<br>写入1清除OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 0 | OUTPUT1 | R | 0h | 清除OUTPUT-XBAR的OUTPUT1输出锁存<br>写入1清除OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |

## 8.4.6.27 OUTPUTLATCHFRC 寄存器

偏移地址：68h　复位值：00000000h

输出 X-BAR 输出锁存强制置位。

表 8-225 OUTPUTLATCHFRC 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUTPUT8 | OUTPUT7 | OUTPUT6 | OUTPUT5 | OUTPUT4 | OUTPUT3 | OUTPUT2 | OUTPUT1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 8-226 OUTPUTLATCHFRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-8 | 保留位 | R-0 | 0h | 保留 |
| 7 | OUTPUT8 | R | 0h | 置位OUTPUT-XBAR的OUTPUT8输出锁存<br>写入1置位OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 6 | OUTPUT7 | R | 0h | 置位OUTPUT-XBAR的OUTPUT7输出锁存<br>写入1置位OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 5 | OUTPUT6 | R | 0h | 置位OUTPUT-XBAR的OUTPUT6输出锁存 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 写入1置位OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 4 | OUTPUT5 | R | 0h | 置位OUTPUT-XBAR的OUTPUT5输出锁存<br>写入1置位OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 3 | OUTPUT4 | R | 0h | 置位OUTPUT-XBAR的OUTPUT4输出锁存<br>写入1置位OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 2 | OUTPUT3 | R | 0h | 置位OUTPUT-XBAR的OUTPUT3输出锁存<br>写入1置位OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 1 | OUTPUT2 | R | 0h | 置位OUTPUT-XBAR的OUTPUT2输出锁存<br>写入1置位OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |
| 0 | OUTPUT1 | R | 0h | 置位OUTPUT-XBAR的OUTPUT1输出锁存<br>写入1置位OUTPUTLATCH寄存器中相应的输出锁存位<br>重置类型：CPU1.SYSRSN |

### 8.4.6.28 OUTPUTLATCHENABLE 寄存器

偏移地址：6ch　复位值：00000000h

输出 X-BAR 输出锁存使能。

表 8-227 OUTPUTLATCHENABLE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUTPUT8 | OUTPUT7 | OUTPUT6 | OUTPUT5 | OUTPUT4 | OUTPUT3 | OUTPUT2 | OUTPUT1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 8-228 OUTPUTLATCHENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-8 | 保留位 | R-0 | 0h | 保留 |
| 7 | OUTPUT8 | R | 0h | 选择输出锁存器以驱动输出XBAR的OUTPUT8<br>0：未选择输出锁存来驱动各自的输出<br>1：选择输出锁存来驱动各自的输出<br>重置类型：CPU1.SYSRSN |
| 6 | OUTPUT7 | R | 0h | 选择输出锁存器以驱动输出XBAR的OUTPUT7<br>0：未选择输出锁存来驱动各自的输出<br>1：选择输出锁存来驱动各自的输出<br>重置类型：CPU1.SYSRSN |
| 5 | OUTPUT6 | R | 0h | 选择输出锁存器以驱动输出XBAR的OUTPUT6<br>0：未选择输出锁存来驱动各自的输出<br>1：选择输出锁存来驱动各自的输出<br>重置类型：CPU1.SYSRSN |
| 4 | OUTPUT5 | R | 0h | 置选择输出锁存器以驱动输出XBAR的OUTPUT5<br>0：未选择输出锁存来驱动各自的输出<br>1：选择输出锁存来驱动各自的输出<br>重置类型：CPU1.SYSRSN |
| 3 | OUTPUT4 | R | 0h | 选择输出锁存器以驱动输出XBAR的OUTPUT4<br>0：未选择输出锁存来驱动各自的输出<br>1：选择输出锁存来驱动各自的输出<br>重置类型：CPU1.SYSRSN |
| 2 | OUTPUT3 | R | 0h | 选择输出锁存器以驱动输出XBAR的OUTPUT3<br>0：未选择输出锁存来驱动各自的输出<br>1：选择输出锁存来驱动各自的输出<br>重置类型：CPU1.SYSRSN |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | OUTPUT2 | R | 0h | 选择输出锁存器以驱动输出XBAR的OUTPUT2<br>0：未选择输出锁存来驱动各自的输出<br>1：选择输出锁存来驱动各自的输出<br>重置类型：CPU1.SYSRSN |
| 0 | OUTPUT1 | R | 0h | 选择输出锁存器以驱动输出XBAR的OUTPUT1<br>0：未选择输出锁存来驱动各自的输出<br>1：选择输出锁存来驱动各自的输出<br>重置类型：CPU1.SYSRSN |

### 8.4.6.29 OUTPUTINV 寄存器

偏移地址：70h　复位值：00000000h

输出 X-BAR 输出反相。

表 8-229 OUTPUTINV 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUTPUT8 | OUTPUT7 | OUTPUT6 | OUTPUT5 | OUTPUT4 | OUTPUT3 | OUTPUT2 | OUTPUT1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 8-230 OUTPUTINV 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-8 | 保留位 | R-0 | 0h | 保留 |
| 7 | OUTPUT8 | R | 0h | 选择OUTPUT-XBAR的OUTPUT8的极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 6 | OUTPUT7 | R | 0h | 选择OUTPUT-XBAR的OUTPUT7的极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 5 | OUTPUT6 | R | 0h | 选择OUTPUT-XBAR的OUTPUT6的极性 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 4 | OUTPUT5 | R | 0h | 选择OUTPUT-XBAR的OUTPUT5的极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 3 | OUTPUT4 | R | 0h | 选择OUTPUT-XBAR的OUTPUT4的极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 2 | OUTPUT3 | R | 0h | 选择OUTPUT-XBAR的OUTPUT3的极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 1 | OUTPUT2 | R | 0h | 选择OUTPUT-XBAR的OUTPUT2的极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 0 | OUTPUT1 | R | 0h | 选择OUTPUT-XBAR的OUTPUT1的极性<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |

### 8.4.6.30 OUTPUTLOCK 寄存器

偏移地址：7ch　复位值：00000000h

输出 X-BAR 配置锁定寄存器。

表 8-231 OUTPUTLOCK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| KEY | | | | | | | |
| R-0W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | LOCK |
| R-0-0h | | | | | | | R/WSonce-0h |

表 8-232 OUTPUTLOCK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | KEY | R-0W | 0h | KEY=0x5a5a时才能设置零比特位 |
| 15-1 | 保留位 | R-0 | 0h | 保留<br>0：驱动有效高输出<br>1：驱动有效低输出<br>重置类型：CPU1.SYSRSN |
| 0 | LOCK | R/WSonce | 0h | 锁定EPWM-XBAR的配置。一旦配置被锁定，对EPWM-XBAR的以下寄存器的写入被阻止。受锁定机制影响的寄存器：OUTPUT-XBAROUTyMUX0TO15CFG<br>OUTPUT-XBAROUTyMUX16TO31CFG<br>OUTPUT-XBAROUTyMUXENABLE<br>OUTPUT-XBAROUTLATENABLE<br>OUTPUT-XBAROUTINV<br>0：允许写入上述寄存器<br>1：不允许写入上述寄存器<br>重置类型：CPU1.SYSRSN |

## 8.4.7 寄存器与 DriverLib 函数的对应关系

### 8.4.7.1 INPUTXBAR 寄存器与 DriverLib 函数的对应关系

表 8-233 INPUTXBAR 寄存器与 DriverLib 函数的对应关系

| 文件 | DriverLib 函数 |
|---|---|
| **INPUT1SELECT** | |
| xbar.h | XBAR_setInputPin |
| **INPUT2SELECT** | |

| 文件 | DriverLib 函数 |
| --- | --- |
| - | 参见 INPUT1SELECT |
| **INPUT3SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT4SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT5SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT6SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT7SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT8SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT9SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT10SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT11SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT12SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT13SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT14SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT15SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUT16SELECT** | |
| - | 参见 INPUT1SELECT |
| **INPUTSELECTLOCK** | |
| xbar.h | XBAR_lockInput |

### 8.4.7.2 XBAR 寄存器与 DriverLib 函数的对应关系

表 8-234 XBAR 寄存器与 DriverLib 函数的对应关系

| 文件 | DriverLib 函数 |
|---|---|
| **FLG1** | |
| xbar.c | XBAR_getInputFlagStatus |
| **FLG2** | |
| xbar.c | XBAR_getInputFlagStatus |
| **FLG3** | |
| xbar.c | XBAR_getInputFlagStatus |
| **FLG4** | |
| xbar.c | XBAR_getInputFlagStatus |
| **CLR1** | |
| xbar.c | XBAR_clearInputFlag |
| **CLR2** | |
| xbar.c | XBAR_clearInputFlag |
| **CLR3** | |
| xbar.c | XBAR_clearInputFlag |
| **CLR4** | |
| xbar.c | XBAR_clearInputFlag |

### 8.4.7.3 EPWMXBAR 寄存器与 DriverLib 函数的对应关系

表 8-235 EPWMXBAR 寄存器与 DriverLib 函数的对应关系

| 文件 | DriverLib 函数 |
|---|---|
| **TRIP4MUX0TO15CFG** | |
| xbar.c | XBAR_setEPWMMuxConfig |
| **TRIP4MUX16TO31CFG** | |
| xbar.c | XBAR_setEPWMMuxConfig |
| **TRIP5MUX0TO15CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP5MUX16TO31CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP7MUX0TO15CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP7MUX16TO31CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |

| 文件 | DriverLib 函数 |
| --- | --- |
| **TRIP8MUX0TO15CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP8MUX16TO31CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP9MUX0TO15CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP9MUX16TO31CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP10MUX0TO15CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP10MUX16TO31CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP11MUX0TO15CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP11MUX16TO31CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP12MUX0TO15CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP12MUX16TO31CFG** | |
| - | 参见 TRIP4MUX0TO15CFG |
| **TRIP4MUXENABLE** | |
| xbar.h | XBAR_enableEPWMMux |
| xbar.h | XBAR_disableEPWMMux |
| **TRIP5MUXENABLE** | |
| - | 参见 TRIP4MUXENABLE |
| **TRIP7MUXENABLE** | |
| - | 参见 TRIP4MUXENABLE |
| **TRIP8MUXENABLE** | |
| - | 参见 TRIP4MUXENABLE |
| **TRIP9MUXENABLE** | |
| - | 参见 TRIP4MUXENABLE |
| **TRIP10MUXENABLE** | |
| - | 参见 TRIP4MUXENABLE |
| **TRIP11MUXENABLE** | |
| - | 参见 TRIP4MUXENABLE |

| 文件 | DriverLib 函数 |
| --- | --- |
| **TRIP12MUXENABLE** | |
| - | 参见 TRIP4MUXENABLE |
| **TRIPOUTINV** | |
| xbar.h | XBAR_invertEPWMSignal |
| **TRIPLOCK** | |
| xbar.h | XBAR_lockEPWM |

### 8.4.7.4 CLBXBAR 寄存器与 DriverLib 函数的对应关系

表 8-236 CLBXBAR 寄存器与 DriverLib 函数的对应关系

| 文件 | DriverLib 函数 |
| --- | --- |
| **AUXSIG0MUX0TO15CFG** | |
| xbar.c | XBAR_setCLBMuxConfig |
| **AUXSIG0MUX16TO31CFG** | |
| xbar.c | XBAR_setCLBMuxConfig |
| **AUXSIG1MUX0TO15CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG1MUX16TO31CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG2MUX0TO15CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG2MUX16TO31CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG3MUX0TO15CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG3MUX16TO31CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG4MUX0TO15CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG4MUX16TO31CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG5MUX0TO15CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG5MUX16TO31CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG6MUX0TO15CFG** | |

| 文件 | DriverLib 函数 |
| --- | --- |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG6MUX16TO31CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG7MUX0TO15CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG7MUX16TO31CFG** | |
| - | 参见 AUXSIG0MUX0TO15CFG |
| **AUXSIG0MUXENABLE** | |
| xbar.h | XBAR_enableCLBMux |
| xbar.h | XBAR_disableCLBMux |
| **AUXSIG1MUXENABLE** | |
| - | 参见 AUXSIG0MUXENABLE |
| **AUXSIG2MUXENABLE** | |
| - | 参见 AUXSIG0MUXENABLE |
| **AUXSIG3MUXENABLE** | |
| - | 参见 AUXSIG0MUXENABLE |
| **AUXSIG4MUXENABLE** | |
| - | 参见 AUXSIG0MUXENABLE |
| **AUXSIG5MUXENABLE** | |
| - | 参见 AUXSIG0MUXENABLE |
| **AUXSIG6MUXENABLE** | |
| - | 参见 AUXSIG0MUXENABLE |
| **AUXSIG7MUXENABLE** | |
| - | 参见 AUXSIG0MUXENABLE |
| **AUXSIGOUTINV** | |
| xbar.h | XBAR_invertCLBSignal |
| **AUXSIGLOCK** | |
| - | |

### 8.4.7.5 OUTPUTXBAR 寄存器与 DriverLib 函数的对应关系

表 8-237 OUTPUTXBAR 寄存器与 DriverLib 函数的对应关系

| 文件 | DriverLib 函数 |
| --- | --- |
| **OUTPUT1MUX0TO15CFG** | |
| xbar.c | XBAR_setOutputMuxConfig |
| **OUTPUT1MUX16TO31CFG** | |

| 文件 | DriverLib 函数 |
| --- | --- |
| - | |
| **OUTPUT2MUX0TO15CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT2MUX16TO31CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT3MUX0TO15CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT3MUX16TO31CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT4MUX0TO15CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT4MUX16TO31CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT5MUX0TO15CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT5MUX16TO31CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT6MUX0TO15CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT6MUX16TO31CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT7MUX0TO15CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT7MUX16TO31CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT8MUX0TO15CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT8MUX16TO31CFG** | |
| - | 参见 OUTPUT1MUX0TO15CFG |
| **OUTPUT1MUXENABLE** | |
| xbar.h | XBAR_enableOutputMux |
| xbar.h | XBAR_disableOutputMux |
| **OUTPUT2MUXENABLE** | |
| - | 参见 OUTPUT1MUXENABLE |
| **OUTPUT3MUXENABLE** | |

| 文件 | DriverLib 函数 |
| --- | --- |
| - | 参见 OUTPUT1MUXENABLE |
| **OUTPUT4MUXENABLE** | |
| - | 参见 OUTPUT1MUXENABLE |
| **OUTPUT5MUXENABLE** | |
| - | 参见 OUTPUT1MUXENABLE |
| **OUTPUT6MUXENABLE** | |
| - | 参见 OUTPUT1MUXENABLE |
| **OUTPUT7MUXENABLE** | |
| - | 参见 OUTPUT1MUXENABLE |
| **OUTPUT8MUXENABLE** | |
| - | 参见 OUTPUT1MUXENABLE |
| **OUTPUTLATCH** | |
| xbar.h | XBAR_setOutputLatchMode |
| xbar.h | XBAR_getOutputLatchStatus |
| xbar.h | XBAR_clearOutputLatch |
| xbar.h | XBAR_forceOutputLatch |
| **OUTPUTLATCHCLR** | |
| xbar.h | XBAR_clearOutputLatch |
| **OUTPUTLATCHFRC** | |
| xbar.h | XBAR_forceOutputLatch |
| **OUTPUTLATCHENABLE** | |
| xbar.h | XBAR_setOutputLatchMode |
| **OUTPUTINV** | |
| xbar.h | XBAR_invertOutputSignal |
| **OUTPUTLOCK** | |
| xbar.h | XBAR_lockOutput |

### 8.4.7.6 TRIGXBAR 寄存器与 DriverLib 函数的对应关系

表 8-238 TRIGXBAR 寄存器与 DriverLib 函数的对应关系

| 文件 | DriverLib 函数 |
| --- | --- |
| **SYNCSELECT** | |
| - | |
| **ADCSOCOUTSELECT** | |
| - | |
| **SYNCSOCLOCK** | |

| 文件 | DriverLib 函数 |
| --- | --- |
| - | |

# 9 直接存储器访问（DMA）

## 9.1 简介

控制器的性能强弱并不能仅仅通过处理器主频来衡量，而应综合考量整个系统的能力。作为系统性能评估的一部分，只要能够降低 CPU 在某一具体功能上的带宽占用，系统整体能力就能够相应提升。在许多应用场景中，系统往往消耗大量 CPU 带宽用于数据搬移，例如将数据从片外存储器搬运到片上存储器、将来自模数转换器（ADC）等外设的数据传输到 RAM，或者在不同外设之间进行数据传输。此外，这些数据在多数情况下并不是以最有利于 CPU 高效处理的格式存在。本章所描述的 DMA（Direct Memory Access，直接存储器访问）模块，能够有效释放 CPU 带宽，并在数据传输过程中对数据进行重新组织，使其更适合后续高效、流水化的处理。

DMA 模块是一种基于事件触发的硬件单元，即 DMA 传输必须由外设事件或软件触发信号启动。尽管可以通过将定时器配置为 DMA 的触发源，从而构建一种周期性、时间驱动的工作模式，但 DMA 模块本身并不具备自主周期性启动内存传输的机制。

该 DMA 模块包含 6 个彼此独立的 DMA 通道，每个通道均可单独配置，并且每个通道都配备了一个独立的 CLIC 中断，用于在 DMA 传输开始或完成时通知 CPU。在这 6 个通道中，有 5 个通道在功能上完全一致，而 通道 1 具备可配置为高于其他通道优先级的能力。DMA 的核心由一个状态机以及与之紧密耦合的地址控制逻辑构成。该地址控制逻辑不仅支持在数据传输过程中对数据块进行重排，还支持在不同缓冲区之间进行“乒乓（Ping-Pong）”式的数据交替传输。这些功能特性将在本章中逐一进行详细说明。

## 9.2 主要特性

DMA 模块特性包括：

- 提供 6 个 DMA 通道，每个通道均配备独立的 CLIC 中断

- 每个 DMA 通道均可独立配置为由多个不同的外设触发源触发
- 字长支持：16 位或 32 位（SPI 外设仅支持 16 位）
- 数据吞吐率：在无仲裁条件下，传输效率为 每字 3 个时钟周期

## 9.3 功能描述

### 9.3.1 DMA 模块结构框图

DMA 模块框图如图所示。

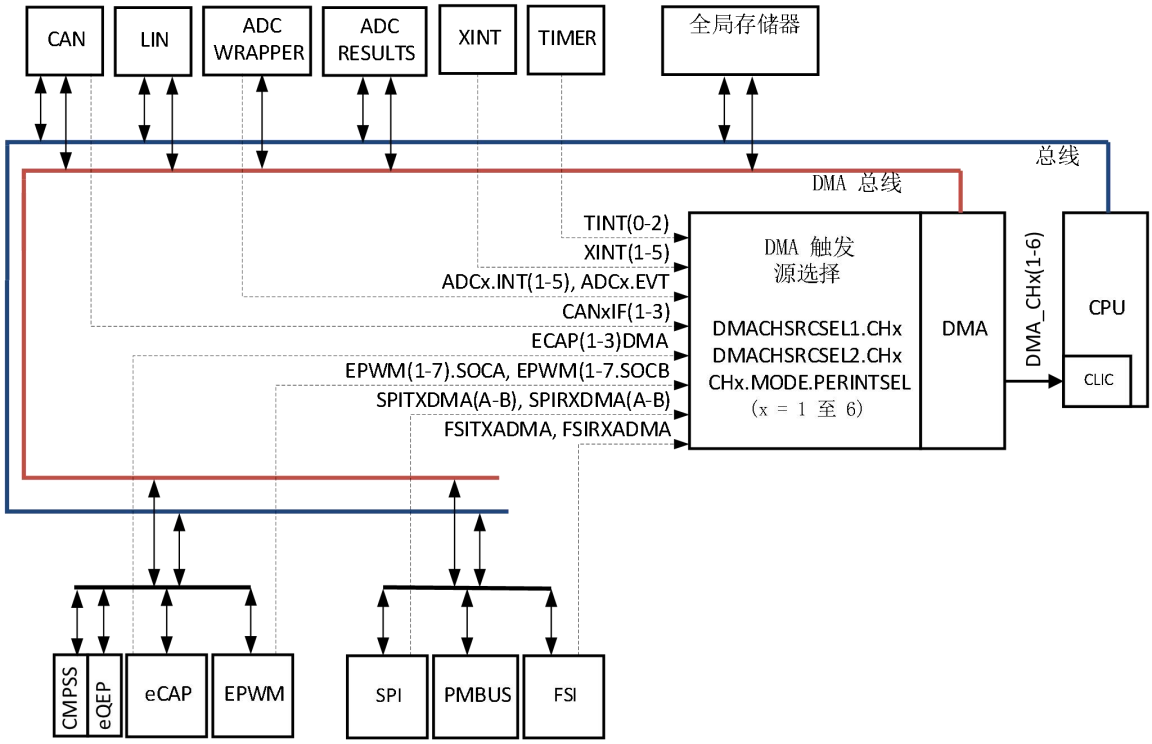



图 9-1 DMA 模块框图

### 9.3.2 DMA 结构

#### 9.3.2.1 外设中断触发源

每个 DMA 通道可以配置为被软件或其他外设触发。DMACHSRCSELx 寄存器可以用于为每个 DMA 通道配置 DMA 触发源；CHx.MODE.PERINTSEL 寄存器位应配置为通道号（CHx.MODE.PERINTSEL=x）。其中有 5 个是外部中断触发源，与大部

分 GPIO 相连，增加了 DMA 事件触发的灵活性。当收到一个外设中断事件时，DMA 会自动发送一个清除信号给中断源防止连续的触发事件到来。

软件触发的时候，可以不管 DMA 通道的模式寄存器中的触发源选择字段 MODE.CHx[PERINTSEL] 是什么，直接通过控制寄存器的外设中断强制位 CONTROL.CHx[PERINTFRC] 的置位来强制触发中断，同样，清除中断时也可以通过控制寄存器中的外设中断清除位 CONTROL.CHx[PERINTCLR] 来清除挂起的 DMA 触发。

PERINTFLG 中断触发标志位：一旦某个特定的中断触发置位了控制寄存器中的中断触发标志位 CONTROL.CHx[PERINTFLG]，该位会一直保持置位（挂起）状态，直到状态机的优先级逻辑允许该通道进行数据传输，数据传输开始，该标志位就会被清零。

如果数据正在传输过程中，一个新的中断触发产生了，在正常的优先级别下，数据传输完成后才会去相应新的中断触发。如果这个新的中断触发还处在挂起状态时，又来了一个新中断触发（第三个中断触发），这时候一个错误标志 CONTROL.CHx[OVRFLG]会被置 1 以表示错误。如果标志位正在清除中时，发生了一个外设中断，那么外设中断具有优先权，外设中断标志位还是会保持置位。

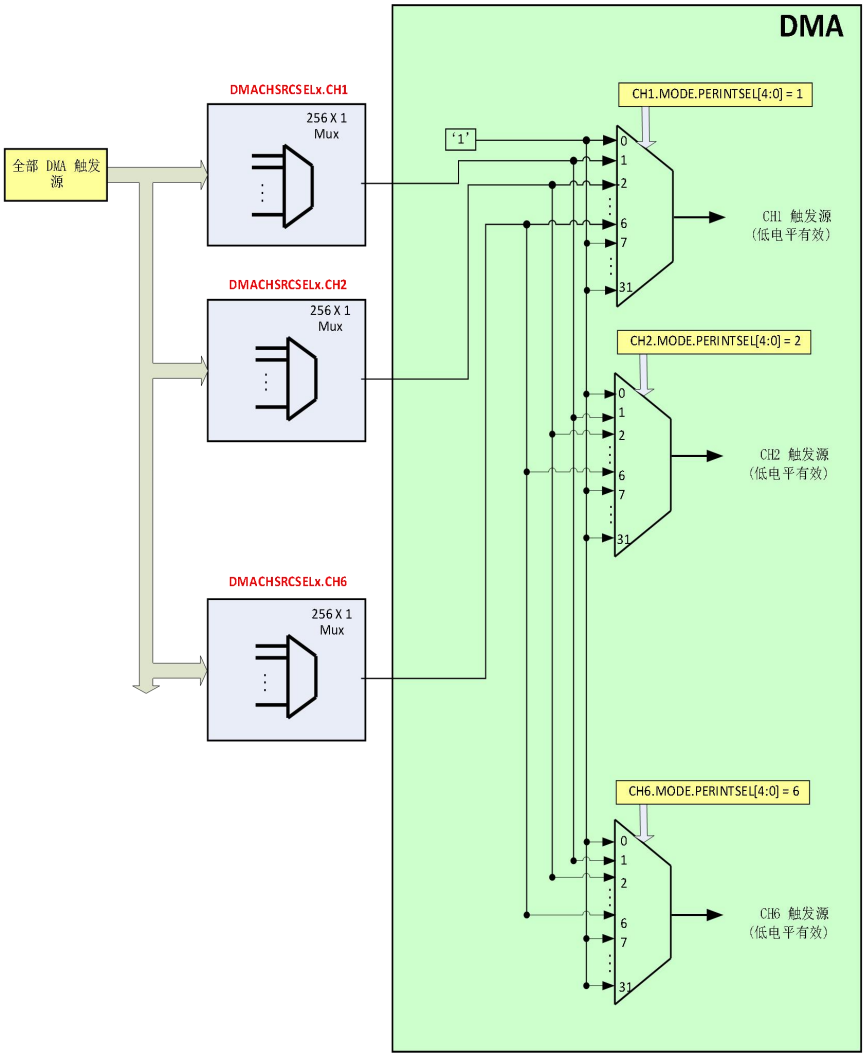



图 9-2 DMA 触发结构

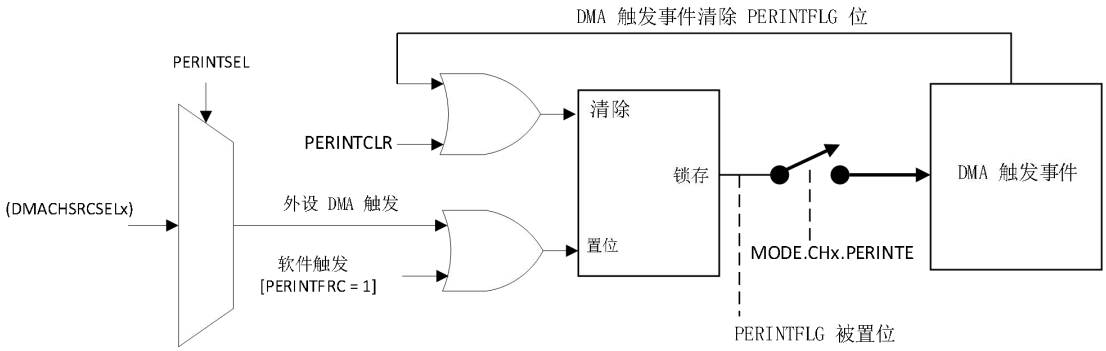



图 9-3 外设中断触发输入框图

DMA 外设中断触发源配置选择如下表所示,

表 9-1 DMA 中断触发源选项

| 选择值（8 位） | DMA中断触发源 |
|---|---|
| 0 | 软件触发源 |

| 选择值（8 位） | DMA中断触发源 |
| --- | --- |
| 1 | ADCA.1 |
| 2 | ADCA.2 |
| 3 | ADCA.3 |
| 4 | ADCA.4 |
| 5 | ADCAEVT |
| 6-10 | 保留 |
| 11 | ADCC.1 |
| 12 | ADCC.2 |
| 13 | ADCC.3 |
| 14 | ADCC.4 |
| 15 | ADCCEVT |
| 16-28 | 保留 |
| 29 | XINT1 |
| 30 | XINT2 |
| 31 | XINT3 |
| 32 | XINT4 |
| 33 | XINT5 |
| 34-35 | 保留 |
| 36 | EPWM1.SOCA |
| 37 | EPWM1.SOCB |
| 38 | EPWM2.SOCA |
| 39 | EPWM2.SOCB |
| 40 | EPWM3.SOCA |
| 41 | EPWM3.SOCB |
| 42 | EPWM4.SOCA |
| 43 | EPWM4.SOCB |
| 44 | EPWM5.SOCA |
| 45 | EPWM5.SOCB |
| 46 | EPWM6.SOCA |
| 47 | EPWM6.SOCB |
| 48 | EPWM7.SOCA |
| 49 | EPWM7.SOCB |
| 50-67 | 保留 |
| 68 | TINT0 |
| 69 | TINT1 |
| 70 | TINT2 |
| 71-74 | 保留 |
| 75 | ECAP1DMA |
| 76 | ECAP2DMA |
| 77 | ECAP3DMA |
| 78-108 | 保留 |
| 109 | SPITXDMAA |
| 110 | SPIRXDMAA |
| 111 | SPITXDMAB |
| 112 | SPIRXDMAB |
| 113-116 | 保留 |
| 117 | LINATXDMA |
| 118 | LINARXDMA |
| 119 | LINBTXDMA |

| 选择值（8 位） | DMA中断触发源 |
| --- | --- |
| 120 | LINBRXDMA |
| 121-122 | 保留 |
| 123 | FSITXADMA |
| 124 | 保留 |
| 125 | FSIRXADMA |
| 126 | 保留 |
| 127 | CLB1INT |
| 128 | CLB2INT |
| 129-166 | 保留 |
| 167 | CANAIF1 |
| 168 | CANAIF2 |
| 169 | CANAIF3 |

#### 9.3.2.2 DMA 总线

DMA 总线包括 32 位地址总线，32 位数据读总线，32 位数据写总线。存储器和寄存器连接到 DMA 总线接口上。

### 9.3.3 地址指针与传输控制

#### 9.3.3.1 DMA 传输的两级嵌套循环

DMA 模块内部状态机是两级嵌套循环结构：内部循环（burst loop）和外部循环（transfer loop）。

一次完整的 DMA 数据传输称为一次 transfer，可以包含多个 burst。

burst：burst 是一次 DMA 传输中的最小数据单元。BURST_SIZE 寄存器可以定义一次 burst 传输的数据量（多少个字），最多允许在一次 burst 中传输 32 个 16bit 的字。

transfer：TRANSFER_SIZE 寄存器可以设置一次 DMA 传输中执行多少次 burst 循环。TRANSFER_SIZE 是 16 位的寄存器，最大值是 65536，基本上可以满足任何实际的传输需求。

通知 CPU：在一次 DMA 传输的开始或结尾，会通过 CLIC 向 CPU 发送一个中断，通过模式寄存器中的 CHINTMODE（通道中断产生模式选择）位来决定是在开始还是结束时发送。

#### 9.3.3.2 地址指针介绍

每个 DMA 通道都包含源地址影子指针 SRC_ADDR 和目的地址影子指针 DST_ADDR，这些指针可以在传输状态机运行期间被独立的控制。每次传输开始之前，每个指针影子寄存器中的值会复制到各自的活动寄存器中。

在突发循环中，每完成一个字的传输，地址寄存器 SRC/DST_ADDR 会按照 BUREST_STEP 寄存器中有符号数进行递增/递减。

#### 9.3.3.3 地址递增的方式

在传输循环中，每完成一次 burst 传输，有两种方法可以修改活动地址指针：

第一种方法（默认方法）是一次 burst 完成后，源/目的地址指针会按照 SRC/DST_TRANSFER_STEP 寄存器中的有符号数进行递增或递减。

第二种方法是通过一个叫 wrap（“地址回绕”）的过程，如果启用地址回绕，那么当执行够 WRAP_SIZE 中设定的 burst 次数后，源/目的起始地址 BEG_ADDR 会加上回绕步长寄存器 WRAP_STEP 中的有符号数，进行递增或递减，从而得到下一次 burst 传输的地址。并且把源/目的起始地址 BEG_ADDR 的值复制给源/目的地址 ADDR。同时回绕计数器 SRC/DST_WRAP_COUNT 会重新加载回绕长度寄存器 SRC/DST_WRAP_SIZE 中的值，为下次回绕做准备，这就允许再一次传输过程中发生多次回绕操作。

transfer_step 和 WRAP 是二选一的关系，选择一种与 burst 传输搭配。如果想要屏蔽 WRAP，将源地址和目的地址的 WRAP_SIZE 值设置得大于 TRANSFER_SIZE 即可。

#### 9.3.3.4 影子寄存器和活动寄存器

每个 DMA 通道有 2 个起始地址影子指针 SRC_BEG_ADDR_SHADOW 和 DST_BEG_ADDR_SHADOW，允许源和目的相互独立的控制。如果 SRC_ADDR 和

DST_ADDR 一样，那么在传输开始之前，活动寄存器 SRC/DST_BEG_ADDR 将从自身映射单元（shadow counterpart）中加载数据。

DMA 的地址指针有活动寄存器（active register）和影子寄存器（shadow register），当 DMA 传输开始时，影子寄存器会被复制到活动寄存器，从而允许在 DMA 使用活动寄存器工作期间，改变影子寄存器的值，以便用于下一次传输。这种操作允许在不中断 DMA 通道执行的情况下进行乒乓缓冲方案。

有活动寄存器和影子寄存器的地址指针有：

① 源/目的地址指针 (SRC/DST_ADDR)

影子寄存器（shadow register）当中的值即为读/写操作的首地址，每次传输开始时，影子寄存器中的值都会被装载到活动寄存器（active register）中。

② 源/目的起始地址指针 (SRC/DST_BEG_ADDR)

这是一个回绕 wrap 指针。每次传输开始时，影子寄存器中的值都会被装载到活动寄存器中，活动寄存器的值在添加到 SRC/DST_ADDR 寄存器中之前，将现根据 SRC/DST_WRAP_STEP 寄存器中的值递增。

#### 9.3.3.5 传输过程中的循环次数控制

对于每个通道而言，传输过程由以下长度值进行控制：

① 源和目的突发传送长度 (BURST_SIZE)

BURST_SIZE 定义了一次突发传输所传递的字的个数。在突发传输开始前，BURST_SIZE 的值会被加载到 BURST_COUNT 寄存器中，每完成一个字的传输，BURST_COUNT 减 1，直到减为 0 时，说明本次突发传输结束，下一个通道可以得到响应。当前通道的行为由 MODE 寄存器中的 ONE_SHOT 位定义，每次突发传输的最大字数由外设决定。对于 ADC 而言，突发传输字可以为 16 个。对于 McBSP 而言，突发传输字的个数被限制为 1，因为其没有接收与发送缓冲器。对于 RAM 单元，突发传输的最大字的个数可以由 BURST_SIZE 寄存器设定，最大可以达到 32。

② 源和目的传输次数 (TRANSFER_SIZE)

TRANSFER_SIZE 指定了在一次 DMA 传输中的突发传送次数。在传输开始时，TRANSFER_SIZE 被装载到 TRANSFER_COUNT 寄存器中。TRANSFER_COUNT 寄存器中的值是剩余传输的次数，每完成一次 burst，TRANSFER_COUNT 减 1，直到 TRANSFER_COUNT 归零，表示 DMA 传输完成。

③ 源/目的回绕长度 (SRC/DST_WRAP_SIZE)

SRC/DST_WRAP_SIZE 定义了在进行地址回绕前需要执行的突发传输次数。在每次传输开始时，SRC/DST_WRAP_SIZE 的值被加载到 SRC/DST_WRAP_COUNT 寄存器中，SRC/DST_WRAP_COUNT 表示剩余的突发次数，每完成一次 burst，WRAP_COUNT 减 1，当 SRC/DST_WRAP_COUNT 归零时，执行源/目的地址的回绕。如果要禁止回绕功能（wrap），设定这些寄存器的值要大于 TRANSFER_SIZE 的值。

注意：写入 SIZE 寄存器的值的 size 要比预期的 size 小一个，因此，要传输 3 个 16-bit 的字，需要向 SIZE 寄存器中写入 2。无论 DATASIZE 位是什么状态，SIZE 寄存器中的值都是对于 16-bit 地址而言。因此要传输 3 个 32-bit 的字，需要向 SIZE 寄存器中写入 5。

#### 9.3.3.6 地址增量步长控制

源/目的地址的改变可以通过以下步长来控制：

① 源/目的突发步长 (SRC/DST_BURST_STEP)

每次突发传送，源地址和目的地址的步长增量都是由 SRC/DST_BURST_STEP 寄存器定义的。寄存器中的值是有符号二进制形式，地址指针按要求递增或递减。如果不要求地址指针增量步进（不要求递增或递减），可以将此寄存器的值设置为 0。

② 源/目的传输步长 (SRC/DST_TRANSFER_STEP)

这个寄存器定义了在当前突发传输完成后，下一次突发传输的地址偏移量。只有访问的寄存器或内存单元存在固定的地址间隔时，才能用这个功能。

寄存器中的值是有符号二进制形式，地址指针按要求递增或递减。

③ 源/目的地址回绕步长(SRC/DST_WRAP_STEP)

当回绕计数器归零时，BEG_ADDR 按照 WRAP_STEP 中的有符号数进行递增或递减，从而设定新的地址。

注意：无论 DATASIZE 位是什么状态，STEP 寄存器中的值都是对于 16-bit 地址而言。因此要增加 1 个 32-bit 的地址，需要向 STEP 寄存器中写入 2。

#### 9.3.3.7 状态机运行模式控制

控制 DMA 两级嵌套循环状态机的运行模式：

① 单次触发模式 (ONESHOT)

在一次外设中断触发信号到来时，如果单次触发模式是使能的，那么在 TRANSFER_COUNT 归零之前 DMA 会连续执行突发传送。如果单次触发模式被禁止，则每次突发传输过程都需要中断触发信号进行触发，直到 TRANSFER_COUNT 归零。

在 MODE.CHx[ONESHOT]位的默认设置中，DMA 在一次外设中断触发下仅发生一次突发传输。在此次突发传输完成之后，即使当前通道的触发信号再次到来，状态机也会根据优先级顺序移动到下一个通道，这样可以防止一个通道独占 DMA 总线。如果要传送的总数据量大于一次突发传送的最大数据量，那么可以通过将 MODE.CHx[ONESHOT] 位置 1，来完成整个传输过程。但需要注意的是，这可能会造成一次触发时间占用绝大部分 DMA 带宽的情况。

注意：当 ONESHOT 模式使能时，DMA 将连续传输突发数据，直到 TRANSFER_COUNT 归零，这可能会占用外设的带宽，并导致 CPU 暂停。为了避免这种情况，可以配置 CPU 定时器（orsimilar）并禁用单次触发模式。

高优先级模式和单次触发模式不能在 CH1 上同时使用。当 CH1 处于高优先级模式时，其他通道可以用单次触发模式。

② 连续触发模式 (CONTINUOUS)

如果连续触发模式被禁止，那么在传送结束后，会将 CONTROL 寄存器中的 RUNSTS 位清零，从而禁止 DMA 通道工作。如果想要在此通道发起新一次的传送，

那么首先要将 CONTROL 寄存器中的 RUN 位置 1，来重新启用该通道。如果连续触发模式被使能，则 RUNSTS 为在每次传送结束时不会被清零。

#### 9.3.3.8 通道中断模式控制

通过 MODE 寄存器中的 CHINTMODE 位可以将 CPU 中断配置成在传输开始时触发或在传输结束时触发。如果要用连续模式实现乒乓操作，则中断在传送开始时发生。如果 DMA 没有工作在连续模式，则中断通常在传送结束时产生。通过 MODE 寄存器中的 CONTINUOUS 位可以设定在传输完成后，当前通道是继续使能还是禁止工作。

#### 9.3.3.9 DMA 模块状态转换图

以上所有特性和模式选择如 DMA 状态框图所示：

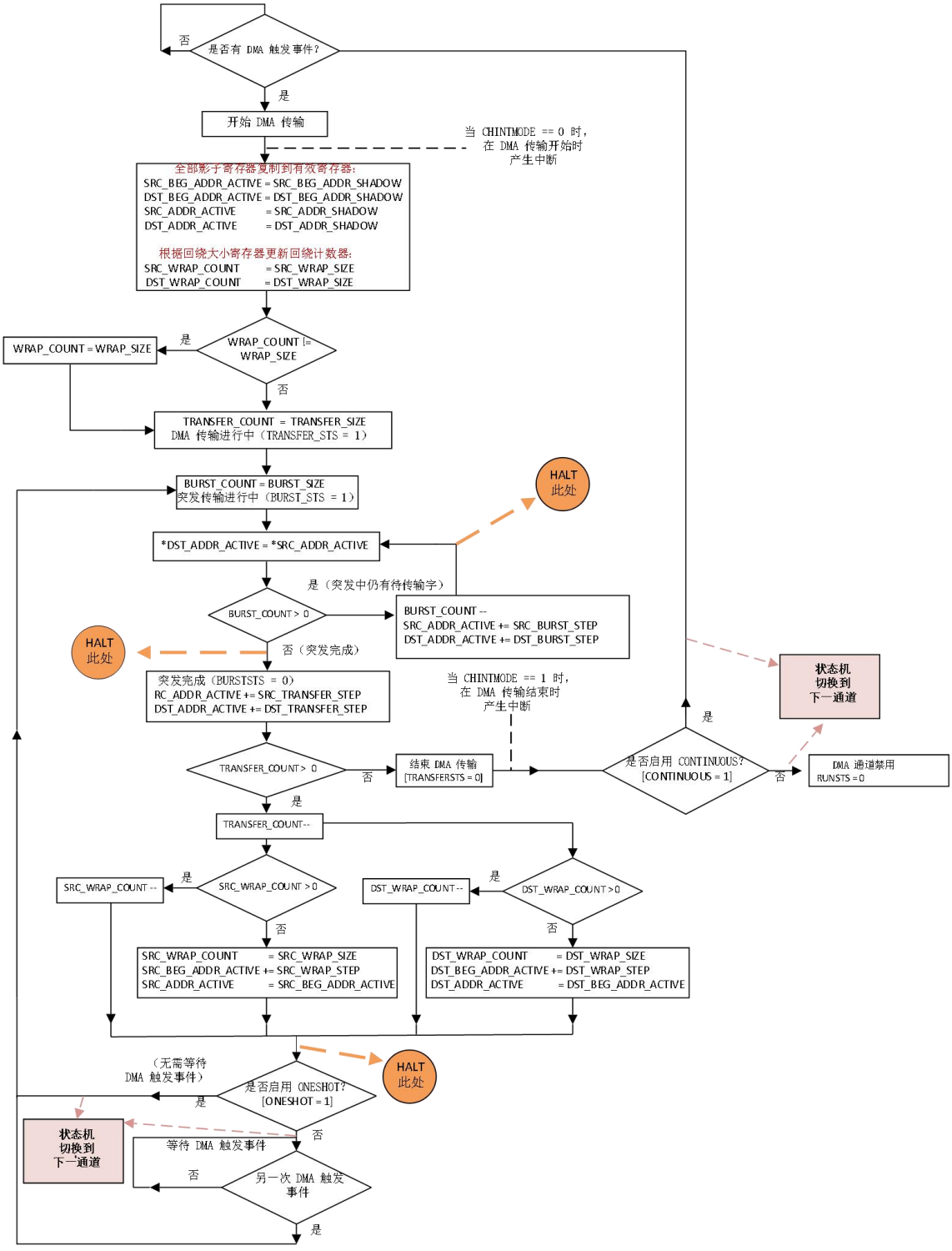

否
是否有 DMA 触发事件?
是
开始 DMA 传输
当 CHINTMODE == 0 时，
在 DMA 传输开始时
产生中断
全部影子寄存器复制到有效寄存器:
SRC_BEG_ADDR_ACTIVE = SRC_BEG_ADDR_SHADOW
DST_BEG_ADDR_ACTIVE = DST_BEG_ADDR_SHADOW
SRC_ADDR_ACTIVE = SRC_ADDR_SHADOW
DST_ADDR_ACTIVE = DST_ADDR_SHADOW
根据回绕大小寄存器更新回绕计数器:
SRC_WRAP_COUNT = SRC_WRAP_SIZE
DST_WRAP_COUNT = DST_WRAP_SIZE
是
WRAP_COUNT != WRAP_SIZE
WRAP_COUNT = WRAP_SIZE
否
TRANSFER_COUNT = TRANSFER_SIZE
DMA 传输进行中（TRANSFER_STS = 1）
BURST_COUNT = BURST_SIZE
突发传输进行中（BURST_STS = 1）
HALT
此处
*DST_ADDR_ACTIVE = *SRC_ADDR_ACTIVE
是（突发中仍有待传输字）
BURST_COUNT > 0
BURST_COUNT --
SRC_ADDR_ACTIVE += SRC_BURST_STEP
DST_ADDR_ACTIVE += DST_BURST_STEP
HALT
此处
否（突发完成）
突发完成（BURSTSTS = 0）
RC_ADDR_ACTIVE += SRC_TRANSFER_STEP
DST_ADDR_ACTIVE += DST_TRANSFER_STEP
当 CHINTMODE == 1 时，
在 DMA 传输结束时
产生中断
状态机
切换到
下一通道
是
TRANSFER_COUNT > 0
否
结束 DMA 传输
[TRANSFERSTS = 0]
是否启用 CONTINUOUS?
[CONTINUOUS = 1]
否
DMA 通道禁用
RUNSTS = 0
是
TRANSFER_COUNT--
SRC_WRAP_COUNT --
是
SRC_WRAP_COUNT > 0
否
SRC_WRAP_COUNT = SRC_WRAP_SIZE
SRC_BEG_ADDR_ACTIVE += SRC_WRAP_STEP
SRC_ADDR_ACTIVE = SRC_BEG_ADDR_ACTIVE
DST_WRAP_COUNT --
是
DST_WRAP_COUNT > 0
否
DST_WRAP_COUNT = DST_WRAP_SIZE
DST_BEG_ADDR_ACTIVE += DST_WRAP_STEP
DST_ADDR_ACTIVE = DST_BEG_ADDR_ACTIVE
（无需等待
DMA 触发事件）
是
是否启用 ONESHOT?
[ONESHOT = 1]
HALT
此处
否
状态机
切换到
下一通道
等待 DMA 触发事件
否
另一次 DMA 触发
事件
是

图 9-4 DMA 状态框图

#### 9.3.3.10 状态转换过程详细介绍

①首先启动 DMA 通道（RUNSTS=1），并且产生一次中断触发之后，DMA 开始一次传输。

- 首先把源和目的地址的影子寄存器的值复制到活动寄存器中（影子寄存器是可读写的，活动寄存器时只读的）。
- 对 transfer 进行初始化，TRANSFER_COUNT（外环传输计数寄存器）会从 TRANSFER_SIZE（外环传输长度寄存器）里面加载数据。表明一次传输中要进行多少次突发。
- 同时 WRAP_COUNT（回绕计数寄存器）会从 WRAP_SIZE（回绕长度寄存器中）里加载数据。
- 控制寄存器中的 TRANSFERSTS 传输状态标志位被置 1，表示 DMA 正在执行一次传输。

②检测通道中断产生模式选择位 CHINTMODE，来决定是在开始还是结束时向 CPU 发送一次中断。

如果 CHINTMODE=0，那么 DMA 通道会在传输开始之前向 CPU 发送一次中断（通过 CLIC），那么此时就会向 CPU 发送一次中断，然后再进行下一步。

如果 CHINTMODE=1，会在一次传送结束的时候发送中断，此时不发送中断，直接仅进行下一步。

③接着，对 burst 的寄存器进行初始化

- 突发传送长度寄存器 BURST_SIZE 的值会被加载到突发传送计数 BURST_COUNT 寄存器中。说明一次突发中要传送多少个字的数据。
- 并且将突发传送状态标志位 BURSTSTS 置 1，说明 DMA 正在执行或挂起一次突发传送。
- 将外设中断触发事件标志位 PERINTFLG 清零。

④然后进行 burst 传输，从活动源地址中 SRC_ADDR 中读取数据，然后向活动目的地址中写数据。（此处可暂停），这就完成了一个字的传输。

⑤每完成一个字的传输，BURST_COUNT 会减 1，同时源/目的地址都会按照突发步长寄存器 BURST_STEP 中的有符号数进行递增/递减，直到 BURST_COUNT 减到 0 时，说明一次突发传输完成，进行下一步。这个就是内循环（burst loop）。

⑥在一次 burst 传送结束后，源/目的地址会按照 TRANSFER_STEP 寄存器中的有符号值进行递增或递减。

⑦突发传送状态标志位 BURSTSTS 被清零，说明此时没有有效的突发传送。

⑧检测 TRANSFER_COUNT == 0?

- 如果 TRANSFER_COUNT 清零，表示在一次传输过程中剩余的突发传送次数为 0 次，说明 DMA 传输完成，传输状态标志位 TRANSFERSTS 会被清零。
- 如果 TRANSFER_COUNT 不为 0，说明还要继续进行突发传送，剩余的突发次数就是 TRANSFER_COUNT 中的值。

⑨当 TRANSFER_COUNT 不为 0 时，继续进行突发传输。

这时检测地址回绕计数器 WRAP_COUNT == 0? WRAP_COUNT 中的值表示在进行地址回绕之前剩余的突发传输次数。

如果 WRAP_COUNT == 0，则进行一次地址回绕：

- 源/目的起始地址指针会按照 WRAP_STEP 中的有符号数值进行递增或递减，从而得到下一次 burst 的起始地址。
- 并且将起始地址指针寄存器 BEG_ADDR 的值加载到活动地址指针寄存器 ADDR 中。
- 并且，回绕计数寄存器 WRAP_COUNT 会重新加载回绕长度寄存器 WRAP_SIZE 中的值，为下一次回绕做准备，在一次传输中允许发生多次回绕操作。

如果 WRAP_COUNT 不为 0，则 WRAP_COUNT 减 1。

WRAP_STEP 和 TRANSFER_STEP 都能改变地址，TRANSFER_STEP 是在 ADDR 的基础上进行增量，WRAP_STEP 是在 BEG_ADDR 的基础上进行增量，所以二者不能同时使用，二选一的关系，在 WRAP_STEP 生效时，TRANSFER_STEP 被忽略，WRAP_STEP 不生效时，使用 TRANSFER_STEP，即使是 WRAP 模式下。

如果要禁用地址回绕，将 WRAP_SIZE 设置得大于 TRANSFER_SIZE 即可。

最后，然后 TRANSFER_COUNT 减 1，说明完成了一次 burst 传输。

⑩检测 ONESHOT== 1?　即检测是否是单次触发模式。

- 如果 ONESHOT== 1 则是单次触发模式，一次中断触发信号可以完成所有的 burst 传送，然后跳转到步骤③
- 如果 ONESHOT=0，则每次突发传输都需要中断信号，需要等待下一次的中断触发信号才能开始新一次 burst，当有中断触发信号到来时，则跳转到步骤②。

⑪如果一次 DMA 传输完成，即 TRANSFER_COUNT==0，接着步骤⑧，则传输状态标志位 TRANSFERSTS 清零，表示当前没有有效 DMA 传输。

⑫然后再检测一次通道中断产生模式选择位 CHINTMODE==1?

- 如果 CHINTMODE==1，说明是在一次传输结束时产生中断事件，DMA 通道会在此时向 CPU 发送一次中断（通过 CLIC）。然后在跳转到下一步。
- 如果 CHINTMODE==0，说明是在一次传输开始时向 CPU 发送一次中断，那么说明已经在开始时发送过了，此处不用再发送。直接跳转到下一步。

⑬检测 CONTINUOUS== 1?

- 如果 CONTINUOUS==1，说明启用连续触发模式，会等待下一次的中断触发信号，然后跳转到步骤 1，重新初始化 DMA 通道。
- 如果 CONTINUOUS==0，说明禁止连续触发模式，那么在一次传输完成后，会将 CONTROL 寄存器中的运行状态标志位 RUNSTS 清零，从而禁止 DMA 通道工作。如果想要再发起一次新的传输，需要将通道运行位 RUN 置 1，来重新开启 DMA 通道。

跳转到步骤①

### 9.3.3.11 状态转换过程总结

简单来说就是：每个 DMA 传输对应一个 transfer，可以在 transfer 的开始或结束时向 CPU 发送中断。每个 transfer 会传输 M 个 burst。每个 burst 会传输 N 个字。每个 burst 完成后，地址会递增一个 burst 步长。每个 burst 完成后，地址会递增一个 transfer 步长。或者完成 K 个 burst（由 wrap_size 决定）后，可以发生地址回绕，新的地址为起始地址+wrap_step，然后在进行下一次 burst 传输。

## 9.3.4 DMA 的流水线和吞吐量介绍

### 9.3.4.1 DMA 模块的 3 级流水线机制

正常的 DMA 传输采用 3 级流水线，将数据传输主要分成如下阶段：

①输出当前的源地址，产生（N+1）的源地址

②读取数据

③输出目的地址，并发出目的数据，产生（N+1）的目的地址

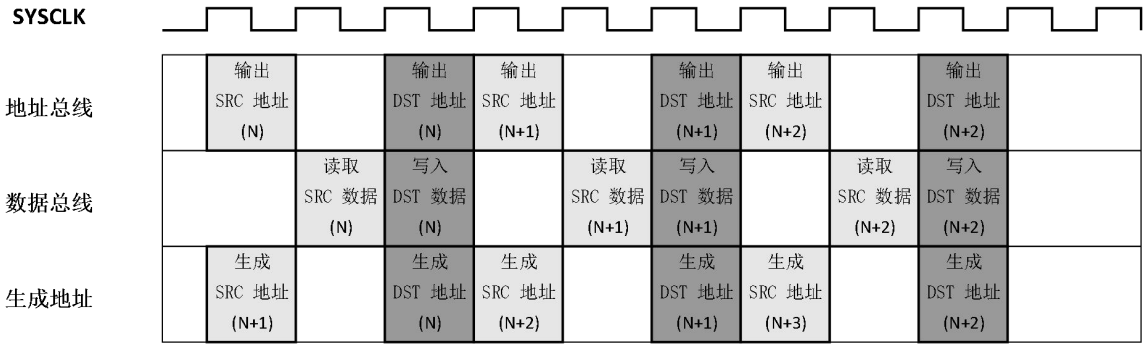



图 9-5 DMA 的 3 级流水线传输

有一种情况是比较特殊的:比如，当 DMA 通道被配置成使用 McBSPs 作为数据源时。在读取 McBSP DRR 寄存器时，读操作会增加一个时钟周期，如下图所示

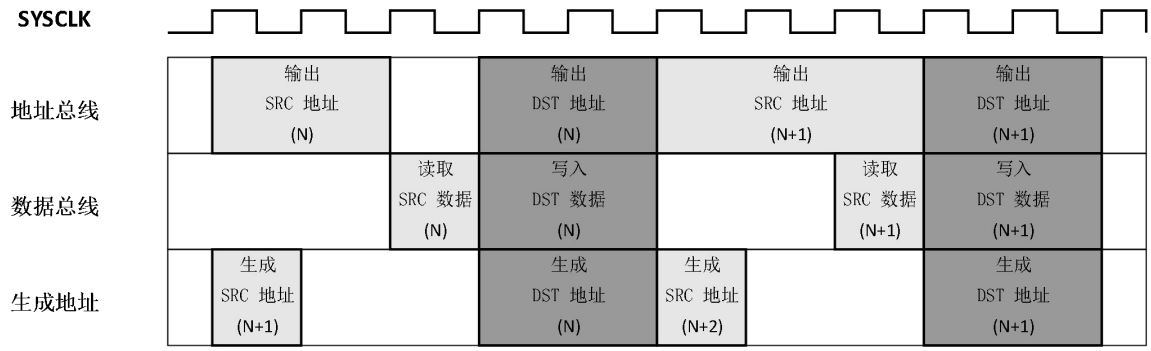



图 9-6 带读延时的 3 级流水线传输

### 9.3.4.2 DMA 通道的吞吐量介绍

会影响 DMA 通道吞吐量的操作，如下：

在每次突发传输之前，要增加一个周期的延迟

当从 CH1 高优先级中断返回时，会增加一个周期延时

与 CPU 的冲突可能会增加延迟。

32-bit 的传输速度是 16-bit 的传输速度的两倍，即传输 32 bit 字所需要的时间与传输 16 bit 字所需的时间相同。

例如，为了将 128 个 16 bit 字从 ADC 传输到 RAM，可以将 DMA 通道配置成 1 个 burst 传递 16 个 16 位的字，每个字要花 3 个时钟周期，8 次 burst 能传完，总共需要 392 个时钟周期：

8 bursts*[(3 cycles/word*16 words/burst)+1]=392 cycles

但是，如果将 DMA 通道配置成传递 32bit 的字，1 此 burst 传递 8 个 32 位的字，每个字花费 3 个时钟周期，一共需要 8 次 burst 传完，工作需要 200 个时钟周期。

8bursts*[(3 cycles/word*8 words/burst)+1]=200cycles

花费的时间差不多减半，也就是说对于 DMA 传输而言，采用 32 位数据传输能将吞吐量最大化，采用 16bit 字传输，则浪费了一半的吞吐量

## 9.3.5 DMA 通道优先级

有两种 DMA 通道的优先级方案：Round-Robin 轮转模式和通道 1 高优先级模式。

#### 9.3.5.1 Round-Robin 轮询模式

在这种模式下，所有通道具有相同的优先级，每一个通道采用“轮询”响应的形式，如下所示：

CH1→CH2→CH3→CH4→CH5→CH6→CH1→CH2→...

在这种情况下，每个通道传输了一个 burst 字后，下一个通道就会被响应。我们可以指定每个通道的突发字数量。一旦 CH6 或最后启用的通道得到了响应，并且没有其他通道处于挂起的状态，那么轮询模式状态机将进入空闲状态。

在空闲状态下，通道 1（如果启用的话）总是先被响应，但是，如果 DMA 正在处理另一个通道 x，则在 x 与最后通道之间的通道（X～6 之间），都会在通道 1 之前被响应。所以在一定意义上说，所有通道都具有相同的优先级。

例如，在轮询模式下使能了 CH1,CH4,CH5，并且 CH4 正在执行中，在 CH4 响应完成之前，CH1 和 CH5 都接到了来自各自外设的中断触发，此时 CH1 和 CH5 都处于挂起状态，当 CH4 完成突发传输时，CH5 将会被响应，只有等到 CH5 完成响应后，CH1 才会得到响应。CH1 完成后，如果没有挂起的通道了，伦次模式状态机将进入空闲状态。

更复杂的示例如下所示：

- 假设所有通道都使能，并且 DMA 处于空闲状态。
- 在同一周期内，CH1,CH3,CH5 同时被第一次触发。
- 首先响应 CH1， CH3 和 CH5 会处于挂起状态。
- 在 CH1 突发完成之前，DMA 又收到了一个来自 CH2 的请求，现在挂起的请求分别来自 CH2，CH3 和 CH5.
- 在完成 CH1 突发传输之后，CH2 将被响应，因为在轮询模式方案中，CH2 是 CH1 的下一个
- 完成 CH2 的突发传输后，将对 CH3 进行服务，然后是 CH5。
- 当正在进行 CH5 的突发传输时，DMA 又收到了来自 CH1，CH3 和 CH6 的请求

- 那么 CH5 突发传输完成后，将进行 CH6 的突发，因为在循环方案中，CH6 是 CH5 的下一个。
- 最后是 CH1 突发，然后是 CH3 突发
- 在 CH3 突发传输完成后，如果没有其他触发产生，那么轮询模式状态机将进入空闲状态。

轮询模式状态机可以通过 DMACTRL 寄存器的优先权重启位 DMACTRL[PRIORITYRESET]复位成空闲状态。

#### 9.3.5.2 channel1 高优先级模式

在这种模式下，CH1 具有高优先权。其他通道在突发传输时，如果发生 CH1 触发事件，则正在突发传输的通道完成当前字或者+1 字的传输后（取决于 CH1 触发发生时，DMA 传输处于哪个阶段），就被中止（没有完成整个突发传输），然后 CH1 得到响应，当 CH1 完成突发传输后，再返回到刚刚被停止执行的通道。除了 CH1 外，其他通道的优先级相同，其他各通道的优先级与轮询模式下的一样。

Higher Priority: CH1

Lower priority: CH2 → CH3 → CH4 → CH5 → CH6 → CH2 → …

例如，在 CH1 高优先级时，CH1，CH4 和 CH5 都处于启动状态，此时 CH4 正在被执行。在 CH4 完成突发之前，CH1 和 CH5 都收到了来自各自外设的中断触发信号，此时 CH1 和 CH5 都处于挂起状态。当 CH4 传送完当前字后，无论 DMA 是否完成整个 CH4 的突发传输，CH4 都会被暂停执行，然后 DMA 对 CH1 进行响应，当 CH1 的突发传输完成后，CH4 将恢复执行。当 CH4 传输完成后，CH5 将得到响应，当 CH5 传输完成后，如果没有其他通道处于挂起状态，轮询模式状态机将进入空闲状态。

通常 CH1 高优先级模式会在 ADC 采集数据中使用，因为 ADC 的数据速率太快。当然，CH1 高优先级模式也可以用在跟外围设备的连接中。

注意：CH1 高优先级模式和 ONESHOT（单次触发）模式不能同时在 CH1 上使用。但 CH1 处于高优先级模式时，其他通道上可以用 ONESHOT 模式。

### 9.3.6 DMA 中断触发溢出检测

DMA 有溢出检测逻辑。当 DMA 接收到一次外设中断触发信号时，CONTROL 寄存器中的 PERINTFLG 位会被置 1，并将相应的通道在 DMA 状态机中挂起。当该通道的突发传送开始时，PERINTFLG 标志位被清零。如果 PERINTFLG 标志位被置 1，并且在该通道的突发传送开始前，又有新的触发信号到来，则这个新的触发信号会丢失。并且将 CONTROL 寄存器中的溢出标志位 OVERFLG 置 1，如图 DMA 溢出检测逻辑所示。如果溢出中断被使能，那么该通道将向 CLIC 模块发出中断请求。

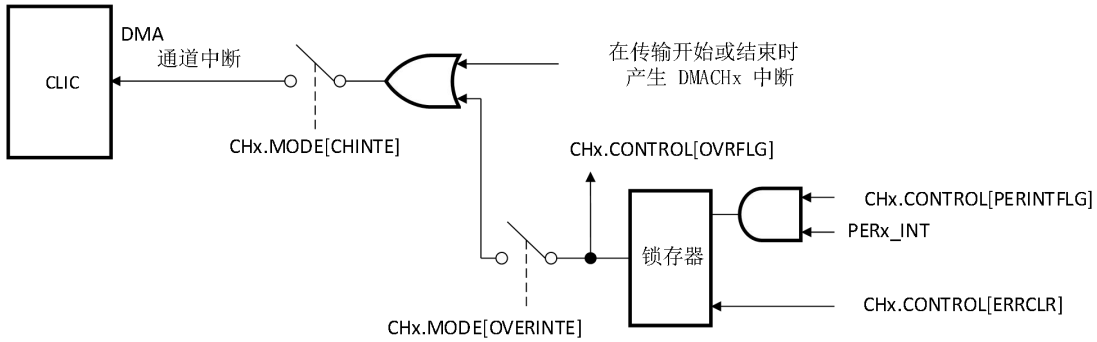



图 9-7 DMA 溢出检测逻辑

## 9.4 寄存器描述

### 9.4.1 DMA 基地址表

表 9-2 DMA 基地址表

| 位域名 | | DriverLib 命名 | 基地址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| 实例 | 结构体 | | | | | |
| DmaRegs | DMA_REGS | DMA_BASE | 0x0060_2000 | - | - | - |
| Dmach1Regs | DMA_CH_REGS | DMA_CH1_BASE | 0x0060_2040 | - | - | - |
| Dmach2Regs | DMA_CH_REGS | DMA_CH2_BASE | 0x0060_2080 | - | - | - |
| Dmach3Regs | DMA_CH_REGS | DMA_CH3_BASE | 0x0060_20C0 | - | - | - |
| Dmach4Regs | DMA_CH_REGS | DMA_CH4_BASE | 0x0060_2100 | - | - | - |
| Dmach5Regs | DMA_CH_REGS | DMA_CH5_BASE | 0x0060_2140 | - | - | - |
| Dmach6Regs | DMA_CH_REGS | DMA_CH6_BASE | 0x0060_2180 | - | - | - |

## 9.4.2 DMA_REGS 寄存器

表中列出了 DMA_REGS 寄存器组中所有采用存储器映射方式访问的寄存器。凡未在表中列出的寄存器偏移地址，均应视为保留地址，其寄存器内容不得进行任何读写或修改操作。

表 9-3 DMA_REGS 寄存器

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | DMACTRL | DMA 控制寄存器 |
| 2h | DEBUGCTRL | 调试控制寄存器 |
| 8h | PRIORITYCTRL1 | 优先级控制 1 寄存器 |
| Ch | PRIORITYSTAT | 优先级状态寄存器 |

为适配表格单元格空间限制，复杂的位访问类型采用编码形式进行表示。表中给出了本节中用于表示访问类型的各类编码说明。

表 9-4 DMA_REGS 访问类型编码

| 访问类型 | 代码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回0 |
| 写类型 | | |
| W | W | 写 |
| W1S | W<br>1S | 写<br>1表示设置 |
| 复位或默认值 | | |
| -n | | 复位后的值或默认值 |

### 9.4.2.1 DMACTRL 寄存器

偏移地址：0h　复位值：0000h

DMA 控制寄存器

表 9-5 DMACTRL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | PRIORITYRESET | HARDRESET |
| R-0h | | | | | | R-0/W1S-0h | R-0/W1S-0h |

表 9-6 DMACTRL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-2 | RESERVED | R | 0h | 保留。 |
| 1 | PRIORITYRESET | R-0/W1S | 0h | 优先级复位位。写入 1 复位轮询状态机，服务从第一个已使能通道开始。忽略写入 0，读取始终返回 0。<br>写入 1 后，所有待处理的突发传输完成后才复位通道优先级状态机。如果 CH1 配置为高优先级通道，且写入此位时 CH1 正在执行突发传输，则 CH1 的突发传输和下一个待处理的低优先级突发传输均完成后，状态机才复位。<br>若 CH1 为高优先级，状态机从 CH2（或下一个已使能的最高优先级通道）重新开始。<br>复位类型：SYSRSn。 |
| 0 | HARDRESET | R-0/W1S | 0h | 硬复位位。写入 1 复位整个 DMA，并中止当前访问，效果类似于器件复位。忽略写入 0，读取始终返回 0。<br>每个通道还提供一个软复位位，可执行更温和的复位；请参阅通道控制寄存器。<br>写入此位后延迟一个周期生效。因此，软件在访问任何其他 DMA 寄存器前，必须等待一个周期，例如执行一条 NOP 指令。<br>复位类型：SYSRSn。 |

### 9.4.2.2 DEBUGCTRL 寄存器

偏移地址：2h　复位值：0000h

调试控制寄存器

表 9-7 DEBUGCTRL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FREE | 保留 | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |

表 9-8 DEBUGCTRL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | FREE | R/W | 0h | 仿真控制。指定发生仿真暂停事件时的行为。<br>0h (R/W)：DMA 完成当前读写操作后暂停。<br>1h (R/W)：仿真暂停期间 DMA 继续运行。<br>复位类型：SYSRSn。 |
| 14-0 | RESERVED | R | 0h | 保留。 |

### 9.4.2.3 PRIORITYCTRL1 寄存器

偏移地址：8h 复位值：0000h

优先级控制 1 寄存器

表 9-9 PRIORITYCTRL1 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | CH1PRIORITY |
| R-0h | | | | | | | R/W-0h |

表 9-10 PRIORITYCTRL1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-1 | RESERVED | R | 0h | 保留。 |
| 0 | CH1PRIORITY | R/W | 0h | DMA 通道 1 优先级。选择 CH1 是否具有高优先级。只有所有通道均禁用时才可更改优先级。更改后，应先执行优先级复位，再重新启动通道。<br>0h (R/W)：CH1 与其他通道优先级相同。<br>1h (R/W)：CH1 优先级高于其他通道。<br>复位类型：SYSRSn。 |

### 9.4.2.4 PRIORITYSTAT 寄存器

偏移地址：Ch　复位值：0000h

优先级状态寄存器

表 9-11 PRIORITYSTAT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | ACTIVESTS_SHADOW | | | 保留 | ACTIVESTS | | |
| R-0h | R-0h | | | R-0h | R-0h | | |

表 9-12 PRIORITYSTAT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-7 | RESERVED | R | 0h | 保留。 |
| 6-4 | ACTIVESTS_SHADOW | R | 0h | 活动通道状态影子字段。仅当 CH1 处于高优先级模式时有意义。服务 CH1 时，将 ACTIVESTS 位复制到影子位，以指示被 CH1 中断的通道。CH1 服务完成后，将影子位复制回 ACTIVESTS。若此字段为零或与 ACTIVESTS 相同，则没有因 CH1 中断而等待的通道。CH1 不是高优先级通道时，应忽略这些位。<br>0h (R/W)：无活动通道。<br>1h (R/W)：CH 1。<br>2h (R/W)：CH 2。<br>3h (R/W)：CH 3。<br>4h (R/W)：CH 4。<br>5h (R/W)：CH 5。<br>6h (R/W)：CH 6。<br>7h (R/W)：保留。<br>复位类型：SYSRSn。 |
| 3 | RESERVED | R | 0h | 保留。 |
| 2-0 | ACTIVESTS | R | 0h | 活动通道状态。指示当前活动或正在执行传输的通道（如果有）。<br>0h (R/W)：无活动通道。<br>1h (R/W)：CH 1。<br>2h (R/W)：CH 2。<br>3h (R/W)：CH 3。<br>4h (R/W)：CH 4。<br>5h (R/W)：CH 5。<br>6h (R/W)：CH 6。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 7h (R/W)：保留。<br>复位类型：SYSRSn。 |

### 9.4.3 DMA_CH_REGS 寄存器

表中列出了存储器映射寄存器。未列出的偏移地址应视为保留位置，不应修改这些位置的寄存器内容。

表 9-13 DMA_CH_REGS 寄存器

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 00h | MODE | 模式寄存器 |
| 02h | CONTROL | 控制寄存器 |
| 04h | BURST_SIZE | Burst 尺寸寄存器 |
| 06h | BURST_COUNT | Burst 计数寄存器 |
| 08h | SRC_BURST_STEP | 源地址 Burst 步长寄存器 |
| 0Ah | DST_BURST_STEP | 目的地址 Burst 步长寄存器 |
| 0Ch | TRANSFER_SIZE | Transfer 尺寸寄存器 |
| 0Eh | TRANSFER_COUNT | Transfer 计数寄存器 |
| 10h | SRC_TRANSFER_STEP | 源地址 Transfer 步长寄存器 |
| 12h | DST_TRANSFER_STEP | 目的地址 Transfer 步长寄存器 |
| 14h | SRC_WRAP_SIZE | 源地址 Wrap 尺寸寄存器 |
| 16h | SRC_WRAP_COUNT | 源地址 Wrap 计数寄存器 |
| 18h | SRC_WRAP_STEP | 源地址 Wrap 步长寄存器 |
| 1Ah | DST_WRAP_SIZE | 目的地址 Wrap 尺寸寄存器 |
| 1Ch | DST_WRAP_COUNT | 目的地址 Wrap 计数寄存器 |
| 1Eh | DST_WRAP_STEP | 目的地址 Wrap 步长寄存器 |
| 20h | SRC_BEG_ADDR_SHADOW | 源起始地址影子寄存器 |
| 24h | SRC_ADDR_SHADOW | 源地址影子寄存器 |
| 28h | SRC_BEG_ADDR_ACTIVE | 源起始地址活动寄存器 |
| 2Ch | SRC_ADDR_ACTIVE | 源地址活动寄存器 |
| 30h | DST_BEG_ADDR_SHADOW | 目的起始地址影子寄存器 |
| 34h | DST_ADDR_SHADOW | 目的地址影子寄存器 |

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 38h | DST_BEG_ADDR_ACTIVE | 目的起始地址活动寄存器 |
| 3Ch | DST_ADDR_ACTIVE | 目的地址活动寄存器 |

表 9-14 DMA_CH_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1S | W<br>1S | 写<br>1置位 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |

### 9.4.3.1 MODE 寄存器

偏移地址：0h　复位值：0000h

模式寄存器

表 9-15 MODE 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CHINTE | DATASIZE | 保留 | 保留 | CONTINUOUS | ONESHOT | CHINTMODE | PERINTE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OVRINTE | 保留 | | PERINTSEL | | | | |
| R/W-0h | R-0h | | R/W-0h | | | | |

表 9-16 MODE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | CHINTE | R/W | 0h | 通道中断使能位。使能 DMA 通道的 CPU 中断。<br>0h (R/W)：禁用中断。<br>1h (R/W)：使能中断。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 14 | DATASIZE | R/W | 0h | 数据大小模式位。决定 DMA 通道每次读写操作传输 16 位还是 32 位数据。无论如何设置，其他 DMA 寄存器中的数据长度和偏移量均以 16 位字为单位。指针步进增量必须配置为适合 32 位字。<br>0h (R/W)：16 位数据传输。<br>1h (R/W)：32 位数据传输。<br>复位类型：SYSRSn。 |
| 13 | RESERVED | R/W | 0h | 保留。 |
| 12 | RESERVED | R/W | 0h | 保留。 |
| 11 | CONTINUOUS | R/W | 0h | 连续模式位。置 1 时，TRANSFER_COUNT 为零后，通道重新初始化并等待下一次事件触发。否则，DMA 停止并清除 RUNSTS 位。<br>复位类型：SYSRSn。 |
| 10 | ONESHOT | R/W | 0h | 单次触发模式。置 1 时，每次外设事件触发均使通道执行一次完整传输；否则，每次触发仅执行一次突发传输。<br>复位类型：SYSRSn。 |
| 9 | CHINTMODE | R/W | 0h | 通道中断生成模式。指定 DMA 通道在传输的哪个时刻产生 CPU 中断。<br>0h (R/W)：新传输开始时产生中断。<br>1h (R/W)：传输结束时产生中断。<br>复位类型：SYSRSn。 |
| 8 | PERINTE | R/W | 0h | 外设事件触发使能。使能 DMA 通道的外设事件触发。<br>0h (R/W)：禁用外设事件触发，所选外设和软件均不能启动 DMA 突发传输。<br>1h (R/W)：使能外设事件触发。<br>复位类型：SYSRSn。 |
| 7 | OVRINTE | R/W | 0h | 溢出中断使能。决定 DMA 模块检测到溢出事件时是否产生 CPU 中断。<br>0h (R/W)：禁用溢出中断。<br>1h (R/W)：使能溢出中断。<br>复位类型：SYSRSn。 |
| 6-5 | RESERVED | R | 0h | 保留。 |
| 4-0 | PERINTSEL | R/W | 0h | 外设事件触发源选择。这些为兼容旧设计保留的位，应设置为通道编号。实际触发源通过 DMA_CLA_SRC_SEL_REGS 组中的 DMACHSRCSELn 寄存器选择。<br>复位类型：SYSRSn。 |

### 9.4.3.2 CONTROL 寄存器

偏移地址：2h　复位值：0000h

控制寄存器

表 9-17 CONTROL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | OVRFLG | RUNSTS | BURSTSTS | TRANSFERSTS | 保留 | 保留 | PERINTFLG |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERRCLR | 保留 | 保留 | PERINTCLR | PERINTFRC | SOFTRESET | HALT | RUN |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 9-18 CONTROL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留。 |
| 14 | OVRFLG | R | 0h | 溢出标志。表示 PERINTFLG 已置位时又收到外设事件触发。可通过写入 ERRCLR 位清除此标志。<br>0h (R/W)：未检测到溢出。<br>1h (R/W)：检测到溢出。<br>复位类型：SYSRSn。 |
| 13 | RUNSTS | R | 0h | 运行状态标志。表示 DMA 通道已准备好响应外设事件触发。向 RUN 位写入 1 时置位。传输完成（TRANSFER_COUNT = 0）且未使能连续模式时清零；HARDRESET、SOFTRESET 或 HALT 位置位时也清零。<br>0h (R/W)：通道已禁用。<br>1h (R/W)：通道已使能。<br>复位类型：SYSRSn。 |
| 12 | BURSTSTS | R | 0h | 突发状态标志。DMA 突发传输开始时置位，同时将 BURST_COUNT 设为 BURST_SIZE。BURST_COUNT 达到零，或 HARDRESET / SOFTRESET 位置位时清零。<br>0h (R/W)：无突发传输活动。<br>1h (R/W)：DMA 当前正在服务或挂起此通道的突发传输。<br>复位类型：SYSRSn。 |
| 11 | TRANSFERSTS | R | 0h | 传输状态标志。DMA 传输开始时置位，将地址寄存器复制到影子寄存器组，并将 TRANSFER_COUNT 设为 TRANSFER_SIZE。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | TRANSFER_COUNT 达到零，或 HARDRESET / SOFTRESET 位置位时清零。<br>0h (R/W)：无传输活动。<br>1h (R/W)：通道处于一次传输过程中，无论当前是否正在传输一个数据突发。<br>复位类型：SYSRSn。 |
| 10 | RESERVED | R | 0h | 保留。 |
| 9 | RESERVED | R | 0h | 保留。 |
| 8 | PERINTFLG | R | 0h | 外设事件触发标志。表示外设事件触发是否已到达。第一次突发传输开始时自动清零。<br>0h (R/W)：等待事件触发。<br>1h (R/W)：有待处理的事件触发。<br>复位类型：SYSRSn。 |
| 7 | ERRCLR | R-0/W1S | 0h | 清除错误。写入 1 清除 OVRFLG，通常在初始化 DMA 模块或检测到溢出时执行。若设置此位的同时发生溢出事件，则溢出优先，OVRFLG 被置位。<br>复位类型：SYSRSn。 |
| 6 | RESERVED | R-0/W1S | 0h | 保留。 |
| 5 | RESERVED | R-0/W1S | 0h | 保留。 |
| 4 | PERINTCLR | R-0/W1S | 0h | 清除外设事件触发。写入 1 清除 PERINTFLG，从而取消待处理的事件触发，通常在初始化 DMA 模块时执行。若设置此位的同时事件触发到达，则触发优先，PERINTFLG 被置位。<br>复位类型：SYSRSn。 |
| 3 | PERINTFRC | R-0/W1S | 0h | 强制外设事件触发。如果 MODE 寄存器的 PERINTE 已置位，向此位写入 1 将置位 PERINTFLG，触发 DMA 突发传输。可用此位通过软件启动 DMA 传输。<br>复位类型：SYSRSn。 |
| 2 | SOFTRESET | R-0/W1S | 0h | 通道软复位。写入 1 后，在当前读写访问完成时将通道置于默认状态：<br>RUNSTS = 0<br>TRANSFERSTS = 0<br>BURSTSTS = 0<br>BURST_COUNT = 0<br>TRANSFER_COUNT = 0<br>SRC_WRAP_COUNT = 0<br>DST_WRAP_COUNT = 0<br>写入此位后延迟一个周期生效。因此，软件在访问任何其他 DMA 寄存器前，必须等待一个周 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 期，例如执行一条 NOP 指令。<br>复位类型：SYSRSn。 |
| 1 | HALT | R-0/W1S | 0h | 暂停通道。写入 1 后，在当前读写访问完成时，DMA 通道保持当前状态并暂停。<br>复位类型：SYSRSn。 |
| 0 | RUN | R-0/W1S | 0h | 运行通道。写入 1 使能 DMA 通道，并将 RUNSTS 置 1。此位也用于恢复已暂停的通道。<br>通常在完成配置后，通过 RUN 位启动 DMA 通道。通道随后等待第一次外设事件触发（PERINTFLG == 1），以启动突发传输。<br>复位类型：SYSRSn。 |

### 9.4.3.3 BURST_SIZE 寄存器

偏移地址：4h　复位值：0000h

突发大小寄存器

表 9-19 BURST_SIZE 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | BURSTSIZE | | | | |
| R-0h | | | R/W-0h | | | | |

表 9-20 BURST_SIZE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-5 | RESERVED | R | 0h | 保留。 |
| 4-0 | BURSTSIZE | R/W | 0h | 指定突发传输大小，以 16 位字为单位。实际大小等于 BURSTSIZE + 1。<br>复位类型：SYSRSn。 |

### 9.4.3.4 BURST_COUNT 寄存器

偏移地址：6h　复位值：0000h

突发计数寄存器

表 9-21 BURST_COUNT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | BURSTCOUNT | | | | |
| R-0h | | | R-0h | | | | |

表 9-22 BURST_COUNT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-5 | RESERVED | R | 0h | 保留。 |
| 4-0 | BURSTCOUNT | R | 0h | 表示当前突发传输中剩余的字数。<br>0h (R/W)：突发传输剩余 0 个字。<br>1h (R/W)：突发传输剩余 1 个字。<br>2h (R/W)：突发传输剩余 2 个字。<br>3h (R/W)：突发传输剩余 3 个字。<br>4h (R/W)：突发传输剩余 4 个字。<br>5h (R/W)：突发传输剩余 5 个字。<br>6h (R/W)：突发传输剩余 6 个字。<br>7h (R/W)：突发传输剩余 7 个字。<br>8h (R/W)：突发传输剩余 8 个字。<br>9h (R/W)：突发传输剩余 9 个字。<br>Ah (R/W)：突发传输剩余 10 个字。<br>Bh (R/W)：突发传输剩余 11 个字。<br>Ch (R/W)：突发传输剩余 12 个字。<br>Dh (R/W)：突发传输剩余 13 个字。<br>Eh (R/W)：突发传输剩余 14 个字。<br>Fh (R/W)：突发传输剩余 15 个字。<br>10h (R/W)：突发传输剩余 16 个字。<br>11h (R/W)：突发传输剩余 17 个字。<br>12h (R/W)：突发传输剩余 18 个字。<br>13h (R/W)：突发传输剩余 19 个字。<br>14h (R/W)：突发传输剩余 20 个字。<br>15h (R/W)：突发传输剩余 21 个字。<br>16h (R/W)：突发传输剩余 22 个字。<br>17h (R/W)：突发传输剩余 23 个字。<br>18h (R/W)：突发传输剩余 24 个字。<br>19h (R/W)：突发传输剩余 25 个字。<br>1Ah (R/W)：突发传输剩余 26 个字。<br>1Bh (R/W)：突发传输剩余 27 个字。<br>1Ch (R/W)：突发传输剩余 28 个字。<br>1Dh (R/W)：突发传输剩余 29 个字。<br>1Eh (R/W)：突发传输剩余 30 个字。<br>1Fh (R/W)：突发传输剩余 31 个字。<br>复位类型：SYSRSn。 |

### 9.4.3.5 SRC_BURST_STEP 寄存器

偏移地址：8h　复位值：0000h

源突发步长寄存器

表 9-23 SRC_BURST_STEP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SRCBURSTSTEP | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SRCBURSTSTEP | | | | | | | |
| R/W-0h | | | | | | | |

表 9-24 SRC_BURST_STEP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | SRCBURSTSTEP | R/W | 0h | 指定突发传输中每个字传输后的源地址变化量。该值必须为 -4096 至 4095（含端点）之间的 16 位二进制补码。突发传输中的每次读写操作之后，将此值加到源地址。<br>0h (R/W)：地址不变。<br>1h (R/W)：地址加 1。<br>2h (R/W)：地址加 2。<br>FFEh (R/W)：地址加 4094。<br>FFFh (R/W)：地址加 4095。<br>F000h (R/W)：地址减 4096。<br>F001h (R/W)：地址减 4095。<br>FFFEh (R/W)：地址减 2。<br>FFFFh (R/W)：地址减 1。<br>复位类型：SYSRSn。 |

### 9.4.3.6 DST_BURST_STEP 寄存器

偏移地址：Ah　复位值：0000h

目的突发步长寄存器

表 9-25 DST_BURST_STEP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DSTBURSTSTEP | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DSTBURSTSTEP | | | | | | | |
| R/W-0h | | | | | | | |

表 9-26 DST_BURST_STEP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | DSTBURSTSTEP | R/W | 0h | 指定突发传输中每个字传输后的目的地址变化量。该值必须为 -4096 至 4095（含端点）之间的 16 位二进制补码。突发传输中的每次读写操作之后，将此值加到目的地址。<br>0h (R/W)：地址不变。<br>1h (R/W)：地址加 1。<br>2h (R/W)：地址加 2。<br>FFEh (R/W)：地址加 4094。<br>FFFh (R/W)：地址加 4095。<br>F000h (R/W)：地址减 4096。<br>F001h (R/W)：地址减 4095。<br>FFFEh (R/W)：地址减 2。<br>FFFFh (R/W)：地址减 1。<br>复位类型：SYSRSn。 |

### 9.4.3.7 TRANSFER_SIZE 寄存器

偏移地址：Ch　复位值：0000h

传输大小寄存器

表 9-27 TRANSFER_SIZE 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| TRANSFERSIZE | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TRANSFERSIZE | | | | | | | |
| R/W-0h | | | | | | | |

表 9-28 TRANSFER_SIZE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | TRANSFERSIZE | R/W | 0h | 指定传输大小，以突发传输次数为单位。实际大小等于 TRANSFERSIZE + 1。<br>复位类型：SYSRSn。 |

### 9.4.3.8 TRANSFER_COUNT 寄存器

偏移地址：Eh　复位值：0000h

传输计数寄存器

表 9-29 TRANSFER_COUNT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| TRANSFERCOUNT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TRANSFERCOUNT | | | | | | | |
| R-0h | | | | | | | |

表 9-30 TRANSFER_COUNT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | TRANSFERCOUNT | R | 0h | 表示当前传输中剩余的突发传输次数。<br>复位类型：SYSRSn。 |

## 9.4.3.9 SRC_TRANSFER_STEP 寄存器

偏移地址：10h　复位值：0000h

源传输步长寄存器

表 9-31 SRC_TRANSFER_STEP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SRCTRANSFERSTEP | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SRCTRANSFERSTEP | | | | | | | |
| R/W-0h | | | | | | | |

表 9-32 SRC_TRANSFER_STEP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | SRCTRANSFERSTEP | R/W | 0h | 指定每次突发传输完成后的源地址变化量。该值必须为 -4096 至 4095（含端点）之间的 16 位二进制补码。每次突发传输完成后，将此值加到源地址。<br>0h (R/W)：地址不变。<br>1h (R/W)：地址加 1。<br>2h (R/W)：地址加 2。<br>FFEh (R/W)：地址加 4094。<br>FFFh (R/W)：地址加 4095。<br>F000h (R/W)：地址减 4096。<br>F001h (R/W)：地址减 4095。<br>FFFEh (R/W)：地址减 2。<br>FFFFh (R/W)：地址减 1。<br>复位类型：SYSRSn。 |

### 9.4.3.10 DST_TRANSFER_STEP 寄存器

偏移地址：12h　复位值：0000h

目的传输步长寄存器

表 9-33 DST_TRANSFER_STEP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DSTTRANSFERSTEP | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DSTTRANSFERSTEP | | | | | | | |
| R/W-0h | | | | | | | |

表 9-34 DST_TRANSFER_STEP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | DSTTRANSFERSTEP | R/W | 0h | 指定每次突发传输完成后的目的地址变化量。该值必须为 -4096 至 4095（含端点）之间的 16 位二进制补码。每次突发传输完成后，将此值加到目的地址。<br>0h (R/W)：地址不变。<br>1h (R/W)：地址加 1。<br>2h (R/W)：地址加 2。<br>FFEh (R/W)：地址加 4094。<br>FFFh (R/W)：地址加 4095。<br>F000h (R/W)：地址减 4096。<br>F001h (R/W)：地址减 4095。<br>FFFEh (R/W)：地址减 2。<br>FFFFh (R/W)：地址减 1。<br>复位类型：SYSRSn。 |

### 9.4.3.11 SRC_WRAP_SIZE 寄存器

偏移地址：14h　复位值：FFFFh

源回绕大小寄存器

表 9-35 SRC_WRAP_SIZE 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| WRAPSIZE | | | | | | | |
| R/W-FFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WRAPSIZE | | | | | | | |
| R/W-FFFFh | | | | | | | |

表 9-36 SRC_WRAP_SIZE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | WRAPSIZE | R/W | FFFFh | 指定源地址回绕到起始地址之前的突发传输次数。实际次数等于 WRAPSIZE + 1。要禁用回绕功能，应将 WRAPSIZE 设置为大于 TRANSFERSIZE 的值。<br>复位类型：SYSRSn。 |

### 9.4.3.12 SRC_WRAP_COUNT 寄存器

偏移地址：16h　复位值：0000h

源回绕计数寄存器

表 9-37 SRC_WRAP_COUNT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| WRAPSIZE | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WRAPSIZE | | | | | | | |
| R-0h | | | | | | | |

表 9-38 SRC_WRAP_COUNT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | WRAPSIZE | R | 0h | 表示源地址回绕之前剩余的突发传输次数。<br>复位类型：SYSRSn。 |

### 9.4.3.13 SRC_WRAP_STEP 寄存器

偏移地址：18h　复位值：0000h

源回绕步长寄存器

表 9-39 SRC_WRAP_STEP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| WRAPSTEP | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WRAPSTEP | | | | | | | |
| R/W-0h | | | | | | | |

表 9-40 SRC_WRAP_STEP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | WRAPSTEP | R/W | 0h | 指定回绕计数器达到零时，源起始地址的变化量。该值必须为 -4096 至 4095（含端点）之间的 16 位二进制补码。发生回绕时，将此值加到源地址。<br>0h (R/W)：地址不变。<br>1h (R/W)：地址加 1。<br>2h (R/W)：地址加 2。<br>FFEh (R/W)：地址加 4094。<br>FFFh (R/W)：地址加 4095。<br>F000h (R/W)：地址减 4096。<br>F001h (R/W)：地址减 4095。<br>FFFEh (R/W)：地址减 2。<br>FFFFh (R/W)：地址减 1。<br>复位类型：SYSRSn。 |

### 9.4.3.14 DST_WRAP_SIZE 寄存器

偏移地址：1Ah　复位值：FFFFh

目的回绕大小寄存器

表 9-41 DST_WRAP_SIZE 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| WRAPSIZE | | | | | | | |
| R/W-FFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WRAPSIZE | | | | | | | |
| R/W-FFFFh | | | | | | | |

表 9-42 DST_WRAP_SIZE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | WRAPSIZE | R/W | FFFFh | 指定目的地址回绕到起始地址之前的突发传输次数。实际次数等于 WRAPSIZE + 1。要禁用回绕功能，应将 WRAPSIZE 设置为大于 TRANSFERSIZE 的值。<br>复位类型：SYSRSn。 |

### 9.4.3.15 DST_WRAP_COUNT 寄存器

偏移地址：1Ch　复位值：0000h

目的回绕计数寄存器

表 9-43 DST_WRAP_COUNT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| WRAPSIZE | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WRAPSIZE | | | | | | | |
| R-0h | | | | | | | |

表 9-44 DST_WRAP_COUNT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | WRAPSIZE | R | 0h | 表示目的地址回绕之前剩余的突发传输次数。<br>复位类型：SYSRSn。 |

## 9.4.3.16 DST_WRAP_STEP 寄存器

偏移地址：1Eh　复位值：0000h

目的回绕步长寄存器

表 9-45 DST_WRAP_STEP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| WRAPSTEP | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WRAPSTEP | | | | | | | |
| R/W-0h | | | | | | | |

表 9-46 DST_WRAP_STEP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | WRAPSTEP | R/W | 0h | 指定回绕计数器达到零时，目的起始地址的变化量。该值必须为 -4096 至 4095（含端点）之间的 16 位二进制补码。发生回绕时，将此值加到目的地址。<br>0h (R/W)：地址不变。<br>1h (R/W)：地址加 1。<br>2h (R/W)：地址加 2。<br>FFEh (R/W)：地址加 4094。<br>FFFh (R/W)：地址加 4095。<br>F000h (R/W)：地址减 4096。<br>F001h (R/W)：地址减 4095。<br>FFFEh (R/W)：地址减 2。<br>FFFFh (R/W)：地址减 1。<br>复位类型：SYSRSn。 |

### 9.4.3.17 SRC_BEG_ADDR_SHADOW 寄存器

偏移地址：20h　复位值：00000000h

源起始地址影子寄存器

表 9-47 SRC_BEG_ADDR_SHADOW 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BEGADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BEGADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BEGADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BEGADDR | | | | | | | |
| R/W-0h | | | | | | | |

表 9-48 SRC_BEG_ADDR_SHADOW 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | BEGADDR | R/W | 0h | 源起始地址影子寄存器。<br>传输开始时，将此寄存器中的值加载到 SRC_BEG_ADDR_ACTIVE，用作源地址的起始值。传输期间可以安全更新此寄存器。<br>复位类型：SYSRSn。 |

### 9.4.3.18 SRC_ADDR_SHADOW 寄存器

偏移地址：24h　复位值：00000000h

源地址影子寄存器

表 9-49 SRC_ADDR_SHADOW 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDR | | | | | | | |
| R/W-0h | | | | | | | |

表 9-50 SRC_ADDR_SHADOW 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | ADDR | R/W | 0h | 源地址影子寄存器。<br>传输开始时，将此寄存器中的值加载到 SRC_ADDR_ACTIVE，用作源地址的值。传输期间可以安全更新此寄存器。<br>复位类型：SYSRSn。 |

## 9.4.3.19 SRC_BEG_ADDR_ACTIVE 寄存器

偏移地址：28h　复位值：00000000h

源起始地址活动寄存器

表 9-51 SRC_BEG_ADDR_ACTIVE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BEGADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BEGADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BEGADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BEGADDR | | | | | | | |
| R-0h | | | | | | | |

表 9-52 SRC_BEG_ADDR_ACTIVE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | BEGADDR | R | 0h | 源起始地址活动寄存器。<br>传输进行期间，本寄存器保存当前源地址的起始值。该地址可在回绕后更新。<br>传输开始时，从 SRC_BEG_ADDR_SHADOW 寄存器加载影子地址。<br>复位类型：SYSRSn。 |

### 9.4.3.20 SRC_ADDR_ACTIVE 寄存器

偏移地址：2Ch　复位值：00000000h

源地址活动寄存器

表 9-53 SRC_ADDR_ACTIVE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDR | | | | | | | |
| R-0h | | | | | | | |

表 9-54 SRC_ADDR_ACTIVE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | ADDR | R | 0h | 源地址活动寄存器。<br>传输进行期间，本寄存器保存源地址的当前值。该地址可在一次写操作、一次突发传输或回绕后改变。<br>复位类型：SYSRSn。 |

### 9.4.3.21 DST_BEG_ADDR_SHADOW 寄存器

偏移地址：30h　复位值：00000000h

目的起始地址影子寄存器

表 9-55 DST_BEG_ADDR_SHADOW 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BEGADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BEGADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BEGADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BEGADDR | | | | | | | |
| R/W-0h | | | | | | | |

表 9-56 DST_BEG_ADDR_SHADOW 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | BEGADDR | R/W | 0h | 目的起始地址影子寄存器。<br>传输开始时，将此寄存器中的值加载到 DST_BEG_ADDR_ACTIVE，用作目的地址的起始值。传输期间可以安全更新此寄存器。<br>复位类型：SYSRSn。 |

## 9.4.3.22 DST_ADDR_SHADOW 寄存器

偏移地址：34h　复位值：00000000h

目的地址影子寄存器

表 9-57 DST_ADDR_SHADOW 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDR | | | | | | | |
| R/W-0h | | | | | | | |

表 9-58 DST_ADDR_SHADOW 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | ADDR | R/W | 0h | 目的地址影子寄存器。<br>传输开始时，将此寄存器中的值加载到 DST_ADDR_ACTIVE，用作目的地址的值。传输期间可以安全更新此寄存器。<br>复位类型：SYSRSn。 |

### 9.4.3.23 DST_BEG_ADDR_ACTIVE 寄存器

偏移地址：38h　复位值：00000000h

目的起始地址活动寄存器

表 9-59 DST_BEG_ADDR_ACTIVE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BEGADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BEGADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BEGADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BEGADDR | | | | | | | |
| R-0h | | | | | | | |

表 9-60 DST_BEG_ADDR_ACTIVE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | BEGADDR | R | 0h | 目的起始地址活动寄存器。<br>传输进行期间，本寄存器保存源地址的当前目的值。该地址可在回绕后更新。<br>传输开始时，从 DST_BEG_ADDR_SHADOW 寄存器加载影子地址。<br>复位类型：SYSRSn。 |

### 9.4.3.24 DST_ADDR_ACTIVE 寄存器

偏移地址：3Ch　复位值：00000000h

目的地址活动寄存器

表 9-61 DST_ADDR_ACTIVE 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADDR | | | | | | | |
| R-0h | | | | | | | |

表 9-62 DST_ADDR_ACTIVE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | ADDR | R | 0h | 目的地址活动寄存器。<br>传输进行期间，本寄存器保存目的地址的当前值。该地址可在一次写操作、一次突发传输或回绕后改变。<br>复位类型：SYSRSn。 |

### 9.4.4 DMA 寄存器与 DriverLib 函数对应表

表 9-63 DMA 寄存器与 DriverLib 函数对应表

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| CTRL | dma.h | DMA_initController |
| DEBUGCTRL | dma.h | DMA_setEmulationMode |
| PRIORITYCTRL1 | dma.h | DMA_setPriorityMode |
| PRIORITYSTAT | - | - |
| MODE | dma.c | DMA_configMode |
| MODE | dma.h | DMA_enableTrigger |
| MODE | dma.h | DMA_disableTrigger |
| MODE | dma.h | DMA_enableInterrupt |
| MODE | dma.h | DMA_disableInterrupt |
| MODE | dma.h | DMA_enableOverrunInterrupt |
| MODE | dma.h | DMA_disableOverrunInterrupt |
| MODE | dma.h | DMA_setInterruptMode |
| CONTROL | dma.h | DMA_triggerSoftReset |
| CONTROL | dma.h | DMA_forceTrigger |

| 寄存器 | 文件 | DriverLib 函数 |
| --- | --- | --- |
| CONTROL | dma.h | DMA_clearTriggerFlag |
| CONTROL | dma.h | DMA_getTransferStatusFlag |
| CONTROL | dma.h | DMA_getBurstStatusFlag |
| CONTROL | dma.h | DMA_getRunStatusFlag |
| CONTROL | dma.h | DMA_getOverflowFlag |
| CONTROL | dma.h | DMA_getTriggerFlagStatus |
| CONTROL | dma.h | DMA_startChannel |
| CONTROL | dma.h | DMA_stopChannel |
| CONTROL | dma.h | DMA_clearErrorFlag |
| BURST_SIZE | dma.c | DMA_configBurst |
| BURST_COUNT | - | - |
| SRC_BURST_STEP | dma.c | DMA_configBurst |
| DST_BURST_STEP | dma.c | DMA_configBurst |
| TRANSFER_SIZE | dma.c | DMA_configTransfer |
| TRANSFER_COUNT | - | - |
| SRC_TRANSFER_STEP | dma.c | DMA_configTransfer |
| DST_TRANSFER_STEP | dma.c | DMA_configTransfer |
| SRC_WRAP_SIZE | dma.c | DMA_configWrap |
| SRC_WRAP_COUNT | - | - |
| SRC_WRAP_STEP | dma.c | DMA_configWrap |
| DST_WRAP_SIZE | dma.c | DMA_configWrap |
| DST_WRAP_COUNT | - | - |
| DST_WRAP_STEP | dma.c | DMA_configWrap |
| SRC_BEG_ADDR_SHADOW | dma.c | DMA_configAddresses |
| SRC_BEG_ADDR_SHADOW | dma.h | DMA_configSourceAddress |
| SRC_ADDR_SHADOW | dma.c | DMA_configAddresses |
| SRC_ADDR_SHADOW | dma.h | DMA_configSourceAddress |
| SRC_BEG_ADDR_ACTIVE | - | - |
| SRC_ADDR_ACTIVE | - | - |
| DST_BEG_ADDR_SHADOW | dma.c | DMA_configAddresses |
| DST_BEG_ADDR_SHADOW | dma.h | DMA_configDestAddress |
| DST_ADDR_SHADOW | dma.c | DMA_configAddresses |
| DST_ADDR_SHADOW | dma.h | DMA_configDestAddress |
| DST_BEG_ADDR_ACTIVE | - | - |
| DST_ADDR_ACTIVE | - | - |

# 10 可配置逻辑块（CLB）

## 10.1 简介

可配置逻辑块（CLB）是一组可配置模块的集合，能够通过软件实现互联，以执行自定义数字逻辑功能。借助一组交叉开关互联电路，可配置逻辑块能够增强现有外设的功能；这类交叉开关可为现有控制外设（如 ePWM、eCAP 和 eQEP）提供高度的互联能力。同时，交叉开关还支持可配置逻辑块与 GPIO 相连接。通过这种方式，可配置逻辑块可被配置为与器件外设进行交互，既可以实现简易脉宽调制发生器这类小型逻辑功能，也能够用于执行自定义串行数据交换协议。

## 10.2 主要特性

本设备中具有两组 CLB，每组 CLB 包含以下特性：

- 4 个相同单元块，每个单元块集成组合逻辑与时序逻辑模块，支持软件配置实现自定义数字逻辑功能
- 多类型输入输出管理，8 个输入可从全局信号、本地信号中选择，8 个输出可复制为 24 个输出信号
- 最大频率 150MHz，从 2 型 CLB 起含预分频模块
- 每个 TILE 中包含 3 组计数器、LUT4、FSM，8 个 OUTLUT，1 个高级控制器，1 个异步输出调节模块，1 个静态开关模块
- CPU 与 DMA 接口特性

## 10.3 功能描述

CLB 子系统包含多个相同的单元块。CLB 子系统中有四个这样的单元块；其他器件可包含更多或更少的单元块。单元块编号为 1 到 N，其中 N 是器件上的总单元块数。每个单元块包含组合逻辑块和时序逻辑块，以及本文档后续将介绍的其他专用硬件。图中展示了该器件中 CLB 子系统的结构。

单元块包含核心逻辑，提供逻辑重配置能力。每个 CLB 单元块都与一个独立的 CPU 接口相关联，该接口包含控制和配置单元块内逻辑所需的寄存器。CPU 接口还包含数据传输缓冲区，这些缓冲区可用作可配置逻辑的一部分，与器件的其他部分交换数据。图中展示了单元块、CPU 接口和器件之间的连接。

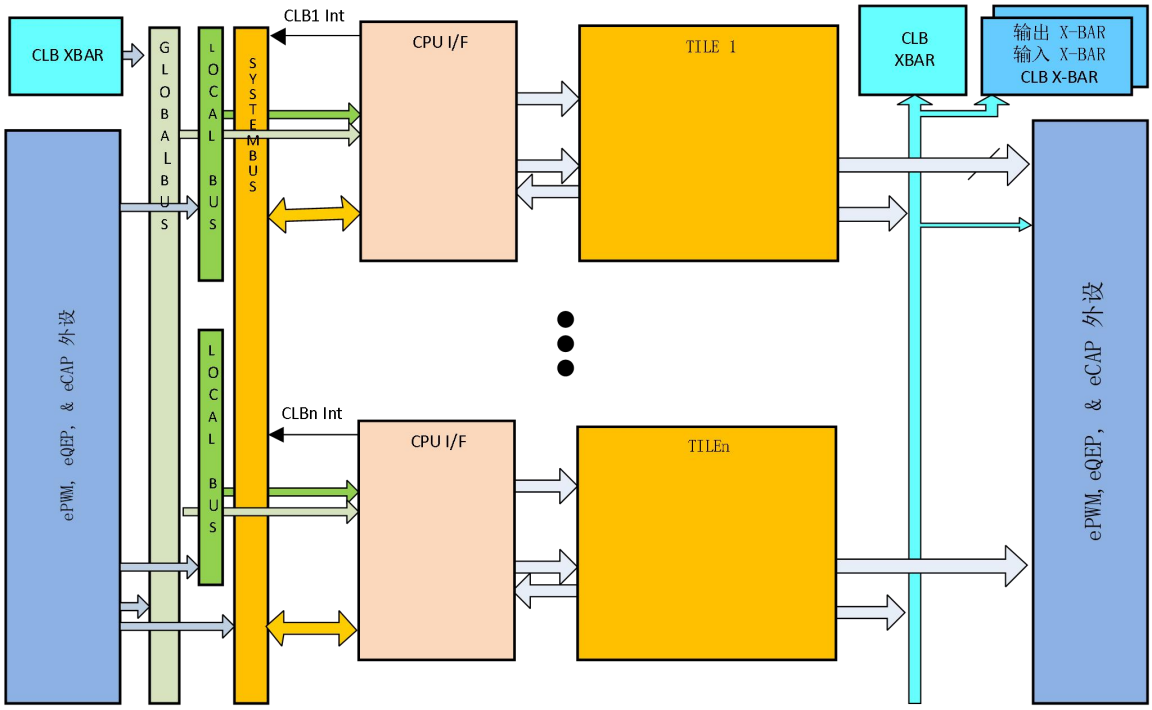



图 10-1 设备中 CLB 子系统框图

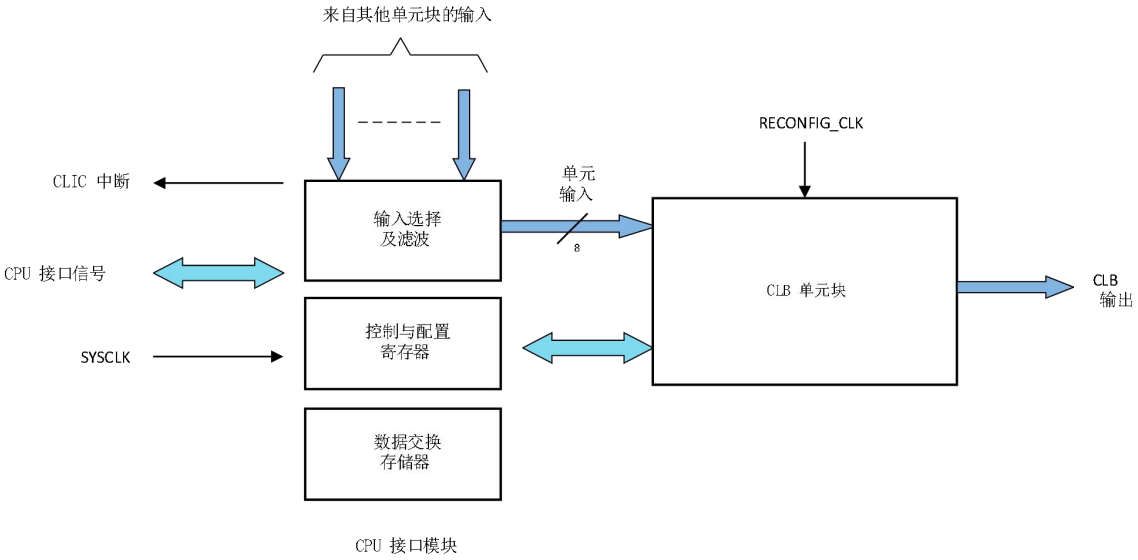



图 10-2 CLB TILE 与 CPU 接口框图

### 10.3.1 CLB 时钟

在该器件中，CLB 时钟被称为 CLBx 时钟，可通过 SysCtl_enablePeripheral 函数借助 SYSCTL_PERIPH_CLK_CLBx 来启用或禁用。其最大频率为 100MHz，并且可以通过修改 CLBx 时钟来启用和配置该时钟。

从 CLB 类型 2 开始，提供了一个时钟预分频器模块。该预分频器模块能够生成 CLB 时钟信号的预分频版本，可用作 CLB TILE 计数器的输入。

注意：

预分频器逻辑不会改变 CLB 的实际时钟速度。预分频器会生成一个选通信号（该信号可以以设定的预分频速率进行切换），此信号作为另一个输入信号提供给 CLB 逻辑，且仅在需要时才会使用该选通信号。

预分频模块如图所示。

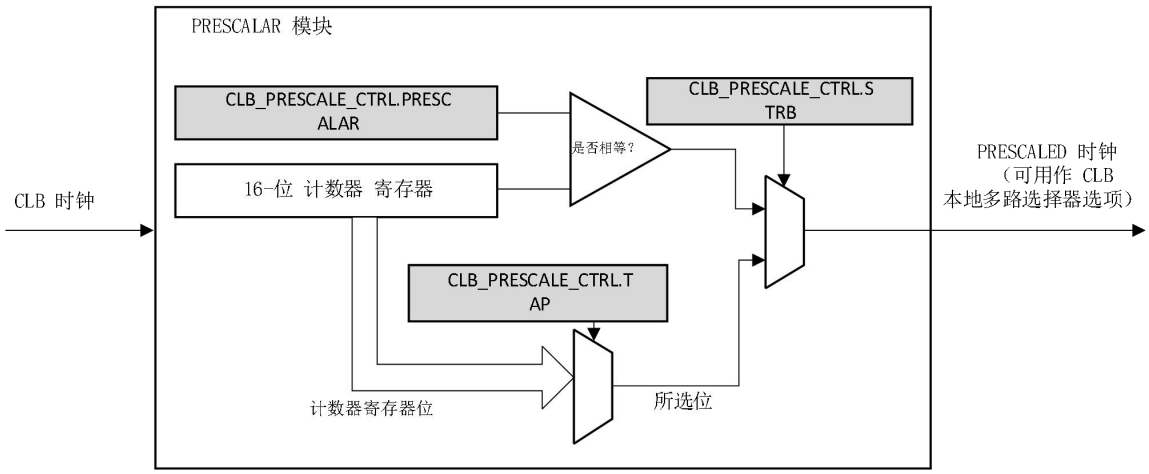



图 10-3 CLB 时钟预分频

### 10.3.2 CLB 输入/输出连接

#### 10.3.2.1 概述

该器件中有 4 个 CLB 模块实例。每个 CLB 实例都有一组公共输入信号，称为全局输入信号。此外，每个 CLB 实例还有一组特定的输入信号，这些信号对每个

实例是唯一的，称为本地输入信号。CLB 的 8 个输入中的每个输入都可从任意全局输入信号或本地输入信号中选择。

注意：通过交叉开关（XBAR）路由到 CLB 的信号必须在 CLB 内部同步。

### 10.3.2.2 CLB 输入选择

每个 CLB 模块有 8 个输入应用于可重配置逻辑单元。这些输入中的每个都可由一组预定义信号选择性驱动。两级多路选择器（mux）结构允许每个 CLB 实例的每个输入选择一个信号。

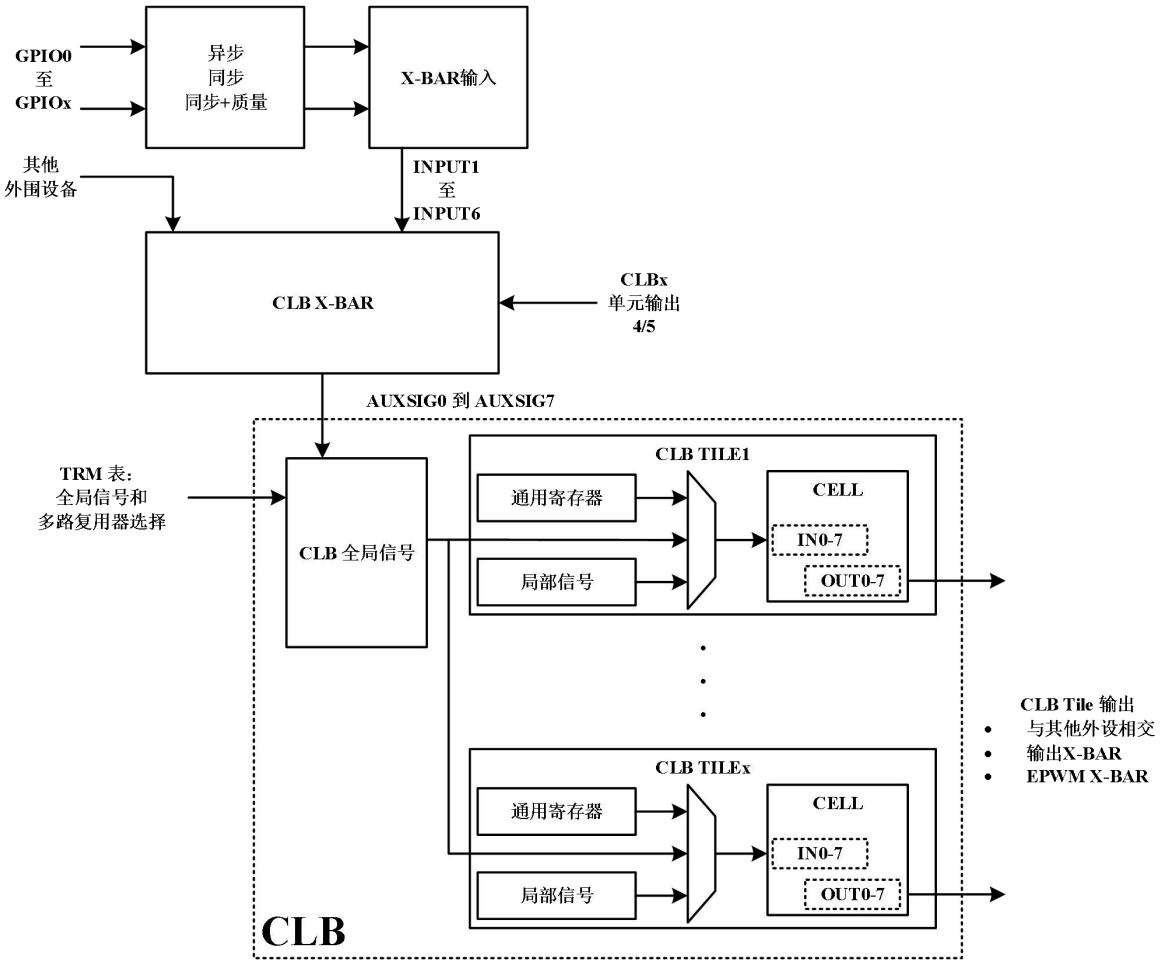



图 10-4 GPIO 到 CLB 单元块连接

一组信号为所有 CLB 实例共用，在图中称为全局输入。另一组信号为每个 CLB 实例独有，在图中称为本地输入。

寄存器 CLB_LCL_MUX_SEL_1 和 CLB_LCL_MUX_SEL_2 控制 8 个输入中每个输入的本地多路选择器选择。多路选择器控制寄存器 CLB_GLBL_MUX_SEL_1 和 CLB_GLBL_MUX_SEL_2 控制 8 个输入中每个输入的全局多路选择器选择。

当本地多路选择器选择值为 0 时，所选全局多路选择器输入信号将连接到对应的 CLB 输入。例如，设置 CLB_LCL_MUX_SEL_IN_0 = 0 且 CLB_GLBL_MUX_SEL_IN_0 = 8 时，全局多路选择器输入 8 将连接到 CLB 输入 0。输入滤波功能可用于启用 CLB 输入的边沿检测，还可使输入与 CLB 时钟同步。

全局多路选择器设置如表所示，本地输入多路选择器设置如表所示。

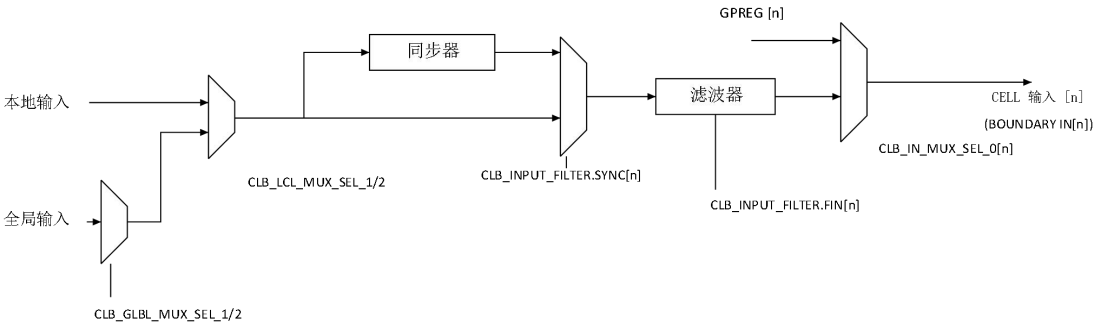



图 10-5 CLB 输入多路选择与滤波

图中展示了如何对异步信号（本例中为 ePWM 信号）进行同步。

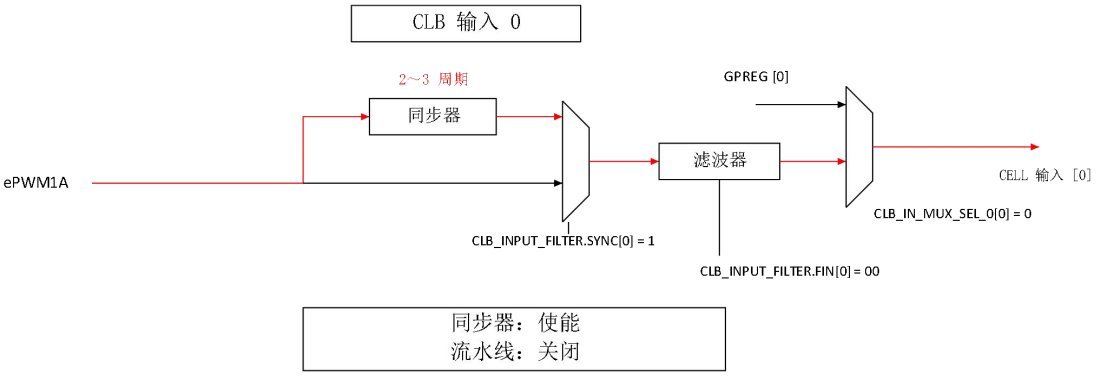



图 10-6 CLB 输入同步实例

注意：如果下表中的信号标明需要同步，则必须通过 CLB_INPUT_FILTER 寄存器中相应的 SYNC 位启用 CLB 输入同步器。该同步会给输入添加 2-3 个 CLB 时钟周期的延迟，延迟可能为 2 个或 3 个周期，无法预测。如果标明需要同步的信号在进入 CLB 单元块前未先进行同步，可能会存在亚稳态风险。这种亚稳态可能会

因电压、温度和晶圆制造工艺的不同而导致错误。请注意，此要求是对 GPIO 输入同步的补充，且独立于 GPIO 输入同步。

表 10-1 全局信号和多路选择器选择

| 选中的值 | CLB1 输入 | CLB2 输入 | CLB3 输入 | CLB4 输入 | 同步要求 |
|---|---|---|---|---|---|
| 0 | EPWM1A | EPWM1A | EPWM1A | EPWM1A | 使能 |
| 1 | EPWM1A_OE | EPWM1A_OE | EPWM1A_OE | EPWM1A_OE | 使能 |
| 2 | EPWM1B | EPWM1B | EPWM1B | EPWM1B | 使能 |
| 3 | EPWM1B_OE | EPWM1B_OE | EPWM1B_OE | EPWM1B_OE | 使能 |
| 4 | EPWM1_CTR_ZERO | EPWM1_CTR_ZERO | EPWM1_CTR_ZERO | EPWM1_CTR_ZERO | 关闭 |
| 5 | EPWM1_CTR_PRD | EPWM1_CTR_PRD | EPWM1_CTR_PRD | EPWM1_CTR_PRD | 关闭 |
| 6 | EPWM1_CTR_DIR | EPWM1_CTR_DIR | EPWM1_CTR_DIR | EPWM1_CTR_DIR | 关闭 |
| 7 | EPWM1_TBCLK | EPWM1_TBCLK | EPWM1_TBCLK | EPWM1_TBCLK | 关闭 |
| 8 | EPWM1_CTR_CMPA | EPWM1_CTR_CMPA | EPWM1_CTR_CMPA | EPWM1_CTR_CMPA | 关闭 |
| 9 | EPWM1_CTR_CMPB | EPWM1_CTR_CMPB | EPWM1_CTR_CMPB | EPWM1_CTR_CMPB | 关闭 |
| 10 | EPWM1_CTR_CMPC | EPWM1_CTR_CMPC | EPWM1_CTR_CMPC | EPWM1_CTR_CMPC | 关闭 |
| 11 | EPWM1_CTR_CMPD | EPWM1_CTR_CMPD | EPWM1_CTR_CMPD | EPWM1_CTR_CMPD | 关闭 |
| 12 | EPWM1A_AQ | EPWM1A_AQ | EPWM1A_AQ | EPWM1A_AQ | 关闭 |
| 13 | EPWM1B_AQ | EPWM1B_AQ | EPWM1B_AQ | EPWM1B_AQ | 关闭 |
| 14 | EPWM1A_DB | EPWM1A_DB | EPWM1A_DB | EPWM1A_DB | 关闭 |
| 15 | EPWM1B_DB | EPWM1B_DB | EPWM1B_DB | EPWM1B_DB | 关闭 |
| 16 | EPWM2A | EPWM2A | EPWM2A | EPWM2A | 使能 |
| 17 | EPWM2A_OE | EPWM2A_OE | EPWM2A_OE | EPWM2A_OE | 使能 |
| 18 | EPWM2B | EPWM2B | EPWM2B | EPWM2B | 使能 |
| 19 | EPWM2B_OE | EPWM2B_OE | EPWM2B_OE | EPWM2B_OE | 使能 |
| 20 | EPWM2_CTR_ZERO | EPWM2_CTR_ZERO | EPWM2_CTR_ZERO | EPWM2_CTR_ZERO | 关闭 |
| 21 | EPWM2_CTR_PRD | EPWM2_CTR_PRD | EPWM2_CTR_PRD | EPWM2_CTR_PRD | 关闭 |
| 22 | EPWM2_CTR_DIR | EPWM2_CTR_DIR | EPWM2_CTR_DIR | EPWM2_CTR_DIR | 关闭 |
| 23 | EPWM2_TBCLK | EPWM2_TBCLK | EPWM2_TBCLK | EPWM2_TBCLK | 关闭 |
| 24 | EPWM2_CTR_CMPA | EPWM2_CTR_CMPA | EPWM2_CTR_CMPA | EPWM2_CTR_CMPA | 关闭 |
| 25 | EPWM2_CTR_CMPB | EPWM2_CTR_CMPB | EPWM2_CTR_CMPB | EPWM2_CTR_CMPB | 关闭 |
| 26 | EPWM2_CTR_CMPC | EPWM2_CTR_CMPC | EPWM2_CTR_CMPC | EPWM2_CTR_CMPC | 关闭 |
| 27 | EPWM2_CTR_CMPD | EPWM2_CTR_CMPD | EPWM2_CTR_CMPD | EPWM2_CTR_CMPD | 关闭 |
| 28 | EPWM2A_AQ | EPWM2A_AQ | EPWM2A_AQ | EPWM2A_AQ | 关闭 |
| 29 | EPWM2B_AQ | EPWM2B_AQ | EPWM2B_AQ | EPWM2B_AQ | 关闭 |
| 30 | EPWM2A_DB | EPWM2A_DB | EPWM2A_DB | EPWM2A_DB | 关闭 |

| 选中的值 | CLB1 输入 | CLB2 输入 | CLB3 输入 | CLB4 输入 | 同步要求 |
|---|---|---|---|---|---|
| 31 | EPWM2B_DB | EPWM2B_DB | EPWM2B_DB | EPWM2B_DB | 关闭 |
| 32 | EPWM3A | EPWM3A | EPWM3A | EPWM3A | 使能 |
| 33 | EPWM3A_OE | EPWM3A_OE | EPWM3A_OE | EPWM3A_OE | 使能 |
| 34 | EPWM3B | EPWM3B | EPWM3B | EPWM3B | 使能 |
| 35 | EPWM3B_OE | EPWM3B_OE | EPWM3B_OE | EPWM3B_OE | 使能 |
| 36 | EPWM3_CTR_ZERO | EPWM3_CTR_ZERO | EPWM3_CTR_ZERO | EPWM3_CTR_ZERO | 关闭 |
| 37 | EPWM3_CTR_PRD | EPWM3_CTR_PRD | EPWM3_CTR_PRD | EPWM3_CTR_PRD | 关闭 |
| 38 | EPWM3_CTR_DIR | EPWM3_CTR_DIR | EPWM3_CTR_DIR | EPWM3_CTR_DIR | 关闭 |
| 39 | EPWM3_TBCLK | EPWM3_TBCLK | EPWM3_TBCLK | EPWM3_TBCLK | 关闭 |
| 40 | EPWM3_CTR_CMPA | EPWM3_CTR_CMPA | EPWM3_CTR_CMPA | EPWM3_CTR_CMPA | 关闭 |
| 41 | EPWM3_CTR_CMPB | EPWM3_CTR_CMPB | EPWM3_CTR_CMPB | EPWM3_CTR_CMPB | 关闭 |
| 42 | EPWM3_CTR_CMPC | EPWM3_CTR_CMPC | EPWM3_CTR_CMPC | EPWM3_CTR_CMPC | 关闭 |
| 43 | EPWM3_CTR_CMPD | EPWM3_CTR_CMPD | EPWM3_CTR_CMPD | EPWM3_CTR_CMPD | 关闭 |
| 44 | EPWM3A_AQ | EPWM3A_AQ | EPWM3A_AQ | EPWM3A_AQ | 关闭 |
| 45 | EPWM3B_AQ | EPWM3B_AQ | EPWM3B_AQ | EPWM3B_AQ | 关闭 |
| 46 | EPWM3A_DB | EPWM3A_DB | EPWM3A_DB | EPWM3A_DB | 关闭 |
| 47 | EPWM3B_DB | EPWM3B_DB | EPWM3B_DB | EPWM3B_DB | 关闭 |
| 48 | EPWM4A | EPWM4A | EPWM4A | EPWM4A | 使能 |
| 49 | EPWM4A_OE | EPWM4A_OE | EPWM4A_OE | EPWM4A_OE | 使能 |
| 50 | EPWM4B | EPWM4B | EPWM4B | EPWM4B | 使能 |
| 51 | EPWM4B_OE | EPWM4B_OE | EPWM4B_OE | EPWM4B_OE | 使能 |
| 52 | EPWM4_CTR_ZERO | EPWM4_CTR_ZERO | EPWM4_CTR_ZERO | EPWM4_CTR_ZERO | 关闭 |
| 53 | EPWM4_CTR_PRD | EPWM4_CTR_PRD | EPWM4_CTR_PRD | EPWM4_CTR_PRD | 关闭 |
| 54 | EPWM4_CTR_DIR | EPWM4_CTR_DIR | EPWM4_CTR_DIR | EPWM4_CTR_DIR | 关闭 |
| 55 | EPWM4_TBCLK | EPWM4_TBCLK | EPWM4_TBCLK | EPWM4_TBCLK | 关闭 |
| 56 | EPWM4_CTR_CMPA | EPWM4_CTR_CMPA | EPWM4_CTR_CMPA | EPWM4_CTR_CMPA | 关闭 |
| 57 | EPWM4_CTR_CMPB | EPWM4_CTR_CMPB | EPWM4_CTR_CMPB | EPWM4_CTR_CMPB | 关闭 |
| 58 | EPWM4_CTR_CMPC | EPWM4_CTR_CMPC | EPWM4_CTR_CMPC | EPWM4_CTR_CMPC | 关闭 |
| 59 | EPWM4_CTR_CMPD | EPWM4_CTR_CMPD | EPWM4_CTR_CMPD | EPWM4_CTR_CMPD | 关闭 |
| 60 | EPWM4A_AQ | EPWM4A_AQ | EPWM4A_AQ | EPWM4A_AQ | 关闭 |
| 61 | EPWM4B_AQ | EPWM4B_AQ | EPWM4B_AQ | EPWM4B_AQ | 关闭 |
| 62 | EPWM4A_DB | EPWM4A_DB | EPWM4A_DB | EPWM4A_DB | 关闭 |
| 63 | EPWM4B_DB | EPWM4B_DB | EPWM4B_DB | EPWM4B_DB | 关闭 |
| 64 | AUXSIG0 | AUXSIG0 | AUXSIG0 | AUXSIG0 | 使能 |
| 65 | AUXSIG1 | AUXSIG1 | AUXSIG1 | AUXSIG1 | 使能 |
| 66 | AUXSIG2 | AUXSIG2 | AUXSIG2 | AUXSIG2 | 使能 |

| 选中的值 | CLB1 输入 | CLB2 输入 | CLB3 输入 | CLB4 输入 | 同步要求 |
|---|---|---|---|---|---|
| 67 | AUXSIG3 | AUXSIG3 | AUXSIG3 | AUXSIG3 | 使能 |
| 68 | AUXSIG4 | AUXSIG4 | AUXSIG4 | AUXSIG4 | 使能 |
| 69 | AUXSIG5 | AUXSIG5 | AUXSIG5 | AUXSIG5 | 使能 |
| 70 | AUXSIG6 | AUXSIG6 | AUXSIG6 | AUXSIG6 | 使能 |
| 71 | CLB1_OUT0 | CLB1_OUT0 | CLB1_OUT0 | CLB1_OUT0 | 关闭 |
| 72 | CLB1_OUT1 | CLB1_OUT1 | CLB1_OUT1 | CLB1_OUT1 | 关闭 |
| 73 | CLB1_OUT2 | CLB1_OUT2 | CLB1_OUT2 | CLB1_OUT2 | 关闭 |
| 74 | CLB1_OUT3 | CLB1_OUT3 | CLB1_OUT3 | CLB1_OUT3 | 关闭 |
| 75 | CLB1_OUT4 | CLB1_OUT4 | CLB1_OUT4 | CLB1_OUT4 | 关闭 |
| 76 | CLB1_OUT5 | CLB1_OUT5 | CLB1_OUT5 | CLB1_OUT5 | 关闭 |
| 77 | CLB1_OUT6 | CLB1_OUT6 | CLB1_OUT6 | CLB1_OUT6 | 关闭 |
| 78 | CLB1_OUT7 | CLB1_OUT7 | CLB1_OUT7 | CLB1_OUT7 | 关闭 |
| 79 | CLB2_OUT0 | CLB2_OUT0 | CLB2_OUT0 | CLB2_OUT0 | 关闭 |
| 80 | CLB2_OUT1 | CLB2_OUT1 | CLB2_OUT1 | CLB2_OUT1 | 关闭 |
| 81 | CLB2_OUT2 | CLB2_OUT2 | CLB2_OUT2 | CLB2_OUT2 | 关闭 |
| 82 | CLB2_OUT3 | CLB2_OUT3 | CLB2_OUT3 | CLB2_OUT3 | 关闭 |
| 83 | CLB2_OUT4 | CLB2_OUT4 | CLB2_OUT4 | CLB2_OUT4 | 关闭 |
| 84 | CLB2_OUT5 | CLB2_OUT5 | CLB2_OUT5 | CLB2_OUT5 | 关闭 |
| 85 | CLB2_OUT6 | CLB2_OUT6 | CLB2_OUT6 | CLB2_OUT6 | 关闭 |
| 86 | CLB2_OUT7 | CLB2_OUT7 | CLB2_OUT7 | CLB2_OUT7 | 关闭 |
| 87 | CLB3_OUT0 | CLB3_OUT0 | CLB3_OUT0 | CLB3_OUT0 | 关闭 |
| 88 | CLB3_OUT1 | CLB3_OUT1 | CLB3_OUT1 | CLB3_OUT1 | 关闭 |
| 89 | CLB3_OUT2 | CLB3_OUT2 | CLB3_OUT2 | CLB3_OUT2 | 关闭 |
| 90 | CLB3_OUT3 | CLB3_OUT3 | CLB3_OUT3 | CLB3_OUT3 | 关闭 |
| 91 | CLB3_OUT4 | CLB3_OUT4 | CLB3_OUT4 | CLB3_OUT4 | 关闭 |
| 92 | CLB3_OUT5 | CLB3_OUT5 | CLB3_OUT5 | CLB3_OUT5 | 关闭 |
| 93 | CLB3_OUT6 | CLB3_OUT6 | CLB3_OUT6 | CLB3_OUT6 | 关闭 |
| 94 | CLB3_OUT7 | CLB3_OUT7 | CLB3_OUT7 | CLB3_OUT7 | 关闭 |
| 95 | CLB4_OUT0 | CLB4_OUT0 | CLB4_OUT0 | CLB4_OUT0 | 关闭 |
| 96 | CLB4_OUT1 | CLB4_OUT1 | CLB4_OUT1 | CLB4_OUT1 | 关闭 |
| 97 | CLB4_OUT2 | CLB4_OUT2 | CLB4_OUT2 | CLB4_OUT2 | 关闭 |
| 98 | CLB4_OUT3 | CLB4_OUT3 | CLB4_OUT3 | CLB4_OUT3 | 关闭 |
| 99 | CLB4_OUT4 | CLB4_OUT4 | CLB4_OUT4 | CLB4_OUT4 | 关闭 |
| 100 | CLB4_OUT5 | CLB4_OUT5 | CLB4_OUT5 | CLB4_OUT5 | 关闭 |
| 101 | CLB4_OUT6 | CLB4_OUT6 | CLB4_OUT6 | CLB4_OUT6 | 关闭 |
| 102 | CLB4_OUT7 | CLB4_OUT7 | CLB4_OUT7 | CLB4_OUT7 | 关闭 |
| 103-127 | 保留 | 保留 | 保留 | 保留 | 保留 |

注意：

EPWMxA_OE 和 EPWMxB_OE 指相应 EPWM 模块的跳闸输出。

EPWMxA_AQ 和 EPWMxB_AQ 指相应 EPWM 模块中 AQ 子模块的输出。

EPWMxA_DB 和 EPWMBx_DB 指相应 EPWM 模块中 DB 子模块的输出。

注意：如果下表中的信号标明需要同步，则必须通过 CLB_INPUT_FILTER 寄存器中相应的 SYNC 位启用 CLB 输入同步器。该同步会给输入添加 2-3 个 CLB 时钟周期的延迟，延迟可能为 2 个或 3 个周期，无法预测。如果标明需要同步的信号在进入 CLB 单元块前未先进行同步，可能会存在亚稳态风险。这种亚稳态可能会因电压、温度和晶圆制造工艺的不同而导致错误。请注意，此要求是对 GPIO 输入同步的补充，且独立于 GPIO 输入同步。

表 10-2 本地信号和多路选择器选择

| 选中的值 | 输入 CLB1 | 输入 CLB2 | 输入 CLB3 | 输入 CLB4 | 同步要求 |
|---|---|---|---|---|---|
| 0 | CLB1_GLB_MUX_OUT | CLB2_GLB_MUX_OUT | CLB3_GLB_MUX_OUT | CLB4_GLB_MUX_OUT | 使能 |
| 1 | EPWM1_DCAEVT1 | EPWM2_DCAEVT1 | EPWM3_DCAEVT1 | EPWM4_DCAEVT1 | 使能 |
| 2 | EPWM1_DCAEVT2 | EPWM2_DCAEVT2 | EPWM3_DCAEVT2 | EPWM4_DCAEVT2 | 使能 |
| 3 | EPWM1_DCBEVT1 | EPWM2_DCBEVT1 | EPWM3_DCBEVT1 | EPWM4_DCBEVT1 | 使能 |
| 4 | EPWM1_DCBEVT2 | EPWM2_DCBEVT2 | EPWM3_DCBEVT2 | EPWM4_DCBEVT2 | 使能 |
| 5 | EPWM1_DCAH | EPWM2_DCAH | EPWM3_DCAH | EPWM4_DCAH | 使能 |
| 6 | EPWM1_DCAL | EPWM2_DCAL | EPWM3_DCAL | EPWM4_DCAL | 使能 |
| 7 | EPWM1_DCBH | EPWM2_DCBH | EPWM3_DCBH | EPWM4_DCBH | 使能 |
| 8 | EPWM1_DCBL | EPWM2_DCBL | EPWM3_DCBL | EPWM4_DCBL | 使能 |
| 9 | EPWM1_OST | EPWM2_OST | EPWM3_OST | EPWM4_OST | 使能 |
| 10 | EPWM1_CBC | EPWM2_CBC | EPWM3_CBC | EPWM4_CBC | 使能 |
| 11 | ECAP1IN0 | ECAP2IN0 | ECAP3IN0 | ECAP4IN0 | 使能 |
| 12 | ECAP1_OUT | ECAP2_OUT | ECAP3_OUT | ECAP4_OUT | 关闭 |
| 13 | ECAP1_OUT_EN | ECAP2_OUT_EN | ECAP3_OUT_EN | ECAP4_OUT_EN | 关闭 |
| 14 | ECAP1_CEVT1 | ECAP2_CEVT1 | ECAP3_CEVT1 | ECAP4_CEVT1 | 关闭 |
| 15 | ECAP1_CEVT2 | ECAP2_CEVT2 | ECAP3_CEVT2 | ECAP4_CEVT2 | 关闭 |
| 16 | ECAP1_CEVT3 | ECAP2_CEVT3 | ECAP3_CEVT3 | ECAP4_CEVT3 | 关闭 |
| 17 | ECAP1_CEVT4 | ECAP2_CEVT4 | ECAP3_CEVT4 | ECAP4_CEVT4 | 关闭 |
| 18 | EQEP1A | EQEP2A | 保留 | 保留 | 使能 |
| 19 | EQEP1B | EQEP2B | 保留 | 保留 | 使能 |
| 20 | EQEP1I | EQEP2I | 保留 | 保留 | 使能 |
| 21 | EQEP1S | EQEP2S | 保留 | 保留 | 关闭 |
| 22 | CPU1_TBCLKSYNC | CPU1_TBCLKSYNC | 保留 | 保留 | 关闭 |
| 23 | 保留 | 保留 | 保留 | 保留 | 保留 |
| 24 | CPU1_HALT | CPU1_HALT | 保留 | 保留 | 使能 |
| 25 | 保留 | 保留 | 保留 | 保留 | 保留 |

| 选中的值 | 输入 CLB1 | 输入 CLB2 | 输入 CLB3 | 输入 CLB4 | 同步要求 |
|---|---|---|---|---|---|
| 26 | SPIA_CLK | SPIB_CLK | SPIA_CLK | SPIB_CLK | 使能 |
| 27 | SPIA_SIMO_IN | SPIB_SIMO_IN | SPIA_SIMO_IN | SPIB_SIMO_IN | 使能 |
| 28 | SPIA_STE | SPIB_STE | SPIA_STE | SPIB_STE | 使能 |
| 29 | SCIA_TX | SCIB_TX | SCIA_TX | SCIB_TX | 使能 |
| 30 | 保留 | 保留 | 保留 | 保留 | 关闭 |
| 31 | CLB1_PSCLK | CLB2_PSCLK | CLB3_PSCLK | CLB4_PSCLK | 关闭 |

GPREG 可由 CPU 访问，该寄存器的位可用作 CLB 单元块的边界输入。例如，CLB1 的 GPREG [0] 可用作对应 CLB 单元块的边界输入 0（单元输入 0）。

要实现多个单元块之间的连接，可使用 CLBx 的 OUT4/5，并通过 CLB 交叉开关和全局信号多路选择器连接到 CLBy 的边界输入 z。

另一种方法是将 CLBx 的 OUT0-7 连接到 GPIO，然后通过输入交叉开关将信号重新引入器件，并通过 CLB 交叉开关和全局信号多路选择器连接到 CLBy 的边界输入 z。

要将 GPIO 用作 CLB 的输入，必须利用输入交叉开关和 CLB 交叉开关。图中展示了如何将 GPIO 用作 CLB 单元块的输入。

#### 10.3.2.3 CLB 输出选择

CLB 的 8 个输出会被复制，生成 24 个输出信号。这 24 个输出中的每个输出在 CLB 输出使能寄存器 CLB_OUT_EN 中都有一个独立的使能位。CLB 输出连接到器件中的 ePWM、eCAP、eQEP 和交叉开关模块，用户可通过 CLB 增强这些模块的功能。图中展示了 CLB 的输出。

8 个输出经复制后共生成 24 个输出（如图所示）。其中部分新输出可通过 CLB 全局多路选择器输入实现单元块间连接。

注意：OUTLUT0 的输出连接到 OUT0、OUT8 和 OUT16，信号相同，但每个 OUTy 可访问的外设不同。同理，OUTLUT1 连接到 OUT1、OUT9 和 OUT17，信号相同。

CLBx_OUT12 至 CLBx_OUT15 为非寄存输出，与 CLB 时钟异步。

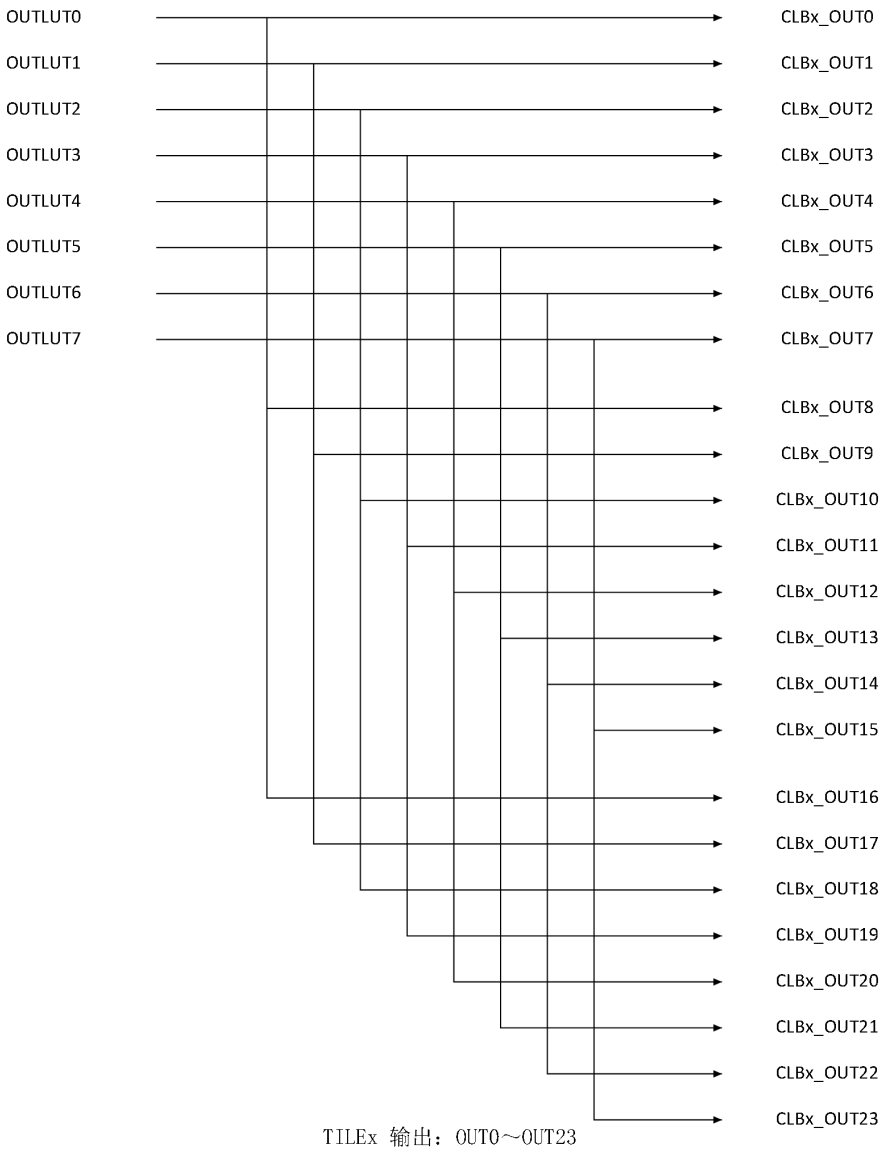



图 10-7 CLB 输出

### 10.3.2.4 CLB 输出信号多路选择器

每个 CLB 输出信号都会经过一个外部多路选择器，该多路选择器与特定的外设信号交叉，如图所示。多路选择器的输出连接到原始外设信号的目的地，默认设置为外设信号直通。多路选择器由 CLB 输出使能寄存器 CLB_OUT_EN 中的位 [n] 控制。

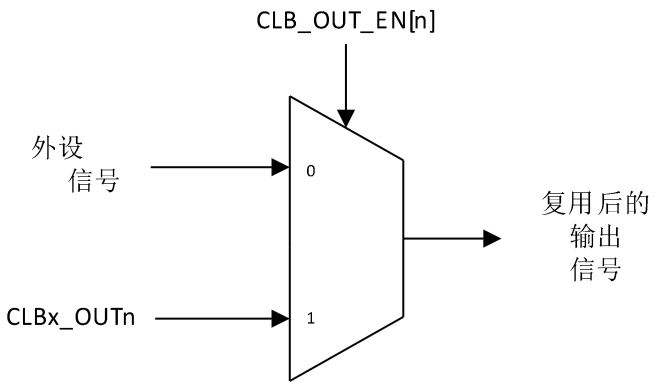



图 10-8 CLB 输出信号多路选择器

例如，若需用 CLB1 的 OUT0 覆盖 EPWM1A 信号，需将 OUT0 的输出使能位置 1。

下表展示了外设信号与 CLB 输出的分配关系。

表 10-3 CLB 输出信号多路选择器表

| CLB 输出 | 输出查找表 (OUTLUT) | CLB1 目标端 | CLB2 目标端 | CLB3 目标端 | CLB4 目标端 |
|---|---|---|---|---|---|
| 0 | OUTLUT0 | EPWM1A | EPWM2A | EPWM3A | EPWM4A |
| 1 | OUTLUT1 | EPWM1A_OE | EPWM2A_OE | EPWM3A_OE | EPWM4A_OE |
| 2 | OUTLUT2 | EPWM1B | EPWM2B | EPWM3B | EPWM4B |
| 3 | OUTLUT3 | EPWM1B_OE | EPWM2B_OE | EPWM3B_OE | EPWM4B_OE |
| 4 | OUTLUT4 | EPWM1A_AQ | EPWM2A_AQ | EPWM3A_AQ | EPWM4A_AQ |
| 5 | OUTLUT5 | EPWM1B_AQ | EPWM2B_AQ | EPWM3B_AQ | EPWM4B_AQ |
| 6 | OUTLUT6 | EPWM1A_DB | EPWM2A_DB | EPWM3A_DB | EPWM4A_DB |
| 7 | OUTLUT7 | EPWM1B_DB | EPWM2B_DB | EPWM3B_DB | EPWM4B_DB |
| 8 | OUTLUT0 | EQEP1_QCLK | EQEP2_QCLK | SPIA_CLK | SPIB_CLK |
| 9 | OUTLUT1 | EQEP1_QDIR | EQEP2_QDIR | SPIA_SPIDAT | SPIB_SPIDAT |
| 10 | OUTLUT2 | 保留 | 保留 | SPIA_STE | SPIB_STE |
| 11 | OUTLUT3 | 保留 | 保留 | SCIA_RX | SCIB_RX |
| 12 | OUTLUT4 | XBARs | XBARs | XBARs | XBARs |
| 13 | OUTLUT5 | XBARs | XBARs | XBARs | XBARs |
| 14 | OUTLUT6 | ECAP1_OUT_EN | ECAP2_OUT_EN | ECAP3_OUT_EN | ECAP4_OUT_EN |
| 15 | OUTLUT7 | ECAP1_OUT | ECAP2_OUT | ECAP3_OUT | ECAP4_OUT |
| 16 | OUTLUT0 | Global Mux | Global Mux | Global Mux | Global Mux |
| 17 | OUTLUT1 | Global Mux | Global Mux | Global Mux | Global Mux |
| 18 | OUTLUT2 | Global Mux | Global Mux | Global Mux | Global Mux |
| 19 | OUTLUT3 | Global Mux | Global Mux | Global Mux | Global Mux |
| 20 | OUTLUT4 | Global Mux | Global Mux | Global Mux | Global Mux |
| 21 | OUTLUT5 | Global Mux | Global Mux | Global Mux | Global Mux |
| 22 | OUTLUT6 | Global Mux | Global Mux | Global Mux | Global Mux |
| 23 | OUTLUT7 | Global Mux | Global Mux | Global Mux | Global Mux |

注意：当 CLB 驱动 eQEP 时，不支持以下设置：

PCRM = 0（通过软件索引标记复位）

IEL = 3（通过软件索引标记锁存）

FIDF、FIMF 和 QDLF 不会呈现预期行为

### 10.3.3 CLB 单元块

CLB 单元块的用途是提供 CLB 的逻辑重配置能力。CLB 单元块包含以下子模块：

计数器（Counter）：可配置为加法器、计数器或移位器。作为加法器时可执行加 / 减运算；作为计数器时可向上 / 向下计数；作为移位器时可向左 / 向右移位。计数器的事件输入和复位输入可自由连接到同一单元块内的其他任何子模块。从第二代 CLB 开始，计数器模块还可工作在串行器或线性反馈移位寄存器模式。每个单元块有 3 个计数器。

LUT4：4 输入查找表子模块，可实现最多 4 个输入的任意组合布尔方程。每个 CLB 单元块有 3 个 LUT4 子模块。

有限状态机（FSM）：可配置为单个 4 状态有限状态机，或两个独立的 2 状态有限状态机。FSM 接收两个外部输入，生成两个状态输出和一个组合输出。不用作状态机时，FSM 子模块可接收两个外部输入，用作 4 输入 LUT。每个 CLB 单元块有 3 个 FSM 子模块。

输出 LUT：3 输入查找表子模块，可实现最多 3 个输入的任意组合布尔方程。每个 CLB 单元块有 8 个此类模块，分别与一个单元块输出关联。

异步输出调节块（AOC）：主要用途是对单元块输出或直接对单元块输入提供异步调节能力。

高级控制器（HLC）：事件驱动型模块，可处理最多 4 个并发事件，事件可来自其他任何模块的输出。每个事件发生时，会执行一组预定义操作。HLC 还为 CPU 子系统提供数据交换和中断机制。有 4 个工作寄存器（R0、R1、R2、R3），可用于基本操作，以及修改或设置 3 个计数器模块的值。与其他子模块不同，每个 CLB 单元块仅有一个 HLC。

静态开关块：为上述所有模块提供动态连接能力。用户可连接各子模块，但限制是子模块不能在单元块内形成组合逻辑环路。

一个 CLB 单元块包含 3 组计数器模块、FSM 和 LUT4，1 个高级控制器和 8 个输出 LUT 模块。子模块编号如图所示。

LUT 子模块的功能通过寄存器字段配置，该字段包含所需查找表输出的二进制模式。例如，4 输入 LUT 有 16 种可能的输入组合，每种组合对应输出端所需的二进制 0 或 1。因此，寄存器字段长度可为 16 位，每位代表一种二进制模式的期望结果。输入模式序列从 0000 开始，依次到 1111 结束。FSM 子模块中的 16 位状态方程也采用类似的编码方法。

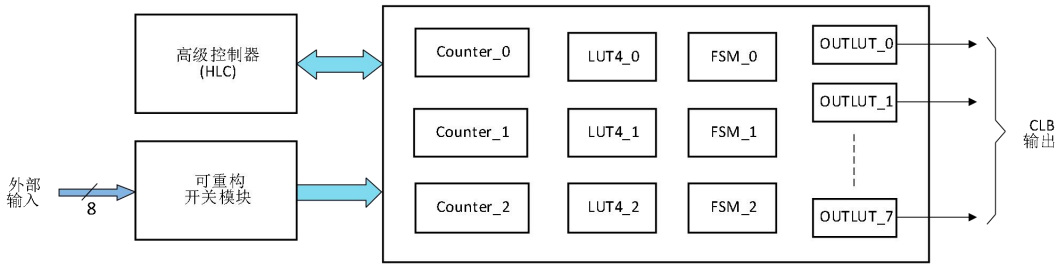



图 10-9 CLB 单元块子模块

## 10.3.4 静态开关块

静态开关块为 CLB 单元块内的子模块提供可配置连接。所有子模块的输出和 8 个外部输入都连接到单元块内的一条公共内部总线。每个输入端口都有一个 32 选 1 多路选择器和相关的 5 位选择值，用户可通过该选择值从总线上选择一个输入。唯一的限制是某些信号（如下所述）在设计中被固定，以防止意外形成组合逻辑环路。

表 10-4 输出表

| 位序 | 信号连接 | 备注 |
|---|---|---|
| 0 | 常 0 | 该选择值用于将输入信号置 0 |
| 1 | COUNTER_0 MATCH2 | |
| 2 | COUNTER_0 ZERO | |
| 3 | COUNTER_0 MATCH1 | |
| 4 | FSM_0 STATE_BIT_0 | |
| 5 | FSM_0 STATE_BIT_1 | |
| 6 | FSM_0 LUT 输出 | |
| 7 | LUT4_0 输出 | |
| 8 | 常 1 | 该选择值用于将输入信号置 1 |

| 位序 | 信号连接 | 备注 |
|---|---|---|
| 9 | COUNTER_1 MATCH2 | |
| 10 | COUNTER_1 ZERO | |
| 11 | COUNTER_1 MATCH1 | |
| 12 | FSM_1 STATE_BIT_0 | |
| 13 | FSM_1 STATE_BIT_1 | |
| 14 | FSM_1 LUT 输出 | |
| 15 | LUT4_1 输出 | |
| 16 | 常 0 | |
| 17 | COUNTER_2 MATCH2 | |
| 18 | COUNTER_2 ZERO | |
| 19 | COUNTER_2 MATCH1 | |
| 20 | FSM_2 STATE_BIT_0 | |
| 21 | FSM_2 STATE_BIT_1 | |
| 22 | FSM_2 LUT 输出 | |
| 23 | LUT4_2 输出 | |
| 24 | 外部输入 0 | |
| 25 | 外部输入 1 | |
| 26 | 外部输入 2 | |
| 27 | 外部输入 3 | |
| 28 | 外部输入 4 | |
| 29 | 外部输入 5 | |
| 30 | 外部输入 6 | |
| 31 | 外部输入 7 | |

表 10-5 输入表

| 模块名 | 接口名 | 描述 |
|---|---|---|
| Counter | RESET | 用作计数器时表现为复位高有效 |
| | MODE_0 | 用作计数器时的使能信号，此输入为 1 时计数器计数 |
| | MODE_1 | 用作计数器时的方向控制信号，输入为 1 时计数器递增，否则递减 |
| LUT | IN0 | 4 输入 LUT 的输入 0 |
| | IN1 | 4 输入 LUT 的输入 1 |
| | IN2 | 4 输入 LUT 的输入 2 |
| | IN3 | 4 输入 LUT 的输入 3 |
| FSM | EXT_IN0 | FSM 模块的输入 0 |
| | EXT_IN1 | FSM 模块的输入 1 |
| | EXTRA_EXT_IN0 | FSM 模块的额外外部输入 0，仅在配置为 LUT 模式时有效 |
| | EXTRA_EXT_IN1 | FSM 模块的额外外部输入 1，仅在配置为 LUT 模式时有效 |

静态开关块允许用户将任何子模块的输入连接到表中的任何输出，因此容易形成组合逻辑环路。为防止这种情况，每个子模块的输入路径中都会断开某些路径，这些端口位置被置 0，如表所示。

表 10-6 为防止组合逻辑环路而置零的端口

| 模块名 | 将输入多路选择器的端口置零，以防止产生组合逻辑环路 |
|---|---|
| LUT_0 | LUT_0、LUT_1、LUT_2 输出，FSM_0、FSM_1、FSM_2 输出 |
| FSM_0 | LUT_1、LUT_2 输出，FSM_0、FSM_1、FSM_2 输出 |
| LUT_1 | LUT_1、LUT_2 输出，FSM_1、FSM_2 输出 |
| FSM_1 | LUT_2 输出，FSM_1、FSM_2 输出 |
| LUT_2 | LUT_2 输出，FSM_2 输出 |
| FSM_2 | FSM_2 输出 |

## 10.3.5 2 计数器模块

### 10.3.5.1 计数器描述

计数器模块是复杂的功能子模块，可配置为计数器、加法器或移位器。除正常运行控制外，该模块还有一个专用的 EVENT 输入，可触发加法、减法或移位操作，或向计数器寄存器加载数据。计数器子模块的输入如图所示。

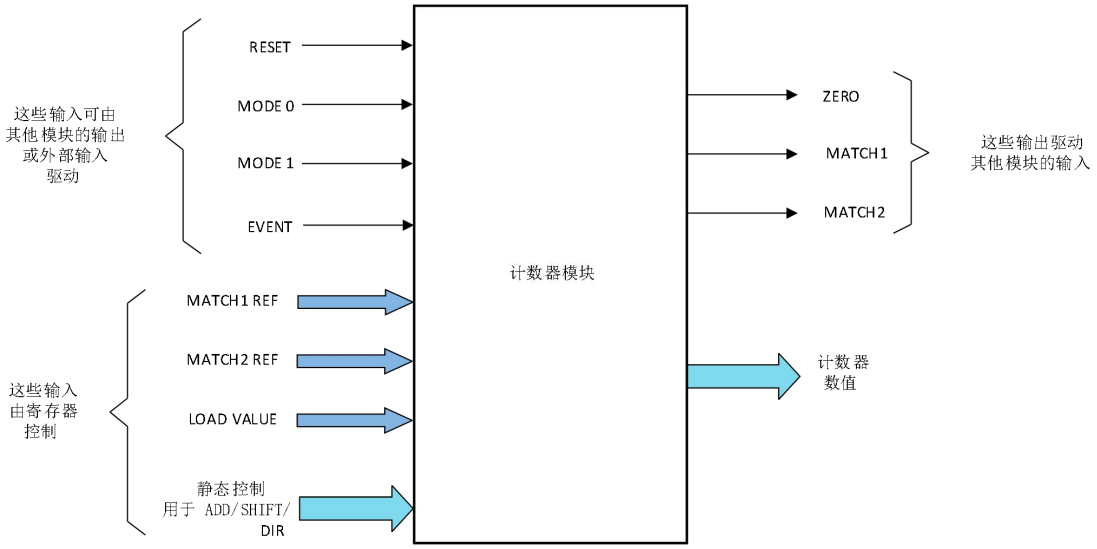



图 10-10 计数器模块

### 10.3.5.2 计数器操作

计数器模块的核心是一个 32 位计数寄存器。该寄存器可在计数开始前静态加载，或在运行时动态加载。计数器子模块的操作由以下输入决定，每个输入都可连接到 CLB 单元块内其他任何模块的输出，这些连接通过配置静态开关块实现。

计数器输入包括：

RESET：最高优先级输入，优先于所有其他输入。输入为电平敏感型，只要输入保持高电平，计数器就会在下一个时钟周期复位为 0。

MODE_0：计数器使能输入。仅当该输入为高电平时，计数器才开始计数（计数方向由 MODE_1 设置决定）；输入为低电平时，不进行计数。

MODE_1：计数器方向控制输入。输入为高电平时，在 MODE_0 为高电平的每个时钟周期，计数器递增；输入为低电平时，在 MODE_0 为高电平的每个时钟周期，计数器递减。向上计数时，计数器在达到 0xFFFF FFFF 后回绕至 0x0000 0000；向下计数时，在达到 0x0000 0000 后回绕至 0xFFFF FFFF。唯一例外是当 EVENT 恰好同时发生时，会向计数器加载不同的值。

EVENT：用于基于特定事件触发计数器操作，事件可来自其他任何模块的输出或单元块的外部输入。计数器的静态控制输入定义了有效事件发生时计数器的行为，有效事件定义为 EVENT 输入的上升沿。计数器可配置为执行以下操作之一：

从 LOAD VALUE 寄存器加载预定义的 32 位值到计数寄存器；

将计数器寄存器的内容向左或向右移位预定义的位数（0-31 位）；

加或减预定义的 32 位值，加减法视为 32 位无符号运算，无饱和处理。

注意：EVENT 输入上升沿的作用仅持续一个周期，下一个周期计数器将根据 MODE_0、MODE_1 和 RESET 输入继续操作。

MATCH1 REF 和 MATCH2 REF 是 32 位参考值，用于生成 MATCH1 和 MATCH2 输出。当计数器寄存器值与 32 位 MATCH1 REF 值匹配时，MATCH1 输出变为高电平；MATCH2 的行为与 MATCH2 REF 寄存器类似。MATCH1 和 MATCH2 的参考值可在操作开始前设置一次，或动态修改。高级控制器可向 MATCH1 REF 和 MATCH2 REF 寄存器加载所需值。

注意：计数器加载寄存器和匹配寄存器不支持内存映射。

计数器模块的三个逻辑输出包括：

ZERO：当计数器寄存器为 0 时，输出为高电平。

MATCH1：当计数器寄存器值等于 MATCH1 REF 输入寄存器值时，输出为高电平。

MATCH2：当计数器寄存器值等于 MATCH2 REF 输入寄存器值时，输出为高电平。

计数器模块的操作由 CFG_MISC_CTRL 寄存器控制，该寄存器的以下三位与每个计数器相关（x 代表计数器实例：0、1 或 2）；

COUNT_EVENT_CTRL_x：定义计数器在事件发生时执行加法还是移位操作。值为 0 时，事件发生时加载静态值；值为 1 时，执行加 / 移位操作。要使计数器的间接加载和 HLC 加载生效，该位必须置 0。

COUNT_ADD_SHIFT_x：值为 1 时执行加法，值为 0 时执行移位。

COUNT_DIR_x：值为 1 时执行左移或加法，值为 0 时执行右移或减法。

表中展示了根据输入和控制寄存器位的计数器模块逻辑操作。当 EVENT=0 时，向上计数和向下计数为正常操作。CNTVAL 寄存器的操作如下：

表 10-7 计数器模块操作模式

| 事件（EVENT） | MODE_0 | MODE_1 | COUNT_EVENT_CTRL_x | COUNT_ADD_SHIFT_x | COUNT_DIR_x | CNTVAL 行为 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | X | X | X | 无 |
| 0 | 0 | 1 | X | X | X | 无 |
| 0 | 1 | 0 | X | X | X | 递减 |
| 0 | 1 | 1 | X | X | X | 递增 |
| 1 | X | X | 0 | X | X | 加载 |
| 1 | X | X | 1 | 0 | 0 | 右移 |
| 1 | X | X | 1 | 0 | 1 | 左移 |
| 1 | X | X | 1 | 1 | 0 | 减 |
| 1 | X | X | 1 | 1 | 1 | 加 |

加载：CNTVAL = EVENT_LOAD_VAL

右移：CNTVAL = CNTVAL >> EVENT_LOAD_VAL

左移：CNTVAL = CNTVAL << EVENT_LOAD_VAL

减：CNTVAL = CNTVAL - EVENT_LOAD_VAL

加：CNTVAL = CNTVAL + EVENT_LOAD_VAL

#### 10.3.5.3 串行器模式

从第二代 CLB 开始，计数器模块可工作在串行器模式。在此模式下，该模块用作移位寄存器（也称为串行器）。EVENT 输入用于向串行器移位 1 位数据，MATCH1 和 MATCH2 均可配置为输出移位寄存器数据。通过 CLB_COUNT_MATCH_TAP_SEL 的 MATCH1/2_TAP_SEL 位，可将计数器的任意位位置引出到 MATCH1/MATCH2 输出。此模式下计数器的移位和方向由 MODE_0（使能）和 MODE_1（方向）控制。

要启用串行器模式，必须设置 CLB_MISC_CONTROL.COUNT_SERIALIZER_0（针对计数器 0）。

#### 10.3.5.4 线性反馈移位寄存器（LFSR）模式

从第二代 CLB 开始，计数器模块可工作在线性反馈移位寄存器模式。通过配置 LFSR 的特性，计数器模块可用于计算串行位流的 CRC。LFSR 的多项式存储在 MATCH2 参考寄存器中，反馈位位置存储在 MATCH1 参考寄存器中。

要启用 LFSR 模式，必须同时设置 CLB_MISC_CONTROL.COUNT_SERIALIZER_0（针对计数器 0）和 COUNT0_LFSR_EN。

通过修改 MODE1 的值（0 或 1），可选择两种类型的 LFSR，如图所示。

LFSR 类型 1 结构（MODE_1 = 0）

CNT_VAL 是 32 位计数器的有效寄存器
CNT_VAL_next 是下一有效周期写入 CNT_VAL 的 32 位值
(MODE_0 == 1 时的时钟沿)
LFSR_MSB = CNT_VAL[MATCH1_REF[4:0]]
Poly[31:0] 为用作 CRC 多项式的 MATCH2_REF

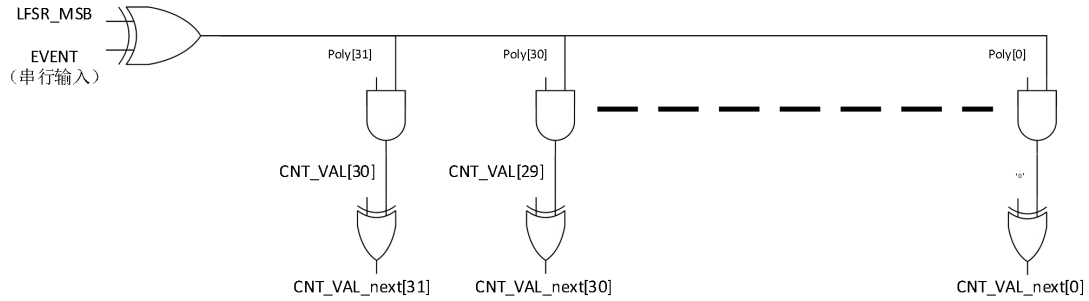



图 10-11 线性反馈移位寄存器（LFSR）模式示意图

LFSR 类型 2 结构（MODE_1 = 1）

CNT_VAL 是 32 位计数器寄存器
CNT_VAL_next 是下一有效周期写入 CNT_VAL 的 32 位值
(MODE_0 == 1 时的时钟沿)
LFSR_MSB = CNT_VAL[MATCH1_REF[4:0]]
Poly[31:0] 为用作 CRC 多项式的 MATCH2_REF

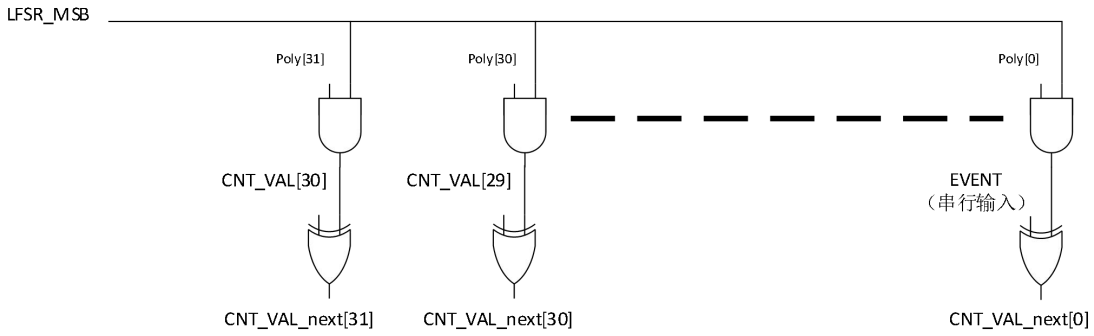



图 10-12 LFSR 模式

## 10.3.5.5 FSM 模块

有限状态机（FSM）模块可构建最多 4 个状态的可编程有限状态机。FSM 模块有两个寄存器位和两个外部输入，可编程为两个 2 状态机或一个 4 状态机。为提高灵活性，还有两个辅助输入（EXTRA_EXT_IN0 和 EXTRA_EXT_IN1），可通过放弃状态功能创建更大的组合函数。FSM 的结构如图所示。

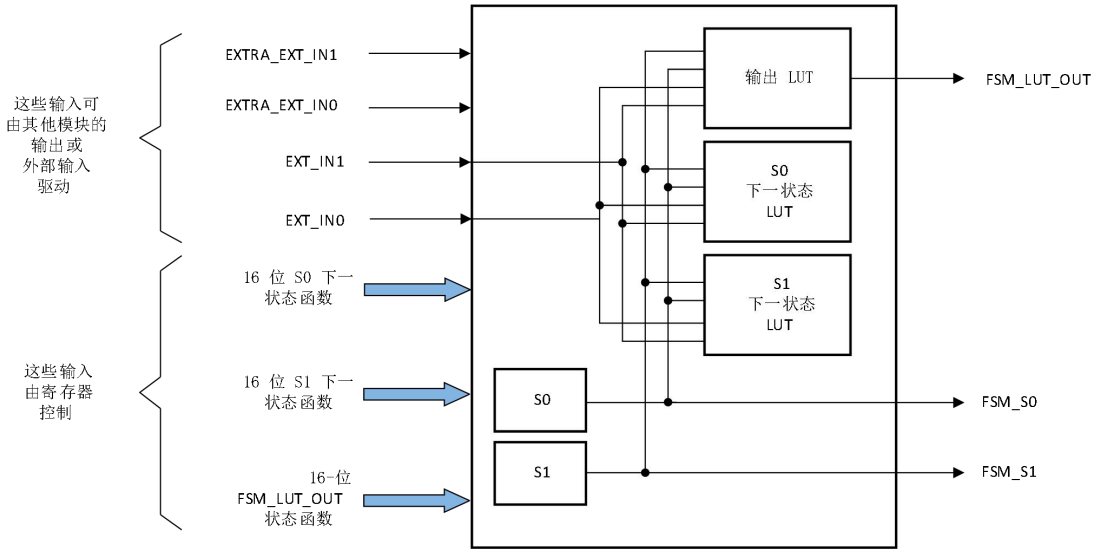



图 10-13 FSM 模块

FSM 模块的信号和功能如下：

EXT_IN0 和 EXT_IN1：两个外部输入，可用于控制输出 FSM_LUT_OUT 或状态 S0 和 S1 中的任意一个。

S0 和 S1：两个状态位，具有独立的状态控制方程。

16 位 S0 方程：定义函数（EXT_IN1、EXT_IN0、S1、S0），按定义顺序的四位用作 16 位寄存器的索引，以确定 S0 的次态。

16 位 S1 方程：定义函数（EXT_IN1、EXT_IN0、S1、S0），按定义顺序的四位用作 16 位寄存器的索引，以确定 S1 的次态。

16 位输出方程：定义函数（EXT_IN1、EXT_IN0、S1、S0），按定义顺序的四位用作 16 位寄存器的索引，以确定 FSM_LUT_OUT 的输出值。提供额外的可配置层级，若未使用状态 S0 和 S1，FSM_LUT_OUT 可使用额外输入。

通过一个额外位选择 EXTRA_EXT_IN0 替代 S0，另一个额外位选择 EXTRA_EXT_IN1 替代 S1。这样，通过分别放弃一个或两个状态位，可有效为 FSM_LUT_OUT 构建 3 输入或 4 输入 LUT。

CFG_MISC_CTRL 寄存器控制 FSM 模块的操作，每个 FSM 模块使用该寄存器的两位，以确定 FSM 输出 LUT 函数使用状态变量 S0/S1 还是相应的额外外部

输入信号 FSM_EXTRA_EXT_INx。值为 0 时使用状态位，值为 1 时使用 FSM_EXTRA_EXT_IN0/FSM_EXTRA_EXT_IN1 信号。

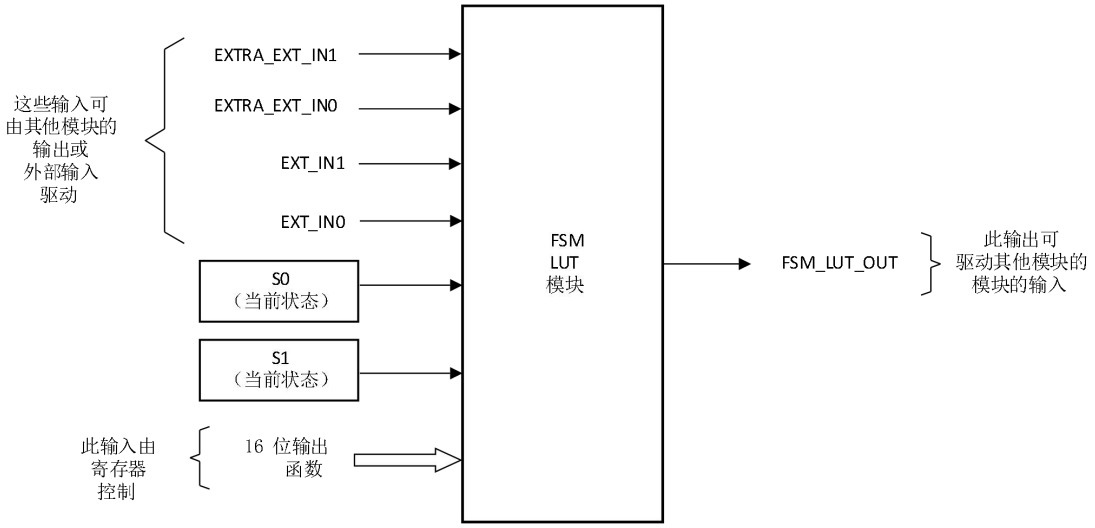



图 10-14 FSM LUT 模块

### 10.3.5.6 LUT4 模块

简单的 4 输入查找表（LUT）模块，输入为 IN0、IN1、IN2 和 IN3（如图所示）。通过编程每个 LUT4 模块关联的 16 位控制寄存器，可实现使用这四个输入的任意组合布尔方程。

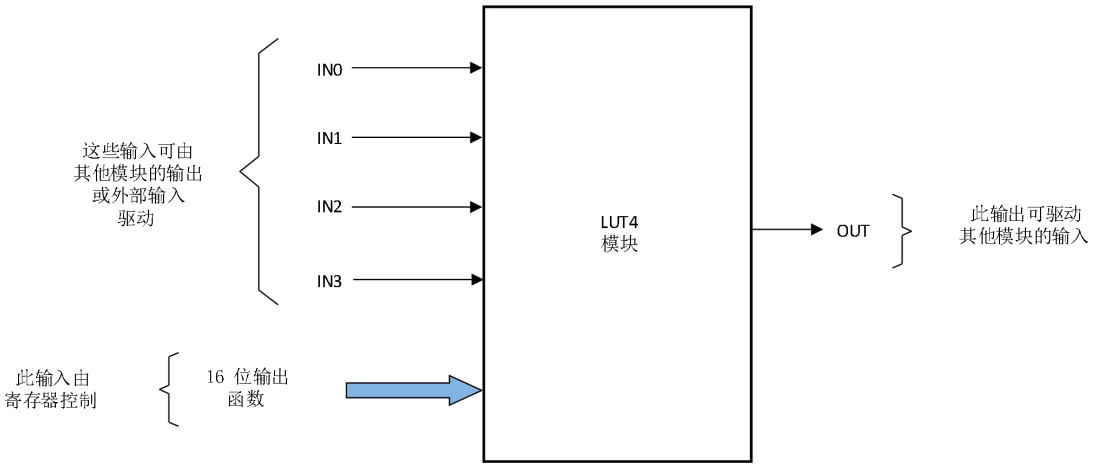



图 10-15 LUT4 模块

### 10.3.5.7 输出 LUT 模块

输出 LUT 模块的功能与 LUT4 模块非常相似，但仅有三个输入。与其他子模块不同，这些模块的输出用于引出单元块，因此不能被单元块内的其他任何模块使用。输出 LUT 模块可实现三个输入的任意组合函数。

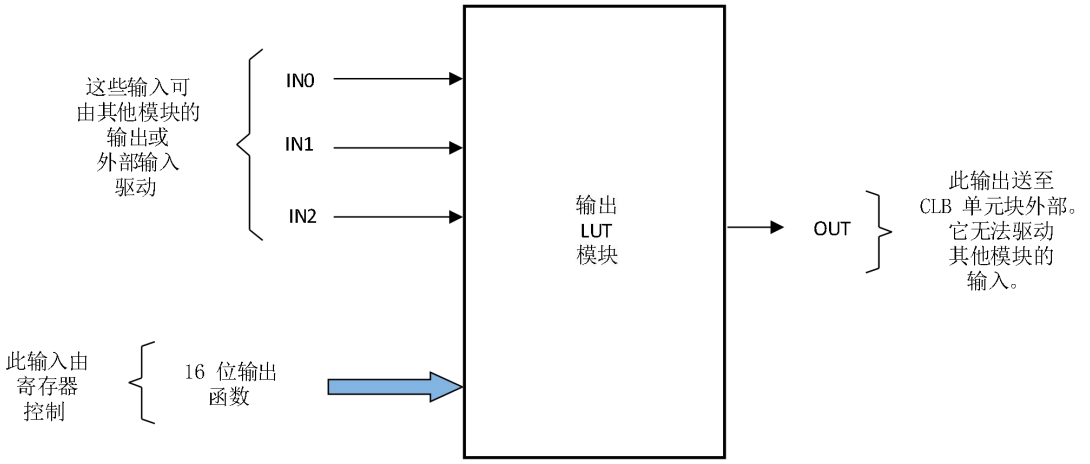



图 10-16 输出 LUT 模块

### 10.3.5.8 异步输出调节（AOC）模块

AOC 模块的逻辑分为三个阶段，输入在进入下一级之前，每个阶段都会经过不同类型的逻辑修改，如图所示。

该模块有 8 个输入，每个输入可选择 CLB 的相应边界输入或 CLB 单元块输出（例如，AOC 模块的输入 0 可在 CLB 边界输入 0 和 CLB 单元块输出 0 之间选择）。若选择 CLB 单元块输出 0，该输出在发送到后续异步信号调节阶段之前始终会被寄存。每个阶段都提供旁路选项，可直接将信号原样发送到下一级。

阶段 1：输入信号可在发送到下一级之前进行反相。

阶段 2：来自阶段 1 的信号可与门控控制信号进行门控操作，门控控制信号可来自软件寄存器或任意 CLB 单元块输出，门控功能可为逻辑与、或、异或。

阶段 3：输入信号可在其上升沿置位或清零输出，这是纯异步的置位 / 清零操作，无需时钟。释放控制信号为高电平时，将输出恢复到默认状态（选择异步清零时默认状态为高电平，选择异步置位时默认状态为低电平）。释放控制信号可来自软件

寄存器或任意 CLB 单元块输出。此外，也可选择将输入信号延迟一个时钟周期，而非执行异步置位和清零操作。

CLB 单元块与 AOC 模块的交互如图所示。

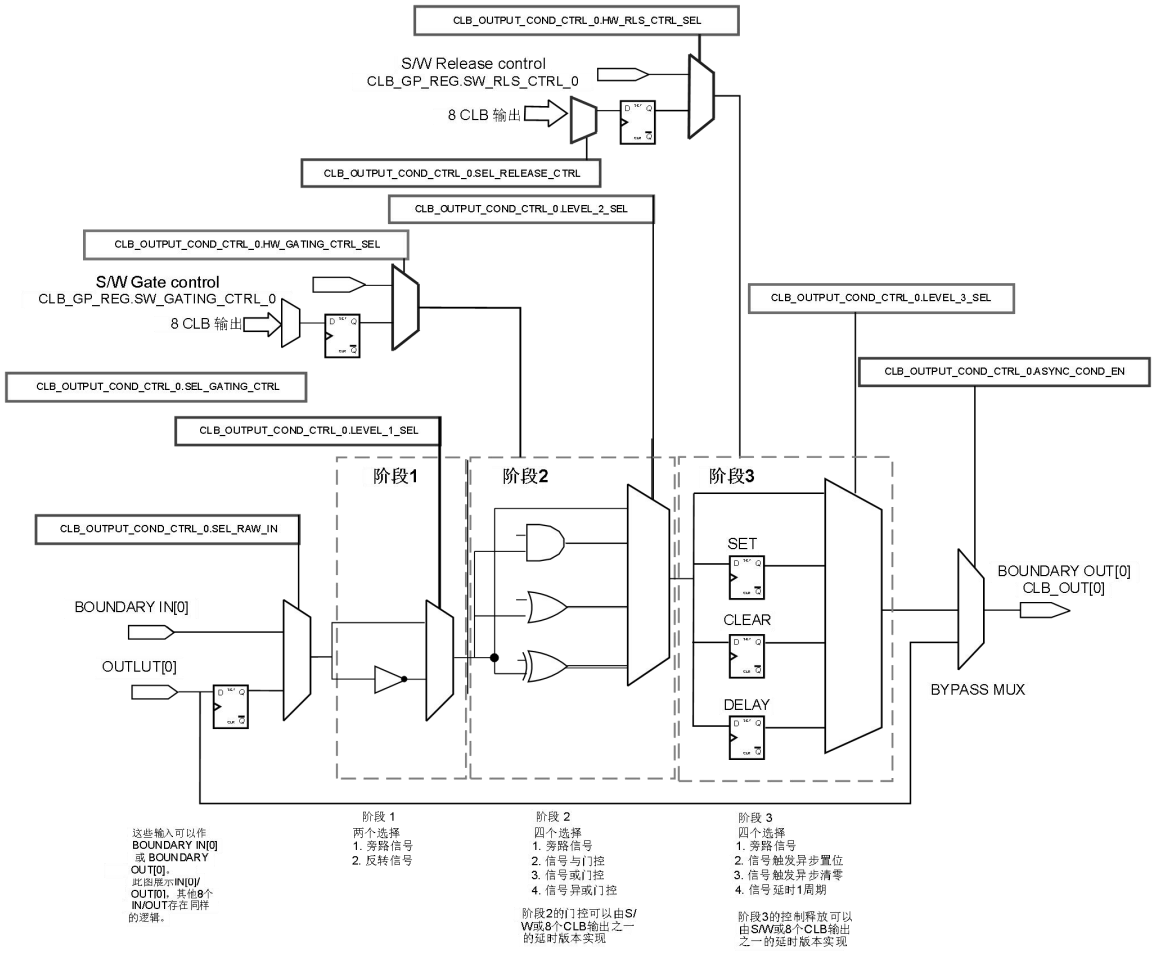



图 10-17 AOC 模块

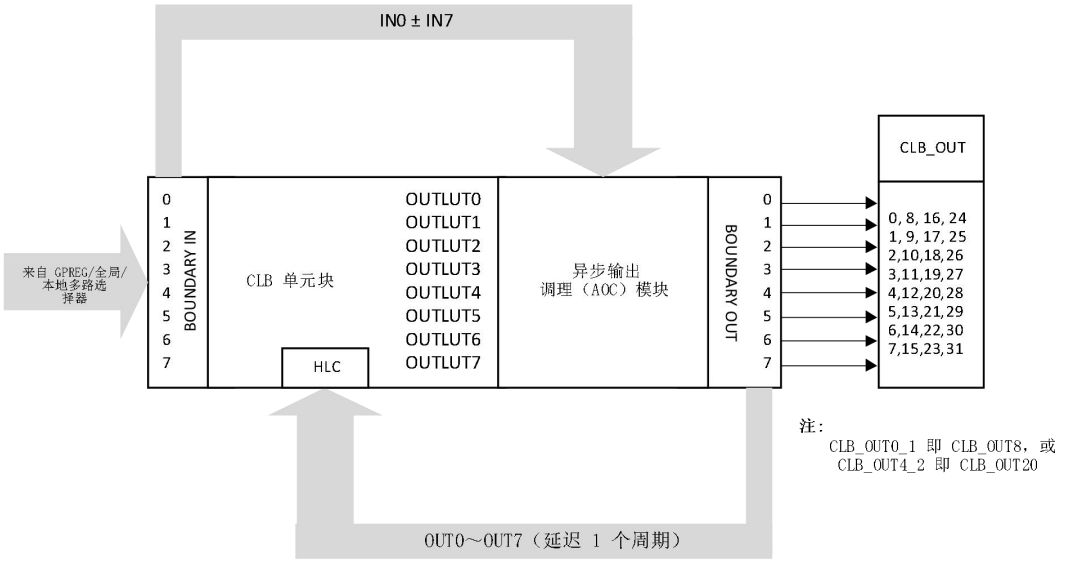



图 10-18 AOC 模块与 CLB 单元块

注意：仅 CLB_OUT12 至 CLB_OUT15 可用作异步输出，与 CLB_OUT4_1 至 CLB_OUT7_1 相同。可通过 GPIO 输出交叉开关将 OUT4_1 和 OUT5_1 异步输出路由到 GPIO。

### 10.3.5.9 高级控制器（HLC）

高级控制器（HLC）比 CLB 单元块内的其他模块复杂得多，主要执行两项功能：

提供与器件其他部分的通信和数据交换功能，通过两种方式实现：一是对四个通用 HLC 寄存器（R0-R3）的全局访问路径；二是 CPU 与 HLC 之间的 PUSH 和 PULL FIFO。通用 HLC 寄存器设计为仅在器件配置时写入，为避免意外行为，运行时不得写入。PUSH 和 PULL FIFO 是运行时数据交换的主要途径。

提供可编程的事件驱动型动作系统，执行计算、逻辑功能操作和数据移动。换句话说，可配置事件触发 CLB 单元块内的一组预定义动作，或启动与器件其他部分的数据交换。

HLC 的架构如图所示。HLC 是事件驱动型系统，可处理最多 4 个并发事件，事件可从单元块内其他任何模块的输出或外部输入中选择。

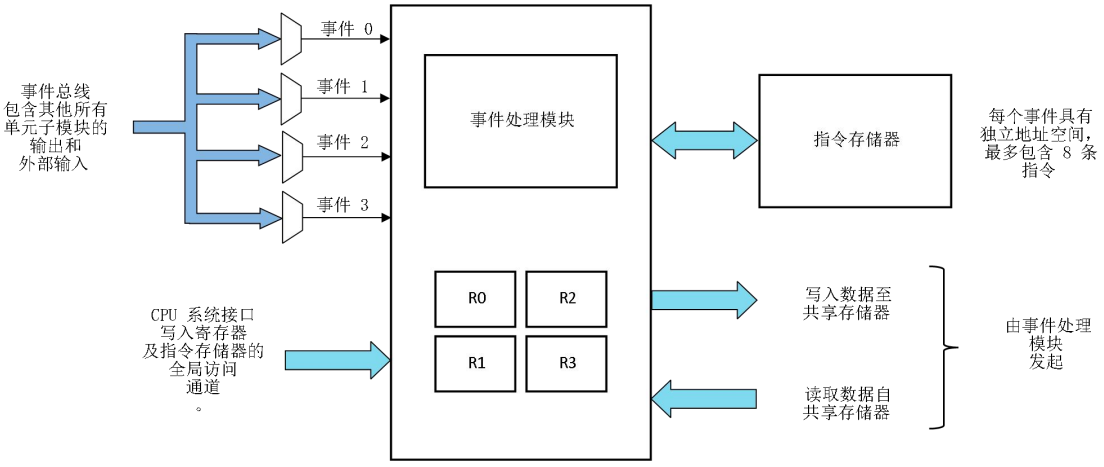



图 10-19 HLC 结构

## 10.3.5.10 高级控制器事件

四个 HLC 事件中的每个事件都有一个专用地址，用于执行指令。事件从表中列出的信号中选择，编号最小的事件（事件 0）优先级最高，编号最大的事件（事件 3）优先级最低。

表 10-8 HLC 事件列表

| 索引 | HLC 事件选择 |
|---|---|
| 0 | 恒 0 |
| 1 | COUNTER_0 MATCH2 |
| 2 | COUNTER_0 ZERO |
| 3 | COUNTER_0 MATCH1 |
| 4 | FSM_0 STATE_BIT_0 |
| 5 | FSM_0 STATE_BIT_1 |
| 6 | FSM_0 LUT 输出 |
| 7 | LUT4_0 输出 |
| 8 | 恒 1 |
| 9 | COUNTER_1 MATCH2 |
| 10 | COUNTER_1 ZERO |
| 11 | COUNTER_1 MATCH1 |
| 12 | FSM_1 STATE_BIT_0 |
| 13 | FSM_1 STATE_BIT_1 |
| 14 | FSM_1 LUT 输出 |
| 15 | LUT4_1 输出 |
| 16 | 恒 0 |
| 17 | COUNTER_2 MATCH2 |
| 18 | COUNTER_2 ZERO |
| 19 | COUNTER_2 MATCH1 |
| 20 | FSM_2 STATE_BIT_0 |
| 21 | FSM_2 STATE_BIT_1 |
| 22 | FSM_2 LUT 输出 |

| 索引 | HLC 事件选择 |
|---|---|
| 23 | LUT4_2 输出 |
| 24 | 外部输入 0 |
| 25 | 外部输入 1 |
| 26 | 外部输入 2 |
| 27 | 外部输入 3 |
| 28 | 外部输入 4 |
| 29 | 外部输入 5 |
| 30 | 外部输入 6 |
| 31 | 外部输入 7 |

从第二代 CLB 开始，提供额外的 HLC 输入，可通过选择 HLC 模块的备用多路选择器选项（HLC_ALT_MUX_SEL_n = 1）进行选择，如表所示。

表 10-9 HLC 备用事件列表

| 索引 | HLC 事件选择 |
|---|---|
| 0 | CLB_OUT_0 |
| 1 | CLB_OUT_1 |
| 2 | CLB_OUT_2 |
| 3 | CLB_OUT_3 |
| 4 | CLB_OUT_4 |
| 5 | CLB_OUT_5 |
| 6 | CLB_OUT_6 |
| 7 | CLB_OUT_7 |
| 8 | CLB_OUT_0.INVERTED |
| 9 | CLB_OUT_1.INVERTED |
| 10 | CLB_OUT_2.INVERTED |
| 11 | CLB_OUT_3.INVERTED |
| 12 | CLB_OUT_4.INVERTED |
| 13 | CLB_OUT_5.INVERTED |
| 14 | CLB_OUT_6.INVERTED |
| 15 | CLB_OUT_7.INVERTED |
| 16 | CLB_ASYNC_OUT_0 |
| 17 | CLB_ASYNC_OUT_1 |
| 18 | CLB_ASYNC_OUT_2 |
| 19 | CLB_ASYNC_OUT_3 |
| 20 | CLB_ASYNC_OUT_4 |
| 21 | CLB_ASYNC_OUT_5 |
| 22 | CLB_ASYNC_OUT_6 |
| 23 | CLB_ASYNC_OUT_7 |
| 24 | CLB_ASYNC_OUT_0.INVERTED |
| 25 | CLB_ASYNC_OUT_1.INVERTED |
| 26 | CLB_ASYNC_OUT_2.INVERTED |
| 27 | CLB_ASYNC_OUT_3.INVERTED |
| 28 | CLB_ASYNC_OUT_4.INVERTED |
| 29 | CLB_ASYNC_OUT_5.INVERTED |
| 30 | CLB_ASYNC_OUT_6.INVERTED |
| 31 | CLB_ASYNC_OUT_7.INVERTED |

### 10.3.5.11 高级控制器指令

指令存储器每个事件最多支持 8 条指令，每个指令序列在对应事件的上升沿触发。从第二代 CLB 开始，支持通过上升沿和下降沿两种方式触发指令执行。

若多个事件同时触发，相关指令序列会按优先级顺序依次执行。

表 10-10 HLC 指令地址范围

| 地址 | 指令 |
|---|---|
| 00000~00111 | 事件 0 |
| 01000~01111 | 事件 1 |
| 10000~10111 | 事件 2 |
| 11000~11111 | 事件 3 |

HLC 指令格式如表所示。

表 10-11 HLC 指令格式

| 最低指令位 | 操作码 | 源（3 位） | 目的（3 位） |
|---|---|---|---|
| 该位置 1 时，会在当前指令执行完毕后终止指令流运行 | MOV（00000） | 源可以是 R0、R1、R2、R3、C0、C1、C2 | 目的可以是 R0、R1、R2、R3、C0、C1、C2 |
| | MOV_T1（00001） | | 注意：ADD/SUB 指令的目的只能是 R0、R1、R2 或 R3 |
| | MOV_T2（00010） | | |
| | PUSH（00011） | | |
| | PULL（00100） | | |
| | ADD（00101） | | |
| | SUB（00110） | | |
| | INTR（00111） | | |

R0、R1、R2、R3 是 HLC 中的 4 个 32 位通用寄存器，C0、C1、C2 是 CLB Tile 中的 3 个计数器寄存器。<源> 用于表示数据来源，< 目的 > 用于表示数据目标。HLC 指令的详细描述如表所示。

表 10-12 HLC 指令描述

| 指令 | 描述 |
|---|---|
| ADD <源>, < 目的 > | 执行无符号 32 位加法运算，运算规则为 <目的>=< 目的 >+< 源 >。< 源 > 可以是 R0、R1、R2、R3、C0、C1 或 C2，< 目的 > 只能是 R0、R1、R2 或 R3。 |
| INTR <6 位常数> | 通过 CPU 接口触发中断，6 位常量会存储至中断标志寄存器 CLB_INTR_TAG_REG。若连续调用多条 INTR 指令，仅第一条生效；若需多次调用，需在各条 INTR 指令之间插入其他 HLC 指令，确保所有中断调用均能生效。 |

| 指令 | 描述 |
| --- | --- |
| | 注意：从第二代 CLB 开始，该模块可产生不可屏蔽中断（NMI），此功能默认禁用，需通过配置寄存器位 CLB_LOAD_EN.NMI_EN 手动使能。 |
| MOV <源>, < 目的 > | 将 <源> 的数据传输至 < 目的 >。< 源 > 和 < 目的 > 均可为 R0、R1、R2、R3、C0、C1 或 C2 中的任意一个。对于计数器的间接加载和 HLC 加载操作，必须将 COUNT_EVENT_CTRL_x 位配置为 “加载模式”（即置 0），才能确保加载操作生效。 |
| MOV_T1 <源>, < 目的 > | 将 <源> 的数据写入 < 目的 > 计数器的 Match1 寄存器。< 源 > 可以是寄存器 R0、R1、R2、R3 中的任意一个，也可以是与 C0、C1、C2 关联的计数器数值；< 目的 > 指计数器 C0、C1、C2 中任意一个的 Match1 寄存器。<br>示例：<br>·将 C1 中的计数值写入寄存器 R0：MOV_T1 C1 R0<br>·将 R2 中的数值写入计数器 C0 的 Match1 寄存器：MOV_T1 R2 C0 |
| MOV_T2 <源>, < 目的 > | 与 MOV_T1 指令功能类似，将 <源> 的数据写入 < 目的 > 计数器的 Match2 寄存器。< 源 > 可以是寄存器 R0、R1、R2、R3 中的任意一个，也可以是与 C0、C1、C2 关联的计数器数值；< 目的 > 指计数器 C0、C1、C2 中任意一个的 Match2 寄存器。 |
| PULL <目的> | 将 CPU 接口中数据交换拉取内存缓冲区的数据传输至 <目的> 寄存器，< 目的 > 可选取 R0、R1、R2 或 R3 中的任意一个。从 HLC 视角来看，该指令会从内部 4 字深度的先进先出（FIFO）缓冲区中读取（拉取）数据。<br>注意：若使用多个 HLC EVENT 通道，禁止连续使用 PUSH/PULL 指令（即不能背靠背执行）。 |
| PUSH <源> | 将 <源> 的数据传输至 CPU 接口中的数据交换推送内存缓冲区。< 源 > 可以是寄存器 R0、R1、R2、R3 或计数器 C0、C1、C2 中的任意一个。从 HLC 视角来看，该指令会将数据推送至内部 4 字深度的先进先出（FIFO）缓冲区中。 |
| SUB <源>, < 目的 > | 执行无符号 32 位减法运算，运算规则为 <目的> = < 目的 > - < 源 >。< 源 > 可以是 R0、R1、R2、R3、C0、C1 或 C2 中的任意一个，< 目的 > 仅能为 R0、R1、R2 或 R3 之一。 |

MOV、MOV_T1、MOV_T2、ADD、SUB 及 INTR 指令的执行耗时为一个时钟周期；PUSH 和 PULL 指令需两个时钟周期完成执行。需注意，PUSH 与 PULL 指令具备流水线保护机制，即对某一寄存器执行 PUSH/PULL 操作后，该寄存器可被立即调用。

当多个事件被同时触发时，若高优先级事件的最后一条指令为 PUSH 或 PULL，在该高优先级事件结束与下一事件启动之间，会产生一个额外的时钟周期延迟；若最后一条指令非 PUSH 或 PULL，则事件之间无时钟周期延迟。

若存在多个事件进入 HLC 的可能性，则不得连续排布 PUSH/PULL 指令；需对 HLC 指令进行重排，确保每条 PUSH 和 PULL 指令后均紧跟非 PUSH/PULL 类指

令。若任意时刻仅有一个事件可进入 HLC，则连续排布 PUSH/PULL 指令不会产生问题。

### 10.3.5.12 3 <源> 和 < 目的 >

<源> 和 < 目的 > 寄存器通过 3 位编码表示，具体编码规则如表所示。

表 10-13 HLC 寄存器编码

| 位 | 寄存器 |
|---|---|
| 000 | R0 |
| 001 | R1 |
| 010 | R2 |
| 011 | R3 |
| 100 | C0 |
| 101 | C1 |
| 110 | C2 |

MOV、MOV_T1、MOV_T2、ADD、SUB 和 INTR 指令的执行需 1 个周期，PUSH 和 PULL 指令的执行需 2 个周期。注意，PUSH 和 PULL 指令具备流水线保护功能，即对某个寄存器执行 PUSH/PULL 操作后，可立即使用该寄存器。

若多个事件同时触发，若高优先级事件的最后一条指令是 PUSH 或 PULL，则高优先级事件结束与下一个事件开始之间会存在额外的 1 个周期延迟；若最后一条指令不是 PUSH 或 PULL，则事件之间无周期延迟。

若存在多个事件可能进入 HLC 的情况，不得将 PUSH/PULL 指令背靠背放置，需重新排序 HLC 指令，确保每个 PUSH 和 PULL 指令后都紧跟一条非 PUSH/PULL 指令。若任意时刻仅有一个事件能进入 HLC，则 PUSH/PULL 指令背靠背执行无问题。

### 10.3.5.13 4 PUSH 和 PULL 指令的操作（溢出和下溢检测）

HLC 的 PUSH 和 PULL 操作用于与主机系统进行数据交换，两者分别对应独立的 FIFO 缓冲区。例如，一系列 PUSH 操作会写入线性映射内存缓冲区的连续地址。

CPU 可分别对 PUSH 缓冲区执行读操作、对 PULL 缓冲区执行写操作，以与 HLC 交换数据。HLC 推送的数据由 CPU 从 PUSH 缓冲区读取，CPU 发送给 HLC 的数据由 CPU 写入 PULL 缓冲区，再由 HLC 通过 PULL 指令读取。

若需将某个 CLB 输入用作通用寄存器（GPREG），可通过该输入指示 CPU 何时向 FIFO 写入数据。

PUSH 和 PULL 操作各有独立的地址指针，每次 HLC 执行 PUSH 或 PULL 操作时，对应的地址指针会递增。这些地址指针支持内存映射，CPU 可读取其数值，也可随时对其进行写操作或重置。

溢出和下溢检测可通过直接读取 PUSH 和 PULL 地址指针的数值实现。本设备的 CLB 模块中，PUSH 和 PULL FIFO 的深度均为 4 个 32 位字。若 CPU 向 PULL 缓冲区启动新的数据传输时，发现地址指针数值大于 4，则表示发生下溢，即 HLC 从缓冲区读取的数据量超过了 CPU 写入的数据量。

注意：若启用了多个 HLC 事件，背靠背执行 PUSH 和 PULL 指令（无论顺序如何）均无法正常工作。

## 10.4 寄存器描述

### 10.4.1 CLB 寄存器基地址

表 10-14 CLB 寄存器基地址

| CLB实例 | 寄存器组 | DriverLib名称 | 基地址 |
|---|---|---|---|
| Clb1LogicCfgRegs | CLB_LOGIC_CONFIG_REGS | CLB1_LOGICCFG_BASE | 0x0061_0000 |
| Clb1LogicCtrlRegs | CLB_LOGIC_CONTROL_REGS | CLB1_LOGICCTRL_BASE | 0x0061_0200 |
| Clb1DataExchRegs | CLB_DATA_EXCHANGE_REGS | CLB1_DATAEXCH_BASE | 0x0061_0400 |
| Clb2LogicCfgRegs | CLB_LOGIC_CONFIG_REGS | CLB2_LOGICCFG_BASE | 0x0061_0800 |
| Clb2LogicCtrlRegs | CLB_LOGIC_CONTROL_REGS | CLB2_LOGICCTRL_BASE | 0x0061_0A00 |
| Clb2DataExchRegs | CLB_DATA_EXCHANGE_REGS | CLB2_DATAEXCH_BASE | 0x0061_0C00 |

### 10.4.2 CLB_LOGIC_CONFIG_REGS 寄存器

CLB_LOGIC_CONFIG_REGS 寄存器组的存储映射偏移地址如下表所示，表中未列出的偏移地址都是保留地址空间。

表 10-15 CLB_LOGIC_CONFIG_REGS

| 偏移地址 | 寄存器标识 | 寄存器名称 |
|---|---|---|
| 0x4 | CLB_COUNT_RESET | Counter RESET输入源 |
| 0x8 | CLB_COUNT_MODE_1 | Counter MODE_1输入源 |
| 0xC | CLB_COUNT_MODE_0 | Counter MODE_0输入源 |
| 0x10 | CLB_COUNT_EVENT | Counter EVENT输入源 |
| 0x14 | CLB_FSM_EXTRA_IN0 | FSM EXTRA_EXT_IN0输入源 |
| 0x18 | CLB_FSM_EXTERNAL_IN0 | FSM EXT_IN0输入源 |
| 0x1C | CLB_FSM_EXTERNAL_IN1 | FSM EXT_IN1输入源 |
| 0x20 | CLB_FSM_EXTRA_IN1 | FSM EXTRA_EXT_IN1输入源 |
| 0x24 | CLB_LUT4_IN0 | LUT4 IN0输入源 |
| 0x28 | CLB_LUT4_IN1 | LUT4 IN1输入源 |
| 0x2C | CLB_LUT4_IN2 | LUT4 IN2输入源 |
| 0x30 | CLB_LUT4_IN3 | LUT4 IN3输入源 |
| 0x38 | CLB_FSM_LUT_FN1_0 | FSM_1和FSM_0的LUT函数 |
| 0x3C | CLB_FSM_LUT_FN2 | FSM_2的LUT函数 |
| 0x40 | CLB_LUT4_FN1_0 | LUT4_1和LUT4_0的LUT函数 |
| 0x44 | CLB_LUT4_FN2 | LUT4_2的LUT函数 |
| 0x48 | CLB_FSM_NEXT_STATE_0 | FSM_0的Next-State状态函数 |
| 0x4C | CLB_FSM_NEXT_STATE_1 | FSM_1的Next-State状态函数 |
| 0x50 | CLB_FSM_NEXT_STATE_2 | FSM_2的Next-State状态函数 |
| 0x54 | CLB_MISC_CONTROL | Counter和FSM的静态控制 |
| 0x58 | CLB_OUTPUT_LUT_0 | OUTLUT0的输入源和LUT函数 |
| 0x5C | CLB_OUTPUT_LUT_1 | OUTLUT1的输入源和LUT函数 |
| 0x60 | CLB_OUTPUT_LUT_2 | OUTLUT2的输入源和LUT函数 |
| 0x64 | CLB_OUTPUT_LUT_3 | OUTLUT3的输入源和LUT函数 |
| 0x68 | CLB_OUTPUT_LUT_4 | OUTLUT4的输入源和LUT函数 |
| 0x6C | CLB_OUTPUT_LUT_5 | OUTLUT5的输入源和LUT函数 |
| 0x70 | CLB_OUTPUT_LUT_6 | OUTLUT6的输入源和LUT函数 |
| 0x74 | CLB_OUTPUT_LUT_7 | OUTLUT7的输入源和LUT函数 |
| 0x78 | CLB_HLC_EVENT_SEL | HLC的EVENT选择器 |
| 0x7C | CLB_COUNT_MATCH_TAP_SEL | MATCH1和MATCH2输出的Counter数位选择 |
| 0x80 | CLB_OUTPUT_COND_CTRL_0 | OUT0的输出调节控制 |

| 偏移地址 | 寄存器标识 | 寄存器名称 |
|---|---|---|
| 0x84 | CLB_OUTPUT_COND_CTRL_1 | OUT1的输出调节控制 |
| 0x88 | CLB_OUTPUT_COND_CTRL_2 | OUT2的输出调节控制 |
| 0x8C | CLB_OUTPUT_COND_CTRL_3 | OUT3的输出调节控制 |
| 0x90 | CLB_OUTPUT_COND_CTRL_4 | OUT4的输出调节控制 |
| 0x94 | CLB_OUTPUT_COND_CTRL_5 | OUT5的输出调节控制 |
| 0x98 | CLB_OUTPUT_COND_CTRL_6 | OUT6的输出调节控制 |
| 0x9C | CLB_OUTPUT_COND_CTRL_7 | OUT7的输出调节控制 |

下表列出了寄存器描述使用的访问类型编码：

表 10-16 CLB_LOGIC_CONFIG_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1C | W<br>1C | 写<br>1清除 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i、j、k、l、m、n | | 用于寄存器名称、偏移地址或地址时，表示属于重复寄存器组的寄存器数组值。寄存器组按层次组织，数组以公式表示。 |
| y | | 用于寄存器名称、偏移地址或地址时，表示寄存器数组值。 |

### 10.4.2.1 CLB_COUNT_RESET 寄存器

偏移地址：4h　复位值：00000000h

计数器模块 RESET

表 10-17 CLB_COUNT_RESET 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-18 CLB_COUNT_RESET 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的计数器复位输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的计数器复位输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的计数器复位输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.2 CLB_COUNT_MODE_1 寄存器

偏移地址：8h　复位值：00000000h

计数器模块 MODE_1

表 10-19 CLB_COUNT_MODE_1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-20 CLB_COUNT_MODE_1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的计数器 MODE_1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的计数器 MODE_1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的计数器 MODE_1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.3 CLB_COUNT_MODE_0 寄存器

偏移地址：Ch　复位值：00000000h

计数器模块 MODE_0

表 10-21 CLB_COUNT_MODE_0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-22 CLB_COUNT_MODE_0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的计数器 MODE_0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的计数器 MODE_0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的计数器 MODE_0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.4 CLB_COUNT_EVENT 寄存器

偏移地址：10h　复位值：00000000h

计数器模块 EVENT

表 10-23 CLB_COUNT_EVENT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-24 CLB_COUNT_EVENT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的计数器事件输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的计数器事件输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的计数器事件输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.5 CLB_FSM_EXTRA_IN0 寄存器

偏移地址：14h　复位值：00000000h

FSM 额外外部输入 EXT_IN0

表 10-25 CLB_FSM_EXTRA_IN0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-26 CLB_FSM_EXTRA_IN0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的FSM 模块额外外部 IN0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的FSM 模块额外外部 IN0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的FSM 模块额外外部 IN0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.6 CLB_FSM_EXTERNAL_IN0 寄存器

偏移地址：18h　复位值：00000000h

FSM 外部输入 EXT_IN0

表 10-27 CLB_FSM_EXTERNAL_IN0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-28 CLB_FSM_EXTERNAL_IN0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的FSM 模块 EXT_IN0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的FSM 模块 EXT_IN0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的FSM 模块 EXT_IN0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.7 CLB_FSM_EXTERNAL_IN1 寄存器

偏移地址：1Ch　复位值：00000000h

FSM 外部输入 EXT_IN1

表 10-29 CLB_FSM_EXTERNAL_IN1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-30 CLB_FSM_EXTERNAL_IN1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的FSM 模块 EXT_IN1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的FSM 模块 EXT_IN1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的FSM 模块 EXT_IN1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.8 CLB_FSM_EXTRA_IN1 寄存器

偏移地址：20h　复位值：00000000h

FSM 额外外部输入 EXT_IN1

表 10-31 CLB_FSM_EXTRA_IN1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-32 CLB_FSM_EXTRA_IN1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的FSM 模块额外外部 IN1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的FSM 模块额外外部 IN1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的FSM 模块额外外部 IN1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.9 CLB_LUT4_IN0 寄存器

偏移地址：24h　复位值：00000000h

LUT4_0/1/2 的 IN0 输入源

表 10-33 CLB_LUT4_IN0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-34 CLB_LUT4_IN0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的LUT4 模块 IN0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的LUT4 模块 IN0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的LUT4 模块 IN0输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.10 CLB_LUT4_IN1 寄存器

偏移地址：28h　复位值：00000000h

LUT4_0/1/2 的 IN1 输入源

表 10-35 CLB_LUT4_IN1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-36 CLB_LUT4_IN1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的LUT4 模块 IN1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的LUT4 模块 IN1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的LUT4 模块 IN1输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.11 CLB_LUT4_IN2 寄存器

偏移地址：2Ch　复位值：00000000h

LUT4_0/1/2 的 IN2 输入源

表 10-37 CLB_LUT4_IN2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-38 CLB_LUT4_IN2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的LUT4 模块 IN2输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的LUT4 模块 IN2输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的LUT4 模块 IN2输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.12 CLB_LUT4_IN3 寄存器

偏移地址：30h　复位值：00000000h

LUT4_0/1/2 的 IN3 输入源

表 10-39 CLB_LUT4_IN3 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | SEL_2 | | | | | SEL_1 | |
| R/W1C-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_1 | | | SEL_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-40 CLB_LUT4_IN3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14-10 | SEL_2 | R/W | 0h | 单元 2 的LUT4 模块 IN3输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | SEL_1 | R/W | 0h | 单元 1 的LUT4 模块 IN3输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | SEL_0 | R/W | 0h | 单元 0 的LUT4 模块 IN3输入选择。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.13 CLB_FSM_LUT_FN1_0 寄存器

偏移地址：38h　复位值：00000000h

FSM 单元 1 和 0 的 LUT 函数

表 10-41 CLB_FSM_LUT_FN1_0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| FN1 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FN1 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN0 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FN0 | | | | | | | |
| R/W-0h | | | | | | | |

表 10-42 CLB_FSM_LUT_FN1_0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | FN1 | R/W | 0h | 单元 1 的FSM 模块 LUT 输出函数。<br>复位类型：SYSRSn。 |
| 15-0 | FN0 | R/W | 0h | 单元 0 的FSM 模块 LUT 输出函数。<br>复位类型：SYSRSn。 |

### 10.4.2.14 CLB_FSM_LUT_FN2 寄存器

偏移地址：3Ch 复位值：00000000h

FSM 单元 2 的 LUT 函数

表 10-43 CLB_FSM_LUT_FN2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FN1 | | | | | | | |
| R/W-0h | | | | | | | |

表 10-44 CLB_FSM_LUT_FN2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R/W1C | 0h | 保留。 |
| 15-0 | FN1 | R/W | 0h | 单元 2 的LUT4 输出函数。<br>复位类型：SYSRSn。 |

## 10.4.2.15 CLB_LUT4_FN1_0 寄存器

偏移地址：40h 复位值：00000000h

单元 1 和 0 的 LUT4 模块 LUT 函数

表 10-45 CLB_LUT4_FN1_0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| FN1 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FN1 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN0 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FN0 | | | | | | | |
| R/W-0h | | | | | | | |

表 10-46 CLB_LUT4_FN1_0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | FN1 | R/W | 0h | 单元 1 的LUT4 输出函数。<br>复位类型：SYSRSn。 |
| 15-0 | FN0 | R/W | 0h | 单元 0 的LUT4 输出函数。<br>复位类型：SYSRSn。 |

## 10.4.2.16 CLB_LUT4_FN2 寄存器

偏移地址：44h 复位值：00000000h

单元 2 的 LUT4 模块 LUT 函数

表 10-47 CLB_LUT4_FN2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FN1 | | | | | | | |
| R/W-0h | | | | | | | |

表 10-48 CLB_LUT4_FN2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R/W1C | 0h | 保留。 |
| 15-0 | FN1 | R/W | 0h | 单元 2 的LUT4 输出函数。<br>复位类型：SYSRSn。 |

## 10.4.2.17 CLB_FSM_NEXT_STATE_0 寄存器

偏移地址：48h　复位值：00000000h

单元 0 的 FSM 下一状态方程

表 10-49 CLB_FSM_NEXT_STATE_0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| S1 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| S1 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| S0 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| S0 | | | | | | | |
| R/W-0h | | | | | | | |

表 10-50 CLB_FSM_NEXT_STATE_0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | S1 | R/W | 0h | 单元 0 的 S1 的 FSM 下一状态函数。<br>复位类型：SYSRSn。 |
| 15-0 | S0 | R/W | 0h | 单元 0 的 S0 的 FSM 下一状态函数。<br>复位类型：SYSRSn。 |

## 10.4.2.18 CLB_FSM_NEXT_STATE_1 寄存器

偏移地址：4Ch　复位值：00000000h

单元 1 的 FSM 下一状态方程

表 10-51 CLB_FSM_NEXT_STATE_1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| S1 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| S1 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| S0 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| S0 | | | | | | | |
| R/W-0h | | | | | | | |

表 10-52 CLB_FSM_NEXT_STATE_1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | S1 | R/W | 0h | 单元 1 的 S1 的 FSM 下一状态函数。<br>复位类型：SYSRSn。 |
| 15-0 | S0 | R/W | 0h | 单元 1 的 S0 的 FSM 下一状态函数。<br>复位类型：SYSRSn。 |

## 10.4.2.19 CLB_FSM_NEXT_STATE_2 寄存器

偏移地址：50h　复位值：00000000h

单元 2 的 FSM 下一状态方程

表 10-53 CLB_FSM_NEXT_STATE_2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| S1 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| S1 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| S0 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| S0 | | | | | | | |
| R/W-0h | | | | | | | |

表 10-54 CLB_FSM_NEXT_STATE_2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | S1 | R/W | 0h | 单元 2 的 S1 的 FSM 下一状态函数。<br>复位类型：SYSRSn。 |
| 15-0 | S0 | R/W | 0h | 单元 2 的 S0 的 FSM 下一状态函数。<br>复位类型：SYSRSn。 |

## 10.4.2.20 CLB_MISC_CONTROL 寄存器

偏移地址：54h　复位值：00000000h

计数器和 FSM 的静态控制

表 10-55 CLB_MISC_CONTROL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | COUNT2_LFSR_EN | COUNT1_LFSR_EN | COUNT0_LFSR_EN |
| R-0-0h | | | | | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| COUNT2_MATCH2_TAP_EN | COUNT1_MATCH2_TAP_EN | COUNT0_MATCH2_TAP_EN | COUNT2_MATCH1_TAP_EN | COUNT1_MATCH1_TAP_EN | COUNT0_MATCH1_TAP_EN | FSM_EXTRA_SEL1_2 | FSM_EXTRA_SEL0_2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FSM_EXTRA_SEL1_1 | FSM_EXTRA_SEL0_1 | FSM_EXTRA_SEL1_0 | FSM_EXTRA_SEL0_0 | COUNT_SERIALIZER_2 | COUNT_SERIALIZER_1 | COUNT_SERIALIZER_0 | COUNT_EVENT_CTRL_2 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNT_DIR_2 | COUNT_ADD_SHIFT_2 | COUNT_EVENT_CTRL_1 | COUNT_DIR_1 | COUNT_ADD_SHIFT_1 | COUNT_EVENT_CTRL_0 | COUNT_DIR_0 | COUNT_ADD_SHIFT_0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 10-56 CLB_MISC_CONTROL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R-0 | 0h | 保留。 |
| 26 | COUNT2_LFSR_EN | R/W | 0h | 定义计数器 2 是否工作在 LFSR 模式。只有在 SERIALIZER 模式下才应置 1。<br>0：选择普通串行化操作。<br>1：选择 LFSR 工作模式。<br>复位类型：SYSRSn。 |
| 25 | COUNT1_LFSR_EN | R/W | 0h | 定义计数器 1 是否工作在 LFSR 模式。只有在 SERIALIZER 模式下才应置 1。<br>0：选择普通串行化操作。<br>1：选择 LFSR 工作模式。<br>复位类型：SYSRSn。 |
| 24 | COUNT0_LFSR_EN | R/W | 0h | 定义计数器 0 是否工作在 LFSR 模式。只有在 SERIALIZER 模式下才应置 1。<br>0：选择普通串行化操作。<br>1：选择 LFSR 工作模式。<br>复位类型：SYSRSn。 |
| 23 | COUNT2_MATCH2_TAP_EN | R/W | 0h | 定义 Match2 输出来自匹配单元，还是来自计数器某个位的抽头。<br>0：选择 Match2 比较输出。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：选择 Match2_Tap_val 定义的位。<br>复位类型：SYSRSn。 |
| 22 | COUNT1_MATCH2_TAP_EN | R/W | 0h | 定义 Match2 输出来自匹配单元，还是来自计数器某个位的抽头。<br>0：选择 Match2 比较输出。<br>1：选择 Match2_Tap_val 定义的位。<br>复位类型：SYSRSn。 |
| 21 | COUNT0_MATCH2_TAP_EN | R/W | 0h | 定义 Match2 输出来自匹配单元，还是来自计数器某个位的抽头。<br>0：选择 Match2 比较输出。<br>1：选择 Match2_Tap_val 定义的位。<br>复位类型：SYSRSn。 |
| 20 | COUNT2_MATCH1_TAP_EN | R/W | 0h | 定义 Match1 输出来自匹配单元，还是来自计数器某个位的抽头。<br>0：选择 Match1 比较输出。<br>1：选择 Match1_Tap_val 定义的位。<br>复位类型：SYSRSn。 |
| 19 | COUNT1_MATCH1_TAP_EN | R/W | 0h | 定义 Match1 输出来自匹配单元，还是来自计数器某个位的抽头。<br>0：选择 Match1 比较输出。<br>1：选择 Match1_Tap_val 定义的位。<br>复位类型：SYSRSn。 |
| 18 | COUNT0_MATCH1_TAP_EN | R/W | 0h | 定义 Match1 输出来自匹配单元，还是来自计数器某个位的抽头。<br>0：选择 Match1 比较输出。<br>1：选择 Match1_Tap_val 定义的位。<br>复位类型：SYSRSn。 |
| 17 | FSM_EXTRA_SEL1_2 | R/W | 0h | 定义单元 2 的 FSM LUT 输入选择。<br>0：选择状态 S1。<br>1：选择 EXTRA_EXT_IN1。<br>复位类型：SYSRSn。 |
| 16 | FSM_EXTRA_SEL0_2 | R/W | 0h | 定义单元 2 的 FSM LUT 输入选择。<br>0：选择状态 S0。<br>1：选择 EXTRA_EXT_IN0。<br>复位类型：SYSRSn。 |
| 15 | FSM_EXTRA_SEL1_1 | R/W | 0h | 定义单元 1 的 FSM LUT 输入选择。<br>0：选择状态 S1。<br>1：选择 EXTRA_EXT_IN1。<br>复位类型：SYSRSn。 |
| 14 | FSM_EXTRA_SEL0_1 | R/W | 0h | 定义单元 1 的 FSM LUT 输入选择。<br>0：选择状态 S0。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：选择 EXTRA_EXT_IN0。<br>复位类型：SYSRSn。 |
| 13 | FSM_EXTRA_SEL1_0 | R/W | 0h | 定义单元 0 的 FSM LUT 输入选择。<br>0：选择状态 S1。<br>1：选择 EXTRA_EXT_IN1。<br>复位类型：SYSRSn。 |
| 12 | FSM_EXTRA_SEL0_0 | R/W | 0h | 定义单元 0 的 FSM LUT 输入选择。<br>0：选择状态 S0。<br>1：选择 EXTRA_EXT_IN0。<br>复位类型：SYSRSn。 |
| 11 | COUNT_SERIALIZER_2 | R/W | 0h | 控制单元 2 的计数器是否处于串行化模式。<br>0：普通模式。<br>1：串行化模式。<br>复位类型：SYSRSn。 |
| 10 | COUNT_SERIALIZER_1 | R/W | 0h | 控制单元 1 的计数器是否处于串行化模式。<br>0：普通模式。<br>1：串行化模式。<br>复位类型：SYSRSn。 |
| 9 | COUNT_SERIALIZER_0 | R/W | 0h | 控制单元 0 的计数器是否处于串行化模式。<br>0：普通模式。<br>1：串行化模式。<br>复位类型：SYSRSn。 |
| 8 | COUNT_EVENT_CTRL_2 | R/W | 0h | 控制单元 2 发生 EVENT 时的动作。必须为 0，计数器的间接加载和 HLC 加载才能生效。<br>0：不进行加法或移位，加载预定义值。<br>1：根据其他位，使用预定义值执行加法 / 移位。<br>复位类型：SYSRSn。 |
| 7 | COUNT_DIR_2 | R/W | 0h | 控制单元 2 的加法 / 移位方向。<br>0：右移或减法。<br>1：左移或加法。<br>复位类型：SYSRSn。 |
| 6 | COUNT_ADD_SHIFT_2 | R/W | 0h | 控制单元 2 的计数器在 EVENT 发生时执行加法还是移位。<br>0：移位。<br>1：加法。<br>复位类型：SYSRSn。 |
| 5 | COUNT_EVENT_CTRL_1 | R/W | 0h | 控制单元 1 发生 EVENT 时的动作。必须为 0，计数器的间接加载和 HLC 加载才能生效。<br>0：不进行加法或移位，加载预定义值。<br>1：根据其他位，使用预定义值执行加法 / 移 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 位。<br>复位类型：SYSRSn。 |
| 4 | COUNT_DIR_1 | R/W | 0h | 控制单元 1 的加法 / 移位方向。<br>0：右移或减法。<br>1：左移或加法。<br>复位类型：SYSRSn。 |
| 3 | COUNT_ADD_SHIFT_1 | R/W | 0h | 控制单元 1 的计数器在 EVENT 发生时执行加法还是移位。<br>0：移位。<br>1：加法。<br>复位类型：SYSRSn。 |
| 2 | COUNT_EVENT_CTRL_0 | R/W | 0h | 控制单元 0 发生 EVENT 时的动作。必须为 0，计数器的间接加载和 HLC 加载才能生效。<br>0：不进行加法或移位，加载预定义值。<br>1：根据其他位，使用预定义值执行加法 / 移位。<br>复位类型：SYSRSn。 |
| 1 | COUNT_DIR_0 | R/W | 0h | 控制单元 0 的加法 / 移位方向。<br>0：右移或减法。<br>1：左移或加法。<br>复位类型：SYSRSn。 |
| 0 | COUNT_ADD_SHIFT_0 | R/W | 0h | 控制单元 0 的计数器在 EVENT 发生时执行加法还是移位。<br>0：移位。<br>1：加法。<br>复位类型：SYSRSn。 |

### 10.4.2.21 CLB_OUTPUT_LUT_0 寄存器

偏移地址：58h　复位值：00000000h

Out0 的输入选择和 LUT 函数

表 10-57 CLB_OUTPUT_LUT_0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | FN | | | | | | |
| R/W1C-0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN | IN2 | | | | | IN1 | |
| R/W-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IN1 | | | IN0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-58 CLB_OUTPUT_LUT_0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-23 | RESERVED | R/W1C | 0h | 保留。 |
| 22-15 | FN | R/W | 0h | 输出 LUT 的输出函数。<br>复位类型：SYSRSn。 |
| 14-10 | IN2 | R/W | 0h | 输出 LUT 的 IN2 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | IN1 | R/W | 0h | 输出 LUT 的 IN1 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | IN0 | R/W | 0h | 输出 LUT 的 IN0 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

## 10.4.2.22 CLB_OUTPUT_LUT_1 寄存器

偏移地址：5Ch　复位值：00000000h

Out1 的输入选择和 LUT 函数

表 10-59 CLB_OUTPUT_LUT_1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | FN | | | | | | |
| R/W1C-0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN | IN2 | | | | | IN1 | |
| R/W-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IN1 | | | IN0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-60 CLB_OUTPUT_LUT_1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-23 | RESERVED | R/W1C | 0h | 保留。 |
| 22-15 | FN | R/W | 0h | 输出 LUT 的输出函数。<br>复位类型：SYSRSn。 |
| 14-10 | IN2 | R/W | 0h | 输出 LUT 的 IN2 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | IN1 | R/W | 0h | 输出 LUT 的 IN1 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | IN0 | R/W | 0h | 输出 LUT 的 IN0 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

## 10.4.2.23 CLB_OUTPUT_LUT_2 寄存器

偏移地址：60h　复位值：00000000h

Out2 的输入选择和 LUT 函数

表 10-61 CLB_OUTPUT_LUT_2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | FN | | | | | | |
| R/W1C-0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN | IN2 | | | | | IN1 | |
| R/W-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IN1 | | | IN0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-62 CLB_OUTPUT_LUT_2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-23 | RESERVED | R/W1C | 0h | 保留。 |
| 22-15 | FN | R/W | 0h | 输出 LUT 的输出函数。<br>复位类型：SYSRSn。 |
| 14-10 | IN2 | R/W | 0h | 输出 LUT 的 IN2 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | IN1 | R/W | 0h | 输出 LUT 的 IN1 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | IN0 | R/W | 0h | 输出 LUT 的 IN0 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

## 10.4.2.24 CLB_OUTPUT_LUT_3 寄存器

偏移地址：64h　复位值：00000000h

Out3 的输入选择和 LUT 函数

表 10-63 CLB_OUTPUT_LUT_3 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | FN | | | | | | |
| R/W1C-0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN | IN2 | | | | | IN1 | |
| R/W-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IN1 | | | IN0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-64 CLB_OUTPUT_LUT_3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-23 | RESERVED | R/W1C | 0h | 保留。 |
| 22-15 | FN | R/W | 0h | 输出 LUT 的输出函数。<br>复位类型：SYSRSn。 |
| 14-10 | IN2 | R/W | 0h | 输出 LUT 的 IN2 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | IN1 | R/W | 0h | 输出 LUT 的 IN1 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | IN0 | R/W | 0h | 输出 LUT 的 IN0 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

## 10.4.2.25 CLB_OUTPUT_LUT_4 寄存器

偏移地址：68h　复位值：00000000h

Out4 的输入选择和 LUT 函数

表 10-65 CLB_OUTPUT_LUT_4 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | FN | | | | | | |
| R/W1C-0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN | IN2 | | | | | IN1 | |
| R/W-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IN1 | | | IN0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-66 CLB_OUTPUT_LUT_4 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-23 | RESERVED | R/W1C | 0h | 保留。 |
| 22-15 | FN | R/W | 0h | 输出 LUT 的输出函数。<br>复位类型：SYSRSn。 |
| 14-10 | IN2 | R/W | 0h | 输出 LUT 的 IN2 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | IN1 | R/W | 0h | 输出 LUT 的 IN1 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | IN0 | R/W | 0h | 输出 LUT 的 IN0 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

## 10.4.2.26 CLB_OUTPUT_LUT_5 寄存器

偏移地址：6Ch　复位值：00000000h

Out5 的输入选择和 LUT 函数

表 10-67 CLB_OUTPUT_LUT_5 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | FN | | | | | | |
| R/W1C-0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN | IN2 | | | | | IN1 | |
| R/W-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IN1 | | | IN0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-68 CLB_OUTPUT_LUT_5 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-23 | RESERVED | R/W1C | 0h | 保留。 |
| 22-15 | FN | R/W | 0h | 输出 LUT 的输出函数。<br>复位类型：SYSRSn。 |
| 14-10 | IN2 | R/W | 0h | 输出 LUT 的 IN2 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | IN1 | R/W | 0h | 输出 LUT 的 IN1 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | IN0 | R/W | 0h | 输出 LUT 的 IN0 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

## 10.4.2.27 CLB_OUTPUT_LUT_6 寄存器

偏移地址：70h　复位值：00000000h

Out6 的输入选择和 LUT 函数

表 10-69 CLB_OUTPUT_LUT_6 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | FN | | | | | | |
| R/W1C-0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN | IN2 | | | | | IN1 | |
| R/W-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IN1 | | | IN0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-70 CLB_OUTPUT_LUT_6 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-23 | RESERVED | R/W1C | 0h | 保留。 |
| 22-15 | FN | R/W | 0h | 输出 LUT 的输出函数。<br>复位类型：SYSRSn。 |
| 14-10 | IN2 | R/W | 0h | 输出 LUT 的 IN2 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | IN1 | R/W | 0h | 输出 LUT 的 IN1 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | IN0 | R/W | 0h | 输出 LUT 的 IN0 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

## 10.4.2.28 CLB_OUTPUT_LUT_7 寄存器

偏移地址：74h　复位值：00000000h

Out7 的输入选择和 LUT 函数

表 10-71 CLB_OUTPUT_LUT_7 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | FN | | | | | | |
| R/W1C-0h | R/W-0h | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FN | IN2 | | | | | IN1 | |
| R/W-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IN1 | | | IN0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-72 CLB_OUTPUT_LUT_7 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-23 | RESERVED | R/W1C | 0h | 保留。 |
| 22-15 | FN | R/W | 0h | 输出 LUT 的输出函数。<br>复位类型：SYSRSn。 |
| 14-10 | IN2 | R/W | 0h | 输出 LUT 的 IN2 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | IN1 | R/W | 0h | 输出 LUT 的 IN1 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | IN0 | R/W | 0h | 输出 LUT 的 IN0 输入选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

## 10.4.2.29 CLB_HLC_EVENT_SEL 寄存器

偏移地址：78h　复位值：00000000h

高级控制器事件选择寄存器

表 10-73 CLB_HLC_EVENT_SEL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ALT_EVENT3_SEL | ALT_EVENT2_SEL | ALT_EVENT1_SEL | ALT_EVENT0_SEL | EVENT3_SEL | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| EVENT3_SEL | EVENT2_SEL | | | | | EVENT1_SEL | |
| R/W-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EVENT1_SEL | | | EVENT0_SEL | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-74 CLB_HLC_EVENT_SEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-24 | RESERVED | R-0 | 0h | 保留。 |
| 23 | ALT_EVENT3_SEL | R/W | 0h | 定义 EVENT3 的备用输入选择。<br>复位类型：SYSRSn。 |
| 22 | ALT_EVENT2_SEL | R/W | 0h | 定义 EVENT2 的备用输入选择。<br>复位类型：SYSRSn。 |
| 21 | ALT_EVENT1_SEL | R/W | 0h | 定义 EVENT1 的备用输入选择。<br>复位类型：SYSRSn。 |
| 20 | ALT_EVENT0_SEL | R/W | 0h | 定义 EVENT0 的备用输入选择。<br>复位类型：SYSRSn。 |
| 19-15 | EVENT3_SEL | R/W | 0h | 高级控制器 EVENT3 的 5 位选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 14-10 | EVENT2_SEL | R/W | 0h | 高级控制器 EVENT2 的 5 位选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 9-5 | EVENT1_SEL | R/W | 0h | 高级控制器 EVENT1 的 5 位选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |
| 4-0 | EVENT0_SEL | R/W | 0h | 高级控制器 EVENT0 的 5 位选择值。请参阅静态开关模块输出多路复用表。<br>复位类型：SYSRSn。 |

### 10.4.2.30 CLB_COUNT_MATCH_TAP_SEL 寄存器

偏移地址：7Ch　复位值：00000000h

match1 和 match2 输出的计数器抽头值

表 10-75 CLB_COUNT_MATCH_TAP_SEL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | COUNT2_MATCH2 | | | | | COUNT1_MATCH2 | |
| R-0-0h | R/W-0h | | | | | R/W-0h | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| COUNT1_MATCH2 | | | COUNT0_MATCH2 | | | | |
| R/W-0h | | | R/W-0h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | COUNT2_MATCH1 | | | | | COUNT1_MATCH1 | |
| R-0-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNT1_MATCH1 | | | COUNT0_MATCH1 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-76 CLB_COUNT_MATCH_TAP_SEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | RESERVED | R-0 | 0h | 保留。 |
| 30-26 | COUNT2_MATCH2 | R/W | 0h | 计数器单元 2 的 Match2 抽头的 5 位多路复用选择值。<br>复位类型：SYSRSn。 |
| 25-21 | COUNT1_MATCH2 | R/W | 0h | 计数器单元 1 的 Match2 抽头的 5 位多路复用选择值。<br>复位类型：SYSRSn。 |
| 20-16 | COUNT0_MATCH2 | R/W | 0h | 计数器单元 0 的 Match2 抽头的 5 位多路复用选择值。<br>复位类型：SYSRSn。 |
| 15 | RESERVED | R-0 | 0h | 保留。 |
| 14-10 | COUNT2_MATCH1 | R/W | 0h | 计数器单元 2 的 Match1 抽头的 5 位多路复用选择值。<br>复位类型：SYSRSn。 |
| 9-5 | COUNT1_MATCH1 | R/W | 0h | 计数器单元 1 的 Match1 抽头的 5 位多路复用选择值。<br>复位类型：SYSRSn。 |
| 4-0 | COUNT0_MATCH1 | R/W | 0h | 计数器单元 0 的 Match1 抽头的 5 位多路复用选择值。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 复位类型：SYSRSn。 |

## 10.4.2.31 CLB_OUTPUT_COND_CTRL_0 寄存器

偏移地址：80h　复位值：00000000h

输出 0 的输出调理控制

表 10-77 CLB_OUTPUT_COND_CTRL_0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | ASYNC_COND_EN | SEL_RAW_IN | HW_RLS_CTRL_SEL | HW_GATING_CTRL_SEL | SEL_RELEASE_CTRL | | |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_GATING_CTRL | | | LEVEL_3_SEL | | LEVEL_2_SEL | | LEVEL_1_SEL |
| R/W1C-0h | | | R/W1C-0h | | R/W1C-0h | | R/W1C-0h |

表 10-78 CLB_OUTPUT_COND_CTRL_0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14 | ASYNC_COND_EN | R/W1C | 0h | 控制输出通过异步调理模块还是旁路该模块。<br>0：旁路异步调理模块。<br>1：使能异步调理路径。<br>复位类型：SYSRSn。 |
| 13 | SEL_RAW_IN | R/W1C | 0h | 控制送入输出调理模块逻辑的是 CELL 输出还是 CELL 输入。<br>0：使用 CELL 输出，内部延迟 1 个周期。<br>1：使用 CELL 原始输入。<br>复位类型：SYSRSn。 |
| 12 | HW_RLS_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为释放控制。<br>0：软件寄存器值用作释放控制。<br>1：所选 CELL 输出用作释放控制。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 11 | HW_GATING_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为门控控制。<br>0：软件寄存器值用作门控控制。<br>1：所选 CELL 输出用作门控控制。<br>复位类型：SYSRSn。 |
| 10-8 | SEL_RELEASE_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为释放控制。<br>复位类型：SYSRSn。 |
| 7-5 | SEL_GATING_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为门控控制。<br>复位类型：SYSRSn。 |
| 4-3 | LEVEL_3_SEL | R/W1C | 0h | 控制第三级多路复用选择。<br>00：输入信号原样送至输出。<br>01：输入信号上升沿使输出异步清零。<br>10：输入信号上升沿使输出异步置位。<br>11：输入信号延迟 1 个时钟周期后送至输出。<br>复位类型：SYSRSn。 |
| 2-1 | LEVEL_2_SEL | R/W1C | 0h | 控制第二级多路复用选择。<br>00：输入信号送至下一级。<br>01：输入信号与 Gating_control 进行 AND 运算，结果送至下一级。<br>10：输入信号与 Gating_control 进行 OR 运算，结果送至下一级。<br>11：输入信号与 Gating_control 进行 XOR 运算，结果送至下一级。<br>复位类型：SYSRSn。 |
| 0 | LEVEL_1_SEL | R/W1C | 0h | 第一级多路复用选择值。<br>0：信号直接送至下一级。<br>1：信号反相后送至下一级。<br>复位类型：SYSRSn。 |

## 10.4.2.32 CLB_OUTPUT_COND_CTRL_1 寄存器

偏移地址：84h　复位值：00000000h

输出 1 的输出调理控制

表 10-79 CLB_OUTPUT_COND_CTRL_1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | ASYNC_COND_EN | SEL_RAW_IN | HW_RLS_CTRL_SEL | HW_GATING_CTRL_SEL | SEL_RELEASE_CTRL | | |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_GATING_CTRL | | | LEVEL_3_SEL | | LEVEL_2_SEL | | LEVEL_1_SEL |
| R/W1C-0h | | | R/W1C-0h | | R/W1C-0h | | R/W1C-0h |

表 10-80 CLB_OUTPUT_COND_CTRL_1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14 | ASYNC_COND_EN | R/W1C | 0h | 控制输出通过异步调理模块还是旁路该模块。<br>0：旁路异步调理模块。<br>1：使能异步调理路径。<br>复位类型：SYSRSn。 |
| 13 | SEL_RAW_IN | R/W1C | 0h | 控制送入输出调理模块逻辑的是 CELL 输出还是 CELL 输入。<br>0：使用 CELL 输出，内部延迟 1 个周期。<br>1：使用 CELL 原始输入。<br>复位类型：SYSRSn。 |
| 12 | HW_RLS_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为释放控制。<br>0：软件寄存器值用作释放控制。<br>1：所选 CELL 输出用作释放控制。<br>复位类型：SYSRSn。 |
| 11 | HW_GATING_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为门控控制。<br>0：软件寄存器值用作门控控制。<br>1：所选 CELL 输出用作门控控制。<br>复位类型：SYSRSn。 |
| 10-8 | SEL_RELEASE_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为释放控制。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 7-5 | SEL_GATING_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为门控控制。<br>复位类型：SYSRSn。 |
| 4-3 | LEVEL_3_SEL | R/W1C | 0h | 控制第三级多路复用选择。<br>00：输入信号原样送至输出。<br>01：输入信号上升沿使输出异步清零。<br>10：输入信号上升沿使输出异步置位。<br>11：输入信号延迟 1 个时钟周期后送至输出。<br>复位类型：SYSRSn。 |
| 2-1 | LEVEL_2_SEL | R/W1C | 0h | 控制第二级多路复用选择。<br>00：输入信号送至下一级。<br>01：输入信号与 Gating_control 进行 AND 运算，结果送至下一级。<br>10：输入信号与 Gating_control 进行 OR 运算，结果送至下一级。<br>11：输入信号与 Gating_control 进行 XOR 运算，结果送至下一级。<br>复位类型：SYSRSn。 |
| 0 | LEVEL_1_SEL | R/W1C | 0h | 第一级多路复用选择值。<br>0：信号直接送至下一级。<br>1：信号反相后送至下一级。<br>复位类型：SYSRSn。 |

### 10.4.2.33 CLB_OUTPUT_COND_CTRL_2 寄存器

偏移地址：88h　复位值：00000000h

输出 2 的输出调理控制

表 10-81 CLB_OUTPUT_COND_CTRL_2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | ASYNC_COND_EN | SEL_RAW_IN | HW_RLS_CTRL_SEL | HW_GATING_CTRL_SEL | SEL_RELEASE_CTRL | | |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_GATING_CTRL | | | LEVEL_3_SEL | | LEVEL_2_SEL | | LEVEL_1_SEL |
| R/W1C-0h | | | R/W1C-0h | | R/W1C-0h | | R/W1C-0h |

表 10-82 CLB_OUTPUT_COND_CTRL_2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14 | ASYNC_COND_EN | R/W1C | 0h | 控制输出通过异步调理模块还是旁路该模块。<br>0：旁路异步调理模块。<br>1：使能异步调理路径。<br>复位类型：SYSRSn。 |
| 13 | SEL_RAW_IN | R/W1C | 0h | 控制送入输出调理模块逻辑的是 CELL 输出还是 CELL 输入。<br>0：使用 CELL 输出，内部延迟 1 个周期。<br>1：使用 CELL 原始输入。<br>复位类型：SYSRSn。 |
| 12 | HW_RLS_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为释放控制。<br>0：软件寄存器值用作释放控制。<br>1：所选 CELL 输出用作释放控制。<br>复位类型：SYSRSn。 |
| 11 | HW_GATING_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为门控控制。<br>0：软件寄存器值用作门控控制。<br>1：所选 CELL 输出用作门控控制。<br>复位类型：SYSRSn。 |
| 10-8 | SEL_RELEASE_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为释放控制。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 7-5 | SEL_GATING_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为门控控制。<br>复位类型：SYSRSn。 |
| 4-3 | LEVEL_3_SEL | R/W1C | 0h | 控制第三级多路复用选择。<br>00：输入信号原样送至输出。<br>01：输入信号上升沿使输出异步清零。<br>10：输入信号上升沿使输出异步置位。<br>11：输入信号延迟 1 个时钟周期后送至输出。<br>复位类型：SYSRSn。 |
| 2-1 | LEVEL_2_SEL | R/W1C | 0h | 控制第二级多路复用选择。<br>00：输入信号送至下一级。<br>01：输入信号与 Gating_control 进行 AND 运算，结果送至下一级。<br>10：输入信号与 Gating_control 进行 OR 运算，结果送至下一级。<br>11：输入信号与 Gating_control 进行 XOR 运算，结果送至下一级。<br>复位类型：SYSRSn。 |
| 0 | LEVEL_1_SEL | R/W1C | 0h | 第一级多路复用选择值。<br>0：信号直接送至下一级。<br>1：信号反相后送至下一级。<br>复位类型：SYSRSn。 |

### 10.4.2.34 CLB_OUTPUT_COND_CTRL_3 寄存器

偏移地址：8Ch　复位值：00000000h

输出 3 的输出调理控制

表 10-83 CLB_OUTPUT_COND_CTRL_3 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | ASYNC_COND_EN | SEL_RAW_IN | HW_RLS_CTRL_SEL | HW_GATING_CTRL_SEL | SEL_RELEASE_CTRL | | |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_GATING_CTRL | | | LEVEL_3_SEL | | LEVEL_2_SEL | | LEVEL_1_SEL |
| R/W1C-0h | | | R/W1C-0h | | R/W1C-0h | | R/W1C-0h |

表 10-84 CLB_OUTPUT_COND_CTRL_3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14 | ASYNC_COND_EN | R/W1C | 0h | 控制输出通过异步调理模块还是旁路该模块。<br>0：旁路异步调理模块。<br>1：使能异步调理路径。<br>复位类型：SYSRSn。 |
| 13 | SEL_RAW_IN | R/W1C | 0h | 控制送入输出调理模块逻辑的是 CELL 输出还是 CELL 输入。<br>0：使用 CELL 输出，内部延迟 1 个周期。<br>1：使用 CELL 原始输入。<br>复位类型：SYSRSn。 |
| 12 | HW_RLS_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为释放控制。<br>0：软件寄存器值用作释放控制。<br>1：所选 CELL 输出用作释放控制。<br>复位类型：SYSRSn。 |
| 11 | HW_GATING_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为门控控制。<br>0：软件寄存器值用作门控控制。<br>1：所选 CELL 输出用作门控控制。<br>复位类型：SYSRSn。 |
| 10-8 | SEL_RELEASE_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为释放控制。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 7-5 | SEL_GATING_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为门控控制。<br>复位类型：SYSRSn。 |
| 4-3 | LEVEL_3_SEL | R/W1C | 0h | 控制第三级多路复用选择。<br>00：输入信号原样送至输出。<br>01：输入信号上升沿使输出异步清零。<br>10：输入信号上升沿使输出异步置位。<br>11：输入信号延迟 1 个时钟周期后送至输出。<br>复位类型：SYSRSn。 |
| 2-1 | LEVEL_2_SEL | R/W1C | 0h | 控制第二级多路复用选择。<br>00：输入信号送至下一级。<br>01：输入信号与 Gating_control 进行 AND 运算，结果送至下一级。<br>10：输入信号与 Gating_control 进行 OR 运算，结果送至下一级。<br>11：输入信号与 Gating_control 进行 XOR 运算，结果送至下一级。<br>复位类型：SYSRSn。 |
| 0 | LEVEL_1_SEL | R/W1C | 0h | 第一级多路复用选择值。<br>0：信号直接送至下一级。<br>1：信号反相后送至下一级。<br>复位类型：SYSRSn。 |

### 10.4.2.35 CLB_OUTPUT_COND_CTRL_4 寄存器

偏移地址：90h　复位值：00000000h

输出 4 的输出调理控制

表 10-85 CLB_OUTPUT_COND_CTRL_4 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | ASYNC_COND_EN | SEL_RAW_IN | HW_RLS_CTRL_SEL | HW_GATING_CTRL_SEL | SEL_RELEASE_CTRL | | |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_GATING_CTRL | | | LEVEL_3_SEL | | LEVEL_2_SEL | | LEVEL_1_SEL |
| R/W1C-0h | | | R/W1C-0h | | R/W1C-0h | | R/W1C-0h |

表 10-86 CLB_OUTPUT_COND_CTRL_4 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14 | ASYNC_COND_EN | R/W1C | 0h | 控制输出通过异步调理模块还是旁路该模块。<br>0：旁路异步调理模块。<br>1：使能异步调理路径。<br>复位类型：SYSRSn。 |
| 13 | SEL_RAW_IN | R/W1C | 0h | 控制送入输出调理模块逻辑的是 CELL 输出还是 CELL 输入。<br>0：使用 CELL 输出，内部延迟 1 个周期。<br>1：使用 CELL 原始输入。<br>复位类型：SYSRSn。 |
| 12 | HW_RLS_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为释放控制。<br>0：软件寄存器值用作释放控制。<br>1：所选 CELL 输出用作释放控制。<br>复位类型：SYSRSn。 |
| 11 | HW_GATING_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为门控控制。<br>0：软件寄存器值用作门控控制。<br>1：所选 CELL 输出用作门控控制。<br>复位类型：SYSRSn。 |
| 10-8 | SEL_RELEASE_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为释放控制。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 7-5 | SEL_GATING_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为门控控制。<br>复位类型：SYSRSn。 |
| 4-3 | LEVEL_3_SEL | R/W1C | 0h | 控制第三级多路复用选择。<br>00：输入信号原样送至输出。<br>01：输入信号上升沿使输出异步清零。<br>10：输入信号上升沿使输出异步置位。<br>11：输入信号延迟 1 个时钟周期后送至输出。<br>复位类型：SYSRSn。 |
| 2-1 | LEVEL_2_SEL | R/W1C | 0h | 控制第二级多路复用选择。<br>00：输入信号送至下一级。<br>01：输入信号与 Gating_control 进行 AND 运算，结果送至下一级。<br>10：输入信号与 Gating_control 进行 OR 运算，结果送至下一级。<br>11：输入信号与 Gating_control 进行 XOR 运算，结果送至下一级。<br>复位类型：SYSRSn。 |
| 0 | LEVEL_1_SEL | R/W1C | 0h | 第一级多路复用选择值。<br>0：信号直接送至下一级。<br>1：信号反相后送至下一级。<br>复位类型：SYSRSn。 |

### 10.4.2.36 CLB_OUTPUT_COND_CTRL_5 寄存器

偏移地址：94h　复位值：00000000h

输出 5 的输出调理控制

表 10-87 CLB_OUTPUT_COND_CTRL_5 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | ASYNC_COND_EN | SEL_RAW_IN | HW_RLS_CTRL_SEL | HW_GATING_CTRL_SEL | SEL_RELEASE_CTRL | | |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_GATING_CTRL | | | LEVEL_3_SEL | | LEVEL_2_SEL | | LEVEL_1_SEL |
| R/W1C-0h | | | R/W1C-0h | | R/W1C-0h | | R/W1C-0h |

表 10-88 CLB_OUTPUT_COND_CTRL_5 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14 | ASYNC_COND_EN | R/W1C | 0h | 控制输出通过异步调理模块还是旁路该模块。<br>0：旁路异步调理模块。<br>1：使能异步调理路径。<br>复位类型：SYSRSn。 |
| 13 | SEL_RAW_IN | R/W1C | 0h | 控制送入输出调理模块逻辑的是 CELL 输出还是 CELL 输入。<br>0：使用 CELL 输出，内部延迟 1 个周期。<br>1：使用 CELL 原始输入。<br>复位类型：SYSRSn。 |
| 12 | HW_RLS_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为释放控制。<br>0：软件寄存器值用作释放控制。<br>1：所选 CELL 输出用作释放控制。<br>复位类型：SYSRSn。 |
| 11 | HW_GATING_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为门控控制。<br>0：软件寄存器值用作门控控制。<br>1：所选 CELL 输出用作门控控制。<br>复位类型：SYSRSn。 |
| 10-8 | SEL_RELEASE_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为释放控制。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 7-5 | SEL_GATING_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为门控控制。<br>复位类型：SYSRSn。 |
| 4-3 | LEVEL_3_SEL | R/W1C | 0h | 控制第三级多路复用选择。<br>00：输入信号原样送至输出。<br>01：输入信号上升沿使输出异步清零。<br>10：输入信号上升沿使输出异步置位。<br>11：输入信号延迟 1 个时钟周期后送至输出。<br>复位类型：SYSRSn。 |
| 2-1 | LEVEL_2_SEL | R/W1C | 0h | 控制第二级多路复用选择。<br>00：输入信号送至下一级。<br>01：输入信号与 Gating_control 进行 AND 运算，结果送至下一级。<br>10：输入信号与 Gating_control 进行 OR 运算，结果送至下一级。<br>11：输入信号与 Gating_control 进行 XOR 运算，结果送至下一级。<br>复位类型：SYSRSn。 |
| 0 | LEVEL_1_SEL | R/W1C | 0h | 第一级多路复用选择值。<br>0：信号直接送至下一级。<br>1：信号反相后送至下一级。<br>复位类型：SYSRSn。 |

### 10.4.2.37 CLB_OUTPUT_COND_CTRL_6 寄存器

偏移地址：98h　复位值：00000000h

输出 6 的输出调理控制

表 10-89 CLB_OUTPUT_COND_CTRL_6 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | ASYNC_COND_EN | SEL_RAW_IN | HW_RLS_CTRL_SEL | HW_GATING_CTRL_SEL | SEL_RELEASE_CTRL | | |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_GATING_CTRL | | | LEVEL_3_SEL | | LEVEL_2_SEL | | LEVEL_1_SEL |
| R/W1C-0h | | | R/W1C-0h | | R/W1C-0h | | R/W1C-0h |

表 10-90 CLB_OUTPUT_COND_CTRL_6 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14 | ASYNC_COND_EN | R/W1C | 0h | 控制输出通过异步调理模块还是旁路该模块。<br>0：旁路异步调理模块。<br>1：使能异步调理路径。<br>复位类型：SYSRSn。 |
| 13 | SEL_RAW_IN | R/W1C | 0h | 控制送入输出调理模块逻辑的是 CELL 输出还是 CELL 输入。<br>0：使用 CELL 输出，内部延迟 1 个周期。<br>1：使用 CELL 原始输入。<br>复位类型：SYSRSn。 |
| 12 | HW_RLS_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为释放控制。<br>0：软件寄存器值用作释放控制。<br>1：所选 CELL 输出用作释放控制。<br>复位类型：SYSRSn。 |
| 11 | HW_GATING_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为门控控制。<br>0：软件寄存器值用作门控控制。<br>1：所选 CELL 输出用作门控控制。<br>复位类型：SYSRSn。 |
| 10-8 | SEL_RELEASE_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为释放控制。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 7-5 | SEL_GATING_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为门控控制。<br>复位类型：SYSRSn。 |
| 4-3 | LEVEL_3_SEL | R/W1C | 0h | 控制第三级多路复用选择。<br>00：输入信号原样送至输出。<br>01：输入信号上升沿使输出异步清零。<br>10：输入信号上升沿使输出异步置位。<br>11：输入信号延迟 1 个时钟周期后送至输出。<br>复位类型：SYSRSn。 |
| 2-1 | LEVEL_2_SEL | R/W1C | 0h | 控制第二级多路复用选择。<br>00：输入信号送至下一级。<br>01：输入信号与 Gating_control 进行 AND 运算，结果送至下一级。<br>10：输入信号与 Gating_control 进行 OR 运算，结果送至下一级。<br>11：输入信号与 Gating_control 进行 XOR 运算，结果送至下一级。<br>复位类型：SYSRSn。 |
| 0 | LEVEL_1_SEL | R/W1C | 0h | 第一级多路复用选择值。<br>0：信号直接送至下一级。<br>1：信号反相后送至下一级。<br>复位类型：SYSRSn。 |

### 10.4.2.38 CLB_OUTPUT_COND_CTRL_7 寄存器

偏移地址：9Ch　复位值：00000000h

输出 7 的输出调理控制

表 10-91 CLB_OUTPUT_COND_CTRL_7 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | ASYNC_COND_EN | SEL_RAW_IN | HW_RLS_CTRL_SEL | HW_GATING_CTRL_SEL | SEL_RELEASE_CTRL | | |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_GATING_CTRL | | | LEVEL_3_SEL | | LEVEL_2_SEL | | LEVEL_1_SEL |
| R/W1C-0h | | | R/W1C-0h | | R/W1C-0h | | R/W1C-0h |

表 10-92 CLB_OUTPUT_COND_CTRL_7 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-15 | RESERVED | R/W1C | 0h | 保留。 |
| 14 | ASYNC_COND_EN | R/W1C | 0h | 控制输出通过异步调理模块还是旁路该模块。<br>0：旁路异步调理模块。<br>1：使能异步调理路径。<br>复位类型：SYSRSn。 |
| 13 | SEL_RAW_IN | R/W1C | 0h | 控制送入输出调理模块逻辑的是 CELL 输出还是 CELL 输入。<br>0：使用 CELL 输出，内部延迟 1 个周期。<br>1：使用 CELL 原始输入。<br>复位类型：SYSRSn。 |
| 12 | HW_RLS_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为释放控制。<br>0：软件寄存器值用作释放控制。<br>1：所选 CELL 输出用作释放控制。<br>复位类型：SYSRSn。 |
| 11 | HW_GATING_CTRL_SEL | R/W1C | 0h | 控制使用硬件（CELL 输出）还是软件（GP_REG）作为门控控制。<br>0：软件寄存器值用作门控控制。<br>1：所选 CELL 输出用作门控控制。<br>复位类型：SYSRSn。 |
| 10-8 | SEL_RELEASE_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为释放控制。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 7-5 | SEL_GATING_CTRL | R/W1C | 0h | 3 位多路复用选择值，从 8 个 CELL 输出中选择一个作为门控控制。<br>复位类型：SYSRSn。 |
| 4-3 | LEVEL_3_SEL | R/W1C | 0h | 控制第三级多路复用选择。<br>00：输入信号原样送至输出。<br>01：输入信号上升沿使输出异步清零。<br>10：输入信号上升沿使输出异步置位。<br>11：输入信号延迟 1 个时钟周期后送至输出。<br>复位类型：SYSRSn。 |
| 2-1 | LEVEL_2_SEL | R/W1C | 0h | 控制第二级多路复用选择。<br>00：输入信号送至下一级。<br>01：输入信号与 Gating_control 进行 AND 运算，结果送至下一级。<br>10：输入信号与 Gating_control 进行 OR 运算，结果送至下一级。<br>11：输入信号与 Gating_control 进行 XOR 运算，结果送至下一级。<br>复位类型：SYSRSn。 |
| 0 | LEVEL_1_SEL | R/W1C | 0h | 第一级多路复用选择值。<br>0：信号直接送至下一级。<br>1：信号反相后送至下一级。<br>复位类型：SYSRSn。 |

## 10.4.2.39 CLB_MISC_ACCESS_CTRL 寄存器

偏移地址：A0h　复位值：0000h

杂项访问与使能控制

表 10-93 CLB_MISC_ACCESS_CTRL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | BLKEN | SCLICN |
| R/W1C-0h | | | | | | R/W1C-0h | R/W1C-0h |

表 10-94 CLB_MISC_ACCESS_CTRL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-2 | RESERVED | R/W1C | 0h | 保留。 |
| 1 | BLKEN | R/W1C | 0h | 用于阻止写入 CLB_OUT_EN。<br>0：允许写入 CLB_OUT_EN。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：禁止写入 CLB_OUT_EN。<br>复位类型：SYSRSn。 |
| 0 | SCLICN | R/W1C | 0h | 指示 SPI 缓冲器导出 CLB 输出数据功能的状态。<br>0：禁用此功能。<br>1：使能此功能。<br>复位类型：SYSRSn。 |

## 10.4.2.40 CLB_SPI_DATA_CTRL_HI 寄存器

偏移地址：A2h　复位值：0000h

CLB 至 SPI 缓冲器高位控制

表 10-95 CLB_SPI_DATA_CTRL_HI 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | SHIFT | | | | |
| R/W1C-0h | | | R/W1C-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | STRB | | | | |
| R/W1C-0h | | | R/W1C-0h | | | | |

表 10-96 CLB_SPI_DATA_CTRL_HI 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-13 | RESERVED | R/W1C | 0h | 保留。 |
| 12-8 | SHIFT | R/W1C | 0h | 此 5 位值指定寄存器 R0 中作为输出数据最低有效位的起始位位置。<br>00000：输出数据为 R0[15:0]。<br>00001：输出数据为 R0[16:1]。<br>00010：输出数据为 R0[17:2]。<br>00011：输出数据为 R0[18:3]。<br>……<br>10000：输出数据为 R0[31:16]。<br>复位类型：SYSRSn。 |
| 7-5 | RESERVED | R/W1C | 0h | 保留。 |
| 4-0 | STRB | R/W1C | 0h | 此 5 位值选择一个 HLC_EVENT 输入作为 data_valid 选通信号。<br>复位类型：SYSRSn。 |

### 10.4.3 CLB_LOGIC_CONTROL_REGS 寄存器

表中列出了存储器映射寄存器。未列出的偏移地址应视为保留位置，不应修改这些位置的寄存器内容。

表 10-97 CLB_LOGIC_CONTROL_REGS 寄存器

| 偏移地址 | 寄存器标识 | 寄存器名称 |
|---|---|---|
| 0x0 | CLB_LOAD_EN | 全局使能和间接加载使能控制 |
| 0x4 | CLB_LOAD_ADDR | 间接加载地址 |
| 0x8 | CLB_LOAD_DATA | 间接加载数据 |
| 0xC | CLB_INPUT_FILTER | 双沿检测的输入滤波器选择和同步器 |
| 0x10 | CLB_IN_MUX_SEL_0 | TILE输入选择：信号或GPREG |
| 0x14 | CLB_LCL_MUX_SEL_1 | 本地输入的MUX选择 |
| 0x18 | CLB_LCL_MUX_SEL_2 | 本地输入的MUX选择 |
| 0x1C | CLB_BUF_PTR | PUSH和PULL指针 |
| 0x20 | CLB_GP_REG | 为TILE提供输入的通用寄存器 |
| 0x24 | CLB_OUT_EN | TILE输出使能 |
| 0x28 | CLB_GLBL_MUX_SEL_1 | 全局输入的MUX选择 |
| 0x2C | CLB_GLBL_MUX_SEL_2 | 全局输入的MUX选择 |
| 0x30 | CLB_PRESCALE_CTRL | 预分频寄存器控制 |
| 0x40 | CLB_INTR_TAG_REG | 中断标签寄存器 |
| 0x44 | CLB_LOCK | LOCK控制寄存器 |
| 0x5C | CLB_DBG_OUT_2 | CLB输入和最终的异步输出 |
| 0x60 | CLB_DBG_R0 | HLC的R0寄存器 |
| 0x64 | CLB_DBG_R1 | HLC的R1寄存器 |
| 0x68 | CLB_DBG_R2 | HLC的R2寄存器 |
| 0x6C | CLB_DBG_R3 | HLC的R3寄存器 |
| 0x70 | CLB_DBG_C0 | Counter_0寄存器 |
| 0x74 | CLB_DBG_C1 | Counter_1寄存器 |
| 0x78 | CLB_DBG_C2 | Counter_2寄存器 |
| 0x7C | CLB_DBG_OUT | TILE中一些子模块的输出 |

表 10-98 CLB_LOGIC_CONTROL_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| R-1 | R<br>-1 | 读<br>返回 1 |
| 写类型 | | |

| 访问类型 | 编码 | 说明 |
|---|---|---|
| W | W | 写 |
| W1C | W<br>1C | 写<br>1清除 |
| WSonce | W<br>Sonce | 写<br>仅可置位一次 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i、j、k、l、m、n | | 用于寄存器名称、偏移地址或地址时，表示属于重复寄存器组的寄存器数组值。寄存器组按层次组织，数组以公式表示。 |
| y | | 用于寄存器名称、偏移地址或地址时，表示寄存器数组值。 |

### 10.4.3.1 CLB_LOAD_EN 寄存器

偏移地址：0h　复位值：0000h

全局使能与间接加载使能控制，仅全局使能位受 LOCK 保护

表 10-99 CLB_LOAD_EN 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | PIPELINE_EN | NMI_EN | STOP | GLOBAL_EN | LOAD_EN |
| R-0-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 10-100 CLB_LOAD_EN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-5 | RESERVED | R-0 | 0h | 保留。 |
| 4 | PIPELINE_EN | R/W | 0h | 控制 HLC 和计数器模块中所有 CLB 操作的流水线处理。置 1 时使能流水线操作。<br>复位类型：SYSRSn。 |
| 3 | NMI_EN | R/W | 0h | 控制 HLC 每次执行 INTR 操作时，是否在产生中断的同时产生 NMI。默认禁用 NMI；置 1 时使能。<br>复位类型：SYSRSn。 |
| 2 | STOP | R/W | 0h | 定义 CPU 调试暂停（HALT）期间 CELL 中时序元件的行为。置 0 时忽略调试暂停条件。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | GLOBAL_EN | R/W | 0h | CELL 逻辑的全局使能信号，同时作为 CELL 逻辑的软复位。此位从 1 清为 0 时，CLB 输出（包括 LUT 和 OUTLUT 输出）被门控；即 GLOBAL_EN 为低时，CLB 输出为低。此外，FSM 和 AOC 模块也被复位。<br>注意：此位变低时，COUNTER 模块和 HLC 仅暂停，内部不会复位，因此可在 GLOBAL_EN 为 0 时预加载这些子模块。通常完成所有其他配置后再设置此位。此位受 LOCK 保护。<br>复位类型：SYSRSn。 |
| 0 | LOAD_EN | R/W | 0h | 写入时将此位置 1，可为 CELL 中的间接寄存器加载产生一个加载使能脉冲。<br>复位类型：SYSRSn。 |

### 10.4.3.2 CLB_LOAD_ADDR 寄存器

偏移地址：4h　复位值：00000000h

间接地址

表 10-101 CLB_LOAD_ADDR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | ADDR | | | | | |
| R-0-0h | | R/W-0h | | | | | |

表 10-102 CLB_LOAD_ADDR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-6 | RESERVED | R-0 | 0h | 保留。 |
| 5-0 | ADDR | R/W | 0h | 写入 CELL 间接地址空间所使用的地址位。<br>复位类型：SYSRSn。 |

### 10.4.3.3 CLB_LOAD_DATA 寄存器

偏移地址：8h　复位值：00000000h

间接加载数据

表 10-103 CLB_LOAD_DATA 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DATA | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

表 10-104 CLB_LOAD_DATA 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | DATA | R/W | 0h | 保存用于写入 CELL 间接地址空间的 32 位数据。<br>复位类型：SYSRSn。 |

### 10.4.3.4 CLB_INPUT_FILTER 寄存器

偏移地址：Ch　复位值：00000000h

边沿检测与同步器的输入滤波选择

表 10-105 CLB_INPUT_FILTER 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PIPE7 | PIPE6 | PIPE5 | PIPE4 | PIPE3 | PIPE2 | PIPE1 | PIPE0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SYNC7 | SYNC6 | SYNC5 | SYNC4 | SYNC3 | SYNC2 | SYNC1 | SYNC0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FIN7 | | FIN6 | | FIN5 | | FIN4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FIN3 | | FIN2 | | FIN1 | | FIN0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 10-106 CLB_INPUT_FILTER 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | PIPE7 | R/W | 0h | 使能输入 7 的流水线处理。<br>复位类型：SYSRSn。 |
| 30 | PIPE6 | R/W | 0h | 使能输入 6 的流水线处理。<br>复位类型：SYSRSn。 |
| 29 | PIPE5 | R/W | 0h | 使能输入 5 的流水线处理。<br>复位类型：SYSRSn。 |
| 28 | PIPE4 | R/W | 0h | 使能输入 4 的流水线处理。<br>复位类型：SYSRSn。 |
| 27 | PIPE3 | R/W | 0h | 使能输入 3 的流水线处理。<br>复位类型：SYSRSn。 |
| 26 | PIPE2 | R/W | 0h | 使能输入 2 的流水线处理。<br>复位类型：SYSRSn。 |
| 25 | PIPE1 | R/W | 0h | 使能输入 1 的流水线处理。<br>复位类型：SYSRSn。 |
| 24 | PIPE0 | R/W | 0h | 使能输入 0 的流水线处理。<br>复位类型：SYSRSn。 |
| 23 | SYNC7 | R/W | 0h | 输入 7 的同步器选择控制。<br>复位类型：SYSRSn。 |
| 22 | SYNC6 | R/W | 0h | 输入 6 的同步器选择控制。<br>复位类型：SYSRSn。 |
| 21 | SYNC5 | R/W | 0h | 输入 5 的同步器选择控制。<br>复位类型：SYSRSn。 |
| 20 | SYNC4 | R/W | 0h | 输入 4 的同步器选择控制。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 19 | SYNC3 | R/W | 0h | 输入 3 的同步器选择控制。<br>复位类型：SYSRSn。 |
| 18 | SYNC2 | R/W | 0h | 输入 2 的同步器选择控制。<br>复位类型：SYSRSn。 |
| 17 | SYNC1 | R/W | 0h | 输入 1 的同步器选择控制。<br>复位类型：SYSRSn。 |
| 16 | SYNC0 | R/W | 0h | 输入 0 的同步器选择控制。<br>复位类型：SYSRSn。 |
| 15-14 | FIN7 | R/W | 0h | CELL 输入 7 的输入滤波选择。使用 2 位定义边沿滤波。<br>00：不滤波。<br>01：上升沿检测。<br>10：下降沿检测。<br>11：任意边沿检测。<br>复位类型：SYSRSn。 |
| 13-12 | FIN6 | R/W | 0h | CELL 输入 6 的输入滤波选择。使用 2 位定义边沿滤波。<br>00：不滤波。<br>01：上升沿检测。<br>10：下降沿检测。<br>11：任意边沿检测。<br>复位类型：SYSRSn。 |
| 11-10 | FIN5 | R/W | 0h | CELL 输入 5 的输入滤波选择。使用 2 位定义边沿滤波。<br>00：不滤波。<br>01：上升沿检测。<br>10：下降沿检测。<br>11：任意边沿检测。<br>复位类型：SYSRSn。 |
| 9-8 | FIN4 | R/W | 0h | CELL 输入 4 的输入滤波选择。使用 2 位定义边沿滤波。<br>00：不滤波。<br>01：上升沿检测。<br>10：下降沿检测。<br>11：任意边沿检测。<br>复位类型：SYSRSn。 |
| 7-6 | FIN3 | R/W | 0h | CELL 输入 3 的输入滤波选择。使用 2 位定义边沿滤波。<br>00：不滤波。<br>01：上升沿检测。<br>10：下降沿检测。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：任意边沿检测。<br>复位类型：SYSRSn。 |
| 5-4 | FIN2 | R/W | 0h | CELL 输入 2 的输入滤波选择。使用 2 位定义边沿滤波。<br>00：不滤波。<br>01：上升沿检测。<br>10：下降沿检测。<br>11：任意边沿检测。<br>复位类型：SYSRSn。 |
| 3-2 | FIN1 | R/W | 0h | CELL 输入 1 的输入滤波选择。使用 2 位定义边沿滤波。<br>00：不滤波。<br>01：上升沿检测。<br>10：下降沿检测。<br>11：任意边沿检测。<br>复位类型：SYSRSn。 |
| 1-0 | FIN0 | R/W | 0h | CELL 输入 0 的输入滤波选择。使用 2 位定义边沿滤波。<br>00：不滤波。<br>01：上升沿检测。<br>10：下降沿检测。<br>11：任意边沿检测。<br>复位类型：SYSRSn。 |

### 10.4.3.5 CLB_IN_MUX_SEL_0 寄存器

偏移地址：10h　复位值：00000000h

信号与 GP 寄存器之间的输入选择

表 10-107 CLB_IN_MUX_SEL_0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL_GP_IN_7 | SEL_GP_IN_6 | SEL_GP_IN_5 | SEL_GP_IN_4 | SEL_GP_IN_3 | SEL_GP_IN_2 | SEL_GP_IN_1 | SEL_GP_IN_0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 10-108 CLB_IN_MUX_SEL_0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-8 | RESERVED | R-0 | 0h | 保留。 |
| 7 | SEL_GP_IN_7 | R/W | 0h | 输入 7 的选择控制，选择外部输入或 CLB_GP_REG[7]。<br>0：来自选定的外部输入。<br>1：来自 CLB_GP_REG[7]。<br>复位类型：SYSRSn。 |
| 6 | SEL_GP_IN_6 | R/W | 0h | 输入 6 的选择控制，选择外部输入或 CLB_GP_REG[6]。<br>0：来自选定的外部输入。<br>1：来自 CLB_GP_REG[6]。<br>复位类型：SYSRSn。 |
| 5 | SEL_GP_IN_5 | R/W | 0h | 输入 5 的选择控制，选择外部输入或 CLB_GP_REG[5]。<br>0：来自选定的外部输入。<br>1：来自 CLB_GP_REG[5]。<br>复位类型：SYSRSn。 |
| 4 | SEL_GP_IN_4 | R/W | 0h | 输入 4 的选择控制，选择外部输入或 CLB_GP_REG[4]。<br>0：来自选定的外部输入。<br>1：来自 CLB_GP_REG[4]。<br>复位类型：SYSRSn。 |
| 3 | SEL_GP_IN_3 | R/W | 0h | 输入 3 的选择控制，选择外部输入或 CLB_GP_REG[3]。<br>0：来自选定的外部输入。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：来自 CLB_GP_REG[3]。<br>复位类型：SYSRSn。 |
| 2 | SEL_GP_IN_2 | R/W | 0h | 输入 2 的选择控制，选择外部输入或 CLB_GP_REG[2]。<br>0：来自选定的外部输入。<br>1：来自 CLB_GP_REG[2]。<br>复位类型：SYSRSn。 |
| 1 | SEL_GP_IN_1 | R/W | 0h | 输入 1 的选择控制，选择外部输入或 CLB_GP_REG[1]。<br>0：来自选定的外部输入。<br>1：来自 CLB_GP_REG[1]。<br>复位类型：SYSRSn。 |
| 0 | SEL_GP_IN_0 | R/W | 0h | 输入 0 的选择控制，选择外部输入或 CLB_GP_REG[0]。<br>0：来自选定的外部输入。<br>1：来自 CLB_GP_REG[0]。<br>复位类型：SYSRSn。 |

### 10.4.3.6 CLB_LCL_MUX_SEL_1 寄存器

偏移地址：14h　复位值：00000000h

本地多路复用器的输入选择

表 10-109 CLB_LCL_MUX_SEL_1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MISC_INPUT_SEL_3 | MISC_INPUT_SEL_2 | MISC_INPUT_SEL_1 | MISC_INPUT_SEL_0 | 保留 | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | LCL_MUX_SEL_IN_3 | | | |
| R-0-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LCL_MUX_SEL_IN_3 | LCL_MUX_SEL_IN_2 | | | | | LCL_MUX_SEL_IN_1 | |
| R/W-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LCL_MUX_SEL_IN_1 | | | LCL_MUX_SEL_IN_0 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-110 CLB_LCL_MUX_SEL_1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MISC_INPUT_SEL_3 | R/W | 0h | 置 1 时，对应 LCL_MUX_SEL_IN 的取值范围为 32 至 63。<br>复位类型：SYSRSn。 |
| 30 | MISC_INPUT_SEL_2 | R/W | 0h | 置 1 时，对应 LCL_MUX_SEL_IN 的取值范围为 32 至 63。<br>复位类型：SYSRSn。 |
| 29 | MISC_INPUT_SEL_1 | R/W | 0h | 置 1 时，对应 LCL_MUX_SEL_IN 的取值范围为 32 至 63。<br>复位类型：SYSRSn。 |
| 28 | MISC_INPUT_SEL_0 | R/W | 0h | 置 1 时，对应 LCL_MUX_SEL_IN 的取值范围为 32 至 63。<br>复位类型：SYSRSn。 |
| 27-20 | RESERVED | R-0 | 0h | 保留。 |
| 19-15 | LCL_MUX_SEL_IN_3 | R/W | 0h | 输入 3 的本地多路复用控制的 5 位选择值。请参阅本地信号与多路复用选择表。<br>复位类型：SYSRSn。 |
| 14-10 | LCL_MUX_SEL_IN_2 | R/W | 0h | 输入 2 的本地多路复用控制的 5 位选择值。请参阅本地信号与多路复用选择表。<br>复位类型：SYSRSn。 |
| 9-5 | LCL_MUX_SEL_IN_1 | R/W | 0h | 输入 1 的本地多路复用控制的 5 位选择值。请参阅本地信号与多路复用选择表。<br>复位类型：SYSRSn。 |
| 4-0 | LCL_MUX_SEL_IN_0 | R/W | 0h | 输入 0 的本地多路复用控制的 5 位选择值。请参阅本地信号与多路复用选择表。<br>复位类型：SYSRSn。 |

### 10.4.3.7 CLB_LCL_MUX_SEL_2 寄存器

偏移地址：18h　复位值：00000000h

本地多路复用器的输入选择

表 10-111 CLB_LCL_MUX_SEL_2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MISC_INPUT_SEL_7 | MISC_INPUT_SEL_6 | MISC_INPUT_SEL_5 | MISC_INPUT_SEL_4 | 保留 | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | LCL_MUX_SEL_IN_7 | | | |
| R-0-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LCL_MUX_SEL_IN_7 | LCL_MUX_SEL_IN_6 | | | | | LCL_MUX_SEL_IN_5 | |
| R/W-0h | R/W-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LCL_MUX_SEL_IN_5 | | | LCL_MUX_SEL_IN_4 | | | | |
| R/W-0h | | | R/W-0h | | | | |

表 10-112 CLB_LCL_MUX_SEL_2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MISC_INPUT_SEL_7 | R/W | 0h | 置 1 时，对应 LCL_MUX_SEL_IN 的取值范围为 32 至 63。<br>复位类型：SYSRSn。 |
| 30 | MISC_INPUT_SEL_6 | R/W | 0h | 置 1 时，对应 LCL_MUX_SEL_IN 的取值范围为 32 至 63。<br>复位类型：SYSRSn。 |
| 29 | MISC_INPUT_SEL_5 | R/W | 0h | 置 1 时，对应 LCL_MUX_SEL_IN 的取值范围为 32 至 63。<br>复位类型：SYSRSn。 |
| 28 | MISC_INPUT_SEL_4 | R/W | 0h | 置 1 时，对应 LCL_MUX_SEL_IN 的取值范围为 32 至 63。<br>复位类型：SYSRSn。 |
| 27-20 | RESERVED | R-0 | 0h | 保留。 |
| 19-15 | LCL_MUX_SEL_IN_7 | R/W | 0h | 输入 7 的本地多路复用控制的 5 位选择值。请参阅本地信号与多路复用选择表。<br>复位类型：SYSRSn。 |
| 14-10 | LCL_MUX_SEL_IN_6 | R/W | 0h | 输入 6 的本地多路复用控制的 5 位选择值。请参阅本地信号与多路复用选择表。<br>复位类型：SYSRSn。 |
| 9-5 | LCL_MUX_SEL_IN_5 | R/W | 0h | 输入 5 的本地多路复用控制的 5 位选择值。请参阅本地信号与多路复用选择表。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 4-0 | LCL_MUX_SEL_IN_4 | R/W | 0h | 输入 4 的本地多路复用控制的 5 位选择值。请参阅本地信号与多路复用选择表。<br>复位类型：SYSRSn。 |

### 10.4.3.8 CLB_BUF_PTR 寄存器

偏移地址：1Ch　复位值：00000000h

PUSH 和 PULL 指针

表 10-113 CLB_BUF_PTR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PUSH | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PULL | | | | | | | |
| R/W-0h | | | | | | | |

表 10-114 CLB_BUF_PTR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-24 | RESERVED | R-0 | 0h | 保留。 |
| 23-16 | PUSH | R/W | 0h | 8 位指针，表示高级控制器已从缓冲器取出的数据值数量。此计数器在 0xff 之后回绕。最低 2 位用作操作的实际指针。<br>复位类型：SYSRSn。 |
| 15-8 | RESERVED | R-0 | 0h | 保留。 |
| 7-0 | PULL | R/W | 0h | 8 位指针，表示高级控制器已写入缓冲器的数据值数量。最低 2 位用作操作的实际指针。<br>复位类型：SYSRSn。 |

### 10.4.3.9 CLB_GP_REG 寄存器

偏移地址：20h　复位值：00000000h

CELL 输入通用寄存器

表 10-115 CLB_GP_REG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SW_RLS_CTRL_7 | SW_RLS_CTRL_6 | SW_RLS_CTRL_5 | SW_RLS_CTRL_4 | SW_RLS_CTRL_3 | SW_RLS_CTRL_2 | SW_RLS_CTRL_1 | SW_RLS_CTRL_0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SW_GATING_CTRL_7 | SW_GATING_CTRL_6 | SW_GATING_CTRL_5 | SW_GATING_CTRL_4 | SW_GATING_CTRL_3 | SW_GATING_CTRL_2 | SW_GATING_CTRL_1 | SW_GATING_CTRL_0 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| REG | | | | | | | |
| R/W-0h | | | | | | | |

表 10-116 CLB_GP_REG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | SW_RLS_CTRL_7 | R/W | 0h | 异步输出调理模块输出 7 的软件释放控制。<br>复位类型：SYSRSn。 |
| 30 | SW_RLS_CTRL_6 | R/W | 0h | 异步输出调理模块输出 6 的软件释放控制。<br>复位类型：SYSRSn。 |
| 29 | SW_RLS_CTRL_5 | R/W | 0h | 异步输出调理模块输出 5 的软件释放控制。<br>复位类型：SYSRSn。 |
| 28 | SW_RLS_CTRL_4 | R/W | 0h | 异步输出调理模块输出 4 的软件释放控制。<br>复位类型：SYSRSn。 |
| 27 | SW_RLS_CTRL_3 | R/W | 0h | 异步输出调理模块输出 3 的软件释放控制。<br>复位类型：SYSRSn。 |
| 26 | SW_RLS_CTRL_2 | R/W | 0h | 异步输出调理模块输出 2 的软件释放控制。<br>复位类型：SYSRSn。 |
| 25 | SW_RLS_CTRL_1 | R/W | 0h | 异步输出调理模块输出 1 的软件释放控制。<br>复位类型：SYSRSn。 |
| 24 | SW_RLS_CTRL_0 | R/W | 0h | 异步输出调理模块输出 0 的软件释放控制。<br>复位类型：SYSRSn。 |
| 23 | SW_GATING_CTRL_7 | R/W | 0h | 异步输出调理模块输出 7 的软件门控控制。<br>复位类型：SYSRSn。 |
| 22 | SW_GATING_CTRL_6 | R/W | 0h | 异步输出调理模块输出 6 的软件门控控制。<br>复位类型：SYSRSn。 |
| 21 | SW_GATING_CTRL_5 | R/W | 0h | 异步输出调理模块输出 5 的软件门控控制。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 20 | SW_GATING_CTRL_4 | R/W | 0h | 异步输出调理模块输出 4 的软件门控控制。<br>复位类型：SYSRSn。 |
| 19 | SW_GATING_CTRL_3 | R/W | 0h | 异步输出调理模块输出 3 的软件门控控制。<br>复位类型：SYSRSn。 |
| 18 | SW_GATING_CTRL_2 | R/W | 0h | 异步输出调理模块输出 2 的软件门控控制。<br>复位类型：SYSRSn。 |
| 17 | SW_GATING_CTRL_1 | R/W | 0h | 异步输出调理模块输出 1 的软件门控控制。<br>复位类型：SYSRSn。 |
| 16 | SW_GATING_CTRL_0 | R/W | 0h | 异步输出调理模块输出 0 的软件门控控制。<br>复位类型：SYSRSn。 |
| 15-8 | RESERVED | R-0 | 0h | 保留。 |
| 7-0 | REG | R/W | 0h | 此 8 位在 CLB_IN_MUX_SEL_0 寄存器选择相应位时，直接连接到 CELL 的 8 个输入。<br>复位类型：SYSRSn。 |

### 10.4.3.10 CLB_OUT_EN 寄存器

偏移地址：24h　复位值：00000000h

CELL 输出使能寄存器

表 10-117 CLB_OUT_EN 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| OUT0 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| OUT0 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| OUT0 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUT0 | | | | | | | |
| R/W-0h | | | | | | | |

表 10-118 CLB_OUT_EN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | OUT0 | R/W | 0h | 此 32 位直接驱动 OUTPUT_EN 信号。使能位 x（x = 0:31）将覆盖复用到 CLB OUTx 的对应外设信号。<br>复位类型：SYSRSn。 |

### 10.4.3.11 CLB_GLBL_MUX_SEL_1 寄存器

偏移地址：28h　复位值：00000000h

CELL 输入的全局多路复用选择

表 10-119 CLB_GLBL_MUX_SEL_1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | GLBL_MUX_SEL_IN_3 | | | |
| R-0-0h | | | | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GLBL_MUX_SEL_IN_3 | | | GLBL_MUX_SEL_IN_2 | | | | |
| R/W-0h | | | R/W-0h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GLBL_MUX_SEL_IN_2 | | GLBL_MUX_SEL_IN_1 | | | | | |
| R/W-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GLBL_MUX_SEL_IN_1 | GLBL_MUX_SEL_IN_0 | | | | | | |
| R/W-0h | R/W-0h | | | | | | |

表 10-120 CLB_GLBL_MUX_SEL_1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-28 | RESERVED | R-0 | 0h | 保留。 |
| 27-21 | GLBL_MUX_SEL_IN_3 | R/W | 0h | 输入 3 的全局多路复用控制的 7 位选择值。请参阅全局信号与多路复用选择表。<br>复位类型：SYSRSn。 |
| 20-14 | GLBL_MUX_SEL_IN_2 | R/W | 0h | 输入 2 的全局多路复用控制的 7 位选择值。请参阅全局信号与多路复用选择表。<br>复位类型：SYSRSn。 |
| 13-7 | GLBL_MUX_SEL_IN_1 | R/W | 0h | 输入 1 的全局多路复用控制的 7 位选择值。请参阅全局信号与多路复用选择表。<br>复位类型：SYSRSn。 |
| 6-0 | GLBL_MUX_SEL_IN_0 | R/W | 0h | 输入 0 的全局多路复用控制的 7 位选择值。请参阅全局信号与多路复用选择表。<br>复位类型：SYSRSn。 |

### 10.4.3.12 CLB_GLBL_MUX_SEL_2 寄存器

偏移地址：2Ch　复位值：00000000h

CELL 输入的全局多路复用选择

表 10-121 CLB_GLBL_MUX_SEL_2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | GLBL_MUX_SEL_IN_7 | | | |
| R-0-0h | | | | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| GLBL_MUX_SEL_IN_7 | | | GLBL_MUX_SEL_IN_6 | | | | |
| R/W-0h | | | R/W-0h | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| GLBL_MUX_SEL_IN_6 | | GLBL_MUX_SEL_IN_5 | | | | | |
| R/W-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GLBL_MUX_SEL_IN_5 | GLBL_MUX_SEL_IN_4 | | | | | | |
| R/W-0h | R/W-0h | | | | | | |

表 10-122 CLB_GLBL_MUX_SEL_2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-28 | RESERVED | R-0 | 0h | 保留。 |
| 27-21 | GLBL_MUX_SEL_IN_7 | R/W | 0h | 输入 7 的全局多路复用控制的 7 位选择值。请参阅全局信号与多路复用选择表。<br>复位类型：SYSRSn。 |
| 20-14 | GLBL_MUX_SEL_IN_6 | R/W | 0h | 输入 6 的全局多路复用控制的 7 位选择值。请参阅全局信号与多路复用选择表。<br>复位类型：SYSRSn。 |
| 13-7 | GLBL_MUX_SEL_IN_5 | R/W | 0h | 输入 5 的全局多路复用控制的 7 位选择值。请参阅全局信号与多路复用选择表。<br>复位类型：SYSRSn。 |
| 6-0 | GLBL_MUX_SEL_IN_4 | R/W | 0h | 输入 4 的全局多路复用控制的 7 位选择值。请参阅全局信号与多路复用选择表。<br>复位类型：SYSRSn。 |

## 10.4.3.13 CLB_PRESCALE_CTRL 寄存器

偏移地址：30h　复位值：00000000h

预分频器寄存器控制

表 10-123 CLB_PRESCALE_CTRL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PRESCALE | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PRESCALE | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | TAP | | | | STRB | CLKEN |
| R-0-0h | | R/W-0h | | | | R/W-0h | R/W-0h |

表 10-124 CLB_PRESCALE_CTRL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | PRESCALE | R/W | 0h | 16 位预分频值，作为计数器达到该值后复位的参考值。计数器为简单递增计数器，计数到参考值后复位为 0；只要计数器使能，此过程便持续重复。此 16 位寄存器值用作 16 位计数器清零的参考值。<br>复位类型：SYSRSn。 |
| 15-6 | RESERVED | R-0 | 0h | 保留。 |
| 5-2 | TAP | R/W | 0h | 抽头选择值。此 4 位从计数器寄存器的 16 个位中选择一个作为输出抽头。<br>0000：选择计数器位 0。<br>0001：选择计数器位 1。<br>……<br>1111：选择计数器位 15。<br>复位类型：SYSRSn。 |
| 1 | STRB | R/W | 0h | 置 0 时，每当计数器值与 PRESCALE_VALUE 匹配，便输出选通信号。置 1 时，输出 TAP_SELECT_VALUE 所选计数器寄存器位的值。<br>复位类型：SYSRSn。 |
| 0 | CLKEN | R/W | 0h | 使能预分频时钟 / 选通信号发生器。使用 16 位计数器产生选通信号，或将选定的计数器位输出至 CLB CELL。该通用选通信号 / 预分频时钟可供 CELL 逻辑按需使用，并通过一个 LCL_IN 多路复用端口送入 CELL。<br>复位类型：SYSRSn。 |

### 10.4.3.14 CLB_INTR_TAG_REG 寄存器

偏移地址：40h　复位值：0000h

中断标签寄存器

表 10-125 CLB_INTR_TAG_REG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | TAG | | | | | |
| R-0-0h | | R/W-0h | | | | | |

表 10-126 CLB_INTR_TAG_REG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-6 | RESERVED | R-0 | 0h | 保留。 |
| 5-0 | TAG | R/W | 0h | 高级控制器使用此 6 位，在触发中断时设置标签值。由于可通过 VBUS 写入，因此也可通过 VBUS 接口清除此值。<br>复位类型：SYSRSn。 |

### 10.4.3.15 CLB_LOCK 寄存器

偏移地址：44h　复位值：00000000h

锁定控制寄存器

表 10-127 CLB_LOCK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| WSonce-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| KEY | | | | | | | |
| WSonce-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | LOCK |
| R-0-0h | | | | | | | R/W-0h |

表 10-128 CLB_LOCK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | KEY | WSonce | 0h | 此 16 位作为密钥，使能对本寄存器位 0 的写入。只有执行单次 32 位写入，并且位 31:16 等于 0x5a5a、位 0 为 1 时，才可将位 0 写为 1。其他所有写入均被忽略，包括分开的 16 位写入。此字段受 EALLOW 保护。<br>复位类型：SYSRSn。 |
| 15-1 | RESERVED | R-0 | 0h | 保留。 |
| 0 | LOCK | R/W | 0h | 此位为仅可置位一次的位（Set Once）。一旦置为 1，只有复位（SYSRSN 0）才能将其清为 0。<br>复位类型：SYSRSn。 |

## 10.4.3.16 CLB_HLC_INSTR_READ_PTR 寄存器

偏移地址：48h　复位值：0000h

HLC 指令读取指针

表 10-129 CLB_HLC_INSTR_READ_PTR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | READ_PTR | | | | |
| R-0-0h | | | R/W-0h | | | | |

表 10-130 CLB_HLC_INSTR_READ_PTR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-5 | RESERVED | R-0 | 0h | 保留。 |
| 4-0 | READ_PTR | R/W | 0h | 此 5 位值用作读取 HLC 指令存储器的地址指针。<br>复位类型：SYSRSn。 |

## 10.4.3.17 CLB_HLC_INSTR_VALUE 寄存器

偏移地址：4Ch　复位值：0000h

HLC 指令读取值

表 10-131 CLB_HLC_INSTR_VALUE 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | INSTR | | | |
| R-0-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INSTR | | | | | | | |
| R-0h | | | | | | | |

表 10-132 CLB_HLC_INSTR_VALUE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R-0 | 0h | 保留。 |
| 11-0 | INSTR | R | 0h | 此 12 位值用于读取 CLB_HLC_INSTR_READ_PTR 寄存器所指 HLC 指令存储器地址的内容。<br>复位类型：SYSRSn。 |

## 10.4.3.18 CLB_DBG_OUT_2 寄存器

偏移地址：5Ch 复位值：00000000h

CLB 输入与最终异步输出的状态观测

表 10-133 CLB_DBG_OUT_2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| IN | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OUT | | | | | | | |
| R/W-0h | | | | | | | |

表 10-134 CLB_DBG_OUT_2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R/W1C | 0h | 保留。 |
| 15-8 | IN | R/W | 0h | 反映经过选择和输入调理后，最终送入 CELL 的 8 个输入的状态。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 7-0 | OUT | R/W | 0h | 反映输出调理模块 8 个输出的状态。<br>复位类型：SYSRSn。 |

### 10.4.3.19 CLB_DBG_R0 寄存器

偏移地址：60h　复位值：00000000h

高级控制器的 R0

表 10-135 CLB_DBG_R0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBG | | | | | | | |
| R-0h | | | | | | | |

表 10-136 CLB_DBG_R0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | DBG | R | 0h | CLB_DBG_R0。<br>复位类型：SYSRSn。 |

### 10.4.3.20 CLB_DBG_R1 寄存器

偏移地址：64h　复位值：00000000h

高级控制器的 R1

表 10-137 CLB_DBG_R1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBG | | | | | | | |
| R-0h | | | | | | | |

表 10-138 CLB_DBG_R1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | DBG | R | 0h | CLB_DBG_R1。<br>复位类型：SYSRSn。 |

## 10.4.3.21 CLB_DBG_R2 寄存器

偏移地址：68h　复位值：00000000h

高级控制器的 R2

表 10-139 CLB_DBG_R2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBG | | | | | | | |
| R-0h | | | | | | | |

表 10-140 CLB_DBG_R2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | DBG | R | 0h | CLB_DBG_R2。<br>复位类型：SYSRSn。 |

## 10.4.3.22 CLB_DBG_R3 寄存器

偏移地址：6Ch　复位值：00000000h

高级控制器的 R3

表 10-141 CLB_DBG_R3 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBG | | | | | | | |
| R-0h | | | | | | | |

表 10-142 CLB_DBG_R3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | DBG | R | 0h | CLB_DBG_R3。<br>复位类型：SYSRSn。 |

## 10.4.3.23 CLB_DBG_C0 寄存器

偏移地址：70h　复位值：00000000h

单元 0 的计数值

表 10-143 CLB_DBG_C0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBG | | | | | | | |
| R-0h | | | | | | | |

表 10-144 CLB_DBG_C0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | DBG | R | 0h | CLB_DBG_C0。<br>复位类型：SYSRSn。 |

## 10.4.3.24 CLB_DBG_C1 寄存器

偏移地址：74h　复位值：00000000h

单元 1 的计数值

表 10-145 CLB_DBG_C1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBG | | | | | | | |
| R-0h | | | | | | | |

表 10-146 CLB_DBG_C1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | DBG | R | 0h | CLB_DBG_C1。<br>复位类型：SYSRSn。 |

## 10.4.3.25 CLB_DBG_C2 寄存器

偏移地址：78h　复位值：00000000h

单元 2 的计数值

表 10-147 CLB_DBG_C2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DBG | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBG | | | | | | | |
| R-0h | | | | | | | |

表 10-148 CLB_DBG_C2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | DBG | R | 0h | CLB_DBG_C2。<br>复位类型：SYSRSn。 |

## 10.4.3.26 CLB_DBG_OUT 寄存器

偏移地址：7Ch　复位值：00010100h

CELL 中各单元的输出

表 10-149 CLB_DBG_OUT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| OUT7 | OUT6 | OUT5 | OUT4 | OUT3 | OUT2 | OUT1 | OUT0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| LUT42_OUT | FSM2_LUTOUT | FSM2_S1 | FSM2_S0 | COUNT2_MATCH1 | COUNT2_ZERO | COUNT2_MATCH2 | 保留 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-1-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LUT41_OUT | FSM1_LUTOUT | FSM1_S1 | FSM1_S0 | COUNT1_MATCH1 | COUNT1_ZERO | COUNT1_MATCH2 | 保留 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-1-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LUT40_OUT | FSM0_LUTOUT | FSM0_S1 | FSM0_S0 | COUNT0_MATCH1 | COUNT0_ZERO | COUNT0_MATCH2 | 保留 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0-0h |

表 10-150 CLB_DBG_OUT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | OUT7 | R | 0h | CELL 输出 7。<br>复位类型：SYSRSn。 |
| 30 | OUT6 | R | 0h | CELL 输出 6。<br>复位类型：SYSRSn。 |
| 29 | OUT5 | R | 0h | CELL 输出 5。<br>复位类型：SYSRSn。 |
| 28 | OUT4 | R | 0h | CELL 输出 4。<br>复位类型：SYSRSn。 |
| 27 | OUT3 | R | 0h | CELL 输出 3。<br>复位类型：SYSRSn。 |
| 26 | OUT2 | R | 0h | CELL 输出 2。<br>复位类型：SYSRSn。 |
| 25 | OUT1 | R | 0h | CELL 输出 1。<br>复位类型：SYSRSn。 |
| 24 | OUT0 | R | 0h | CELL 输出 0。<br>复位类型：SYSRSn。 |
| 23 | LUT42_OUT | R | 0h | 单元 2 的 LUT4_OUT。<br>复位类型：SYSRSn。 |
| 22 | FSM2_LUTOUT | R | 0h | 单元 2 的 FSM_LUT_OUT。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 21 | FSM2_S1 | R | 0h | 单元 2 的 FSM_S1。<br>复位类型：SYSRSn。 |
| 20 | FSM2_S0 | R | 0h | 单元 2 的 FSM_S0。<br>复位类型：SYSRSn。 |
| 19 | COUNT2_MATCH1 | R | 0h | 单元 2 的 COUNT_MATCH1。<br>复位类型：SYSRSn。 |
| 18 | COUNT2_ZERO | R | 0h | 单元 2 的 COUNT_ZERO。<br>复位类型：SYSRSn。 |
| 17 | COUNT2_MATCH2 | R | 0h | 单元 2 的 COUNT_MATCH2。<br>复位类型：SYSRSn。 |
| 16 | RESERVED | R-1 | 1h | 保留。 |
| 15 | LUT41_OUT | R | 0h | 单元 1 的 LUT4_OUT。<br>复位类型：SYSRSn。 |
| 14 | FSM1_LUTOUT | R | 0h | 单元 1 的 FSM_LUT_OUT。<br>复位类型：SYSRSn。 |
| 13 | FSM1_S1 | R | 0h | 单元 1 的 FSM_S1。<br>复位类型：SYSRSn。 |
| 12 | FSM1_S0 | R | 0h | 单元 1 的 FSM_S0。<br>复位类型：SYSRSn。 |
| 11 | COUNT1_MATCH1 | R | 0h | 单元 1 的 COUNT_MATCH1。<br>复位类型：SYSRSn。 |
| 10 | COUNT1_ZERO | R | 0h | 单元 1 的 COUNT_ZERO。<br>复位类型：SYSRSn。 |
| 9 | COUNT1_MATCH2 | R | 0h | 单元 1 的 COUNT_MATCH2。<br>复位类型：SYSRSn。 |
| 8 | RESERVED | R-1 | 1h | 保留。 |
| 7 | LUT40_OUT | R | 0h | 单元 0 的 LUT4_OUT。<br>复位类型：SYSRSn。 |
| 6 | FSM0_LUTOUT | R | 0h | 单元 0 的 FSM_LUT_OUT。<br>复位类型：SYSRSn。 |
| 5 | FSM0_S1 | R | 0h | 单元 0 的 FSM_S1。<br>复位类型：SYSRSn。 |
| 4 | FSM0_S0 | R | 0h | 单元 0 的 FSM_S0。<br>复位类型：SYSRSn。 |
| 3 | COUNT0_MATCH1 | R | 0h | 单元 0 的 COUNT_MATCH1。<br>复位类型：SYSRSn。 |
| 2 | COUNT0_ZERO | R | 0h | 单元 0 的 COUNT_ZERO。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | COUNT0_MATCH2 | R | 0h | 单元 0 的 COUNT_MATCH2。<br>复位类型：SYSRSn。 |
| 0 | RESERVED | R-0 | 0h | 保留。 |

## 10.4.4 CLB_DATA_EXCHANGE_REGS 寄存器

CLB_DATA_EXCHANGE_REGS 寄存器组的存储映射偏移地址如下表所示，表中未列出的偏移地址都是保留地址空间。

表 10-151 CLB_DATA_EXCHANGE_REGS

| 偏移地址 | 寄存器标识 | 寄存器名称 |
|---|---|---|
| 0x0 + i | CLB_PUSH_y | HLC PUSH FIFO寄存器 |
| 0x200 + i | CLB_PULL_y | HLC PULL FIFO寄存器 |

下表列出了寄存器描述使用的访问类型编码：

表 10-152 CLB_DATA_EXCHANGE_REGS 访问类型编码

| 访问类型 | 编码 | 描述 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| 写类型 | | |
| W | W | 写 |
| 复位或默认值 | | |
| *-n* | | 复位值或默认值 |
| 寄存器数组变量 | | |
| i,j,k,l,m,n | | 当这些变量用在寄存器名称、偏移量或地址中，它们表示一个寄存器数组值，并且这个寄存器是一组重复寄存器的一部分。这组寄存器组成一个层次化结构，数组用公式表示。 |
| y | | 当这个变量用在寄存器名称、偏移量或地址中，它表示一个寄存器数组值。 |

### 10.4.4.1 CLB_PUSH_y（y = 0h~3h）

偏移地址：0h + i（i = y * 4h）；复位值：0h。

表 10-153 CLB_PUSH 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PUSH | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PUSH | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PUSH | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PUSH | | | | | | | |
| R-0h | | | | | | | |

表 10-154 CLB_PUSH_y 寄存器描述（y = 0h~3h）

| 数位 | 域 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | PUSH | R | 0h | HLC的PUSH FIFO寄存器。<br>复位类型：SYSRSn |

### 10.4.4.2 CLB_PULL_y（y = 0h~3h）

偏移地址：200h + i（i = y * 4h）；复位值：0h。

表 10-155 CLB_PULL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PULL | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PULL | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PULL | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PULL | | | | | | | |
| R/W-0h | | | | | | | |

表 10-156 CLB_PULL_y 寄存器描述（y = 0h~3h）

| 数位 | 域 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | PULL | R/W | 0h | HLC的PULL FIFO寄存器。<br>复位类型：SYSRSn |

### 10.4.5 CLB 寄存器与 DriverLib 函数对应表

表 10-157 CLB 寄存器与 DriverLib 函数对应表

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| COUNT_RESET | clb.h | CLB_selectCounterInputs |
| COUNT_MODE_1 | clb.h | CLB_selectCounterInputs |
| COUNT_MODE_0 | clb.h | CLB_selectCounterInputs |
| COUNT_EVENT | clb.h | CLB_selectCounterInputs |
| FSM_EXTRA_IN0 | clb.h | CLB_selectFSMInputs |
| FSM_EXTERNAL_IN0 | clb.h | CLB_selectFSMInputs |
| FSM_EXTERNAL_IN1 | clb.h | CLB_selectFSMInputs |
| FSM_EXTRA_IN1 | clb.h | CLB_selectFSMInputs |
| LUT4_IN0 | clb.h | CLB_selectLUT4Inputs |
| LUT4_IN1 | clb.h | CLB_selectLUT4Inputs |
| LUT4_IN2 | clb.h | CLB_selectLUT4Inputs |
| LUT4_IN3 | clb.h | CLB_selectLUT4Inputs |
| FSM_LUT_FN1_0 | clb.h | CLB_configFSMLUTFunction |
| FSM_LUT_FN2 | clb.h | CLB_configFSMLUTFunction |
| LUT4_FN1_0 | clb.h | CLB_configLUT4Function |
| LUT4_FN2 | clb.h | CLB_configLUT4Function |
| FSM_NEXT_STATE_0 | clb.h | CLB_configFSMNextState |
| FSM_NEXT_STATE_1 | clb.h | CLB_configFSMNextState |
| FSM_NEXT_STATE_2 | clb.h | CLB_configFSMNextState |
| MISC_CONTROL | clb.h | CLB_configMiscCtrlModes |
| OUTPUT_LUT_0 | clb.h | CLB_configOutputLUT |
| OUTPUT_LUT_1 | - | 参见 OUTPUT_LUT_0 |
| OUTPUT_LUT_2 | - | 参见 OUTPUT_LUT_0 |
| OUTPUT_LUT_3 | - | 参见 OUTPUT_LUT_0 |
| OUTPUT_LUT_4 | - | 参见 OUTPUT_LUT_0 |
| OUTPUT_LUT_5 | - | 参见 OUTPUT_LUT_0 |
| OUTPUT_LUT_6 | - | 参见 OUTPUT_LUT_0 |
| OUTPUT_LUT_7 | - | 参见 OUTPUT_LUT_0 |
| HLC_EVENT_SEL | clb.h | CLB_configHLCEventSelect |
| COUNT_MATCH_TAP_SEL | clb.h | CLB_configCounterTapSelects |
| OUTPUT_COND_CTRL_0 | clb.h | CLB_configAOC |
| OUTPUT_COND_CTRL_1 | - | - |
| OUTPUT_COND_CTRL_2 | - | - |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| OUTPUT_COND_CTRL_3 | - | - |
| OUTPUT_COND_CTRL_4 | - | - |
| OUTPUT_COND_CTRL_5 | - | - |
| OUTPUT_COND_CTRL_6 | - | - |
| OUTPUT_COND_CTRL_7 | - | - |
| MISC_ACCESS_CTRL | clb.h | CLB_disableOutputMaskUpdates |
| MISC_ACCESS_CTRL | clb.h | CLB_enableOutputMaskUpdates |
| MISC_ACCESS_CTRL | clb.h | CLB_disableSPIBufferAccess |
| MISC_ACCESS_CTRL | clb.h | CLB_enableSPIBufferAccess |
| SPI_DATA_CTRL_HI | clb.h | CLB_configSPIBufferLoadSignal |
| SPI_DATA_CTRL_HI | clb.h | CLB_configSPIBufferShift |
| LOAD_EN | clb.h | CLB_enableCLB |
| LOAD_EN | clb.h | CLB_disableCLB |
| LOAD_EN | clb.h | CLB_enableNMI |
| LOAD_EN | clb.h | CLB_disableNMI |
| LOAD_EN | clb.h | CLB_writeInterface |
| LOAD_EN | clb.h | CLB_enablePipelineMode |
| LOAD_EN | clb.h | CLB_disablePipelineMode |
| LOAD_ADDR | clb.h | CLB_writeInterface |
| LOAD_DATA | clb.h | CLB_writeInterface |
| INPUT_FILTER | clb.h | CLB_selectInputFilter |
| INPUT_FILTER | clb.h | CLB_enableSynchronization |
| INPUT_FILTER | clb.h | CLB_disableSynchronization |
| INPUT_FILTER | clb.h | CLB_enableInputPipelineMode |
| INPUT_FILTER | clb.h | CLB_disableInputPipelineMode |
| IN_MUX_SEL_0 | clb.h | CLB_configGPInputMux |
| LCL_MUX_SEL_1 | clb.h | CLB_configLocalInputMux |
| LCL_MUX_SEL_2 | clb.h | CLB_configLocalInputMux |
| BUF_PTR | clb.c | CLB_clearFIFOs |
| GP_REG | clb.h | CLB_writeSWReleaseControl |
| GP_REG | clb.h | CLB_writeSWGateControl |
| GP_REG | clb.h | CLB_setGPREG |
| GP_REG | clb.h | CLB_getGPREG |
| OUT_EN | clb.h | CLB_setOutputMask |
| GLBL_MUX_SEL_1 | clb.h | CLB_configGlobalInputMux |
| GLBL_MUX_SEL_2 | clb.h | CLB_configGlobalInputMux |

| 寄存器 | 文件 | DriverLib 函数 |
| --- | --- | --- |
| PRESCALE_CTRL | clb.h | CLB_configureClockPrescalar |
| PRESCALE_CTRL | clb.h | CLB_configureStrobeMode |
| INTR_TAG_REG | clb.h | CLB_getInterruptTag |
| INTR_TAG_REG | clb.h | CLB_clearInterruptTag |
| LOCK | clb.h | CLB_enableLock |
| HLC_INSTR_READ_PTR | - | - |
| HLC_INSTR_VALUE | - | - |
| DBG_OUT_2 | - | - |
| DBG_R0 | - | - |
| DBG_R1 | - | - |
| DBG_R2 | - | - |
| DBG_R3 | - | - |
| DBG_C0 | - | - |
| DBG_C1 | - | - |
| DBG_C2 | - | - |
| DBG_OUT | clb.h | CLB_getOutputStatus |
| PUSH | clb.c | CLB_readFIFOs |
| PULL | clb.c | CLB_clearFIFOs |
| PULL | clb.c | CLB_writeFIFOs |

# 11 主机接口控制器（HIC）

## 11.1 简介

### 11.1.1 特征

HIC 模块允许外部主机控制器通过模拟 ASRAM 协议直接访问设备。它有两种操作模式：直接访问和邮箱访问。在直接访问模式下，设备资源由外部主机直接写入和读取。在邮箱访问模式下，外部主机或设备向缓冲区进行读写，并在完成时相互通知。出于安全原因，HIC 必须由本设备开启使用（HICGCR.HICEN 和 HICMODECR.EN_DEVACC）。

HIC 模块具有以下特性：

- 8 位和 16 位的可配置 I/O 数据线
- 直接和邮箱访问模式
- 8 条地址线和 8 个可配置基址，共 2048 个可能的可寻址区域
- 使用邮箱访问模式时，两个 64 字节的缓冲区用于外部主机和设备
- 从其他外围设备触发 HIC

### 11.1.2 HIC 框架图

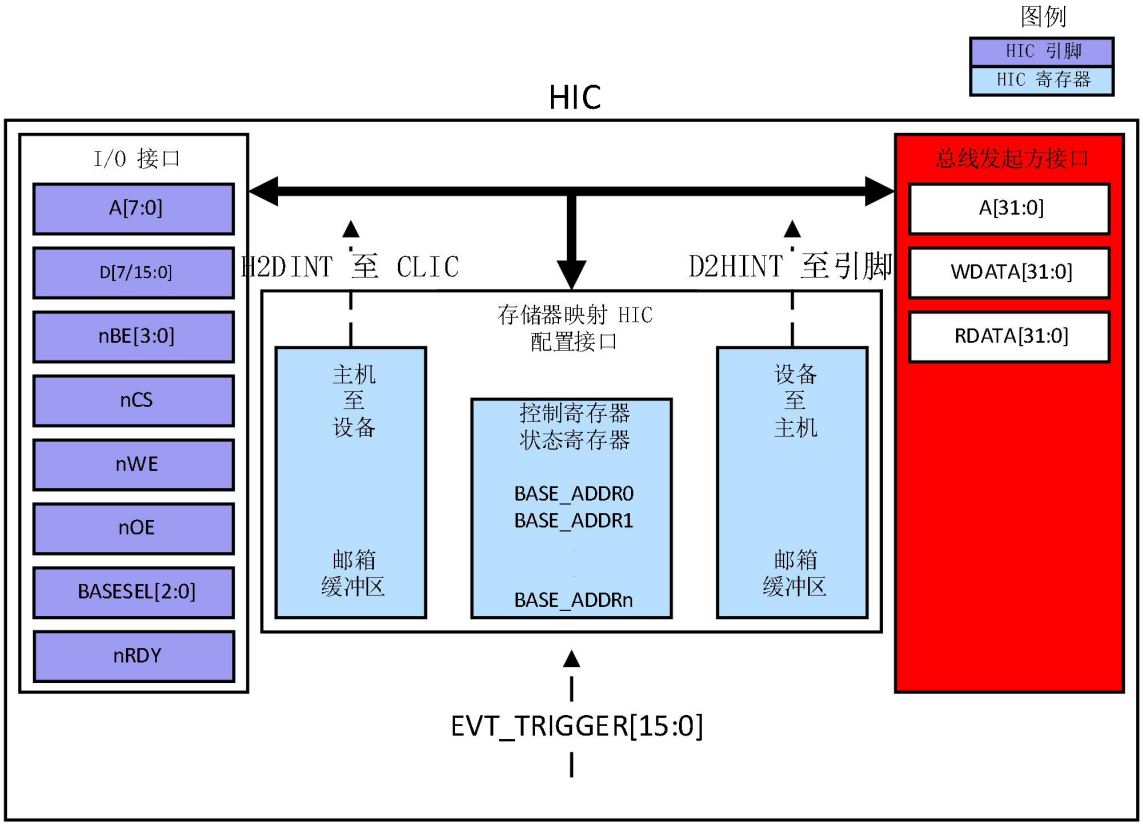



图 11-1 HIC 模块示意图

## 11.2 主要特性

- HIC 主要由三部分组成
- I/O 接口：输入和输出（I/O）接口模拟 ASRAM 协议，在外部主机和设备之间执行读写操作。
- 配置接口：配置接口由构成 HIC 模块的寄存器组成。这些寄存器包括 HIC 的控制、数据和状态寄存器，可由外部主机和设备处理器访问。
- 总线启动器接口：总线启动器接口是外部主机访问设备上的存储器或外设的端口。（注：HIC 外设访问范围中，DCAN 扩展为 FDCAN+DCAN）

### 11.2.1 HIC 资源访问

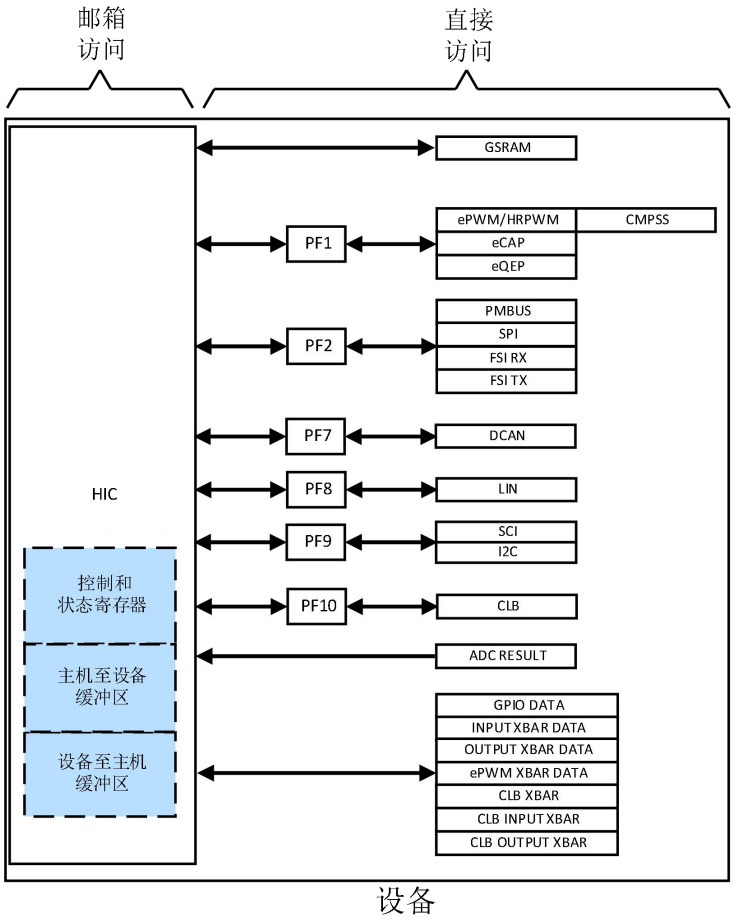



图 11-2 HIC 内存映射

### 11.2.2 外部主机交互

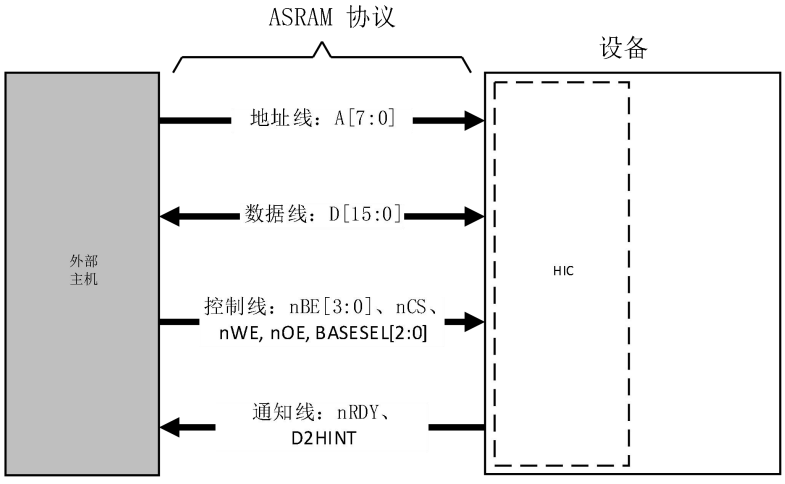



图 11-3 HIC 连接

表 11-1 HIC 连接表

| 连接组 | 连接名 | 方向 | 描述 |
|---|---|---|---|
| 地址线 | A[7:0] | 输入 | 地址线 Bit0 to Bit7 |
| 数据线 | D[15:0] | 输入/输出 | 数据线 Bit0 to Bit15 |
| 控制线 | nBE[3:0] | 输入 | 字节使能 Bit0 to Bit3 |
| | nCS | 输入 | 芯片选择 |
| | nWE | 输入 | 写使能 |
| | nOE/RnW | 输入 | 输出使能/读非写 |
| | BASESEL[2:0] | 输入 | 基地址 Bit0 to Bit2 |
| 通知线 | nRDY | 输出 | 就绪/等待 |
| | D2HINT | 输出 | 设备到主机中断 |

## 11.2.3 中断触发

### 11.2.3.1 H2DINT

H2DINT 中断信号连接到内核中断处理模块（CLIC），可通过写 HICH2DTOKEN 寄存器来触发 H2DINT，触发源如图：

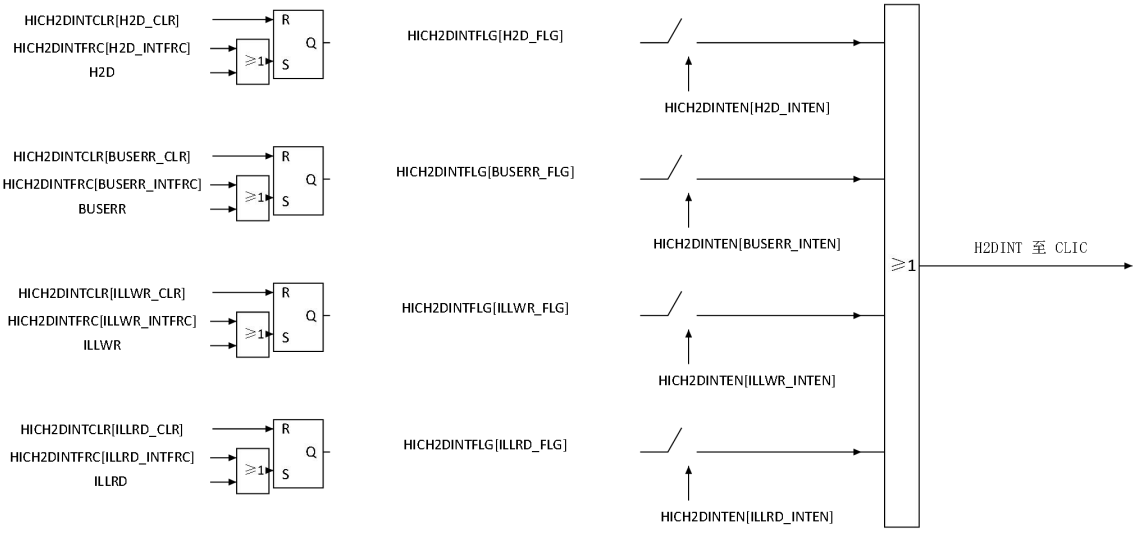



图 11-4 HIC 主机到设备中断源

触发源列表：

表 11-2 HIC 主机到设备触发源列表

| 信号来源 | 描述 |
|---|---|
| H2D | 主机到设备所传输的数据已完成将触发H2DINT端口中断（用于写操作指示写数据传输完成）（写寄存器HICH2DINTFRC实现） |

| 信号来源 | 描述 |
| --- | --- |
| BUSERR | 当host和device同时写入HIC内部资源的相同地址时，会发生此错误。主机数据将被丢弃，而设备数据将被保留。当同时访问HIC内部资源的不同地址或者相同或不同外设时，HIC均支持此操作，请求正常发出 |
| ILLWR | host写HIC内部寄存器不可写的位置时触发（即read-only for host位域） |
| ILLRD | 如果上一个写操作未完成期间（数据等待），且host发起读请求时，会发生此错误 |

### 11.2.3.2 D2HINT

设备通过 pin 口给外部主机的中断信号 D2HINT，通过写 HICD2HTOKEN 寄存器来触发 D2HINT，触发源如图：

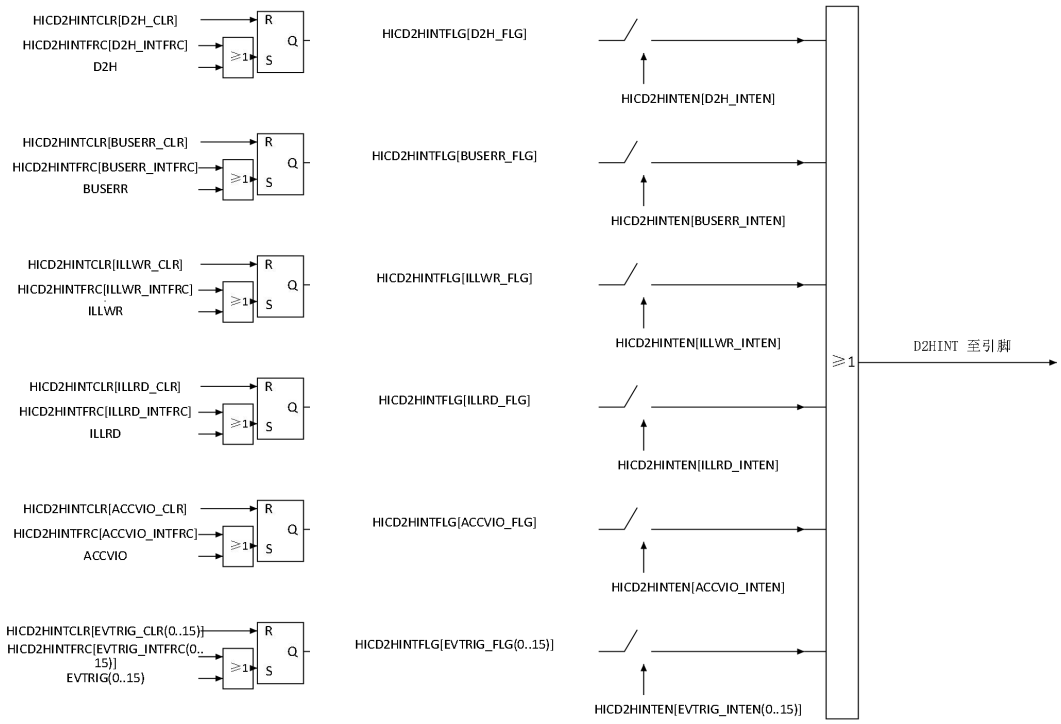



图 11-5 HIC 设备到主机中断源

触发源列表：

表 11-3 HIC 设备到主机触发源列表

| 信号来源 | 描述 |
| --- | --- |
| D2H | 设备到主机的数据准备就绪将触发D2HINT端口中断（D2H_BUF被cpu写入有效数据后希望host来读取，向其发送中断）（pin脚传输触发） |

| 信号来源 | 描述 |
| --- | --- |
| BUSERR | 当主机和设备同时写入同一内存位置时，会发生此错误。主机数据将被丢弃，而设备数据将被保留 |
| ILLWR | host写HIC-MMR不可写的位置（read-only for host）时触发<br>注：非read-only for host（host可写）位域:<br>HICBASESEL、HICHOSTCR、HICERRADDR. D2H_ERR_ADDR、HICH2DTOKEN |
| ILLRD | 如果上一个写操作未完成期间（当前单此写操作已完成，此处表明当前状态为数据等待状态），且host发起读请求时，会发生此错误 |
| ACCVIO | 当主机尝试访问SOC上非外设帧时，resp返回1触发 |
| EVTRIG_FLG | 外设中断 |

D2HINT 也能被不同的外设触发。

表 11-4 事件触发（EVTRIG）源

| 事件触发 | 外设中断源 |
| --- | --- |
| EVTRIG[0] | FSIRX_INT1, FSIRX_INT2 |
| EVTRIG[1] | FSITX_INT1, FSITX_INT2 |
| EVTRIG[2] | SPIB_TX, SPIB_RX |
| EVTRIG[3] | SPIA_TX, SPIA_RX |
| EVTRIG[4] | EPWM2, EPWM2_TZ |
| EVTRIG[5] | EPWM1, EPWM1_TZ |
| EVTRIG[6] | CLB2 |
| EVTRIG[7] | CLB1 |
| EVTRIG[8..15] | Reserved |

例如：FSI 接收器可配置为在数据标签匹配时触发中断事件（FSIRX_INT1），通过使能相应的FSIRX2HINTEN.EVVTRIG 位，可使能该事件以向主机生成HIC_INT。如果 HIC_INT 线连接到主机的具有中断能力的输入引脚，则每当在接收到的数据上存在数据标签匹配时，FSI 接收器可以向主机 CPU 生成直接中断，从而消除来自设备 CPU 的任何干预。

## 11.3 功能描述

### 11.3.1 邮箱（Mailbox）访问模式

在邮箱访问模式下，外部主机只能看到 HIC 内存映射区域，而看不到设备资源和外设。HIC 的存储器映射区域包括 HIC 控制、状态和数据寄存器。该区域的 128 个字节保留用于 HIC 控制和状态寄存器。剩余的 128 个字节用作数据/缓冲区空间，并且每个被分成 64 个字节作为主机到设备（H2D）缓冲区和设备到主机（D2H）缓冲区。注意，H2D 和 D2H 缓冲区之间的这种区别仅在名称上。设备和外部主机均可通过配置寄存器 HICMODECR 中的 H2DBUF_DEVWREN 和 D2HBUF_HOSTWREN 字段，从而去访问 128 字节的 H2D 和 D2H 组合缓冲区。图中显示了该模式所需的最小外部连接。

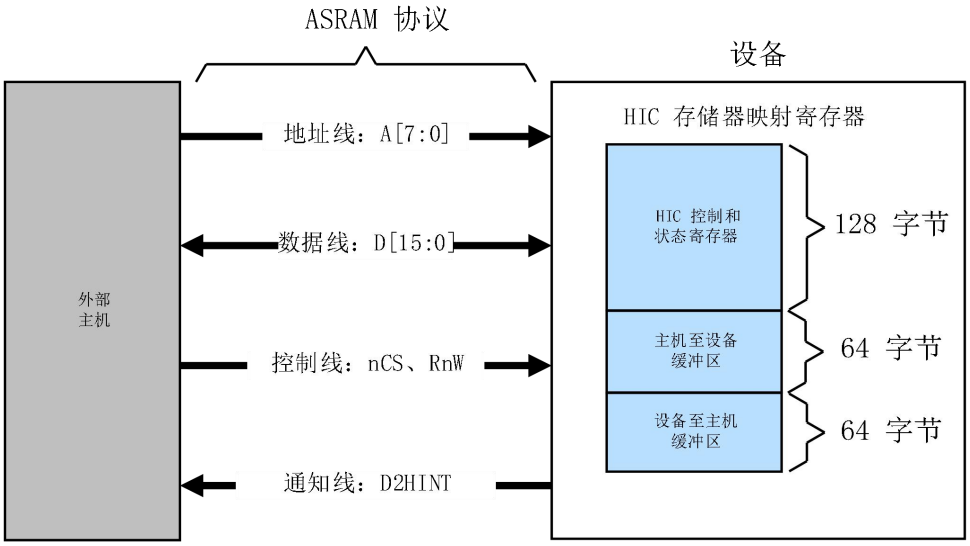



图 11-6 HIC 邮箱访问模式示意图

若要将 HIC 设置为“Mailbox 访问模式”，BASESEL[2:0]引脚必须设置为 000。然后，地址线 A[7:0]可用于在 HIC 存储器映射寄存器的 256 个字节中进行选择。

写入或读取 HIC 内部区域，需要：

通过将 0xA 写入寄存器 RWGCR.RWEN 来使能 HIC；使用 RWMODECR.RW_MODE、RWMODECR.BEN_PRESENT、

RWMODECR.RDY_PRESENT 和 RWPINPOLCR 寄存器配置 HIC 引脚的可用性和极性。

Host 向本设备写数据：Host 写入 H2D buffer（即 H2D_BUF 寄存器）。（ASRAM 协议传输）完成写入后，Host 通过向 H2DTOKEN 寄存器写入任何值以触发设备中断（H2DINT）。中断使得设备从 H2D buffer（H2D_BUF 寄存器）中读取数据。

Host 从本设备中读数据：Device 写入 D2H buffer（即 D2H_BUF 寄存器）。（CPU、DMA 等主设备），完成写入后，Device 通过向 D2HTOKEN 寄存器写入任何值以触发 Host 中断（D2HINT）。中断使得 Host 对 D2H buffer（D2H_BUF 寄存器）的内容发起读操作。

注：如果对 D2H_BUF 和 H2D_BUF 的写入超过存储器区域（128 字节），是不会生成错误警报的，所以应用程序需要确保写入容量不超存储范围。硬件会对不超过存储区域的数据进行写入操作，超出部分不进行响应和处理。

### 11.3.2 直接访问模式

在直接访问模式下，Host 通过 HIC 可以直接读取和写入内存映射中指定的设备位置。要读取的设备位置是 8 位地址线 A[7:0]和 24 位寄存器 HICDBADDR {#}的组合，这两个地址段连接为：

全 32 位地址={HICDBADDR {#}.BASE_ADDR[31:8],A[7:0]}

有 8 个 HICDBADDR {#}寄存器(2^3)A 与 A[7:0]地址线(2^8)组合总共可表示 2048(2^11)个地址位置。可通过 BASESEL 寄存器或 BASESEL[2:0]引脚选择将哪个 BASEADDR {#}寄存器与 A[7:0]级联（通过引脚选择可使得传输更快，因为不需要每次传输都要先配置 BASESEL 寄存器）。图中显示了该模式所需的最小外部连接。

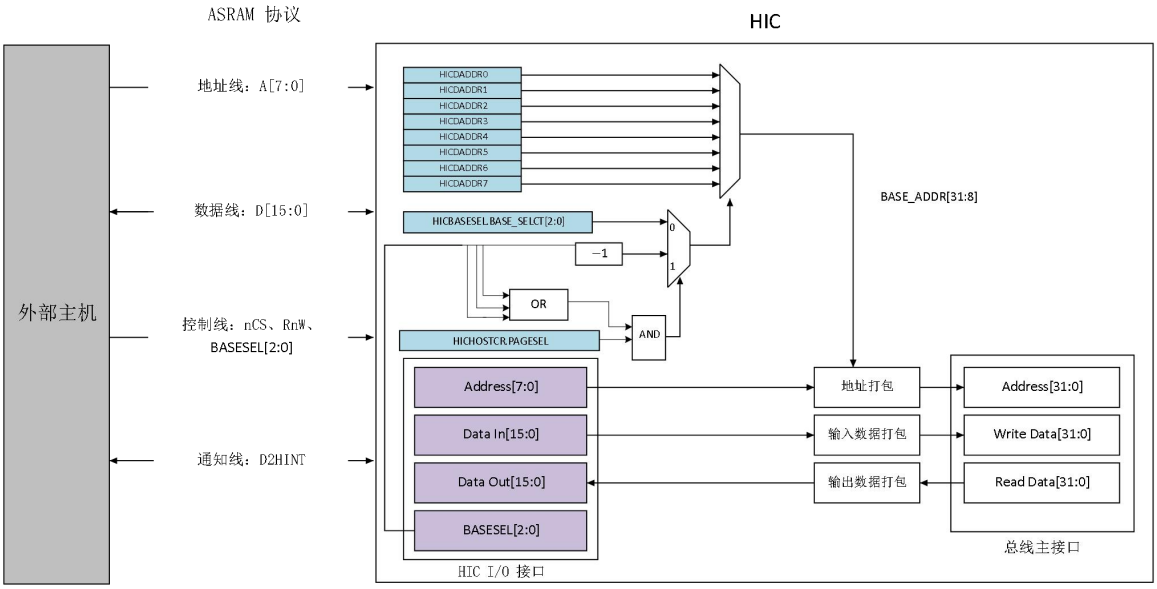



图 11-7 HIC 直接访问模式示意图

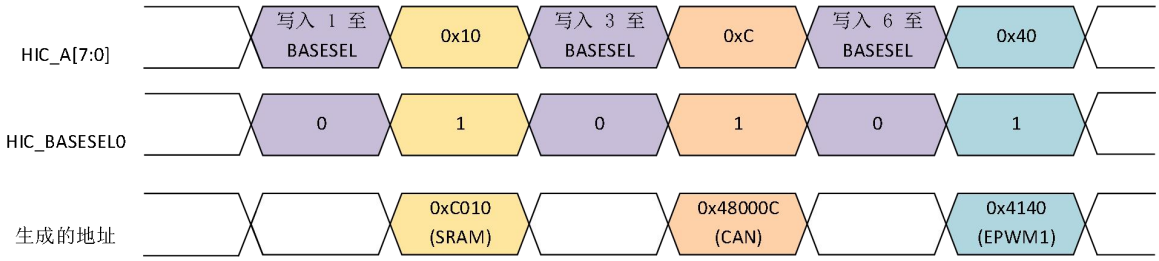



图 11-8 基于寄存器 BASESEL 选择的跨区域访问

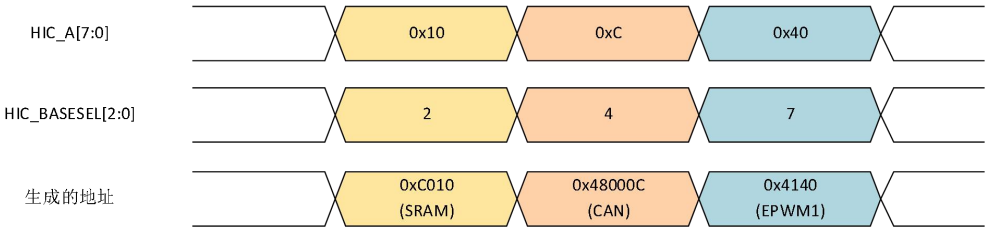



图 11-9 基于 I/O BASESEL 选择的跨区域访问

### 11.3.2.1 直接访问模式操作

将 HIC 设置为直接访问模式，BASESEL[2:0]引脚值需要大于 000。地址线 A[7:0]与寄存器 DBADDR {#}配合使用，用于读取或写入设备中的某个位置。下表显示了外部主机的设备地址空间视图。

表 11-5 外部主机的地址空间视图

| BASESEL[2:0]引脚 | HICBASESEL.BASE_SELECT[2:0]寄存器 | HICHOSTCR.PAGES寄存器 | 地址空间 | 内容 | 访问模式 |
|---|---|---|---|---|---|
| 000 | X | x | HIC 内存映射寄存器 - 256 字节 | HIC 控制，状态，D2H_BUF和H2D_BUF 寄存器 | 邮箱模式 |
| 001 | X | 1 | {HICDBADDR0.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| 010 | X | 1 | {HICDBADDR1.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| 011 | X | 1 | {HICDBADDR2.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| 100 | X | 1 | {HICDBADDR3.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| 101 | X | 1 | {HICDBADDR4.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| 110 | X | 1 | {HICDBADDR5.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| 111 | X | 1 | {HICDBADDR6.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| x (not 000) | 000 | 0 | {HICDBADDR0.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| x (not 000) | 001 | 0 | {HICDBADDR1.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| x (not 000) | 010 | 0 | {HICDBADDR2.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| x (not 000) | 011 | 0 | {HICDBADDR3.BASE_ADDR[31:8], | 查看HIC内存映射 | 直接模式 |

| BASESEL[2:0]引脚 | HICBASESEL.BASE_SELECT[2:0]寄存器 | HICHOSTCR.PAGES寄存器 | 地址空间 | 内容 | 访问模式 |
|---|---|---|---|---|---|
| | | | A[7:0]} - 256 Locations | | |
| x (not 000) | 100 | 0 | {HICDBADDR4.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| x (not 000) | 101 | 0 | {HICDBADDR5.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| x (not 000) | 110 | 0 | {HICDBADDR6.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| x (not 000) | 111 | 0 | {HICDBADDR7.BASE_ADDR[31:8], A[7:0]} - 256 Locations | 查看HIC内存映射 | 直接模式 |
| 000 | xxx | 0 | HIC 内存映射寄存器 - 256 字节 | HIC 控制，状态，D2H_BUF 和 H2D_BUF 寄存器 | 邮箱模式 |
| (1) "x" 表示不关心 | | | | | |

注：表中 Address Space 一栏中的“-256 Locations”表示可访问 256 个地址空间；

HICDBADDR7 不能被 BASESEL[2:0]pin 选中(000 已经被占据，只有剩下 7 种组合)；只有使用 BASESEL.BASE_SELECT[2:0]进行配置时能选择到 HICDBADDR7。

#### 11.3.2.2 直接访问模式读写流程

Device 通过将 0xA 写入寄存器 HICGCR.HICEN 来使能 HIC；Device 使用 HICMODECR.EN_DEVACC 寄存器启用直接访问模式访问地址权限（可访问设备内部区域）；Device 使用 RIMODCR.RW_MODE、RIMODCR.BEN_PRESENT、RIMODCR.RDY_PRESENT 和 RIMPINPOLCR 寄存器配置 HIC 引脚的有效性和极性；Device 配置 HICDBADDR {#} 寄存器；host 写入数据到 Device，Host 使用 BASESEL[2:0]引脚选择一个 HICDBADDR {#}，与地址线 A[7:0]组合形成访问地址将写数据写入；

要从设备读取数据，外部主机使用 BASESEL[2:0]引脚选择一个 ADRISDBADDR{#}，与地址线 A[7:0]组合形成访问地址将数据读出。

### 11.3.3 读写相关控制

HIC 有两种控制读写的模式：单引脚模式和双引脚模式。

#### 11.3.3.1 单引脚读写模式

在单引脚读/写访问中，nOE（RnW）确定正在执行的事务是读还是写。如果 nOE 为低电平，则将数据从主机写入设备；如果 nOE 为高电平，则由主机从设备中读取数据。

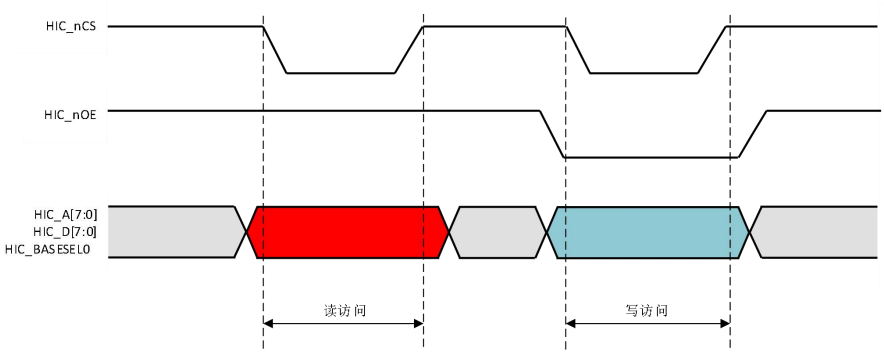



图 11-10 单引脚模式下的读/写访问

#### 11.3.3.2 双引脚读写模式

在双引脚读/写访问中，nOE 和 nWE 引脚确定正在执行的事务是读还是写。在默认极性状态下，nOE 变为低电平表示从设备向主机读取数据，nWE 变为低电平表示主机向设备写入数据。

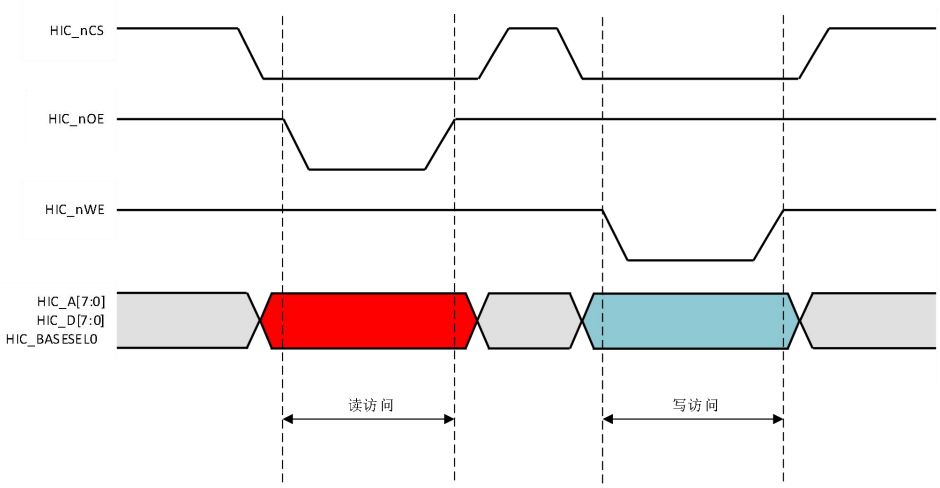



图 11-11 双引脚模式下的读/写访问

### 11.3.4 数据打包和解包

HIC 与主机数据传输可以配置为 8 位和 16 位模式，使用 HIC MODECR.DW_MODE 寄存器选择主机数据传输模式。

如果 host 写操作访问对象为 HIC MMR，则写数据有效字节排列无限制(存在 strb 信号表示哪个字节有效)。

如果 host 写操作访问对象为 GS0/外设，则 HIC 侧写数据的有效字节必须是连续的（不能出现 B3xxmB1xxn 跨字节写入的情况）；如果不连续，有效字节间数据(xxm)会以 0 值代替。

在 HIC 启动器端口中，访问位宽限制为 16 位和 32 位模式，可使用寄存器 HICHOSTCR.ACCSIZE 进行选择。如果主机访问端与启动器端访问位宽不匹配，HIC 将根据具体配置在内部对数据进行自动打包和解包。

注：如果 initiator(device)侧配置为 16bit，则 host 每次读请求均会由 HIC 发出；如果 initiator(device)侧配置为 32bit，HIC 会根据当前数据缓存是否可复用来决定读请求是否发出。

例如，如果主机访问模式使用 nBE 引脚或 HICMODECR.DW_MODE 寄存器设置为 16 位，而启动器端口访问模式使用 HOSTHOSTCR.ACCSIZE 寄存器设置为 16 位，则不会进行数据打包或解包。如果主机访问模式设置为 8 位，启动器端口访问模式设

置为 16 位，若先接收到位于低地址的 8 位数据，则将等待另一个 8 位数据，最后将两个 8 位数据打包为一个 16 位数据位宽的访问请求。

写操作的数据打包和解包：

host 配置为 8bit：

表 11-6 host 为 8bit 模式下的写 MMR 情况

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 访问MMR情况 | HICMODECR.DW_MODE | 主机端口写 | | | 16 位 | | 32 位 | |
| | | A[7:0] | 8 位 DW_MODE：D[7:0]；16 位 DW_MODE：D[15:0] | nBE[1:0]（低有效） | DWAB[31:0]（实际从HIC-initialtor发出去的请求的地址）(HICDBADDR.BASE_ADDR[31:8],A[7:0]) | DWDB[31:0]（数据） | DWAB[31:0] | DWDB[31:0] |
| | 8bit | 0x0 | B0 | x | | | | |
| | | 0x1 | B1 | x | 0x0(双有效字节以最低有效字节位置为地址) | [15:0] = B1B0 | | |
| | | 0x2 | B2 | x | | | | |
| | | 0x3 | B3 | x | 0x2 | [31:16] = B3B2 | 0x0 | [31:0] = B3B2B1B0 |
| | | | | | | | | |
| | | 0x1 | B1 | | 0x1 | [15:0] = B1XX | | |
| | | | | | | | | |
| | | 0x3 | B3 | | 0x3 | [31:16] = B3XX | 0x3 | [31:0] = B3XXB1XX |
| | | | | | | | | |
| | | 0xA | | | | | | |
| | | 0xA+n | | | （n>1）<br>ERROR：数据超出16bit缓存 | | （n>3）<br>ERROR：数据超出32bit缓存 | |

表 11-7 host 为 8bit 模式下的写 GS0/外设情况

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 访问GS0/外设情况 | HICMODECR.DW_MODE | 主机端口写 | | | 16 位 | | 32 位 | |
| | | A[7:0] | D[7:0] for 8-bit DW_MODED[15:0] for 16-bit DW_MODE | nBE[1:0]（低有效） | DWAB[31:0]（实际从HIC-initialtor发出去的请求的地址）(HICDBADDR.BASE_ADDR[31:8],A[7:0]) | DWDB[31:0]（数据） | DWAB[31:0] | DWDB[31:0] |
| | 8bit | 0x0 | B0 | x | | | | |
| | | 0x1 | B1 | x | 0x0(双有效字节以最低有效字节位置为地址) | [15:0] = B1B0 | | |
| | | 0x2 | B2 | x | | | | |
| | | 0x3 | B3 | x | 0x2 | [31:16] = B3B2 | 0x0 | [31:0] = B3B2B1B0 |
| | | | | | | | | |
| | | 0x1 | B1 | | 0x1 | [15:0] = B100 | | |
| | | | | | | | | |
| | | 0x3 | B3 | | 0x3 | [31:16] = B300 | 0x3 | [31:0] = B300B1XX |
| | | | | | | | | |
| | | 0xA | | | | | | |
| | | 0xA+n | | | （n>1）ERROR：数据超出16bit缓存 | | （n>3）ERROR：数据超出32bit缓存 | |

发生错误则请求缓存清空：

8-主机-16-启动器：舍弃前一个 byte

8-主机-32-启动器：舍弃之前(32bit)所有有效字节

16-主机-16-启动器：舍弃前一个 byte

16-主机-32-启动器：舍弃之前(32bit)所有有效字节

错误例子：

Host：8bit；hic：32bit

操作 1：host 向地址 0 写 B1

操作 2：host 向地址 5 写 B5

会报错（操作 1 结束之后，处于数据等待状态，但是操作 2 直接从 0-3 的组跳到 4-7 的组，就会报错）

host 配置为 16bit：

表 11-8 host 为 16bit 模式下的写 MMR 情况

| | | 主机端口写 | | | 16 位 | | 32 位 | |
|---|---|---|---|---|---|---|---|---|
| | HICMODECR.DW_MODE | A[7:0] | 8 位 DW_MODED[7:0]；16 位 DW_MODE：D[15:0] | nBE[1:0](低有效 | DWAB[31:0]（实际从HIC-initialtor发出去的请求的地址）(HICDBADDR.BASE_ADDR[31:9],A[7:0],{0}) | DWDB[31:0](数据) | DWAB[31:0] | DWDB[31:0] |
| 访问MMR情况 | 16bit | 0x0 | B1B0 | 2'b00 | 0x0<br>(双有效字节以最低有效字节位置为地址) | [15:0] = B1B0 | | |
| | | 0x1 | B3B2 | 2'b00 | 0x2 | [31:16] = B3B2 | 0x2 | [31:0] = B3B2B1B0 |
| | | 0x2 | B5B4 | 2'b00 | 0x4 | [15:0] = B5B4 | | |
| | | 0x3 | B7B6 | 2'b00 | 0x6 | [31:16] = B7B6 | 0x4 | [31:0] = B7B6B5B4 |
| | | | | | | | | |
| | | 0x0 | XXB0 | 2'b10 | | | | |
| | | 0x0 | B1XX | 2'b01 | 0x0 | [15:0] = B1B0 | | |
| | | 0x1 | XXB2 | 2'b10 | | | | |
| | | 0x1 | B3XX | 2'b01 | 0x2 | [31:16] = B3B2 | BADDRx+0x3 | [31:0] = B3B2B1B0 |
| | | | | | | | | |
| | | 0x0 | B1XX | 2'b01 | 0x1 | [15:0] = B1XX | | |
| | | | | | | | | |
| | | 0x1 | B3XX | 2'b01 | 0x3 | [31:16] = B3XX | BADDRx+0x3 | [31:0] = B3XXB1XX |
| | | | | | | | | |
| | | 0xA | XXB0 | 2'b10 | | | | |
| | | 0xA+n（n>1） | B1XX | 2'b01 | （n>1）ERROR：数据超出16bit缓存 | | （n>3）ERROR：数据超出32bit缓存 | |

表 11-9 host 为 16bit 模式下的写 GS0/外设情况

| | | 主机端口写 | | | 16 位 | | 32 位 | |
|---|---|---|---|---|---|---|---|---|
| | HICMODECR.DW_MODE | A[7:0] | D[7:0] for 8-bit DW_MODE<br>D[15:0] for 16-bit DW_MODE | nBE[1:0] (低有效) | DWAB[31:0] (实际从HIC-initialtor发出去的请求的地址) (HICDBADDR.BASE_ADDR[31:9],A[7:0],{0}) | DWDB[31:0] (数据) | DWAB[31:0] | DWDB[31:0] |
| 访问GS0/外设情况 | 16bit | 0x0 | B1B0 | 2'b00 | 0x0<br>(双有效字节以最低有效字节位置为地址) | [15:0] = B1B0 | | |
| | | 0x1 | B3B2 | 2'b00 | 0x2 | [31:16] = B3B2 | 0x2 | [31:0] = B3B2B1B0 |
| | | 0x2 | B5B4 | 2'b00 | 0x4 | [15:0] = B5B4 | | |
| | | 0x3 | B7B6 | 2'b00 | 0x6 | [31:16] = B7B6 | 0x4 | [31:0] = B7B6B5B4 |
| | | | | | | | | |
| | | 0x0 | XXB0 | 2'b10 | | | | |
| | | 0x0 | B1XX | 2'b01 | 0x0 | [15:0] = B1B0 | | |
| | | 0x1 | XXB2 | 2'b10 | | | | |
| | | 0x1 | B3XX | 2'b01 | 0x2 | [31:16] = B3B2 | BADDRx+0x3 | [31:0] = B3B2B1B0 |
| | | | | | | | | |
| | | 0x0 | B1XX | 2'b01 | 0x1 | [15:0] = B1XX | | |
| | | | | | | | | |
| | | 0x1 | B3XX | 2'b01 | 0x3 | [31:16] = B3XX | BADDRx+0x3 | [31:0] = B300B1XX |
| | | | | | | | | |
| | | 0xA | XXB0 | 2'b10 | | | | |
| | | 0xA+n（n>1） | B1XX | 2'b01 | （n>1）<br>ERROR：数据超出16bit缓存 | | （n>3）<br>ERROR：数据超出32bit缓存 | |

读操作的数据打包和解包：

host 配置为 8bit：

表 11-10 host 为 8bit 模式下的读操作

| HICMODECRDW_MODE | 主机端口读 | | 启动器端口读 | | 内容 |
|---|---|---|---|---|---|
| | A[7:0] | 返回主机的RDATA | DRAB[31:0](HICDBADDR.BASE_ADDR[31:8],A[7:0]) | DRDB[31:0] | |
| 8-bit（0） | 0x0 | B0 | 0x0 | [31:0] = B3B2B1B0 | 从主机进行随机读取会触发启动器端口上的32位读取 |
| | 0x1 | B1 | | | 在对齐的32位字内读取的顺序字节从内部数据缓冲区返回 |
| | 0x2 | B2 | | | |
| | 0x3 | B3 | | | |
| | 0x4 | B4 | 0x4 | [31:0] = B7B6B5B4 | 在对齐的32位字边界之外读取的字节触发启动器端口上的32位读取。 |
| | | | | | |
| | 0x1 | B1 | 0x1 | [31:0] = B3B2B1B0 | 从主机进行随机读取会触发启动器端口上的32位读取 |
| | | | | | |
| | 0x3 | B3 | | | 在对齐的32位字内读取的顺序字节从内部数据缓冲区返回 |
| | 0x4 | B4 | 0x4 | [31:0] = B7B6B5B4 | 在对齐的32位字边界之外读取的字节触发启动器端口上的32位读取。 |
| | 0x5 | B5 | | | 在对齐的32位字内读取的顺序字节从内部数据缓冲区返回 |

host 配置为 16bit：

表 11-11 host 为 16bit 模式下的读操作

| HICMODECR.DW_MODE | 主机端口读 | | 初始器端口读 | | 内容 |
|---|---|---|---|---|---|
| | A[7:0] | return host RDATA | DRAB[31:0](HICDBADDR.BASE_ADDR[31:8],A[7:0]) | DRDB[31:0] | |
| 16-bit（1） | 0x0 | B1B0 | 0x0 | [31:0] = B3B2B1B0 | 从主机随机读取会触发启动器端口上的32位读取 |

| HICMODECR.DW_MODE | 主机端口读 | | 初始器端口读 | | 内容 |
|---|---|---|---|---|---|
| | A[7:0] | return host RDATA | DRAB[31:0](HICDBADDR.BASE_ADDR[31:8],A[7:0]) | DRDB[31:0] | |
| | 0x1 | B3B2 | | | 从内部数据缓冲区返回对齐的32位字内的连续16位读取。 |
| | 0x2 | B5B4 | 0x4 | [31:0] = B7B6B5B4 | 从主机随机读取会触发启动器端口上的32位读取 |
| | 0x3 | B7B6 | | | 从内部数据缓冲区返回对齐的32位字内的连续16位读取。 |
| | | | | | |
| | 0x1 | B3B2 | 0x2 | [31:0] = B3B2B1B0 | 从主机随机读取会触发启动器端口上的32位读取 |
| | 0x2 | B5B4 | 0x4 | [31:0] = B7B6B5B4 | 从主机随机读取会触发启动器端口上的32位读取 |
| | 0x3 | B7B6 | | | 从内部数据缓冲区返回对齐的32位字内的连续16位读取。 |

### 11.3.5 地址传输

由于启动器端口使用8位对齐寻址，而主机端口可以在8位和16位模式下工作，因此需要进行地址转换才能正确工作。下表显示了在8位数据位宽和16位数据宽度模式下执行的地址转换示例。

表 11-12 比特数据宽度模式下的地址转换

| HICMODECR.DW_MODE | 主机端口地址 | 启动器端口地址 (HICDBADDR.BASE_ADDR[31:8], A[7:0]) |
|---|---|---|
| 8-bit ("0") | 0x0 | 0x0 |
| | 0x1 | 0x1 |
| | 0x2 | 0x2 |
| | 0x80 | 0x80 |
| | 0xFE | 0xFE |

| HICMODECR.DW_MODE | 主机端口地址 | 启动器端口地址<br>(HICDBADDR.BASE_ADDR[31:8], A[7:0]) |
|---|---|---|
| | 0xFF | 0xFF |

表 11-13 16 比特数据宽度模式下的地址转换

| HICMODECR.DW_MODE | 主机端口地址 | 启动器端口地址<br>(HICDBADDR.BASE_ADDR[31:9], A[7:0], {0}) |
|---|---|---|
| 16-bit<br>("1") | 0x0 | 0x0 |
| | 0x1 | 0x2 |
| | 0x2 | 0x4 |
| | 0x4 | 0x8 |
| | 0x80 | 0x100 |
| | 0xFE | 0x1FC |
| | 0xFF | 0x1FE |

注：如果要访问的地址中比特位 8（addr[8]）的值为 1（如 0x6666_6F22），此时如果 host 访问位宽配置为 8bit，则所选的基地址寄存器需要配置为 0x6666_6F00，A[7:0]设置为 0x22；如果 host 访问位宽配置为 16bit，则基地址寄存器需要配置为设置为 0x6666_6E00， A[7:0]设置为 0x91（最终总偏移量为 0x122）

总结：host 为 8bit，使用 BASE_ADDR[31:8]，offset 值即 A[7:0]；

host 为 16bit，使用 BASE_ADDR[31:9]，offset 值为:{A[7:0],1'b0}；

### 11.3.6 错误传输类型

HIC 具有寄存器 HICH2DINTFLG、HICD2HINTFLG、HICERRADDR 和 HICACCVIOADDR 用于在发生访问错误时捕获 HIC 的状态。捕获的错误包括：

#### 11.3.6.1 非法写

如果为写访问指定了不正确的地址范围，也会发生此错误。

#### 11.3.6.2 非法读

如果从上一个事务中未完全写入的位置进行读取，也会发生此错误。

例如，HIC 正在等待另一个字节以完成 16 位写事务，但在第二个字节之前获得读访问。

#### 11.3.6.3 总线错误

当主机和设备同时写入内存位置时，会发生此错误。主机数据将被丢弃，而设备数据将被保留。

#### 11.3.6.4 启动器端口访问冲突错误（ACCVIO）

当主机尝试访问 SOC 上非外设帧时，resp 返回 1 触发。

#### 11.3.6.5 HICERRADDR 寄存器行为

D2H_BASE_SEL：当捕获到 D2H_ERR_ADDR 时，BASE_SEL 引脚的值；

当第一次捕获到错误时，该值会被冻结，直到 HICD2HINTFLG 寄存器的 BUSERR_FLG，ILLWR_FLG，和 ILLRD_FLG 标志位被用户清空；

当发生一次错误事件，HIC_BASESEL 引脚的值将被捕获，存放到该寄存器位域中，同时在 HICD2HINTFLG 寄存器中设置对应的错误标志；

对于 PAGESEL=0 的情况，用户需要读 BASE_SELECT 寄存器，来找出正确的 BASE_SEL 值

D2H_ERR_ADDR：捕获的 HOST 地址，当为 HOST 捕获的错误事务（作为 D2HINT 中断的响应）

当发生 BUSERR，ILLWD，ILLRD 错误时，HOST 地址将被捕获，存放到该寄存器位域中，同时在 HICD2HINTFLG 寄存器中设置对应的错误标志。该值会被冻结，直到 HICD2HINTFLG 寄存器的 BUSERR_FLG，ILWD_FLG，和 ILRD_FLG 标志位被用户清空

H2D_BASE_SEL：当捕获到 H2D_ERR_ADDR 时，BASE_SEL 引脚的值；

当第一次捕获到错误时，该值会被冻结，直到 HICH2DINTFLG 寄存器的 BUSERR_FLG，ILLWR_FLG，和 ILLRD_FLG 标志位被用户清空；

当发生一次错误事件，HIC_BASESEL 引脚的值将被捕获，存放到该寄存器位域中，同时在 HICH2DINTFLG 寄存器中设置对应的错误标志；

对于 PAGESEL=0 的情况，用户需要读 BASE_SELECT 寄存器，来找出正确的 BASE_SEL 值

H2D_ERR_ADDR：捕获的 HOST 地址，当为 DEVICE 捕获的错误事务（作为 H2DINT 中断的响应）

当发生 BUSERR，ILLWD，ILLRD 错误时，HOST 地址将被捕获，存放到该寄存器位域中，同时在 HICH2DINTFLG 寄存器中设置对应的错误标志。该值会被冻结，直到 HICH2DINTFLG 寄存器的 BUSERR_FLG，ILWD_FLG，和 ILRD_FLG 标志位被用户清空

### 11.3.7 安全

在设备和 HIC 模块中实现了多个安全级别，以防止未经授权的访问：

级别 1：在对设备区域进行任何访问之前，设备必须打开 HIC 的时钟，并使用 HICGCR.HICEN 启动 HIC。主机对内部配置寄存器为只读权限。

级别 2：可使用锁定寄存器锁定配置寄存器，防止进一步写入。此寄存器对主机为只读。

EALLOW 相关

1.增加 EALLOW 配置寄存器：配置为 1 是开启 HIC 的 EALLOW 访问

2.EALLOW 寄存器受 LOCK COMMIT 保护

增加 COMMIT

1) lock 寄存器是可以把配置寄存器的信息锁住，就不会被无意修改
2) commit 是把 lock 寄存器锁住，这样 lock 也不会被无意修改，然后 commit 寄存器修改一次后，就只能通过复位清零来修改值（注：HICCOMMIT 寄存器的偏移量为 0x74）

### 11.3.8 减少引脚数量的使用场景

HIC 不需要所有引脚工作，下面描述了减少引脚集的用例。

#### 11.3.8.1 读写控制引脚

仅使用一个引脚 nOE/RnW 引脚即可实现读/写控制，而不是传统的两个引脚：nOE 和 nWE 引脚。通过配置 HICMODECR.RW_MODE 寄存器实现。

#### 11.3.8.2 BASESEL Pins

如果 BASESEL 引脚未使用，HIC 将被限制为邮箱访问模式，在该模式下，外部主机只能访问 HIC 内存映射寄存器。然而，通过直接访问模式访问设备寄存器，只需要一个 BASESEL 引脚。

在此配置中，BASESEL0 引脚确定正在进行的访问是邮箱访问还是直接访问。BASESEL0 设置为 0 表示邮箱访问; BASESEL0 设置为 1 表示直接访问。此配置将基址选择限制为只能选择 DBADDR0。通过配置寄存器 HICHOSTCR.PAGESEL 实现。

## 11.4 寄存器描述

### 11.4.1 HIC 基地址

表 11-14 HIC 基地址表

| 比特位域名 | | DriverLib名 | 基地址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| 实例 | 结构 | | | | | |
| HicRegs | HIC_CFG_REGS | HIC_BASE | 0x0062_2000 | YSE | YES | YES |

### 11.4.2 HIC_CFG_REGS 寄存器

表 11-15 HIC_CFG_REGS 寄存器

| 偏移量 | 寄存器缩略名 | 寄存器名 |
|---|---|---|
| 0h | HICREV | 模块修订寄存器 |
| 4h | HICGCR | 全局控制寄存器 |
| 8h | HICLOCK | 锁寄存器 |
| Ch | HICMODECR | 模式控制寄存器 |
| 10h | HICPINPOLCR | 引脚极性控制寄存器 |
| 14h | HICBASESEL | 基地址选择寄存器 |

| 偏移量 | 寄存器缩略名 | 寄存器名 |
| --- | --- | --- |
| 18h | HICHOSTCR | 主机控制寄存器 |
| 1Ch | HICERRADDR | 主机错误地址寄存器 |
| 20h | HICH2DTOKEN | 主机到设备令牌寄存器 |
| 24h | HICD2HTOKEN | 设备到主机令牌寄存器 |
| 28h | HICDBADDR0 | 设备基地址寄存器0 |
| 2Ch | HICDBADDR1 | 设备基地址寄存器1 |
| 30h | HICDBADDR2 | 设备基地址寄存器2 |
| 34h | HICDBADDR3 | 设备基地址寄存器3 |
| 38h | HICDBADDR4 | 设备基地址寄存器4 |
| 3Ch | HICDBADDR5 | 设备基地址寄存器5 |
| 40h | HICDBADDR6 | 设备基地址寄存器6 |
| 44h | HICDBADDR7 | 设备基地址寄存器7 |
| 50h | HICH2DINTEN | H2D中断使能寄存器 |
| 54h | HICH2DINTFLG | H2D中断状态标志寄存器 |
| 58h | HICH2DINTCLR | H2D中断状态清除寄存器 |
| 5Ch | HICH2DINTFRC | H2D中断强制设置寄存器 |
| 60h | HICD2HINTEN | D2H中断使能寄存器 |
| 64h | HICD2HINTFLG | D2H中断状态标志寄存器 |
| 68h | HICD2HINTCLR | D2H中断状态清除寄存器 |
| 6Ch | HICD2HINTFRC | D2H中断强制设置寄存器 |
| 70h | HICACCVIOADDR | 非法访问地址寄存器 |
| 74h | HICCOMMIT | COMMIT寄存器 |
| 80h | H2D_BUF0 | 主机到设备缓冲区0 |
| 84h | H2D_BUF1 | 主机到设备缓冲区1 |
| 88h | H2D_BUF2 | 主机到设备缓冲区2 |
| 8Ch | H2D_BUF3 | 主机到设备缓冲区3 |
| 90h | H2D_BUF4 | 主机到设备缓冲区4 |
| 94h | H2D_BUF5 | 主机到设备缓冲区5 |
| 98h | H2D_BUF6 | 主机到设备缓冲区6 |
| 9Ch | H2D_BUF7 | 主机到设备缓冲区7 |
| A0h | H2D_BUF8 | 主机到设备缓冲区8 |
| A4h | H2D_BUF9 | 主机到设备缓冲区9 |
| A8h | H2D_BUF10 | 主机到设备缓冲区10 |
| ACh | H2D_BUF11 | 主机到设备缓冲区11 |
| B0h | H2D_BUF12 | 主机到设备缓冲区12 |
| B4h | H2D_BUF13 | 主机到设备缓冲区13 |
| B8h | H2D_BUF14 | 主机到设备缓冲区14 |
| BCh | H2D_BUF15 | 主机到设备缓冲区15 |
| C0h | D2H_BUF0 | 设备到主机缓冲区0 |
| C4h | D2H_BUF1 | 设备到主机缓冲区1 |
| C8h | D2H_BUF2 | 设备到主机缓冲区2 |
| CCh | D2H_BUF3 | 设备到主机缓冲区3 |
| D0h | D2H_BUF4 | 设备到主机缓冲区4 |

| 偏移量 | 寄存器缩略名 | 寄存器名 |
|---|---|---|
| D4h | D2H_BUF5 | 设备到主机缓冲区5 |
| D8h | D2H_BUF6 | 设备到主机缓冲区6 |
| DCh | D2H_BUF7 | 设备到主机缓冲区7 |
| E0h | D2H_BUF8 | 设备到主机缓冲区8 |
| E4h | D2H_BUF9 | 设备到主机缓冲区9 |
| E8h | D2H_BUF10 | 设备到主机缓冲区10 |
| ECh | D2H_BUF11 | 设备到主机缓冲区11 |
| F0h | D2H_BUF12 | 设备到主机缓冲区12 |
| F4h | D2H_BUF13 | 设备到主机缓冲区13 |
| F8h | D2H_BUF14 | 设备到主机缓冲区14 |
| FCh | D2H_BUF15 | 设备到主机缓冲区15 |

表 11-16 HIC_CFG_REGS 访问类型代码

| 访问类型 | 标志代码 | 描述 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回0 |
| 写类型 | | |
| W | W | 写 |
| W1C | W<br>1C | 写<br>1用于清除 |
| W1S | W<br>1S | 写<br>1用于设置 |
| 复位或默认值 | | |
| *-n* | | 复位后的值或默认值 |

### 11.4.2.1 HICREV 寄存器

偏移地址：0h　复位值：40000000h

模块修订版本寄存器。

表 11-17 HICREV 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SCHEME | | 保留 | | FUNC | | | |
| R-1h | | R-0-0h | | R-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FUNC | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RTL | | | | | MAJOR | | |
| R-0h | | | | | R-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CUSTOM | | MINOR | | | | | |
| R-0h | | R-0h | | | | | |

表 11-18 HICREV 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-30 | SCHEME | R | 1h | 该位域确定了HIC模块的方案，返回01<br>对主机只读<br>复位类型：SYSRSn |
| 29-28 | RESERVED | R-0 | 0h | 保留位 |
| 27-16 | FUNC | R | 0h | 功能发行编号<br>反映软件兼容性。如果没有软件兼容性级别，则为兼容模块分配一个唯一的功能编号，保持相同编号<br>对主机只读<br>复位类型：SYSRSn |
| 15-11 | RTL | R | 0h | 设计发行编号<br>由于规范变更或发布后设计变更而增加的版本。当MAJOR或MINOR递增时复位为0。<br>对主机只读<br>复位类型：SYSRSn |
| 10-8 | MAJOR | R | 0h | 主要修订编号<br>代表模块的重大变更（例如，添加 / 更改了全新的功能）。此模块的主修订版本号<br>对主机只读<br>复位类型：SYSRSn |
| 7-6 | CUSTOM | R | 0h | 自定义模块编号<br>表示模块的特殊版本。标准软件可能不支持该版本。<br>对主机只读<br>复位类型：SYSRSn |
| 5-0 | MINOR | R | 0h | 次要修订编号 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 代表对模块所做的小改动（例如，增强现有功能）。此模块的次要修订版本号 |

### 11.4.2.2 HICGCR 寄存器

偏移地址：4h　复位值：00000000h

全局控制寄存器。

表 11-19 HICGCR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | HICEN | | | |
| R-0-0h | | | | R/W-0h | | | |

表 11-20 HICGCR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留位 |
| 3-0 | HICEN | R/W | 0h | 主机接口使能<br>该位域控制主机接口的操作<br>0xA：主机接口启用。主机对MBOX和设备区域的访问已启用。<br>其他：主机接口被禁用。主机对MBOX和设备区域的访问被禁用。主机仍然可以读取所有MMR，并对HIC模块的可写入MMR进行写入。<br>对主机只读<br>复位类型：SYSRSn |

### 11.4.2.3 HICLOCK 寄存器

偏移地址：8h　复位值：00000000h

锁定寄存器。

表 11-21 HICLOCK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| WRITE_ENABLE_KEY | | | | | | | |
| W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| WRITE_ENABLE_KEY | | | | | | | |
| W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | LOCK |
| R-0-0h | | | | | | | R/W-0h |

表 11-22 HICLOCK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | WRITE_ENABLE_KEY | W | 0h | 这16位作为key，用于启用对这个寄存器的第0位的写入。<br>只有通过单个32位写入操作，且第31到16位等于0x5a5a时，才能向第0位写入“1”。所有其他写入操作都会被忽略，包括单独的16位写入操作。<br>对于此位域，读值始终返回0。<br>对主机只读<br>复位类型：SYSRSn |
| 15-1 | RESERVED | R-0 | 0h | 保留位 |

### 11.4.2.4 HICMODECR 寄存器

偏移地址：Ch　复位值：00000000h

模式控制寄存器。

表 11-23 HICMODECR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | EN_HOSTWREALLOW | EN_DEVACC | D2HBUF_HOSTWREN | H2DBUF_DEVWREN |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | RDY_PRESENT | BEN_PRESENT | RW_MODE | 保留 | | DW_MODE | |
| R-0-0h | R/W-0h | R/W-0h | R/W-0h | R-0-0h | | R/W-0h | |

表 11-24 HICMODECR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-12 | RESERVED | R-0 | 0h | 保留位 |
| 11 | EN_HOSTWREALLOW | R/W | 0h | 允许主机写入HOSTCR.EALLOW寄存器位域<br>0：主机写入HOSTCR.EALLOW寄存器位域的操作将被忽略<br>1：主机可以写入并重新修改HOSTCR.EALLOW寄存器位域<br>对主机只读<br>复位类型：SYSRSn |
| 10 | EN_DEVACC | R/W | 0h | 允许主机访问设备区域<br>0：主机不能访问设备区域 - 只有HIC内部MMR和MBOX区域可访问<br>1：主机可以访问HIC内部地址空间和被BASEx+寄存器选中的设备地址空间<br>对主机只读<br>复位类型：SYSRSn |
| 9 | D2HBUF_HOSTWREN | R/W | 0h | 对主机的D2H缓冲区写使能<br>0：D2H缓冲区只能被设备写入<br>1：D2H缓冲区也可以被主机写入<br>对于用户想要更大D2H缓冲区区域（且对H2D缓冲区区域的需求有限或没有需求）的情况，此功能很有帮助。<br>对主机只读<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 8 | H2DBUF_DEVWREN | R/W | 0h | 对设备的H2D缓冲区写使能<br>0：H2D缓冲区只能被主机写入<br>1：H2D缓冲区也可以被设备写入<br>对于用户需要更大H2D缓冲区区域（且对D2H缓冲区区域的需求有限或没有）的情况，此功能很有帮助。<br>对主机只读<br>复位类型：SYSRSn |
| 7 | RESERVED | R-0 | 0h | 保留位 |
| 6 | RDY_PRESENT | R/W | 0h | ready引脚存在<br>定义了nRDY引脚是否存在<br>0：'nRDY'引脚不存在。主机必须确保在数据手册要求的周期内保持控制引脚处于激活状态。<br>1：'nRDY'引脚存在。只要nRDY信号有效，主机必须保持控制/数据信号<br>对主机只读<br>复位类型：SYSRSn |
| 5 | BEN_PRESENT | R/W | 0h | 字节使能引脚存在<br>定义了字节使能引脚是否存在<br>0：'nBE'引脚不存在。DW_MODE位域定义了固定数据位宽。<br>1：'nBE'引脚存在。每次访问的数据总线都通过nBE引脚状态进行验证。<br>对主机只读<br>复位类型：SYSRSn |
| 4 | RW_MODE | R/W | 0h | 读-写 模式<br>定义引脚来控制读-写操作<br>0：'nOE'和'nWE'都可以用来控制读写操作。<br>1：’nOE‘引脚将作为一个单独的 ‘RnW’ 引脚来控制读取和写入操作。<br>对主机只读<br>复位类型：SYSRSn |
| 3-2 | RESERVED | R-0 | 0h | 保留位 |
| 1-0 | DW_MODE | R/W | 0h | 数据位宽模式<br>主机访问的数据位宽<br>0x0：8比特数据端口<br>0x1：16比特数据端口<br>其他：默认为16比特数据端口<br>对主机只读<br>复位类型：SYSRSn |

### 11.4.2.5 HICPINPOLCR 寄存器

偏移地址：10h　复位值：00000000h

引脚极性控制寄存器。

表 11-25 HICPINPOLCR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | RDY_POL | WE_POL | OE_POL | BEN_POL | CS_POL |
| R-0-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 11-26 HICPINPOLCR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-5 | RESERVED | R-0 | 0h | 保留位 |
| 4 | RDY_POL | R/W | 0h | 定义就绪（nRDY）输出引脚的极性<br>0：RDY引脚低电平激活<br>1：RDY引脚高电平激活<br>对主机只读<br>复位类型：SYSRSn |
| 3 | WE_POL | R/W | 0h | 定义写使能（nWE）输入引脚的极性<br>0：写使能引脚低电平激活<br>1：写使能引脚高电平激活<br>如果HIC被配置使用一个单RnW引脚，那么这个比特的配置将被忽略。<br>对主机只读<br>复位类型：SYSRSn |
| 2 | OE_POL | R/W | 0h | 定义输出使能（nOE）输入引脚的极性<br>0：输出使能引脚低电平激活<br>1：输出使能引脚高电平激活<br>如果HIC被配置使用一个单RnW引脚，那么这个比特的配置将被忽略。<br>对主机只读<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | BEN_POL | R/W | 0h | 定义字节使能（nBE）输入引脚的极性<br>0：字节使能引脚低电平激活<br>1：字节使能引脚高电平激活<br>对主机只读<br>复位类型：SYSRSn |
| 0 | CS_POL | R/W | 0h | 定义片选（nCS）输入引脚的极性<br>0：片选引脚低电平激活<br>1：片选引脚高电平激活<br>对主机只读<br>复位类型：SYSRSn |

### 11.4.2.6 HICBASESEL 寄存器

偏移地址：14h　复位值：00000000h

基地址选择寄存器。

表 11-27 HICBASESEL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | BASE_SELECT | | |
| R-0-0h | | | | | R/W-0h | | |

表 11-28 HICBASESEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-3 | RESERVED | R-0 | 0h | 保留位 |
| 2-0 | BASE_SELECT | R/W | 0h | 基地址寄存器集合的选择值<br>000：选择基地址0寄存器<br>001：选择基地址1寄存器<br>010：选择基地址2寄存器<br>011：选择基地址3寄存器<br>100：选择基地址4寄存器<br>101：选择基地址5寄存器<br>110：选择基地址6寄存器 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 111：选择基地址7寄存器<br>对主机可读可写<br>复位类型：SYSRSn |

### 11.4.2.7 HICHOSTCR 寄存器

偏移地址：18h　复位值：00000000h

主机控制寄存器。

表 11-29 HICHOSTCR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HKEY | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | PAGESEL | ACCSIZE | EALLOW_EN |
| R-0-0h | | | | | R/W-0h | R/W-0h | R/W-0h |

表 11-30 HICHOSTCR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | RESERVED | R-0 | 0h | 保留位 |
| 15-8 | HKEY | R/W | 0h | 启用/禁用对HOSTCR[7:0]位域写入的KEY<br>只有在写入期间将HKEY设置为0xA5时，HOSTCR[7:0]位域才能被写入。在HKEY位域上写任何其他模式将被忽略。<br>从该位域读将始终返回0<br>对主机可读可写<br>复位类型：SYSRSn |
| 7-3 | RESERVED | R-0 | 0h | 保留位 |
| 2 | PAGESEL | R/W | 0h | 在BASE_SEL引脚和BASE_SELECT寄存器直接的选择<br>0：对于一个主机访问设备区域，使用BASE_SELECT寄存器值选择需要的HICDBADDRx寄存器 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：对于一个主机访问设备区域，使用BASE_SEL引脚值选择需要的HICDBADDRx寄存器<br>该位域只能与HKEY位域一起写入<br>对主机可读可写<br>复位类型：SYSRSn |
| 1 | ACCSIZE | R/W | 0h | 对HIC的16或32比特访问的选择<br>0：<br>在主机端口的写 - MMR或启动器端口的写是16比特大小。对于8比特主机，数据将被打包为16比特，然后才会触发对MMR或启动器端口的写入。<br>在主机端口的读 - 从MMR或启动器端口读出16比特数据。<br>1：<br>在主机端口的写 - MMR或启动器端口的写是32比特大小。对于8/16比特主机，数据将被打包为32比特，然后才会触发对MMR或启动器端口的写入。<br>在主机端口的读 - 从MMR或启动器端口读出32比特数据。<br>该位域只能与HKEY位域一起写入<br>对主机可读可写<br>复位类型：SYSRSn |
| 0 | EALLOW_EN | R/W | 0h | 定义针对HIC启动器端口的EALLOW信号<br>0：HIC不能写到受EALLOW保护的外设寄存器<br>1：HIC可以访问受EALLOW保护的外设寄存器<br>如果HICMODECR.EN_HOSTWREALLOW=1，<br>该位域只能与HKEY位域一起写入<br>对主机可读可写<br>复位类型：SYSRSn |

### 11.4.2.8 HICERRADDR 寄存器

偏移地址：1Ch　复位值：00000000h

主机错误地址寄存器。

表 11-31 HICERRADDR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| H2D_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| H2D_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| H2D_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| H2D_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |

表 11-32 HICERRADDR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | H2D_TOKEN | R/W | 0h | 主机到设备令牌是一个通用目的寄存器，可用于从主机到设备软件发送任何信息。典型用法是发送H2DBUF区域的缓冲区计数。<br>向该寄存器的低半字写入任何数据都会自动触发H2DINT中断。<br>对主机可读可写<br>复位类型：SYSRSn |

### 11.4.2.9 HICH2DTOKEN 寄存器

偏移地址：20h　复位值：00000000h

主机到器件令牌寄存器。

表 11-33 HICH2DTOKEN 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| H2D_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| H2D_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| H2D_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| H2D_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |

表 11-34 HICH2DTOKEN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | H2D_TOKEN | R/W | 0h | 主机到设备令牌是一个通用目的寄存器，可用于从主机到设备软件发送任何信息。典型用法是发送H2DBUF区域的缓冲区计数。<br>向该寄存器的低半字写入任何数据都会自动触发H2DINT中断。<br>对主机可读可写<br>复位类型：SYSRSn |

## 11.4.2.10 HICD2HTOKEN 寄存器

偏移地址：24h　复位值：00000000h

器件到主机令牌寄存器。

表 11-35 HICD2HTOKEN 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| D2H_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| D2H_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| D2H_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| D2H_TOKEN | | | | | | | |
| R/W-0h | | | | | | | |

表 11-36 HICD2HTOKEN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | D2H_TOKEN | R/W | 0h | 设备到主机令牌是一个通用目的寄存器，可用于从设备到主机软件发送任何信息。典型用法是发送D2HBUF区域的缓冲区计数。<br>向该寄存器的低半字写入任何数据都会自动触发D2HINT中断。<br>对主机只读<br>复位类型：SYSRSn |

## 11.4.2.11 HICDBADDR0 寄存器

偏移地址：28h　复位值：00000000h

器件基地址寄存器 0。

表 11-37 HICDBADDR0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BASE_ADDR | 保留 | | | | | | |
| R/W-0h | R-0-0h | | | | | | |

表 11-38 HICDBADDR0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | BASE_ADDR | R/W | 0h | 主机要访问的设备地址空间内区域的基地址<br>寄存器中的有效位与输入主机地址适当连接，从而在HIC的启动器端口上形成完整的32位地址。<br>当DW_MODE=8比特时，BASE_ADDR[31..7]应当被编程。在这种情况下，主机可以通过启动器端口访问256字节的设备区域。<br>当DW_MODE=16比特时，BASE_ADDR[31..8]应当被编程。在这种情况下，主机可以通过启动器端口访问512字节的设备区域。<br>注意：通过HICBASESEL.BASE_SELECT位域值来确定用于访问的精确HICDBADDR0寄存器。<br>对主机只读<br>复位类型：SYSRSn |
| 6-0 | RESERVED | R-0 | 0h | 保留位 |

## 11.4.2.12 HICDBADDR1 寄存器

偏移地址：2Ch　复位值：00000000h

器件基地址寄存器 1。

表 11-39 HICDBADDR1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BASE_ADDR | 保留 | | | | | | |
| R/W-0h | R-0-0h | | | | | | |

表 11-40 HICDBADDR1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | BASE_ADDR | R/W | 0h | 主机要访问的设备地址空间内区域的基地址<br>寄存器中的有效位与输入主机地址适当连接，从而在HIC的启动器端口上形成完整的32位地址。<br>当DW_MODE=8比特时，BASE_ADDR[31..7]应当被编程。在这种情况下，主机可以通过启动器端口访问256字节的设备区域。<br>当DW_MODE=16比特时，BASE_ADDR[31..8]应当被编程。在这种情况下，主机可以通过启动器端口访问512字节的设备区域。<br>注意：通过HICBASESEL.BASE_SELECT位域值来确定用于访问的精确HICDBADDR1寄存器。<br>对主机只读<br>复位类型：SYSRSn |
| 6-0 | RESERVED | R-0 | 0h | 保留位 |

## 11.4.2.13 HICDBADDR2 寄存器

偏移地址：30h　复位值：00000000h

器件基地址寄存器 2。

表 11-41 HICDBADDR2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BASE_ADDR | 保留 | | | | | | |
| R/W-0h | R-0-0h | | | | | | |

表 11-42 HICDBADDR2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | BASE_ADDR | R/W | 0h | 主机要访问的设备地址空间内区域的基地址<br>寄存器中的有效位与输入主机地址适当连接，从而在HIC的启动器端口上形成完整的32位地址。<br>当DW_MODE=8比特时，BASE_ADDR[31..7]应当被编程。在这种情况下，主机可以通过启动器端口访问256字节的设备区域。<br>当DW_MODE=16比特时，BASE_ADDR[31..8]应当被编程。在这种情况下，主机可以通过启动器端口访问512字节的设备区域。<br>注意：通过HICBASESEL.BASE_SELECT位域值来确定用于访问的精确HICDBADDR2寄存器。<br>对主机只读<br>复位类型：SYSRSn |
| 6-0 | RESERVED | R-0 | 0h | 保留位 |

## 11.4.2.14 HICDBADDR3 寄存器

偏移地址：34h　复位值：00000000h

器件基地址寄存器 3。

表 11-43 HICDBADDR3 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BASE_ADDR | 保留 | | | | | | |
| R/W-0h | R-0-0h | | | | | | |

表 11-44 HICDBADDR3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | BASE_ADDR | R/W | 0h | 主机要访问的设备地址空间内区域的基地址<br>寄存器中的有效位与输入主机地址适当连接，从而在HIC的启动器端口上形成完整的32位地址。<br>当DW_MODE=8比特时，BASE_ADDR[31..7]应当被编程。在这种情况下，主机可以通过启动器端口访问256字节的设备区域。<br>当DW_MODE=16比特时，BASE_ADDR[31..8]应当被编程。在这种情况下，主机可以通过启动器端口访问512字节的设备区域。<br>注意：通过HICBASESEL.BASE_SELECT位域值来确定用于访问的精确HICDBADDR3寄存器。<br>对主机只读<br>复位类型：SYSRSn |
| 6-0 | RESERVED | R-0 | 0h | 保留位 |

## 11.4.2.15 HICDBADDR4 寄存器

偏移地址：38h　复位值：00000000h

器件基地址寄存器 4。

表 11-45 HICDBADDR4 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BASE_ADDR | 保留 | | | | | | |
| R/W-0h | R-0-0h | | | | | | |

表 11-46 HICDBADDR4 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | BASE_ADDR | R/W | 0h | 主机要访问的设备地址空间内区域的基地址<br>寄存器中的有效位与输入主机地址适当连接，从而在HIC的启动器端口上形成完整的32位地址。<br>当DW_MODE=8比特时，BASE_ADDR[31..7]应当被编程。在这种情况下，主机可以通过启动器端口访问256字节的设备区域。<br>当DW_MODE=16比特时，BASE_ADDR[31..8]应当被编程。在这种情况下，主机可以通过启动器端口访问512字节的设备区域。<br>注意：通过HICBASESEL.BASE_SELECT位域值来确定用于访问的精确HICDBADDR4寄存器。<br>对主机只读<br>复位类型：SYSRSn |
| 6-0 | RESERVED | R-0 | 0h | 保留位 |

## 11.4.2.16 HICDBADDR5 寄存器

偏移地址：3Ch　复位值：00000000h

器件基地址寄存器 5。

表 11-47 HICDBADDR5 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BASE_ADDR | 保留 | | | | | | |
| R/W-0h | R-0-0h | | | | | | |

表 11-48 HICDBADDR5 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | BASE_ADDR | R/W | 0h | 主机要访问的设备地址空间内区域的基地址<br>寄存器中的有效位与输入主机地址适当连接，从而在HIC的启动器端口上形成完整的32位地址。<br>当DW_MODE=8比特时，BASE_ADDR[31..7]应当被编程。在这种情况下，主机可以通过启动器端口访问256字节的设备区域。<br>当DW_MODE=16比特时，BASE_ADDR[31..8]应当被编程。在这种情况下，主机可以通过启动器端口访问512字节的设备区域。<br>注意：通过HICBASESEL.BASE_SELECT位域值来确定用于访问的精确HICDBADDR5寄存器。<br>对主机只读<br>复位类型：SYSRSn |
| 6-0 | RESERVED | R-0 | 0h | 保留位 |

## 11.4.2.17 HICDBADDR6 寄存器

偏移地址：40h　复位值：00000000h

器件基地址寄存器 6。

表 11-49 HICDBADDR6 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BASE_ADDR | 保留 | | | | | | |
| R/W-0h | R-0-0h | | | | | | |

表 11-50 HICDBADDR6 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | BASE_ADDR | R/W | 0h | 主机要访问的设备地址空间内区域的基地址<br>寄存器中的有效位与输入主机地址适当连接，从而在HIC的启动器端口上形成完整的32位地址。<br>当DW_MODE=8比特时，BASE_ADDR[31..7]应当被编程。在这种情况下，主机可以通过启动器端口访问256字节的设备区域。<br>当DW_MODE=16比特时，BASE_ADDR[31..8]应当被编程。在这种情况下，主机可以通过启动器端口访问512字节的设备区域。<br>注意：通过HICBASESEL.BASE_SELECT位域值来确定用于访问的精确HICDBADDR6寄存器。<br>对主机只读<br>复位类型：SYSRSn |
| 6-0 | RESERVED | R-0 | 0h | 保留位 |

## 11.4.2.18 HICDBADDR7 寄存器

偏移地址：44h　复位值：00000000h

器件基地址寄存器 7。

表 11-51 HICDBADDR7 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BASE_ADDR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BASE_ADDR | 保留 | | | | | | |
| R/W-0h | R-0-0h | | | | | | |

表 11-52 HICDBADDR7 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | BASE_ADDR | R/W | 0h | 主机要访问的设备地址空间内区域的基地址<br>寄存器中的有效位与输入主机地址适当连接，从而在HIC的启动器端口上形成完整的32位地址。<br>当DW_MODE=8比特时，BASE_ADDR[31..7]应当被编程。在这种情况下，主机可以通过启动器端口访问256字节的设备区域。<br>当DW_MODE=16比特时，BASE_ADDR[31..8]应当被编程。在这种情况下，主机可以通过启动器端口访问512字节的设备区域。<br>注意：通过HICBASESEL.BASE_SELECT位域值来确定用于访问的精确HICDBADDR7寄存器。<br>对主机只读<br>复位类型：SYSRSn |
| 6-0 | RESERVED | R-0 | 0h | 保留位 |

## 11.4.2.19 HICH2DINTEN 寄存器

偏移地址：50h　复位值：00000000h

主机到器件中断使能。

表 11-53 HICH2DINTEN 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | ILLRD_INTEN | ILLWR_INTEN | BUSERR_INTEN | H2D_INTEN |
| R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 11-54 HICH2DINTEN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留位 |
| 3 | ILLRD_INTEN | R/W | 0h | 启用非法读错误中断<br>0：非法读事件不会导致中断<br>1：非法读事件会在H2DINT端口上触发一个中断<br>对主机可读可写<br>复位类型：SYSRSn |
| 2 | ILLWR_INTEN | R/W | 0h | 启用非法写错误中断<br>0：非法写事件不会导致中断<br>1：非法写事件会在H2DINT端口上触发一个中断<br>对主机可读可写<br>复位类型：SYSRSn |
| 1 | BUSERR_INTEN | R/W | 0h | 启用总线错误中断<br>0：总线错误不会导致中断<br>1：总线错误会在H2DINT端口上触发一个中断<br>对主机可读可写<br>复位类型：SYSRSn |
| 0 | H2D_INTEN | R/W | 0h | 启用主机到设备数据就绪中断<br>0：主机到设备数据就绪不会导致中断<br>1：主机到设备数据就绪会在H2DINT端口上触发一个中断<br>对主机可读可写<br>复位类型：SYSRSn |

### 11.4.2.20 HICH2DINTFLG 寄存器

偏移地址：54h　复位值：00000000h

主机到器件中断状态标志。

表 11-55 HICH2DINTFLG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | ILLRD_FLG | ILLWR_FLG | BUSERR_FLG | H2D_FLG |
| R-0-0h | | | | R-0h | R-0h | R-0h | R-0h |

表 11-56 HICH2DINTFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留位 |
| 3 | ILLRD_FLG | R | 0h | 非法读错误中断状态<br>0：非法读事件未被检测到<br>1：非法读事件被检测到<br>对主机只读<br>复位类型：SYSRSn |
| 2 | ILLWR_FLG | R | 0h | 非法写错误中断状态<br>0：非法写事件未被检测到<br>1：非法写事件被检测到<br>对主机只读<br>复位类型：SYSRSn |
| 1 | BUSERR_FLG | R | 0h | 总线错误中断状态<br>0：总线错误事件未被检测到<br>1：总线错误事件被检测到<br>当主机和设备同时对单个寄存器进行写访问时，会产生BusError，这将导致数据丢失。<br>对主机只读<br>复位类型：SYSRSn |
| 0 | H2D_FLG | R | 0h | 主机到设备数据就绪中断状态 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 0：主机到设备数据就绪事件未被检测到<br>1：主机到设备数据就绪事件被检测到<br>对主机只读<br>复位类型：SYSRSn |

### 11.4.2.21 HICH2DINTCLR 寄存器

偏移地址：58h　复位值：00000000h

主机到器件中断状态清除。

表 11-57 HICH2DINTCLR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | ILLRD_CLR | ILLWR_CLR | BUSERR_CLR | H2D_CLR |
| R-0-0h | | | | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |

表 11-58 HICH2DINTCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留位 |
| 3 | ILLRD_CLR | R/W1C | 0h | 清除非法读错误标志位，从而清除中断<br>写'1'来清除这个标志位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-清除<br>复位类型：SYSRSn |
| 2 | ILLWR_CLR | R/W1C | 0h | 清除非法写错误标志位，从而清除中断<br>写'1'来清除这个标志位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-清除<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | BUSERR_CLR | R/W1C | 0h | 清除总线错误标志位，从而清除中断<br>写'1'来清除这个标志位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-清除<br>复位类型：SYSRSn |
| 0 | H2D_CLR | R/W1C | 0h | 清除主机到设备数据就绪标志位，从而清除中断<br>写'1'来清除这个标志位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-清除<br>复位类型：SYSRSn |

### 11.4.2.22 HICH2DINTFRC 寄存器

偏移地址：5Ch　复位值：00000000h

主机到器件中断强制置位。

表 11-59 HICH2DINTFRC 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | ILLRD_INTFRC | ILLWR_INTFRC | BUSERR_INTFRC | H2D_INTFRC |
| R-0-0h | | | | R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h |

表 11-60 HICH2DINTFRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-4 | RESERVED | R-0 | 0h | 保留位 |
| 3 | ILLRD_INTFRC | R/W1S | 0h | 强制执行非法读错误中断<br>写'1'来设置该位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-设置 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 复位类型：SYSRSn |
| 2 | ILLWR_INTFRC | R/W1S | 0h | 强制执行非法写错误中断<br>写'1'来设置该位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-设置<br>复位类型：SYSRSn |
| 1 | BUSERR_INTFRC | R/W1S | 0h | 强制执行总线错误中断<br>写'1'来设置BUSERR_FLG<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-设置<br>复位类型：SYSRSn |
| 0 | H2D_INTFRC | R/W1S | 0h | 强制执行主机到设备数据就绪中断<br>写'1'来设置H2D_FLG<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-设置<br>复位类型：SYSRSn |

## 11.4.2.23 HICD2HINTEN 寄存器

偏移地址：60h　复位值：00000000h

器件到主机中断使能。

表 11-61 HICD2HINTEN 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| EVTRIG_INTEN | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EVTRIG_INTEN | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | ACCVIO_INTEN | ILLRD_INTEN | ILLWR_INTEN | BUSERR_INTEN | D2H_INTEN |
| R-0-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 11-62 HICD2HINTEN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | EVTRIG_INTEN | R/W | 0h | 启用事件触发器中断<br>0：在EVT_TRIG[n]（n=0..15）上发生事件时，D2HINT端口不会触发中断<br>1：在EVT_TRIG[n]（n=0..15）上检测到激活的高电平事件时，D2HINT端口上触发中断<br>对主机可读可写<br>复位类型：SYSRSn |
| 15-5 | RESERVED | R-0 | 0h | 保留位 |
| 4 | ACCVIO_INTEN | R/W | 0h | 启用访问违例（针对设备访问）中断<br>0：访问违例不会导致中断<br>1：访问违例将会在D2HINT端口触发中断<br>对主机可读可写<br>复位类型：SYSRSn |
| 3 | ILLRD_INTEN | R/W | 0h | 启用非法读错误中断<br>0：非法读事件不会导致中断<br>1：非法读事件会在D2HINT端口上触发一个中断<br>对主机可读可写<br>复位类型：SYSRSn |
| 2 | ILLWR_INTEN | R/W | 0h | 启用非法写错误中断<br>0：非法写事件不会导致中断<br>1：非法写事件会在D2HINT端口上触发一个中断<br>对主机可读可写<br>复位类型：SYSRSn |
| 1 | BUSERR_INTEN | R/W | 0h | 启用总线错误中断<br>0：总线错误不会导致中断<br>1：总线错误会在D2HINT端口上触发一个中断<br>对主机可读可写<br>复位类型：SYSRSn |
| 0 | D2H_INTEN | R/W | 0h | 启用主机到设备数据就绪中断<br>0：设备到主机数据就绪不会导致中断<br>1：设备到主机数据就绪会在D2HINT端口上触发一个中断<br>对主机可读可写<br>复位类型：SYSRSn |

## 11.4.2.24 HICD2HINTFLG 寄存器

偏移地址：64h　复位值：00000000h

器件到主机中断状态标志。

表 11-63 HICD2HINTFLG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| EVTRIG_FLG | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EVTRIG_FLG | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | ACCVIO_FLG | ILLRD_FLG | ILLWR_FLG | BUSERR_FLG | D2H_FLG |
| R-0-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |

表 11-64 HICD2HINTFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | EVTRIG_FLG | R | 0h | 事件触发状态<br>0：在EVT_TRIG[n]（n=0..15）上没有事件被检测到<br>1：在EVT_TRIG[n]（n=0..15）上有事件被检测到<br>对主机只读<br>复位类型：SYSRSn |
| 15-5 | RESERVED | R-0 | 0h | 保留位 |
| 4 | ACCVIO_FLG | R | 0h | 访问违例错误状态<br>0：启动器端口访问违例事件没有被检测到<br>1：启动器端口访问违例事件有被检测到<br>对主机只读<br>复位类型：SYSRSn |
| 3 | ILLRD_FLG | R | 0h | 非法读错误状态<br>0：非法读事件未被检测到<br>1：非法读事件被检测到<br>对主机只读<br>复位类型：SYSRSn |
| 2 | ILLWR_FLG | R | 0h | 非法写错误状态<br>0：非法写事件未被检测到<br>1：非法写事件被检测到<br>对主机只读<br>复位类型：SYSRSn |
| 1 | BUSERR_FLG | R | 0h | 总线错误事件状态<br>0：总线错误事件未被检测到 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：总线错误事件被检测到<br>当主机和设备同时对单个寄存器进行写访问时，会产生数据丢失，从而产生BusError<br>对主机只读<br>复位类型：SYSRSn |
| 0 | D2H_FLG | R | 0h | 主机到设备数据就绪事件状态<br>0：设备到主机数据就绪事件未被检测到<br>1：设备到主机数据就绪事件被检测到<br>对主机只读<br>复位类型：SYSRSn |

### 11.4.2.25 HICD2HINTCLR 寄存器

偏移地址：68h　复位值：00000000h

器件到主机中断状态清除。

表 11-65 HICD2HINTCLR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| EVTRIG_CLR | | | | | | | |
| R/W1C-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EVTRIG_CLR | | | | | | | |
| R/W1C-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | ACCVIO_CLR | ILLRD_CLR | ILLWR_CLR | BUSERR_CLR | D2H_CLR |
| R-0-0h | | | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |

表 11-66 HICD2HINTCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | EVTRIG_CLR | R/W1C | 0h | 清除事件触发器标志位，从而清除中断<br>写'1'来清除这个标志位EVTRIG_FLG[n]（0..15）<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-清除<br>复位类型：SYSRSn |
| 15-5 | RESERVED | R-0 | 0h | 保留位 |
| 4 | ACCVIO_CLR | R/W1C | 0h | 清除访问违例错误标志位，从而清除中断 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 写'1'来清除这个标志位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-清除<br>复位类型：SYSRSn |
| 3 | ILLRD_CLR | R/W1C | 0h | 清除非法读错误标志位，从而清除中断<br>写'1'来清除这个标志位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-清除<br>复位类型：SYSRSn |
| 2 | ILLWR_CLR | R/W1C | 0h | 清除非法写错误标志位，从而清除中断<br>写'1'来清除这个标志位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-清除<br>复位类型：SYSRSn |
| 1 | BUSERR_CLR | R/W1C | 0h | 清除总线错误标志位，从而清除中断<br>写'1'来清除这个标志位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-清除<br>复位类型：SYSRSn |
| 0 | D2H_CLR | R/W1C | 0h | 清除设备到主机数据就绪标志位，从而清除中断<br>写'1'来清除这个标志位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-清除<br>复位类型：SYSRSn |

## 11.4.2.26 HICD2HINTFRC 寄存器

偏移地址：6Ch　复位值：00000000h

器件到主机中断强制置位。

表 11-67 HICD2HINTFRC 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| EVTRIG_INTFRC | | | | | | | |
| R/W1S-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EVTRIG_INTFRC | | | | | | | |
| R/W1S-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | ACCVIO_INTFRC | ILLRD_INTFRC | ILLWR_INTFRC | BUSERR_INTFRC | D2H_INTFRC |
| R-0-0h | | | R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h |

表 11-68 HICD2HINTFRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | EVTRIG_INTFRC | R/W1S | 0h | 强制执行事件触发器中断<br>写'1'来设置该位EVTRIG_FLG[n]（n=0..15）<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-设置<br>复位类型：SYSRSn |
| 15-5 | RESERVED | R-0 | 0h | 保留位 |
| 4 | ACCVIO_INTFRC | R/W1S | 0h | 强制执行访问违例中断<br>写'1'来设置该位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-设置<br>复位类型：SYSRSn |
| 3 | ILLRD_INTFRC | R/W1S | 0h | 强制执行非法读错误中断<br>写'1'来设置该位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-设置<br>复位类型：SYSRSn |
| 2 | ILLWR_INTFRC | R/W1S | 0h | 强制执行非法写错误中断<br>写'1'来设置该位<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-设置<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | BUSERR_INTFRC | R/W1S | 0h | 强制执行总线错误中断<br>写'1'来设置BUSERR_FLG<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-设置<br>复位类型：SYSRSn |
| 0 | D2H_INTFRC | R/W1S | 0h | 强制执行设备到主机数据就绪中断<br>写'1'来设置H2D_FLG<br>写'0'无影响<br>读此位一直返回'0'<br>对主机读-写-到-设置<br>复位类型：SYSRSn |

## 11.4.2.27 HICACCVIOADDR 寄存器

偏移地址：70h　复位值：00000000h

访问违规地址。

表 11-69 HICACCVIOADDR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ACCVIO_ADDR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ACCVIO_ADDR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ACCVIO_ADDR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACCVIO_ADDR | | | | | | | |
| R-0h | | | | | | | |

表 11-70 HICACCVIOADDR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | ACCVIO_ADDR | R | 0h | 启动器端口访问设备区域的地址，该地址导致未实现范围或访问保护的违例。该值将在首次捕获时冻结，直至用户清除HICD2HINTFLG寄存器的ACCVIO_FLG标志位。<br>当HIC向其启动器端口发出访问，允许主机访问设备区域时，若检测到违规，启动器端口的地址 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 将被记录到该寄存器位域，并在HICD2HINTFLG寄存器中设置对应的错误标志。<br>对主机只读<br>复位类型：SYSRSn |

### 11.4.2.28 H2D_BUF0 寄存器

偏移地址：80h　复位值：00000000h

主机到器件缓冲区 0。

表 11-71 H2D_BUF0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-72 H2D_BUF0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

### 11.4.2.29 H2D_BUF1 寄存器

偏移地址：84h　复位值：00000000h

主机到器件缓冲区 1。

表 11-73 H2D_BUF1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-74 H2D_BUF1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.30 H2D_BUF2 寄存器

偏移地址：88h　复位值：00000000h

主机到器件缓冲区 2。

表 11-75 H2D_BUF2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-76 H2D_BUF2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

### 11.4.2.31 H2D_BUF3 寄存器

偏移地址：8Ch　复位值：00000000h

主机到器件缓冲区 3。

表 11-77 H2D_BUF3 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-78 H2D_BUF3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

### 11.4.2.32 H2D_BUF4 寄存器

偏移地址：90h　复位值：00000000h

主机到器件缓冲区 4。

表 11-79 H2D_BUF4 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-80 H2D_BUF4 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.33 H2D_BUF5 寄存器

偏移地址：94h　复位值：00000000h

主机到器件缓冲区 5。

表 11-81 H2D_BUF5 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-82 H2D_BUF5 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.34 H2D_BUF6 寄存器

偏移地址：98h　复位值：00000000h

主机到器件缓冲区 6。

表 11-83 H2D_BUF6 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-84 H2D_BUF6 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.35 H2D_BUF7 寄存器

偏移地址：9Ch　复位值：00000000h

主机到器件缓冲区 7。

表 11-85 H2D_BUF7 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-86 H2D_BUF7 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.36 H2D_BUF8 寄存器

偏移地址：A0h　复位值：00000000h

主机到器件缓冲区 8。

表 11-87 H2D_BUF8 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-88 H2D_BUF8 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.37 H2D_BUF9 寄存器

偏移地址：A4h　复位值：00000000h

主机到器件缓冲区 9。

表 11-89 H2D_BUF9 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-90 H2D_BUF9 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.38 H2D_BUF10 寄存器

偏移地址：A8h　复位值：00000000h

主机到器件缓冲区 10。

表 11-91 H2D_BUF10 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-92 H2D_BUF10 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.39 H2D_BUF11 寄存器

偏移地址：ACh　复位值：00000000h

主机到器件缓冲区 11。

表 11-93 H2D_BUF11 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-94 H2D_BUF11 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.40 H2D_BUF12 寄存器

偏移地址：B0h　复位值：00000000h

主机到器件缓冲区 12。

表 11-95 H2D_BUF12 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-96 H2D_BUF12 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.41 H2D_BUF13 寄存器

偏移地址：B4h　复位值：00000000h

主机到器件缓冲区 13。

表 11-97 H2D_BUF13 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-98 H2D_BUF13 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.42 H2D_BUF14 寄存器

偏移地址：B8h　复位值：00000000h

主机到器件缓冲区 14。

表 11-99 H2D_BUF14 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-100 H2D_BUF14 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.43 H2D_BUF15 寄存器

偏移地址：BCh　复位值：00000000h

主机到器件缓冲区 15。

表 11-101 H2D_BUF15 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-102 H2D_BUF15 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 主机写入的数据，用于设备使用<br>这些寄存器可以被主要主机读写，但是如果H2DBUF_DEVWREN比特被设置为'1'，这些寄存器也可以被设备写<br>复位类型：SYSRSn |

## 11.4.2.44 D2H_BUF0 寄存器

偏移地址：C0h　复位值：00000000h

器件到主机缓冲区 0。

表 11-103 D2H_BUF0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-104 D2H_BUF0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

## 11.4.2.45 D2H_BUF1 寄存器

偏移地址：C4h　复位值：00000000h

器件到主机缓冲区 1。

表 11-105 D2H_BUF1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-106 D2H_BUF1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

### 11.4.2.46 D2H_BUF2 寄存器

偏移地址：C8h　复位值：00000000h

器件到主机缓冲区 2。

表 11-107 D2H_BUF2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-108 D2H_BUF2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

### 11.4.2.47 D2H_BUF3 寄存器

偏移地址：CCh　复位值：00000000h

器件到主机缓冲区 3。

表 11-109 D2H_BUF3 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-110 D2H_BUF3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

## 11.4.2.48 D2H_BUF4 寄存器

偏移地址：D0h　复位值：00000000h

器件到主机缓冲区 4。

表 11-111 D2H_BUF4 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-112 D2H_BUF4 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

### 11.4.2.49 D2H_BUF5 寄存器

偏移地址：D4h　复位值：00000000h

器件到主机缓冲区 5。

表 11-113 D2H_BUF5 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-114 D2H_BUF5 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

### 11.4.2.50 D2H_BUF6 寄存器

偏移地址：D8h　复位值：00000000h

器件到主机缓冲区 6。

表 11-115 D2H_BUF6 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-116 D2H_BUF6 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

## 11.4.2.51 D2H_BUF7 寄存器

偏移地址：DCh　复位值：00000000h

器件到主机缓冲区 7。

表 11-117 D2H_BUF7 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-118 D2H_BUF7 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

### 11.4.2.52 D2H_BUF8 寄存器

偏移地址：E0h　复位值：00000000h

器件到主机缓冲区 8。

表 11-119 D2H_BUF8 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-120 D2H_BUF8 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

### 11.4.2.53 D2H_BUF9 寄存器

偏移地址：E4h　复位值：00000000h

器件到主机缓冲区 9。

表 11-121 D2H_BUF9 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-122 D2H_BUF9 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

## 11.4.2.54 D2H_BUF10 寄存器

偏移地址：E8h　复位值：00000000h

器件到主机缓冲区 10。

表 11-123 D2H_BUF10 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-124 D2H_BUF10 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

### 11.4.2.55 D2H_BUF11 寄存器

偏移地址：ECh　复位值：00000000h

器件到主机缓冲区 11。

表 11-125 D2H_BUF11 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-126 D2H_BUF11 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

### 11.4.2.56 D2H_BUF12 寄存器

偏移地址：F0h　复位值：00000000h

器件到主机缓冲区 12。

表 11-127 D2H_BUF12 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-128 D2H_BUF12 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

## 11.4.2.57 D2H_BUF13 寄存器

偏移地址：F4h　复位值：00000000h

器件到主机缓冲区 13。

表 11-129 D2H_BUF13 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-130 D2H_BUF13 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

## 11.4.2.58 D2H_BUF14 寄存器

偏移地址：F8h　复位值：00000000h

器件到主机缓冲区 14。

表 11-131 D2H_BUF14 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-132 D2H_BUF14 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

## 11.4.2.59 D2H_BUF15 寄存器

偏移地址：FCh　复位值：00000000h

器件到主机缓冲区 15。

表 11-133 D2H_BUF15 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data | | | | | | | |
| R/W-0h | | | | | | | |

表 11-134 D2H_BUF15 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | Data | R/W | 0h | 设备写入的数据，用于主机使用<br>这些寄存器可以被设备写，但是如果D2HBUF_HOSTWREN比特被设置为'1'，这些寄存器也可以被主机写<br>复位类型：SYSRSn |

## 11.4.3 HIC 寄存器与 DriverLib 函数的对应关系

表 11-135 HIC 寄存器与 DriverLib 函数的对应关系

| 文件 | DriverLib 函数 |
|---|---|
| **REV** | |
| hic.h | HIC_getRevision |
| **GCR** | |
| hic.h | HIC_enableHostInterface |
| **LOCK** | |
| hic.h | HIC_enableLock |
| hic.h | HIC_disableLock |
| **MODECR** | |
| hic.h | HIC_setConfig |
| hic.h | HIC_getConfig |
| **PINPOLCR** | |
| hic.h | HIC_setPinPolarity |
| hic.h | HIC_getPinPolarity |
| **BASESEL** | |

| 文件 | DriverLib 函数 |
| --- | --- |
| hic.h | HIC_selectBaseAddress |
| hic.h | HIC_getSelectedBaseAddressIndex |
| **HOSTCR** | |
| hic.h | HIC_configureHostAccessMode |
| hic.h | HIC_getHostAccessMode |
| **ERRADDR** | |
| hic.h | HIC_getErrorAddress |
| **H2DTOKEN** | |
| hic.h | HIC_getH2DToken |
| **D2HTOKEN** | |
| hic.h | HIC_setD2HToken |
| hic.h | HIC_getD2HToken |
| **DBADDR0** | |
| hic.h | HIC_setBaseAddress |
| hic.h | HIC_getBaseAddress |
| **DBADDR1** | |
| - | |
| **DBADDR2** | |
| - | |
| **DBADDR3** | |
| - | |
| **DBADDR4** | |
| - | |
| **DBADDR5** | |
| - | |
| **DBADDR6** | |
| - | |
| **DBADDR7** | |
| - | |
| **H2DINTEN** | |
| hic.h | HIC_enableH2DInterrupt |
| hic.h | HIC_disableH2DInterrupt |
| **H2DINTFLG** | |
| hic.h | HIC_getH2DInterruptStatus |
| **H2DINTCLR** | |
| hic.h | HIC_clearH2DInterrupt |

| 文件 | DriverLib 函数 |
| --- | --- |
| **H2DINTFRC** | |
| hic.h | HIC_forceH2DInterrupt |
| **D2HINTEN** | |
| hic.h | HIC_enableD2HInterrupt |
| hic.h | HIC_disableD2HInterrupt |
| **D2HINTFLG** | |
| hic.h | HIC_getD2HInterruptStatus |
| **D2HINTCLR** | |
| hic.h | HIC_clearD2HInterrupt |
| **D2HINTFRC** | |
| hic.h | HIC_forceD2HInterrupt |
| **ACCVIOADDR** | |
| hic.h | HIC_getAccessViolationAddress |
| **H2D_BUF0** | |
| hic.h | HIC_readH2DBuffer |
| hic.h | HIC_writeH2DBuffer |
| hic.h | HIC_clearH2DBuffer |
| **H2D_BUF1** | |
| - | |
| **H2D_BUF2** | |
| - | |
| **H2D_BUF3** | |
| - | |
| **H2D_BUF4** | |
| - | |
| **H2D_BUF5** | |
| - | |
| **H2D_BUF6** | |
| - | |
| **H2D_BUF7** | |
| - | |
| **H2D_BUF8** | |
| - | |
| **H2D_BUF9** | |
| - | |
| **H2D_BUF10** | |

| 文件 | DriverLib 函数 |
| --- | --- |
| - | |
| **H2D_BUF11** | |
| - | |
| **H2D_BUF12** | |
| - | |
| **H2D_BUF13** | |
| - | |
| **H2D_BUF14** | |
| - | |
| **H2D_BUF15** | |
| - | |
| **D2H_BUF0** | |
| hic.h | HIC_readD2HBuffer |
| hic.h | HIC_writeD2HBuffer |
| hic.h | HIC_clearD2HBuffer |
| **D2H_BUF1** | |
| - | |
| **D2H_BUF2** | |
| - | |
| **D2H_BUF3** | |
| - | |
| **D2H_BUF4** | |
| - | |
| **D2H_BUF5** | |
| - | |
| **D2H_BUF6** | |
| - | |
| **D2H_BUF7** | |
| - | |
| **D2H_BUF8** | |
| - | |
| **D2H_BUF9** | |
| - | |
| **D2H_BUF10** | |
| - | |
| **D2H_BUF11** | |

| 文件 | DriverLib 函数 |
| --- | --- |
| - | |
| **D2H_BUF12** | |
| - | |
| **D2H_BUF13** | |
| - | |
| **D2H_BUF14** | |
| - | |
| **D2H_BUF15** | |
| - | |

# 12 模拟子系统

## 12.1 简介

该设备的模拟模块包含模数转换器（ADC）、温度传感器和比较器子系统（CMPSS）。

### 12.1.1 特征

模拟子系统具备以下特性：

- 电压参考灵活
- 模数转换器（ADC）可参考 VREFHI 和 VSSA 引脚电压
- VREFHI 引脚电压可由外部驱动或内部带隙电压参考生成
- 内部电压参考范围可选 0V 至 3.3V 或 0V 至 2.5V
- 比较器模数转换器（DAC）参考 VDDA 和 VSSA，亦可参考 VDAC 引脚和 VSSA
- 引脚使用灵活
- 比较器子系统输入与数字输入与 ADC 输入复用
- 所有 ADC 均通过内部连接实现偏移自校准 VREFLO

### 12.1.2 模块示意图

以下模拟子系统框图展示了不同集成模拟模块与设备引脚之间的连接关系。这些引脚分为两类：模拟模块输入/输出引脚和参考引脚。

模拟引脚按 CMPSS 模块周围的不同模拟组进行排列。连接 CMPSS 输入引脚的引脚可直接用于 CMPSS ，无需额外操作，同时不影响其作为 ADC 输入端口的使用。

VDAC 参考引脚可用于为 CMPSS 模块设置替代参考范围（默认情况下 CMPSS DAC 以 VDDA 和 VSSA 为参考）。使用该引脚作为参考可防止通道被用

作 ADC 输入（但若需要，ADC 仍可采样 VDAC 电压）。参考选择可根据每个 CMPSS 的模块配置进行调整，具体操作通过模块配置寄存器完成。

以下说明适用于所有封装型号：

- 并非所有设备都提供全部模拟引脚，具体可用引脚请参阅设备数据手册。
- VREFHI 和 VREFLO 的允许电压范围请参照设备数据手册。
- VREFHI 引脚需外接电容，具体数值请查阅设备数据手册。
- 对于 CMPSS 模块，无论选择 VDAC 还是 VDDA 作为高参考， VSSA 均作为低参考。

图中展示了各模拟组的结构布局。

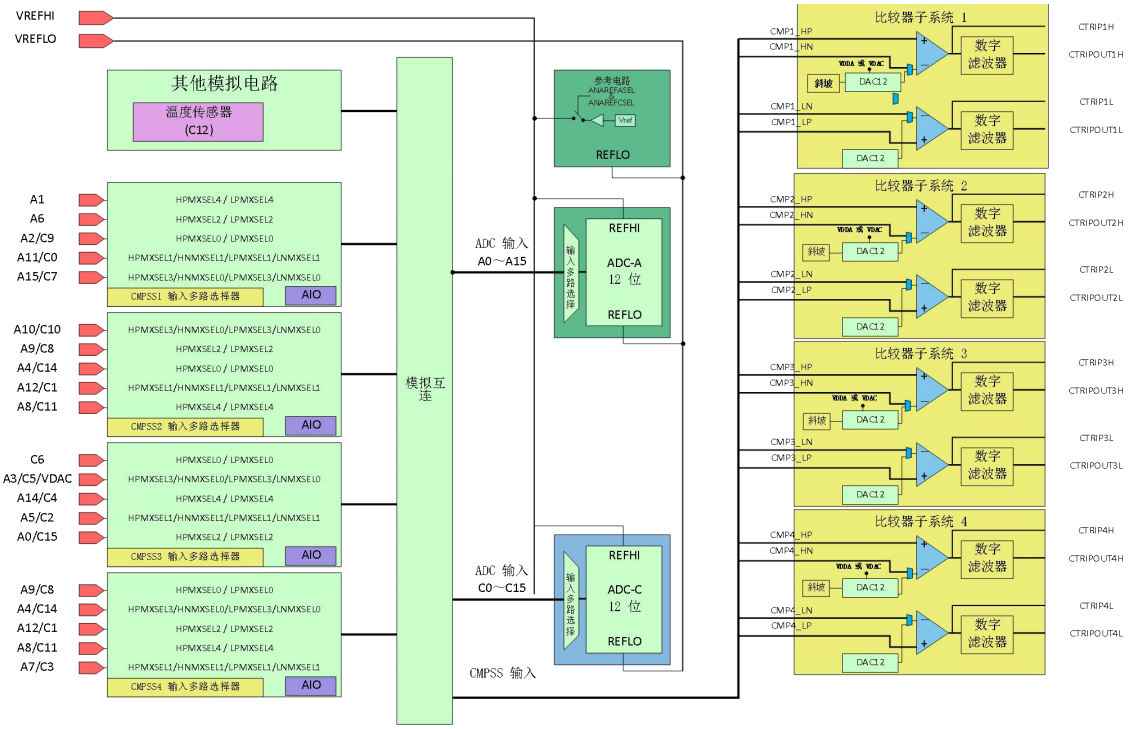



图 12-1 模拟子系统框图（80 引脚与 64 引脚 QFP）

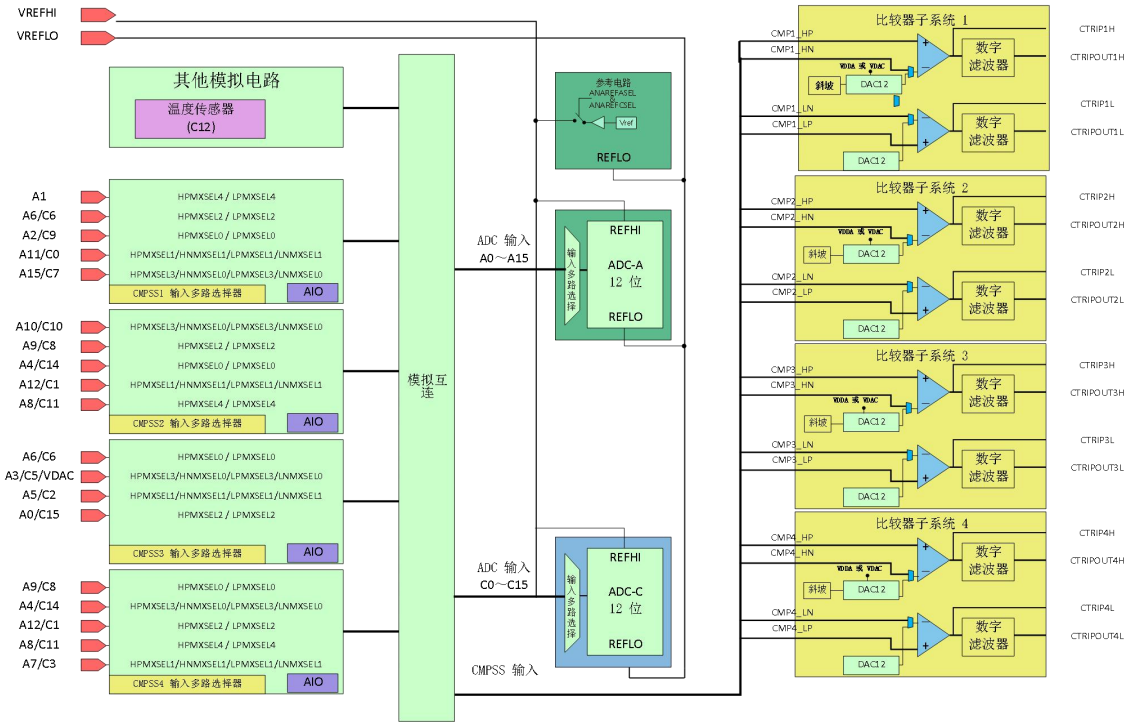
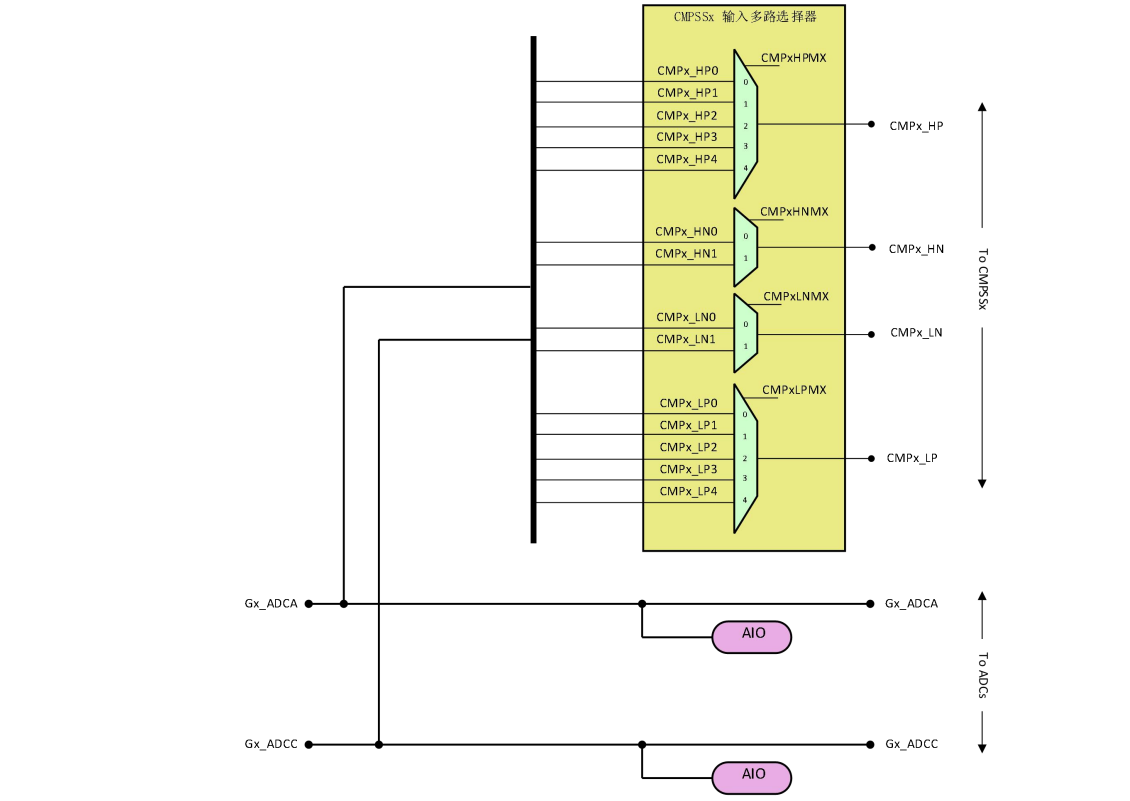



图 12-2 模拟子系统框图（48 引脚 QFN）




注: AIOs support digital input 模式 仅.

图 12-3 模拟组连接

## 12.2 优化上电时间

模数转换器（ADC）与缓冲型数模转换器（DAC）共享同一参考电路。当需要时，使用这些模块的应用程序可通过共享参考电路优化上电时间。当其中一个使用共享参考的模块在内部参考模式下完成初始化后，后续模块的上电时间可通过从最小上电时间要求中减去参考模块的上电时间来优化。

例如，若 ADCA 需要 tADCPUINT 模块在内部参考模式下启动，而 tADCPUEXT 模块在外部参考模式下启动，应用程序无需等待 tADCPUINT 模块启动第二个 ADC 实例（如 ADCC 在内部参考模式下）。此时，即使两个 ADC 均处于内部参考模式，应用程序启动 ADCC 后可直接等待 tADCPUEXT 模块。同理，若应用程序需使用 DACA 模块在内部参考模式下运行，则启动后的等待时间应为 tDACPUEXT，而非更长的 tDACPUINT。

此外，在 2.5V 与 3.3V 电压范围切换时，内部参考模式下启动也存在等待时间。具体等待时间参数请参阅设备数据手册。

## 12.3 ADC 引脚（AIO）的数字输入

部分 GPIO 与模拟引脚复用，仅具备数字输入功能，这类引脚也称为 AIO（模拟输入输出）。该端口仅支持 AIO 选项的引脚只能以输入模式工作，具体 AIO 信号列表请参阅设备数据手册。默认情况下，这些引脚作为模拟引脚工作，GPIO 处于高阻抗状态。通过 GPyAMSEL 寄存器可配置这些引脚的数字或模拟操作模式。

注意：若将具有尖锐边缘（高 dv/dt）的数字信号连接至 AIOs，可能会与相邻的模拟信号产生串扰。因此，当相邻通道用于模拟功能时，应限制连接至 AIOs 的信号边缘速率。

## 12.4 模数转换器引脚上的数字输入输出（AGPIOs）

部分 GPIO 与模拟引脚复用，具备数字输入输出功能，这类引脚也称为 AGPIO。与 AIO 不同，AGPIO 拥有完整的输入输出功能。

默认情况下 AGPIO 未连接，需进行配置。表中展示了 AGPIO 的配置方法：若需启用模拟功能，需从模拟子系统设置寄存器 AGPIOCTRLx；若需启用数字功能，则需从通用输入/输出（GPIO）章节设置寄存器 GPxAMSEL。

表 12-1 AGPIO 配置

| AGPIOCTRLx.GPIOy（默认值 = 0） | GPxAMSEL.GPIOy（默认值 = 1） | 引脚连接至 ADC | 引脚连接至 GPIOy |
|---|---|---|---|
| 0 | 0 | — | 是 |
| 0 | 1 | —(1) | —(1) |
| 1 | 0 | — | 是 |
| 1 | 1 | 是 | — |

(1) 默认情况下， AGPIO 引脚未连接任何信号。需从表格中选择其他行之一来配置引脚功能。

注意：若将具有尖锐边缘（高 dv/dt）的数字信号连接至 AGPIOs，可能会与相邻的模拟信号产生串扰。因此，当相邻通道用于模拟功能时，用户必须限制连接至 AGPIOs 的信号边缘速率。

## 12.5 模拟引脚与内部连接

表 12-2 模拟引脚与内部连接

封装引脚对应关系

| 引脚名称 | 64 QFP(5) | 48 QFP | 48 QFN | 32 QFN |
|---|---|---|---|---|
| VREFHI | 16 | 12 | 12 | —(4) |
| VREFLO | 17 | 13 | 13 | —(4) |
| 模拟组 1 | | | | |
| A6 | 6 | 4(1) | 4(1) | 2(1) |
| A2/C9 | 9 | 6 | 6 | 4 |
| A15/C7 | 10 | 7(1) | 7(1) | 5(1) |
| A11/C0 | 12 | 8 | 8 | 6(1) |
| A1 | 14 | 10 | 10 | 7(1) |
| 模拟组 2 | | | | |

| 引脚名称 | 64 QFP(5) | 48 QFP | 48 QFN | 32 QFN |
|---|---|---|---|---|
| A10/C10 | 25 | 21 | 21 | 13(1) |
| 模拟组 3 | | | | |
| C6 | 7 | 4(1) | 4(1) | 2(1) |
| A3/C5 | 8 | 5 | 5 | 3 |
| A14/C4 | 11 | 7(1) | 7(1) | 5(1) |
| A5/C2 | 13 | 9 | 9 | 6(1) |
| A0/C15/CMP1_DACL | 15 | 11 | 11 | 7(1) |
| 模拟组 4 | | | | |
| A7/C3 | 19 | 15 | 15 | 8(1) |
| 组合模拟组 2/4 | | | | |
| A12/C1 | 18 | 14 | 14 | 8(1) |
| A8/C11 | 20 | 16 | 16 | 9 |
| A4/C14 | 23 | 19 | 19 | 12 |
| A9/C8 | 24 | 20 | 20 | 13(1) |
| 其他模拟信号 | | | | |
| TempSensor(2) | — | — | — | — |
| A16/C16 | 2 | 2 | 2 | 32 |
| A17/C17 | 27 | — | — | — |
| A18/C18 | 28 | — | — | — |
| A19/C19 | 29 | 23 | 22 | — |
| A20/C20 | 30 | 24 | 23 | — |

模拟通道与 GPIO 对应关系

| 引脚名称 | ADC A | ADC C | DAC | AIO 输入 / GPIO |
|---|---|---|---|---|
| VREFHI | | | | |
| VREFLO | A13 | C13 | | |
| 模拟组 1 | | | | |
| A6 | A6 | — | | GPIO228(3) |
| A2/C9 | A2 | C9 | | GPIO224(3) |
| A15/C7 | A15 | C7 | | AIO233 |
| A11/C0 | A11 | C0 | | AIO237 |

| 引脚名称 | ADC A | ADC C | DAC | AIO 输入 / GPIO |
|---|---|---|---|---|
| A1 | A1 | — | | AIO232 |
| 模拟组 2 | | | | |
| A10/C10 | A10 | C10 | | GPIO230(3) |
| 模拟组 3 | | | | |
| C6 | — | C6 | | GPIO226(3) |
| A3/C5 | A3 | C5 | | GPIO242(3) |
| A14/C4 | A14 | C4 | | AIO239 |
| A5/C2 | A5 | C2 | | AIO244 |
| A0/C15/CMP1_DACL | A0 | C15 | CMP1_DACL | AIO231 |
| 模拟组 4 | | | | |
| A7/C3 | A7 | C3 | | AIO245 |
| 组合模拟组 2/4 | | | | |
| A12/C1 | A12 | C1 | | AIO238 |
| A8/C11 | A8 | C11 | | AIO241 |
| A4/C14 | A4 | C14 | | AIO225 |
| A9/C8 | A9 | C8 | | GPIO227(3) |
| 其他模拟信号 | | | | |
| TempSensor(2) | — | C12 | | |
| A16/C16 | A16 | C16 | | GPIO28(3) |
| A17/C17 | A17 | C17 | | GPIO20(3) |
| A18/C18 | A18 | C18 | | GPIO21(3) |
| A19/C19 | A19 | C19 | | GPIO13(3) |
| A20/C20 | A20 | C20 | | GPIO12(3) |

比较器输入对应关系

| 引脚名称 | 比较器高正输入 | 比较器高负输入 | 比较器低正输入 | 比较器低负输入 |
|---|---|---|---|---|
| VREFHI | | | | |
| VREFLO | | | | |
| 模拟组 1 | CMP1 | | | |

| 引脚名称 | 比较器高正输入 | 比较器高负输入 | 比较器低正输入 | 比较器低负输入 |
| --- | --- | --- | --- | --- |
| A6 | CMP1<br>(HPMXSEL=2) | | CMP1<br>(LPMXSEL=2) | |
| A2/C9 | CMP1<br>(HPMXSEL=0) | | CMP1<br>(LPMXSEL=0) | |
| A15/C7 | CMP1<br>(HPMXSEL=3) | CMP1<br>(HNMXSEL=0) | CMP1<br>(LPMXSEL=3) | CMP1<br>(LNMXSEL=0) |
| A11/C0 | CMP1<br>(HPMXSEL=1) | CMP1<br>(HNMXSEL=1) | CMP1<br>(LPMXSEL=1) | CMP1<br>(LNMXSEL=1) |
| A1 | CMP1<br>(HPMXSEL=4) | | CMP1<br>(LPMXSEL=4) | |
| 模拟组 2 | CMP2 | | | |
| A10/C10 | CMP2<br>(HPMXSEL=3) | CMP2<br>(HNMXSEL=0) | CMP2<br>(LPMXSEL=3) | CMP2<br>(LNMXSEL=0) |
| 模拟组 3 | CMP3 | | | |
| C6 | CMP3<br>(HPMXSEL=0) | | CMP3<br>(LPMXSEL=0) | |
| A3/C5 | CMP3<br>(HPMXSEL=3) | CMP3<br>(HNMXSEL=0) | CMP3<br>(LPMXSEL=3) | CMP3<br>(LNMXSEL=0) |
| A14/C4 | CMP3<br>(HPMXSEL=4) | | CMP3<br>(LPMXSEL=4) | |
| A5/C2 | CMP3<br>(HPMXSEL=1) | CMP3<br>(HNMXSEL=1) | CMP3<br>(LPMXSEL=1) | CMP3<br>(LNMXSEL=1) |
| A0/C15/CMP1_DACL | CMP3<br>(HPMXSEL=2) | | CMP3<br>(LPMXSEL=2) | |
| 模拟组 4 | CMP4 | | | |
| A7/C3 | CMP4<br>(HPMXSEL=1) | CMP4<br>(HNMXSEL=1) | CMP4<br>(LPMXSEL=1) | CMP4<br>(LNMXSEL=1) |
| 组合模拟组 2/4 | CMP2/4 | | | |
| A12/C1 | CMP2<br>(HPMXSEL=1)<br>CMP4<br>(HPMXSEL=2) | CMP2<br>(HNMXSEL=1) | CMP2<br>(LPMXSEL=1)<br>CMP4<br>(LPMXSEL=2) | CMP2<br>(LNMXSEL=1) |
| A8/C11 | CMP2<br>(HPMXSEL=4)<br>CMP4<br>(HPMXSEL=4) | | CMP2<br>(LPMXSEL=4)<br>CMP4<br>(LPMXSEL=4) | |
| A4/C14 | CMP2<br>(HPMXSEL=0) | CMP4<br>(HNMXSEL=0) | CMP2<br>(LPMXSEL=0) | CMP4<br>(LNMXSEL=0) |

| 引脚名称 | 比较器高正输入 | 比较器高负输入 | 比较器低正输入 | 比较器低负输入 |
|---|---|---|---|---|
| | CMP4<br>(HPMXSEL=3) | | CMP4<br>(LPMXSEL=3) | |
| A9/C8 | CMP2<br>(HPMXSEL=2)<br>CMP4<br>(HPMXSEL=0) | | CMP2<br>(LPMXSEL=2)<br>CMP4<br>(LPMXSEL=0) | |
| 其他模拟信号 | | | | |
| TempSensor(2) | CMP2<br>(HPMXSEL=5) | | | |
| A16/C16 | | | | |
| A17/C17 | | | | |
| A18/C18 | | | | |
| A19/C19 | | | | |
| A20/C20 | | | | |

(1)A6 和 C6 引脚为双键合引脚#4。

(2)可选外部参考电压，适用于片上 COMPDAC。无论该引脚用于 ADC 输入还是 COMPDAC 参考，均存在 VSSA 内部电容。若用作 VDAC 参考，需在该引脚上至少放置一个 1μF 电容。

(3)内部连接，不连接至设备引脚。

表 12-3 模拟信号描述

| 信号名称 | 说明 |
|---|---|
| AIOx | ADC 引脚上的数字输入 |
| Ax | ADC A 输入 |
| Cx | ADC C 输入 |
| CMPx_HNy | 比较器子系统高比较器负输入 |
| CMPx_HPy | 比较器子系统高比较器正输入 |
| CMPx_LNy | 比较器子系统低比较器负输入 |
| CMPx_LPy | 比较器子系统低比较器正输入 |
| CMPx_DACL | 较低 CMPSS DAC 的 DAC 输出（可引出至外部引脚） |
| TempSensor | 内部温度传感器 |

表 12-4 参考总结

| 模块 | 参考选项 | 配置位置 | 寄存器 | DriverLib 函数 | 备注 |
|---|---|---|---|---|---|
| ADC | 内部或外部 | 模拟系统 | AnalogSubsysRegs.<br>ANAREFCTL.bit.<br>ANAREFxSEL | ADC_setVREF | 两种选项均需使用 VREFHI 引脚。 |
| ADC | 3.3 V 或<br>2.5 V<br>内部参考范围 | 模拟系统 | AnalogSubsysRegs.<br>ANAREFCTL.bit.<br>ANAREFx2P5SEL | ADC_setVREF | 仅适用于内部参考模式。 |
| CMPSS DAC | VDDA | CMPSS 模块 | 不可配置 | | |

# 12.6 寄存器描述

本节阐述模拟子系统寄存器。

## 12.6.1 ASBSYS 基地址表

表 12-5 ASBSYS 基地址表

| 位域名称 | | DriverLib 名称 | 基地址 | 流水线保护 |
|---|---|---|---|---|
| 实例 | 结构 | | | |
| AnalogSubsysRegs | ANALOG_SUBSYS_REGS | ANALOGSUBSYS_BASE | 0x65_1000_65_127F | YES |

## 12.6.2 ANALOG_SUBSYS_REGS 寄存器

表中列出了 ANALOG_SUBSYS_REGS 寄存器的内存映射寄存器。表中未列出的所有寄存器偏移地址均应视为保留位置，且不得修改寄存器内容。

复杂位访问类型经编码后可适配于小型表格单元格。下表展示了本节中用于访问类型的编码方案。

表 12-6 ANALOG_SUBSYS_REGS 寄存器

| 地址偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0x48 | EXTROSCCSR1 | ExtR 振荡器状态寄存器 |
| 0xBC | CONFIGLOCK | 用于所有配置寄存器的锁存寄存器 |
| 0xC0 | TSNSCTL | 温度传感器控制寄存器 |
| 0xD0 | ANAREFCTL | 模拟参考控制寄存器 |
| 0xE0 | VMONCTL | 电压检测控制寄存器 |
| 0x104 | CMPHPMXSEL | 用于从 CopmHP 输入端口中的多个信号源中选择其中一个的位信号。 |
| 0x108 | CMPLPMXSEL | 用于从 CopmLP 输入端口中的多个信号源中选择其中一个的位信号。 |
| 0x10C | CMPHNMXSEL | 用于从 CopmHN 输入端口中的多个信号源中选择其中一个的位信号。 |
| 0x10E | CMPLNMXSEL | 用于从 CopmLN 输入端口中的多个信号源中选择其中一个的位信号。 |
| 0x110 | ADCDACLOOPBACK | 支持从 DAC 到 ADC 的环回测试 |
| 0x116 | CMPSSCTL | 锁存寄存器 |
| 0x11C | LOCK | 锁存寄存器 |
| 0x214 | AGPIOCTRLA | AGPIO 控制寄存器 |
| 0x230 | AGPIOCTRLH | AGPIO 控制寄存器 |

表 12-7 ANALOG SUBSYSRegs 访问类型代码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R-0 | 读 返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1S | W1S | 写 1 置位 |
| WSonce | WSonce | 一次性写（仅能置位一次） |
| 复位或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i,k,l,m,n | | 当这些变量用于寄存器名称、偏移地址或地址时，指代寄存器数组的值，该寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组以公式表示。 |
| y | | 当该变量用于寄存器名称、偏移地址或地址时，指代寄存器数组的值。 |

## 12.6.2.1 EXTROSCCSR1 寄存器

偏移地址：48h　复位值：0000h

EXTROSCCSR1 寄存器的各字段说明如表中所示。

表 12-8 EXTROSCCSR1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| OSCSTATUS | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | 保留 | 保留 | | 保留 | | | 保留 |
| R/W-0h | R/W-0h | R/W-0h | | R/W-0h | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | | | 保留 | | 保留 | 保留 |
| R/W-0h | R/W-0h | | | R/W-0h | | R/W-0h | R/W-0h |

表 12-9 EXTROSCCSR1 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-24 | OSCSTATUS | R | 0h | ExtR 的运行状态。ExtR 使能完成后，该值为 0xE7。0xE7：ExtR 使能成功完成。其他值：ExtR 使能尚未完成。复位类型：XRSn |
| 23-16 | RESERVED | R/W | 0h | 保留 |
| 15 | RESERVED | R/W | 0h | 保留 |
| 14 | RESERVED | R/W | 0h | 保留 |
| 13-12 | RESERVED | R/W | 0h | 保留 |
| 11-9 | RESERVED | R/W | 0h | 保留 |
| 8-7 | RESERVED | R/W | 0h | 保留 |
| 6-4 | RESERVED | R/W | 0h | 保留 |
| 3-2 | RESERVED | R/W | 0h | 保留 |
| 1 | RESERVED | R/W | 0h | 保留 |
| 0 | RESERVED | R/W | 0h | 保留 |

### 12.6.2.2 CONFIGLOCK 寄存器

偏移地址：BCh　复位值：0000h

CONFIGLOCK 寄存器的各字段说明如表中所示。

表 12-10 CONFIGLOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | 保留 | 保留 | AGPIOCTRL | 保留 | 保留 | 保留 |
| R-0-0h | | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |

表 12-11 CONFIGLOCK 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-6 | RESERVED | R-0 | 0h | 保留 |
| 5 | RESERVED | R/WSonce | 0h | 保留 |
| 4 | RESERVED | R/WSonce | 0h | 保留 |
| 3 | AGPIOCTRL | R/WSonce | 0h | 锁定所有 AGPIOCTRL 寄存器。设置该位后，将禁止后续对该寄存器进行任何写操作。该位只能通过复位清除。复位类型：SYSRSn |
| 2 | RESERVED | R/WSonce | 0h | 保留 |
| 1 | RESERVED | R/WSonce | 0h | 保留 |
| 0 | RESERVED | R/WSonce | 0h | 保留 |

### 12.6.2.3 TSNSCTL 寄存器

偏移地址：C0h　复位值：8800h

TSNSCTLK 寄存器的各字段说明如表中所示。

表 12-12 TSNSCTL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | 保留 | | | |
| R/W-8h | | | | R/W-8h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | ENABLE |
| R/W-0h | | | | | | | R/W-0h |

表 12-13 TSNSCTL 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R/W | 8h | 保留 |
| 11-8 | RESERVED | R/W | 8h | 保留 |
| 7-1 | RESERVED | R/W | 0h | 保留 |
| 0 | ENABLE | R/W | 0h | 温度传感器使能。该位用于使能温度传感器输出到 ADC。0：禁用1：使能复位类型：SYSRSn |

### 12.6.2.4 ANAREFCTL 寄存器

偏移地址：D0h　复位值：0007h

ANAREFCTL 寄存器的各字段说明如表中所示。

表 12-14 ANAREFCTL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| ANAREFSEL_SUP_OVERRIDE | 保留 | | | | 保留 | 保留 | ANAREFA2P5SEL |
| R/W-0h | R-0-0h | | | | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | 保留 | 保留 | ANAREFASEL |
| R-0-0h | | | | | R/W-1h | R/W-1h | R/W-1h |

表 12-15 ANAREFCTL 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | ANAREFSEL_SUP_OVERRIDE | R/W | 0h | 此配置用于确定（覆盖）ADC 的外部参考电压选择。1：ADC 使用 VDDA/VSSA 作为外部参考电压。在此模式下，ANAREFSEL 必须设置为 1，以选择外部参考模式。0：ADC 使用 VREFHI/VREFLO 作为参考引脚。在此模式下，ANAREFSEL 可以为 0（内部参考模式）或 1（外部参考模式）。复位类型：XRSn |
| 14-11 | RESERVED | R-0 | 0h | 保留 |
| 10 | RESERVED | R/W | 0h | 保留 |
| 9 | RESERVED | R/W | 0h | 保留 |
| 8 | ANAREFA2P5SEL | R/W | 0h | 模拟参考电压 2.5V 源选择。在内部参考模式下，该位用于选择内部参考缓冲器驱动到 VREFHI 引脚上的电压。缓冲器可以向该引脚提供 1.65V，此时参考范围为 0～3.3V；也可以提供 2.5V，此时参考范围为 0～2.5V。如果在这两种模式之间切换，必须等待足够的时间，使外部电容充电至新的电压，然后才能使用 ADC 或缓冲 DAC。0：内部 1.65V 参考 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 模式（3.3V 参考范围）1：内部 2.5V 参考模式（2.5V 参考范围）<br>复位类型：XRSn |
| 7-3 | RESERVED | R-0 | 0h | 保留 |
| 2 | RESERVED | R/W | 1h | 保留 |
| 1 | RESERVED | R/W | 1h | 保留 |
| 0 | ANAREFASEL | R/W | 1h | 模拟参考模式选择。该位用于选择 VREFHI 引脚使用内部参考模式还是外部参考模式。0：内部参考模式（芯片向 VREFHI 引脚驱动电压）1：外部参考模式（系统向 VREFHI 引脚输入电压）。复位类型：XRSn |

### 12.6.2.5 VMONCTL 寄存器

偏移地址：E0h　复位值：0000h

VMONCTL 寄存器的各字段说明如表中所示。

表 12-16 VMONCTL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | BORLVMONDIS |
| R-0-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | 保留 | |
| R-0-0h | | | | | | R/W-0h | |

表 12-17 VMONCTL 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-9 | RESERVED | R-0 | 0h | 保留 |
| 8 | BORLVMONDIS | R/W | 0h | VDDIO 上的 BORL 禁用控制。0：VDDIO 上的 BORL 使能。当 VDDIO 低于 VDDIO 的较低 BOR 阈值时，将触发 BOR 电路。1：VDDIO 上的 BORL 禁用。当 VDDIO 低于 VDDIO 的较低 BOR 阈值时，不会触发 BOR 电路。复位类型：SYSRSn |
| 7-2 | RESERVED | R-0 | 0h | 保留 |
| 1-0 | RESERVED | R/W | 0h | 保留 |

### 12.6.2.6 CMPHPMXSEL 寄存器

偏移地址：104h　复位值：0000h

CMPHPMXSEL 寄存器的各字段说明如表中所示。

表 12-18 CMPHPMXSEL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | 保留 | | | 保留 | | |
| R-0-0h | | R/W-0h | | | R/W-0h | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | 保留 | | | CMP4HPMXSEL | | | CMP3HPMXSEL |
| R-0-0h | R/W-0h | | | R/W-0h | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMP3HPMXSEL | | CMP2HPMXSEL | | | CMP1HPMXSEL | | |
| R/W-0h | | R/W-0h | | | R/W-0h | | |

表 12-19 CMPHPMXSEL 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-22 | RESERVED | R-0 | 0h | 保留 |
| 21-19 | RESERVED | R/W | 0h | 保留 |
| 18-16 | RESERVED | R/W | 0h | 保留 |
| 15 | RESERVED | R-0 | 0h | 保留 |
| 14-12 | RESERVED | R/W | 0h | 保留 |
| 11-9 | CMP4HPMXSEL | R/W | 0h | CMP4HPMXSEL 位。注意：仅 0～4 为有效值，其余值保留。复位类型：XRSn |
| 8-6 | CMP3HPMXSEL | R/W | 0h | CMP3HPMXSEL 位。注意：仅 0～4 为有效值，其余值保留。复位类型：XRSn |
| 5-3 | CMP2HPMXSEL | R/W | 0h | CMP2HPMXSEL 位。注意：仅 0～5 为有效值，其余值保留。复位类型：XRSn |
| 2-0 | CMP1HPMXSEL | R/W | 0h | CMP1HPMXSEL 位。注意：仅 0～4 为有效值，其余值保留。复位类型：XRSn |

## 12.6.2.7 CMPLPMXSEL 寄存器

偏移地址：108h　复位值：0000h

CMPLPMXSEL 寄存器的各字段说明如表中所示。

表 12-20 CMPLPMXSEL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | 保留 | | | 保留 | | |
| R-0-0h | | R/W-0h | | | R/W-0h | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | 保留 | | | CMP4LPMXSEL | | | CMP3LPMXSEL |
| R-0-0h | R/W-0h | | | R/W-0h | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMP3LPMXSEL | | CMP2LPMXSEL | | | CMP1LPMXSEL | | |
| R/W-0h | | R/W-0h | | | R/W-0h | | |

表 12-21 CMPLPMXSEL 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-22 | RESERVED | R-0 | 0h | 保留 |
| 21-19 | RESERVED | R/W | 0h | 保留 |
| 18-16 | RESERVED | R/W | 0h | 保留 |
| 15 | RESERVED | R-0 | 0h | 保留 |
| 14-12 | RESERVED | R/W | 0h | 保留 |
| 11-9 | CMP4LPMXSEL | R/W | 0h | CMP4LPMXSEL 位。注意：仅 0～4 为有效值，其余值保留。复位类型：XRSn |
| 8-6 | CMP3LPMXSEL | R/W | 0h | CMP3LPMXSEL 位。注意：仅 0～4 为有效值，其余值保留。复位类型：XRSn |
| 5-3 | CMP2LPMXSEL | R/W | 0h | CMP2LPMXSEL 位。注意：仅 0～5 为有效值，其余值保留。复位类型：XRSn |
| 2-0 | CMP1LPMXSEL | R/W | 0h | CMP1LPMXSEL 位。注意：仅 0～4 为有效值，其余值保留。复位类型：XRSn |

### 12.6.2.8 CMPHNMXSEL 寄存器

偏移地址：10Ch　复位值：0000h

CMPHNMXSEL 寄存器的各字段说明如表中所示。

表 12-22 CMPHNMXSEL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | 保留 | 保留 | CMP4HNMXSEL | CMP3HNMXSEL | CMP2HNMXSEL | CMP1HNMXSEL |
| R-0-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 12-23 CMPHNMXSEL 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-7 | RESERVED | R-0 | 0h | 保留 |
| 6 | RESERVED | R/W | 0h | 保留 |
| 5 | RESERVED | R/W | 0h | 保留 |
| 4 | RESERVED | R/W | 0h | 保留 |
| 3 | CMP4HNMXSEL | R/W | 0h | CMP4HNMXSEL 位。复位类型：XRSn |
| 2 | CMP3HNMXSEL | R/W | 0h | CMP3HNMXSEL 位。复位类型：XRSn |
| 1 | CMP2HNMXSEL | R/W | 0h | CMP2HNMXSEL 位。复位类型：XRSn |
| 0 | CMP1HNMXSEL | R/W | 0h | CMP1HNMXSEL 位。复位类型：XRSn |

### 12.6.2.9 CMPLNMXSEL 寄存器

偏移地址：10Eh　复位值：0000h

CMPLNMXSEL 寄存器的各字段说明如表中所示。

表 12-24 CMPLNMXSEL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | 保留 | 保留 | CMP4LNMXSEL | CMP3LNMXSEL | CMP2LNMXSEL | CMP1LNMXSEL |
| R-0-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 12-25 CMPLNMXSEL 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-7 | RESERVED | R-0 | 0h | 保留 |
| 6 | RESERVED | R/W | 0h | 保留 |
| 5 | RESERVED | R/W | 0h | 保留 |
| 4 | RESERVED | R/W | 0h | 保留 |
| 3 | CMP4LNMXSEL | R/W | 0h | CMP4LNMXSEL 位。复位类型：XRSn |
| 2 | CMP3LNMXSEL | R/W | 0h | CMP3LNMXSEL 位。复位类型：XRSn |
| 1 | CMP2LNMXSEL | R/W | 0h | CMP2LNMXSEL 位。复位类型：XRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 0 | CMP1LNMXSEL | R/W | 0h | CMP1LNMXSEL 位。复位类型：XRSn |

### 12.6.2.10 CMPSSCTL 寄存器

偏移地址：116h　复位值：0000h

CMPSSCTL 寄存器的各字段说明如表中所示。

表 12-26 CMPSSCTL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CMPSSCTLEN | 保留 | | | | | | |
| R/W-0h | R-0-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | CMP1LDACOUTEN |
| R-0-0h | | | | | | | R/W-0h |

表 12-27 CMPSSCTL 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | CMPSSCTLEN | R/W | 0h | CMPSS 控制使能位。0：该寄存器中的其余配置均被禁用。1：该寄存器中的其余配置均被使能。该位出于安全目的而添加。当该位为 0 时，该寄存器中的其他配置均不生效。复位类型：SYSRSn |
| 14-1 | RESERVED | R-0 | 0h | 保留 |
| 0 | CMP1LDACOUTEN | R/W | 0h | 0：CMPSS1.COMPL 使能，关联的 DAC 将作为比较器的参考电压。1：CMPSS1.COMPL 禁用，关联的 DAC 将作为通用 DAC 使用。复位类型：SYSRSn |

### 12.6.2.11 LOCK 寄存器

偏移地址：11Ch　复位值：0000h

LOCK 寄存器的各字段说明如表中所示。

表 12-28 LOCK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | CMPSSCTL | VREGCTL | CMPLNMXSEL |
| R-0-0h | | | | | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMPHNMXSEL | CMPLPMXSEL | CMPHPMXSEL | 保留 | 保留 | VMONCTL | ANAREFCTL | TSNSCTL |
| R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |

表 12-29 LOCK 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-11 | RESERVED | R-0 | 0h | 保留 |
| 10 | CMPSSCTL | R/WSonce | 0h | CMPSSCTL 寄存器锁定。设置该位后，将禁止后续对相应寄存器的任何写操作。该位只能通过复位清零。复位类型：SYSRSn |
| 9 | VREGCTL | R/WSonce | 0h | VREGCTL 寄存器锁定。设置该位后，将禁止后续对相应寄存器的任何写操作。该位只能通过复位清零。复位类型：SYSRSn |
| 8 | CMPLNMXSEL | R/WSonce | 0h | CMPLNMXSEL 寄存器锁定。设置该位后，将禁止后续对相应寄存器的任何写操作。该位只能通过复位清零。复位类型：SYSRSn |
| 7 | CMPHNMXSEL | R/WSonce | 0h | CMPHNMXSEL 寄存器锁定。设置该位后，将禁止后续对相应寄存器的任何写操作。该位只能通过复位清零。复位类型：SYSRSn |
| 6 | CMPLPMXSEL | R/WSonce | 0h | CMPLPMXSEL 寄存器锁定。设置该位后，将禁止后续对相应寄存器的任何写操作。该位只能通过复位清零。复位类型：SYSRSn |
| 5 | CMPHPMXSEL | R/WSonce | 0h | CMPHPMXSEL 寄存器锁定。设置该位后，将禁止后续对相应寄存器的任何写操作。该位只能通过复位清零。复位类型：SYSRSn |
| 4 | RESERVED | R/WSonce | 0h | 保留 |
| 3 | RESERVED | R/WSonce | 0h | 保留 |
| 2 | VMONCTL | R/WSonce | 0h | VMONCTL 寄存器锁定。设置该位后，将禁止后续对相应寄存器的任何写操作。该位只能通过复位 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 清零。复位类型：SYSRSn |
| 1 | ANAREFCTL | R/WSonce | 0h | ANAREFCTL 寄存器锁定。设置该位后，将禁止后续对相应寄存器的任何写操作。该位只能通过复位清零。复位类型：SYSRSn |
| 0 | TSNSCTL | R/WSonce | 0h | TSNSCTL 寄存器锁定。设置该位后，将禁止后续对相应寄存器的任何写操作。该位只能通过复位清零。复位类型：SYSRSn |

### 12.6.2.12 AGPIOCTRLA 寄存器

偏移地址：214h　复位值：0000h

AGPIOCTRLA 寄存器的各字段说明如表中所示。

表 12-30 AGPIOCTRLA 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | 保留 | 保留 | GPIO28 | 保留 | 保留 | 保留 | 保留 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | 保留 | GPIO21 | GPIO20 | 保留 | 保留 | 保留 | 保留 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | 保留 | GPIO13 | GPIO12 | 保留 | 保留 | 保留 | 保留 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 12-31 AGPIOCTRLA 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | RESERVED | R/W | 0h | 保留 |
| 30 | RESERVED | R/W | 0h | 保留 |
| 29 | RESERVED | R/W | 0h | 保留 |
| 28 | GPIO28 | R/W | 0h | GPIO28 一次性配置，用于确定是否使能 GPIO28 的 AGPIO 功能。0：禁用 AGPIO 功能1：使能 AGPIO 功能复位类型：XRSn |
| 27 | RESERVED | R/W | 0h | 保留 |
| 26 | RESERVED | R/W | 0h | 保留 |
| 25 | RESERVED | R/W | 0h | 保留 |
| 24 | RESERVED | R/W | 0h | 保留 |
| 23 | RESERVED | R/W | 0h | 保留 |
| 22 | RESERVED | R/W | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 21 | GPIO21 | R/W | 0h | GPIO21 一次性配置，用于确定是否使能 GPIO21 的 AGPIO 功能。0：禁用 AGPIO 功能1：使能 AGPIO 功能复位类型：XRSn |
| 20 | GPIO20 | R/W | 0h | GPIO20 一次性配置，用于确定是否使能 GPIO20 的 AGPIO 功能。0：禁用 AGPIO 功能1：使能 AGPIO 功能复位类型：XRSn |
| 19 | RESERVED | R/W | 0h | 保留 |
| 18 | RESERVED | R/W | 0h | 保留 |
| 17 | RESERVED | R/W | 0h | 保留 |
| 16 | RESERVED | R/W | 0h | 保留 |
| 15 | RESERVED | R/W | 0h | 保留 |
| 14 | RESERVED | R/W | 0h | 保留 |
| 13 | GPIO13 | R/W | 0h | GPIO13 一次性配置，用于确定是否使能 GPIO13 的 AGPIO 功能。0：禁用 AGPIO 功能1：使能 AGPIO 功能复位类型：XRSn |
| 12 | GPIO12 | R/W | 0h | GPIO12 一次性配置，用于确定是否使能 GPIO12 的 AGPIO 功能。0：禁用 AGPIO 功能1：使能 AGPIO 功能复位类型：XRSn |
| 11 | RESERVED | R/W | 0h | 保留 |
| 10 | RESERVED | R/W | 0h | 保留 |
| 9 | RESERVED | R/W | 0h | 保留 |
| 8 | RESERVED | R/W | 0h | 保留 |
| 7 | RESERVED | R/W | 0h | 保留 |
| 6 | RESERVED | R/W | 0h | 保留 |
| 5 | RESERVED | R/W | 0h | 保留 |
| 4 | RESERVED | R/W | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | RESERVED | R/W | 0h | 保留 |
| 1 | RESERVED | R/W | 0h | 保留 |
| 0 | RESERVED | R/W | 0h | 保留 |

### 12.6.2.13 AGPIOCTRLH 寄存器

偏移地址：230h　复位值：0000h

AGPIOCTRLH 寄存器的各字段说明如表中所示。

表 12-32 AGPIOCTRLH 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | 保留 | 保留 | 保留 | 保留 | GPIO242 | 保留 | 保留 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | GPIO230 | 保留 | GPIO228 | GPIO227 | GPIO226 | 保留 | GPIO224 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 12-33 AGPIOCTRLH 寄存器字段描述

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | RESERVED | R/W | 0h | 保留 |
| 30 | RESERVED | R/W | 0h | 保留 |
| 29 | RESERVED | R/W | 0h | 保留 |
| 28 | RESERVED | R/W | 0h | 保留 |
| 27 | RESERVED | R/W | 0h | 保留 |
| 26 | RESERVED | R/W | 0h | 保留 |
| 25 | RESERVED | R/W | 0h | 保留 |
| 24 | RESERVED | R/W | 0h | 保留 |
| 23 | RESERVED | R/W | 0h | 保留 |
| 22 | RESERVED | R/W | 0h | 保留 |
| 21 | RESERVED | R/W | 0h | 保留 |
| 20 | RESERVED | R/W | 0h | 保留 |
| 19 | RESERVED | R/W | 0h | 保留 |
| 18 | GPIO242 | R/W | 0h | GPIO242 一次性配置，用于确定是否使能 GPIO242 的 AGPIO 功能。0：禁用 AGPIO 功能 1：使能 AGPIO 功能复位类型：XRSn |
| 17 | RESERVED | R/W | 0h | 保留 |
| 16 | RESERVED | R/W | 0h | 保留 |
| 15 | RESERVED | R/W | 0h | 保留 |
| 14 | RESERVED | R/W | 0h | 保留 |
| 13 | RESERVED | R/W | 0h | 保留 |
| 12 | RESERVED | R/W | 0h | 保留 |
| 11 | RESERVED | R/W | 0h | 保留 |
| 10 | RESERVED | R/W | 0h | 保留 |
| 9 | RESERVED | R/W | 0h | 保留 |
| 8 | RESERVED | R/W | 0h | 保留 |
| 7 | RESERVED | R/W | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 6 | GPIO230 | R/W | 0h | GPIO230 一次性配置，用于确定是否使能 GPIO230 的 AGPIO 功能。0：禁用 AGPIO 功能 1：使能 AGPIO 功能复位类型：XRSn |
| 5 | RESERVED | R/W | 0h | 保留 |
| 4 | GPIO228 | R/W | 0h | GPIO228 一次性配置，用于确定是否使能 GPIO228 的 AGPIO 功能。0：禁用 AGPIO 功能 1：使能 AGPIO 功能复位类型：XRSn |
| 3 | GPIO227 | R/W | 0h | GPIO227 一次性配置，用于确定是否使能 GPIO227 的 AGPIO 功能。0：禁用 AGPIO 功能 1：使能 AGPIO 功能复位类型：XRSn |
| 2 | GPIO226 | R/W | 0h | GPIO226 一次性配置，用于确定是否使能 GPIO226 的 AGPIO 功能。0：禁用 AGPIO 功能 1：使能 AGPIO 功能复位类型：XRSn |
| 1 | RESERVED | R/W | 0h | 保留 |
| 0 | GPIO224 | R/W | 0h | GPIO224 一次性配置，用于确定是否使能 GPIO224 的 AGPIO 功能。0：禁用 AGPIO 功能 1：使能 AGPIO 功能复位类型：XRSn |

# 13 模数转换器 （ADC）

## 13.1 简介

ADC 模块是一个 12 位（SAR）ADC。ADC 模块中包含数字控制和模拟功能。模拟功能包含通道选择多路复用器（MUX）、采样保持（S/H）电路、逐次逼近电路、电压基准电路和其他模拟支持电路。数字控制包含可编程转换逻辑、结果寄存器、与模拟电路的接口、与外设总线的接口、后处理电路以及与其他片上模块的接口等。

## 13.2 主要特性

每个 ADC 均具备以下特性：

- 12 位分辨率
- 由 VREFHI 和 VREFLO 引脚设定的比率式外部参考电压
- 可选的 2.5V 或 3.3V 内部参考电压
- 单端信号转换
- 支持最多 21 个通道的输入多路复用器
- 16 个可配置的 SOC
- 16 个独立可寻址的结果寄存器
- 多种触发源：
    1) 软件触发（S/W）
    2) 软件立即启动
    3) 所有 ePWM 模块
    4) ADCSOC A 或 B 触发
- GPIO XINT2（外部中断 2）
- CPU 定时器 0/1/2
- ADCINT1/2（ADC 中断 1/2）
- 四路灵活的 CLIC 中断

- 可配置的中断分配突发模式
- 四个后置处理模块，每个模块包含：
  1) 饱和偏移校准
  2) 与设定值的偏差计算
  3) 高 / 低阈值及过零比较（具备中断和 ePWM 跳闸触发能力）
  4) 触发至采样的延迟捕获

## 13.3 功能描述

展示了 ADC 核心与 ADC 封装模块的框图。

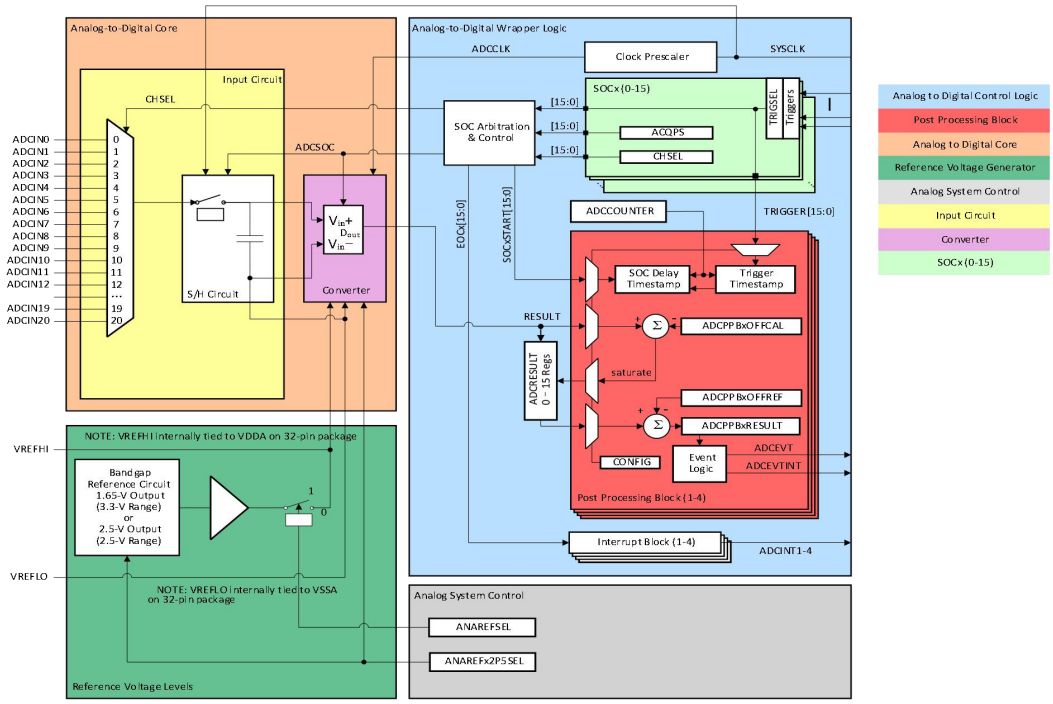



图 13-1 adc 模块框图

ADC 框图反映了器件内部可配置的 ADC 通道数量。实际可用的外部 ADC 输入通道数则因具体器件型号及封装类型而异。

ADC 可配置性：

部分 ADC 配置由 SOC 独立控制，而其余配置则按 ADC 模块进行全局控制。表汇总了 ADC 的基本配置选项及其可配置程度。后续章节将对这些配置进行详细讨论。

表 13-1 ADC 选项与配置级别

| 选择 | 可配置性 |
| --- | --- |
| Clock | 每个模块（1） |
| Resolution | 不可配置（仅支持 12 位） |
| Single mode | 不可配置（仅支持单端） |
| 基准电压源 | 所有模块均支持外部或内部连接 |
| 触发源 | 每个SOC(1) |
| 转换通道 | 每个SOC |
| 采样窗口 | 每个SOC(1) |
| EOC位置 | 每个模块 |
| 突发模式 | 每个模块(1) |

(1) 若以不同方式向不同的 ADC 模块写入这些值，可能会导致 ADC 以异步方式运行。

### 13.3.1 时钟配置

基准 ADC 时钟直接由系统时钟（SYSCLK）提供。SYSCLK 用于生成 ADC 采集窗口。寄存器 ADCCTL2 有一个 PRESCALE 字段，用于确定 ADCCLK。ADCCLK 用于为转换器提供时钟，并且仅在转换阶段有效。在所有其他时间，包括采样保持窗口期间，ADCCLK 信号均被关闭。大约需要 10.5 个 ADCCLK 周期才能将电压处理成转换结果。

为确定 ADCCTL2 的合适预置值，请参阅器件数据手册以获取最大 SYSCLK 与 ADCCLK 频率。

### 13.3.2 分辨率

ADC 的分辨率决定了模拟信号被量化为数字值的精细程度。每个 ADC 模块支持 12 位固定分辨率。

### 13.3.3 参考电压

#### 13.3.3.1 外部参考模式

使用外部参考时，外部向 VREFHI 和 VREFLO 输入电压。使用内部参考时，内部参考电路将产生的电压从管脚 VREFHI 输出，此时通过配置选择 1.65v 或 2.5v 的内部参考电压。

- 对于没有外部 VREFHI 和 VREFLO 引脚的器件， VREFHI 引脚在内部与 VDDA 引脚相连，而 VREFLO 引脚则内部连接到器件的模拟地线 VSSA 。对于缺少 VREFHI 和 VREFLO 引脚的封装，不支持内部参考模式。有关具备 VREFHI 和 VREFLO 引脚的封装信息，请参阅器件数据表。
- 请查阅器件数据表，以获取 VREFHI 和 VREFLO 引脚的允许电压范围。
- 外部参考模式需要在 VREFHI 引脚上连接一个外部电容器。具体所需电容值，请参阅器件数据表。

#### 13.3.3.2 内部参考模式

在内部参考模式下，器件将电压驱动至 VREFHI 引脚。随后， VREFHI 和 VREFLO 引脚设定 ADC 转换范围。对于不具备 VREFHI 和 VREFLO 引脚的封装，则无法使用内部参考模式。有关支持 VREFHI 和 VREFLO 引脚的封装信息，请参阅器件数据手册。内部参考电压可配置为 2.5V 或 1.65V。当选择 1.65V 内部参考电压时，ADC 输入信号在转换前会被内部除以 2，从而将 ADC 转换范围有效设定为从 VREFLO 到 3.3V。

内部参考模式还需要在 VREFHI 引脚上连接一个外部电容器。具体所需电容值请参阅器件数据手册。

#### 13.3.3.3 选择参考模式

电压参考模式必须通过使用 ADC_setVREF() 或 SetVREF() 函数进行配置，具体选择取决于所采用的头文件。使用任一上述函数均可确保将正确的微调值加载至 ADC 微调寄存器中。该函数必须在器件复位后至少调用一次。禁止通过直接向 anarefctl 寄存器写入来配置电压参考模式。

### 13.3.4 信号模式

ADC 支持单端信号传输。在单端模式下，转换器的输入电压通过单个引脚（ADCINx）进行采样，该引脚以 VREFLO 为参考电平。

### 13.3.5 电压转换关系

对于给定的模拟输入电压，预期的数字转换结果见下表。小数部分均取整。

表 13-2 模拟信号与 12 位数字信号的转换公式

| 模拟输入信号 | 转换后的12位数值 |
|---|---|
| ADCINy ≤ VREFLO | ADCRESULTx = 0 |
| VREFLO < ADCINy < VREFHI | ADCRESULTx = 4096 x $(\frac{\text{ADCINy} - \text{VREFLO}}{\text{VREFHI} - \text{VREFLO}})$ |
| ADCINy ≥ VREFHI | ADCRESULTx = 4095 |

根据给定的 ADC 转换结果，对应的模拟输入值如下表所示。该值对应于可产生该转换结果的模拟电压可能范围的中心点。

表 13-3 12 位数模转换公式

| 数字值 | 模拟等效 |
|---|---|
| When ADCRESULTy = 0 | ADCINx ≤ VREFLO |
| When 0 < ADCRESULTy < 4095 | ADCINx = (VREFHI − VREFLO)$(\frac{\text{ADCRESULTy}}{4096})$ +VREFLO |
| When ADCRESULTy = 4095 | ADCINx ≥ VREFHI |

### 13.3.6 SOC 工作原理

ADC 触发与转换序列化的实现依赖于可配置的转换起始设置（SOCs）。每个 SOC 都是一个配置集，用于定义单个通道的单次转换流程。该配置集包含三个参

数：启动转换的触发源、待转换通道以及采集（采样）窗口时长。当接收到针对某 SOC 配置的触发信号时，封装器将确保使用指定采集窗口时长对指定通道进行采集。针对相同的触发信号、通道及采集窗口时长，可配置多个 SOC。通过配置多个 SOC 使用同一触发信号，可生成一系列连续转换；若配置多个 SOC 使用同一触发信号及同一通道，则可实现过采样功能。

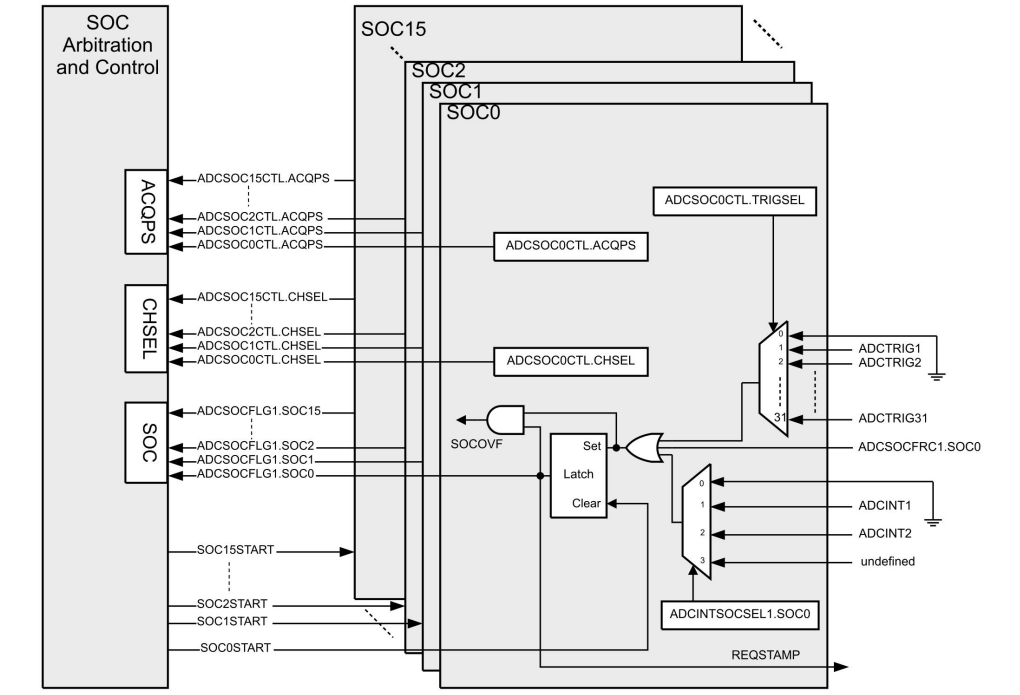



图 13-2 SOC 框图

### 13.3.6.1 SOC 工作原理

每个 SOC 均配备一个配置寄存器 ADCSOCxCTL。通过该寄存器，可对触发源、待转换通道及采样窗口时长进行配置。

### 13.3.6.2 触发源

每个 SOC 均可配置为响应多种输入触发信号中的任一触发条件。SOCx 的主要触发选择功能位于 ADCSOCxCTL. trigsel 寄存器中，该寄存器支持以下触发源：

- 禁用模式（仅限软件控制）
- CPU 定时器 0/1/2

- GPIO ：输入 X-Bar INPUT5
- 各 ePWM 模块的 adcsoca 或 adcsocb 信号。

此外，当 ADCINT1 或 ADCINT2 标志位被置位时，SOC 也可触发运行。此功能通过配置 ADCINTSOCSEL1 寄存器（适用于 SOC0 至 SOC7）或 ADCINTSOCSEL2 寄存器（适用于 SOC8 至 SOC15）实现。该机制可有效支持连续转换操作的实现。

表 13-4 adc SOC 触发源

| TRIGSEL | 触发源 |
|---|---|
| 0h | 软件触发 |
| 1h | CPU1_TINT0 |
| 2h | CPU1_TINT1 |
| 3h | CPU1_TINT2 |
| 4h | INPUTXBAR5 |
| 5h | EPWM1_ADCSOCA |
| 6h | EPWM1_ADCSOCB |
| 7h | EPWM2_ADCSOCA |
| 8h | EPWM2_ADCSOCB |
| 9h | EPWM3_ADCSOCA |
| Ah | EPWM3_ADCSOCB |
| Bh | EPWM4_ADCSOCA |
| Ch | EPWM4_ADCSOCB |
| Dh | EPWM5_ADCSOCA |
| Eh | EPWM5_ADCSOCB |
| Fh | EPWM6_ADCSOCA |
| 10h | EPWM6_ADCSOCB |
| 11h | EPWM7_ADCSOCA |
| 12h | EPWM7_ADCSOCB |
| 13h-3Fh | Reserved |

### 13.3.6.3 ADC 采样窗口

外部信号源在快速、有效驱动模拟信号的能力上存在差异。为达到额定分辨率，信号源需要将 ADC 内核中的采样电容充电至信号电压的 0.5 LSB 以内。

采集窗口是允许采样电容进行充电的时间长度，可通过 ADCSOCxCTL.ACQPS 寄存器对每个 SOCx 进行配置。

ACQPS 是一个 9 位寄存器，可配置为 0～511 之间的数值，对应的采集窗口时长为：采集窗口时长 =(ACQPS + 1) × 系统时钟 (SYSCLK) 周期

采集窗口时长基于系统时钟 (SYSCLK)，而非 ADC 时钟 (ADCCLK)。

所选采集窗口时长至少需要大于等于一个 ADCCLK 周期。

### 13.3.6.4 ADC 输入模式

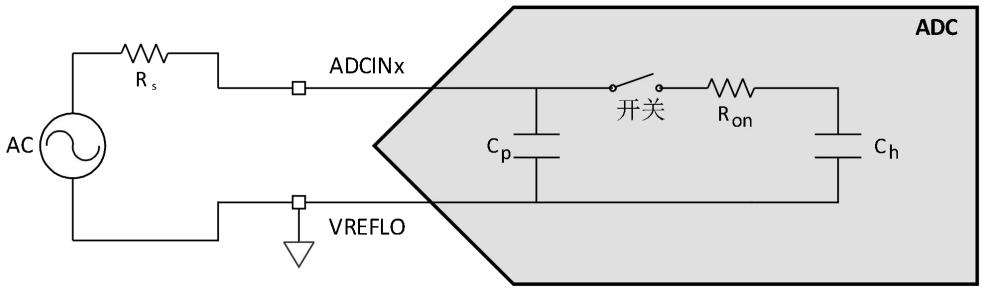



图 13-3 adc 单端工作模式模型

### 13.3.6.5 通道选择

每个 SOC 均可配置为转换任意 ADC 通道。该行为由 ADCSOCxCTL CHSEL 寄存器为 SOCx 选择。

表 13-5 adc 输入引脚的通道选择

| CHSEL | 输入 |
|---|---|
| 0 | ADCIN0 |
| 1 | ADCIN1 |
| 2 | ADCIN2 |
| 3 | ADCIN3 |
| 4 | ADCIN4 |
| 5 | ADCIN5 |
| 6 | ADCIN6 |
| 7 | ADCIN7 |
| 8 | ADCIN8 |
| 9 | ADCIN9 |
| 10 | ADCIN10 |
| 11 | ADCIN11 |
| 12 | ADCIN12 |
| 13 | ADCIN13 |
| 14 | ADCIN14 |
| 15 | ADCIN15 |
| 16 | ADCIN16 |
| 17 | ADCIN17 |
| 18 | ADCIN18 |
| 19 | ADCIN19 |
| 20 | ADCIN20 |

### 13.3.7 SOC 配置示例

#### 13.3.7.1 EPWM 触发信号转换

要配置 ADCA ，使其在 ePWM 定时器达到周期匹配时仅对通道 ADCIN1 执行一次转换，需要完成以下几项工作。首先，必须配置 ePWM3 以生成 SOCA 或 SOCB 信号（此处，SOC 指的是 ePWM 模块中的信号）。具体方法请参阅《增强型脉宽调制器模块（ePWM）》章节。假设选择了 SOCB 信号，则任意选择 SOC5。16 个 SOC 中的任意一个均可使用。假设需要 100ns 的采样窗口，且 SYSCLK 频率为 120MHz，则采集窗口持续时间应为 100ns/8.333ns = 12 个周期。 ACQPS 字段必须设置为 12 - 1 = 11。

AdcaRegs.ADCSOC5CTL.bit. CHSEL = 1； //SOC5 将 ADCINA1 转换为

AdcaRegs.ADCSOC5CTL.bit. ACQPS = 11； //SOC5 使用 12 SYSCLK 周期的采样时间

AdcaRegs.ADCSOC5CTL.bit. trigsel = 10； //SOC5 在 ePWM3 SOCB 上开始转换

按照当前配置：当 ePWM3 发生周期匹配并产生 SOCB 信号时，如果 ADC 处于空闲状态，将立即开始对 ADCINA1 通道（SOC5）进行采样。如果 ADC 正忙，则当 SOC5 获得优先级后，开始对 ADCINA1 采样。

ADC 控制逻辑会按照设定的 100ns 采集窗口宽度对 ADCINA1 进行采样。采样完成后，ADC 立即将采样得到的电压转换为数字量。转换完成后，结果将存入 ADCRESULT5 寄存器。

#### 13.3.7.2 基于 ePWM 触发器的过采样转换

为将 ADC 配置为对 ADCINA1 进行 4 倍过采样，我们采用与前例相同的配置方案，但将其应用于 SOC5、SOC6、SOC7 和 SOC8。

AdcaRegs.ADCSOC5CTL.bit.CHSEL = 1; //SOC5 converts ADCINA1

AdcaRegs.ADCSOC5CTL.bit.ACQPS = 9; //SOC5 uses a sample duration of 10 SYSCLK cycles

AdcaRegs.ADCSOC5CTL.bit.TRIGSEL = 10; //SOC5 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC6CTL.bit.CHSEL = 1; //SOC6 converts ADCINA1

AdcaRegs.ADCSOC6CTL.bit.ACQPS = 9; //SOC6 uses a sample duration of 10 SYSCLK cycles

AdcaRegs.ADCSOC6CTL.bit.TRIGSEL = 10; //SOC6 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC7CTL.bit.CHSEL = 1; //SOC7 converts ADCINA1

AdcaRegs.ADCSOC7CTL.bit.ACQPS = 9; //SOC7 uses a sample duration of 10 SYSCLK cycles

AdcaRegs.ADCSOC7CTL.bit.TRIGSEL = 10; //SOC7 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC8CTL.bit.CHSEL = 1; //SOC8 converts ADCINA1

AdcaRegs.ADCSOC8CTL.bit.ACQPS = 9; //SOC8 uses a sample duration of 10 SYSCLK cycles

AdcaRegs.ADCSOC8CTL.bit.TRIGSEL = 10; //SOC8 begins conversion on ePWM3 SOCB

按照配置，当 ePWM3 发生周期匹配并产生 SOCB 信号时：如果 ADC 处于空闲状态，将立即开始对 ADCINA1 通道（SOC5）进行采样。如果 ADC 正忙，则在 SOC5 获得优先级后开始对 ADCINA1 采样。

SOC5 转换完成后，SOC6 便开始对 ADCINA1 进行转换，同时 SOC5 的结果存入 ADCRESULT5 寄存器。最终，4 路转换将依次完成，结果分别存入对应寄存器：

SOC5 → ADCRESULT5

SOC6 → ADCRESULT6

SOC7 → ADCRESULT7

SOC8 → ADCRESULT8

### 13.3.7.3 CPU 定时器触发器的多次转换

本示例展示如何对具有不同采集窗口要求的多路信号进行采样。示例中使用 CPU1 定时器 2 来产生触发信号。

在设计包含多路信号的采样方案时，一个良好的设计步骤是：先列出所有信号及其所需采集窗口，然后据此计算每个信号需要的 SYSCLK 周期数，进而得到 ACQPS 寄存器的配置值。下表给出了具体示例，其中假设系统时钟 SYSCLK 为 150MHz（时钟周期为 6.667ns）。

表 13-6 多信号采样示例要求

| 信号名 | 采样窗口要求 | 获取窗口系统时钟周期数 | ACQPS 寄存器值 |
|---|---|---|---|
| Signal 1 | >200ns | 200/6.667 | 30-1=29 |
| Signal 2 | >740ns | 740/6.667 | 111-1=110 |
| Signal 3 | >183.33ns | 183.33/6.667 | 28-1=27 |
| Signal 4 | >485ns | 485/6.667 | 73-1=72 |

接下来确定每个信号应连接的 ADC 引脚

表 13-7 多信号采样中的连接实例

| 信号名 | ADC引脚 | CHSEL值 |
|---|---|---|
| Signal 1 | ADCIN5 | 5 |
| Signal 2 | ADCIN0 | 0 |
| Signal 3 | ADCIN3 | 3 |
| Signal 4 | ADCIN2 | 2 |

根据已整理的信息，生成 SOC 配置方案：

AdcaRegs.ADCSOC0CTL.bit.CHSEL = 5; //SOC0 converts ADCINA5

AdcaRegs.ADCSOC0CTL.bit.ACQPS = 29; //SOC0 uses a sample duration of 24 SYSCLK cycles

AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 3; //SOC0 begins conversion on CPU1 Timer 2

AdcaRegs.ADCSOC1CTL.bit.CHSEL = 0; //SOC1 converts ADCINA0

AdcaRegs.ADCSOC1CTL.bit.ACQPS = 110; //SOC1 uses a sample duration of 89 SYSCLK cycles

AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 3; //SOC1 begins conversion on CPU1 Timer 2

AdcaRegs.ADCSOC2CTL.bit.CHSEL = 3; //SOC2 converts ADCINA3

AdcaRegs.ADCSOC2CTL.bit.ACQPS = 27; //SOC2 uses a sample duration of 22 SYSCLK cycles

AdcaRegs.ADCSOC2CTL.bit.TRIGSEL = 3; //SOC2 begins conversion on CPU1 Timer 2

AdcaRegs.ADCSOC3CTL.bit.CHSEL = 2; //SOC3 converts ADCINA2

AdcaRegs.ADCSOC3CTL.bit.ACQPS = 72; //SOC3 uses a sample duration of 59 SYSCLK cycles

AdcaRegs.ADCSOC3CTL.bit.TRIGSEL = 3; //SOC3 begins conversion on CPU1 Timer 2

按照配置，当 CPU1 定时器 2 产生触发事件时，SOC0、SOC1、SOC2 和 SOC3 将按此顺序依次完成采样与转换。ACINA5（信号 1）的转换结果存放在 ADCRESULT0 中。同理，ADCINA0（信号 2）、ADCINA3（信号 3）和 ADCINA2（信号 4）的结果分别存放在 ADCRESULT1、ADCRESULT2 和 ADCRESULT3 中。

### 13.3.7.4 SOC 的软件触发配置

无论 SOC 是否已配置为接受特定触发信号，软件触发机制均可设定 SOC 的转换参数。该过程通过向 ADCSOCFRC1 寄存器写入位值实现。对于前例所述场景，若无需等待 CPU1 定时器 2 生成触发信号，可通过以下语句完成软件触发操作：

AdcaRegs.ADCSOCFRC1.all = 0x000F; //set SOC flags for SOC0 to SOC3

### 13.3.8 ADC 转换优先级

当多个 SOC 标志位同时被设置时，将根据以下两种优先级方式之一确定转换顺序。默认优先级方式为轮询机制。在该机制中，任何单个 SOC 均不天然拥有高于其他 SOC 的优先级；优先级取决于轮询指针（rrpointer）。在 ADCSOCPRICTL 寄存器中体现的 rrpointer 指向最近被转换的 SOC。具有最高优先级的 SOC 被分配给大于 rrpointer 值的下一个 SOC，若该值超过第 15 个 SOC，则循环回退至 SOC0。复位时，rrpointer 的值为 16——因为值 0 表示已进行过转换。当 rrpointer 等于 16 时，最高优先级将分配给 SOC0。rrpointer 将在 ADC 模块复位时，或当将复位值写入 ADCSOCPRICTL 寄存器时被重置。对 ADC 模块进行复位的操作是：针对对应的 ADC 实例写入并清除 SOFTPRES 位。

ADCSOCPRICTL 寄存器中的 SOCPRIORITY 字段可用于将高优先级从单个 SOC 分配至所有 SOC。当配置为高优先级时，任一当前转换完成后，SOC 将中断轮询轮转，并插入作为下一个转换。转换完成后，轮询轮转将从转换被中断的位置继续执行。若同时触发两个高优先级 SOC，则编号较小的 SOC 具有优先权。高优先级模式首先分配给 SOC0，随后按数字递增顺序依次分配。写入 SOCPRIORITY 字段的值定义了首个非高优先级 SOC。换言之，若将值 4 写入 SOCPRIORITY 字段，则 SOC0、SOC1、SOC2 及 SOC3 均被定义为高优先级，其中 SOC0 的优先级最高。

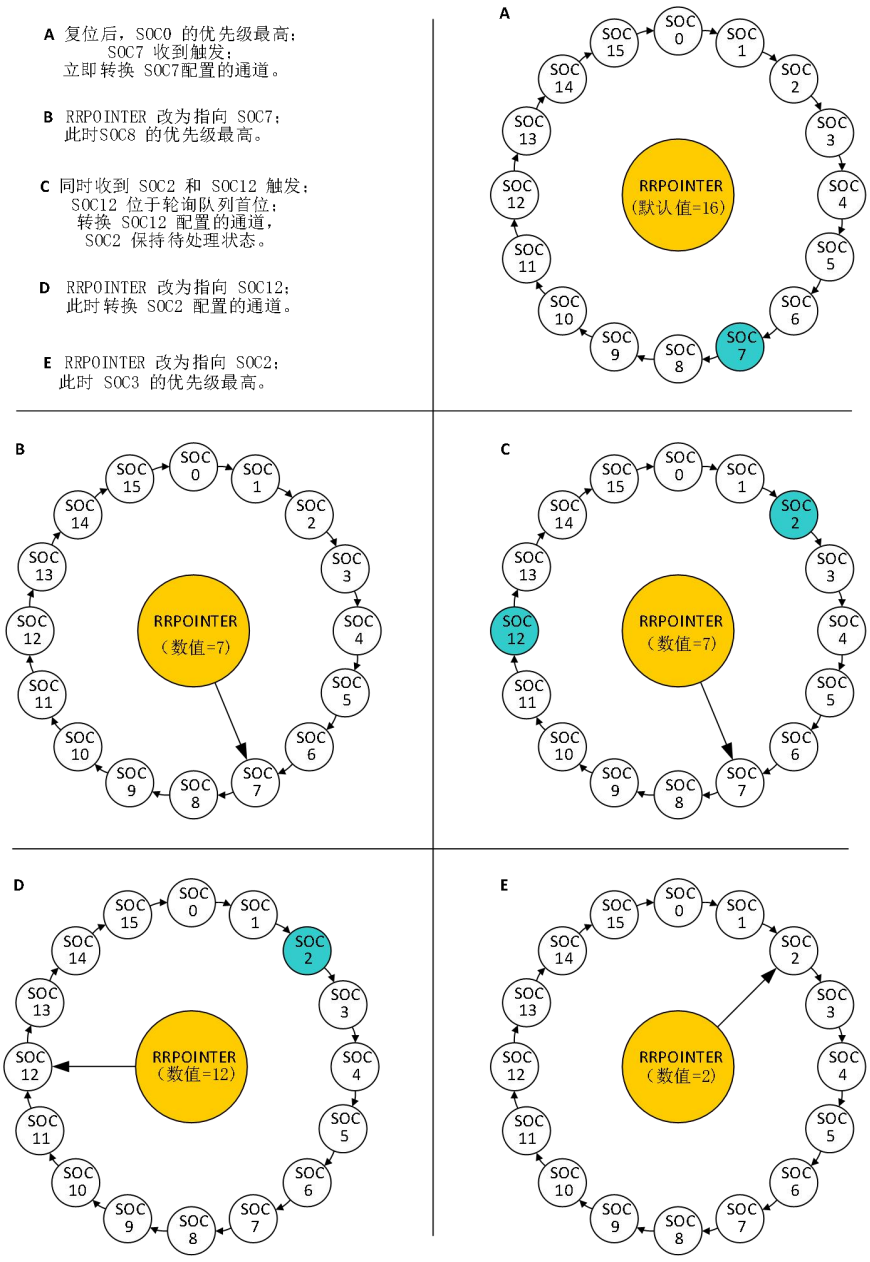



图 13-4 轮询优先级示例

ADCSOCPRICTL 寄存器中的 SOCPRIORITY 字段可用于将高优先级分配给单个至全部 SOC。当某个 SOC 被配置为高优先级时，它会在当前任意一次转换完成后中断轮询机制，并作为下一个待转换对象插入执行。该高优先级转换完成后，轮询机制将从被中断的位置继续执行。

如果两个高优先级 SOC 同时被触发，则编号较小的 SOC 优先。

高优先级模式按如下规则分配：优先分配给 SOC0，然后按编号递增顺序依次分配。写入 SOCPRIORITY 字段的值，定义了第一个不属于高优先级的 SOC。换言之：若向 SOCPRIORITY 写入数值 4，则 SOC0、SOC1、SOC2、SOC3 被定义为高优先级，其中 SOC0 优先级最高。

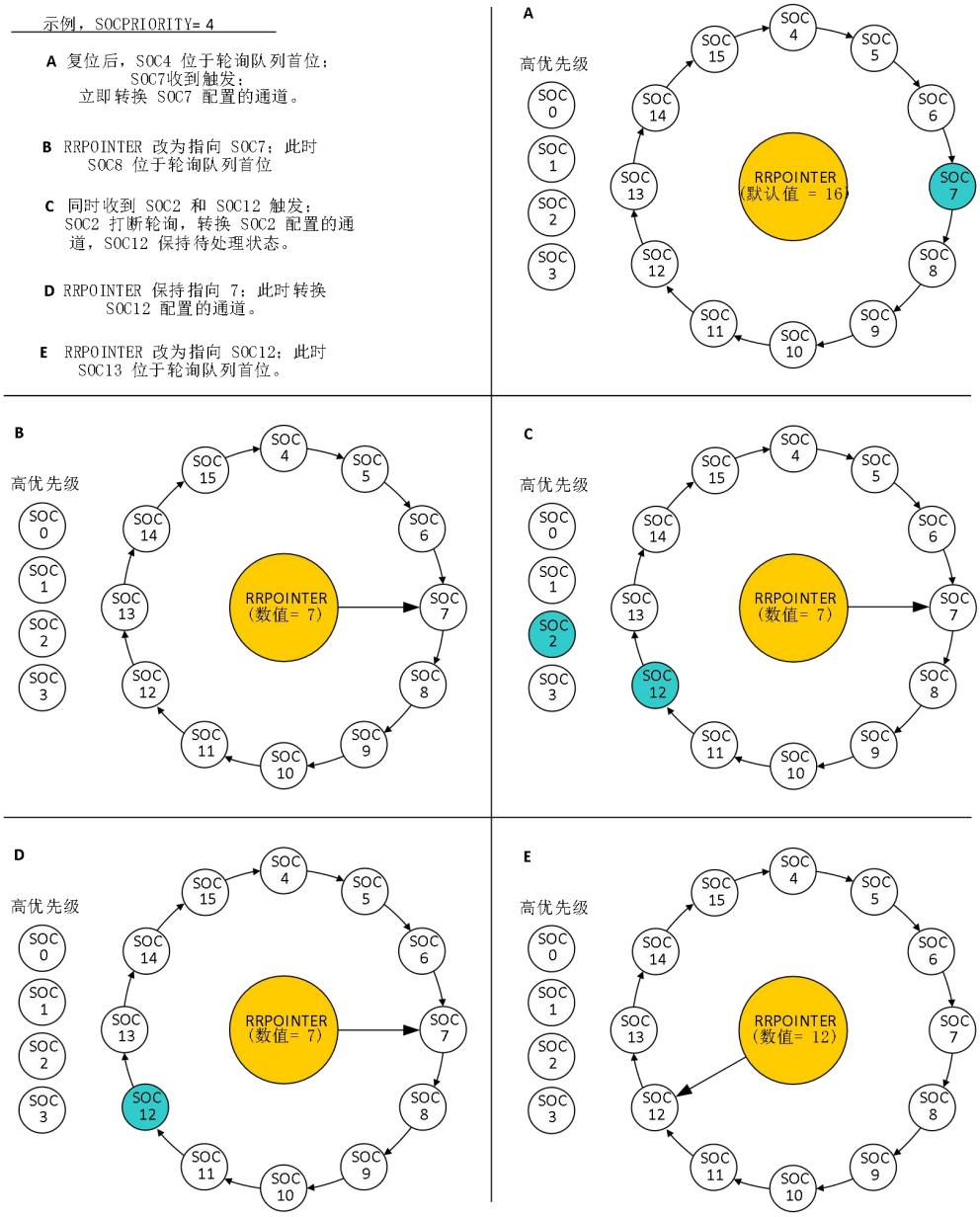



图 13-5 高优先级示例

### 13.3.9 ADC 突发模式

突发模式（Burst mode） 允许单个触发信号按照轮询顺序，一次遍历一个或多个 SOC。将 ADCBURSTCTL 寄存器中的 BURSTEN 位置 1，即可将 ADC 配置为突发模式。这会使得 TRIGSEL 字段被忽略，但仅对配置为轮询模式的 SOC 有效（高优先级 SOC 不受影响）。

所有轮询模式的 SOC 不再由 TRIGSEL 字段触发，而是根据 ADCBURSTCTL 寄存器中的 BURSTTRIG 字段进行触发。

当接收到突发触发信号时，ADC 封装层不会将所有轮询 SOC 都置为待转换，而仅转换(ADCBURSTCTL.BURSTSIZE + 1) 个 SOC。

第一个被选中的 SOC 是根据轮询指针优先级最高的那个，随后依次选择后续 SOC，直到总共选中 BURSTSIZE 个 SOC 为止。

在将 ADC 配置为突发模式时，用户必须确保当前一轮突发转换已完成，再接收下一轮突发触发信号。(ADCBURSTCTL.BURSTSIZE + 1) 的取值必须小于或等于配置为轮询优先级的 SOC 数量。

例如：若 SOCPRIORITY = 12，即 SOC12、SOC13、SOC14、SOC15 工作在轮询模式下，则为保证突发模式正常工作，ADCBURSTCTL.BURSTSIZE 的配置值必须 ≤ 3。

#### 13.3.9.1 突发模式配置示例

突发模式可用于在交替触发时采样不同的信号组：

```
AdcaRegs.BURSTCTL.BURSTEN = 1; //Enable ADC burst mode
AdcaRegs.BURSTCTL.BURSTTRIG = 3; //CPU1 Timer 2 triggers burst of conversions
AdcaRegs.BURSTCTL.BURSTSIZE = 1; //conversion bursts are 1 + 1 = 2 conversions long
AdcaRegs.SOCPRICTL.bit.SOCPRIORITY = 12; //SOC0 to SOC11 are high priority
AdcaRegs.ADCSOC12CTL.bit.CHSEL = 7; //SOC12 converts ADCINA7
```

AdcaRegs.ADCSOC12CTL.bit.ACQPS = 19; //SOC12 uses sample duration of 20 SYSCLK cycles

AdcaRegs.ADCSOC13CTL.bit.CHSEL = 5; //SOC13 converts ADCINA5

AdcaRegs.ADCSOC13CTL.bit.ACQPS = 19; //SOC13 uses sample duration of 20 SYSCLK cycles

AdcaRegs.ADCSOC14CTL.bit.CHSEL = 2; //SOC14 converts ADCINA2

AdcaRegs.ADCSOC14CTL.bit.ACQPS = 19; //SOC14 uses sample duration of 20 SYSCLK cycles

AdcaRegs.ADCSOC15CTL.bit.CHSEL = 3; //SOC15 converts ADCINA3

AdcaRegs.ADCSOC15CTL.bit.ACQPS = 19; //SOC15 uses sample duration of 20 SYSCLK cycles

当接收到来自 CPU1 定时器 2 的第一次触发时，如果 ADC 处于空闲状态，将立即执行 SOC12 和 SOC13 的转换。如果 ADC 正忙，则在 SOC12、SOC13 获得优先级后再进行转换。SOC12 和 SOC13 的转换结果分别保存在 ADCRESULT12 和 ADCRESULT13 中。

SOC13 转换完成后，轮询指针会将最高优先级赋予 SOC14。因此，当下一次 CPU1 定时器 2 的触发到来时，SOC14 和 SOC15 会被置为待转换状态并最终完成转换。SOC14 和 SOC15 的结果分别保存在 ADCRESULT14 和 ADCRESULT15 中。

后续的触发将在转换 SOC12、SOC13 与转换 SOC14、SOC15 之间交替切换。

上述示例在两组转换之间交替切换，使用类似的方法也可以实现三组或更多组不同的转换。

## 13.3.9.2 突发模式优先级示例

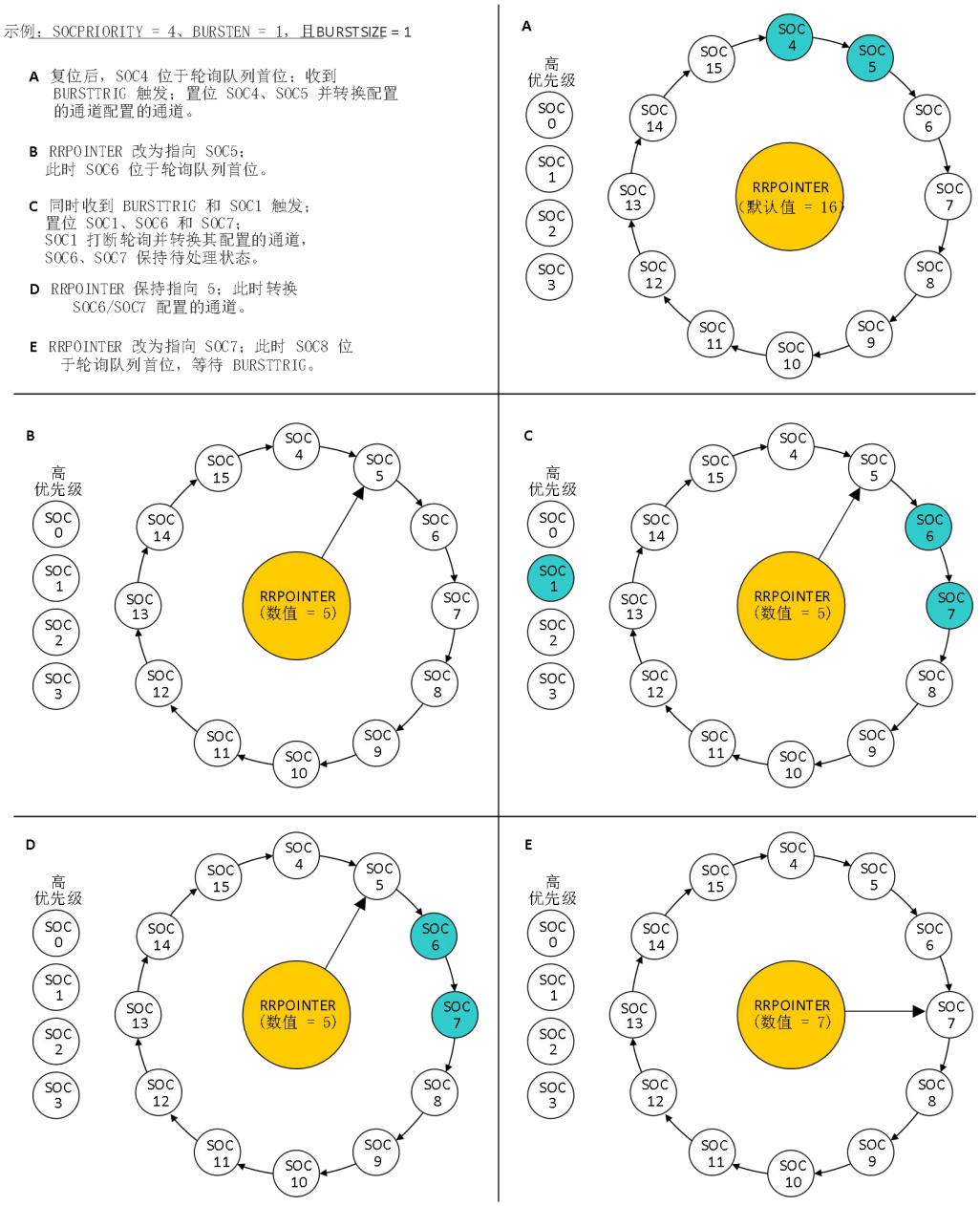



图 13-6 高优先级与突发模式示例

## 13.3.10 EOC 和中断操作

每个 SOC 都有对应的转换结束（EOC）信号。该 EOC 信号可用于触发 ADC 中断。

ADC 可配置为在采集窗口结束时或电压转换完成时产生 EOC 脉冲。这一功能通过 ADCCTL1 寄存器中的 INTPULSEPOS 位进行配置。

每个 ADC 模块具有 4 个可配置的 ADC 中断。这些中断可由 16 路 EOC 信号中的任意一路触发。可直接读取每个 ADCINT 对应的标志位，以判断相关 SOC 是否转换完成，或将该中断传递至 CLIC 模块。

ADCCTL1 寄存器中的 ADCBSY 位被清零，并不表示一组 SOC 中的所有转换都已完成，仅表示 ADC 已准备好处理下一次转换。若要判断一个 SOC 序列是否全部转换完成，可将 ADCINT 标志与序列中最后一个 SOC 关联，并监测该 ADCINT 标志。

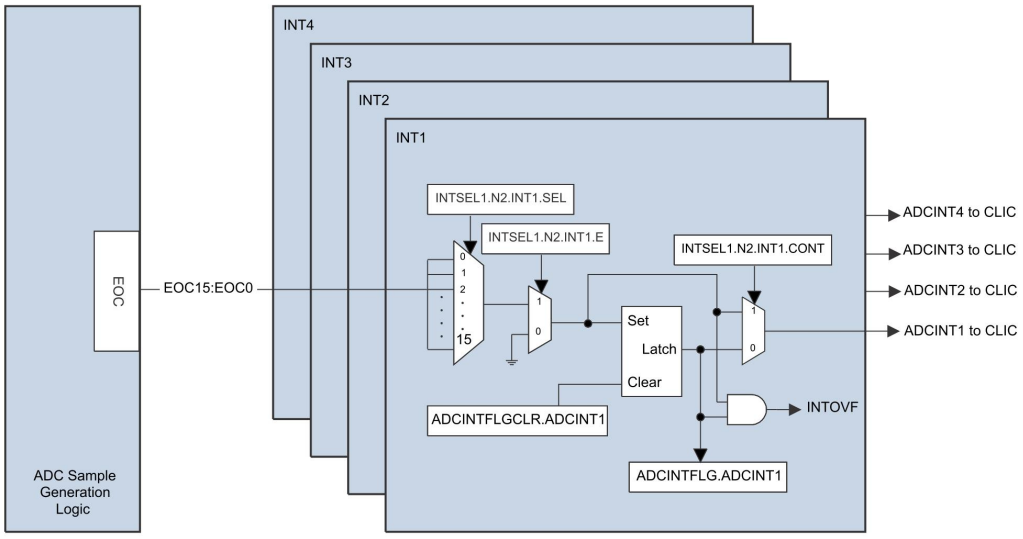



图 13-7 ADC EOC 中断

## 13.3.11 中断溢出操作

若 EOC 信号试图置位 ADCINTFLG 寄存器中的某一标志位，但该标志位已处于置位状态，则会发生中断溢出。

默认情况下，溢出中断不会传递至 CLIC 模块。当 ADCINTFLG 寄存器中某个指定标志位发生溢出时，ADCINOVF 寄存器中对应的标志位会被置位。该溢出标志位仅用于检测溢出事件的发生，并不会阻止后续中断向 CLIC 模块传递。

当 ADC 中断溢出发生时，应用程序必须在中断服务函数内或后台循环中检查对应的 ADCINTOVF 标志位，并在检测到溢出时执行相应的处理操作。以下代码片段展示了在尝试清除 ADCINT 标志位后，如何在中断服务程序中检查 ADCINTOVF 标志位：

```
// Clear the interrupt flag
AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; //clear INT1 flag for ADC-A
// Check if an overflow has occurred
if(1 == AdcaRegs.ADCINTOVF.bit.ADCINT1) //ADCINT overflow occurred
{
AdcaRegs.ADCINTOVFCLR.bit.ADCINT1 = 1 //Clear overflow flag
AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1 //Re-clear ADCINT flag
}
//
// Clear the interrupt flag
//
ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);
//
// Check if an overflow has occurred
//
if(true == ADC_getInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1))
{
ADC_clearInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1);
ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);
}
```

### 13.3.12 连续中断操作

ADCINTSEL1N2 和 ADCINTSEL3N4 寄存器中的 INTxCONT 位，用于配置当某一 ADCINTFLG 标志位因上一次中断尚未清除时，中断该如何处理。

该模式默认禁用，此时额外的重叠中断不会发送到 CLIC 模块。使能该模式后，ADC 中断将始终发送到 CLIC 模块。

如果在 ADCINTFLG 置位期间又产生中断，无论 INTxCONT 位如何配置，ADCINTOVF 寄存器都会保持置位。

### 13.3.13 预触发中断模式

使能预触发中断模式可以让应用程序在 ADC 转换结果尚未就绪时就进入 ADC 中断服务程序。这样应用程序可以提前完成必要的预处理工作，从而在 ADC 结果一旦就绪时立即对其进行处理。

如果提前中断的触发时间过早，应用程序就需要空等待，直到更新后的 ADC 结果就绪。为避免这种情况，提前中断模式下 ADC 中断的进入时间可通过 ADCINTCYCLE 寄存器中的 DELAY 字段进行配置。

要使用可配置的中断时间，ADC 必须工作在提前中断模式下。为此，需将 ADCCTL1 寄存器中的 INTPULSEPOS 位清零（置 0）。

ADCINTCYCLE 寄存器中的 DELAY 值，用于设定在 SOC 脉冲下降沿之后、ADCINT 标志位置位之前额外插入的 SYSCLK 周期数。

如果 DELAY 的值超过转换结束（EOC）时刻，则 ADC 中断将与 EOC 信号同时产生。

当 INTPULSEPOS 被置为 1 时，向 DELAY 写入任何值都不会对中断产生造成影响

### 13.3.14 后处理单元

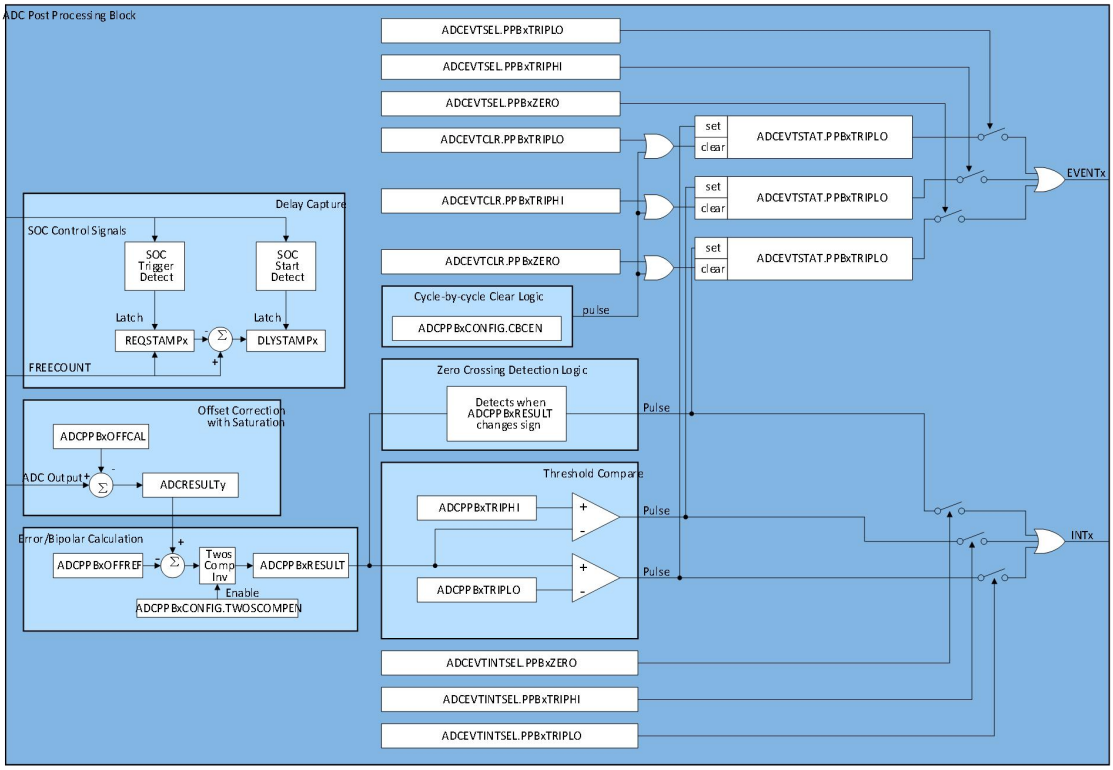



图 13-8 ADC ppb 框图

每个 ADC 模块包含 4 个后处理模块（PPB）。通过 ADCPPBxCONFIG.CONFIG 位字段，可将这些后处理模块与 16 个结果寄存器中的任意一个相关联。

后处理模块可实现以下功能：

- 消除与 ADCIN 通道相关的偏移误差
- 减去一个参考值
- 标记过零点，并可选择触发 PWM 及产生中断
- 标记上限或下限比较阈值，并可选择触发 PWM 及产生中断
- 记录关联的 SOC 触发信号到实际开始采样之间的延迟时间

### 13.3.14.1 ppb 偏移校准

在许多应用中，外部传感器和信号源会产生偏移量。对 ADC 偏移进行全局校准不足以补偿这些随通道不同而变化的偏移。后处理模块（PPB）可以零开销消除这些偏移，在严苛的控制环路中节省大量时钟周期。

偏移校正的实现方式如下：首先将 ADCPPBxCONFIG.CONFIG 指向目标 SOC，然后向 ADCPPBxOFFCAL.OFFCAL 寄存器写入偏移校正值。后处理模块会自动将原始转换结果与 OFFCAL 寄存器中的数值进行加 / 减运算，并将结果存入 ADCRESULT 寄存器。该加法 / 减法操作会进行饱和限制：减法饱和值为 0，加法饱和值为 4095。

向 OFFCAL 寄存器写入 0 可有效禁用偏移校正功能，原始结果将不经修改直接送入 ADCRESULT 寄存器。

可以将多个 PPB 指向同一个 SOC。这种情况下，实际生效的 OFFCAL 值来自编号最大的那个 PPB。

在 SOC0 上使用 PPB 时需要格外小心，因为所有 PPB 默认都指向该 SOC。这可能导致编号较大的 PPB 意外覆盖编号较小的 PPB 的偏移校正值。

### 13.3.14.2 ppb 误差校准

首先将 ADCPPBxCONFIG.CONFIG 指向目标 SOC，然后向 ADCPPBxOFFCAL.OFFREF 寄存器写入数值。后处理模块会自动用 ADCRESULT 中的值减去 OFFREF 寄存器中的值，并将结果存入 ADCPPBxRESULT 寄存器。该减法运算会生成一个符号扩展的 32 位结果。

通过将 ADCPPBxCONFIG 寄存器中的 TWOSCOMPEN 位置 1，还可以在将计算结果存入 ADCPPBxRESULT 寄存器之前，对其进行选择性取反。

注：

(1)不能向 ADCPPBxOFFREF 寄存器写入大于 12 位的数值。

(2)由于 ADCPPBxRESULT 寄存器对每个 PPB 都是独立的，因此可以将多个 PPB 指向同一个 SOC，并使每个 PPB 得到不同的结果。

(3)向 ADCPPBxOFFREF 寄存器写入 0 可有效禁用误差计算功能，此时 ADCRESULT 的值将原样送入 ADCPPBxRESULT 寄存器。

(4)向 ADCPPBxOFFREF 写入新值会立即更新 ADCPPBxRESULT 寄存器。但是，来自 PPB 的标志位要等到下一次转换结束（EOC）才会更新。例如：如果修改 ADCPPBxOFFREF 导致 ADCPPBxRESULT 改变符号，但下一次转换又将结果变回修改前的符号，则不会置位 ADCPPBxZERO 标志。

### 13.3.14.3 ppb 限值检测和过零检测

许多应用会对 ADC 转换结果进行阈值检测。PPB 可以自动对上限、下限进行检测，或在 ADCPPBxRESULT 发生符号变化时进行检测。基于这些比较结果，PPB 可以自动产生 PWM 触发信号和中断，降低从采样到 ePWM 的延迟，并减少软件开销。该功能也使得注重安全的应用能够在无需 CPU 干预的情况下，根据 ADC 转换结果超量程直接触发 ePWM。

要使能该功能，首先将 ADCPPBxCONFIG.CONFIG 指向目标 SOC，然后向 ADCPPBxTRIPHI.LIMITHI 和 ADCPPBxTRIPLO.LIMITLO 寄存器中的一个或两个写入阈值（过零点检测无需额外配置）。当超过这些阈值时，ADCEVTSTAT 寄存器中的 PPBxTRIPHI 位或 PPBxTRIPLO 位将被置位。

ADCEVTSTAT 中的 PPBxZERO 位由 EOC 信号控制，而不是由 ADCPPBxRESULT 的符号变化直接控制。

ADCEVTCLR 寄存器具有对应的位用于清除这些事件标志。ADCEVTSEL 寄存器具有对应的位，允许事件传递到 PWM。ADCEVTINTSEL 寄存器具有对应的位，允许事件传递到 CLIC。

如果同一 ADC 模块中不同的 PPB 事件需要执行不同的操作，则 ADCEVTINT 中断服务函数（ISR） 必须读取 ADCEVTSTAT 寄存器中的 PPB 事件标志，以判断是哪个事件触发了中断。

如果需要分别为上限比较、下限比较和过零点检测生成不同的 ePWM 保护触发信号，可以通过将多个 PPB 指向同一个 SOC 来实现。

过零点检测电路会将结果为 0 的值视为正数。

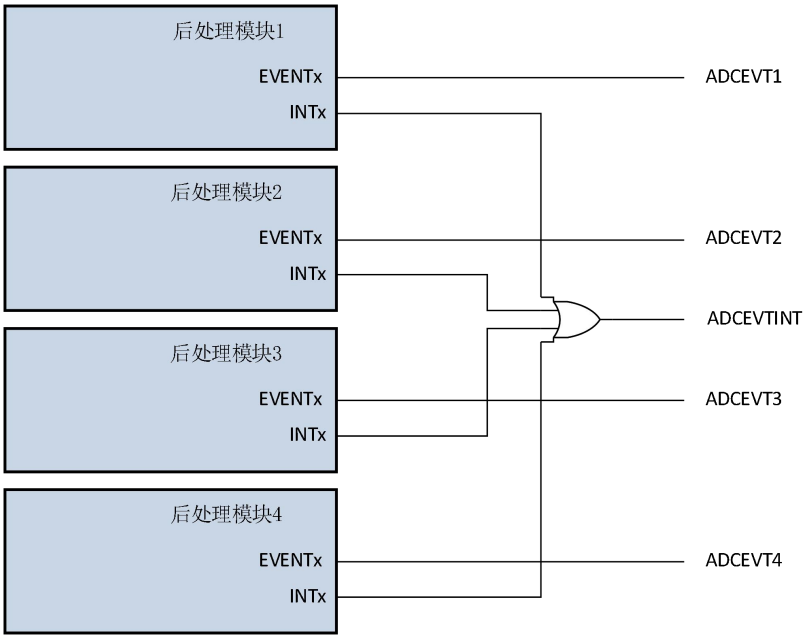



图 13-9 ADC ppb 中断事件

### 13.3.14.4 ppb 采样延迟捕获

当多个控制环路在同一个 ADC 上异步运行时，有可能出现来自两个或多个环路的 ADC 请求发生冲突，导致其中一次采样被延迟。这在系统中会表现为测量误差。通过知晓延迟何时发生以及延迟时长，软件可以使用外推算法来减小误差。

为此，每个 PPB 在 ADCPPBxSTAMP 寄存器中都有 DLYSTAMP 字段。该字段记录了对应 SOC 被触发到 SOC 开始转换之间的 SYSCLK 周期数。

系统提供一个基于 SYSCLK 的全局 12 位自由运行计数器，存放在 ADCCOUNTER 寄存器的 FREECOUNT 字段中。当对应 SOC 的触发信号到达时，

计数器的值会被加载到 ADCPPBxTRIPLO.REQSTAMP 位域。当该 SOC 的实际采样窗口开始时，用当前 FREECOUNT 值减去 REQSTAMP 中的值，并将结果存入 DLYSTAMP。

注：如果在 SOC 触发与 SOC 采集实际开始之间经过超过 4096 SYSCLK 个周期，自由计数寄存器可能会多次溢出，从而导致 dlystamp 值不正确。在使用极慢的转换时，请特别注意避免这种情况发生。如果关联的 SOC 是通过软件触发的，则样本延迟捕获功能将无法正常工作。然而，如果通过软件触发另一个 SOC，导致与 PPB 关联的 SOC 出现延迟，则样本延迟捕获仍能正确记录延迟时间。

### 13.3.15 开路/短路检测电路

开路/短路检测电路（osdetect）可用于检测系统中的引脚故障。

- 分压电阻的容差可能存在较大差异，因此不可使用该功能来检测转换精度。
- 有关模拟输入通道的具体实现与可用性，请参考数据手册。
- 由于驱动阻抗较高，采样保持（S+H）时间需要远大于 ADC 的最小要求值

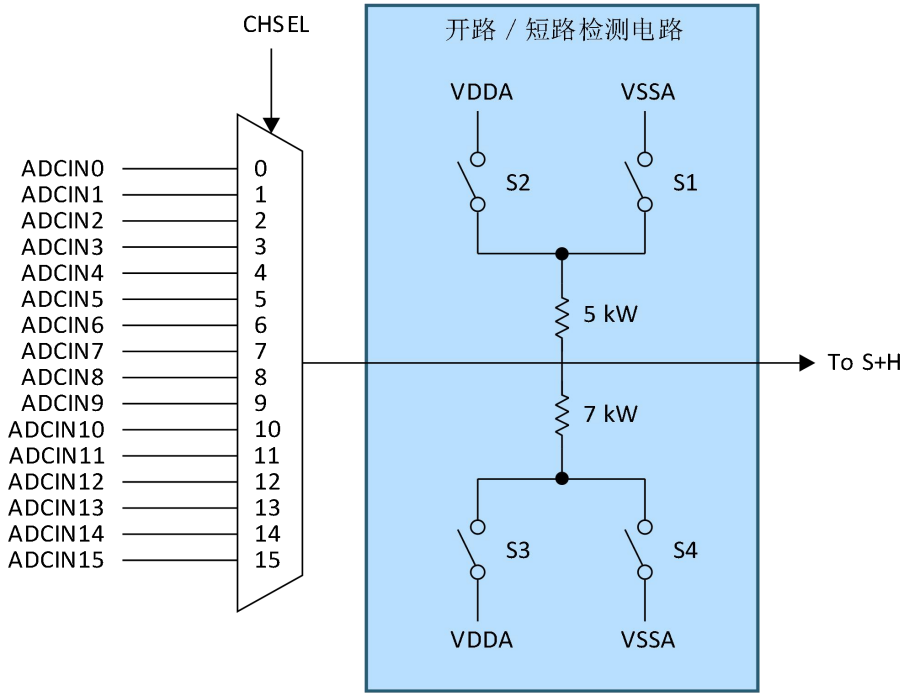



图 13-10 ADC 开路/短路检测电路

表 13-8 DETECTCFG 设置

| ADCOSDETECT.DETECTCFG | 源电压 | S4 | S3 | S2 | S1 | 驱动阻抗 |
|---|---|---|---|---|---|---|
| 0 | Off | Open | Open | Open | Open | Open |
| 1 | Zero Scale | Closed | Open | Open | Closed | 5K ‖ 7K |
| 2 | Full Scale | Open | Closed | Closed | Open | 5K ‖ 7K |
| 3 | 5/12 VDDA | Open | Closed | Open | Closed | 5K ‖ 7K |
| 4 | 7/12 VDDA | Closed | Open | Closed | Open | 5K ‖ 7K |
| 5 | Zero Scale | Open | Open | Open | Closed | 5K |
| 6 | Full Scale | Open | Open | Closed | Open | 5K |
| 7 | Zero Scale | Closed | Open | Open | Open | 7K |

### 13.3.15.1 开路/短路检测操作

图中给出了带有 OSDETECT 功能的典型电路，包括带串联电阻 RS 的信号源、并联电容 CP、等效 OSDETECT 电阻 ROSDETECT 以及电压 VOSDETECT。该电路可作为计算进入采样保持电容信号电平的基础。ROSDETECT 和 VOSDETECT 分别是由 OSDETECT 电路提供的等效输入电阻和等效电压源，不同配置下的取值如表所示。当使能 OSDETECT 功能且由信号源 VS 驱动时，推导采样保持（S/H）输入信号请参考图。

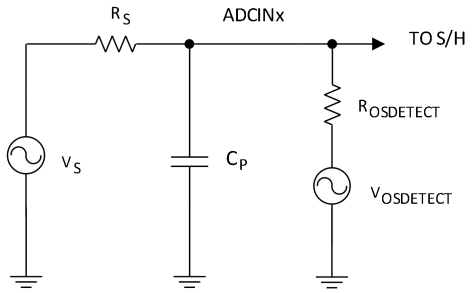



图 13-11 启用 osdetect 时的输入电路等效电路

输入阻抗 RS 和 CP 是信号源的组成部分，也可在设计中用于对信号进行预处理，或控制信号建立时间以满足采样保持（S/H）的要求。

使用 OSDETECT 功能时必须考虑输入路径，因为它会影响转换结果。例如，在使能该功能时驱动输入信号，会使信号 VS 通过 RS 连接到 OSDETECT 电路，从而影响 ADC 结果。较大的 CP 值（数百 pF 以上级别）需要使用更长的 ACQPS（采样窗口），以确保转换前输入信号已完成建立。

使能该电路的步骤：

对 ADC 进行转换配置（例如：通道、SOC、ACQPS、预分频器、触发源等）。

启动一次转换并查看转换结果。

根据输入端的驱动源以及 RS 和 CP 的数值对结果进行解析。如果 VS 信号可以与输入引脚断开，则可使用该电路检测输入引脚开路与短路。

### 13.3.15.2 检测输入引脚的开路

通过循环切换不同的 OSDETECT 配置，输入信号会被拉向对应的内部参考电压。驱动能力良好的输入（引脚未开路） 受到的影响极小。但如果引脚处于开路状态，采样到的电压会接近表中给出的参考电压。

### 13.3.15.3 检测输入引脚的短路

通过循环切换不同的 OSDETECT 配置，输入信号会被拉向对应的内部输出电压。具有一定驱动能力的输入（引脚未短路） 会被拉向各个对应的输出电压。但如果引脚发生短路，信号则会保持在同一电压不变。

### 13.3.15.4 ADC 上电顺序

系统上电完成后，ADC 处于掉电禁用状态。上电启用 ADC 需按照以下步骤执行：

1.使能所需的 ADC 时钟。

2.在 ADCCTL2 的 PRESCALE 字段中配置所需的 ADC 时钟分频。

3.通过置位 ADCCTL1 中的 ADCPWDNZ 位，将 ADC 上电。

4.采样前需要插入一段延时，具体所需时间请参考数据手册。

如果同时给多个 ADC 上电，步骤 1 和步骤 3 可通过一条写指令完成对所有 ADC 的配置。并且，只要延时发生在所有 ADC 都已开始上电之后，只需要执行一次延时。

### 13.3.16 ADC 校准

在生产制造与测试过程中，会对 ADC 的增益、偏移和线性度进行校准。这些校准参数存储在 OTP 存储器中，并可通过 C 调用函数加载。

Device_cal () 函数将 ADC 的校准值从 OTP 存储器复制到对应的校准寄存器。

在加载正确的出厂校准参数之前，ADC 及其他模拟模块无法保证按照数据手册指标正常工作。同样，如果将非出厂校准值写入校准寄存器，ADC（及其他模块）也无法保证符合数据手册的性能规范。

Boot ROM 会自动调用校准函数，因此校准参数初始加载无需用户干预。

### 13.3.17 ADC 0 位校准

ADC 偏移误差会在出厂测试期间进行测定与校准。但是，如果最终应用有特殊需求，用户仍可自行执行偏移校准。本节介绍在单端模式下，通过内部连接 VREFLO 执行偏移校准的方法。

零偏移误差定义为：当输入电压为 VREFLO 时，转换结果与 0 的差值。零偏移误差可正可负。为校正该误差，需向 ADCOFFTRIM 寄存器写入大小相等、极性相反的校正值。该寄存器中的数值会在结果送入 ADC 结果寄存器之前生效。整个操作完全在 ADC 内核内部完成，因此不会影响结果时序，并且无论校准值如何，ADC 的全动态范围均保持不变。

### 13.3.18 ADC 时序

将模拟电压转换为数字量的过程可分为采样保持阶段（S+H 阶段）和转换阶段。

ADC 采样保持电路（S+H）由 SYSCLK 提供时钟，而 ADC 转换过程由 ADCCLK 提供时钟。ADCCLK 是根据 ADCCTL2 寄存器中的 PRESCALE 位域对 SYSCLK 进行分频后产生的。

采样保持时间计算公式为：当前转换 SOC 的 ACQPS 位域的值加 1，再乘以 SYSCLK 周期。

用户必须确保该时长同时超过 1 个 ADCCLK 周期以及数据手册中规定的最小 S+H 持续时间。

转换时间约为 10.5 个 ADCCLK 周期，实际精确的转换时间一定是整数个 SYSCLK 周期。

### 13.3.18.1 adc 时序图表

下图展示了在以下假设条件下，两个 SOC 的 ADC 转换时序：

1.SOC0 和 SOC1 配置为使用同一触发源。

2.触发发生时，无其他 SOC 正在转换或等待转换。

3.轮询指针当前状态使 SOC0 优先转换。

4.ADCINTSEL 配置为在 SOC0 转换结束时置位 ADCINT 标志（该标志是否传递到 CPU 并产生中断，由 CLIC 模块配置决定）。

表中描述了下图中的时序参数。表中列出了 ADC 时序。

表 13-9 ADC 时序参数说明

| 参数 | 描述 |
| --- | --- |
| tSH | 采样保持窗口（S+H 窗口）的时长在该窗口结束时，采样保持电容上的电压将被转换为数字量。时长由 (ACQPS + 1) 个 SYSCLK 周期 决定。每个 SOC 均可独立配置 ACQPS，因此不同 SOC 的 tSH 不一定相同。<br>注意：无论器件时钟如何配置，采样保持电容上的电压都会在 S+H 窗口结束前约 5ns 被锁存。 |
| tLAT | 从采样保持窗口结束，到转换结果锁存到 ADCRESULTx 寄存器的时间。如果在此时间之前读取 ADCRESULTx 寄存器，将返回上一次转换的结果。 |
| tEOC | 从当前采样保持（S+H）窗口结束，到下一次 ADC 转换的采样保持窗口可以开始的时间。下一次采样可以在转换结果被锁存之前就启动。 |
| tINT | 从采样保持窗口结束到 ADCINT 标志被置位的时间（若已配置）。<br>若 ADCCTL1 寄存器中的 INTPULSEPOS 位置 1，则 tINT 与转换结束信号（EOC）同步。<br>若 INTPULSEPOS 位为 0，且 ADCINTCYCLE 寄存器中的 OFFSET 字段不为 0，则在 ADCINT 标志置位前会有 OFFSET 个 SYSCLK 周期的延迟。该延迟可用于在采样数据刚好就绪时进入中断服务函数。<br>若 INTPULSEPOS 位为 0，则 tINT 与采样保持窗口结束时刻同步。如果 tINT 触发读取 ADC 结果寄存器（通过触发读取结果的中断服务函数），必须确保读取操作发生在结果锁存之后，否则会读到上一次的结果。 |

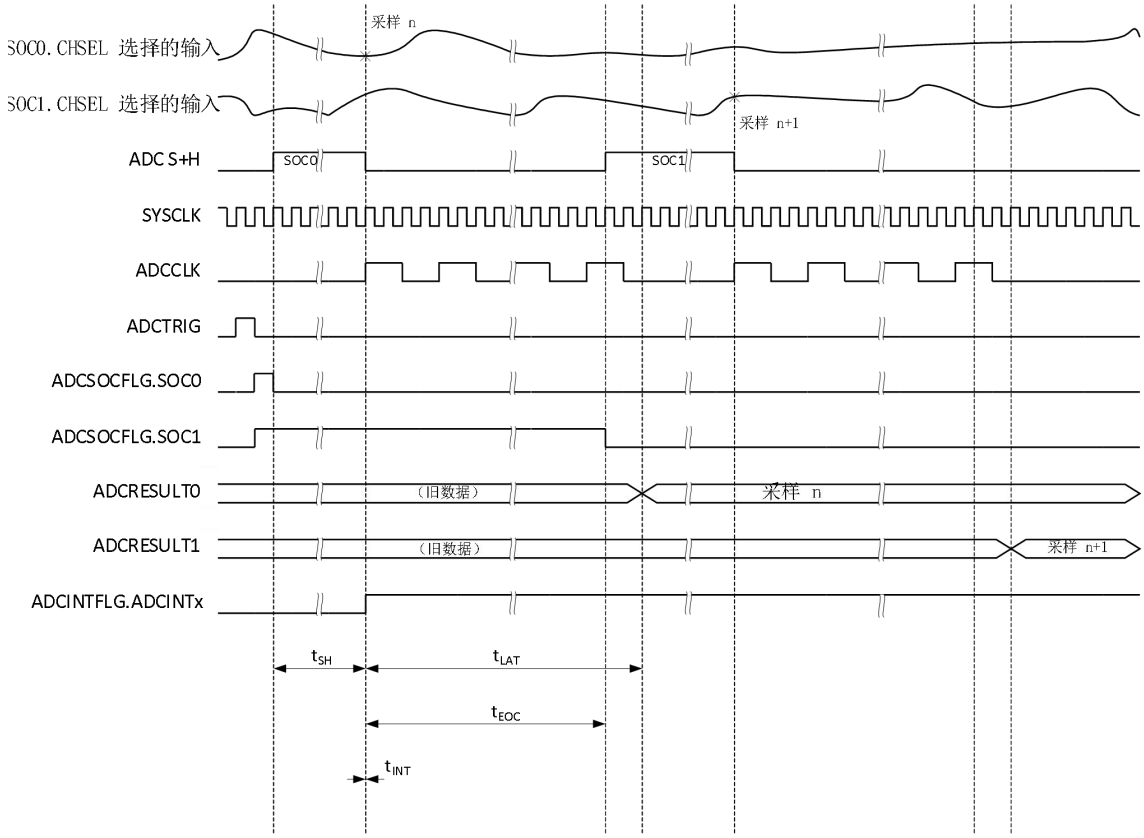



图 13-12 预触发中断模式下 12 位 adc 的时序

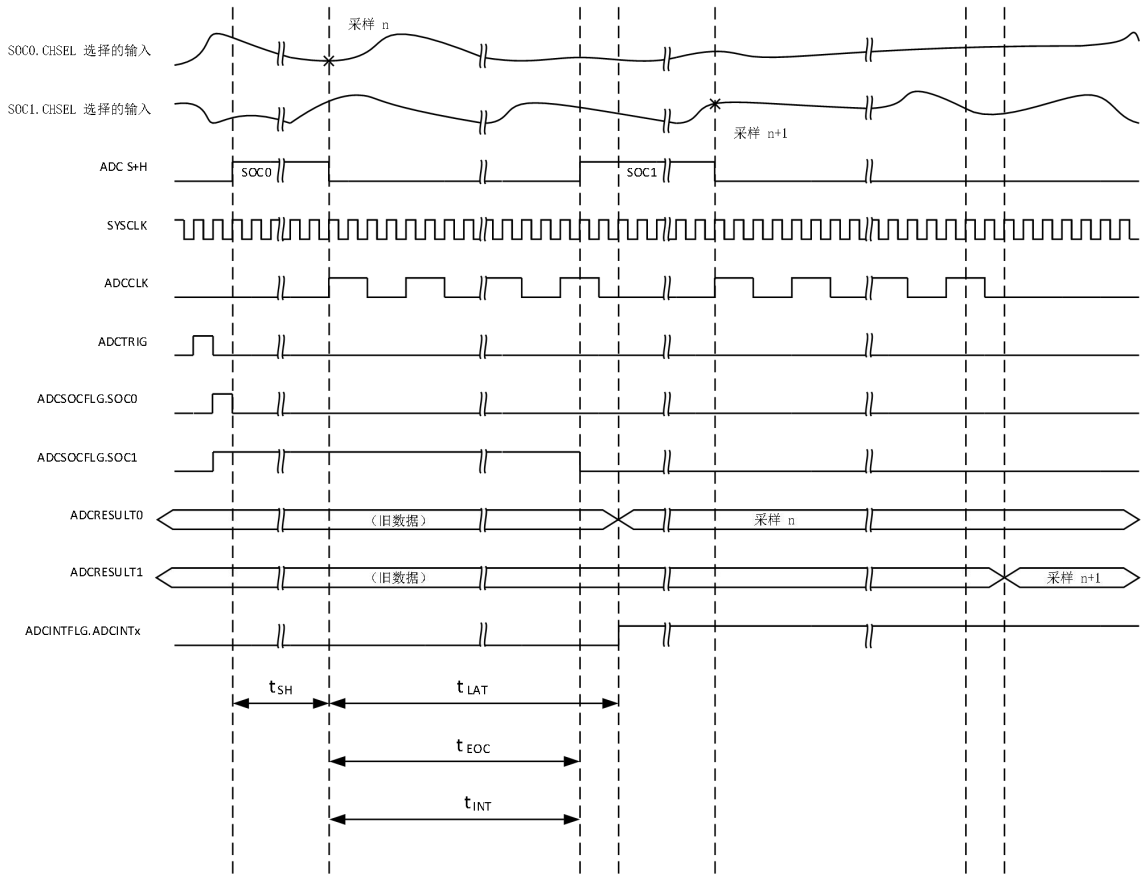



图 13-13 延迟中断模式下 12 位 adc 的时序

表 13-10 12 位 adc 的时序

| ADCCLK 预分频 | | SYSCLK 周期数 | | | |
|---|---|---|---|---|---|
| ADCCTL2. PRESCALE | 分频比例 | tEOC | tLAT | tINT（提前） | tINT（延迟） |
| 0 | 1 | 11 | 13 | 0 | 11 |
| 2 | 2 | 21 | 23 | 0 | 21 |
| 4 | 3 | 31 | 34 | 0 | 31 |
| 6 | 4 | 41 | 44 | 0 | 41 |
| 8 | 5 | 51 | 55 | 0 | 51 |
| 10 | 6 | 61 | 65 | 0 | 61 |
| 12 | 7 | 71 | 76 | 0 | 71 |
| 14 | 8 | 81 | 86 | 0 | 81 |

## 13.3.19 附加信息

以下章节包含更多实用信息

### 13.3.19.1 确保同步进行

为了获得最佳性能，设备上的所有 ADC 必须同步运行。设备的数据手册中列出了在同步和异步模式下，对于那些在两种工作模式下不同的参数的性能指标。为确保同步运行，设备上的所有 ADC 必须以锁步方式运行。这可以通过向所有 ADC 写入配置值来实现，使所有 ADC 的采样和转换阶段完全对齐。实现这一目标最简便的方法是，为每个 ADC 的 SOC 配置寄存器写入相同的值，包括触发选择和 ACQPS（S+H 持续时间）。此外，同步 ADC 还必须为 SOC 优先级控制、突发模式、突发触发和突发大小设置相同的值。

#### 13.3.19.1.1 基本同步操作

以下示例配置了两个 SOC，分别位于 ADCA 和 ADCC 上，其触发选择器和 ACQPS 值均相同。这使得 ADCA 与 ADCC 之间能够实现同步操作。对于拥有超过两个 ADC 的设备，可采用相同的原理来同步所有 ADC。

**Example: Basic Synchronous Operation**

AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4; //SOC0 converts ADCINA4

AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 uses sample duration of 20 SYSCLK cycles

AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 begins conversion on ePWM3 SOCB

AdccRegs.ADCSOC0CTL.bit.CHSEL = 0; //SOC0 converts ADCINC0

AdccRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 uses sample duration of 20 SYSCLK cycles

AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC1CTL.bit.CHSEL = 4; //SOC1 converts ADCINA4

AdcaRegs.ADCSOC1CTL.bit.ACQPS = 30; //SOC1 uses sample duration of 31 SYSCLK cycles

AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 10; //SOC1 begins conversion on ePWM3 SOCB

AdccRegs.ADCSOC1CTL.bit.CHSEL = 1; //SOC1 converts ADCINC1

AdccRegs.ADCSOC1CTL.bit.ACQPS = 30; //SOC1 uses sample duration of 31 SYSCLK cycles

AdccRegs.ADCSOC1CTL.bit.TRIGSEL = 10; //SOC1 begins conversion on ePWM3 SOCB

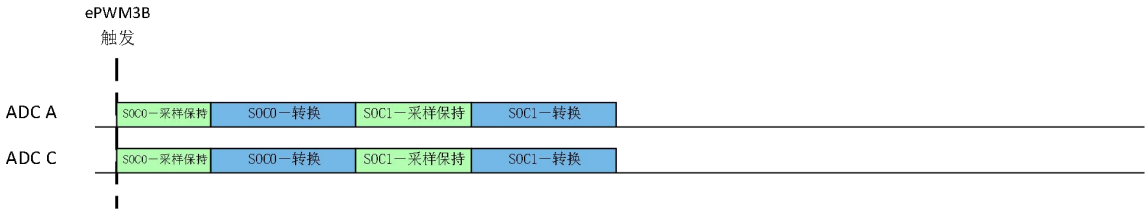



图 13-14 示例：基本同步操作

从图中可以注意到几点：首先，对于相同数量的 SOC，其 ACQPS 值必须相同；而对于不同数量的 SOC，可以使用不同的 ACQPS 值。因此，同步操作并不需要一个全局的 S+H 时间，而只需确保同时采样的通道具有相同的 S+H 时长即可。另一个重要的点是，任何通道选择值均可用于任意 SOC。最后，本示例假设采用轮询方式运行。若需使用高优先级 SOC，则必须在所有 ADC 上配置相同的优先级。

### 13.3.19.1.2 多触发源同步操作

只要每组 SOC 的触发选择和 ACQPS 设置完全相同，就可以使用多个触发源同时实现同步操作。以下示例展示了在使用三个 SOC 和两个触发源的情况下， ADCA 与 ADCC 之间的同步操作。图中表明，任何相对触发时序组合都能实现同步操作。

**Example: Synchronous Operation With Multiple Trigger Sources**

AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4; //SOC0 converts ADCINA4

AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 uses sample duration of 20 SYSCLK cycles

AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 begins conversion on ePWM3 SOCB

AdccRegs.ADCSOC0CTL.bit.CHSEL = 0; //SOC0 converts ADCINC0

AdccRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 uses sample duration of 20 SYSCLK cycles

AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC1CTL.bit.CHSEL = 4; //SOC1 converts ADCINA4

AdcaRegs.ADCSOC1CTL.bit.ACQPS = 30; //SOC1 uses sample duration of 31 SYSCLK cycles

AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 10; //SOC1 begins conversion on ePWM3 SOCB

AdccRegs.ADCSOC1CTL.bit.CHSEL = 1; //SOC1 converts ADCINC1

AdccRegs.ADCSOC1CTL.bit.ACQPS = 30; //SOC1 uses sample duration of 31 SYSCLK cycles

AdccRegs.ADCSOC1CTL.bit.TRIGSEL = 10; //SOC1 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC2CTL.bit.CHSEL = 0; //SOC2 converts ADCINA0

AdcaRegs.ADCSOC2CTL.bit.ACQPS = 19; //SOC2 uses sample duration of 20 SYSCLK cycles

AdcaRegs.ADCSOC2CTL.bit.TRIGSEL = 2; //SOC2 begins conversion on CPU Timer1

AdccRegs.ADCSOC2CTL.bit.CHSEL = 2; //SOC2 converts ADCINC2

AdccRegs.ADCSOC2CTL.bit.ACQPS = 19; //SOC2 uses sample duration of 20 SYSCLK cycles

AdccRegs.ADCSOC2CTL.bit.TRIGSEL = 2; //SOC2 begins conversion on CPU Timer1

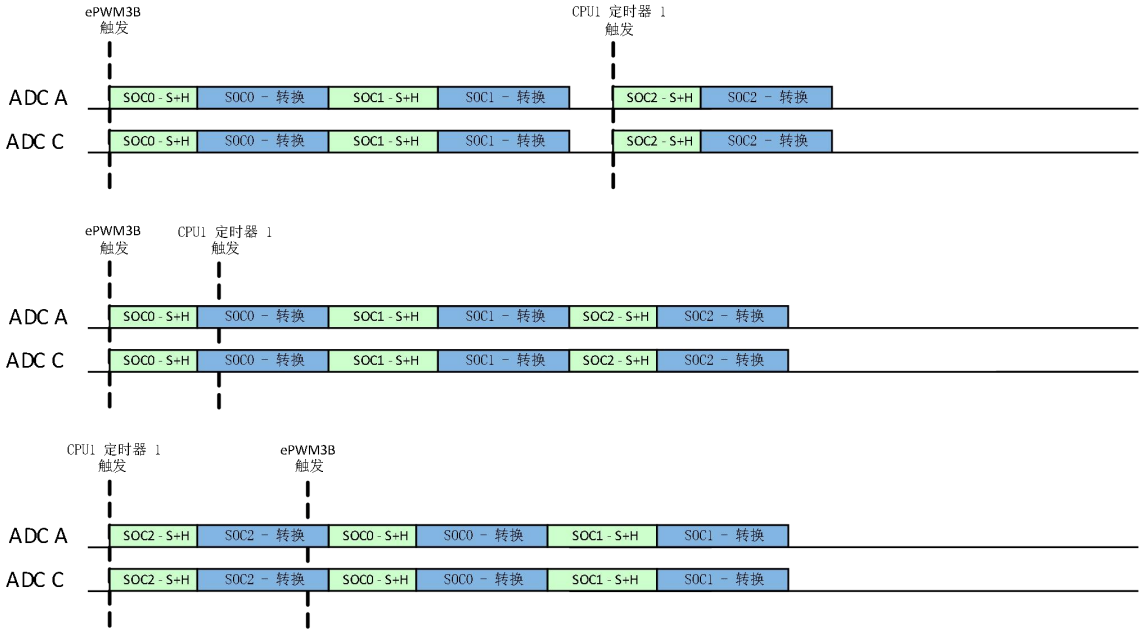



图 13-15 示例：多触发源同步操作

请注意，除软件触发外，trigsel 字段中可选择的任何触发源均可使用。由于无法同时为所有 ADC 发出软件触发信号，因此系统可能会采用异步运行模式。此外，只要 ADCINTSOCSEL1 和 ADCINTSOCSEL2 寄存器对所有 ADC 进行相同的配置，且未使用软件触发来启动转换链，则 ADCINT1 或 ADCINT2 也可用作触发源。

### 13.3.19.1.3 SOC 数值不一致的同步运行

若仅使用单一触发源，单个 ADC 可比其他 ADC 使用更多 SOC，且仍能保持同步运行。

**Example: Synchronous Operation With Uneven SOC Numbers**

AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4; //SOC0 converts ADCINA4

AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 uses sample duration of 20 SYSCLK cycles

AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 begins conversion on ePWM3 SOCB

AdccRegs.ADCSOC0CTL.bit.CHSEL = 0; //SOC0 converts ADCINC0

AdccRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 uses sample duration of 20 SYSCLK cycles

AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC1CTL.bit.CHSEL = 4; //SOC1 converts ADCINA4

AdcaRegs.ADCSOC1CTL.bit.ACQPS = 30; //SOC1 uses sample duration of 31 SYSCLK cycles

AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 10; //SOC1 begins conversion on ePWM3 SOCB

AdccRegs.ADCSOC1CTL.bit.CHSEL = 1; //SOC1 converts ADCINC1

AdccRegs.ADCSOC1CTL.bit.ACQPS = 30; //SOC1 uses sample duration of 31 SYSCLK cycles

AdccRegs.ADCSOC1CTL.bit.TRIGSEL = 10; //SOC1 begins conversion on ePWM3 SOCB

AdcaRegs.ADCSOC2CTL.bit.CHSEL = 0; //SOC2 converts ADCINA0

AdcaRegs.ADCSOC2CTL.bit.ACQPS = 30; //SOC2 uses sample duration of 31 SYSCLK cycles

AdcaRegs.ADCSOC2CTL.bit.TRIGSEL = 10; //SOC2 begins conversion on ePWM3 SOCB

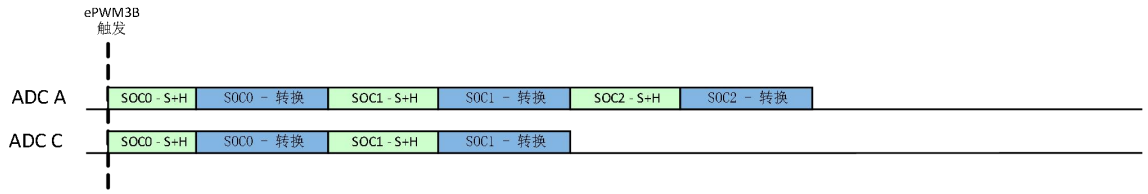



图 13-16 示例：SOC 数值不一致时的同步操作

请注意，如果触发信号在所有 SOC 完成转换之前再次出现， ADCC 会立即开始对 SOC0 进行转换，而 ADCA 则需等到 SOC2 完成后再重新开始转换 SOC0。这会导致异步操作，因此必须注意避免触发信号溢出。

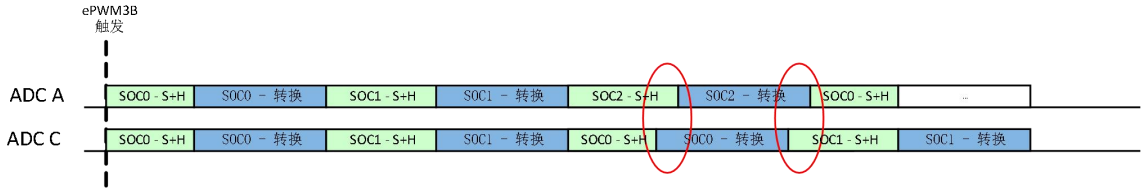



图 13-17 示例：SOC 数值不一致的异步操作——触发溢出

### 13.3.19.1.4 非重叠转换

如果用户能够确保转换时序不重叠，则无需在所有 ADC 上对所有 SOC 进行相同的配置，以实现与同步操作等效的性能。例如，若系统中的两个 ADC 触发信号来自两个始终相差 180 度的 ePWM 源，则 SOC0 可分别用于 ADCA 和 ADCC ，采用不同的触发源及不同的 ACQPS 值。

**Example: Operation with Non-overlapping Conversions**

//ePWM3 SOCA and SOCB are 180 degrees out of phase

AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4; //SOC0 converts ADCINA4

AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 uses sample duration of 20 SYSCLK cycles

AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 begins conversion on ePWM3 SOCB

AdccRegs.ADCSOC0CTL.bit.CHSEL = 0; //SOC0 converts ADCINC0

AdccRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 uses sample duration of 20 SYSCLK cycles

AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 9; //SOC0 begins conversion on ePWM3 SOCA

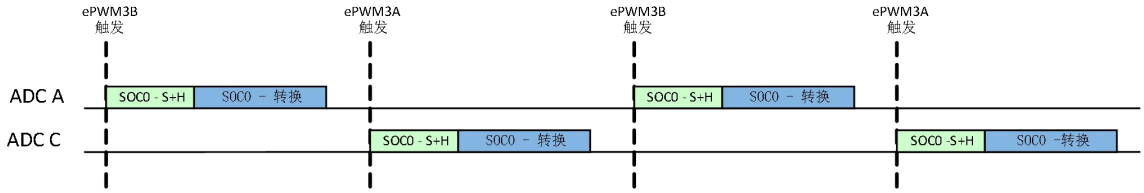



图 13-18 示例：采用非重叠转换的同步等效运行

### 13.3.19.2 选择采样窗口持续时间

为确保 ADC 的正确工作，输入至 ADC 的信号必须有足够的时间对采样与保持电容（Ch）进行充电。通常，S+H 持续时间的选择应使采样电容的充电值达到最终值的½ LSB 或¼ LSB 以内，具体取决于可接受的归一化误差范围。确定所需归一化时间的最佳方法是模拟 ADC 及其驱动电路，以确保其具备足够的归一化性能。此外，也可利用 RC 归一化模型来估算所需的归一化时间，该模型的时间常数由以下公式给出：

$$\tau = RS + Ron \times Ch + RS \times CS + Cp$$

所需时间常数的数量由以下方程给出

$$k = \ln(2^n / settling_error) - \ln((CS + Cp) / CH)$$

因此，总 S+H 时间必须至少设置为

$$t = k \cdot \tau$$

以下参数由器件数据手册中的 ADC 输入模型提供：

- n = ADC 分辨率（单位：位）
- RON = ADC 采样开关电阻（单位：Ω）
- CH = ADC 采样电容（单位：pF）
- Cp = ADC 通道寄生输入电容（单位：pF）；而以下参数则取决于具体应用设计：
- 定点误差 = 允许的定点误差（单位：LSB）
- Rs = ADC 驱动电路源阻抗（通常为Ω或 kΩ）
- CS = ADC 输入引脚上的电容（通常为 pF 或 nF）。

例如，假设以下参数：

- n = 12 位
- RON = 500Ω
- CH = 12.5pF
- Cp = 12.7pF
- 定点误差 = ¼ LSB
- Rs = 180Ω
- Cs = 150pF。

时间常数的计算公式如下：

τ = (180Ω + 500Ω) × 12.5pF + 180Ω × (150pF + 12.7pF) = 37.8ns

所需的时间常数个数为：

k = ln(2^12 / 0.25) - ln((150pF + 12.7pF) / 12.5pF) = 9.70 - 2.57 = 7.13

因此，S+H 时间必须至少设置为：37.8ns × 7.13 = 270ns。若 SYSCLK = 120MHz，则每个 SYSCLK 周期为 8.33ns。S+H 持续时间为 270ns / 8.33ns = 32.4 SYSCLK 周期，因此该输入的 ACQPS 值至少设为上取整(32.4) – 1 = 31。虽然这可以粗略估算所需的采样窗口，但更好的方法是构建一个电路模型，其中包含 ADC 输入模型、源阻抗/电容模型以及任何板级寄生效应，并使用 SPICE（或类似软件）进行仿真，以验证采样电容器能否稳定达到预期精度。

### 13.3.19.3 实现同步采样

虽然每个 ADC 均未配备双 S+H 电路，但实现同步采样可通过设置两个或更多 ADC 模块的 SOC 触发器，使其使用相同的触发源来完成。以下示例展示了基于 ePWM3 事件的两个 ADC 的同步采样过程：ADCINA3 和 ADCINC5 均被采样。采用 20 SYSCLK 周期的采集窗口，但也可采用不同的时长。

AdcaRegs.ADCSOC0CTL.bit.CHSEL = 3; //SOC0 converts ADCINA3

AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 uses sample duration of 20 SYSCLK cycles

AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 begins conversion on ePWM3 SOCB

AdccRegs.ADCSOC0CTL.bit.CHSEL = 5; //SOC0 converts ADCINC5

AdccRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 uses sample duration of 20 SYSCLK cycles

AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 begins conversion on ePWM3 SOCB

当接收到 ePWM3 触发信号时，所有 ADC 将立即并行开始转换。所有转换结果均存储在各 ADC 对应的 ADCRESULT0 寄存器中。请注意，此机制的前提是：在触发信号接收时，所有 ADC 均处于空闲状态；若有一个或多个 ADC 正处于忙状态，则其采样时间将不会完全同步。

### 13.3.19.4 结果寄存器映射

系统中每个内存总线控制器均配备 ADC 结果寄存器与 ADC PPB 结果寄存器。总线控制器涵盖特定产品系列及型号下的所有 CPU。对于每个总线控制器，无需配置访问参数即可实现对结果寄存器的读取操作；当多个总线控制器同时尝试读取 ADC 结果时，亦不会发生访问冲突。

### 13.3.19.5 内部温度传感器

内部温度传感器可测量器件的结温。传感器的输出信号可通过内部连接经由 ADC 进行采样；此功能可在 ADCC 上的 ADCIN12 通道及 CMPSS2_HP5 输入端启用，方法是设置 tsnsctl 寄存器中的使能位。若需将温度传感器读数转换为温度值，则需将该读数传递至 ADC 驱动库中的 ADC_getTemperatureC()函数。

### 13.3.19.6 外部参考电路设计

图中展示了外部电压参考生成电路的基本结构。为获得最佳性能，外部生成的参考电压在驱动至 ADC 参考引脚之前，必须经过一个具有良好带宽和低输出阻抗的精

密运算放大器进行缓冲。应在 PCB 上将高参考引脚与低参考引脚之间的电容尽可能靠近引脚布置，以帮助吸收高频电流；有时还需在此电容串联一个电阻（通常<1 Ω），以维持运算放大器的稳定性。

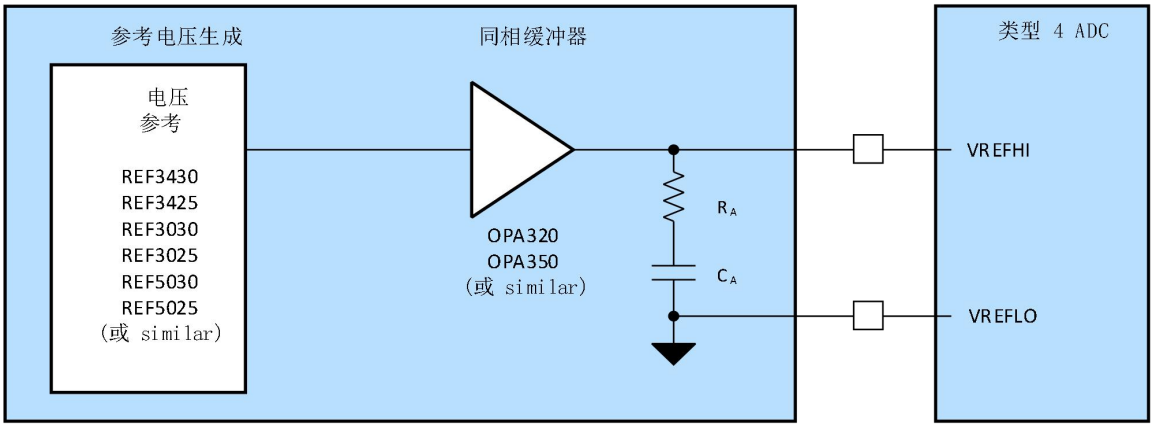



图 13-19 示例：ADC 参考系统

## 13.3.19.7 ADC 回环测试

为了系统诊断或功能安全目的，用户应用程序可以对 ADC 模块进行环回测试，以验证 ADC 是否正确转换。利用第一个 CMPSS 模块中的 DAC 输出，可配置设备向转换器的输入端提供一系列已知电压，并将转换结果与预期结果进行比对。环回测试模式可通过将模拟子系统模块中的 adcloopback 寄存器中对应于待测 ADC 模块的位设置为 1 来启用。图中展示了 CMPSS DAC 输出与 ADC 之间的连接关系。

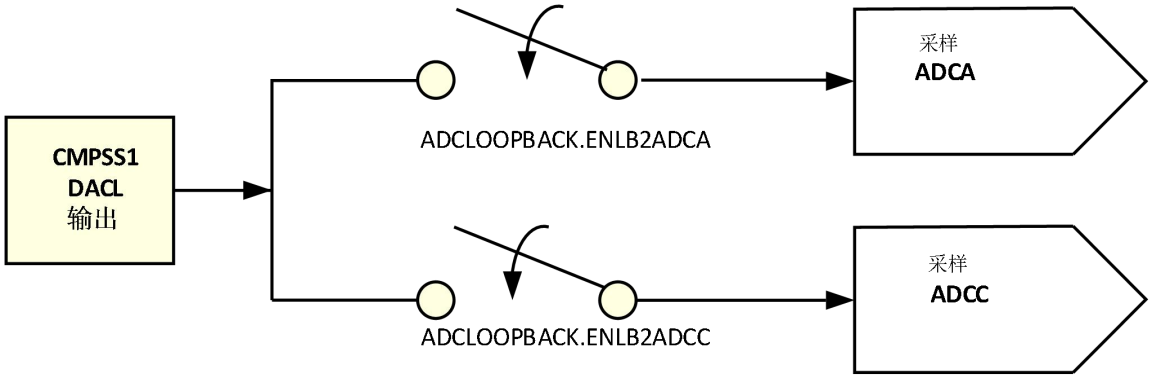



图 13-20 示例：CMPSS 至 ADC 回环连接

在 ADC 环回测试模式下，需注意以下特殊事项：

- ADC 模块始终采样 CMPSS1 DACL 输出，无论 ADCSOCxCTL 的 CHSEL 字段中选择了哪个通道。• 在转换 DAC 输出时，最小采样窗口大小（ACQPS）为 4.27µs（在 120MHz SYSCLK 下相当于 512 SYSCLK 周期）。• CMPSS DAC 的输出分辨率为 6 位，输入 DACVAL 的低 6 位会被舍弃。• ADC 环回测试模式会影响 CMPSS 的触发电压。请避免在常规 CMPSS 运行期间启用 ADC 环回模式。

### 13.3.19.8 内部测试模式

用于诊断目的，ADC 可以使用一个称为“internaltest”的特殊输入选择多路复用器来采样各种内部节点电压。当启用内部测试模式时，“internaltest”多路复用器的选择会覆盖 ADC-A 的输入通道多路复用器：ADC-A 将采样“internaltest”的选择，而非由“ADCSOCxCTL”选定的通道。 CHSEL . 内部测试模式可用于采样 vddcore 电压、VREFLO 、 VDDA 、 VSSA 以及 CMPSS DAC 输出。要启用内部测试模式，需将所需的节点选择写入“internaltestctl”模拟子系统寄存器中的“testsel”字段。出于安全考虑，“internaltestctl”寄存器包含一个写入密钥字段，写入该字段时必须同时写入值 0xA5A5，才能使写入生效；否则，对该寄存器的写入将被忽略。有关“internaltest”多路复用器与各内部器件电压节点连接的详细信息，请参阅“internaltestctl”寄存器描述。要禁用内部测试模式，需将 0 写入“internaltestctl”寄存器中的“testsel”字段。使用内部测试模式时，需注意以下几点：• 当配置非零值时，“internaltestctl”的“testsel”字段会覆盖“ADCA”上的“ADCSOCxCTL. CHSEL ”值。• 在转换“internaltest”时，最小采样窗口大小（ACQPS）为 4.27µs（在 120MHz SYSCLK 下相当于 512 SYSCLK 周期）。• “CMPSS1”DAC 输出至“internaltest”的有效分辨率为 6 位。

### 13.3.19.9 ADC 增益和偏移校准

利用内部测试混频器，可对多个 ADC 之间的增益进行平衡。通过计算各 ADC 之间的相对增益误差，可在软件中对转换结果进行后处理，使每个 ADC 在每个等效输入下均输出相同的信号。要激活增益校准模式，需配置 internaltestctl 模拟子系统寄存

器的 testsel 字段，选择 ENZ_CALIB_GAIN_3P3V。在此模式下，所有 ADC 在触发时采样的电压为( VREFHI * 0.9)，从而覆盖 ADCSOCxCTL 中的通道选择设置。CHSEL．要关闭增益校准模式并恢复至正常工作状态，需将 internaltestctl 的 testsel 字段配置为 0。

## 13.4 寄存器描述

### 13.4.1 ADC 基地址表

表 13-11 ADC 基地址表

| 位字段实例 | 位字段结构 | DriverLib 名称 | 基地址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| AdcaResultRegs | ADC_RESULT_REGS | ADCARESULT_BASE | 0x0063_0B00 | 是 | 是 | - |
| AdccResultRegs | ADC_RESULT_REGS | ADCCRESULT_BASE | 0x0063_0B40 | 是 | 是 | - |
| AdcaRegs | ADC_REGS | ADCA_BASE | 0x0063_7400 | - | - | 是 |
| AdccRegs | ADC_REGS | ADCC_BASE | 0x0063_7500 | - | - | 是 |

### 13.4.2 ADC_RESULT_REGS 寄存器

表中列出了存储器映射寄存器。未列出的偏移地址应视为保留位置，不应修改这些位置的寄存器内容。

表 13-12 ADC_RESULT_REGS 寄存器

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | ADCRESULT0 | ADC 结果 0 寄存器 |
| 2h | ADCRESULT1 | ADC 结果 1 寄存器 |
| 4h | ADCRESULT2 | ADC 结果 2 寄存器 |
| 6h | ADCRESULT3 | ADC 结果 3 寄存器 |
| 8h | ADCRESULT4 | ADC 结果 4 寄存器 |
| Ah | ADCRESULT5 | ADC 结果 5 寄存器 |
| Ch | ADCRESULT6 | ADC 结果 6 寄存器 |
| Eh | ADCRESULT7 | ADC 结果 7 寄存器 |
| 10h | ADCRESULT8 | ADC 结果 8 寄存器 |
| 12h | ADCRESULT9 | ADC 结果 9 寄存器 |
| 14h | ADCRESULT10 | ADC 结果 10 寄存器 |

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 16h | ADCRESULT11 | ADC 结果 11 寄存器 |
| 18h | ADCRESULT12 | ADC 结果 12 寄存器 |
| 1Ah | ADCRESULT13 | ADC 结果 13 寄存器 |
| 1Ch | ADCRESULT14 | ADC 结果 14 寄存器 |
| 1Eh | ADCRESULT15 | ADC 结果 15 寄存器 |
| 20h | ADCPPB1RESULT | ADC 后处理模块 1 结果寄存器 |
| 24h | ADCPPB2RESULT | ADC 后处理模块 2 结果寄存器 |
| 28h | ADCPPB3RESULT | ADC 后处理模块 3 结果寄存器 |
| 2Ch | ADCPPB4RESULT | ADC 后处理模块 4 结果寄存器 |

表 13-13 ADC_RESULT_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i、j、k、l、m、n | | 用于寄存器名称、偏移地址或地址时，表示属于重复寄存器组的寄存器数组值。寄存器组按层次组织，数组以公式表示。 |
| y | | 用于寄存器名称、偏移地址或地址时，表示寄存器数组值。 |

### 13.4.2.1 ADCRESULT0 寄存器

偏移地址：0h　复位值：0000h

ADC 结果 0 寄存器

表 13-14 ADCRESULT0 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-15 ADCRESULT0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 0，16 位 ADC 结果。ADC 完成 SOC0 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.2 ADCRESULT1 寄存器

偏移地址：2h　复位值：0000h

ADC 结果 1 寄存器

表 13-16 ADCRESULT1 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-17 ADCRESULT1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 1，16 位 ADC 结果。ADC 完成 SOC1 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.3 ADCRESULT2 寄存器

偏移地址：4h　复位值：0000h

ADC 结果 2 寄存器

表 13-18 ADCRESULT2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-19 ADCRESULT2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 2，16 位 ADC 结果。ADC 完成 SOC2 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.4 ADCRESULT3 寄存器

偏移地址：6h　复位值：0000h

ADC 结果 3 寄存器

表 13-20 ADCRESULT3 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-21 ADCRESULT3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 3，16 位 ADC 结果。ADC 完成 SOC3 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.5 ADCRESULT4 寄存器

偏移地址：8h　复位值：0000h

ADC 结果 4 寄存器

表 13-22 ADCRESULT4 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-23 ADCRESULT4 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 4，16 位 ADC 结果。ADC 完成 SOC4 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.6 ADCRESULT5 寄存器

偏移地址：Ah　复位值：0000h

ADC 结果 5 寄存器

表 13-24 ADCRESULT5 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-25 ADCRESULT5 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 5，16 位 ADC 结果。ADC 完成 SOC5 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.7 ADCRESULT6 寄存器

偏移地址：Ch　复位值：0000h

ADC 结果 6 寄存器

表 13-26 ADCRESULT6 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-27 ADCRESULT6 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 6，16 位 ADC 结果。ADC 完成 SOC6 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.8 ADCRESULT7 寄存器

偏移地址：Eh　复位值：0000h

ADC 结果 7 寄存器

表 13-28 ADCRESULT7 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-29 ADCRESULT7 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 7，16 位 ADC 结果。ADC 完成 SOC7 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.9 ADCRESULT8 寄存器

偏移地址：10h　复位值：0000h

ADC 结果 8 寄存器

表 13-30 ADCRESULT8 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-31 ADCRESULT8 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 8，16 位 ADC 结果。ADC 完成 SOC8 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.10 ADCRESULT9 寄存器

偏移地址：12h　复位值：0000h

ADC 结果 9 寄存器

表 13-32 ADCRESULT9 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-33 ADCRESULT9 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 9，16 位 ADC 结果。ADC 完成 SOC9 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.11 ADCRESULT10 寄存器

偏移地址：14h　复位值：0000h

ADC 结果 10 寄存器

表 13-34 ADCRESULT10 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-35 ADCRESULT10 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 10，16 位 ADC 结果。ADC 完成 SOC10 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.12 ADCRESULT11 寄存器

偏移地址：16h　复位值：0000h

ADC 结果 11 寄存器

表 13-36 ADCRESULT11 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-37 ADCRESULT11 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 11，16 位 ADC 结果。ADC 完成 SOC11 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.13 ADCRESULT12 寄存器

偏移地址：18h　复位值：0000h

ADC 结果 12 寄存器

表 13-38 ADCRESULT12 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-39 ADCRESULT12 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 12，16 位 ADC 结果。ADC 完成 SOC12 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.14 ADCRESULT13 寄存器

偏移地址：1Ah　复位值：0000h

ADC 结果 13 寄存器

表 13-40 ADCRESULT13 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-41 ADCRESULT13 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 13，16 位 ADC 结果。ADC 完成 SOC13 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

### 13.4.2.15 ADCRESULT14 寄存器

偏移地址：1Ch　复位值：0000h

ADC 结果 14 寄存器

表 13-42 ADCRESULT14 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-43 ADCRESULT14 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 14，16 位 ADC 结果。ADC 完成 SOC14 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

## 13.4.2.16 ADCRESULT15 寄存器

偏移地址：1Eh　复位值：0000h

ADC 结果 15 寄存器

表 13-44 ADCRESULT15 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-45 ADCRESULT15 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RESULT | R | 0h | ADC 结果 15，16 位 ADC 结果。ADC 完成 SOC15 转换后，将数字结果写入此字段。<br>复位类型：SYSRSn。 |

## 13.4.2.17 ADCPPB1RESULT 寄存器

偏移地址：20h　复位值：00000000h

ADC 后处理模块 1 结果寄存器

表 13-46 ADCPPB1RESULT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SIGN | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SIGN | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PPBRESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PPBRESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-47 ADCPPB1RESULT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | SIGN | R | 0h | 符号扩展位。这些位与位 16 的值相同。<br>注意：如果与此后处理模块关联的是 12 位转换，则 SIGN 位向下扩展至位 12，且均与位 12 的值相同。<br>复位类型：SYSRSn。 |
| 15-0 | PPBRESULT | R | 0h | ADC 后处理模块 1 的结果。偏移 / 参考值减法的转换后处理结果存储于此寄存器。相关 ADCRESULT 可用后，再经过 1 个 SYSCLK 周期，此结果才可用。如果通过轮询 ADCINTFLG 确定读取 PPBRESULT 的时机，可能需要添加一条 NOP 指令，确保更新后的处理结果已写入寄存器。<br>注意：若关联的是 12 位转换，PPBRESULT 仅使用位 12:0。<br>复位类型：SYSRSn。 |

## 13.4.2.18 ADCPPB2RESULT 寄存器

偏移地址：24h　复位值：00000000h

ADC 后处理模块 2 结果寄存器

表 13-48 ADCPPB2RESULT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SIGN | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SIGN | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PPBRESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PPBRESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-49 ADCPPB2RESULT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | SIGN | R | 0h | 符号扩展位。这些位与位 16 的值相同。<br>注意：如果与此后处理模块关联的是 12 位转换，则 SIGN 位向下扩展至位 12，且均与位 12 的值相同。<br>复位类型：SYSRSn。 |
| 15-0 | PPBRESULT | R | 0h | ADC 后处理模块 2 的结果。偏移 / 参考值减法的转换后处理结果存储于此寄存器。相关 ADCRESULT 可用后，再经过 1 个 SYSCLK 周期，此结果才可用。如果通过轮询 ADCINTFLG 确定读取 PPBRESULT 的时机，可能需要添加一条 NOP 指令，确保更新后的处理结果已写入寄存器。<br>注意：若关联的是 12 位转换，PPBRESULT 仅使用位 12:0。<br>复位类型：SYSRSn。 |

## 13.4.2.19 ADCPPB3RESULT 寄存器

偏移地址：28h　复位值：00000000h

ADC 后处理模块 3 结果寄存器

表 13-50 ADCPPB3RESULT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SIGN | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SIGN | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PPBRESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PPBRESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-51 ADCPPB3RESULT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | SIGN | R | 0h | 符号扩展位。这些位与位 16 的值相同。<br>注意：如果与此后处理模块关联的是 12 位转换，则 SIGN 位向下扩展至位 12，且均与位 12 的值相同。<br>复位类型：SYSRSn。 |
| 15-0 | PPBRESULT | R | 0h | ADC 后处理模块 3 的结果。偏移 / 参考值减法的转换后处理结果存储于此寄存器。相关 ADCRESULT 可用后，再经过 1 个 SYSCLK 周期，此结果才可用。如果通过轮询 ADCINTFLG 确定读取 PPBRESULT 的时机，可能需要添加一条 NOP 指令，确保更新后的处理结果已写入寄存器。<br>注意：若关联的是 12 位转换，PPBRESULT 仅使用位 12:0。<br>复位类型：SYSRSn。 |

## 13.4.2.20 ADCPPB4RESULT 寄存器

偏移地址：2Ch　复位值：00000000h

ADC 后处理模块 4 结果寄存器

表 13-52 ADCPPB4RESULT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SIGN | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SIGN | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PPBRESULT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PPBRESULT | | | | | | | |
| R-0h | | | | | | | |

表 13-53 ADCPPB4RESULT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | SIGN | R | 0h | 符号扩展位。这些位与位 16 的值相同。<br>注意：如果与此后处理模块关联的是 12 位转换，则 SIGN 位向下扩展至位 12，且均与位 12 的值相同。<br>复位类型：SYSRSn。 |
| 15-0 | PPBRESULT | R | 0h | ADC 后处理模块 4 的结果。偏移 / 参考值减法的转换后处理结果存储于此寄存器。相关 ADCRESULT 可用后，再经过 1 个 SYSCLK 周期，此结果才可用。如果通过轮询 ADCINTFLG 确定读取 PPBRESULT 的时机，可能需要添加一条 NOP 指令，确保更新后的处理结果已写入寄存器。<br>注意：若关联的是 12 位转换，PPBRESULT 仅使用位 12:0。<br>复位类型：SYSRSn。 |

### 13.4.3 ADC_REGS 寄存器

表中列出了存储器映射寄存器。未列出的偏移地址应视为保留位置，不应修改这些位置的寄存器内容。

表 13-54 ADC_REGS 寄存器

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | ADCCTL1 | ADC 控制 1 寄存器 |
| 2h | ADCCTL2 | ADC 控制 2 寄存器 |

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 4h | ADCBURSTCTL | ADC 突发控制寄存器 |
| 6h | ADCINTFLG | ADC 中断标志寄存器 |
| 8h | ADCINTFLGCLR | ADC 中断标志清除寄存器 |
| Ah | ADCINTOVF | ADC 中断溢出寄存器 |
| Ch | ADCINTOVFCLR | ADC 中断溢出清除寄存器 |
| Eh | ADCINTSEL1N2 | ADC 中断 1 和 2 选择寄存器 |
| 10h | ADCINTSEL3N4 | ADC 中断 3 和 4 选择寄存器 |
| 12h | ADCSOCPRICTL | ADC SOC 优先级控制寄存器 |
| 14h | ADCINTSOCSEL1 | ADC 中断 SOC 选择 1 寄存器 |
| 16h | ADCINTSOCSEL2 | ADC 中断 SOC 选择 2 寄存器 |
| 18h | ADCSOCFLG1 | ADC SOC 标志 1 寄存器 |
| 1Ah | ADCSOCFRC1 | ADC SOC 强制 1 寄存器 |
| 1Ch | ADCSOCOVF1 | ADC SOC 溢出 1 寄存器 |
| 1Eh | ADCSOCOVFCLR1 | ADC SOC 溢出清除 1 寄存器 |
| 20h | ADCSOC0CTL | ADC SOC0 控制寄存器 |
| 24h | ADCSOC1CTL | ADC SOC1 控制寄存器 |
| 28h | ADCSOC2CTL | ADC SOC2 控制寄存器 |
| 2Ch | ADCSOC3CTL | ADC SOC3 控制寄存器 |
| 30h | ADCSOC4CTL | ADC SOC4 控制寄存器 |
| 34h | ADCSOC5CTL | ADC SOC5 控制寄存器 |
| 38h | ADCSOC6CTL | ADC SOC6 控制寄存器 |
| 3Ch | ADCSOC7CTL | ADC SOC7 控制寄存器 |
| 40h | ADCSOC8CTL | ADC SOC8 控制寄存器 |
| 44h | ADCSOC9CTL | ADC SOC9 控制寄存器 |
| 48h | ADCSOC10CTL | ADC SOC10 控制寄存器 |
| 4Ch | ADCSOC11CTL | ADC SOC11 控制寄存器 |
| 50h | ADCSOC12CTL | ADC SOC12 控制寄存器 |
| 54h | ADCSOC13CTL | ADC SOC13 控制寄存器 |
| 58h | ADCSOC14CTL | ADC SOC14 控制寄存器 |
| 5Ch | ADCSOC15CTL | ADC SOC15 控制寄存器 |
| 60h | ADCEVTSTAT | ADC 事件状态寄存器 |
| 64h | ADCEVTCLR | ADC 事件清除寄存器 |
| 68h | ADCEVTSEL | ADC 事件选择寄存器 |

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 6Ch | ADCEVTINTSEL | ADC 事件中断选择寄存器 |
| 70h | ADCOSDETECT | ADC 开路 / 短路检测寄存器 |
| 72h | ADCCOUNTER | ADC 计数器寄存器 |
| 74h | ADCREV | ADC 修订版本寄存器 |
| 76h | ADCOFFTRIM | ADC 偏移修调寄存器 |
| 80h | ADCPPB1CONFIG | ADC PPB1 配置寄存器 |
| 82h | ADCPPB1STAMP | ADC PPB1 采样延迟时间戳寄存器 |
| 84h | ADCPPB1OFFCAL | ADC PPB1 偏移校准寄存器 |
| 86h | ADCPPB1OFFREF | ADC PPB1 偏移参考寄存器 |
| 88h | ADCPPB1TRIPHI | ADC PPB1 高限跳闸寄存器 |
| 8Ch | ADCPPB1TRIPLO | ADC PPB1 低限跳闸 / 触发时间戳寄存器 |
| 90h | ADCPPB2CONFIG | ADC PPB2 配置寄存器 |
| 92h | ADCPPB2STAMP | ADC PPB2 采样延迟时间戳寄存器 |
| 94h | ADCPPB2OFFCAL | ADC PPB2 偏移校准寄存器 |
| 96h | ADCPPB2OFFREF | ADC PPB2 偏移参考寄存器 |
| 98h | ADCPPB2TRIPHI | ADC PPB2 高限跳闸寄存器 |
| 9Ch | ADCPPB2TRIPLO | ADC PPB2 低限跳闸 / 触发时间戳寄存器 |
| A0h | ADCPPB3CONFIG | ADC PPB3 配置寄存器 |
| A2h | ADCPPB3STAMP | ADC PPB3 采样延迟时间戳寄存器 |
| A4h | ADCPPB3OFFCAL | ADC PPB3 偏移校准寄存器 |
| A6h | ADCPPB3OFFREF | ADC PPB3 偏移参考寄存器 |
| A8h | ADCPPB3TRIPHI | ADC PPB3 高限跳闸寄存器 |
| ACh | ADCPPB3TRIPLO | ADC PPB3 低限跳闸 / 触发时间戳寄存器 |
| B0h | ADCPPB4CONFIG | ADC PPB4 配置寄存器 |
| B2h | ADCPPB4STAMP | ADC PPB4 采样延迟时间戳寄存器 |
| B4h | ADCPPB4OFFCAL | ADC PPB4 偏移校准寄存器 |
| B6h | ADCPPB4OFFREF | ADC PPB4 偏移参考寄存器 |
| B8h | ADCPPB4TRIPHI | ADC PPB4 高限跳闸寄存器 |
| BCh | ADCPPB4TRIPLO | ADC PPB4 低限跳闸 / 触发时间戳寄存器 |
| DEh | ADCINTCYCLE | ADC 提前中断生成周期 |
| E4h | ADCINLTRIM2 | ADC 线性度修调 2 寄存器 |
| E8h | ADCINLTRIM3 | ADC 线性度修调 3 寄存器 |

表 13-55 ADC_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1C | W<br>1C | 写<br>1清除 |
| W1S | W<br>1S | 写<br>1置位 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i、j、k、l、m、n | | 用于寄存器名称、偏移地址或地址时，表示属于重复寄存器组的寄存器数组值。寄存器组按层次组织，数组以公式表示。 |
| y | | 用于寄存器名称、偏移地址或地址时，表示寄存器数组值。 |

### 13.4.3.1 ADCCTL1 寄存器

偏移地址：0h　复位值：0000h

ADC 控制 1 寄存器

表 13-56 ADCCTL1 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | ADCBSY | 保留 | ADCBSYCHN | | | |
| R-0h | | R-0h | R-0h | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADCPWDNZ | 保留 | | | | INTPULSEPOS | 保留 | |
| R/W-0h | R-0h | | | | R/W-0h | R-0h | |

表 13-57 ADCCTL1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | RESERVED | R | 0h | 保留。 |
| 13 | ADCBSY | R | 0h | ADC 忙状态。产生 ADC SOC 时置位，采样保持（S/H）脉冲下降沿之后四个 ADC 时钟，由硬件清零。ADC 状态机使用此位判断 ADC 是 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 否可进行采样。<br>0：ADC 可采样下一个通道。<br>1：ADC 忙，无法采样其他通道。<br>复位类型：SYSRSn。 |
| 12 | RESERVED | R | 0h | 保留。 |
| 11-8 | ADCBSYCHN | R | 0h | ADC 忙通道。产生 ADC 转换启动（SOC）时设置。<br>ADCBSY = 0：保存上一次完成转换的 SOC 值。<br>ADCBSY = 1：反映当前正在处理的 SOC。<br>0h：SOC0 正在处理或是上一次完成转换的 SOC。<br>1h：SOC1 正在处理或是上一次完成转换的 SOC。<br>2h：SOC2 正在处理或是上一次完成转换的 SOC。<br>3h：SOC3 正在处理或是上一次完成转换的 SOC。<br>4h：SOC4 正在处理或是上一次完成转换的 SOC。<br>5h：SOC5 正在处理或是上一次完成转换的 SOC。<br>6h：SOC6 正在处理或是上一次完成转换的 SOC。<br>7h：SOC7 正在处理或是上一次完成转换的 SOC。<br>8h：SOC8 正在处理或是上一次完成转换的 SOC。<br>9h：SOC9 正在处理或是上一次完成转换的 SOC。<br>Ah：SOC10 正在处理或是上一次完成转换的 SOC。<br>Bh：SOC11 正在处理或是上一次完成转换的 SOC。<br>Ch：SOC12 正在处理或是上一次完成转换的 SOC。<br>Dh：SOC13 正在处理或是上一次完成转换的 SOC。<br>Eh：SOC14 正在处理或是上一次完成转换的 SOC。<br>Fh：SOC15 正在处理或是上一次完成转换的 SOC。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 7 | ADCPWDNZ | R/W | 0h | ADC 掉电控制，低电平有效。控制模拟内核中全部模拟电路的上电和掉电。<br>0：内核中全部模拟电路掉电。<br>1：内核中全部模拟电路上电。<br>复位类型：SYSRSn。 |
| 6-3 | RESERVED | R | 0h | 保留。 |
| 2 | INTPULSEPOS | R/W | 0h | ADC 中断脉冲位置。<br>0：ADC 开始转换（采集窗口结束）后，再经过 ADCINTCYCLE.OFFSET 寄存器指定数量的 SYSCLK 周期，产生中断脉冲。<br>1：转换结束时，在 ADC 结果锁存到结果寄存器前 1 个周期产生中断脉冲。<br>复位类型：SYSRSn。 |
| 1-0 | RESERVED | R | 0h | 保留。 |

### 13.4.3.2 ADCCTL2 寄存器

偏移地址：2h　复位值：0000h

ADC 控制 2 寄存器

表 13-58 ADCCTL2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | 保留 | | | | |
| R-0h | | | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | 保留 | | PRESCALE | | | |
| R/W-0h | R/W-0h | R-0h | | R/W-0h | | | |

表 13-59 ADCCTL2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-13 | RESERVED | R | 0h | 保留。 |
| 12-8 | RESERVED | R | 0h | 保留。 |
| 7 | RESERVED | R/W | 0h | 保留。 |
| 6 | RESERVED | R/W | 0h | 保留。 |
| 5-4 | RESERVED | R | 0h | 保留。 |
| 3-0 | PRESCALE | R/W | 0h | ADC 时钟预分频。<br>0000：ADCCLK = 输入时钟 / 1.0<br>0001：保留。<br>0010：ADCCLK = 输入时钟 / 2.0<br>0011：保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 0100：ADCCLK = 输入时钟 / 3.0<br>0101：保留。<br>0110：ADCCLK = 输入时钟 / 4.0<br>0111：保留。<br>1000：ADCCLK = 输入时钟 / 5.0<br>1001：保留。<br>1010：ADCCLK = 输入时钟 / 6.0<br>1011：保留。<br>1100：ADCCLK = 输入时钟 / 7.0<br>1101：保留。<br>1110：ADCCLK = 输入时钟 / 8.0<br>1111：保留。<br>复位类型：SYSRSn。 |

### 13.4.3.3 ADCBURSTCTL 寄存器

偏移地址：4h　复位值：0000h

ADC 突发控制寄存器

表 13-60 ADCBURSTCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| BURSTEN | 保留 | | | BURSTSIZE | | | |
| R/W-0h | R-0h | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | BURSTTRIGSEL | | | | | |
| R-0h | | R/W-0h | | | | | |

表 13-61 ADCBURSTCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | BURSTEN | R/W | 0h | SOC 突发模式使能。<br>0：禁用突发模式。<br>1：使能突发模式。<br>复位类型：SYSRSn。 |
| 14-12 | RESERVED | R | 0h | 保留。 |
| 11-8 | BURSTSIZE | R/W | 0h | SOC 突发大小选择。决定突发转换序列启动后转换多少个 SOC。首个转换的 SOC 由轮询指针决定；每完成一个 SOC 转换，指针便前进。<br>0h：转换 1 个 SOC。<br>1h：转换 2 个 SOC。<br>2h：转换 3 个 SOC。<br>3h：转换 4 个 SOC。<br>4h：转换 5 个 SOC。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 5h：转换 6 个 SOC。<br>6h：转换 7 个 SOC。<br>7h：转换 8 个 SOC。<br>8h：转换 9 个 SOC。<br>9h：转换 10 个 SOC。<br>Ah：转换 11 个 SOC。<br>Bh：转换 12 个 SOC。<br>Ch：转换 13 个 SOC。<br>Dh：转换 14 个 SOC。<br>Eh：转换 15 个 SOC。<br>Fh：转换 16 个 SOC。<br>复位类型：SYSRSn。 |
| 7-6 | RESERVED | R | 0h | 保留。 |
| 5-0 | BURSTTRIGSEL | R/W | 0h | SOC 突发触发源选择，配置启动突发转换序列的触发源。<br>00h：BURSTTRIG0 — 仅软件<br>01h：BURSTTRIG1 — CPU1 定时器 0, TINT0n<br>02h：BURSTTRIG2 — CPU1 定时器 1, TINT1n<br>03h：BURSTTRIG3 — CPU1 定时器 2, TINT2n<br>04h：BURSTTRIG4 — GPIO, 输入 X-BAR INPUT5<br>05h：BURSTTRIG5 — ePWM1, ADCSOCA<br>06h：BURSTTRIG6 — ePWM1, ADCSOCB<br>07h：BURSTTRIG7 — ePWM2, ADCSOCA<br>08h：BURSTTRIG8 — ePWM2, ADCSOCB<br>09h：BURSTTRIG9 — ePWM3, ADCSOCA<br>0Ah：BURSTTRIG10 — ePWM3, ADCSOCB<br>0Bh：BURSTTRIG11 — ePWM4, ADCSOCA<br>0Ch：BURSTTRIG12 — ePWM4, ADCSOCB<br>0Dh：BURSTTRIG13 — ePWM5, ADCSOCA<br>0Eh：BURSTTRIG14 — ePWM5, ADCSOCB<br>0Fh：BURSTTRIG15 — ePWM6, ADCSOCA<br>10h：BURSTTRIG16 — ePWM6, ADCSOCB<br>11h：BURSTTRIG17 — ePWM7, ADCSOCA<br>12h：BURSTTRIG18 — ePWM7, ADCSOCB<br>13h - 3Fh：保留。<br>复位类型：SYSRSn。 |

### 13.4.3.4 ADCINTFLG 寄存器

偏移地址：6h　复位值：0000h

ADC 中断标志寄存器

表 13-62 ADCINTFLG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | ADCINT4 | ADCINT3 | ADCINT2 | ADCINT1 |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

表 13-63 ADCINTFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R | 0h | 保留。 |
| 3 | ADCINT4 | R | 0h | ADC 中断 4 标志。读取此标志可判断自上次清除以来是否产生过对应 ADCINT 脉冲。<br>0：未产生 ADC 中断脉冲。<br>1：已产生 ADC 中断脉冲。<br>如果 ADC 中断配置为连续中断模式（INTSELxNy 寄存器），即使标志已置位，每次选定的 EOC 事件仍会产生中断脉冲。若未使能连续模式，则在用户通过 ADCINTFLGCLR 清除此标志之前，不再产生中断脉冲，而是在 ADCINTOVF 中记录 ADC 中断溢出事件。<br>复位类型：SYSRSn。 |
| 2 | ADCINT3 | R | 0h | ADC 中断 3 标志。读取此标志可判断自上次清除以来是否产生过对应 ADCINT 脉冲。<br>0：未产生 ADC 中断脉冲。<br>1：已产生 ADC 中断脉冲。<br>如果 ADC 中断配置为连续中断模式（INTSELxNy 寄存器），即使标志已置位，每次选定的 EOC 事件仍会产生中断脉冲。若未使能连续模式，则在用户通过 ADCINTFLGCLR 清除此标志之前，不再产生中断脉冲，而是在 ADCINTOVF 中记录 ADC 中断溢出事件。<br>复位类型：SYSRSn。 |
| 1 | ADCINT2 | R | 0h | ADC 中断 2 标志。读取此标志可判断自上次清除以来是否产生过对应 ADCINT 脉冲。<br>0：未产生 ADC 中断脉冲。<br>1：已产生 ADC 中断脉冲。<br>如果 ADC 中断配置为连续中断模式（INTSELxNy 寄存器），即使标志已置位，每次选定的 EOC 事件仍会产生中断脉冲。若未使能连续模式，则在用户通过 ADCINTFLGCLR 清除此标志之前，不再产生中断脉冲，而是在 ADCINTOVF 中记录 ADC 中断溢出事件。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 0 | ADCINT1 | R | 0h | ADC 中断 1 标志。读取此标志可判断自上次清除以来是否产生过对应 ADCINT 脉冲。<br>0：未产生 ADC 中断脉冲。<br>1：已产生 ADC 中断脉冲。<br>如果 ADC 中断配置为连续中断模式（INTSELxNy 寄存器），即使标志已置位，每次选定的 EOC 事件仍会产生中断脉冲。若未使能连续模式，则在用户通过 ADCINTFLGCLR 清除此标志之前，不再产生中断脉冲，而是在 ADCINTOVF 中记录 ADC 中断溢出事件。<br>复位类型：SYSRSn。 |

### 13.4.3.5 ADCINTFLGCLR 寄存器

偏移地址：8h　复位值：0000h

ADC 中断标志清除寄存器

表 13-64 ADCINTFLGCLR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | ADCINT4 | ADCINT3 | ADCINT2 | ADCINT1 |
| R-0h | | | | R-0/W1C-0h | R-0/W1C-0h | R-0/W1C-0h | R-0/W1C-0h |

表 13-65 ADCINTFLGCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R | 0h | 保留。 |
| 3 | ADCINT4 | R-0/W1C | 0h | ADC 中断 4 标志清除。读取返回 0。<br>0：无操作。<br>1：清除 ADCINTFLG 中对应标志位。若软件置位清除位与硬件置位标志发生在同一周期，硬件优先，溢出位不会置位。<br>复位类型：SYSRSn。 |
| 2 | ADCINT3 | R-0/W1C | 0h | ADC 中断 3 标志清除。读取返回 0。<br>0：无操作。<br>1：清除 ADCINTFLG 中对应标志位。若软件置位清除位与硬件置位标志发生在同一周期，硬件优先，溢出位不会置位。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | ADCINT2 | R-0/W1C | 0h | ADC 中断 2 标志清除。读取返回 0。<br>0：无操作。<br>1：清除 ADCINTFLG 中对应标志位。若软件置位清除位与硬件置位标志发生在同一周期，硬件优先，溢出位不会置位。<br>复位类型：SYSRSn。 |
| 0 | ADCINT1 | R-0/W1C | 0h | ADC 中断 1 标志清除。读取返回 0。<br>0：无操作。<br>1：清除 ADCINTFLG 中对应标志位。若软件置位清除位与硬件置位标志发生在同一周期，硬件优先，溢出位不会置位。<br>复位类型：SYSRSn。 |

### 13.4.3.6 ADCINTOVF 寄存器

偏移地址：Ah　复位值：0000h

ADC 中断溢出寄存器

表 13-66 ADCINTOVF 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | ADCINT4 | ADCINT3 | ADCINT2 | ADCINT1 |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |

表 13-67 ADCINTOVF 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R | 0h | 保留。 |
| 3 | ADCINT4 | R | 0h | ADC 中断 4 溢出标志。指示产生 ADCINT 脉冲时是否发生溢出。对应 ADCINTFLG 位已置位时，若又产生选定的 EOC 触发，则发生溢出。<br>0：未检测到 ADC 中断溢出事件。<br>1：检测到 ADC 中断溢出事件。<br>溢出位不受连续模式位状态影响，无论连续模式如何设置，都可产生溢出条件。<br>复位类型：SYSRSn。 |
| 2 | ADCINT3 | R | 0h | ADC 中断 3 溢出标志。指示产生 ADCINT 脉冲时是否发生溢出。对应 ADCINTFLG 位已置位时，若又产生选定的 EOC 触发，则发生溢出。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 0：未检测到 ADC 中断溢出事件。<br>1：检测到 ADC 中断溢出事件。<br>溢出位不受连续模式位状态影响，无论连续模式如何设置，都可产生溢出条件。<br>复位类型：SYSRSn。 |
| 1 | ADCINT2 | R | 0h | ADC 中断 2 溢出标志。指示产生 ADCINT 脉冲时是否发生溢出。对应 ADCINTFLG 位已置位时，若又产生选定的 EOC 触发，则发生溢出。<br>0：未检测到 ADC 中断溢出事件。<br>1：检测到 ADC 中断溢出事件。<br>溢出位不受连续模式位状态影响，无论连续模式如何设置，都可产生溢出条件。<br>复位类型：SYSRSn。 |
| 0 | ADCINT1 | R | 0h | ADC 中断 1 溢出标志。指示产生 ADCINT 脉冲时是否发生溢出。对应 ADCINTFLG 位已置位时，若又产生选定的 EOC 触发，则发生溢出。<br>0：未检测到 ADC 中断溢出事件。<br>1：检测到 ADC 中断溢出事件。<br>溢出位不受连续模式位状态影响，无论连续模式如何设置，都可产生溢出条件。<br>复位类型：SYSRSn。 |

### 13.4.3.7 ADCINTOVFCLR 寄存器

偏移地址：Ch　复位值：0000h

ADC 中断溢出清除寄存器

表 13-68 ADCINTOVFCLR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | ADCINT4 | ADCINT3 | ADCINT2 | ADCINT1 |
| R-0h | | | | R-0/W1C-0h | R-0/W1C-0h | R-0/W1C-0h | R-0/W1C-0h |

表 13-69 ADCINTOVFCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 3 | ADCINT4 | R-0/W1C | 0h | ADC 中断 4 溢出清除。<br>0：无操作。<br>1：清除 ADCINTOVF 中对应溢出位。若软件设置此位与硬件设置 ADCINTOVF 溢出位发生在同一时钟周期，硬件优先，ADCINTOVF 位将置位。<br>复位类型：SYSRSn。 |
| 2 | ADCINT3 | R-0/W1C | 0h | ADC 中断 3 溢出清除。<br>0：无操作。<br>1：清除 ADCINTOVF 中对应溢出位。若软件设置此位与硬件设置 ADCINTOVF 溢出位发生在同一时钟周期，硬件优先，ADCINTOVF 位将置位。<br>复位类型：SYSRSn。 |
| 1 | ADCINT2 | R-0/W1C | 0h | ADC 中断 2 溢出清除。<br>0：无操作。<br>1：清除 ADCINTOVF 中对应溢出位。若软件设置此位与硬件设置 ADCINTOVF 溢出位发生在同一时钟周期，硬件优先，ADCINTOVF 位将置位。<br>复位类型：SYSRSn。 |
| 0 | ADCINT1 | R-0/W1C | 0h | ADC 中断 1 溢出清除。<br>0：无操作。<br>1：清除 ADCINTOVF 中对应溢出位。若软件设置此位与硬件设置 ADCINTOVF 溢出位发生在同一时钟周期，硬件优先，ADCINTOVF 位将置位。<br>复位类型：SYSRSn。 |

### 13.4.3.8 ADCINTSEL1N2 寄存器

偏移地址：Eh　复位值：0000h

ADC 中断 1 和 2 选择寄存器

表 13-70 ADCINTSEL1N2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | INT2CONT | INT2E | 保留 | INT2SEL | | | |
| R-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | INT1CONT | INT1E | 保留 | INT1SEL | | | |
| R-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | | | |

表 13-71 ADCINTSEL1N2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留。 |
| 14 | INT2CONT | R/W | 0h | ADCINT2 连续中断模式。<br>0：用户清除 ADCINTFLG 中的 ADCINT2 标志之前，不再产生 ADCINT2 脉冲。<br>1：无论标志位是否已清除，每次产生 EOC 脉冲时均产生 ADCINT2 脉冲。<br>复位类型：SYSRSn。 |
| 13 | INT2E | R/W | 0h | ADCINT2 中断使能。<br>0：禁用 ADCINT2。<br>1：使能 ADCINT2。<br>复位类型：SYSRSn。 |
| 12 | RESERVED | R | 0h | 保留。 |
| 11-8 | INT2SEL | R/W | 0h | ADCINT2 的 EOC 源选择。<br>0h：EOC0 触发 ADCINT2。<br>1h：EOC1 触发 ADCINT2。<br>2h：EOC2 触发 ADCINT2。<br>3h：EOC3 触发 ADCINT2。<br>4h：EOC4 触发 ADCINT2。<br>5h：EOC5 触发 ADCINT2。<br>6h：EOC6 触发 ADCINT2。<br>7h：EOC7 触发 ADCINT2。<br>8h：EOC8 触发 ADCINT2。<br>9h：EOC9 触发 ADCINT2。<br>Ah：EOC10 触发 ADCINT2。<br>Bh：EOC11 触发 ADCINT2。<br>Ch：EOC12 触发 ADCINT2。<br>Dh：EOC13 触发 ADCINT2。<br>Eh：EOC14 触发 ADCINT2。<br>Fh：EOC15 触发 ADCINT2。<br>复位类型：SYSRSn。 |
| 7 | RESERVED | R | 0h | 保留。 |
| 6 | INT1CONT | R/W | 0h | ADCINT1 连续中断模式。<br>0：用户清除 ADCINTFLG 中的 ADCINT1 标志之前，不再产生 ADCINT1 脉冲。<br>1：无论标志位是否已清除，每次产生 EOC 脉冲时均产生 ADCINT1 脉冲。<br>复位类型：SYSRSn。 |
| 5 | INT1E | R/W | 0h | ADCINT1 中断使能。<br>0：禁用 ADCINT1。<br>1：使能 ADCINT1。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 4 | RESERVED | R | 0h | 保留。 |
| 3-0 | INT1SEL | R/W | 0h | ADCINT1 的 EOC 源选择。<br>0h：EOC0 触发 ADCINT1。<br>1h：EOC1 触发 ADCINT1。<br>2h：EOC2 触发 ADCINT1。<br>3h：EOC3 触发 ADCINT1。<br>4h：EOC4 触发 ADCINT1。<br>5h：EOC5 触发 ADCINT1。<br>6h：EOC6 触发 ADCINT1。<br>7h：EOC7 触发 ADCINT1。<br>8h：EOC8 触发 ADCINT1。<br>9h：EOC9 触发 ADCINT1。<br>Ah：EOC10 触发 ADCINT1。<br>Bh：EOC11 触发 ADCINT1。<br>Ch：EOC12 触发 ADCINT1。<br>Dh：EOC13 触发 ADCINT1。<br>Eh：EOC14 触发 ADCINT1。<br>Fh：EOC15 触发 ADCINT1。<br>复位类型：SYSRSn。 |

### 13.4.3.9 ADCINTSEL3N4 寄存器

偏移地址：10h　复位值：0000h

ADC 中断 3 和 4 选择寄存器

表 13-72 ADCINTSEL3N4 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | INT4CONT | INT4E | 保留 | INT4SEL | | | |
| R-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | INT3CONT | INT3E | 保留 | INT3SEL | | | |
| R-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | | | |

表 13-73 ADCINTSEL3N4 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留。 |
| 14 | INT4CONT | R/W | 0h | ADCINT4 连续中断模式。<br>0：用户清除 ADCINTFLG 中的 ADCINT4 标志之前，不再产生 ADCINT4 脉冲。<br>1：无论标志位是否已清除，每次产生 EOC 脉冲时均产生 ADCINT4 脉冲。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 13 | INT4E | R/W | 0h | ADCINT4 中断使能。<br>0：禁用 ADCINT4。<br>1：使能 ADCINT4。<br>复位类型：SYSRSn。 |
| 12 | RESERVED | R | 0h | 保留。 |
| 11-8 | INT4SEL | R/W | 0h | ADCINT4 的 EOC 源选择。<br>0h：EOC0 触发 ADCINT4。<br>1h：EOC1 触发 ADCINT4。<br>2h：EOC2 触发 ADCINT4。<br>3h：EOC3 触发 ADCINT4。<br>4h：EOC4 触发 ADCINT4。<br>5h：EOC5 触发 ADCINT4。<br>6h：EOC6 触发 ADCINT4。<br>7h：EOC7 触发 ADCINT4。<br>8h：EOC8 触发 ADCINT4。<br>9h：EOC9 触发 ADCINT4。<br>Ah：EOC10 触发 ADCINT4。<br>Bh：EOC11 触发 ADCINT4。<br>Ch：EOC12 触发 ADCINT4。<br>Dh：EOC13 触发 ADCINT4。<br>Eh：EOC14 触发 ADCINT4。<br>Fh：EOC15 触发 ADCINT4。<br>复位类型：SYSRSn。 |
| 7 | RESERVED | R | 0h | 保留。 |
| 6 | INT3CONT | R/W | 0h | ADCINT3 连续中断模式。<br>0：用户清除 ADCINTFLG 中的 ADCINT3 标志之前，不再产生 ADCINT3 脉冲。<br>1：无论标志位是否已清除，每次产生 EOC 脉冲时均产生 ADCINT3 脉冲。<br>复位类型：SYSRSn。 |
| 5 | INT3E | R/W | 0h | ADCINT3 中断使能。<br>0：禁用 ADCINT3。<br>1：使能 ADCINT3。<br>复位类型：SYSRSn。 |
| 4 | RESERVED | R | 0h | 保留。 |
| 3-0 | INT3SEL | R/W | 0h | ADCINT3 的 EOC 源选择。<br>0h：EOC0 触发 ADCINT3。<br>1h：EOC1 触发 ADCINT3。<br>2h：EOC2 触发 ADCINT3。<br>3h：EOC3 触发 ADCINT3。<br>4h：EOC4 触发 ADCINT3。<br>5h：EOC5 触发 ADCINT3。<br>6h：EOC6 触发 ADCINT3。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 7h：EOC7 触发 ADCINT3。<br>8h：EOC8 触发 ADCINT3。<br>9h：EOC9 触发 ADCINT3。<br>Ah：EOC10 触发 ADCINT3。<br>Bh：EOC11 触发 ADCINT3。<br>Ch：EOC12 触发 ADCINT3。<br>Dh：EOC13 触发 ADCINT3。<br>Eh：EOC14 触发 ADCINT3。<br>Fh：EOC15 触发 ADCINT3。<br>复位类型：SYSRSn。 |

### 13.4.3.10 ADCSOCPRICTL 寄存器

偏移地址：12h　复位值：0200h

ADC SOC 优先级控制寄存器

表 13-74 ADCSOCPRICTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | RRPOINTER | |
| R-0h | | | | | | R-10h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RRPOINTER | | | SOCPRIORITY | | | | |
| R-10h | | | R/W-0h | | | | |

表 13-75 ADCSOCPRICTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-10 | RESERVED | R | 0h | 保留。 |
| 9-5 | RRPOINTER | R | 10h | 轮询指针。保存上一个完成转换的轮询 SOCx，供轮询机制确定转换顺序。<br>00h：SOC0 为上一个完成转换的轮询 SOC，SOC1 具有最高轮询优先级。<br>01h：SOC1 为上一个完成转换的轮询 SOC，SOC2 具有最高轮询优先级。<br>02h：SOC2 为上一个完成转换的轮询 SOC，SOC3 具有最高轮询优先级。<br>03h：SOC3 为上一个完成转换的轮询 SOC，SOC4 具有最高轮询优先级。<br>04h：SOC4 为上一个完成转换的轮询 SOC，SOC5 具有最高轮询优先级。<br>05h：SOC5 为上一个完成转换的轮询 SOC，SOC6 具有最高轮询优先级。<br>06h：SOC6 为上一个完成转换的轮询 SOC， |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | SOC7 具有最高轮询优先级。<br>07h：SOC7 为上一个完成转换的轮询 SOC，SOC8 具有最高轮询优先级。<br>08h：SOC8 为上一个完成转换的轮询 SOC，SOC9 具有最高轮询优先级。<br>09h：SOC9 为上一个完成转换的轮询 SOC，SOC10 具有最高轮询优先级。<br>0Ah：SOC10 为上一个完成转换的轮询 SOC，SOC11 具有最高轮询优先级。<br>0Bh：SOC11 为上一个完成转换的轮询 SOC，SOC12 具有最高轮询优先级。<br>0Ch：SOC12 为上一个完成转换的轮询 SOC，SOC13 具有最高轮询优先级。<br>0Dh：SOC13 为上一个完成转换的轮询 SOC，SOC14 具有最高轮询优先级。<br>0Eh：SOC14 为上一个完成转换的轮询 SOC，SOC15 具有最高轮询优先级。<br>0Fh：SOC15 为上一个完成转换的轮询 SOC，SOC0 具有最高轮询优先级。<br>10h：复位值，表示尚无 SOC 完成转换，SOC0 具有最高轮询优先级。通过 SOFTPRES 复位 ADC 模块或写入 ADCSOCPRICTL 寄存器时，设为此值。后一种情况下，若当前正在转换，则先完成转换，再使新优先级生效。<br>其他值：无效。<br>复位类型：SYSRSn。 |
| 4-0 | SOCPRIORITY | R/W | 0h | SOC 优先级。确定 SOCx 高优先级模式与轮询仲裁的分界点。<br>00h：所有通道均采用轮询模式。<br>01h：SOC0 为高优先级，SOC1-SOC15 采用轮询模式。<br>02h：SOC0-SOC1 为高优先级，SOC2-SOC15 采用轮询模式。<br>03h：SOC0-SOC2 为高优先级，SOC3-SOC15 采用轮询模式。<br>04h：SOC0-SOC3 为高优先级，SOC4-SOC15 采用轮询模式。<br>05h：SOC0-SOC4 为高优先级，SOC5-SOC15 采用轮询模式。<br>06h：SOC0-SOC5 为高优先级，SOC6-SOC15 采用轮询模式。<br>07h：SOC0-SOC6 为高优先级，SOC7-SOC15 采用轮询模式。<br>08h：SOC0-SOC7 为高优先级，SOC8-SOC15 采用轮询模式。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 09h：SOC0-SOC8 为高优先级，SOC9-SOC15 采用轮询模式。<br>0Ah：SOC0-SOC9 为高优先级，SOC10-SOC15 采用轮询模式。<br>0Bh：SOC0-SOC10 为高优先级，SOC11-SOC15 采用轮询模式。<br>0Ch：SOC0-SOC11 为高优先级，SOC12-SOC15 采用轮询模式。<br>0Dh：SOC0-SOC12 为高优先级，SOC13-SOC15 采用轮询模式。<br>0Eh：SOC0-SOC13 为高优先级，SOC14-SOC15 采用轮询模式。<br>0Fh：SOC0-SOC14 为高优先级，SOC15 采用轮询模式。<br>10h：所有 SOC 均为高优先级，按 SOC 编号仲裁。<br>其他值：无效选择。<br>复位类型：SYSRSn。 |

### 13.4.3.11 ADCINTSOCSEL1 寄存器

偏移地址：14h　复位值：0000h

ADC 中断 SOC 选择 1 寄存器

表 13-76 ADCINTSOCSEL1 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SOC7 | | SOC6 | | SOC5 | | SOC4 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SOC3 | | SOC2 | | SOC1 | | SOC0 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 13-77 ADCINTSOCSEL1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 15-14 | SOC7 | R/W | 0h | SOC7 的 ADC 中断触发选择。选择触发 SOC7 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC7，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC7。<br>10：ADCINT2 触发 SOC7。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：无效选择。<br>复位类型：SYSRSn。 |
| 13-12 | SOC6 | R/W | 0h | SOC6 的 ADC 中断触发选择。选择触发 SOC6 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC6，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC6。<br>10：ADCINT2 触发 SOC6。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 11-10 | SOC5 | R/W | 0h | SOC5 的 ADC 中断触发选择。选择触发 SOC5 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC5，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC5。<br>10：ADCINT2 触发 SOC5。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 9-8 | SOC4 | R/W | 0h | SOC4 的 ADC 中断触发选择。选择触发 SOC4 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC4，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC4。<br>10：ADCINT2 触发 SOC4。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 7-6 | SOC3 | R/W | 0h | SOC3 的 ADC 中断触发选择。选择触发 SOC3 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC3，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC3。<br>10：ADCINT2 触发 SOC3。<br>11：无效选择。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 5-4 | SOC2 | R/W | 0h | SOC2 的 ADC 中断触发选择。选择触发 SOC2 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC2，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC2。<br>10：ADCINT2 触发 SOC2。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 3-2 | SOC1 | R/W | 0h | SOC1 的 ADC 中断触发选择。选择触发 SOC1 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC1，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC1。<br>10：ADCINT2 触发 SOC1。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 1-0 | SOC0 | R/W | 0h | SOC0 的 ADC 中断触发选择。选择触发 SOC0 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC0，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC0。<br>10：ADCINT2 触发 SOC0。<br>11：无效选择。<br>复位类型：SYSRSn。 |

### 13.4.3.12 ADCINTSOCSEL2 寄存器

偏移地址：16h　复位值：0000h

ADC 中断 SOC 选择 2 寄存器

表 13-78 ADCINTSOCSEL2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SOC15 | | SOC14 | | SOC13 | | SOC12 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SOC11 | | SOC10 | | SOC9 | | SOC8 | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 13-79 ADCINTSOCSEL2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | SOC15 | R/W | 0h | SOC15 的 ADC 中断触发选择。选择触发 SOC15 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC15，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC15。<br>10：ADCINT2 触发 SOC15。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 13-12 | SOC14 | R/W | 0h | SOC14 的 ADC 中断触发选择。选择触发 SOC14 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC14，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC14。<br>10：ADCINT2 触发 SOC14。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 11-10 | SOC13 | R/W | 0h | SOC13 的 ADC 中断触发选择。选择触发 SOC13 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC13，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC13。<br>10：ADCINT2 触发 SOC13。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 9-8 | SOC12 | R/W | 0h | SOC12 的 ADC 中断触发选择。选择触发 SOC12 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 00：无 ADCINT 触发 SOC12，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC12。<br>10：ADCINT2 触发 SOC12。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 7-6 | SOC11 | R/W | 0h | SOC11 的 ADC 中断触发选择。选择触发 SOC11 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC11，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC11。<br>10：ADCINT2 触发 SOC11。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 5-4 | SOC10 | R/W | 0h | SOC10 的 ADC 中断触发选择。选择触发 SOC10 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC10，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC10。<br>10：ADCINT2 触发 SOC10。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 3-2 | SOC9 | R/W | 0h | SOC9 的 ADC 中断触发选择。选择触发 SOC9 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC9，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC9。<br>10：ADCINT2 触发 SOC9。<br>11：无效选择。<br>复位类型：SYSRSn。 |
| 1-0 | SOC8 | R/W | 0h | SOC8 的 ADC 中断触发选择。选择触发 SOC8 的 ADCINT，也可不选择。此触发源是 ADCSOCxCTL 中 TRIGSEL 字段所选触发源之外的附加触发源。<br>00：无 ADCINT 触发 SOC8，仅由 TRIGSEL 决定 SOC0 的触发。<br>01：ADCINT1 触发 SOC8。<br>10：ADCINT2 触发 SOC8。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 11：无效选择。<br>复位类型：SYSRSn。 |

### 13.4.3.13 ADCSOCFLG1 寄存器

偏移地址：18h　复位值：0000h

ADC SOC 标志 1 寄存器

表 13-80 ADCSOCFLG1 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SOC15 | SOC14 | SOC13 | SOC12 | SOC11 | SOC10 | SOC9 | SOC8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SOC7 | SOC6 | SOC5 | SOC4 | SOC3 | SOC2 | SOC1 | SOC0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 13-81 ADCSOCFLG1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 15 | SOC15 | R | 0h | SOC15 转换启动标志，指示 SOC15 转换状态。<br>0：SOC15 无待处理采样。<br>1：SOC15 已收到触发，有待处理采样。SOC15 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 14 | SOC14 | R | 0h | SOC14 转换启动标志，指示 SOC14 转换状态。<br>0：SOC14 无待处理采样。<br>1：SOC14 已收到触发，有待处理采样。SOC14 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 13 | SOC13 | R | 0h | SOC13 转换启动标志，指示 SOC13 转换状态。<br>0：SOC13 无待处理采样。<br>1：SOC13 已收到触发，有待处理采样。SOC13 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 12 | SOC12 | R | 0h | SOC12 转换启动标志，指示 SOC12 转换状态。<br>0：SOC12 无待处理采样。<br>1：SOC12 已收到触发，有待处理采样。SOC12 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 11 | SOC11 | R | 0h | SOC11 转换启动标志，指示 SOC11 转换状态。<br>0：SOC11 无待处理采样。<br>1：SOC11 已收到触发，有待处理采样。SOC11 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 10 | SOC10 | R | 0h | SOC10 转换启动标志，指示 SOC10 转换状态。<br>0：SOC10 无待处理采样。<br>1：SOC10 已收到触发，有待处理采样。SOC10 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 9 | SOC9 | R | 0h | SOC9 转换启动标志，指示 SOC9 转换状态。<br>0：SOC9 无待处理采样。<br>1：SOC9 已收到触发，有待处理采样。SOC9 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 8 | SOC8 | R | 0h | SOC8 转换启动标志，指示 SOC8 转换状态。<br>0：SOC8 无待处理采样。<br>1：SOC8 已收到触发，有待处理采样。SOC8 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 7 | SOC7 | R | 0h | SOC7 转换启动标志，指示 SOC7 转换状态。<br>0：SOC7 无待处理采样。<br>1：SOC7 已收到触发，有待处理采样。SOC7 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 6 | SOC6 | R | 0h | SOC6 转换启动标志，指示 SOC6 转换状态。<br>0：SOC6 无待处理采样。<br>1：SOC6 已收到触发，有待处理采样。SOC6 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 5 | SOC5 | R | 0h | SOC5 转换启动标志，指示 SOC5 转换状态。<br>0：SOC5 无待处理采样。<br>1：SOC5 已收到触发，有待处理采样。SOC5 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 4 | SOC4 | R | 0h | SOC4 转换启动标志，指示 SOC4 转换状态。<br>0：SOC4 无待处理采样。<br>1：SOC4 已收到触发，有待处理采样。SOC4 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 3 | SOC3 | R | 0h | SOC3 转换启动标志，指示 SOC3 转换状态。<br>0：SOC3 无待处理采样。<br>1：SOC3 已收到触发，有待处理采样。SOC3 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 2 | SOC2 | R | 0h | SOC2 转换启动标志，指示 SOC2 转换状态。<br>0：SOC2 无待处理采样。<br>1：SOC2 已收到触发，有待处理采样。SOC2 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 1 | SOC1 | R | 0h | SOC1 转换启动标志，指示 SOC1 转换状态。<br>0：SOC1 无待处理采样。<br>1：SOC1 已收到触发，有待处理采样。SOC1 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 0 | SOC0 | R | 0h | SOC0 转换启动标志，指示 SOC0 转换状态。<br>0：SOC0 无待处理采样。<br>1：SOC0 已收到触发，有待处理采样。SOC0 转换开始时自动清零。<br>若同一周期同时收到置位和清除请求，无论各自来源如何，此位均置位，清除请求被忽略。此时，不论此位此前是否已置位，ADCSOCOVF1 中的溢出位均不受影响。<br>复位类型：SYSRSn。 |

### 13.4.3.14 ADCSOCFRC1 寄存器

偏移地址：1Ah　复位值：0000h

ADC SOC 强制 1 寄存器

表 13-82 ADCSOCFRC1 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SOC15 | SOC14 | SOC13 | SOC12 | SOC11 | SOC10 | SOC9 | SOC8 |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SOC7 | SOC6 | SOC5 | SOC4 | SOC3 | SOC2 | SOC1 | SOC0 |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 13-83 ADCSOCFRC1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | SOC15 | R-0/W1S | 0h | SOC15 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC15 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC15 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC15 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 14 | SOC14 | R-0/W1S | 0h | SOC14 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC14 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC14 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC14 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 13 | SOC13 | R-0/W1S | 0h | SOC13 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC13 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC13 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC13 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 12 | SOC12 | R-0/W1S | 0h | SOC12 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC12 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC12 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC12 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 11 | SOC11 | R-0/W1S | 0h | SOC11 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC11 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC11 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC11 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 10 | SOC10 | R-0/W1S | 0h | SOC10 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC10 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：将 SOC10 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC10 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 9 | SOC9 | R-0/W1S | 0h | SOC9 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC9 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC9 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC9 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 8 | SOC8 | R-0/W1S | 0h | SOC8 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC8 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC8 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC8 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 7 | SOC7 | R-0/W1S | 0h | SOC7 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC7 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC7 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC7 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | ADCSOCFLG1 位此前是否已置位，<br>ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 6 | SOC6 | R-0/W1S | 0h | SOC6 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC6 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC6 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC6 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 5 | SOC5 | R-0/W1S | 0h | SOC5 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC5 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC5 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC5 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 4 | SOC4 | R-0/W1S | 0h | SOC4 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC4 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC4 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC4 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 3 | SOC3 | R-0/W1S | 0h | SOC3 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC3 标志强制置 1，可 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC3 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC3 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 2 | SOC2 | R-0/W1S | 0h | SOC2 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC2 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC2 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC2 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 1 | SOC1 | R-0/W1S | 0h | SOC1 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC1 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC1 标志置 1，获得优先权后启动转换。<br>若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC1 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |
| 0 | SOC0 | R-0/W1S | 0h | SOC0 强制转换启动位。写入 1 将 ADCSOCFLG1 中的 SOC0 标志强制置 1，可用于软件启动转换。忽略写入 0，读取始终为 0。<br>0：无操作。<br>1：将 SOC0 标志置 1，获得优先权后启动转换。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 若软件设置此位与硬件清除 ADCSOCFLG1 的 SOC0 位发生在同一周期，软件优先，ADCSOCFLG1 位将置位。此时，不论 ADCSOCFLG1 位此前是否已置位，ADCSOCOVF1 的溢出位均不受影响。<br>复位类型：SYSRSn。 |

### 13.4.3.15 ADCSOCOVF1 寄存器

偏移地址：1Ch　复位值：0000h

ADC SOC 溢出 1 寄存器

表 13-84 ADCSOCOVF1 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SOC15 | SOC14 | SOC13 | SOC12 | SOC11 | SOC10 | SOC9 | SOC8 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SOC7 | SOC6 | SOC5 | SOC4 | SOC3 | SOC2 | SOC1 | SOC0 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 13-85 ADCSOCOVF1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 15 | SOC15 | R | 0h | SOC15 转换启动溢出标志。表示已有 SOC15 事件待处理时，硬件又产生 SOC15 事件。<br>0：无 SOC15 事件溢出。<br>1：发生 SOC15 事件溢出。<br>溢出不会阻止处理 SOC15 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 14 | SOC14 | R | 0h | SOC14 转换启动溢出标志。表示已有 SOC14 事件待处理时，硬件又产生 SOC14 事件。<br>0：无 SOC14 事件溢出。<br>1：发生 SOC14 事件溢出。<br>溢出不会阻止处理 SOC14 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 13 | SOC13 | R | 0h | SOC13 转换启动溢出标志。表示已有 SOC13 事件待处理时，硬件又产生 SOC13 事件。<br>0：无 SOC13 事件溢出。<br>1：发生 SOC13 事件溢出。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 溢出不会阻止处理 SOC13 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 12 | SOC12 | R | 0h | SOC12 转换启动溢出标志。表示已有 SOC12 事件待处理时，硬件又产生 SOC12 事件。<br>0：无 SOC12 事件溢出。<br>1：发生 SOC12 事件溢出。<br>溢出不会阻止处理 SOC12 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 11 | SOC11 | R | 0h | SOC11 转换启动溢出标志。表示已有 SOC11 事件待处理时，硬件又产生 SOC11 事件。<br>0：无 SOC11 事件溢出。<br>1：发生 SOC11 事件溢出。<br>溢出不会阻止处理 SOC11 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 10 | SOC10 | R | 0h | SOC10 转换启动溢出标志。表示已有 SOC10 事件待处理时，硬件又产生 SOC10 事件。<br>0：无 SOC10 事件溢出。<br>1：发生 SOC10 事件溢出。<br>溢出不会阻止处理 SOC10 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 9 | SOC9 | R | 0h | SOC9 转换启动溢出标志。表示已有 SOC9 事件待处理时，硬件又产生 SOC9 事件。<br>0：无 SOC9 事件溢出。<br>1：发生 SOC9 事件溢出。<br>溢出不会阻止处理 SOC9 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 8 | SOC8 | R | 0h | SOC8 转换启动溢出标志。表示已有 SOC8 事件待处理时，硬件又产生 SOC8 事件。<br>0：无 SOC8 事件溢出。<br>1：发生 SOC8 事件溢出。<br>溢出不会阻止处理 SOC8 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 位。<br>复位类型：SYSRSn。 |
| 7 | SOC7 | R | 0h | SOC7 转换启动溢出标志。表示已有 SOC7 事件待处理时，硬件又产生 SOC7 事件。<br>0：无 SOC7 事件溢出。<br>1：发生 SOC7 事件溢出。<br>溢出不会阻止处理 SOC7 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 6 | SOC6 | R | 0h | SOC6 转换启动溢出标志。表示已有 SOC6 事件待处理时，硬件又产生 SOC6 事件。<br>0：无 SOC6 事件溢出。<br>1：发生 SOC6 事件溢出。<br>溢出不会阻止处理 SOC6 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 5 | SOC5 | R | 0h | SOC5 转换启动溢出标志。表示已有 SOC5 事件待处理时，硬件又产生 SOC5 事件。<br>0：无 SOC5 事件溢出。<br>1：发生 SOC5 事件溢出。<br>溢出不会阻止处理 SOC5 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 4 | SOC4 | R | 0h | SOC4 转换启动溢出标志。表示已有 SOC4 事件待处理时，硬件又产生 SOC4 事件。<br>0：无 SOC4 事件溢出。<br>1：发生 SOC4 事件溢出。<br>溢出不会阻止处理 SOC4 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 3 | SOC3 | R | 0h | SOC3 转换启动溢出标志。表示已有 SOC3 事件待处理时，硬件又产生 SOC3 事件。<br>0：无 SOC3 事件溢出。<br>1：发生 SOC3 事件溢出。<br>溢出不会阻止处理 SOC3 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 2 | SOC2 | R | 0h | SOC2 转换启动溢出标志。表示已有 SOC2 事件待处理时，硬件又产生 SOC2 事件。<br>0：无 SOC2 事件溢出。<br>1：发生 SOC2 事件溢出。<br>溢出不会阻止处理 SOC2 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 1 | SOC1 | R | 0h | SOC1 转换启动溢出标志。表示已有 SOC1 事件待处理时，硬件又产生 SOC1 事件。<br>0：无 SOC1 事件溢出。<br>1：发生 SOC1 事件溢出。<br>溢出不会阻止处理 SOC1 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |
| 0 | SOC0 | R | 0h | SOC0 转换启动溢出标志。表示已有 SOC0 事件待处理时，硬件又产生 SOC0 事件。<br>0：无 SOC0 事件溢出。<br>1：发生 SOC0 事件溢出。<br>溢出不会阻止处理 SOC0 事件，仅表示遗漏了一次硬件触发。写入 ADCSOCFRC1 不影响此位。<br>复位类型：SYSRSn。 |

### 13.4.3.16 ADCSOCOVFCLR1 寄存器

偏移地址：1Eh　复位值：0000h

ADC SOC 溢出清除 1 寄存器

表 13-86 ADCSOCOVFCLR1 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SOC15 | SOC14 | SOC13 | SOC12 | SOC11 | SOC10 | SOC9 | SOC8 |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SOC7 | SOC6 | SOC5 | SOC4 | SOC3 | SOC2 | SOC1 | SOC0 |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 13-87 ADCSOCOVFCLR1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | SOC15 | R-0/W1S | 0h | SOC15 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC15 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC15 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 14 | SOC14 | R-0/W1S | 0h | SOC14 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC14 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC14 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 13 | SOC13 | R-0/W1S | 0h | SOC13 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC13 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC13 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 12 | SOC12 | R-0/W1S | 0h | SOC12 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC12 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC12 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 11 | SOC11 | R-0/W1S | 0h | SOC11 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC11 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC11 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 10 | SOC10 | R-0/W1S | 0h | SOC10 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC10 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC10 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 9 | SOC9 | R-0/W1S | 0h | SOC9 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC9 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC9 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 8 | SOC8 | R-0/W1S | 0h | SOC8 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC8 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC8 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 7 | SOC7 | R-0/W1S | 0h | SOC7 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC7 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC7 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 6 | SOC6 | R-0/W1S | 0h | SOC6 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC6 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：清除 SOC6 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 5 | SOC5 | R-0/W1S | 0h | SOC5 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC5 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC5 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 4 | SOC4 | R-0/W1S | 0h | SOC4 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC4 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC4 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 3 | SOC3 | R-0/W1S | 0h | SOC3 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC3 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC3 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 2 | SOC2 | R-0/W1S | 0h | SOC2 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC2 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC2 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 1 | SOC1 | R-0/W1S | 0h | SOC1 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC1 溢出标志。忽略写入 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC1 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |
| 0 | SOC0 | R-0/W1S | 0h | SOC0 转换启动溢出清除位。写入 1 清除 ADCSOCOVF1 中 SOC0 溢出标志。忽略写入 0，读取始终返回 0。<br>0：无操作。<br>1：清除 SOC0 溢出标志。<br>若软件设置此位与硬件设置 ADCSOCOVF1 溢出位发生在同一时钟周期，硬件优先，ADCSOCOVF1 位将置位。<br>复位类型：SYSRSn。 |

### 13.4.3.17 ADCSOC0CTL 寄存器

偏移地址：20h　复位值：00000000h

ADC SOC0 控制寄存器

表 13-88 ADCSOC0CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-89 ADCSOC0CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC0 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC0 字段一起，配置将 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | ADCSOCFLG1 中 SOC0 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n<br>04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn。 |
| 19-15 | CHSEL | R/W | 0h | SOC0 通道选择。选择 ADC 收到 SOC0 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |
| 8-0 | ACQPS | R/W | 0h | SOC0 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

### 13.4.3.18 ADCSOC1CTL 寄存器

偏移地址：24h　复位值：00000000h

ADC SOC1 控制寄存器

表 13-90 ADCSOC1CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-91 ADCSOC1CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC1 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC1 字段一起，配置将 ADCSOCFLG1 中 SOC1 标志置位的触发源，以便获得优先权后启动转换。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n<br>04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn。 |
| 19-15 | CHSEL | R/W | 0h | SOC1 通道选择。选择 ADC 收到 SOC1 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | Fh：ADCIN15<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |
| 8-0 | ACQPS | R/W | 0h | SOC1 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

## 13.4.3.19 ADCSOC2CTL 寄存器

偏移地址：28h　复位值：00000000h

ADC SOC2 控制寄存器

表 13-92 ADCSOC2CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-93 ADCSOC2CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC2 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC2 字段一起，配置将 ADCSOCFLG1 中 SOC2 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n<br>04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC2 通道选择。选择 ADC 收到 SOC2 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 14-9 | RESERVED | R | 0h | 保留。 |
| 8-0 | ACQPS | R/W | 0h | SOC2 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

## 13.4.3.20 ADCSOC3CTL 寄存器

偏移地址：2Ch　复位值：00000000h

ADC SOC3 控制寄存器

表 13-94 ADCSOC3CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-95 ADCSOC3CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC3 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC3 字段一起，配置将 ADCSOCFLG1 中 SOC3 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 03h：ADCTRIG3 — CPU1 定时器 2, TINT2n<br>04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC3 通道选择。选择 ADC 收到 SOC3 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 8-0 | ACQPS | R/W | 0h | SOC3 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

## 13.4.3.21 ADCSOC4CTL 寄存器

偏移地址：30h　复位值：00000000h

ADC SOC4 控制寄存器

表 13-96 ADCSOC4CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-97 ADCSOC4CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC4 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC4 字段一起，配置将 ADCSOCFLG1 中 SOC4 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC4 通道选择。选择 ADC 收到 SOC4 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 8-0 | ACQPS | R/W | 0h | SOC4 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

## 13.4.3.22 ADCSOC5CTL 寄存器

偏移地址：34h　复位值：00000000h

ADC SOC5 控制寄存器

表 13-98 ADCSOC5CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-99 ADCSOC5CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC5 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC5 字段一起，配置将 ADCSOCFLG1 中 SOC5 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC5 通道选择。选择 ADC 收到 SOC5 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 8-0 | ACQPS | R/W | 0h | SOC5 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

### 13.4.3.23 ADCSOC6CTL 寄存器

偏移地址：38h　复位值：00000000h

ADC SOC6 控制寄存器

表 13-100 ADCSOC6CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-101 ADCSOC6CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC6 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC6 字段一起，配置将 ADCSOCFLG1 中 SOC6 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC6 通道选择。选择 ADC 收到 SOC6 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 8-0 | ACQPS | R/W | 0h | SOC6 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

### 13.4.3.24 ADCSOC7CTL 寄存器

偏移地址：3Ch　复位值：00000000h

ADC SOC7 控制寄存器

表 13-102 ADCSOC7CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-103 ADCSOC7CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC7 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC7 字段一起，配置将 ADCSOCFLG1 中 SOC7 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC7 通道选择。选择 ADC 收到 SOC7 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 8-0 | ACQPS | R/W | 0h | SOC7 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

### 13.4.3.25 ADCSOC8CTL 寄存器

偏移地址：40h　复位值：00000000h

ADC SOC8 控制寄存器

表 13-104 ADCSOC8CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-105 ADCSOC8CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC8 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC8 字段一起，配置将 ADCSOCFLG1 中 SOC8 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC8 通道选择。选择 ADC 收到 SOC8 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 8-0 | ACQPS | R/W | 0h | SOC8 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

### 13.4.3.26 ADCSOC9CTL 寄存器

偏移地址：44h　复位值：00000000h

ADC SOC9 控制寄存器

表 13-106 ADCSOC9CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-107 ADCSOC9CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC9 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC9 字段一起，配置将 ADCSOCFLG1 中 SOC9 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC9 通道选择。选择 ADC 收到 SOC9 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 8-0 | ACQPS | R/W | 0h | SOC9 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

## 13.4.3.27 ADCSOC10CTL 寄存器

偏移地址：48h　复位值：00000000h

ADC SOC10 控制寄存器

表 13-108 ADCSOC10CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-109 ADCSOC10CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC10 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC10 字段一起，配置将 ADCSOCFLG1 中 SOC10 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC10 通道选择。选择 ADC 收到 SOC10 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 8-0 | ACQPS | R/W | 0h | SOC10 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

## 13.4.3.28 ADCSOC11CTL 寄存器

偏移地址：4Ch　复位值：00000000h

ADC SOC11 控制寄存器

表 13-110 ADCSOC11CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-111 ADCSOC11CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC11 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC11 字段一起，配置将 ADCSOCFLG1 中 SOC11 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC11 通道选择。选择 ADC 收到 SOC11 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 8-0 | ACQPS | R/W | 0h | SOC11 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

## 13.4.3.29 ADCSOC12CTL 寄存器

偏移地址：50h　复位值：00000000h

ADC SOC12 控制寄存器

表 13-112 ADCSOC12CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-113 ADCSOC12CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC12 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC12 字段一起，配置将 ADCSOCFLG1 中 SOC12 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC12 通道选择。选择 ADC 收到 SOC12 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |
| 8-0 | ACQPS | R/W | 0h | SOC12 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

### 13.4.3.30 ADCSOC13CTL 寄存器

偏移地址：54h　复位值：00000000h

ADC SOC13 控制寄存器

表 13-114 ADCSOC13CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-115 ADCSOC13CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC13 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC13 字段一起，配置将 ADCSOCFLG1 中 SOC13 标志置位的触发源，以便获得优先权后启动转换。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n<br>04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC13 通道选择。选择 ADC 收到 SOC13 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |
| 8-0 | ACQPS | R/W | 0h | SOC13 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

## 13.4.3.31 ADCSOC14CTL 寄存器

偏移地址：58h　复位值：00000000h

ADC SOC14 控制寄存器

表 13-116 ADCSOC14CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-117 ADCSOC14CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC14 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC14 字段一起，配置将 ADCSOCFLG1 中 SOC14 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n<br>03h：ADCTRIG3 — CPU1 定时器 2, TINT2n<br>04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC14 通道选择。选择 ADC 收到 SOC14 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 14-9 | RESERVED | R | 0h | 保留。 |
| 8-0 | ACQPS | R/W | 0h | SOC14 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

## 13.4.3.32 ADCSOC15CTL 寄存器

偏移地址：5Ch　复位值：00000000h

ADC SOC15 控制寄存器

表 13-118 ADCSOC15CTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRIGSEL | | |
| R-0h | | | | | R/W-0h | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRIGSEL | | | | CHSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CHSEL | 保留 | | | | | | ACQPS |
| R/W-0h | R-0h | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACQPS | | | | | | | |
| R/W-0h | | | | | | | |

表 13-119 ADCSOC15CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-27 | RESERVED | R | 0h | 保留。 |
| 26-20 | TRIGSEL | R/W | 0h | SOC15 触发源选择。与 ADCINTSOCSEL1 寄存器中的 SOC15 字段一起，配置将 ADCSOCFLG1 中 SOC15 标志置位的触发源，以便获得优先权后启动转换。<br>00h：ADCTRIG0 — 仅软件<br>01h：ADCTRIG1 — CPU1 定时器 0, TINT0n<br>02h：ADCTRIG2 — CPU1 定时器 1, TINT1n |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 03h：ADCTRIG3 — CPU1 定时器 2, TINT2n<br>04h：ADCTRIG4 — GPIO, ADCEXTSOC<br>05h：ADCTRIG5 — ePWM1, ADCSOCA<br>06h：ADCTRIG6 — ePWM1, ADCSOCB<br>07h：ADCTRIG7 — ePWM2, ADCSOCA<br>08h：ADCTRIG8 — ePWM2, ADCSOCB<br>09h：ADCTRIG9 — ePWM3, ADCSOCA<br>0Ah：ADCTRIG10 — ePWM3, ADCSOCB<br>0Bh：ADCTRIG11 — ePWM4, ADCSOCA<br>0Ch：ADCTRIG12 — ePWM4, ADCSOCB<br>0Dh：ADCTRIG13 — ePWM5, ADCSOCA<br>0Eh：ADCTRIG14 — ePWM5, ADCSOCB<br>0Fh：ADCTRIG15 — ePWM6, ADCSOCA<br>10h：ADCTRIG16 — ePWM6, ADCSOCB<br>11h：ADCTRIG17 — ePWM7, ADCSOCA<br>12h：ADCTRIG18 — ePWM7, ADCSOCB<br>13h - 7Fh：保留。<br>复位类型：SYSRSn |
| 19-15 | CHSEL | R/W | 0h | SOC15 通道选择。选择 ADC 收到 SOC15 时要转换的通道。<br>0h：ADCIN0<br>1h：ADCIN1<br>2h：ADCIN2<br>3h：ADCIN3<br>4h：ADCIN4<br>5h：ADCIN5<br>6h：ADCIN6<br>7h：ADCIN7<br>8h：ADCIN8<br>9h：ADCIN9<br>Ah：ADCIN10<br>Bh：ADCIN11<br>Ch：ADCIN12<br>Dh：ADCIN13<br>Eh：ADCIN14<br>Fh：ADCIN15<br>10h：ADCIN16<br>11h：ADCIN17<br>12h：ADCIN18<br>13h：ADCIN19<br>14h：ADCIN20<br>15h-1Fh:保留<br>复位类型：SYSRSn。 |
| 14-9 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 8-0 | ACQPS | R/W | 0h | SOC15 采集预分频，控制此 SOC 的采样保持窗口。为保证 ADC 正常工作，配置的采集时间必须至少为一个 ADCCLK 周期。器件数据手册还规定了最小采样保持窗口持续时间。<br>000h：采样窗口宽 1 个系统时钟周期。<br>001h：宽 2 个系统时钟周期。<br>002h：宽 3 个系统时钟周期。<br>……<br>1FFh：宽 512 个系统时钟周期。<br>复位类型：SYSRSn。 |

### 13.4.3.33 ADCEVTSTAT 寄存器

偏移地址：60h　复位值：0000h

ADC 事件状态寄存器

表 13-120 ADCEVTSTAT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | PPB4ZERO | PPB4TRIPLO | PPB4TRIPHI | 保留 | PPB3ZERO | PPB3TRIPLO | PPB3TRIPHI |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | PPB2ZERO | PPB2TRIPLO | PPB2TRIPHI | 保留 | PPB1ZERO | PPB1TRIPLO | PPB1TRIPHI |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 13-121 ADCEVTSTAT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留。 |
| 14 | PPB4ZERO | R | 0h | 后处理模块 4 的过零标志。置位表示 ADCPPB4RESULT 寄存器的符号已改变。此位由 EOC 信号门控。<br>注意：即使 ADCEVTINTSEL 中对应使能位未设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 13 | PPB4TRIPLO | R | 0h | 后处理模块 4 的低限跳闸标志。置位表示发生了数字比较低限跳闸事件。<br>注意：即使 ADCEVTINTSEL 中对应使能位未 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 12 | PPB4TRIPHI | R | 0h | 后处理模块 4 的高限跳闸标志。置位表示发生了数字比较高限跳闸事件。<br>注意：即使 ADCEVTINTSEL 中对应使能位未设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 11 | RESERVED | R | 0h | 保留。 |
| 10 | PPB3ZERO | R | 0h | 后处理模块 3 的过零标志。置位表示 ADCPPB3RESULT 寄存器的符号已改变。此位由 EOC 信号门控。<br>注意：即使 ADCEVTINTSEL 中对应使能位未设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 9 | PPB3TRIPLO | R | 0h | 后处理模块 3 的低限跳闸标志。置位表示发生了数字比较低限跳闸事件。<br>注意：即使 ADCEVTINTSEL 中对应使能位未设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 8 | PPB3TRIPHI | R | 0h | 后处理模块 3 的高限跳闸标志。置位表示发生了数字比较高限跳闸事件。<br>注意：即使 ADCEVTINTSEL 中对应使能位未设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 期时，硬件优先。<br>复位类型：SYSRSn。 |
| 7 | RESERVED | R | 0h | 保留。 |
| 6 | PPB2ZERO | R | 0h | 后处理模块 2 的过零标志。置位表示 ADCPPB2RESULT 寄存器的符号已改变。此位由 EOC 信号门控。<br>注意：即使 ADCEVTINTSEL 中对应使能位未设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 5 | PPB2TRIPLO | R | 0h | 后处理模块 2 的低限跳闸标志。置位表示发生了数字比较低限跳闸事件。<br>注意：即使 ADCEVTINTSEL 中对应使能位未设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 4 | PPB2TRIPHI | R | 0h | 后处理模块 2 的高限跳闸标志。置位表示发生了数字比较高限跳闸事件。<br>注意：即使 ADCEVTINTSEL 中对应使能位未设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 3 | RESERVED | R | 0h | 保留。 |
| 2 | PPB1ZERO | R | 0h | 后处理模块 1 的过零标志。置位表示 ADCPPB1RESULT 寄存器的符号已改变。此位由 EOC 信号门控。<br>注意：即使 ADCEVTINTSEL 中对应使能位未设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | PPB1TRIPLO | R | 0h | 后处理模块 1 的低限跳闸标志。置位表示发生了数字比较低限跳闸事件。<br>注意：即使 ADCEVTINTSEL 中对应使能位未设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 0 | PPB1TRIPHI | R | 0h | 后处理模块 1 的高限跳闸标志。置位表示发生了数字比较高限跳闸事件。<br>注意：即使 ADCEVTINTSEL 中对应使能位未设置，这些位也会置位。因此，中断服务程序可能需要同时检查 ADCEVTSTAT 和 ADCEVTINTSEL，以确定中断源。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |

### 13.4.3.34 ADCEVTCLR 寄存器

偏移地址：64h　复位值：0000h

ADC 事件清除寄存器

表 13-122 ADCEVTCLR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | PPB4ZERO | PPB4TRIPLO | PPB4TRIPHI | 保留 | PPB3ZERO | PPB3TRIPLO | PPB3TRIPHI |
| R-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | PPB2ZERO | PPB2TRIPLO | PPB2TRIPHI | 保留 | PPB1ZERO | PPB1TRIPLO | PPB1TRIPHI |
| R-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 13-123 ADCEVTCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留。 |
| 14 | PPB4ZERO | R-0/W1S | 0h | 后处理模块 4 的过零清除。清除 ADCEVTSTAT 中对应过零标志。<br>注意：软件清除位与硬件置位标志发生在同一周 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 期时，硬件优先。<br>复位类型：SYSRSn。 |
| 13 | PPB4TRIPLO | R-0/W1S | 0h | 后处理模块 4 的低限跳闸清除。清除 ADCEVTSTAT 中对应低限跳闸标志。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 12 | PPB4TRIPHI | R-0/W1S | 0h | 后处理模块 4 的高限跳闸清除。清除 ADCEVTSTAT 中对应高限跳闸标志。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 11 | RESERVED | R | 0h | 保留。 |
| 10 | PPB3ZERO | R-0/W1S | 0h | 后处理模块 3 的过零清除。清除 ADCEVTSTAT 中对应过零标志。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 9 | PPB3TRIPLO | R-0/W1S | 0h | 后处理模块 3 的低限跳闸清除。清除 ADCEVTSTAT 中对应低限跳闸标志。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 8 | PPB3TRIPHI | R-0/W1S | 0h | 后处理模块 3 的高限跳闸清除。清除 ADCEVTSTAT 中对应高限跳闸标志。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 7 | RESERVED | R | 0h | 保留。 |
| 6 | PPB2ZERO | R-0/W1S | 0h | 后处理模块 2 的过零清除。清除 ADCEVTSTAT 中对应过零标志。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 5 | PPB2TRIPLO | R-0/W1S | 0h | 后处理模块 2 的低限跳闸清除。清除 ADCEVTSTAT 中对应低限跳闸标志。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 4 | PPB2TRIPHI | R-0/W1S | 0h | 后处理模块 2 的高限跳闸清除。清除 ADCEVTSTAT 中对应高限跳闸标志。<br>注意：软件清除位与硬件置位标志发生在同一周 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 期时，硬件优先。<br>复位类型：SYSRSn。 |
| 3 | RESERVED | R | 0h | 保留。 |
| 2 | PPB1ZERO | R-0/W1S | 0h | 后处理模块 1 的过零清除。清除 ADCEVTSTAT 中对应过零标志。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 1 | PPB1TRIPLO | R-0/W1S | 0h | 后处理模块 1 的低限跳闸清除。清除 ADCEVTSTAT 中对应低限跳闸标志。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |
| 0 | PPB1TRIPHI | R-0/W1S | 0h | 后处理模块 1 的高限跳闸清除。清除 ADCEVTSTAT 中对应高限跳闸标志。<br>注意：软件清除位与硬件置位标志发生在同一周期时，硬件优先。<br>复位类型：SYSRSn。 |

### 13.4.3.35 ADCEVTSEL 寄存器

偏移地址：68h　复位值：0000h

ADC 事件选择寄存器

表 13-124 ADCEVTSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | PPB4ZERO | PPB4TRIPLO | PPB4TRIPHI | 保留 | PPB3ZERO | PPB3TRIPLO | PPB3TRIPHI |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | PPB2ZERO | PPB2TRIPLO | PPB2TRIPHI | 保留 | PPB1ZERO | PPB1TRIPLO | PPB1TRIPHI |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h |

表 13-125 ADCEVTSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留。 |
| 14 | PPB4ZERO | R/W | 0h | 后处理模块 4 的过零事件使能。置位后，对应过零标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 PWM 模 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 块 产生后续事件。<br>复位类型：SYSRSn。 |
| 13 | PPB4TRIPLO | R/W | 0h | 后处理模块 4 的低限跳闸事件使能。置位后，对应低限跳闸标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 PWM 模块 产生后续事件。<br>复位类型：SYSRSn。 |
| 12 | PPB4TRIPHI | R/W | 0h | 后处理模块 4 的高限跳闸事件使能。置位后，对应高限跳闸标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 PWM 模块 产生后续事件。<br>复位类型：SYSRSn。 |
| 11 | RESERVED | R | 0h | 保留。 |
| 10 | PPB3ZERO | R/W | 0h | 后处理模块 3 的过零事件使能。置位后，对应过零标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 PWM 模块 产生后续事件。<br>复位类型：SYSRSn。 |
| 9 | PPB3TRIPLO | R/W | 0h | 后处理模块 3 的低限跳闸事件使能。置位后，对应低限跳闸标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 PWM 模块 产生后续事件。<br>复位类型：SYSRSn。 |
| 8 | PPB3TRIPHI | R/W | 0h | 后处理模块 3 的高限跳闸事件使能。置位后，对应高限跳闸标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 PWM 模块 产生后续事件。<br>复位类型：SYSRSn。 |
| 7 | RESERVED | R | 0h | 保留。 |
| 6 | PPB2ZERO | R/W | 0h | 后处理模块 2 的过零事件使能。置位后，对应过零标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 PWM 模块 产生后续事件。<br>复位类型：SYSRSn。 |
| 5 | PPB2TRIPLO | R/W | 0h | 后处理模块 2 的低限跳闸事件使能。置位后，对应低限跳闸标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 PWM 模块 产生后续事件。<br>复位类型：SYSRSn。 |
| 4 | PPB2TRIPHI | R/W | 0h | 后处理模块 2 的高限跳闸事件使能。置位后，对应高限跳闸标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | PWM 模块 产生后续事件。<br>复位类型：SYSRSn。 |
| 3 | RESERVED | R | 0h | 保留。 |
| 2 | PPB1ZERO | R/W | 0h | 后处理模块 1 的过零事件使能。置位后，对应过零标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 PWM 模块 产生后续事件。<br>复位类型：SYSRSn。 |
| 1 | PPB1TRIPLO | R/W | 0h | 后处理模块 1 的低限跳闸事件使能。置位后，对应低限跳闸标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 PWM 模块 产生后续事件。<br>复位类型：SYSRSn。 |
| 0 | PPB1TRIPHI | R/W | 0h | 后处理模块 1 的高限跳闸事件使能。置位后，对应高限跳闸标志的上升沿可激活送往 PWM 模块 的事件信号。必须先清除此标志，才能向 PWM 模块 产生后续事件。<br>复位类型：SYSRSn。 |

## 13.4.3.36 ADCEVTINTSEL 寄存器

偏移地址：6Ch　复位值：0000h

ADC 事件中断选择寄存器

表 13-126 ADCEVTINTSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | PPB4ZERO | PPB4TRIPLO | PPB4TRIPHI | 保留 | PPB3ZERO | PPB3TRIPLO | PPB3TRIPHI |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | PPB2ZERO | PPB2TRIPLO | PPB2TRIPHI | 保留 | PPB1ZERO | PPB1TRIPLO | PPB1TRIPHI |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h |

表 13-127 ADCEVTINTSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留。 |
| 14 | PPB4ZERO | R/W | 0h | 后处理模块 4 的过零事件中断使能。置位后，对应过零标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才能向 CLIC 产 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 生后续中断。<br>复位类型：SYSRSn。 |
| 13 | PPB4TRIPLO | R/W | 0h | 后处理模块 4 的低限跳闸事件中断使能。置位后，对应低限跳闸标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才能向 CLIC 产生后续中断。<br>复位类型：SYSRSn。 |
| 12 | PPB4TRIPHI | R/W | 0h | 后处理模块 4 的高限跳闸事件中断使能。置位后，对应高限跳闸标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才能向 CLIC 产生后续中断。<br>复位类型：SYSRSn。 |
| 11 | RESERVED | R | 0h | 保留。 |
| 10 | PPB3ZERO | R/W | 0h | 后处理模块 3 的过零事件中断使能。置位后，对应过零标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才能向 CLIC 产生后续中断。<br>复位类型：SYSRSn。 |
| 9 | PPB3TRIPLO | R/W | 0h | 后处理模块 3 的低限跳闸事件中断使能。置位后，对应低限跳闸标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才能向 CLIC 产生后续中断。<br>复位类型：SYSRSn。 |
| 8 | PPB3TRIPHI | R/W | 0h | 后处理模块 3 的高限跳闸事件中断使能。置位后，对应高限跳闸标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才能向 CLIC 产生后续中断。<br>复位类型：SYSRSn。 |
| 7 | RESERVED | R | 0h | 保留。 |
| 6 | PPB2ZERO | R/W | 0h | 后处理模块 2 的过零事件中断使能。置位后，对应过零标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才能向 CLIC 产生后续中断。<br>复位类型：SYSRSn。 |
| 5 | PPB2TRIPLO | R/W | 0h | 后处理模块 2 的低限跳闸事件中断使能。置位后，对应低限跳闸标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才能向 CLIC 产生后续中断。<br>复位类型：SYSRSn。 |
| 4 | PPB2TRIPHI | R/W | 0h | 后处理模块 2 的高限跳闸事件中断使能。置位后，对应高限跳闸标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 能向 CLIC 产生后续中断。<br>复位类型：SYSRSn。 |
| 3 | RESERVED | R | 0h | 保留。 |
| 2 | PPB1ZERO | R/W | 0h | 后处理模块 1 的过零事件中断使能。置位后，对应过零标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才能向 CLIC 产生后续中断。<br>复位类型：SYSRSn。 |
| 1 | PPB1TRIPLO | R/W | 0h | 后处理模块 1 的低限跳闸事件中断使能。置位后，对应低限跳闸标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才能向 CLIC 产生后续中断。<br>复位类型：SYSRSn。 |
| 0 | PPB1TRIPHI | R/W | 0h | 后处理模块 1 的高限跳闸事件中断使能。置位后，对应高限跳闸标志的上升沿可激活送往 CLIC 的事件中断信号。必须先清除此标志，才能向 CLIC 产生后续中断。<br>复位类型：SYSRSn。 |

### 13.4.3.37 ADCOSDETECT 寄存器

偏移地址：70h　复位值：0000h

ADC 开路 / 短路检测寄存器

表 13-128 ADCOSDETECT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | DETECTCFG | | |
| R-0h | | | | | R/W-0h | | |

表 13-129 ADCOSDETECT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-3 | RESERVED | R | 0h | 保留。 |
| 2-0 | DETECTCFG | R/W | 0h | ADC 开路 / 短路检测配置。定义开路 / 短路检测电路状态。<br>0h：禁用检测电路。<br>1h：使能检测电路，零量程。<br>2h：使能检测电路，满量程。<br>3h：使能检测电路，标称 5/12 量程。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 4h：使能检测电路，标称 7/12 量程。<br>5h：使能检测电路，通过标称 5K 下拉至 VSSA。<br>6h：使能检测电路，通过标称 5K 上拉至 VDDA。<br>7h：使能检测电路，通过标称 7K 下拉至 VSSA。<br>复位类型：SYSRSn。 |

### 13.4.3.38 ADCCOUNTER 寄存器

偏移地址：72h　复位值：0000h

ADC 计数器寄存器

表 13-130 ADCCOUNTER 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | FREECOUNT | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FREECOUNT | | | | | | | |
| R-0h | | | | | | | |

表 13-131 ADCCOUNTER 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11-0 | FREECOUNT | R | 0h | ADC 自由运行计数器值。反映自由运行 ADC 计数器的状态。<br>复位类型：SYSRSn。 |

### 13.4.3.39 ADCREV 寄存器

偏移地址：74h　复位值：0005h

ADC 修订版本寄存器

表 13-132 ADCREV 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| REV | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TYPE | | | | | | | |
| R-5h | | | | | | | |

表 13-133 ADCREV 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | REV | R | 0h | ADC 修订版本，用于记录不同修订版本的差异。首个版本标记为 00h。<br>复位类型：SYSRSn。 |
| 7-0 | TYPE | R | 5h | ADC 类型。此 ADC 始终为 5。<br>复位类型：SYSRSn。 |

## 13.4.3.40 ADCOFFTRIM 寄存器

偏移地址：76h 复位值：0h

ADC 偏移修调寄存器

表 13-134 ADCOFFTRIM 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | BYPASS | ADC校准 |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADC校准 | | | | | | | |
| R/W-0h | | | | | | | |

表 13-135 ADCOFFTRIM 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-10 | RESERVED | R | 0h | 保留。 |
| 9 | BYPASS | R/W | 0h | 1:表示旁路ADC内部数据转换<br>0:表示不旁路ADC内部数据转换 |
| 8-0 | ADC校准 | R/W | 000000000h | ADC校准 |

## 13.4.3.41 ADCPPB1CONFIG 寄存器

偏移地址：80h 复位值：0000h

ADC PPB1 配置寄存器

表 13-136 ADCPPB1CONFIG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | CBCEN | TWOSCOMPEN | CONFIG | | | |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | | | |

表 13-137 ADCPPB1CONFIG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-6 | RESERVED | R | 0h | 保留。 |
| 5 | CBCEN | R/W | 0h | ADC 后处理模块逐周期使能。置位后，若事件条件已不再存在，转换后硬件处理电路在进行转换时自动清除 ADCEVTSTAT。<br>复位类型：SYSRSn。 |
| 4 | TWOSCOMPEN | R/W | 0h | ADC 后处理模块 1 二进制补码使能。置位后，转换后硬件处理电路对偏移 / 参考减法单元的输出取二进制补码，再将结果存入 ADCPPB1RESULT。<br>0：ADCPPB1RESULT = ADCRESULTx - ADCPPB1OFFREF<br>1：ADCPPB1RESULT = ADCPPB1OFFREF - ADCRESULTx<br>复位类型：SYSRSn。 |
| 3-0 | CONFIG | R/W | 0h | ADC 后处理模块 1 配置。定义与该模块关联的 SOC/EOC/RESULT。<br>0000：SOC0/EOC0/RESULT0 与后处理模块 1 关联。<br>0001：SOC1/EOC1/RESULT1 与后处理模块 1 关联。<br>0010：SOC2/EOC2/RESULT2 与后处理模块 1 关联。<br>0011：SOC3/EOC3/RESULT3 与后处理模块 1 关联。<br>0100：SOC4/EOC4/RESULT4 与后处理模块 1 关联。<br>0101：SOC5/EOC5/RESULT5 与后处理模块 1 关联。<br>0110：SOC6/EOC6/RESULT6 与后处理模块 1 关联。<br>0111：SOC7/EOC7/RESULT7 与后处理模块 1 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 关联。<br>1000：SOC8/EOC8/RESULT8 与后处理模块 1 关联。<br>1001：SOC9/EOC9/RESULT9 与后处理模块 1 关联。<br>1010：SOC10/EOC10/RESULT10 与后处理模块 1 关联。<br>1011：SOC11/EOC11/RESULT11 与后处理模块 1 关联。<br>1100：SOC12/EOC12/RESULT12 与后处理模块 1 关联。<br>1101：SOC13/EOC13/RESULT13 与后处理模块 1 关联。<br>1110：SOC14/EOC14/RESULT14 与后处理模块 1 关联。<br>1111：SOC15/EOC15/RESULT15 与后处理模块 1 关联。<br>复位类型：SYSRSn。 |

## 13.4.3.42 ADCPPB1STAMP 寄存器

偏移地址：82h　复位值：0000h

ADC PPB1 采样延迟时间戳寄存器

表 13-138 ADCPPB1STAMP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | DLYSTAMP | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DLYSTAMP | | | | | | | |
| R-0h | | | | | | | |

表 13-139 ADCPPB1STAMP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11-0 | DLYSTAMP | R | 0h | ADC 后处理模块 1 延迟时间戳。SOC 开始采样时，从 ADCCOUNTER.FREECOUNT 的值减去 REQSTAMP，并将结果加载至此字段，得到 SOC 触发到实际开始采样之间的系统时钟周期数。<br>复位类型：SYSRSn。 |

## 13.4.3.43 ADCPPB1OFFCAL 寄存器

偏移地址：84h　复位值：0000h

ADC PPB1 偏移校准寄存器

表 13-140 ADCPPB1OFFCAL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | OFFCAL | |
| R-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OFFCAL | | | | | | | |
| R/W-0h | | | | | | | |

表 13-141 ADCPPB1OFFCAL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-10 | RESERVED | R | 0h | 保留。 |
| 9-0 | OFFCAL | R/W | 0h | ADC 后处理模块 1 偏移校正。以数字方式消除 ADCIN 电路固有的系统级偏移。将 ADC 输出减去此 10 位有符号值后，存入 ADCRESULT。<br>000h：不变，ADC 输出直接存入 ADCRESULT。<br>001h：将 ADC 输出 - 1 存入 ADCRESULT。<br>002h：将 ADC 输出 - 2 存入 ADCRESULT。<br>……<br>200h：将 ADC 输出 + 512 存入 ADCRESULT。<br>……<br>3FFh：将 ADC 输出 + 1 存入 ADCRESULT。<br>注意：16 位模式下，减法结果在存入 ADCRESULT 前饱和限制于 0000h 和 FFFFh；12 位模式下，饱和限制于 0000h 和 0FFFh。<br>注意：多个 PPB 指向同一 SOC 时，仅应用编号最高的 PPB 的 OFFCAL。<br>复位类型：SYSRSn。 |

## 13.4.3.44 ADCPPB1OFFREF 寄存器

偏移地址：86h　复位值：0000h

ADC PPB1 偏移参考寄存器

表 13-142 ADCPPB1OFFREF 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| OFFREF | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OFFREF | | | | | | | |
| R/W-0h | | | | | | | |

表 13-143 ADCPPB1OFFREF 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | OFFREF | R/W | 0h | ADC 后处理模块 1 偏移校正。通过减去参考值计算反馈误差，或将单极性信号转换为双极性信号。从 ADCRESULT 减去此 16 位无符号值，再经过可选的二进制补码处理，存入 ADCPPB1RESULT。此减法不作饱和处理。<br>0000h：不变，直接传递 ADCRESULT。<br>0001h：传递 ADCRESULT - 1。<br>0002h：传递 ADCRESULT - 2。<br>……<br>8000h：传递 ADCRESULT - 32,768。<br>……<br>FFFFh：传递 ADCRESULT - 65,535。<br>注意：12 位模式下，此寄存器仍为 16 位。用户必须确保仅向其中写入 12 位值。<br>复位类型：SYSRSn。 |

## 13.4.3.45 ADCPPB1TRIPHI 寄存器

偏移地址：88h　复位值：00000000h

ADC PPB1 高限跳闸寄存器

表 13-144 ADCPPB1TRIPHI 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | HSIGN |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LIMITHI | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LIMITHI | | | | | | | |
| R/W-0h | | | | | | | |

表 13-145 ADCPPB1TRIPHI 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | 保留。 |
| 16 | HSIGN | R/W | 0h | 高限符号位。16 位 ADC 模式下，这是 LIMITHI 字段的符号位（第 17 位）。<br>复位类型：SYSRSn。 |
| 15-0 | LIMITHI | R/W | 0h | ADC 后处理模块 1 高限跳闸阈值。设置数字比较器高限阈值。16 位模式下，全部 17 位与 ADCPPB1RESULT 的 PPBRESULT 字段的 17 位比较；12 位模式下，位 12:0 与 ADCPPB1RESULT 的 PPBRESULT 字段的位 12:0 比较。<br>复位类型：SYSRSn。 |

## 13.4.3.46 ADCPPB1TRIPLO 寄存器

偏移地址：8Ch　复位值：00000000h

ADC PPB1 低限跳闸 / 触发时间戳寄存器

表 13-146 ADCPPB1TRIPLO 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQSTAMP | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQSTAMP | | | | 保留 | | | LSIGN |
| R-0h | | | | R-0h | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LIMITLO | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LIMITLO | | | | | | | |
| R/W-0h | | | | | | | |

表 13-147 ADCPPB1TRIPLO 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | REQSTAMP | R | 0h | ADC 后处理模块 1 请求时间戳。触发将 ADCSOCFLG1 中关联的 SOC 标志置位时，将 ADCCOUNTER.FREECOUNT 的值加载至此字段。<br>复位类型：SYSRSn。 |
| 19-17 | RESERVED | R | 0h | 保留。 |
| 16 | LSIGN | R/W | 0h | 低限符号位。16 位 ADC 模式下，这是 LIMITLO 字段的符号位（第 17 位）。<br>复位类型：SYSRSn。 |
| 15-0 | LIMITLO | R/W | 0h | ADC 后处理模块 1 低限跳闸阈值。设置数字比较器低限阈值。16 位模式下，全部 17 位与 ADCPPB1RESULT 的 PPBRESULT 字段的 17 位比较；12 位模式下，位 12:0 与 ADCPPB1RESULT 的 PPBRSULT 字段的位 12:0 比较。<br>复位类型：SYSRSn。 |

## 13.4.3.47 ADCPPB2CONFIG 寄存器

偏移地址：90h　复位值：0000h

ADC PPB2 配置寄存器

表 13-148 ADCPPB2CONFIG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | CBCEN | TWOSCOMPEN | CONFIG | | | |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | | | |

表 13-149 ADCPPB2CONFIG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-6 | RESERVED | R | 0h | 保留。 |
| 5 | CBCEN | R/W | 0h | ADC 后处理模块逐周期使能。置位后，若事件条件已不再存在，转换后硬件处理电路在进行转换时自动清除 ADCEVTSTAT。<br>复位类型：SYSRSn。 |
| 4 | TWOSCOMPEN | R/W | 0h | ADC 后处理模块 2 二进制补码使能。置位后，转换后硬件处理电路对偏移 / 参考减法单元的输出取二进制补码，再将结果存入 ADCPPB2RESULT。<br>0：ADCPPB2RESULT = ADCRESULTx - ADCPPB2OFFREF<br>1：ADCPPB2RESULT = ADCPPB2OFFREF - ADCRESULTx<br>复位类型：SYSRSn。 |
| 3-0 | CONFIG | R/W | 0h | ADC 后处理模块 2 配置。定义与该模块关联的 SOC/EOC/RESULT。<br>0000：SOC0/EOC0/RESULT0 与后处理模块 2 关联。<br>0001：SOC1/EOC1/RESULT1 与后处理模块 2 关联。<br>0010：SOC2/EOC2/RESULT2 与后处理模块 2 关联。<br>0011：SOC3/EOC3/RESULT3 与后处理模块 2 关联。<br>0100：SOC4/EOC4/RESULT4 与后处理模块 2 关联。<br>0101：SOC5/EOC5/RESULT5 与后处理模块 2 关联。<br>0110：SOC6/EOC6/RESULT6 与后处理模块 2 关联。<br>0111：SOC7/EOC7/RESULT7 与后处理模块 2 关联。<br>1000：SOC8/EOC8/RESULT8 与后处理模块 2 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 关联。<br>1001：SOC9/EOC9/RESULT9 与后处理模块 2 关联。<br>1010：SOC10/EOC10/RESULT10 与后处理模块 2 关联。<br>1011：SOC11/EOC11/RESULT11 与后处理模块 2 关联。<br>1100：SOC12/EOC12/RESULT12 与后处理模块 2 关联。<br>1101：SOC13/EOC13/RESULT13 与后处理模块 2 关联。<br>1110：SOC14/EOC14/RESULT14 与后处理模块 2 关联。<br>1111：SOC15/EOC15/RESULT15 与后处理模块 2 关联。<br>复位类型：SYSRSn。 |

### 13.4.3.48 ADCPPB2STAMP 寄存器

偏移地址：92h　复位值：0000h

ADC PPB2 采样延迟时间戳寄存器

表 13-150 ADCPPB2STAMP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | DLYSTAMP | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DLYSTAMP | | | | | | | |
| R-0h | | | | | | | |

表 13-151 ADCPPB2STAMP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11-0 | DLYSTAMP | R | 0h | ADC 后处理模块 2 延迟时间戳。SOC 开始采样时，从 ADCCOUNTER.FREECOUNT 的值减去 REQSTAMP，并将结果加载至此字段，得到 SOC 触发到实际开始采样之间的系统时钟周期数。<br>复位类型：SYSRSn。 |

## 13.4.3.49 ADCPPB2OFFCAL 寄存器

偏移地址：94h　复位值：0000h

ADC PPB2 偏移校准寄存器

表 13-152 ADCPPB2OFFCAL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | OFFCAL | |
| R-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OFFCAL | | | | | | | |
| R/W-0h | | | | | | | |

表 13-153 ADCPPB2OFFCAL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-10 | RESERVED | R | 0h | 保留。 |
| 9-0 | OFFCAL | R/W | 0h | ADC 后处理模块 2 偏移校正。以数字方式消除 ADCIN 电路固有的系统级偏移。将 ADC 输出减去此 10 位有符号值后，存入 ADCRESULT。<br>000h：不变，ADC 输出直接存入 ADCRESULT。<br>001h：将 ADC 输出 - 1 存入 ADCRESULT。<br>002h：将 ADC 输出 - 2 存入 ADCRESULT。<br>……<br>200h：将 ADC 输出 + 512 存入 ADCRESULT。<br>……<br>3FFh：将 ADC 输出 + 1 存入 ADCRESULT。<br>注意：16 位模式下，减法结果在存入 ADCRESULT 前饱和限制于 0000h 和 FFFFh；12 位模式下，饱和限制于 0000h 和 0FFFh。<br>注意：多个 PPB 指向同一 SOC 时，仅应用编号最高的 PPB 的 OFFCAL。<br>复位类型：SYSRSn。 |

## 13.4.3.50 ADCPPB2OFFREF 寄存器

偏移地址：96h　复位值：0000h

ADC PPB2 偏移参考寄存器

表 13-154 ADCPPB2OFFREF 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| OFFREF | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OFFREF | | | | | | | |
| R/W-0h | | | | | | | |

表 13-155 ADCPPB2OFFREF 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | OFFREF | R/W | 0h | ADC 后处理模块 2 偏移校正。通过减去参考值计算反馈误差，或将单极性信号转换为双极性信号。从 ADCRESULT 减去此 16 位无符号值，再经过可选的二进制补码处理，存入 ADCPPB2RESULT。此减法不作饱和处理。<br>0000h：不变，直接传递 ADCRESULT。<br>0001h：传递 ADCRESULT - 1。<br>0002h：传递 ADCRESULT - 2。<br>……<br>8000h：传递 ADCRESULT - 32,768。<br>……<br>FFFFh：传递 ADCRESULT - 65,535。<br>注意：12 位模式下，此寄存器仍为 16 位。用户必须确保仅向其中写入 12 位值。<br>复位类型：SYSRSn。 |

## 13.4.3.51 ADCPPB2TRIPHI 寄存器

偏移地址：98h　复位值：00000000h

ADC PPB2 高限跳闸寄存器

表 13-156 ADCPPB2TRIPHI 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | HSIGN |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LIMITHI | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LIMITHI | | | | | | | |
| R/W-0h | | | | | | | |

表 13-157 ADCPPB2TRIPHI 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | 保留。 |
| 16 | HSIGN | R/W | 0h | 高限符号位。16 位 ADC 模式下，这是 LIMITHI 字段的符号位（第 17 位）。<br>复位类型：SYSRSn。 |
| 15-0 | LIMITHI | R/W | 0h | ADC 后处理模块 2 高限跳闸阈值。设置数字比较器高限阈值。16 位模式下，全部 17 位与 ADCPPB2RESULT 的 PPBRESULT 字段的 17 位比较；12 位模式下，位 12:0 与 ADCPPB2RESULT 的 PPBRESULT 字段的位 12:0 比较。<br>复位类型：SYSRSn。 |

## 13.4.3.52 ADCPPB2TRIPLO 寄存器

偏移地址：9Ch　复位值：00000000h

ADC PPB2 低限跳闸 / 触发时间戳寄存器

表 13-158 ADCPPB2TRIPLO 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQSTAMP | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQSTAMP | | | | 保留 | | | LSIGN |
| R-0h | | | | R-0h | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LIMITLO | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LIMITLO | | | | | | | |
| R/W-0h | | | | | | | |

表 13-159 ADCPPB2TRIPLO 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | REQSTAMP | R | 0h | ADC 后处理模块 2 请求时间戳。触发将 ADCSOCFLG1 中关联的 SOC 标志置位时，将 ADCCOUNTER.FREECOUNT 的值加载至此字段。<br>复位类型：SYSRSn。 |
| 19-17 | RESERVED | R | 0h | 保留。 |
| 16 | LSIGN | R/W | 0h | 低限符号位。16 位 ADC 模式下，这是 LIMITLO 字段的符号位（第 17 位）。<br>复位类型：SYSRSn。 |
| 15-0 | LIMITLO | R/W | 0h | ADC 后处理模块 2 低限跳闸阈值。设置数字比较器低限阈值。16 位模式下，全部 17 位与 ADCPPB2RESULT 的 PPBRESULT 字段的 17 位比较；12 位模式下，位 12:0 与 ADCPPB2RESULT 的 PPBRSULT 字段的位 12:0 比较。<br>复位类型：SYSRSn。 |

## 13.4.3.53 ADCPPB3CONFIG 寄存器

偏移地址：A0h　复位值：0000h

ADC PPB3 配置寄存器

表 13-160 ADCPPB3CONFIG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | CBCEN | TWOSCOMPEN | CONFIG | | | |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | | | |

表 13-161 ADCPPB3CONFIG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-6 | RESERVED | R | 0h | 保留。 |
| 5 | CBCEN | R/W | 0h | ADC 后处理模块逐周期使能。置位后，若事件条件已不再存在，转换后硬件处理电路在进行转换时自动清除 ADCEVTSTAT。<br>复位类型：SYSRSn。 |
| 4 | TWOSCOMPEN | R/W | 0h | ADC 后处理模块 3 二进制补码使能。置位后，转换后硬件处理电路对偏移 / 参考减法单元的输出取二进制补码，再将结果存入 ADCPPB3RESULT。<br>0：ADCPPB3RESULT = ADCRESULTx - ADCPPB3OFFREF<br>1：ADCPPB3RESULT = ADCPPB3OFFREF - ADCRESULTx<br>复位类型：SYSRSn。 |
| 3-0 | CONFIG | R/W | 0h | ADC 后处理模块 3 配置。定义与该模块关联的 SOC/EOC/RESULT。<br>0000：SOC0/EOC0/RESULT0 与后处理模块 3 关联。<br>0001：SOC1/EOC1/RESULT1 与后处理模块 3 关联。<br>0010：SOC2/EOC2/RESULT2 与后处理模块 3 关联。<br>0011：SOC3/EOC3/RESULT3 与后处理模块 3 关联。<br>0100：SOC4/EOC4/RESULT4 与后处理模块 3 关联。<br>0101：SOC5/EOC5/RESULT5 与后处理模块 3 关联。<br>0110：SOC6/EOC6/RESULT6 与后处理模块 3 关联。<br>0111：SOC7/EOC7/RESULT7 与后处理模块 3 关联。<br>1000：SOC8/EOC8/RESULT8 与后处理模块 3 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 关联。<br>1001：SOC9/EOC9/RESULT9 与后处理模块 3 关联。<br>1010：SOC10/EOC10/RESULT10 与后处理模块 3 关联。<br>1011：SOC11/EOC11/RESULT11 与后处理模块 3 关联。<br>1100：SOC12/EOC12/RESULT12 与后处理模块 3 关联。<br>1101：SOC13/EOC13/RESULT13 与后处理模块 3 关联。<br>1110：SOC14/EOC14/RESULT14 与后处理模块 3 关联。<br>1111：SOC15/EOC15/RESULT15 与后处理模块 3 关联。<br>复位类型：SYSRSn。 |

### 13.4.3.54 ADCPPB3STAMP 寄存器

偏移地址：A2h　复位值：0000h

ADC PPB3 采样延迟时间戳寄存器

表 13-162 ADCPPB3STAMP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | DLYSTAMP | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DLYSTAMP | | | | | | | |
| R-0h | | | | | | | |

表 13-163 ADCPPB3STAMP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11-0 | DLYSTAMP | R | 0h | ADC 后处理模块 3 延迟时间戳。SOC 开始采样时，从 ADCCOUNTER.FREECOUNT 的值减去 REQSTAMP，并将结果加载至此字段，得到 SOC 触发到实际开始采样之间的系统时钟周期数。<br>复位类型：SYSRSn。 |

## 13.4.3.55 ADCPPB3OFFCAL 寄存器

偏移地址：A4h　复位值：0000h

ADC PPB3 偏移校准寄存器

表 13-164 ADCPPB3OFFCAL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | OFFCAL | |
| R-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OFFCAL | | | | | | | |
| R/W-0h | | | | | | | |

表 13-165 ADCPPB3OFFCAL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-10 | RESERVED | R | 0h | 保留。 |
| 9-0 | OFFCAL | R/W | 0h | ADC 后处理模块 3 偏移校正。以数字方式消除 ADCIN 电路固有的系统级偏移。将 ADC 输出减去此 10 位有符号值后，存入 ADCRESULT。<br>000h：不变，ADC 输出直接存入 ADCRESULT。<br>001h：将 ADC 输出 - 1 存入 ADCRESULT。<br>002h：将 ADC 输出 - 2 存入 ADCRESULT。<br>……<br>200h：将 ADC 输出 + 512 存入 ADCRESULT。<br>……<br>3FFh：将 ADC 输出 + 1 存入 ADCRESULT。<br>注意：16 位模式下，减法结果在存入 ADCRESULT 前饱和限制于 0000h 和 FFFFh；12 位模式下，饱和限制于 0000h 和 0FFFh。<br>注意：多个 PPB 指向同一 SOC 时，仅应用编号最高的 PPB 的 OFFCAL。<br>复位类型：SYSRSn。 |

## 13.4.3.56 ADCPPB3OFFREF 寄存器

偏移地址：A6h　复位值：0000h

ADC PPB3 偏移参考寄存器

表 13-166 ADCPPB3OFFREF 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| OFFREF | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OFFREF | | | | | | | |
| R/W-0h | | | | | | | |

表 13-167 ADCPPB3OFFREF 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | OFFREF | R/W | 0h | ADC 后处理模块 3 偏移校正。通过减去参考值计算反馈误差，或将单极性信号转换为双极性信号。从 ADCRESULT 减去此 16 位无符号值，再经过可选的二进制补码处理，存入 ADCPPB3RESULT。此减法不作饱和处理。<br>0000h：不变，直接传递 ADCRESULT。<br>0001h：传递 ADCRESULT - 1。<br>0002h：传递 ADCRESULT - 2。<br>……<br>8000h：传递 ADCRESULT - 32,768。<br>……<br>FFFFh：传递 ADCRESULT - 65,535。<br>注意：12 位模式下，此寄存器仍为 16 位。用户必须确保仅向其中写入 12 位值。<br>复位类型：SYSRSn。 |

## 13.4.3.57 ADCPPB3TRIPHI 寄存器

偏移地址：A8h　复位值：00000000h

ADC PPB3 高限跳闸寄存器

表 13-168 ADCPPB3TRIPHI 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | HSIGN |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LIMITHI | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LIMITHI | | | | | | | |
| R/W-0h | | | | | | | |

表 13-169 ADCPPB3TRIPHI 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | 保留。 |
| 16 | HSIGN | R/W | 0h | 高限符号位。16 位 ADC 模式下，这是 LIMITHI 字段的符号位（第 17 位）。<br>复位类型：SYSRSn。 |
| 15-0 | LIMITHI | R/W | 0h | ADC 后处理模块 3 高限跳闸阈值。设置数字比较器高限阈值。16 位模式下，全部 17 位与 ADCPPB3RESULT 的 PPBRESULT 字段的 17 位比较；12 位模式下，位 12:0 与 ADCPPB3RESULT 的 PPBRESULT 字段的位 12:0 比较。<br>复位类型：SYSRSn。 |

## 13.4.3.58 ADCPPB3TRIPLO 寄存器

偏移地址：ACh　复位值：00000000h

ADC PPB3 低限跳闸 / 触发时间戳寄存器

表 13-170 ADCPPB3TRIPLO 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQSTAMP | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQSTAMP | | | | 保留 | | | LSIGN |
| R-0h | | | | R-0h | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LIMITLO | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LIMITLO | | | | | | | |
| R/W-0h | | | | | | | |

表 13-171 ADCPPB3TRIPLO 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | REQSTAMP | R | 0h | ADC 后处理模块 3 请求时间戳。触发将 ADCSOCFLG1 中关联的 SOC 标志置位时，将 ADCCOUNTER.FREECOUNT 的值加载至此字段。<br>复位类型：SYSRSn。 |
| 19-17 | RESERVED | R | 0h | 保留。 |
| 16 | LSIGN | R/W | 0h | 低限符号位。16 位 ADC 模式下，这是 LIMITLO 字段的符号位（第 17 位）。<br>复位类型：SYSRSn。 |
| 15-0 | LIMITLO | R/W | 0h | ADC 后处理模块 3 低限跳闸阈值。设置数字比较器低限阈值。16 位模式下，全部 17 位与 ADCPPB3RESULT 的 PPBRESULT 字段的 17 位比较；12 位模式下，位 12:0 与 ADCPPB3RESULT 的 PPBRSULT 字段的位 12:0 比较。<br>复位类型：SYSRSn。 |

## 13.4.3.59 ADCPPB4CONFIG 寄存器

偏移地址：B0h　复位值：0000h

ADC PPB4 配置寄存器

表 13-172 ADCPPB4CONFIG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | CBCEN | TWOSCOMPEN | CONFIG | | | |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | | | |

表 13-173 ADCPPB4CONFIG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-6 | RESERVED | R | 0h | 保留。 |
| 5 | CBCEN | R/W | 0h | ADC 后处理模块逐周期使能。置位后，若事件条件已不再存在，转换后硬件处理电路在进行转换时自动清除 ADCEVTSTAT。<br>复位类型：SYSRSn。 |
| 4 | TWOSCOMPEN | R/W | 0h | ADC 后处理模块 4 二进制补码使能。置位后，转换后硬件处理电路对偏移 / 参考减法单元的输出取二进制补码，再将结果存入 ADCPPB4RESULT。<br>0：ADCPPB4RESULT = ADCRESULTx - ADCPPB4OFFREF<br>1：ADCPPB4RESULT = ADCPPB4OFFREF - ADCRESULTx<br>复位类型：SYSRSn。 |
| 3-0 | CONFIG | R/W | 0h | ADC 后处理模块 4 配置。定义与该模块关联的 SOC/EOC/RESULT。<br>0000：SOC0/EOC0/RESULT0 与后处理模块 4 关联。<br>0001：SOC1/EOC1/RESULT1 与后处理模块 4 关联。<br>0010：SOC2/EOC2/RESULT2 与后处理模块 4 关联。<br>0011：SOC3/EOC3/RESULT3 与后处理模块 4 关联。<br>0100：SOC4/EOC4/RESULT4 与后处理模块 4 关联。<br>0101：SOC5/EOC5/RESULT5 与后处理模块 4 关联。<br>0110：SOC6/EOC6/RESULT6 与后处理模块 4 关联。<br>0111：SOC7/EOC7/RESULT7 与后处理模块 4 关联。<br>1000：SOC8/EOC8/RESULT8 与后处理模块 4 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 关联。<br>1001：SOC9/EOC9/RESULT9 与后处理模块 4 关联。<br>1010：SOC10/EOC10/RESULT10 与后处理模块 4 关联。<br>1011：SOC11/EOC11/RESULT11 与后处理模块 4 关联。<br>1100：SOC12/EOC12/RESULT12 与后处理模块 4 关联。<br>1101：SOC13/EOC13/RESULT13 与后处理模块 4 关联。<br>1110：SOC14/EOC14/RESULT14 与后处理模块 4 关联。<br>1111：SOC15/EOC15/RESULT15 与后处理模块 4 关联。<br>复位类型：SYSRSn。 |

### 13.4.3.60 ADCPPB4STAMP 寄存器

偏移地址：B2h　复位值：0000h

ADC PPB4 采样延迟时间戳寄存器

表 13-174 ADCPPB4STAMP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | DLYSTAMP | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DLYSTAMP | | | | | | | |
| R-0h | | | | | | | |

表 13-175 ADCPPB4STAMP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11-0 | DLYSTAMP | R | 0h | ADC 后处理模块 4 延迟时间戳。SOC 开始采样时，从 ADCCOUNTER.FREECOUNT 的值减去 REQSTAMP，并将结果加载至此字段，得到 SOC 触发到实际开始采样之间的系统时钟周期数。<br>复位类型：SYSRSn。 |

## 13.4.3.61 ADCPPB4OFFCAL 寄存器

偏移地址：B4h　复位值：0000h

ADC PPB4 偏移校准寄存器

表 13-176 ADCPPB4OFFCAL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | OFFCAL | |
| R-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OFFCAL | | | | | | | |
| R/W-0h | | | | | | | |

表 13-177 ADCPPB4OFFCAL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-10 | RESERVED | R | 0h | 保留。 |
| 9-0 | OFFCAL | R/W | 0h | ADC 后处理模块 4 偏移校正。以数字方式消除 ADCIN 电路固有的系统级偏移。将 ADC 输出减去此 10 位有符号值后，存入 ADCRESULT。<br>000h：不变，ADC 输出直接存入 ADCRESULT。<br>001h：将 ADC 输出 - 1 存入 ADCRESULT。<br>002h：将 ADC 输出 - 2 存入 ADCRESULT。<br>……<br>200h：将 ADC 输出 + 512 存入 ADCRESULT。<br>……<br>3FFh：将 ADC 输出 + 1 存入 ADCRESULT。<br>注意：16 位模式下，减法结果在存入 ADCRESULT 前饱和限制于 0000h 和 FFFFh；12 位模式下，饱和限制于 0000h 和 0FFFh。<br>注意：多个 PPB 指向同一 SOC 时，仅应用编号最高的 PPB 的 OFFCAL。<br>复位类型：SYSRSn。 |

## 13.4.3.62 ADCPPB4OFFREF 寄存器

偏移地址：B6h　复位值：0000h

ADC PPB4 偏移参考寄存器

表 13-178 ADCPPB4OFFREF 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| OFFREF | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OFFREF | | | | | | | |
| R/W-0h | | | | | | | |

表 13-179 ADCPPB4OFFREF 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | OFFREF | R/W | 0h | ADC 后处理模块 4 偏移校正。通过减去参考值计算反馈误差，或将单极性信号转换为双极性信号。从 ADCRESULT 减去此 16 位无符号值，再经过可选的二进制补码处理，存入 ADCPPB4RESULT。此减法不作饱和处理。<br>0000h：不变，直接传递 ADCRESULT。<br>0001h：传递 ADCRESULT - 1。<br>0002h：传递 ADCRESULT - 2。<br>……<br>8000h：传递 ADCRESULT - 32,768。<br>……<br>FFFFh：传递 ADCRESULT - 65,535。<br>注意：12 位模式下，此寄存器仍为 16 位。用户必须确保仅向其中写入 12 位值。<br>复位类型：SYSRSn。 |

## 13.4.3.63 ADCPPB4TRIPHI 寄存器

偏移地址：B8h　复位值：00000000h

ADC PPB4 高限跳闸寄存器

表 13-180 ADCPPB4TRIPHI 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | HSIGN |
| R-0h | | | | | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LIMITHI | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LIMITHI | | | | | | | |
| R/W-0h | | | | | | | |

表 13-181 ADCPPB4TRIPHI 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-17 | RESERVED | R | 0h | 保留。 |
| 16 | HSIGN | R/W | 0h | 高限符号位。16 位 ADC 模式下，这是 LIMITHI 字段的符号位（第 17 位）。<br>复位类型：SYSRSn。 |
| 15-0 | LIMITHI | R/W | 0h | ADC 后处理模块 4 高限跳闸阈值。设置数字比较器高限阈值。16 位模式下，全部 17 位与 ADCPPB4RESULT 的 PPBRESULT 字段的 17 位比较；12 位模式下，位 12:0 与 ADCPPB4RESULT 的 PPBRESULT 字段的位 12:0 比较。<br>复位类型：SYSRSn。 |

## 13.4.3.64 ADCPPB4TRIPLO 寄存器

偏移地址：BCh　复位值：00000000h

ADC PPB4 低限跳闸 / 触发时间戳寄存器

表 13-182 ADCPPB4TRIPLO 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REQSTAMP | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REQSTAMP | | | | 保留 | | | LSIGN |
| R-0h | | | | R-0h | | | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| LIMITLO | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LIMITLO | | | | | | | |
| R/W-0h | | | | | | | |

表 13-183 ADCPPB4TRIPLO 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-20 | REQSTAMP | R | 0h | ADC 后处理模块 4 请求时间戳。触发将 ADCSOCFLG1 中关联的 SOC 标志置位时，将 ADCCOUNTER.FREECOUNT 的值加载至此字段。<br>复位类型：SYSRSn。 |
| 19-17 | RESERVED | R | 0h | 保留。 |
| 16 | LSIGN | R/W | 0h | 低限符号位。16 位 ADC 模式下，这是 LIMITLO 字段的符号位（第 17 位）。<br>复位类型：SYSRSn。 |
| 15-0 | LIMITLO | R/W | 0h | ADC 后处理模块 4 低限跳闸阈值。设置数字比较器低限阈值。16 位模式下，全部 17 位与 ADCPPB4RESULT 的 PPBRESULT 字段的 17 位比较；12 位模式下，位 12:0 与 ADCPPB4RESULT 的 PPBRSULT 字段的位 12:0 比较。<br>复位类型：SYSRSn。 |

## 13.4.3.65 ADCINTCYCLE 寄存器

偏移地址：DEh　复位值：0000h

ADC 提前中断生成周期

表 13-184 ADCINTCYCLE 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DELAY | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DELAY | | | | | | | |
| R/W-0h | | | | | | | |

表 13-185 ADCINTCYCLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | DELAY | R/W | 0h | ADC 提前中断生成周期延迟。以系统时钟周期为单位，定义从 ADCSOC 下降沿到中断产生的延迟。<br>复位类型：SYSRSn。 |

## 13.4.3.66 ADCINLTRIM2 寄存器

偏移地址：E4h　复位值：8000h

ADC 线性度修调 2 寄存器

表 13-186 ADCINLTRIM2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INLTRIM63TO32 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INLTRIM63TO32 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INLTRIM63TO32 | | | | | | | |
| R/W-80h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INLTRIM63TO32 | | | | | | | |
| R/W-0h | | | | | | | |

表 13-187 ADCINLTRIM2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | INLTRIM63TO32 | R/W | 8000h | ADC 线性度修调位 63-32。除非勘误或其他文档明确要求，否则不应修改此寄存器。修改其内容可能导致模块运行超出数据手册规格。<br>复位类型：SYSRSn。 |

### 13.4.3.67 ADCINLTRIM3 寄存器

偏移地址：E8h　复位值：00000000h

ADC 线性度修调 3 寄存器

表 13-188 ADCINLTRIM3 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| INLTRIM95TO64 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INLTRIM95TO64 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INLTRIM95TO64 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INLTRIM95TO64 | | | | | | | |
| R/W-0h | | | | | | | |

表 13-189 ADCINLTRIM3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | INLTRIM95TO64 | R/W | 0h | ADC 线性度修调位 95-64。除非勘误或其他文档明确要求，否则不应修改此寄存器。修改其内容可能导致模块运行超出数据手册规格。<br>复位类型：SYSRSn。 |

## 13.4.4 ADC 寄存器与 DriverLib 函数对应表

表 13-190 ADC 寄存器与 DriverLib 函数对应表

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| ADCCTL1 | adc.h | ADC_setInterruptPulseMode |
| ADCCTL1 | adc.h | ADC_enableConverter |
| ADCCTL1 | adc.h | ADC_disableConverter |
| ADCCTL1 | adc.h | ADC_isBusy |
| ADCCTL2 | adc.h | ADC_setPrescaler |
| ADCBURSTCTL | adc.h | ADC_setBurstModeConfig |
| ADCBURSTCTL | adc.h | ADC_enableBurstMode |
| ADCBURSTCTL | adc.h | ADC_disableBurstMode |
| ADCINTFLG | adc.h | ADC_getInterruptStatus |
| ADCINTFLG | adc.h | ADC_clearInterruptStatus |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| ADCINTFLGCLR | adc.h | ADC_clearInterruptStatus |
| ADCINTOVF | adc.h | ADC_getInterruptOverflowStatus |
| ADCINTOVF | adc.h | ADC_clearInterruptOverflowStatus |
| ADCINTOVFCLR | adc.h | ADC_clearInterruptOverflowStatus |
| ADCINTSEL1N2 | adc.h | ADC_enableInterrupt |
| ADCINTSEL1N2 | adc.h | ADC_disableInterrupt |
| ADCINTSEL1N2 | adc.h | ADC_setInterruptSource |
| ADCINTSEL1N2 | adc.h | ADC_enableContinuousMode |
| ADCINTSEL1N2 | adc.h | ADC_disableContinuousMode |
| ADCINTSEL3N4 | - | 参见 INTSEL1N2 |
| ADCSOCPRICTL | adc.h | ADC_setSOCPriority |
| ADCINTSOCSEL1 | adc.h | ADC_setInterruptSOCTrigger |
| ADCINTSOCSEL2 | - | 参见 INTSOCSEL1 |
| ADCSOCFLG1 | - | - |
| ADCSOCFRC1 | adc.h | ADC_forceSOC |
| ADCSOCFRC1 | adc.h | ADC_forceMultipleSOC |
| ADCSOCOVF1 | - | - |
| ADCSOCOVFCLR1 | - | - |
| ADCSOC0CTL | adc.h | ADC_setupSOC |
| ADCSOC1CTL | - | 参见 SOC0CTL |
| ADCSOC2CTL | - | 参见 SOC0CTL |
| ADCSOC3CTL | - | 参见 SOC0CTL |
| ADCSOC4CTL | - | 参见 SOC0CTL |
| ADCSOC5CTL | - | 参见 SOC0CTL |
| ADCSOC6CTL | - | 参见 SOC0CTL |
| ADCSOC7CTL | - | 参见 SOC0CTL |
| ADCSOC8CTL | - | 参见 SOC0CTL |
| ADCSOC9CTL | - | 参见 SOC0CTL |
| ADCSOC10CTL | - | 参见 SOC0CTL |
| ADCSOC11CTL | - | 参见 SOC0CTL |
| ADCSOC12CTL | - | 参见 SOC0CTL |
| ADCSOC13CTL | - | 参见 SOC0CTL |
| ADCSOC14CTL | - | 参见 SOC0CTL |
| ADCSOC15CTL | - | 参见 SOC0CTL |
| ADCEVTSTAT | adc.h | ADC_getPPBEventStatus |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| ADCEVTCLR | adc.h | ADC_clearPPBEventStatus |
| ADCEVTSEL | adc.h | ADC_enablePPBEvent |
| ADCEVTSEL | adc.h | ADC_disablePPBEvent |
| ADCEVTINTSEL | adc.h | ADC_enablePPBEventInterrupt |
| ADCEVTINTSEL | adc.h | ADC_disablePPBEventInterrupt |
| ADCOSDETECT | adc.h | ADC_configOSDetectMode |
| ADCCOUNTER | - | - |
| ADCREV | - | - |
| ADCOFFTRIM | adc.c | ADC_setOffsetTrim |
| ADCOFFTRIM | adc.c | ADC_setOffsetTrimAll |
| ADCPPB1CONFIG | adc.h | ADC_setupPPB |
| ADCPPB1CONFIG | adc.h | ADC_enablePPBEventCBCClear |
| ADCPPB1CONFIG | adc.h | ADC_disablePPBEventCBCClear |
| ADCPPB1CONFIG | adc.h | ADC_enablePPBTwosComplement |
| ADCPPB1CONFIG | adc.h | ADC_disablePPBTwosComplement |
| ADCPPB1STAMP | adc.h | ADC_getPPBDelayTimeStamp |
| ADCPPB1OFFCAL | adc.h | ADC_setPPBCalibrationOffset |
| ADCPPB1OFFREF | adc.h | ADC_setPPBReferenceOffset |
| ADCPPB1TRIPHI | adc.c | ADC_setPPBTripLimits |
| ADCPPB1TRIPLO | adc.c | ADC_setPPBTripLimits |
| ADCPPB2CONFIG | - | 参见 PPB1CONFIG |
| ADCPPB2STAMP | - | 参见 PPB1STAMP |
| ADCPPB2OFFCAL | - | 参见 PPB1OFFCAL |
| ADCPPB2OFFREF | - | 参见 PPB1OFFREF |
| ADCPPB2TRIPHI | - | 参见 PPB1TRIPHI |
| ADCPPB2TRIPLO | - | 参见 PPB1TRIPLO |
| ADCPPB3CONFIG | - | 参见 PPB1CONFIG |
| ADCPPB3STAMP | - | 参见 PPB1STAMP |
| ADCPPB3OFFCAL | - | 参见 PPB1OFFCAL |
| ADCPPB3OFFREF | - | 参见 PPB1OFFREF |
| ADCPPB3TRIPHI | - | 参见 PPB1TRIPHI |
| ADCPPB3TRIPLO | - | 参见 PPB1TRIPLO |
| ADCPPB4CONFIG | - | 参见 PPB1CONFIG |
| ADCPPB4STAMP | - | 参见 PPB1STAMP |
| ADCPPB4OFFCAL | - | 参见 PPB1OFFCAL |

| 寄存器 | 文件 | DriverLib 函数 |
| --- | --- | --- |
| ADCPPB4OFFREF | - | 参见 PPB1OFFREF |
| ADCPPB4TRIPHI | - | 参见 PPB1TRIPHI |
| ADCPPB4TRIPLO | - | 参见 PPB1TRIPLO |
| ADCINTCYCLE | adc.h | ADC_setInterruptCycleOffset |
| ADCINLTRIM2 | adc.c | ADC_setINLTrim |
| ADCINLTRIM3 | - | - |
| ADCRESULT0 | adc.h | ADC_readResult |
| ADCRESULT1 | - | 参见 RESULT0 |
| ADCRESULT2 | - | 参见 RESULT0 |
| ADCRESULT3 | - | 参见 RESULT0 |
| ADCRESULT4 | - | 参见 RESULT0 |
| ADCRESULT5 | - | 参见 RESULT0 |
| ADCRESULT6 | - | 参见 RESULT0 |
| ADCRESULT7 | - | 参见 RESULT0 |
| ADCRESULT8 | - | 参见 RESULT0 |
| ADCRESULT9 | - | 参见 RESULT0 |
| ADCRESULT10 | - | 参见 RESULT0 |
| ADCRESULT11 | - | 参见 RESULT0 |
| ADCRESULT12 | - | 参见 RESULT0 |
| ADCRESULT13 | - | 参见 RESULT0 |
| ADCRESULT14 | - | 参见 RESULT0 |
| ADCRESULT15 | - | 参见 RESULT0 |
| ADCPPB1RESULT | adc.h | ADC_readPPBResult |
| ADCPPB2RESULT | - | 参见 PPB1RESULT |
| ADCPPB3RESULT | - | 参见 PPB1RESULT |
| ADCPPB4RESULT | - | 参见 PPB1RESULT |

# 14 比较器子系统（CMPSS）

比较器子系统（CMPSS）由模拟比较器和辅助电路组成，适用于峰值电流模式控制、开关模式电源、功率因数校正、电压跳闸监控等电力应用场景。

本器件包含 1 种 CMPSS 模块。

## 14.1 简介

比较器子系统围绕多个模块构建而成。每个子系统包含两个比较器、两个参考 12 位 DAC（数模转换器）和两个数字滤波器。此外，子系统还包含一个斜坡发生器，且该斜坡发生器仅支持向下斜坡模式。

每个模块中的比较器以 “H” 或 “L” 标识，分别代表高（High）和低（Low）。每个比较器会生成一个数字输出，用于指示同相输入端电压是否大于反相输入端电压。比较器的同相输入端由外部引脚驱动；反相输入端可由外部引脚驱动，也可由可编程参考 12 位 DAC 驱动。每个比较器的输出会经过一个可编程数字滤波器，以滤除虚假跳闸信号；若无需滤波，也可直接使用未滤波输出。

斜坡发生器（仅完整版本的 CMPSS 模块具备）可选择性地用于控制子系统中高电平比较器的参考 12 位 DAC 值。

CMPSS 的框图如图所示。

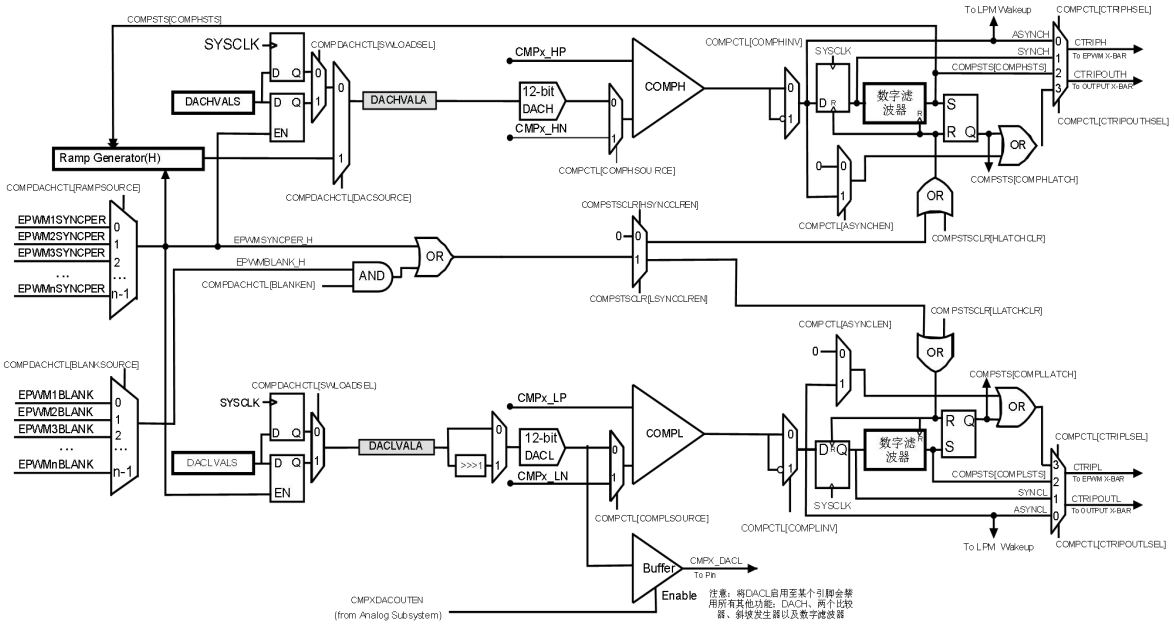



图 14-1 CMPSS 模块框图

CTRIPx（x="H" 或 "L"）信号连接至 ePWM 交叉开关，用于 ePWM 跳闸响应。有关 ePWM 交叉开关多路复用器配置的详细说明，请参考 ePWM 章节。CTRIPxOUTx（x="H" 或 "L"）信号连接至输出交叉开关，用于外部信号传输。有关输出交叉开关多路复用器配置的详细说明，请参考 GPIO 章节。CMPxDACL 是低电平比较器 DAC 输出，仅在 CMPSS1 模块中可用。若要使能该 DAC 输出，需通过模拟子系统的寄存器 CMPxDACOUTEN 进行设置。

注意：

若将 DACL 输出至引脚，则会禁用所有其他功能，包括高电平 DAC、两个比较器、斜坡发生器和数字滤波器。

## 14.2 主要特性

每个 CMPSS 包含以下功能：

- 两个模拟比较器
- 两个可编程参考 12 位 DAC
- 一个递减型斜坡发生器
- 两个数字滤波器，最大滤波器时钟预分频系数为 $2^{16}$

- 支持子模块与 EPWMSYNCPER 信号同步
- 支持通过 EPWMBLANK 信号扩展清零信号
- 支持输出与 SYSCLK（系统时钟）同步
- 支持输出锁存功能
- 支持输出反相功能
- 输入支持滞回特性
- 比较器反相输入端可选择由外部信号或参考 DAC 驱动
- VDDA 作为 DAC 参考电压
- 低电平比较器 DAC 输出可路由至外部引脚（仅部分实例支持，与比较功能互斥）

## 14.3 功能描述

### 14.3.1 比较器

当同相输入端电压大于反相输入端电压时，比较器输出高电平数字信号；当同相输入端电压小于反相输入端电压时，输出低电平数字信号。比较器的示意图如图所示。

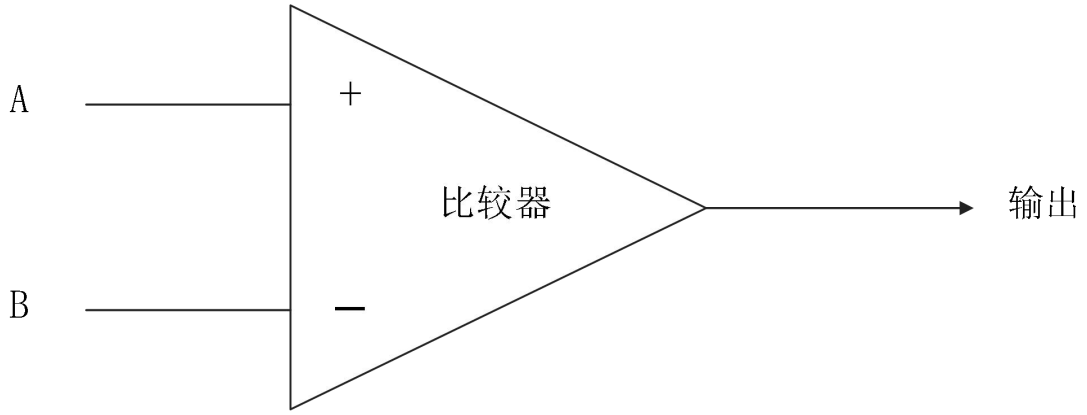



图 14-2 比较器框图

表 14-1 比较器输入与输出关系

| 输入电压关系 | 输出 |
|---|---|
| 电压 A > 电压 B | 1 |
| 电压 A < 电压 B | 0 |

### 14.3.2 参考 DAC

每个参考 12 位 DAC 可配置为向相应比较器的反相输入端提供参考电压。部分 CMPSS 实例还允许将低电平 DAC 输出路由至引脚，用作外部 DAC。在此情况下，CMPSS 模块的所有其他功能均不可用，包括高电平 DAC、两个比较器、斜坡发生器和数字滤波器。

每个参考 12 位 DAC 均配有两组 DACxVAL 寄存器：DACxVALA 和 DACxVALS。DACxVALA 是只读寄存器，用于主动控制参考 12 位 DAC 的值；DACxVALS 是可写影子寄存器，其值可立即加载至 DACxVALA，或与下一个 EPWMSYNCPER 事件同步加载。高 / 低电平参考 12 位 DAC（DACx）可选择从斜坡发生器获取 DACxVALA 值，而非从寄存器 DACxVALS 获取。

参考 12 位 DAC 的工作范围由 DACREF 和 VSSA 界定。

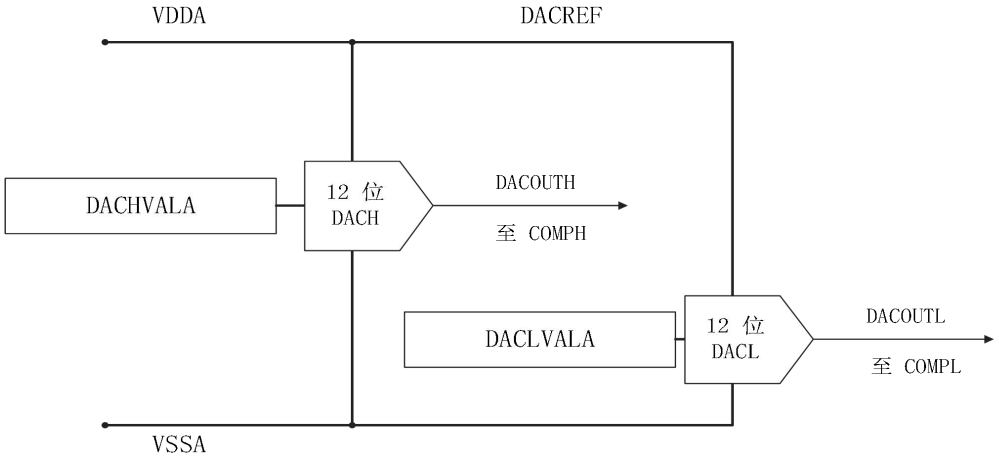



图 14-3 参考 DAC 框图

参考 12 位 DAC 的输出可通过公式（14-1）计算：

$$DACOUT = \frac{(1 + DACVALA) \times DACREF}{4096} \quad (14\text{-}1)$$

注意：

当 DACH 和 DACL 分别驱动高 / 低电平比较器时，其中一个比较器的跳闸可能会暂时干扰另一个比较器的 DAC 输出。该干扰的幅度和持续时间在器件数据手册中分别以 “CMPSS DAC 输出干扰” 和 “CMPSS DAC 干扰时间” 进行规定。

用户在设计系统时必须格外谨慎：若输入信号跨越 DACH 或 DACL 的阈值并触发相应比较器跳闸，则在 “CMPSS DAC 干扰时间” 内，输入信号需与另一个比较器的跳闸点保持大于 “CMPSS DAC 输出干扰” 的差值。此外，DACH 的设置值必须始终高于 DACL 的设置值。若不使用 DACL，需将 DACLVALS 寄存器设置为 0，以避免比较器 COMPL 跳闸并影响 DACH，此时 DACHVALS 的设置无限制；同理，若不使用 DACH，需将 DACHVALS 寄存器设置为最大值。可在编程参考 DAC 值之前使能 CMPSS 实例。

## 14.3.3 斜坡发生器

本节将详细介绍斜坡发生器的特性与工作原理。

### 14.3.3.1 斜坡发生器概述

当选择斜坡发生器模式时，其会为高电平参考 12 位 DAC 提供下降斜坡输入。在此模式下，参考 12 位 DAC 以 RAMPSTS 递减寄存器的高 12 位作为输入，而 RAMPSTS 递减寄存器的低 4 位实际上用作下降斜坡速率的预分频器，可通过 RAMPxSTEPVALA 进行配置。

通过设置 DACSOURCE = 1 可使能斜坡发生器。选择 DACSOURCE = 1 后，RAMPSTS 会从 RAMPxREFS 加载初始值，并保持静态直至接收到选定的 EPWMSYNCPER 信号。接收到选定的 EPWMSYNCPER 信号后，后续每个 SYSCLK 周期都会从 RAMPSTS 中减去 RAMPHSTEPVALA 的值。

若要避免在 EPWMSYNCPER 事件后的一个 SYSCLK 周期内立即开始减法操作，可使用 RAMPDLYA 寄存器（用作延迟计数器）来延迟 RAMPSTS 的减法启

动。接收到 EPWMSYNCPER 事件后，RAMPDLYA 的值每个 SYSCLK 周期递减 1，直至减至 0，此时才开始 RAMPSTS 的减法操作。

### 14.3.3.2 斜坡发生器工作原理

斜坡发生器会在 DACSOURCE、EPWMSYNCPER、EPWMSYNCPER_H 和 COMPHSTS 的每个上升沿发生状态变化：

DACSOURCE 上升沿：RAMPHREFA、RAMPHSTEPVALA 和 RAMPDLYA 从影子寄存器加载值；RAMPSTS 从 RAMPHREFS 加载值。

选定的 EPWMSYNCPER_H 上升沿：RAMPHREFA、RAMPHSTEPVALA 和 RAMPDLYA 从影子寄存器加载值；RAMPSTS 从 RAMPHREFS 加载值，并在 RAMPDLYA 计数器减至 0 时开始递减。

RAMPLOADSEL = 1 时 COMPHSTS 上升沿：RAMPHREFA、RAMPxREFA、RAMPxSTEPVALA 和 RAMPDLYA 从影子寄存器加载值；RAMPSTS 从 RAMPxREFS 加载值，并停止递减。

RAMPLOADSEL = 0 时 COMPHSTS 上升沿：RAMPSTS 从 RAMPHREFA 加载值，并停止递减。

此外，若 RAMPSTS 的值减至 0，则该寄存器将保持静态为 0，直至接收到下一个 EPWMSYNCPER_H 信号。这些状态变化如图中的斜坡发生器框图所示。

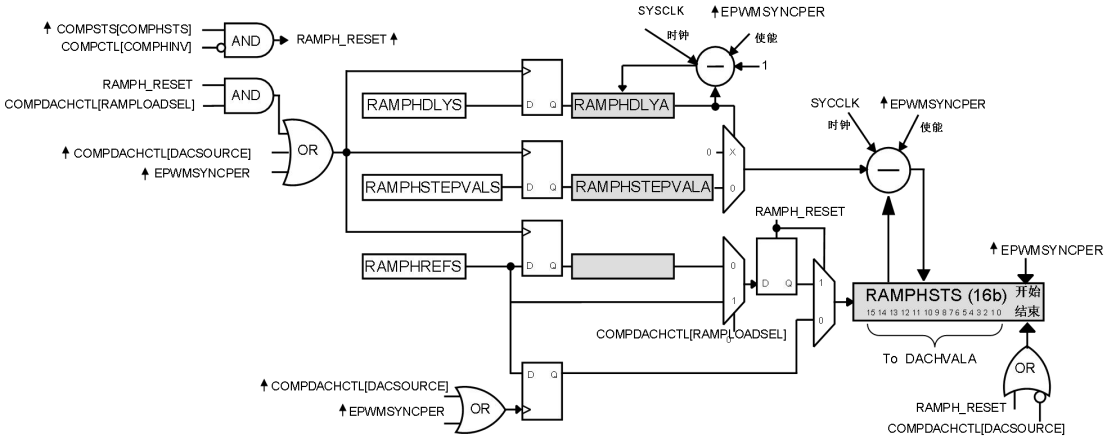



图 14-4 斜坡发生器框图

#### 14.3.3.3 边界情况下的斜坡发生器工作原理

由于斜坡发生器会在 EPWMSYNCPER_H 和 COMPHSTS 的每个上升沿发生状态变化，因此当这两个事件同时发生或间隔极短时，会出现以下行为：

情况 1：COMPHSTS 上升沿比 EPWMSYNCPER_H 上升沿早一个或多个周期发生。RAMPSTS 在 COMPHSTS 上升沿事件时停止递减；当 RAMPDLYA 减至 0 时，RAMPSTS 在 EPWMSYNCPER_H 上升沿事件时开始递减。

情况 2：COMPHSTS 上升沿与 EPWMSYNCPER_H 上升沿同时发生。EPWMSYNCPER 上升沿事件优先，当 RAMPDLYA 减至 0 时 RAMPSTS 开始递减；COMPHSTS 上升沿事件被忽略，不会暂停 RAMPSTS。

情况 3：COMPHSTS 上升沿在 EPWMSYNCPER_H 上升沿之后、但 RAMPDLYA 减至 0 之前发生。当 RAMPDLYA 减至 0 时，RAMPSTS 不会递减。

情况 4：COMPHSTS 上升沿与 EPWMSYNCPER_H 上升沿后 RAMPDLYA 减至 0 的时刻同时发生。RAMPSTS 不会递减。

上述行为也如图所示。

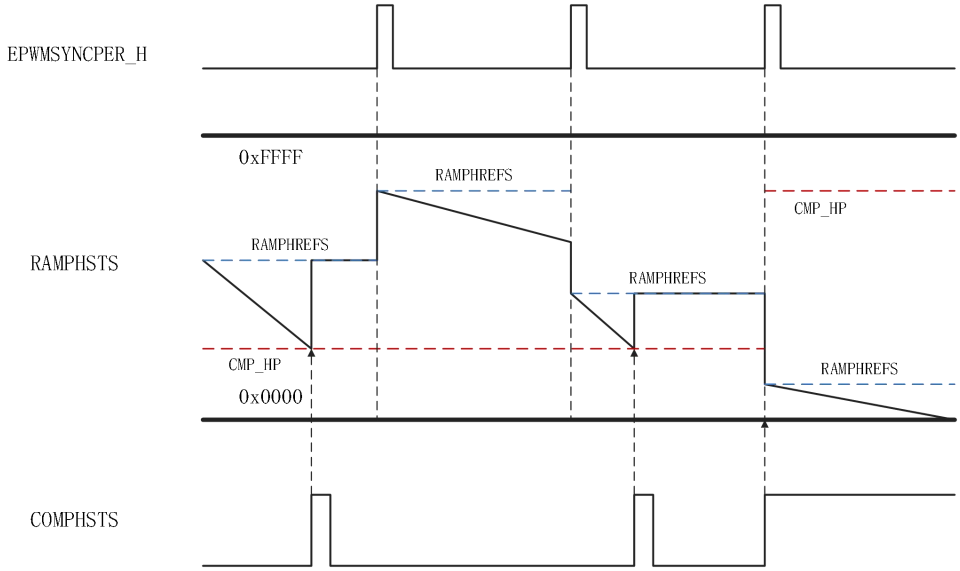



图 14-5 斜坡发生器工作时序图

### 14.3.4 数字滤波器

数字滤波器基于从输入获取的 FIFO（先进先出）采样窗口（SAMPWIN）工作，滤波器输出为采样窗口内的多数值（多数值由阈值（THRESH）定义）。若未达到多数阈值，滤波器输出保持不变。

为确保滤波器正常工作，THRESH 的值必须大于 SAMPWIN/2 且小于等于 SAMPWIN。

预分频功能（CLKPRESCALE）用于确定滤波器采样速率，FIFO 每隔预分频系统时钟周期捕获一个采样值，并丢弃 FIFO 中的旧数据。

注意：对于 SAMPWIN、THRESH 和 CLKPRESCALE，数字滤波器内部实际使用的值均为配置值 + 1。即实际采样数 = SAMPWIN + 1，实际阈值 = THRESH + 1，实际预分频系数 = CLKPRESCALE + 1。

数字滤波器的概念模型如图所示。

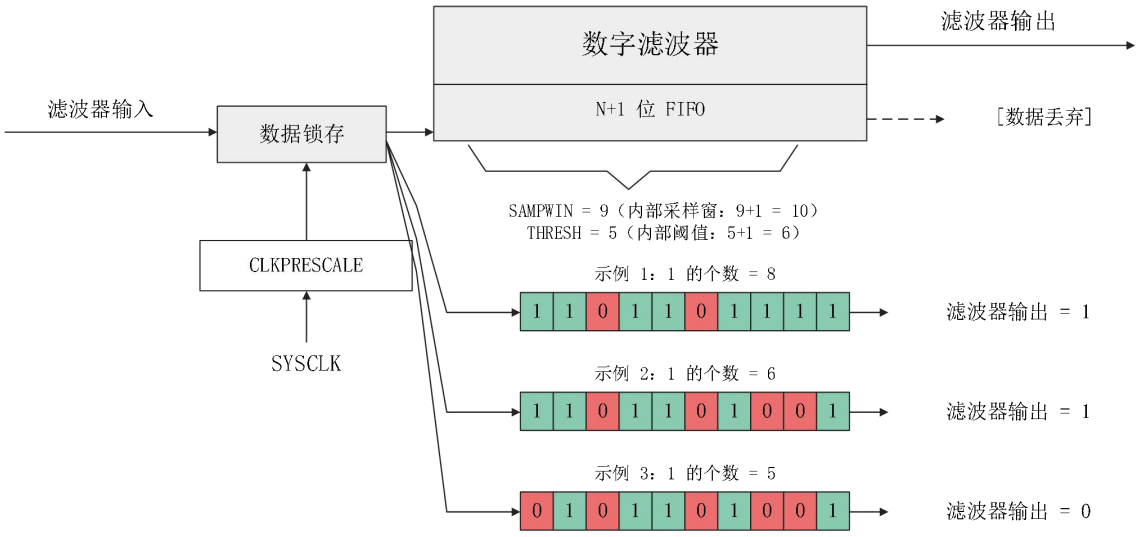



图 14-6 数字滤波器工作原理示意图

滤波器实现的等效 C 代码如下：

```
if (FILTER_OUTPUT == 0) {
if (Num_1s_in_SAMPWIN >= THRESH) {
FILTER_OUTPUT = 1;
```

```
}
} else {
if (Num_0s_in_SAMPWIN >= THRESH) {
FILTER_OUTPUT = 0;
}
}
```

#### 14.3.4.1 滤波器初始化流程

为确保数字滤波器正常工作，建议按照以下初始化流程操作：

1.配置并使能比较器。

2.配置数字滤波器工作参数：

- 设置 SAMPWIN，定义 FIFO 窗口中要监控的采样数。
- 设置 THRESH，定义多数判定所需的阈值。
- 设置 CLKPRESCALE，定义数字滤波器时钟预分频值。

3.通过设置 FILINIT 初始化数字 FIFO 窗口中的采样值。

（注：使用 FILINIT 会清空 FIFO 中的所有样本，输出状态会变为 0，属于 filter 的使能信号，需要用户注意）

4. 如果需要使用锁存路径，可使用 COMPSTSCLR 清除 COMPSTS 锁存器。

5. 配置 CTRIP 和 CTRIPOUT 信号路径。

6. 如果有需要，配置目标模块（例如 ePWM、GPIO 等）以接收经过滤波后的信号。

### 14.3.5 CMPSS 的使用

#### 14.3.5.1 LATCHCLR、EPWMSYNCPER 和 EPWMBLANK 信号

LATCHCLR 信号在经过所需延迟后，将数字滤波器、同步块和锁存输出置于复位状态（0）。可通过软件使用 xLATCHCLR（x=H 或 L）激活 LATCHCLR 信号；

当 xSYNCCLREN（x=H 或 L）置 1 时，也可通过 EPWMSYNCPER 激活 LATCHCLR 信号。若需要更长的 LATCHCLR 信号，可通过设置 BLANKEN，利用 EPWMBLANK 信号扩展 LATCHCLR 信号的持续时间。

EPWMSYNCPER 和 EPWMBLANK（BLANKWDW）分别来自 ePWM 的时基子模块和数字比较子模块。有关这两个信号的生成方式的详细说明，请参考 ePWM 章节中相应的子模块部分。

当 COMPDACCTL [SWLOADSEL] = 1 时，用于加载 DACxVALA 的 EPWMSYNCPER 信号为电平触发加载。若 ePWM 的 TBCTR 和 TBPRD 均为 0，则 EPWMSYNCPER 保持高电平，无论 COMPDACCTL [SWLOADSEL] 的值如何，DACxVALA 都会立即从 DACxVALS 加载值。因此，需先配置 ePWM，再将 COMPDACCTL [SWLOADSEL] 设置为 1。

注意：

CMPSS 从 ePWM 接收的同步信号名称已从 PWMSYNC 更新为 EPWMSYNCPER（SYNCPER/PWMSYNCPER/EPWMxSYNCPER），以避免与其他 ePWM 同步信号（EPWMSYNCI 和 EPWMSYNCO）混淆。

#### 14.3.5.2 同步器、数字滤波器和锁存器延迟

同步块会引入 1-2 个系统时钟周期的延迟；若数字滤波器被旁路（所有滤波器设置均为 0），则数字滤波器会引入 2 个系统时钟周期的延迟；锁存器会引入 1 个系统时钟周期的延迟。

#### 14.3.5.3 CMPSS 校准

CMPSS 存在两种偏移误差来源：比较器偏移误差和 COMPDAC 偏移误差。在数据手册中，比较器偏移误差称为输入参考偏移误差，COMPDAC 偏移误差称为静态偏移误差，具体数值请参考器件数据手册。

若比较器的两个输入端均由引脚驱动，则仅存在比较器偏移误差；若比较器的反相输入端由 COMPDAC 驱动，则仅存在 COMPDAC 偏移误差（因 COMPDAC 偏移误差已包含比较器偏移误差）。

由于存在偏移误差，必须对 CMPSS 进行校准，以确保跳闸在预期电平发生。若比较器的反相输入端由 COMPDAC 驱动，可按照以下流程进行校准：

校准前注意事项：

1. 比较器的非反相输入端需要一个静态直流信号。

2. 在校准过程中可以禁用迟滞功能，校准完成后可重新启用。

3. 输入信号有噪声会增加校准难度，因此要根据非反相输入信号的噪声情况，使用具有非零滤波器设置的锁存器。

校准流程（COMPDAC 值递减扫描法）：

1. 将 compdac 的起始值设为最大值 0xFFF。

l 可选操作：不将 compdac 的起始值设为最大值，而是设置为 Vtarget + 静态偏移误差 + 裕量。其中，Vtarget 是非反相输入端的近似直流电压，静态偏移误差是 compdac 偏移误差的规格值，裕量是一定的保护带宽。这种方式能加快校准速度，但仅在已知 Vtarget 的情况下适用。另外，如果不知道 Vtarget，可以使用 ADC 来转换得到 Vtarget 的值。

2. 将 compdac 的值减 1。

3. 等待 compdac 稳定。

4. 清除锁存器。

5. 等待锁存器可能的置位。

6. 如果锁存器置位，表明找到了跳变代码，校准结束。

可选校验操作：可以通过以下步骤再次检查跳变代码：

a. 将 compdac 的值加 1。

b. 清除锁存器。

c. 等待锁存器可能的置位。

d. 锁存器应保持未置位状态。

7. 如果锁存器未置位，返回步骤 2 并重复上述过程。

若比较器的两个输入端均由引脚驱动，也可按照上述流程校准 CMPSS，此时需通过外部扫描比较器反相输入端的电压。

#### 14.3.5.4 CMPSS 时钟的使能与禁用

若在比较器工作期间禁用 CMPSS 模块的时钟，会出现以下行为：

比较器不受影响，继续根据输入端电压产生跳闸信号。

若参考 12 位 DAC 驱动比较器的反相输入端，则反相输入端电压保持静态不变，但 DACVALA 无法再通过斜坡发生器或 DACVALS 更新。

斜坡发生器、同步块和数字滤波器冻结在当前状态。

重新使能 CMPSS 的时钟后，时钟将恢复至禁用前的状态。

### 14.3.6 CMPSS DAC 输出

部分完整版本的 CMPSS 模块实例可选择将低电平 DAC 输出路由至外部引脚，用作外部 DAC。该功能受以下限制：

此功能仅在部分实例上可用；请查阅器件数据手册，以确定特定实例的 CMPSS DAC 是否有引脚引出。

当启用输出 DAC 时，CMPSS 模块的所有其他功能都不可用。这包括高 DAC、两个比较器、斜坡生成逻辑和数字滤波器。

有关 CMPSS DAC 输出的有效分辨率，请参阅器件数据手册。

注意：

CMPSS 低电平 DAC 输出不一定与低电平比较器输入共用同一引脚。有关 CMPSS 输出 DAC 对应的引脚，请参考器件数据手册的引脚分配表。

## 14.4 寄存器描述

本节介绍 CMPSS 寄存器。

### 14.4.1 CMPSS 基址表

表 14-2 CMPSS 基址表

| 实例 | 结构 | DriverLib 名称 | 基址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| Cmpss1Regs | CMPSS_REGS | CMPSS1_BASE | 0x0061_C000 | 是 | 是 | 是 |
| Cmpss2Regs | CMPSS_REGS | CMPSS2_BASE | 0x0061_C040 | 是 | 是 | 是 |
| Cmpss3Regs | CMPSS_REGS | CMPSS3_BASE | 0x0061_C080 | 是 | 是 | 是 |
| Cmpss4Regs | CMPSS_REGS | CMPSS4_BASE | 0x0061_C0C0 | 是 | 是 | 是 |

### 14.4.2 CMPSS_REGS 寄存器

表中列出了 CMPSS_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，不应修改寄存器内容。

表 14-3 CMPSS_REGS 寄存器

| 偏移 | 首字母缩写词 | 寄存器名称 |
|---|---|---|
| 0x0 | COMPCTL | CMPSS 比较器控制寄存器 |
| 0x2 | COMPHYSCTL | CMPSS 比较器迟滞控制寄存器 |
| 0x4 | COMPSTS | CMPSS 比较器状态寄存器 |
| 0x6 | COMPSTSCLR | CMPSS 比较器状态清零寄存器 |
| 0x8 | COMPDACCTL | CMPSS DAC 控制寄存器 |
| 0xC | DACHVALS | CMPSS 高 DAC 值影子寄存器 |
| 0xE | DACHVALA | CMPSS 高 DAC 值活动寄存器 |
| 0x10 | RAMPMAXREFA | CMPSS 斜坡最大参考活动寄存器 |
| 0x14 | RAMPMAXREFS | CMPSS 斜坡最大参考影子寄存器 |
| 0x18 | RAMPDECVALA | CMPSS 斜坡递减值活动寄存器 |
| 0x1C | RAMPDECVALS | CMPSS 斜坡递减值影子寄存器 |
| 0x20 | RAMPSTS | CMPSS 斜坡状态寄存器 |
| 0x24 | DACLVALS | CMPSS 低 DAC 值影子寄存器 |
| 0x26 | DACLVALA | CMPSS 低 DAC 值活动寄存器 |
| 0x28 | RAMPDLYA | CMPSS 斜坡延迟活动寄存器 |
| 0x2A | RAMPDLYS | CMPSS 斜坡延迟影子寄存器 |
| 0x2C | CTRIPLFILCTL | CTRIPL 滤波器控制寄存器 |
| 0x2E | CTRIPLFILCLKCTL | CTRIPL 滤波器时钟控制寄存器 |
| 0x30 | CTRIPHFILCTL | CTRIPH 滤波器控制寄存器 |
| 0x32 | CTRIPHFILCLKCTL | CTRIPH 滤波器时钟控制寄存器 |
| 0x34 | COMPLOCK | CMPSS 锁定寄存器 |

复杂的位访问类型被编码以适合小表格单元格。表中显示了本节中用于访问类型的代码。

表 14-4 CMPSS_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | | |
| R | R | 读取 |
| R-0 | R -0 | 读取 返回 0 |
| 写入类型 | | |
| W | W | 写入 |
| W1S | W 1S | 写入 1 以置位 |
| WSonce | W Sonce | 写入 设置一次 |
| 复位或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i,j,k,l,m,n | | 当这些变量用在寄存器名称、偏移量或地址中时，它们指的是寄存器数组的值，其中寄存器是重复寄存器组的一部分。寄存器组形成一个层次结构，数组用公式表示。 |
| y | | 当此变量用在寄存器名称、偏移量或地址中时，它指的是寄存器数组的值。 |

## 14.4.2.1 COMPCTL 寄存器

偏移地址：0h 复位值：0000h

CMPSS 比较器控制寄存器

表 14-5 COMPCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| COMPDACE | ASYNCLEN | CTRIPOUTLSEL | | CTRIPLSEL | | COMPLINV | COMPLSOURCE |
| R/W-0h | R/W-0h | R/W-0h | | R/W-0h | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | ASYNCHEN | CTRIPOUTHSEL | | CTRIPHSEL | | COMPHINV | COMPHSOURCE |
| R-0h | R/W-0h | R/W-0h | | R/W-0h | | R/W-0h | R/W-0h |

表 14-6 COMPCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | COMPDACE | R/W | 0h | 比较器/DAC 使能。<br>0 比较器/DAC 禁用<br>1 比较器/DAC 使能<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 14 | ASYNCLEN | R/W | 0h | 低比较器异步路径使能。当 CTRIPLSEL=3 或 CTRIPOUTLSEL=3 时，允许异步比较器输出馈入带有锁存数字滤波器信号的或门。<br>0 异步比较器输出不馈入带有锁存数字滤波器输出的或门<br>1 异步比较器输出馈入带有锁存数字滤波器输出的或门 。<br>复位类型：SYSRSn |
| 13-12 | CTRIPOUTLSEL | R/W | 0h | 低比较器 CTRIPOUTL 源选择。<br>0 异步比较器输出驱动 CTRIPOUTL<br>1 同步比较器输出驱动 CTRIPOUTL<br>2 数字滤波器输出驱动 CTRIPOUTL<br>3 数字滤波器的锁存输出驱动 CTRIPOUTL<br>复位类型：SYSRSn |
| 11-10 | CTRIPLSEL | R/W | 0h | 低比较器 CTRIPL 源选择。<br>0 异步比较器输出驱动 CTRIPL<br>1 同步比较器输出驱动 CTRIPL<br>2 数字滤波器输出驱动 CTRIPL<br>3 数字滤波器的锁存输出驱动 CTRIPL 。复位类型：SYSRSn |
| 9 | COMPLINV | R/W | 0h | 低比较器输出反相。<br>0 比较器输出不反相<br>1 比较器输出反相。<br>复位类型：SYSRSn |
| 8 | COMPLSOURCE | R/W | 0h | 低比较器输入源。<br>0 比较器的反相输入由内部 DAC 驱动<br>1 比较器的反相输入通过外部引脚驱动。<br>复位类型：SYSRSn |
| 7 | RESERVED | R | 0h | 保留 |
| 6 | ASYNCHEN | R/W | 0h | 高比较器异步路径使能。当 CTRIPHSEL=3 或 CTRIPOUTHSEL=3 时，允许异步比较器输出馈入带有锁存数字滤波器信号的或门。<br>0 异步比较器输出不馈入带有锁存数字滤波器输出的或门<br>1 异步比较器输出馈入带有锁存数字滤波器输出的或门。<br>复位类型：SYSRSn |
| 5-4 | CTRIPOUTHSEL | R/W | 0h | 高比较器 CTRIPOUTH 源选择。<br>0 异步比较器输出驱动 CTRIPOUTH<br>1 同步比较器输出驱动 CTRIPOUTH<br>2 数字滤波器输出驱动 CTRIPOUTH<br>3 数字滤波器的锁存输出驱动 CTRIPOUTH<br>复位类型：SYSRSn |
| 3-2 | CTRIPHSEL | R/W | 0h | 高比较器 CTRIPH 源选择。<br>0 异步比较器输出驱动 CTRIPH<br>1 同步比较器输出驱动 CTRIPH |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 2 数字滤波器输出驱动 CTRIPH<br>3 数字滤波器的锁存输出驱动 CTRIPH 。<br>复位类型：SYSRSn |
| 1 | COMPHINV | R/W | 0h | 高比较器输出反相。<br>0 比较器输出不反相<br>1 比较器输出反相<br>复位类型：SYSRSn |
| 0 | COMPHSOURCE | R/W | 0h | 高比较器输入源。<br>0 比较器的反相输入由内部 DAC 驱动<br>1 比较器的反相输入通过外部引脚驱动<br>复位类型：SYSRSn |

### 14.4.2.2 COMPHYSCTL 寄存器

偏移地址：2h 复位值：0000h

CMPSS 比较器迟滞控制寄存器

表 14-7 COMPHYSCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | COMPHYS | | | |
| R-0h | | | | R/W-0h | | | |

表 14-8 COMPHYSCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R | 0h | 保留 |
| 3-0 | COMPHYS | R/W | 0h | 比较器迟滞。设置比较器输入端的迟滞量。<br>0 无<br>1 设置为典型迟滞<br>2 设置为 2 倍典型迟滞<br>3 设置为 3 倍典型迟滞<br>4 设置为 4 倍典型迟滞<br>其他：未定义<br>复位类型：SYSRSn |

### 14.4.2.3 COMPSTS 寄存器

偏移地址：4h 复位值：0000h

CMPSS 比较器状态寄存器

表 14-9 COMPSTS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | COMPLLATCH | COMPLSTS |
| R-0h | | | | | | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | COMPHLATCH | COMPHSTS |
| R-0h | | | | | | R-0h | R-0h |

表 14-10 COMPSTS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-10 | RESERVED | R | 0h | 保留 |
| 9 | COMPLLATCH | R | 0h | 低比较器数字滤波器输出的锁存值。 复位类型：SYSRSn |
| 8 | COMPLSTS | R | 0h | 低比较器数字滤波器输出。 复位类型：SYSRSn |
| 7-2 | RESERVED | R | 0h | 保留 |
| 1 | COMPHLATCH | R | 0h | 高比较器数字滤波器输出的锁存值。 复位类型：SYSRSn |
| 0 | COMPHSTS | R | 0h | 高比较器数字滤波器输出。 复位类型：SYSRSn |

### 14.4.2.4 COMPSTSCLR 寄存器

偏移地址：6h 复位值：0000h

CMPSS 比较器状态清零寄存器

表 14-11 COMPSTSCLR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | LSYNCCLREN | LLATCHCLR | 保留 |
| R-0h | | | | | R/W-0h | R-0/W1S-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | HSYNCCLREN | HLATCHCLR | 保留 |
| R-0h | | | | | R/W-0h | R-0/W1S-0h | R-0h |

表 14-12 COMPSTSCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-11 | RESERVED | R | 0h | 保留 |
| 10 | LSYNCCLREN | R/W | 0h | 低比较器锁存 EPWMSYNCPER 清零。使能 EPWMSYNCPER 复位低比较器数字滤波器输出锁存器 COMPSTS[COMPLLATCH]。<br>0 EPWMSYNCPER 不会复位锁存器 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1 EPWMSYNCPER 将复位锁存器<br>复位类型：SYSRSn |
| 9 | LLATCHCLR | R-0/W1S | 0h | 低比较器锁存软件清零。执行低比较器数字滤波器输出锁存器 COMPSTS[COMPLLATCH] 的软件复位。读取始终返回 0。<br>0 无效<br>1 为 COMPSTS[COMPLLATCH] 生成锁存复位信号的单个脉冲<br>复位类型：SYSRSn |
| 8-3 | RESERVED | R | 0h | 保留 |
| 2 | HSYNCCLREN | R/W | 0h | 高比较器锁存 EPWMSYNCPER 清零。使能 EPWMSYNCPER 复位高比较器数字滤波器输出锁存器 COMPSTS[COMPHLATCH]。<br>0 EPWMSYNCPER 不会复位锁存器<br>1 EPWMSYNCPER 将复位锁存器<br>复位类型：SYSRSn |
| 1 | HLATCHCLR | R-0/W1S | 0h | 高比较器锁存软件清零。执行高比较器数字滤波器输出锁存器 COMPSTS[COMPHLATCH] 的软件复位。读取始终返回 0。<br>0 无效<br>1 为 COMPSTS[COMPHLATCH] 生成锁存复位信号的单个脉冲<br>复位类型：SYSRSn |
| 0 | RESERVED | R | 0h | 保留 |

## 14.4.2.5 COMPDACCTL 寄存器

偏移地址：8h 复位值：0000h

CMPSS DAC 控制寄存器

表 14-13 COMPDACCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FREESOFT | | 保留 | BLANKEN | BLANKSOURCE | | | |
| R/W-0h | | R-0h | R/W-0h | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SWLOADSEL | RAMPLOADSEL | SELREF | RAMPSOURCE | | | | DACSOURCE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | R/W-0h |

表 14-14 COMPDACCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | FREESOFT | R/W | 0h | 自由运行或软件运行仿真行为。仿真挂起期间斜坡发生器的行为。<br>00b 仿真挂起期间斜坡发生器立即停止 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 01b 斜坡发生器完成当前斜坡并在仿真挂起期间的下一个 EPWMSYNCPER 处停止<br>1Xb 斜坡发生器自由运行。<br>复位类型：SYSRSn |
| 13 | RESERVED | R | 0h | 保留 |
| 12 | BLANKEN | R/W | 0h | EPWMBLANK 使能。此位使能 EPWMBLANK 信号。<br>0 EPWMBLANK 信号被禁用。<br>1 EPWMBLANK 信号被使能。<br>复位类型：SYSRSn |
| 11-8 | BLANKSOURCE | R/W | 0h | EPWMBLANK 源选择。此位字段决定哪个 EPWMnBLANK 作为 EPWMBLANK 信号传递。 其中 n 代表器件上可用的 EPWMBLANK 信号的最大数量：<br>0 EPWM1BLANK 1 EPWM2BLANK<br>2 EPWM3BLANK … n-1 EPWMnBLANK<br>复位类型：SYSRSn |
| 7 | SWLOADSEL | R/W | 0h | 软件加载选择。决定 DACxVALA 是从 DACxVALS 在 SYSCLK 上更新还是在 EPWMSYNCPER 上更新。<br>0 DACxVALA 在 SYSCLK 上从 DACxVALS 更新<br>1 DACxVALA 在 EPWMSYNCPER 上从 DACxVALS 更新<br>复位类型：SYSRSn |
| 6 | RAMPLOADSEL | R/W | 0h | 斜坡加载选择。决定当 COMPSTS[COMPHSTS] 被触发时，RAMPSTS 是从 RAMPMAXREFA 还是从 RAMPMAXREFS 更新。<br>0 RAMPSTS 从 RAMPMAXREFA 加载<br>1 RAMPSTS 从 RAMPMAXREFS 加载<br>复位类型：SYSRSn |
| 5 | SELREF | R/W | 0h | DAC 参考选择。决定哪个电压源用作内部比较器 DAC 的参考。<br>0 VDDA 是 DAC 的电压参考<br>1 VDAC 是 DAC 的电压参考<br>复位类型：SYSRSn |
| 4-1 | RAMPSOURCE | R/W | 0h | EPWMSYNCPER 源选择。决定在 CMPSS 模块内部使用哪个 EPWMnSYNCPER 信号。 其中 n 代表器件上可用的 EPWMSYNCPER 信号的最大数量：<br>0 EPWM1SYNCPER<br>1 EPWM2SYNCPER<br>2 EPWM3SYNCPER …<br>n-1 EPWMnSYNCPER 复位类型：SYSRSn |
| 0 | DACSOURCE | R/W | 0h | DAC 源选择。决定 DACHVALA 是从 DACHVALS 更新还是从斜坡发生器更新。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 0 DACHVALA 从 DACHVALS 更新<br>1 DACHVALA 从斜坡发生器更新<br>复位类型：SYSRSn |

### 14.4.2.6 DACHVALS 寄存器

偏移地址：Ch　复位值：0000h

CMPSS 高 DAC 值影子寄存器

表 14-15 DACHVALS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | DACVAL | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DACVAL | | | | | | | |
| R/W-0h | | | | | | | |

表 14-16 DACHVALS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留 |
| 11-0 | DACVAL | R/W | 0h | 高 DAC 影子值。当 COMPDACCTL[DACSOURCE]=0 时，DACHVALS 的值在由 COMPDACCTL[SWLOADSEL] 选择的触发信号上加载到 DACHVALA 中。 复位类型：SYSRSn |

### 14.4.2.7 DACHVALA 寄存器

偏移地址：Eh　复位值：0000h

CMPSS 高 DAC 值活动寄存器

表 14-17 DACHVALA 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | DACVAL | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DACVAL | | | | | | | |
| R-0h | | | | | | | |

表 14-18 DACHVALA 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留 |
| 11-0 | DACVAL | R | 0h | 高 DAC 活动值。高 DAC 主动驱动的值。 复位类型：SYSRSn |

### 14.4.2.8 RAMPMAXREFA 寄存器

偏移地址：10h 复位值：0000h

CMPSS 斜坡最大参考活动寄存器

表 14-19 RAMPMAXREFA 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RAMPMAXREF | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RAMPMAXREF | | | | | | | |
| R-0h | | | | | | | |

表 14-20 RAMPMAXREFA 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RAMPMAXREF | R | 0h | 斜坡最大参考活动值。要加载到斜坡发生器 RAMPSTS 的锁存值。 复位类型：SYSRSn |

### 14.4.2.9 RAMPMAXREFS 寄存器

偏移地址：14h 复位值：0000h

CMPSS 斜坡最大参考影子寄存器

表 14-21 RAMPMAXREFS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RAMPMAXREF | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RAMPMAXREF | | | | | | | |
| R/W-0h | | | | | | | |

表 14-22 RAMPMAXREFS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RAMPMAXREF | R/W | 0h | 斜坡最大参考影子。要加载到斜坡发生器 RAMPSTS 的未锁存值。 复位类型：SYSRSn |

### 14.4.2.10 RAMPDECVALA 寄存器

偏移地址：18h　复位值：0000h

CMPSS 斜坡递减值活动寄存器

表 14-23 RAMPDECVALA 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RAMPDECVAL | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RAMPDECVAL | | | | | | | |
| R-0h | | | | | | | |

表 14-24 RAMPDECVALA 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RAMPDECVAL | R | 0h | 斜坡递减值活动。将从 RAMPSTS 减去的锁存值。 复位类型：SYSRSn |

### 14.4.2.11 RAMPDECVALS 寄存器

偏移地址：1Ch　复位值：0000h

CMPSS 斜坡递减值影子寄存器

表 14-25 RAMPDECVALS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RAMPDECVAL | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RAMPDECVAL | | | | | | | |
| R/W-0h | | | | | | | |

表 14-26 RAMPDECVALS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RAMPDECVAL | R/W | 0h | 斜坡递减值影子。要加载到 RAMPDECVALA 的未锁存值。 复位类型：SYSRSn |

### 14.4.2.12 RAMPSTS 寄存器

偏移地址：20h　复位值：0000h

CMPSS 斜坡状态寄存器

表 14-27 RAMPSTS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RAMPVALUE | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RAMPVALUE | | | | | | | |
| R-0h | | | | | | | |

表 14-28 RAMPSTS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RAMPVALUE | R | 0h | 斜坡值。斜坡发生器的当前值。 复位类型：SYSRSn |

## 14.4.2.13 DACLVALS 寄存器

偏移地址：24h 复位值：0000h

CMPSS 低 DAC 值影子寄存器

表 14-29 DACLVALS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | DACVAL | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DACVAL | | | | | | | |
| R/W-0h | | | | | | | |

表 14-30 DACLVALS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留 |
| 11-0 | DACVAL | R/W | 0h | 低 DAC 影子值。在由 COMPDACCTL[SWLOADSEL] 选择的触发信号上加载到 DACLVALA 的值。 复位类型：SYSRSn |

## 14.4.2.14 DACLVALA 寄存器

偏移地址：26h 复位值：0000h

CMPSS 低 DAC 值活动寄存器

表 14-31 DACLVALA 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | DACVAL | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DACVAL | | | | | | | |
| R-0h | | | | | | | |

表 14-32 DACLVALA 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留 |
| 11-0 | DACVAL | R | 0h | 低 DAC 活动值。低 DAC 主动驱动的值。 复位类型：SYSRSn |

## 14.4.2.15 RAMPDLYA 寄存器

偏移地址：28h 复位值：0000h

CMPSS 斜坡延迟活动寄存器

表 14-33 RAMPDLYA 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | DELAY | | | | |
| R-0h | | | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DELAY | | | | | | | |
| R-0h | | | | | | | |

表 14-34 RAMPDLYA 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-13 | RESERVED | R | 0h | 保留 |
| 12-0 | DELAY | R | 0h | 斜坡延迟活动值。在接收到 EPWMSYNCPER 后，延迟斜坡发生器递减器启动的周期数的锁存值。 复位类型：SYSRSn |

## 14.4.2.16 RAMPDLYS 寄存器

偏移地址：2Ah 复位值：0000h

CMPSS 斜坡延迟影子寄存器

表 14-35 RAMPDLYS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | DELAY | | | | |
| R-0h | | | R/W-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DELAY | | | | | | | |
| R/W-0h | | | | | | | |

表 14-36 RAMPDLYS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-13 | RESERVED | R | 0h | 保留 |
| 12-0 | DELAY | R/W | 0h | 斜坡延迟影子值。要加载到 RAMPDLYA 的未锁存值。 复位类型：SYSRSn |

## 14.4.2.17 CTRIPLFILCTL 寄存器

偏移地址：2Ch 复位值：0000h

CTRIPL 滤波器控制寄存器

表 14-37 CTRIPLFILCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FILINIT | 保留 | THRESH | | | | | SAMPWIN |
| R-0/W1S-0h | R-0h | R/W-0h | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SAMPWIN | | | | 保留 | | | |
| R/W-0h | | | | R-0h | | | |

表 14-38 CTRIPLFILCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | FILINIT | R-0/W1S | 0h | 低滤波器初始化。<br>0 无效<br>1 将所有采样初始化为滤波器输入值<br>复位类型：SYSRSn |
| 14 | RESERVED | R | 0h | 保留 |
| 13-9 | THRESH | R/W | 0h | 低滤波器多数表决阈值。在采样窗口内必须出现至少 THRESH 个相反状态的样本，才能使输出改变状态。使用的阈值是 THRESH+1。 复位类型：SYSRSn |
| 8-4 | SAMPWIN | R/W | 0h | 低滤波器采样窗口大小。要监控的样本数为 SAMPWIN+1。 复位类型：SYSRSn |
| 3-0 | RESERVED | R | 0h | 保留 |

### 14.4.2.18 CTRIPLFILCLKCTL 寄存器

偏移地址：2Eh　复位值：0000h

CTRIPL 滤波器时钟控制寄存器

表 14-39 CTRIPLFILCLKCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | CLKPRESCALE | |
| R-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLKPRESCALE | | | | | | | |
| R/W-0h | | | | | | | |

表 14-40 CTRIPLFILCLKCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-10 | RESERVED | R | 0h | 保留 |
| 9-0 | CLKPRESCALE | R/W | 0h | 低滤波器采样时钟预分频。采样之间的系统时钟数为 CLKPRESCALE+1。 复位类型：SYSRSn |

### 14.4.2.19 CTRIPHFILCTL 寄存器

偏移地址：30h　复位值：0000h

CTRIPH 滤波器控制寄存器

表 14-41 CTRIPHFILCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FILINIT | 保留 | THRESH | | | | | SAMPWIN |
| R-0/W1S-0h | R-0h | R/W-0h | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SAMPWIN | | | | 保留 | | | |
| R/W-0h | | | | R-0h | | | |

表 14-42 CTRIPHFILCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | FILINIT | R-0/W1S | 0h | 高滤波器初始化。<br>0 无效<br>1 将所有采样初始化为滤波器输入值<br>复位类型：SYSRSn |
| 14 | RESERVED | R | 0h | 保留 |
| 13-9 | THRESH | R/W | 0h | 高滤波器多数表决阈值。在采样窗口内必须出现至少 THRESH 个相反状态的样本，才能使输出 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 改变状态。使用的阈值是 THRESH+1。 复位类型：SYSRSn |
| 8-4 | SAMPWIN | R/W | 0h | 高滤波器采样窗口大小。要监控的样本数为 SAMPWIN+1。 复位类型：SYSRSn |
| 3-0 | RESERVED | R | 0h | 保留 |

### 14.4.2.20 CTRIPHFILCLKCTL 寄存器

偏移地址：32h 复位值：0000h

CTRIPH 滤波器时钟控制寄存器

表 14-43 CTRIPHFILCLKCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | CLKPRESCALE | |
| R-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLKPRESCALE | | | | | | | |
| R/W-0h | | | | | | | |

表 14-44 CTRIPHFILCLKCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-10 | RESERVED | R | 0h | 保留 |
| 9-0 | CLKPRESCALE | R/W | 0h | 高滤波器采样时钟预分频。采样之间的系统时钟数为 CLKPRESCALE+1。 复位类型：SYSRSn |

### 14.4.2.21 COMPLOCK 寄存器

偏移地址：34h 复位值：0000h

CMPSS 锁定寄存器

表 14-45 COMPLOCK 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | 保留 | CTRIP | DACCTL | COMPHYSCTL | COMPCTL |
| R-0h | | | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h | R/WSonce-0h |

表 14-46 COMPLOCK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-5 | RESERVED | R | 0h | 保留 |
| 4 | RESERVED | R/WSonce | 0h | 保留 |
| 3 | CTRIP | R/WSonce | 0h | 锁定对 CTRIPxFILTCTL 和 CTRIPxFILCLKCTL* 寄存器的写访问。<br>0 CTRIPxFILCTL 和 CTRIPxFILCLKCTL* 寄存器未锁定。向此位写入 0 无效。<br>1 CTRIPxFILCTL 和 CTRIPxFILCLKCTL* 寄存器已锁定。只有系统复位才能清除此位。<br>复位类型：SYSRSn |
| 2 | DACCTL | R/WSonce | 0h | 锁定对 COMPDACCTL 寄存器的写访问。<br>0 COMPDACCTL 寄存器未锁定。向此位写入 0 无效。<br>1 COMPDACCTL 寄存器已锁定。只有系统复位才能清除此位。<br>复位类型：SYSRSn |
| 1 | COMPHYSCTL | R/WSonce | 0h | 锁定对 COMPHYSCTL 寄存器的写访问。<br>0 COMPHYSCTL 寄存器未锁定。向此位写入 0 无效。<br>1 COMPHYSCTL 寄存器已锁定。只有系统复位才能清除此位。<br>复位类型：SYSRSn |
| 0 | COMPCTL | R/WSonce | 0h | 锁定对 COMPCTL 寄存器的写访问。<br>0 COMPCTL 寄存器未锁定。向此位写入 0 无效。<br>1 COMPCTL 寄存器已锁定。只有系统复位才能清除此位。<br>复位类型：SYSRSn |

## 14.4.3 CMPSS 寄存器与 DriverLib 函数对应表

表 14-47 CMPSS 寄存器与 DriverLib 函数对应表

| 寄存器 | 文件 | Driverlib 函数 |
|---|---|---|
| COMPCTL | cmpss.h | CMPSS_enableModule |
| | cmpss.h | CMPSS_disableModule |
| | cmpss.h | CMPSS_configHighComparator |
| | cmpss.h | CMPSS_configLowComparator |
| | cmpss.h | CMPSS_configOutputsHigh |
| | cmpss.h | CMPSS_configOutputsLow |
| COMPHYSCTL | cmpss.h | CMPSS_setHysteresis |
| COMPSTS | cmpss.c | CMPSS_configLatchOnPWMSYNC |
| | cmpss.h | CMPSS_getStatus |
| | cmpss.h | CMPSS_clearFilterLatchHigh |
| | cmpss.h | CMPSS_clearFilterLatchLow |

| 寄存器 | 文件 | Driverlib 函数 |
|---|---|---|
| | cmpss.h | CMPSS_enableLatchResetOnPWMSYNCHigh |
| | cmpss.h | CMPSS_disableLatchResetOnPWMSYNCHigh |
| | cmpss.h | CMPSS_enableLatchResetOnPWMSYNCLow |
| | cmpss.h | CMPSS_disableLatchResetOnPWMSYNCLow |
| COMPSTSCLR | cmpss.c | CMPSS_configLatchOnPWMSYNC |
| | cmpss.h | CMPSS_clearFilterLatchHigh |
| | cmpss.h | CMPSS_clearFilterLatchLow |
| | cmpss.h | CMPSS_enableLatchResetOnPWMSYNCHigh |
| | cmpss.h | CMPSS_disableLatchResetOnPWMSYNCHigh |
| | cmpss.h | CMPSS_enableLatchResetOnPWMSYNCLow |
| | cmpss.h | CMPSS_disableLatchResetOnPWMSYNCLow |
| COMPDACCTL | cmpss.c | CMPSS_configRamp |
| | cmpss.h | CMPSS_configDAC |
| | cmpss.h | CMPSS_configureSyncSource |
| | cmpss.h | CMPSS_configBlanking |
| | cmpss.h | CMPSS_enableBlanking |
| | cmpss.h | CMPSS_disableBlanking |
| DACHVALS | cmpss.h | CMPSS_setDACValueHigh |
| DACHVALA | cmpss.h | CMPSS_getDACValueHigh |
| RAMPMAXREFA | cmpss.h | CMPSS_getMaxRampValue |
| RAMPMAXREFS | cmpss.c | CMPSS_configRamp |
| | cmpss.h | CMPSS_setMaxRampValue |
| RAMPDECVALA | cmpss.h | CMPSS_getRampDecValue |
| RAMPDECVALS | cmpss.c | CMPSS_configRamp |
| | cmpss.h | CMPSS_setRampDecValue |
| RAMPSTS | - | |
| DACLVALS | cmpss.h | CMPSS_setDACValueLow |
| DACLVALA | cmpss.h | CMPSS_getDACValueLow |
| RAMPDLYA | cmpss.h | CMPSS_getRampDelayValue |
| RAMPDLYS | cmpss.c | CMPSS_configRamp |
| | cmpss.h | CMPSS_setRampDelayValue |
| CTRIPLFILCTL | cmpss.c | CMPSS_configFilterLow |
| | cmpss.h | CMPSS_initFilterLow |
| CTRIPLFILCLKCTL | cmpss.c | CMPSS_configFilterLow |
| CTRIPHFILCTL | cmpss.c | CMPSS_configFilterHigh |
| | cmpss.h | CMPSS_initFilterHigh |
| CTRIPHFILCLKCTL | cmpss.c | CMPSS_configFilterHigh |
| COMPLOCK | - | |

# 15 增强型脉宽调制器（ePWM）

## 15.1 简介

增强型脉冲宽度调制器（ePWM）外设是控制商用和工业设备中众多电力电子系统的关键元件。这些系统包括数字电机控制、开关模式电源控制、不间断电源（UPS）以及其他形式的功率转换。ePWM 外设还能执行数模（DAC）转换功能，其中占空比等同于 DAC 模拟值，因此它有时也被称为功率 DAC。本章适用于具有新增寄存器保护功能的 4 型 ePWM。

### 15.1.1 介绍

本章包括概述以及每个子模块的相关信息：

- 时基（TB）子模块
- 计数器比较（CC）子模块
- 动作限定器（AQ）子模块
- 死区发生器（DB）子模块
- PWM 斩波器（PC）子模块
- 跳闸区（TZ）子模块
- 事件触发（ET）子模块
- 数字比较（DC）子模块

ePWM 有以下增强功能：

- 寄存器地址映射：ePWM 有一些特殊的寄存器来实现新功能。ePWM 寄存器地址空间已重新映射，以实现更好的对齐和更便捷的使用。
- 延迟跳闸功能：已添加相关更改以实现死区插入能力，从而支持例如峰值电流模式控制类应用场景所需的延迟跳闸功能。这是通过允许比较器事件作为触发事件（事件 T1 和 T2）进入动作限定器来实现的。如果使用比较器 T1/T2 事件来编辑 PWM，PWM 波形不会立即发生变化。相反，波形会与下一个 TBCLK 同步。

- 死区发生器子模块增强：DBCTL 寄存器的影子寄存器功能，可实现动态配置更改。
- 寄存器的单触发和全局加载：ePWM 允许从影子寄存器到活动寄存器的单触发和全局加载功能，以避免例如多相应用中的部分加载。ePWM 还允许对影子到活动的加载事件进行可编程预分频。ePWM 的全局加载功能可以通过消除中断并确保所有寄存器同时加载来简化 ePWM 软件。
- 跳闸区子模块增强：已添加独立标志以反映每个 TZ 源的跳闸状态。对跳闸区子模块进行了更改，以支持某些功率转换器开关技术，如谷值开关。
- 数字比较子模块增强：消隐窗口滤波器寄存器宽度已从 8 位增加到 16 位。DCCAP 功能已增强，以提供更多的可编程性。
- PWM 同步相关增强：ePWM 允许基于 CMPC 和 CMPD 事件生成 PWM SYNCOUT。这些事件也可用于 PWM 同步脉冲选择。

  ePWM 有如下功能：

- 高分辨率死区能力：在半周期时钟模式下，为死区 RED 和 FED 增加了高分辨率能力。
- 死区发生器子模块增强：ePWM 具有在任一 PWM 输出上启用 RED 和 FED 的功能。通过 14 位计数器提供更大的死区，并且死区/死区高分辨率寄存器具有影子寄存器功能。
- ePWMxB 输出上可用的高分辨率扩展：能够在 ePWMxB 输出上启用高分辨率周期和占空比控制。
- 计数器比较子模块增强：ePWM 允许通过额外的计数器比较 CMPC 和 CMPD 生成中断和 SOC 事件。
- 事件触发子模块增强：用于发出中断请求和 ADC 转换开始的预分频逻辑扩展到每 15 个事件。该子模块允许在同步事件时通过软件初始化事件计数器。

- 数字比较子模块增强：数字比较跳闸选择逻辑[DCTRIPSEL]除了能够对所有（多达 14 个[外部和内部源]）进行“或”运算以创建相应的 DCxEVTs 外，还有多达 12 个由输入 X-BAR 逻辑选择的外部跳闸源。
- 对 TBPRD 和 CMPx 寄存器的同步写入：此功能允许将任意 ePWM 模块的 TBPRD、CMPA:CMPAHR、CMPB:CMPBHR、CMPC 和 CMPD 的写入与其他任何 ePWM 模块相关联，并且如果需要，还允许所有 ePWM 模块与某个特定的 ePWM 模块相关联。
- TBPRD 和 CMP 寄存器在 SYNC 信号上的影子到有效加载：此功能支持 TBPRD 和 CMPA/B/C/D 寄存器的同步写入。

一个高效的 PWM 外设必须能够以最小的 CPU 开销或干预生成复杂的脉冲宽度波形，并且必须具有高度的可编程性和灵活性，同时易于理解和使用。本文所述的 ePWM 单元通过在每个 PWM 通道的基础上分配所有所需的定时和控制资源来满足这些要求。避免了资源的交叉耦合或共享；相反，ePWM 由较小的单通道子模块构建而成，这些子模块具有独立的资源，可以根据需要协同工作以形成一个系统。这种模块化方法形成了一种正交架构，并提供了对该外设结构更清晰的视图，有助于用户快速理解其操作。

在本文档中，信号或子模块名称中的字母 x 用于表示器件上的通用 ePWM 实例。例如，输出信号 EPWMxA 和 EPWMxB 指的是来自 ePWMx 实例的输出信号。因此，EPWM1A 和 EPWM1B 属于 ePWM1，同样，EPWM4A 和 EPWM4B 属于 ePWM4。

- 提高死区分辨率：死区时钟已得到增强，允许半周期时钟以实现分辨率翻倍。
- 增强的中断和 SOC 生成：现在可以在 TBCTR == 零和 TBCTR == 周期事件时生成中断和 ADC 转换启动信号。此功能支持双边缘 PWM 控制。此外，ADC 转换启动信号可以由数字比较子模块中定义的事件生成。
- 高分辨率周期能力：提供启用高分辨率周期的能力。

- 数字比较子模块：数字比较子模块通过为数字比较信号提供滤波、消隐和改进的跳闸功能，增强了事件触发和跳闸区子模块。此类功能对于峰值电流模式控制和支持模拟比较器至关重要。

注意：

送往 CMPSS 的同步信号名称已从 PWMSYNC 更新为 EPWMSYNCPER（SYNCPER/PWMSYNCPER/EPWMxSYNCPER），以避免与其他 EPWM 同步信号 EPWMSYNCI 和 EPWMSYNCO 混淆。

### 15.1.2 子模块概述

ePWM 模块代表一个完整的 PWM 通道，由两个 PWM 输出组成：EPWMxA 和 EPWMxB。如图所示，器件内部集成了多个 ePWM 模块实例。每个 ePWM 实例基本相同，但有一个例外。部分实例包含硬件扩展功能，可实现对 PWM 输出更精确的控制。每个 ePWM 模块都以数字编号标识，编号从 1 开始。例如，ePWM1 是系统中的第一个实例，ePWM3 是第三个实例，而 ePWMx 则表示任意实例。

ePWM 模块通过时钟同步机制链接在一起，以便在需要时作为一个整体系统运行。此外，该同步机制还可扩展至捕获外设子模块（eCAP）。子模块的数量取决于器件，并基于目标应用需求而定。子模块也可独立运行。

每个 ePWM 模块都与图中所示的输入/输出信号相连。这些信号将在后续章节中详细描述。

ePWM 模块的连接顺序可能与图中所示的不同。每个 ePWM 模块由八个子模块组成，并通过图中所示的信号在系统内部相连。

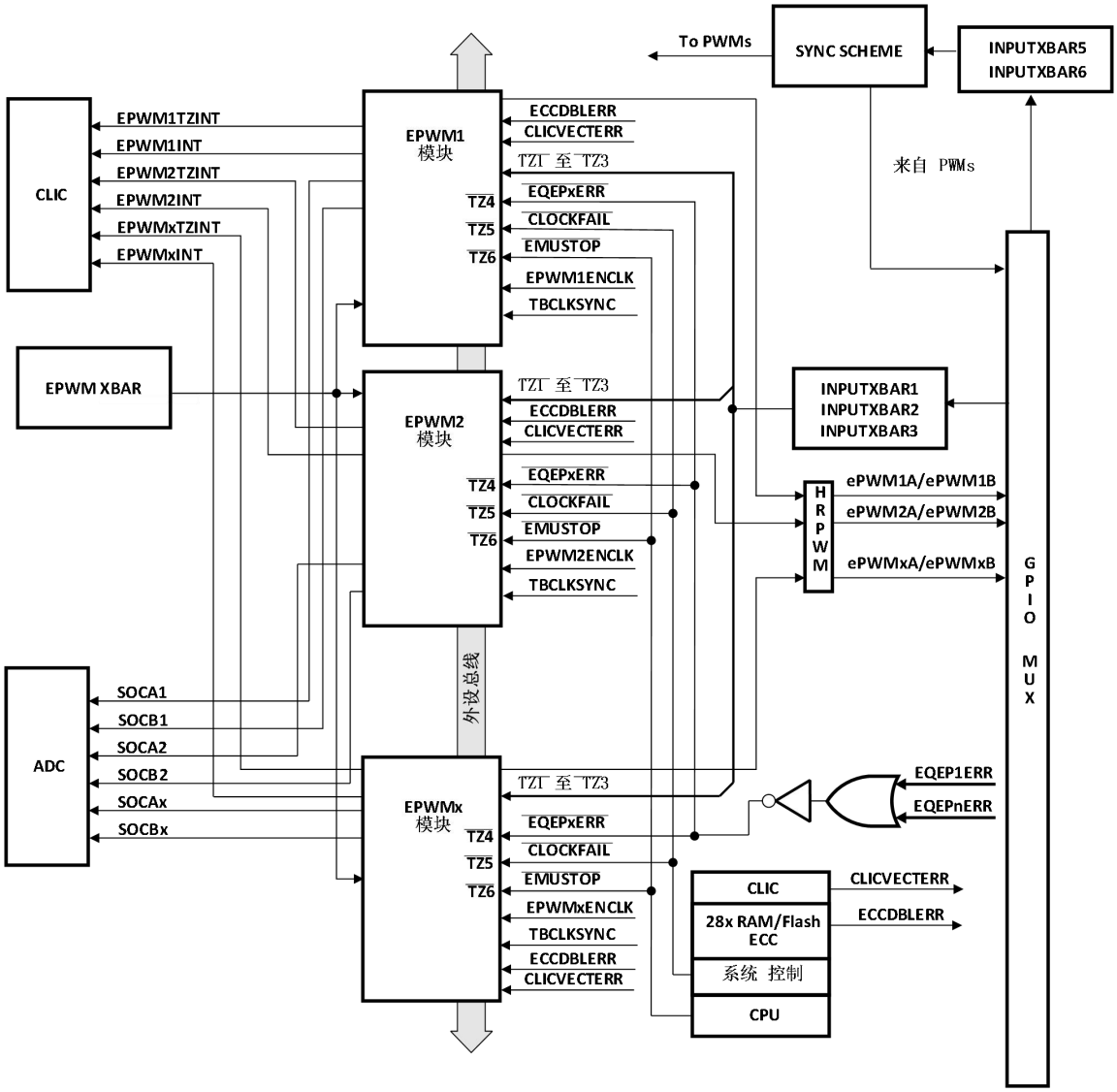



图 15-1 多个 ePWM 模块

A. 此信号仅存在于带有 eQEP 子模块的设备上。

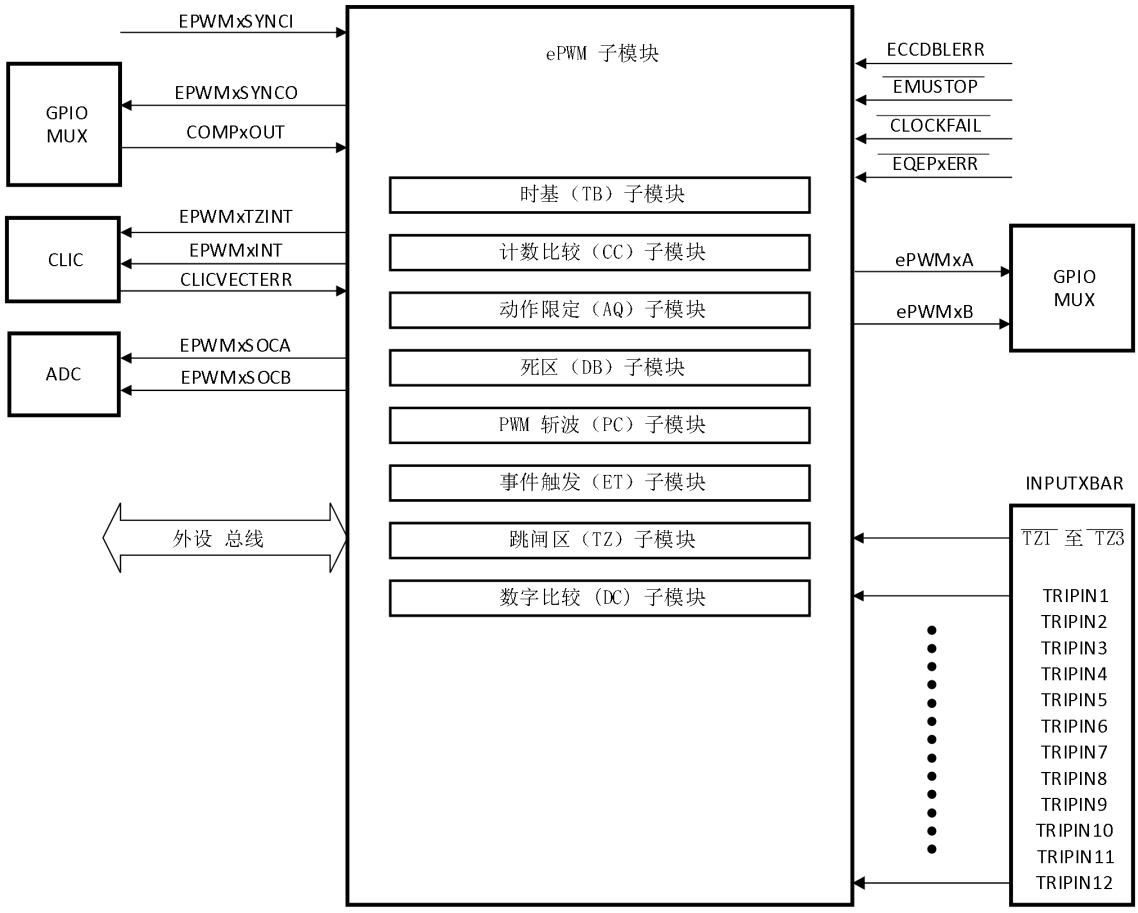



图 15-2 ePWM 模块的子模块和信号连接

图中展示了单个 ePWM 模块更多的内部细节。ePWM 模块使用的主要信号如下：

- PWM 输出信号（EPWMxA 和 EPWMxB）

  PWM 输出信号可在器件外部使用。

- 跳闸区信号（TZ1 至 TZ6）

这些输入信号用于向 ePWM 模块发出其外部故障状态的警报。器件上的每个子模块都可配置为使用或忽略任何跳闸区信号。通过 GPIO 外设，利用输入 X-BAR 逻辑，可将 TZ1 至 TZ3 跳闸区信号配置为异步输入，详情参见图。

TZ4 与反相的 EQEPx 错误信号（EQEPxERR）相连，该信号可由任意一个 EQEP 子模块（对于配备 EQEP 模块的器件）生成。

TZ5 与系统时钟故障逻辑相连，TZ6 与 CPU 的 EMUSTOP 输出相连。这使得在时钟故障或 CPU 停机时能够配置跳闸动作。

- 时基同步输入（EPWMxSYNCI）、输出（EPWMxSYNCO）和外设（EPWMxSYNCPER）信号

每个 ePWM 模块都可通过 EPWMSYNCINSEL 与其他 ePWM 模块或其他外设同步。每个 ePWM 模块还可生成同步输出信号。EPWMxSYNCOUT 的来源可通过 EPWMSYNCOUTEN 和 TBCTL2.OSHTSYNCMODE 进行选择和使能。

每个 ePWM 模块还会生成另一个名为 EPWMxSYNCPER 的 PWMSYNC 信号。EPWMxSYNCPER 输送至 CMPSS 以实现同步。其功能通过 HRPCTL 寄存器进行配置，但与 HRPWM 无关。

- ADC 转换开始信号（EPWMxSOCA 和 EPWMxSOCB）

每个 ePWM 模块都有两个 ADC 转换开始信号。任何 ePWM 模块都可触发转换开始。触发转换开始的事件在 ePWM 的事件触发子模块中进行配置。

- 比较器输出信号（COMPxOUT）

比较器模块的输出信号可通过输入 X-BAR 和 EPWM X-BAR 输送至 12 个跳闸输入[TRIPIN1 - TRIPIN12]中的一个或全部，并与跳闸区信号一起生成数字比较事件。

- 外设总线

外设总线为 32 位宽，允许对 ePWM 寄存器文件进行 16 位和 32 位写入操作。

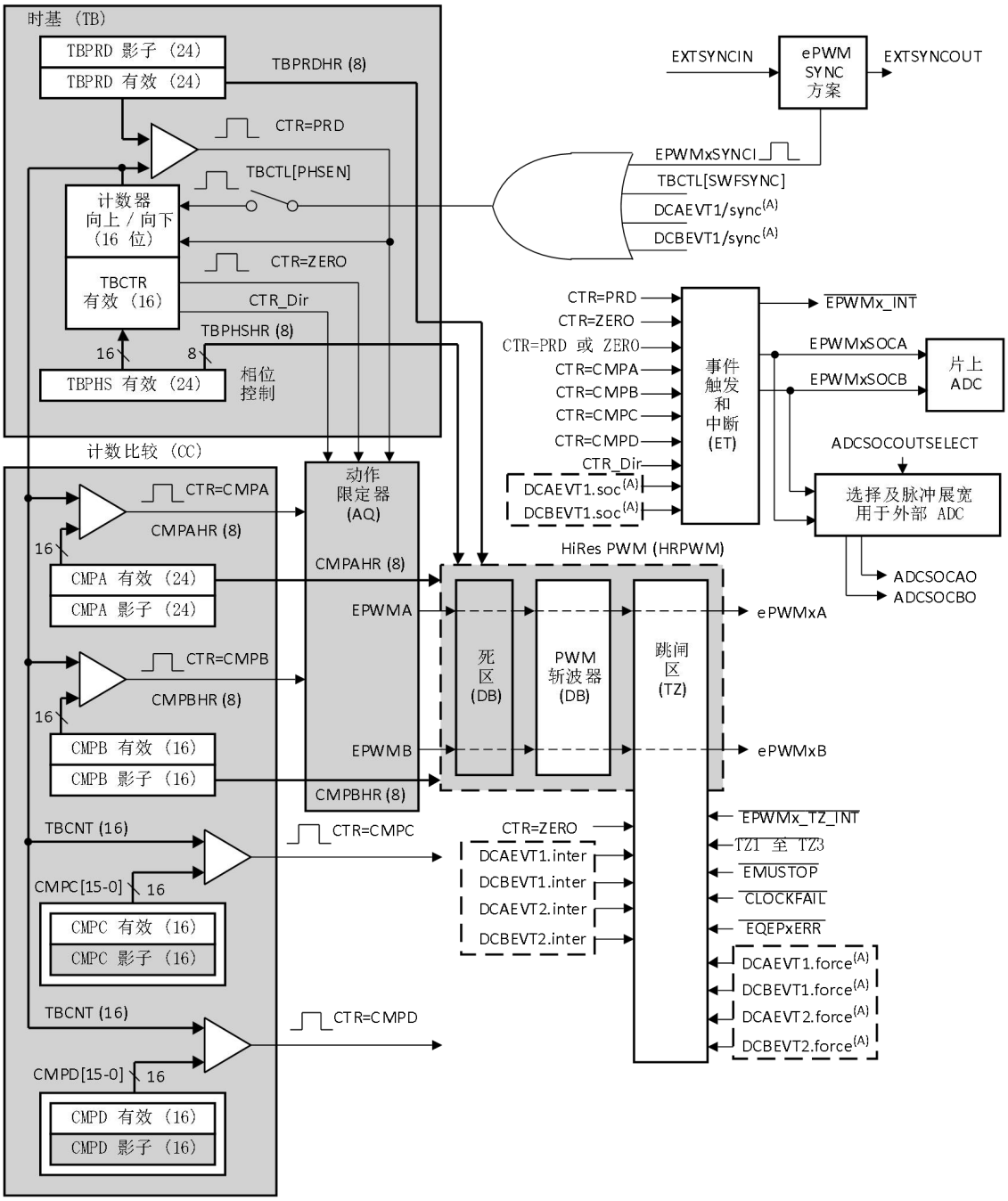



图 15-3 ePWM 模块与关键内部信号互连

A. 这些事件由 ePWM 数字比较（DC）子模块根据 TRIPIN 输入的电平生成。

### 15.1.3 配置设备引脚

要将设备输入引脚连接到模块，必须使用输入 X-BAR 和 EPWM X-BAR。某些需要外部信号的例子包括 TZx、TRIPx 和 EXTSYNCIN。设备上的任何 GPIO 都可配置

为输入。通过将相应的 GPxQSEL 寄存器位设置为 11b，可将 GPIO 输入限定设置为异步模式。内部上拉可在 GPyPUD 寄存器中配置。由于使用了 GPIO 模式，GPyINV 寄存器可对信号进行反相。此外，除了输入 X-BAR 外，一些 TRIPx（TRIP4-12，不包括 TRIP6）信号还必须通过 ePWM X-Bar 进行路由。

必须为此外设配置 GPIO 多路复用寄存器。为避免引脚上出现毛刺，必须先配置 GPyGMUX 位（同时将相应的 GPyMUX 位保持为默认的零），然后将 GPyMUX 寄存器写入所需值。

## 15.2 主要特性

每个 ePWM 模块支持以下功能：

- 专用 16 位时基计数器，具有周期和频率控制功能
- 两个 PWM 输出（EPWMxA 和 EPWMxB），可用于以下配置：
- 两个独立的 PWM 输出，采用单沿操作
- 两个独立的 PWM 输出，采用双沿对称操作
- 一个独立的 PWM 输出，采用双沿非对称操作
- 通过软件对 PWM 信号进行异步覆盖控制。
- 可编程相位控制支持，可相对于其他 ePWM 模块进行滞后或超前操作。
- 逐周期的硬件锁定（同步）相位关系。
- 死区生成，具有独立的上升沿和下降沿延迟控制。
- 在故障情况下，可编程跳闸区域分配，包括逐周期跳闸和单脉冲跳闸。
- 跳闸条件可强制 PWM 输出处于高电平、低电平或高阻抗状态逻辑电平。
- 所有事件都可以触发 CPU 中断和 ADC 转换启动（SOC）
- 可编程事件预分频可最大限度地减少中断时的 CPU 开销。
- 通过高频载波信号进行 PWM 斩波，适用于脉冲变压器栅极驱动。

## 15.3 功能描述

每个 ePWM 外设都包含八个子模块。这些子模块中的每一个都执行特定任务，且可通过软件进行配置。

表中列出了主要子模块以及主要配置参数列表。

表 15-1 子模块配置参数

| 子模块 | 配置参数或选项 |
| --- | --- |
| 时基模块（TB） | ·相对于 ePWM 时钟（EPWMCLK）缩放时基时钟（TBCLK）。<br>·配置 PWM 时基计数器（TBCTR）的频率或周期。<br>·设置时基计数器工作模式：<br>– 递增计数模式：用于非对称 PWM<br>– 递减计数模式：用于非对称 PWM<br>– 递增-递减计数模式：用于对称 PWM<br>·配置相对于其他 ePWM 模块的时基相位。<br>·通过硬件或软件实现模块间时基计数器同步。<br>·配置同步事件后时基计数器的计数方向（递增 / 递减）。<br>·根据 EPWMXLINK 配置，对所有 PWM 对应的 TBPRD 寄存器进行同步写入。<br>·配置设备被仿真器暂停时时基计数器的工作状态。<br>·指定 ePWM 模块同步输出信号的来源。<br>·配置本模块寄存器的单触发加载与全局加载。 |
| 计数器比较模块（CC） | ·为 EPWMxA、EPWMxB 输出配置 PWM 占空比。<br>·指定 EPWMxA 或 EPWMxB 输出上开关事件的触发时刻。<br>·通过附加比较器，为中断与 SOC 生成配置可编程延时。<br>·根据 EPWMXLINK 配置，对所有 PWM 对应的 CMPA、CMPB、CMPC、CMPD 寄存器进行同步写入。<br>·配置本模块寄存器的单触发加载与全局加载。 |
| 动作限定器（AQ） | ·当时基计数器比较、跳闸区子模块或比较器事件发生时，指定执行的动作类型：<br>– 不执行动作<br>– EPWMxA、EPWMxB 输出置高<br>– EPWMxA、EPWMxB 输出置低<br>– EPWMxA、EPWMxB 输出翻转<br>·通过软件控制强制 PWM 输出状态<br>·通过软件配置与控制 PWM 死区<br>·配置本模块寄存器的单触发加载与全局加载 |
| 死区发生器（DB） | ·控制上下管开关之间传统的互补死区关系<br>·指定输出上升沿延迟值<br>·指定输出下降沿延迟值<br>·完全旁路死区模块，此时 PWM 波形直接通过不做修改<br>·可选择启用半周期时钟以获得双倍分辨率<br>·允许 ePWMxB 相对于 ePWMxA 输出进行相移<br>·配置本模块寄存器的单触发加载与全局加载 |

| 子模块 | 配置参数或选项 |
| --- | --- |
| PWM 斩波器（PC） | ·生成斩波（载波）频率<br>·设置斩波脉冲串中首个脉冲的脉宽<br>·设置第二个及后续脉冲的占空比<br>·完全旁路 PWM 斩波器模块，此时 PWM 波形直接通过不做修改 |
| 跳闸区（TZ） | ·配置 ePWM 模块对部分、全部或不响应任何跳闸区信号 / 数字比较事件<br>·故障发生时指定跳闸动作：<br>– 强制 EPWMxA、EPWMxB 置高<br>– 强制 EPWMxA、EPWMxB 置低<br>– 强制 EPWMxA、EPWMxB 为高阻态<br>– 配置 EPWMxA、EPWMxB 忽略所有跳闸条件<br>·配置 ePWM 对各跳闸区信号的响应方式：<br>– 单触发<br>– 逐周期<br>·使能跳闸区触发中断功能<br>·完全旁路跳闸区模块<br>·逐周期跳闸清除的可编程选项<br>·可独立配置时基计数器递减计数时的跳闸动作 |
| 事件触发器（ET） | ·使能能触发中断的 ePWM 事件<br>·使能能触发 ADC 开始转换的 ePWM 事件<br>·指定事件触发速率（每次事件触发、每 2 次直至每 15 次触发一次）<br>·轮询、置位或清除事件标志位 |
| 数字比较器（DC） | ·使能比较器（COMP）模块输出与跳闸区信号，通过输入交叉开关（Input X-BAR）配置以生成事件及滤波事件<br>·指定事件滤波选项，用于捕获 TBCTR 计数值、生成消隐窗口，或根据捕获值在 PWM 输出 / 时基计数器中插入延迟 |

## 15.4 时基（TB）子模块

每个 ePWM 模块都有一个时基子模块，用于确定该 ePWM 模块的所有事件时序。内置的同步逻辑使多个 ePWM 模块的时基能够作为一个整体系统协同工作。

图中展示了 ePWM 内部的时基子模块。

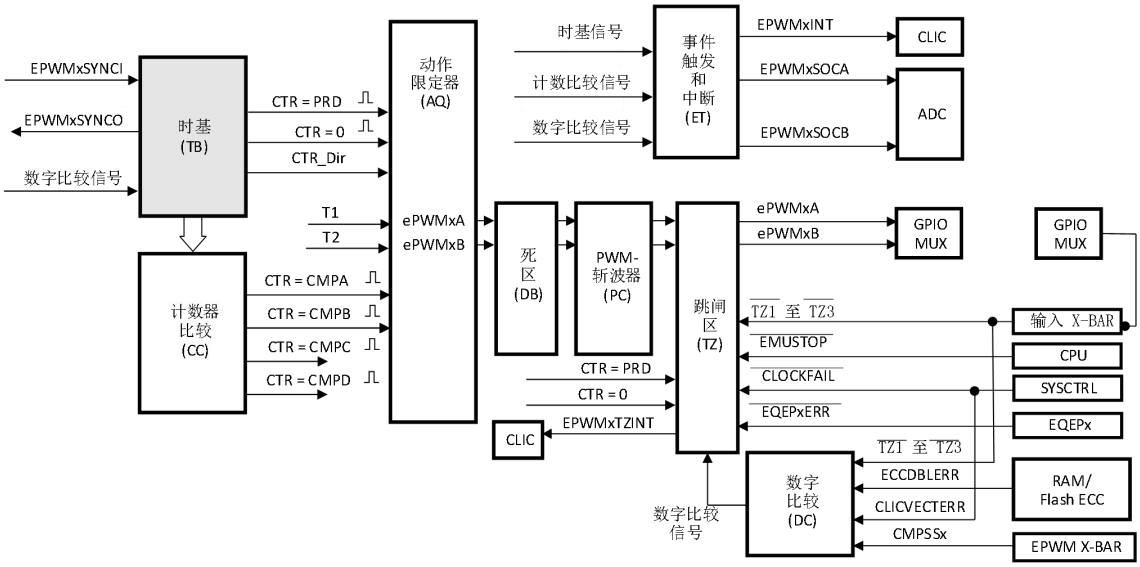



图 15-4 时基子模块

## 15.4.1 时基子模块的用途

时基子模块可配置为以下功能：

- 指定增强型脉冲宽度调制（ePWM）时基计数器（TBCTR）的频率或周期，以控制事件发生的频率。
- 管理与其他 ePWM 模块的时基同步。
- 维持与其他 ePWM 模块的相位关系。
- 将时基计数器设置为向上计数、向下计数或向上-向下计数模式。
- 生成以下事件：
- CTR = PRD：时基计数器等于指定周期（TBCTR = TBPRD）。
- CTR = Zero：时基计数器等于零（TBCTR = 0x00）。

时基

- 配置时基时钟的频率；时基时钟是增强型脉冲宽度调制时钟（EPWMCLK）的分频版本。这使得时基计数器能够以较慢的速率递增/递减。

注意：

如果应用程序代码要求在时基计数器（TBCTR）计数时通过软件更新其值，请注意，时基模块至少需要 1 个时基时钟（TBCLK）周期才能使与时间相关的事件生

效。因此，为使事件生效，在计数器向上计数的情况下，时基计数器（TBCTR）可写入 TBCTR = PRD-1 而非 TBCTR = PRD；在计数器向下计数的情况下，可写入 TBCTR = 1 而非 TBCTR = 0。

## 15.4.2 控制和监控时基子模块

图中的框图展示了时基子模块的关键信号和寄存器。

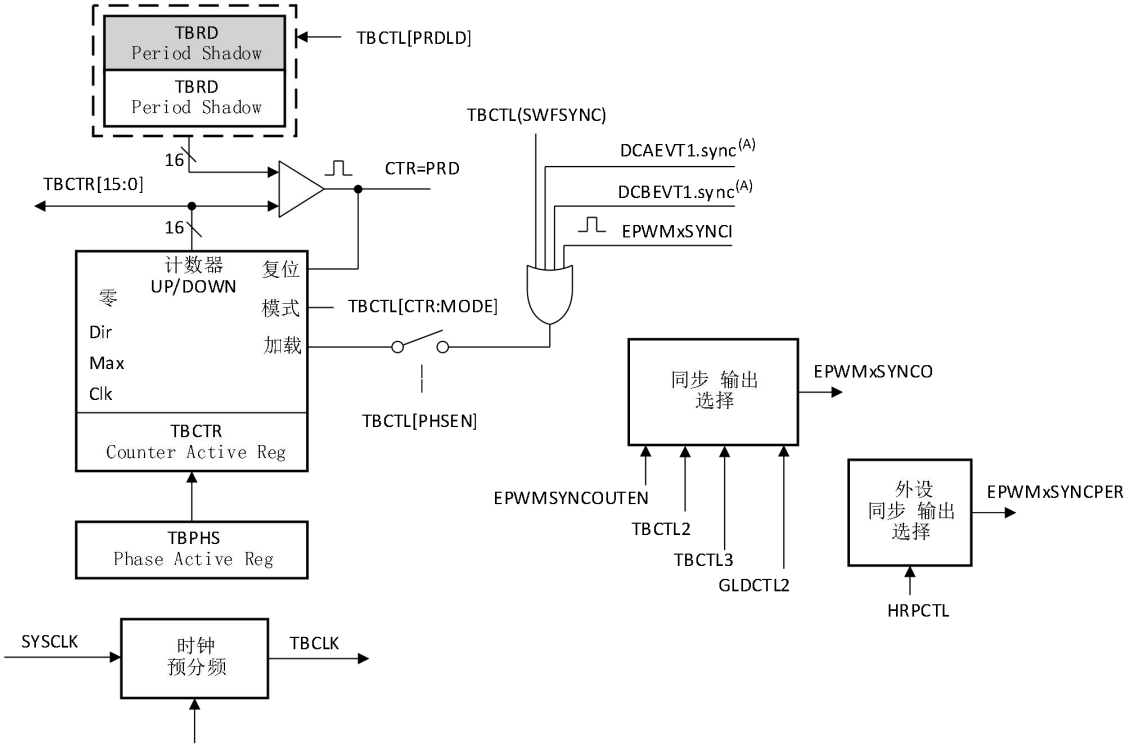



图 15-5 时基子模块的信号和寄存器

这些信号由数字比较器（DC）子模块产生

表 15-2 关键时基信号

| 信号 | 说明 |
|---|---|
| EPWMxSYNCI | 时基同步输入。<br>用于将时基计数器与其他 ePWM 模块的计数器同步的输入脉冲。 |
| EPWMxSYNCO | 时基同步输出。<br>该输出脉冲用于同步其他 ePWM 模块的计数器。可通过 EPWMSYNCOUTEN、TBCTL2、TBCTL3 和 GLDCTL2 选择输出脉冲的来源。 |
| EPWMxSYNCPER | 时基外设同步输出。<br>该输出信号用于将 CMPSS 与 EPWM 同步。可通过 HRPCTL 寄存器配置该输出信号。注意：该信号与 HRPWM 无关。 |
| CTR = PRD | 时基计数器等于设定周期。 |

| 信号 | 说明 |
| --- | --- |
| | 每当计数器值等于有效周期寄存器值时产生该信号，即 TBCTR = TBPRD。 |
| CTR = Zero | 时基计数器等于零。<br>每当计数器值为零时产生该信号，即 TBCTR 等于 0x00。 |
| CTR = CMPB | 时基计数器等于有效比较 B 寄存器（TBCTR = CMPB）。<br>该事件由计数器比较子模块产生，并用于同步输出逻辑。 |
| CTR_dir | 时基计数器方向。<br>指示 ePWM 时基计数器当前方向。计数器递增时该信号为高，递减时为低。 |
| CTR_max | 时基计数器等于最大值（TBCTR = 0xFFFF）。<br>当 TBCTR 达到最大值时产生的事件，该信号仅用作状态位。 |
| TBCLK | 时基时钟。<br>是 ePWM 时钟（EPWMCLK）经预分频后的时钟，供 ePWM 内部所有子模块使用。该时钟决定时基计数器递增或递减的速率。 |

### 15.4.3 计算 PWM 周期和频率

PWM 事件的频率由时基周期（TBPRD）寄存器和时基计数器的模式控制。图中展示了当周期设置为 4（TBPRD=4）时，上计数、下计数和上下计数时基计数器模式下的周期（Tpwm）和频率（Fpwm）关系。每一步的时间增量由时基时钟（TBCLK）定义，该时钟是增强型 PWM 时钟（EPWMCLK）的分频版本。

时基计数器有三种工作模式，可通过时基控制寄存器（TBCTL）进行选择：

- 上下计数模式：在上下计数模式下，时基计数器从 0 开始递增，直到达到周期（TBPRD）值。当达到周期值时，时基计数器随后递减，直到计数器回到 0。此时，计数器重复该模式并开始递增。
- 上计数模式：在上计数模式下，时基计数器从 0 开始递增，直到计数器达到周期寄存器（TBPRD）中的值。当达到周期值时，时基计数器重置为 0 并再次开始递增。
- 下计数模式：在下计数模式下，时基计数器从周期（TBPRD）值开始递减，直到计数器达到 0。当计数器达到 0 时，时基计数器重置为周期值并再次开始递减。

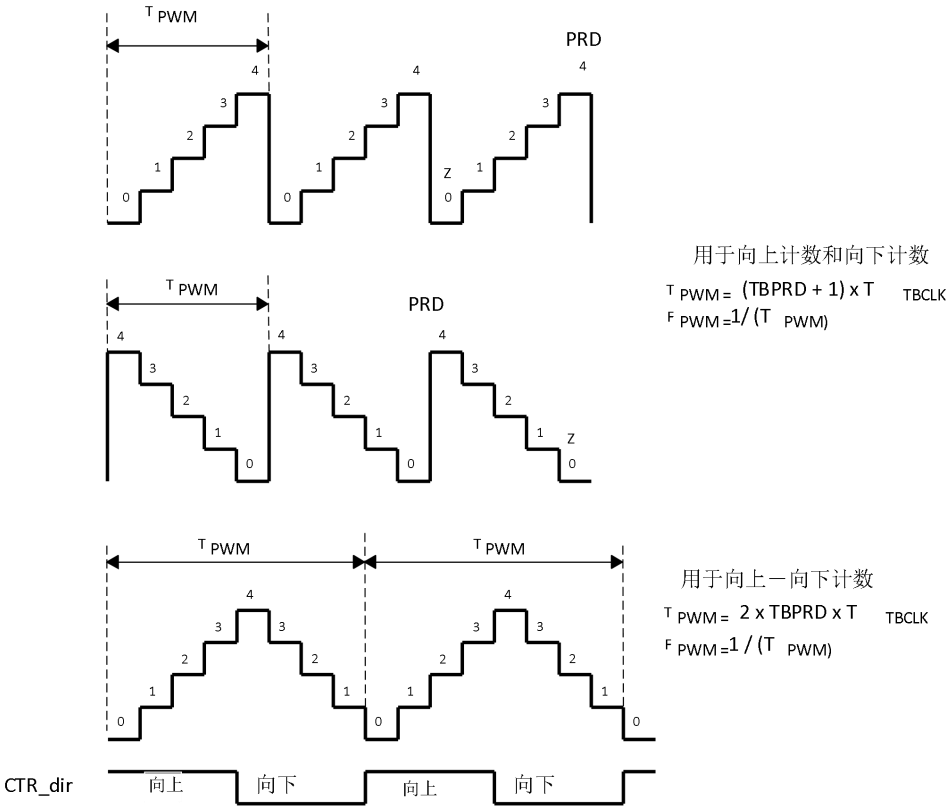



图 15-6 时基频率与周期

### 15.4.3.1 时基周期影子寄存器

时基周期寄存器（TBPRD）有一个影子寄存器。影子寄存器功能使寄存器的更新能够与硬件同步。以下定义用于描述 ePWM 模块中的所有影子寄存器：

- 活动寄存器：活动寄存器控制硬件，并负责硬件引发或调用的操作。
- 影子寄存器：影子寄存器缓冲区为活动寄存器提供临时存放位置，对任何控制硬件没有直接影响。在特定的时间点，影子寄存器的内容会传输到活动寄存器。这可防止因软件异步修改寄存器而导致的损坏或误操作。

影子周期寄存器的内存地址与活动寄存器相同。读写哪个寄存器由 TBCTL[PRDLD]位决定。该位对 TBPRD 影子寄存器的启用和禁用如下：

- 时基周期影子模式：当 TBCTL[PRDLD] = 0 时，TBPRD 影子寄存器启用。对 TBPRD 内存地址的读写操作均指向影子寄存器。当其时基计数器等于零（TBCTR = 0x00）和/或出现由 TBCTL2[PRDLDSYNC]位确定的同步事件时，影

子寄存器的内容会传输到活动寄存器（TBPRD（活动）← TBPRD（影子））。仅当 TBCTL[PRDLD] = 0 时，PRDLDSYNC 位才有效。默认情况下，TBPRD 影子寄存器是启用的。

通过配置全局加载配置寄存器（GLDCFG）中的相应位，全局加载控制机制也可与时基周期寄存器一起使用。当选择全局加载模式时，对于所有启用了该模式的寄存器，其影子寄存器到活动寄存器的内容传输会在由全局影子到活动加载控制寄存器（GLDCTL）中的配置位所定义的同一事件发生。

- 时基周期立即加载模式：如果选择了立即加载模式（TBCTL[PRDLD] = 1），那么对 TBPRD 内存地址的读写操作会直接作用于活动寄存器。

#### 15.4.3.2 时基时钟同步

外设时钟使能寄存器中的 TBCLKSYNC 位允许所有用户将所有已启用的 ePWM 模块与时基时钟（TBCLK）进行全局同步。当该位被设置时，所有已启用的 ePWM 模块时钟会随着 TBCLK 的第一个上升沿对齐而启动。对于同步的 TBCLK，每个 ePWM 模块的预分频器必须设置为相同的值。

启用 ePWM 时钟的正确步骤如下：

1. 在 PCLKCRx 寄存器中启用 ePWM 模块时钟
2. 设置 TBCLKSYNC = 0
3. 配置 ePWM 模块
4. 设置 TBCLKSYNC = 1

#### 15.4.3.3 时基计数器同步

增强型脉冲宽度调制（ePWM）同步方案提高了 ePWM 模块同步的灵活性。

每个 ePWM 模块都有一个同步输入（SYNCI）、一个同步输出（SYNCO）和一个

外设同步输出（SYNCPER）。在图中，EXTSYNCIN1 源自 INPUTXBAR5，

EXTSYNCIN2 源自 INPUTXBAR6，可通过配置选择任何通用输入/输出（GPIO）作为同步输入。

图中显示了可用于 EXTSYNCOUT 的源。

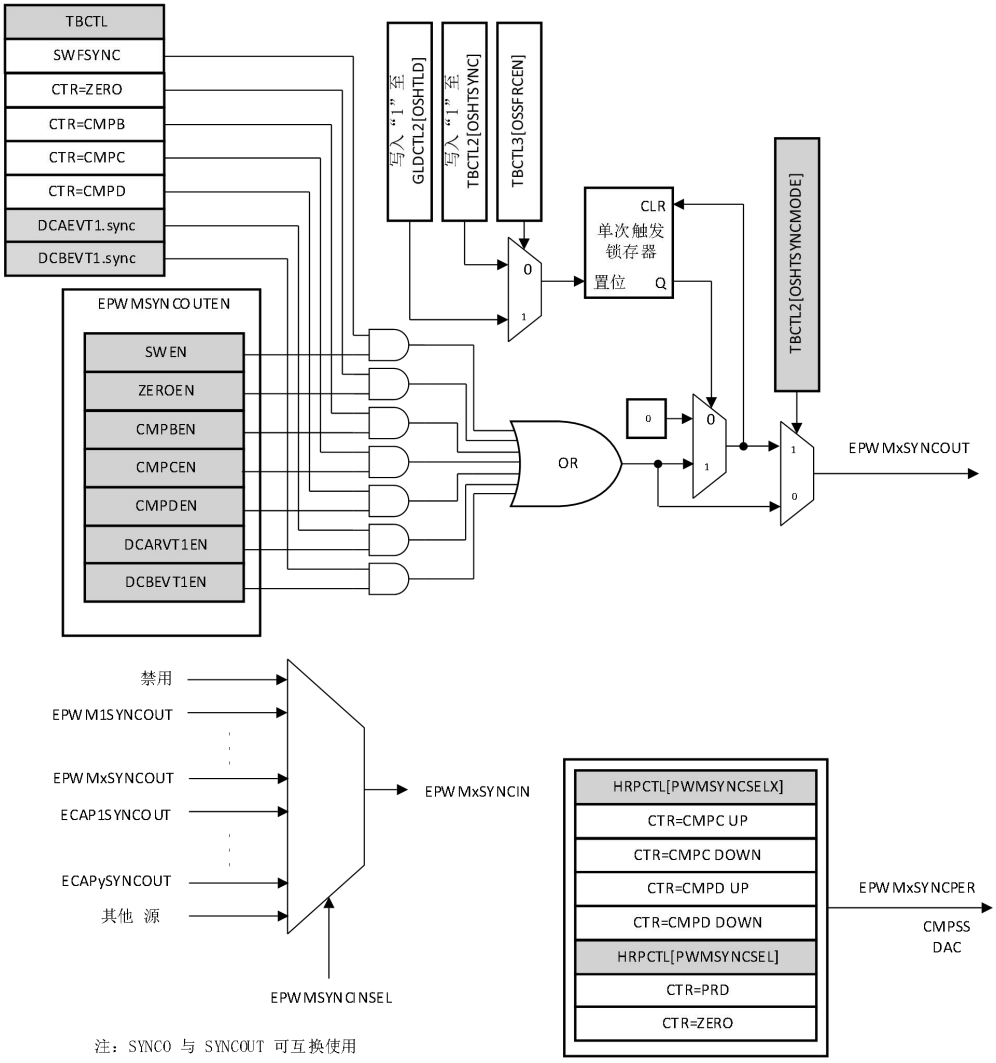



图 15-7 时基计数器同步方案

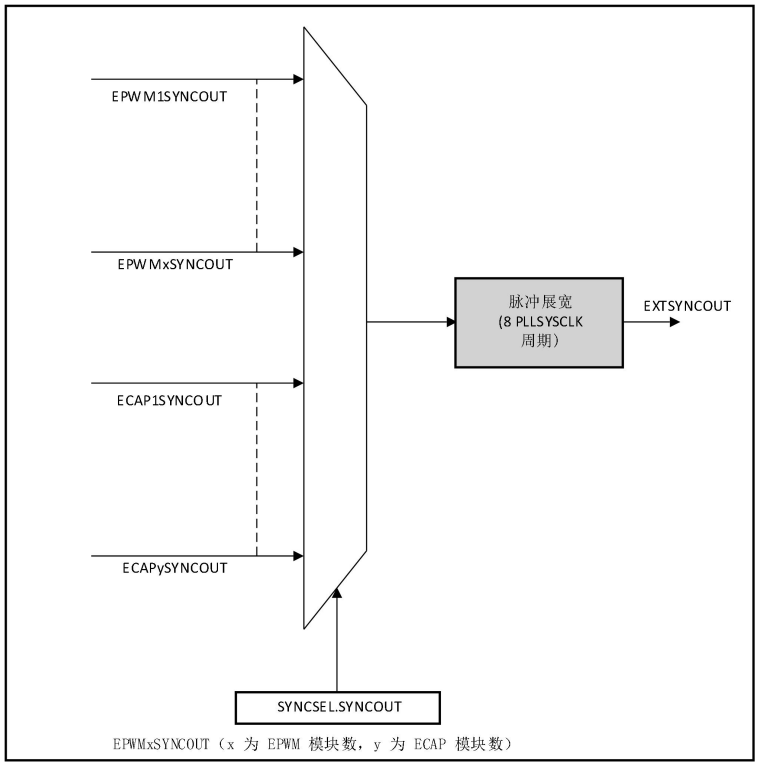



图 15-8 ePWM 外部同步输出

每个 ePWM 模块都可配置为使用或忽略同步输入。如果 TBCTL[PHSEN]位被置位，那么当出现以下任一情况时，ePWM 模块的时基计数器（TBCTR）会自动加载相位寄存器（TBPHS）的内容：

- EPWMxSYNCI：同步输入脉冲：当检测到输入同步脉冲时，相位寄存器的值会加载到计数器寄存器中（TBPHS→TBCTR）。此操作在下一个有效的时基时钟（TBCLK）边沿发生。

  从内部控制模块到目标模块的延迟如下：

- 若（TBCLK = EPWMCLK）：2×EPWMCLK
- 若（TBCLK < EPWMCLK）：1×TBCLK

  脉冲展宽（8 个 PLLSYSCLK）

- 软件强制同步脉冲：向 TBCTL[SWFSYNC]控制位写入 1 会触发软件强制同步。该脉冲与同步输入信号进行“或”运算，因此与 EPWMxSYNCI 上的脉冲具有相同的效果。

- 数字比较事件同步脉冲：DCAEVT1 和 DCBEVT1 数字比较事件可配置为生成同步脉冲，其效果与 EPWMxSYNCI 相同。

注意:

如果 EPWMxSYNCI 信号保持高电平，同步不会持续发生。EPWMxSYNCI 是上升沿触发的。

该功能可使 ePWM 模块自动同步到另一个 ePWM 模块的时基。可以对不同 ePWM 模块产生的波形添加超前或滞后相位控制，以实现它们之间的同步。

在增减计数模式下，TBCTL [PHSDIR] 位用于配置同步事件发生后时基计数器的立即方向。新方向与同步事件发生前的方向无关。PHSDIR 位在递增计数或递减计数模式下被忽略。示例见图。

将 TBCTL [PHSEN] 位清零可配置 ePWM 忽略同步输入脉冲。

### 15.4.3.4 ePWM 同步源选择

表中给出了 ePWM 同步输入和输出的信号源。

表 15-3 ePWM 同步源选择

| EPWMSYNCINSEL.SEL, ECAPSYNCINSEL.SEL | 同步源 |
|---|---|
| 0 | EPWM_SYNC_DISABLE |
| 1 | EPWM1_SYNCOUT |
| 2 | EPWM2_SYNCOUT |
| 3 | EPWM3_SYNCOUT |
| 4 | EPWM4_SYNCOUT |
| 5 | EPWM5_SYNCOUT |
| 6 | EPWM6_SYNCOUT |
| 7 | EPWM7_SYNCOUT |
| 8-16 | 保留 |
| 17 | ECAP1_SYNCOUT |
| 18 | ECAP2_SYNCOUT |
| 19 | ECAP3_SYNCOUT |
| 20-23 | 保留 |
| 24 | INPUTXBAR5 |
| 25 | INPUTXBAR6 |

### 15.4.4 多 ePWM 模块时基时钟的相位锁定

TBCLKSYNC 位可用于全局同步器件上所有使能的 ePWM 模块的时基时钟。该位是器件时钟使能寄存器的一部分，本手册的相关章节对其进行了描述。当 TBCLKSYNC=0 时，所有 ePWM 模块的时基时钟都会停止（默认状态）。当 TBCLKSYNC = 1 时，所有 ePWM 时基时钟都会随着 TBCLK 的上升沿启动并对齐。为实现完美同步的 TBCLK，每个 ePWM 模块的 TBCTL 寄存器中的预分频器位必须设置为相同值。启用 ePWM 时钟的正确步骤如下：

启用各个 ePWM 模块时钟（相关描述见对应章节）。

设置 TBCLKSYNC=0，这将停止所有已启用 ePWM 模块内的时基时钟。

配置预分频器值和所需的 ePWM 模式。

设置 TBCLKSYNC=1。

### 15.4.5 ePWM 模块间 TBPRD 和 CMPx 寄存器的同步写入

在变频应用中，需要在 ePWM 模块间同步写入 TBPRD 和 CMPx 寄存器。这可以避免出现以下情况：CTR=0 或 CTR = PRD 脉冲导致这些寄存器在 ePWM 模块间所有寄存器更新完成之前就从影子寄存器加载到活动寄存器（导致部分寄存器加载新的影子值，而其他寄存器加载旧的影子值）。为支持这一功能，新增了 ePWM 寄存器链接方案，适用于 PWM 模块间的 TBPRD:TBPRDHR、CMPA:CMPAHR、CMPB:CMPBHR、CMPC 和 CMPD 寄存器。

有关对应 ePWM 的链接寄存器位域值，请参阅 EPWMXLINK 的寄存器描述。使用 EPWMXLINK 的一个示例是，通过 ePWM2 的 EPWMXLINK [CMPALINK] 寄存器位域，将 ePWM2 的 CMPA 与 ePWM1 的 CMPA 进行链接。在这种情况下，写入 ePWM1 的 CMPA 也会更改 ePWM2 的 CMPA 值。

### 15.4.6 时基计数器模式和时序波形

时基计数器工作在以下四种模式之一：

非对称向上计数模式

非对称向下计数模式

对称向上 - 向下计数模式

冻结模式（时基计数器保持当前值不变）

为说明前三种模式的工作原理，以下时序图展示了事件的生成时机以及时基对 EPWMxSYNCI 信号的响应。

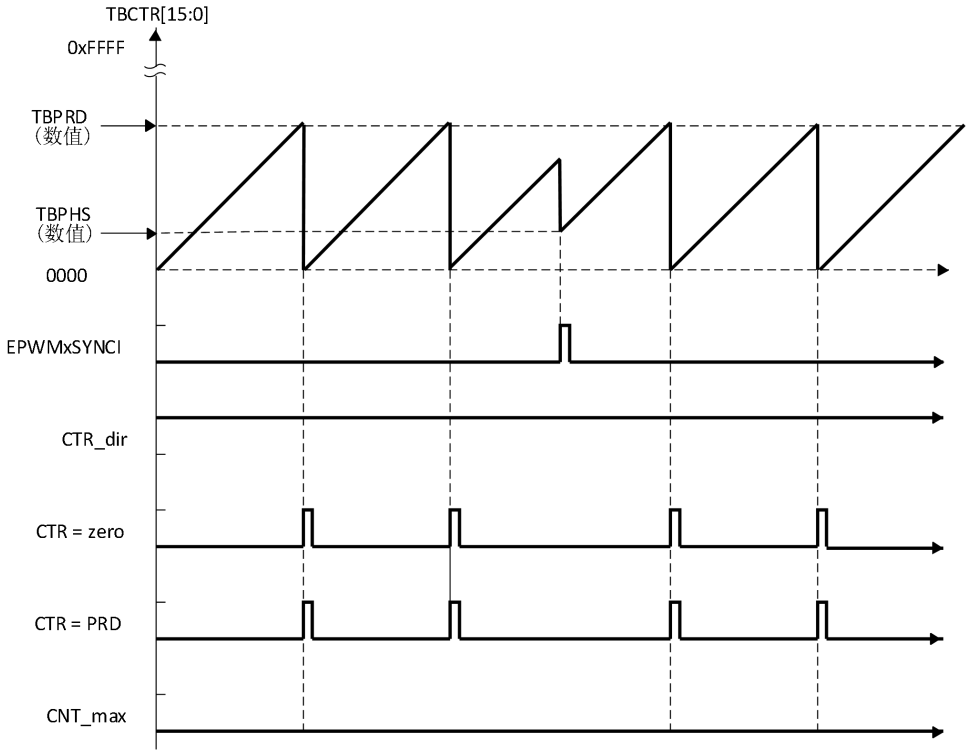



图 15-9 时基向上计数模式波形

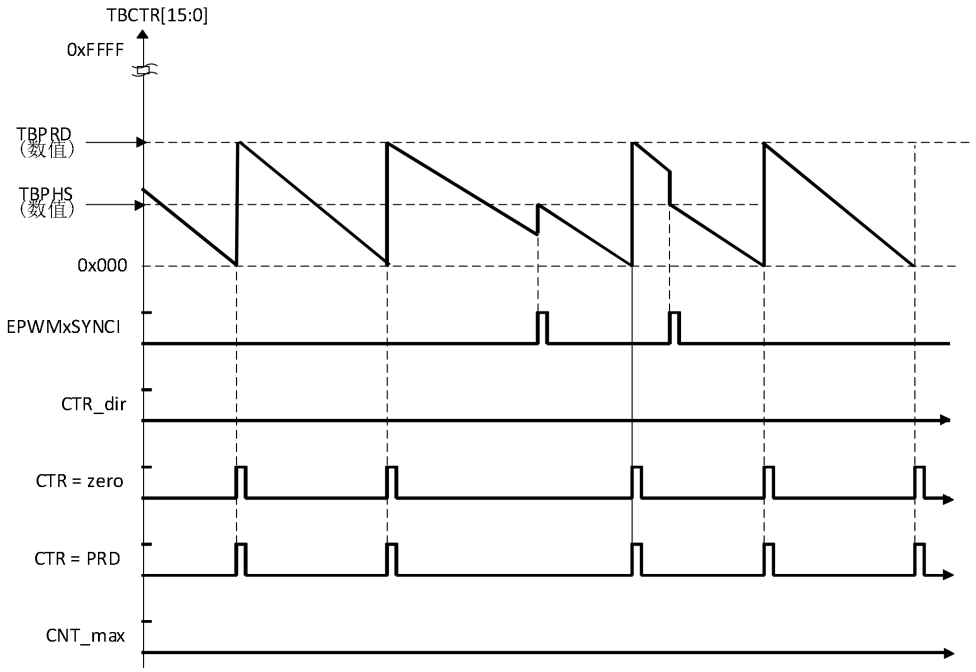



图 15-10 时基向下计数模式波形

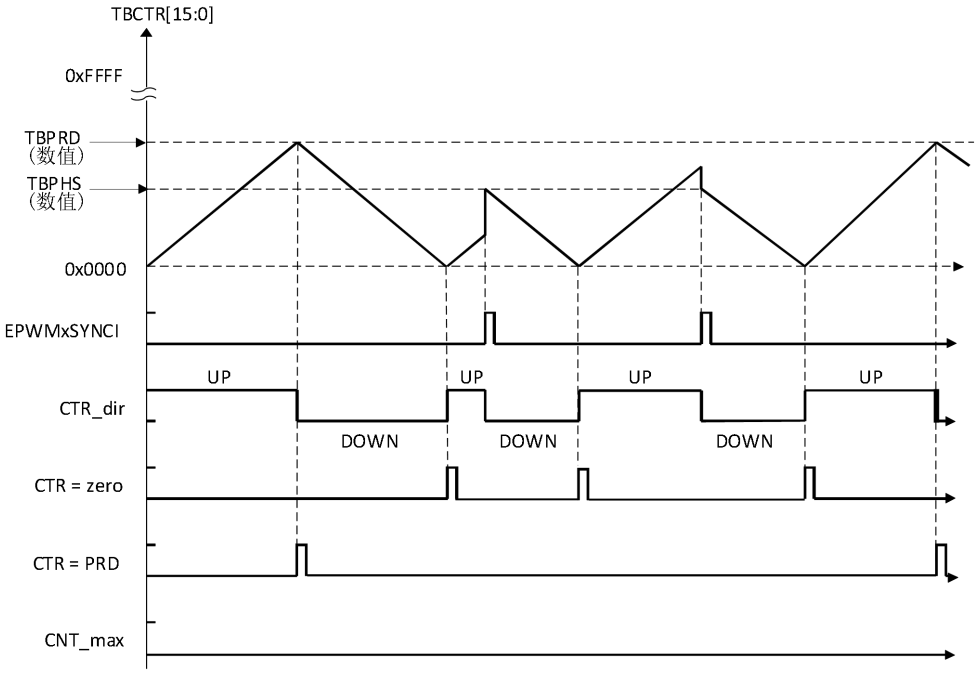



图 15-11 时基向上 - 向下计数波形，TBCTL [PHSDIR = 0] 同步事件时向下计数

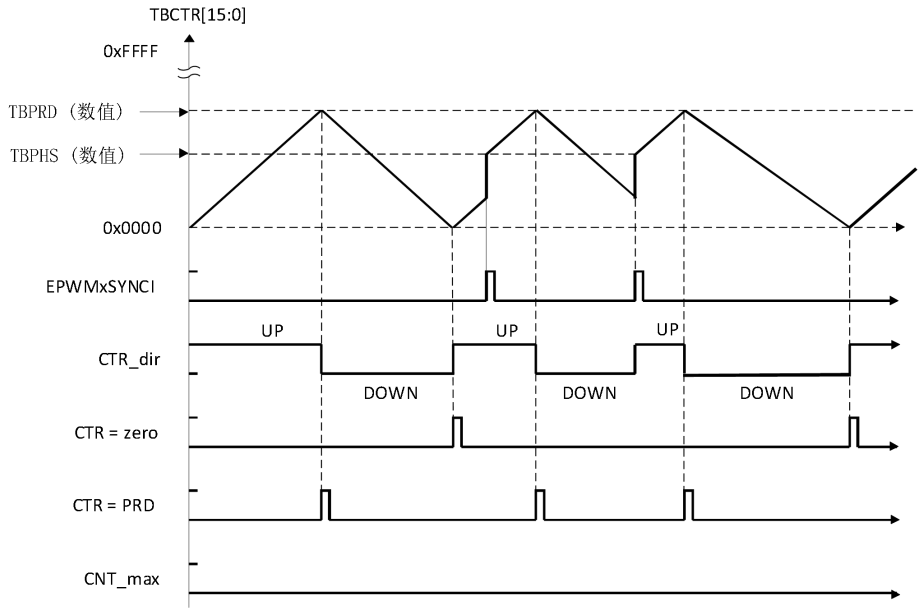



图 15-12 时基向上 - 向下计数波形，TBCTL [PHSDIR = 1] 同步事件时向上计数

## 15.4.7 全局加载

图中展示了与全局加载功能相关的信号和寄存器。

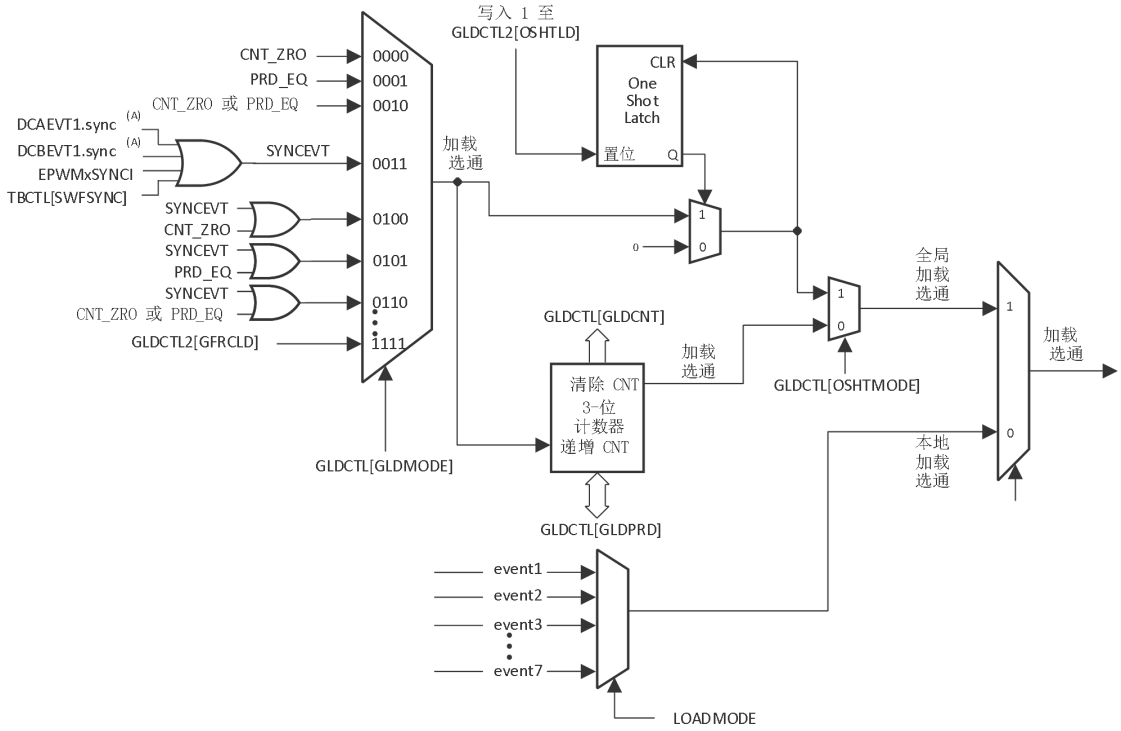



图 15-13 全局加载：信号和寄存器

注：仅当 PHSEN 置位时，SYNCEVT 信号才会传播。

启用此功能后，所有启用了该模式的寄存器，其影子寄存器到活动寄存器的内容传输都会在全局影子到活动加载控制寄存器（GLDCTL [GLDMODE]）的配置位所定义的同一事件上发生。当 GLDCTL [GLD] = 1 时，各个影子寄存器的影子到活动加载事件选择位将被忽略，全局加载模式将对 GLDCFG [REGx] 所启用的对应寄存器生效。

当 GLDCTL [GLD] = 1 且 GLDCFG[REGx]=0 时，全局加载模式不会影响对应寄存器（REGx）。各个影子寄存器的影子到活动加载事件选择位将决定影子寄存器到活动寄存器的内容传输方式。

#### 15.4.7.1 全局加载预分频器

此功能允许选择影子到活动的传输在选定全局加载脉冲（GLDCTL [GLDMODE]）的 “N” 次出现时发生一次。对于无法或未配置为使用全局加载机制的寄存器（即 GLDCTL [GLD] = 0 或 GLDCFG [REGx] = 0），此预分频功能不可用。

#### 15.4.7.2 单触发加载模式

此功能允许用户使影子寄存器到活动寄存器的传输发生一次。当 GLDCTL2[OSHTLD]=1 时，配置为使用全局加载机制的寄存器，其影子到活动寄存器的传输将在 GLDCTL [GLDMODE] 选定的事件上发生。

通过使用 GLDCTL2 [GFRCLD]，可以通过软件强制将内容从影子寄存器加载到活动寄存器。还可以通过使用 EPWMXLINK [GLDCTL2LINK] 在多个 PWM 模块间链接 GLDCTL2 寄存器。结合上述单触发加载模式功能，这提供了一种在特定 PWM 事件时（或根据需要在同一时钟周期内）正确更新一个或多个 PWM 模块中的多个 PWM 寄存器的方法。这在变频应用和 / 或多相交错应用中非常有用。

注：高分辨率模式启用时，不得使用单触发加载模式。

### 15.4.7.3 单触发同步模式

要启用单触发同步模式以生成 SYNCOUT 脉冲，请配置 TBCTL2 [OSHTSYNCMODE] 位并设置 TBCTL2 [OSHTSYNC] 位，如图所示。

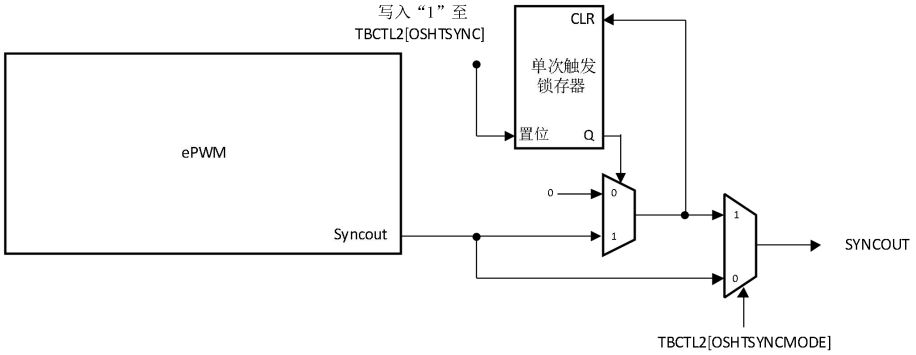



图 15-14 单触发同步模式

## 15.5 计数器比较（CC）子模块

图中展示了 ePWM 内部的计数器比较子模块。

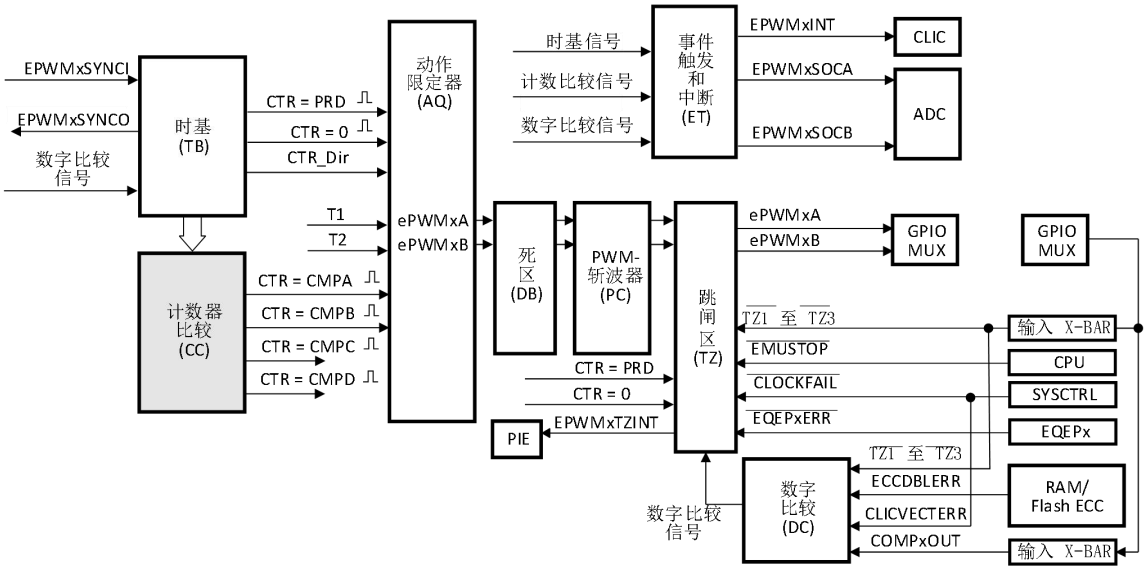



图 15-15 计数器比较子模块

### 15.5.1 计数器比较子模块的用途

计数器比较子模块接收时基计数器值作为输入，并将该值与计数器比较 A（CMPA）、计数器比较 B（CMPB）、计数器比较 C（CMPC）和计数器比较 D

（CMPD）寄存器进行持续比较。当时基计数器的值等于某个比较寄存器的值时，计数器比较单元会生成相应的事件。

计数器比较的功能包括：

基于 CMPA、CMPB、CMPC 和 CMPD 寄存器的可编程时间戳生成事件：

CTR = CMPA：时基计数器等于计数器比较 A 寄存器（TBCTR = CMPA）

CTR = CMPB：时基计数器等于计数器比较 B 寄存器（TBCTR = CMPB）

CTR = CMPC：时基计数器等于计数器比较 C 寄存器（TBCTR = CMPC）

CTR = CMPD：时基计数器等于计数器比较 D 寄存器（TBCTR = CMPD）

如果通过计数器比较 A（CMPA）和计数器比较 B（CMPB）对动作限定器子模块进行了适当配置，则控制 PWM 占空比。

对新的比较值进行影子处理，以防止在活动 PWM 周期内发生损坏或毛刺。

### 15.5.2 计数器比较子模块的控制和监控

计数器比较子模块的工作原理如图所示。

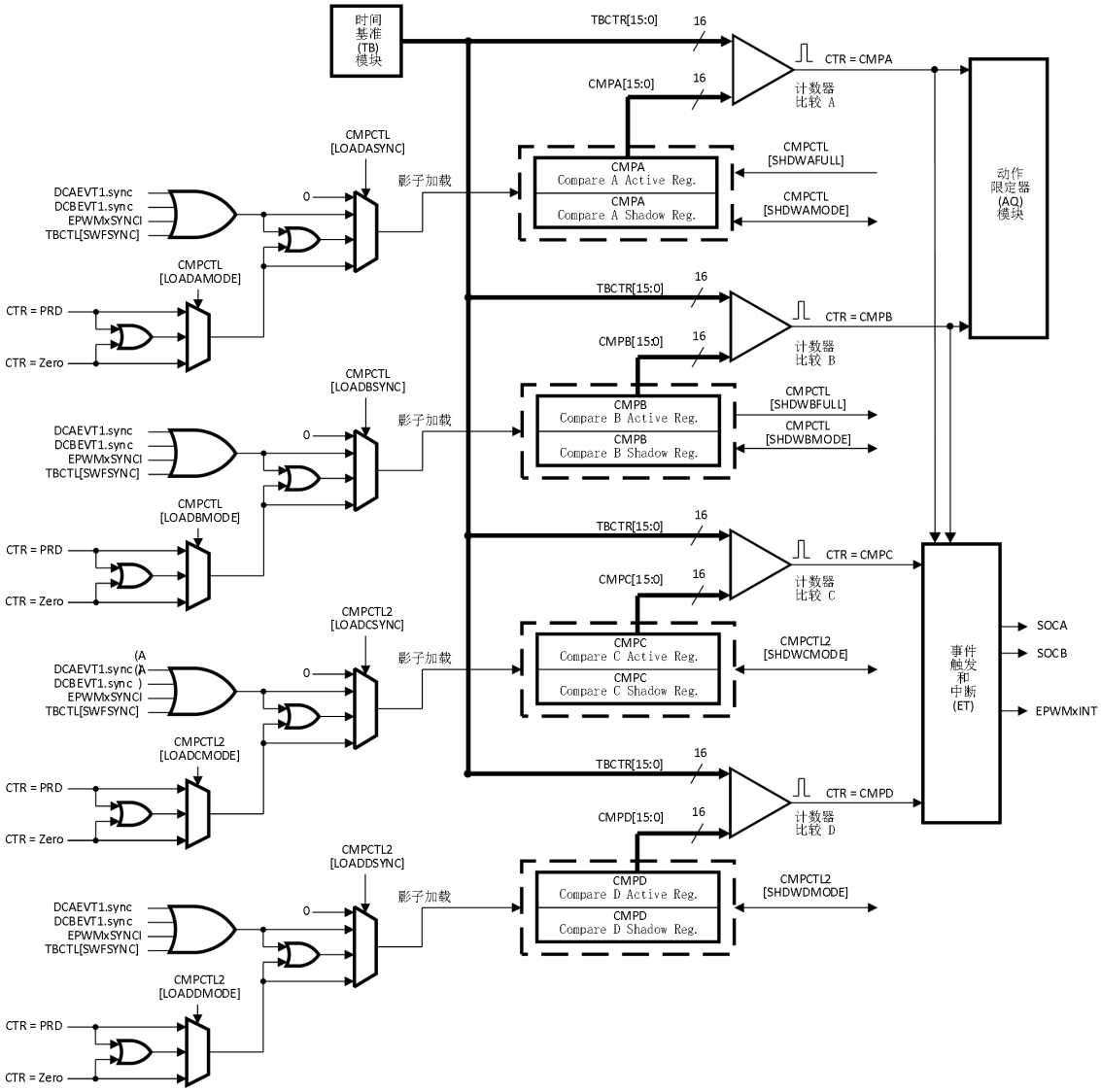



图 15-16 计数器比较子模块的详细视图

A.这些事件由 ePWM 数字比较（DC）子模块根据 TRIPIN 输入信号电平（例如 CMPSSx 和 TZ 信号）生成。

### 15.5.3 计数器比较子模块的操作要点

计数器比较子模块负责生成可用于动作限定器和事件触发子模块的事件。共有四个独立的比较事件：

CTR = CMPA：时基计数器等于计数器比较 A 寄存器（TBCTR = CMPA）。

CTR = CMPB：时基计数器等于计数器比较 B 寄存器（TBCTR = CMPB）。

CTR = CMPC：时基计数器等于计数器比较 C 寄存器（TBCTR = CMPC）。此事件仅可用于在事件触发子模块中生成事件。

CTR = CMPD：时基计数器等于计数器比较 D 寄存器（TBCTR = CMPD）。此事件仅可用于在事件触发子模块中生成事件。

在向上计数或向下计数模式下，每个事件每个周期仅发生一次。在向上 - 向下计数模式下，如果比较值在 0x00 - TBPRD 之间，则每个事件每个周期发生两次；如果比较值等于 0x00 或 TBPRD，则每个事件每个周期发生一次。这些事件会被发送到动作限定器子模块，在该子模块中，事件会根据计数器方向进行限定，并在启用时转换为动作。

计数器比较寄存器 CMPA 和 CMPB 各有一个相关联的影子寄存器。影子处理提供了一种使寄存器更新与硬件同步的方法。使用影子处理时，仅在特定时刻更新活动寄存器。这可以防止由于软件异步修改寄存器而导致的损坏或误操作。活动寄存器和影子寄存器的内存地址相同。写入或读取哪个寄存器由 CMPCTL [SHDWAMODE] 和 CMPCTL [SHDWBMODE] 位决定。这些位分别启用和禁用 CMPC 影子寄存器和 CMPD 影子寄存器。两种加载模式的行为如下：

(1)影子模式

通过清除 CMPCTL [SHDWAMODE] 位启用 CMPA 的影子模式，通过清除 CMPCTL [SHDWBMODE] 位启用 CMPB 的影子模式。默认情况下，CMPA 和 CMPB 的影子模式均启用。

如果启用了影子寄存器，则影子寄存器的内容会根据 CMPCTL [LOADAMODE]、CMPCTL [LOADBMODE]、CMPCTL [LOADASYNC] 和 CMPCTL [LOADBSYNC] 寄存器位的指定，在以下事件之一发生时传输到活动寄存器：

CTR = PRD：时基计数器等于周期值（TBCTR = TBPRD）。

CTR = Zero：时基计数器等于零（TBCTR=0×00）

CTR = PRD 和 CTR = Zero 两者

由 DCAEVT1、DCBEVT1、EPWMxSYNCI 或 TBCTL [SWFSYNC] 引起的同步事件

同步事件或由 LOADAMODE/LOADBMODE 选择的事件两者

计数器比较子模块仅使用活动寄存器的内容生成发送到动作限定器的事件。

(2)立即加载模式

如果选择了立即加载模式（即 CMPCTL [SHDWAMODE] = 1 或 CMPCTL [SHDWBMODE] = 1），则对寄存器的读取或写入会直接访问活动寄存器。

### 15.5.4 计数模式时序波形

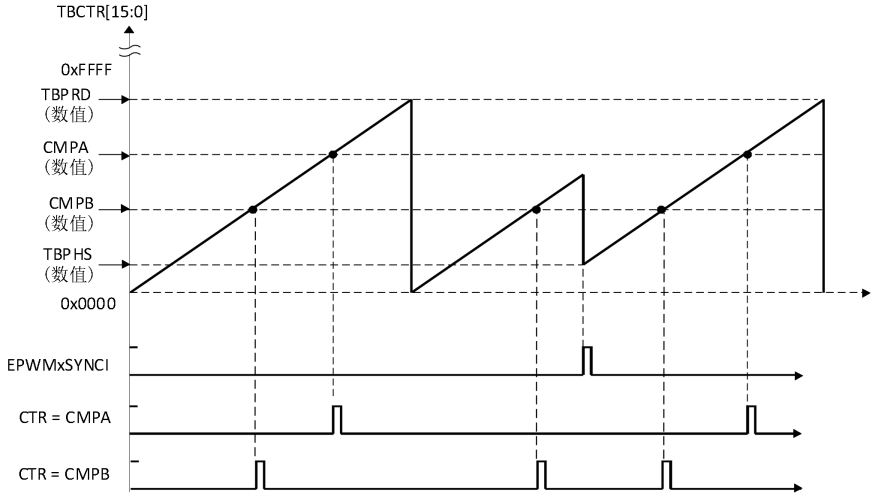
计数器比较模块可以在所有三种计数模式下生成比较事件：

向上计数模式：用于生成非对称 PWM 波形。

向下计数模式：用于生成非对称 PWM 波形。

向上 - 向下计数模式：用于生成对称 PWM 波形。

为了最好地说明前三种模式的工作原理，图 中的时序图展示了事件的生成时机以及 EPWMxSYNCI 信号的交互方式。




EPWMxSYNCI 外部同步事件可能导致 TBCTR 计数序列不连续，从而跳过比较事件。这属于正常操作，必须予以考虑。

图 15-17 向上计数模式下的计数器比较事件波形

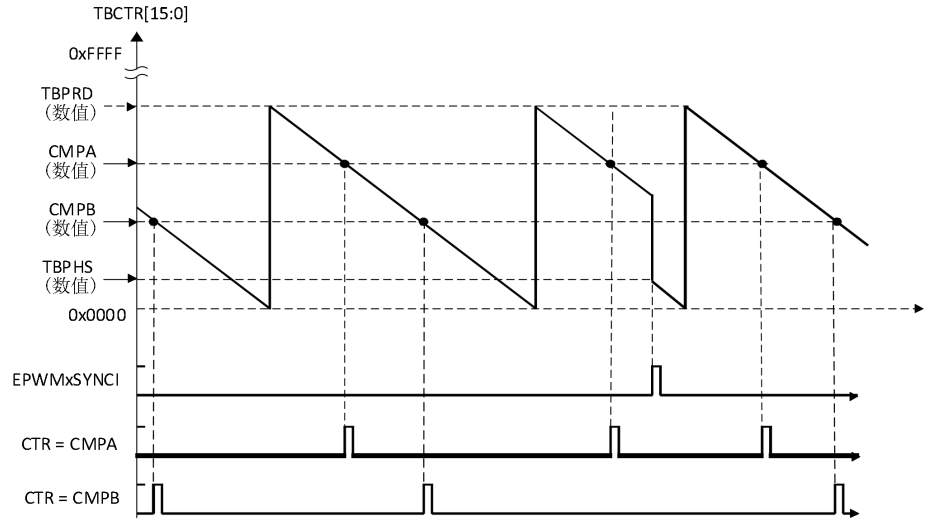



图 15-18 向下计数模式下的计数器比较事件

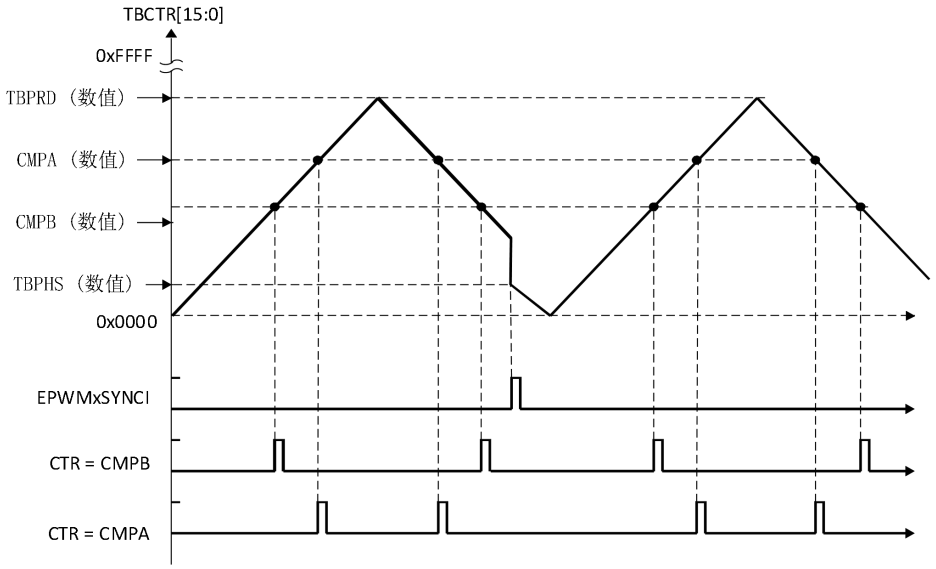



图 15-19 向上 - 向下计数模式下的计数器比较事件，TBCTL [PHSDIR = 0] 同步事件时向下计数

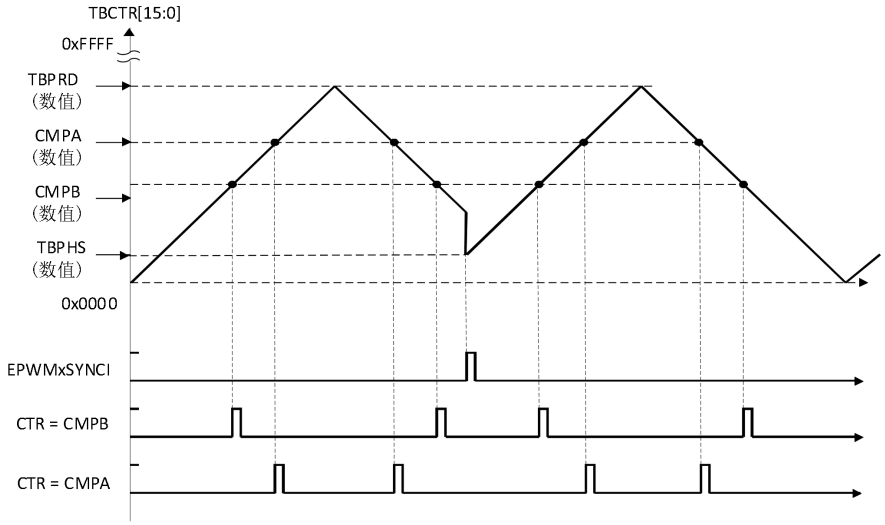



图 15-20 向上 - 向下计数模式下的计数器比较事件，TBCTL [PHSDIR = 1] 同步事件时向上计数

## 15.6 动作限定器（AQ）子模块

动作限定器子模块在波形构建和 PWM 生成中起着最重要的作用。动作限定器子模块决定哪些事件会转换为各种动作类型，从而在 EPWMxA 和 EPWMxB 输出端产生所需的开关波形。

图中展示了 ePWM 内部的动作限定器子模块。

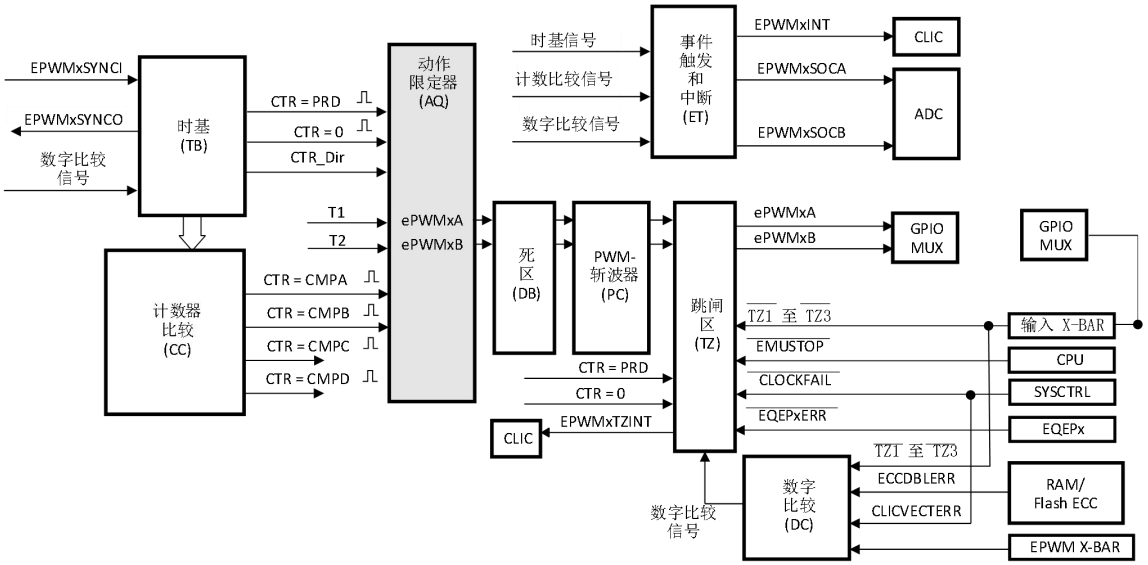



图 15-21 动作限定器子模块

### 15.6.1 动作限定器子模块的用途

动作限定器子模块负责以下工作：

基于以下事件限定并生成动作（置位、清除、翻转）：

CTR = PRD：时基计数器等于周期值（TBCTR = TBPRD）。

CTR = Zero：时基计数器等于零（TBCTR=0×00）

CTR = CMPA：时基计数器等于计数器比较 A 寄存器（TBCTR = CMPA）

CTR = CMPB：时基计数器等于计数器比较 B 寄存器（TBCTR = CMPB）

T1、T2 事件：基于比较器、跳闸或同步输入事件的触发事件

管理这些事件同时发生时的优先级。

当时基计数器递增和递减时，对事件提供独立控制。

### 15.6.2 动作限定器子模块控制和状态寄存器定义

动作限定器子模块的工作原理如图所示。

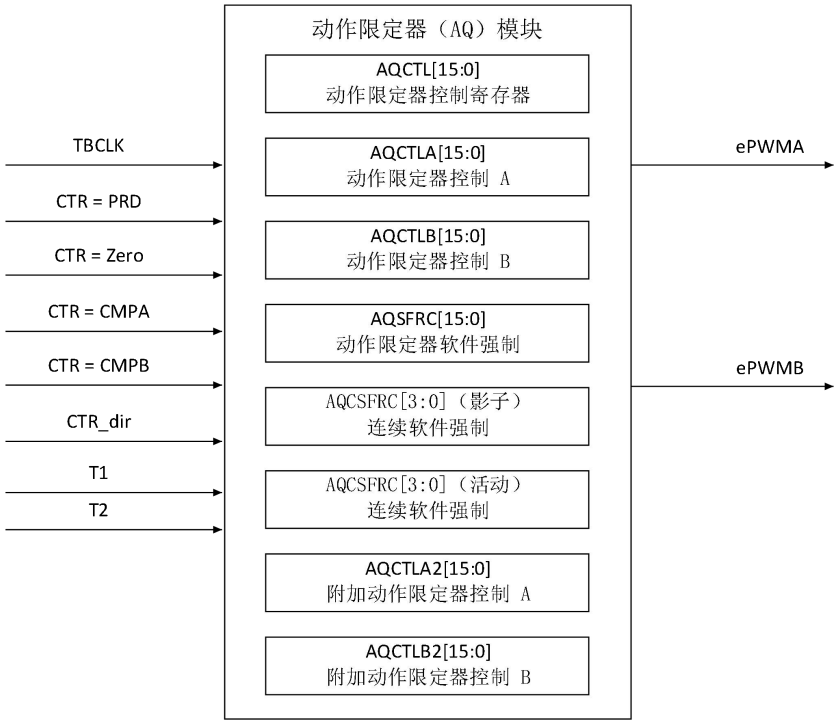



图 15-22 动作限定器子模块的输入和输出

为方便起见，下表再次总结了可能的输入事件。

表 15-4 动作限定器子模块的可能输入事件

| 信号 | 描述 | 比较的寄存器 |
|---|---|---|
| CTR = PRD | 时基计数器等于周期值 | TBCTR = TBPRD |
| CTR = Zero | 时基计数器等于 0 | TBCTR = 0x00 |
| CTR = CMPA | 时基计数器等于计数器比较 A | TBCTR = CMPA |
| CTR = CMPB | 时基计数器等于计数器比较 B | TBCTR = CMPB |
| T1 事件 | 基于比较器、跳闸或同步输入事件 | 无 |
| T2 事件 | 基于比较器、跳闸或同步输入事件 | 无 |
| 软件强制事件 | 由软件启动的异步事件 | 无 |

软件强制动作是一种有用的异步事件。此控制由 AQSFRC 和 AQCSFRC 寄存器处理。

注：如果 CSFA 未在影子模式下使用，则必须配置 RLDCSF 位以禁用影子模式。

动作限定器子模块控制当特定事件发生时，两个输出 EPWMxA 和 EPWMxB 的行为。动作限定器子模块的事件输入会进一步根据计数器方向（向上或向下）进行限定。这允许在计数上升沿和计数下降沿阶段对输出进行独立动作。

施加在输出 EPWMxA 和 EPWMxB 上的可能动作包括：

置高：将输出 EPWMxA 或 EPWMxB 设置为高电平。

清零：将输出 EPWMxA 或 EPWMxB 设置为低电平。

翻转：如果 EPWMxA 或 EPWMxB 当前为高电平，则将输出拉低；如果当前为低电平，则将输出拉高。

无动作：保持输出 EPWMxA 和 EPWMxB 与当前设置的电平相同。尽管“无动作”选项会阻止事件对 EPWMxA 和 EPWMxB 输出产生动作，但此事件仍可触发中断和 ADC 转换启动。

动作是为每个输出（EPWMxA 或 EPWMxB）独立指定的。可以配置任何一个或所有事件在给定输出上生成动作。例如，CTR = CMPA 和 CTR = CMPB 都可以对输出 EPWMxA 进行操作。所有限定器动作都使用本节中的控制寄存器进行配置。

为清晰起见，本章中的插图使用了一组象征性的动作。这些符号在图中进行了总结。每个符号表示一个时间标记动作。一些动作的时间是固定的（零和周期），而

CMPA 和 CMPB 动作是可移动的，其时间位置分别通过计数器比较 A 和 B 寄存器进行编程。要关闭或禁用某个动作，请使用“无动作”选项（复位时的默认值）。

表 15-5 EPWMxA 和 EPWMxB 输出的可能动作限定器动作

| 软件强制 | TB 计数器等于零 | TB 计数器等于比较 A | TB 计数器等于比较 B | TB 计数器等于周期 | 触发事件 T1 | 触发事件 T2 | 动作 |
|---|---|---|---|---|---|---|---|
| SW × | Z × | CA × | CB × | P × | T1 × | T2 × | 不执行操作 |
| SW ↓ | Z ↓ | CA ↓ | CB ↓ | P ↓ | T1 ↓ | T2 ↓ | 清零为低电平 |
| SW ↑ | Z ↑ | CA ↑ | CB ↑ | P ↑ | T1 ↑ | T2 ↑ | 置位为高电平 |
| SW | Z | CA | CB | P | T1 | T2 | 翻转 |

动作限定器触发事件源选择寄存器（AQTSRCSEL）用于选择 T1 和 T2 事件的源。动作限定器子模块中跳闸 / 数字比较事件的 T1/T2 选择和配置独立于跳闸区域子模块中该事件的配置。某个跳闸事件可以配置为在跳闸区域子模块中引起跳闸动作，也可以不配置，但同一事件可被动作限定器用于生成 T1/T2 以控制 PWM 生成。

### 15.6.3 动作限定器事件优先级

ePWM 动作限定器可能同时接收多个事件。在这种情况下，硬件会为事件分配优先级。一般规则是，时间上较晚发生的事件优先级较高，软件强制事件的优先级始终最高。向上 - 向下计数模式下的事件优先级级别如表所示。优先级级别 1 为最高优先级，级别 10 为最低优先级。优先级会根据 TBCTR 的方向略有变化。

表 15-6 向上 - 向下计数模式下的动作限定器事件优先级

| 优先级级别 | TBCTR 递增时的事件（TBCTR 从 0 到 TBPRD） | TBCTR 递减时的事件（TBCTR 从 TBPRD 到 1） |
|---|---|---|
| 1(最高) | 软件强制事件 | 软件强制事件 |
| 2 | 向上计数时的 T1（T1U） | 向下计数时的 T1（T1D） |
| 3 | 向上计数时的 T2（T2U） | 向下计数时的 T2（T2D） |
| 4 | 向上计数时计数器等于 CMPB(CBU) | 向下计数时计数器等于 CMPB(CBD) |
| 5 | 向上计数时计数器等于 CMPA(CAU) | 向下计数时计数器等于 CMPA(CAD) |
| 6(最低) | 计数器等于零 | 计数器等于周期（TBPRD） |

表中显示了向上计数模式下的动作限定器优先级。在这种情况下，计数器方向始终定义为向上；因此，不会发生向下计数事件。

表 15-7 向上计数模式下的动作限定器事件优先级

| 优先级级别 | 事件 |
|---|---|
| 1（最高） | 软件强制事件 |
| 2 | 计数器等于周期（TBPRD） |
| 3 | 向上计数时的 T1（T1U） |
| 4 | 向上计数时的 T2（T2U） |
| 5 | 向上计数时计数器等于 CMPB（CBU） |
| 6 | 向上计数时计数器等于 CMPA（CAU） |
| 7（最低） | 计数器等于零 |

表中显示了向下计数模式下的动作限定器优先级。在这种情况下，计数器方向始终定义为向下；因此，不会发生向上计数事件。

表 15-8 向下计数模式下的动作限定器事件优先级

| 优先级级别 | 事件 |
|---|---|
| 1（最高） | 软件强制事件 |
| 2 | 计数器等于零 |
| 3 | 向下计数时的 T1（T1D） |
| 4 | 向下计数时的 T2（T2D） |
| 5 | 向下计数时计数器等于 CMPB（CBD） |
| 6 | 向下计数时计数器等于 CMPA（CAD） |
| 7（最低） | 计数器等于周期（TBPRD） |

可以将比较值设置为大于周期值。在这种情况下，动作的发生方式如表所示。

表 15-9 当 CMPA/CMPB 大于周期时的行为

| 计数器模式 | 向上计数事件时的比较（CAU/CBU） | 向下计数事件时的比较（CAD/CBD） |
|---|---|---|
| 向上计数模式 | 如果 CMPA/CMPB ≤ TBPRD 周期，则事件在比较匹配时发生（TBCTR = CMPA 或 CMPB）。如果 CMPA/CMPB > TBPRD，则事件不会发生。 | 从不发生。 |
| 向下计数模式 | 从不发生。 | 如果 CMPA/CMPB < TBPRD，则事件在比较匹配时发生（TBCTR = CMPA 或 CMPB）。如果 CMPA/CMPB ≥ TBPRD，则事件在周期匹配时发生（TBCTR = TBPRD）。 |
| 向上 - 向下计数模式 | 如果 CMPA/CMPB < TBPRD 且计数器正在递增，则事件在比较匹配时发生（TBCTR = CMPA 或 CMPB）。如果 CMPA/CMPB ≥ TBPRD，则事件在周期匹配时发生（TBCTR = TBPRD）。 | 如果 CMPA/CMPB < TBPRD 且计数器正在递减，则事件在比较匹配时发生（TBCTR = CMPA 或 CMPB）。如果 CMPA/CMPB ≥ TBPRD，则事件在周期匹配时发生（TBCTR = TBPRD）。 |

### 15.6.4 AQCTLA 和 AQCTLB 影子模式操作

在 2 型及更高版本的 ePWM 上，新增了 AQCTLA 和 AQCTLB 寄存器的影子处理功能，以便在相位变化时，动作限定器模式的更改必须在周期结束时发生。此外，还支持这些寄存器在同步时的影子到活动加载。通过 AQCTL [SHDWAQAMODE] 和 AQCTL [SHDWAQBMODE] 位启用和禁用 AQCTLA 影子寄存器和 AQCTLB 影子寄存器。这两位分别启用和禁用 AQCTLA 影子寄存器和 AQCTLB 影子寄存器。两种加载模式的行为如下：

（1）影子模式

通过设置 AQCTL [SHDWAQAMODE] 位启用 AQCTLA 的影子模式，通过设置 AQCTL [SHDWAQBMODE] 位启用 AQCTLB 的影子模式。默认情况下，AQCTLA 和 AQCTLB 的影子模式均禁用。

如果启用了影子寄存器，则影子寄存器的内容会根据 AQCTL [LDAQAMODE]、AQCTL [LDAQBMODE]、AQCTL [LDAQASYNC] 和 AQCTL [LDAQBSYNC] 寄存器位的指定，在以下事件之一发生时传输到活动寄存器：

CTR = PRD：时基计数器等于周期值（TBCTR = TBPRD）。

CTR = Zero：时基计数器等于零（TBCTR=0×00）

CTR = PRD 和 CTR = Zero 两者

由 DCAEVT1、DCBEVT1、EPWMxSYNCI 或 TBCTL [SWFSYNC] 引起的同步事件

同步事件或由 LDAQAMODE/LDAQBMODE 选择的事件两者

（2）全局加载支持

通过在全局加载配置寄存器（GLDCFG）中配置相应的位，全局加载控制机制也可用于 AQCTLA:AQCTLA2、AQCTLB:AQCTLB2 和 AQCSFRC 寄存器。当选择全局加载模式时，所有启用了该模式的寄存器，其影子寄存器到活动寄存器的内容

传输都会在全局影子到活动加载控制寄存器（GLDCTL）的配置位所定义的同一事件上发生。

（3）立即加载模式

如果选择了立即加载模式（即 AQCTL [SHDWAQAMODE] = 0 或 AQCTL [SHDWAQBMODE] = 0），则对寄存器的读取或写入会直接访问活动寄存器。

注：动作限定器输出 A/B 控制寄存器（AQCTLA 和 AQCTLB）在 CMPA=0 或 CMPB=0 边界上的影子到活动加载。

如果计数器比较 A 寄存器（CMPA）或计数器比较 B 寄存器（CMPB）被设置为 0 值，并且 AQCTLA 和 AQCTLB 上的动作限定器动作配置为与影子到活动加载在同一时刻发生（即 CMPA＝0 且 AQCTLA 影子到活动加载在 TBCTR＝0 时使用 AQCTL 寄存器的 LDAQAMODE 和 LDAQAMODE 位），则两个事件会产生冲突。建议在 TBCTR = 0 边界上使用动作限定器输出 A/B 控制寄存器的影子到活动加载时，使用非零计数器比较值。

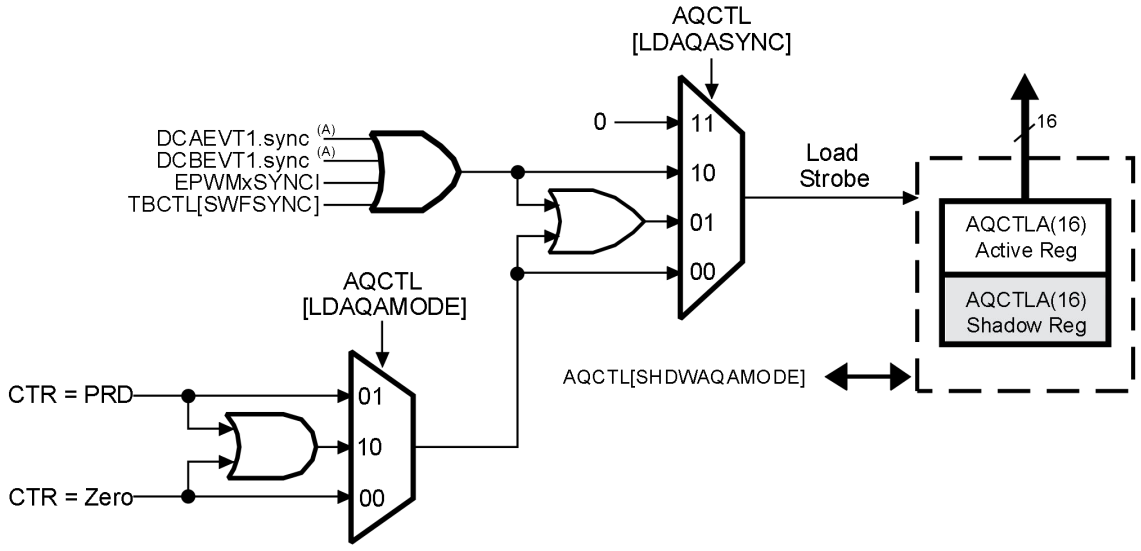



图 15-23 AQCTL [SHDWAQAMODE]

A.这些事件由 ePWM 数字比较（DC）子模块根据 TRIPIN 输入信号电平（例如 CMPSSx 和 TZ 信号）生成。

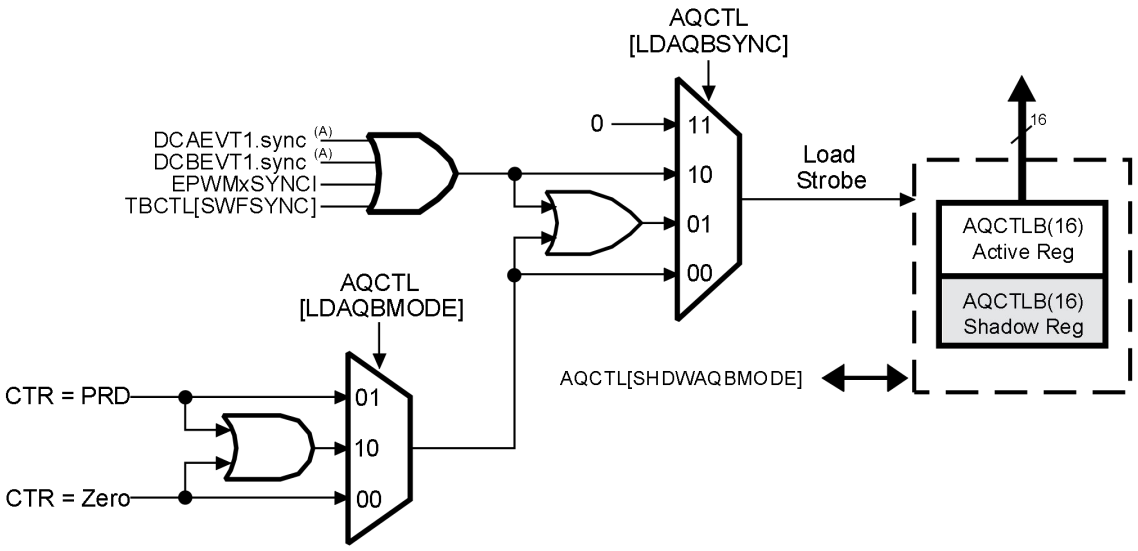



图 15-24 AQCTL [SHDWAQBMODE]

A.这些事件由 ePWM 数字比较（DC）子模块根据 TRIPIN 输入信号电平（例如 CMPSSx 和 TZ 信号）生成。

### 15.6.5 常见波形的配置要求

注：本章中的波形显示了 ePWM 在静态比较寄存器值下的行为。在运行系统中，活动比较寄存器（CMPA 和 CMPB）通常每个周期从相应的影子寄存器更新一次。指定更新发生的时间：当时基计数器达到零或当时基计数器达到周期时。在某些情况下，基于新值的动作可能会延迟一个周期，或者基于旧值的动作可能会额外生效一个周期。某些 PWM 配置可以避免这种情况。这些配置包括但不限于以下内容：

使用向上 - 向下计数模式生成对称 PWM：

如果在零时刻加载 CMPA/CMPB，则使用大于或等于 1 的 CMPA/CMPB 值。

如果在周期时刻加载 CMPA/CMPB，则使用小于或等于 TBPRD - 1 的 CMPA/CMPB 值。这意味着在一个 PWM 周期中始终有至少一个 TBCLK 周期的脉冲，当脉冲非常短时，系统往往会忽略它。

使用向上 - 向下计数模式生成非对称 PWM：

要实现 50%-0% 非对称 PWM，请使用以下配置：在周期时刻加载 CMPA/CMPB，并使用周期动作清除 PWM，使用比较向上动作置位 PWM。将比较值从 0 调制到 TBPRD，以实现 50%-0% PWM 占空比。

当使用向上计数模式生成非对称 PWM 时：

要实现 0-100% 非对称 PWM，必须在 TBPRD 时刻加载 CMPA/CMPB。当 CMPA/CMPB 不在 TBCTR = PRD 时加载时，根据写入的时序和写入 CMPA/CMPB 的值，可能会出现边界条件。使用零动作置位 PWM，并使用比较向上动作清除 PWM。将比较值从 0 调制到 TBPRD + 1，以实现 0-100% PWM 占空比。

当使用向上计数模式生成启用了死区的非对称 PWM 时：

要实现 0%-100% PWM，请使用以下配置：当 CMPA 值过于接近 0 或 PRD，导致满足以下条件（CMPX <死区）或（CMPX> PRD - 死区）时，AQCTL 寄存器为 CMPX 指定的动作不会生效。为避免这种情况，仅在这些条件下必须更改 AQCTL 设置，为 CAU 或 CAD 事件生成高脉冲或低脉冲（均置位或均清除）。确保此软件更新与 PWM 载波周期同步，并且启用了影子模式。

当使用向上 - 向下计数模式生成启用了死区的非对称 PWM 时：

要实现 0%-100% PWM，请使用以下配置：当 CMPA 值过于接近 0 或 PRD，导致满足以下条件（CMPX <死区 / 2）或（CMPX> PRD -（死区）/2）时，AQCTL 寄存器为 CMPX 指定的动作不会生效。为避免这种情况，仅在这些条件下必须更改 AQCTL 设置，为 CAU 或 CAD 事件生成高脉冲或低脉冲（均置位或均清除）。确保此软件更新与 PWM 载波周期同步，并且启用了影子模式。

图中展示了如何使用 TBCTR 的向上 - 向下计数模式生成对称 PWM 波形。在这种模式下，通过在波形的向上计数和向下计数部分使用相等的比较匹配，可以实现 0%-100% 占空比调制。在所示示例中，使用 CMPA 进行比较。当计数器递增时，CMPA 匹配将 PWM 输出拉高。同样，当计数器递减时，比较匹配将 PWM 信号

拉低。当 CMPA = 0 时，PWM 信号在整个周期内为高电平，占空比为 100%。当 CMPA = TBPRD 时，PWM 信号为低电平，占空比为 0%。

在实际使用此配置时，如果在零时刻加载 CMPA/CMPB，则使用大于或等于 1 的 CMPA/CMPB 值。如果在周期时刻加载 CMPA/CMPB，则使用小于或等于 TBPRD - 1 的 CMPA/CMPB 值。这意味着在一个 PWM 周期中始终有至少一个 TBCLK 周期的脉冲，当脉冲非常短时，系统往往会忽略它。

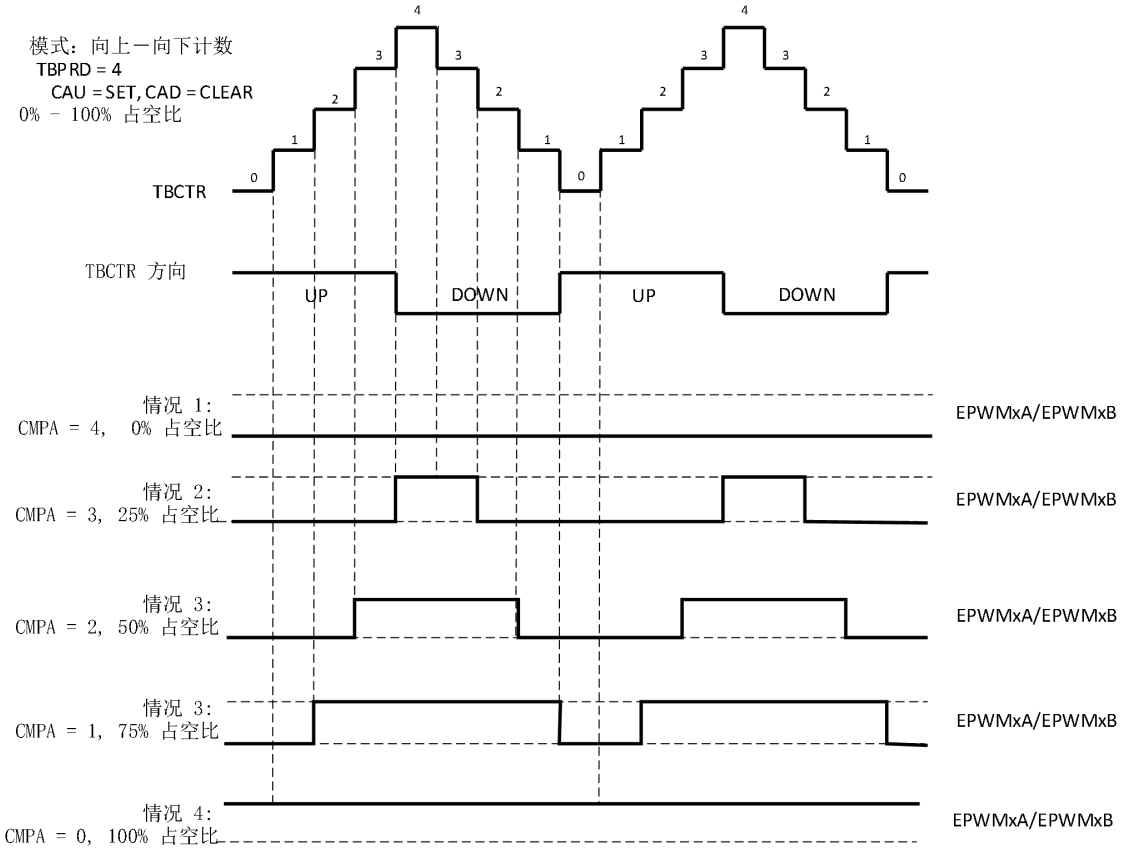



图 15-25 向上 - 向下计数模式对称波形

图 中的 PWM 波形展示了一些常见的动作限定器配置。图中和示例中使用的一些约定如下：

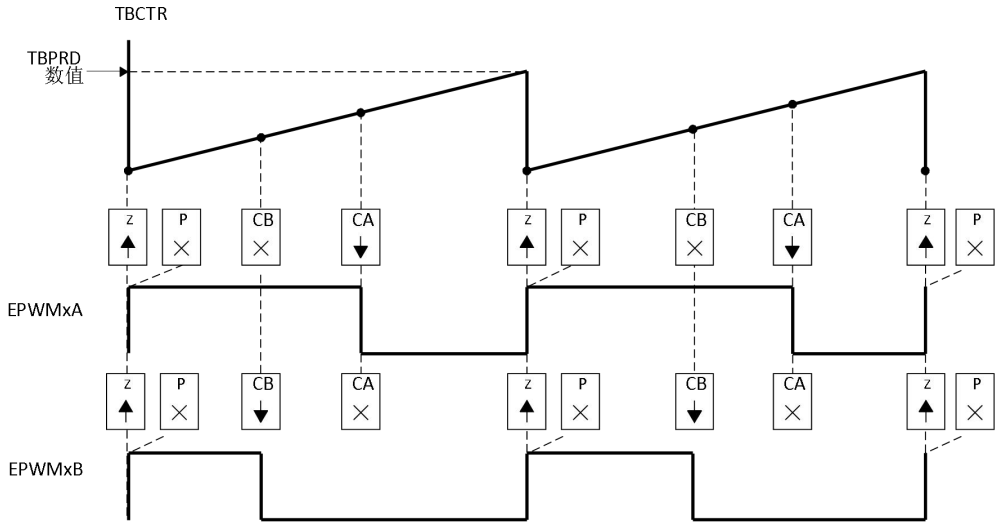
TBPRD、CMPA 和 CMPB 指写入相应寄存器的值。硬件使用的是活动寄存器，而不是影子寄存器。

CMPx 指 CMPA 或 CMPB。

EPWMxA 和 EPWMxB 指来自 ePWMx 的输出信号。

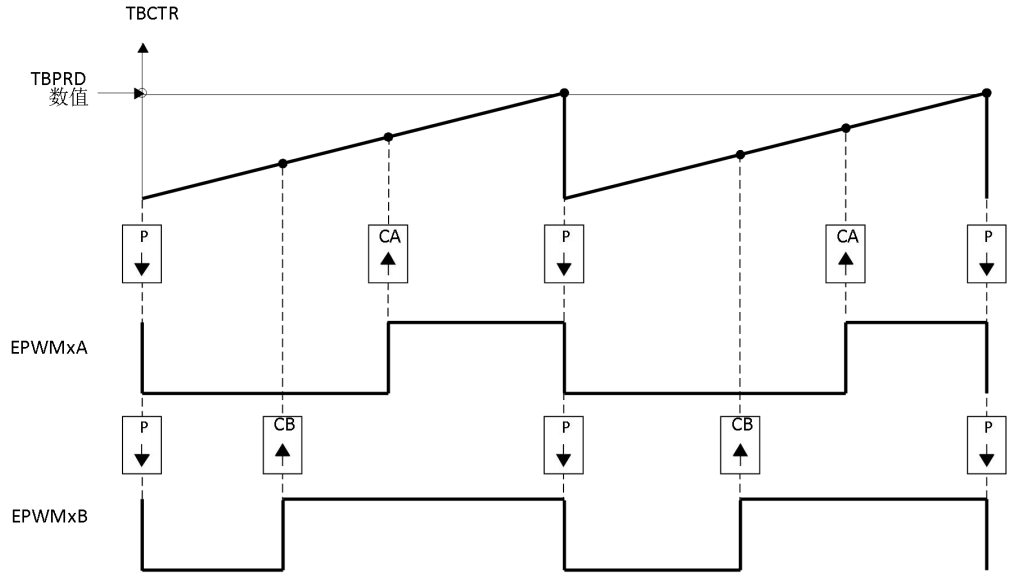
Up-Down 表示向上 - 向下计数模式，Up 表示向上计数模式，Down 表示向下计数模式。

Sym = 对称，Asym = 非对称。



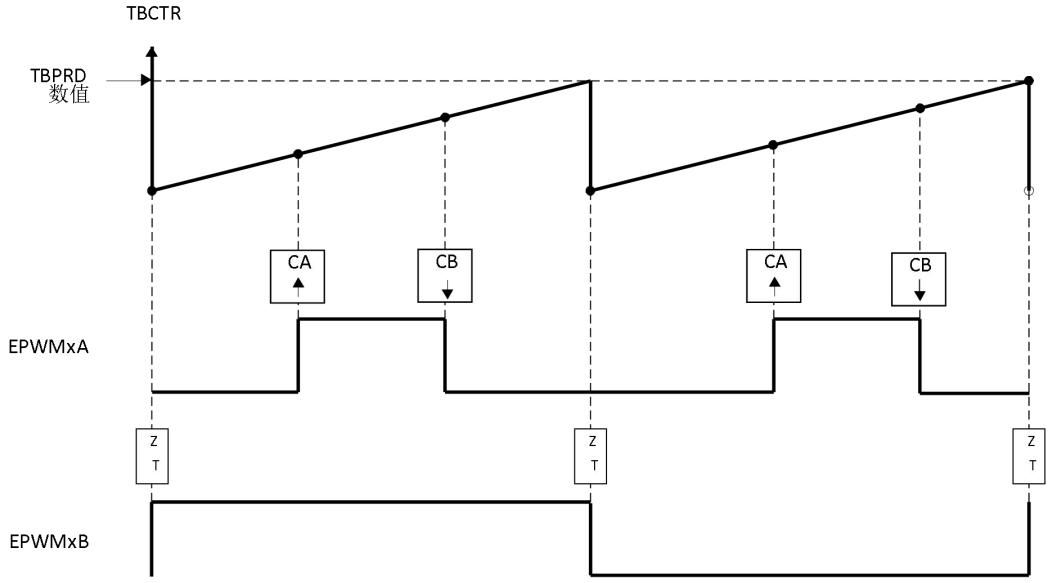

A. PWM 周期 = (TBPRD + 1) × $T_{TBCLK}$
B. EPWMxA 的占空比调制由 CMPA 设置，高电平有效（高电平占空比与 CMPA 成正比）。
C. EPWMxB 的占空比调制由 CMPB 设置，高电平有效（高电平占空比与 CMPB 成正比）。
D. 为完整起见，此图示出“不执行操作”（X）；后续图中省略。
E. 零点与周期点的动作看似同时发生，实际间隔一个 TBCLK 周期。TBCTR 从 周期 至 0000.

图 15-26 向上、单边缘非对称波形，EPWMxA 和 EPWMxB 上的独立调制 - 高电平有效




A. PWM 周期 = (TBPRD + 1) × $T_{TBCLK}$
B. EPWMxA 的占空比调制由 CMPA 设置，低电平有效（低电平占空比与 CMPA 成正比）。
C. EPWMxB 的占空比调制由 CMPB 设置，低电平有效（低电平占空比与 CMPB 成正比）。

图 15-27 向上、单边缘非对称波形，EPWMxA 和 EPWMxB 上的独立调制 - 低电平有效

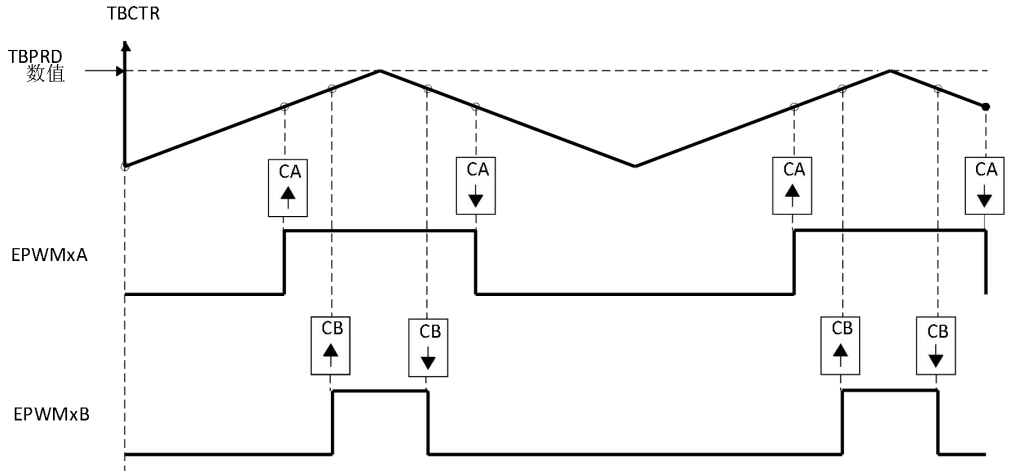



A. PWM 频率 = 1/((TBPRD + 1) × $T_{TBCLK}$)

B. 脉冲可置于 PWM 周期（0000～TBPRD）内的任意位置

C. 高电平占空比与（CMPB - CMPA）成正比

图 15-28 向上计数、脉冲定位非对称波形，EPWMxA 上的独立调制



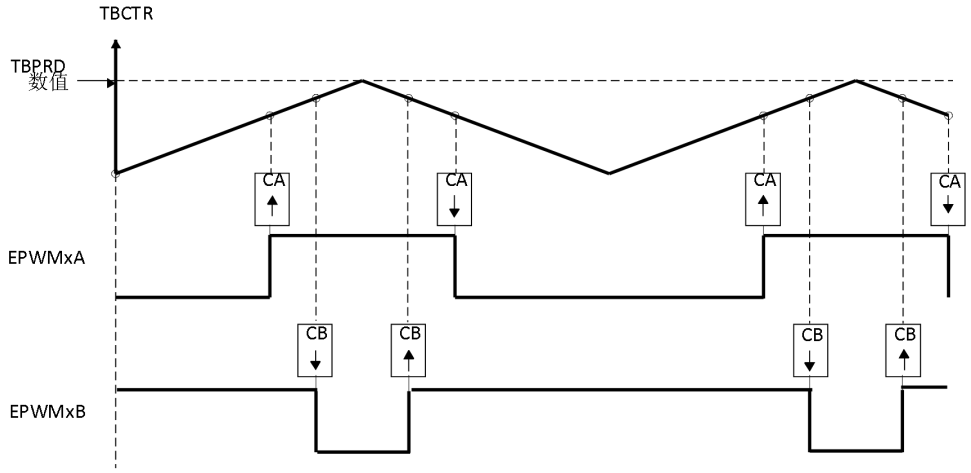

A. PWM 周期 = 2 x TBPRD × $T_{TBCLK}$

B. EPWMxA 的占空比调制由 CMPA 设置，低电平有效（低电平占空比与 CMPA 成正比）。

C. EPWMxB 的占空比调制由 CMPB 设置，低电平有效（低电平占空比与 CMPB 成正比）。

D. EPWMxA 和 EPWMxB 输出可驱动独立的功率开关。

图 15-29 向上 - 向下计数、双边缘对称波形，EPWMxA 和 EPWMxB 上的独立调制 - 低电平有效




A. PWM 周期 = 2 × TBPRD × $T_{TBCLK}$

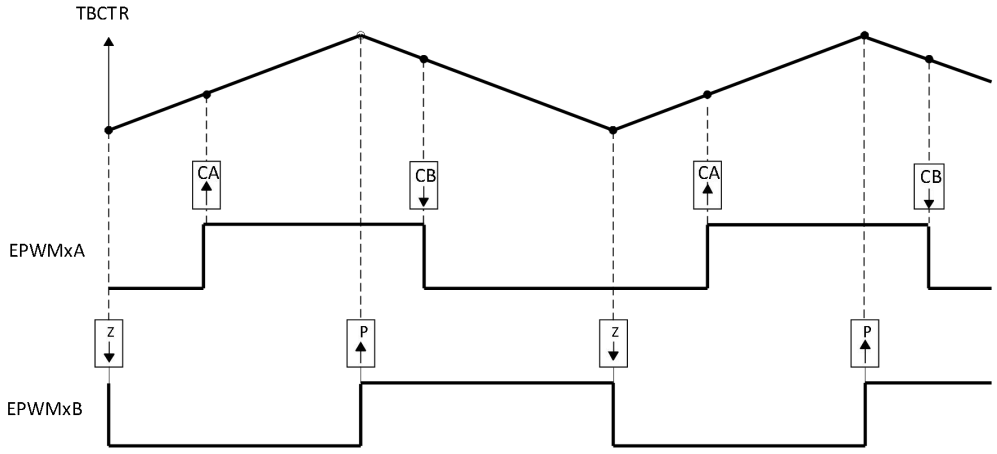
B. EPWMxA 的占空比调制由 CMPA 设置，低电平有效，即低电平占空比与 CMPA 成正比。

C. EPWMxB 的占空比调制由 CMPB 设置，高电平有效，即高电平占空比与 CMPB 成正比。

D. EPWMx 输出可驱动上 / 下桥臂（互补）功率开关。

E. 死区 = CMPB – CMPA（边沿位置完全由软件编程）。如果需要更传统的边沿延迟方法，也可使用死区模块。

图 15-30 向上 - 向下计数、双边缘对称波形，EPWMxA 和 EPWMxB 上的独立调制 - 互补




A. PWM 周期 = 2 × TBPRD × TBCLK

B. 上升沿和下降沿可在 PWM 周期内非对称定位，从而实现脉冲定位技术。

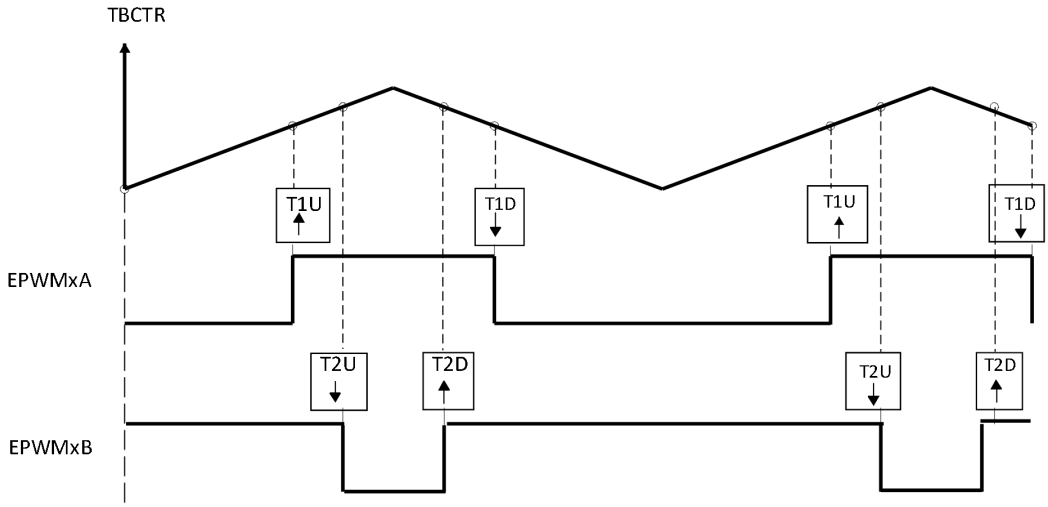
C. EPWMxA 的占空比调制由 CMPA 和 CMPB 设置。

D. EPWMxA 的低电平占空比与（CMPA + CMPB）成正比。

E. 若要改为高电平有效，需反转 CMPA 和 CMPB 的动作（CMPA 时清零，CMPB 时置位）。

F. EPWMxB 的占空比固定为 50%（使用 EPWMxB 的剩余动作资源）。

图 15-31 向上 - 向下计数、双边缘非对称波形，EPWMxA 上的独立调制 - 低电平有效




A. PWM 周期 = 2 × TBPRD × TTBCLK
B. 计数器向上和向下计数时采用独立的 T1 事件动作，生成 EPWMxA输出
C. 计数器向上和向下计数时采用独立的 T2 事件动作，生成 EPWMxB输出
D. 选择 TZ1 作为 T1 的源。
E. 选择 TZ2 作为 T2 的源。

图 15-32 向上 - 向下计数、利用 T1 和 T2 事件生成 PWM 波形

## 15.7 死区生成器（DB）子模块

图中展示了 ePWM 内部的死区子模块。

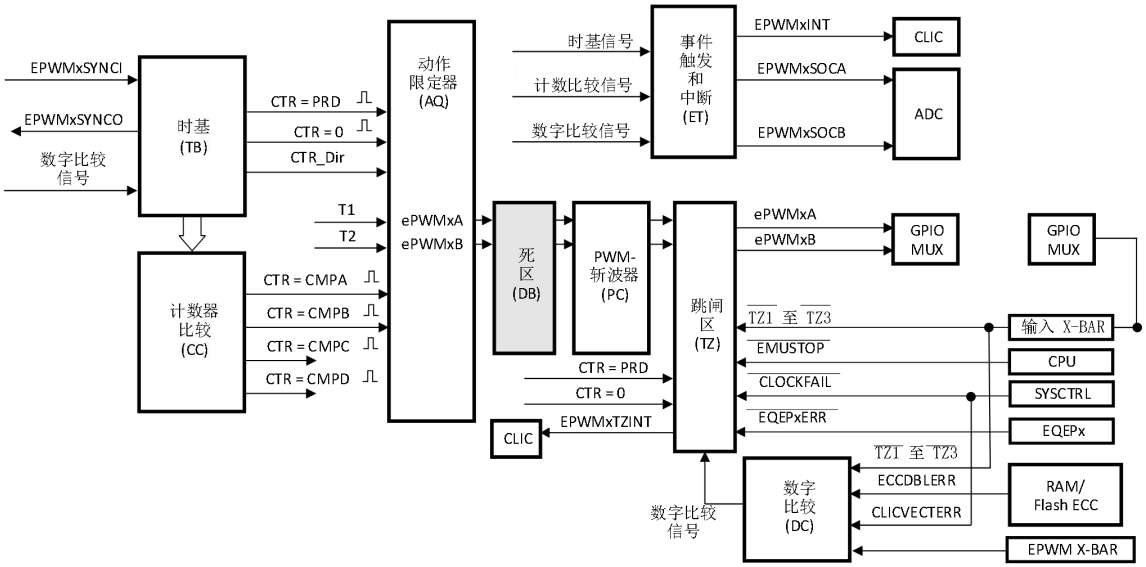



图 15-33 死区子模块

### 15.7.1 死区子模块的用途

动作限定器（AQ）模块部分讨论了 AQ 模块如何通过利用 ePWM 模块的 CMPA 和 CMPB 资源完全控制边缘位置来生成所需的死区。然而，如果需要更传统的基于边缘延迟的带极性控制的死区，则必须使用此处描述的死区子模块。

死区模块的关键功能包括：

从单个 EPWMxA 输入生成具有死区关系的适当信号对（EPWMxA 和 EPWMxB）。

为信号对编程：

- 高电平有效（AH）
- 低电平有效（AL）
- 高电平互补（AHC）
- 低电平互补（ALC）

为上升沿添加可编程延迟（RED）。

为下降沿添加可编程延迟（FED）。

可以从信号路径中完全旁路（注意图中的虚线）。

### 15.7.2 死区子模块的附加操作模式

在 1 型 ePWM 上，RED 可以出现在一个通道输出上，FED 可以出现在另一个通道输出上。

以下列表显示了 1 型和 4 型模块在死区操作模式方面的明显区别：

通过在图中添加 S6、S7 和 S8，RED 和 FED 可以出现在 A 通道和 B 通道输出上。此外，RED 和 FED 可以一起应用于 A 通道或 B 通道输出，以允许 B 通道相对于 A 通道进行相移。

注：使用死区子模块的附加操作模式使 B 通道相对于 A 通道进行相移时，关于 ePWMxA 和 ePWMxB 输出的工作占空比，RED 和 FED 延迟的选择存在限制。

死区计数器也已增加到 14 位。

死区和死区高分辨率寄存器现在具有影子功能。

使用 DBREDHR 和 DBFEDHR 寄存器启用了高分辨率死区 RED 和 FED。

注：启用高分辨率死区时，PWM 斩波器不启用。

高分辨率死区 RED 和 FED 需要半周期时钟模式（DBCTL [HALFCYCLE] = 1）。

不能将 RED 和 FED 一起应用于 ePWMxA 和 ePWMxB。RED 和 FED 只能一起应用于输出 A 或输出 B。

B 通道相对于 A 通道的相移：当使用 DEDB_MODE 位并通过相移量延迟上升沿和下降沿，从 PWMxA 导出 PWMxB 时。当 PWMxA 上的占空比值小于此相移量时，PWMxA 的下降沿优先于 PWMxB 的延迟上升沿。确保应用于死区模块的当前波形的占空比值大于所需的相移量。

ePWM 模块的 4 型动作限定器和死区输出比 2 型 ePWM 模块延迟一个 TBCLK 周期，尽管 4 型的行为与 3 型 PWM 相同。在所有情况下，PWMA 和 PWMB 信号都会延迟。

（1）影子模式

通过设置 DBCTL [SHDWDBREDMODE] 位启用 DBRED 的影子模式，通过设置 DBCTL [SHDWDBFEDMODE] 位启用 DBFED 的影子模式。默认情况下，DBRED 和 DBFED 的影子模式均禁用。

如果启用了影子寄存器，则影子寄存器的内容会根据 DBCTL [LOADREDMODE] 和 DBCTL [LOADFEDMODE] 寄存器位的指定，在以下事件之一发生时传输到活动寄存器：

CTR = PRD：时基计数器等于周期值（TBCTR = TBPRD）。

CTR = Zero：时基计数器等于零（TBCTR=0×00）

CTR = PRD 和 CTR = Zero 两者

DBCTL 寄存器可以进行影子处理。通过设置 DBCTL2 [SHDWDBCTLMODE] 位启用 DBCTL 的影子模式。如果启用了影子寄存器，则影子寄存器的内容会根据 DBCTL2 [LOADDBCTLMODE] 寄存器位的指定，在以下事件之一发生时传输到活动寄存器：

CTR = PRD：时基计数器等于周期值（TBCTR = TBPRD）

CTR = Zero：时基计数器等于零（TBCTR=0×00）

CTR = PRD 和 CTR = Zero 两者

注：应用程序软件必须在对 DBRED 和 DBFED 寄存器进行编程之前，在 DBCTL [SHDWDBREDMODE] 和 DBCTL [SHDWDBFEDMODE] 中启用影子加载模式。如果在对 DBRED 和 DBFED 寄存器进行编程后启用影子寄存器，则 DBRED 和 DBFED 寄存器会加载 0 值。

（2）全局加载支持

通过在全局加载配置寄存器（GLDCFG）中配置相应的位，全局加载控制机制也可用于 DBRED:DBREDHR、DBFED:DBFEDHR 和 DBCTL 寄存器。当选择全局加载模式时，所有启用了该模式的寄存器，其影子寄存器到活动寄存器的内容传输都会在全局影子到活动加载控制寄存器（GLDCTL）的配置位所定义的同一事件上发生。

注：当 DB 计数器正在计数时，如果 DBRED/DBFED 活动寄存器加载了新的影子值，则新的 DBRED/DBFED 值仅影响下一个 PWMx 边缘，而不影响当前边缘。

当全局影子到活动加载设置为在 CTR=ZERO 时发生时，死区值不能为零。同样，当全局影子到活动加载设置为在 CTR=PRD 时发生时，死区值不能为 PRD。

TBPRDHR 不能与全局加载一起使用。如果应用中必须更改高分辨率周期，用户必须从 ePWM ISR（ISR 必须与 PWM 开关周期同步）写入各个周期寄存器，同时也写入全局加载单触发位。

### 15.7.3 死区子模块的操作要点

死区子模块的配置选项如图所示。

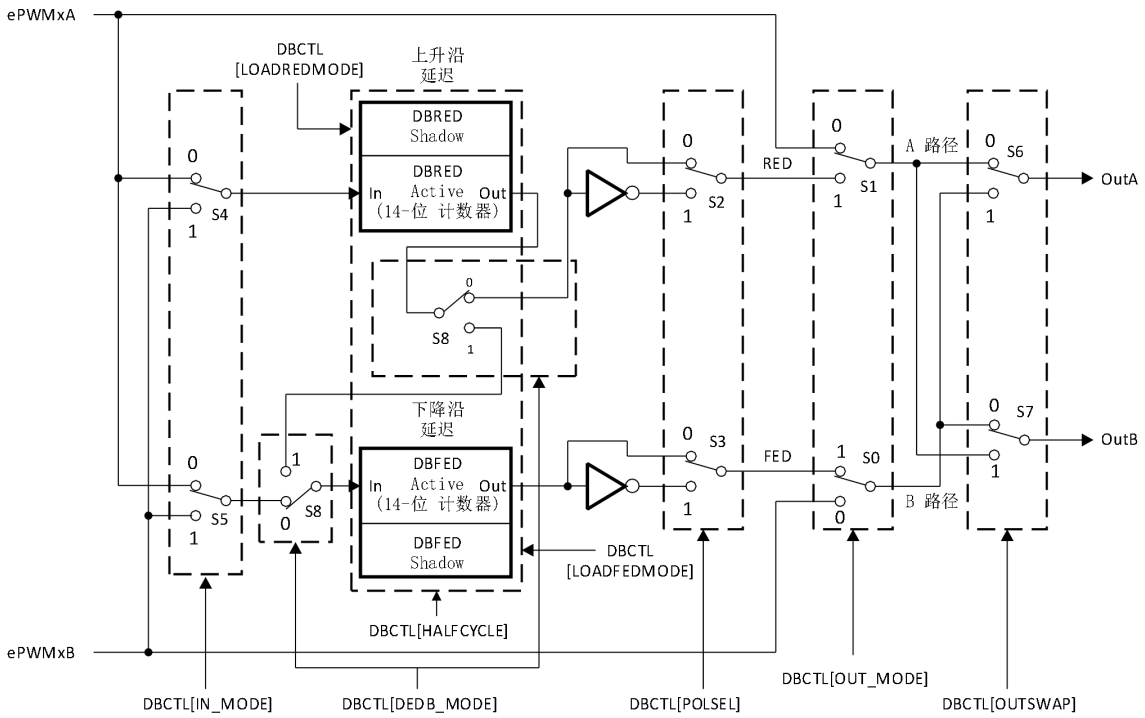



图 15-34 死区子模块的配置选项

尽管支持所有组合，但并非所有组合都是典型的使用模式。表中记录了一些传统的死区配置。这些模式假设 DBCTL [IN_MODE] 配置为 EPWMxA 输入作为下降沿和上升沿延迟的源。通过更改输入信号源，可以实现增强模式或非传统模式。表中显示的模式分为以下几类：

模式 1：旁路下降沿延迟（FED）和上升沿延迟（RED）：允许用户将死区子模块从 PWM 信号路径中完全禁用。

模式 2-5：传统死区极性设置：这些代表典型的极性配置，可以满足可用的工业功率开关栅极驱动器所需的所有高电平有效和低电平有效模式。这些典型情况的波形如图所示。

模式 6：旁路上升沿延迟（RED）和模式 7：旁路下降沿延迟（FED）：最后两个条目显示了旁路下降沿延迟（FED）或上升沿延迟（RED）块的组合。

图中显示了 占空比 时典型情况的波形。

表 15-10 传统死区操作模式

| 模式 | 模式描述 | DBCTL[POLSEL] | | DBCTL[OUT_MODE] | |
|---|---|---|---|---|---|
| | | S3 | S2 | S1 | S0 |
| 1 | EPWMxA 和 EPWMxB 直通（无延迟） | X | X | 0 | 0 |
| 2 | 高电平互补（AHC） | 1 | 0 | 1 | 1 |
| 3 | 低电平互补（ALC） | 0 | 1 | 1 | 1 |
| 4 | 高电平有效（AH） | 0 | 0 | 1 | 1 |
| 5 | 低电平有效（AL） | 1 | 1 | 1 | 1 |
| 6 | EPWMxA 输出 = EPWMxA 输入（无延迟）EPWMxB 输出 = EPWMxA 输入（带下降沿延迟） | 0 或 1 | 0 或 1 | 0 | 1 |
| 7 | EPWMxA 输出 = EPWMxA 输入（带上升沿延迟） | 0 或 1 | 0 或 1 | 1 | 0 |

表 15-11 附加死区操作模式

| 模式描述 | DBCTL[DEDB-MODE] | DBCTL[OUTSWAP] | |
|---|---|---|---|
| | S8 | S6 | S7 |
| EPWMxA 和 EPWMxB 信号如 OUT-MODE 位所定义。 | 0 | 0 | 0 |
| EPWMxA = 如 OUT-MODE 位所定义的 A 路径。EPWMxB = 如 OUT-MODE 位所定义的 A 路径（上升沿延迟或延迟旁路的 A 信号路径） | 0 | 0 | 1 |
| EPWMxA = 如 OUT-MODE 位所定义的 B 路径（下降沿延迟或延迟旁路的 B 信号路径）EPWMxB = 如 OUT-MODE 位所定义的 B 路径 | 0 | 1 | 0 |
| EPWMxA = 如 OUT-MODE 位所定义的 B 路径（下降沿延迟或延迟旁路的 B 信号路径）EPWMxB = 如 OUT-MODE 位所定义的 A 路径（上升沿延迟或延迟旁路的 A 信号路径） | 0 | 1 | 1 |
| 上升沿延迟仅应用于 A 信号路径上由 S4 开关（IN-MODE 位）选择的 EPWMxA / EPWMxB。下降沿延迟仅应用于 B 信号路径上由 S5 开关（IN-MODE 位）选择的 EPWMxA / EPWMxB。 | 0 | X | X |
| 上升沿延迟和下降沿延迟应用于由 S4 开关（IN-MODE 位）选择的源，并仅输出到 B 信号路径。(1) | 1 | X | X |

(1) 当此位设置为 1 时，用户始终可以要么设置 OUT_MODE 位使 Apath = InA，要么设置 OUTSWAP 位使 EPWMxA = Bpath。否则，EPWMxA 无效。

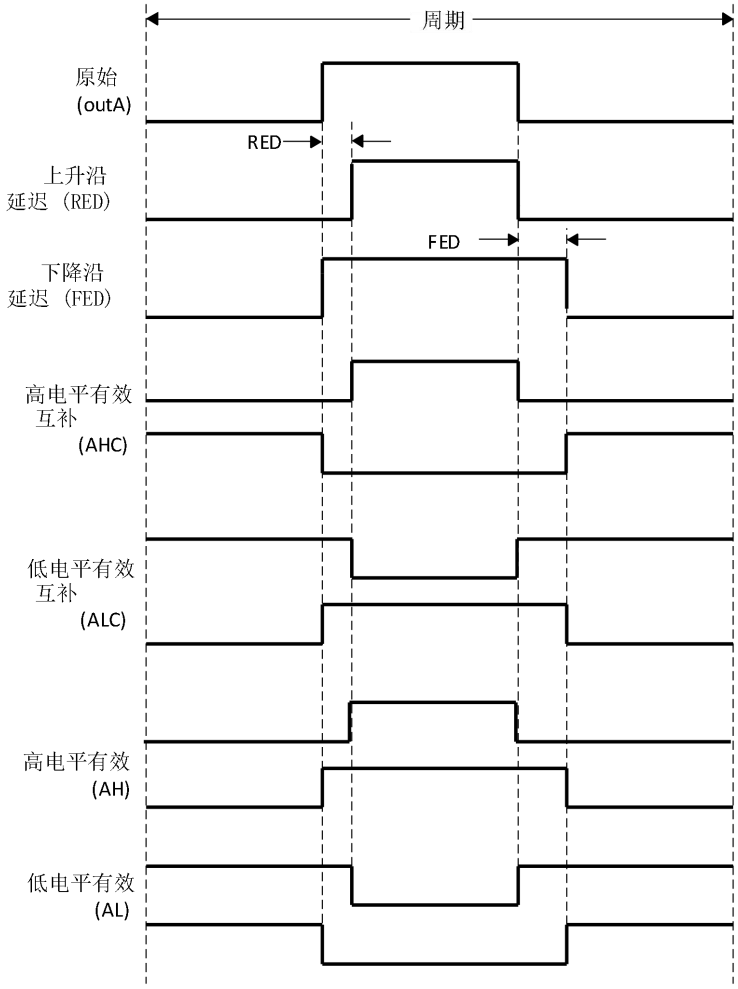



图 15-35 典型情况的死区波形（占空比）

死区子模块支持独立的上升沿（RED）和下降沿（FED）延迟值。延迟量通过 DBRED 和 DBFED 寄存器进行编程。这些是 10 位寄存器，其值表示信号边缘延迟的 TBCLK（时基时钟）脉冲数。例如，计算下降沿延迟和上升沿延迟的公式如下：

- FED=DBFED×TTBCLK
- RED=DBRED×TTBCLK

其中 TTBCLK 是 TBCLK 的周期，即 EPWMCLK 的预分频版本。

为方便起见，表中显示了各种 TBCLK 选项的延迟值。这些延迟值是基于 100MHz 的 ePWM 输入时钟频率计算的。

表 15-12 死区延迟值（以微秒为单位）与 DBFED 和 DBRED 的关系

| 死区值 | 死区延迟（微秒） | | |
|---|---|---|---|
| DBFED, DBRED | TBCLK = EPWMCLK/1 | TBCLK = EPWMCLK/2 | TBCLK = EPWMCLK/4 |

| 死区值 | 死区延迟（微秒） | | |
|---|---|---|---|
| 1 | 0.01 | 0.02 | 0.04 |
| 5 | 0.05 | 0.10 | 0.20 |
| 10 | 0.10 | 0.20 | 0.40 |
| 100 | 1.00 | 2.00 | 4.00 |
| 200 | 2.00 | 4.00 | 8.00 |
| 400 | 4.00 | 8.00 | 16.00 |
| 500 | 5.00 | 10.00 | 20.00 |
| 600 | 6.00 | 12.00 | 24.00 |
| 700 | 7.00 | 14.00 | 28.00 |
| 800 | 8.00 | 16.00 | 32.00 |
| 900 | 9.00 | 18.00 | 36.00 |
| 1000 | 10.00 | 20.00 | 40.00 |

启用半周期时钟时，计算下降沿延迟和上升沿延迟的公式变为：

FED=DBFED×TTBCLK/2

RED=DBRED×TTBCLK/2

## 15.8 PWM 斩波器（PC）子模块

PWM 斩波器子模块允许高频载波信号调制由动作限定器和死区子模块生成的 PWM 波形。如果需要基于脉冲变压器的栅极驱动器来控制功率开关元件，此功能非常重要。

图中展示了 ePWM 内部的 PWM 斩波器子模块。

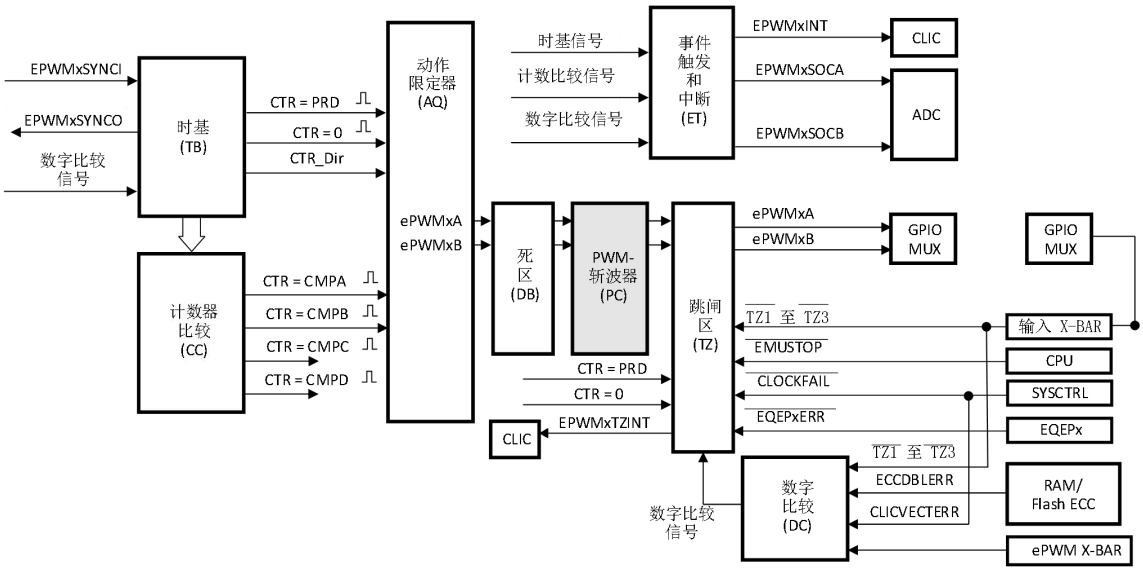



图 15-36 PWM 斩波器子模块

### 15.8.1 PWM 斩波器子模块的用途

PWM 斩波器子模块的关键功能包括：

可编程斩波（载波）频率。

可编程第一个脉冲的脉冲宽度。

可编程第二个及后续脉冲的占空比。

如果不需要，可以完全旁路。

### 15.8.2 PWM 斩波器子模块的操作要点

图中展示了 PWM 斩波器子模块的操作细节。载波时钟源自 EPWMCLK。时钟频率和占空比通过 PCCTL 寄存器中的 CHPFREQ 和 CHPDUTY 位进行控制。单触发块的功能是提供高能量的第一个脉冲，以确保功率开关快速可靠地导通，而后续脉冲则为维持脉冲，确保功率开关保持导通状态。单触发宽度通过 OSHTWTH 位进行编程。可以使用 CHPEN 位完全禁用（旁路）PWM 斩波器子模块。

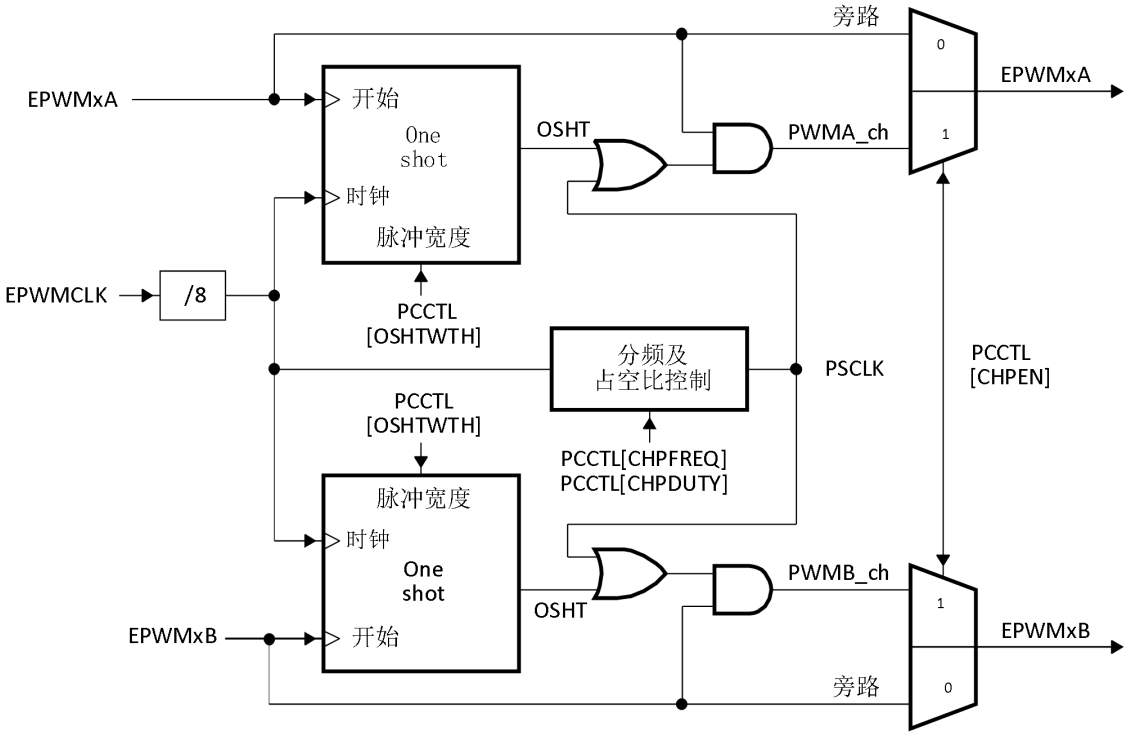



图 15-37 PWM 斩波器子模块的操作细节

### 15.8.3 波形

图中仅展示了斩波动作的简化波形；未显示单触发和占空比控制。单触发和占空比控制的细节在以下部分中讨论。

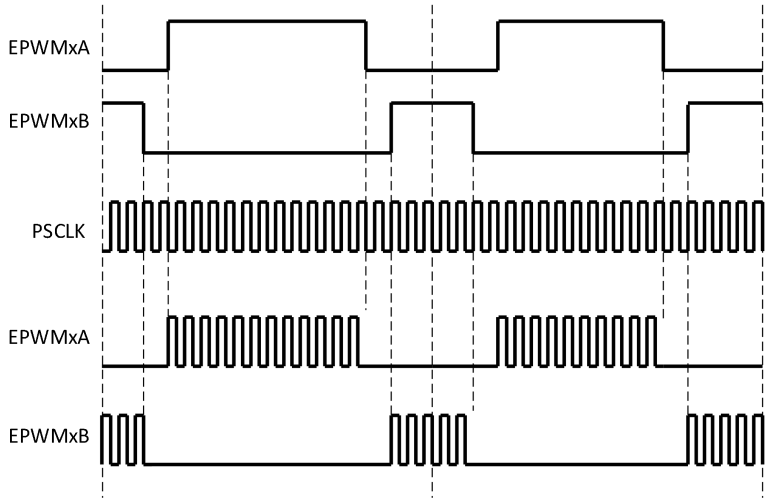



图 15-38 仅显示斩波动作的简单 PWM 斩波器子模块波形

#### 15.8.3.1 单触发脉冲

第一个脉冲的宽度可以编程为 16 种可能的脉冲宽度值之一。第一个脉冲的宽度或周期由下式给出：

T1stpulse=TEPWMCLK×8×OSHTWTH

其中 TEPWMCLK 是系统时钟（EPWMCLK）的周期，OSHTWTH 是四位控制位（值从 1 到 16）。

图中展示了第一个脉冲和后续的维持脉冲，表中给出了 EPWMCLK=80MHz 时可能的脉冲宽度值。

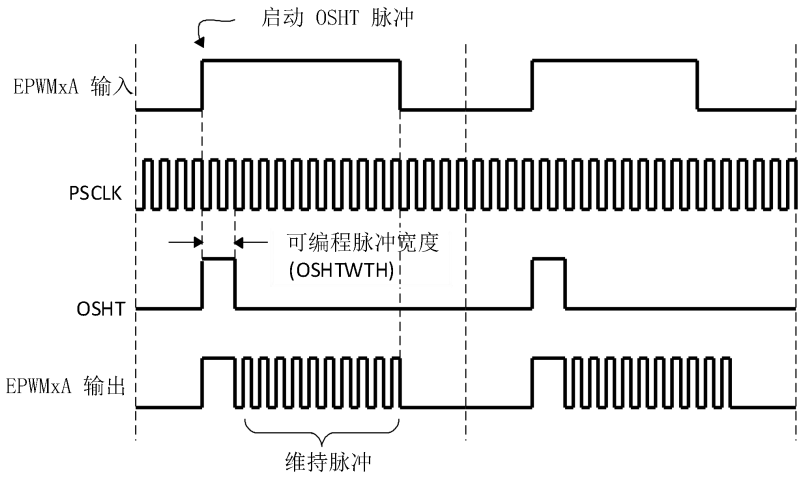



图 15-39 展示第一个脉冲和后续维持脉冲的 PWM 斩波器子模块波形

表 15-13 EPWMCLK = 80MHz 时可能的脉冲宽度值

| OSHTWTH（十六进制） | 脉冲宽度（纳秒） |
|---|---|
| 0 | 100 |
| 1 | 200 |
| 2 | 300 |
| 3 | 400 |
| 4 | 500 |
| 5 | 600 |
| 6 | 700 |
| 7 | 800 |
| 8 | 900 |
| 9 | 1000 |
| A | 1100 |
| B | 1200 |
| C | 1300 |
| D | 1400 |
| E | 1500 |
| F | 1600 |

### 15.8.3.2 占空比控制

基于脉冲变压器的栅极驱动器设计需要考虑变压器及相关电路的磁特性，磁饱和就是其中一个重要考量因素。为助力栅极驱动器设计，第二个及后续脉冲的占空比已实现可编程配置。这些维持脉冲能确保在功率开关导通期间，栅极始终保持正确的驱动强度和极性，因此通过软件控制可编程占空比，可对设计进行优化调整。

图中展示了通过编程 CHPDUTY 位实现的占空比控制功能。用户可选择 7 种不同的占空比，范围从 12.5% 到 87.5%。

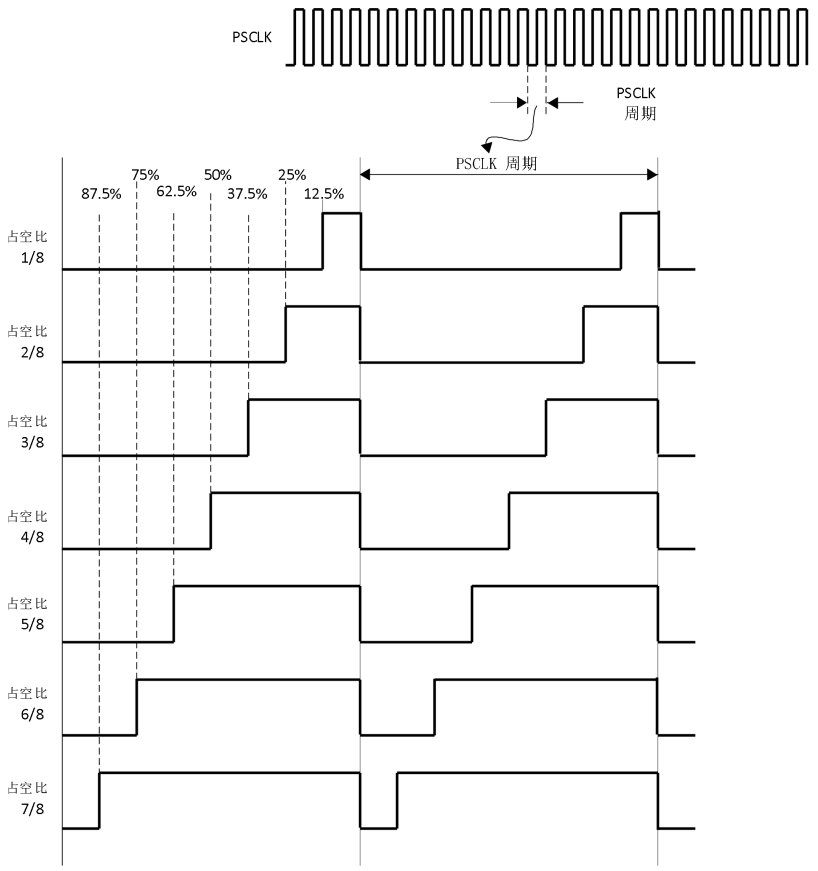



图 15-40 PWM 斩波器子模块波形（展示维持脉冲的脉冲宽度（占空比）控制）

## 15.9 跳闸区域（TZ）子模块

每个 ePWM 模块均与 6 个 TZn 信号（TZ1 至 TZ6）相连。其中，TZ1 至 TZ3 源自 GPIO 多路复用器；在配备 eQEP 模块的器件上，TZ4 源自反相的 EQEPxERR 信号；TZ5 与系统时钟故障逻辑相连；TZ6 源自 CPU 的 EMUSTOP 输出。这些信号用于指示外部故障或跳闸情况，当故障发生时，可通过编程设置 ePWM 输出的响应方式。

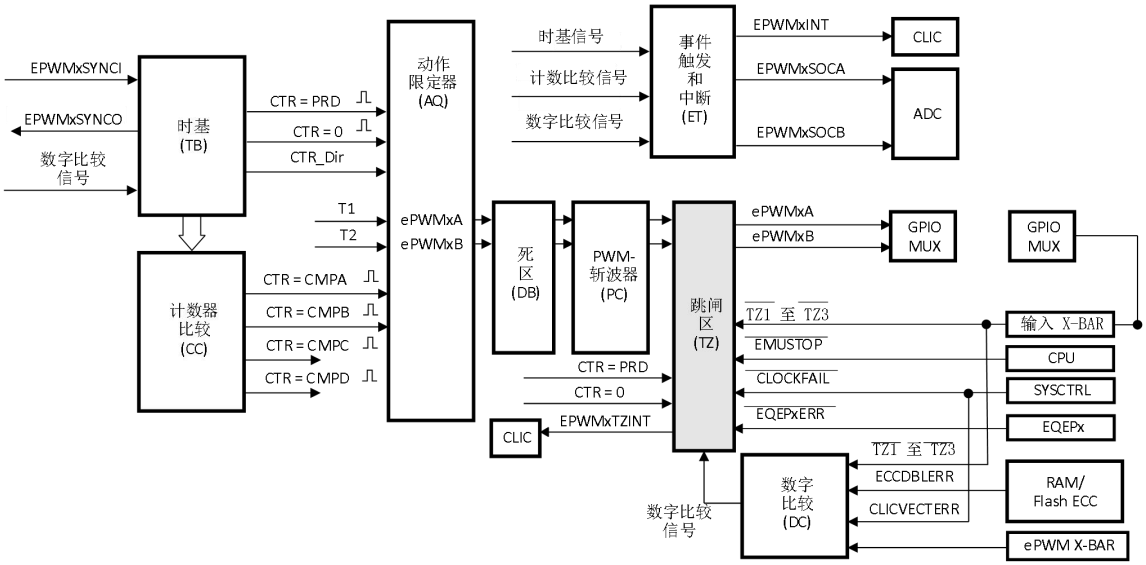



图 15-41 跳闸区域子模块

### 15.9.1 跳闸区域子模块的用途

跳闸区域子模块的核心功能包括：

跳闸输入 TZ1 至 TZ6 可灵活映射至任意 ePWM 模块。

发生故障时，输出 EPWMxA 和 EPWMxB 可被强制设置为以下状态之一：

高电平

低电平

高阻态

无动作（保持当前状态）

支持单触发跳闸（OSHT），适用于严重短路或过流等重大故障场景。

支持逐周期跳闸（CBC），适用于电流限制运行场景。

支持基于片上模拟比较器模块输出和 TZ1 至 TZ3 信号状态的数字比较跳闸（DC）功能。

每个跳闸区域输入和数字比较（DC）子模块的 DCAEVT1/2 或 DCBEVT1/2 强制事件均可配置为单触发或逐周期运行模式。

可在任意跳闸区域输入上生成中断。

支持软件强制跳闸功能。

若无需使用跳闸区域子模块，可将其完全旁路。

### 15.9.2 跳闸区域子模块的操作要点

以下章节将详细介绍跳闸区域子模块的操作要点和配置选项。跳闸区域信号 TZ1 至 TZ6（统称 TZn）为低电平有效输入信号。当其中任一信号变为低电平，或根据 TZDCSEL 寄存器的事件选择产生 DCAEVT1/2 或 DCBEVT1/2 强制事件时，即表示发生跳闸事件。每个 ePWM 模块可单独配置为忽略或使用各个跳闸区域信号或 DC 事件，具体由该 ePWM 模块的 TZSEL 寄存器决定。跳闸区域信号可通过 GPIO 多路复用器块同步至 ePWM 时钟（EPWMCLK）并进行数字滤波。TZn 输入上至少 3×TBCLK 的低脉冲宽度即可触发 ePWM 模块的故障状态；若脉冲宽度小于该值，逐周期（CBC）或单触发（OST）锁存器可能无法锁存跳闸状态。异步跳闸功能可确保在时钟缺失等情况下，TZn 输入上的有效事件仍能触发输出跳闸。需正确配置 GPIO 或外设。

每个 TZn 输入可单独配置为向 ePWM 模块提供逐周期或单触发跳闸事件。DCAEVT1 和 DCBEVT1 事件可配置为直接触发 ePWM 模块跳闸或提供单触发跳闸事件；同理，DCAEVT2 和 DCBEVT2 事件可配置为直接触发 ePWM 模块跳闸或提供逐周期跳闸事件。具体配置由 TZSEL [DCAEVT1/2]、TZSEL [DCBEVT1/2]、TZSEL [CBCn] 和 TZSEL [OSHTn] 控制位（其中 n 对应跳闸输入）决定。

（1）逐周期（CBC）

当发生逐周期跳闸事件时，会立即对 EPWMxA 和 EPWMxB 输出执行 TZCTL [TZA] 和 TZCTL [TZB] 位指定的动作，表中列出了部分可能的动作。通过适当配置 TZCTL2 寄存器中的位，可根据事件发生时计数器的计数方向（递增或递减）指定独立动作，仅当 TZCTL2 寄存器中的 ETZE 位置位时，TZCTL2 寄存器指定的动作才会生效。

此外，发生逐周期跳闸事件时，会置位逐周期跳闸事件标志（TZFLG [CBC]）；若在 TZEINT 寄存器和中断控制器中启用了该中断，则会生成 EPWMx_TZINT 中断。同时，TZCBCFLG 寄存器中也会置位与引发 CBC 事件对应的标志位。

若通过 TZEINT 寄存器启用了 CBC 中断，且通过 TZSEL 寄存器选择 DCAEVT2 或 DCBEVT2 作为 CBC 跳闸源，则无需在 TZEINT 寄存器中额外启用 DCAEVT2 或 DCBEVT2 中断，因为 DC 事件会通过 CBC 机制触发中断。

若跳闸事件不再存在，将根据 TZCLR [CBCPULSE] 的选择自动清除输入上的指定状态。因此，在该模式下，每个 PWM 周期都会清除或重置跳闸事件。TZFLG [CBC] 和 TZCBCFLG 标志位需通过向 TZCLR [CBC] 和 TZCBCCLR 标志位写入数据手动清除；若清除这些寄存器位时逐周期跳闸事件仍存在，则这些位会立即再次置位。

（2）单触发（OSHT）

当发生单触发跳闸事件时，会立即对 EPWMxA 和 EPWMxB 输出执行 TZCTL [TZA] 和 TZCTL [TZB] 位指定的动作，表中列出了部分可能的动作。通过适当配置 TZCTL2 寄存器中的位，可根据事件发生时计数器的计数方向（递增或递减）指定独立动作，仅当 TZCTL2 寄存器中的 ETZE 位置位时，TZCTL2 寄存器指定的动作才会生效。

此外，发生单触发跳闸事件时，会置位单触发跳闸事件标志（TZFLG [OST]）；若在 TZEINT 寄存器和中断控制器中启用了该中断，则会生成 EPWMx_TZINT 中断。同时，TZOSTFLG 寄存器中也会置位与引发 OST 事件对应的标志位。单触发跳闸状态需通过向 TZCLR [OST] 位写入数据手动清除；若需要，可通过向 TZOSTCLR 寄存器中的对应位写入数据手动清除 TZOSTFLG 寄存器位。

若通过 TZEINT 寄存器启用了单触发中断，且通过 TZSEL 寄存器选择 DCAEVT1 或 DCBEVT1 作为 OSHT 跳闸源，则无需在 TZEINT 寄存器中额外启用 DCAEVT1 或 DCBEVT1 中断，因为 DC 事件会通过 OSHT 机制触发中断。

注：确保 OST 的 TRIPIN 源变为非活动状态后，再清除 TZFLG 和 TZOSTFLG 标志。否则，若启用了中断，根据清除标志的时机，可能会发生 OST 中断，且 OST 标志会变为零。

（3）数字比较事件（DCAEVT1/2 和 DCBEVT1/2）

数字比较 DCAEVT1/2 或 DCBEVT2 事件由 TZDCSEL 寄存器选择的 DCAH/DCAL 和 DCBH/DCBL 信号组合生成。DCAH/DCAL 和 DCBH/DCBL 信号的源通过 DCTRIPSEL 寄存器选择，可为跳闸区域输入引脚或模拟比较器 CMPSSx 信号。

当发生数字比较事件时，会立即对 EPWMxA 和 EPWMxB 输出执行 TZCTL [DCAEVT1/2] 和 TZCTL [DCBEVT1/2] 位指定的动作，表中列出了可能的动作。通过适当配置 TZCTLDCA 和 TZCTLDCB 寄存器中的位，可根据事件发生时计数器的计数方向（递增或递减）指定独立动作，仅当 TZCTL2 寄存器中的 ETZE 位置位时，TZCTLDCA 和 TZCTLDCB 寄存器指定的动作才会生效。

此外，会置位相应的 DC 跳闸事件标志（TZFLG [DCAEVT1/2] / TZFLG [DCBEVT1/2]）；若在 TZEINT 寄存器和中断控制器中启用了该中断，则会生成 EPWMx_TZINT 中断。

当 DC 跳闸事件不再存在时，会自动清除引脚上的指定状态。TZFLG [DCAEVT1/2] 或 TZFLG [DCBEVT1/2] 标志位需通过向 TZCLR [DCAEVT1/2] 或 TZCLR [DCBEVT1/2] 位写入数据手动清除；若清除该标志时 DC 跳闸事件仍存在，则该标志会立即再次置位。

发生跳闸事件时执行的动作可通过 TZCTL、TZCTL2、TZCTLDCA 和 TZCTLDCB 寄存器位域为每个 ePWM 输出引脚单独配置。表中列出了跳闸事件发生时可执行的部分动作。

表 15-14 跳闸事件的可能动作

| TZCTL 寄存器位域设置 | EPWMxA 和 EPWMxB 状态 | 说明 |
|---|---|---|
| 0,0 | 高阻态 | 已跳闸 |
| 0,1 | 强制为高电平状态 | 已跳闸 |

| TZCTL 寄存器位域设置 | EPWMxA 和 EPWMxB 状态 | 说明 |
| --- | --- | --- |
| 1,0 | 强制为低电平状态 | 已跳闸 |
| 1,1 | 无变化 | 无动作。输出保持当前设置的电平不变。 |

示例 15-1. 跳闸配置示例

场景 A：

TZ1 上的单触发跳闸事件将 EPWM1A、EPWM1B 拉低，同时将 EPWM2A、EPWM2B 强制为高电平。

按以下方式配置 ePWM1 寄存器：

TZSEL [OSHT1] = 1：启用 TZ1 作为 ePWM1 的单触发事件源

TZCTL [TZA] = 2：跳闸事件发生时，EPWM1A 强制为低电平

TZCTL [TZB] = 2：跳闸事件发生时，EPWM1B 强制为低电平

按以下方式配置 ePWM2 寄存器：

TZSEL [OSHT1] = 1：启用 TZ1 作为 ePWM2 的单触发事件源

TZCTL [TZA] = 1：跳闸事件发生时，EPWM2A 强制为高电平

TZCTL [TZB] = 1：跳闸事件发生时，EPWM2B 强制为高电平

场景 B：

TZ5 上的逐周期事件将 EPWM1A、EPWM1B 拉低；TZ1 或 TZ6 上的单触发事件将 EPWM2A 置于高阻态。

按以下方式配置 ePWM1 寄存器：

TZSEL [CBC5] = 1：启用 TZ5 作为 ePWM1 的逐周期事件源

TZCTL [TZA] = 2：跳闸事件发生时，EPWM1A 强制为低电平

TZCTL [TZB] = 2：跳闸事件发生时，EPWM1B 强制为低电平

按以下方式配置 ePWM2 寄存器：

TZSEL [OSHT1] = 1：启用 TZ1 作为 ePWM2 的单触发事件源

TZSEL [OSHT6] = 1：启用 TZ6 作为 ePWM2 的单触发事件源

TZCTL [TZA] = 0：跳闸事件发生时，EPWM2A 置于高阻态

TZCTL [TZB] = 3：EPWM2B 忽略跳闸事件

注：配置 GPIO 和 INPUT X-BAR/EPWM X-BAR 选项时，需注意 X-BAR 输入选择的更改可能会引发意外事件。因此，应在启用 ePWM 跳闸区域之前，完成 GPIO 和 X-BAR 输入配置。若需要在 ePWM 跳闸区域启用时更改 GPIO/X-BAR 配置，用户可通过清除 TZSEL 寄存器关闭跳闸功能，待 INPUT XBAR 选择更改完成后，重新配置跳闸选择（TZSEL）。

### 15.9.3 生成跳闸事件中断

图中分别展示了跳闸区域子模块的控制逻辑和中断逻辑。

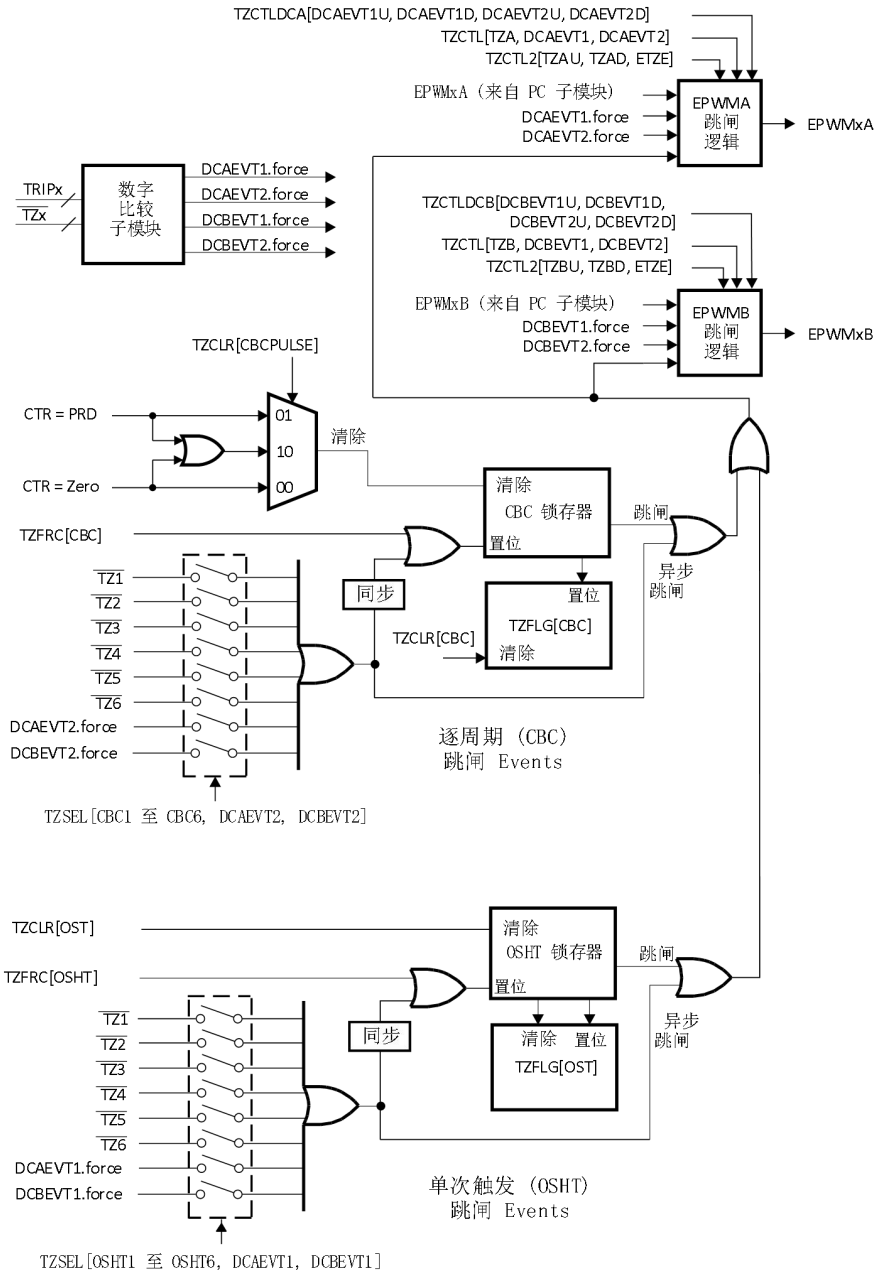



图 15-42 跳闸区域子模块模式控制逻辑

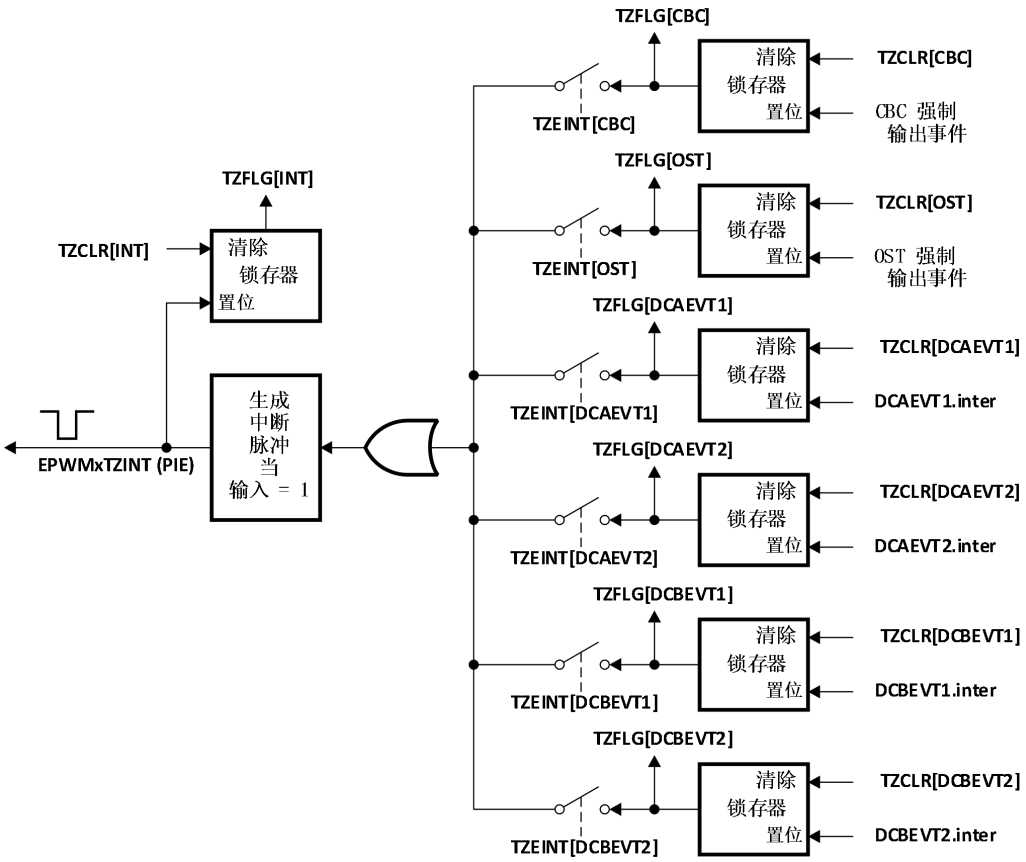



图 15-43 跳闸区域子模块中断逻辑

CBC、OST 和 DCxEVTy 的各个标志可用于检测 EPWMxTZINT 中断的来源。当多个源用于生成 EPWMxTZINT 中断时，读取和清除标志的操作会根据具体事件执行不同动作。

## 15.10 事件触发（ET）子模块

事件触发子模块的核心功能包括：

接收时基、计数器比较和数字比较子模块生成的事件输入。

利用时基方向信息对递增 / 递减事件进行限定。

利用预分频逻辑触发中断请求和 ADC 转换启动，支持：

每个事件触发一次

每两个事件触发一次

最多每十五个事件触发一次

通过事件计数器和标志提供事件生成的完整可见性。

支持软件强制触发中断和 ADC 转换启动。

事件触发子模块对时基子模块、计数器比较子模块和数字比较子模块生成的事件进行管理，当选中的事件发生时，向 CPU 生成中断并向 ADC 发送转换启动脉冲。

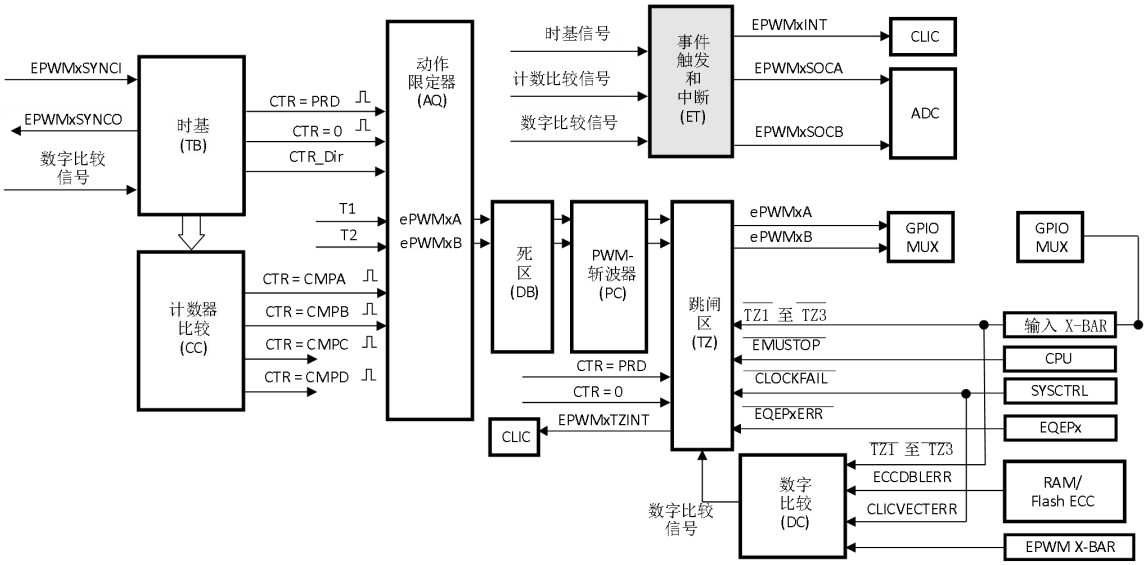



图 15-44 事件触发子模块

## 15.10.1 ePWM 事件触发子模块的操作概述

事件触发子模块监控各种事件条件（如图中左侧输入所示），并可配置为在触发中断请求或 ADC 转换启动之前对这些事件进行预分频。事件触发预分频逻辑支持以下触发方式：

- 每个事件触发一次
- 每两个事件触发一次
- 最多每十五个事件触发一次

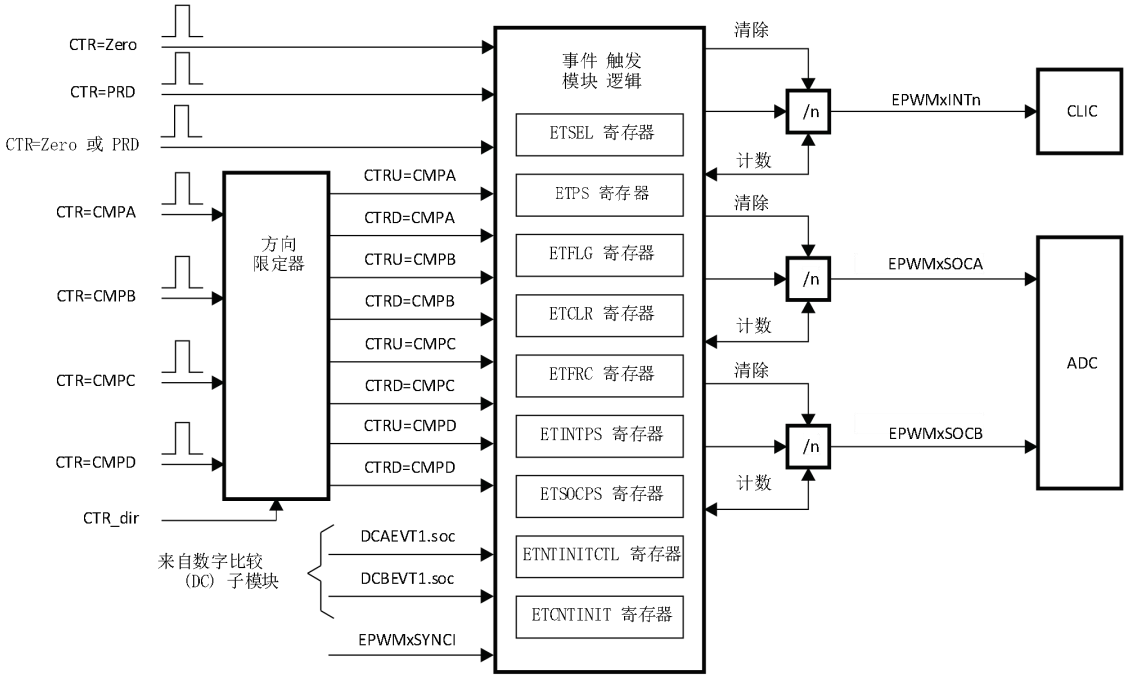



图 15-45 事件触发子模块（展示事件输入和预分频输出）

各寄存器功能说明如下：

ETSEL：选择触发中断或启动 ADC 转换的事件。

ETPS：编程上述事件预分频选项。

ETFLG：指示选中事件和预分频事件状态的标志位。

ETCLR：允许通过软件清除 ETFLG 寄存器中的标志位。

ETFRC：允许通过软件强制触发事件，适用于调试或软件干预场景。

ETINTPS：编程中断事件预分频选项，支持最多 15 个事件的计数和周期配置。

ETSOCPS：编程 SOC 事件预分频选项，支持最多 15 个事件的计数和周期配置。

ETCNTINITCTL：这些位启用通过 SYNC 事件或软件强制初始化 ETCNTINIT。

ETCNTINIT：这些位允许在 SYNC 事件（或软件强制）时，将 INT/SOCA/SOCB 计数器初始化为用户编程值。

中断周期（ETPS [INTPRD]）位指定生成中断脉冲所需的事件数量，可选配置如下：

不生成中断

每个事件生成一次中断

每两个事件生成一次中断

每三个事件生成一次中断

ETPS [INTPSSEL] 位的选择决定了是否由 ETINTPS 寄存器、INTCNT2 和 INTPRD2 位域决定事件频率（每 0-15 个事件中断一次）。

可通过中断选择（ETSEL [INTSEL]）和（ETSEL [INTSELCMP]）位配置引发中断的事件，事件可选如下：

时基计数器等于零（TBCTR = 0x00）

时基计数器等于周期值（TBCTR = TBPRD）

时基计数器等于零或周期值（TBCTR = 0x00 || TBCTR = TBPRD）

计时器递增时，时基计数器等于比较器 A 寄存器（CMPA）

计时器递减时，时基计数器等于比较器 A 寄存器（CMPA）

计时器递增时，时基计数器等于比较器 B 寄存器（CMPB）

计时器递减时，时基计数器等于比较器 B 寄存器（CMPB）

计时器递增时，时基计数器等于比较器 C 寄存器（CMPC）

计时器递减时，时基计数器等于比较器 C 寄存器（CMPC）

计时器递增时，时基计数器等于比较器 D 寄存器（CMPD）

计时器递减时，时基计数器等于比较器 D 寄存器（CMPD）

已发生的事件数量可根据 ETPS [INTPSSEL] 的选择，从中断事件计数器 ETPS [INTCNT] 或 ETINTPS [INTCNT2] 寄存器位读取。即当指定事件发生时，ETPS [INTCNT] 或 ETINTPS [INTCNT2] 位会递增，直至达到由 ETPS [INTPRD] 或 ETINTPS [INTPRD2]（再次由 ETPS [INTPSSEL] 的选择决定）指定的值。当 ETPS [INTCNT] = ETPS [INTPRD] 时，计数器停止计数并置位计数器输出；仅当向中断控制器发送中断时，计数器才会清除。

当 ETPS [INTCNT] 达到 ETPS [INTPRD] 时，会出现以下行为（该行为同样适用于 ETINTPS [INTCNT2] 和 ETINTPS [INTPRD2]）：

若启用了中断（ETSEL [INTEN] = 1）且中断标志位为清除状态（ETFLG [INT] = 0），则生成中断脉冲，置位中断标志位（ETFLG [INT] = 1），并清除事件计数器（ETPS [INTCNT] = 0），计数器重新开始对事件计数。

若禁用了中断（ETSEL [INTEN] = 0）或中断标志位为置位状态（ETFLG [INT] = 1），则当计数器达到周期值（ETPS [INTCNT] = ETPS [INTPRD]）时，计数器停止对事件计数。

若启用了中断，但中断标志位已置位，则计数器保持输出高电平，直至 ENTFLG [INT] 标志位被清除。这允许在处理一个中断时，挂起另一个中断。

向 INTPRD 位写入 0 会自动清除计数器（INTCNT = 0）并重置计数器输出（因此不生成中断）；向 INTPRD 写入其他值时，INTCNT 保留先前的值；INTCNT 溢出时会重置。向 ETFRC [INT] 位写入 1 会递增事件计数器 INTCNT；当 INTCNT = INTPRD 时，计数器的行为如前所述；当 INTPRD = 0 时，计数器被禁用，因此不检测事件，且 ETFRC [INT] 位也会被忽略（ETINTPS [INTCNT2] 和 ETINTPS [INTPRD2] 同理）。

根据上述定义，使用 INTCNT 和 INTPRD 可实现每个事件、每两个事件或每三个事件触发一次中断；使用 INTCNT2 和 INTPRD2 可实现每个事件最多每 15 个事件触发一次中断。

可基于 ETCNTINITCTL [INTINITEN] 中的选择，用 ETCNTINIT [INTINIT] 中的值初始化 INTCNT2 值。当 ETCNTINITCTL [INTINITEN] 置位时，由 ETCNTINITCTL [INTINITFRC] 决定是通过 SYNC 事件还是软件强制将 INTCNT2 计数器初始化为 ETCNTINIT [INTINIT] 的内容。

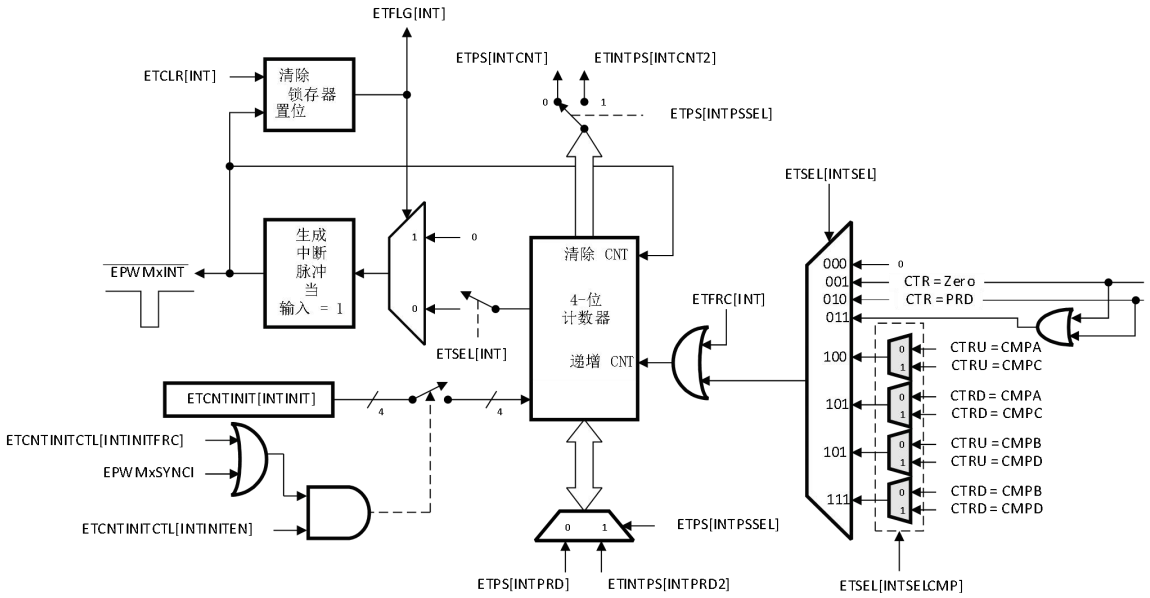



图 15-46 事件触发中断生成器

图中展示了事件触发的转换启动 A（SOCA）脉冲生成器的工作原理。增强功能包括：ETSEL 寄存器中定义的 SOCASELCMP 和 SOCBSELCMP 位域分别启用 CMPC 和 CMPD 事件以启动转换；ETPS [SOCPSSEL] 位域决定是否由 SOCACNT2 和 SOCAPRD2 控制。ETPS [SOCACNT] 计数器和 ETPS [SOCAPRD] 周期值的行为与中断生成器类似，但脉冲会持续生成。即生成脉冲时会锁存脉冲标志 ETFLG [SOCA]，但中断生成器不会停止后续脉冲生成。启用和禁用位 ETSEL [SOCAEN] 会停止脉冲生成，但输入事件仍会计数直至达到周期值（与中断生成逻辑一致）。触发 SOCA 和 SOCB 脉冲的事件可在 ETSEL [SOCASEL] 和 ETSEL [SOCBSEL] 位中分别配置，可选事件与中断生成逻辑的可选事件相同，此外还包括来自数字比较（DC）子模块的 DCAEVT1.soc 和 DCBEVT1.soc 事件信号。SOCACNT2 初始化方案与中断生成器非常相似，具有相应的启用、值初始化以及 SYNC 或软件强制选项。

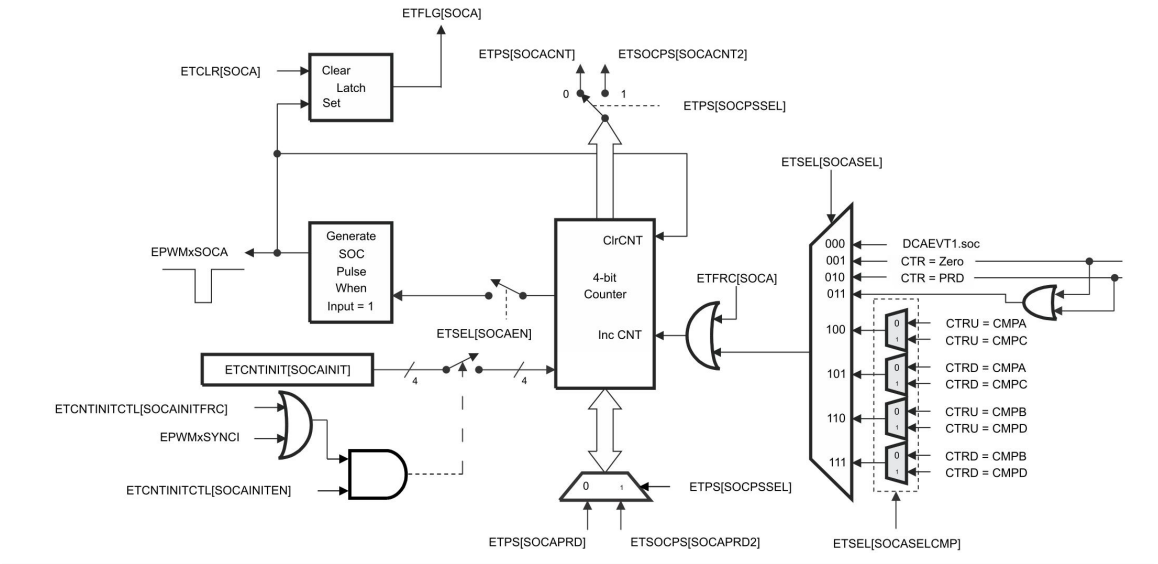



图 15-47 事件触发 SOCA 脉冲生成器

图中展示了事件触发的转换启动 B（SOCB）脉冲生成器的工作原理，其操作方式与 SOCA 相同。

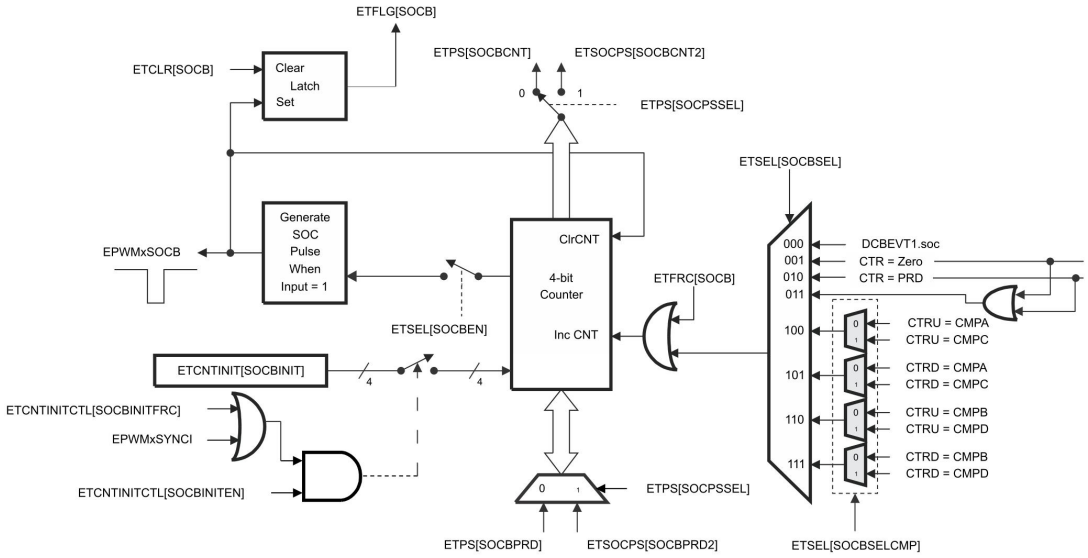



图 15-48 事件触发 SOCB 脉冲生成器

## 15.11 数字比较（DC）子模块

图中展示了数字比较（DC）子模块信号与 ePWM 系统中其他子模块的接口位置。

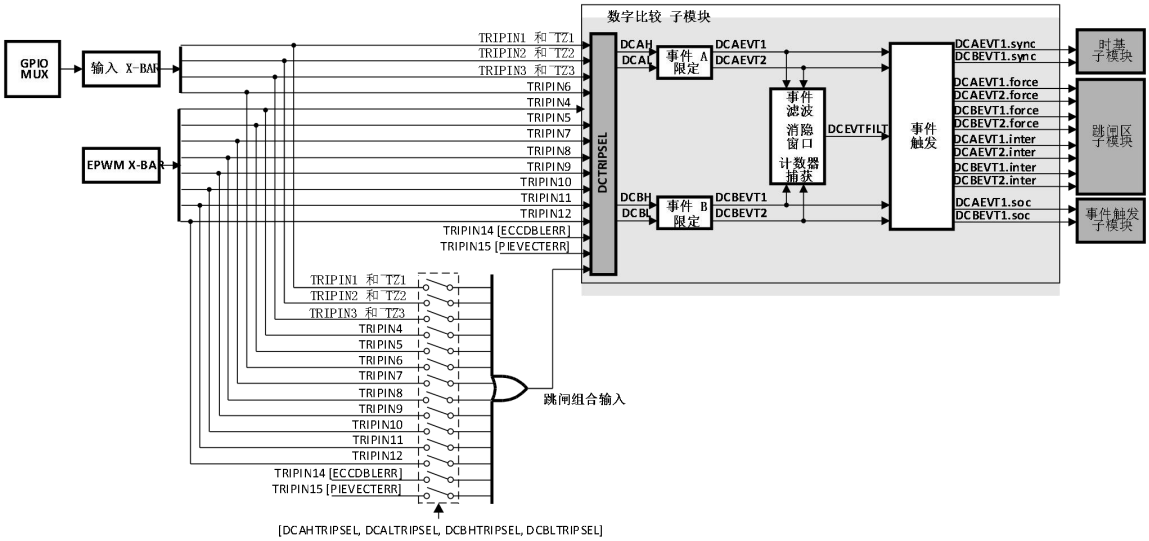



图 15-49 数字比较子模块高层框图

eCAP 输入信号源自 Input X-BAR 信号，如图所示。

在本器件上，任何 GPIO 引脚都可灵活映射为跳闸区域输入和跳闸输入，分别连接至跳闸区域子模块和数字比较子模块。Input X-BAR 输入选择（INPUTxSELECT）寄存器定义了哪些 GPIO 引脚被分配为跳闸区域输入 / 跳闸输入。

数字比较（DC）子模块对 ePWM 模块外部的信号（例如来自模拟比较器的 CMPSSx 信号）进行比较，直接生成 PWM 事件 / 动作，然后提供给事件触发、跳闸区域和时基子模块，具体功能包括：

- 生成跳闸区域中断
- 生成 ADC 转换启动信号
- 强制事件
- 生成同步事件，用于同步 ePWM 模块 TBCTR

此外，该子模块支持消隐窗口功能，可过滤 DC 事件信号中的噪声或干扰脉冲。

注：在启用时钟并为相应 ePWM 外设配置跳闸输入之前，用户需确保所选引脚上的状态正确，以避免跳闸信号的误锁存。

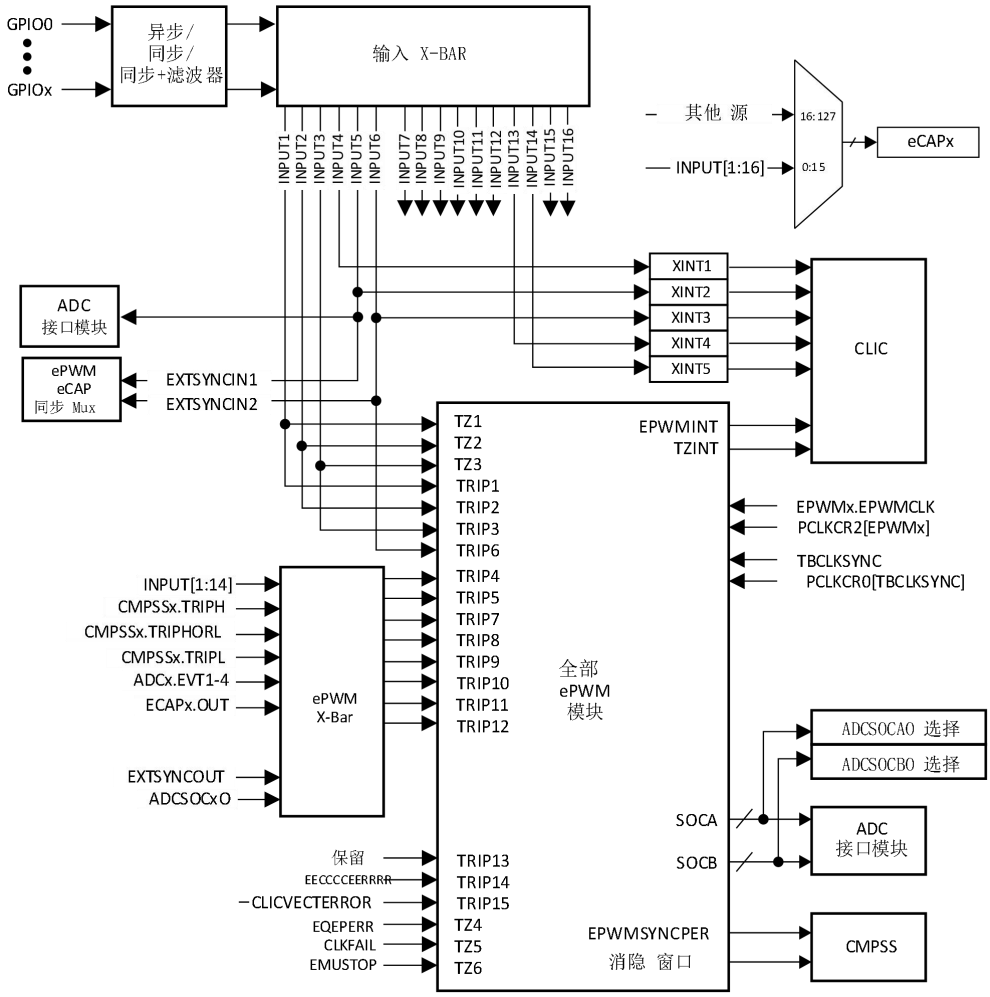



图 15-50 GPIO 多路复用器至跳闸输入的连接

## 15.11.1 数字比较子模块的用途

数字比较子模块的核心功能包括：

模拟比较器（COMP）模块输出通过 Input X-BAR、EPWM X-BAR、外部 GPIO 外设、中断控制器信号、ECC 错误信号以及 TZ1、TZ2 和 TZ3 输入，生成数字比较 A 高 / 低（DCAH、DCAL）和数字比较 B 高 / 低（DCBH、DCBL）信号。

DCAH/L 和 DCBH/L 信号触发事件，这些事件可经过滤波或直接应用于跳闸区域、事件触发和时基子模块，以实现以下功能：

- 生成跳闸区域中断
- 生成 ADC 转换启动信号

- 强制事件
- 生成同步事件，用于同步 ePWM 模块 TBCTR

事件滤波（消隐窗口逻辑）可选择性地消隐输入信号，以去除噪声。

### 15.11.2 使用 CMPSS 增强跳闸动作

为允许多个 CMPSS 同时影响 DCA/BEVTx 事件和跳闸动作，系统采用“或”逻辑将来自 ePWM 模块外部源的所有跳闸输入（最多 15 个）汇集起来，通过 DCTRIPSEL 寄存器作为“组合输入”提供给 DCAH、DCAL、DCBH 和 DCBL。具体配置方式为在 DCTRIPSEL 寄存器中选择“跳闸组合输入”（值为 0xF）。

用户可自主选择哪些跳闸输入通过组合逻辑生成 DCAH、DCAL、DCBH 和 DCBL 信号，具体通过 DCAHTRIPSEL、DCALTRIPSEL、DCBHTRIPSEL 和 DCBLTRIPSEL 寄存器选择实现。选中的组合输入会传递至 DCTRIPSEL 寄存器。

### 15.11.3 使用 CMPSS 逐周期触发 ePWM 跳闸

当使用 CMPSS 逐周期触发 ePWM 跳闸时，需采取措施防止一个 PWM 周期内的比较器跳闸状态断言延伸至下一个周期。CMPSS 可向下游 ePWM 模块发送跳闸状态信号。对于峰值电流模式控制等应用，每个 PWM 周期仅预期一次跳闸事件。在某些情况下，若未采取预防措施，持续的或晚期的跳闸事件（在 PWM 周期末尾发生）可能会延续至下一个 PWM 周期。如果 CMPSS 数字滤波器或 ePWM 数字比较（DC）子模块配置为对比较器跳闸信号进行限定，则 ePWM 跳闸逻辑需经过 N 个时钟周期的限定后，才能对跳闸信号的逻辑变化做出响应。一旦 ePWM 跳闸状态被限定，在比较器跳闸信号撤销后，该跳闸状态仍会保持 N 个时钟周期有效。若在 PWM 周期结束前 N 个时钟周期内，限定后的比较器跳闸信号仍处于断言状态，则跳闸状态需在下一个 PWM 周期开始后才能清除，因此新的 PWM 周期一开始就会检测到跳闸状态。

为避免这种不期望的跳闸状态，应用程序可采取以下措施，确保 ePWM 跳闸逻辑看到的限定跳闸信号在每个 PWM 周期结束前撤销：

设计系统时，确保在 PWM 周期结束前 N 个时钟周期内，比较器不会断言跳闸信号。

至少在 PWMSYNCPER 信号前两个时钟周期，使用 ePWM 事件滤波器激活比较器跳闸信号的消隐功能，并在下一个 PWM 周期内至少持续消隐 N 个时钟周期。

若使用 CMPSS COMPxLATCH 路径，则需在 PWM 周期结束前至少 N 个时钟周期清除 COMPxLATCH。可通过软件（使用 COMPSTSCLR）或生成早期 PWMSYNCPER 信号实现清除。本器件上的 ePWM 模块支持通过 HRPCTL 寄存器，在 CMPC 或 CMPD 匹配时生成 PWMSYNCPER，从而可在 PWM 周期内任意位置放置 PWMSYNCPER。

### 15.11.4 数字比较子模块的操作要点

以下章节将详细介绍数字比较子模块的操作要点和配置选项。

#### 15.11.4.1 数字比较事件

可通过 DCTRIPSEL 位选择跳闸区域输入（TZ1、TZ2 和 TZ3）和来自模拟比较器（COMP）模块的 CMPSSx 信号，生成数字比较 A 高 / 低（DCAH/L）和数字比较 B 高 / 低（DCBH/L）信号。随后，通过配置 TZDCSEL 寄存器对选中的 DCAH/L 和 DCBH/L 信号的动作进行限定，生成 DCAEVT1/2 和 DCBEVT1/2 事件（事件限定 A 和 B）。

当 TZn 信号用作 DCEVT 跳闸功能时，会被视为普通输入信号，可定义为高电平有效或低电平有效输入。当 TZn、DCAEVTx.force 或 DCBEVTx.force 信号有效时，ePWM 输出会异步跳闸。为使该状态保持锁存，需至少 3×TBCLK 的同步脉冲宽度；若脉冲宽度 ＜3×TBCLK 同步脉冲宽度，逐周期（CBC）或单触发（OST）锁存器可能无法锁存跳闸状态。

DCAEVT1/2 和 DCBEVT1/2 事件可经过滤波，生成滤波后的事件信号（DCEVTFILT），也可绕过滤波。滤波相关内容将在 “事件滤波” 部分进一步讨论。DCAEVT1/2 和 DCBEVT1/2 事件信号或滤波后的 DCEVTFILT 事件信号均可触发跳闸区域模块强制动作、TZ 中断、ADC SOC 或 PWM 同步信号。

强制信号：DCAEVT1/2.force 信号强制触发跳闸区域状态，可通过 TZCTL、TZCTLDCA、TZCTLDCB 寄存器配置直接影响 EPWMxA 引脚的输出；或者，若通过 TZSEL 寄存器选择 DCAEVT1/2 信号作为单触发或逐周期跳闸源，则 DCAEVT1/2.force 信号可通过 TZCTL 或 TZCTL2 寄存器配置影响跳闸动作。DCBEVT1/2.force 信号的行为类似，但影响的是 EPWMxB 输出引脚而非 EPWMxA 输出引脚。

TZCTL、TZCTL2、TZCTLDCA 和 TZCTLDCB 寄存器上冲突动作的优先级如下（优先级高的动作覆盖优先级低的动作）：

输出 EPWMxA：TZA（最高）→ DCAEVT1 → DCAEVT2（最低）；TZAU（最高）→ DCAEVT1U → DCAEVT2U（最低）；TZAD（最高）→ DCAEVT1D → DCAEVT2D（最低）

输出 EPWMxB：TZB（最高）→ DCBEVT1 → DCBEVT2（最低）；TZBU（最高）→ DCBEVT1U → DCBEVT2U（最低）；TZBD（最高）→ DCBEVT1D → DCBEVT2D（最低）

中断信号：DCAEVT1/2.interrupt 信号向中断控制器生成跳闸区域中断。需在 TZEINT 寄存器中置位 DCAEVT1、DCAEVT2、DCBEVT1 或 DCBEVT2 位以启用中断。一旦发生上述任一事件，将触发 EPWMxTZINT 中断，且需置位 TZCLR 寄存器中的对应位以清除中断。

SOC 信号：DCAEVT1.soc 信号与事件触发子模块接口，可通过 ETSEL [SOCASEL] 位选择作为生成 ADC 转换启动 A（SOCA）脉冲的事件；同理，

DCBEVT1.soc 信号可通过 ETSEL [SOCBSEL] 位选择作为生成 ADC 转换启动 B（SOCB）脉冲的事件。

同步信号：DCAEVT1.sync 和 DCBEVT1.sync 事件与 EPWMxSYNCI 输入信号和 TBCTL [SWFSYNC] 信号进行“或”运算，生成时基计数器的同步脉冲。

注：图中展示了 DCxEVT1、DCxEVT2 或 DCEVTFLT 信号如何处理生成数字比较 A 和 B 事件的强制信号、中断信号、SOC 信号和同步信号。

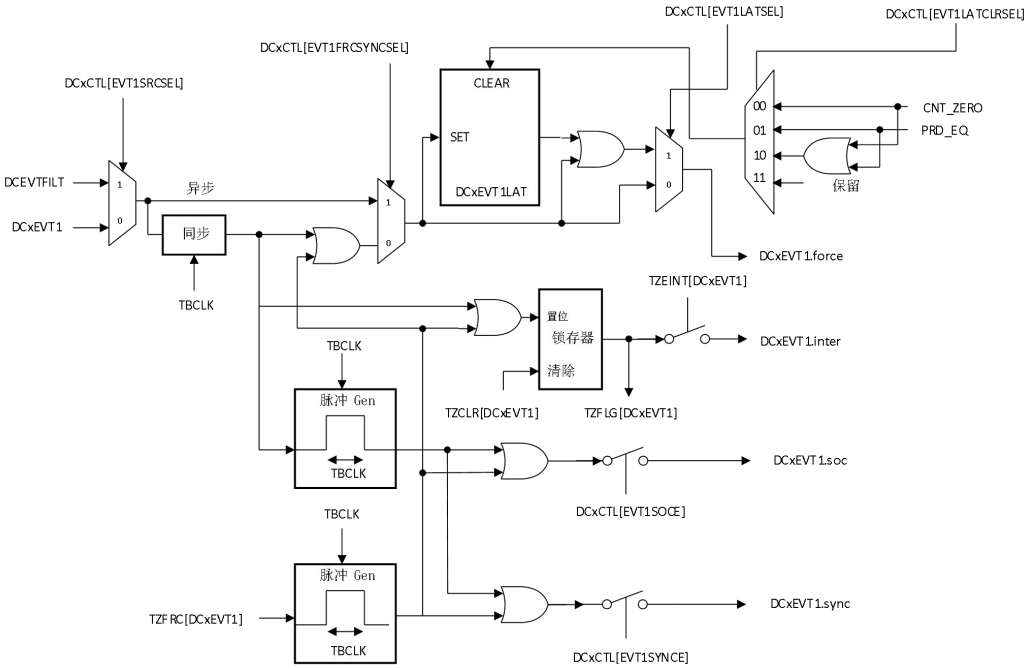



图 15-51 DCxEVT1 事件触发

在移相全桥（PSFB）转换器等应用中，可能需要对逐周期（CBC）跳闸事件和单触发（OST）跳闸事件执行不同动作，这可通过 DCxEVT1LAT 实现。

可根据 DCxCTL.EVTy.LATCLRSEL 的设置，在 CNT=0、CTR=PRD 或 CNT=0 或 CTR=PRD 事件时清除该锁存器，这与逐周期（CBC）锁存器清除机制类似。

可根据 DCxCTL.EVTyLATSEL 的值，选择 DCxEVTy.force 信号为锁存版本或非锁存版本。

可通过读取 DCxCTL.EVTyLAT 字段访问 DCxEVTyLAT 信号的状态。

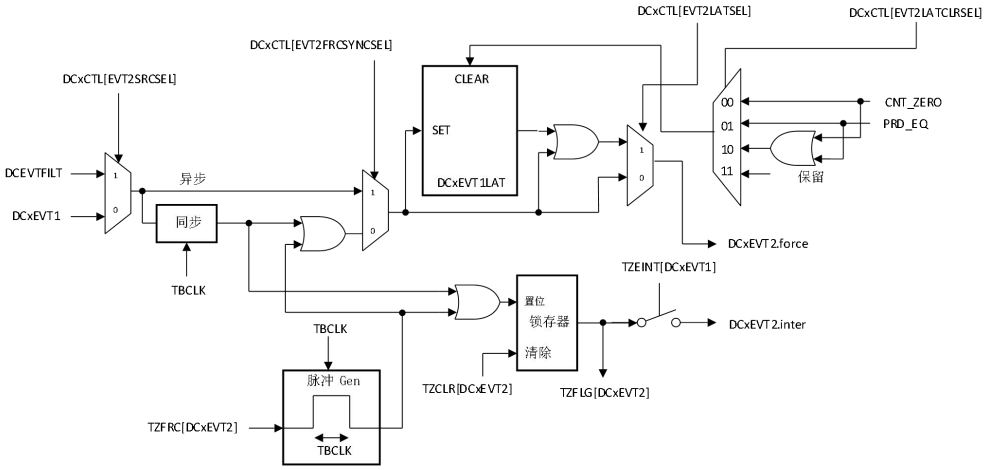



图 15-52 DCxEVT2 事件触发

### 15.11.4.2 事件滤波

消隐控制逻辑：DCAEVT1/2 和 DCBEVT1/2 事件可通过事件滤波逻辑进行滤波，通过选择性地消隐事件一段时间来去除噪声。该功能适用于以下场景：模拟比较器输出可选择触发 DCAEVT1/2 和 DCBEVT1/2 事件，消隐控制逻辑用于在触发 PWM 输出跳闸、生成中断或 ADC 转换启动前过滤信号上的潜在噪声。消隐控制逻辑用于定义消隐窗口，在窗口有效期间忽略信号上的所有事件。消隐窗口通过 DCFCTL、DCFOFFSET 和 DCFWINDOW 寄存器配置：DCFCTL 寄存器启用消隐窗口，并将消隐窗口与 CTR=PRD 脉冲、CTR=0 脉冲或 CTR=PRD 和 CTR=0 脉冲对齐（由 DCFCTL [PULSESEL] 指定）；DCFCTL [SRCSEL] 选择 DCEVTFILT 信号的 DCxEVTy 事件源；应用程序在 DCFOFFSET 寄存器中编程 TBCLK 计数的偏移值，该值决定了在 CTR=PRD 或 CTR=0 脉冲后多久开始消隐窗口；消隐窗口的持续时间（偏移计数器过期后的 TBCLK 计数数量）由应用程序写入 DCFWINDOW 寄存器。在消隐窗口结束前后，事件仍可像之前一样生成 SOC、同步、中断和强制信号。图中展示了事件滤波逻辑的细节。

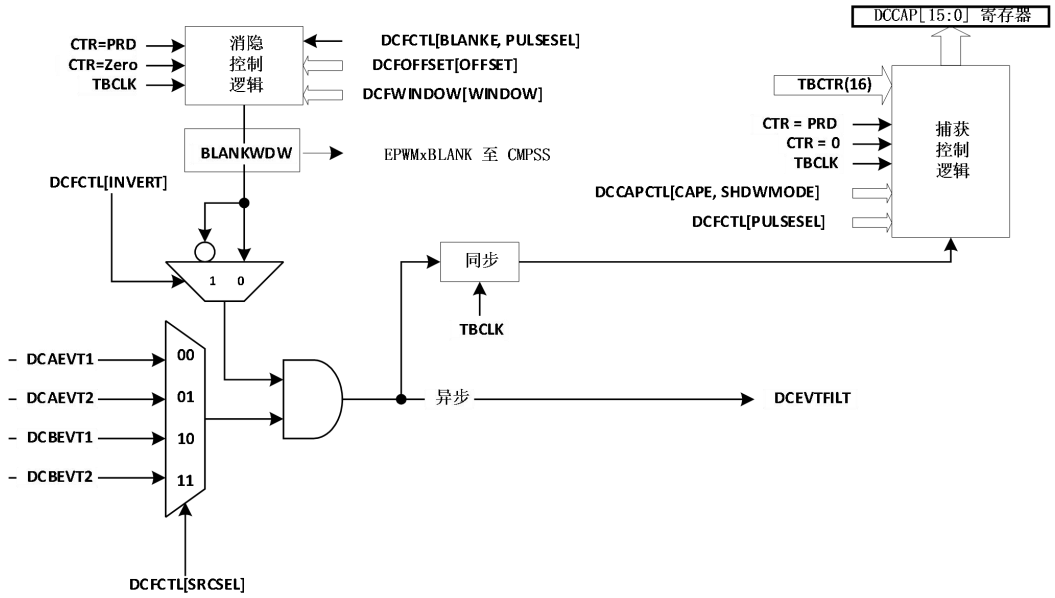



图 15-53 事件滤波

捕获控制逻辑：启用捕获控制逻辑后，选中的 DCxEVTy 事件（由 DCCAPCTL 寄存器配置）会触发将 TBCTR 捕获到活动寄存器。CPU 直接从活动寄存器读取数据，除非通过 DCCAPCTL [SHDWMODE] 启用了影子模式。启用影子模式后，活动寄存器信息会在 DCFCCTL [PULSESEL] 指定的事件上复制到影子寄存器，CPU 从影子寄存器读取数据。选中 DCxEVTy 事件后，需等待 DCFCCTL [PULSESEL] 定义的事件发生或 DCCAPCTL [CAPSTS] 处的比较事件标志通过 DCCAPCTL [CAPCLR] 清除后，才能发生后续捕获事件。

注：必须正确配置 ePWM 消隐窗口，确保跳闸输入在消隐窗口过期后至少保持 3 个 ePWM 周期有效。

图中展示了 ePWM 周期内偏移和消隐窗口的几种时序情况。需注意，若消隐窗口跨越 CTR=0 或 CTR=PRD 边界，则下一个窗口仍在 CTR=0 或 CTR=PRD 脉冲后的相同偏移值处开始。

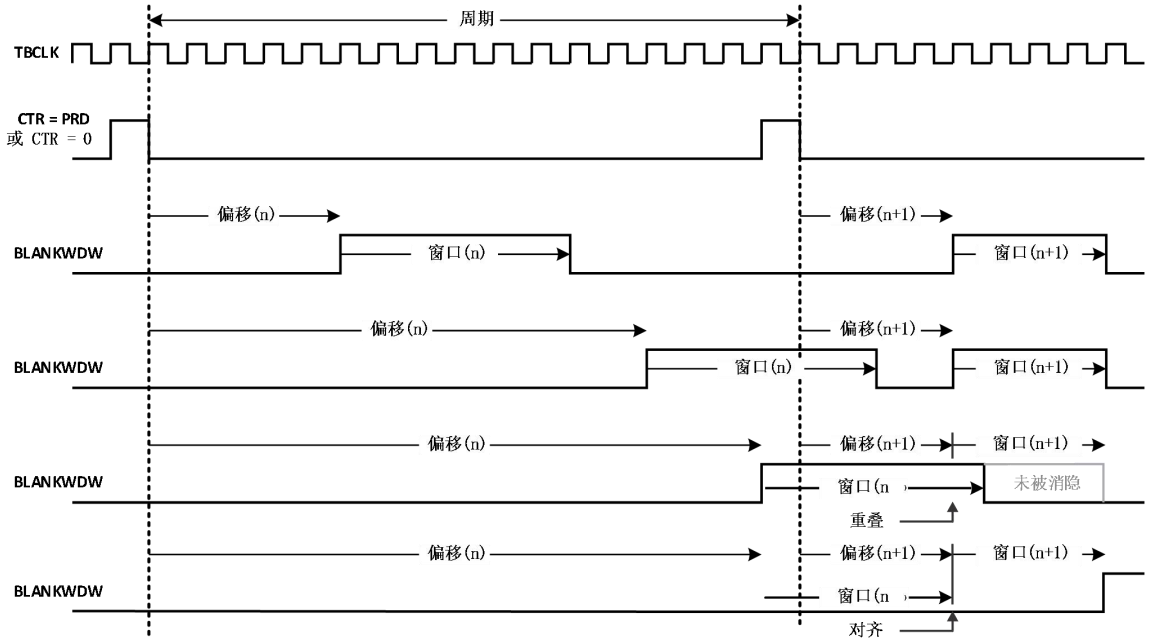



图 15-54 消隐窗口时序图

## 15.11.5 谷值切换

事件滤波功能与“事件滤波”部分描述的事件滤波逻辑共同实现谷值切换功能。该功能无需额外外部电路即可实现可编程谷值切换，模块提供片上硬件机制，可实现以下功能：

捕获振荡周期。

精确延迟 PWM 切换时刻。

允许在延迟生效前设置可编程的边沿数量。

提供多种触发和事件选择。

便于在系统 / 工作条件变化时灵活调整以实现最佳性能。

需对 DCxEVTy 信号进行进一步处理以支持谷值切换，以下简要描述启用谷值切换功能的步骤：

通过 DCFCTL [SRCSEL] 选择一个 DCxEVTy 事件作为谷值切换模块的输入（可选择添加消隐窗口（消隐控制逻辑）），即选择上述比较器输出（或外部输入）作为谷值切换模块的输入。

通过边沿选择逻辑（DCFCTL [EDGEMODE, EDGECOUNT]）配置边沿滤波器，以捕获 “n” 个上升沿、下降沿或两者。

选择用于重置和重启边沿滤波器的正确事件（VCAPCTL [TRIGSEL]），边沿捕获事件由该选中的边沿触发或激活。

启用谷值捕获逻辑（VCAPCTL [VCAPE]）。

选择指示振荡周期测量捕获开始的起始边沿（VCNTCFG [STARTEDGE]），此时 16 位计数器开始计数。

选择指示 16 位计数器停止计数的停止边沿（VCNTCFG [STOPEDGE]），捕获的计数器值（CNTVAL）提供振荡周期信息。

停止边沿值必须始终大于起始边沿值。

配置并将捕获的延迟（CNTVAL）应用于经过边沿滤波的 DCxEVTy 信号。CNTVAL 值可直接应用，或与软件编程值（用于偏移调整）（SWVDELVAL）结合应用，也可仅应用延迟的一部分（可选择是否结合 SWVDELVAL），这有助于正确应用与谷值点对应的延迟（VCAPCTL [VDELAYDIV]）。

配置 VCAPCTL [EDGEFILTDLYSEL]，以基于上述捕获的值应用硬件延迟。

计数器停止后，计数器值会复制到 CNTVAL 寄存器，计数器重置为零；需等待 VCAPCTL [TRIGSEL] 选中的事件再次发生，才能进行下一次捕获。本实现中，软件触发用作 VCAPCTL [TRIGSEL] 的源；触发事件发生时，无论计数器当前状态如何，计数器都会重置，并在起始边沿发生时从零开始计数；同理，触发事件发生时，边沿滤波器会重置，并在起始边沿发生时从零开始计数。

谷值切换模块的输出（DCEVTFILT）随后用于同步 PWM 时基，该过程如图所示。

注：Ware 中提供了展示谷值切换硬件和软件使用方法的特定应用示例。

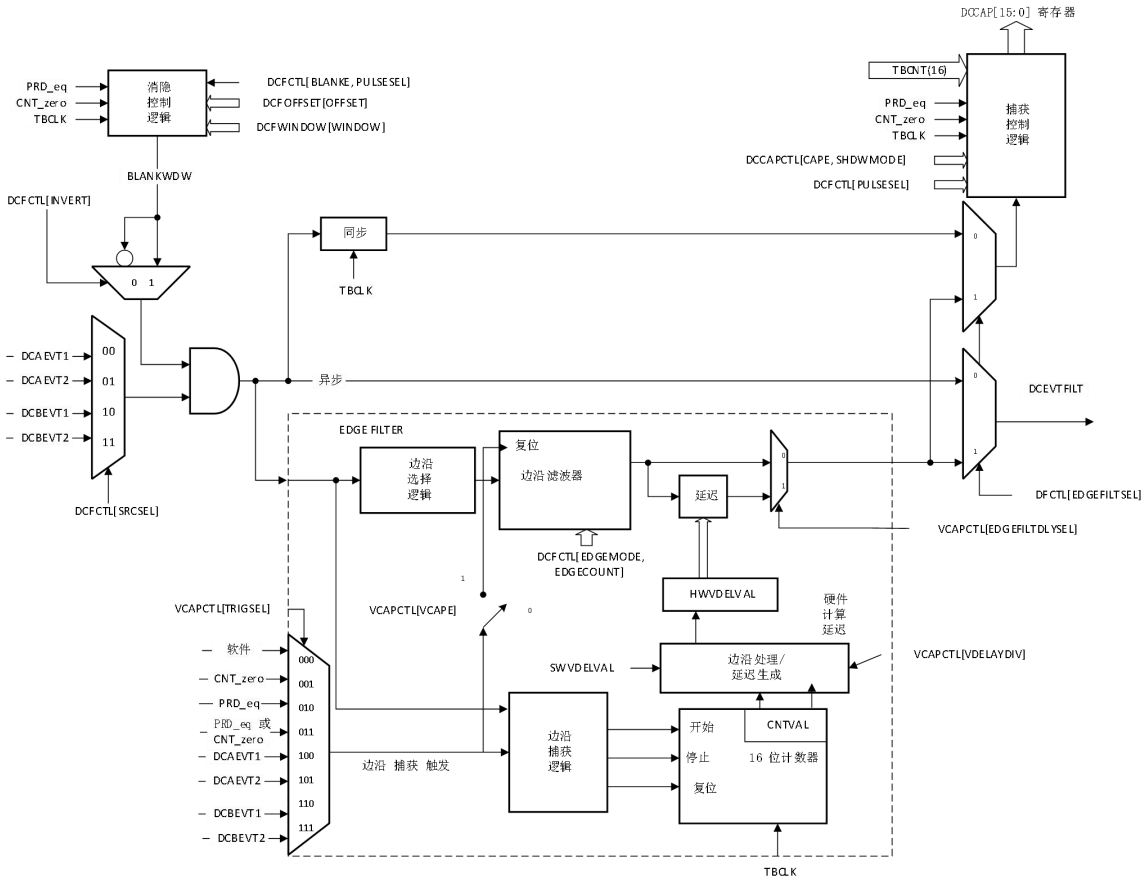



图 15-55 谷值切换

## 15.12 ePWM 交叉开关（X-BAR）

图中展示了 ePWM 交叉开关（X-BAR）的架构。该模块支持选择各种触发源连接至 8 个专用 EPWM 跳闸输入，即 TRIP4、TRIP5、TRIP7、TRIP8、TRIP9、TRIP10、TRIP11 和 TRIP12。

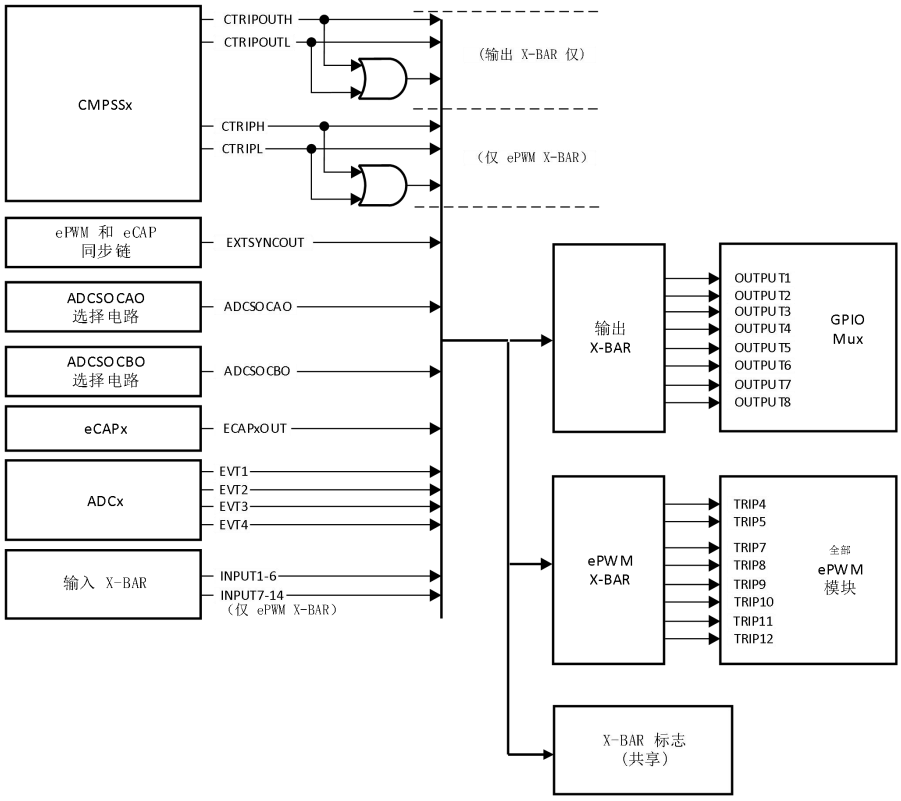



图 15-56 ePWM X-BAR

# 15.13 功率拓扑结构的应用

ePWM 模块具备独立运行所需的全部本地资源，也可与其他相同的 ePWM 模块同步运行。

## 15.13.1 多模块概述

前文均围绕单个模块的运行展开说明。为便于理解多模块在系统中的协同工作原理，下文将采用图中所示的简化框图来表示本手册中描述的 ePWM 模块。该简化的 ePWM 框图仅展示了说明多开关功率拓扑结构如何通过多个 ePWM 模块协同控制所需的关键资源。

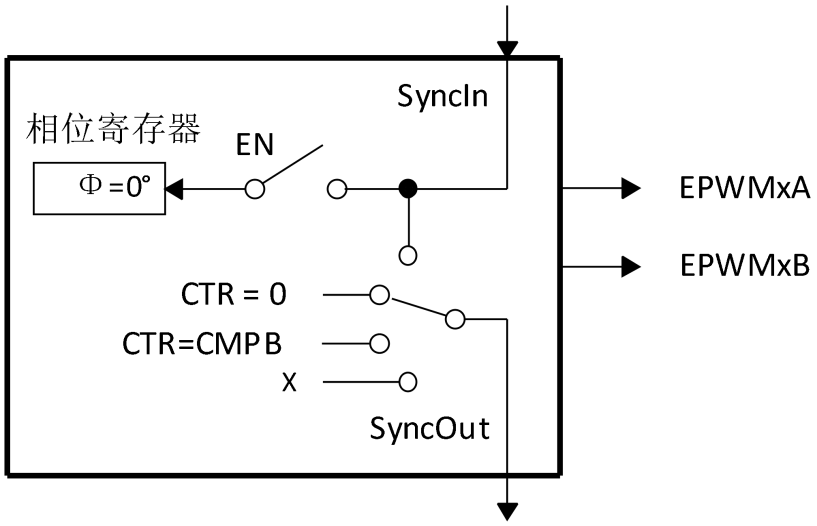



图 15-57 简化的 ePWM 模块

## 15.13.2 关键配置功能

每个模块可配置的关键选项如下：

（1）同步输入（SyncIn）选项

闭合使能（EN）开关时，在输入同步选通信号到来时，用相位寄存器加载自身计数器。

断开使能开关，不执行任何操作或忽略输入同步选通信号。

同步直通 —— 同步输出（SyncOut）与同步输入（SyncIn）相连。

主模式，在 PWM 边界处提供同步信号 —— 同步输出与计数器等于周期（CTR = PRD）事件相连。

主模式，在任意可编程时刻提供同步信号 —— 同步输出与计数器等于比较器 B（CTR = CMPB）事件相连。

模块工作在独立模式，不向其他模块提供同步信号 —— 同步输出接无效状态（X）（禁用）。

（2）同步输出（SyncOut）选项

同步直通 —— 同步输出与同步输入相连。

主模式，在 PWM 边界处提供同步信号 —— 同步输出与计数器等于周期（CTR = PRD）事件相连。

主模式，在任意可编程时刻提供同步信号 —— 同步输出与计数器等于比较器 B（CTR = CMPB）事件相连。

模块工作在独立模式，不向其他模块提供同步信号 —— 同步输出接无效状态（X）（禁用）。

对于每种同步输出选择，模块还可选择在同步输入选通信号到来时加载新的相位值，或忽略该值（即通过使能开关控制）。尽管存在多种配置组合，但最常见的两种模式 —— 主模块模式和从模块模式如图所示。

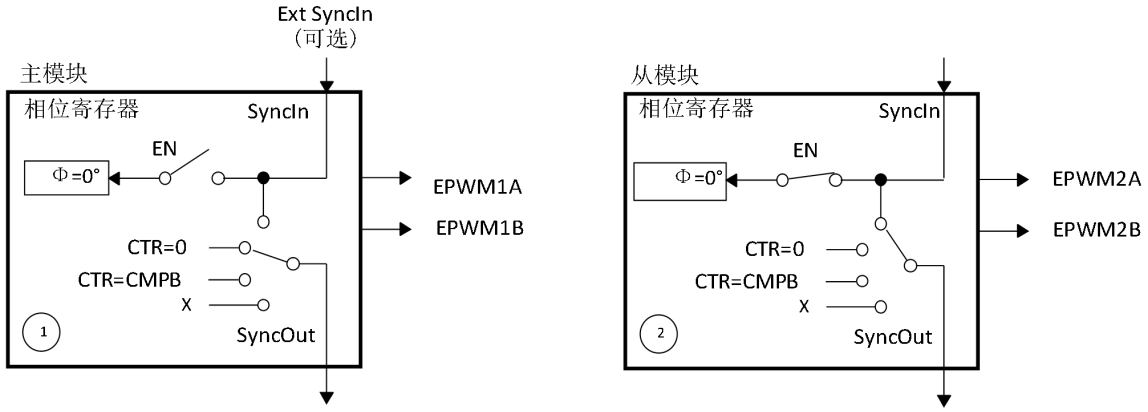



图 15-58 典型主模块配置的 ePWM1 和从模块配置的 ePWM2

### 15.13.3 控制多个具有独立频率的降压转换器

最简单的功率转换器拓扑结构之一是降压转换器。单个配置为主模式的 ePWM 模块可控制两个工作在相同 PWM 频率的降压级。若每个降压转换器需要独立的频率控制，则需为每个转换器级分配一个 ePWM 模块。图中展示了四个降压级，每个降压级均运行在独立频率下。在此情况下，所有四个 ePWM 模块均配置为主模式，且不使用同步功能。请注意，尽管有四个降压级，但仅展示了三个波形。

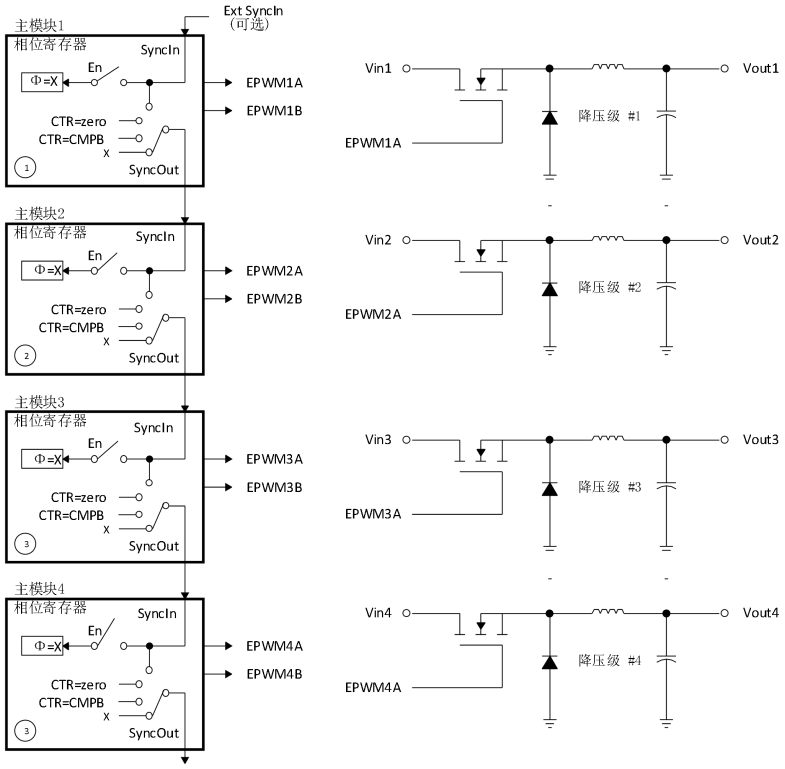



图 15-59 四个降压级的控制。此处 F_PWM1≠F_PWM2≠F_PWM3≠F_PWM4

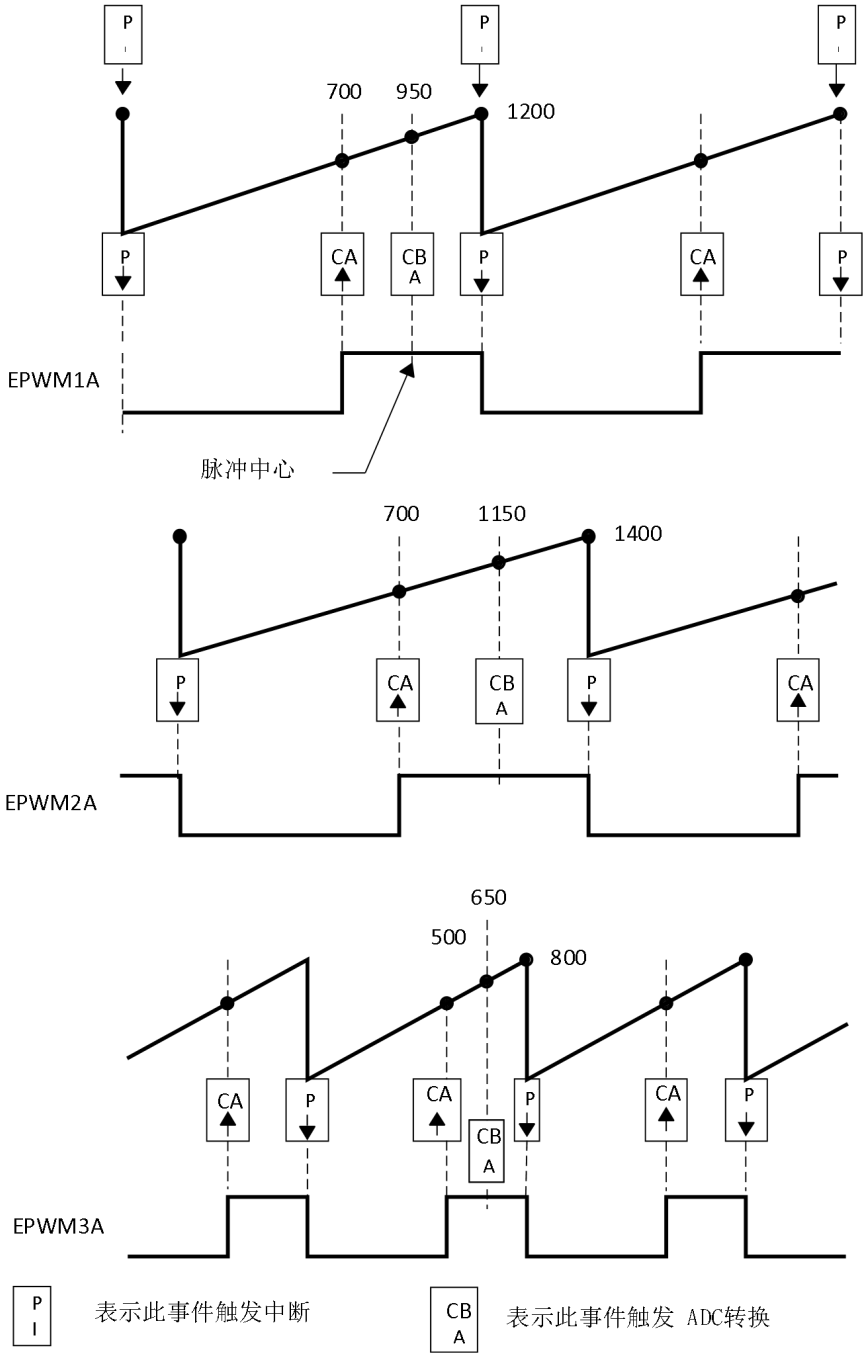
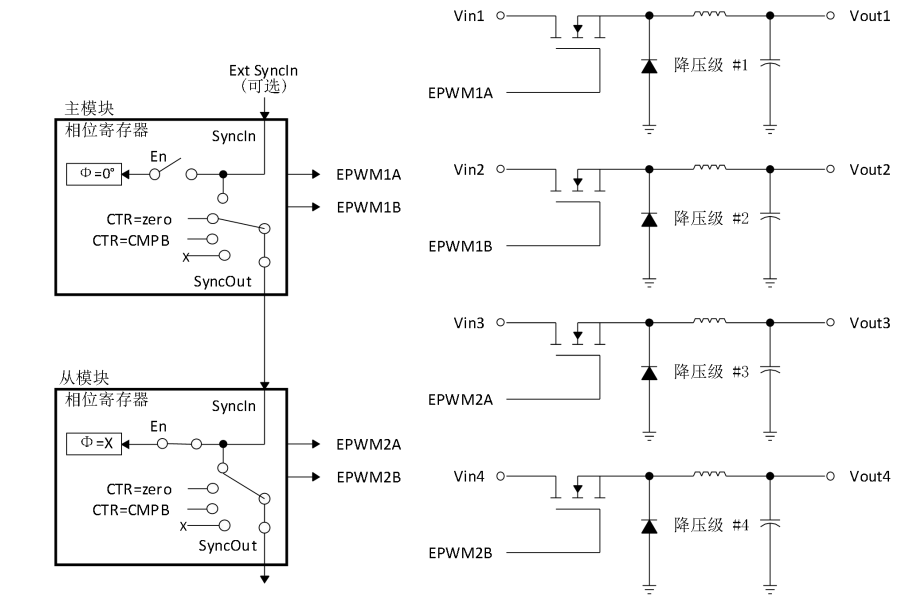



图 15-60 四个降压级控制的降压波形（注：此处仅展示三个降压级）

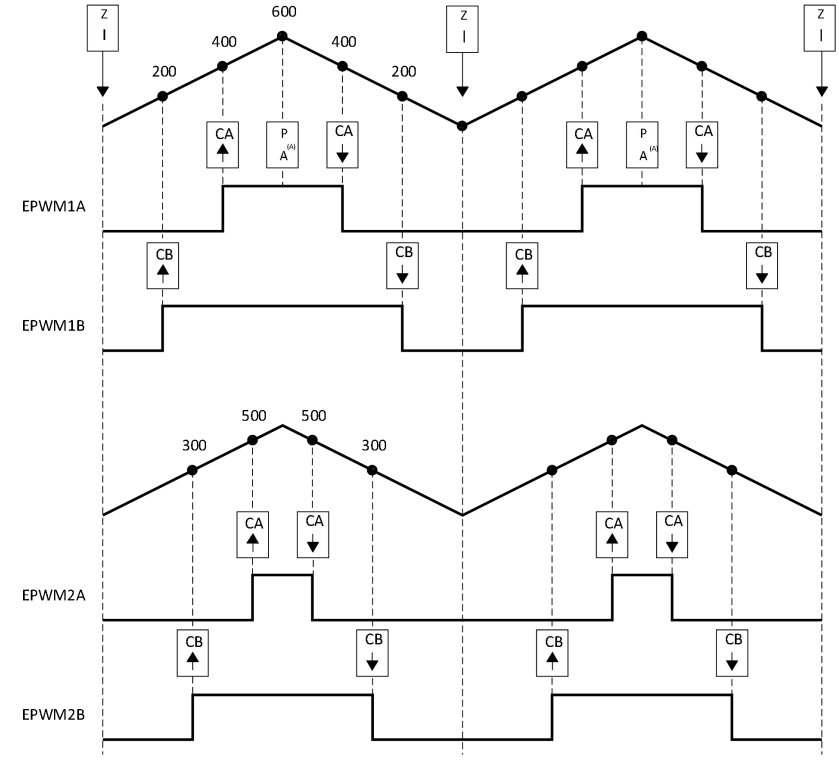
## 15.13.4 控制多个具有相同频率的降压转换器

若需要同步功能，可将 ePWM 模块 2 配置为从模块，其运行频率为模块 1 的整数倍（N）。主模块向从模块发送的同步信号可确保这些模块保持同步锁定。图中展示了此类配置；图中展示了该配置生成的波形。




A. ϕ = X 表示相位寄存器的值无关紧要

图 15-61 四个降压级的控制（注：此处 F_PWM2＝N×F_PWM1）




A. 启动 ADC 转换。

图 15-62 四个降压级控制的降压波形（注：此处 F_PWM2＝F_PWM1）

## 15.13.5 控制多个半桥（HHB）转换器

需要控制多个开关元件的拓扑结构也可通过这些 ePWM 模块实现。单个 ePWM 模块可控制一个半桥级，且该控制可扩展至多个级。

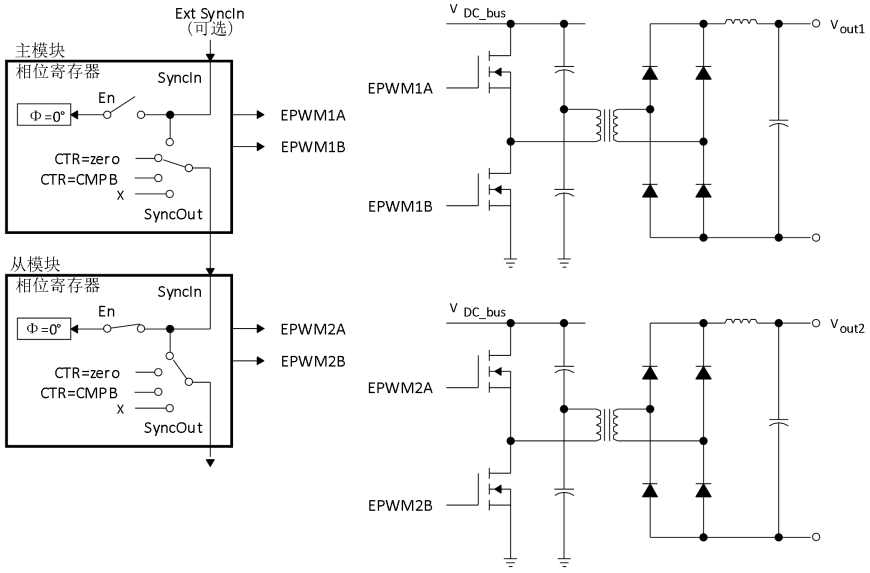



图 15-63 两个半桥级的控制（注：此处 F_PWM2＝N×F_PWM1）

ePWM 模块 2（从模块）配置为同步直通模式；若需要，该配置可允许 ePWM 模块 3 控制第三个半桥，且最重要的是，该半桥可与主 ePWM 模块 1 保持同步。

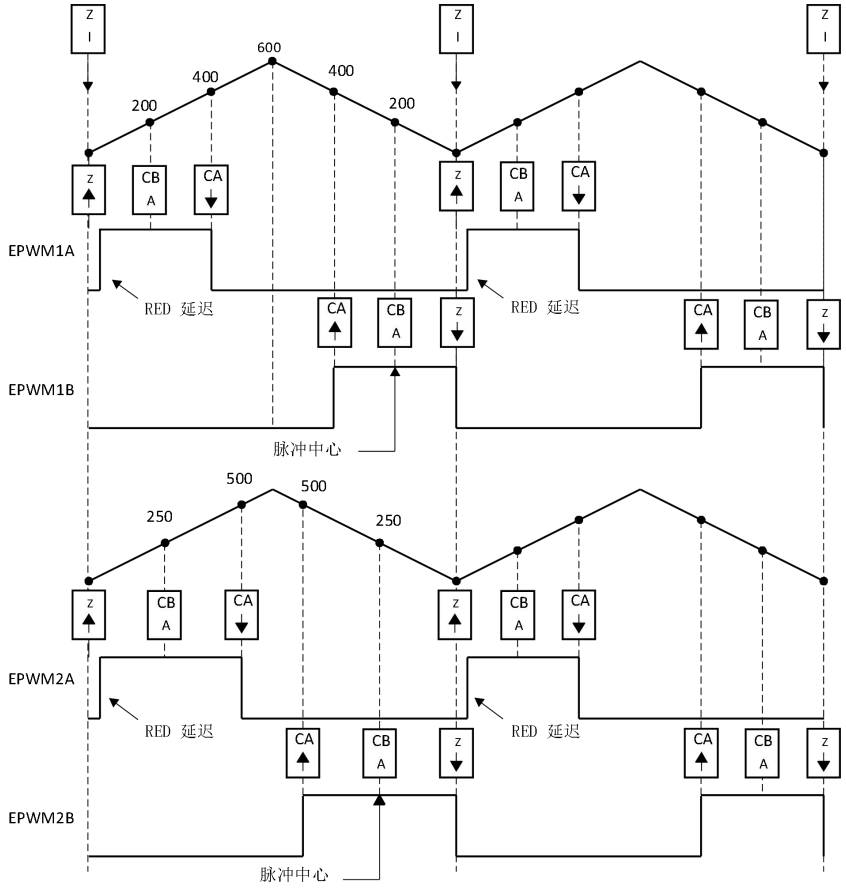



图 15-64 两个半桥级控制的半桥波形（注：此处 F_PWM2 = F_PWM1）

### 15.13.6 控制用于电机（交流感应电机和永磁同步电机）的双三相逆变器

多个模块控制单个功率级的理念可扩展至三相逆变器场景。在这种情况下，六个开关元件由三个 PWM 模块控制，每个模块对应逆变器的一个桥臂。每个桥臂必须以相同频率开关，且所有桥臂必须同步。主 - 从配置可轻松满足这一要求。图中展示了六个 PWM 模块如何控制两个独立的三相逆变器，每个逆变器驱动一台电机。

与前文所示场景相同，可选择让每个逆变器运行在不同频率（模块 1 和模块 4 为主模块，如图所示），或通过一个主模块（模块 1）和五个从模块实现两个逆变器的同步。在这种情况下，模块 4、5、6 的频率（均相等）可为模块 1、2、3 频率（均相等）的整数倍。

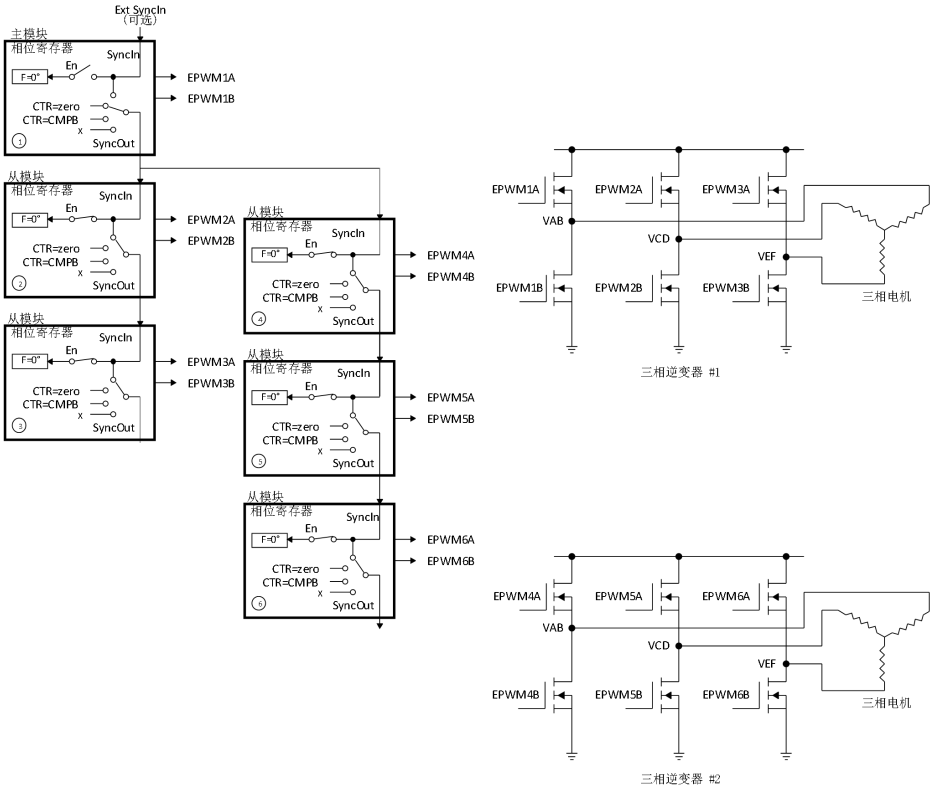



图 15-65 双三相逆变器级的控制（常用于电机控制）

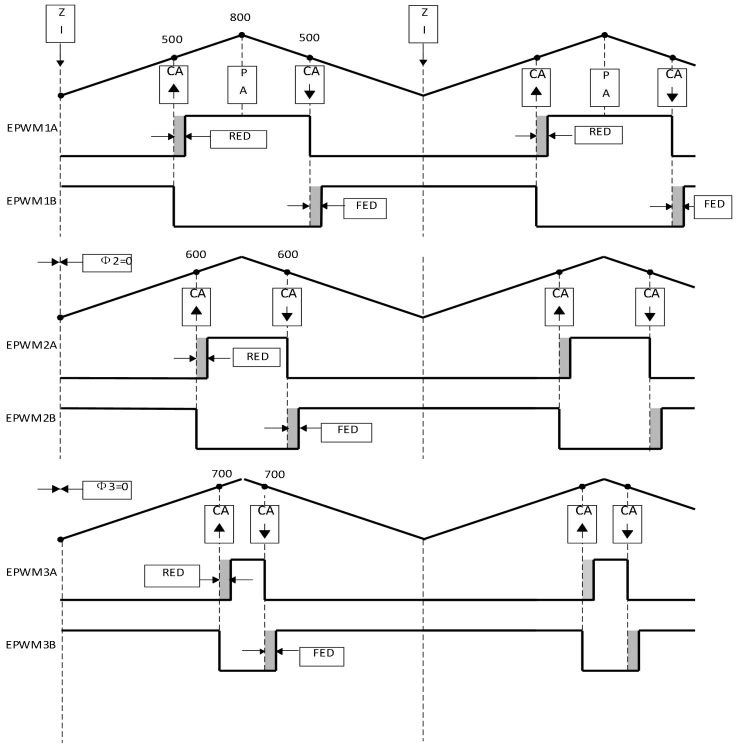



图 15-66 双三相逆变器级控制的三相逆变器波形（仅展示一个逆变器）

### 15.13.7 PWM 模块间相位控制的实际应用

到目前为止，所有示例均未使用相位寄存器（TBPHS），该寄存器要么设置为零，要么视为无关项。然而，通过向 TBPHS 写入适当的值，多个 PWM 模块可应对另一类依赖桥臂（或级）间相位关系才能正常工作的功率拓扑结构。如时基子模块部分所述，可配置 PWM 模块，使同步输入（SyncIn）脉冲触发将 TBPHS 寄存器的值加载到 TBCTR 寄存器中。为说明这一概念，图中展示了一个主模块和一个从模块，二者的相位关系为 120°（即从模块超前主模块）。

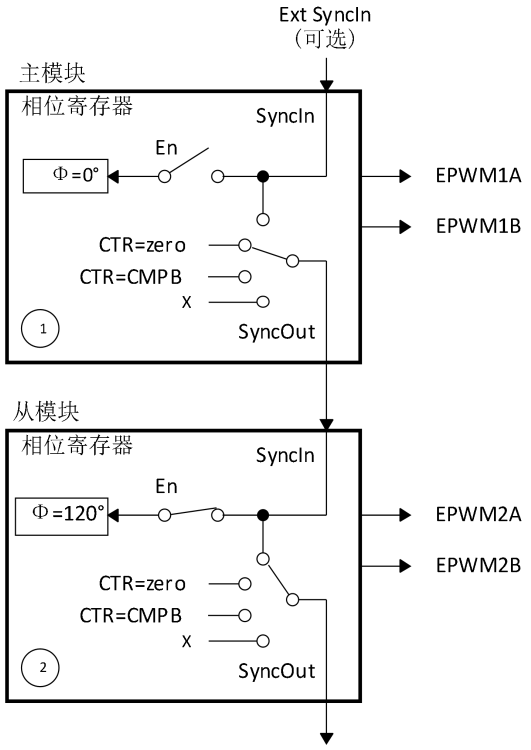



图 15-67 配置两个 PWM 模块以实现相位控制

图中展示了该配置的相关时序波形。在此示例中，主模块和从模块的 TBPRD 均为 600。对于从模块，TBPHS 为 200（即 200/600×360°=120°）。每当主模块生成同步输入（SyncIn）脉冲（CTR = PRD）时，TBPHS = 200 的值会加载到从模块的 TBCTR 寄存器中，因此从模块的时基始终超前主模块的时基 120°。

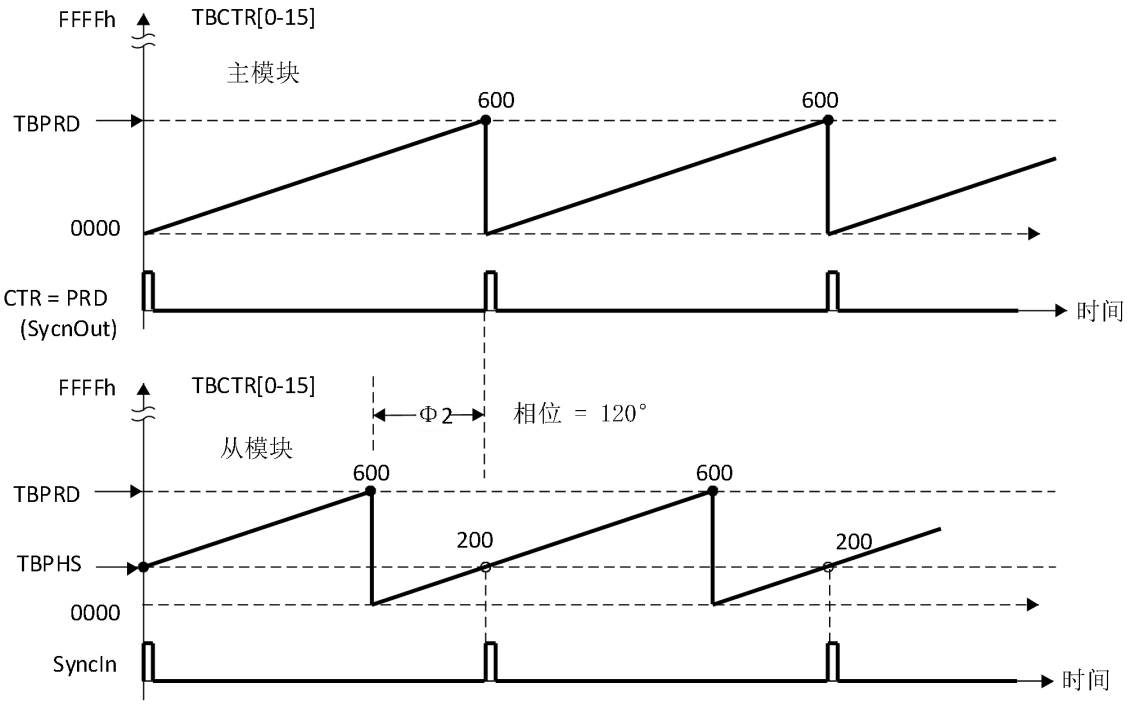



图 15-68 两个模块间相位控制的相关时序波形

对于 N 相系统，TBPHS 值可通过以下公式计算：

TBPHS (N, M) = (TBPRD / N) × (M - 1)

其中：

N = 相数

M = PWM 模块编号

示例：对于三相系统（N=3），TBPRD = 600：

TBPHS(3,2) = (600/3)×(2-1) = 200（即从模块 2 的相位值）

TBPHS(3,3) = 400（即从模块 3 的相位值）

### 15.13.8 控制三相交错式 DC/DC 转换器

利用模块间相位偏移的一种常用功率拓扑结构如图所示。该系统使用三个 PWM 模块，其中模块 1 配置为主模块。为确保系统正常工作，相邻模块间的相位关系必须为 120°。这可通过将从模块 2 和 3 的 TBPHS 寄存器分别设置为周期值的 1/3

和 2/3 来实现。例如，若周期寄存器加载的值为 600 个计数，则 TBPHS（从模块 2）= 200，TBPHS（从模块 3）= 400。两个从模块均与主模块 1 同步。

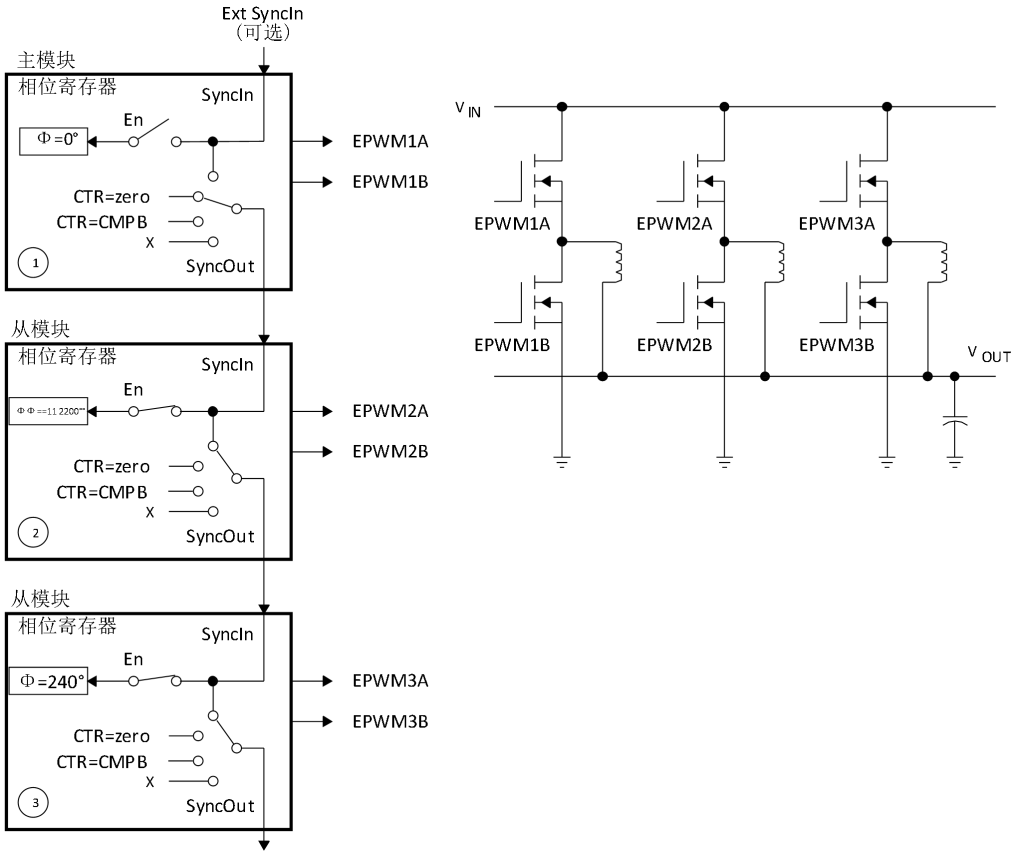



图 15-69 三相交错式 DC/DC 转换器的控制

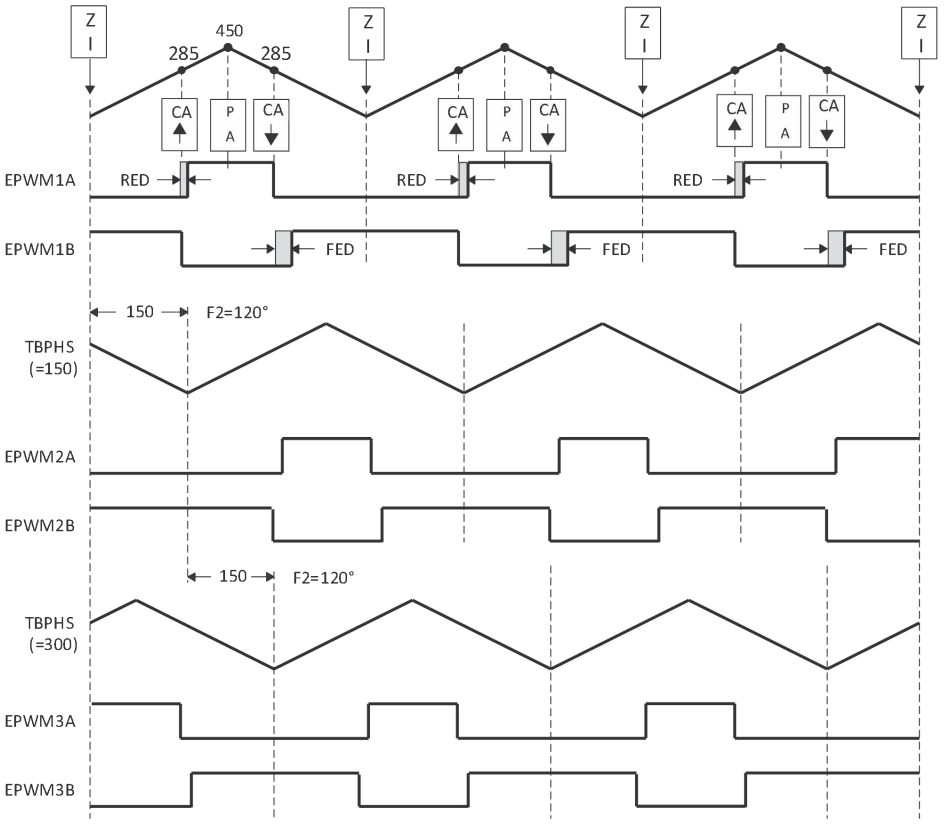



图 15-70 三相交错式 DC/DC 转换器控制的波形

## 15.13.9 控制零电压开关全桥（ZVSFB）转换器

下图所示示例假设桥臂（模块）间的相位关系为静态或恒定。在这种情况下，通过调制占空比实现控制。也可逐周期动态改变相位值。此功能适用于控制一类称为零电压开关（ZVS）全桥转换器的功率拓扑结构。在这类拓扑结构中，受控参数并非占空比（占空比保持在约 50% 恒定），而是桥臂间的相位关系。此类系统可通过分配两个 PWM 模块的资源来控制单个功率级（进而控制四个开关元件）实现。图中展示了一个主 - 从模块组合，二者同步工作以控制一个全桥。在此情况下，主模块和从模块均需以相同的 PWM 频率开关。相位通过从模块的相位寄存器（TBPHS）控制。主模块的相位寄存器未使用，因此可初始化为零。

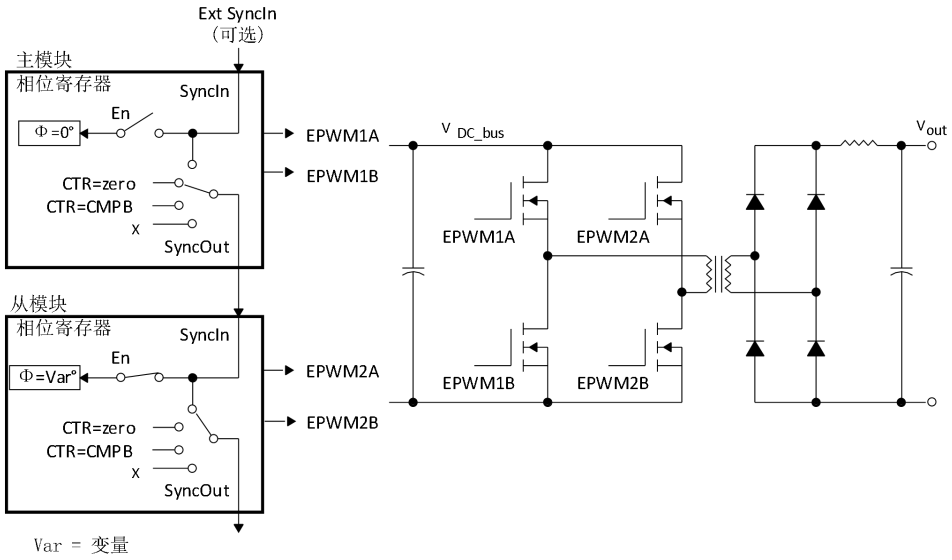



图 15-71 全桥级的控制（F_PWM2 = F_PWM1）

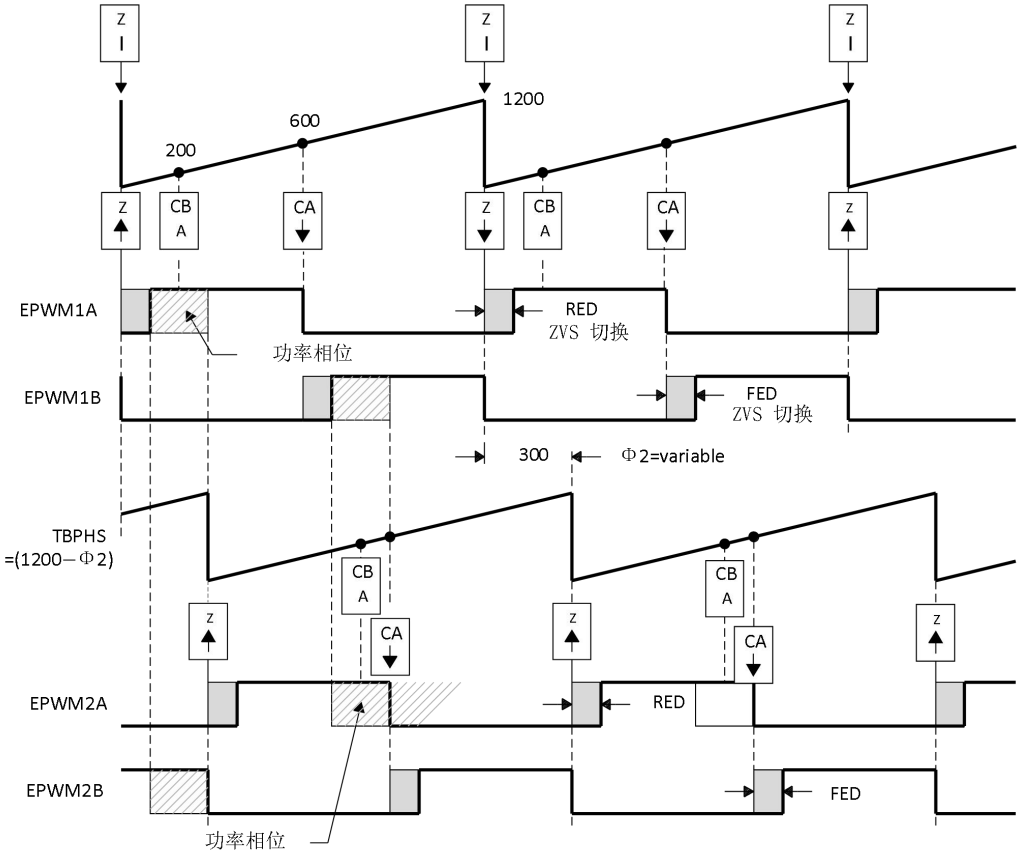



图 15-72 零电压开关全桥波形

## 15.13.10 控制峰值电流模式控制的降压模块

峰值电流控制技术具有诸多优势，例如自动过流限制、对输入电压变化的快速校正以及减少磁饱和等。图中展示了 ePWM1A 与片上模拟比较器在降压转换器拓扑结构中的应用。输出电流通过电流检测电阻检测，并输入到片上比较器的正输入端。内部可编程 12 位 DAC 可用于在比较器的负输入端提供参考峰值电流，也可在此输入端连接外部参考。比较器的输出是数字比较子模块的输入。ePWM 模块配置为一旦检测到的电流达到参考峰值，立即触发 ePWM1A 输出跳闸。系统采用逐周期跳闸机制。图中展示了该配置生成的波形。

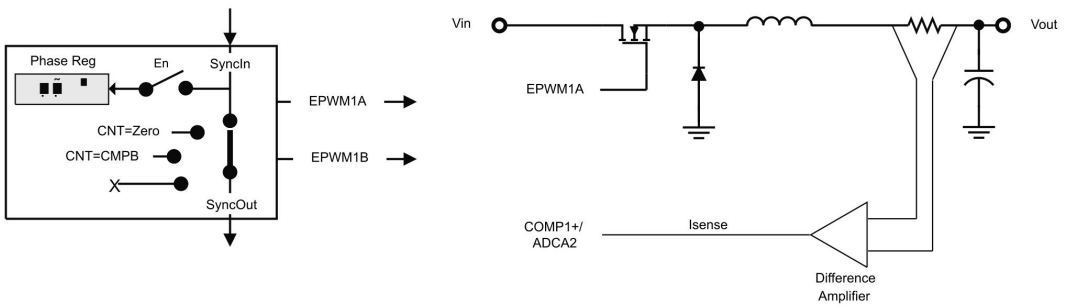



图 15-73 降压转换器的峰值电流模式控制

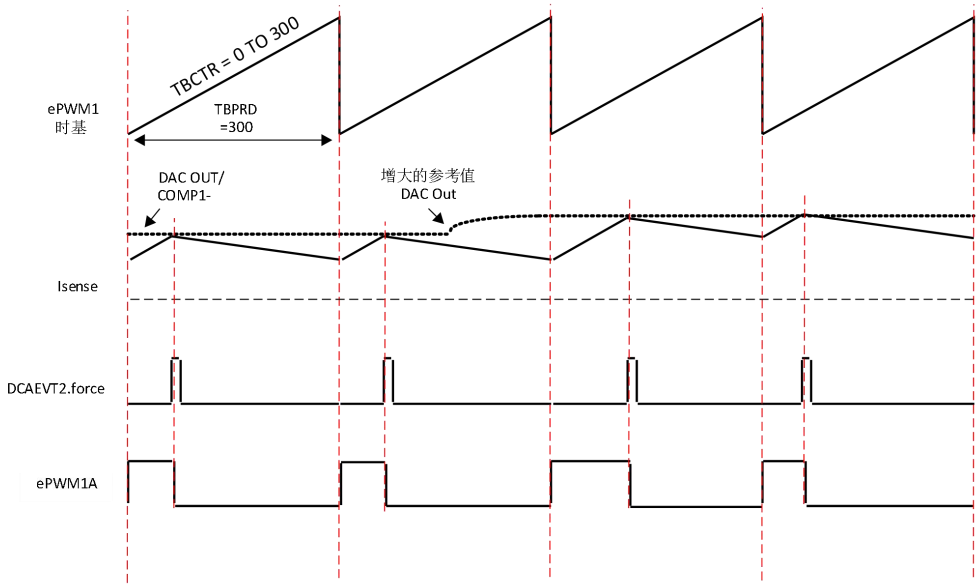
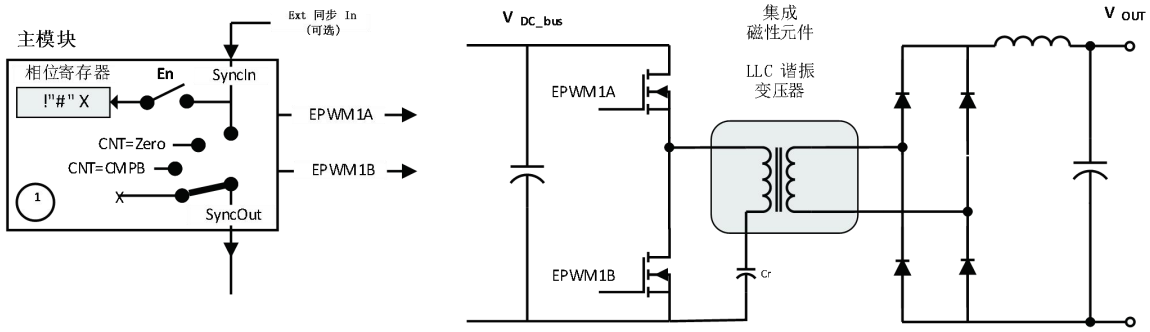



图 15-74 降压转换器控制的峰值电流模式控制波形

## 15.13.11 控制全桥 LLC 谐振转换器

谐振转换器的多种拓扑结构在电力电子领域已应用多年。此外，全桥 LLC 谐振转换器拓扑结构近年来在许多对效率和功率密度有要求的消费电子应用中备受青睐。本示例详细介绍了 ePWM1 的单通道配置，但该配置可轻松扩展至多通道。在此示例中，受控参数并非占空比（占空比保持在约 50% 恒定），而是频率。尽管死区未被控制且保持恒定为 300ns（即 100MHz TBCLK 下为 30 个计数），但用户可实时更新死区，通过调整足够的时延以实现软开关，从而提高效率。




注：Θ = X 表示相位寄存器的值无关紧要

图 15-75 两个谐振转换器级的控制

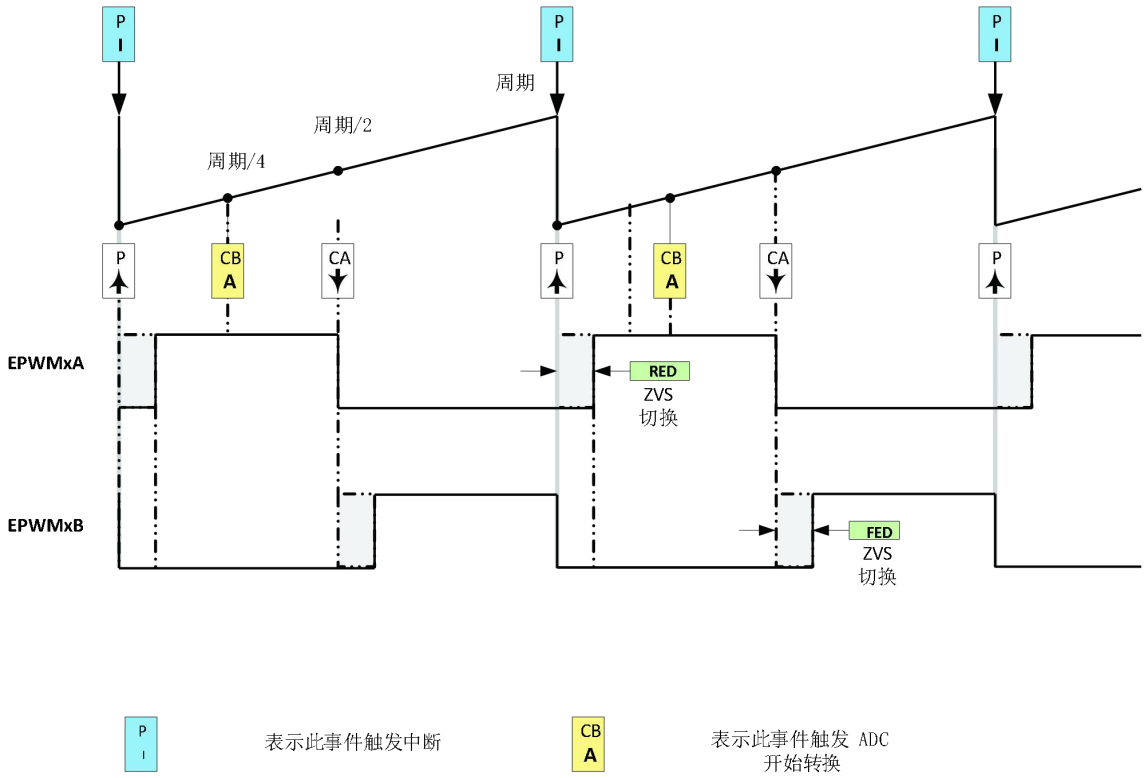



图 15-76 全桥 LLC 谐振转换器的 PWM 波形

## 15.14 寄存器锁定保护

寄存器锁定保护机制用于防止失控代码意外写入关键 ePWM 寄存器，导致寄存器损坏。EPWMLOCK 寄存器包含锁定位的定义（表中列出了锁定位及对应的寄存器）。该寄存器还设有一个 KEY 字段；仅当 KEY 字段写入值 0xa5a5 时，对该寄存器的写入操作才会成功。

表 15-15 锁定位及对应的寄存器

| 位域 | 定义 | 被锁定的寄存器 |
|---|---|---|
| HRLOCK | HRPWM 寄存器组锁定 | HRCNFG、HRPWR、HRMSTEP、HRPCTL |
| GLLOCK | 全局加载寄存器组锁定 | GLDCTL、GLDCFG |
| TZCFGLOCK | 跳闸区域寄存器组锁定 | TZSEL、TZDCSEL、TZCTL、TZCTL2、TZCTLDCA、TZCTLDCB、TZEINT |
| TZCLRLOCK | 跳闸区域清除寄存器组锁定 | TZCLR、TZCBCCLR、TZOSTCLR、TZFRC |
| DCLOCK | 数字比较寄存器组锁定 | DCTRIPSEL、DCACTL、DCBCTL、DCFCTL、DCCAPCTL、DCAHTRIPSEL、DCALTRIPSEL、DCBHTRIPSEL、DCBLTRIPSEL |

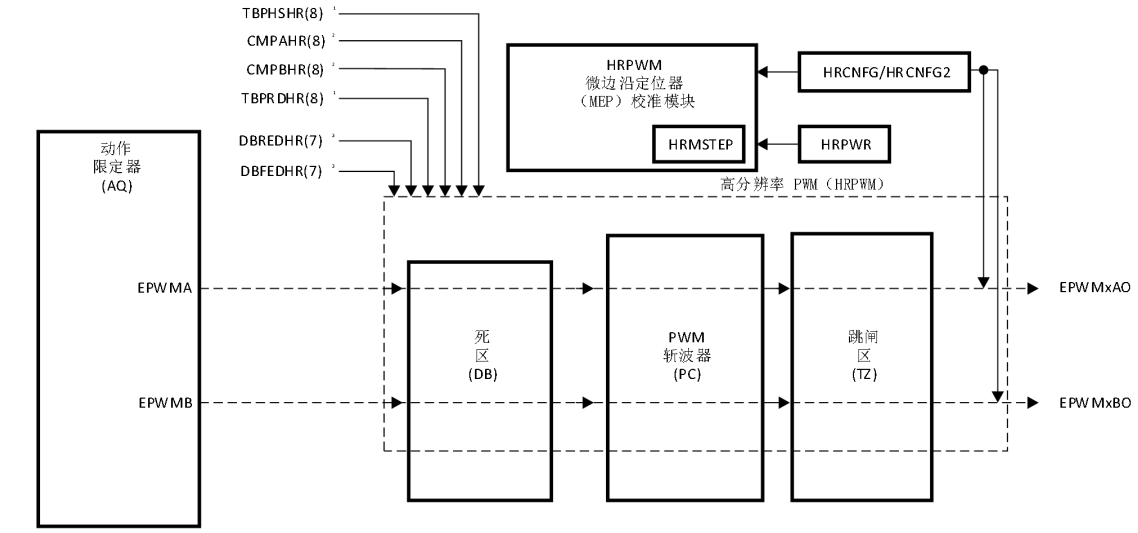
注：由于同一寄存器中存在 KEY 字段，仅当 KEY 匹配时，32 位写入操作才会成功。对该寄存器高半部分或低半部分的 16 位写入操作将被忽略。

## 15.15 高分辨率脉冲宽度调制器（HRPWM）

图中给出了 HRPWM 的框图。该模块提升了传统数字脉宽调制器（PWM）的时间分辨率性能。当 PWM 分辨率降至约 9~10 位以下时，通常会采用 HRPWM。HRPWM 的主要特性包括：

- 具备扩展的时间分辨率能力
- 可同时用于占空比控制和相移控制方式
- 通过对比较寄存器 A、比较寄存器 B 以及相位寄存器的扩展，实现更精细的时间粒度控制或边沿定位
- 利用 PWM 的 A、B 信号通路实现，即作用于 EPWMxA 与 EPWMxB 输出端
- 支持半周期时钟工作模式下下降沿与上升沿延迟的死区高精度控制
- 具备自检诊断软件模式，用于检查微边沿定位器（MEP）逻辑是否按设计正常运行
- 支持 EPWMxA 与 EPWMxB 输出端的高精度输出互换
- 可通过对 EPWMxA 信号输出取反，实现 EPWMxB 信号输出的高精度控制
- 在搭载 ePWM 模块的器件上，支持对 EPWMxA 与 EPWMxB 输出进行高精度周期、占空比和相位控制

注：请查阅器件数据手册，确认您所用器件是否配备支持高精度周期功能的 ePWM 模块




A. 来自 ePWM 时基（TB）子模块
B. 来自 ePWM 计数器-compare（CC）子模块
C. 来自 ePWM Deadband（DB）子模块

图 15-77 HRPWM 模块框图

增强型脉宽调制（ePWM）外设可实现与数模转换器（DAC）在数学上等效的功能。如图所示，常规方式生成的脉宽调制（PWM）信号，其有效分辨率取决于 PWM 频率（或周期）与系统时钟频率。

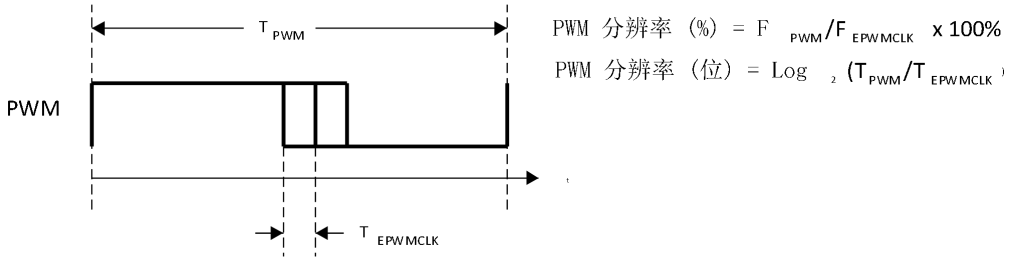



图 15-78 传统生成 PWM 的分辨率计算

如果所需的 PWM 工作频率在 PWM 模式下无法提供足够的分辨率，可考虑使用 HRPWM。作为 HRPWM 所提供性能提升的示例，表中列出了不同 PWM 频率下的分辨率（以位为单位）。这些值假设微边缘定位器（MEP）步长为 180ps。有关 MEP 的典型和最大性能规格，请参阅器件数据手册。

表 15-16 PWM 和 HRPWM 的分辨率

| PWM 频率（kHz） | 常规分辨率（PWM）<br>100MHz EPWMCLK | | 高分辨率（HRPWM） | |
|---|---|---|---|---|
| | 位数 | % | 位数 | % |
| 20 | 12.3 | 0.02 | 18.1 | 0.000 |
| 50 | 11 | 0.05 | 16.8 | 0.001 |

| PWM 频率（kHz） | 常规分辨率（PWM）<br>100MHz EPWMCLK | | 高分辨率（HRPWM） | |
|---|---|---|---|---|
| | 位数 | % | 位数 | % |
| 100 | 10 | 0.1 | 15.8 | 0.002 |
| 150 | 9.4 | 0.15 | 15.2 | 0.003 |
| 200 | 9 | 0.2 | 14.8 | 0.004 |
| 250 | 8.6 | 0.25 | 14.4 | 0.005 |
| 500 | 7.6 | 0.5 | 13.4 | 0.009 |
| 1000 | 6.6 | 1 | 12.4 | 0.018 |
| 1500 | 6.1 | 1.5 | 11.9 | 0.027 |
| 2000 | 5.6 | 2 | 11.4 | 0.036 |

尽管不同应用有所差异，但典型的低频 PWM 操作（低于 250kHz）不需要 HRPWM。HRPWM 功能最适用于功率转换拓扑的高频 PWM 需求，例如：

- 单相降压、升压和反激式转换器
- 多相降压、升压和反激式转换器
- 移相全桥转换器
- D 类功率放大器的直接调制

### 15.15.1 HRPWM 的工作原理描述

HRPWM 基于微边缘定位器（MEP）技术。MEP 逻辑能够通过细分传统 PWM 发生器的一个粗系统时钟，实现对边缘的超高精度定位，时间步长精度约为 150ps。有关特定器件的典型 MEP 步长，请参阅器件数据手册。HRPWM 还具备自检软件诊断模式，可检查 MEP 逻辑是否在所有工作条件下按设计运行。

图中展示了一个粗系统时钟与边缘位置（以 MEP 步长表示）之间的关系，这通过比较器 A 扩展寄存器（CMPAHR）中的 8 位字段进行控制。CMPBHR 的工作逻辑与此相同。

要生成 HRPWM 波形，需配置 ePWM 寄存器以生成特定频率和极性的传统 PWM。HRPWM 与 ePWM 寄存器协同工作，以扩展边缘分辨率。尽管存在多种编程组合，但实际应用中只需几种即可。

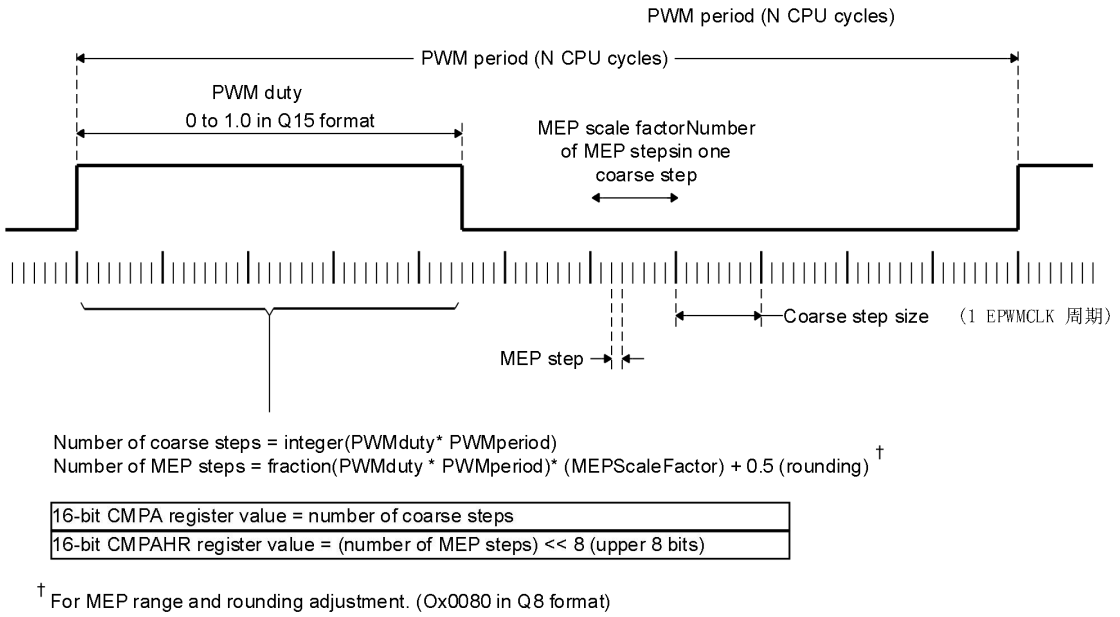



图 15-79 基于 MEP 的工作逻辑

### 15.15.1.1 控制 HRPWM 功能

HRPWM 的 MEP 由六个扩展寄存器控制。这些 HRPWM 寄存器与用于控制 PWM 操作的 16 位 TBPHS、TBPRD、CMPA、CMPB、DBRED 和 DBFED 寄存器串联使用：

TBPHSHR：时基相位高分辨率寄存器

CMPAHR：计数器比较 A 高分辨率寄存器（适用于通道 A 的 AQ 输出，与 CMPA 无关）

TBPRDHR：时基周期高分辨率寄存器（部分器件支持）

CMPBHR：计数器比较 B 高分辨率寄存器（适用于通道 B 的 AQ 输出，与 CMPB 无关）

DBREDHR：死区发生器上升沿延迟高分辨率寄存器

DBFEDHR：死区发生器下降沿延迟高分辨率寄存器

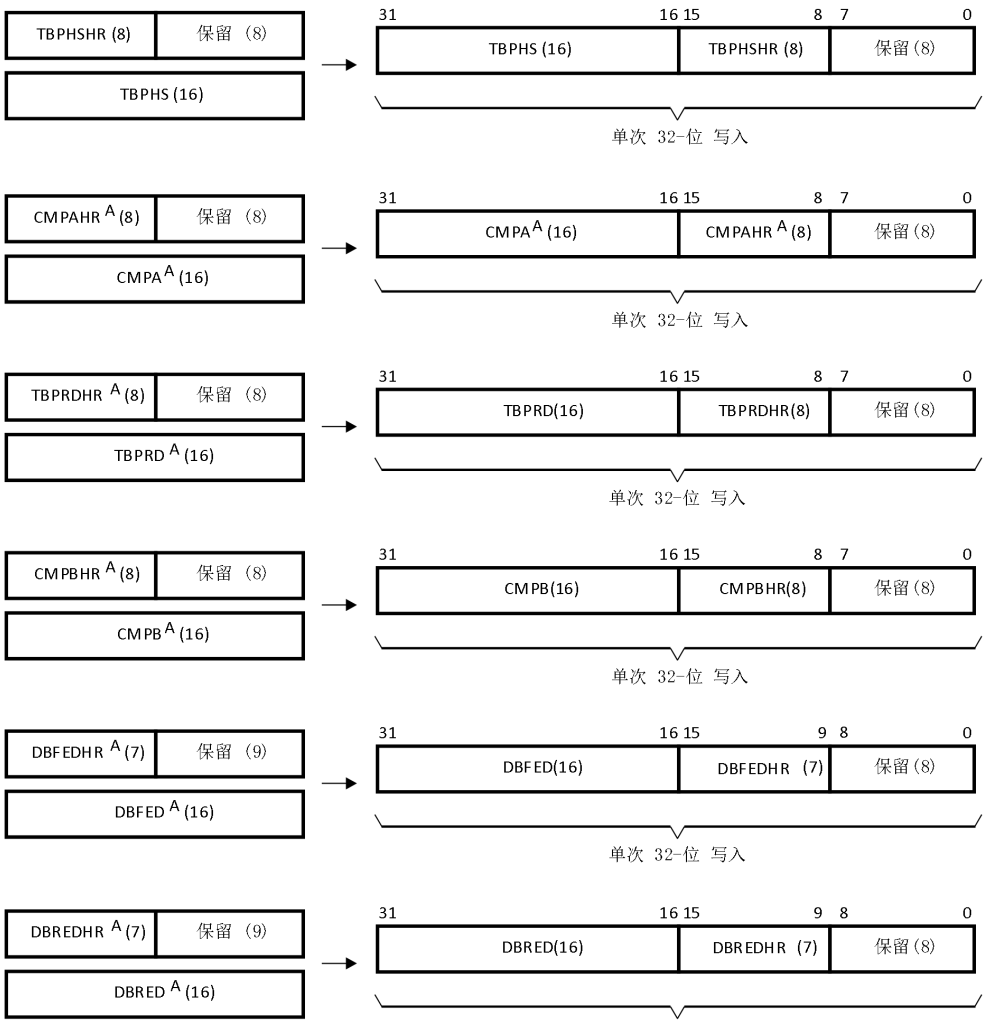



图 15-80 HRPWM 扩展寄存器和内存配置

A. 根据设备类型，这些寄存器可进行镜像操作，并可写入两个不同的内存地址。

注：死区上升沿延迟和下降沿延迟的 HRPWM 功能仅在死区半周期时钟操作期间适用。由于相同原因，其 MEP 步长数量（位 15:9）比占空比和相位高分辨率寄存器少一半。

HRPWM 功能通过通道 A 和 B 的 PWM 信号路径进行控制。通过正确配置 HRCNFG2 寄存器，死区信号路径也支持 HRPWM。图中展示了 HRPWM 如何与 8 位扩展寄存器接口。

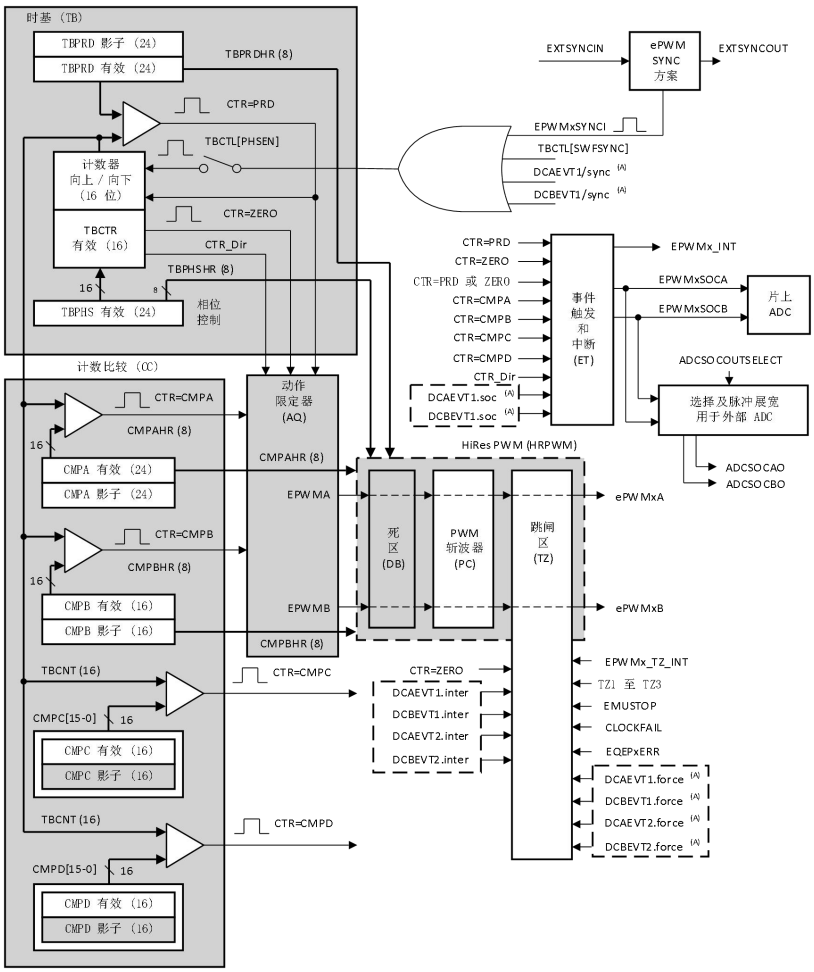



图 15-81 HRPWM 系统接口

A.这些事件由 ePWM 数字比较（DC）子模块根据 TRIPIN 输入电平生成。

### 15.15.1.2 HRPWM 源时钟

每个 HRPWM 模块均由相应的 EPWMxCLK 提供时钟。HRCAL 有独立的时钟。例如，HRPWM1 由 EPWM1CLK 提供时钟，而 HRPWM2 由 EPWM2CLK 提供时钟。图中展示了 HRCAL 和 HRPWM 模块均由各自的 ePWM 时钟源提供时钟。

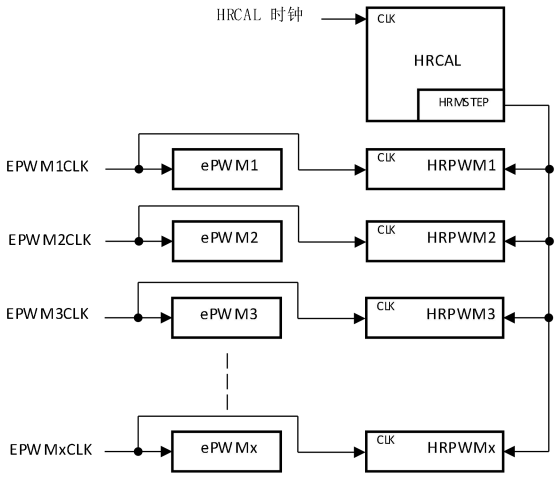



图 15-82 HRPWM 和 HRCAL 源时钟

### 15.15.1.3 配置 HRPWM

一旦 ePWM 配置为提供特定频率和极性的传统 PWM，就可以通过对该特定 ePWM 模块的寄存器空间中的 HRCNFG 寄存器进行编程来配置 HRPWM。该寄存器提供以下配置选项：

（1）边缘模式

MEP 可配置为在上升沿（RE）、下降沿（FE）或同时在两个边缘（BE）提供精确的位置控制。FE 和 RE 用于需要占空比控制的功率拓扑（CMPA 或 CMPB 高分辨率控制），而 BE 用于需要移相的拓扑，例如移相全桥（TBPHS 或 TBPRD 高分辨率控制）。

（2）控制模式

MEP 可配置为由 CMPAHR/CMPBHR 寄存器（占空比控制）或 TBPHSHR 寄存器（相位控制）控制。RE 或 FE 控制模式可与 CMPAHR 或 CMPBHR 寄存器配合使用。BE 控制模式可与 TBPHSHR 寄存器配合使用。当 MEP 由 TBPRDHR 寄存器控制（周期控制）时，占空比和相位也可通过其各自的高分辨率寄存器进行控制。

（3）影子模式

该模式提供与常规 PWM 模式相同的影子（双缓冲）选项。此选项仅在操作 CMPAHR、CMPBHR 和 TBPRDHR 寄存器时有效，且可选择与 CMPA/CMPB 寄存器的常规加载选项相同。如果使用 TBPHSHR，则此选项无效。

（4）高分辨率 B 信号控制

ePWM 通道的 B 信号路径可通过在 ePWMxB 引脚上输出高分辨率 ePWMxA 信号的反相信号来生成高分辨率输出。Type 2 或 Type 4 HRPWM 模块还可独立于 A 信号路径在 B 信号路径上启用高分辨率功能。

（5）交换 ePWMxA 和 ePWMxB 输出

此模式支持高分辨率 A 和 B 输出的交换。模式选择允许 A 和 B 输出保持不变，或 A 输出从 B 引脚输出、B 输出从 A 引脚输出。

（6）自动转换模式

此模式仅与比例因子优化（SFO）软件结合使用。对于 Type 4 HRPWM 模块，以下以 CMPAHR 为例描述自动转换模式。如果启用自动转换，则 CMPAHR = 小数部分（PWM 占空比 ×PWM 周期）<<8。比例因子优化软件在后台代码中计算 MEP 比例因子，并自动用计算出的每个粗步的 MEP 步数更新 HRMSTEP 寄存器。然后，MEP 校准模块使用 HRMSTEP 和 CMPAHR 寄存器中的值，自动计算由小数占空比表示的相应 MEP 步数，并相应地移动高分辨率 ePWM 信号边缘。如果禁用自动转换，则 CMPAHR 寄存器的行为与 Type 0 HRPWM 模块相同，且 CMPAHR =（小数部分（PWM 占空比 ×PWM 周期）×MEP 比例因子 + 0.5）<<8）。在此模式下，所有计算都需要由用户代码执行，且 HRMSTEP 寄存器将被忽略。高分辨率周期的自动转换与高分辨率占空比的自动转换行为相同。高分辨率周期模式必须始终启用自动转换。

注：如果 HRPWM 模块配置为上下计数模式，则 HRPWM 寄存器的影子模式必须设置为在零和周期时加载。用户的新值仅在 CTR=ZERO 时加载到影子寄存器，

但寄存器的影子模式必须设置为零和周期。CTR=PRD 事件用于 HRPWM 模块内部的特定逻辑。

自动转换模式对 CMPBHR、DBREDHR 和 DBFEDHR 执行计算。比例因子优化软件在后台代码中计算 MEP 比例因子，并自动用计算出的每个粗步的 MEP 步数更新 HRMSTEP 寄存器。然后，MEP 校准模块使用 HRMSTEP 和 CMPBHR 或 DBREDHR/DBFEDHR 寄存器中的值，自动计算由小数部分表示的相应 MEP 步数，并相应地移动高分辨率 ePWM 信号边缘。如果禁用自动转换，则 CMPBHR 的行为与 CMPAHR 相同，即 CMPBHR =（小数部分（PWM 占空比 ×PWM 周期）× MEP 比例因子 + 0.5）<<8。

当应用程序需要访问 ePWM1 和 HRPWM 中的任一模块时，应禁用这两个模块的保护。

#### 15.15.1.4 配置死区上升沿和下降沿的高分辨率

一旦 ePWM 配置为提供特定频率、极性的传统 PWM，并且在半周期时钟模式下启用了死区，就可以通过对该特定 ePWM 模块寄存器空间中的 HRCNFG2 寄存器进行编程来启用死区 RED 和 FED 线路的高分辨率操作。该寄存器提供以下配置选项：

（1）边缘模式

MEP 可配置为在死区上升沿（RED）、死区下降沿（FED）或同时在两个边缘（死区 RED 信号的上升沿和死区 FED 信号的下降沿）提供精确的位置控制。

（2）控制模式

选择在高分辨率模式下将影子值加载到 DBRED 和 DBFED 的活动寄存器的时间事件。选择与 ePWM DBCTL [LOADREDMODE] 和 DBCTL [LOADFEDMODE] 位中的选择匹配的脉冲。

### 15.15.1.5 工作原理

MEP 逻辑能够将边缘放置在 255 个（8 位）离散时间步长之一（有关典型 MEP 步长，请参阅器件数据手册）。MEP 与时基模块（TBM）和计数器比较模块（CCM）配合使用，确保时间步长的应用，并且在宽范围的 PWM 频率、系统时钟频率和其他工作条件下保持边缘放置精度。表中展示了 HRPWM 支持的典型工作频率范围。

表 15-17 MEP 步长、PWM 频率和分辨率之间的关系

| 系统时钟（MHz） | 每个EPWMCLK的MEP步长(1)(2)(3) | PWM 最小频率（Hz）(4) | PWM 最大频率（MHz） | 最大频率下的分辨率（位）(5) |
|---|---|---|---|---|
| 60.0 | 93 | 916 | 3.00 | 10.9 |
| 70.0 | 79 | 1068 | 3.50 | 10.6 |
| 80.0 | 69 | 1221 | 4.00 | 10.4 |
| 90.0 | 62 | 1373 | 4.50 | 10.3 |
| 100.0 | 56 | 1526 | 5.00 | 10.1 |

（1）TBCLK = EPWMCLK。

（2）表中数据基于 180ps 的 MEP 时间分辨率（示例值）。有关 MEP 限制，请参阅器件数据手册。

（3）本示例中应用的 MEP 步长 = TEPWMCLK / 180ps。

（4）PWM 最小频率基于最大周期值（TBPRD = 65535）。PWM 模式为非对称向上计数。

（5）分辨率（以位为单位）针对指定的最大 PWM 频率给出。

边缘定位

注：以下示例使用 [CMPA:CMPAHR] 寄存器组合呈现。如果打算使用 [CMPB:CMPBHR] 进行占空比控制，其工作原理和公式相同。

在典型的功率控制环路中，数字控制器会发出占空比指令，通常以单位或百分比形式表示。假设在特定工作点，要求的占空比为 0.405（即 40.5% 的导通时间），且所需的转换器 PWM 频率为 1.25MHz。在系统时钟为 100MHz 的传统 PWM 生成中，占空比选项接近 40.5%。如图所示，计数为 32 的比较值（占空比 = 40%）

是最接近 40.5% 的值，相当于 320ns 的边缘位置，而非期望的 324ns。相关数据如表所示。

通过利用 MEP，可以实现更接近期望的 324ns 的边缘位置。表中显示，除 CMPA 值外，MEP 的 22 个步长（CMPAHR 寄存器）可将边缘定位在 323.96ns，几乎实现零误差。本示例假设 MEP 的步长分辨率为 180ps。

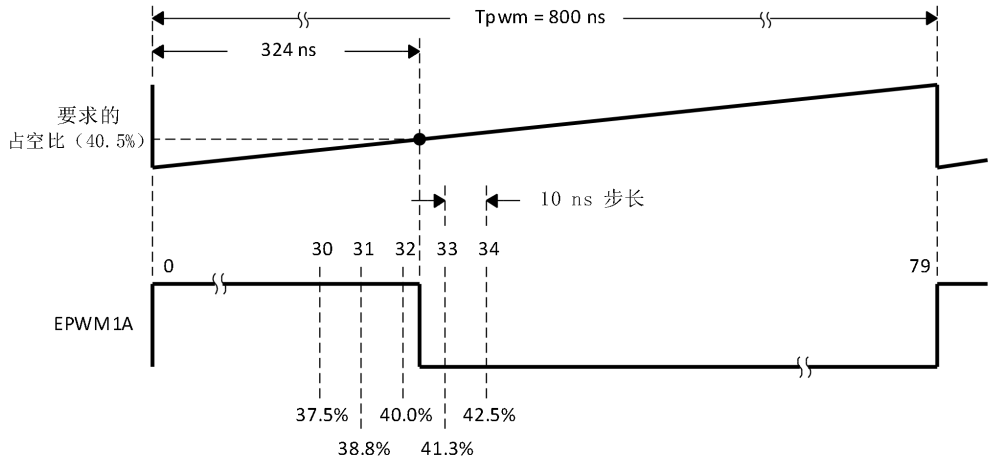



图 15-83 要求占空比 =40.5% 时所需的 PWM 波形

表 15-18 CMPA 与占空比（左）以及 [CMPA:CMPAHR] 与占空比（右）

| CMPA（计数）(1)(2)(3) | 占空比（%） | 高电平时间（ns） | CMPA（计数） | CMPAHR（计数） | 占空比（%） | 高电平时间（ns） |
|---|---|---|---|---|---|---|
| 28 | - | 280 | 32 | 18 | - | 323.24 |
| 29 | - | 290 | 32 | 19 | - | 323.42 |
| 30 | - | 300 | 32 | 20 | - | 323.60 |
| 31 | - | 310 | 32 | 21 | - | 323.78 |
| 32 | - | 320 | 32 | 22 | - | 323.96 |
| 33 | - | 330 | 32 | 23 | - | 324.14 |
| 34 | - | 340 | 32 | 24 | - | 324.32 |
| - | - | - | 32 | 25 | - | 324.50 |
| 要求值 | 40.5 | 324 | 32 | 26 | - | 324.68 |
| - | - | - | 32 | 27 | - | 324.86 |

（1）本示例假设的 MEP 步长 = 180ps。有关典型和最大 MEP 值，请参阅特定器件的数据手册。

（2）TBCLK=100MHz，即 10ns。

（3）PWM 周期寄存器值为 80 时，PWM 周期 = 80×10ns=800ns，PWM 频率 = 1/800ns=1.25MHz。

缩放考虑因素

前面已演示了如何利用标准 CMPA 和 MEP（CMPAHR）寄存器的资源在时间上精确放置边缘。然而，在实际应用中，需要无缝为 CPU 提供从标幺值（小数）占空比到写入 [CMPA:CMPAHR] 寄存器组合的最终整数（非小数）表示的映射函数。

要实现此映射方案，需要两步缩放过程。

本示例假设：

TBCLK=10ns(100MHz)

PWM 频率=1.25MHz(1/800ns)

所需 PWM 占空比，PWMDuty=0.405(40.5%)

以粗步表示的 PWM 周期，=80

PWMPeriod(800ns/10ns)

在 180ps(10ns/180ps)下每粗步的=55

MEP 步数，MEP_ScaleFactor

使 CMPAHR 保持在 1-255 范围内的 =0.5（Q8 格式下为 0080h）

值以及小数舍入常数（默认值）

步骤 1：CMPA 寄存器的百分比整数占空比值转换

CMPA 寄存器值 = int(PWMDuty×PWMPeriod);int 指整数部分

= int(0.405×80)

= int(32.4)

CMPA 寄存器值 = 32（20h）

步骤 2：CMPAHR 寄存器的小数值转换

CMPAHR =(frac(PWMDuty×PWMPeriod)×MEP_ScaleFactor + 0.5）<<8; frac 指小数部分

=(frac(32.4)×55 + 0.5)<<8;移位是为了将值移至 CMPAHR 的高字节

=（0.4×55 + 0.5）<<8

=（22 + 0.5）<<8

=22.5×256（左移 8 位等同于乘以 256）

=5760（1680h）

CMPAHR 值 = 1680h CMPAHR value = 1600h（硬件忽略低 8 位）

注：如果设置了 AUTOCONV 位（HRCNFG.6）且 MEP_ScaleFactor 在 HRMSTEP 寄存器中，则 CMPAHR/CMPBHR 寄存器值 = 小数部分（PWMDuty× PWM 周期）<<8。其余转换计算由硬件自动执行，正确的 MEP 缩放信号边缘将出现在 ePWM 通道输出上。如果未设置 AUTOCONV，则上述计算必须由软件执行。

MEP 比例因子（MEP_ScaleFactor）随系统时钟和 DSP 工作条件而变化。

比例因子在有限范围内变化缓慢，因此优化 C 函数可以在后台循环中以极低的频率运行。

CMPA、CMPB、CMPAHR 和 CMPBHR 寄存器在内存中配置为 CPU 的 32 位数据功能可将其作为单个串联值写入，即 [CMPA:CMPAHR]、[CMPB:CMPBHR] 等。

映射方案已在 C 语言和汇编语言中实现，如 15.15.1.8 节所示。实际实现利用了 32 位 CPU 架构。

对于每个周期都很关键的时间敏感型控制环路，建议使用汇编版本。这是一个周期优化函数（11 个 EPWMCLK 周期），接受 Q15 占空比值作为输入，并写入单个 [CMPA:CMPAHR] 值。

占空比范围限制

在高分辨率模式下，MEP 并非在 PWM 周期的 100% 时间内都处于活动状态，其工作情况如下：

未启用高分辨率周期（TBPRDHR）控制时，在周期开始后的三个 EPWMCLK 周期内激活。

使用 HRPCTL 寄存器启用高分辨率周期（TBPRDHR）控制时：

向上计数模式：从周期开始后的三个 EPWMCLK 周期到周期结束前的三个 EPWMCLK 周期。

上下计数模式：向上计数时，从 CTR=0 后的三个周期到 CTR=PRD 前的三个周期；向下计数时，从 CTR=PRD 后的三个周期到 CTR=0 前的三个周期。

使用 DBREDHR 或 DBFEDHR 时，DBRED 或 DBFED（与具有高分辨率位移的边缘对应的寄存器）必须大于或等于 7。

占空比范围限制如图 所示。此限制对 MEP 施加了占空比限制。例如，无法在接近 0% 占空比的整个范围内实现精确的边缘控制。当高分辨率周期控制禁用时，尽管 HRPWM 功能在前三个周期不可用，但常规 PWM 占空比控制仍可在低至 0% 占空比的范围内完全运行。在大多数应用中，这不会成为问题，因为控制器的调节点通常不会设计在接近 0% 占空比的位置。当启用高分辨率周期控制（HRPCTL [HRPE]=1）时，占空比不得落在限制范围内，否则 ePWMxA 输出可能会出现未定义行为。

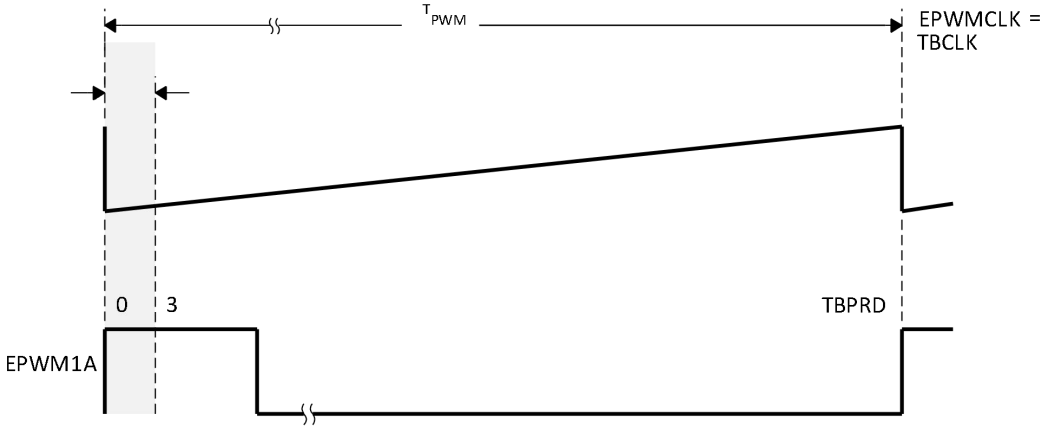



图 15-84 低百分比占空比范围限制示例（HRPCTL [HRPE]=0）

表 15-19 三个 EPWMCLK/TBCLK 周期的占空比范围限制

| PWM 频率(1)（kHz） | 3 个周期的最小占空比 | 3 个周期的最大占空比(2) |
|---|---|---|
| 200 | 0.6% | 99.4% |
| 400 | 1.2% | 98.8% |
| 600 | 1.8% | 98.2% |
| 800 | 2.4% | 97.6% |
| 1000 | 3% | 97% |
| 1200 | 3.6% | 96.4% |
| 1400 | 4.2% | 95.8% |

| PWM 频率(1)（kHz） | 3 个周期的最小占空比 | 3 个周期的最大占空比(2) |
|---|---|---|
| 1600 | 4.8% | 95.2% |
| 1800 | 5.4% | 94.6% |
| 2000 | 6% | 94% |

（1）EPWMCLK=TBCLK=100MHz

（2）此限制仅在启用高分辨率周期（TBPRDHR）控制时适用。

如果应用要求 HRPWM 在最小占空比限制以下运行，则在未启用高分辨率周期（HRPCTL [HRPE]=0）时，可将 HRPWM 配置为向下计数模式，由 MEP 控制上升沿位置（REP）。如图所示，在此配置下，最小占空比限制不再是问题。然而，存在最大占空比限制，其百分比数值与表中给出的相同。

警告：如果应用启用了高分辨率周期控制（HRPWM [HRPE]=1），占空比不得落在限制范围内，否则 ePWM 输出可能会出现未定义行为。

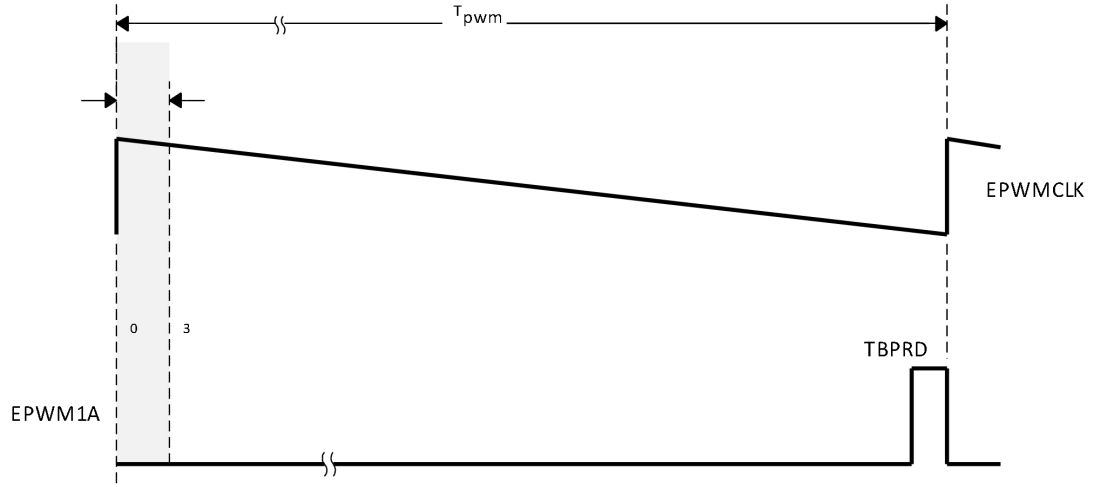



图 15-85 高百分比占空比范围限制示例（HRPCTL [HRPE]=0）

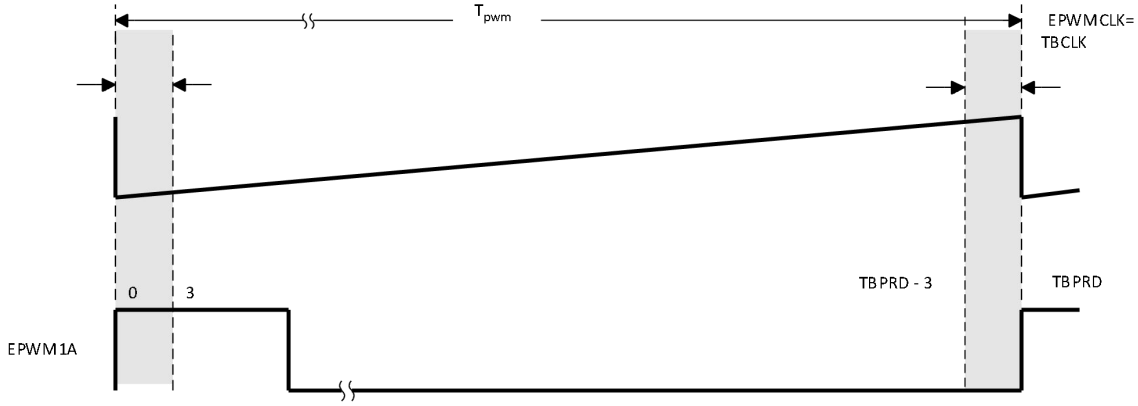



图 15-86 向上计数占空比范围限制示例（HRPCTL [HRPE]=1）

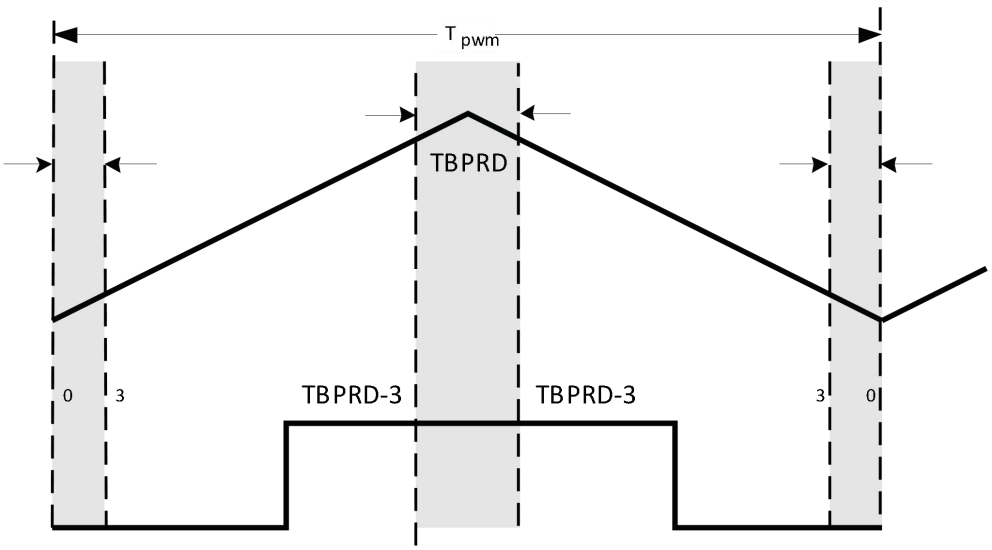



图 15-87 上下计数占空比范围限制示例（HRPCTL [HRPE]=1）

高分辨率周期

Type 1 及以上 ePWM 模块支持使用 MEP 逻辑的高分辨率周期控制。

注：当仅在 ePWMxA 上启用高分辨率周期控制而不在 ePWMxB 输出上启用时（反之亦然），非高分辨率输出在向上计数模式下会有 ±1 个 TBCLK 周期的抖动，在上下计数模式下会有 ±2 个 TBCLK 周期的抖动。

### 15.15.1.6 2 节中描述的占空比缩放过程也适用于高分辨率周期：

本示例假设：

TBCLK=10ns(100MHz)

所需 PWM 频率 = 175kHz(周期为 571.428ns)

在 180ps(10ns/180ps)下每粗步的=55

MEP 步数，MEP_ScaleFactor

使 CMPAHR 保持在 1-255 范围内的 =0.5（Q8 格式下为 0080h）

值以及小数舍入常数（默认值）

问题：

向上计数模式：

如果 TBPRD=571，则 PWM 频率 = 174.82kHz（周期 =(571+1)×TBCLK）

如果 TBPRD=570，则 PWM 频率 = 175.13kHz（周期 =(570+1)×TBCLK）

上下计数模式：

如果 TBPRD=286，则 PWM 频率 = 174.82kHz（周期 =(286×2)×TBCLK）

如果 TBPRD=285，则 PWM 频率 = 175.44kHz（周期 =(285×2)×TBCLK）

解决方案：

在 180ps 下每个粗步有 55 个 MEP 步长：

步骤 1：TBPRD 寄存器的百分比整数周期值转换

整数周期值 = 571×TTBCLK

= int(571.428)×TTBCLK

= int(PWMperiod) ×TTBCLK

向上计数模式：

TBPRD=570（TBPRD = 周期值 - 1）=023Ah

上下计数模式：

TBPRD=285（TBPRD = 周期值 / 2）=011Dh

步骤 2：TBPRDHR 寄存器的小数值转换

向上计数模式：

TBPRDHR 寄存器值 =(frac(PWMperiod)×MEP_ScaleFactor+0.5)

如果启用自动转换且

HRMSTEP=MEP_ScaleFactor 值（55）：

= frac(PWMperiod)<<8（移位是为了将值移至 TBPRDHR 的高字节）

TBPRDHR 寄存器值 = frac(571.428)<<8

=0.428×256

=6D00h

然后，自动转换自动执行计算，= ((TBPRDHR (15:0) >>8)×HRMSTEP

使 TBPRDHR MEP 延迟通过+80h) <<8

硬件缩放为：= (006Dh×55 +80h) >>8

=(17EBh)>>8

周期 MEP 延迟 = 0017h MEP 步长

上下计数模式：

TBPRDHR 寄存器值=(frac(PWMperiod)×MEP_ScaleFactor+0.5)

如果启用自动转换且

HRMSTEP=MEP_ScaleFactor 值（55）：

= frac(PWMperiod/ 2)<<8（移位是为了将值移至 TBPRDHR 的高字节）

TBPRDHR 寄存器值 = frac(285.714)<<8

=0.714×256

=B600h

然后，自动转换自动执行计算,=((TBPRDHR(15:0)>>8)×HRMSTEP+80h)<<8

使 TBPRDHR MEP 延迟通过= (00B6h×55 +80h) >>8

硬件缩放为：=(279Ah)>>8

周期 MEP 延迟 = 0027h MEP 步长

高分辨率周期配置

要使用高分辨率周期，必须按照以下确切顺序初始化 ePWMx 模块：

以下步骤使用带影子寄存器的 CMPA 以及相应的 HRCNFG 位在 EPWMxA 上进行高分辨率操作。要在 EPWMxB 上进行高分辨率操作，请对 B 通道字段进行相应替换。

启用 ePWMx 时钟。

启用 HRPWM 时钟。

禁用 TBCLKSYNC。

配置 ePWMx 寄存器（AQ、TBPRD、CC 等）：

ePWMx 仅可配置为向上计数或上下计数模式。高分辨率周期与向下计数模式不兼容。

TBPRD 和 CC 寄存器必须配置为影子加载。

CMPCTL [LOADAMODE]：向上计数模式下，CMPCTL [LOADAMODE]=1（在 CTR=PRD 时加载）；上下计数模式下，CMPCTL [LOADAMODE]=2（在 CTR=0 或 CTR=PRD 时加载）。

配置 HRCNFG 寄存器，使得：

HRCNFG [HRLOAD]=2（在 CTR=0 或 CTR=PRD 时加载）

HRCNFG [AUTOCONV]=1（启用自动转换）

HRCNFG [EDGMODE]=3（MEP 控制两个边缘）

要使 TBPHS:TBPHSHR 与高分辨率周期同步，请同时设置 HRPCTL [TBPSHRLOADE]=1 和 TBCTL [PHSEN]=1。在上下计数模式下，无论 TBPHSHR 的内容如何，都必须将这些位设置为 1。

启用高分辨率周期控制（HRPCTL [HRPE]=1）。

启用 TBCLKSYNC。

TBCTL[SWFSYNC]=1。

HRMSTEP 必须包含准确的 MEP 比例因子（每个 EPWMCLK 粗步的 MEP 步数），因为已启用自动转换。MEP 比例因子可使用 15.15.2 节中描述的 SFO () 函数获取。

要控制高分辨率周期，请写入 TBPRDHR 寄存器。

注：启用高分辨率周期模式时，EPWMxSYNC 脉冲会给 PWM 带来 ±1-2 个周期的抖动（向上计数模式下为 ±1 个周期，上下计数模式下为 ±2 个周期）。因此，不能将 EPWMxSYNCO 源设置为 CTR=0 或 CTR=CMPB。否则，每个 PWM 周期都会因同步脉冲而出现抖动。

当 EPWMxSYNCI 是 EPWMxSYNCO 源时，仅在高分辨率周期初始化期间可以发出一次软件同步脉冲。如果在 PWM 运行时施加软件同步脉冲，则同步脉冲时 PWM 输出会出现抖动。

### 15.15.1.7 死区高分辨率操作

注：在向上计数模式下，启用任何高分辨率模式时，死区模块不可用。

本示例假设：

系统时钟 = 10ns（100MHz）

在半周期模式下启用死区，TBCLK=EPWMCLK

所需 PWM 频率 = 1.33MHz（1/750ns）

所需 PWM 占空比 = 0.5（50%）

所需死区上升沿延迟 = 占空比的 5%

所需死区上升沿延迟（ns）=（0.05×375ns）=18.75ns

注：与使用 HRPWM 时的占空比限制类似，要使用高分辨率死区，DBRED 和 DBFED 值必须大于 3。

死区延迟值是 DBFED 和 DBRED 的函数：

启用半周期时钟时，计算下降沿延迟和上升沿延迟的公式变为：

FED = DBFED×TBCLK/2

RED = DBRED×TBCLK/2

计算 DBRED 和 DBFED 值：

所需死区上升沿延迟（ns）=18.75ns

DBRED = RED/(TBCLK/2)

=18.75ns/5ns

所需 DBRED=3.75

在 180ps 下每个粗步有 55 个 MEP 步长：

步骤 1：DBRED 寄存器的整数死区值转换

整数 DBRED 值 = int(RED/(TBCLK/2))

= int(3.75）

DBRED=3

步骤 2：死区高分辨率寄存器 DBREDHR 的小数值转换

DBREDHR 寄存器值 =(fra(所需 DBRED)×MEP_ScaleFactor + 0.5）<<8（移位是为了将值移至 DBREDHR 的高字节）

=（fra(3.75）×55 + 0.5）<<8

=（0.75×55 + 0.5）<<8

=41.75×256（左移 8 位等同于乘以 256）

DBREDHR 值 = 29C0h MEP 步长（硬件忽略上述计算的 DBREDHR 值中的低 9 位）

注：如果设置了 AUTOCONV 位（HRCNFG.6）且 MEP_ScaleFactor 在 HRMSTEP 寄存器中，则 DBREDHR:DBRED = fra(所需 DB 值）<<8。其余转换计算由硬件自动执行，正确的 MEP 缩放信号边缘将出现在 ePWM 通道输出上。如果未设置 AUTOCONV，则上述计算必须由软件执行。

### 15.15.1.8 比例因子优化软件（SFO）

微边缘定位器（MEP）逻辑能够将边缘放置在 255 个离散时间步长之一。如前所述，这些步长的大小约为 150ps（有关特定器件的典型 MEP 步长，请参阅器件数据手册）。MEP 步长大小会因最坏情况的工艺参数、工作温度和电压而变化。MEP 步长大小随电压降低和温度升高而增大，随电压升高和温度降低而减小。

要有效利用 MEP 功能，软件需要知道 MEP 缩放因子的正确值。为此，HRPWM 模块具有内置的自检和诊断功能，可用于确定任何工作条件下的最佳 MEP 比例因子值。

### 15.15.1.9 HRPWM 示例：使用优化的汇编代码

理解如何使用 HRPWM 功能的最佳方式是通过两个实际示例：

使用非对称 PWM（向上计数）且极性为高电平有效（Active High）的简单降压转换器。

使用简单 R+C 重建滤波器的 DAC 功能。

以下示例的初始化和配置代码均以 C 语言编写。为便于理解，使用了以下宏定义。

示例 15-2 假设 MEP 步长为 150ps，且不使用 SFO 库。

5.示例 17-2：HRPWM 头文件的宏定义

```
// HRPWM（高分辨率 PWM）
// ================================
// HRCNFG
#define HR_Disable 0x0
#define HR_REP 0x1 // 上升沿位置
#define HR_FEP 0x2 // 下降沿位置
#define HR_BEP 0x3 // 两个边缘位置
#define HR_CMP 0x0 // 由 CMPAHR 控制
#define HR_PHS 0x1 // 由 TBPHSHR 控制
#define HR_CTR_ZERO 0x0 // CTR=零事件
#define HR_CTR_PRD 0x1 // CTR=周期事件
#define HR_CTR_ZERO_PRD 0x2 // CTR=零或周期事件
#define HR_NORM_B 0x0 // 正常 ePWMxB 输出
#define HR_INVERT_B 0x1 // ePWMxB 是 ePWMxA 输出的反相
```

实现简单降压转换器

本示例中，PWM 要求如下：

PWM 频率 = 1MHz（即 TBPRD=100）

PWM 模式 = 非对称向上计数

分辨率 = 12.7 位（MEP 步长为 150ps）

图 展示了所需的 PWM 波形。如前所述，ePWM1 模块的配置与正常情况几乎相同，只是需要启用 / 选择相应的 MEP 选项。

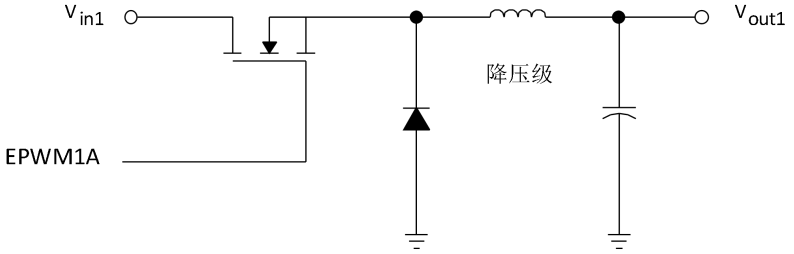



图 15-88 使用单个 PWM 的简单降压受控转换器

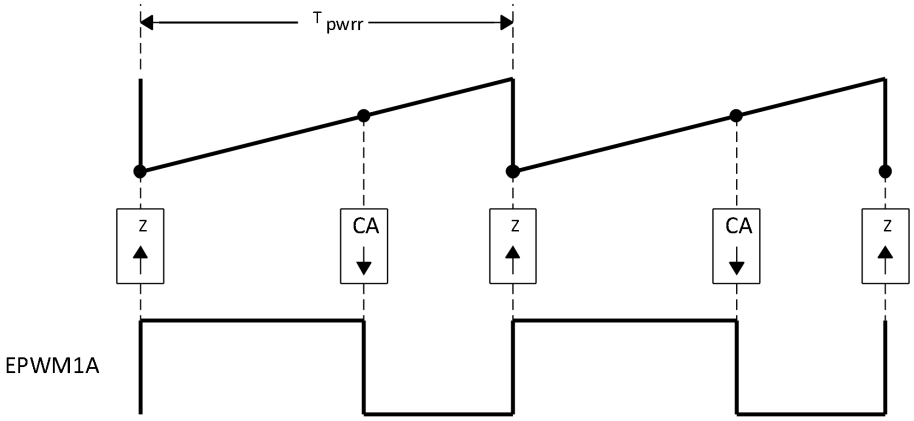



图 15-89 为简单降压受控转换器生成的 PWM 波形

所示示例代码包括两个主要部分：

- 初始化代码（执行一次）
- 运行时代码（通常在 ISR 内执行）

示例 15-3 展示了初始化代码。第一部分配置为传统 PWM，第二部分设置 HRPWM 资源。

本示例假设 MEP 步长为 150ps，且不使用 SFO 库。

示例 15-4 展示了 HRPWM 降压转换器的运行时代码的汇编示例。

示例 17-3：HRPWM 降压转换器初始化代码

```
void HrBuckDrvCnf(void)
{
// 首先配置传统 PWM
EPwm1Regs.TBCTL.bit.PRDLD = TB_IMMEDIATE; // 设置立即加载
EPwm1Regs.TBPRD = 100; // 周期设置为 1000kHz PWM
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // EPWM1 是主设备
hrbuck_period = 200; // 用于 Q15 到 Q0 的缩放
EPwm1Regs.EPWMSYNCOUTEN.all = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
// 注：此处初始化 ChB 仅用于比较目的，非必需
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // 可选
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW; // 可选
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_SET; // 可选
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR; // 可选
// 现在配置 HRPWM 资源
EALLOW; // 注：这些寄存器受保护，仅作用于 ChA
EPwm1Regs.HRCNFG.all = 0x0; // 首先清除所有位
EPwm1Regs.HRCNFG.bit.EDGMODE = HR_FEP; // 控制下降沿位置
EPwm1Regs.HRCNFG.bit.CTLMODE = HR_CMP; // 由 CMPAHR 控制 MEP
```

```
EPwm1Regs.HRCNFG.bit.HRLOAD = HR_CTR_ZERO; // 在 CTR=零时影子加载
EDIS;
MEP_ScaleFactor = 66*256; // 从 100MHz 的典型比例因子开始
// 注：使用 SFO 函数动态更新 MEP_ScaleFactor
}
```

示例 17-4：HRPWM 降压转换器运行时代码

```
asm
EPWM1_BASE .set 0x6800
CMPAHR1 .set EPWM1_BASE+0x8
;================================================
HRBUCK_DRV; （可在 ISR 或循环内执行）
;================================================
MOVW DP, #_HRBUCK_In
MOVL XAR2,@_HRBUCK_In ; 指向输入 Q15 占空比（XAR2）
MOVL XAR3,#CMPAHR1 ; 指向 HRPWM CMPA 寄存器（XAR3）
; EPWM1A（HRPWM）的输出
MOV T,*XAR2 ; T <= 占空比
MPYU ACC,T,@_hrbuck_period ; 基于周期的 Q15 到 Q0 缩放
MOV T,@_MEP_ScaleFactor ; MEP 比例因子（来自优化器软件）
MPYU P,T,@AL ; P <= T * AL，优化器缩放
MOVH @AL,P ; AL <= P，将结果移回 ACC
ADD ACC, #0x080 ; MEP 范围和舍入调整
MOVL *XAR3,ACC ; CMPA:CMPAHR(31:8) <= ACC
; EPWM1B（常规分辨率）的输出 可选 - 仅用于比较目的
MOV *+XAR3[2],AH ; 将 ACCH 存储到常规 CMPB
```

使用 R+C 重建滤波器实现 DAC 功能

本示例中，PWM 要求如下：

PWM 频率 = 400kHz（即 TBPRD=250）

PWM 模式 = 非对称向上计数

分辨率 = 14 位（MEP 步长 = 150ps）

图中展示了 DAC 功能和所需的 PWM 波形。如前所述，ePWM1 模块的配置与正常情况几乎相同，只是需要启用 / 选择相应的 MEP 选项。

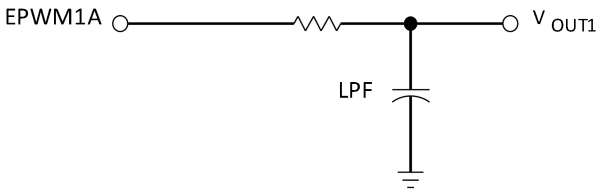



图 15-90 用于基于 PWM 的 DAC 的简单重建滤波器

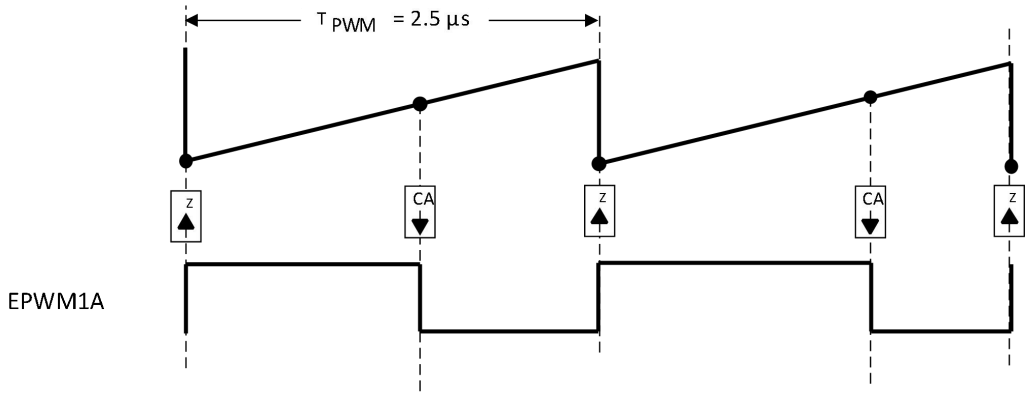



图 15-91 为 PWM DAC 功能生成的 PWM 波形

所示示例代码包括两个主要部分：

- 初始化代码（执行一次）
- 运行时代码（通常在 ISR 内执行）

本示例假设典型的 MEP 步长，且不使用 SFO 库。

示例 15-5 展示了初始化代码。第一部分配置为传统 PWM，第二部分设置 HRPWM 资源。

示例 15-6 展示了可在高速 ISR 循环中执行的运行时代码的汇编示例。

示例 17-5：PWM DAC 功能初始化代码

```
void HrPwmDacDrvCnf(void)
{
// 首先配置传统 PWM
EPwm1Regs.TBCTL.bit.PRDLD = TB_IMMEDIATE; // 设置立即加载
EPwm1Regs.TBPRD = 250; // 周期设置为 400kHz PWM
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // EPWM1 是主设备
hrDAC_period = 250; // 用于 Q15 到 Q0 的缩放
EPwm1Regs.EPWMSYNCOUTEN.all = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
// 注：此处初始化 ChB 仅用于比较目的，非必需
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // 可选
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW; // 可选
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_SET; // 可选
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR; // 可选
// 现在配置 HRPWM 资源
EALLOW; // 注：这些寄存器受保护，仅作用于 ChA
EPwm1Regs.HRCNFG.all = 0x0; // 首先清除所有位
EPwm1Regs.HRCNFG.bit.EDGMODE = HR_FEP; // 控制下降沿位置
```

```
EPwm1Regs.HRCNFG.bit.CTLMODE = HR_CMP; // 由 CMPAHR 控制 MEP
EPwm1Regs.HRCNFG.bit.HRLOAD = HR_CTR_ZERO; // 在 CTR=零时影子加载
EDIS;
MEP_ScaleFactor = 66*256; // 从 100MHz 的典型比例因子开始
// 使用 SFO 函数动态更新 MEP_ScaleFactor
}
```

示例 17-6：PWM DAC 功能运行时代码

```
EPWM1_BASE .set 0x6800
CMPAHR1 .set EPWM1_BASE+0x8
;=================================================
HRPWM_DAC_DRV; （可在 ISR 或循环内执行）
;=================================================
MOVW DP, #_HRDAC_In
MOVL XAR2,@_HRDAC_In ; 指向输入 Q15 占空比（XAR2）
MOVL XAR3,#CMPAHR1 ; 指向 HRPWM CMPA 寄存器（XAR3）
; EPWM1A（HRPWM）的输出
MOV T,*XAR2 ; T <= 占空比
MPY ACC,T,@_hrDAC_period ; 基于周期的 Q15 到 Q0 缩放
ADD ACC,@_HrDAC_period<<15 ; 双极性操作的偏移量
MOV T,@_MEP_ScaleFactor ; MEP 比例因子（来自优化器软件）
MPYU P,T,@AL ; P <= T * AL，优化器缩放
MOVH @AL,P ; AL <= P，将结果移回 ACC
ADD ACC, #0x080 ; MEP 范围和舍入调整
MOVL *XAR3,ACC ; CMPA:CMPAHR(31:8) <= ACC
; EPWM1B（常规分辨率）的输出 可选 - 仅用于比较目的
```

MOV *+XAR3[2],AH ; 将 ACCH 存储到常规 CMPB

## 15.15.2 SFO 库软件 - SFO_TI_Build_V8.lib

表中列出了 SFO_TI_Build_V8.lib 库的核心特性。

表 15-20 SFO_TI_Build_V8.lib 库特性

| | SFO_TI_Build_V8.lib | 单位 |
|---|---|---|
| 完成检查? | 是 | 函数返回值 |
| 在无中断情况下重复调用SFO()以更新 MEP_ScaleFactor所需的典型周期数 | 130,000 | EPWMCLK 周期 |

### 15.15.2.1 比例因子优化器函数 - int SFO ()

该例程驱动微边沿定位器（MEP）校准模块执行 SFO 诊断，并确定器件在任意给定时刻适用的 MEP 比例因子（每个粗粒度 EPWMCLK 周期对应的 MEP 步数）。

若 EPWMCLK = TBCLK = 100MHz，且假设 MEP 步长为 150 皮秒，则在 100MHz 频率下的典型比例因子值为：每个 TBCLK 单位（10 纳秒）对应 66 个 MEP 步。

该函数返回一个 MEP 比例因子值：

MEP_ScaleFactor = 每个 EPWMCLK 周期对应的 MEP 步数

函数使用约束

SFO () 最低支持 EPWMCLK = TBCLK = 50MHz。MEP 诊断逻辑依赖 EPWMCLK 而非 TBCLK，因此 EPWMCLK 的频率限制至关重要。低于 50MHz 时，受器件工艺偏差影响，低温、高内核电压条件下 MEP 步长可能减小，导致 255 个 MEP 步长无法覆盖完整的 EPWMCLK 周期。

可在任意时刻调用 SFO ()，在 MEP 校准模块上执行 SFO 诊断。

函数用法

SFO () 可在 ePWM 通道以 HRPWM 模式运行时在后台调用，其计算的比例因子结果适用于所有 HRPWM 模式下的 ePWM 通道（因函数利用 MEP 校准模块的诊断逻辑，该逻辑独立于 ePWM 通道运行）。

函数返回值含义：

返回 1：校准完成，已计算出新的比例因子；

返回 0：校准仍在进行中；

返回 2：出现错误（MEP_ScaleFactor 超过每个粗粒度 EPWMCLK 周期的最大 255 个精细步长），此时 HRMSTEP 寄存器保留最后一个小于 256 的 MEP 比例因子值，用于自动转换。

占空比限制说明：

未启用高分辨率周期控制时，所有 HRPWM 模式的 ePWM 模块仅存在 3 个 EPWMCLK 周期的最小占空比限制；

启用高分辨率周期控制时，额外增加 PWM 周期结束前 3 个 EPWMCLK 周期的占空比限制。

寄存器协作逻辑：

若设置 HRCNFG[AUTOCONV] = 1（启用自动转换），SFO () 会更新 HRMSTEP 寄存器的比例因子；应用软件仅需在后台运行 SFO () 时，设置 CMPAHR = 小数部分(PWM 占空比×PWM 周期) << 8、CMPBHR = 小数部分(PWM 占空比×PWM 周期) << 8 或 TBPRDHR = 小数部分(PWM 周期)，MEP 校准模块会自动计算对应的 MEP 步数并调整高分辨率 ePWM 信号边缘。

若清除 HRCNFG[AUTOCONV]（禁用自动转换），HRMSTEP 寄存器失效，应用软件需手动执行计算：

小数部分占空比周期比例因子

TBPHSHR、CMPBHR、DBREDHR 和 DBFEDHR 遵循相同逻辑；使用 TBPRDHR 时必须启用自动转换。

以下代码片段展示如何通过 driverlib 函数实现 HRPWM 占空比控制：

```
float32_t dutyFine = 85.62;
float32_t count = (dutyFine * (float32_t)(EPWM_TIMER_TBPRD << 8))/100;
uint32_t compCount = (count);
HRPWM_setCounterCompareValue(EPWM1_BASE,
HRPWM_COUNTER_COMPARE_A, compCount);
HRPWM_setCounterCompareValue(EPWM1_BASE,
HRPWM_COUNTER_COMPARE_B, compCount);
```

该程序可作为后台任务在低速循环中运行，仅需消耗极少的 CPU 周期。SFO 功能所需的执行重复频率取决于应用程序的运行环境。与所有数字 CMOS 器件一样，温度和电源电压的变化会对 MEP 的工作产生影响。不过，在大多数应用中，这些参数变化较为缓慢，因此通常每 5 至 10 秒执行一次 SFO 功能即可满足需求。若预期参数变化更为迅速，则可提高执行频率以适配应用场景。需要注意的是，SFO 功能的重复频率并无上限限制，因此其执行速度可达到后台循环所能支持的最快速率。

在使用 HRPWM 功能时，HRPWM 逻辑在 PWM 周期的前 3 个 EPWMCLK 周期内不工作（若使用 TBPRDHR，则 PWM 周期的最后 3 个 EPWMCLK 周期也不工作）。在该配置下运行应用程序时，若高分辨率周期控制被禁用（HRPCTL[HRPE=0]）且 CMPA/CMPB 寄存器值小于 3 个周期，则必须将 CMPAHR/CMPBHR 寄存器清零。若高分辨率周期控制已启用（HRPCTL[HRPE=1]），CMPA 寄存器值不得低于 3 或高于 TBPRD-3。这样可避免 PWM 信号出现任何意外跳变。

### 15.15.2.2 软件使用步骤

软件库函数 SFO () 为系列 ePWM 模块的 HRPWM 功能计算 MEP 比例因子，该因子为 1-255 的整数，代表系统时钟周期内可用的微步边缘位置数量，存储于整数变量 MEP_ScaleFactor 中（示例见表）。

表 15-21 比例因子值说明

| 软件函数调用 | 功能描述 | 更新的变量 |
| --- | --- | --- |
| SFO() | 计算 MEP 比例因子并写入 HRMSTEP寄存器 | MEP_ScaleFactor 和 HRMSTEP 寄存器 |

建议按以下步骤使用 HRPWM 功能的 SFO 函数：

步骤 1：添加头文件

需按如下方式包含 SFO_V8.h 文件。在使用 SFO 库函数时，该头文件为必选文件。要使 SFO() 函数正常运行，项目中必须包含对应的(Device)_Device.h 和(Device)_Epwm_defines.h 文件。若最终应用中使用了自定义头文件，则这些头文件为可选包含项。

8.示例 15-7：添加 “头文件” 的示例方法

```
#include "F28x7x_Device.h"       // F28x7x 系列器件头文件
#include "F28x7x_EPwm_defines.h"   // ePWM 相关定义
#include "SFO_V8.h"                  // SFO 库函数（HRPWM 功能必需）
```

步骤 2：变量声明

如下所示，声明一个用于存储比例因子值的整型变量。

9.示例 15-8：声明变量

```
int MEP_ScaleFactor = 0;   // 比例因子存储变量
volatile struct EPWM_REGS *ePWM[] = {0, &EPwm1Regs, &EPwm2Regs, &EPwm3Regs, &EPwm4Regs};
```

步骤 3：MEP_ScaleFactor 初始化

SFO () 无需 MEP_ScaleFactor 提供初始值。在应用代码中使用 MEP_ScaleFactor 前，需调用 SFO () 驱动 MEP 校准模块计算比例因子。

在使用 MEP_ScaleFactor 之前，作为一次性初始化代码的一部分，请包含以下内容：

10.示例 15-9：使用比例系数进行初始化

```
// 使用函数 SFO()初始化 MEP_ScaleFactor
```

while (SFO() == 0) {}  // MEP 校准模块完成 MEP_ScaleFactor 计算

步骤 4：应用代码实现

应用程序运行期间，器件温度与供电电压均可能出现波动。为确保各 ePWM 模块均采用合适的比例因子，可将比例因子优化（SFO）函数作为慢速后台循环的一部分，定期重新运行。下文将展示相关示例。

注：

请参阅器件专用 C/C++头文件中的 HRPWM_SFO 示例。

11.示例 15-10：SFO 函数调用

```
main()
{
int status;
// 用户代码：ePWM1、2、3、4 配置为 HRPWM 模式
// 当 MEP 校准模块使用 SFO 检测计算出新的 MEP_ScaleFactor
// status 返回 1
status = SFO();
if(status==2) {ESTOP0;} // MEP_ScaleFactor 超过最大允许值 255
// 函数返回 2（错误情况）
}
```

# 15.16 寄存器描述

## 15.16.1 ePWM 基地址表

表中列出了 ePWM 基地址。

表 15-22 ePWM 基地址表

| 位域名称 | | DriverLib 名称 | 基地址 | 流水线保护 |
|---|---|---|---|---|
| 实例 | 结构 | | | |
| EPwm1Regs | EPWM_REGS | EPWM1_BASE | 0x0061_4000 | 是 |
| EPwm2Regs | EPWM_REGS | EPWM2_BASE | 0x0061_4200 | 是 |
| EPwm3Regs | EPWM_REGS | EPWM3_BASE | 0x0061_4400 | 是 |

| 位域名称 | | DriverLib 名称 | 基地址 | 流水线保护 |
|---|---|---|---|---|
| 实例 | 结构 | | | |
| EPwm4Regs | EPWM_REGS | EPWM4_BASE | 0x0061_4600 | 是 |
| EPwm5Regs | EPWM_REGS | EPWM5_BASE | 0x0061_4800 | 是 |
| EPwm6Regs | EPWM_REGS | EPWM6_BASE | 0x0061_4A00 | 是 |
| EPwm7Regs | EPWM_REGS | EPWM7_BASE | 0x0061_4C00 | 是 |

## 15.16.2 EPWM_REGS 寄存器

表中列出了 EPWM_REGS 寄存器的内存映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，不应修改寄存器内容。

表 15-23 EPWM_REGS 寄存器

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 0x0000 | TBCTL | 时基控制寄存器 |
| 0x0002 | TBCTL2 | 时基控制寄存器 2 |
| 0x0006 | EPWMSYNCINSEL | EPWMxSYNCIN 源选择寄存器 |
| 0x0008 | TBCTR | 时基计数器寄存器 |
| 0x000A | TBSTS | 时基状态寄存器 |
| 0x000C | EPWMSYNCOUTEN | EPWMxSYNCOUT 源使能寄存器 |
| 0x000E | TBCTL3 | 时基控制寄存器 3 |
| 0x0010 | CMPCTL | 计数器比较控制寄存器 |
| 0x0012 | CMPCTL2 | 计数器比较控制寄存器 2 |
| 0x0018 | DBCTL | 死区发生器控制寄存器 |
| 0x001A | DBCTL2 | 死区发生器控制寄存器 2 |
| 0x0020 | AQCTL | 动作限定器控制寄存器 |
| 0x0022 | AQTSRCSEL | 动作限定器触发事件源选择寄存器 |
| 0x0028 | PCCTL | PWM 斩波器控制寄存器 |
| 0x0030 | VCAPCTL | 谷值捕获控制寄存器 |
| 0x0032 | VCNTCFG | 谷值计数器配置寄存器 |

| 偏移量 | 缩写 | 寄存器名称 |
| --- | --- | --- |
| 0x0040 | HRCNFG | HRPWM 配置寄存器 |
| 0x0042 | HRPWR | HRPWM 电源寄存器 |
| 0x004C | HRMSTEP | HRPWM MEP 步长寄存器 |
| 0x004E | HRCNFG2 | HRPWM 配置寄存器 2 |
| 0x005A | HRPCTL | 高分辨率周期控制寄存器 |
| 0x005C | TRREM | HRPWM 高分辨率余数寄存器 |
| 0x0068 | GLDCTL | 全局 PWM 加载控制寄存器 |
| 0x006A | GLDCFG | 全局 PWM 加载配置寄存器 |
| 0x0070 | EPWMXLINK | EPWMx 链接寄存器 |
| 0x0080 | AQCTLA | 动作限定器输出 A 控制寄存器 |
| 0x0082 | AQCTLA2 | 动作限定器输出 A 附加控制寄存器 |
| 0x0084 | AQCTLB | 动作限定器输出 B 控制寄存器 |
| 0x0086 | AQCTLB2 | 动作限定器输出 B 附加控制寄存器 |
| 0x008E | AQSFRC | 动作限定器软件强制寄存器 |
| 0x0092 | AQCSFRC | 动作限定器连续软件强制寄存器 |
| 0x00A0 | DBREDHR | 死区上升沿延迟高分辨率镜像寄存器 |
| 0x00A2 | DBRED | 死区上升沿延迟计数寄存器 |
| 0x00A4 | DBFEDHR | 死区下降沿延迟高分辨率寄存器 |
| 0x00A6 | DBFED | 死区下降沿延迟计数寄存器 |
| 0x00C0 | TBPHS | 时基相位寄存器 |
| 0x00C4 | TBPRDHR | 时基周期高分辨率寄存器 |
| 0x00C6 | TBPRD | 时基周期寄存器 |
| 0x00D4 | CMPA | 计数器比较 A 寄存器 |

| 偏移量 | 缩写 | 寄存器名称 |
| --- | --- | --- |
| 0x00D8 | CMPB | 计数器比较 B 寄存器 |
| 0x00DE | CMPC | 计数器比较 C 寄存器 |
| 0x00E2 | CMPD | 计数器比较 D 寄存器 |
| 0x00E8 | GLDCTL2 | 全局 PWM 加载控制寄存器 2 |
| 0x00EE | SWVDELVAL | 软件谷值模式延迟寄存器 |
| 0x0100 | TZSEL | 跳闸区选择寄存器 |
| 0x0104 | TZDCSEL | 跳闸区数字比较选择寄存器 |
| 0x0108 | TZCTL | 跳闸区控制寄存器 |
| 0x010A | TZCTL2 | 附加跳闸区控制寄存器 |
| 0x010C | TZCTLDCA | 数字比较 A 跳闸区控制寄存器 |
| 0x010E | TZCTLDCB | 数字比较 B 跳闸区控制寄存器 |
| 0x011A | TZEINT | 跳闸区中断使能寄存器 |
| 0x0126 | TZFLG | 跳闸区标志寄存器 |
| 0x0128 | TZCBCFLG | 逐周期跳闸标志寄存器 |
| 0x012A | TZOSTFLG | 单触发跳闸标志寄存器 |
| 0x012E | TZCLR | 跳闸区清除寄存器 |
| 0x0130 | TZCBCCLR | 逐周期跳闸清除寄存器 |
| 0x0132 | TZOSTCLR | 单触发跳闸清除寄存器 |
| 0x0136 | TZFRC | 跳闸区强制寄存器 |
| 0x0148 | ETSEL | 事件触发选择寄存器 |
| 0x014C | ETPS | 事件触发预分频寄存器 |
| 0x0150 | ETFLG | 事件触发标志寄存器 |
| 0x0154 | ETCLR | 事件触发清除寄存器 |

| 偏移量 | 缩写 | 寄存器名称 |
| --- | --- | --- |
| 0x0158 | ETFRC | 事件触发强制寄存器 |
| 0x015C | ETINTPS | 事件触发中断预分频寄存器 |
| 0x0160 | ETSOCPS | 事件触发 SOC 预分频寄存器 |
| 0x0164 | ETCNTINITCTL | 事件触发计数器初始化控制寄存器 |
| 0x0168 | ETCNTINIT | 事件触发计数器初始化寄存器 |
| 0x0180 | DCTRIPSEL | 数字比较跳闸选择寄存器 |
| 0x0186 | DCACTL | 数字比较 A 控制寄存器 |
| 0x0188 | DCBCTL | 数字比较 B 控制寄存器 |
| 0x018E | DCFCTL | 数字比较滤波控制寄存器 |
| 0x0190 | DCCAPCTL | 数字比较捕获控制寄存器 |
| 0x0192 | DCFOFFSET | 数字比较滤波偏移寄存器 |
| 0x0194 | DCFOFFSETCNT | 数字比较滤波偏移计数器 |
| 0x0196 | DCFWINDOW | 数字比较滤波窗口寄存器 |
| 0x0198 | DCFWINDOWCNT | 数字比较滤波窗口计数器 |
| 0x019E | DCCAP | 数字比较计数器捕获寄存器 |
| 0x01A4 | DCAHTRIPSEL | 数字比较 AH 跳闸选择寄存器 |
| 0x01A6 | DCALTRIPSEL | 数字比较 AL 跳闸选择寄存器 |
| 0x01A8 | DCBHTRIPSEL | 数字比较 BH 跳闸选择寄存器 |
| 0x01AA | DCBLTRIPSEL | 数字比较 BL 跳闸选择寄存器 |
| 0x01F4 | EPWMLOCK | EPWM 锁定寄存器 |
| 0x01FA | HWVDELVAL | 硬件谷值模式延迟寄存器 |
| 0x01FC | VCNTVAL | 硬件谷值计数器寄存器 |

复杂的位访问类型经过编码以适应小型表格单元格。表中显示了本节中用于访问类型的编码。

表 15-24 EPWM_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R-0 | 读 返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1S | W1S | 写 1 置位 |
| WSonce | WSonce | 一次性写（仅能置位一次） |
| 复位或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i,k,l,m,n | | 当这些变量用于寄存器名称、偏移地址或地址时，指代寄存器数组的值，该寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组以公式表示。 |
| y | | 当该变量用于寄存器名称、偏移地址或地址时，指代寄存器数组的值。 |

## 15.16.2.1 TBCTL 寄存器

偏移地址：0h　复位值：0083h

时基控制寄存器

表 15-25 TBCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FREE_SOFT | | PHSDIR | CLKDIV | | | HSPCLKDIV | |
| R/W-0h | | R/W-0h | R/W-0h | | | R/W-1h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HSPCLKDIV | SWFSYNC | 保留 | | PRDLD | PHSEN | CTRMODE | |
| R/W-1h | R-0/W1S-0h | R-0h | | R/W-0h | R/W-0h | R/W-3h | |

表 15-26 TBCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | FREE_SOFT | R/W | 0h | 仿真模式位。选择仿真事件期间 ePWM 时基计数器的行为。<br>00：在下一次时基计数器递增或递减后停止。<br>01：计数器完成一个完整周期后停止：向上计数模式在 TBCTR = TBPRD 时停止；向下计数模 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 式在 TBCTR = 0x00 时停止；上下计数模式在 TBCTR = 0x00 时停止。<br>1x：自由运行。<br>复位类型：SYSRSn |
| 13 | PHSDIR | R/W | 0h | 相位方向位。仅用于时基计数器的上下计数模式，指定同步事件发生并从相位寄存器 TBPHS 装载新相位值后 TBCTR 的计数方向，与同步事件前的计数方向无关。向上计数和向下计数模式忽略此位。<br>0：同步事件后向下计数。<br>1：同步事件后向上计数。<br>复位类型：SYSRSn |
| 12-10 | CLKDIV | R/W | 0h | 时基时钟预分频位。选择时基时钟预分频值：TBCLK = EPWMCLK/(HSPCLKDIV * CLKDIV)。<br>000：/1（复位默认值）<br>001：/2<br>010：/4<br>011：/8<br>100：/16<br>101：/32<br>110：/64<br>111：/128<br>复位类型：SYSRSn |
| 9-7 | HSPCLKDIV | R/W | 1h | 高速时基时钟预分频位。决定时基时钟预分频值的一部分：TBCLK = EPWMCLK / (HSPCLKDIV x CLKDIV)。<br>000：/1<br>001：/2（复位默认值）<br>010：/4<br>011：/6<br>100：/8<br>101：/10<br>110：/12<br>111：/14<br>复位类型：SYSRSn |
| 6 | SWFSYNC | R-0/W1S | 0h | 软件强制同步脉冲。<br>0：写 0 无效；读取始终返回 0。<br>1：写 1 强制产生一次同步脉冲。<br>设置 EPWMSYNCOUTEN.SWEN 位，可使 SWFSYNC 影响 EPWMxSYNCO。<br>复位类型：SYSRSn |
| 5-4 | RESERVED | R | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 3 | PRDLD | R/W | 0h | 活动周期寄存器从影子寄存器装载的选择。<br>0：当 TBCTR 等于零和 / 或发生同步事件时（由 TBCTL2[PRDLDSYNC] 决定），将影子寄存器装载至周期寄存器 TBPRD。读写 TBPRD 访问影子寄存器。<br>1：立即模式（旁路影子寄存器）。读写 TBPRD 直接访问活动寄存器。<br>复位类型：SYSRSn |
| 2 | PHSEN | R/W | 0h | 允许从相位寄存器装载计数器。<br>0：不从时基相位寄存器 TBPHS 装载时基计数器 TBCTR。<br>1：当 EPWMxSYNCI 输入信号或软件强制同步信号发生时，允许从 TBPHS 装载计数器，并允许影子到活动寄存器的装载事件，参见位 6。<br>复位类型：SYSRSn |
| 1-0 | CTRMODE | R/W | 3h | 计数器模式。时基计数器模式通常只配置一次，正常运行时不更改。更改模式将在下一个 TBCLK 边沿生效，计数器从模式更改前的当前值继续递增或递减。<br>00：向上计数。<br>01：向下计数。<br>10：上下计数。<br>11：冻结计数器（复位默认值）。<br>复位类型：SYSRSn |

### 15.16.2.2 TBCTL2 寄存器

偏移地址：2h　复位值：0000h

时基控制寄存器 2

表 15-27 TBCTL2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| PRDLDSYNC | | 保留 | | 保留 | | | |
| R/W-0h | | R-0h | | R-0-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OSHTSYNC | OSHTSYNCMODE | 保留 | 保留 | | | | |
| R-0/W1S-0h | R/W-0h | R/W-0h | R-0-0h | | | | |

表 15-28 TBCTL2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | PRDLDSYNC | R/W | 0h | SYNC 事件触发周期寄存器从影子到活动的装载。<br>00：仅在 TBCTR = 0 时装载 TBPRD（与旧版相同）。<br>01：在 TBCTR = 0 或发生 SYNC 时均装载 TBPRD。<br>10：仅在收到 SYNC 时装载 TBPRD。<br>11：保留。<br>注意：仅在 TBCTL[PRDLD]=0 时有效。<br>复位类型：SYSRSn |
| 13-12 | RESERVED | R | 0h | 保留 |
| 11-8 | RESERVED | R-0 | 0h | 保留 |
| 7 | OSHTSYNC | R-0/W1S | 0h | 单次同步位。<br>0：写 0 无效。<br>1：允许一个同步脉冲通过。<br>复位类型：SYSRSn |
| 6 | OSHTSYNCMODE | R/W | 0h | 单次同步使能位。<br>0：禁用单次同步模式。<br>1：使能单次同步模式。<br>复位类型：SYSRSn |
| 5 | RESERVED | R/W | 0h | 保留 |
| 4-0 | RESERVED | R-0 | 0h | 保留 |

### 15.16.2.3 EPWMSYNCINSEL 寄存器

偏移地址：6h　复位值：0001h

EPWMxSYNCIN 源选择寄存器

表 15-29 EPWMSYNCINSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | SEL | | | | |
| R-0h | | | R/W-1h | | | | |

表 15-30 EPWMSYNCINSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-5 | RESERVED | R | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 4-0 | SEL | R/W | 1h | 这些位决定 EPWMxSYNCI 信号源。<br>0x00：禁用。<br>其他值见“ePWM 同步选择”表。<br>复位类型：SYSRSn |

### 15.16.2.4 TBCTR 寄存器

偏移地址：8h　复位值：0000h

时基计数器寄存器

表 15-31 TBCTR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| TBCTR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TBCTR | | | | | | | |
| R/W-0h | | | | | | | |

表 15-32 TBCTR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | TBCTR | R/W | 0h | 时基计数器寄存器。<br>复位类型：SYSRSn |

### 15.16.2.5 TBSTS 寄存器

偏移地址：Ah　复位值：0001h

时基状态寄存器

表 15-33 TBSTS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | CTRMAX | SYNCI | CTRDIR |
| R-0-0h | | | | | R/W1C-0h | R/W1C-0h | R-1h |

表 15-34 TBSTS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-3 | RESERVED | R-0 | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 2 | CTRMAX | R/W1C | 0h | 时基计数器最大值锁存状态位。<br>0：读到 0 表示时基计数器从未达到最大值；写 0 无效。<br>1：读到 1 表示时基计数器已达到最大值 0xFFFF；写 1 清除锁存的事件。<br>复位类型：SYSRSn |
| 1 | SYNCI | R/W1C | 0h | 输入同步锁存状态位。<br>0：写 0 无效；读到 0 表示未发生外部同步事件。<br>1：读到 1 表示已发生外部同步事件 EPWMxSYNCI；写 1 清除锁存的事件。<br>复位类型：SYSRSn |
| 0 | CTRDIR | R | 1h | 时基计数器方向状态位。<br>0：当前向下计数。<br>1：当前向上计数。<br>注意：仅在计数器未冻结时有效。<br>复位类型：SYSRSn |

### 15.16.2.6 EPWMSYNCOUTEN 寄存器

偏移地址：Ch　复位值：0001h

EPWMxSYNCOUT 源使能寄存器

表 15-35 EPWMSYNCOUTEN 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | DCBEVT1EN | DCAEVT1EN | CMPDEN | CMPCEN | CMPBEN | ZEROEN | SWEN |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-1h |

表 15-36 EPWMSYNCOUTEN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留 |
| 7 | RESERVED | R | 0h | 保留 |
| 6 | DCBEVT1EN | R/W | 0h | 使能由 DCBEVT1.sync 事件置位 EPWMxSYNCO 信号。<br>0：禁用。<br>1：发生 DCBEVT1.sync 时，EPWMxSYNCO |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 输出持续一个 PWM 时钟周期的脉冲。<br>复位类型：SYSRSn |
| 5 | DCAEVT1EN | R/W | 0h | 使能由 DCAEVT1.sync 事件置位 EPWMxSYNCOUT 信号。<br>0：禁用。<br>1：发生 DCAEVT1.sync 时，EPWMxSYNCOUT 输出持续一个 PWM 时钟周期的脉冲。<br>复位类型：SYSRSn |
| 4 | CMPDEN | R/W | 0h | 使能由 TBCTR = CMPD 事件置位 EPWMxSYNCO 信号。<br>0：禁用。<br>1：发生 TBCTR = CMPD 时，EPWMxSYNCO 输出持续一个 PWM 时钟周期的脉冲。<br>复位类型：SYSRSn |
| 3 | CMPCEN | R/W | 0h | 使能由 TBCTR = CMPC 事件置位 EPWMxSYNCO 信号。<br>0：禁用。<br>1：发生 TBCTR = CMPC 时，EPWMxSYNCO 输出持续一个 PWM 时钟周期的脉冲。<br>复位类型：SYSRSn |
| 2 | CMPBEN | R/W | 0h | 使能由 TBCTR = CMPB 事件置位 EPWMxSYNCO 信号。<br>0：禁用。<br>1：发生 TBCTR = CMPB 时，EPWMxSYNCO 输出持续一个 PWM 时钟周期的脉冲。<br>复位类型：SYSRSn |
| 1 | ZEROEN | R/W | 0h | 使能由 TBCTR 变为 0x0000 事件置位 EPWMxSYNCOUT 信号。<br>0：禁用。<br>1：发生 TBCTR 变为 0x0000 时，EPWMxSYNCOUT 输出持续一个 PWM 时钟周期的脉冲。<br>复位类型：SYSRSn |
| 0 | SWEN | R/W | 1h | 使能由 TBCTL.SWFSYNC 位置位 事件置位 EPWMxSYNCO 信号。<br>0：禁用。<br>1：发生 TBCTL.SWFSYNC 位置位 时，EPWMxSYNCO 输出持续一个 PWM 时钟周期的脉冲。<br>复位类型：SYSRSn |

### 15.16.2.7 TBCTL3 寄存器

偏移地址：Eh　复位值：0000h

时基控制寄存器 3

表 15-37 TBCTL3 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | OSSFRCEN |
| R-0h | | | | | | | R/W-0h |

表 15-38 TBCTL3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-1 | RESERVED | R | 0h | 保留 |
| 0 | OSSFRCEN | R/W | 0h | 决定由哪个位置位 EPWMxSYNCOUT 单次锁存器。<br>0：由 TBCTL2[OSHTSYNC] 置位。<br>1：由 GLDCTL2[OSHTLD] 置位。<br>复位类型：SYSRSn |

### 15.16.2.8 CMPCTL 寄存器

偏移地址：10h　复位值：0000h

计数比较控制寄存器

表 15-39 CMPCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | LOADBSYNC | | LOADASYNC | | SHDWBFULL | SHDWAFULL |
| R-0-0h | | R/W-0h | | R/W-0h | | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | SHDWBMODE | 保留 | SHDWAMODE | LOADBMODE | | LOADAMODE | |
| R-0-0h | R/W-0h | R-0-0h | R/W-0h | R/W-0h | | R/W-0h | |

表 15-40 CMPCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | RESERVED | R-0 | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 13-12 | LOADBSYNC | R/W | 0h | SYNC 事件触发 CMPB 从影子到活动寄存器的装载。<br>00：按 LOADBMODE（位 1、0，与旧版相同）装载 CMPB:CMPBHR。<br>01：按 LOADBMODE 或发生 SYNC 时均装载 CMPB:CMPBHR。<br>10：仅在收到 SYNC 时装载 CMPB:CMPBHR。<br>11：保留。<br>注意：仅在 CMPCTL[SHDWBMODE] = 0 时有效。<br>复位类型：SYSRSn |
| 11-10 | LOADASYNC | R/W | 0h | SYNC 事件触发 CMPA 从影子到活动寄存器的装载。<br>00：按 LOADAMODE（位 1、0，与旧版相同）装载 CMPA:CMPAHR。<br>01：按 LOADAMODE 或发生 SYNC 时均装载 CMPA:CMPAHR。<br>10：仅在收到 SYNC 时装载 CMPA:CMPAHR。<br>11：保留。<br>注意：仅在 CMPCTL[SHDWAMODE] = 0 时有效。<br>复位类型：SYSRSn |
| 9 | SHDWBFULL | R | 0h | 计数比较 B（CMPB）影子寄存器满状态标志。装载选通信号发生后此位自动清除。<br>0：CMPB 影子寄存器尚未满。<br>1：CMPB 影子寄存器已满，CPU 写操作将覆盖当前影子值。<br>复位类型：SYSRSn |
| 8 | SHDWAFULL | R | 0h | 计数比较 A（CMPA）影子寄存器满状态标志。对 CMPA:CMPAHR 执行 32 位写或对 CMPA 执行 16 位写时置位；对 CMPAHR 执行 16 位写不影响此标志。装载选通信号发生后此位自动清除。<br>0：CMPA 影子寄存器尚未满。<br>1：CMPA 影子寄存器已满，CPU 写操作将覆盖当前影子值。<br>复位类型：SYSRSn |
| 7 | RESERVED | R-0 | 0h | 保留 |
| 6 | SHDWBMODE | R/W | 0h | 计数比较 B（CMPB）寄存器工作模式。<br>0：影子模式，采用双缓冲；CPU 写操作访问影子寄存器。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：立即模式，仅使用活动比较寄存器；CPU 读写直接访问活动寄存器，立即执行比较动作。<br>复位类型：SYSRSn |
| 5 | RESERVED | R-0 | 0h | 保留 |
| 4 | SHDWAMODE | R/W | 0h | 计数比较 A（CMPA）寄存器工作模式。<br>0：影子模式，采用双缓冲；CPU 写操作访问影子寄存器。<br>1：立即模式，仅使用活动比较寄存器；CPU 读写直接访问活动寄存器，立即执行比较动作。<br>复位类型：SYSRSn |
| 3-2 | LOADBMODE | R/W | 0h | 活动计数比较 B（CMPB）从影子寄存器装载的选择模式。<br>00：在 CTR = Zero（TBCTR = 0x0000）时装载。<br>01：在 CTR = PRD（TBCTR = TBPRD）时装载。<br>10：在 CTR = Zero 或 CTR = PRD 时装载。<br>11：冻结（禁止装载）。<br>注意：立即模式（CMPCTL[SHDWBMODE] = 1）下无效。<br>复位类型：SYSRSn |
| 1-0 | LOADAMODE | R/W | 0h | 活动计数比较 A（CMPA）从影子寄存器装载的选择模式。<br>00：在 CTR = Zero（TBCTR = 0x0000）时装载。<br>01：在 CTR = PRD（TBCTR = TBPRD）时装载。<br>10：在 CTR = Zero 或 CTR = PRD 时装载。<br>11：冻结（禁止装载）。<br>注意：立即模式（CMPCTL[SHDWAMODE] = 1）下无效。<br>复位类型：SYSRSn |

### 15.16.2.9 CMPCTL2 寄存器

偏移地址：12h　复位值：0000h

计数比较控制寄存器 2

表 15-41 CMPCTL2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | LOADDSYNC | | LOADCSYNC | | 保留 | |
| R-0-0h | | R/W-0h | | R/W-0h | | R-0-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | SHDWDMODE | 保留 | SHDWCMODE | LOADDMODE | | LOADCMODE | |
| R-0-0h | R/W-0h | R-0-0h | R/W-0h | R/W-0h | | R/W-0h | |

表 15-42 CMPCTL2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | RESERVED | R-0 | 0h | 保留 |
| 13-12 | LOADDSYNC | R/W | 0h | SYNC 事件触发 CMPD 从影子到活动寄存器的装载。<br>00：按 LOADDMODE装载 CMPD。<br>01：按 LOADDMODE 或发生 SYNC 时均装载 CMPD。<br>10：仅在收到 SYNC 时装载 CMPD。<br>11：保留。<br>注意：仅在 CMPCTL2[SHDWDMODE] = 0 时有效。<br>复位类型：SYSRSn |
| 11-10 | LOADCSYNC | R/W | 0h | SYNC 事件触发 CMPC 从影子到活动寄存器的装载。<br>00：按 LOADCMODE装载 CMPC。<br>01：按 LOADCMODE 或发生 SYNC 时均装载 CMPC。<br>10：仅在收到 SYNC 时装载 CMPC。<br>11：保留。<br>注意：仅在 CMPCTL2[SHDWCMODE] = 0 时有效。<br>复位类型：SYSRSn |
| 9-7 | RESERVED | R-0 | 0h | 保留 |
| 6 | SHDWDMODE | R/W | 0h | 计数比较 D（CMPD）寄存器工作模式。<br>0：影子模式，采用双缓冲；CPU 写操作访问影子寄存器。<br>1：立即模式，仅使用活动比较寄存器；CPU 读写直接访问活动寄存器，立即执行比较动作。<br>复位类型：SYSRSn |
| 5 | RESERVED | R-0 | 0h | 保留 |
| 4 | SHDWCMODE | R/W | 0h | 计数比较 C（CMPC）寄存器工作模式。<br>0：影子模式，采用双缓冲；CPU 写操作访问影子寄存器。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：立即模式，仅使用活动比较寄存器；CPU 读写直接访问活动寄存器，立即执行比较动作。<br>复位类型：SYSRSn |
| 3-2 | LOADDMODE | R/W | 0h | 活动计数比较 D（CMPD）从影子寄存器装载的选择模式。<br>00：在 CTR = Zero（TBCTR = 0x0000）时装载。<br>01：在 CTR = PRD（TBCTR = TBPRD）时装载。<br>10：在 CTR = Zero 或 CTR = PRD 时装载。<br>11：冻结（禁止装载）。<br>注意：立即模式下无效。<br>复位类型：SYSRSn |
| 1-0 | LOADCMODE | R/W | 0h | 活动计数比较 C（CMPC）从影子寄存器装载的选择模式。<br>00：在 CTR = Zero（TBCTR = 0x0000）时装载。<br>01：在 CTR = PRD（TBCTR = TBPRD）时装载。<br>10：在 CTR = Zero 或 CTR = PRD 时装载。<br>11：冻结（禁止装载）。<br>注意：立即模式下无效。<br>复位类型：SYSRSn |

## 15.16.2.10 DBCTL 寄存器

偏移地址：18h　复位值：0000h

死区发生器控制寄存器

表 15-43 DBCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| HALFCYCLE | DEDB_MODE | OUTSWAP | | SHDWDBFEDMODE | SHDWDBREDMODE | LOADFEDMODE | |
| R/W-0h | R/W-0h | R/W-0h | | R/W-0h | R/W-0h | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| LOADREDMODE | | IN_MODE | | POLSEL | | OUT_MODE | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 15-44 DBCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | HALFCYCLE | R/W | 0h | 半周期时钟使能。<br>0：使能全周期时钟，死区计数器以 TBCLK 频 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 率工作。<br>1：使能半周期时钟，死区计数器以 TBCLK*2 频率工作。<br>复位类型：SYSRSn |
| 14 | DEDB_MODE | R/W | 0h | 死区双边沿 B 模式控制（S8 开关）。<br>0：仅在 A 信号路径中，对 S4 开关（IN-MODE 位）所选的 InA/InB 施加上升沿延迟；仅在 B 信号路径中，对 S5 开关（INMODE 位）所选的 InA/InB 施加下降沿延迟。<br>1：对 S4 开关（INMODE 位）所选信号同时施加上升沿和下降沿延迟，仅输出到 B 信号路径。<br>注意：此位置 1 时，必须将 OUT_MODE 配置为 Apath = InA，或将 OUTSWAP 配置为 OutA=Bpath，否则 OutA 无效。<br>复位类型：SYSRSn |
| 13-12 | OUTSWAP | R/W | 0h | 死区输出交换控制。位 13 控制 S6 开关，位 12 控制 S7 开关。<br>00：OutA 和 OutB 按 OUT-MODE 位定义。<br>01：OutA = OUT-MODE 定义的 A 路径；OutB = OUT-MODE 定义的 A 路径（上升沿延迟或旁路延迟的 A 信号路径）。<br>10：OutA = OUT-MODE 定义的 B 路径（下降沿延迟或旁路延迟的 B 信号路径）；OutB = OUT-MODE 定义的 B 路径。<br>11：OutA = OUT-MODE 定义的 B 路径（下降沿延迟或旁路延迟的 B 信号路径）；OutB = OUT-MODE 定义的 A 路径（上升沿延迟或旁路延迟的 A 信号路径）。<br>复位类型：SYSRSn |
| 11 | SHDWDBFEDMODE | R/W | 0h | FED 死区装载模式。<br>0：立即模式，仅使用活动 DBFED 寄存器，CPU 读写直接访问活动寄存器，立即执行 FED 死区动作。<br>1：影子模式，采用双缓冲，CPU 写操作访问影子寄存器。复位默认使用立即模式，以兼容旧版。<br>复位类型：SYSRSn |
| 10 | SHDWDBREDMODE | R/W | 0h | RED 死区装载模式。<br>0：立即模式，仅使用活动 DBRED 寄存器，CPU 读写直接访问活动寄存器，立即执行 RED 死区动作。<br>1：影子模式，采用双缓冲，CPU 写操作访问影子寄存器。复位默认使用立即模式，以兼容旧 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 版。<br>复位类型：SYSRSn |
| 9-8 | LOADFEDMODE | R/W | 0h | 活动 DBFED 从影子寄存器装载的选择模式。<br>00：计数器 = 0（CNT_eq）时装载。<br>01：计数器 = 周期（PRD_eq）时装载。<br>10：计数器 = 0 或周期时装载。<br>11：冻结（禁止装载）。<br>注意：立即模式下无效。<br>复位类型：SYSRSn |
| 7-6 | LOADREDMODE | R/W | 0h | 活动 DBRED 从影子寄存器装载的选择模式。<br>00：计数器 = 0（CNT_eq）时装载。<br>01：计数器 = 周期（PRD_eq）时装载。<br>10：计数器 = 0 或周期时装载。<br>11：冻结（禁止装载）。<br>注意：立即模式下无效。<br>复位类型：SYSRSn |
| 5-4 | IN_MODE | R/W | 0h | 死区输入模式控制。位 5 控制 S5 开关，位 4 控制 S4 开关，用于选择下降沿和上升沿延迟的输入源。典型死区波形默认以 EPWMxA In 同时作为两种延迟的输入源。<br>00：来自动作限定器的 EPWMxA In 同时用于下降沿和上升沿延迟。<br>01：EPWMxB In 用于上升沿延迟，EPWMxA In 用于下降沿延迟，均来自动作限定器。<br>10：EPWMxA In 用于上升沿延迟，EPWMxB In 用于下降沿延迟，均来自动作限定器。<br>11：来自动作限定器的 EPWMxB In 同时用于上升沿和下降沿延迟。<br>复位类型：SYSRSn |
| 3-2 | POLSEL | R/W | 0h | 极性选择控制。位 3 控制 S3 开关，位 2 控制 S2 开关，允许在信号送出死区子模块前选择性地将其中一路延迟信号反相。以下说明对应数字电机控制逆变器单个桥臂的典型上 / 下管控制，假设 DBCTL[OUT_MODE] = 1,1 且 DBCTL[IN_MODE] = 0x0。也可采用其他增强模式，但不属于典型用法。<br>00：高电平有效（AH），EPWMxA 和 EPWMxB 均不反相（默认）。<br>01：低电平有效互补（ALC），EPWMxA 反相。<br>10：高电平有效互补（AHC），EPWMxB 反相。<br>11：低电平有效（AL），EPWMxA 和 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | EPWMxB 均反相。<br>复位类型：SYSRSn |
| 1-0 | OUT_MODE | R/W | 0h | 死区输出模式控制。位 1 控制 S1 开关，位 0 控制 S0 开关。<br>00：完全禁用或旁路 DBM，POLSEL 和 IN-MODE 位无效。<br>01：Apath = InA（A 路径旁路延迟）；Bpath = FED（B 路径下降沿延迟）。<br>10：Apath = RED（A 路径上升沿延迟）；Bpath = InB（B 路径旁路延迟）。<br>11：完全使能 DBM，即 RED 和 FED 均有效。<br>复位类型：SYSRSn |

## 15.16.2.11 DBCTL2 寄存器

偏移地址：1Ah　复位值：0000h

死区发生器控制寄存器 2

表 15-45 DBCTL2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | SHDWDBCTLMODE | LOADDBCTLMODE | |
| R-0-0h | | | | | R/W-0h | R/W-0h | |

表 15-46 DBCTL2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-3 | RESERVED | R-0 | 0h | 保留 |
| 2 | SHDWDBCTLMODE | R/W | 0h | DBCTL 装载模式。<br>0：立即模式，仅使用活动 DBCTL 寄存器，CPU 读写直接访问活动寄存器。<br>1：影子模式，对 DBCTL 位 [5:0] 的所有读写均访问影子部分；其他位仍访问活动寄存器。<br>复位类型：SYSRSn |
| 1-0 | LOADDBCTLMODE | R/W | 0h | 活动 DBCTL 从影子寄存器装载的选择模式。<br>00：计数器 = 0（CNT_eq）时装载。<br>01：计数器 = 周期（PRD_eq）时装载。<br>10：计数器 = 0 或周期时装载。<br>11：冻结（禁止装载）。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 注意：立即模式下无效。<br>复位类型：SYSRSn |

## 15.16.2.12 AQCTL 寄存器

偏移地址：20h　复位值：0000h

动作限定器控制寄存器

表 15-47 AQCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | LDAQBSYNC | | LDAQASYNC | |
| R-0-0h | | | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | SHDWAQBMODE | 保留 | SHDWAQAMOE | LDAQBMODE | | LDAQAMODE | |
| R-0-0h | R/W-0h | R-0-0h | R/W-0h | R/W-0h | | R/W-0h | |

表 15-48 AQCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R-0 | 0h | 保留 |
| 11-10 | LDAQBSYNC | R/W | 0h | SYNC 事件触发 AQCTLB 从影子到活动寄存器的装载。<br>00：按 LDAQBMODE 装载。<br>01：按 LDAQBMODE 或发生 SYNC 时均装载。<br>10：仅在收到 SYNC 时装载。<br>11：保留。<br>注意：仅在 AQCTL[SHDWAQBMODE] = 1 时有效。<br>复位类型：SYSRSn |
| 9-8 | LDAQASYNC | R/W | 0h | SYNC 事件触发 AQCTLA 从影子到活动寄存器的装载。<br>00：按 LDAQAMODE 装载。<br>01：按 LDAQAMODE 或发生 SYNC 时均装载。<br>10：仅在收到 SYNC 时装载。<br>11：保留。<br>注意：仅在 AQCTL[SHDWAQAMODE] = 1 时有效。<br>复位类型：SYSRSn |
| 7 | RESERVED | R-0 | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 6 | SHDWAQBMODE | R/W | 0h | 动作限定器 B 寄存器工作模式。<br>1：影子模式，采用双缓冲，CPU 写操作访问影子寄存器。<br>0：立即模式，仅使用活动动作限定寄存器，CPU 读写直接访问活动寄存器。<br>复位类型：SYSRSn |
| 5 | RESERVED | R-0 | 0h | 保留 |
| 4 | SHDWAQAMODE | R/W | 0h | 动作限定器 A 寄存器工作模式。<br>1：影子模式，采用双缓冲，CPU 写操作访问影子寄存器。<br>0：立即模式，仅使用活动动作限定寄存器，CPU 读写直接访问活动寄存器。<br>复位类型：SYSRSn |
| 3-2 | LDAQBMODE | R/W | 0h | 活动动作限定器 B 从影子寄存器装载的选择模式。<br>00：在 CTR = Zero（TBCTR = 0x0000）时装载。<br>01：在 CTR = PRD（TBCTR = TBPRD）时装载。<br>10：在 CTR = Zero 或 CTR = PRD 时装载。<br>11：冻结（禁止装载）。<br>注意：立即模式下无效。<br>复位类型：SYSRSn |
| 1-0 | LDAQAMODE | R/W | 0h | 活动动作限定器 A 从影子寄存器装载的选择模式。<br>00：在 CTR = Zero（TBCTR = 0x0000）时装载。<br>01：在 CTR = PRD（TBCTR = TBPRD）时装载。<br>10：在 CTR = Zero 或 CTR = PRD 时装载。<br>11：冻结（禁止装载）。<br>注意：立即模式下无效。<br>复位类型：SYSRSn |

### 15.16.2.13 AQTSRCSEL 寄存器

偏移地址：22h　复位值：0000h

动作限定器触发事件源选择寄存器

表 15-49 AQTSRCSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| T2SEL | | | | T1SEL | | | |
| R/W-0h | | | | R/W-0h | | | |

表 15-50 AQTSRCSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |
| 7-4 | T2SEL | R/W | 0h | T2 事件源选择位。<br>0000：DCAEVT1<br>0001：DCAEVT2<br>0010：DCBEVT1<br>0011：DCBEVT2<br>0100：TZ1<br>0101：TZ2<br>0110：TZ3<br>0111：EPWMxSYNCI<br>1000：DCEVTFILT<br>其他值：保留。<br>复位类型：SYSRSn |
| 3-0 | T1SEL | R/W | 0h | T1 事件源选择位。<br>0000：DCAEVT1<br>0001：DCAEVT2<br>0010：DCBEVT1<br>0011：DCBEVT2<br>0100：TZ1<br>0101：TZ2<br>0110：TZ3<br>0111：EPWMxSYNCI<br>1000：DCEVTFILT<br>其他值：保留。<br>复位类型：SYSRSn |

## 15.16.2.14 PCCTL 寄存器

偏移地址：28h　复位值：0000h

PWM 斩波器控制寄存器

表 15-51 PCCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | CHPDUTY | | |
| R-0-0h | | | | | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CHPFREQ | | | OSHTWTH | | | | CHPEN |
| R/W-0h | | | R/W-0h | | | | R/W-0h |

表 15-52 PCCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-11 | RESERVED | R-0 | 0h | 保留 |
| 10-8 | CHPDUTY | R/W | 0h | 斩波时钟占空比。<br>000：占空比 = 1/8（12.5%）<br>001：占空比 = 2/8（25.0%）<br>010：占空比 = 3/8（37.5%）<br>011：占空比 = 4/8（50.0%）<br>100：占空比 = 5/8（62.5%）<br>101：占空比 = 6/8（75.0%）<br>110：占空比 = 7/8（87.5%）<br>111：保留。<br>复位类型：SYSRSn |
| 7-5 | CHPFREQ | R/W | 0h | 斩波时钟频率。<br>000：1 分频（不预分频；100 MHz TBCLK 时为 12.5 MHz）。<br>001：2 分频（100 MHz TBCLK 时为 6.25 MHz）。<br>010：3 分频（100 MHz TBCLK 时为 4.16 MHz）。<br>011：4 分频（100 MHz TBCLK 时为 3.12 MHz）。<br>100：5 分频（100 MHz TBCLK 时为 2.50 MHz）。<br>101：6 分频（100 MHz TBCLK 时为 2.08 MHz）。<br>110：7 分频（100 MHz TBCLK 时为 1.78 MHz）。<br>111：8 分频（100 MHz TBCLK 时为 1.56 MHz）。<br>复位类型：SYSRSn |
| 4-1 | OSHTWTH | R/W | 0h | 单次脉冲宽度。<br>0000：宽度为 1 x EPWMCLK / 8（100 MHz EPWMCLK 时为 80 ns）。<br>0001：宽度为 2 x EPWMCLK / 8（100 MHz EPWMCLK 时为 160 ns）。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0010：宽度为 3 x EPWMCLK / 8（100 MHz EPWMCLK 时为 240 ns）。<br>0011：宽度为 4 x EPWMCLK / 8（100 MHz EPWMCLK 时为 320 ns）。<br>0100：宽度为 5 x EPWMCLK / 8（100 MHz EPWMCLK 时为 400 ns）。<br>0101：宽度为 6 x EPWMCLK / 8（100 MHz EPWMCLK 时为 480 ns）。<br>0110：宽度为 7 x EPWMCLK / 8（100 MHz EPWMCLK 时为 560 ns）。<br>0111：宽度为 8 x EPWMCLK / 8（100 MHz EPWMCLK 时为 640 ns）。<br>1000：宽度为 9 x EPWMCLK / 8（100 MHz EPWMCLK 时为 720 ns）。<br>1001：宽度为 10 x EPWMCLK / 8（100 MHz EPWMCLK 时为 800 ns）。<br>1010：宽度为 11 x EPWMCLK / 8（100 MHz EPWMCLK 时为 880 ns）。<br>1011：宽度为 12 x EPWMCLK / 8（100 MHz EPWMCLK 时为 960 ns）。<br>1100：宽度为 13 x EPWMCLK / 8（100 MHz EPWMCLK 时为 1040 ns）。<br>1101：宽度为 14 x EPWMCLK / 8（100 MHz EPWMCLK 时为 1120 ns）。<br>1110：宽度为 15 x EPWMCLK / 8（100 MHz EPWMCLK 时为 1200 ns）。<br>1111：宽度为 16 x EPWMCLK / 8（100 MHz EPWMCLK 时为 1280 ns）。<br>复位类型：SYSRSn |
| 0 | CHPEN | R/W | 0h | PWM 斩波使能。<br>0：禁用（旁路）PWM 斩波功能。<br>1：使能斩波功能。<br>复位类型：SYSRSn |

## 15.16.2.15 VCAPCTL 寄存器

偏移地址：30h　复位值：0000h

谷底捕获控制寄存器

表 15-53 VCAPCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | EDGEFILTDLYSEL | VDELAYDIV | |
| R-0-0h | | | | | R/W-0h | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VDELAYDIV | 保留 | | TRIGSEL | | | VCAPSTART | VCAPE |
| R/W-0h | R-0-0h | | R/W-0h | | | R-0/W1S-0h | R/W-0h |

表 15-54 VCAPCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-11 | RESERVED | R-0 | 0h | 保留 |
| 10 | EDGEFILTDLYSEL | R/W | 0h | 谷底开关模式延迟选择。<br>0：不对边沿滤波器输出施加延迟。<br>1：对边沿滤波器输出施加 HWDELAYVAL 延迟。<br>复位类型：SYSRSn |
| 9-7 | VDELAYDIV | R/W | 0h | 谷底延迟模式分频使能。<br>000：HWVDELVAL = SWVDELVAL<br>001：HWVDELVAL = VCNTVAL+SWVDELVAL<br>010：HWVDELVAL = VCNTVAL>>1+SWVDELVAL<br>011：HWVDELVAL = VCNTVAL>>2+SWVDELVAL<br>100：HWVDELVAL = VCNTVAL>>4+SWVDELVAL<br>注意：可使用这些位对连续边沿捕获之间的延迟值进行可选分频。<br>复位类型：SYSRSn |
| 6-5 | RESERVED | R-0 | 0h | 保留 |
| 4-2 | TRIGSEL | R/W | 0h | 已捕获事件数量状态。<br>000：通过软件写 VCAPCTL[VCAPSTART] 触发捕获序列。<br>001：CNT_zero 事件触发。<br>010：PRD_eq 事件触发。<br>011：CNT_zero 或 PRD_eq 事件触发。<br>100：DCAEVT1 事件触发。<br>101：DCAEVT2 事件触发。<br>110：DCBEVT1 事件触发。<br>111：DCBEVT2 事件触发。<br>注意：此字段所选事件触发谷底捕获序列。所选事件发生后，捕获序列进入就绪状态；根据 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | DCFCTL[SRCSEL] 所选事件执行捕获。<br>DCFCTL[SRCSEL] 和 VCAPCTL[TRIGSEL] 不得选择同一事件。无论当前捕获状态如何，只要 VCAPCTL[TRIGSEL] 所选事件发生，就会重新触发捕获序列。<br>复位类型：SYSRSn |
| 1 | VCAPSTART | R-0/W1S | 0h | 谷底捕获启动。<br>0：写 0 无效。<br>1：若 VCAPCTL[TRIGSEL]=0x0，则触发一次捕获序列。<br>注意：此位用于通过软件启动谷底捕获序列；只有 VCAPCTL[TRIGSEL] 选择软件触发时才有效。写 1 将产生一次捕获序列触发。<br>复位类型：SYSRSn |
| 0 | VCAPE | R/W | 0h | 谷底捕获使能 / 禁用。<br>0：禁用。<br>1：使能。<br>复位类型：SYSRSn |

## 15.16.2.16 VCNTCFG 寄存器

偏移地址：32h　复位值：0000h

谷底计数器配置寄存器

表 15-55 VCNTCFG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| STOPEDGESTS | 保留 | | | STOPEDGE | | | |
| R-0h | R-0-0h | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STARTEDGESTS | 保留 | | | STARTEDGE | | | |
| R-0h | R-0-0h | | | R/W-0h | | | |

表 15-56 VCNTCFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | STOPEDGESTS | R | 0h | 停止边沿状态位。<br>0：停止边沿尚未发生。<br>1：停止边沿已发生。<br>注意：只有在 VCAPCTL[TRIGSEL] 所选触发脉冲发生、触发序列进入就绪状态后，且发生 STOPEDGE 时，才会置位。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | VCAPCTL[TRIGSEL] 所选触发脉冲发生时复位此位。<br>复位类型：SYSRSn |
| 14-12 | RESERVED | R-0 | 0h | 保留 |
| 11-8 | STOPEDGE | R/W | 0h | 计数器停止边沿选择。VCAPCTL[TRIGSEL] 所选触发脉冲使计数操作进入就绪状态后，谷底计数器在此字段所选数量的事件发生时停止计数。<br>0000：不停止。<br>0001：第 1 个边沿。<br>0010：第 2 个边沿。<br>0011：第 3 个边沿。<br>…<br>1,1,1,1：第 15 个边沿。<br>复位类型：SYSRSn |
| 7 | STARTEDGESTS | R | 0h | 启动边沿状态位。<br>0：启动边沿尚未发生。<br>1：启动边沿已发生。<br>注意：只有在 VCAPCTL[TRIGSEL] 所选触发脉冲发生、触发序列进入就绪状态后，且发生 STARTEDGE 时，才会置位。<br>VCAPCTL[TRIGSEL] 所选触发脉冲发生时复位此位。<br>复位类型：SYSRSn |
| 6-4 | RESERVED | R-0 | 0h | 保留 |
| 3-0 | STARTEDGE | R/W | 0h | 计数器启动边沿选择。VCAPCTL[TRIGSEL] 所选触发脉冲使计数操作进入就绪状态后，谷底计数器在此字段所选数量的事件发生时启动计数。<br>0000：不启动。<br>0001：第 1 个边沿。<br>0010：第 2 个边沿。<br>0011：第 3 个边沿。<br>…<br>1111：第 15 个边沿。<br>复位类型：SYSRSn |

### 15.16.2.17 HRCNFG 寄存器

偏移地址：40h　复位值：0000h

HRPWM 配置寄存器

此寄存器仅可在具备 HRPWM 功能的 EPWM 模块上访问。

表 15-57 HRCNFG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | 保留 | HRLOADB | | CTLMODEB | EDGMODEB | |
| R/W-0h | | R-0-0h | R/W-0h | | R/W-0h | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SWAPAB | AUTOCONV | SELOUTB | HRLOAD | | CTLMODE | EDGMODE | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | R/W-0h | R/W-0h | |

表 15-58 HRCNFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | RESERVED | R/W | 0h | 保留 |
| 13 | RESERVED | R-0 | 0h | 保留 |
| 12-11 | HRLOADB | R/W | 0h | 影子模式位。选择将 CMPBHR 影子值装载到活动寄存器的时基事件。<br>00：在 CTR = Zero（TBCTR = 0x0000）时装载。<br>01：在 CTR = PRD（TBCTR = TBPRD）时装载。<br>10：在 CTR = Zero 或 CTR = PRD 时装载。<br>11：保留。<br>复位类型：SYSRSn |
| 10 | CTLMODEB | R/W | 0h | 控制模式位。选择控制 MEP 的寄存器（CMP/TBPRD 或 TBPHS）。<br>0：CMPBHR(8) 或 TBPRDHR(8) 控制边沿位置，即占空比或周期控制模式（复位默认值）。<br>1：TBPHSHR(8) 控制边沿位置，即相位控制模式。<br>复位类型：SYSRSn |
| 9-8 | EDGMODEB | R/W | 0h | 边沿模式位。选择由微边沿定位（MEP）逻辑控制的 PWM 边沿。<br>00：禁用 HRPWM（复位默认值）。<br>01：MEP 控制上升沿（CMPBHR）。<br>10：MEP 控制下降沿（CMPBHR）。<br>11：MEP 控制双边沿（TBPHSHR 或 TBPRDHR）。<br>复位类型：SYSRSn |
| 7 | SWAPAB | R/W | 0h | 交换 ePWM A、B 输出信号。<br>0：ePWMxA 和 ePWMxB 输出不变。<br>1：ePWMxA 信号出现在 ePWMxB 输出端，ePWMxB 信号出现在 ePWMxA 输出端。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 6 | AUTOCONV | R/W | 0h | 自动转换延迟线值。选择 CMPAHR/TBPRDHR/TBPHSHR 中的占空比 / 周期 / 相位小数部分是由 HRMSTEP 中的 MEP 比例因子自动缩放，还是由应用软件计算并手动缩放。SFO 库函数自动以适当的 MEP 比例因子更新 HRMSTEP。<br>0：禁用自动 HRMSTEP 缩放。<br>1：使能自动 HRMSTEP 缩放。<br>若应用软件手动缩放占空比或相位的小数部分（例如占空比计算中，软件设置 CMPAHR = (fraction(PWMduty * PWMperiod) * MEP Scale Factor)<<8 + 0x080），必须禁用此模式。<br>复位类型：SYSRSn |
| 5 | SELOUTB | R/W | 0h | EPWMxB 输出选择位。选择 ePWMxB 通道的输出信号。反相处理会考虑高分辨率模式，反相信号包含所有高分辨率调整；反相是修改 ePWMxB 信号的最后一步。<br>0：ePWMxB 正常输出。<br>1：ePWMxB 输出 ePWMxA 信号的反相版本。<br>复位类型：SYSRSn |
| 4-3 | HRLOAD | R/W | 0h | 影子模式位。选择将 CMPAHR 影子值装载到活动寄存器的时基事件。<br>00：在 CTR = Zero（TBCTR = 0x0000）时装载。<br>01：在 CTR = PRD（TBCTR = TBPRD）时装载。<br>10：在 CTR = Zero 或 CTR = PRD 时装载。<br>11：保留。<br>复位类型：SYSRSn |
| 2 | CTLMODE | R/W | 0h | 控制模式位。选择控制 MEP 的寄存器（CMP/TBPRD 或 TBPHS）。<br>0：CMPAHR(8) 或 TBPRDHR(8) 控制边沿位置，即占空比或周期控制模式（复位默认值）。<br>1：TBPHSHR(8) 控制边沿位置，即相位控制模式。<br>复位类型：SYSRSn |
| 1-0 | EDGMODE | R/W | 0h | 边沿模式位。选择由微边沿定位（MEP）逻辑控制的 PWM 边沿。<br>00：禁用 HRPWM（复位默认值）。<br>01：MEP 控制上升沿（CMPAHR）。<br>10：MEP 控制下降沿（CMPAHR）。<br>11：MEP 控制双边沿（TBPHSHR 或 TBPRDHR）。<br>复位类型：SYSRSn |

### 15.16.2.18 HRPWR 寄存器

偏移地址：42h　复位值：0000h

HRPWM 电源寄存器

此寄存器仅可在具备 HRPWM 功能的 EPWM 模块上访问。

表 15-59 HRPWR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CALPWRON | 保留 | | | | | 保留 | |
| R/W-0h | R-0-0h | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | 保留 | 保留 | 保留 | 保留 | 保留 | |
| R/W-0h | | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | |

表 15-60 HRPWR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | CALPWRON | R/W | 0h | MEP 校准电源控制位（仅 ePWM1 可用）。<br>0：禁用 HRPWM 内的 MEP 校准逻辑，降低功耗。<br>1：使能 MEP 校准逻辑。<br>复位类型：SYSRSn |
| 14-10 | RESERVED | R-0 | 0h | 保留 |
| 9-6 | RESERVED | R/W | 0h | 保留 |
| 5 | RESERVED | R/W | 0h | 保留 |
| 4 | RESERVED | R | 0h | 保留 |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | RESERVED | R/W | 0h | 保留 |
| 1-0 | RESERVED | R/W | 0h | 保留 |

### 15.16.2.19 HRMSTEP 寄存器

偏移地址：4Ch　复位值：0000h

HRPWM MEP 步长寄存器

此寄存器仅可在具备 HRPWM 功能的 EPWM 模块上访问。只允许 16 位访问；调试器以 32 位模式访问可能显示错误值。

表 15-61 HRMSTEP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HRMSTEP | | | | | | | |
| R/W-0h | | | | | | | |

表 15-62 HRMSTEP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |
| 7-0 | HRMSTEP | R/W | 0h | 高分辨率 MEP 步长。使能自动转换（HRCNFG[AUTOCONV] = 1）时，此 8 位字段存储 MEP_ScaleFactor（每个粗调步长中的 MEP 步数）。硬件用其将 CMPAHR、CMPBHR、DBFEDHR、DBREDHR、TBPHSHR 或 TBPRDHR 中的值自动转换为高分辨率 ePWM 输出上经缩放的微边沿延迟。每次校准结束时，SFO 校准软件写入此寄存器。<br>复位类型：SYSRSn |

## 15.16.2.20 HRCNFG2 寄存器

偏移地址：4Eh　复位值：0000h

HRPWM 配置 2 寄存器

此寄存器仅可在具备 HRPWM 功能的 EPWM 模块上访问。只允许 16 位访问；调试器以 32 位模式访问可能显示错误值。

表 15-63 HRCNFG2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | 保留 | 保留 | | | | | |
| R/W-0h | R-0/W1S-0h | R-0-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | CTLMODEDBFED | | CTLMODEDBRED | | EDGMODEDB | |
| R-0-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 15-64 HRCNFG2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R/W | 0h | 保留 |
| 14 | RESERVED | R-0/W1S | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 13-6 | RESERVED | R-0 | 0h | 保留 |
| 5-4 | CTLMODEDBFED | R/W | 0h | 影子模式位，选择应与 DBCTL[LOADFEDMODE] 一致。选择将 DBFEDHR 影子值装载到活动寄存器的时基事件。<br>00：在 CTR = Zero（TBCTR = 0x0000）时装载。<br>01：在 CTR = PRD（TBCTR = TBPRD）时装载。<br>10：在 CTR = Zero 或 CTR = PRD 时装载。<br>11：保留。<br>复位类型：SYSRSn |
| 3-2 | CTLMODEDBRED | R/W | 0h | 影子模式位，选择应与 DBCTL[LOADREDMODE] 一致。选择将 DBREDHR 影子值装载到活动寄存器的时基事件。<br>00：在 CTR = Zero（TBCTR = 0x0000）时装载。<br>01：在 CTR = PRD（TBCTR = TBPRD）时装载。<br>10：在 CTR = Zero 或 CTR = PRD 时装载。<br>11：保留。<br>复位类型：SYSRSn |
| 1-0 | EDGMODEDB | R/W | 0h | 边沿模式位。选择由微边沿定位（MEP）逻辑控制的 PWM 边沿。<br>00：禁用 HRPWM（复位默认值）。<br>01：MEP 控制上升沿（DBREDHR）。<br>10：MEP 控制下降沿（DBFEDHR）。<br>11：MEP 控制双边沿（DBREDHR 的上升沿或 DBFEDHR 的下降沿）。<br>复位类型：SYSRSn |

## 15.16.2.21 HRPCTL 寄存器

偏移地址：5Ah 复位值：0000h

高分辨率周期控制寄存器

此寄存器仅可在具备 HRPWM 功能的 EPWM 模块上访问。

表 15-65 HRPCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | PWMSYNCSELX | | | 保留 | TBPHSHRLOADE | PWMSYNCSEL | HRPE |
| R-0-0h | R/W-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 15-66 HRPCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-7 | RESERVED | R-0 | 0h | 保留 |
| 6-4 | PWMSYNCSELX | R/W | 0h | EPWMSYNCPER 的扩展选择位。<br>000：由 PWMSYNCSEL 定义 EPWMSYNCPER，为默认配置（兼容旧版 EPWM）。<br>001：保留。<br>010：保留。<br>011：保留。<br>100：CTR = CMPC，向上计数。<br>101：CTR = CMPC，向下计数。<br>110：CTR = CMPD，向上计数。<br>111：CTR = CMPD，向下计数。<br>复位类型：SYSRSn |
| 3 | RESERVED | R/W | 0h | 保留 |
| 2 | TBPHSHRLOADE | R/W | 0h | TBPHSHR 装载使能。允许在 SYNCIN、TBCTL[SWFSYNC] 或数字比较事件发生时，以高分辨率相位同步 ePWM 模块，使多个同频 ePWM 模块实现高分辨率相位对齐。<br>0：禁用这些事件触发的高分辨率相位同步。<br>1：在 SYNCIN、TBCTL[SWFSYNC] 或数字比较器同步事件发生时，使用高分辨率相位寄存器 TBPHSHR 的内容同步相位。TBCTL[PHSEN] 独立控制在 SYNCIN 或 TBCTL[SWFSYNC] 事件时以 TBPHS 装载 TBCTR；但若需结合高分辨率周期功能控制相位，也需使能此位。<br>在上下计数模式使能高分辨率周期时，即使 TBPHSHR = 0x0000，此位和 TBCTL[PHSEN] 也都必须置 1。仅使能高分辨率占空比时，无需设置此位。<br>复位类型：SYSRSn |
| 1 | PWMSYNCSEL | R/W | 0h | PWMSYNC 源选择位，选择送往 CMPSS 和 GPDAC 的 EPWMSYNCPER 信号源。<br>0：CTR = PRD，时基计数器等于周期（TBCTR |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | = TBPRD）。<br>1：CTR = zero，时基计数器等于零（TBCTR = 0x00）。<br>复位类型：SYSRSn |
| 0 | HRPE | R/W | 0h | 高分辨率周期使能。<br>0：禁用高分辨率周期功能，ePWM 按 Type 4 ePWM 工作。<br>1：使能高分辨率周期，HRPWM 可同时控制占空比和频率的高分辨率。使能后不支持 TBCTL[CTRMODE] = 0,1（向下计数模式）。<br>复位类型：SYSRSn |

### 15.16.2.22 TRREM 寄存器

偏移地址：5Ch　复位值：0000h

HRPWM 高分辨率余数寄存器

此寄存器仅可在具备 HRPWM 功能的 EPWM 模块上访问。

表 15-67 TRREM 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | TRREM | | |
| R-0-0h | | | | | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TRREM | | | | | | | |
| R/W-0h | | | | | | | |

表 15-68 TRREM 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-11 | RESERVED | R-0 | 0h | 保留 |
| 10-0 | TRREM | R/W | 0h | HRPWM 余数位。此 11 位值保存 HRPWM 算法及硬件计算的余数部分。<br>注意：<br>1. 在 SYNCIN、TBCTL[SWFSYNC] 或 DC 事件（若使能）发生时，可用 TBPHSHR 自动初始化 TRREM 的低 8 位；也可由 CPU 写入。<br>2. 更新优先级：同步事件（软件或硬件）将 TBPHSHR 复制到 TRREM，优先级最高；HRPWM 硬件更新次之；CPU 写入最低。<br>3. 非对称模式不使用位 10，可将其强制置零。向上计数模式和上下计数模式分别将 TRREM 初始化为 0x0 和 0x100。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 非对称模式：<br>TRREM[7:0] = TBPHSHR[15:8]<br>TRREM[10,9,8] = 0,0,0<br>对称模式：<br>TRREM[7:0] = TBPHSHR[15:8]<br>TRREM[10,9,8] = 0,0,1<br>复位类型：SYSRSn |

## 15.16.2.23 GLDCTL 寄存器

偏移地址：68h　复位值：0000h

全局 PWM 装载控制寄存器

表 15-69 GLDCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | GLDCNT | | | GLDPRD | |
| R-0-0h | | | R-0h | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| GLDPRD | 保留 | OSHTMODE | GLDMODE | | | | GLD |
| R/W-0h | R-0-0h | R/W-0h | R/W-0h | | | | R/W-0h |

表 15-70 GLDCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-13 | RESERVED | R-0 | 0h | 保留 |
| 12-10 | GLDCNT | R | 0h | 全局装载选通计数器。表示已发生的所选事件数。<br>000：无事件。<br>001：1 个事件。<br>010：2 个事件。<br>011：3 个事件。<br>100：4 个事件。<br>101：5 个事件。<br>110：6 个事件。<br>111：7 个事件。<br>复位类型：SYSRSn |
| 9-7 | GLDPRD | R/W | 0h | 全局装载选通周期选择。选择所选事件发生多少次后产生装载选通信号。<br>000：禁用计数器。<br>001：GLDCNT = 001 时产生选通信号（第 1 个事件）。<br>010：GLDCNT = 010 时产生选通信号（第 2 个事件）。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 011：GLDCNT = 011 时产生选通信号（第 3 个事件）。<br>100：GLDCNT = 011 时产生选通信号（第 4 个事件）。<br>101：GLDCNT = 001 时产生选通信号（第 5 个事件）。<br>110：GLDCNT = 010 时产生选通信号（第 6 个事件）。<br>111：GLDCNT = 011 时产生选通信号（第 7 个事件）。<br>复位类型：SYSRSn |
| 6 | RESERVED | R-0 | 0h | 保留 |
| 5 | OSHTMODE | R/W | 0h | 单次装载模式控制。<br>0：禁用单次装载模式，每次所选装载选通信号均执行影子到活动装载。<br>1：使能单次模式，阻止所有装载选通信号，直到向 GLDCTL2[OSHTLD] 写 1。<br>注意：单次模式仅可在使能全局影子到活动装载模式（GLDCTL[GLD]=1）时使用。<br>复位类型：SYSRSn |
| 4-1 | GLDMODE | R/W | 0h | 选择用于影子到活动重新装载的全局装载脉冲。<br>0000：计数器 = 0（CNT_ZRO）时装载。<br>0001：计数器 = 周期（PRD_EQ）时装载。<br>0010：计数器 = 0 或周期时装载。<br>0011：SYNCEVT 时装载；SYNCEVT 为 DCAEVT1.sync、DCBEVT1.sync、EPWMxSYNCI 和 TBCTL[SWFSYNC] 的逻辑或。<br>0100：SYNCEVT 或 CNT_ZRO 时装载。<br>0101：SYNCEVT 或 PRD_EQ 时装载。<br>0110：SYNCEVT、CNT_ZRO 或 PRD_EQ 时装载。<br>1000：保留。<br>…<br>1110：保留。<br>1111：写 GLDCTL2[GFRCLD] 时装载。<br>复位类型：SYSRSn |
| 0 | GLD | R/W | 0h | 全局影子到活动装载事件控制。<br>0：各影子寄存器按各自重新装载控制位执行影子到活动装载（兼容旧版 EPWM）。<br>1：所有影子到活动装载事件由 GLDCTL 的 GLDMODE 位定义；所有影子寄存器使用同一装载脉冲，忽略各自的 LOADMODE 位。<br>复位类型：SYSRSn |

### 15.16.2.24 GLDCFG 寄存器

偏移地址：6Ah　复位值：0000h

全局 PWM 装载配置寄存器

表 15-71 GLDCFG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | AQCSFRC | AQCTLB_AQCTLB2 | AQCTLA_AQCTLA2 |
| R-0-0h | | | | | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBCTL | DBFED_DBFEDHR | DBRED_DBREDHR | CMPD | CMPC | CMPB_CMPBHR | CMPA_CMPAHR | TBPRD_TBPRDHR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 15-72 GLDCFG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-11 | RESERVED | R-0 | 0h | 保留 |
| 10 | AQCSFRC | R/W | 0h | AQCSFRC 全局装载事件配置。<br>0：即使 GLDCTL(GLD)=1，寄存器仍使用本地重新装载配置（兼容旧版 EPWM）。<br>1：此位置位且 GLDCTL(GLD)=1 时，寄存器使用全局装载配置。<br>复位类型：SYSRSn |
| 9 | AQCTLB_AQCTLB2 | R/W | 0h | AQCTLB_AQCTLB2 全局装载事件配置。<br>0：即使 GLDCTL(GLD)=1，寄存器仍使用本地重新装载配置（兼容旧版 EPWM）。<br>1：此位置位且 GLDCTL(GLD)=1 时，寄存器使用全局装载配置。<br>复位类型：SYSRSn |
| 8 | AQCTLA_AQCTLA2 | R/W | 0h | AQCTLA_AQCTLA2 全局装载事件配置。<br>0：即使 GLDCTL(GLD)=1，寄存器仍使用本地重新装载配置（兼容旧版 EPWM）。<br>1：此位置位且 GLDCTL(GLD)=1 时，寄存器使用全局装载配置。<br>复位类型：SYSRSn |
| 7 | DBCTL | R/W | 0h | DBCTL 全局装载事件配置。<br>0：即使 GLDCTL(GLD)=1，寄存器仍使用本地重新装载配置（兼容旧版 EPWM）。<br>1：此位置位且 GLDCTL(GLD)=1 时，寄存器使用全局装载配置。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 6 | DBFED_DBFEDHR | R/W | 0h | DBFED_DBFEDHR 全局装载事件配置。<br>0：即使 GLDCTL(GLD)=1，寄存器仍使用本地重新装载配置（兼容旧版 EPWM）。<br>1：此位置位且 GLDCTL(GLD)=1 时，寄存器使用全局装载配置。<br>复位类型：SYSRSn |
| 5 | DBRED_DBREDHR | R/W | 0h | DBRED_DBREDHR 全局装载事件配置。<br>0：即使 GLDCTL(GLD)=1，寄存器仍使用本地重新装载配置（兼容旧版 EPWM）。<br>1：此位置位且 GLDCTL(GLD)=1 时，寄存器使用全局装载配置。<br>复位类型：SYSRSn |
| 4 | CMPD | R/W | 0h | CMPD 全局装载事件配置。<br>0：即使 GLDCTL(GLD)=1，寄存器仍使用本地重新装载配置（兼容旧版 EPWM）。<br>1：此位置位且 GLDCTL(GLD)=1 时，寄存器使用全局装载配置。<br>复位类型：SYSRSn |
| 3 | CMPC | R/W | 0h | CMPC 全局装载事件配置。<br>0：即使 GLDCTL(GLD)=1，寄存器仍使用本地重新装载配置（兼容旧版 EPWM）。<br>1：此位置位且 GLDCTL(GLD)=1 时，寄存器使用全局装载配置。<br>复位类型：SYSRSn |
| 2 | CMPB_CMPBHR | R/W | 0h | CMPB_CMPBHR 全局装载事件配置。<br>0：即使 GLDCTL(GLD)=1，寄存器仍使用本地重新装载配置（兼容旧版 EPWM）。<br>1：此位置位且 GLDCTL(GLD)=1 时，寄存器使用全局装载配置。<br>复位类型：SYSRSn |
| 1 | CMPA_CMPAHR | R/W | 0h | CMPA_CMPAHR 全局装载事件配置。<br>0：即使 GLDCTL(GLD)=1，寄存器仍使用本地重新装载配置（兼容旧版 EPWM）。<br>1：此位置位且 GLDCTL(GLD)=1 时，寄存器使用全局装载配置。<br>复位类型：SYSRSn |
| 0 | TBPRD_TBPRDHR | R/W | 0h | TBPRD_TBPRDHR 全局装载事件配置。<br>0：即使 GLDCTL(GLD)=1，寄存器仍使用本地重新装载配置（兼容旧版 EPWM）。<br>1：此位置位且 GLDCTL(GLD)=1 时，寄存器使用全局装载配置。<br>复位类型：SYSRSn |

### 15.16.2.25 EPWMXLINK 寄存器

偏移地址：70h　复位值：000XXXXXh

EPWMx 链接寄存器

此寄存器控制哪些 EPWM 与其他 EPWM 模块链接。各模块默认复位值可能不同；复位值将各 EPWM 模块链接到自身，以防意外链接模块。

表 15-73 EPWMXLINK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| GLDCTL2LINK | | | | 保留 | | | |
| R/W-0h | | | | R-0-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | CMPDLINK | | | |
| R-0-0h | | | | R/W-X | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CMPCLINK | | | | CMPBLINK | | | |
| R/W-X | | | | R/W-X | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMPALINK | | | | TBPRDLINK | | | |
| R/W-X | | | | R/W-X | | | |

表 15-74 EPWMXLINK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-28 | GLDCTL2LINK | R/W | 0h | GLDCTL2 链接位。写入下列编码所选 ePWM 模块的 GLDCTL2 寄存器时，也将同时写入当前 ePWM 模块的 GLDCTL2 寄存器。<br>0000：ePWM1<br>0001：ePWM2<br>…<br>依此直到最后一个 ePWM 实例；其他值均保留。<br>复位类型：SYSRSn |
| 27-20 | RESERVED | R-0 | 0h | 保留 |
| 19-16 | CMPDLINK | R/W | X | CMPD 链接位。写入下列编码所选 ePWM 模块的 CMPD 寄存器时，也将同时写入当前 ePWM 模块的 CMPD 寄存器。<br>0000：ePWM1<br>0001：ePWM2<br>…<br>依此直到最后一个 ePWM 实例；其他值均保 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 留。<br>复位类型：SYSRSn |
| 15-12 | CMPCLINK | R/W | X | CMPC 链接位。写入下列编码所选 ePWM 模块的 CMPC 寄存器时，也将同时写入当前 ePWM 模块的 CMPC 寄存器。<br>0000：ePWM1<br>0001：ePWM2<br>…<br>依此直到最后一个 ePWM 实例；其他值均保留。<br>复位类型：SYSRSn |
| 11-8 | CMPBLINK | R/W | X | CMPB_CMPBHR 链接位。写入下列编码所选 ePWM 模块的 CMPB_CMPBHR 寄存器时，也将同时写入当前 ePWM 模块的 CMPB_CMPBHR 寄存器。<br>0000：ePWM1<br>0001：ePWM2<br>…<br>依此直到最后一个 ePWM 实例；其他值均保留。<br>复位类型：SYSRSn |
| 7-4 | CMPALINK | R/W | X | CMPA_CMPAHR 链接位。写入下列编码所选 ePWM 模块的 CMPA_CMPAHR 寄存器时，也将同时写入当前 ePWM 模块的 CMPA_CMPAHR 寄存器。<br>0000：ePWM1<br>0001：ePWM2<br>…<br>依此直到最后一个 ePWM 实例；其他值均保留。<br>复位类型：SYSRSn |
| 3-0 | TBPRDLINK | R/W | X | TBPRD_TBPRDHR 链接位。写入下列编码所选 ePWM 模块的 TBPRD:TBPRDHR 寄存器时，也将同时写入当前 ePWM 模块的 TBPRD_TBPRDHR 寄存器。<br>0000：ePWM1<br>0001：ePWM2<br>…<br>依此直到最后一个 ePWM 实例；其他值均保留。<br>复位类型：SYSRSn |

## 15.16.2.26 AQCTLA 寄存器

偏移地址：80h　复位值：0000h

输出 A 动作限定器控制寄存器

表 15-75 AQCTLA 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | CBD | | CBU | |
| R-0-0h | | | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CAD | | CAU | | PRD | | ZRO | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 15-76 AQCTLA 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R-0 | 0h | 保留 |
| 11-10 | CBD | R/W | 0h | 向下计数时 TBCTR = CMPB 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxA 输出低电平。<br>10：置位，强制 EPWMxA 输出高电平。<br>11：翻转 EPWMxA 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 9-8 | CBU | R/W | 0h | 向上计数时 TBCTR = CMPB 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxA 输出低电平。<br>10：置位，强制 EPWMxA 输出高电平。<br>11：翻转 EPWMxA 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 7-6 | CAD | R/W | 0h | 向下计数时 TBCTR = CMPA 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxA 输出低电平。<br>10：置位，强制 EPWMxA 输出高电平。<br>11：翻转 EPWMxA 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 5-4 | CAU | R/W | 0h | 向上计数时 TBCTR = CMPA 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxA 输出低电平。<br>10：置位，强制 EPWMxA 输出高电平。<br>11：翻转 EPWMxA 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 3-2 | PRD | R/W | 0h | TBCTR = TBPRD 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxA 输出低电平。<br>10：置位，强制 EPWMxA 输出高电平。<br>11：翻转 EPWMxA 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 1-0 | ZRO | R/W | 0h | TBCTR = 0 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxA 输出低电平。<br>10：置位，强制 EPWMxA 输出高电平。<br>11：翻转 EPWMxA 输出，低变高、高变低。<br>复位类型：SYSRSn |

## 15.16.2.27 AQCTLA2 寄存器

偏移地址：82h　复位值：0000h

输出 A 附加动作限定器控制寄存器

表 15-77 AQCTLA2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| T2D | | T2U | | T1D | | T1U | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 15-78 AQCTLA2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |
| 7-6 | T2D | R/W | 0h | 向下计数时发生 T2 事件 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 00：不执行动作（禁用）。<br>01：清除，强制 EPWMxA 输出低电平。<br>10：置位，强制 EPWMxA 输出高电平。<br>11：翻转 EPWMxA 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 5-4 | T2U | R/W | 0h | 向上计数时发生 T2 事件 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxA 输出低电平。<br>10：置位，强制 EPWMxA 输出高电平。<br>11：翻转 EPWMxA 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 3-2 | T1D | R/W | 0h | 向下计数时发生 T1 事件 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxA 输出低电平。<br>10：置位，强制 EPWMxA 输出高电平。<br>11：翻转 EPWMxA 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 1-0 | T1U | R/W | 0h | 向上计数时发生 T1 事件 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxA 输出低电平。<br>10：置位，强制 EPWMxA 输出高电平。<br>11：翻转 EPWMxA 输出，低变高、高变低。<br>复位类型：SYSRSn |

## 15.16.2.28 AQCTLB 寄存器

偏移地址：84h　复位值：0000h

输出 B 动作限定器控制寄存器

表 15-79 AQCTLB 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | CBD | | CBU | |
| R-0-0h | | | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CAD | | CAU | | PRD | | ZRO | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 15-80 AQCTLB 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R-0 | 0h | 保留 |
| 11-10 | CBD | R/W | 0h | 向下计数时 TBCTR = CMPB 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxB 输出低电平。<br>10：置位，强制 EPWMxB 输出高电平。<br>11：翻转 EPWMxB 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 9-8 | CBU | R/W | 0h | 向上计数时 TBCTR = CMPB 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxB 输出低电平。<br>10：置位，强制 EPWMxB 输出高电平。<br>11：翻转 EPWMxB 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 7-6 | CAD | R/W | 0h | 向下计数时 TBCTR = CMPA 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxB 输出低电平。<br>10：置位，强制 EPWMxB 输出高电平。<br>11：翻转 EPWMxB 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 5-4 | CAU | R/W | 0h | 向上计数时 TBCTR = CMPA 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxB 输出低电平。<br>10：置位，强制 EPWMxB 输出高电平。<br>11：翻转 EPWMxB 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 3-2 | PRD | R/W | 0h | TBCTR = TBPRD 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxB 输出低电平。<br>10：置位，强制 EPWMxB 输出高电平。<br>11：翻转 EPWMxB 输出，低变高、高变低。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1-0 | ZRO | R/W | 0h | TBCTR = 0 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxB 输出低电平。<br>10：置位，强制 EPWMxB 输出高电平。<br>11：翻转 EPWMxB 输出，低变高、高变低。<br>复位类型：SYSRSn |

## 15.16.2.29 AQCTLB2 寄存器

偏移地址：86h　复位值：0000h

输出 B 附加动作限定器控制寄存器

表 15-81 AQCTLB2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| T2D | | T2U | | T1D | | T1U | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 15-82 AQCTLB2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |
| 7-6 | T2D | R/W | 0h | 向下计数时发生 T2 事件 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxB 输出低电平。<br>10：置位，强制 EPWMxB 输出高电平。<br>11：翻转 EPWMxB 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 5-4 | T2U | R/W | 0h | 向上计数时发生 T2 事件 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxB 输出低电平。<br>10：置位，强制 EPWMxB 输出高电平。<br>11：翻转 EPWMxB 输出，低变高、高变低。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 3-2 | T1D | R/W | 0h | 向下计数时发生 T1 事件 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxB 输出低电平。<br>10：置位，强制 EPWMxB 输出高电平。<br>11：翻转 EPWMxB 输出，低变高、高变低。<br>复位类型：SYSRSn |
| 1-0 | T1U | R/W | 0h | 向上计数时发生 T1 事件 时的动作。<br>注意：按定义，在上下计数模式中，计数器等于 0 时，方向定义为 1，即向上计数。<br>00：不执行动作（禁用）。<br>01：清除，强制 EPWMxB 输出低电平。<br>10：置位，强制 EPWMxB 输出高电平。<br>11：翻转 EPWMxB 输出，低变高、高变低。<br>复位类型：SYSRSn |

### 15.16.2.30 AQSFRC 寄存器

偏移地址：8Eh　复位值：0000h

动作限定器软件强制寄存器

表 15-83 AQSFRC 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RLDCSF | | OTSFB | ACTSFB | | OTSFA | ACTSFA | |
| R/W-0h | | R-0/W1S-0h | R/W-0h | | R-0/W1S-0h | R/W-0h | |

表 15-84 AQSFRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |
| 7-6 | RLDCSF | R/W | 0h | AQCSFRC 活动寄存器从影子寄存器重新装载的选项。<br>00：时基计数器等于零时装载。<br>01：时基计数器等于周期时装载。<br>10：时基计数器等于零或周期时装载。<br>11：立即装载（CPU 直接访问活动寄存器，不从影子寄存器装载）。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 5 | OTSFB | R-0/W1S | 0h | 输出 B 的单次软件强制事件。<br>0：写 0 无效，读取始终返回 0。完成寄存器写入（即启动强制事件）后，此位自动清除。这是一次性的强制事件，可被输出 B 的后续事件覆盖。<br>1：启动一次软件强制事件。<br>复位类型：SYSRSn |
| 4-3 | ACTSFB | R/W | 0h | 调用单次软件强制 B 时的动作。<br>00：不执行动作（禁用）。<br>01：清除（低电平）。<br>10：置位（高电平）。<br>11：翻转（低变高，高变低）。<br>注意：此动作不受计数方向 CNT_dir 限定。<br>复位类型：SYSRSn |
| 2 | OTSFA | R-0/W1S | 0h | 输出 A 的单次软件强制事件。<br>0：写 0 无效，读取始终返回 0。完成寄存器写入（即启动强制事件）后，此位自动清除。这是一次性的强制事件，可被输出 A 的后续事件覆盖。<br>1：启动一次软件强制事件。<br>复位类型：SYSRSn |
| 1-0 | ACTSFA | R/W | 0h | 调用单次软件强制 A 时的动作。<br>00：不执行动作（禁用）。<br>01：清除（低电平）。<br>10：置位（高电平）。<br>11：翻转（低变高，高变低）。<br>注意：此动作不受计数方向 CNT_dir 限定。<br>复位类型：SYSRSn |

## 15.16.2.31 AQCSFRC 寄存器

偏移地址：92h　复位值：0000h

动作限定器连续软件强制寄存器

表 15-85 AQCSFRC 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | CSFB | | CSFA | |
| R-0-0h | | | | R/W-0h | | R/W-0h | |

表 15-86 AQCSFRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R-0 | 0h | 保留 |
| 3-2 | CSFB | R/W | 0h | 输出 B 的连续软件强制。立即模式下，在下一个 TBCLK 边沿生效；影子模式下，在影子值装载至活动寄存器后的下一个 TBCLK 边沿生效。使用 AQSFRC[RLDCSF] 配置影子模式。<br>00：禁用软件强制，无效。<br>01：强制输出 B 持续为低电平。<br>10：强制输出 B 持续为高电平。<br>11：禁用软件强制，无效。<br>复位类型：SYSRSn |
| 1-0 | CSFA | R/W | 0h | 输出 A 的连续软件强制。立即模式下，在下一个 TBCLK 边沿生效；影子模式下，在影子值装载至活动寄存器后的下一个 TBCLK 边沿生效。<br>00：禁用软件强制，无效。<br>01：强制输出 A 持续为低电平。<br>10：强制输出 A 持续为高电平。<br>11：禁用软件强制，无效。<br>复位类型：SYSRSn |

## 15.16.2.32 DBREDHR 寄存器

偏移地址：A0h　复位值：0000h

死区发生器上升沿延迟高分辨率镜像寄存器

表 15-87 DBREDHR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DBREDHR | | | | | | | 保留 |
| R/W-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | 保留 |
| R-0h | | | | | | | R-0h |

表 15-88 DBREDHR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-9 | DBREDHR | R/W | 0h | 死区上升沿延迟高分辨率位。<br>复位类型：SYSRSn |
| 8 | RESERVED | R | 0h | 保留 |
| 7-1 | RESERVED | R | 0h | 保留 |
| 0 | RESERVED | R | 0h | 保留 |

### 15.16.2.33 DBRED 寄存器

偏移地址：A2h　复位值：0000h

死区发生器上升沿延迟高分辨率镜像寄存器

表 15-89 DBRED 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | DBRED | | | | | |
| R-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBRED | | | | | | | |
| R/W-0h | | | | | | | |

表 15-90 DBRED 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | RESERVED | R | 0h | 保留 |
| 13-0 | DBRED | R/W | 0h | 上升沿延迟值。<br>复位类型：SYSRSn |

### 15.16.2.34 DBFEDHR 寄存器

偏移地址：A4h　复位值：0000h

死区发生器下降沿延迟高分辨率寄存器

表 15-91 DBFEDHR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DBFEDHR | | | | | | | 保留 |
| R/W-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | 保留 |
| R-0h | | | | | | | R-0h |

表 15-92 DBFEDHR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-9 | DBFEDHR | R/W | 0h | 死区下降沿延迟高分辨率位。<br>复位类型：SYSRSn |
| 8 | RESERVED | R | 0h | 保留 |
| 7-1 | RESERVED | R | 0h | 保留 |
| 0 | RESERVED | R | 0h | 保留 |

### 15.16.2.35 DBFED 寄存器

偏移地址：A6h　复位值：0000h

死区发生器下降沿延迟计数寄存器

表 15-93 DBFED 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | DBFED | | | | | |
| R-0h | | R/W-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DBFED | | | | | | | |
| R/W-0h | | | | | | | |

表 15-94 DBFED 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | RESERVED | R | 0h | 保留 |
| 13-0 | DBFED | R/W | 0h | 下降沿延迟计数，14 位计数器。<br>复位类型：SYSRSn |

### 15.16.2.36 TBPHS 寄存器

偏移地址：C0h　复位值：00000000h

时基相位高位

表 15-95 TBPHS 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TBPHS | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TBPHS | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TBPHSHR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TBPHSHR | | | | | | | |
| R/W-0h | | | | | | | |

表 15-96 TBPHS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | TBPHS | R/W | 0h | 相位偏移寄存器。设置所选 ePWM 时基计数器相对于提供同步输入信号的时基的相位。<br>若 TBCTL[PHSEN] = 0，则忽略同步事件，不向时基计数器装载相位值。<br>若 TBCTL[PHSEN] = 1，则同步事件发生时，将 TBPHS 相位值装载至 TBCTR。同步事件可由 EPWMxSYNCI 输入同步信号或软件强制同步发起。<br>复位类型：SYSRSn |
| 15-0 | TBPHSHR | R/W | 0h | 高分辨率相位偏移寄存器。不得使用 TBPHSHR，必须改用 TRREM（HRPWM 余数寄存器）模拟其功能。此寄存器低 8 位被忽略：写入无效，读取返回零。<br>复位类型：SYSRSn |

## 15.16.2.37 TBPRDHR 寄存器

偏移地址：C4h　复位值：0000h

时基周期高分辨率寄存器

表 15-97 TBPRDHR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| TBPRDHR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TBPRDHR | | | | | | | |
| R/W-0h | | | | | | | |

表 15-98 TBPRDHR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | TBPRDHR | R/W | 0h | 周期高分辨率位。高 8 位包含周期值的高分辨率部分。TBPRDHR 不受 TBCTL[PRDLD] 影响，读写始终访问影子寄存器。仅在使能高分辨率周期功能时使用此寄存器，且仅支持高分辨率周期控制的 ePWM 模块具有此寄存器。低 8 位被忽略：写入无效，读取返回零。<br>复位类型：SYSRSn |

## 15.16.2.38 TBPRD 寄存器

偏移地址：C6h　复位值：0000h

时基周期寄存器

表 15-99 TBPRD 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| TBPRD | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TBPRD | | | | | | | |
| R/W-0h | | | | | | | |

表 15-100 TBPRD 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | TBPRD | R/W | 0h | 时基周期寄存器。决定时基计数器周期，从而设置 PWM 频率。TBCTL[PRDLD] 控制影子模式的使能 / 禁用，默认为影子模式。<br>TBCTL[PRDLD] = 0：使能影子模式，读写自动访问影子寄存器；时基计数器等于零时，从影子寄存器装载活动寄存器。<br>TBCTL[PRDLD] = 1：禁用影子模式，读写直接访问实际控制硬件的活动寄存器。<br>活动寄存器和影子寄存器共享同一存储器映射地址。<br>复位类型：SYSRSn |

## 15.16.2.39 CMPA 寄存器

偏移地址：D4h　复位值：00000000h

计数比较 A 寄存器

表 15-101 CMPA 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CMPA | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CMPA | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CMPAHR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMPAHR | | | | | | | |
| R/W-0h | | | | | | | |

表 15-102 CMPA 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | CMPA | R/W | 0h | 比较 A 寄存器。活动 CMPA 的值持续与 TBCTR 比较；相等时，计数比较模块产生“时基计数器等于计数比较 A”事件。事件送往动作限定器，经过限定后转换为一个或多个动作；根据 AQCTLA 和 AQCTLB 的配置，动作可作用于 EPWMxA 或 EPWMxB。可定义的动作包括：不执行动作（忽略事件）；清除（将 EPWMxA 和 / 或 EPWMxB 拉低）；置位（将 EPWMxA 和 / 或 EPWMxB 拉高）；翻转 EPWMxA 和 / 或 EPWMxB。<br>CMPCTL[SHDWAMODE] 控制影子模式的使能 / 禁用，默认使能。<br>若 CMPCTL[SHDWAMODE] = 0，读写自动访问影子寄存器；CMPCTL[LOADAMODE] 决定哪个事件将影子值装载至活动寄存器。写入前，可读取 CMPCTL[SHDWAFULL] 判断影子寄存器是否已满。<br>若 CMPCTL[SHDWAMODE] = 1，禁用影子寄存器，读写直接访问实际控制硬件的活动寄存器。<br>无论哪种模式，活动寄存器和影子寄存器都共享同一存储器映射地址。<br>复位类型：SYSRSn |
| 15-0 | CMPAHR | R/W | 0h | 比较 A HRPWM 扩展寄存器。高 8 位包含计数比较 A 值的高分辨率部分（最高有效 8 位）。CMPA:CMPAHR 可用一次 32 位读写访问。如 CMPA 说明所述，由 CMPCTL[SHDWAMODE] 使能或禁用影子模式。此寄存器低 8 位被忽略。<br>复位类型：SYSRSn |

## 15.16.2.40 CMPB 寄存器

偏移地址：D8h　复位值：00000000h

比较 B 寄存器

表 15-103 CMPB 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CMPB | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CMPB | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CMPBHR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMPBHR | | | | | | | |
| R/W-0h | | | | | | | |

表 15-104 CMPB 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | CMPB | R/W | 0h | 比较 B 寄存器。活动 CMPB 的值持续与 TBCTR 比较；相等时，计数比较模块产生“时基计数器等于计数比较 B”事件。事件送往动作限定器，经过限定后转换为一个或多个动作；根据 AQCTLA 和 AQCTLB 的配置，动作可作用于 EPWMxA 或 EPWMxB。可定义的动作包括：不执行动作（忽略事件）；清除（将 EPWMxA 和 / 或 EPWMxB 拉低）；置位（将 EPWMxA 和 / 或 EPWMxB 拉高）；翻转 EPWMxA 和 / 或 EPWMxB。<br>CMPCTL[SHDWBMODE] 控制影子模式的使能 / 禁用，默认使能。<br>若 CMPCTL[SHDWBMODE] = 0，读写自动访问影子寄存器；CMPCTL[LOADBMODE] 决定哪个事件将影子值装载至活动寄存器。写入前，可读取 CMPCTL[SHDWBFULL] 判断影子寄存器是否已满。<br>若 CMPCTL[SHDWBMODE] = 1，禁用影子寄存器，读写直接访问实际控制硬件的活动寄存器。<br>无论哪种模式，活动寄存器和影子寄存器都共享同一存储器映射地址。<br>复位类型：SYSRSn |
| 15-0 | CMPBHR | R/W | 0h | 比较 B 高分辨率位。此寄存器低 8 位被忽略。<br>复位类型：SYSRSn |

### 15.16.2.41 CMPC 寄存器

偏移地址：DEh　复位值：0000h

计数比较 C 寄存器

LINK 功能访问必须始终使用 16 位访问。

表 15-105 CMPC 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CMPC | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMPC | | | | | | | |
| R/W-0h | | | | | | | |

表 15-106 CMPC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | CMPC | R/W | 0h | 比较 C 寄存器。活动 CMPC 的值持续与 TBCTR 比较；相等时，计数比较模块产生“时基计数器等于计数比较 C”事件。<br>CMPCTL2[SHDWCMODE] 控制影子模式的使能 / 禁用，默认使能。<br>若 CMPCTL2[SHDWCMODE] = 0，读写自动访问影子寄存器；CMPCTL2[LOADCMODE] 决定哪个事件将影子值装载至活动寄存器。<br>若 CMPCTL2[SHDWCMODE] = 1，禁用影子寄存器，读写直接访问实际控制硬件的活动寄存器。<br>无论哪种模式，活动寄存器和影子寄存器都共享同一存储器映射地址。<br>复位类型：SYSRSn |

### 15.16.2.42 CMPD 寄存器

偏移地址：E2h　复位值：0000h

计数比较 D 寄存器

LINK 功能访问必须始终使用 16 位访问。

表 15-107 CMPD 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CMPD | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CMPD | | | | | | | |
| R/W-0h | | | | | | | |

表 15-108 CMPD 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | CMPD | R/W | 0h | 比较 D 寄存器。活动 CMPD 的值持续与 TBCTR 比较；相等时，计数比较模块产生“时基计数器等于计数比较 D”事件。<br>CMPCTL2[SHDWDMODE] 控制影子模式的使能 / 禁用，默认使能。<br>若 CMPCTL2[SHDWDMODE] = 0，读写自动访问影子寄存器；CMPCTL2[LOADDMODE] 决定哪个事件将影子值装载至活动寄存器。<br>若 CMPCTL2[SHDWDMODE] = 1，禁用影子寄存器，读写直接访问实际控制硬件的活动寄存器。<br>无论哪种模式，活动寄存器和影子寄存器都共享同一存储器映射地址。<br>复位类型：SYSRSn |

## 15.16.2.43 GLDCTL2 寄存器

偏移地址：E8h　复位值：0000h

全局 PWM 装载控制寄存器 2

表 15-109 GLDCTL2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | GFRCLD | OSHTLD |
| R-0-0h | | | | | | R-0/W1S-0h | R-0/W1S-0h |

表 15-110 GLDCTL2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-2 | RESERVED | R-0 | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | GFRCLD | R-0/W1S | 0h | 单次模式下强制装载事件。<br>0：忽略写 0，读取始终返回 0。<br>1：在事件预分频计数器输入端强制产生一次装载事件，用于测试和 / 或在全局装载模式下通过软件强制装载事件。<br>复位类型：SYSRSn |
| 0 | OSHTLD | R-0/W1S | 0h | 单次模式下使能重新装载事件。<br>0：忽略写 0，读取始终返回 0。<br>1：置位单次锁存条件。所选装载选通信号发生时，执行一次影子到活动重新装载，然后清除锁存器。因此写 1 允许一个装载选通事件通过，并阻止后续选通事件。<br>复位类型：SYSRSn |

### 15.16.2.44 SWVDELVAL 寄存器

偏移地址：EEh　复位值：0000h

软件谷底模式延迟寄存器

表 15-111 SWVDELVAL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SWVDELVAL | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SWVDELVAL | | | | | | | |
| R/W-0h | | | | | | | |

表 15-112 SWVDELVAL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | SWVDELVAL | R/W | 0h | 软件谷底延迟值寄存器。可选用此寄存器定义由 VCAPCTL[VDELAYDIV] 指定的硬件计算延迟 HWDELAYVAL 的偏移值。<br>复位类型：SYSRSn |

### 15.16.2.45 TZSEL 寄存器

偏移地址：100h　复位值：0000h

跳闸区选择寄存器

表 15-113 TZSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DCBEVT1 | DCAEVT1 | OSHT6 | OSHT5 | OSHT4 | OSHT3 | OSHT2 | OSHT1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCBEVT2 | DCAEVT2 | CBC6 | CBC5 | CBC4 | CBC3 | CBC2 | CBC1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 15-114 TZSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | DCBEVT1 | R/W | 0h | 数字比较输出 B 事件 1 选择。<br>0：禁用 DCBEVT1 作为本 ePWM 模块的单次跳闸源。<br>1：使能 DCBEVT1 作为本 ePWM 模块的单次跳闸源。<br>复位类型：SYSRSn |
| 14 | DCAEVT1 | R/W | 0h | 数字比较输出 A 事件 1 选择。<br>0：禁用 DCAEVT1 作为本 ePWM 模块的单次跳闸源。<br>1：使能 DCAEVT1 作为本 ePWM 模块的单次跳闸源。<br>复位类型：SYSRSn |
| 13 | OSHT6 | R/W | 0h | 跳闸区 6（TZ6）选择。<br>0：禁用 TZ6 作为本 ePWM 模块的单次跳闸源。<br>1：使能 TZ6 作为本 ePWM 模块的单次跳闸源。<br>复位类型：SYSRSn |
| 12 | OSHT5 | R/W | 0h | 跳闸区 5（TZ5）选择。<br>0：禁用 TZ5 作为本 ePWM 模块的单次跳闸源。<br>1：使能 TZ5 作为本 ePWM 模块的单次跳闸源。<br>复位类型：SYSRSn |
| 11 | OSHT4 | R/W | 0h | 跳闸区 4（TZ4）选择。<br>0：禁用 TZ4 作为本 ePWM 模块的单次跳闸源。<br>1：使能 TZ4 作为本 ePWM 模块的单次跳闸源。<br>复位类型：SYSRSn |
| 10 | OSHT3 | R/W | 0h | 跳闸区 3（TZ3）选择。<br>0：禁用 TZ3 作为本 ePWM 模块的单次跳闸源。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：使能 TZ3 作为本 ePWM 模块的单次跳闸源。<br>复位类型：SYSRSn |
| 9 | OSHT2 | R/W | 0h | 跳闸区 2（TZ2）选择。<br>0：禁用 TZ2 作为本 ePWM 模块的单次跳闸源。<br>1：使能 TZ2 作为本 ePWM 模块的单次跳闸源。<br>复位类型：SYSRSn |
| 8 | OSHT1 | R/W | 0h | 跳闸区 1（TZ1）选择。<br>0：禁用 TZ1 作为本 ePWM 模块的单次跳闸源。<br>1：使能 TZ1 作为本 ePWM 模块的单次跳闸源。<br>复位类型：SYSRSn |
| 7 | DCBEVT2 | R/W | 0h | 数字比较输出 B 事件 2 选择。<br>0：禁用 DCBEVT2 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>1：使能 DCBEVT2 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>复位类型：SYSRSn |
| 6 | DCAEVT2 | R/W | 0h | 数字比较输出 A 事件 2 选择。<br>0：禁用 DCAEVT2 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>1：使能 DCAEVT2 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>复位类型：SYSRSn |
| 5 | CBC6 | R/W | 0h | 跳闸区 6（TZ6）选择。<br>0：禁用 TZ6 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>1：使能 TZ6 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>复位类型：SYSRSn |
| 4 | CBC5 | R/W | 0h | 跳闸区 5（TZ5）选择。<br>0：禁用 TZ5 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>1：使能 TZ5 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>复位类型：SYSRSn |
| 3 | CBC4 | R/W | 0h | 跳闸区 4（TZ4）选择。<br>0：禁用 TZ4 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>1：使能 TZ4 作为本 ePWM 模块的逐周期 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | （CBC）跳闸源。<br>复位类型：SYSRSn |
| 2 | CBC3 | R/W | 0h | 跳闸区 3（TZ3）选择。<br>0：禁用 TZ3 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>1：使能 TZ3 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>复位类型：SYSRSn |
| 1 | CBC2 | R/W | 0h | 跳闸区 2（TZ2）选择。<br>0：禁用 TZ2 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>1：使能 TZ2 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>复位类型：SYSRSn |
| 0 | CBC1 | R/W | 0h | 跳闸区 1（TZ1）选择。<br>0：禁用 TZ1 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>1：使能 TZ1 作为本 ePWM 模块的逐周期（CBC）跳闸源。<br>复位类型：SYSRSn |

## 15.16.2.46 TZDCSEL 寄存器

偏移地址：104h　复位值：0000h

跳闸区数字比较器选择寄存器

表 15-115 TZDCSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | DCBEVT2 | | | DCBEVT1 |
| R-0-0h | | | | R/W-0h | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCBEVT1 | | DCAEVT2 | | DCAEVT1 | | | |
| R/W-0h | | R/W-0h | | R/W-0h | | | |

表 15-116 TZDCSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R-0 | 0h | 保留 |
| 11-9 | DCBEVT2 | R/W | 0h | 数字比较输出 B 事件 2 选择。<br>000：禁用事件。<br>001：DCBH = 低，DCBL = 任意。<br>010：DCBH = 高，DCBL = 任意。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 011：DCBL = 低，DCBH = 任意。<br>100：DCBL = 高，DCBH = 任意。<br>101：DCBL = 高，DCBH = 低。<br>110：保留。<br>111：保留。<br>复位类型：SYSRSn |
| 8-6 | DCBEVT1 | R/W | 0h | 数字比较输出 B 事件 1 选择。<br>000：禁用事件。<br>001：DCBH = 低，DCBL = 任意。<br>010：DCBH = 高，DCBL = 任意。<br>011：DCBL = 低，DCBH = 任意。<br>100：DCBL = 高，DCBH = 任意。<br>101：DCBL = 高，DCBH = 低。<br>110：保留。<br>111：保留。<br>复位类型：SYSRSn |
| 5-3 | DCAEVT2 | R/W | 0h | 数字比较输出 A 事件 2 选择。<br>000：禁用事件。<br>001：DCAH = 低，DCAL = 任意。<br>010：DCAH = 高，DCAL = 任意。<br>011：DCAL = 低，DCAH = 任意。<br>100：DCAL = 高，DCAH = 任意。<br>101：DCAL = 高，DCAH = 低。<br>110：保留。<br>111：保留。<br>复位类型：SYSRSn |
| 2-0 | DCAEVT1 | R/W | 0h | 数字比较输出 A 事件 1 选择。<br>000：禁用事件。<br>001：DCAH = 低，DCAL = 任意。<br>010：DCAH = 高，DCAL = 任意。<br>011：DCAL = 低，DCAH = 任意。<br>100：DCAL = 高，DCAH = 任意。<br>101：DCAL = 高，DCAH = 低。<br>110：保留。<br>111：保留。<br>复位类型：SYSRSn |

## 15.16.2.47 TZCTL 寄存器

偏移地址：108h　复位值：0000h

跳闸区控制寄存器

表 15-117 TZCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | DCBEVT2 | | DCBEVT1 | |
| R-0-0h | | | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCAEVT2 | | DCAEVT1 | | TZB | | TZA | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |

表 15-118 TZCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R-0 | 0h | 保留 |
| 11-10 | DCBEVT2 | R/W | 0h | 数字比较输出 B 事件 2 对 EPWMxB 的动作。<br>00：高阻态（EPWMxB 为高阻态）。<br>01：强制 EPWMxB 为高电平。<br>10：强制 EPWMxB 为低电平。<br>11：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 9-8 | DCBEVT1 | R/W | 0h | 数字比较输出 B 事件 1 对 EPWMxB 的动作。<br>00：高阻态（EPWMxB 为高阻态）。<br>01：强制 EPWMxB 为高电平。<br>10：强制 EPWMxB 为低电平。<br>11：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 7-6 | DCAEVT2 | R/W | 0h | 数字比较输出 A 事件 2 对 EPWMxA 的动作。<br>00：高阻态（EPWMxA 为高阻态）。<br>01：强制 EPWMxA 为高电平。<br>10：强制 EPWMxA 为低电平。<br>11：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 5-4 | DCAEVT1 | R/W | 0h | 数字比较输出 A 事件 1 对 EPWMxA 的动作。<br>00：高阻态（EPWMxA 为高阻态）。<br>01：强制 EPWMxA 为高电平。<br>10：强制 EPWMxA 为低电平。<br>11：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 3-2 | TZB | R/W | 0h | TZ1 至 TZ6、DCAEVT1/2、DCBEVT1/2 对 EPWMxB 的跳闸动作。跳闸事件发生时，对 EPWMxB 执行以下动作；哪些跳闸区引脚可产生事件由 TZSEL 定义。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 00：高阻态（EPWMxB 为高阻态）。<br>01：强制 EPWMxB 为高电平。<br>10：强制 EPWMxB 为低电平。<br>11：不对 EPWMxB 执行动作。<br>复位类型：SYSRSn |
| 1-0 | TZA | R/W | 0h | TZ1 至 TZ6、DCAEVT1/2、DCBEVT1/2 对 EPWMxA 的跳闸动作。跳闸事件发生时，对 EPWMxA 执行以下动作；哪些跳闸区引脚可产生事件由 TZSEL 定义。<br>00：高阻态（EPWMxA 为高阻态）。<br>01：强制 EPWMxA 为高电平。<br>10：强制 EPWMxA 为低电平。<br>11：不对 EPWMxA 执行动作。<br>复位类型：SYSRSn |

## 15.16.2.48 TZCTL2 寄存器

偏移地址：10Ah　复位值：0000h

附加跳闸区控制寄存器

表 15-119 TZCTL2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| ETZE | 保留 | | | TZBD | | | TZBU |
| R/W-0h | R-0-0h | | | R/W-0h | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TZBU | | TZAD | | | TZAU | | |
| R/W-0h | | R/W-0h | | | R/W-0h | | |

表 15-120 TZCTL2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | ETZE | R/W | 0h | TZCTL2 使能。<br>0：使用 TZCTL 的跳闸动作（兼容旧版 EPWM）。<br>1：使用 TZCTL2、TZCTLDCA、TZCTLDCB 定义的跳闸动作，忽略 TZCTL 设置。<br>复位类型：SYSRSn |
| 14-12 | RESERVED | R-0 | 0h | 保留 |
| 11-9 | TZBD | R/W | 0h | 向下计数时，TZ1 至 TZ6 对 EPWMxB 的跳闸动作。<br>000：高阻态（EPWMxB 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。<br>110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 8-6 | TZBU | R/W | 0h | 向上计数时，TZ1 至 TZ6、DCAEVT1/2、DCBEVT1/2 对 EPWMxB 的跳闸动作。<br>000：高阻态（EPWMxB 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。<br>011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。<br>110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 5-3 | TZAD | R/W | 0h | 向下计数时，TZ1 至 TZ6、DCAEVT1/2、DCBEVT1/2 对 EPWMxA 的跳闸动作。<br>000：高阻态（EPWMxA 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。<br>011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。<br>110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 2-0 | TZAU | R/W | 0h | 向上计数时，TZ1 至 TZ6、DCAEVT1/2、DCBEVT1/2 对 EPWMxA 的跳闸动作。<br>000：高阻态（EPWMxA 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。<br>011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。<br>110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |

## 15.16.2.49 TZCTLDCA 寄存器

偏移地址：10Ch　复位值：0000h

数字比较 A 跳闸区控制寄存器

表 15-121 TZCTLDCA 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | DCAEVT2D | | | DCAEVT2U |
| R-0-0h | | | | R/W-0h | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCAEVT2U | | DCAEVT1D | | | DCAEVT1U | | |
| R/W-0h | | R/W-0h | | | R/W-0h | | |

表 15-122 TZCTLDCA 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R-0 | 0h | 保留 |
| 11-9 | DCAEVT2D | R/W | 0h | 向下计数时，数字比较输出 A 事件 2 对 EPWMxA 的动作。<br>000：高阻态（EPWMxA 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。<br>011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。<br>110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 8-6 | DCAEVT2U | R/W | 0h | 向上计数时，数字比较输出 A 事件 2 对 EPWMxA 的动作。<br>000：高阻态（EPWMxA 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。<br>011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。<br>110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 5-3 | DCAEVT1D | R/W | 0h | 向下计数时，数字比较输出 A 事件 1 对 EPWMxA 的动作。<br>000：高阻态（EPWMxA 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。<br>011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 2-0 | DCAEVT1U | R/W | 0h | 向上计数时，数字比较输出 A 事件 1 对 EPWMxA 的动作。<br>000：高阻态（EPWMxA 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。<br>011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。<br>110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |

## 15.16.2.50 TZCTLDCB 寄存器

偏移地址：10Eh　复位值：0000h

数字比较 B 跳闸区控制寄存器

表 15-123 TZCTLDCB 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 保留 | | | | DCBEVT2D | | | DCBEVT2U |
| R-0-0h | | | | R/W-0h | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCBEVT2U | | DCBEVT1D | | DCBEVT1U | | | |
| R/W-0h | | R/W-0h | | R/W-0h | | | |

表 15-124 TZCTLDCB 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 15-12 | RESERVED | R-0 | 0h | 保留 |
| 11-9 | DCBEVT2D | R/W | 0h | 向下计数时，数字比较输出 B 事件 2 对 EPWMxB 的动作。<br>000：高阻态（EPWMxB 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。<br>011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。<br>110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 8-6 | DCBEVT2U | R/W | 0h | 向上计数时，数字比较输出 B 事件 2 对 EPWMxB 的动作。<br>000：高阻态（EPWMxB 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。<br>011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。<br>110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 5-3 | DCBEVT1D | R/W | 0h | 向下计数时，数字比较输出 B 事件 1 对 EPWMxB 的动作。<br>000：高阻态（EPWMxB 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。<br>011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。<br>110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |
| 2-0 | DCBEVT1U | R/W | 0h | 向上计数时，数字比较输出 B 事件 1 对 EPWMxB 的动作。<br>000：高阻态（EPWMxB 为高阻态）。<br>001：强制高电平。<br>010：强制低电平。<br>011：翻转（低变高，高变低）。<br>100：保留。<br>101：保留。<br>110：保留。<br>111：不执行动作，禁用跳闸动作。<br>复位类型：SYSRSn |

## 15.16.2.51 TZEINT 寄存器

偏移地址：11Ah　复位值：0000h

跳闸区中断使能寄存器

表 15-125 TZEINT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | DCBEVT2 | DCBEVT1 | DCAEVT2 | DCAEVT1 | OST | CBC | 保留 |
| R-0-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0-0h |

表 15-126 TZEINT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-7 | RESERVED | R-0 | 0h | 保留 |
| 6 | DCBEVT2 | R/W | 0h | 数字比较输出 B 事件 2 中断使能。<br>0：禁用。<br>1：使能。<br>复位类型：SYSRSn |
| 5 | DCBEVT1 | R/W | 0h | 数字比较输出 B 事件 1 中断使能。<br>0：禁用。<br>1：使能。<br>复位类型：SYSRSn |
| 4 | DCAEVT2 | R/W | 0h | 数字比较输出 A 事件 2 中断使能。<br>0：禁用。<br>1：使能。<br>复位类型：SYSRSn |
| 3 | DCAEVT1 | R/W | 0h | 数字比较输出 A 事件 1 中断使能。<br>0：禁用。<br>1：使能。<br>复位类型：SYSRSn |
| 2 | OST | R/W | 0h | 跳闸区单次中断使能。<br>0：禁用单次中断生成。<br>1：使能中断生成；单次跳闸事件将产生 EPWMx_TZINT CLIC 中断。<br>复位类型：SYSRSn |
| 1 | CBC | R/W | 0h | 跳闸区逐周期中断使能。<br>0：禁用逐周期中断生成。<br>1：使能中断生成；逐周期跳闸事件将产生 EPWMx_TZINT CLIC 中断。<br>复位类型：SYSRSn |
| 0 | RESERVED | R-0 | 0h | 保留 |

## 15.16.2.52 TZFLG 寄存器

偏移地址：126h　复位值：0000h

跳闸区标志寄存器

表 15-127 TZFLG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | DCBEVT2 | DCBEVT1 | DCAEVT2 | DCAEVT1 | OST | CBC | INT |
| R-0-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 15-128 TZFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-7 | RESERVED | R-0 | 0h | 保留 |
| 6 | DCBEVT2 | R | 0h | 数字比较输出 B 事件 2 的锁存状态标志。<br>0：DCBEVT2 未发生跳闸事件。<br>1：DCBEVT2 所定义事件已发生跳闸。<br>复位类型：SYSRSn |
| 5 | DCBEVT1 | R | 0h | 数字比较输出 B 事件 1 的锁存状态标志。<br>0：DCBEVT1 未发生跳闸事件。<br>1：DCBEVT1 所定义事件已发生跳闸。<br>复位类型：SYSRSn |
| 4 | DCAEVT2 | R | 0h | 数字比较输出 A 事件 2 的锁存状态标志。<br>0：DCAEVT2 未发生跳闸事件。<br>1：DCAEVT2 所定义事件已发生跳闸。<br>复位类型：SYSRSn |
| 3 | DCAEVT1 | R | 0h | 数字比较输出 A 事件 1 的锁存状态标志。<br>0：DCAEVT1 未发生跳闸事件。<br>1：DCAEVT1 所定义事件已发生跳闸。<br>复位类型：SYSRSn |
| 2 | OST | R | 0h | 单次跳闸事件锁存状态标志。<br>0：未发生单次跳闸事件。<br>1：被选作单次跳闸源的引脚已发生跳闸事件。向 TZCLR 写入适当值可清除此位。<br>复位类型：SYSRSn |
| 1 | CBC | R | 0h | 逐周期跳闸事件锁存状态标志。<br>0：未发生逐周期跳闸事件。<br>1：被选作逐周期跳闸源的信号已发生跳闸事件。TZFLG[CBC] 保持置位，直到用户手动清除；清除时若逐周期跳闸事件仍存在，CBC 会立即再次置位。若跳闸条件已消失，ePWM 时基计数器达到零（TBCTR = 0x00）时，自动解除信号上的指定状态。无论在周期中的何时清除 CBC 标志，只有 TBCTR = 0x00 时才解除信号 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 状态。<br>向 TZCLR 写入适当值可清除此位。<br>复位类型：SYSRSn |
| 0 | INT | R | 0h | 跳闸中断锁存状态标志。<br>0：未产生中断。<br>1：跳闸条件已产生 EPWMx_TZINT CLIC 中断。<br>清除此标志前不再产生 EPWMx_TZINT CLIC 中断。若清除时 CBC 或 OST 仍置位，将再次产生中断脉冲。清除所有标志位可防止继续产生中断。向 TZCLR 写入适当值可清除此位。<br>复位类型：SYSRSn |

## 15.16.2.53 TZCBCFLG 寄存器

偏移地址：128h　复位值：0000h

跳闸区 CBC 标志寄存器

表 15-129 TZCBCFLG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCBEVT2 | DCAEVT2 | CBC6 | CBC5 | CBC4 | CBC3 | CBC2 | CBC1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 15-130 TZCBCFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |
| 7 | DCBEVT2 | R | 0h | 数字比较输出 B 事件 2 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 DCBEVT2 未发生跳闸。<br>1：读到 1 表示 DCBEVT2 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 6 | DCAEVT2 | R | 0h | 数字比较输出 A 事件 2 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 DCAEVT2 未发生跳闸。<br>1：读到 1 表示 DCAEVT2 所选事件已发生跳闸。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 5 | CBC6 | R | 0h | CBC6 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 CBC6 未发生跳闸。<br>1：读到 1 表示 CBC6 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 4 | CBC5 | R | 0h | CBC5 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 CBC5 未发生跳闸。<br>1：读到 1 表示 CBC5 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 3 | CBC4 | R | 0h | CBC4 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 CBC4 未发生跳闸。<br>1：读到 1 表示 CBC4 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 2 | CBC3 | R | 0h | CBC3 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 CBC3 未发生跳闸。<br>1：读到 1 表示 CBC3 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 1 | CBC2 | R | 0h | CBC2 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 CBC2 未发生跳闸。<br>1：读到 1 表示 CBC2 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 0 | CBC1 | R | 0h | CBC1 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 CBC1 未发生跳闸。<br>1：读到 1 表示 CBC1 所选事件已发生跳闸。<br>复位类型：SYSRSn |

## 15.16.2.54 TZOSTFLG 寄存器

偏移地址：12Ah　复位值：0000h

跳闸区 OST 标志寄存器

表 15-131 TZOSTFLG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCBEVT1 | DCAEVT1 | OST6 | OST5 | OST4 | OST3 | OST2 | OST1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 15-132 TZOSTFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 7 | DCBEVT1 | R | 0h | 数字比较输出 B 事件 1 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 DCBEVT1 未发生跳闸。<br>1：读到 1 表示 DCBEVT1 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 6 | DCAEVT1 | R | 0h | 数字比较输出 A 事件 1 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 DCAEVT1 未发生跳闸。<br>1：读到 1 表示 DCAEVT1 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 5 | OST6 | R | 0h | OST6 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 OST6 未发生跳闸。<br>1：读到 1 表示 OST6 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 4 | OST5 | R | 0h | OST5 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 OST5 未发生跳闸。<br>1：读到 1 表示 OST5 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 3 | OST4 | R | 0h | OST4 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 OST4 未发生跳闸。<br>1：读到 1 表示 OST4 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 2 | OST3 | R | 0h | OST3 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 OST3 未发生跳闸。<br>1：读到 1 表示 OST3 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 1 | OST2 | R | 0h | OST2 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 OST2 未发生跳闸。<br>1：读到 1 表示 OST2 所选事件已发生跳闸。<br>复位类型：SYSRSn |
| 0 | OST1 | R | 0h | OST1 跳闸锁存器的锁存状态标志。<br>0：读到 0 表示 OST1 未发生跳闸。<br>1：读到 1 表示 OST1 所选事件已发生跳闸。<br>复位类型：SYSRSn |

## 15.16.2.55 TZCLR 寄存器

偏移地址：12Eh　复位值：0000h

跳闸区清除寄存器

表 15-133 TZCLR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CBCPULSE | | 保留 | | | | | |
| R/W-0h | | R-0-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | DCBEVT2 | DCBEVT1 | DCAEVT2 | DCAEVT1 | OST | CBC | INT |
| R-0-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 15-134 TZCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | CBCPULSE | R/W | 0h | 逐周期（CBC）跳闸锁存器清除脉冲。选择清除 CBC 跳闸锁存器的脉冲。<br>00：CTR = zero 脉冲清除（与旧版设计相同）。<br>01：CTR = PRD 脉冲清除。<br>10：CTR = zero 或 CTR = PRD 脉冲清除。<br>11：不清除 CBC 跳闸锁存器。<br>复位类型：SYSRSn |
| 13-7 | RESERVED | R-0 | 0h | 保留 |
| 6 | DCBEVT2 | R-0/W1S | 0h | 数字比较输出 B 事件 2 清除标志。<br>0：写 0 无效，读取始终返回 0。<br>1：写 1 清除 DCBEVT2 事件跳闸条件。<br>复位类型：SYSRSn |
| 5 | DCBEVT1 | R-0/W1S | 0h | 数字比较输出 B 事件 1 清除标志。<br>0：写 0 无效，读取始终返回 0。<br>1：写 1 清除 DCBEVT1 事件跳闸条件。<br>复位类型：SYSRSn |
| 4 | DCAEVT2 | R-0/W1S | 0h | 数字比较输出 A 事件 2 清除标志。<br>0：写 0 无效，读取始终返回 0。<br>1：写 1 清除 DCAEVT2 事件跳闸条件。<br>复位类型：SYSRSn |
| 3 | DCAEVT1 | R-0/W1S | 0h | 数字比较输出 A 事件 1 清除标志。<br>0：写 0 无效，读取始终返回 0。<br>1：写 1 清除 DCAEVT1 事件跳闸条件。<br>复位类型：SYSRSn |
| 2 | OST | R-0/W1S | 0h | 单次（OST）跳闸锁存器清除标志。<br>0：无效，读取始终返回 0。<br>1：清除此跳闸（置位）条件。<br>复位类型：SYSRSn |
| 1 | CBC | R-0/W1S | 0h | 逐周期（CBC）跳闸锁存器清除标志。<br>0：无效，读取始终返回 0。<br>1：清除此跳闸（置位）条件。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 0 | INT | R-0/W1S | 0h | 全局中断清除标志。<br>0：无效，读取始终返回 0。<br>1：清除本 ePWM 模块的跳闸中断标志 TZFLG[INT]。<br>注意：清除此标志前，不再产生 EPWMx_TZINT CLIC 中断。若清除 TZFLG[INT] 时仍有其他标志位置位，将再次产生中断脉冲。清除所有标志位可防止继续产生中断。<br>复位类型：SYSRSn |

## 15.16.2.56 TZCBCCLR 寄存器

偏移地址：130h　复位值：0000h

跳闸区 CBC 清除寄存器

表 15-135 TZCBCCLR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCBEVT2 | DCAEVT2 | CBC6 | CBC5 | CBC4 | CBC3 | CBC2 | CBC1 |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 15-136 TZCBCCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |
| 7 | DCBEVT2 | R-0/W1S | 0h | 数字比较输出 B 事件 2 的逐周期（CBC）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZCBCFLG[DCBEVT2]。<br>复位类型：SYSRSn |
| 6 | DCAEVT2 | R-0/W1S | 0h | 数字比较输出 A 事件 2 的逐周期（CBC）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZCBCFLG[DCAEVT2]。<br>复位类型：SYSRSn |
| 5 | CBC6 | R-0/W1S | 0h | CBC6 的逐周期（CBC）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZCBCFLG[CBC6]。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 4 | CBC5 | R-0/W1S | 0h | CBC5 的逐周期（CBC）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZCBCFLG[CBC5]。<br>复位类型：SYSRSn |
| 3 | CBC4 | R-0/W1S | 0h | CBC4 的逐周期（CBC）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZCBCFLG[CBC4]。<br>复位类型：SYSRSn |
| 2 | CBC3 | R-0/W1S | 0h | CBC3 的逐周期（CBC）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZCBCFLG[CBC3]。<br>复位类型：SYSRSn |
| 1 | CBC2 | R-0/W1S | 0h | CBC2 的逐周期（CBC）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZCBCFLG[CBC2]。<br>复位类型：SYSRSn |
| 0 | CBC1 | R-0/W1S | 0h | CBC1 的逐周期（CBC）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZCBCFLG[CBC1]。<br>复位类型：SYSRSn |

## 15.16.2.57 TZOSTCLR 寄存器

偏移地址：132h　复位值：0000h

跳闸区 OST 清除寄存器

表 15-137 TZOSTCLR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCBEVT1 | DCAEVT1 | OST6 | OST5 | OST4 | OST3 | OST2 | OST1 |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 15-138 TZOSTCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |
| 7 | DCBEVT1 | R-0/W1S | 0h | 数字比较输出 B 事件 1 的单次（OST）跳闸清除标志。<br>0：写 0 无效。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：写 1 清除 TZOSTFLG[DCBEVT1]。<br>复位类型：SYSRSn |
| 6 | DCAEVT1 | R-0/W1S | 0h | 数字比较输出 A 事件 1 的单次（OST）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZOSTFLG[DCAEVT1]。<br>复位类型：SYSRSn |
| 5 | OST6 | R-0/W1S | 0h | OST6 的单次（OST）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZOSTFLG[OST6]。<br>复位类型：SYSRSn |
| 4 | OST5 | R-0/W1S | 0h | OST5 的单次（OST）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZOSTFLG[OST5]。<br>复位类型：SYSRSn |
| 3 | OST4 | R-0/W1S | 0h | OST4 的单次（OST）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZOSTFLG[OST4]。<br>复位类型：SYSRSn |
| 2 | OST3 | R-0/W1S | 0h | OST3 的单次（OST）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZOSTFLG[OST3]。<br>复位类型：SYSRSn |
| 1 | OST2 | R-0/W1S | 0h | OST2 的单次（OST）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZOSTFLG[OST2]。<br>复位类型：SYSRSn |
| 0 | OST1 | R-0/W1S | 0h | OST1 的单次（OST）跳闸清除标志。<br>0：写 0 无效。<br>1：写 1 清除 TZOSTFLG[OST1]。<br>复位类型：SYSRSn |

### 15.16.2.58 TZFRC 寄存器

偏移地址：136h　复位值：0000h

跳闸区强制寄存器

表 15-139 TZFRC 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | DCBEVT2 | DCBEVT1 | DCAEVT2 | DCAEVT1 | OST | CBC | 保留 |
| R-0-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0-0h |

表 15-140 TZFRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-7 | RESERVED | R-0 | 0h | 保留 |
| 6 | DCBEVT2 | R-0/W1S | 0h | 数字比较输出 B 事件 2 强制标志。<br>0：写 0 无效，读取始终返回 0。<br>1：写 1 强制产生 DCBEVT2 事件跳闸条件，并置位 TZFLG[DCBEVT2]。<br>复位类型：SYSRSn |
| 5 | DCBEVT1 | R-0/W1S | 0h | 数字比较输出 B 事件 1 强制标志。<br>0：写 0 无效，读取始终返回 0。<br>1：写 1 强制产生 DCBEVT1 事件跳闸条件，并置位 TZFLG[DCBEVT1]。<br>复位类型：SYSRSn |
| 4 | DCAEVT2 | R-0/W1S | 0h | 数字比较输出 A 事件 2 强制标志。<br>0：写 0 无效，读取始终返回 0。<br>1：写 1 强制产生 DCAEVT2 事件跳闸条件，并置位 TZFLG[DCAEVT2]。<br>复位类型：SYSRSn |
| 3 | DCAEVT1 | R-0/W1S | 0h | 数字比较输出 A 事件 1 强制标志。<br>0：写 0 无效，读取始终返回 0。<br>1：写 1 强制产生 DCAEVT1 事件跳闸条件，并置位 TZFLG[DCAEVT1]。<br>复位类型：SYSRSn |
| 2 | OST | R-0/W1S | 0h | 通过软件强制产生单次跳闸事件。<br>0：忽略写 0，读取始终返回 0。<br>1：强制产生单次跳闸事件，并置位 TZFLG[OST]。<br>复位类型：SYSRSn |
| 1 | CBC | R-0/W1S | 0h | 通过软件强制产生逐周期跳闸事件。<br>0：忽略写 0，读取始终返回 0。<br>1：强制产生逐周期跳闸事件，并置位 TZFLG[CBC]。<br>复位类型：SYSRSn |
| 0 | RESERVED | R-0 | 0h | 保留 |

### 15.16.2.59 ETSEL 寄存器

偏移地址：148h　复位值：0000h

事件触发选择寄存器

表 15-141 ETSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SOCBEN | SOCBSEL | | | SOCAEN | SOCASEL | | |
| R/W-0h | R/W-0h | | | R/W-0h | R/W-0h | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | INTSELCMP | SOCBSELCMP | SOCASELCMP | INTEN | INTSEL | | |
| R-0-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | | |

表 15-142 ETSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | SOCBEN | R/W | 0h | ADC 启动转换 B（EPWMxSOCB）脉冲使能。<br>0：禁用 EPWMxSOCB。<br>1：使能 EPWMxSOCB 脉冲。<br>复位类型：SYSRSn |
| 14-12 | SOCBSEL | R/W | 0h | EPWMxSOCB 选择选项，决定何时产生脉冲。<br>000：使能 DCBEVT1.soc 事件。<br>001：使能 TBCTR = 0x00 事件。<br>010：使能 TBCTR = TBPRD 事件。<br>011：使能 TBCTR = 0x00 或 TBCTR = TBPRD 事件，适用于上下计数模式。<br>100：使能向上计数时 TBCTR 等于 CMPA 或 CMPC 的事件。<br>101：使能向下计数时 TBCTR 等于 CMPA 或 CMPC 的事件。<br>110：使能向上计数时 TBCTR 等于 CMPB 或 CMPD 的事件。<br>111：使能向下计数时 TBCTR 等于 CMPB 或 CMPD 的事件。<br>具体比较事件由 SOCBSELCMP 位决定。<br>复位类型：SYSRSn |
| 11 | SOCAEN | R/W | 0h | ADC 启动转换 A（EPWMxSOCA）脉冲使能。<br>0：禁用 EPWMxSOCA。<br>1：使能 EPWMxSOCA 脉冲。<br>复位类型：SYSRSn |
| 10-8 | SOCASEL | R/W | 0h | EPWMxSOCA 选择选项，决定何时产生脉冲。<br>000：使能 DCAEVT1.soc 事件。<br>001：使能 TBCTR = 0x00 事件。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 010：使能 TBCTR = TBPRD 事件。<br>011：使能 TBCTR = 0x00 或 TBCTR = TBPRD 事件，适用于上下计数模式。<br>100：使能向上计数时 TBCTR 等于 CMPA 或 CMPC 的事件。<br>101：使能向下计数时 TBCTR 等于 CMPA 或 CMPC 的事件。<br>110：使能向上计数时 TBCTR 等于 CMPB 或 CMPD 的事件。<br>111：使能向下计数时 TBCTR 等于 CMPB 或 CMPD 的事件。<br>具体比较事件由 SOCASELCMP 位决定。<br>复位类型：SYSRSn |
| 7 | RESERVED | R-0 | 0h | 保留 |
| 6 | INTSELCMP | R/W | 0h | EPWMxINT 比较寄存器选择。<br>0：将向上 / 向下计数时 TBCTR 等于 CMPA，以及向上 / 向下计数时 TBCTR 等于 CMPB 的事件送入 INTSEL 选择多路复用器。<br>1：将向上 / 向下计数时 TBCTR 等于 CMPC，以及向上 / 向下计数时 TBCTR 等于 CMPD 的事件送入 INTSEL 选择多路复用器。<br>复位类型：SYSRSn |
| 5 | SOCBSELCMP | R/W | 0h | EPWMxSOCB 比较寄存器选择。<br>0：将向上 / 向下计数时 TBCTR 等于 CMPA，以及向上 / 向下计数时 TBCTR 等于 CMPB 的事件送入 SOCBSEL 选择多路复用器。<br>1：将向上 / 向下计数时 TBCTR 等于 CMPC，以及向上 / 向下计数时 TBCTR 等于 CMPD 的事件送入 SOCBSEL 选择多路复用器。<br>复位类型：SYSRSn |
| 4 | SOCASELCMP | R/W | 0h | EPWMxSOCA 比较寄存器选择。<br>0：将向上 / 向下计数时 TBCTR 等于 CMPA，以及向上 / 向下计数时 TBCTR 等于 CMPB 的事件送入 SOCASEL 选择多路复用器。<br>1：将向上 / 向下计数时 TBCTR 等于 CMPC，以及向上 / 向下计数时 TBCTR 等于 CMPD 的事件送入 SOCASEL 选择多路复用器。<br>复位类型：SYSRSn |
| 3 | INTEN | R/W | 0h | ePWM 中断 EPWMx_INT 生成使能。<br>0：禁用 EPWMx_INT 生成。<br>1：使能 EPWMx_INT 生成。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 2-0 | INTSEL | R/W | 0h | EPWMx_INT 选择选项，决定何时产生中断。<br>000：保留。<br>001：使能 TBCTR = 0x00 事件。<br>010：使能 TBCTR = TBPRD 事件。<br>011：使能 TBCTR = 0x00 或 TBCTR = TBPRD 事件，适用于上下计数模式。<br>100：使能向上计数时 TBCTR 等于 CMPA 或 CMPC 的事件。<br>101：使能向下计数时 TBCTR 等于 CMPA 或 CMPC 的事件。<br>110：使能向上计数时 TBCTR 等于 CMPB 或 CMPD 的事件。<br>111：使能向下计数时 TBCTR 等于 CMPB 或 CMPD 的事件。<br>具体比较事件由 INTSELCMP 位决定。<br>复位类型：SYSRSn |

## 15.16.2.60 ETPS 寄存器

偏移地址：14Ch　复位值：0000h

事件触发预分频寄存器

表 15-143 ETPS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SOCBCNT | | SOCBPRD | | SOCACNT | | SOCAPRD | |
| R-0h | | R/W-0h | | R-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | SOCPSSEL | INTPSSEL | INTCNT | | INTPRD | |
| R-0-0h | | R/W-0h | R/W-0h | R-0h | | R/W-0h | |

表 15-144 ETPS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | SOCBCNT | R | 0h | ePWM ADC 启动转换 B 事件（EPWMxSOCB）计数器。表示 ETSEL[SOCBSEL] 所选事件已发生的次数。<br>00：未发生事件。<br>01：已发生 1 个事件。<br>10：已发生 2 个事件。<br>11：已发生 3 个事件。<br>复位类型：SYSRSn |
| 13-12 | SOCBPRD | R/W | 0h | ePWM ADC 启动转换 B 事件（EPWMxSOCB）周期选择。决定 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | ETSEL[SOCBSEL] 所选事件发生多少次后生成 EPWMxSOCB 脉冲。必须先使能脉冲（ETSEL[SOCBEN] = 1）。即使前次转换已使 ETFLG[SOCB] = 1，仍会生成 SOCB 脉冲；脉冲产生后 ETPS[SOCBCNT] 自动清零。<br>00：禁用 SOCB 事件计数器，不产生 EPWMxSOCB 脉冲。<br>01：第 1 个事件时产生脉冲，ETPS[SOCBCNT] = 0,1。<br>10：第 2 个事件时产生脉冲，ETPS[SOCBCNT] = 1,0。<br>11：第 3 个事件时产生脉冲，ETPS[SOCBCNT] = 1,1。<br>复位类型：SYSRSn |
| 11-10 | SOCACNT | R | 0h | ePWM ADC 启动转换 A 事件（EPWMxSOCA）计数器。表示 ETSEL[SOCASEL] 所选事件已发生的次数。<br>00：未发生事件。<br>01：已发生 1 个事件。<br>10：已发生 2 个事件。<br>11：已发生 3 个事件。<br>复位类型：SYSRSn |
| 9-8 | SOCAPRD | R/W | 0h | ePWM ADC 启动转换 A 事件（EPWMxSOCA）周期选择。决定 ETSEL[SOCASEL] 所选事件发生多少次后生成 EPWMxSOCA 脉冲。必须先使能脉冲（ETSEL[SOCAEN] = 1）。即使前次转换已使 ETFLG[SOCA] = 1，仍会生成 SOCA 脉冲；脉冲产生后 ETPS[SOCACNT] 自动清零。<br>00：禁用 SOCA 事件计数器，不产生 EPWMxSOCA 脉冲。<br>01：第 1 个事件时产生脉冲，ETPS[SOCACNT] = 0,1。<br>10：第 2 个事件时产生脉冲，ETPS[SOCACNT] = 1,0。<br>11：第 3 个事件时产生脉冲，ETPS[SOCACNT] = 1,1。<br>复位类型：SYSRSn |
| 7-6 | RESERVED | R-0 | 0h | 保留 |
| 5 | SOCPSSEL | R/W | 0h | EPWMxSOC A/B 预分频选择。<br>0：选择 ETPS[SOCACNT/SOCBCNT] 和 [SOCAPRD/SOCBPRD]，决定事件频率（每 0–3 个事件产生一次 SOC 脉冲）。<br>1：选择 ETSOCPS[SOCACNT2/SOCBCNT2] 和 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | [SOCAPRD2/SOCBPRD2]，决定事件频率（每 0–15 个事件产生一次 SOC 脉冲）。<br>复位类型：SYSRSn |
| 4 | INTPSSEL | R/W | 0h | EPWMxINTn 预分频选择。<br>0：选择 ETPS[INTCNT, INTPRD]，决定事件频率（每 0–3 个事件产生一次中断）。<br>1：选择 ETINTPS[INTCNT2, INTPRD2]，决定事件频率（每 0–15 个事件产生一次中断）。<br>复位类型：SYSRSn |
| 3-2 | INTCNT | R | 0h | ePWM 中断事件（EPWMx_INT）计数器。表示 ETSEL[INTSEL] 所选事件已发生的次数。产生中断脉冲时自动清零。若中断禁用（ETSEL[INT] = 0），或中断标志已置位（ETFLG[INT] = 1），计数器在达到周期值 ETPS[INTCNT] = ETPS[INTPRD] 时停止计数。<br>00：未发生事件。<br>01：已发生 1 个事件。<br>10：已发生 2 个事件。<br>11：已发生 3 个事件。<br>复位类型：SYSRSn |
| 1-0 | INTPRD | R/W | 0h | ePWM 中断（EPWMx_INT）周期选择。决定 ETSEL[INTSEL] 所选事件发生多少次后产生中断。必须先使能中断（ETSEL[INT] = 1）。若前次中断已将 ETFLG[INT] 置 1，则必须通过 ETCLR[INT] 清除标志后才能再次产生中断；这允许处理一个中断的同时挂起另一个中断。中断产生后，ETPS[INTCNT] 自动清零。<br>写入与当前计数值相等的 INTPRD 时，若中断已使能且状态标志已清除，将触发中断。写入小于当前计数值的 INTPRD 将导致未定义状态。在写入新的零或非零 INTPRD 值的同时若发生计数事件，计数器递增。<br>00：禁用中断事件计数器，不产生中断，忽略 ETFRC[INT]。<br>01：INTCNT = 01 时产生中断（第 1 个事件）。<br>10：ETPS[INTCNT] = 1,0 时产生中断（第 2 个事件）。<br>11：ETPS[INTCNT] = 1,1 时产生中断（第 3 个事件）。<br>复位类型：SYSRSn |

## 15.16.2.61 ETFLG 寄存器

偏移地址：150h　复位值：0000h

事件触发标志寄存器

表 15-145 ETFLG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | SOCB | SOCA | 保留 | INT |
| R-0-0h | | | | R-0h | R-0h | R-0-0h | R-0h |

表 15-146 ETFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R-0 | 0h | 保留 |
| 3 | SOCB | R | 0h | ePWM ADC 启动转换 A（EPWMxSOCB）锁存状态标志。与 ETFLG[INT] 不同，即使标志已置位，EPWMxSOCB 仍继续输出脉冲。<br>0：未发生事件。<br>1：EPWMxSOCB 已产生启动转换脉冲；即使此标志置位，仍继续产生 EPWMxSOCB 输出。<br>复位类型：SYSRSn |
| 2 | SOCA | R | 0h | ePWM ADC 启动转换 A（EPWMxSOCA）锁存状态标志。与 ETFLG[INT] 不同，即使标志已置位，EPWMxSOCA 仍继续输出脉冲。<br>0：未发生事件。<br>1：EPWMxSOCA 已产生启动转换脉冲；即使此标志置位，仍继续产生 EPWMxSOCA 输出。<br>复位类型：SYSRSn |
| 1 | RESERVED | R-0 | 0h | 保留 |
| 0 | INT | R | 0h | ePWM 中断（EPWMx_INT）锁存状态标志。<br>0：未发生事件。<br>1：已产生 ePWMx 中断（EPWMx_INT）；清除此标志前，不再产生中断。ETFLG[INT] 仍置位时，最多可挂起一个中断；挂起的中断要在清除 ETFLG[INT] 后才会产生。<br>复位类型：SYSRSn |

## 15.16.2.62 ETCLR 寄存器

偏移地址：154h　复位值：0000h

事件触发清除寄存器

表 15-147 ETCLR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | SOCB | SOCA | 保留 | INT |
| R-0-0h | | | | R-0/W1S-0h | R-0/W1S-0h | R-0-0h | R-0/W1S-0h |

表 15-148 ETCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R-0 | 0h | 保留 |
| 3 | SOCB | R-0/W1S | 0h | ePWM ADC 启动转换 A（EPWMxSOCB）标志清除位。<br>0：写 0 无效，读取始终返回 0。<br>1：清除 ETFLG[SOCB] 标志。<br>复位类型：SYSRSn |
| 2 | SOCA | R-0/W1S | 0h | ePWM ADC 启动转换 A（EPWMxSOCA）标志清除位。<br>0：写 0 无效，读取始终返回 0。<br>1：清除 ETFLG[SOCA] 标志。<br>复位类型：SYSRSn |
| 1 | RESERVED | R-0 | 0h | 保留 |
| 0 | INT | R-0/W1S | 0h | ePWM 中断（EPWMx_INT）标志清除位。<br>0：写 0 无效，读取始终返回 0。<br>1：清除 ETFLG[INT] 标志，允许继续产生中断脉冲。<br>复位类型：SYSRSn |

## 15.16.2.63 ETFRC 寄存器

偏移地址：158h　复位值：0000h

事件触发强制寄存器

表 15-149 ETFRC 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | SOCB | SOCA | 保留 | INT |
| R-0-0h | | | | R-0/W1S-0h | R-0/W1S-0h | R-0-0h | R-0/W1S-0h |

表 15-150 ETFRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R-0 | 0h | 保留 |
| 3 | SOCB | R-0/W1S | 0h | SOCB 强制位。仅在 ETSEL 中使能事件时才产生 SOCB 脉冲，但无论是否使能都会置位 ETFLG[SOCB]。<br>0：忽略写 0，读取始终返回 0。<br>1：在 EPWMxSOCB 上产生脉冲并置位 SOCBFLG；此位用于测试。<br>复位类型：SYSRSn |
| 2 | SOCA | R-0/W1S | 0h | SOCA 强制位。仅在 ETSEL 中使能事件时才产生 SOCA 脉冲，但无论是否使能都会置位 ETFLG[SOCA]。<br>0：忽略写 0，读取始终返回 0。<br>1：在 EPWMxSOCA 上产生脉冲并置位 SOCAFLG；此位用于测试。<br>复位类型：SYSRSn |
| 1 | RESERVED | R-0 | 0h | 保留 |
| 0 | INT | R-0/W1S | 0h | INT 强制位。仅在 ETSEL 中使能事件时才会产生中断，但无论是否使能都会置位 INT 标志。<br>0：忽略写 0，读取始终返回 0。<br>1：在 EPWMxINT 上产生中断并置位 INT 标志；此位用于测试。<br>复位类型：SYSRSn |

## 15.16.2.64 ETINTPS 寄存器

偏移地址：15Ch　复位值：0000h

事件触发中断预分频寄存器

表 15-151 ETINTPS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INTCNT2 | | | | INTPRD2 | | | |
| R-0h | | | | R/W-0h | | | |

表 15-152 ETINTPS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R-0 | 0h | 保留 |
| 7-4 | INTCNT2 | R | 0h | EPWMxINT 计数器 2。当 ETPS[INTPSSEL] = 1 时，表示所选事件已发生的次数。<br>0000：无事件。<br>0001：1 个事件。<br>0010：2 个事件。<br>0011：3 个事件。<br>0100：4 个事件。<br>…<br>1111：15 个事件。<br>复位类型：SYSRSn |
| 3-0 | INTPRD2 | R/W | 0h | EPWMxINT 周期 2 选择。当 ETPS[INTPSSEL] = 1 时，选择所选事件发生多少次后产生中断。<br>0000：禁用计数器。<br>0001：INTCNT = 1 时产生中断（第 1 个事件）。<br>0010：INTCNT = 2 时产生中断（第 2 个事件）。<br>0011：INTCNT = 3 时产生中断（第 3 个事件）。<br>0100：INTCNT = 4 时产生中断（第 4 个事件）。<br>…<br>1111：INTCNT = 15 时产生中断（第 15 个事件）。<br>复位类型：SYSRSn |

## 15.16.2.65 ETSOCPS 寄存器

偏移地址：160h　复位值：0000h

事件触发 SOC 预分频寄存器

表 15-153 ETSOCPS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SOCBCNT2 | | | | SOCBPRD2 | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SOCACNT2 | | | | SOCAPRD2 | | | |
| R-0h | | | | R/W-0h | | | |

表 15-154 ETSOCPS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | SOCBCNT2 | R | 0h | EPWMxSOCB 计数器 2。当 ETPS[SOCPSSEL] = 1 时，表示所选事件已发生的次数。<br>0000：无事件。<br>0001：1 个事件。<br>0010：2 个事件。<br>0011：3 个事件。<br>0100：4 个事件。<br>…<br>1111：15 个事件。<br>复位类型：SYSRSn |
| 11-8 | SOCBPRD2 | R/W | 0h | EPWMxSOCB 周期 2 选择。当 ETPS[SOCPSSEL] = 1 时，选择所选事件发生多少次后产生SOCB 脉冲。<br>0000：禁用计数器。<br>0001：SOCBCNT2 = 1 时产生SOC 脉冲（第 1 个事件）。<br>0010：SOCBCNT2 = 2 时产生SOC 脉冲（第 2 个事件）。<br>0011：SOCBCNT2 = 3 时产生SOC 脉冲（第 3 个事件）。<br>0100：SOCBCNT2 = 4 时产生SOC 脉冲（第 4 个事件）。<br>…<br>1111：SOCBCNT2 = 15 时产生SOC 脉冲（第 15 个事件）。<br>复位类型：SYSRSn |
| 7-4 | SOCACNT2 | R | 0h | EPWMxSOCA 计数器 2。当 ETPS[SOCPSSEL] = 1 时，表示所选事件已发生的次数。<br>0000：无事件。<br>0001：1 个事件。<br>0010：2 个事件。<br>0011：3 个事件。<br>0100：4 个事件。<br>… |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1111：15 个事件。<br>复位类型：SYSRSn |
| 3-0 | SOCAPRD2 | R/W | 0h | EPWMxSOCA 周期 2 选择。当 ETPS[SOCPSSEL] = 1 时，选择所选事件发生多少次后产生SOCA 脉冲。<br>0000：禁用计数器。<br>0001：SOCACNT2 = 1 时产生SOC 脉冲（第 1 个事件）。<br>0010：SOCACNT2 = 2 时产生SOC 脉冲（第 2 个事件）。<br>0011：SOCACNT2 = 3 时产生SOC 脉冲（第 3 个事件）。<br>0100：SOCACNT2 = 4 时产生SOC 脉冲（第 4 个事件）。<br>…<br>1111：SOCACNT2 = 15 时产生SOC 脉冲（第 15 个事件）。<br>复位类型：SYSRSn |

## 15.16.2.66 ETCNTINITCTL 寄存器

偏移地址：164h　复位值：0000h

事件触发计数器初始化控制寄存器

表 15-155 ETCNTINITCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SOCBINITEN | SOCAINITEN | INTINITEN | SOCBINITFRC | SOCAINITFRC | INTINITFRC | 保留 | |
| R/W-0h | R/W-0h | R/W-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |

表 15-156 ETCNTINITCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 15 | SOCBINITEN | R/W | 0h | EPWMxSOCB 计数器 2 初始化使能。<br>0：无效。<br>1：使能在 SYNC 事件或软件强制时，以 ETCNTINIT[SOCBINIT] 的内容初始化 EPWMxSOCB 计数器。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 14 | SOCAINITEN | R/W | 0h | EPWMxSOCA 计数器 2 初始化使能。<br>0：无效。<br>1：使能在 SYNC 事件或软件强制时，以 ETCNTINIT[SOCAINIT] 的内容初始化 EPWMxSOCA 计数器。<br>复位类型：SYSRSn |
| 13 | INTINITEN | R/W | 0h | EPWMxINT 计数器 2 初始化使能。<br>0：无效。<br>1：使能在 SYNC 事件或软件强制时，以 ETCNTINIT[INTINIT] 的内容初始化 EPWMxINT 计数器 2。<br>复位类型：SYSRSn |
| 12 | SOCBINITFRC | R-0/W1S | 0h | EPWMxSOCB 计数器 2 强制初始化。<br>0：无效。<br>1：强制以 ETCNTINIT[SOCBINIT] 的内容初始化 ET EPWMxSOCB 计数器。<br>复位类型：SYSRSn |
| 11 | SOCAINITFRC | R-0/W1S | 0h | EPWMxSOCA 计数器 2 强制初始化。<br>0：无效。<br>1：强制以 ETCNTINIT[SOCAINIT] 的内容初始化 ET EPWMxSOCA 计数器。<br>复位类型：SYSRSn |
| 10 | INTINITFRC | R-0/W1S | 0h | EPWMxINT 计数器 2 强制初始化。<br>0：无效。<br>1：强制以 ETCNTINIT[INTINIT] 的内容初始化 ET EPWMxINT 计数器。<br>复位类型：SYSRSn |
| 9-0 | RESERVED | R-0 | 0h | 保留 |

## 15.16.2.67 ETCNTINIT 寄存器

偏移地址：168h　复位值：0000h

事件触发计数器初始化寄存器

表 15-157 ETCNTINIT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | SOCBINIT | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SOCAINIT | | | | INTINIT | | | |
| R/W-0h | | | | R/W-0h | | | |

表 15-158 ETCNTINIT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留 |
| 11-8 | SOCBINIT | R/W | 0h | EPWMxSOCB 计数器 2 初始化位。ePWM SYNC 事件或软件强制发生时，以此寄存器内容初始化 ET EPWMxSOCB 计数器。<br>复位类型：SYSRSn |
| 7-4 | SOCAINIT | R/W | 0h | EPWMxSOCA 计数器 2 初始化位。ePWM SYNC 事件或软件强制发生时，以此寄存器内容初始化 ET EPWMxSOCA 计数器。<br>复位类型：SYSRSn |
| 3-0 | INTINIT | R/W | 0h | EPWMxINT 计数器 2 初始化位。ePWM SYNC 事件或软件强制发生时，以此寄存器内容初始化 ET EPWMxINT 计数器。<br>复位类型：SYSRSn |

## 15.16.2.68 DCTRIPSEL 寄存器

偏移地址：180h　复位值：0000h

数字比较跳闸选择寄存器

表 15-159 DCTRIPSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DCBLCOMPSEL | | | | DCBHCOMPSEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCALCOMPSEL | | | | DCAHCOMPSEL | | | |
| R/W-0h | | | | R/W-0h | | | |

表 15-160 DCTRIPSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | DCBLCOMPSEL | R/W | 0h | 数字比较 B 低输入选择位。<br>0000：TRIPIN1<br>0001：TRIPIN2<br>0010：TRIPIN3<br>0011：TRIPIN4<br>…<br>1011：TRIPIN12<br>1100：保留<br>1101：TRIPIN14<br>1110：TRIPIN15<br>1111：跳闸组合输入（将 DCBLTRIPSEL 所选 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 所有跳闸输入进行逻辑或）。<br>复位类型：SYSRSn |
| 11-8 | DCBHCOMPSEL | R/W | 0h | 数字比较 B 高输入选择位。<br>0000：TRIPIN1<br>0001：TRIPIN2<br>0010：TRIPIN3<br>0011：TRIPIN4<br>…<br>1011：TRIPIN12<br>1100：保留<br>1101：TRIPIN14<br>1110：TRIPIN15<br>1111：跳闸组合输入（将 DCBHTRIPSEL 所选所有跳闸输入进行逻辑或）。<br>复位类型：SYSRSn |
| 7-4 | DCALCOMPSEL | R/W | 0h | 数字比较 A 低输入选择位。<br>0000：TRIPIN1<br>0001：TRIPIN2<br>0010：TRIPIN3<br>0011：TRIPIN4<br>…<br>1011：TRIPIN12<br>1100：保留<br>1101：TRIPIN14<br>1110：TRIPIN15<br>1111：跳闸组合输入（将 DCALTRIPSEL 所选所有跳闸输入进行逻辑或）。<br>复位类型：SYSRSn |
| 3-0 | DCAHCOMPSEL | R/W | 0h | 数字比较 A 高输入选择位。<br>0000：TRIPIN1<br>0001：TRIPIN2<br>0010：TRIPIN3<br>0011：TRIPIN4<br>…<br>1011：TRIPIN12<br>1100：保留<br>1101：TRIPIN14<br>1110：TRIPIN15<br>1111：跳闸组合输入（将 DCAHTRIPSEL 所选所有跳闸输入进行逻辑或）。<br>复位类型：SYSRSn |

## 15.16.2.69 DCACTL 寄存器

偏移地址：186h　复位值：0000h

数字比较 A 控制寄存器

表 15-161 DCACTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | 保留 | | 保留 | 保留 | | EVT2FRCSYNCSEL | EVT2SRCSEL |
| R-0-0h | R-0-0h | | R-0-0h | R-0-0h | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | | 保留 | EVT1SYNCE | EVT1SOCE | EVT1FRCSYNCSEL | EVT1SRCSEL |
| R-0h | R-0h | | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 15-162 DCACTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R-0 | 0h | 保留 |
| 14-13 | RESERVED | R-0 | 0h | 保留 |
| 12 | RESERVED | R-0 | 0h | 保留 |
| 11-10 | RESERVED | R-0 | 0h | 保留 |
| 9 | EVT2FRCSYNCSEL | R/W | 0h | DCAEVT2 强制/同步选择。<br>0：源与 EPWMCLK 同步。<br>1：源异步直接通过。<br>复位类型：SYSRSn |
| 8 | EVT2SRCSEL | R/W | 0h | DCAEVT2 源选择。<br>0：源为 DCAL 信号。<br>1：源为 DCEVTFILT 信号。<br>复位类型：SYSRSn |
| 7 | RESERVED | R | 0h | 保留 |
| 6-5 | RESERVED | R | 0h | 保留 |
| 4 | RESERVED | R | 0h | 保留 |
| 3 | EVT1SYNCE | R/W | 0h | 使能 DCAEVT1.sync 作为同步事件源。<br>0：禁用。<br>1：使能。<br>复位类型：SYSRSn |
| 2 | EVT1SOCE | R/W | 0h | 使能 DCAEVT1.soc 作为 ADC SOC 触发源。<br>0：禁用。<br>1：使能。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | EVT1FRCSYNCSEL | R/W | 0h | DCAEVT1 强制/同步选择。<br>0：源与 EPWMCLK 同步。<br>1：源异步直接通过。<br>复位类型：SYSRSn |
| 0 | EVT1SRCSEL | R/W | 0h | DCAEVT1 源选择。<br>0：源为 DCAH 信号。<br>1：源为 DCEVTFILT 信号。<br>复位类型：SYSRSn |

## 15.16.2.70 DCBCTL 寄存器

偏移地址：188h　复位值：0000h

数字比较 B 控制寄存器

表 15-163 DCBCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | 保留 | | 保留 | 保留 | | EVT2FRCSYNCSEL | EVT2SRCSEL |
| R-0h | R-0h | | R-0h | R-0-0h | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | | 保留 | EVT1SYNCE | EVT1SOCE | EVT1FRCSYNCSEL | EVT1SRCSEL |
| R-0h | R-0h | | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 15-164 DCBCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留 |
| 14-13 | RESERVED | R | 0h | 保留 |
| 12 | RESERVED | R | 0h | 保留 |
| 11-10 | RESERVED | R-0 | 0h | 保留 |
| 9 | EVT2FRCSYNCSEL | R/W | 0h | DCBEVT2 强制/同步选择。<br>0：源与 EPWMCLK 同步。<br>1：源异步直接通过。<br>复位类型：SYSRSn |
| 8 | EVT2SRCSEL | R/W | 0h | DCBEVT2 源选择。<br>0：源为 DCBL 信号。<br>1：源为 DCEVTFILT 信号。<br>复位类型：SYSRSn |
| 7 | RESERVED | R | 0h | 保留 |
| 6-5 | RESERVED | R | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 4 | RESERVED | R | 0h | 保留 |
| 3 | EVT1SYNCE | R/W | 0h | 使能 DCBEVT1.sync 作为同步事件源。<br>0：禁用。<br>1：使能。<br>复位类型：SYSRSn |
| 2 | EVT1SOCE | R/W | 0h | 使能 DCBEVT1.soc 作为 ADC SOC 触发源。<br>0：禁用。<br>1：使能。<br>复位类型：SYSRSn |
| 1 | EVT1FRCSYNCSEL | R/W | 0h | DCBEVT1 强制/同步选择。<br>0：源与 EPWMCLK 同步。<br>1：源异步直接通过。<br>复位类型：SYSRSn |
| 0 | EVT1SRCSEL | R/W | 0h | DCBEVT1 源选择。<br>0：源为 DCBH 信号。<br>1：源为 DCEVTFILT 信号。<br>复位类型：SYSRSn |

## 15.16.2.71 DCFCTL 寄存器

偏移地址：18Eh　复位值：0000h

数字比较滤波器控制寄存器

表 15-165 DCFCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| EDGESTATUS | | | EDGECOUNT | | | EDGEMODE | |
| R-0h | | | R/W-0h | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | EDGEFILTSEL | PULSESEL | | BLANKINV | BLANKE | SRCSEL | |
| R-0-0h | R/W-0h | R/W-0h | | R/W-0h | R/W-0h | R/W-0h | |

表 15-166 DCFCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-13 | EDGESTATUS | R | 0h | 边沿状态。表示当前已捕获的边沿总数。当该值与 EDGECOUNT 相等时，状态位清零，并产生宽度为一个 TBCLK 的脉冲，可从 DCEVTFILT 信号输出。向 EDGECOUNT 写入 000 可复位边沿计数器。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 12-10 | EDGECOUNT | R/W | 0h | 边沿计数。选择在 DCEVTFILT 上产生宽度为一个 TBCLK 的脉冲前需要计数的边沿数。<br>000：不计边沿，将当前 EDGESTATUS 复位为 0,0,0。<br>001：1 个边沿。<br>010：2 个边沿。<br>011：3 个边沿。<br>100：4 个边沿。<br>101：5 个边沿。<br>110：6 个边沿。<br>111：7 个边沿。<br>复位类型：SYSRSn |
| 9-8 | EDGEMODE | R/W | 0h | 边沿模式选择。<br>00：低到高边沿。<br>01：高到低边沿。<br>10：双边沿。<br>11：保留。<br>复位类型：SYSRSn |
| 7 | RESERVED | R-0 | 0h | 保留 |
| 6 | EDGEFILTSEL | R/W | 0h | 边沿滤波器选择。<br>0：不选择边沿滤波器。<br>1：选择边沿滤波器。<br>复位类型：SYSRSn |
| 5-4 | PULSESEL | R/W | 0h | 消隐与捕获对齐的脉冲选择。<br>00：时基计数器等于周期（TBCTR = TBPRD）。<br>01：时基计数器等于零（TBCTR = 0x00）。<br>10：时基计数器等于零（TBCTR = 0x00）或周期（TBCTR = TBPRD）。<br>11：保留。<br>复位类型：SYSRSn |
| 3 | BLANKINV | R/W | 0h | 消隐窗口反相。<br>0：消隐窗口不反相。<br>1：消隐窗口反相。<br>复位类型：SYSRSn |
| 2 | BLANKE | R/W | 0h | 消隐窗口使能 / 禁用。<br>0：禁用消隐窗口。<br>1：使能消隐窗口。<br>复位类型：SYSRSn |
| 1-0 | SRCSEL | R/W | 0h | 滤波模块信号源选择。<br>00：DCAEVT1 信号。<br>01：DCAEVT2 信号。<br>10：DCBEVT1 信号。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 11：DCBEVT2 信号。<br>复位类型：SYSRSn |

## 15.16.2.72 DCCAPCTL 寄存器

偏移地址：190h　复位值：0000h

数字比较捕获控制寄存器

表 15-167 DCCAPCTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CAPMODE | CAPCLR | CAPSTS | 保留 | | | | |
| R/W-0h | R-0/W1S-0h | R-0h | R-0-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | SHDWMODE | CAPE |
| R-0-0h | | | | | | R/W-0h | R/W-0h |

表 15-168 DCCAPCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | CAPMODE | R/W | 0h | 计数器捕获模式。<br>0：发生 DCEVTFILT 且捕获已使能时，将当前 TBCNT 值捕获到活动寄存器。相应跳闸事件发生后，忽略后续跳闸（捕获）事件，直到 DCFCTL[PULSESEL] 所选的下一个 PRD_eq 或 CNT_zero 事件重新触发捕获机制。若通过 DCCAPCTL[SHDWMODE] 使能活动模式，CPU 读取返回活动寄存器值。若使能影子模式，则在 DCFCTL[PULSESEL] 所选的 PRD_eq 或 CNT_zero 事件时，将活动寄存器复制到影子寄存器，CPU 读取返回影子寄存器值。<br>1：发生 DCEVTFILT 且捕获已使能时，将当前 TBCNT 值捕获到活动寄存器。相应跳闸事件发生后置位 CAPSTS，忽略后续跳闸（捕获）事件，直到清除此位。向 DCCAPCTL[CAPCLR] 写入可清除 CAPSTS，并重新触发捕获机制。若通过 DCCAPCTL[SHDWMODE] 使能活动模式，CPU 读取返回活动寄存器值。若使能影子模式，则在 DCFCTL[PULSESEL] 所选的 PRD_eq 或 CNT_zero 事件时，将活动寄存器复制到影子寄存器，CPU 读取返回影子寄存器值。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 14 | CAPCLR | R-0/W1S | 0h | DC 捕获锁存状态清除标志。<br>0：写 0 无效。<br>1：写 1 清除 CAPSTS 置位条件。<br>复位类型：SYSRSn |
| 13 | CAPSTS | R | 0h | 捕获事件锁存状态标志。<br>0：未发生 DC 捕获事件。<br>1：已发生 DC 捕获事件。<br>复位类型：SYSRSn |
| 12-2 | RESERVED | R-0 | 0h | 保留 |
| 1 | SHDWMODE | R/W | 0h | TBCTR 计数器捕获影子模式选择。<br>0：使能影子模式；在 DCFCTL[PULSESEL] 所定义的 TBCTR = TBPRD 或 TBCTR = zero 事件时，将 DCCAP 活动寄存器复制到影子寄存器。CPU 读取 DCCAP 返回影子寄存器内容。<br>1：活动模式，禁用影子寄存器；CPU 读取 DCCAP 始终返回活动寄存器内容。<br>复位类型：SYSRSn |
| 0 | CAPE | R/W | 0h | TBCTR 计数器捕获使能 / 禁用。<br>0：禁用时基计数器捕获。<br>1：使能时基计数器捕获。<br>复位类型：SYSRSn |

## 15.16.2.73 DCFOFFSET 寄存器

偏移地址：192h　复位值：0000h

数字比较滤波器偏移寄存器

表 15-169 DCFOFFSET 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DCFOFFSET | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCFOFFSET | | | | | | | |
| R/W-0h | | | | | | | |

表 15-170 DCFOFFSET 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | DCFOFFSET | R/W | 0h | 消隐窗口偏移。此 16 位值指定从消隐窗口参考点到窗口开始生效的 TBCLK 周期数。参考点为 DCFCTL[PULSESEL] 定义的周期或零事件。此偏移寄存器具有影子寄存器，活动寄存器在指定 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 参考点装载；同时初始化偏移计数器并开始向下计数。计数结束后应用消隐窗口；若消隐窗口当前已有效，则重新启动消隐窗口计数器。<br>复位类型：SYSRSn |

## 15.16.2.74 DCFOFFSETCNT 寄存器

偏移地址：194h　复位值：0000h

数字比较滤波器偏移计数器寄存器

表 15-171 DCFOFFSETCNT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DCFOFFSETCNT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCFOFFSETCNT | | | | | | | |
| R-0h | | | | | | | |

表 15-172 DCFOFFSETCNT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | DCFOFFSETCNT | R | 0h | 消隐偏移计数器。此只读 16 位值表示偏移计数器当前值。计数器向下计数至零后停止，直到 DCFCTL[PULSESEL] 定义的下一个周期或零事件重新装载。偏移计数器不受 free/soft 仿真位影响，即使器件因仿真停止而暂停，也会继续向下计数。<br>复位类型：SYSRSn |

## 15.16.2.75 DCFWINDOW 寄存器

偏移地址：196h　复位值：0000h

数字比较滤波器窗口寄存器

表 15-173 DCFWINDOW 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DCFWINDOW | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCFWINDOW | | | | | | | |
| R/W-0h | | | | | | | |

表 15-174 DCFWINDOW 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | DCFWINDOW | R/W | 0h | 消隐窗口宽度。<br>00h：不产生消隐窗口。<br>01-FFFFh：以 TBCLK 周期为单位指定消隐窗口宽度。偏移计数器计数结束后，消隐窗口开始，窗口计数器装载并开始向下计数。若消隐窗口当前有效时偏移计数器计数结束，消隐窗口计数器不会重新启动，窗口会被提前截断；应注意避免此情况。消隐窗口可跨越 PWM 周期边界。<br>复位类型：SYSRSn |

## 15.16.2.76 DCFWINDOWCNT 寄存器

偏移地址：198h　复位值：0000h

数字比较滤波器窗口计数器寄存器

表 15-175 DCFWINDOWCNT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DCFWINDOWCNT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCFWINDOWCNT | | | | | | | |
| R-0h | | | | | | | |

表 15-176 DCFWINDOWCNT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | DCFWINDOWCNT | R | 0h | 消隐窗口计数器。此只读 16 位值表示窗口计数器当前值。计数器向下计数至零后停止，直到偏移计数器再次达到零时重新装载。<br>复位类型：SYSRSn |

## 15.16.2.77 DCCAP 寄存器

偏移地址：19Eh　复位值：0000h

数字比较计数器捕获寄存器

表 15-177 DCCAP 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| DCCAP | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DCCAP | | | | | | | |
| R-0h | | | | | | | |

表 15-178 DCCAP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | DCCAP | R | 0h | 数字比较时基计数器捕获。将 DCCAPCLT[CAPE] 置 1 可使能时基计数器捕获。使能后，此寄存器反映滤波事件 DCEVTFLT 的低到高边沿到来时的 TBCTR 值。忽略后续捕获事件，直到 DCFCTL[PULSESEL] 所选的下一个周期或零事件。DCCAPCTL[SHDWMODE] 控制 DCCAP 影子模式，默认使能。<br>DCCAPCTL[SHDWMODE] = 0：使能影子模式；在 DCFCTL[PULSESEL] 定义的 TBCTR = TBPRD 或 TBCTR = zero 事件时，将活动寄存器复制到影子寄存器。CPU 读取返回影子值。<br>DCCAPCTL[SHDWMODE] = 1：禁用影子模式；CPU 读取返回活动值。活动寄存器和影子寄存器共享同一存储器映射地址。<br>复位类型：SYSRSn |

## 15.16.2.78 DCAHTRIPSEL 寄存器

偏移地址：1A4h　复位值：0000h

数字比较 AH 跳闸选择

表 15-179 DCAHTRIPSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | TRIPINPUT15 | TRIPINPUT14 | 保留 | TRIPINPUT12 | TRIPINPUT11 | TRIPINPUT10 | TRIPINPUT9 |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TRIPINPUT8 | TRIPINPUT7 | TRIPINPUT6 | TRIPINPUT5 | TRIPINPUT4 | TRIPINPUT3 | TRIPINPUT2 | TRIPINPUT1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 15-180 DCAHTRIPSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留 |
| 14 | TRIPINPUT15 | R/W | 0h | TRIP 输入 15。<br>0：不将跳闸输入 15 选为逻辑或组合输入。<br>1：将跳闸输入 15 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 13 | TRIPINPUT14 | R/W | 0h | TRIP 输入 14。<br>0：不将跳闸输入 14 选为逻辑或组合输入。<br>1：将跳闸输入 14 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 12 | RESERVED | R/W | 0h | 保留 |
| 11 | TRIPINPUT12 | R/W | 0h | TRIP 输入 12。<br>0：不将跳闸输入 12 选为逻辑或组合输入。<br>1：将跳闸输入 12 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 10 | TRIPINPUT11 | R/W | 0h | TRIP 输入 11。<br>0：不将跳闸输入 11 选为逻辑或组合输入。<br>1：将跳闸输入 11 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 9 | TRIPINPUT10 | R/W | 0h | TRIP 输入 10。<br>0：不将跳闸输入 10 选为逻辑或组合输入。<br>1：将跳闸输入 10 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 8 | TRIPINPUT9 | R/W | 0h | TRIP 输入 9。<br>0：不将跳闸输入 9 选为逻辑或组合输入。<br>1：将跳闸输入 9 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 7 | TRIPINPUT8 | R/W | 0h | TRIP 输入 8。<br>0：不将跳闸输入 8 选为逻辑或组合输入。<br>1：将跳闸输入 8 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 6 | TRIPINPUT7 | R/W | 0h | TRIP 输入 7。<br>0：不将跳闸输入 7 选为逻辑或组合输入。<br>1：将跳闸输入 7 选为 DCAH 多路复用器的逻 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 辑或组合输入。<br>复位类型：SYSRSn |
| 5 | TRIPINPUT6 | R/W | 0h | TRIP 输入 6。<br>0：不将跳闸输入 6 选为逻辑或组合输入。<br>1：将跳闸输入 6 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 4 | TRIPINPUT5 | R/W | 0h | TRIP 输入 5。<br>0：不将跳闸输入 5 选为逻辑或组合输入。<br>1：将跳闸输入 5 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 3 | TRIPINPUT4 | R/W | 0h | TRIP 输入 4。<br>0：不将跳闸输入 4 选为逻辑或组合输入。<br>1：将跳闸输入 4 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 2 | TRIPINPUT3 | R/W | 0h | TRIP 输入 3。<br>0：不将跳闸输入 3 选为逻辑或组合输入。<br>1：将跳闸输入 3 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 1 | TRIPINPUT2 | R/W | 0h | TRIP 输入 2。<br>0：不将跳闸输入 2 选为逻辑或组合输入。<br>1：将跳闸输入 2 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 0 | TRIPINPUT1 | R/W | 0h | TRIP 输入 1。<br>0：不将跳闸输入 1 选为逻辑或组合输入。<br>1：将跳闸输入 1 选为 DCAH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |

## 15.16.2.79 DCALTRIPSEL 寄存器

偏移地址：1A6h　复位值：0000h

数字比较 AL 跳闸选择

表 15-181 DCALTRIPSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | TRIPINPUT15 | TRIPINPUT14 | 保留 | TRIPINPUT12 | TRIPINPUT11 | TRIPINPUT10 | TRIPINPUT9 |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TRIPINPUT8 | TRIPINPUT7 | TRIPINPUT6 | TRIPINPUT5 | TRIPINPUT4 | TRIPINPUT3 | TRIPINPUT2 | TRIPINPUT1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 15-182 DCALTRIPSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留 |
| 14 | TRIPINPUT15 | R/W | 0h | TRIP 输入 15。<br>0：不将跳闸输入 15 选为逻辑或组合输入。<br>1：将跳闸输入 15 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 13 | TRIPINPUT14 | R/W | 0h | TRIP 输入 14。<br>0：不将跳闸输入 14 选为逻辑或组合输入。<br>1：将跳闸输入 14 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 12 | RESERVED | R/W | 0h | 保留 |
| 11 | TRIPINPUT12 | R/W | 0h | TRIP 输入 12。<br>0：不将跳闸输入 12 选为逻辑或组合输入。<br>1：将跳闸输入 12 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 10 | TRIPINPUT11 | R/W | 0h | TRIP 输入 11。<br>0：不将跳闸输入 11 选为逻辑或组合输入。<br>1：将跳闸输入 11 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 9 | TRIPINPUT10 | R/W | 0h | TRIP 输入 10。<br>0：不将跳闸输入 10 选为逻辑或组合输入。<br>1：将跳闸输入 10 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 8 | TRIPINPUT9 | R/W | 0h | TRIP 输入 9。<br>0：不将跳闸输入 9 选为逻辑或组合输入。<br>1：将跳闸输入 9 选为 DCAL 多路复用器的逻 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 辑或组合输入。<br>复位类型：SYSRSn |
| 7 | TRIPINPUT8 | R/W | 0h | TRIP 输入 8。<br>0：不将跳闸输入 8 选为逻辑或组合输入。<br>1：将跳闸输入 8 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 6 | TRIPINPUT7 | R/W | 0h | TRIP 输入 7。<br>0：不将跳闸输入 7 选为逻辑或组合输入。<br>1：将跳闸输入 7 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 5 | TRIPINPUT6 | R/W | 0h | TRIP 输入 6。<br>0：不将跳闸输入 6 选为逻辑或组合输入。<br>1：将跳闸输入 6 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 4 | TRIPINPUT5 | R/W | 0h | TRIP 输入 5。<br>0：不将跳闸输入 5 选为逻辑或组合输入。<br>1：将跳闸输入 5 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 3 | TRIPINPUT4 | R/W | 0h | TRIP 输入 4。<br>0：不将跳闸输入 4 选为逻辑或组合输入。<br>1：将跳闸输入 4 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 2 | TRIPINPUT3 | R/W | 0h | TRIP 输入 3。<br>0：不将跳闸输入 3 选为逻辑或组合输入。<br>1：将跳闸输入 3 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 1 | TRIPINPUT2 | R/W | 0h | TRIP 输入 2。<br>0：不将跳闸输入 2 选为逻辑或组合输入。<br>1：将跳闸输入 2 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 0 | TRIPINPUT1 | R/W | 0h | TRIP 输入 1。<br>0：不将跳闸输入 1 选为逻辑或组合输入。<br>1：将跳闸输入 1 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |

## 15.16.2.80 DCBHTRIPSEL 寄存器

偏移地址：1A8h　复位值：0000h

数字比较 BH 跳闸选择

表 15-183 DCBHTRIPSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | TRIPINPUT15 | TRIPINPUT14 | 保留 | TRIPINPUT12 | TRIPINPUT11 | TRIPINPUT10 | TRIPINPUT9 |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TRIPINPUT8 | TRIPINPUT7 | TRIPINPUT6 | TRIPINPUT5 | TRIPINPUT4 | TRIPINPUT3 | TRIPINPUT2 | TRIPINPUT1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 15-184 DCBHTRIPSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留 |
| 14 | TRIPINPUT15 | R/W | 0h | TRIP 输入 15。<br>0：不将跳闸输入 15 选为逻辑或组合输入。<br>1：将跳闸输入 15 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 13 | TRIPINPUT14 | R/W | 0h | TRIP 输入 14。<br>0：不将跳闸输入 14 选为逻辑或组合输入。<br>1：将跳闸输入 14 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 12 | RESERVED | R/W | 0h | 保留 |
| 11 | TRIPINPUT12 | R/W | 0h | TRIP 输入 12。<br>0：不将跳闸输入 12 选为逻辑或组合输入。<br>1：将跳闸输入 12 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 10 | TRIPINPUT11 | R/W | 0h | TRIP 输入 11。<br>0：不将跳闸输入 11 选为逻辑或组合输入。<br>1：将跳闸输入 11 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 9 | TRIPINPUT10 | R/W | 0h | TRIP 输入 10。<br>0：不将跳闸输入 10 选为逻辑或组合输入。<br>1：将跳闸输入 10 选为 DCBH 多路复用器的逻 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 辑或组合输入。<br>复位类型：SYSRSn |
| 8 | TRIPINPUT9 | R/W | 0h | TRIP 输入 9。<br>0：不将跳闸输入 9 选为逻辑或组合输入。<br>1：将跳闸输入 9 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 7 | TRIPINPUT8 | R/W | 0h | TRIP 输入 8。<br>0：不将跳闸输入 8 选为逻辑或组合输入。<br>1：将跳闸输入 8 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 6 | TRIPINPUT7 | R/W | 0h | TRIP 输入 7。<br>0：不将跳闸输入 7 选为逻辑或组合输入。<br>1：将跳闸输入 7 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 5 | TRIPINPUT6 | R/W | 0h | TRIP 输入 6。<br>0：不将跳闸输入 6 选为逻辑或组合输入。<br>1：将跳闸输入 6 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 4 | TRIPINPUT5 | R/W | 0h | TRIP 输入 5。<br>0：不将跳闸输入 5 选为逻辑或组合输入。<br>1：将跳闸输入 5 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 3 | TRIPINPUT4 | R/W | 0h | TRIP 输入 4。<br>0：不将跳闸输入 4 选为逻辑或组合输入。<br>1：将跳闸输入 4 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 2 | TRIPINPUT3 | R/W | 0h | TRIP 输入 3。<br>0：不将跳闸输入 3 选为逻辑或组合输入。<br>1：将跳闸输入 3 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 1 | TRIPINPUT2 | R/W | 0h | TRIP 输入 2。<br>0：不将跳闸输入 2 选为逻辑或组合输入。<br>1：将跳闸输入 2 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 0 | TRIPINPUT1 | R/W | 0h | TRIP 输入 1。<br>0：不将跳闸输入 1 选为逻辑或组合输入。<br>1：将跳闸输入 1 选为 DCBH 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |

## 15.16.2.81 DCBLTRIPSEL 寄存器

偏移地址：1AAh　复位值：0000h

数字比较 BL 跳闸选择

表 15-185 DCBLTRIPSEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | TRIPINPUT15 | TRIPINPUT14 | 保留 | TRIPINPUT12 | TRIPINPUT11 | TRIPINPUT10 | TRIPINPUT9 |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TRIPINPUT8 | TRIPINPUT7 | TRIPINPUT6 | TRIPINPUT5 | TRIPINPUT4 | TRIPINPUT3 | TRIPINPUT2 | TRIPINPUT1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 15-186 DCBLTRIPSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留 |
| 14 | TRIPINPUT15 | R/W | 0h | TRIP 输入 15。<br>0：不将跳闸输入 15 选为逻辑或组合输入。<br>1：将跳闸输入 15 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 13 | TRIPINPUT14 | R/W | 0h | TRIP 输入 14。<br>0：不将跳闸输入 14 选为逻辑或组合输入。<br>1：将跳闸输入 14 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 12 | RESERVED | R/W | 0h | 保留 |
| 11 | TRIPINPUT12 | R/W | 0h | TRIP 输入 12。<br>0：不将跳闸输入 12 选为逻辑或组合输入。<br>1：将跳闸输入 12 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 10 | TRIPINPUT11 | R/W | 0h | TRIP 输入 11。<br>0：不将跳闸输入 11 选为逻辑或组合输入。<br>1：将跳闸输入 11 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 9 | TRIPINPUT10 | R/W | 0h | TRIP 输入 10。<br>0：不将跳闸输入 10 选为逻辑或组合输入。<br>1：将跳闸输入 10 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 8 | TRIPINPUT9 | R/W | 0h | TRIP 输入 9。<br>0：不将跳闸输入 9 选为逻辑或组合输入。<br>1：将跳闸输入 9 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 7 | TRIPINPUT8 | R/W | 0h | TRIP 输入 8。<br>0：不将跳闸输入 8 选为逻辑或组合输入。<br>1：将跳闸输入 8 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 6 | TRIPINPUT7 | R/W | 0h | TRIP 输入 7。<br>0：不将跳闸输入 7 选为逻辑或组合输入。<br>1：将跳闸输入 7 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 5 | TRIPINPUT6 | R/W | 0h | TRIP 输入 6。<br>0：不将跳闸输入 6 选为逻辑或组合输入。<br>1：将跳闸输入 6 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 4 | TRIPINPUT5 | R/W | 0h | TRIP 输入 5。<br>0：不将跳闸输入 5 选为逻辑或组合输入。<br>1：将跳闸输入 5 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 3 | TRIPINPUT4 | R/W | 0h | TRIP 输入 4。<br>0：不将跳闸输入 4 选为逻辑或组合输入。<br>1：将跳闸输入 4 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 2 | TRIPINPUT3 | R/W | 0h | TRIP 输入 3。<br>0：不将跳闸输入 3 选为逻辑或组合输入。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：将跳闸输入 3 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 1 | TRIPINPUT2 | R/W | 0h | TRIP 输入 2。<br>0：不将跳闸输入 2 选为逻辑或组合输入。<br>1：将跳闸输入 2 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |
| 0 | TRIPINPUT1 | R/W | 0h | TRIP 输入 1。<br>0：不将跳闸输入 1 选为逻辑或组合输入。<br>1：将跳闸输入 1 选为 DCAL 多路复用器的逻辑或组合输入。<br>复位类型：SYSRSn |

## 15.16.2.82 EPWMLOCK 寄存器

偏移地址：1F4h　复位值：00000000h

EPWM 锁定寄存器

表 15-187 EPWMLOCK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| KEY | | | | | | | |
| R-0/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | DCLOCK | TZCLRLOCK | TZCFGLOCK | GLLOCK | HRLOCK |
| R-0h | | | R/WOnce-0h | R/WOnce-0h | R/WOnce-0h | R/WOnce-0h | R/WOnce-0h |

表 15-188 EPWMLOCK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-16 | KEY | R-0/W | 0h | 只有将此字段写为 0xa5a5 时，对此寄存器的写入才会成功。<br>注意：[1] 由于 KEY 的要求，只有 32 位写入才能成功（且 KEY 必须匹配）。对寄存器高半 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 部或低半部的 16 位写入均被忽略。<br>复位类型：SYSRSn |
| 15-5 | RESERVED | R | 0h | 保留 |
| 4 | DCLOCK | R/WOnce | 0h | 0：偏移 0xC0 至 0xD9 的数字比较寄存器受 EALLOW 保护。<br>1：这些寄存器被锁定，不可写。<br>复位类型：SYSRSn |
| 3 | TZCLRLOCK | R/WOnce | 0h | 0：偏移 0x97 至 0x9B 的跳闸区寄存器受 EALLOW 保护。<br>1：这些寄存器被锁定，不可写。<br>复位类型：SYSRSn |
| 2 | TZCFGLOCK | R/WOnce | 0h | 0：偏移 0x80 至 0x8D 的跳闸区寄存器，以及偏移 0x9D 的 TZTRIPOUTSEL 受 EALLOW 保护。<br>1：这些寄存器被锁定，不可写。<br>复位类型：SYSRSn |
| 1 | GLLOCK | R/WOnce | 0h | 0：偏移 0x34 至 0x35 的全局装载寄存器受 EALLOW 保护。<br>1：这些寄存器被锁定，不可写。<br>复位类型：SYSRSn |
| 0 | HRLOCK | R/WOnce | 0h | 0：偏移 0x20 至 0x2D 的 HRPWM 寄存器受 EALLOW 保护。<br>1：偏移 0x20 和 0x2D 的 HRPWM 寄存器被锁定，不可写。<br>复位类型：SYSRSn |

### 15.16.2.83 HWVDELVAL 寄存器

偏移地址：1FAh　复位值：0000h

硬件谷底模式延迟寄存器

表 15-189 HWVDELVAL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| HWVDELVAL | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HWVDELVAL | | | | | | | |
| R-0h | | | | | | | |

表 15-190 HWVDELVAL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | HWVDELVAL | R | 0h | 硬件谷底延迟值寄存器。此只读寄存器反映按 VCAPCTL[VDELAYDIV] 定义的公式计算出的硬件延迟值。它反映最新硬件计算结果，每次触发谷底捕获序列并更新 VCAP1 和 VCAP2 值时都可能改变。<br>复位类型：SYSRSn |

## 15.16.2.84 VCNTVAL 寄存器

偏移地址：1FCh　复位值：0000h

硬件谷底计数器寄存器

表 15-191 VCNTVAL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| VCNTVAL | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| VCNTVAL | | | | | | | |
| R-0h | | | | | | | |

表 15-192 VCNTVAL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | VCNTVAL | R | 0h | 谷底时基计数器寄存器。反映 VCNTCFG 所选 STOPEDGE 发生时捕获的 VCNT 值。<br>复位类型：SYSRSn |

## 15.16.3 寄存器与 DriverLib 函数对应表

### 15.16.3.1 EPWM 寄存器与 DriverLib 函数对应表

表 15-193 EPWM 寄存器与 DriverLib 函数对应表

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| TBCTL | epwm.c | EPWM_setEmulationMode |
| TBCTL | epwm.h | EPWM_setCountModeAfterSync |
| TBCTL | epwm.h | EPWM_setClockPrescaler |
| TBCTL | epwm.h | EPWM_forceSyncPulse |
| TBCTL | epwm.h | EPWM_setOneShotSyncOutTrigger |
| TBCTL | epwm.h | EPWM_setPeriodLoadMode |
| TBCTL | epwm.h | EPWM_enablePhaseShiftLoad |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| TBCTL | epwm.h | EPWM_disablePhaseShiftLoad |
| TBCTL | epwm.h | EPWM_setTimeBaseCounterMode |
| TBCTL | epwm.h | EPWM_selectPeriodLoadEvent |
| TBCTL | epwm.h | EPWM_enableOneShotSync |
| TBCTL | epwm.h | EPWM_disableOneShotSync |
| TBCTL | epwm.h | EPWM_startOneShotSync |
| TBCTL2 | epwm.h | EPWM_selectPeriodLoadEvent |
| TBCTL2 | epwm.h | EPWM_enableOneShotSync |
| TBCTL2 | epwm.h | EPWM_disableOneShotSync |
| TBCTL2 | epwm.h | EPWM_startOneShotSync |
| SYNCINSEL | epwm.h | EPWM_setSyncInPulseSource |
| TBCTR | epwm.h | EPWM_setTimeBaseCounter |
| TBCTR | epwm.h | EPWM_getTimeBaseCounterValue |
| TBSTS | epwm.h | EPWM_getTimeBaseCounterOverflowStatus |
| TBSTS | epwm.h | EPWM_clearTimeBaseCounterOverflowEvent |
| TBSTS | epwm.h | EPWM_getSyncStatus |
| TBSTS | epwm.h | EPWM_clearSyncEvent |
| TBSTS | epwm.h | EPWM_getTimeBaseCounterDirection |
| SYNCOUTEN | epwm.h | EPWM_enableSyncOutPulseSource |
| SYNCOUTEN | epwm.h | EPWM_disableSyncOutPulseSource |
| TBCTL3 | epwm.h | EPWM_setOneShotSyncOutTrigger |
| CMPCTL | epwm.h | EPWM_setCounterCompareShadowLoadMode |
| CMPCTL | epwm.h | EPWM_disableCounterCompareShadowLoadMode |
| CMPCTL | epwm.h | EPWM_getCounterCompareShadowStatus |
| CMPCTL2 | epwm.h | EPWM_setCounterCompareShadowLoadMode |
| CMPCTL2 | epwm.h | EPWM_disableCounterCompareShadowLoadMode |
| DBCTL | epwm.h | EPWM_setDeadBandOutputSwapMode |
| DBCTL | epwm.h | EPWM_setDeadBandDelayMode |
| DBCTL | epwm.h | EPWM_setDeadBandDelayPolarity |
| DBCTL | epwm.h | EPWM_setRisingEdgeDeadBandDelayInput |
| DBCTL | epwm.h | EPWM_setFallingEdgeDeadBandDelayInput |
| DBCTL | epwm.h | EPWM_setDeadBandControlShadowLoadMode |
| DBCTL | epwm.h | EPWM_disableDeadBandControlShadowLoadMode |
| DBCTL | epwm.h | EPWM_setRisingEdgeDelayCountShadowLoadMode |
| DBCTL | epwm.h | EPWM_disableRisingEdgeDelayCountShadowLoadMode |
| DBCTL | epwm.h | EPWM_setFallingEdgeDelayCountShadowLoadMode |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| DBCTL | epwm.h | EPWM_disableFallingEdgeDelayCountShadowLoadMode |
| DBCTL | epwm.h | EPWM_setDeadBandCounterClock |
| DBCTL2 | epwm.h | EPWM_setDeadBandControlShadowLoadMode |
| DBCTL2 | epwm.h | EPWM_disableDeadBandControlShadowLoadMode |
| AQCTL | epwm.h | EPWM_setActionQualifierShadowLoadMode |
| AQCTL | epwm.h | EPWM_disableActionQualifierShadowLoadMode |
| AQCTL | epwm.h | EPWM_setActionQualifierAction |
| AQCTL | epwm.h | EPWM_setActionQualifierActionComplete |
| AQCTL | epwm.h | EPWM_setAdditionalActionQualifierActionComplete |
| AQTSRCSEL | epwm.h | EPWM_setActionQualifierT1TriggerSource |
| AQTSRCSEL | epwm.h | EPWM_setActionQualifierT2TriggerSource |
| PCCTL | epwm.h | EPWM_enableChopper |
| PCCTL | epwm.h | EPWM_disableChopper |
| PCCTL | epwm.h | EPWM_setChopperDutyCycle |
| PCCTL | epwm.h | EPWM_setChopperFreq |
| PCCTL | epwm.h | EPWM_setChopperFirstPulseWidth |
| VCAPCTL | epwm.h | EPWM_enableValleyCapture |
| VCAPCTL | epwm.h | EPWM_disableValleyCapture |
| VCAPCTL | epwm.h | EPWM_startValleyCapture |
| VCAPCTL | epwm.h | EPWM_setValleyTriggerSource |
| VCAPCTL | epwm.h | EPWM_enableValleyHWDelay |
| VCAPCTL | epwm.h | EPWM_disableValleyHWDelay |
| VCAPCTL | epwm.h | EPWM_setValleyDelayDivider |
| VCNTCFG | epwm.h | EPWM_setValleyTriggerEdgeCounts |
| VCNTCFG | epwm.h | EPWM_getValleyEdgeStatus |
| HRCNFG | - | - |
| HRPWR | - | - |
| HRMSTEP | - | - |
| HRCNFG2 | - | - |
| HRPCTL | - | - |
| TRREM | - | - |
| GLDCTL | epwm.h | EPWM_enableGlobalLoad |
| GLDCTL | epwm.h | EPWM_disableGlobalLoad |
| GLDCTL | epwm.h | EPWM_setGlobalLoadTrigger |
| GLDCTL | epwm.h | EPWM_setGlobalLoadEventPrescale |
| GLDCTL | epwm.h | EPWM_getGlobalLoadEventCount |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| GLDCTL | epwm.h | EPWM_disableGlobalLoadOneShotMode |
| GLDCTL | epwm.h | EPWM_enableGlobalLoadOneShotMode |
| GLDCTL | epwm.h | EPWM_setGlobalLoadOneShotLatch |
| GLDCTL | epwm.h | EPWM_forceGlobalLoadOneShotEvent |
| GLDCFG | epwm.h | EPWM_enableGlobalLoadRegisters |
| GLDCFG | epwm.h | EPWM_disableGlobalLoadRegisters |
| XLINK | epwm.h | EPWM_setupEPWMLinks |
| AQCTLA | epwm.h | EPWM_setActionQualifierAction |
| AQCTLA | epwm.h | EPWM_setActionQualifierActionComplete |
| AQCTLA | epwm.h | EPWM_setAdditionalActionQualifierActionComplete |
| AQCTLA2 | epwm.h | EPWM_setActionQualifierAction |
| AQCTLA2 | epwm.h | EPWM_setAdditionalActionQualifierActionComplete |
| AQCTLB | - | 参见 AQCTLA |
| AQCTLB2 | - | 参见 AQCTLA2 |
| AQSFRC | epwm.h | EPWM_setActionQualifierContSWForceShadowMode |
| AQSFRC | epwm.h | EPWM_setActionQualifierSWAction |
| AQSFRC | epwm.h | EPWM_forceActionQualifierSWAction |
| AQCSFRC | epwm.h | EPWM_setActionQualifierContSWForceAction |
| DBREDHR | - | - |
| DBRED | epwm.h | EPWM_setRisingEdgeDelayCount |
| DBFEDHR | - | - |
| DBFED | epwm.h | EPWM_setFallingEdgeDelayCount |
| TBPHS | epwm.h | EPWM_setPhaseShift |
| TBPRDHR | - | - |
| TBPRD | epwm.h | EPWM_setTimeBasePeriod |
| TBPRD | epwm.h | EPWM_getTimeBasePeriod |
| CMPA | epwm.h | EPWM_setCounterCompareValue |
| CMPA | epwm.h | EPWM_getCounterCompareValue |
| CMPB | - | 参见 CMPA |
| CMPC | epwm.h | EPWM_setCounterCompareShadowLoadMode |
| CMPC | epwm.h | EPWM_disableCounterCompareShadowLoadMode |
| CMPC | epwm.h | EPWM_getCounterCompareShadowStatus |
| CMPD | - | 参见 CMPC |
| GLDCTL2 | epwm.h | EPWM_setGlobalLoadOneShotLatch |
| GLDCTL2 | epwm.h | EPWM_forceGlobalLoadOneShotEvent |
| SWVDELVAL | epwm.h | EPWM_setValleySWDelayValue |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| TZSEL | epwm.h | EPWM_enableTripZoneSignals |
| TZSEL | epwm.h | EPWM_disableTripZoneSignals |
| TZDCSEL | epwm.h | EPWM_setTripZoneDigitalCompareEventCondition |
| TZCTL | epwm.h | EPWM_enableTripZoneAdvAction |
| TZCTL | epwm.h | EPWM_disableTripZoneAdvAction |
| TZCTL | epwm.h | EPWM_setTripZoneAction |
| TZCTL | epwm.h | EPWM_setTripZoneAdvAction |
| TZCTL | epwm.h | EPWM_setTripZoneAdvDigitalCompareActionA |
| TZCTL | epwm.h | EPWM_setTripZoneAdvDigitalCompareActionB |
| TZCTL2 | epwm.h | EPWM_enableTripZoneAdvAction |
| TZCTL2 | epwm.h | EPWM_disableTripZoneAdvAction |
| TZCTL2 | epwm.h | EPWM_setTripZoneAdvAction |
| TZCTL2 | epwm.h | EPWM_setTripZoneAdvDigitalCompareActionA |
| TZCTL2 | epwm.h | EPWM_setTripZoneAdvDigitalCompareActionB |
| TZCTLDCA | epwm.h | EPWM_setTripZoneAdvDigitalCompareActionA |
| TZCTLDCB | epwm.h | EPWM_setTripZoneAdvDigitalCompareActionB |
| TZEINT | epwm.h | EPWM_enableTripZoneInterrupt |
| TZEINT | epwm.h | EPWM_disableTripZoneInterrupt |
| TZFLG | epwm.h | EPWM_getTripZoneFlagStatus |
| TZCBCFLG | epwm.h | EPWM_getCycleByCycleTripZoneFlagStatus |
| TZOSTFLG | epwm.h | EPWM_getOneShotTripZoneFlagStatus |
| TZCLR | epwm.h | EPWM_selectCycleByCycleTripZoneClearEvent |
| TZCLR | epwm.h | EPWM_clearTripZoneFlag |
| TZCBCCLR | epwm.h | EPWM_clearCycleByCycleTripZoneFlag |
| TZOSTCLR | epwm.h | EPWM_clearOneShotTripZoneFlag |
| TZFRC | epwm.h | EPWM_forceTripZoneEvent |
| ETSEL | epwm.h | EPWM_enableInterrupt |
| ETSEL | epwm.h | EPWM_disableInterrupt |
| ETSEL | epwm.h | EPWM_setInterruptSource |
| ETSEL | epwm.h | EPWM_enableADCTrigger |
| ETSEL | epwm.h | EPWM_disableADCTrigger |
| ETSEL | epwm.h | EPWM_setADCTriggerSource |
| ETPS | epwm.h | EPWM_setInterruptEventCount |
| ETPS | epwm.h | EPWM_setADCTriggerEventPrescale |
| ETFLG | epwm.h | EPWM_getEventTriggerInterruptStatus |
| ETFLG | epwm.h | EPWM_getADCTriggerFlagStatus |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| ETCLR | epwm.h | EPWM_clearEventTriggerInterruptFlag |
| ETCLR | epwm.h | EPWM_clearADCTriggerFlag |
| ETFRC | epwm.h | EPWM_forceEventTriggerInterrupt |
| ETFRC | epwm.h | EPWM_forceADCTrigger |
| ETINTPS | epwm.h | EPWM_setInterruptEventCount |
| ETINTPS | epwm.h | EPWM_getInterruptEventCount |
| ETSOCPS | epwm.h | EPWM_setADCTriggerEventPrescale |
| ETSOCPS | epwm.h | EPWM_getADCTriggerEventCount |
| ETCNTINITCTL | epwm.h | EPWM_enableInterruptEventCountInit |
| ETCNTINITCTL | epwm.h | EPWM_disableInterruptEventCountInit |
| ETCNTINITCTL | epwm.h | EPWM_forceInterruptEventCountInit |
| ETCNTINITCTL | epwm.h | EPWM_enableADCTriggerEventCountInit |
| ETCNTINITCTL | epwm.h | EPWM_disableADCTriggerEventCountInit |
| ETCNTINITCTL | epwm.h | EPWM_forceADCTriggerEventCountInit |
| ETCNTINIT | epwm.h | EPWM_enableInterruptEventCountInit |
| ETCNTINIT | epwm.h | EPWM_disableInterruptEventCountInit |
| ETCNTINIT | epwm.h | EPWM_forceInterruptEventCountInit |
| ETCNTINIT | epwm.h | EPWM_setInterruptEventCountInitValue |
| ETCNTINIT | epwm.h | EPWM_enableADCTriggerEventCountInit |
| ETCNTINIT | epwm.h | EPWM_disableADCTriggerEventCountInit |
| ETCNTINIT | epwm.h | EPWM_forceADCTriggerEventCountInit |
| ETCNTINIT | epwm.h | EPWM_setADCTriggerEventCountInitValue |
| DCTRIPSEL | epwm.h | EPWM_selectDigitalCompareTripInput |
| DCTRIPSEL | epwm.h | EPWM_enableDigitalCompareTripCombinationInput |
| DCACTL | epwm.h | EPWM_setDigitalCompareEventSource |
| DCACTL | epwm.h | EPWM_setDigitalCompareEventSyncMode |
| DCACTL | epwm.h | EPWM_enableDigitalCompareADCTrigger |
| DCACTL | epwm.h | EPWM_disableDigitalCompareADCTrigger |
| DCACTL | epwm.h | EPWM_enableDigitalCompareSyncEvent |
| DCACTL | epwm.h | EPWM_disableDigitalCompareSyncEvent |
| DCACTL | epwm.h | EPWM_setDigitalCompareCBCLatchMode |
| DCACTL | epwm.h | EPWM_selectDigitalCompareCBCLatchClearEvent |
| DCACTL | epwm.h | EPWM_getDigitalCompareCBCLatchStatus |
| DCBCTL | - | 参见 DCACTL |
| DCFCTL | epwm.h | EPWM_enableDigitalCompareBlankingWindow |
| DCFCTL | epwm.h | EPWM_disableDigitalCompareBlankingWindow |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| DCFCTL | epwm.h | EPWM_enableDigitalCompareWindowInverseMode |
| DCFCTL | epwm.h | EPWM_disableDigitalCompareWindowInverseMode |
| DCFCTL | epwm.h | EPWM_setDigitalCompareBlankingEvent |
| DCFCTL | epwm.h | EPWM_setDigitalCompareFilterInput |
| DCFCTL | epwm.h | EPWM_enableDigitalCompareEdgeFilter |
| DCFCTL | epwm.h | EPWM_disableDigitalCompareEdgeFilter |
| DCFCTL | epwm.h | EPWM_setDigitalCompareEdgeFilterMode |
| DCFCTL | epwm.h | EPWM_setDigitalCompareEdgeFilterEdgeCount |
| DCFCTL | epwm.h | EPWM_getDigitalCompareEdgeFilterEdgeCount |
| DCFCTL | epwm.h | EPWM_getDigitalCompareEdgeFilterEdgeStatus |
| DCCAPCTL | epwm.h | EPWM_enableDigitalCompareCounterCapture |
| DCCAPCTL | epwm.h | EPWM_disableDigitalCompareCounterCapture |
| DCCAPCTL | epwm.h | EPWM_setDigitalCompareCounterShadowMode |
| DCCAPCTL | epwm.h | EPWM_getDigitalCompareCaptureStatus |
| DCCAPCTL | epwm.h | EPWM_clearDigitalCompareCaptureStatusFlag |
| DCCAPCTL | epwm.h | EPWM_configureDigitalCompareCounterCaptureMode |
| DCFOFFSET | epwm.h | EPWM_setDigitalCompareWindowOffset |
| DCFOFFSET | epwm.h | EPWM_getDigitalCompareBlankingWindowOffsetCount |
| DCFOFFSETCNT | epwm.h | EPWM_getDigitalCompareBlankingWindowOffsetCount |
| DCFWINDOW | epwm.h | EPWM_setDigitalCompareWindowLength |
| DCFWINDOW | epwm.h | EPWM_getDigitalCompareBlankingWindowLengthCount |
| DCFWINDOWCNT | epwm.h | EPWM_getDigitalCompareBlankingWindowLengthCount |
| DCCAP | epwm.h | EPWM_enableDigitalCompareCounterCapture |
| DCCAP | epwm.h | EPWM_disableDigitalCompareCounterCapture |
| DCCAP | epwm.h | EPWM_setDigitalCompareCounterShadowMode |
| DCCAP | epwm.h | EPWM_getDigitalCompareCaptureStatus |
| DCCAP | epwm.h | EPWM_clearDigitalCompareCaptureStatusFlag |
| DCCAP | epwm.h | EPWM_configureDigitalCompareCounterCaptureMode |
| DCCAP | epwm.h | EPWM_getDigitalCompareCaptureCount |
| DCAHTRIPSEL | epwm.h | EPWM_enableDigitalCompareTripCombinationInput |
| DCAHTRIPSEL | epwm.h | EPWM_disableDigitalCompareTripCombinationInput |
| DCALTRIPSEL | - | 参见 DCAHTRIPSEL |
| DCBHTRIPSEL | - | 参见 DCAHTRIPSEL |
| DCBLTRIPSEL | - | 参见 DCAHTRIPSEL |
| LOCK | epwm.h | EPWM_lockRegisters |
| HWVDELVAL | epwm.h | EPWM_getValleyHWDelay |

| 寄存器 | 文件 | DriverLib 函数 |
| --- | --- | --- |
| VCNTVAL | epwm.h | EPWM_getValleyCount |

### 15.16.3.2 HRPWM 寄存器与 DriverLib 函数对应表

表 15-194 HRPWM 寄存器与 DriverLib 函数对应表

| 寄存器 | 文件 | DriverLib 函数 |
| --- | --- | --- |
| TBCTL | - | - |
| TBCTL2 | - | - |
| EPWMSYNCINSEL | - | - |
| TBCTR | - | - |
| TBSTS | - | - |
| EPWMSYNCOUTEN | - | - |
| TBCTL3 | - | - |
| CMPCTL | - | - |
| CMPCTL2 | - | - |
| DBCTL | - | - |
| DBCTL2 | - | - |
| AQCTL | - | - |
| AQTSRCSEL | - | - |
| PCCTL | - | - |
| VCAPCTL | - | - |
| VCNTCFG | - | - |
| HRCNFG | hrpwm.h | HRPWM_setMEPEdgeSelect |
| HRCNFG | hrpwm.h | HRPWM_setMEPControlMode |
| HRCNFG | hrpwm.h | HRPWM_setCounterCompareShadowLoadEvent |
| HRCNFG | hrpwm.h | HRPWM_setOutputSwapMode |
| HRCNFG | hrpwm.h | HRPWM_setChannelBOutputPath |
| HRCNFG | hrpwm.h | HRPWM_enableAutoConversion |
| HRCNFG | hrpwm.h | HRPWM_disableAutoConversion |
| HRCNFG | hrpwm.h | HRPWM_setDeadbandMEPEdgeSelect |
| HRCNFG | hrpwm.h | HRPWM_setRisingEdgeDelayLoadMode |
| HRCNFG | hrpwm.h | HRPWM_setFallingEdgeDelayLoadMode |
| HRPWR | - | - |
| HRMSTEP | hrpwm.h | HRPWM_setMEPStep |
| HRCNFG2 | hrpwm.h | HRPWM_setDeadbandMEPEdgeSelect |
| HRCNFG2 | hrpwm.h | HRPWM_setRisingEdgeDelayLoadMode |
| HRCNFG2 | hrpwm.h | HRPWM_setFallingEdgeDelayLoadMode |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| HRPCTL | hrpwm.h | HRPWM_enablePeriodControl |
| HRPCTL | hrpwm.h | HRPWM_disablePeriodControl |
| HRPCTL | hrpwm.h | HRPWM_enablePhaseShiftLoad |
| HRPCTL | hrpwm.h | HRPWM_disablePhaseShiftLoad |
| HRPCTL | hrpwm.h | HRPWM_setSyncPulseSource |
| TRREM | hrpwm.h | HRPWM_setTranslatorRemainder |
| GLDCTL | - | - |
| GLDCFG | - | - |
| EPWMXLINK | - | - |
| AQCTLA | - | - |
| AQCTLA2 | - | - |
| AQCTLB | - | - |
| AQCTLB2 | - | - |
| AQSFRC | - | - |
| AQCSFRC | - | - |
| DBREDHR | hrpwm.h | HRPWM_setRisingEdgeDelay |
| DBREDHR | hrpwm.h | HRPWM_setHiResRisingEdgeDelayOnly |
| DBRED | hrpwm.h | HRPWM_setRisingEdgeDelay |
| DBRED | hrpwm.h | HRPWM_setHiResRisingEdgeDelayOnly |
| DBFEDHR | hrpwm.h | HRPWM_setFallingEdgeDelay |
| DBFEDHR | hrpwm.h | HRPWM_setHiResFallingEdgeDelayOnly |
| DBFED | hrpwm.h | HRPWM_setFallingEdgeDelay |
| DBFED | hrpwm.h | HRPWM_setHiResFallingEdgeDelayOnly |
| TBPHS | hrpwm.h | HRPWM_setPhaseShift |
| TBPHS | hrpwm.h | HRPWM_setHiResPhaseShiftOnly |
| TBPRDHR | hrpwm.h | HRPWM_setTimeBasePeriod |
| TBPRDHR | hrpwm.h | HRPWM_setHiResTimeBasePeriodOnly |
| TBPRDHR | hrpwm.h | HRPWM_getTimeBasePeriod |
| TBPRDHR | hrpwm.h | HRPWM_getHiResTimeBasePeriodOnly |
| TBPRD | hrpwm.h | HRPWM_setTimeBasePeriod |
| TBPRD | hrpwm.h | HRPWM_setHiResTimeBasePeriodOnly |
| TBPRD | hrpwm.h | HRPWM_getTimeBasePeriod |
| TBPRD | hrpwm.h | HRPWM_getHiResTimeBasePeriodOnly |
| CMPA | hrpwm.h | HRPWM_setCounterCompareValue |
| CMPA | hrpwm.h | HRPWM_setHiResCounterCompareValueOnly |
| CMPA | hrpwm.h | HRPWM_getCounterCompareValue |

| 寄存器 | 文件 | DriverLib 函数 |
| --- | --- | --- |
| CMPA | hrpwm.h | HRPWM_getHiResCounterCompareValueOnly |
| CMPB | hrpwm.h | HRPWM_setCounterCompareValue |
| CMPB | hrpwm.h | HRPWM_setHiResCounterCompareValueOnly |
| CMPB | hrpwm.h | HRPWM_getCounterCompareValue |
| CMPB | hrpwm.h | HRPWM_getHiResCounterCompareValueOnly |
| CMPC | - | - |
| CMPD | - | - |
| GLDCTL2 | - | - |
| SWVDELVAL | - | - |
| TZSEL | - | - |
| TZDCSEL | - | - |
| TZCTL | - | - |
| TZCTL2 | - | - |
| TZCTLDCA | - | - |
| TZCTLDCB | - | - |
| TZEINT | - | - |
| TZFLG | - | - |
| TZCBCFLG | - | - |
| TZOSTFLG | - | - |
| TZCLR | - | - |
| TZCBCCLR | - | - |
| TZOSTCLR | - | - |
| TZFRC | - | - |
| ETSEL | - | - |
| ETPS | - | - |
| ETFLG | - | - |
| ETCLR | - | - |
| ETFRC | - | - |
| ETINTPS | - | - |
| ETSOCPS | - | - |
| ETCNTINITCTL | - | - |
| ETCNTINIT | - | - |
| DCTRIPSEL | - | - |
| DCACTL | - | - |
| DCBCTL | - | - |
| DCFCTL | - | - |

| 寄存器 | 文件 | DriverLib 函数 |
| --- | --- | --- |
| DCCAPCTL | - | - |
| DCFOFFSET | - | - |
| DCFOFFSETCNT | - | - |
| DCFWINDOW | - | - |
| DCFWINDOWCNT | - | - |
| DCCAP | - | - |
| DCAHTRIPSEL | - | - |
| DCALTRIPSEL | - | - |
| DCBHTRIPSEL | - | - |
| DCBLTRIPSEL | - | - |
| EPWMLOCK | hrpwm.h | HRPWM_lockRegisters |
| HWVDELVAL | - | - |
| VCNTVAL | - | - |

# 16 增强型捕获模块（eCAP）

## 16.1 简介

本章描述了增强型捕获（eCAP）模块，该模块用于需要精确外部事件时序的系统中。增强型捕获（eCAP）模块属于 Type 2 eCAP。

eCAP 模块具备以下功能：

- 旋转机械转速测量（例如通过霍尔传感器检测的齿轮传动系统）
- 位置传感器脉冲间隔时间测量
- 脉冲序列信号的周期与占空比测量
- 解码由占空比编码的电流/电压传感器所获取的电流或电压幅值

本章所述 eCAP 模块功能包括：

- 4 个事件时间戳寄存器（每个 32 位）
- 最多四个时序时间戳捕获事件的边沿极性选择
- 任一事件触发中断
- 单次捕获最多四个事件时间戳
- 采用四通道循环缓冲区的连续模式时间戳捕获
- 绝对时间戳捕获
- 差分（Delta）模式时间戳捕获
- 当未启用捕获模式时，eCAP 模块可配置为单通道 PWM 输出

与 Type 0 eCAP 相比，Type 1 eCAP 的捕获功能通过以下新增特性得到增强：

- 事件过滤器复位位
- 向 ECCTL2[ctrfiltreset]写入 1 可清除事件过滤器、模数计数器及所有待处理中断标志。复位该位有助于初始化和调试。
- 模数计数器状态位
- 模数计数器（ECCTL2[modcntrsts]）指示下一个加载的捕获寄存器。在 Type 0 eCAP 中，无法获取模数计数器的当前状态。

- DMA 触发源
- 新增 eCAPxDMA 作为 DMA 触发源。 CEVT [1-4]可配置为 eCAPxDMA 的触发源。
- 输入多路复用器
- ECCTL0[inputsel]可选择 128 个输入信号中的一个。
- EALLOW 保护
- 为关键寄存器添加 EALLOW 保护功能。为保持与 Type-0 的软件兼容性，请配置 DEV_CFG_REGS.ecaptype 以解除这些寄存器的保护。

Type2eCAP 的捕获功能较 Type1eCAP 有所增强，新增以下特性：

- 新增 ECAPxSYNCINSEL 寄存器
- 每个 eCAP 均配备该寄存器以选择外部 SYNCIN 。每个 eCAP 可配置独立 SYNCIN 信号。

## 16.2 描述

eCAP 模块代表一个完整的采集通道，可根据目标设备需求多次实例化。本指南中，一个 eCAP 通道包含以下独立关键资源：

- 采集输入可通过输入 X-BAR 连接
- 128:1 输入多路复用器
- 输出 X-BAR 用于 APWM 模式配置输出
- 32 位时间基准（计数器）
- 4 个 32 位时间戳采集寄存器（CAP1-CAP4）
- 四阶段时序器（模 4 计数器），其与外部事件（eCAP 引脚上升/下降沿）保持同步。
- 模数计数器状态寄存器（modcntrsts）用于指示时序器状态
- 所有四个事件的独立边沿极性（上升/下降沿）选择
- 输入捕获信号预缩放（2-62 或旁路模式）

- 单稳态比较寄存器（两位）用于在 1-4 个时间戳事件后冻结捕获
- 采用四层循环缓冲器（CAP1-CAP4）方案控制连续时间戳捕获
- 可重置事件过滤器、模数计数器及中断标志
- 支持四个捕获事件中的任意中断功能
- 独立 DMA 触发器
- EALLOW 保护以控制寄存器

## 16.3 eCAP 设备引脚配置

输入 X-BAR 将设备引脚连接至模块作为输入端口。设备上的任何 GPIO 均可配置为输入端口。通过设置 GPxQSELn 寄存器位，可将 GPIO 输入验证模式设置为同步或异步。使用同步输入有助于抗噪，但会影响 eCAP 精度±2 个周期。内部上拉功能可通过 GPyPUD 寄存器进行配置。由于采用 GPIO 模式，GPyINV 寄存器可对信号进行反相处理。

对于新型 1 型 eCAP 模块，还需配置 128:1 输入多路复用器（参见图中）。该多路复用器可通过配置 ECCTL0.inportsel 来选择表中详述的多种输入信号。

输出 X-BAR 必须用于将输出号连接至 OUTPUTXBARx 输出端口。随后需配置 GPIO 多路复用器，将 OUTPUTXBARx 线路连接至 GPIO 多路复用器的多个 IO 引脚之一。为避免引脚出现毛刺，需先配置 GPyGMUX 位（同时保持对应 GPyMUX 位默认为零），再将 GPyMUX 寄存器写入目标值。

表 16-1 eCAP 输入选择

| ECCTL0.INPUTSEL | eCAP 输入 |
|---|---|
| 0 | INPUTXBAR1 |
| 1 | INPUTXBAR2 |
| 2 | INPUTXBAR3 |
| 3 | INPUTXBAR4 |
| 4 | INPUTXBAR5 |
| 5 | INPUTXBAR6 |
| 6 | INPUTXBAR7 |

| ECCTL0.INPUTSEL | eCAP 输入 |
| --- | --- |
| 7 | INPUTXBAR8 |
| 8 | INPUTXBAR9 |
| 9 | INPUTXBAR10 |
| 10 | INPUTXBAR11 |
| 11 | INPUTXBAR12 |
| 12 | INPUTXBAR13 |
| 13 | INPUTXBAR14 |
| 14 | INPUTXBAR15 |
| 15 | INPUTXBAR16 |
| 16 | CLB1_OUT14 |
| 17 | CLB1_OUT15 |
| 18 | CLB2_OUT14 |
| 19 | CLB2_OUT15 |
| 20 | DCANA_INT0 |
| 21-23 | 保留 |
| 24 | OUTPUTXBAR1 |
| 25 | OUTPUTXBAR2 |
| 26 | OUTPUTXBAR3 |
| 27 | OUTPUTXBAR4 |
| 28 | OUTPUTXBAR5 |
| 29 | OUTPUTXBAR6 |
| 30 | OUTPUTXBAR7 |
| 31 | OUTPUTXBAR8 |
| 32-35 | 保留 |
| 36 | ADCCEVT1 |
| 37 | ADCCEVT2 |
| 38 | ADCCEVT3 |
| 39 | ADCCEVT4 |
| 40-43 | 保留 |
| 44 | ADCAEVT1 |
| 45 | ADCAEVT2 |

| ECCTL0.INPUTSEL | eCAP 输入 |
| --- | --- |
| 46 | ADCAEVT3 |
| 47 | ADCAEVT4 |
| 48 | FSIRXA_MEASURE |
| 49 | FSIRXA_MEASURE_RISE |
| 50 | FSIRXA_MEASURE_FALL |
| 51-95 | 保留 |
| 96 | CMPSS1_CTRIPL |
| 97 | CMPSS2_CTRIPL |
| 98 | CMPSS3_CTRIPL |
| 99 | CMPSS4_CTRIPL |
| 100-107 | 保留 |
| 108 | CMPSS1_CTRIPH |
| 109 | CMPSS2_CTRIPH |
| 110 | CMPSS3_CTRIPH |
| 111 | CMPSS4_CTRIPH |
| 112-114 | 保留 |
| 115 | GPIO8 |
| 116 | GPIO9 |
| 117 | GPIO22 |
| 118 | GPIO23 |
| 119-119 | 保留 |
| 120 | CMPSS1_CTRIPH_OR_CTRIPL |
| 121 | CMPSS2_CTRIPH_OR_CTRIPL |
| 122 | CMPSS3_CTRIPH_OR_CTRIPL |
| 123 | CMPSS4_CTRIPH_OR_CTRIPL |
| 124-126 | 保留 |
| 127 | ECAP1—INPUTXBAR7；ECAP2—INPUTXBAR8；<br>ECAP3—INPUTXBAR9 |

注意：ECAPxIN 的宽度必须至少达到 2 个 SYSCLK 周期，才能被 eCAP 模块正确捕获；否则，输入脉冲可能被 SYSCLK 漏采。

## 16.4 捕获和 APWM 操作模式

当 eCAP 模块未用于输入捕获时，可利用其资源实现单通道 PWM 发生器（支持 32 位功能）。该计数器采用递增模式运行，为非对称脉宽调制（PWM）波形提供时间基准。CAP1 和 CAP2 寄存器分别作为活动周期寄存器和比较寄存器，CAP3 和 CAP4 寄存器则分别作为周期寄存器和比较影子寄存器。图中展示了捕获模式与辅助脉宽调制（APWM）模式的高级别操作视图。

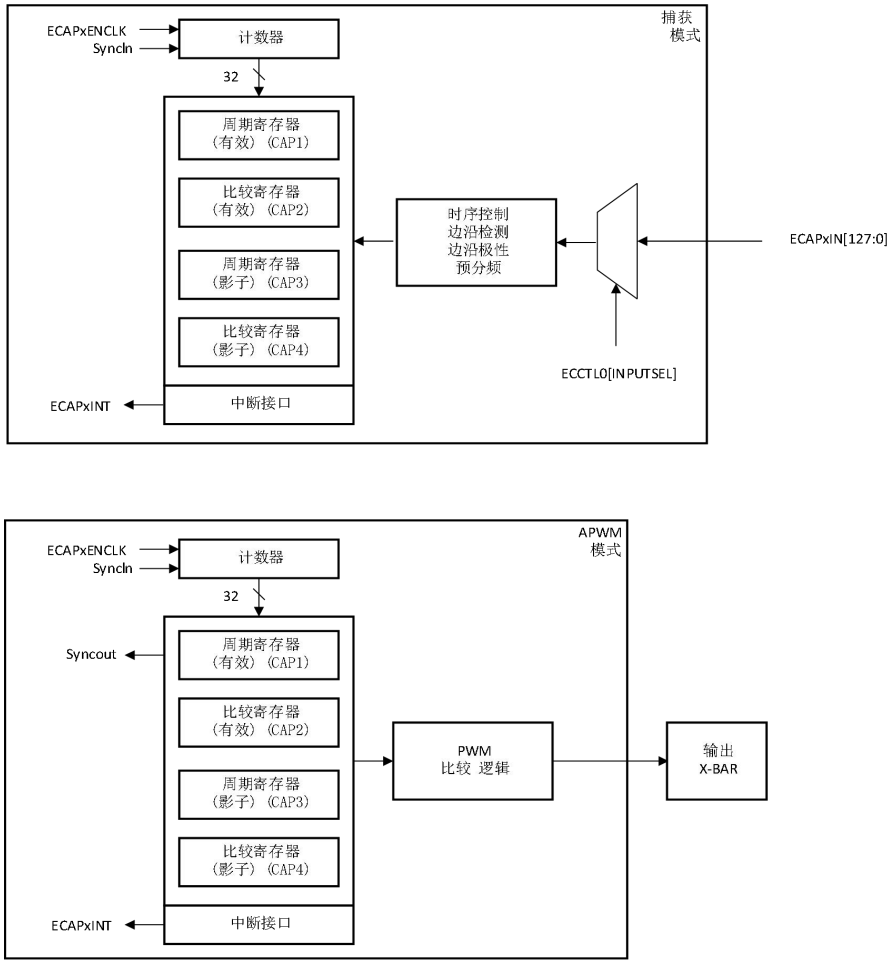



图 16-1 捕获和 APWM 操作模式

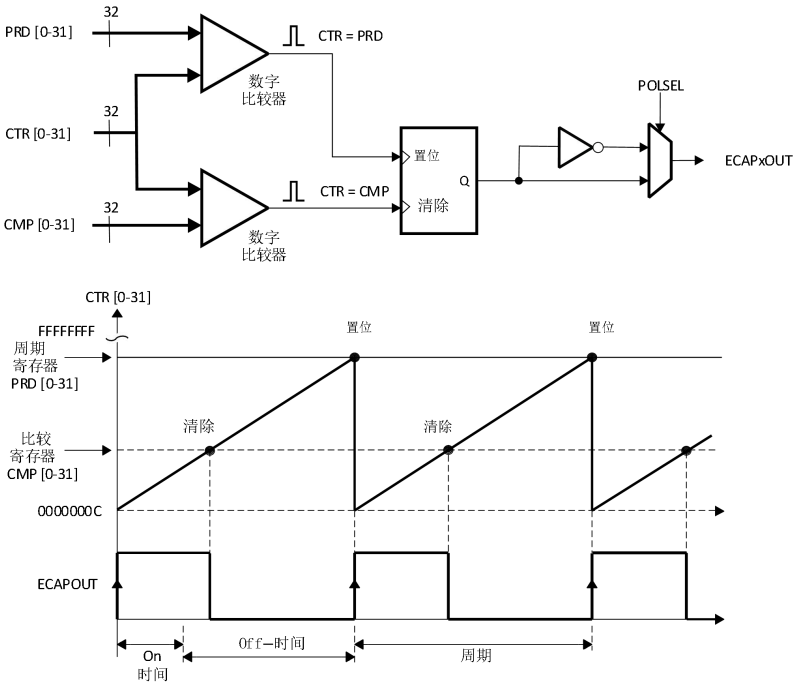
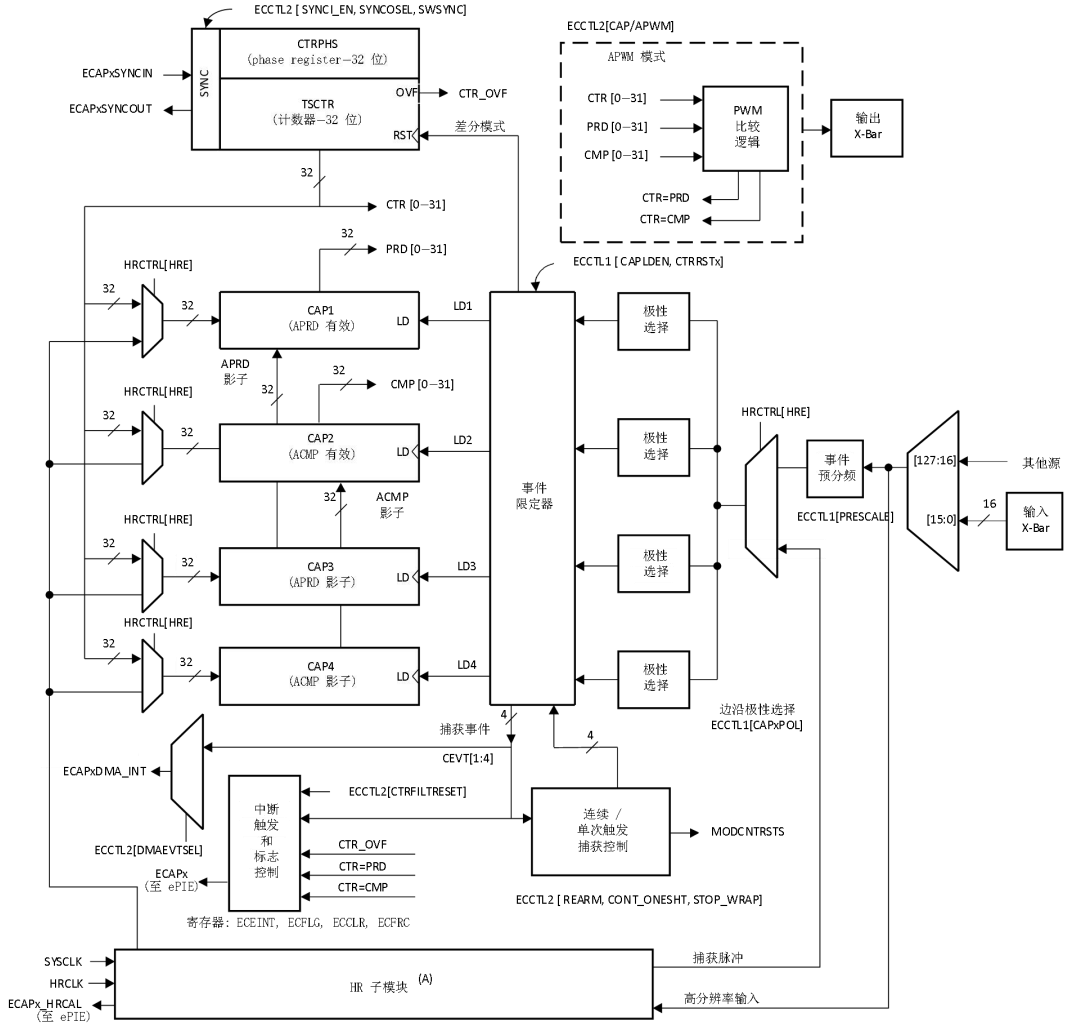



图 16-2 APWM 模式下计数器比较与 PRD 对 eCAP 输出的影响

## 16.5 捕获方式描述

图中展示了实现捕获功能的各个组件。

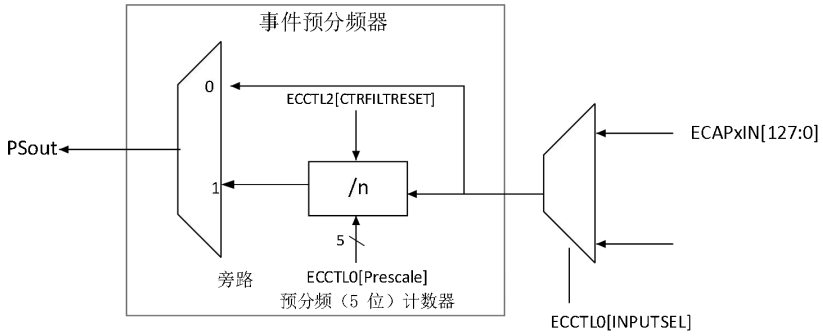



A. 并非所有 eCAP 模块都提供 HRCAP 子模块；不提供时，高分辨率多路选择器和硬件不会实现

图 16-3 eCAP 捕获单元功能框图

## 16.5.1 事件预分频器

输入捕获信号（脉冲序列）可采用 N=2-62（2 的倍数）进行预分频，或直接跳过预分频器。该功能在输入极高频信号时尤为实用。图中展示功能框图，图中呈现预分频器的工作原理。事件预分频器可通过设置 ECCTL2.ctrfiltreset 寄存器位进行复位。




A. 预分频值为 1（ECCTL1[13:9] = 0,0,0,0,0）时，输入捕获信号完全旁路预分频逻辑。

B. 更改预分频配置后的第一个上升沿不送入捕获逻辑；预分频值从第二个上升沿起生效。

图 16-4 事件预标度控制

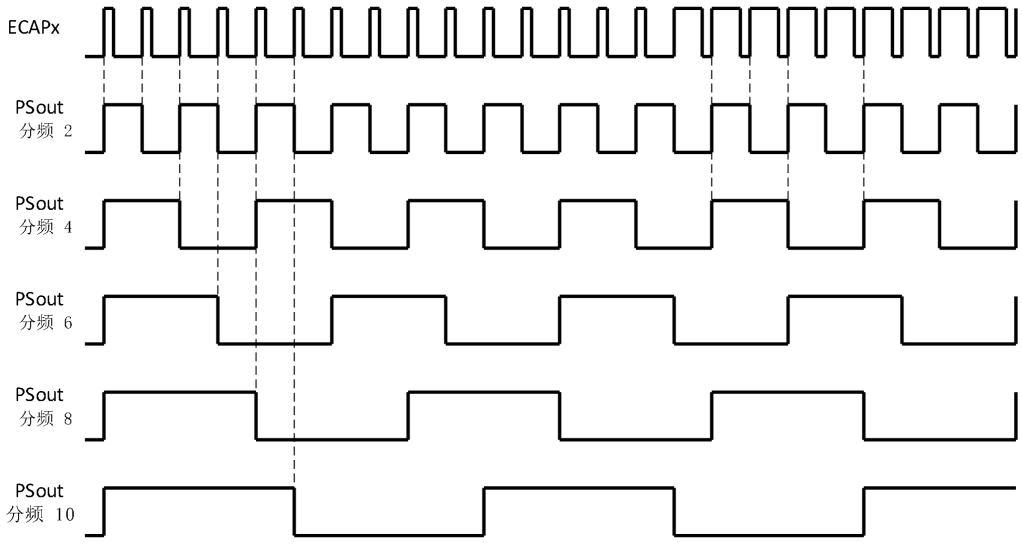



图 16-5 预标度函数波形

## 16.5.2 边缘极性选择与限定符

功能与特性包括：

- 采用四个独立的边沿极性（上升沿/下降沿）选择多路复用器，每个捕获事件对应一个。
- 每个边沿（最多 4 个）由 Modulo4 时序器进行事件校验。
- Mod4 计数器将边沿事件选通至对应的 CAPx 寄存器。CAPx 寄存器在下降沿时加载。

### 16.5.3 连续/一次性控制

eCAP 在连续/单次模式下的工作原理：

- Mod4（2 位）计数器通过边沿触发事件（CEVT1-CEVT4）进行递增。
- Mod4 计数器持续计数（0->1->2->3->0），除非被停止，否则会进行循环计数。
- 在单次操作模式下，通过 2 位停止寄存器（STOP_WRAP）比较 Mod4 计数器输出，当两者相等时，停止 Mod4 计数器并抑制 CAP1-CAP4 寄存器的后续加载。在此模式下，若通过配置 ECCTL1. CTRRSTx 位使 TSCCTR 计数器在捕获事件（CEVTx）时复位，即使达到 STOP_WRAP 值且未触发复位（rearm）操作，系统仍会在 CEVTx 捕获事件后持续复位 TSCCTR 计数器。

该连续/单次触发模块通过单次触发动作控制 Mod4 计数器的启动、停止及复位（归零）功能，该动作可由停止值比较器触发，并通过软件控制重新激活。

激活后，eCAP 模块将等待 1-4 次（由停止值定义）捕获事件，随后冻结 Mod4 计数器及 CAP1-4 寄存器（时间戳）的内容。

重新装填使 eCAP 模块为下一次捕获序列做好准备。同时，重新装填会将 Mod4 计数器清零，并允许再次加载 CAP1-4 寄存器（前提是 caplden 位已设置）。

在连续模式下，Mod4 计数器继续运行（0→1→2→3→0，单次动作被忽略，捕获值继续以循环缓冲序列写入 CAP1-4 寄存器）。

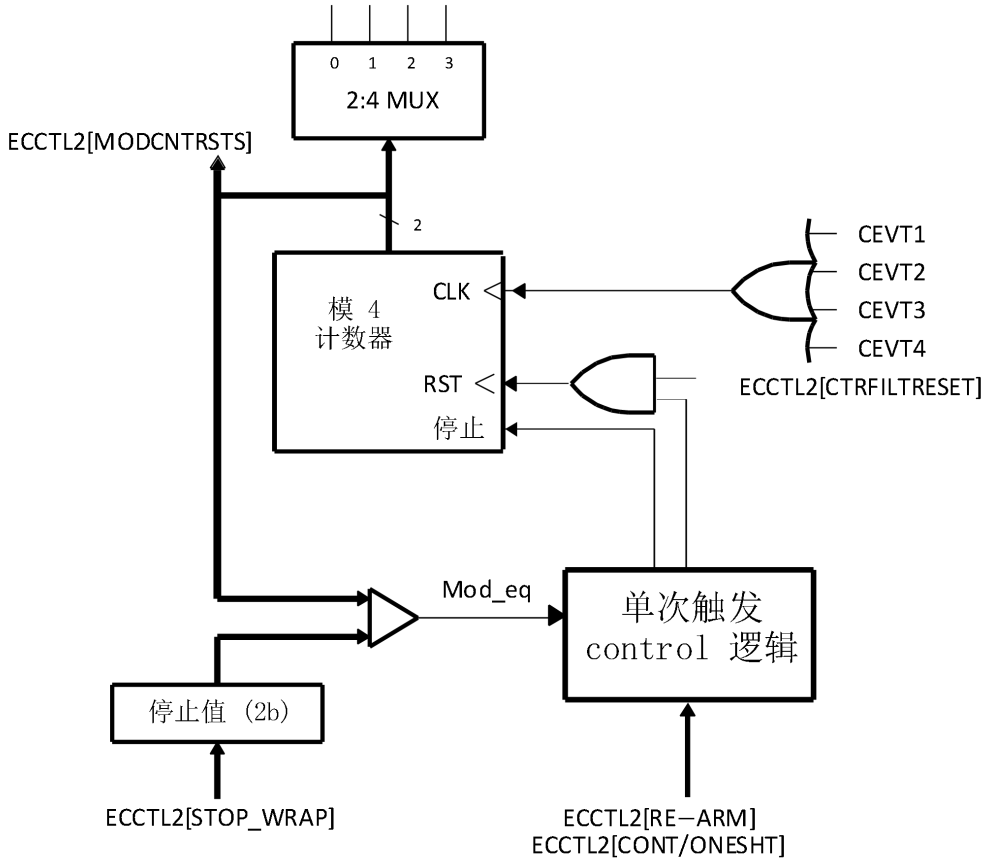



图 16-6 连续/一次性阻滞的详细说明

## 16.5.4 32 位计数器与相位控制

该计数器为事件捕获提供时间基准，采用系统时钟进行计时。

通过硬件和软件强制同步，配置相位寄存器实现与其他计数器的同步。该功能在 APWM 模式下特别实用，当需要模块间相位偏移时尤为有效。

在四个事件加载信号中任一触发时，均可选择重置 32 位计数器，这对时间差捕获非常有用。系统会先捕获 32 位计数器的当前值，随后通过 LD1-LD4 任一信号将计数器重置为 0。

## 16.5.5 CAP1-CAP4 寄存器

这些 32 位寄存器由 32 位计数器定时器总线 CTR[0-31]提供，当对应的 LD 输入被选通时，这些寄存器会被加载（捕获时间戳）。

控制位 caplden 可抑制捕获寄存器的加载。在单稳态操作期间，当停止条件发生时（StopValue=Mod4），该位会自动清除（加载被抑制）。

在 APWM 模式下，CAP1 和 CAP2 寄存器分别成为活动周期寄存器和比较寄存器。

在 APWM 操作期间，CAP3 和 CAP4 寄存器则分别成为 CAP1 和 CAP2 的影子寄存器（APRD 和 ACMP）。

### 16.5.6 eCAP 同步

eCAP 模块可通过选择公共 SYNCIN 源实现同步。eCAP 的 SYNCIN 源可采用软件同步或外部同步。外部同步信号可来自 eCAP、X-Bar 或 EPWM 。如图所示，eCAP 模块的 SWSYNC 与 SYNC 信号进行逻辑或运算。如图所示，SYNC 信号由 ECAPxSYNCINSEL[SEL]的选择决定。

注意：若 SWSYNC 被强制退出 ECAPx，同步脉冲接收端的 ECAPs 将延迟一个时钟周期。该延迟可通过 CTRPHS 寄存器进行校正。若存在外部 SYNC 信号，所有 ECAPs 将同时接收到该信号。

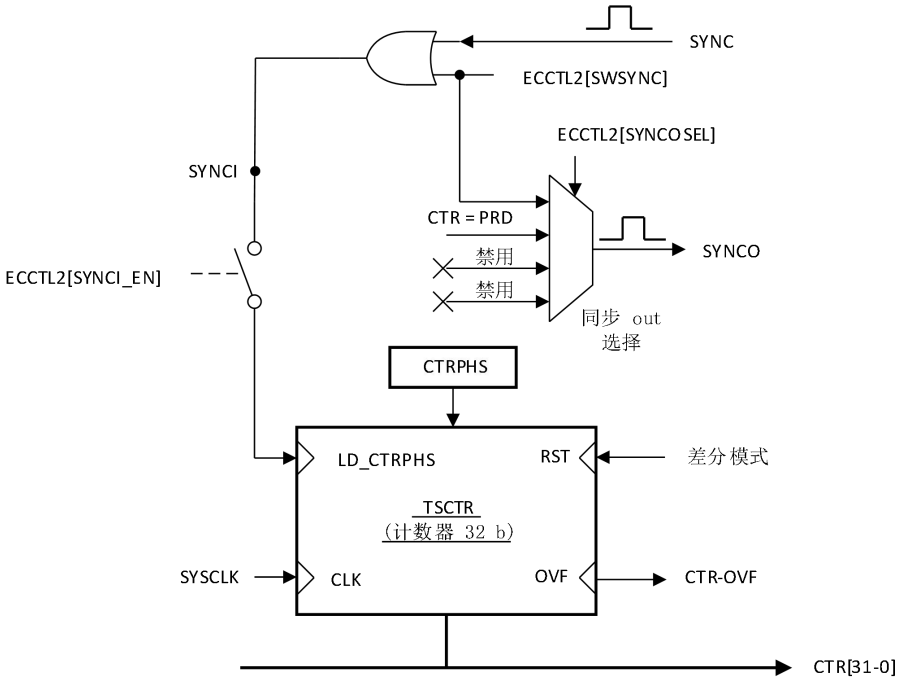



图 16-7 计数器与同步块的详细说明

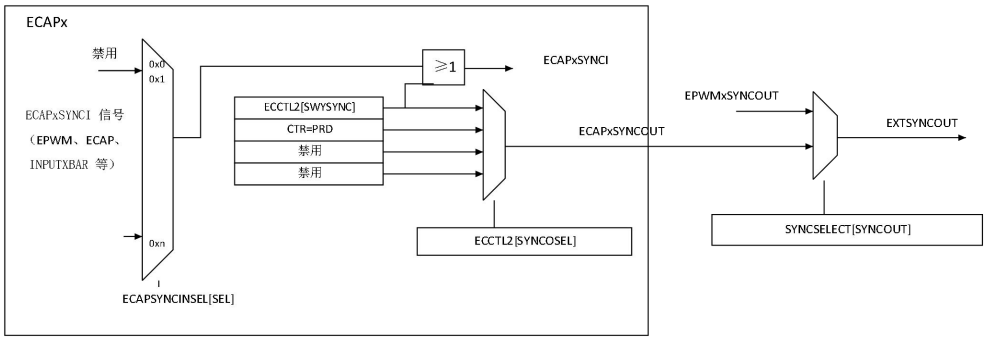



图 16-8 eCAP 同步机制

### 16.5.6.1 示例 1 - 使用 ECAP 模块的 SWSYNC

使用 SWSYNC 与 ECAP1 和 ECAP2 的步骤如下：

- 配置 ECAP [1..2].ecapsyncinsel.SEL=0x0，以禁用传入 eCAP1 的外部 SYNCIN 。
- 配置 ECAP [1..2].ECCTL2. SWSYNC =0x1，强制 TSCTR 计数器的软件同步。

若需与其他 eCAP 模块使用 SWSYNC ，请确保前序 eCAP 链路未生成干扰软件同步的同步输出信号。

## 16.5.7 中断控制

eCAP 中断控制的操作与功能包括（参见图）：

- 中断信号可由捕获事件（CEVT1-CEVT4， CTROVF ）或 APWM 事件（CTR=PRD，CTR=CMP）触发。
- 通过 CTROVF 功能，还可将计数器溢出事件（ffffffff->00000000）作为中断源。
- 捕获事件通过极性选择和 Mod4 门控分别实现边沿触发和时序排序。
- 可从 eCAPx 模块中选择其中一个事件作为中断源发送至 CLIC。
- 共可生成七种中断事件（CEVT1、CEVT2、CEVT3、CEVT4、 CNTOVF 、CTR=PRD、CTR=CMP）。
- 中断使能寄存器（ECEINT）用于启用/禁用单个中断源。中断标志寄存器（ECFLG）指示是否已锁存中断事件，并包含全局中断标志位（INT）。仅当满足以下条件时才会向 CLIC 生成中断脉冲：任一中断事件被启用、标志位为 1 且 INT 标志位为 0。中断服务例程必须先通过中断清除寄存器（ECCLR）清除全局

中断标志位及被服务事件，才能生成其他中断脉冲。通过向 ECCTL2[clrfiltreset] 写入 1 可重置事件过滤器并清除所有中断标志。若需强制触发中断事件，可使用中断强制寄存器（ECFRC），此功能适用于测试场景。

注意：CEVT1、CEVT2、CEVT3、CEVT4 标志仅在捕获模式下有效（ECCTL2[CAP/ APWM ==0]）。CTR=PRD、CTR=CMP 标志仅在 APWM 模式下有效（ECCTL2[CAP/ APWM ==1]）。 CNTOVF 标志在两种模式下均有效。

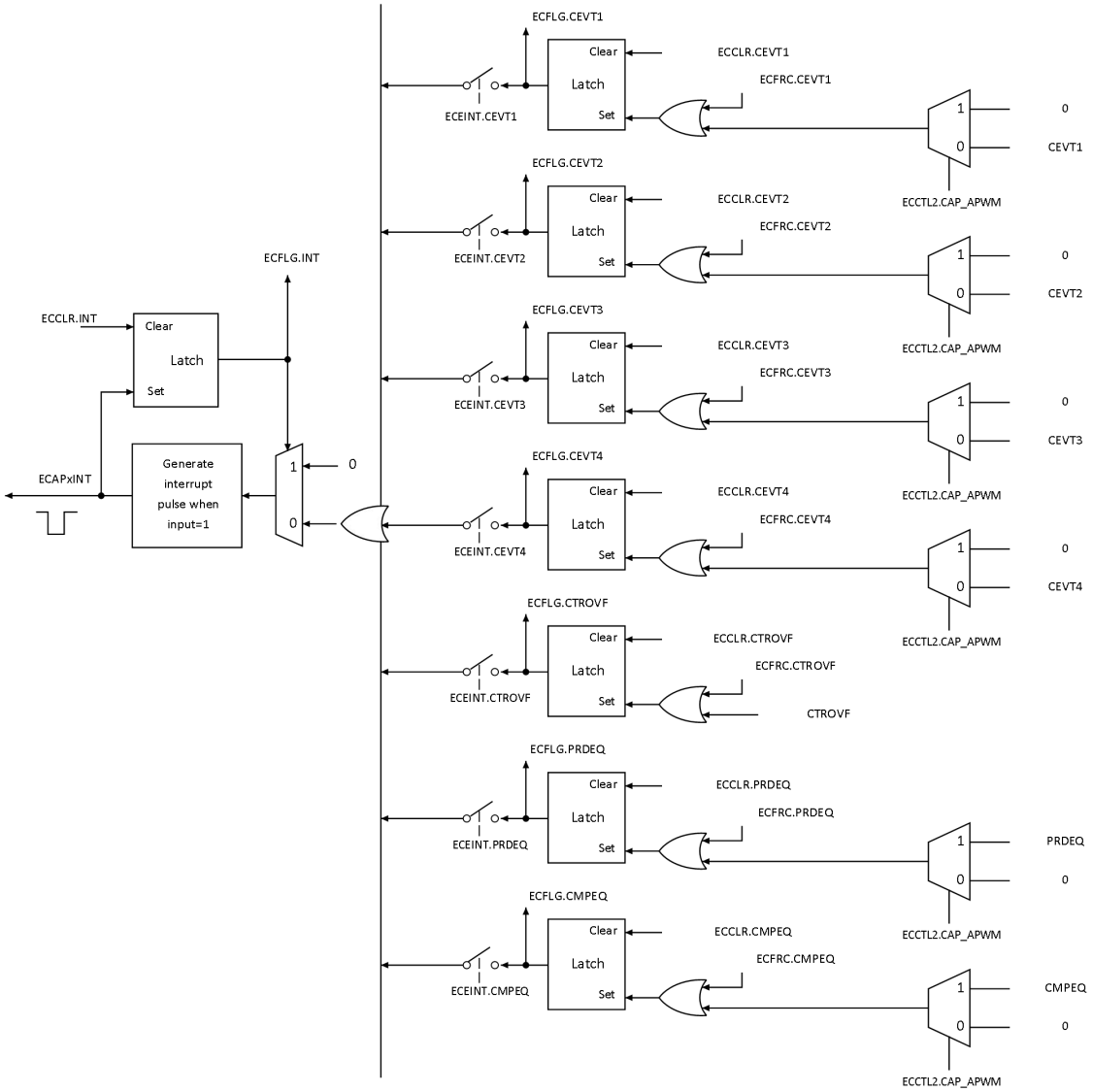



图 16-9 eCAP 模块中断

### 16.5.8 DMA 中断

在 0 型 eCAP 模块中，CPU 需通过 DMA 开始数据传输。1 型 eCAP 新增了一个独立的 DMA 触发器（ECAP_DMA_INT），该触发器支持通过 DMA 将捕获数据从 eCAP 寄存器持续传输至片上存储器。通过 ECCTL2 [dmaevtsel]，可选择四个可用中断事件（CEVT1、CEVT2、CEVT3 和 CEVT4）中的任意一个作为 ECAP_DMA_INT 的触发源。

### 16.5.9 影子负载与锁定控制

在捕获模式下，该逻辑会阻止（锁定）APRD 和 ACMP 寄存器对 CAP1 或 CAP2 的影子加载。

在 APWM 模式下，影子加载功能启用，允许两种选择：

- 即时加载——写入新值时， APRD 或 ACMP 会立即传输至 CAP1 或 CAP2。
- 周期相等时，CTR[31:0]=PRD[31:0]。

### 16.5.10 APWM 模式操作

APWM 模块主要操作亮点：

- 时间戳计数器总线通过 2 个数字（32 位）比较器实现可比性。
- 当 CAP1/2 寄存器未处于捕获模式时，其内容可作为 APWM 模式下的周期和比较值。
- 通过影子寄存器 APRD 和 ACMP（CAP3/4）实现双缓冲功能，影子寄存器内容会在写入时立即传输至 CAP1/2 寄存器，或在 CTR=PRD 触发时进行传输。
- 在 APWM 模式下，对 CAP1/CAP2 活动寄存器的写入操作会同步将相同值写入对应的影子寄存器 CAP3/CAP4，从而模拟即时模式。对影子寄存器 CAP3/CAP4 的写入会触发影子模式。
- 初始化阶段需同时对周期和比较寄存器进行写入，系统会自动将初始值复制至影子寄存器。运行时后续比较更新时，请使用影子寄存器。

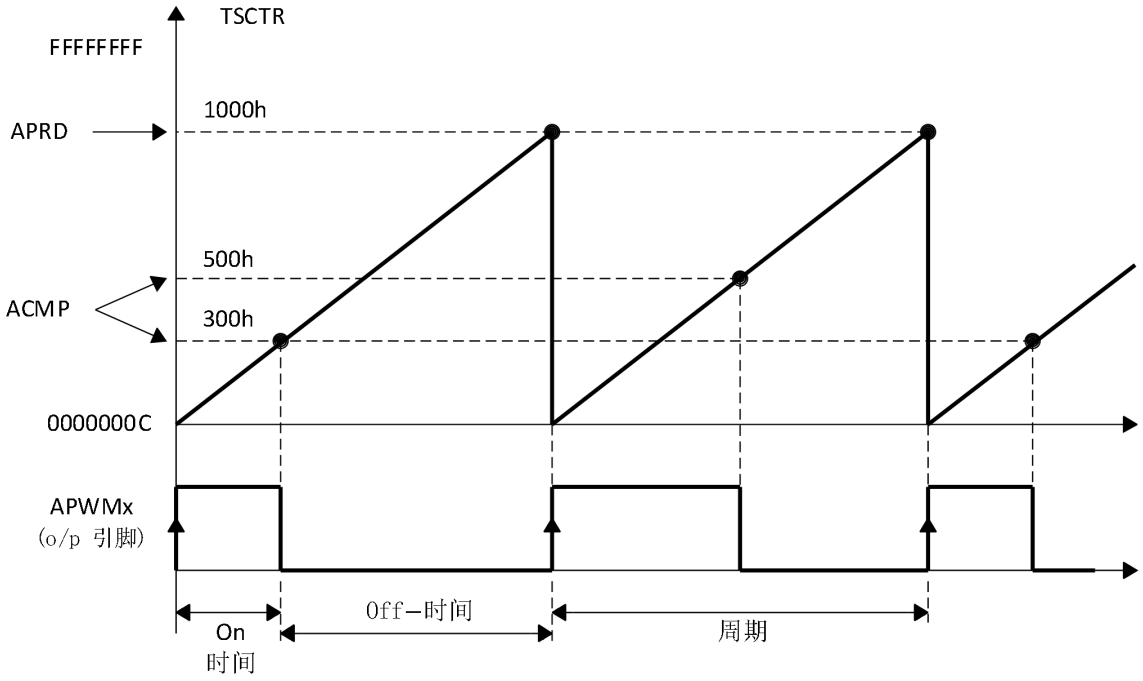



图 16-10 PWM 波形细节的 APWM 模式操作

APWM 高电平有效模式（apwmpol == 0）的行为如下：

CMP = 0x00000000, 低电平持续时间(0%占空比)

CMP = 0x00000001, 输出高 1 周期

CMP = 0x00000002, 输出高 2 周期

CMP = PERIOD, 输出高电平，仅 1 个周期例外（占空比<100%）

CMP = PERIOD+1, 全周期高输出（占空比=100%）

CMP > PERIOD+1, 全周期输出高

APWM 低电平有效模式（apwmpol == 1）的行为如下：

CMP = 0x00000000, 周期高电平输出(0%占空比)

CMP = 0x00000001, 输出低 1 周期

CMP = 0x00000002, 输出低 2 周期

CMP = PERIOD, 输出低电平，仅 1 个周期例外（占空比<100%）

CMP = PERIOD+1, 全周期低电平输出

CMP > PERIOD+1, 全周期输出低电平

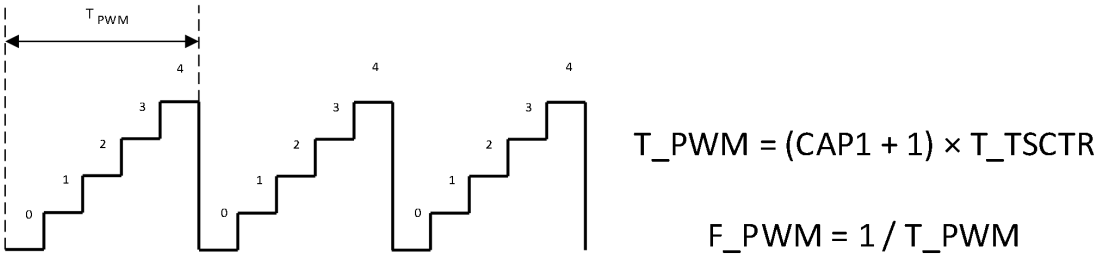



图 16-11 时基频率和周期的计算

# 16.6 eCAP 模块的应用

以下章节通过应用示例展示 eCAP 模块的操作方法。

## 16.6.1 示例 1—绝对时间戳操作上升沿触发

图中展示了连续捕获操作（Mod4 计数器循环）的示例。在此图中， TSCTR 在不复位的情况下持续递增，捕获事件仅在上升沿触发，从而提供周期（及频率）信息。

当触发事件时，系统首先捕获 TSCTR 内容（时间戳），随后将 Mod4 计数器递增至下一状态。当 TSCTR 达到 ffffffff（最大值）时，Mod4 计数器会循环回 00000000（图中未显示）。若发生此情况， CTROVF（计数器溢出）标志将被置位，并触发中断（若已启用）。捕获的时间戳在图示所示时刻（第四次事件后）有效，因此 CEVT4 事件可便捷触发中断，使 CPU 能够从 CAPx 寄存器读取数据。

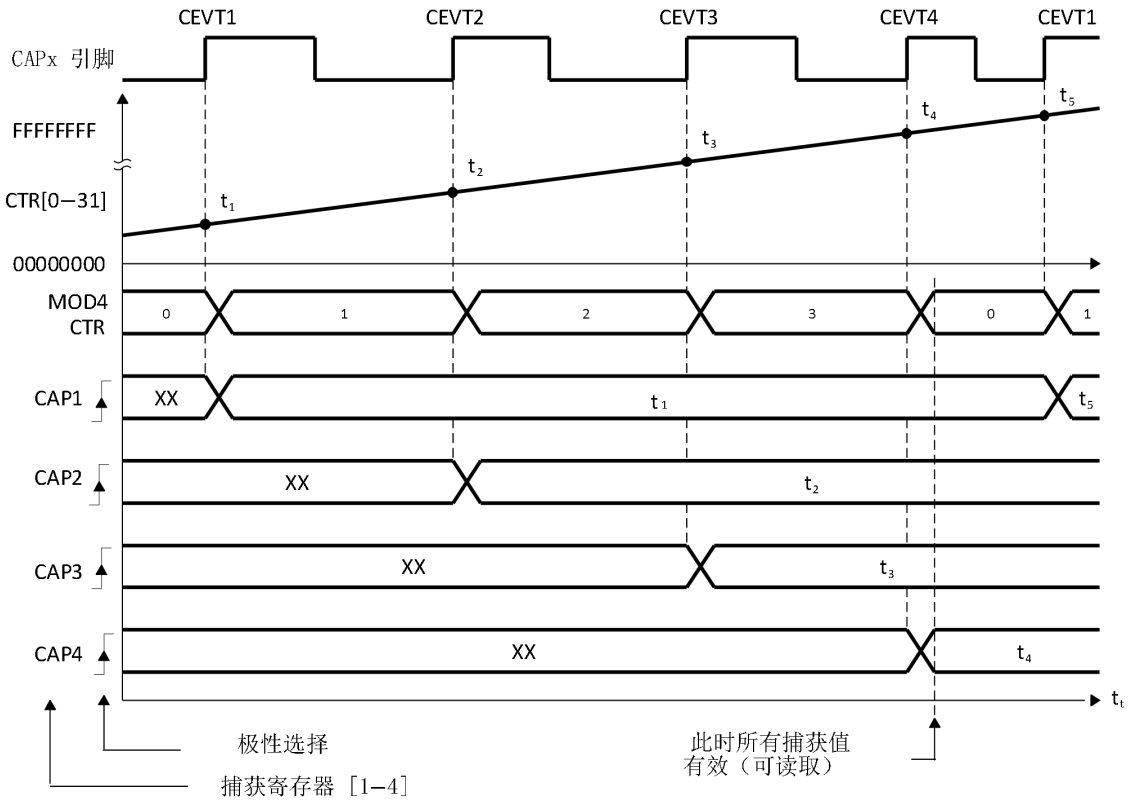



图 16-12 绝对时间戳和上升沿检测的捕获序列

## 16.6.2 示例 2—绝对时间戳操作上升沿与下降沿触发

如图所示，eCAP 工作模式与前一节基本一致，但捕获事件被限定为上升沿或下降沿，此时可获取周期和占空比信息，即：Period1 = t3 – t1，Period2 = t5 – t3，依此类推。占空比 1（导通时间百分比）= (t2 – t1) / Period1 × 100%，同理。占空比 1（关断时间百分比）= (t3 – t2) / Period1 × 100%，依此类推。

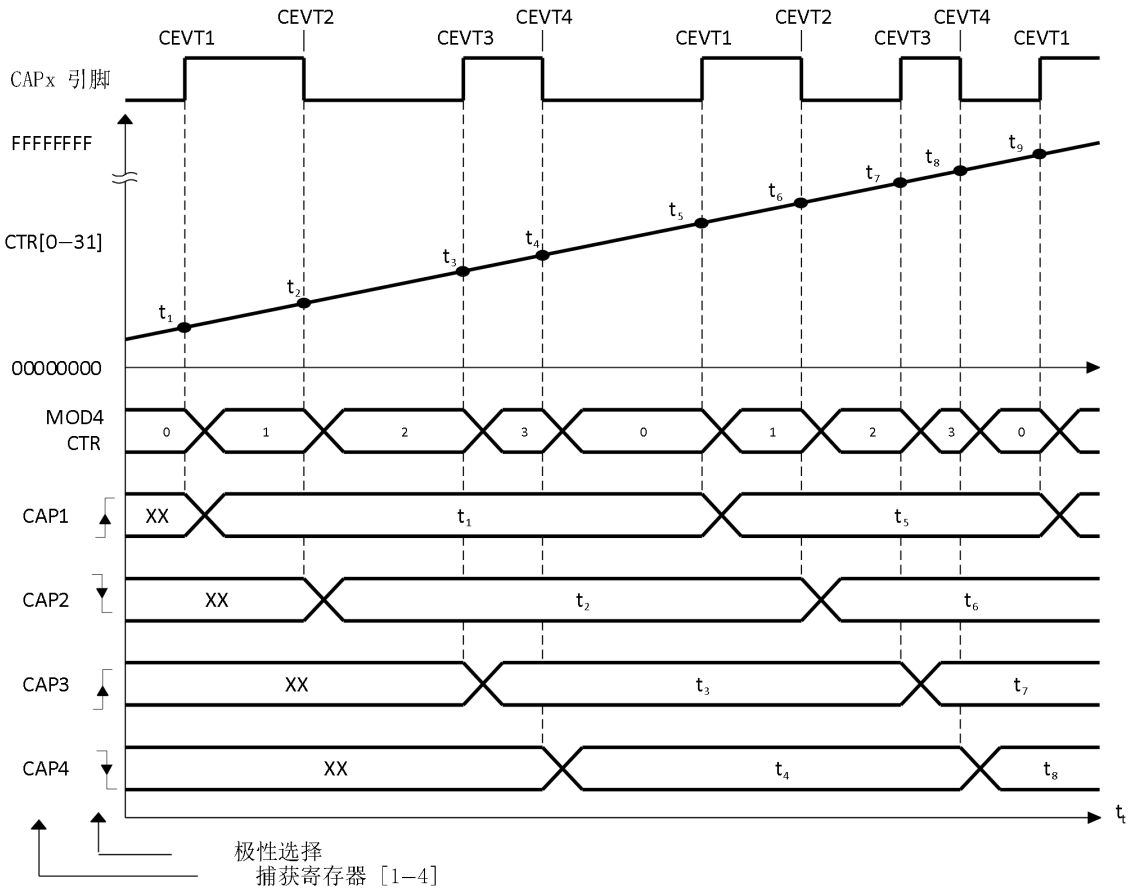



图 16-13 基于上升沿和下降沿检测的绝对时间戳捕获序列

### 16.6.3 示例 3—时间差（Delta）上升沿触发操作

图中展示了 eCAP 模块如何从脉冲序列波形中采集时序数据。该模块采用连续捕获模式（Mod4 计数器循环计数）。在 Delta 时间模式下， TSCTR 会在每个有效事件时重置为零。此时仅捕获上升沿事件：首先捕获 TSCTR 内容（时间戳），随后 TSCTR 重置为零，Mod4 计数器随即进入下一计数状态。若 TSCTR 在下一个事件前达到最大值 FFFFFFFFh 后溢出，TSCTR 将回绕至 00000000h 并继续计数，同时设置 CNTOVF（计数器溢出）标志，并触发中断（若已启用）。Delta 时间模式的优势在于 CAPx 内容可直接提供时序数据，无需 CPU 计算，即 Period1=T1、Period2=T2 等。如图所示，CEVT1 事件是读取时序数据的理想触发点，T1、T2、T3、T4 在此均为有效时序。

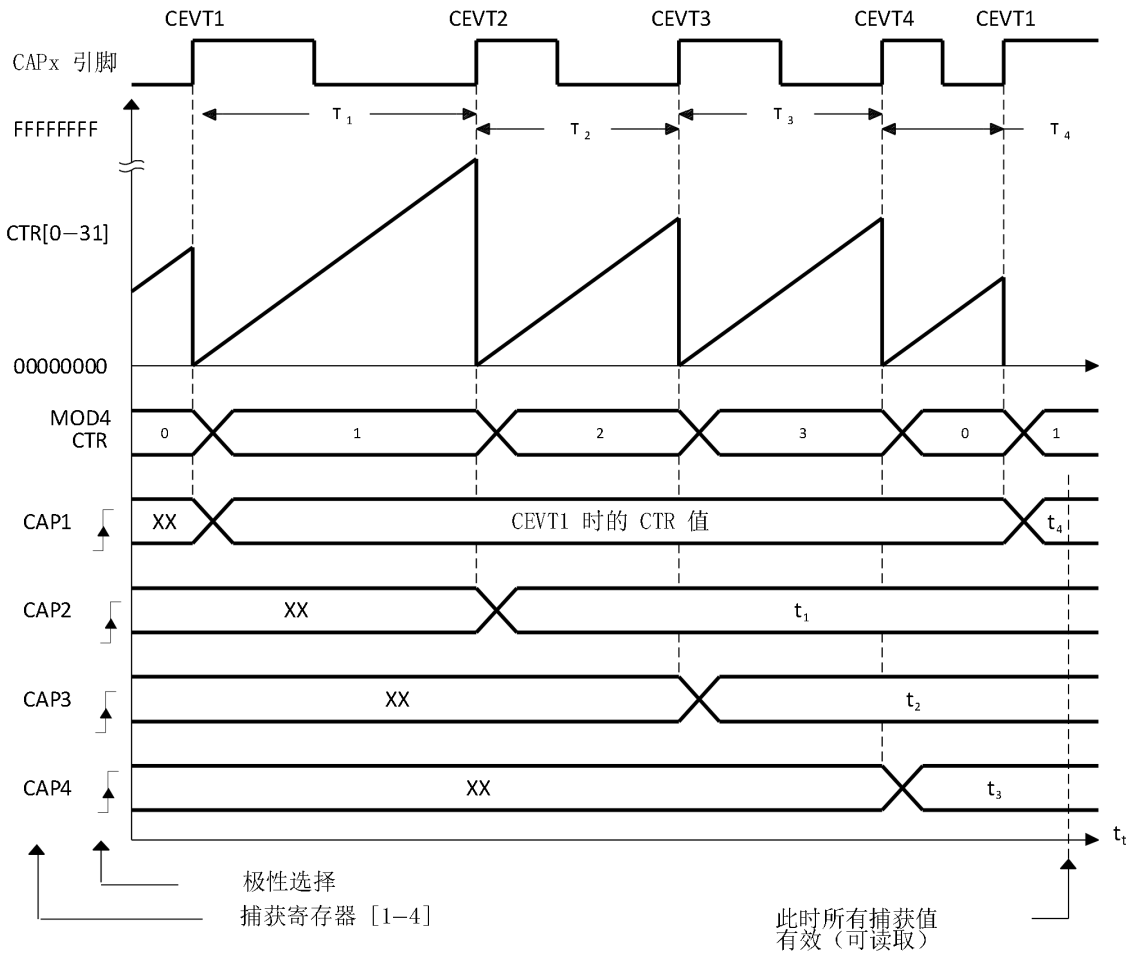



图 16-14 Delta 模式下时间戳与上升沿检测的捕获序列

## 16.6.4 示例 4—时间差（Delta）操作上升沿与下降沿触发

在图中，eCAP 的工作模式与前一节基本一致，主要区别在于捕获事件被限定为上升沿或下降沿。这种设计现在同时提供了周期和占空比信息，具体表现为：周期 1 = T1 + T2，周期 2 = T3 + T4，依此类推。占空比 1（占空比百分比）= T1 / 周期 1 × 100%，占空比 2（占空比百分比）= T2 / 周期 1 × 100%，依此类推。

初始化阶段需将周期和比较寄存器写入初始值，该操作会自动将初始值复制到影子寄存器中。在运行时进行后续比较更新时，必须使用影子寄存器。

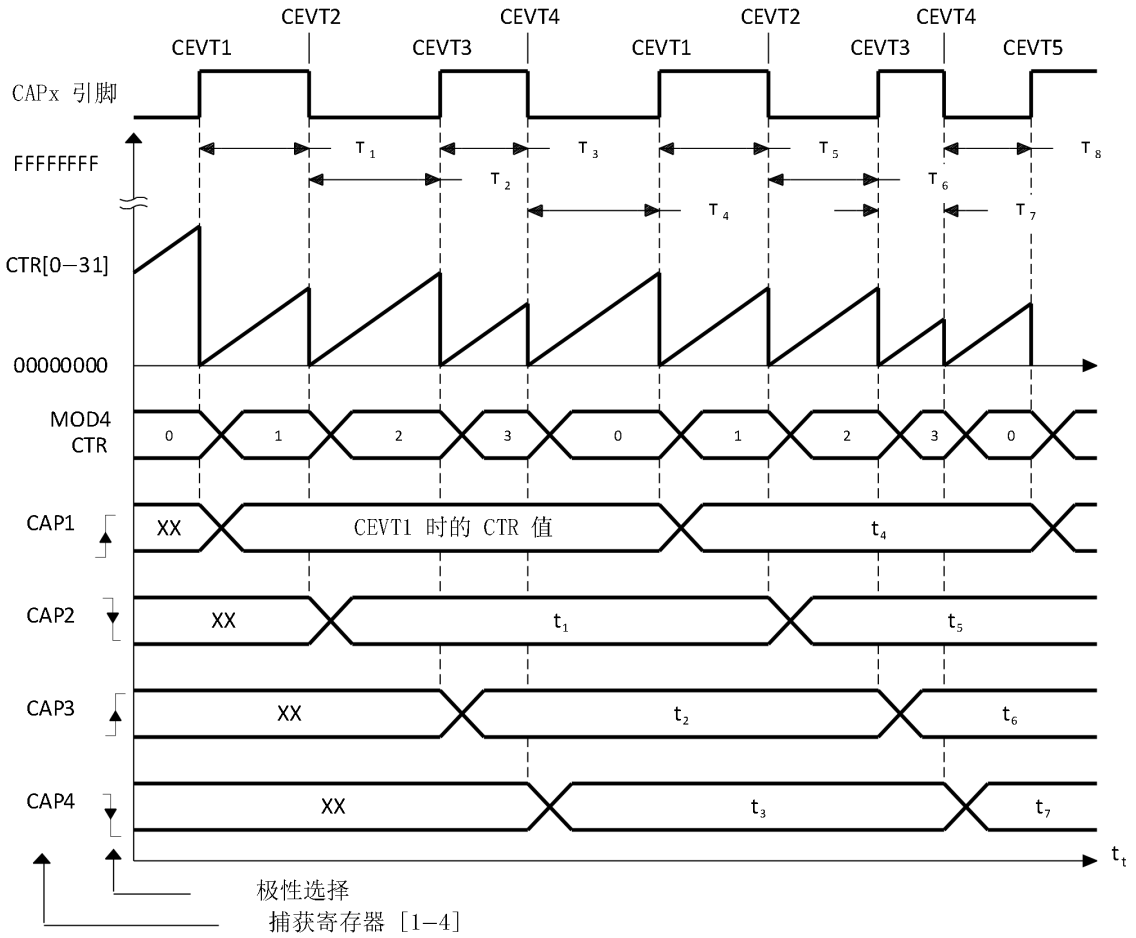



图 16-15 基于上升沿和下降沿检测的 Delta 模式时间戳捕获序列

## 16.7 APWM 模式的应用

本示例中，eCAP 模块被配置为 PWM 发生器。此时，APWMx 输出引脚生成一个非常简单的单通道 PWM 波形。 PWM 极性设为高电平有效，这意味着比较值（CAP2 寄存器此时作为比较寄存器）代表周期的导通时间（高电平）。反之，若 apwmpol 位设为低电平有效，则比较值代表关断时间。

### 16.7.1 示例 1—简单 PWM 生成（独立通道）

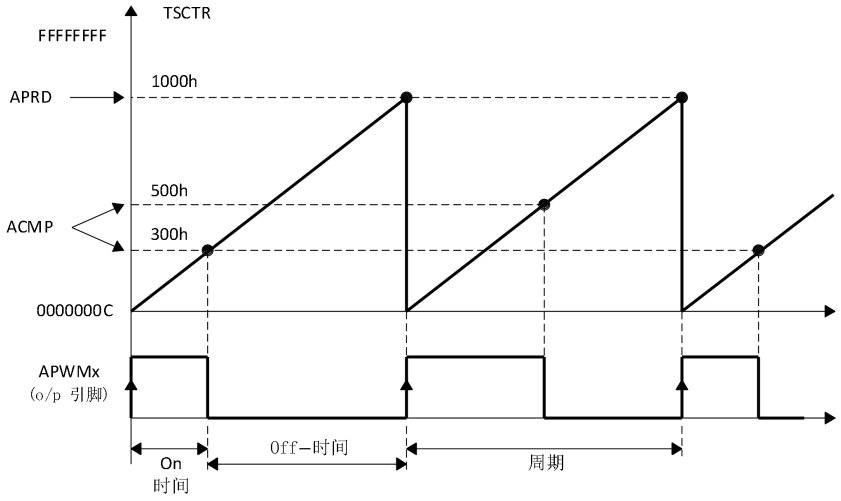



图 16-16 PWM APWM 模式运行波形细节

## 16.8 软件

### 16.8.1 ECAP 示例

注：这些示例位于 Ware 安装目录中，具体路径为：Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/ecap

#### 16.8.1.1 eCAP APWM 示例

文件： ecap_ex1_apwm .c

该程序以 APWM 模式配置 eCAP 模块。 PWM 波形将输出至 GPIO5。通过影子寄存器配置 PWM 频率在 5Hz 至 10Hz 之间变化，用于加载下一周期/比较值。

#### 16.8.1.2 eCAP 捕获 PWM 示例

文件： ecap_ex2_capture_pwm .c

本示例配置 ePWM3A 以实现：

- 升计数模式
- 周期起始值为 500，递增至 8000
- PRD 的切换输出

eCAP1 模块用于捕获 ePWM3A 输出信号的上升沿与下降沿之间的时间差。

外部连接：

- eCAP1 连接至 GPIO16
- ePWM3A 连接至 GPIO4
- 将 GPIO4 与 GPIO16 相连。

观察变量：

- ecap1PassCount - 成功捕获次数
- ecap1IntCount - 中断计数

### 16.8.1.3 eCAP APWM 移相示例

文件： ecap_ex3_apwm_phase_shift .c

该程序以 APWM 模式配置 eCAP1 和 eCAP2 模块，生成占空比和频率相同的两路相移 PWM 输出。通过更新定义的宏指令，可自定义频率、占空比和相位值。默认设置为 10kHz 频率、50%占空比和 30%相移。eCAP2 输出比 eCAP1 提前 30%，GPIO5 和 GPIO6 作为 eCAP1/2 的输出端口，可通过示波器/CRO 探针观察波形。

### 16.8.1.4 eCAP 软件同步示例

文件： ecap_ex4_sw_sync .c

本示例配置 ePWM3A 为：

- 上升计数模式
- 周期起始值为 500，递增至 8000
- PRD 输出切换

eCAP1、eCAP2 和 eCAP3 被配置为捕获 ePWM3A 输出信号的上升沿与下降沿之间的时间间隔。

外部连接

- eCAP1、eCAP2、eCAP3 连接至 GPIO16

- ePWM3A 连接至 GPIO4
- 将 GPIO4 连接至 GPIO16

观察变量：

- ecapPassCount - 成功捕获次数
- ecap3IntCount - 中断计数

# 16.9 寄存器描述

## 16.9.1 eCAP 基地址表

表中列出了 eCAP 基地址。

表 16-2 eCAP 基地址表

| 位域名称 | | DriverLib 名称 | 基地址 | 流水线保护 |
|---|---|---|---|---|
| 实例 | 结构 | | | |
| ECap1Regs | ECAP_REGS | ECAP1_BASE | 0x0061_A000 | 是 |
| ECap2Regs | ECAP_REGS | ECAP2_BASE | 0x0061_A080 | 是 |
| ECap3Regs | ECAP_REGS | ECAP3_BASE | 0x0061_A100 | 是 |

## 16.9.2 ECAP_REGS 寄存器

表中列出了 ECAP_REGS 寄存器的内存映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，不应修改寄存器内容。

表 16-3 ECAP_REGS 寄存器

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 0x0000 | TSCTR | 时间戳计数器 |
| 0x0004 | CTRPHS | 计数器相位偏移值寄存器 |
| 0x0008 | CAP1 | 捕获 1 寄存器 |
| 0x000C | CAP2 | 捕获 2 寄存器 |
| 0x0010 | CAP3 | 捕获 3 寄存器 |
| 0x0014 | CAP4 | 捕获 4 寄存器 |
| 0x0024 | ECCTL0 | 捕获控制寄存器 0 |
| 0x0028 | ECCTL1 | 捕获控制寄存器 1 |
| 0x002A | ECCTL2 | 捕获控制寄存器 2 |
| 0x002C | ECEINT | 捕获中断使能寄存器 |
| 0x002E | ECFLG | 捕获中断标志寄存器 |
| 0x0030 | ECCLR | 捕获中断清除寄存器 |
| 0x0032 | ECFRC | 捕获中断强制寄存器 |

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 0x003C | ECAPSYNCINSEL | 同步源选择寄存器 |

复杂的位访问类型经过编码以适应小型表格单元格。表中显示了本节中用于访问类型的编码。

表 16-4 ECAP_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R-0 | 读 返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1S | W1S | 写 1 置位 |
| WSonce | WSonce | 一次性写（仅能置位一次） |
| 复位或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i,k,l,m,n | | 当这些变量用于寄存器名称、偏移地址或地址时，指代寄存器数组的值，该寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组以公式表示。 |
| y | | 当该变量用于寄存器名称、偏移地址或地址时，指代寄存器数组的值。 |

### 16.9.2.1 TSCTR 寄存器

偏移地址：0h 复位值：00000000h

时间戳计数器

表 16-5 TSCTR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TSCTR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TSCTR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TSCTR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSCTR | | | | | | | |
| R/W-0h | | | | | | | |

表 16-6 TSCTR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | TSCTR | R/W | 0h | 用作捕获时基的 32 位活动计数器寄存器。<br>HR 模式下：<br>1）读取此寄存器得到 HRCOUNTER 的值，不可写入。<br>2）可通过 CTRFILTRESET 复位。<br>3）未同步到 SYSCLK 时钟域，因此读取值可能不准确。<br>复位类型：SYSRSn |

### 16.9.2.2 CTRPHS 寄存器

偏移地址：4h　复位值：00000000h

计数器相位偏移值寄存器

表 16-7 CTRPHS 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CTRPHS | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CTRPHS | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CTRPHS | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CTRPHS | | | | | | | |
| R/W-0h | | | | | | | |

表 16-8 CTRPHS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | CTRPHS | R/W | 0h | 计数器相位值寄存器，可设置相位滞后或超前。发生 SYNCI 事件或通过控制位进行软件强制时，将 CTRPHS 装载至 TSCTR，用于相对于其他 eCAP 和 EPWM 时基实现相位控制同步。此寄存器不适用于 HR 模式。<br>复位类型：SYSRSn |

### 16.9.2.3 CAP1 寄存器

偏移地址：8h　复位值：00000000h

捕获 1 寄存器

表 16-9 CAP1 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAP1 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CAP1 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CAP1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CAP1 | | | | | | | |
| R/W-0h | | | | | | | |

表 16-10 CAP1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | CAP1 | R/W | 0h | 此寄存器可由以下方式装载（写入）：<br>捕获事件发生时的时间戳计数器 TSCTR 值。<br>软件写入，可用于测试或初始化。<br>APWM 模式下的 ARPD 影子寄存器 CAP3。<br>复位类型：SYSRSn |

### 16.9.2.4 CAP2 寄存器

偏移地址：Ch　复位值：00000000h

捕获 2 寄存器

表 16-11 CAP2 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAP2 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CAP2 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CAP2 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CAP2 | | | | | | | |
| R/W-0h | | | | | | | |

表 16-12 CAP2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | CAP2 | R/W | 0h | 此寄存器可由以下方式装载（写入）：<br>捕获事件发生时的时间戳（计数器值）。<br>软件写入，可用于测试。<br>APWM 模式下的 ACMP 影子寄存器 CAP4。<br>复位类型：SYSRSn |

### 16.9.2.5 CAP3 寄存器

偏移地址：10h　复位值：00000000h

捕获 3 寄存器

表 16-13 CAP3 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAP3 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CAP3 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CAP3 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CAP3 | | | | | | | |
| R/W-0h | | | | | | | |

表 16-14 CAP3 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | CAP3 | R/W | 0h | CMP 模式下，此寄存器是时间戳捕获寄存器。<br>APWM 模式下，此寄存器是周期影子寄存器 APRD，可通过它更新 PWM 周期值；此时 CAP3（APRD）是 CAP1 的影子寄存器。<br>复位类型：SYSRSn |

### 16.9.2.6 CAP4 寄存器

偏移地址：14h　复位值：00000000h

捕获 4 寄存器

表 16-15 CAP4 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CAP4 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CAP4 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CAP4 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CAP4 | | | | | | | |
| R/W-0h | | | | | | | |

表 16-16 CAP4 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | CAP4 | R/W | 0h | CMP 模式下，此寄存器是时间戳捕获寄存器。<br>APWM 模式下，此寄存器是比较影子寄存器 ACMP，可通过它更新 PWM 比较值；此时 CAP4（ACMP）是 CAP2 的影子寄存器。<br>复位类型：SYSRSn |

### 16.9.2.7 ECCTL0 寄存器

偏移地址：24h　复位值：0000007Fh

捕获控制寄存器 0

表 16-17 ECCTL0 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | INPUTSEL | | | | | | |
| R-0-0h | R/W-7Fh | | | | | | |

表 16-18 ECCTL0 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-7 | RESERVED | R-0 | 0h | 保留 |
| 6-0 | INPUTSEL | R/W | 7Fh | 捕获输入源选择位。<br>0000000：捕获输入为 ECAPxINPUT[0]。<br>0000001：捕获输入为 ECAPxINPUT[1]。<br>0000010：捕获输入为 ECAPxINPUT[2]。<br>…<br>1111111：捕获输入为 ECAPxINPUT[127]。<br>复位类型：CPU1.SYSRSn |

### 16.9.2.8 ECCTL1 寄存器

偏移地址：28h　复位值：0000h

捕获控制寄存器 1

表 16-19 ECCTL1 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FREE_SOFT | | PRESCALE | | | | | CAPLDEN |
| R/W-0h | | R/W-0h | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CTRRST4 | CAP4POL | CTRRST3 | CAP3POL | CTRRST2 | CAP2POL | CTRRST1 | CAP1POL |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 16-20 ECCTL1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | FREE_SOFT | R/W | 0h | 仿真控制。<br>0h (R/W) = 仿真暂停时 TSCTR 计数器立即停止。<br>1h (R/W) = TSCTR 计数器运行直到等于 0。<br>2h (R/W) = TSCTR 计数器不受仿真暂停影响（自由运行）。<br>3h (R/W) = TSCTR 计数器不受仿真暂停影响（自由运行）。<br>复位类型：SYSRSn |
| 13-9 | PRESCALE | R/W | 0h | 事件滤波器预分频选择。<br>0h (R/W) = 1 分频（不预分频，旁路预分频器）。<br>1h (R/W) = 2 分频。<br>2h (R/W) = 4 分频。<br>3h (R/W) = 6 分频。<br>4h (R/W) = 8 分频。<br>5h (R/W) = 10 分频。<br>1Eh (R/W) = 60 分频。<br>1Fh (R/W) = 62 分频。<br>复位类型：SYSRSn |
| 8 | CAPLDEN | R/W | 0h | 使能在捕获事件发生时装载 CAP1–4 寄存器。<br>注意：此位不会禁止生成 CEVTn 事件。<br>0h (R/W) = 禁用捕获事件发生时的 CAP1–4 寄存器装载。<br>1h (R/W) = 使能捕获事件发生时的 CAP1–4 寄存器装载。<br>复位类型：SYSRSn |
| 7 | CTRRST4 | R/W | 0h | 捕获事件 4 的计数器复位控制。<br>0h (R/W) = 捕获事件 4 不复位计数器（绝对时间戳操作）。<br>1h (R/W) = 捕获事件 4 的时间戳后复位计数器（用于差分模式）。<br>复位类型：SYSRSn |
| 6 | CAP4POL | R/W | 0h | 捕获事件 4 极性选择。<br>0h (R/W) = 在上升沿（RE）触发捕获事件 4。<br>1h (R/W) = 在下降沿（FE）触发捕获事件 4。<br>复位类型：SYSRSn |
| 5 | CTRRST3 | R/W | 0h | 捕获事件 3 的计数器复位控制。<br>0h (R/W) = 捕获事件 3 不复位计数器（绝对时间戳操作）。<br>1h (R/W) = 捕获事件 3 的时间戳后复位计数器 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | （用于差分模式）。<br>复位类型：SYSRSn |
| 4 | CAP3POL | R/W | 0h | 捕获事件 3 极性选择。<br>0h (R/W) = 在上升沿（RE）触发捕获事件 3。<br>1h (R/W) = 在下降沿（FE）触发捕获事件 3。<br>复位类型：SYSRSn |
| 3 | CTRRST2 | R/W | 0h | 捕获事件 2 的计数器复位控制。<br>0h (R/W) = 捕获事件 2 不复位计数器（绝对时间戳操作）。<br>1h (R/W) = 捕获事件 2 的时间戳后复位计数器（用于差分模式）。<br>复位类型：SYSRSn |
| 2 | CAP2POL | R/W | 0h | 捕获事件 2 极性选择。<br>0h (R/W) = 在上升沿（RE）触发捕获事件 2。<br>1h (R/W) = 在下降沿（FE）触发捕获事件 2。<br>复位类型：SYSRSn |
| 1 | CTRRST1 | R/W | 0h | 捕获事件 1 的计数器复位控制。<br>0h (R/W) = 捕获事件 1 不复位计数器（绝对时间戳操作）。<br>1h (R/W) = 捕获事件 1 的时间戳后复位计数器（用于差分模式）。<br>复位类型：SYSRSn |
| 0 | CAP1POL | R/W | 0h | 捕获事件 1 极性选择。<br>0h (R/W) = 在上升沿（RE）触发捕获事件 1。<br>1h (R/W) = 在下降沿（FE）触发捕获事件 1。<br>复位类型：SYSRSn |

### 16.9.2.9 ECCTL2 寄存器

偏移地址：2Ah　复位值：0006h

捕获控制寄存器 2

表 16-21 ECCTL2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| MODCNTRSTS | | DMAEVTSEL | | CTRFILTRESET | APWMPOL | CAP_APWM | SWSYNC |
| R-0h | | R/W-0h | | R-0/W1C-0h | R/W-0h | R/W-0h | R-0/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SYNCO_SEL | | SYNCI_EN | TSCTRSTOP | REARM | STOP_WRAP | | CONT_ONESHT |
| R/W-0h | | R/W-0h | R/W-0h | R-0/W1S-0h | R/W-3h | | R/W-0h |

表 16-22 ECCTL2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | MODCNTRSTS | R | 0h | 读取模计数器当前状态。<br>00b (R) = 下一个捕获事件装载 CAP1。<br>01b (R) = 下一个捕获事件装载 CAP2。<br>10b (R) = 下一个捕获事件装载 CAP3。<br>11b (R) = 下一个捕获事件装载 CAP4。<br>复位类型：CPU1.SYSRSn |
| 13-12 | DMAEVTSEL | R/W | 0h | DMA 事件选择。<br>00b (R/W) = DMA 中断源为 CEVT1。<br>01b (R/W) = DMA 中断源为 CEVT2。<br>10b (R/W) = DMA 中断源为 CEVT3。<br>11b (R/W) = DMA 中断源为 CEVT4。<br>注意：还必须将 ECCTL1.CAPLDEN 置 1，才能产生 ECAPxDMA_INT。<br>复位类型：CPU1.SYSRSn |
| 11 | CTRFILTRESET | R-0/W1C | 0h | 复位位。<br>0h (R) = 无效。<br>1h (W) = 复位事件滤波器、计数器、模计数器，以及 CEVT[1,2,3,4]、CNTOVF、HRERROR 标志。<br>注意：若配置 ECAP 时捕获到杂散输入，可使用此位使捕获模块从已知状态启动。<br>复位类型：CPU1.SYSRSn |
| 10 | APWMPOL | R/W | 0h | APWM 输出极性选择，仅适用于 APWM 工作模式。<br>0h (R/W) = 输出高电平有效，比较值定义高电平时间。<br>1h (R/W) = 输出低电平有效，比较值定义低电平时间。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 9 | CAP_APWM | R/W | 0h | CAP/APWM 工作模式选择。<br>0h (R/W) = ECAP 工作于捕获模式，强制采用以下配置：禁止通过 CTR = PRD 事件复位 TSCTR；禁止 CAP1 和 CAP2 的影子装载；允许用户使能 CAP1–4 装载；CAPx/APWMx 引脚作为捕获输入。<br>1h (R/W) = ECAP 工作于 APWM 模式，强制采用以下配置：CTR = PRD 事件（周期边界）复位 TSCTR；允许 CAP1 和 CAP2 的影子装载；禁止将时间戳装载至 CAP1–4；CAPx/APWMx 引脚作为 APWM 输出。<br>复位类型：SYSRSn |
| 8 | SWSYNC | R-0/W1S | 0h | 软件强制计数器 TSCTR 同步。允许通过软件产生同步脉冲；APWM 模式下也可由 CTR = PRD 事件提供同步脉冲。<br>0h (R/W) = 写 0 无效，读取始终返回 0。<br>1h (R/W) = 若 SYNCO_SEL 位为 0,0，则强制当前 ECAP 模块及下游 ECAP 模块执行 TSCTR 影子装载。写 1 后，此位返回 0。<br>复位类型：SYSRSn |
| 7-6 | SYNCO_SEL | R/W | 0h | 同步输出选择。<br>0h (R/W) = 同步输出信号为 SWSYNC。<br>1h (R/W) = 选择 CTR = PRD 事件作为同步输出；此选择仅在 APWM 模式下有意义。<br>2h (R/W) = 禁用同步输出。<br>3h (R/W) = 禁用同步输出。<br>复位类型：SYSRSn |
| 5 | SYNCI_EN | R/W | 0h | 计数器 TSCTR 同步输入选择模式。<br>0h (R/W) = 禁用同步输入。<br>1h (R/W) = 使能在 SYNCI 信号或软件强制事件发生时，将 CTRPHS 装载至 TSCTR。<br>复位类型：SYSRSn |
| 4 | TSCTRSTOP | R/W | 0h | 时间戳计数器 TSCTR 停止（冻结）控制。<br>0h (R/W) = TSCTR 停止。<br>1h (R/W) = TSCTR 自由运行。<br>复位类型：SYSRSn |
| 3 | REARM | R-0/W1S | 0h | 重新准备控制。此功能在单次和连续模式下均有效。<br>0h (R/W) = 无效，读取始终返回 0。<br>1h (R/W) = 按以下步骤准备单次序列：1）将 Mod4 计数器复位为零；2）解除 Mod4 计数器冻结；3）使能捕获寄存器装载。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 2-1 | STOP_WRAP | R/W | 3h | 单次模式的停止值：指定冻结 CAP(1–4) 寄存器、停止捕获序列前允许发生的捕获次数（1–4）。连续模式的回绕值：指定循环缓冲区在哪个捕获寄存器（1–4）处回绕并重新开始。<br>注意：STOP_WRAP 与 Mod4 计数器比较；相等时停止（冻结）Mod4 计数器，并禁止捕获寄存器装载。单次模式下，在重新准备前阻止后续中断事件。<br>0h (R/W) = 单次模式在捕获事件 1 后停止，连续模式在事件 1 后回绕。<br>1h (R/W) = 单次模式在捕获事件 2 后停止，连续模式在事件 2 后回绕。<br>2h (R/W) = 单次模式在捕获事件 3 后停止，连续模式在事件 3 后回绕。<br>3h (R/W) = 单次模式在捕获事件 4 后停止，连续模式在事件 4 后回绕。<br>复位类型：SYSRSn |
| 0 | CONT_ONESHT | R/W | 0h | 连续 / 单次模式控制，仅适用于捕获模式。<br>0h (R/W) = 连续模式。<br>1h (R/W) = 单次模式。<br>复位类型：SYSRSn |

### 16.9.2.10 ECEINT 寄存器

偏移地址：2Ch　复位值：0000h

捕获中断使能寄存器

中断使能位（CEVT1 等）可阻止所选事件产生中断。事件仍会锁存到 ECFLG 标志位，并可通过 ECFRC/ECCLR 强制置位或清除。

配置外设模式和中断的正确顺序如下：

1. 禁用全局中断。
2. 停止 eCAP 计数器。
3. 禁用 eCAP 中断。
4. 配置外设寄存器。
5. 清除杂散 eCAP 中断标志。
6. 使能 eCAP 中断。

7. 启动 eCAP 计数器。

8. 使能全局中断。

表 16-23 ECEINT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | 保留 |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CTR_EQ_CMP | CTR_EQ_PRD | CTROVF | CEVT4 | CEVT3 | CEVT2 | CEVT1 | 保留 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

表 16-24 ECEINT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-9 | RESERVED | R | 0h | 保留 |
| 8 | RESERVED | R/W | 0h | 保留 |
| 7 | CTR_EQ_CMP | R/W | 0h | 计数器等于比较值中断使能。<br>0h (R/W) = 禁用计数器等于比较值作为中断源。<br>1h (R/W) = 使能计数器等于比较值作为中断源。<br>复位类型：SYSRSn |
| 6 | CTR_EQ_PRD | R/W | 0h | 计数器等于周期中断使能。<br>0h (R/W) = 禁用计数器等于周期作为中断源。<br>1h (R/W) = 使能计数器等于周期作为中断源。<br>复位类型：SYSRSn |
| 5 | CTROVF | R/W | 0h | 计数器溢出中断使能。<br>0h (R/W) = 禁用计数器溢出作为中断源。<br>1h (R/W) = 使能计数器溢出作为中断源。<br>复位类型：SYSRSn |
| 4 | CEVT4 | R/W | 0h | 捕获事件 4中断使能。<br>0h (R/W) = 禁用捕获事件 4作为中断源。<br>1h (R/W) = 使能捕获事件 4作为中断源。<br>复位类型：SYSRSn |
| 3 | CEVT3 | R/W | 0h | 捕获事件 3中断使能。<br>0h (R/W) = 禁用捕获事件 3作为中断源。<br>1h (R/W) = 使能捕获事件 3作为中断源。<br>复位类型：SYSRSn |
| 2 | CEVT2 | R/W | 0h | 捕获事件 2中断使能。<br>0h (R/W) = 禁用捕获事件 2作为中断源。<br>1h (R/W) = 使能捕获事件 2作为中断源。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | CEVT1 | R/W | 0h | 捕获事件 1中断使能。<br>0h (R/W) = 禁用捕获事件 1作为中断源。<br>1h (R/W) = 使能捕获事件 1作为中断源。<br>复位类型：SYSRSn |
| 0 | RESERVED | R | 0h | 保留 |

### 16.9.2.11 ECFLG 寄存器

偏移地址：2Eh 复位值：0000h

捕获中断标志寄存器

表 16-25 ECFLG 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | 保留 |
| R-0h | | | | | | | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CTR_CMP | CTR_PRD | CTROVF | CEVT4 | CEVT3 | CEVT2 | CEVT1 | INT |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 16-26 ECFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-9 | RESERVED | R | 0h | 保留 |
| 8 | RESERVED | R | 0h | 保留 |
| 7 | CTR_CMP | R | 0h | 计数器等于比较值状态标志。仅在 APWM 模式下有效。<br>0h (R/W) = 未发生事件。<br>1h (R/W) = TSCTR 达到比较寄存器 ACMP 的值。<br>复位类型：SYSRSn |
| 6 | CTR_PRD | R | 0h | 计数器等于周期状态标志。仅在 APWM 模式下有效。<br>0h (R/W) = 未发生事件。<br>1h (R/W) = TSCTR 达到周期寄存器 APRD 的值并已复位。<br>复位类型：SYSRSn |
| 5 | CTROVF | R | 0h | 计数器溢出状态标志。在 CAP 和 APWM 模式下均有效。<br>0h (R/W) = 未发生事件。<br>1h (R/W) = TSCTR 从 FFFFFFFF 变为 00000000。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 4 | CEVT4 | R | 0h | 捕获事件 4状态标志。仅在 CAP 模式下有效。<br>0h (R/W) = 未发生事件。<br>1h (R/W) = ECAPx 引脚发生第 4 个事件。<br>复位类型：SYSRSn |
| 3 | CEVT3 | R | 0h | 捕获事件 3状态标志。仅在 CAP 模式下有效。<br>0h (R/W) = 未发生事件。<br>1h (R/W) = ECAPx 引脚发生第 3 个事件。<br>复位类型：SYSRSn |
| 2 | CEVT2 | R | 0h | 捕获事件 2状态标志。仅在 CAP 模式下有效。<br>0h (R/W) = 未发生事件。<br>1h (R/W) = ECAPx 引脚发生第 2 个事件。<br>复位类型：SYSRSn |
| 1 | CEVT1 | R | 0h | 捕获事件 1状态标志。仅在 CAP 模式下有效。<br>0h (R/W) = 未发生事件。<br>1h (R/W) = ECAPx 引脚发生第 1 个事件。<br>复位类型：SYSRSn |
| 0 | INT | R | 0h | 全局中断状态标志。<br>0h (R/W) = 未发生事件。<br>1h (R/W) = 已产生中断。<br>复位类型：SYSRSn |

### 16.9.2.12 ECCLR 寄存器

偏移地址：30h 复位值：0000h

捕获中断清除寄存器

表 16-27 ECCLR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | 保留 |
| R-0h | | | | | | | R-0/W1C-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CTR_CMP | CTR_PRD | CTROVF | CEVT4 | CEVT3 | CEVT2 | CEVT1 | INT |
| R-0/W1C-0h | R-0/W1C-0h | R-0/W1C-0h | R-0/W1C-0h | R-0/W1C-0h | R-0/W1C-0h | R-0/W1C-0h | R-0/W1C-0h |

表 16-28 ECCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-9 | RESERVED | R | 0h | 保留 |
| 8 | RESERVED | R-0/W1C | 0h | 保留 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 7 | CTR_CMP | R-0/W1C | 0h | 计数器等于比较值状态清除。<br>0h (R/W) = 写 0 无效，读取始终返回 0。<br>1h (R/W) = 写 1 清除 CTR=CMP 标志。<br>复位类型：SYSRSn |
| 6 | CTR_PRD | R-0/W1C | 0h | 计数器等于周期状态清除。<br>0h (R/W) = 写 0 无效，读取始终返回 0。<br>1h (R/W) = 写 1 清除 CTR=PRD 标志。<br>复位类型：SYSRSn |
| 5 | CTROVF | R-0/W1C | 0h | 计数器溢出状态清除。<br>0h (R/W) = 写 0 无效，读取始终返回 0。<br>1h (R/W) = 写 1 清除 CTROVF 标志。<br>复位类型：SYSRSn |
| 4 | CEVT4 | R-0/W1C | 0h | 捕获事件 4状态清除。<br>0h (R/W) = 写 0 无效，读取始终返回 0。<br>1h (R/W) = 写 1 清除 CEVT4 标志。<br>复位类型：SYSRSn |
| 3 | CEVT3 | R-0/W1C | 0h | 捕获事件 3状态清除。<br>0h (R/W) = 写 0 无效，读取始终返回 0。<br>1h (R/W) = 写 1 清除 CEVT3 标志。<br>复位类型：SYSRSn |
| 2 | CEVT2 | R-0/W1C | 0h | 捕获事件 2状态清除。<br>0h (R/W) = 写 0 无效，读取始终返回 0。<br>1h (R/W) = 写 1 清除 CEVT2 标志。<br>复位类型：SYSRSn |
| 1 | CEVT1 | R-0/W1C | 0h | 捕获事件 1状态清除。<br>0h (R/W) = 写 0 无效，读取始终返回 0。<br>1h (R/W) = 写 1 清除 CEVT1 标志。<br>复位类型：SYSRSn |
| 0 | INT | R-0/W1C | 0h | ECAP 全局中断状态清除。<br>0h (R/W) = 写 0 无效，读取始终返回 0。<br>1h (R/W) = 写 1 清除 INT 标志；若任一事件标志为 1，允许继续产生中断。<br>复位类型：SYSRSn |

## 16.9.2.13 ECFRC 寄存器

偏移地址：32h 复位值：0000h

捕获中断强制寄存器

表 16-29 ECFRC 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | 保留 |
| R-0h | | | | | | | R-0/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CTR_CMP | CTR_PRD | CTROVF | CEVT4 | CEVT3 | CEVT2 | CEVT1 | 保留 |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0h |

表 16-30 ECFRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-9 | RESERVED | R | 0h | 保留 |
| 8 | RESERVED | R-0/W1S | 0h | 保留 |
| 7 | CTR_CMP | R-0/W1S | 0h | 强制计数器等于比较值中断。仅在 APWM 模式下有效。<br>0h (R/W) = 无效，读取始终返回 0。<br>1h (R/W) = 置位 CTR_CMP 标志。<br>复位类型：SYSRSn |
| 6 | CTR_PRD | R-0/W1S | 0h | 强制计数器等于周期中断。仅在 APWM 模式下有效。<br>0h (R/W) = 无效，读取始终返回 0。<br>1h (R/W) = 置位 CTR_PRD 标志。<br>复位类型：SYSRSn |
| 5 | CTROVF | R-0/W1S | 0h | 强制计数器溢出。<br>0h (R/W) = 无效，读取始终返回 0。<br>1h (R/W) = 置位 CTROVF 标志。<br>复位类型：SYSRSn |
| 4 | CEVT4 | R-0/W1S | 0h | 强制捕获事件 4。仅在 CAP 模式下有效。<br>0h (R/W) = 无效，读取始终返回 0。<br>1h (R/W) = 置位 CEVT4 标志。<br>复位类型：SYSRSn |
| 3 | CEVT3 | R-0/W1S | 0h | 强制捕获事件 3。仅在 CAP 模式下有效。<br>0h (R/W) = 无效，读取始终返回 0。<br>1h (R/W) = 置位 CEVT3 标志。<br>复位类型：SYSRSn |
| 2 | CEVT2 | R-0/W1S | 0h | 强制捕获事件 2。仅在 CAP 模式下有效。<br>0h (R/W) = 无效，读取始终返回 0。<br>1h (R/W) = 置位 CEVT2 标志。<br>复位类型：SYSRSn |
| 1 | CEVT1 | R-0/W1S | 0h | 强制捕获事件 1。仅在 CAP 模式下有效。<br>0h (R/W) = 无效，读取始终返回 0。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1h (R/W) = 置位 CEVT1 标志。<br>复位类型：SYSRSn |
| 0 | RESERVED | R | 0h | 保留 |

## 16.9.2.14 ECAPSYNCINSEL 寄存器

偏移地址：3Ch 复位值：00000001h

SYNC 源选择寄存器

表 16-31 ECAPSYNCINSEL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | SEL | | | | |
| R-0h | | | R/W-1h | | | | |

表 16-32 ECAPSYNCINSEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-5 | RESERVED | R | 0h | 保留 |
| 4-0 | SEL | R/W | 1h | 选择 SYNCIN 信号源。<br>0h (R/W) = 禁用送往 eCAP 的 Syncin<br>1h (R/W) = EPWM1SYNCOUT<br>2h (R/W) = EPWM2SYNCOUT<br>3h (R/W) = EPWM3SYNCOUT<br>4h (R/W) = EPWM4SYNCOUT<br>5h (R/W) = EPWM5SYNCOUT<br>6h (R/W) = EPWM6SYNCOUT<br>7h (R/W) = EPWM7SYNCOUT<br>8h (R/W) = 保留（RSVD）<br>9h (R/W) = 保留（RSVD）<br>Ah (R/W) = 保留（RSVD）<br>Bh (R/W) = 保留（RSVD）<br>Ch (R/W) = 保留（RSVD）<br>Dh (R/W) = 保留（RSVD） |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | Eh (R/W) = 保留（RSVD）<br>Fh (R/W) = 保留（RSVD）<br>10h (R/W) = 保留（RSVD）<br>11h (R/W) = ECAP1SYNCOUT<br>12h (R/W) = ECAP2SYNCOUT<br>13h (R/W) = ECAP3SYNCOUT<br>14h (R/W) = 保留（RSVD）<br>15h (R/W) = 保留（RSVD）<br>16h (R/W) = 保留（RSVD）<br>17h (R/W) = 保留（RSVD）<br>18h (R/W) = INPUTXBAROUT5<br>19h (R/W) = INPUTXBAROUT6<br>1Ah (R/W) = 保留（RSVD）<br>1Bh (R/W) = 保留（RSVD）<br>1Ch (R/W) = 保留（RSVD）<br>1Dh (R/W) = 保留（RSVD）<br>1Eh (R/W) = 保留（RSVD）<br>1Fh (R/W) = 保留（RSVD）<br>复位类型：SYSRSn |

### 16.9.3 ECAP 寄存器与 DriverLib 函数对应表

表 16-33 ECAP 寄存器与 DriverLib 函数对应表

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| TSCTR | ecap.h | ECAP_getTimeBaseCounter |
| CTRPHS | ecap.h | ECAP_setPhaseShiftCount |
| CAP1 | ecap.h | ECAP_setAPWMPeriod |
| CAP1 | ecap.h | ECAP_getEventTimeStamp |
| CAP2 | ecap.h | ECAP_setAPWMCompare |
| CAP2 | ecap.h | ECAP_getEventTimeStamp |
| CAP3 | ecap.h | ECAP_setAPWMShadowPeriod |
| CAP3 | ecap.h | ECAP_getEventTimeStamp |
| CAP4 | ecap.h | ECAP_setAPWMShadowCompare |
| CAP4 | ecap.h | ECAP_getEventTimeStamp |
| ECCTL0 | ecap.h | ECAP_selectECAPInput |
| ECCTL1 | ecap.c | ECAP_setEmulationMode |
| ECCTL1 | ecap.h | ECAP_setEventPrescaler |
| ECCTL1 | ecap.h | ECAP_setEventPolarity |
| ECCTL1 | ecap.h | ECAP_enableCounterResetOnEvent |
| ECCTL1 | ecap.h | ECAP_disableCounterResetOnEvent |
| ECCTL1 | ecap.h | ECAP_enableTimeStampCapture |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| ECCTL1 | ecap.h | ECAP_disableTimeStampCapture |
| ECCTL2 | ecap.h | ECAP_setCaptureMode |
| ECCTL2 | ecap.h | ECAP_reArm |
| ECCTL2 | ecap.h | ECAP_enableCaptureMode |
| ECCTL2 | ecap.h | ECAP_enableAPWMMode |
| ECCTL2 | ecap.h | ECAP_enableLoadCounter |
| ECCTL2 | ecap.h | ECAP_disableLoadCounter |
| ECCTL2 | ecap.h | ECAP_loadCounter |
| ECCTL2 | ecap.h | ECAP_setSyncOutMode |
| ECCTL2 | ecap.h | ECAP_stopCounter |
| ECCTL2 | ecap.h | ECAP_startCounter |
| ECCTL2 | ecap.h | ECAP_setAPWMPolarity |
| ECCTL2 | ecap.h | ECAP_resetCounters |
| ECCTL2 | ecap.h | ECAP_setDMASource |
| ECCTL2 | ecap.h | ECAP_getModuloCounterStatus |
| ECEINT | ecap.h | ECAP_enableInterrupt |
| ECEINT | ecap.h | ECAP_disableInterrupt |
| ECFLG | ecap.h | ECAP_getInterruptSource |
| ECFLG | ecap.h | ECAP_getGlobalInterruptStatus |
| ECCLR | ecap.h | ECAP_clearInterrupt |
| ECCLR | ecap.h | ECAP_clearGlobalInterrupt |
| ECFRC | ecap.h | ECAP_forceInterrupt |
| SYNCINSEL | ecap.h | ECAP_setSyncInPulseSource |

# 17 高分辨率捕捉（HRCAP）

## 17.1 简介

本章阐述高分辨率捕获（HRCAP）模块的工作原理。本文所述的 HRCAP 子模块属于 Type1 eCAP 系统。 HRCAP 模块对外部脉冲宽度的测量精度高于 eCAP 模块。

HRCAP 模块的用途包括：

- 电容式触摸应用
- 脉冲序列周期与占空比的高分辨率测量
- 瞬时速度测量
- 瞬时频率测量
- 隔离边界电压测量
- 距离/声呐测量与扫描
- 流量测量

### 17.1.1 功能

HRCAP 模块具备以下功能：

- 支持非高分辨率或高分辨率模式下的脉冲宽度捕获
- 绝对模式脉冲宽度捕获
- 连续或单次捕获
- 支持下降沿或上升沿中断
- 4 深度缓冲区的连续脉冲宽度捕获
- 用于高精度捕获的硬件校准逻辑

所有先前的资源均可通过输入 X-BAR 在任意引脚上使用。

### 17.1.2 描述

相较于 Type0 HRCAP 的改进包括：

- 简化校准方案：

- HRCAP 始终处于运行状态，无需离线校准
- 校准过程始终在后台运行，大幅降低软件校准开销
- 计算小数部分的软件开销降低
- 整数部分和小数部分统一采用 32 位编码
- 启用 HRCAP 增强功能时可访问所有 eCAP 硬件
- HRCAP 使用方式现已与 eCAP 统一

eCAP3 新增了 HRCAP 增强功能，支持相对于 SYSCLK 异步的信号捕获。每个 HRCAP 子模块除硬件校准模块外，还包含一个捕获通道。启用 HRCAP 增强功能时，所有 eCAP 硬件均可访问；但事件过滤器和输入限定器无效，因其与 SYSCLK 同步。

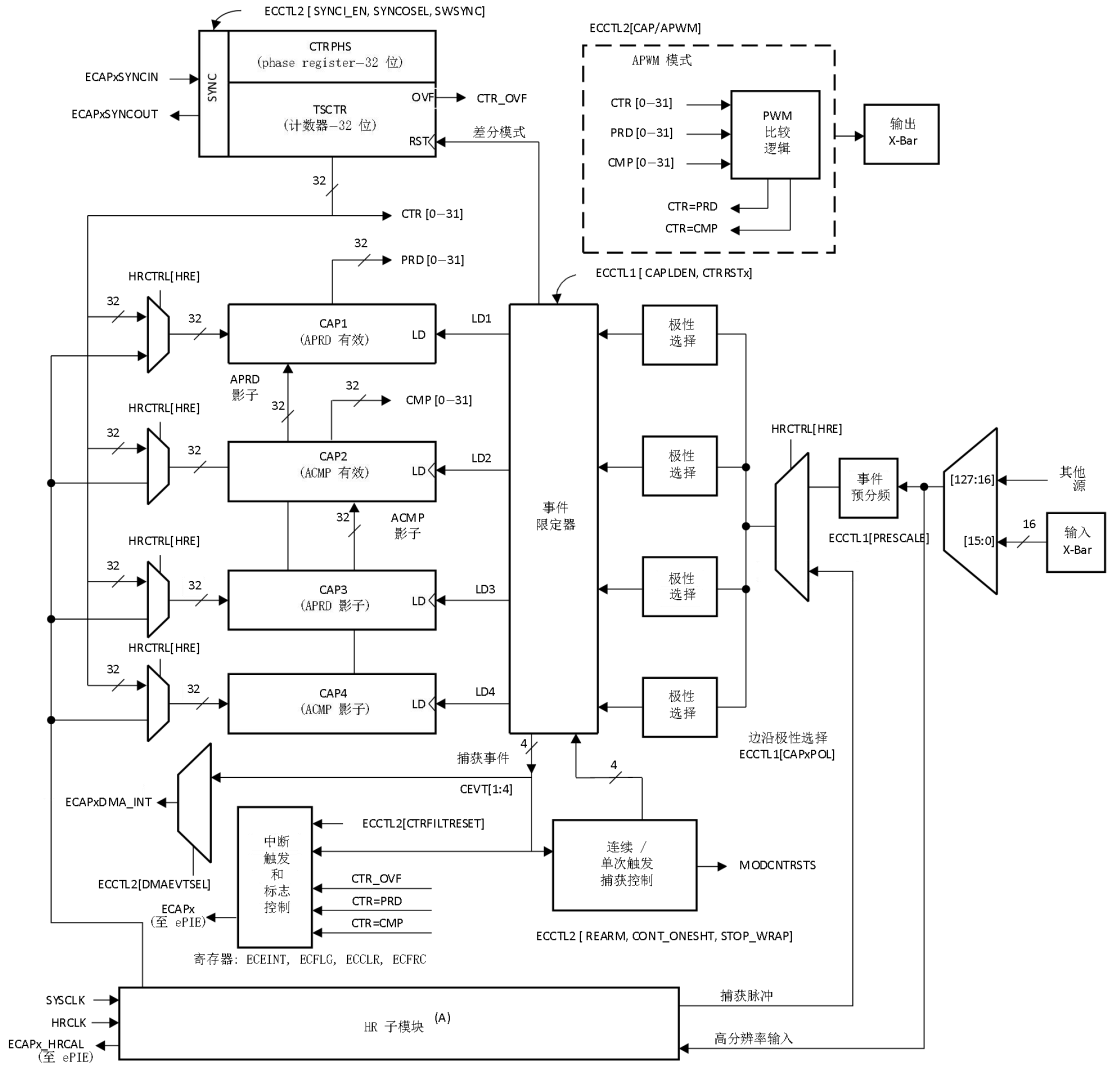
每个 HRCAP 通道具备以下独立关键资源：

- 对应 eCAP 的全部硬件
- 高分辨率校准逻辑
- 专用校准中断

## 17.2 操作细节

图中展示了实现 eCAP 模块高分辨率捕获功能的各个组件。需注意的是，现有 eCAP 资源会被重复使用，因此在启用 HRCAP 增强功能前必须先配置好 eCAP 模块。为简化操作，建议采用绝对时间戳测量。

所有 HRCAP 测量均为相对时间测量，以最小步长为单位。系统已提供校准硬件和软件功能，可将相对时间测量转换为秒制时间测量。仅当需要时间转换测量时，才需使用校准硬件和软件。




A. 并非所有 eCAP 模块都提供 HRCAP 子模块；不提供时，高分辨率多路选择器和硬件不会实现

图 17-1 HRCAP 操作框图

## 17.2.1 HRCAP 时钟

与之前的 Type-0 HRCAP 模块不同，具备 HRCAP 功能的 Type-1eCAP 模块无需第二个 PLL 。但该模块仍需 SYSCLK 和第二个异步时钟源 HRCLK。HRCLK 对温度和电压变化敏感，因此在使用时间转换测量时，需定期进行连续校准。

## 17.2.2 HRCAP 初始化序列

以下是 HRCAP 初始化的步骤序列。当使用 HRCAP 进行相对时间测量时，仅需执行步骤 1-5；若需进行时间转换测量，则需完成全部步骤。具体操作如下：

通过 HRCAP_enableHighResolutionClock()启用 HRCLK 功能

延迟 1μs

通过 HRCAP_enableHighResolution()启用高分辨率模式

延迟 1μs

按需配置 eCAP 模块（包括中断设置）

通过 HRCAP_setCalibrationPeriod()设定校准周期

通过 HRCAP_setCalibrationMode()启用持续校准

通过 HRCAP_enableCalibrationInterrupt()启用中断功能

通过 HRCAP_startCalibration()启动校准程序

### 17.2.3 HRCAP 中断

HRCAP 增强功能利用了现有的 eCAP 中断，并额外使用了 hrcalint（该功能仅由硬件校准模块使用）。hrcalint 可由以下条件触发：

sysclkctr=hrcalibperiod；

sysclkctr 或 hrclkctr 发生溢出。

图中展示了该逻辑。

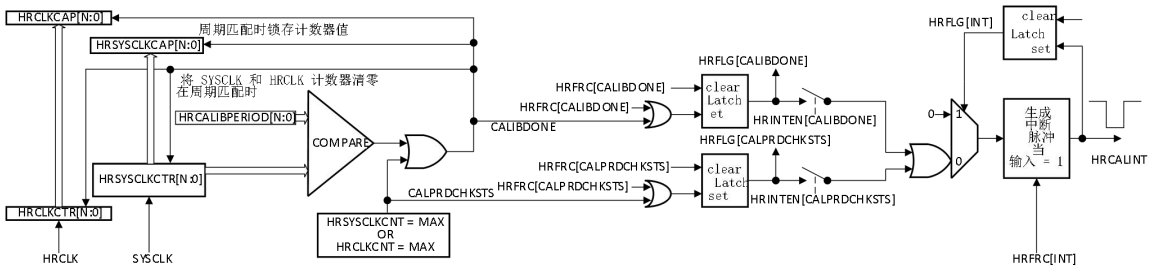



图 17-2 hrcap 校准

### 17.2.4 HRCAP 校准

注意：为使 HRCAP 校准正常工作，系统需在 SYSCLK 频率>100MHz 的条件下运行。

以下章节仅适用于时间转换测量，相对时间测量无需校准。 HRCAP 子模块捕获的所有数值均以 HRCLK 周期为单位。 HRCLK 速度随温度和电压变化显著，因

此需要使用比例因子将捕获值转换为 SYSCLK 域。出于相同原因，需定期重新计算比例因子。 HRCAP 子模块设有校准模块，用于在 HRCLK 与 SYSCLK 之间计算比例因子时降低软件开销。

校准模块包含以下关键资源：

- HRSYSCLKCNT：连接至 SYSCLK 的 32 位计数器，当 calibstart 设置时开始计数。
- HRCLKCNT：连接至 HRCLK 的 32 位计数器，当 calibstart 设置时开始计数。
- HRCALIBPERIOD：校准周期，当 hrsysclkcnt 等于该寄存器值时校准停止。
- HRSYSCLKCAP：在校准周期匹配时，将 hrsysclkcnt 的值捕获至 hrsysclkcap。
- HRCLKCAP：在校准周期匹配时，将 hrclkcnt 的值捕获至 hrclkcap。
- HRCALINT：在校准周期匹配时触发的中断，或当任一计数器寄存器发生溢出时触发。

校准逻辑由两个自由运行的计数器组成：一个由 HRCLK（hrclkctr）驱动，另一个由 SYSCLK（hrsysclkctr）驱动。当 hrsysclkctr 等于 hrcalibperiod 时，校准模块会捕获并重置两个计数器的数值，随后触发中断信号，表明新的比例因子已准备好进行计算。比例因子可通过将 hrsysclkcap 除以 hrclkcap 获得（参见式（17-1））。系统提供了 DriverLib 函数 HRCAP_getScaleFactor 用于确定比例因子，该函数可在校准中断服务例程内部调用。若任一计数器发生溢出，校准周期检查标志位将被设置。校准模块的完整细节详见图。

$$\text{ScaleFactor} = \frac{\text{HRSYSCLKCAP}}{\text{HRCLKCAP}} \qquad (17\text{-}1)$$

注意：

- 即使经过校准，1.2V VDD 电源上的噪声仍会对 HRCAP 子模块的标准偏差产生负面影响。需确保 1.2V 电源清洁，并在使用该子模块时将时钟树的启用与禁用等内部噪声事件降至最低。

- 当 HRCLK > SYSCLK 时，校准会提前终止以避免校准不准确，此时 SYSCLK 必须至少设置为 100MHz。

#### 17.2.4.1 应用比例因子

已提供 DriverLib 函数用于对捕获值应用比例因子，即 HRCAP_getEventTimeStampNanoseconds()。式（17-2）展示了如何在不使用 DriverLib 函数的情况下将原始计数转换为纳秒。

$$\text{Measurement(ns)} = \frac{\text{RawCount} \times \text{scaleFactor}}{128} \times \text{SysClkPrd(ns)} \qquad (17\text{-}2)$$

(1) 尺度因子不会自动应用于捕获的数值。用户需如式（17-2）所示，将该尺度因子应用于所有捕获的数值。

## 17.3 已知例外

HRCAP 模式下：

- 启用或禁用核心时钟会直接影响 HRCAP 子模块的标准差。测量过程中请勿启用或禁用核心时钟。
- TSCTR 不可写入，但可通过 ECCTL2[ctrfiltreset]重置 TSCTR 。
- 使用 HRCAP 增强功能时，输入同步不适用，因为 HRCAP 子模块与 SYSCLK 不同步。
- 事件过滤功能不适用于 HRCAP 模式，这违背了 HRCAP 设计初衷——事件过滤器的输出与 SYSCLK 同步。
- 高分辨率模式建议采用绝对时间模式。若使用时间差模式，当捕获事件重置时间基准计数器时，若未考虑小数部分值可能导致结果不准确。
- 实际捕获值 =（捕获值）–（重置计数器的参考事件的分数值）
- 高频输入信号时，CPU 可能无法跟上捕获速度。此时推荐使用单次模式，该模式允许设备在 CPU 准备就绪前捕获最多四个边沿信号。eCAP 同样适用此模式，但此时可使用事件过滤器降低捕获频率。

## 17.4 软件

### 17.4.1 HRCAP 示例

注：这些示例位于 IDE 安装目录中。

#### 17.4.1.1 HRCAP 捕获与校准示例

文件： hrcap_ex1_capture .c

本示例配置 ECAP 使用 HRCAP 功能来捕获输入GPIO2 信号边沿之间的时间间隔。

外部连接

用户必须向 GPIO2 提供信号。xclkout 已配置为输出 GPIO ，可通过外部跳线实现该功能。具体 GPIO 选择请参阅 Sysconfig 文件。

观察变量：

- onTime1, onTime2
- offTime1, offTime2
- period1, period2

## 17.5 寄存器描述

### 17.5.1 HRCAP 基地址表

表中列出了存储器映射寄存器。未列出的偏移地址应视为保留位置，不应修改这些位置的寄存器内容。

表 17-1 HRCAP_REGS 寄存器

| 实例 | 结构 | DriverLib 名称 | 基址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| HRCap3Regs | HRCAP_REGS | HRCAP3_BASE | 0x0061_A140 | 是 | 是 | 是 |

### 17.5.2 HRCAP_REGS 寄存器

表中列出了 HRCAP_REGS 寄存器的内存映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，不应修改寄存器内容。

表 17-2 HRCAP_REGS 寄存器

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 0x0000 | HRCTL | 高分辨率控制寄存器 |
| 0x0008 | HRINTEN | 高分辨率校准中断使能寄存器 |
| 0x000C | HRFLG | 高分辨率校准中断标志寄存器 |
| 0x0010 | HRCLR | 高分辨率校准中断清除寄存器 |
| 0x0014 | HRFRC | 高分辨率校准中断强制寄存器 |
| 0x0018 | HRCALPRD | 高分辨率校准周期寄存器 |
| 0x001C | HRSYSCLKCTR | 高分辨率校准 SYSCLK 计数器寄存器 |
| 0x0020 | HRSYSCLKCAP | 高分辨率校准 SYSCLK 捕获寄存器 |
| 0x0024 | HRCLKCTR | 高分辨率校准 HRCLK 计数器寄存器 |
| 0x0028 | HRCLKCAP | 高分辨率校准 HRCLK 捕获寄存器 |

复杂的位访问类型经过编码以适应小型表格单元格。表中显示了本节中用于访问类型的编码。

表 17-3 HRCAP_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R-0 | 读 返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1S | W1S | 写 1 置位 |
| WSonce | WSonce | 一次性写（仅能置位一次） |
| 复位或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i,k,l,m,n | | 当这些变量用于寄存器名称、偏移地址或地址时，指代寄存器数组的值，该寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组以公式表示。 |
| y | | 当该变量用于寄存器名称、偏移地址或地址时，指代寄存器数组的值。 |

### 17.5.2.1 HRCTL 寄存器

偏移地址：0h　复位值：00000000h

高分辨率控制寄存器

表 17-4 HRCTL 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | CALIBCONT | CALIBSTS | CALIBSTART | PRDSEL | HRCLKE | HRE |
| R/W-0h | | R/W-0h | R-0h | R-0/W1S-0h | R/W-0h | R/W-0h | R/W-0h |

表 17-5 HRCTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-6 | RESERVED | R/W | 0h | 保留 |
| 5 | CALIBCONT | R/W | 0h | 连续模式校准选择位。<br>0：禁用连续模式。<br>1：使能连续模式，当前校准周期结束后自动重新开始校准。<br>复位类型：CPU1.SYSRSn |
| 4 | CALIBSTS | R | 0h | 校准状态位。<br>0：没有正在进行的校准周期。<br>1：校准周期正在进行。<br>复位类型：CPU1.SYSRSn |
| 3 | CALIBSTART | R-0/W1S | 0h | 校准启动位。<br>0：无效。<br>1：启动校准周期。<br>复位类型：CPU1.SYSRSn |
| 2 | PRDSEL | R/W | 0h | 校准周期匹配选择位。<br>0：使用 SYSCLK 计数器进行周期匹配（复位默认值）。<br>1：保留。<br>复位类型：CPU1.SYSRSn |
| 1 | HRCLKE | R/W | 0h | 高分辨率时钟使能位。<br>0：禁用高分辨率时钟（复位默认值）。<br>1：使能高分辨率时钟。通过 HRE 位使能高分辨率功能前，应先使能此时钟。<br>复位类型：CPU1.SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 0 | HRE | R/W | 0h | 高分辨率使能位。<br>0：禁用高分辨率模式（复位默认值）。<br>1：使能高分辨率模式，使高分辨率功能可访问 ECAP 的捕获寄存器和边沿事件模式。<br>注意：使能模块前，必须先通过 HRCLKE 位使能高分辨率时钟，并留出一定的启动稳定时间。<br>复位类型：CPU1.SYSRSn |

### 17.5.2.2 HRINTEN 寄存器

偏移地址：8h　复位值：00000000h

高分辨率校准中断使能寄存器

表 17-6 HRINTEN 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | CALPRDCHKSTS | CALIBDONE | 保留 |
| R-0-0h | | | | | R/W-0h | R/W-0h | R-0-0h |

表 17-7 HRINTEN 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-3 | RESERVED | R-0 | 0h | 保留 |
| 2 | CALPRDCHKSTS | R/W | 0h | 校准周期检查状态中断使能。<br>0：禁用校准周期检查状态中断状态。<br>1：使能校准周期检查状态中断状态。<br>复位类型：CPU1.SYSRSn |
| 1 | CALIBDONE | R/W | 0h | 校准完成中断使能。<br>0：禁用校准完成中断。<br>1：使能校准完成中断。<br>复位类型：CPU1.SYSRSn |
| 0 | RESERVED | R-0 | 0h | 保留 |

### 17.5.2.3 HRFLG 寄存器

偏移地址：Ch　复位值：00000000h

高分辨率校准中断标志寄存器

表 17-8 HRFLG 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | CALPRDCHKSTS | CALIBDONE | CALIBINT |
| R-0-0h | | | | | R-0h | R-0h | R-0h |

表 17-9 HRFLG 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-3 | RESERVED | R-0 | 0h | 保留 |
| 2 | CALPRDCHKSTS | R | 0h | 校准周期检查状态标志位。<br>1：某个计数器溢出，使校准在 PRDCHK 之前结束。<br>0：未发生事件。<br>注意：此位保持锁存，直到用户通过 HRCLR[CALPRDCHKSTS] 清除。<br>复位类型：CPU1.SYSRSn |
| 1 | CALIBDONE | R | 0h | 校准完成中断标志位。<br>1：校准周期已完成。<br>0：校准周期尚未完成。<br>注意：此位保持锁存，直到用户通过 HRCLR[CALIBDONE] 清除。<br>复位类型：CPU1.SYSRSn |
| 0 | CALIBINT | R | 0h | 全局校准中断状态标志。<br>1：CALIBDONE 或 CALPRDCHKSTS 已产生中断。<br>0：未产生中断。<br>复位类型：CPU1.SYSRSn |

### 17.5.2.4 HRCLR 寄存器

偏移地址：10h　复位值：00000000h

高分辨率校准中断清除寄存器

表 17-10 HRCLR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | CALPRDCHKSTS | CALIBDONE | CALIBINT |
| R-0-0h | | | | | R-0/W1C-0h | R-0/W1C-0h | R-0/W1C-0h |

表 17-11 HRCLR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-3 | RESERVED | R-0 | 0h | 保留 |
| 2 | CALPRDCHKSTS | R-0/W1C | 0h | 清除校准周期检查状态标志位。<br>1：清除 CALPRDCHKSTS 标志寄存器位。<br>0：无效。<br>注意：若用户清除标志位的同一周期内，硬件事件尝试置位该标志，则硬件优先于 CPU 写入。<br>复位类型：CPU1.SYSRSn |
| 1 | CALIBDONE | R-0/W1C | 0h | 清除校准完成中断标志位。<br>1：清除 CALIBDONE 中断标志寄存器位。<br>0：无效。<br>注意：若用户清除标志位的同一周期内，硬件事件尝试置位该标志，则硬件优先于 CPU 写入。<br>复位类型：CPU1.SYSRSn |
| 0 | CALIBINT | R-0/W1C | 0h | 清除全局校准中断标志。<br>1：清除全局中断标志；若有任一事件标志置位，允许继续产生中断。<br>0：无效。<br>复位类型：CPU1.SYSRSn |

### 17.5.2.5 HRFRC 寄存器

偏移地址：14h　复位值：00000000h

高分辨率校准中断强制寄存器

表 17-12 HRFRC 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | CALPRDCHKSTS | CALIBDONE | 保留 |
| R-0-0h | | | | | R-0/W1S-0h | R-0/W1S-0h | R-0-0h |

表 17-13 HRFRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-3 | RESERVED | R-0 | 0h | 保留 |
| 2 | CALPRDCHKSTS | R-0/W1S | 0h | 强制 CALPRDCHKSTS 标志。<br>0：无效。<br>1：置位 CALPRDCHKSTS 标志。<br>复位类型：CPU1.SYSRSn |
| 1 | CALIBDONE | R-0/W1S | 0h | 强制 CALIBDONE 标志。<br>0：无效。<br>1：置位 CALIBDONE 标志。<br>复位类型：CPU1.SYSRSn |
| 0 | RESERVED | R-0 | 0h | 保留 |

### 17.5.2.6 HRCALPRD 寄存器

偏移地址：18h　复位值：003FFFFFh

高分辨率校准周期寄存器

表 17-14 HRCALPRD 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PRD | | | | | | | |
| R/W-003FFFFFh | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PRD | | | | | | | |
| R/W-003FFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PRD | | | | | | | |
| R/W-003FFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRD | | | | | | | |
| R/W-003FFFFFh | | | | | | | |

表 17-15 HRCALPRD 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | PRD | R/W | 003FFFFFh | 用于设置校准周期的寄存器。周期值与 HRSYSCLKCTR 比较；匹配时产生中断，并捕获各计数器寄存器的值。<br>复位类型：CPU1.SYSRSn |

## 17.5.2.7 HRSYSCLKCTR 寄存器

偏移地址：1Ch　复位值：00000000h

高分辨率校准 SYSCLK 计数器寄存器

表 17-16 HRSYSCLKCTR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| HRSYSCLKCTR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| HRSYSCLKCTR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HRSYSCLKCTR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HRSYSCLKCTR | | | | | | | |
| R-0h | | | | | | | |

表 17-17 HRSYSCLKCTR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | HRSYSCLKCTR | R | 0h | 当前 SYSCLK 计数器值。<br>复位类型：CPU1.SYSRSn |

### 17.5.2.8 HRSYSCLKCAP 寄存器

偏移地址：20h　复位值：00000000h

高分辨率校准 SYSCLK 捕获寄存器

表 17-18 HRSYSCLKCAP 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| HRSYSCLKCAP | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| HRSYSCLKCAP | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HRSYSCLKCAP | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HRSYSCLKCAP | | | | | | | |
| R-0h | | | | | | | |

表 17-19 HRSYSCLKCAP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | HRSYSCLKCAP | R | 0h | 校准周期结束时，将 HRSYSCLKCTR 捕获到此寄存器。<br>复位类型：CPU1.SYSRSn |

### 17.5.2.9 HRCLKCTR 寄存器

偏移地址：24h　复位值：00000000h

高分辨率校准 HRCLK 计数器寄存器

表 17-20 HRCLKCTR 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| HRCLKCTR | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| HRCLKCTR | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HRCLKCTR | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HRCLKCTR | | | | | | | |
| R-0h | | | | | | | |

表 17-21 HRCLKCTR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | HRCLKCTR | R | 0h | 当前 HRCLK 计数器值。<br>注意：HRCLK 未同步到 SYSCLK 时钟域，因此读取值可能不准确。<br>复位类型：CPU1.SYSRSn |

## 17.5.2.10 HRCLKCAP 寄存器

偏移地址：28h　复位值：00000000h

高分辨率校准 HRCLK 捕获寄存器

表 17-22 HRCLKCAP 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| HRCLKCAP | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| HRCLKCAP | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| HRCLKCAP | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HRCLKCAP | | | | | | | |
| R-0h | | | | | | | |

表 17-23 HRCLKCAP 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | HRCLKCAP | R | 0h | 校准周期结束时，将 HRCLKCTR 捕获到此寄存器。<br>注意：HRCLK 未同步到 SYSCLK 时钟域，因此读取值可能不准确。<br>复位类型：CPU1.SYSRSn |

### 17.5.3 HRCAP 寄存器与 DriverLib 函数对应表

表 17-24 HRCAP 寄存器与 DriverLib 函数对应表

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| TSCTR | - | - |
| CTRPHS | - | - |
| CAP1 | - | - |
| CAP2 | - | - |
| CAP3 | - | - |
| CAP4 | - | - |
| ECCTL0 | - | - |
| ECCTL1 | - | - |
| ECCTL2 | - | - |
| ECEINT | - | - |
| ECFLG | - | - |
| ECCLR | - | - |
| ECFRC | - | - |
| ECAPSYNCINSEL | - | - |
| HRCTL | hrcap.h | HRCAP_enableHighResolution |
| HRCTL | hrcap.h | HRCAP_disableHighResolution |
| HRCTL | hrcap.h | HRCAP_enableHighResolutionClock |
| HRCTL | hrcap.h | HRCAP_disbleHighResolutionClock |
| HRCTL | hrcap.h | HRCAP_startCalibration |
| HRCTL | hrcap.h | HRCAP_setCalibrationMode |
| HRCTL | hrcap.h | HRCAP_isCalibrationBusy |
| HRINTEN | hrcap.h | HRCAP_enableCalibrationInterrupt |
| HRINTEN | hrcap.h | HRCAP_disableCalibrationInterrupt |
| HRFLG | hrcap.h | HRCAP_getCalibrationFlags |
| HRCLR | hrcap.h | HRCAP_clearCalibrationFlags |
| HRFRC | hrcap.h | HRCAP_forceCalibrationFlags |
| HRCALPRD | hrcap.h | HRCAP_setCalibrationPeriod |

| 寄存器 | 文件 | DriverLib 函数 |
| --- | --- | --- |
| HRCALPRD | hrcap.h | HRCAP_configCalibrationPeriod |
| HRSYSCLKCTR | - | - |
| HRSYSCLKCAP | hrcap.h | HRCAP_getCalibrationClockPeriod |
| HRCLKCTR | - | - |
| HRCLKCAP | hrcap.h | HRCAP_getCalibrationClockPeriod |

# 18 增强型正交编码器脉冲（eQEP）

增强型正交编码器脉冲（eQEP）模块用于直接与线性或旋转式增量编码器接口，从旋转机械中获取位置、方向和转速信息，适用于高性能运动与位置控制系统。

## 18.1 简介

增量式编码盘沿其外围刻有一圈槽孔轨道，如图所示。这些槽孔形成明暗相间的线条图案。码盘分辨率定义为每转一圈所出现的明暗线对数量（每转线条数）。通常，会额外增加第二条轨道，用于产生每转一次的信号（索引信号：QEPI），该信号可用于指示绝对位置。

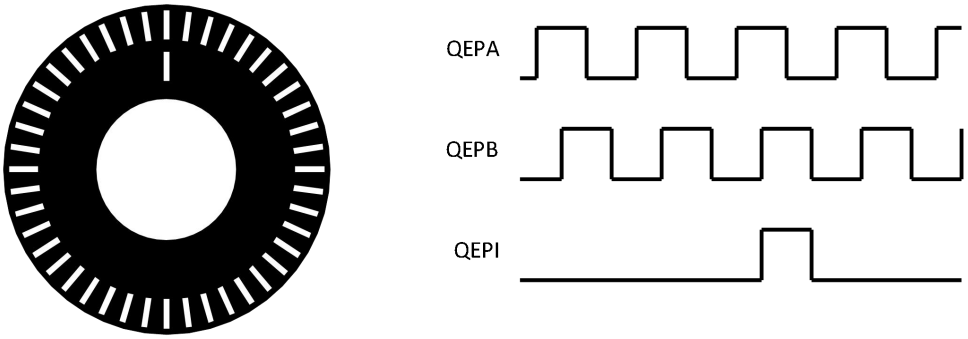



图 18-1 增量式编码器码盘结构及输出信号波形

为了获取方向信息，码盘上的栅线通过两个不同的光电元件进行读取。这两个光电元件在观测码盘图案时，彼此之间存在线对间距 1/4 的机械偏移。该偏移通过遮光板或掩膜板实现，它将光电元件的感光范围限制在码盘栅线的指定区域。当码盘旋转时，两个光电元件会产生相互相位差 90°的信号，这两个信号通常被称为正交信号 QEPA 和 QEPB。对于大多数编码器而言，顺时针方向定义为：QEPA 通道相位超前于 QEPB 通道；反之则为逆时针方向。

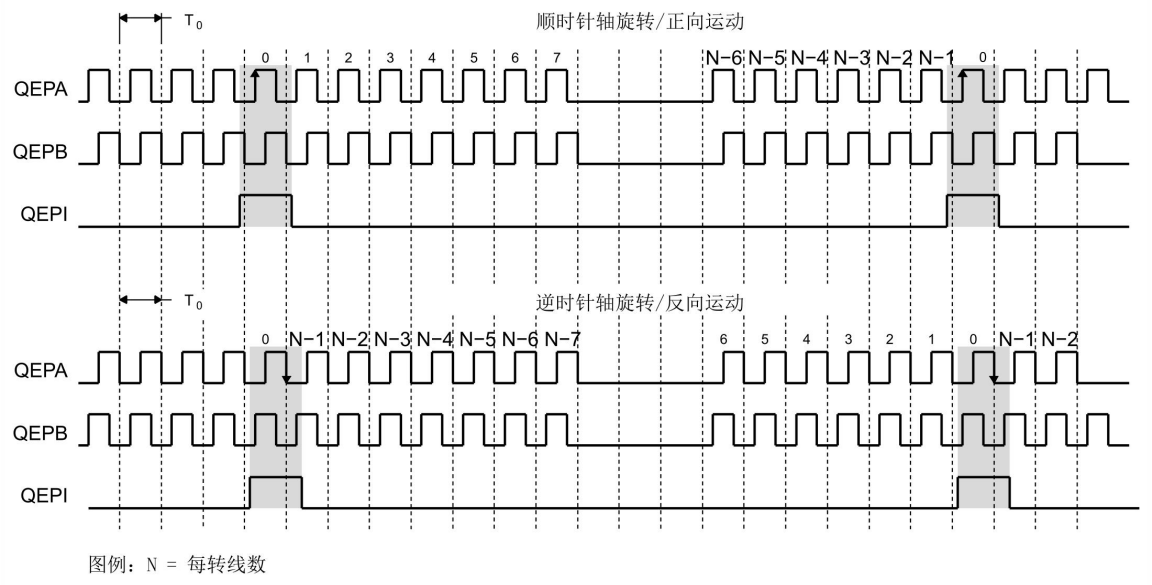



图 18-2 QEP 编码器用于正向/反向运动的输出信号

编码轮通常与电机同轴同步旋转一圈，也可以通过齿轮传动与电机形成特定转速比。因此，从 QEPA 和 QEPB 输出的数字信号频率会随电机转速成比例变化。例如，一个 2000 线的编码器直接连接到转速为 5000 转 / 分钟（rpm）的电机上，输出信号频率为 166.6 kHz。因此，处理器通过测量 QEPA 或 QEPB 任意一路的输出频率，即可确定电机转速。

不同制造商生产的正交编码器均提供两种索引脉冲形式（门控索引脉冲或非门控索引脉冲），如图所示。非标准形式的索引脉冲为非门控型；在非门控配置中，索引脉冲沿沿与 A 及 B 信号未必重合。门控索引脉冲则可与四个正交信号沿中的任一沿对齐，其脉冲宽度可为正交信号周期的四分之一、二分之一或完整周期。

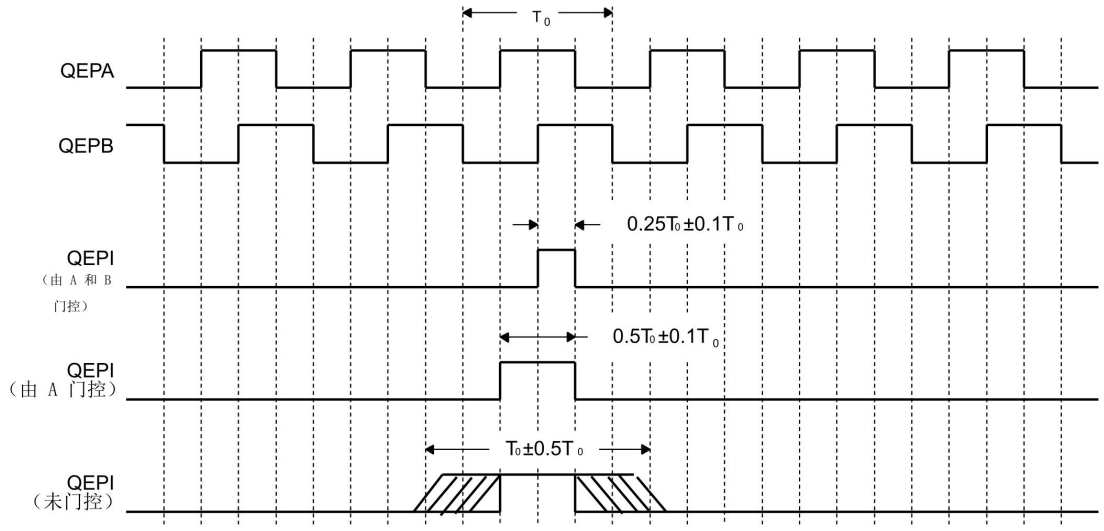



图 18-3 索引脉冲示例

轴式编码器的一些典型应用包括机器人技术以及以鼠标形式实现的计算机输入。在鼠标内部，您可以观察到鼠标滚轮绕着一对轴（左/右轴与上/下轴）旋转。这些轴分别与光学轴式编码器相连，从而向计算机精确传递鼠标移动的速度及方向信息。

$$v(k) \approx \frac{x(k)-x(k-1)}{T} = \frac{\Delta X}{T} \qquad (18\text{-}1)$$

$$v(k) \approx \frac{X}{t(k)-t(k-1)} = \frac{X}{\Delta T} \qquad (18\text{-}2)$$

式（18-1）是转速估算的常规方法，该方法需要一个时基来为转速计算提供单位时间事件。单位时间本质上是转速计算频率的倒数。

在每个单位时间事件内读取一次编码器计数值（位置值）。通过将当前读数减去上一次读数，得到差值 x(k)−x(k−1)。随后，用该差值乘以已知常数 1/T（其中 T 为单位时间事件之间的固定时间间隔，且为已知量），即可得到估算转速。

基于式（18-1）的估算方法存在固有的精度限制，该限制直接与位置传感器分辨率和单位时间周期 T 相关。例如，考虑一个每转 500 线的正交编码器，转速计算频率为 400Hz。在用于位置检测时，正交编码器可实现四倍分辨率提升，在本例中即为每转 2000 个计数。因此，该编码器可检测的最小转动量为 0.0005 转，在 400Hz 采样频率下对应的转速分辨率为 12 转 / 分钟。尽管该分辨率在中高速工况下可以满足要求（例如在 1200 转 / 分钟时误差仅为 1%），但在低速工况下明显

不足。事实上，当转速低于 12 转 / 分钟时，转速估算结果在大部分时间内会错误地显示为零。

在低速工况下，式（18-2）能提供更精确的估算方法。该方法需要能够输出固定间隔脉冲序列的位置传感器，例如前文所述的正交编码器。对于给定的传感器分辨率，每个脉冲的宽度由电机转速决定。通过测量连续正交脉冲边沿之间的流逝时间，可利用式（18-2）计算电机转速。但该方法与式（18-1）一样，存在相反的局限性：当电机转速较高且传感器分辨率较高时，时间间隔 ΔT 会很小，从而更容易受到定时器分辨率的影响，这会给高速转速估算带来较大误差。

对于宽转速范围的系统（即需要同时进行低速和高速转速估算），一种可行方案是：在低速时采用式（18-2），当电机转速上升至某一设定阈值以上时，由 DSP 软件切换为式（18-1）进行计算。

### 18.1.1 EQEP 输入

eQEP 输入包括两个用于正交时钟模式或方向计数模式的引脚、一个索引信号（或零位标记），以及一个选通输入。eQEP 模块要求 QEPA、QEPB 以及 QEPI 输入在进入模块前先与 SYSCLK 同步。应用程序代码可以在任意使能了 eQEP 功能的 GPIO 引脚上开启 GPIO 输入同步功能。

·QEPA/XCLK 和 QEPB/XDIR

这两个引脚可用于正交时钟模式或方向计数模式。

- 正交时钟模式：eQEP 编码器会输出两路相位相差 90 电角度的方波信号（A 相和 B 相）。利用该相位关系可以判断输入轴的旋转方向，并统计从索引位置开始的 eQEP 脉冲数，从而得到相对位置信息。当轴正向（顺时针）旋转时，QEPA 信号超前于 QEPB 信号，反之则相反。正交解码单元利用这两路输入信号，生成正交时钟信号和方向信号。

- 方向计数模式：在方向计数模式下，方向信号与时钟信号直接由外部器件提供。部分位置编码器采用此类输出方式，而非正交信号输出。其中，QEPA 引脚提供时钟输入，QEPB 引脚提供方向输入。

·QEPI：

索引信号或零位标记 eQEP 编码器使用索引信号来确定一个绝对起始位置，位置信息以此为基准通过正交脉冲进行增量式编码。该引脚连接至 eQEP 编码器的索引输出端，可选择为每一圈旋转复位一次位置计数器。当索引引脚上出现预期事件时，该信号可用于初始化或锁存位置计数器。

·QEPS：

选通输入该通用选通信号可在选通引脚上出现预期事件时，对位置计数器进行初始化或锁存。该信号通常连接至传感器或限位开关，用于通知系统电机已到达设定位置。

eQEP 的输入信号（QEPA、QEPB、QEPI 和 QEPS）可以来自多种信号源，包括器件引脚、CMPSSx 或 PWMXBARx。一个典型应用场景是：若电机控制系统中使用 正余弦传感器（SinCos transducer） 估算电机轴位置，且索引信号来自传统旋转编码器，则可将 eQEP 信号（QEPA、QEPB、QEPI）的来源配置为 CMPSSx 的输出，由 CMPSSx 对 Sin、Cos 及 Index 信号进行解码。图中展示了这一场景。

用户可通过 QEPSRCSEL 寄存器 对输入信号（QEPA、QEPB、QEPI）的来源进行配置，如表所示。

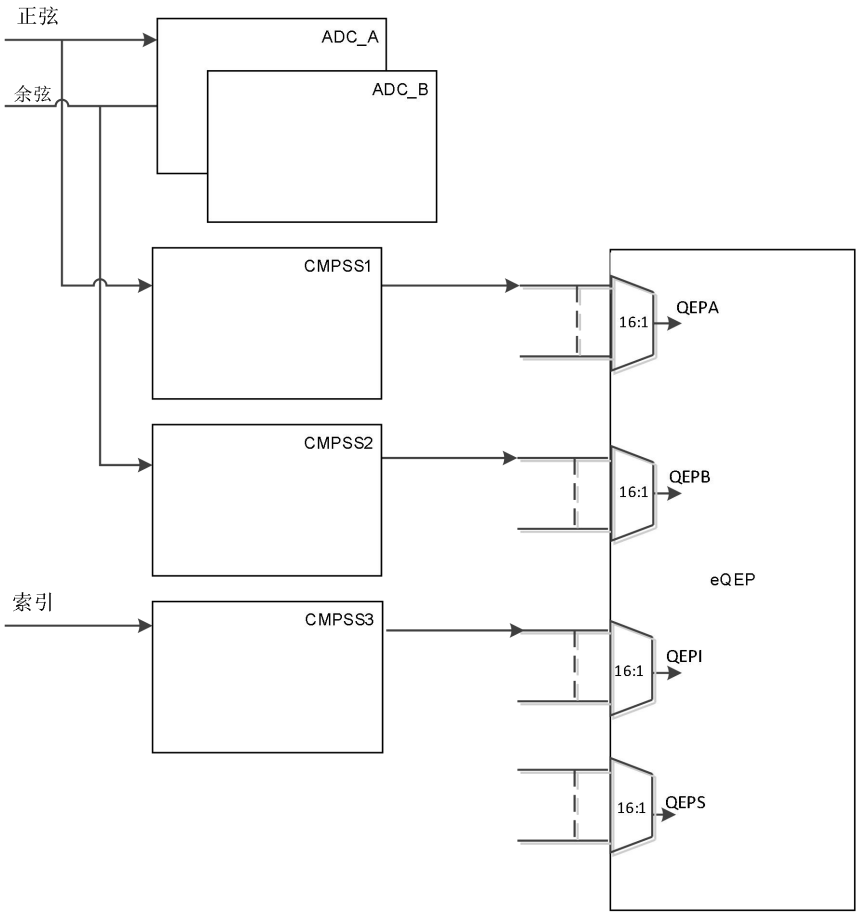



图 18-4 利用 eQEP 解码正弦余弦换能器信号

表 18-1 eQEP 信号输入源选择

| QEPASEL, QEPBSEL, QEPISEL | 输入信号 |
|---|---|
| 0 | DEVICE_PIN |
| 1 | CMPSS1_CTRIPH |
| 2 | CMPSS2_CTRIPH |
| 3 | CMPSS3_CTRIPH |
| 4 | CMPSS4_CTRIPH |
| 5-7 | Reserved |
| 8 | ZERO |
| 9 | EPWMXBAR1 |
| 10 | EPWMXBAR2 |
| 11 | EPWMXBAR3 |
| 12 | EPWMXBAR4 |
| 13 | EPWMXBAR5 |
| 14 | EPWMXBAR6 |
| 15 | EPWMXBAR7 |

配置 QEPSRCSEL 寄存器以选择 QEPA、QEPB 和 QEPI 信号的来源时，可能会导致这些信号出现意外的跳变。如果 eQEP 模块此时已经处于运行状态，这种跳变可能会引发非预期的后果。

## 18.2 主要特性

- 针对每个引脚的可编程输入滤波功能（属于 GPIO 多路复用器的一部分）
- 正交解码单元（QDU）
- 用于位置检测的位置计数器及控制单元（PCCU）
- 用于低速检测的正交边沿捕获单元（QCAP）
- 用于速度 / 频率测量的单位时基单元（UTIME）
- 用于检测堵转 / 停转的看门狗定时器（QWDOG）
- 正交模式适配器（QMA）

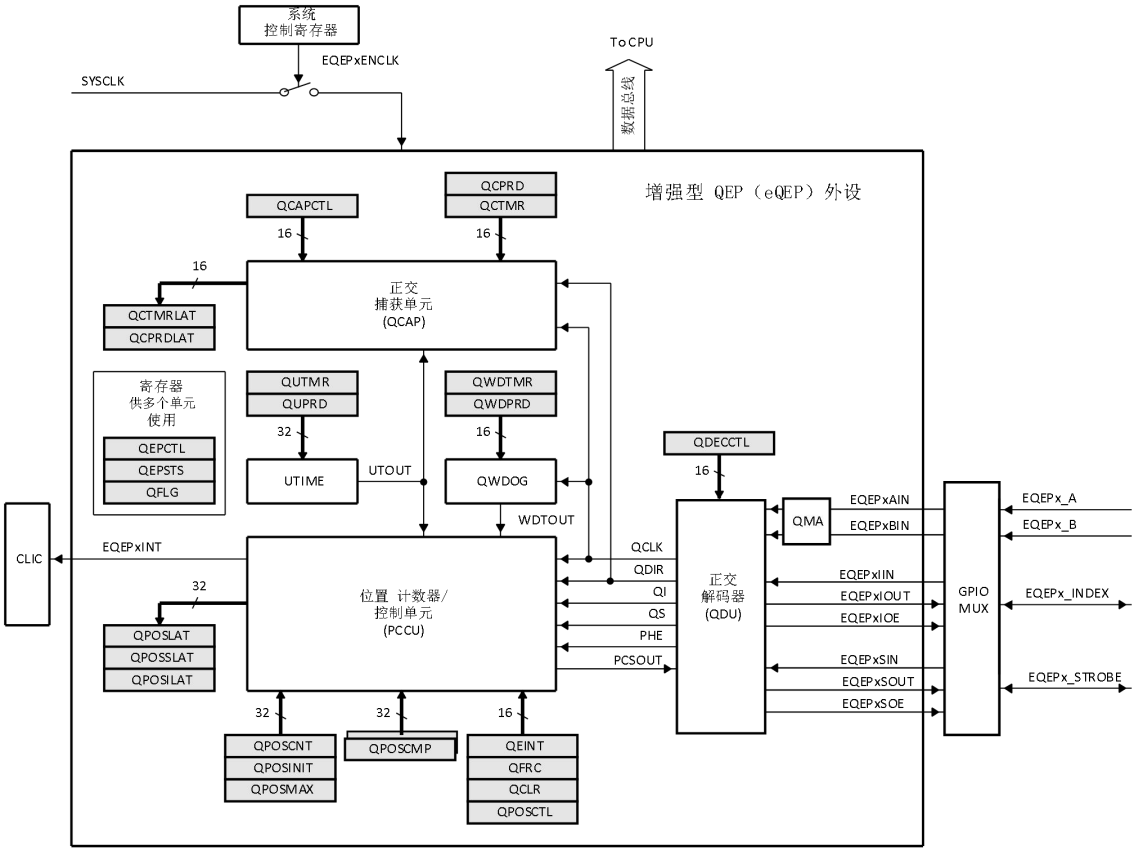



图 18-5 eQEP 外部功能框图

## 18.3 功能描述

### 18.3.1 正交解码器单元（QDU）

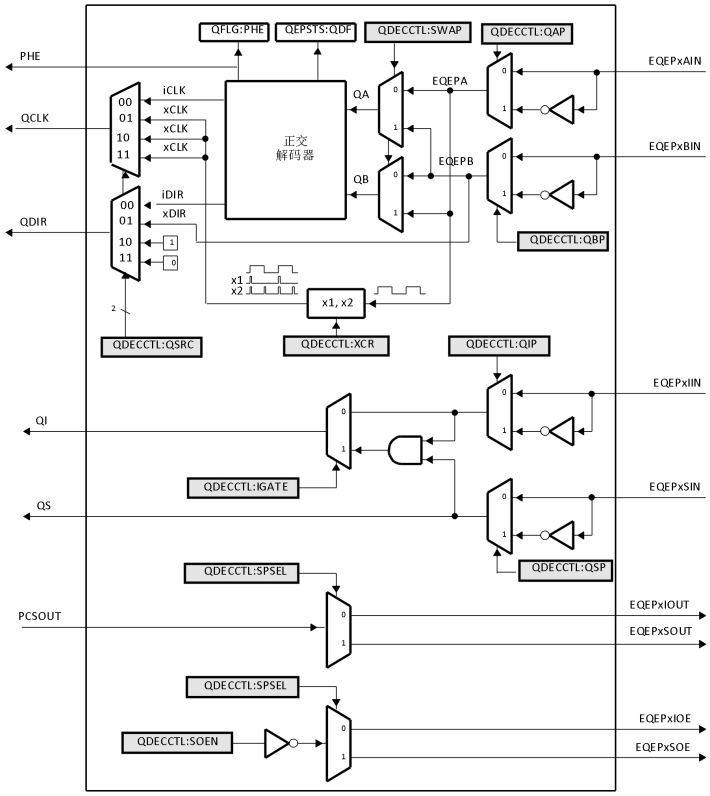



图 18-6 eQEP QDU 单元功能框图

#### 18.3.1.1 位置计数输入模式

根据接口输入需求，通过 QDECCTL[QSRC] 位选择输入到位置计数器的时钟与方向信号，可选模式如下：

- 正交计数模式
- 方向计数模式
- 向上计数模式
- 向下计数模式
- 位置计数模式

在正交计数模式下，正交解码器会为位置计数器生成方向信号与时钟信号。

(1)方向解码

eQEP 电路的方向解码逻辑会判断（QEPA、QEPB）两路信号中哪一路处于超前状态，并据此更新 QEPSTS[QDF] 位中的方向信息。表 分别以真值表和状态机形式给出了方向解码逻辑。

电路会对 QEPA 与 QEPB 信号的双边沿均进行检测，以此为位置计数器产生计数脉冲。因此，由 eQEP 逻辑生成的时钟频率为每路输入信号频率的四倍。图中展示了根据 eQEP 输入信号进行方向解码与时钟生成的过程。

表 18-2 eQEP 正交解码器真值表

| 上一边沿 | 当前边沿 | 方向 | 位置计数器 |
|---|---|---|---|
| QA↑ | QB↑<br>QB↓<br>QA↓ | 上<br>下<br>翻转 | 增<br>减<br>增或减 |
| QA↓ | QB↓<br>QB↑<br>QA↑ | 上<br>下<br>翻转 | 增<br>减<br>增或减 |
| QB↑ | QA↑<br>QA↓<br>QB↓ | 下<br>上<br>翻转 | 减<br>增<br>增或减 |
| QB↓ | QA↓<br>QA↑<br>QB↑ | 下<br>上<br>翻转 | 减<br>增<br>增或减 |

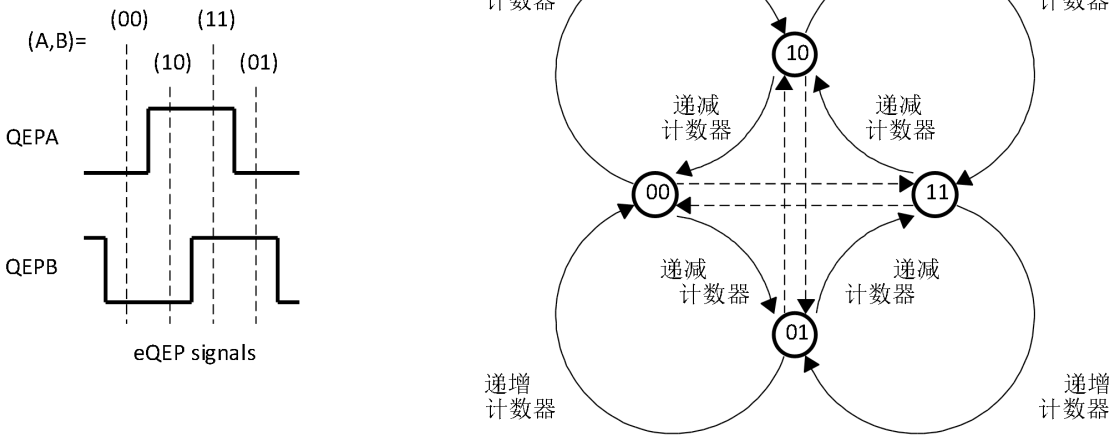



图 18-7 正交解码器状态机

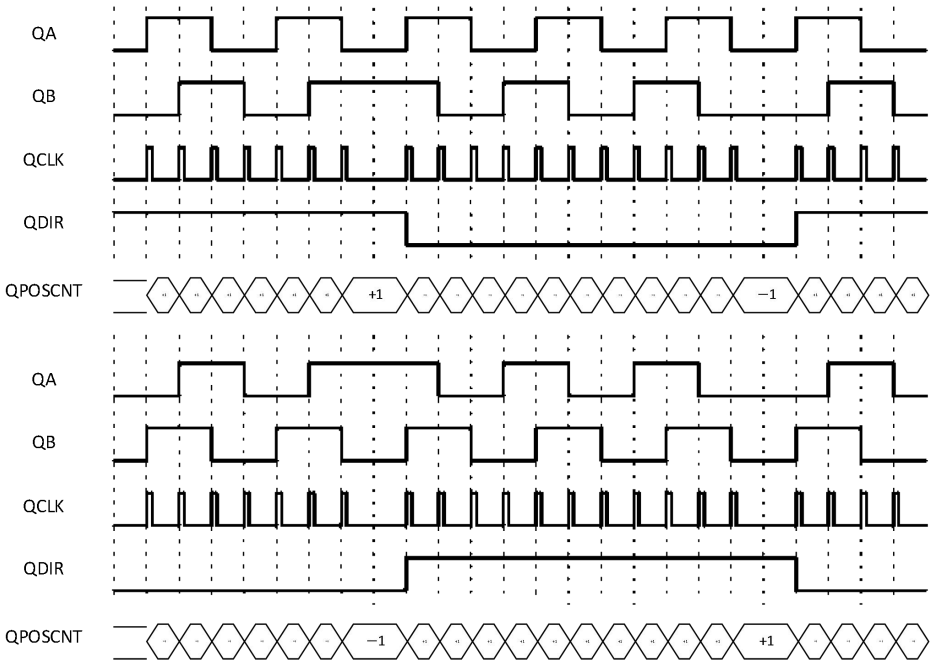


图 18-8 正交时钟和方向解码

(2)相位错误标志

在正常工作状态下，正交输入信号 QEPA 与 QEPB 相位相差 90°。当检测到 QEPA 与 QEPB 信号同时出现边沿跳变时， 相位错误标志位（PHE） 会在 QFLG 寄存器中被置位，此时位置计数器 QPOSCNT 的数值可能出现错误，并产生 1 或 3 个计数的偏移。该事件可选择性地产生中断。图中由虚线标记的状态跳转均为无效跳转，会触发相位错误。

(3)计数倍频

eQEP 位置计数器会在 QEPA、QEPB 两路输入信号的上升沿与下降沿均产生正交时钟（QCLK），从而实现 4 倍于输入时钟的分辨率。

(4)方向计数

在正常正交计数模式下，QEPA 信号接入正交解码器的 QA 输入端，QEPB 信号接入 QB 输入端。通过将 QDECCTL 寄存器中的 SWAP 位置 1 可启用反向计数功能，该操作会交换正交解码器的两路输入，进而反转计数方向。

1）方向计数模式

部分位置编码器会提供方向信号与时钟信号输出，而非正交信号输出。针对这类情况，可以使用方向计数模式。其中，QEPA 输入端为位置计数器提供时钟信号，QEPB 输入端提供方向信息。当方向输入为高电平时，位置计数器在 QEPA 输入的每个上升沿递增计数；方向输入为低电平时，则递减计数。

2）向上计数模式

计数器方向信号被硬件固定为向上计数，位置计数器用于测量 QEPA 输入信号的频率。将 QDECCTL [XCR] 位清零，可以使能在 QEPA 输入的双边沿向位置计数器产生时钟，从而将测量分辨率提高 2 倍。在向上计数模式下，建议应用程序不要将 QEPB 配置为 GPIO 多路复用选项，或者确保 QEPB 输入端不会产生信号边沿。

3）向下计数模式

计数器方向信号被硬件固定为向下计数，位置计数器用于测量 QEPA 输入信号的频率。将 QDECCTL[XCR] 位清零，可以使能在 QEPA 输入的双边沿向位置计数器产生时钟，从而将测量分辨率提高 2 倍。在向下计数模式下，应用程序不得将 QEPB 配置为 GPIO 多路复用功能，或确保 QEPB 输入端不会产生信号边沿。

#### 18.3.1.2 EQEP 输入极性选择

每个 eQEP 输入都可以通过 QDECCTL[8:5] 控制位进行取反。例如，将 QDECCTL[QIP] 位置 1 即可对索引输入进行取反。

#### 18.3.1.3 位置比较同步输出

增强型 eQEP 外设包含一个位置比较单元，用于在位置计数器寄存器（QPOSCNT）与位置比较寄存器（QPOSCMP）的值匹配时，产生位置比较同步信号。该同步信号可通过 eQEP 外设的索引引脚或选通引脚输出。

将 QDECCTL [SOEN] 位置 1 可使能位置比较同步输出功能，QDECCTL [SPSEL] 位则用于选择使用 eQEP 索引引脚还是 eQEP 选通引脚。

### 18.3.2 位置计数和控制单元（PCCU）

位置计数器及控制单元提供两个配置寄存器（QEPCTL 和 QPOSCTL），用于设置位置计数器的工作模式、位置计数器初始化 / 锁存模式，以及用于生成同步信号的位置比较逻辑。

#### 18.3.2.1 位置计数器操作模式

位置计数器的数据可以通过多种方式捕获。在某些系统中，位置计数器会连续累计多圈的计数值，其计数值用于提供相对于已知参考点的位置信息。一个典型例子是安装在控制打印机打印头的电机上的正交编码器。在这种应用中，先将打印头移动到原点位置以复位位置计数器，随后位置计数器便提供相对于原点的绝对位置信息。

在另一些系统中，位置计数器会利用索引脉冲在每一圈时复位，从而提供相对于索引脉冲位置的转子角度。

位置计数器可配置为以下四种工作模式：

索引事件触发位置计数器复位

最大位置触发位置计数器复位

首次索引事件触发位置计数器复位

单位时基超时事件触发位置计数器复位（频率测量）

在上述所有工作模式下，位置计数器溢出时复位为 0，下溢时加载为 QPOSMAX 寄存器的值。当位置计数器计数值超过 QPOSMAX 后继续向上计数时，发生溢出；当计数值为 0 后继续向下计数时，发生下溢。相关中断标志位会在 QFLG 寄存器中置位，用于指示溢出 / 下溢状态。

索引事件时的位置计数器复位(QEPCTL[PCRM]=00)

如果索引事件在正向运动期间发生，则位置计数器将在下一个 eQEP 时钟复位为 0。如果索引事件在反向运动期间发生，则位置计数器将在下一个 eQEP 时钟复位为 QPOSMAX 寄存器中的值。

首个索引标记定义为第一个索引边沿之后的正交边沿。eQEP 外设会在 QEPSTS 寄存器中记录首个索引标记的发生状态（QEPSTS [FIMF]）以及首次索引事件标记时的方向（QEPSTS [FIDF]）。eQEP 外设还会记住首个索引标记对应的正交边沿，以便在后续索引事件复位操作中使用相同的相对正交跳变沿。

例如，如果首次复位操作发生在正向旋转时 QEPB 的下降沿，则后续所有复位都必须与此边沿对齐：正向旋转时使用 QEPB 下降沿，反向旋转时使用 QEPB 上升沿。

在每个索引事件标记时刻，位置计数器的值会被锁存到 QPOSILAT 寄存器，方向信息则记录在 QEPSTS [QDLF] 位中。如果锁存的值不等于 0 或 QPOSMAX，则位置计数器错误标志位（QEPSTS [PCEF]）与错误中断标志位（QFLG [PCE]）将被置位。位置计数器错误标志位（QEPSTS [PCEF]）在每个索引事件标记时更新，错误对应的中断标志位（QFLG [PCE]）一旦置位，只能通过软件清零。

当 PCRM=0 时，索引事件锁存配置位 QEPCTL [IEL] 必须配置为 00 或 11；并且位置计数器错误标志 / 中断标志仅在索引事件复位模式下产生。在每个索引标记时刻，位置计数器的值都会被锁存到 IPOSLAT 寄存器中。

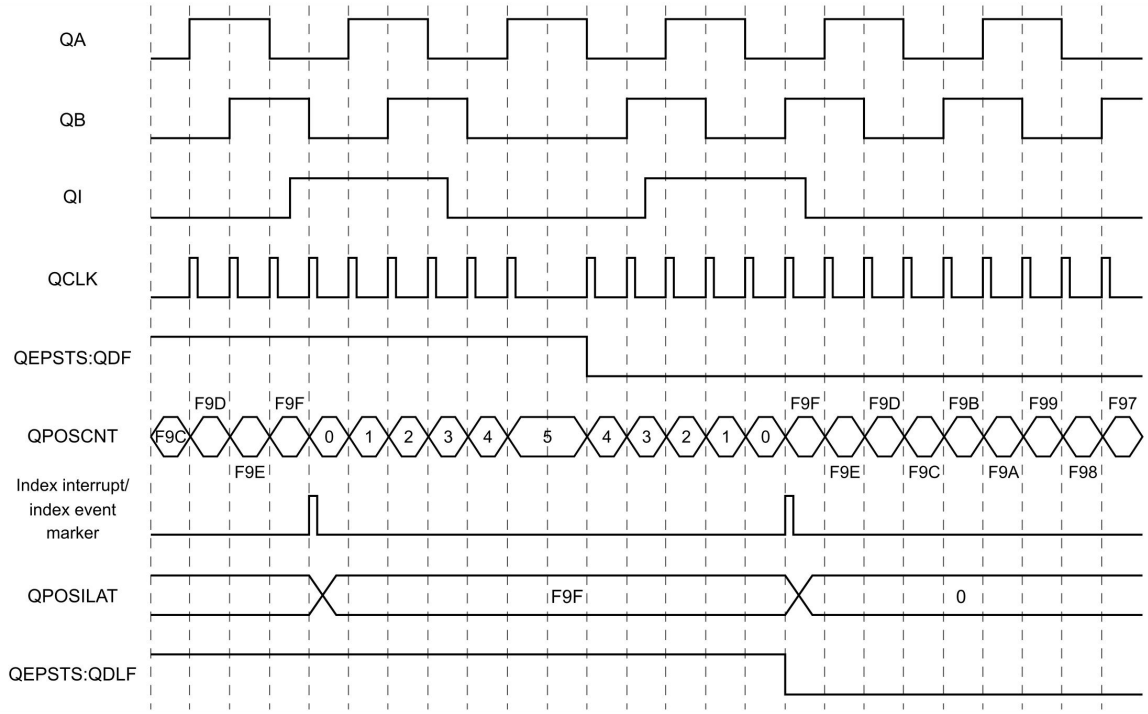


图 18-9 通过索引脉冲实现 1000 线编码器的位置计数器复位（qposmax = 3999 或 0xF9F）

若边界条件中索引事件与前一个 QCLK 边之间的时间间隔小于 SYSCLK 周期，则 qposcnt 会在同一 SYSCLK 周期内重置为零或 qposmax，且不等待下一个 QCLK 边发生。

计数到最大位置时的位置计数器复位(QEPCTL[PCRM]=01)

如果位置计数器的值等于 QPOSMAX，则在正向运动时，位置计数器将在下一个 eQEP 时钟复位为 0，并置位位置计数器溢出标志。如果位置计数器的值等于 0，则在反向运动时，位置计数器将在下一个 eQEP 时钟复位为 QPOSMAX，并置位位置计数器下溢标志。图中展示了该模式下位置计数器的复位操作。

首个索引标记相关字段（QEPSTS [FIDF] 和 QEPSTS [FIMF]）在此模式下无效。

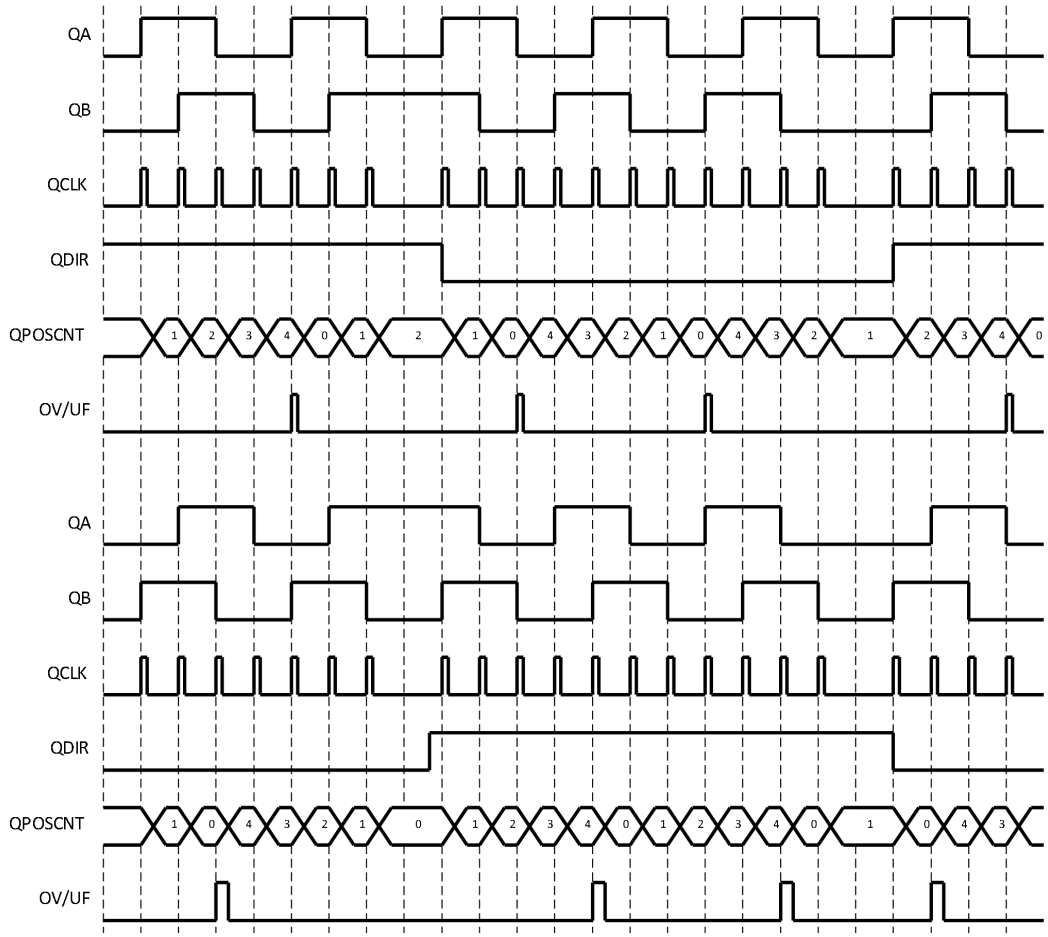



图 18-10 位置计数上溢/下溢（(QPOSMAX = 4）

首次索引事件计数器复位(QEPCTL[PCRM]=11)

在此模式下，当发生单位时基事件时，QPOSCNT 会被置为 0 或 QPOSMAX，具体取决于由 QDECCTL [QSRC] 位所选的方向模式。该模式适用于频率测量。

#### 18.3.2.2 位置计数锁存

eQEP 的索引输入和选通输入可进行配置，当对应引脚上发生指定事件时，能分别将位置计数器（QPOSCNT）锁存到 QPOSILAT 和 QPOSSLAT 寄存器中。

（1）索引事件锁存

在某些应用场合，不希望在每次索引事件时都复位位置计数器，而是需要让位置计数器工作在完整 32 位模式下（QEPCTL [PCRM] = 01 和 QEPCTL [PCRM] = 10 模式）。

在这种情况下，eQEP 位置计数器可配置为在以下事件发生时执行锁存操作，并且在每个索引事件标记时刻，方向信息会被记录到 QEPSTS [QDLF] 位中。

上升沿锁存（QEPCTL [IEL]=01）

下降沿锁存（QEPCTL [IEL]=10）

索引事件标记锁存（QEPCTL [IEL]=11）

该功能作为一种错误检测机制尤为实用，可用于检查位置计数器在两次索引事件之间是否累计了正确的计数值。例如，一个 1000 线的编码器，在同一方向上移动时，两次索引事件之间应累计计数 4000 次。

当位置计数器的值被锁存到 QPOSILAT 寄存器时，索引事件锁存中断标志位（QFLG [IEL]）会被置位。当 QEPCTL [PCRM] = 00 时，索引事件锁存配置位（QEPCTL [IEL]）将被忽略。

升沿锁存：（QEPCTL [IEL]=01）在索引输入的每个上升沿，将位置计数器的值（QPOSCNT）锁存到 QPOSILAT 寄存器。

下降沿锁存：（QEPCTL [IEL] = 10）在索引输入的每个下降沿，将位置计数器的值（QPOSCNT）锁存到 QPOSILAT 寄存器。

索引事件标记 / 软件索引标记锁存：（QEPCTL [IEL] = 11）首个索引标记定义为第一个索引边沿之后的正交边沿。eQEP 外设会在 QEPSTS 寄存器中记录首个索引标记的发生状态（QEPSTS [FIMF]）以及首次索引事件标记时的方向（QEPSTS [FIDF]）。eQEP 外设还会记住首个索引标记对应的正交边沿，以便在锁存位置计数器时使用相同的相对正交跳变沿（QEPCTL [IEL]=11）。

图中展示了采用索引事件标记器的位置计数器锁存器

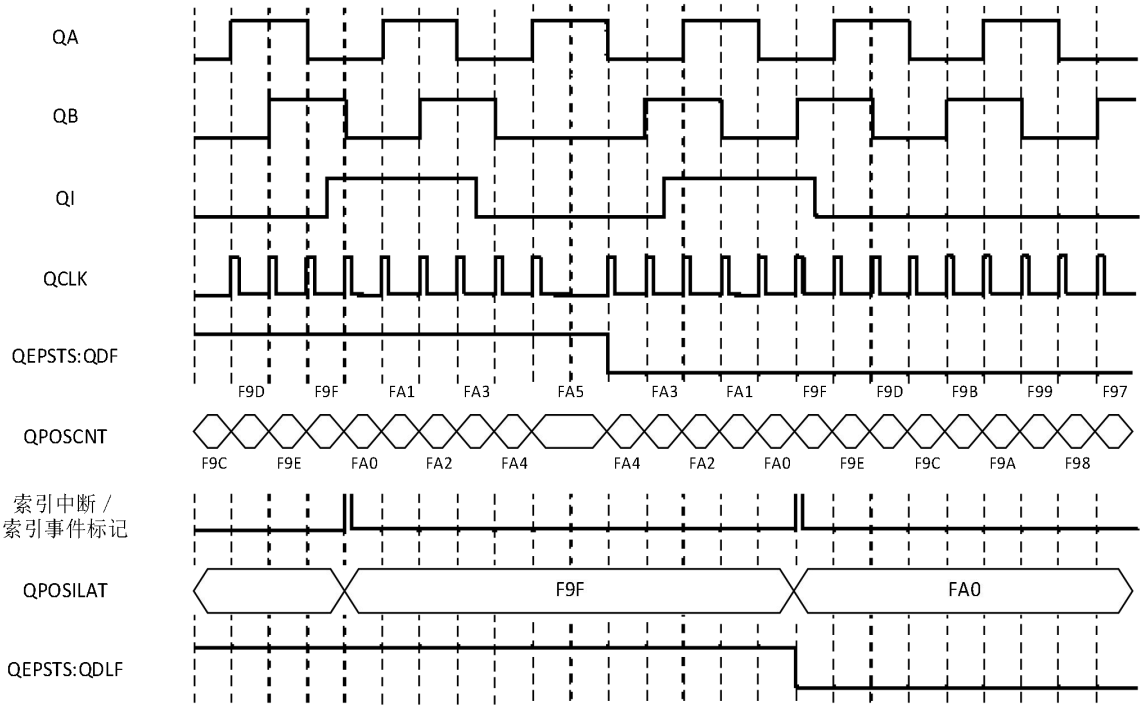



图 18-11 适用于 1000 线编码器的软件索引标记（QEPCTL [IEL] = 1）

(2)选通事件锁存

通过清零 QEPCTL[SEL] 位，可在选通输入的上升沿将位置计数器的值锁存到 QPOSSLAT 寄存器。

如果将 QEPCTL[SEL] 位置 1，则：

正向运行时，在选通输入的上升沿锁存位置计数器值到 QPOSSLAT；反向运行时，在选通输入的下降沿锁存。

当位置计数器被锁存到 QPOSSLAT 寄存器时，选通事件锁存中断标志位 QFLG[SEL] 会被置位。

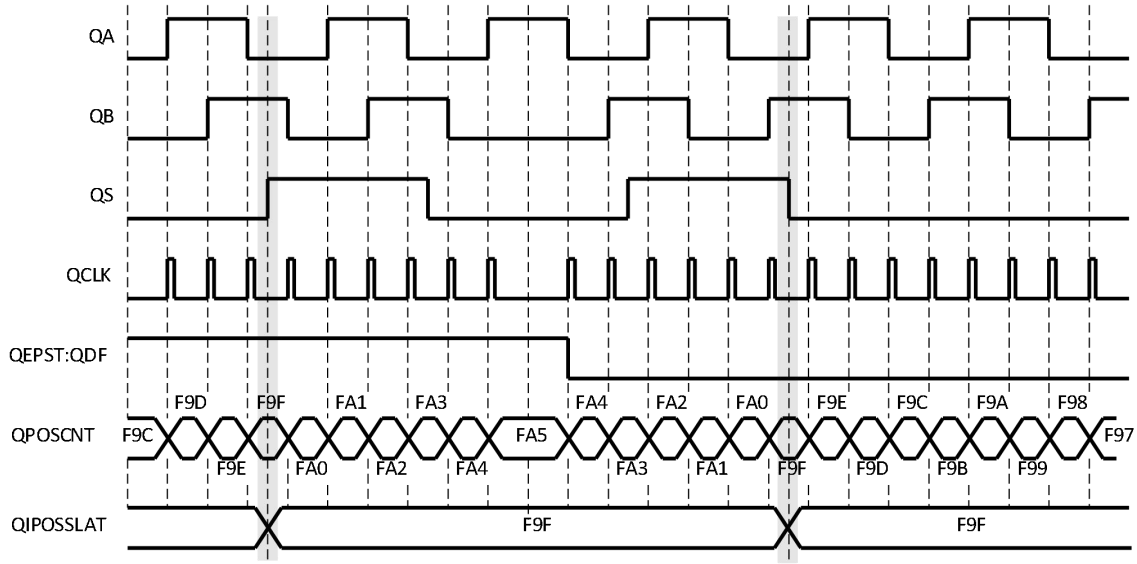



图 18-12 选通事件锁存器（QEPCTL [SEL]=1）

通过配置寄存器 QEPSTROBESEL.STROBESEL，可在 ADCSOCA 和 ADCSOCB 事件发生时锁存位置计数器的值，如图所示。

若要将 ADCSOCA/B 事件用作 QS 信号，需要将 QEPSRCSEL.QEPSSEL 配置为非零值。

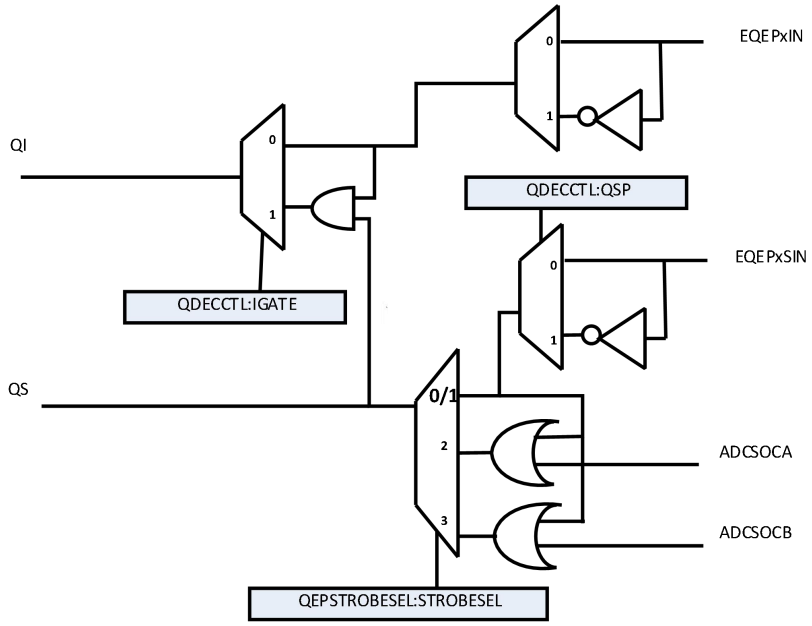



图 18-13 adcsoca/adcsocb 事件中的锁存位置计数器

### 18.3.2.3 位置计数初始化

可通过以下事件对位置计数器进行初始化：索引事件、选通事件、软件初始化。

（1）索引事件初始化（IEI）

QEPI 索引输入可用于在索引输入的上升沿或下降沿触发位置计数器初始化。若 QEPCTL[IEI] 位配置为 10，则在索引输入的上升沿，使用 QPOSINIT 寄存器中的值初始化位置计数器（QPOSCNT）。反之，若 QEPCTL [IEI] 位配置为 11，则在索引输入的下降沿执行初始化。

（2）选通事件初始化（SEI）

若 QEPCTL[SEI] 位配置为 10，则在选通输入的上升沿，使用 QPOSINIT 寄存器中的值初始化位置计数器。

若 QEPCTL [SEL] 位配置为 11，则：

正向运行时，在选通输入的上升沿初始化位置计数器；

反向运行时，在选通输入的下降沿初始化位置计数器。

（3）软件初始化（SWI）

向 QEPCTL[SWI] 位写入 1，即可通过软件对位置计数器进行初始化。该位不会自动清零。在该位保持置 1 期间，若再次向其写入 1，位置计数器将被重新初始化。

### 18.3.2.4 EQEP 位置比较单元

eQEP 外设包含一个位置比较单元，用于在位置比较匹配时产生同步输出和中断。图中为其功能框图。位置比较寄存器（QPOSCMP）带有影子寄存器，可以通过 QPOSCTL [PSSHDW] 位来使能或关闭影子模式。如果未使能影子模式，CPU 将直接写入有效位置比较寄存器。

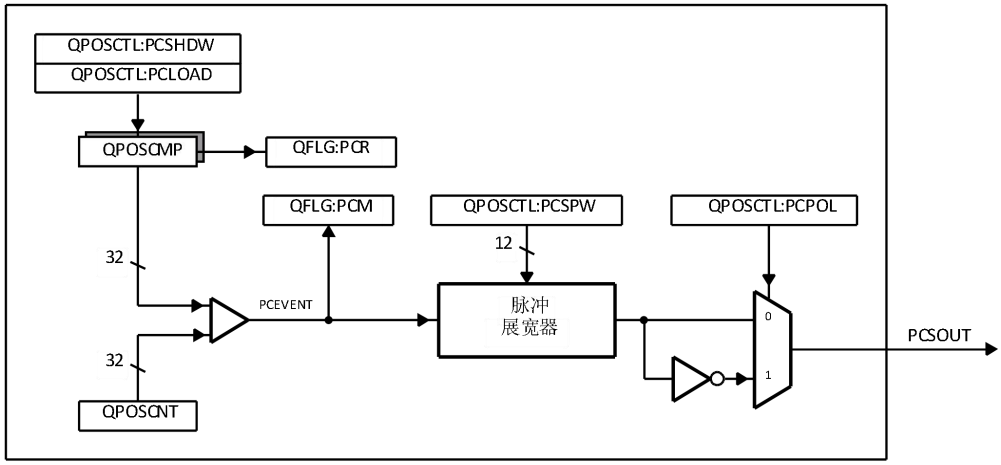



图 18-14 eQEP 位置比较单元

在影子模式下，你可以配置位置比较单元（QPOSCTL [PCLOAD]），使其在以下事件发生时将影子寄存器的值加载到有效寄存器中，并在加载完成后产生位置比较准备就绪中断（QFLG [PCR]）。

位置计数器归零事件时加载：当位置计数器的值（QPOSCNT）与有效位置比较寄存器（QPOSCMP）的值相等时，位置比较匹配标志（QFLG [PCM]）置位；同时在比较匹配时输出可编程脉冲宽度的位置比较同步信号，用于触发外部设备。

例如，若 QPOSCMP = 2，则位置比较单元会在以下时刻产生位置比较事件：

正向计数时，eQEP 位置计数器从 1 变为 2

反向计数时，eQEP 位置计数器从 3 变为 2。

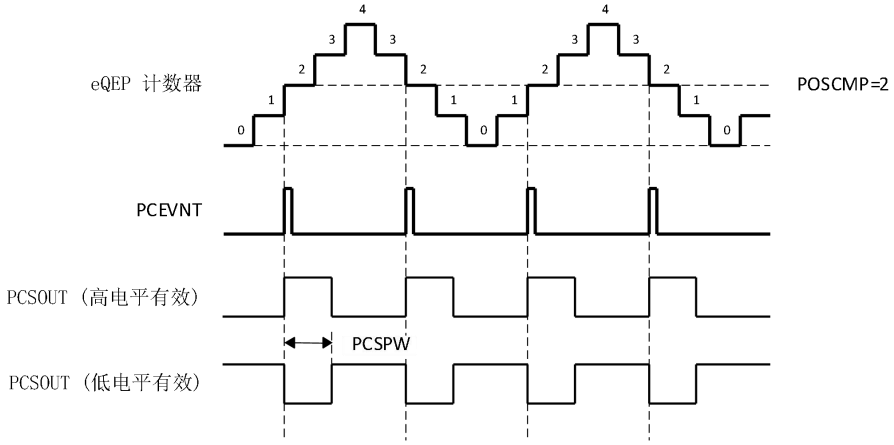



图 18-15 eQEP 位置比较事件生成点

位置比较单元中的脉冲展宽逻辑会在位置比较匹配时，生成一个可编程宽度的位置比较同步脉冲输出。如果上一个位置比较脉冲仍在有效状态时，发生了新的位置比较匹配事件，则脉冲展宽逻辑会从新的位置比较事件开始，生成指定时长的脉冲，如图所示。

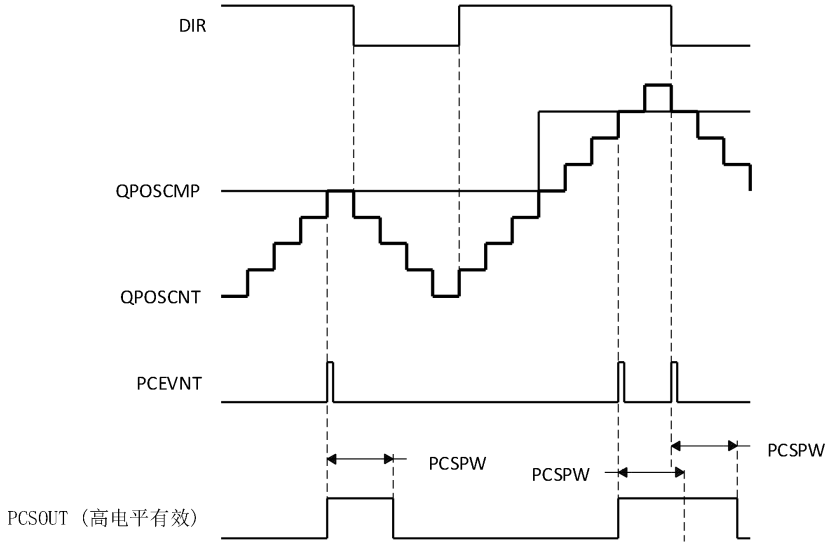



图 18-16 eQEP 位置比较同步输出脉冲扩展器

### 18.3.3 EQEP 边沿捕获单元

eQEP 外设包含一个集成的边沿捕获单元，用于测量单位位置事件之间的时间间隔，如图所示。该功能通常用于低速测量，计算公式如下：

$$v(k) = \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \qquad (18\text{-}3)$$

式中：

X = 单位位置，由正交边沿的整数倍定义（见图）

ΔT = 单位位置事件之间的经过时间

v(k) = 时刻 k 对应的瞬时转速

eQEP 捕获定时器（QCTMR）由预分频后的系统时钟 SYSCLKOUT 驱动，其预分频系数通过 QCAPCTL [CCPS] 位配置。

每次发生单位位置事件时，捕获定时器（QCTMR）的值会被锁存到捕获周期寄存器（QCPRD），随后捕获定时器复位，并在 QEPSTS 中置位 UPEVNT 标志位，

以指示新值已锁存至 QCPRD 寄存器。软件在读取周期寄存器进行低速测量前可检查该状态标志，并通过写 1 清除该标志。

单位位置事件之间的时间测量值（ΔT）在满足以下条件时有效：

单位位置事件之间的计数值不超过 65535。

单位位置事件之间未发生方向变化。

若 eQEP 捕获定时器在单位位置事件之间发生溢出，定时器会在状态寄存器中置位捕获溢出标志（QEPSTS [COEF]），并将 QCPRDLAT 寄存器置为 0xFFFF。若单位位置事件之间发生方向改变，状态寄存器中的错误标志（QEPSTS [CDEF]）会被置位，同时 QCPRDLAT 寄存器置为 0xFFFF。

捕获定时器（QCTMR）和捕获周期寄存器（QCPRD）可配置为在以下事件时锁存：

CPU 读取 QPOSCNT 寄存器

单位超时事件

若 QEPCTL [QCLM] 位清零，则当 CPU 读取位置计数器（QPOSCNT）时，捕获定时器与捕获周期的值会分别锁存到 QCTMRLAT 和 QCPRDLAT 寄存器。

若 QEPCTL [QCLM] 位置 1，则在发生单位超时时，位置计数器、捕获定时器和捕获周期的值会分别锁存到 QPOSLAT、QCTMRLAT 和 QCPRDLAT 寄存器。

图中展示了捕获单元与位置计数器的协同工作过程。

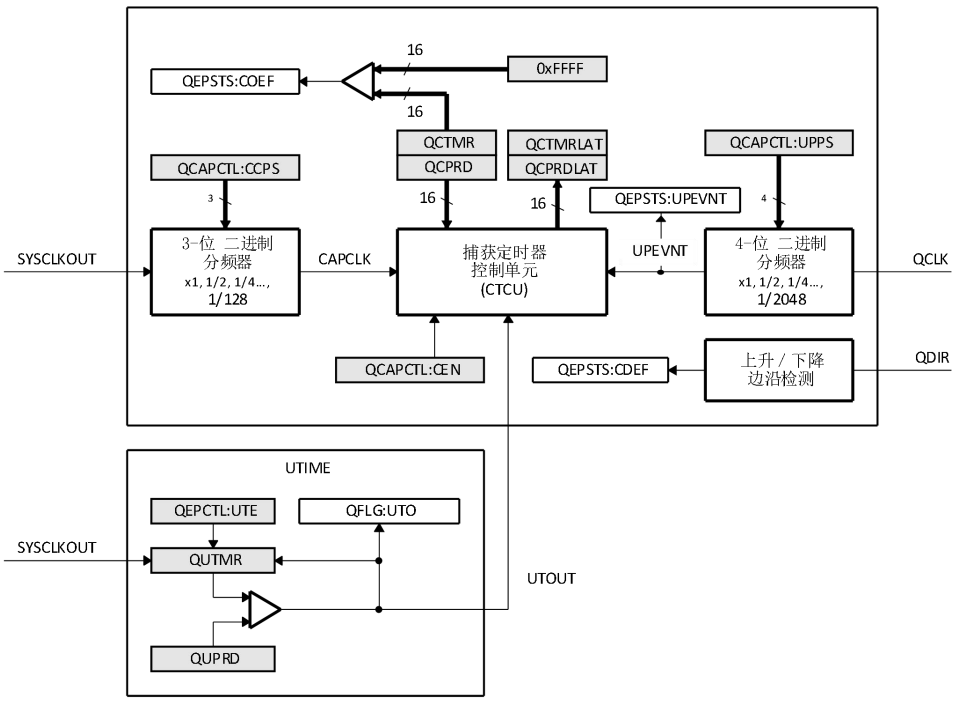



图 18-17 eQEP 边沿捕获单元

QCAPCTL [UPPS] 预分频器不可以动态修改（例如将单位事件预分频从 QCLK/4 切换为 QCLK/8）。这样做会导致未定义行为。QCAPCTL [CPPS] 预分频器仅可在捕获单元禁用后动态修改（例如将 CAPCLK 预分频模式从 SYSCLK/4 切换为 SYSCLK/8）。

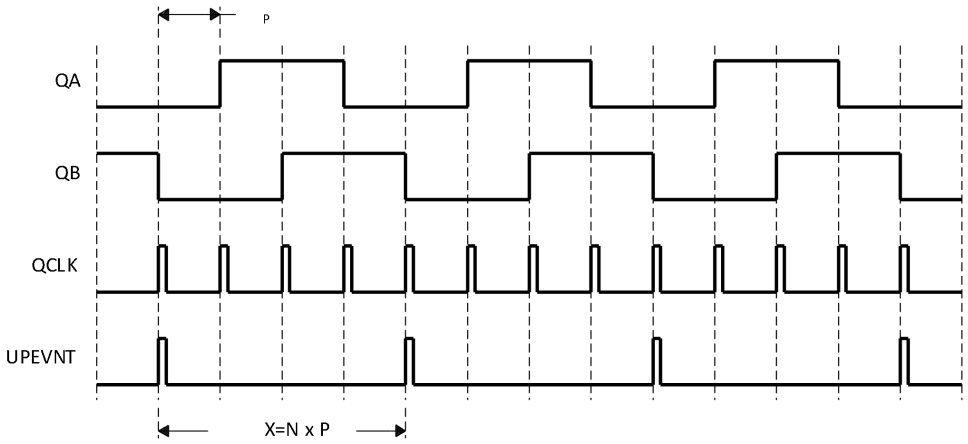



图 18-18 低速测量单元位置事件（qcapctl[ UPPS ] = 0010）

N = 通过 qcapctl[ UPPS ]位选择的正交周期数量

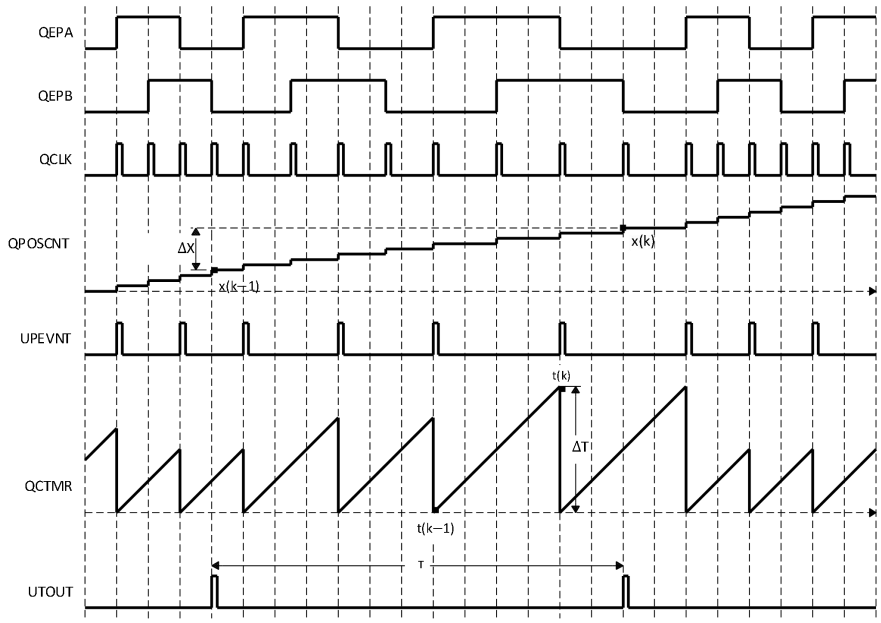



图 18-19 eQEP 边缘捕获单元 - 时序细节

速度计算方程：

$$v(k) = \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \qquad (18\text{-}4)$$

式中：

v(k) = 时刻 k 的瞬时速度

x(k) = 时刻 k 的位置

x(k-1) = 时刻 k-1 的位置

T = 固定单位时间，或速度计算频率的倒数

ΔX = 单位时间内的位置增量

X = 固定单位位置

ΔT = 单位位置移动所用的时间增量

t(k) = 时刻 k

t(k-1) = 时刻 k-1

单位时间(T)和单位周期(X)通过 QUPRD 和 qcapctl[ UPPS ]寄存器进行配置。增量位置输出和增量时间输出可通过 qposlat 和 qcprdlat 寄存器获取。

表 18-3 EQEP 边沿捕获单元数据表

| 参数 | 配置或者计数的相关寄存器 |
|---|---|
| T | 单元周期寄存器（QUPRD） |
| ΔX | 增量位置 = QPOSLAT (k) - QPOSLAT (K-1) |
| X | 由传感器分辨率和QCAPCTL[ UPPS ]位定义的固定单元位置 |
| ΔT | 捕获周期锁存器(QCPRDLAT) |

## 18.3.4 EQEP 看门狗

eQEP 外设包含一个 16 位看门狗定时器，用于监测正交时钟，以反映运动控制系统是否正常工作。eQEP 看门狗定时器由 SYSCLKOUT/64 提供时钟，而正交时钟事件（脉冲）会复位该看门狗定时器。如果直到周期匹配（QWDPRD = QWDTMR）仍未检测到正交时钟事件，则看门狗定时器将超时，并置位看门狗中断标志（QFLG [WTO]）。超时值可通过看门狗周期寄存器（QWDPRD）进行编程配置。

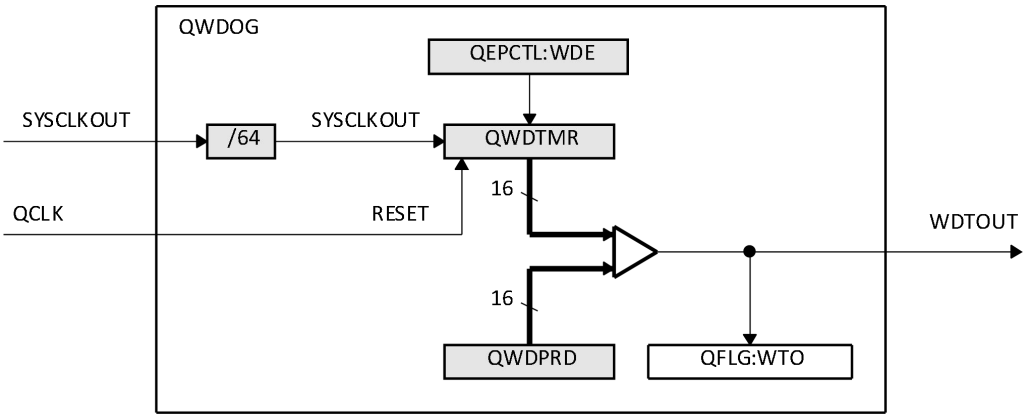



图 18-20 eQEP 看门狗定时器

## 18.3.5 EQEP 单元时间基准

eQEP 外设包含一个 32 位定时器（QUTMR），由系统时钟 SYSCLKOUT 驱动，用于产生周期性中断以进行速度计算，见图。每当单元定时器（QUTMR）的值与单元周期寄存器（QUPRD）的值匹配时，eQEP 外设会将单元定时器（QUTMR）清零，并置位单元超时中断标志（QFLG [UTO]）。每当定时器计数值等于配置的周期值时，单元定时器就会复位。

eQEP 外设可配置为在单元超时事件发生时锁存位置计数器、捕获定时器及捕获周期的值。

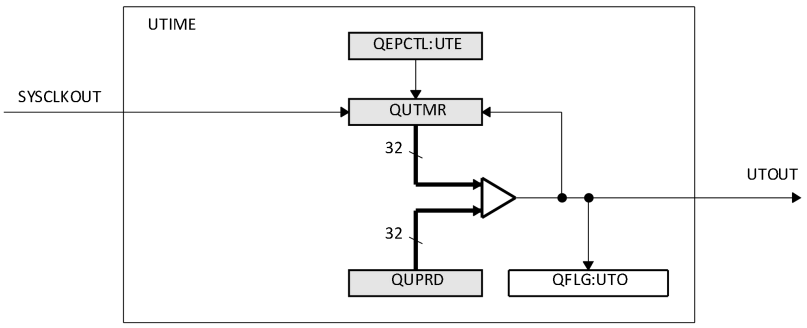



图 18-21 eQEP 单元定时器基准

### 18.3.6 QMA 单元

图中展示了 QMA 模块如何集成到 eQEP 模块中。

复位时，QMA 逻辑默认处于旁路状态，来自引脚的 EQEPA 与 EQEPB 信号直接输入 eQEP 模块。通过配置 QMACTRL [MODE] 寄存器使能 QMA 模块后，EQEPA 和 EQEPB 输入信号将先经该模块处理，处理后的 EQEPA、EQEPB 信号再送入 eQEP 模块。QMA 模块要求 eQEP 模块配置为方向计数模式，并根据目标模式的正常运行需求，在 EQEPA 输入端产生时钟信号、在 EQEPB 输入端产生方向信号。

QMA 模块内部的 xCLKMOD 模块通过检测外部 EQEPA、EQEPB 信号的跳变，为 eQEP 模块的 EQEPA 输入端生成时钟信号。

QMA 模块内部的 xDIRMOD 模块通过检测外部 EQEPA、EQEPB 信号的跳变，为 eQEP 模块的 EQEPB 输入端生成方向信号。

QMA 模块包含错误检测逻辑，可检测 EQEPA、EQEPB 输入信号上的非法跳变。QMA 模块的错误与中断功能已集成在 eQEP 模块内部。此外，可通过 QMALOCK 寄存器锁定 QMACTRL 寄存器的配置。

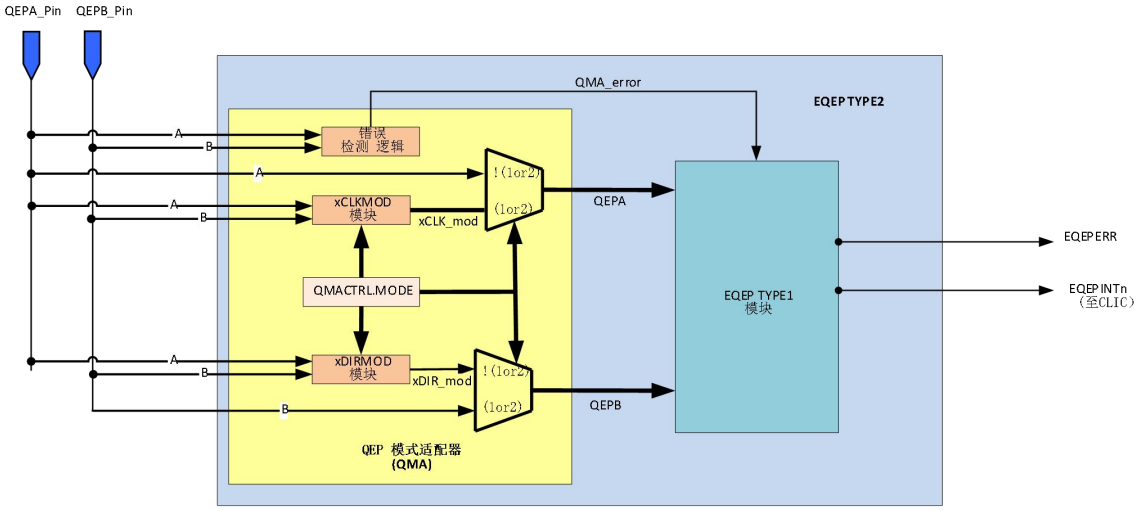



图 18-22 QMA 模块框图

### 18.3.6.1 模块操作

通过配置 QMACTRL 寄存器，QMA 模块可工作在以下模式：

QMA 模式 1（QMACTRL [MODE]=1）

QMA 模式 2（QMACTRL [MODE]=2）

QMA 模式 1 (QMACTRL[MODE]=1)

该模式适用于 EQEPA 与 EQEPB 输入默认状态为高电平的场合。在此模式下，QMA 输出对应关系如图所示：

QMA 输出的 EQEPA = 来自引脚的 EQEPA 输入 与 EQEPB 输入 相 “与”

QMA 输出的 EQEPB = QMA 根据 EQEPA、EQEPB 输入生成的方向信号

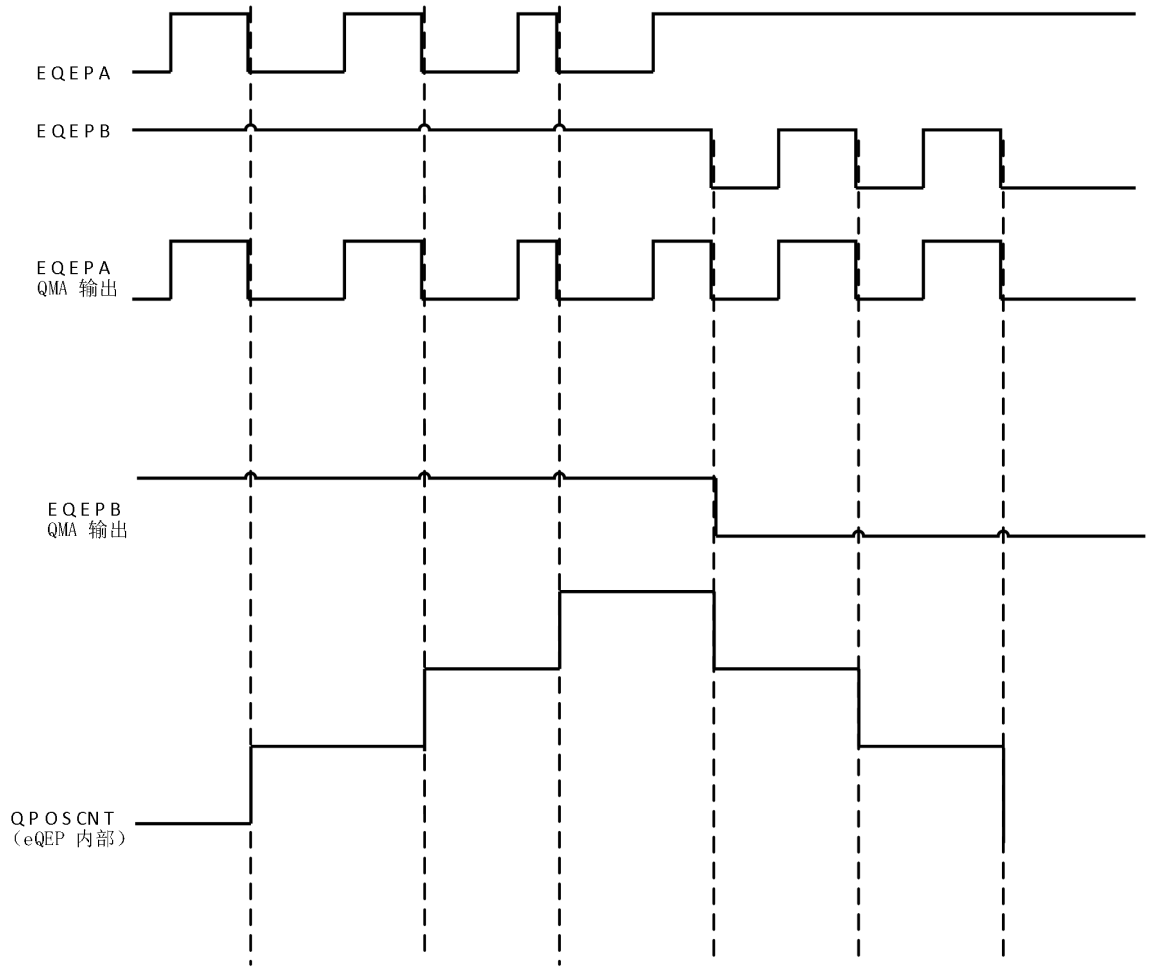



图 18-23 QMA 模式 1

QMA 模式 2 (QMACTRL[MODE]=2)

该模式适用于 EQEPA 与 EQEPB 输入默认状态为低电平 的场合。在此模式下，QMA 的输出对应关系如图所示：

QMA 输出的 EQEPA 为来自引脚的 EQEPA 输入与 EQEPB 输入相或

QMA 输出的 EQEPB 为 QMA 根据 EQEPA、EQEPB 输入生成的方向信号

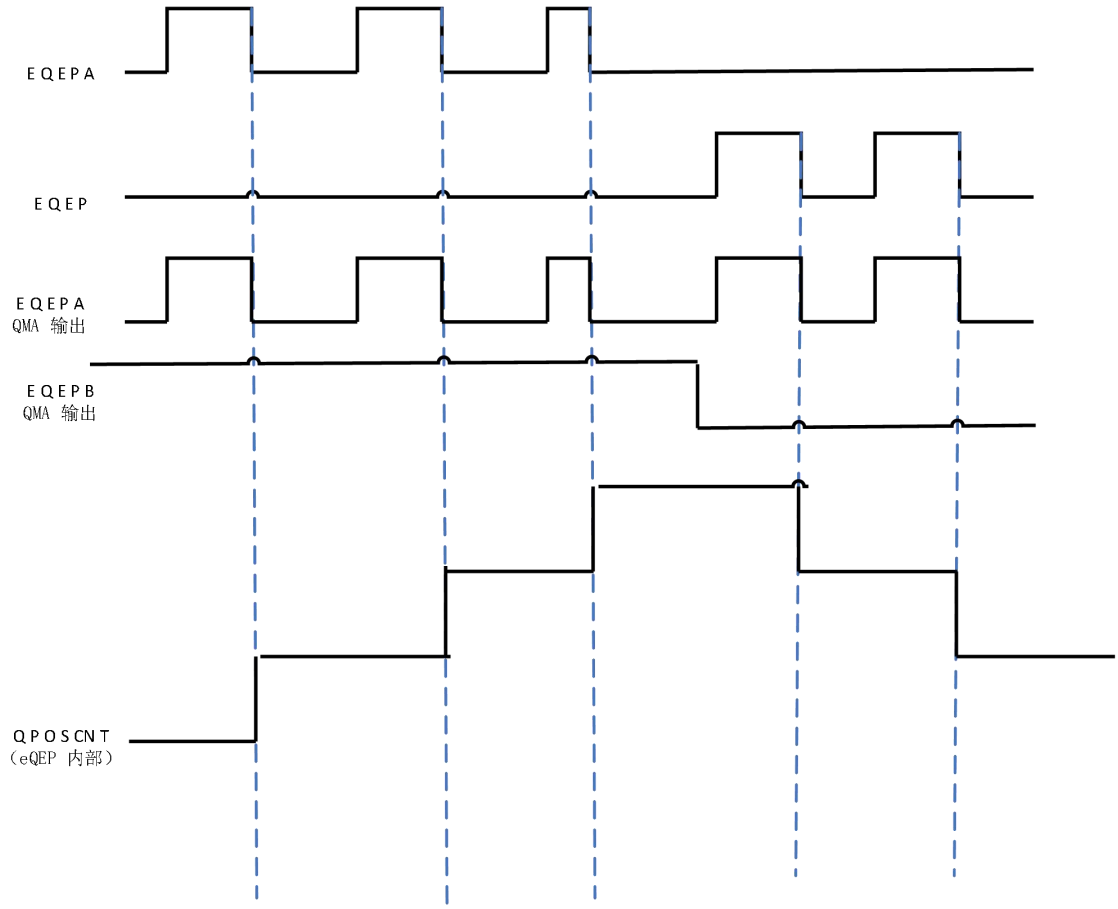



图 18-24 QMA 模式 2

### 18.3.6.2 中断和错误产生

错误检测逻辑会检测 EQEPA 和 EQEPB 信号上的非法跳变，并生成错误信号。该错误信号可用于触发 eQEP 中断和错误输出。

## 18.3.7 EQEP 中断结构

图中展示了 eQEP 模块中中断机制的工作原理。

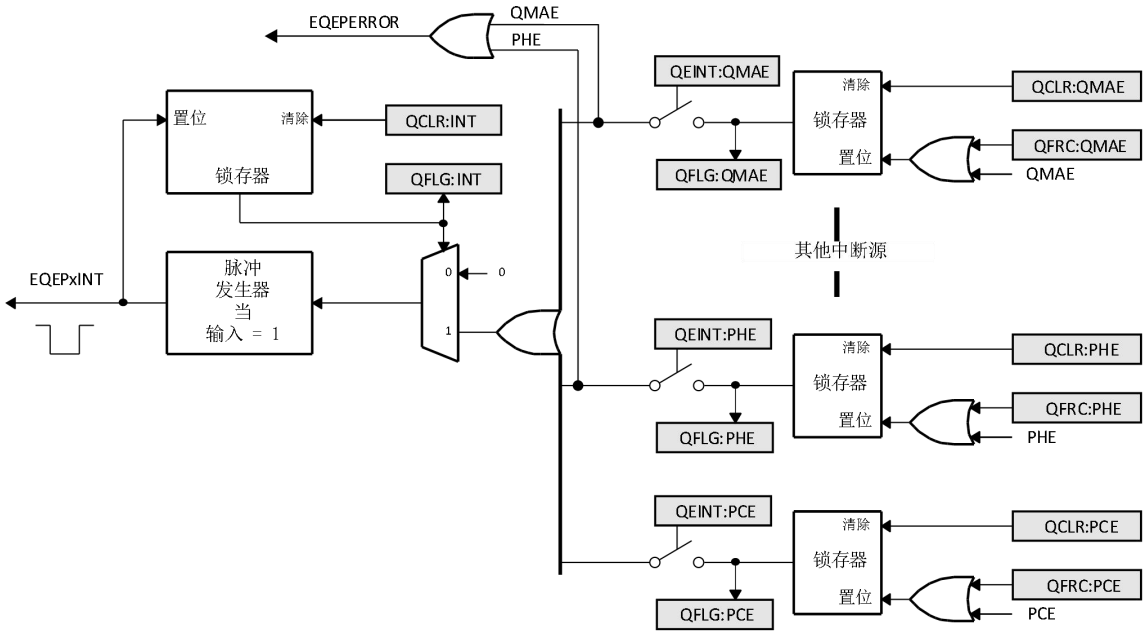



图 18-25 eQEP 中断生成

可产生 11 种中断事件：PCE、PHE、QDC、WTO、PCU、PCO、PCR、PCM、SEL、IEL 及 UTO。中断控制寄存器（QEINT） 用于使能 / 禁止各中断事件源。中断标志寄存器（QFLG） 用于指示是否有中断事件被锁存，并包含全局中断标志位（INT）。

满足以下条件时，会向 CLIC 产生中断脉冲：

- 在 QEINT 寄存器中已使能对应 eQEP 事件的中断；
- QFLG 寄存器中对应 eQEP 事件的中断标志位已置位；

针对上一次产生的中断事件，全局中断状态标志位 QFLG [INT] 已被清零。

在产生其他中断脉冲之前，中断服务程序必须通过中断清除寄存器（QCLR），清除全局中断标志位以及已处理事件的标志位。如果 QFLG 寄存器内的任一标志位未被清除，后续中断事件将不会向 CLIC 发出中断请求。

可以通过 中断强制寄存器（QFRC）强制触发中断事件，该功能常用于测试。

## 18.4 寄存器描述

### 18.4.1 EQEP 基地址表

表 18-4 EQEP 基地址表

| 位域名称 | | DriverLib 名称 | 基地址 | 流水线保护 |
|---|---|---|---|---|
| 实例 | 结构 | | | |
| EQep1Regs | EQEP1寄存器 | EQEP1_BASE | 0x61_8000 | - |
| EQep2Regs | EQEP2寄存器 | EQEP2_BASE | 0x61_8080 | - |

### 18.4.2 EQEP_REGS 寄存器

表中列出了 EQEP_REGS 寄存器的内存映射寄存器。表中未列出的所有寄存器偏移地址均应视为保留地址，且不得修改寄存器内容。

表 18-5 EQEP_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| **00h** | QPOSCNT | 位置计数器 |
| **04h** | QPOSINIT | 位置计数器初始值 |
| **08h** | QPOSMAX | 最大位置计数值 |
| **0Ch** | QPOSCMP | 位置比较值 |
| **10h** | QPOSILAT | 索引位置锁存 |
| **14h** | QPOSSLAT | 选通位置锁存 |
| **18h** | QPOSLAT | 位置锁存 |
| **1Ch** | QUTMR | QEP 单元定时器 |
| **20h** | QUPRD | QEP 单元周期 |
| **24h** | QWDTMR | QEP 看门狗定时器 |
| **26h** | QWDPRD | QEP 看门狗周期 |
| **28h** | QDECCTL | 正交解码控制 |
| **2Ah** | QEPCTL | QEP 控制 |
| **2Ch** | QCAPCTL | 正交捕获控制 |
| **2Eh** | QPOSCTL | 位置比较控制 |
| **30h** | QEINT | QEP 中断控制 |
| **32h** | QFLG | QEP 中断标志 |
| **34h** | QCLR | QEP 中断清除 |
| **36h** | QFRC | QEP 中断强制 |
| **38h** | QEPSTS | QEP 状态 |
| **3Ah** | QCTMR | QEP 捕获定时器 |
| **3Ch** | QCPRD | QEP 捕获周期 |
| **3Eh** | QCTMRLAT | QEP 捕获锁存 |
| **40h** | QCPRDLAT | QEP 捕获周期锁存 |
| **60h** | REV | QEP 版本号 |
| **64h** | QEPSTROBESEL | QEP 选通选择寄存器 |
| **68h** | QMACTRL | QMA 控制寄存器 |
| **6Ch** | QEPSRCSEL | QEP 信号源选择寄存器 |

### 18.4.2.1 QPOSCNT 寄存器

偏移地址：0h　复位值：0000h

QPOSCNT 寄存器的各字段说明如表中所示。

表 18-6 QPOSCNT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| QPOSCNT | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QPOSCNT | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QPOSCNT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QPOSCNT | | | | | | | |
| R/W-0h | | | | | | | |

表 18-7 QPOSCNT 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | QPOSCNT | R/W | 0h | 位置计数器该 32 位位置计数器寄存器根据方向输入，在每个 eQEP 脉冲时进行加 / 减计数。该计数器作为位置积分器，其计数值与相对于给定参考点的位置成正比。当计数器正在加 / 减计数时，该寄存器为只读。注意：建议仅在初始化期间对位置计数器寄存器（QPOSCNT）执行写操作，即 eQEP 位置计数器禁用时（QEPCTL 的 QPEN 位为 0）。一旦位置计数器使能（QPEN 位为 1），写入 eQEP 位置计数器寄存器可能会导致不可预期的结果。<br>复位类型：SYSRSn |

### 18.4.2.2 QPOSINIT 寄存器

偏移地址：4h　复位值：0000h

QPOSINIT 寄存器的各字段说明如表中所示。

表 18-8 QPOSINIT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| QPOSINIT | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QPOSINIT | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QPOSINIT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QPOSINIT | | | | | | | |
| R/W-0h | | | | | | | |

表 18-9 QPOSINIT 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | QPOSINIT | R/W | 0h | 位置计数器初始值该寄存器存储用于根据外部选通或索引事件初始化位置计数器的位置值。也可通过软件初始化位置计数器。对该寄存器的写入必须始终为完整 32 位写入。<br>复位类型：SYSRSn |

### 18.4.2.3 QPOSMAX 寄存器

偏移地址：8h　复位值：0000h

QPOSMAX 寄存器的各字段说明如表中所示。

表 18-10 QPOSMAX 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| QPOSMAX | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QPOSMAX | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QPOSMAX | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QPOSMAX | | | | | | | |
| R/W-0h | | | | | | | |

表 18-11 QPOSMAX 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | QPOSMAX | R/W | 0h | 最大位置计数值该寄存器保存位置计数器的最大值。对该寄存器的写入必须始终为完整 32 位写入。复位类型：SYSRSn |

### 18.4.2.4 QPOSCMP 寄存器

偏移地址：Ch　复位值：0000h

QPOSCMP 寄存器的各字段说明如表中所示。

表 18-12 QPOSCMP 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| QPOSCMP | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QPOSCMP | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QPOSCMP | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QPOSCMP | | | | | | | |
| R/W-0h | | | | | | | |

表 18-13 QPOSCMP 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | QPOSCMP | R/W | 0h | 位置比较值该寄存器中的位置比较值与位置计数器（QPOSCNT）进行比较，比较匹配时产生同步输出和 / 或中断。对该寄存器的写入必须始终为完整 32 位写入。<br>复位类型：SYSRSn |

### 18.4.2.5 QPOSILAT 寄存器

偏移地址：10h　复位值：0000h

QPOSILAT 寄存器的各字段说明如表中所示。

表 18-14 QPOSILAT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| QPOSILAT | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QPOSILAT | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QPOSILAT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QPOSILAT | | | | | | | |
| R-0h | | | | | | | |

表 18-15 QPOSILAT 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | QPOSILAT | R | 0h | 索引位置锁存当发生由 QEPCTL [IEL] 位定义的索引事件时，位置计数器的值会被锁存到该寄存器中。复位类型：SYSRSn |

### 18.4.2.6 QPOSSLAT 寄存器

偏移地址：14h　复位值：0000h

QPOSSLAT 寄存器的各字段说明如表中所示。

表 18-16 QPOSSLAT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| QPOSSLAT | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QPOSSLAT | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QPOSSLAT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QPOSSLAT | | | | | | | |
| R-0h | | | | | | | |

表 18-17 QPOSSLAT 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | QPOSSLAT | R | 0h | 选通位置锁存在由 QEPCTL [SEL] 位定义的选通事件触发时，位置计数器的值会被锁存至该寄存器。复位类型：SYSRSn |

### 18.4.2.7 QPOSLAT 寄存器

偏移地址：18h　复位值：0000h

QPOSLAT 寄存器的各字段说明如表中所示。

表 18-18 QPOSLAT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| QPOSLAT | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QPOSLAT | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QPOSLAT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QPOSLAT | | | | | | | |
| R-0h | | | | | | | |

表 18-19 QPOSLAT 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | QPOSLAT | R | 0h | 位置锁存单元定时器溢出事件时，位置计数器的值被锁存到该寄存器中。复位类型：SYSRSn |

### 18.4.2.8 QUTMR 寄存器

偏移地址：1Ch　复位值：0000h

QUTMR 寄存器的各字段说明如表中所示。

表 18-20 QUTMR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| QUTMR | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QUTMR | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QUTMR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QUTMR | | | | | | | |
| R/W-0h | | | | | | | |

表 18-21 QUTMR 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | QUTMR | R/W | 0h | QEP 单元定时器该寄存器作为单元时间事件生成的时基。当定时器值与单元时间周期值匹配时，产生单元时间事件。对该寄存器的写入必须始终为完整 32 位写入。复位类型：SYSRSn |

### 18.4.2.9 QUPRD 寄存器

偏移地址：20h　复位值：0000h

QUPRD 寄存器的各字段说明如表中所示。

表 18-22 QUPRD 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| QUPRD | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| QUPRD | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QUPRD | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QUPRD | | | | | | | |
| R/W-0h | | | | | | | |

表 18-23 QUPRD 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | QUPRD | R/W | 0h | QEP 单元周期该寄存器包含单元定时器用于生成周期性单元时间事件的周期计数值。这些事件会周期性锁存 eQEP 位置信息，并可选择产生中断。对该寄存器的写入必须始终为完整 32 位写入。复位类型：SYSRSn |

### 18.4.2.10 QWDTMR 寄存器

偏移地址：24h　复位值：0000h

QWDTMR 寄存器的各字段说明如表中所示。

表 18-24 QWDTMR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| QWDTMR | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QWDTMR | | | | | | | |
| R/W-0h | | | | | | | |

表 18-25 QWDTMR 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-0 | QWDTMR | R/W | 0h | QEP 看门狗定时器该寄存器作为看门狗的时基，用于检测电机堵转。当定时器值与看门狗周期值匹配时，产生看门狗超时中断。当正交时钟出现边沿跳变（表明电机运动）时，该寄存器会被复位。复位类型：SYSRSn |

### 18.4.2.11 QWDPRD 寄存器

偏移地址：26h　复位值：0000h

QWDPRD 寄存器的各字段说明如表中所示。

表 18-26 QWDPRD 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| QWDPRD | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QWDPRD | | | | | | | |
| R/W-0h | | | | | | | |

表 18-27 QWDPRD 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-0 | QWDPRD | R/W | 0h | QEP 看门狗周期该寄存器包含 eQEP 外设看门狗定时器的溢出计数值。当看门狗定时器的值与看门狗周期值匹配时，产生看门狗超时中断。复位类型：SYSRSn |

### 18.4.2.12 QDECCTL 寄存器

偏移地址：28h　复位值：0000h

QDECCTL 寄存器的各字段说明如表中所示。

表 18-28 QDECCTL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| QSRC | | SOEN | SPSEL | XCR | SWAP | IGATE | QAP |
| R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QBP | QIP | QSP | 保留 | | | | |
| R/W-0h | R/W-0h | R/W-0h | R-0h | | | | |

表 18-29 QDECCTL 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-14 | QSRC | R/W | 0h | 位置计数器源选择<br>0h (R/W)= 正交计数模式 (QCLK=iCLK,QDIR=iDIR)<br>1h (R/W)= 方向计数模式 (QCLK=xCLK,QDIR=xDIR)<br>2h (R/W)= 频率测量向上计数模式 (QCLK=xCLK,QDIR=1)<br>3h (R/W)= 频率测量向下计数模式 (QCLK=xCLK,QDIR=0)<br>复位类型：SYSRSn |
| 13 | SOEN | R/W | 0h | 同步输出使能<br>0h (R/W)= 禁用位置比较同步输出<br>1h (R/W)= 启用位置比较同步输出<br>复位类型：SYSRSn |
| 12 | SPSEL | R/W | 0h | 同步输出引脚选择<br>0h (R/W)= 索引引脚用于同步输出<br>1h (R/W)= 选通引脚用于同步输出<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 11 | XCR | R/W | 0h | 外部时钟速率<br>0h (R/W)=2 倍分辨率：计数上升 / 下降沿<br>1h (R/W)=1 倍分辨率：仅计数上升沿<br>复位类型：SYSRSn |
| 10 | SWAP | R/W | 0h | 位置计数器 CLK/DIR 信号源<br>0h (R/W)= 正交时钟输入不交换<br>1h (R/W)= 正交时钟输入交换<br>复位类型：SYSRSn |
| 9 | IGATE | R/W | 0h | 索引脉冲门控选项<br>0h (R/W)= 禁用索引脉冲门控<br>1h (R/W)= 通过选通信号门控索引引脚<br>复位类型：SYSRSn |
| 8 | QAP | R/W | 0h | QEPA 输入极性<br>0h (R/W)= 无效果<br>1h (R/W)= 取反 QEPA 输入<br>复位类型：SYSRSn |
| 7 | QBP | R/W | 0h | QEPB 输入极性<br>0h (R/W)= 无效果<br>1h (R/W)= 取反 QEPB 输入<br>复位类型：SYSRSn |
| 6 | QIP | R/W | 0h | QEPI 输入极性<br>0h (R/W)= 无效果<br>1h (R/W)= 取反 QEPI 输入<br>复位类型：SYSRSn |
| 5 | QSP | R/W | 0h | QEPS 输入极性<br>0h (R/W)= 无效果<br>1h (R/W)= 取反 QEPS 输入<br>复位类型：SYSRSn |
| 4-0 | RESERVED | R | 0h | 保留 |

### 18.4.2.13 QEPCTL 寄存器

偏移地址：2Ah 复位值：0000h

QEPCTL 寄存器的各字段说明如表中所示。

表 18-30 QEPCTL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| FREE_SOFT | | PCRM | | SEI | | IEI | |
| R/W-0h | | R/W-0h | | R/W-0h | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SWI | SEL | IEL | | QPEN | QCLM | UTE | WDE |
| R/W-0h | R/W-0h | R/W-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 18-31 QEPCTL 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-14 | FREE_SOFT | R/W | 0h | 仿真模式<br>0h (R/W)= 仿真暂停时，位置计数器、看门狗、单元定时器、捕获定时器立即停止<br>1h (R/W)= 仿真暂停时，各计数器继续计数直至溢出 / 周期匹配<br>2h (R/W)= 仿真暂停对各计数器无影响<br>3h (R/W)= 与 FREE_SOFT_2 相同<br>复位类型：SYSRSn |
| 13-12 | PCRM | R/W | 0h | 位置计数器复位模式<br>0h (R/W)= 索引事件复位位置计数器<br>1h (R/W)= 到达最大位置时复位位置计数器<br>2h (R/W)= 首次索引事件复位位置计数器<br>3h (R/W)= 单元时间事件复位位置计数器<br>复位类型：SYSRSn |
| 11-10 | SEI | R/W | 0h | 选通事件初始化位置计数器<br>0h,1h (R/W)= 无动作（功能禁用）<br>2h (R/W)= 在 QEPS 信号上升沿初始化位置计数器<br>3h (R/W)= 顺时针方向：QEPS 上升沿初始化；逆时针方向：QEPS 下降沿初始化<br>复位类型：SYSRSn |
| 9-8 | IEI | R/W | 0h | 索引事件初始化位置计数器<br>0h,1h (R/W)= 无动作（功能禁用）<br>2h (R/W)= 在 QEPI 信号上升沿初始化位置计数器 (QPOSCNT=QPOSINIT)<br>3h (R/W)= 在 QEPI 信号下降沿初始化位置计数器 (QPOSCNT=QPOSINIT)<br>复位类型：SYSRSn |
| 7 | SWI | R/W | 0h | 软件初始化位置计数器<br>0h (R/W)= 无动作（功能禁用）<br>1h (R/W)= 初始化位置计数器 (QPOSCNT=QPOSINIT)，该位不会自动清零 |

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| | | | | 复位类型：SYSRSn |
| 6 | SEL | R/W | 0h | 选通事件锁存位置计数器<br>0h (R/W)= 在 QEPS 选通信号上升沿锁存位置计数器 (QPOSSLAT=QPOSCNT)<br>1h (R/W)= 顺时针方向：QEPS 上升沿锁存；逆时针方向：QEPS 下降沿锁存<br>复位类型：SYSRSn |
| 5-4 | IEL | R/W | 0h | 索引事件锁存位置计数器（软件索引标记）<br>0h (R/W)= 保留<br>1h (R/W)= 在索引信号上升沿锁存位置计数器2h (R/W)= 在索引信号下降沿锁存位置计数器3h (R/W)= 软件索引标记，索引事件时锁存位置计数器与方向标志<br>复位类型：SYSRSn |
| 3 | QPEN | R/W | 0h | 正交位置计数器使能 / 软件复位<br>0h (R/W)= 复位 eQEP 内部运行标志与只读寄存器，控制配置寄存器不受影响<br>1h (R/W)= 使能 eQEP 位置计数器<br>复位类型：SYSRSn |
| 2 | QCLM | R/W | 0h | QEP 捕获锁存模式<br>0h (R/W)=CPU 读取 QPOSCNT 时锁存捕获定时器 / 周期值<br>1h (R/W)= 单元定时器溢出时锁存位置、捕获定时器及周期值<br>复位类型：SYSRSn |
| 1 | UTE | R/W | 0h | QEP 单元定时器使能<br>0h (R/W)= 禁用 eQEP 单元定时器<br>1h (R/W)= 使能单元定时器<br>复位类型：SYSRSn |
| 0 | WDE | R/W | 0h | QEP 看门狗使能<br>0h (R/W)= 禁用 eQEP 看门狗定时器<br>1h (R/W)= 使能 eQEP 看门狗定时器<br>复位类型：SYSRSn |

### 18.4.2.14 QCAPCTL 寄存器

偏移地址：2Ch 复位值：0000h

QCAPCTL 寄存器的各字段说明如表中所示。

表 18-32 QCAPCTL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CEN | 保留 | | | | | | |
| R/W-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | CCPS | | | UPPS | | | |
| R-0h | R/W-0h | | | R/W-0h | | | |

表 18-33 QCAPCTL 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15 | CEN | R/W | 0h | eQEP 捕获单元使能<br>0h (R/W)= 禁用 eQEP 捕获单元<br>1h (R/W)= 使能 eQEP 捕获单元复位类型：SYSRSn |
| 14-7 | RESERVED | R | 0h | 保留复位类型：SYSRSn |
| 6-4 | CCPS | R/W | 0h | eQEP 捕获定时器时钟预分频<br>0h(R/W)=CAPCLK=SYSCLKOUT/1<br>1h(R/W)=CAPCLK=SYSCLKOUT/2<br>2h(R/W)=CAPCLK=SYSCLKOUT/4<br>3h(R/W)=CAPCLK=SYSCLKOUT/8<br>4h(R/W)=CAPCLK=SYSCLKOUT/16<br>5h(R/W)=CAPCLK=SYSCLKOUT/32<br>6h(R/W)=CAPCLK=SYSCLKOUT/64<br>7h(R/W)=CAPCLK=SYSCLKOUT/128<br>复位类型：SYSRSn |
| 3-0 | UPPS | R/W | 0h | 单元位置事件预分频<br>0h(R/W)=UPEVNT=QCLK/1<br>1h(R/W)=UPEVNT=QCLK/2<br>2h(R/W)=UPEVNT=QCLK/4<br>3h(R/W)=UPEVNT=QCLK/8<br>4h(R/W)=UPEVNT=QCLK/16<br>5h(R/W)=UPEVNT=QCLK/32<br>6h(R/W)=UPEVNT=QCLK/64<br>7h(R/W)=UPEVNT=QCLK/128<br>8h(R/W)=UPEVNT=QCLK/256<br>9h(R/W)=UPEVNT=QCLK/512<br>Ah(R/W)=UPEVNT=QCLK/1024<br>Bh(R/W)=UPEVNT=QCLK/2048<br>Ch~Fh (R/W)= 保留复位类型：SYSRSn |

### 18.4.2.15 QPOSCTL 寄存器

偏移地址：2Eh　复位值：0000h

QPOSCTL 寄存器的各字段说明如表中所示。

表 18-34 QPOSCTL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| PCSHDW | PCLOAD | PCPOL | PCE | PCSPW | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCSPW | | | | | | | |
| R/W-0h | | | | | | | |

表 18-35 QPOSCTL 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15 | PCSHDW | R/W | 0h | 位置比较影子寄存器使能<br>0h (R/W)= 禁用影子寄存器，立即加载<br>1h (R/W)= 使能影子寄存器<br>复位类型：SYSRSn |
| 14 | PCLOAD | R/W | 0h | 位置比较影子寄存器加载模式<br>0h (R/W)= 当 QPOSCNT=0 时加载<br>1h (R/W)= 当 QPOSCNT=QPOSCMP 时加载<br>复位类型：SYSRSn |
| 13 | PCPOL | R/W | 0h | 同步输出极性<br>0h (R/W)= 高电平有效脉冲输出<br>1h (R/W)= 低电平有效脉冲输出<br>复位类型：SYSRSn |
| 12 | PCE | R/W | 0h | 位置比较功能使能<br>0h (R/W)= 禁用位置比较单元<br>1h (R/W)= 使能位置比较单元<br>复位类型：SYSRSn |
| 11-0 | PCSPW | R/W | 0h | 位置比较同步输出脉冲宽度选择<br>0h (R/W)=1×4×SYSCLKOUT 周期<br>1h (R/W)=2×4×SYSCLKOUT 周期FFFh<br>(R/W)=4096×4×SYSCLKOUT 周期<br>复位类型：SYSRSn |

### 18.4.2.16 QEINT 寄存器

偏移地址：30h 复位值：0000h

QEINT 寄存器的各字段说明如表中所示。

表 18-36 QEINT 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | QMAE | UTO | IEL | SEL | PCM |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCR | PCO | PCU | WTO | QDC | QPE | PCE | 保留 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

表 18-37 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-13 | RESERVED | R | 0h | 保留 |
| 12 | QMAE | R/W | 0h | QMA 错误中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |
| 11 | UTO | R/W | 0h | 单元定时器溢出中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |
| 10 | IEL | R/W | 0h | 索引事件锁存中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |
| 9 | SEL | R/W | 0h | 选通事件锁存中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |
| 8 | PCM | R/W | 0h | 位置比较匹配中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |
| 7 | PCR | R/W | 0h | 位置比较就绪中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |
| 6 | PCO | R/W | 0h | 位置计数器上溢中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 5 | PCU | R/W | 0h | 位置计数器下溢中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |
| 4 | WTO | R/W | 0h | 看门狗超时中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |
| 3 | QDC | R/W | 0h | 正交方向变化中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |
| 2 | QPE | R/W | 0h | 正交相位错误中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |
| 1 | PCE | R/W | 0h | 位置计数器错误中断使能<br>0h (R/W)= 禁用中断<br>1h (R/W)= 使能中断<br>复位类型：SYSRSn |
| 0 | RESERVED | R | 0h | 保留 |

### 18.4.2.17 QFLG 寄存器

偏移地址：32h　复位值：0000h

QFLG 寄存器的各字段说明如表中所示。

表 18-38 QFLG 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | QMAE | UTO | IEL | SEL | PCM |
| R-0h | | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCR | PCO | PCU | WTO | QDC | PHE | PCE | INT |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 18-39 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-13 | RESERVED | R | 0h | 保留复位类型：SYSRSn |
| 12 | QMAE | R | 0h | QMA 错误中断标志0h (R/W)= 未产生中断1h (R/W)= 已产生中断<br>复位类型：SYSRSn |
| 11 | UTO | R | 0h | 单元定时器溢出中断标志0h (R/W)= 未产生中断1h (R/W)=eQEP 单元定时器周期匹配时置位复位类型：SYSRSn |
| 10 | IEL | R | 0h | 索引事件锁存中断标志0h (R/W)= 未产生中断1h (R/W)= 将 QPOSCNT 锁存到 QPOSILAT 后该位置位<br>复位类型：SYSRSn |
| 9 | SEL | R | 0h | 选通事件锁存中断标志0h (R/W)= 未产生中断1h (R/W)= 将 QPOSCNT 锁存到 QPOSSLAT 后该位置位<br>复位类型：SYSRSn |
| 8 | PCM | R | 0h | eQEP 比较匹配事件中断标志0h (R/W)= 未产生中断1h (R/W)= 位置比较匹配时该位置位<br>复位类型：SYSRSn |
| 7 | PCR | R | 0h | 位置比较就绪中断标志0h (R/W)= 未产生中断1h (R/W)= 将影子寄存器值传输到有效位置比较寄存器后该位置位复位类型：SYSRSn |
| 6 | PCO | R | 0h | 位置计数器上溢中断标志0h (R/W)= 未产生中断1h (R/W)= 位置计数器上溢时该位置位复位类型：SYSRSn |
| 5 | PCU | R | 0h | 位置计数器下溢中断标志0h (R/W)= 未产生中断1h (R/W)= 位置计数器下溢时该位置位<br>复位类型：SYSRSn |
| 4 | WTO | R | 0h | 看门狗超时中断标志0h (R/W)= 未产生中断1h (R/W)= 看门狗超时时置位<br>复位类型：SYSRSn |
| 3 | QDC | R | 0h | 正交方向变化中断标志0h (R/W)= 未产生中断1h (R/W)= 已产生中断<br>复位类型：SYSRSn |
| 2 | PHE | R | 0h | 正交相位错误中断标志0h (R/W)= 未产生中断1h (R/W)=QEPA 与 QEPB 同时跳变时置位 |

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| | | | | 复位类型：SYSRSn |
| 1 | PCE | R | 0h | 位置计数器错误中断标志0h (R/W)= 未产生中断1h (R/W)= 位置计数器错误<br>复位类型：SYSRSn |
| 0 | INT | R | 0h | 全局中断状态标志0h (R/W)= 未产生中断1h (R/W)= 已产生中断<br>复位类型：SYSRSn |

### 18.4.2.18 QCLR 寄存器

偏移地址：34h 复位值：0000h

QCLR 寄存器的各字段说明如表中所示。

表 18-40 QCLR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | QMAE | UTO | IEL | SEL | PCM |
| R-0h | | | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCR | PCO | PCU | WTO | QDC | PHE | PCE | INT |
| R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h | R-0/W1S-0h |

表 18-41 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-13 | RESERVED | R | 0h | 保留复位类型：SYSRSn |
| 12 | QMAE | R-0/W1S | 0h | 清除 QMA 错误中断标志0h (R/W)= 无操作1h (R/W)= 清除中断标志复位类型：SYSRSn |
| 11 | UTO | R-0/W1S | 0h | 清除单元定时器溢出中断标志0h (R/W)= 无操作1h (R/W)= 清除中断标志复位类型：SYSRSn |
| 10 | IEL | R-0/W1S | 0h | 清除索引事件锁存中断标志0h (R/W)= 无操作1h (R/W)= 清除中断标志复位类型：SYSRSn |
| 9 | SEL | R-0/W1S | 0h | 清除选通事件锁存中断标志0h (R/W)= 无操作1h (R/W)= 清除中断标志复位类型：SYSRSn |
| 8 | PCM | R-0/W1S | 0h | 清除 eQEP 比较匹配事件中断标志<br>0h (R/W)= 无操作 |

| 位 | 字段 | 类型 | 复位值 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 1h (R/W)= 清除中断标志<br>复位类型：SYSRSn |
| 7 | PCR | R-0/W1S | 0h | 清除位置比较就绪中断标志<br>0h (R/W)= 无操作<br>1h (R/W)= 清除中断标志<br>复位类型：SYSRSn |
| 6 | PCO | R-0/W1S | 0h | 清除位置计数器上溢中断标志<br>0h (R/W)= 无操作<br>1h (R/W)= 清除中断标志<br>复位类型：SYSRSn |
| 5 | PCU | R-0/W1S | 0h | 清除位置计数器下溢中断标志<br>0h (R/W)= 无操作<br>1h (R/W)= 清除中断标志<br>复位类型：SYSRSn |
| 4 | WTO | R-0/W1S | 0h | 清除看门狗超时中断标志<br>0h (R/W)= 无操作<br>1h (R/W)= 清除中断标志<br>复位类型：SYSRSn |
| 3 | QDC | R-0/W1S | 0h | 清除正交方向变化中断标志<br>0h (R/W)= 无操作<br>1h (R/W)= 清除中断标志<br>复位类型：SYSRSn |
| 2 | PHE | R-0/W1S | 0h | 清除正交相位错误中断标志<br>0h (R/W)= 无操作<br>1h (R/W)= 清除中断标志<br>复位类型：SYSRSn |
| 1 | PCE | R-0/W1S | 0h | 清除位置计数器错误中断标志<br>0h (R/W)= 无操作<br>1h (R/W)= 清除中断标志<br>复位类型：SYSRSn |
| 0 | INT | R-0/W1S | 0h | 全局中断清除标志<br>0h (R/W)= 无操作<br>1h (R/W)= 清除中断标志<br>复位类型：SYSRSn |

### 18.4.2.19 QFRC 寄存器

偏移地址：36h　复位值：0000h

QFRC 寄存器的各字段说明如表中所示。

表 18-42 QFRC 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | QMAE | UTO | IEL | SEL | PCM |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCR | PCO | PCU | WTO | QDC | PHE | PCE | 保留 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |

表 18-43 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-13 | RESERVED | R | 0h | 保留复位类型：SYSRSn |
| 12 | QMAE | R/W | 0h | 强制触发 QMA 错误中断<br>0h (R/W)= 无效果<br>1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 11 | UTO | R/W | 0h | 强制触发单元定时器溢出中断<br>0h (R/W)= 无效果<br>1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 10 | IEL | R/W | 0h | 强制触发索引事件锁存中断<br>0h (R/W)= 无效果<br>1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 9 | SEL | R/W | 0h | 强制触发选通事件锁存中断<br>0h (R/W)= 无效果<br>1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 8 | PCM | R/W | 0h | 强制触发位置比较匹配中断<br>0h (R/W)= 无效果<br>1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 7 | PCR | R/W | 0h | 强制触发位置比较就绪中断<br>0h (R/W)= 无效果<br>1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 6 | PCO | R/W | 0h | 强制触发位置计数器上溢中断<br>0h (R/W)= 无效果 |

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| | | | | 1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 5 | PCU | R/W | 0h | 强制触发位置计数器下溢中断<br>0h (R/W)= 无效果<br>1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 4 | WTO | R/W | 0h | 强制触发看门狗超时中断<br>0h (R/W)= 无效果<br>1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 3 | QDC | R/W | 0h | 强制触发正交方向变化中断<br>0h (R/W)= 无效果<br>1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 2 | PHE | R/W | 0h | 强制触发正交相位错误中断<br>0h (R/W)= 无效果<br>1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 1 | PCE | R/W | 0h | 强制触发位置计数器错误中断<br>0h (R/W)= 无效果<br>1h (R/W)= 强制触发中断<br>复位类型：SYSRSn |
| 0 | RESERVED | R | 0h | 保留复位类型：SYSRSn |

### 18.4.2.20 QEPSTS 寄存器

偏移地址：38h　复位值：0000h

QEPSTS 寄存器的各字段说明如表中所示。

表 18-44 QEPSTS 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| UPEVNT | FIDF | QDF | QDLF | COEF | CDEF | FIMF | PCEF |
| R/W1C-0h | R-0h | R-0h | R-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R-0h |

表 18-45 QEPSTS 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留复位类型：SYSRSn |
| 7 | UPEVNT | R/W1C | 0h | 单元位置事件标志<br>0h (R/W)= 未检测到单元位置事件<br>1h (R/W)= 检测到单元位置事件，写 1 清零<br>复位类型：SYSRSn |
| 6 | FIDF | R | 0h | 首次索引标记方向首次索引事件时锁存方向状态<br>0h (R/W)= 首次索引事件时为逆时针旋转（反向运动）<br>1h (R/W)= 首次索引事件时为顺时针旋转（正向运动）复位类型：SYSRSn |
| 5 | QDF | R | 0h | 正交方向标志<br>0h (R/W)= 逆时针旋转（反向运动）<br>1h (R/W)= 顺时针旋转（正向运动）<br>复位类型：SYSRSn |
| 4 | QDLF | R | 0h | eQEP 方向锁存标志<br>0h (R/W)= 索引事件标记时为逆时针旋转（反向运动）<br>1h (R/W)= 索引事件标记时为顺时针旋转（正向运动）<br>复位类型：SYSRSn |
| 3 | COEF | R/W1C | 0h | 捕获溢出错误标志<br>0h (R/W)= 未发生溢出<br>1h (R/W)=eQEP 捕获定时器 (QCTMR) 发生溢出，写 1 清零<br>复位类型：SYSRSn |
| 2 | CDEF | R/W1C | 0h | 捕获方向错误标志<br>0h (R/W)= 未发生捕获方向错误<br>1h (R/W)= 捕获位置事件间发生方向变化，写 1 清零<br>复位类型：SYSRSn |
| 1 | FIMF | R/W1C | 0h | 首次索引标记标志<br>0h (R/W)= 未发生首次索引脉冲<br>1h (R/W)= 首次索引脉冲发生时置位，写 1 清零<br>复位类型：SYSRSn |
| 0 | PCEF | R | 0h | 位置计数器错误标志该位非 sticky，每个索引事件都会更新<br>0h (R/W)= 上次索引跳变期间未发生错误 |

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| | | | | 1h (R/W)= 位置计数器错误<br>复位类型：SYSRSn |

### 18.4.2.21 QCTMR 寄存器

偏移地址：3Ah 复位值：0000h

QCTMR 寄存器的各字段说明如表中所示。

### 18.4.2.22 QCPRD 寄存器

偏移地址：3Ch 复位值：0000h

QCPRD 寄存器的各字段说明如表中所示。

表 18-46 QCPRD 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| QCPRD | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QCPRD | | | | | | | |
| R/W-0h | | | | | | | |

表 18-47 QCPRD 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-0 | QCPRD | R/W | 0h | 该寄存器保存上一次连续 eQEP 位置事件之间的周期计数值<br>复位类型：SYSRSn |

### 18.4.2.23 QCTMRLAT 寄存器

偏移地址：3Eh 复位值：0000h

QCTMRLAT 寄存器的各字段说明如表中所示。

表 18-48 QCTMRLAT 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| QCTMRLAT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QCTMRLAT | | | | | | | |
| R-0h | | | | | | | |

表 18-49 QCTMRLAT 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-0 | QCTMRLAT | R | 0h | eQEP 捕获定时器值可在两种事件下锁存至该寄存器：单元定时器溢出事件、读取 eQEP 位置计数器<br>复位类型：SYSRSn |

### 18.4.2.24 QCPRDLAT 寄存器

偏移地址：40h　复位值：0000h

QCPRDLAT 寄存器的各字段说明如表中所示。

表 18-50 QCPRDLAT 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| QCPRDLAT | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QCPRDLAT | | | | | | | |
| R-0h | | | | | | | |

表 18-51 QCPRDLAT 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 15-0 | QCPRDLAT | R | 0h | eQEP 捕获周期值可在两种事件下锁存至该寄存器：单元定时器溢出事件、读取 eQEP 位置计数器<br>复位类型：SYSRSn |

### 18.4.2.25 REV 寄存器

偏移地址：60h　复位值：0001h

**REV 寄存器的各字段说明如表中所示。**

表 18-52 REV 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-1h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-1h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-1h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | |
| R-0-1h | | | | | | | |

表 18-53 REV 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-0 | RESERVED | R-0 | 1h | EQEP版本信息 |

## 18.4.2.26 QEPSTROBESEL 寄存器

偏移地址：64h　复位值：0000h

**QEPSTROBESEL 寄存器的各字段说明如表中所示。**

表 18-54 QEPSTROBESEL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | STROBESEL | |
| R-0-0h | | | | | | R/W-0h | |

表 18-55 QEPSTROBESEL 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R-0 | 0h | 保留复位类型：SYSRSn |
| 1-0 | STROBESEL | R/W | 0h | 选通信号源选择<br>0h (R/W)= 经过极性多路选择器后的 QEP 选通信号<br>1h (R/W)= 经过极性多路选择器后的 QEP 选通信号<br>2h (R/W)= 经过极性多路选择器后的 QEP 选通信号与 ADCSOCA 相或<br>3h (R/W)= 经过极性多路选择器后的 QEP 选通信号与 ADCSOCB 相或<br>复位类型：SYSRSn |

## 18.4.2.27 QMACTRL 寄存器

偏移地址：68h　复位值：0000h

**QMACTRL 寄存器的各字段说明如表中所示。**

表 18-56 QMACTRL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | MODE | | |
| R-0-0h | | | | | R/W-0h | | |

表 18-57 QMACTRL 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-3 | RESERVED | R-0 | 0h | 保留复位类型：SYSRSn |
| 2-0 | MODE | R/W | 0h | QMA 模式选择<br>000：QMA 模块旁路<br>001：选择 QMA 模式 1 工作 |

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| | | | | 010：选择 QMA 模式 2 工作<br>011：QMA 模块旁路（保留）<br>1xx：QMA 模块旁路（保留）<br>复位类型：SYSRSn |

### 18.4.2.28 QEPSRCSEL 寄存器

偏移地址：6Ch　复位值：0000h

**QEPSRCSEL** 寄存器的各字段说明如表所示。

表 18-58 QEPSRCSEL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| QEPSSEL | | | | QEPISEL | | | |
| R/W-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| QEPBSEL | | | | QEPASEL | | | |
| R/W-0h | | | | R/W-0h | | | |

表 18-59 QEPSRCSEL 寄存器字段描述

| 位 | 字段 | 类型 | 复位值 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R-0 | 0h | 保留复位类型：SYSRSn |
| 15-12 | QEPSSEL | R/W | 0h | QEP 选通信号源选择0x0：来自器件引脚（默认）<br>其他值：固定为 0复位类型：SYSRSn |
| 11-8 | QEPISEL | R/W | 0h | QEP 索引信号源选择<br>0x0：器件引脚（默认）<br>0x1：CMPSS1.CTRIPH<br>0x2：CMPSS2.CTRIPH<br>0x3：CMPSS3.CTRIPH<br>0x4：CMPSS4.CTRIPH<br>0x5~0x8：保留<br>0x9：PWMXBAR.1<br>0xA：PWMXBAR.2 |

| 位 | 字段 | 类型 | 复位值 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 0xB：PWMXBAR.3<br>0xC：PWMXBAR.4<br>0xD：PWMXBAR.5<br>0xE：PWMXBAR.6<br>0xF：PWMXBAR.7<br>注：配置这些位前需先禁用 eQEP，否则可能导致异常行为复位类型：SYSRSn |
| 7-4 | QEPBSEL | R/W | 0h | QEPB 信号源选择<br>0x0：器件引脚（默认）<br>0x1：CMPSS1.CTRIPH<br>0x2：CMPSS2.CTRIPH<br>0x3：CMPSS3.CTRIPH<br>0x4：CMPSS4.CTRIPH<br>0x5~0x8：保留<br>0x9：PWMXBAR.1<br>0xA：PWMXBAR.2<br>0xB：PWMXBAR.3<br>0xC：PWMXBAR.4<br>0xD：PWMXBAR.5<br>0xE：PWMXBAR.6<br>0xF：PWMXBAR.7注：配置这些位前需先禁用 eQEP，否则可能导致异常行为<br>复位类型：SYSRSn |
| 3-0 | QEPASEL | R/W | 0h | QEPA 信号源选择<br>0x0：器件引脚（默认）<br>0x1：CMPSS1.CTRIPH<br>0x2：CMPSS2.CTRIPH<br>0x3：CMPSS3.CTRIPH<br>0x4：CMPSS4.CTRIPH<br>0x5~0x8：保留<br>0x9：PWMXBAR.1<br>0xA：PWMXBAR.2<br>0xB：PWMXBAR.3<br>0xC：PWMXBAR.4<br>0xD：PWMXBAR.5<br>0xE：PWMXBAR.6<br>0xF：PWMXBAR.7<br>注：配置这些位前需先禁用 eQEP，否则可能导致异常行为复位类型：SYSRSn |

# 19 控制器局域网（CAN）

本章介绍控制器局域网（CAN）模块。CAN 是一种串行通信协议，能高效支持分布式实时控制，且可靠性高。该 CAN 模块支持最高 1 Mbit/s 的比特率，符合 ISO11898-1（CAN 2.0B）协议规范。本器件使用名为 DCAN 的 CAN IP（本章以下简称 CAN）。

## 19.1 概述

### 19.1.1 功能特性

CAN 模块具备以下功能特性：

- 兼容 ISO11898-1（CAN 协议规范 2.0 A 和 B 版本）
- 比特率最高可达 1 Mbit/s
- 支持多种时钟源
- 32 个消息对象（本文档中也称为“邮箱”，二者可互换使用），每个消息对象具备以下属性：
  - 可配置为接收或发送模式
  - 支持标准（11 位）或扩展（29 位）标识符
  - 支持可编程标识符接收掩码
  - 支持数据帧和远程帧
  - 可存储 0~8 字节数据
  - 配置和数据 RAM 支持奇偶校验
- 每个消息对象配备独立标识符掩码
- 消息对象支持可编程 FIFO 模式
- 支持用于自测试的可编程环回模式
- 支持调试的挂起模式
- 软件模块复位功能

- 总线关闭状态后，可通过可编程 32 位定时器自动恢复总线连接
- 消息 RAM 奇偶校验机制
- 两条中断线
- 支持直接内存访问（DMA）

注意：

当 CAN 比特时钟为 100 MHz 时，最小比特率可达 3.90625 kbit/s。

根据所用定时设置，片上零引脚振荡器的精度可能无法满足 CAN 协议要求，此时需使用外部时钟源。

### 19.1.2 框图

图中为 CAN 模块的框图，以下对部分主要模块进行说明。

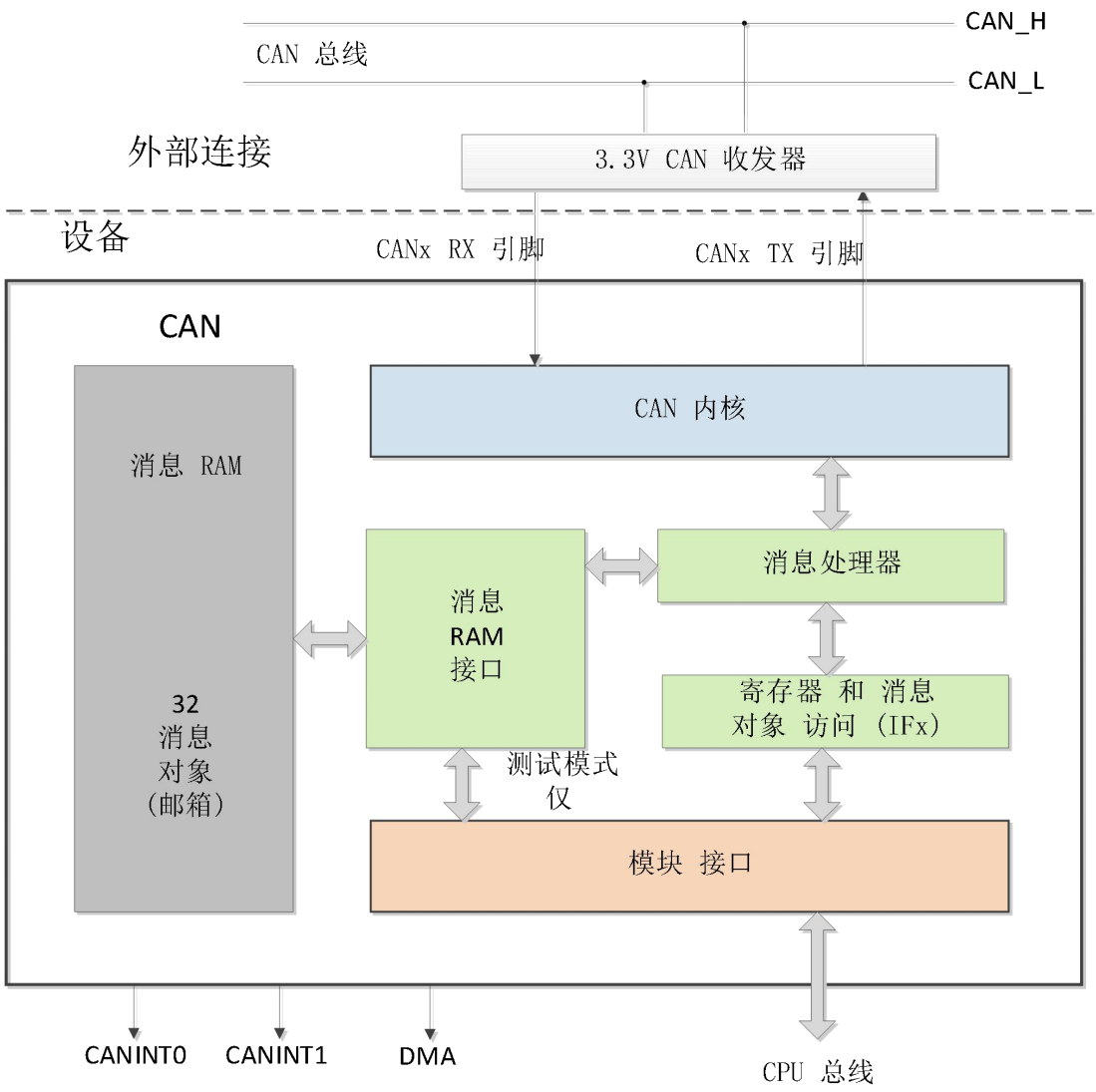



图 19-1 CAN 模块框图

### 19.1.3 CAN 内核

CAN 内核由 CAN 协议控制器和收发移位寄存器（Rx/Tx Shift register）组成，负责处理所有 ISO 11898-1 协议规定的功能。

#### 19.1.3.1 消息处理器（Message Handler）

消息处理器是一个状态机，用于控制单端口消息 RAM 与 CAN 内核收发移位寄存器之间的数据传输，同时根据控制寄存器的配置处理接收滤波和中断请求生成。

#### 19.1.3.2 消息 RAM（Message RAM）

CAN 消息 RAM 可存储 32 条 CAN 消息。

#### 19.1.3.3 寄存器与消息对象访问接口（IFx）

通过间接访问消息对象确保数据一致性。正常工作时，CPU 和 DMA 对消息 RAM 的所有访问均通过接口寄存器完成。IFx 寄存器可视为访问消息对象（邮箱）的“窗口”。

三个接口寄存器组控制 CPU 对消息 RAM 的读写访问（见图中）：两个支持读写访问的接口寄存器组（IF1 和 IF2），以及一个仅支持读访问的接口寄存器组（IF3）。接口寄存器与消息 RAM 的字长一致。

在专用测试模式下，消息 RAM 采用内存映射方式，可直接访问。

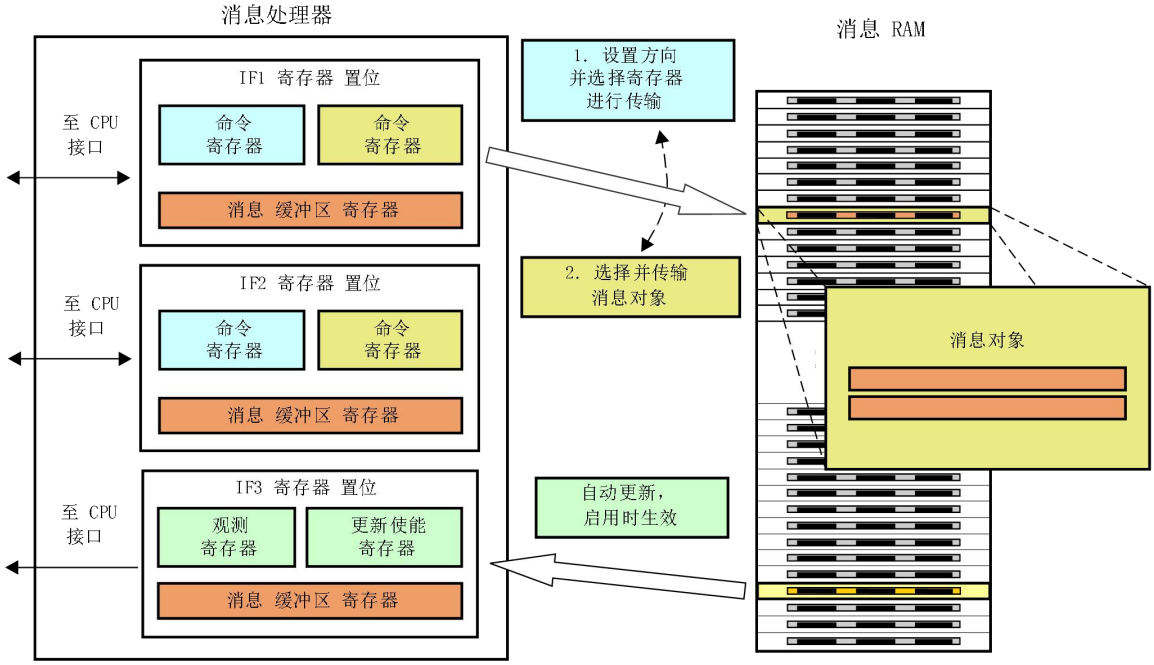



图 19-2 通过 IFx 寄存器访问消息对象

## 19.2 功能描述

CAN 模块按照 ISO 11898-1 协议进行 CAN 通信，比特率可编程设置，最高可达 1 Mbit/s。连接物理层（CAN 总线）需配备 CAN 收发器芯片。

CAN 网络通信时，可配置独立的消息对象，消息对象和标识符掩码存储在消息 RAM 中。

与消息处理相关的所有功能均由消息处理器实现，包括：接收滤波、CAN 内核与消息 RAM 之间的消息传输、发送请求处理以及中断或 DMA 请求生成。

CPU 可通过模块接口直接访问 CAN 的寄存器组，这些寄存器用于控制和配置 CAN 内核与消息处理器，以及访问消息 RAM。

## 19.3 工作模式

### 19.3.1 初始化模式

通过以下方式进入初始化模式：软件设置（置位 CAN_CTL 寄存器的 Init 位）、硬件复位或总线关闭（bus-off）状态。Init 位置位时，CAN 总线的收发数据传输停止，CAN_TX 输出为隐性电平（高电平），CAN 错误计数器不更新。置位 Init 位不会改变其他配置寄存器的值。

初始化 CAN 控制器时，CPU 需配置 CAN 位定时和用于 CAN 通信的消息对象。无需使用的消息对象可通过 MsgVal 位清 0 禁用。

仅当 CAN 控制寄存器的 Init 位和 CCE 位均置位时，才能访问 Bit Timing 寄存器进行配置。

Init 位清 0 即完成软件初始化，之后比特流处理器（BSP）需等待检测到 11 个连续的隐性位（即总线空闲），才能与 CAN 总线上的数据传输同步，参与总线活动并启动消息传输。

消息对象的初始化与 Init 位状态无关，但在启动消息传输前，所有消息对象必须配置特定标识符或设为“无效”状态。

正常工作期间，CPU 可修改消息对象的配置。当接口寄存器的消息对象传输至消息 RAM 后，若修改后的消息对象编号小于或等于之前找到的消息对象编号，接收滤波会对其生效。即使在 CAN 帧接收挂起时修改消息对象，也能确保数据一致性。

### 19.3.2 CAN 消息传输（正常工作模式）

CAN 初始化完成且 Init 位清 0 后，CAN 内核与 CAN 总线同步，准备进行通信。

接收的消息若通过接收滤波，会存储到对应的消息对象中，完整消息（消息标识符 MSGID、数据长度码 DLC 和最多 8 字节数据）均存入该消息对象。例如，若使用标识符掩码，存储接收消息时，消息对象中被掩码设为“无关”（don't care）的 MSGID 位可能会发生变化。

CPU 可通过接口寄存器随时读写每个消息，即使存在并发访问，消息处理器也能确保数据一致性。

CPU 可更新待发送的消息：若存在该消息的永久消息对象（配置期间设置的 MSGID、控制位和多 CAN 传输配置），仅需更新数据字节；若一个消息对象需分配多个发送消息，则每次发送请求前需完整配置该消息对象。

可同时请求多个消息对象的传输，消息对象按内部优先级依次发送。即使发送请求仍在挂起，也可随时更新消息或设为“无效”，但挂起传输启动前更新消息会丢弃原数据字节。

根据消息对象的配置，接收标识符匹配的远程帧可自动触发发送请求。

#### 19.3.2.1 禁用自动重传

根据 CAN 规范（见 ISO11898，6.3.3 恢复管理），CAN 提供自动重传机制，用于重传仲裁失败或传输过程中因错误干扰的帧。只有传输成功后，才会向用户确认帧传输服务完成。

默认情况下，自动重传功能启用，可通过置位 CAN 控制寄存器的 DAR 位禁用。

#### 19.3.2.2 自动总线恢复（Auto-Bus-On）

CAN 进入总线关闭状态后，CPU 可通过清 0Init 位启动总线关闭恢复序列；若未执行该操作，模块将保持总线关闭状态。

CAN 提供自动总线恢复功能，通过 ABO 位启用。置位 ABO 位后，CAN 会自动启动总线关闭恢复序列，序列启动可通过用户定义的时钟周期数延迟。

**注意：**

若 CAN 模块因多次 CAN 总线错误进入总线关闭状态，会停止所有总线活动并自动置位 Init 位。应用程序清 0Init 位（或通过自动总线恢复功能）后，器件需等待 129 次总线空闲（即 129×11 个连续隐性位）才能恢复正常工作，无法通过置位或清 0Init 位缩短总线关闭恢复序列。恢复序列结束时，错误计数器复位。Init 位清 0 后，每次检测到 11 个连续隐性位，都会向错误与状态寄存器写入 Bit0 错误码，CPU 可通过该错误码检查 CAN 总线是否处于显性锁定状态或持续受干扰，同时监控总线关闭恢复序列的进度。

### 19.3.3 测试模式

CAN 模块提供多种测试模式，主要用于自测试。图中有助于理解各类测试模式的工作原理（该图仅展示模块行为特征，非精确的门级实现，未包含 GPIO 多路复用和 I/O 缓冲器）。

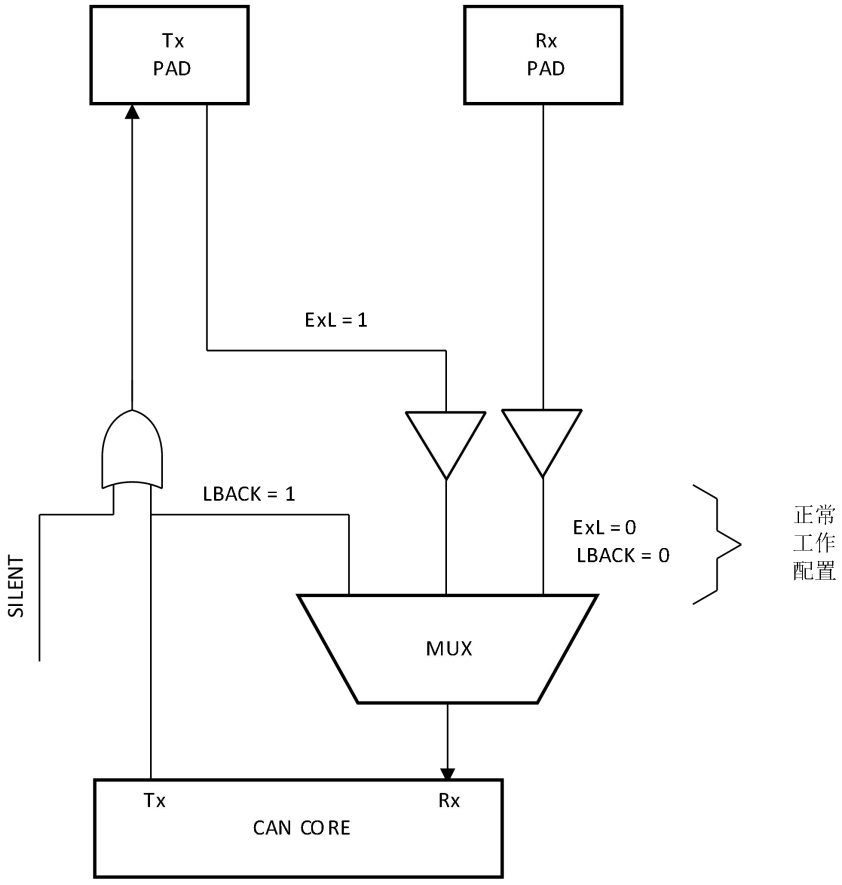



图 19-3 CAN 多路选择器（CAN_MUX）

所有测试模式均需置位 CAN 控制寄存器的 Test 位（设为 1），才能启用对 CAN_TEST 寄存器的写访问。

### 19.3.3.1 静默模式（Silent Mode）

静默模式可用于分析 CAN 总线上的通信流量，且不会通过发送显性位（如应答位、过载标志、主动错误标志）影响 CAN 总线。CAN 仍能接收有效的数据帧和远程帧，但不发送任何显性位，接收的帧会内部路由至 CAN 内核。

图中为静默模式下 CAN_TX 和 CAN_RX 信号与 CAN 内核的连接关系。置位测试寄存器（CAN_TEST）的 Silent 位（设为 1）可激活静默模式。在 ISO 11898-1 中，静默模式也称为总线监控模式。

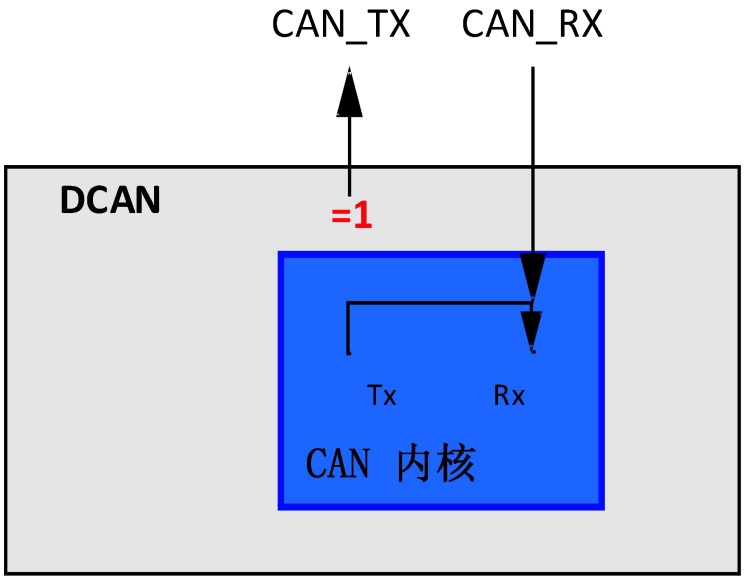



图 19-4 静默模式下的 CAN 内核

### 19.3.3.2 环回模式（Loopback Mode）

环回模式主要用于硬件自测试功能。该模式下，CAN 内核采用 Tx 输出到 Rx 输入的内部反馈，发送的消息被视为接收消息，若通过接收滤波则存储到消息对象中，CAN 内核忽略 CAN_RX 输入引脚的实际值，发送的消息仍可在 CAN_TX 引脚监测。

为避免依赖外部激励，环回模式下 CAN 内核忽略应答错误（数据帧/远程帧应答时隙中采样到隐性位）。

下图为环回模式下 CAN_TX 和 CAN_RX 信号与 CAN 内核的连接关系。置位 CAN_TEST 寄存器的 LBack 位（设为 1）可激活环回模式。

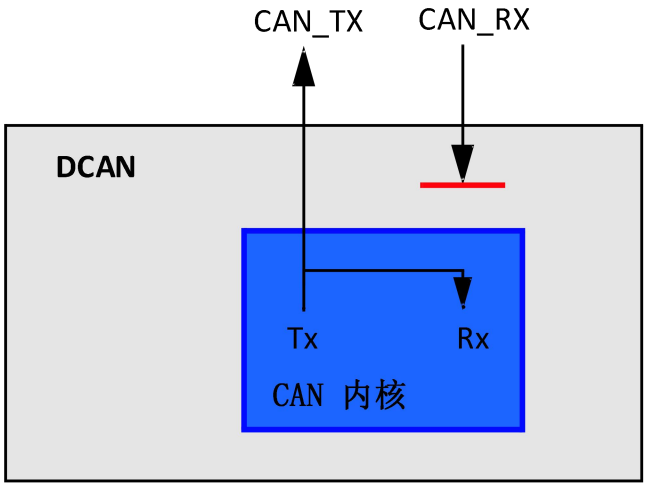


图 19-5 环回模式下的 CAN 内核

环回模式下，CAN 内核到 Tx 引脚的信号路径以及 Tx 引脚回 CAN 内核的信号路径未纳入测试范围。

### 19.3.3.3 外部环回模式（External Loopback Mode）

外部环回模式与环回模式类似，但包含 CAN 内核到 Tx 引脚的信号路径以及 Tx 引脚回 CAN 内核的信号路径。选择外部环回模式后，CAN 内核连接至 Tx 引脚的输入缓冲器，可测试 Tx 引脚的 I/O 电路。置位测试寄存器的 ExL 位（设为 1）可激活外部环回模式。

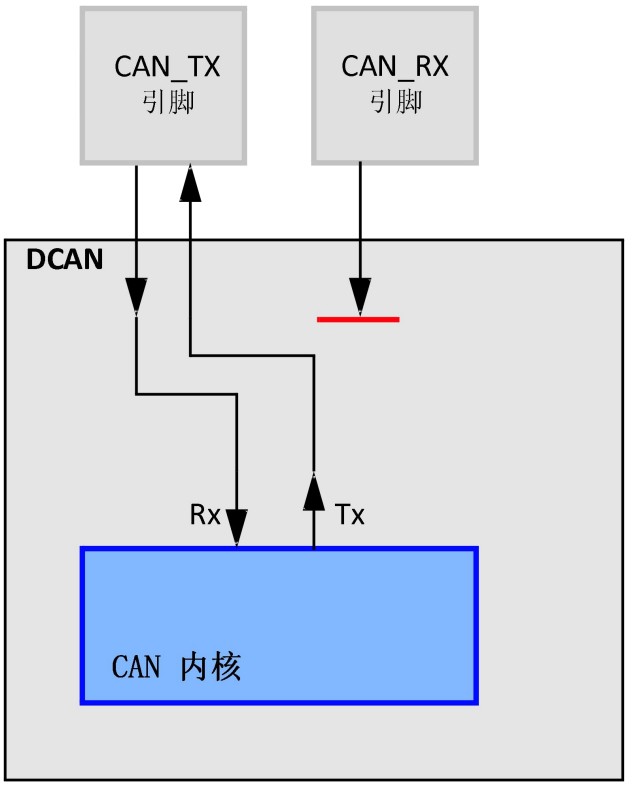



图 19-6 外部环回模式下的 CAN 内核

注意：当环回模式激活（LBack 位置 1）时，ExL 位会被忽略。

#### 19.3.3.4 环回模式与静默模式组合

同时置位 LBack 位和 Silent 位，可组合使用环回模式和静默模式。该模式下可测试 CAN 硬件，且不影响 CAN 网络，CAN_RX 引脚与 CAN 内核断开连接，CAN_TX 引脚不发送任何显性位。

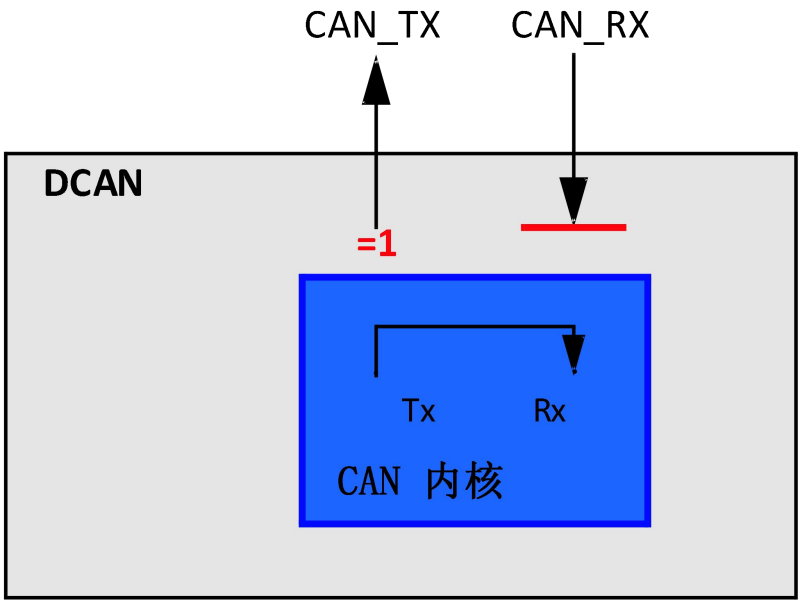



图 19-7 环回+静默模式下的 CAN 内核

## 19.4 多种时钟源

CAN 位定时时钟通常源自系统时钟（SYSCLK），也可直接源自外部时钟源（XTAL）。

注意：CAN 内核的每个比特时间需至少编程为 8 个时钟周期。要实现 1 Mbit/s 的传输速率，振荡器频率需不低于 8 MHz。

## 19.5 中断功能

CAN 模块可通过两条中断线（CANINT0 和 CANINT1）产生中断，分别通过置位 CAN 控制寄存器的 IE0 位和 IE1 位启用。

CAN 提供三类中断源：消息对象中断、状态变化中断和错误中断。可通过中断寄存器中的中断标识符 Int0ID 和 Int1ID 确定中断源，无中断挂起时该寄存器值为 0。

每条中断线保持激活状态，直至中断寄存器中的对应字段（Int0ID 或 Int1ID）恢复为 0（即中断原因复位），或 IE0 位和 IE1 位清 0。

Int0ID 字段值为 0x8000 时，表示 CAN 内核已更新（不一定改变）错误与状态寄存器（Error and Status Register），触发中断（该中断优先级最高）。CPU 可通过读取错误与状态寄存器更新（复位）RxOk、TxOk 和 LEC 状态位，但 CPU 的写访问不会产生或复位中断。

Int0ID 和 Int1ID 值在 1 至最后一个消息对象编号之间时，表示中断源为某个消息对象，指向优先级最高的挂起消息中断。消息对象 1 优先级最高，最后一个消息对象优先级最低。

中断服务程序读取中断源消息时，可同时读取消息并复位消息对象的 IntPnd 位（通过 IF1 或 IF2 命令寄存器的 ClrIntPnd 位）。IntPnd 位清 0 后，中断寄存器指向下一个具有挂起中断的消息对象。

CAN 模块具备模块级中断使能和确认机制。要使能 CANINT0/CANINT1 中断，需置位 CAN_GLB_INT_EN 寄存器中的对应位；处理中断时，应先清除相应消息对象或状态变化标志，再通过 CAN_GLB_INT_CLR 完成模块级中断确认。

### 19.5.1 消息对象中断

消息对象中断由消息对象的事件产生，受 IntPnd、TxIE 和 RxIE 标志控制。消息对象中断可路由至 CANINT0 或 CANINT1 线，通过中断多路复用寄存器控制。

注意：向 CAN_IFnMCTL 寄存器的 IntPnd 位写入数据可强制产生中断。

### 19.5.2 状态变化中断

错误与状态寄存器中的 RxOk、TxOk 和 LEC 事件属于状态变化中断，通过置位 CAN 控制寄存器的 SIE 位启用该类中断。SIE 位置位时，无论总线是否出错、CAN 通信是否有效，也无论消息 RAM 如何配置，每个 CAN 帧都会产生状态变化中断。状态变化中断仅能路由至 CANINT0 线，需置位 CAN_CTL 寄存器的 IE0 位启用。

### 19.5.3 错误中断

PER、BOff 和 EWarn 事件属于错误中断，通过置位 EIE 位启用该类中断。错误中断仅能路由至 CANINT0 线，需置位 CAN_CTL 寄存器的 IE0 位启用。

### 19.5.4 中断拓扑

CAN 的中断拓扑如图所示。收发操作的邮箱中断可路由至 CANINT0 和 CANINT1 线，错误和状态中断仅能路由至 CANINT0 线。

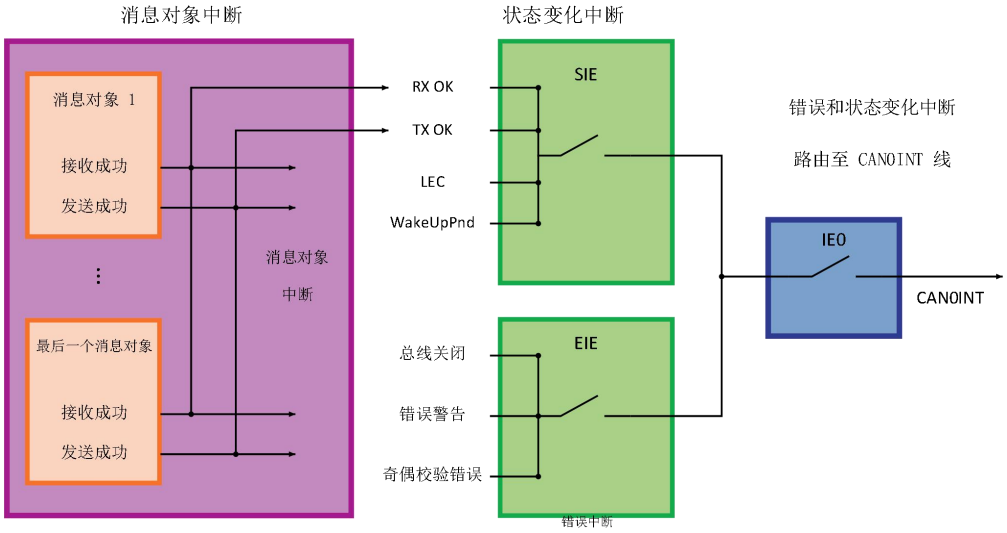



图 19-8 CAN 中断拓扑 1

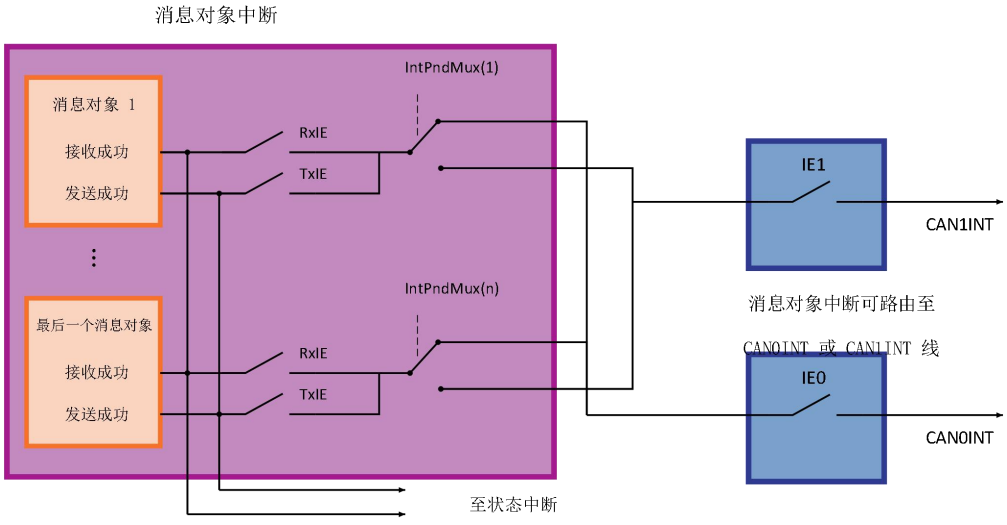



图 19-9 CAN 中断拓扑 2

## 19.6 DMA 功能

CAN 模块提供三个 DMA 触发输出，分别对应三个接口寄存器 IF1、IF2 和 IF3，可通过 CAN_CTL 寄存器的 DE1、DE2 和 DE3 位启用。

IF1 和 IF2 寄存器的更新分别由对 IF1 和 IF2 命令寄存器的写访问触发。启用后，置位 IF1CMD 或 IF2CMD 寄存器的 DMAactive 位，下次对应接口可用时会产生 DMA 请求。

若消息对象编程为自动更新 IF3，则接收 CAN 消息时 IF3 寄存器内容会自动更新；启用 IF3 DMA 请求后，所有 IF3 更新都会触发 DMA 请求。

CAN 内部 IFx 更新完成后，DMA 请求激活，直至首次访问相关 IFx 寄存器组（即对 IF 寄存器组的首次读或写访问会清除 DMA 请求）。

## 19.7 奇偶校验机制

CAN 提供奇偶校验机制，确保消息 RAM 数据的完整性，消息 RAM 中每个字（32 位）计算一个奇偶校验位。

写访问时，奇偶校验信息存储到消息 RAM；读访问时，将计算的奇偶校验位与消息 RAM 中存储的奇偶校验位进行比对。

通过 CAN 控制寄存器的 PMD 位字段可启用或禁用奇偶校验功能：禁用时，写访问数据区不改变消息 RAM 中的奇偶校验位，读访问不进行校验；启用时，CAN 自动生成并校验奇偶校验位，若数据位的模 2 和为 1，则置位奇偶校验位（即奇偶校验位置位时，数据中 1 的个数为奇数）。

### 19.7.1 奇偶校验错误处理

对消息 RAM 的任何读访问（例如 CAN 帧传输启动时）都会校验消息对象的奇偶性。检测到奇偶校验错误时，置位错误与状态寄存器的 PER 位；若启用错误中断，会产生中断。为避免通过 CAN 总线传输无效数据，消息对象的 MsgVal 位清 0。

无论是否存在奇偶校验错误，CPU 均可读取消息对象数据，因此应用程序需确保读取数据的有效性（例如奇偶校验错误中断时立即检查奇偶校验错误码寄存器）。

## 19.8 调试模式

模块支持外部调试单元，提供暂停 CAN 活动、调试器访问消息 RAM 内容等功能。连接外部调试器且内核暂停时，自动进入调试模式。

进入调试模式前，电路会等待传输启动、接收完成或识别到总线空闲状态；若置位 IDS 位，调试器会立即中断当前传输或接收。之后 CAN 进入调试模式，由 CAN 控制寄存器的 InitDbg 标志指示。调试模式下，可访问所有 CAN 寄存器，读取保留位返回 0，写入保留位无效果；消息 RAM 采用内存映射方式，外部调试单元可读。

**注意：**调试模式下，无法通过 IFx 寄存器组访问消息 RAM。

调试模式下写入控制寄存器可能影响 CAN 状态机和后续消息处理。调试时，以下 CAN 寄存器的自动清除功能禁用：

- 错误与状态寄存器（通过读取清除状态标志）
- IF1/IF2 命令寄存器（通过读写清除 DMA Active 标志）

## 19.9 模块初始化

硬件复位后，CAN 控制寄存器的 Init 位置位，所有 CAN 协议功能禁用。启用 CAN 协议功能前，需完成位定时和消息对象的配置。

硬件复位会将消息对象的 MsgVal、NewDat、IntPnd 和 TxRqst 位清 0。消息对象配置需将 IF1/IF2 接口寄存器组的掩码、仲裁、控制和数据位编程为目标值，向对应 IF1/IF2 命令寄存器的[7:0]位写入消息对象编号，即可将 IF1/IF2 接口寄存器内容加载到消息 RAM 中对应的消息对象。

位定时配置需同时置位 CAN 控制寄存器的 Init 位和 CCE 位，消息对象配置无需该操作。

CAN 控制寄存器的 Init 位清 0 后，CAN 内核的 CAN 协议控制器状态机和消息处理器状态机开始控制 CAN 内部数据流：通过接收滤波的接收消息存储到消息 RAM，挂起发送请求的消息加载到 CAN 内核的移位寄存器并通过 CAN 总线发送。

CPU 可在清 0Init 位和 CCE 位的同时启用中断线（置位 IE0 和 IE1 位），也可同时启用状态中断 EIE 和 SIE。

CAN 通信可采用中断驱动或查询模式控制：中断寄存器指向 IntPnd=1 的消息对象，即使禁用 CPU 的中断线（IE0 和 IE1 位清 0），该寄存器仍会更新。

CPU 可通过 NewData 寄存器和传输请求寄存器并行查询所有消息对象的 NewDat 和 TxRqst 位；若所有发送对象集中在低编号，所有接收对象集中在高编号，可简化查询操作。

## 19.10 消息对象配置

初始化结束前需完成整个消息 RAM 的配置，CAN 通信期间也可修改消息对象的配置。

### 19.10.1 数据帧发送对象配置

表中为发送对象的初始化配置。

表 19-1 发送对象初始化配置

| MsgVal | Arb | 数据 | Mask | EoB | Dir | NewDat | MsgLst | RxIE | TxIE | IntPnd | RmtEn | TxRqst |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 应用定义 | 应用定义 | 应用定义 | 1 | 1 | 0 | 0 | 0 | 应用定义 | 0 | 应用定义 | 0 |

- 仲裁位（ID[28:0]和 Xtd 位）由应用程序定义，指定输出消息的标识符和类型。使用 11 位标识符（标准帧）时（Xtd=0），标识符编程到 ID[28:18]，ID[17:0]可忽略。

- 数据寄存器（DLC[3:0]和 Data0-7）由应用程序定义，数据有效前不得置位 TxRqst 和 RmtEn。
- 置位 TxIE 位时，消息对象发送成功后 IntPnd 位置位；
- 置位 RmtEn 位时，接收匹配的远程帧会置位 TxRqst 位，远程帧会自动由数据帧应答。
- 置位 UMask=1 时，可使用掩码位（Msk[28:0]、UMask、MXtd 和 MDir 位）允许同类标识符的远程帧组置位 TxRqst 位，Dir 位不得掩码；若不允许远程帧置位 TxRqst 位（RmtEn=0），需禁用标识符掩码（UMask=0）。

### 19.10.2 远程帧发送对象配置

无需专门配置远程帧发送对象，置位接收对象的 TxRqst 位即可发送远程帧，该远程帧的标识符与接收对象配置的数据帧标识符一致。

### 19.10.3 数据帧单个接收对象配置

表中为数据帧接收对象的初始化配置。

表 19-2 数据帧单个接收对象初始化配置

| MsgVal | Arb | 数据 | Mask | EoB | Dir | NewDat | MsgLst | RxIE | TxIE | IntPnd | RmtEn | TxRqst |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 应用定义 | 应用定义 | 应用定义 | 1 | 0 | 0 | 0 | 应用定义 | 0 | 0 | 0 | 0 |

- 仲裁位（ID[28:0]和 Xtd 位）由应用程序定义，指定可接收消息的标识符和类型。使用 11 位标识符（标准帧）时（Xtd=0），标识符编程到 ID[28:18]，ID[17:0]可忽略；接收 11 位标识符的数据帧时，ID[17:0]置 0。
- 消息处理器将数据帧存储到消息对象时，会存储接收的数据长度码和对应的数据字节；若数据长度码小于 8，消息对象的剩余字节可能被非指定值覆盖。
- 置位 UMask=1 时，可使用掩码位（Msk[28:0]、UMask、MXtd 和 MDir 位）允许同类标识符的数据帧组被接收；典型应用中 Dir 位不得掩码；若掩码位部分设为“无关”（don't care），仲裁寄存器的对应位会被存储数据帧的位覆盖。

- 置位 RxIE 位时，接收的数据帧被接受并存储到消息对象后，IntPnd 位置位；
- 置位 TxRqst 位时，触发发送与仲裁位存储的标识符一致的远程帧；若接收滤波使用掩码位（UMask=1），仲裁位可能发生变化。

### 19.10.4 远程帧单个接收对象配置

表 19-3 远程帧单个接收对象初始化配置

| MsgVal | Arb | 数据 | Mask | EoB | Dir | NewDat | MsgLst | RxIE | TxIE | IntPnd | RmtEn | TxRqst |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 应用定义 | 应用定义 | 应用定义 | 1 | 1 | 0 | 0 | 应用定义 | 0 | 0 | 0 | 0 |

- 远程帧接收对象可用于监控 CAN 总线上的远程帧，存储在接收对象中的远程帧不会触发数据帧发送，远程帧接收对象可扩展为 FIFO 缓冲器。
- 需置位 UMask=1，掩码位（Msk[28:0]、UMask、MXtd 和 MDir 位）可设为“必须匹配”或“无关”，允许同类标识符的远程帧组被接收；典型应用中 Dir 位不得掩码。
- 仲裁位（ID[28:0]和 Xtd 位）由应用程序定义，指定可接收远程帧的标识符和类型；若掩码位部分设为“无关”，仲裁位的对应位会被存储远程帧的位覆盖；使用 11 位标识符（标准帧）时（Xtd=0），标识符编程到 ID[28:18]，ID[17:0]可忽略；接收 11 位标识符的远程帧时，ID[17:0]置 0。
- 数据长度码（DLC[3:0]）由应用程序定义，消息处理器将远程帧存储到消息对象时，会存储接收的数据长度码，消息对象的数据字节保持不变。
- 置位 RxIE 位时，接收的远程帧被接受并存储到消息对象后，IntPnd 位置位。

### 19.10.5 FIFO 缓冲器配置

除 EoB 位外，FIFO 缓冲器中的接收对象配置与单个接收对象一致。

将多个消息对象组合为 FIFO 时，需将这些消息对象的标识符和掩码（若使用）编程为匹配值。由于消息对象的隐含优先级，编号最小的消息对象为 FIFO 缓冲器的第一个消息对象；FIFO 缓冲器中除最后一个消息对象外，其余所有消息对象的 EoB 位需编程为 0，最后一个消息对象的 EoB 位置位，配置为块结束。

## 19.11 消息处理

初始化完成后，CAN 模块与 CAN 总线上的通信流量同步，对接收的消息进行接收滤波，将接受的帧存储到指定消息对象。应用程序需更新待发送消息的数据，启用并请求发送；接收匹配的远程帧可自动触发发送请求。

应用程序可读取接收并接受的消息，同一消息对象在接收下一条消息前未读取的消息会被覆盖；消息读取可采用中断驱动方式，或查询 NewDat 位后读取。

### 19.11.1 消息处理器概述

消息处理器状态机控制CAN内核收发移位寄存器与消息RAM之间的数据传输，执行以下任务：

- 消息 RAM 到 CAN 内核的数据传输（待发送消息）
- CAN 内核到消息 RAM 的数据传输（接收消息）
- CAN 内核到接收滤波单元的数据传输
- 扫描消息 RAM 查找匹配的消息对象（接收滤波）
- 当 IF1/IF2 寄存器修改的消息对象优先级等于或高于上次扫描找到的消息对象时，重新扫描该消息对象
- TxRqst 标志处理
- 中断标志处理

消息处理器寄存器包含所有消息对象的状态标志，按以下类别分组：

- 传输请求标志
- 新数据标志
- 中断挂起标志
- 消息有效寄存器

无需通过 IFx 寄存器分别收集每个消息对象的上述状态信息，这些消息处理器寄存器提供了快速便捷的概览方式（例如所有挂起的传输请求），且均为只读。

### 19.11.2 接收/发送优先级

消息对象的接收/发送优先级与消息编号关联，而非 CAN 标识符：消息对象 1 优先级最高，消息对象 32 优先级最低。若多个传输请求挂起，按对应消息对象的优先级处理，因此可将高优先级消息放在低编号消息对象中。

数据帧或远程帧的接收滤波也按消息对象升序进行，因此被低编号消息对象接受的帧不会被高编号消息对象接受。对于需要记录 CAN 总线所有消息流量的节点，可将最后一个消息对象配置为接受其他消息对象未接受的任何数据帧或远程帧。

### 19.11.3 事件驱动 CAN 通信中的消息传输

若 CAN 内核的移位寄存器准备就绪，且 IFx 寄存器与消息 RAM 之间无数据传输，会评估消息有效寄存器中的 MsgVal 位和传输请求寄存器中的 TxRqst 位。消息处理器将优先级最高的有效挂起传输请求消息对象加载到移位寄存器并启动传输，同时清 0 该消息对象的 NewDat 位。

发送成功且传输启动后未向消息对象写入新数据（NewDat=0）时，清 0TxRqst 位；置位 TxIE 位时，发送成功后 IntPnd 位置位。若仲裁失败或传输过程中发生错误，CAN 总线空闲后会重新传输该消息；若期间有更高优先级消息的传输请求，按优先级顺序传输消息。

通过置位 CAN 控制寄存器的 DAR 位禁用自动重传模式时，接口寄存器组的消息控制寄存器中 TxRqst 和 NewDat 位的行为如下：

- 传输启动时，对应接口寄存器组的 TxRqst 位清 0，NewDat 位保持置位；
- 传输成功完成时，NewDat 位清 0；

传输失败（仲裁失败或错误）时，NewDat 位保持置位，需应用程序重新置位 TxRqst 位重启传输。

接收远程帧无需接收对象存储，若匹配的发送对象中 RmtEn 位置位，远程帧会自动触发数据帧发送。

### 19.11.4 更新发送对象

CPU 可通过 IF1 和 IF2 接口寄存器随时更新发送对象的数据字节，更新前无需清 0 MsgVal 或 TxRqst 位。

即使仅需更新部分数据字节，对应 IF1/IF2 数据 A 寄存器或 IF1/IF2 数据 B 寄存器中的四个字节在传输至消息对象前必须有效：CPU 需向 IF1/IF2 数据寄存器写入所有四个字节，或先将消息对象传输至 IF1/IF2 数据寄存器，再写入新数据字节。

仅更新数据字节时，可先向命令寄存器的[23:16]位写入 0x87，再向命令寄存器的[7:0]位写入消息对象编号，同时更新数据字节并置位 TxRqst 和 NewDat 位。

事件驱动 CAN 通信中，为避免数据更新时已在进行的传输结束后清 0TxRqst 位，需同时置位 TxRqst 和 NewDat 位；同时置位后，新传输启动时 NewDat 位清 0。

### 19.11.5 修改发送对象

若实现的消息对象数量不足以仅作为永久消息对象使用，可动态管理发送对象。CPU 可将完整消息（仲裁、控制和数据）写入接口寄存器，向命令寄存器的[23:16]位写入 0xB7，即可将整个消息对象内容传输至消息对象，该操作前无需清 0MsgVal 或 TxRqst 位。

若该消息对象之前的传输请求未完成但已在进行中，会继续传输，但传输受干扰时不会重传；仅需更新正在传输消息的数据字节时，向命令寄存器的[23:16]位写入 0x87。

注意：更新发送对象后，接口寄存器组包含该对象的实际内容副本，包括未更新的部分。

### 19.11.6 接收消息的接收滤波

当输入消息的仲裁和控制位（标识符+IDE+RTR+DLC）完全移入 CAN 内核的移位寄存器后，消息处理器开始扫描消息 RAM 查找匹配的有效消息对象：

- 接收滤波单元加载 CAN 内核移位寄存器中的仲裁位；

- 将消息对象 1 的仲裁和掩码位（包括 MsgVal、UMask、NewDat 和 EoB）加载到接收滤波单元，与移位寄存器中的仲裁位比对；对后续所有消息对象重复上述操作，直至找到匹配的消息对象或到达消息 RAM 末尾；
- 找到匹配对象后停止扫描，消息处理器根据接收帧的类型（数据帧或远程帧）执行后续操作。

### 19.11.7 数据帧接收

消息处理器将 CAN 内核移位寄存器中的消息存储到消息 RAM 中对应的消息对象，不仅存储数据字节，还存储所有仲裁位和数据长度码，确保即使使用仲裁掩码寄存器，数据字节仍与标识符关联。

置位 NewDat 位，指示接收到新数据（CPU 尚未读取）；CPU 读取消息对象时需清 0NewDat 位；接收时若 NewDat 位已置位，置位 MsgLst 位，指示之前的数据（CPU 未读取）丢失；置位 RxIE 位时，IntPnd 位置位，使中断寄存器指向该消息对象。

清 0 该消息对象的 TxRqst 位，防止刚接收数据帧时发送远程帧。

### 19.11.8 远程帧接收

接收远程帧时，考虑匹配消息对象的三种不同配置：

1. Dir=1（方向=发送）、RmtEn=1、UMask=1 或 0：接收匹配的远程帧时，置位该消息对象的 TxRqst 位，消息对象其余部分保持不变；

2. Dir=1（方向=发送）、RmtEn=0、UMask=0：忽略远程帧，消息对象保持不变；

3. Dir=1（方向=发送）、RmtEn=0、UMask=1：远程帧处理方式与数据帧类似，接收匹配的远程帧时，清 0 该消息对象的 TxRqst 位，将移位寄存器中的仲裁和控制位（标识符+IDE+RTR+DLC）存储到消息 RAM 的消息对象中，置位该消息对象的 NewDat 位，消息对象的数据字节保持不变。

### 19.11.9 读取接收消息

CPU 可通过 IFx 接口寄存器随时读取接收消息，消息处理器状态机确保数据一致性。

通常，CPU 向命令寄存器的[23:16]位写入 0x7F，再向[7:0]位写入消息对象编号，该组合操作将完整接收消息从消息 RAM 传输至接口寄存器组，同时清 0 消息 RAM 中的 NewDat 和 IntPnd 位（接口寄存器组中的这些位不变），消息控制寄存器中的这些位始终反映清 0 前的状态。

若消息对象使用掩码进行接收滤波，仲裁位显示接收的是哪个匹配消息；NewDat 位的实际值指示自上次读取该消息对象后是否接收新消息；MsgLst 位的实际值指示自上次读取该消息对象后是否接收多个消息，MsgLst 位不会自动清 0。

### 19.11.10 为接收对象请求新数据

CPU 可通过远程帧请求其他 CAN 节点为接收对象提供新数据，置位接收对象的 TxRqst 位会发送带有该接收对象标识符的远程帧，触发其他 CAN 节点发送匹配的数据帧；若匹配的数据帧在远程帧发送前接收，自动清 0 TxRqst 位。

无需修改消息对象内容即可置位 TxRqst 位，需向命令寄存器的[23:16]位写入 0x84。

### 19.11.11 FIFO 缓冲器中存储接收消息

多个消息对象可组合为一个或多个 FIFO 缓冲器，每个 FIFO 缓冲器配置为存储特定（一组）标识符的接收消息，FIFO 缓冲器中消息对象的仲裁和掩码寄存器相同。除最后一个消息对象外，FIFO 缓冲器中所有消息对象的 EoB（缓冲器结束）位为 0，最后一个消息对象的 EoB 位为 1。

标识符与 FIFO 缓冲器匹配的接收消息存储到该 FIFO 缓冲器的消息对象中，从编号最小的消息对象开始。

消息存储到 FIFO 缓冲器的消息对象时，置位该消息对象的 NewDat 位；EoB=0 时置位 NewDat 位，消息处理器会锁定该消息对象，禁止进一步写访问，直至 CPU 清 0NewDat 位。

消息持续存储到 FIFO 缓冲器，直至到达该 FIFO 缓冲器的最后一个消息对象；若之前的消息对象均未通过清 0NewDat 位释放，该 FIFO 缓冲器的后续所有消息均存储到最后一个消息对象（EoB=1），覆盖该消息对象中之前的消息。

### 19.11.12 从 FIFO 缓冲器读取消息

应用程序清空 FIFO 缓冲器前，可在组合为 FIFO 缓冲器的消息对象组中累积多条消息（避免数据丢失）。长度为 N 的 FIFO 缓冲器可存储自上次清空后接收的 N-1 条消息加最后一条消息。清空 FIFO 缓冲器需从编号最小的 FIFO 对象开始，读取并清 0 所有消息对象的 NewDat 位，可通过图中所示示例的子程序实现。

注意：需读取并清空 FIFO 缓冲器的所有消息对象，才能存储下一批消息；否则无法维持真正的 FIFO 功能，因为部分读取的缓冲器中的消息对象会按正常（降序）优先级重新填充。

从 FIFO 缓冲器消息对象读取并清 0NewDat 位的操作，与从单个消息对象读取的操作一致。

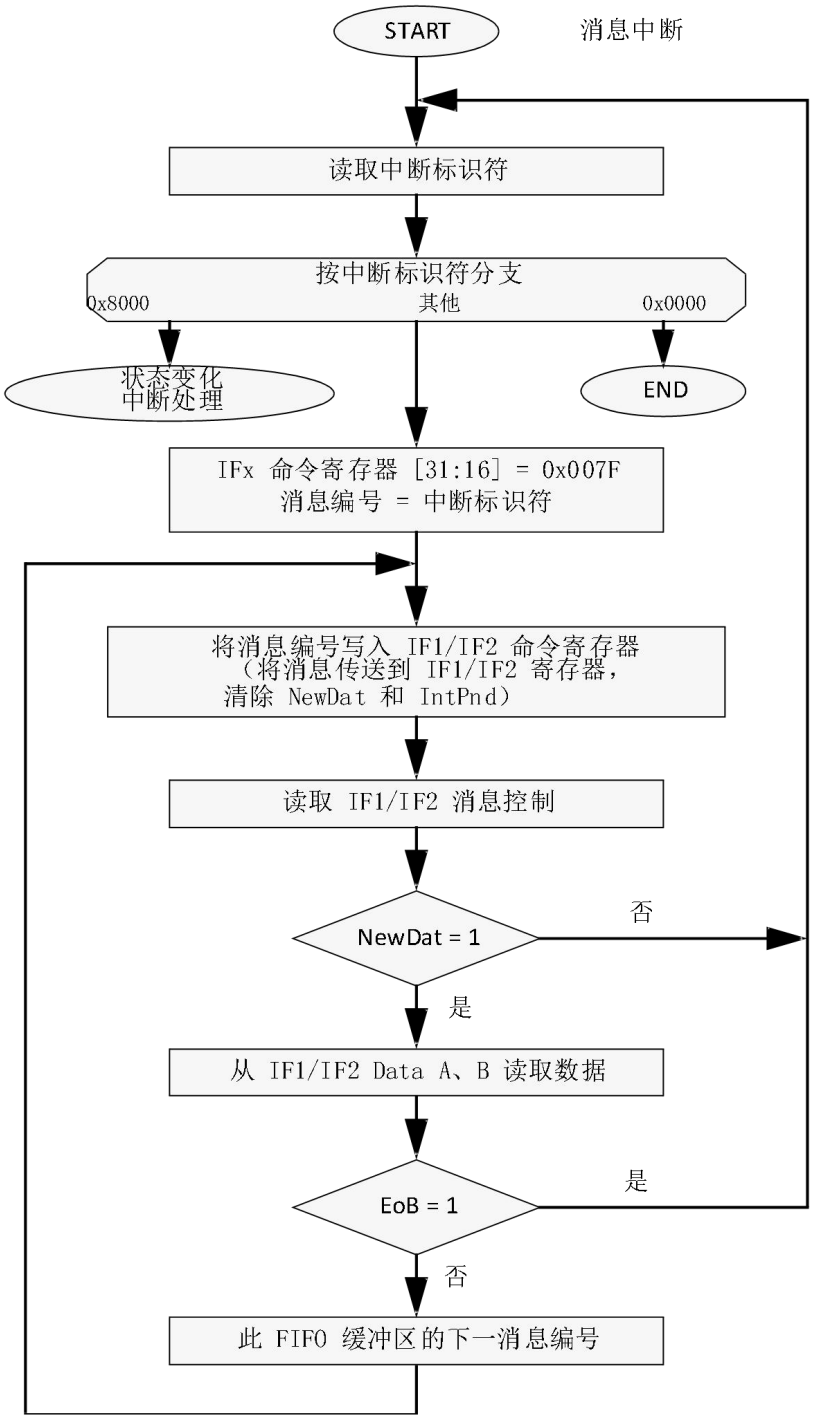



图 19-10 FIFO 缓冲器的 CPU 处理（中断驱动）

## 19.12 位定时与同步

CAN 模块支持高达 1000 kbit/s 的比特率。每个 CAN 节点都有一个时钟发生器，通常衍生自晶体振荡器。位定时参数可以为每个 CAN 节点单独配置，即使各节点的振荡器频率 (Fosc) 可能不同，也能建立统一的比特率。

这些振荡器的频率并非绝对稳定。温度或电压的变化以及组件老化都会引起微小波动。只要波动保持在特定的振荡器容差范围 (df) 内，CAN 节点就能通过对比特流进行重同步 (Resynchronizing) 来补偿不同的比特率。在许多情况下，CAN 位同步可以修正位定时配置中的错误，使其达到仅偶尔产生错误帧的程度。在仲裁情况下，当两个或多个 CAN 节点同时尝试传输一帧时，错误的采样点 (Sample Point) 位置可能导致其中一个发送器进入错误被动 (Error Passive) 状态。

分析此类偶发错误需要详尽了解 CAN 节点内部的位同步机制以及节点在总线上的交互。即使位定时配置中的微小错误不会导致立即失效，也会显著降低 CAN 网络的性能。

### 19.12.1 比特时间组成

每个 CAN 比特时间分为四个时间段，如图所示：

- 同步段（SYNC_SEG）：长度固定为 1 个时间量子（t_q），用于同步 CAN 总线上的所有节点，通过检测 CAN 总线的隐性电平到显性电平跳变（边沿）实现同步。
- 传播段（PROP_SEG）：用于补偿 CAN 总线的物理延迟（信号传播时间），可配置为 1~8 个 t_q。
- 相位缓冲段 1（PHASE_SEG1）：用于补偿节点间的相位偏移，可配置为 1~8 个 t_q，支持自动相位补偿（硬同步和重同步）。
- 相位缓冲段 2（PHASE_SEG2）：功能与 PHASE_SEG1 类似，可配置为 1~8 个 t_q，其长度可通过重同步缩短，但最短不小于 1 个 t_q。

时间量子 t_q 的长度（位时间的基本时间单位）由 CAN_CLK 时钟 与 波特率预分频器（Baud Rate Prescalers：BRPE 与 BRP）决定。

通过这两个预分频器的组合，可以编程设定从 1 到 1024 的分频值 ：

$$t_q = \frac{\text{波特率预分频值}}{\text{CAN_CLK}} \qquad (19\text{-}1)$$

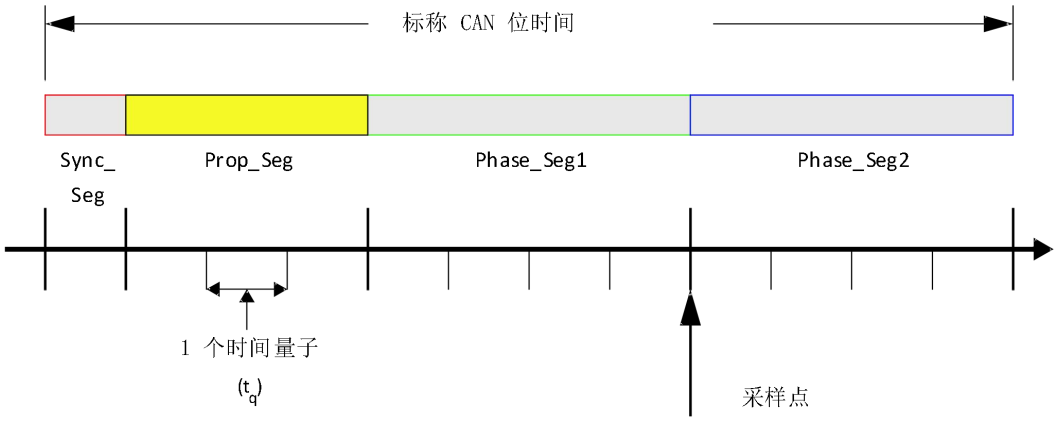



图 19-11 CAN 比特时间结构

除同步段的长度固定外，其余时间段均为可编程参数。

下表给出了 CAN 协议要求的可编程范围。

表 19-4 比特时间组成数据表

| 参数 | 范围 | 说明 |
|---|---|---|
| Sync_Seg | 1 t_q（固定） | 将总线输入同步到 CAN_CLK |
| Prop_Seg | 1 … 8 t_q | 补偿物理传播延迟 |
| Phase_Seg1 | 1 … 8 t_q | 可在同步过程中暂时延长 |
| Phase_Seg2 | 1 … 8 t_q | 可在同步过程中暂时缩短 |
| Synchronization Jump Width (SJW) | 1 … 4 t_q | 不得大于任一相位缓冲段 |

### 19.12.1.1 同步段（Synchronization Segment）

同步段（Sync_Seg）是位时间中预期 CAN 总线电平边沿出现的位置。

如果总线电平边沿出现在 Sync_Seg 之外，则该边沿到 Sync_Seg 的距离被称为相位误差（phase error）。

### 19.12.1.2 传播时间段（Propagation Time Segment）

传播时间段用于补偿 CAN 网络中的物理传播延迟。

这些延迟包括：

- 总线上信号传播时间
- CAN 节点内部的电路延迟

当 CAN 节点根据总线上的位流进行同步时，由于节点之间的信号传播时间，接收节点可能与发送节点产生相位偏移。

CAN 协议中的两个重要机制要求发送节点能够检测到其他节点发送的 dominant 位：

- 无破坏位级仲裁（nondestructive bitwise arbitration）
- 接收节点发送的 ACK 位（dominant acknowledge bit）

因此，当一个 CAN 节点发送位流时，它也必须能够接收到来自其他节点的 dominant 位。

#### 19.12.1.3 相位缓冲段与同步 (Phase Buffer Segments and Synchronization)

相位缓冲段（Phase_Seg1 和 Phase_Seg2）以及同步跳转宽度 (SJW) 用于补偿振荡器容差。相位段围绕采样点分布，并可通过同步操作延长或缩短。SJW 定义了重同步机制在相位段定义的限制范围内移动采样点的最大距离，以补偿边沿相位错误。

同步发生在隐性到显性的边沿上，其目的是控制边沿与采样点之间的距离。通过在每个时间份额内采样实际总线电平，并将其与前一个采样点的电平进行比较来检测边沿。只有当前一采样点为隐性且当前份额电平为显性时，才能进行同步。

同步分为两类：硬同步 (Hard Synchronization) 和重同步 (Resynchronization)。

硬同步： 在帧开始时执行一次，无论相位错误如何，位时间都从 Sync_Seg 结束处重新开始。

位重同步： 导致位时间的缩短或延长，使采样点相对于边沿发生偏移。

当相位错误为正（边沿较晚）时，延长 Phase_Seg1。增加量为相位错误量与 SJW 中的较小者。

当相位错误为负（边沿较早）时，缩短 Phase_Seg2。减小量为相位错误绝对值与 SJW 中的较小者。

#### 19.12.1.4 振荡器容差范围 (Oscillator Tolerance Range)

随着 CAN 协议 1.2 版本的引入，在显性到隐性边沿上进行同步的选项已过时。现在仅考虑从隐性到显性的边沿进行同步。协议更新至 2.0 (A 和 B) 版本对比特率容差没有影响。

振荡器实际频率必须满足：

$(1 - df) \times f_nom \leq f_osc \leq (1 + df) \times f_nom$

式中：f_osc 为振荡器实际频率，f_nom 为标称频率，df 为相对频率容差。

该范围取决于 Phase_Seg1、Phase_Seg2、SJW 以及位时间的比例。最大容差由以下两个条件定义（必须同时满足）：

$df \leq SJW / (20 \times N_bit)$

$df \leq \min(Phase_Seg1, Phase_Seg2) / [2 \times (13 \times N_bit - Phase_Seg2)]$

必须考虑到 SJW 不能大于相位缓冲段中的较小值，且传播时间段限制了位时间中可用于相位缓冲段的部分。

配置组合 Prop_Seg=1 且 Phase_Seg1=Phase_Seg2=SJW=4 时，可获得最大可能的振荡器容差，即 1.58%。这种传播时间段仅占位时间 10% 的组合不适用于短位时间；该组合可用于比特率高达 125 kbit/s（位时间为 8µs）、总线长度为 40 米的场景。

### 19.12.2 CAN 位定时的配置 (Configuration of the CAN Bit Timing)

在 CAN 模块中，位定时配置通过两个寄存器字节进行编程，此外还提供第三个字节用于 4 位波特率预分频器扩展 (BRPE)。Prop_Seg 与 Phase_Seg1 的总和（合称为 TSEG1）与 Phase_Seg2（作为 TSEG2）组合在一个字节中；SJW 与 BRP（加上第三个字节中的 BRPE）组合在另一个字节中（见图中）。

在该位定时寄存器中，组件 TSEG1、TSEG2、SJW 和 BRP 编程的数值比其功能值小 1 ；因此，编程范围不是 [1...n]，而是 [0...n-1]。例如，功能范围为 [1...4] 的 SJW 仅需 2 位即可表示。

位时间长度可按编程值或功能值表示，其中 t_q 为时间量子：

编程值：

t_bit = (TSEG1 + TSEG2 + 3) × t_q

功能值：

t_bit = (Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2) × t_q

位定时寄存器是 CAN 协议控制器的配置输入。波特率预分频器定义时间量子长度：t_q = (BRP + 1) / f_CAN_CLK；TSEG1、TSEG2 和 SJW 决定每个位时间包含的时间量子数。位定时状态机在每个时间量子到达时执行处理并确定采样点。

CAN 协议控制器的其余部分，即位流处理器 (BSP) 状态机，在每个位时间的采样点处评估一次。移位寄存器将待发送的消息序列化，并将接收到的消息并行化。加载和移位操作由 BSP 控制。

BSP 负责消息帧的封装与解析，包括固定格式位、位填充、CRC 校验和错误管理。采样点之后用于计算下一发送位的时间称为信息处理时间（Information Processing Time，IPT）；本 CAN 内核的 IPT 为 0 t_q。

IPT 取决于具体控制器实现，但通常不超过 2 t_q。Phase_Seg2 的配置长度不得小于 IPT；重同步时 Phase_Seg2 可被临时缩短，这不会改变已配置的总线位定时参数。

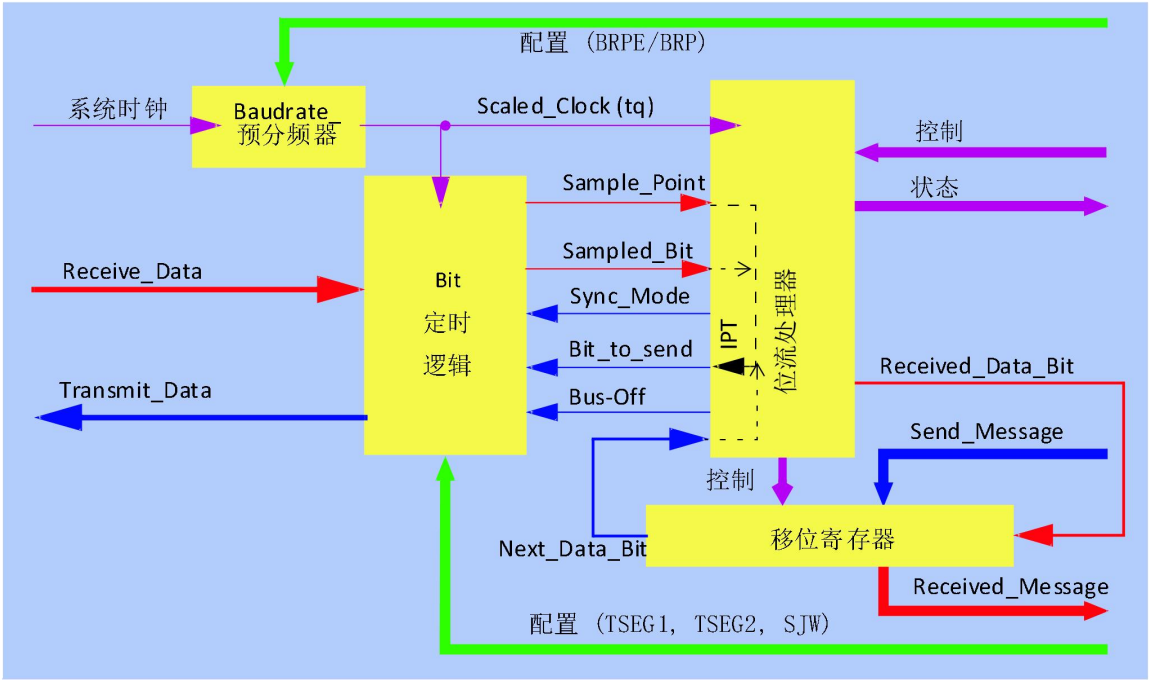



图 19-12 CAN 核心的 CAN 协议控制器的结构

### 19.12.2.1 位定时参数的计算 (Calculation of the Bit Timing Parameters)

位定时配置的计算通常从预期的比特率或位时间开始。最终的位时间（比特率的倒数）必须是 CAN 时钟周期的整数倍。

注意： 8MHz 是以 1MBit/s 比特率运行 CAN 所需的最低 CAN 时钟频率。

t_bit = 1 / BitRate

N_bit = t_bit / t_q

一个位时间通常由 8～25 个时间量子组成。时间量子长度由波特率预分频器定义：t_q = (BRP + 1) / f_CAN_CLK。可按下列步骤迭代选择参数，使计算得到的位时间满足目标比特率。

定义 Prop_Seg：根据最大总线长度、收发器延迟和节点内部延迟估算传播时间，再向上取整为 t_q 的整数倍。

Sync_Seg = 1 t_q

N_remaining = N_bit - Sync_Seg - Prop_Seg

Phase_Seg1 = floor(N_remaining / 2)

Phase_Seg2 = ceil(N_remaining / 2)

分配相位缓冲段：Sync_Seg 固定为 1 t_q，剩余时间量子分配给 Phase_Seg1 和 Phase_Seg2。若剩余数量为偶数，两者相等；若为奇数，则 Phase_Seg2 比 Phase_Seg1 多 1 t_q。

检查 Phase_Seg2 下限：Phase_Seg2 不得短于网络中任一节点的 IPT；IPT 取决于具体器件，可在 0~2 t_q 之间（本 CAN 内核为 0 t_q）。

设置 SJW：同步跳转宽度设为最大值，即 4 和 Phase_Seg1 中的较小者。

计算振荡器容差：使用 19.12.1.4 节中的公式核算。如果存在多种配置方案，应优先选择容差裕量较大的方案。

最终的配置值写入位定时寄存器：

(Phase_Seg2-1) & (Phase_Seg1+Prop_Seg-1) & (SynchronizationJumpWidth-1) & (Prescaler-1)

### 19.12.2.2 高波特率位定时示例 (Example for Bit Timing at High Baudrate)

在此示例中，CAN_CLK 频率为 10MHz，BRP 为 0，比特率为 1MBit/s ：

表 19-5 高波特率位定时示例 (Example for Bit Timing at High Baudrate)数据表

| 参数 | 值 | 关系 / 公式 |
|---|---|---|
| t_q | 100 ns | t_CAN_CLK |
| delay of bus driver | 90 ns | |
| delay of receiver circuit | 40 ns | |
| delay of bus line (40 m) | 220 ns | |
| t_prop | 700 ns | 2 × delays = 7 · t_q |
| t_SJW | 100 ns | 1 · t_q |
| t_TSeg1 | 800 ns | t_prop + t_SJW |
| t_TSeg2 | 100 ns | Information Processing Time + 1 · t_q |
| t_Sync-Seg | 100 ns | 1 · t_q |
| bit time | 1000 ns | t_Sync-Seg + t_TSeg1 + t_TSeg2 |

| 参数 | 值 | 关系 / 公式 |
| --- | --- | --- |
| tolerance for CAN_CLK | 0.388 % | min(TSeg1, TSeg2) / (2 × ((13 × bit time) − TSeg2)) |

功能值：TSEG1 = 8，TSEG2 = 1，SJW = 1，Prescaler = 1。

对应的编程字段值为 TSEG1-1 = 7、TSEG2-1 = 0、SJW-1 = 0、Prescaler-1 = 0，位定时寄存器编程值为 0x0000 0700。

#### 19.12.2.3 低波特率位定时示例 (Example for Bit Timing at Low Baudrate)

在此示例中，CAN_CLK 频率为 2MHz，BRP 为 1，比特率为 100 kbit/s ：

表 19-6 低波特率位定时示例 (Example for Bit Timing at Low Baudrate)数据表

| 参数 | 值 | 关系 / 公式 |
| --- | --- | --- |
| t_q | 1 µs | 2 · t_CAN_CLK |
| delay of bus driver | 200 ns | |
| delay of receiver circuit | 80 ns | |
| delay of bus line (40 m) | 220 ns | |
| t_prop | 1 µs | 1 · t_q |
| t_SJW | 4 µs | 4 · t_q |
| t_TSeg1 | 5 µs | t_prop + t_SJW |
| t_TSeg2 | 4 µs | Information Processing Time + 4 · t_q |
| t_Sync-Seg | 1 µs | 1 · t_q |
| bit time | 10 µs | t_Sync-Seg + t_TSeg1 + t_TSeg2 |
| tolerance for CAN_CLK | 1.58 % | min(TSeg1, TSeg2) / (2 × ((13 × bit time) − TSeg2)) |

功能值：TSEG1 = 5，TSEG2 = 4，SJW = 4，Prescaler = 2。

对应的编程字段值为 TSEG1-1 = 4、TSEG2-1 = 3、SJW-1 = 3、Prescaler-1 = 1，位定时寄存器编程值为 0x0000 34C1。

### 19.12.3 同步机制

CAN 总线支持两种同步方式，确保所有节点的比特时间对齐：

硬同步：在每个帧起始的第一个隐性→显性边沿执行；无论相位误差如何，位时间都从同步段（SYNC_SEG）结束处重新开始。

重同步：在帧内（同步段之外）检测到总线边沿时执行，可延长 Phase_Seg1 或缩短 Phase_Seg2，调整量不超过同步跳转宽度（SJW，功能值）。

## 19.13 接口寄存器组（IFx）

接口寄存器组用于控制 CPU 对消息 RAM 的读写访问。模块提供两组用于读写访问的接口寄存器组（IF1 和 IF2），以及一组仅用于读取访问的接口寄存器组（IF3）。

由于消息 RAM 的结构限制，无法直接更改消息对象中的单个位或字节。对消息 RAM 的访问始终是以完整的消息对象为单位进行的。因此，从 IF1/IF2 寄存器向消息 RAM 传输数据时，需要消息处理器执行一个“读-修改-写”周期。首先，消息对象中不需要更改的部分会从消息 RAM 读取到接口寄存器组中，在完成更新后，接口寄存器组的全部内容将被写回该消息对象。

在对消息对象进行部分写入后，接口寄存器组中那些未在命令寄存器（Command Register）中选中的部分，将被设置为该目标消息对象的实际内容。在对消息对象进行部分读取后，接口寄存器组中未被选中的部分将保持不变。

通过对传输数据进行缓冲，接口寄存器组避免了 CPU 访问消息 RAM 与 CAN 消息收发之间的冲突。一个完整的消息对象或其部分内容可以在单次传输中于消息 RAM 和 IF1/IF2 寄存器组之间完成交换。这种在所有选中部分上并行执行的传输操作，确保了 CAN 消息的数据一致性。

存在一种可能导致消息 RAM 写访问丢失的情况。如果所访问的消息对象 MsgVal = 1 且 CAN 通信正在进行，则从 IFx 寄存器向消息 RAM 的传输可能会丢失。其原因是，当主机消息处理器（Host Message Handler）正在为同一消息对象接收消息并执行“读-修改-写”访问时，IFx 寄存器的写入操作恰好发生在其间。

接收邮箱（Receive Mailboxes）： 为避免此问题，在更改以下任何字段之前应重置 MsgVal：Id28-0、Xtd、Dir、DLC3-0、RxIE、TxIE、RmtEn、EoB、Umask、Msk28-0、MXtd 和 MDir。

发送邮箱（Transmit Mailboxes）： 为避免此问题，在更改以下任何字段之前应重置 MsgVal：Dir、RxIE、TxIE、RmtEn、EoB、Umask、Msk28-0、MXtd 和 MDir。

其他字段： 未列出的其他字段（如 Data）可以更改，无需担心写访问丢失。

### 19.13.1 消息接口寄存器组 1 和 2 (IF1 和 IF2)

IF1 和 IF2 寄存器组允许与消息对象之间进行数据传输。接口的 IFxCMD 寄存器用于控制数据传输的方向。

如果 IFxCMD 寄存器设置为写入（Write），则由 IFxCMD 寄存器选中的消息对象字段将被其他 IFx 寄存器中的值覆盖。

如果 IFxCMD 寄存器设置为读取（Read），则由 IFxCMD 寄存器选中的消息对象字段将从消息对象复制到其他 IFx 寄存器中。

接口允许传输整个消息对象或其各个独立部分。当预期的消息对象编号被写入 IFxCMD 寄存器的位 [7:0] 时，传输开始。当 CPU 发起传输时，消息处理器会将相应命令寄存器中的 Busy 位置为 1。传输完成后，Busy 位恢复为 0。（见图中）

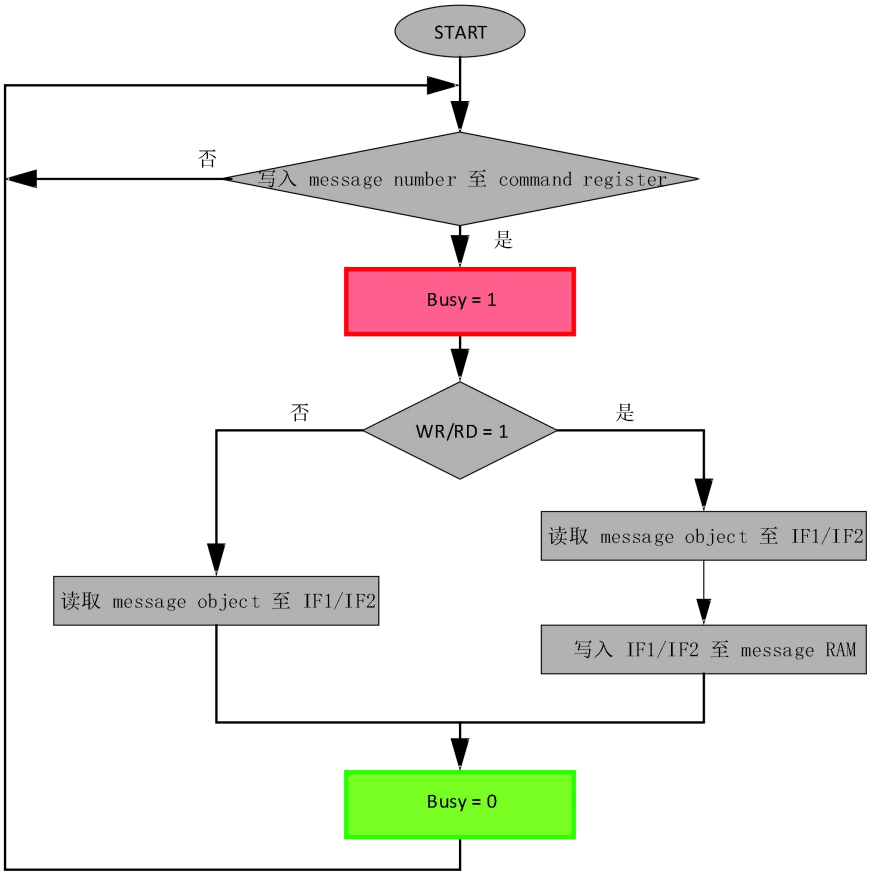



图 19-13 IF1/IF2 寄存器和消息 RAM 之间的数据传输

### 19.13.2 消息接口寄存器组 3 (IF3)

IF3 寄存器组可以自动从接收到的消息对象中更新，无需 CPU 发起从消息 RAM 的传输。

自动更新功能可以针对每个消息对象单独编程（参见 IF3 更新使能寄存器）。

消息 RAM 中所有配置为自动更新的有效消息对象都会被检查其 NewDat 标志是否激活。

如果发现此类对象，且当前没有 DMA 传输正在进行，该消息对象将根据 IF3 观察寄存器（Observation register）的控制传输到 IF3 寄存器。

如果有多个 NewDat 标志处于激活状态，编号最小的消息对象在自动 IF3 更新中具有最高优先级。

传输到 IF3 后，消息对象中的 NewDat 位将被重置。

当 CAN 模块内部的 IF3 更新完成后，将触发 DMA 请求。该 DMA 请求保持激活状态，直到对 IF3 寄存器进行第一次读访问为止。

必须通过设置 CAN 控制寄存器中的 DE3 位来使能 DMA 功能。

注意： IF3 寄存器组不能用于将数据传输到消息对象中。

## 19.14 消息 RAM 组织

CAN 消息 RAM 包含消息对象（Message Objects）以及用于消息对象的奇偶校验位。消息 RAM 中共有 32 个消息对象。在正常操作期间，对消息 RAM 的访问是通过接口寄存器组（Interface Register sets）完成的，CPU 无法直接访问消息 RAM。

接口寄存器组 IF1 和 IF2 为 CPU 提供了对消息 RAM 的间接读/写访问。IF1 和 IF2 寄存器组可以缓冲待传输至消息对象或从中传出的控制数据和用户数据。第三个接口寄存器组 IF3 可以配置为：当消息对象在接收到 CAN 消息并更新后，自动从消息 RAM 接收控制数据和用户数据。CPU 无需启动从消息 RAM 到 IF3 寄存器组的传输。

消息处理器（Message Handler）通过调度，避免并发访问消息 RAM 与 CAN 帧接收/发送之间可能产生的冲突。消息 RAM 仅在调试模式（Debug mode）下可以被直接访问。

### 19.14.1 消息对象结构

表中展示了消息对象的结构。灰色字段表示消息对象中由专用寄存器体现的部分。例如，所有消息对象的发送请求标志（Transmit request flags）都集中体现在集中的发送请求寄存器中。

表 19-7 消息对象的结构

| 消息对象 | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UMask | Msk[28:0] | MXtd | MDir | EoB | 未使用 | NewDat | MsgLst | RxIE | TxIE | IntPnd | RmtEn | TxRqst |
| MsgVal | ID[28:0] | Xtd | Dir | DLC[3:0] | Data 0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |

表 19-8 消息对象字段描述

| 名称 | 值 | 描述 |
|---|---|---|
| MsgVal | | 消息有效位 |
| | 0 | 消息对象被消息处理器忽略。 |
| | 1 | 消息对象将被消息处理器使用。注意：即使更改标识符位 ID[28:0]、控制位 Xtd、Dir 或数据长度代码 DLC[3:0]，该位也可以保持为 1。 |
| UMask | | 使用验收掩码 (Use Acceptance Mask) |
| | 0 | 忽略掩码位（Msk[28:0]、MXtd 和 MDir），不用于接收滤波。 |
| | 1 | 掩码位用于接收滤波。 注意：如果 UMask 位设置为 1，则必须在 MsgVal 设置为 1 之前的消息对象初始化期间编程掩码位。 |
| ID[28:0] | | 消息标识符 (Message Identifier) |
| | ID[28:0] | 29 位（“扩展”）标识符位。 |
| | ID[28:18] | 11 位（“标准”）标识符位。 |
| Msk[28:0] | | 标识符掩码 (Identifier Mask) |
| | 0 | 消息标识符中的对应位不用于接收滤波（无关位/don't care）。 |
| | 1 | 消息标识符中的对应位用于接收滤波。 注意： 本模块的位功能与旧款 eCAN 模块的局部验收掩码（Local Acceptance Mask，LAM）位功能相反：在 eCAN 中，1 表示对应位不用于过滤、0 表示用于过滤；本模块中 1 表示用于过滤、0 表示无关位。 |
| Xtd | | 扩展标识符位 (Extended Identifier) |
| | 0 | 该消息对象使用 11 位（“标准”）标识符。 |
| | 1 | 该消息对象使用 29 位（“扩展”）标识符。 |
| MXtd | | 掩码扩展标识符位 (Mask Extended Identifier) |
| | 0 | 扩展标识符位 (IDE) 对接收滤波无影响。 |
| | 1 | 扩展标识符位 (IDE) 用于接收滤波。 注意： 当消息对象使用 11 位标识符时，接收到的数据帧标识符被写入位 ID[28:18]。接收滤波仅考虑这些位以及掩码位 Msk[28:18]。 |
| Dir | | 消息方向 (Message Direction) |
| | 0 | 方向 = 接收：在 TxRqst 时，发送一个带有此消息对象标识符的远程帧。接收到具有匹配标识符的数据帧时，该消息存储在此消息对象中。 |
| | 1 | 方向 = 发送：在 TxRqst 时，发送一个数据帧。接收到具有匹配标识符的远程帧时，此消息对象的 TxRqst 位被置位（如果 RmtEn = 1）。 |
| MDir | | 掩码消息方向位 (Mask Message Direction) |
| | 0 | 消息方向位 (Dir) 对接收滤波无影响。 |
| | 1 | 消息方向位 (Dir) 用于接收滤波。 |
| EoB | | 块结束位 (End of Block) |
| | 0 | 该消息对象是 FIFO 缓冲块的一部分，且不是该 FIFO 缓冲块的最后一个消息对象。 |
| | 1 | 该消息对象是单个消息对象，或是 FIFO 缓冲块中的最后一个消息对象。 注意：此位用于连接多个消息对象以构建 FIFO 缓冲区。对于单个消息对象（不属于 FIFO），此位必须始终设置为 1。 |
| NewDat | | 新数据标志 (New Data) |

| 名称 | 值 | 描述 |
| --- | --- | --- |
| | 0 | 自上次 CPU 清除此标志以来，消息处理器未向此消息对象的数据字节写入新数据。 |
| | 1 | 消息处理器或 CPU 已向此消息对象的数据字节写入新数据。 |
| MsgLst | | 消息丢失标志 (Message Lost)（仅对接收方向的消息对象有效） |
| | 0 | 自上次 CPU 重置此位以来，没有消息丢失。 |
| | 1 | 消息处理器在 NewDat 仍为 1 时向此消息对象存储了新消息，因此之前的消息已被覆盖。 |
| RxIE | | 接收中断使能 (Receive Interrupt Enable) |
| | 0 | 成功接收一帧后不触发 IntPnd。 |
| | 1 | 成功接收一帧后触发 IntPnd。 |
| TxIE | | 发送中断使能 (Transmit Interrupt Enable) |
| | 0 | 成功发送一帧后不触发 IntPnd。 |
| | 1 | 成功发送一帧后触发 IntPnd。 |
| IntPnd | | 中断挂起标志 (Interrupt Pending) |
| | 0 | 此消息对象不是中断源。 |
| | 1 | 此消息对象是中断源。如果没有其他更高优先级的中断源，中断寄存器中的中断标识符指向此消息对象。 |
| RmtEn | | 远程使能 (Remote Enable) |
| | 0 | 接收到远程帧时，TxRqst 不变。 |
| | 1 | 接收到远程帧时，TxRqst 被置位。 |
| TxRqst | | 发送请求 (Transmit Request) |
| | 0 | 此消息对象未在等待传输。 |
| | 1 | 此消息对象已请求传输且尚未完成。 |
| DLC[3:0] | | 数据长度代码 (Data length code) |
| | 0-8 | 数据帧具有 0-8 个数据字节。 |
| | 9-15 | 数据帧具有 8 个数据字节。 注意：消息对象的数据长度代码必须定义为与其他节点上具有相同标识符的对应对象相同的值。当消息处理器存储数据帧时，它会将 DLC 写入为接收消息给出的值。 |
| Data 0-7 | | CAN 数据帧的第 1 至第 8 个数据字节 |
| | | 注意：Data 0 字节是接收期间移入 CAN 内核移位寄存器的第一个数据字节，Data 7 是最后一个。消息处理器存储数据帧时，会将所有 8 个数据字节写入消息对象。如果 DLC 小于 8，消息对象剩余的字节可能会被未定义的值覆盖。 |

## 19.14.2 在 RAM 中寻址消息对象 (Addressing Message Objects in RAM)

消息对象在 RAM 中的起始位置计算公式为：

消息 RAM 基地址 + (消息对象编号) * 0x20

这意味着消息对象 1 起始于偏移量 0x0020；消息对象 2 起始于偏移量 0x0040，依此类推 。

注意：

* 0 不是有效的消息对象编号。

* 在地址 0x0000 处，存放的是最后一个消息对象 (32)（具有最低优先级）。

* 写入未实现的消息对象地址可能会覆盖已实现的消息对象。

* 消息对象编号 1 具有最高优先级。

表 19-9 调试模式下的消息 RAM 寻址

| 消息对象编号 | 基地址偏移量 | 字编号 | 调试模式内容 |
|---|---|---|---|
| 最后实现的 (此处为 32) | 0x0000 | 1 | 奇偶校验 (Parity) |
| | 0x0004 | 2 | MXtd, MDir, Mask |
| | 0x0008 | 3 | Xtd, Dir, ID |
| | 0x000C | 4 | 控制位 (Ctrl) |
| | 0x0010 | 5 | 数据字节 3-0 |
| | 0x0014 | 6 | 数据字节 7-4 |
| 1 | 0x0020 | 1 | 奇偶校验 (Parity) |
| | 0x0024 | 2 | MXtd, MDir, Mask |
| | 0x0028 | 3 | Xtd, Dir, ID |
| | 0x002C | 4 | 控制位 (Ctrl) |
| | 0x0030 | 5 | 数据字节 3-0 |
| | 0x0034 | 6 | 数据字节 7-4 |
| 2 | 0x0040 | 1 | 奇偶校验 (Parity) |
| | 0x0044 | 2 | MXtd, MDir, Mask |
| | 0x0048 | 3 | Xtd, Dir, ID |
| | 0x004C | 4 | 控制位 (Ctrl) |
| | 0x0050 | 5 | 数据字节 3-0 |
| | 0x0054 | 6 | 数据字节 7-4 |
| ... | ... | ... | ... |
| 31 | 0x03E0 | 1 | 奇偶校验 (Parity) |
| | 0x03E4 | 2 | MXtd, MDir, Mask |
| | 0x03E8 | 3 | Xtd, Dir, ID |
| | 0x03EC | 4 | 控制位 (Ctrl) |
| | 0x03F0 | 5 | 数据字节 3-0 |
| | 0x03F4 | 6 | 数据字节 7-4 |

### 19.14.3 消息 RAM 地址映射（调试模式）

在调试模式下，消息 RAM 是内存映射的。这允许外部调试单元访问消息 RAM。

注意： 在调试模式期间，无法使用 IFx 寄存器组访问消息 RAM。

表 19-10 调试模式下的消息 RAM 表示

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

MsgAddr + 0x00

表 19-11 消息 RAM 地址映射（调试模式）数据表

| 保留 | |
|---|---|
| 保留 | Parity[4:0] |

MsgAddr + 0x04

表 19-12 消息 RAM 地址映射（调试模式）数据表

| MXtd | MDir | 保留 | Msk[28-16] |
|---|---|---|---|
| Msk[15-0] | | | |

MsgAddr + 0x08

表 19-13 消息 RAM 地址映射（调试模式）数据表

| 保留 | Xtd | Dir | ID[28:16] |
|---|---|---|---|
| ID[15:0] | | | |

MsgAddr + 0x0C

表 19-14 消息 RAM 地址映射（调试模式）数据表

| 保留 | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 保留 | MsgLst | 保留 | UMask | TxIE | RxIE | RmtEn | 保留 | EOB | 保留 | DLC[3:0] |

MsgAddr + 0x10

表 19-15 消息 RAM 地址映射（调试模式）数据表

| Data3 | Data2 |
|---|---|
| Data1 | Data0 |

MsgAddr + 0x14

表 19-16 消息 RAM 地址映射（调试模式）数据表

| Data7 | Data6 |
|---|---|
| Data5 | Data4 |

## 19.15 寄存器描述

表中列出 CAN 模块的实例名、寄存器结构体名、DriverLib 名称与基地址，并标明该实例是否支持 DMA、是否支持 HIC（主机接口控制器）以及是否受流水线保护。

表 19-17 CAN 基地址表

| 实例 | 结构体 | DriverLib 名称 | 基地址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| CanaRegs | CAN_REGS | CANA_BASE | 0x0067_2000 | 是 | 是 | 是 |

### 19.15.1 访问类型与使用约定

寄存器字段表采用下列访问类型。写寄存器前应同时确认 Init、CCE、Test、Busy 或密钥字段等访问保护条件。

表 19-18 寄存器字段表访问类型

| 访问类型 | 含义 | 使用说明 |
|---|---|---|
| R | 只读 | 软件只能读取，写入无效。 |
| R/W | 读/写 | 软件可以读取和写入。 |
| W | 只写 | 软件写入有效；读取值不用于判断状态。 |
| R-0/W1C | 读 0/写 1 清零 | 读取返回 0；向该位写 1 执行清除或触发动作。 |
| X | 不确定 | 复位后值不确定，使用前必须由软件确认。 |

**注：**CAN_ES 中的 PER、RxOk、TxOk 为读清除位，读取该寄存器即将其清零，读取值反映清零前的状态。

### 19.15.2 寄存器汇总

表 19-19 CAN 寄存器汇总

| 偏移地址 | 寄存器 | 中文名称 | 复位值 |
|---|---|---|---|
| 0h | CAN_CTL | CAN 控制寄存器 | 00001401h |
| 4h | CAN_ES | 错误与状态寄存器 | 00000007h |
| 8h | CAN_ERRC | 错误计数器寄存器 | 00000000h |
| Ch | CAN_BTR | 位定时寄存器 | 00002301h |
| 10h | CAN_INT | 中断源寄存器 | 00000000h |

| 偏移地址 | 寄存器 | 中文名称 | 复位值 |
| --- | --- | --- | --- |
| 14h | CAN_TEST | 测试寄存器 | 00000000h |
| 1Ch | CAN_PERR | 奇偶校验错误代码寄存器 | 00000XXXh |
| 40h | CAN_RAM_INIT | CAN RAM 初始化寄存器 | 00000005h |
| 50h | CAN_GLB_INT_EN | CAN 全局中断使能寄存器 | 00000000h |
| 54h | CAN_GLB_INT_FLG | CAN 全局中断标志寄存器 | 00000000h |
| 58h | CAN_GLB_INT_CLR | CAN 全局中断清除寄存器 | 00000000h |
| 80h | CAN_ABOTR | 自动总线恢复定时寄存器 | 00000000h |
| 84h | CAN_TXRQ_X | 发送请求组状态寄存器 | 00000000h |
| 88h | CAN_TXRQ_21 | 发送请求寄存器 | 00000000h |
| 98h | CAN_NDAT_X | 新数据组状态寄存器 | 00000000h |
| 9Ch | CAN_NDAT_21 | 新数据寄存器 | 00000000h |
| ACh | CAN_IPEN_X | 中断挂起组状态寄存器 | 00000000h |
| B0h | CAN_IPEN_21 | 中断挂起寄存器 | 00000000h |
| C0h | CAN_MVAL_X | 消息有效组状态寄存器 | 00000000h |
| C4h | CAN_MVAL_21 | 消息有效寄存器 | 00000000h |
| D8h | CAN_IP_MUX21 | 消息中断路由寄存器 | 00000000h |
| 100h | CAN_IF1CMD | IF1 命令寄存器 | 00000001h |
| 104h | CAN_IF1MSK | IF1 验收掩码寄存器 | FFFFFFFFh |
| 108h | CAN_IF1ARB | IF1 仲裁寄存器 | 00000000h |
| 10Ch | CAN_IF1MCTL | IF1 消息控制寄存器 | 00000000h |
| 110h | CAN_IF1DATA | IF1 数据 A 寄存器 | 00000000h |
| 114h | CAN_IF1DATB | IF1 数据 B 寄存器 | 00000000h |
| 120h | CAN_IF2CMD | IF2 命令寄存器 | 00000001h |
| 124h | CAN_IF2MSK | IF2 验收掩码寄存器 | FFFFFFFFh |
| 128h | CAN_IF2ARB | IF2 仲裁寄存器 | 00000000h |
| 12Ch | CAN_IF2MCTL | IF2 消息控制寄存器 | 00000000h |
| 130h | CAN_IF2DATA | IF2 数据 A 寄存器 | 00000000h |
| 134h | CAN_IF2DATB | IF2 数据 B 寄存器 | 00000000h |
| 140h | CAN_IF3OBS | IF3 观察寄存器 | 00000000h |

| 偏移地址 | 寄存器 | 中文名称 | 复位值 |
|---|---|---|---|
| 144h | CAN_IF3MSK | IF3 验收掩码寄存器 | FFFFFFFFh |
| 148h | CAN_IF3ARB | IF3 仲裁寄存器 | 00000000h |
| 14Ch | CAN_IF3MCTL | IF3 消息控制寄存器 | 00000000h |
| 150h | CAN_IF3DATA | IF3 数据 A 寄存器 | 00000000h |
| 154h | CAN_IF3DATB | IF3 数据 B 寄存器 | 00000000h |
| 160h | CAN_IF3UPD | IF3 自动更新使能寄存器 | 00000000h |

### 19.15.2.1 CAN_CTL 寄存器

偏移地址：0h　复位值：00001401h

该寄存器用于配置 CAN 模块的中断、奇偶校验、调试模式行为等。

表 19-20 CAN_CTL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | 保留 | 保留 |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | DE3 | DE2 | DE1 | IE1 | INITDBG |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SWR | 保留 | PMD | | | | ABO | IDS |
| R-0/W1C-0h | R-0h | R/W-5h | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Test | CCE | DAR | 保留 | EIE | SIE | IE0 | Init |
| R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-1h |

表 19-21 CAN_CTL 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | 保留。 |
| 25 | RESERVED | R/W | 0h | 保留。 |
| 24 | RESERVED | R/W | 0h | 保留。 |
| 23-21 | RESERVED | R | 0h | 保留。 |
| 20 | DE3 | R/W | 0h | 使能 IF3 的 DMA 请求线。<br>0 禁止 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 1 使能<br>注：IF3 的挂起 DMA 请求将保持有效，直到首次访问 IF3 寄存器之一。<br>复位类型：SYSRSn |
| 19 | DE2 | R/W | 0h | 使能 IF2 的 DMA 请求线。<br>0 禁止<br>1 使能<br>注：IF2 的挂起 DMA 请求将保持有效，直到首次访问 IF2 寄存器之一。<br>复位类型：SYSRSn |
| 18 | DE1 | R/W | 0h | 使能 IF1 的 DMA 请求线。<br>0 禁止<br>1 使能<br>注：IF1 的挂起 DMA 请求将保持有效，直到首次访问 IF1 寄存器之一。<br>复位类型：SYSRSn |
| 17 | IE1 | R/W | 0h | 中断线 1 使能。<br>0 CANINT1 被禁止。<br>1 CANINT1 被使能。中断将把 CANINT1 线置为 1，该线保持有效，直到挂起的中断被处理完毕。<br>复位类型：SYSRSn |
| 16 | INITDBG | R | 0h | 调试模式状态位：该位指示用于调试访问的内部初始化状态。<br>0 未处于调试模式，或已请求调试模式但尚未进入。<br>1 已请求调试模式并已在内部进入，CAN 模块已准备好接受调试访问。<br>复位类型：SYSRSn |
| 15 | SWR | R-0/W1C | 0h | 软件复位使能位：该位激活软件复位。<br>0 正常工作。<br>1 模块被强制进入复位状态。该位将在执行软件复位后的一个时钟周期自动清零。<br>注：要执行软件复位，必须遵循以下步骤：<br>• 1. 置位 INIT 位，关闭 CAN 通信。<br>• 2. 置位 SWR 位。<br>注：该位受 Init 位写保护。若使用 SWR 位复位模块，则不会丢失任何用户配置。只有状态位以及为下一次 CAN 事务需要复位的逻辑会一起被复位。若使用 SOFTPRES 寄存器复位模块，则整个模块都会被复位，包括配置寄存器。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 14 | RESERVED | R | 0h | 保留。 |
| 13-10 | PMD | R/W | 5h | 奇偶校验开/关。<br>0101 奇偶校验功能被禁止<br>任何其他值 奇偶校验功能被使能<br>复位类型：SYSRSn |
| 9 | ABO | R/W | 0h | 自动总线开启使能。<br>0 自动总线开启（Auto-Bus-On）功能被禁止<br>1 自动总线开启（Auto-Bus-On）功能被使能<br>复位类型：SYSRSn |
| 8 | IDS | R/W | 0h | 中断调试支持使能。<br>0 当请求调试模式时，CAN 模块将等待已经开始的一次发送或接收完成后再进入调试模式。<br>1 当请求调试模式时，CAN 模块将中断任何发送或接收，并立即进入调试模式。<br>复位类型：SYSRSn |
| 7 | Test | R/W | 0h | 测试模式使能。<br>0 禁止测试模式（正常工作）<br>1 使能测试模式<br>复位类型：SYSRSn |
| 6 | CCE | R/W | 0h | 配置更改使能。<br>0 CPU 对配置寄存器没有写访问权限。<br>1 CPU 对配置寄存器有写访问权限（当 Init 位置位时）。<br>复位类型：SYSRSn |
| 5 | DAR | R/W | 0h | 禁止自动重传。<br>0 使能“不成功”消息的自动重传。<br>1 禁止自动重传。<br>复位类型：SYSRSn |
| 4 | RESERVED | R | 0h | 保留。 |
| 3 | EIE | R/W | 0h | 错误中断使能。<br>0 禁止——PER、BOff 和 EWarn 位不能产生中断。<br>1 使能——PER、BOff 和 EWarn 位可以在 CANINT0 线上产生中断，并影响中断寄存器。<br>复位类型：SYSRSn |
| 2 | SIE | R/W | 0h | 状态变化中断使能。<br>0 禁止——RxOk、TxOk 和 LEC 位不能产生中断。<br>1 使能——RxOk、TxOk 和 LEC 可以在 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | CANINT0 线上产生中断。<br>复位类型：SYSRSn |
| 1 | IE0 | R/W | 0h | 中断线 0 使能。<br>0 CANINT0 被禁止。<br>1 CANINT0 被使能。中断将把 CANINT0 线置为 1，该线保持有效，直到挂起的中断被处理完毕。<br>复位类型：SYSRSn |
| 0 | Init | R/W | 1h | 初始化模式。<br>该位用于在位定时配置和消息 RAM 初始化期间使 CAN 模块保持非活动状态。在发生总线关闭（bus off）事件期间，该位会被自动置位。清除该位不会缩短总线恢复时间。<br>0 CAN 模块正常处理消息<br>1 CAN 模块忽略总线活动<br>复位类型：SYSRSn |

### 19.15.2.2 CAN_ES 寄存器

偏移地址：4h　复位值：00000007h

该寄存器指示 CAN 模块的错误状态（如果有）。中断由 PER、BOff 和 EWarn 位产生（前提是 CAN 控制寄存器中的 EIE 位已置位），并由 RxOk、TxOk 和 LEC 位产生（前提是 CAN 控制寄存器中的 SIE 位已置位）。EPass 位的变化不会产生中断。读取错误与状态寄存器会清除 PER、RxOk 和 TxOk 位，并将 LEC 置为值“7”。此外，中断寄存器中的状态中断值（0x8000）将被替换为下一个较低优先级的中断值。为便于调试支持，在调试/挂起模式下，错误与状态寄存器的自动清除功能（通过读取清除状态标志）被禁止。

表 19-22 CAN_ES 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | 保留 | 保留 | PER |
| R-0h | | | | | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BOff | EWarn | EPass | RxOk | TxOk | LEC | | |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-7h | | |

表 19-23 CAN_ES 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | 保留。 |
| 10 | RESERVED | R | 0h | 保留。 |
| 9 | RESERVED | R | 0h | 保留。 |
| 8 | PER | R | 0h | 检测到奇偶校验错误：该位将在 CPU 读取本寄存器后被复位。<br>0 自上次读访问以来未检测到奇偶校验错误。<br>1 奇偶校验机制在消息 RAM 中检测到奇偶校验错误。<br>复位类型：SYSRSn |
| 7 | BOff | R | 0h | 总线关闭状态位：<br>0 CAN 模块未处于总线关闭（Bus-Off）状态。<br>1 CAN 模块处于总线关闭（Bus-Off）状态。<br>复位类型：SYSRSn |
| 6 | EWarn | R | 0h | 警告状态位：<br>0 两个错误计数器都低于 96 的错误警告限值。<br>1 至少有一个错误计数器已达到 96 的错误警告限值。<br>复位类型：SYSRSn |
| 5 | EPass | R | 0h | 错误被动状态。<br>0 在 CAN 总线出错时，CAN 可以发送主动错误帧。<br>1 CAN 内核处于 CAN 规范所定义的错误被动状 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 态。<br>复位类型：SYSRSn |
| 4 | RxOk | R | 0h | 接收状态位：该位指示接收状态。该位将在 CPU 读取本寄存器后被复位。<br>0 自上次该位被 CPU 读取以来，没有消息被成功接收。该位从不会被 CAN 内部事件复位。<br>1 自上次该位被 CPU 读访问复位以来，已有消息被成功接收。该位的置位与接收滤波的结果无关。<br>复位类型：SYSRSn |
| 3 | RxOk | R | 0h | 发送状态位：该位指示发送状态。该位将在 CPU 读取本寄存器后被复位。<br>0 自上次该位被 CPU 读取以来，没有消息被成功发送。该位从不会被 CAN 内部事件复位。<br>1 自上次该位被 CPU 读访问清除以来，已有消息被成功发送（无错误且被至少一个其他节点应答）。<br>复位类型：SYSRSn |
| 2-0 | LEC | R | 7h | 最后错误代码。<br>LEC 字段指示 CAN 总线上最后一次错误的类型。当一条消息被无错误地传输（接收或发送）时，该字段将被清为“0”。每当 CPU 读取本寄存器时，该字段将被复位为“7”。<br>0 无错误<br>1 填充错误（Stuff Error）：在接收消息的某个不允许出现的位置上检测到连续多于五个相同的位。<br>2 格式错误（Form Error）：接收帧的某个固定格式部分格式错误。<br>3 应答错误（Ack Error）：该 CAN 内核发送的消息未被另一个节点应答。<br>4 位 1 错误（Bit1 Error）：在消息发送期间（仲裁字段除外），器件想要发送一个隐性电平（逻辑值“1”的位），但监视到的总线值为显性。<br>5 位 0 错误（Bit0 Error）：在消息发送期间（或应答位、或主动错误标志、或过载标志期间），器件想要发送一个显性电平（逻辑值“0”），但监视到的总线电平为隐性。在总线关闭恢复期间，每当监视到 11 个连续隐性位时就会置位该状态。这使 CPU 能够监视总线关闭恢复序列的进展（表明总线未卡在显性状态或未被持续干扰）。<br>6 CRC 错误（CRC Error）：在接收的消息中，CRC 校验和错误。（收到的报文 CRC 与针对所接收数据计算出的 CRC 不匹配）。<br>7 自上次 CPU 读取错误与状态寄存器以来，未检 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 测到 CAN 总线事件。对错误与状态寄存器的任何读访问都会将 LEC 重新初始化为值“7”。<br>复位类型：SYSRSn |

### 19.15.2.3 CAN_ERRC 寄存器

偏移地址：8h　复位值：00000000h

该寄存器反映发送错误计数器和接收错误计数器的值。

表 19-24 CAN_ERRC 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RP | REC | | | | | | |
| R-0h | R-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TEC | | | | | | | |
| R-0h | | | | | | | |

表 19-25 CAN_ERRC 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15 | RP | R | 0h | 接收错误被动。<br>0 接收错误计数器低于错误被动电平。<br>1 接收错误计数器已达到 CAN 规范所定义的错误被动电平。<br>复位类型：SYSRSn |
| 14-8 | REC | R | 0h | 接收错误计数器。<br>接收错误计数器的实际状态（取值范围为 0 到 127）。<br>复位类型：SYSRSn |
| 7-0 | TEC | R | 0h | 发送错误计数器。<br>发送错误计数器的实际状态（取值范围为 0 到 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 255）。<br>复位类型：SYSRSn |

### 19.15.2.4 CAN_BTR 寄存器

偏移地址：Ch　复位值：00002301h

该寄存器用于配置 CAN 模块的位定时参数。只有当 CAN 控制寄存器中的 CCE 和 Init 位均置位时，该寄存器才可写。CAN 位时间可在 8 到 25 个时间量子的范围内编程。CAN 时间量子可在 1 到 1024 个 CAN_CLK 周期的范围内编程。

表 19-26 CAN_BTR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | BRPE | | | |
| R-0h | | | | R/W-0h | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | TSEG2 | | | TSEG1 | | | |
| R-0h | R/W-2h | | | R/W-3h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SJW | | BRP | | | | | |
| R/W-0h | | R/W-1h | | | | | |

表 19-27 CAN_BTR 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | 保留。 |
| 19-16 | BRPE | R/W | 0h | 波特率预分频器扩展。<br>有效的编程值为 0 到 15。<br>通过编程 BRPE，波特率预分频器可扩展到最大 1024 的值。<br>注：该位受 CCE 位写保护。<br>复位类型：SYSRSn |
| 15 | RESERVED | R | 0h | 保留。 |
| 14-12 | TSEG2 | R/W | 2h | 采样点之后的时间段。有效的编程值为 0 到 7。用于位定时的实际 TSeg2 值将为编程的 TSeg2 值加 1。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 注：该位受 CCE 位写保护。<br>复位类型：SYSRSn |
| 11-8 | TSEG1 | R/W | 3h | 采样点之前的时间段。有效的编程值为 1 到 15。用于位定时的实际 TSeg1 值将为编程的 TSeg1 值加 1。<br>注：该位受 CCE 位写保护。<br>复位类型：SYSRSn |
| 7-6 | SJW | R/W | 0h | 同步跳转宽度。有效的编程值为 0 到 3。用于同步的实际 SJW 值将为编程的 SJW 值加 1。<br>注：该位受 CCE 位写保护。<br>复位类型：SYSRSn |
| 5-0 | BRP | R/W | 1h | 波特率预分频器。<br>用于生成位时间量子的 CAN_CLK 频率分频值。位时间由该量子的整数倍构成。有效的编程值为 0 到 63。<br>用于位定时的实际 BRP 值将为编程的 BRP 值加 1。<br>注：该位受 CCE 位写保护。<br>复位类型：SYSRSn |

### 19.15.2.5 CAN_INT 寄存器

偏移地址：10h　复位值：00000000h

该寄存器用于识别中断源。

表 19-28 CAN_INT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| INT1ID | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| INT0ID | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT0ID | | | | | | | |
| R-0h | | | | | | | |

表 19-29 CAN_INT 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | 保留。 |
| 23-16 | INT1ID | R | 0h | 中断 1 原因。<br>0x00 无挂起的中断。<br>0x01-0x20 引起该中断的消息对象（邮箱）编号。<br>0x21-0xFF 未使用。<br>如果有多个中断挂起，CAN 中断寄存器将指向优先级最高的挂起中断。<br>注：CANINT1 中断线保持有效，直到 INT1ID 达到值 0（中断原因被复位）或直到 IE0 被清除。消息中断通过清除该邮箱的 IntPnd 位来清除。在消息中断中，邮箱的中断优先级随消息编号增大而降低。<br>复位类型：SYSRSn |
| 15-0 | INT0ID | R | 0h | 中断 0 原因。<br>0x0000 - 无挂起的中断。<br>0x0001 - 0x0020 - 引起该中断的消息对象编号。<br>0x0021 - 0x7FFF - 未使用。<br>0x8000 - 错误与状态寄存器的值不为 0x07。<br>0x8001 - 0xFFFF - 未使用。<br>如果有多个中断挂起，CAN 中断寄存器将指向优先级最高的挂起中断。<br>注：CANINT0 中断线保持有效，直到 INT0ID 达到值 0（中断原因被复位）或直到 IE0 被清除。状态中断具有最高优先级。在消息中断中，消息对象的中断优先级随消息编号增大而降低。<br>复位类型：SYSRSn |

### 19.15.2.6 CAN_TEST 寄存器

偏移地址：14h　复位值：00000000h

该寄存器用于配置所支持的各种测试选项。对于所有测试模式，CAN 控制寄存器中的 Test 位都需要置为 1。如果 Test 位置位，则 RDA、EXL、Tx1、Tx0、LBack 和 Silent 位可写。Rx 位监视 CANRX 引脚的状态，因此只可读。当 Test 位被清除时，所有测试寄存器功能都被禁止。

**注意：**将 Tx[1:0] 设为非 00 的值会干扰报文传输。

**注意：**当环回模式激活（LBack 位置 1）时，EXL 位将被忽略。

表 19-30 CAN_TEST 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | RDA | EXL |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RX | TX | | LBACK | SILENT | 保留 | | |
| R-0h | R/W-0h | | R/W-0h | R/W-0h | R-0h | | |

表 19-31 CAN_TEST 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | 保留。 |
| 9 | RDA | R/W | 0h | RAM 直接访问使能：<br>0 正常工作。<br>1 在测试模式下使能对 RAM 的直接访问。<br>复位类型：SYSRSn |
| 8 | EXL | R/W | 0h | 外部回环模式：<br>0 禁止。<br>1 使能。<br>复位类型：SYSRSn |
| 7 | RX | R | 0h | 监视 CANRX 引脚的实际值：<br>0 CAN 总线为显性。<br>1 CAN 总线为隐性。<br>复位类型：SYSRSn |
| 6-5 | TX | R/W | 0h | CANTX 引脚的控制：<br>00 正常工作，CANTX 由 CAN 内核控制。<br>01 可在 CANTX 引脚上监视采样点。<br>10 CANTX 引脚驱动显性值。<br>11 CANTX 引脚驱动隐性值。<br>复位类型：SYSRSn |
| 4 | LBACK | R/W | 0h | 回环模式：<br>0 禁止。<br>1 使能。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 3 | SILENT | R/W | 0h | 静默模式：<br>0 禁止。<br>1 使能。<br>复位类型：SYSRSn |
| 2-0 | RESERVED | R | 0h | 保留。 |

### 19.15.2.7 CAN_PERR 寄存器

偏移地址：1Ch　复位值：00000XXXh

该寄存器指示检测到奇偶校验错误的字/邮箱编号。如果检测到奇偶校验错误，错误与状态寄存器中的 PER 标志将被置位。该位不会被奇偶校验机制复位，必须通过读取错误与状态寄存器来复位。除 PER 标志外，奇偶校验错误代码寄存器还将指示检测到奇偶校验错误的存储区。如果检测到多于一个发生奇偶校验错误的字，将显示带有奇偶校验错误的最高字编号。检测到奇偶校验错误后，该寄存器将保持最后一次错误代码，直到断电为止。

表 19-32 CAN_PERR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | WORD_NUM | | |
| R-0h | | | | | R-X | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MSG_NUM | | | | | | | |
| R-X | | | | | | | |

表 19-33 CAN_PERR 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-11 | RESERVED | R | 0h | 保留。 |
| 10-8 | WORD_NUM | R | X | 0x01-0x05 检测到奇偶校验错误的字编号。<br>邮箱的 RDA 字编号（1 到 5）（依据 RDA 模 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 式下的消息 RAM 表示）。<br>复位类型：SYSRSn |
| 7-0 | MSG_NUM | R | X | 0x01-0x21 检测到奇偶校验错误的邮箱编号。<br>复位类型：SYSRSn |

### 19.15.2.8 CAN_RAM_INIT 寄存器

偏移地址：40h 复位值：00000005h

该寄存器用于初始化邮箱 RAM。它会清除整个邮箱 RAM，包括 MsgVal 位。

表 19-34 CAN_RAM_INIT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | RAM_INIT_DONE | CAN_RAM_INIT | KEY3 | KEY2 | KEY1 | KEY0 |
| R-0h | | R-0h | R/W-0h | R/W-0h | R/W-1h | R/W-0h | R/W-1h |

表 19-35 CAN_RAM_INIT 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-6 | RESERVED | R | 0h | 保留。 |
| 5 | RAM_INIT_DONE | R | 0h | CAN 邮箱 RAM 初始化状态：<br>0 读：初始化正在进行中，或初始化尚未启动。<br>1 读：初始化完成。<br>复位类型：SYSRSn |
| 4 | CAN_RAM_INIT | R/W | 0h | 启动 CAN 邮箱 RAM 初始化：<br>0 读：初始化已完成或初始化尚未启动。<br>写：无操作。<br>1 读：初始化正在进行中。<br>写：启动 CAN 邮箱 RAM 初始化。初始化完成 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 后，该位将被自动清为 0。<br>复位类型：SYSRSn |
| 3 | KEY3 | R/W | 0h | 参见 Key 0。<br>复位类型：SYSRSn |
| 2 | KEY2 | R/W | 1h | 参见 Key 0。<br>复位类型：SYSRSn |
| 1 | KEY1 | R/W | 0h | 参见 Key 0。<br>复位类型：SYSRSn |
| 0 | KEY0 | R/W | 1h | KEY3-KEY0 必须为 0101（即 0x5），对该寄存器的任何写操作才有效。<br>CAN RAM 初始化完成后，这些位将恢复为其复位状态。<br>复位类型：SYSRSn |

### 19.15.2.9 CAN_GLB_INT_EN 寄存器

偏移地址：50h　复位值：00000000h

该寄存器用于使能到 CLIC 的中断线。

表 19-36 CAN_GLB_INT_EN 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | GLBINT1_EN | GLBINT0_EN |
| R-0h | | | | | | R/W-0h | R/W-0h |

表 19-37 CAN_GLB_INT_EN 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 1 | GLBINT1_EN | R/W | 0h | CANINT1 的全局中断使能。<br>0 CANINT1 不向 CLIC 产生中断。<br>1 如果发生中断条件，CANINT1 向 CLIC 产生中断。<br>复位类型：SYSRSn |
| 0 | GLBINT0_EN | R/W | 0h | CANINT0 的全局中断使能。<br>0 CANINT0 不向 CLIC 产生中断。<br>1 如果发生中断条件，CANINT0 向 CLIC 产生中断。<br>复位类型：SYSRSn |

### 19.15.2.10 CAN_GLB_INT_FLG 寄存器

偏移地址：54h 复位值：00000000h

该寄存器指示到 CLIC 的中断线是否以及何时处于活动状态。

表 19-38 CAN_GLB_INT_FLG 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | INT1_FLG | INT0_FLG |
| R-0h | | | | | | R-0h | R-0h |

表 19-39 CAN_GLB_INT_FLG 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留。 |
| 1 | INT1_FLG | R | 0h | CANINT1 标志。<br>0 未产生中断。<br>1 由于 CANINT1 而产生中断（中断条件参见 CAN 中断状态寄存器）。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 0 | INT0_FLG | R | 0h | CANINT0 标志。<br>0 未产生中断。<br>1 由于 CANINT0 而产生中断（中断条件参见 CAN 中断状态寄存器）。<br>复位类型：SYSRSn |

### 19.15.2.11 CAN_GLB_INT_CLR 寄存器

偏移地址：58h　复位值：00000000h

该寄存器用于清除到 CLIC 的中断。

表 19-40 CAN_GLB_INT_CLR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | INT1_FLG_CLR | INT0_FLG_CLR |
| R-0h | | | | | | W-0h | W-0h |

表 19-41 CAN_GLB_INT_CLR 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留。 |
| 1 | INT1_FLG_CLR | W | 0h | CANINT1 的全局中断标志清除。<br>0 无影响。<br>1 写 1 以清除全局中断标志寄存器的对应位，并允许 CLIC 接收来自 CANINT1 的下一个中断。<br>复位类型：SYSRSn |
| 0 | INT0_FLG_CLR | W | 0h | CANINT0 的全局中断标志清除。<br>0 无影响。<br>1 写 1 以清除全局中断标志寄存器的对应位，并允许 CLIC 接收来自 CANINT0 的下一个中断。<br>复位类型：SYSRSn |

### 19.15.2.12 CAN_ABOTR 寄存器

偏移地址：80h　复位值：00000000h

该寄存器用于在总线关闭恢复序列开始之前引入一个可变延时。

表 19-42 CAN_ABOTR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| ABO_Time | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ABO_Time | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ABO_Time | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ABO_Time | | | | | | | |
| R/W-0h | | | | | | | |

表 19-43 CAN_ABOTR 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | ABO_Time | R/W | 0h | 自动总线开启定时器。<br>在通过清除 Init 位启动总线关闭恢复序列之前的时钟周期数。“Clock”指 CAN 模块的输入时钟。该功能必须通过置位 CAN 控制寄存器中的 ABO 位来使能。<br>自动总线开启定时器由一个 32 位计数器实现，当模块进入总线关闭状态时该计数器开始向下计数至零。在此阶段之后，计数器将重新装载 ABO Time 寄存器的预装载值。<br>注：当自动总线开启定时器正在运行时对 CAN 控制寄存器进行写访问，自动总线开启过程将被中止。<br>注：在调试模式下，正在运行的自动总线开启定时器将被暂停。<br>复位类型：SYSRSn |

### 19.15.2.13 CAN_TXRQ_X 寄存器

偏移地址：84h　复位值：00000000h

借助这些位，CPU 可以检测 CAN 发送请求 21 寄存器（CAN_TXRQ_21）中是否有一个或多个位置位。该寄存器中的每一位代表一组八个邮箱。如果这些消息对象中至少有一个 TxRqst 位置位，则该寄存器中的对应位将被置位。

表 19-44 CAN_TXRQ_X 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | TxRqstReg2 | | TxRqstReg1 | |
| R-0h | | | | R-0h | | R-0h | |

表 19-45 CAN_TXRQ_X 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | 保留。 |
| 3-2 | TxRqstReg2 | R | 0h | 发送请求寄存器 2 标志：<br>位 2 代表 CAN_TXRQ_21 的字节 2。如果该字节中有一个或多个位置位，则位 2 将被置位。<br>位 3 代表 CAN_TXRQ_21 寄存器的字节 3。如果该字节中有一个或多个位置位，则位 3 将被置位。<br>复位类型：SYSRSn |
| 1-0 | TxRqstReg1 | R | 0h | 发送请求寄存器 1 标志：<br>位 0 代表 CAN_TXRQ_21 寄存器的字节 0。如果该字节中有一个或多个位置位，则位 0 将被置位。<br>位 1 代表 CAN_TXRQ_21 寄存器的字节 1。如果该字节中有一个或多个位置位，则位 1 将被置位。<br>复位类型：SYSRSn |

### 19.15.2.14 CAN_TXRQ_21 寄存器

偏移地址：88h　复位值：00000000h

该寄存器保存各邮箱的 TxRqst 位。通过读出这些位，CPU 可以检查是否有挂起的发送请求。特定邮箱中的 TxRqst 位可由 CPU 通过 IF1/IF2 消息接口寄存器置位/复位，也可由消息处理器在接收到远程帧之后或在成功发送之后置位/复位。

表 19-46 CAN_TXRQ_21 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TxRqst | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TxRqst | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TxRqst | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TxRqst | | | | | | | |
| R-0h | | | | | | | |

表 19-47 CAN_TXRQ_21 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | TxRqst | R | 0h | 发送请求位（针对所有消息对象）。<br>0 未请求发送该消息对象。<br>1 该消息对象的发送已被请求且尚未完成。<br>注：位 0 对应邮箱 1，位 1 对应邮箱 2，位 2 对应邮箱 3，……，位 31 对应邮箱 32。<br>复位类型：SYSRSn |

### 19.15.2.15 CAN_NDAT_X 寄存器

偏移地址：98h　复位值：00000000h

借助这些位，CPU 可以检测 CAN 新数据 21 寄存器（CAN_NDAT_21）中是否有一个或多个位置位。该寄存器中的每一位代表一组八个邮箱。如果这些邮箱中至少有一个 NewDat 位置位，则该寄存器中的对应位将被置位。

表 19-48 CAN_NDAT_X 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | NewDatReg2 | | NewDatReg1 | |
| R-0h | | | | R-0h | | R-0h | |

表 19-49 CAN_NDAT_X 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | 保留。 |
| 3-2 | NewDatReg2 | R | 0h | 新数据寄存器 2 标志：<br>位 2 代表 CAN_NDAT_21 寄存器的字节 2。如果该字节中有一个或多个位置位，则位 2 将被置位。<br>位 3 代表 CAN_NDAT_21 寄存器的字节 3。如果该字节中有一个或多个位置位，则位 3 将被置位。<br>复位类型：SYSRSn |
| 1-0 | NewDatReg1 | R | 0h | 新数据寄存器 1 标志：<br>位 0 代表 CAN_NDAT_21 寄存器的字节 0。如果该字节中有一个或多个位置位，则位 0 将被置位。<br>位 1 代表 CAN_NDAT_21 寄存器的字节 1。如果该字节中有一个或多个位置位，则位 1 将被置位。<br>复位类型：SYSRSn |

### 19.15.2.16 CAN_NDAT_21 寄存器

偏移地址：9Ch　复位值：00000000h

该寄存器保存所有邮箱的 NewDat 位。通过读出 NewDat 位，CPU 可以检查哪些邮箱的数据部分已被更新。特定邮箱的 NewDat 位可由 CPU 通过 IFx“消息接

口”寄存器置位/复位，也可由消息处理器在接收到数据帧之后或在成功发送之后置位/复位。

表 19-50 CAN_NDAT_21 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| NewDat | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| NewDat | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| NewDat | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| NewDat | | | | | | | |
| R-0h | | | | | | | |

表 19-51 CAN_NDAT_21 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | NewDat | R | 0h | 新数据位（针对所有消息对象）。<br>0 自上次该标志被 CPU 清除以来，消息处理器未向该消息对象的数据部分写入新数据。<br>1 消息处理器或 CPU 已向该消息对象的数据部分写入新数据。<br>注：位 0 对应邮箱 1，位 1 对应邮箱 2，位 2 对应邮箱 3，……，位 31 对应邮箱 32。<br>复位类型：SYSRSn |

## 19.15.2.17 CAN_IPEN_X 寄存器

偏移地址：ACh　复位值：00000000h

借助这些位，CPU 可以检测 CAN 中断挂起 21 寄存器（CAN_IPEN_21）中是否有一个或多个位置位。该寄存器中的每一位代表一组八个邮箱。如果这些邮箱中至少有一个 IntPnd 位置位，则该寄存器中的对应位将被置位。

表 19-52 CAN_IPEN_X 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | IntPndReg2 | | IntPndReg1 | |
| R-0h | | | | R-0h | | R-0h | |

表 19-53 CAN_IPEN_X 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | 保留。 |
| 3-2 | IntPndReg2 | R | 0h | 中断挂起寄存器 2 标志：<br>位 2 代表 CAN_IPEN_21 寄存器的字节 2。如果该字节中有一个或多个位置位，则位 2 将被置位。<br>位 3 代表 CAN_IPEN_21 寄存器的字节 3。如果该字节中有一个或多个位置位，则位 3 将被置位。<br>复位类型：SYSRSn |
| 1-0 | IntPndReg1 | R | 0h | 中断挂起寄存器 1 标志：<br>位 0 代表 CAN_IPEN_21 寄存器的字节 0。如果该字节中有一个或多个位置位，则位 0 将被置位。<br>位 1 代表 CAN_IPEN_21 寄存器的字节 1。如果该字节中有一个或多个位置位，则位 1 将被置位。<br>复位类型：SYSRSn |

## 19.15.2.18 CAN_IPEN_21 寄存器

偏移地址：B0h　复位值：00000000h

该寄存器保存各邮箱的 IntPnd 位。通过读出这些位，CPU 可以检查邮箱中挂起的中断。特定邮箱的 IntPnd 位可由 CPU 通过 IF1/IF2 接口寄存器组置位/复位，也可由消息处理器在一次接收或一次成功发送之后置位/复位。

表 19-54 CAN_IPEN_21 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| IntPnd | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IntPnd | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| IntPnd | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IntPnd | | | | | | | |
| R-0h | | | | | | | |

表 19-55 CAN_IPEN_21 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | IntPnd | R | 0h | 中断挂起位：该寄存器包含指示 32 个邮箱中每一个邮箱的中断挂起情况的位。<br>0 该邮箱不是中断源。<br>1 该邮箱是中断源。<br>注：位 0 对应邮箱 1，位 1 对应邮箱 2，位 2 对应邮箱 3，……，位 31 对应邮箱 32。<br>复位类型：SYSRSn |

## 19.15.2.19 CAN_MVAL_X 寄存器

偏移地址：C0h　复位值：00000000h

借助这些位，CPU 可以检测 CAN 消息有效 2_1 寄存器（CAN_MVAL_21）中是否有一个或多个位置位。该寄存器中的每一位代表一组八个邮箱。如果这些邮箱中至少有一个 MsgVal 位置位，则该寄存器中的对应位将被置位。

表 19-56 CAN_MVAL_X 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | MsgValReg2 | | MsgValReg1 | |
| R-0h | | | | R-0h | | R-0h | |

表 19-57 CAN_MVAL_X 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-4 | RESERVED | R | 0h | 保留。 |
| 3-2 | MsgValReg2 | R | 0h | 消息有效寄存器 2 标志：<br>位 2 代表 CAN_MVAL_21 寄存器的字节 2。如果该字节中有一个或多个位置位，则位 2 将被置位。<br>位 3 代表 CAN_MVAL_21 寄存器的字节 3。如果该字节中有一个或多个位置位，则位 3 将被置位。<br>复位类型：SYSRSn |
| 1-0 | MsgValReg1 | R | 0h | 消息有效寄存器 1 标志：<br>位 0 代表 CAN_MVAL_21 寄存器的字节 0。如果该字节中有一个或多个位置位，则位 0 将被置位。<br>位 1 代表 CAN_MVAL_21 寄存器的字节 1。如果该字节中有一个或多个位置位，则位 1 将被置位。<br>复位类型：SYSRSn |

## 19.15.2.20 CAN_MVAL_21 寄存器

偏移地址：C4h　复位值：00000000h

该寄存器保存所有邮箱的 MsgVal 位。通过读出 MsgVal 位，CPU 可以检查哪个邮箱有效。特定邮箱的 MsgVal 位可由 CPU 通过 IF1/2“消息接口”寄存器置位/复位。

表 19-58 CAN_MVAL_21 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MsgValReg | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MsgValReg | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MsgValReg | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MsgValReg | | | | | | | |
| R-0h | | | | | | | |

表 19-59 CAN_MVAL_21 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | MsgValReg | R | 0h | 消息有效位（针对所有消息对象）。<br>0 该消息对象被消息处理器忽略。<br>1 该消息对象已配置，将被消息处理器考虑。<br>注：位 0 对应邮箱 1，位 1 对应邮箱 2，位 2 对应邮箱 3，……，位 31 对应邮箱 32。<br>复位类型：SYSRSn |

## 19.15.2.21 CAN_IP_MUX21 寄存器

偏移地址：D8h　复位值：00000000h

IntMux 位为每个邮箱决定：当该邮箱的 IntPnd 位置位时，两条中断线（CANINT0 或 CANINT1）中的哪一条将被置为有效。两条中断线都可以通过置位或清除 CAN 控制寄存器中的 IE0 和 IE1 位来全局使能或禁止。这也会影响中断寄存器中的 INT0ID 或 INT1ID 标志。

表 19-60 CAN_IP_MUX21 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| IntMux | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IntMux | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| IntMux | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IntMux | | | | | | | |
| R/W-0h | | | | | | | |

表 19-61 CAN_IP_MUX21 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | IntMux | R/W | 0h | 中断多路复用位：<br>0 如果对应的 IntPnd 标志为 1，则 CANINT0 线有效。<br>1 如果对应的 IntPnd 标志为 1，则 CANINT1 线有效。<br>注：位 0 对应邮箱 32，位 1 对应邮箱 1，位 2 对应邮箱 2，……，位 31 对应邮箱 31。<br>复位类型：SYSRSn |

## 19.15.2.22 CAN_IF1CMD 寄存器

偏移地址：100h　复位值：00000001h

IF1/IF2 命令寄存器配置并启动 IF1/IF2 寄存器组与消息 RAM 之间的传输。可以配置消息对象的哪些部分应被传输。当 CPU 将消息编号写入 IF1/IF2 命令寄存器的位 [7:0] 时，传输即开始。在该写操作中，Busy 位被自动置为“1”，以指示传输正在进行。经过 4 到 14 个时钟周期后，接口寄存器与消息 RAM 之间的传输完成，Busy 位被清除。当消息传输与 CAN 消息发送、接收滤波或消息存储同时发生时，需要最多的周期数。

如果 CPU 连续写 IF1/IF2 两个命令寄存器（即第一次传输尚未完成时又请求第二次传输），则第二次传输将在第一次传输完成后开始。写该寄存器时应注意：(1)

不要向整个寄存器写 0；(2) 应以单次 32 位写操作写入，或先写高 16 位、再写低 16 位。

**注意：**Busy 位为 1 时，IF1/IF2 寄存器组受写保护。

**注意：**为支持调试，在调试/挂起模式下，IF1/IF2 命令寄存器的自动清除功能（通过读/写清除 DMAactive 标志）被禁用。

**注意：**如果向位 [7:0] 写入无效的消息编号，消息处理器可能转而访问某个已实现（有效）的消息对象。

表 19-62 CAN_IF1CMD 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIR | Mask | Arb | Control | ClrIntPnd | TXRQST | DATA_A | DATA_B |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Busy | DMAactive | 保留 | | | | | |
| R-0h | R/W-0h | R-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MSG_NUM | | | | | | | |
| R/W-1h | | | | | | | |

表 19-63 CAN_IF1CMD 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | 保留。 |
| 23 | DIR | R/W | 0h | 写/读。<br>0 方向 = 读：传输方向为从由消息编号（位 [7:0]）寻址的消息对象到 IF1/IF2 寄存器组。即把数据从邮箱传输到所选的 IF1/IF2 消息缓冲寄存器。<br>1 方向 = 写：传输方向为从 IF1/IF2 寄存器组到由消息编号（位 [7:0]）寻址的消息对象。即把数据从所选的 IF1/IF2 消息缓冲寄存器传输到邮箱。IF1/IF2 命令掩码寄存器的其他位根据传输方向具有不同的功能。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 22 | Mask | R/W | 0h | 访问掩码位。<br>0 掩码位不会被更改。<br>1（方向 = 读）：掩码位（标识符掩码 + MDir + MXtd）将从由消息编号（位 [7:0]）寻址的消息对象传输到 IF1/IF2 寄存器组。<br>1（方向 = 写）：掩码位（标识符掩码 + MDir + MXtd）将从 IF1/IF2 寄存器组传输到由消息编号（位 [7:0]）寻址的消息对象。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 21 | Arb | R/W | 0h | 访问仲裁位。<br>0 仲裁位不会被更改。<br>1（方向 = 读）：仲裁位（标识符 + Dir + Xtd + MsgVal）将从由消息编号（位 [7:0]）寻址的消息对象传输到对应的 IF1/IF2 寄存器组。<br>1（方向 = 写）：仲裁位（标识符 + Dir + Xtd + MsgVal）将从 IF1/IF2 寄存器组传输到由消息编号（位 [7:0]）寻址的消息对象。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 20 | Control | R/W | 0h | 访问控制位。<br>如果该寄存器中的 TxRqst/NewDat 位（位 [18]）置位，则 IF1 消息控制寄存器中的 TxRqst/NewDat 位将被忽略。<br>0 控制位不会被更改。<br>1（方向 = 读）：消息控制位将从由消息编号（位 [7:0]）寻址的消息对象传输到 IF1 寄存器组。<br>1（方向 = 写）：消息控制位将从 IF1 寄存器组传输到由消息编号（位 [7:0]）寻址的消息对象。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 19 | ClrIntPnd | R/W | 0h | 清除中断挂起位。<br>0 IntPnd 位不会被更改。<br>1（方向 = 读）：清除消息对象中的 IntPnd 位。<br>1（方向 = 写）：该位被忽略。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 18 | TXRQST | R/W | 0h | 访问发送请求（TxRqst）/新数据（NewDat）位。<br>0（方向 = 读）：NewDat 位不会被更改。<br>0（方向 = 写）：TxRqst/NewDat 位将根据 Control 位来处理。<br>1（方向 = 读）：清除消息对象中的 NewDat |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 位。<br>1（方向 = 写）：置位消息对象中的 TxRqst/NewDat。<br>注：如果通过置位该寄存器中的 TxRqst/NewDat 来请求一次 CAN 发送，则消息对象中的 TxRqst/NewDat 位将被置为 1，与 IF1/IF2 消息控制寄存器中的值无关。<br>注：对消息对象的读访问可以与控制位 IntPnd 和 NewDat 的复位组合在一起。传输到 IF1/IF2 消息控制寄存器的这些位的值始终反映复位之前的状态。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 17 | DATA_A | R/W | 0h | 访问数据字节 0-3。<br>0 数据字节 0-3 不会被更改。<br>1（方向 = 读）：数据字节 0-3 将从由消息编号（位 [7:0]）寻址的消息对象传输到对应的 IF1/IF2 寄存器组。<br>1（方向 = 写）：数据字节 0-3 将从 IF1/IF2 寄存器组传输到由消息编号（位 [7:0]）寻址的消息对象。<br>注：消息传输的持续时间与要传输的字节数无关。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 16 | DATA_B | R/W | 0h | 访问数据字节 4-7。<br>0 数据字节 4-7 不会被更改。<br>1（方向 = 读）：数据字节 4-7 将从由消息编号（位 [7:0]）寻址的消息对象传输到对应的 IF1/IF2 寄存器组。<br>1（方向 = 写）：数据字节 4-7 将从 IF1/IF2 寄存器组传输到由消息编号（位 [7:0]）寻址的消息对象。<br>注：消息传输的持续时间与要传输的字节数无关。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 15 | Busy | R | 0h | Busy 标志。<br>0 IF1/IF2 寄存器组与消息 RAM 之间没有正在进行的传输。<br>1 IF1/IF2 寄存器组与消息 RAM 之间正在进行传输。<br>在消息编号被写入位 [7:0] 之后，该位被置为 1。IF1/IF2 寄存器组将受写保护。读写操作完成后该 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 位被清除。<br>复位类型：SYSRSn |
| 14 | DMAactive | R/W | 0h | 由 IF1 更新引起的 DMA 触发状态。<br>0 没有活动的 IF1 DMA 请求。<br>1 在 IF1 与消息 RAM 之间完成一次传输后请求 DMA。DMA 请求保持有效，直到对 IF1 寄存器之一的首次读或写；当 DMAactive 为 1 时，对消息编号（位 [7:0]）的写是一个例外。<br>注：由于 DMAactive 位的自动复位特性，该位必须为每个后续 DMA 周期单独置位。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 13-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | MSG_NUM | R/W | 1h | 消息 RAM 中用于数据传输的消息对象编号。<br>0x00 无效的消息编号<br>0x01-0x20 有效的消息编号<br>0x21-0xFF 无效的消息编号<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |

## 19.15.2.23 CAN_IF1MSK 寄存器

偏移地址：104h　复位值：FFFFFFFFh

IF1/IF2 掩码寄存器的各位镜像消息对象的掩码位。

**注意：**IF1/IF2 命令寄存器的 Busy 位为 1 时，IF1/IF2 寄存器组受写保护。

表 19-64 CAN_IF1MSK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MXtd | MDir | 保留 | Msk | | | | |
| R/W-1h | R/W-1h | R-1h | R/W-1FFFFFFFh | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Msk | | | | | | | |
| R/W-1FFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Msk | | | | | | | |
| R/W-1FFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Msk | | | | | | | |
| R/W-1FFFFFFFh | | | | | | | |

表 19-65 CAN_IF1MSK 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | MXtd | R/W | 1h | 掩码扩展标识符。<br>0 扩展标识符位（IDE）对接收滤波没有影响。<br>1 扩展标识符位（IDE）用于接收滤波。<br>当消息对象使用 11 位（“标准”）标识符时，接收数据帧的标识符被写入位 ID[28:18]。对于接收滤波，只考虑这些位连同掩码位 Msk[28:18]。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 30 | MDir | R/W | 1h | 掩码消息方向。<br>0 消息方向位（Dir）对接收滤波没有影响。<br>1 消息方向位（Dir）用于接收滤波。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 29 | RESERVED | R | 1h | 保留。 |
| 28-0 | Msk | R/W | 1FFFFFFFh | 标识符掩码。<br>0 消息对象标识符中的对应位不用于接收滤波（无关位）。<br>1 消息对象标识符中的对应位用于接收滤波。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |

## 19.15.2.24 CAN_IF1ARB 寄存器

偏移地址：108h　复位值：00000000h

IF1/IF2 仲裁寄存器的各位镜像消息对象的仲裁位。仲裁位 ID[28:0]、Xtd 和 Dir 用于定义发出消息的标识符和类型，并（连同掩码位 Msk[28:0]、MXtd 和 MDir）用于对接收消息进行接收滤波。接收到的消息被存储到标识符匹配且 Direction = 接收（数据帧）或 Direction = 发送（远程帧）的有效消息对象中。扩展帧只能存储在 Xtd = 1 的消息对象中，标准帧则存储在 Xtd = 0 的消息对象中。如果一条接收到的消息（数据帧或远程帧）匹配多于一个有效消息对象，它将被存储到消息编号最小的那个中。

表 19-66 CAN_IF1ARB 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MsgVal | Xtd | Dir | ID | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ID | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ID | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ID | | | | | | | |
| R/W-0h | | | | | | | |

表 19-67 CAN_IF1ARB 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | MsgVal | R/W | 0h | 消息有效。<br>0 邮箱被禁用。（该消息对象被消息处理器忽略）。<br>1 邮箱被使能。（该消息对象将由消息处理器使用）。<br>CPU 应在初始化期间、在复位 CAN 控制寄存器中的 Init 位之前，复位所有未使用消息对象的 MsgVal 位。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 30 | Xtd | R/W | 0h | 扩展标识符。<br>0 该消息对象使用 11 位（“标准”）标识符。<br>1 该消息对象使用 29 位（“扩展”）标识符。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 29 | Dir | R/W | 0h | 消息方向。<br>0 方向 = 接收：在 TxRqst 时，发送一个带有该消息对象标识符的远程帧。在接收到标识符匹配的数据帧时，该帧被存储到该消息对象中。<br>1 方向 = 发送：在 TxRqst 时，相应的消息对象作为数据帧被发送。在接收到标识符匹配的远程帧时，该消息对象的 TxRqst 位被置位（如果 RmtEn = 1）。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 28-0 | ID | R/W | 0h | 消息标识符。<br>ID[28:0] 29 位标识符（“扩展帧”）<br>ID[28:18] 11 位标识符（“标准帧”）<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |

### 19.15.2.25 CAN_IF1MCTL 寄存器

偏移地址：10Ch　复位值：00000000h

IF1/IF2 消息控制寄存器的各位镜像消息对象的消息控制位。该寄存器具有与中断、接收掩码、远程帧和 FIFO 选项有关的控制/状态位。

表 19-68 CAN_IF1MCTL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| NewDat | MsgLst | IntPnd | UMask | TxIE | RxIE | RmtEn | TxRqst |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EoB | 保留 | | | DLC | | | |
| R/W-0h | R-0h | | | R/W-0h | | | |

表 19-69 CAN_IF1MCTL 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15 | NewDat | R/W | 0h | 新数据。<br>0 自上次该标志被 CPU 清除以来，消息处理器未向该消息对象的数据部分写入新数据。<br>1 消息处理器或 CPU 已向该消息对象的数据部分写入新数据。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 14 | MsgLst | R/W | 0h | 消息丢失（仅对方向 = 接收的消息对象有效）。<br>0 自上次该位被 CPU 复位以来没有消息丢失。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 1 当 NewDat 仍处于置位状态时，消息处理器向该对象存储了一条新消息，因此先前的消息已被覆盖。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 13 | IntPnd | R/W | 0h | 中断挂起。<br>0 该消息对象不是中断源。<br>1 该消息对象是中断源。如果没有其他更高优先级的中断源，中断寄存器中的中断标识符将指向该消息对象。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 12 | UMask | R/W | 0h | 使用接收掩码。<br>0 忽略掩码。<br>1 使用掩码（Msk[28:0]、MXtd 和 MDir）进行接收滤波。<br>如果 UMask 位置为 1，则必须在消息对象初始化期间、在 MsgVal 置为 1 之前，对消息对象的掩码位进行编程。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 11 | TxIE | R/W | 0h | 发送中断使能。<br>0 帧成功发送后不会触发 IntPnd。<br>1 帧成功发送后将触发 IntPnd。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 10 | RxIE | R/W | 0h | 接收中断使能。<br>0 帧成功接收后不会触发 IntPnd。<br>1 帧成功接收后将触发 IntPnd。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 9 | RmtEn | R/W | 0h | 远程使能。<br>0 在接收到远程帧时，TxRqst 不变。<br>1 在接收到远程帧时，TxRqst 被置位。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 8 | TxRqst | R/W | 0h | 发送请求。<br>0 该消息对象未在等待发送。<br>1 该消息对象的发送已被请求且尚未完成。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 7 | EoB | R/W | 0h | 块结束。<br>0 该消息对象是 FIFO 缓冲区块的一部分，且不是该 FIFO 缓冲区块的最后一个消息对象。<br>1 该消息对象是单个消息对象，或者是 FIFO 缓冲区块中的最后一个消息对象。<br>注：该位用于将多个消息对象串联起来以构建 FIFO 缓冲。对于单个消息对象（不属于 FIFO 缓冲），该位必须始终置为 1。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 6-4 | RESERVED | R | 0h | 保留。 |
| 3-0 | DLC | R/W | 0h | 数据长度代码。<br>0-8 数据帧具有 0-8 个数据字节。<br>9-15 数据帧具有 8 个数据字节。<br>注：消息对象的数据长度代码必须与其他节点上具有相同标识符的所有对应对象中定义的一致。当消息处理器存储一个数据帧时，它会把 DLC 写为接收消息所给出的值。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |

### 19.15.2.26 CAN_IF1DATA 寄存器

偏移地址：110h　复位值：00000000h

该寄存器提供对 CAN 报文数据字节的访问窗口。CAN 报文的数据字节按以下顺序存储在 IF1/IF2 寄存器中。在 CAN 数据帧中，Data 0 是第一个字节，Data 7 是最后一个被发送或接收的字节。在 CAN 的串行位流中，每个字节的 MSB 将首先被发送。该寄存器中的所有位都受 Busy 位写保护。

表 19-70 CAN_IF1DATA 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data_3 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data_2 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data_1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data_0 | | | | | | | |
| R/W-0h | | | | | | | |

表 19-71 CAN_IF1DATA 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | Data_3 | R/W | 0h | 数据字节 3。<br>复位类型：SYSRSn |
| 23-16 | Data_2 | R/W | 0h | 数据字节 2。<br>复位类型：SYSRSn |
| 15-8 | Data_1 | R/W | 0h | 数据字节 1。<br>复位类型：SYSRSn |
| 7-0 | Data_0 | R/W | 0h | 数据字节 0。<br>复位类型：SYSRSn |

## 19.15.2.27 CAN_IF1DATB 寄存器

偏移地址：114h　复位值：00000000h

该寄存器提供对 CAN 报文数据字节的访问窗口。CAN 报文的数据字节按以下顺序存储在 IF1/IF2 寄存器中。在 CAN 数据帧中，Data 0 是第一个字节，Data 7 是最后一个被发送或接收的字节。在 CAN 的串行位流中，每个字节的 MSB 将首先被发送。该寄存器中的所有位都受 Busy 位写保护。

表 19-72 CAN_IF1DATB 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data_7 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data_6 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data_5 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data_4 | | | | | | | |
| R/W-0h | | | | | | | |

表 19-73 CAN_IF1DATB 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | Data_7 | R/W | 0h | 数据字节 7。<br>复位类型：SYSRSn |
| 23-16 | Data_6 | R/W | 0h | 数据字节 6。<br>复位类型：SYSRSn |
| 15-8 | Data_5 | R/W | 0h | 数据字节 5。<br>复位类型：SYSRSn |
| 7-0 | Data_4 | R/W | 0h | 数据字节 4。<br>复位类型：SYSRSn |

## 19.15.2.28 CAN_IF2CMD 寄存器

偏移地址：120h　复位值：00000001h

IF1/IF2 命令寄存器配置并启动 IF1/IF2 寄存器组与消息 RAM 之间的传输。可以配置消息对象的哪些部分应被传输。当 CPU 将消息编号写入 IF1/IF2 命令寄存器的位 [7:0] 时，传输即开始。在该写操作中，Busy 位被自动置为“1”，以指示传输正在进行。经过 4 到 14 个时钟周期后，接口寄存器与消息 RAM 之间的传输完成，Busy 位被清除。当消息传输与 CAN 消息发送、接收滤波或消息存储同时发生时，需要最多的周期数。

如果 CPU 连续写 IF1/IF2 两个命令寄存器（即第一次传输尚未完成时又请求第二次传输），则第二次传输将在第一次传输完成后开始。写该寄存器时应注意：(1) 不要向整个寄存器写 0；(2) 应以单次 32 位写操作写入，或先写高 16 位、再写低 16 位。

**注意：**Busy 位为 1 时，IF1/IF2 寄存器组受写保护。

**注意：**为支持调试，在调试/挂起模式下，IF1/IF2 命令寄存器的自动清除功能（通过读/写清除 DMAactive 标志）被禁用。

**注意：**如果向位 [7:0] 写入无效的消息编号，消息处理器可能转而访问某个已实现（有效）的消息对象。

表 19-74 CAN_IF2CMD 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DIR | Mask | Arb | Control | ClrIntPnd | TxRqst | DATA_A | DATA_B |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Busy | DMAactive | 保留 | | | | | |
| R-0h | R/W-0h | R-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MSG_NUM | | | | | | | |
| R/W-1h | | | | | | | |

表 19-75 CAN_IF2CMD 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | 保留。 |
| 23 | DIR | R/W | 0h | 写/读。<br>0 方向 = 读：传输方向为从由消息编号（位 [7:0]）寻址的消息对象到 IF1/IF2 寄存器组。即把数据从邮箱传输到所选的 IF1/IF2 消息缓冲寄存器。<br>1 方向 = 写：传输方向为从 IF1/IF2 寄存器组到由消息编号（位 [7:0]）寻址的消息对象。即把数据从所选的 IF1/IF2 消息缓冲寄存器传输到邮箱。IF1/IF2 命令掩码寄存器的其他位根据传输方向具 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 有不同的功能。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 22 | Mask | R/W | 0h | 访问掩码位。<br>0 掩码位不会被更改。<br>1（方向 = 读）：掩码位（标识符掩码 + MDir + MXtd）将从由消息编号（位 [7:0]）寻址的消息对象传输到 IF1/IF2 寄存器组。<br>1（方向 = 写）：掩码位（标识符掩码 + MDir + MXtd）将从 IF1/IF2 寄存器组传输到由消息编号（位 [7:0]）寻址的消息对象。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 21 | Arb | R/W | 0h | 访问仲裁位。<br>0 仲裁位不会被更改。<br>1（方向 = 读）：仲裁位（标识符 + Dir + Xtd + MsgVal）将从由消息编号（位 [7:0]）寻址的消息对象传输到对应的 IF1/IF2 寄存器组。<br>1（方向 = 写）：仲裁位（标识符 + Dir + Xtd + MsgVal）将从 IF1/IF2 寄存器组传输到由消息编号（位 [7:0]）寻址的消息对象。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 20 | Control | R/W | 0h | 访问控制位。<br>如果该寄存器中的 TxRqst/NewDat 位（位 [18]）置位，则 IF1 消息控制寄存器中的 TxRqst/NewDat 位将被忽略。<br>0 控制位不会被更改。<br>1（方向 = 读）：消息控制位将从由消息编号（位 [7:0]）寻址的消息对象传输到 IF1 寄存器组。<br>1（方向 = 写）：消息控制位将从 IF1 寄存器组传输到由消息编号（位 [7:0]）寻址的消息对象。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 19 | ClrIntPnd | R/W | 0h | 清除中断挂起位。<br>0 IntPnd 位不会被更改。<br>1（方向 = 读）：清除消息对象中的 IntPnd 位。<br>1（方向 = 写）：该位被忽略。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 18 | TxRqst | R/W | 0h | 访问发送请求（TxRqst）/新数据（NewDat）位。<br>0（方向 = 读）：NewDat 位不会被更改。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 0（方向 = 写）：TxRqst/NewDat 位将根据 Control 位来处理。<br>1（方向 = 读）：清除消息对象中的 NewDat 位。<br>1（方向 = 写）：置位消息对象中的 TxRqst/NewDat。<br>注：如果通过置位该寄存器中的 TxRqst/NewDat 来请求一次 CAN 发送，则消息对象中的 TxRqst/NewDat 位将被置为 1，与 IF1/IF2 消息控制寄存器中的值无关。<br>注：对消息对象的读访问可以与控制位 IntPnd 和 NewDat 的复位组合在一起。传输到 IF1/IF2 消息控制寄存器的这些位的值始终反映复位之前的状态。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 17 | DATA_A | R/W | 0h | 访问数据字节 0-3。<br>0 数据字节 0-3 不会被更改。<br>1（方向 = 读）：数据字节 0-3 将从由消息编号（位 [7:0]）寻址的消息对象传输到对应的 IF1/IF2 寄存器组。<br>1（方向 = 写）：数据字节 0-3 将从 IF1/IF2 寄存器组传输到由消息编号（位 [7:0]）寻址的消息对象。<br>注：消息传输的持续时间与要传输的字节数无关。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 16 | DATA_B | R/W | 0h | 访问数据字节 4-7。<br>0 数据字节 4-7 不会被更改。<br>1（方向 = 读）：数据字节 4-7 将从由消息编号（位 [7:0]）寻址的消息对象传输到对应的 IF1/IF2 寄存器组。<br>1（方向 = 写）：数据字节 4-7 将从 IF1/IF2 寄存器组传输到由消息编号（位 [7:0]）寻址的消息对象。<br>注：消息传输的持续时间与要传输的字节数无关。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 15 | Busy | R | 0h | Busy 标志。<br>0 IF1/IF2 寄存器组与消息 RAM 之间没有正在进行的传输。<br>1 IF1/IF2 寄存器组与消息 RAM 之间正在进行传输。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 在消息编号被写入位 [7:0] 之后，该位被置为 1。IF1/IF2 寄存器组将受写保护。读写操作完成后该位被清除。<br>复位类型：SYSRSn |
| 14 | DMAactive | R/W | 0h | 由 IF2 更新引起的 DMA 触发状态。<br>0 没有活动的 IF2 DMA 请求。<br>1 在 IF2 与消息 RAM 之间完成一次传输后请求 DMA。DMA 请求保持有效，直到对 IF2 寄存器之一的首次读或写；当 DMAactive 为 1 时，对消息编号（位 [7:0]）的写是一个例外。<br>注：由于 DMAactive 位的自动复位特性，该位必须为每个后续 DMA 周期单独置位。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 13-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | MSG_NUM | R/W | 1h | 消息 RAM 中用于数据传输的消息对象编号。<br>0x00 无效的消息编号<br>0x01-0x20 有效的消息编号<br>0x21-0xFF 无效的消息编号<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |

## 19.15.2.29 CAN_IF2MSK 寄存器

偏移地址：124h　复位值：FFFFFFFFh

IF1/IF2 掩码寄存器的各位镜像消息对象的掩码位。

**注意：**IF1/IF2 命令寄存器的 Busy 位为 1 时，IF1/IF2 寄存器组受写保护。

表 19-76 CAN_IF2MSK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MXtd | MDir | 保留 | Msk | | | | |
| R/W-1h | R/W-1h | R-1h | R/W-1FFFFFFFh | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Msk | | | | | | | |
| R/W-1FFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Msk | | | | | | | |
| R/W-1FFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Msk | | | | | | | |
| R/W-1FFFFFFFh | | | | | | | |

表 19-77 CAN_IF2MSK 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | MXtd | R/W | 1h | 掩码扩展标识符。<br>0 扩展标识符位（IDE）对接收滤波没有影响。<br>1 扩展标识符位（IDE）用于接收滤波。<br>当消息对象使用 11 位（“标准”）标识符时，接收数据帧的标识符被写入位 ID[28:18]。对于接收滤波，只考虑这些位连同掩码位 Msk[28:18]。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 30 | MDir | R/W | 1h | 掩码消息方向。<br>0 消息方向位（Dir）对接收滤波没有影响。<br>1 消息方向位（Dir）用于接收滤波。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 29 | RESERVED | R | 1h | 保留。 |
| 28-0 | Msk | R/W | 1FFFFFFFh | 标识符掩码。<br>0 消息对象标识符中的对应位不用于接收滤波（无关位）。<br>1 消息对象标识符中的对应位用于接收滤波。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |

## 19.15.2.30 CAN_IF2ARB 寄存器

偏移地址：128h　复位值：00000000h

IF1/IF2 仲裁寄存器的各位镜像消息对象的仲裁位。仲裁位 ID[28:0]、Xtd 和 Dir 用于定义发出消息的标识符和类型，并（连同掩码位 Msk[28:0]、MXtd 和 MDir）用于对接收消息进行接收滤波。接收到的消息被存储到标识符匹配且 Direction = 接收（数据帧）或 Direction = 发送（远程帧）的有效消息对象中。扩展帧只能存储在 Xtd = 1 的消息对象中，标准帧则存储在 Xtd = 0 的消息对象中。如果一条接收到的消息（数据帧或远程帧）匹配多于一个有效消息对象，它将被存储到消息编号最小的那个中。

表 19-78 CAN_IF2ARB 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MsgVal | Xtd | Dir | ID | | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ID | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ID | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ID | | | | | | | |
| R/W-0h | | | | | | | |

表 19-79 CAN_IF2ARB 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | MsgVal | R/W | 0h | 消息有效。<br>0 邮箱被禁用。（该消息对象被消息处理器忽略）。<br>1 邮箱被使能。（该消息对象将由消息处理器使用）。<br>CPU 应在初始化期间、在复位 CAN 控制寄存器中的 Init 位之前，复位所有未使用消息对象的 MsgVal 位。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 30 | Xtd | R/W | 0h | 扩展标识符。<br>0 该消息对象使用 11 位（“标准”）标识符。<br>1 该消息对象使用 29 位（“扩展”）标识符。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 29 | Dir | R/W | 0h | 消息方向。<br>0 方向 = 接收：在 TxRqst 时，发送一个带有该消息对象标识符的远程帧。在接收到标识符匹配的数据帧时，该帧被存储到该消息对象中。<br>1 方向 = 发送：在 TxRqst 时，相应的消息对象作为数据帧被发送。在接收到标识符匹配的远程帧时，该消息对象的 TxRqst 位被置位（如果 RmtEn = 1）。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 28-0 | ID | R/W | 0h | 消息标识符。<br>ID[28:0] 29 位标识符（“扩展帧”）<br>ID[28:18] 11 位标识符（“标准帧”）<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |

## 19.15.2.31 CAN_IF2MCTL 寄存器

偏移地址：12Ch　复位值：00000000h

IF1/IF2 消息控制寄存器的各位镜像消息对象的消息控制位。该寄存器具有与中断、接收掩码、远程帧和 FIFO 选项有关的控制/状态位。

表 19-80 CAN_IF2MCTL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| NewDat | MsgLst | IntPnd | UMask | TxIE | RxIE | RmtEn | TxRqst |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EoB | 保留 | | | DLC | | | |
| R/W-0h | R-0h | | | R/W-0h | | | |

表 19-81 CAN_IF2MCTL 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15 | NewDat | R/W | 0h | 新数据。<br>0 自上次该标志被 CPU 清除以来，消息处理器未向该消息对象的数据部分写入新数据。<br>1 消息处理器或 CPU 已向该消息对象的数据部分写入新数据。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 14 | MsgLst | R/W | 0h | 消息丢失（仅对方向 = 接收的消息对象有效）。<br>0 自上次该位被 CPU 复位以来没有消息丢失。<br>1 当 NewDat 仍处于置位状态时，消息处理器向该对象存储了一条新消息，因此先前的消息已被覆盖。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 13 | IntPnd | R/W | 0h | 中断挂起。<br>0 该消息对象不是中断源。<br>1 该消息对象是中断源。如果没有其他更高优先级的中断源，中断寄存器中的中断标识符将指向该消息对象。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 12 | UMask | R/W | 0h | 使用接收掩码。<br>0 忽略掩码。<br>1 使用掩码（Msk[28:0]、MXtd 和 MDir）进行接收滤波。<br>如果 UMask 位置为 1，则必须在消息对象初始化期间、在 MsgVal 置为 1 之前，对消息对象的掩码位进行编程。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 11 | TxIE | R/W | 0h | 发送中断使能。<br>0 帧成功发送后不会触发 IntPnd。<br>1 帧成功发送后将触发 IntPnd。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 10 | RxIE | R/W | 0h | 接收中断使能。<br>0 帧成功接收后不会触发 IntPnd。<br>1 帧成功接收后将触发 IntPnd。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 9 | RmtEn | R/W | 0h | 远程使能。<br>0 在接收到远程帧时，TxRqst 不变。<br>1 在接收到远程帧时，TxRqst 被置位。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 8 | TxRqst | R/W | 0h | 发送请求。<br>0 该消息对象未在等待发送。<br>1 该消息对象的发送已被请求且尚未完成。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 7 | EoB | R/W | 0h | 块结束。<br>0 该消息对象是 FIFO 缓冲区块的一部分，且不是该 FIFO 缓冲区块的最后一个消息对象。<br>1 该消息对象是单个消息对象，或者是 FIFO 缓冲区块中的最后一个消息对象。<br>注：该位用于将多个消息对象串联起来以构建 FIFO 缓冲。对于单个消息对象（不属于 FIFO 缓冲），该位必须始终置为 1。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |
| 6-4 | RESERVED | R | 0h | 保留。 |
| 3-0 | DLC | R/W | 0h | 数据长度代码。<br>0-8 数据帧具有 0-8 个数据字节。<br>9-15 数据帧具有 8 个数据字节。<br>注：消息对象的数据长度代码必须与其他节点上具有相同标识符的所有对应对象中定义的一致。当消息处理器存储一个数据帧时，它会把 DLC 写为接收消息所给出的值。<br>注：该位受 Busy 位写保护。<br>复位类型：SYSRSn |

### 19.15.2.32 CAN_IF2DATA 寄存器

偏移地址：130h　复位值：00000000h

该寄存器提供对 CAN 报文数据字节的访问窗口。CAN 报文的数据字节按以下顺序存储在 IF1/IF2 寄存器中。在 CAN 数据帧中，Data 0 是第一个字节，Data 7

是最后一个被发送或接收的字节。在 CAN 的串行位流中，每个字节的 MSB 将首先被发送。该寄存器中的所有位都受 Busy 位写保护。

表 19-82 CAN_IF2DATA 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data_3 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data_2 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data_1 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data_0 | | | | | | | |
| R/W-0h | | | | | | | |

表 19-83 CAN_IF2DATA 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | Data_3 | R/W | 0h | 数据字节 3。<br>复位类型：SYSRSn |
| 23-16 | Data_2 | R/W | 0h | 数据字节 2。<br>复位类型：SYSRSn |
| 15-8 | Data_1 | R/W | 0h | 数据字节 1。<br>复位类型：SYSRSn |
| 7-0 | Data_0 | R/W | 0h | 数据字节 0。<br>复位类型：SYSRSn |

## 19.15.2.33 CAN_IF2DATB 寄存器

偏移地址：134h　复位值：00000000h

该寄存器提供对 CAN 报文数据字节的访问窗口。CAN 报文的数据字节按以下顺序存储在 IF1/IF2 寄存器中。在 CAN 数据帧中，Data 0 是第一个字节，Data 7 是最后一个被发送或接收的字节。在 CAN 的串行位流中，每个字节的 MSB 将首先被发送。该寄存器中的所有位都受 Busy 位写保护。

表 19-84 CAN_IF2DATB 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data_7 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data_6 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data_5 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data_4 | | | | | | | |
| R/W-0h | | | | | | | |

表 19-85 CAN_IF2DATB 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | Data_7 | R/W | 0h | 数据字节 7。<br>复位类型：SYSRSn |
| 23-16 | Data_6 | R/W | 0h | 数据字节 6。<br>复位类型：SYSRSn |
| 15-8 | Data_5 | R/W | 0h | 数据字节 5。<br>复位类型：SYSRSn |
| 7-0 | Data_4 | R/W | 0h | 数据字节 4。<br>复位类型：SYSRSn |

### 19.15.2.34 CAN_IF3OBS 寄存器

偏移地址：140h　复位值：00000000h

IF3 寄存器组可以自动更新为接收到的消息对象，无需由 CPU 启动从消息 RAM 的传输。

IF3 观察寄存器的观测标志（位 [4:0]）用于确定必须读取 IF3 接口寄存器组的哪些数据段，才能完成一次 DMA 读周期。所有标记的数据段都读取完成后，CAN 才允许用新数据更新 IF3 接口寄存器组。

支持任意顺序的单字节或半字访问。使用字节或半字访问时，只有某数据段的所有字节都已读取，该段才被标记为完成。

**注意：**如果使用了 IF3 更新使能且未设置任何观测标志，则相应的消息对象会被复制到 IF3，而不会激活 DMA 请求线，也不会等待 DMA 读访问。

对该寄存器的写访问会通过复位 DMA 线中止挂起的 DMA 周期，并使能 IF3 接口寄存器组的新数据更新。为避免数据不一致，在重新配置 IF3 观察寄存器之前应禁用 DMA 控制器。当前读周期的状态可通过状态标志（位 [12:8]）观察。

因此，应用程序可利用观测状态位和 IF3Upd 位，在轮询或中断模式下获得 IF3 新内容的通知。

表 19-86 CAN_IF3OBS 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| IF3Upd | 保留 | | IF3SDB | IF3SDA | IF3SC | IF3SA | IF3SM |
| R-0h | R-0h | | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | Data_B | Data_A | Ctrl | Arb | Mask |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 19-87 CAN_IF3OBS 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15 | IF3Upd | R | 0h | IF3 更新数据。<br>0 自上次 IF3 读取以来没有加载新数据。<br>1 自上次 IF3 读取以来已加载新数据。<br>复位类型：SYSRSn |
| 14-13 | RESERVED | R | 0h | 保留。 |
| 12 | IF3SDB | R | 0h | IF3 数据 B 读访问状态。<br>0 所有 Data B 字节均已读出，或未被标记为要读取。<br>1 Data B 段仍有数据待读出。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 11 | IF3SDA | R | 0h | IF3 数据 A 读访问状态。<br>0 所有 Data A 字节均已读出，或未被标记为要读取。<br>1 Data A 段仍有数据待读出。<br>复位类型：SYSRSn |
| 10 | IF3SC | R | 0h | IF3 控制位读访问状态。<br>0 所有 Control 段字节均已读出，或未被标记为要读取。<br>1 Control 段仍有数据待读出。<br>复位类型：SYSRSn |
| 9 | IF3SA | R | 0h | IF3 仲裁数据读访问状态。<br>0 所有 Arbitration 数据字节均已读出，或未被标记为要读取。<br>1 Arbitration 段仍有数据待读出。<br>复位类型：SYSRSn |
| 8 | IF3SM | R | 0h | IF3 掩码数据读访问状态。<br>0 所有 Mask 数据字节均已读出，或未被标记为要读取。<br>1 Mask 段仍有数据待读出。<br>复位类型：SYSRSn |
| 7-5 | RESERVED | R | 0h | 保留。 |
| 4 | Data_B | R/W | 0h | Data B 读观测。<br>0 不读取 Data B 段。<br>1 必须读取 Data B 段以使能下一次 IF3 更新。<br>复位类型：SYSRSn |
| 3 | Data_A | R/W | 0h | Data A 读观测。<br>0 不读取 Data A 段。<br>1 必须读取 Data A 段以使能下一次 IF3 更新。<br>复位类型：SYSRSn |
| 2 | Ctrl | R/W | 0h | Ctrl 读观测。<br>0 不读取 Ctrl 段。<br>1 必须读取 Ctrl 段以使能下一次 IF3 更新。<br>复位类型：SYSRSn |
| 1 | Arb | R/W | 0h | 仲裁数据读观测。<br>0 不读取仲裁数据。<br>1 必须读取仲裁数据以使能下一次 IF3 更新。<br>复位类型：SYSRSn |
| 0 | Mask | R/W | 0h | 掩码数据读观测。<br>0 不读取掩码数据。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 1 必须读取掩码数据以使能下一次 IF3 更新。<br>复位类型：SYSRSn |

## 19.15.2.35 CAN_IF3MSK 寄存器

偏移地址：144h 复位值：FFFFFFFFh

该寄存器提供对所选邮箱接收掩码的访问窗口。

表 19-88 CAN_IF3MSK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MXtd | MDir | 保留 | Msk | | | | |
| R-1h | R-1h | R-1h | R-1FFFFFFFh | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Msk | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Msk | | | | | | | |
| R-1FFFFFFFh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Msk | | | | | | | |
| R-1FFFFFFFh | | | | | | | |

表 19-89 CAN_IF3MSK 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | MXtd | R | 1h | 掩码扩展标识符。<br>0 扩展标识符位（IDE）对接收滤波没有影响。<br>1 扩展标识符位（IDE）用于接收滤波。<br>注：当消息对象使用 11 位（“标准”）标识符时，接收数据帧的标识符被写入位 ID[28:18]。对于接收滤波，只考虑这些位连同掩码位 Msk[28:18]。<br>复位类型：SYSRSn |
| 30 | MDir | R | 1h | 掩码消息方向。<br>0 消息方向位（Dir）对接收滤波没有影响。<br>1 消息方向位（Dir）用于接收滤波。<br>复位类型：SYSRSn |
| 29 | RESERVED | R | 1h | 保留。 |
| 28-0 | Msk | R | 1FFFFFFFh | 标识符掩码。<br>0 消息对象标识符中的对应位不用于接收滤波（无关位）。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 1 消息对象标识符中的对应位用于接收滤波。<br>复位类型：SYSRSn |

### 19.15.2.36 CAN_IF3ARB 寄存器

偏移地址：148h　复位值：00000000h

IF3 仲裁寄存器的各位镜像消息对象的仲裁位。

表 19-90 CAN_IF3ARB 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MsgVal | Xtd | Dir | ID | | | | |
| R-0h | R-0h | R-0h | R-0h | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ID | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ID | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ID | | | | | | | |
| R-0h | | | | | | | |

表 19-91 CAN_IF3ARB 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | MsgVal | R | 0h | 消息有效。<br>0 该消息对象被消息处理器忽略。<br>1 该消息对象将由消息处理器使用。<br>CPU 应在初始化期间、在复位 CAN 控制寄存器中的 Init 位之前，复位所有未使用消息对象的 MsgVal 位。<br>复位类型：SYSRSn |
| 30 | Xtd | R | 0h | 扩展标识符。<br>0 该消息对象使用 11 位（“标准”）标识符。<br>1 该消息对象使用 29 位（“扩展”）标识符。<br>复位类型：SYSRSn |
| 29 | Dir | R | 0h | 消息方向。<br>0 方向 = 接收：在 TxRqst 时，发送一个带有该消息对象标识符的远程帧。在接收到标识符匹配的数据帧时，该消息被存储到该消息对象中。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 1 方向 = 发送：在 TxRqst 时，相应的消息对象作为数据帧被发送。在接收到标识符匹配的远程帧时，该消息对象的 TxRqst 位被置位（如果 RmtEn = 1）。<br>复位类型：SYSRSn |
| 28-0 | ID | R | 0h | 消息标识符。<br>ID[28:0] 29 位标识符（“扩展帧”）<br>ID[28:18] 11 位标识符（“标准帧”）<br>复位类型：SYSRSn |

## 19.15.2.37 CAN_IF3MCTL 寄存器

偏移地址：14Ch　复位值：00000000h

IF3 消息控制寄存器的各位镜像消息对象的消息控制位。

表 19-92 CAN_IF3MCTL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| NewDat | MsgLst | IntPnd | UMask | TxIE | RxIE | RmtEn | TxRqst |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EoB | 保留 | | | DLC | | | |
| R-0h | R-0h | | | R-0h | | | |

表 19-93 CAN_IF3MCTL 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留。 |
| 15 | NewDat | R | 0h | 新数据。<br>0 自上次该标志被 CPU 清除以来，消息处理器未向该消息对象的数据部分写入新数据。<br>1 消息处理器或 CPU 已向该消息对象的数据部分写入新数据。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 14 | MsgLst | R | 0h | 消息丢失（仅对方向 = 接收的消息对象有效）。<br>0 自上次该位被 CPU 复位以来没有消息丢失。<br>1 当 NewDat 仍处于置位状态时，消息处理器向该对象存储了一条新消息，因此先前的消息已被覆盖。<br>复位类型：SYSRSn |
| 13 | IntPnd | R | 0h | 中断挂起。<br>0 该消息对象不是中断源。<br>1 该消息对象是中断源。如果没有其他更高优先级的中断源，中断寄存器中的中断标识符将指向该消息对象。<br>复位类型：SYSRSn |
| 12 | UMask | R | 0h | 使用接收掩码。<br>0 忽略掩码。<br>1 使用掩码（Msk[28:0]、MXtd 和 MDir）进行接收滤波。<br>如果 UMask 位置为 1，则必须在消息对象初始化期间、在 MsgVal 置为 1 之前，对消息对象的掩码位进行编程。<br>复位类型：SYSRSn |
| 11 | TxIE | R | 0h | 发送中断使能。<br>0 帧成功发送后不会触发 IntPnd。<br>1 帧成功发送后将触发 IntPnd。<br>复位类型：SYSRSn |
| 10 | RxIE | R | 0h | 接收中断使能。<br>0 帧成功接收后不会触发 IntPnd。<br>1 帧成功接收后将触发 IntPnd。<br>复位类型：SYSRSn |
| 9 | RmtEn | R | 0h | 远程使能。<br>0 在接收到远程帧时，TxRqst 不变。<br>1 在接收到远程帧时，TxRqst 被置位。<br>复位类型：SYSRSn |
| 8 | TxRqst | R | 0h | 发送请求。<br>0 该消息对象未在等待发送。<br>1 该消息对象的发送已被请求且尚未完成。<br>复位类型：SYSRSn |
| 7 | EoB | R | 0h | 块结束。<br>0 该消息对象是 FIFO 缓冲区块的一部分，且不是该 FIFO 缓冲区块的最后一个消息对象。<br>1 该消息对象是单个消息对象，或者是 FIFO 缓 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 冲区块中的最后一个消息对象。<br>注：该位用于将多个消息对象串联起来以构建 FIFO 缓冲。对于单个消息对象（不属于 FIFO 缓冲），该位必须始终置为 1。<br>复位类型：SYSRSn |
| 6-4 | RESERVED | R | 0h | 保留。 |
| 3-0 | DLC | R | 0h | 数据长度代码。<br>0-8 数据帧具有 0-8 个数据字节。<br>9-15 数据帧具有 8 个数据字节。<br>注：消息对象的数据长度代码必须与其他节点上具有相同标识符的所有对应对象中定义的一致。当消息处理器存储一个数据帧时，它会把 DLC 写为接收消息所给出的值。<br>复位类型：SYSRSn |

### 19.15.2.38 CAN_IF3DATA 寄存器

偏移地址：150h　复位值：00000000h

该寄存器提供对 CAN 报文数据字节的访问窗口。

表 19-94 CAN_IF3DATA 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data_3 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data_2 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data_1 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data_0 | | | | | | | |
| R-0h | | | | | | | |

表 19-95 CAN_IF3DATA 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | Data_3 | R | 0h | 数据字节 3。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 23-16 | Data_2 | R | 0h | 数据字节 2。<br>复位类型：SYSRSn |
| 15-8 | Data_1 | R | 0h | 数据字节 1。<br>复位类型：SYSRSn |
| 7-0 | Data_0 | R | 0h | 数据字节 0。<br>复位类型：SYSRSn |

### 19.15.2.39 CAN_IF3DATB 寄存器

偏移地址：154h　复位值：00000000h

该寄存器提供对 CAN 报文数据字节的访问窗口。

表 19-96 CAN_IF3DATB 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Data_7 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Data_6 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data_5 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data_4 | | | | | | | |
| R-0h | | | | | | | |

表 19-97 CAN_IF3DATB 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | Data_7 | R | 0h | 数据字节 7。<br>复位类型：SYSRSn |
| 23-16 | Data_6 | R | 0h | 数据字节 6。<br>复位类型：SYSRSn |
| 15-8 | Data_5 | R | 0h | 数据字节 5。<br>复位类型：SYSRSn |
| 7-0 | Data_4 | R | 0h | 数据字节 4。<br>复位类型：SYSRSn |

### 19.15.2.40 CAN_IF3UPD 寄存器

偏移地址：160h　复位值：00000000h

IF3 寄存器组的自动更新功能可以针对每个消息对象进行配置。如果专用的 IF3UpdEn 标志置位，则该消息对象被使能自动 IF3 更新。这意味着该消息对象上处于活动状态的 NewDat 标志（例如由于接收到一个 CAN 帧）将触发把整个消息对象自动复制到 IF3 寄存器组。

**注意：**不应为发送对象设置 IF3 更新使能。

表 19-98 CAN_IF3UPD 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| IF3UpdEn | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| IF3UpdEn | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| IF3UpdEn | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IF3UpdEn | | | | | | | |
| R/W-0h | | | | | | | |

表 19-99 CAN_IF3UPD 寄存器字段说明

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | IF3UpdEn | R/W | 0h | IF3 更新使能（针对所有消息对象）。<br>0 该消息对象的自动 IF3 更新被禁止。<br>1 该消息对象的自动 IF3 更新被使能。如果消息对象的 NewDat 标志处于活动状态，则该消息对象被调度复制到 IF3 寄存器组。<br>复位类型：SYSRSn |

# 20 快速串行接口（FSI）

快速串行接口（FSI）模块是一种串行通信外设，可通过隔离器件实现可靠的高速通信。电气隔离器件用于两套无共用电源与地连接的不同电子电路之间必须进行信息交互的场景。

尽管隔离器件为这类信号通信提供了便利，但也会在信号线上引入较大延迟，并增加信号间的偏斜。FSI 专为需要跨隔离壁垒通信、且无需额外增加元器件的系统场景而设计，以确保实现可靠的高速通信。

FSI 由相互独立的发送器（FSITX）和接收器（FSIRX）内核构成。FSITX 与 FSIRX 内核可独立配置与运行。

## 20.1 FSI 特性

FSI 模块具有以下特性：

- 独立的发送器与接收器内核
- 源同步传输
- 双倍数据速率（DDR）
- 单条或两条数据线
- 可配置的数据长度
- 偏斜调整模块，用于补偿电路板及系统延迟失配
- 帧错误检测
- 可配置帧标记，用于消息过滤
- 硬件 Ping 功能，用于检测通信过程中的线路中断（Ping 看门狗）
- 每个 FSI 内核支持两个中断
- 外部触发帧生成
- 硬件或软件计算的 CRC
- 嵌入式 ECC 计算模块

- 寄存器写保护
- FSI-SPI 兼容模式（功能有限）
- 标记匹配通知

# 20.2 FSI 结构

## 20.2.1 CPU 接口

下图展示了每个 FSI 模块的 CPU 接口。

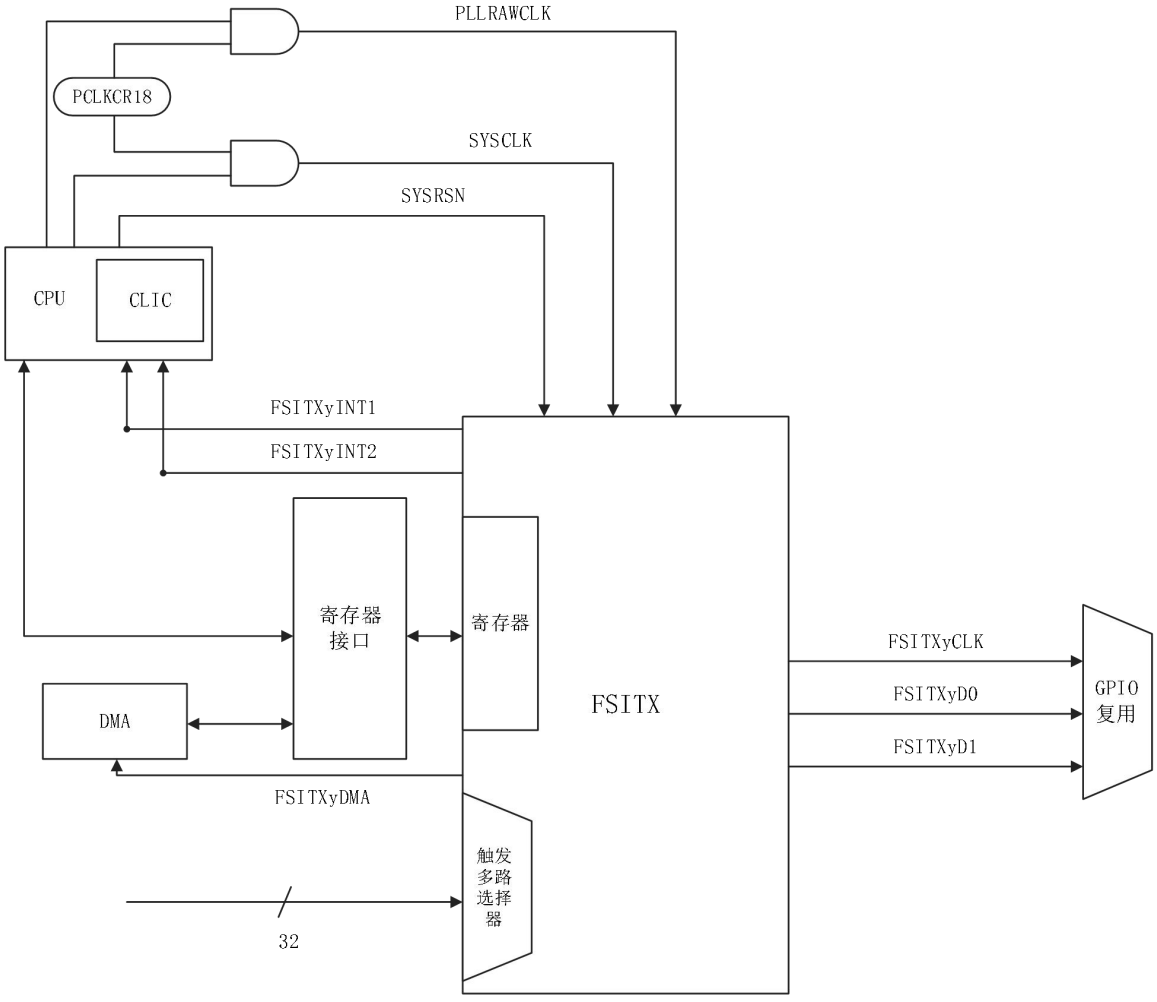



图 20-1 FSITX 结构

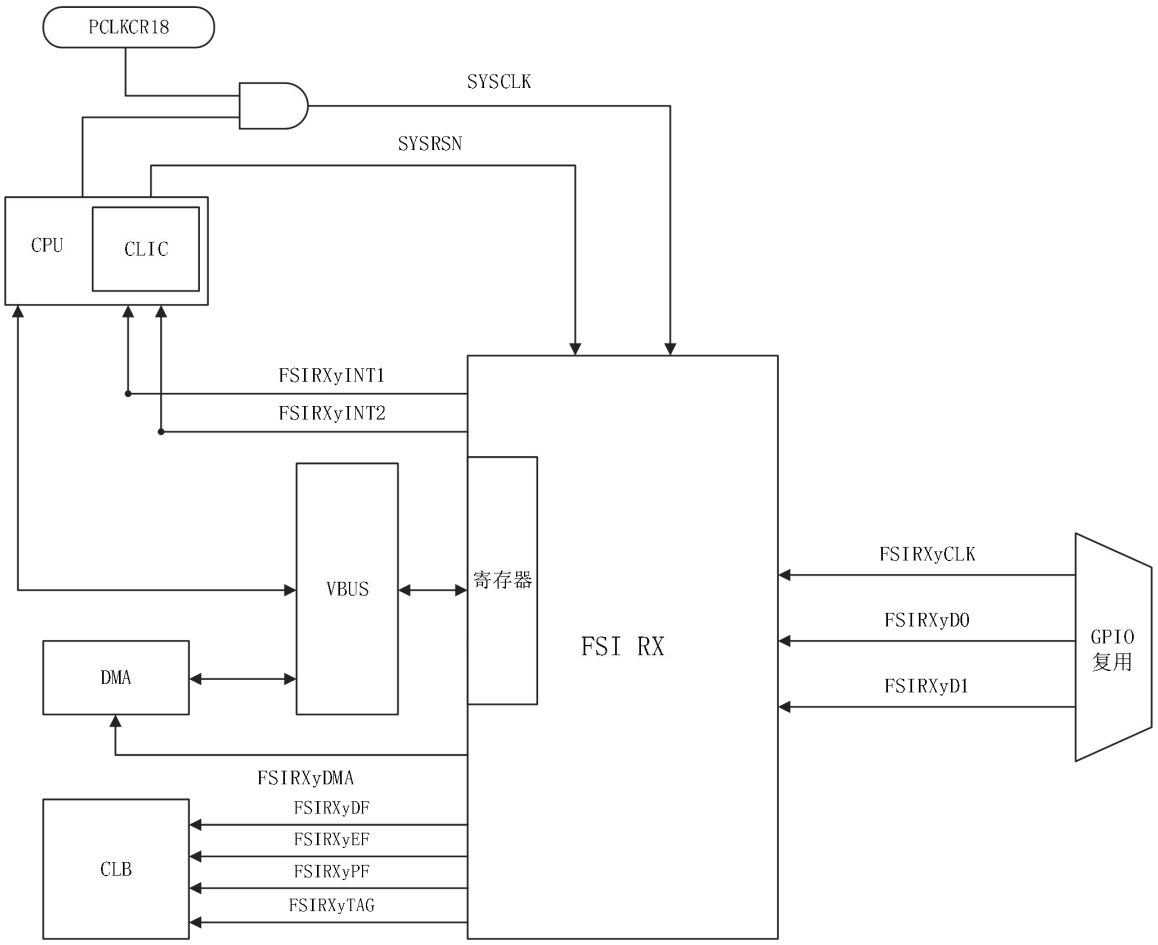



图 20-2 FSIRX 结构

## 20.2.2 信号描述

FSI 是一种点对点通信协议。因此，一个 FSI 发送器内核直接与单个 FSI 接收器内核进行通信。同样，一个 FSI 接收器内核仅从单个 FSI 发送器内核接收数据。

每个 FSI 内核包含三路信号：一路时钟信号和两路数据信号。数据在发送或接收时，每一帧字段的最高位（MSB）优先传输。如果不使用多通道传输，TXD1 和 RXD1 信号可以悬空不连接，对应的 GPIO 可重新用于其他应用需求。表中- 和表中 - 描述了可通过 PADCONFIG 寄存器选择并引出到器件引脚的各类信号。

最大 RXCLK 率不得超过 SYSCLK /2。

表 20-1 FSI 接收端核心信号

| 信号名称 | 方向 | 描述 | 非活动电平 |
|---|---|---|---|
| RXCLK | 输入 | FSI 接收模块的接收时钟输入信号。必须连接至发送端 FSI 模块的 TXCLK。 | 高电平 (Logic High) |
| RXD0 | 输入 | 接收的主数据输入线。必须连接至发送端 FSI 模块的 TXD0。 | 高电平 (Logic High) |
| RXD1 | 输入 | 接收的辅助数据输入线。如需使用多通道（multilane）传输，该信号必须连接至发送端 FSI 模块的 TXD1。 | 高电平 (Logic High) |

表 20-2 FSI 发送端核心信号

| 信号名称 | 方向 | 描述 | 非活动电平 |
|---|---|---|---|
| TXCLK | 输出 | 发送时钟信号，由 FSI 发送模块驱动。在数据传输期间，帧起始阶段（Preamble）之前会发送 4 个时钟沿，帧最后一位之后会跟 4 个时钟沿（Postamble）。数据在时钟的双边沿进行传输。<br>在 FSI-SPI 兼容模式下，不会发送前导（Preamble）和后导（Postamble）时钟沿。数据仅在时钟的单边沿传输：数据在时钟上升沿发送，在下降沿接收。 | 高电平 (Logic High) |
| TXD0 | 输出 | 传输的主数据输出线，由 FSI 发送模块驱动。<br>当 FSI 配置为多通道（multilane）传输模式时，TXD0 包含数据字和 CRC 字节的所有偶数位（even-numbered bits）。其他帧字段（如帧类型、帧起始、标签 Tag 和帧结束）将完整传输。 | 高电平 (Logic High) |
| TXD1 | 输出 | 当 FSI 配置为多通道传输模式时的辅助数据输出线，由 FSI 发送模块驱动。<br>传输期间，数据位会被拆分到 TXD0 和 TXD1 上。TXD1 包含数据字和 CRC 字节的所有奇数位（odd-numbered bits）。这仅适用于数据字和 CRC 字节；对于其他帧相关信息（如帧类型、帧起始、标签 Tag 和帧结束），该信号线的状态与 TXD0 完全一致。 | 高电平 (Logic High) |

#### 20.2.2.1 引脚配置

配置设备引脚：必须配置 GPIO 多路复用器寄存器，以将此外设连接到设备引脚。为避免引脚上出现毛刺，需先配置 GPyGMUX 位（同时保持相应的 GPyMUX 位为默认值 0），然后再将 GPyMUX 寄存器写入所需值。某些 I/O 功能由 GPIO 寄存器设置定义，与该外设无关。对于输入信号，需通过将相应的 GPxQSELn 寄存器位设置为 0x3，将 GPIO 输入资格设置为异步模式。内部上拉电阻可在 GPyPUD 寄存器中进行配置。

### 20.2.3 中断

每个 FSI 模块包含多个中断源，这些中断源可被分配到两个不同的中断向量：INT1 和 INT2。每个中断源在 EVT_STS、EVT_FRC 和 EVT_CLR 寄存器中分别具有对应的状态标志位、强制触发位和清除位。

每个中断均可分配至中断向量 INT1 或 INT2，以实现两个优先级等级。或者，也可以禁止该中断源产生任何中断，但其状态标志位仍可被软件置位并监测。发送端事件通过 TX_INT_CTRL 寄存器分配至相应中断向量。接收端事件则通过 RX_INT1_CTRL 和 RX_INT2_CTRL 寄存器分配中断向量。如果不需要某个中断，请确保在对应的 INT_CTRL 寄存器中不使能该位。

#### 20.2.3.1 TX 中断

发送器可产生以下中断：

帧完成（FRAME_DONE）：该事件表示 FSI 已完成一帧数据的发送。

缓冲器下溢（BUF_UNDERRUN）：该事件表示发送缓冲器出现下溢。当发送器试图从 CPU 或 DMA 尚未写入数据的地址读取数据时，会发生缓冲器下溢。

缓冲器上溢（BUF_OVERRUN）：当缓冲器出现上溢时产生该中断。若数据在被发送前就被新数据覆盖，则会发生缓冲器上溢。

Ping 帧触发（PING_TRIGGERED）：当 Ping 帧被触发时产生该中断。当 Ping 计数器超时或外部 Ping 触发事件发生时，该标志位被置位。

#### 20.2.3.2 RX 中断

接收器内核可通过多种不同事件产生中断：

Ping 看门狗超时 (PING_WD_TO)：此事件表示 Ping 看门狗定时器已超时。这意味接收器在 RX_PING_WD_REF 寄存器指定的时限内未接收到有效帧。

帧看门狗超时 (FRAME_WD_TO)：此事件表示帧看门狗定时器已超时。该超时的触发条件通过 RX_FRAME_WD_CTRL 寄存器进行设置。一旦检测到帧起始阶

段（Start of Frame），帧看门狗计数器就会从 0 开始计数。帧结束阶段（End of Frame）必须在看门狗计数器达到参考值之前完成。如果未按时完成，看门狗就会超时并生成此事件。如果发生此事件，接收器必须进行软件复位（Soft Reset）并随后重新同步（Resynchronization），以恢复正常工作。

CRC 校验错误 (CRC_ERR)：此错误表示发生了 CRC 校验错误。当接收到的 CRC 与本地计算出的 CRC 不匹配时，会生成 CRC 错误。

帧类型错误 (TYPE_ERR)：此错误表示接收到了无效的帧类型。如果发生此错误，接收器必须进行软件复位（Soft Reset）并随后重新同步，以恢复正常工作。

帧结束错误 (EOF_ERR)：此错误表示接收到了无效的帧结束位模式（Bit Pattern）。如果发生此错误，接收器必须进行软件复位（Soft Reset）并随后重新同步，以恢复正常工作。

接收缓冲区上溢 (BUF_OVERRUN)：此事件表示接收缓冲区（Receive Buffer）发生了上溢（溢出）状况。

接收缓冲区下溢 (BUF_UNDERRUN)：此事件表示接收缓冲区发生了下溢状况。当软件尝试去读取一个空缓冲区时，就会触发此状况。

帧完成 (FRAME_DONE)：此事件表示接收到了一个有效且无错误的帧。

接收到错误帧 (ERR_FRAME)：此事件表示接收到了一个错误帧（Error Frame）。

接收到 Ping 帧 (PING_FRAME)：此事件表示接收到了一个 Ping 帧。

帧覆写/溢出 (FRAME_OVERRUN)：此事件表示在 FRAME_DONE 标志位仍处于置位状态（未清除）时，又接收到了一个新的帧。

接收到数据帧 (DATA_FRAME)：此事件表示接收到了一个数据帧（Data Frame）。

Ping 标签匹配 (PING_TAG_MATCH)：此事件表示接收到了一个带有点匹配标签（Tag）的 Ping 帧。

数据标签匹配 (DATA_TAG_MATCH)：此事件表示接收到了一个带有匹配标签的数据帧。

错误标签匹配 (ERROR_TAG_MATCH)：此事件表示接收到了一个带有匹配标签的错误帧。

注意：如果中断时间被注册到两个中断向量中，则两个中断向量都会触发。没有硬件检测是否存在中断任务重叠。

#### 20.2.3.3 配置中断

要配置 FSI 的中断，应用程序必须为每个所需事件选择中断向量：发送端使用 TX_INT_CTRL 寄存器，接收端使用 RX_INT1_CTRL 和 RX_INT2_CTRL 寄存器。

如果一个事件同时被注册到两个中断向量上，两个中断都会触发。硬件不会对中断向量分配冲突进行检查。

#### 20.2.3.4 处理中断

在中断服务程序内部，用户必须通过 EVT_CLR 寄存器清除事件标志，然后对 CPU 中断进行应答。

如果某个事件在软件清除对应标志位之前多次发生，不会产生新的中断。

若多个事件同时发生，或在时间上非常接近，有可能在单次中断内处理多个状态。每个标志位由硬件独立置位，且必须由应用软件清除。如果发生多个不同事件，中断服务程序可按照应用所需的任意顺序处理各个事件。不建议在每次中断服务程序中清除整个中断状态寄存器，这可能导致应用程序丢失事件，从而对程序运行造成不利影响。下面给出接收器的中断处理示例流程；发送端的处理流程与之类似。

接收到中断时，将接收事件与错误状态标志寄存器（RX_EVT_STS）的当前状态复制到一个局部快照变量中。

读取快照中的所有位，以确定需要处理的事件。

对快照中识别到的每个事件执行必要操作。

向接收事件与错误清除寄存器（RX_EVT_CLR）写入该快照值，仅清除在中断服务程序开始时已置位的那些中断。

对每次触发的中断服务程序重复上述流程。

由于仅处理并清除了事件快照，在刚执行完的中断服务程序期间有可能产生了其他事件，因此在中断服务程序末尾，事件标志仍可能处于置位状态。当中断服务程序一结束，就会产生新的中断，该标志依然保持置位，可被相应处理。

与多个事件相关、且在同一中断服务程序中处理的软件访问，可能引发竞态条件，导致软件无法按预期运行。例如，若用户需要同时使能 Ping 帧和数据帧事件，建议使用不同的中断线。如果两个事件在同一中断线内处理，当两个事件在时间上非常接近时，软件可能只能响应其中一个事件。

### 20.2.4 DMA 接口

发射器和接收器均支持使用 DMA 进行自动数据传输。DMA 触发器与中断信号无关，DMA 事件仅在数据帧完成时触发。发射器的 DMA 触发器可通过将 TX_DMA_CTRL 设置为 1 来启用；同时，发射器还需将 TX_OPER_CTRL_LO.START_MODE 设置为 0x2，以便允许对 TX_FRAME_CTRL 的 start 位或 TX_FRAME_TAG_UDATA 寄存器进行写操作以启动传输。接收器的 DMA 触发器可通过将 RX_DMA_CTRL.DMA_EVT_EN 设置为 1 来启用。

### 20.2.5 外部帧触发多路复用器：

FSI 设备的发射模块连接有两个多路复用器，这些多路复用器用于选择启动 ping 帧和通用帧的触发源。每个类型的帧均可独立配置这些多路复用器。应用程序可为每种帧类型选择一个触发源。使用这些触发源是可选的。外部 ping 帧触发可通过将 TX_PING_CTRL 中的 EXT_TRIG_SEL 设置为所需触发源的索引来配置；同时必须将 TX_PING_CTRL 中的 EXT_TRIG_EN 设置为“1”，以允许该触发源生成 ping 帧。通用帧触发可通过将 TX_OPER_CTRL_HI 中的 EXT_TRIG_SEL 设置为所需触发源的

索引来配置；且需将 TX_OPER_CTRL_LO 中的 START_MODE 设置为“0x1”，方可通过外部触发发帧。

表 20-3 FSI 外部触发源和索引

| EXT_TRIG_SEL | 信号触发源 |
|---|---|
| 0 | EPWMXBAR1 |
| 1 | EPWMXBAR2 |
| 2 | EPWMXBAR3 |
| 3 | EPWMXBAR4 |
| 4 | EPWMXBAR5 |
| 5 | EPWMXBAR6 |
| 6 | EPWMXBAR7 |
| 7 | EPWMXBAR8 |
| 8 | EPWM1_SOCA |
| 9 | EPWM1_SOCB |
| 10 | EPWM2_SOCA |
| 11 | EPWM2_SOCB |
| 12 | EPWM3_SOCA |
| 13 | EPWM3_SOCB |
| 14 | EPWM4_SOCA |
| 15 | EPWM4_SOCB |
| 16 | EPWM5_SOCA |
| 17 | EPWM5_SOCB |
| 18 | EPWM6_SOCA |
| 19 | EPWM6_SOCB |
| 20 | EPWM7_SOCA |
| 21 | EPWM7_SOCB |
| 22-39 | Reserved |
| 40 | CLB1_OUT30 |
| 41 | CLB1_OUT31 |
| 42 | CLB2_OUT30 |
| 43 | CLB2_OUT31 |
| 44-51 | Reserved |
| 52 | ADCSOCA |
| 53 | ADCSOCB |
| 54 | CPU_INIT0 |
| 55 | CPU_INIT1 |
| 56 | CPU_INIT2 |

## 20.3 功能描述

### 20.3.1 操作介绍

快速串行接口发射器与接收器模块（FSI_TX/FSI_RX）是设备上两个完全独立的模块。每个模块均拥有独立的控制寄存器组、时钟信号及中断处理机制。下文将分别阐述发射器与接收器的帧格式以及各项初始化与配置流程。

### 20.3.2 FSI 发送模块

FSI 发射器模块负责数据的帧化、CRC 生成、 TXCLK 、TXD0 和 TXD1 的信号生成以及中断生成。发射器核心的操作通过可编程控制寄存器进行控制和配置。这些控制寄存器允许 CPU 对 FSI 接收器的操作进行编程、控制和监控。发射数据缓冲区可由 CPU 和 DMA 访问。发射器具有以下特性：

- 自动 ping 帧生成
- 外部触发 ping 帧
- 外部触发数据帧
- 软件可配置帧长度
- 16 字节数据缓冲区
- 数据缓冲区欠流和溢出检测
- 硬件生成的数据位 CRC 校验
- 对选定数据进行软件 ECC 计算
- 支持 DMA。

图中展示了 FSI 发射器的高层框图。图中展示了发射器核心子模块的框图。接下来的章节将详细描述 FSI 发射器的各项功能。

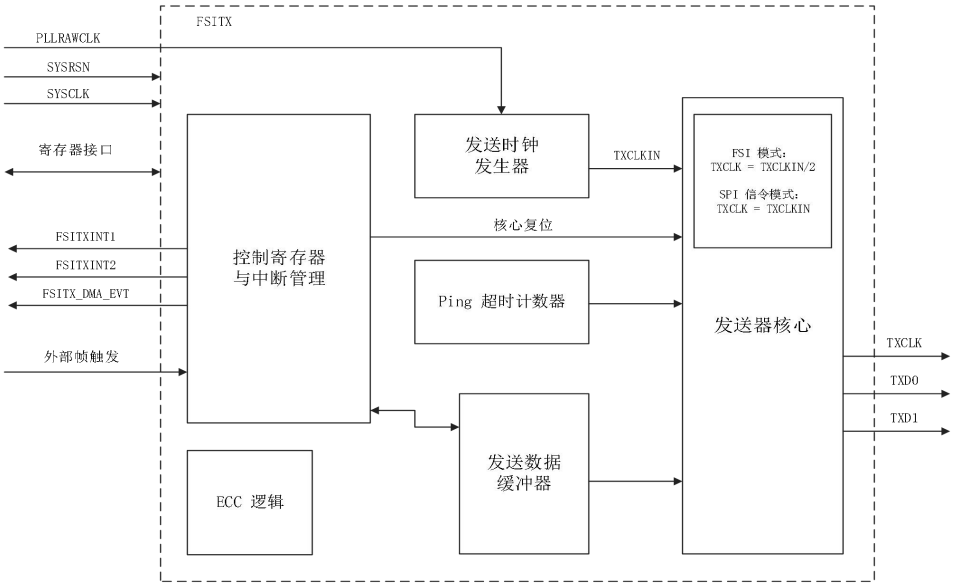



图 20-3 FSI 发射机框图

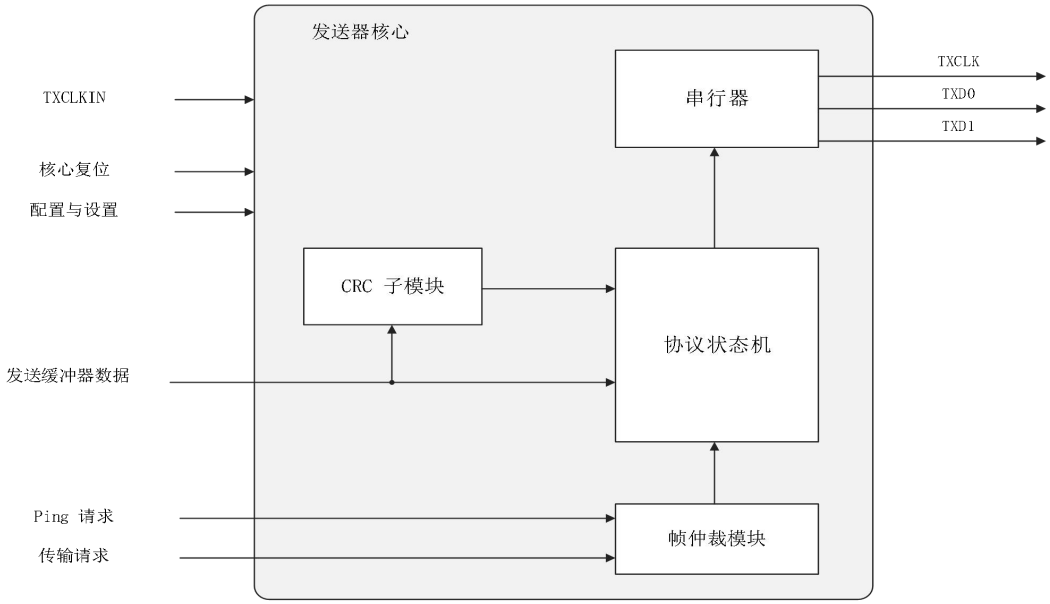



图 20-4 FSI 发射机核心框图

### 20.3.2.1 初始化

在首次初始化时，或者因下溢（Underrun）状况触发模块复位之后，发送器模块会执行以下初始化顺序，以启动或恢复传输操作：

1.将 TX_CLK_CTRL.CLK_RST 位置 1，随后再将其清零，以此对发送器时钟进行初始化。

2.将 TX_OPER_CTRL_LO.SEL_PLLCLK 位置 1，将发送器核心的时钟源设置为 PLLRAWCLK。

3.通过向 TX_CLK_CTRL.PRESCALE_VAL 写入目标值，设置时钟预分频系数（Prescaler Value）。

4.将 TX_CLK_CTRL.CLK_EN 位置 1，以使能发送器时钟分频器。

5.向 TX_MASTER_CTRL 写入 0xA501，对发送器模块施加软件复位（Soft Reset）。

6.等待 4 个 TXCLK 时钟周期。

7.向 TX_MASTER_CTRL 写入 0xA500，将发送器核心从复位状态中释放（解除复位）。

在完成初始化和配置之后，发送器模块会在开始传输前与接收器模块进行同步。

### 20.3.2.2 FSI TX 时钟

发射器核心寄存器和控制逻辑均基于设备系统时钟（SYSCLK）运行。 FSI 发射时钟（TXCLK）源自 pllrawclk。pllrawclk 通过配置时钟预分频值（TX_CLK_CTRL.PRESCALE_VAL）并设置时钟分频使能位（TX_CLK_CTRL.CLK_EN）来进行分频。时钟预分频值可设置为将 pllrawclk 分频至 1（TX_CLK_CTRL.PRESCALE_VAL = 0x0 或 0x1）至 255（TX_CLK_CTRL.PRESCALE_VAL = 0xFF）。尽管 TXCLK 和 SYSCLK 均源自 pllrawclk，但 TXCLK 与 SYSCLK 之间是异步关系。

TXCLK 的配置值绝不应设置为快于 SYSCLK /2

### 20.3.2.3 传输帧

在发送器端，Ping 帧是唯一无需任何额外软件或 DMA 干预，即可完成设置并发送的帧。Ping 帧可以通过以下三种来源中的任何一种（或全部）进行发送：自动 Ping 定时器、软件触发或外部触发。

每种可用的帧类型都可以通过多种方式发送。通常情况下，在发送帧之前必须执行以下步骤。这些步骤可以在设置启动条件（Start Condition）之前的任意顺序下执行：

配置帧类型（Configure the frame type）

设置帧标签（Set the frame tag）

如果要发送的帧是数据帧：

设置用户数据（Set the user data）

向数据缓冲区写入数据（Write to the data buffer）

如果该帧是软件定义帧长度的类型，设置字长/数据长度（Set the word length）

设置启动条件（Set the start condition）

帧起始传输限制：可通过 TX_OPER_CTRL_LO 中的 START_MODE 位所选方法之一启动新的帧传输。若当前正因硬件触发的 PING 定时器而进行 PING 帧传输，则当正在进行的 PING 传输完成时，新的帧传输将立即开始。一旦启动了帧起始传输，下一个帧起始传输将在首个帧开始传输其“帧结束”（EOF）字段时被识别。若在当前传输尚未到达 EOF 字段之前，接收到新的起始触发信号，则该触发信号将失效，且不会发出任何通知。

未实现硬件检查机制，用于验证软件写入的类型字段是否有效。若使用了无效类型并启动帧传输，则行为如下：

- 传输的帧结构与 NWORD 数据帧完全相同。数据帧的大小由 TX_FRAME_CTRL.N_WORDS 寄存器中的值决定。
- 传输的数据帧中的帧类型字段将按程序设定的方式传输。如果 FSI 接收器接收到此字段，则会生成“类型错误”。该机制可用于在接收帧中强制产生“类型错误”，以进行测试目的。

#### 20.3.2.3.1 软件触发帧

传输数据帧最基本的方式是通过软件实现。每个步骤均需由应用程序处理。若要通过软件发送数据帧，必须执行以下步骤：步骤 1 至 6 在设置 TX_FRAME_CTRL.start 之前，可按任意顺序执行。某些字段无需在每次传输时都进行重新配置；帧标签、用户数据及帧类型均为“粘性”字段——除非经软件修改，否则这些字段将在后续帧中被重复传输。

1.将待传输的数据写入发送数据缓冲区的下一个位置。

2.将 `TX_FRAME_CTRL` 的 `FRAME_TYPE` 设置为相应帧类型对应的值。

3.若 `TX_FRAME_CTRL` 的 `FRAME_TYPE` 设置为 0011（即软件定义长度数据帧的帧类型），则将 `TX_FRAME_CTRL.N_WORDS` 设置为待传输字数减 1；例如，若需传输 16 个字，则 `N = 16`，应将 `TX_FRAME_CTRL.N_WORDS` 设置为 15。

4.在帧组装完成后再进行传输时， FSITX 硬件会计算出待传输的 CRC 值。若 `TX_OPER_CTRL_LO.SW_CRC` 为 1，则应用程序可自行计算自定义 CRC 值，并将该值设置为 `TX_USER_CRC`。

5.将 `TX_FRAME_TAG_UDATA.FRAME_TAG` 设置为所需标签。

6.将 `TX_FRAME_TAG_UDATA.USER_DATA` 设置为所需用户数据。

7.将 `TX_FRAME_CTRL.start` 设置为 1，以启动数据帧的传输。

一旦帧传输开始，`TX_FRAME_CTRL.start` 信号将被硬件置零。如需监控帧是否已完成，软件可轮询 `TX_EVT_STS.FRAME_DONE`。

#### 20.3.2.3.2 外部触发帧

发射器可在外部源触发时传输帧。要使用外部触发器传输帧，应用程序必须遵循第 22.3.2.3.1 节中描述的相同流程。唯一的区别在于，在步骤 7 中，当满足外部触发条件时，起始条件将自动设置，而非通过软件设定。需要注意的是，通过外部触发传输帧时，待发送的帧信息是从前一节所述的相同寄存器中提取的。因此，可以通过外

部触发发送任何类型的帧，包括 ping 帧、错误帧和数据帧。此外， FSI 没有硬件机制来判断是否发生了多个触发事件。 FSITX 会直接接收这些数据，而应用程序则需确保这些数据已按需更新。通过使用 TX_EVT_STS 字段，无论是通过轮询还是中断方式，应用程序都可以填充或更新下一次要发送的帧信息。

#### 20.3.2.3.3 ping 帧生成

假设 FSI 发射器已正确初始化，可使用以下序列来配置并发送 ping 帧。

##### 20.3.2.3.3.1 自动 ping 帧生成

要生成周期性 ping 帧，必须遵循以下步骤：

1. 通过向 TX_PING_CTRL.CNT_RST 写入 1 来初始化 ping 计数器。
2. 将所需 ping 标签设置为 TX_PING_TAG.TAG。
3. 将 ping 定时器参考值设置为 TX_PING_TO_REF.TO_REF。
4. 通过向 TX_PING_CTRL.TIMER_EN 写入 1 来启用 ping 定时器。

Ping 定时器是一个自由运行的计数器，从 0 开始递增。Ping 定时器计数器的当前值可通过 TX_PING_TO_CNT 获取。当 TX_PING_TO_CNT 的当前值与参考值 TX_PING_TO_REF 相匹配时，TX_EVT_STS.PING_TRIGGERED 会被置位。此时，TX_PING_TO_CNT 将重置为 0 并重新开始计数，直至再次出现匹配情况，或通过软件（将 TX_PING_CTRL.TIMER_EN 设置为 0）停止 Ping 定时器。

##### 20.3.2.3.3.2 软件触发 ping 帧

软件还可手动生成 ping 帧。通过软件发送 ping 帧的流程与发送其他类型帧的流程非常相似，必须遵循以下步骤：

1.将 `TX_FRAME_CTRL` 的 `FRAME_TYPE` 字段设置为 `0000‘b`，以表示正在发送的帧为 Ping Frame。

2.将 `TX_FRAME_TAG_UDATA` 的 `FRAME_TAG` 字段设置为所需值。

3. 将 `TX_FRAME_CTRL` 的 `start` 字段写入 `1`，即可启动传输过程。

一旦帧传输开始，`TX_FRAME_CTRL.start` 信号将被硬件置零。如需监控帧是否已完成，软件可轮询 `TX_EVT_STS.FRAME_DONE`。

#### 20.3.2.3.3.3 外部触发 ping 帧

生成 ping 帧的最后一个输入源为外部触发器。可从最多 32 个不同的触发器中进行选择。

Ping 帧可由外部触发源和内部 ping 定时器共同触发。若 TX_PING_CTRL.EXT_TRIG_EN 设置为 1，则优先采用外部触发源，忽略 ping 定时器。

#### 20.3.2.3.4 使用 dma 传输帧

FSI 发射器可发送通过 DMA 连续传输的数据。每当数据帧传输完成时，都会生成一个 DMA 触发信号，该信号与设置 TX_EVT_STS 的 FRAME_DONE 信号同步，从而置位 FRAME_DONE 标志。要通过 DMA 传输连续数据，需对发射器进行以下配置：首先，将 TX_DMA_CTRL 中的 DMA_EVT_EN 设为 1，以便 DMA 触发信号能够传递至 DMA 模块；其次，将 TX_OPER_CTRL_LO 的 START_MODE 设为 0x2。至此，发射器即可通过软件写入 TX_FRAME_CTRL 的 start 或 TX_FRAME_TAG_UDATA 字段来启动传输。

DMA 还需正确配置，以便 FSI 能够发送数据。利用 DMA 持续向发送缓冲区写入数据的一种方法如下：

1.设置两个 DMA 通道，使其由同一个 FSI 发射器和 DMA 触发器触发。

2.配置其中一个通道用于填充发送缓冲区。

3.配置另一个通道用于设置帧标签和用户数据字段。

4.由于 FSI 发送缓冲区是一个 16 字节的循环缓冲区，需确保负责该数据缓冲区的 DMA 通道在复制完 16 个字节后能够完成环绕操作。

由于必须写入帧标签和用户数据才能启动帧的传输，需使用两个连续的 DMA 通道。此举可确保 DMA 通道始终按顺序执行：负责数据缓冲区的 DMA 通道应为编号较小的一条，而处理标签/用户数据的 DMA 通道则应为下一条。例如：可将 DMA 通道 3 配置为处理数据缓冲区，同时将 DMA 通道 4 配置为处理标签及用户数据。

#### 20.3.2.4 发送缓冲区管理

FSI 发射器配备了一个 16 字节的缓冲区，用于从该缓冲区提取待传输数据。此缓冲区采用环形缓冲区实现，而非 FIFO 结构，因此需特别注意正确解读缓冲区溢出、下溢以及 TX_BUF_PTR_STS 寄存器的状态。这些标志位和指针均基于软件或 DMA 将缓冲区视为环形缓冲区的假设而运作。这种工作模式是使溢出、下溢及指针状态具有实际意义的唯一方式。若数据由 DMA 提供且存在其他周期性触发机制试图启动传输，则下溢将构成严重错误。当发生下溢时，意味着缓冲区已失去同步状态。这不仅会影响当前传输过程，还可能因环形缓冲区特性而影响所有后续传输。在此情况下，需通过软复位来彻底恢复系统状态。另一种方法是：软件可手动停止传输、重置缓冲区指针、清除剩余错误条件后，再重新启动传输。软件方法需经过若干步骤，而软复位则为单次操作，并能确保控制寄存器被完全重置。得益于发射缓冲区的灵活性，软件可实现简单的乒乓缓冲区，或从缓冲区任意位置随机加载并发送数据。若按此方式使用缓冲区，则可忽略错误标志和状态字段，而不会对发射器功能产生不利影响。此外，若以这种方式使用 CURR_WORD_CNT，则该寄存器也将无效。应用程序可通过向 TX_BUF_PTR_LOAD 写入 4 位索引值来手动设置缓冲区指针；此举可强制发射器从缓冲区中指定位置开始选取数据。

#### 20.3.2.5 CRC 冗余校验

FSI 发射器可以通过嵌入式硬件 CRC 子模块或提供用户自定义值的方式，向正在传输的帧提供 CRC 值。这一功能可通过适当设置 TX_OPER_CTRL_LO.SW_CRC 来控制。若选择硬件生成 CRC（即 TX_OPER_CTRL_LO.SW_CRC = 0，默认设

置），CRC 值将由硬件根据 CRC 多项式 0x7（$x^8 + x^2 + x + 1$）计算数据字段和用户数据字段的 CRC 值。当帧传输时，发射器模块会自动计算数据字段的 CRC 值，无需用户干预。若选择软件生成 CRC（即 TX_OPER_CTRL_LO.SW_CRC = 1），则必须由软件计算 CRC 值并将其存入 TX_USER_CRC 寄存器。下一帧传输时，将使用 TX_USER_CRC 寄存器中存储的值替代硬件生成的 CRC 值。由于 TX_USER_CRC 寄存器可由软件编程，应用程序可利用此字段作为额外的数据字段，用于特定目的。若按此方式使用 TX_USER_CRC，则接收端对 CRC 值的检测将无效，必须忽略该检测结果。

#### 20.3.2.6 发射器必须执行软复位的条件

与接收器不同，此处无法检测到需要执行软复位的错误。发生缓冲区溢出或下溢时，是否需要执行软复位以恢复正常运行，由应用软件根据恢复策略决定。

#### 20.3.2.7 复位

SYSRSn 可对整个发射器模块及所有发射器寄存器进行复位；发射器核心可通过 SYSRSn 或通过向 TX_MASTER_CTRL 写入“1”来复位。模块复位会将所有寄存器重置为默认状态。

### 20.3.3 FSI 接收模块

接收器模块与 FSI 时钟（RXCLK）以及经过可选可编程延迟线后的数据线（RXD0 和 RXD1）相连接。接收器核心负责数据帧的构建、CRC 计算及帧相关的错误检测。接收器的位时钟和状态机由 RXCLK 输入驱动，该输入与设备系统时钟异步。接收器控制寄存器允许 CPU 对 FSI 接收器的操作进行编程、控制和监控。接收数据缓冲区可由 CPU 和 DMA 访问。

接收器核心具备以下特性：

- 16 字节数据缓冲区
- 支持多种帧类型

- Ping 帧看门狗
- 帧看门狗
- 硬件级 CRC 计算与比较
- ECC 检测
- 对输入信号的可编程延迟线控制
- DMA 支持
- FSI -SPI 兼容模式

图中展示了 FSI 接收器内部模块的高层次概览。图中显示了 FSI 接收器核心子模块的视图。并非所有数据路径和内部连接均在此图中展示。接下来的章节将详细描述 FSI 接收器模块的各个方面。

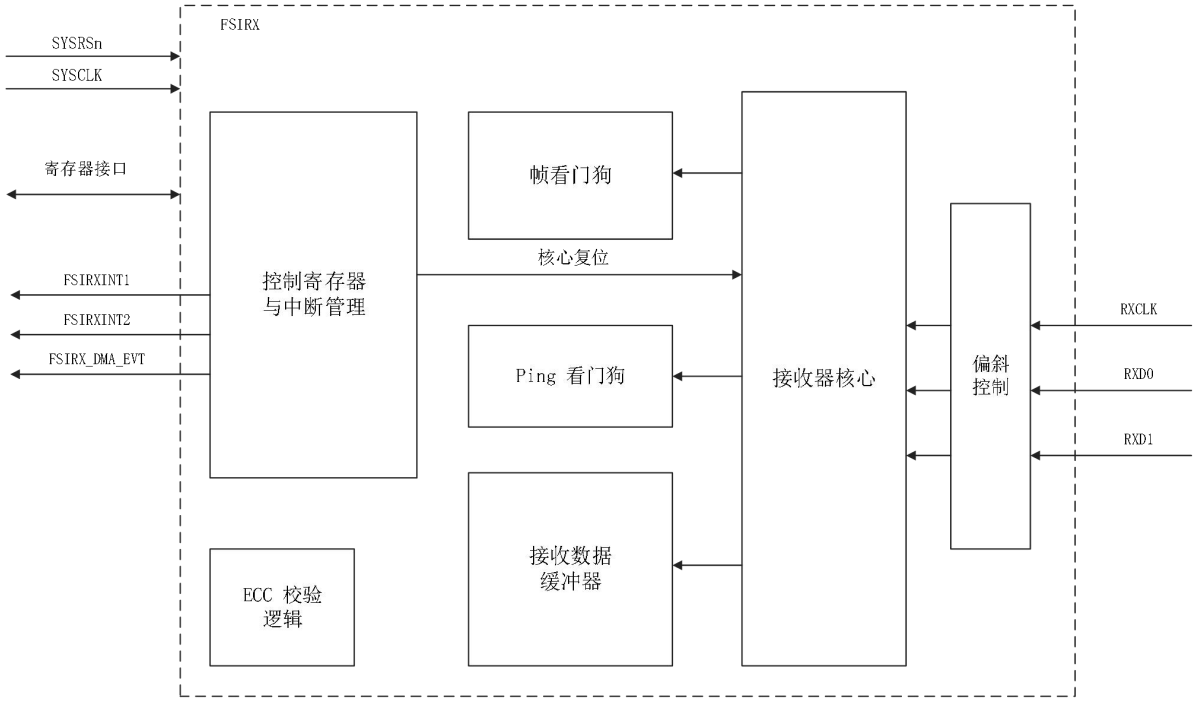



图 20-5 FSI 接收机框图

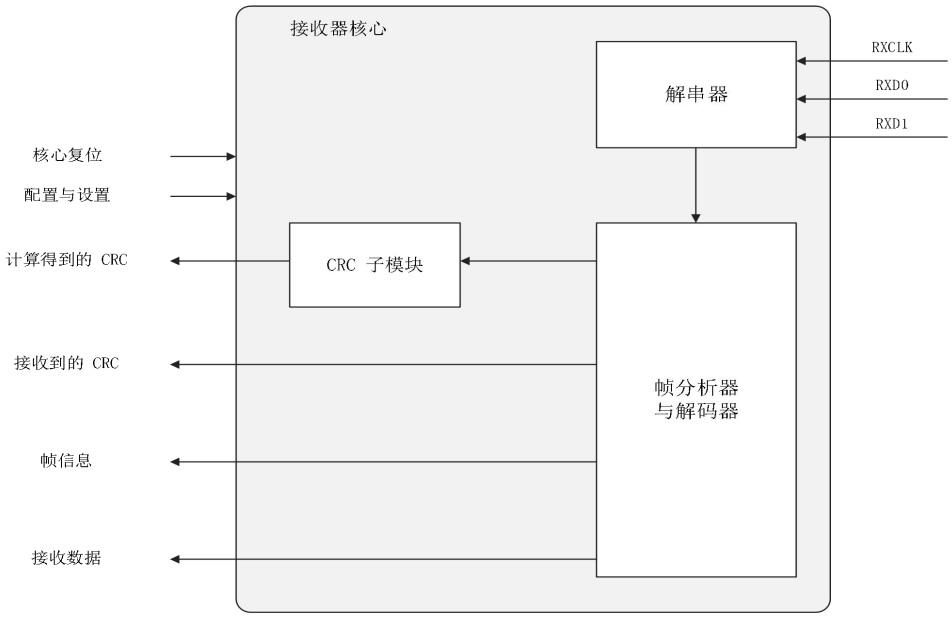



图 20-6 FSI 接收器核心框图

### 20.3.3.1 初始化

在首次初始化或发生任何帧错误后执行模块复位时，接收器模块会在进行任何其他初始化操作之前，先置位再释放接收器核心复位位（RX_MASTER_CTRL.CORE_RST）。一旦接收器模块完成初始化，将执行以下步骤：

1.如有需要，为相应的中断线路分配中断源；

2.如有需要，配置 ping 看门狗以定期检查与发射端之间的链路是否处于激活状态；

3.如有需要，配置帧看门狗以确保每个帧均能在预设的时间窗口内被接收；

4.通过向 RX_BUF_PTR_LOAD 寄存器写入数据，对接收缓冲区指针进行初始化。接收到的数据将从该寄存器中加载的地址开始存入缓冲区；

5. 确保 RX_EVT_STS 寄存器中的所有错误标志均已清零。至此，接收器已准备好接收任何传入帧。此时，软件可对 RX_EVT_STS 寄存器进行轮询，以检测各种特定条件。例如，当 RX_EVT_STS 中的 FRAME_DONE 标志位置位且无其他标志位置位时，表示接收器已成功接收了无错误的帧。随后，应用程序需配置各项功能特性，

如 ping 看门狗、帧看门狗、DMA、外部触发等——这些功能特性将在后续章节中详细阐述。至此，接收器模块已准备好与发射端进行同步，进而开始接收数据。

#### 20.3.3.2 FSI 接收时钟

接收器模块的寄存器与控制逻辑均由设备系统时钟（SYSCLK）驱动；而接收器状态机则由接收器输入时钟引脚（RXCLK）驱动。

RXCLK 绝不应快于 SYSCLK

#### 20.3.3.3 FSI 接收帧

一旦接收器经过正确的配置和同步，接收到的消息将按以下方式处理。请注意，接收器中没有用于门控输入数据的片选信号；每个有效的时钟边沿都会将数据锁存至接收器中。接收到的帧头信息会被存入相应的寄存器字段：

- RX_FRAME_INFO：FRAME_TYPE 字段包含所接收帧的类型；
- RX_FRAME_TAG_UDATA：FRAME_TAG 字段包含所接收帧的标签；
- RX_FRAME_TAG_UDATA：USER_DATA 字段包含所接收的用户数据。

若在接收过程中发生任何错误情况（例如 CRC 不匹配、帧错误、帧超时、缓冲区溢出或 ping 监控器超时），则相应的标志位将在 RX_EVT_STS 寄存器中置位。

若在运行过程中任何时刻发生帧错误，接收模块必须先进行复位，并与发送端重新同步，方可成功接收下一帧。以下错误均归类为帧错误：

- 类型错误
- CRC 错误
- 帧结束错误

##### 20.3.3.3.1 使用 DMA 接收帧

FSI 接收器可连续接收数据，并通过 DMA 将数据从接收缓冲区传输出去。每当接收到一个数据帧时，就会生成一个 DMA 触发信号；该触发信号与设置 RX_EVT_STS 的 FRAME_DONE 信号同步。要使用 DMA 接收连续数据，需对接收

器进行一些配置：首先，将 RX_DMA_CTRL 中的 DMA_EVT_EN 设为 1，这允许 DMA 触发信号传递至 DMA 模块。此时，接收器可在接收到数据帧时触发 DMA 事件。此外，还需正确配置 DMA，以便 FSI 能够接收数据。利用接收器持续向 DMA 馈送数据的一种方法如下：

- 配置两个 DMA 通道，使其由 FSI 接收器的 DMA 触发信号触发。
- 配置其中一个 DMA 通道，将数据从接收缓冲区复制到更大的数据缓冲区。
- 配置下一个 DMA 通道，将接收到的数据帧标签和用户数据复制到另一个数据缓冲区。
- 由于 FSI 接收缓冲区是一个 16 字节的循环缓冲区，需确保负责该数据缓冲区的 DMA 通道在复制完 16 字节后能够完成环绕操作。

与发送器不同，对于处理数据缓冲区的 DMA 通道，并无要求其必须先于处理接收到的标签和用户数据的 DMA 通道执行。

#### 20.3.3.4 ping 帧看门狗

Ping 帧看门狗是一种硬件支持的自动错误检测机制，用于监测与发射器的连接状态。该看门狗会监控 ping 帧之间的时间间隔。如果发射器已设置为定期发送 ping 帧，则接收器可配置为在指定时间内监测是否已收到该帧。若 ping 帧之间的间隔时间超过预设的时钟周期数，则会触发一个事件，该事件可生成中断或由软件进行监控。此看门狗配备了一个专用计数器，在成功接收到 ping 帧时，该计数器会被重置并重新启动。看门狗计数器以 SYSCLK 的速率递增。可选地，可配置看门狗在成功接收到任何帧时重置——此选项允许接收器监测任何成功的帧，以表明连接仍保持有效且发射器仍在按预期正常工作。配置 Ping 帧看门狗以使其正常运行：

1.通过将 RX_PING_WD_CTRL.PING_WD_RST 设置为 1，随后将其清零，来重置 ping 看门狗计数器。

2.将 RX_OPER_CTRL.PING_WD_RST_MODE 设置为所需的看门狗重置事件：若需仅针对 ping 帧，则设置为 0；若需针对任意帧，则设置为 1。

3.将 RX_PING_WD_REF 设置为帧间最大时间间隔，并额外增加 10 个 SYSCLK 周期，以补偿时钟同步所需的时间。

4.通过将 RX_PING_WD_CTRL.PING_EN 设置为 1，启用 ping 看门狗功能。

Ping watchdog 现已启用，可对 Ping 帧进行监控。若 RX_PING_WD_CNT 的值达到 RX_PING_WD_REF 中预设的值，则 RX_EVT_STS.PING_WD_TO 标志将被置位；若已配置，则可在该事件发生时触发中断。

#### 20.3.3.5 帧看门狗

帧看门狗是接收器可用来监控任何错误状态的附加功能。该专用看门狗会监测单个帧被接收所需的时间。当接收器检测到正确的帧起始条件时，看门狗开始计数。若在预期的 SYSCLK 周期内未检测到帧结束条件，则帧看门狗将被触发，可生成中断或由软件进行监控。该看门狗会在帧起始和帧结束条件下分别自动启动和停止。帧看门狗与 SYSCLK 相连。配置帧看门狗以使其正常工作：

1.通过将 RX_FRAME_WD_CTRL 中的 FRAME_WD_CNT_RST 设置为 1，随后将该位清零，从而重置帧看门狗计数器。

2.将 RX_FRAME_WD_REF 中的 FRAME_WD_REF 设置为可接收的最长帧中预计存在的最大 SYSCLK 周期数，并额外增加 10 个 SYSCLK 周期以考虑时钟同步问题。

3.通过将 RX_FRAME_WD_CTRL 中的 FRAME_WD_CNT_EN 设置为 1，启用帧看门狗，使其能够检测到失败帧。

若 RX_FRAME_WD_CNT 达到 RX_FRAME_WD_REF 中编程的值，则 RX_EVT_STS 中的 FRAME_WD_TO 标志会被置位。若该标志已启用，则可在此事件上生成中断。若帧看门狗中断发生，接收器核心将处于无法接收新传输数据的无效状态；从帧看门狗超时状态中恢复的唯一方法是执行软复位，随后再与发送器重新同步。

### 20.3.3.6 延迟线控制

接收器模块在每个外部信号输入端—— RXCLK 、RXD0 和 RXD1——上均设有可编程延迟线。这些延迟元件会在相应线路中引入延迟，以方便调整由系统级组件（如信号缓冲器、铁氧体珠、隔离器等）或板级延迟（如走线长度不均、电缆过长等）引起的信号延迟。延迟长度可通过设置各线路的 RX_DLY_LINE_CTRL 寄存器值来控制。默认情况下，延迟线元件不会引入任何延迟。仅当 FSIRX 处于软复位状态时，方可调整延迟值，确保在此过程中没有正在进行的传输。图中展示了输入信号的延迟线电路结构示意图。对于 RXCLK 、RXD0 和 RXD1 的实现方式均与该图相同。所有电路的行为都相似。

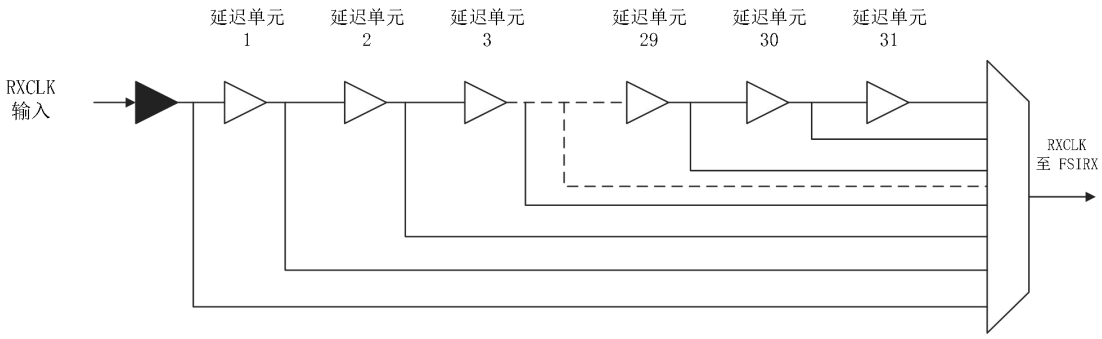



图 20-7 FSI 延迟线控制电路

### 20.3.3.7 缓冲区控制

FSI 接收器配备了一个 16 字节的缓冲区，当数据被接收后，数据会被复制到该缓冲区中。此缓冲区采用环形缓冲区实现，而非 FIFO 结构，因此必须谨慎处理缓冲区溢出、下溢以及 RX_BUF_PTR_STS 寄存器的状态。这些标志位和指针的运作基于软件或 DMA 将缓冲区视为环形缓冲区的假设。若接收器状态机进入错误状态，则软件无法干净地处理此情况，因为没有指定的接收时钟。为了使接收器能够检测到干净的重新同步，状态机需处于正常工作状态而非错误状态。从错误状态恢复的唯一方法是重置整个接收器模块。对于溢出和下溢情况，接收器将无法再验证缓冲区内数值的有效性。因此，最佳的恢复方法是重置 FSI 并与发送器重新同步。得益于接收缓冲区的灵活性，软件可以实现简单的乒乓缓冲区，或随机从缓冲区的任意位置接收和读

取数据。若按此方式使用缓冲区，则可忽略这些标志位和状态字段，而不会对接收器功能产生不利影响。此外，若以这种方式使用 CURR_WORD_CNT，则其值也将无效。应用程序可通过向 RX_BUF_PTR_LOAD 写入 4 位索引来手动设置缓冲区指针。这强制接收端从缓冲区中指定位置开始存储接收到的数据。

#### 20.3.3.8 CRC 冗余校验

接收模块会自动计算接收到的数据中的 CRC 值。接收到的 CRC 值会被存入`RX_CRC_INFO`中的`RX_CRC`字段。硬件对接收到的数据计算出的 CRC 值则会被存入`RX_CRC_INFO`中的`CALC_CRC`字段。这些值由硬件进行比较，若存在不匹配，则会设置`RX_EVT_STS`中的`CRC_ERROR`标志位。当启用时，接收器可以根据`RX_EVT_STS`中的`CRC_ERROR`触发中断。由于 CRC 仅用于数据帧，因此在 ping 帧和错误帧期间，`RX_CRC_INFO`中的`RX_CRC`和`RX_CRC_INFO`中的`CALC_CRC`值将无效。如果发送模块正在发送软件定义的 CRC 值（FSITX.TX_OPER_CTRL_LO.SW_CRC = 1），且接收到的 CRC 值与硬件计算出的值不匹配时，接收模块会触发 CRC 错误事件。由于这是应用程序级别的决策， FSIRX 可以安全地忽略该 CRC 错误事件。应用程序软件需要使用与发送端相同的自定义算法来计算并验证接收到的 CRC 值，并采取相应的措施。CRC 字段也可作为应用程序特定的值使用，而非 CRC 值。应用程序可根据需要使用`RX_CRC_INFO`中的`RX_CRC`。在此情况下，所有 CRC 错误和标志均可被忽略。

#### 20.3.3.9 利用接收器标签寄存器的零位

接收器标签寄存器（接收帧标签和用户数据（RX_FRAME_TAG_UDATA）寄存器以及接收器 ping 标签（RX_PING_TAG）寄存器）的最低有效位被设置为 0。实际接收到的标签位于第 4 至第 1 位的位置。这样做是为了方便用户软件创建一个函数表，可根据标签值调用相应的函数。函数指针需要占用 32 位存储空间，因此，每个后续的指针都会偏移 2 位。如果第一个指针位于地址 x，则第二个指针位于地址 x+2，第

三个指针位于地址 x+4，依此类推。通过将 LSB 设为 0，标签寄存器中的 5 位（第 4 至第 0 位）现在可以直接用作函数指针表的索引。

#### 20.3.3.10 接收器必须执行软复位的条件

接收器在每个时钟沿处接收数据。虽然存在特定的模式可确定帧的起始点并标示帧的结束点，但这些模式也可能出现在帧内正常运行过程中的任何时刻。若接收器在任何时刻未能检测到有效的帧，则必须对模块进行复位，以确保后续帧能够被正确接收。当接收到的帧中出现以下任一错误时，可能需要对接收器进行复位，并与发送端重新同步：

- 帧类型错误
- 帧结束错误
- Ping 帧看门狗超时
- 帧看门狗超时
- 由于时钟信号噪声干扰导致接收器处于无效状态。

可通过监测接收器核心状态（RX_VIS_1、RX_CORE_STS）来判断接收器核心是否已进入需执行软复位以恢复通信的错误状态；帧类型错误和帧结束错误均会触发该位置位，此类情况下需执行软复位。帧看门狗超时则必然要求进行复位，因为此时接收器状态机仍在等待更多信息。RX_CORE_STS 可用于判断导致帧失败的原因是否为噪声事件。Ping 帧看门狗不会导致 RX_CORE_STS 置位；与帧看门狗类似，接收器损坏并非导致 Ping 帧超时的原因——发送端可能已离线，且从未发送过 Ping 帧。或者，在空闲期间，可能会发生噪声事件，导致接收器进入损坏状态。由于接收器能够在 ping 帧看门超时中断处理程序中检测到此事件，此类事件不会丢失，应用程序可据此采取相应的措施。由于接收器由 RXCLK 而非 SYSCLK 时钟驱动，噪声干扰的时钟或数据线可能会导致某些内部设计约束被违反，从而使接收器核心逻辑进入未定义状态。请确保时钟和数据线满足设备数据手册中所述 FSI 模块的电气特性及时序要求。若未能满足这些要求，可能导致接收器状态机进入不可恢复的错误状态。

此时，只能通过执行软复位来恢复接收器。要确定接收器核心在意外帧错误后的状态，应用程序必须检查接收器核心的状态位。除了上述错误外，缓冲区溢出或下溢也可能需要执行软复位，以重新与本地应用程序软件同步。关于因溢出或下溢而需对接收器进行复位的要求，取决于应用程序的具体需求。当接收器被置于软复位状态后，应用程序必须通知另一端设备的发送器开始新的同步阶段。实现这一目标最简单的方法是发送带有指定标签的 ping 或错误帧。如果应用程序并未使用检测到错误的设备上的 FSITX ，则必须采用其他方法。另一端设备必须停止传输并开始新的同步阶段。

#### 20.3.3.11 FSI RX 复位

接收器模块及寄存器由 SYSRSn 复位；接收器核心则可通过 SYSRSn 或通过向 RX_MASTER_CTRL 寄存器写入“1”来复位。模块复位会将寄存器重置为默认状态；在执行模块复位后，必须对接收器模块进行重新初始化，并重建数据链路。

### 20.3.4 帧格式

FSI 模块以帧的形式传输和接收信息。每个帧包含多个相位，不同相位中可找到不同的信息。相位的数量及帧的总长度会根据所传输的帧类型而变化。帧的长度可能短至 16 位（用于 ping 或错误帧），也可能长达 288 位（用于 16 字数据帧）。在正常传输模式下，帧开始前有四个前导时钟边沿，帧结束后有四个后导时钟边沿。数据在两个时钟边沿上传输（双数据速率）。基本帧结构如表中所示。帧中的每个相位（如帧起始、帧类型等）均按最高有效位优先的方式传输。表中描述了 FSI 使用的基本帧结构，并根据所传输的帧类型进行了相应调整。

表 20-4 基本框架结构

| 空闲状态 | 先导序列 | 帧起始 | 帧类型 | 用户数据 | 用户字 | CRC | 帧标签 | 帧结束 | 后缀 | 空闲状态 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1111 | 1001 | 4bit | 8bit | 1-16word | 8bit | 4bit | 0110 | 1111 | |

FSI 还支持 FSI -SPI 兼容模式。SPI 兼容帧结构与标准 FSI 帧相似，但存在差异。

FSI 中的“one word”一词指代 16 位。术语“frame”与“packet”可互换使用，均用于描述 FSI 的信令格式。

### 20.3.4.1 FSI 帧结构

帧结构的各个阶段均有详细说明：

- 空闲状态：在空闲状态下，时钟线和数据线均被驱动至高电平，即非活动状态。
- 前导码：前导码阶段包含四个时钟沿（或两个完整的时钟脉冲），在此期间数据信号保持在高电平。这些时钟沿用于刷新接收器逻辑，并为接收器逻辑接收新帧做好准备。此阶段在 SPI 兼容模式下并不存在。
- 帧起始位：帧起始位阶段包含两个时钟脉冲，其上传输的四比特数据为“1001”。
- 帧类型：帧类型阶段包含两个时钟脉冲，其上传输的 4 位帧类型码通过数据线发送。发送器在传输帧之前必须设置 TX_FRAME_CTRL 结构体中的 FRAME_TYPE 字段；接收到的帧类型将存储于 RX_FRAME_INFO 结构体的 FRAME_TYPE 字段中。
- 用户数据：用户数据阶段包含一个完全可由用户配置的数据字段，该字段的使用方式不受任何限制。此阶段仅适用于数据帧。待传输的用户数据可通过向 TX_FRAME_TAG_UDATA 结构体的 USER_DATA 字段写入值来设定；接收到的用户数据则存储于 RX_FRAME_TAG_UDATA 结构体的 USER_DATA 字段中。
- 数据：数据阶段包含正在传输的数据。这些数据从发送器的发送缓冲区中提取，并存入接收器的接收缓冲区。Word 0 首先被传输。此阶段仅存在于数据帧中。根据传输的帧类型，该阶段可能包含 1 至 16 个字，具体取决于所选的帧类型。
- CRC 字节：CRC 字节包含传输数据的 CRC 值。该阶段中的值可根据 TX_OPER_CTRL_LO.SW_CRC 位来自硬件或软件。CRC 字节仅存在于数据帧中。
- 帧标签：帧标签包含 4 位用户自定义的帧标签。该字段在应用程序中的使用不受任何限制。发送器将此标签提供给数据帧的 TX_FRAME_TAG_UDATA.FRAME_TAG 位。Ping 帧则使用

TX_PING_TAG.TAG 中定义的标签。接收器可通过 RX_FRAME_TAG_UDATA.FRAME_TAG 访问接收到的帧标签。

- 帧结束符：帧结束符包含四个时钟沿，即四个比特，0110，通过数据线传输。
- 后缀：后缀包含四个额外的时钟沿，期间数据线保持高电平。后缀之后，时钟和数据线均被驱动至高电平（非活动状态）。此阶段在 FSI -SPI 兼容模式下不存在。

### 20.3.4.2 FSI 帧类型

FSI 硬件能够生成并处理多种预定义的帧类型。不同的帧类型可由应用程序用于指示不同类型的事件或向接收方传递不同信息。不同的帧类型会影响传输帧中包含哪些阶段和数据字段。表中简要概述了 FSI 所使用的各种帧类型。每种帧类型将在下文各小节中进行更详细的描述。

表 20-5 框架类型与 4 位编码

| 帧类型 | 编码 | 描述 |
|---|---|---|
| PING | 0000 | 通常用于检查线路完整性。Ping帧可由软件发送，亦可由硬件自动发送。 |
| ERROE | 1111 | 错误帧通常用于错误发生时，或任何一方希望向另一方发出注意信号的场合；不过，用户软件也可将错误帧用于任何其他目的。 |
| DATA_1_WORD | 0100 | 1字数据包（16位数据） |
| DATA_2_WORD | 0101 | 2字数据包（32位数据） |
| DATA_4_WORD | 0110 | 4字数据包（64位数据） |
| DATA_6_WORD | 0111 | 6字数据包（96位数据） |
| DATA_N_WORD | 0011 | N（1–16）字数据包，其中软件已将指定寄存器中的数据字数量编程设定；发送模块与接收模块均须配置相同的数值。 |
| 保留 | 0001, 0010, and 1000-1110 | 保留 |

#### 20.3.4.2.1 Ping 帧

Ping 帧是 FSI 可生成的最基础帧类型之一。表中展示了 Ping 帧的结构。

表 20-6 Ping 帧

| 空闲状态 | 前导码 | 帧起始（SOF） | 帧类型 | 帧标签 | 帧结束（EOF） | 后导码 | 空闲状态 |
|---|---|---|---|---|---|---|---|
| | 1111 | 1001 | 0000 | xxxx | 0110 | 1111 | |

Ping 帧类型始终为 0000。帧标签由应用程序定义。定时器和软件触发的 Ping 帧拥有独立的帧标签。Ping 帧中不传输任何数据或 CRC 校验码。Ping 帧的主要目的是定期向接收端发送通知，以确保发送端与接收端之间存在活跃连接。发送端和接收端核心均实现了不同的功能，以使 Ping 帧能够作为线路中断检测功能运行。在发送端，Ping 帧是唯一可直接建立并传输而无需任何额外软件或 DMA 干预的帧。Ping 帧可由以下三种来源（或全部）之一进行传输：自动 Ping 定时器、软件或外部触发源。接收端配备有 Ping 看门狗，可检测在预设窗口内是否未收到 Ping 帧，从而使得接收端能够判断接收端与发送端之间的连接是否已中断。

#### 20.3.4.2.2 错误帧

错误帧与 ping 帧相似，均不包含传输的数据字段。尽管该帧被命名为“错误帧”，但其具体用途由应用程序决定，因为对于此类帧的传输方式及传输时机，并未设定任何限制。表中展示了错误帧的结构。

表 20-7 错误帧

| 空闲状态 | 前导码 | 帧起始（SOF） | 帧类型 | 帧标签 | 帧结束（EOF） | 后导码 | 空闲状态 |
|---|---|---|---|---|---|---|---|
| | 1111 | 1001 | 1111 | xxxx | 0110 | 1111 | |

错误帧的结构与 ping 帧相同，不传输任何数据或 CRC 值。所有错误帧的帧类型均设为 1111，而帧标签则由 TX_FRAME_TAG_UDATA 寄存器中的软件定义。接收端可根据帧类型字段判断是否接收到错误帧；基于此特性，接收端可从 RX_FRAME_TAG_UDATA 寄存器中读取入站帧标签，并针对最多 16 种不同的条件采取相应处理措施。

#### 20.3.4.2.3 数据帧

数据帧是最复杂的帧类型。正如其名称所示，这类帧用于传输数据。表中展示了数据帧的通用结构。

表 20-8 数据帧

| 空闲状态 | 前导码 | 帧起始（SOF） | 帧类型 | 用户数据 | 数据字 | CRC 字节 | 帧标签 | 帧结束（EOF） | 后导码 | 空闲状态 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1111 | 1001 | 0xxx | xxxx xxxx | 1-16 words | xxxx xxxx | xxxx | 0110 | 1111 | |

帧类型字段用于表示帧类型的 4 位编码。帧类型的列表可参见表中。所传输的数据字数量由所选的帧类型决定。该帧类型支持四种固定长度的数据帧：1 个字、2 个字、4 个字和 6 个字。此外，还存在一种用户自定义数据长度帧类型，在此类型中，数据字的数量由软件进行固定配置；在此帧类型下，可传输 1 至 16 个字之间的任意数量数据。该长度需在发射机的 TX_FRAME_CTRL 寄存器中的 N_WORDS 字段以及接收机的 RX_OPER_CTRL 寄存器中进行配置。

### 20.3.4.3 多通道传输

为了实现更高的数据速率，FSI 可以同时使用 D0 和 D1 进行传输。此时与只使用 D0 传输的帧格式不同的是，数据帧中，奇数位置的比特由 TXD0 传输，偶数位置的比特由 TXD1 传输。多通道模式只能传输数据帧。

下面说明传输格式，以传输以下数据为例：

8-bit user data: u7u6u5u4u3u2u1u0

16-bit data: d15d14d13d12…d1d0

8-bit CRC: c7c6c5c4c3c2c1c0

数据在两条数据线上的分布如表所示。

表 20-9 多通道传输

| 空闲状态 | 前导码 | 帧起始（SOF） | 帧类型 | 用户数据 | 数据字 | CRC 字节 | 帧标签 | 帧结束（EOF） | 后导码 | 空闲状态 |
|---|---|---|---|---|---|---|---|---|---|---|
| TXD0 | 1111 | 1001 | 0011 | u7u5u3u1 | d15d13…d1 | c7c5c3c1 | xxxx | 0110 | 1111 | |
| TXD1 | 1111 | 1001 | 0011 | u6u4u2u0 | d14d12…d0 | c6c4c2c0 | xxxx | 0110 | 1111 | |

### 20.3.5 刷新序列

每当接收器执行软复位时，接收器需要从发送器处获取一个刷新序列，才能开始接收和解码帧。接收器核心具有异步复位机制，使得即使在没有接收时钟的情况下，也能对接收模块进行复位。然而，由于设计原因，此复位操作与接收时钟（RXCLK）同步释放。因此，接收器需要五个完整的时钟脉冲才能脱离复位状态。发送刷新模式可确保这些时钟边沿被正确接收，并能正确解析随后发送给接收器的任何帧。该刷新序列由两条数据线上的单次翻转以及时钟线上连续的五个脉冲组成。如果 FSI 接收器是从标准 SPI 接口接收数据，则来自 SPI 的数据字“0xFFFF”会产生与刷新序列相同的效果。

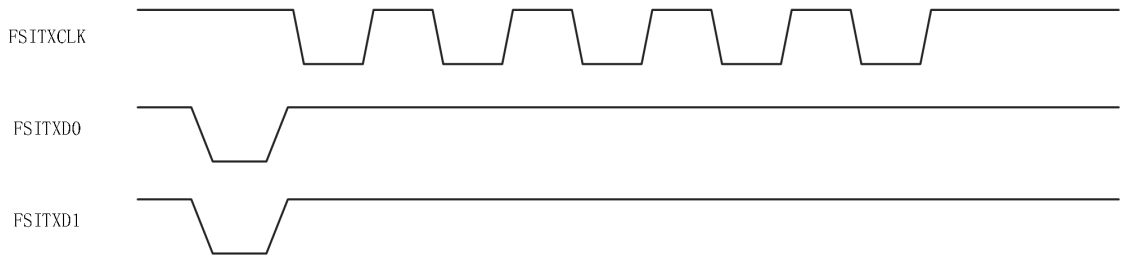



图 20-8 冲刷序列

### 20.3.6 内部自环

发射器与接收器核心可在内部相互连接，以便进行开发和调试。此功能可通过将`RX_MASTER_CTRL.INT_LOOPBACK` 设置为 1 来实现。内部环回功能可将来自相应发射器的信号路由至对应的接收器引脚；发射器无需进行任何配置。图中展示了采用内部环回的信号连接方式。

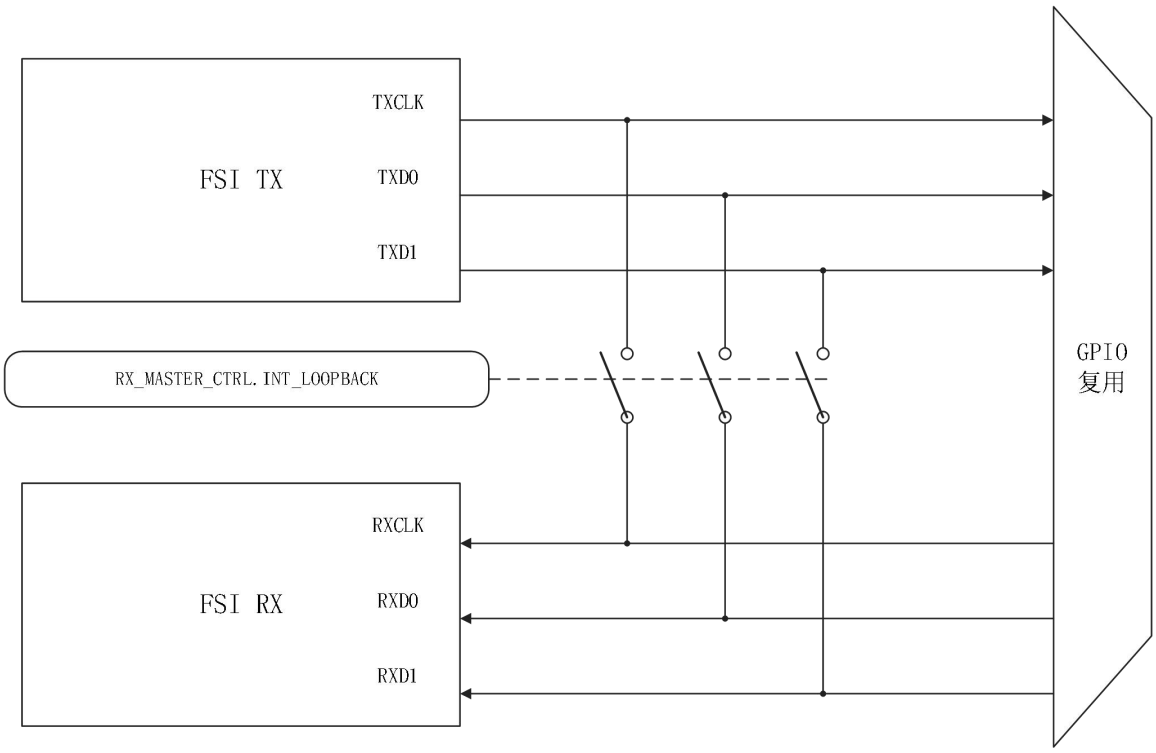



图 20-9 回环模式

## 20.3.7 CRC 产生

FSI 采用 CRC-8 算法，使用多项式 0x07 进行内部硬件 CRC 生成。该多项式也可表示为 $x^8 + x^2 + x + 1$。例如，对于一个包含 2 个字的数据包，计算过程如下：Data-1 = 0x4433；Data-0 = 0x2211；用户数据 = 0xAA。CRC 的计算时，按以下顺序取字节（从第一个到最后一个）：0xAA – 字节 0，用户数据；0x11 – 字节 1，Data-0，最低有效字节；0x22 – 字节 2，Data-0，最高有效字节；0x33 – 字节 3，Data-1，最低有效字节；0x44 – 字节 4，Data-1，最高有效字节。

## 20.3.8 ECC 模块

FSI 模块在发射器和接收器中均配备 16 位或 32 位 ECC 计算模块。使用此模块是可选的。请注意，ECC 独立于发射器和接收器核心中的硬件 CRC 计算模块。以下示例展示了应用程序需要计算并传输 ECC 值的情况：在 FSITX 模块中：

1.配置 ECC 模块以支持 32 位数据，设置 TX_OPER_CTRL_HI 中的 ECC_SEL 为 1。

2. 将数据写入 TX_ECC_DATA 寄存器及发送缓冲区。

3. 读取 TX_ECC_VAL 寄存器，该寄存器包含根据数据计算出的 8 位 ECC 值。

4. 将 8 位数据从 TX_ECC_VAL 复制到 TX_FRAME_TAG_UDATA 。USER_DATA。

5. 设置起始条件以开始传输。在 FSIRX 模块中，执行相反的操作流程。

一旦接收到数据帧，用户软件可以执行以下操作：

1.将数据从接收缓冲区复制到 RX_ECC_DATA 寄存器。

2.将包含已传输 ECC 值的接收用户数据从 RX_FRAME_TAG_UDATA 。USER_DATA 复制到 RX_ECC_VAL 寄存器。

3.读取 RX_ECC_LOG 寄存器，其中包含使用 RX_ECC_DATA 和 RX_ECC_VAL 寄存器进行 ECC 计算的结果。

a. 若未检测到 ECC 错误，则 RX_ECC_LOG 为 0，正确的数据位于 RX_ECC_SEC_DATA 中。

b. 若检测到单比特错误，则 RX_ECC_LOG.SBE 为 1，自动校正后的数据位于 RX_ECC_SEC_DATA 中。

c.若发生多处比特错误，则 RX_ECC_LOG.MBE 为 1；RX_ECC_SEC_DATA 中的数据无效，不得使用。

采用“2 字数据帧”结合利用用户数据进行纠错码（ECC）计算，是实现 ECC 检测的一种可能方案；另一种选择是采用更大的数据帧，并将其中的一个数据字分配给作为 ECC 值的位。

### 20.3.9 帧匹配

FSI 接收器核心具备在接收到的数据或 ping 帧的标签分别与 RX_FRAME_TAG_CMP 或 RX_PING_TAG_CMP 中的参考标签匹配时，生成中断的能力。每个标签比较寄存器不仅允许用户选择参考标签（TAG_REF），还可设置掩码（TAG_MASK）以忽略参考标签中指定的位。要启用标签匹配功能，必须置位

CMP_EN 位。当标签比较功能已启用且发生成功匹配时，RX_EVT_ERR_STATUS 寄存器中的 PING_TAG_MATCH、DATA_TAG_MATCH 或 ERROR_TAG_MATCH 位将被置位。相应的匹配寄存器状态可通过写入 RX_EVT_ERR_CLEAR 寄存器来清除。此外，若需通过软件设置匹配寄存器状态，用户必须通过写入 RX_EVT_ERR_SET 寄存器来实现。标签匹配方案并非过滤方案，而仅是一种通知机制，用于在数据或 ping 帧中检测到特定标签时向用户发出警报。可配置 RX_INTR_EVT_CTRL_1 和 RX_INTR_EVT_CTRL_2 中断，在标签匹配事件发生时触发中断。当启用标签匹配功能时，另一项可用特性是广播功能——可通过在 RX_FRAME_TAG_CMP 或 RX_PING_TAG_CMP 中置位 BRDCST_EN 来启用该功能。当广播模式启用时，标签的第三位被视为广播位：若接收到的标签的第三位被置位，则无论其他标签位是否匹配，该标签均被视为匹配。请注意，由 TAG_REF 和 TAG_MASK 引起的匹配也被视为匹配。需要注意的是，标签匹配方案并非过滤方案。例如，如果启用了标签匹配且接收到的帧其标签与匹配结果不符，该帧仍会被保存在缓冲区中，并相应地设置对应的事件状态位（FRAME_DONE、DATA_FRAME_RCVD）。在多从机 TDM 配置中需要使用标签匹配；然而，标签匹配也可根据需要用于单从机配置中。

### 20.3.10 TDM 配置

本设备中的 FSI 模块支持多从机 TDM 配置，而单个主设备可控制多个从设备。要使用上述多从机 TDM 配置中的 FSI 模块，从设备必须启用标签匹配功能并配合 CLB 模块使用。这种多从机 TDM 配置通过 FSI 模块的时分复用技术实现支持。当 TDM 功能启用时，每个从设备也必须启用标签匹配功能。图中展示了单个主设备与多个从设备通信的示意图：从设备中的所有 FSI 接收模块均直接连接至主设备的发送模块；从设备的发送模块则串联排列，使得每个发送模块均连接至下一个从设备，而最后一个从设备的输出端则连接至主设备的接收模块。每个从设备可根据接收到的数据帧的标签，决定是否传输数据或进入旁路模式——在此模式下，前一个从设

备的发送模块会直接连接至下一个从设备。这一过程通过使用 FSI 发送模块 TDM_IN 来实现。

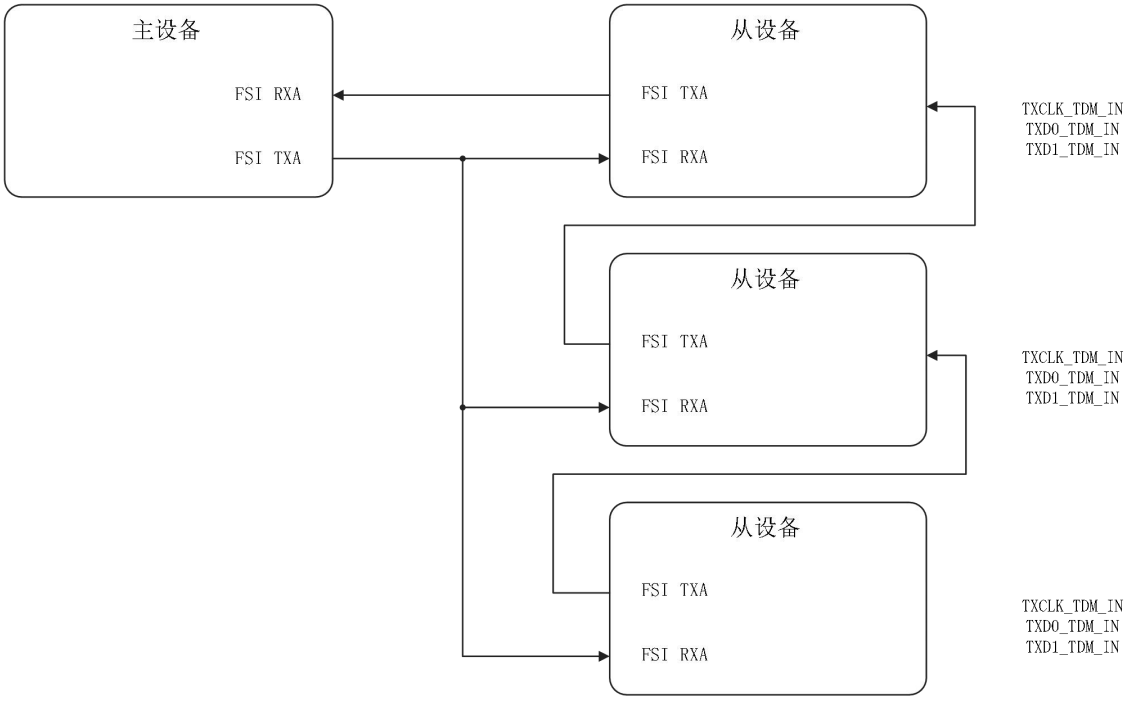



图 20-10 FSI 多从机 TDM 配置

当使用 FSI 发射器模块于 TDM 模式时，若需发射器进入旁路模式，则需使用 TXCLK_TDM_IN、TXD0_TDM_IN 和 TXD1_TDM_IN 引脚。图中展示了 FSI 模块在多从机 TDM 模式下的工作方式。

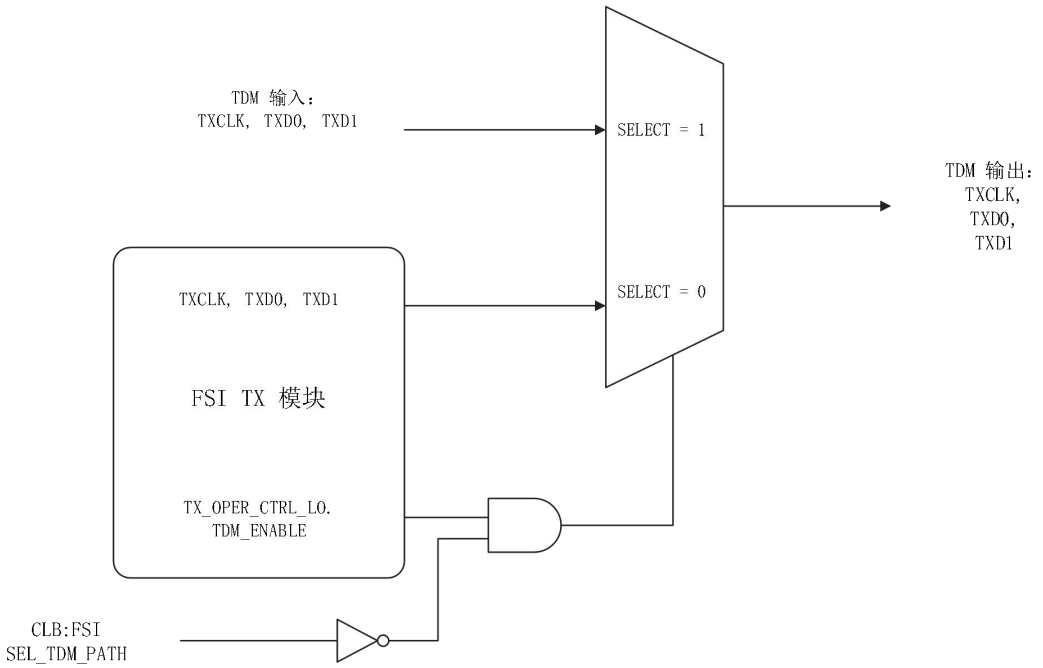



图 20-11 FSI 发射机多从机 TDM 复用

SEL_TDM_PATH 信号源自 CLB 模块。 CLB 模块还负责生成 FSI 发射器的发射触发信号。 CLB 模块需经过配置，以根据 FSI 接收器模块生成的数据状态、ping 信号及帧标签匹配情况来决定何时生成 FSI 发射触发信号。 FSITX 模块则需配置为在外部触发下进行传输，并需选择相应的 CLB 触发输入端口。在广播场景下（当 FSI 标签匹配通知所有从设备已发生匹配时），每个从设备内的 CLB 模块会生成触发信号和 SEL_TDM_PATH 信号。关键在于，触发信号与 SEL_TDM_PATH 信号必须在不同的时间间隔内生成，且两者之间不得重叠。图中展示了在广播场景下，由 RX_TRIGx 信号或从设备的 CLB 模块生成的 FSI 发射触发信号以及多从设备的 TDM SEL_TDM_PATH 信号的示例。

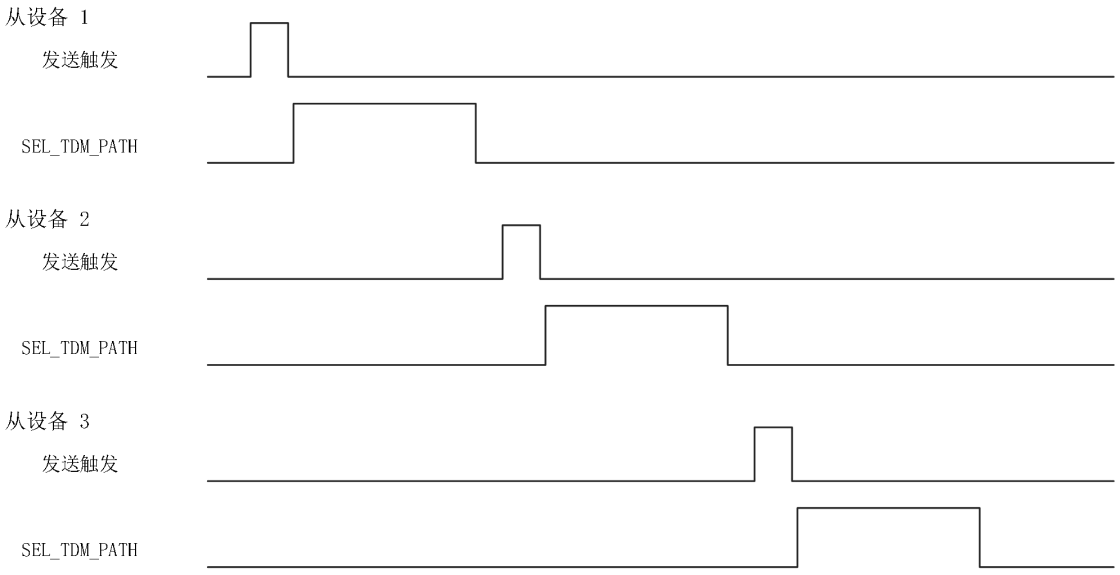



图 20-12 FSI 多从机 TDM 配置的生成信号

### 20.3.11 FSI 触发器生成

RX_TRIGx 外部触发器可用于启动 FSITX 传输。若需要 TDM 模式（多从设备配置），则必须使用 RX_TRIG0。在传输开始时，必须使用 RX_TRIG0 作为触发源；同时，可编程延展宽度的 RX_TRIG0 信号可作为 SEL_TDM_PATH 信号（该信号决定本地 FSITX 是否处于激活状态或进入旁路模式）。

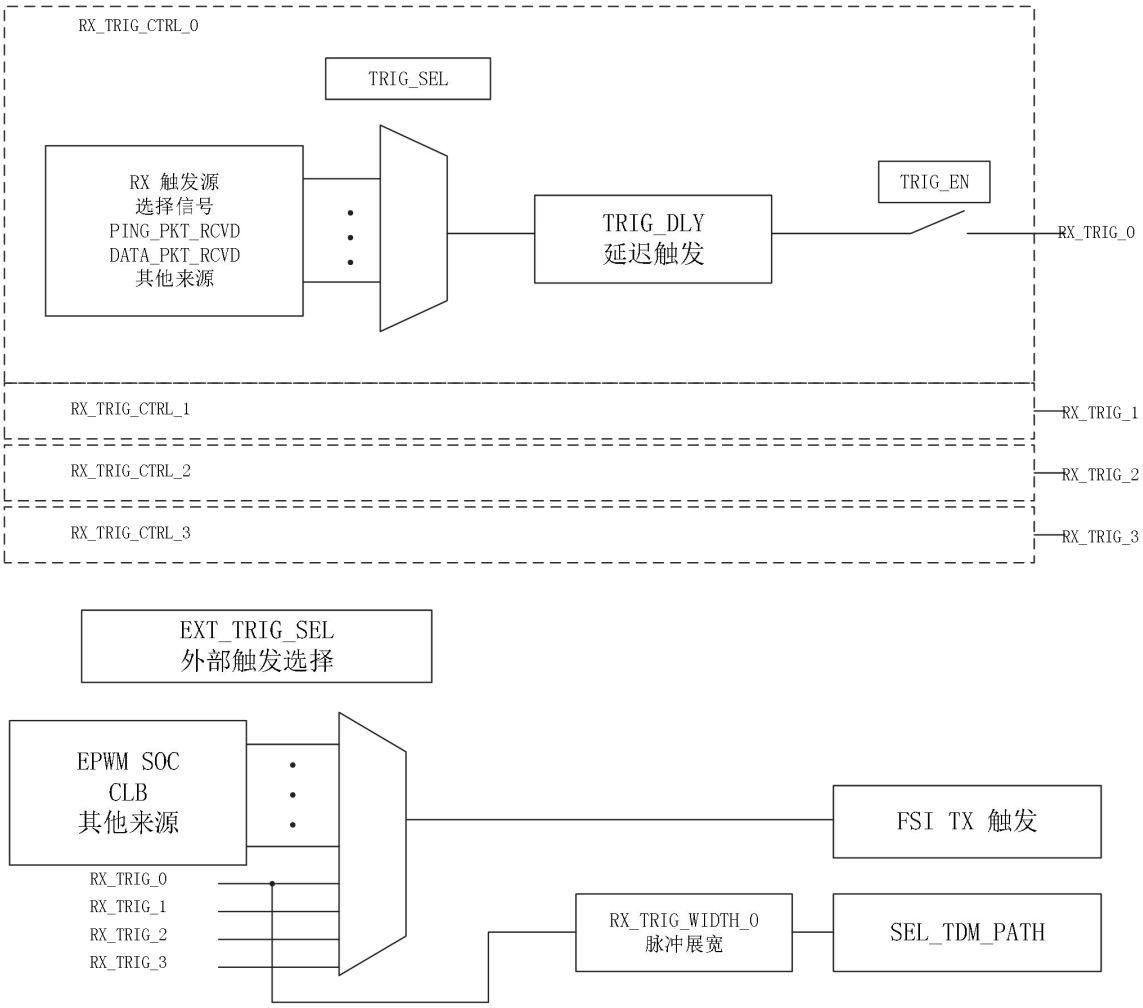



图 20-13 RX_TRIGx FSI 触发器

RX_TRIGx 信号的信号源可通过 RX_TRIG_CTRL_x 中的 TRIG_SEL 位进行选择，具体如表中所示。

表 20-10 RX_TRIGx 触发器选择信号

| RX_TRIG_CTRLx.TRIG_SEL | 信号选择 |
|---|---|
| 0 | 收到ping数据包 |
| 1 | 数据包接收 |
| 2 | 错误数据包接收 |
| 3 | Ping帧标签匹配发生 |
| 4 | 数据帧标签匹配发生 |

| RX_TRIG_CTRLx.TRIG_SEL | 信号选择 |
|---|---|
| 5 | 错误帧标签匹配发生 |
| 6 | 帧完成 |
| 7 | 保留 |
| 8-15 | 保留 |

RX_TRIGx 信号可选择性地进行延迟（此功能可用于 TDM 场景），该延迟可通过 RX_TRIG_CTRL_x 的 TRIG_DLY 寄存器实现。

### 20.3.12 SPI 兼容模式

FSI 支持 SPI 兼容模式。虽然 FSI 可以与标准 SPI 模块通信，但 FSI 仅支持有限的配置选项。该兼容模式的功能如下：

- 数据在时钟的上升沿发送，在下降沿接收。
- 仅支持 16 位字长。
- TXD1 作为活动低电平的片选信号驱动，该信号在整个帧传输期间保持低电平。
- 不需要接收器片选输入，RXD1 不被使用。数据在每个有效的时钟边沿移入接收器。
- 不传输前导码或后导码。所有信号在帧阶段结束后均返回空闲状态。
- 在 SPI 从机配置下无法进行数据传输，因为 FSI 的 TXCLK 无法接受外部时钟源。表中列出了 FSI -SPI 兼容模式的帧结构。此模式中包含每个帧阶段。如果 FSI 向标准 SPI 模块发送数据，SPI 必须解码该帧结构；同样，如果 FSI 被配置为 SPI 从机，则标准 SPI 必须对要发送的数据进行编码。

表 20-11 SPI 兼容模式帧

| 空闲状态 | 帧起始（SOF） | 帧类型 | 用户数据 | 数据字 | CRC 字节 | 帧标签 | 帧结束（EOF） | 空闲状态 |
|---|---|---|---|---|---|---|---|---|
| | 1001 | 4bits | 8bits | 1-16 words | 8bits | 4bits | 0110 | |

由于标准 SPI 模块需对各种帧数据进行编码，这限制了可连接至 FSI 的 SPI 模式下使用的模块类型。配对的 SPI 模块必须具备足够的功能来对帧进行编码和解码。若 FSI 传输至标准 16 位 SPI 接口，则数据将以以下方式排列。表中提供的示例假设已发送了 DATA_2_WORD 帧。

表 20-12 标准 SPI 接收数据的内容

| SPI 数据 | 数据内容 |
|---|---|
| spi word 0 | 1001, 0100, 8-bit User Data |
| spi word 1 | Data word 1 |
| spi word 2 | Data word 2 |
| spi word 3 | 8-bit CRC, 4-bit Frame Tag, 0110 |

### 20.3.12.1 可用 SPI 模式

FSI 在与 SPI 模块通信时，可采用几种不同的布线方案。

#### 20.3.12.1.1 FSITX 作为 SPI 主控器，仅支持发送功能：

FSITX 可作为独立的 SPI 主控模块运行。在此模式下， TXCLK 连接至 SPICLK ，TXD0 连接至 spisimo，TXD1 连接至芯片选择引脚 SPISTE 。

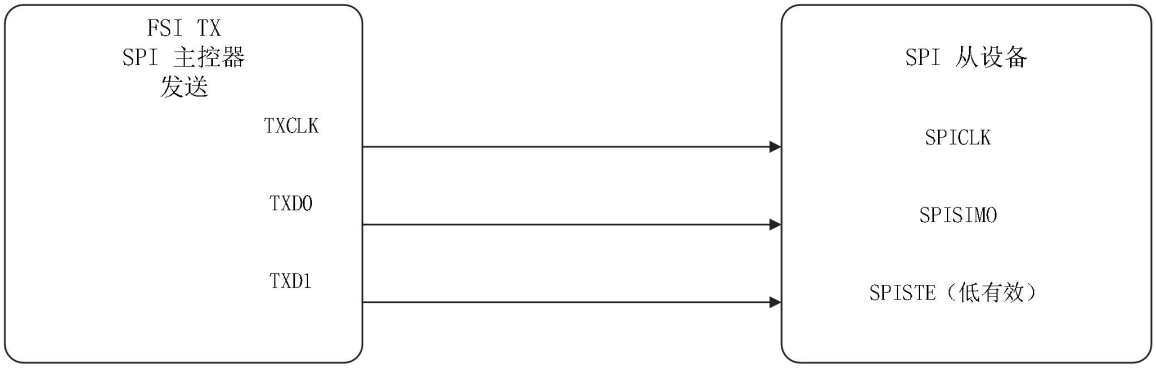



图 20-14 FSITX 作为 SPI 主控器，仅发送模式

当 FSI 为 SPI 发射器时，应用程序可检查帧错误、行中断、CRC 错误及数据的 ECC 校验。这些信息均由每个 FSI 帧中的硬件进行编码。SPI 接收器则需借助软件来处理这些信息。

表 20-13 FSI 作为主发射器，SPI 作为从接收器

| 作用 | 可用行 | 描述 |
| --- | --- | --- |
| 数据帧校 | 是 | 可在SPI接收器的软件中实现 |
| 行分隔符检测 | 是 | 可在SPI接收器的软件中实现，但需要额外的软件开销，如定时器或看门狗。 |
| CRC校验 | 是 | 可在SPI接收器的软件中实现。对于具有 VCRC 功能的设备，此方法更为高效 |
| 数据ECC校验 | 是 | 可在SPI接收器的软件中实现。 |
| 突发终止帧检测 | 否 | |
| 双边数据速率 | 否 | |
| 帧间信号线上的噪声恢复 | 否 | |
| 信号线上的时序偏移调整 | 否 | |

#### 20.3.12.1.1.1 初始化

若要将 FSITX 模块配置为仅作为 SPI 主设备进行发送操作，请按照标准的 FSITX 初始化流程进行。在释放 FSITX 从复位状态之前，需将 TX_OPER_CTRL_LO.SPI_MODE 设置为 1，此操作可启用 SPI 时钟方案及信号结构。

#### 20.3.12.1.1.2 操作

在 FSI -SPI 兼容模式下， FSITX 模块的运行方式与该模块处于标准 FSI 模式时相同。应用程序可利用帧定时器、ping 帧、外部帧触发器等功能。

### 20.3.12.1.2 FSIRX 作为 SPI 从设备，仅接收

FSIRX 可作为独立的 SPI 从模块运行。在此配置下， RXCLK 连接至 SPICLK ，RXD0 连接至 spisimo，而 RXD1 则未使用。当连接到 FSIRX 时，无需使用片选信号。这是因为 FSIRX 能够响应任何输入时钟边沿。若存在噪声或不希望出现的时钟跳变，则需执行清空序列，以使 FSIRX 模块与主设备重新同步。

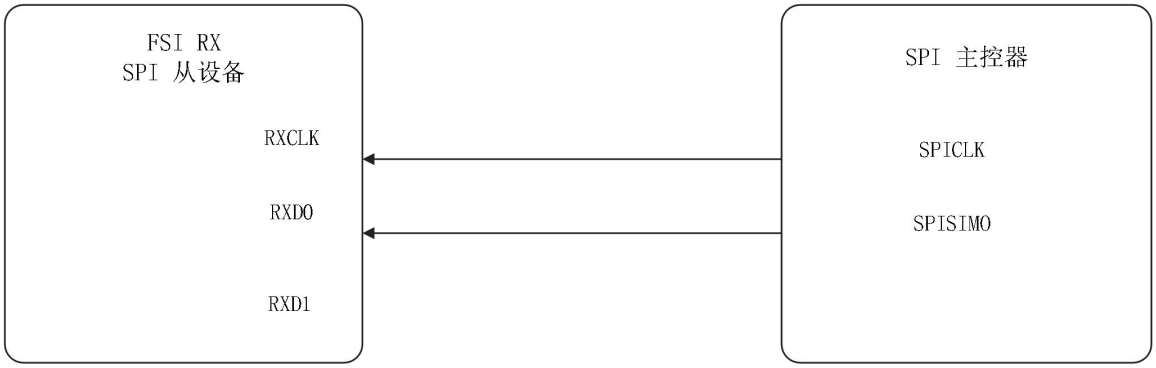



图 20-15 FSIRX 作为 SPI 从设备，仅接收模式

当 FSI 作为 SPI 接收器与 SPI 发送器通信时，应用程序能够检测帧错误、线路中断、CRC 错误、数据的 ECC 校验以及突发终止的帧。需要注意的是， FSI 可以硬件层面处理所有这些情况，但 SPI 发送器必须将这些信息编码到待传输的数据中。

表 20-14 SPI 作为主发射器， FSI 作为从接收器

| 作用 | 可用行 | 描述 |
|---|---|---|
| 数据帧校验 | 是 | 可在SPI接收器的软件中实现 |
| 行分隔符检测 | 是 | 可在SPI接收器的软件中实现，但需要额外的软件开销，如定时器或看门狗。 |
| CRC校验 | 是 | 可在SPI接收器的软件中实现。对于具有 VCRC 功能的设备，此方法更为高效 |
| 数据ECC校验 | 是 | 可在SPI接收器的软件中实现。 |
| 突发终止帧检测 | 是 | 此功能通过 FSI 设置帧看门狗计数器来实现 |
| 双边数据速率 | 否 | |
| 帧间信号线上的噪声恢复 | 是 | 当传输之间的时间点上，时钟线或数据线出现噪声时，帧的初始清空模式会忽略这些噪声的影响，并使接收端在看到真实的“帧起始”模式时重新同步；因此，该系统对帧间噪声的抑制能力非常高 |
| 信号线上的时序偏移调整 | 是 | FSI 接收器能够对输入信号线添加延迟。 |

#### 20.3.12.1.2.1 初始化

若要配置 FSIRX 模块作为仅接收的 SPI 从设备，请按照标准的 FSIRX 初始化流程进行操作。在将 FSIRX 从复位状态释放之前，需将 RX_OPER_CTRL 中的 SPI_MODE 设置为 1，此操作可启用 SPI 时钟方案及信号结构。

#### 20.3.12.1.2.2 操作

FSIRX 模块在 FSI -SPI 兼容模式下的运行方式与该模块处于标准 FSI 模式时相同。应用程序可利用帧监控器、Ping 监控器、CRC 校验和 ECC 校验等功能。

### 20.3.12.1.3 FSITX 与 FSIRX ：模拟全双工 SPI 主设备

在这一配置中， FSITX 是主时钟。 FSITX 模块驱动 SPI 从设备上的 TXCLK（SPICLK）、TXD0（spisimo）和 TXD1（SPISTE /芯片选择）。spisimo 信号会连接回 RXD0 信号。 RXCLK 可以通过内部 SPI 配对功能或外部布线来应用，具体取决于应用需求。由于 FSITX 和 RX 模块是独立的， FSIRX 也可以被视为一个额外的 SPI 从设备。为了使 FSI 完全模拟 SPI 主机，需要一些软件逻辑支持。

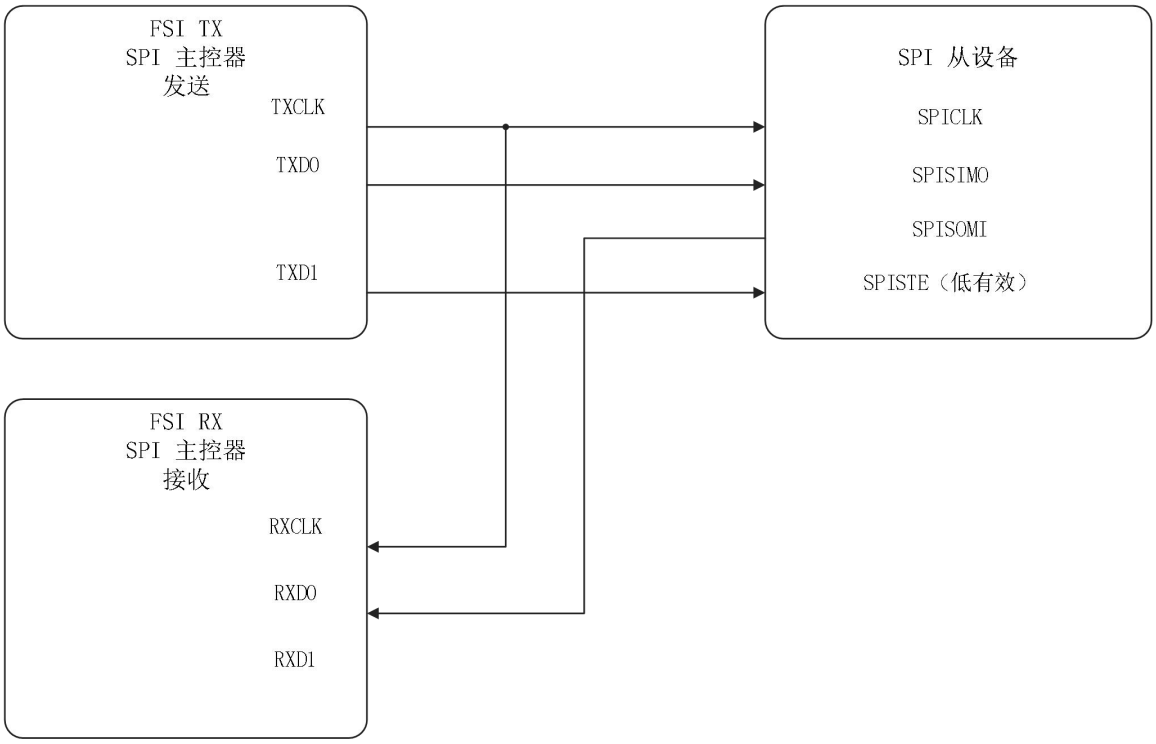


图 20-16 FSITX 与 FSIRX 作为 SPI 主控器，全双工模式

#### 20.3.12.1.3.1 初始化

要配置 FSITX 和 RX 模块以实现全双工 SPI 主模式操作，需遵循前文各章节中描述的每个模块的初始化说明。 FSITX 和 RX 模块均须设置相应的 SPI_MODE 位，这可启用 SPI 时钟方案及信号结构。若需要内部时钟环回，则 FSIRX 模块还需将 RX_MASTER_CTRL 设置为 1，从而在内部将 TXCLK 与 RXCLK 连接起来。若使用内部时钟环回，用于 RXCLK 的 GPIO 可重新分配给其他应用需求。若应用需要外部时钟环回，则需确保 TXCLK 已连接至 RXCLK 。此步骤在 SPI 从设备位于隔离屏障另一侧且 TXCLK 启动与 RXD0 上接收到 spisomi 数据之间存在延迟的情况下是必需的。

#### 20.3.12.1.3.2 操作

在此运行模式下，需编写部分更高层级的软件来模拟完整的 SPI 主模块。发送模块无法获知接收模块所接收到的数据；但 TX 与 RX 模块仍可利用其提供的各项其他功能，例如：ping 帧定时器、ping 帧及帧看门狗、CRC 与 ECC 错误检测器等。关于这些功能的配置流程，本文件另有章节予以说明。

## 20.4 编程指南

本节介绍了 FSI 的各类操作流程与功能特性。

### 20.4.1 建立通信链路

一旦发射器和接收器模块完成配置，它们在交换数据之前必须进行一定的同步操作。由于接收器可以在任何时钟跳变时接收数据，因此接收器核心逻辑需要被刷新，以正确解析新有效帧的起始点。这尤其适用于 FSI 模块位于不同的设备上且可能彼此隔离的情况。以下示例提供了一种在两个独立设备上建立干净通信链路的方法，这些设备可以按任意顺序启动。请注意，这仅是示例中的同步方法。根据具体应用需求，可以采用不同的方法。同步过程中最关键的一点是确保接收器能够被正确刷新，并准

备好接收完整的帧而无错误。如何实现这一点取决于具体应用。图中展示了本示例中设备的连接方式。尽管 FSI 协议中没有真正的主节点或从节点概念，但本示例使用这一术语作为描述数据流的一种简单方式。

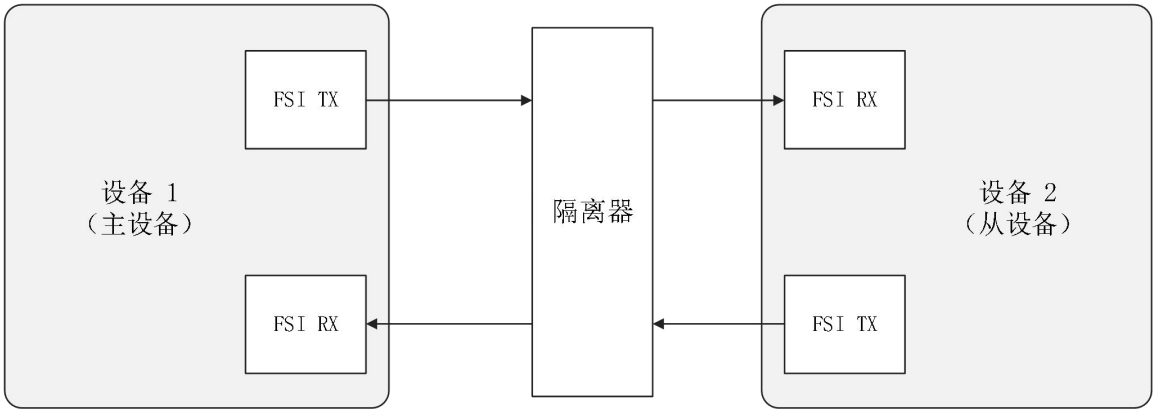



图 20-17 点对点连接

设备 1 为主节点，负责驱动初始化序列的执行；设备 2 为从节点，用于响应主节点的命令。在本示例及实际应用场景中，无论是主节点还是从节点，均无法准确获知对方何时已准备好接收通信。主节点与从节点的示例序列将在下述小节中予以说明。

#### 20.4.1.1 建立主设备与通信链路

以下序列展示了主设备节点如何在标准通信链路之外无需外部信号的情况下与从设备建立通信链接。

1.对 FSITX 和 FSIRX 模块均施加核心复位，然后解除复位。

2. 配置发射器和接收器以实现所需操作。

3. 设置接收中断，用于检测传入的传输。

4. 开始 ping 循环：

- 发送清空序列。
- 发送带有帧标签“0000”的 ping 帧。
- 等待一段时间（由应用程序决定）。
- 如果 FSIRX 已收到有效的 ping 帧，则继续；否则，再次循环该过程。
- 如果接收到的 ping 帧标签为“0001”，则继续；否则，再次循环该过程。

发送带有帧标签“0001”的 ping 帧。此时，主设备的发送和接收通道均已成功从从设备接收到了帧。通信链路已建立，标准应用通信可开始进行。

#### 20.4.1.2 建立从设备的通信链路

以下序列展示了从属设备节点如何在标准通信链路之外无需外部信号的情况下与主设备建立通信链接。

1.对 FSITX 和 FSIRX 模块均执行核心复位，然后释放复位。

2.配置发射器和接收器以实现所需操作。

3.设置接收器中断，以检测到来自发送端的传输。

4.等待接收器中断。

5.如果 FSIRX 已接收到有效的 ping 帧，则继续；否则，返回步骤 4。

6.若接收到的帧标签为 0000，则继续；否则，丢弃该传输并返回步骤 4。

7.发送清空序列。

8.发送一个帧标签为 0001 的 ping 帧。

9.等待接收器中断。

10.如果 FSIRX 已接收到有效的 ping 帧，则继续；否则，返回步骤 4。

11.若接收到的 ping 帧标签为 0001，则继续；否则，若接收到的帧标签为 0000，则返回步骤 9。这种情况可能发生在步骤 8 中尚未收到从属设备响应之前，第二 ping 帧已处于传输过程中。

此时，发射器和接收器模块均已成功从主设备端接收到 ping 帧。通信链路已建立，常规通信可开始进行。应用程序可配置发射器发送周期性 ping 帧，初始化接收器的 ping 帧和帧看门狗，并启动应用程序所需的通信流程。

## 20.5 寄存器描述

### 20.5.1 寄存器总结

#### 20.5.1.1 FSI 基地址表

表 20-15 FSI 基地址表

| 位域名称 | | DriverLib 名称 | 基地址 | 流水线保护 |
|---|---|---|---|---|
| 实例 | 结构 | | | |
| FsiTxaRegs | FSI_TX_REGS | FSITXA_BASE | 0x0062_3000 | 是 |
| FsiRxaRegs | FSI_RX_REGS | FSIRXA_BASE | 0x0062_3100 | 是 |

#### 20.5.1.2 FSITX 寄存器

表中列出了 FSI_TX_REGS 寄存器的内存映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，不应修改寄存器内容。

表 20-16 FSITX 寄存器

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 0x0000 | TX_MASTER_CTRL | 发送主控制寄存器 |
| 0x0004 | TX_CLK_CTRL | 发送时钟控制寄存器 |
| 0x0008 | TX_OPER_CTRL_LO | 发送操作控制寄存器低位 |
| 0x000A | TX_OPER_CTRL_HI | 发送操作控制寄存器高位 |
| 0x000C | TX_FRAME_CTRL | 发送帧控制寄存器 |
| 0x000E | TX_FRAME_TAG_UDATA | 发送帧标签和用户数据寄存器 |
| 0x0010 | TX_BUF_PTR_LOAD | 发送缓冲区指针控制加载寄存器 |
| 0x0012 | TX_BUF_PTR_STS | 发送缓冲区指针控制状态寄存器 |
| 0x0014 | TX_PING_CTRL | 发送 ping 控制寄存器 |
| 0x0016 | TX_PING_TAG | 发送 ping 标签寄存器 |
| 0x0018 | TX_PING_TO_REF | 发送 ping 超时计数器参考值寄存器 |
| 0x001C | TX_PING_TO_CNT | 发送 ping 超时当前计数值寄存器 |
| 0x0020 | TX_INT_CTRL | 发送中断事件控制寄存器 |
| 0x0022 | TX_DMA_CTRL | 发送 DMA 事件控制寄存器 |
| 0x0024 | TX_LOCK_CTRL | 发送锁定控制寄存器 |
| 0x0028 | TX_EVT_STS | 发送事件和错误状态标志寄存器 |
| 0x002C | TX_EVT_CLR | 发送事件和错误清除寄存器 |
| 0x002E | TX_EVT_FRC | 发送事件和错误标志强制寄存器 |
| 0x0030 | TX_USER_CRC | 发送用户定义 CRC 寄存器 |
| 0x0040 | TX_ECC_DATA | 发送 ECC 数据寄存器 |
| 0x0044 | TX_ECC_VAL | 发送 ECC 值寄存器 |
| 0x0080 + y ×2 (y = | TX_BUF_BASE_y | 发送缓冲区基地址寄存器 |

| 偏移量 | 缩写 | 寄存器名称 |
| --- | --- | --- |
| 0..15) | | |

### 20.5.1.3 FSIRX 寄存器

表中列出了 FSI_RX_REGS 寄存器的内存映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，不应修改寄存器内容。

表 20-17 FSIRX 寄存器

| 偏移量 | 缩写 | 寄存器名称 |
| --- | --- | --- |
| 0x0000 | RX_MASTER_CTRL | 接收主控制寄存器 |
| 0x0008 | RX_OPER_CTRL | 接收操作控制寄存器 |
| 0x000C | RX_FRAME_INFO | 接收帧信息寄存器 |
| 0x000E | RX_FRAME_TAG_UDATA | 接收帧标签和用户数据寄存器 |
| 0x0010 | RX_DMA_CTRL | 接收 DMA 事件控制寄存器 |
| 0x0014 | RX_EVT_STS | 接收事件和错误状态标志寄存器 |
| 0x0016 | RX_CRC_INFO | 接收 CRC 信息寄存器 |
| 0x0018 | RX_EVT_CLR | 接收事件和错误清除寄存器 |
| 0x001A | RX_EVT_FRC | 接收事件和错误标志强制寄存器 |
| 0x001C | RX_BUF_PTR_LOAD | 接收缓冲区指针加载寄存器 |
| 0x001E | RX_BUF_PTR_STS | 接收缓冲区指针状态寄存器 |
| 0x0020 | RX_FRAME_WD_CTRL | 接收帧看门狗控制寄存器 |
| 0x0024 | RX_FRAME_WD_REF | 接收帧看门狗计数器参考值寄存器 |
| 0x0028 | RX_FRAME_WD_CNT | 接收帧看门狗当前计数值寄存器 |
| 0x002C | RX_PING_WD_CTRL | 接收 ping 看门狗控制寄存器 |
| 0x002E | RX_PING_TAG | 接收 ping 标签寄存器 |
| 0x0030 | RX_PING_WD_REF | 接收 ping 看门狗计数器参考值寄存器 |
| 0x0034 | RX_PING_WD_CNT | 接收 ping 看门狗当前计数值寄存器 |
| 0x0038 | RX_INT1_CTRL | 接收中断 1 控制寄存器 |
| 0x003A | RX_INT2_CTRL | 接收中断 2 控制寄存器 |
| 0x003C | RX_LOCK_CTRL | 接收锁定控制寄存器 |
| 0x0040 | RX_ECC_DATA | 接收 ECC 数据寄存器 |
| 0x0044 | RX_ECC_VAL | 接收 ECC 值寄存器 |
| 0x0048 | RX_ECC_SEC_DATA | 接收 ECC 校正数据寄存器 |
| 0x004C | RX_ECC_LOG | 接收 ECC 日志和状态寄存器 |
| 0x0050 | RX_FRAME_TAG_CMP | 接收帧标签比较寄存器 |
| 0x0052 | RX_PING_TAG_CMP | 接收 ping 标签比较寄存器 |
| 0x0060 | RX_DLYLINE_CTRL | 接收延迟线控制寄存器 |
| 0x0070 | RX_VIS_1 | 接收调试可见性寄存器 1 |

### 20.5.1.4 FSI 访问类型编码

复杂的位访问类型经过编码以适应小型表格单元格。表中显示了本节中用于访问类型的编码。

表 20-18 FSI 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R-0 | 读 返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1S | W1S | 写 1 置位 |
| WSonce | WSonce | 一次性写（仅能置位一次） |
| 复位或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i,k,l,m,n | | 当这些变量用于寄存器名称、偏移地址或地址时，指代寄存器数组的值，该寄存器是一组重复寄存器的一部分。寄存器组形成层次结构，数组以公式表示。 |
| y | | 当该变量用于寄存器名称、偏移地址或地址时，指代寄存器数组的值。 |

### 20.5.1.5 FSITX 寄存器字段说明

#### 20.5.1.5.1 TX_MASTER_CTRL 寄存器

偏移地址：0h　复位值：0000h

TX_MASTER_CTRL 寄存器的各字段说明如表中所示。

表 20-19 TX_MASTER_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| 写（W）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | | | FLUSH | CORE_RST |
| 读（R）-0h | | | | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |

表 20-20 TX_MASTER_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-8 | KEY | 写（W） | 0h | 写保护。写本寄存器其他两位时，必须向此处写入0xA5，当此处为0xA5时写入才是有效的。<br>复位类型：SYSRSn |
| 7-2 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 1 | FLUSH | 读 / 写（R/W） | 0h | 0：清除<br>1：启动冲刷序列。设置1后应保持5个TXCLK再清0<br>复位类型：SYSRSn |
| 0 | CORE_RST | 读 / 写（R/W） | 0h | 0：撤销复位<br>1：复位<br>复位类型：SYSRSn |

### 20.5.1.5.2 TX_CLK_CTRL 寄存器

偏移地址：4h　复位值：0000h

TX_CLK_CTRL 寄存器的各字段说明如表中所示。

表 20-21 TX_CLK_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | PRESCALE_VAL | |
| 读（R）-0h | | | | | | 读 / 写（R/W）-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRESCALE_VAL | | | | | | CLK_EN | CLK_RST |
| 读 / 写（R/W）-0h | | | | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |

表 20-22 TX_CLK_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-10 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 9-2 | PRESCALE_VAL | 读 / 写（R/W） | 0h | 时钟预分频量。TXCLKIN=clk_in/ PRESCALE_VAL |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
| --- | --- | --- | --- | --- |
| | | | | 0：1<br>1：1<br>……<br>FF：255<br>复位类型：SYSRSn |
| 1 | CLK_EN | 读 / 写（R/W） | 0h | 时钟分频使能。时钟是经过该分频器到达TX core的，如果此位为0，则TX core没有时钟。<br>0：使能无效<br>1：分频和时钟有效<br>复位类型：SYSRSn |
| 0 | CLK_RST | 读 / 写（R/W） | 0h | 0：分频器依据PRESCALE_VAL和CLK_EN进行启动<br>1：分频器被复位为0，且保持复位直到软件向该位写入0<br>复位类型：SYSRSn |

### 20.5.1.5.3 TX_OPER_CTRL_LO 寄存器

偏移地址：8h　复位值：0000h

TX_OPER_CTRL_LO 寄存器的各字段说明如表中所示。

表 20-23 TX_OPER_CTRL_LO 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 保留（RESERVED） | | | | | | TDM_ENABLE | SEL_PLLCLK |
| 读（R）-0h | | | | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PING_TO_MODE | SW_CRC | START_MODE | | | SPI_MODE | DATA_WIDTH | |
| 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | |

表 20-24 TX_OPER_CTRL_LO 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-10 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 9 | TDM_ENABLE | 读 / 写（R/W） | 0h | 发送 TDM 模式使能位。0：不使能发送 TDM 模式；1：使能发送 TDM 模式，用于多远程 TDM 操作。<br>复位类型：SYSRSn |
| 8 | SEL_PLLCLK | 读 / 写（R/W） | 0h | 0：SYSCLK作为prescaler的输入时钟<br>1：PLLRAWCLK作为prescaler的输入时钟<br>复位类型：SYSRSn |
| 7 | PING_TO_MODE | 读 / 写（R/W） | 0h | 0：Ping定时器只会根据定时器超时而重置<br>1：Ping定时器会根据定时器超时和软件触发而重置<br>复位类型：SYSRSn |
| 6 | SW_CRC | 读 / 写（R/W） | 0h | 0：发送的CRC为硬件计算的值<br>1：发送的CRC为TX_USER_CRC寄存器中存入的值<br>复位类型：SYSRSn |
| 5-3 | START_MODE | 读 / 写（R/W） | 0h | 0：只有软件写TX_FRAME_CTRL.START时初始化新发送<br>1：外部触发上升沿会初始化新发送，外部触发信号应保持至少3个SYSCLK周期<br>2：写TX_FRAME_CTRL.START或TX_FRAME_TAG_UDATA都会初始化新发送<br>复位类型：SYSRSn |
| 2 | SPI_MODE | 读 / 写（R/W） | 0h | 0：FSI模式<br>1：SPI兼容模式<br>复位类型：SYSRSn |
| 1-0 | DATA_WIDTH | 读 / 写（R/W） | 0h | 0：只使用TXD0<br>1：使用TXD0和TXD1传输<br>其他值保留<br>复位类型：SYSRSn |

### 20.5.1.5.4 TX_OPER_CTRL_HI 寄存器

偏移地址：Ah　复位值：0000h

TX_OPER_CTRL_HI 寄存器的各字段说明如表中所示。

表 20-25 TX_OPER_CTRL_HI 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | EXT_TRIG_SEL | | | | |
| 读（R）-0h | | | 读 / 写（R/W）-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EXT_TRIG_SEL | ECC_SEL | FORCE_ERR | 保留（RESERVED） | | | | |
| 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读（R）-0h | | | | |

表 20-26 TX_OPER_CTRL_HI 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-13 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 12-7 | EXT_TRIG_SEL | 读 / 写（R/W） | 0h | 决定32个外部输入的哪一个会作为外部触发（普通帧）<br>0：触发1作为触发源<br>……<br>1F：触发32作为触发源<br>复位类型：SYSRSn |
| 6 | ECC_SEL | 读 / 写（R/W） | 0h | 0：32bit ECC<br>1：16bit ECC<br>复位类型：SYSRSn |
| 5 | FORCE_ERR | 读 / 写（R/W） | 0h | 当SW_CRC为1时不要使用此模式，否则会产生冲突。<br>0：CRC不强制为0<br>1：在buffer过载和欠载情况下，CRC会强制为0<br>复位类型：SYSRSn |
| 4-0 | 保留（RESERVED） | 读（R） | 0h | 保留 |

### 20.5.1.5.5 TX_FRAME_CTRL 寄存器

偏移地址：Ch　复位值：0000h

TX_FRAME_CTRL 寄存器的各字段说明如表中所示。

表 20-27 TX_FRAME_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| START | 保留（RESERVED） | | | | | | |
| 读 / 写（R/W）-0h | 读（R）-0h | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| N_WORDS | | | | FRAME_TYPE | | | |
| 读 / 写（R/W）-0h | | | | 读 / 写（R/W）-0h | | | |

表 20-28 TX_FRAME_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15 | START | 读 / 写（R/W） | 0h | 0：向该位写0不会产生任何影响<br>1：开始传输。该位会自动由硬件清空<br>复位类型：SYSRSn |
| 14-8 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 7-4 | N_WORDS | 读 / 写（R/W） | 0h | DATA_N_WORD数据帧的字长度<br>0：1字数据帧（16bit）<br>……<br>F：16字数据帧（256bit）<br>复位类型：SYSRSn |
| 3-0 | FRAME_TYPE | 读 / 写（R/W） | 0h | 发送帧类型<br>0000：ping帧<br>0100：DATA_1_WORD帧<br>0101：DATA_2_WORD帧<br>0110：DATA_4_WORD帧<br>0111：DATA_6_WORD帧<br>0011：DATA_N_WORD帧，N_WORDS中的值<br>1111：错误帧<br>其他值保留<br>复位类型：SYSRSn |

## 20.5.1.5.6 TX_FRAME_TAG_UDATA 寄存器

偏移地址：Eh　复位值：0000h

TX_FRAME_TAG_UDATA 寄存器的各字段说明如表中所示。

表 20-29 TX_FRAME_TAG_UDATA 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| USER_DATA | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | FRAME_TAG | | | |
| 读（R）-0h | | | | 读 / 写（R/W）-0h | | | |

表 20-30 TX_FRAME_TAG_UDATA 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-8 | USER_DATA | 读 / 写（R/W） | 0h | User Data phase中的值，可以任意使用，不会影响硬件行为<br>复位类型：SYSRSn |
| 7-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3-0 | FRAME_TAG | 读 / 写（R/W） | 0h | 只在软件触发的发送中使用<br>Frame Tag phase中的值，可以任意使用，不会影响硬件行为<br>对于外部触发的发送，不使用该寄存器，而是TX_PING_TAG<br>复位类型：SYSRSn |

### 20.5.1.5.7 TX_BUF_PTR_LOAD 寄存器

偏移地址：10h　复位值：0000h

TX_BUF_PTR_LOAD 寄存器的各字段说明如表中所示。

表 20-31 TX_BUF_PTR_LOAD 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | BUF_PTR_LOAD | | | |
| 读（R）-0h | | | | 读 / 写（R/W）-0h | | | |

表 20-32 TX_BUF_PTR_LOAD 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3-0 | BUF_PTR_LOAD | 读 / 写（R/W） | 0h | 强迫buffer指针指向一个位置，下一次传输将从该位置开始读取数据。设置该寄存器后，TX_BUF_PTR_STS中的值在最少3 SYSCLK + 3 TXCLK后相应改变。<br>会清零CURR_WORD_CNT寄存器。<br>在发送中时改变该寄存器会破坏正在传输的帧。<br>复位类型：SYSRSn |

## 20.5.1.5.8 TX_BUF_PTR_STS 寄存器

偏移地址：12h　复位值：0000h

TX_BUF_PTR_STS 寄存器的各字段说明如表中所示。

表 20-33 TX_BUF_PTR_STS 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | CURR_WORD_CNT | | | | |
| 读（R）-0h | | | 读（R）-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | CURR_BUF_PTR | | | |
| 读（R）-0h | | | | 读（R）-0h | | | |

表 20-34 TX_BUF_PTR_STS 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-13 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 12-8 | CURR_WORD_CNT | 读（R） | 0h | 当前没有正在传输的帧时有效，表示当前buffer中剩余的字数<br>复位类型：SYSRSn |
| 7-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3-0 | CURR_BUF_PTR | 读（R） | 0h | 当前没有正在传输的帧时有效，表示当前buffer指针的位置 |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| | | | | 复位类型：SYSRSn |

### 20.5.1.5.9 TX_PING_CTRL 寄存器

偏移地址：14h　复位值：0000h

TX_PING_CTRL 寄存器的各字段说明如表中所示。

表 20-35 TX_PING_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | EXT_TRIG_SEL |
| 读（R）-0h | | | | | | | 读 / 写（R/W）-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EXT_TRIG_SEL | | | | | EXT_TRIG_EN | TIMER_EN | CNT_RST |
| 读 / 写（R/W）-0h | | | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |

表 20-36 TX_PING_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-9 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 8-3 | EXT_TRIG_SEL | 读 / 写（R/W） | 0h | 决定32个外部触发输入的哪一个会作为源来产生ping帧<br>0：触发1作为源<br>……<br>1F：触发32作为源<br>只有使能EXT_TRIG_EN时才会产生ping帧<br>复位类型：SYSRSn |
| 2 | EXT_TRIG_EN | 读 / 写（R/W） | 0h | 0：外部触发不产生ping帧<br>1：外部触发产生ping帧，ping定时器被无效<br>复位类型：SYSRSn |
| 1 | TIMER_EN | 读 / 写（R/W） | 0h | 0：ping定时器无效，不产生ping帧<br>1：开启ping定时器，当定时值达到TX_PING_TO_REF后会产生ping帧 |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| | | | | 当使能TIMER_EN时应该不使能EXT_TRIG_EN<br>复位类型：SYSRSn |
| 0 | CNT_RST | 读 / 写（R/W） | 0h | 当该寄存器为1时ping定时器将保持为0（复位状态），需要手动写该寄存器为0以释放复位<br>复位类型：SYSRSn |

### 20.5.1.5.10 TX_PING_TAG 寄存器

偏移地址：16h　复位值：0000h

TX_PING_TAG 寄存器的各字段说明如表中所示。

表 20-37 TX_PING_TAG 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | TAG | | | |
| 读（R）-0h | | | | 读 / 写（R/W）-0h | | | |

表 20-38 TX_PING_TAG 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3-0 | TAG | 读 / 写（R/W） | 0h | ping frame tag，会在每个ping帧（外部触发/ping定时器）中发送。如果ping帧是认为产生的，则frame tag值存在TX_FRAME_TAG_UDATA.FRAME_TAG中，而不是该寄存器的值<br>复位类型：SYSRSn |

### 20.5.1.5.11 TX_PING_TO_REF 寄存器

偏移地址：18h　复位值：00000000h

TX_PING_TO_REF 寄存器的各字段说明如表中所示。

表 20-39 TX_PING_TO_REF 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TO_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TO_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TO_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TO_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |

表 20-40 TX_PING_TO_REF 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 31-0 | TO_REF | 读 / 写（R/W） | 0h | Ping定时器超时值<br>Ping定时器从0开始计数，达到该值后会产生超时事件，触发ping帧，然后重置为0并重新计数<br>复位类型：SYSRSn |

### 20.5.1.5.12 TX_PING_TO_CNT 寄存器

偏移地址：1Ch　复位值：00000000h

TX_PING_TO_CNT 寄存器的各字段说明如表中所示。

表 20-41 TX_PING_TO_CNT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TO_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TO_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TO_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TO_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |

表 20-42 TX_PING_TO_CNT 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 31-0 | TO_CNT | 读（R） | 0h | Ping定时器当前值<br>复位类型：SYSRSn |

### 20.5.1.5.13 TX_INT_CTRL 寄存器

偏移地址：20h　复位值：0000h

TX_INT_CTRL 寄存器的各字段说明如表中所示。

表 20-43 TX_INT_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | INT2_EN_PING_TO | INT2_EN_BUF_OVERRUN | INT2_EN_BUF_UNDERRUN | INT2_EN_FRAME_DONE |
| 读（R）-0h | | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | INT1_EN_PING_TO | INT1_EN_BUF_OVERRUN | INT1_EN_BUF_UNDERRUN | INT1_EN_FRAME_DONE |
| 读（R）-0h | | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |

表 20-44 TX_INT_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-12 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 11 | INT2_EN_PING_TO | 读 / 写（R/W） | 0h | 0：不触发<br>1：ping定时器事件触发INT2<br>复位类型：SYSRSn |
| 10 | INT2_EN_BUF_OVERRUN | 读 / 写（R/W） | 0h | 0：不触发<br>1：buffer过载事件触发INT2<br>复位类型：SYSRSn |
| 9 | INT2_EN_BUF_UNDERRUN | 读 / 写（R/W） | 0h | 0：不触发<br>1：buffer欠载事件触发INT2<br>复位类型：SYSRSn |
| 8 | INT2_EN_FRAME_DONE | 读 / 写（R/W） | 0h | 0：不触发<br>1：frame done事件触发INT2<br>复位类型：SYSRSn |
| 7-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3 | INT1_EN_PING_TO | 读 / 写（R/W） | 0h | 0：不触发<br>1：ping定时器事件触发INT1<br>复位类型：SYSRSn |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 2 | INT1_EN_BUF_OVERRUN | 读 / 写（R/W） | 0h | 0：不触发<br>1：buffer过载事件触发INT1<br>复位类型：SYSRSn |
| 1 | INT1_EN_BUF_UNDERRUN | 读 / 写（R/W） | 0h | 0：不触发<br>1：buffer欠载事件触发INT1<br>复位类型：SYSRSn |
| 0 | INT1_EN_FRAME_DONE | 读 / 写（R/W） | 0h | 0：不触发<br>1：frame done事件触发INT1<br>复位类型：SYSRSn |

### 20.5.1.5.14 TX_DMA_CTRL 寄存器

偏移地址：22h　复位值：0000h

TX_DMA_CTRL 寄存器的各字段说明如表中所示。

表 20-45 TX_DMA_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | | | | DMA_EVT_EN |
| 读（R）-0h | | | | | | | 读 / 写（R/W）-0h |

表 20-46 TX_DMA_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-1 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 0 | DMA_EVT_EN | 读 / 写（R/W） | 0h | 0：不产生<br>1：传输完成一帧后会产生DMA事件<br>DMA事件只会产生在数据帧中<br>复位类型：SYSRSn |

### 20.5.1.5.15 TX_LOCK_CTRL 寄存器

偏移地址：24h　复位值：0000h

TX_LOCK_CTRL 寄存器的各字段说明如表中所示。

表 20-47 TX_LOCK_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| 写（W）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | | | | LOCK |
| 读（R）-0h | | | | | | | 读 / 写（R/W）-0h |

表 20-48 TX_LOCK_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-8 | KEY | 写（W） | 0h | 写保护。写本寄存器其他位时，必须向此处写入0xA5。<br>复位类型：SYSRSn |
| 7-1 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 0 | LOCK | 读 / 写（R/W） | 0h | 0：TX控制寄存器可以被修改<br>1：TX控制寄存器无法被修改，直到该位被SYSRSN清除<br>复位类型：SYSRSn |

### 20.5.1.5.16 TX_EVT_STS 寄存器

偏移地址：28h　复位值：0000h

TX_EVT_STS 寄存器的各字段说明如表中所示。

表 20-49 TX_EVT_STS 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | PING_TRIGGERED | BUF_OVERRUN | BUF_UNDERRUN | FRAME_DONE |
| 读（R）-0h | | | | 读（R）-0h | 读（R）-0h | 读（R）-0h | 读（R）-0h |

表 20-50 TX_EVT_STS 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3 | PING_TRIGGERED | 读（R） | 0h | 0：没有触发ping帧<br>1：有ping帧被ping定时器或外部触发触发<br>复位类型：SYSRSn |
| 2 | BUF_OVERRUN | 读（R） | 0h | 0：没有发生buffer过载<br>1：发生buffer过载<br>复位类型：SYSRSn |
| 1 | BUF_UNDERRUN | 读（R） | 0h | 0：没有发生buffer欠载<br>1：发生buffer欠载<br>复位类型：SYSRSn |
| 0 | FRAME_DONE | 读（R） | 0h | 0：没有发生frame done<br>1：发生frame done<br>复位类型：SYSRSn |

### 20.5.1.5.17 TX_EVT_CLR 寄存器

偏移地址：2Ch　复位值：0000h

TX_EVT_CLR 寄存器的各字段说明如表中所示。

表 20-51 TX_EVT_CLR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | PING_TRIGGERED | BUF_OVERRUN | BUF_UNDERRUN | FRAME_DONE |
| 读（R）-0h | | | | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h |

表 20-52 TX_EVT_CLR 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3 | PING_TRIGGERED | 写（W） | 0h | 写1将TX_EVT_STS中的对应位清0<br>在PING_TO_MODE为0时，为硬件触发ping超时事件。硬件触发ping超时发生时可能有帧正在发送，所以可能发生在该位写1但TX_EVT_STS中的对应位没有清0的情况。<br>复位类型：SYSRSn |
| 2 | BUF_OVERRUN | 写（W） | 0h | 写1将TX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 1 | BUF_UNDERRUN | 写（W） | 0h | 写1将TX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 0 | FRAME_DONE | 写（W） | 0h | 写1将TX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |

### 20.5.1.5.18 TX_EVT_FRC 寄存器

偏移地址：2Eh　复位值：0000h

TX_EVT_FRC 寄存器的各字段说明如表中所示。

表 20-53 TX_EVT_FRC 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | PING_TRIGGERED | BUF_OVERRUN | BUF_UNDERRUN | FRAME_DONE |
| 读（R）-0h | | | | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h |

表 20-54 TX_EVT_FRC 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3 | PING_TRIGGERED | 写（W） | 0h | 写1将TX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 2 | BUF_OVERRUN | 写（W） | 0h | 写1将TX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 1 | BUF_UNDERRUN | 写（W） | 0h | 写1将TX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 0 | FRAME_DONE | 写（W） | 0h | 写1将TX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |

### 20.5.1.5.19 TX_USER_CRC 寄存器

偏移地址：30h　复位值：0000h

TX_USER_CRC 寄存器的各字段说明如表中所示。

表 20-55 TX_USER_CRC 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| USER_CRC | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |

表 20-56 TX_USER_CRC 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-8 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 7-0 | USER_CRC | 读 / 写（R/W） | 0h | 当TX_OPER_CTRL_LO.SW_CRC = 1时使用<br>复位类型：SYSRSn |

### 20.5.1.5.20 TX_ECC_DATA 寄存器

偏移地址：40h　复位值：00000000h

TX_ECC_DATA 寄存器的各字段说明如表中所示。

表 20-57 TX_ECC_DATA 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DATA_HIGH | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DATA_HIGH | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DATA_LOW | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA_LOW | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |

表 20-58 TX_ECC_DATA 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 31-16 | DATA_HIGH | 读 / 写（R/W） | 0h | 写该寄存器会进行整体32bit的ECC计算，更新TX_ECC_VAL，所以32bit应一起写入<br>复位类型：SYSRSn |
| 15-0 | DATA_LOW | 读 / 写（R/W） | 0h | 写该寄存器会进行这16bit的ECC计算，更新TX_ECC_VAL<br>复位类型：SYSRSn |

### 20.5.1.5.21 TX_ECC_VAL 寄存器

偏移地址：44h　复位值：000Ch

TX_ECC_VAL 寄存器的各字段说明如表中所示。

表 20-59 TX_ECC_VAL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | ECC_VAL | | | | | | |
| 读（R）-0h | 读（R）-Ch | | | | | | |

表 20-60 TX_ECC_VAL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-7 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 6-0 | ECC_VAL | 读（R） | Ch | 计算的ECC结果<br>复位类型：SYSRSn |

### 20.5.1.5.22 TX_BUF_BASE_y 寄存器

偏移地址：0x0080 + y×2 (y = 0..15)　复位值：0000h

TX_BUF_BASE_y 寄存器的各字段说明如表中所示。

表 20-61 TX_BUF_BASE_y 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| BASE_ADDRESS | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BASE_ADDRESS | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |

表 20-62 TX_BUF_BASE_y 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-0 | BASE_ADDRESS | 读 / 写（R/W） | 0h | 发送数据缓冲区基地址。发送器使用的 16 字数据缓冲区的基地址。<br>复位类型：SYSRSn |

## 20.5.1.6 FSIRX 寄存器字段说明

### 20.5.1.6.1 RX_MASTER_CTRL 寄存器

偏移地址：0h　复位值：0000h

RX_MASTER_CTRL 寄存器的各字段说明如表中所示。

表 20-63 RX_MASTER_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| 写（W）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | INPUT_ISOLATE | SPI_PAIRING | INT_LOOPBACK | CORE_RST |
| 读（R）-0h | | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |

表 20-64 RX_MASTER_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-8 | KEY | 写（W） | 0h | 写保护。写本寄存器其他位时，必须向此处写入0xA5，当此处为0xA5时写入才是有效的。<br>复位类型：SYSRSn |
| 7-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3 | INPUT_ISOLATE | 读 / 写（R/W） | 0h | RX 输入隔离位。写 1 时，FSI RX 输入（RXCLK、RXD0、RXD1）与器件引脚驱动隔离，并保持为无效电平 1；便于将 RX 输入切换到另一组器件引脚。<br>复位类型：SYSRSn |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 2 | SPI_PAIRING | 读 / 写（R/W） | 0h | 0：不启用<br>1：启用SPI时钟配对，只在SPI模式下有效<br>复位类型：SYSRSn |
| 1 | INT_LOOPBACK | 读 / 写（R/W） | 0h | 0：不启用<br>1：启用回环模式，RX会优先从TX端接收信号而不是引脚<br>复位类型：SYSRSn |
| 0 | CORE_RST | 读 / 写（R/W） | 0h | 0：撤销复位<br>1：复位<br>复位类型：SYSRSn |

### 20.5.1.6.2 RX_OPER_CTRL 寄存器

偏移地址：8h　复位值：0000h

RX_OPER_CTRL 寄存器的各字段说明如表中所示。

表 20-65 RX_OPER_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | PING_WD_RST_MODE |
| 读（R）-0h | | | | | | | 读 / 写（R/W）-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ECC_SEL | N_WORDS | | | | SPI_MODE | DATA_WIDTH | |
| 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | |

表 20-66 RX_OPER_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-9 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 8 | PING_WD_RST_MODE | 读 / 写（R/W） | 0h | 0：ping看门狗只会被ping帧重置<br>1：ping看门狗会被接收到的任何帧重置<br>复位类型：SYSRSn |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 7 | ECC_SEL | 读 / 写（R/W） | 0h | 0：32bit ECC<br>1：16bit ECC<br>复位类型：SYSRSn |
| 6-3 | N_WORDS | 读 / 写（R/W） | 0h | 应与TX中设置一致<br>复位类型：SYSRSn |
| 2 | SPI_MODE | 读 / 写（R/W） | 0h | 0：FSI模式<br>1：SPI兼容模式<br>复位类型：SYSRSn |
| 1-0 | DATA_WIDTH | 读 / 写（R/W） | 0h | 0：只使用RXD0<br>1：使用RXD0和RXD1传输<br>其他值保留<br>复位类型：SYSRSn |

### 20.5.1.6.3 RX_FRAME_INFO 寄存器

偏移地址：Ch　复位值：0000h

RX_FRAME_INFO 寄存器的各字段说明如表中所示。

表 20-67 RX_FRAME_INFO 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | FRAME_TYPE | | | |
| 读（R）-0h | | | | 读（R）-0h | | | |

表 20-68 RX_FRAME_INFO 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3-0 | FRAME_TYPE | 读（R） | 0h | 上一个成功接收的帧的帧类型，但不指示ping帧<br>0100：DATA_1_WORD<br>0101：DATA_2_WORD<br>0110：DATA_4_WORD<br>0111：DATA_6_WORD |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| | | | | 0011：DATA_N_WORD，具体值写在RX_OPER_CTRL.N_WORDS<br>1111：错误帧<br>其他值保留<br>复位类型：SYSRSn |

### 20.5.1.6.4 RX_FRAME_TAG_UDATA 寄存器

偏移地址：Eh　复位值：0000h

RX_FRAME_TAG_UDATA 寄存器的各字段说明如表中所示。

表 20-69 RX_FRAME_TAG_UDATA 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| USER_DATA | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | FRAME_TAG | | | | 保留（RESERVED） |
| 读（R）-0h | | | 读（R）-0h | | | | 读（R）-0h |

表 20-70 RX_FRAME_TAG_UDATA 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-8 | USER_DATA | 读（R） | 0h | 上一个成功接收的帧的user data<br>复位类型：SYSRSn |
| 7-5 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 4-1 | FRAME_TAG | 读（R） | 0h | 上一个成功接收的帧的frame tag<br>复位类型：SYSRSn |
| 0 | 保留（RESERVED） | 读（R） | 0h | 保留 |

### 20.5.1.6.5 RX_DMA_CTRL 寄存器

偏移地址：10h　复位值：0000h

RX_DMA_CTRL 寄存器的各字段说明如表中所示。

表 20-71 RX_DMA_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | | | | DMA_EVT_EN |
| 读（R）-0h | | | | | | | 读 / 写（R/W）-0h |

表 20-72 RX_DMA_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-1 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 0 | DMA_EVT_EN | 读 / 写（R/W） | 0h | 0：不产生<br>1：接收完成一帧后会产生DMA事件<br>DMA事件只会产生在数据帧中<br>复位类型：SYSRSn |

### 20.5.1.6.6 RX_EVT_STS 寄存器

偏移地址：14h　复位值：0000h

RX_EVT_STS 寄存器的各字段说明如表中所示。

表 20-73 RX_EVT_STS 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | ERROR_TAG_MATCH | DATA_TAG_MATCH | PING_TAG_MATCH | DATA_FRAME | FRAME_OVERRUN | PING_FRAME | ERR_FRAME |
| 读（R）-0h | 读（R）-0h | 读（R）-0h | 读（R）-0h | 读（R）-0h | 读（R）-0h | 读（R）-0h | 读（R）-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BUF_UNDERRUN | FRAME_DONE | BUF_OVERRUN | EOF_ERR | TYPE_ERR | CRC_ERR | FRAME_WD_TO | PING_WD_TO |
| 读（R）-0h | 读（R）-0h | 读（R）-0h | 读（R）-0h | 读（R）-0h | 读（R）-0h | 读（R）-0h | 读（R）-0h |

表 20-74 RX_EVT_STS 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 14 | ERROR_TAG_MATCH | 读（R） | 0h | 错误帧标签匹配标志。1 表示接收到与屏蔽标签参考值匹配的错误帧；也可通过写 RX_EVT_FRC 强制置位。0：未收到匹配的错误帧。<br>复位类型：SYSRSn |
| 13 | DATA_TAG_MATCH | 读（R） | 0h | 数据帧标签匹配标志。1 表示接收到与屏蔽标签参考值匹配的数据帧；也可通过写 RX_EVT_FRC 强制置位。0：未收到匹配的数据帧。<br>复位类型：SYSRSn |
| 12 | PING_TAG_MATCH | 读（R） | 0h | ping 帧标签匹配标志。1 表示接收到与屏蔽标签参考值匹配的 ping 帧；也可通过写 RX_EVT_FRC 强制置位。0：未收到匹配的 ping 帧。<br>复位类型：SYSRSn |
| 11 | DATA_FRAME | 读（R） | 0h | 接收到帧<br>软件通过写RX_EVT_FRC可以置1，也可以写RX_EVT_CLR清0<br>复位类型：SYSRSn |
| 10 | FRAME_OVERRUN | 读（R） | 0h | 当收到新的帧时 DATA_FRAME_RCVD/ERROR_FRAME_RCVD仍然为1<br>0：没有发生frame过载<br>1：发生frame过载<br>软件通过写RX_EVT_FRC可以置1，也可以写RX_EVT_CLR清0<br>复位类型：SYSRSn |
| 9 | PING_FRAME | 读（R） | 0h | 0：没有收到ping帧<br>1：收到ping帧<br>软件通过写RX_EVT_FRC可以置1，也可以写RX_EVT_CLR清0<br>复位类型：SYSRSn |
| 8 | ERR_FRAME | 读（R） | 0h | 0：没有收到错误帧<br>1：收到错误帧<br>软件通过写RX_EVT_FRC可以置1，也可以 |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
| --- | --- | --- | --- | --- |
| | | | | 写RX_EVT_CLR清0<br>复位类型：SYSRSn |
| 7 | BUF_UNDERRUN | 读（R） | 0h | 0：没有发生buffer欠载<br>1：发生buffer欠载<br>软件通过写RX_EVT_FRC可以置1，也可以写RX_EVT_CLR清0<br>复位类型：SYSRSn |
| 6 | FRAME_DONE | 读（R） | 0h | 0：没有成功接收一帧<br>1：成功接收一帧<br>软件通过写RX_EVT_FRC可以置1，也可以写RX_EVT_CLR清0<br>复位类型：SYSRSn |
| 5 | BUF_OVERRUN | 读（R） | 0h | 0：没有发生buffer过载<br>1：发生buffer过载<br>软件通过写RX_EVT_FRC可以置1，也可以写RX_EVT_CLR清0<br>复位类型：SYSRSn |
| 4 | EOF_ERR | 读（R） | 0h | 0：没有接收到非法end of frame<br>1：接收到非法end of frame<br>软件通过写RX_EVT_FRC可以置1，也可以写RX_EVT_CLR清0<br>复位类型：SYSRSn |
| 3 | TYPE_ERR | 读（R） | 0h | 0：没有接收到非法frame type<br>1：接收到非法frame type<br>软件通过写RX_EVT_FRC可以置1，也可以写RX_EVT_CLR清0<br>复位类型：SYSRSn |
| 2 | CRC_ERR | 读（R） | 0h | 0：没有发生CRC错误<br>1：发生CRC错误<br>软件通过写RX_EVT_FRC可以置1，也可以写RX_EVT_CLR清0<br>复位类型：SYSRSn |
| 1 | FRAME_WD_TO | 读（R） | 0h | 0：没有发生帧看门狗超时事件<br>1：发生帧看门狗超时事件<br>软件通过写RX_EVT_FRC可以置1，也可以写RX_EVT_CLR清0<br>复位类型：SYSRSn |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 0 | PING_WD_TO | 读（R） | 0h | 0：没有发生ping看门狗超时事件<br>1：发生ping看门狗超时事件<br>软件通过写RX_EVT_FRC可以置1，也可以写RX_EVT_CLR清0<br>复位类型：SYSRSn |

### 20.5.1.6.7 RX_CRC_INFO 寄存器

偏移地址：16h　复位值：0000h

RX_CRC_INFO 寄存器的各字段说明如表中所示。

表 20-75 RX_CRC_INFO 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| CALC_CRC | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RX_CRC | | | | | | | |
| 读（R）-0h | | | | | | | |

表 20-76 RX_CRC_INFO 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-8 | CALC_CRC | 读（R） | 0h | 硬件计算的CRC值<br>复位类型：SYSRSn |
| 7-0 | RX_CRC | 读（R） | 0h | 接收到的CRC值<br>复位类型：SYSRSn |

### 20.5.1.6.8 RX_EVT_CLR 寄存器

偏移地址：18h　复位值：0000h

RX_EVT_CLR 寄存器的各字段说明如表中所示。

表 20-77 RX_EVT_CLR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | ERROR_TAG_MATCH | DATA_TAG_MATCH | PING_TAG_MATCH | DATA_FRAME | FRAME_OVERRUN | PING_FRAME | ERR_FRAME |
| 读（R）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BUF_UNDERRUN | FRAME_DONE | BUF_OVERRUN | EOF_ERR | TYPE_ERR | CRC_ERR | FRAME_WD_TO | PING_WD_TO |
| 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h |

表 20-78 RX_EVT_CLR 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 14 | ERROR_TAG_MATCH | 写（W） | 0h | 清除错误帧标签匹配标志。0：写 0 无影响；1：写 1 清除 RX_EVT_STS 中对应标志。<br>复位类型：SYSRSn |
| 13 | DATA_TAG_MATCH | 写（W） | 0h | 清除数据帧标签匹配标志。0：写 0 无影响；1：写 1 清除 RX_EVT_STS 中对应标志。<br>复位类型：SYSRSn |
| 12 | PING_TAG_MATCH | 写（W） | 0h | 清除 ping 帧标签匹配标志。0：写 0 无影响；1：写 1 清除 RX_EVT_STS 中对应标志。<br>复位类型：SYSRSn |
| 11 | DATA_FRAME | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 10 | FRAME_OVERRUN | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 9 | PING_FRAME | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 8 | ERR_FRAME | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
| --- | --- | --- | --- | --- |
| 7 | BUF_UNDERRUN | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 6 | FRAME_DONE | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 5 | BUF_OVERRUN | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 4 | EOF_ERR | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 3 | TYPE_ERR | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 2 | CRC_ERR | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 1 | FRAME_WD_TO | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |
| 0 | PING_WD_TO | 写（W） | 0h | 写1将RX_EVT_STS中的对应位清0<br>复位类型：SYSRSn |

### 20.5.1.6.9 RX_EVT_FRC 寄存器

偏移地址：1Ah　复位值：0000h

RX_EVT_FRC 寄存器的各字段说明如表中所示。

表 20-79 RX_EVT_FRC 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 保留（RESERVED） | ERROR_TAG_MATCH | DATA_TAG_MATCH | PING_TAG_MATCH | DATA_FRAME | FRAME_OVERRUN | PING_FRAME | ERR_FRAME |
| 读（R）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BUF_UNDERRUN | FRAME_DONE | BUF_OVERRUN | EOF_ERR | TYPE_ERR | CRC_ERR | FRAME_WD_TO | PING_WD_TO |
| 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h | 写（W）-0h |

表 20-80 RX_EVT_FRC 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 14 | ERROR_TAG_MATCH | 写（W） | 0h | 强制错误帧标签匹配标志。写 1 使 RX_EVT_STS 中对应标志置位，用于软件模拟事件并测试关联的软件/ISR。<br>复位类型：SYSRSn |
| 13 | DATA_TAG_MATCH | 写（W） | 0h | 强制数据帧标签匹配标志。写 1 使 RX_EVT_STS 中对应标志置位，用于软件模拟事件并测试关联的软件/ISR。<br>复位类型：SYSRSn |
| 12 | PING_TAG_MATCH | 写（W） | 0h | 强制 ping 帧标签匹配标志。写 1 使 RX_EVT_STS 中对应标志置位，用于软件模拟事件并测试关联的软件/ISR。<br>复位类型：SYSRSn |
| 11 | DATA_FRAME | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 10 | FRAME_OVERRUN | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 9 | PING_FRAME | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 8 | ERR_FRAME | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 7 | BUF_UNDERRUN | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 6 | FRAME_DONE | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 5 | BUF_OVERRUN | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 4 | EOF_ERR | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 3 | TYPE_ERR | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 2 | CRC_ERR | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 1 | FRAME_WD_TO | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |
| 0 | PING_WD_TO | 写（W） | 0h | 写1将RX_EVT_STS中的对应位置1<br>复位类型：SYSRSn |

### 20.5.1.6.10 RX_BUF_PTR_LOAD 寄存器

偏移地址：1Ch　复位值：0000h

RX_BUF_PTR_LOAD 寄存器的各字段说明如表中所示。

表 20-81 RX_BUF_PTR_LOAD 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | BUF_PTR_LOAD | | | |
| 读（R）-0h | | | | 读 / 写（R/W）-0h | | | |

表 20-82 RX_BUF_PTR_LOAD 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-4 | 保留<br>（RESERVED） | 读（R） | 0h | 保留 |
| 3-0 | BUF_PTR_LOAD | 读 / 写<br>（R/W） | 0h | 强迫buffer指针指向一个位置，下一次传输将从该位置开始写入数据。设置该寄存器后，RX_BUF_PTR_SIS中的值在最少3 SYSCLK + 3 RXCLK后相应改变。<br>会清零CURR_WORD_CNT寄存器。<br>复位类型：SYSRSn |

### 20.5.1.6.11 RX_BUF_PTR_STS 寄存器

偏移地址：1Eh　复位值：0000h

RX_BUF_PTR_STS 寄存器的各字段说明如表中所示。

表 20-83 RX_BUF_PTR_STS 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | CURR_WORD_CNT | | | | |
| 读（R）-0h | | | 读（R）-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | CURR_BUF_PTR | | | |
| 读（R）-0h | | | | 读（R）-0h | | | |

表 20-84 RX_BUF_PTR_STS 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-13 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 12-8 | CURR_WORD_CNT | 读（R） | 0h | 当前没有正在传输的帧时有效，表示当前buffer中剩余的字数<br>复位类型：SYSRSn |
| 7-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3-0 | CURR_BUF_PTR | 读（R） | 0h | 当前没有正在传输的帧时有效，表示当前buffer指针的位置<br>复位类型：SYSRSn |

### 20.5.1.6.12 RX_FRAME_WD_CTRL 寄存器

偏移地址：20h　复位值：0000h

RX_FRAME_WD_CTRL 寄存器的各字段说明如表中所示。

表 20-85 RX_FRAME_WD_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | | | FRAME_WD_EN | FRAME_WD_CNT_RST |
| 读（R）-0h | | | | | | 读/写（R/W）-0h | 读/写（R/W）-0h |

表 20-86 RX_FRAME_WD_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-2 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 1 | FRAME_WD_EN | 读 / 写（R/W） | 0h | 帧看门狗计数器为RX_FRAME_WD_CNT，当接收到合法start of frame后开始从0计数。当计数值达到RX_FRAME_WD_REF，会产生帧看门狗超时事件（RX_EVT_STS.FRAME_WD_TO），并且计数器重置为0继续计数至下一次合法的start of frame<br>0：帧看门狗不启动<br>1：帧看门狗启动<br>复位类型：SYSRSn |
| 0 | FRAME_WD_CNT_RST | 读 / 写（R/W） | 0h | 当该寄存器为1时帧看门狗计数器将保持为0（复位状态），需要手动写该寄存器为0以释放复位<br>复位类型：SYSRSn |

## 20.5.1.6.13 RX_FRAME_WD_REF 寄存器

偏移地址：24h　复位值：00000000h

RX_FRAME_WD_REF 寄存器的各字段说明如表中所示。

表 20-87 RX_FRAME_WD_REF 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| FRAME_WD_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FRAME_WD_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FRAME_WD_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FRAME_WD_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |

表 20-88 RX_FRAME_WD_REF 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 31-0 | FRAME_WD_REF | 读 / 写（R/W） | 0h | 帧看门狗计数器参考值<br>复位类型：SYSRSn |

### 20.5.1.6.14 RX_FRAME_WD_CNT 寄存器

偏移地址：28h　复位值：00000000h

RX_FRAME_WD_CNT 寄存器的各字段说明如表中所示。

表 20-89 RX_FRAME_WD_CNT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| FRAME_WD_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| FRAME_WD_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| FRAME_WD_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FRAME_WD_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |

表 20-90 RX_FRAME_WD_CNT 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 31-0 | FRAME_WD_CNT | 读（R） | 0h | 帧看门狗计数器当前值<br>会被以下情况重置为0：写FRME_WD_CNT_RST、达到FRAME_WD_REF、成功接收数据帧<br>复位类型：SYSRSn |

### 20.5.1.6.15 RX_PING_WD_CTRL 寄存器

偏移地址：2Ch　复位值：0000h

RX_PING_WD_CTRL 寄存器的各字段说明如表中所示。

表 20-91 RX_PING_WD_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | | | PING_WD_EN | PING_WD_RST |
| 读（R）-0h | | | | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |

表 20-92 RX_PING_WD_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-2 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 1 | PING_WD_EN | 读 / 写（R/W） | 0h | ping看门狗计数器为RX_PING_WD_CNT，使能该寄存器后RX_PING_WD_CNT从0开始计数。当计数值达到RX_PING_WD_REF，会产生ping看门狗超时事件（RX_EVT_STS.PING_WD_TO），并且计数器重置为0继续计数<br>0：ping看门狗不启动<br>1：ping看门狗启动<br>复位类型：SYSRSn |
| 0 | PING_WD_RST | 读 / 写（R/W） | 0h | 当该寄存器为1时ping看门狗计数器将保持为0（复位状态），需要手动写该寄存器为0以释放复位<br>复位类型：SYSRSn |

## 20.5.1.6.16 RX_PING_TAG 寄存器

偏移地址：2Eh　复位值：0000h

RX_PING_TAG 寄存器的各字段说明如表中所示。

表 20-93 RX_PING_TAG 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | PING_TAG | | | | 保留（RESERVED） |
| 读（R）-0h | | | 读（R）-0h | | | | 读（R）-0h |

表 20-94 RX_PING_TAG 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-5 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 4-1 | PING_TAG | 读（R） | 0h | 上一个成功接收的ping帧的frame tag<br>复位类型：SYSRSn |
| 0 | 保留（RESERVED） | 读（R） | 0h | 保留 |

## 20.5.1.6.17 RX_PING_WD_REF 寄存器

偏移地址：30h　复位值：00000000h

RX_PING_WD_REF 寄存器的各字段说明如表中所示。

表 20-95 RX_PING_WD_REF 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PING_WD_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PING_WD_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PING_WD_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PING_WD_REF | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |

表 20-96 RX_PING_WD_REF 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 31-0 | PING_WD_REF | 读 / 写（R/W） | 0h | ping看门狗计数器参考值<br>复位类型：SYSRSn |

### 20.5.1.6.18 RX_PING_WD_CNT 寄存器

偏移地址：34h　复位值：00000000h

RX_PING_WD_CNT 寄存器的各字段说明如表中所示。

表 20-97 RX_PING_WD_CNT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| PING_WD_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| PING_WD_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PING_WD_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PING_WD_CNT | | | | | | | |
| 读（R）-0h | | | | | | | |

表 20-98 RX_PING_WD_CNT 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 31-0 | PING_WD_CNT | 读（R） | 0h | ping看门狗计数器当前值<br>复位类型：SYSRSn |

### 20.5.1.6.19 RX_INT1_CTRL 寄存器

偏移地址：38h　复位值：0000h

RX_INT1_CTRL 寄存器的各字段说明如表中所示。

表 20-99 RX_INT1_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | INT1_EN_ERROR_TAG_MATCH | INT1_EN_DATA_TAG_MATCH | INT1_EN_PING_TAG_MATCH | INT1_EN_DATA_FRAME | INT1_EN_FRAME_OVERRUN | INT1_EN_PING_FRAME | INT1_EN_ERR_FRAME |
| 读（R）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT1_EN_UNDERRUN | INT1_EN_FRAME_DONE | INT1_EN_OVERRUN | INT1_EN_EOF_ERR | INT1_EN_TYPE_ERR | INT1_EN_CRC_ERR | INT1_EN_FRAME_WD_TO | INT1_EN_PING_WD_TO |
| 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |

表 20-100 RX_INT1_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 14 | INT1_EN_ERROR_TAG_MATCH | 读 / 写（R/W） | 0h | 使能“收到标签匹配的错误帧”事件触发 RX_INT1 中断。0：该事件不触发 RX_INT1；1：该事件触发 RX_INT1。<br>复位类型：SYSRSn |
| 13 | INT1_EN_DATA_TAG_MATCH | 读 / 写（R/W） | 0h | 使能“收到标签匹配的数据帧”事件触发 RX_INT1 中断。0：该事件不触发 RX_INT1；1：该事件触发 RX_INT1。<br>复位类型：SYSRSn |
| 12 | INT1_EN_PING_TAG_MATCH | 读 / 写（R/W） | 0h | 使能“收到标签匹配的 ping 帧”事件触发 RX_INT1 中断。0：该事件不触发 RX_INT1；1：该事件触发 RX_INT1。<br>复位类型：SYSRSn |
| 11 | INT1_EN_DATA_FRAME | 读 / 写（R/W） | 0h | 0：不触发<br>1：接收数据帧触发INT1<br>复位类型：SYSRSn |
| 10 | INT1_EN_FRAME_OVERRUN | 读 / 写（R/W） | 0h | 0：不触发<br>1：帧过载事件触发INT1<br>复位类型：SYSRSn |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
| --- | --- | --- | --- | --- |
| 9 | INT1_EN_PING_FRAME | 读 / 写（R/W） | 0h | 0：不触发<br>1：接收ping帧触发INT1<br>复位类型：SYSRSn |
| 8 | INT1_EN_ERR_FRAME | 读 / 写（R/W） | 0h | 0：不触发<br>1：接收错误帧触发INT1<br>复位类型：SYSRSn |
| 7 | INT1_EN_UNDERRUN | 读 / 写（R/W） | 0h | 0：不触发<br>1：buffer欠载事件触发INT1<br>复位类型：SYSRSn |
| 6 | INT1_EN_FRAME_DONE | 读 / 写（R/W） | 0h | 0：不触发<br>1：frame done事件触发INT1<br>复位类型：SYSRSn |
| 5 | INT1_EN_OVERRUN | 读 / 写（R/W） | 0h | 0：不触发<br>1：buffer过载事件触发INT1<br>复位类型：SYSRSn |
| 4 | INT1_EN_EOF_ERR | 读 / 写（R/W） | 0h | 0：不触发<br>1：非法帧尾事件触发INT1<br>复位类型：SYSRSn |
| 3 | INT1_EN_TYPE_ERR | 读 / 写（R/W） | 0h | 0：不触发<br>1：非法帧类型事件触发INT1<br>复位类型：SYSRSn |
| 2 | INT1_EN_CRC_ERR | 读 / 写（R/W） | 0h | 0：不触发<br>1：CRC错误事件触发INT1<br>复位类型：SYSRSn |
| 1 | INT1_EN_FRAME_WD_TO | 读 / 写（R/W） | 0h | 0：不触发<br>1：帧超时事件触发INT1<br>复位类型：SYSRSn |
| 0 | INT1_EN_PING_WD_TO | 读 / 写（R/W） | 0h | 0：不触发<br>1：ping超时事件触发INT1<br>复位类型：SYSRSn |

## 20.5.1.6.20 RX_INT2_CTRL 寄存器

偏移地址：3Ah　复位值：0000h

RX_INT2_CTRL 寄存器的各字段说明如表中所示。

表 20-101 RX_INT2_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | INT2_EN_ERROR_TAG_MATCH | INT2_EN_DATA_TAG_MATCH | INT2_EN_PING_TAG_MATCH | INT2_EN_DATA_FRAME | INT2_EN_FRAME_OVERRUN | INT2_EN_PING_FRAME | INT2_EN_ERR_FRAME |
| 读（R）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| INT2_EN_UNDERRUN | INT2_EN_FRAME_DONE | INT2_EN_OVERRUN | INT2_EN_EOF_ERR | INT2_EN_TYPE_ERR | INT2_EN_CRC_ERR | INT2_EN_FRAME_WD_TO | INT2_EN_PING_WD_TO |
| 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |

表 20-102 RX_INT2_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 14 | INT2_EN_ERROR_TAG_MATCH | 读 / 写（R/W） | 0h | 使能“收到标签匹配的错误帧”事件触发 RX_INT2 中断。0：该事件不触发 RX_INT2；1：该事件触发 RX_INT2。<br>复位类型：SYSRSn |
| 13 | INT2_EN_DATA_TAG_MATCH | 读 / 写（R/W） | 0h | 使能“收到标签匹配的数据帧”事件触发 RX_INT2 中断。0：该事件不触发 RX_INT2；1：该事件触发 RX_INT2。<br>复位类型：SYSRSn |
| 12 | INT2_EN_PING_TAG_MATCH | 读 / 写（R/W） | 0h | 使能“收到标签匹配的 ping 帧”事件触发 RX_INT2 中断。0：该事件不触发 RX_INT2；1：该事件触发 RX_INT2。<br>复位类型：SYSRSn |
| 11 | INT2_EN_DATA_FRAME | 读 / 写（R/W） | 0h | 0：不触发<br>1：接收数据帧触发INT2<br>复位类型：SYSRSn |
| 10 | INT2_EN_FRAME_OVERRUN | 读 / 写（R/W） | 0h | 0：不触发<br>1：帧过载事件触发INT2 |

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
| --- | --- | --- | --- | --- |
| | | | | 复位类型：SYSRSn |
| 9 | INT2_EN_PING_FRAME | 读 / 写（R/W） | 0h | 0：不触发<br>1：接收ping帧触发INT2<br>复位类型：SYSRSn |
| 8 | INT2_EN_ERR_FRAME | 读 / 写（R/W） | 0h | 0：不触发<br>1：接收错误帧触发INT2<br>复位类型：SYSRSn |
| 7 | INT2_EN_UNDERRUN | 读 / 写（R/W） | 0h | 0：不触发<br>1：buffer欠载事件触发INT2<br>复位类型：SYSRSn |
| 6 | INT2_EN_FRAME_DONE | 读 / 写（R/W） | 0h | 0：不触发<br>1：frame done事件触发INT2<br>复位类型：SYSRSn |
| 5 | INT2_EN_OVERRUN | 读 / 写（R/W） | 0h | 0：不触发<br>1：buffer过载事件触发INT2<br>复位类型：SYSRSn |
| 4 | INT2_EN_EOF_ERR | 读 / 写（R/W） | 0h | 0：不触发<br>1：非法帧尾事件触发INT2<br>复位类型：SYSRSn |
| 3 | INT2_EN_TYPE_ERR | 读 / 写（R/W） | 0h | 0：不触发<br>1：非法帧类型事件触发INT2<br>复位类型：SYSRSn |
| 2 | INT2_EN_CRC_ERR | 读 / 写（R/W） | 0h | 0：不触发<br>1：CRC错误事件触发INT2<br>复位类型：SYSRSn |
| 1 | INT2_EN_FRAME_WD_TO | 读 / 写（R/W） | 0h | 0：不触发<br>1：帧超时事件触发INT2<br>复位类型：SYSRSn |
| 0 | INT2_EN_PING_WD_TO | 读 / 写（R/W） | 0h | 0：不触发<br>1：ping超时事件触发INT2<br>复位类型：SYSRSn |

### 20.5.1.6.21 RX_LOCK_CTRL 寄存器

偏移地址：3Ch　复位值：0000h

RX_LOCK_CTRL 寄存器的各字段说明如表中所示。

表 20-103 RX_LOCK_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| KEY | | | | | | | |
| 写（W）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | | | | LOCK |
| 读（R）-0h | | | | | | | 读 / 写（R/W）-0h |

表 20-104 RX_LOCK_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-8 | KEY | 写（W） | 0h | 写保护。写本寄存器其他位时，必须向此处写入0xA5。<br>复位类型：SYSRSn |
| 7-1 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 0 | LOCK | 读 / 写（R/W） | 0h | 0：TX控制寄存器可以被修改<br>1：TX控制寄存器无法被修改，直到该位被SYSRS清除<br>复位类型：SYSRSn |

### 20.5.1.6.22 RX_ECC_DATA 寄存器

偏移地址：40h　复位值：00000000h

RX_ECC_DATA 寄存器的各字段说明如表中所示。

表 20-105 RX_ECC_DATA 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DATA_HIGH | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| DATA_HIGH | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| DATA_LOW | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA_LOW | | | | | | | |
| 读 / 写（R/W）-0h | | | | | | | |

表 20-106 RX_ECC_DATA 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 31-16 | DATA_HIGH | 读 / 写（R/W） | 0h | 写该寄存器会进行整体32bit的ECC计算，更新TX_ECC_VAL，所以32bit应一起写入<br>复位类型：SYSRSn |
| 15-0 | DATA_LOW | 读 / 写（R/W） | 0h | 写该寄存器会进行这16bit的ECC计算，更新TX_ECC_VAL<br>复位类型：SYSRSn |

## 20.5.1.6.23 RX_ECC_VAL 寄存器

偏移地址：44h　复位值：0000h

RX_ECC_VAL 寄存器的各字段说明如表中所示。

表 20-107 RX_ECC_VAL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | ECC_VAL | | | | | | |
| 读（R）-0h | 读 / 写（R/W）-0h | | | | | | |

表 20-108 RX_ECC_VAL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-7 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 6-0 | ECC_VAL | 读 / 写（R/W） | 0h | 用户将传输得到的ECC（USER_DATA）自行拷贝到此处<br>复位类型：SYSRSn |

### 20.5.1.6.24 RX_ECC_SEC_DATA 寄存器

偏移地址：48h　复位值：00000000h

RX_ECC_SEC_DATA 寄存器的各字段说明如表中所示。

表 20-109 RX_ECC_SEC_DATA 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SEC_DATA | | | | | | | |
| 读（R）-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SEC_DATA | | | | | | | |
| 读（R）-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SEC_DATA | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEC_DATA | | | | | | | |
| 读（R）-0h | | | | | | | |

表 20-110 RX_ECC_SEC_DATA 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 31-0 | SEC_DATA | 读（R） | 0h | 经过ECC校正的数据<br>该寄存器有效的情况：无单比特错误、单比特错误<br>发生多比特错误则无效<br>复位类型：SYSRSn |

### 20.5.1.6.25 RX_ECC_LOG 寄存器

偏移地址：4Ch　复位值：0003h

RX_ECC_LOG 寄存器的各字段说明如表中所示。

表 20-111 RX_ECC_LOG 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | | | MBE | SBE |
| 读（R）-0h | | | | | | 读（R）-1h | 读（R）-1h |

表 20-112 RX_ECC_LOG 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-2 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 1 | MBE | 读（R） | 1h | ECC检测发现多比特错误<br>复位类型：SYSRSn |
| 0 | SBE | 读（R） | 1h | ECC检测发现单比特错误<br>复位类型：SYSRSn |

### 20.5.1.6.26 RX_FRAME_TAG_CMP 寄存器

偏移地址：50h　复位值：0000h

RX_FRAME_TAG_CMP 寄存器的各字段说明如表中所示。

表 20-113 RX_FRAME_TAG_CMP 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | BROADCAST_EN | CMP_EN |
| 读（R）-0h | | | | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TAG_MASK | | | | TAG_REF | | | |
| 读 / 写（R/W）-0h | | | | 读 / 写（R/W）-0h | | | |

表 20-114 RX_FRAME_TAG_CMP 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-10 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 9 | BROADCAST_EN | 读 / 写（R/W） | 0h | 广播使能位。置 1 时允许接收 ping 帧广播；收到标签的第 3 位为 1 时，无论标签比较结果如何都会触发 ping 标签匹配事件。<br>复位类型：SYSRSn |
| 8 | CMP_EN | 读 / 写（R/W） | 0h | 帧标签比较使能位。置 1 时，将输入帧标签与 TAG_REF 中存储的标签参考值进行比较；TAG_MASK、TAG_REF 与输入帧标签匹配时触发相应帧标签匹配事件。<br>复位类型：SYSRSn |
| 7-4 | TAG_MASK | 读 / 写（R/W） | 0h | 帧标签屏蔽位。某位为 0 时参与输入帧标签与 TAG_REF 的比较；为 1 时忽略该位。该屏蔽值仅用于非 ping 帧。<br>复位类型：SYSRSn |
| 3-0 | TAG_REF | 读 / 写（R/W） | 0h | 帧标签参考值。与 TAG_MASK 和输入帧标签比较时使用的参考标签；仅用于非 ping 帧。<br>复位类型：SYSRSn |

## 20.5.1.6.27 RX_PING_TAG_CMP 寄存器

偏移地址：52h　复位值：0000h

RX_PING_TAG_CMP 寄存器的各字段说明如表中所示。

表 20-115 RX_PING_TAG_CMP 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | BROADCAST_EN | CMP_EN |
| 读（R）-0h | | | | | | 读 / 写（R/W）-0h | 读 / 写（R/W）-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TAG_MASK | | | | TAG_REF | | | |
| 读 / 写（R/W）-0h | | | | 读 / 写（R/W）-0h | | | |

表 20-116 RX_PING_TAG_CMP 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-10 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 9 | BROADCAST_EN | 读 / 写（R/W） | 0h | 广播使能位。置 1 时允许接收 ping 帧广播；收到标签的第 3 位为 1 时，无论标签比较结果如何都会触发 ping 标签匹配事件。<br>复位类型：SYSRSn |
| 8 | CMP_EN | 读 / 写（R/W） | 0h | ping 标签比较使能位。置 1 时，将输入 ping 标签与 TAG_REF 中存储的标签参考值进行比较；TAG_MASK、TAG_REF 与输入 ping 标签匹配时触发 ping 标签匹配事件。<br>复位类型：SYSRSn |
| 7-4 | TAG_MASK | 读 / 写（R/W） | 0h | ping 标签屏蔽位。某位为 0 时参与输入 ping 标签与 TAG_REF 的比较；为 1 时忽略该位。该屏蔽值仅用于 ping 帧。<br>复位类型：SYSRSn |
| 3-0 | TAG_REF | 读 / 写（R/W） | 0h | ping 标签参考值。与 TAG_MASK 和输入 ping 标签比较时使用的参考标签；仅用于 ping 帧。<br>复位类型：SYSRSn |

## 20.5.1.6.28 RX_DLYLINE_CTRL 寄存器

偏移地址：60h　复位值：0000h

RX_DLYLINE_CTRL 寄存器的各字段说明如表中所示。

表 20-117 RX_DLYLINE_CTRL 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | RXD1_DLY | | | | | RXD0_DLY | |
| 读（R）-0h | 读 / 写（R/W）-0h | | | | | 读 / 写（R/W）-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXD0_DLY | | | RXCLK_DLY | | | | |
| 读 / 写（R/W）-0h | | | 读 / 写（R/W）-0h | | | | |

表 20-118 RX_DLYLINE_CTRL 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 14-10 | RXD1_DLY | 读 / 写（R/W） | 0h | 0：0个延迟单位<br>……<br>1F：31个延迟单位<br>复位类型：SYSRSn |
| 9-5 | RXD0_DLY | 读 / 写（R/W） | 0h | 0：0个延迟单位<br>……<br>1F：31个延迟单位<br>复位类型：SYSRSn |
| 4-0 | RXCLK_DLY | 读 / 写（R/W） | 0h | 0：0个延迟单位<br>……<br>1F：31个延迟单位<br>复位类型：SYSRSn |

## 20.5.1.6.29 RX_VIS_1 寄存器

偏移地址：70h　复位值：00000000h

RX_VIS_1 寄存器的各字段说明如表中所示。

表 20-119 RX_VIS_1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留（RESERVED） | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留（RESERVED） | | | | RX_CORE_STS | 保留（RESERVED） | | |
| 读（R）-0h | | | | 读（R）-0h | 读（R）-0h | | |

表 20-120 RX_VIS_1 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 31-4 | 保留（RESERVED） | 读（R） | 0h | 保留 |
| 3 | RX_CORE_STS | 读（R） | 0h | 该寄存器指示RX core状态。<br>0：正常工作<br>1：需要复位<br>仅指示type error和eof error<br>复位类型：SYSRSn |
| 2-0 | 保留（RESERVED） | 读（R） | 0h | 保留 |

## 20.5.1.6.30 RX_BUF_BASE_y 寄存器

偏移地址：0x0080 + y×2 (y = 0..15)　复位值：0000h

RX_BUF_BASE_y 寄存器的各字段说明如表中所示。

表 20-121 RX_BUF_BASE_y 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| BASE_ADDRESS | | | | | | | |
| 读（R）-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BASE_ADDRESS | | | | | | | |
| 读（R）-0h | | | | | | | |

表 20-122 RX_BUF_BASE_y 寄存器字段描述

| 位 | 字段名称 | 访问类型 | 复位默认值 | 字段描述 |
|---|---|---|---|---|
| 15-0 | BASE_ADDRESS | 读（R） | 0h | 接收数据缓冲区基地址。接收器使用的 16 字数据缓冲区的基地址。<br>复位类型：SYSRSn |

# 21 内部集成电路总线接口（I2C）

## 21.1 简介

I2C 模块支持任何从或主 I2C 兼容器件。图中显示了连接的多个 I2C 模块的示例，用于从一个设备到其他设备的双向传输。

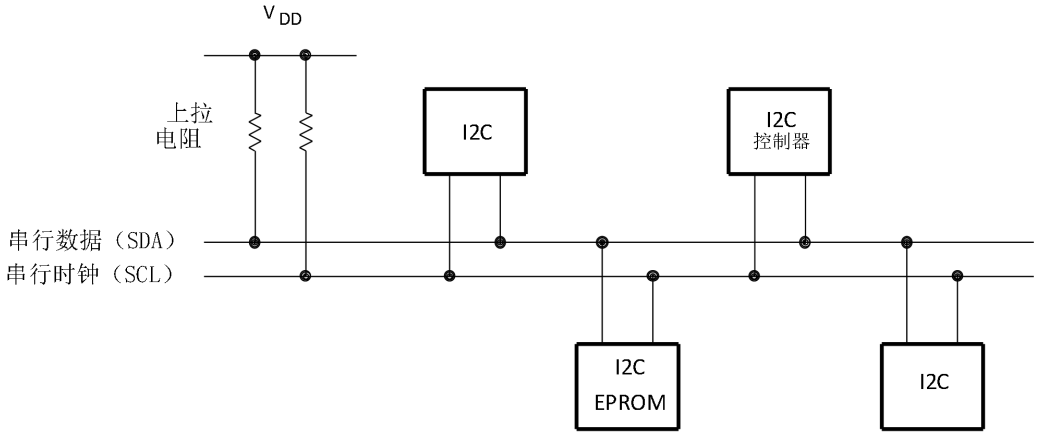



图 21-1 多 I2C 设备互联

## 21.2 主要特性

I2C 模块具有以下特性：

- 符合恩智浦半导体 I2C 总线规范（版本 2.1）
  - 支持 8 位格式传输
  - 7 位和 10 位寻址模式
  - 广播地址
  - 开始字节模式
  - 支持多个主发送器和从接收器
  - 支持多个从发送器和主接收器
  - 组合式主发送/接收和接收/发送模式
  - 数据传输速率从 10kbps 到 400kbps(快速模式)
- 接收 FIFO 和发送器 FIFO(16 深 x 8 位 FIFO)

- 支持 2 个 eCLIC 中断
  - I2Cx 中断-以下任何事件均可配置为生成 I2Cx 中断：
    - 发送数据就绪
    - 接收数据就绪
    - 寄存器访问就绪
    - 收到未确认标识
    - 仲裁失败
    - 检测到停止条件
    - 作为从设备寻址
  - I2Cx_FIFO 中断：
    - 发送 FIFO 中断
    - 接收 FIFO 中断
    - 模块启用和禁用功能
    - 自由的数据格式模式

## 21.3 功能描述

### 21.3.1 功能概述

连接到 I2C 总线的每个器件由唯一地址识别。根据设备的功能，每个设备既可以作为发射器，也可以作为接收器。在执行数据传输时，连接到 I2C 总线的器件也可被视为主机或从机。主设备是在总线上发起数据传输并产生时钟信号以允许该传输的设备。在此传输过程中，此主机寻址的任何设备都被视为从机。I2C 模块支持多主机模式，在这种模式下，一个或多个能够控制 I2C 总线的器件可以连接到同一 I2C 总线。

对于数据通信，I2C 模块有一个串行数据引脚(SDA)和一个串行时钟引脚(SCL)，如图所示。这两个引脚在 CPU 设备和连接到 I2C 总线的其他设备之间传输信息。SDA 和 SCL 引脚都是双向的，每个引脚都必须使用上拉电阻连接到正电源电压。当

总线空闲时，两个引脚均为高。这两个引脚的驱动器具有开漏配置，以执行所需的“线与”功能。

有两种主要的传输技术：

- 标准模式：仅发送 n 个数据值，其中 n 是你在 I2C 模块寄存器中编程的值。
- 重复模式：持续发送数据值，直到你使用软件启动停止条件或新的启动条件。

I2C 模块由以下主要模块组成：

- 一个串行接口：一个数据引脚(SDA)和一个时钟引脚(SCL)
- 数据寄存器和 FIFO，用于临时保存在 SDA 引脚和 CPU 之间传输的接收数据和传输数据
- 控制和状态寄存器，使 CPU 能够访问 I2C 模块寄存器和 FIFO 的外设总线接口
- 时钟同步器，用于同步 I2C 输入时钟(来自设备时钟生成器)和 SCL 引脚上的时钟，同步与不同时钟速度的主机的数据传输
- 预分频器，用于对驱动到 I2C 模块的输入时钟进行分频
- SDA 和 SCL 两个引脚上的噪声滤波器
- 仲裁器，用于处理 I2C 模块作为主机时与其他主机之间的总线仲裁；中断生成逻辑，用于向 CPU 发送中断请求
- FIFO 中断生成逻辑，以便 FIFO 访问可以与 I2C 模块中的数据接收和数据传输同步

图中显示了在非 FIFO 模式下用于发送和接收的四个寄存器。CPU 将数据写入 I2CDXR，并从 I2CDRR 读取接收的数据。将 I2C 模块配置为发送器时，写入 I2CDXR 的数据将复制到 I2CXSR，并在 SDA 引脚上逐位移出。当 I2C 模块配置为接收器时，接收的数据将移位到 I2CRSR，然后复制到 I2CDRR。

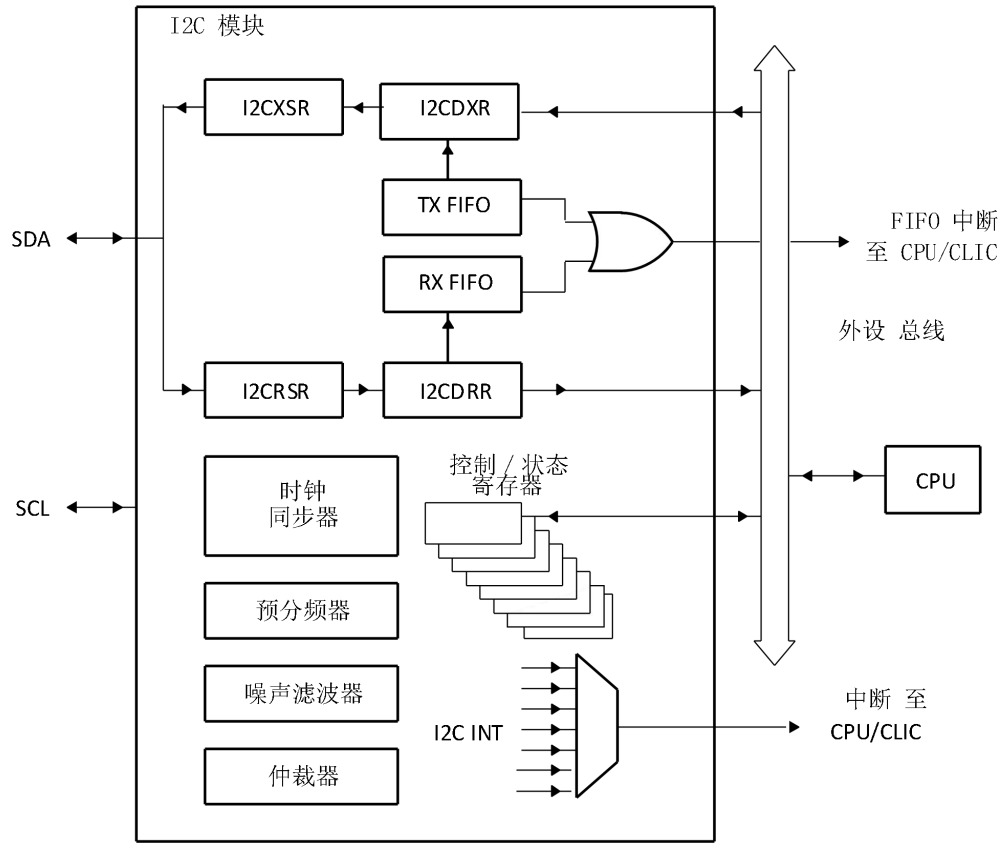



图 21-2 I2C 框架图

## 21.3.2 时钟生成

I2C 模块时钟决定 I2C 模块的工作频率。I2C 模块中的可编程预分频器将 SYSCLK 分频以产生 I2C 模块时钟，该 I2C 模块时钟进一步分频以产生 SCL 引脚上的 I2C 主时钟。图中显示了 I2C 模块的时钟产生图。

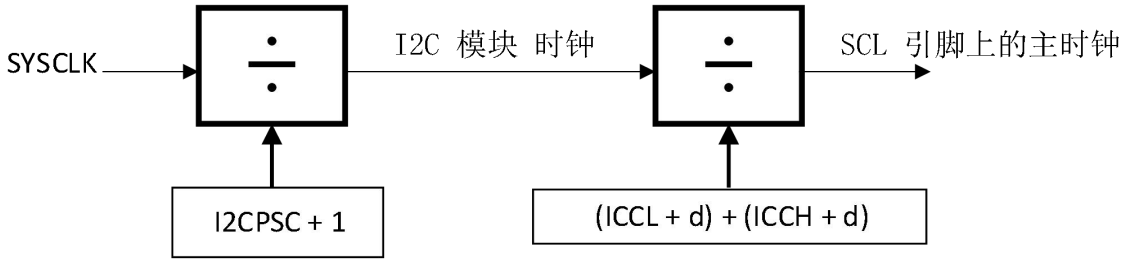



图 21-3 I2C 时钟模块

要指定分频值，请初始化预分频寄存器 I2CPSC 的 IPSC 字段。由此产生的频率为：

$$\text{I2C 模块时钟（Fmod）} = \frac{\text{SYSCLK}}{\text{I2CPSC} + 1} \qquad (21\text{-}1)$$

必须在 I2C 模块处于复位状态（I2CMDR 中的 IRS=0）时初始化预分频器。仅当 IRS 更改为 1 时，预分频才生效。当 IRS=1 时，更改 IPSC 值不起作用。当 I2C 模块配置为 I2C 总线上的主机时，主时钟出现在 SCL 引脚上。该时钟控制 I2C 模块和从机之间的通信时序。如图所示，I2C 模块中的第二个时钟分频器对模块时钟进行分频以产生主时钟。时钟分频器使用 I2CCLKL 的 ICCL 值来细分模块时钟信号的低部分，并使用 I2CCLKH 的 ICCH 值来细分模块时钟信号的高部分。

### 21.3.3 I2C 时钟驱动寄存器

当 I2C 模块工作在主模式时，I2C 模块时钟被进一步分频，以用作 SCL 引脚上的主时钟。如图所示，主时钟的形状取决于两个分频数值：

- I2CCLKL 中的 ICCL。对于每个主时钟周期，ICCL 确定信号处于低电平的时间量。
- I2CCKLH 中的 ICCH。对于每个主时钟周期，ICCH 确定信号为高电平的时间量。

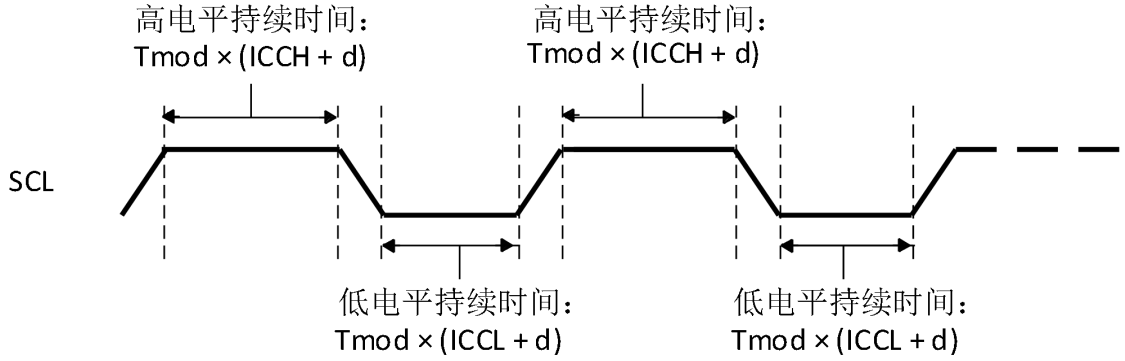



图 21-4 时钟分频值的作用（ICCL 和 ICCH）

主时钟周期(TMST)是 I2C 模块时钟(Tmod)周期的倍数：

$$\text{主时钟周期（Tmst）} = \frac{(\text{ICCH} + \text{d}) + (\text{ICCL} + \text{d})}{\text{I2C 模块时钟（Fmod）}} \qquad (21\text{-}2)$$

其中 d 取决于分频数值 IPSC，如表所示。IPSC 在 I2CPSC 寄存器中描述。

表 21-1 延迟 d 对分频值 IPSC 的依赖关系

| IPSC | d |
|---|---|
| 0 | 7 |
| 1 | 6 |
| >1 | 5 |

## 21.4 模块详细操作

### 21.4.1 引脚配置

必须将 GPIO 多路复用器寄存器配置为将该外设连接到设备引脚。为避免引脚上的毛刺，必须首先配置 GPyGMUX 位(同时将相应的 GPyMUX 位保持为默认零)，然后将 GPyMUX 寄存器写入所需的值。某些 IO 功能由独立于该外设的 GPIO 寄存器设置定义。对于输入信号，必须通过将适当的 GPxQSELn 寄存器位设置为 11b，将 GPIO 输入限定设置为异步模式。内部上拉可在 GPyPUD 寄存器中配置。

### 21.4.2 输入和输出电压电平

主设备为传输的每个数据位产生一个时钟脉冲。由于可以连接到 I2C 总线的各种不同技术器件，逻辑 0(低)和逻辑 1(高)的电平不是固定的，取决于相关的 VDD 电平。

### 21.4.3 上拉电阻

所选上拉电阻必须满足 I2C 标准时序。在大多数情况下，2.2kΩ的总总线电阻就足够了。还建议在 SCL 和 SDA 引脚上使用的上拉阻值相匹配。

### 21.4.4 数据有效性

在时钟的高电平期间，SDA 上的数据必须稳定(见图)。在普通数据位传输期间，SDA 只允许在 SCL 为低电平时改变；起始条件和停止条件除外。

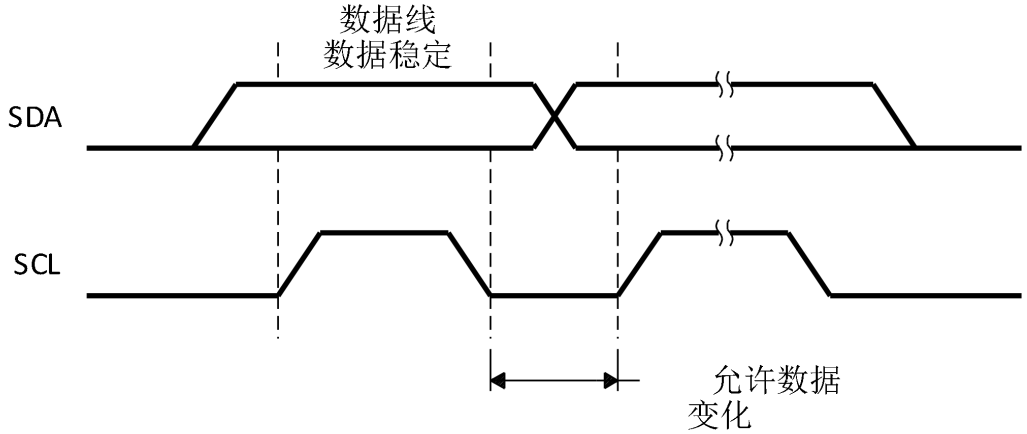



图 21-5 I2C 总线比特传输

### 21.4.5 操作模式

I2C 模块有四种基本工作模式，可支持作为主机和从机的数据传输。模式的名称和说明见表。

如果 I2C 模块是主机，则 I2C 模块从主机发送器开始，通常发送特定从机的地址。向从机提供数据时，I2C 模块必须保持为主机发送器。要从从机接收数据，必须将 I2C 模块更改为主机-接收器模式。

如果 I2C 模块是从机，则 I2C 模块从从机开始，通常在 I2C 模块识别来自主机的从机地址时发送确认。如果主机向 I2C 模块发送数据，则该模块必须保持从机接收器。如果主机已从 I2C 模块请求数据，则必须将该模块更改为从机-发送器模式。

表 21-2 操作模式

| 操作模式 | 描述 |
| --- | --- |
| 从机接收模式 | I2C模块是从机，从主机接收数据。<br>所有从机都在此模式下开始。在此模式下，SDA上接收的串行数据位与主机产生的时钟脉冲一起移位。作为从机，I2C模块不会产生时钟信号，但在接收到一个字节后需要器件干预(I2CSTR中的RSFULL=1)时，可以将SCL保持为低电平。 |
| 从机发送模式 | I2C模块是从机，向主机发送数据。<br>该模式只能从从机-接收器模式进入；I2C模块必须首先从主机接收命令。使用任何7位/10位寻址格式时，如果从机地址字节与地址相同(在I2COAR中)且主机已发送R/W=1，则I2C模块进入从机发送器模式。作为从机发送器，I2C模块随后利用主机产生的时钟脉冲移出SDA上的串行数据。作为从机时，I2C模块不会产生时钟信号，但在一个字节发送后需要器件干预(I2CSTR中的XSMT=0)时，它可以将SCL保持为低电平。 |
| 主机接收模式 | I2C模块是主机，从从机接收数据。<br>该模式只能从主机-发送器模式进入；I2C模块必须首先向从机发送命令。当使用任何7位/10位寻址格式时，I2C模块在发送从属地址字节和R/W=1后进入主-接收器模式。<br>SDA上的串行数据位通过SCL上I2C模块产生的时钟脉冲移位到I2C模块。在接收到一个字节后，当需要器件干预(I2CSTR中的RSFULL=1)时，时钟脉冲被禁止，SCL保持低电平。 |
| 主机发送模式 | I2C模块是主机，向从机发送控制信息和数据。<br>所有主机都在此模式下开始。在此模式下，以任何7位/10位寻址格式汇编的数据在SDA上移出。位移位与SCL上的I2C模块产生的时钟脉冲同步。发送一个字节后，当需要器件干预(I2CSTR中的XSMT=0)时，时钟脉冲被禁止，SCL保持低电平。 |

总而言之，SCL 在以下情况下保持低电平：

- 在从机-接收器模式下，当检测到溢出条件时(RSFULL=1)。
- 当检测到下溢条件(XSMT=0)时，在从机-发送器模式下。

当 I2C 主机请求数据时，I2C 从机接受并提供数据。

- 要在从机-接收器模式下释放 SCL，请从 I2CDRR 读取数据。
- 要在从机-发送器模式下释放 SCL，请将数据写入 I2CDXR。
- 要在不处理数据的情况下强制释放，请使用 I2CMDR.IRS 位重置模块。

表 21-3 由 I2CMDR 的 RM、STT 和 STP 位定义的主发送器/接收器总线行为

| RM | STT | STP | 总线活动(1) | 说明 |
|---|---|---|---|---|
| 0 | 0 | 0 | 无 | 无活动。 |
| 0 | 0 | 1 | P | 停止条件。 |
| 0 | 1 | 0 | S-A-D..(n)..D. | 起始条件、从设备地址、n 个数据字节（n 为 I2CCNT 寄存器中的值）。 |
| 0 | 1 | 1 | S-A-D..(n)..D-P | 起始条件、从设备地址、n 个数据字节、停止条件（n 为 I2CCNT 寄存器中的值）。 |
| 1 | 0 | 0 | 无 | 无活动。 |
| 1 | 0 | 1 | P | 停止条件。 |
| 1 | 1 | 0 | S-A-D-D-D. | 重复模式传输：产生起始条件、发送从设备地址，然后连续传输数据，直到出现停止条件或下一个起始条件。 |
| 1 | 1 | 1 | 无 | 保留的位组合（无活动）。 |

## 21.4.6 开始和结束条件

当 I2C 模块配置为 I2C 总线上的主机时，启动和停止条件可由 I2C 模块生成。如图所示：

- 启动条件定义为当 SCL 为高时，SDA 线上从高到低的转变。主机驱动该条件以指示数据传输的开始。
- 停止条件定义为当 SCL 为高时，SDA 线上从低到高的转变。主机驱动该条件以指示数据传输的结束。

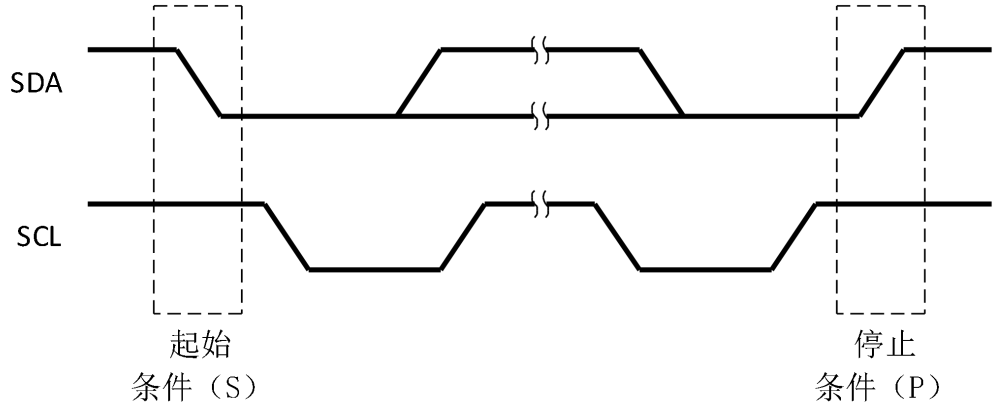


图 21-6 I2C 开始和结束条件

在启动条件之后和随后的停止条件之前，I2C 总线被视为忙，并且 I2CSTR 的总线忙(BB)位为 1。在停止条件和下一个启动条件之间，该总线被认为是空闲的，而 BB 为 0。

为了使 I2C 模块在启动条件下开始数据传输，I2CMDR 中的主模式位(MST)和启动条件位(STT)必须都为 1。为了使 I2C 模块以停止条件结束数据传输，停止条件位(STP)必须设置为 1。当 BB 位设置为 1 且 STT 位设置为 1 时，会产生重复的启动条件。

在复位状态(IRS=0)时，I2C 外设无法检测到启动或停止条件。当 I2C 外设处于重置状态(IRS=0)时，BB 位保持清除状态(BB=0)。当 I2C 外设退出重置(IRS 设置为 1)时，在检测到启动或停止条件之前，BB 位不会正确反映 I2C 总线状态。

在使用 I2C 启动第一次数据传输之前，请遵循以下步骤：

1. 通过将 IRS 位设置为 1，使 I2C 外设退出重置状态后，等待一段时间，此时间段大于最长数据传输所需的总时间，从而确保 I2C 总线上至少出现一个启动或停止条件，并已被 BB 位捕获。在此周期之后，BB 位将正确反映 I2C 总线的状态。
2. 在继续操作之前，检查 BB 位并确认 BB=0(总线不忙)。
3. 开始数据传输。

在传输之间不重置 I2C 外设可确保 BB 位反映实际的总线状态。如果用户必须在传输之间对 I2C 外设进行重置，则每次使 I2C 外设退出重置时重复步骤 1 至 3。

### 21.4.7 非重复和重复模式

非重复模式：

- 当 I2CMDR.RM=0 时，I2C 模块配置为非重复模式。
- I2CCNT 寄存器确定要发送或接收的字节数。
- 如果 I2CMDR 中的 STP=0，则当内部数据计数器倒计时至 0 时，ARDY 位被置 1。

- 如果 STP=1，则 ARDY 位不会置位，并且当内部数据计数器倒计时到 0 时，I2C 模块会产生一个停止条件。

重复模式：

- 当 I2CMDR.RM=1 时，I2C 模块配置为重复模式。
- I2CCNT 寄存器内容不确定要发送或接收的字节数。
- 发送或接收的字节数可由软件控制。
- 在每个字节的发送和接收结束时设置 ARDY 位。

### 21.4.8 串口数据格式

图中显示了I2C总线上的数据传输示例。I2C模块支持1至8位数据值。在图中，传输的是 8 位数据。SDA 线上的每一位相当于 SCL 线上的 1 个脉冲，值总是以最高有效位(MSB)为先进行传输。可以发送或接收的数据值的数量是不受限制的。图中使用的串行数据格式为 7 位寻址格式。I2C 模块支持图中所示的格式，并在附图后面的段落中描述。

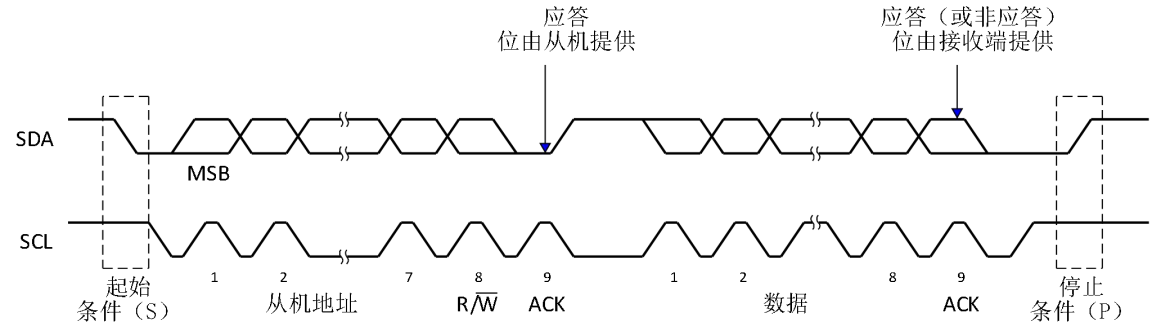



图 21-7 I2C 模块数据传输(7 位寻址，显示 8 位数据配置)

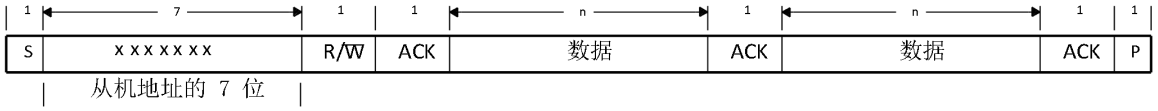



图 21-8 I2C 模块 7 位寻址格式(I2CMDR 中 FDF=0，XA=0)




图 21-9 I2C 模块 10 位寻址格式(I2CMDR 中 FDF=0，XA=1)

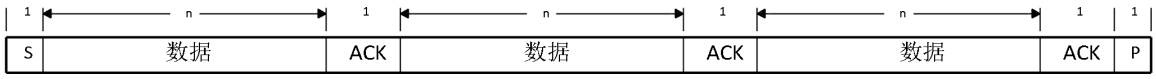



图 21-10 I2C 模块自由数据格式(I2CMDR 中 FDF=1)

## 21.4.9 时钟同步

正常情况下，只有一个主器件产生时钟信号 SCL。然而，在仲裁过程中，有两个或更多主机，时钟必须同步，以便可以比较数据输出。图中说明了时钟同步。SCL 的 WIRED-AND 属性意味着首先在 SCL 上产生低电平周期的设备凌驾于其他设备之上。在这种从高到低的转变中，其他器件的时钟发生器被迫开始低周期。SCL 由低电平周期最长的器件保持低电平。完成低周期的其他设备必须等待 SCL 释放，然后才能开始高周期。获得 SCL 上的同步信号，其中最慢的设备确定低周期的长度，而最快的设备确定高周期的长度。

如果器件将时钟线拉下更长时间，结果是所有时钟发生器都必须进入等待状态。以这种方式，从设备减慢快速主设备的速度，而慢速设备产生足够的时间来存储接收的字节或准备要传输的字节。

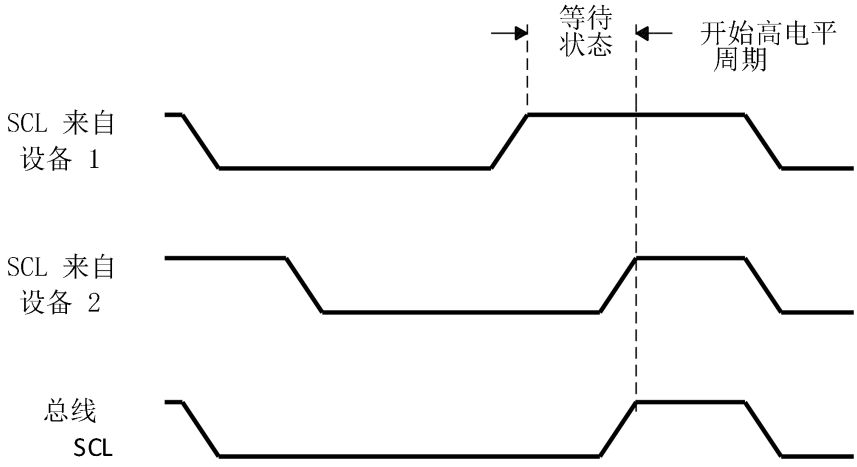



图 21-11 仲裁过程中两个 I2C 时钟发生器的同步

## 21.4.10 仲裁

如果两个或多个主发送器尝试几乎同时在同一总线上开始传输，则调用仲裁程序。仲裁程序使用竞争发送器在串行数据总线(SDA)上提供的数据。图中说明了两个设备

之间的仲裁过程。将 SDA 线路释放为高的第一个主发送器被另一个驱动 SDA 低的主发送器否决。仲裁程序将优先权给予传输具有最低二进制值的串行数据流的设备。如果两个或更多设备发送相同的第一个字节，则仲裁将在后续字节上继续。

如果在串行传输期间，当向 SDA 发送重复的开始条件或停止条件时，仲裁程序仍在进行中，则涉及的主发射器必须在格式帧中的相同位置发送重复的开始条件或停止条件。下列各项之间不允许进行仲裁：

- 重复开始条件和数据位之间不允许仲裁
- 停止条件和数据位
- 重复开始条件和停止条件

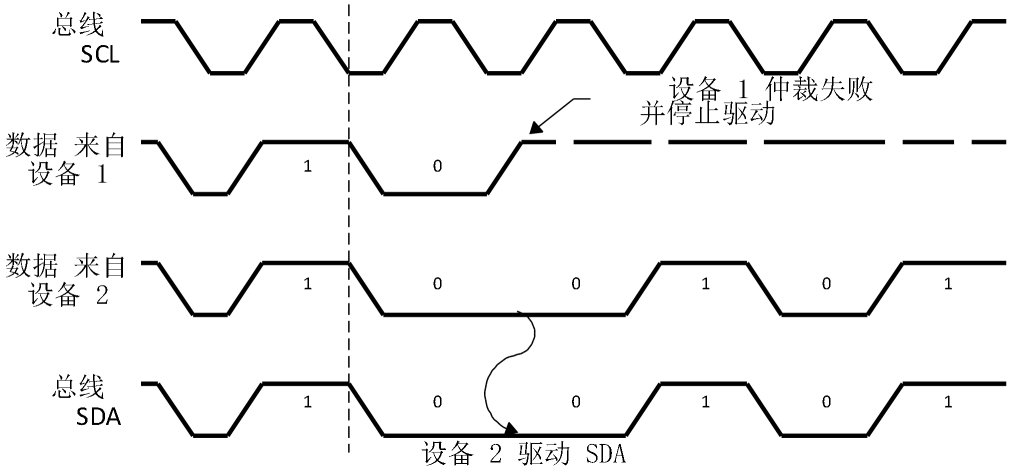



图 21-12 两个主发射器的仲裁程序

## 21.4.11 数字回环模式

I2C 模块支持一种称为数字环回的自检模式，该模式通过设置 I2CMDR 寄存器中的 DLB 位来使能。在此模式下，从 I2CDXR 寄存器发送的数据在 I2CDRR 寄存器中接收。数据遵循内部路径，需要 n 个周期才能到达 I2CDRR，计算如下：

$$n = \frac{8 \times SYSCLK}{I2C\ 模块时钟（Fmod）} \quad （21\text{-}3）$$

发送时钟和接收时钟相同。外部 SDA 引脚上看到的地址是 I2COAR 寄存器中的地址。图中显示了数字环回模式下的信号路由。

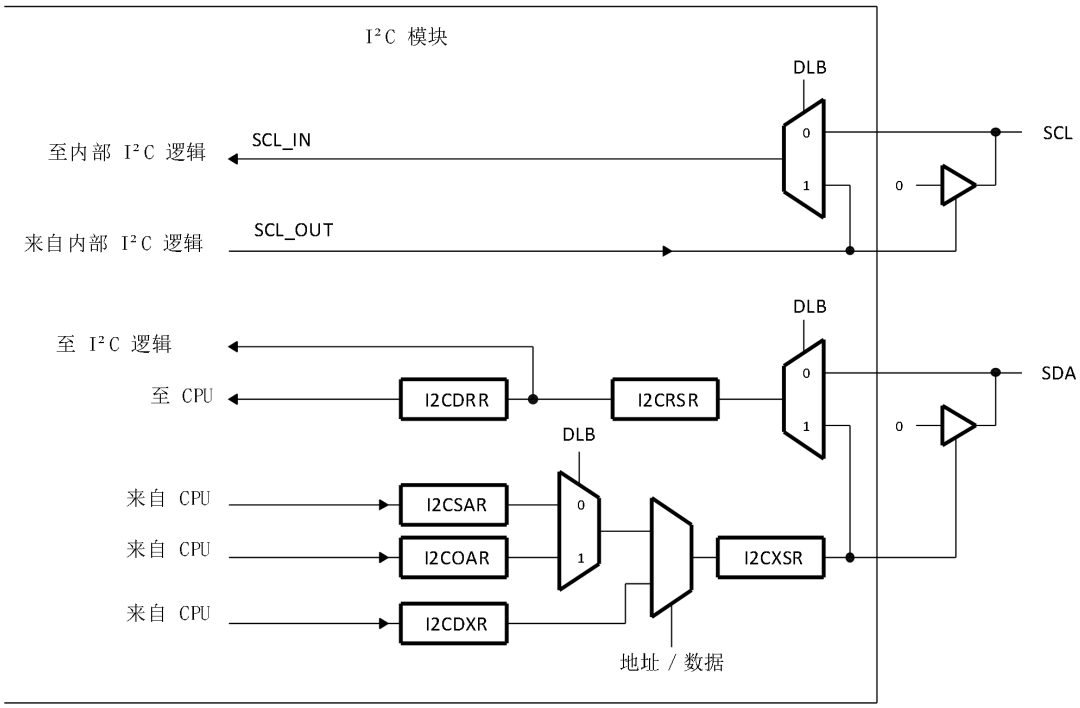



图 21-13 显示数字环回模式(DLB)位影响的引脚图

## 21.4.12 NACK 位生成

当 I2C 模块是接收器(主机或从机)时，I2C 模块可以确认或忽略发送器发送的位。要忽略任何新位，I2C 模块必须在总线上的确认周期期间发送一个无确认(NACK)位。表中总结了允许 I2C 模块发送 NACK 位的各种方法。

表 21-4 生成 NACK 位的方法

| 条件 | NACK位生成选项 |
|---|---|
| 从机接收 模式 | 允许溢出条件(I2CSTR中的RSFULL=1)重置模块(I2CMDR中的IRS=0)<br>在要接收的最后一个数据位的上升沿之前设置I2CMDR的NACKMOD位 |
| 主机-接收器模式和重复模式(I2CMDR中的Rm=1) | 生成停止条件(I2CMDR中的STP=1)<br>重置模块(I2CMDR中的IRS=0)<br>在要接收的最后一个数据位的上升沿之前设置I2CMDR的NACKMOD位 |
| 主机-接收器模式和非重复模式(I2CMDR中的Rm=0) | 如果I2CMDR中的STP=1，则允许内部数据计数器倒计数到0，从而强制停止条件。<br>如果STP=0，则使STP=1以生成停止条件<br>重置模块(I2CMDR中的IRS=0)<br>设置STP=1以生成停止条件<br>将I2CMDR的NACKMOD位设置在您想要接收的最后一个数据位的上升沿之前 |

## 21.5 I2C 的中断请求生成

每个 I2C 模块可以产生两个 CPU 中断。

### 21.5.1 基础 I2C 中断请求

I2C 模块生成表中所述的中断请求。如图所示，所有请求都通过仲裁器被多路传输到向 CPU 发出的单个 I2C 中断请求。每个中断请求在状态寄存器(I2CSTR)中有一个标志位，在中断使能寄存器(I2CIER)中有一个使能位。当指定事件之一发生时，标志位被设置。如果相应的使能位为 0，则中断请求被阻止。如果使能位为 1，则请求作为 I2C 中断转发到 CPU。I2C 中断是 CPU 的可屏蔽中断之一。与任何可屏蔽中断请求一样，如果该请求在 CPU 中正确启用，则 CPU 执行相应的中断服务例程(I2CINT1A_ISR)。I2C 中断的 I2CINT1A_ISR 可通过读取中断源寄存器 I2CISRC 来确定中断源。然后，I2CINT1A_ISR 可以分支到相应子程序。

在 CPU 读取 I2CISRC 之后，会发生以下事件：

1. I2CSTR 中源中断的标志被清零。例外：读取 I2CISRC 时，I2CSTR 中的 ARDY、RRDY 和 XRDY 位不会被清除。要清除这些位中的一位，请向该位写入 1。
2. 仲裁器确定剩余的中断请求中哪一个具有最高优先级，将该中断的代码写入 I2CISRC，并将中断请求转发到 CPU。

表 21-5 基础 I2C 中断请求的描述

| I2C 中断请求 | 中断源 |
|---|---|
| XRDYINT | 发送就绪条件：数据发送寄存器（I2CDXR）已准备好接收新数据，因为之前的数据已从 I2CDXR 复制到发送移位寄存器（I2CXSR）。<br>除了使用 XRDYINT 中断之外，CPU 还可以轮询状态寄存器 I2CSTR 中的 XRDY 位。在 FIFO 模式下，不得使用 XRDYINT，而应使用 FIFO 中断。 |
| PRDYINT | 接收就绪条件：数据接收寄存器（I2CDRR）已准备好被读取，因为数据已从接收移位寄存器（I2CRSR）复制到 I2CDRR。<br>除了使用 RRDYINT 中断之外，CPU 还可以轮询 I2CSTR 中的 RRDY 位。在 FIFO 模式下，不得使用 RRDYINT，而应使用 FIFO 中断。 |
| ARDYINT | 寄存器访问就绪条件：I2C 模块寄存器已准备好被访问，因为先前编程写入的地址、数据和命令值均已使用完毕。<br>产生 ARDYINT 中断的具体事件，与置位 I2CSTR 中 ARDY 位的事件相同。<br>除了使用 ARDYINT 中断之外，CPU 还可以轮询 ARDY 位。 |

| I2C 中断请求 | 中断源 |
| --- | --- |
| NACKINT | 无应答条件：I2C 模块被配置为主机-发送器，但未收到来自从机-接收器的应答。<br>除了使用 NACKINT 中断之外，CPU 还可以轮询 I2CSTR 中的 NACK 位。 |
| ARBLINT | 仲裁丢失条件：I2C 模块在与另一个主机-发送器的仲裁竞争中失利。<br>除了使用 ARBLINT 中断之外，CPU 还可以轮询 I2CSTR 中的 ARBL 位。 |
| SCDINT | 检测到停止条件：在 I2C 总线上检测到了 STOP 条件（停止位）。<br>除了使用 SCDINT 中断之外，CPU 还可以轮询状态寄存器 I2CSTR 中的 SCD 位。 |
| AASINT | 被寻址为从机条件：I2C模块已被I2C总线上的另一个主机寻址为从机设备。<br>除了使用AASINT中断之外，CPU还可以轮询状态寄存器I2CSTR中的AAS位。 |

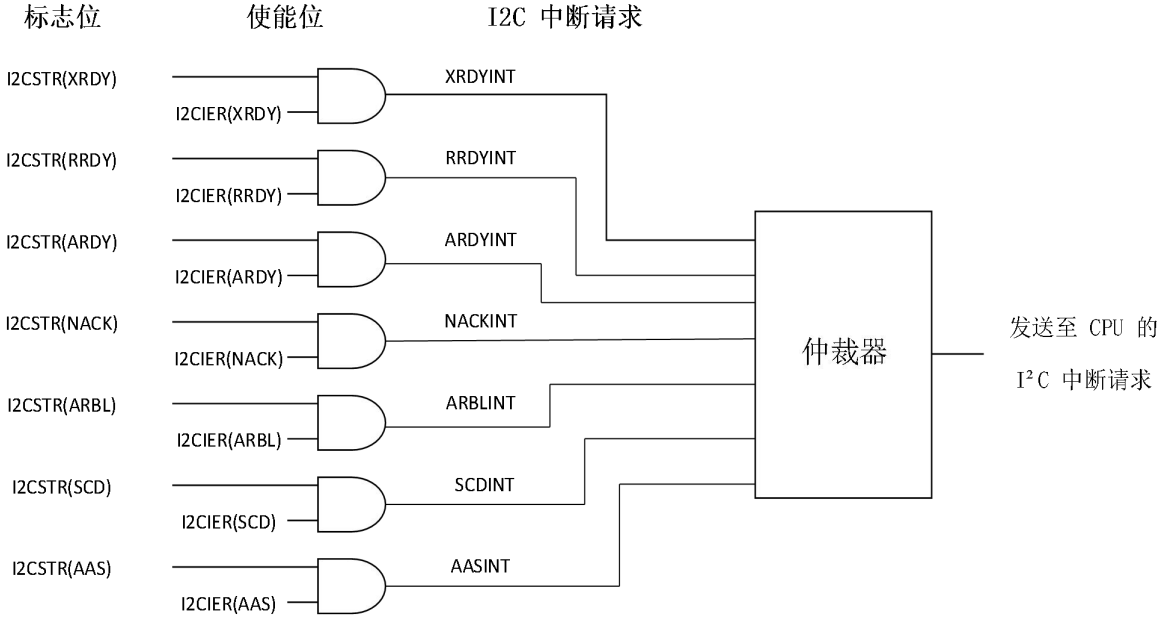



图 21-14 使能 I2C 中断请求的路径

基本 I2C 中断请求的优先级按优先级从高到低的顺序列出：

1. ARBLINT
2. NACKINT
3. ARDYINT
4. RRDYINT
5. XRDYINT
6. SCDINT

7. AASINT

如果事务在字节中间中止，正常的传输中断时序会使过时数据保留在传输缓冲区中。要避免这种情况，请设置 I2CEMDR 寄存器中的 FCM 位。此位设置时，仅当需要数据进行总线处理时才会产生发送数据就绪中断。在主机模式下，当接收到地址字节的 ACK 时，首先产生中断。在从机模式下，当地址匹配时，首先产生中断。当确认数据时，会产生进一步的中断。在此模式下，XRDY 在发送就绪中断的同时被置位。I2C 模块在 I2CEMDR 寄存器中有一个向后兼容位(BC)。图中的时序图展示了将后向兼容位配置为从发送器时，向后兼容位对 I2C 模块寄存器和中断的影响。

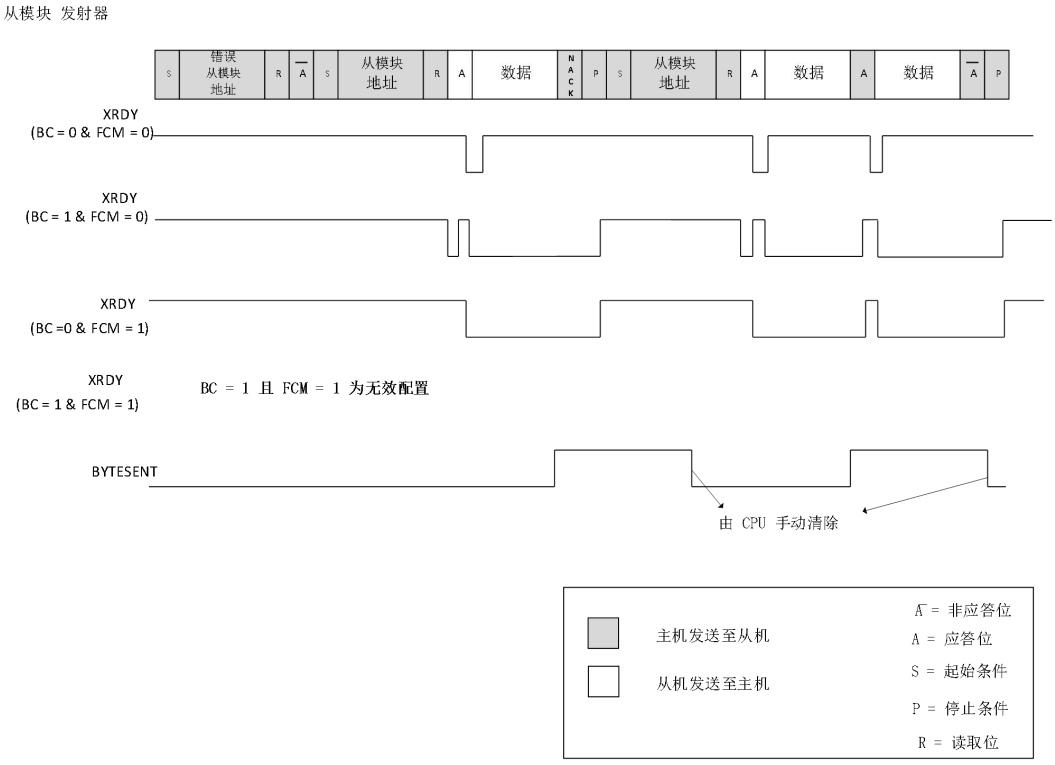



图 21-15 向后兼容模式和前向兼容位，从发送器

## 21.5.2 I2C FIFO 中断

除 7 个基本 I2C 中断外，发送和接收 FIFO 均具有产生中断(I2CINT2A)的能力。发送 FIFO 可配置为在发送定义数量的字节(最多 16 个)后生成中断。接收 FIFO 可配置为在接收定义数量的字节(最多 16 个)后生成中断。这两个中断源一起或运算为单个可屏蔽 CPU 中断。图中显示了 I2C FIFO 中断的结构。然后，中断服务例程可以读取

FIFO 中断状态标志，以确定中断来自哪个源。参见 I2C 发送 FIFO 寄存器(I2CFFTX)和 I2C 接收 FIFO 寄存器(I2CFFRX)说明。

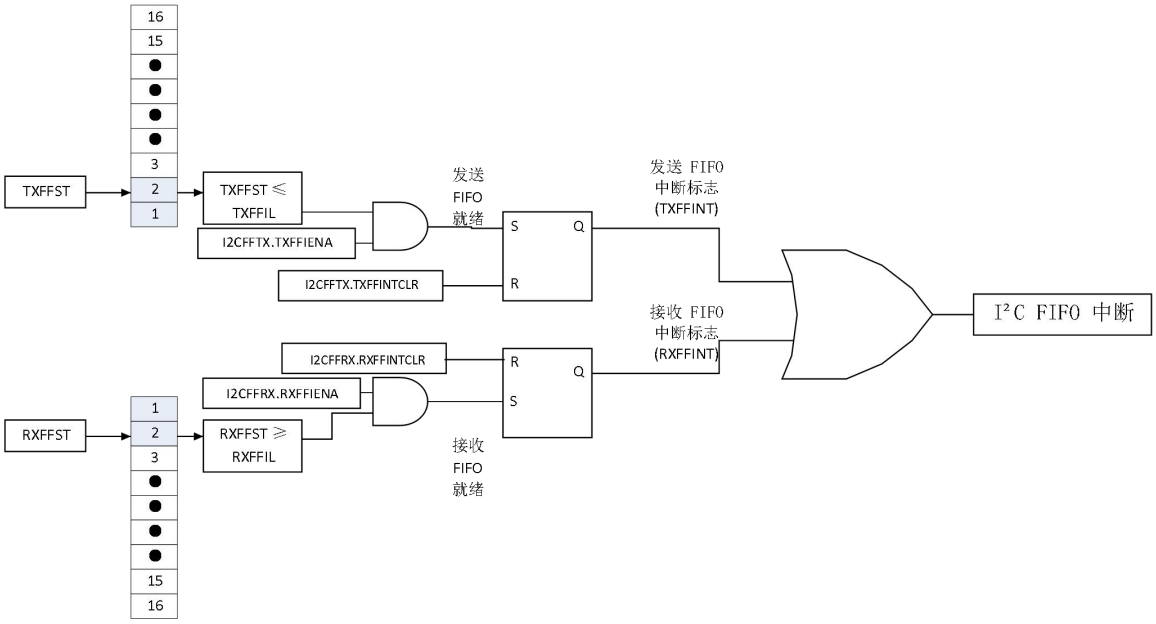



图 21-16 I2C FIFO 中断

# 21.6 寄存器描述

## 21.6.1 寄存器基地址

表 21-6 I2C 寄存器基地址表

| 比特位域名 | | DriverLib名 | 基地址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| **实例** | **结构** | | | | | |
| I2caRegs | I2C_REGS | I2CA_BASE | 0x0069_0000 | - | YES | YES |
| I2cbRegs | I2C_REGS | I2CA_BASE | 0x0069_0080 | - | YES | |

## 21.6.2 I2C_REGS 寄存器

表 21-7 I2C 寄存器

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 00h | I2COAR | BGCRC Enable |
| 02h | I2CIER | BGCRC Control register 1 |
| 04h | I2CSTR | BGCRC Control register 2 |
| 06h | I2CCLKL | Start address for the BGCRC check |
| 08h | I2CCLKH | Seed for CRC calculation |
| 0Ah | I2CCNT | Golden CRC to be compared against |
| 0Ch | I2CDRR | CRC calculated |

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 0Eh | I2CSAR | Current address register |
| 10h | I2CDXR | BGCRC windowed watchdog configuration |
| 12h | I2CMDR | BGCRC windowed watchdog min value |
| 14h | I2CISRC | BGCRC windowed watchdog max value |
| 16h | I2CEMDR | BGCRC windowed watchdog count |
| 18h | I2CPSC | BGCRC NMI flag register |
| 40h | I2CFFTX | BGCRC NMI flag clear register |
| 42h | I2CFFRX | BGCRC NMI flag force registe<br>r |

下表列出了本节使用的访问类型编码。

表 21-8 I2C_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |
| W1C | W<br>1C | 写<br>1清除 |
| W1S | W<br>1S | 写<br>1置位 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |
| 寄存器数组变量 | | |
| i、j、k、l、m、n | | 用于寄存器名称、偏移地址或地址时，表示属于重复寄存器组的寄存器数组值。这些组按层次组织，数组用公式表示。 |
| y | | 用于寄存器名称、偏移地址或地址时，表示寄存器数组值。 |

## 21.6.2.1 I2COAR 寄存器

偏移地址：0h　复位值：0000h

I2C 自身地址。

表 21-9 I2COAR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | OAR | |
| R-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OAR | | | | | | | |
| R/W-0h | | | | | | | |

表 21-10 I2COAR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-10 | RESERVED | R | 0h | 保留。 |
| 9-0 | OAR | R/W | 0h | 7 位寻址模式（I2CMDR 中 XA = 0）：<br>00h—7Fh：位 6—0 提供 I2C 模块的 7 位从机地址。位 9—7 应写入 0。<br>10 位寻址模式（I2CMDR 中 XA = 1）：<br>000h—3FFh：位 9—0 提供 I2C 模块的 10 位从机地址。<br>复位类型：SYSRSn。 |

### 21.6.2.2 I2CIER 寄存器

偏移地址：2h　复位值：0000h

I2C 中断使能。

表 21-11 I2CIER 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | AAS | SCD | XRDY | RRDY | ARDY | NACK | ARBL |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 21-12 I2CIER 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-7 | RESERVED | R | 0h | 保留。 |
| 6 | AAS | R/W | 0h | 被寻址为从机中断使能。<br>0h (R/W)：禁用中断请求。<br>1h (R/W)：使能中断请求。<br>复位类型：SYSRSn。 |
| 5 | SCD | R/W | 0h | 检测到 STOP 条件中断使能。<br>0h (R/W)：禁用中断请求。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1h (R/W)：使能中断请求。<br>复位类型：SYSRSn。 |
| 4 | XRDY | R/W | 0h | 发送数据就绪中断使能。<br>使用 FIFO 模式时，不应设置此位。<br>0h (R/W)：禁用中断请求。<br>1h (R/W)：使能中断请求。<br>复位类型：SYSRSn。 |
| 3 | RRDY | R/W | 0h | 接收数据就绪中断使能。<br>使用 FIFO 模式时，不应设置此位。<br>0h (R/W)：禁用中断请求。<br>1h (R/W)：使能中断请求。<br>复位类型：SYSRSn。 |
| 2 | ARDY | R/W | 0h | 寄存器访问就绪中断使能。<br>0h (R/W)：禁用中断请求。<br>1h (R/W)：使能中断请求。<br>复位类型：SYSRSn。 |
| 1 | NACK | R/W | 0h | 非应答中断使能。<br>0h (R/W)：禁用中断请求。<br>1h (R/W)：使能中断请求。<br>复位类型：SYSRSn。 |
| 0 | ARBL | R/W | 0h | 仲裁丢失中断使能。<br>0h (R/W)：禁用中断请求。<br>1h (R/W)：使能中断请求。<br>复位类型：SYSRSn。 |

### 21.6.2.3 I2CSTR 寄存器

偏移地址：4h　复位值：0410h

I2C 状态。

表 21-13 I2CSTR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | SDIR | NACKSNT | BB | RSFULL | XSMT | AAS | AD0 |
| R-0h | R/W1C-0h | R/W1C-0h | R-0h | R-0h | R-1h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | BYTESENT | SCD | XRDY | RRDY | ARDY | NACK | ARBL |
| R-0h | R/W1C-0h | R/W1C-0h | R-1h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |

表 21-14 I2CSTR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 14 | SDIR | R/W1C | 0h | 从机方向位。<br>0h (R/W)：I2C 未被寻址为从机发送器。以下任一事件会清除 SDIR：<br>• 手动清除，向此位写入 1。<br>• 使能数字回环模式。<br>• I2C 总线上出现 START 或 STOP 条件。<br>1h (R/W)：I2C 被寻址为从机发送器。<br>复位类型：SYSRSn。 |
| 13 | NACKSNT | R/W1C | 0h | NACK 已发送位，用于 I2C 模块的接收模式。使用 NACK 模式时，此位会受到影响（参见 NACKMOD 的说明）。<br>0h (R/W)：未发送 NACK。以下任一事件会清除 NACKSNT：<br>• 手动清除，向此位写入 1。<br>• I2C 模块复位，包括向 I2CMDR 的 IRS 位写入 0，或整个器件复位。<br>1h (R/W)：已发送 NACK，即在 I2C 总线的应答周期内发送了一个非应答位。<br>复位类型：SYSRSn。 |
| 12 | BB | R | 0h | 总线忙位，指示 I2C 总线是忙碌还是空闲、可进行下一次数据传输。更多说明请参阅表后的段落。<br>0h (R/W)：总线空闲。以下任一事件会清除 BB：<br>• I2C 模块接收或发送 STOP 位（总线空闲）。<br>• I2C 模块复位。<br>1h (R/W)：总线忙。I2C 模块已在总线上接收或发送 START 位。<br>复位类型：SYSRSn。 |
| 11 | RSFULL | R | 0h | 接收移位寄存器满位，指示接收期间的溢出状态。当新数据已进入接收移位寄存器 I2CRSR，而接收寄存器 I2CDRR 中的旧数据尚未读取时，发生溢出。SDA 引脚上到来的新位将覆盖 I2CRSR 中的位。读取前一份数据之前，新数据不会复制到 ICDRR。<br>0h (R/W)：未检测到溢出。以下任一事件会清除 RSFULL：<br>• CPU 读取 I2CDRR；仿真器读取不影响此位。<br>• I2C 模块复位。<br>1h (R/W)：检测到溢出。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 10 | XSMT | R | 1h | 发送移位寄存器空位。XSMT = 0 表示发送器发生欠载。I2CXSR 为空，且自上次 I2CDXR 向 I2CXSR 传送后尚未向 I2CDXR 装载新数据时，发生欠载。只有 I2CDXR 中有新数据后，才会进行下一次 I2CDXR 到 I2CXSR 的传送。若新数据未及时送入，SDA 引脚可能再次发送前一次数据。<br>0h (R/W)：检测到欠载（空）。<br>1h (R/W)：未检测到欠载（非空）。以下任一事件会置位 XSMT：<br>• 向 I2CDXR 写入数据。<br>• I2C 模块复位。<br>复位类型：SYSRSn。 |
| 9 | AAS | R | 0h | 被寻址为从机标志位。<br>0h (R/W)：7 位寻址模式下，接收到 NACK、STOP 条件或重复 START 条件时，清除 AAS。10 位寻址模式下，接收到 NACK、STOP 条件或不同于 I2C 外设自身从机地址的从机地址时，清除 AAS。<br>1h (R/W)：I2C 模块已识别到自身从机地址或全零地址（通用呼叫）。<br>复位类型：SYSRSn。 |
| 8 | AD0 | R | 0h | 地址 0 标志位。<br>0h (R/W)：AD0 已被 START 或 STOP 条件清除。<br>1h (R/W)：检测到全零地址（通用呼叫）。<br>复位类型：SYSRSn。 |
| 7 | RESERVED | R | 0h | 保留。 |
| 6 | BYTESENT | R/W1C | 0h | 字节发送完成指示。主机 / 从机成功通过 SCL / SDA 线路发送一个字节后，置位 BYTESENT。此诊断状态必须由软件显式清除。为避免边界条件问题，硬件没有自动清除机制；若不清除，将持续反映旧状态。<br>0h (R/W)：I2C 模块尚未完成下一数据字节的发送。以下任一事件会清除 BYTESENT：<br>• 手动清除，向此位写入 1。<br>• I2C 模块复位。<br>1h (R/W)：I2C 模块已完成一个字节的发送。<br>复位类型：SYSRSn。 |
| 5 | SCD | R/W1C | 0h | STOP 条件检测位。I2C 发送或接收到 STOP 条件时，置位 SCD。I2C 模块会等到 SCD 置位后才清除 I2CMDR[STP]。<br>0h (R/W)：自上次清除 SCD 以来未检测到 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | STOP 条件。以下任一事件会清除 SCD：<br>• CPU 读取值为 110b（检测到 STOP 条件）的 I2CISRC；仿真器读取不影响此位。<br>• 手动清除，向此位写入 1。<br>• I2C 模块复位。<br>1h (R/W)：I2C 总线上已检测到 STOP 条件。<br>复位类型：SYSRSn。 |
| 4 | XRDY | R | 1h | 发送数据就绪中断标志位。非 FIFO 模式下，XRDY 表示 I2CDXR 已准备好接收新数据。<br>FCM = 0：前一份数据从 I2CDXR 复制到 I2CXSR 后，CPU 可轮询 XRDY 或使用 XRDY 中断请求。FIFO 模式应改用 TXFFINT。<br>FCM = 1：仅在需要下一份数据时置位 XRDY，写入 I2CDXR 后撤销。FCM 模式支持轮询和中断两种数据传输方式。<br>0h (R/W)：I2CDXR 未就绪。向 I2CDXR 写入数据会清除 XRDY。<br>1h (R/W)：I2CDXR 已就绪，数据已从 I2CDXR 复制到 I2CXSR。I2C 模块复位时也会将 XRDY 强制置 1。<br>复位类型：SYSRSn。 |
| 3 | RRDY | R/W1C | 0h | 接收数据就绪中断标志位。非 FIFO 模式下，数据从 I2CRSR 复制到 I2CDRR 后，RRDY 表示 I2CDRR 可供读取。CPU 可轮询 RRDY 或使用 RRDY 中断请求。FIFO 模式应改用 RXFFINT。<br>0h (R/W)：I2CDRR 未就绪。以下任一事件会清除 RRDY：<br>• CPU 读取 I2CDRR；仿真器读取不影响此位。<br>• 手动清除，向此位写入 1。<br>• I2C 模块复位。<br>1h (R/W)：I2CDRR 已就绪，数据已从 I2CRSR 复制到 I2CDRR。<br>复位类型：SYSRSn。 |
| 2 | ARDY | R/W1C | 0h | 寄存器访问就绪中断标志位，仅适用于主机模式。之前配置的地址、数据和命令值已被使用后，ARDY 表示 I2C 模块寄存器可供访问。CPU 可轮询 ARDY 或使用 ARDY 中断请求。<br>0h (R/W)：寄存器尚不可访问。以下任一事件会清除 ARDY：<br>• I2C 模块开始使用当前寄存器内容。<br>• 手动清除，向此位写入 1。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | • I2C 模块复位。<br>1h (R/W)：寄存器可供访问。非重复模式（I2CMDR 中 RM = 0）下，若 STP = 0，则内部数据计数器减至 0 时置位 ARDY；若 STP = 1，则 ARDY 不受影响，模块改为在计数器减至 0 时产生 STOP 条件。重复模式（RM = 1）下，每发送完 I2CDXR 中的一个字节便置位 ARDY。<br>复位类型：SYSRSn。 |
| 1 | NACK | R/W1C | 0h | 非应答中断标志位，用于 I2C 模块作为主机发送器时，指示从从机接收器收到的是 ACK 还是 NACK。CPU 可轮询 NACK 或使用 NACK 中断请求。<br>0h (R/W)：收到 ACK / 未收到 NACK。以下任一事件会清除此位：<br>• 从机接收器已发送 ACK。<br>• 手动清除，向此位写入 1。<br>• CPU 读取 I2CISRC，且其中包含 NACK 中断代码；仿真器读取不影响此位。<br>• I2C 模块复位。<br>1h (R/W)：收到 NACK，硬件检测到一个非应答位。<br>注意：I2C 模块执行通用呼叫传输时，即使一个或多个从机发送应答，NACK 仍为 1。<br>复位类型：SYSRSn。 |
| 0 | ARBL | R/W1C | 0h | 仲裁丢失中断标志位，仅适用于主机发送器。主要指示 I2C 模块与其他主机 / 发送器竞争时是否丢失仲裁。CPU 可轮询 ARBL 或使用 ARBL 中断请求。<br>0h (R/W)：未丢失仲裁。以下任一事件会清除 AL：<br>• 手动清除，向此位写入 1。<br>• CPU 读取 I2CISRC，且其中包含 AL 中断代码；仿真器读取不影响此位。<br>• I2C 模块复位。<br>1h (R/W)：丢失仲裁。以下任一事件会置位 AL：<br>• 两个或更多发送器几乎同时开始传输并竞争时，I2C 模块检测到自己丢失仲裁。<br>• BB（总线忙）为 1 时，I2C 模块尝试开始传输。<br>AL 变为 1 后，I2CMDR 的 MST 和 STP 位被清除，模块转为从机接收器。<br>复位类型：SYSRSn。 |

### 21.6.2.4 I2CCLKL 寄存器

偏移地址：6h　复位值：0000h

I2C 时钟低电平时间分频。

表 21-15 I2CCLKL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| I2CCLKL | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2CCLKL | | | | | | | |
| R/W-0h | | | | | | | |

表 21-16 I2CCLKL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | I2CCLKL | R/W | 0h | 时钟低电平时间分频值。模块时钟周期乘以 (ICCL + d)，得到主机时钟的低电平持续时间。d 是由预分频器决定的调整因子。<br>注意：为正确生成 I2C 时钟，这些位必须设置为非零值。<br>复位类型：SYSRSn。 |

### 21.6.2.5 I2CCLKH 寄存器

偏移地址：8h　复位值：0000h

I2C 时钟高电平时间分频。

表 21-17 I2CCLKH 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| I2CCLKH | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2CCLKH | | | | | | | |
| R/W-0h | | | | | | | |

表 21-18 I2CCLKH 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | I2CCLKH | R/W | 0h | 时钟高电平时间分频值。模块时钟周期乘以 (ICCL + d)，得到主机时钟的高电平持续时间。d 是由预分频器决定的调整因子。<br>注意：为正确生成 I2C 时钟，这些位必须设置 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 为非零值。<br>复位类型：SYSRSn。 |

### 21.6.2.6 I2CCNT 寄存器

偏移地址：ah　复位值：0000h

I2C 数据计数。

表 21-19 I2CCNT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| I2CCNT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I2CCNT | | | | | | | |
| R/W-0h | | | | | | | |

表 21-20 I2CCNT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | I2CCNT | R/W | 0h | 数据计数值，表示要发送或接收的数据字节数。若指定 STOP 条件（STP = 1），每发送一个字节后 I2CCNT 递减，直到减至 0，并由此产生 STOP 条件。I2CMDR 的 RM = 1 时，I2CCNT 的值无关紧要。<br>0h (R/W)：数据计数值为 65536。<br>1h (R/W)：数据计数值为 1。<br>2h (R/W)：数据计数值为 2。<br>FFFFh (R/W)：数据计数值为 65535。<br>复位类型：SYSRSn。 |

### 21.6.2.7 I2CDRR 寄存器

偏移地址：ch　复位值：0000h

I2C 数据接收。

表 21-21 I2CDRR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R-0h | | | | | | | |

表 21-22 I2CDRR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | DATA | R | 0h | 接收数据。<br>复位类型：SYSRSn。 |

### 21.6.2.8 I2CSAR 寄存器

偏移地址：eh　复位值：03FFh

I2C 从机地址。

表 21-23 I2CSAR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | SAR | |
| R-0h | | | | | | R/W-3FFh | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SAR | | | | | | | |
| R/W-3FFh | | | | | | | |

表 21-24 I2CSAR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-10 | RESERVED | R | 0h | 保留。 |
| 9-0 | SAR | R/W | 3FFh | 7 位寻址模式（I2CMDR 中 XA = 0）：<br>00h—7Fh：位 6—0 提供 I2C 模块在主机发送模式下发送的 7 位从机地址。位 9—7 应写入 0。<br>10 位寻址模式（I2CMDR 中 XA = 1）：<br>000h—3FFh：位 9—0 提供 I2C 模块在主机发送模式下发送的 10 位从机地址。<br>复位类型：SYSRSn。 |

### 21.6.2.9 I2CDXR 寄存器

偏移地址：10h　复位值：0000h

I2C 数据发送。

表 21-25 I2CDXR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| R/W-0h | | | | | | | |

表 21-26 I2CDXR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | DATA | R/W | 0h | 发送数据。<br>复位类型：SYSRSn。 |

## 21.6.2.10 I2CMDR 寄存器

偏移地址：12h　复位值：0000h

I2C 模式。

表 21-27 I2CMDR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| NACKMOD | FREE | STT | 保留 | STP | MST | TRX | XA |
| R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RM | DLB | IRS | STB | FDF | BC | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | | |

表 21-28 I2CMDR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | NACKMOD | R/W | 0h | NACK 模式位，仅适用于接收器模式。<br>0h (R/W)：从机接收模式下，I2C 模块在每个应答周期向发送器发送 ACK，仅在设置 NACKMOD 后发送 NACK。主机接收模式下，在内部数据计数器减至 0 之前，每个应答周期均发送 ACK；减至 0 时向发送器发送 NACK。若需提前发送 NACK，必须设置 NACKMOD。<br>1h (R/W)：无论从机接收还是主机接收模式，模块均在总线的下一个应答周期向发送器发送 NACK。发送完成后，NACKMOD 自动清除。<br>重要：要在下一应答周期发送 NACK，必须在最后一个数据位的上升沿之前设置 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | NACKMOD。<br>复位类型：SYSRSn。 |
| 14 | FREE | R/W | 0h | 控制遇到调试器断点时 I2C 模块的行为。<br>0h (R/W)：作为主机时，若断点发生时 SCL 为低，则无论模块是发送器还是接收器，均立即停止并持续拉低 SCL；若 SCL 为高，则等待其变低后停止。作为从机时，断点会使模块在当前发送 / 接收完成后停止。<br>1h (R/W)：自由运行，发生断点时仍继续工作。<br>复位类型：SYSRSn。 |
| 13 | STT | R/W | 0h | START 条件位，仅适用于主机模式。RM、STT 和 STP 位决定何时开始和停止数据传输（参见表中）。STT 和 STP 可用于终止重复模式。IRS = 0 时，此位不可写。0h (R/W)：主机模式下，产生 START 条件后自动清除 STT。1h (R/W)：主机模式下，将 STT 置 1 会使模块在 I2C 总线上产生 START 条件。复位类型：SYSRSn。 |
| 12 | RESERVED | R | 0h | 保留。 |
| 11 | STP | R/W | 0h | STOP 条件位，仅适用于主机模式。RM、STT 和 STP 位决定何时开始和停止数据传输。STT 和 STP 可用于终止重复模式。IRS = 0 时，此位不可写。非重复模式下，至少传输一个字节后才能产生 STOP 条件。模块会等到 I2CSTR[SCD] 置位后才清除此位。为避免扰乱 I2C 状态机，发起新消息前必须等待此位清零。<br>0h (R/W)：产生 STOP 条件后自动清除 STP。<br>1h (R/W)：器件已设置 STP，使模块在内部数据计数器减至 0 时产生 STOP 条件。<br>复位类型：SYSRSn。 |
| 10 | MST | R/W | 0h | 主机模式位，决定模块处于从机还是主机模式。I2C 主机产生 STOP 条件时，MST 自动从 1 变为 0。<br>0h (R/W)：从机模式，从主机接收串行时钟。<br>1h (R/W)：主机模式，在 SCL 引脚产生串行时钟。<br>复位类型：SYSRSn。 |
| 9 | TRX | R/W | 0h | 发送器模式位，在适用时选择发送或接收模式。<br>0h (R/W)：接收器模式，通过 SDA 引脚接收数据。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1h (R/W)：发送器模式，通过 SDA 引脚发送数据。<br>复位类型：SYSRSn。 |
| 8 | XA | R/W | 0h | 扩展地址使能位。<br>0h (R/W)：7 位寻址（普通地址模式）。发送的从机地址来自 I2CSAR 的位 6—0，自身从机地址来自 I2COAR 的位 6—0，均为 7 位。<br>1h (R/W)：10 位寻址（扩展地址模式）。发送的从机地址来自 I2CSAR 的位 9—0，自身从机地址来自 I2COAR 的位 9—0，均为 10 位。<br>复位类型：SYSRSn。 |
| 7 | RM | R/W | 0h | 重复模式位，仅适用于主机发送器或主机接收器。RM、STT 和 STP 决定何时开始和停止数据传输。<br>0h (R/W)：非重复模式。I2CCNT 的值决定模块接收 / 发送的字节数。<br>1h (R/W)：重复模式。每次写入 I2CDXR 都发送一个字节；FIFO 模式下持续发送直到发送 FIFO 为空，直至手动设置 STP。忽略 I2CCNT 的值。可通过 ARDY 位 / 中断判断 I2CDXR（或 FIFO）何时可接收更多数据，或何时全部数据已发送、CPU 可写入 STP 位。<br>复位类型：SYSRSn。 |
| 6 | DLB | R/W | 0h | 数字回环模式位。<br>0h (R/W)：禁用数字回环。<br>1h (R/W)：使能数字回环。为正确运行，MST 必须为 1。I2CDXR 发出的数据通过内部路径，经过 n 个器件周期后进入 I2CDRR，其中：<br>n =（I2C 输入时钟频率 / 模块时钟频率）× 8。<br>发送时钟同时用作接收时钟。SDA 引脚发送的地址为 I2COAR 中的地址。<br>注意：数字回环不支持自由数据格式（FDF = 1）。<br>复位类型：SYSRSn。 |
| 5 | IRS | R/W | 0h | I2C 模块复位位。<br>0h (R/W)：模块处于复位 / 禁用状态。此位清零时，I2CSTR 中所有状态位均设为默认值。<br>1h (R/W)：使能模块。如果 I2C 外设占用了总线，此操作会释放总线。<br>复位类型：SYSRSn。 |
| 4 | STB | R/W | 0h | START 字节模式位，仅适用于主机。根据 Philips Semiconductors I2C 总线规范 2.1 版，START 字节可帮助需要额外时间检测 START |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 条件的从机。模块作为从机时，无论 STB 为何值，均忽略主机发送的 START 字节。<br>0h (R/W)：不使用 START 字节模式。<br>1h (R/W)：使用 START 字节模式。设置 STT 后，模块依次产生：<br>1. START 条件；<br>2. START 字节（0000 0001b）；<br>3. 一个虚拟应答时钟脉冲；<br>4. 重复 START 条件。<br>随后按正常流程发送 I2CSAR 中的从机地址。<br>复位类型：SYSRSn。 |
| 3 | FDF | R/W | 0h | 自由数据格式模式位。<br>0h (R/W)：禁用自由数据格式，使用 XA 选择的 7 / 10 位寻址格式。<br>1h (R/W)：使能自由数据格式，使用第 9.2.5 节所述的自由数据（无地址）格式。数字回环模式（DLB = 1）不支持该格式。<br>复位类型：SYSRSn。 |
| 2-0 | BC | R/W | 0h | 位计数字段，定义模块下一次接收或发送的数据字节包含的位数（1—8）。BC 选择的位数必须与另一器件的数据长度一致。BC = 000b 时，一个数据字节包含 8 位。BC 不影响地址字节，地址字节始终为 8 位。<br>注意：位数小于 8 时，接收数据在 I2CDRR(7-0) 中右对齐，其余位未定义。写入 I2CDXR 的发送数据也必须右对齐。<br>0h (R/W)：每个数据字节 8 位。<br>1h (R/W)：每个数据字节 1 位。<br>2h (R/W)：每个数据字节 2 位。<br>3h (R/W)：每个数据字节 3 位。<br>4h (R/W)：每个数据字节 4 位。<br>5h (R/W)：每个数据字节 5 位。<br>6h (R/W)：每个数据字节 6 位。<br>7h (R/W)：每个数据字节 7 位。<br>复位类型：SYSRSn。 |

### 21.6.2.11 I2CISRC 寄存器

偏移地址：14h　复位值：0000h

I2C 中断源。

表 21-29 I2CISRC 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | WRITE_ZEROS | | | |
| R-0h | | | | R/W-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | INTCODE | | |
| R-0h | | | | | R-0h | | |

表 21-30 I2CISRC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11-8 | WRITE_ZEROS | R/W | 0h | 内部测试位。这些保留位应始终写入 0。<br>复位类型：SYSRSn。 |
| 7-3 | RESERVED | R | 0h | 保留。 |
| 2-0 | INTCODE | R | 0h | 中断代码字段。INTCODE 的二进制代码表示引发 I2C 中断的事件。CPU 读取会清除此字段；若另一个较低优先级中断已使能且待处理，则装载该中断对应的值，否则保持清零。以下事件按优先级从高到低列出：代码 1（仲裁丢失）最高，代码 7（被寻址为从机）最低。<br>仲裁丢失、检测到 NACK 或检测到 STOP 条件时，CPU 读取还会清除 I2CSTR 中对应的中断标志。仿真器读取不影响本字段或 I2CSTR 的状态位。<br>0h (R/W)：无。<br>1h (R/W)：仲裁丢失。<br>2h (R/W)：检测到非应答条件。<br>3h (R/W)：寄存器可供访问。<br>4h (R/W)：接收数据就绪。<br>5h (R/W)：发送数据就绪。<br>6h (R/W)：检测到 STOP 条件。<br>7h (R/W)：被寻址为从机。<br>复位类型：SYSRSn。 |

## 21.6.2.12 I2CEMDR 寄存器

偏移地址：16h　复位值：0001h

I2C 扩展模式。

表 21-31 I2CEMDR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | FCM | BC |
| R-0h | | | | | | R/W-0h | R/W-1h |

表 21-32 I2CEMDR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-2 | RESERVED | R | 0h | 保留。 |
| 1 | FCM | R/W | 0h | 前向兼容模式。非 FIFO 模式下，设置此位可使模块仅在需要发送数据时才请求数据，而不取决于发送缓冲区的数据状态。此寄存器影响 XRDY 行为，因此必须在释放 IRS（I2CMDR[5]）后设置。<br>0h (R/W)：旧功能有效：缓冲区数据复制到移位寄存器时或 START 条件有效时请求发送数据。非法 START、仲裁丢失或 NACK 条件后会再次使用旧数据。<br>1h (R/W)：新功能有效：仅在收到地址 / 数据 ACK 后请求数据。<br>复位类型：SYSRSn。 |
| 0 | BC | R/W | 1h | 向后兼容模式。在从机发送模式下，此位影响 I2CSTR 中 XRDY 和 XSMT 的时序。<br>0h (R/W)：详情参见“向后兼容模式位——从机发送器”图。<br>1h (R/W)：详情参见“向后兼容模式位——从机发送器”图。<br>复位类型：SYSRSn。 |

## 21.6.2.13 I2CPSC 寄存器

偏移地址：18h　复位值：0000h

I2C 预分频。

表 21-33 I2CPSC 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IPSC | | | | | | | |
| R/W-0h | | | | | | | |

表 21-34 I2CPSC 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | IPSC | R/W | 0h | I2C 预分频值，决定对 CPU 时钟分频以生成 I2C 模块时钟的比例：<br>模块时钟频率 = I2C 输入时钟频率 / (IPSC + 1)。<br>注意：必须在 I2C 模块处于复位状态时（I2CMDR 中 IRS = 0）初始化 IPSC。<br>复位类型：SYSRSn。 |

## 21.6.2.14 I2CFFTX 寄存器

偏移地址：40h　复位值：0000h

I2C FIFO 发送。

表 21-35 I2CFFTX 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | I2CFFEN | TXFFRST | TXFFST | | | | |
| R-0h | R/W-0h | R/W-0h | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXFFINT | TXFFINTCLR | TXFFIENA | TXFFIL | | | | |
| R-0h | R-0/W1S-0h | R/W-0h | R/W-0h | | | | |

表 21-36 I2CFFTX 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RESERVED | R | 0h | 保留。 |
| 14 | I2CFFEN | R/W | 0h | I2C FIFO 模式使能位。发送或接收 FIFO 要正常工作，均必须使能此位。<br>0h (R/W)：禁用 I2C FIFO 模式。<br>1h (R/W)：使能 I2C FIFO 模式。<br>复位类型：SYSRSn。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 13 | TXFFRST | R/W | 0h | 发送 FIFO 复位。<br>0h (R/W)：将发送 FIFO 指针复位为 0000，并使发送 FIFO 保持复位状态。<br>1h (R/W)：使能发送 FIFO 运行。<br>复位类型：SYSRSn。 |
| 12-8 | TXFFST | R | 0h | 发送 FIFO 状态。<br>xxxxx：发送 FIFO 包含 xxxxx 个字节。<br>00000：发送 FIFO 为空。<br>注意：这些位复位为 0，因此使能发送 FIFO 并使 I2C 退出复位后，发送 FIFO 中断标志将被置位。若已使能中断，则产生发送 FIFO 中断。为避免由此带来的不利影响，应在使能发送 FIFO 且 I2C 退出复位后，向 TXFFINTCLR 写入 1。<br>复位类型：SYSRSn。 |
| 7 | TXFFINT | R | 0h | 发送 FIFO 中断标志。CPU 向 TXFFINTCLR 写入 1 可清除此位。若 TXFFIENA 已置位，则本位置位时产生中断。<br>0h (R/W)：未出现发送 FIFO 中断条件。<br>1h (R/W)：已出现发送 FIFO 中断条件。<br>复位类型：SYSRSn。 |
| 6 | TXFFINTCLR | R-0/W1S | 0h | 发送 FIFO 中断标志清除。<br>0h (R/W)：写入 0 无影响，读取返回 0。<br>1h (R/W)：写入 1 清除 TXFFINT 标志。<br>复位类型：SYSRSn。 |
| 5 | TXFFIENA | R/W | 0h | 发送 FIFO 中断使能。<br>0h (R/W)：禁用，TXFFINT 置位时不产生中断。<br>1h (R/W)：使能，TXFFINT 置位时产生中断。<br>复位类型：SYSRSn。 |
| 4-0 | TXFFIL | R/W | 0h | 发送 FIFO 中断阈值，设置置位发送中断标志的 FIFO 状态阈值。TXFFST4-0 的值小于或等于本字段时，置位 TXFFINT；若 TXFFIENA 已置位，则产生中断。本器件发送 FIFO 深度为 16，因此中断阈值不能配置为超过 16。<br>复位类型：SYSRSn。 |

## 21.6.2.15 I2CFFRX 寄存器

偏移地址：42h　复位值：0000h

I2C FIFO 接收。

表 21-37 I2CFFRX 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | RXFFRST | RXFFST | | | | |
| R-0h | | R/W-0h | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXFFINT | RXFFINTCLR | RXFFIENA | RXFFIL | | | | |
| R-0h | R-0/W1S-0h | R/W-0h | R/W-0h | | | | |

表 21-38 I2CFFRX 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | RESERVED | R | 0h | 保留。 |
| 13 | RXFFRST | R/W | 0h | I2C 接收 FIFO 复位位。<br>0h (R/W)：将接收 FIFO 指针复位为 0000，并使接收 FIFO 保持复位状态。<br>1h (R/W)：使能接收 FIFO 运行。<br>复位类型：SYSRSn。 |
| 12-8 | RXFFST | R | 0h | 接收 FIFO 状态。<br>xxxxx：接收 FIFO 包含 xxxxx 个字节。<br>00000：接收 FIFO 为空。<br>复位类型：SYSRSn。 |
| 7 | RXFFINT | R | 0h | 接收 FIFO 中断标志。CPU 向 RXFFINTCLR 写入 1 可清除此位。若 RXFFIENA 已置位，则本位置位时产生中断。<br>0h (R/W)：未出现接收 FIFO 中断条件。<br>1h (R/W)：已出现接收 FIFO 中断条件。<br>复位类型：SYSRSn。 |
| 6 | RXFFINTCLR | R-0/W1S | 0h | 接收 FIFO 中断标志清除位。<br>0h (R/W)：写入 0 无影响，读取返回 0。<br>1h (R/W)：写入 1 清除 RXFFINT 标志。<br>复位类型：SYSRSn。 |
| 5 | RXFFIENA | R/W | 0h | 接收 FIFO 中断使能位。<br>0h (R/W)：禁用，RXFFINT 置位时不产生中断。<br>1h (R/W)：使能，RXFFINT 置位时产生中断。<br>复位类型：SYSRSn。 |
| 4-0 | RXFFIL | R/W | 0h | 接收 FIFO 中断阈值，设置置位接收中断标志的 FIFO 状态阈值。RXFFST4-0 的值大于或等于本字段时，置位 RXFFINT；若 RXFFIENA 已置位，则产生中断。<br>注意：这些位复位为 0，因此使能接收 FIFO 且 I2C 退出复位时，接收 FIFO 中断标志将被置位；若已使能中断，则产生接收 FIFO 中断。为 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 避免这种情况，应在设置 RXFFRST 的同一条指令中或之前修改本字段。本器件接收 FIFO 深度为 16，因此中断阈值不能配置为超过 16。<br>复位类型：SYSRSn。 |

### 21.6.3 I2C 寄存器与 DriverLib 函数的对应关系

表 21-39 I2C 寄存器与 DriverLib 函数的对应关系

| 文件 | DriverLib 函数 |
|---|---|
| **OAR** | |
| i2c.h | I2C_setOwnAddress |
| **IER** | |
| i2c.c | I2C_enableInterrupt |
| i2c.c | I2C_disableInterrupt |
| **STR** | |
| i2c.c | I2C_getInterruptStatus |
| i2c.c | I2C_clearInterruptStatus |
| i2c.h | I2C_isBusBusy |
| i2c.h | I2C_getStatus |
| i2c.h | I2C_clearStatus |
| **CLKL** | |
| i2c.c | I2C_initController |
| i2c.c | I2C_initControllerModuleFrequency |
| **CLKH** | |
| i2c.c | I2C_initController |
| i2c.c | I2C_initControllerModuleFrequency |
| **CNT** | |
| i2c.h | I2C_setDataCount |
| **DRR** | |
| i2c.h | I2C_getData |
| **TAR** | |
| i2c.h | I2C_setTargetAddress |
| **DXR** | |
| i2c.h | I2C_putData |
| **MDR** | |
| i2c.h | I2C_enableModule |
| i2c.h | I2C_disableModule |

| 文件 | DriverLib 函数 |
| --- | --- |
| i2c.h | I2C_setConfig |
| i2c.h | I2C_setBitCount |
| i2c.h | I2C_sendStartCondition |
| i2c.h | I2C_sendStopCondition |
| i2c.h | I2C_sendNACK |
| i2c.h | I2C_getStopConditionStatus |
| i2c.h | I2C_setAddressMode |
| i2c.h | I2C_setEmulationMode |
| i2c.h | I2C_enableLoopback |
| i2c.h | I2C_disableLoopback |
| **ISRC** | |
| i2c.h | I2C_getInterruptSource |
| **EMDR** | |
| i2c.h | I2C_setExtendedMode |
| **PSC** | |
| i2c.c | I2C_initController |
| i2c.c | I2C_initControllerModuleFrequency |
| i2c.c | I2C_configureModuleFrequency |
| i2c.c | I2C_configureModuleClockFrequency |
| i2c.h | I2C_getPreScaler |
| **FFTX** | |
| i2c.c | I2C_enableInterrupt |
| i2c.c | I2C_disableInterrupt |
| i2c.c | I2C_getInterruptStatus |
| i2c.c | I2C_clearInterruptStatus |
| i2c.h | I2C_enableFIFO |
| i2c.h | I2C_disableFIFO |
| i2c.h | I2C_setFIFOInterruptLevel |
| i2c.h | I2C_getFIFOInterruptLevel |
| i2c.h | I2C_getTxFIFOStatus |
| **FFRX** | |
| i2c.c | I2C_enableInterrupt |
| i2c.c | I2C_disableInterrupt |
| i2c.c | I2C_getInterruptStatus |
| i2c.c | I2C_clearInterruptStatus |
| i2c.h | I2C_enableFIFO |

| 文件 | DriverLib 函数 |
| --- | --- |
| i2c.h | I2C_disableFIFO |
| i2c.h | I2C_setFIFOInterruptLevel |
| i2c.h | I2C_getFIFOInterruptLevel |
| i2c.h | I2C_getRxFIFOStatus |

# 22 本地互联网络（LIN）

## 22.1 简介

SCI/LIN 符合 LIN 规范包中规定的 LIN 2.1 协议。SCI/LIN 模块可通过编程设置为 SCI 模式或 LIN 模式。SCI 硬件的功能经过增强，以实现 LIN 兼容性。

SCI 模块是一个通用异步收发器（UART），采用标准的非归零格式。

LIN 标准基于 SCI（UART）串行数据链路格式。其通信理念为单主节点 / 多从节点模式，通过消息标识实现任意网络节点间的多播传输。

在本章中，兼容模式指的是 SCI/LIN 模块的 SCI 模式功能。尽管 LIN 和 SCI 共用寄存器，但寄存器描述中会通过注释说明不同模式下寄存器及位的使用方法。

## 22.2 主要特性

SCI 特性：

- 标准通用异步收发传输器（UART）通信
- 支持全双工或半双工操作
- 标准非归零（NRZ）格式
- 兼容模式下的双缓冲接收和发送功能
- 支持两条可单独启用的中断线：0 级和 1 级
- 可配置的帧格式，每个字符 3 至 13 位，具体基于以下内容：
  - 数据字长可编程为 1 至 8 位
  - 地址位模式下的附加地址位
  - 奇偶校验可编程为 0 或 1 个奇偶校验位，奇校验或偶校验
  - 停止位可编程为 1 或 2 个
- 异步通信模式
- 两种多处理器通信格式，允许两个以上设备之间进行通信

• 睡眠模式可用，以在多处理器通信期间释放 CPU 资源，然后唤醒以接收传入消息

• 24 位可编程波特率支持 224 种不同的波特率，提供高精度的波特率选择

• 在 150MHz 外设时钟下，可实现的最高波特率为 4.6875Mbits/s

• 能够使用直接存储器访问（DMA）进行数据的发送和接收

• 五个错误标志和七个状态标志提供有关 SCI 事件的详细信息

• 两个外部引脚：LINRX 和 LINTX

• 多缓冲接收和发送单元

注：SCI/LIN 模块在功能上与 SCI 模块兼容，但由于寄存器控制结构不同，并非直接软件兼容。

SCI/LIN 模块不支持 UART 硬件流控制。此功能可使用通用 I/O 引脚在软件中实现。

SCI/LIN 模块不支持等时模式，因为没有 SCICLK 引脚。

LIN 特性：

• 兼容 LIN 1.3、2.0 和 2.1 协议

• 可配置波特率高达 20 kbps

• 两个外部引脚：LINRX 和 LINTX。

• 多缓冲接收和发送单元

• 用于消息过滤的识别掩码

• 自动主节点报头生成

– 可编程同步间隔场

– 同步场

– 标识符场

• 从节点自动同步

– 同步间隔检测

– 可选波特率更新

– 同步验证

• 231 种可编程传输速率，带 7 个小数位

• 从收发器的 LINRX 显性电平唤醒

• 自动唤醒支持

– 唤醒信号生成

– 唤醒信号的过期时间

• 自动总线空闲检测

• 错误检测

– 位错误

– 总线错误

– 无响应错误

– 校验和错误

– 同步场错误

– 奇偶校验错误

• 能够使用 DMA 进行发送和接收数据。

• 2 条带优先级编码的中断线，用于：

– 接收

– 发送

– 标识符、错误和状态

• 支持 LIN 2.0 校验和

• 增强型同步器有限状态机支持帧处理

• 增强型扩展帧处理

• 增强型波特率发生器

• 更新唤醒/进入睡眠状态

## 22.3 功能描述

SCI/LIN 模块包含核心 SCI 块，并添加了支持 LIN 协议的子块。

SCI 模块的三个主要组件如下：

• 发送器（TX）包含两个主要寄存器，用于执行双缓冲：

– 发送器数据缓冲寄存器（SCITD）包含由 CPU 加载、将要传输到移位寄存器进行发送的数据。

– 发送器移位寄存器（SCITXSHF）从数据缓冲器（SCITD）加载数据，并将数据逐位移到 LINTX 引脚上。

• 波特率时钟发生器

– 可编程波特率发生器从输入时钟 VCLK 生成经过缩放的波特率时钟

– LIN VCLK 等于 SYSCLK 频率

• 接收器（RX）包含两个主要寄存器，用于执行双缓冲：

– 接收器移位寄存器（SCIRXSHF）从 LINRX 引脚逐位移入数据，并将完整数据传输到接收数据缓冲器中。

– 接收器数据缓冲寄存器（SCIRD）包含从接收器移位寄存器传输过来的接收数据

SCI 接收器和发送器均为双缓冲结构，且各有独立的使能位和中断位。接收器和发送器可以分别独立运行，也可以在全双工模式下同时运行。

为保证数据完整性，SCI 会对接收的数据进行检测，检查是否存在断路、奇偶校验、溢出和帧格式等错误。通过 24 位波特率选择寄存器，可将比特率（波特率）编程设置为超过 1600 万种不同的速率。图 22-1 展示了详细的 SCI 框图。

SCI/LIN 模块以独立 SCI 为基础，增加了错误检测器（奇偶校验计算器、校验和计算器和位监控器）、掩码滤波器、同步器以及多缓冲接收器和发送器。作为实现 LIN 兼容性的硬件增强部分，SCI 接口、DMA 控制子块和波特率发生器均经过了修改。图 22-2 展示了 SCI/LIN 框图。

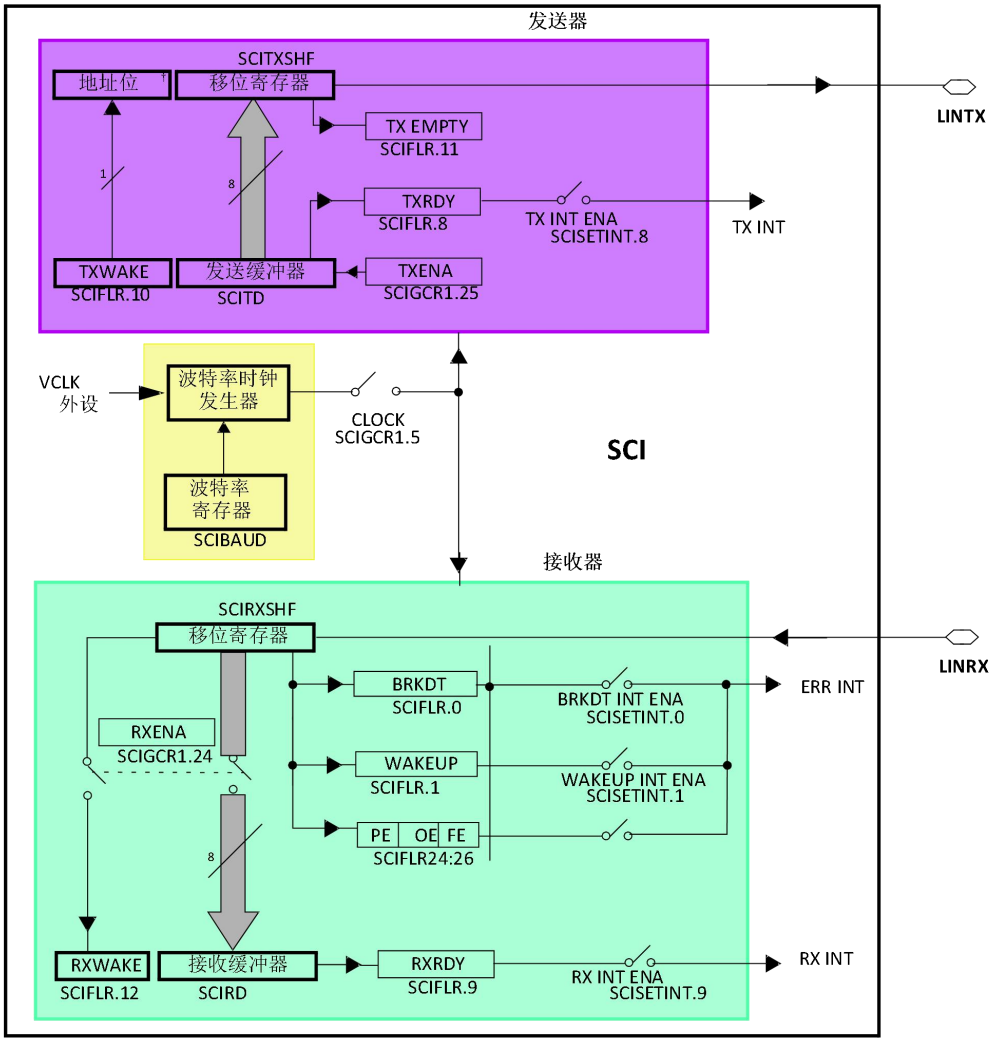

发送器
SCITXSHF
地址位
移位寄存器
LINTX
TX EMPTY
SCIFLR.11
1
8
TXRDY
SCIFLR.8
TX INT ENA
SCISETINT.8
TX INT
TXWAKE
SCIFLR.10
发送缓冲器
SCITD
TXENA
SCIGCR1.25
VCLK
外设
波特率时钟
发生器
CLOCK
SCIGCR1.5
SCI
波特率
寄存器
SCIBAUD
接收器
SCIRXSHF
移位寄存器
LINRX
BRKDT
SCIFLR.0
BRKDT INT ENA
SCISETINT.0
ERR INT
RXENA
SCIGCR1.24
WAKEUP
SCIFLR.1
WAKEUP INT ENA
SCISETINT.1
PE
OE
FE
SCIFLR24:26
8
RXWAKE
SCIFLR.12
接收缓冲器
SCIRD
RXRDY
SCIFLR.9
RX INT ENA
SCISETINT.9
RX INT

图 22-1 SCI 框图

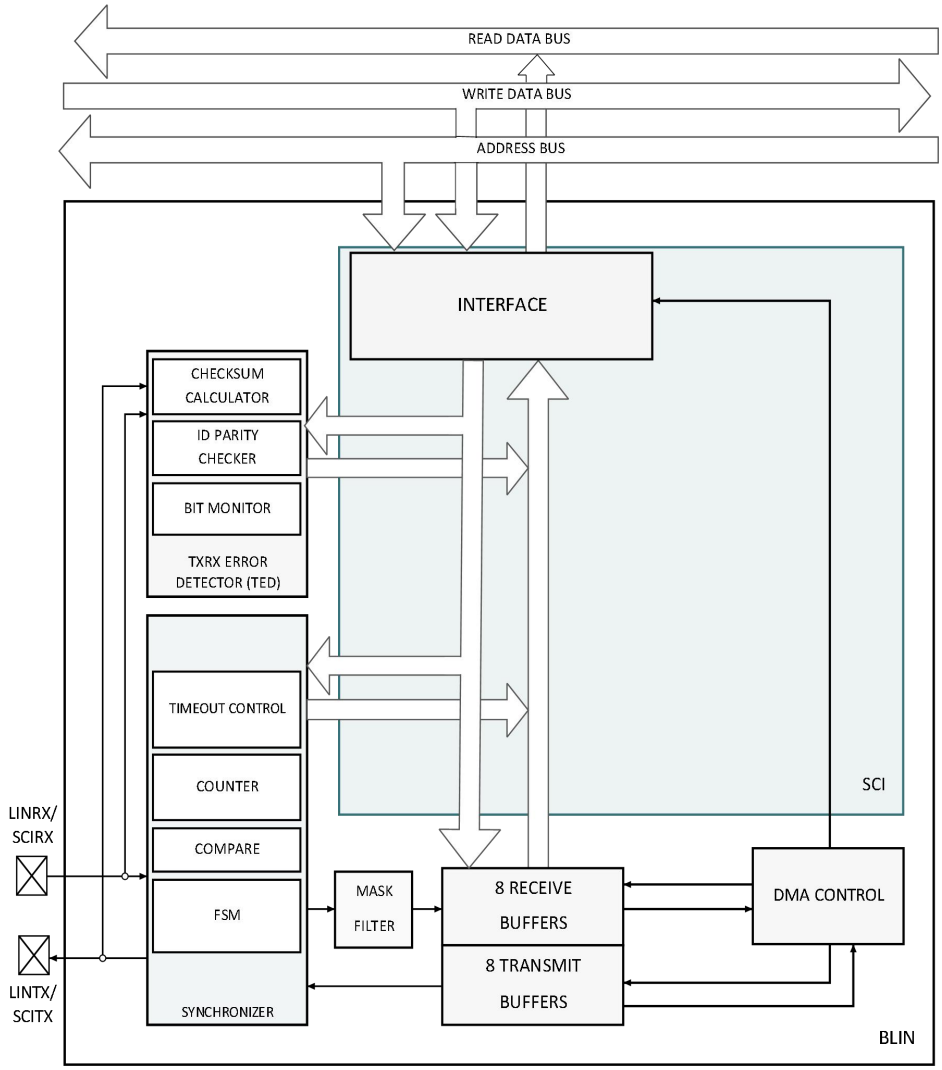



图 22-2 SCI/LIN 框图

## 22.3.1 串行通信接口模块

### 22.3.1.1 SCI 通信格式

SCI 模块可根据多种应用需求进行配置。由于不同应用的通信格式存在差异，SCI/LIN 的诸多属性支持用户自定义配置，具体配置选项如下：

- SCI 帧格式
- SCI 时序模式
- SCI 波特率
- SCI 多处理器模式

### 22.3.1.1.1 SCI 帧格式

SCI 采用可编程帧格式，所有帧均包含以下组成部分：

- 1 个起始位
- 1-8 个数据位
- 0 个或 1 个地址位
- 0 个或 1 个校验位
- 1 个或 2 个停止位

发送器与接收器的帧格式均可通过 SCIGCR1 寄存器中的相应位进行编程设置。发送和接收数据均采用不归零（NRZ）格式，即空闲状态下发送线和接收线均为逻辑高电平。每帧传输以起始位开始，此时发送器会将 SCI 线拉低（逻辑低电平）。起始位之后，帧数据按照最低有效位（LSB）优先的顺序进行发送和接收。

若 SCI 配置为地址位模式，每帧中会包含一个地址位；若配置为空闲线模式，则所有帧中均不包含地址位。图中展示了包含地址位和不包含地址位的帧格式。

当 PARITY ENA（校验使能）位被置 1 时，每帧中都会包含一个校验位。校验位的值取决于帧中 “1” 的个数，以及通过 PARITY ENA 位选择的是奇校验还是偶校验。图中的两个示例均启用了校验功能。

所有帧都包含一个停止位，且停止位始终为高电平。帧末尾的这个高电平用于标识一帧的结束，以确保通信设备之间的同步。若 SCIGCR1 寄存器中的 STOP 位被置 1，则会传输两个停止位。图中的示例每帧均使用 1 个停止位。

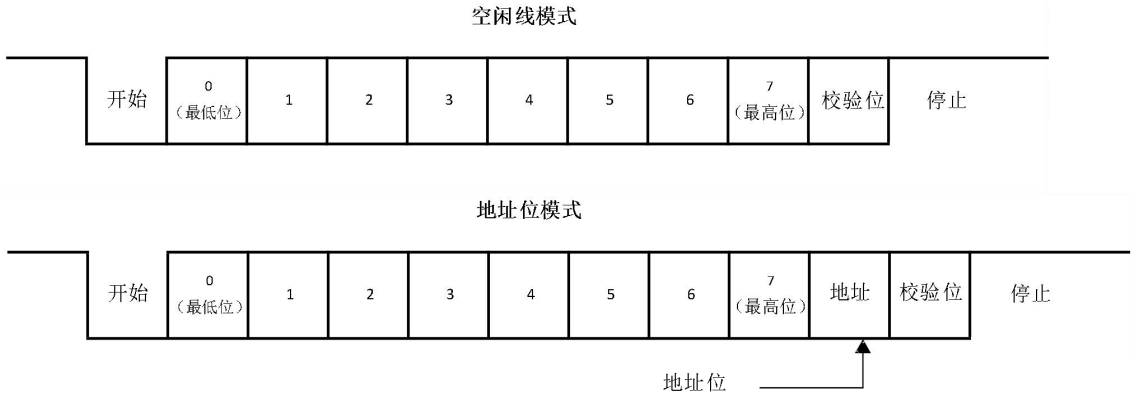



图 22-3 典型 SCI 数据帧格式

#### 22.3.1.1.2 SCI 异步时序模式

通过设置 SCIGCR1 寄存器中的 TIMING MODE（时序模式）位，可将 SCI 配置为使用异步时序模式。

在异步时序模式下，仅需通过接收数据线和发送数据线，即可与采用标准通用异步收发器（UART）协议的设备进行接口连接。

异步时序模式中，帧的每个位持续 16 个 SCI 波特时钟周期，因此每个位包含 16 个采样值（每个时钟周期对应一个采样值）。当 SCI 工作在异步模式时，所有通信设备的波特率必须尽可能匹配，否则会导致接收错误。

在异步时序模式下，接收器检测到 LINRX 引脚出现下降沿后，若后续前 4 个采样值均为逻辑电平 0，则判定为检测到有效起始位。一旦 LINRX 引脚上检测到下降沿，SCI 便认为有帧正在接收，并与总线实现同步。

为避免将噪声误判为起始位，SCI 要求 LINRX 线至少保持 4 个连续的 SCI 波特时钟周期的低电平，才会判定检测到有效起始位；若不满足该条件，则认为总线处于空闲状态。当检测到有效起始位后，SCI 会在第 7、8、9 个 SCI 波特时钟周期对 LINRX 线的电平值进行采样，并通过对这 3 个采样值进行多数表决，来确定存储到 SCI 接收器移位寄存器中的位值。通过在位的中间时刻进行采样，SCI 可减少由传播延迟、上升 / 下降时间以及数据线噪声所导致的错误。图中展示了在异步时序模式下，接收器对起始位和数据位的采样过程。

发送器发送每个位时，会持续 16 个 SCI 波特时钟周期。在某个位对应的第一个时钟周期内，发送器会将该位的值移位到 LINTX 引脚，随后在后续 16 个 SCI 波特时钟周期内，LINTX 引脚上将保持当前位的值不变。

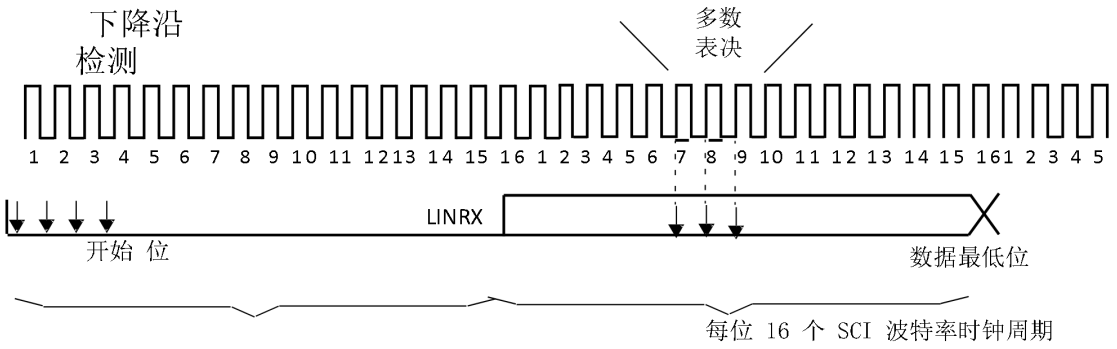



图 22-4 异步通信位时序

### 22.3.1.1.3 SCI 波特率

SCI/LIN 的内部串行时钟由外设 VCLK（外设时钟）以及寄存器中的预分频器 P 和 M 共同决定。SCI 通过 BRS（波特率选择）寄存器中 24 位整数预分频器 P 的值来选择所需的波特率，此外，4 位小数分频器 M 可进一步优化波特率的选择精度。

在异步时序模式下，SCI 按照以下公式生成波特时钟：

$$\text{SCICLK 频率} = \frac{\text{VCLK 频率}}{\text{P} + 1 + \frac{\text{M}}{16}} \quad (22\text{-}1)$$

$$\text{异步波特率} = \frac{\text{SCICLK 频率}}{16} \quad (22\text{-}2)$$

当 P=0 时：

$$\text{异步波特率} = \frac{\text{VCLK 频率}}{32} \quad (22\text{-}3)$$

#### 22.3.1.1.3.1 SCI 异步模式下的超小数分频器

超小数分频器适用于 SCI 异步模式（空闲线模式和地址位模式）。在 4 位小数分频器 M（BRS [27:24]）的基础上，超小数分频器额外使用一个 3 位调制值。表中值为 1 的位对应的（位时间）会额外增加一个 VCLK 周期。若字符长度超过 10，则调制表会采用原始表（表中）的循环版本。

根据 BRS [30:28] 的值，波特率会在数据字段范围内变化，其平均值为外设内部时钟的 “d” 分数倍：0<d<1。关于基于 “U” 值（BRS [30:28]）的简单 “d” 平均值计算。

瞬时位时间以 VCLK 周期（$T_{VCLK}$） 表示如下：

对于所有非 0 的 P 值，以及所有 M 值和 d 值（0 或 1）：

$$T^{i}bit = [16(P + 1 + \frac{M}{16}) + d] \times T_{VCLK} \quad (22\text{-}4)$$

当 P=0,$T_{bit}$=32×$T_{VCLK}$

平均位时间以 VCLK 周期（$T_{VCLK}$）表示如下：

对于所有非 0 的 P 值，以及所有 M 值和 d 值（0<d<1）：

$$T^{a}bit = [16(P + 1 + \frac{M}{16}) + d] \times T_{VCLK} \quad (22\text{-}5)$$

当 P=0,$T_{bit}$=32×$T_{VCLK}$

表 22-1 SCI 模式下的超小数位调制（标准配置）

| 标准配置 = 起始位 +8 个数据位 + 停止位 | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| BRS[30:28] | 起始位 | D[0] | D[1] | D[2] | D[3] | D[4] | D[5] | D[6] | D[7] | 停止位 |
| 0h | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1h | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2h | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 3h | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 4h | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 5h | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 6h | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 7h | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

表 22-2 SCI 模式下的超小数位调制（最大配置）

| 最大配置 = 起始位 +8 个数据位 + 地址位 + 校验位 + 停止位 0+ 停止位 1 | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BRS[30:28] | 起始位 | D[0] | D[1] | D[2] | D[3] | D[4] | D[5] | D[6] | D[7] | 地址位 | 校验位 | 停止位 0 | 停止位 1 |
| 0h | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1h | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2h | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 3h | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 4h | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 5h | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 6h | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 7h | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

表 22-3 SCI 模式（最小配置）

| 最小配置 = 起始位 +1 个数据位 + 停止位 | | | |
|---|---|---|---|
| BRS[30:28] | 起始位 | D[0] | 停止位 |
| 0h | 0 | 0 | 0 |
| 1h | 1 | 0 | 0 |
| 2h | 1 | 0 | 0 |
| 3h | 1 | 0 | 1 |
| 4h | 1 | 0 | 1 |
| 5h | 1 | 1 | 1 |
| 6h | 1 | 1 | 1 |
| 7h | 1 | 1 | 1 |

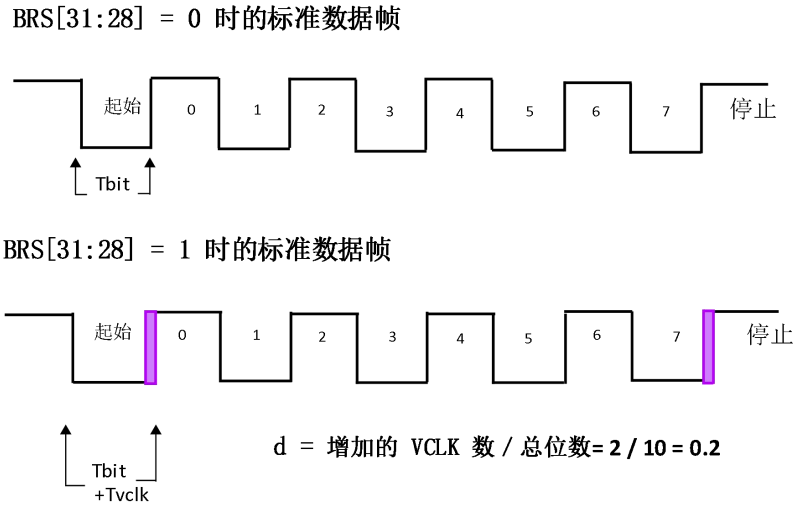



图 22-5 超小数分频器示例

### 22.3.1.1.4 多处理器通信模式

在部分应用场景中，SCI 可能需要连接多个串行通信设备。在这种多处理器配置下，可向所有连接的设备发送多帧数据，也可向单个设备发送数据。当数据仅发送给单个设备时，接收设备需判断自身是否为目标接收设备；若数据并非发送给自身，则可忽略后续数据。若 SCI 网络中仅包含两个设备，则无需进行地址寻址，因此也不需要采用多处理器通信方案。

SCI 支持两种多处理器通信模式，可通过 COMM MODE（通信模式）位进行选择：

- 空闲线模式
- 地址位模式

当 SCI 未用于多处理器环境时，软件可将所有帧均视为数据帧。在此情况下，空闲线模式与地址位模式的唯一区别在于，采用地址位协议发送的每帧中会额外包含一个地址位。

SCI 支持全双工通信，即通过发送引脚和接收引脚可同时进行数据的发送与接收。但 SCI 所采用的协议规定，同一时刻总线上仅允许一个设备进行数据发送，SCI 不具备总线仲裁功能。

#### 22.3.1.1.4.1 空闲线多处理器模式

在空闲线多处理器模式下，若一帧之前存在 10 个或更多的空闲位，则该帧为地址帧；若一帧之前的空闲位少于 10 个，则该帧为数据帧。图中展示了空闲线模式下多个数据块和帧的格式。

采用空闲线模式发送地址帧有以下两种方法：

方法 1：通过软件控制，在发送前一个数据块的最后一帧数据与新数据块的地址帧之间，刻意设置一段空闲周期。

方法 2：配置 SCI，使其在前一个数据块的最后一帧数据与新数据块的地址帧之间自动插入一段空闲周期。

方法 1 仅需通过软件中的延迟循环即可实现，而方法 2 需借助发送缓冲区和 TXWAKE 位，具体步骤如下：

1. 将 TXWAKE 位置 1。

2. 向 SCITD 写入一个虚拟数据值，这会触发 SCI 在发送器移位寄存器为空后立即开始插入空闲周期。

3. 等待 SCI 将 TXWAKE 标志位清零。

4. 向 SCITD 写入地址值。

如步骤 3 所述，软件可等待 SCI 将 TXWAKE 位清零。实际上，SCI 在置位 TXRDY（即把 SCITD 中的数据传输到 SCITXSHF）的同时，会将 TXWAKE 位清零。因此，若 TX INT ENA（发送中断使能）位已置 1，则当数据从 SCITD

传输到 SCITXSHF 时，会产生一个中断，同时 SCI 将 TXWAKE 位清零。若采用这种中断方式，软件无需通过轮询 TXWAKE 位来等待其清零。

在使用空闲线多处理器通信时，软件必须确保在发送地址帧之前（可采用上述任一方法），空闲时间超过 10 个位周期；同时，软件还需确保向发送器写入数据帧的速度足够快，以避免帧与帧之间出现 10 个位周期的延迟。若不满足这些条件，会导致其他接收设备对数据产生误解读。

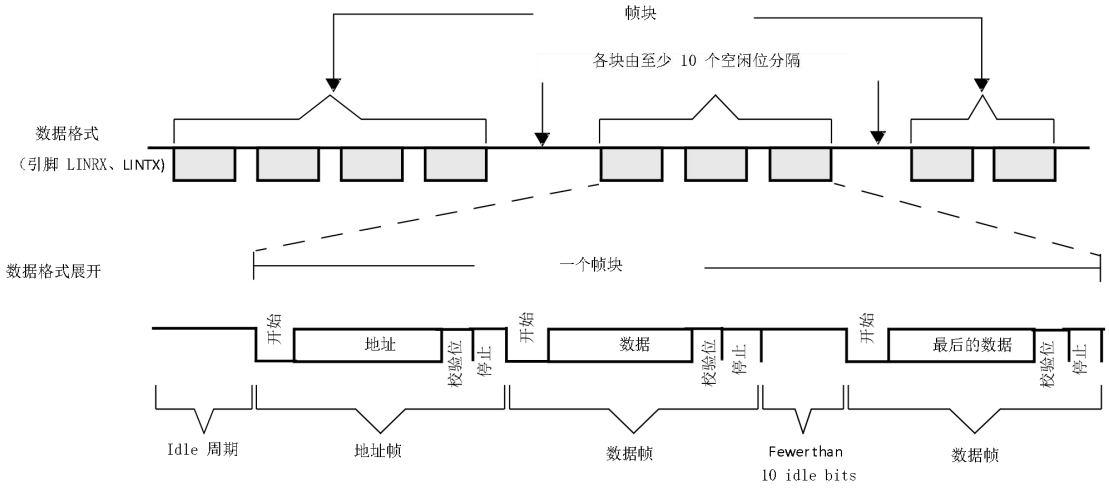



图 22-6 空闲线多处理器通信格式

### 22.3.1.1.4.2 地址位多处理器模式

在地址位协议中，每帧的数据字段之后会额外包含一个地址位。地址位为 1 时，该帧为地址帧；地址位为 0 时，该帧为数据帧。此模式下，空闲周期的时序无关紧要。图中展示了地址位模式下多个数据块和帧的格式。

当使用地址位模式时，TXWAKE 位的值会作为地址位进行发送。若要发送地址帧，软件必须将 TXWAKE 位置 1。当 SCITD 中的内容从 TXWAKE 寄存器移位输出时，TXWAKE 位会被清零，因此，除非 TXWAKE 位被写为 1，否则所有发送的帧均为数据帧。

在地址位模式下，发送地址帧之前无需向 SCITD 写入虚拟数据。当 TXWAKE 位被写为 1 后，向 SCITD 写入的第一个字节在发送时，其地址位会被置 1。

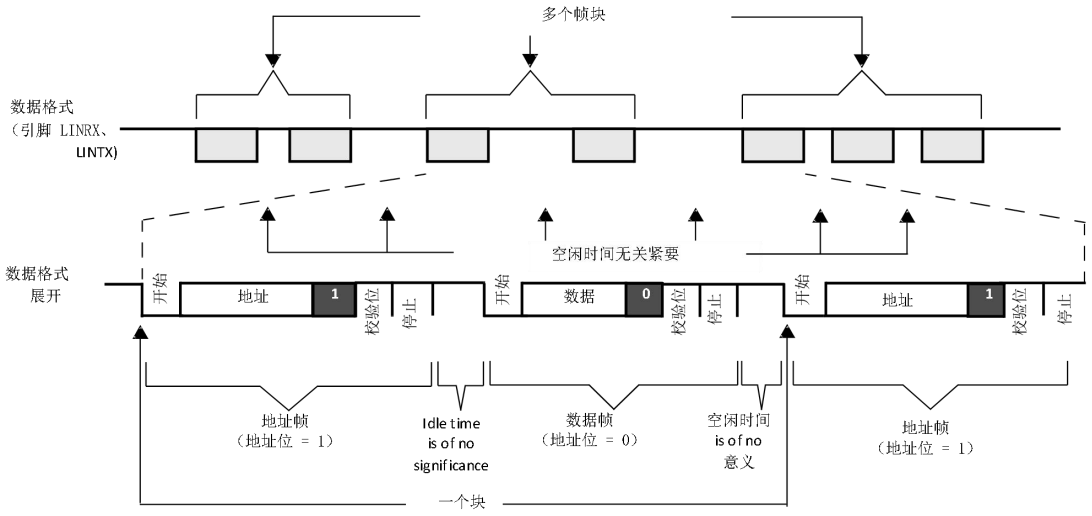



图 22-7 地址位多处理器通信格式

### 22.3.1.1.5 SCI 多缓冲模式

为降低中断模式或 DMA 模式下接收或发送数据时的 CPU 负载，SCI/LIN 模块配备了 8 个独立的接收缓冲区和 8 个独立的发送缓冲区。通过置位 MBUF MODE 位，可启用多缓冲模式。

多缓冲 3 位计数器会对从 SCIRXSHF 传输到 RDy 接收缓冲区，以及从 TDy 发送缓冲区传输到 SCITXSHF 的数据字节数进行计数。3 位比较寄存器中存储的是预期接收或发送的数据字节数，SCIFORMAT 寄存器中的 LENGTH 值表示预期长度，该值会被加载到 3 位比较寄存器中。

若接收的帧不存在响应接收错误（如帧错误、溢出错误等），则在接收完响应后，会产生接收中断（RX 中断，详见 SCIINTVECT0 和 SCIINTVECT1 寄存器）、置位 SCIFLR 寄存器中的接收就绪 RXRDY 标志，同时还可能产生 DMA 请求（RXDMA）。

发送完响应后，会产生发送中断（TX 中断）、置位 SCIFLR 寄存器中的发送就绪 TXRDY 标志，同时可能产生 DMA 请求（TXDMA）。

下图分别展示了接收多缓冲和发送多缓冲的功能框图。

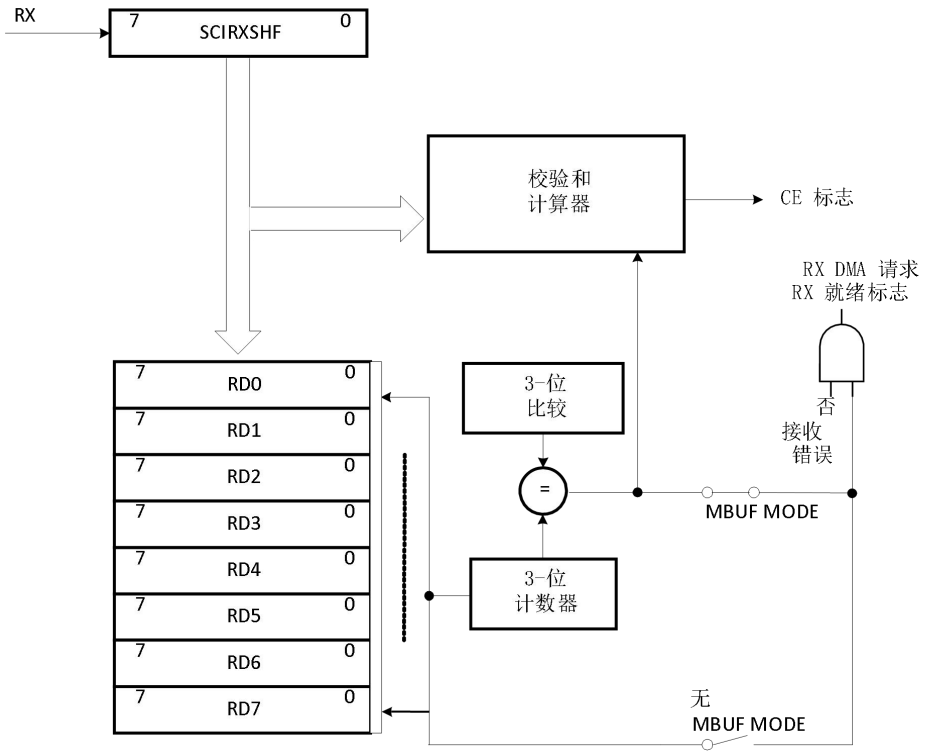



图 22-8 接收缓冲

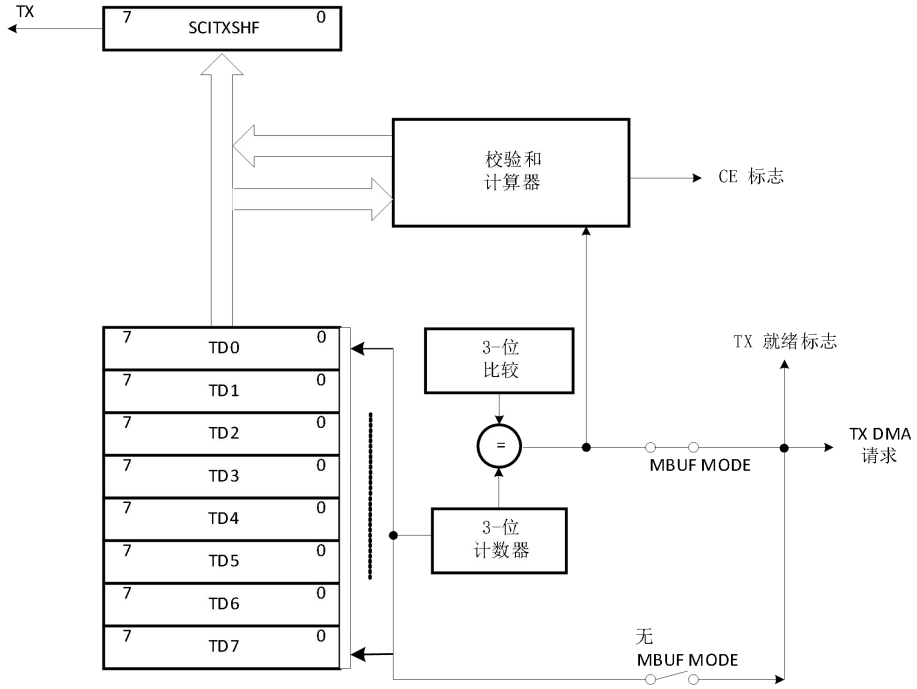



图 22-9 发送缓冲

### 22.3.1.2 SCI 中断

SCI/LIN 模块拥有两条中断线（电平 0 和电平 1），可连接到 CLIC 中断控制器（见图）。两个偏移寄存器 SCIINTVECT0 和 SCIINTVECT1 会根据各自的优先级编码器，确定是哪个标志触发了中断。每种中断条件在 SCISETINT 寄存器和 SCICLRINT 寄存器中均有对应的位，用于启用或禁用该中断。

每条中断还对应一个位，可将该中断设置为中断电平 0（INT0）或中断电平 1（INT1）。默认情况下，所有中断均处于中断电平 0。通过 SCISETINTLVL 寄存器可将指定中断设置为电平 1；通过 SCICLEARINTLVL 寄存器可将指定中断的电平重置为默认的电平 0。

中断向量寄存器 SCIINTVECT0 和 SCIINTVECT1 会返回挂起中断线 INT0 或 INT1 的向量。若存在多个挂起中断，中断向量寄存器会存储优先级最高的中断向量。

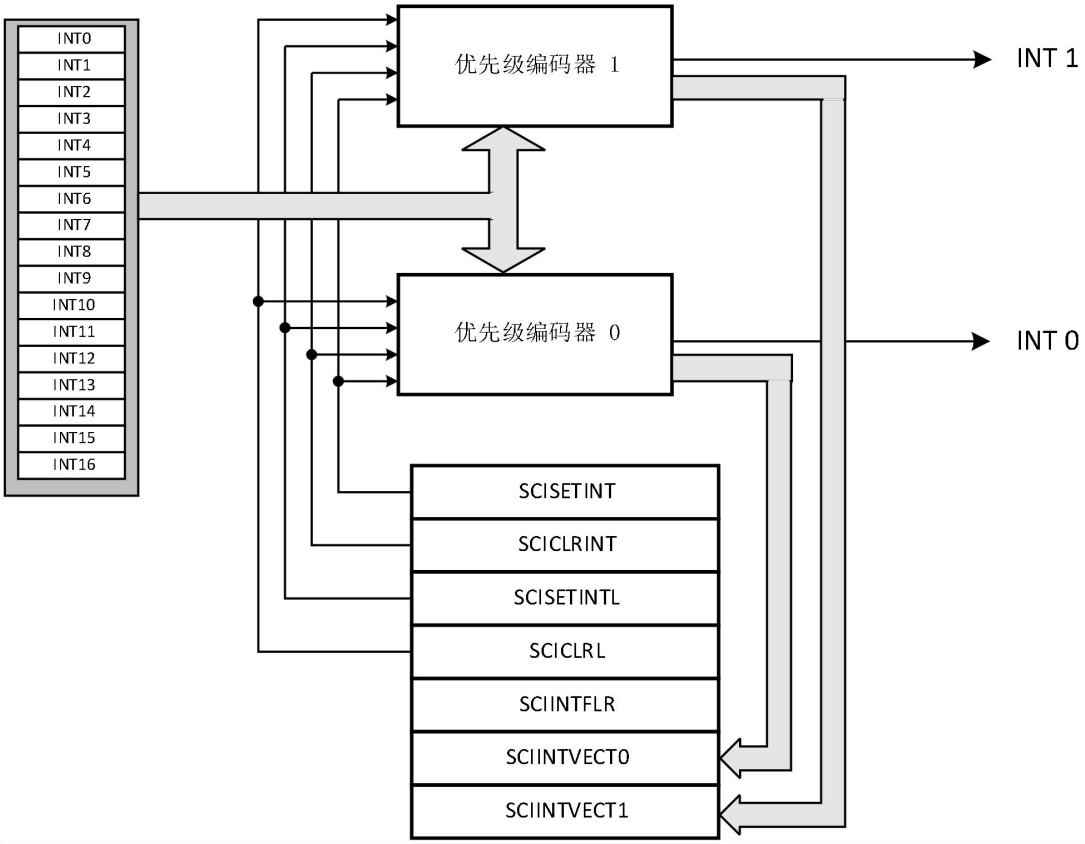



图 22-10 通用中断方案

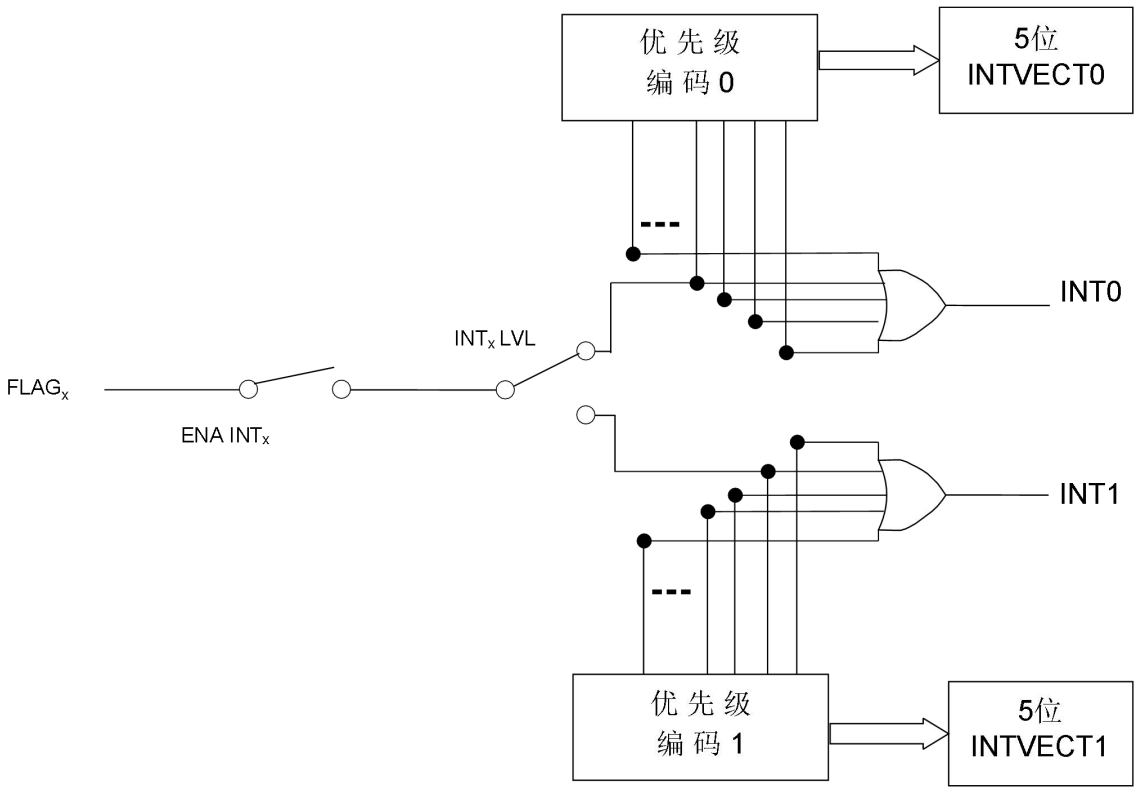



图 22-11 给定标志的中断产生

### 22.3.1.2.1 发送中断

若要使用发送中断功能，必须置位 SETTXINT 位，且清零 SCISETINT 寄存器中的 SET_TX_DMA 位。当 SCI 将 SCITD 中的内容传输到移位寄存器 SCITXSHF 时，会置位发送就绪（TXRDY）标志，该标志表明 SCITD 已准备好接收新的数据写入。此外，若 SCITD 和 SCITXSHF 寄存器均为空，SCI 会置位 TX EMPTY 位。若 SET TX INT 位已置 1，则当 TXRDY 标志变为高电平时，会产生发送中断。与发送 DMA 请求不同，置位 SET TX INT 位后不会立即产生发送中断，只有在数据首次从 SCITD 传输到 SCITXSHF（即需先向 SCITD 写入第一个数据）后，才会产生发送中断。若要继续发送数据，可在发送中断服务程序中向 SCITD 写入数据。

向 SCITD 寄存器写入数据会将 TXRDY 位清零。当该数据被传输到 SCITXSHF 寄存器后，TXRDY 位会再次置位。可通过置位 CLR TX INT 位来暂停中断请求；但当 SETTXINT 位再次置 1 时，TXRDY 中断会再次生效。通过

禁用发送器（清零 TXENA（发送使能）位）、软件复位（SWnRST）或设备硬件复位，可清除发送中断请求，直至向 SCITD 写入下一组数据。

#### 22.3.1.2.2 接收中断

当 SCI 将新接收的数据从 SCIRXSHF 传输到 SCIRD 时，会置位接收就绪（RXRDY）标志，该标志表明 SCI 存在新的待读取数据。通过置位 SETRXINT 位，可启用接收中断。若 SETRXINT 位已置位，且 SCI 置位了 RXRDY 标志，则会产生接收中断。可在中断服务程序中读取接收的数据。

对于同时配备 SCI 和 DMA 控制器的设备，需清零 SET_RX_DMA 位，以选择中断功能。

#### 22.3.1.2.3 唤醒中断

若 RX 线上的总线活动阻止设备进入掉电模式，或 RX 线上的活动导致设备从掉电模式退出，SCI 会置位 WAKEUP 标志。若 WAKEUP 中断已启用（SET WAKEUP INT 位已置位），则一旦 WAKEUP 标志置位，就会触发唤醒中断。

#### 22.3.1.2.4 错误中断

SCI 模块支持对以下错误检测提供中断功能：

- 校验错误（PE）
- 帧错误（FE）
- break 检测错误（BRKDT）
- 溢出错误（OE）
- 位错误（BE）

SCI/LIN 模块共有 16 个中断源，在 SCI 模式下支持 8 个中断，具体如表所示。

若上述所有错误（PE、FE、BRKDT、OE、BE）均被标记，且对应的中断已启用，则会为被标记的错误产生中断。只有当一帧数据传输或接收完成且未检测到

任何错误时，该消息对发送器和接收器而言才是有效的。上述每种错误对应的标志均位于接收器状态寄存器（SCIFLR）中。

表 22-4 SCI/LIN 中断

| 偏移量（1） | 中断类型 | 适用于 SCI 模式 | 适用于 LIN 模式 |
|---|---|---|---|
| 0 | 无中断 | - | - |
| 1 | 唤醒 | 是 | 是 |
| 2 | 同步字段不一致错误（ISFE） | 否 | 是 |
| 3 | 校验错误（PE） | 是 | 是 |
| 4 | 标识符（ID） | 否 | 是 |
| 5 | 物理总线错误（PBE） | 否 | 是 |
| 6 | 帧错误（FE） | 是 | 是 |
| 7 | Break 检测（BRKDT） | 是 | 否 |
| 8 | 校验和错误（CE） | 否 | 是 |
| 9 | 溢出错误（OE） | 是 | 是 |
| 10 | 位错误（BE） | 是 | 是 |
| 11 | 接收 | 是 | 是 |
| 12 | 发送 | 是 | 是 |
| 13 | 无响应错误（NRE） | 否 | 是 |
| 14 | 唤醒信号超时（150ms） | 否 | 是 |
| 15 | 三次唤醒信号超时（1.5s） | 否 | 是 |
| 16 | 超时（总线空闲，4s） | 否 | 是 |

（1）偏移量 1 对应的中断优先级最高，偏移量 16 对应的中断优先级最低。

表 22-5 SCI 接收器状态标志

| SCI 标志 | 寄存器 | 位 | 复位后的值（1） |
|---|---|---|---|
| CE | SCIFLR | 29 | 0 |
| ISFE | SCIFLR | 28 | 0 |
| NRE | SCIFLR | 27 | 0 |
| FE | SCIFLR | 26 | 0 |
| OE | SCIFLR | 25 | 0 |
| PE | SCIFLR | 24 | 0 |
| RXWAKE | SCIFLR | 12 | 0 |
| RXRDY | SCIFLR | 9 | 0 |
| BUSY | SCIFLR | 3 | 0 |
| IDLE | SCIFLR | 2 | 1 |
| WAKEUP | SCIFLR | 1 | 0 |
| BRKDT | SCIFLR | 0 | 0 |

（1）当时，标志会保持复位值不变。

表 22-6 SCI 发送器状态标志

| SCI 标志 | 寄存器 | 位 | 复位后的值（1） |
|---|---|---|---|
| BE | SCIFLR | 31 | 0 |
| PBE | SCIFLR | 30 | 0 |
| TXWAKE | SCIFLR | 10 | 0 |
| TXEMPTY | SCIFLR | 11 | 1 |
| TXRDY | SCIFLR | 8 | 1 |

（1）当时，标志会保持复位值不变。

### 22.3.1.3 SCI DMA 接口

SCI/LIN 模块支持接收（RXDMA 请求）和发送（TXDMA 请求）的 DMA 请求。当禁用多缓冲模式时（SCIGCR1 寄存器中的 mbufmode 被清零），DMA 必须配置为与 SCITD / SCIRD 寄存器进行数据传输；当启用多缓冲模式时（SCIGCR1 寄存器中的 mbufmode 被设置），则需与 TDy/RDy 寄存器进行数据传输。

#### 22.3.1.3.1 接收 DMA 请求

CPU 可通过 SETRXDMA（置位接收 DMA）位和 CLRRXDMA（清零接收 DMA）位，分别启用和禁用该 DMA 功能。

在启用 DMA 的 SCI 多缓冲模式下，接收器会将每个接收的字符加载到 RDy 缓冲区中。当接收并成功复制完编程设定数量（LENGTH）的最后一个字符到对应的 RDy 缓冲区后，会触发 RX DMA 请求。

若禁用多缓冲选项，则会以字节为单位产生 DMA 请求。

在多处理器模式下，SCI 可针对地址帧产生接收中断，针对数据帧产生 DMA 请求，也可针对地址帧和数据帧均产生 DMA 请求，具体由 SET_RX_DMA_ALL 位控制。

在多处理器模式下，若 SLEEP 位已置位，则不会针对接收的数据帧产生 DMA 请求。软件必须先清零 SLEEP 位，才能接收数据帧。

#### 22.3.1.3.2 发送 DMA 请求

CPU 可通过 SET_TX_DMA 位和 CLR_TX_DMA 位，分别启用和禁用该 DMA 功能。

在 SCI 多缓冲模式下，一旦 TXRDY 位置位，或发送完编程设定数量（LENGTH）的字符（最多 8 个数据字节，存储在 LINTD0 和 LINTD1 寄存器的发送缓冲区（TDy）中）后，会产生 DMA 请求，以便为下一次传输重新加载发送缓冲区。若禁用多缓冲选项，则会以字节为单位产生 DMA 请求。

#### 22.3.1.4 SCI 配置

在 SCI 发送或接收数据之前，需对 SCI 寄存器进行正确配置。设备上电或系统级复位后，SCI 寄存器的每个位都会被设置为默认状态。只有当 SCIGCR0 中的 RESET 位被置 1 后，才能对这些寄存器进行写操作。其中，SCIGCR1 中的 SWnRST 位尤为重要，该位为低电平有效，复位后初始值为 0，会使 SCI 保持在复位状态，直至该位被编程为 1。因此，所有 SCI 配置操作都可在将 SWnRST 位写为 1 之前完成。

以下列出了软件在传输或接收数据之前可执行的配置步骤。只要在配置 SCI 的整个过程中，SWnRST 位始终保持为 0，寄存器的编程顺序就无关紧要。

- 将 RESET 位置 1，启用 SCI。
- 在配置 SCI 之前，将 SWnRST 位清零。
- 通过编程 SCIGCR1 寄存器，选择所需的帧格式。
- 将 SCIPIO0 中的 RX FUNC 位和 TX FUNC 位均置 1，将 LINRX 和 LINTX 引脚配置为 SCI 功能引脚。
- 通过编程 BRS 寄存器，选择通信所需的波特率。
- 将 SCIGCR1 寄存器中的 CLOCK 位置 1，选择内部时钟。
- 将 SCIGCR1 寄存器中的 CONT 位置 1，使 SCI 在仿真断点时不会暂停，直至当前的接收或传输操作完成（该位仅在仿真环境中使用）。
- 将 SCIGCR1 寄存器中的 LOOP BACK 位置 1，将发送器与接收器在内部连接（该功能用于执行自测试）。
- 若需要接收数据，将 SCIGCR1 寄存器中的 RXENA 位置 1。

- 若需要发送数据，将 SCIGCR1 寄存器中的 TXENA 位置 1。
- SCI 配置完成后，将 SWnRST 位置 1。
- 执行数据接收或发送操作。

#### 22.3.1.4.1 接收数据

只有当 RX FUNC 位和 RXENA 位均置 1 时，SCI 接收器才能启用接收消息功能。若 RX FUNC 位未置 1，则 LINRX 引脚会作为通用 I/O 引脚使用，而非 SCI 功能引脚。

SCI 模块可通过以下任一模式接收数据：

- 单缓冲（标准）模式
- 多缓冲模式

检测到有效的空闲周期后，当数据通过 LINRX 引脚到达时，SCI 会自动接收数据。

##### 22.3.1.4.1.1 单缓冲模式接收数据

当 SCIGCR1 中的 MBUF MODE 位清零时，SCI 工作在单缓冲模式。在此模式下，当 SCI 将新接收的数据从 SCIRXSHF 传输到 SCIRD 时，会置位 RXRDY 位；当读取完 SCIRD 中的新数据后，SCI 会将 RXRDY 位清零。此外，当数据从 SCIRXSHF 传输到 SCIRD 时，若检测到接收数据存在帧错误（FE）、溢出错误（OE）或校验错误（PE）等错误情况，SCI 会置位相应的标志位。这些错误情况均支持可配置的中断功能。若发生唤醒或 break 检测错误，对应的状态位也会置位，但这些位的置位时机不一定与新数据加载到 SCIRD 的时机一致。

可通过以下方式接收数据：

1. 轮询接收就绪标志
2. 接收中断
3. DMA

采用轮询方式时，软件可轮询 RXRDY 位，一旦该位变为高电平，就从 SCIRD 寄存器中读取数据。但轮询方式会不必要地增加 CPU 负载，为避免这种情况，可采用中断或 DMA 方式。若采用中断方式，需置位 SET RX INT 位；若采用 DMA 方式，需置位 SET RX DMA 位。RXRDY 位一旦置位，就会产生中断或 DMA 请求。

#### 22.3.1.4.1.2 多缓冲模式接收数据

当 SCIGCR1 中的 MBUF MODE 位置 1 时，SCI 工作在多缓冲模式。在此模式下，当接收缓冲区中接收到编程设定数量的数据（即完整的一帧数据）后，SCI 会置位 RXRDY 位。多缓冲模式下的错误检测逻辑与单缓冲模式类似，不同之处在于，多缓冲模式下的错误检测逻辑会对完整的一帧数据进行监控。与单缓冲模式相同，可采用轮询、DMA 或中断方式读取数据。读取完 SCIRD 中的新数据后，SCI 会将 RXRDY 位清零。

### 22.3.1.4.2 发送数据

只有当 TX FUNC 位和 TXENA 位均置 1 时，SCI 发送器才能启用。若 TX FUNC 位未置 1，则 LINTX 引脚会作为通用 I/O 引脚使用，而非 SCI 功能引脚。在 TXENA 位被置 1 之前，写入 SCITD 的任何值都不会被传输。通过这两个控制位，可独立地使 SCI 发送器保持非活动状态，而不影响接收器。

SCI 模块可通过以下任一模式发送数据：

- 单缓冲（标准）模式
- 多缓冲模式

#### 22.3.1.4.2.1 单缓冲模式发送数据

当 SCIGCR1 中的 MBUF MODE 位清零时，SCI 工作在单缓冲模式。在此模式下，SCI 会等待数据写入 SCITD，然后将数据传输到 SCITXSHF 并进行发送。TXRDY 和 TX EMPTY 位用于指示发送缓冲区的状态：当发送器准备好接收写入

SCITD 的数据时，TXRDY 位置位；此外，若 SCITD 和 SCITXSHF 均为空，则 TX EMPTY 位也会置位。

可通过以下方式发送数据：

1. 轮询发送就绪标志

2. 发送中断

3. DMA

采用轮询方式时，软件需轮询 TXRDY 位，待该位变为高电平后，再将数据写入 SCITD 寄存器。但轮询方式会不必要地增加 CPU 负载，为避免这种情况，可采用中断或 DMA 方式。若采用中断方式，需置位 SET TX INT 位；若采用 DMA 方式，需置位 SET TX DMA 位。TXRDY 位一旦置位，就会产生中断或 DMA 请求。当 SCI 完成所有待发送帧的传输后，SCITXSHF 寄存器和 SCITD 均为空，此时 TXRDY 位置位，若相关功能已启用，会产生中断或 DMA 请求。由于所有数据均已传输完成，需终止中断或 DMA 请求，可通过禁用发送中断（清零 TX INT 位）、禁用 DMA 请求（清零 TX DMA 位）或禁用发送器（清零 TXENA 位）来实现。

注意：读取 SCIINTVECT0 或 SCIINTVECT1 寄存器中对应的中断偏移量，无法清除 TXRDY 标志。

#### 22.3.1.4.2.2 多缓冲模式发送数据

当 SCIGCR1 中的 MBUF MODE 位置 1 时，SCI 工作在多缓冲模式。与单缓冲模式相同，可采用轮询、DMA 或中断方式将待发送数据写入相关寄存器。待发送数据需写入 SCITD 寄存器。SCI 会等待数据写入 SCITD 寄存器，并自动将编程设定数量的字节传输到 SCITXSHF，然后逐一发送。

#### 22.3.1.5 SCI 低功耗模式

SCI/LIN 模块可进入本地低功耗模式或全局低功耗模式。全局低功耗模式由系统触发，不受 SCI/LIN 模块控制。在全局低功耗模式下，SCI/LIN 的所有时钟都会被关闭，因此该模块完全处于非活动状态。

本地低功耗模式通过置位 POWERDOWN 位触发，置位该位后，SCI/LIN 内部逻辑和模块寄存器的时钟会停止。置位 POWERDOWN 位会使 SCI 进入本地低功耗模式，清零该位则会使 SCI/LIN 退出本地低功耗模式。在本地掉电模式下，所有寄存器均可访问，因为任何寄存器访问操作都会仅为该次访问启用 SCI 的时钟。

唤醒中断用于使 SCI 在 LINRX 引脚检测到低电平时，自动退出低功耗模式，同时该中断会清零 POWERDOWN 位。若唤醒中断被禁用，则无论何时请求进入低功耗模式，SCI/LIN 都会立即进入，且 LINRX 引脚上的任何活动都不会使 SCI 退出低功耗模式。

注意：

在接收和传输过程中启用本地低功耗模式，若唤醒中断已启用，且在接收器正在接收数据时请求进入低功耗模式，则 SCI 会立即产生唤醒中断，以清零掉电位，从而阻止 SCI 进入低功耗模式，确保当前的接收操作完成。反之，若唤醒中断被禁用，则 SCI 会完成当前的接收操作后，再进入低功耗模式。

##### 22.3.1.5.1 多处理器通信的睡眠模式

当 SCI 接收数据并将其从 SCIRXSHF 传输到 SCIRD 时，会置位 RXRDY 位；若 RX INT ENA 位已置位，SCI 还会产生中断。该中断会触发 CPU 在接收下一帧数据之前，读取新接收的帧。在多处理器通信模式下，可增强这种默认行为，以实现对新数据的选择性指示。当 SCI 接收到与自身地址不匹配的地址帧时，可通过睡眠模式忽略该不匹配地址之后的数据，直至接收到下一地址帧。睡眠模式可同时用于空闲线多处理器模式和地址位多处理器模式。

若通过 SLEEP 位启用了睡眠模式，则 SCI 仅会将地址帧的数据从 SCIRXSHF 传输到 SCIRD。因此，在睡眠模式下，所有数据帧都会在 SCIRXSHF 寄存器中组装，但不会移位到 SCIRD 中，也不会触发接收中断或 DMA 请求。接收到地址帧后，SCIRXSHF 中的内容会被移入 SCIRD，此时软件需读取 SCIRD，并将接收到的地址与之前在软件中设置并存储在内存某处的地址进行比较（SCI 不具备地址比较的硬件功能），以判断自身是否为目标接收设备。若 SCI 是目标接收设备，软件必须清零 SLEEP 位，以便 SCI 将后续数据帧的数据加载到 SCIRD 中。

当 SCI 被寻址，且通过软件禁用睡眠模式以接收数据后，SCI 可通过检查 RXWAKE 位（SCIFLR.12）来判断何时接收到下一地址帧。若 SCIRD 中的当前值为地址，则该位置 1；若 SCIRD 中存储的是数据，则该位清零。若 RXWAKE 位置 1，软件需检查 SCIRD 中的地址是否与自身地址匹配。若 SCIRD 中的地址仍指向该 SCI，则可保持睡眠模式禁用状态；否则，可再次置位 SLEEP 位。

以下是睡眠模式操作的典型事件序列：

1. 配置 SCI，并启用睡眠模式和接收功能。

2. 接收到地址帧，产生接收中断。

3. 软件将接收到的地址帧与自身地址进行比较，判断该 SCI 不是目标接收设备，因此不改变 SLEEP 位的值。

4. 多个数据帧被移位到 SCIRXSHF 中，但不会传输到 SCIRD，也不会产生接收中断。

5. 接收到新的地址帧，产生接收中断。

6. 软件将接收到的地址帧与自身地址进行比较，判断该 SCI 是目标接收设备，并清零 SLEEP 位。

7. 移位到 SCIRXSHF 中的数据被传输到 SCIRD，每接收一帧数据帧都会产生接收中断。

8. 在每个中断服务程序中，软件通过检查 RXWAKE 位，判断当前帧是否为地址帧。

9. 接收到另一个地址帧，RXWAKE 位置 1，软件判断该 SCI 不是目标接收设备，并将 SLEEP 位重新置 1，后续数据帧不会产生接收中断。

通过忽略非目标数据帧，可减少中断的产生次数，从而避免 CPU 因处理这些无关数据的中断而占用过多资源。采用睡眠模式有助于释放部分 CPU 资源。

在睡眠模式激活期间，除 RXRDY 标志外，SCI 会继续更新接收器状态标志（见表）。这样，若接收线上发生错误，应用程序可立即响应错误并采取相应的纠正措施。

由于在睡眠模式启用时，RXRDY 位不会针对数据帧进行更新，因此 SCI 可在启用睡眠模式的同时，采用轮询算法。在这种情况下，当 RXRDY 位置位时，软件就知道接收到了新的地址帧。若该 SCI 不是目标接收设备，软件无需改变 SLEEP 位的值，可继续轮询 RXRDY 位。

## 22.3.2 本地互联网络模块

### 22.3.2.1 LIN 通信格式

SCI/LIN 模块可工作在 LIN 模式或 SCI 模式。LIN 模式运行所需的波特率生成、DMA 控制以及额外的接收 / 发送缓冲区等增强功能，同样也是增强型缓冲 SCI 模块的一部分。通过置位 SCIGCR1 中的 LIN MODE 位，可将模块配置为 LIN 模式。

注意：SCI/LIN 模块基于 SCI 平台构建，采用类似的采样方案：每个位进行 16 次采样，并对第 8、9、10 次采样结果进行多数表决；对于起始位，则采用前 3 次采样结果。

#### 22.3.2.1.1 LIN 标准

为实现与 LIN 2.0 标准的兼容性，相较于 LIN 1.3，额外增加了以下功能：

1. 支持 LIN 2.0 校验和

2. 增强型同步器有限状态机（FSM），支持帧处理

3. 增强型扩展帧处理功能

4. 增强型波特率发生器

5. 唤醒 / 睡眠更新功能

LIN 模块通过硬件实现了 LIN 协议规范 1.3 版和 2.0 版中定义的、占用 CPU 性能较多的功能。LIN 模块的主模式兼容 LIN 2.1 标准。

#### 22.3.2.1.2 消息帧

LIN 协议定义了消息帧格式，如图所示。每个帧包含一个主机头、一个响应、一个帧内响应间隔和字节间间隔。帧内响应间隔和字节间间隔可以为 0。

LIN 协议定义中不存在仲裁机制，因此，若多个从节点对同一个主节点头进行响应，会被检测为错误。

LIN 总线是单通道线与总线，具有两种二进制电平：显性电平（表示值 0）和隐性电平（表示值 1）。

消息帧

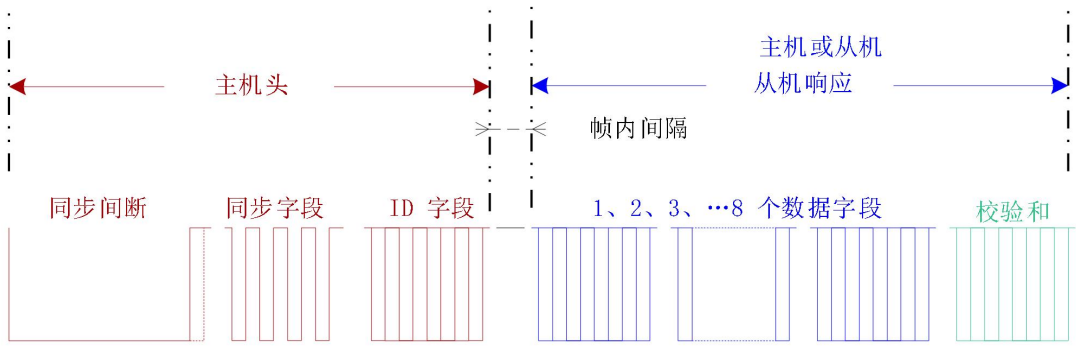



图 22-12 LIN 协议消息帧格式：主机头及响应

##### 22.3.2.1.2.1 消息头

消息头由主节点发起（见图），包含三个字段序列：

- 同步中断字段：用于标识消息的开始

- 同步字段：用于传递 LIN 总线的比特率信息
- ID 字段：用于标识消息的内容

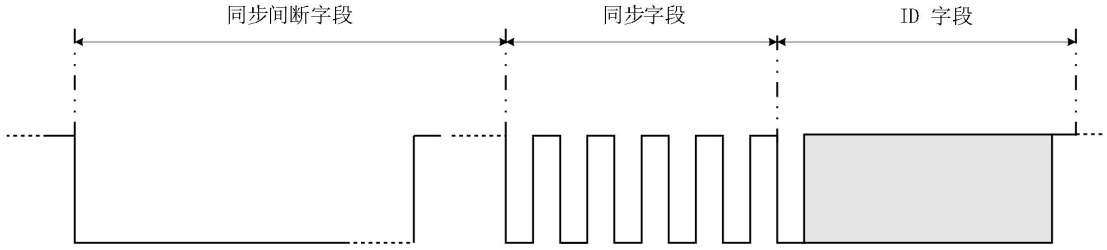



图 22-13 头的三个字段：同步中断、同步、ID

### 22.3.2.1.2.2 响应

响应格式如图所示，响应包含两种字段：数据字段和校验和字段。每个数据字段包含 1 个数据字节、1 个起始位和 1 个停止位，共 10 位，且按照最低有效位优先的顺序传输。校验和字段包含 1 个校验和字节、1 个起始位和 1 个停止位，其中校验和字节是对响应中所有数据字段的字节进行模 256 求和后取反得到的值。

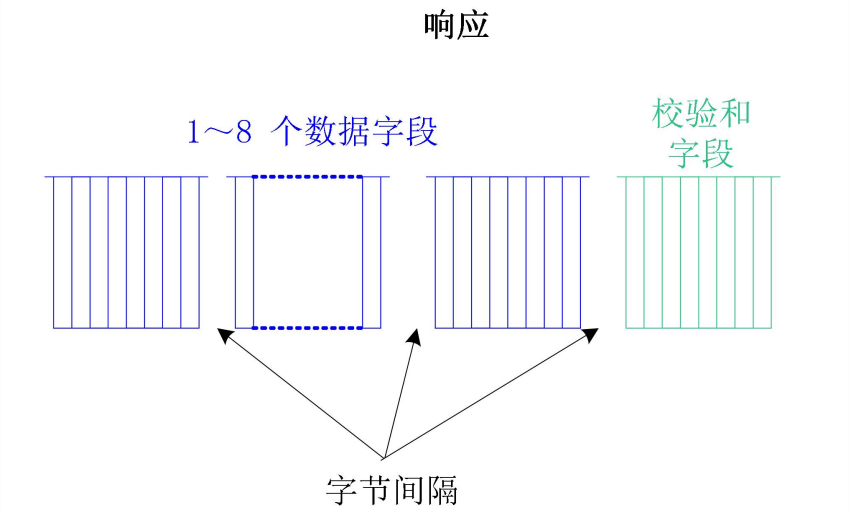



图 22-14 LIN 消息帧的响应格式

响应的格式为 N 个数据字段和 1 个校验和字段组成的数据流。通常情况下，N 的取值范围为 1-8，但扩展命令帧除外（见 22.3.2.1.6 节）。响应长度 N 可通过 ID 字段中的可选长度控制位（适用于 LIN 1.x 之前的标准，见表），或通过 SCIFORMAT 寄存器的 LENGTH [18:16] 位（见表）来指定。SCI/LIN 模块兼容 LIN 2.0 标准，支持 1-8 字节的响应长度。

表 22-7 LIN 1.3 之前标准中通过 IDBYTE 字段位 [5:4] 表示的响应长度信息

| IDBYTE[5:4] | 数据字节数 |
|---|---|
| 00 | 2 |
| 01 | 2 |
| 10 | 4 |
| 11 | 8 |

表 22-8 通过 SCIFORMAT [18:16] 编程设置的响应长度

| SCIFORMAT[18:16] | 字节数 |
|---|---|
| 000 | 1 |
| 001 | 2 |
| 010 | 3 |
| 011 | 4 |
| 100 | 5 |
| 101 | 6 |
| 110 | 7 |
| 111 | 8 |

#### 22.3.2.1.3 同步器

在主节点与从节点之间的通信中，同步器主要实现以下三个功能：生成主节点头数据流、与 LIN 总线同步以进行响应、本地检测超时。通过对 BRSR 寄存器中的预分频器进行编程，可设置比特率，使其与 LIN 描述文件中的 LIN_speed 值匹配。

LIN 同步器具体实现以下功能：主节点头信号生成、从节点检测并与消息头同步（可选波特率调整）、响应传输时序控制和超时控制。

LIN 同步器能够在任何时候检测到输入的 break 信号，并初始化通信。

#### 22.3.2.1.4 波特率

每个节点的传输波特率在初始时由 CPU 配置，该配置决定了位时间 $T_{bit}$。位时间由波特率选择寄存器（BRSR）中的字段 P 和 M 推导得出。此外，BRSR 寄存器中还有一个 3 位小数分频值（字段 U），可进一步微调数据字段的波特率。

BRSR 寄存器中预分频器值的取值范围如下：

P = 0, 1, 2, 3, . . . , $2^{24}$ - 1

M = 0, 1, 2, . . . , 15

U = 0, 1, 2, 3, 4, 5, 6, 7

BRSR 寄存器中的 P、M、U 值均支持用户编程。P 和 M 分频器可同时用于 SCI 模式和 LIN 模式，以选择波特率；U 值是一个额外的 3 位值，其作用是在每个上增加 “$aT_{VCLK}$ ”（其中 a=0,1）。若 ADAPT 位已置位，且 LIN 从节点工作在自适应波特率模式，则在接收消息头的过程中，通过测量同步字段，可自动获取所有这些分频器的值。

LIN 协议定义的波特率范围为：

$$1\ \text{kHz} \leq F_{LINCLK} \leq 20\ \text{kHz} \quad (22\text{-}6)$$

所有传输的位均按照周期 $T_{bit}$ 进行移入和移出。

#### 22.3.2.1.4.1 小数分频器

BRSR 寄存器中的 M 字段用于对整数预分频器 P 进行修正，以实现波特率的微调。M 值以 P 值的 1/16 为步长进行增量调整。

位时间以 VCLK 周期表示，公式如下：

对于所有非 0 的 P 值以及所有 M 值：

$$T_{bit} = 16(P + 1 + \frac{M}{16}) \times T_{VCLK} \quad (22\text{-}7)$$

$$P = 0：T_{bit} = 32 \times T_{VCLK} \quad (22\text{-}8)$$

因此，LINCLK（LIN 时钟）频率的计算公式如下：

对于所有非 0 的 P 值，

$$F_{LINCLK} = \frac{F_{VCLK}}{16(P + 1 + \frac{M}{16})} \quad (22\text{-}9)$$

对于 P=0，

$$F_{LINCLK} = \frac{F_{VCLK}}{32} \quad (22\text{-}10)$$

#### 22.3.2.1.4.2 超小数分频器

超小数分频方案适用于以下模式：

- LIN 主模式（同步字段 + 标识符字段 + 响应字段 + 校验和字段）
- LIN 从模式（响应字段 + 校验和字段）

### 22.3.2.1.4.2.1 LIN 模式下的超小数分频器

在 4 位小数分频器 M（BRSR [27:24]）的基础上，超小数分频器额外使用一个 3 位调制值，如表所示。同步字段（0x55）、标识符字段和响应字段均可视为由起始位和停止位封装的 8 位数据字节。表中值为 1 的位对应的（位时间）会额外增加一个 VCLK 周期。在 LIN 主模式下，位调制适用于同步字段 + 标识符字段 + 响应字段；在 LIN 从模式下，位调制适用于标识符字段 + 响应字段。

表 22-9 LIN 主模式和从模式下的超小数位调制

| BRSR[30:28] | 起始位 | D[0] | D[1] | D[2] | D[3] | D[4] | D[5] | D[6] | 停止位 |
|---|---|---|---|---|---|---|---|---|---|
| 0h | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1h | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2h | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3h | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 4h | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 5h | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 6h | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 7h | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

根据 BRSR [30:28] 的值，LIN 数据字段的波特率会在一定范围内变化，其平均值为外设内部时钟的 d 分数倍（0<d<1）。

瞬时位时间 $T_{bit}^{i}$ 以 $T_{VCLK}$ 表示，公式如下：

对于所有非 0 的 P 值，以及所有 M 值和 d 值（0 或 1），

$$T^{i}bit = [16(P + 1 + \frac{M}{16}) + d] \times T_{VCLK} \quad (22\text{-}11)$$

$$P = 0：T_{bit} = 32 \times T_{VCLK} \quad (22\text{-}12)$$

平均位时间 $T_{bit}^{a}$ 以 $T_{VCLK}$ 表示，公式如下：

对于所有非 0 的 P 值，以及所有 M 值和 d 值（0<d<1），

$$T^{a}bit = [16(P + 1 + \frac{M}{16}) + d] \times T_{VCLK} \quad (22\text{-}13)$$

$$P = 0：T_{bit} = 32 \times T_{VCLK} \quad (22\text{-}14)$$

### 22.3.2.1.5 消息头生成

LIN 协议消息头数据流支持自动生成，无需 CPU 干预。CPU 或 DMA 触发消息头生成后，由 LIN 状态机负责处理生成过程。主节点通过 CPU 或 DMA 向 LINID 寄存器中的 IDBYTE 写入数据，从而启动消息头生成。消息头始终由主节

点发送，用于启动 LIN 通信，包含三个字段：同步中断字段、同步字段和标识符字段，如图所示。

注：LIN 协议在标识符中使用校验位，长度控制位是 LIN 协议的可选部分。

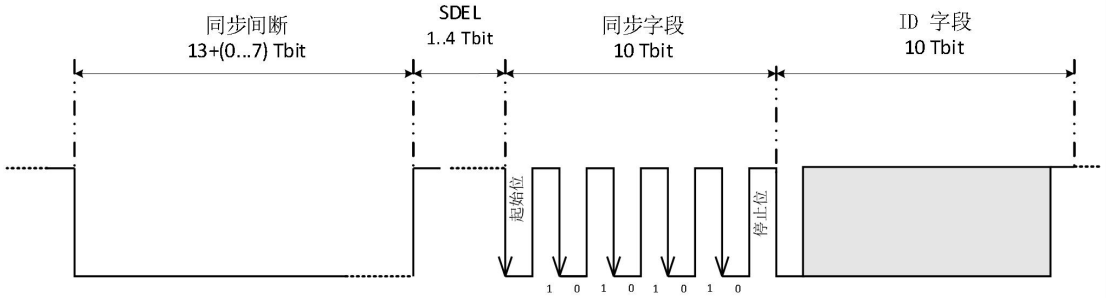



图 22-15 以 Tbit 为单位的消息帧头

- 中断字段由以下两部分组成：
  - 同步中断（SYNCH BREAK）：包含最少 13 个显性低电平位，最多 20 个显性位。通过 LINCOMP 寄存器中的 3 位 SBREAK 值，可将同步中断长度从最小值进行延长。
  - 同步中断定界符（SDEL）：包含最少 1 个隐性高电平位，最多 4 个隐性位。该定界符用于标识同步中断字段的结束，其长度由 LINCOMP 寄存器中的 2 位 SDEL 值决定。
- 同步字段（SYNCH FIELD）包含 1 个起始位、1 个字节（值为 0x55）和 1 个停止位，用于传递（位时间）信息，并实现 LIN 总线节点之间的重新同步。
- ID 字段中，6 位可用作标识符，还可包含可选的长度控制位和两个可选的标识符校验位。若 PARITY ENA（校验使能）位已置位，则会使用并检查标识符校验位。若不使用长度控制位，则总共可包含 64 个标识符（含校验位）；若既不使用长度控制位，也不使用校验位，则最多可包含 256 个标识符。标识符字段的结构如图 22-16 所示。

注：可选长度控制位仅适用于 LIN 1.3 之前的标准。若兼容 LIN 1.3 之前的标准，IDBYTE 字段用于传递响应长度信息；对于 LIN 协议的后续版本，响应长度存储在 SCIFORMAT 寄存器中。

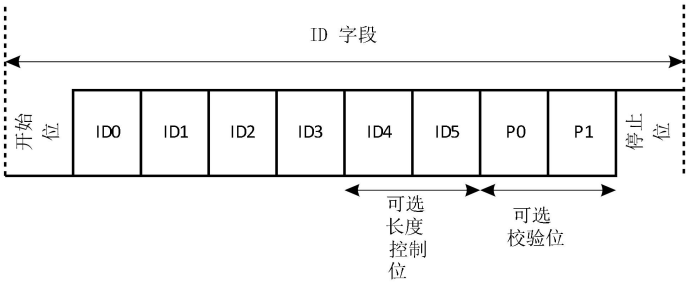



图 22-16 ID 字段

注：若配置为从节点的 LIN 模块工作在多缓冲模式下，且在传输数据过程中接收到新的消息头，则该模块可能会用之前被中断响应的数据（而非与新标识符对应的 data）进行响应。为避免这种情况，可采用以下步骤：

1. 在响应传输过程中，检查是否发生位错误（BE）。若 BE 标志置位，表明 LIN 总线上发生了冲突（此处是由于新的同步中断导致）。

2. 在位错误中断服务程序（ISR）中，针对输入的标识符（ID），在发送匹配（TX Match）时，将下一组待传输数据配置到 TD0 和 TD1 寄存器中。在写入 TD0/TD1 之前，需确保之前未因位错误而进行过更新，否则可能会针对同一个标识符两次写入 TD0/TD1。

3. 接收到完整的标识符后，根据匹配结果，该节点会传输新配置的数据。

#### 22.3.2.1.5.1 事件触发帧处理

LIN 2.0 协议采用事件触发帧，这类帧偶尔会引发冲突，其处理需通过软件实现。

若没有从节点对事件触发帧的消息头进行响应，主节点会置位 NRE 标志；若 NRE 中断已启用，则会产生 NRE 中断。若发生冲突，在产生 NRE 错误之前，可能会先出现帧错误和校验和错误。这些错误会被标记，若对应的中断已启用，则会产生相应的中断。

帧错误和校验和错误的产生取决于响应从节点的行为和同步情况。若从节点完全同步，且在冲突发生后立即停止传输，则即使发生了冲突，也可能仅标记 NRE

错误。为判断在 NRE 错误标记之前是否已接收到一个字节，可将 BUS BUSY 标志作为指示符。

当接收到消息头的第一个位时，BUS BUSY 标志置位，直至消息头接收完成后才清零；当接收到响应的第一个位时，该标志会再次置位。若发生冲突，该标志会与 NRE 标志在同一周期内清零。

软件可按照以下流程进行处理：

- 消息头接收完成后（轮询 RXID 标志），等待 BUS BUSY 标志置位或 NRE 标志置位。
- 若在 NRE 标志置位之前，BUS BUSY 标志未置位，则表明确实无响应（总线上未传输任何数据）。
- 若 BUS BUSY 标志置位，则继续等待 NRE 标志置位或接收成功。若 NRE 标志置位，则表明总线上发生了冲突。

即使发生冲突，仍可在接收缓冲区（寄存器 LINRD0 和 LINRD1）中读取到（已损坏的）接收数据。

#### 22.3.2.1.5.2 消息头接收与自适应波特率

LIN 模块提供了一项可选功能：从节点的波特率可调整为检测到的比特率。通过置位 ADAPT 位，可启用自适应波特率选项。从节点在接收消息头的过程中，会通过检测同步字段来测量波特率。若 ADAPT 位已置位，从节点会将测量得到的波特率与自身编程设定的波特率进行比较，必要时调整为 LIN 总线的波特率。

从节点可将波特率调整为与编程设定波特率偏差在 ±10% 范围内的任意测量波特率。例如，若编程设定的期望波特率为 20kbps，则从节点可检测 18kbps 至 22kbps 范围内的任意波特率，并相应地进行调整。MBRSR 寄存器预分频器的值可通过以下公式计算：

$$\mathrm{MBR} = \frac{F_{VCLK}}{1.1 \times F_{LINCLK}} \qquad (22\text{-}15)$$

LIN 同步器会进行两项测量：BRK_count 和 BAUD_count（见图）。在接收消息头以验证同步字段的过程中，始终会计算这两个值（见图）。

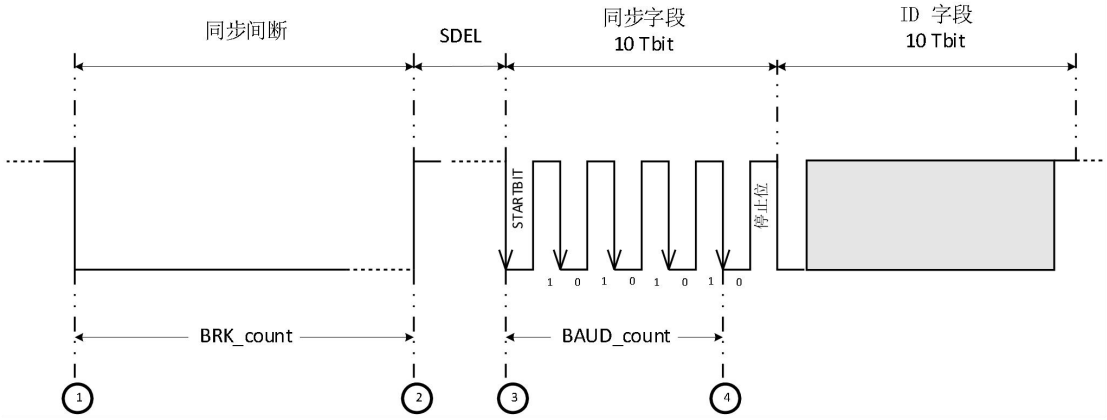



图 22-17 同步测量

通过测量 BRK_count 和 BAUD_count 的值，可检测到有效的同步中断序列，具体检测方式如图所示。图中的 4 个编号事件标识了同步器计数器的启动/停止时刻，同步器计数器以 VCLK 作为时基。

同步器计数器用于相对于检测节点的（位时间）测量同步中断的时长。根据 LIN 协议要求，从节点接收同步中断时，采用 11 作为阈值。为检测同步中断的显性数据流，同步器计数器在 LINRX 引脚出现下降沿时启动，在出现上升沿时停止。检测到同步中断定界符后，会保存同步器计数器的值，然后将计数器复位。

检测到 5 个连续的下降沿时，会测量 BAUD_count 的值。位时序计算及精度一致性验证均按照 LIN 2.0 版本的建议实现。从节点可通过将 BAUD_count 除以 8，计算出单个的时长。此外，为验证检测到的边沿之间的一致性，需进行以下评估：

BAUD_count +（BAUD_count 右移 2 位）+（BAUD_count 右移 3 位）≤ BRK_count

将 BAUD_count 值右移 3 位，并使用第一个非有效位进行舍入，以获得位时间（Tbit）单位。若 ADAPT 位已置位，则会将检测到的波特率与预编程的波特率进行比较。

如图所示，在消息头接收处理过程中，对于固定波特率的情况，若测量得到的 BRK_count 值小于 11，则判定同步中断无效。若 ADAPT 位已置位，则会使用 MBRS 寄存器测量 BRK_count 和 BAUD_count 的值，并自动调整为 LIN 总线允许的任意波特率（参考《LIN 协议规范包 2.0》）。

注：在自适应模式下，MBRS 分频器的设置需确保其最大波特率不超过 LIN 网络中预期工作波特率的 10%。否则，可能会将 0x00 数据字节误检测为同步中断。

从机节点对应的同步间隔阈值为 11 个位时间（$T_{bit}$）。而根据 LIN v1.3 规范的规定，同步间隔信号本身的长度为 13 个位时间（$T_{bit}$）。

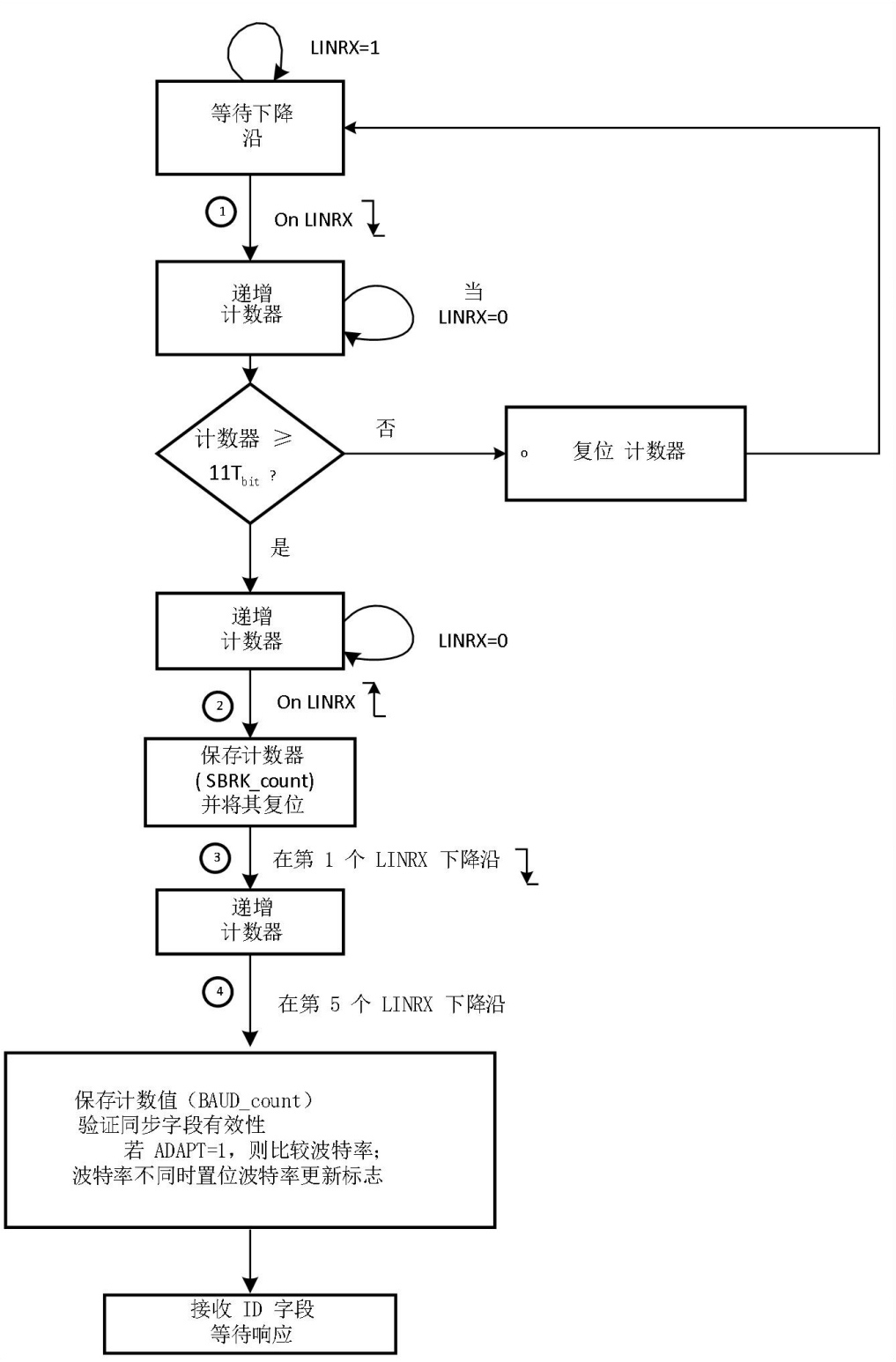



图 22-18 同步验证流程与波特率校准

若未在规定的容差范围内检测到同步字段，则会置位同步字段不一致错误（ISFE）标志。若 SCISETINT 寄存器中的相应位已置位（即 ISFE 中断已启用），则会产生 ISFE 中断。只有在同步字段验证成功后，才能接收标识符字节。无论何时检测到有效的中断（时长大于 11），接收器状态机都会复位，以接收新的帧，但该复位操作仅在响应状态下有效；若在接收消息头的过程中出现额外的同步中断，则不会执行复位。

注：当发生同步字段不一致（ISFE）错误时，建议应用程序复位 SWnRST 位，然后重新置位该位，以确保内部状态机恢复到正常状态。

#### 22.3.2.1.6 扩展帧处理

LIN 2.0 及更早版本的协议包含两个扩展帧，其标识符分别为 62（用户自定义）和 63（保留扩展）。用户自定义帧（标识符 62，即 0x3E）的响应数据长度不受限制，该标识符对应的长度在网络配置时设定，并在 LIN 总线节点之间共享。

当接收到标识符为 0x3E 的消息头时，会触发扩展帧通信（见图）。与标准帧不同，扩展帧通信触发后，在发送下一个消息头之前，需先停止当前的扩展帧通信。通过置位 STOP EXT FRAME（停止扩展帧）位，可停止扩展帧通信。

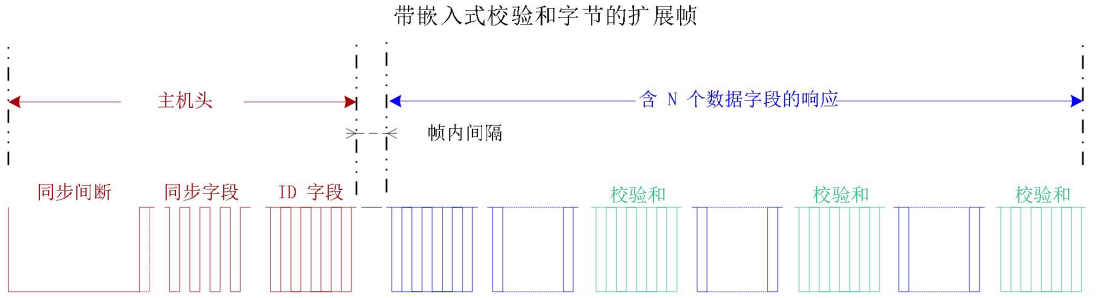



图 22-19 扩展帧响应中的可选嵌入式校验和

若已启用 ID 中断且存在匹配，接收到标识符 62（0x3E）时会产生标识符中断。该中断使 CPU 能够通过软件计数器跟踪发送的字节数，并决定何时计算并插入校验和字节（建议按固定周期插入）。为实现此过程，需使用 SC（位。向发送校验和位（SC）写入数据，会触发自动发送校验和字节的操作。根据 LIN 协议要求，最后一个数据字段可以始终是校验和。

在网络配置时会设定校验和插入的周期，接收节点可利用该周期评估正在传输的消息的校验和，从而提高可靠性。

对于发送节点，每次置位发送校验和位（SC）时，会自动嵌入校验和。对于接收节点，每次置位比较校验和位（CC）时，会对校验和进行比较（见图）。

注：LIN 2.0 增强型校验和不适用于保留标识符，保留标识符始终使用标准校验和。

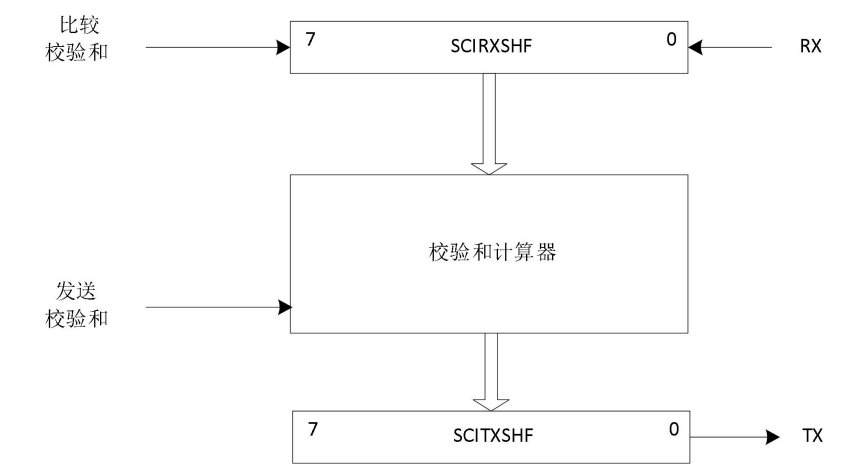



图 22-20 扩展帧的校验和比对与发送

### 22.3.2.1.7 超时控制

任何监听总线且等待主节点发起响应的 LIN 节点，都可标记无响应错误超时事件。LIN 协议定义了四种类型的超时事件，均由 LIN 模块的硬件进行处理，具体如下：

- 无响应超时错误
- 总线空闲检测
- 唤醒信号超时
- 三次唤醒信号超时

#### 22.3.2.1.7.1 无响应错误（NRE）

当任何等待响应的节点等待（最大帧时长）后，消息帧仍未在允许的最大时长内完成传输时，会发生无响应错误。超时后，会在 SCIFLR（SCI 标志寄存器）的

NRE（无响应错误）位中标记无响应错误（NRE）；若 NRE 中断已启用，则会产生中断。

根据 LIN 1.3 标准的规定，传输一帧的最短时长为：

$T_{FRAME_MIN} = T_{HEADER_MIN} + T_{DATA_FIELD} + T_{CHECKSUM_FIELD} = 44 + 10N$

其中，N 为数据字段的数量。

最长帧时长的计算公式为：

$T_{FRAME_MAX} = T_{FRAME_MIN} \times 1.4 = (44 + 10N) \times 1.4$

超时值由数据字段数量 N 推导得出，具体数值见表。N 值可嵌入消息头的 ID 字段中，也可在描述文件中定义。若为后一种情况，则由 SCIFORMAT 寄存器中的 3 位 CHAR 值指定 N 的值。

注：标识符字段的长度编码不适用于两个扩展帧标识符（标识符 0x3E（62）和 0x3F（63））。对于这两个标识符，其标识符字段之后可跟随任意数量的数据字节字段。此外，LIN 2.0 协议规范指出，标识符 0x3F（63）不可使用。对于这两种情况，LIN 硬件不处理 NRE。

表 22-10 以 Tbit 为单位的超时值

| N | 数据字段时长（$T_{DATA_FIELD}$） | 最短帧时长（$T_{FRAME_MIN}$） | 最长帧时长（$T_{FRAME_MAX}$） |
|---|---|---|---|
| 1 | 10 | 54 | 76 |
| 2 | 20 | 64 | 90 |
| 3 | 30 | 74 | 104 |
| 4 | 40 | 84 | 118 |
| 5 | 50 | 94 | 132 |
| 6 | 60 | 104 | 146 |
| 7 | 70 | 114 | 160 |
| 8 | 80 | 124 | 174 |

### 22.3.2.1.7.2 总线空闲检测

第二种超时类型发生在节点检测到 LIN 总线处于非活动状态时，即总线上未检测到隐性电平和显性电平之间的跳变。总线非活动状态的判定条件为：总线保持无活动状态的时间至少为 4 秒（对于 20kbps 的最高总线速率，相当于 80,000

个周期）。若节点在 TIMEOUT 位已置位的情况下检测到总线无活动，则认为 LIN 总线处于睡眠模式。应用程序软件可利用 Timeout 标志判断 LIN 总线何时处于非活动状态，并通过写入 POWERDOWN 位，将 LIN 模块置于睡眠模式。

注：标记超时后，在进入低功耗模式之前，必须先触发 SWnRESET。这是因为在空闲周期之前，总线上可能存在未完成的帧，复位操作可确保接收器恢复初始状态。

#### 22.3.2.1.7.3 唤醒信号超时与三次唤醒信号超时

第三种和第四种超时类型与唤醒信号相关。发起唤醒操作的节点必须等待主节点在规定时间内发送消息头，若超时则触发相应超时事件。

### 22.3.2.1.8 TXRX 错误检测器（TED）

TXRX 错误检测器逻辑（TED）可检测以下错误来源。该逻辑由位监视器、标识符校验器和校验和错误检测器组成，具体可检测的错误如下：

- 位错误（BE）
- 物理总线错误（PBE）
- 标识符校验错误（PE）
- 校验和错误（CE）

上述所有错误（BE、PBE、PE、CE）都会被标记；若对应的中断已启用，则会为标记的错误产生中断。只有当一帧数据传输或接收完成且未检测到任何错误时，该消息对发送器和接收器而言才是有效的。

#### 22.3.2.1.8.1 位错误

当监控到的位值与发送的位值不同时，会在该位的传输时刻检测到位错误（BE）。SCIFLR 寄存器中的 BE 标志用于指示位错误。标记位错误后，传输会在不晚于下一个字节时中止。位监视器通过读取 LINRX 引脚上的电平（如图所示），确保 LINTX 引脚上发送的位在 LIN 总线上的实际值正确。

注：若在从节点响应过程中因接收消息头而发生位错误，则不会为新帧设置无响应错误（NRE）/ 超时（TIMEOUT）标志。

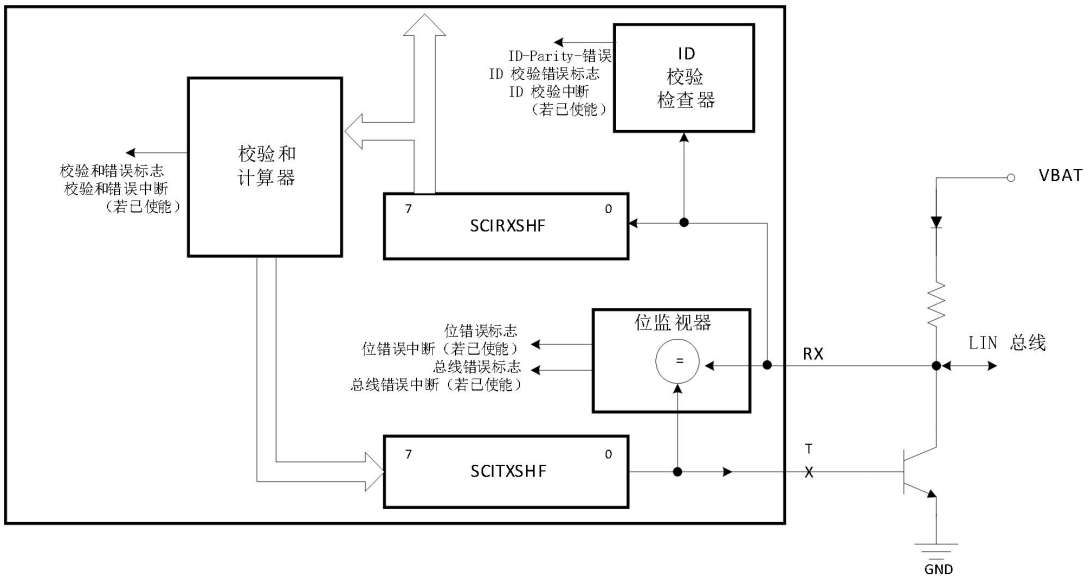



图 22-21 TXRX 错误检测器

### 22.3.2.1.8.2 物理总线错误

若总线上无法生成有效消息（如总线短路至地（GND）或电源（VBAT）），主节点必须检测出物理总线错误（PBE）。在位监视器检测物理总线错误的过程中：若在消息头传输时无法生成同步中断（例如，总线短路至 VBAT），或无法生成同步中断定界符（例如，总线短路至 GND），则判定为物理总线错误。一旦同步中断定界符验证通过，此帧中所有其他监控位值与发送位值的偏差均会标记为位错误（BE）。

### 22.3.2.1.8.3 标识符校验错误

若启用了校验功能，当接收到的标识符（ID）字节的两个校验位中任意一个与接收节点计算出的校验值不匹配时，会检测到标识符校验错误（PE）。这两个校验位通过以下混合校验算法生成：

奇校验位（P1）：P1 = ID1 ⊕ ID3 ⊕ ID4 ⊕ ID5（⊕表示异或运算）

偶校验位（P0）：P0 = ID0 ⊕ ID1 ⊕ ID2 ⊕ ID4

若检测到 ID 校验错误，会标记该错误，且接收到的标识符无效。

### 22.3.2.1.8.4 校验和错误

在接收端，若对所有接收数据字节（若为增强型校验和类型，则包含标识符字节）计算模 256 和，并与校验和字节相加后结果不为 0xFF，则会检测到校验和错误（CE）并进行标记。计算模 256 和时，需对每个字节进行带进位加法运算，每次加法产生的进位位会加到结果和的最低有效位上。

对于发送节点，采用标准校验和实现时，消息末尾发送的校验和字节是所有数据字节求和后取反得到的值（见图）。采用符合 LIN 2.0 标准的增强型校验和实现时，校验和字节是标识符字节与所有数据字节求和后取反得到的值（见图）。标准校验和实现始终可用于保留标识符 60 至 63，在此情况下，CTYPE 位会被覆盖。对于信号承载帧标识符（0 至 59），所使用的校验和类型由 CTYPE 位决定。

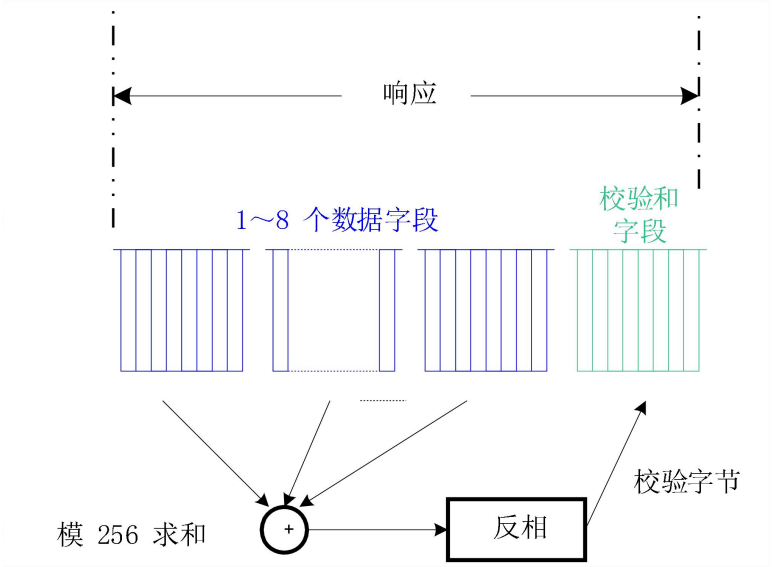



图 22-22 发送节点的标准校验和生成

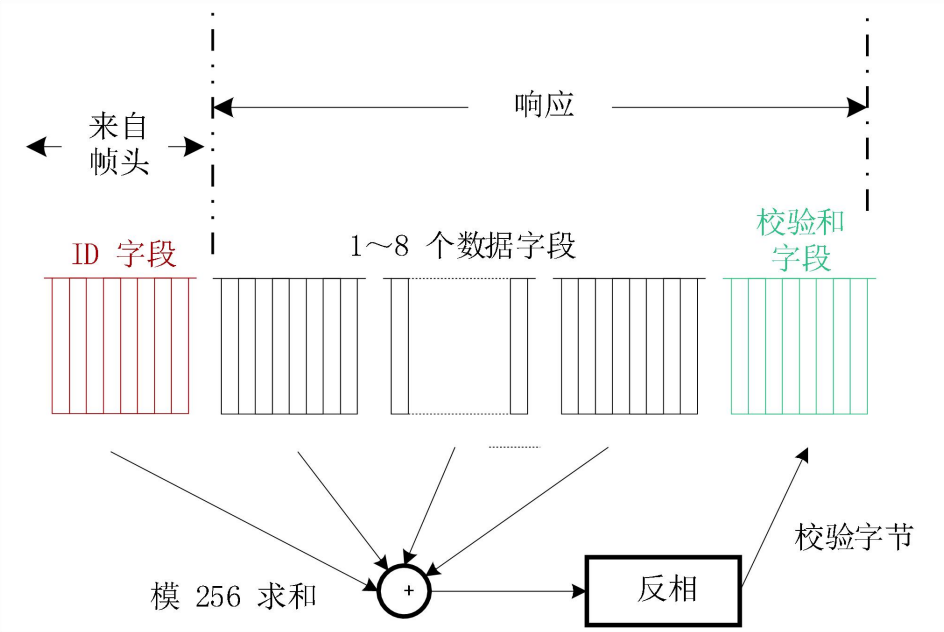



图 22-23 发送节点的 LIN 2.0 兼容型增强校验和生成

### 22.3.2.1.9 消息过滤与验证

消息过滤会利用完整的标识符来确定哪些节点参与响应（接收或发送响应）。因此，需使用两个接收掩码，如图所示。在接收消息头的过程中，所有节点都会对 ID 字段（其结构详见图）进行过滤，以确定自身是否需要对当前消息进行响应发送或响应接收。消息标识符过滤包含两个掩码：一个用于允许响应接收，另一个用于触发响应发送（见图）。所有节点会将接收到的标识符与 LINID 寄存器的“ID 响应器任务字节”中存储的标识符进行比较，并使用 LINMASK 寄存器中的 RX ID MASK 和 TX ID MASK 字段，对无需比较的标识符位进行过滤。

若 RX 匹配且无校验错误，同时 RXENA 位已置 1，则会设置 ID RX 标志；若对应的中断已启用，会产生中断。若 TX 匹配且无校验错误，同时 TXENA 位已置 1，则会设置 ID TX 标志；若 SCISETINT 寄存器中对应的中断已启用，会产生中断。

被掩码的位在比较过程中视为“无关位”。若要为一组标识符构建掩码，可使用异或运算。

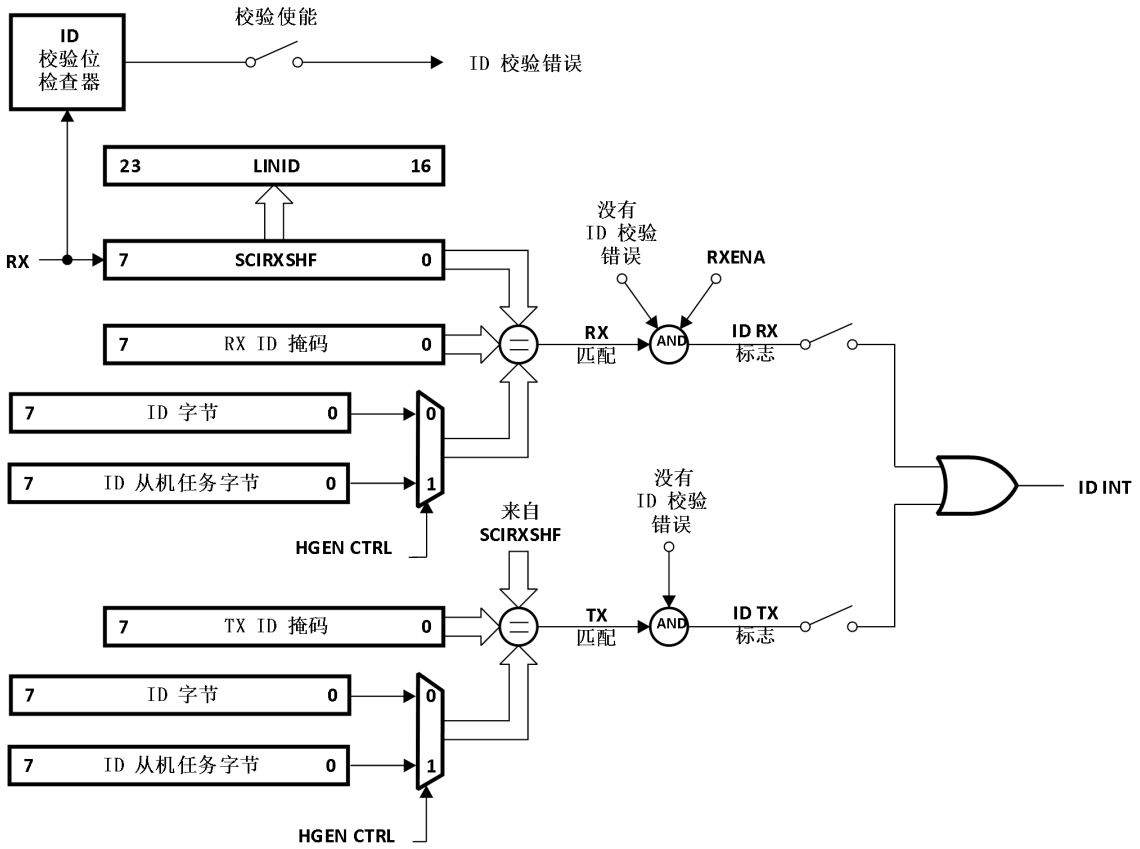



图 22-24 ID 的接收、过滤与验证

例如，若要构建一个掩码以接收标识符 0x26 和 0x25，且（即比较 5 个最高有效位（MSB），过滤 3 个最低有效位（LSB）），则接收掩码可通过以下方式计算：

$$(0x26 + 0x25) \oplus 0x20 = 0x07 \qquad (22\text{-}16)$$

全 0 掩码会将移位寄存器中接收到的标识符的所有位，与 LINID [7:0] 寄存器中的 ID 字节进行逐位比较。若 HGEN_CTRL 寄存器置 1，掩码为 0xFF 时将始终判定为匹配；全 1 掩码会屏蔽接收到的标识符的所有位，因此无论 LINID 寄存器中响应器任务 ID 字节字段的内容如何，都会判定为标识符匹配。

注：当 HGEN 控制位为 0 时，LIN 节点会将接收到的标识符与 LINID 寄存器中的标识符字节段进行比对，并通过 LINMASK 寄存器中的 RX ID MASK 和 TX ID MASK，对无需参与比对的标识位进行过滤。

若接收标识符匹配、无校验错误且RXENA已置 1，系统会置位ID RX 标志；若相关中断使能，此时将触发中断。掩码全 0 时，移位寄存器中接收标识符的所有位都会与 LINID [7:0] 中的标识符字节段进行比对；掩码全 1 时，接收标识符的所有位都会被过滤，不会产生匹配结果。

如果 HGEN CTLR = 1：

- 接收的 ID 会与 ID-Responder Task 字节进行比较，比较时使用 RXID 掩码和 TXID 掩码。
- 全为 1 的掩码始终会产生匹配。
- 全为 0 的掩码意味着所有位必须相同才能产生匹配。
- 如果掩码中有一些位为 1，那么这些位不用于过滤标准。

如果 HGEN CTRL = 0：

- 接收的 ID 会与 ID 字节进行比较，比较时使用 RXID 掩码和 TXID 掩码。
- 全为 1 的掩码不会产生匹配。
- 全为 0 的掩码意味着所有位必须相同才能产生匹配。
- 如果掩码中有一些位为 1，那么这些位不用于过滤标准。

在报头接收期间，接收的标识符会被复制到接收 ID 字段 LINID[23:16]。如果没有奇偶校验错误，且存在 TX 匹配或 RX 匹配，则会设置相应的 TX 或 RX ID 标志。如果 ID 中断已启用，则会产生 ID 中断。

ID 中断产生后，CPU 可以读取接收 ID 字段 LINID[23:16]，并确定要加载到发送缓冲区中的响应。

注：当字节 0 写入 TD0（LINTD0[31:24]）时，响应传输会自动生成。

在多缓冲区模式下，当所有响应数据字节和校验和字节都复制到移位寄存器 SCITXSHF 时，TXRDY 标志会被置位。在非多缓冲区模式下，每次有一个字节复制到 SCITXSHF 寄存器时，TXRDY 标志就会被置位；此外，在帧的最后一个字节（即校验和字节）复制到 SCITXSHF 寄存器后，该标志也会被置位。

在多缓冲区模式下，当发送缓冲区 TDy 和 SCITXSHF 移位寄存器都为空且校验和已发送时，TXEMPTY 标志会被置位。在非多缓冲区模式下，每次 TD0 和 SCITXSHF 为空时，TXEMPTY 都会被置位，但帧的最后一个字节除外，此时还必须发送校验和字节。

如果启用了奇偶校验，所有从机接收节点都会使用接收到的 ID 字节的全部八位来验证标识符。如果检测到 ID 奇偶校验错误，SCI/LIN 会标记标识符已损坏。

#### 22.3.2.1.10 接收缓冲区

为了在中断模式或 DMA 模式下接收 LIN N 字节（N = 1–8）响应时降低 CPU 负载，SCI/LIN 模块配备了八个接收缓冲区。这些缓冲区可以将整个 LIN 响应存储在就绪（RDy）接收缓冲区中。图中展示了接收缓冲区的情况。

数据字节后的校验和字节由内部校验和计算器进行验证。校验和错误（CE）标志指示校验和错误，若在 SCISETINT 寄存器中启用了相关功能，则会产生 CE 中断。

如果启用了多缓冲区模式，3 位多缓冲区计数器会对从 SCIRXSHF 寄存器传输到就绪接收缓冲区的数据字节进行计数；如果禁用了多缓冲区模式，则对传输到 RD0 的数据字节进行计数。3 位比较寄存器中存储的是预期接收的数据字节数。如果 ID 字节字段不传递消息长度，则长度（LENGTH）值表示预期长度，并用于加载 3 位比较寄存器。可以通过 COMM MODE 位选择使用长度控制字段还是 LENGTH 值。

如果帧没有响应接收错误（例如，没有校验和错误、帧错误和溢出错误），则在接收响应后可能会产生接收中断、接收就绪（RXRDY）标志以及 DMA 请求（RXDMA）。在确认接收之前，会先对校验和字节进行比较。根据是否启用多缓冲区模式（MBUF MODE 位），可以为每个接收的字节或整个响应生成 DMA 请求。

注：

在多缓冲区模式下，以下是与清除 RXRDY 标志位相关的场景：

1. 无法通过读取 SCIINTVECT0/1 寄存器中相应的中断偏移量来清除 RXRDY 标志。

2. 当 LENGTH（长度）小于或等于 4 时，读取 RD0 寄存器会清除 RXRDY 标志。

3. 当 LENGTH（长度）大于 4 时，读取 RD1 寄存器会清除 RXRDY 标志。

#### 22.3.2.1.11 发送缓冲区

为了降低在中断模式或 DMA 模式下传输 LIN N 字节（N=1–8）响应时的 CPU 负载，SCI/LIN 模块在 LINTD0 和 LINTD1 中设有 8 个发送缓冲区，即 TD0–TD7。借助这些发送缓冲区，可将整个 LIN 响应字段预加载到 TXy 发送缓冲区中。可选地，当多缓冲区模式未启用（MBUF MODE 位）时，可逐字节进行 DMA 传输。图中展示了发送缓冲区。

如果启用了多缓冲区模式，3 位多缓冲区计数器会对从 TDy 发送缓冲区寄存器传输的数据字节进行计数；如果禁用了多缓冲区模式，则对从 TD0 到 SCITXSHF 传输的数据字节进行计数。3 位比较寄存器中包含预期要传输的数据字节数。如果 ID 字段不用于传递消息长度，则 LENGTH 值表示预期长度，并被用来加载 3 位比较寄存器。可通过 COMM MODE 位选择使用长度控制字段还是 LENGTH 值。

传输响应后，可能会产生发送中断（TX 中断）、发送就绪标志（TXRDY 标志）以及 DMA 请求（TXDMA）。根据多缓冲区模式是否启用（MBUF MODE 位），可为每个传输的字节或整个响应生成 DMA 请求。

校验和字节由校验和计算器自动生成，并在数据字段传输完成后发送。3 位多缓冲区计数器对从 TDy 缓冲区传输到 SCITXSHF 寄存器的数据字节进行计数。

注：

通过使用 SCICLRINT 寄存器禁用相应中断，或者使用 TXENA 位禁用发送器，可在将下一系列数据写入发送缓冲区 LINTD0 和 LINTD1 之前消除发送中断请求。

### 22.3.2.2 LIN 中断

如 22.3.1.2 节所述，LIN 模式和 SCI 模式共享同一个中断模块。SCI/LIN 模块共有 16 个中断源，其中 8 个为 LIN 模式专用，具体如表所示。

图中展示了 LIN 消息帧的结构，及其可能产生的 LIN 中断的时序和顺序。

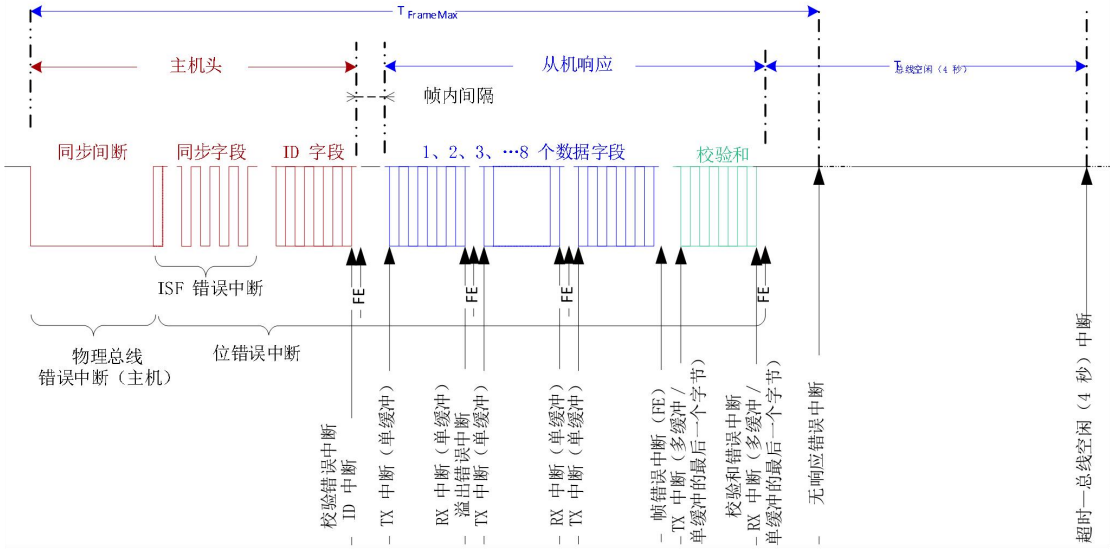



图 22-25 显示 LIN 中断时序和顺序的 LIN 消息帧

### 22.3.2.3 LIN 中断服务

处理中断时，需在清除全局中断（LIN_GLB_INT_CLR）之前，先清除标志寄存器（SCIFLR）中对应的标志。中断服务程序（ISR）可遵循以下步骤，以防止产生虚假中断或重复中断：

- 清除 SCIFLR 寄存器中的 LIN 中断标志。
- 读取 LIN 中断状态寄存器，确认标志已清除。
- 清除 LIN_GLB_INT_CLR 中的全局中断标志位。

注：发送中断会在 LIN 发送器准备好接收新数据之前产生。在 LIN 发送中断服务程序内部，软件可等待缓冲区完全为空后再加载下一批数据，具体可通过轮询 SCIFLR.BUSY（总线忙）标志，直至其变为 0 来实现。

#### 22.3.2.4 LIN DMA 接口

LIN DMA 接口采用 SCI DMA 接口逻辑。SCI/LIN 模块支持接收（RXDMA 请求）和发送（TXDMA 请求）的 DMA 请求。根据多缓冲使能控制位（MBUF MODE）是否启用，DMA 传输分为两种模式。

注：不要使用 DMA 向多个 slave ID 传输数据。向 LINID 寄存器写入数据会触发新的传输，而 DMA 可能在 LIN 状态机准备好接收新标识符之前就向 LINID 寄存器写入数据，导致 LIN 错过此次传输。

##### 22.3.2.4.1 LIN 接收 DMA 请求

在 LIN 模式下：

- 若启用多缓冲选项，当接收到的响应（最多 8 个数据字节）传输到接收缓冲区（RDy）后，会产生 DMA 请求。
- 若禁用多缓冲选项，则会以字节为单位产生 DMA 请求，直至接收完所有预期的响应数据字段。

可通过 SET RX DMA 位和 CLR RX DMA 位，分别启用和禁用该 DMA 功能。

##### 22.3.2.4.2 LIN 发送 DMA 请求

在 LIN 模式下：

- 若启用多缓冲选项，当完成一次传输（最多 8 个数据字节，存储在 LINTD0 和 LINTD1 寄存器的发送缓冲区（TDy）中）后，会产生 DMA 请求，以便为下一次传输重新加载发送缓冲区。
- 若禁用多缓冲选项，则会以字节为单位产生 DMA 请求，直至所有字节传输完成。

可通过 SET TX DMA 位和 CLR TX DMA 位，分别启用和禁用该 DMA 功能。

### 22.3.2.5 LIN 配置

以下列出了软件在 LIN 模式下传输或接收数据之前可执行的配置步骤。只要在配置 LIN 的整个过程中，SCIGCR1 中的 SWnRST 位始终保持为 0，寄存器的编程顺序就无关紧要。

- 置位 RESET 位，启用 LIN。
- 在配置 LIN 之前，将 SWnRST 位清零。
- 置位 RX FUNC 位和 TX FUNC 位，启用 LINRX 和 LINTX 引脚。
- 编程 LIN MODE 位，选择 LIN 模式。
- 编程 CLOCK 位，选择主模式或从模式。
- 编程 SCIGCR1 寄存器，选择所需的帧格式（校验和、校验、长度控制）。
- 编程 MBUF MODE 位，选择多缓冲模式。
- 编程 BRSR，选择通信所需的波特率。
- 编程 MBRSR，设置通信所需的最大波特率。
- 置位 CONT 位，使 LIN 在仿真断点时不会暂停，直至当前的接收或传输操作完成（该位仅在仿真环境中使用）。
- 若需要执行自测试，可置位 LOOP BACK 位，将发送器与接收器在内部连接。
- 若需要接收数据，选择 RXENA 位并置 1。
- 若需要发送数据，选择 TXENA 位并置 1。
- 选择 LINMASK 存器中的 RX ID MASK 和 TX ID MASK 字段并进行配置。
- LIN 配置完成后，将 SWnRST 位置 1。
- 执行数据接收或发送操作。

注：若 TXENA 位已置位且 SWnRST 位已释放，LIN 仅会产生新的 DMA 请求，不会产生新的发送中断请求。若使用中断方式，需通过软件写入待传输数据，然后写入 LIN TX 相关寄存器来启动第一次传输。

#### 22.3.2.5.1 接收数据

只有当 RX FUNC 位和 RXENA 位都设置为 1 时，LIN 接收器才被启用以接收消息。如果未设置 RX FUNC 位，LINRX 引脚将作为通用 I/O 引脚而非 LIN 功能引脚工作。在通过 RX 匹配接收到有效的 LIN ID 后，ID RX FLAG 会被置位。如果启用了 ID 中断，则会产生 ID 中断。

##### 22.3.2.5.1.1 单缓冲模式接收数据

当 MBUF 模式位被清零时，将选择单缓冲区模式。在此模式下，当 LIN 将新接收的数据从 SCIRXSHF 传输到 RD0 时，会设置 RXRDY 位。当 RD0 中的新数据被读取后，SCI 会清除 RXRDY 位。此外，当数据从 SCIRXSHF 传输到 RD0 时，如果在接收的数据中检测到帧错误（FE）、溢出错误（OE）或奇偶校验错误（PE）中的任何一种，LIN 会设置相应的标志位。这些错误状态支持可配置的中断功能。

您可以通过以下方式接收数据：

1. 查询接收就绪标志
2. 接收中断
3. DMA

在查询方式中，软件可以查询 RXRDY 位，一旦该位被置高，就从 LINRD0 寄存器的 RD0 字节读取数据。选择查询方式会给 CPU 带来不必要的负担。为避免这种情况，您可以使用中断或 DMA 方式。要使用中断方式，需设置 SET RX INT 位。要使用 DMA 方式，必须设置 SET RX DMA 位。RXRDY 位一被置位，就会立即产生中断请求或 DMA 请求。如果通过将比较校验和（CC）位置 1 来启用校验和方案，会对当前正在接收的字节（预期为校验和字节）进行校验和比较。一旦接收到校验和，CC 位就会被清除。如果存在校验和错误，会立即标记校验和错误（CE）。

##### 22.3.2.5.1.2 多缓冲模式接收数据

当 MBUF 模式位设置为 1 时，将选择多缓冲区模式。在此模式下，LIN 在接收缓冲区中接收到编程数量的数据和校验和字段（即完整帧）后，会设置 RXRDY 位。

错误条件检测逻辑与单缓冲区模式类似，不同之处在于该逻辑会监控完整帧。与单缓冲区模式一样，可以使用轮询、DMA 或中断方法读取数据。必须根据字节数从 LINRD0 和 LINRD1 寄存器读取接收的数据。当长度小于或等于 4 时，读取 LINRD0 寄存器会清除 RXRDY 标志。当长度大于 4 时，读取 LINRD1 寄存器会清除 RXRDY 标志。如果在数据接收期间通过将比较校验和（CC）位设置为 1 来启用校验和方案，那么在接收到由长度字段指示的编程数据字节数之后接收到的字节将被视为校验和字节。一旦接收到校验和并进行比较，CC 位就会被清除。

#### 22.3.2.5.2 发送数据

当 TX FUNC 位和 TXENA 位均被设置为 1 时，LIN 发送器被启用。如果未设置 TX FUNC 位，LINTX 引脚将作为通用 I/O 引脚而非 LIN 功能引脚。在 TXENA 位被设置为 1 之前写入 TD0 的任何值都不会被发送。这两个控制位使得 LIN 发送器能够独立于接收器保持非活动状态。

在通过 TX 匹配接收到有效的 LIN ID 后，ID TX 标志被置位。如果启用了 ID 中断，则会产生 ID 中断。

##### 22.3.2.5.2.1 单缓冲模式发送数据

当 MBUF 模式位被清零时，将选择单缓冲模式。在此模式下，LIN 会等待数据写入 TD0，然后将数据传输到 SCITXSHF 并进行发送。TXRDY 和 TX EMPTY 位指示发送缓冲区的状态。也就是说，当发送器准备好接收写入 TD0 的数据时，TXRDY 位会被置位。此外，如果 TD0 和 SCITXSHF 都为空，那么 TX EMPTY 位也会被置位。

您可以通过以下方式发送数据：

1. 轮询发送就绪标志
2. 发送中断
3. DMA

在轮询方式中，软件可以在向 TD0 写入数据之前轮询 TXRDY 位是否为高。选择轮询方式会不必要地增加 CPU 的负载。为避免这种情况，您可以使用中断或 DMA 方式。要使用中断方式，需置位 SET TX INT 位。要使用 DMA 方式，需置位 SET TX DMA 位。一旦 TXRDY 位置位，就会立即产生中断或 DMA 请求。当 LIN 完成所有待发帧的传输后，SCITXSHF 寄存器和 TD0 会变为空，TXRDY 位置位，并且如果已启用，会产生中断/DMA 请求。由于所有数据都已发送，可以暂停中断/DMA 请求。这可以通过禁用发送中断（CLR TX INT）/DMA 请求（CLR TX DMA 位）或禁用发送器（清除 TXENA 位）来实现。如果通过将发送校验和（SC）位置 1 启用了校验和方案，则会在当前字节传输完成后发送校验和字节。校验和字节发送完成后，SC 位会被清零。

注：无法通过读取 SCIINTVECT0 或 SCIINTVECT1 寄存器中相应的中断偏移量来清除 TXRDY 标志。

#### 22.3.2.5.2.2 多缓冲模式发送数据

当 MBUF 模式位被设置为 1 时，多缓冲区模式被选中。与单缓冲区模式类似，您可以使用轮询、DMA 或中断方法来写入要传输的数据。必须根据字节数将传输的数据写入 LINTD0 和 LINTD1 寄存器。LIN 会等待数据写入 LINTD0 寄存器的字节 0（TD0），并将设定数量的字节传输到 SCITXSHF，以自动逐个传输。如果通过将发送校验和（SC）位设置为 1 来启用校验和方案，则在校验和会在 LENGTH 字段指示的设定数量数据字节的最后一个字节传输完成后发送。SC 位会在校验和字节传输完成后被清零。

### 22.3.3 低功耗模式

SCI/LIN 模块可置于本地低功耗模式或全局低功耗模式。全局低功耗模式由系统触发，不受 SCI/LIN 模块控制。在全局低功耗模式下，SCI/LIN 的所有时钟都会关闭，因此该模块完全处于非活动状态。如果在接收器正在接收数据时请求进入全

局低功耗模式，SCI/LIN 会完成当前的接收工作，然后再进入低功耗模式，也就是说，只有当忙位（SCIFLR.3）被清零时，模块才会进入低功耗模式。

LIN 模块进入低功耗模式的条件有两个：一是 LINRX 引脚上超过 4 秒没有活动（可能是持续的隐性电平或显性电平）；二是接收到睡眠命令帧。一旦超时标志（SCIFLR.4）被置位或接收到睡眠命令，应用程序软件必须设置掉电位（SCIGCR2.0），使模块进入本地低功耗模式。唤醒信号会终止 LIN 总线的睡眠模式。

注：在接收和发送期间启用本地低功耗模式

如果唤醒中断已启用，且在接收器接收数据时请求低功耗模式，那么 SCI/LIN 会立即产生一个唤醒中断以清除掉电位。因此，SCI/LIN 将被阻止进入低功耗模式，并完成当前的接收操作。否则，如果唤醒中断被禁用，SCI/LIN 会先完成当前的接收操作，然后再进入低功耗模式。

#### 22.3.3.1 进入睡眠模式

在 LIN 协议中，睡眠命令用于向所有节点广播睡眠模式。睡眠命令由一个诊断主请求帧构成，其标识符为 0x3C（60），第一个数据字段为 0x00。一旦所有节点接收到睡眠命令，总线上就不能有任何活动：此时总线处于睡眠模式。

通过设置 POWERDOWN 位可进入本地低功耗模式；设置该位会停止向 SCI/LIN 内部逻辑和寄存器提供时钟。清除 POWERDOWN 位会使 SCI/LIN 退出本地低功耗模式。在本地掉电模式下，所有寄存器均可访问。如果在低功耗模式下访问某个寄存器，这种访问只会为该特定访问启用模块的时钟。

#### 22.3.3.2 唤醒

唤醒中断用于允许 SCI/LIN 模块自动退出低功耗模式。当在接收 RX 引脚检测到低电平时，会触发 SCI/LIN 唤醒，这会清除掉电（POWERDOWN）位。

注：如果唤醒中断被禁用，那么只要有请求，SCI/LIN 就会进入低功耗模式，但接收 RX 引脚上的低电平不会导致 SCI/LIN 退出低功耗模式。

在 LIN 模式下，任何节点都可以通过发送唤醒信号来终止休眠模式，参见图。从节点检测到总线处于休眠模式且有唤醒请求待处理时，会发送唤醒信号。唤醒信号是 LIN 总线上持续 时间的显性值；对于 LIN 总线波特率，这至少为 5 个 。唤醒信号通过发送一个 0xF0 字节生成，该字节包含 5 个显性 和 5 个隐性 。

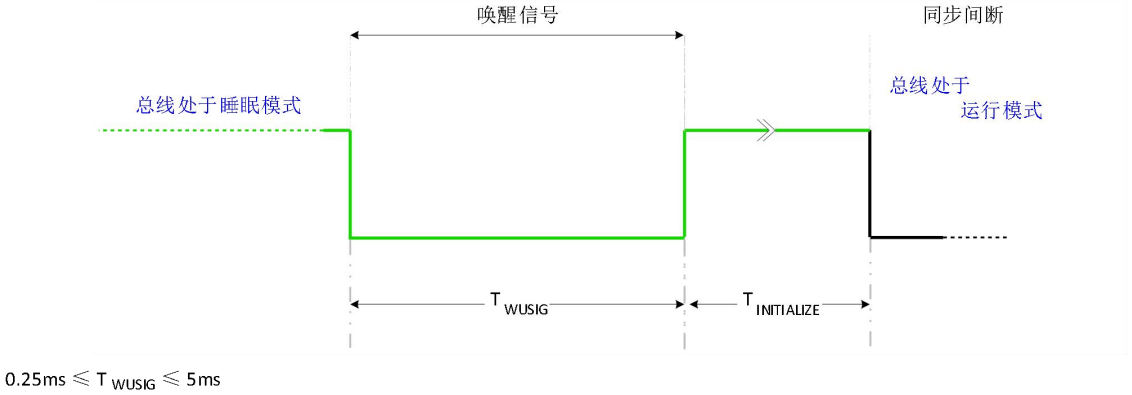



图 22-26 唤醒信号生成

假设总线没有噪声或负载效应，向 TD0 写入 0xF0 会加载发送器，以满足 的唤醒信号时序要求。然后，设置 GENWU 位会发送 TD0 中预加载的值，用于唤醒信号传输。

注：仅当 SWnRST 被设置为 1 且节点处于掉电模式时，GENWU 位才能被设置/复位。该位在检测到有效的同步中断时会被清零。发送唤醒请求的主设备在接收到唤醒脉冲后会退出掉电模式。SWnRST 会将该位清零，这可用于阻止主设备发送更多的唤醒请求。

LIN 收发器在接收到唤醒信号后，会通过 RX 引脚上的显性电平或者向稳压器发送信号的方式，将其转换为用于唤醒微控制器的信号。当 POWERDOWN 位被设置时，如果 LIN 模块在 RX 引脚上检测到从隐性到显性的跳变沿（下降沿），且 SCISETINT 寄存器中启用了唤醒中断，那么 LIN 模块就会产生唤醒中断。

根据 LIN 协议 2.0，LIN 收发器若检测到总线上的显性电平持续时间超过 150 毫秒，会将其识别为唤醒请求。LIN 从设备在检测到从显性到隐性的跳变沿（唤醒信号结束）后，会在不到 100 毫秒（ <100 毫秒）的时间内做好监听总线的准备。

#### 22.3.3.3 唤醒超时

LIN 协议为唤醒序列定义了以下超时规则：

1.向总线发送唤醒信号后，所有节点会等待主节点发送消息头。若在发送唤醒信号后 150ms（20kHz 波特率下为 3000 个周期）内未检测到同步字段，则发起第一次唤醒请求的节点会发送新的唤醒信号。

2.该唤醒序列最多重复两次；在三次尝试唤醒 LIN 总线后，唤醒信号生成功能会暂停 1.5 秒（20kHz 波特率下为 30000 个周期）（即三次中断后的暂停时长）。

注：为实现与 LIN 1.3 超时条件的兼容性，需设置 MBRS（主波特率选择）寄存器，确保 LIN 2.0（基于实时）时序符合 LIN 1.3 的位时间基准。发起唤醒的节点可相应地设置 MBRS 寄存器，使目标时长满足 “128Tbits× 编程预分频器” 的要求。

LIN 模块通过硬件实现，处理 LIN 协议定义的唤醒超时时间。

### 22.3.4 仿真模式

在仿真模式下，CONT 位决定了程序暂停时 SCI/LIN 模块的运行方式。SCI/LIN 的计数器在调试模式下会受到该位的影响：当 CONT 位置 1 时，计数器不会停止；当 CONT 位清零时，计数器在调试模式下会停止。

在仿真模式下，对任何 SCI/LIN 寄存器的读取操作，都不会对 SCIFLR 寄存器中的标志位产生任何影响。

注：若在帧的发送或接收过程中进入仿真模式，且未设置 CONT 位，则通信大概率无法正常进行。建议在仿真模式下设置 CONT 位，以确保通信正常。

## 22.4 寄存器描述

SCI/LIN 模块寄存器基于 SCI 寄存器，并增加了由 SCIGCR1 寄存器中的 LIN MODE 位启用的功能寄存器。

这些寄存器支持 32 位读取或写入。通过以下部分列出的寄存器对 SCI/LIN 进行控制和访问。可编程的功能包括 LIN 协议模式、通信和时序模式、波特率值、帧格式、DMA 请求和中断配置。

### 22.4.1 LIN 基地址表

表 22-11 LIN 基地址表

| 位 字段 名称 | DriverLib 名称 | 基地址 | DMA | HIC | 流水线保护 | 实例 |
|---|---|---|---|---|---|---|
| LinaRegs | LIN_REGS | LINA_BASE | 0x0068_0000 | YES | YES | YES |
| LinbRegs | LIN_REGS | LINB_BASE | 0x0068_0200 | YES | YES | YES |

### 22.4.2 LIN_REGS 寄存器

表中列出了 LIN_REGS 寄存器的内存映射寄存器。表中未列出的所有寄存器偏移地址应视为保留位置，不应修改寄存器内容。

表 22-12 LIN_REGS 寄存器

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0x0 | SCIGCR0 | 全局控制寄存器 0 |
| 0x8 | SCIGCR1 | 全局控制寄存器 1 |
| 0x10 | SCIGCR2 | 全局控制寄存器 2 |
| 0x18 | SCISETINT | 中断使能寄存器 |
| 0x20 | SCICLEARINT | 中断禁止寄存器 |
| 0x28 | SCISETINTLVL | 置位中断级别寄存器 |
| 0x30 | SCICLEARINTLVL | 清零中断级别寄存器 |
| 0x38 | SCIFLR | 标志寄存器 |
| 0x40 | SCIINTVECT0 | 中断向量偏移寄存器 0 |
| 0x48 | SCIINTVECT1 | 中断向量偏移寄存器 1 |
| 0x50 | SCIFORMAT | 长度控制寄存器 |
| 0x58 | BRSR | 波特率选择寄存器 |
| 0x60 | SCIED | 仿真缓冲寄存器 |
| 0x68 | SCIRD | 接收数据缓冲寄存器 |
| 0x70 | SCITD | 发送数据缓冲寄存器 |
| 0x78 | SCIPIO0 | 引脚控制寄存器 0 |
| 0x88 | SCIPIO2 | 引脚控制寄存器 2 |
| 0xC0 | LINCOMP | 比较寄存器 |
| 0xC8 | LINRD0 | 接收数据寄存器 0 |
| 0xD0 | LINRD1 | 接收数据寄存器 1 |
| 0xD8 | LINMASK | 接收掩码寄存器 |
| 0xE0 | LINID | LIN ID 寄存器 |
| 0xE8 | LINTD0 | 发送数据寄存器 0 |
| 0xF0 | LINTD1 | 发送数据寄存器 1 |
| 0xF8 | MBRSR | 最大波特率选择寄存器 |
| 0x120 | IODFTCTRL | 用于 LIN 的 IODFT |

| 偏移 | 缩写 | 寄存器名称 |
|---|---|---|
| 0x1C0 | LIN_GLB_INT_EN | LIN 全局中断使能寄存器 |
| 0x1C8 | LIN_GLB_INT_FLG | LIN 全局中断标志寄存器 |
| 0x1D0 | LIN_GLB_INT_CLR | LIN 全局中断清除寄存器 |

复杂的位访问类型经过编码以适应小表格单元格。表中显示了本节中使用的访问类型代码。

表 22-13 LIN_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | R | R |
| 写入类型 | W | W |
| W1C | W | 1C |
| W1S | W | 1S |
| 复位或默认值 | -n | 复位后的值或默认值 |
| 寄存器数组变量 | i,j,k,l,m,n | 当这些变量用在寄存器名称、偏移或地址中时，它们指的是寄存器数组的值，其中该寄存器是重复寄存器组的一部分。寄存器组形成层次结构，数组用公式表示。 |
| y | 当此变量用在寄存器名称、偏移或地址中时，它指的是寄存器数组的值。 | |

### 22.4.2.1 SCIGCR0 寄存器

偏移地址：0h　复位值：00000000h

SCIGCR0 寄存器定义模块复位。

表 22-14 SCIGCR0 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | RESET |
| R-0h | | | | | | | R/W-0h |

表 22-15 SCIGCR0 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-1 | RESERVED | R | 0h | 保留 |
| 0 | RESET | R/W | 0h | 此位复位 SCI/LIN 模块。此位在 LIN 或 SCI 兼容模式下有效。此位影响 SCI/LIN 模块的复位状态。<br>复位类型：SYSRSn<br>0h (R/W) = SCI/LIN 模块保持复位状态。<br>1h (R/W) = SCI/LIN 模块退出复位状态。 |

### 22.4.2.2 SCIGCR1 寄存器

偏移地址：8h　复位值：00000000h

SCIGCR1 寄存器定义 SCI 使用的帧格式、协议和通信模式。

表 22-16 SCIGCR1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | TXENA | RXENA |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | CONT | LOOPBACK |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | STOPEXTFRAME | HGENCTRL | CTYPE | MBUFMODE | ADAPT | SLEEP |
| R-0h | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SWnRST | LINMODE | CLK_Master | STOP | PARITY | PARITYENA | TIMINGMODE | COMMMODE |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 22-17 SCIGCR1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-26 | RESERVED | R | 0h | 保留 |
| 25 | TXENA | R/W | 0h | 发送使能。此位在 LIN 和 SCI 模式下有效。仅当 TXENA 位置位时，数据才会从 SCITD 或 LIN 模式下的 TDy（y=0, 1,…7）缓冲区传输到 SCITXSHF 移出寄存器。注意：在 TXENA 置位之前写入 SCITD 或发送多缓冲区的数据不会被发送。如果在传输进行时清除 TXENA，则之前写入 SCITD 的数据将被发送（包括 LIN 模式下的校验和字节）。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 复位类型：SYSRSn<br>0h (R/W) = 禁止从 SCITD 或 TDy 传输到 SCITXSHF<br>1h (R/W) = 允许从 SCITD 或 TDy 传输数据到 SCITXSHF |
| 24 | RXENA | R/W | 0h | 接收使能。此位在 LIN 或 SCI 兼容模式下有效。RXENA 允许或阻止数据从 SCIRXSHF 传输到 SCIRD 或接收多缓冲区。注意：清除 RXENA 会阻止接收字符传输到接收缓冲区或多缓冲区，阻止 RX 状态标志（参见表中）被接收数据更新，并禁止接收和错误中断。但是，无论 RXENA 状态如何，移位寄存器都会继续汇编数据。注意：如果在帧接收完成之前清除 RXENA，则该帧的数据不会传输到接收缓冲区。注意：如果在帧接收完成之前设置 RXENA，则该帧的数据将传输到接收缓冲区。如果在 SCIRXSHF 汇编帧的过程中设置 RXENA，则该帧的状态标志不保证准确。为确保状态标志正确反映总线上特定帧期间的检测内容，应在检测到该帧之前设置 RXENA。<br>复位类型：SYSRSn<br>0h (R/W) = 阻止接收器将数据从移位缓冲区传输到接收缓冲区或多缓冲区<br>1h (R/W) = 允许接收器将数据从移位缓冲区传输到接收缓冲区或多缓冲区 |
| 23-18 | RESERVED | R | 0h | 保留 |
| 17 | CONT | R/W | 0h | 暂停时继续。此位仅在调试程序时有效，它决定 SCI/LIN 在程序暂停时的工作方式。此位影响 LIN 计数器。当此位置位时，计数器在调试期间不停止。当此位清零时，计数器在调试期间停止。<br>复位类型：SYSRSn<br>0h (R/W) = 当进入调试模式时，SCI/LIN 状态机冻结。发送和 LIN 计数器暂停，并在退出调试模式时恢复。<br>1h (R/W) = 当进入调试模式时，SCI/LIN 继续运行，直到当前发送和接收功能完成。 |
| 16 | LOOPBACK | R/W | 0h | 回环位。此位在 LIN 或 SCI 兼容模式下有效。可以使用此位选择 SCI/LIN 的自检选项。如果 LINTX 和 LINRX 引脚配置为 SCI/LIN 功能，则 LINTX 引脚内部连接到 LINRX 引脚。在外部，回环操作期间，LINTX 引脚输出高电平，LINRX 引脚处于高阻抗状态。如果在 SCI/LIN 正在发送或接收数据时更改此位的值，可能导致错误。<br>复位类型：SYSRSn<br>0h (R/W) = 禁止回环模式。<br>1h (R/W) = 使能回环模式。 |
| 15-14 | RESERVED | R | 0h | 保留 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| 13 | STOPEXTFRAME | R/W | 0h | 停止扩展帧通信。此位仅在 LIN 模式下有效。此位只能在扩展帧通信期间写入。当扩展帧通信停止时，此位自动清除。<br>复位类型：SYSRSn<br>0h (R/W) = 无影响<br>1h (R/W) = 一旦当前帧发送/接收完成，扩展帧通信将停止。 |
| 12 | HGENCTRL | R/W | 0h | HGEN 控制位。此位仅在 LIN 模式下有效。此位控制掩码过滤比较的类型。<br>复位类型：SYSRSn<br>0h (R/W) = 使用 ID-字节进行 ID 过滤。 LINID 寄存器中的 RECEIVEDID 和 IDBYTE 字段用于检测匹配（使用 TX/RXMASK 值）。LINMASK 寄存器中的掩码 0xFF 将导致不匹配。<br>1h (R/W) = 使用 ID-SLAVETask 字节进行 ID 过滤（推荐）。 LINID 寄存器中的 RECEIVEDID 和 IDSLAVETASKBYTE 字段用于检测匹配（使用 TX/RXMASK 值）。LINMASK 寄存器中的掩码 0xFF 将导致始终匹配。 |
| 11 | CTYPE | R/W | 0h | 校验和类型。此位仅在 LIN 模式下有效。此位控制要使用的校验和类型：经典或增强。<br>复位类型：SYSRSn<br>0h (R/W) = 使用经典校验和。该校验和与 LIN 1.3 从节点兼容。经典校验和包含对所有数据字节的溢出进位模-256 和。使用标识符 60 (0x3C) 到 63 (0x3F) 发送的帧必须始终使用经典校验和。<br>1h (R/W) = 使用增强校验和。增强校验和与 LIN 2.0 及更新版本从节点兼容。增强校验和包含对所有数据字节和受保护标识符的溢出进位模-256 和。 |
| 10 | MBUFMODE | R/W | 0h | 多缓冲模式。此位在 LIN 或 SCI 兼容模式下有效。此位控制接收/发送缓冲区的使用，即，是使用 RX/TX 多缓冲区还是使用单个寄存器 RD0/TD0。<br>复位类型：SYSRSn<br>0h (R/W) = 禁止多缓冲模式。<br>1h (R/W) = 使能多缓冲模式。 |
| 9 | ADAPT | R/W | 0h | 自适应模式使能。此模式仅在 LIN 模式下有效。此位在同步字段检测期间起作用。根据节点能力文件定义，可以使用此位启用两种 LIN 协议位速率模式：自动或选择。软件和网络配置将决定使用前两种模式中的哪一种。当此位清除时，应使用 LIN 2.0 协议固定位速率。如果 ADAPT 位置位，检测波特率的 LIN 从节点将其与 BRSR 寄存器中的预分频器进行比较，如果不同则更新。BRSR 寄存器将更新为新值。如果未设置此位，则不会调 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 整 BRSR 寄存器。此字段仅在 LIN 模式下可写。<br>复位类型：SYSRSn<br>0h (R/W) = 禁止自动波特率调整。<br>1h (R/W) = 使能自动波特率调整。 |
| 8 | SLEEP | R/W | 0h | SCI 休眠。仅 SCI 兼容模式。在多处理器配置中，此位控制接收休眠功能。清除此位可使 SCI 退出休眠模式。当 SLEEP 位置位时，接收器仍然工作，但仅当检测到地址帧时才更新 RXRDY 并加载 SCIRD 新数据。其余接收器状态标志会更新，如果设置了相应的中断使能位，无论 SLEEP 位值如何，都会请求错误中断。这样，如果在 SCI 休眠时接收数据线上检测到错误，软件可以立即处理错误状况。当检测到地址字节时，SLEEP 位不会自动清除。此字段仅在 SCI 模式下可写。<br>复位类型：SYSRSn<br>0h (R/W) = 禁止休眠模式。<br>1h (R/W) = 使能休眠模式。 |
| 7 | SWnRST | R/W | 0h | 软件复位（低有效）。此位在 LIN 或 SCI 兼容模式下有效。SCI/LIN 仅应在 SWnRST = 0 时配置。只有以下配置位可以在运行时更改（即，当 SWnRESET = 1 时）：- 停止扩展帧 (SCIGCR1[13]) - CC 位 (SCIGCR2[17]) - SC 位 (SCIGCR2[16])<br>复位类型：SYSRSn<br>0h (R/W) = SCI/LIN 处于复位状态，不会发送或接收数据。向此位写 0 会初始化 SCI/LIN 状态机和操作标志。所有受影响的逻辑保持在复位状态，直到向此位写入 1。<br>1h (R/W) = SCI/LIN 处于就绪状态，可以进行发送和接收。在此位设置为 1 后，模块的配置不应更改。 |
| 6 | LINMODE | R/W | 0h | LIN 模式。此位控制模块的工作模式。<br>复位类型：SYSRSn<br>0h (R/W) = 禁止 LIN 模式，使能 SCI 兼容模式。<br>1h (R/W) = 使能 LIN 模式，禁止 SCI 兼容模式。 |
| 5 | CLK_Master | R/W | 0h | SCI 内部时钟使能或 LIN 主/从配置。在 SCI 模式下，此位使能 SCI 模块的时钟。在 LIN 模式下，此位决定 LIN 节点是从节点还是主节点。<br>复位类型：SYSRSn<br>0h (R/W) = SCI 兼容模式：保留。 LIN 模式：模块处于从模式。<br>1h (R/W) = SCI 兼容模式：使能 SCI 模块的时钟。 LIN 模式：节点处于主模式。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 4 | STOP | R/W | 0h | SCI 停止位数。此位仅在 SCI 兼容模式下有效。注意：接收器仅检查一个停止位。然而，在空闲线模式下，接收器等待第二个停止位结束（如果 STOP = 1）才开始检查空闲周期。此字段仅在 SCI 模式下可写。<br>复位类型：SYSRSn<br>0h (R/W) = 使用一位停止位。<br>1h (R/W) = 使用两位停止位。 |
| 3 | PARITY | R/W | 0h | SCI 奇偶校验奇/偶选择。此位仅在 SCI 兼容模式下有效。如果 PARITY ENA 位 (SCIGCR1.2) 置位，PARITY 指定奇校验或偶校验。校验位是根据每帧中的数据位和地址位（在地址位模式下）计算得出。帧中的起始和停止字段不包括在校验计算中。此字段仅在 SCI 模式下可写。复位类型：SYSRSn 0h (R/W) = 使用奇校验。SCI 发送并期望在校验位接收到一个值，使帧中值为 1 的位总数为奇数。1h (R/W) = 使用偶校验。SCI 发送并期望在校验位接收到一个值，使帧中值为 1 的位总数为偶数。 |
| 2 | PARITYENA | R/W | 0h | 奇偶校验使能。使能或禁止奇偶校验功能。复位类型：SYSRSn 0h (R/W) = SCI 兼容模式：禁止奇偶校验，传输时不生成校验位，接收时不期望校验位。 LIN 模式：禁止 ID 校验验证。1h (R/W) = SCI 兼容模式：使能奇偶校验。传输时生成校验位，接收时期望校验位。 LIN 模式：使能 ID 校验验证。 |
| 1 | TIMINGMODE | R/W | 0h | SCI 时序模式位。此位仅在 SCI 兼容模式下有效。使用 SCI 模式时必须将其设置为 1。此位将 SCI 配置为异步操作。复位类型：SYSRSn 0h (R/W) = 保留。1h (R/W) = 当模块配置为 SCI 操作时必须设置为 1。 |
| 0 | COMMMODE | R/W | 0h | SCI/LIN 通信模式位。在兼容模式下，它选择 SCI 通信模式。在 LIN 模式下，它为 ID 字段位 ID4 和 ID5 选择长度控制选项。复位类型：SYSRSn 0h (R/W) = SCI 兼容模式：使用空闲线模式。 LIN 模式：ID4 和 ID5 不用于长度控制。1h (R/W) = SCI 兼容模式：使用地址位模式。 LIN 模式：ID4 和 ID5 用于长度控制。 |

### 22.4.2.3 SCIGCR2 寄存器

偏移地址：10h　复位值：00000000h

SCIGCR2 寄存器用于在扩展帧期间发送或比较校验和字节、生成唤醒以及控制 LIN 模块的低功耗模式。

表 22-18 SCIGCR2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | CC | SC |
| R-0h | | | | | | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | GENWU |
| R-0h | | | | | | | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | POWERDOWN |
| R-0h | | | | | | | R/W-0h |

表 22-19 SCIGCR2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-18 | RESERVED | R | 0h | 保留 |
| 17 | CC | R/W | 0h | 比较校验和。此模式仅在 LIN 模式下有效。此位由接收器用于扩展帧以触发校验和比较。用户将通过向此位写 1 来启动此事务。在非多缓冲模式下，一旦 CC 位置位，将对当前正在接收的字节进行校验和比较，预期该字节为校验字节。在多缓冲模式下，与 CC 位相关的场景如下：- 如果在数据接收期间设置 CC 位，则在接收到由 SCIFORMAT[18:16] 指示的已编程数据字节数后接收的字节将被视为校验和字节。- 如果在 IDLE 期间（即帧间空间）设置 CC 位，则下一个即时字节将被视为校验和字节。如果存在校验和错误，将立即标记 CE。此位在校验和成功比较后自动清除。复位类型：SYSRSn 0h (R/W) = 无影响 1h (R/W) = 比较预期校验字节的校验和 |
| 16 | SC | R/W | 0h | 发送校验和。此模式仅在 LIN 模式下有效。此位由发送器用于扩展帧以发送校验字节。在非多缓冲模式下，校验字节将在当前字节传输之后发送。在多缓冲模式下，校验字节将在最后一个字节计数之后发送（由 SCIFORMAT[18:16] 指示）。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 不发送校验字节。1h (R/W) = 将发送校验字节。校验字节发送后，此位将自动清除。如果在发送第一个字节之前（即在帧间空间期间）设置此位，则不会发送校验和。 |
| 15-9 | RESERVED | R | 0h | 保留 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 8 | GENWU | R/W | 0h | 生成唤醒信号。此位通过发送 TDO 缓冲区值来控制唤醒信号的生成。此位在接收到有效同步间隔时清除。复位类型：SYSRSn 0h (R/W) = 无影响 1h (R/W) = 发送 TDO 用于唤醒。此位将在 SWnRST (SCIGCR1.7) 时清除。 |
| 7-1 | RESERVED | R | 0h | 保留 |
| 0 | POWERDOWN | R/W | 0h | 掉电。此位在 LIN 或 SCI 兼容模式下有效。当掉电位设置时，SCI/LIN 模块试图进入本地低功耗模式。如果当接收器正在主动接收数据并且唤醒中断被禁止时设置 POWERDOWN 位，则 SCI/LIN 将延迟进入低功耗模式，直到接收完成。在 LIN 模式下，用户可以在接收到休眠命令或检测到总线空闲（超过 4 秒，即 20kHz 时 80,000 个周期）时设置 POWERDOWN 位。复位类型：SYSRSn 0h (R/W) = 正常操作 1h (R/W) = 请求本地低功耗模式 |

### 22.4.2.4 SCISETINT 寄存器

偏移地址：18h　复位值：00000000h

SCISETINT 寄存器用于使能 LIN 模块中可用的各种中断。

表 22-20 SCISETINT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SETBEINT | SETPBEINT | SETCEINT | SETISFEINT | SETNREINT | SETFEINT | SETOEINT | SETPEINT |
| R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | SET_RX_DMA_ALL | SET_RX_DMA | SET_TX_DMA |
| R-0h | | | | | R/W1S-0h | R/W1S-0h | R/W1S-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | SETIDINT | 保留 | | | SETRXINT | SETTXINT |
| R-0h | | R/W1S-0h | R-0h | | | R/W1S-0h | R/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SETTOA3WUSINT | SETTOAWUSINT | 保留 | SETTIMEOUTINT | 保留 | | SETWAKEUPINT | SETBRKDTINT |
| R/W1S-0h | R/W1S-0h | R-0h | R/W1S-0h | R-0h | | R/W1S-0h | R/W1S-0h |

表 22-21 SCISETINT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | SETBEINT | R/W1S | 0h | 设置位错误中断。此位仅在 LIN 模式下有效。设置此位使能 SCI/LIN 模块在位错误发生时产生中断。此字段仅在 LIN 模式下可写。<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 0h (R/W) = 禁止中断。向此位写 0 无效。<br>1h (R/W) = 使能中断。 |
| 30 | SETPBEINT | R/W1S | 0h | 设置物理总线错误中断。此位仅在 LIN 模式下有效。设置此位使能 SCI/LIN 模块在物理总线错误发生时产生中断。此字段仅在 LIN 模式下可写。<br>复位类型：SYSRSn<br>0h (R/W) = 禁止中断。向此位写 0 无效。<br>1h (R/W) = 使能中断。 |
| 29 | SETCEINT | R/W1S | 0h | 设置校验和错误中断。此位仅在 LIN 模式下有效。设置此位使能 SCI/LIN 模块在校验和错误发生时产生中断。此字段仅在 LIN 模式下可写。<br>复位类型：SYSRSn<br>0h (R/W) = 禁止中断。向此位写 0 无效。<br>1h (R/W) = 使能中断。 |
| 28 | SETISFEINT | R/W1S | 0h | 设置不一致同步字段错误中断。此位仅在 LIN 模式下有效。设置此位使能 SCI/LIN 模块在不一致同步字段错误发生时产生中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 27 | SETNREINT | R/W1S | 0h | 设置无响应错误中断。此位仅在 LIN 模式下有效。设置此位使能 SCI/LIN 模块在无响应错误发生时产生中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 26 | SETFEINT | R/W1S | 0h | 设置帧错误中断。此位在 LIN 或 SCI 兼容模式下有效。设置此位使能 SCI/LIN 模块在帧错误发生时产生中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 25 | SETOEINT | R/W1S | 0h | 设置溢出错误中断。此位在 LIN 或 SCI 兼容模式下有效。设置此位使能 SCI/LIN 模块在溢出错误发生时产生中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 24 | SETPEINT | R/W1S | 0h | 设置奇偶校验中断。此位在 LIN 或 SCI 兼容模式下有效。设置此位使能 SCI/LIN 模块在奇偶校验错误发生时产生中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 23-19 | RESERVED | R | 0h | 保留 |
| 18 | SET_RX_DMA_ALL | R/W1S | 0h | 设置接收器 DMA 用于地址和数据帧。此位在 LIN 或 SCI 兼容模式下有效。要使能地址和数据帧的 RX DMA 请求，必须设置此位。如果清除此位，则为地址帧生成 RX 中断请求，为数据帧生成 DMA 请求。复位类型：SYSRSn 0h (R/W) = 禁止地址帧的接收器 DMA 请求（使能地址帧的 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | RX 中断请求）。向此位写 0 无效。1h (R/W) = 使能地址和数据帧的接收器 DMA 请求。 |
| 17 | SET_RX_DMA | R/W1S | 0h | 设置接收器 DMA。此位在 LIN 或 SCI 兼容模式下有效。要使能接收器的 DMA 请求，必须设置此位。如果清除，则根据 SETRXINT 生成中断请求。复位类型：SYSRSn 0h (R/W) = 禁止接收器 DMA 请求。向此位写 0 无效。1h (R/W) = 使能接收器 DMA 请求。 |
| 16 | SET_TX_DMA | R/W1S | 0h | 设置发送 DMA。此位在 LIN 或 SCI 兼容模式下有效。要使能发送器的 DMA 请求，必须设置此位。如果清除，则根据 SETTXINT 生成中断请求。复位类型：SYSRSn 0h (R/W) = 禁止发送 DMA 请求。向此位写 0 无效。1h (R/W) = 使能发送 DMA 请求。 |
| 15-14 | RESERVED | R | 0h | 保留 |
| 13 | SETIDINT | R/W1S | 0h | 设置标识符中断。此位仅在 LIN 模式下有效。设置此位使能接收到有效匹配标识符后产生中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 12-10 | RESERVED | R | 0h | 保留 |
| 9 | SETRXINT | R/W1S | 0h | 设置接收器中断。设置此位使能 SCI/LIN 在帧完全接收后，数据从 SCIRXSHF 传输到 SCIRD 时产生接收中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 8 | SETTXINT | R/W1S | 0h | 设置发送器中断。设置此位使能 SCI/LIN 在数据从 SCITD 传输到 SCITXSHF 且 TXRDY 位置位时产生发送中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 7 | SETTOA3WUSINT | R/W1S | 0h | 设置 3 次唤醒信号后超时中断。此位仅在 LIN 模式下有效。设置此位使能 SCI/LIN 在发送 3 次唤醒信号后发生超时时产生中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 6 | SETTOAWUSINT | R/W1S | 0h | 设置唤醒信号后超时中断。此位仅在 LIN 模式下有效。设置此位使能 SCI/LIN 在发送一次唤醒信号后发生超时时产生中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 5 | RESERVED | R | 0h | 保留 |
| 4 | SETTIMEOUTINT | R/W1S | 0h | 设置超时中断。此位仅在 LIN 模式下有效。设置此位使能 SCI/LIN 在没有 LIN 总线活动（总线空闲）至少 4 秒时产生中断。此字段仅在 LIN |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 3-2 | RESERVED | R | 0h | 保留 |
| 1 | SETWAKEUPINT | R/W1S | 0h | 设置唤醒中断。此位在 LIN 或 SCI 兼容模式下有效。设置此位使能 SCI/LIN 产生唤醒中断并退出低功耗模式。唤醒中断在唤醒脉冲的下降沿置位。如果使能，当请求本地低功耗模式而接收器忙时，或在低功耗模式期间在 SCIRX 引脚上检测到低电平时，唤醒中断会置位。如果模块不处于掉电模式，则不会在唤醒脉冲时置位唤醒中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |
| 0 | SETBRKDTINT | R/W1S | 0h | 设置间断检测中断。此位仅在 SCI 兼容模式下有效。设置此位使能 SCI/LIN 在 LINRX 引脚上检测到间断条件时产生中断。此字段仅在 SCI 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。 |

### 22.4.2.5 SCICLEARINT 寄存器

偏移地址：20h　复位值：00000000h

SCICLEARINT 寄存器用于禁止已启用的中断，而无需访问 SCISETINT 寄存器。

表 22-22 SCICLEARINT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CLRBEINT | CLRPBEINT | CLRCEINT | CLRISFEINT | CLRNREINT | CLRFEINT | CLROEINT | CLRPEINT |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | 保留 | SETRXDMA | CLRTXDMA |
| R-0h | | | | | R-0h | R/W1C-0h | R/W1C-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | CLRIDINT | 保留 | | | CLRRXINT | CLRTXINT |
| R-0h | | R/W1C-0h | R-0h | | | R/W1C-0h | R/W1C-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLRTOA3WUSINT | CLRTOAWUSINT | 保留 | CLRTIMEOUTINT | 保留 | | CLRWAKEUPINT | CLRBRKDTINT |
| R/W1C-0h | R/W1C-0h | R-0h | R/W1C-0h | R-0h | | R/W1C-0h | R/W1C-0h |

表 22-23 SCICLEARINT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | CLRBEINT | R/W1C | 0h | 清除位错误中断。此位仅在 LIN 模式下有效。设置此位将禁止位错误中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 30 | CLRPBEINT | R/W1C | 0h | 清除物理总线错误中断。此位仅在 LIN 模式下有效。设置此位将禁止物理总线错误中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 29 | CLRCEINT | R/W1C | 0h | 清除校验和错误中断。此位仅在 LIN 模式下有效。设置此位将禁止校验和错误中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 28 | CLRISFEINT | R/W1C | 0h | 清除不一致同步字段错误中断。此位仅在 LIN 模式下有效。设置此位将禁止 ISFE 中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 27 | CLRNREINT | R/W1C | 0h | 清除无响应错误中断。此位仅在 LIN 模式下有效。设置此位将禁止无响应错误中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 26 | CLRFEINT | R/W1C | 0h | 清除帧错误中断。此位在 LIN 或 SCI 兼容模式下有效。设置此位将禁止帧错误中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 25 | CLROEINT | R/W1C | 0h | 清除溢出错误中断。此位在 LIN 或 SCI 兼容模式下有效。设置此位将禁止溢出中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 24 | CLRPEINT | R/W1C | 0h | 清除奇偶校验中断。此位在 LIN 或 SCI 兼容模式下有效。设置此位将禁止奇偶校验错误中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 23-19 | RESERVED | R | 0h | 保留 |
| 18 | RESERVED | R | 0h | 保留 |
| 17 | SETRXDMA | R/W1C | 0h | 清除接收器 DMA。此位在 LIN 或 SCI 兼容模式下有效。设置此位将禁止接收 DMA 请求。复位 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 类型：SYSRSn 0h (R/W) = 禁止接收器 DMA 请求。向此位写 0 无效。1h (R/W) = 使能接收器 DMA 请求。向此位写 1 将禁止 DMA 请求并清除此位。 |
| 16 | CLRTXDMA | R/W1C | 0h | 清除发送 DMA。此位在 LIN 或 SCI 兼容模式下有效。设置此位将禁止发送 DMA 请求。复位类型：SYSRSn 0h (R/W) = 禁止发送 DMA 请求。向此位写 0 无效。1h (R/W) = 使能发送 DMA 请求。向此位写 1 将禁止 DMA 请求并清除此位。 |
| 15-14 | RESERVED | R | 0h | 保留 |
| 13 | CLRIDINT | R/W1C | 0h | 清除标识符中断。此位仅在 LIN 模式下有效。设置此位将禁止 ID 中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 12-10 | RESERVED | R | 0h | 保留 |
| 9 | CLRRXINT | R/W1C | 0h | 清除接收器中断。此位在 LIN 或 SCI 兼容模式下有效。设置此位将禁止接收器中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 8 | CLRTXINT | R/W1C | 0h | 清除发送器中断。此位在 LIN 或 SCI 兼容模式下有效。设置此位将禁止发送器中断。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 7 | CLRTOA3WUSINT | R/W1C | 0h | 清除 3 次唤醒信号后超时中断。此位仅在 LIN 模式下有效。设置此位将禁止 3 次唤醒信号后超时中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 6 | CLRTOAWUSINT | R/W1C | 0h | 清除唤醒信号后超时中断。此位仅在 LIN 模式下有效。设置此位将禁止一次唤醒信号后超时中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 5 | RESERVED | R | 0h | 保留 |
| 4 | CLRTIMEOUTINT | R/W1C | 0h | 清除超时中断。此位仅在 LIN 模式下有效。设置此位将禁止超时（LIN 总线空闲）中断。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 3-2 | RESERVED | R | 0h | 保留 |
| 1 | CLRWAKEUPINT | R/W1C | 0h | 清除唤醒中断。此位在 LIN 或 SCI 兼容模式下有效。设置此位将禁止唤醒中断。复位类型： |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |
| 0 | CLRBRKDTINT | R/W1C | 0h | 清除间断检测中断。此位仅在 SCI 兼容模式下有效。设置此位将禁止间断检测中断。此字段仅在 SCI 模式下可写。复位类型：SYSRSn 0h (R/W) = 禁止中断。向此位写 0 无效。1h (R/W) = 使能中断。向此位写 1 将禁止中断并清除此位。 |

### 22.4.2.6 SCISETINTLVL 寄存器

偏移地址：28h 复位值：00000000h

CISETINTLVL 寄存器用于将单个中断源映射到 INT1 中断线路。

表 22-24 SCISETINTLVL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| SETBEINTLVL | SETPBEINTLVL | SETCEINTLVL | SETISFEINTLVL | SETNREINTLVL | SETFEINTLVL | SETOEINTLVL | SETPEINTLVL |
| R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h | R/W1S-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | 保留 | 保留 | |
| R-0h | | | | | R-0h | R-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | SETIDINTLVL | 保留 | | | SETRXINTOVO | SETTXINTLVL |
| R-0h | | R/W1S-0h | R-0h | | | R/W1S-0h | R/W1S-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SETTOA3WUSINTLVL | SETTOAWUSINTLVL | 保留 | SETTIMEOUTINTLVL | 保留 | | SETWAKEUPINTLVL | SETBRKDTINTLVL |
| R/W1S-0h | R/W1S-0h | R-0h | R/W1S-0h | R-0h | | R/W1S-0h | R/W1S-0h |

表 22-25 SCISETINTLVL 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | SETBEINTLVL | R/W1S | 0h | 设置位错误中断级别。此位仅在 LIN 模式下有效。写入此位将位错误中断级别映射到 INT1 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 30 | SETPBEINTLVL | R/W1S | 0h | 设置物理总线错误中断级别。此位仅在 LIN 模式下有效。写入此位将物理总线错误中断级别映射到 INT1 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 29 | SETCEINTLVL | R/W1S | 0h | 设置校验和错误中断级别。此位仅在 LIN 模式下有效。写入此位将校验和错误中断级别映射到 INT1 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 28 | SETISFEINTLVL | R/W1S | 0h | 设置不一致同步字段错误中断级别。此位仅在 LIN 模式下有效。写入此位将不一致同步字段错误中断级别映射到 INT1 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 27 | SETNREINTLVL | R/W1S | 0h | 设置无响应错误中断级别。此位仅在 LIN 模式下有效。写入此位将无响应错误中断级别映射到 INT1 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 26 | SETFEINTLVL | R/W1S | 0h | 设置帧错误中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将帧错误中断级别映射到 INT1 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 25 | SETOEINTLVL | R/W1S | 0h | 设置溢出错误中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将溢出错误中断级别映射到 INT1 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 24 | SETPEINTLVL | R/W1S | 0h | 设置奇偶校验错误中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将奇偶校验错误中断级别映射到 INT1 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 23-19 | RESERVED | R | 0h | 保留 |
| 18 | RESERVED | R | 0h | 保留 |
| 17-16 | RESERVED | R | 0h | 保留 |
| 15-14 | RESERVED | R | 0h | 保留 |
| 13 | SETIDINTLVL | R/W1S | 0h | 设置 ID 中断级别。此位仅在 LIN 模式下有效。写入此位将 ID 中断级别映射到 INT1 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 12-10 | RESERVED | R | 0h | 保留 |
| 9 | SETRXINTOVO | R/W1S | 0h | 设置接收器中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将接收器中断级别映射到 INT1 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 8 | SETTXINTLVL | R/W1S | 0h | 设置发送器中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将发送器中断级别映射到 INT1 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 7 | SETTOA3WUSINTLVL | R/W1S | 0h | 设置 3 次唤醒信号后超时中断级别。此位仅在 LIN 模式下有效。写入此位将 3 次唤醒信号后超时中断级别映射到 INT1 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 6 | SETTOAWUSINTLVL | R/W1S | 0h | 设置唤醒信号后超时中断级别。此位仅在 LIN 模式下有效。写入此位将唤醒后超时中断级别映射到 INT1 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 5 | RESERVED | R | 0h | 保留 |
| 4 | SETTIMEOUTINTLVL | R/W1S | 0h | 设置超时中断级别。此位仅在 LIN 模式下有效。写入此位将超时中断级别映射到 INT1 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 3-2 | RESERVED | R | 0h | 保留 |
| 1 | SETWAKEUPINTLVL | R/W1S | 0h | 设置唤醒中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将唤醒中断级别映射到 INT1 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |
| 0 | SETBRKDTINTLVL | R/W1S | 0h | 设置间断检测中断级别。此位仅在 SCI 兼容模式下有效。写入此位将间断检测中断级别映射到 INT1 线路。此字段仅在 SCI 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。 |

### 22.4.2.7 SCICLEARINTLVL 寄存器

偏移地址：30h　复位值：00000000h

SCICLEARINTLVL 寄存器用于将单个中断源映射到 INT0 线路。

表 22-26 SCICLEARINTLVL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CLRBEINTLVL | CLRPBEINTLVL | CLRCEINTLVL | CLRISFEINTLVL | CLRNREINTLVL | CLRFEINTLVL | CLROEINTLVL | CLRPEINTLVL |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | 保留 | 保留 | |
| R-0h | | | | | R-0h | R-0h | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | CLRIDINTLVL | 保留 | | | CLRRXINTLVL | CLRTXINTLVL |
| R-0h | | R/W1C-0h | R-0h | | | R/W1C-0h | R/W1C-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLRTOA3WUSINTLVL | CLRTOAWUSINTLVL | 保留 | CLRTIMEOUTINTLVL | 保留 | | CLRWAKEUPINTLVL | CLRBRKDTINTLVL |
| R/W1C-0h | R/W1C-0h | R-0h | R/W1C-0h | R-0h | | R/W1C-0h | R/W1C-0h |

表 22-27 SCICLEARINTLVL 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | CLRBEINTLVL | R/W1C | 0h | 清除位错误中断级别。此位仅在 LIN 模式下有效。写入此位将位错误中断级别映射到 INT0 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 30 | CLRPBEINTLVL | R/W1C | 0h | 清除物理总线错误中断级别。此位仅在 LIN 模式下有效。写入此位将物理总线错误中断级别映射到 INT0 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 29 | CLRCEINTLVL | R/W1C | 0h | 清除校验和错误中断级别。此位仅在 LIN 模式下有效。写入此位将校验和错误中断级别映射到 INT0 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 28 | CLRISFEINTLVL | R/W1C | 0h | 清除不一致同步字段错误中断级别。此位仅在 LIN 模式下有效。写入此位将不一致同步字段错误中断级别映射到 INT0 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 27 | CLRNREINTLVL | R/W1C | 0h | 清除无响应错误中断级别。此位仅在 LIN 模式下有效。写入此位将无响应错误中断级别映射到 INT0 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 26 | CLRFEINTLVL | R/W1C | 0h | 清除帧错误中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将帧错误中断级别映射到 INT0 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 25 | CLROEINTLVL | R/W1C | 0h | 清除溢出错误中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将溢出错误中断级别映射到 INT0 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 24 | CLRPEINTLVL | R/W1C | 0h | 清除奇偶校验错误中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将奇偶校验错误中断级别映射到 INT0 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 23-19 | RESERVED | R | 0h | 保留 |
| 18 | RESERVED | R | 0h | 保留 |
| 17-16 | RESERVED | R | 0h | 保留 |
| 15-14 | RESERVED | R | 0h | 保留 |
| 13 | CLRIDINTLVL | R/W1C | 0h | 清除 ID 中断级别。此位仅在 LIN 模式下有效。写入此位将 ID 中断级别映射到 INT0 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 12-10 | RESERVED | R | 0h | 保留 |
| 9 | CLRRXINTLVL | R/W1C | 0h | 清除接收器中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将接收器中断级别映射到 INT0 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 8 | CLRTXINTLVL | R/W1C | 0h | 清除发送器中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将发送器中断级别映射到 INT0 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 7 | CLRTOA3WUSINTLVL | R/W1C | 0h | 清除 3 次唤醒信号后超时中断级别。此位仅在 LIN 模式下有效。写入此位将 3 次唤醒信号后超时中断级别映射到 INT0 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 6 | CLRTOAWUSINTLVL | R/W1C | 0h | 清除唤醒信号后超时中断级别。此位仅在 LIN 模式下有效。写入此位将唤醒后超时中断级别映射到 INT0 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 5 | RESERVED | R | 0h | 保留 |
| 4 | CLRTIMEOUTINTLVL | R/W1C | 0h | 清除超时中断级别。此位仅在 LIN 模式下有效。写入此位将超时中断级别映射到 INT0 线路。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 3-2 | RESERVED | R | 0h | 保留 |
| 1 | CLRWAKEUPINTLVL | R/W1C | 0h | 清除唤醒中断级别。此位在 LIN 或 SCI 兼容模式下有效。写入此位将唤醒中断级别映射到 INT0 线路。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。向此位写 0 无效。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |
| 0 | CLRBRKDTINTLVL | R/W1C | 0h | 清除间断检测中断级别。此位仅在 SCI 兼容模式下有效。写入此位将间断检测中断级别映射到 INT0 线路。此字段仅在 SCI 模式下可写。复位类型：SYSRSn 0h (R/W) = 中断级别映射到 INT0 线路。1h (R/W) = 中断级别映射到 INT1 线路。向此位写 1 将把中断映射到 INT0 并清除此位。 |

### 22.4.2.8 SCIFLR 寄存器

偏移地址：38h　复位值：00000904h

SCIFLR 寄存器指示 LIN 模块各种中断源的当前状态。

表 22-28 SCIFLR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BE | PBE | CE | ISFE | NRE | FE | OE | PE |
| R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h | R/W1C-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | IDRXFLAG | IDTXFLAG | RXWAKE | TXEMPTY | TXWAKE | RXRDY | TXRDY |
| R-0h | R/W1C-0h | R/W1C-0h | R-0h | R-1h | R/W-0h | R/W1C-0h | R-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TOA3WUS | TOAWUS | 保留 | TIMEOUT | BUSY | IDLE | WAKEUP | BRKDT |
| R/W1C-0h | R/W1C-0h | R-0h | R/W1C-0h | R-0h | R-1h | R/W1C-0h | R/W1C-0h |

表 22-29 SCIFLR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | BE | R/W1C | 0h | 位错误标志。此位仅在 LIN 模式下有效。当发生位错误时，此位置位。这由内部位监视器中的位监视器检测。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 向此位写 1 - 接收新的同步间隔 此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 未检测到位错误。1h (R/W) = 检测到位错误。 |
| 30 | PBE | R/W1C | 0h | 物理总线错误标志。此位仅在 LIN 模式下有效。当发生物理总线错误时，此位置位。这由 TED 中的位监视器检测。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 向此位写 1 - 接收新的同步间隔 注意：仅当无法生成同步间隔（由于总线对 VBAT 短路）或无法生成同步间隔定界符（由于总线对 GND 短路）时，才会标记 PBE。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 未检测到物理总线错误。1h (R/W) = 检测到物理总线错误。 |
| 29 | CE | R/W1C | 0h | 校验和错误标志。此位仅在 LIN 模式下有效。当接收节点检测到校验和错误时，此位置位。要使用的校验和类型取决于 SCIGCR1.CTYPE 位。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 向此位写 1 - 接收新的同步间隔 此字段仅在 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 未检测到校验和错误。1h (R/W) = 检测到校验和错误。 |
| 28 | ISFE | R/W1C | 0h | 不一致同步字段错误标志。此位仅在 LIN 模式下有效。当同步器在报头接收期间检测到不一致的同步字段错误时，此位置位。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 向此位写 1 - 接收新的同步间隔 此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 未检测到不一致同步字段错误。1h (R/W) = 检测到不一致同步字段错误。 |
| 27 | NRE | R/W1C | 0h | 无响应错误标志。此位仅在 LIN 模式下有效。当对在 TFRAME_MAX 内完成的主机报头没有响应时，此位置位。此超时周期适用于长度未知的消息帧（标识符 0 到 61）。此错误由模块的同步器检测。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 向此位写 1 - 接收新的同步间隔 此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 未检测到无响应错误。1h (R/W) = 检测到无响应错误。 |
| 26 | FE | R/W1C | 0h | 帧错误标志。此位在 LIN 或 SCI 兼容模式下有效。当未找到预期的停止位时，此位置位。在 SCI 兼容模式下，仅检查第一个停止位。缺少停止位表示与起始位的同步已丢失，且字符帧不正确。检测到帧错误会导致 SCI 产生错误中断（如果 RXERR INT ENA 位置位）。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 向此位写 1 - 接收新字符（SCI 兼容模式）或帧（LIN 模式）在多缓冲模式下，帧在 SCIFORMAT 寄存器中定义。复位类型：SYSRSn 0h (R/W) = 未检测到帧错误。1h (R/W) = 检测到帧错误。 |
| 25 | OE | R/W1C | 0h | 溢出错误标志。此位在 LIN 或 SCI 兼容模式下有效。当数据从 SCIRXSHF 传输到 SCIRD 覆盖 SCIRD 或 RDy 缓冲区中已有的未读数据时，此位置位。检测到溢出错误会导致 LIN 产生错误中断（如果 SET OE INT 位为 1）。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | - 向此位写 1 复位类型：SYSRSn 0h (R/W) = 未检测到溢出错误。1h (R/W) = 检测到溢出错误。 |
| 24 | PE | R/W1C | 0h | 奇偶校验错误标志。此位在 LIN 或 SCI 兼容模式下有效。当在接收数据中检测到奇偶校验错误时，此位置位。在 SCI 地址位模式下，校验位是根据接收帧的数据和地址位字段计算得出的。在空闲线模式下，仅使用数据来计算校验位。当接收到的字符的 1 的个数与其校验位不匹配时，会产生错误。有关奇偶校验检查的更多信息，请参见“SCI 全局控制寄存器 (SCIGCR1)” 描述。如果奇偶校验功能被禁止（即 SCIGCR1.2 = 0），则 PE 标志被禁止并读为 0。检测到奇偶校验错误会导致 LIN 产生错误中断（如果 SET PE INT 位 = 1）。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 接收新字符（SCI 兼容模式）或帧（LIN 模式）- 向此位写 1 复位类型：SYSRSn 0h (R/W) = 无奇偶校验错误或奇偶校验被禁止。1h (R/W) = 检测到奇偶校验错误。 |
| 23-16 | RESERVED | R | 0h | 保留 |
| 15 | RESERVED | R | 0h | 保留 |
| 14 | IDRXFLAG | R/W1C | 0h | 接收时标识符标志。此位仅在 LIN 模式下有效。当接收到具有 RX 匹配且无 ID 校验错误的标识符时，此标志置位。当此标志置位时，表示在 RX 匹配上收到了新的有效标识符。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 读取 LINID 寄存器 - 向此位写 1 此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 未收到有效 ID。1h (R/W) = 在 RX 匹配上收到了有效的 ID RX，存储在 LINID[23:16] 中。 |
| 13 | IDTXFLAG | R/W1C | 0h | 发送时标识符标志。此位仅在 LIN 模式下有效。当接收到具有 TX 匹配且无 ID 校验错误的标识符时，此标志置位。当此标志置位时，表示在 TX 匹配上收到了新的有效标识符。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - RESET 位 (SCIGCR0.0) - 设置 SWnRESET - 系统复位 - 读取 LINID 寄存器 - 向此位写 1 此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 未收到有效 ID。1h (R/W) = 在 TX 匹配上收到了有效 ID，存储在 LINID[23:16] 中。 |
| 12 | RXWAKE | R | 0h | 接收器唤醒检测标志。此位仅在 SCI 兼容模式下有效。SCI 设置此位以指示 SCIRD 中当前的数据 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | 是地址。此位可通过以下方式清除：- RESET 位 - 设置 SWnRESET 位 (SCIGCR1.7) - 系统复位 - 接收数据帧 此位在 SCI 模式下可写。复位类型：SYSRSn 0h (R/W) = SCIRD 中的数据不是地址。1h (R/W) = SCIRD 中的数据是地址。 |
| 11 | TXEMPTY | R | 1h | 发送器空标志。此标志的值指示发送器缓冲寄存器 (SCITD/TDy) 和移位寄存器 (SCITXSHF) 的内容。在多缓冲模式下，此标志指示 TDx 寄存器和移位寄存器 (SCITXSHF) 的值。在非多缓冲模式下，此标志指示 LINTD0（字节）和移位寄存器 (SCITXSHF) 的值。此位通过以下方式置位：- RESET 位 (SCIGCR0.0) - 设置 SWnRESET 位 (SCIGCR1.7) - 系统复位。注意：此位不会引起中断请求。复位类型：SYSRSn 0h (R/W) = 兼容模式或无多缓冲的 LIN：发送器缓冲器或移位寄存器（或两者）中已加载数据。使用多缓冲模式的 LIN 模式：多缓冲区或移位寄存器（或全部）中已加载数据。1h (R/W) = 兼容模式或无多缓冲的 LIN：发送器缓冲区和移位寄存器都为空。使用多缓冲模式的 LIN 模式：多缓冲区和移位寄存器都为空。 |
| 10 | TXWAKE | R/W | 0h | SCI 发送器唤醒方法选择。此位仅在 SCI 兼容模式下有效。TXWAKE 位控制 SCITD 中的数据是应作为地址帧还是数据帧使用多处理器通信格式发送。在将字节写入 SCITD 之前，此位由软件设置为 1 或 0，并在数据从 SCITD 传输到 SCITXSHF 时或通过系统复位被 SCI 清除。TXWAKE 不会由 SWnRESET 位 (SCIGCR1.7) 清除。复位类型：SYSRSn 0h (R/W) = 地址位模式：要发送的帧将是数据（地址位 = 0）。空闲线模式：要发送的帧将是数据。1h (R/W) = 地址位模式：要发送的帧将是地址（地址位 = 1）。空闲线模式：接下来的要发送的帧将是地址（向此位写 1，然后向 SCITD 写虚拟数据，将导致在下一帧发送之前出现 11 位周期的空闲周期）。 |
| 9 | RXRDY | R/W1C | 0h | 接收器就绪标志。在 SCI 兼容模式下，接收器设置此位以指示 SCIRD 包含新数据并准备好由 CPU 读取。在 LIN 模式下，一旦在多缓冲模式下接收到有效帧，RXRDY 就会置位，有效帧即无错误接收的消息帧。在非多缓冲模式下，RXRDY 对每个接收到的字节置位，并在帧的最后一个字节无错误时置位。当 RXRDY 标志位置位时，如果中断使能位被设置，SCI/LIN 会生成接收中断 (SCISETINT.9)。RXRDY 可通过以下方式清除：- RESET 位 (SCIGCR0.0) - 设置 SWnRESET - 系统复位 - 向此位写 1 - 在 SCI 兼容模式下读取 |

| 位 | 字段 | 类型 | 复位 | 描述 |
| --- | --- | --- | --- | --- |
| | | | | SCIRD - 在 LIN 模式下读取响应的最后一个数据字节 RDy 注意：RXRDY 标志无法通过读取 SCIINTVECT0/1 寄存器中的相应中断偏移来清除。复位类型：SYSRSn 0h (R/W) = SCIRD/RDy 中没有新数据。1h (R/W) = 新数据准备好从 SCIRD 读取。 |
| 8 | TXRDY | R | 1h | 发送器缓冲寄存器就绪标志。当置位时，此位指示发送缓冲寄存器（兼容模式下为 SCITD，MBUF 模式下为 LINTD0、LINTD1）已准备好从 CPU 写入获取另一个字符。在 SCI 兼容模式下，向 SCITD 写入数据会自动清除此位。在 LIN 模式下，一旦字节 0 (TD0) 写入 LINTD0，此位就被清除。当 TX 缓冲区的数据移入 SCITXSHF 寄存器后，此位置位。如果 DMA 使能位被设置，此事件可触发发送 DMA 事件。此位可通过以下方式设置为 1：- RESET 位 (SCIGCR0.0) - 设置 SWnRESET (SCIGCR1.7) - 系统复位 注意：TXRDY 标志无法通过读取 SCIINTVECT0/1 寄存器中的相应中断偏移来清除。注意：可以通过 SCICLEARINT 寄存器禁止相应的中断，或通过 TXENA 位禁止发送器 (SCIGCR1.25=0)，来消除发送中断请求，直到下一系列数据写入发送缓冲区 LINTD0 和 LINTD1 为止。复位类型：SYSRSn 0h (R/W) = 兼容模式：SCITD 已满。LIN 模式：多缓冲区已满。1h (R/W) = 兼容模式：SCITD 已准备好接收下一个字符。LIN 模式：多缓冲区已准备好接收下一个字符。 |
| 7 | TOA3WUS | R/W1C | 0h | 3 次唤醒信号后超时标志。此位仅在 LIN 模式下有效。如果在 3 次唤醒信号后未收到同步间隔且已过去 1.5 秒，则此标志置位。在发出另一轮唤醒信号之前使用此到期时间。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 向此位写 1 此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 3 次唤醒信号后无超时。1h (R/W) = 3 次唤醒信号且 1.5 秒后超时。 |
| 6 | TOAWUS | R/W1C | 0h | 唤醒信号后超时标志。此位仅在 LIN 模式下有效。如果在发送唤醒信号后未收到同步间隔，则此位置位。在发出另一个唤醒信号之前，使用至少 150 ms 的到期时间。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 向此位写 1 此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | (R/W) = 一次唤醒信号后无超时 (150 ms)。1h (R/W) = 一次唤醒信号后超时。 |
| 5 | RESERVED | R | 0h | 保留 |
| 4 | TIMEOUT | R/W1C | 0h | LIN 总线空闲超时标志。此位仅在 LIN 模式下有效。如果至少 4 秒没有 LIN 总线活动，则此位置位。LIN 总线活动指从隐性到显性的跳变。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 向此位写 1 此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 未检测到总线空闲。1h (R/W) = 检测到 LIN 总线空闲。 |
| 3 | BUSY | R | 0h | 总线忙标志。此位在 LIN 模式和 SCI 兼容模式下有效。此位指示接收器是否正在接收帧。一旦接收器检测到起始位的开始，BUSY 位就设置为 1。当帧接收完成时，BUSY 位被清除。如果 SET WAKEUP INT 被设置且在此位置位时请求掉电，则 SCI/LIN 会自动阻止进入低功耗模式并生成唤醒中断。BUSY 位由 SCI 接收器直接控制，但可通过以下方式清除：- 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位。复位类型：SYSRSn 0h (R/W) = 接收器当前未在接收帧。1h (R/W) = 接收器当前正在接收帧。 |
| 2 | IDLE | R | 1h | SCI 接收器处于空闲状态。此位仅在 SCI 兼容模式下有效。当此位置位时，SCI 寻找空闲周期以与比特流重新同步。当此位置位时，接收器不接收任何数据。总线必须空闲 11 个位周期才能清除此位。SCI 在以下情况进入此状态：- 系统复位后 - 设置 SWnRESET 位 (SCIGCR1.7) - 从掉电模式退出后 此位在 SCI 模式下可写。复位类型：SYSRSn 0h (R/W) = 检测到空闲周期，SCI 准备好接收。1h (R/W) = 未检测到空闲周期，SCI 不会接收任何数据。 |
| 1 | WAKEUP | R/W1C | 0h | 唤醒标志。此位仅在 LIN 模式下有效。当接收器或发送器活动使模块退出掉电模式时，SCI/LIN 设置此位。如果 SET WAKEUP INT 位 (SCISETINT.1) 被设置，则生成中断。此位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移。 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 向此位写 1。此字段仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 不从掉电模式唤醒。1h (R/W) = 从掉电模式唤醒。 |
| 0 | BRKDT | R/W1C | 0h | SCI 间断检测标志。此位仅在 SCI 兼容模式下有效。当 SCI 检测到 LINRX 引脚上的间断条件时，此位置位。当 LINRX 引脚在缺少第一个停止 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 位后（即发生帧错误后）连续保持低电平至少 10 位时，发生间断条件。检测到间断条件会导致 SCI 产生错误中断（如果 BRKDT INT ENA 位被设置）。BRKDT 位可通过以下方式清除：- 读取 SCIINTVECT0/1 寄存器中相应的中断偏移。 - 设置 SWnRESET 位 (SCIGCR1.7) - RESET 位 (SCIGCR0.0) - 系统复位 - 向此位写 1。此位在 SCI 模式下可写。复位类型：SYSRSn 0h (R/W) = 未检测到间断条件。1h (R/W) = 检测到间断条件。 |

### 22.4.2.9 SCIINTVECT0 寄存器

偏移地址：40h　复位值：00000000h

SCIINTVECT0 寄存器指示 INT0 中断线路的偏移量。

表 22-30 SCIINTVECT0 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | INTVECT0 | | | | |
| R-0h | | | R-0h | | | | |

表 22-31 SCIINTVECT0 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-5 | RESERVED | R | 0h | 保留 |
| 4-0 | INTVECT0 | R | 0h | INT0 的中断向量偏移。此寄存器指示中断线路 INT0 的偏移。读取此寄存器会将其值更新为 SCIFLR 中下一个最高优先级的待处理中断，并清除与读取的偏移相对应的标志。注意：接收 (SCIFLR.9) 和发送 (SCIFLR.8) 中断的标志无法通过读取此寄存器中相应的偏移向量来清除（详见 SCIFLR 寄存器描述）。复位类型：SYSRSn |

### 22.4.2.10 SCIINTVECT1 寄存器

偏移地址：48h　复位值：00000000h

SCIINTVECT1 寄存器指示 INT1 中断线路的偏移量。

表 22-32 SCIINTVECT1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | INTVECT1 | | | | |
| R-0h | | | R-0h | | | | |

表 22-33 SCIINTVECT1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-5 | RESERVED | R | 0h | 保留 |
| 4-0 | INTVECT1 | R | 0h | INT1 的中断向量偏移。此寄存器指示中断线路 INT1 的偏移。读取此寄存器会将其值更新为 SCIFLR 中下一个最高优先级的待处理中断，并清除与读取的偏移相对应的标志。注意：接收 (SCIFLR.9) 和发送 (SCIFLR.8) 中断的标志无法通过读取此寄存器中相应的偏移向量来清除（详见 SCIFLR 寄存器描述）。复位类型：SYSRSn |

### 22.4.2.11 SCIFORMAT 寄存器

偏移地址：50h　复位值：00000000h

SCIFORMAT 寄存器用于设置字符和帧长度。

表 22-34 SCIFORMAT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | LENGTH | | |
| R-0h | | | | | R/W-0h | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | CHAR | | |
| R-0h | | | | | R/W-0h | | |

表 22-35 SCIFORMAT 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-19 | RESERVED | R | 0h | 保留 |
| 18-16 | LENGTH | R/W | 0h | 帧长度控制位。在 LIN 模式下，这些位指示响应字段中的字节数，从 1 到 8 字节。在缓冲 SCI 模式下，这些位指示字符数。当这些位用于指示 LIN 响应长度 (SCIGCR1[0] = 1) 时，当存在 ID RX 匹配时，此值应更新为响应的预期长度。在缓冲 SCI 模式下，这些位指示字符数，每个字符位数由 SCIFORMAT[2:0] 定义。即，这些位指示发送器/接收器格式的字符数：1 到 8。最多可以有八个字符，每个字符八位。复位类型：SYSRSn 0h (R/W) = 响应字段有 1 个字节/字符。1h (R/W) = 响应字段有 2 个字节/字符。2h (R/W) = 响应字段有 3 个字节/字符。3h (R/W) = 响应字段有 4 个字节/字符。4h (R/W) = 响应字段有 5 个字节/字符。5h (R/W) = 响应字段有 6 个字节/字符。6h (R/W) = 响应字段有 7 个字节/字符。7h (R/W) = 响应字段有 8 个字节/字符。 |
| 15-3 | RESERVED | R | 0h | 保留 |
| 2-0 | CHAR | R/W | 0h | 字符长度控制位。这些位仅在 SCI 兼容模式下有效。这些位设置 SCI 字符长度，从 1 到 8 位。注意：在兼容模式或缓冲 SCI 模式下，当接收长度小于 8 位的数据时，它在 SCIRD/RDy 中左对齐，末尾补零。从 SCIRD 读取的数据应由软件移位，以使接收数据右对齐。注意：写入 SCITD 的数据应右对齐，但不需要前导零填充。这些位仅在 SCI 模式下可写。复位类型：SYSRSn 0h (R/W) = 字符为 1 位长。1h (R/W) = 字符为 2 位长。2h (R/W) = 字符为 3 位长。3h (R/W) = 字符为 4 位 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 长。4h (R/W) = 字符为 5 位长。5h (R/W) = 字符为 6 位长。6h (R/W) = 字符为 7 位长。7h (R/W) = 字符为 8 位长。 |

### 22.4.2.12 BRSR 寄存器

偏移地址：58h　复位值：00000000h

BRSR 寄存器用于配置 LIN 模块的波特率。

表 22-36 BRSR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | U | | | M | | | |
| R-0h | R/W-0h | | | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SCI_LIN_PSH | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SCI_LIN_PSL | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SCI_LIN_PSL | | | | | | | |
| R/W-0h | | | | | | | |

表 22-37 BRSR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | RESERVED | R | 0h | 保留 |
| 30-28 | U | R/W | 0h | 超小数分频器选择 (U)。这些位是波特率规范的一个额外小数部分。这些位允许对 M 小数值每个值提供额外 7 个中间值，从而对小数波特率进行超精细调谐。复位类型：SYSRSn |
| 27-24 | M | R/W | 0h | SCI/LIN 4 位小数分频器选择 (M)。这些位在 LIN 或 SCI 异步模式下有效。这些位用于为 SCI/LIN 模块选择波特率，是波特率规范的小数部分。M 分频器允许对 P 整数预分频器的每个值提供 15 个额外的中间值，从而对波特率进行微调。复位类型：SYSRSn |
| 23-16 | SCI_LIN_PSH | R/W | 0h | 预分频器 P（高位）。SCI/LIN 24 位整数预分频器选择。这些位用于为 SCI/LIN 模块选择波特率。这些位在 LIN 模式和 SCI 兼容模式下有效。SCI/LIN 有一个内部生成的串行时钟，由 LIN 模块输入时钟和该寄存器中的预分频器 P 和 M 决定。SCI/LIN 使用 24 位整数预分频器 P 值来选择超过 16,700,000 种可用波特率中的一种。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 额外的 4 位小数预分频器 M 可进一步优化波特率选择。复位类型：SYSRSn |
| 15-0 | SCI_LIN_PSL | R/W | 0h | 预分频器 P（低位）。SCI/LIN 24 位整数预分频器选择。这些位用于为 SCI/LIN 模块选择波特率。这些位在 LIN 模式和 SCI 兼容模式下有效。SCI/LIN 有一个内部生成的串行时钟，由 LIN 模块输入时钟和该寄存器中的预分频器 P 和 M 决定。SCI/LIN 使用 24 位整数预分频器 P 值来选择超过 16,700,000 种可用波特率中的一种。额外的 4 位小数预分频器 M 可进一步优化波特率选择。复位类型：SYSRSn |

### 22.4.2.13 SCIED 寄存器

偏移地址：60h　复位值：00000000h

SCIED 寄存器是 SCIRD 寄存器的副本，读取它不会影响 RXRDY 标志，供仿真器使用。

表 22-38 SCIED 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ED | | | | | | | |
| R-0h | | | | | | | |

表 22-39 SCIED 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | 保留 |
| 7-0 | ED | R | 0h | 接收器仿真数据。此位仅在 SCI 兼容模式下有效。读取 SCIED(7-0) 不会清除 RXRDY 标志。此寄存器应仅由必须持续读取数据缓冲区而不影响 RXRDY 标志的仿真器使用。复位类型：SYSRSn |

### 22.4.2.14 SCIRD 寄存器

偏移地址：68h　复位值：00000000h

SCIRD 寄存器用于存储和读取接收到的数据。

表 22-40 SCIRD 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RD | | | | | | | |
| R-0h | | | | | | | |

表 22-41 SCIRD 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | 保留 |
| 7-0 | RD | R | 0h | 接收数据。此位仅在 SCI 兼容模式下有效。当帧完全接收后，帧中的数据从接收器移位寄存器 SCIRXSHF 传输到此寄存器。当此传输发生时，RXRDY 标志被设置，如果 RX INT ENA (SCISETINT0.9) 被设置，则生成接收中断。当从 SCIRD 读取数据时，RXRDY 标志自动清除。当 SCI 接收长度少于 8 位的数据时，它将该数据以左对齐格式加载到此寄存器，末尾补零。因此，您的软件应对数据执行正确位数的逻辑移位，使其右对齐。复位类型：SYSRSn |

### 22.4.2.15 SCITD 寄存器

偏移地址：70h　复位值：00000000h

SCITD 寄存器是应用软件写入要发送数据的地方。

表 22-42 SCITD 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TD | | | | | | | |
| R/W-0h | | | | | | | |

表 22-43 SCITD 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | 保留 |
| 7-0 | TD | R/W | 0h | 发送数据。此位仅在 SCI 兼容模式下有效。要发送的数据写入此寄存器。数据从此寄存器传输到发送移位寄存器 SCITXSHF 会设置 TXRDY 标志(SCIFLR.8)，表示 SCITD 已准备好加载另一个字节的数据。注意：如果 TX INT ENA (SCISETINT.8) 被设置，此数据传输也会引起中断。注意：写入 SCITD 寄存器且长度少于 8 位的数据必须右对齐，但不需要前导零填充。复位类型：SYSRSn |

### 22.4.2.16 SCIPIO0 寄存器

偏移地址：78h　复位值：00000000h

SCIPIO0 寄存器用于使能 LINTX 和 LINRX 引脚。

表 22-44 SCIPIO0 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | TXFUNC | RXFUNC | 保留 |
| R-0h | | | | | R/W-0h | R/W-0h | R-0h |

表 22-45 SCIPIO0 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-3 | RESERVED | R | 0h | 保留 |
| 2 | TXFUNC | R/W | 0h | 发送引脚功能。此位在 LIN 或 SCI 模式下有效。此位定义 LINTX 引脚的功能。复位类型：SYSRSn 0h (R/W) = LINTX 引脚被禁止。1h (R/W) = LINTX 引脚被使能。 |
| 1 | RXFUNC | R/W | 0h | 接收引脚功能。此位在 LIN 或 SCI 模式下有效。此位定义 LINRX 引脚的功能。复位类型：SYSRSn 0h (R/W) = LINRX 引脚被禁止。1h (R/W) = LINRX 引脚被使能。 |
| 0 | RESERVED | R | 0h | 保留 |

## 22.4.2.17 SCIPIO2 寄存器

偏移地址：88h　复位值：00000000h

SCIPIO2 寄存器指示 LINTX 和 LINRX 引脚的当前状态。

表 22-46 SCIPIO2 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | TXIN | RXIN | 保留 |
| R-0h | | | | | R-0h | R-0h | R-0h |

表 22-47 SCIPIO2 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-3 | RESERVED | R | 0h | 保留 |
| 2 | TXIN | R | 0h | 发送数据输入。此位在 LIN 或 SCI 兼容模式下有效。此位包含 LINTX 引脚的当前值。复位类型：SYSRSn |
| 1 | RXIN | R | 0h | 接收数据输入。此位在 LIN 或 SCI 兼容模式下有效。此位包含 LINRX 引脚的当前值。复位类型：SYSRSn |
| 0 | RESERVED | R | 0h | 保留 |

### 22.4.2.18 LINCOMP 寄存器

偏移地址：C0h　复位值：00000000h

LINCOMPARE 寄存器用于配置同步定界符和同步间隔扩展。

表 22-48 LINCOMP 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | SDEL | |
| R-0h | | | | | | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | SBREAK | | |
| R-0h | | | | | R/W-0h | | |

表 22-49 LINCOMP 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-16 | RESERVED | R | 0h | 保留 |
| 15-10 | RESERVED | R | 0h | 保留 |
| 9-8 | SDEL | R/W | 0h | 2 位同步定界符比较。这些位仅在 LIN 模式下有效。这些位用于配置同步字段中同步定界符的 Tbit 数。同步定界符的时间延迟计算为：TSDEL = (SDEL + 1)Tbit 这些位仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 同步定界符有 1 个 Tbit。1h (R/W) = 同步定界符有 2 个 Tbit。2h (R/W) = 同步定界符有 3 个 Tbit。3h (R/W) = 同步定界符有 4 个 Tbit。 |
| 7-3 | RESERVED | R | 0h | 保留 |
| 2-0 | SBREAK | R/W | 0h | 3 位同步间隔扩展。仅 LIN 模式。这些位用于配置同步间隔的 Tbit 数，以将同步字段中最小 13 Tbit 扩展到最大 20 Tbit。同步间隔的时间延迟计算为：TSYNBRK = 13Tbit + SBREAK x Tbit 这些位仅在 LIN 模式下可写。复位类型：SYSRSn 0h (R/W) = 同步间隔无额外 Tbit。1h (R/W) = 同步间隔有 1 个额外 Tbit。2h (R/W) = 同步间隔有 2 个额外 Tbit。3h (R/W) = 同步间隔有 3 个额外 Tbit。4h (R/W) = 同步间隔有 4 个额外 Tbit。5h (R/W) = 同步间隔有 5 个额外 Tbit。6h (R/W) = 同步间隔有 6 个额外 Tbit。7h (R/W) = 同步间隔有 7 个额外 Tbit。 |

### 22.4.2.19 LINRD0 寄存器

偏移地址：C8h　复位值：00000000h

LINRD0 寄存器包含接收到的 LIN 帧数据的低 4 字节。

表 22-50 LINRD0 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RD0 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RD1 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RD2 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RD3 | | | | | | | |
| R-0h | | | | | | | |

表 22-51 LINRD0 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | RD0 | R | 0h | 8 位接收缓冲区 0。在 SCIRXSHFT 寄存器中接收的每个响应数据字节，根据接收到的字节数传输到相应的 RDy 寄存器。读取此字节会清除 RXDY 字节。注意：RD 等同于 LIN 帧的数据字节 。复位类型：SYSRSn |
| 23-16 | RD1 | R | 0h | 8 位接收缓冲区 1。在 SCIRXSHFT 寄存器中接收的每个响应数据字节，根据接收到的字节数传输到相应的 RDy 寄存器。复位类型：SYSRSn |
| 15-8 | RD2 | R | 0h | 8 位接收缓冲区 2。在 SCIRXSHFT 寄存器中接收的每个响应数据字节，根据接收到的字节数传输到相应的 RDy 寄存器。复位类型：SYSRSn |
| 7-0 | RD3 | R | 0h | 8 位接收缓冲区 3。在 SCIRXSHFT 寄存器中接收的每个响应数据字节，根据接收到的字节数传输到相应的 RDy 寄存器。复位类型：SYSRSn |

### 22.4.2.20 LINRD1 寄存器

偏移地址：D0h　复位值：00000000h

LINRD1 寄存器包含接收到的 LIN 帧数据的高 4 字节。

LINRD1 寄存器字段描述

表 22-52 LINRD1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RD4 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RD5 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RD6 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RD7 | | | | | | | |
| R-0h | | | | | | | |

表 22-53 LINRD1 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | RD4 | R | 0h | 8 位接收缓冲区 4。在 SCIRXSHFT 寄存器中接收的每个响应数据字节，根据接收到的字节数传输到相应的 RDy 寄存器。复位类型：SYSRSn |
| 23-16 | RD5 | R | 0h | 8 位接收缓冲区 5。在 SCIRXSHFT 寄存器中接收的每个响应数据字节，根据接收到的字节数传输到相应的 RDy 寄存器。复位类型：SYSRSn |
| 15-8 | RD6 | R | 0h | 8 位接收缓冲区 6。在 SCIRXSHFT 寄存器中接收的每个响应数据字节，根据接收到的字节数传输到相应的 RDy 寄存器。复位类型：SYSRSn |
| 7-0 | RD7 | R | 0h | 8 位接收缓冲区 7。在 SCIRXSHFT 寄存器中接收的每个响应数据字节，根据接收到的字节数传输到相应的 RDy 寄存器。复位类型：SYSRSn |

### 22.4.2.21 LINMASK 寄存器

偏移地址：D8h　复位值：00000000h

LINMASK 寄存器用于配置用于过滤接收和发送帧的传入 ID 消息的掩码。

表 22-54 LINMASK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RXIDMASK | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXIDMASK | | | | | | | |
| R/W-0h | | | | | | | |

表 22-55 LINMASK 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | 保留 |
| 23-16 | RXIDMASK | R/W | 0h | 接收ID掩码。<br>该字段仅在LIN模式下有效。这个8位掩码用于对接收到的ID报文进行过滤，并与ID字节做比对。当接收ID与接收ID掩码比对匹配时，会置位ID接收标志；若已使能，则触发ID中断。掩码中的0代表该位需要和ID字节进行比对。掩码中的1代表该位被屏蔽过滤，不参与比对。当SCIGCR1.HGENCTRL置1时，如果需要比对完整ID，该字段必须设置为0xFF。<br>复位类型：SYSRSn |
| 15-8 | RESERVED | R | 0h | 保留 |
| 7-0 | TXIDMASK | R/W | 0h | 发送传输ID掩码。<br>该字段仅在LIN模式下有效。这个8位掩码用于对接收到的ID报文进行过滤，并与ID字节做比较。当接收ID与发送ID掩码比对匹配时，将置位ID发送标志；若已使能，则触发ID中断。掩码中的0代表该位需要和ID字节进行比对；掩码中的1代表该位被屏蔽，不参与比对。<br>当HGENCTRL置1时，如果需要对完整ID进行比对，则该字段必须设置为0xFF。<br>复位类型：SYSRSn |

## 22.4.2.22 LINID 寄存器

偏移地址：E0h　复位值：00000000h

LINID 寄存器包含 LIN 通信的标识字段。

注意：为确保与未来 LIN 模块的软件兼容性，必须将 HGEN CTRL 位设置为 1，将 RX ID MASK 字段设置为 FFh，将 TX ID MASK 字段设置为 FFh。

表 22-56 LINID 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RECEIVEDID | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| IDSLAVETASKBYTE | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IDBYTE | | | | | | | |
| R/W-0h | | | | | | | |

表 22-57 LINID 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | 保留 |
| 23-16 | RECEIVEDID | R | 0h | 接收到的 ID。此位仅在 LIN 模式下有效。此字节包含当前消息标识符。在报头接收期间，如果没有 ID 校验错误且存在 RX/TX 匹配，则接收到的 ID 从 SCIRXSHF 寄存器复制到此字节。注意：如果在 ID 接收期间检测到帧错误 (FE)，接收到的 ID 也不会复制到 LINID 寄存器。复位类型：SYSRSn |
| 15-8 | IDSLAVETASKBYTE | R/W | 0h | ID 从任务字节。此字段仅在 LIN 模式下有效。此字节包含标识符，传入报头的接收 ID 将与该标识符进行比较，以决定 LIN 节点是需要进行 RX 响应、TX 响应还是无需操作。这些位仅在 LIN 模式下可写。复位类型：SYSRSn |
| 7-0 | IDBYTE | R/W | 0h | ID 字节。此字段仅在 LIN 模式下有效。此字节是 LIN 模式消息 ID。在主节点上，CPU 向此寄存器写操作会启动报头传输。对于从任务，当 HGENCTRL (SCIGCR1.12) 为“0”时，此字节用于消息过滤。这些位仅在 LIN 模式下可写。复位类型：SYSRSn |

### 22.4.2.23 LINTD0 寄存器

偏移地址：E8h　复位值：00000000h

LINTD0 寄存器包含要发送数据的低 4 字节。

注意：TD 等同于 LIN 帧的数据字节 。

表 22-58 LINTD0 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TD0 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TD1 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TD2 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TD3 | | | | | | | |
| R/W-0h | | | | | | | |

表 22-59 LINTD0 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | TD0 | R/W | 0h | 8 位发送缓冲区 0。要发送的字节 0 写入此寄存器，然后复制到 SCITXSHF 进行发送。一旦字节 0 写入 TDO 缓冲区，发送将被启动。复位类型：SYSRSn |
| 23-16 | TD1 | R/W | 0h | 8 位发送缓冲区 1。要发送的字节 1 写入此寄存器，然后复制到 SCITXSHF 进行发送。复位类型：SYSRSn |
| 15-8 | TD2 | R/W | 0h | 8 位发送缓冲区 2。要发送的字节 2 写入此寄存器，然后复制到 SCITXSHF 进行发送。复位类型：SYSRSn |
| 7-0 | TD3 | R/W | 0h | 8 位发送缓冲区 3。要发送的字节 3 写入此寄存器，然后复制到 SCITXSHF 进行发送。复位类型：SYSRSn |

### 22.4.2.24 LINTD1 寄存器

偏移地址：F0h　复位值：00000000h

LINTD1 寄存器包含要发送数据的高 4 字节。

注意：TD 等同于 LIN 帧的数据字节 。

表 22-60 LINTD1 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TD4 | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TD5 | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TD6 | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TD7 | | | | | | | |
| R/W-0h | | | | | | | |

表 22-61 LINTD1 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | TD4 | R/W | 0h | 8 位发送缓冲区 4。要发送的字节 4 写入此寄存器，然后复制到 SCITXSHF 进行发送。复位类型：SYSRSn |
| 23-16 | TD5 | R/W | 0h | 8 位发送缓冲区 5。要发送的字节 5 写入此寄存器，然后复制到 SCITXSHF 进行发送。复位类型：SYSRSn |
| 15-8 | TD6 | R/W | 0h | 8 位发送缓冲区 6。要发送的字节 6 写入此寄存器，然后复制到 SCITXSHF 进行发送。复位类型：SYSRSn |
| 7-0 | TD7 | R/W | 0h | 8 位发送缓冲区 7。要发送的字节 7 写入此寄存器，然后复制到 SCITXSHF 进行发送。复位类型：SYSRSn |

### 22.4.2.25 MBRSR 寄存器

偏移地址：F8h　复位值：00000DACh

MBRSR 寄存器用于配置 LIN 网络预期的最大波特率。

表 22-62 MBRSR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | MBR | | | | |
| R-0h | | | R/W-DACh | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MBR | | | | | | | |
| R/W-DACh | | | | | | | |

表 22-63 MBRSR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-13 | RESERVED | R | 0h | 保留 |
| 12-0 | MBR | R/W | DACh | 最大波特率预分频器。此字段仅在 LIN 模式下有效。如果 ADAPT 位置位，则在同步阶段使用此 13 位预分频器。这样，使用自动或选择位速率模式的 SCI/LIN 响应器，如果接收到的波特率在预期波特率的 10%（更低或更高）范围内，则会自动检测任何 LIN 总线合法速率。MBR = VCLK / (1.1*预期波特率)。MBR 值应编程为允许最大波特率不超过 LIN 网络中预期运行波特率的 10%。否则，0x00 数据字节可能被误检测为同步间隔。默认值适用于 70MHz LINCLK 和大约 18kHz 的预期波特率 (0xDAC)。此 MBR 预分频器由唤醒和空闲时间计数器使用，以实现相对于 20kHz 速率的恒定到期时间。复位类型：SYSRSn |

## 22.4.2.26 IODFTCTRL 寄存器

偏移地址：120h　复位值：00000500h

IODFTCTRL 寄存器用于仿真各种错误和测试条件。

表 22-64 IODFTCTRL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| BERRENA | PBERRENA | CERRENA | ISFERRENA | 保留 | FERRENA | PERRENA | BRKDTERRENA |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | PINSAMPLEMASK | | TXSHIFT | | |
| R/W-0h | | | R/W-0h | | R/W-0h | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | IODFTENA | | | |
| R-0h | | | | R/W-5h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | LPBENA | RXPENA |
| R-0h | | | | | | R/W-0h | R/W-0h |

表 22-65 IODFTCTRL 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | BERRENA | R/W | 0h | 位错误使能位。此位仅在 LIN 模式下有效。此位用于创建位错误。当此位置位时，接收到的位与 1 进行 OR 运算，并传递给位监视器电路。复位类型：SYSRSn |
| 30 | PBERRENA | R/W | 0h | 物理总线错误使能位。此位仅在 LIN 模式下有效。此位用于创建物理总线错误。当此位置位时，在同步间隔字段传输期间接收到的位与 1 进行 OR 运算，并传递给位监视器电路。复位类型：SYSRSn |
| 29 | CERRENA | R/W | 0h | 校验和错误使能位。此位仅在 LIN 模式下有效。此位用于创建校验和错误。当此位置位时，接收校验和计算器中 CTYPE（校验和类型）的极性被改变，以便生成校验和错误。复位类型：SYSRSn |
| 28 | ISFERRENA | R/W | 0h | 不一致同步字段错误使能位。此位仅在 LIN 模式下有效。此位用于创建 ISF 错误。当此位置位时，同步字段中的位宽会变化，使得 ISF 检查失败并设置错误标志。复位类型：SYSRSn |
| 27 | RESERVED | R | 0h | 保留 |
| 26 | FERRENA | R/W | 0h | 此位用于创建帧错误。此位仅在 SCI 兼容模式下有效。当此位置位时，接收到的停止位与 ‘0’ 进行 AND 运算，并传递给停止位检查电路。复位类型：SYSRSn |
| 25 | PERRENA | R/W | 0h | 仅兼容模式。此位仅在 SCI 兼容模式下有效。此位用于创建奇偶校验错误。当此位置位时，在兼容模式下，接收到的校验位被翻转，从而发生奇偶校验错误。复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 24 | BRKDTERRENA | R/W | 0h | 仅兼容模式。此位仅在 SCI 兼容模式下有效。此位用于创建 BRKDT 错误（仅 SCI 模式）。当此位置位时，帧的停止位与 ‘0’ 进行 AND 运算，并传递给 RSM，从而发生帧错误。然后 RX 引脚被强制持续低电平 10 个 Tbit，从而发生 BRKDT 错误。复位类型：SYSRSn |
| 23-21 | RESERVED | R/W | 0h | 保留 |
| 20-19 | PINSAMPLEMASK | R/W | 0h | 引脚采样掩码。这些位定义采样编号，在该采样点，正在发送的 TX 引脚值将被反转，以验证接收引脚使用多数检测电路是否正确采样。注意：在引脚采样掩码的 IODFT 模式测试期间，预分频器 P 必须编程为大于 2。复位类型：SYSRSn 0h (R/W) = 无掩码 1h (R/W) = 在 TBIT_CENTER 处反转 TX 引脚值 2h (R/W) = 在 TBIT_CENTER + SCLK 处反转 TX 引脚值 3h (R/W) = 在 TBIT_CENTER + 2 SCLK 处反转 TX 引脚值 |
| 18-16 | TXSHIFT | R/W | 0h | 发送移位。这些位定义 LINTX 上值的延迟量，以使 LINRX 上的值是异步的。（不适用于起始位）复位类型：SYSRSn 0h (R/W) = 无延迟 1h (R/W) = 延迟 1 个 SCLK 2h (R/W) = 延迟 2 个 SCLK 3h (R/W) = 延迟 3 个 SCLK 4h (R/W) = 延迟 4 个 SCLK 5h (R/W) = 延迟 5 个 SCLK 6h (R/W) = 延迟 6 个 SCLK 7h (R/W) = 延迟 7 个 SCLK |
| 15-12 | RESERVED | R | 0h | 保留 |
| 11-8 | IODFTENA | R/W | 5h | IO DFT 使能密钥。此字段用于使能 SCI/LIN 模块的 IODFT 模式以进行测试。复位类型：SYSRSn 0h (R/W) = IODFT 被禁止 1h (R/W) = IODFT 被禁止 2h (R/W) = IODFT 被禁止 3h (R/W) = IODFT 被禁止 4h (R/W) = IODFT 被禁止 5h (R/W) = IODFT 被禁止 6h (R/W) = IODFT 被禁止 7h (R/W) = IODFT 被禁止 8h (R/W) = IODFT 被禁止 9h (R/W) = IODFT 被禁止 Ah (R/W) = IODFT 被使能 Bh (R/W) = IODFT 被禁止 Ch (R/W) = IODFT 被禁止 Dh (R/W) = IODFT 被禁止 Eh (R/W) = IODFT 被禁止 Fh (R/W) = IODFT 被禁止 |
| 7-2 | RESERVED | R | 0h | 保留 |
| 1 | LPBENA | R/W | 0h | 模块回环使能。在模拟回环模式下，可以测试通过 I/O 的完整通信路径，而在数字回环模式下，I/O 缓冲器被排除在此路径之外。复位类型：SYSRSn 0h (R/W) = 数字回环被使能。1h (R/W) = 在模块 I/O DFT 模式下模拟回环被使能（当 IODFTENA = 1010 时） |
| 0 | RXPENA | R/W | 0h | 模块通过接收引脚的模拟回环使能。此位定义在仅模拟回环模式下，通信路径中是包含发送还是接收引脚的 I/O 缓冲器。复位类型：SYSRSn 0h (R/W) |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | = 通过发送引脚的模拟回环被使能。1h (R/W) = 通过接收引脚的模拟回环被使能。 |

### 22.4.2.27 LIN_GLB_INT_EN 寄存器

偏移地址：1C0h　复位值：00000000h

LIN_GLB_INT_EN 寄存器用于使能 INT0 和 INT1 中断线路传播到 CLIC 块。

表 22-66 LIN_GLB_INT_EN 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | GLBINT1_EN | GLBINT0_EN |
| R-0h | | | | | | R/W-0h | R/W-0h |

表 22-67 LIN_GLB_INT_EN 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | GLBINT1_EN | R/W | 0h | LIN INT1 的全局中断使能。此位决定 INT1 中断线路是否向 CLIC 生成中断。复位类型：SYSRSn<br>0h (R/W) = LIN INT1 线路不向 CLIC 生成中断。<br>1h (R/W) = 如果发生已使能的中断条件，LIN INT1 线路向 CLIC 生成中断。 |
| 0 | GLBINT0_EN | R/W | 0h | LIN INT0 的全局中断使能。此位决定 INT0 中断线路是否向 CLIC 生成中断。复位类型：SYSRSn<br>0h (R/W) = LIN INT0 线路不向 CLIC 生成中断。<br>1h (R/W) = 如果发生已使能的中断条件，LIN INT0 线路向 CLIC 生成中断。 |

### 22.4.2.28 LIN_GLB_INT_FLG 寄存器

偏移地址：1C8h　复位值：00000000h

LIN_GLB_INT_FLG 寄存器包含 INT0 和 INT1 标志的当前状态。

表 22-68 LIN_GLB_INT_FLG 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | INT1_FLG | INT0_FLG |
| R-0h | | | | | | R-0h | R-0h |

表 22-69 LIN_GLB_INT_FLG 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | INT1_FLG | R | 0h | LIN INT1 的全局中断标志。此位指示是否由于 INT1 中断线路上已启用的中断而向 CLIC 生成了中断。关于生成中断的条件，请参考 LIN 中断状态寄存器。此位可通过向 LIN_GLB_INT_CLR 寄存器中的相应位写 1 来清除。复位类型：SYSRSn<br>0h (R/W) = INT1 线上无活动中断。1h (R/W) = 由于 INT1 中断线路上已启用的中断而生成了中断。 |
| 0 | INT0_FLG | R | 0h | LIN INT0 的全局中断标志。此位指示是否由于 INT0 中断线路上已启用的中断而向 CLIC 生成了中断。关于生成中断的条件，请参考 LIN 中断状态寄存器。此位可通过向 LIN_GLB_INT_CLR 寄存器中的相应位写 1 来清除。复位类型：SYSRSn<br>0h (R/W) = INT0 线上无活动中断。1h (R/W) = 由于 INT0 中断线路上已启用的中断而生成了中断。 |

### 22.4.2.29 LIN_GLB_INT_CLR 寄存器

偏移地址：1D0h　复位值：00000000h

LIN_GLB_INT_CLR 寄存器用于清除 LIN_GLB_INT_FLG 寄存器中的中断标志。

表 22-70 LIN_GLB_INT_CLR 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | INT1_FLG_CLR | INT0_FLG_CLR |
| R-0h | | | | | | R/W1C-0h | R/W1C-0h |

表 22-71 LIN_GLB_INT_CLR 寄存器字段描述

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-2 | RESERVED | R | 0h | 保留 |
| 1 | INT1_FLG_CLR | R/W1C | 0h | LIN INT1 的全局中断标志清除。此位用于清除 LIN_GLB_INT_FLG 寄存器中的相应位。写 1 清除 INT1_FLG 位。写 0 无效。复位类型：SYSRSn |
| 0 | INT0_FLG_CLR | R/W1C | 0h | LIN INT0 的全局中断标志清除。此位用于清除 LIN_GLB_INT_FLG 寄存器中的相应位。写 1 清除 INT0_FLG 位。写 0 无效。复位类型：SYSRSn |

# 23 电源管理总线（PMBus）

## 23.1 简介

### 23.1.1 特征

- 支持主机和从机模式
- 支持 I2C 模式
- 支持三种传输速度：标准模式：达 100kHz 快速模式：达 400kHz
- 数据包纠错
- CONTROL 和 ALERT 信号
- 高低时钟时间输出
- 四字节发送和接收缓存
- 一个可屏蔽中断，由以下条件产生：
    1) 接收数据准备就绪；
    2) 传输缓存空；
    3) 从机地址接收；
    4) 信息结束；
    5) ALERT 输入断言；
    6) 时钟低电平时间输出；
    7) 时钟高电平时间输出；
    8) 总线空闲；

### 23.1.2 模块示意图

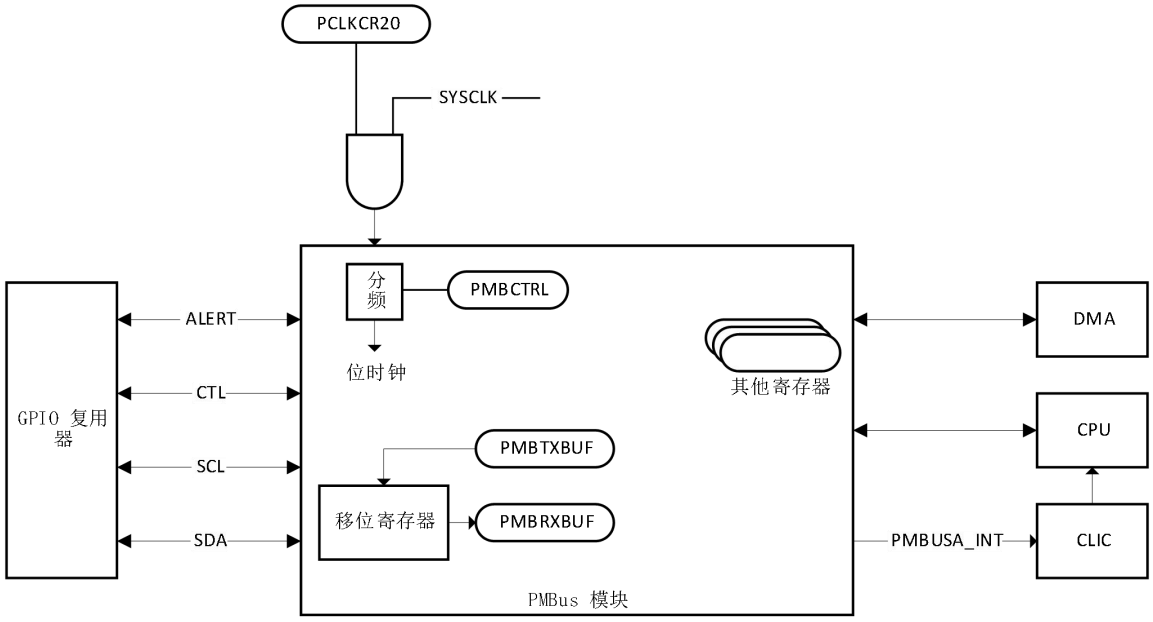



图 23-1 PMBus 模块示意图

PMBus 模块处理 PMBus 协议的较低级别。除了控制信号电平和定时、解析地址和缓冲数据外，它还直接支持复杂的事务，如读字和进程调用。

以下是 PMBus 的四个信号：

- SCL: 总线时钟，它通常由主机控制，但可能会被 slave 控制在低位，以延迟事务并 允许更多的时间处理。
- SDA：双向数据线。
- CONTROL：可以触发中断的从机输入，可用来关闭从设备。
- ALERT： 是否允许从机请求主机响应的从机输出主机输入信号。

SCL 和 SDA 的时序由 SYSCLK 产生。为了符合 PMBus 时序要求，由 PMBTIMCLK 产生的时钟分频不得高于 10MHz。

### 23.1.3 配置设备引脚

GPIO 选择寄存器必须配置为连接这一外部设备。为了避免引脚错误，GPyGMUX 为必须先被配置（同时将对应的 GPyMUX 为保持为默认值 0），然后配置 GPyMUX 寄存器中对应的值。

一些 IO 功能是独立于外设的 GPIO 寄存器设置定义的。对于输入信号，通过将 GPxQSELn 寄存器设置为 11b，可以将 GPIO 的输入信号设置为异步模式。内部上拉可在 GPyPUD 寄存器中设置。

## 23.2 从机模式运行

该部分介绍从机模式下的配置和操作。

### 23.2.1 配置

要配置该模块，将时钟分频器配置 PMBSTRL 寄存器，以产生小于 10MHz 的时钟频率。要激活从机模式，在 PMBCTRL 寄存器中设置 SLAVE_EN 位。接下来，设置 PMBSC 寄存器。以下选项是可配置的：

- 从机地址和掩码（SLAVE_ADDR,SLAVE_MASK）：设置接收消息的从机地址和掩码。
- 手动从地址确认（MAN_SLAVE_ACK）：启动后，允许软件决定是否响应地址。禁用时，根据从地址和掩码自动做出响应。
- PEC 使能（PEC_ENA）：如果在总线上使用包错误检查（PEC）则设置该位。
- 手动命令字节确认（MAN_CMD）：与手动确认从机响应类似，设置该位允许软件决定是否确认 ACK 响应字节。
- 自动确认字节数（RX_BYTE_ACK_CNT）：通常设置为最大值，他允许整个接收 buffer 使用。然而，如果应用程序要求尽快监测错误消息并不确认，则可以使用较小的值。

通过向 PMBACK 寄存器写入一个 1 来完成手动确认。即使使用自动确认，也需要对 PMBACK 进行一些写操作。如果信息（不包括地址）长度超过 4 字节，则每个四字节的数据包将被确认。PMBus 模块延长时钟（保持低），直到发出 ACK。然后模块将数据线拉低并释放时钟，向主机发送 ACK 信号。

如果一个完整的消息或消息的最后一小部分小于 4 字节（或 RX_BYTE_ACK_CNT 限制），则不写入 PMBACK。

在 PMBACK 写一个 0，发送一个 NACK。这只能在模块等待确认完成。如果在任何其它时间写入 0，则在下一条消息中发出 NACK。

### 23.2.2 信息处理

该部分描述 PMBus 的一些消息类型，以及如何确定在从属状态下接收消息类型。本节面向最有效的操作模式，具有自动地址和命令确认，也面向具有包错误检测启用。

如果禁用自动地址确认，则所有消息都从设置 SLAVE_ADDR_READY 位开始。读命令有两条指令，一条用于读，一条用于写。如果自动地址确认使能，那么 DATA_READY 也会设置高。如果消息没有 PEC，则可用字节数为 n-1。例如，有 PEC，QUICK 命令有一个字节，没有 PEC，QUICK 命令为 0 字节。

需要注意的是字节计数不会随着字节到达而增加。在接收到停止消息、接收缓冲区已满或者发生故障之前，PMBST 寄存器中不会设置任何位。然而，所有适当的位置一起放在寄存器中。接收快速命令所需要做的就是通向 PMBACK 寄存器写入 1 来确认消息。

#### 23.2.2.1 快捷指令

PMBus 模块在从模式下收到的快速指令需要对收到的设备进行简单的确认。在地址自动确认模式下，模块处理快速命令，无需固件交互。在接收消息结束后，固件可以选择读取 PMBus 寄存器中的接收地址。在手动地址确认模式下，地址通过写入 PMBACK 寄存器来确认。




图 23-2 快速指令消息

#### 23.2.2.2 发送字节

字节消息发送由设备地址、数据字节和可选的 PEC 字节组成。为了正确处理 PEC 字节，PEC 运算必须使能 PMBUS 寄存器。在自动地址确认模式，数据和可选的 PEC 字节在没有固件交互时被确认。模块产生一个消息结束中断，读取状态寄

存器并找到数据就绪准备指示位。在手动模式下，地址通过固件确认，而剩下的数据和 PEC 字节由模块确认。

PMBus 模块储存数据字节在 PMBRXBUF 寄存器。数据字节将储存在 7-0 位。在无 PEC 模式，PMBSTS 寄存器中的接收字节数量表示接收一个字节。如果 PEC 运算使能，PEC 字节将被储存在 PMBRXBUF 寄存器，PEC 字节占据 15-8 位。PMBSTS 寄存器中的接收字节数量表示接收两个字节。PEC 在 PMBSTS 寄存器中的有效位表示接收 PEC 字节的有效性。

当发送字节接收以后，数据准备位与 EOM 一起设置，假设 PEC 有效，则 PEC 位有效。读字节计数寄存器（RD_BYTE_CONT）将包含一个 2。接收发送字节命令所需要的就是通向 PMBACK 寄存器写 1 来确认消息。在执行 ACK 之前，从 PMBACK 寄存器的最低字节读取字节。




图 23-3 发送字节消息有无 PEC

### 23.2.2.3 接收字节

接收字节消息发送由设备地址、数据字节和可选的 PEC 字节组成。在自动地址确认模式下，固件在接收从地址后接收数据请求中断。发送到主机的数据字节存储在 PMBTXBUF 寄存器的 7-0 位。PMBSC 寄存器中的发送字节计数为 1。如果使能 PEC 处理，则 PMBSC 寄存器中的传输 PEC 位（第 19 位）与启用 PEC 位（第 15 位）一起设置为 1。该模块在消息结束时自动附加计算的 PEC 字节。




图 23-4 接收字节消息有无 PEC

### 23.2.2.4 写字节与写字

写字节/写字报文由从机地址、命令字节、数据字节和可选的 PEC 字节组成。在自动地址确认模式，数据字节和可选的 PEC 字节在无固件交互时确认。命令字的确认是通过 PMBSC 寄存器配置的。固件在所有情况下都接收到消息结束中断，除了带有 PEC 消息的写字，读取状态寄存器并找到数据就绪指示位。

在带有 PEC 字节消息的写字过程，在接受到 4 个字节的后启用数据就绪准备中断（命令字节，2 个数据字节和 PEC 字节）。固件从 PMBRXBUF 寄存器中读取数据，并且必须写入 PMBACK 寄存器以确认传给主机。PMBus 模块保持 SCL 低，直到固件响应接收到数据。

在其它情况下，EOM 中断被接收，数据可以从 PMBRXBUF 寄存器中读取。固件不需要发送确认回给主机。

写字节消息看起来和发送字节一致，除了 RD_BYTE_COUNT 寄存器包含一个 3，写字的 RD_BYTE_COUNT 为 4。

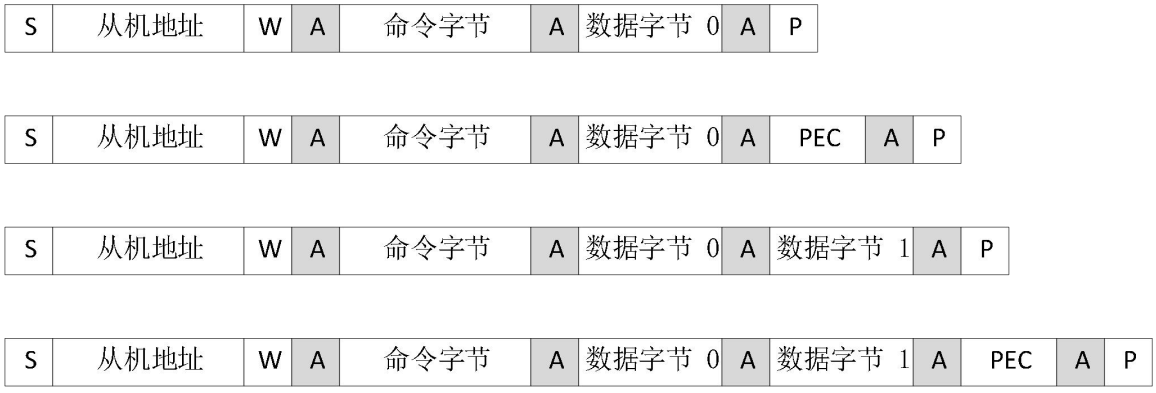



图 23-5 写字节和写字消息有无 PEC

### 23.2.2.5 读字节与读字

读字节/读字报文由从机地址、命令字节、数据字节和可选的 PEC 字节组成。地址和命令确认通过 PMBus 寄存器配置。在自动模式下，PMBus 模块在接收到重复起始条件及随后发送的从机地址后提供数据就绪和数据请求中断。接收到的命令字节位于 PMBRXBUF 寄存器的第 7-0 位。固件通过将数据字节存储到 PMBTXBUF 寄存器和 PMBSC 寄存器中的 TX 字节计数位来响应数据请求。如

果 PEC 处理被使能，发送 PEC 位也被置位。一个 EOM 中断表示对主机的信息发送完成。

当重复开始（SR）信号被接收时，该数据就绪位将被置位，RD_BYTE_COUNT 置 1。此时，该操作无法与组命令发送字节消息区分开来。当通过读取发送相同的设备地址时，数据请求位将被置位。如果在收到设备地址之前数据已经写入 PMTXBUF 寄存器，则数据请求位不会被置位。如果组的命令是还被期待，读取数据准备与 RD_BYTE_COUNT 置 1，这是必要的。然后等着看接下来的事件是否是 EOM 或数据请求。如果它是一个 EOM，命令应该被处理为一组发送字节。如果它是一个数据请求，该指令应该被处理为一个读操作。根据命令，它可以是读取的字节、字或块。如果轮询 PMBus 模块，则数据就绪和数据请求位都可能在轮询间隔之间设置。在设计固件时应考虑到这一点。

一旦读取命令被识别，您必须通过向 PMBTXBUF 寄存器写入数据来响应。还需要确保 PMBSC 寄存器中的值正确。必须适当设置发送字节计数和 PEC 位。对于读取字节，发送字节计数可以加载 1。如果需要发送 PEC 字节，则必须设置 TX_PEC 位。此后，数据可以写入 PMBTXBUF，从而开始传输。所有字节应同时写入 PMBTXBUF。主机收到消息后，会对最后一个字节进行 NACK，以表明已接收到正确的字节数。这将导致 PMBSTS 寄存器中的 EOM 位被设置，向固件指示读取字节消息已完成。

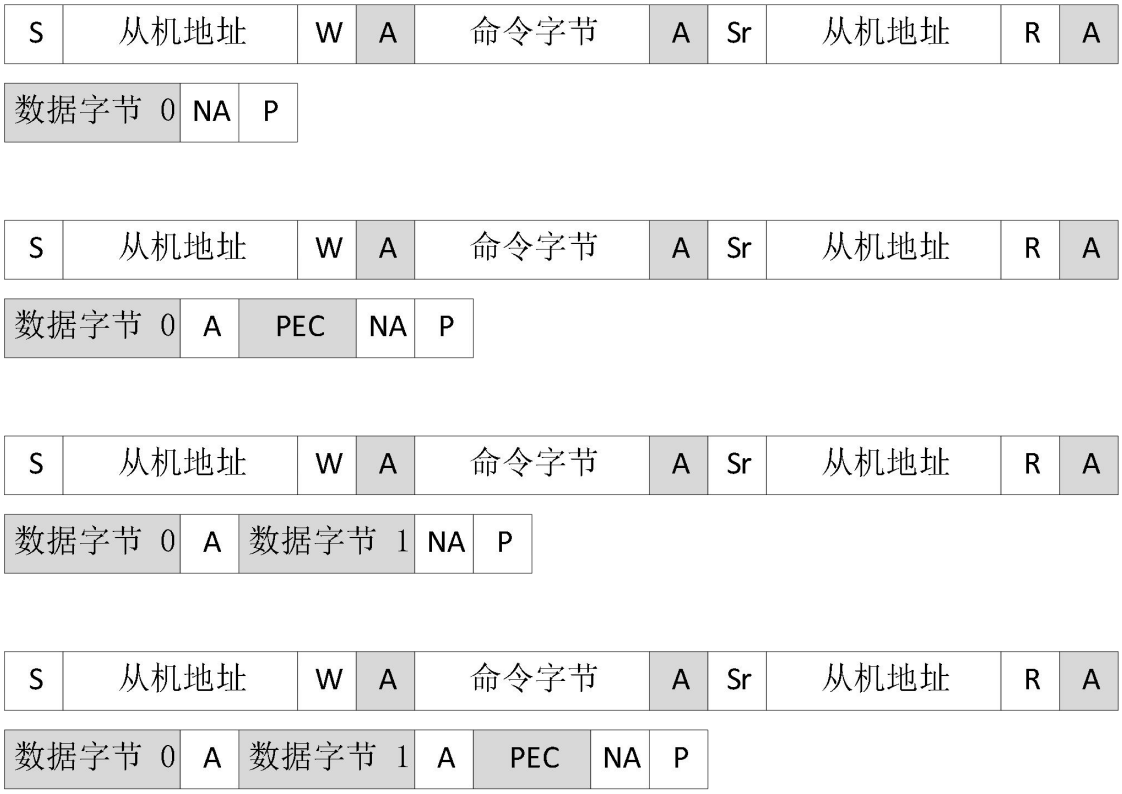



图 23-6 读字节和读字消息有无 PEC

### 23.2.2.6 进程调用

进程调用（图中）协议由一个写字消息和一个读字消息组成，两个消息之间没有停止条件。地址和命令确认通过 PMBSC 寄存器进行配置。在自动模式下，收到重复起始条件及随后发送的从机地址后，PMBus 模块提供数据就绪和数据请求中断。在 PMBSTS 寄存器中设置重复起始位以指示接收到进程调用消息的第一部分。接收到的命令字节位于 PMBRXBUF 寄存器的第 7-0 位，而从主机接收的两个数据字节位于位 23-8。收到来自模块的重复启动和数据请求后，固件将使用 2 个数据字节对 PMBTXBUF 进行编程，以发送给主机。如果使能 PEC 处理，则 PMBSC 寄存器中的发送 PEC 位被置位。EOM 中断将指示模块已完成进程调用消息的读取字部分。




图 23-7 进程调用消息有无 PEC

### 23.2.2.7 块写

块写入（图中）协议在结构上与写字类似，只是报文中的数据超过两个字节。在收到命令字节、块长度和 2 个数据字节后，PMBus 模块产生数据就绪中断。模块等待固件读取接收到的数据并对确认寄存器进行编程。在等待来自固件的 ACK 时，模块会将时钟线驱动为低电平，从而停止总线。数据就绪中断将以每 4 个数据字节的频率在消息的持续时间内继续。接收到的字节数可以在 PMBSTS 寄存器中找到。在消息结束时，PMBRXBUF 寄存器中可以存储少于 4 个字节。可以检查 PEC 有效位以确定接收到的 PEC 值是否准确。

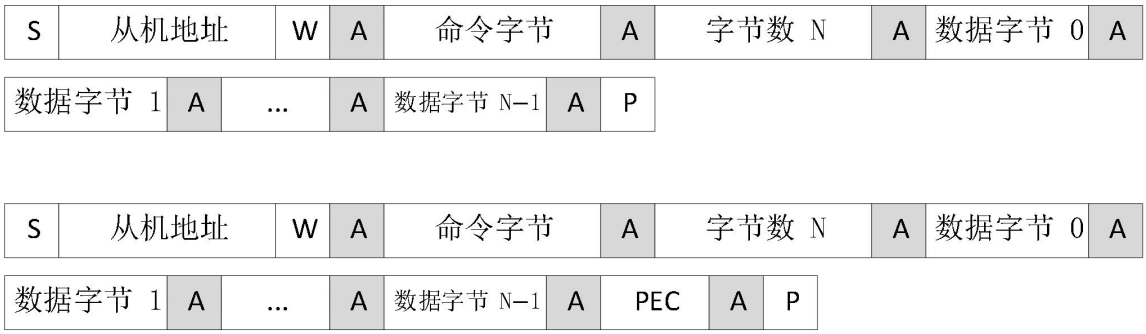



图 23-8 块写消息有无 PEC

### 23.2.2.8 块读

块读取协议在结构上类似于读取字，不同之处在于消息中有 2 个以上的数据字节。在接收到重复的从地址后，PMBus 模块会产生数据就绪和数据请求中断。从主机接收的命令字节可以在 PMBRXBUF 寄存器的位 7-0 中找到。SCL 线保持低电平，直到固件将数据字节编程到 PMBTXBUF 寄存器中。在寄存器的初始

编程期间，固件需要将块长度加载到 PMBTXBUF 寄存器的第 7-0 位。发送 4 个字节后，模块将发出数据请求中断并再次保持 SCL 为低电平，直到固件将附加数据写入 PMBTXBUF 寄存器。

块读取的开始与读取字或读取字节相同，但第一次加载 TX_COUNT 时为 4，且未设置 TX_PEC。固件不是在第一次传输后等待 EOM，而是等待数据请求，表明主设备已准备好接收更多数据。直到最后 4 个或更少字节，固件简单地将 4 写入 TX_COUNT，然后将 4 个字节写入 PMBTXBUF。TX_PEC 保持清零。然后，当要传输最后 4 个或更少字节时，固件写出适当的字节数，设置 TX_PEC 位，并将数据写入 PMBTXBUF。PMBus 模块将写出数据，然后是 PEC，然后当主机 NACK PEC 时 EOM 位将被设置。




图 23-9 块读消息有无 PEC

### 23.2.2.9 块读-块写调用

块写入-块读取进程调用协议结合了块写入和块读取协议，消除了两个消息之间的停止条件。块写入-块读取进程调用消息的处理类似于进程调用消息的操作模式。确认地址和命令字节后，PMBus 模块在检测到 4 个数据字节或重复启动条件时产生数据就绪中断。收到重复启动后，固件会要求加载发送数据发送给主控。PMBTXBUF 寄存器初始编程的第 7-0 位必须表示发送到主机的块数据的字节数。




图 23-10 块读块写调用消息有无 PEC

### 23.2.2.10 警报响应

当主机检测到来自 PMBus 上的从机的警报条件时，使用警报响应消息（图中）。在自动地址确认模式下，当检测到警报响应地址时，PMBus 模块向主设备提供确认，并在 PMBSC 寄存器中发送编程的从设备地址。如果 PMBCTRL 寄存器中的 Alert En 位先前已设置，则模块仅响应该消息。收到警报响应消息后，模块将清除 PMBCTRL 寄存器中的警报条件和使能位。在手动地址确认模式下，固件必须从 PMBRXBUF 寄存器读取接收到的地址并将所需的从地址发送回主地址。PMBCTRL 寄存器必须重新编程以禁用用于启动来自主机的警报响应消息的警报 En 位。




图 23-11 警报响应消息

### 23.2.2.11 扩展命令

PMBus 模块提供对允许额外 256 个命令代码的扩展命令的支持。两个命令字节都与数据字节一起存储在 PMBRXBUF 寄存器中。在识别扩展命令消息时，使用了 PMBSTS 寄存器中的重复起始位和 Rd 字节计数位。对于扩展命令写字节/写字消息（图中），两个命令字节存储在 PMBRXBUF 寄存器的位 15-0 中。初始命令字节应保存命令扩展代码，表示扩展命令协议的使用。重传设备地址后接收到的重复起始位也被设置。对于 Ext Cmd Write Byte 消息，Rd Byte Count 等于 3，对于 Ext Cmd Write Word 消息，Rd Byte Count 等于 4。

对于扩展命令读取字节/读取字消息，模块将在接收到重复的设备地址后生成数据就绪和数据请求中断。这两个命令字节位于 PMBRXBUF 寄存器的第 15-0 位，初始命令字节与命令扩展代码相匹配。固件将需要加载传输数据以完成将消息返回给主机。




图 23-12 扩展命令写字节与写字消息有无 PEC

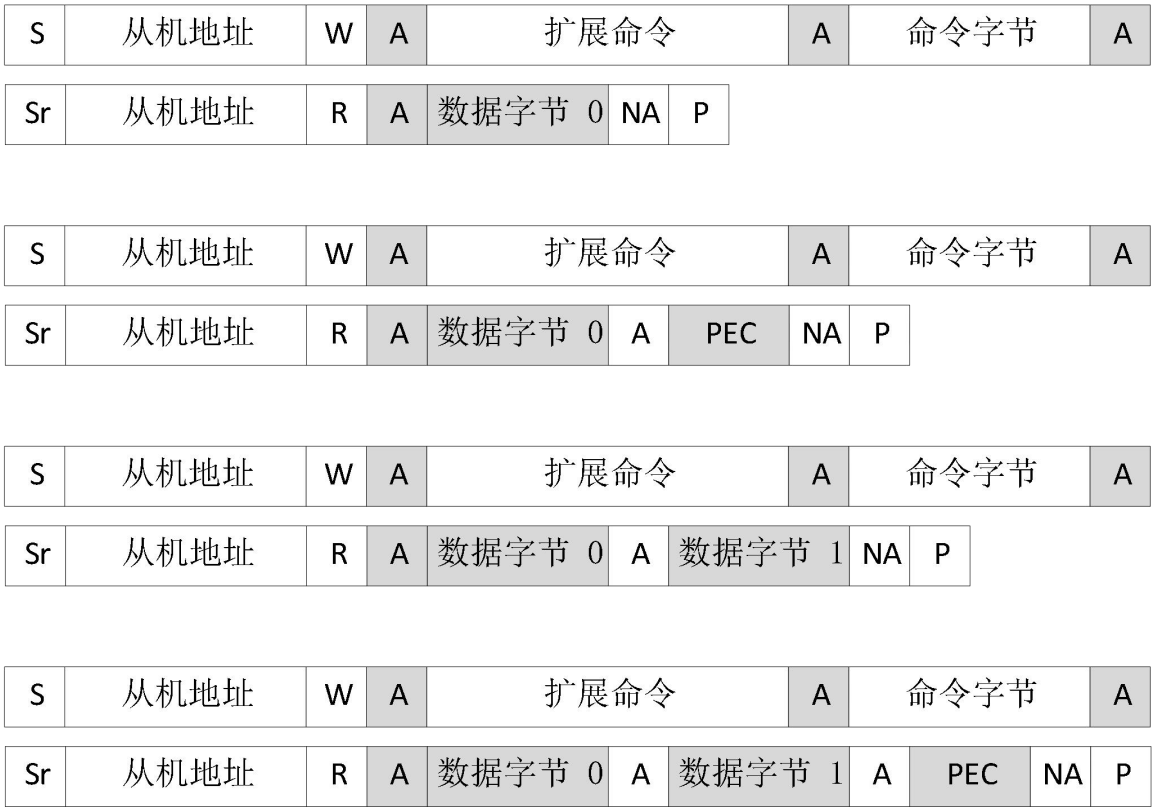



图 23-13 扩展命令读字节与读字消息有无 PEC

### 23.2.2.12 组命令

PMBus 模块支持组命令协议。组命令协议用于在同一消息中向多个设备发送命令。当总线上的设备在组命令消息结束时检测到停止条件时，将同时执行接收到的命令。在地址和命令确认之后，模块在检测到 4 个数据字节或在总线上传输重复启动时提供数据就绪中断。根据使用组命令消息的要求，固件应在处理接收到的命令之前等待 EOM 中断。

对于组命令，一旦接收到重复启动，数据就绪位将被设置。然后可以将数据读入内存。但是在 EOM 位被设置之前不应处理数据，这将在所有消息都已收到时发生。除了这个延迟的 EOM 之外，从固件在接收组命令方面与任何其他写入消息没有区别。




图 23-14 组命令消息有无 PEC

## 23.3 主机模式运行

本节介绍主模式下 PMBus 模块的配置和操作。

### 23.3.1 配置

首先，将合适的时钟分频器写入 PMBCTRL 寄存器的 CLKDIV 字段，以产生小于 10 MHz 的位时钟频率。要激活主模式，需要设置 MASTER_EN 位并清除 PMBCTRL 寄存器中的 SLAVE_EN 位。对于每个事务，设置 PMBMC 寄存器。以下选项是可配置的：

从地址(SLAVE_ADDR)：设置下一个事务的从地址。

PEC 使能 (PEC_ENA)：如果在总线上使用数据包错误检查 (PEC)，则设置该位。

扩展命令代码使能(EXT_CMD)：设置后，使用两个字节的命令。

命令代码使能(CMD_ENA)：设置后，在事务开始时发送命令字节。

字节计数（BYTE_COUNT）： 确定的数目的数据字节来传输。该数量不包括所述块长度字节，需要的时候自动生成。

特殊命令启用（GRP_CMD 和 PRC_CALL）：启用组命令和进程调用的特殊行为。

写入 PMBMC 寄存器将启动传输。不需要手动确认接收到的数据。

### 23.3.2 信息处理

本节介绍每种命令类型的行为和所需的配置。

#### 23.3.2.1 快捷指令

将所需的从设备地址编程到 PMBMC 中，即可在主模式下启动快速命令（图中）。通过向位 15-8 写入全零，PMBMC 寄存器中的字节数被配置为 0 字节。在传输设备地址后，PMBus 模块将监控地址的从机确认。如果地址未被确认，状态寄存器中的 NACK 位被使能，PMBus 模块自动在总线上发送一个停止条件以终止消息。如果地址被确认，则向处理器发出数据请求。固件向 PMBACK 写入零以终止消息，强制 PMBus 模块将停止条件写入总线。




图 23-15 快捷指令消息

#### 23.3.2.2 发送字节

发送字节消息（图中）由器件地址、单个数据字节和可选的 PEC 字节组成。要启动发送字节消息，将要发送到从器件的数据字节装入 PMBTXBUF 寄存器的第 7-0 位。PMBMC 寄存器配置有设备地址。为了发送带有报文的 PEC 字节，当地址被编程时，PMBMC 寄存器中的 PEC_EN 位被置为高电平。







图 23-16 发送字节消息有无 PEC

#### 23.3.2.3 接收字节

接收字节消息由器件地址、单个数据字节和可选的 PEC 字节组成。主设备通过接收字节消息从从设备读取数据。为了启动接收字节消息，固件将器件地址、R/W 位和可选的 PEC_EN 编程到 PMBMC 寄存器中。将 R/W 位置 1 以指示读取消息类型（从从设备传输到主设备的数据）。

在对 PMBMC 寄存器进行编程后，PMBus 模块发送接收字节消息。固件可以等待来自 PMBus 模块的消息结束中断来验证消息传输的准确性。收到 EOM 中断后，将读取 PMBSTS 寄存器以验证从设备地址的正确确认并确定 PMBRXBUF 寄存器中是否有任何数据可供读取。如果 PMBMC 寄存器中的 PEC_EN 被置位，则还检查 PMBSTS 寄存器中的 PEC_VALID 位，以确认随数据接收的 PEC 字节有效。




图 23-17 接收字节消息有无 PEC

#### 23.3.2.4 写字节写字

写字节和写字消息由器件地址、命令字节、传输的数据字节和可选的 PEC 字节组成。写字节报文包括单个字节，而写字报文支持向相应的从模块传输 2 个字节。与发送字节协议类似，PMBMC 寄存器配置为发送 1 或 2 个字节，设置 CMD_EN 位以启用命令字节传输，并设置可选的 PEC_EN 位。

启用命令字节传输后，PMBTXBUF 寄存器的格式与发送字节协议不同。在位 7-0 中，固件必须对要发送到从机的命令字节进行编程。数据字节被编程到位 15-8 和位 23-16。

在对 PMBMC 寄存器进行编程后，PMBus 模块发送写字节/字消息。 固件可以等待来自模块的消息结束中断来验证消息传输的准确性。PMBSTS 寄存器指示从设备是否正确确认了消息。

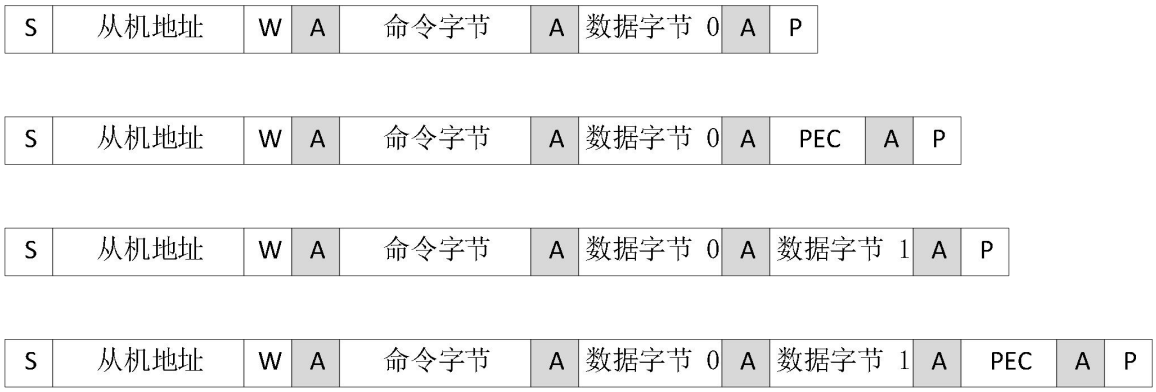



图 23-18 写字节写字消息有无 PEC

### 23.3.2.5 读字节读字

读字节和读字消息（图中）由器件地址、命令字节、从器件接收的数据字节和可选的 PEC 字节组成。读字节消息包括单个字节，而读字消息协议支持从从机接收 2 个字节。与接收字节协议类似，PMBMC 寄存器被配置为接收 1 或 2 个字节，CMD_EN 位被设置，PEC_EN 被配置为期望或不期望附加到消息的 PEC 字节。PMBus 模块将在从从机接收到预期字节数后或从机未正确确认消息的任何部分后自动终止消息。

除了对 PMBMC 寄存器进行编程之外，固件还需要将命令字节加载到 PMBTXBUF 寄存器的第 7-0 位。从从机接收到的任何数据都可以在 PMBRXBUF 寄存器中找到。

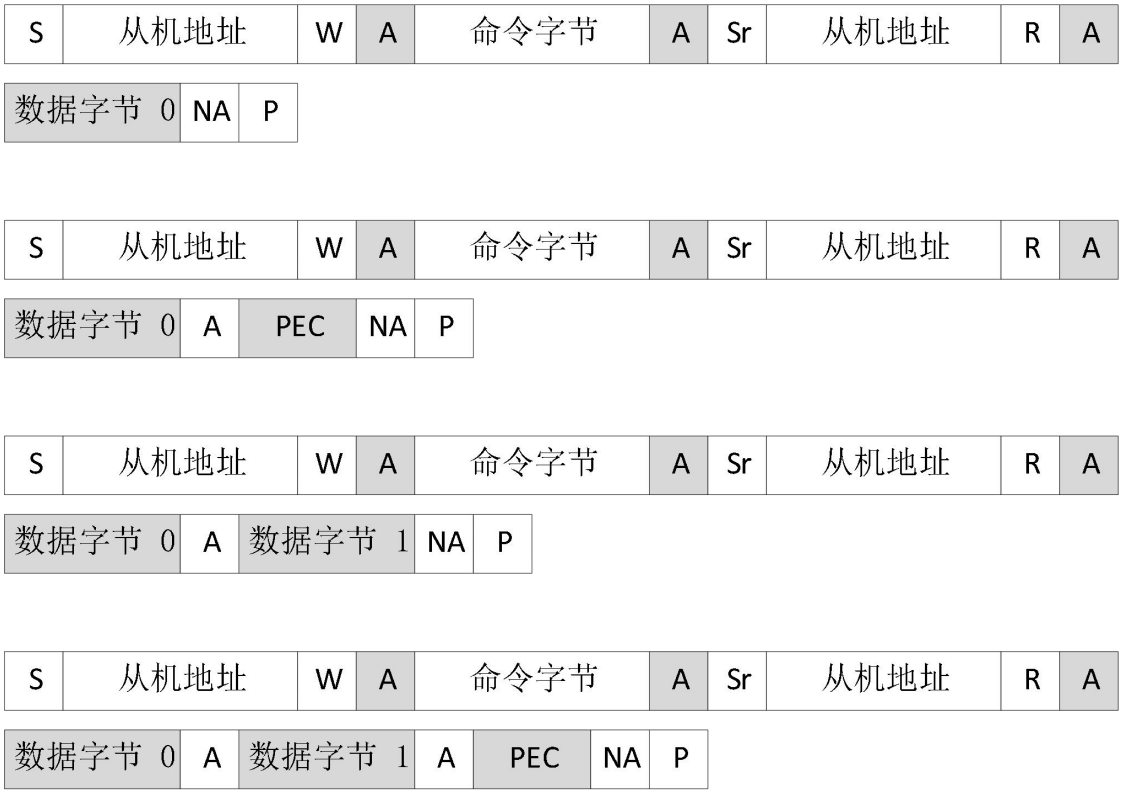



图 23-19 读字节读字消息有无 PEC

### 23.3.2.6 进程调用

进程调用协议由一条写入字消息和一条读取字消息组成，两条消息之间没有停止条件。PEC 字节可以作为消息协议的一个选项附加到从机读取的数据中。PMBMC 寄存器包括一个 PRC_CALL 位，它允许将进程调用消息传输到 PMBus。当进程调用位使能时，PMBus 模块将自动生成重复启动条件并启动消息的读取字部分。

为了完成进程调用的写字部分，在 PMBTXBUF 寄存器的第 7-0 位加载命令字节，并将数据字节加载到寄存器的第 23-8 位。

在对 PMBMC 寄存器进行编程后，PMBus 模块发送进程调用消息。固件可以等待来自模块的消息结束中断以确定消息的有效性。收到 EOM 后，PMBSTS 寄存器应指示收到来自进程调用消息的读取字部分的 2 个字节以及发送数据的从机确认状态。如果使能 PEC 处理，则 PMBSTS 寄存器中的 PEC_VALID 位指示在消息的读取字部分期间从从机接收的 PEC 字节的准确性。

对于下一个非进程调用消息，必须禁用 PMBMC 寄存器中的 PRC_CALL 位。请注意，对 PMBMC 寄存器的任何写入都会启动一条消息，因此在固件需要传输新消息之前，不建议重新配置主机。




图 23-20 进程调用消息有无 PEC

### 23.3.2.7 块写

块写入协议在结构上类似于写入字，不同之处在于消息中传输的数据字节超过 2 个。此外，命令字节之后的第一个数据字节指定数据字节块的长度。与大多数消息协议一样，PEC 字节可以附加到从机写入数据的末尾。

在总线上启动块写入消息，请使用字节计数位中的块长度对 PMBMC 寄存器进行编程。块长度是数据字节数，不包括命令字节和包含块长度的第一个数据字节。如果固件指定的数据字节数超过 2，PMBus 模块会自动将块长度插入报文中。初始写数据加载到 PMBTXBUF 寄存器中。位 7-0 表示命令字节，其余 3 个字节表示块长度后的前三个数据字节。

在对 PMBMC 寄存器进行编程之后，将发送块写消息。如果块长度超过三个字节，PMBus 模块将提供数据请求中断，表明需要在 PMBTXBUF 寄存器中添加额外的数据字节。PMBus 模块假设如果需要超过 4 个字节来完成消息，固件将在编程 PMBTXBUF 寄存器时使用所有 4 个字节。如果完成块写入消息所需的字节少于 4 个字节，则固件只需对 PMBTXBUF 寄存器的相应位进行编程。

消息完成后，PMBus 模块发出 EOM 中断。可以检查 PMBSTS 寄存器以验证从设备是否接受了写数据块。

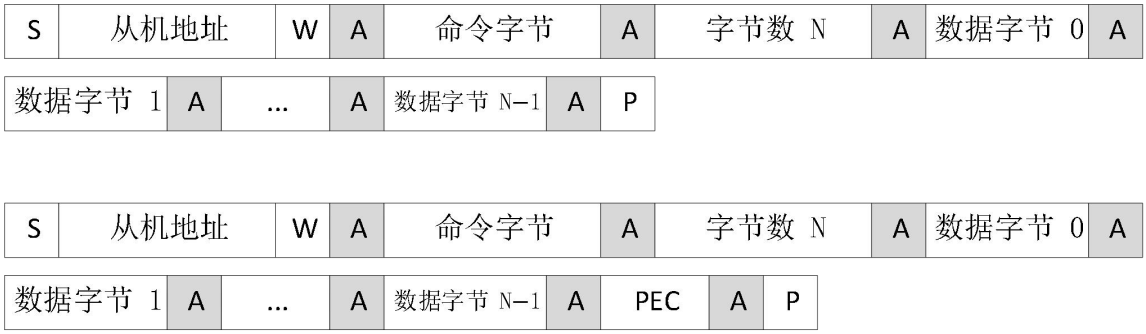



图 23-21 块写信息有无 PEC

### 23.3.2.8 块读

块读取协议在其结构上类似于读取字，不同之处在于从从机接收到 2 个以上的数据字节。从机传输的第一个数据字节表示从机正在写入的数据的块长度。如果启用 PEC 处理，则从设备将 PEC 字节附加到消息的末尾。

在 PMBus 上启动块读取消息，请使用字节计数位中的块长度对 PMBMC 寄存器进行编程。 此计数不包括命令字节，任何从属地址和块长度在字节消息中。在对 PMBMC 寄存器进行编程之前，将要发送到从机的命令字节写入 PMBTXBUF 寄存器的第 7-0 位 。

配置 PMBMC 寄存器后，发送块读消息。模块在从从机收到 4 个数据字节后中断固件。如果块长度为 3，EOM 中断将与数据就绪中断同时接收。否则，只有数据就绪中断被断言，表明固件已准备好读取 4 个字节。在消息结束时，PMBRXBUF 寄存器中可以存储少于 4 个字节。PMBSTS 寄存器中的 RX 字节计数位指示最终数据传输中可用的字节数。固件可以在检测到消息结束中断时验证接收到的 PEC。




图 23-22 块读信息有无 PEC

### 23.3.2.9 块写-块读进程调用

块写入-块读取进程调用协议结合了块写入和块读取协议，消除了两个消息之间的停止条件。主机的操作类似于块写入操作。将块长度加载到 PMBMC 寄存器的字节计数位提供消息的块写入部分的长度。此外，必须启用 PMBMC 寄存器中的 PRC_CALL 位。完成消息的块写入部分后，PMBus 模块将自动在 PMBus 上发出重复启动条件并开始传输消息的块读取部分。重复启动条件后 PMBus 模块的操作与它在简单块读取消息中的操作相同。




图 23-23 块写-块读进程调用信息有无 PEC

### 23.3.2.10 警报响应

当主机检测到来自从机的警报条件时，使用警报响应消息。在主模式下，警报响应消息只是一个接收字节消息，PEC 禁用，从地址设置为 0xC（警报响应地址）。PMBus 模块检测输入上的警报条件并中断固件以指示警报条件的断言（从设备希望与主设备通信）。使用警报响应地址对 PMBMC 寄存器进行编程会启动警报响应消息并提供请求服务的从设备地址。收到 EOM 中断后，将在 PMBRXBUF 寄存器中找到设备地址。




图 23-24 警报响应信息

### 23.3.2.11 扩展指令

PMBus 模块支持扩展命令，允许额外的 256 个命令代码。通过置位 PMBMC 寄存器内的 EXT_CMD 位，两个命令字节在消息协议上传输。扩展命令可以添加到写字节和写字（图中）和读字节和读字（图中）协议。PMBus 模块在扩展命令模式下的操作与这些格式类似。在对写数据或读报文的第一部分进行编程时，第二个命令字节与其余数据字节一起加载到 PMBTXBUF 寄存器的第 15-8 位。除了在写消息中包含重复启动条件和从地址之外，模块的其余操作与之前的协议相同。固件不需要支持写入消息中的这些附加字节。模块将解释 EXT_CMD 位并进行适当的格式更改。




图 23-25 扩展指令写字节写字信息有无 PEC

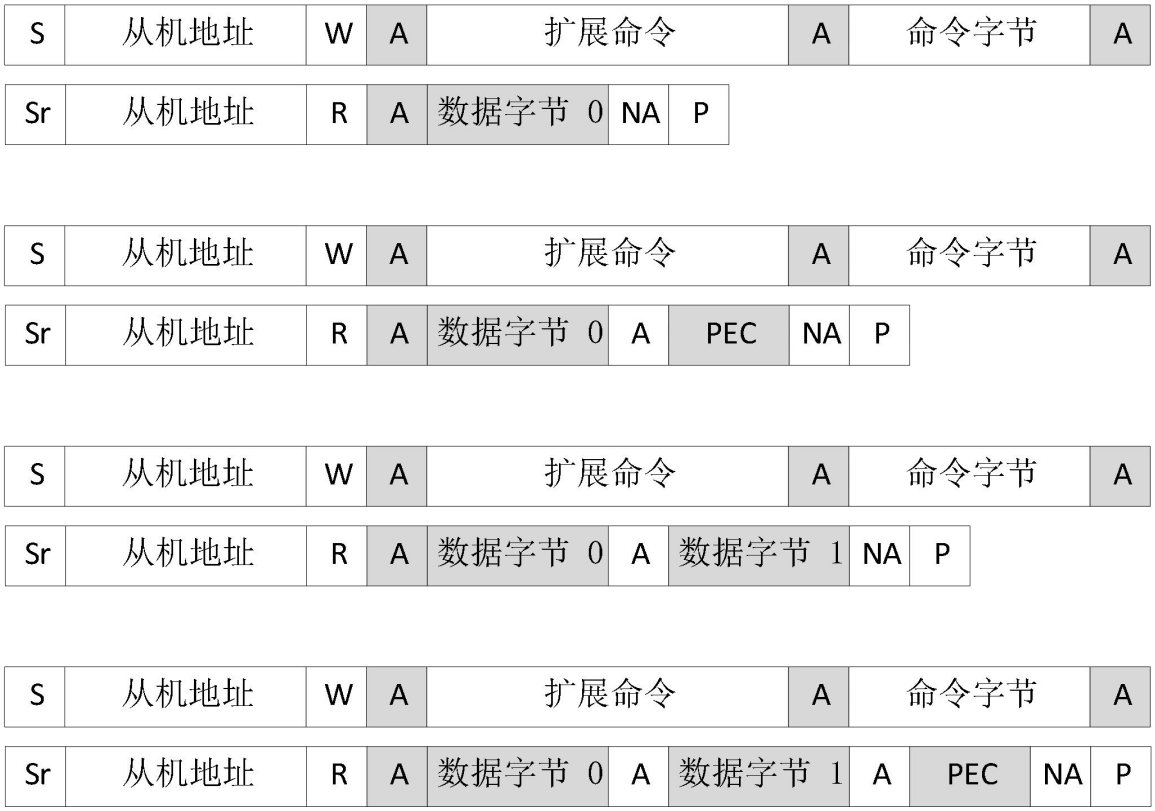



图 23-26 扩展指令读字节读字信息有无 PEC

### 23.3.2.12 组命令

组命令（图中）协议用于在同一消息中向多个设备发送命令。当总线上的设备在组命令消息结束时检测到停止条件时，将同时执行接收到的命令。要启动组命令，在对消息中的第一个设备的从地址进行编程时，必须设置 PMBMC 寄存器中的 GRP_CMD 位。消息的其余部分作为写入字节/字消息进行处理。在组命令消息的第一部分结束时，固件会在 PMBMC 寄存器中编程下一个设备地址。PMBus 模块将在总线上发送一个重复的开始并开始消息的下一部分。在编程组命令消息的最后一个设备地址时，固件必须在编程 PMBMC 寄存器时禁用 GRP_CMD 位。




图 23-27 组命令信息有无 PEC

# 23.4 寄存器描述

本节描述了电源管理总线模块寄存器。

## 23.4.1 PMBUS 基地址表

表 23-1 PMBUS 基地址表

| 实例 | 结构 | DriverLib 名称 | 基地址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| PmbusaRegs | PMBUS_REGS | PMBUSA_BASE | 0x0062_1000 | 是 | 是 | 是 |

## 23.4.2 PMBUS_REGS 寄存器

表中列出了 PMBUS_REGS 寄存器的存储器映射寄存器。表中未列出的所有寄存器偏移地址都应视为保留位置，不应修改寄存器内容。

表 23-2 PMBUS_REGS 寄存器

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 0x0 | PMBMC | PMBUS 主模式控制寄存器 |
| 0x4 | PMBTXBUF | PMBUS 发送缓冲器 |
| 0x8 | PMBRXBUF | PMBUS 接收缓冲器 |
| 0xC | PMBACK | PMBUS 确认寄存器 |
| 0x10 | PMBSTS | PMBUS 状态寄存器 |
| 0x14 | PMBINTM | PMBUS 中断屏蔽寄存器 |
| 0x18 | PMBSC | PMBUS 从模式配置寄存器 |

| 偏移量 | 缩写 | 寄存器名称 |
|---|---|---|
| 0x1C | PMBHSA | PMBUS 保持从机地址寄存器 |
| 0x20 | PMBCTRL | PMBUS 控制寄存器 |
| 0x24 | PMBTIMCTL | PMBUS 时序控制寄存器 |
| 0x28 | PMBTIMCLK | PMBUS 时钟时序寄存器 |
| 0x2C | PMBTIMSTSETUP | PMBUS 启动建立时间寄存器 |
| 0x30 | PMBTIMBIDLE | PMBUS 总线空闲时间寄存器 |
| 0x34 | PMBTIMLOWTIMOUT | PMBUS 时钟低电平超时值寄存器 |
| 0x38 | PMBTIMHIGHTIMOUT | PMBUS 时钟高电平超时值寄存器 |

表 23-3 PMBUS_REGS 访问类型代码

| 访问类型 | 代码 | 描述 |
|---|---|---|
| 读取类型 | R | R |
| | RC | RC |
| 写入类型 | W | W |
| 复位或默认值 | -n | 复位后的值或默认值 |
| 寄存器数组变量 | i,j,k,l,m,n | 当这些变量在寄存器名称、偏移量或地址中使用时，它们指的是寄存器数组中某个寄存器的值，该寄存器属于一组重复寄存器的一部分。这些寄存器组构成了一个层次结构，该数组用一个公式表示。 |
| | y | 当此变量在寄存器名称、偏移量或地址中使用时，它指的是寄存器数组中某个寄存器的值。 |

### 23.4.2.1 PMBMC 寄存器

偏移地址：0h　复位值：00000000h

**PMBUS 主模式控制寄存器**

表 23-4 PMBMC 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | PRC_CALL | GRP_CMD | PEC_ENA | EXT_CMD | CMD_ENA |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| BYTE_COUNT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SLAVE_ADDR | | | | | | | RW |
| R/W-0h | | | | | | | R/W-0h |

表 23-5 PMBMC 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-21 | RESERVED | R | 0h | 保留 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 20 | PRC_CALL | R/W | 0h | 0 = 除过程调用消息之外的所有消息的默认状态<br>1 = 使能过程调用消息的传输<br>复位类型：SYSRSn |
| 19 | GRP_CMD | R/W | 0h | 0 = 除组命令消息之外的所有消息的默认状态<br>1 = 使能组命令消息的传输<br>复位类型：SYSRSn |
| 18 | PEC_ENA | R/W | 0h | 0 = 禁用 PEC 处理<br>1 = 使能 PEC 字节传输/接收<br>复位类型：SYSRSn |
| 17 | EXT_CMD | R/W | 0h | 0 = 命令代码使用 1 个字节<br>1 = 命令代码使用 2 个字节<br>复位类型：SYSRSn |
| 16 | CMD_ENA | R/W | 0h | 0 = 禁用在主站启动的消息中使用命令代码<br>1 = 使能在主站启动的消息中使用命令代码<br>复位类型：SYSRSn |
| 15-8 | BYTE_COUNT | R/W | 0h | 指示当前消息中传输的数据字节数。字节计数不包括任何设备地址、命令字或块消息中的块长度。在块消息中，PMBus 接口会根据字节计数设置自动将块长度插入到消息中。固件只需加载要传输的地址、命令字和数据。PMBus 接口最多支持 255 个字节的字节写操作。<br>复位类型：SYSRSn |
| 7-1 | SLAVE_ADDR | R/W | 0h | 指定当前消息所指向的从机的地址。<br>复位类型：SYSRSn |
| 0 | RW | R/W | 0h | 0 = 消息为写事务（数据从主机到从机）<br>1 = 消息为读事务（数据从从机到主机）<br>复位类型：SYSRSn |

### 23.4.2.2 PMBTXBUF 寄存器

偏移地址：4h　复位值：00000000h

PMBUS 发送缓冲器

表 23-6 PMBTXBUF 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TXDATA | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TXDATA | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TXDATA | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXDATA | | | | | | | |
| R/W-0h | | | | | | | |

表 23-7 PMBTXBUF 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | TXDATA | R/W | 0h | 位 31-24: BYTE3 - 从发送数据缓冲器传输的最后一个数据字节位<br>23-16: BYTE2 - 从发送数据缓冲器传输的第三个数据字节位<br>15-8: BYTE1 - 从发送数据缓冲器传输的第二个数据字节位<br>7-0: BYTE0 - 从发送数据缓冲器传输的第一个数据字节<br>复位类型：SYSRSn |

### 23.4.2.3 PMBRXBUF 寄存器

偏移地址：8h 复位值：00000000h

**PMBUS 接收缓冲器**

表 23-8 PMBRXBUF 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| RXDATA | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| RXDATA | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| RXDATA | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXDATA | | | | | | | |
| R-0h | | | | | | | |

表 23-9 PMBRXBUF 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-0 | RXDATA | R | 0h | 位 31-24: BYTE3 - 接收数据缓冲器中接收到的最后一个数据字节位<br>23-16: BYTE2 - 接收数据缓冲器中接收到的第三个数据字节位<br>15-8: BYTE1 - 接收数据缓冲器中接收到的第二个数据字节位<br>7-0: BYTE0 - 接收数据缓冲器中接收的第一个数据字节<br>复位类型：SYSRSn |

### 23.4.2.4 PMBACK 寄存器

偏移地址：Ch 复位值：00000000h

PMBUS 确认寄存器

表 23-10 PMBACK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | ACK |
| R-0h | | | | | | | R/W-0h |

表 23-11 PMBACK 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | 保留 |
| 0 | ACK | R/W | 0h | 0 = NACK 接收到的数据<br>1 = 确认接收到的数据，在 PMBus 上发出 ACK 时位清零<br>复位类型：SYSRSn |

### 23.4.2.5 PMBSTS 寄存器

偏移地址：10h　复位值：00340000h

**PMBUS 状态寄存器**

表 23-12 PMBSTS 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | SCL_RAW | SDA_RAW | CONTROL_RAW | ALERT_RAW | CONTROL_EDGE | ALERT_EDGE |
| R-0h | | R-1h | R-1h | R-0h | R-1h | RC-0h | RC-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MASTER | LOST_ARB | BUS_FREE | UNIT_BUSY | RPT_START | SLAVE_ADDR_READY | CLK_HIGH_DETECTED | CLK_LOW_TIMEOUT |
| RC-0h | RC-0h | RC-0h | RC-0h | RC-0h | RC-0h | RC-0h | RC-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PEC_VALID | NACK | EOM | DATA_REQUEST | DATA_READY | RD_BYTE_COUNT | | |
| RC-0h | RC-0h | RC-0h | RC-0h | RC-0h | RC-0h | | |

表 23-13 PMBSTS 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-22 | RESERVED | R | 0h | 保留 |
| 21 | SCL_RAW | R | 1h | 0 = 观察到 PMBus 时钟引脚处于逻辑低电平<br>1 = 观察到 PMBus 时钟引脚处于逻辑高电平<br>复位类型：SYSRSn |
| 20 | SDA_RAW | R | 1h | 0 = 观察到 PMBus 数据引脚处于逻辑低电平<br>1 = 观察到 PMBus 数据引脚处于逻辑高电平<br>复位类型：SYSRSn |
| 19 | CONTROL_RAW | R | 0h | 0 = 观察到控制引脚处于逻辑低电平<br>1 = 观察到控制引脚处于逻辑高电平<br>复位类型：SYSRSn |
| 18 | ALERT_RAW | R | 1h | 0 = 观察到警报引脚处于逻辑低电平<br>1 = 观察到警报引脚处于逻辑高电平<br>复位类型：SYSRSn |
| 17 | CONTROL_EDGE | RC | 0h | 0 = 控制引脚未发生跳变<br>1 = 控制引脚已被 PMBus 上的另一个设备断言<br>复位类型：SYSRSn |
| 16 | ALERT_EDGE | RC | 0h | 0 = 警报引脚未发生跳变<br>1 = 警报引脚已被 PMBus 上的另一个设备断言<br>复位类型：SYSRSn |
| 15 | MASTER | RC | 0h | 0 = PMBus 接口处于从模式或空闲模式<br>1 = PMBus 接口处于主模式<br>复位类型：SYSRSn |
| 14 | LOST_ARB | RC | 0h | 0 = 主机已获得 PMBus 的控制权<br>1 = 主机失去仲裁并失去 PMBus 控制权<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 13 | BUS_FREE | RC | 0h | 0 = PMBus 正在处理当前消息<br>1 = PMBus 可用于新消息<br>复位类型：SYSRSn |
| 12 | UNIT_BUSY | RC | 0h | 0 = PMBus 接口空闲，准备好发送/接收消息<br>1 = PMBus 接口忙碌，正在处理当前消息<br>复位类型：SYSRSn |
| 11 | RPT_START | RC | 0h | 0 = 接口未收到重复起始条件<br>1 = 接口收到重复起始条件<br>复位类型：SYSRSn |
| 10 | SLAVE_ADDR_READY | RC | 0h | 0 = 表示没有可读取的从机地址<br>1 = 从机地址已准备好，可从接收数据寄存器（位 6:0）读取<br>复位类型：SYSRSn |
| 9 | CLK_HIGH_DETECTED | RC | 0h | 0 = 未检测到时钟高电平条件<br>1 = 消息期间时钟高电平超过 50us<br>复位类型：SYSRSn |
| 8 | CLK_LOW_TIMEOUT | RC | 0h | 0 = 未检测到时钟低电平超时<br>1 = 检测到时钟低电平超时，时钟保持低电平超过 35ms<br>复位类型：SYSRSn |
| 7 | PEC_VALID | RC | 0h | 0 = 接收到的 PEC 无效（如果 EOM 已断言）<br>1 = 接收到的 PEC 有效注意：PEC_VALID 状态在消息期间无关紧要。只有在 EOM 之后，此位才有有效值。<br>复位类型：SYSRSn |
| 6 | NACK | RC | 0h | 0 = 发送的数据已被接收方接受<br>1 = 接收方未接受发送的数据<br>复位类型：SYSRSn |
| 5 | EOM | RC | 0h | 0 = 消息仍在进行中或 PMBus 处于空闲状态。<br>1 = 检测到当前消息结束<br>复位类型：SYSRSn |
| 4 | DATA_REQUEST | RC | 0h | 0 = PMBus 接口不需要数据<br>1 = PMBus 接口请求额外数据。启用 PMBus 时钟拉伸以暂停总线<br>复位类型：SYSRSn |
| 3 | DATA_READY | RC | 0h | 0 = 处理器无可读取数据<br>1 = PMBus 接口读取缓冲器已满，固件需要在进一步总线活动之前读取数据。启用 PMBus 时钟拉伸以暂停总线，直到固件读取数据。<br>复位类型：SYSRSn |
| 2-0 | RD_BYTE_COUNT | RC | 0h | 0 = 无接收数据<br>1 = 接收到 1 个字节。数据位于接收数据寄存器的位 7-0<br>2 = 接收到 2 个字节。数据位于接收数据寄存器的位 15-0 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 3 = 接收到 3 个字节。数据位于接收数据寄存器的位 23-0<br>4 = 接收到 4 个字节。数据位于接收数据寄存器的位 31-0<br>复位类型：SYSRSn |

### 23.4.2.6 PMBINTM 寄存器

偏移地址：14h　复位值：000003FFh

**PMBUS 中断屏蔽寄存器**

表 23-14 PMBINTM 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | CLK_HIGH_DETECT | LOST_ARB |
| R-0h | | | | | | R/W-1h | R/W-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CONTROL | ALERT | EOM | SLAVE_ADDR_READY | DATA_REQUEST | DATA_READY | BUS_LOW_TIMEOUT | BUS_FREE |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h |

表 23-15 PMBINTM 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | 保留 |
| 9 | CLK_HIGH_DETECT | R/W | 1h | 0 = 如果消息期间时钟高电平超过 50us，则生成中断<br>1 = 禁用时钟高电平检测的中断生成<br>复位类型：SYSRSn |
| 8 | LOST_ARB | R/W | 1h | 0 = 在断言失去仲裁标志时生成中断<br>1 = 禁用在断言失去仲裁标志时生成中断<br>复位类型：SYSRSn |
| 7 | CONTROL | R/W | 1h | 0 = 在断言控制标志时生成中断<br>1 = 禁用在断言控制标志时生成中断<br>复位类型：SYSRSn |
| 6 | ALERT | R/W | 1h | 0 = 在断言警报标志时生成中断<br>1 = 禁用在断言警报标志时生成中断<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 5 | EOM | R/W | 1h | 0 = 在断言消息结束标志时生成中断<br>1 = 禁用在断言消息结束标志时生成中断<br>复位类型：SYSRSn |
| 4 | SLAVE_ADDR_READY | R/W | 1h | 0 = 在断言从机地址就绪标志时生成中断<br>1 = 禁用在断言从机地址就绪标志时生成中断<br>复位类型：SYSRSn |
| 3 | DATA_REQUEST | R/W | 1h | 0 = 在断言数据请求标志时生成中断<br>1 = 禁用在断言数据请求标志时生成中断<br>复位类型：SYSRSn |
| 2 | DATA_READY | R/W | 1h | 0 = 在断言数据就绪标志时生成中断<br>1 = 禁用在断言数据就绪标志时生成中断<br>复位类型：SYSRSn |
| 1 | BUS_LOW_TIMEOUT | R/W | 1h | 0 = 在断言时钟低电平超时标志时生成中断<br>1 = 禁用在断言时钟低电平超时标志时生成中断<br>复位类型：SYSRSn |
| 0 | BUS_FREE | R/W | 1h | 0 = 在断言总线空闲标志时生成中断<br>1 = 禁用在断言总线空闲标志时生成中断<br>复位类型：SYSRSn |

### 23.4.2.7 PMBSC 寄存器

偏移地址：18h　复位值：00607F7Ch

表 23-16 PMBSC 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | RX_BYTE_ACK_CNT | | MAN_CMD | TX_PEC | TX_COUNT | | |
| R-0h | R/W-3h | | R/W-0h | R/W-0h | R/W-0h | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| PEC_ENA | SLAVE_MASK | | | | | | |
| R/W-0h | R/W-7Fh | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAN_SLAVE_ACK | SLAVE_ADDR | | | | | | |
| R/W-0h | R/W-7Ch | | | | | | |

表 23-17 PMBSC 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-23 | RESERVED | R | 0h | 保留 |
| 22-21 | RX_BYTE_ACK_CNT | R/W | 3h | 配置在从模式下接收数据时自动确认的数据字节数。00 = 从机接收到 1 个字节。固件需手动确认每个接收到的字节。01 = 从机接收到 2 个字节。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 硬件自动确认第一个接收到的字节。固件需在接收到第二个字节后手动确认。10 = 从机接收到 3 个字节。硬件自动确认前 2 个接收到的字节。固件需在接收到第三个字节后手动确认。11 = 从机接收到 4 个字节。硬件自动确认前 3 个接收到的字节。固件需在接收到第四个字节后手动确认。复位类型：SYSRSn |
| 20 | MAN_CMD | R/W | 0h | 0 = 从机自动确认接收到的命令代码<br>1 = 接收到命令代码后生成数据请求标志，固件需发出 ACK 以继续消息<br>复位类型：SYSRSn |
| 19 | TX_PEC | R/W | 0h | 在从机需要在消息末尾发送 PEC 字节时断言。在发送完 TX Byte Cnt（位 18:16）指示的数据字节数后，PMBus 接口将发送计算出的 PEC 字节。<br>复位类型：SYSRSn |
| 18-16 | TX_COUNT | R/W | 0h | 0 = 无有效字节<br>1 = 一个有效字节，字节 #0（发送数据寄存器的位 7:0）<br>2 = 两个有效字节，字节 #0 和 #1（发送数据寄存器的位 15:0）<br>3 = 三个有效字节，字节 #0-2（发送数据寄存器的位 23:0）<br>4 = 四个有效字节，字节 #0-3（发送数据寄存器的位 31:0）<br>复位类型：SYSRSn |
| 15 | PEC_ENA | R/W | 0h | 0 = PEC 处理禁用1 = PEC 处理使能复位类型：SYSRSn |
| 14-8 | SLAVE_MASK | R/W | 7Fh | 用于地址检测，从机掩码使能从机能确认多个设备地址。将从机掩码中的某个位写 ‘0’ 会使能从机地址中的对应位为 ‘1’ 或 ‘0’ 时都能匹配。将所有掩码位写 ‘0’ 可使 PMBus 接口确认任何设备地址。上电时，从机掩码默认为 7Fh，表示从机只会确认编程到从机地址（位 6-0）中的地址。<br>复位类型：SYSRSn |
| 7 | MAN_SLAVE_ACK | R/W | 0h | 0 = 从机自动确认 SLAVE_ADDR（位 6:0）中指定的设备地址1 = 启用手动从机地址确认模式。固件需要在每条消息中读取接收到的地址并进行确认注意：当 PMBCTRL 寄存器的位 31 (I2C_mode) 置位时，建议仅使用手动确认从机地址 (MAN_SLAVE_ACK =1)。<br>复位类型：SYSRSn |
| 6-0 | SLAVE_ADDR | R/W | 7Ch | 配置从机的当前设备地址。用于自动从机地址确认模式（默认模式）。PMBus 接口会将接收到的设备地址与存储在从机地址位中的值以及从机掩码位中配置的掩码进行比较。如果匹配，从机将确认设备地址。 |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| | | | | 复位类型：SYSRSn |

### 23.4.2.8 PMBHSA 寄存器

偏移地址：1Ch　复位值：00000000h

**PMBUS 保持从机地址寄存器**

表 23-18 PMBHSA 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SLAVE_ADDR | | | | | | | SLAVE_RW |
| R-0h | | | | | | | R-0h |

表 23-19 PMBHSA 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | 保留 |
| 7-1 | SLAVE_ADDR | R | 0h | 存储的、已被从机确认的设备地址<br>复位类型：SYSRSn |
| 0 | SLAVE_RW | R | 0h | 存储的、来自已被从机确认的地址的 R/W 位<br>0 = 写访问<br>1 = 读访问<br>复位类型：SYSRSn |

### 23.4.2.9 PMBCTRL 寄存器

偏移地址：20h　复位值：00200000h

**PMBUS 控制寄存器**

表 23-20 PMBCTRL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| I2CMODE | 保留 | | | CLKDIV | | | |
| R/W-0h | R-0h | | | R/W-0h | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CLKDIV | MASTER_EN | SLAVE_EN | CLK_LO_DIS | 保留 | 保留 | SCL_DIR | SCL_VALUE |
| R/W-0h | R/W-0h | R/W-1h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SCL_MODE | SDA_DIR | SDA_VALUE | SDA_MODE | CNTL_DIR | CNTL_VALUE | CNTL_MODE | ALERT_DIR |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ALERT_VALUE | ALERT_MODE | CNTL_INT_EDGE | 保留 | FAST_MODE | BUS_LO_INT_EDGE | ALERT_EN | RESET |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 23-21 PMBCTRL 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31 | I2CMODE | R/W | 0h | 0 = PMBUS 模式1 = I2C 模式<br>复位类型：SYSRSn |
| 30-28 | RESERVED | R | 0h | 保留 |
| 27-23 | CLKDIV | R/W | 0h | PMBUS 发送/接收 FSM (FSM_CLK) 的时钟是 SYSCLK 时钟的分频版本。频率(FSM_CLK) = 频率(SYSCLK)/(CLKDIV+1)注意：FSM_CLK 应小于或等于 10MHz。复位类型：SYSRSn |
| 22 | MASTER_EN | R/W | 0h | 0 = 禁用 PMBus 主机功能<br>1 = 启用 PMBus 主机功能<br>复位类型：SYSRSn |
| 21 | SLAVE_EN | R/W | 1h | 0 = 禁用 PMBus 从机功能<br>1 = 启用 PMBus 从机功能<br>复位类型：SYSRSn |
| 20 | CLK_LO_DIS | R/W | 0h | 0 = 时钟低电平超时启用<br>1 = 时钟低电平超时禁用<br>复位类型：SYSRSn |
| 19 | RESERVED | R/W | 0h | 保留 |
| 18 | RESERVED | R/W | 0h | 保留 |
| 17 | SCL_DIR | R/W | 0h | 0 = PMBus 时钟引脚配置为输出<br>1 = PMBus 时钟引脚配置为输入<br>复位类型：SYSRSn |
| 16 | SCL_VALUE | R/W | 0h | 0 = PMBus 时钟引脚在 GPIO 模式下驱动为低电平<br>1 = PMBus 时钟引脚在 GPIO 模式下驱动为高电平<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 15 | SCL_MODE | R/W | 0h | 0 = PMBus 时钟引脚配置为功能模式<br>1 = PMBus 时钟引脚配置为 GPIO<br>复位类型：SYSRSn |
| 14 | SDA_DIR | R/W | 0h | 0 = PMBus 数据引脚配置为输出<br>1 = PMBus 数据引脚配置为输入<br>复位类型：SYSRSn |
| 13 | SDA_VALUE | R/W | 0h | 0 = PMBus 数据引脚在 GPIO 模式下驱动为低电平<br>1 = PMBus 数据引脚在 GPIO 模式下驱动为高电平<br>复位类型：SYSRSn |
| 12 | SDA_MODE | R/W | 0h | 0 = PMBus 数据引脚在 GPIO 模式下驱动为低电平<br>1 = PMBus 数据引脚在 GPIO 模式下驱动为高电平<br>复位类型：SYSRSn |
| 11 | CNTL_DIR | R/W | 0h | 0 = 控制引脚配置为输出<br>1 = 控制引脚配置为输入<br>复位类型：SYSRSn |
| 10 | CNTL_VALUE | R/W | 0h | 0 = 控制引脚在 GPIO 模式下驱动为低电平<br>1 = 控制引脚在 GPIO 模式下驱动为高电平复位类型：SYSRSn |
| 9 | CNTL_MODE | R/W | 0h | 0 = 控制引脚配置为功能模式（默认）<br>1 = 控制引脚配置为 GPIO<br>复位类型：SYSRSn |
| 8 | ALERT_DIR | R/W | 0h | 0 = 警报引脚配置为输出<br>1 = 警报引脚配置为输入<br>复位类型：SYSRSn |
| 7 | ALERT_VALUE | R/W | 0h | 0 = 警报引脚在 GPIO 模式下驱动为低电平<br>1 = 警报引脚在 GPIO 模式下驱动为高电平<br>复位类型：SYSRSn |
| 6 | ALERT_MODE | R/W | 0h | 0 = 警报引脚配置为功能模式<br>1 = 警报引脚配置为 GPIO<br>复位类型：SYSRSn |
| 5 | CNTL_INT_EDGE | R/W | 0h | 0 = 在控制信号的下降沿生成中断<br>1 = 在控制信号的上升沿生成中<br>断复位类型：SYSRSn |
| 4 | RESERVED | R/W | 0h | 保留 |
| 3 | FAST_MODE | R/W | 0h | 0 = 标准 100 KHz 模式启用<br>1 = 快速模式启用（PMBus 上 400KHz 操作）<br>复位类型：SYSRSn |
| 2 | BUS_LO_INT_EDGE | R/W | 0h | 0 = 在时钟低电平超时的上升沿生成中断<br>1 = 在时钟低电平超时的下降沿生成中断<br>复位类型：SYSRSn |

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 1 | ALERT_EN | R/W | 0h | 0 = PMBus 警报不由从机驱动，在 PMBus 上拉高<br>1 = PMBus 警报由从机驱动为低电平<br>复位类型：SYSRSn |
| 0 | RESET | R/W | 0h | 0 = 不复位内部状态机（默认）<br>1 = 控制状态机复位到初始状态注意：软件复位后应通过读取状态寄存器 PMBSTS 来显式清除该寄存器，因为软件复位不会清除此位。<br>复位类型：SYSRSn |

## 23.4.2.10 PMBTIMCTL 寄存器

偏移地址：24h　复位值：00000000h

**PMBUS 时序控制寄存器**

表 23-22 PMBTIMCTL 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | TIM_OVERRIDE |
| R-0h | | | | | | | R/W-0h |

表 23-23 PMBTIMCTL 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-1 | RESERVED | R | 0h | 保留 |
| 0 | TIM_OVERRIDE | R/W | 0h | 0 PMBUS 状态机使用时序参数的默认设置。<br>1 PMBUS 状态机将使用以下寄存器中的设置：* PMBTIMCLK* PMBTIMSTSETUP* PMBTIMBIDLE* PMBTIMLOWTIMOUT* PMBTIMHIGHTIMOUT<br>复位类型：SYSRSn |

## 23.4.2.11 PMBTIMCLK 寄存器

偏移地址：28h　复位值：0060002Fh

**PMBUS 时钟时序寄存器**

表 23-24 PMBTIMCLK 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CLK_FREQ | | | | | | | |
| R/W-60h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLK_HIGH_LIMIT | | | | | | | |
| R/W-2Fh | | | | | | | |

表 23-25 PMBTIMCLK 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-24 | RESERVED | R | 0h | 保留 |
| 23-16 | CLK_FREQ | R/W | 60h | 定义 PMBUS 主时钟周期内 PMBUS FSM 输入时钟的数量。一个时钟周期内的 FSM 时钟数 = (CLK_FREQ+4)<br>复位类型：SYSRSn |
| 15-8 | RESERVED | R | 0h | 保留 |
| 7-0 | CLK_HIGH_LIMIT | R/W | 2Fh | 定义 PMBUS 主时钟高脉冲内的 PMBUS FSM 输入时钟数量。一个时钟高脉冲内的 FSM 时钟数 = (CLK_HIGH_LIMIT+3)<br>复位类型：SYSRSn |

### 23.4.2.12 PMBTIMSTSETUP 寄存器

偏移地址：2Ch　复位值：0000002Fh

**PMBUS 启动建立时间寄存器**

表 23-26 PMBTIMSTSETUP 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TSU_STA | | | | | | | |
| R/W-2Fh | | | | | | | |

表 23-27 PMBTIMSTSETUP 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-8 | RESERVED | R | 0h | 保留 |
| 7-0 | TSU_STA | R/W | 2Fh | 确定 PMBUS 主时钟最后一个上升沿与下一个启动沿之间的建立时间，TSU_STA 值以 PMBUS FSM 时钟周期数定义该建立时间。<br>复位类型：SYSRSn |

### 23.4.2.13 PMBTIMBIDLE 寄存器

偏移地址：30h　复位值：000001F3h

表 23-28 PMBTIMBIDLE 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | BUSIDLE | |
| R-0h | | | | | | R/W-1F3h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BUSIDLE | | | | | | | |
| R/W-1F3h | | | | | | | |

表 23-29 PMBTIMBIDLE 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | 保留 |
| 9-0 | BUSIDLE | R/W | 1F3h | 确定 PMBUS 时钟和数据均为高电平的持续时间，以判定总线处于空闲状态。BUSIDLE 值以 PMBUS FSM 时钟周期数表示。<br>复位类型：SYSRSn |

## 23.4.2.14 PMBTIMLOWTIMOUT 寄存器

偏移地址：34h 复位值：0005572Fh

PMBUS 时钟低电平超时值寄存器

表 23-30 PMBTIMLOWTIMOUT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | CLKLOWTIMOUT | | | |
| R-0h | | | | R/W-0005572Fh | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| CLKLOWTIMOUT | | | | | | | |
| R/W-0005572Fh | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLKLOWTIMOUT | | | | | | | |
| R/W-0005572Fh | | | | | | | |

表 23-31 PMBTIMLOWTIMOUT 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-20 | RESERVED | R | 0h | 保留 |
| 19-0 | CLKLOWTIMOUT | R/W | 0005572Fh | 确定 PMBUS 时钟处于低电平的持续时间，如果超过此时间，将导致时钟低电平超时条件。CLKLOWTIMOUT 值以 PMBUS FSM 时钟周期数表示。<br>复位类型：SYSRSn |

## 23.4.2.15 PMBTIMHIGHTIMOUT 寄存器

偏移地址：38h 复位值：000001F3h

PMBUS 时钟高电平超时值寄存器

表 23-32 PMBTIMHIGHTIMOUT 寄存器位布局表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| 保留 | | | | | | CLKHIGHTIMOUT | |
| R-0h | | | | | | R/W-1F3h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLKHIGHTIMOUT | | | | | | | |
| R/W-1F3h | | | | | | | |

表 23-33 PMBTIMHIGHTIMOUT 寄存器

| 位 | 字段 | 类型 | 复位 | 描述 |
|---|---|---|---|---|
| 31-10 | RESERVED | R | 0h | 保留 |
| 9-0 | CLKHIGHTIMOUT | R/W | 1F3h | 确定 PMBUS 时钟处于高电平的持续时间，如果超过此时间，将导致时钟高电平超时条件。<br>CLKHIGHTIMOUT 值以 PMBUS FSM 时钟周期数表示。<br>复位类型：SYSRSn |

# 24 串行通信接口（SCI）

## 24.1 简介

本章介绍串行通信接口（SCI）模块的特性及工作原理。SCI 是一种双线异步串行端口，通常被称为UART。SCI 模块支持 CPU 与其他采用标准不归零（NRZ）格式的异步外设之间进行数字通信。SCI 的接收器和发送器各配备一个 16 级深度的 FIFO（先进先出）缓冲区，以减少服务开销，且各自拥有独立的使能位和中断位。两者可独立工作以实现半双工通信，也可同时工作以实现全双工通信。

为确保数据完整性，SCI 会对接收的数据进行间断检测、奇偶校验、溢出和帧错误检查。通过 16 位波特率选择寄存器，可将波特率编程为不同的速率。

### 24.1.1 特性

SCI 模块具有以下特性：

- 两个外部引脚（若不用于 SCI，均可作为 GPIO 使用）：
  - SCITXD：SCI 发送输出引脚
  - SCIRXD：SCI 接收输入引脚
- 支持 65,536 种可编程波特率设置
- 数据字格式：
  - 1 个起始位
  - 数据字长度可在 1-8 位之间编程配置
  - 可选偶校验/奇校验/无校验位
  - 1 个或 2 个停止位
  - 额外的地址区分位（仅地址位模式下）
- 四种错误检测标志：奇偶校验错误、溢出错误、帧错误和间断检测
- 两种多处理器唤醒模式：空闲线模式和地址位模式
- 半双工或全双工操作
- 双缓冲接收和发送功能

- 发送器和接收器操作可通过带状态标志的中断驱动算法或查询算法实现
- 发送器和接收器具有独立的中断使能位（BRKDT 间断检测中断除外）
- 不归零（NRZ）格式

增强特性包括：

- 自动波特率检测硬件逻辑
- 16 级发送/接收 FIFO 缓冲区

### 24.1.2 CPU 接口

SCI CPU 接口如图 24-1 所示；端口工作方式由本章寄存器配置与控制。

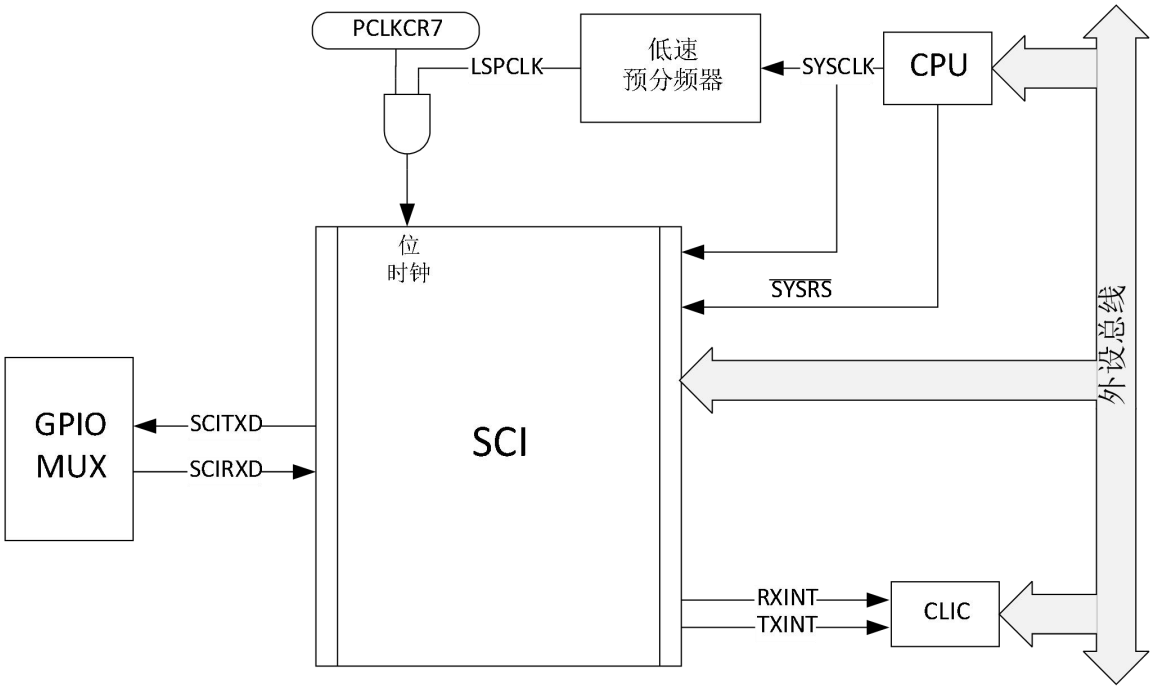



图 24-1 SCI CPU 接口

## 24.2 架构

全双工操作中使用的主要组件如图 24-2 所示，包括：

1) 发送器（TX）及主要寄存器：

- SCITXBUF - 发送数据缓冲寄存器，存储待发送的 CPU 加载数据；
- TXSHF 寄存器 - 发送移位寄存器，从 SCITXBUF 寄存器接收数据，并将数据逐位移位至 SCITXD 引脚；

2) 接收器（RX）及主要寄存器：

- RXSHF 寄存器 - 接收移位寄存器，从 SCIRXD 引脚逐位移入数据；
- SCIRXBUF - 接收数据缓冲寄存器，存储供 CPU 读取的数据。来自远程处理器的数据先加载至 RXSHF 寄存器，再传入 SCIRXBUF；SCIRXEMU 只是 SCIRXBUF 的仿真别名，不是独立缓冲区。

3) 可编程波特率发生器；

4) 控制和状态寄存器；

SCI 的接收器和发送器可独立工作，也可同时工作。

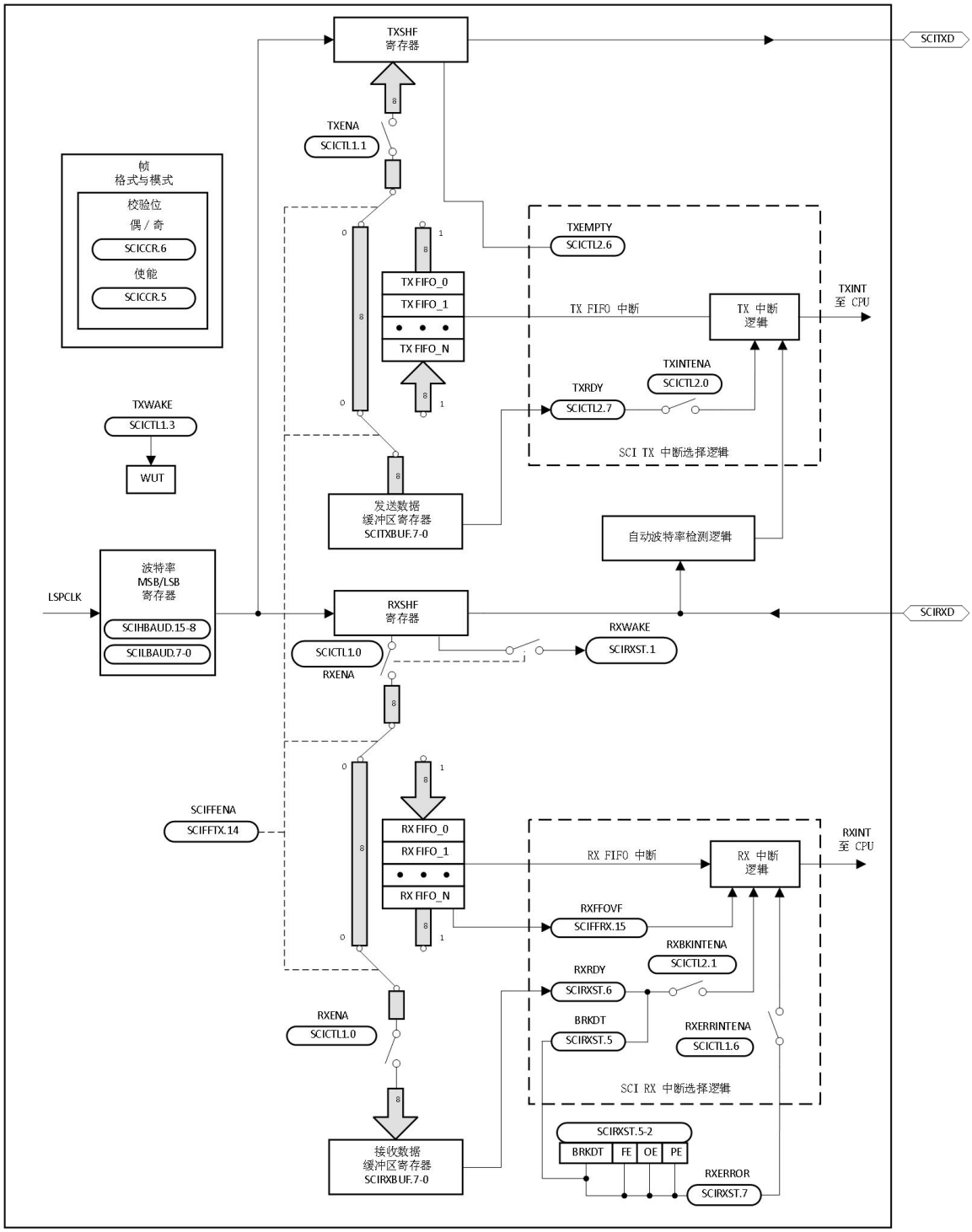



图 24-2 SCI 模块框图

## 24.3 SCI 模块信号汇总

SCI 各信号名称的汇总说明如表 24-1 所示。

表 24-1 SCI 模块信号汇总

| 信号类型 | 信号名称 | 描述 |
|---|---|---|
| 外部信号 | SCIRXD | SCI异步串行端口接收数据 |
| | SCITXD | SCI异步串行端口发送数据 |
| 控制信号 | 波特时钟（Baud clock） | 低速外设时钟（LSPCLK）预分频后的时钟 |
| 中断信号 | TXINT | 发送中断 |
| | RXINT | 接收中断 |

## 24.4 多处理器和异步通信模式

SCI 支持两种多处理器协议：空闲线多处理器模式和地址位多处理器模式。这些协议支持多个处理器之间高效的数据传输。

SCI 提供通用异步收发器（UART）通信模式，可与多种常用外设接口。数据传输特性包括：

- 1 个起始位
- 1-8 个数据位
- 1 个偶校验/奇校验位或无校验位
- 1 个或 2 个停止位

## 24.5 SCI 可编程数据格式

SCI 的接收和发送数据均采用不归零（NRZ）格式。如图 24-3 所示，NRZ 数据格式包括：

- 1 个起始位
- 1-8 个数据位
- 1 个可选的偶校验/奇校验位
- 1 个或 2 个停止位
- 1 个额外的地址区分位（仅地址位模式下）

数据的基本单位称为字符，长度为 1-8 位。每个数据字符均包含起始位、1 个或 2 个停止位，以及可选的校验位和地址位。带有格式信息的数据字符称为帧，如图 24-3 所示。




图 24-3 典型 SCI 数据帧格式

可通过 SCICCR 寄存器对数据格式进行编程。用于编程数据格式的位如表 24-2 所示。

表 24-2 通过 SCICCR 寄存器编程数据格式

| 位 | 位名称 | 标识 | 功能 |
|---|---|---|---|
| 2-0 | SCICHAR | SCICCR.2:0 | 选择字符（数据）长度（1-8位） |
| 5 | PARITYENA（ENABLE） | SCICCR.5 | 置1时启用校验功能，清0时禁用校验功能 |
| 6 | PARITY（EVEN/ODD） | SCICCR.6 | 校验使能时，清0选择奇校验，置1选择偶校验 |
| 7 | STOPBITS | SCICCR.7 | 确定发送的停止位数量：清0为1个停止位，置1为2个停止位 |

## 24.6 SCI 多处理器通信

多处理器通信格式允许一个处理器通过同一串行链路向其他处理器高效发送数据块。一条串行总线上同一时间只能进行一次传输，即同一时间只能有一个发送方。

（1）地址字节

发送方发送的信息块的第一个字节为地址字节，所有接收方都会读取该字节。只有地址匹配的接收方才会被地址字节后的数据流触发中断；地址不匹配的接收方则保持未进入中断的状态，直至下一个地址字节出现。

（2）睡眠位

串行链路上的所有处理器需将 SCI 的 SLEEP 位（SCICTL1 寄存器的第 2 位）置 1，使处理器仅在检测到地址字节时产生中断。当处理器读取到与应用程序设置的 CPU 设备地址一致的块地址时，程序必须将 SLEEP 位清 0，使 SCI 在接收到每个数据字节时都能产生中断。

即使 SLEEP 位为 1，接收器仍会工作，但除非检测到地址字节且接收帧中的地址位为 1（仅适用于地址位模式），否则接收器不会将 RXRDY、RXINT 或任何接收错误状态位置 1。SCI 不会自动修改 SLEEP 位，需通过软件进行操作。

### 24.6.1 地址字节识别

处理器对地址字节的识别方式因所采用的多处理器模式而异：

空闲线模式在地址字节前留有一段空闲时间，该模式无额外的地址/数据位，处理超过 10 字节的数据块时比地址位模式更高效，典型的非多处理器 SCI 通信需使用空闲线模式。

地址位模式在每个字节中添加一个额外的位（即地址位）来区分地址和数据，处理多个小数据块时更高效，因为与空闲模式不同，该模式无需在数据块之间等待。但在高传输速率下，程序可能无法及时避免传输流中出现 10 位的空闲状态。

### 24.6.2 SCI 发送和接收功能控制

通过 ADDRIDLE_MODE 位（SCICCR 寄存器的第 3 位）可软件选择多处理器模式。两种模式均使用 TXWAKE 标志位（SCICTL1 寄存器的第 3 位）、RXWAKE 标志位（SCIRXST 寄存器的第 1 位）和 SLEEP 标志位（SCICTL1 寄存器的第 2 位）来控制 SCI 发送器和接收器的相关功能。

### 24.6.3 接收流程

两种多处理器模式下的接收流程如下：

1. 接收到地址块后，SCI 端口唤醒并请求中断（非 FIFO 模式下，需使能 SCICTL2 寄存器的第 1 位 RXBKINTENA 以请求中断；FIFO 模式下，需通过

SCIFFRX 寄存器的 RXFFIENA 位使能该功能，并将同一寄存器的 RXFFIL 字段设置为 1）。SCI 读取该块的第一帧（包含目标地址）。

2. 通过中断进入软件程序，检查传入的地址，并将该地址字节与存储在内存中的设备地址字节进行比对。

3. 若比对结果显示该数据块是发送至本设备 CPU 的，则 CPU 将 SLEEP 位清 0 并读取后续数据块；若不匹配，则软件程序保持 SLEEP 位为 1 并退出，直至检测到下一个块起始信号前不再接收中断。

## 24.7 空闲线多处理器模式

在空闲线多处理器协议（ADDRIDLE_MODE 位=0）中，数据块之间的空闲时间长于块内帧之间的空闲时间，以此区分不同数据块。一帧之后出现 10 个或更多高电平位的空闲时间，表示新数据块的开始。单个位的时间由波特率（比特率）直接计算得出。空闲线多处理器通信格式如图 24-4 所示（ADDRIDLE_MODE 位为 SCICCR 寄存器的第 3 位）。

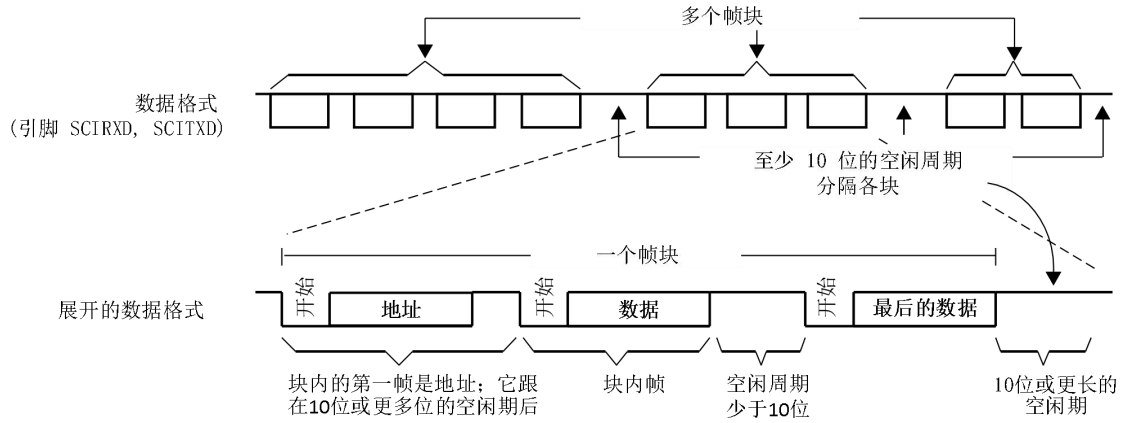



图 24-4 空闲线多处理器通信格式

### 24.7.1 空闲线模式步骤

空闲线模式的工作步骤如下：

1. SCI 接收到块起始信号后唤醒。

2. 处理器识别到下一个 SCI 中断。

3. 中断服务程序将接收到的地址（由远程发送器发送）与 ISR（中断服务程序）地址进行比对。

4. 若本 CPU 是目标地址，则服务程序将 SLEEP 位清 0 并接收后续数据块。

5. 若本 CPU 不是目标地址，则 SLEEP 位保持置 1 状态，CPU 继续执行主程序，直至检测到下一个块起始信号前不会被 SCI 端口中断。

**注意：**

在空闲线模式下，若 SCI 从 FIFO 中读取所有 RXDATA 的时间超过 10 个位周期，则可能错过即将到来的块起始信号检测。

当 SCI 识别到 10 个位周期的空闲状态时，RXWAKE 逻辑仅触发一次；即使读取 SCIRXBUF 后（会清除唤醒状态）线路仍保持空闲，RXWAKE 逻辑也不会再次触发。

因此，若 ISR 通过 SCIRXBUF 读取 FIFO 中所有 RXDATA 的时间超过 10 个位周期，SCI 可能无法检测到下一个块起始信号。该情况在 FIFO 模式和非 FIFO 模式下均适用，只要 CPU 从 SCIRXBUF/FIFO 读取数据的时间超过 SCI 的 10 个位周期。

为避免此问题，建议采用以下任一方法：

在 ISR 结束时，读取完所有 RX 数据后，对 SCICTL1.SWRESET 位执行一次 0→1 的软件复位序列。

读取 SCIRXBUF 寄存器前先检查 RXWAKE 状态位，若 RXWAKE 置 1，则在 ISR 结束时不为 RX 设置 SLEEP 位。

### 24.7.2 块起始信号发送方法

发送块起始信号有两种方式：

1. 通过延迟前一个数据块最后一帧的发送时间与新数据块地址帧的发送时间之间的间隔，刻意留出 10 个或更多位的空闲时间。

2. SCI 端口在写入 SCITXBUF 寄存器前，先将 TXWAKE 位（SCICTL1 寄存器的第 3 位）置 1，这样会发送一个恰好 11 个位的空闲时间。该方法可使串行通信

线路的空闲时间最短（需在设置 TXWAKE 后、发送地址前，向 SCITXBUF 写入一个无关字节，以触发空闲时间的发送）。

### 24.7.3 临时唤醒（WUT）标志

与 TXWAKE 位相关联的是临时唤醒（WUT）标志。WUT 是一个内部标志，与 TXWAKE 形成双缓冲结构。当 TXSHF 从 SCITXBUF 加载数据时，WUT 从 TXWAKE 加载值，同时 TXWAKE 位清 0。该结构如图 24-5 所示。

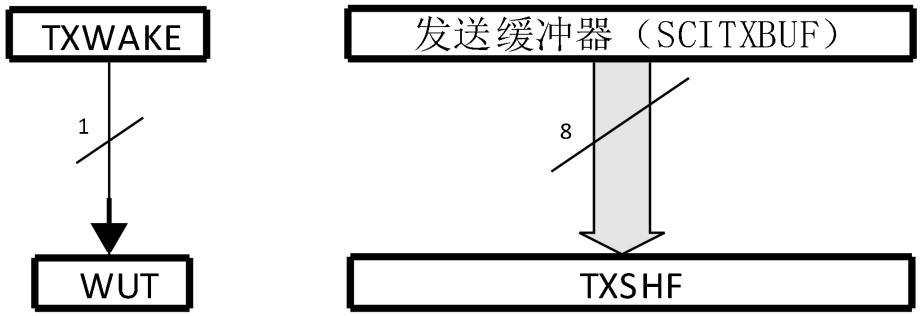



图 24-5 双缓冲 WUT 和 TXSHF 结构

#### 24.7.3.1 块起始信号发送流程

在数据块传输序列中发送一个宽度为 11 位的块起始信号，步骤如下：

1. 将 TXWAKE 位置 1。

2. 向 SCITXBUF 寄存器（发送数据缓冲区）写入一个数据字（与内容无关，即“无关字节”），以发送块起始信号（写入的第一个数据字在块起始信号发送期间会被抑制，之后将被忽略）。当 TXSHF（发送移位寄存器）再次空闲时，SCITXBUF 的内容会移位至 TXSHF，TXWAKE 的值会移位至 WUT，随后 TXWAKE 清 0。由于 TXWAKE 被置 1，前一帧的最后一个停止位之后会发送一个 11 个位的空闲周期，替代起始位、数据位和校验位。

3. 向 SCITXBUF 写入新的地址值。必须先向 SCITXBUF 寄存器写入一个无关数据字，使 TXWAKE 位的值能移位至 WUT。当无关数据字移位至 TXSHF 寄存器后，可再次写入 SCITXBUF（必要时也可写入 TXWAKE），因为 TXSHF 和 WUT 均为双缓冲结构。

### 24.7.4 接收器操作

无论 SLEEP 位状态如何，接收器都会持续工作。但除非检测到地址帧，否则接收器不会设置 RXRDY 位和错误状态位，也不会请求接收中断。

## 24.8 地址位多处理器模式

在地址位协议（ADDRIDLE_MODE 位=1）中，每个帧都包含一个额外的地址位，紧跟在最后一个数据位之后。数据块第一帧的地址位置 1，其余帧的地址位置 0，空闲周期的时序无关紧要（如图 24-6 所示）。

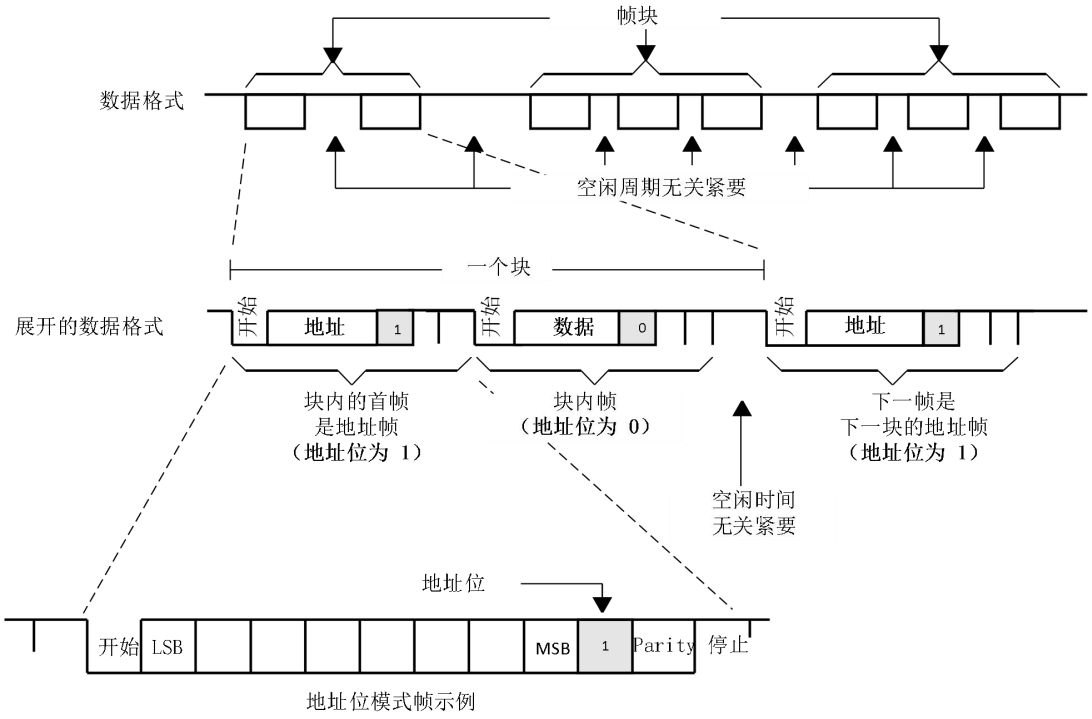



图 24-6 地址位多处理器通信格式

### 24.8.1 地址发送流程

TXWAKE 位的值会存入地址位。传输过程中，当 SCITXBUF 寄存器和 TXWAKE 分别加载至 TXSHF 寄存器和 WUT 时，TXWAKE 复位为 0，WUT 成为当前帧的地址位值。因此，发送地址的步骤如下：

1. 将 TXWAKE 位置 1，并向 SCITXBUF 寄存器写入相应的地址值。当该地址值传输至 TXSHF 寄存器并移位输出时，地址位会以 1 的形式发送，通知串行链路上的其他处理器读取该地址。

2. TXSHF 和 WUT 加载完成后，可再次写入 SCITXBUF 和 TXWAKE（由于 TXSHF 和 WUT 均为双缓冲结构，可立即写入）。

3. 保持 TXWAKE 位为 0，以发送数据块中的非地址帧。

**注意：**

一般而言，地址位格式通常用于 11 字节及以下的数据帧。该格式会在所有传输的数据字节中添加一个位（地址帧为 1，数据帧为 0）。空闲线格式通常用于 12 字节及以上的数据帧。

## 24.9 SCI 通信格式

SCI 异步通信格式支持单线（单向）或双线（双向）通信。该模式下，帧结构包括 1 个起始位、1-8 个数据位、1 个可选的偶校验/奇校验位，以及 1 个或 2 个停止位（如图 24-7 所示）。每个数据位对应 8 个 SCICLK 周期。

接收器在检测到有效起始位后开始工作。有效起始位定义为 4 个连续的内部 SCICLK 周期的低电平（如图 24-7 所示）。若其中任一位不为低电平，则处理器重新开始寻找下一个起始位。

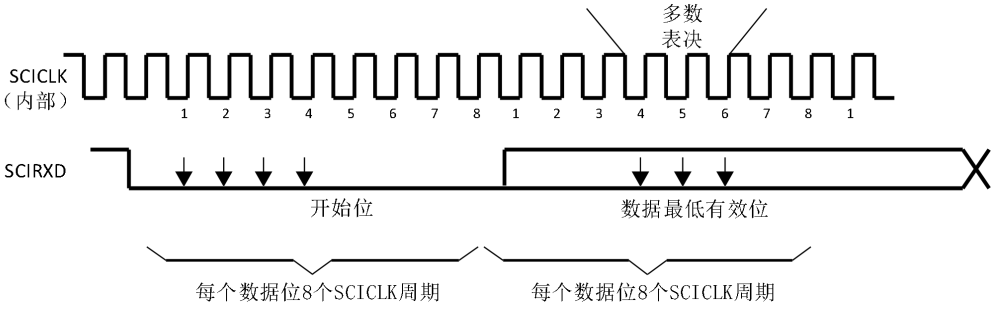



图 24-7 SCI 异步通信格式

对于起始位之后的位，处理器通过在位中间进行三次采样（分别在第 4、5、6 个 SCICLK 周期），并基于多数表决（三取二）原则确定位值。

由于接收器与帧同步，外部发送和接收设备无需使用同步串行时钟，时钟可本地生成。

### 24.9.1 通信模式下的接收器信号

图 24-8 展示了接收器信号时序示例，基于以下假设条件：

- 地址位唤醒模式（空闲线模式下无地址位）
- 每个字符 6 位

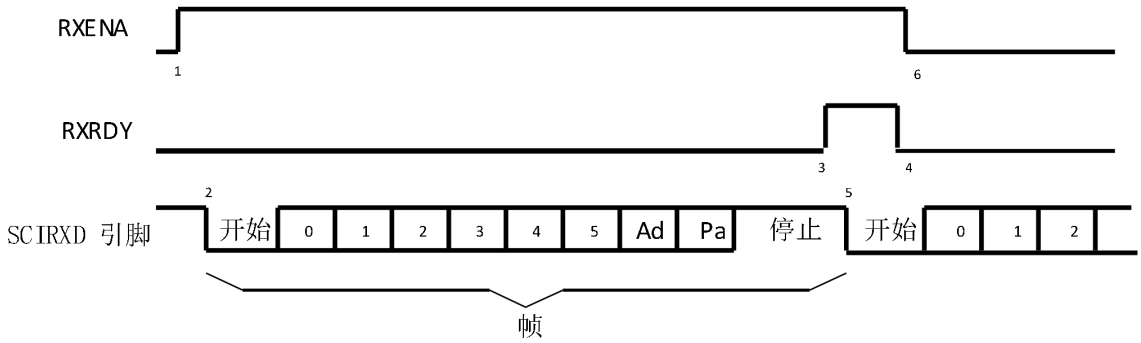



图 24-8 SCI 通信模式下的接收信号时序

说明：

1. RXENA 位（SCICTL1 寄存器的第 0 位）置高，使能接收器。

2. 数据通过 SCIRXD 引脚传入，检测到起始位。

3. 数据从 RXSHF 移位至接收缓冲寄存器（SCIRXBUF），请求中断。RXRDY 标志位（SCIRXST 寄存器的第 6 位）置高，指示已接收到新字符。

4. 程序读取 SCIRXBUF，RXRDY 标志位自动清 0。

5. 下一个字节的数据通过 SCIRXD 引脚传入，检测到起始位后起始位检测标志清 0。

6. RXENA 位置低，禁用接收器。数据仍会在 RXSHF 中组装，但不会传输至接收缓冲寄存器。

### 24.9.2 通信模式下的发送器信号

图 24-9 展示了发送器信号时序示例，基于以下假设条件：

- 地址位唤醒模式（空闲线模式下无地址位）

- 每个字符 3 位数据

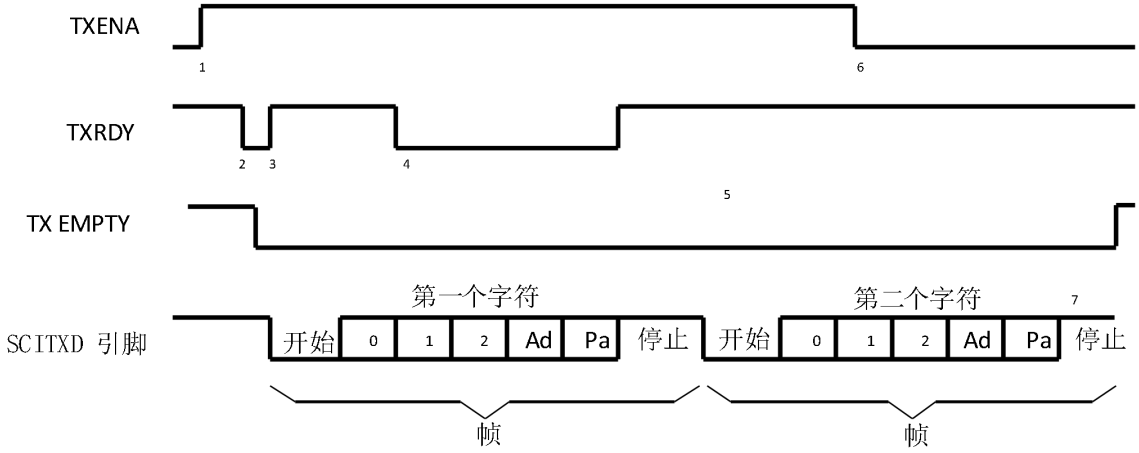



图 24-9 SCI 通信模式下的发送信号时序

说明：

1. TXENA 位（SCICTL1 寄存器的第 1 位）置高，使能发送器发送数据。

2. 向 SCITXBUF 写入数据，此时：（1）发送器不再为空；（2）TXRDY 位清低。

3. SCI 将数据传输至移位寄存器（TXSHF），发送器准备接收第二个字符（TXRDY 位置高），并请求中断（需设置 SCICTL2 寄存器的第 0 位 TXINTENA 以启用中断）。

4. TXRDY 位置高后（步骤 3），程序向 SCITXBUF 写入第二个字符（写入后 TXRDY 位再次清低）。

5. 第一个字符传输完成，开始将第二个字符传输至移位寄存器 TXSHF。

6. TXENA 位置低，禁用发送器；SCI 完成当前字符的传输。

7. 第二个字符传输完成，发送器为空并准备接收新字符。

## 24.10 SCI 端口中断

SCI 的接收器和发送器可通过中断控制。SCICTL2 寄存器包含一个标志位（TXRDY），用于指示发送缓冲器就绪状态（SCITXBUF 可接收下一个字符）；SCIRXST 寄存器包含两个中断标志位（RXRDY 和 BRKDT），以及一个

RXERROR 中断标志（由 FE、OE、BRKDT 和 PE 条件的逻辑或产生）。发送器和接收器拥有独立的中断使能位。未使能时，中断不会触发，但状态标志仍会保持活跃，反映传输和接收状态。

SCI 可为接收器和发送器提供独立的中断请求。当接收与发送中断处于同一优先级时，接收器中断始终优先于发送器中断，以降低接收缓冲区溢出的风险。

若 RXBKINTENA 位（SCICTL2 寄存器的第 1 位）置 1，当发生以下任一事件时，会触发接收器外设中断请求：

- SCI 接收到完整帧，并将 RXSHF 寄存器中的数据传输至 SCIRXBUF 寄存器，此时 RXRDY 标志位（SCIRXST 寄存器的第 6 位）置 1 并触发中断。
- 检测到间断条件（SCIRXD 引脚在缺失停止位后保持低电平 9.625 个位周期），此时 BRKDT 标志位（SCIRXST 寄存器的第 5 位）置 1 并触发中断。

若 TXINTENA 位（SCICTL2 寄存器的第 0 位）置 1，当 SCITXBUF 寄存器中的数据传输至 TXSHF 寄存器时（指示 CPU 可向 SCITXBUF 写入新数据），会触发发送器外设中断请求，此时 TXRDY 标志位（SCICTL2 寄存器的第 7 位）置 1 并触发中断。

**注意：**

SCI 模块中断响应时间——若应用程序中存在严格的时序要求，可能会偶尔触发 BRKDT 或其他错误（如 FE/PE）。

中断需在检测到停止位的约 7/8（约 0.875 个位时间）后才会触发。ISR（中断服务程序）进入前的实际延迟为：((7×波特时钟周期)/8 + 3×系统时钟周期)。

SCI 在 RX ISR 完成前不会开始读取额外的位/字符，因此需在 下一个字节的起始位开始前完成 ISR。无论中断原因如何，ISR 的允许完成时间约为 1/8 个位时间（约 0.125 个位时间）。

若 ISR 未在下一个起始位开始前（RX 线路再次变低前）完成，SCI 模块会在错误的位置过晚读取起始位，导致后续所有位读取错误，直至下一个对齐正确的起始位（ISR 有足够时间在起始位前处理完成）。

为避免错误（补偿 ISR 开始所需的 0.875 个位时间），建议采用以下方法：

1、保持 RX ISR 简洁。RX ISR 仅用于将 FIFO/缓冲区中的数据转移至内存，后续在非时间关键的函数中处理数据。

2、避免 SCI RX ISR 中过多的中断嵌套，防止嵌套导致 SCI RX ISR 的完成时间超过约 0.125 个位时间的窗口。

3、若需要额外时间（超过约 0.125 个位时间），可在向设备的 SCI RX 引脚传输额外数据前，在其他设备的固件中添加延迟。可通过以下方式在其他设备中添加延迟：

1、向设备发送带 2 个停止位的字节，可为 RX ISR 提供约 1.125 个位时间的处理时间。

2、在向设备传输的其他设备固件中，每发送一个字节后添加手动延迟，可为 RX ISR 提供（延迟时间 + 约 0.125 个位时间）的处理时间。

3、在向设备传输的其他设备固件中，每次的 RX 中断触发后添加手动延迟，可为本设备的 RX ISR 提供（延迟时间 + 约 0.125 个位时间）的处理时间。该方法实现难度较大，因为需要其他设备预测其发送的数据何时会触发设备的 RX 中断（例如 RX-FIFO 达到阈值、发送 BRKDT、发生 RXERROR 等）。

注意：

由 RXRDY 和 BRKDT 位触发的中断由 RXBKINTENA 位（SCICTL2 寄存器的第 1 位）控制；由 RXERROR 位触发的中断由 RXERRINTENA 位（SCICTL1 寄存器的第 6 位）控制。

### 24.10.1 间断检测

“间断（break）”信号（也称为“间断检测”或“间断序列”）可发送至模块，向总线传递特定状态。该间断信号定义为一定时长的低电平脉冲，通常至少为一个数据包的宽度（包含缺失的停止位）。SCI 有两种主要方法检测总线上的“间断”信号，每种方法均有特定限制。

第一种间断检测方法是读取 SCIRXST.BRKDT 位。触发 BRKDT 位的间断条件为：SCI 接收器数据线（SCIRXD）在缺失第一个停止位后，持续保持低电平至少 9.625 个位周期。若 SCIRX 线路在这 9.625 个位周期内的任一时刻变高，则 SCI 不会标记间断检测。要触发第一个停止位缺失，通常的方法是将 RX 线路保持低电平：

- 1 个起始位
- 8 个数据位
- （可选）1 个地址位
- （可选）1 个校验位
- 1 个停止位
- 额外 9.625 个位周期的低电平

总计低电平保持时间为：无校验位和地址位时 19.625 个位时间、仅使用校验位或地址位之一时 20.625 个位时间、同时使用校验位和地址位时 21.625 个位时间。

第二种间断检测方法是使用 SCIRXST.FE、SCIRXST.PE 和 SCIRXBUF.SAR 字段检测连续 10 个或 11 个低电平位。ISR 代码可通过以下标志和接收数据的组合判断是否发生间断检测：

- 连续 11 个低电平位（需启用校验）：
  - FE==1
  - 奇校验时 PE==1，偶校验时 PE==0
  - SCIRXBUF.SAR（接收字符）==0x00
- 连续 10 个低电平位（需禁用校验）：

- FE==1
- SCIRXBUF.SAR（接收字符）==0x00

## 24.11 SCI 波特率计算

内部生成的串行时钟由低速外设时钟（LSPCLK）和波特率选择寄存器决定。SCI 使用波特率选择寄存器的 16 位值，为给定的 LSPCLK 选择 65,536 种不同的串行时钟速率之一。

有关波特率计算的公式，请参阅波特率选择寄存器的位描述。表 24-3 列出了常见 SCI 波特率对应的波特率选择值。LSPCLK/16 为最大波特率，例如，若 LSPCLK 为 100 MHz，则最大波特率为 6.25 Mbit/s。

表 24-3 常见 SCI 波特率对应的异步波特寄存器值（LSPCLK 时钟频率为 100 MHz）

| 波特率 | BRR（十六进制） | BRR（十进制） | 实际波特率 | 误差百分比 |
|---|---|---|---|---|
| 2400 | 1457h | 5207 | 2400 | 0 |
| 4800 | A2Bh | 2603 | 4800 | 0 |
| 9600 | 515h | 1301 | 9601 | 0.01% |
| 19200 | 28Ah | 650 | 19201 | 0.01% |
| 38400 | 144h | 324 | 38462 | 0.16% |

## 24.12 SCI 增强特性

SCI 支持自动波特率检测和发送/接收 FIFO 缓冲区功能。以下章节将详细说明 FIFO 的操作。

### 24.12.1 SCI FIFO 描述

以下步骤详细说明 FIFO 特性及 SCI FIFO 的编程方法：

1. 复位：复位时，SCI 默认工作在标准 SCI 模式，FIFO 功能禁用，FIFO 寄存器（SCIFFTX、SCIFFRX 和 SCIFFCT）处于非活跃状态。

2. 标准 SCI 模式：标准 SCI 模式正常工作，以 TXINT/RXINT 中断作为模块的中断源。

3. FIFO 使能：将 SCIFFTX.SCIFFENA 置 1 可启用 FIFO 增强功能。将 SCIFFTX.SCIRST 清 0 会复位 SCI 通道及 TX/RX FIFO 中断状态，并清空 FIFO；重新置 1 后恢复收发。

4. 寄存器激活：所有 SCI 寄存器和 SCI FIFO 寄存器（SCIFFTX、SCIFFRX 和 SCIFFCT）均处于活跃状态。

5. 中断：FIFO 模式支持两个中断——发送 FIFO（TXINT）和接收 FIFO（RXINT）。RXINT 为 SCI FIFO 接收、接收错误和接收 FIFO 溢出条件的公共中断；标准 SCI 的 TXINT 禁用，该中断作为 SCI 发送 FIFO 中断。

6. 缓冲区：发送端和接收端各设一个 16 级 FIFO。发送 FIFO 数据宽度为 8 位，接收 FIFO 数据宽度为 10 位（8 位数据加帧错误与奇偶校验错误标志）。启用 FIFO 后，写入 SCITXBUF 的字符进入 TX FIFO，TXSHF 在 SCIFFCT 配置的可选延迟之后直接从 TX FIFO 装载；此时不应把 SCITXBUF 视为额外一级 FIFO。RX FIFO 顶部字符通过 SCIRXBUF 读取。

7. 延迟传输：FIFO 中的数据传输至发送移位寄存器的速率可编程配置。SCIFFCT 寄存器的 FFTXDLY7-FFTXDLY0 位（第 7-0 位）定义了数据传输之间的延迟，延迟单位为 SCI 波特时钟周期。8 位 FFTXDLY 字段的编码范围为 0–255 个波特时钟周期。延迟为 0 时，SCI 模块可连续传输 FIFO 中的数据；延迟为 255 个时钟周期时，SCI 模块以最大延迟模式传输数据，FIFO 中相邻数据的传输间隔为 255 个波特时钟周期。可编程延迟支持 SCI 与低速 SCI/UART 通信，且几乎无需 CPU 干预。

8. FIFO 状态：TXFFST 和 RXFFST（位 12-8）分别表示发送、接收 FIFO 中当前的数据字数。TXFIFORESET 或 RXFIFORESET 清 0 时，相应 FIFO 指针清零并保持复位；重新置 1 后恢复工作。

9. 可编程中断阈值：发送和接收 FIFO 均可触发 CPU 中断。TXFFST 小于或等于 TXFFIL 时产生发送 FIFO 中断；RXFFST 大于或等于 RXFFIL 时产生

接收 FIFO 中断。TXFFIL 与 RXFFIL 的复位值分别为 00000b 和 11111b，应用程序必须按吞吐量与 ISR 开销要求重新配置。

图 24-10 详细说明非 FIFO/FIFO 模式下 SCI 中断的操作/配置。

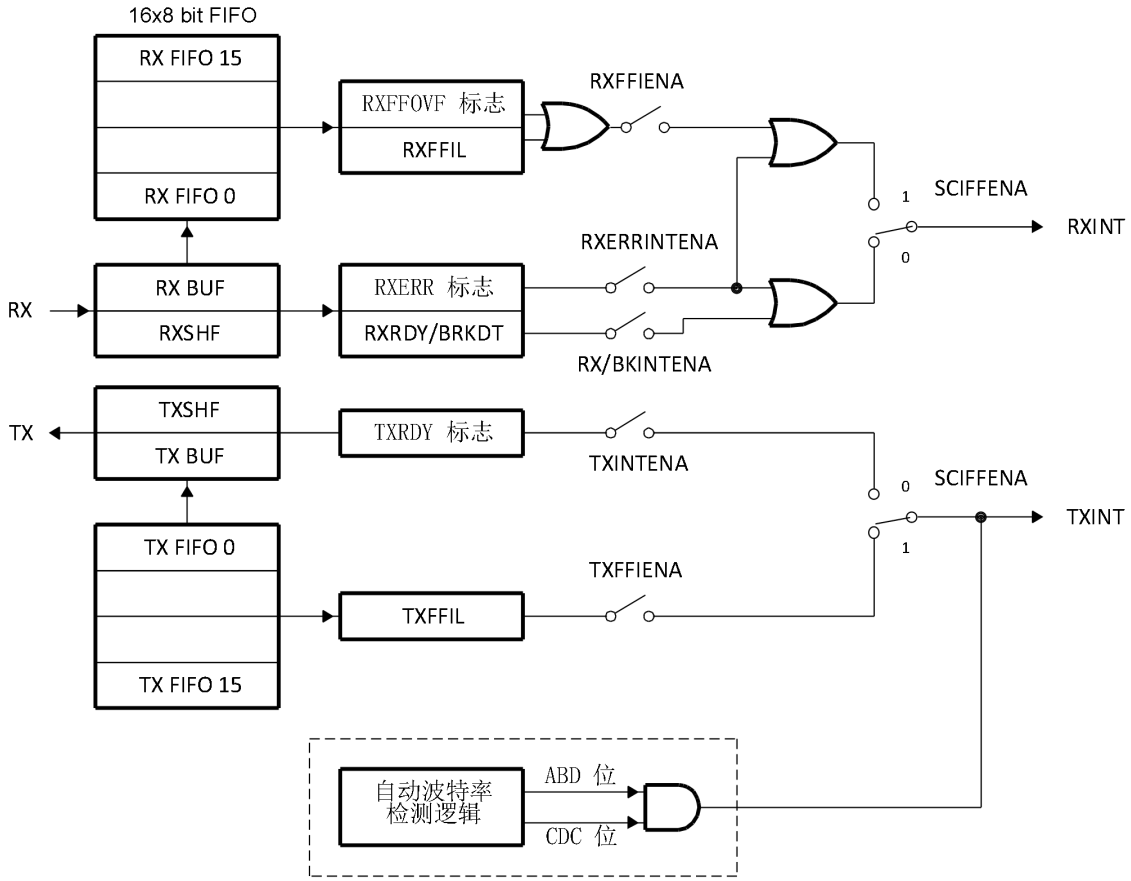



图 24-10 SCI FIFO 中断标志

表 24-4 SCI 中断标志

| FIFO选项 | SCI中断源 | 中断标志 | 中断使能 | FIFO使能 (SCIFFENA) | 中断线路 |
|---|---|---|---|---|---|
| 无FIFO的SCI | 接收错误 | RXERR | RXERRINTENA | 0 | RXINT |
| | 接收间断 | BRKDT | RXBKINTENA | 0 | RXINT |
| | 数据接收 | RXRDY | RXBKINTENA | 0 | RXINT |
| | 发送空 | TXRDY | TXINTENA | 0 | TXINT |
| 带FIFO的SCI | 接收错误和接收间断 | RXERR | RXERRINTENA | 1 | RXINT |
| | FIFO接收 | RXFFIL | RXFFIENA | 1 | RXINT |
| | 发送空 | TXFFIL | TXFFIENA | 1 | TXINT |

| FIFO选项 | SCI中断源 | 中断标志 | 中断使能 | FIFO使能 (SCIFFENA) | 中断线路 |
|---|---|---|---|---|---|
| 自动波特率 | 自动波特率检测完成 | ABD | 无关 | x | TXINT |

**注：**

FIFO 模式下，TXSHF 在延迟值后直接加载，SCITXBUF 不使用。

RXERR 可由 BRKDT、FE、OE、PE 标志置 1。FIFO 模式下，间断检测中断仅通过 RXERR 标志触发。

### 24.12.2 SCI 自动波特率

大多数 SCI 模块没有内置的自动波特率检测硬件逻辑。这些 SCI 模块通常集成在时钟速率依赖于 PLL 复位值的嵌入式控制器中，且嵌入式控制器的时钟在最终设计后可能发生变化。增强特性集中的 SCI 模块支持硬件自动波特率检测逻辑，以下章节将详细说明自动波特率检测功能的启用流程。

### 24.12.3 自动波特率检测流程

SCIFFCT 寄存器的 ABD 和 CDC 位控制自动波特率逻辑，SCIRST 位置 1 可使自动波特率逻辑工作。

若 ABD 置 1 且 CDC 为 1（指示自动波特率对齐），则触发 SCI 发送 FIFO 中断（TXINT）。中断服务后，需通过软件清 0CDC 位；若中断服务后 CDC 仍保持置 1，则无法再次触发中断。

自动波特率检测流程如下：

1. 通过设置 SCIFFCT 寄存器的 CDC 位（第 13 位）并向 ABDCLR 位（第 14 位）写入 1 以清 0ABD 位（第 15 位），启用 SCI 的自动波特率检测模式。

2. 将 BRR 初始化为非零值，并使对应波特率不超过 500 kbit/s（BRR≥1）。

3. 允许 SCI 从主机接收字符“A”或“a”（需以期望的波特率发送）。若接收到的第一个字符为“A”或“a”，则自动波特率检测硬件会检测输入波特率并置 1ABD 位。

4. 自动检测硬件会将等效的波特率十六进制值更新至波特率寄存器，并向 CPU 触发中断。

5. 响应中断：向 SCIFFCT.ABDCLR（位 14）写 1 清除 ABD（位 15），再将 CDC 清 0，禁止后续自动波特率锁定。

6. 读取接收缓冲区中的字符“A”或“a”，清空缓冲区及缓冲区状态。

7. 若 ABD 置 1 且 CDC 为 1（指示自动波特率对齐），则触发 SCI 发送 FIFO 中断（TXINT）。中断服务后，需通过软件清 0 CDC 位。

**注意：**

在高波特率下，输入数据位的转换速率可能受收发器和连接器性能影响。虽然正常串行通信可能不受影响，但该转换速率会限制高波特率（通常超过 100 kBd）下自动波特率检测的可靠性，导致自动波特率锁定功能失效。

为避免此问题，建议采用以下方法：

1、主机与 SCI 引导加载程序通过较低的波特率实现波特率锁定。

2、之后主机可与加载的应用程序握手，将 SCI 波特率寄存器设置为期望的较高波特率。

## 24.13 寄存器描述

### 24.13.1 SCI 基地址表

表 24-5 SCI 基地址表

| 实例 | 结构 | DriverLib 名称 | 基地址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| SciaRegs | SCI_REGS | SCIA_BASE | 0x0069_1000 | - | YES | YES |

### 24.13.2 SCI_REGS 寄存器

表 24-6 列出了 SCI_REGS 寄存器的内存映射寄存器。表中未列出的所有寄存器偏移地址均为保留地址，不得修改其寄存器内容。

表 24-6 SCI_REGS 寄存器列表

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | SCICCR | 通信控制寄存器 |

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 2h | SCICTL1 | 控制寄存器 1 |
| 4h | SCIHBAUD | 波特率（高位）寄存器 |
| 6h | SCILBAUD | 波特率（低位）寄存器 |
| 8h | SCICTL2 | 控制寄存器 2 |
| Ah | SCIRXST | 接收状态寄存器 |
| Ch | SCIRXEMU | 接收仿真缓冲寄存器 |
| Eh | SCIRXBUF | 接收数据缓冲寄存器 |
| 12h | SCITXBUF | 发送数据缓冲寄存器 |
| 14h | SCIFFTX | FIFO 发送寄存器 |
| 16h | SCIFFRX | FIFO 接收寄存器 |
| 18h | SCIFFCT | FIFO 控制寄存器 |
| 1Eh | SCIPRI | SCI 优先级控制寄存器 |

复杂位访问类型采用编码形式以便在表格中展示，表 24-7 列出了本节中使用的访问类型编码。

表 24-7 SCI_REGS 访问类型编码

| 访问类型 | 编码 | 描述 |
|---|---|---|
| 读类型 | R | 只读 |
| | R-0 | 读取时返回 0 |
| 写类型 | W | 只写 |
| | W1S | 写 1 置位 |
| 复位或默认值 | -n | 复位后的取值或默认值 |

注：位布局表中的 R/W-0h 表示该字段可读写、复位值为 0h；R-0h 表示读取返回 0、复位值为 0h；R-0/W1S 表示读取返回 0、写 1 置位。

### 24.13.2.1 SCICCR 寄存器

偏移地址：0h　复位值：0000h

通信控制寄存器

SCICCR 定义 SCI 使用的字符格式、协议和通信模式。

表 24-8 SCICCR 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STOPBITS | PARITY | PARITYENA | LOOPBKENA | ADDRIDLE_MODE | SCICHAR | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | | |

表 24-9 SCICCR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7 | STOPBITS | R/W | 0h | SCI 停止位数量。此位指定发送的停止位数量；接收器只检查一个停止位。<br>0h (R/W) = 1 个停止位。<br>1h (R/W) = 2 个停止位。<br>复位类型：SYSRSn |
| 6 | PARITY | R/W | 0h | SCI 奇偶校验选择。PARITY ENABLE（SCICCR 位 5）置位时，PARITY（位 6）指定奇校验或偶校验，即发送和接收字符中值为 1 的位数为奇数或偶数。<br>0h (R/W) = 奇校验。<br>1h (R/W) = 偶校验。<br>复位类型：SYSRSn |
| 5 | PARITYENA | R/W | 0h | SCI 奇偶校验使能。启用或禁用校验功能。在地址位多处理器模式（本寄存器位 3 设置）下，启用校验时地址位也参与计算。对于不足 8 位的字符，应屏蔽未使用位，使其不参与校验计算。<br>0h (R/W) = 禁用；发送时不生成校验位，接收时不期望校验位。<br>1h (R/W) = 启用奇偶校验。<br>复位类型：SYSRSn |
| 4 | LOOPBKENA | R/W | 0h | 回环测试模式使能。此模式在内部将 Tx 引脚连接到 Rx 引脚。<br>0h (R/W) = 禁用回环测试。<br>1h (R/W) = 启用回环测试。<br>复位类型：SYSRSn |
| 3 | ADDRIDLE_MODE | R/W | 0h | SCI 多处理器模式控制。选择多处理器协议。多处理器通信使用 SLEEP 和 TXWAKE 功能（分别为 SCICTL1 位 2 和位 3），因此不同于其他通信模式。地址位模式会增加一个帧位，而空闲线模式不会增加此位且兼容 RS-232 类型通 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 信，所以普通通信通常使用空闲线模式。<br>0h (R/W) = 选择空闲线模式协议。<br>1h (R/W) = 选择地址位模式协议。<br>复位类型：SYSRSn |
| 2-0 | SCICHAR | R/W | 0h | 字符长度控制位 2–0。选择 1 至 8 位字符长度。不足 8 位的字符在 SCIRXBUF 和 SCIRXEMU 中右对齐，在 SCIRXBUF 中高位补零；SCITXBUF 不要求高位补零。<br>0h (R/W) = SCICHAR_LENGTH_1<br>1h (R/W) = SCICHAR_LENGTH_2<br>2h (R/W) = SCICHAR_LENGTH_3<br>3h (R/W) = SCICHAR_LENGTH_4<br>4h (R/W) = SCICHAR_LENGTH_5<br>5h (R/W) = SCICHAR_LENGTH_6<br>6h (R/W) = SCICHAR_LENGTH_7<br>7h (R/W) = SCICHAR_LENGTH_8<br>复位类型：SYSRSn |

### 24.13.2.2 SCICTL1 寄存器

偏移地址：2h　复位值：0000h

控制寄存器 1

SCICTL1 控制接收器和发送器使能、TXWAKE、SLEEP 以及 SCI 软件复位。

表 24-10 SCICTL1 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | RXERRINTENA | SWRESET | 保留 | TXWAKE | SLEEP | TXENA | RXENA |
| R-0h | R/W-0h | R/W-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 24-11 SCICTL1 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-7 | RESERVED | R | 0h | 保留。 |
| 6 | RXERRINTENA | R/W | 0h | SCI 接收错误中断使能。置位后，当错误使 RXERROR（SCIRXST 位 7）置位时产生中断。<br>0h (R/W) = 禁用接收错误中断。<br>1h (R/W) = 启用接收错误中断。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 5 | SWRESET | R/W | 0h | SCI 软件复位，低电平有效。写 0 将 SCI 状态机和运行标志（SCICTL2、SCIRXST）初始化为复位状态，但不会复位 FIFO 指针或清除 TX/RX FIFO 数据。若需清除 FIFO，应执行 SWRESET + TXFFINT + RXFFINT，或使用通道复位 SCIFFTX[SCIRST]。SWRESET 不影响配置位；受影响逻辑保持复位，直到向此位写 1。本节寄存器图下方给出了复位位值。因此系统复位后，应写 1 重新启用 SCI。接收器检测到间断（BRKDT，SCIRXST 位 5）后应清除此位。SWRESET 影响运行标志，但不影响配置位，也不会将配置位恢复为复位值。SWRESET 有效后，标志冻结，直到解除复位。受影响标志及 SWRESET 后的值为：<br>TXRDY，SCICTL2 位 7：1；TXEMPTY，SCICTL2 位 6：1。<br>RXWAKE，SCIRXST 位 1：0；PE，位 2：0；OE，位 3：0；FE，位 4：0；BRKDT，位 5：0；RXRDY，位 6：0；RXERROR，位 7：0。<br>0h (R/W) = 将 SCI 状态机和运行标志（SCICTL2、SCIRXST）初始化为复位状态。<br>1h (R/W) = 系统复位后重新启用 SCI。写 0 后再写 1 不要求等待特定时间。<br>复位类型：SYSRSn |
| 4 | RESERVED | R | 0h | 保留。 |
| 3 | TXWAKE | R/W | 0h | SCI 发送器唤醒方法选择。TXWAKE 根据 ADDRIDLE_MODE（SCICCR 位 3）指定的空闲线或地址位发送模式选择数据发送功能。<br>0h (R/W) = 未选择发送功能。在空闲线模式下，先向 TXWAKE 写 1，再向 SCITXBUF 写数据，可生成 11 个数据位时间的空闲期；在地址位模式下，先向 TXWAKE 写 1，再向 SCITXBUF 写数据，可将该帧地址位置 1。<br>1h (R/W) = 所选发送功能取决于空闲线或地址位模式。SWRESET（SCICTL1 位 5）不清除 TXWAKE；系统复位或将 TXWAKE 传送至 WUT 标志时清除。<br>复位类型：SYSRSn |
| 2 | SLEEP | R/W | 0h | SCI 休眠。在多处理器配置中，此位控制接收器休眠；清除此位使 SCI 退出休眠。SLEEP 置位时接收器仍工作，但除非检测到地址字节，否则不更新接收缓冲器就绪位（SCIRXST 位 6，RXRDY）或错误状态位（SCIRXST 位 5–2：BRKDT、FE、OE、PE）。检测到地址字节不会 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 清除 SLEEP。<br>0h (R/W) = 禁用休眠模式。<br>1h (R/W) = 启用休眠模式。<br>复位类型：SYSRSn |
| 1 | TXENA | R/W | 0h | SCI 发送器使能。仅当 TXENA 置位时才通过 SCITXD 发送数据。清零后，先发送完此前写入 SCITXBUF 的全部数据，再停止发送。TXENA 禁用期间写入 SCITXBUF 的数据，即使后来启用 TXENA 也不会发送。<br>0h (R/W) = 禁用发送器。<br>1h (R/W) = 启用发送器。<br>复位类型：SYSRSn |
| 0 | RXENA | R/W | 0h | SCI 接收器使能。SCIRXD 接收的数据先进入接收移位寄存器，再进入接收缓冲器。此位控制向缓冲器传送数据。清除 RXENA 将停止向两个接收缓冲器传送字符，并停止产生接收中断；但不会阻止 RX 错误中断。应通过 RXERRINTENA 禁用 RX 错误中断，通过 RXBKINTENA 阻止 BRKDT 中断传播。<br>RXENA 清零期间，接收移位寄存器仍可组装字符；若在接收一个字符期间置位 RXENA，完整字符将传送至 SCIRXEMU 和 SCIRXBUF。<br>0h (R/W) = 阻止接收字符传送至 SCIRXEMU 和 SCIRXBUF。<br>1h (R/W) = 将接收字符传送至 SCIRXEMU 和 SCIRXBUF。<br>复位类型：SYSRSn |

### 24.13.2.3 SCIHBAUD 寄存器

偏移地址：4h　复位值：0000h

波特率高字节寄存器

SCIHBAUD 和 SCILBAUD 的值指定 SCI 波特率。

表 24-12 SCIHBAUD 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BAUD | | | | | | | |
| R/W-0h | | | | | | | |

表 24-13 SCIHBAUD 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | BAUD | R/W | 0h | SCI 16 位波特率选择寄存器 SCIHBAUD（高字节）。内部串行时钟由低速外设时钟 LSPCLK 和两个波特率选择寄存器决定。SCI 使用其 16 位值，在通信模式下选择 64K 种串行时钟速率之一。<br>BRR = (SCIHBAUD << 8) + SCILBAUD<br>SCI 异步波特率 = LSPCLK / ((BRR + 1) × 8)<br>或：BRR = LSPCLK / (SCI 异步波特率 × 8) − 1<br>上述公式仅适用于 0 < BRR < 65536。BRR = 0 时，SCI 异步波特率 = LSPCLK / 16。<br>BRR 为波特率选择寄存器中的 16 位值，以十进制表示。<br>复位类型：SYSRSn |

### 24.13.2.4 SCILBAUD 寄存器

偏移地址：6h　复位值：0000h

波特率低字节寄存器

SCIHBAUD 和 SCILBAUD 的值指定 SCI 波特率。

表 24-14 SCILBAUD 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BAUD | | | | | | | |
| R/W-0h | | | | | | | |

表 24-15 SCILBAUD 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | BAUD | R/W | 0h | 参见 SCIHBAUD 的详细说明。<br>复位类型：SYSRSn |

### 24.13.2.5 SCICTL2 寄存器

偏移地址：8h　复位值：00C0h

控制寄存器 2

SCICTL2 控制接收就绪、间断检测和发送就绪中断，以及发送器就绪和空标志。

表 24-16 SCICTL2 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXRDY | TXEMPTY | 保留 | | | | RXBKINTENA | TXINTENA |
| R-1h | R-1h | R-0h | | | | R/W-0h | R/W-0h |

表 24-17 SCICTL2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7 | TXRDY | R | 1h | 发送缓冲器就绪标志。置位表示 SCITXBUF 可以接收另一个字符；向 SCITXBUF 写数据自动清除此位。当此标志与 TXINTENA（SCICTL2.0）均置位时，提出发送中断请求。使 SWRESET（SCICTL1.5）有效或系统复位可将 TXRDY 置 1。<br>0h (R) = SCITXBUF 已满。<br>1h (R) = SCITXBUF 可接收下一个字符。<br>复位类型：SYSRSn |
| 6 | TXEMPTY | R | 1h | 发送器空标志，表示 SCITXBUF 和 TXSHF 的内容状态。SWRESET（SCICTL1.5）有效或系统复位可置位此位。此位不产生中断请求。<br>0h (R) = 发送缓冲器或移位寄存器，或两者中有数据。<br>1h (R) = 发送缓冲器和移位寄存器均为空。<br>复位类型：SYSRSn |
| 5-2 | RESERVED | R | 0h | 保留。 |
| 1 | RXBKINTENA | R/W | 0h | 接收缓冲器/间断检测中断使能。控制由 RXRDY 或 BRKDT（SCIRXST.6、.5）置位引起的中断请求，但不阻止这些标志置位。<br>0h (R/W) = 禁用 RXRDY/BRKDT 中断。<br>1h (R/W) = 启用 RXRDY/BRKDT 中断。<br>复位类型：SYSRSn |
| 0 | TXINTENA | R/W | 0h | SCITXBUF 寄存器中断使能。控制 TXRDY（SCICTL2.7）置位引起的中断请求，但不阻止表示 SCITXBUF 可接收字符的 TXRDY 标志置位。0 禁用 TXRDY 中断；1 启用。<br>在非 FIFO 模式中，首次启用或禁用后再次启用 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 发送中断时，必须向 SCITXBUF 写入虚拟或有效数据，才会产生第一次发送中断。如果写入数据后才启用 TXINTENA，则不会产生中断。<br>0h (R/W) = 禁用 TXRDY 中断。<br>1h (R/W) = 启用 TXRDY 中断。<br>复位类型：SYSRSn |

### 24.13.2.6 SCIRXST 寄存器

偏移地址：Ah　复位值：0000h

接收状态寄存器

SCIRXST 包含 7 个接收状态标志，其中 RXRDY、BRKDT 可直接产生中断请求；RXERROR 为错误标志的逻辑或，也可经 RXERRINTENA 产生中断。每当完整字符传送至 SCIRXEMU 和 SCIRXBUF 时，状态标志更新。

表 24-18 SCIRXST 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXERROR | RXRDY | BRKDT | FE | OE | PE | RXWAKE | 保留 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 24-19 SCIRXST 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7 | RXERROR | R | 0h | SCI 接收器错误标志。RXERROR 是接收状态寄存器中 BRKDT、FE、OE、PE（位 5–2）的逻辑或。此位为 1 且 RXERRINTENA（SCICTL1.6）置位时产生中断，可在中断服务程序中快速检查错误。此标志不能直接清除，只能由有效 SWRESET、通道复位 SCIRST 或系统复位清除。<br>注意 SCI 模块中断响应时间：应用时序紧张时，可能偶发 BRKDT、FE 或 PE。检测到约 7/8 个停止位（约 0.875 位时间）后才触发中断。进入 ISR 前的实际延迟为 ((7 × BAUD_CLK_PERIOD) / 8 + 3 × SYSCLK_PERIOD)。RX ISR 完成前 SCI 不会开始读取其他位或字符，因此 ISR 必须在下一 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 字节起始位开始前完成。不论中断原因如何，整个 ISR 只有约 1/8 位时间（0.125 位时间）。如果 RX 线再次变低、下一起始位开始前 ISR 未完成，SCI 会在错误位置过晚读取起始位，之后所有位都可能读错，直到遇到下一个正确对齐且 ISR 有足够处理时间的起始位。<br>建议：1. 缩短 RX ISR，仅将 FIFO / 缓冲器数据移至内存，在其他实时性要求较低的函数中处理。2. 避免 RX ISR 中过度嵌套其他中断，确保在约 0.125 位时间内完成。3. 若需要更多处理时间，可在另一设备固件向本设备 SCI RX 继续发送数据前加入延迟：A. 每字节使用 2 个停止位，可提供约 1.125 位时间；B. 每发送一个字节后增加延迟，可提供“延迟 + 约 0.125 位时间”；C. 每次本设备 RX 中断后增加延迟，也可提供相同处理时间，但需要另一设备预测何时触发 RX 中断，因此较难实现。RX FIFO 达到阈值、发送 BRKDT 或发生 RXERROR 等均可触发 RX 中断。<br>0h (R) = 无错误标志置位。<br>1h (R) = 一个或多个错误标志置位。<br>复位类型：SYSRSn |
| 6 | RXRDY | R | 0h | SCI 接收就绪标志。SCIRXBUF 中有新字符可读时置位；若 RXBKINTENA（SCICTL2.1）为 1，则产生接收中断。读取 SCIRXBUF、有效 SWRESET、通道复位 SCIRST 或系统复位可清除此位。<br>0h (R) = SCIRXBUF 中无新字符。<br>1h (R) = SCIRXBUF 中字符可读。<br>复位类型：SYSRSn |
| 5 | BRKDT | R | 0h | SCI 间断检测标志。首次停止位缺失后，SCIRXD 连续保持低电平至少 9.625 位时间时，SCI 检测到间断条件并置位；若其间任意时刻变高，则不会置位。通常需将 RX 拉低持续 1 个起始位、8 个数据位、可选地址位、可选校验位、1 个停止位及额外 9.625 位时间：无校验 / 地址位时共 19.625 位时间，只有其中一位时 20.625，两位都有时 21.625。<br>若希望检测连续 11 个低电平位的间断序列，在启用奇校验时可使用 FE==1 && PE==1 && SCIRXBUF.SAR（接收字符）==0x00；偶校验时改为 PE==0。若不使用校验位，可检测 10 个低电平位，此时无需考虑 PE。<br>RXBKINTENA 为 1 时，间断产生接收中断，但不装载接收缓冲器。SLEEP 为 1 时也可能产 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 生 BRKDT 中断。有效 SWRESET、SCIRST 或系统复位可清除 BRKDT；间断后收到字符不能清除。<br>BRKDT 置位后，如不执行软件、通道或系统复位，RXRDY 不会置位，首次检测错误的中断之后也不会再有中断。若要接收更多字符，必须切换 SWRESET、执行 SCIRST 或系统复位。<br>注意：系统容易出现间断时，应在 SCI-RX 引脚加上拉电阻，确保正确回高和抗噪能力。可用示波器监视本设备 SCI-RX 是否有宽于 9.625 位时间的低电平；若并非有意发送间断，应修正发送设备的软件。正常情况下不应出现这种低电平。<br>0h (R) = 无间断条件。<br>1h (R) = 已发生间断条件。<br>复位类型：SYSRSn |
| 4 | FE | R | 0h | SCI 帧错误标志。预期停止位未出现时置位；只检查第一个停止位。停止位缺失表明已失去与起始位的同步且字符帧格式错误。清除 SWRESET 位、通道复位 SCIRST 或系统复位可复位 FE。<br>注意：除接收休眠模式外，FE 先于 BRKDT 置位。在休眠模式下，若无 RX WAKEUP 且 RXD 低电平超过 10 位时间，则置 BRKDT 而不置 FE。<br>0h (R) = 未检测到帧错误。<br>1h (R) = 检测到帧错误。<br>复位类型：SYSRSn |
| 3 | OE | R | 0h | SCI 接收溢出错误标志。前一字符尚未被 CPU 或 DMAC 完整读取，下一字符已传送至 SCIRXEMU 和 SCIRXBUF 时置位；前一字符被覆盖并丢失。有效 SWRESET、通道复位 SCIRST 或系统复位清除此位。<br>0h (R) = 无接收溢出错误。<br>1h (R) = 检测到接收溢出错误。<br>复位类型：SYSRSn |
| 2 | PE | R | 0h | SCI 奇偶校验错误标志。收到字符中 1 的数量与校验位不匹配时置位；地址位参与计算。未启用校验生成和检测时，PE 禁用且读为 0。有效 SWRESET、通道复位 SCIRST 或系统复位清除此位。<br>0h (R) = 无校验错误或校验已禁用。<br>1h (R) = 检测到校验错误。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 1 | RXWAKE | R | 0h | 接收器唤醒检测标志。<br>0h (R) = 未检测到接收器唤醒条件。<br>1h (R) = 检测到唤醒条件。地址位多处理器模式（SCICCR.3 = 1）下，RXWAKE 表示 SCIRXBUF 中字符的地址位；空闲线多处理器模式下，检测到 SCIRXD 空闲时置位。<br>RXWAKE 为只读标志，可通过以下方式清除：将地址字节后的第一个字节传送至 SCIRXBUF（仅非 FIFO 模式）；读取 SCIRXBUF；有效 SWRESET；通道复位 SCIRST；系统复位。<br>复位类型：SYSRSn |
| 0 | RESERVED | R | 0h | 保留。 |

### 24.13.2.7 SCIRXEMU 寄存器

偏移地址：Ch　复位值：0000h

接收仿真缓冲寄存器

正常 SCI 接收操作读取 SCIRXBUF。SCIRXEMU 主要供仿真器使用，可连续读取接收数据以更新屏幕而不清除 RXRDY。系统复位清除 SCIRXEMU；仿真器监视窗口应使用此寄存器查看 SCIRXBUF 内容。SCIRXEMU 没有独立物理实现，仅以另一地址访问 SCIRXBUF，且不清除 RXRDY。

表 24-20 SCIRXEMU 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERXDT | | | | | | | |
| R-0h | | | | | | | |

表 24-21 SCIRXEMU 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | ERXDT | R | 0h | 接收仿真缓冲器数据。<br>复位类型：SYSRSn |

### 24.13.2.8 SCIRXBUF 寄存器

偏移地址：Eh　复位值：0000h

接收数据缓冲器

当前接收数据从 RXSHF 移至接收缓冲器时，RXRDY 置位，数据可读；若 RXBKINTENA（SCICTL2.1）置位，还会产生中断。读取 SCIRXBUF 清除 RXRDY；系统复位清除 SCIRXBUF。

表 24-22 SCIRXBUF 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SCIFFFE | SCIFFPE | 保留 | | | | | |
| R-0h | R-0h | R-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SAR | | | | | | | |
| R-0h | | | | | | | |

表 24-23 SCIRXBUF 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | SCIFFFE | R | 0h | SCI FIFO 帧错误标志，仅在启用 FIFO 时适用。SCIFFFE 对应 SCIRXBUF 中正在接收 / 读取的特定数据，每份 FIFO 接收数据均包含此信息；SCIRXST.FE 则为总体错误标志，任何已接收数据有帧错误都会置位。<br>0h (R) = 接收位 7–0 字符时无帧错误。<br>1h (R) = 接收位 7–0 字符时有帧错误。此位关联 FIFO 顶部字符。<br>复位类型：SYSRSn |
| 14 | SCIFFPE | R | 0h | SCI FIFO 奇偶校验错误标志，仅在启用 FIFO 时适用。SCIFFPE 对应 SCIRXBUF 中特定接收 / 读取数据，每份 FIFO 数据均包含此信息；SCIRXST.PE 为总体错误标志，任何接收数据有校验错误都会置位。<br>注意：接收期间更改奇偶校验或其他设置后，SCI 不会按新设置重新解释已收到的数据。更改参数时应清除 FIFO，或接受更改可能导致数据错误。若 RX 校验错误间歇出现，可能是 ISR 过长，应限制中断嵌套、提高 SCI 中断优先级并尽量将处理移出 ISR，使 ISR 时间最短。<br>0h (R) = 接收位 7–0 字符时无校验错误。<br>1h (R) = 接收位 7–0 字符时有校验错误。此位关联 FIFO 顶部字符。<br>复位类型：SYSRSn |
| 13-8 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 7-0 | SAR | R | 0h | 接收字符位。<br>复位类型：SYSRSn |

### 24.13.2.9 SCITXBUF 寄存器

偏移地址：12h　复位值：0000h

发送数据缓冲器

待发送数据写入 SCITXBUF，必须右对齐，因为不足 8 位字符的高位被忽略。数据移至发送移位寄存器 TXSHF 后，TXRDY（SCICTL2.7）置位，表示可接收下一组数据；TXINTENA（SCICTL2.0）置位时还会产生中断。

表 24-24 SCITXBUF 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXDT | | | | | | | |
| R/W-0h | | | | | | | |

表 24-25 SCITXBUF 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | TXDT | R/W | 0h | 发送数据缓冲器。<br>复位类型：SYSRSn |

### 24.13.2.10 SCIFFTX 寄存器

偏移地址：14h　复位值：A000h

FIFO 发送寄存器

SCIFFTX 控制发送 FIFO 中断、FIFO 增强功能以及 SCI 发送和接收通道复位。

表 24-26 SCIFFTX 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SCIRST | SCIFFENA | TXFIFORESET | TXFFST | | | | |
| R/W-1h | R/W-0h | R/W-1h | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXFFINT | TXFFINTCLR | TXFFIENA | TXFFIL | | | | |
| R-0h | R-0/W1S-0h | R/W-0h | R/W-0h | | | | |

表 24-27 SCIFFTX 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | SCIRST | R/W | 1h | SCI 复位。<br>0：写 0 执行 SWRESET，并复位 TXFFINT 和 RXFFINT，实质上清除 TX/RX FIFO 内容。SCI 保持复位直到写 1；同时复位 RXFFOVF、PE、OE、FE、RXERROR、BRKDT、RXRDY、RXWAKE，并将 TXRDY 和 TXEMPTY 置 1。<br>1：SCI FIFO 可恢复发送或接收；自动波特率逻辑工作也要求 SCIRST 为 1。<br>复位类型：SYSRSn |
| 14 | SCIFFENA | R/W | 0h | SCI FIFO 使能。<br>0h (R/W) = 禁用 FIFO 增强功能。<br>1h (R/W) = 启用 FIFO 增强功能。<br>复位类型：SYSRSn |
| 13 | TXFIFORESET | R/W | 1h | 发送 FIFO 复位。<br>0h (R/W) = FIFO 指针复位为 0 并保持复位。<br>1h (R/W) = 重新启用发送 FIFO。<br>复位类型：SYSRSn |
| 12-8 | TXFFST | R | 0h | FIFO 状态。<br>0h (R) = 发送 FIFO 为空。<br>1h (R) = 发送 FIFO 中有 1 个字。<br>2h (R) = 发送 FIFO 中有 2 个字。<br>3h (R) = 发送 FIFO 中有 3 个字。<br>4h (R) = 发送 FIFO 中有 4 个字。<br>5h (R) = 发送 FIFO 中有 5 个字。<br>6h (R) = 发送 FIFO 中有 6 个字。<br>7h (R) = 发送 FIFO 中有 7 个字。<br>8h (R) = 发送 FIFO 中有 8 个字。<br>9h (R) = 发送 FIFO 中有 9 个字。<br>Ah (R) = 发送 FIFO 中有 10 个字。<br>Bh (R) = 发送 FIFO 中有 11 个字。<br>Ch (R) = 发送 FIFO 中有 12 个字。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | Dh (R) = 发送 FIFO 中有 13 个字。<br>Eh (R) = 发送 FIFO 中有 14 个字。<br>Fh (R) = 发送 FIFO 中有 15 个字。<br>10h (R) = 发送 FIFO 中有 16 个字。<br>复位类型：SYSRSn |
| 7 | TXFFINT | R | 0h | 发送 FIFO 中断，只读位。<br>0h (R) = 未发生 TXFIFO 中断。<br>1h (R) = 已发生 TXFIFO 中断。<br>复位类型：SYSRSn |
| 6 | TXFFINTCLR | R-0/W1S | 0h | 发送 FIFO 中断清除。<br>0h (W1S) = 对 TXFFINT 标志无影响；读回 0。<br>1h (W1S) = 清除位 7 的 TXFFINT 标志。<br>复位类型：SYSRSn |
| 5 | TXFFIENA | R/W | 0h | 发送 FIFO 中断使能。<br>0h (R/W) = 禁用。<br>1h (R/W) = 启用；TXFFST 小于或等于触发阈值 TXFFIL（位 4–0）时触发中断。<br>复位类型：SYSRSn |
| 4-0 | TXFFIL | R/W | 0h | TXFFIL4–0 发送 FIFO 中断阈值。TXFFST4–0 小于或等于 TXFFIL4–0 时产生中断。用于产生中断的最大设置值不得超过 TX FIFO 深度。复位默认值为 00000b。用户应权衡 CPU 中断服务开销与 SCI 总线带宽利用率，按应用设置 TXFFIL。<br>复位类型：SYSRSn |

### 24.13.2.11 SCIFFRX 寄存器

偏移地址：16h　复位值：201Fh

FIFO 接收寄存器

SCIFFRX 控制接收 FIFO 中断、复位及溢出状态。

表 24-28 SCIFFRX 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RXFFOVF | RXFFOVRCLR | RXFIFORESET | RXFFST | | | | |
| R-0h | R-0/W1S-0h | R/W-1h | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXFFINT | RXFFINTCLR | RXFFIENA | RXFFIL | | | | |
| R-0h | W-0h | R/W-0h | R/W-1Fh | | | | |

表 24-29 SCIFFRX 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RXFFOVF | R | 0h | 接收 FIFO 溢出标志，本身不能产生中断；此条件发生在接收中断有效期间，应在接收中断中处理。RXFFOVRCLR、通道复位 SCIRST 或系统复位可清除此位。<br>0h (R) = FIFO 未溢出，只读位。<br>1h (R) = FIFO 已溢出，只读位；收到超过 16 个字，最先接收的字丢失。<br>复位类型：SYSRSn |
| 14 | RXFFOVRCLR | R-0/W1S | 0h | RXFFOVF 清除。注意：RXFFIL 和 RXFFOVF 标志执行逻辑或，发生溢出时必须同时通过 RXFFINTCLR 与 RXFFOVRCLR 清除，否则会阻止后续中断。<br>0h (W1S) = 对 RXFFOVF 无影响，读回 0。<br>1h (W1S) = 清除位 15 的 RXFFOVF。<br>复位类型：SYSRSn |
| 13 | RXFIFORESET | R/W | 1h | 接收 FIFO 复位。<br>0h (R/W) = 指针复位为 0 并保持复位。<br>1h (R/W) = 重新启用接收 FIFO。<br>复位类型：SYSRSn |
| 12-8 | RXFFST | R | 0h | FIFO 状态。<br>0h (R) = 接收 FIFO 为空。<br>1h (R) = 接收 FIFO 中有 1 个字。<br>2h (R) = 接收 FIFO 中有 2 个字。<br>3h (R) = 接收 FIFO 中有 3 个字。<br>4h (R) = 接收 FIFO 中有 4 个字。<br>5h (R) = 接收 FIFO 中有 5 个字。<br>6h (R) = 接收 FIFO 中有 6 个字。<br>7h (R) = 接收 FIFO 中有 7 个字。<br>8h (R) = 接收 FIFO 中有 8 个字。<br>9h (R) = 接收 FIFO 中有 9 个字。<br>Ah (R) = 接收 FIFO 中有 10 个字。<br>Bh (R) = 接收 FIFO 中有 11 个字。<br>Ch (R) = 接收 FIFO 中有 12 个字。<br>Dh (R) = 接收 FIFO 中有 13 个字。<br>Eh (R) = 接收 FIFO 中有 14 个字。<br>Fh (R) = 接收 FIFO 中有 15 个字。<br>10h (R) = 接收 FIFO 中有 16 个字。<br>复位类型：SYSRSn |
| 7 | RXFFINT | R | 0h | 接收 FIFO 中断，只读位。<br>0h (R) = 未发生 RXFIFO 中断。<br>1h (R) = 已发生 RXFIFO 中断。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 6 | RXFFINTCLR | W | 0h | 接收 FIFO 中断清除。注意：RXFFIL 和 RXFFOVF 标志执行逻辑或，发生溢出时必须同时通过 RXFFINTCLR 与 RXFFOVRCLR 清除，否则会阻止后续中断。<br>0h (W) = 对 RXFFINT 标志无影响，读回 0。<br>1h (W) = 清除位 7 的 RXFFINT。<br>复位类型：SYSRSn |
| 5 | RXFFIENA | R/W | 0h | 接收 FIFO 中断使能。<br>0h (R/W) = 禁用。<br>1h (R/W) = 启用；RXFFST 大于或等于 RXFFIL（位 4–0）时触发中断。<br>复位类型：SYSRSn |
| 4-0 | RXFFIL | R/W | 1Fh | 接收 FIFO 中断阈值。RXFFST4–0 大于或等于 RXFFIL4–0 时产生中断。用于产生中断的最大设置值不得超过 RX FIFO 深度。复位默认值为 11111b。用户应权衡 CPU 中断服务开销和接收 SCI 数据的有效利用，按应用设置 RXFFIL。<br>复位类型：SYSRSn |

### 24.13.2.12 SCIFFCT 寄存器

偏移地址：18h　复位值：0000h

FIFO 控制寄存器

SCIFFCT 包含自动波特率检测状态、自动波特率标志清除及 A 检测校准位。

表 24-30 SCIFFCT 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| ABD | ABDCLR | CDC | 保留 | | | | |
| R-0h | W-0h | R/W-0h | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FFTXDLY | | | | | | | |
| R/W-0h | | | | | | | |

表 24-31 SCIFFCT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | ABD | R | 0h | 自动波特率检测 ABD 位。<br>0h (R) = 检测未完成，尚未成功接收字符 A 或 a。<br>1h (R) = 硬件已在 SCI 接收寄存器检测到 A 或 a，自动检测完成。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 14 | ABDCLR | W | 0h | ABD 清除位。<br>0h (W) = 对 ABD 无影响，读回 0。<br>1h (W) = 清除位 15 的 ABD 标志。<br>复位类型：SYSRSn |
| 13 | CDC | R/W | 0h | CDC 校准 A 检测位。<br>0h (R/W) = 禁用自动波特率对齐。<br>1h (R/W) = 启用自动波特率对齐。<br>复位类型：SYSRSn |
| 12-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | FFTXDLY | R/W | 0h | FIFO 传送延迟。定义 FIFO 发送缓冲器每次向发送移位寄存器传送之间的延迟，以 SCI 串行波特率时钟周期计。8 位 FFTXDLY 字段的编码范围为 0–255 个波特率时钟周期。<br>FIFO 模式下，只有移位寄存器完成最后一位移出后，才能填充 FIFO 与移位寄存器之间的 SCITXBUF，以使传送间延迟反映到数据流；不应将 SCITXBUF 视为额外一级缓冲器。延迟发送功能有助于实现无需标准 UART 的 RTS/CTS 控制的自动流控。<br>配置 1 个停止位时，相邻帧间增加的延迟等于 FFTXDLY 设置的波特率时钟周期数；配置 2 个停止位时，等于设置值减 1。<br>复位类型：SYSRSn |

### 24.13.2.13 SCIPRI 寄存器

偏移地址：1Eh　复位值：0000h

SCI 优先级控制寄存器

SCIPRI 决定仿真暂停事件发生时的行为。

表 24-32 SCIPRI 寄存器位布局表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | FREESOFT | | 保留 | | |
| R-0h | | | R/W-0h | | R-0h | | |

表 24-33 SCIPRI 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 7-5 | RESERVED | R | 0h | 保留。 |
| 4-3 | FREESOFT | R/W | 0h | 控制仿真暂停事件（例如调试器命中断点）时的行为。外设可继续当前操作（自由运行），或在停止模式下立即停止，或完成当前接收 / 发送序列后停止。<br>0h (R/W) = 暂停时立即停止。<br>1h (R/W) = 完成当前接收 / 发送序列后停止。<br>2h (R/W) = 自由运行。<br>3h (R/W) = 自由运行。<br>复位类型：SYSRSn |
| 2-0 | RESERVED | R | 0h | 保留。 |

### 24.13.3 SCI 寄存器与 DriverLib 函数对应表

表 24-34 SCI 寄存器与 DriverLib 函数对应表

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| SCICCR | sci.c | SCI_setConfig |
| SCICCR | sci.h | SCI_setParityMode |
| SCICCR | sci.h | SCI_getParityMode |
| SCICCR | sci.h | SCI_setAddrMultiProcessorMode |
| SCICCR | sci.h | SCI_setIdleMultiProcessorMode |
| SCICCR | sci.h | SCI_getConfig |
| SCICCR | sci.h | SCI_enableLoopback |
| SCICCR | sci.h | SCI_disableLoopback |
| SCICTL1 | sci.c | SCI_enableInterrupt |
| SCICTL1 | sci.c | SCI_disableInterrupt |
| SCICTL1 | sci.c | SCI_setWakeFlag |
| SCICTL1 | sci.h | SCI_enableModule |
| SCICTL1 | sci.h | SCI_disableModule |
| SCICTL1 | sci.h | SCI_enableTxModule |
| SCICTL1 | sci.h | SCI_disableTxModule |
| SCICTL1 | sci.h | SCI_enableRxModule |
| SCICTL1 | sci.h | SCI_disableRxModule |
| SCICTL1 | sci.h | SCI_enableSleepMode |
| SCICTL1 | sci.h | SCI_disableSleepMode |
| SCICTL1 | sci.h | SCI_performSoftwareReset |
| SCIHBAUD | sci.c | SCI_setConfig |
| SCIHBAUD | sci.c | SCI_setBaud |
| SCIHBAUD | sci.h | SCI_lockAutobaud |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| SCIHBAUD | sci.h | SCI_getConfig |
| SCILBAUD | sci.c | SCI_setConfig |
| SCILBAUD | sci.c | SCI_setBaud |
| SCILBAUD | sci.h | SCI_lockAutobaud |
| SCILBAUD | sci.h | SCI_getConfig |
| SCICTL2 | sci.c | SCI_enableInterrupt |
| SCICTL2 | sci.c | SCI_disableInterrupt |
| SCICTL2 | sci.c | SCI_getInterruptStatus |
| SCICTL2 | sci.h | SCI_isSpaceAvailableNonFIFO |
| SCICTL2 | sci.h | SCI_isTransmitterBusy |
| SCIRXST | sci.c | SCI_getInterruptStatus |
| SCIRXST | sci.h | SCI_isDataAvailableNonFIFO |
| SCIRXST | sci.h | SCI_getRxStatus |
| SCIRXEMU | - | - |
| SCIRXBUF | sci.c | SCI_readCharArray |
| SCIRXBUF | sci.h | SCI_readCharBlockingFIFO |
| SCIRXBUF | sci.h | SCI_readCharBlockingNonFIFO |
| SCIRXBUF | sci.h | SCI_readCharNonBlocking |
| SCITXBUF | sci.c | SCI_writeCharArray |
| SCITXBUF | sci.h | SCI_writeCharBlockingFIFO |
| SCITXBUF | sci.h | SCI_writeCharBlockingNonFIFO |
| SCITXBUF | sci.h | SCI_writeCharNonBlocking |
| SCIFFTX | sci.c | SCI_enableInterrupt |
| SCIFFTX | sci.c | SCI_disableInterrupt |
| SCIFFTX | sci.c | SCI_getInterruptStatus |
| SCIFFTX | sci.c | SCI_clearInterruptStatus |
| SCIFFTX | sci.h | SCI_setFIFOInterruptLevel |
| SCIFFTX | sci.h | SCI_getFIFOInterruptLevel |
| SCIFFTX | sci.h | SCI_disableModule |
| SCIFFTX | sci.h | SCI_enableFIFO |
| SCIFFTX | sci.h | SCI_disableFIFO |
| SCIFFTX | sci.h | SCI_isFIFOEnabled |
| SCIFFTX | sci.h | SCI_resetTxFIFO |
| SCIFFTX | sci.h | SCI_resetChannels |
| SCIFFTX | sci.h | SCI_getTxFIFOStatus |
| SCIFFTX | sci.h | SCI_isTransmitterBusy |

| 寄存器 | 文件 | DriverLib 函数 |
| --- | --- | --- |
| SCIFFRX | sci.c | SCI_enableInterrupt |
| SCIFFRX | sci.c | SCI_disableInterrupt |
| SCIFFRX | sci.c | SCI_getInterruptStatus |
| SCIFFRX | sci.c | SCI_clearInterruptStatus |
| SCIFFRX | sci.h | SCI_setFIFOInterruptLevel |
| SCIFFRX | sci.h | SCI_getFIFOInterruptLevel |
| SCIFFRX | sci.h | SCI_enableFIFO |
| SCIFFRX | sci.h | SCI_resetRxFIFO |
| SCIFFRX | sci.h | SCI_getRxFIFOStatus |
| SCIFFRX | sci.h | SCI_getOverflowStatus |
| SCIFFRX | sci.h | SCI_clearOverflowStatus |
| SCIFFCT | sci.h | SCI_lockAutobaud |
| SCIPRI | - | - |

# 25 串行外设接口（SPI）

本章介绍串行外设接口（SPI），它是一种高速同步串行输入输出（I/O）端口，支持将可编程长度（1~16 位）的串行位流以可编程的位传输速率移入和移出设备。SPI 通常用于微控制器（MCU）与外部外设或其他控制器之间的通信，典型应用包括利用移位寄存器、显示驱动器和模数转换器（ADC）等器件实现外部 I/O 或外设扩展。SPI 支持主/从工作模式，可实现多设备通信，且该端口配备 16 级收发先进先出（FIFO）缓冲区，能有效降低 CPU 的服务开销。

## 25.1 概述

### 25.1.1 功能特性

SPI 模块的功能特性如下：

- SPISOMI：SPI 从机输出/主机输入引脚
- SPISIMO：SPI 从机输入/主机输出引脚
- SPISTE：SPI 从机发送使能引脚
- SPICLK：SPI 串行时钟引脚

注：若未使用 SPI 模块，上述四个引脚均可作为通用输入输出（GPIO）引脚使用。

- 两种工作模式：主模式和从模式
- 波特率：支持 125 种可编程速率，实际可使用的最大波特率受 SPI 引脚所用 I/O 缓冲器最大速率限制。
- 数据字长：1~16 位
- 四种时钟方案（由时钟极性位和时钟相位位控制）：
  1) 无相位延迟的下降沿：SPICLK 高电平有效，SPI 在 SPICLK 信号下降沿发送数据，上升沿接收数据。
  2) 带相位延迟的下降沿：SPICLK 高电平有效，SPI 在 SPICLK 信号下降沿前半个周期发送数据，下降沿接收数据。

3) 无相位延迟的上升沿：SPICLK 低电平有效，SPI 在 SPICLK 信号上升沿发送数据，下降沿接收数据。

4) 带相位延迟的上升沿：SPICLK 低电平有效，SPI 在 SPICLK 信号上升沿前半个周期发送数据，上升沿接收数据。

- 同时收发操作（发送功能可通过软件禁用）
- 发送和接收操作支持中断驱动或查询算法实现
- 16 级收发 FIFO 缓冲区
- 直接内存访问（DMA）支持
- 高速模式
- 延迟发送控制
- 三线 SPI 模式
- 双 SPI 模块器件中，支持数字音频接口接收模式下的 SPISTE 信号反相

### 25.1.2 框图

图中展示了 SPI CPU 接口。

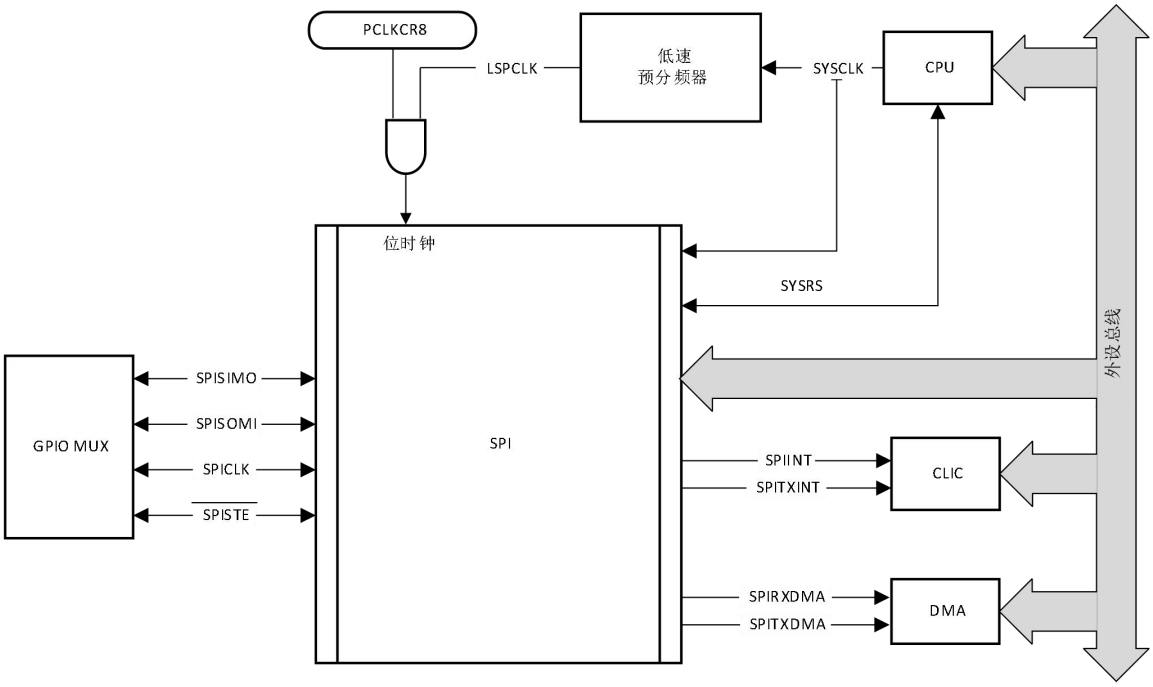



图 25-1 SPI CPU 接口框图

## 25.2 系统级集成

本节介绍适用于器件集成的各类功能，这些功能需要配置器件中的其他模块，相关内容不在本章讨论范围内。

### 25.2.1 SPI 模块信号

下表对 SPI 模块的信号进行了分类和概述。

表 25-1 SPI 模块信号概述

| 信号类型 | 信号名称 | 描述 |
|---|---|---|
| 外部信号 | SPICLK | SPI时钟 |
| | SPISIMO | SPI从机入/主机出 |
| | SPISOMI | SPI从机出/主机入 |
| | SPISTE | SPI从机发送使能 |
| 控制信号 | SPI时钟速率 | 由低速外设时钟（LSPCLK）提供 |
| 中断信号 | SPIINT/SPIRXINT | 非FIFO模式下为发送/接收中断（统称SPIINT），FIFO模式下为接收中断 |
| | SPITXINT | FIFO模式下的发送中断 |
| DMA触发信号 | SPITXDMA | 向DMA发出的发送请求 |
| | SPIRXDMA | 向DMA发出的接收请求 |

#### 25.2.1.1 特殊注意事项

当 SPI 工作在从模式时，SPISTE 信号可屏蔽杂散时钟和数据脉冲：SPISTE 为高电平时，从机无法接收数据，避免从机与主机失去同步。不建议将 SPISTE 始终接有效电平。

若 SPI 从机与主机失去同步，翻转 SPISWRESET 位可复位内部位计数器及模块中的各类状态标志。位计数器复位后，SPI 会将下一个时钟跳变解读为新传输的第一位。

#### 25.2.1.2 用 GPIO 模拟 SPISTE

在许多系统中，一个 SPI 主机需要通过多个 SPISTE 实例连接多个 SPI 从机。本 SPI 模块本身不支持多个 SPISTE 信号，但可以用 GPIO 在软件中模拟

这一行为。采用该配置时，SPI 必须配置为主机：不使用 GPIO 复用选择 SPISTE，而是把引脚配置为 GPIO 输出，每个 SPI 从机对应一个 GPIO。发送数据前，应用程序把目标 GPIO 驱动到有效电平；发送完成后，立即把该 GPIO 片选驱动到无效电平。对于共用 SPICLK、SPISIMO 和 SPISOMI 线路的多个从机，可以重复这一过程。

### 25.2.2 配置器件引脚

必须配置 GPIO 复用寄存器，才能把 SPI 外设连接到器件引脚。为避免引脚出现毛刺，应先写 GPyGMUX 位（此时保持相应的 GPyMUX 位为默认值 0），然后再把 GPyMUX 寄存器写成所需值。

部分 I/O 功能由 GPIO 寄存器独立于本外设定义。对于输入信号，必须把相应的 GPxQSELn 寄存器位设为 11b，使 GPIO 输入限定为异步模式。内部上拉可通过 GPyPUD 寄存器配置。

GPIO 复用与设置的更多说明，见第 7 章“通用输入/输出接口（GPIO）”。

#### 25.2.2.1 高速模式所需 GPIO

SPI 的高速模式在全部 GPIO 复用选项上都可用。要启用高速增强功能，应将 SPICCR.HS_MODE 置 1，并确保引脚上的电容负载不超过器件数据手册规定的值。

不工作在高速模式，或引脚电容负载超过数据手册规定值时，应将 SPICCR.HS_MODE 清 0。

### 25.2.3 SPI 中断

本节介绍SPI模块的可用中断，该模块包含两个中断线：SPIINT/SPIRXINT和SPITXINT。非FIFO模式下，所有可用中断合并为单个SPIINT中断；FIFO模式下，可独立产生SPIRXINT和SPITXINT中断。

#### 25.2.3.1 SPIINT/SPIRXINT

非 FIFO 模式下，接收中断记为 SPIINT；启用 FIFO 增强功能后，接收中断记为 SPIRXINT。两种模式使用同一接收中断请求通道。

非 FIFO 模式：两种情况可触发中断——传输完成（INT_FLAG）或接收器溢出（OVERRUN_FLAG），均触发 SPIINT 中断。

传输完成标志（INT_FLAG）：表示 SPI 已完成最后一位的发送/接收，可进行服务，该位置 1 的同时，接收字符被存入接收缓冲区（SPIRXBUF）；若 SPIINTENA 位置 1，INT_FLAG 会在 SPIINT 向量上触发中断。

接收器溢出标志（OVERRUN_FLAG）：表示前一个字符尚未从缓冲区读取，新的发送/接收操作已完成；若 OVERRUNINTENA 位置 1 且 OVERRUN_FLAG 此前已清 0，该标志会在 SPIINT 向量上触发中断。

FIFO 模式下，当接收 FIFO 当前状态（RXFFST）大于或等于接收 FIFO 中断阈值（RXFFIL）时，接收 FIFO 中断标志（RXFFINT）置 1；若 RXFFIENA=1，则产生 SPIRXINT 中断请求。

#### 25.2.3.2 SPITXINT

非 FIFO 模式下，SPITXINT 中断不可用。

FIFO 模式下，当发送 FIFO 当前状态（TXFFST）小于或等于发送 FIFO 中断阈值（TXFFIL）时，发送 FIFO 中断标志（TXFFINT）置 1；若 TXFFIENA=1，则产生 SPITXINT 中断请求。

图中展示了这些控制位对 SPI 中断产生的影响。

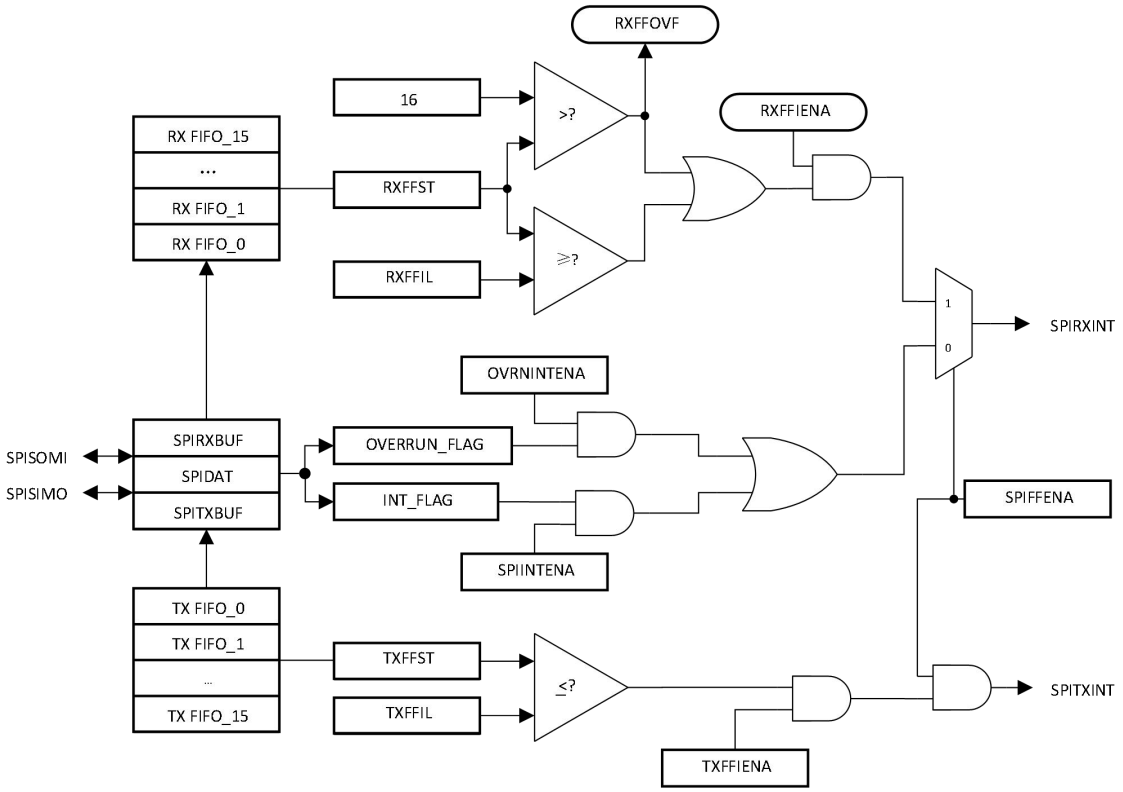



图 25-2 SPI 中断标志与使能逻辑生成框图

表 25-2 SPI 中断标志模式

| FIFO选项 | SPI中断源 | 中断标志 | 中断使能 | FIFO使能（SPIFFENA） | 中断线(1) |
|---|---|---|---|---|---|
| 无FIFO的SPI | 接收溢出 | RXOVRN | OVRNINTENA | 0 | SPIRXINT |
| | 数据接收 | SPIINT | SPIINTENA | 0 | SPIRXINT |
| | 发送空 | SPIINT | SPIINTENA | 0 | SPIRXINT |
| SPI FIFO模式 | FIFO接收 | RXFFIL | RXFFIENA | 1 | SPIRXINT |
| | 发送空 | TXFFIL | TXFFIENA | 1 | SPITXINT |

(1) 非 FIFO 模式下，SPIRXINT 与 SPIINT 为同一中断。

### 25.2.4 DMA 支持

CPU 和 DMA 均可通过内部外设总线访问 SPI 数据寄存器，且仅支持 16 位寄存器读写。每个 SPI 模块可产生两个 DMA 事件：SPITXDMA 和 SPIRXDMA，通过合理配置 SPIFFTX.TXFFIL 和 SPIFFRX.RXFFIL 可控制 DMA 事件的产生：当

TXFFST 小于中断阈值（TXFFIL）时，SPITXDMA 触发；当 RXFFST 大于或等于中断阈值（RXFFIL）时，SPIRXDMA 触发。

只有启用 FIFO 增强功能，SPI 才能产生 DMA 触发信号。

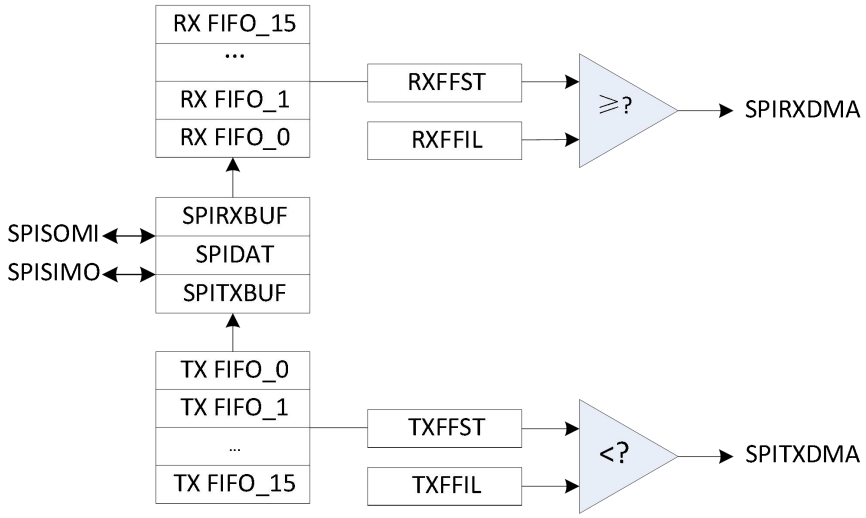



图 25-3 SPI DMA 触发框图

## 25.3 SPI 工作原理

本节介绍 SPI 的各类工作模式，包括工作模式、中断、数据格式、时钟源和初始化的说明，并给出数据传输的典型时序图。

### 25.3.1 工作原理概述

图中为两个控制器（主/从）之间通过 SPI 通信的典型连接方式。

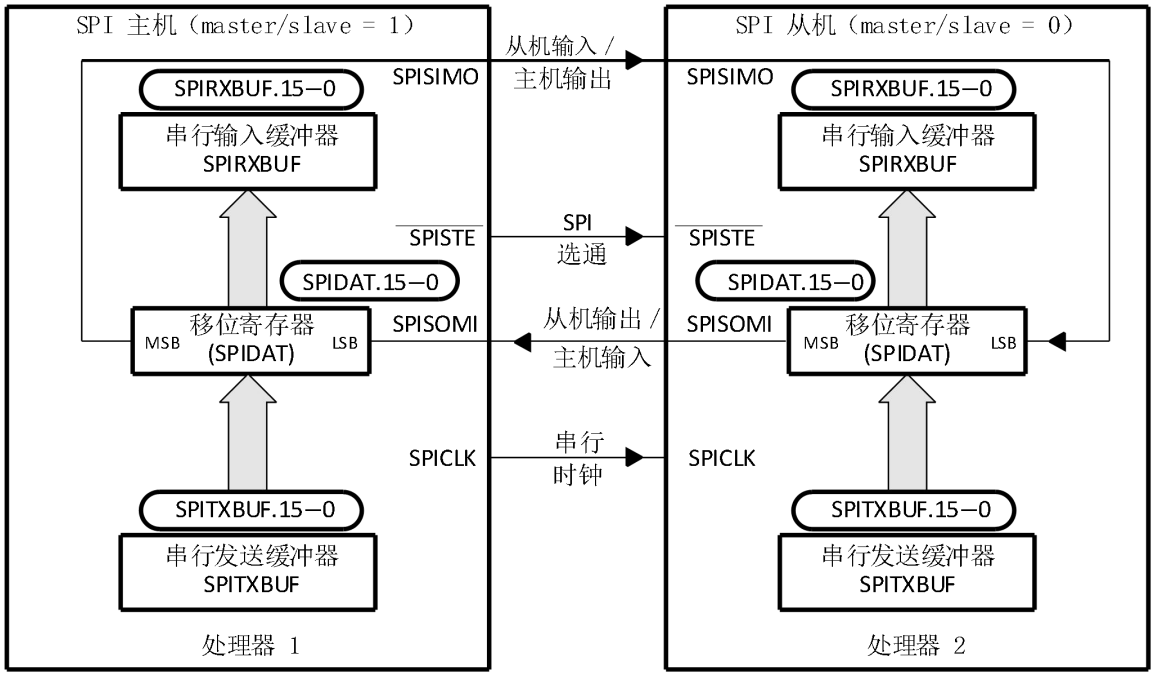



图 25-4 SPI 主/从连接示意图

主机通过发送 SPICLK 信号实现数据传输，主从机均在 SPICLK 的一个边沿将数据移出移位寄存器，在另一个边沿将数据锁存移入移位寄存器。若 CLK_PHASE 位置 1，数据会在 SPICLK 跳变前半个周期进行收发，因此主从机可同时完成数据的发送和接收，由应用程序判断数据为有效数据还是伪数据。

数据传输有三种方式：

- 主机发送有效数据，从机发送伪数据；
- 主机和从机均发送有效数据；
- 主机发送伪数据，从机发送有效数据。

由于主机控制 SPICLK 信号，可随时发起数据传输，而主机检测从机是否准备好广播数据则由软件实现。

SPI 工作于主模式或从模式，由 MASTER_SLAVE 位选择工作模式及 SPICLK 信号的来源。

### 25.3.2 主模式

主模式下（MASTER_SLAVE=1），SPI 通过 SPICLK 引脚为整个串行通信网络提供串行时钟，通过 SPISIMO 引脚输出数据，从 SPISOMI 引脚锁存数据。

网络的收发位传输速率由 SPIBRR 寄存器决定，该寄存器支持 125 种不同的数据传输速率。

向 SPIDAT 或 SPITXBUF 写入数据，会触发 SPI 从 SPISIMO 引脚以最高有效位（MSB）优先的方式发送数据，同时，SPISOMI 引脚上的数据逐位移入 SPIDAT 的最低位（LSB）端。当指定位数的数据传输完成后，接收数据会被移入接收缓冲区（SPIRXBUF），供 CPU 读取，且数据在 SPIRXBUF 中采用右对齐存储。

当指定位数的数据通过 SPIDAT 完成移位后，会发生以下事件：

1. SPIDAT 中的内容被移入 SPIRXBUF；

2. INT_FLAG 位置 1；

3. 若发送缓冲区 SPITXBUF 中有有效数据（由发送缓冲区满标志 BUFFULL_FLAG 指示），则该数据被移入 SPIDAT 并发送；否则，当 SPIDAT 中所有位移出后，SPICLK 停止；

4. 若 SPIINTENA 位置 1，触发中断。

典型应用中，SPISTE 引脚作为 SPI 从机器件的片选使能引脚，主机向从机发送数据前将该引脚拉低，传输完成后拉回高电平。

图中为 SPI 主模式的框图，展示了主模式下的基本控制模块。

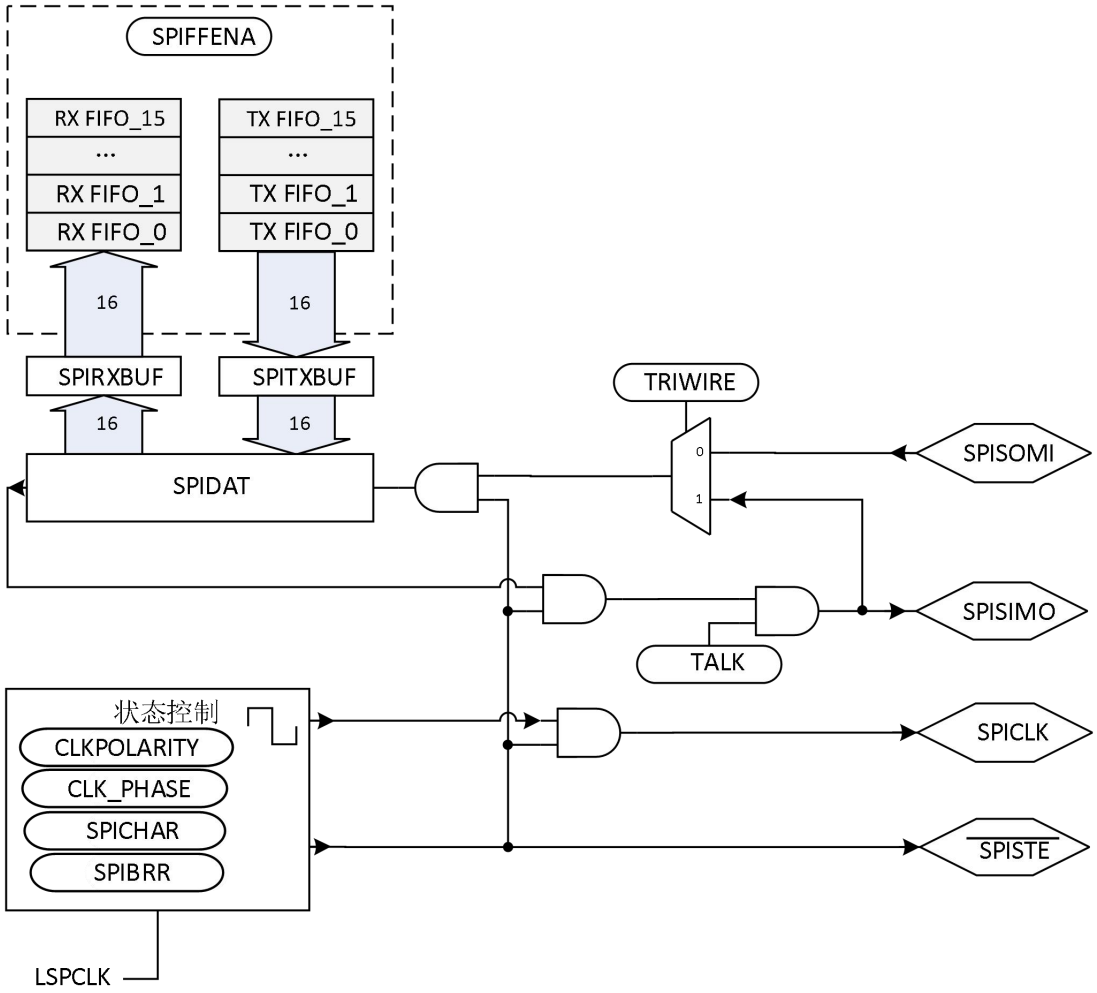



图 25-5 SPI 模块主模式配置框图

### 25.3.3 从模式

从模式下（MASTER_SLAVE=0），SPI 从 SPISOMI 引脚移出数据，从 SPISIMO 引脚移入数据，SPICLK 引脚作为串行移位时钟的输入，由外部网络主机提供，传输速率由该时钟决定，且 SPICLK 输入频率不得超过低速外设时钟（LSPCLK）频率的 1/4。

当从外部网络主机接收到 SPICLK 信号的有效边沿时，写入 SPIDAT 或 SPITXBUF 的数据被发送至网络。当 SPIDAT 中当前字符的所有位移出后，写入 SPITXBUF 寄存器的字符会被复制到 SPIDAT；若此前无字符复制到 SPIDAT，则写入 SPITXBUF 的任意字符会立即复制到 SPIDAT；若此前已有字符复制到

SPIDAT，则需等待 SPIDAT 中当前字符的所有位移出后，SPITXBUF 中的数据才会复制到 SPIDAT。

接收数据时，SPI 等待网络主机发送 SPICLK 信号，然后将 SPISIMO 引脚上的数据移入 SPIDAT。若从机需要同时发送数据且 SPIDAT 尚未加载数据，则必须在 SPICLK 信号开始前将字符写入 SPITXBUF。

当 TALK 位清 0 时，数据发送被禁用，输出引脚（SPISOMI）进入高阻态；若在传输过程中清 0TALK 位，尽管 SPISOMI 被强制置为高阻态，当前字符仍会完成传输，确保 SPI 能正确接收输入数据。利用 TALK 位，可将多个从机连接到同一网络，但同一时间仅允许一个从机驱动 SPISOMI 线路。

SPISTE 引脚作为从机选择引脚：SPISTE 为低电平时，SPI 从机可向串行数据线传输数据；SPISTE 为高电平时，SPI 从机的串行移位寄存器停止工作，串行输出引脚进入高阻态。该设计支持多个从机连接到同一网络，且同一时间仅选择一个从机。

图中为 SPI 从模式的框图，展示了从模式下的基本控制模块。

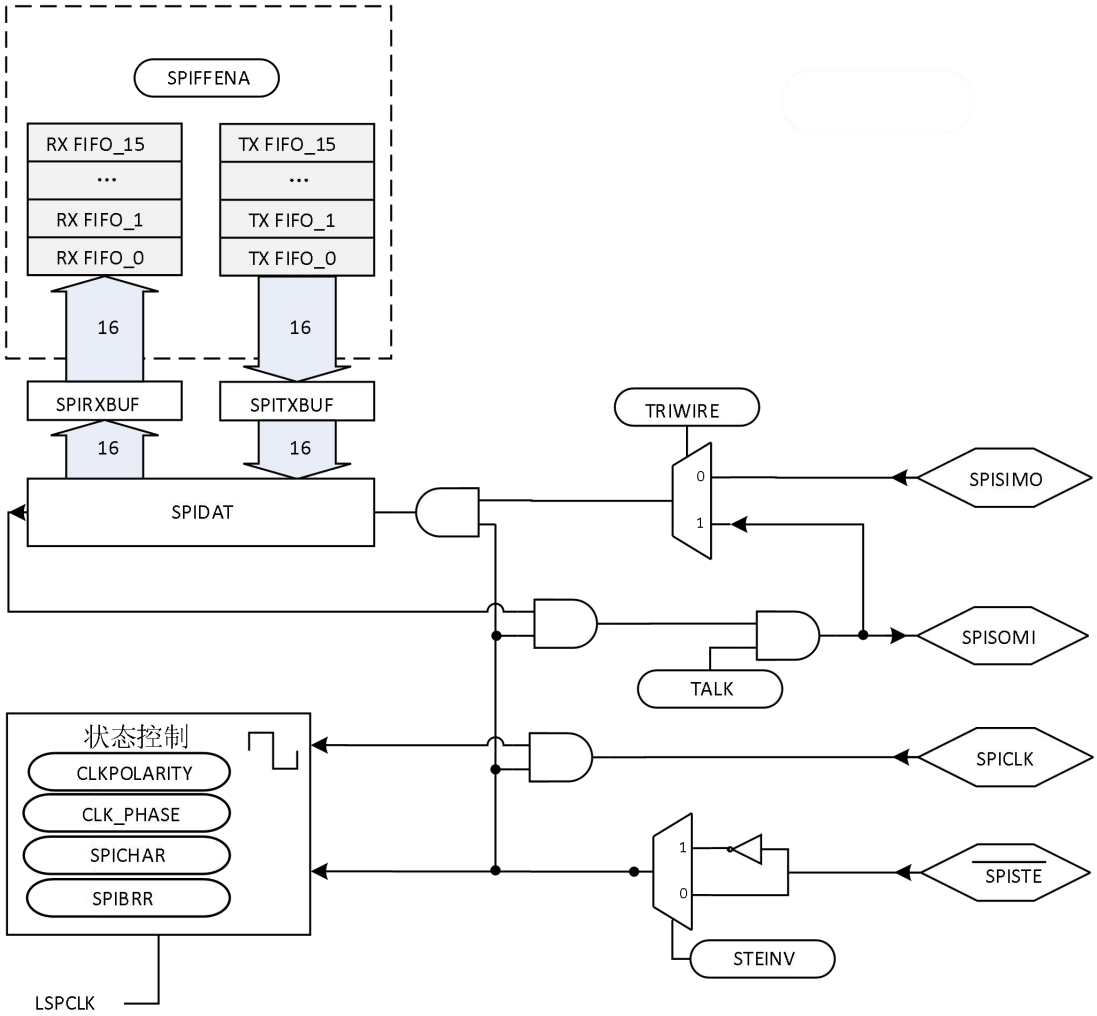



图 25-6 SPI 模块从模式配置框图

## 25.3.4 数据格式

SPICCR.SPICHAR 位段（4 位）指定数据字符的位数（1~16 位），状态控制逻辑根据该信息对收发的位数进行计数，判断一个完整字符是否处理完成。

对于位数小于 16 位的字符，遵循以下规则：

- 向 SPIDAT 和 SPITXBUF 写入数据时，需采用左对齐方式；
- 从 SPIRXBUF 读取数据时，数据为右对齐格式；
- SPIRXBUF 中存储最近接收的右对齐字符，以及此前传输中左移剩余的任意位（见例 25-1）。

例 25-1 SPIRXBUF 的数据位传输示例

条件：

传输字符长度=1 位（由 SPICHAR 位指定）；

SPIDAT 当前值=737Bh。




图 25-7 数据格式示意图

（注：x 为 SPISOMI 引脚的电平值，高电平为 1，低电平为 0，示例基于主模式。）

### 25.3.5 波特率选择

SPI 模块支持 125 种不同的波特率和四种不同的时钟方案，根据 SPI 工作在主/从模式，SPICLK 引脚分别输出 SPI 时钟信号或接收外部 SPI 时钟信号。

从模式：SPICLK 引脚从外部源接收时钟，频率不得超过 LSPCLK 频率的 1/4；

主模式：SPI 生成时钟并通过 SPICLK 引脚输出，频率不得超过 LSPCLK 频率的 1/4。

注：波特率不得超过 GPIO 的最大额定翻转频率。GPIO 最大翻转频率请参见器件数据手册。

例 25-2 介绍 SPI 波特率的确定方法，例 25-3 介绍标准 SPI 模式（HS_MODE=0）下 SPI 模块的波特率计算方法。

例 25-2 波特率确定方法

当 SPIBRR=3~127 时：

$$\text{SPI 波特率} = \frac{\text{LSPCLK}}{\text{SPIBRR} + 1} \quad (25\text{-}1)$$

当 SPIBRR=0、1 或 2 时：

$$\text{SPI 波特率} = \frac{\text{LSPCLK}}{4} \quad (25\text{-}2)$$

其中：

LSPCLK：器件的低速外设时钟频率；

SPIBRR：SPI 主设备中 SPIBRR 寄存器的内容。

确定 SPIBRR 的写入值时，需知晓器件的系统时钟（LSPCLK）频率（器件特定）和工作波特率。

例 25-3 非高速模式（HS_MODE=0）下的波特率计算

$$\begin{aligned}\text{SPI 波特率} &= \frac{\text{LSPCLK}}{\text{SPIBRR}+1}\ （\text{LSPCLK} = 50\ \text{MHz}） \\ &= \frac{50\times10^6}{3+1} \\ &= 12.5\ \text{Mbps}\end{aligned} \qquad （25\text{-}3）$$

### 25.3.6 SPI 时钟方案

时钟极性选择位（CLKPOLARITY）和时钟相位选择位（CLK_PHASE）控制 SPICLK 引脚的四种时钟方案：CLKPOLARITY 选择时钟的有效边沿（上升沿/下降沿），CLK_PHASE 选择时钟的半周期延迟，四种时钟方案如下：

- 无延迟的下降沿：SPI 在 SPICLK 下降沿发送数据，上升沿接收数据；
- 带延迟的下降沿：SPI 在 SPICLK 下降沿前半个周期发送数据，下降沿接收数据；
- 无延迟的上升沿：SPI 在 SPICLK 上升沿发送数据，下降沿接收数据；
- 带延迟的上升沿：SPI 在 SPICLK 上升沿前半个周期发送数据，上升沿接收数据。

SPI 时钟方案的选择方法见表 25-3，四种时钟方案对应收发数据的示例见图 25-8。

表 25-3 SPI 时钟方案选择指南

| SPICLK方案 | CLKPOLARITY(1) | CLK_PHASE(1) |
|---|---|---|
| 无延迟的上升沿 | 0 | 0 |
| 带延迟的上升沿 | 0 | 1 |
| 无延迟的下降沿 | 1 | 0 |
| 带延迟的下降沿 | 1 | 1 |

不同厂商对 CLK_PHASE 和 CLKPOLARITY 的定义存在差异，为保证正常工作，需根据目标波形确定时钟相位和时钟极性的配置。

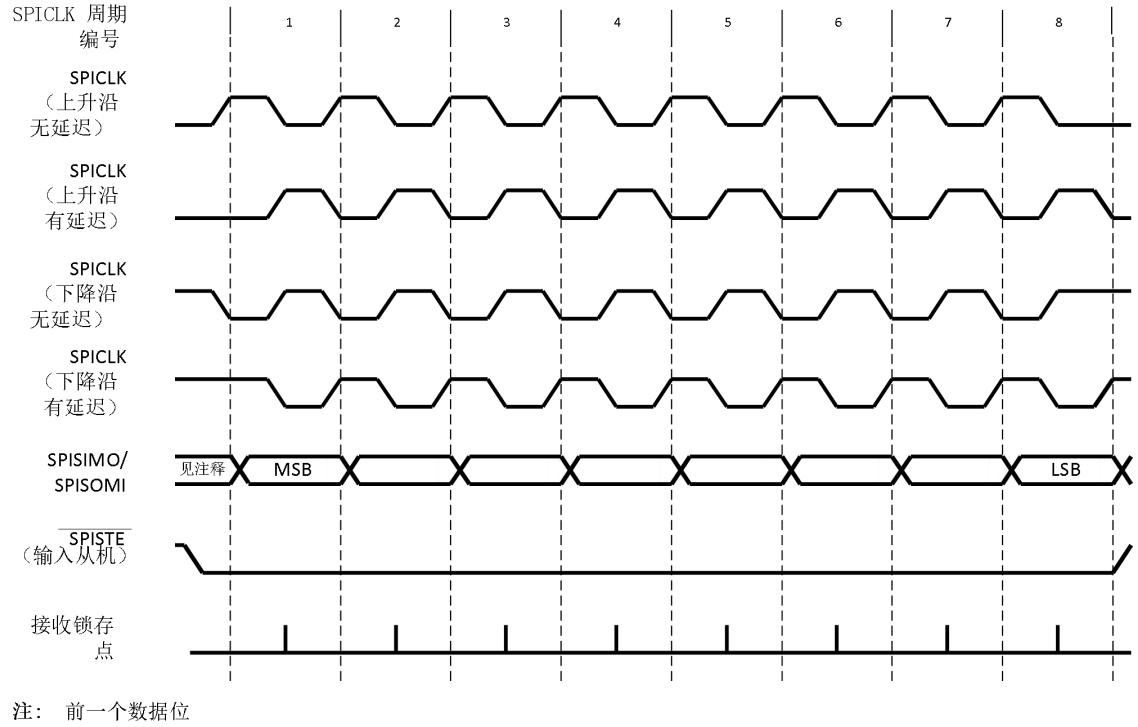



图 25-8 SPICLK 信号选项时序图

仅当（SPIBRR+1）为偶数值时，SPICLK 保持对称性；当（SPIBRR+1）为奇数值且 SPIBRR>3 时，SPICLK 变为非对称：

若 CLKPOLARITY 位清 0，SPICLK 的低脉冲比高脉冲长一个 LSPCLK 周期；

若 CLKPOLARITY 位置 1，SPICLK 的高脉冲比低脉冲长一个 LSPCLK 周期（见图 25-9）。

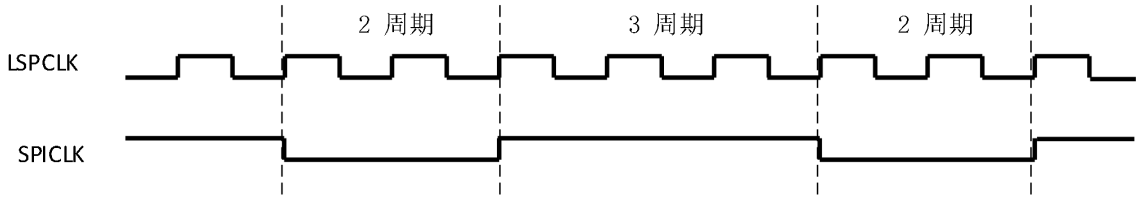



图 25-9 当(SPIBRR+1)为奇数、SPIBRR>3 且 CLKPOLARITY=1 时，SPICLK 与 LSPCLK 的特性关系图

## 25.3.7 SPI FIFO 说明

以下步骤介绍 FIFO 的功能特性，并为 SPI FIFO 的编程提供指导：

1. 复位：复位后，SPI 以标准 SPI 模式上电，FIFO 功能禁用，FIFO 寄存器 SPIFFTX、SPIFFRX 和 SPIFFCT 保持非激活状态；

2. 标准 SPI 模式：标准 CPU SPI 模式以 SPIINT/SPIRXINT 作为中断源；

3. 模式切换：置 1SPIFFTX 寄存器中的 SPIFFENA 位，可启用 FIFO 模式，SPIRST 可在操作的任意阶段复位 FIFO 模式；

4. 激活寄存器：所有 SPI 寄存器和 SPI FIFO 寄存器（SPIFFTX、SPIFFRX、SPIFFCT）均处于激活状态；

5. 中断：FIFO 模式包含两个中断——发送 FIFO 中断（SPITXINT）和接收 FIFO 中断（SPIRXINT）。SPIRXINT 为 SPI FIFO 接收、接收错误和接收 FIFO 溢出的公共中断，标准 SPI 中收发共用的单个 SPIINT 中断被禁用，该中断向量作为 SPI 接收 FIFO 中断使用。

6. 缓冲区：收发缓冲区各扩展有 16 字的 FIFO，标准 SPI 的 1 字发送缓冲区（SPITXBUF）作为发送 FIFO 与移位寄存器之间的过渡缓冲区，仅当移位寄存器的最后一位移出后，才会从发送 FIFO 向该缓冲区加载数据；

7. 延迟传输：FIFO 中的发送数据向发送移位寄存器传输的速率可编程，SPIFFCT 寄存器的 7-0 位（TXDLY7-TXDLY0）定义了数据传输的间隔延迟，延迟单位为 SPI 串行时钟周期。8 位寄存器可设置 0~255 个 SPICLK 周期的延迟：延迟为 0 时，SPI 模块以连续模式传输数据，FIFO 数据连续移出；延迟为 255 个时钟周期时，SPI 模块以最大延迟模式传输数据，FIFO 中相邻数据的移出间隔为 255 个 SPICLK 周期。可编程延迟实现了与各类低速 SPI 外设（如 EEPROM、ADC、DAC 等）的无缝连接；

8. FIFO 状态位：收发 FIFO 均有状态位（TXFFST/RXFFST），实时指示 FIFO 中的有效数据字数。将发送 FIFO 复位位（TXFIFO）和接收 FIFO 复位位（RXFIFORESET）清 0，可将 FIFO 指针复位为 0 并保持复位；置 1 后，FIFO 从起始位置恢复工作；

9. 可编程中断阈值：收发 FIFO 均可产生 CPU 中断和 DMA 触发：当发送 FIFO 状态位（TXFFST）小于或等于中断触发阈值位（TXFFIL）时，产生发送中断（SPITXINT）；当接收 FIFO 状态位（RXFFST）大于或等于中断触发阈值位（RXFFIL）时，产生接收中断（SPIRXINT）。该设计为 SPI 的收发部分提供了可编程的中断触发机制，接收 FIFO 和发送 FIFO 的触发阈值位默认值分别为 11111b（即 1Fh）和 00000b。

### 25.3.8 SPI DMA 传输

#### 25.3.8.1 基于 DMA 的 SPI 发送数据

当发送 FIFO 配合 DMA 使用时，DMA 突发长度（DMA_BURST_SIZE）不得大于 16-TXFFIL，防止 DMA 向已满的 FIFO 写入数据而导致数据丢失；突发长度超过该上限的配置不推荐使用。

完整的数据传输步骤如下：

1. 计算待发送数据的总字数[NUM_WORDS]；

2. 确定发送 FIFO 阈值[TXFFIL]；

3. 计算 DMA 传输次数[DMA_TRANSFER_SIZE]；

4. 计算 DMA 突发长度[DMA_BURST_SIZE]；

5. 根据计算值配置 DMA；

6. 根据计算值配置带 FIFO 的 SPI。

示例：通过 DMA 向 SPI 传输 128 字数据

NUM_WORDS：128

TXFFIL：8

DMA_TRANSFER_SIZE：(NUM_WORDS/TXFFIL) -1 = (128/8)-1 = 15（共 16 次传输）

DMA_BURST_SIZE：(16-TXFFIL) -1 = (16-8)-1 =7（每次突发传输 8 字）

注：为保证 DMA 配置正常，避免将 TXFFIL 设为 0h 或 10h。

#### 25.3.8.2 基于 DMA 的 SPI 接收数据

当接收 FIFO 配合 DMA 使用时，DMA 突发长度（BURST_SIZE）不得大于 RXFFIL，防止 DMA 从空的 FIFO 读取数据。为保证 DMA 能正确从接收 FIFO 接收所有数据，DMA 突发长度可等于 RXFFIL，且为 SPI 总传输次数的整数约数。

完整的数据接收步骤如下：

计算待接收数据的总字数[NUM_WORDS]；

确定所需的 FIFO 阈值[RXFFIL]；

计算 DMA 总传输次数[DMA_TRANSFER_SIZE]；

计算 DMA 突发长度[DMA_BURST_SIZE]；

根据计算值配置 DMA；

根据计算值配置带 FIFO 的 SPI。

示例：通过 DMA 从 SPI 接收 200 字数据

NUM_WORDS：200

RXFFIL：4

DMA_TRANSFER_SIZE：(NUM_WORDS/RXFFIL) -1 = (200/4)-1 =49（共 50 次传输）

DMA_BURST_SIZE：RXFFIL-1 =3（每次突发传输 4 字）

注：为保证 DMA 配置正常，避免将 RXFFIL 设为 0h。

### 25.3.9 SPI 高速模式

SPI 模块的全双工通信速率最高可达 LSPCLK/4（LSPCLK 等于 SYSCLK），最大额定速率请参考器件数据手册。

要达到最大全双工速率，设计需遵循以下限制：

仅支持单主机对单从机的配置；

引脚负载不得超过器件数据手册规定值。

### 25.3.10 SPI 三线模式说明

三线 SPI 模式支持通过 3 个引脚实现 SPI 通信，替代标准的 4 引脚方案。

主模式：TRIWIRE 位置 1 启用三线 SPI 模式后，SPISIMOx 变为双向的 SPIMOMIx（SPI 主机出/主机入）引脚，SPISOMIx 不再被 SPI 使用；

从模式：TRIWIRE 位置 1 后，SPISOMIx 变为双向的 SPISISOx（SPI 从机入/从机出）引脚，SPISIMOx 不再被 SPI 使用。

表 25-4 列出了主/从模式下，三线与四线 SPI 模式的引脚功能差异。

表 25-4 四线与三线 SPI 的引脚功能对比

| 四线SPI | 三线SPI（主模式） | 三线SPI（从模式） |
|---|---|---|
| SPICLKx | SPICLKx | SPICLKx |
| SPISTEx | SPISTEx | SPISTEx |
| SPISIMOx | SPIMOMIx（双向） | 空闲 |
| SPISOMIx | 空闲 | SPISISOx（双向） |

三线模式下，SPI 的收发路径内部连接，因此 SPI 模块发送的任何数据都会被自身接收，应用程序需执行伪读操作，清除 SPI 数据寄存器中额外接收的数据。

TALK 位在三线 SPI 模式中起重要作用：发送数据时需置 1 该位，读取数据前需清 0 该位。主模式下发起读操作时，应用程序需在清 0TALK 位的同时，向 SPI 数据寄存器 SPIDAT 写入伪数据（无数据从 SPIMOMI 引脚发送），再从接收缓冲寄存器 SPIRXBUF 读取数据。

三线主模式和三线从模式的示意图分别见图 25-10 和图 25-11。

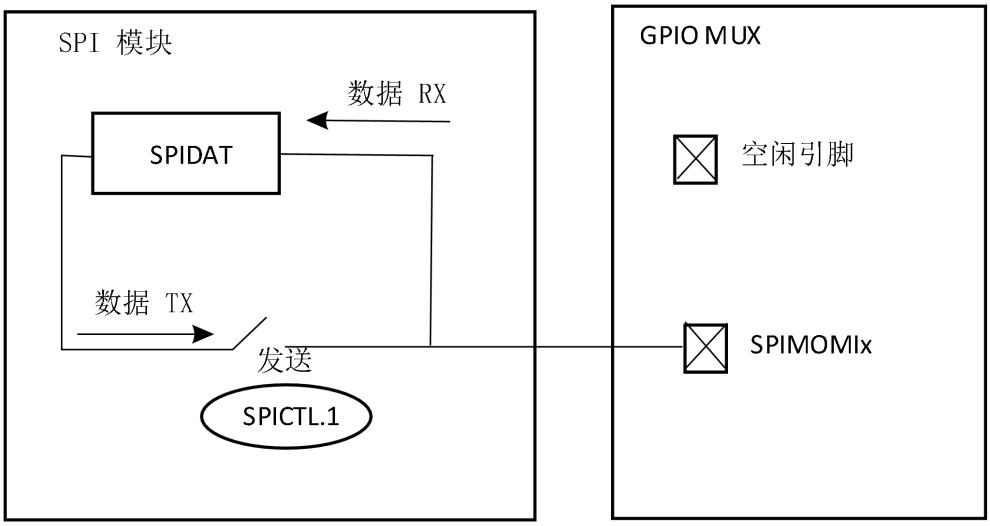



图 25-10 SPI 三线主模式示意图

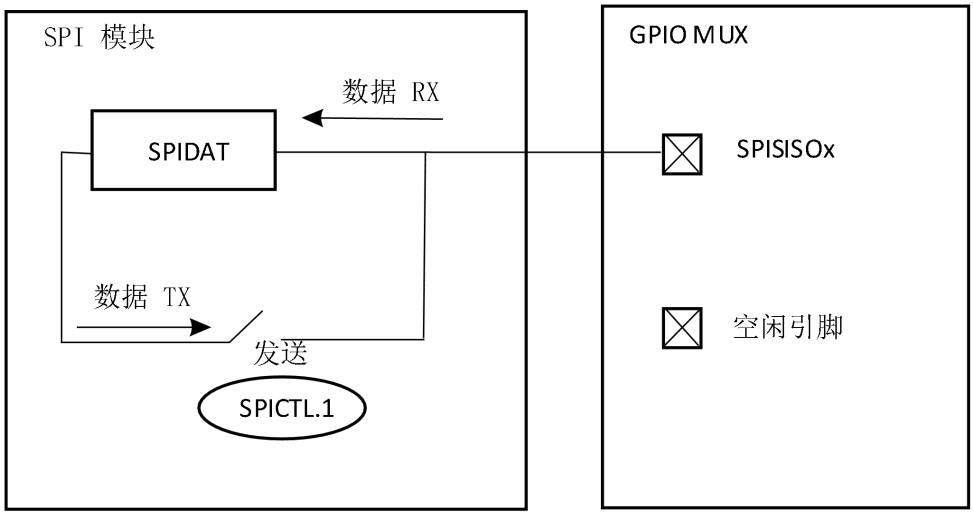



图 25-11 SPI 三线从模式示意图

表 25-5 列出了不同 SPI 模式下，TALK 位置 1/清 0 时的数据收发情况。

表 25-5 SPI 三线模式的引脚配置

| 引脚模式 | SPIPRI[TRIWIRE] | SPICTL[TALK] | SPISIMO | SPISOMI |
|---|---|---|---|---|
| 主模式 | 0（四线） | 任意 | 发送 | 接收 |
| | 1（三线） | 0 | 接收 | 与SPI断开连接 |
| | 1（三线） | 1 | 收发双向 | 与SPI断开连接 |
| 从模式 | 0（四线） | 任意 | 接收 | 发送 |
| | 1（三线） | 0 | 与SPI断开连接 | 接收 |
| | 1（三线） | 1 | 与SPI断开连接 | 收发双向 |

## 25.4 编程步骤

本节介绍 SPI 各类工作模式的配置步骤。

### 25.4.1 复位后的初始化

系统复位后，SPI 外设被强制配置为以下默认状态：

配置为从模块（MASTER_SLAVE=0）；

发送功能禁用（TALK=0）；

数据在 SPICLK 信号的下降沿锁存输入；

字符长度默认为 1 位；

SPI 中断禁用；

SPIDAT 中的数据复位为 0000h。

### 25.4.2 SPI 配置

本节介绍 SPI 模块的工作配置步骤，为防止初始化修改过程中或修改后出现意外情况，需先清 0SPISWRESET 位，再进行初始化修改，初始化完成后置 1 该位。当 SPI 保持复位状态（SPISWRESET=0）时，可按任意顺序修改配置，以下为逻辑上的 SPI 配置步骤（除 SPISWRESET 位外，SPI 寄存器支持 16 位单次写入，顺序无强制要求）。

注：通信过程中不得修改 SPI 配置。

SPI 配置步骤：

1. 清 0SPI 软件复位位（SPISWRESET），将 SPI 置为复位状态；

2. 按需求配置 SPI：

- 选择主/从模式（MASTER_SLAVE）；
- 选择 SPICLK 的极性和相位（CLKPOLARITY、CLK_PHASE）；
- 设置目标波特率（SPIBRR）；
- 按需启用高速模式（HS_MODE）；
- 设置 SPI 字符长度（SPICHAR）；

- 清 0SPI 标志位（OVERRUN_FLAG、INT_FLAG）；
- 按需启用 SPISTE 信号反相（STEINV）；
- 按需启用三线模式（TRIWIRE）；
- 若使用 FIFO 增强功能：
- 启用 FIFO 增强功能（SPIFFENA）；
- 清 0FIFO 标志位（TXFFINTCLR、RXFFOVFCLR、RXFFINTCLR）；
- 释放收发 FIFO 复位（TXFIFO、RXFIFORESET）；
- 释放 SPI FIFO 通道的复位（SPIRST）；

3. 若使用中断：

- 非 FIFO 模式：启用接收器溢出中断和/或 SPI 中断（OVERRUNINTENA、SPIINTENA）；
- FIFO 模式：设置收发中断阈值（TXFFIL、RXFFIL），然后启用中断（TXFFIENA、RXFFIENA）；

4. 置 1SPISWRESET 位，释放 SPI 的复位状态。

### 25.4.3 配置高速模式

要达到最高额定速率，需进行以下设置（本例假设器件工作在 100 MHz）：将 LSPCLK 设为 SYSCLK（ClkCfgRegs.LOSPCP.bit.LSPCLKDIV = 0）；在 GPIO_CTRL_REGS 中选择相应的引脚复用选项；在 SPI 配置过程中将 HS_MODE 置 1（SpiaRegs.SPICCR.bit.HS_MODE = 0x1）；将 SPIBRR 设为 3（SpiaRegs.SPIBRR = 0x3），此时 SPICLK = LSPCLK/(SPIBRR+1) = 25 MHz。

其余配置与常规 SPI 工作方式没有区别：数据收发、DMA 操作和中断均照常工作。

### 25.4.4 数据传输示例

当 SPICLK 为非对称时钟时，时序特征与图 25-12 基本一致，仅在 SPICLK 的低脉冲（CLKPOLARITY=0）或高脉冲（CLKPOLARITY=1）期间，每位数据的传输时长增加一个 LSPCLK 周期。

注：下图仅适用于 8 位 SPI，不适用于支持 16 位数据的设备，仅作演示使用。

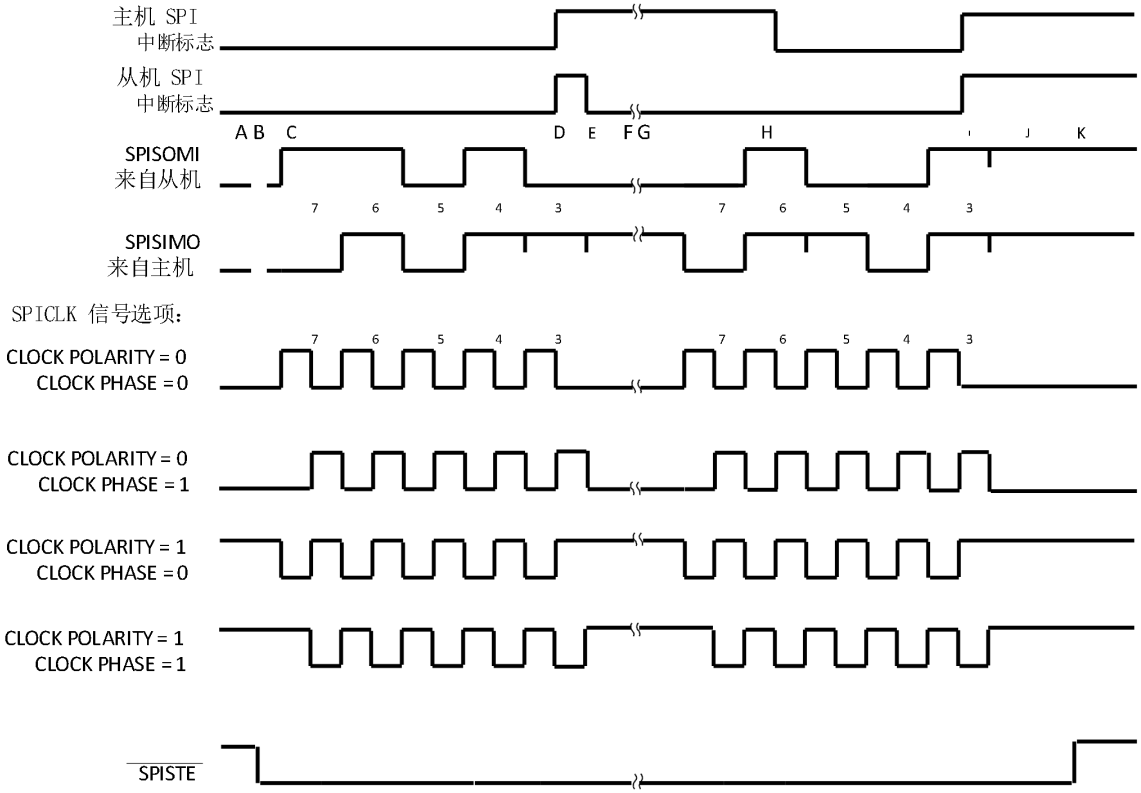



图 25-12 每字符 5 位的 SPI 数据传输时序图

时序说明：

A. 从机向 SPIDAT 写入 0D0h，等待主机移出数据；

B. 主机将从机的 SPISTE 信号拉低（有效）；

C. 主机向 SPIDAT 写入 058h，启动传输过程；

D. 第一个字节传输完成，中断标志位置 1；

E. 从机从 SPIRXBUF 读取 0Bh（右对齐）；

F. 从机向 SPIDAT 写入 04Ch，等待主机移出数据；

G. 主机向 SPIDAT 写入 06Ch，启动传输过程；

H. 主机从 SPIRXBUF 读取 01Ah（右对齐）；

I. 第二个字节传输完成，中断标志位置 1；

J. 主机读取 89h，从机读取 8Dh（均来自各自的 SPIRXBUF），用户软件屏蔽未使用位后，主机接收 09h，从机接收 0Dh；

K. 主机将从机的 SPISTE 信号拉高（无效）。

### 25.4.5 SPI 三线模式代码示例

除标准 SPI 初始化外，配置三线模式需将 TRIWIRE 位（SPIPRI.0）置 1。初始化完成后，三线主/从模式的数据收发需注意以下事项，示例代码演示了相关操作。

三线主模式：需将 SPICLKx、SPISTEx、SPISIMOx 引脚配置为 SPI 引脚（SPISOMIx 可配置为非 SPI 引脚）。主模式发送数据时，主机会接收自身发送的数据（三线模式下 SPISIMOx 与 SPISOMIx 内部连接），因此每次发送数据后，需清除接收缓冲区中的无效数据。

例 25-4 三线主模式发送数据

```
uint16_t data;
uint16_t dummy;
SpiaRegs.SPICTL.bit.TALK = 1;   // 启用发送路径
SpiaRegs.SPITXBUF = data;       // 主机发送数据
while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {}   // 等待数据接收完成
dummy = SpiaRegs.SPIRXBUF;      // 清除无效数据（接收数据与发送数据一致）
```

三线主模式接收数据时，主机需清 0TALK 位（SPICTL.1）关闭发送路径，然后发送伪数据以启动从机的数据传输。由于 TALK 位清 0，与发送模式不同，主机的伪数据不会出现在 SPISIMOx 引脚上，且主机不会接收该伪数据，而是接收从机发送的有效数据。

例 25-5 三线主模式接收数据

```
uint16_t rdata;
uint16_t dummy;
SpiaRegs.SPICTL.bit.TALK = 0;   // 禁用发送路径
SpiaRegs.SPITXBUF = dummy;      // 发送伪数据启动传输（TALK=0，数据不会发送至
                                SPISIMOA引脚）
while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {}   // 等待数据接收完成
rdata = SpiaRegs.SPIRXBUF;      // 主机读取数据
```

三线从模式：需将 SPICLKx、SPISTEx、SPISOMIx 引脚配置为 SPI 引脚（SPISIMOx 可配置为非 SPI 引脚）。与主模式类似，从机发送数据时会接收自身发送的数据，需清除接收缓冲区中的无效数据。

例 25-6 三线从模式发送数据

```
uint16_t data;
uint16_t dummy;
SpiaRegs.SPICTL.bit.TALK = 1;    // 启用发送路径
SpiaRegs.SPITXBUF = data;        // 从机发送数据
while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {}    // 等待数据接收完成
dummy = SpiaRegs.SPIRXBUF;       // 清除无效数据
```

与三线主模式一致，从模式接收数据时需清 0TALK 位，否则从机将正常接收数据。

例 25-7 三线从模式接收数据

```
uint16_t rdata;
SpiaRegs.SPICTL.bit.TALK = 0;    // 禁用发送路径
while(SpiaRegs.SPISTS.bit.INT_FLAG !=1) {}    // 等待数据接收完成
rdata = SpiaRegs.SPIRXBUF;       // 从机读取数据
```

### 25.4.6 数字音频传输中的 SPI STEINV 位

在配备双 SPI 模块的器件中，置 1 其中一个 SPI 模块的 STEINV 位，可使两个 SPI 模块在从模式下接收数字音频的左右声道数据：接收标准低电平有效 SPISTE 信号的 SPI 模块存储右声道数据，接收反相高电平有效 SPISTE 信号的 SPI 模块存储主机发送的左声道数据。

从数字音频接口接收器接收数字音频数据时，SPI 模块的连接方式见图 25-13。

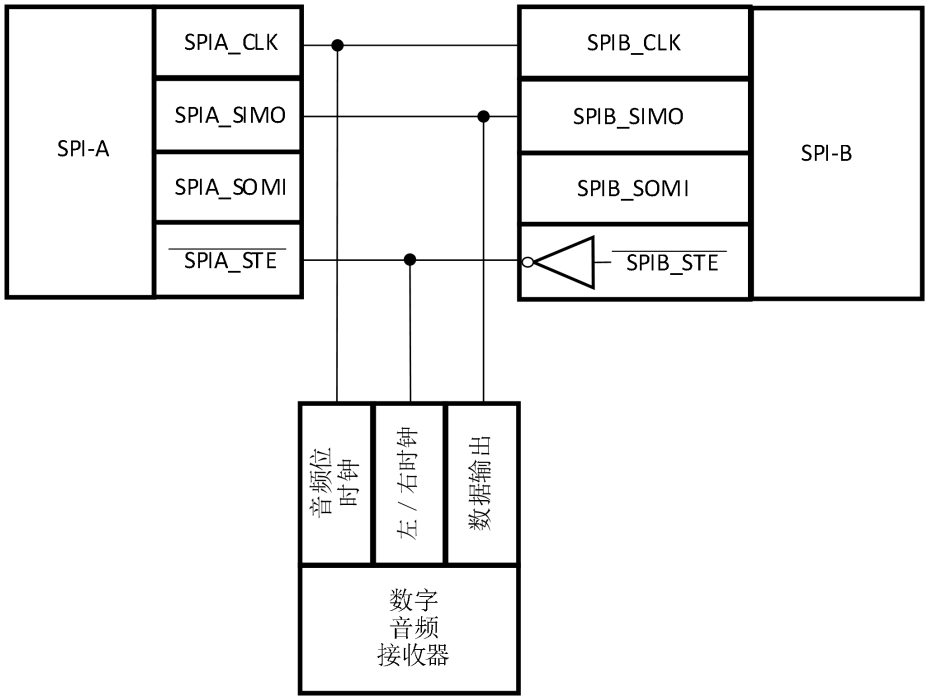



图 25-13 双 SPI 模块的数字音频接收器配置示意图

注：该配置仅适用于从模式（MASTER_SLAVE=0），SPI 配置为主模式（MASTER_SLAVE=1）时，STEINV 位对 SPISTE 引脚无影响。

标准 SPI 的时序要求限制了双 SPI 配置（带 STEINV 位）支持的数字音频接口格式数量，SPI 的时序要求请参考器件数据手册的电气规格。当 SPI 时钟相位配置为 CLKPOLARITY=0、CLK_PHASE=1（数据在时钟上升沿锁存）时，支持标准的右对齐数字音频接口数据格式（见图 25-14）。

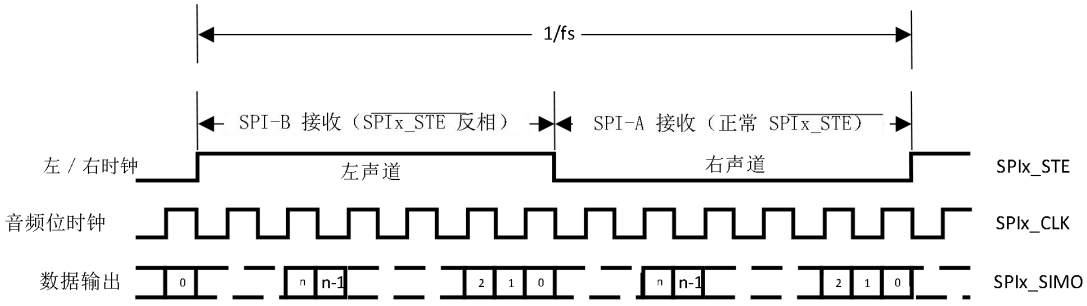



图 25-14 标准右对齐数字音频数据格式时序图

## 25.5 寄存器描述

本节介绍串行外设接口的寄存器。

### 25.5.1 SPI 基地址表

表 25-6 SPI 基地址表

| 实例 | 结构体 | 驱动库名称 | 基地址 | DMA支持 | HIC支持 | 流水线保护 |
|---|---|---|---|---|---|---|
| SpiaRegs | SPI_REGS | SPIA_BASE | 0x0062_0000 | 是 | 是 | 是 |
| SpibRegs | SPI_REGS | SPIB_BASE | 0x0062_0020 | 是 | 是 | 是 |

### 25.5.2 SPI_REGS 寄存器

表 25-7 列出了 SPI_REGS 的内存映射寄存器，表中未列出的寄存器偏移地址均为保留地址，不得修改其寄存器内容。

表 25-7 SPI_REGS 寄存器列表

| 偏移地址 | 缩写 | 寄存器名称 |
|---|---|---|
| 0h | SPICCR | SPI配置控制寄存器 |
| 2h | SPICTL | SPI操作控制寄存器 |
| 4h | SPISTS | SPI状态寄存器 |
| 8h | SPIBRR | SPI波特率寄存器 |
| Ch | SPIRXEMU | SPI仿真缓冲寄存器 |
| Eh | SPIRXBUF | SPI串行输入缓冲寄存器 |
| 10h | SPITXBUF | SPI串行输出缓冲寄存器 |
| 12h | SPIDAT | SPI串行数据寄存器 |
| 14h | SPIFFTX | SPI FIFO发送寄存器 |
| 16h | SPIFFRX | SPI FIFO接收寄存器 |
| 18h | SPIFFCT | SPI FIFO控制寄存器 |
| 1Eh | SPIPRI | SPI优先级控制寄存器 |

复杂的位访问类型采用编码形式展示，表 25-8 列出了本节使用的访问类型编码。

表 25-8 SPI_REGS 访问类型编码

| 访问类型 | 编码 | 描述 |
|---|---|---|
| 读类型 | R | 只读 |
| | RC | 读清0 |
| 写类型 | W | 只写 |
| | W1C | 写1清0 |
| 复位/默认值 | -n | 复位后/默认值 |

### 25.5.2.1 SPICCR 寄存器

偏移地址：0h　复位值：0000h

SPI 配置控制寄存器

SPICCR 控制 SPI 运行配置。

表 25-9 SPICCR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SPISWRESET | CLKPOLARITY | HS_MODE | SPILBK | SPICHAR | | | |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | | | |

表 25-10 SPICCR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7 | SPISWRESET | R/W | 0h | SPI 软件复位。更改配置前应清除此位，恢复运行前再置位。<br>0h (R/W) = 将运行标志初始化为复位状态，清除 RECEIVER OVERRUN（SPISTS.7）、SPI INT FLAG（SPISTS.6）和 TXBUF FULL（SPISTS.5）；SPISTE 变为无效，SPICLK 不论极性如何立即驱动为 0。SPI 配置保持不变。<br>1h (R/W) = SPI 可发送或接收下一个字符。SPI SW RESET 为 0 时写入发送器的字符不会在此位置位后移出，必须向串行数据寄存器重新写入字符。此位置位后一个 SPICLK 周期，SPICLK 恢复非活动状态。<br>复位类型：SYSRSn |
| 6 | CLKPOLARITY | R/W | 0h | 移位时钟极性。此位与 CLOCK PHASE（SPICTL.3）一起控制 SPICLK 的四种时钟方案。<br>0h (R/W) = 数据在上升沿输出、下降沿输入；无 SPI 数据发送时 SPICLK 为低。具体边沿取决于 CLOCK PHASE：为 0 时在上升沿输出、下降沿锁存；为 1 时在首个上升沿前半个周期及后续下降沿输出，在上升沿锁存输入。<br>1h (R/W) = 数据在下降沿输出、上升沿输入；无数据发送时 SPICLK 为高。CLOCK PHASE 为 0 时在下降沿输出、上升沿锁存；为 1 时在首个下降沿前半个周期及后续上升沿输出，在下降 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 沿锁存输入。<br>复位类型：SYSRSn |
| 5 | HS_MODE | R/W | 0h | 高速模式使能。应在 GPxGMUX/GPxMUX 中选择正确 GPIO。<br>0h (R/W) = 禁用高速模式，复位默认值。<br>1h (R/W) = 启用高速模式。<br>复位类型：SYSRSn |
| 4 | SPILBK | R/W | 0h | SPI 回环模式选择。用于器件测试时验证模块，仅在 SPI 主模式下有效。<br>0h (R/W) = 禁用回环，复位默认值。<br>1h (R/W) = 启用回环，SIMO/SOMI 内部连接，用于模块自测。<br>复位类型：SYSRSn |
| 3-0 | SPICHAR | R/W | 0h | 字符长度控制。此 4 位决定一次移位序列中作为一个字符移入或移出的位数。SPICHAR = 字长 − 1。<br>0h (R/W) = 1 位字。<br>1h (R/W) = 2 位字。<br>7h (R/W) = 8 位字。<br>Fh (R/W) = 16 位字。<br>复位类型：SYSRSn |

### 25.5.2.2 SPICTL 寄存器

偏移地址：2h　复位值：0000h

SPI 操作控制寄存器

SPICTL 控制数据传送、中断产生、SPICLK 相位及主 / 从运行模式。

表 25-11 SPICTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | OVERRUNINTENA | CLK_PHASE | MASTER_SLAVE | TALK | SPIINTENA |
| R-0h | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 25-12 SPICTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-5 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 4 | OVERRUNINTENA | R/W | 0h | 接收溢出中断使能。置位后，硬件置 RECEIVER OVERRUN（SPISTS.7）时产生中断；此标志和 SPI INT FLAG（SPISTS.6）共用中断向量。<br>0h (R/W) = 禁用 RECEIVER OVERRUN 中断。<br>1h (R/W) = 启用 RECEIVER_OVERRUN 中断。<br>复位类型：SYSRSn |
| 3 | CLK_PHASE | R/W | 0h | SPI 时钟相位选择。与 CLOCK POLARITY（SPICCR.6）组合产生四种时钟方案。CLOCK PHASE 为高时，不论使用哪一种 SPI 模式，主机或从机都在写 SPIDAT 后、SPICLK 首个边沿前提供第一位数据。<br>0h (R/W) = 正常时钟方案，取决于 CLOCK POLARITY。<br>1h (R/W) = SPICLK 延迟半个周期，极性由 CLOCK POLARITY 决定。<br>复位类型：SYSRSn |
| 2 | MASTER_SLAVE | R/W | 0h | SPI 网络模式控制。决定主机或从机模式。复位初始化时自动配置为从机。<br>0h (R/W) = 从机。<br>1h (R/W) = 主机。<br>复位类型：SYSRSn |
| 1 | TALK | R/W | 0h | 发送使能。TALK 可将串行数据输出置为高阻态，禁用主机或从机发送。传送期间禁用时，发送移位寄存器继续工作直到前一个字符移完；禁用后仍可接收字符并更新状态标志。系统复位清除 TALK。<br>0h (R/W) = 禁用发送：从模式下 SPISOMI 若未配置为通用 I/O 则进入高阻态；主模式下 SPISIMO 若未配置为通用 I/O 则进入高阻态。<br>1h (R/W) = 启用发送。使用 4 引脚方案时，应启用接收器的 SPISTEn 输入。<br>复位类型：SYSRSn |
| 0 | SPIINTENA | R/W | 0h | SPI 中断使能。控制发送 / 接收中断的产生，不影响 SPI INT FLAG（SPISTS.6）。<br>0h (R/W) = 禁用中断。<br>1h (R/W) = 启用中断。<br>复位类型：SYSRSn |

### 25.5.2.3 SPISTS 寄存器

偏移地址：4h　复位值：0000h

SPI 状态寄存器

SPISTS 包含中断和状态位。

表 25-13 SPISTS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OVERRUN_FLAG | INT_FLAG | BUFFULL_FLAG | 保留 | | | | |
| W1C-0h | RC-0h | R-0h | R-0h | | | | |

表 25-14 SPISTS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7 | OVERRUN_FLAG | W1C | 0h | SPI 接收溢出标志，仅可读或清除。前一字符尚未从缓冲器读取，新的接收或发送操作已完成时由硬件置位。清除方式：向此位写 1；向 SPI SW RESET（SPICCR.7）写 0；系统复位。OVERRUN INT ENA（SPICTL.4）置位时，首次置此标志只请求一次中断；标志已置位后再次溢出不会增加中断。因此每次溢出后需向 SPISTS.7 写 1 清除，才能产生新的溢出中断。若 ISR 未清除该标志，退出后不会立即再次进入溢出中断。<br>0h (R/W) = 未发生接收溢出。<br>1h (R/W) = 上个字符在应用读取前被 SPI 覆盖，因而丢失；写 1 清除。此标志与 SPI INT FLAG（SPISTS.6）共用向量，应在 ISR 中清除，以明确下次接收中断的来源。<br>复位类型：SYSRSn |
| 6 | INT_FLAG | RC | 0h | SPI 中断标志，只读。硬件置位表示已完成最后一位发送或接收，等待服务；同时将接收字符放入缓冲器。SPI INT ENA（SPICTL.0）置位时，此标志请求中断。清除方式：读取 SPIRXBUF；向 SPI SW RESET（SPICCR.7）写 0；系统复位。<br>注意：启用 FIFO 时不应使用此位，因为将接收字从 SPIRXBUF 复制到接收 FIFO 的内部过程会清除此位，应改用 FIFO 状态或中断位。<br>0h (R/W) = 未接收或发送完整字。<br>1h (R/W) = 已完成最后一位发送或接收，等待服务。<br>复位类型：SYSRSn |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 5 | BUFFULL_FLAG | R | 0h | SPI 发送缓冲器满标志，只读。向 SPITXBUF 写字符时置 1；前一字符移出完成、此字符自动装入 SPIDAT 时清除。<br>0h (R/W) = 发送缓冲器未满。<br>1h (R/W) = 发送缓冲器已满。<br>复位类型：SYSRSn |
| 4-0 | RESERVED | R | 0h | 保留。 |

### 25.5.2.4 SPIBRR 寄存器

偏移地址：8h　复位值：0000h

SPI 波特率寄存器

SPIBRR 包含波特率选择位。

表 25-15 SPIBRR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | SPI_BIT_RATE | | | | | | |
| R-0h | R/W-0h | | | | | | |

表 25-16 SPIBRR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-7 | RESERVED | R | 0h | 保留。 |
| 6-0 | SPI_BIT_RATE | R/W | 0h | SPI 波特率控制。主模式下决定位传输速率，可选择 125 种由 CPU 时钟 LSPCLK 决定的速率。每个 SPICLK 周期移位一个数据位，SPICLK 为引脚输出的波特率时钟。<br>从模式下由主机通过 SPICLK 引脚提供时钟，此字段不影响 SPICLK；输入时钟频率不得超过从机 LSPCLK/4。主模式下由 SPI 产生时钟并输出至 SPICLK。<br>SPIBRR = 3 至 127：SPI 波特率 = LSPCLK / (SPIBRR + 1)。<br>SPIBRR = 0、1 或 2：SPI 波特率 = LSPCLK / 4。<br>3h (R/W) = LSPCLK/4。<br>4h (R/W) = LSPCLK/5。<br>7Eh (R/W) = LSPCLK/127。<br>7Fh (R/W) = LSPCLK/128。<br>复位类型：SYSRSn |

### 25.5.2.5 SPIRXEMU 寄存器

偏移地址：Ch　复位值：0000h

SPI 仿真缓冲寄存器

SPIRXEMU 包含接收数据，读取不会清除 SPISTS 中的 SPI INT FLAG。它并非独立的实际寄存器，而是供仿真器读取 SPIRXBUF 内容且不清除标志的虚拟地址。

表 25-17 SPIRXEMU 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| ERXBn | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ERXBn | | | | | | | |
| R-0h | | | | | | | |

表 25-18 SPIRXEMU 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | ERXBn | R | 0h | 仿真缓冲器接收数据。与 SPIRXBUF 基本相同，但读取不清除 SPI INT FLAG（SPISTS.6）。SPIDAT 收到完整字符后，将其传送至 SPIRXEMU 和 SPIRXBUF 供读取，同时置位 SPI INT FLAG。<br>此镜像寄存器用于支持仿真。仿真器正常运行时会不断读取控制寄存器以更新屏幕；读取 SPIRXBUF 会清除中断标志，读取 SPIRXEMU 则不会。因此使用 SPIRXEMU 可更准确地模拟 SPI 实际运行，建议正常仿真运行模式下查看此寄存器。<br>复位类型：SYSRSn |

### 25.5.2.6 SPIRXBUF 寄存器

偏移地址：Eh　复位值：0000h

SPI 串行输入缓冲寄存器

SPIRXBUF 包含接收数据。读取会清除 SPISTS 的 SPI INT FLAG；启用 FIFO 时，还会递减 SPIFFRX 的 RXFFST 计数器。

表 25-19 SPIRXBUF 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RXBn | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXBn | | | | | | | |
| R-0h | | | | | | | |

表 25-20 SPIRXBUF 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | RXBn | R | 0h | 接收数据。SPIDAT 收到完整字符后传送至 SPIRXBUF 供读取，同时置位 SPI INT FLAG（SPISTS.6）。由于数据以最高有效位优先移入 SPI，因此在此寄存器中右对齐存储。<br>复位类型：SYSRSn |

### 25.5.2.7 SPITXBUF 寄存器

偏移地址：10h　复位值：0000h

SPI 串行输出缓冲寄存器

SPITXBUF 保存下一个待发送字符。写入后置位 SPISTS 的 TX BUF FULL；当前字符发送完成后自动将内容装入 SPIDAT 并清除标志。若当前没有传送，写入数据直接进入 SPIDAT，TX BUF FULL 不置位。在主模式下，没有当前传送时，写此寄存器与写 SPIDAT 一样启动传送。

表 25-21 SPITXBUF 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| TXBn | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXBn | | | | | | | |
| R/W-0h | | | | | | | |

表 25-22 SPITXBUF 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | TXBn | R/W | 0h | 发送数据缓冲器，保存下一个待发送字符。当前字符发送完成且 TX BUF FULL 置位时，自动传送至 SPIDAT 并清除该标志。写入 SPITXBUF 的数据必须左对齐。<br>复位类型：SYSRSn |

### 25.5.2.8 SPIDAT 寄存器

偏移地址：12h　复位值：0000h

SPI 串行数据寄存器

SPIDAT 是发送和接收移位寄存器。写入的数据在后续 SPICLK 周期中从 MSB 端移出；每移出一位，同时从 LSB 端移入一位。

表 25-23 SPIDAT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SDATn | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SDATn | | | | | | | |
| R/W-0h | | | | | | | |

表 25-24 SPIDAT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-0 | SDATn | R/W | 0h | 串行数据移位寄存器。<br>• TALK（SPICTL.1）置位时，为串行输出引脚提供数据。<br>• 主模式下启动数据传送。启动时应检查第 25.5.2.1 节 CLOCK POLARITY（SPICCR.6）和第 25.5.2.2 节 CLOCK PHASE（SPICTL.3）的要求。主模式下向 SPIDAT 写伪数据可启动接收序列。<br>不足 16 位字符不会由硬件自动对齐，因此发送数据必须左对齐写入，接收数据应右对齐读取。<br>复位类型：SYSRSn |

### 25.5.2.9 SPIFFTX 寄存器

偏移地址：14h　复位值：A000h

SPI FIFO 发送寄存器

SPIFFTX 包含输出 FIFO 的控制和状态位，包括复位、中断阈值、占用状态、中断使能及清除。

表 25-25 SPIFFTX 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| SPIRST | SPIFFENA | TXFIFO | TXFFST | | | | |
| R/W-1h | R/W-0h | R/W-1h | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXFFINT | TXFFINTCLR | TXFFIENA | TXFFIL | | | | |
| R-0h | W-0h | R/W-0h | R/W-0h | | | | |

表 25-26 SPIFFTX 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | SPIRST | R/W | 1h | SPI 复位。<br>0h (R/W) = 复位发送和接收通道，FIFO 寄存器配置位保持不变。<br>1h (R/W) = FIFO 可恢复发送或接收，不影响 SPI 寄存器位。<br>复位类型：SYSRSn |
| 14 | SPIFFENA | R/W | 0h | SPI FIFO 增强功能使能。<br>0h (R/W) = 禁用。<br>1h (R/W) = 启用。<br>复位类型：SYSRSn |
| 13 | TXFIFO | R/W | 1h | TX FIFO 复位。<br>0h (R/W) = 指针复位为 0 并保持复位。<br>1h (R/W) = 解除发送 FIFO 复位。<br>复位类型：SYSRSn |
| 12-8 | TXFFST | R | 0h | 发送 FIFO 状态。<br>0h (R/W) = 空。<br>1h (R/W) = 1 个字。<br>2h (R/W) = 2 个字。<br>10h (R/W) = 16 个字，最大值。<br>1Fh (R/W) = 保留。<br>复位类型：SYSRSn |
| 7 | TXFFINT | R | 0h | TX FIFO 中断标志，只读。<br>0h (R/W) = 未发生中断。<br>1h (R/W) = 已发生中断。<br>复位类型：SYSRSn |
| 6 | TXFFINTCLR | W | 0h | TXFIFO 中断清除。<br>0h (R/W) = 对 TXFIFINT 无影响，读回 0。<br>1h (R/W) = 清除 SPIFFTX[TXFFINT]。<br>复位类型：SYSRSn |
| 5 | TXFFIENA | R/W | 0h | TX FIFO 中断使能。<br>0h (R/W) = 禁用基于 TXFFIL 匹配（小于或等于）的中断。<br>1h (R/W) = 启用基于 TXFFIL 匹配（小于或等 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 于）的中断。<br>复位类型：SYSRSn |
| 4-0 | TXFFIL | R/W | 0h | 发送 FIFO 中断阈值。TXFFST4–0 小于或等于 TXFFIL4–0 时产生中断。<br>0h (R/W) = TX 缓冲器中无剩余字时请求中断。<br>1h (R/W) = 剩余 1 个字或为空时请求中断。<br>2h (R/W) = 剩余不超过 2 个字时请求中断。<br>10h (R/W) = 剩余不超过 16 个字时请求中断。<br>1Fh (R/W) = 保留。<br>复位类型：SYSRSn |

### 25.5.2.10 SPIFFRX 寄存器

偏移地址：16h　复位值：201Fh

SPI FIFO 接收寄存器

SPIFFRX 包含输入 FIFO 的控制和状态位，包括复位、中断阈值、占用状态、中断使能及清除。

表 25-27 SPIFFRX 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RXFFOVF | RXFFOVFCLR | RXFIFORESET | RXFFST | | | | |
| R-0h | W-0h | R/W-1h | R-0h | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RXFFINT | RXFFINTCLR | RXFFIENA | RXFFIL | | | | |
| R-0h | W-0h | R/W-0h | R/W-1Fh | | | | |

表 25-28 SPIFFRX 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15 | RXFFOVF | R | 0h | 接收 FIFO 溢出标志，只读。<br>0h (R/W) = 未溢出。<br>1h (R/W) = 已溢出；FIFO 收到超过 16 个字，最先收到的字丢失。<br>复位类型：SYSRSn |
| 14 | RXFFOVFCLR | W | 0h | 接收 FIFO 溢出清除。<br>0h (R/W) = 对 RXFFOVF 无影响，读回 0。<br>1h (R/W) = 清除 SPIFFRX[RXFFOVF]。<br>复位类型：SYSRSn |
| 13 | RXFIFORESET | R/W | 1h | 接收 FIFO 复位。<br>0h (R/W) = 指针复位为 0 并保持复位。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1h (R/W) = 重新启用接收 FIFO。<br>复位类型：SYSRSn |
| 12-8 | RXFFST | R | 0h | 接收 FIFO 状态。<br>0h (R/W) = 空。<br>1h (R/W) = 1 个字。<br>2h (R/W) = 2 个字。<br>10h (R/W) = 16 个字，最大值。<br>1Fh (R/W) = 保留。<br>复位类型：SYSRSn |
| 7 | RXFFINT | R | 0h | 接收 FIFO 中断标志，只读。<br>0h (R/W) = 未发生 RXFIFO 中断。<br>1h (R/W) = 已发生 RXFIFO 中断。<br>复位类型：SYSRSn |
| 6 | RXFFINTCLR | W | 0h | 接收 FIFO 中断清除。<br>0h (R/W) = 对 RXFIFINT 无影响，读回 0。<br>1h (R/W) = 清除 SPIFFRX[RXFFINT]。<br>复位类型：SYSRSn |
| 5 | RXFFIENA | R/W | 0h | RX FIFO 中断使能。<br>0h (R/W) = 禁用基于 RXFFIL 匹配（大于或等于）的中断。<br>1h (R/W) = 启用基于 RXFFIL 匹配（大于或等于）的中断。<br>复位类型：SYSRSn |
| 4-0 | RXFFIL | R/W | 1Fh | 接收 FIFO 中断阈值。RXFFST4–0 大于或等于 RXFFIL4–0 时产生中断。复位默认值 11111，使通常为空的接收 FIFO 不会频繁中断。<br>0h (R/W) = RX 缓冲器有 0 个或更多字时请求中断。<br>1h (R/W) = 有 1 个或更多字时请求中断。<br>2h (R/W) = 有 2 个或更多字时请求中断。<br>10h (R/W) = 有 16 个字时请求中断。<br>1Fh (R/W) = 保留。<br>复位类型：SYSRSn |

### 25.5.2.11 SPIFFCT 寄存器

偏移地址：18h 复位值：0000h

SPI FIFO 控制寄存器

SPIFFCT 控制 FIFO 发送延迟。

表 25-29 SPIFFCT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TXDLY | | | | | | | |
| R/W-0h | | | | | | | |

表 25-30 SPIFFCT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RESERVED | R | 0h | 保留。 |
| 7-0 | TXDLY | R/W | 0h | FIFO 发送延迟。定义从 FIFO 发送缓冲器向发送移位寄存器每次传送之间的延迟，以 SPI 串行时钟周期计。8 位字段可定义 0 至 255 个周期。<br>FIFO 模式下，必须等移位寄存器移完最后一位后才能填充 FIFO 与移位寄存器之间的 TXBUF，使传送间延迟反映在数据流中。<br>TXBUF 不应被视为额外一级 FIFO。<br>0h (R/W) = 前一个字发送完成后，立即将下一个 FIFO 字传送至 SPITXBUF。<br>1h (R/W) = 完成后 1 个串行时钟周期传送。<br>2h (R/W) = 完成后 2 个串行时钟周期传送。<br>FFh (R/W) = 完成后 255 个串行时钟周期传送。<br>复位类型：SYSRSn |

### 25.5.2.12 SPIPRI 寄存器

偏移地址：1Eh　复位值：0000h

SPI 优先级控制寄存器

SPIPRI 控制 SPI 辅助功能，包括仿真控制、SPISTE 反相及三线控制。

表 25-31 SPIPRI 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | SOFT | FREE | 保留 | | STEINV | TRIWIRE |
| R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h |

表 25-32 SPIPRI 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-7 | RESERVED | R | 0h | 保留。 |
| 6 | RESERVED | R/W | 0h | 保留。 |
| 5 | SOFT | R/W | 0h | 仿真软运行，仅 FREE 为 0 时有效。<br>0h (R/W) = TSUSPEND 有效时在位流中途停止。若不发生系统复位而解除 TSUSPEND，则继续移出 DATBUF 中待发送的位。例如 SPIDAT 已移出 8 位中的 3 位时冻结；随后不复位 SPI 而解除 TSUSPEND，会从停止位置（第 4 位）继续，并从该点发送 8 位。<br>1h (R/W) = 若仿真暂停发生在首个 SPICLK 脉冲之前，传送不会开始；若发生在传送开始后，则完成数据移出。开始时刻取决于波特率。标准 SPI 模式下，移位寄存器和缓冲器中的字发送完，即 TXBUF 和 SPIDAT 均空后停止；FIFO 模式下，TX FIFO 和 SPIDAT 均空后停止。<br>复位类型：SYSRSn |
| 4 | FREE | R/W | 0h | 仿真自由运行。仿真暂停（例如命中断点）时可继续运行，或在停止模式下立即停止 / 完成当前接收发送序列后停止。<br>0h (R/W) = 由 SOFT 选择仿真模式。<br>1h (R/W) = 自由运行，无论是否暂停及暂停时刻如何，SPI 均继续工作。<br>复位类型：SYSRSn |
| 3-2 | RESERVED | R | 0h | 保留。 |
| 1 | STEINV | R/W | 0h | SPISTEn 反相。在具有两个 SPI 模块的器件上，将其中一个模块的 SPISTE 反相，可接收左右声道数字音频。仅适用于从模式；主模式（MASTER_SLAVE = 1）下写此位无效。<br>0h (R/W) = SPISTEn 低有效，正常。<br>1h (R/W) = SPISTE 高有效，反相。<br>复位类型：SYSRSn |
| 0 | TRIWIRE | R/W | 0h | SPI 三线模式使能。<br>0h (R/W) = 正常四线模式。<br>1h (R/W) = 启用三线模式，未使用引脚成为 GPIO。主模式中 SPISIMO 成为 SPIMOMI（主机收发）引脚，SPISOMI 可用于非 SPI 功能；从模式中 SPISOMI 成为 SPISISO（从机收发）引脚，SPISIMO 可用于非 SPI 功能。<br>复位类型：SYSRSn |

### 25.5.3 SPI 寄存器与 DriverLib 函数对应表

表 25-33 SPI 寄存器与 DriverLib 函数对应表

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| SPICCR | spi.c | SPI_setConfig |
| SPICCR | spi.c | SPI_clearInterruptStatus |
| SPICCR | spi.c | SPI_pollingNonFIFOTransaction |
| SPICCR | spi.c | SPI_pollingFIFOTransaction |
| SPICCR | spi.h | SPI_enableModule |
| SPICCR | spi.h | SPI_disableModule |
| SPICCR | spi.h | SPI_setcharLength |
| SPICCR | spi.h | SPI_enableLoopback |
| SPICCR | spi.h | SPI_disableLoopback |
| SPICCR | spi.h | SPI_enableHighSpeedMode |
| SPICCR | spi.h | SPI_disableHighSpeedMode |
| SPICTL | spi.c | SPI_setConfig |
| SPICTL | spi.c | SPI_enableInterrupt |
| SPICTL | spi.c | SPI_disableInterrupt |
| SPICTL | spi.h | SPI_enableTalk |
| SPICTL | spi.h | SPI_disableTalk |
| SPISTS | spi.c | SPI_getInterruptStatus |
| SPISTS | spi.c | SPI_clearInterruptStatus |
| SPISTS | spi.h | SPI_writeDataBlockingNonFIFO |
| SPISTS | spi.h | SPI_readDataBlockingNonFIFO |
| SPIBRR | spi.c | SPI_setConfig |
| SPIBRR | spi.c | SPI_setBaudRate |
| SPIRXEMU | spi.h | SPI_readRxEmulationBuffer |
| SPIRXBUF | spi.h | SPI_readDataNonBlocking |
| SPIRXBUF | spi.h | SPI_readDataBlockingFIFO |
| SPIRXBUF | spi.h | SPI_readDataBlockingNonFIFO |
| SPITXBUF | spi.h | SPI_writeDataNonBlocking |
| SPITXBUF | spi.h | SPI_writeDataBlockingFIFO |
| SPITXBUF | spi.h | SPI_writeDataBlockingNonFIFO |
| SPIDAT | - | - |
| SPIFFTX | spi.c | SPI_enableInterrupt |
| SPIFFTX | spi.c | SPI_disableInterrupt |
| SPIFFTX | spi.c | SPI_getInterruptStatus |
| SPIFFTX | spi.c | SPI_clearInterruptStatus |

| 寄存器 | 文件 | DriverLib 函数 |
|---|---|---|
| SPIFFTX | spi.h | SPI_enableFIFO |
| SPIFFTX | spi.h | SPI_disableFIFO |
| SPIFFTX | spi.h | SPI_resetTxFIFO |
| SPIFFTX | spi.h | SPI_setFIFOInterruptLevel |
| SPIFFTX | spi.h | SPI_getFIFOInterruptLevel |
| SPIFFTX | spi.h | SPI_getTxFIFOStatus |
| SPIFFTX | spi.h | SPI_isBusy |
| SPIFFTX | spi.h | SPI_reset |
| SPIFFRX | spi.c | SPI_enableInterrupt |
| SPIFFRX | spi.c | SPI_disableInterrupt |
| SPIFFRX | spi.c | SPI_getInterruptStatus |
| SPIFFRX | spi.c | SPI_clearInterruptStatus |
| SPIFFRX | spi.h | SPI_enableFIFO |
| SPIFFRX | spi.h | SPI_disableFIFO |
| SPIFFRX | spi.h | SPI_resetRxFIFO |
| SPIFFRX | spi.h | SPI_setFIFOInterruptLevel |
| SPIFFRX | spi.h | SPI_getFIFOInterruptLevel |
| SPIFFRX | spi.h | SPI_getRxFIFOStatus |
| SPIFFCT | spi.h | SPI_setTxFifoTransmitDelay |
| SPIPRI | spi.h | SPI_enableTriWire |
| SPIPRI | spi.h | SPI_disableTriWire |
| SPIPRI | spi.h | SPI_setPTESignalPolarity |
| SPIPRI | spi.h | SPI_setEmulationMode |

# 26 嵌入式实时分析和诊断寄存器（ERAD）

## 26.1 ERAD 特性

嵌入式实时分析和诊断寄存器（Embedded Real-time Analysis and Diagnostic，ERAD）增强了 CPU 外部设备的调试和系统分析功能，对于许多实时系统，并不总是可以连接调试器并执行侵入式调试。在这些情况下，用户代码可以设置和控制 ERAD 模块来调试和分析系统，而不会干扰最终应用程序。

ERAD 模块包含八个 EBC 单元，这些单元可以监控系统总线（例如地址总线和数据总线），并在符合设定条件时生成输出事件，同时生成实时中断或不可屏蔽中断。

ERAD 模块还包含四个 SEC 单元，这些单元可以与 EBC 单元结合使用，用于更全面的系统剖析和性能分析。这些计数器可以用于统计系统中某些事件的发生频率或持续时间。

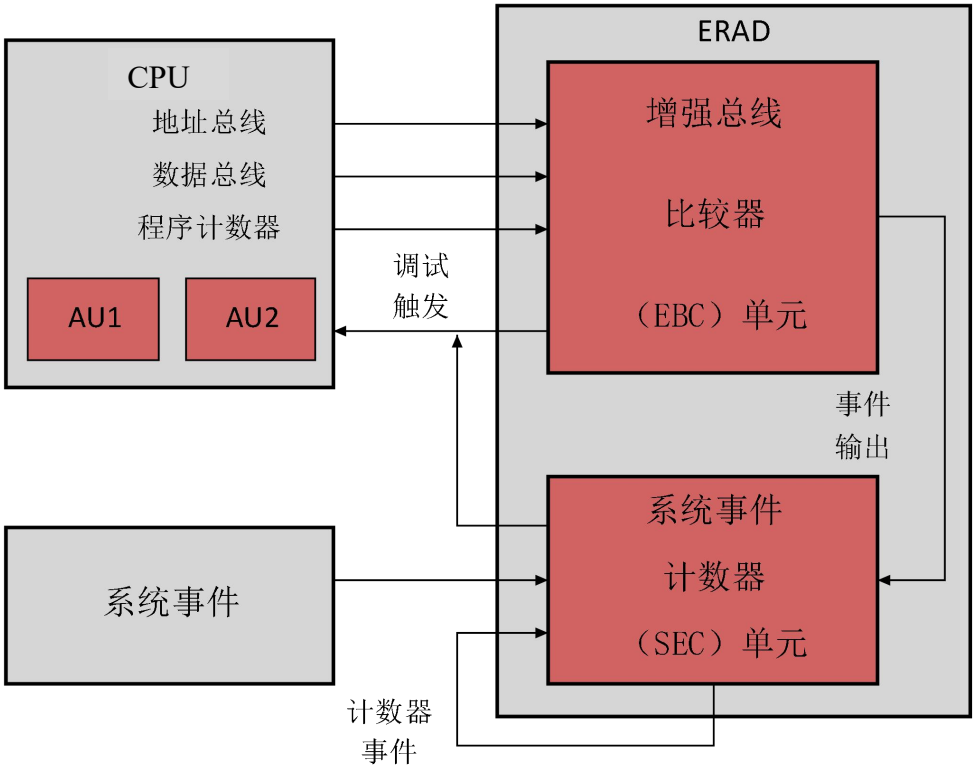



图 26-1 ERAD 概述

## 26.2 EBC 单元（Enhanced Bus Comparator）

增强型总线比较器(EBC)单元使用直接存储器接口连接 CPU。这包括程序地址总线、数据总线、数据地址，写和读取数据总线等。通常，EBC 由在 CPU 上运行的用户应用程序也可以配置和使用 EBC 单元来生成事件和中断，以进行实时诊断。

EBC 单元支持的功能：

- 为指令取指匹配生成跟踪标记，并生成实时中断；
- 监控数据地址及读写总线，并生成实时中断；
- 生成事件触发输出供其他模块使用

EBC 单元不支持的功能：

不支持链式断点；

链式断点（Chained Breakpoints）是一种调试技术，用于在硬件或软件中设置多个关联的断点，使它们之间具有特定的依赖关系。当一个断点触发时，其他与之关联的断点也会触发或启用。

不能监测 DMA、BGCRC、HIC 传输；

不能监测 CLA 总线

EBC 可以通过地址掩码来监测一个地址范围，并根据大于、小于或等于事件生成输出。

EBC 可以和 SEC 搭配使用，SEC 对系统活动或代码执行进行计数和分析。

### 26.2.1 EBC 单元操作

EBC 单元支持以下操作：

程序跟踪 (Program Trace)

功能：程序跟踪与硬件断点类似，但不直接暂停 CPU。

执行过程：当指令到达 D2 阶段时，会生成一个实时中断（RTOSINT）。如果由于不连续性而在取指缓冲区中丢弃了指令，则不会生成 RTOSINT。这对于监测程序执行而不影响其正常运行非常有用。

数据跟踪 (Data Trace)

功能：数据跟踪与监视点类似，但其处理方式略有不同。

执行过程：当访问指定的数据存储位置时，数据跟踪会生成一个实时中断（RTOSINT），而不是暂停 CPU。这允许程序继续执行，同时在需要时进行监测。

表 26-1 EBC 监测总线选择

| 总线 | 说明 |
| --- | --- |
| PAD | 指令总线的地址线 |
| PC-ID | 程序计数值 |
| DWAB | 数据写地址总线 |
| DRAB | 数据读地址总线 |
| DWDB | 写数据总线 |
| DRDB | 读数据总线 |
| PC Aligned Match | PC对齐阶段匹配 |
| PC R | 读指令PC到lsu阶段匹配 |
| PC W | 写指令PC到lsu阶段匹配 |

### 26.2.2 事件屏蔽和导出

不同 EBC 单元生成的事件可通过 OR 和 AND 逻辑组合，生成所需的新事件。有四种 AND 和四种 OR 组合可以使用掩码导出，以抑制不希望发生的事件。

这些事件可以配置为生成 RTOSINT。此外，AND 和 OR 事件也可用作系统事件计数器单元的输入，用于事件计数和系统分析。

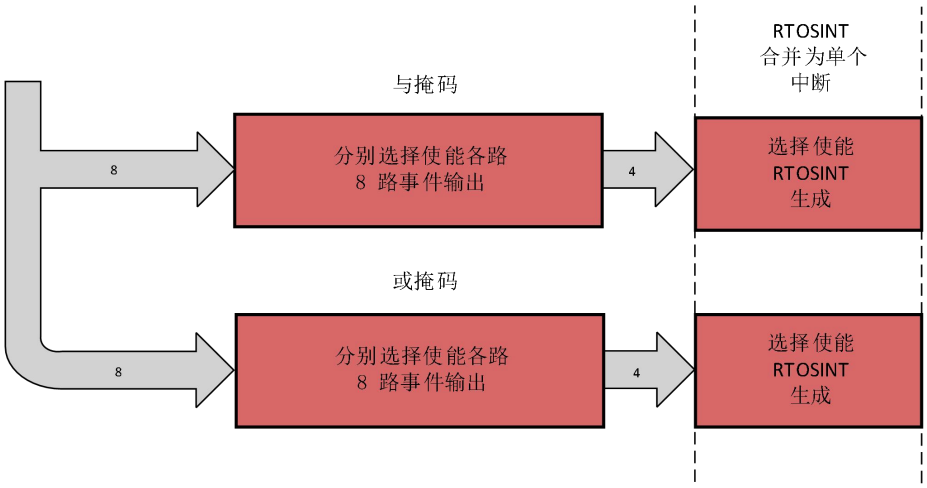



图 26-2 EBC 单元事件屏蔽

使用 AND 和 OR 掩码步骤如下：

1.配置 GLBL_EVENT_AND_MASK 和 GLBL_EVENT_OR_MASK 寄存器来为四个可用掩码中的任何一个选择所需的 EBC 单元输出。

2. 要为配置的掩码启用 RTOSINT，请将 1 写入 GLBL_AND_EVENT_INT_MASK 或 GLBL_OR_EVENT_INT_MASK 寄存器中的相应位。

3.要将掩码输出用作系统事件计数器单元的输入，请将 INPUT_SEL 寄存器配置为所需掩码的复用器值。

## 26.3 SEC 单元（System Event Counter）

SEC 单元提供系统性能分析和调试功能。SEC 单元包含计数器，可增强各种系统场景中的调试和分析过程 ，例如：

- 分析代码段
- 计算指定内存读写之间的持续时间
- 统计系统事件（例如中断）
- 计算系统事件之间的持续时间
- 系统定时器
- 测量代码段中等待状态的数量

- 测量多次迭代中一对事件之间的最大时间间隔
- 链接计数器以链接事件或创建更大的计数器

此外，SEC 单元还具有以下功能：

1、作为一个计数器，对以下事件计数：

由 EBC 单元生成的任何匹配事件。

由 EBC 单元生成的事件。这些事件可以用来启动和停止计数。

系统事件，外设中断，计时器中断等。这些系统事件可以用于启动和停止计数的输入事件。

2、如果计数器操作达到参考值，则生成一个 RTOSINT 或 NMI 中断

3、计数操作有两种模式：

持续事件模式：只要事件处于活动状态，计数器就会计算事件持续时钟周期；

事件模式：事件发生的次数

### 26.3.1 系统事件计数器模式

以下是 SEC 单元的运行模式。当 SEC 模块收到复位输入信号时，计数器初始化为零，并始终向上计数。

连续计数：用于计算 CPU 周期，当 SEC_CNTL.CNT_INP_SEL_EN 设置为 0，SEC_CNTL.START_STOP_MODE 均设置为 0 时，连续模式处于活动状态，此时自由计数。

事件计数：事件触发一次，计数一次，计数器将达到在 SEC_REF 寄存器中定义的设定参考值。在达到参考值时，计数器会生成一个事件，该事件可以向 CPU 发送一个实时中断或一个不可屏蔽中断。

启动-停止计数：设置两个事件：Start、Stop。Start 事件开始之后开始计数，直到 Stop 事件发生，一旦发生了第一个 start，就会忽略其他的 start，直到停止事件发生为止。

在任何计数器操作模式下，32 位计数器值都有可能溢出。如果发生溢出，计数器值将重置为零并继续计数，并且 SEC_STATUS 寄存器中的 OVERFLOW 位将设置为高电平。OVERFLOW 位保持高电平，直到计数器重置，或者应用程序将 1 写入 SEC_CLEAR 寄存器的 OVERFLOW_CLEAR 位。

#### 26.3.1.1 有效电平和边沿计数

SEC 单元可配置为计数选定输入的有效电平或边沿。

每个 SEC 单元都有来自 EBC 单元的八个输入和来自设备中其他事件的许多输入。每个 SEC 单元都可以配置为对任何输入事件进行计数或仅在每个周期进行计数。例如，如果输入事件发生并且处于活动状态 25 个周期，则 SEC 单元计数器在事件模式下仅递增 1；而在持续时间模式下，计数器递增 25。

#### 26.3.1.2 最大模式

每次发生停止事件，且计数器停止时，都将针对寄存器中存在的当前 MAX_COUNT 检查计数器值。如果新值更大，则将更新 MAX_COUNT 寄存器。计数器总是在停止事件时重置为零。

#### 26.3.1.3 连续模式

计数器不会因为停止事件而重置。当一个开始事件再次发生时，它将停止计数并恢复计数。在累积计数模式下，MAX_COUNT 无效。

#### 26.3.1.4 输入信号选择

可以从系统中的各种信号中选择 SEC 输入，以启用调试和系统分析。表中显示了 SEC 输入。

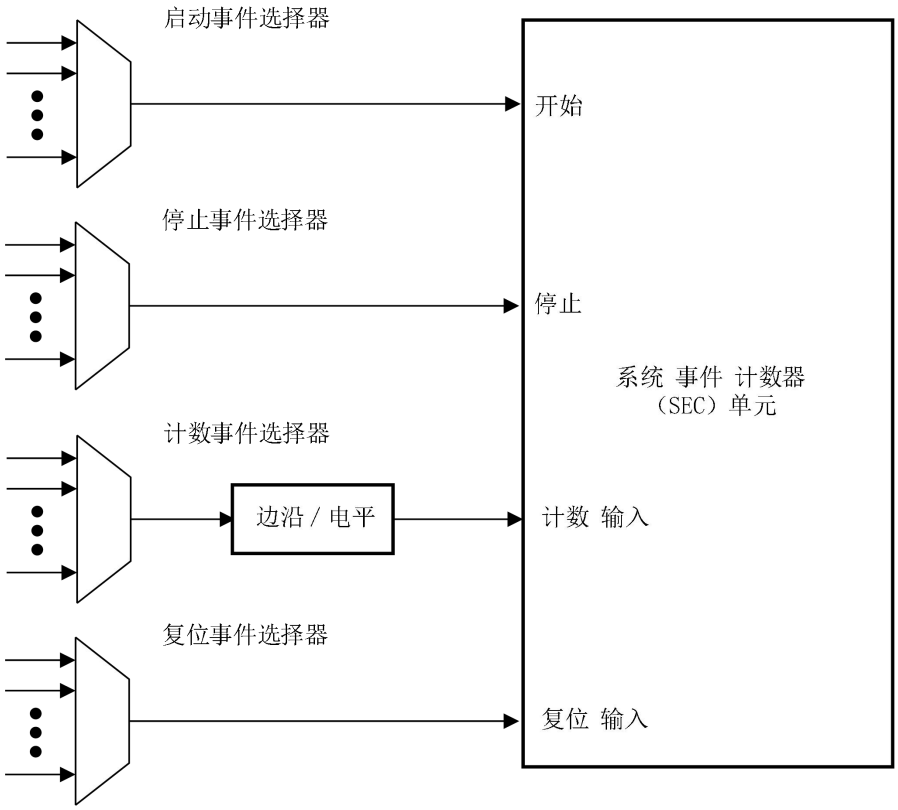



图 26-3 SEC 单元事件计数

表 26-2 事件选择器多路复用器信号

| SEC\STA\STO \RST_INP_SEL | 事件输入选择（EVENT） | 触发边沿 | 同步需求 |
|---|---|---|---|
| 0 | EBC1 | 高 | 无 |
| 1 | EBC2 | 高 | 无 |
| 2 | EBC3 | 高 | 无 |
| 3 | EBC4 | 高 | 无 |
| 4 | EBC5 | 高 | 无 |
| 5 | EBC6 | 高 | 无 |
| 6 | EBC7 | 高 | 无 |
| 7 | EBC8 | 高 | 无 |
| 8 | COUNTER1_EVENT | 高 | 无 |
| 9 | COUNTER2_EVENT | 高 | 无 |
| 10 | COUNTER3_EVENT | 高 | 无 |
| 11 | COUNTER4_EVENT | 高 | 无 |
| 12 | ERAD_OR_MASK0 | 高 | 无 |
| 13 | ERAD_OR_MASK1 | 高 | 无 |
| 14 | ERAD_OR_MASK2 | 高 | 无 |
| 15 | ERAD_OR_MASK3 | 高 | 无 |
| 16 | ERAD_AND_MASK0 | 高 | 无 |
| 17 | ERAD_AND_MASK1 | 高 | 无 |
| 18 | ERAD_AND_MASK2 | 高 | 无 |

| SEC\STA\STO \RST_INP_SEL | 事件输入选择（EVENT） | 触发边沿 | 同步需求 |
|---|---|---|---|
| 19 | ERAD_AND_MASK3 | 高 | 无 |
| 20 | FSIRXA_DATA_PKT_RCVD | 高 | 无 |
| 21 | FSIRXA_ERROR_PKT_RCVD | 高 | 无 |
| 22 | FSIRXA_PING_PKT_RCVD | 高 | 无 |
| 23 | FSIRXA_PING_TAG_MATCH | 高 | 无 |
| 24 | FSIRXA_DATA_TAG_MATCH | 高 | 无 |
| 25 | FSIRXA_ERROR_TAG_MATCH | 高 | 无 |
| 26 | FSIRXA_FRAME_DONE | 高 | 无 |
| 27 | ADCSOCA | 低 | 无 |
| 28 | ADCSOCB | 低 | 无 |
| 29 | EPWMXBAR1 | 低 | 需要 |
| 30 | EPWMXBAR2 | 低 | 需要 |
| 31 | EPWMXBAR3 | 低 | 需要 |
| 32 | EPWMXBAR4 | 低 | 需要 |
| 33 | EPWMXBAR5 | 低 | 需要 |
| 34 | EPWMXBAR6 | 低 | 需要 |
| 35 | EPWMXBAR7 | 低 | 需要 |
| 36 | EPWMXBAR8 | 低 | 需要 |
| 37 | INPUTXBAR1 | 低 | 需要 |
| 38 | INPUTXBAR2 | 低 | 需要 |
| 39 | INPUTXBAR3 | 低 | 需要 |
| 40 | INPUTXBAR4 | 低 | 需要 |
| 41 | INPUTXBAR5 | 低 | 需要 |
| 42 | INPUTXBAR6 | 低 | 需要 |
| 43 | INPUTXBAR7 | 低 | 需要 |
| 44 | INPUTXBAR8 | 低 | 需要 |
| 45 | INPUTXBAR9 | 低 | 需要 |
| 46 | INPUTXBAR10 | 低 | 需要 |
| 47 | INPUTXBAR11 | 低 | 需要 |
| 48 | INPUTXBAR12 | 低 | 需要 |
| 49 | INPUTXBAR13 | 低 | 需要 |
| 50 | INPUTXBAR14 | 低 | 需要 |
| 51 | INPUTXBAR15 | 低 | 需要 |
| 52 | INPUTXBAR16 | 低 | 需要 |
| 53 | CMPSS1_CTRIPH_OR_CTRIPL | 高 | 需要 |
| 54 | CMPSS2_CTRIPH_OR_CTRIPL | 高 | 需要 |
| 55 | CMPSS3_CTRIPH_OR_CTRIPL | 高 | 需要 |
| 56 | CMPSS4_CTRIPH_OR_CTRIPL | 高 | 需要 |
| 57 | HIC_nOE | 低 | 需要 |
| 58 | HIC_nWE | 低 | 需要 |
| 59 | HIC_nRDY | 低 | 无 |
| 60 | CLIC software interrupt | 高 | 无 |
| 61 | TIMER2 | 高 | 无 |

| SEC\STA\STO \RST_INP_SEL | 事件输入选择（EVENT） | 触发边沿 | 同步需求 |
|---|---|---|---|
| 62 | TIMER1 | 高 | 无 |
| 63 | RAM access violation interrupt | 高 | 无 |
| 64 | Flash correctable error | 高 | 无 |
| 65 | RAM correctable errcc interrupt | 高 | 无 |
| 66 | FPU underflow interrupt | 高 | 无 |
| 67 | FPU overflow interrupt | 高 | 无 |
| 68 | Flash Wrapper Operation done interrupt | 高 | 无 |
| 69 | XINT5 interrupt | 高 | 无 |
| 70 | XINT4 interrupt | 高 | 无 |
| 71 | XINT3 interrupt | 高 | 无 |
| 72 | ADCC4 interrupt | 高 | 无 |
| 73 | ADCC3 interrupt | 高 | 无 |
| 74 | ADCC2 interrupt | 高 | 无 |
| 75 | ADCC event interrupt | 高 | 无 |
| 76 | ADCA4 interrupt | 高 | 无 |
| 77 | ADCA3 interrupt | 高 | 无 |
| 78 | ADCA2 interrupt | 高 | 无 |
| 79 | ADCA event interrupt | 高 | 无 |
| 80 | HICA Interrupt | 高 | 无 |
| 81 | BGCRC Interrupt | 高 | 无 |
| 82 | FDCAN interrupt | 高 | 无 |
| 83 | CANA interrupt 1 | 高 | 无 |
| 84 | CANA interrupt 0 | 高 | 无 |
| 85 | SCIA TX interrupt | 高 | 无 |
| 86 | SCIA RX interrupt | 高 | 无 |
| 87 | DCC1 interrupt | 高 | 无 |
| 88 | PMBUSA interrupt | 高 | 无 |
| 89 | LINB interrupt 1 | 高 | 无 |
| 90 | LINB interrupt 0 | 高 | 无 |
| 91 | LINA interrupt 1 | 高 | 无 |
| 92 | LINA interrupt 0 | 高 | 无 |
| 93 | I2CB FIFO interrupt | 高 | 无 |
| 94 | I2CB interrupt | 高 | 无 |
| 95 | I2CA FIFO interrupt | 高 | 无 |
| 96 | I2CA interrupt | 高 | 无 |
| 97 | DCC0 interrupt | 高 | 无 |
| 98 | FSITX INT4 | 高 | 无 |
| 99 | FSITX INT3 | 高 | 无 |
| 100 | FSITX INT2 | 高 | 无 |
| 101 | FSITX INT1 | 高 | 无 |
| 102 | DMA CH6 interrupt | 高 | 无 |
| 103 | DMA CH5 interrupt | 高 | 无 |
| 104 | DMA CH4 interrupt | 高 | 无 |

| SEC\STA\STO \RST_INP_SEL | 事件输入选择（EVENT） | 触发边沿 | 同步需求 |
|---|---|---|---|
| 105 | DMA CH3 interrupt | 高 | 无 |
| 106 | DMA CH2 interrupt | 高 | 无 |
| 107 | DMA CH1 interrupt | 高 | 无 |
| 108 | SPIB TX interrupt | 高 | 无 |
| 109 | SPIB RX interrupt | 高 | 无 |
| 110 | SPIA TX interrupt | 高 | 无 |
| 111 | SPIA RX interrupt | 高 | 无 |
| 112 | CLB2 interrupt | 高 | 无 |
| 113 | CLB1 interrupt | 高 | 无 |
| 114 | EQEP2 interrupt | 高 | 无 |
| 115 | EQEP1 interrupt | 高 | 无 |
| 116 | ECAP3 inermupt 2 | 高 | 无 |
| 117 | ECAP3 interrupt | 高 | 无 |
| 118 | ECAP2 interrupt | 高 | 无 |
| 119 | ECAP1 interrupt | 高 | 无 |
| 120 | EPWM7 interrupt | 高 | 无 |
| 121 | EPWM6 interrupt | 高 | 无 |
| 122 | EPWM5 interrupt | 高 | 无 |
| 123 | EPWM4 interrupt | 高 | 无 |
| 124 | EPWM3 interrupt | 高 | 无 |
| 125 | EPWM2 interrupt | 高 | 无 |
| 126 | EPWM1 interrupt | 高 | 无 |
| 127 | EPWM7 trip zone interrupt | 高 | 无 |
| 128 | EPWM6 trip zone interrupt | 高 | 无 |
| 129 | EPWM5 trip zone interrupt | 高 | 无 |
| 130 | EPWM4 trip zone interrupt | 高 | 无 |
| 131 | EPWM3 trip zone interrupt | 高 | 无 |
| 132 | EPWM2 trip zone interrupt | 高 | 无 |
| 133 | EPWM1 trip zone interrupt | 高 | 无 |
| 134 | ERAD | 高 | 无 |
| 135 | WAKE | 高 | 无 |
| 136 | TIMER0 | 高 | 无 |
| 137 | XINT2 | 高 | 无 |
| 138 | XINT1 | 高 | 无 |
| 139 | ADCC1 | 高 | 无 |
| 140 | ADCA1 | 高 | 无 |
| 141 | DEBUG_INT | 高 | 无 |
| 142 | RNMI | 高 | 无 |
| 143 | UNMI | 高 | 无 |

### 26.3.2 根据事件复位

外部事件可以重置计数器。此外，所有计数器事件输出都应用回每个计数器输入 MUX，选择可用作重置输入的事件。启用后，重置输入上的高电平有效会导致计数器重置。这提供了一个强大的功能，允许设置阈值监视器。如果两个事件之间的距离超过某个阈值，这可用于标记中断。

### 26.3.3 运行条件

SEC 单元在 CPU 正常运行的情况下能够准确计数 CPU 周期。这意味着在 CPU 不受调试器干预的情况下，计数器能提供精确的周期测量。

调试器控制：当 CPU 在调试模式下被单步执行（逐条指令）时，计数器的准确性可能会受到影响。调试操作（如单步执行、断点、运行时修改等）会中断或改变 CPU 的正常执行流程，这可能导致计数器记录的周期数与 CPU 正常执行时的周期数不同。

中断和断点：调试过程中设置的中断和断点可能会暂停 CPU 执行或改变其行为，从而影响计数器的测量。

单步执行：逐条指令执行会导致额外的控制操作，这些操作可能会干扰计数器的正常计数。

## 26.4 ERAD 编程顺序

ERAD 模块可用于设置硬件中断。本节介绍了设置硬件中断或使用定时器对系统进行分析的编程序列。

### 26.4.1 设置硬件中断编程顺序

总的逻辑：

1、确保设备处于 IDLE 模式；

2、设置地址参考、掩码、总线选择和停止位；

3、全局启用寄存器中启用相应的模块位；

4、使用完成后，写 1 以清除 EVENT_FILED，使模块返回启用状态。

举例：硬件中断的示例编程序列如下。

在地址 0x201000 上设置监测点。

读取 EBC_STATUS 以确认模块处于 IDLE 状态；

将 0x0 写入 EBC_MASK；

将 0x201000 写入 EBC_REF；

设置 EBC_CNTL.RTOSINT = 1 和 EBC_CNTL.BUS_SEL = 0000 (PAD)。

启用全局使能寄存器中的相应模块位。

举例：在读取从 0x121010 到 0x12101F 的地址时设置中断。

读取 EBC_STATUS 以确认模块处于 IDLE 状态。

向 EBC_MASK 写入 0xF。

向 EBC_REF 写入 0x121010。

设置 EBC_CNTL. RTOSINT = 1 和 EBC_CNTL.BUS_SEL = 0000 (PAD)。

启用全局使能寄存器中的相应模块位。

举例：在写入地址 0xFF10101A 时设置硬件中断

读取 EBC_STATUS 以确认模块处于 IDLE 状态。

将 0x0 写入 EBC_MASK。

将 0xFF10101A 写入 EBC_REF。

设置 EBC_CNTL. RTOSINT = 1 和 EBC_CNTL.BUS_SEL = 0000 (PAD)。

启用全局使能寄存器中的相应模块位。

### 26.4.2 定时器和计数器编程顺序

使用 SEC 设备时，无论调试软件还是应用程序对设备进行编程，编程顺序都是相同的。典型的单元编程顺序如下。

确保设备处于 IDLE 模式。

设置计数器基准、计数器寄存器（如果需要清零启动，则清零/复位）和 SEC_CNTL。

使能全局使能寄存器中相应的模块位。

使用完成后，写入 1 以清除 EVENT_FIRED 粘性位。这将使模块回到启用状态。

设置一个自由运行的计数器：

读取 SEC_STATUS，确认模块处于 IDLE 状态。

将 0x0 写入 SEC_INPUT_SEL。

向 SEC_CNTL 写入 0x0000。

在 GLBL_ENABLE 寄存器中启用模块。

设置计数器，计算地址 0x1000 和 0x1210 之间的持续时间：

读取并确保所有权设置符合预期。如果没有，请根据需要在继续操作前获取所有权。

读取 SEC_STATUS，确认模块处于 IDLE 状态。

设置 EBC 单元 1，以生成 PAD = 0x1000 的事件。

设置 EBC 单元 2 以生成 PAD= 0x1210 的事件。

设置开始和停止输入选择，分别使用比较器事件 1 和 2。这可通过写入 SEC_INPUT_SEL.STA_INP_SEL=0x0 和 SEC_INPUT_SEL_2.STO_INP_SEL = 0x1 来实现。

在 START_STOP_MODE 运行模式下启用计数器。这可通过写入 SEC_CNTL. START_STOP_MODE = 1。

在 GLBL_ENABLE 寄存器中启用模块。

设置计数器，计算地址为 0x2010 的函数被调用的次数，如果次数达到 0x300，则触发 RTOSINT：

读取并确保所有权设置符合预期。如果没有，请根据需要在继续操作前获取所有权。

读取 SEC_STATUS，确认模块处于 IDLE 状态。

设置 EBC 单元 1，生成 PAD = 0x2010 的事件。

设置计数器，选择比较器事件 1 作为计数事件。这可通过写入 SEC_INPUT_SEL.CNT_INP_SEL= 0 和 SEC_CNTL.CNT_INP_SEL_EN = 1 来实现。

写入 0x300 SEC_REF。

在 EVENT_MODE 中启用计数器，并允许计数器在计数与参考值匹配时生成 RTOSINT。这可以通过向 SEC_CNTL（位 3 = 1 和位 7 = 1）写入 0x88 来实现。

在 GLBL_ENABLE 寄存器中启用模块。

本节描述嵌入式实时分析和诊断寄存器。

## 26.5 寄存器描述

本节描述嵌入式实时分析和诊断寄存器。

### 26.5.1 ERAD 基地址表

表 26-3 ERAD 基地址表

| 位字段实例 | 位字段结构 | DriverLib 名称 | 基地址 | DMA | HIC | 流水线保护 |
|---|---|---|---|---|---|---|
| EradGlobalRegs | ERAD_GLOBAL_REGS | ERAD_GLOBAL_BASE | 0x0000_7000 | - | - | 是 |
| EradHWBP1Regs | ERAD_HWBP_REGS | ERAD_HWBP1_BASE | 0x0000_7200 | - | - | 是 |
| EradHWBP2Regs | ERAD_HWBP_REGS | ERAD_HWBP2_BASE | 0x0000_7210 | - | - | 是 |
| EradHWBP3Regs | ERAD_HWBP_REGS | ERAD_HWBP3_BASE | 0x0000_7220 | - | - | 是 |
| EradHWBP4Regs | ERAD_HWBP_REGS | ERAD_HWBP4_BASE | 0x0000_7230 | - | - | 是 |
| EradHWBP5Regs | ERAD_HWBP_REGS | ERAD_HWBP5_BASE | 0x0000_7240 | - | - | 是 |
| EradHWBP6Regs | ERAD_HWBP_REGS | ERAD_HWBP6_BASE | 0x0000_7250 | - | - | 是 |
| EradHWBP7Regs | ERAD_HWBP_REGS | ERAD_HWBP7_BASE | 0x0000_7260 | - | - | 是 |
| EradHWBP8Regs | ERAD_HWBP_REGS | ERAD_HWBP8_BASE | 0x0000_7270 | - | - | 是 |
| EradCounter1Regs | ERAD_COUNTER_REGS | ERAD_COUNTER1_BASE | 0x0000_7300 | - | - | 是 |
| EradCounter2Regs | ERAD_COUNTER_REGS | ERAD_COUNTER2_BASE | 0x0000_7320 | - | - | 是 |
| EradCounter3Regs | ERAD_COUNTER_REGS | ERAD_COUNTER3_BASE | 0x0000_7340 | - | - | 是 |
| EradCounter4Regs | ERAD_COUNTER_REGS | ERAD_COUNTER4_BASE | 0x0000_7360 | - | - | 是 |

## 26.5.2 ERAD_GLOBAL_REGS 寄存器

表中列出了 ERAD_GLOBAL_REGS 的存储器映射寄存器。表中未列出的寄存器偏移地址均应视为保留位置，不应修改这些位置的寄存器内容。

表 26-4 ERAD_GLOBAL_REGS 寄存器

| 偏移地址 | 缩写 | 寄存器名称 | 写保护 |
|---|---|---|---|
| 0h | GLBL_EVENT_STAT | 全局事件状态寄存器 | |
| 4h | GLBL_HALT_STAT | 全局停止状态寄存器 | |
| 8h | GLBL_ENABLE | 全局使能寄存器 | |
| Ch | GLBL_CTM_RESET | 全局计数器复位 | |
| 10h | GLBL_NMI_CTL | 全局调试 NMI 控制 | |
| 18h | GLBL_EVENT_AND_MASK | 全局总线比较器事件 AND 掩码寄存器 | |
| 1Ch | GLBL_EVENT_OR_MASK | 全局总线比较器事件 OR 掩码寄存器 | |
| 20h | GLBL_AND_EVENT_INT_MASK | 全局 AND 事件中断掩码寄存器 | |
| 24h | GLBL_OR_EVENT_INT_MASK | 全局 OR 事件中断掩码寄存器 | |

复杂的位访问类型采用编码形式，以便放入较小的表格单元格。下表列出了本节使用的访问类型编码。

表 26-5 ERAD_GLOBAL_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |

### 26.5.2.1 GLBL_EVENT_STAT 寄存器

偏移地址：0h　复位值：0000h

全局事件状态寄存器。

表 26-6 GLBL_EVENT_STAT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | CTM4 | CTM3 | CTM2 | CTM1 |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HWBP8 | HWBP7 | HWBP6 | HWBP5 | HWBP4 | HWBP3 | HWBP2 | HWBP1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 26-7 GLBL_EVENT_STAT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11 | CTM4 | R | 0h | 此位直接反映计数器单元 4的 EVENT_FIRED 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 10 | CTM3 | R | 0h | 此位直接反映计数器单元 3的 EVENT_FIRED 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 9 | CTM2 | R | 0h | 此位直接反映计数器单元 2的 EVENT_FIRED 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 8 | CTM1 | R | 0h | 此位直接反映计数器单元 1的 EVENT_FIRED 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 7 | HWBP8 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 8 的 EVENT_FIRED 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 6 | HWBP7 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 7 的 EVENT_FIRED 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 5 | HWBP6 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 6 的 EVENT_FIRED 位状态。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 4 | HWBP5 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 5 的 EVENT_FIRED 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 3 | HWBP4 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 4 的 EVENT_FIRED 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 2 | HWBP3 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 3 的 EVENT_FIRED 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 1 | HWBP2 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 2 的 EVENT_FIRED 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 0 | HWBP1 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 1 的 EVENT_FIRED 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |

### 26.5.2.2 GLBL_HALT_STAT 寄存器

偏移地址：4h　复位值：0000h

全局停止状态寄存器。

表 26-8 GLBL_HALT_STAT 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | CTM4 | CTM3 | CTM2 | CTM1 |
| R-0h | | | | R-0h | R-0h | R-0h | R-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HWBP8 | HWBP7 | HWBP6 | HWBP5 | HWBP4 | HWBP3 | HWBP2 | HWBP1 |
| R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h | R-0h |

表 26-9 GLBL_HALT_STAT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11 | CTM4 | R | 0h | 此位直接反映计数器单元 4的 completed 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 10 | CTM3 | R | 0h | 此位直接反映计数器单元 3的 completed 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 9 | CTM2 | R | 0h | 此位直接反映计数器单元 2的 completed 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 8 | CTM1 | R | 0h | 此位直接反映计数器单元 1的 completed 位状态。<br>0：无事件。<br>1：事件已触发。<br>复位类型：ERAD_RESET |
| 7 | HWBP8 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 8的 completed 位状态。<br>0：未完成。<br>1：已完成。<br>复位类型：ERAD_RESET |
| 6 | HWBP7 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 7的 completed 位状态。<br>0：未完成。<br>1：已完成。<br>复位类型：ERAD_RESET |
| 5 | HWBP6 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 6的 completed 位状态。<br>0：未完成。<br>1：已完成。<br>复位类型：ERAD_RESET |
| 4 | HWBP5 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 5的 completed 位状态。<br>0：未完成。<br>1：已完成。<br>复位类型：ERAD_RESET |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 3 | HWBP4 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 4 的 completed 位状态。<br>0：未完成。<br>1：已完成。<br>复位类型：ERAD_RESET |
| 2 | HWBP3 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 3 的 completed 位状态。<br>0：未完成。<br>1：已完成。<br>复位类型：ERAD_RESET |
| 1 | HWBP2 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 2 的 completed 位状态。<br>0：未完成。<br>1：已完成。<br>复位类型：ERAD_RESET |
| 0 | HWBP1 | R | 0h | 此位直接反映增强型总线比较器（EBC）单元 1 的 completed 位状态。<br>0：未完成。<br>1：已完成。<br>复位类型：ERAD_RESET |

### 26.5.2.3 GLBL_ENABLE 寄存器

偏移地址：8h　复位值：0000h

全局使能寄存器。

表 26-10 GLBL_ENABLE 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | CTM4 | CTM3 | CTM2 | CTM1 |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HWBP8 | HWBP7 | HWBP6 | HWBP5 | HWBP4 | HWBP3 | HWBP2 | HWBP1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 26-11 GLBL_ENABLE 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11 | CTM4 | R/W | 0h | 此位直接反映计数器单元 4的 ENABLE 位状态。<br>0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 10 | CTM3 | R/W | 0h | 此位直接反映计数器单元 3的 ENABLE 位状态。<br>0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |
| 9 | CTM2 | R/W | 0h | 此位直接反映计数器单元 2的 ENABLE 位状态。<br>0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |
| 8 | CTM1 | R/W | 0h | 此位直接反映计数器单元 1的 ENABLE 位状态。<br>0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |
| 7 | HWBP8 | R/W | 0h | 此位直接反映增强型总线比较器（EBC）单元 8的 ENABLE 位状态。<br>0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |
| 6 | HWBP7 | R/W | 0h | 此位直接反映增强型总线比较器（EBC）单元 7的 ENABLE 位状态。<br>0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |
| 5 | HWBP6 | R/W | 0h | 此位直接反映增强型总线比较器（EBC）单元 6的 ENABLE 位状态。<br>0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |
| 4 | HWBP5 | R/W | 0h | 此位直接反映增强型总线比较器（EBC）单元 5的 ENABLE 位状态。<br>0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |
| 3 | HWBP4 | R/W | 0h | 此位直接反映增强型总线比较器（EBC）单元 4的 ENABLE 位状态。<br>0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |
| 2 | HWBP3 | R/W | 0h | 此位直接反映增强型总线比较器（EBC）单元 3的 ENABLE 位状态。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |
| 1 | HWBP2 | R/W | 0h | 此位直接反映增强型总线比较器（EBC）单元 2 的 ENABLE 位状态。<br>0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |
| 0 | HWBP1 | R/W | 0h | 此位直接反映增强型总线比较器（EBC）单元 1 的 ENABLE 位状态。<br>0：禁用。<br>1：启用。<br>复位类型：ERAD_RESET |

### 26.5.2.4 GLBL_CTM_RESET 寄存器

偏移地址：Ch　复位值：0000h

全局计数器复位。

表 26-12 GLBL_CTM_RESET 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | CTM4 | CTM3 | CTM2 | CTM1 |
| R-0h | | | | R-0/W-0h | R-0/W-0h | R-0/W-0h | R-0/W-0h |

表 26-13 GLBL_CTM_RESET 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R | 0h | 保留。 |
| 3 | CTM4 | R-0/W | 0h | 此位直接复位计数器单元 4的状态。<br>0：无影响。<br>1：复位。<br>复位类型：ERAD_RESET |
| 2 | CTM3 | R-0/W | 0h | 此位直接复位计数器单元 3的状态。<br>0：无影响。<br>1：复位。<br>复位类型：ERAD_RESET |
| 1 | CTM2 | R-0/W | 0h | 此位直接复位计数器单元 2的状态。<br>0：无影响。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 1：复位。<br>复位类型：ERAD_RESET |
| 0 | CTM1 | R-0/W | 0h | 此位直接复位计数器单元 1的状态。<br>0：无影响。<br>1：复位。<br>复位类型：ERAD_RESET |

### 26.5.2.5 GLBL_NMI_CTL 寄存器

偏移地址：10h　复位值：0000h

全局调试 NMI 控制。

表 26-14 GLBL_NMI_CTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | CTM4 | CTM3 | CTM2 | CTM1 |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HWBP8 | HWBP7 | HWBP6 | HWBP5 | HWBP4 | HWBP3 | HWBP2 | HWBP1 |
| R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 26-15 GLBL_NMI_CTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11 | CTM4 | R/W | 0h | 此位使能计数器单元 4向 CPU 产生 NMI。<br>0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |
| 10 | CTM3 | R/W | 0h | 此位使能计数器单元 3向 CPU 产生 NMI。<br>0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |
| 9 | CTM2 | R/W | 0h | 此位使能计数器单元 2向 CPU 产生 NMI。<br>0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |
| 8 | CTM1 | R/W | 0h | 此位使能计数器单元 1向 CPU 产生 NMI。<br>0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |
| 7 | HWBP8 | R/W | 0h | 此位使能增强型总线比较器（EBC）单元 8向 CPU 产生 NMI。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |
| 6 | HWBP7 | R/W | 0h | 此位使能增强型总线比较器（EBC）单元 7向 CPU 产生 NMI。<br>0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |
| 5 | HWBP6 | R/W | 0h | 此位使能增强型总线比较器（EBC）单元 6向 CPU 产生 NMI。<br>0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |
| 4 | HWBP5 | R/W | 0h | 此位使能增强型总线比较器（EBC）单元 5向 CPU 产生 NMI。<br>0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |
| 3 | HWBP4 | R/W | 0h | 此位使能增强型总线比较器（EBC）单元 4向 CPU 产生 NMI。<br>0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |
| 2 | HWBP3 | R/W | 0h | 此位使能增强型总线比较器（EBC）单元 3向 CPU 产生 NMI。<br>0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |
| 1 | HWBP2 | R/W | 0h | 此位使能增强型总线比较器（EBC）单元 2向 CPU 产生 NMI。<br>0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |
| 0 | HWBP1 | R/W | 0h | 此位使能增强型总线比较器（EBC）单元 1向 CPU 产生 NMI。<br>0：不产生 NMI。<br>1：产生 NMI。<br>复位类型：ERAD_RESET |

### 26.5.2.6 GLBL_EVENT_AND_MASK 寄存器

偏移地址：18h　复位值：FFFFFFFFh

全局总线比较器事件 AND 掩码寄存器。

表 26-16 GLBL_EVENT_AND_MASK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MASK4_HWBP8 | MASK4_HWBP7 | MASK4_HWBP6 | MASK4_HWBP5 | MASK4_HWBP4 | MASK4_HWBP3 | MASK4_HWBP2 | MASK4_HWBP1 |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MASK3_HWBP8 | MASK3_HWBP7 | MASK3_HWBP6 | MASK3_HWBP5 | MASK3_HWBP4 | MASK3_HWBP3 | MASK3_HWBP2 | MASK3_HWBP1 |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MASK2_HWBP8 | MASK2_HWBP7 | MASK2_HWBP6 | MASK2_HWBP5 | MASK2_HWBP4 | MASK2_HWBP3 | MASK2_HWBP2 | MASK2_HWBP1 |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MASK1_HWBP8 | MASK1_HWBP7 | MASK1_HWBP6 | MASK1_HWBP5 | MASK1_HWBP4 | MASK1_HWBP3 | MASK1_HWBP2 | MASK1_HWBP1 |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h |

表 26-17 GLBL_EVENT_AND_MASK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MASK4_HWBP8 | R/W | 1h | 增强型总线比较器（EBC）单元 8 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND4 输出。<br>1：在 HWBP_EVENT_AND4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 30 | MASK4_HWBP7 | R/W | 1h | 增强型总线比较器（EBC）单元 7 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND4 输出。<br>1：在 HWBP_EVENT_AND4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 29 | MASK4_HWBP6 | R/W | 1h | 增强型总线比较器（EBC）单元 6 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND4 输出。<br>1：在 HWBP_EVENT_AND4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 28 | MASK4_HWBP5 | R/W | 1h | 增强型总线比较器（EBC）单元 5 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND4 输出。<br>1：在 HWBP_EVENT_AND4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 27 | MASK4_HWBP4 | R/W | 1h | 增强型总线比较器（EBC）单元 4 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND4 输出。<br>1：在 HWBP_EVENT_AND4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 26 | MASK4_HWBP3 | R/W | 1h | 增强型总线比较器（EBC）单元 3 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND4 输出。<br>1：在 HWBP_EVENT_AND4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 25 | MASK4_HWBP2 | R/W | 1h | 增强型总线比较器（EBC）单元 2 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND4 输出。<br>1：在 HWBP_EVENT_AND4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 24 | MASK4_HWBP1 | R/W | 1h | 增强型总线比较器（EBC）单元 1 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND4 输出。<br>1：在 HWBP_EVENT_AND4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 23 | MASK3_HWBP8 | R/W | 1h | 增强型总线比较器（EBC）单元 8 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND3 输出。<br>1：在 HWBP_EVENT_AND3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 22 | MASK3_HWBP7 | R/W | 1h | 增强型总线比较器（EBC）单元 7 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND3 输出。<br>1：在 HWBP_EVENT_AND3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 21 | MASK3_HWBP6 | R/W | 1h | 增强型总线比较器（EBC）单元 6 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND3 输出。<br>1：在 HWBP_EVENT_AND3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 20 | MASK3_HWBP5 | R/W | 1h | 增强型总线比较器（EBC）单元 5 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND3 输出。<br>1：在 HWBP_EVENT_AND3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 19 | MASK3_HWBP4 | R/W | 1h | 增强型总线比较器（EBC）单元 4 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND3 输出。<br>1：在 HWBP_EVENT_AND3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 18 | MASK3_HWBP3 | R/W | 1h | 增强型总线比较器（EBC）单元 3 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND3 输出。<br>1：在 HWBP_EVENT_AND3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 17 | MASK3_HWBP2 | R/W | 1h | 增强型总线比较器（EBC）单元 2 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND3 输出。<br>1：在 HWBP_EVENT_AND3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 16 | MASK3_HWBP1 | R/W | 1h | 增强型总线比较器（EBC）单元 1 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND3 输出。<br>1：在 HWBP_EVENT_AND3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 15 | MASK2_HWBP8 | R/W | 1h | 增强型总线比较器（EBC）单元 8 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND2 输出。<br>1：在 HWBP_EVENT_AND2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 14 | MASK2_HWBP7 | R/W | 1h | 增强型总线比较器（EBC）单元 7 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND2 输出。<br>1：在 HWBP_EVENT_AND2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 13 | MASK2_HWBP6 | R/W | 1h | 增强型总线比较器（EBC）单元 6 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND2 输出。<br>1：在 HWBP_EVENT_AND2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 12 | MASK2_HWBP5 | R/W | 1h | 增强型总线比较器（EBC）单元 5 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND2 输出。<br>1：在 HWBP_EVENT_AND2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 11 | MASK2_HWBP4 | R/W | 1h | 增强型总线比较器（EBC）单元 4 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND2 输出。<br>1：在 HWBP_EVENT_AND2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 10 | MASK2_HWBP3 | R/W | 1h | 增强型总线比较器（EBC）单元 3 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND2 输出。<br>1：在 HWBP_EVENT_AND2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 9 | MASK2_HWBP2 | R/W | 1h | 增强型总线比较器（EBC）单元 2 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND2 输出。<br>1：在 HWBP_EVENT_AND2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 8 | MASK2_HWBP1 | R/W | 1h | 增强型总线比较器（EBC）单元 1 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND2 输出。<br>1：在 HWBP_EVENT_AND2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 7 | MASK1_HWBP8 | R/W | 1h | 增强型总线比较器（EBC）单元 8 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND1 输出。<br>1：在 HWBP_EVENT_AND1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 6 | MASK1_HWBP7 | R/W | 1h | 增强型总线比较器（EBC）单元 7 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND1 输出。<br>1：在 HWBP_EVENT_AND1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 5 | MASK1_HWBP6 | R/W | 1h | 增强型总线比较器（EBC）单元 6 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND1 输出。<br>1：在 HWBP_EVENT_AND1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 4 | MASK1_HWBP5 | R/W | 1h | 增强型总线比较器（EBC）单元 5 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND1 输出。<br>1：在 HWBP_EVENT_AND1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 3 | MASK1_HWBP4 | R/W | 1h | 增强型总线比较器（EBC）单元 4 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND1 输出。<br>1：在 HWBP_EVENT_AND1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 2 | MASK1_HWBP3 | R/W | 1h | 增强型总线比较器（EBC）单元 3 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND1 输出。<br>1：在 HWBP_EVENT_AND1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 1 | MASK1_HWBP2 | R/W | 1h | 增强型总线比较器（EBC）单元 2 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND1 输出。<br>1：在 HWBP_EVENT_AND1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 0 | MASK1_HWBP1 | R/W | 1h | 增强型总线比较器（EBC）单元 1 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_AND1 输出。<br>1：在 HWBP_EVENT_AND1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |

### 26.5.2.7 GLBL_EVENT_OR_MASK 寄存器

偏移地址：1Ch　复位值：FFFFFFFFh

全局总线比较器事件 OR 掩码寄存器。

表 26-18 GLBL_EVENT_OR_MASK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MASK4_HWBP8 | MASK4_HWBP7 | MASK4_HWBP6 | MASK4_HWBP5 | MASK4_HWBP4 | MASK4_HWBP3 | MASK4_HWBP2 | MASK4_HWBP1 |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MASK3_HWBP8 | MASK3_HWBP7 | MASK3_HWBP6 | MASK3_HWBP5 | MASK3_HWBP4 | MASK3_HWBP3 | MASK3_HWBP2 | MASK3_HWBP1 |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MASK2_HWBP8 | MASK2_HWBP7 | MASK2_HWBP6 | MASK2_HWBP5 | MASK2_HWBP4 | MASK2_HWBP3 | MASK2_HWBP2 | MASK2_HWBP1 |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MASK1_HWBP8 | MASK1_HWBP7 | MASK1_HWBP6 | MASK1_HWBP5 | MASK1_HWBP4 | MASK1_HWBP3 | MASK1_HWBP2 | MASK1_HWBP1 |
| R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h | R/W-1h |

表 26-19 GLBL_EVENT_OR_MASK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31 | MASK4_HWBP8 | R/W | 1h | 增强型总线比较器（EBC）单元 8 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR4 输出。<br>1：在 HWBP_EVENT_OR4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 30 | MASK4_HWBP7 | R/W | 1h | 增强型总线比较器（EBC）单元 7 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR4 输出。<br>1：在 HWBP_EVENT_OR4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 29 | MASK4_HWBP6 | R/W | 1h | 增强型总线比较器（EBC）单元 6 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR4 输出。<br>1：在 HWBP_EVENT_OR4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 28 | MASK4_HWBP5 | R/W | 1h | 增强型总线比较器（EBC）单元 5 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | HWBP_EVENT_OR4 输出。<br>1：在 HWBP_EVENT_OR4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 27 | MASK4_HWBP4 | R/W | 1h | 增强型总线比较器（EBC）单元 4 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR4 输出。<br>1：在 HWBP_EVENT_OR4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 26 | MASK4_HWBP3 | R/W | 1h | 增强型总线比较器（EBC）单元 3 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR4 输出。<br>1：在 HWBP_EVENT_OR4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 25 | MASK4_HWBP2 | R/W | 1h | 增强型总线比较器（EBC）单元 2 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR4 输出。<br>1：在 HWBP_EVENT_OR4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 24 | MASK4_HWBP1 | R/W | 1h | 增强型总线比较器（EBC）单元 1 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR4 输出。<br>1：在 HWBP_EVENT_OR4 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 23 | MASK3_HWBP8 | R/W | 1h | 增强型总线比较器（EBC）单元 8 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR3 输出。<br>1：在 HWBP_EVENT_OR3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 22 | MASK3_HWBP7 | R/W | 1h | 增强型总线比较器（EBC）单元 7 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR3 输出。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | 1：在 HWBP_EVENT_OR3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 21 | MASK3_HWBP6 | R/W | 1h | 增强型总线比较器（EBC）单元 6 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR3 输出。<br>1：在 HWBP_EVENT_OR3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 20 | MASK3_HWBP5 | R/W | 1h | 增强型总线比较器（EBC）单元 5 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR3 输出。<br>1：在 HWBP_EVENT_OR3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 19 | MASK3_HWBP4 | R/W | 1h | 增强型总线比较器（EBC）单元 4 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR3 输出。<br>1：在 HWBP_EVENT_OR3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 18 | MASK3_HWBP3 | R/W | 1h | 增强型总线比较器（EBC）单元 3 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR3 输出。<br>1：在 HWBP_EVENT_OR3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 17 | MASK3_HWBP2 | R/W | 1h | 增强型总线比较器（EBC）单元 2 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR3 输出。<br>1：在 HWBP_EVENT_OR3 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 16 | MASK3_HWBP1 | R/W | 1h | 增强型总线比较器（EBC）单元 1 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR3 输出。<br>1：在 HWBP_EVENT_OR3 输出中屏蔽相应的 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 15 | MASK2_HWBP8 | R/W | 1h | 增强型总线比较器（EBC）单元 8 的 AND 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR2 输出。<br>1：在 HWBP_EVENT_OR2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 14 | MASK2_HWBP7 | R/W | 1h | 增强型总线比较器（EBC）单元 7 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR2 输出。<br>1：在 HWBP_EVENT_OR2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 13 | MASK2_HWBP6 | R/W | 1h | 增强型总线比较器（EBC）单元 6 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR2 输出。<br>1：在 HWBP_EVENT_OR2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 12 | MASK2_HWBP5 | R/W | 1h | 增强型总线比较器（EBC）单元 5 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR2 输出。<br>1：在 HWBP_EVENT_OR2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 11 | MASK2_HWBP4 | R/W | 1h | 增强型总线比较器（EBC）单元 4 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR2 输出。<br>1：在 HWBP_EVENT_OR2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 10 | MASK2_HWBP3 | R/W | 1h | 增强型总线比较器（EBC）单元 3 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR2 输出。<br>1：在 HWBP_EVENT_OR2 输出中屏蔽相应的 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 9 | MASK2_HWBP2 | R/W | 1h | 增强型总线比较器（EBC）单元 2 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR2 输出。<br>1：在 HWBP_EVENT_OR2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 8 | MASK2_HWBP1 | R/W | 1h | 增强型总线比较器（EBC）单元 1 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR2 输出。<br>1：在 HWBP_EVENT_OR2 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 7 | MASK1_HWBP8 | R/W | 1h | 增强型总线比较器（EBC）单元 8 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR1 输出。<br>1：在 HWBP_EVENT_OR1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 6 | MASK1_HWBP7 | R/W | 1h | 增强型总线比较器（EBC）单元 7 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR1 输出。<br>1：在 HWBP_EVENT_OR1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 5 | MASK1_HWBP6 | R/W | 1h | 增强型总线比较器（EBC）单元 6 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR1 输出。<br>1：在 HWBP_EVENT_OR1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 4 | MASK1_HWBP5 | R/W | 1h | 增强型总线比较器（EBC）单元 5 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR1 输出。<br>1：在 HWBP_EVENT_OR1 输出中屏蔽相应的 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| | | | | HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 3 | MASK1_HWBP4 | R/W | 1h | 增强型总线比较器（EBC）单元 4 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR1 输出。<br>1：在 HWBP_EVENT_OR1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 2 | MASK1_HWBP3 | R/W | 1h | 增强型总线比较器（EBC）单元 3 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR1 输出。<br>1：在 HWBP_EVENT_OR1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 1 | MASK1_HWBP2 | R/W | 1h | 增强型总线比较器（EBC）单元 2 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR1 输出。<br>1：在 HWBP_EVENT_OR1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |
| 0 | MASK1_HWBP1 | R/W | 1h | 增强型总线比较器（EBC）单元 1 的 OR 事件屏蔽。<br>0：允许相应的 HWBP_EVENT 参与 HWBP_EVENT_OR1 输出。<br>1：在 HWBP_EVENT_OR1 输出中屏蔽相应的 HWBP_EVENT。<br>复位类型：ERAD_RESET |

### 26.5.2.8 GLBL_AND_EVENT_INT_MASK 寄存器

偏移地址：20h　复位值：0000h

全局 AND 事件中断掩码寄存器。

表 26-20 GLBL_AND_EVENT_INT_MASK 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | RTOSINT_MASK4 | RTOSINT_MASK3 | RTOSINT_MASK2 | RTOSINT_MASK1 |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 26-21 GLBL_AND_EVENT_INT_MASK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R | 0h | 保留。 |
| 3 | RTOSINT_MASK4 | R/W | 0h | 全局 AND 事件 MASK4 的 RTOSINT 产生屏蔽。<br>1：允许相应的 GLOBAL_EVENT_AND 产生 RTOSINT。<br>0：禁止相应的 GLOBAL_EVENT_AND 产生 RTOSINT。<br>复位类型：ERAD_RESET |
| 2 | RTOSINT_MASK3 | R/W | 0h | 全局 AND 事件 MASK3 的 RTOSINT 产生屏蔽。<br>1：允许相应的 GLOBAL_EVENT_AND 产生 RTOSINT。<br>0：禁止相应的 GLOBAL_EVENT_AND 产生 RTOSINT。<br>复位类型：ERAD_RESET |
| 1 | RTOSINT_MASK2 | R/W | 0h | 全局 AND 事件 MASK2 的 RTOSINT 产生屏蔽。<br>1：允许相应的 GLOBAL_EVENT_AND 产生 RTOSINT。<br>0：禁止相应的 GLOBAL_EVENT_AND 产生 RTOSINT。<br>复位类型：ERAD_RESET |
| 0 | RTOSINT_MASK1 | R/W | 0h | 全局 AND 事件 MASK1 的 RTOSINT 产生屏蔽。<br>1：允许相应的 GLOBAL_EVENT_AND 产生 RTOSINT。<br>0：禁止相应的 GLOBAL_EVENT_AND 产生 RTOSINT。<br>复位类型：ERAD_RESET |

### 26.5.2.9 GLBL_OR_EVENT_INT_MASK 寄存器

偏移地址：24h　复位值：0000h

全局 OR 事件中断掩码寄存器。

表 26-22 GLBL_OR_EVENT_INT_MASK 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | RTOSINT_MASK4 | RTOSINT_MASK3 | RTOSINT_MASK2 | RTOSINT_MASK1 |
| R-0h | | | | R/W-0h | R/W-0h | R/W-0h | R/W-0h |

表 26-23 GLBL_OR_EVENT_INT_MASK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-4 | RESERVED | R | 0h | 保留。 |
| 3 | RTOSINT_MASK4 | R/W | 0h | 全局 OR 事件 MASK3 的 RTOSINT 产生屏蔽。<br>1：允许相应的 GLOBAL_EVENT_OR 产生 RTOSINT。<br>0：禁止相应的 GLOBAL_EVENT_OR 产生 RTOSINT。<br>复位类型：ERAD_RESET |
| 2 | RTOSINT_MASK3 | R/W | 0h | 全局 OR 事件 MASK2 的 RTOSINT 产生屏蔽。<br>1：允许相应的 GLOBAL_EVENT_OR 产生 RTOSINT。<br>0：禁止相应的 GLOBAL_EVENT_OR 产生 RTOSINT。<br>复位类型：ERAD_RESET |
| 1 | RTOSINT_MASK2 | R/W | 0h | 全局 OR 事件 MASK2 的 RTOSINT 产生屏蔽。<br>1：允许相应的 GLOBAL_EVENT_OR 产生 RTOSINT。<br>0：禁止相应的 GLOBAL_EVENT_OR 产生 RTOSINT。<br>复位类型：ERAD_RESET |
| 0 | RTOSINT_MASK1 | R/W | 0h | 全局 OR 事件 MASK1 的 RTOSINT 产生屏蔽。<br>1：允许相应的 GLOBAL_EVENT_OR 产生 RTOSINT。<br>0：禁止相应的 GLOBAL_EVENT_OR 产生 RTOSINT。<br>复位类型：ERAD_RESET |

### 26.5.3 ERAD_HWBP_REGS 寄存器

表中列出了 ERAD_HWBP_REGS 的存储器映射寄存器。表中未列出的寄存器偏移地址均应视为保留位置，不应修改这些位置的寄存器内容。

表 26-24 ERAD_HWBP_REGS 寄存器

| 偏移地址 | 缩 写 | 寄 存 器 名 称 |
|---|---|---|
| 0h | HWBP_MASK | HWBP（EBC）掩码寄存器 |
| 4h | HWBP_REF | HWBP（EBC）参考寄存器 |
| 8h | HWBP_CLEAR | HWBP（EBC）清除寄存器 |
| Ch | HWBP_CNTL | HWBP（EBC）控制寄存器 |
| Eh | HWBP_STATUS | HWBP（EBC）状态寄存器 |

复杂的位访问类型采用编码形式，以便放入较小的表格单元格。下表列出了本节使用的访问类型编码。

表 26-25 ERAD_HWBP_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |

#### 26.5.3.1 HWBP_MASK 寄存器

偏移地址：0h　复位值：00000000h

HWBP（EBC）掩码寄存器。

表 26-26 HWBP_MASK 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MASK | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MASK | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MASK | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MASK | | | | | | | |
| R/W-0h | | | | | | | |

表 26-27 HWBP_MASK 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | MASK | R/W | 0h | 此寄存器保存用于比较的地址掩码。其内容与参考寄存器共同用于判断地址是否匹配。满足以下表达式时，匹配条件为真：<br>(address \| mask) == (ref \| mask)<br>仅当应用程序拥有此单元且 EALLOW 已置位时，CPU 才能写入此寄存器，否则写入将被忽略。仅当调试器拥有此单元时，调试器才能写入此寄存器，否则写入将被忽略。<br>复位类型：ERAD_RESET |

### 26.5.3.2 HWBP_REF 寄存器

偏移地址：4h　复位值：00000000h

HWBP（EBC）参考寄存器。

表 26-28 HWBP_REF 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REF | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REF | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| REF | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| REF | | | | | | | |
| R/W-0h | | | | | | | |

表 26-29 HWBP_REF 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | REF | R/W | 0h | 此寄存器保存用于比较的参考地址。其内容与掩码寄存器共同用于判断地址是否匹配。满足以下表达式时，匹配条件为真：<br>(address \| mask) == (ref \| mask)<br>仅当应用程序拥有此单元且 EALLOW 已置位时，CPU 才能写入此寄存器，否则写入将被忽略。仅当调试器拥有此单元时，调试器才能写入此寄存器，否则写入将被忽略。<br>复位类型：ERAD_RESET |

### 26.5.3.3 HWBP_CLEAR 寄存器

偏移地址：8h　复位值：0000h

HWBP（EBC）清除寄存器。

表 26-30 HWBP_CLEAR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | EVENT_CLR |
| R-0h | | | | | | | R-0/W-0h |

表 26-31 HWBP_CLEAR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-1 | RESERVED | R | 0h | 保留。 |
| 0 | EVENT_CLR | R-0/W | 0h | 事件清除寄存器。<br>0：不执行操作。<br>1：向此位写入 1，将清除 HWBP_STATUS 寄存器中保持置位的 EVENT_FIRED 位，并使断点模块状态机返回 IDLE 状态。<br>读取此位始终返回 0。<br>复位类型：ERAD_RESET |

### 26.5.3.4 HWBP_CNTL 寄存器

偏移地址：Ch　复位值：0000h

HWBP（EBC）控制寄存器。

表 26-32 HWBP_CNTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | 保留 | 保留 | COMP_MODE | |
| R-0h | | | | R/W-0h | R-0h | R/W-0h | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COMP_MODE | RTOSINT | 保留 | BUS_SEL | | | | 保留 |
| R/W-0h | R/W-0h | R-0h | R/W-0h | | | | R-0h |

表 26-33 HWBP_CNTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11 | RESERVED | R/W | 0h | 保留。 |
| 10 | RESERVED | R | 0h | 保留。 |
| 9-7 | COMP_MODE | R/W | 0h | 增强型总线比较器（EBC）的比较模式。<br>000：常规掩码比较。<br>以下模式忽略 HWBP_MSK：<br>100：总线值 GT HWBP_REF。<br>101：总线值 GE HWBP_REF。<br>110：总线值 LT HWBP_REF。<br>111：总线值 LE HWBP_REF。<br>GT 表示大于，GE 表示大于或等于，LT 表示小于，LE 表示小于或等于。<br>复位类型：ERAD_RESET |
| 6 | RTOSINT | R/W | 0h | 此位决定增强型总线比较器（EBC）单元是否在发生匹配事件时产生 RTOSINTn 中断。无论此 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 位处于何种状态，事件输出都会产生。<br>0：EBC 单元不对 CPU 采取任何动作。<br>1：对于匹配的数据访问，EBC 单元置有效 RTOSINTn；对于匹配的程序取指，生成跟踪标记。<br>复位类型：ERAD_RESET |
| 5 | RESERVED | R | 0h | 保留。 |
| 4-1 | BUS_SEL | R/W | 0h | 这些位用于选择哪些 CPU 总线将用于比较，以生成匹配事件。对于每个选定的总线，系统将自动选择相应的选通信号以确定有效访问。<br>0000：PAD（指令总线的地址线）；<br>0001：PC-ID（程序计数值）；<br>0010：DWAB（数据写地址总线）；<br>0011：DRAB（数据读地址总线）；<br>0100：DWDB（写数据总线）；<br>0101：DRDB（读数据总线）；<br>0110：PC Aligned Match（PC对齐阶段匹配）；<br>0111：PC R（R阶段匹配）；<br>1000：PC W（W阶段匹配）<br>所有其他组合均为保留。<br>复位类型：ERAD_RESET |
| 0 | RESERVED | R | 0h | 保留。 |

### 26.5.3.5 HWBP_STATUS 寄存器

偏移地址：Eh　复位值：0000h

HWBP（EBC）状态寄存器。

表 26-34 HWBP_STATUS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| STATUS | | 保留 | | | | | |
| R-0h | | R-0h | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | | EVENT_FIRED |
| R-0h | | | | | | | R-0h |

表 26-35 HWBP_STATUS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | STATUS | R | 0h | 总线比较器状态：<br>00：空闲。<br>10：已启用。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 11：已完成。<br>复位类型：ERAD_RESET |
| 13-1 | RESERVED | R | 0h | 保留。 |
| 0 | EVENT_FIRED | R | 0h | 这是一个保持置位的状态位，每当 HWBP（EBC）单元产生匹配事件时置位。软件可据此判断该 HWBP 模块是否触发过事件。向 HWBP_CLEAR 寄存器的位 0 写入 1 可清除此位。<br>复位类型：ERAD_RESET |

## 26.5.4 ERAD_COUNTER_REGS 寄存器

表中列出了 ERAD_COUNTER_REGS 的存储器映射寄存器。表中未列出的寄存器偏移地址均应视为保留位置，不应修改这些位置的寄存器内容。

表 26-36 ERAD_COUNTER_REGS 寄存器

| 偏移地址 | 缩 写 | 寄 存 器 名 称 |
|---|---|---|
| 0h | CTM_CNTL | 计数器控制寄存器 |
| 2h | CTM_STATUS | 计数器状态寄存器 |
| 4h | CTM_REF | 计数器参考寄存器 |
| 8h | CTM_COUNT | 计数器当前值寄存器 |
| Ch | CTM_MAX_COUNT | 计数器最大计数值寄存器 |
| 10h | CTM_INPUT_SEL | 计数器输入选择寄存器 |
| 12h | CTM_CLEAR | 计数器清除寄存器 |
| 14h | CTM_INPUT_SEL_2 | 计数器输入选择扩展寄存器 |
| 16h | CTM_INPUT_COND | 计数器输入调理寄存器 |

复杂的位访问类型采用编码形式，以便放入较小的表格单元格。下表列出了本节使用的访问类型编码。

表 26-37 ERAD_COUNTER_REGS 访问类型编码

| 访问类型 | 编码 | 说明 |
|---|---|---|
| 读类型 | | |
| R | R | 读 |
| R-0 | R<br>-0 | 读<br>返回 0 |
| 写类型 | | |
| W | W | 写 |

| 访问类型 | 编码 | 说明 |
| --- | --- | --- |
| 复位值或默认值 | | |
| -n | | 复位后的值或默认值 |

### 26.5.4.1 CTM_CNTL 寄存器

偏移地址：0h　复位值：0000h

计数器控制寄存器。

表 26-38 CTM_CNTL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 保留 | | | | CNT_INP_SEL_EN | RST_EN | 保留 | START_STOP_CUMULATIVE |
| R-0h | | | | R/W-0h | R/W-0h | R-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RTOSINT | 保留 | 保留 | RST_ON_MATCH | EVENT_MODE | START_STOP_MODE | 保留 | |
| R/W-0h | R-0h | R-0h | R/W-0h | R/W-0h | R/W-0h | R-0h | |

表 26-39 CTM_CNTL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
| --- | --- | --- | --- | --- |
| 15-12 | RESERVED | R | 0h | 保留。 |
| 11 | CNT_INP_SEL_EN | R/W | 0h | 0：禁用 input_select 寄存器对计数输入的选择，计数器始终对 CPU 周期计数。<br>1：启用 input_select 寄存器对计数输入的选择，计数器对计数输入寄存器选定的事件计数。<br>复位类型：ERAD_RESET |
| 10 | RST_EN | R/W | 0h | 此位决定是否启用复位输入。设为 1 时，只要选定的复位输入变为有效高电平，计数器就复位为 0。计数器复位时不产生事件。设为 0 时，计数器忽略复位输入。<br>复位类型：ERAD_RESET |
| 9 | RESERVED | R | 0h | 保留。 |
| 8 | START_STOP_CUMULATIVE | R/W | 0h | 此位决定计数器是累计 n 次连续启动—停止事件的周期数，还是在每次停止事件时清零，以记录连续启动—停止序列中的 MAX_COUNT。<br>0：在 START_STOP 模式下，每次停止事件都清零计数器，MAX_COUNT 记录最大值。<br>1：在 START_STOP 模式下，计数器在连续的启动—停止事件之间累计计数，任何停止事件都不会清零计数器。此位置位时，MAX_COUNT 寄存器无效。<br>复位类型：ERAD_RESET |

| 位 | 字 段 | 访问类型 | 复位值 | 说 明 |
|---|---|---|---|---|
| 7 | RTOSINT | R/W | 0h | 此位决定计数值与参考值匹配时，计数器模块是否产生 RTOSINTn 中断。无论此位处于何种状态，事件输出都会产生。<br>0：计数器单元不对 CPU 采取任何动作。<br>1：计数值与参考值匹配时，计数器单元置有效 RTOSINTn。<br>复位类型：ERAD_RESET |
| 6 | RESERVED | R | 0h | 保留。 |
| 5 | RESERVED | R | 0h | 保留。 |
| 4 | RST_ON_MATCH | R/W | 0h | 此位决定计数器达到参考值后是否复位为 0。<br>0：计数器保持在参考值，进入 COMPLETED 状态并停止继续计数。<br>1：计数器达到匹配值后复位为 0，并保持启用状态。<br>复位类型：ERAD_RESET |
| 3 | EVENT_MODE | R/W | 0h | 此位决定计数器按事件电平还是事件边沿计数。<br>0：只要计数输入处于有效高电平，计数器就递增。<br>1：计数器仅在计数输入的上升沿计数。<br>复位类型：ERAD_RESET |
| 2 | START_STOP_MODE | R/W | 0h | 此位决定计数器是否工作在 START_STOP 模式。<br>0：正常计数模式，计数器不依赖 START 和 STOP 事件。<br>1：启动—停止模式。仅在 START 输入有效后开始计数，并持续对选定事件计数，直至检测到 STOP 事件。<br>复位类型：ERAD_RESET |
| 1-0 | RESERVED | R | 0h | 保留。 |

### 26.5.4.2 CTM_STATUS 寄存器

偏移地址：2h　复位值：0000h

计数器状态寄存器。

表 26-40 CTM_STATUS 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| STATUS | | | | 保留 | | | |
| R-0h | | | | R-0h | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | OVERFLOW | EVENT_FIRED |
| R-0h | | | | | | R-0h | R-0h |

表 26-41 CTM_STATUS 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-12 | STATUS | R | 0h | 计数器单元状态：<br>00：空闲。<br>10：已启用。<br>11：已完成。<br>复位类型：ERAD_RESET |
| 11-2 | RESERVED | R | 0h | 保留。 |
| 1 | OVERFLOW | R | 0h | 这是一个保持置位的状态位，每当计数器达到 0xffffffff 后溢出并回绕时置位。向 CTM_CLEAR 寄存器的位 1（OVERFLOW_CLEAR）写入 1 可清除此位。<br>复位类型：ERAD_RESET |
| 0 | EVENT_FIRED | R | 0h | 这是一个保持置位的状态位，每当 CTM 单元产生匹配事件时置位。软件可据此判断该 CTM 模块是否触发过事件。向 CTM_CLEAR 寄存器的位 0（EVENT_CLEAR）写入 1 可清除此位。<br>复位类型：ERAD_RESET |

### 26.5.4.3 CTM_REF 寄存器

偏移地址：4h　复位值：00000000h

计数器参考寄存器。

表 26-42 CTM_REF 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| REF | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| REF | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| REF | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| REF | | | | | | | |
| R/W-0h | | | | | | | |

表 26-43 CTM_REF 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | REF | R/W | 0h | 此寄存器保存用于比较的计数器参考值。计数值与参考寄存器匹配时（同时考虑寄存器的高、低 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| | | | | 两半），计数器将产生事件。<br>仅当应用程序拥有此单元且 EALLOW 已置位时，CPU 才能写入此寄存器，否则写入将被忽略。仅当调试器拥有此单元时，调试器才能写入此寄存器，否则写入将被忽略。<br>仅当某个 REF 寄存器被写入非零值时，才启用参考值匹配。<br>复位类型：ERAD_RESET |

### 26.5.4.4 CTM_COUNT 寄存器

偏移地址：8h　复位值：00000000h

计数器当前值寄存器。

表 26-44 CTM_COUNT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| COUNT | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| COUNT | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| COUNT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COUNT | | | | | | | |
| R/W-0h | | | | | | | |

表 26-45 CTM_COUNT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | COUNT | R/W | 0h | 此寄存器保存当前计数值。计数值与参考寄存器匹配时（同时考虑寄存器的高、低两半），计数器将产生事件。<br>仅当应用程序拥有此单元且 EALLOW 已置位时，CPU 才能写入此寄存器，否则写入将被忽略。仅当调试器拥有此单元时，调试器才能写入此寄存器，否则写入将被忽略。<br>复位类型：ERAD_RESET |

### 26.5.4.5 CTM_MAX_COUNT 寄存器

偏移地址：Ch　复位值：00000000h

计数器最大计数值寄存器。

表 26-46 CTM_MAX_COUNT 寄存器表

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| MAX_COUNT | | | | | | | |
| R/W-0h | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| MAX_COUNT | | | | | | | |
| R/W-0h | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| MAX_COUNT | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MAX_COUNT | | | | | | | |
| R/W-0h | | | | | | | |

表 26-47 CTM_MAX_COUNT 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 31-0 | MAX_COUNT | R/W | 0h | 此寄存器保存已记录的最大计数值，仅在启动—停止工作模式下有效。<br>仅当应用程序拥有此单元且 EALLOW 已置位时，CPU 才能写入此寄存器，否则写入将被忽略。仅当调试器拥有此单元时，调试器才能写入此寄存器，否则写入将被忽略。<br>复位类型：ERAD_RESET |

### 26.5.4.6 CTM_INPUT_SEL 寄存器

偏移地址：10h　复位值：0000h

计数器输入选择寄存器。

表 26-48 CTM_INPUT_SEL 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| STA_INP_SEL | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CNT_INP_SEL | | | | | | | |
| R/W-0h | | | | | | | |

表 26-49 CTM_INPUT_SEL 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | STA_INP_SEL | R/W | 0h | 这些位决定选择哪个输入作为计数器的 START 事件。这些输入连接到断点模块和计数器模块的事件输出，以及其他系统事件。这些位仅在 START_STOP 计数模式下有效。<br>复位类型：ERAD_RESET |
| 7-0 | CNT_INP_SEL | R/W | 0h | 这些位决定选择哪个输入来使能计数。这些输入连接到断点模块和计数器模块的事件输出，以及其他系统事件。<br>复位类型：ERAD_RESET |

### 26.5.4.7 CTM_CLEAR 寄存器

偏移地址：12h　复位值：0000h

计数器清除寄存器。

表 26-50 CTM_CLEAR 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| R-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | | | | | OVERFLOW_CLEAR | EVENT_CLEAR |
| R-0h | | | | | | R-0/W-0h | R-0/W-0h |

表 26-51 CTM_CLEAR 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-2 | RESERVED | R | 0h | 保留。 |
| 1 | OVERFLOW_CLEAR | R-0/W | 0h | 清除 OVERFLOW。<br>0：不执行操作。<br>1：向此位写入 1，将清除 CTM_STATUS 寄存器中保持置位的 OVERFLOW 位。<br>读取此位始终返回 0。<br>复位类型：ERAD_RESET |
| 0 | EVENT_CLEAR | R-0/W | 0h | 清除 EVENT_FIRED。<br>0：不执行操作。<br>1：向此位写入 1，将清除 CTM_STATUS 寄存器中保持置位的 EVENT_FIRED 位，并使断点模块状态机返回 IDLE 状态。<br>读取此位始终返回 0。<br>复位类型：ERAD_RESET |

### 26.5.4.8 CTM_INPUT_SEL_2 寄存器

偏移地址：14h　复位值：0000h

计数器输入选择扩展寄存器。

表 26-52 CTM_INPUT_SEL_2 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RST_INP_SEL | | | | | | | |
| R/W-0h | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STO_INP_SEL | | | | | | | |
| R/W-0h | | | | | | | |

表 26-53 CTM_INPUT_SEL_2 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-8 | RST_INP_SEL | R/W | 0h | 这些位选择用作复位输入的事件输入，仅在复位使能位（Enable Reset）设为 1 时有效。<br>复位类型：ERAD_RESET |
| 7-0 | STO_INP_SEL | R/W | 0h | 这些位决定选择哪个输入作为计数器的 STOP 事件。这些输入连接到断点模块和计数器模块的事件输出，以及其他系统事件。这些位仅在 START_STOP 计数模式下有效。<br>复位类型：ERAD_RESET |

### 26.5.4.9 CTM_INPUT_COND 寄存器

偏移地址：16h　复位值：0000h

计数器输入调理寄存器。

表 26-54 CTM_INPUT_COND 寄存器表

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| 保留 | | RST_INP_SYNCH | RST_INP_INV | 保留 | | STO_INP_SYNCH | STO_INP_INV |
| R-0h | | R/W-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 保留 | | STA_INP_SYNCH | STA_INP_INV | 保留 | | CTM_INP_SYNCH | CTM_INP_INV |
| R-0h | | R/W-0h | R/W-0h | R-0h | | R/W-0h | R/W-0h |

表 26-55 CTM_INPUT_COND 寄存器字段说明

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 15-14 | RESERVED | R | 0h | 保留。 |

| 位 | 字段 | 访问类型 | 复位值 | 说明 |
|---|---|---|---|---|
| 13 | RST_INP_SYNCH | R/W | 0h | 为选定的复位输入启用两级同步器。<br>复位类型：ERAD_RESET |
| 12 | RST_INP_INV | R/W | 0h | 反转选定的复位输入。<br>复位类型：ERAD_RESET |
| 11-10 | RESERVED | R | 0h | 保留。 |
| 9 | STO_INP_SYNCH | R/W | 0h | 为选定的停止输入启用两级同步器。<br>复位类型：ERAD_RESET |
| 8 | STO_INP_INV | R/W | 0h | 反转选定的停止输入。<br>复位类型：ERAD_RESET |
| 7-6 | RESERVED | R | 0h | 保留。 |
| 5 | STA_INP_SYNCH | R/W | 0h | 为选定的启动输入启用两级同步器。<br>复位类型：ERAD_RESET |
| 4 | STA_INP_INV | R/W | 0h | 反转选定的启动输入。<br>复位类型：ERAD_RESET |
| 3-2 | RESERVED | R | 0h | 保留。 |
| 1 | CTM_INP_SYNCH | R/W | 0h | 为选定的计数器输入启用两级同步器。<br>复位类型：ERAD_RESET |
| 0 | CTM_INP_INV | R/W | 0h | 反转选定的计数器输入。<br>复位类型：ERAD_RESET |

# 27 修订历史记录

表 27-1 修订历史记录

| 版本号 | 描述 | 日期 |
| --- | --- | --- |
| V1.0 | 首次发布 | 2026年7月30日 |
| | | |
| | | |
| | | |