ARM® Cortex®-M0

32-位微控制器

# NuMicro® 家族
# M0519 系列
# 技术参考手册

www.nuvoton.com

目录

1 概述 ..... 10
2 特征 ..... 11
3 缩略语 ..... 14
4 型号信息或引脚表 ..... 16
4.1 M0519 选型指南 ..... 16
4.2 引脚配置 ..... 17
4.3 引脚功能描述 ..... 20
5 方块图 ..... 27
6 功能描述 ..... 28
6.1 ARM® Cortex®-M0 内核 ..... 28
6.2 系统管理 ..... 30
6.3 时钟控制器 ..... 130
6.4 Flash存储控制器 (FMC) ..... 150
6.5 通用 I/O (GPIO) ..... 175
6.6 定时器控制器 (TIMER) ..... 200
6.7 基本PWM发生器和捕获定时器 (BPWM) ..... 217
6.8 增强型PWM发生器 (EPWM) ..... 246
6.9 增强型输入捕获定时器 (ECAP) ..... 279
6.10 看门狗定时器 (WDT) ..... 296
6.11 窗口看门狗定时器 (WWDT) ..... 304
6.12 通用异步收发器 (UART) ..... 314
6.13 I²C串行接口控制器(I²C) ..... 367
6.14 串行外围设备接口 (SPI) ..... 395
6.15 硬件除法器 (HDIV) ..... 428
6.16 增强型模数转换器 (EADC) ..... 435
6.17 模拟比较器 (ACMP) ..... 482
6.18 运算放大器 (OPA) ..... 492
7 电气特性 ..... 499
8 封装尺寸 ..... 500
8.1 LQFP 100L (14x14x1.4 mm封装2.0mm) ..... 500
8.2 LQFP 64L (7x7x1.4 mm 封装 2.0 mm) ..... 501

8.3 LQFP 48L (7x7x1.4mm封装2.0mm) ..... 502
9 修改历史 ..... 503

图集

图 4-1 NuMicro® M0519 选型代码 ..... 16
图 4-2 NuMicro® M0519VxxAE 系列 LQFP-100 引脚图 ..... 17
图 4-3 NuMicro® M0519SxxAE 系列 LQFP-64 引脚图 ..... 18
图 4-4 NuMicro® M0519LxxAE 系列 LQFP-48 引脚图 ..... 19
图 5-1 NuMicro® M0519 系列方块图 ..... 27
图 6-1 控制功能框图 ..... 28
图 6-2 NuMicro® M0519 配电图 ..... 31
图 6-3 时钟发生器框图 ..... 131
图 6-4 时钟发生器全局视图 ..... 132
图 6-5 系统时钟框图 ..... 133
图 6-6 SysTick 时钟控制框图 ..... 133
图 6-7 分频器的时钟源 ..... 134
图 6-8 分频器框图 ..... 134
图 6-9 存储器控制框图 ..... 151
图 6-10 Flash存储器组织结构 ..... 153
图 6-11 BS位启动选择流程示例 ..... 154
图 6-12 IAP使能时代码的执行范围 ..... 155
图 6-13 APROM 和 LDROM 启动的程序执行范围 ..... 156
图 6-14 ISP流程示例 ..... 158
图 6-15 Flash 存储器结构 (DFVSEN = 1) ..... 159
图 6-16 GPIO 控制器 框图 ..... 176
图 6-17 推挽输出 ..... 177
图 6-18 开漏输出 ..... 177
图 6-19 准双向 I/O 模式 ..... 178
图 6-20 增加型PWM输出驱动控制 ..... 179
图 6-21 定时器控制器框图 ..... 201
图 6-22 定时器控制器时钟源 ..... 201
图 6-23 连续计数模式 ..... 203
图 6-24 外部捕捉模式 ..... 204
图 6-25 外部事件复位计数器模式 ..... 205
图 6-26 BPWM 时钟源控制 ..... 218
图 6-27 BPWM 结构图 ..... 218
图 6-28 BPWM定时器内部比较器输出图形说明 ..... 219

图 6-29 BPWM定时器工作时序 ..... 220
图 6-30 BPWM 边沿对齐中断产生时序波形 ..... 220
图 6-31 中心对齐输出波形 ..... 221
图 6-32 BPWM中心对齐中断产生时序波形 ..... 222
图 6-33 BPWM双缓冲说明 ..... 223
图 6-34 BPWM 控制输出占空比 ..... 223
图 6-35 带有死区产生的BPWM对输出 ..... 224
图 6-36 初始状态和极性控制并在上升沿插入死区 ..... 225
图 6-37 BPWM 触发 EADC 时序图 ..... 226
图 6-38 捕获操作时序 ..... 227
图 6-39 BPWM 中断结构框图 ..... 227
图 6-40 EPWM 方块图 ..... 247
图 6-41 EPWM 时钟源控制 ..... 247
图 6-42 PWM 时基发生器 ..... 248
图 6-43 边沿对齐模式PWM ..... 250
图 6-44 PWM0 边沿对齐下的输出波形 ..... 250
图 6-45 边沿对齐流程图 ..... 251
图 6-46 中心对齐模式 ..... 252
图 6-47 PWM0 中心对齐下的输出波形 ..... 253
图 6-48 中心对齐流程图 (INT_TYPE = 0) ..... 254
图 6-49 PWM刹车功能 ..... 255
图 6-50 PWM 刹车条件 (边沿对齐模式) ..... 256
图 6-51 PWM 刹车条件 (中心对齐模式) ..... 256
图 6-52 PWM输出驱动控制 ..... 257
图 6-53 死区时间插入 ..... 258
图 6-54 上升沿死区时间插入时的初始状态和极性控制 ..... 259
图 6-55 图解屏蔽控制 ..... 260
图 6-56 EPWM 触发EADC时序图 ..... 262
图 6-57 EPWM中断的结构图 ..... 263
图 6-58 输入捕获定时器/计数器时钟源控制 ..... 279
图 6-59 输入捕获定时器/计数器架构 ..... 280
图 6-60 噪音过滤器架构 ..... 280
图 6-61 噪音过滤器采样时钟选择 ..... 281
图 6-62 输入捕获定时器/计数器功能模块 ..... 282

图 6-63 输入捕获单元 ..... 283
图 6-64 输入捕获定时器/计数器的中断架构框图 ..... 284
图 6-65 看门狗定时器框图 ..... 296
图 6-66 看门狗定时器时钟 ..... 297
图 6-67 看门狗定时器超时间隔和复位周期时序 ..... 299
图 6-68 窗口看门狗定时器框图 ..... 304
图 6-69 窗口看门狗定时器时钟框图 ..... 305
图 6-70 WWDT 复位和重载行为 ..... 306
图 6-71 UART 时钟控制框图 ..... 315
图 6-72 UART 框图 ..... 315
图 6-73 发送延时操作 ..... 319
图 6-74 UART nCTS 唤醒情况1 ..... 319
图 6-75 UART nCTS 唤醒情况2 ..... 320
图 6-76 自动流控框图 ..... 322
图 6-77 UART nCTS自动流控使能 ..... 323
图 6-78 UART nRTS自动流控使能 ..... 323
图 6-79 UART nRTS软件方式下的自动流控 ..... 324
图 6-80 IrDA 框图 ..... 324
图 6-81 IrDA 时序框图 ..... 326
图 6-82 RS-485在自动方向模式下nRTS的驱动电平 ..... 328
图 6-83 RS-485 在软件控制模式下nRTS的驱动电平 ..... 328
图 6-84 RS-485 帧结构 ..... 329
图 6-85 LIN 帧结构 ..... 330
图 6-86 LIN 字节结构 ..... 330
图 6-87 LIN 模式Break侦测 ..... 333
图 6-88 LIN同步域测量 ..... 336
图 6-89 当LINS_DUM_EN (UA_LIN_CTL[3]) = 1，在自动重同步模式下UA_BAUD 更新顺序 ..... 337
图 6-90 当LINS_DUM_EN (UA_LIN_CTL[3]) = 0，在自动重同步模式下UA_BAUD 更新顺序 ..... 337
图 6-91 $I^2C$ 控制器框图 ..... 368
图 6-92 $I^2C$ 总线时序 ..... 368
图 6-93 $I^2C$ 协议 ..... 369
图 6-94 START 和 STOP 条件 ..... 369
图 6-95 $I^2C$ 总线上的位传输 ..... 371
图 6-96 $I^2C$ 总线上的应答 ..... 371

图 6-97主机向从机传输数据 ..... 371
图 6-98 主机向从机读取数据 ..... 372
图 6-99 通过当前状态控制I²C总线 ..... 372
图 6-100 主机发送模式控制流程 ..... 373
图 6-101 主机接收模式控制流程 ..... 374
图 6-102 从机模式控制流程 ..... 376
图 6-103 GC 模式 ..... 377
图 6-104 总裁丢失 ..... 378
图 6-105 I²C数据移位方向 ..... 379
图 6-106 I²C 超时计数框图 ..... 381
图 6-107 EEPROM 随即读 ..... 382
图 6-108 EEPROM 随机读协议 ..... 383
图 6-109 SPI 模块图 ..... 395
图 6-110 SPI 主模式应用框图 ..... 397
图 6-111 SPI 从模式应用框图 ..... 398
图 6-112 一次传输32位 ..... 399
图 6-113 自动片选 (SS_LVL = 0, SP_CYCLE > 0x2) ..... 400
图 6-114 字节重排序功能 ..... 400
图 6-115 字节间隙时序图 ..... 401
图 6-116 FIFO 阈值判别 ..... 402
图 6-117 FIFO 模式框图 ..... 403
图 6-118 发送 FIFO 缓存实例 ..... 404
图 6-119 接收FIFO 缓存示例 ..... 405
图 6-120 SPI 主模式时序 ..... 407
图 6-121 SPI 主模式时序图 (变换SPI_CLK相位) ..... 408
图 6-122 SPI 从模式时序 ..... 409
图 6-123 SPI 从模式时序 (变换SPI时钟相位) ..... 410
图 6-124 硬件除法器操作流程 ..... 428
图 6-125 ADCA 转换器框图 ..... 436
图 6-126 ADCB 转换器框图 ..... 437
图 6-127 ADC 时钟控制 ..... 437
图 6-128 单次采样模式时序图 ..... 438
图 6-129 SAMPLEA0~3 和 SAMPLEB0~3 控制框图 ..... 439
图 6-130 SAMPLEA4~7 和SAMPLEB4~7控制框图 ..... 440

图 6-131 采样模组转换优先级仲裁算法 ..... 441
图 6-132 PWM触发ADC转换 ..... 441
图 6-133 选择特定的采样模块的转换完成信号产生ADINT0中断 ..... 443
图 6-134 选择特定的采样模块的转换完成信号产生ADINT1 中断 ..... 443
图 6-135 选择特定的采样模块的转换完成信号产生ADINT2 中断 ..... 444
图 6-136 选择特定的采样模块的转换完成信号产生ADINT3 中断 ..... 444
图 6-137 转换开始延迟时间 ..... 445
图 6-138 A/D 增加采样时钟 ..... 446
图 6-139 A/D 转换结果监控逻辑单元 ..... 446
图 6-140 A/D 控制器中断 ..... 447
图 6-141 模拟比较器模块图 ..... 483
图 6-142 模拟比较器控制器中断 ..... 484
图 6-143 比较器迟滞功能 ..... 485
图 6-144 运算放大器模块图 ..... 493
图 6-145 模拟比较器中断设计的运算放大器中断标志位 ..... 494

表集

表 6-1 地址空间分布 ..... 33
表 6-2 异常模式 ..... 74
表 6-3 系统中断向量表 ..... 75
表 6-4 向量表 ..... 76
表 6-5 时钟稳定计数值表 ..... 130
表 6-6 芯片空闲/掉电模式控制表 ..... 137
表 6-7 存储器地址映像 ..... 152
表 6-8 ISP 命令表 ..... 156
表 6-9 刹车源，刹车条件和动作 ..... 256
表 6-10 Even/Odd 输出驱动模式 ..... 257
表 6-11 看门狗定时器超时间隔周期选择 ..... 298
表 6-12 WWDT 分频值选择 ..... 306
表 6-13 WWDT WINCMP 的设置限制 ..... 307
表 6-14 波特率公式表 ..... 317
表 6-15 UART 控制器波特率参数设置示例表 ..... 318
表 6-16 UART 控制器波特率寄存器设置示例表 ..... 319
表 6-17 UART控制器中断源和标志列表 ..... 321
表 6-18 UART 线控的数据位和停止位长度设置 ..... 321
表 6-19 UART 线控校验设置 ..... 322
表 6-20 LIN主模式时报头选择 ..... 331
表 6-21 $I^2C$状态码描述 ..... 380

# 1 概述

M0519 是 ARM® Cortex®-M0 内核 32 位控制器。工作频率可达72MHz，内嵌128K/64K 代码闪存, 4K 数据闪存，8K ISP 闪存，16K SRAM

此芯片内嵌增强的、思考用于电机控制的 PWM，以及Timers, WDT, UART, SPI, I2C, PWM Timer, GPIO, 12-bit ADC, 低压检测。

另外， M0519 有 ISP (In-System Programming)功能, ICP (In-Circuit Programming) 和 IAP (In-Application Programming) 功能。

# 2 特征

- 内核
  - ARM® Cortex®-M0内核，主频最高达 72 MHz
  - 一个24-bit 系统定时器
  - 支持 WFI 唤醒的低功耗模式
  - 单周期 32-位硬件乘法器
  - 4个中断优先级
  - SWD接口防真， 支持两个观察点和四个断点
- 工作电压 2.5V to 5.5V
- 内存
  - 128K/64K 字节闪存用于存放代码(APROM)
  - 4KB 闪存用于存放数据 (Data Flash)
  - 8KB 闪存用于加载代码 (LDROM)
  - 支持 ISP 和 IAP 功能
  - 支持2线 SWD/ICE烧录防真接口
  - 支持并口快速烧录
  - 16K bytes SRAM
- 时钟
  - 内建 22.1184 MHz RC 时钟 HIRC (精度 < 2% @ -40˚C ~ +105˚C)
  - 内建 10 kHz RC 时钟 LIRC 用于WDT 和唤醒功能
  - 内建 4~24 MHz 振荡器 HXT
  - 内建 PLL 可把HXT或HIRC倍频到 72 MHz
  - 有时钟输出脚
- 硬件除法器
  - 支持 32-位有符号被除数， 16-位除数
- GPIO口
  - 输入，输出，准双向，OD门四种模式
  - 输入电平TTL/斯密特选择
  - 可位操作
  - 都可醒置 为电平或边沿中断
  - 电流驱动能力强 (16 mA @ 5V)
  - INT0， INT1 单独的中断向量
  - LQFP100/64/48 封装分别支持 82/51/38 个GPIO
- 定时器
  - 4个 24-位定时器， 带8位预分频功能
  - 定时模式：一次，周期，反转，连续
  - 可对引脚信号计数
- 看门狗
  - 时钟源 LIRC(预设) 或 HCLK/2048
  - 8 个溢出时间选择： 1.6ms ~ 26.0s
  - 能把CPU从掉电状态唤醒
  - 溢出时，复位功中断功能可选择
  - 复位延迟时间可选
- 窗看门狗

- 时钟 HCLK/2048 (预设) 或 LIRC
- 窗 6位计数， 11位预分频
- 能把CPU从掉电状态 唤醒

- 基本 PWM
  - 1 个16-位 PWM, 控制两个PWM 引脚
  - 可配置成输入捕获功能
- 增强 PWM
  - 两个16位PWM, 控制 6个PWM引脚， 有死区控制，极性控制和刹车功能
  - 复位后PWM引脚输入功能
- 增强输入捕获
  - 2个 24-位输入捕获单元
  - 每个单元可3 个输入: ECAPx_IC0, ECAPx_IC1 和 ECAPx_IC2
- UART
  - 两个与 16550 兼容的 UART
  - 波特率可编程
  - 收发各 16 bytes FIFO
  - 支持流控 (TX, RX, CTS 和 RTS)
  - 支持红外功能
  - 支持RS-485功能
- SPI
  - 3 个 SPI
  - 支持主/从模式
  - 全双工同步收发
  - 收发长度 8 to 32 位
  - 高低位在前可选
  - Rx， Tx 单独醒置上沿或下沿
  - 32-位传输可插入空闲时间
- $I^2C$
  - 主 / 人模式，位率最高到 1 Mbit/s
  - 双向数据传输
  - 支持多主通信
  - 多主支持仲裁
  - 通信速率可编程
  - 4 多个从机地址带掩码
- ADC
  - 两个 ADC
  - 每个ADC有 8 个通道, 转换数据可醒置12位或10位
  - 16 个结果寄存器
  - 采样率 800ksps
  - 俩操作模式:
    - ◆ 单采样模式: 只转换一个通道.
    - ◆ 同时采样模式: 两个通道同时转换.
  - 两个比较寄存器
  - 启动转换：软件，引脚触发，Timer 0~3 或 PWM
- 3 个模拟比较器

- 两个运放
- BOD电压
  - 4.4V/3.7V/2.7V/2.2V
  - 中断或复位功能
- 操作电压: 2.5V to 5.5V
- 低压复位
- 96-位唯一序列号
- 操作温度: -40℃~105℃
- 开发工具: 并口烧录或 ICP
- 封装:
  - 符合绿色环保标准(RoHS)
  - LQFP 100/64/48-pin

# 3 缩略语

| 缩写 | 描述 |
|---|---|
| ACMP | Analog Comparator Controller |
| ADC | Analog-to-Digital Converter |
| AES | Advanced Encryption Standard |
| APB | Advanced Peripheral Bus |
| AHB | Advanced High-Performance Bus |
| BOD | Brown-out Detection |
| CAN | Controller Area Network |
| DAP | Debug Access Port |
| DES | Data Encryption Standard |
| EBI | External Bus Interface |
| EPWM | Enhanced Pulse Width Modulation |
| FIFO | First In, First Out |
| FMC | Flash Memory Controller |
| FPU | Floating-point Unit |
| GPIO | General-Purpose Input/Output |
| HCLK | The Clock of Advanced High-Performance Bus |
| HIRC | 22.1184 MHz Internal High Speed RC Oscillator |
| HXT | 4~24 MHz External High Speed Crystal Oscillator |
| IAP | In Application Programming |
| ICP | In Circuit Programming |
| ISP | In System Programming |
| LDO | Low Dropout Regulator |
| LIN | Local Interconnect Network |
| LIRC | 10 kHz internal low speed RC oscillator (LIRC) |
| MPU | Memory Protection Unit |
| NVIC | Nested Vectored Interrupt Controller |
| PCLK | The Clock of Advanced Peripheral Bus |
| PDMA | Peripheral Direct Memory Access |
| PLL | Phase-Locked Loop |
| PWM | Pulse Width Modulation |
| QEI | Quadrature Encoder Interface |
| SDIO | Secure Digital Input/Output |
| SPI | Serial Peripheral Interface |

| | |
|---|---|
| SPS | Samples per Second |
| TDES | Triple Data Encryption Standard |
| TMR | Timer Controller |
| UART | Universal Asynchronous Receiver/Transmitter |
| UCID | Unique Customer ID |
| USB | Universal Serial Bus |
| WDT | Watchdog Timer |
| WWDT | Window Watchdog Timer |

# 4 型号信息或引脚表

## 4.1 M0519 选型指南

### 4.1.1 M0519 选 型指南

| Part Number | APROM (KB) | RAM (KB) | Data Flash (KB) | LDROM (KB) | I/O | Timer (32-Bit) | Connectivity | | | | Capture | PWM | ADC (12-Bit) | OPA | Comp. | ISP/ICP/IAP | Package |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | UART | SPI | $I^2C$ | LIN | | | | | | | |
| M0519LD3AE | 64 | 16 | 4 | 8 | 38 | 4 | 2 | 1 | 1 | 2 | - | 6 | 16 | - | 2 | v | LQFP48 |
| M0519LE3AE | 128 | 16 | Config. | 8 | 38 | 4 | 2 | 1 | 1 | 2 | - | 6 | 16 | - | 2 | v | LQFP48 |
| M0519SD3AE | 64 | 16 | 4 | 8 | 51 | 4 | 2 | 2 | 1 | 2 | - | 10 | 16 | 1 | 2 | v | LQFP64 |
| M0519SE3AE | 128 | 16 | Config. | 8 | 51 | 4 | 2 | 2 | 1 | 2 | - | 10 | 16 | 1 | 2 | v | LQFP64 |
| M0519VE3AE | 128 | 16 | Config. | 8 | 82 | 4 | 2 | 3 | 1 | 2 | 6 | 14 | 16 | 2 | 3 | v | LQFP100 |

### 4.1.2 NuMicro® M0519 命名规则

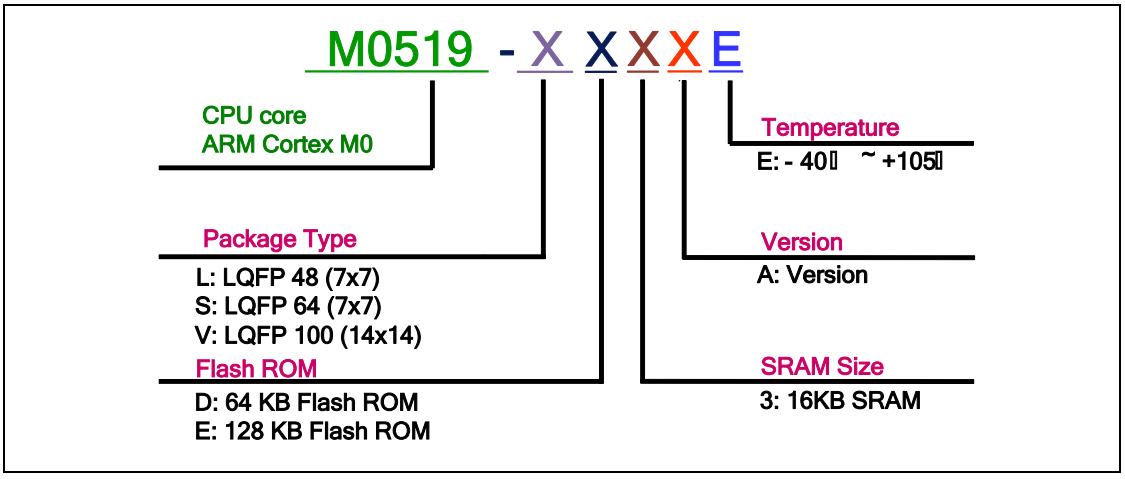



图 4-1 NuMicro® M0519 选型代码

## 4.2 引脚配置

### 4.2.1 LQFP 100-pin

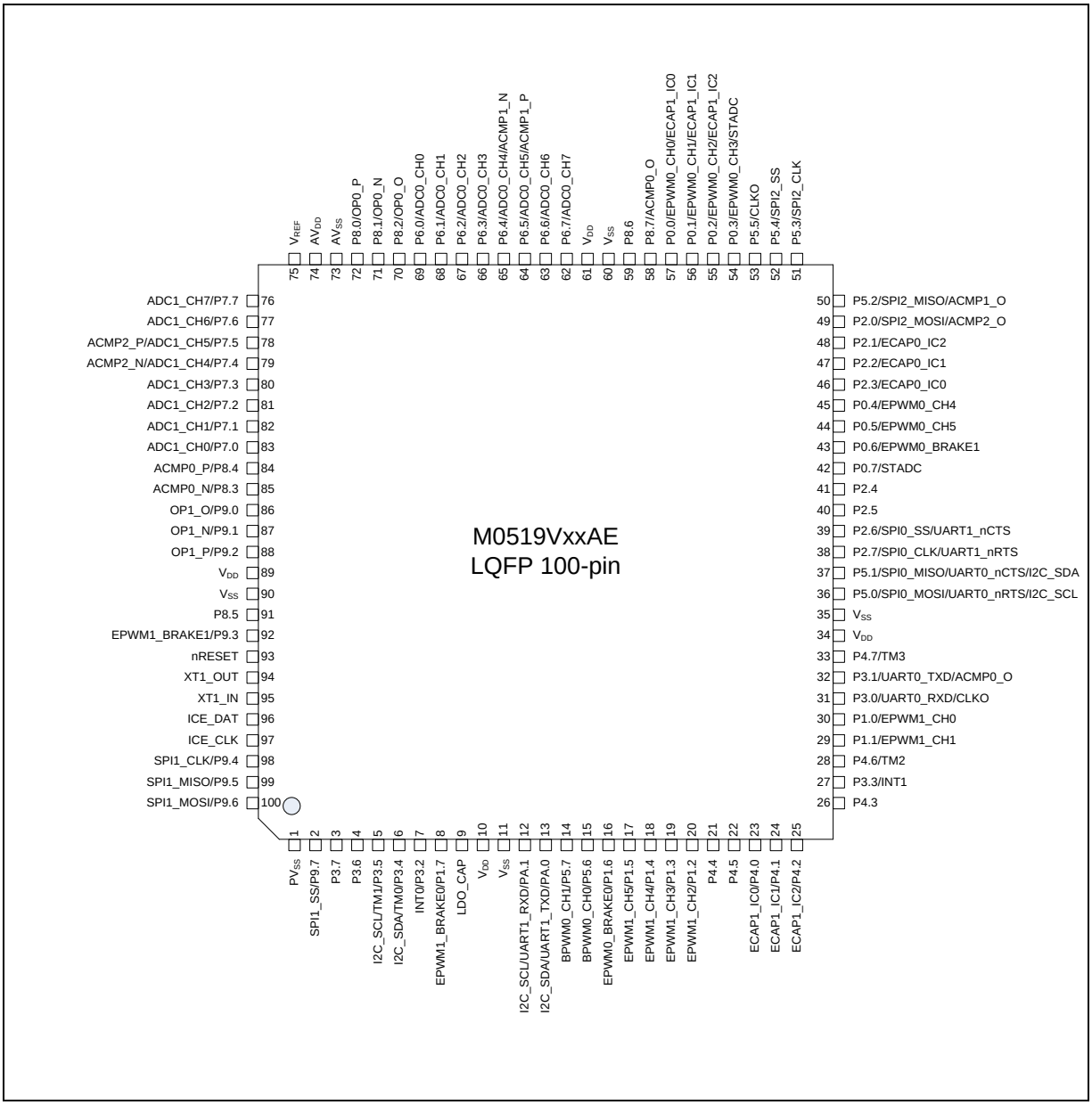



图 4-2 NuMicro® M0519VxxAE 系列 LQFP-100 引脚图

### 4.2.2 LQFP 64-pin

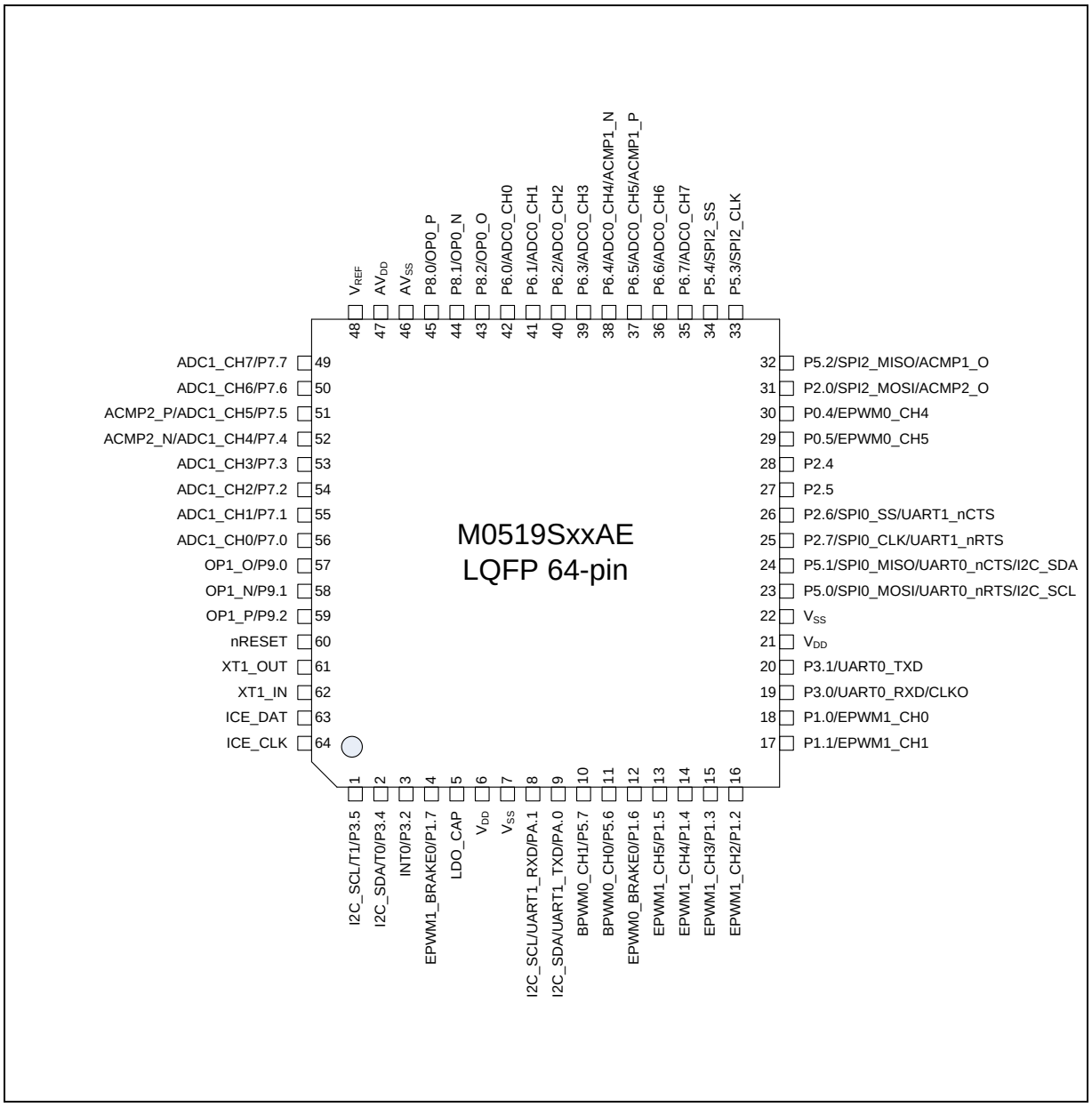



图 4-3 NuMicro® M0519SxxAE 系列 LQFP-64 引脚图

### 4.2.3 LQFP 48-pin

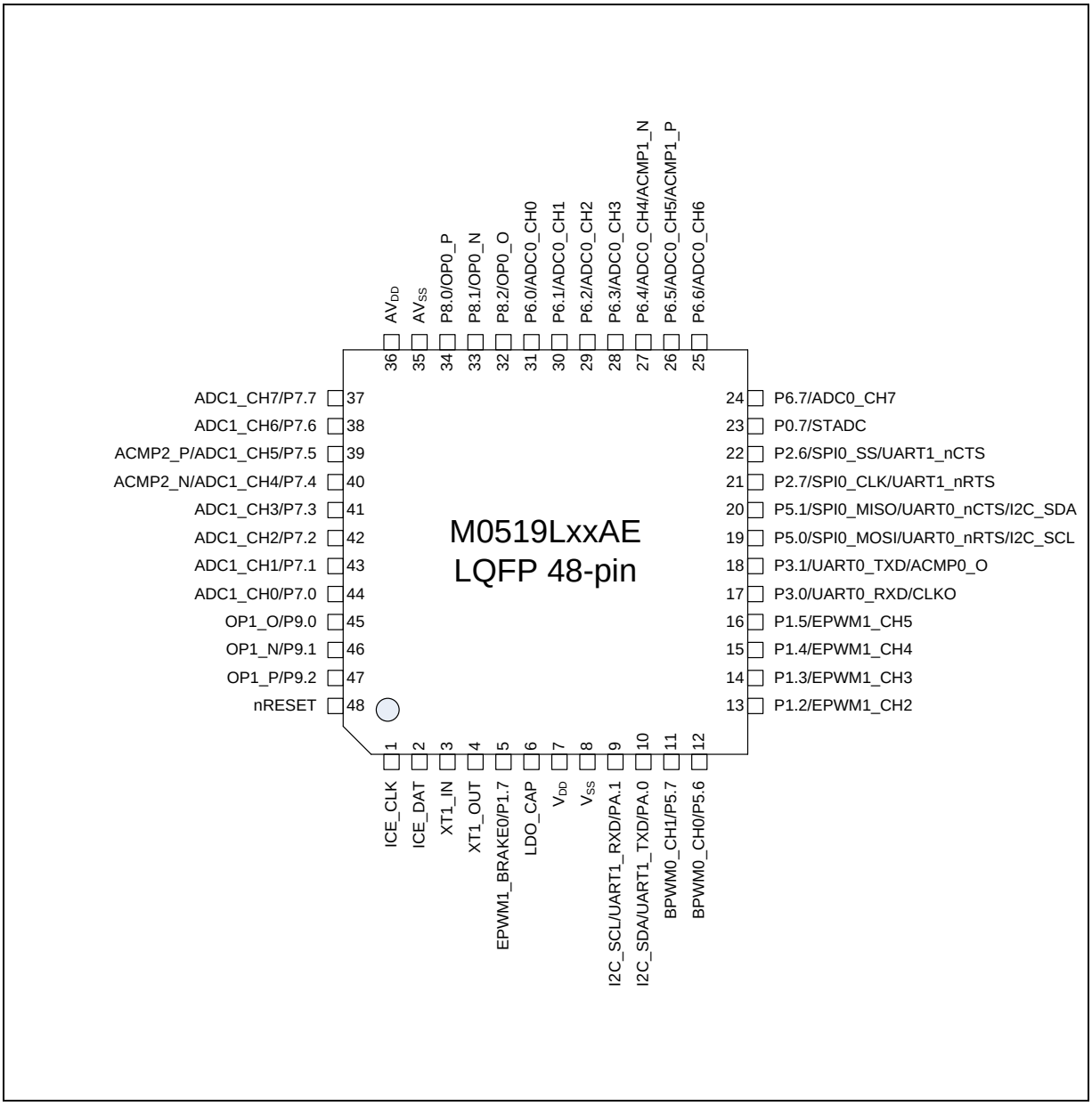



图 4-4 NuMicro® M0519LxxAE 系列 LQFP-48 引脚图

## 4.3 引脚功能描述

| 引脚 号 | | | 引脚名 | 引脚型号[1] | 描述 |
|---|---|---|---|---|---|
| 100- pin | 64-pin | 48-pin | | | |
| 10 | 6 | 7 | $V_{DD}$ | P | 数字电源 |
| 34 | | | | | |
| 61 | 21 | | | | |
| 89 | | | | | |
| 11 | 7 | 8 | $V_{SS}$ | P | 数字地 |
| 35 | | | | | |
| 60 | 22 | | | | |
| 90 | | | | | |
| 9 | 5 | 6 | LDO_CAP | P | LDO 输出，需外接 1uF 电容 |
| 1 | - | - | $PV_{SS}$ | P | PLL 地 |
| 74 | 47 | 36 | $AV_{DD}$ | AP | 模拟电源 |
| 73 | 46 | 35 | $AV_{SS}$ | AP | 模电 拟地 |
| 75 | 48 | - | $V_{REF}$ | AP | ADC参考电压输入<br>需接 1uF 电容. |
| 93 | 60 | 48 | nRESET | I (ST) | 复位: 内有上拉，外对地接电容即可，低电平时间要大于768个22M 时钟周期 |
| 94 | 61 | 4 | XT1_OUT | O | |
| 95 | 62 | 3 | XT1_IN | I (ST) | |
| 96 | 63 | 2 | ICE_DAT | I/O | |
| 97 | 64 | 1 | ICE_CLK | I | |
| 57 | - | - | P0.0 | I/O | |
| | | | PWM0_CH0 | O | PWM00 输出脚 |
| | | | ECAP1_IC0 | I | 捕获10 输入脚 |
| 56 | - | - | P0.1 | I/O | |
| | | | PWM0_CH1 | O | PWM01输出脚 |
| | | | ECAP1_IC1 | I | 捕获11输入脚 |
| 55 | - | - | P0.2 | I/O | |
| | | | PWM0_CH2 | O | PWM02输出脚 |
| | | | ECAP1_IC2 | I | 捕获12输入脚 |
| 54 | - | - | P0.3 | I/O | |

| 引脚 号 | | | 引脚名 | 引脚型号[1] | 描述 |
|---|---|---|---|---|---|
| 100- pin | 64-pin | 48-pin | | | |
| | | | PWM0_CH3 | O | PWM03输出脚 |
| | | | STADC | I | ADC 触发引脚 |
| 45 | 30 | - | P0.4 | I/O | |
| | | | PWM0_CH4 | O | PWM04输出脚 |
| 44 | 29 | - | P0.5 | I/O | |
| | | | PWM0_CH5 | O | PWM05输出脚 |
| 43 | - | - | P0.6 | I/O | |
| | | | PWM0_BRAKE1 | I | PWM0 的Brake1 功能引脚 |
| 42 | - | 23 | P0.7 | I/O | |
| | | | STADC | I | ADC 触发引脚 |
| 30 | 18 | - | P1.0 | I/O | |
| | | | PWM1_CH0 | O | PWM10输出脚 |
| 29 | 17 | - | P1.1 | I/O | |
| | | | PWM1_CH1 | O | PWM11输出脚 |
| 20 | 16 | 13 | P1.2 | I/O | |
| | | | PWM1_CH2 | O | PWM12输出脚 |
| 19 | 15 | 14 | P1.3 | I/O | |
| | | | PWM1_CH3 | O | PWM13输出脚 |
| 18 | 14 | 15 | P1.4 | I/O | |
| | | | PWM1_CH4 | O | PWM14输出脚 |
| 17 | 13 | 16 | P1.5 | I/O | |
| | | | PWM1_CH5 | O | PWM15输出脚 |
| 16 | 12 | - | P1.6 | I/O | |
| | | | PWM0_BRAKE0 | I | PWM0的 Brake0功能引脚 |
| 8 | 4 | 5 | P1.7 | I/O | |
| | | | PWM1_BRAKE0 | I | PWM1的 Brake0功能引脚 |
| 49 | 31 | - | P2.0 | I/O | |
| | | | SPI2_MOSI | I/O | SPI2 主出从入 |
| | | | ACMP2_O | AO | 模拟比较器2输出 |
| 48 | - | - | P2.1 | I/O | |
| | | | ECAP0_IC2 | I | 捕获0的输入2 |
| 47 | - | - | P2.2 | I/O | |

| 引脚 号 | | | 引脚名 | 引脚型号[1] | 描述 |
|---|---|---|---|---|---|
| 100- pin | 64-pin | 48-pin | | | |
| | | | ECAP0_IC1 | I | 捕获0的输入1 |
| 46 | - | - | P2.3 | I/O | |
| | | | ECAP0_IC0 | I | 捕获0的输入0 |
| 41 | 28 | - | P2.4 | I/O | |
| 40 | 27 | - | P2.5 | I/O | |
| 39 | 26 | 22 | P2.6 | I/O | |
| | | | SPI0_SS | I/O | SPI0从机片选 |
| | | | UART1_nCTS | I | UART1 CTS 引脚 |
| 38 | 25 | 21 | P2.7 | I/O | |
| | | | SPI0_CLK | I/O | SPI0 时钟引脚 |
| | | | UART1_nRTS | O | UART1 RTS引脚 |
| 31 | 19 | 17 | P3.0 | I/O | |
| | | | UART0_RXD | I | UART0 接收 |
| 32 | 20 | 18 | P3.1 | I/O | |
| | | | UART0_TXD | O | UART0 发送 |
| | | | ACMP0_O | AO | 比较器0输出 |
| 7 | 3 | - | P3.2 | I/O | |
| | | | INT0 | I | 中断0输入脚 |
| 27 | - | - | P3.3 | I/O | |
| | | | INT1 | I | 中断1输入脚 |
| 6 | 2 | - | P3.4 | I/O | |
| | | | TM0 | I/O | Timer0 外部时钟 |
| | | | I2C0_SDA | I/O | I2C0 数据 |
| 5 | 1 | - | P3.5 | I/O | |
| | | | TM1 | I/O | Timer1 外时钟 |
| | | | I2C0_SCL | I/O | I2C0 时钟 |
| 4 | - | - | P3.6 | I/O | |
| 3 | - | - | P3.7 | I/O | |
| 23 | - | - | P4.0 | I/O | |
| | | | ECAP1_IC0 | I | 捕获1的输入0 |
| 24 | - | - | P4.1 | I/O | |
| | | | ECAP1_IC1 | I | 捕获1的输入1 |

| 引脚 号 | | | 引脚名 | 引脚型号[1] | 描述 |
|---|---|---|---|---|---|
| 100- pin | 64-pin | 48-pin | | | |
| 25 | - | - | P4.2 | I/O | |
| | | | ECAP1_IC2 | I | 捕获1的输入2 |
| 26 | - | - | P4.3 | I/O | |
| 21 | - | - | P4.4 | I/O | |
| 22 | - | - | P4.5 | I/O | |
| 28 | - | - | P4.6 | I/O | |
| | | | TM2 | I/O | Timer2 外时钟 |
| 33 | - | - | P4.7 | I/O | |
| | | | TM3 | I/O | Timer3 外时钟 |
| 36 | 23 | 19 | P5.0 | I/O | |
| | | | SPI0_MOSI | I/O | SPI0 主出从入 |
| | | | UART0_nRTS | O | UART0 RTS引脚 |
| 37 | 24 | 20 | P5.1 | I/O | |
| | | | SPI0_MISO | I/O | SPI0 主入从出 |
| | | | UART0_nCTS | I | UART0 CTS引脚 |
| 50 | 32 | - | P5.2 | I/O | |
| | | | SPI2_MISO | I/O | SPI2 主入从出 |
| | | | ACMP1_O | AO | 比较器1输出 |
| 51 | 33 | - | P5.3 | I/O | |
| | | | SPI2_CLK | I/O | SPI2 时钟引脚 |
| 52 | 34 | - | P5.4 | I/O | |
| | | | SPI2_SS | I/O | SPI2 片选引脚 |
| 53 | - | - | P5.5 | I/O | |
| | | | CLKO | O | 时钟输出 |
| 15 | 11 | 12 | P5.6 | I/O | |
| | | | PWM2_CH0 | I/O | PWM20引脚 |
| 14 | 10 | 11 | P5.7 | I/O | |
| | | | PWM2_CH1 | I/O | PWM21 引脚 |
| 69 | 42 | 31 | P6.0 | I/O | |
| | | | ADC0_CH0 | AI | ADC00 引脚 |
| 68 | 41 | 30 | P6.1 | I/O | |
| | | | ADC0_CH1 | AI | ADC01 引脚 |

| 引脚 号 | | | 引脚名 | 引脚型号[1] | 描述 |
|---|---|---|---|---|---|
| 100- pin | 64-pin | 48-pin | | | |
| 67 | 40 | 29 | P6.2 | I/O | |
| | | | ADC0_CH2 | AI | ADC02引脚 |
| 66 | 39 | 28 | P6.3 | I/O | |
| | | | ADC0_CH3 | AI | ADC03引脚 |
| 65 | 38 | 27 | P6.4 | I/O | |
| | | | ADC0_CH4 | AI | ADC04引脚 |
| | | | ACMP1_N | AI | 比较器1负输入 |
| 64 | 37 | 26 | P6.5 | I/O | |
| | | | ADC0_CH5 | AI | ADC05引脚 |
| | | | ACMP1_P | AI | 比较器1正输入 |
| 63 | 36 | 25 | P6.6 | I/O | |
| | | | ADC0_CH6 | AI | ADC06引脚 |
| 62 | 35 | 24 | P6.7 | I/O | |
| | | | ADC0_CH7 | AI | ADC07引脚 |
| 83 | 56 | 44 | P7.0 | I/O | |
| | | | ADC1_CH0 | AI | ADC10引脚 |
| 82 | 55 | 43 | P7.1 | I/O | |
| | | | ADC1_CH1 | AI | ADC11引脚 |
| 81 | 54 | 42 | P7.2 | I/O | |
| | | | ADC1_CH2 | AI | ADC12引脚 |
| 80 | 53 | 41 | P7.3 | I/O | |
| | | | ADC1_CH3 | AI | ADC13引脚 |
| 79 | 52 | 40 | P7.4 | I/O | |
| | | | ADC1_CH4 | AI | ADC 14引脚 |
| | | | ACMP2_N | AI | 比较器2负输入 |
| 78 | 51 | 39 | P7.5 | I/O | |
| | | | ADC1_CH5 | AI | ADC15引脚 |
| | | | ACMP2_P | AI | A比较器2正输入 |
| 77 | 50 | 38 | P7.6 | I/O | |
| | | | ADC1_CH6 | AI | ADC16引脚 |
| 76 | 49 | 37 | P7.7 | I/O | |
| | | | ADC1_CH7 | AI | ADC17引脚 |

| 引脚 号 | | | 引脚名 | 引脚型号[1] | 描述 |
|---|---|---|---|---|---|
| 100- pin | 64-pin | 48-pin | | | |
| 72 | 45 | 34 | P8.0 | I/O | |
| | | | OP0_P | AI | 运放0正输入 |
| 71 | 44 | 33 | P8.1 | I/O | |
| | | | OP0_N | AI | 运放0负输入 |
| 70 | 43 | 32 | P8.2 | I/O | |
| | | | OP0_O | AO | 运放0输出 |
| 85 | - | - | P8.3 | I/O | |
| | | | ACMP0_N | AI | 模拟比较器0负输入端 |
| 84 | - | - | P8.4 | I/O | |
| | | | ACMP0_P | AI | 模拟比较器0正输入端 |
| 91 | - | - | P8.5 | I/O | |
| 59 | - | - | P8.6 | I/O | |
| 58 | - | - | P8.7 | I/O | |
| | | | ACMP0_O | O | 模拟比较器0输出端 |
| 86 | 57 | 45 | P9.0 | I/O | |
| | | | OP1_O | AO | 运放1输出 |
| 87 | 58 | 46 | P9.1 | I/O | |
| | | | OP1_N | AI | 运放1负输入 |
| 88 | 59 | 47 | P9.2 | I/O | |
| | | | OP1_P | AI | 运放1正输入 |
| 92 | - | - | P9.3 | I/O | |
| | | | PWM1_BRAKE1 | I | PWM1 Brake1功能引脚 |
| 98 | - | - | P9.4 | I/O | |
| | | | SPI1_CLK | I/O | SPI1时钟 |
| 99 | - | - | P9.5 | I/O | |
| | | | SPI1_MISO | I/O | SPI1 主入从出 |
| 100 | - | - | P9.6 | I/O | |
| | | | SPI1_MOSI | I/O | SPI1 主出从入 |
| 2 | - | - | P9.7 | I/O | |
| | | | SPI1_SS | I/O | SPI1从机片选 |
| 13 | 9 | 10 | PA.0 | I/O | |
| | | | UART1_TXD | O | UART1 发送 |

| 引脚 号 | | | 引脚名 | 引脚型号[1] | 描述 |
|---|---|---|---|---|---|
| 100- pin | 64-pin | 48-pin | | | |
| | | | I2C0_SDA | **I/O** | I2C0 数据 |
| 12 | 8 | 9 | PA.1 | **I/O** | |
| | | | UART1_RXD | **I** | UART1 接收 |
| | | | I2C0_SCL | **I/O** | I2C0 时钟 |

**注意:** 引脚类型 I = 数字输入, O = 数字输出; AI = 模拟输入; P = 电源; AP = 模拟电源

# 5 方块图

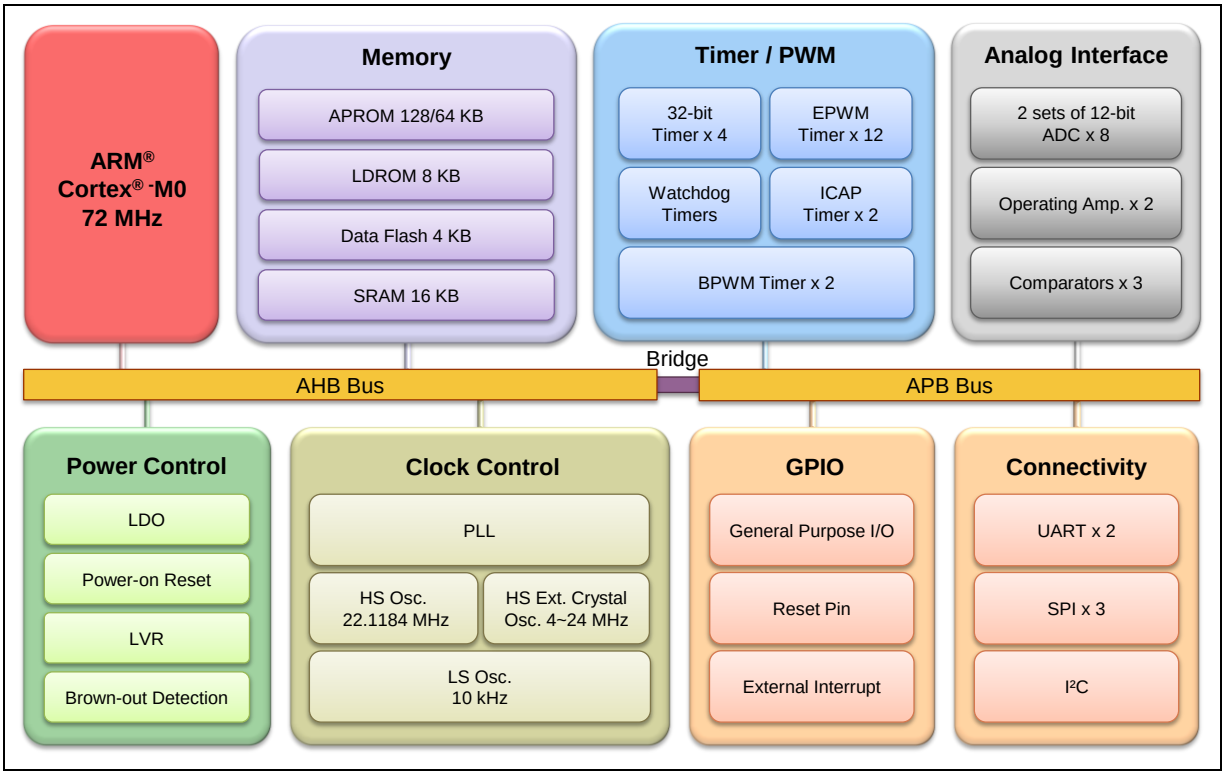



图 5-1 NuMicro® M0519 系列方块图

# 6 功能描述

## 6.1 ARM® Cortex®-M0 内核

Cortex®-M0 是一个可配置，多级，32位精简指令处理器。 执行Thumb 指令，有硬件调试功能。有两种工作模式：句柄模式——进入中断后的模式；线程模式——复位后的模式，中断返回也可以进入这种模式。

图 6-1 图示的处理器的功能.

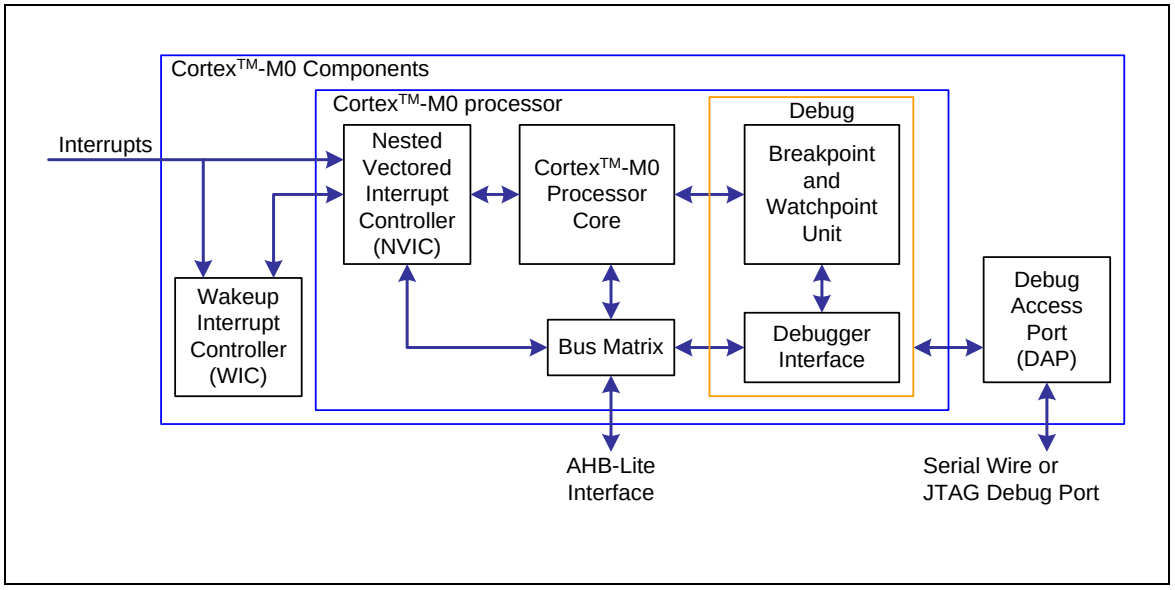



图 6-1 控制功能框图

具备以下功能:

- 低逻辑门数处理器:
  - ARMv6-M Thumb® 指令集
  - Thumb-2 技术
  - ARMv6-M 兼容的24位系统定时器
  - 32-bit 硬件乘法器
  - 数据为小端格式
  - 确定的中断响应时间
  - 中断时，可打断多周期乘法和 加载/存储复合指令
  - 中断代码可用纯C语言写
  - 进入低功耗指令 (WFI，WFE), 可配置中断返回自动再进入低功耗功能
- NVIC:
  - 32个中断源，4个优先级
  - (NMI) 非屏蔽中断功能

- 支持电平中断，和跳变中断功能
- 掉电模式支持中断唤醒

- 调式功能
  - 四个断点
  - 两个观察点
  - PC指针采样寄存器 (PCSR)
  - 单步和向量抓取功能
- 总线接口:
  - 单 32-位 AMBA-3 AHB-Lite 系统接口
  - 单 32-位从口支持 DAP (Debug Access Port)

## 6.2 系统管理

### 6.2.1 概况

系统管理包含以下功能:

- 系统复位
- 系统电源
- 系统内存布局
- 系统管理寄存器：PID, 芯片复位 , 多功能引脚 控制
- 系统定时器 (SysTick)
- 中断向量控制 (NVIC)
- 系统控制寄存器

### 6.2.2 系统复位

包括以下复位源，复位后可在寄存器 RSTSRC查到是哪个复位源引起的复位。

- 硬件复位
  - 上电复位 (POR)
  - 复位引脚复位 (nRESET)
  - 看门狗复位 (WDT)
  - 低压复位 (LVR)
  - BOD复位 (BOD)
- 软件复位
  - SYS 系统复位 - SYSRESETREQ (AIRCR[2])
  - CPU复位 CPU_RST (IPRSTC1[1])
  - 芯片复位 CHIP_RST (IPRSTC1[0])

芯片复位 CHIP_RST (IPRST1[0]) 与上电复位相同。

系统复位 SYSRESETREQ (AIRCR[2]) 不复位晶振和 BS (ISPCON[1]) 位，其它都复位。

### 6.2.3 系统电源描述

配电划分为两部分.

- 模拟部分由 $AV_{DD}$， $AV_{SS}$ 供电。
- 数字部分和内部1.8V LDO由 $V_{DD}$， $V_{SS}$ 供电。

片内LDO输出端 LDO_CAP, 应在最靠近引脚处接个电容。模拟电压 ($AV_{DD}$) 应与数字电压($V_{DD}$)大小一样。

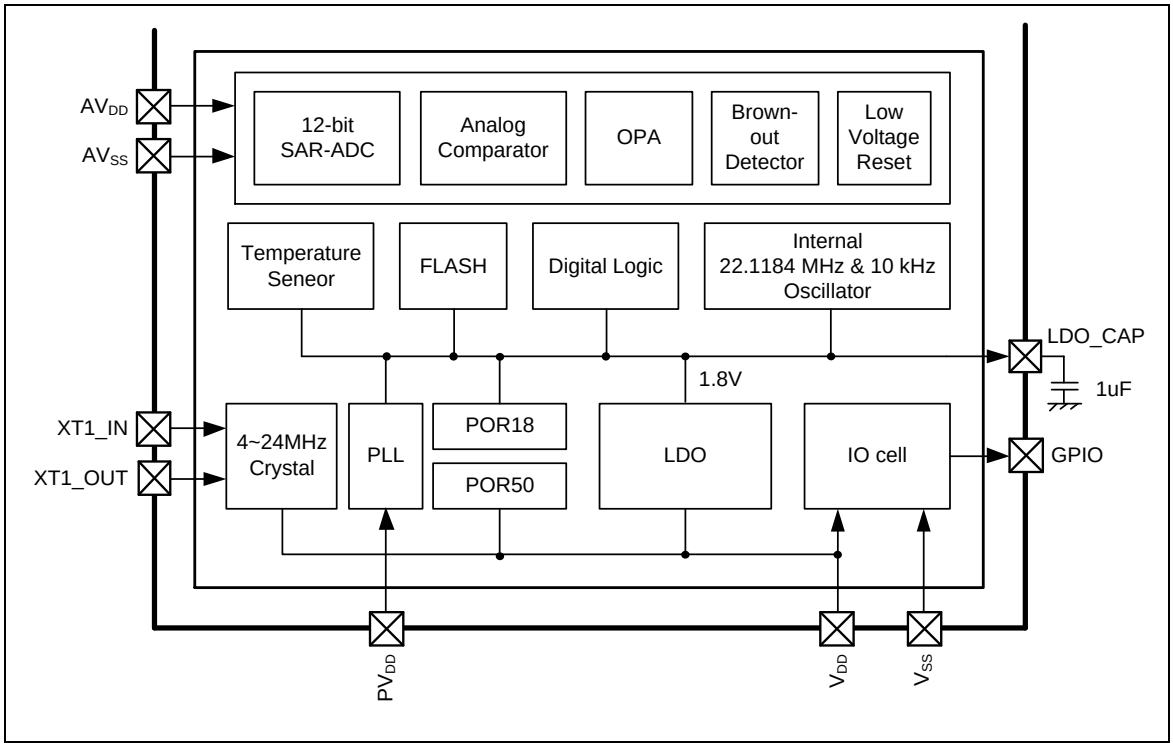



图 6-2 NuMicro® M0519 配电图

### 6.2.4 系统内存布局

M0519 有 4G-byte 地址空间。具体分配见表 6-1. M0519 数据仅支持小端格式。

| 地址空间 | 符号 | 描述 |
|---|---|---|
| **Flash and SRAM Memory Space** | | |
| 0x0000_0000 – 0x0001_FFFF | FLASH_BA | FLASH 存储空间 (128 KB) |
| 0x2000_0000 – 0x2000_3FFF | SRAM_BA | SRAM 存储空间 (16 KB) |
| **AHB 控制空间 (0x5000_0000 – 0x501F_FFFF)** | | |
| 0x5000_0000 – 0x5000_01FF | GCR_BA | 系统全局控制寄存器 |
| 0x5000_0200 – 0x5000_02FF | CLK_BA | 时钟控制寄存器 |
| 0x5000_0300 – 0x5000_03FF | INT_BA | 多路中断控制寄存器 |
| 0x5000_4000 – 0x5000_7FFF | GPIO_BA | GPIO控制寄存器 |
| 0x5000_C000 – 0x5000_FFFF | FMC_BA | Flash 内存控制寄存器 |
| 0x5001_4000 – 0x5001_7FFF | HDIV_BA | 硬件除法器寄存器 |
| **APB1控制空间 (0x4000_0000 ~ 0x400F_FFFF)** | | |
| 0x4000_4000 – 0x4000_7FFF | WDT_BA | 看门狗定时器控制寄存器 |
| 0x4000_4100 – 0x4000_7FFF | WWDT_BA | 窗看门狗定时器控制寄存器 |
| 0x4001_0000 – 0x4001_3FFF | TMR01_BA | Timer0/Timer1 控制寄存器 |
| 0x4002_0000 – 0x4002_3FFF | I2C0_BA | $I^2C0$ 接口控制寄存器 |
| 0x4003_0000 – 0x4003_3FFF | SPI0_BA | 带主/从功能的SPI0控制寄存器 |
| 0x4003_4000 – 0x4003_7FFF | SPI1_BA | 带主/从功能的SPI1控制寄存器 |
| 0x4004_0000 – 0x4004_3FFF | BPWM0_BA | 基本 PWM0控制寄存器 |
| 0x4005_0000 – 0x4005_3FFF | UART0_BA | UART0控制寄存器 |
| 0x400D_0000 – 0x400D_3FFF | ACMP_BA | 模拟比较器控制寄存器 |
| 0x400E_0000 – 0x400E_3FFF | EADC_BA | 增强型模拟数字转换（EADC）控制寄存器 |
| 0x400F_0000 – 0x400F_3FFF | OPA_BA | 运算放大器控制寄存器 |
| **APB2控制空间 (0x4010_0000 ~ 0x401F_FFFF)** | | |
| 0x4011_0000 – 0x4011_3FFF | TMR23_BA | Timer2/Timer3控制寄存器 |
| 0x4013_0000 – 0x4013_3FFF | SPI2_BA | 带主/从功能的SPI2控制寄存器 |
| 0x4015_0000 – 0x4015_3FFF | UART1_BA | UART1控制寄存器 |
| Reserved | | |
| 0x4019_0000 – 0x4019_3FFF | EPWM0_BA | 增强型 PWM0控制寄存器 |
| 0x4019_4000 – 0x4019_7FFF | EPWM1_BA | 增强型 PWM1控制寄存器 |

| 地址空间 | 符号 | 描述 |
|---|---|---|
| 0x401B_0000 – 0x401B_3FFF | ECAP0_BA | 增强型 输入捕获 0控制寄存器 |
| 0x401B_4000 – 0x401B_7FFF | ECAP1_BA | 增强型 输入捕获 1控制寄存器 |
| Reserved | | |
| **系统控制空间 (0xE000_E000 ~ 0xE000_EFFF)** | | |
| 0xE000_E010 – 0xE000_E01F | SYST_BA | 系统定时器控制寄存器 |
| 0xE000_E100 – 0xE000_E4EF | NVIC_BA | 外部中断控制器控制寄存器 |
| 0xE000_ED00 – 0xE000_ED3F | SCS_BA | 系统控制寄存器 |

表 6-1 地址空间分布

### 6.2.5 系统管理控制寄存器

**R**: 只读, **W**: 只写, **R/W**: 读/写

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **GCR 基地址:**<br>**GCR_BA = 0x5000_0000** | | | | |
| **PDID** | GCR_BA+0x00 | R | 器件ID | 0x0035_X0XX |
| **RSTSRC** | GCR_BA+0x04 | R/W | 系统复位源寄存器 | 0x0000_00XX |
| **IPRSTC1** | GCR_BA+0x08 | R/W | 复位控制寄存器1 | 0x0000_0000 |
| **IPRSTC2** | GCR_BA+0x0C | R/W | 复位控制寄存器2 | 0x0000_0000 |
| **BODCR** | GCR_BA+0x18 | R/W | BOD控制 | 0x0000_008X |
| **TEMPCR** | GCR_BA+0x1C | R/W | 温度传感器 | 0x0000_0000 |
| **PORCR** | GCR_BA+0x24 | R/W | 上电复位控制 | 0x0000_00XX |
| **P0_MFP** | GCR_BA+0x30 | R/W | P0 引脚多功能配置 | 0x0000_0000 |
| **P1_MFP** | GCR_BA+0x34 | R/W | P1引脚多功能配置 | 0x0000_0000 |
| **P2_MFP** | GCR_BA+0x38 | R/W | P2引脚多功能配置 | 0x0000_0000 |
| **P3_MFP** | GCR_BA+0x3C | R/W | P3引脚多功能配置 | 0x0000_0000 |
| **P4_MFP** | GCR_BA+0x40 | R/W | P4引脚多功能配置 | 0x0000_0000 |
| **P5_MFP** | GCR_BA+0x44 | R/W | P5引脚多功能配置 | 0x0000_0000 |
| **P6_MFP** | GCR_BA+0x48 | R/W | P6引脚多功能配置 | 0x0000_0000 |
| **P7_MFP** | GCR_BA+0x4C | R/W | P7引脚多功能配置 | 0x0000_0000 |
| **P8_MFP** | GCR_BA+0x50 | R/W | P8引脚多功能配置 | 0x0000_0000 |
| **P9_MFP** | GCR_BA+0x54 | R/W | P9引脚多功能配置 | 0x0000_0000 |
| **PA_MFP** | GCR_BA+0x58 | R/W | PA引脚多功能配置 | 0x0000_0000 |
| **REGWRPROT** | GCR_BA+0x100 | R/W | 寄存器写保护寄存器 | 0x0000_0000 |

### 6.2.6 寄存器描述

**器件ID (PDID)**

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PDID** | GCR_BA+0x00 | R | 器件ID | 0x0035_X0XX[1] |

.

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **PDID** | **产品器件识别码**<br>该寄存器反应设备的器件号码<br>软件可以读取该寄存器来识别所使用的器件 |

## 复位源 (RSTSRC)

复位后读此，可知道是什么原因引起的复位。

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **RSTSRC** | GCR_BA+0x04 | R/W | 系统复位源寄存器 | 0x0000_00XX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **RSTS_CPU** | **Reserved** | **RSTS_SYS** | **RSTS_BOD** | **RSTS_LVR** | **RSTS_WDT** | **RSTS_RESET** | **RSTS_POR** |

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留 |
| [7] | **RSTS_CPU** | **CPU 复位**<br>软件对CPU_RST (IPRSTC1[1]) 写1 造成内核与 Flash (FMC)复位后，此位置1<br>0 = 复位源不是CPU 复位<br>1 = 软件对CPU_RST(IPRSTC1[1]) 写 1.造成的复位<br>**注意：**此位写1清0. |
| [6] | **Reserved** | 保留 |
| [5] | **RSTS_SYS** | **系统复位**<br>0 = 复位源非来至内核.<br>1 =复位源来至内核，软件对 SYSRESETREQ (AIRCR[2] 写 1,造成的复位，此位置1<br>**注意：**此位写1清0. |
| [4] | **RSTS_BOD** | **BOD复位**<br><br>0 = 复位源不是来至 BOD.<br>1 = 复位源来至BOD检到低压.<br>**注意：**此位写1清0. |
| [3] | **RSTS_LVR** | **低压复位**<br>0 = 复位源不是来至 LVR.<br>1 = 复位源来至 LVR 电路.<br>**注意：**此位写1清0.. |

| 位 | 描述 | |
|---|---|---|
| [2] | **RSTS_WDT** | **WDT复位**<br>0 = 复位源不是WDT/WWDT.<br>1 = WDT / WWDT 复位了芯片.<br>**注意1：**此位写1清0.<br>**Note2:** WDT复位时， WTRF(WTCR[2]) 置1，WWDT复位时， WWDTRF(WWDTSR) 置1 |
| [1] | **RSTS_RESET** | **复位脚复位**<br>0 = 复位源不是引脚.<br>1 = 复位源来至复位引脚 .<br>**注意：**此位写1清0.. |
| [0] | **RSTS_POR** | **上电复位**<br>0 = 复位源不是以下二者<br>1 = 芯片重新上电了，或对 CHIP_RST (IPRSTC1[0]) 写1造成的复位.<br>**注意：**此位写1清0.. |

## 外设复位控制 (IPRSTC1)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IPRSTC1** | GCR_BA+0x08 | R/W | 复位控制寄存器1 | 0x0000_0000 |

| 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** |
| | | | **HDIV_RST** | | | **CPU_RST** | **CHIP_RST** |

| 位 | 描述 | |
|---|---|---|
| [31:5] | **Reserved** | 保留 |
| [4] | **HDIV_RST** | **硬件除法器复位(写保护位)**<br>写1硬件除法器复位，再写0，回到正常工作状态。<br>0 = 硬件除法器正常工作.<br>1 = 硬件除法器处在复位状态. |
| [3:2] | **Reserved** | 保留 |
| [1] | **CPU_RST** | **Cortex®-M0 内核 复位 (写保护位)**<br>此位写1 CPU和 Flash (FMC)复位, 两个时钟周期后此位自动回0.<br>0 = CPU 正常操作状态.<br>1 = 复位 CPU. |
| [0] | **CHIP_RST** | **芯片复位 (写保护位)**<br>此位写1，与芯片重新上电一样，两个时钟周期后此位回0.<br>0 = 芯片正常工作状态.<br>1 =复位芯片 |

## 外设复位控制 (IPRSTC2)

写1对应外设异步复位，写0回到工作状态。

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IPRSTC2** | GCR_BA+0x0C | R/W | 复位控制寄存器2 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| | | **OPA_RST** | **EADC_RST** | **ECAP1_RST** | **ECAP0_RST** | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| | **ACMP_RST** | **EPWM1_RST** | **EPWM0_RST** | **BPWM0_RST** | | **UART1_RST** | **UART0_RST** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| | **SPI2_RST** | **SPI1_RST** | **SPI0_RST** | | | | **I2C0_RST** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| | | **TMR3_RST** | **TMR2_RST** | **TMR1_RST** | **TMR0_RST** | **GPIO_RST** | |

| 位 | 描述 | |
|---|---|---|
| [31:26] | **Reserved** | 保留 |
| [29] | **OPA_RST** | 写1让 OPA0，OPA1 在复位状态，写0 OPA0，OPA1回到正常工作状态 |
| [28] | **EADC_RST** | 写1让 EADC在复位状态，写0 EADC回到正常工作状态 |
| [27] | **ECAP1_RST** | 写1让 ECAP1 在复位状态，写0ECAP1回到正常工作状态 |
| [26] | **ECAP0_RST** | 写1让 ECAP0在复位状态，写0 ECAP0回到正常工作状态 |
| [25:23] | **Reserved** | 保留 |
| [23] | **Reserved** | 保留 |
| [22] | **ACMP_RST** | 写1让 ACMP 在复位状态，写0 ACMP 回到正常工作状态 |
| [21] | **EPWM1_RST** | 写1让EPWM1 在复位状态，写0 EPWM1 回到正常工作状态 |
| [20] | **EPWM0_RST** | 写1让 EPWM0 在复位状态，写0 EPWM0 回到正常工作状态 |
| [19] | **BPWM0_RST** | 写1让BPWM0 在复位状态，写0 BPWM0 回到正常工作状态 |
| [18] | **Reserved** | 保留 |
| [17] | **UART1_RST** | 写1让 UART1在复位状态，写0 UART1回到正常工作状态 |
| [16] | **UART0_RST** | 写1让 UART0 在复位状态，写0 UART0 回到正常工作状态 |
| [15] | **Reserved** | 保留 |
| [14] | **SPI2_RST** | 写1让 SPI2 在复位状态，写0 SPI2回到正常工作状态 |
| [13] | **SPI1_RST** | 写1让 SPI1 在复位状态，写0 SPI1回到正常工作状态 |
| [12] | **SPI0_RST** | 写1让SPI0在复位状态，写0 SPI0回到正常工作状态 |

| 位 | 描述 | |
|---|---|---|
| [11:9] | **Reserved** | 保留 |
| [8] | **I2C0_RST** | 写1让 I2C0 在复位状态，写0 I2C0回到正常工作状态 |
| [7:6] | **Reserved** | 保留 |
| [5] | **TMR3_RST** | 写1让 TMR3 在复位状态，写0 TMR3回到正常工作状态 |
| [4] | **TMR2_RST** | 写1让TMR2 在复位状态，写0 TMR2回到正常工作状态 |
| [3] | **TMR1_RST** | 写1让TMR1 在复位状态，写0 TMR1 回到正常工作状态 |
| [2] | **TMR0_RST** | 写1让TMR0在复位状态，写0 TMR0 回到正常工作状态 |
| [1] | **GPIO_RST** | 写1让 GPIO 在复位状态，写0 GPIO 回到正常工作状态 |
| [0] | **Reserved** | 保留 |

## BOD控制 (BODCR)

BODCR 部分位是复位时配置字加载。部分位是受保护的。解锁需要对地址0x5000_0100连续写 “59h”, “16h”, “88h” .具体参考寄存器 REGWRPROT ， 地址GCR_BA+0x100.

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **BODCR** | GCR_BA+0x18 | R/W | BOD控制 | 0x0000_008X |

| 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** |
| **LVR_EN** | **BOD_OUT** | **BOD_LPM** | **BOD_INTF** | **BOD_RSTEN** | **BOD_VL** | | **BOD_EN** |

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留 |
| [7] | **LVR_EN** | **低压复位使能 (写保护位)**<br>预设使能<br>0 = 禁用低压复位<br>1 = 低压复位使能（预设值）. |
| [6] | **BOD_OUT** | **BOD输出值**<br>0 = BOD输出0. 意味着电源电压高于BOD电压或者 BOD_EN 功能被 禁止了.<br>1 = BOD输出 1. BOD使能了且电源电压低于BOD设定值。 |
| [5] | **BOD_LPM** | **BOD低功耗模式 (写保护位)**<br>0 = BOD 正常操作模式(预设).<br>1 =使能BOD 低功耗模式<br>**注意1:** BOD正常常模式耗 电约100uA，低功耗模式耗 电约10uA，但低功耗 模式响应会变慢。 |
| [4] | **BOD_INTF** | **BOD中断标志位**<br>0 = 未检到电压低于 BOD设定值.<br>1 = 检到电源电压低于BOD设定值，又高于设定值，此位置1，若使能了中断，会产生中断。.<br>**Note:** 写1清0. |

| 位 | 描述 | |
|---|---|---|
| [3] | **BOD_RSTEN** | **BOD复位使能位 (写保护位)**<br>0 = BOD中断功能.<br>如果BOD功能使能了 (BOD_EN =1), BOD_OUT=1时会申请中断. BOD 中断会保持到 BOD_EN 写成 0 时，是否执行中断代码，还要看中断向量 的配置 NVIC BOD<br>1 = BOD复位功能<br>若使能了BOD功能 (BOD_EN =1), BOD_OUT =1时， 会让芯片保持复位直到电压高于BOD 配置电压值。<br>**注意:** 此位上电后由配置字 CBRST (Config0[20])加载 |
| [2:1] | **BOD_VL** | **BOD电压选 择(写保护位)**<br>上电时由配置字节的 BOV (Config0[22:21]) 加载<br>00 = BOD电压 2.2V.<br>01 = BOD电压2.7V.<br>10 = BOD电压 3.7V.<br>11 = BOD电压 4.4V. |
| [0] | **BOD_EN** | **BOD使能位 (写保护位)**<br>上电时由 CBODEN (Config0[23]) 加载<br>0 = BOD功能禁用.<br>1 = BOD功能使能 |

## 温度传感器控制 (TEMPCR)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **TEMPCR** | GCR_BA+0x1C | R/W | 温度传感器 | 0x0000_0000 |

| 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** |
| | | | | | | | **VTEMP_EN** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **VTEMP_EN** | **温感使能位**<br>0 = 禁用（预设值）<br>1 = 使能<br>温感使能后，可以由ADC测出输出值，计算出温度值 |

## 上电复位控制 (PORCR)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PORCR** | GCR_BA+0x24 | R/W | 上电复位控制 | 0x0000_00XX |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留 |
| [15:0] | **POR_DIS_CODE** | **上电复位使能 (写保护位)**<br>上电后POR产生一复位信号，电源噪声可能会让POR产生多次复位信号，用户可以写0x5AA5关掉POR复位功能，与其它值或下面这些复位芯片后，POR功能会再次使能:<br>复位引脚复位, WDT复位, WWDT复位，LVR 复位, BOD 复位, ICE 复位指令和软件复位指令。 |

## P0 多功能控制 (P0_MFP)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P0_MFP** | GCR_BA+0x30 | R/W | P0 引脚多功能配置 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **P0_TYPE** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| | | | | **P0_ALT** | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P0_MFP** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [m+16]<br>m=0,1..7 | **P0_TYPE[m]** | **端口0斯密特触发输入使能位**<br>0 = 端口 0 位m禁止斯密特<br>1 =端口0 位m使能斯密特功能 |
| [15:12] | **Reserved** | 保留 |
| [11] | **P0_ALT[3]** | 见 P0_MFP[3]. |
| [10] | **P0_ALT[2]** | 见 P0_MFP[2]. |
| [9] | **P0_ALT[1]** | 见 P0_MFP[1]. |
| [8] | **P0_ALT[0]** | 见 P0_MFP[0]. |
| [7] | **P0_MFP[7]** | **P0.7 多功能选择**<br>0 = GPIO 功能<br>1 = STADC 功能. |
| [6] | **P0_MFP[6]** | **P0.6 多功能选择**<br>0 = GPIO 功能.<br>1 = EPWM0_BRAKE1 功能. |
| [5] | **P0_MFP[5]** | **P0.5 多功能选择**<br>0 = GPIO 功能.<br>1 = EPWM0_CH5功能 |
| [4] | **P0_MFP[4]** | **P0.4多功能选择**<br>0 = GPIO 功能<br>1 = EPWM0_CH4功能. |

| 位 | 描述 | |
|---|---|---|
| [3] | **P0_MFP[3]** | **P0.3多功能选择**<br>(P0_ALT[3], P0_MFP[3]) 的值选配功能如下.<br>(0, 0) = GPIO 功能.<br>(0, 1) = EPWM0_CH3 功能<br>(1, 1) = STADC 功能 |
| [2] | **P0_MFP[2]** | **P0.2 多功能选择**<br>(P0_ALT[2], P0_MFP[2]) 的值选配功能如下.<br>(0, 0) = GPIO P0.2 功能<br>(0, 1) = EPWM0_CH2 功能<br>(1, 1) = ECAP1_IC2 功能 |
| [1] | **P0_MFP[1]** | **P0.1多功能选择**<br>(P0_ALT[2], P0_MFP[2]) 的值选配功能如下.<br>(0, 0) = GPIO 功能.<br>(0, 1) = EPWM0_CH1 功能.<br>(1, 1) = CAP1_IC1 功能 |
| [0] | **P0_MFP[0]** | **P0.0 多功能选择**<br>(P0_ALT[0], P0_MFP[0]) 的值选配功能如下.<br>(0, 0) = GPIO 功能<br>(0, 1) = EPWM0_CH0 功能.<br>(1, 1) = ECAP1_IC0 功能. |

## P1多功能控制(P1_MFP)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P1_MFP** | GCR_BA+0x34 | R/W | P1 引脚多功能配置 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **P1_TYPE** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P1_MFP** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [m+16]<br>m=0,1..7 | **P1_TYPE[m]** | **端口1斯密特触发输入使能位**<br>0 =端口1 位m 禁斯密特功能<br>1 =端口1 位 m 使能斯密特功能. |
| [15:8] | **Reserved** | 保留 |
| [7] | **P1_MFP[7]** | **P1.7 多功能选择**<br>0 = P1.7 GPIO 功能.<br>1 = EPWM1_BRAKE0 功能. |
| [6] | **P1_MFP[6]** | **P1.6 多功能选择**<br>0 = P1.6 GPIO 功能.<br>1 = EPWM0_BRAKE0 功能 |
| [5] | **P1_MFP[5]** | **P1.5 多功能选择**<br>0 = P1.5 GPIO 功能.<br>1 = EPWM1_CH5 功能. |
| [4] | **P1_MFP[4]** | **P1.4 多功能选择**<br>0 = P1.4 GPIO 功能.<br>1 = EPWM1_CH4 功能. |
| [3] | **P1_MFP[3]** | **P1.3 多功能选择**<br>0 = P1.3 GPIO 功能.<br>1 = EPWM1_CH3 功能. |
| [2] | **P1_MFP[2]** | **P1.2 多功能选择**<br>0 = P1.2 GPIO 功能.<br>1 = EPWM1_CH2 功能 |

| 位 | 描述 | |
|---|---|---|
| [1] | **P1_MFP[1]** | **P1.1 多功能选择**<br>0 = P1.1 GPIO 功能.<br>1 = EPWM1_CH1 功能. |
| [0] | **P1_MFP[0]** | **P1.0 多功能选择**<br>0 = P1.0 GPIO 功能.<br>1 = EPWM1_CH0 功能 |

## P2多功能控制(P2_MFP)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P2_MFP** | GCR_BA+0x38 | R/W | P2 引脚多功能配置 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **P2_TYPE** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **P2_ALT[7]** | **P2_ALT[6]** | **P2_ALT[5]** | **P2_ALT[4]** | **Reserved** | | | **P2_ALT[0]** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P2_MFP[7]** | **P2_MFP[6]** | **P2_MFP[5]** | **P2_MFP[4]** | **P2_MFP[3]** | **P2_MFP[2]** | **P2_MFP[1]** | **P2_MFP[0]** |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **P2_TYPE** | **端口2斯密特触发输入使能位**<br>0 =端口2 位m 禁斯密特功能.<br>1 =端口2 位m 使能斯密特功能.<br>m=0,1..7. |
| [15] | **P2_ALT[7]** | 见 P2_MFP[7]. |
| [14] | **P2_ALT[6]** | 见 P2_MFP[6]. |
| [13] | **P2_ALT[5]** | 见 P2_MFP[5]. |
| [12] | **P2_ALT[4]** | 见 P2_MFP[4]. |
| [11:9] | **Reserved** | 保留 |
| [8] | **P2_ALT[0]** | 见 P2_MFP[0]. |
| [7] | **P2_MFP[7]** | **P2.7多功能选择**<br>(P2_ALT[7], P2_MFP[7]) 的值功能定义如下<br>(0, 0) = P2.7 GPIO功能.<br>(1, 0) = SPI0_CLK 功能.<br>(1, 1) = UAT1_nRTS 功能 |
| [6] | **P2_MFP[6]** | **P2.6多功能选择**<br>(P2_ALT[6], P2_MFP[6]) 的值功能定义如下<br>(0, 0) = P2.6 GPIO功能.<br>(0, 1) = 保留<br>(1, 0) = SPI0_SS 功能.<br>(1, 1) = UART1_nCTS 功能 |

| 位 | 描述 | |
|---|---|---|
| [5] | **P2_MFP[5]** | **P2.5 多功能选择**<br>(P2_ALT[5], P2_MFP[5]) 的值功能定义如下<br>(0, 0) = P2.5 GPIO功能.<br>(0, 1) =保留<br>(1, 1) =保留 |
| [4] | **P2_MFP[4]** | **P2.4 多功能选择**<br>(P2_ALT[4], P2_MFP[4]) 的值功能定义如下<br>(0, 0) = P2.4 GPIO功能.<br>(0, 1) =保留.<br>(1, 1) =保留 |
| [3] | **P2_MFP[3]** | **P2.3 多功能选择**<br>0 = P2.3 GPIO功能.<br>1 = ECAP0_IC0 功能. |
| [2] | **P2_MFP[2]** | **P2.2 多功能选择**<br>0 = P2.2 GPIO功能.<br>1 = ECAP0_IC1 功能. |
| [1] | **P2_MFP[1]** | **P2.1 多功能选择**<br>0 = P2.1 GPIO功能<br>1 = ECAP0_IC2 功能. |
| [0] | **P2_MFP[0]** | **P2.0 多功能选择**<br>(P2_ALT[0], P2_MFP[0]) 的值功能定义如下.<br>(0, 0) = P2.0 GPIO功能<br>(0, 1) = SPI2_MOSI 功能.<br>(1, 0) = ACMP2_O 功能 |

## P3多功能控制(P3_MFP)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P3_MFP** | GCR_BA+0x3C | R/W | P3引脚多功能配置 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **P3_TYPE** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | **P3_ALT[5]** | **P3_ALT[4]** | **Reserved** | | **P3_ALT[1]** | **P3_ALT[0]** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P3_MFP[7]** | **P3_MFP[6]** | **P3_MFP[5]** | **P3_MFP[4]** | **P3_MFP[3]** | **P3_MFP[2]** | **P3_MFP[1]** | **P3_MFP[0]** |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **P3_TYPE** | **端口3斯密特触发输入使能位**<br>0 =端口3 位m 禁斯密特功能<br>1 =端口3 位m 使能斯密特功能.<br>m=0,1..7. |
| [15:14] | **Reserved** | 保留 |
| [13] | **P3_ALT[5]** | 见 P3_MFP[5]. |
| [12] | **P3_ALT[4]** | 见 P3_MFP[4]. |
| [11:10] | **Reserved** | 保留 |
| [9] | **P3_ALT[1]** | 见 P3_MFP[1]. |
| [8] | **P3_ALT[0]** | 见 P3_MFP[0]. |
| [7] | **P3_MFP[7]** | **P3.7 多功能选择**<br>应写0. |
| [6] | **P3_MFP[6]** | **P3.6 多功能选择**<br>应写0 |
| [5] | **P3_MFP[5]** | **P3.5 多功能选择**<br>(P3_ALT[5], P3_MFP[5]) 的值功能定义如下.<br>(0, 0) = GPIO 功能 .<br>(0, 1) = TM1 功能.<br>(1, 0) = I2C0_SCL 功能. |

| 位 | 描述 | |
|---|---|---|
| [4] | **P3_MFP[4]** | **P3.4 多功能选择**<br>P3_ALT[4], P3_MFP[4]) 的值功能定义如下<br>(0, 0) = GPIO 功能<br>(0, 1) = TM0 功能.<br>(1, 0) = I2C0_SDA 功能. |
| [3] | **P3_MFP[3]** | **P3.3 多功能选择**<br>0 = GPIO 功能.<br>1 = INT1 功能. |
| [2] | **P3_MFP[2]** | **P3.2 多功能选择**<br>0 = GPIO 功能.<br>1 = INT0 功能 |
| [1] | **P3_MFP[1]** | **P3.1多功能选择**<br>(P3_ALT[1], P3_MFP[1]) 的值功能定义如下<br>(0, 0) = GPIO 功能.<br>(0, 1) = UART0_TXD 功能.<br>(1, 0) = ACMP0_O 功能. |
| [0] | **P3_MFP[0]** | **P3.0 多功能选择**<br>(P3_ALT[0], P3_MFP[0]) 的值功能定义如下.<br>(0, 0) = GPIO 功能.<br>(0, 1) = UART0_RXD 功能.<br>(1, 0) = CLKO 功能 |

## P4多功能控制(P4_MFP)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P4_MFP** | GCR_BA+0x40 | R/W | P4 引脚多功能配置 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **P4_TYPE** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | **P4_ALT[6]** | **Reserved** | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P4_MFP[7]** | **P4_MFP[6]** | **P4_MFP[5]** | **P4_MFP[4]** | **Reserved** | **P4_MFP[2]** | **P4_MFP[1]** | **P4_MFP[0]** |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **P4_TYPE** | **端口4斯密特触发输入使能位**<br>0 =端口4 位m 禁斯密特功能<br>1 =端口4 位m使能斯密特功能.<br>m=0,1..7. |
| [15] | **Reserved** | 保留 |
| [14] | **P4_ALT[6]** | 见 P4_MFP[6]. |
| [13:8] | **Reserved** | 保留 |
| [7] | **P4_MFP[7]** | **P4.7 多功能选择**<br>0 = GPIO 功能.<br>1 = TM3 功能 |
| [6] | **P4_MFP[6]** | **P4.6 多功能选择**<br>(P4_ALT[6], P4_MFP[6]) 的值功能定义如下.<br>(0, 0) = GPIO 功能<br>(0, 1) = TM2功能.<br>(1, 0) = 要留 |
| [5] | **P4_MFP[5]** | **P4.5 多功能选择**<br>应写0， GPIO 功能 |
| [4] | **P4_MFP[4]** | **P4.4 多功能选择**<br>应写0， GPIO 功能 |
| [3] | **Reserved** | 保留 |

| 位 | 描述 | |
|---|---|---|
| [2] | **P4_MFP[2]** | **P4.2 多功能选择**<br>0 = GPIO 功能.<br>1 = ECAP1_IC2 功能 |
| [1] | **P4_MFP[1]** | **P4.1 多功能选择**<br>0 = GPIO 功能.<br>1 = ECAP1_IC1 功能. |
| [0] | **P4_MFP[0]** | **P4.0 多功能选择**<br>0 = GPIO 功能.<br>1 = ECAP1_IC0 功能. |

## P5多功能控制(P5_MFP)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P5_MFP** | GCR_BA+0x44 | R/W | P5引脚多功能配置 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **P5_TYPE** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | **P5_ALT** | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P5_MFP** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **P5_TYPE[m]** | **端口5斯密特触发输入使能位**<br>0 =端口5 位m禁斯密特功能.<br>1 =端口5 位m使能斯密特功能.<br>m=0,1..7. |
| [15:11] | **Reserved** | 保留 |
| [10] | **P5_ALT[2]** | 见 P5_MFP[2]. |
| [9] | **P5_ALT[1]** | 见 P5_MFP[1]. |
| [8] | **P5_ALT[0]** | 见 P5_MFP[0]. |
| [7] | **P5_MFP[7]** | **P5.7 多功能选择**<br>0 = GPIO 功能.<br>1 = BPWM0_CH1 功能. |
| [6] | **P5_MFP[6]** | **P5.6 多功能选择**<br>0 = GPIO 功能.<br>1 = BPWM0_CH0 功能. |
| [5] | **P5_MFP[5]** | **P5.5 多功能选择**<br>0 = GPIO 功能<br>1 = CLKO 功能 |
| [4] | **P5_MFP[4]** | **P5.4 多功能选择**<br>0 = GPIO 功能.<br>1 = SPI2_SS 功能. |

| 位 | 描述 | |
|---|---|---|
| [3] | **P5_MFP[3]** | **P5.3 多功能选择**<br>0 = GPIO 功能.<br>1 = SPI2_CLK 功能. |
| [2] | **P5_MFP[2]** | **P5.2 多功能选择**<br>(P5_ALT[2], P5_MFP[2]) 的值功能定义如下.<br>(0, 0) = GPIO 功能<br>(0, 1) = SPI2_MISO 功能.<br>(1, 0) = ACMP1_O 功能. |
| [1] | **P5_MFP[1]** | **P5.1 多功能选择**<br>(P5_ALT[1], P5_MFP[1]) 的值功能定义如下.<br>(0, 0) = GPIO 功能<br>(0, 1) = SPI0_MISO 功能.<br>(1, 0) = UART0_nCTS 功能.<br>(1, 1) = I2C0_SDA 功能. |
| [0] | **P5_MFP[0]** | **P5.0 多功能选择**<br>(P5_ALT[0], P5_MFP[0]) 的值功能定义如下<br>(0, 0) = GPIO 功能 .<br>(0, 1) = SPI0_MOSI 功能.<br>(1, 0) = UART0_nRTS 功能.<br>(1, 1) = I2C0_SCL 功能. |

## P6多功能控制(P6_MFP)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P6_MFP** | GCR_BA+0x48 | R/W | P6引脚多功能配置 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **P6_TYPE** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P6_MFP[7]** | **P6_MFP[6]** | **P6_MFP[5]** | **P6_MFP[4]** | **P6_MFP[3]** | **P6_MFP[2]** | **P6_MFP[1]** | **P6_MFP[0]** |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **P6_TYPE** | **端口6斯密特触发输入使能位**<br>0 =端口6 位m禁斯密特功能.<br>1 =端口6 位m使能斯密特功能<br>m=0,1..7. |
| [15:8] | **Reserved** | 保留 |
| [7] | **P6_MFP[7]** | **P6.7 多功能选择**<br>0 = GPIO 功能<br>1 = EADC0_CH7 功能 |
| [6] | **P6_MFP[6]** | **P6.6 多功能选择**<br>0 = GPIO 功能.<br>1 = EADC0_CH6 功能. |
| [5] | **P6_MFP[5]** | **P6.5 多功能选择**<br>0 = GPIO 功能.<br>1 = EADC0_CH5 或 ACMP1_P 功能 |
| [4] | **P6_MFP[4]** | **P6.4 多功能选择**<br>0 = GPIO 功能.<br>1 = EADC0_CH4 或 ACMP1_N功能. |
| [3] | **P6_MFP[3]** | **P6.3 多功能选择**<br>0 = GPIO 功能.<br>1 = EADC0_CH3 功能 |

| 位 | 描述 | |
|---|---|---|
| [2] | **P6_MFP[2]** | **P6.2 多功能选择**<br>0 = GPIO功能.<br>1 = EADC0_CH2 功能 |
| [1] | **P6_MFP[1]** | **P6.1 多功能选择**<br>0 = GPIO 功能<br>1 = EADC0_CH1 功能 |
| [0] | **P6_MFP[0]** | **P6.0 多功能选择**<br>0 = GPIO 功能<br>1 = EADC0_CH0 功能 |

## P7多功能控制(P7_MFP)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P7_MFP** | GCR_BA+0x4C | R/W | P7 引脚多功能配置 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **P7_TYPE** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P7_MFP[7]** | **P7_MFP[6]** | **P7_MFP[5]** | **P7_MFP[4]** | **P7_MFP[3]** | **P7_MFP[2]** | **P7_MFP[1]** | **P7_MFP[0]** |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **P7_TYPE** | **端口7斯密特触发输入使能位**<br>0 =端口7 位m禁斯密特功能<br>1 =端口7 位m使能斯密特功能<br>m=0,1..7. |
| [15:8] | **Reserved** | 保留 |
| [7] | **P7_MFP[7]** | **P7.7 多功能选择**<br>0 = GPIO 功能 .<br>1 = EADC1_CH7 功能 |
| [6] | **P7_MFP[6]** | **P7.6 多功能选择**<br>0 = GPIO 功能<br>1 = EADC1_CH6 功能. |
| [5] | **P7_MFP[5]** | **P7.5 多功能选择**<br>0 = GPIO 功能<br>1 = EADC1_CH5 或 ACMP2_P 功能. |
| [4] | **P7_MFP[4]** | **P7.4 多功能选择**<br>0 = GPIO 功能.<br>1 = EADC1_CH4 或 ACMP2_N 功能. |
| [3] | **P7_MFP[3]** | **P7.3 多功能选择**<br>0 = GPIO 功能.<br>1 = EADC1_CH3 功能. |

| 位 | 描述 | |
|---|---|---|
| [2] | **P7_MFP[2]** | **P7.2 多功能选择**<br>0 = GPIO P7.2 功能.<br>1 = EADC1_CH2 功能. |
| [1] | **P7_MFP[1]** | **P7.1 多功能选择**<br>0 = GPIO 功能.<br>1 = EADC1_CH1 功能. |
| [0] | **P7_MFP[0]** | **P7.0多功能选择**<br>0 = GPIO P7.0 功能<br>1 = EADC1_CH0 功能 |

## P8多功能控制(P8_MFP)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P8_MFP** | GCR_BA+0x50 | R/W | P8 引脚多功能配置 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **P8_TYPE** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P8_MFP[7]** | **Reserved** | | **P8_MFP[4]** | **P8_MFP[3]** | **P8_MFP[2]** | **P8_MFP[1]** | **P8_MFP[0]** |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **P8_TYPE[m]** | **端口8斯密特触发输入使能位**<br>0 =端口8 位m禁斯密特功能.<br>1 =端口8 位m使能斯密特功能.<br>m=0,1..7. |
| [15:8] | **Reserved** | 保留 |
| [7] | **P8_MFP[7]** | **P8.7 多功能选择**<br>0 = GPIO 功能.<br>1 = ACMP0_O 功能. |
| [6:5] | **Reserved** | 保留 |
| [4] | **P8_MFP[4]** | **P8.4 多功能选择**<br>0 = GPIO 功能.<br>1 = ACMP0_P 功能 |
| [3] | **P8_MFP[3]** | **P8.3 多功能选择**<br>0 = GPIO 功能.<br>1 = ACMP0_N 功能. |
| [2] | **P8_MFP[2]** | **P8.2 多功能选择**<br>0 = GPIO 功能.<br>1 = OP0_O 功能. |
| [1] | **P8_MFP[1]** | **P8.1 多功能选择**<br>0 = GPIO 功能.<br>1 = OP0_N 功能. |

| 位 | 描述 | |
|---|---|---|
| [0] | **P8_MFP[0]** | **P8.0 多功能选择**<br>0 = GPIO 功能.<br>1 = OP0_P 功能. |

## P9多功能控制(P9_MFP)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P9_MFP** | GCR_BA+0x54 | R/W | P9 引脚多功能配置 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **P9_TYPE** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P9_MFP[7]** | **P9_MFP[6]** | **P9_MFP[5]** | **P9_MFP[4]** | **P9_MFP[3]** | **P9_MFP[2]** | **P9_MFP[1]** | **P9_MFP[0]** |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **P9_TYPE** | **端口9斯密特触发输入使能位**<br>0 =端口9 位m禁斯密特功能.<br>1 =端口9 位m使能斯密特功能.<br>m=0,1..7. |
| [15:8] | **Reserved** | 保留 |
| [7] | **P9_MFP[7]** | **P9.7 多功能选择**<br>0 = GPIO 功能.<br>1 = SPI1_SS 功能. |
| [6] | **P9_MFP[6]** | **P9.6 多功能选择**<br>0 = GPIO 功能.<br>1 = SPI1_MOSI 功能. |
| [5] | **P9_MFP[5]** | **P9.5 多功能选择**<br>0 = GPIO 功能.<br>1 = SPI1_MISO 功能. |
| [4] | **P9_MFP[4]** | **P9.4 多功能选择**<br>0 = GPIO 功能.<br>1 = SPI1_CLK 功能. |
| [3] | **P9_MFP[3]** | **P9.3 多功能选择**<br>0 = GPIO 功能.<br>1 = EPWM1_BRAKE1 功能. |

| 位 | 描述 | |
|---|---|---|
| [2] | **P9_MFP[2]** | **P9.2多功能选择**<br>0 = GPIO 功能.<br>1 = The OP1_P 功能. |
| [1] | **P9_MFP[1]** | **P9.1 多功能选择**<br>0 = GPIO 功能<br>1 = OP1_N 功能. |
| [0] | **P9_MFP[0]** | **P9.0 多功能选择**<br>0 = GPIO 功能<br>1 = OP1_O 功能. |

## PA多功能控制(PA_MFP)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PA_MFP** | GCR_BA+0x58 | R/W | PA 引脚多功能配置 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | **PA_TYPE** | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | **PA_ALT[1]** | **PA_ALT[0]** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | **PA_MFP[1]** | **PA_MFP[0]** |

| 位 | 描述 | |
|---|---|---|
| [31:18] | **Reserved** | 保留 |
| [23:16] | **PA_TYPE** | **端口A斯密特触发输入使能位**<br>0 =端口A 位m禁斯密特功能.<br>1 =端口A 位m使能斯密特功能<br>m=0,1. |
| [15:10] | **Reserved** | 保留 |
| [9] | **PA_ALT[1]** | **PA.1 多功能选择**<br>见 PA_MFP[1]. |
| [8] | **PA_ALT[0]** | **PA.0 多功能选择**<br>见 PA_MFP[0]. |
| [7:2] | **Reserved** | 保留 |
| [1] | **PA_MFP[1]** | **PA.1 多功能选择**<br>PA_ALT[1], PA_MFP[1]) 的值功能定义如下<br>(0, 0) = GPIO 功能<br>(0, 1) = UART1_RXD 功能<br>(1, 1) = I2C0_SCL 功能 |
| [0] | **PA_MFP[0]** | **PA.0 多功能选择**<br>(PA_ALT[0], PA_MFP[0]) 的值功能定义如下<br>(0, 0) = GPIO 功能<br>(0, 1) = UART1_TXD 功能.<br>(1, 1) = I2C0_SDA 功能 |

## 寄存器写保护控制 (REGWRPROT)

为防止某些寄存器受干扰数据变掉，复位后这些寄存器是“上锁”不可改写状态。用户若要改写这些寄存器的值，必须往地址在0x5000_0100的寄存器 REGWRPROT 中依次写入“59h”, “16h” “88h”解锁。数据不对，写入顺序不对，或写入过程被打断，解锁都会失败。

读这个寄存器时，位0若值为1，表示解锁状态，值为0表示锁定状态。

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **REGWRPROT** | GCR_BA+0x100 | R/W | 寄存器写保护寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留 |
| [7:1] | **REGWRPROT** | **寄存器写保护代码 (只写)**<br>依次写入 “59h”, “16h”, “88h”被 保护的寄存器将解锁，可以写入值。<br>写入其它数据将变为锁定状态. |
| [0] | **REGPROTDIS** | **保护状态位 (只读)**<br>0 = 保护状态，对被保护的寄存器的写，将被忽略。<br>1 = 解锁状态，可以写被保护的寄存器。 |

### 6.2.7 系统定时器 (SysTick)

Cortex®-M0 内核提供一 24位下计数系统定时器，当前值 (SYST_CVR) 下计数到 0后,装入 (SYST_RVR)的值，继续下计数，且置1状态位 COUNTFLAG，状态位读清0。

更详细信息见 “ARM® Cortex®-M0 Technical Reference Manual” 和 “ARM® v6-M Architecture Reference Manual”.

#### 6.2.7.1 SysTick 寄存器表

**R**:只读, **W**: 只写, **R/W**: 可读写

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **SYST 基地址:**<br>**SYST_BA = 0xE000_E010** | | | | |
| **SYST_CSR** | SYST_BA+0x00 | R/W | SysTick 控制和状态字 | 0x0000_0000 |
| **SYST_RVR** | SYST_BA+0x04 | R/W | SysTick 周期值 | 0xXXXX_XXXX |
| **SYST_CVR** | SYST_BA+0x08 | R/W | SysTick 当前值 | 0xXXXX_XXXX |

#### 6.2.7.2 SysTick 寄存器描述

## SysTick 控制和状态 (SYST_CSR)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **SYST_CSR** | SYST_BA+0x00 | R/W | SysTick 控制和状态寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | **COUNTFLAG** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | **CLKSRC** | **TICKINT** | **ENABLE** |

| 位 | 描述 | |
|---|---|---|
| [31:17] | **Reserved** | 保留 |
| [16] | **COUNTFLAG** | **System Tick 计数标志**<br>计数从1到0时，COUNTFLAG 位置1<br>此位读清0，写一个新值到 (SYST_CVR)也会清0此位。 |
| [15:3] | **Reserved** | 保留 |
| [2] | **CLKSRC** | **System Tick 时钟源选择**<br>0 = 由 STCLK_S (CLKSEL0[5:3])确定.<br>1 = 以内核时钟计数. |
| [1] | **TICKINT** | **System Tick 中断使能**<br>0 = 禁中断<br>1 = 计数到0产生中断。往当前值寄存器写0不会产生中断申请 |
| [0] | **ENABLE** | **System Tick 计数使能**<br>0 = 禁计数.<br>1 = 使能计数 |

## SysTick 重载值 （SYST_RVR）

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **SYST_RVR** | SYST_BA+0x04 | R/W | SysTick 重装值 | 0xXXXX_XXXX |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:0] | **RELOAD** | **System Tick 重装值**<br>计数到0时，此寄存器中的值，重新装入到定时器，并开始下计数. |

## SysTick 当前值（SYST_CVR）

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **SYST_CVR** | SYST_BA+0x08 | R/W | SysTick 当前计数值 | 0xXXXX_XXXX |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:0] | **CURRENT** | **System Tick 当前值**<br>读：计数器当前值。<br>写：写任何值都清0此寄存器 |

### 6.2.8 可嵌套中断向量控制 (NVIC)

Cortex®-M0 提供以下功能的中断控制:

- 嵌套中断向量
- 处理器状态保存和恢复
- 决定或减短中断响应时间

进入中断即进入 “Handler”模式. 支持32 (IRQ[31:0])个中断， 4 个优先级。低优先级中断代码执行时，若有更高优先级中断发生，会立即转去执行。

发生中断时，寄存器“PC, PSR, LR, R0~R3, R12”由硬件入栈，中断代码首地址从向量表中取出，去执行中断代码，中断代码执行完全后，上述8个寄存器会由硬件出栈，然后返回原来被中断代码处，继续执行。

详细信息见 “ARM® Cortex®-M0 Technical Reference Manual” 和 “ARM® v6-M Architecture Reference Manual”.

#### 6.2.8.1 异常模式和系统中断表

M0519 支持的异常见表6-2，许多中断优先级可配置，优先级0最高，3最低。复位后优先级可配置的中断，优先级都会0 。

| 异常 | 向量号 | 优先级 |
|---|---|---|
| Reset | 1 | -3 |
| NMI | 2 | -2 |
| Hard Fault | 3 | -1 |
| Reserved | 4 ~ 10 | |
| SVCall | 11 | 可配置 |
| Reserved | 12 ~ 13 | |
| PendSV | 14 | 可配置 |
| SysTick | 15 | 可配置 |
| Interrupt (IRQ0 ~ IRQ31) | 16 ~ 47 | 可配置 |

表 6-2 异常模式

| 异常号 | 向量地址 | 中断号 | 中断名 | 外设 | 描述 | 可唤醒CPU |
|---|---|---|---|---|---|---|
| 1 ~ 15 | | - | - | - | 系统异常 | - |
| 16 | 0x40 | 0 | **BOD_INT** | Brown-out | BOD中断 | 是 |
| 17 | 0x44 | 1 | **WDT_INT** | WDT | WDT中断 | 是 |
| 18 | 0x48 | 2 | **EINT0_INT** | P3.2 | 外中断引脚 P3.2 | 是 |
| 19 | 0x4C | 3 | **EINT1_INT** | P3.3 | 外中断引脚 P3.3 | 是 |
| 20 | 0x50 | 4 | **GPG0_INT** | P0~P4 除 P3.2 和 P3.3 外 | GPIOG0 (P0~P4)中断,不包含 P3.2 和 P3.3 | 是 |
| 21 | 0x54 | 5 | **GPG1_INT** | P5~PA | GPIOG1 (P5~PA)中断 | 是 |
| 22 | 0x58 | 6 | **BPWM0_INT** | BPWM0 | BPWM0 中断 | 不 |
| 23 | 0x5C | 7 | **EADC0_INT** | EADC0 | EADC0中断 | 不 |
| 24 | 0x60 | 8 | **TMR0_INT** | TMR0 | Timer 0中断 | 不 |
| 25 | 0x64 | 9 | **TMR1_INT** | TMR1 | Timer 1中断 | 不 |
| 26 | 0x68 | 10 | **TMR2_INT** | TMR2 | Timer 2中断 | 不 |
| 27 | 0x6C | 11 | **TMR3_INT** | TMR3 | Timer 3中断 | 不 |
| 28 | 0x70 | 12 | **UART0_INT** | UART0 | UART0中断 | 是 |
| 29 | 0x74 | 13 | **UART1_INT** | UART1 | UART1中断 | 是 |
| 30 | 0x78 | 14 | **SPI0_INT** | SPI0 | SPI0中断 | 不 |

| 异常号 | 向量地址 | 中断号 | 中断名 | 外设 | 描述 | 可唤醒CPU |
|---|---|---|---|---|---|---|
| 31 | 0x7C | 15 | **SPI1_INT** | SPI1 | SPI1中断 | 不 |
| 32 | 0x80 | 16 | **SPI2_INT** | SPI2 | SPI2中断 | 不 |
| 33 | 0x84 | 17 | **Reserved** | | | - |
| 33 | 0x84 | 17 | **Reserved** | | | 不 |
| 34 | 0x88 | 18 | **I2C0_INT** | $I^2C0$ | $I^2C0$ 中断 | 是 |
| 35 | 0x8C | 19 | **CKD_INT** | CKD | CKD 中断 | 不 |
| 36 | 0x90 | 20 | **Reserved** | | | - |
| 36 | 0x90 | 20 | **Reserved** | | | - |
| 37 | 0x94 | 21 | **EPWM0_INT** | EPWM0 | EPWM0 中断 | 不 |
| 38 | 0x98 | 22 | **EPWM1_INT** | EPWM1 | EPWM1中断 | 不 |
| 39 | 0x9C | 23 | **ECAP0_INT** | ECAP0 | ECAP0中断 | 不 |
| 40 | 0xA0 | 24 | **ECAP1_INT** | ECAP1 | ECAP1中断 | 不 |
| 41 | 0xA4 | 25 | **ACMP_INT** | ACMP | ACAMP 0 或1, 或运放数字输出中断 | 是 (仅 ACMP) |
| 42 | 0xA8 | 26 | **Reserved** | | | - |
| 43 | 0xAC | 27 | **Reserved** | | | - |
| 42 | 0xA8 | 26 | **Reserved** | | | - |
| 43 | 0xAC | 27 | **Reserved** | | | - |
| 44 | 0xB0 | 28 | **PWRWU_INT** | CLKC | 唤醒中断 | - |
| 45 | 0xB4 | 29 | **EADC1_INT** | EADC1 | EADC1 中断 | 不 |
| 46 | 0xB8 | 30 | **EADC2_INT** | EADC2 | EADC2 中断 | 不 |
| 47 | 0xBC | 31 | **EADC3_INT** | EADC3 | EADC3 中断 | 不 |

表 6-3 系统中断向量表

#### 6.2.8.2 向量表

发生使能的中断时，处理器从向量表中取得中断代码首地址， ARMv6-M 版本，向量表固定在地址 0x00000000。

| 向量表地址 | 描述 |
|---|---|
| 0 | SP_main –主栈指针 |
| 向量号 | 代码首地址 |

表 6-4 向量表

#### 6.2.8.3 操作描述

NVIC 中断可以被禁止或使能。禁止或使能中断采用对不同逻辑地址写1的方式，读这两个地址会得到相同的值：1使能，0禁止。

发生中断时，中断挂起位置1，中断返回、复位会清0挂起位。当然，软件也可以将其清0

#### 6.2.8.4 NVIC控制寄存器

**R**:只读, **W**: 只写, **R/W**: 可读写

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC 基址:**<br>**NVIC_BA = 0xE000_E100** | | | | |
| **NVIC_ISER** | NVIC_BA+0x000 | R/W | IRQ0 ~ IRQ31 使能中断 | 0x0000_0000 |
| **NVIC_ICER** | NVIC_BA+0x080 | R/W | IRQ0 ~ IRQ31 禁止中断 | 0x0000_0000 |
| **NVIC_ISPR** | NVIC_BA+0x100 | R/W | IRQ0 ~ IRQ31 挂起中断，即申请中断 | 0x0000_0000 |
| **NVIC_ICPR** | NVIC_BA+0x180 | R/W | IRQ0 ~ IRQ31 清除挂起，即清除中断申请 | 0x0000_0000 |
| **NVIC_IPR0** | NVIC_BA+0x300 | R/W | IRQ0 ~ IRQ3 中断优先级控制 | 0x0000_0000 |
| **NVIC_IPR1** | NVIC_BA+0x304 | R/W | IRQ4 ~ IRQ7 中断优先级控制 | 0x0000_0000 |
| **NVIC_IPR2** | NVIC_BA+0x308 | R/W | IRQ8 ~ IRQ11 中断优先级控制 | 0x0000_0000 |
| **NVIC_IPR3** | NVIC_BA+0x30C | R/W | IRQ12 ~ IRQ15 中断优先级控制 | 0x0000_0000 |
| **NVIC_IPR4** | NVIC_BA+0x310 | R/W | IRQ16 ~ IRQ19 中断优先级控制 | 0x0000_0000 |
| **NVIC_IPR5** | NVIC_BA+0x314 | R/W | IRQ20 ~ IRQ23 中断优先级控制 | 0x0000_0000 |
| **NVIC_IPR6** | NVIC_BA+0x318 | R/W | IRQ24 ~ IRQ27 中断优先级控制 | 0x0000_0000 |
| **NVIC_IPR7** | NVIC_BA+0x31C | R/W | IRQ28 ~ IRQ31 中断优先级控制 | 0x0000_0000 |

## IRQ0 ~ IRQ31 开中断控制（NVIC_ISER）

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_ISER** | NVIC_BA+0x000 | R/W | IRQ0 ~ IRQ31 使能中断 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **SETENA** | **中断使能寄存器**<br>每位控制一个中断使能，从 IRQ0 到 IRQ31 ( 向量号 16 ~ 47).<br>对每一位的操作如下<br>写:<br>0 = 无效.<br>1 =写1使能中断<br>读:<br>0 = 中断被 禁止.<br>1 = 中断已使能. |

## IRQ0 ~ IRQ31 关中断控制 (NVIC_ICER)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_ICER** | NVIC_BA+0x080 | R/W | IRQ0 ~ IRQ31 关中断 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **CLRENA** | 关中断位<br>每位控制一个中断，从 IRQ0 到 IRQ31 ( 向量号 16 ~ 47)<br>对每一位的操作如下<br>写:<br>0 = 无效.<br>1 =写1关中断<br>读:<br>0 = 中断被 禁止.<br>1 = 中断已使能. |

## IRQ0 ~ IRQ31 中断挂起控制 （NVIC_ISPR）

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_ISPR** | NVIC_BA+0x100 | R/W | IRQ0 ~ IRQ31 中断挂起 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **SETPEND** | **中断挂起控制**<br>每位控制一个中断，从 IRQ0 到 IRQ31 ( 向量号 16 ~ 47)<br>对每一位的操作如下<br>写:<br>0 = 无效.<br>1 = 写1中断挂起<br>读:<br>0 = 中断未挂起，即无中断等待处理.<br>1 = 中断已挂起，即有中断等待处理. |

## IRQ0 ~ IRQ31 清除中断挂起 （NVIC_ICPR）

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_ICPR** | NVIC_BA+0x180 | R/W | IRQ0 ~ IRQ31清除挂起 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **CLRPEND** | **清除中断挂起位**<br>每位控制一个中断，从 IRQ0 到 IRQ31 ( 向量号 16 ~ 47)<br>对每一位的操作如下<br>写:<br>0 = 无效.<br>1 = 写1清除中断挂起<br>读:<br>0 = 中断未挂起，即无中断等待处理.<br>1 = 中断已挂起，即有中断等待处理. |

## IRQ0 ~ IRQ3 中断优先级 (NVIC_IPR0)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_IPR0** | NVIC_BA+0x300 | R/W | IRQ0 ~ IRQ3 中断优先级 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **PRI_3** | | **Reserved** | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **PRI_2** | | **Reserved** | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **PRI_1** | | **Reserved** | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PRI_0** | | **Reserved** | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **PRI_3** | **IRQ3 的优先级**<br>0 优先级最高，3优先级最低 |
| [29:24] | **Reserved** | 保留 |
| [23:22] | **PRI_2** | **IRQ2 的优先级**<br>0 优先级最高，3优先级最低 |
| [21:16] | **Reserved** | 保留 |
| [15:14] | **PRI_1** | **IRQ1 的优先级**<br>0 优先级最高，3优先级最低 |
| [13:8] | **Reserved** | 保留 |
| [7:6] | **PRI_0** | **IRQ0 的优先级**<br>0 优先级最高，3优先级最低 |
| [5:0] | **Reserved** | 保留 |

## IRQ4 ~ IRQ7 中断优先级控制 (NVIC_IPR1)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_IPR1** | NVIC_BA+0x304 | R/W | IRQ4 ~ IRQ7 中断优先级控制 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **PRI_7** | | **Reserved** | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **PRI_6** | | **Reserved** | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **PRI_5** | | **Reserved** | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PRI_4** | | **Reserved** | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **PRI_7** | **IRQ7 的优先级**<br>0 优先级最高，3优先级最低 |
| [29:24] | **Reserved** | 保留 |
| [23:22] | **PRI_6** | **IRQ6 的优先级**<br>0 优先级最高，3优先级最低 |
| [21:16] | **Reserved** | 保留 |
| [15:14] | **PRI_5** | **IRQ5 的优先级**<br>0 优先级最高，3优先级最低 |
| [13:8] | **Reserved** | 保留 |
| [7:6] | **PRI_4** | **IRQ4 的优先级**<br>0 优先级最高，3优先级最低 |
| [5:0] | **Reserved** | 保留 |

## IRQ8 ~ IRQ11 中断优先级控制 (NVIC_IPR2)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_IPR2** | NVIC_BA+0x308 | R/W | IRQ8 ~ IRQ11中断优先级控制 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **PRI_11** | | **Reserved** | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **PRI_10** | | **Reserved** | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **PRI_9** | | **Reserved** | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PRI_8** | | **Reserved** | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **PRI_11** | **IRQ11 的优先级**<br>0 优先级最高，3优先级最低 |
| [29:24] | **Reserved** | 保留 |
| [23:22] | **PRI_10** | **IRQ10的优先级**<br>0 优先级最高，3优先级最低 |
| [21:16] | **Reserved** | 保留 |
| [15:14] | **PRI_9** | **IRQ9 的优先级**<br>0 优先级最高，3优先级最低 |
| [13:8] | **Reserved** | 保留 |
| [7:6] | **PRI_8** | **IRQ8 的优先级**<br>0 优先级最高，3优先级最低 |
| [5:0] | **Reserved** | 保留 |

## IRQ12 ~ IRQ15 中断优先级控制 (NVIC_IPR3)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_IPR3** | NVIC_BA+0x30C | R/W | IRQ12 ~ IRQ15中断优先级控制 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **PRI_15** | | **Reserved** | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **PRI_14** | | **Reserved** | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **PRI_13** | | **Reserved** | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PRI_12** | | **Reserved** | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **PRI_15** | **IRQ15的优先级**<br>0 优先级最高，3优先级最低 |
| [29:24] | **Reserved** | 保留 |
| [23:22] | **PRI_14** | **IRQ14 的优先级**<br>0 优先级最高，3优先级最低 |
| [21:16] | **Reserved** | 保留 |
| [15:14] | **PRI_13** | **IRQ13 的优先级**<br>0 优先级最高，3优先级最低 |
| [13:8] | **Reserved** | 保留 |
| [7:6] | **PRI_12** | **IRQ12 的优先级**<br>0 优先级最高，3优先级最低 |
| [5:0] | **Reserved** | 保留 |

## IRQ16 ~ IRQ19 中断优先级控制 (NVIC_IPR4)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_IPR4** | NVIC_BA+0x310 | R/W | IRQ16 ~ IRQ19中断优先级控制 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **PRI_19** | | **Reserved** | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **PRI_18** | | **Reserved** | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **PRI_17** | | **Reserved** | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PRI_16** | | **Reserved** | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **PRI_19** | **IRQ19 的优先级**<br>0 优先级最高，3优先级最低 |
| [29:24] | **Reserved** | 保留 |
| [23:22] | **PRI_18** | **IRQ18 的优先级**<br>0 优先级最高，3优先级最低 |
| [21:16] | **Reserved** | 保留 |
| [15:14] | **PRI_17** | **IRQ17 的优先级**<br>0 优先级最高，3优先级最低 |
| [13:8] | **Reserved** | 保留 |
| [7:6] | **PRI_16** | **IRQ16 的优先级**<br>0 优先级最高，3优先级最低 |
| [5:0] | **Reserved** | 保留 |

## IRQ20 ~ IRQ23 中断优先级控制 (NVIC_IPR5)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_IPR5** | NVIC_BA+0x314 | R/W | IRQ20 ~ IRQ23中断优先级控制 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **PRI_23** | | **Reserved** | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **PRI_22** | | **Reserved** | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **PRI_21** | | **Reserved** | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PRI_20** | | **Reserved** | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **PRI_23** | **IRQ23 的优先级**<br>0 优先级最高，3优先级最低 |
| [29:24] | **Reserved** | 保留 |
| [23:22] | **PRI_22** | **IRQ22 的优先级**<br>0 优先级最高，3优先级最低 |
| [21:16] | **Reserved** | 保留 |
| [15:14] | **PRI_21** | **IRQ21 的优先级**<br>0 优先级最高，3优先级最低 |
| [13:8] | **Reserved** | 保留 |
| [7:6] | **PRI_20** | **IRQ20 的优先级**<br>0 优先级最高，3优先级最低 |
| [5:0] | **Reserved** | 保留 |

## IRQ24 ~ IRQ27 中断优先级控制 (NVIC_IPR6)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_IPR6** | NVIC_BA+0x318 | R/W | IRQ24 ~ IRQ27中断优先级控制 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **PRI_27** | | **Reserved** | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **PRI_26** | | **Reserved** | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **PRI_25** | | **Reserved** | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PRI_24** | | **Reserved** | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **PRI_27** | **IRQ27 的优先级**<br>0 优先级最高，3优先级最低 |
| [29:24] | **Reserved** | 保留 |
| [23:22] | **PRI_26** | **IRQ26 的优先级**<br>0 优先级最高，3优先级最低 |
| [21:16] | **Reserved** | 保留 |
| [15:14] | **PRI_25** | **IRQ25 的优先级**<br>0 优先级最高，3优先级最低 |
| [13:8] | **Reserved** | 保留 |
| [7:6] | **PRI_24** | **IRQ24 的优先级**<br>0 优先级最高，3优先级最低 |
| [5:0] | **Reserved** | 保留 |

## IRQ28 ~ IRQ31 中断优先级控制 (NVIC_IPR7)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NVIC_IPR7** | NVIC_BA+0x31C | R/W | IRQ28 ~ IRQ31中断优先级控制 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **PRI_31** | | **Reserved** | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **PRI_30** | | **Reserved** | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **PRI_29** | | **Reserved** | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PRI_28** | | **Reserved** | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **PRI_31** | **IRQ31 的优先级**<br>0 优先级最高，3优先级最低 |
| [29:24] | **Reserved** | 保留 |
| [23:22] | **PRI_30** | **IRQ30 的优先级**<br>0 优先级最高，3优先级最低 |
| [21:16] | **Reserved** | 保留 |
| [15:14] | **PRI_29** | **IRQ29 的优先级**<br>0 优先级最高，3优先级最低 |
| [13:8] | **Reserved** | 保留 |
| [7:6] | **PRI_28** | **IRQ28 的优先级**<br>0 优先级最高，3优先级最低 |
| [5:0] | **Reserved** | 保留 |

### 6.2.8.5 中断源寄存器

**R**: 只读, **W**: 只写, **R/W**: 可读写

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **INT 基址:**<br>**INT_BA = 0x5000_0300** | | | | |
| **IRQ0_SRC** | INT_BA+0x00 | R | IRQ0 (BOD) 中断源 | 0xXXXX_XXXX |
| **IRQ1_SRC** | INT_BA+0x04 | R | IRQ1 (WDT) 中断源 | 0xXXXX_XXXX |
| **IRQ2_SRC** | INT_BA+0x08 | R | IRQ2 (EINT0) 中断源 | 0xXXXX_XXXX |
| **IRQ3_SRC** | INT_BA+0x0C | R | IRQ3 (EINT1) 中断源 | 0xXXXX_XXXX |
| **IRQ4_SRC** | INT_BA+0x10 | R | IRQ4 (P0-P4) 中断源 | 0xXXXX_XXXX |
| **IRQ5_SRC** | INT_BA+0x14 | R | IRQ5 (P5-PA) 中断源 | 0xXXXX_XXXX |
| **IRQ6_SRC** | INT_BA+0x18 | R | IRQ6 (BPWM0) 中断源 | 0xXXXX_XXXX |
| **IRQ7_SRC** | INT_BA+0x1C | R | IRQ7 (EADC0) 中断源 | 0xXXXX_XXXX |
| **IRQ8_SRC** | INT_BA+0x20 | R | IRQ8 (TMR0) 中断源 | 0xXXXX_XXXX |
| **IRQ9_SRC** | INT_BA+0x24 | R | IRQ9 (TMR1) 中断源 | 0xXXXX_XXXX |
| **IRQ10_SRC** | INT_BA+0x28 | R | IRQ10 (TMR2) 中断源 | 0xXXXX_XXXX |
| **IRQ11_SRC** | INT_BA+0x2C | R | IRQ11 (TMR3) 中断源 | 0xXXXX_XXXX |
| **IRQ12_SRC** | INT_BA+0x30 | R | IRQ12 (UART0) 中断源 | 0xXXXX_XXXX |
| **IRQ13_SRC** | INT_BA+0x34 | R | IRQ13 (UART1) 中断源 | 0xXXXX_XXXX |
| **IRQ14_SRC** | INT_BA+0x38 | R | IRQ14 (SPI0) 中断源 | 0xXXXX_XXXX |
| **IRQ15_SRC** | INT_BA+0x3C | R | IRQ15 (SPI1) 中断源 | 0xXXXX_XXXX |
| **IRQ16_SRC** | INT_BA+0x40 | R | IRQ16 (SPI2) 中断源 | 0xXXXX_XXXX |
| **IRQ18_SRC** | INT_BA+0x48 | R | IRQ18 ($I^2C0$) 中断源 | 0xXXXX_XXXX |
| **IRQ19_SRC** | INT_BA+0x4C | R | IRQ19 (CKD) 中断源 | 0xXXXX_XXXX |
| **IRQ21_SRC** | INT_BA+0x54 | R | IRQ21 (EPWM0) 中断源 | 0xXXXX_XXXX |
| **IRQ22_SRC** | INT_BA+0x58 | R | IRQ22 (EPWM1) 中断源 | 0xXXXX_XXXX |
| **IRQ23_SRC** | INT_BA+0x5C | R | IRQ23 (ECAP0) 中断源 | 0xXXXX_XXXX |
| **IRQ24_SRC** | INT_BA+0x60 | R | IRQ24 (ECAP1) 中断源 | 0xXXXX_XXXX |
| **IRQ25_SRC** | INT_BA+0x64 | R | IRQ25 (ACMP) 中断源 | 0xXXXX_XXXX |
| **IRQ28_SRC** | INT_BA+0x70 | R | IRQ28 (PWRWU) 中断源 | 0xXXXX_XXXX |
| **IRQ29_SRC** | INT_BA+0x74 | R | IRQ29 (EADC1) 中断源 | 0xXXXX_XXXX |

| | | | | |
|---|---|---|---|---|
| **IRQ30_SRC** | INT_BA+0x78 | R | IRQ30 (EADC2) 中断源 | 0xXXXX_XXXX |
| **IRQ31_SRC** | INT_BA+0x7C | R | IRQ31 (EADC3) 中断源 | 0xXXXX_XXXX |
| **NMI_SEL** | INT_BA+0x80 | R/W | NMI 中断源选 择 | 0x0000_0000 |
| **MCU_IRQ** | INT_BA+0x84 | R/W | MCU 中断请求控制 | 0x0000_0000 |
| **MCU_IRQCR** | INT_BA+0x88 | R/W | MCU 中断响应时间控制 | 0x0000_0000 |

## IRQ0 (BOD) 中断源 (IRQ0_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ0_SRC** | INT_BA+0x00 | R | IRQ0 (BOD) 中断源 | 0xXXXX_XXXX |

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

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **BOD_INT** | **IRQ0 中断源**<br>0 = IRQ0 中断源不是来至 (BOD_INT).<br>1 = IRQ0 中断源来至 (BOD_INT). |

## IRQ1 (WDT) 中断源 (IRQ1_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ1_SRC** | INT_BA+0x04 | R | IRQ1 (WDT) 中断源来至 | 0xXXXX_XXXX |

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

| 位 | 描述 | |
|---|---|---|
| [31:2] | **Reserved** | 保留 |
| [1] | **WWDT_INT** | **IRQ1中断源**<br>0 = IRQ1中断源不是来至(WWDT _INT).<br>1 = IRQ1 中断源来至 (WWDT_INT). |
| [0] | **WDT_INT** | **IRQ1中断源**<br>0 = IRQ1 中断源不是来至 (WDT _INT).<br>1 = IRQ1 中断源来至 (WDT_INT). |

## IRQ2 (EINT0) 中断源 (IRQ2_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ2_SRC** | INT_BA+0x08 | R | IRQ2 (EINT0) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **EINT0** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **EINT0** | **IRQ2中断源**<br>0 = IRQ2中断源不是来至 P3.2 (EINT0).<br>1 = IRQ2 中断源来至 P3.2 (EINT0). |

## IRQ3 (EINT1) 中断源 (IRQ3_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ3_SRC** | INT_BA+0x0C | R | IRQ3 (EINT1) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **EINT1** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **EINT1** | **IRQ3 中断源**<br>0 = IRQ3中断源不是来至P3.3 (EINT1).<br>1 = IRQ3中断源来至P3.3 (EINT1). |

## IRQ4 (P0-P4) 中断源(IRQ4_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ4_SRC** | INT_BA+0x10 | R | IRQ4 (P0-P4) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | **P4_INT** | **P3_INT** | **P2_INT** | **P1_INT** | **P0_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:5] | **Reserved** | 保留 |
| [4] | **P4_INT** | **IRQ4 中断源**<br>0 = IRQ4 中断源不是来至 (P4_INT).<br>1 = IRQ4 中断源来至 (P4_INT). |
| [3] | **P3_INT** | **IRQ4中断源**<br>0 = IRQ4 中断源不是来至 (P3_INT).<br>1 = IRQ4 中断源来至 (P3_INT). |
| [2] | **P2_INT** | **IRQ4 中断源**<br>0 = IRQ4 中断源不是来至 (P2_INT).<br>1 = IRQ4 中断源来至 (P2_INT). |
| [1] | **P1_INT** | **IRQ4 中断源**<br>0 = IRQ4 中断源不是来至 (P1_INT).<br>1 = IRQ4 中断源来至 (P1_INT). |
| [0] | **P0_INT** | **IRQ4 中断源**<br>0 = IRQ4 中断源不是来至 (P0_INT).<br>1 = IRQ4 中断源来至(P0_INT). |

## IRQ5 (P5-PA) 中断源(IRQ5_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ5_SRC** | INT_BA+0x14 | R | IRQ5 (P5-PA) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | **PA_INT** | **P9_INT** | **P8_INT** | **P7_INT** | **P6_INT** | **P5_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:6] | **Reserved** | 保留 |
| [5] | **PA_INT** | **IRQ5中断源**<br>0 = IRQ5 中断源不是来至 (PA_INT).<br>1 = IRQ5 中断源来至 (PA_INT). |
| [4] | **P9_INT** | **IRQ5中断源**<br>0 = IRQ5 中断源不是来至 (P9_INT).<br>1 = IRQ5 中断源来至 (P9_INT). |
| [3] | **P8_INT** | **IRQ5中断源**<br>0 = IRQ5 中断源不是来至 (P8_INT).<br>1 = IRQ5 中断源来至 (P8_INT). |
| [2] | **P7_INT** | **IRQ5 中断源**<br>0 = IRQ5 中断源不是来至 (P7_INT).<br>1 = IRQ5 中断源来至 (P7_INT). |
| [1] | **P6_INT** | **IRQ5 中断源**<br>0 = IRQ5 中断源不是来至 (P6_INT).<br>1 = IRQ5 中断源来至 (P6_INT). |
| [0] | **P5_INT** | **IRQ5 中断源**<br>0 = IRQ5 中断源不是来至 (P5_INT).<br>1 = IRQ5 中断源来至 (P5_INT). |

## IRQ6 (BPWM0) 中断源(IRQ6_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ6_SRC** | INT_BA+0x18 | R | IRQ6 (BPWM0) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | **BPCH1_INT** | **BPCH0_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:2] | **Reserved** | 保留 |
| [1] | **BPCH1_INT** | **IRQ6 中断源**<br>0 = IRQ6 中断源不是来至 BPWM0 通道 1 (BPCH1_INT).<br>1 = IRQ6 中断源来至 BPWM0 通道 1 (BPCH1_INT). |
| [0] | **BPCH0_INT** | **IRQ6 中断源**<br>0 = IRQ6 中断源不是来至 BPWM0 通道0 (BPCH0_INT).<br>1 = IRQ6 中断源来至 BPWM0 通道 0 (BPCH0_INT). |

## IRQ7 (EADC0) 中断源(IRQ7_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ7_SRC** | INT_BA+0x1C | R | IRQ7 (EADC0) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **EADC0_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **EADC0_INT** | **IRQ7中断源**<br>0 = IRQ7 中断源不是来至 EADC0 (EADC0_INT).<br>1 = IRQ7 中断源来至 EADC0 (EADC0_INT). |

## IRQ8 (TMR0) 中断源(IRQ8_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ8_SRC** | INT_BA+0x20 | R | IRQ8 (TMR0) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **TMR0_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **TMR0_INT** | **IRQ8中断源**<br>0 = IRQ8 中断源不是来至 (TMR0_INT).<br>1 = IRQ8 中断源来至 (TMR0_INT). |

## IRQ9 (TMR1) 中断源(IRQ9_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ9_SRC** | INT_BA+0x24 | R | IRQ9 (TMR1) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **TMR1_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **TMR1_INT** | **IRQ9中断源**<br>0 = IRQ9 中断源不是来至 (TMR1_INT).<br>1 = IRQ9 中断源来至 (TMR1_INT). |

## IRQ10 (TMR2) 中断源(IRQ10_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ10_SRC** | INT_BA+0x28 | R | IRQ10 (TMR2) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **TMR2_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **TMR2_INT** | **IRQ10中断源**<br>0 = IRQ10 中断源不是来至 (TMR2_INT).<br>1 = IRQ10 中断源来至 (TMR2_INT). |

## IRQ11 (TMR3) 中断源(IRQ11_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ11_SRC** | INT_BA+0x2C | R | IRQ11 (TMR3) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **TMR3_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **TMR3_INT** | **IRQ11中断源**<br>0 = IRQ11 中断源不是来至 (TMR3_INT).<br>1 = IRQ11 中断源来至 (TMR3_INT). |

## IRQ12 (UART0) 中断源(IRQ12_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ12_SRC** | INT_BA+0x30 | R | IRQ12 (UART0) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **UART0_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **UART0_INT** | **IRQ12中断源**<br>0 = IRQ12 中断源不是来至 (UART0_INT).<br>1 = IRQ12 中断源来至 (UART0_INT). |

## IRQ13 (UART1) 中断源(IRQ13_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ13_SRC** | INT_BA+0x34 | R | IRQ13 (UART1) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **UART1_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **UART1_INT** | **IRQ13中断源**<br>0 = IRQ13 中断源不是来至(UART1_INT).<br>1 = IRQ13 中断源来至 (UART1_INT). |

## IRQ14 (SPI0) 中断源(IRQ14_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ14_SRC** | INT_BA+0x38 | R | IRQ14 (SPI0) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **SPI0_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **SPI0_INT** | **IRQ14中断源**<br>0 = IRQ14 中断源不是来至 (SPI0_INT).<br>1 = IRQ14 中断源来至 (SPI0_INT). |

## IRQ15 (SPI1) 中断源(IRQ15_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ15_SRC** | INT_BA+0x3C | R | IRQ15 (SPI1) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **SPI1_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **SPI1_INT** | **IRQ15中断源**<br>0 = IRQ15 中断源不是来至 (SPI1_INT).<br>1 = IRQ15 中断源来至 (SPI1_INT). |

## IRQ16 (SPI2) 中断源(IRQ16_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ16_SRC** | INT_BA+0x40 | R | IRQ16 (SPI2) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **SPI2_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **SPI2_INT** | **IRQ16中断源**<br>0 = IRQ16 中断源不是来至 (SPI2_INT).<br>1 = IRQ16 中断源来至 (SPI2_INT). |

## IRQ18 (I2C) 中断源(IRQ18_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ18_SRC** | INT_BA+0x48 | R | IRQ18 (I2C0) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **I2C0_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **I2C0_INT** | **IRQ18中断源**<br>0 = IRQ18 中断源不是来至 (I2C0_INT).<br>1 = IRQ18 中断源来至 (I2C0_INT). |

## IRQ19 (CKD) 中断源(IRQ19_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ19_SRC** | INT_BA+0x4C | R | IRQ19 (CKD) 中断源 | 0xXXXX_XXXX |

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

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **CKD_INT** | **IRQ19中断源**<br>0 = IRQ19中断源不是来至 (CKD_INT).<br>1 = IRQ19中断源来至 (CKD_INT). |

## IRQ21 (EPWM0) 中断源 (IRQ21_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ21_SRC** | INT_BA+0x54 | R | IRQ21 (EPWM0) | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **EPWM0_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **EPWM0_INT** | **IRQ21 中断源**<br>0 = IRQ21 中断源不是来至 (EPWM0_INT).<br>1 = IRQ21 中断源来至 (EPWM0_INT). |

## IRQ22 (EPWM1) 中断源(IRQ22_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ22_SRC** | INT_BA+0x58 | R | IRQ22 (EPWM1) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **EPWM1_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **EPWM1_INT** | **IRQ22中断源**<br>0 = IRQ22 中断源不是来至 (EPWM1_INT).<br>1 = IRQ22中断源来至 (EPWM1_INT). |

## IRQ23 (ECAP0) 中断源(IRQ23_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ23_SRC** | INT_BA+0x5C | R | IRQ23 (ECAP0) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **ECAP0_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **ECAP0_INT** | **IRQ23中断源**<br>0 = IRQ23 中断源不是来至 (ECAP0_INT).<br>1 = IRQ23 中断源来至 (ECAP0_INT). |

## IRQ24 (ECAP1) 中断源(IRQ24_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ24_SRC** | INT_BA+0x60 | R | IRQ24 (ECAP1) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **ECAP1_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **ECAP1_INT** | **IRQ24中断源**<br>0 = IRQ24 中断源不是来至 (ECAP1_INT).<br>1 = IRQ24 中断源来至 (ECAP1_INT). |

## IRQ25 (ACMP) 中断源(IRQ25_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ25_SRC** | INT_BA+0x64 | R | IRQ25 (ACMP) 中断源 | 0xXXXX_XXXX |

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

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **ACMP_INT** | **IRQ25中断源**<br>0 = IRQ25 中断源不是来至 (ACMP_INT).<br>1 = IRQ25 中断源来至 (ACMP_INT). |

## IRQ28 (PWRWU) 中断源(IRQ28_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ28_SRC** | INT_BA+0x70 | R | IRQ28 (PWRWU) 中断源 | 0xXXXX_XXXX |

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

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **PWRWU_INT** | **IRQ28中断源**<br>0 = IRQ28 中断源不是来至 (PWRWU_INT).<br>1 = IRQ28 中断源来至 (PWRWU_INT). |

## IRQ29 (EADC1) 中断源(IRQ29_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ29_SRC** | INT_BA+0x74 | R | IRQ29 (EADC1) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **EADC1_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **EADC1_INT** | **IRQ29中断源**<br>0 = IRQ29 中断源不是来至 (EADC1_INT).<br>1 = IRQ29 中断源来至 (EADC1_INT). |

## IRQ30 (EADC2) 中断源(IRQ30_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ30_SRC** | INT_BA+0x78 | R | IRQ30 (EADC2) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **EADC2_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **EADC2_INT** | **IRQ30 中断源**<br>0 = IRQ30 中断源不是来至 (EADC2_INT).<br>1 = IRQ30 中断源来至 (EADC2_INT). |

## IRQ31 (EADC3) 中断源(IRQ31_SRC)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **IRQ31_SRC** | INT_BA+0x7C | R | IRQ31 (EADC3) 中断源 | 0xXXXX_XXXX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | **EADC3_INT** |

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **EADC3_INT** | **IRQ31中断源**<br>0 = IRQ31 中断源不是来至 (EADC3_INT).<br>1 = IRQ31 中断源来至 (EADC3_INT). |

## NMI 中断源选 择 (NMI_SEL)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **NMI_SEL** | INT_BA+0x80 | R/W | NMI中断源选 择 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:9] | **Reserved** | 保留 |
| [8] | **NMI_EN** | **NMI 中断使能 (保护位)**<br>0 =关闭 IRQ0~31 到 NMI 的通路. (软件依然可以通过NMI挂起位申请NMI断)<br>1 = 打开IRQ0~31 到 NMI 的通路. |
| [7:5] | | |
| [4:0] | **NMI_SEL** | **NMI 中断源选 择**<br>写一个IRQ0~IRQ31 中断号，这个中断将做为NMI中断， 复位后 的值是0，即选 IRQ0 为 NMI中断 |

## MCU 中断请求控制寄存器 (MCU_IRQ)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **MCU_IRQ** | INT_BA+0x84 | R/W | 中断请求控制寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **MCU_IRQ** | **中断请求控制**<br>MCU_IRQ 收集中断源，向Cortex®-M0. 发送 异步请求<br>MCU_IRQ[n] 为 0时, 表示无中断请求，写 1 会发出一中断请求<br>MCU_IRQ[n] 为 1时, 表示有中断请求。写1会清除这个中断请求 |

## MCU 中断响应时间控制 (MCU_IRQCR)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **MCU_IRQCR** | INT_BA+0x88 | R/W | MCU 中断响应时间控制 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **FAST_IRQ** | **快速中断响应使能位**<br>0 = 中断响应时间固定 13 HCLK,<br>1 = 中断响应时间不固定，但尽快 |

### 6.2.9 系统控制寄存器

**R**: 只读, **W**: 只写, **R/W**: 读/写

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **SCB 基址**<br>**SCB_BA = 0xE000_ED00** | | | | |
| **CPUID** | SCB_BA+0x00 | R | CPUID | 0x410C_C200 |
| **ICSR** | SCB_BA+0x04 | R/W | 中断控制状态 | 0x0000_0000 |
| **AIRCR** | SCB_BA+0x0C | R/W | 应用中断和复位控制 | 0xFA05_0000 |
| **SCR** | SCB_BA+0x10 | R/W | 系统控制 | 0x0000_0000 |
| **SHPR2** | SCB_BA+0x1C | R/W | 系统中断优先级 2 | 0x0000_0000 |
| **SHPR3** | SCB_BA+0x20 | R/W | 系统中断优先级 3 | 0x0000_0000 |

## CPUID (CPUID)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CPUID** | SCB_BA+0x00 | R | CPUID | 0x410C_C200 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **IMPLEMENTER** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | **PART** | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **PARTNO** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PARTNO** | | | | **REVISION** | | | |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **IMPLEMENTER** | **ARM®代码**<br>(ARM® = 0x41).. |
| [23:20] | **Reserved** | 保留 |
| [19:16] | **PART** | **处理器结构**<br>ARMv6-M 结构，读，得到0xC |
| [15:4] | **PARTNO** | **处理器料号**<br>Cortex®-M0系列，读, 得到 0xC20 |
| [3:0] | **REVISION** | **版本号**<br>读，得到0 |

## 中断控制状态(ICSR)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ICSR** | SCB_BA+0x04 | R/W | 中断控制状态 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **NMIPENDSET** | **Reserved** | | **PENDSVSET** | **PENDSVCLR** | **PENDSTSET** | **PENDSTCLR** | **Reserved** |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | **ISRPENDING** | **Reserved** | | | | **VECTPENDING** | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **VECTPENDING** | | | | **Reserved** | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | **VECTACTIVE** | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31] | **NMIPENDSET** | **NMI 挂起**<br>由于此位挂起后MCU会立即去执行NMI代码，硬件清0 此位，所以，只有在执行NMI代码时，再次置1此位，此位才能读到1.<br>写:<br>0 = 无效<br>1 = 让 NMI 挂起.<br>读:<br>0 = NMI 没挂起.<br>1 = NMI 已挂起 |
| [30:29] | **Reserved** | 保留 |
| [28] | **PENDSVSET** | **PendSV挂起**<br>写:<br>0 = 无效<br>1 = 让 PenSV 挂起.<br>读:<br>0 = PenSV 没挂起.<br>1 = PenSV 已挂起<br>写1此位是唯一申请 PenSV中断的办法 |
| [27] | **PENDSVCLR** | **清除PendSV挂起 (只写)**<br>0 = 无效.<br>1 = 清除 PendSV 挂起位. |

| 位 | 描述 | |
|---|---|---|
| [26] | **PENDSTSET** | **SysTick 挂起位**<br>写:<br>0 = 无效<br>1 = 让SysTick 挂起.<br>读:<br>0 = SysTick没挂起.<br>1 = SysTick 已挂起 |
| [25] | **PENDSTCLR** | **清除SysTick 挂起位 (只写)**<br>0 = 无效.<br>1 = 清除SysTick 挂起位. |
| [24:23] | **Reserved** | 保留 |
| [22] | **ISRPENDING** | **中断挂起标志 (只读)**<br>0 = 没有中断挂起.<br>1 = 有中断在挂起 |
| [21:18] | **Reserved** | 保留 |
| [17:12] | **VECTPENDING** | **向量挂起指示 (只读)**<br>0 = 无中断挂起.<br>非0 = 挂起中断中，优先级最高的向量号. |
| [11:6] | **Reserved** | 保留 |
| [5:0] | **VECTACTIVE** | **向量执行指示 (只读)**<br>0 = 线程模式（Thread mode）<br>非0 = 正在执行的中断的向量号. |

## 应用中断和复位控制 (AIRCR)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **AIRCR** | SCB_BA+0x0C | R/W | 应用、中断和复位控制 | 0xFA05_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **VECTKEY** | **钥匙 (只写)**<br>写这个寄存器时， VECTORKEY 心须为 0x05FA, 否则写无效 |
| [15:3] | **Reserved** | 保留 |
| [2] | **SYSRESETREQ** | **系统复位 (只写)**<br>0 = 无效<br>1 = 系统复位 |
| [1:0] | **Reserved** | 保留 |

## 系统控制 (SCR)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **SCR** | SCB_BA+0x10 | R/W | 系统控制 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:5] | **Reserved** | 保留 |
| [4] | **SEVONPEND** | **挂起状态时的发送事件位**<br>0 = 只有使能中断或者事件可以唤醒处理器，不包括禁用中断在内。<br>1 = 使能事件和所有中断，包括禁用中断，都可以唤醒处理器。<br>当一个事件或中断进入挂起状态时，事件信号从 WFE 唤醒处理器。如果处理器不是在等待该事件，那么这个事件会被注册到并影响下一个 WFE。<br>处理器也会在执行一个 SEV 指令或者一个外部事件时被唤醒。 |
| [3] | **Reserved** | 保留 |
| [2] | **SLEEPDEEP** | **深度休眠（掉电）使能位**<br>0 = 休眠模式.<br>1 = 深度休眠模式 |
| [1] | **SLEEPONEXIT** | **中断返回后立即休眠使能位**<br>0 = 返回线程模式后不再休眠.<br>1 = 休眠，深度休眠时，中断唤醒执行中断代码返回后，立即再次休眠. |
| [0] | **Reserved** | 保留 |

## 系统中断优先级 2 (SHPR2)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **SHPR2** | SCB_BA+0x1C | R/W | 系统中断优先级2 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **PRI_11** | | **Reserved** | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **PRI_11** | **系统中断11—— SVCall 的优先级**<br>0优先级电最高，3优先级最低 |
| [29:0] | **Reserved** | 保留 |

## 系统中断优先级3 (SHPR3)

| 寄存器 | 地址偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **SHPR3** | SCB_BA+0x20 | R/W | 系统中断优先级3 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **PRI_15** | | **Reserved** | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **PRI_14** | | **Reserved** | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **PRI_15** | **系统中断15—— SysTick的优先级**<br>0优先级电最高，3优先级最低 |
| [29:24] | **Reserved** | 保留 |
| [23:22] | **PRI_14** | **系统中断14—— PendSV 的优先级**<br>0优先级电最高，3优先级最低 |
| [21:0] | **Reserved** | 保留 |

## 6.3 时钟控制器

### 6.3.1 概述

时钟控制器为整个芯片提供时钟源，包括系统时钟和所有外围设备时钟。该控制器还通过单独时钟的开或关，时钟源选择和分频器来进行功耗控制。SLEEPDEEP (SCR[2]) 位设置为1, 同时CPU Cortex™-M0内核执行WFI指令，芯片将进入掉电模式。直到唤醒中断发生，芯片才会退出掉电模式。在掉电模式下，时钟控制器关闭外部4~24MHz晶振和内部22.1184MHz高速RC振荡器，以降低整个系统功耗。图6-3展示了时钟发生器和时钟源控制的大概。

时钟发生器由如下4个时钟源组成：

- 外部4~24 MHz高速晶振(HXT)
- 可编程的PLL输出时钟频率(PLL 由外部 4~24 MHz 晶振或内部 22.1184 MHz 振荡器提供时钟源)
- 内部22.1184 MHz高速振荡器(HIRC)
- 内部10 kHz低速RC振荡器(LIRC)

每一个时钟源开启时都要等待一段时间频率才能稳定。当开启时钟源时稳定计数器开始计数，相关的时钟稳定指示位 (OSC22M_STB(CLKSTATUS[4]), OSC10K_STB(CLKSTATUS[3]), PLL_STB(CLKSTATUS[2]) 和 XTL12M_STB(CLKSTATUS[0]))在计数器到达表6-5的值时会被置1。用户禁用时钟源 (OSC10K_EN(PWRCON[3]), OSC22M_EN(PWRCON[2]), XTL12M_EN(PWRCON[0]) 和 PD(PLLCON[16]))后，相应的时钟稳定指示位会被自动清除。此外，在芯片进入掉电模式时HXT, HIRC 和 PLL的时钟稳定指示位会被清除，芯片被唤醒后相应使能时钟的稳定计数器会重新计数。

| 时钟源 | 时钟稳定计数值 |
|---|---|
| HXT | 4096 HXT |
| PLL | 6144 PLL 时钟源<br>(当 PLL_SRC(PLLCON[19]) = 0时PPL时钟源是HXT，当 PLL_SRC(PLLCON[19]) = 1时PPL时钟源是HIRC) |
| HIRC | 256 HIRC |
| LIRC | 1 LIRC |

表 6-5 时钟稳定计数值表

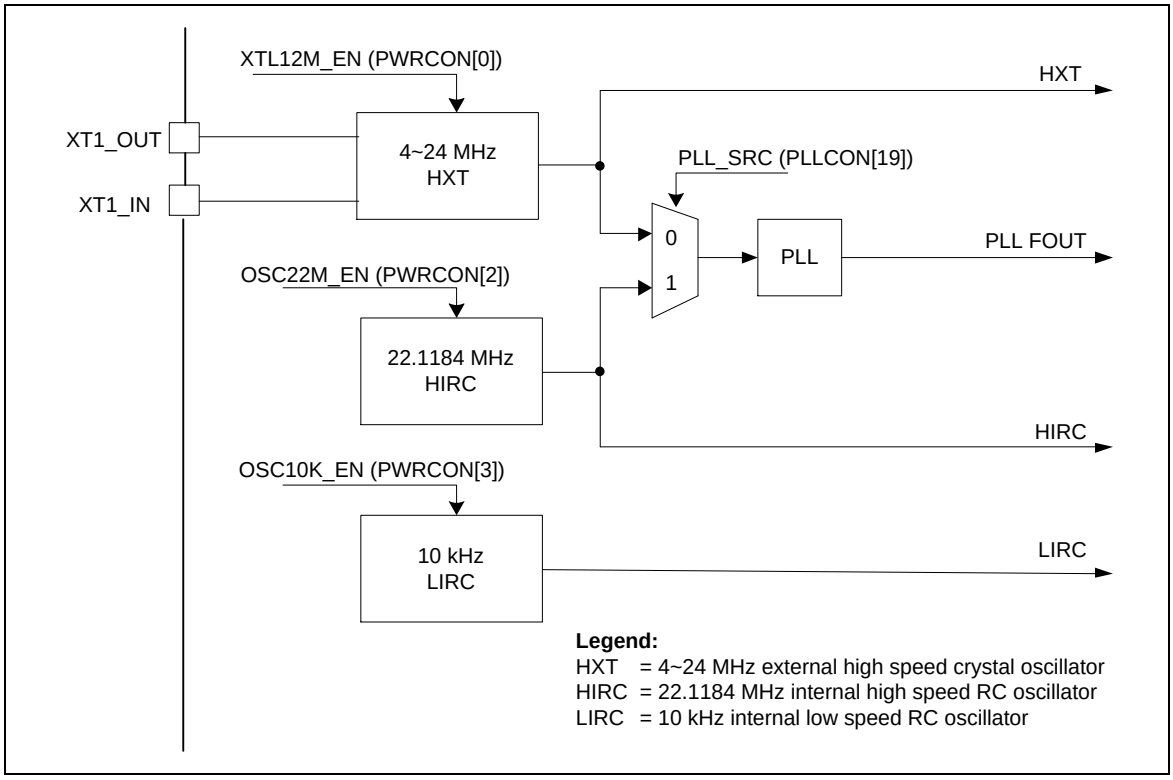
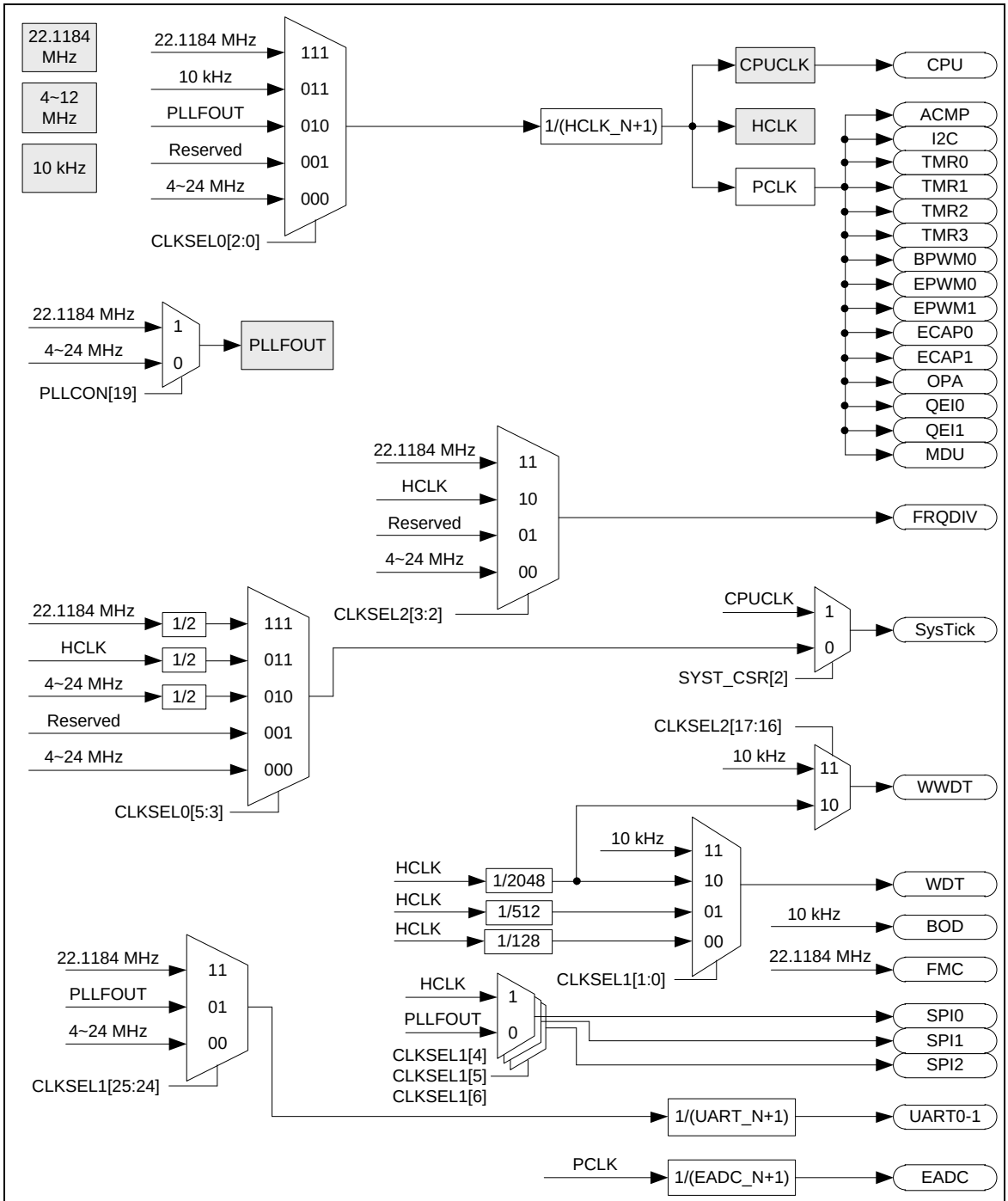

XTL12M_EN (PWRCON[0])
HXT
XT1_OUT
4~24 MHz
HXT
XT1_IN
PLL_SRC (PLLCON[19])
0
1
PLL
PLL FOUT
OSC22M_EN (PWRCON[2])
22.1184 MHz
HIRC
HIRC
OSC10K_EN (PWRCON[3])
10 kHz
LIRC
LIRC
Legend:
HXT = 4~24 MHz external high speed crystal oscillator
HIRC = 22.1184 MHz internal high speed RC oscillator
LIRC = 10 kHz internal low speed RC oscillator

图 6-3 时钟发生器框图




**Note:** Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

图 6-4 时钟发生器全局视图

### 6.3.2 系统时钟和SysTick时钟

系统时钟有 4 个时钟源，由时钟发生器产生。时钟源切换取决于寄存器HCLK_S (CLKSEL0[2:0]). 如 图 6-5 所示

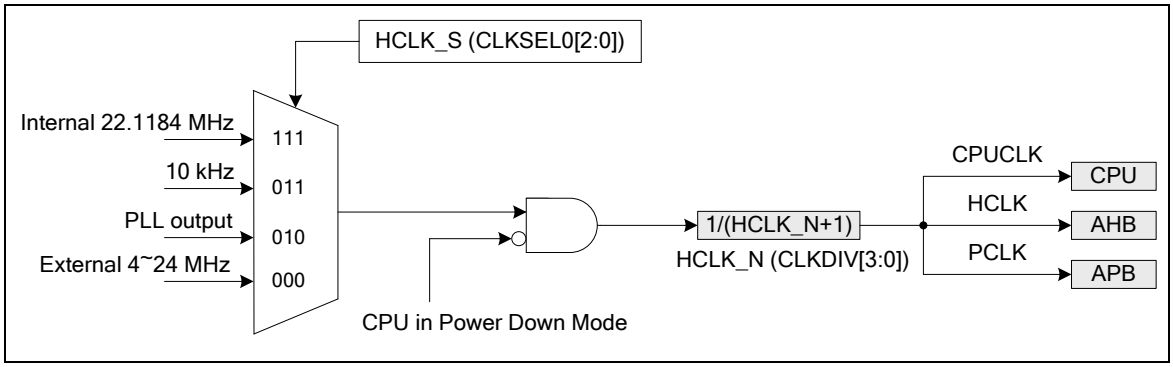



图 6-5 系统时钟框图

Cortex™-M0内核的SysTick 时钟源可以选择CPU时钟或外部时钟(SYST_CSR[2])。如果使用外部时钟，SysTick 时钟 (STCLK) 有 4个时钟源。时钟源切换取决于寄存器 STCLK_S (CLKSEL0[5:3])。如图 6-6.

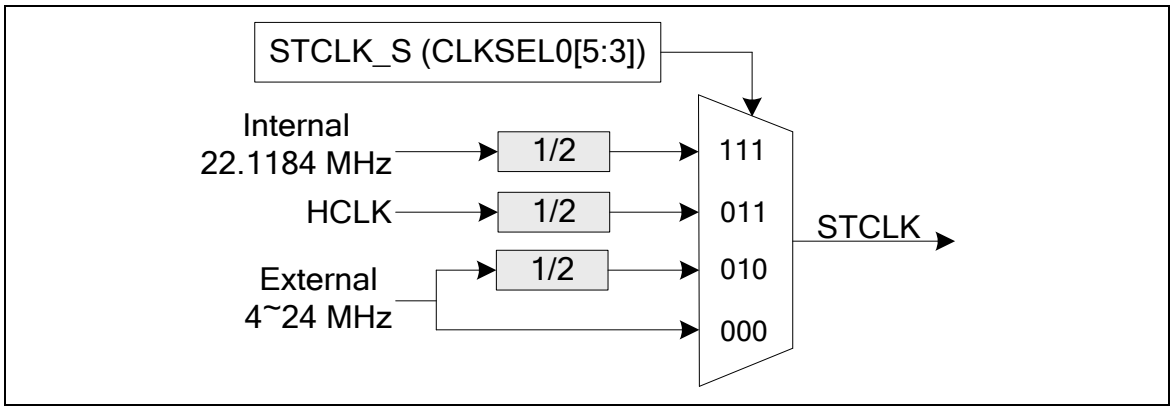



图 6-6 SysTick 时钟控制框图

### 6.3.3 外设时钟

外设的时钟可选择不同的时钟源，请参考6.3.7中CLKSEL1和CLKSEL2寄存器描述。

### 6.3.4 掉电模式（深度睡眠模式）时钟

当芯片进入掉电模式，系统时钟，一些时钟源和一些外设时钟将被关闭。也有一些时钟源与外设时钟仍在工作

如下时钟和外设仍在工作：

- 时钟发生器
  - 内部10 kHz振荡器
- 外设时钟 (当这些外设采用10 kHz作为时钟源)
  - 看门狗定时器

### 6.3.5 分频器输出

该设备带有一个2的若干次幂的频率分频器，该分频器由16个链式的二分频器组成的移位寄存器。其中哪一级的值被输出，由一个16选1的多路转换器选择，该多路转换器接到CLKO管脚上。因此共有16 种时钟分频选择。

输出公式：$F_{CLKO} = F_{FRQDIV_CLK}/2^{(N+1)}$,其中$F_{FRQDIV_CLK}$ 为输入时钟频率，$F_{CLKO}$为时钟分频器输出频率，N 为 FSEL(FRQDIV[3:0])中的4位值。

往DIVIDER_EN (FRQDIV[4])写1，分级计数器开始计数。往DIVIDER_EN (FRQDIV[4])写0，分级计数器禁止。

如果DIVIDER1(FRQDIV[5])设置为1，分频器时钟(FRQDIV_CLK)将忽略2的若干次幂分频器。时钟频率将直接输出到CLKO。

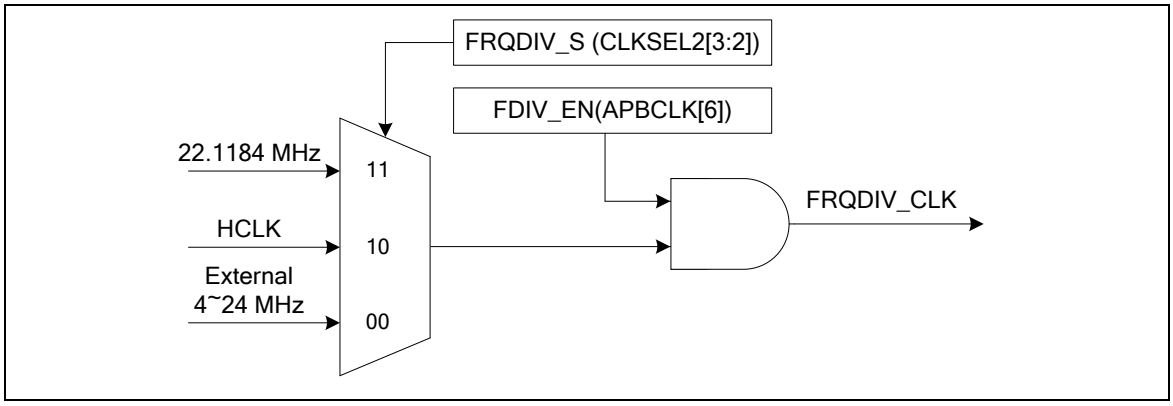



图 6-7 分频器的时钟源

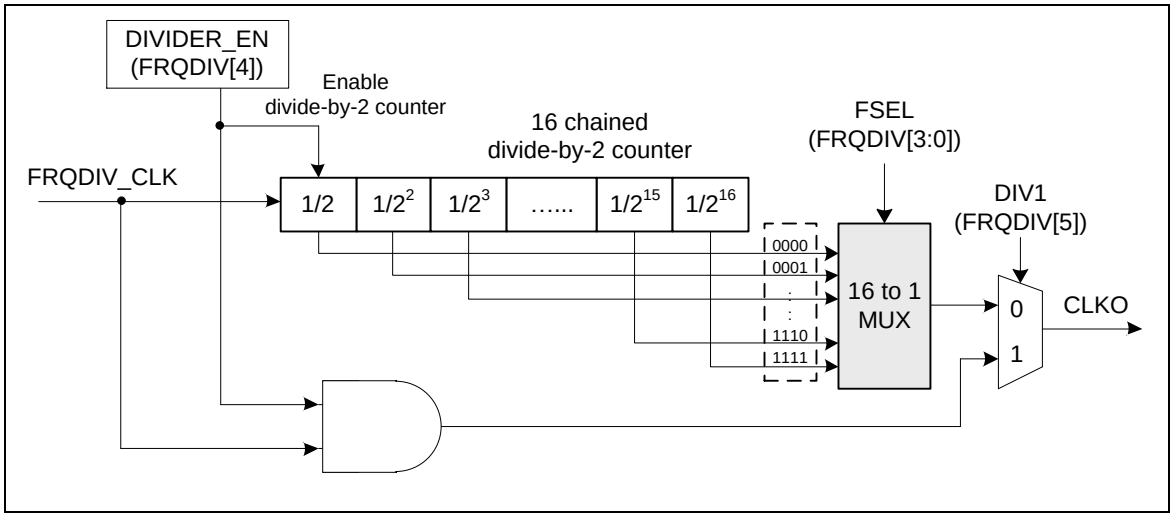



图 6-8 分频器框图

### 6.3.6 寄存器映射

**R:** 只读, **W:** 只写, **R/W:** 读/写

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CLK 基地址:**<br>**CLK_BA = 0x5000_0200** | | | | |
| **PWRCON** | CLK_BA+0x00 | R/W | 系统掉电控制寄存器 | 0x0000_000X |
| **AHBCLK** | CLK_BA+0x04 | R/W | AHB 设备时钟使能控制寄存器 | 0x0000_0005 |
| **APBCLK** | CLK_BA+0x08 | R/W | APB 设备时钟使能控制寄存器 | 0x0000_0001 |
| **CLKSTATUS** | CLK_BA+0x0C | R | 时钟状态监控寄存器 | 0x0000_00XX |
| **CLKSEL0** | CLK_BA+0x10 | R/W | 时钟源选择控制寄存器0 | 0x0000_003X |
| **CLKSEL1** | CLK_BA+0x14 | R/W | 时钟源选择控制寄存器1 | 0xFFFF_FFFF |
| **CLKDIV** | CLK_BA+0x18 | R/W | 时钟分频寄存器 | 0x0000_0000 |
| **CLKSEL2** | CLK_BA+0x1C | R/W | 时钟源选择控制寄存器2 | 0xFFFF_FFFF |
| **PLLCON** | CLK_BA+0x20 | R/W | PLL 控制寄存器 | 0x0005_C22E |
| **FRQDIV** | CLK_BA+0x24 | R/W | 频率分频器控制寄存器 | 0x0000_0000 |

### 6.3.7 寄存器描述

**系统掉电控制寄存器（PWRCON）**

除 BIT[6]，PWRCON 的其他位都受保护，解锁这些位，需要向地址 0x5000_0100 依次写入 “59h”, “16h”, “88h”。 请参考寄存器 REGWRPROT （地址为 GCR_BA+0x100）

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PWRCON** | CLK_BA+0x00 | R/W | 系统掉电控制寄存器 | 0x0000_000X |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | **PD_WU_STS** | **PD_WU_INT_EN** | **Reserved** | **OSC10K_EN** | **OSC22M_EN** | **Reserved** | **XTL12M_EN** |

| 位 | 描述 | |
|---|---|---|
| [31:7] | **Reserved** | 保留 |
| [15:7] | **Reserved** | 保留 |
| [6] | **PD_WU_STS** | **掉电模式唤醒中断状态**<br>该位由掉电唤醒事件被置位，表示从掉电模式恢复<br>写 1 该位清零<br>**注意**：只有在 PD_WU_INT_EN (PWRCON[5]) 被置 1 的时候，该位才工作 |
| [5] | **PD_WU_INT_EN** | **掉电模式唤醒中断使能位（写保护位）**<br>0 = 掉电模式唤醒中断禁用<br>1 = 掉电模式唤醒中断使能<br>**注意1:** 当 PD_WU_STS 和 PD_WU_INT_EN 都为1时，中断将产生<br>**注意2:** 该位受保护，编程该位时，需要依次向地址0x5000_0100写入“59h”, “16h”, “88h”<br>该操作参考寄存器REGWRPROT(地址GCR_BA+0x100) |
| [4] | **Reserved** | 保留 |
| [3] | **OSC10K_EN** | **内部10 KHz低速RC振荡器(LIRC)使能位（写保护）**<br>0 = 禁用10 kHz 内部低速 RC 振荡器(LIRC)<br>1 = 使能10 kHz 内部低速 RC 振荡器(LIRC)<br>**注意:** 该位受保护，编程该位时，需要依次向地址0x5000_0100写入“59h”, “16h”, “88h”<br>该操作参考寄存器REGWRPROT(地址GCR_BA+0x100) |
| [2] | **OSC22M_EN** | **22.1184 MHz 内部高速RC 振荡器 (HIRC)使能位 (写保护)**<br>0 = 禁用22.1184 MHz 内部高速RC振荡器(HIRC)<br>1 = 使能22.1184 MHz 内部高速RC振荡器(HIRC)<br>**注意:** 该位受保护，编程该位时，需要依次向地址0x5000_0100写入“59h”, “16h”, “88h”<br>该操作参考寄存器REGWRPROT(地址GCR_BA+0x100) |

| 位 | 描述 | |
|---|---|---|
| [1] | **Reserved** | 保留 |
| [0] | **XTL12M_EN** | **4~24 MHz 外部高速晶振(HXT)使能位（写保护）**<br>该位的默认值由 flash 控制器的用户配置寄存器CONFIG0 [26:24] 设置。当默认的时钟源为外部 4~24 MHz 高速晶振时，该位自动置 1。<br>0 = 禁用4 ~ 24 MHz 外部高速晶振(HXT)<br>1 = 使能4~24 MHz 外部高速晶振(HXT)<br>**注意:** 该位受保护，编程该位时，需要依次向地址0x5000_0100写入“59h”, “16h”, “88h”<br>该操作参考寄存器REGWRPROT(地址GCR_BA+0x100) |

| 芯片工作模式 | SLEEPDEEP (SCR[2]) | CPU 运行 WFI 指令 | 禁用时钟 |
|---|---|---|---|
| 正常运行模式 | 0 | NO | 通过控制寄存器禁用所有时钟 |
| IDLE 模式<br>(CPU 进入休眠模式) | 0 | YES | 仅禁用 CPU 时钟 |
| 掉电模式<br>(CPU 进入深度休眠模式) | 1 | YES | 大部分时钟被禁用，仅10 kHz 及选它为时钟源的WDT外设时钟可运行。 |

表 6-6 芯片空闲/掉电模式控制表

## AHB设备时钟使能控制寄存器（AHBCLK）

该寄存器各位用于使能/禁用系统时钟，AHB总线设备时钟

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **AHBCLK** | CLK_BA+0x04 | R/W | AHB设备时钟使能控制寄存器 | 0x0000_0005 |

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

| 位 | 描述 | |
|---|---|---|
| [31:5] | **Reserved** | 保留 |
| [4] | **HDIV_EN** | **硬件除法器控制器时钟使能位**<br>0 =禁用硬件除法器外围时钟<br>1 =使能硬件除法器外围时钟 |
| [3:0] | **Reserved** | 保留 |

## APB设备时钟使能控制寄存器（APBCLK）

该寄存器各位用于使能/禁用外设控制器时钟

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **APBCLK** | CLK_BA+0x08 | R/W | APB设备时钟使能控制寄存器 | 0x0000_0001 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | OPA_EN | EADC_EN | ECAP1_EN | ECAP0_EN | Reserved | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| Reserved | ACMP_EN | EPWM1_EN | EPWM0_EN | BPWM0_EN | Reserved | UART1_EN | UART0_EN |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| Reserved | SPI2_EN | SPI1_EN | SPI0_EN | Reserved | | | I2C0_EN |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Reserved | FDIV_EN | TMR3_EN | TMR2_EN | TMR1_EN | TMR0_EN | Reserved | WDT_EN |

| 位 | 描述 | |
|---|---|---|
| [31:26] | **Reserved** | 保留 |
| [29] | **OPA_EN** | **OPA0 和 OPA1时钟使能位**<br>0 =禁用 OPA0 和 OPA1 时钟<br>1 =使能 OPA0 和 OPA1 时钟 |
| [28] | **EADC_EN** | **EADC时钟使能位**<br>0 = 禁用EADC 时钟<br>1 = 使能EADC 时钟 |
| [27] | **ECAP1_EN** | **增强型输入捕获1 时钟使能位**<br>0 =禁用增强型输入捕获1 时钟<br>1 =使能增强型输入捕获1 时钟 |
| [26] | **ECAP0_EN** | **增强型输入捕获0 时钟使能位**<br>0 =禁用增强型输入捕获0 时钟<br>1 =使能增强型输入捕获0 时钟 |
| [25:23] | **Reserved** | 保留 |
| [23] | **Reserved** | 保留 |
| [22] | **ACMP_EN** | **模拟比较器时钟使能位**<br>0 = 禁用模拟比较器时钟<br>1 = 使能模拟比较器时钟 |
| [21] | **EPWM1_EN** | **增强型 PWM1时钟使能位**<br>0 =禁用增强型 PWM1 时钟<br>1 =使能增强型 PWM1 时钟 |

| 位 | 描述 | |
|---|---|---|
| [20] | **EPWM0_EN** | **增强型 PWM0 时钟使能位**<br>0 =禁用增强型 PWM0 时钟<br>1 =使能增强型 PWM0 时钟 |
| [19] | **BPWM0_EN** | **基础型 PWM0时钟使能位**<br>0 =禁用基础型 PWM0 时钟<br>1 =使能基础型 PWM0 时钟 |
| [18] | **Reserved** | 保留 |
| [17] | **UART1_EN** | **UART1时钟使能位**<br>0 = 禁用UART1时钟<br>1 = 使能UART1时钟 |
| [16] | **UART0_EN** | **UART0时钟使能位**<br>0 = 禁用UART0时钟<br>1 = 使能UART0时钟 |
| [15] | **Reserved** | 保留 |
| [14] | **SPI2_EN** | **SPI2时钟使能位**<br>0 = 禁用SPI2时钟<br>1 = 使能SPI2时钟 |
| [13] | **SPI1_EN** | **SPI1时钟使能位**<br>0 = 禁用SPI1时钟<br>1 = 使能SPI1时钟 |
| [12] | **SPI0_EN** | **SPI0时钟使能位**<br>0 = 禁用SPI0时钟<br>1 = 使能SPI0时钟 |
| [11:9] | **Reserved** | 保留 |
| [8] | **I2C0_EN** | **$I^2C0$时钟使能位**<br>0 = 禁用$I^2C0$时钟<br>1 = 使能$I^2C0$时钟 |
| [7] | **Reserved** | 保留 |
| [6] | **FDIV_EN** | **分频器输出时钟使能位**<br>0 =禁用分频器输出时钟<br>1 =使能分频器输出时钟 |
| [5] | **TMR3_EN** | **定时器3时钟使能位**<br>0 = 禁用定时器3时钟<br>1 = 使能定时器3时钟 |
| [4] | **TMR2_EN** | **定时器2时钟使能位**<br>0 = 禁用定时器2时钟<br>1 = 使能定时器2时钟 |

| 位 | 描述 | |
|---|---|---|
| [3] | **TMR1_EN** | **定时器1时钟使能位**<br>0 = 禁用定时器1时钟<br>1 = 使能定时器1时钟 |
| [2] | **TMR0_EN** | **定时器0时钟使能位**<br>0 = 禁用定时器0时钟<br>1 = 使能定时器0时钟 |
| [1] | **Reserved** | 保留 |
| [0] | **WDT_EN** | **看门狗定时器时钟使能位(写保护)**<br>0 = 禁用看门狗时钟<br>1 = 使能看门狗 时钟<br>**注意:** 该位受保护，编程该位时，需要依次向地址0x5000_0100写入“59h”, “16h”, “88h”<br>该操作参考寄存器REGWRPROT(地址GCR_BA+0x100) |

## 时钟状态监控寄存器（CLKSTATUS）

该寄存器各位用于监控芯片时钟源是否稳定，时钟切换是否失败

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CLKSTATUS** | CLK_BA+0x0C | R | 时钟状态监控寄存器 | 0x0000_00XX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **CLK_SW_FAIL** | **Reserved** | | **OSC22M_STB** | **OSC10K_STB** | **PLL_STB** | **Reserved** | **XTL12M_STB** |

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留 |
| [7] | **CLK_SW_FAIL** | **时钟切换失败标志（只读）**<br>0 = 时钟切换成功<br>1 = 时钟切换失败<br>注意：该位是一个标志，表达目前系统时钟源是否与用户在HCLK_S (CLKSEL[2:0])寄存器中定义相同。当用户切换系统时钟，系统时钟将保持原时钟，直到新时钟稳定。新时钟稳定后，该位用于指示是否与用户要设定的系统时钟源相匹配。 |
| [6:5] | **Reserved** | 保留 |
| [4] | **OSC22M_STB** | **内部22.1184 MHz高速RC振荡器(HIRC)时钟源稳定标志（只读）**<br>0 = 22.1184 MHz 内部高速RC 振荡器(HIRC) 时钟不稳定或者禁用<br>1 = 22.1184 MHz 内部高速RC 振荡器(HIRC) 时钟使能并稳定 |
| [3] | **OSC10K_STB** | **内部10 KHz 低速振荡器 (LIRC) 时钟源稳定标志（只读）**<br>0 = 内部10 kHz低速RC振荡器 (LIRC) 时钟不稳定或者禁用<br>1 =内部10 kHz低速RC振荡器(LIRC) 时钟使能并稳定 |
| [2] | **PLL_STB** | **内部 PLL 时钟源稳定标志(只读)**<br>0 = 内部 PLL时钟不稳定或者禁用<br>1 = 内部PLL 时钟稳定为正常模式 |
| [1] | **Reserved** | 保留 |
| [0] | **XTL12M_STB** | **外部4~24 MHz 高速晶振 (HXT) 时钟源稳定标志（只读）**<br>0 = 外部4~24 MHz高速晶振(HXT) 时钟不稳定或者禁用<br>1 = 外部4~24 MHz高速晶振(HXT) 时钟使能并稳定 |

## 时钟源选择控制寄存器0（CLKSEL0）

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CLKSEL0** | CLK_BA+0x10 | R/W | 时钟源选择控制寄存器0 | 0x0000_003X |

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

| 位 | 描述 | |
|---|---|---|
| [31:6] | **Reserved** | 保留 |
| [5:3] | **STCLK_S** | **Cortex™-M0 SysTick 时钟源选择(写保护)**<br>如果SYST_CSR[2] = 1, SysTick 时钟源来自 HCLK<br>如果SYST_CSR[2] = 0, SysTick 时钟源由STCLK_S(CLKSEL0[5:3])定义.<br>000 = 时钟源为外部 4~24 MHz 高速晶振时钟<br>001 = 保留<br>010 = 时钟源为外部 4~24 MHz高速 晶振时钟/2分频<br>011 = 时钟源为 HCLK/2分频<br>111 = 时钟源为内部 22.1184 MHz 高速振荡器时钟/2分频<br>**注意1:** 该位是受保护位。编程时，需要向地址 0x5000_0100 依次写入 “59h”, “16h”, “88h” 来解锁寄存器保护。请参考寄存器 REGWRPROT （地址：GCR_BA+0x100）。<br>**注意2:** 如果 SysTick 时钟源不是来自 HCLK (例如 SYST_CSR[2] = 0), SysTick 时钟源必须小于或等于HCLK/2 |
| [2:0] | **HCLK_S** | **HCLK时钟源选择(写保护)**<br>在时钟切换前，相关时钟源（预选和新选）必须打开。<br>在任何复位后，这 3 位的默认值将从Flash 控制器的用户配置寄存器 CFOSC (CONFIG0 [26:24]) 加载。因此默认值可能为 000b 或者 111b。<br>000 = 时钟源为外部 4~24 MHz高速 晶振时钟<br>001 = 保留<br>010 = 时钟源为 PLL 时钟<br>011 = 时钟源为内部 10 kHz 低速振荡器时钟<br>111=时钟源为内部 22.1184 MHz高速 振荡器时钟<br>注意: 该位是受保护位。编程时，需要向地址 0x5000_0100 依次写入 “59h”, “16h”, “88h” 来解锁寄存器保护。请参考寄存器 REGWRPROT （地址：GCR_BA+0x100） |

## 时钟源选择控制寄存器1（CLKSEL1）

在时钟切换之前，必须打开相关的时钟源（预选和新选）

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CLKSEL1** | CLK_BA+0x14 | R/W | 时钟源选择控制寄存器1 | 0xFFFF_FFFF |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | **UART_S** | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | **SPI2_S** | **SPI1_S** | **SPI0_S** | **Reserved** | | **WDT_S** | |

| 位 | 描述 | |
|---|---|---|
| [31:26] | **Reserved** | 保留 |
| [25:24] | **UART_S** | **UART时钟源选择**<br>00 = 时钟源为外部 4~24 MHz 高速晶振时钟<br>01 = 时钟源为 PLL 时钟<br>10 = 保留<br>11 = 时钟源为内部 22.1184 MHz 高速振荡器时钟 |
| [23:7] | **Reserved** | 保留 |
| [6] | **SPI2_S** | **SPI2时钟源选择**<br>0 =时钟源来自PLL 时钟<br>1 =时钟源来自HCLK |
| [5] | **SPI1_S** | **SPI1时钟源选择**<br>0 =时钟源来自PLL 时钟<br>1 =时钟源来自HCLK |
| [4] | **SPI0_S** | **SPI0时钟源选择**<br>0 =时钟源来自PLL 时钟<br>1 =时钟源来自HCLK |
| [3:2] | **Reserved** | 保留 |
| [1:0] | **WDT_S** | **看门狗定时器时钟源选择（写保护位）**<br>00 =时钟源为 HCLK /128 clock<br>01 =时钟源为 HCLK /512 clock<br>10 = 时钟源为 HCLK /2048 clock<br>11 = 时钟源为内部 10 kHz 低速振荡器时钟<br>注意**:** 该位是受保护位。编程时，需要向地址 0x5000_0100 依次写入 “59h”, “16h”, “88h” 来解锁寄存器保护。请参考寄存器 REGWRPROT （地址：GCR_BA+0x100。 |

## 时钟源选择控制寄存器2（CLKSEL2）

在时钟切换之前，必须打开相关的时钟源（预选和新选）

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CLKSEL2** | CLK_BA+0x1C | R/W | 时钟源选择控制寄存器2 | 0xFFFF_FFFF |

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

| 位 | 描述 | |
|---|---|---|
| [31:18] | **Reserved** | 保留 |
| [17:16] | **WWDT_S** | **窗口看门狗时钟源选择**<br>00 = 保留<br>01 = 保留<br>10 = 时钟源为HCLK/2048 时钟.<br>11 = 时钟源为内部10 kHz低速振荡器时钟 |
| [15:4] | **Reserved** | 保留 |
| [3:2] | **FRQDIV_S** | **时钟分频器时钟源选择**<br>00 = 时钟源为外部 4~24 MHz高速 晶振时钟<br>01 = 保留<br>10 = 时钟源为 HCLK<br>11 = 时钟源为内部 22.1184 MHz 高速振荡器时钟 |
| [1:0] | **Reserved** | 保留 |

## 时钟分频寄存器(CLKDIV)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CLKDIV** | CLK_BA+0x18 | R/W | 时钟分频寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **EADC_N** | **EADC 时钟分频**<br>EADC 时钟频率 = (EADC 时钟源频率) / (EADC_N + 1). |
| [15:12] | **Reserved** | 保留 |
| [11:8] | **UART_N** | **UART时钟分频**<br>UART时钟频率= (UART时钟源频率) / (UART_N + 1). |
| [7:4] | **Reserved** | 保留 |
| [3:0] | **HCLK_N** | **HCLK时钟分频**<br>HCLK时钟频率= (HCLK时钟源频率) / (HCLK_N + 1). |

## PLL 控制寄存器（PLLCON）

PLL 的参考时钟源来自外部 4~24 MHz高速晶振时钟输入或者内部 22.1184 MHz 高速振荡器。该寄存器用于控制 PLL 的输出频率和 PLL 的工作模式。

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PLLCON** | CLK_BA+0x20 | R/W | PLL 控制寄存器 | 0x0005_C22E |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | **FCO_SEL** | **PLL_SRC** | **OE** | **BP** | **PD** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **OUT_DV** | | **IN_DV** | | | | | **FB_DV** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **FB_DV** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:20] | **Reserved** | 保留 |
| [20] | **FCO_SEL** | **PLL FCO 选择**<br>0 = 当FCO频率范围在100MHz 和 200MHz之间时，该位应该清0。<br>1 = 当FCO频率范围在200MHz 和 500MHz之间时，该位应该置1。 |
| [19] | **PLL_SRC** | **PLL 时钟源选择**<br>0 = PLL 时钟源为外部 4~24 MHz 高速晶振<br>1 = PLL 时钟源为内部 22.1184 MHz 高速振荡器 |
| [18] | **OE** | **PLL OE (FOUT 使能) 管脚控制**<br>0 = PLL FOUT 使能<br>1 = PLL FOUT 固定为低 |
| [17] | **BP** | **PLL 旁路控制**<br>0 = PLL 正常模式（默认）<br>1 = PLL 时钟输出与PLL源时钟输入相同 |
| [16] | **PD** | **掉电模式**<br>0 = PLL 正常模式<br>1 = PLL 进入掉电模式（默认） |
| [15:14] | **OUT_DV** | **PLL 输出分频控制位**<br>参考下表公式 |
| [13:9] | **IN_DV** | **PLL 输入分频控制位**<br>参考下表公式 |
| [8:0] | **FB_DV** | **PLL 反馈分频控制位**<br>参考下表公式 |

**输出时钟频率设置**

$$FOUT = FIN \times \frac{NF}{NR} \times \frac{1}{NO}$$

约束条件:

1. $3.2MHz < FIN < 150MHz$
2. $800KHz < Fref = \frac{FIN}{2 \times NR} < 8MHz$
3. $200MHz < FCO = Fref \times 2 \times NF = FIN \times \frac{NF}{NR} < 500MHz, (\text{FCO_SEL} = 1)$

   $100MHz < FCO = Fref \times 2 \times NF = FIN \times \frac{NF}{NR} < 200MHz, (\text{FCO_SEL} = 0)$

| 符合 | 描述 |
|---|---|
| FOUT | 输出时钟频率 |
| FIN | 输入（参考）时钟频率 |
| NR | 输入分频 (IN_DV + 2) |
| NF | 反馈分频 (FB_DV + 2) |
| NO | OUT_DV = “00” : NO = 1<br>OUT_DV = “01” : NO = 2<br>OUT_DV = “10” : NO = 2<br>OUT_DV = “11” : NO = 4 |

**默认频率设置**

默认值: 0xC22E
FIN = 12 MHz
NR = (1+2) = 3
NF = (46+2) = 48
NO = 4
FOUT = 12/4 x 48 x 1/3 = 48MHz

## 频率分频器控制寄存器(FRQDIV)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **FRQDIV** | CLK_BA+0x24 | R/W | 频率分频器控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | **DIV1** | **DIVIDER_EN** | **FSEL** | | | |

| 位 | 描述 | |
|---|---|---|
| [31:6] | **Reserved** | 保留 |
| [5] | **DIV1** | **分频器1使能位**<br>0 = 分频器输出的时钟来自由FSEL分频的时钟源<br>1 = 分频器输出的时钟来自时钟源 |
| [4] | **DIVIDER_EN** | **频率分频器使能位**<br>0 = 禁用频率分频器<br>1 = 使能频率分频器 |
| [3:0] | **FSEL** | **分频器输出频率选择位**<br>输出频率的公式:<br>$F_{CLKO} = F_{FRQDIV_CLK}/2^{(N+1)}$,.<br>$F_{FRQDIV_CLK}$为输入时钟频率<br>$F_{CLKO}$ t为分频器输出时钟频率<br>N为FSEL[3:0] 的4位值。 |

## 6.4 Flash存储控制器 (FMC)

### 6.4.1 概述

NuMicro® M0519系列 具有128/64 K 字节的片上FLASH，用于存储应用程序（APROM）和数据flash，用户通过另外的8K字节LDROM烧录启动代码后可以用ISP更新这些FLASH. 在系统编程(ISP) 用户可以直接更新已经焊接在PCB板上芯片的程序。上电后，Config0的启动选择(CBS)决定Cortex-M0 CPU从APROM或LDROM读取代码。NuMicro® M0519系列也提供了数据 flash，用来存储用户想要在芯片掉电前的数据。对于128 KB APROM设备数据flash与应用程序APROM是共享这128KB的空间，在Config1中配置数据flash的起始地址。对于64KB APROM设备数据flash是固定大小为4KB。

### 6.4.2 特征

- 可运行达72MHz，连续地址访问零等待状态时最高可达50MHz
- 所有嵌入Flash支持512字节页擦除
- 128/64 KB应用程序存储空间(APROM)
- 8KB在系统编程 (ISP) 加载程序空间(LDROM)
- 64KB APROM设备拥有4KB数据flash
- 128KB APROM设备数据flash可配置
- 8字节用户配置块用来控制系统初始化
- 支持在系统编程（ISP）和在应用编程（IAP）更新片上Flash

### 6.4.3 框图

FLASH存储器控制器(FMC)包括AHB从接口，ISP 控制逻辑，烧写器接口和FLASH宏接口时序控制逻辑。FLASH存储器控制器框图如下图所示：

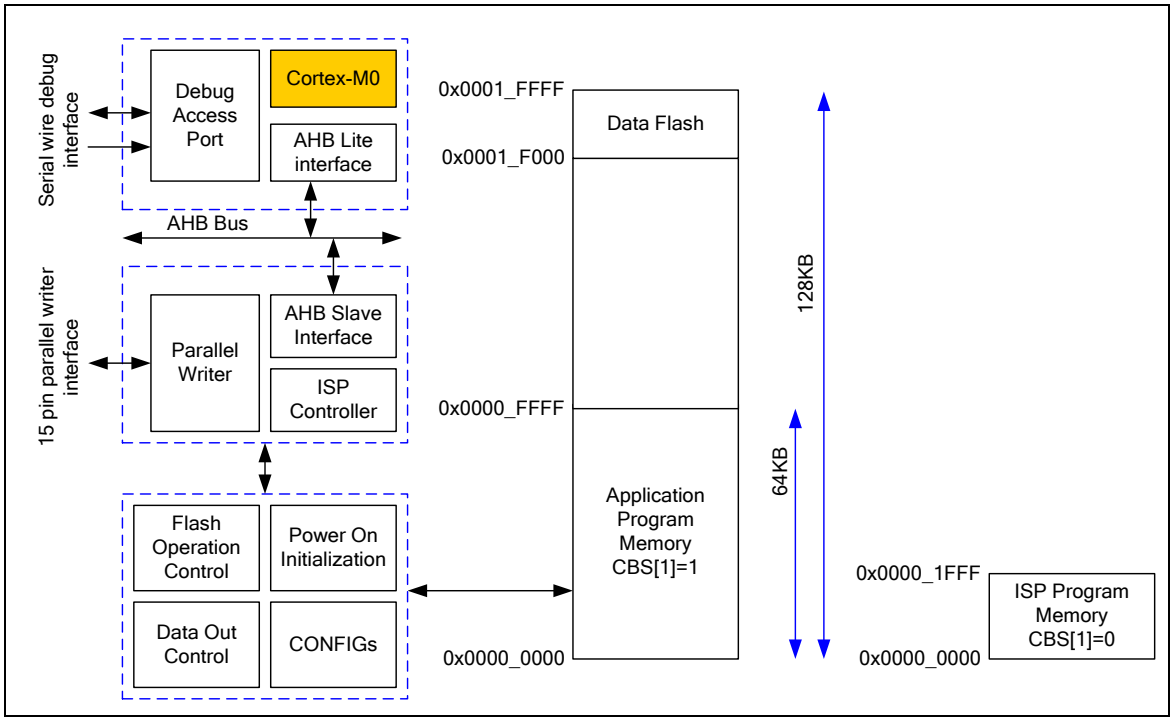



图 6-9 存储器控制框图

### 6.4.4 功能描述

FMC功能包括存储器结构、启动选择、IAP、ISP和ISP流程

#### *6.4.4.1 存储器结构*

NuMicro® M0519系列的flash存储器由程序存储器(APROM)，数据FLASH，ISP载入程序存储器(LDROM)，和用户配置区组成。

程序存储器是用户应用的主要存储器，叫做APROM，用户可以将他们的应用程序写到APROM并设置系统从APROM启动。

设计ISP 装载程序存储器是用来实现在系统编程（ISP）功能。LDROM独立于APROM，也能设置从LDROM启动，因此当APROM的代码损坏时用户可以用LDROM启动系统，避免系统启动失败。

数据FLASH是给用户存储数据的，它可以通过ISP读取或存储器读取，并且通过ISP编程。每个擦除单元是512字节。对于128 KB APROM设备数据flash与应用程序APROM是共享这128KB的空间，如果使能数据Flash位 DFEN (Config0[0])，数据Flash起始地址由DFBADR (Config1[19:0])决定，其大小为0x20000 – DFBADR，同时APROM的大小为128 KB – 数据Flash大小。对于64KB APROM设备数据Flash是固定大小为4KB，起始地址固定在0x0001_F000。

用户配置提供了几个字节来控制系统逻辑，如FLASH安全锁，启动选择，欠压电平，数据FLASH基地址等。用户配置如同是上电设置的保险，上电时用户配置从FLASH存储器装载到它相应的控制寄存器。

在NuMicro® M0519系列中，flash存储器组织跟系统存储映射不同。当用户用ISP命令去读、编程、或者擦除flash，存储器组织被用到。在CPU访问FLASH存储器获取代码或数据的时候，系统存储器映射被用到。例如，当系统设置为从LDROM启动（CBS = 01b）,CPU将从LDROM的0x0 ~ 0x1FFF取代码。但是，如果用户想用ISP读LDROM，仍然要从LDROM的地址0x0010_0000 ~ 0x0010_1FFF去读。

表6-7显示了64/128KB设备的APROM,LDROM, 数据Flash和用户配置 的地址映像信息

| 模块名称 | 设备类型 | 大小 | | 起始地址 | 结束地址 |
|---|---|---|---|---|---|
| APROM | 64 KB | 64 KB | | 0x0000_0000 | 0x0000_FFFF |
| | 128 KB | 数据 Flash 使能 | 128 KB - Data Flash Size | 0x0000_0000 | 0x20000 – (DFBADR – 1) |
| | | 数据 Flash 禁用 | 128 KB | 0x0000_0000 | 0x0001_FFFF |
| Data Flash | 64 KB | 4 KB | | 0x0001_F000 | 0x0001_FFFF |
| | 128 KB | 数据 Flash 使能 | 0x20000-DFBADR | DFBADR | |
| | | 数据 Flash 禁用 | 0 KB | N/A | N/A |
| LDROM | 64/128 KB | 8 KB | | 0x0010_0000 | 0x0010_1FFF |
| User Configuration | 64/128 KB | 2 words | | 0x0030_0000 | 0x0030_0007 |

**注:**一页擦除为512字节，DFBADR的bit0至bit8应该一直保持为0

表 6-7 存储器地址映像

Flash存储器组织结构如图 6-10:

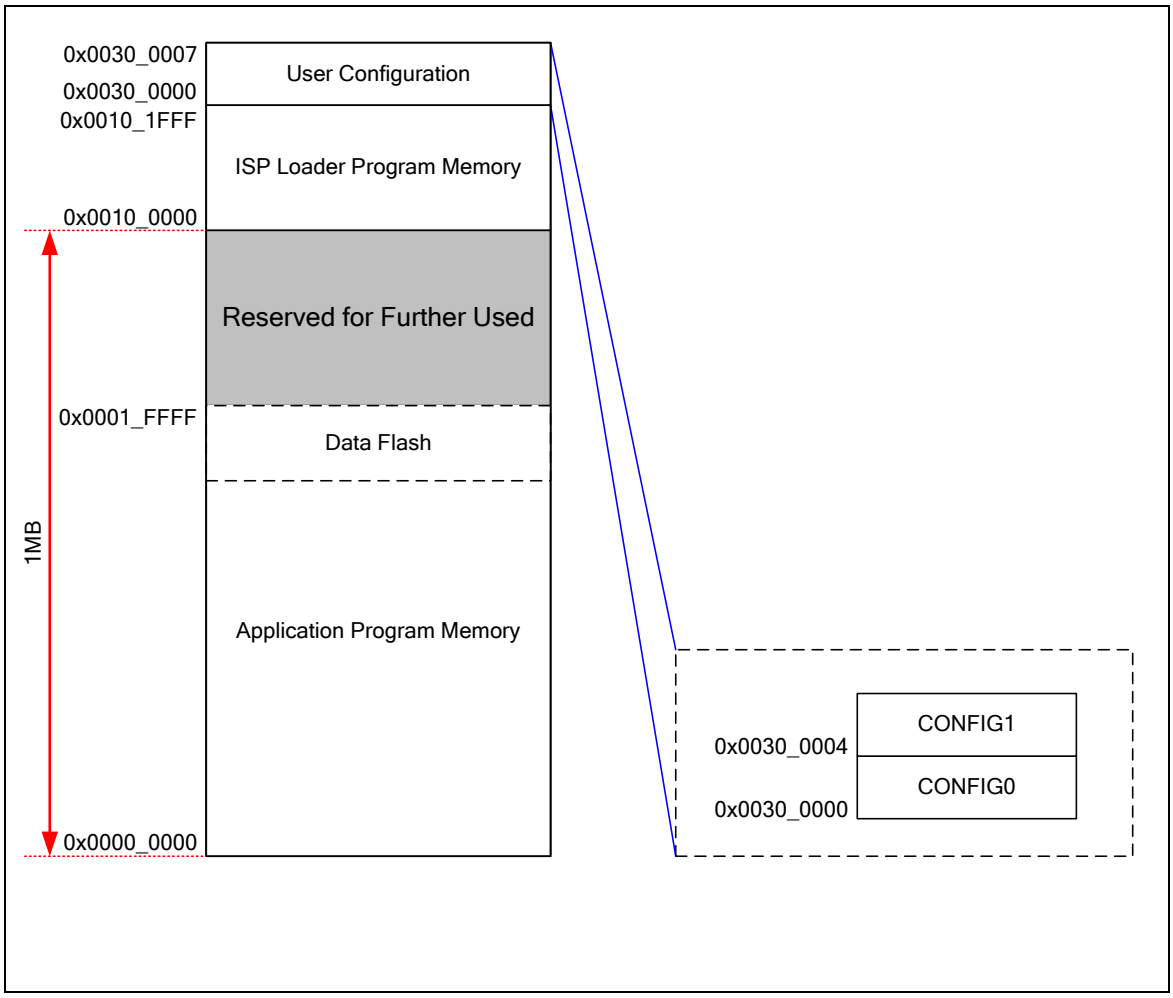



图 6-10 Flash存储器组织结构

#### 6.4.4.1 启动选择

NuMicro® M0519系列给用户提供了四个启动源选择，包括：带有IAP 功能的LDROM、没有IAP 功能的LDROM、带有IAP 功能的APROM和没有IAP 功能的APROM。用户可以通过设置CBS (CONFIG0[7:6])来选择启动源和系统存储器映射。

| CBS | 启动选择/系统存储器映射 | 向量表重映射 |
|---|---|---|
| 00b | 带有IAP 功能的LDROM模式 | 支持 |
| 01b | 没有IAP 功能的LDROM模式 | 不支持 |
| 10b | 带有IAP 功能的APROM模式 | 支持 |
| 11b | 没有IAP 功能的APROM模式 | 不支持 |

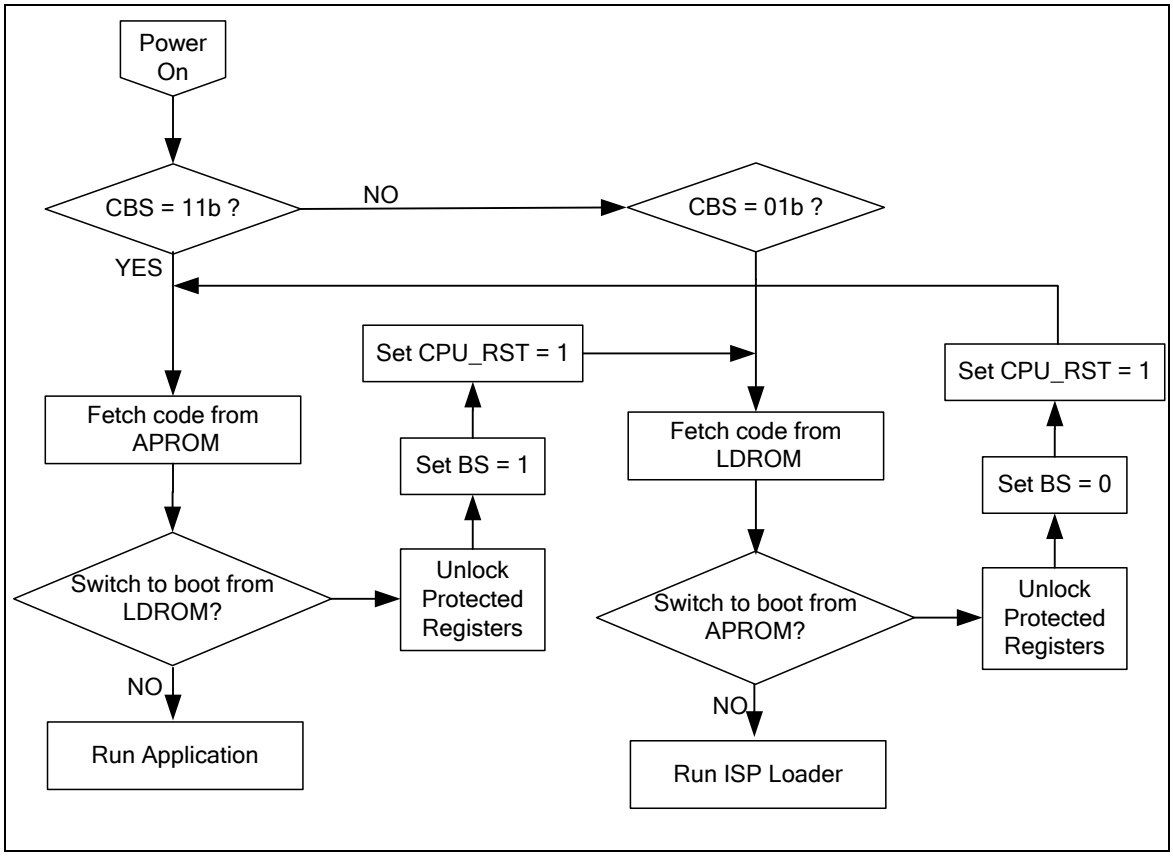



图 6-11 BS位启动选择流程示例

#### 6.4.4.2 *在应用编程 (IAP)*

NuMicro® M0519系列提供了在应用编程（IAP）功能，用户无需复位系统就可以在APROM和LDROM 之间切换执行代码。用户可以通过设置CONFIG0 (CBS[1:0]) 为 10b 或 00b并重启来使能IAP功能。

在芯片从APROM启动使能IAP功能(CBS[1:0] = 10b)的情况下，代码的执行地址范围包括所有的APROM和LDROM。APROM 地址空间保持不变，但8 KB LDROM被映射到0x0010_0000~0x0010_1FFF。

在芯片从LDROM启动使能IAP功能(CBS[1:0] = 00b)的情况下，代码的执行地址范围包括所有的LDROM，和除了第一页的APROM。执行代码不能访问APROM的第一页，因为第一页的执行地址默认变成LDROM第一页的镜像。同时8 KB LDROM也被映射到0x0010_0000~ 0x0010_1FFF。

IAP功能使能时，地址空间映像请参考图 6-12

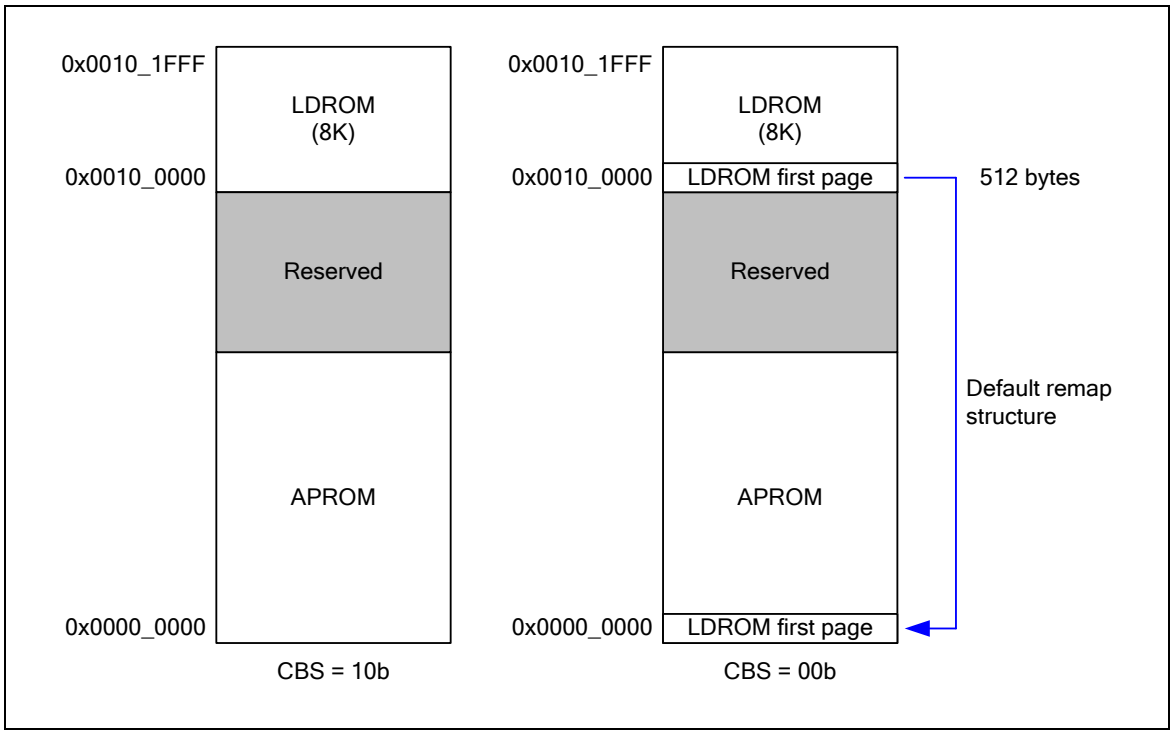



图 6-12 IAP使能时代码的执行范围

当芯片使能了IAP功能，任何时候代码可执行范围内的任何其他页都可被镜像到执行代码的第一页(0x0000_0000~0x0000_01FF)。用户可以通过填目标重映像地址到ISPADR改变第一可执行页的重新映像地址然后执行ISP向量页重映射命令，通过读ISPSTA 寄存器中VECMAP域，用户可以检查改变是否成功。

#### 6.4.4.3 *在系统编程(ISP)*

NuMicro® M0519系列提供在系统编程(ISP)功能，允许用户通过单机ISP固件更新程序存储器. 提供8KB 加载程序内存(LDROM)用于存储ISP固件. 用户可以通过设置CBS[1](Config0[7])位来选择从APROM还是从LDROM取指令来运行.

Config0中的CBS除了设置从APROM或者LDROM启动之外还用于启动后控制系统存储器映射。当CBS[0] = 1 和 CBS[1] = 1为从APROM启动， APROM的应用程序不能通过读储存器的方式来访问

LDROM.同样当CBS[0] = 1 和 CBS[1] = 0为LDROM启动，LDROM的程序也不能通过读储存器的方式来访问APROM。图 6-13 表示当系统从APROM和LDROM启动时的储存器映射。

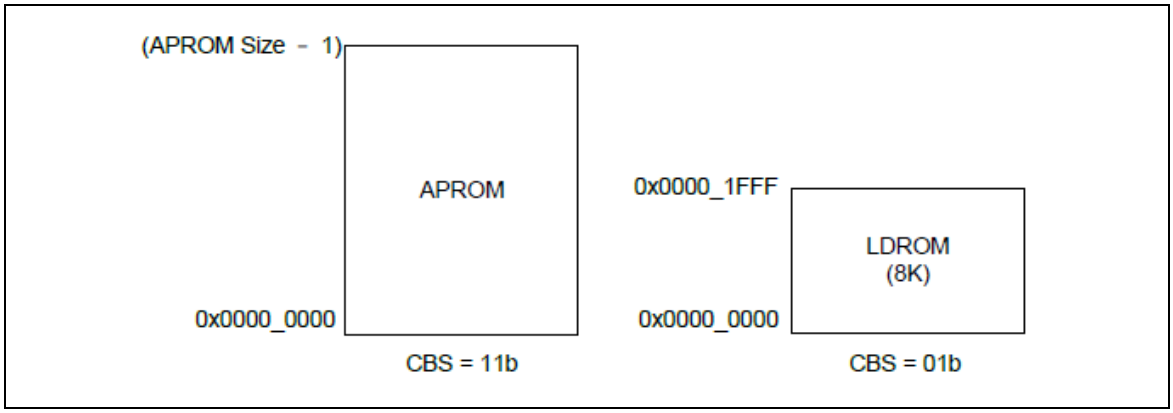



图 6-13 APROM 和 LDROM 启动的程序执行范围

NuMicro® M0519系列支持ISP模式，当烧写失败时设备可以在软件控制重新烧写，避免系统崩溃的风险。因此，更新应用固件的功能,可以让应用更为广泛

ISP可以更新板上的系统固件，各种外设接口使得LDROM固件获取新的代码更容易。执行ISP最常用的方法是在LDROM中的固件通过UART更新，PC一般都是通过串口传输新的APROM代码. ISP下载程序接收后， 通过ISP命令，重新对APROM编程

| ISP 命令 | ISPCMD | ISPADR | ISPDAT |
|---|---|---|---|
| FLASH 页擦除 | 0x22 | FLASH存储器结构的有效地址 ，它必须由512字节/页排列 | N/A |
| FLASH 写 | 0x21 | FLASH存储器结构的有效地址 | 写数据 |
| FLASH 读 | 0x00 | FLASH存储器结构的有效地址 | 返回数据 |
| 读公司 ID | 0x0B | 0x0000_0000 | 0x0000_00DA |
| 读UID | 0x04 | 0x0000_0000 | UID字 0 |
| | | 0x0000_0004 | UID字1 |
| | | 0x0000_0008 | UID字2 |
| 向量页重映射 | 0x2E | APROM 、 LDROM或boot loader 的有效地址，它必须由512字节/页排列 | N/A |

表 6-8 ISP 命令表

#### 6.4.4.4 ISP 流程

NuMicro® M0519系列可以通过配置用户配置位(CBS)选择从APROM 还是LDROM启动。 如果用户想要在没有IAP功能APROM模式下更新应用程序，用户可以置BS (ISPCON[1])位为1然后软件复位让芯片从LDROM启动。开始ISP的第一步是向ISPEN位写1。在向ISPCON寄存器写入前必须要软件向REGWRPROT依次写入0x59, 0x16 和 0x88。这么做是为了保护flash存储器在上电/断点期间受到无意识写入的破坏。

一旦ISPCON寄存器被设置成功，用户可以通过设置ISPCMD来擦除、读或者写。基于flash内存的组织结构，目标flash内存来设置ISPADR。ISPDAT可以用于设置数据写入或者返回 ISPCMD读的数据。

最后设置ISPTRG控制寄存器的ISPGO位来执行相应的ISP功能。当ISP功能完成后ISPGO位将自动清0。为了确保在CPU运行之前ISP功能已经完成，在ISPGO设置之后，应该用ISB指令。

ISP完成后几个错误条件需要检查。如果出现错误，ISP操作就不会开始并且ISP失败标志ISPFF被置位。ISPFF标志只能由软件清除。当ISPFF位保持1下一个ISP程序仍然可以开始，因此，建议在ISP操作完成后检查ISPFF位，如果被置1要清0。

当ISPGO位被置1，CPU 将一直等到ISP操作到完成，这个时期内外围设备跟通常一样保持运行。任何中断请求都不会响应直到CPU完成ISP操作。当ISP操作完成ISPGO位将被硬件自动清0。用户可以通过ISPGO位来检查ISP操作是否完成。

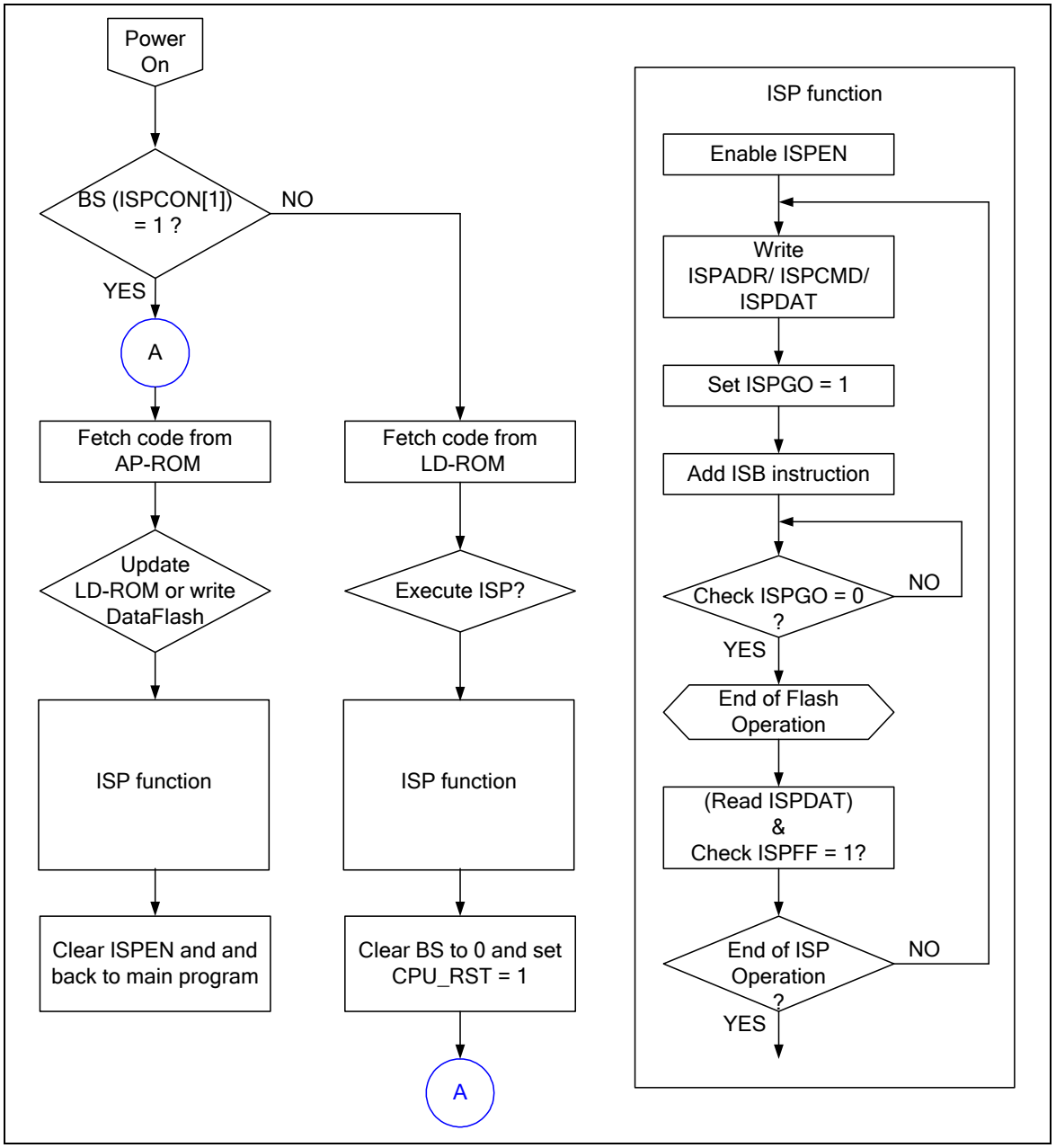



图 6-14 ISP流程示例

#### 6.4.4.5 数据 Flash

NuMicro® M0519为用户提供数据FLASH用于存储数据。通过ISP寄存器读/写。擦除单位为512字节。 若要改变一个字，需要先把所有128个字拷贝到另外页或SRAM中。

对于64KB APROM设备其数据Flash大小固定为4KB。起始地址固定在0x0001_F000。对于128KB APROM设备其数据Flash是与应用程序共享的，大小用户可配置。当DFEN (Config0[0]) 被设置 为1，就没有数据Flash，128KB全部用作APROM。当DFEN (Config0[0]) 被设置为0，数据Flash是与应用程序共享的。起始地址由DFBADR (Config1[19:0])决定。在这个配置下，应用程序存储器大小为(128-0.5*N)KB，数据Flash大小为0.5*N KB。

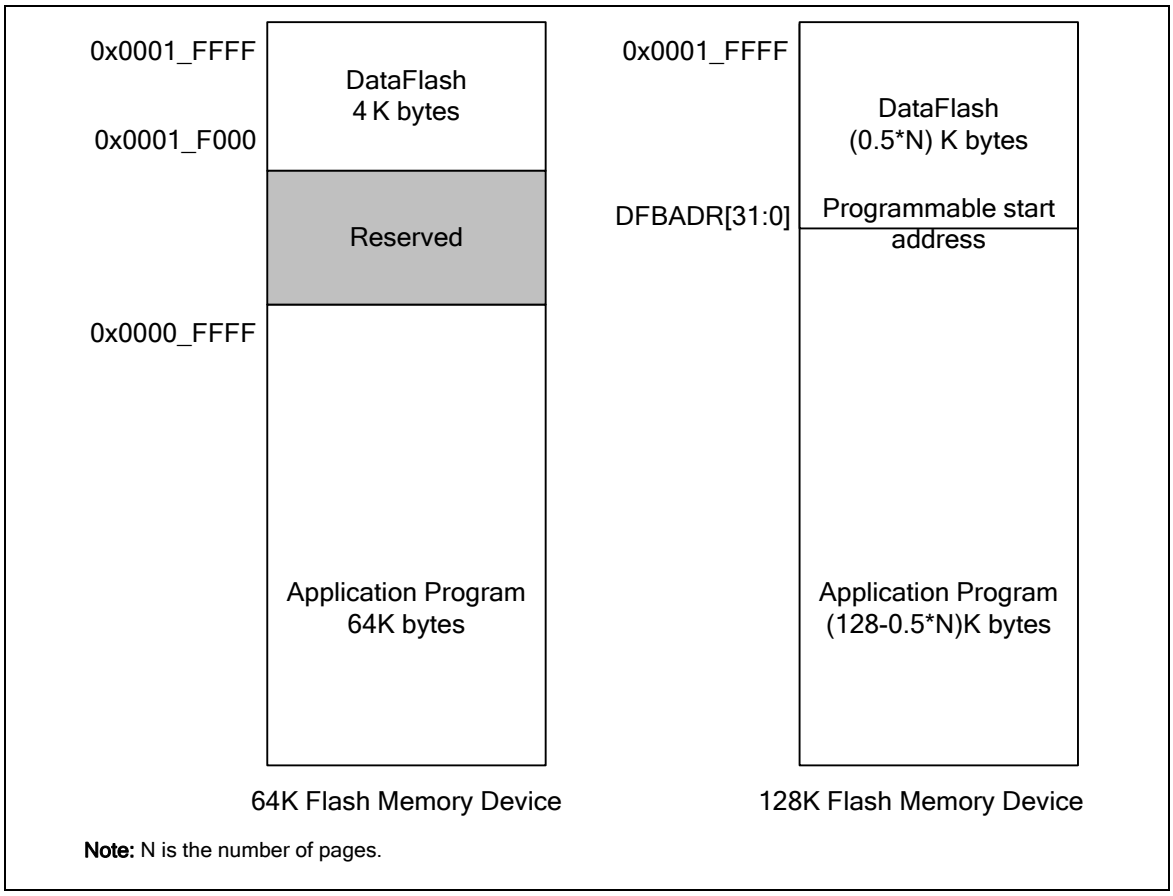



图 6-15 Flash 存储器结构 (DFVSEN = 1)

#### 6.4.4.6 用户配置

用户配置是内部可编程的配置区域，用于启动选项。用户配置位于Flash存储器组织的0x300000，并有两个32位字节。用户配置所有的更改将在系统重启后生效。

## Config0 (地址 = 0x0030_0000)

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CWDTEN | CWDTPDEN | Reserved | | | CFOSC | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CBODEN | CBOV1 | CBOV0 | CBORST | Reserved | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | CHZ_BPWM | CHZ_Odd1 | CHZ_Even1 | CHZ_Odd0 | CHZ_Even0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CBS | | Reserved | | | | LOCK | DFEN |

| Config0 位 | 描述 | |
|---|---|---|
| [31] | **CWDTEN** | **看门狗使能位**<br>0 =看门狗定时器使能，在芯片上电后OSC10K作为看门狗定时器时钟源<br>1 =芯片上电后禁用看门狗定时器 |
| [30] | **CWDTPDEN** | **看门狗时钟掉电使能位**<br>0 = OSC10K 看门狗定时器时钟源一直强制使能<br>1 = OSC10K 看门狗定时器时钟源在掉电模式下由OSC10K_EN (PWRCON[3]) 控制<br>**注:** 此位只有在CWDTEN 为0时有效 |
| [29:27] | **Reserved** | 保留 |
| [26:24] | **CFOSC** | **复位后CPU时钟源选择**<br>000 =外部 4~24 MHz 高速晶振时钟(HXT)<br>111 =内部 22.1184 MHz 高速振荡器时钟(HIRC)<br>其他 =保留<br>复位后，CFOSC的值将被加载到系统寄存器HCLK_S (CLKSEL0 [2:0]) |
| [23] | **CBODEN** | **欠压检测使能**<br>0= 上电后，使能欠压检测<br>1= 上电后，禁用欠压检测 |
| [22:21] | **CBOV1-0** | **欠压电压选择**<br>00 = 2.2 V<br>01 = 2.7 V<br>10 = 3.7 V<br>11 = 4.4 V |
| [20] | **CBORST** | **欠压复位使能**<br>0 = 上电后，使能欠压复位<br>1 = 上电后，禁用欠压复位 |
| [19:13] | **Reserved** | 保留 |
| [12] | **CHZ_BPWM** | **基本 PWM0 端口三态驱动控制**<br>0 =基本PWM端口驱动模式由GPIO模式寄存器(Px_PMD)控制<br>1 = 基本PWM端口驱动模式被强制设置成三态（高阻态）<br>**注：** 任何复位后该位的值会被加载到GPIO控制位HZ_BPWM (PWMPOEN[4])中 |

| | | |
|---|---|---|
| [11] | **CHZ_Odd1** | **增强型 PWM 单元1 奇端口三态驱动控制**<br>0 =增强型PWM单元1奇端口驱动模式由GPIO模式寄存器(Px_PMD)控制<br>1 = 增强型PWM单元1奇端口驱动模式被强制设置成三态（高阻态）<br>**注：**任何复位后该位的值会被加载到GPIO控制位HZ_Odd1 (PWMPOEN[3])中 |
| [10] | **CHZ_Even1** | **增强型 PWM 单元1 偶端口三态驱动控制**<br>0 =增强型PWM单元1偶端口驱动模式由GPIO模式寄存器(Px_PMD)控制<br>1 = 增强型PWM单元1偶端口驱动模式被强制设置成三态（高阻态）<br>**注：**任何复位后该位的值会被加载到GPIO控制位HZ_Even1 (PWMPOEN[2])中 |
| [9] | **CHZ_Odd0** | **增强型 PWM 单元0 奇端口三态驱动控制**<br>0 =增强型PWM单元0奇端口驱动模式由GPIO模式寄存器(Px_PMD)控制<br>1 = 增强型PWM单元0奇端口驱动模式被强制设置成三态（高阻态）<br>**注：**任何复位后该位的值会被加载到GPIO控制位HZ_Odd0 (PWMPOEN[1])中 |
| [8] | **CHZ_Even0** | **增强型 PWM 单元0 偶端口三态驱动控制**<br>0 =增强型PWM单元0偶端口驱动模式由GPIO模式寄存器(Px_PMD)控制<br>1 = 增强型PWM单元0偶端口驱动模式被强制设置成三态（高阻态）<br>**注：**任何复位后该位的值会被加载到GPIO控制位HZ_Even0 (PWMPOEN[0])中 |
| [7:6] | **CBS** | **芯片启动选择**<br>当CBS[0] = 0时IAP功能被使能，APROM的基地址是0X0、LDROM基地址是0x100000。<br>IAP功能意味着系统不用重启，CPU就可以执行和访问APROM 、LDROM。<br>00=由LDROM启动带IAP功能<br>01=由LDROM启动不带IAP功能<br>10=由APROM启动带IAP功能<br>11=由APROM启动不带IAP功能<br>**注：**<br>BS (ISPCON[1])是用在没有IAP模式下选择启动区域。<br>VECMAP (ISPSTA[20:9])是用在带有IAP模式下将选择的区域重映射到0x0~0x1ff |
| [5:2] | **Reserved** | 保留 |
| [1] | **LOCK** | **安全加密**<br>0 = 加密FLASH数据<br>1 = 解除Flash 数据加密<br>当FLASH数据被加密，仅有设备ID， Config0 和Config1 可被烧写器和ICP通过串口调试接口读取. 其他数据锁定为0xFFFFFFFF. 无论数据是否锁定，ISP 都可以读取 。<br>用户需要用ICP烧录工具擦除整个芯片或者用ISP擦除用户配置来解锁 |
| [0] | **DFEN** | **数据 Flash 使能位 (仅 128 KB APROM 设备)**<br>0 = 数据 Flash 使能<br>1 = 数据 Flash 禁用 |

## Config1 (地址 = 0x0030_0004)

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

| Config1 位 | 描述 | |
|---|---|---|
| [31:20] | **Reserved** | 保留 |
| [19:0] | **DFBADR** | **数据 Flash 基地址 (仅 128 KB APROM 设备)**<br>对于128KB APROM设备，数据flash基地址是由用户定义的。由于片上flash页擦除是以512字节为一页，所有bit8-0强制为0。这个配置仅对于128KB flash设备有效。 |

### 6.4.5 寄存器映射

**R**: 只读, **W**: 只写, **R/W**: 读/写

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **FMC 基地址:**<br>**FMC_BA = 0x5000_C000** | | | | |
| **ISPCON** | FMC_BA+0x00 | R/W | ISP 控制寄存器 | 0x0000_000X |
| **ISPADR** | FMC_BA+0x04 | R/W | ISP 地址寄存器 | 0x0000_0000 |
| **ISPDAT** | FMC_BA+0x08 | R/W | ISP 数据寄存器 | 0x0000_0000 |
| **ISPCMD** | FMC_BA+0x0C | R/W | ISP 命令寄存器 | 0x0000_0000 |
| **ISPTRG** | FMC_BA+0x10 | R/W | ISP 触发寄存器 | 0x0000_0000 |
| **DFBADR** | FMC_BA+0x14 | R | 数据Flash起始地址 | 0x000X_XXXX |
| **FATCON** | FMC_BA+0x18 | R/W | Flash访问时间控制寄存器 | 0x0000_0000 |
| **ISPSTA** | FMC_BA+0x40 | R/W | ISP 状态寄存器 | 0x00XX_XX0X |

### 6.4.6 寄存器描述

**ISP 控制寄存器 (ISPCON)**

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ISPCON** | FMC_BA+0x00 | R/W | ISP 控制寄存器 | 0x0000_000X |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | **ISPFF** | **LDUEN** | **CFGUEN** | **APUEN** | **Reserved** | **BS** | **ISPEN** |

| 位 | 描述 | |
|---|---|---|
| [31:7] | **Reserved** | 保留 |
| [6] | **ISPFF** | **ISP失败标志 (写保护位)**<br>当ISP满足下列任一条件时，该位由硬件置位:<br>(1) APUEN等于0时，APROM 写APROM<br>(2) LDUEN等于0时，LDROM 写LDROM<br>(3) CFGUEN等于0时，CONFIG 被擦除或编程<br>(4)ICE连接时且处在LOCK模式下，执行页擦除命令<br>(5)在欠压情况下，执行擦除或是编程命令<br>(6) 定义地址无效，如超过正常范围<br>(7)无效的ISP命令<br>该位写 1 清除 |
| [5] | **LDUEN** | **LDROM更新使能 位(写保护位)**<br>0 = 禁止LDROM更新<br>1 =当芯片在 APROM 中运行时，LDROM 可以更新。 |
| [4] | **CFGUEN** | **配置位更新使能位 (写保护位)**<br>0 = 禁止配置位更新<br>1 = 使能配置位 更新 |
| [3] | **APUEN** | **APROM 更新使能 (写保护位)**<br>0 = 当芯片在APROM中运行时APROM 不能被更新.<br>1 = 当芯片在APROM中运行时APROM 可以被更新. |
| [2] | **Reserved** | 保留 |

| | | |
|---|---|---|
| [1] | **BS** | **启动选择 (写保护位)**<br>置位/清零该位选择下次是由LDROM启动还是由APROM启动, 该位也可作为MCU启动状态的标志, 用于检查芯片是由LDROM还是APROM启动的. 该位在任何复位时（除了CPU 复位 (RSTS_CPU 为 1) 或系统复位 (RSTS_SYS)）都会被初始化为 Config0 的 CBS位的反转值<br>0 = 由APROM启动<br>1 = 由LDROM启动 |
| [0] | **ISPEN** | **ISP 使能位(写保护位)**<br>ISP 使能位，设置该位可以使能ISP功能.<br>0 = 禁用 ISP 功能<br>1 = 使能 ISP 功能 |

## ISP 地址寄存器 (ISPADR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ISPADR** | FMC_BA+0x04 | R/W | ISP 地址寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **ISPADR** | **ISP 地址**<br>NuMicro® M0519系列内嵌最大32Kx32 (128 KB) Flash, 只支持字编程。执行ISP功能时，ISPARD[1:0] 必须为00b |

## ISP 数据寄存器 (ISPDAT)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ISPDAT** | FMC_BA+0x08 | R/W | ISP 数据寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **ISPDAT** | **ISP 数据**<br>ISP写操作之前，写数据到该寄存器<br>ISP读操作后，可从该寄存器读数据 |

## ISP 命令寄存器 (ISPCMD)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ISPCMD** | FMC_BA+0x0C | R/W | ISP 命令寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:6] | **Reserved** | 保留 |
| [5:0] | **CMD** | **ISP 命令**<br>ISP 命令表如下:<br>0x00 = 读FLASH<br>0x04 = 读UID<br>0x0B = 读公司ID (0xDA).<br>0x21 = 写FLASH<br>0x22 = FLASH页擦除<br>0x2E =向量页重映射<br>其它命令都视作无效 |

## ISP 触发控制寄存器 (ISPTRG)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ISPTRG** | FMC_BA+0x10 | R/W | ISP 触发控制寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **ISPGO** | **ISP 开始触发 (写保护)**<br>写 1 开始ISP操作，当ISP操作结束后，该位由硬件自动清零.<br>0 = ISP 操作结束<br>1 = ISP 正在执行<br>这是写保护的位，写这些被保护的位需要依次向地址0x5000_0100写入“59h”，“16h”，“88h”，禁用寄存器保护。参考寄存器REGWRPROT，地址GCR_BA+0x100。<br>**注**：为了确保在CPU运行之前ISP功能已经完成，在ISPGO设置之后，应该用ISB指令。 |

## 数据 Flash 基地址寄存器 (DFBADR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **DFBADR** | FMC_BA+0x14 | R | 数据 Flash 基地址 | 0x000X_XXXX |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **DFBADR** | **数据 Flash 基地址**<br>该寄存器为数据FLASH开始地址寄存器，只读<br>对于128KBflash存储器设备，数据flash大小是由用户配置定义的，当芯片上电时该寄存器的值会从Config1中加载。但是对于64KB设备，其值固定为0x0001_F000。 |

## Flash 访问时间控制寄存器 (FATCON)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **FATCON** | FMC_BA+0x18 | R/W | Flash 访问时间控制寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:7] | **Reserved** | 保留 |
| [6] | **FOM_SEL[1]** | **芯片频率优化模式选择1（写保护位）**<br>当芯工作频率低于 25 MHz时，通过设置FOM_SEL[1:0] = 01，芯片可以更高效的工作。<br>当芯工作频率低于 50 MHz时，通过设置FOM_SEL[1:0] = 00，芯片可以更高效的工作。<br>当芯工作频率高于 50 MHz时，通过设置FOM_SEL[1:0] = 11，芯片可以更高效的工作。<br><br>00=中等频率优化模式使能<br>01=低频率优化模式使能<br>10=保留<br>11=高频率优化模式使能 |
| [5] | **Reserved** | 保留 |
| [4] | **FOM_SEL[0]** | **芯片频率优化模式选择0（写保护位）**<br>当芯工作频率低于 25 MHz时，通过设置FOM_SEL[1:0] = 01，芯片可以更高效的工作。<br>当芯工作频率低于 50 MHz时，通过设置FOM_SEL[1:0] = 00，芯片可以更高效的工作。<br>当芯工作频率高于 50 MHz时，通过设置FOM_SEL[1:0] = 11，芯片可以更高效的工作。<br><br>00=中等频率优化模式使能<br>01=低频率优化模式使能<br>10=保留<br>11=高频率优化模式使能 |
| [3:0] | **Reserved** | 保留 |

## ISP 状态寄存器 (ISPSTA)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ISPSTA** | FMC_BA+0x40 | R/W | ISP 状态寄存器 | 0x00XX_XX0X |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | **VECMAP** | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **VECMAP** | | | | | | | **Reserved** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | **ISPFF** | **Reserved** | | | **CBS** | | **ISPGO** |

| 位 | 描述 | |
|---|---|---|
| [31:21] | **Reserved** | 保留 |
| [20:9] | **VECMAP** | **向量页映像地址（只读）**<br>当前 flash 地址空间 0x0000_0000~0x0000_01FF 映像到地址 {VECMAP[11:0], 9'h000} ~ {VECMAP[11:0], 9'h1FF}<br>**注：**仅在IAP模式开启的情况下向量重映射功能才有效 |
| [8:7] | **Reserved** | 保留 |
| [6] | **ISPFF** | **ISP失败标志（写保护位）**<br>该位是ISPFF (ISPCON[6])的镜像，需要写1到ISPCON[6] 或ISPSTA[6]清除该标志。<br>当 ISP 满足下列条件时，该位由硬件置位：<br>(1) APUEN等于0时，APROM 写APROM<br>(2) LDUEN等于0时，LDROM 写LDROM<br>(3) CFGUEN等于0时，CONFIG 被擦除或编程<br>(4)ICE连接时且处在LOCK模式下，执行页擦除命令<br>(5)在欠压情况下，执行擦除或是编程命令<br>(6) 定义地址无效，如超过正常范围<br>(7)无效的ISP命令<br>写 1 清除该位。 |
| [5:3] | **Reserved** | 保留 |
| [2:1] | **CBS** | **芯片启动选择(只读)**<br>这是CBS在CONFIG0中的镜像<br>当CBS[0] = 0时IAP功能被使能，APROM的基地址是0X0、LDROM基地址是0x100000。<br>IAP功能意味着系统不用重启，CPU就可以执行和访问APROM 、LDROM。<br>00=由LDROM启动带IAP功能<br>01=由LDROM启动不带IAP功能<br>10=由APROM启动带IAP功能<br>11=由APROM启动不带IAP功能<br>**注：**<br>BS (ISPCON[1])是用在没有IAP模式下选择启动区域。<br>VECMAP (ISPSTA[20:9])是用在带有IAP模式下将选择的区域重映射到0x0~0x1ff |

| | | |
|---|---|---|
| [0] | **ISPGO** | **ISP 开始触发（只读）**<br>写 1 开始 ISP 操作，当 ISP 操作结束后，该位由硬件自动清零。<br>0 = ISP 操作结束<br>1 = ISP 正在执行<br>注意：这位同ISPTRG bit0 |

## 6.5 通用 I/O (GPIO)

### 6.5.1 概述

NuMicro® M0519系列多达82个通用I/O管脚和其他功能管脚共享，这取决于芯片的配置。82个管脚分配在P0, P1, P2, P3, P4, P5, P6, P7, P8, P9 和 PA十个端口上。P0/1/2/3/4/5/6/7/8/9最多有8个管脚，PA最多有2个管脚。每个管脚都是独立的，都有相应的寄存器位来控制管脚功能模式与数据。

I/O管脚的I/O类型可由软件独立地配置为输入，输出，开漏或准双向模式。复位之后，所有管脚的I/O 类型为输入模式。在准双向模式中，I/O管脚有一个阻值为110K~300K的弱上拉电阻接到$V_{DD}$上，$V_{DD}$范围从5.0 V 到2.5 V。

### 6.5.2 特性

- 四种 I/O 模式:
  - 准双向模式
  - 推挽输出
  - 开漏输出
  - 高阻态输入
- 通过Px_MFP[23:16]中的Px_TYPE[7:0]，可选TTL/Schmitt 触发输入
- I/O可以配置为边沿/电平触发的中断源
- I/O脚仅在准双向模式，内部上拉电阻才使能
- 使能管脚中断功能也将同时使能了唤醒功能

### 6.5.3 框图

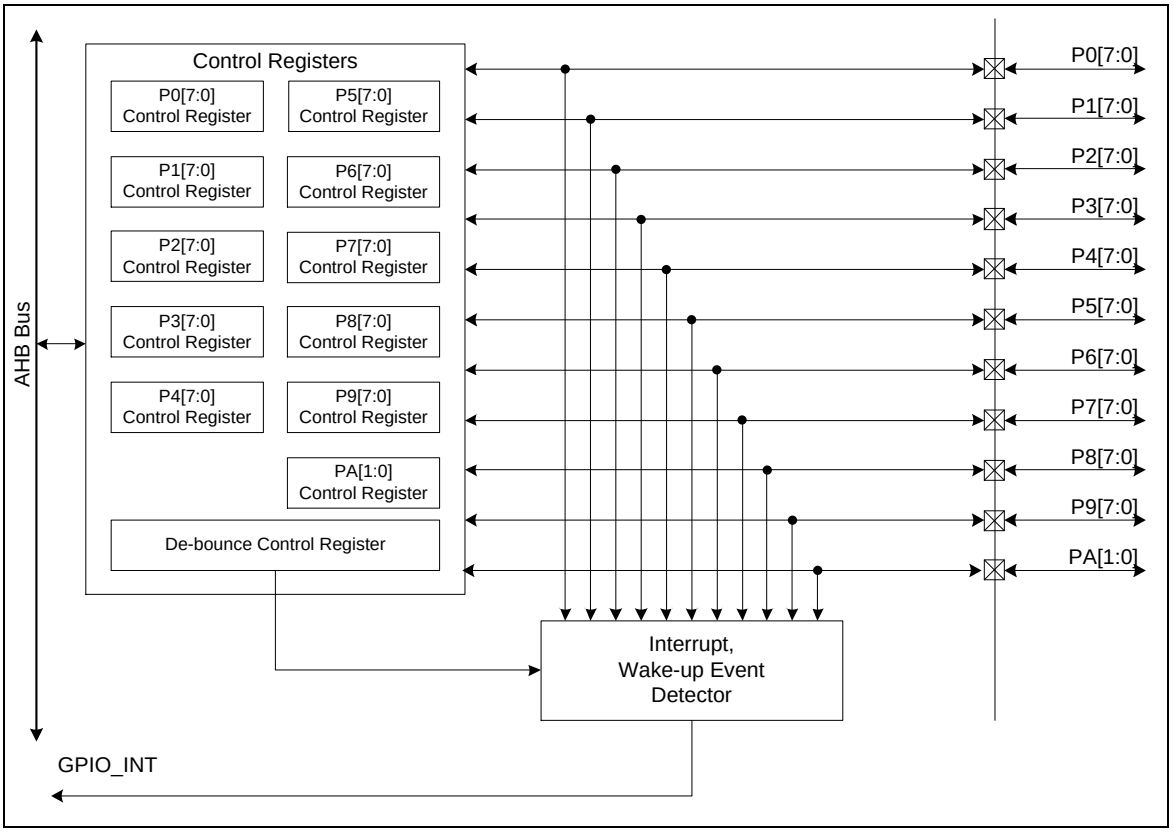



图 6-16 GPIO 控制器 框图

**基本配置**

GPIO 管脚通过配置 P0_MFP, P1_MFP, P2_MFP, P3_MFP, P4_MFP, P5_MFP, P6_MFP, P7_MFP, P8_MFP, P9_MFP 和 PA_MFP寄存器来实现其功能

### 6.5.4 功能描述

#### *6.5.4.1 只输入模式*

设置Px_PMD (PMDn[1:0])为 00b ，Px port [n]为输入模式，I/O管脚为三态（高阻），没有输出驱动能力。Px_PIN的值反映相应端口的状态。

#### *6.5.4.2 推挽输出模式*

设置Px_PMD (PMDn[1:0])为 01b，Px port [n]为推挽输出模式，I/O支持数字输出功能，有拉/灌电流能力。Px_DOUT相应位bit[n]的值被送到相应管脚上

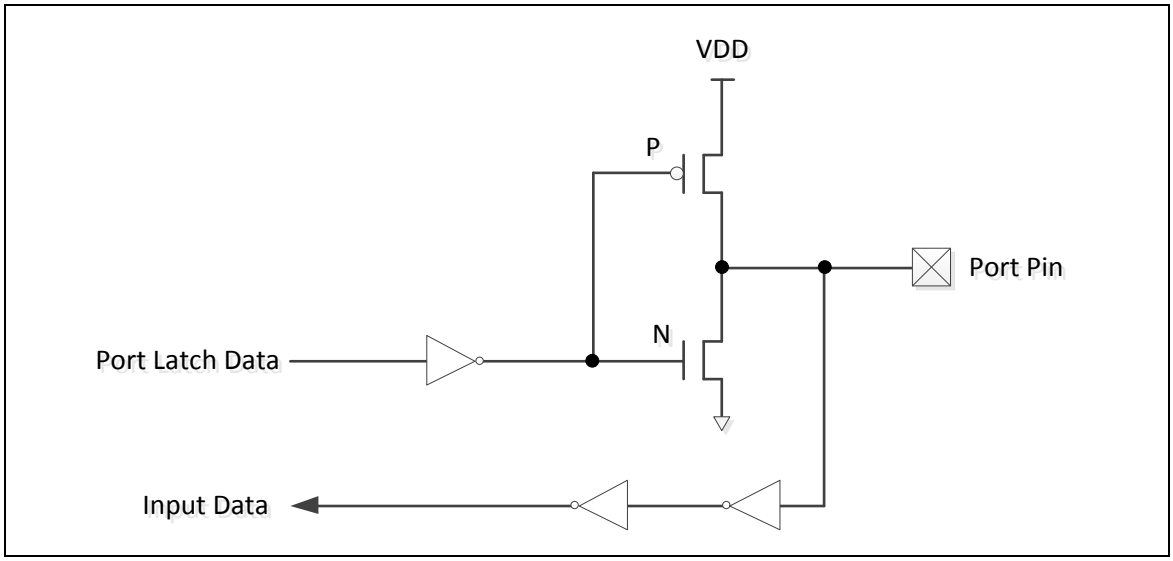



图 6-17 推挽输出

#### *6.5.4.3 开漏模式*

设置Px_PMD (PMDn[1:0])为10b，Px port [n]为开漏模式且I/O管脚数字输出功能仅支持灌电流，驱动到高电平需要一个外加上拉电阻。如果Px_DOUT相应位bit [n]的值为‘0’，管脚上输出低。如果Px_DOUT相应位bit [n]的值为‘1’，该管脚输出为高，可以由外部上拉电阻控制

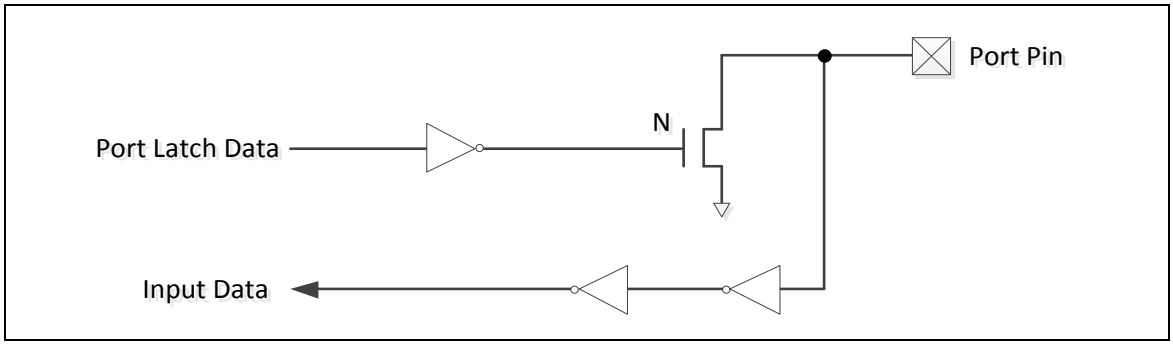



图 6-18 开漏输出

#### *6.5.4.4 准双向模式*

设置Px_PMD (PMDn[1:0])为11b，Px port [n]为准双向模式，I/O同时支持数字输出和输入功能，但拉电流能力仅达数百uA。要实现数字输入，需要先将Px_DOUT相应位置1。准双向输出是80C51及其派生产品常见的模式。若Px_DOUT相应位bit[n]为‘0’，管脚上输出为“低”。若Px_DOUT相应位bit[n]为‘1’，该管脚将检测管脚值。若管脚值为高，没有任何动作。若管脚值为低，在该管脚上将驱

动2个时钟周期的高电平，然后禁止强输出驱动，其后管脚状态由内部上拉电阻控制。注意: 准双向模式拉电流的大小仅有200 uA到30 uA(相应VDD的电压从5.0 V到2.5 V) 。

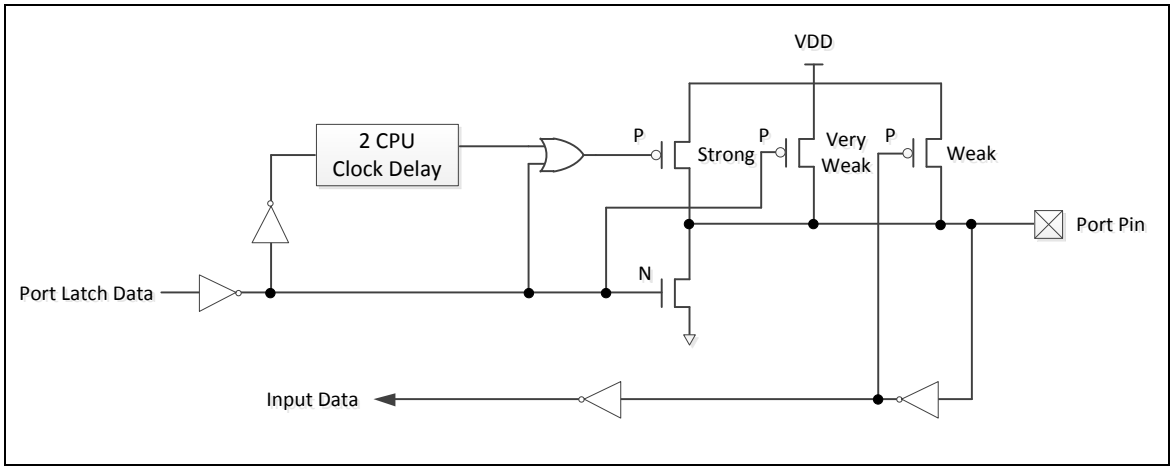



图 6-19 准双向 I/O 模式

*6.5.4.5 GPIO中断和唤醒功能*

每个GPIO管脚都可以通过Px_IEN位和Px_IMD设置成芯片的中断源。有五种中断条件可以设置：低电平触发、高电平触发、下降沿触发、上升沿触发、上升沿和下降沿都可触发。在边沿触发中用户可以通过使能输入信号去抖功能来阻止由噪声引起的意外中断。去抖时钟源和采样周期可以通过DEBOUNCE寄存器来设置。

当芯片进入空闲模式或掉电模式后，GPIO可以用来当作唤醒源，唤醒触发条件设置跟GPIO中断触发设置相同，当GPIO用作唤醒源时要注意：

- **进入掉电模式前确认I/O状态**

当使用GPIO触发来唤醒系统时，用户必须根据相关的唤醒设置在进入空闲模式或省电模式前确认I/O状态。

例如，配置I/O上升沿/高电平触发唤醒，用户必须确保相应的I/O管脚状态在进入空闲模式/省电模式前是低电平。同样配置I/O下降沿/低电平触发唤醒，用户必须确保相应的I/O管脚状态在进入空闲模式/省电模式前是高电平。

### 6.5.5 PWM端口输出驱动控制

M0519内部有2个增强型PWM单元，每个单元有6路输出。增强型PWM的单元0和单元1的端口输出分别是P0.0~P0.5 和 P1.0~P1.5。M0519内部还有一组基本的PWM，该PWM有两路输出。基本的PWM 端口输出是P5.6~P5.7。

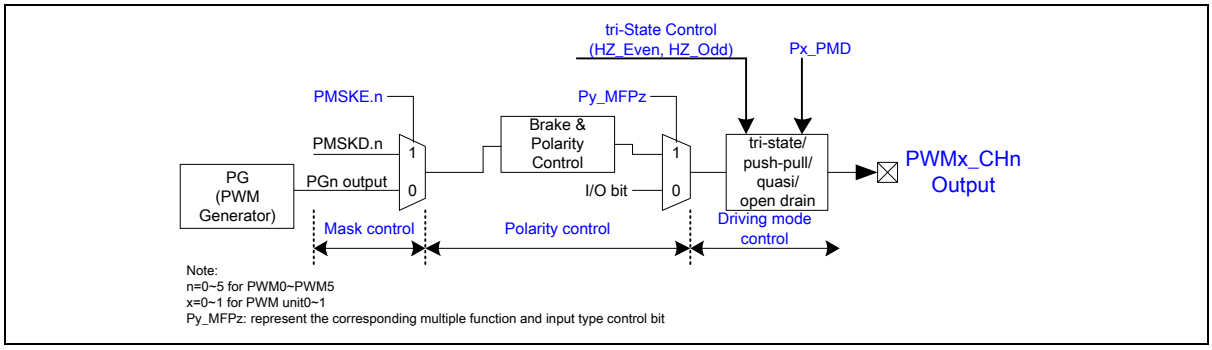


图 6-20 增加型PWM输出驱动控制

任何复位后PWM输出端口驱动模式可以被初始化成高阻态或是通过配置Px_PMD (PMDn[1:0])设置成其它模式。对于HZ_Even0 (PWMPOEN[0])/HZ_Odd0 (PWMPOEN[1])/HZ_Even1 (PWMPOEN[2])/HZ_Odd1 (PWMPOEN[3])/HZ_BPWM (PWMPOEN[3])的初始值是分别通过CHZ_Even0 (Config0[8])/ CHZ_Odd0 (Config0[9])/ CHZ_Even1 (Config0[10])/ CHZ_Odd1 (Config0[11])/CHZ_BPWM (Config0[12])定义的。

如果寄存器的每一个位被设置为0，PWM输出驱动端口模式由GPIO模式寄存器或是多功能寄存器控制。如果寄存器的每一个位被设置为1，PWM输出驱动端口被强制设置成高阻态。

| HZ_Even/HZ_Odd/HZ_BPWM (在PWMPOEN 寄存器内) | PWM 输出驱动模式 |
|---|---|
| 0 | 取决于 Px_PMD 或 Px_MFP |
| 1 | 强制为高阻态 |

表 6-8 PWM 输出驱动模式设置表

### 6.5.6 寄存器映射

**R**: 只读, **W**: 只写, **R/W**: 读/写

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **GPIO 基地址:**<br>**GPIO_BA = 0x5000_4000** | | | | |
| **P0_PMD** | GPIO_BA+0x000 | R/W | GPIO 端口0 管脚 I/O 模式控制 | 0x0000_0000 |
| **P0_OFFD** | GPIO_BA+0x004 | R/W | GPIO 端口0 管脚数字输入通道关闭控制 | 0x0000_0000 |
| **P0_DOUT** | GPIO_BA+0x008 | R/W | GPIO 端口0 数据输出值 | 0x0000_00FF |
| **P0_DMASK** | GPIO_BA+0x00C | R/W | GPIO 端口0 数据输出写屏蔽 | 0x0000_0000 |
| **P0_PIN** | GPIO_BA+0x010 | R | GPIO 端口0 管脚值 | 0x0000_00XX |
| **P0_DBEN** | GPIO_BA+0x014 | R/W | GPIO 端口0 去抖动使能 | 0x0000_0000 |
| **P0_IMD** | GPIO_BA+0x018 | R/W | GPIO 端口0 中断模式控制 | 0x0000_0000 |
| **P0_IEN** | GPIO_BA+0x01C | R/W | GPIO 端口0 中断使能 | 0x0000_0000 |
| **P0_ISF** | GPIO_BA+0x020 | R/W | GPIO 端口0 中断源标志 | 0xXXXX_XXXX |
| **P1_PMD** | GPIO_BA+0x040 | R/W | GPIO 端口1 管脚 I/O 模式控制 | 0x0000_0000 |
| **P1_OFFD** | GPIO_BA+0x044 | R/W | GPIO 端口1 管脚 数字输入通道关闭控制 | 0x0000_0000 |
| **P1_DOUT** | GPIO_BA+0x048 | R/W | GPIO 端口1 数据输出值 | 0x0000_00FF |
| **P1_DMASK** | GPIO_BA+0x04C | R/W | GPIO 端口1 数据输出写屏蔽 | 0x0000_0000 |
| **P1_PIN** | GPIO_BA+0x050 | R | GPIO 端口1 管脚值 | 0x0000_00XX |
| **P1_DBEN** | GPIO_BA+0x054 | R/W | GPIO 端口1 去抖动使能 | 0x0000_0000 |
| **P1_IMD** | GPIO_BA+0x058 | R/W | GPIO 端口1 中断模式控制 | 0x0000_0000 |
| **P1_IEN** | GPIO_BA+0x05C | R/W | GPIO 端口1 中断使能 | 0x0000_0000 |
| **P1_ISF** | GPIO_BA+0x060 | R/W | GPIO 端口1 中断源标志 | 0xXXXX_XXXX |
| **P2_PMD** | GPIO_BA+0x080 | R/W | GPIO 端口2 管脚 I/O 模式控制 | 0x0000_0000 |
| **P2_OFFD** | GPIO_BA+0x084 | R/W | GPIO 端口2 管脚 数字输入通道关闭控制 | 0x0000_0000 |
| **P2_DOUT** | GPIO_BA+0x088 | R/W | GPIO 端口2 数据输出值 | 0x0000_00FF |
| **P2_DMASK** | GPIO_BA+0x08C | R/W | GPIO 端口2 数据输出写屏蔽 | 0x0000_0000 |
| **P2_PIN** | GPIO_BA+0x090 | R | GPIO 端口2 管脚值 | 0x0000_00XX |
| **P2_DBEN** | GPIO_BA+0x094 | R/W | GPIO 端口2 去抖动使能 | 0x0000_0000 |
| **P2_IMD** | GPIO_BA+0x098 | R/W | GPIO 端口2 中断模式控制 | 0x0000_0000 |
| **P2_IEN** | GPIO_BA+0x09C | R/W | GPIO 端口2 中断使能 | 0x0000_0000 |

| | | | | |
|---|---|---|---|---|
| **P2_ISF** | GPIO_BA+0x0A0 | R/W | GPIO 端口2 中断源标志 | 0xXXXX_XXXX |
| **P3_PMD** | GPIO_BA+0x0C0 | R/W | GPIO 端口3 管脚 I/O 模式控制 | 0x0000_0000 |
| **P3_OFFD** | GPIO_BA+0x0C4 | R/W | GPIO 端口3 管脚 数字输入通道关闭控制 | 0x0000_0000 |
| **P3_DOUT** | GPIO_BA+0x0C8 | R/W | GPIO 端口3 数据输出值 | 0x0000_00FF |
| **P3_DMASK** | GPIO_BA+0x0CC | R/W | GPIO 端口3 数据输出写屏蔽 | 0x0000_0000 |
| **P3_PIN** | GPIO_BA+0x0D0 | R | GPIO 端口3 管脚值 | 0x0000_00XX |
| **P3_DBEN** | GPIO_BA+0x0D4 | R/W | GPIO 端口3 去抖动使能 | 0x0000_0000 |
| **P3_IMD** | GPIO_BA+0x0D8 | R/W | GPIO 端口3 中断模式控制 | 0x0000_0000 |
| **P3_IEN** | GPIO_BA+0x0DC | R/W | GPIO 端口3 中断使能 | 0x0000_0000 |
| **P3_ISF** | GPIO_BA+0x0E0 | R/W | GPIO 端口3 中断源标志 | 0xXXXX_XXXX |
| **P4_PMD** | GPIO_BA+0x100 | R/W | GPIO 端口4 管脚 I/O 模式控制 | 0x0000_0000 |
| **P4_OFFD** | GPIO_BA+0x104 | R/W | GPIO 端口4 管脚 数字输入通道关闭控制 | 0x0000_0000 |
| **P4_DOUT** | GPIO_BA+0x108 | R/W | GPIO 端口4 数据输出值 | 0x0000_00FF |
| **P4_DMASK** | GPIO_BA+0x10C | R/W | GPIO 端口4 数据输出写屏蔽 | 0x0000_0000 |
| **P4_PIN** | GPIO_BA+0x110 | R | GPIO 端口4 管脚值 | 0x0000_00XX |
| **P4_DBEN** | GPIO_BA+0x114 | R/W | GPIO 端口4 去抖动使能 | 0x0000_0000 |
| **P4_IMD** | GPIO_BA+0x118 | R/W | GPIO 端口4 中断模式控制 | 0x0000_0000 |
| **P4_IEN** | GPIO_BA+0x11C | R/W | GPIO 端口4 中断使能 | 0x0000_0000 |
| **P4_ISF** | GPIO_BA+0x120 | R/W | GPIO 端口4 中断源标志 | 0xXXXX_XXXX |
| **P5_PMD** | GPIO_BA+0x140 | R/W | GPIO 端口5 管脚 I/O 模式控制 | 0x0000_0000 |
| **P5_OFFD** | GPIO_BA+0x144 | R/W | GPIO 端口5 管脚 数字输入通道关闭控制 | 0x0000_0000 |
| **P5_DOUT** | GPIO_BA+0x148 | R/W | GPIO 端口5 数据输出值 | 0x0000_00FF |
| **P5_DMASK** | GPIO_BA+0x14C | R/W | GPIO 端口5 数据输出写屏蔽 | 0x0000_0000 |
| **P5_PIN** | GPIO_BA+0x150 | R | GPIO 端口5 管脚值 | 0x0000_00XX |
| **P5_DBEN** | GPIO_BA+0x154 | R/W | GPIO 端口5 去抖动使能 | 0x0000_0000 |
| **P5_IMD** | GPIO_BA+0x158 | R/W | GPIO 端口5 中断模式控制 | 0x0000_0000 |
| **P5_IEN** | GPIO_BA+0x15C | R/W | GPIO 端口5 中断使能 | 0x0000_0000 |
| **P5_ISF** | GPIO_BA+0x160 | R/W | GPIO 端口5 中断源标志 | 0xXXXX_XXXX |
| **P6_PMD** | GPIO_BA+0x180 | R/W | GPIO 端口6 管脚 I/O 模式控制 | 0x0000_0000 |
| **P6_OFFD** | GPIO_BA+0x184 | R/W | GPIO 端口6 管脚 数字输入通道关闭控制 | 0x0000_0000 |
| **P6_DOUT** | GPIO_BA+0x188 | R/W | GPIO 端口6 数据输出值 | 0x0000_00FF |

| | | | | |
|---|---|---|---|---|
| **P6_DMASK** | GPIO_BA+0x18C | R/W | GPIO 端口6 数据输出写屏蔽 | 0x0000_0000 |
| **P6_PIN** | GPIO_BA+0x190 | R | GPIO 端口6 管脚值 | 0x0000_00XX |
| **P6_DBEN** | GPIO_BA+0x194 | R/W | GPIO 端口6 去抖动使能 | 0x0000_0000 |
| **P6_IMD** | GPIO_BA+0x198 | R/W | GPIO 端口6 中断模式控制 | 0x0000_0000 |
| **P6_IEN** | GPIO_BA+0x19C | R/W | GPIO 端口6 中断使能 | 0x0000_0000 |
| **P6_ISF** | GPIO_BA+0x200 | R/W | GPIO 端口6 中断源标志 | 0xXXXX_XXXX |
| **P7_PMD** | GPIO_BA+0x1C0 | R/W | GPIO 端口7 管脚 I/O 模式控制 | 0x0000_0000 |
| **P7_OFFD** | GPIO_BA+0x1C4 | R/W | GPIO 端口7 管脚 数字输入通道关闭控制 | 0x0000_0000 |
| **P7_DOUT** | GPIO_BA+0x1C8 | R/W | GPIO 端口7 数据输出值 | 0x0000_00FF |
| **P7_DMASK** | GPIO_BA+0x1CC | R/W | GPIO 端口7 数据输出写屏蔽 | 0x0000_0000 |
| **P7_PIN** | GPIO_BA+0x1D0 | R | GPIO 端口7 管脚值 | 0x0000_00XX |
| **P7_DBEN** | GPIO_BA+0x1D4 | R/W | GPIO 端口7 去抖动使能 | 0x0000_0000 |
| **P7_IMD** | GPIO_BA+0x1D8 | R/W | GPIO 端口7 中断模式控制 | 0x0000_0000 |
| **P7_IEN** | GPIO_BA+0x1DC | R/W | GPIO 端口7 中断使能 | 0x0000_0000 |
| **P7_ISF** | GPIO_BA+0x1E0 | R/W | GPIO 端口7 中断源标志 | 0xXXXX_XXXX |
| **P8_PMD** | GPIO_BA+0x200 | R/W | GPIO 端口8 管脚 I/O 模式控制 | 0x0000_0000 |
| **P8_OFFD** | GPIO_BA+0x204 | R/W | GPIO 端口8 管脚 数字输入通道关闭控制 | 0x0000_0000 |
| **P8_DOUT** | GPIO_BA+0x208 | R/W | GPIO 端口8 数据输出值 | 0x0000_00FF |
| **P8_DMASK** | GPIO_BA+0x20C | R/W | GPIO 端口8 数据输出写屏蔽 | 0x0000_0000 |
| **P8_PIN** | GPIO_BA+0x210 | R | GPIO 端口8 管脚值 | 0x0000_00XX |
| **P8_DBEN** | GPIO_BA+0x214 | R/W | GPIO 端口8 去抖动使能 | 0x0000_0000 |
| **P8_IMD** | GPIO_BA+0x218 | R/W | GPIO 端口8 中断模式控制 | 0x0000_0000 |
| **P8_IEN** | GPIO_BA+0x21C | R/W | GPIO 端口8 中断使能 | 0x0000_0000 |
| **P8_ISF** | GPIO_BA+0x220 | R/W | GPIO 端口8 中断源标志 | 0xXXXX_XXXX |
| **P9_PMD** | GPIO_BA+0x240 | R/W | GPIO 端口9 管脚 I/O 模式控制 | 0x0000_0000 |
| **P9_OFFD** | GPIO_BA+0x244 | R/W | GPIO 端口9 管脚 数字输入通道关闭控制 | 0x0000_0000 |
| **P9_DOUT** | GPIO_BA+0x248 | R/W | GPIO 端口9 数据输出值 | 0x0000_00FF |
| **P9_DMASK** | GPIO_BA+0x24C | R/W | GPIO 端口9 数据输出写屏蔽 | 0x0000_0000 |
| **P9_PIN** | GPIO_BA+0x250 | R | GPIO 端口9 管脚值 | 0x0000_00XX |
| **P9_DBEN** | GPIO_BA+0x254 | R/W | GPIO 端口9 去抖动使能 | 0x0000_0000 |
| **P9_IMD** | GPIO_BA+0x258 | R/W | GPIO 端口9 中断模式控制 | 0x0000_0000 |

| | | | | |
|---|---|---|---|---|
| **P9_IEN** | GPIO_BA+0x25C | R/W | GPIO 端口9 中断使能 | 0x0000_0000 |
| **P9_ISF** | GPIO_BA+0x260 | R/W | GPIO 端口9 中断源标志 | 0xXXXX_XXXX |
| **PA_PMD** | GPIO_BA+0x280 | R/W | GPIO 端口A 管脚 I/O 模式控制 | 0x0000_0000 |
| **PA_OFFD** | GPIO_BA+0x284 | R/W | GPIO 端口A 管脚 数字输入通道关闭控制 | 0x0000_0000 |
| **PA_DOUT** | GPIO_BA+0x288 | R/W | GPIO 端口A 数据输出值 | 0x0000_0003 |
| **PA_DMASK** | GPIO_BA+0x28C | R/W | GPIO 端口A 数据输出写屏蔽 | 0x0000_0000 |
| **PA_PIN** | GPIO_BA+0x290 | R | GPIO 端口A 管脚值 | 0x0000_000X |
| **PA_DBEN** | GPIO_BA+0x294 | R/W | GPIO 端口A 去抖动使能 | 0x0000_0000 |
| **PA_IMD** | GPIO_BA+0x298 | R/W | GPIO 端口A 中断模式控制 | 0x0000_0000 |
| **PA_IEN** | GPIO_BA+0x29C | R/W | GPIO 端口A 中断使能 | 0x0000_0000 |
| **PA_ISF** | GPIO_BA+0x2A0 | R/W | GPIO 端口A 中断源标志 | 0xXXXX_XXXX |
| **DBNCECON** | GPIO_BA+0x2E0 | R/W | 外部中断去抖动控制 | 0x0000_0000 |
| **PWMPOEN** | GPIO_BA+0x2E4 | R/W | PWM 端口输出使能 | 0x0000_00XX |
| **P0_N**<br>**n=0,1..7** | GPIO_BA+0x300<br>+ 0x04 * n | R/W | GPIO P0.n 端口数据输入/输出 | 0x0000_000X |
| **P1_N**<br>**n=0,1..7** | GPIO_BA+0x320<br>+ 0x04 * n | R/W | GPIO P1.n端口数据输入/输出 | 0x0000_000X |
| **P2_N**<br>**n=0,1..7** | GPIO_BA+0x340<br>+ 0x04 * n | R/W | GPIO P2.n端口数据输入/输出 | 0x0000_000X |
| **P3_N**<br>**n=0,1..7** | GPIO_BA+0x360<br>+ 0x04 * n | R/W | GPIO P3.n端口数据输入/输出 | 0x0000_000X |
| **P4_N**<br>**n=0,1..7** | GPIO_BA+0x380<br>+ 0x04 * n | R/W | GPIO P4.n端口数据输入/输出 | 0x0000_000X |
| **P5_N**<br>**n=0,1..7** | GPIO_BA+0x3A0<br>+ 0x04 * n | R/W | GPIO P5.n端口数据输入/输出 | 0x0000_000X |
| **P6_N**<br>**n=0,1..7** | GPIO_BA+0x3C0<br>+ 0x04 * n | R/W | GPIO P6.n端口数据输入/输出 | 0x0000_000X |
| **P7_N**<br>**n=0,1..7** | GPIO_BA+0x3E0<br>+ 0x04 * n | R/W | GPIO P7.n端口数据输入/输出 | 0x0000_000X |
| **P8_N**<br>**n=0,1..7** | GPIO_BA+0x400<br>+ 0x04 * n | R/W | GPIO P8.n端口数据输入/输出 | 0x0000_000X |
| **P9_N**<br>**n=0,1..7** | GPIO_BA+0x420<br>+ 0x04 * n | R/W | GPIO P9.n端口数据输入/输出 | 0x0000_000X |
| **PA_N**<br>**n=0,1** | GPIO_BA+0x440<br>+ 0x04 * n | R/W | GPIO PA.n端口数据输入/输出 | 0x0000_000X |

### 6.5.7 寄存器描述

**GPIO 端口 [0/1/2/3/4/5/6/7/8/9/A] 管脚 I/O 模式控制(Px_PMD)**

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P0_PMD** | GPIO_BA+0x000 | R/W | GPIO 端口 0管脚 I/O 模式控制 | 0x0000_0000 |
| **P1_PMD** | GPIO_BA+0x040 | R/W | GPIO 端口1管脚 I/O 模式控制 | 0x0000_0000 |
| **P2_PMD** | GPIO_BA+0x080 | R/W | GPIO 端口2管脚 I/O 模式控制 | 0x0000_0000 |
| **P3_PMD** | GPIO_BA+0x0C0 | R/W | GPIO 端口3管脚 I/O 模式控制 | 0x0000_0000 |
| **P4_PMD** | GPIO_BA+0x100 | R/W | GPIO 端口4管脚 I/O 模式控制 | 0x0000_0000 |
| **P5_PMD** | GPIO_BA+0x140 | R/W | GPIO 端口5管脚 I/O 模式控制 | 0x0000_0000 |
| **P6_PMD** | GPIO_BA+0x180 | R/W | GPIO 端口6管脚 I/O 模式控制 | 0x0000_0000 |
| **P7_PMD** | GPIO_BA+0x1C0 | R/W | GPIO 端口7管脚 I/O 模式控制 | 0x0000_0000 |
| **P8_PMD** | GPIO_BA+0x200 | R/W | GPIO 端口8管脚 I/O 模式控制 | 0x0000_0000 |
| **P9_PMD** | GPIO_BA+0x240 | R/W | GPIO 端口9管脚 I/O 模式控制 | 0x0000_0000 |
| **PA_PMD** | GPIO_BA+0x280 | R/W | GPIO 端口A管脚 I/O 模式控制 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **PMD7** | | **PMD6** | | **PMD5** | | **PMD4** | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PMD3** | | **PMD2** | | **PMD1** | | **PMD0** | |

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留 |
| [2n+1:2n]<br>n=0,1..7 | **PMDn** | **端口 0-A I/O Pin[N]模式控制**<br>决定Px端口的每个I/O 类型<br>00 = GPIO port [n] 管脚为输入模式<br>01 = GPIO port [n] 管脚为推挽输出模式<br>10 = GPIO port [n] 管脚为开漏模式<br>11 = GPIO port [n] 管脚为准双向模式<br>**注:** 对于PA，n 最大为 1；对于P0/P1/P2/P3/P4/P5/P6/P7/P8/P9，n最大为 7。 |

## GPIO 端口 [0/1/2/3/4/5/6/7/8/9/A] 管脚数字输入通道关闭控制(Px_OFFD)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P0_OFFD** | GPIO_BA+0x004 | R/W | GPIO端口0管脚数字输入通道关闭控制 | 0x0000_0000 |
| **P1_OFFD** | GPIO_BA+0x044 | R/W | GPIO端口1管脚数字输入通道关闭控制 | 0x0000_0000 |
| **P2_OFFD** | GPIO_BA+0x084 | R/W | GPIO端口2管脚数字输入通道关闭控制 | 0x0000_0000 |
| **P3_OFFD** | GPIO_BA+0x0C4 | R/W | GPIO端口3管脚数字输入通道关闭控制 | 0x0000_0000 |
| **P4_OFFD** | GPIO_BA+0x104 | R/W | GPIO端口4管脚数字输入通道关闭控制 | 0x0000_0000 |
| **P5_OFFD** | GPIO_BA+0x144 | R/W | GPIO端口5管脚数字输入通道关闭控制 | 0x0000_0000 |
| **P6_OFFD** | GPIO_BA+0x184 | R/W | GPIO端口6管脚数字输入通道关闭控制 | 0x0000_0000 |
| **P7_OFFD** | GPIO_BA+0x1C4 | R/W | GPIO端口7管脚数字输入通道关闭控制 | 0x0000_0000 |
| **P8_OFFD** | GPIO_BA+0x204 | R/W | GPIO端口8管脚数字输入通道关闭控制 | 0x0000_0000 |
| **P9_OFFD** | GPIO_BA+0x244 | R/W | GPIO端口9管脚数字输入通道关闭控制 | 0x0000_0000 |
| **PA_OFFD** | GPIO_BA+0x284 | R/W | GPIO端口A管脚数字输入通道关闭控制 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **OFFD** | **端口 0-A Pin[N] 数字输入通道关闭控制**<br>用于控制GPIO的数字输入通路是否使能. 如果输入为模拟信号，用户可以关闭输入通道防止漏电<br>0 = 使能IO数据输入通道<br>1 = 关闭IO的数字输入通道(数字输入被拉低)<br>**注:** 对于PA，n 最大为 1；对于P0/P1/P2/P3/P4/P5/P6/P7/P8/P9，n最大为 7。 |
| [15:0] | **Reserved** | 保留 |

## GPIO 端口 [0/1/2/3/4/5/6/7/8/9/A] 数据输出值(Px_DOUT)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P0_DOUT** | GPIO_BA+0x008 | R/W | GPIO端口0数据输出值 | 0x0000_00FF |
| **P1_DOUT** | GPIO_BA+0x048 | R/W | GPIO端口1数据输出值 | 0x0000_00FF |
| **P2_DOUT** | GPIO_BA+0x088 | R/W | GPIO端口2数据输出值 | 0x0000_00FF |
| **P3_DOUT** | GPIO_BA+0x0C8 | R/W | GPIO端口3数据输出值 | 0x0000_00FF |
| **P4_DOUT** | GPIO_BA+0x108 | R/W | GPIO端口4数据输出值 | 0x0000_00FF |
| **P5_DOUT** | GPIO_BA+0x148 | R/W | GPIO端口5数据输出值 | 0x0000_00FF |
| **P6_DOUT** | GPIO_BA+0x188 | R/W | GPIO端口6数据输出值 | 0x0000_00FF |
| **P7_DOUT** | GPIO_BA+0x1C8 | R/W | GPIO端口7数据输出值 | 0x0000_00FF |
| **P8_DOUT** | GPIO_BA+0x208 | R/W | GPIO端口8数据输出值 | 0x0000_00FF |
| **P9_DOUT** | GPIO_BA+0x248 | R/W | GPIO端口9数据输出值 | 0x0000_00FF |
| **PA_DOUT** | GPIO_BA+0x288 | R/W | GPIO端口A数据输出值 | 0x0000_0003 |

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

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留 |
| [7:0] | **DOUT** | **端口 0-A Pin[N]输出值**<br>在GPIO配置成推挽输出，开漏和准双向模式时，控制GPIO相应管脚的状态.<br>0 = GPIO配置成推挽输出，开漏和准双向模式时，GPIO port [0/1/2/3/4/5/6/7/8/9/A] Pin[n] 为低<br>1 = GPIO配置成推挽输出，开漏和准双向模式时，GPIO port [0/1/2/3/4/5/6/7/8/9/A] Pin[n] 为高<br>**注:** 对于PA，n 最大为 1；对于P0/P1/P2/P3/P4/P5/P6/P7/P8/P9，n最大为 7。 |

## GPIO 端口 [0/1/2/3/4/5/6/7/8/9/A] 数据输出写屏蔽(Px _DMASK)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P0_DMASK** | GPIO_BA+0x00C | R/W | GPIO端口0数据输出写屏蔽 | 0x0000_0000 |
| **P1_DMASK** | GPIO_BA+0x04C | R/W | GPIO端口1数据输出写屏蔽 | 0x0000_0000 |
| **P2_DMASK** | GPIO_BA+0x08C | R/W | GPIO端口2数据输出写屏蔽 | 0x0000_0000 |
| **P3_DMASK** | GPIO_BA+0x0CC | R/W | GPIO端口3数据输出写屏蔽 | 0x0000_0000 |
| **P4_DMASK** | GPIO_BA+0x10C | R/W | GPIO端口4数据输出写屏蔽 | 0x0000_0000 |
| **P5_DMASK** | GPIO_BA+0x14C | R/W | GPIO端口5数据输出写屏蔽 | 0x0000_0000 |
| **P6_DMASK** | GPIO_BA+0x18C | R/W | GPIO端口6数据输出写屏蔽 | 0x0000_0000 |
| **P7_DMASK** | GPIO_BA+0x1CC | R/W | GPIO端口7数据输出写屏蔽 | 0x0000_0000 |
| **P8_DMASK** | GPIO_BA+0x20C | R/W | GPIO端口8数据输出写屏蔽 | 0x0000_0000 |
| **P9_DMASK** | GPIO_BA+0x24C | R/W | GPIO端口9数据输出写屏蔽 | 0x0000_0000 |
| **PA_DMASK** | GPIO_BA+0x28C | R/W | GPIO端口A数据输出写屏蔽 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留 |
| [7:0] | **DMASK** | **端口 0-A 数据输出写屏蔽**<br>用于保护相应寄存器Px _DOUT bit[n]. 当设置DMASK bit[n] 为 ‘1’时，相应DOUT[n] bit 被保护，写信号被屏蔽时，不能向保护位写数据。<br>0 =相应的Px _DOUT[n] 位可以被更新<br>1 =保护相应的Px _DOUT[n] 位<br>**注1:** 此功能只保护 Px_DOUT[n] 相应位，不能保护相应位的控制寄存器(P0_n, P1_n, P2_n, P3_n, P4_n, P5_n, P6_n, P7_n, P8_n, P9_n 和 PA_n).<br>**注2:** 对于PA，n 最大为 1；对于P0/P1/P2/P3/P4/P5/P6/P7/P8/P9，n最大为 7。 |

## GPIO 端口 [0/1/2/3/4/5/6/7/8/9/A] 管脚值(Px _PIN)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P0_PIN** | GPIO_BA+0x010 | R | GPIO端口0管脚值 | 0x0000_00XX |
| **P1_PIN** | GPIO_BA+0x050 | R | GPIO端口1管脚值 | 0x0000_00XX |
| **P2_PIN** | GPIO_BA+0x090 | R | GPIO端口2管脚值 | 0x0000_00XX |
| **P3_PIN** | GPIO_BA+0x0D0 | R | GPIO端口3管脚值 | 0x0000_00XX |
| **P4_PIN** | GPIO_BA+0x110 | R | GPIO端口4管脚值 | 0x0000_00XX |
| **P5_PIN** | GPIO_BA+0x150 | R | GPIO端口5管脚值 | 0x0000_00XX |
| **P6_PIN** | GPIO_BA+0x190 | R | GPIO端口6管脚值 | 0x0000_00XX |
| **P7_PIN** | GPIO_BA+0x1D0 | R | GPIO端口7管脚值 | 0x0000_00XX |
| **P8_PIN** | GPIO_BA+0x210 | R | GPIO端口8管脚值 | 0x0000_00XX |
| **P9_PIN** | GPIO_BA+0x250 | R | GPIO端口9管脚值 | 0x0000_00XX |
| **PA_PIN** | GPIO_BA+0x290 | R | GPIO端口A管脚值 | 0x0000_000X |

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

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留 |
| [7:0] | **PIN** | **端口 0-A 管脚数据**<br>这些位的值为各个GPIO管脚真实状态的反映。如果值为'1'，表示相应管脚状态为高，否则为低<br>**注:** 对于PA，n 最大为 1；对于P0/P1/P2/P3/P4/P5/P6/P7/P8/P9，n最大为 7。 |

## GPIO 端口 [0/1/2/3/4/5/6/7/8/9/A] 去抖动使能(Px_DBEN)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P0_DBEN** | GPIO_BA+0x014 | R/W | GPIO端口0去抖动使能 | 0x0000_0000 |
| **P1_DBEN** | GPIO_BA+0x054 | R/W | GPIO端口1去抖动使能 | 0x0000_0000 |
| **P2_DBEN** | GPIO_BA+0x094 | R/W | GPIO端口2去抖动使能 | 0x0000_0000 |
| **P3_DBEN** | GPIO_BA+0x0D4 | R/W | GPIO端口3去抖动使能 | 0x0000_0000 |
| **P4_DBEN** | GPIO_BA+0x114 | R/W | GPIO端口4去抖动使能 | 0x0000_0000 |
| **P5_DBEN** | GPIO_BA+0x154 | R/W | GPIO端口5去抖动使能 | 0x0000_0000 |
| **P6_DBEN** | GPIO_BA+0x194 | R/W | GPIO端口6去抖动使能 | 0x0000_0000 |
| **P7_DBEN** | GPIO_BA+0x1D4 | R/W | GPIO端口7去抖动使能 | 0x0000_0000 |
| **P8_DBEN** | GPIO_BA+0x214 | R/W | GPIO端口8去抖动使能 | 0x0000_0000 |
| **P9_DBEN** | GPIO_BA+0x254 | R/W | GPIO端口9去抖动使能 | 0x0000_0000 |
| **PA_DBEN** | GPIO_BA+0x294 | R/W | GPIO端口A去抖动使能 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留 |
| [7:0] | **DBEN** | **端口 0-A输入信号去抖动使能位**<br>DBEN[n]用于使能相应位的去抖动功能. 如果输入信号脉冲宽度不能被两个连续的去抖动采样周期所采样，则被视为信号反弹，从而不触发中断. 去抖动时钟源由DBNCECON[4]控制, 一个去抖动周期由DBNCECON[3:0]控制。 DBEN[n] 仅用于"边沿触发"中断，不能用于"电平触发"中断。<br>0 = 禁用 bit[n] 去抖动功能<br>1 = 使能 bit[n] 去抖动功能<br>**注:** 对于PA，n 最大为 1；对于P0/P1/P2/P3/P4/P5/P6/P7/P8/P9，n最大为 7。 |

## GPIO 端口 [0/1/2/3/4/5/6/7/8/9/A] 中断模式控制(Px _IMD)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P0_IMD** | GPIO_BA+0x018 | R/W | GPIO端口0中断模式控制 | 0x0000_0000 |
| **P1_IMD** | GPIO_BA+0x058 | R/W | GPIO端口1中断模式控制 | 0x0000_0000 |
| **P2_IMD** | GPIO_BA+0x098 | R/W | GPIO端口2中断模式控制 | 0x0000_0000 |
| **P3_IMD** | GPIO_BA+0x0D8 | R/W | GPIO端口3中断模式控制 | 0x0000_0000 |
| **P4_IMD** | GPIO_BA+0x118 | R/W | GPIO端口4中断模式控制 | 0x0000_0000 |
| **P5_IMD** | GPIO_BA+0x158 | R/W | GPIO端口5中断模式控制 | 0x0000_0000 |
| **P6_IMD** | GPIO_BA+0x198 | R/W | GPIO端口6中断模式控制 | 0x0000_0000 |
| **P7_IMD** | GPIO_BA+0x1D8 | R/W | GPIO端口7中断模式控制 | 0x0000_0000 |
| **P8_IMD** | GPIO_BA+0x218 | R/W | GPIO端口8中断模式控制 | 0x0000_0000 |
| **P9_IMD** | GPIO_BA+0x258 | R/W | GPIO端口9中断模式控制 | 0x0000_0000 |
| **PA_IMD** | GPIO_BA+0x298 | R/W | GPIO端口A中断模式控制 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留 |
| [7:0] | **IMD** | **端口 0-A边沿或电平检测中断控制**<br>IMD[n] 用于控制电平触发或边沿触发中断. 若为边沿触发中断，触发源可由去抖动控制，如果是电平触发中断，输入源由一个HCLK时钟采样并产生中断。<br>0 = 边沿触发中断<br>1 = 电平触发中断<br>如果设置管脚为电平触发模式，则在寄存器Px _IEN中，只能设置一个电平(高电平或者低电平)；若设置了两个电平都触发中断，则设置被忽略，不会产生中断<br>去抖动功能对于边沿触发中断有效，对于电平触发中断无效<br>**注:** 对于PA，n 最大为 1；对于P0/P1/P2/P3/P4/P5/P6/P7/P8/P9，n最大为 7。 |

## GPIO 端口 [0/1/2/3/4/5/6/7/8/9/A] 中断使能控制(Px _IEN)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P0_IEN** | GPIO_BA+0x01C | R/W | GPIO端口0中断使能 | 0x0000_0000 |
| **P1_IEN** | GPIO_BA+0x05C | R/W | GPIO端口1中断使能 | 0x0000_0000 |
| **P2_IEN** | GPIO_BA+0x09C | R/W | GPIO端口2中断使能 | 0x0000_0000 |
| **P3_IEN** | GPIO_BA+0x0DC | R/W | GPIO端口3中断使能 | 0x0000_0000 |
| **P4_IEN** | GPIO_BA+0x11C | R/W | GPIO端口4中断使能 | 0x0000_0000 |
| **P5_IEN** | GPIO_BA+0x15C | R/W | GPIO端口5中断使能 | 0x0000_0000 |
| **P6_IEN** | GPIO_BA+0x19C | R/W | GPIO端口6中断使能 | 0x0000_0000 |
| **P7_IEN** | GPIO_BA+0x1DC | R/W | GPIO端口7中断使能 | 0x0000_0000 |
| **P8_IEN** | GPIO_BA+0x21C | R/W | GPIO端口8中断使能 | 0x0000_0000 |
| **P9_IEN** | GPIO_BA+0x25C | R/W | GPIO端口9中断使能 | 0x0000_0000 |
| **PA_IEN** | GPIO_BA+0x29C | R/W | GPIO端口A中断使能 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **IR_EN** | **端口 0-A输入上升沿或输入高电平中断使能**<br>IR_EN[n] 用于使能相应Px _PIN[n]输入的中断. 置'1'也可以使能管脚唤醒功能<br>当设置 IR_EN[n] 位为'1':<br>如果中断是电平触发模式，输入PIN[n]的状态为高电平时，产生中断.<br>如果中断是边沿触发模式，输入PIN[n]的状态由低电平到高电平变化时，产生中断.<br>0 = 禁用PIN[n]高电平或由低电平到高电平变化的中断<br>1 = 使能PIN[n]高电平或由低电平到高电平变化的中断<br>**注:** 对于PA，n 最大为 1；对于P0/P1/P2/P3/P4/P5/P6/P7/P8/P9，n最大为 7。 |
| [15:8] | **Reserved** | 保留 |

| | | |
|---|---|---|
| [7:0] | **IF_EN[n]** | **端口 0-A输入下降沿或输入低电平中断使能**<br>IF_EN[n] 用于使能相应Px _PIN[n]输入的中断. 置 '1' 也可以使能管脚唤醒功能<br>当设置 IF_EN[n] 位为 '1':<br>如果中断是电平触发模式，输入PIN[n]的状态为低电平时，产生中断.<br>如果中断是边沿触发模式，输入PIN[n]的状态由高电平到低电平变化时，产生中断.<br>0 = 禁用PIN[n]低电平或由高电平到低电平变化的中断<br>1 = 使能PIN[n]低电平或由高电平到低电平变化的中断<br>**注:** 对于PA，n 最大为 1；对于P0/P1/P2/P3/P4/P5/P6/P7/P8/P9，n最大为 7。 |

## GPIO 端口 [0/1/2/3/4/5/6/7/8/9/A] 中断源标志(Px_ISF)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P0_ISF** | GPIO_BA+0x020 | R/W | GPIO端口0中断源标志 | 0xXXXX_XXXX |
| **P1_ISF** | GPIO_BA+0x060 | R/W | GPIO端口1中断源标志 | 0xXXXX_XXXX |
| **P2_ISF** | GPIO_BA+0x0A0 | R/W | GPIO端口2中断源标志 | 0xXXXX_XXXX |
| **P3_ISF** | GPIO_BA+0x0E0 | R/W | GPIO端口3中断源标志 | 0xXXXX_XXXX |
| **P4_ISF** | GPIO_BA+0x120 | R/W | GPIO端口4中断源标志 | 0xXXXX_XXXX |
| **P5_ISF** | GPIO_BA+0x160 | R/W | GPIO端口5中断源标志 | 0xXXXX_XXXX |
| **P6_ISF** | GPIO_BA+0x1A0 | R/W | GPIO端口6中断源标志 | 0xXXXX_XXXX |
| **P7_ISF** | GPIO_BA+0x1E0 | R/W | GPIO端口7中断源标志 | 0xXXXX_XXXX |
| **P8_ISF** | GPIO_BA+0x220 | R/W | GPIO端口8中断源标志 | 0xXXXX_XXXX |
| **P9_ISF** | GPIO_BA+0x260 | R/W | GPIO端口9中断源标志 | 0xXXXX_XXXX |
| **PA_ISF** | GPIO_BA+0x2A0 | R/W | GPIO端口A中断源标志 | 0xXXXX_XXXX |

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

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留 |
| [7:0] | **IF_ISF** | **端口 0-A中断源标志**<br>该位用来指示端口x每个管脚的边沿/电平触发事件，如果GPG0_INT或是GPG1_INT使能位被设置，相应的中断服务程序将会被执行。<br>注当P3.2 和 P3.3中断使能，中断服务向量为INT0_INT 和 INT1_INT<br>读：<br>0 = Px[n]没有中断<br>1 = Px[n] 产生中断<br>写：<br>0 = 无动作<br>1 = 清相应的未处理中断标志 |

| | | |
|---|---|---|
| | | **注:** 对于PA，n 最大为 1；对于P0/P1/P2/P3/P4/P5/P6/P7/P8/P9，n最大为 7。 |

## 外部中断去抖动控制(DBNCECON)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **DBNCECON** | GPIO_BA+0x2E0 | R/W | 外部中断去抖动控制 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:6] | **Reserved** | 保留 |
| [5] | **ICLK_ON** | **中断时钟打开模式**<br>0 = 边沿侦测电路仅在 I/O管脚对应的Px _IEN位置 1有效<br>1 = 复位之后所有 I/O边沿侦测电路使能<br>如果没有相关的特定应用，建议关掉时钟以减少耗电 |
| [4] | **DBCLKSRC** | **去抖动计数器时钟源选择**<br>0 = 去抖动计数器时钟源为 HCLK<br>1 = 去抖动计数器时钟源为内部 10 KHz 时钟 |
| [3:0] | **DBCLKSEL** | **去抖动采样周期选择**<br>0000 =采样中断输入信号, 每 1 clocks一次<br>0001 = 采样中断输入信号, 每 2 clocks一次<br>0010 = 采样中断输入信号, 每 4 clocks一次<br>0011 = 采样中断输入信号, 每 8 clocks一次<br>0100 = 采样中断输入信号, 每 16 clocks一次<br>0101 = 采样中断输入信号, 每 32 clocks一次<br>0110 = 采样中断输入信号, 每 64 clocks一次<br>0111 = 采样中断输入信号, 每 128 clocks一次<br>1000 = 采样中断输入信号, 每 256 clocks一次<br>1001 = 采样中断输入信号, 每 2*256 clocks一次<br>1010 = 采样中断输入信号, 每 4*256 clocks一次<br>1011 = 采样中断输入信号, 每 8*256 clocks一次<br>1100 = 采样中断输入信号, 每 16*256 clocks一次<br>1101 = 采样中断输入信号, 每 32*256 clocks一次<br>1110 = 采样中断输入信号, 每 64*256 clocks一次<br>1111 = 采样中断输入信号, 每 128*256 clocks一次 |

## PWM 端口输出使能 (PWMPOEN)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PWMPOEN** | GPIO_BA+0x2E4 | R/W | PWM端口输出使能 | 0x0000_00XX |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | **HZ_BPWM** | **HZ_Odd1** | **HZ_Even1** | **HZ_Odd0** | **HZ_Even0** |

| 位 | 描述 | |
|---|---|---|
| [31:5] | **Reserved** | 保留 |
| [4] | **HZ_BPWM** | **基本 PWM0 端口输出控制**<br>0 =基本PWM端口驱动模式由GPIO模式寄存器(Px_PMD)或是多功能寄存器(Px_MFP)控制<br>1 = 基本PWM端口驱动模式被强制设置成高阻态<br>**注：**任何复位后该位初始值是从CHZ_BPWM (Config0[12])中加载进来 |
| [3] | **HZ_Odd1** | **增强型 PWM 单元1 奇端口输出控制**<br>0 =增强型PWM单元1奇端口驱动模式由GPIO模式寄存器(Px_PMD)或是多功能寄存器(Px_MFP)控制<br>1 = 增强型PWM单元1奇端口驱动模式被强制设置成高阻态<br>**注：**任何复位后该位初始值是从CHZ_Odd1 (Config0[11])中加载进来 |
| [2] | **HZ_Even1** | **增强型 PWM 单元1 偶端口输出控制**<br>0 =增强型PWM单元1偶端口驱动模式由GPIO模式寄存器(Px_PMD)或是多功能寄存器(Px_MFP)控制<br>1 = 增强型PWM单元1偶端口驱动模式被强制设置成高阻态<br>**注：**任何复位后该位初始值是从CHZ_Even1 (Config0[10])中加载进来 |
| [1] | **HZ_Odd0** | **增强型 PWM 单元0 奇端口输出控制**<br>0 =增强型PWM单元0奇端口驱动模式由GPIO模式寄存器(Px_PMD)或是多功能寄存器(Px_MFP)控制<br>1 = 增强型PWM单元0奇端口驱动模式被强制设置成高阻态<br>**注：**任何复位后该位初始值是从CHZ_Odd0 (Config0[9])中加载进来 |

| 位 | 描述 | |
|---|---|---|
| [0] | **HZ_Even0** | **增强型 PWM 单元0 偶端口输出控制**<br>0 =增强型PWM单元0偶端口驱动模式由GPIO模式寄存器(Px_PMD)或是多功能寄存器(Px_MFP)控制<br>1 = 增强型PWM单元0偶端口驱动模式被强制设置成高阻态<br>**注：**任何复位后该位初始值是从CHZ_Even0 (Config0[8])中加载进来 |

## GPIO Px.n 管脚数据 输入/输出 (Px_n)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **P0_N n=0,1..7** | GPIO_BA+0x300+ 0x04 * n | R/W | GPIO P0.n 端口数据输入/输出 | 0x0000_000X |
| **P1_N n=0,1..7** | GPIO_BA+0x320+ 0x04 * n | R/W | GPIO P1.n 端口数据输入/输出 | 0x0000_000X |
| **P2_N n=0,1..7** | GPIO_BA+0x340+ 0x04 * n | R/W | GPIO P2.n 端口数据输入/输出 | 0x0000_000X |
| **P3_N n=0,1..7** | GPIO_BA+0x360+ 0x04 * n | R/W | GPIO P3.n 端口数据输入/输出 | 0x0000_000X |
| **P4_N n=0,1..7** | GPIO_BA+0x380+ 0x04 * n | R/W | GPIO P4.n 端口数据输入/输出 | 0x0000_000X |
| **P5_N n=0,1..7** | GPIO_BA+0x3A0+ 0x04 * n | R/W | GPIO P5.n 端口数据输入/输出 | 0x0000_000X |
| **P6_N n=0,1..7** | GPIO_BA+0x3C0+ 0x04 * n | R/W | GPIO P6.n 端口数据输入/输出 | 0x0000_000X |
| **P7_N n=0,1..7** | GPIO_BA+0x3E0+ 0x04 * n | R/W | GPIO P7.n 端口数据输入/输出 | 0x0000_000X |
| **P8_N n=0,1..7** | GPIO_BA+0x400+ 0x04 * n | R/W | GPIO P8.n 端口数据输入/输出 | 0x0000_000X |
| **P9_N n=0,1..7** | GPIO_BA+0x420+ 0x04 * n | R/W | GPIO P9.n 端口数据输入/输出 | 0x0000_000X |
| **PA_N n=0,1** | GPIO_BA+0x440+ 0x04 * n | R/W | GPIO PA.n 端口数据输入/输出 | 0x0000_000X |

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

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **Pxn** | **GPIO Px.n管脚 数据输入/输出**<br>写该位可以控制一个GPIO管脚的输出值<br>0 = 设置相应GPIO管脚为低 |

| | | |
|---|---|---|
| | | 1 = 设置相应GPIO管脚为高<br>读该寄存器得到GPIO管脚状态<br>例如: 写P0_0即把值写到P0_DOUT[0]位上, 读P0_0即读取P0_PIN[0]的值.<br>注意：写操作不受Px_DMASK影响 |

## 6.6 定时器控制器 (TIMER)

### 6.6.1 概述

定时器控制器包含 4 组 32位定时器，TIMER0~TIMER3，提供用户便捷的计数定时功能。定时器可执行很多功能，如频率测量，时间延迟，时钟发生，外部输入管脚事件计数和外部捕捉管脚脉宽测量等。

### 6.6.2 特性

- 4 组 32位定时器，带24位向上定时器和一个8位的预分频计数器
- 每个定时器都有独立的时钟源
- 提供 单次, 周期, 触发输出和 连续计数 四种计数操作模式
- 通过TDR (TDR[23:0]) （定时器数据寄存器）可读取内部 24 位向上计数器的值
- 支持事件计数功能
- 支持外部管脚捕捉功能，可用于脉宽测量
- 支持外部管脚捕捉事件复位24位向上计数器

### 6.6.3 框图

定时器控制器的框图和时钟控制如图 6-21 和 图 6-22所示

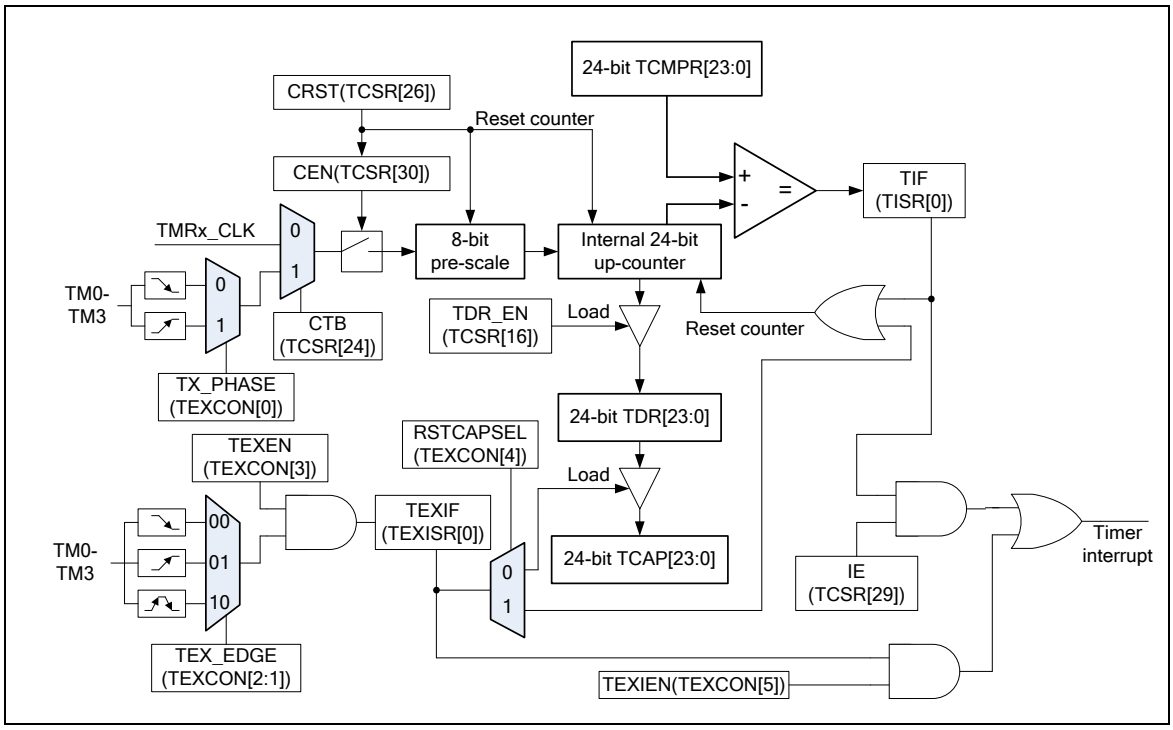



图 6-21 定时器控制器框图

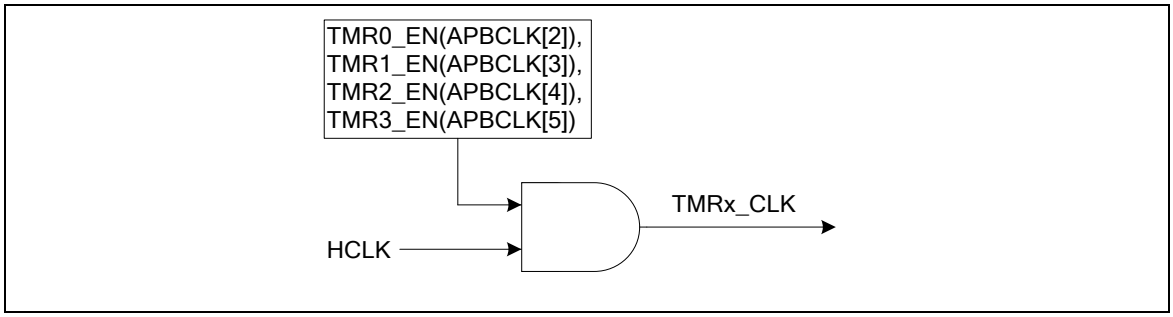



图 6-22 定时器控制器时钟源

### 6.6.4 基本配置

Timer0 ~ Timer3的HCLK时钟源可以通过设置TMRx_EN (APBCLK[5:2])使能

### 6.6.5 功能描述

#### *6.6.5.1 定时器中断标志*

定时器控制器支持两个中断标志：一个是TIF标志，该标志在当定时器计数器值(TDR)与定时器比较值(TCMP)相匹配时置位，另一个是TEXIF标志，该标志在当TMx管脚的变化与TEX_EDGE的设置一致时置位。

#### *6.6.5.2 定时器计数模式*

定时器控制器提供4种定时计数模式：单次、周期、触发输出和连续计数模式。

#### *6.6.5.3 单次模式*

如果定时器工作在单次模式(TCSR[28:27]为00)且 CEN (TCSR[30] 定时器使能位)置1，则定时器的计数器开始计数。一旦定时器计数器的值达到定时器比较器寄存器 (TCMP) 的值时，TIF标志将变为1，TDR的值和CEN位将由定时器控制器清零，然后定时器计数操作停止。与此同时，如果 IE（TCSR[29] 中断使能位）使能，则定时器中断信号产生并送到 NVIC 通知 CPU。

#### *6.6.5.4 周期模式*

如果定时器工作在周期模式(TCSR[28:27]为01)且 CEN置1，则定时器的计数器开始计数。一旦定时器计数器的值达到定时器比较器寄存器 (TCMP) 的值时，TIF标志将变为1，TDR的值将由定时器控制器清零，然后定时器重新计数。与此同时，如果 IE使能，则定时器中断信号产生并送到 NVIC 通知 CPU 。在该模式，定时器控制器周期性地操作计数和 与TCMPR 的值比较，直到 CEN位由软件清0。

#### *6.6.5.5 触发输出 模式*

如果定时器工作在触发输出模式(TCSR[28:27]为10)且 CEN置1，则定时器的计数器开始计数。触发输出模式的计数操作大部分与周期模式是一样的，除了该模式当TIF位设置时，有相关的TM0~TM3管脚来输出信号，因此，管脚TM0~TM3上的触发输出信号以 50% 的占空周期反复改变。

#### *6.6.5.6 连续计数模式*

如果定时器工作在连续计数模式(TCSR[28:27]为11)且 CEN置1，则定时器的计数器开始计数。一旦定时器计数器(TDR)的值达到定时器比较器寄存器 (TCMP) 的值时，TIF标志将变为1，但TDR的值继续保持向上计数。与此同时，如果 IE使能，则定时器中断信号产生并送到 NVIC 通知 CPU 。在该模式，用户可以立刻改变不同的TCMP值，而不需要停止定时器计数和重新开始定时器计数。

例如，先把定时器比较寄存器 (TCMP)的值设置为 80。当定时器计数器的值 (TDR 的值) 达到 80 时,TIF 标志将被置1，定时器计数器继续计数，而且TDR的值将不回到0，而是继续计数，81, 82, 83,⋯ 到 ($2^{24}$ -1)，然后再一次 0, 1, 2, 3, ⋯ 到$2^{24}$ -1，如此往复。接下来，如果软件改变TCMP的

值为200并且清除TIF标志位，当TDR的值达到200时，TIF标志将再次变为1,。最后，软件改变TCMP的值为500并且清除TIF标志，当TDR的值达到500时，TIF标志将再次变为1。

在该模式，计数器计数时连续的。所以该操作模式叫做连续计数模式。

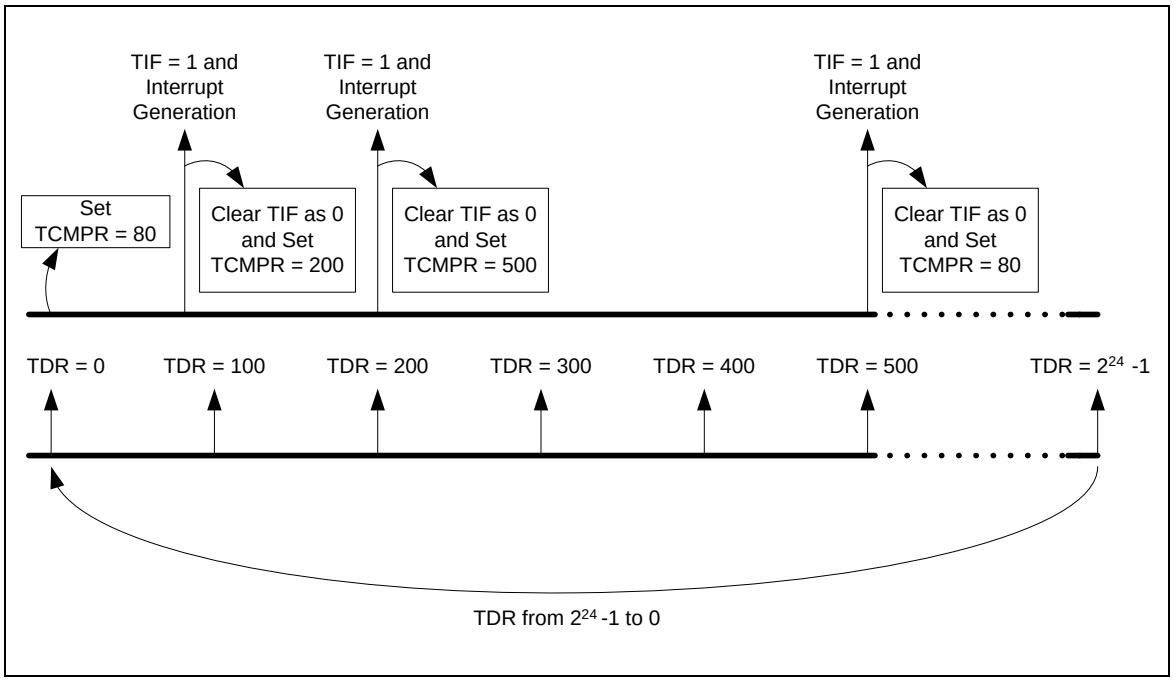



图 6-23 连续计数模式

### 6.6.5.7 *事件计数模式*

定时器控制器也提供这样的应用，能对输入事件 (来自管脚 TMx x=0~3) 计数并将事件的次数反应到TDR的值。也可以称为事件计数功能。该功能下，CTB(TCSR[24])位需置位并且定时器外设时钟源必须设为HCLK。

软件可以通过 TCDB(TEXCON[7]) 位来使能或关闭TMx管脚消抖电路。如果 TMx 管脚的消抖电路关闭，输入事件频率必须少于 1/3HCLK，如果消抖电路打开，输入事件的频率须小于 1/8HCLK，以保证TDR的值是正确的。软件也可以通过设置 TX_PHASE(TEXCON[0]) 来选择边沿检测 TMx 管脚的相位。

事件计数模式下，定时器计数操作模式可以设置为单次，周期，和连续计数模式来计算来自TMx管脚的输入事件TDR的值。

### 6.6.5.8 *外部捕捉模式*

事件捕捉功能是当检测到 TMx_EXT管脚(x=0~3) 边沿电平有变化时，定时器计数器值(TDR)会送到捕捉数据寄存器 (TCAP) 。在该模式下，需把 RSTCAPSEL(TEXCON[4]) 位设置为0，用来选择TMx_EXT变化时用作事件捕捉功能，而且定时器外设时钟源必须设为HCLK。

软件可以通过 TEXDB(TEXCON[6]) 位来使能或关闭 TMx_EXT 管脚消抖电路。在 TMx_EXT 的消抖电路关闭时，TMx_EXT 管脚的转变频率必须少于 1/3 HCLK，在 TMx_EXT 的消抖电路打开时，TMx_EXT 管脚的转变频率必须少于1/8 HCLK，以保证捕捉功能能够正常工作。软件也可以通过设置TEX_EDGE(TEXCON[2:1]) 位来选择 TMx_EXT 管脚的边沿转变检测方式。

在事件捕捉模式，软件不用考虑定时器计数器工作模式的选择，捕捉事件的发生只有当检测到 TMx_EXT 管脚有边沿变化时。

用户必须考虑到当CPU没有清TEXIF状态时，定时器会保持TCAP不变并且会丢弃新捕捉的值。

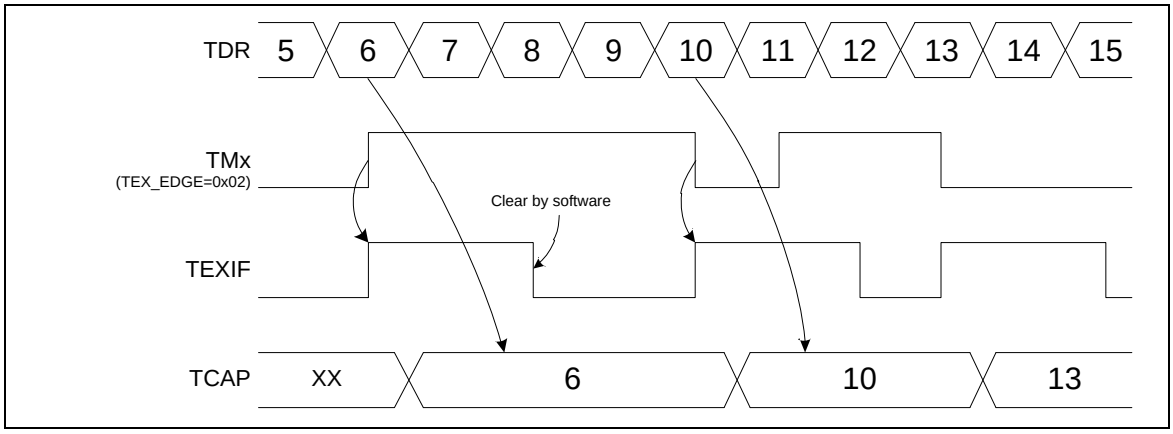



图 6-24 外部捕捉模式

*6.6.5.9 外部事件复位计数模式*

当检测到 TMx_EXT管脚(x=0~3) 有边沿变化时，定时器同样提供事件复位计数器功能来复位 TDR 的值。在该模式，大部分设置与事件捕捉功能相同，除了 RSTCAPSEL(TEXCON[4]) 位必须设置为1来选择 TMx_EXT 转变时用作为事件复位计数器。

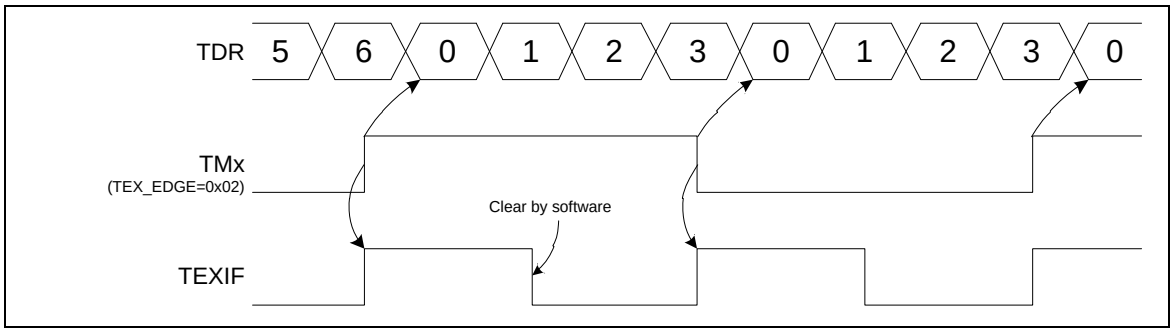



图 6-25 外部事件复位计数器模式

### 6.6.6 寄存器映射

**R**: 只读, **W**: 只写, **R/W**: 读/写

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **TMR 基地址:**<br>**TMR01_BA = 0x4001_0000**<br>**TMR23_BA = 0x4011_0000** | | | | |
| **TCSR0** | TMR01_BA+0x00 | R/W | Timer0控制和状态寄存器 | 0x0000_0005 |
| **TCMPR0** | TMR01_BA+0x04 | R/W | Timer0比较寄存器 | 0x0000_0000 |
| **TISR0** | TMR01_BA+0x08 | R/W | Timer0中断状态寄存器 | 0x0000_0000 |
| **TDR0** | TMR01_BA+0x0C | R | Timer0数据寄存器 | 0x0000_0000 |
| **TCAP0** | TMR01_BA+0x10 | R | Timer0捕捉数据寄存器 | 0x0000_0000 |
| **TEXCON0** | TMR01_BA+0x14 | R/W | Timer0外部控制寄存器 | 0x0000_0000 |
| **TEXISR0** | TMR01_BA+0x18 | R/W | Timer0外部中断状态寄存器 | 0x0000_0000 |
| **TCSR1** | TMR01_BA+0x20 | R/W | Timer1控制和状态寄存器 | 0x0000_0005 |
| **TCMPR1** | TMR01_BA+0x24 | R/W | Timer1比较寄存器 | 0x0000_0000 |
| **TISR1** | TMR01_BA+0x28 | R/W | Timer1中断状态寄存器 | 0x0000_0000 |
| **TDR1** | TMR01_BA+0x2C | R | Timer1数据寄存器 | 0x0000_0000 |
| **TCAP1** | TMR01_BA+0x30 | R | Timer1捕捉数据寄存器 | 0x0000_0000 |
| **TEXCON1** | TMR01_BA+0x34 | R/W | Timer1外部控制寄存器 | 0x0000_0000 |
| **TEXISR1** | TMR01_BA+0x38 | R/W | Timer1外部中断状态寄存器 | 0x0000_0000 |
| **TCSR2** | TMR23_BA+0x00 | R/W | Timer2控制和状态寄存器 | 0x0000_0005 |
| **TCMPR2** | TMR23_BA+0x04 | R/W | Timer2比较寄存器 | 0x0000_0000 |
| **TISR2** | TMR23_BA+0x08 | R/W | Timer2中断状态寄存器 | 0x0000_0000 |
| **TDR2** | TMR23_BA+0x0C | R | Timer2数据寄存器 | 0x0000_0000 |
| **TCAP2** | TMR23_BA+0x10 | R | Timer2捕捉数据寄存器 | 0x0000_0000 |
| **TEXCON2** | TMR23_BA+0x14 | R/W | Timer2外部控制寄存器 | 0x0000_0000 |
| **TEXISR2** | TMR23_BA+0x18 | R/W | Timer2外部中断状态寄存器 | 0x0000_0000 |
| **TCSR3** | TMR23_BA+0x20 | R/W | Timer3控制和状态寄存器 | 0x0000_0005 |
| **TCMPR3** | TMR23_BA+0x24 | R/W | Timer3比较寄存器 | 0x0000_0000 |
| **TISR3** | TMR23_BA+0x28 | R/W | Timer3中断状态寄存器 | 0x0000_0000 |
| **TDR3** | TMR23_BA+0x2C | R | Timer3数据寄存器 | 0x0000_0000 |

| | | | | |
|---|---|---|---|---|
| **TCAP3** | TMR23_BA+0x30 | R | Timer3捕捉数据寄存器 | 0x0000_0000 |
| **TEXCON3** | TMR23_BA+0x34 | R/W | Timer3外部控制寄存器 | 0x0000_0000 |
| **TEXISR3** | TMR23_BA+0x38 | R/W | Timer3外部中断状态寄存器 | 0x0000_0000 |

### 6.6.7 寄存器描述

**定时器控制和状态寄存器(TCSR)**

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **TCSR0** | TMR01_BA+0x00 | R/W | 定时器0控制和状态寄存器 | 0x0000_0005 |
| **TCSR1** | TMR01_BA+0x20 | R/W | 定时器1控制和状态寄存器 | 0x0000_0005 |
| **TCSR2** | TMR23_BA+0x00 | R/W | 定时器2控制和状态寄存器 | 0x0000_0005 |
| **TCSR3** | TMR23_BA+0x20 | R/W | 定时器3控制和状态寄存器 | 0x0000_0005 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| DBGACK_TMR | CEN | IE | MODE | | CRST | CACT | CTB |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | TDR_EN |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PRESCALE | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31] | **DBGACK_TMR** | **仿真器 (ICE) 调试模式响应禁止位 （写保护位）**<br>0 = 仿真器调试模式响应影响定时器计数。<br>当调试器调试模式响应时，定时器计数器将被固定住。<br>1 = 仿真器调试模式响应禁用。<br>无论调试器调试模式响应与否，定时器计数器将持续计数下去。 |
| [30] | **CEN** | **定时器使能位**<br>0 = 停止/暂停计数<br>1 = 开始计数<br>**注意1：**在停止状态，设置 CEN 为 1 将使能 24位向上计数器从上次停止的计数值继续计数。<br>**注意2：**在 单次 模式下 (MODE [28:27] =00)，当相应的定时中断标志(TISR[0] TIF)产生时，该位由硬件自动清零。 |
| [29] | **IE** | **中断使能位**<br>0 = 禁用定时器中断<br>1 = 使能定时器中断<br>如果该位使能，当定时器中断标志TIF (TISR[0]) 置 1， 定时器中断信号产生并通知CPU。 |

| | | |
|---|---|---|
| [28:27] | **MODE** | **定时器操作模式**<br>00 = 定时器工作在单触发模式 (one-shot)<br>01 = 定时器工作在周期模式 (period)<br>10 = 定时器工作在触发输出模式(Toggle-output)<br>11 = 定时器工作在连续计数模式(Continuous Counting） |
| [26] | **CRST** | **定时器复位**<br>0 = 该位写 0 无效<br>1 =如果CACT为1，复位定时器的8位预分频计数器，内部24位向上计数器的值和CEN位 |
| [25] | **CACT** | **定时器激活状态位（只读）**<br>该位表示当前定时器计数器的状态。<br>0 = 定时器计数器未激活<br>1 = 定时器计数器激活 |
| [24] | **CTB** | **计数模式使能位**<br>该位用来使能外部计数管脚功能。<br>0 = 禁用计数器模式<br>1 = 使能计数器模式 |
| [23:17] | **Reserved** | 保留 |
| [16] | **TDR_EN** | **数据加载使能位**<br>当该位置1，计数器正在计数时，定时器计数器值(TDR) 会不断更新来监视内部24位向上计数器值。<br>0 = 定时器数据寄存器更新禁止<br>1 = 当定时器计数器激活， 定时器数据寄存器更新使能 |
| [15:8] | **Reserved** | 保留 |
| [7:0] | **PRESCALE** | **预分频计数器**<br>时钟输入根据 PRESCALE +1 进行预分频。如果 PRESCALE 数值为0，不进行预分频。 |

## 定时器比较寄存器 (TCMPR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **TCMPR0** | TMR01_BA+0x04 | R/W | 定时器0比较寄存器 | 0x0000_0000 |
| **TCMPR1** | TMR01_BA+0x24 | R/W | 定时器1比较寄存器 | 0x0000_0000 |
| **TCMPR2** | TMR23_BA+0x04 | R/W | 定时器2比较寄存器 | 0x0000_0000 |
| **TCMPR3** | TMR23_BA+0x24 | R/W | 定时器3比较寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:0] | **TCMP** | **定时器比较值**<br>TCMP 是24位比较寄存器。当内部 24位向上计数器的值与 TCMP 的值相等时，TIF标志将被置1。<br>超时周期 = (定时器输入时钟周期) * (8-bit PRESCALE + 1) * (24-bit TCMP)<br>**注意1：** 不能向 TCMP 里写 0x0 或 0x1，否则内核将运行到未知状态。<br>**注意2：** 当定时器工作在 连续计数模式，即使软件写一个新的值到 TCMP，24位向上计数定时器将保持继续计数。如果定时器工作在其他模式，如果软件写一个新的值到 TCMP，定时器将使用新比较值并退出当前计数，开始重新计数。 |

## 定时器中断状态寄存器(TISR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **TISR0** | TMR01_BA+0x08 | R/W | 定时器0中断状态寄存器 | 0x0000_0000 |
| **TISR1** | TMR01_BA+0x28 | R/W | 定时器1中断状态寄存器 | 0x0000_0000 |
| **TISR2** | TMR23_BA+0x08 | R/W | 定时器2中断状态寄存器 | 0x0000_0000 |
| **TISR3** | TMR23_BA+0x28 | R/W | 定时器3中断状态寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **TIF** | **定时器中断标志**<br>当内部 24位向上计数定时器与定时器比较值 (TCMP)匹配时，定时器中断状态标志位。.<br>0 = 无影响<br>1 = 计数定时器与TCMP的值相匹配<br>该位写 1 清零。 |

## 定时器数据寄存器(TDR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **TDR0** | TMR01_BA+0x0C | R | 定时器0数据寄存器 | 0x0000_0000 |
| **TDR1** | TMR01_BA+0x2C | R | 定时器1数据寄存器 | 0x0000_0000 |
| **TDR2** | TMR23_BA+0x0C | R | 定时器2数据寄存器 | 0x0000_0000 |
| **TDR3** | TMR23_BA+0x2C | R | 定时器3数据寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:0] | **TDR** | **定时器数据寄存器**<br>该位段反映了内部24位定时器的值或是来自TMx (x=0~3) 管脚的外部事件输入计数器的值<br>如果CTB (TCSR[24] ) =0，用户可以读TDR的值来得到当前24位计数器值<br>如果CTB (TCSR[24] ) =1，用户可以读TDR的值来得到当前24位外部事件输入计数器的值 |

## 定时器捕捉数据寄存器(TCAP)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **TCAP0** | TMR01_BA+0x10 | R | 定时器0捕捉数据寄存器 | 0x0000_0000 |
| **TCAP1** | TMR01_BA+0x30 | R | 定时器1捕捉数据寄存器 | 0x0000_0000 |
| **TCAP2** | TMR23_BA+0x10 | R | 定时器2捕捉数据寄存器 | 0x0000_0000 |
| **TCAP3** | TMR23_BA+0x30 | R | 定时器3捕捉数据寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:0] | **TCAP** | **定时器捕捉数据寄存器**<br>当TEXEN (TEXCON[3])被置1，RSTCAPn (TEXCON[4])被清0，TMx (x=0~3)管脚有一个转变正好符合TEX_EDGE (TEXCON[2:1])设置时，TEXIF (TEXISR[0])会被置1，当前内部 24位向上计数定时器的值将立刻自动被载入到 TCAP域。 |

## 定时器外部控制寄存器(TEXCON)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **TEXCON0** | TMR01_BA+0x14 | R/W | 定时器0外部控制寄存器 | 0x0000_0000 |
| **TEXCON1** | TMR01_BA+0x34 | R/W | 定时器1外部控制寄存器 | 0x0000_0000 |
| **TEXCON2** | TMR23_BA+0x14 | R/W | 定时器2外部控制寄存器 | 0x0000_0000 |
| **TEXCON3** | TMR23_BA+0x34 | R/W | 定时器3外部控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **TCDB** | **TEXDB** | **TEXIEN** | **RSTCAPSEL** | **TEXEN** | **TEX_EDGE** | | **TX_PHASE** |

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留 |
| [7] | **TCDB** | **定时器外部计数器输入管脚防抖动使能位**<br>0 = 禁用TMx (x= 0~3)管脚防抖动<br>1 = 使能TMx (x= 0~3)管脚防抖动<br>如果该位使能，TMx管脚边沿检测带防抖动电路。 |
| [6] | **TEXDB** | **定时器外部捕捉输入管脚防抖动使能位**<br>0 = TMx (x= 0~3)管脚禁用防抖动<br>1 = TMx (x= 0~3)管脚使能防抖动<br>如果该位使能，TMx管脚边沿检测带防抖动电路。 |
| [5] | **TEXIEN** | **定时器外部捕捉中断使能位**<br>0 = TMx (x= 0~3)管脚检测中断禁止<br>1 = TMx (x= 0~3)管脚检测中断使能<br>如果TEXIEN 使能，当TEXIF 标志设为1时，定时器会产生一个外部捕捉中断信号并通知CPU发生了中断。 |
| [4] | **RSTCAPSEL** | **定时器外部复位计数器/ 定时器外部捕捉模式选择**<br>0 = TMx (x= 0~3)管脚上的转变用来保存24位定时器计数器(TDR)值到定时器捕捉值寄存器(TCAP)上(事件捕捉功能)<br>1 = TMx_EXT管脚上的转变用来复位24位定时器计数器(事件复位计数器功能) |
| [3] | **TEXEN** | **定时器外部捕捉管脚使能位**<br>该位使能TMx_管脚上的RSTCAPSEL功能。<br>0 = TMx (x= 0~3)管脚上的RSTCAPSEL功能忽略。<br>1 = TMx (x= 0~3)管脚上的RSTCAPSEL功能激活。 |
| [2:1] | **TEX_EDGE** | **定时器外部捕捉管脚边沿检测选择**<br>00 = Tx_EXT (x= 0~3)管脚下降沿将被检测。<br>01 = Tx_EXT (x= 0~3)管脚上升沿将被检测。 |

| | | |
|---|---|---|
| | | 10 = Tx_EXT (x= 0~3)管脚上升沿或下降沿将被检测。<br>11 = 保留。 |
| [0] | **TX_PHASE** | **定时器外部计数管脚相位检测选择**<br>该位表示TMx (x= 0~3).管脚相位检测。<br>0 = TMx管脚的下降沿将被统计。<br>1 = TMx管脚的上升沿将被统计。 |

## 定时器外部中断状态寄存器 (TEXISR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **TEXISR0** | TMR01_BA+0x18 | R/W | 定时器0外部中断状态寄存器 | 0x0000_0000 |
| **TEXISR1** | TMR01_BA+0x38 | R/W | 定时器1外部中断状态寄存器 | 0x0000_0000 |
| **TEXISR2** | TMR23_BA+0x18 | R/W | 定时器2外部中断状态寄存器 | 0x0000_0000 |
| **TEXISR3** | TMR23_BA+0x38 | R/W | 定时器3外部中断状态寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留 |
| [0] | **TEXIF** | **定时器外部捕捉中断标志位**<br>该位表示外部捕捉中断标志状态。.<br>当TEXEN (TEXCON[3]使能，且RSTCAPn (TEXCON[4]) 清 0，在TMx (x= 0~3)管脚边沿转变与TEX_EDGE (TEXCON[2:1]) 设定相匹配时，该标志将被硬件置位。<br>0 = TMx (x= 0~3)管脚中断没有发生.<br>1 = TMx (x= 0~3)管脚中断发生.<br>**注意1：**该位写 1 清零。<br>**注意2：**在CPU清TEXIF前又有一个新的捕捉事件被侦测使，定时器会保持TCAP值不变并且会丢弃新的捕捉值 |

## 6.7 基本PWM发生器和捕获定时器 (BPWM)

### 6.7.1 概述

NuMicro® M0519系列有一组BPWM发生器(BPWM0)，这一组发生器可以配置成BPWM0_CH0 和BPWM0_CH1两路独立的BPWM输出，或是一组互补的BPWM对 (BPWM0_CH0, BPWM0_CH1)可配置成带死区功能输出。

BPWM发生器有一个8位预分频器、一个带有可选5种除频(1, 1/2, 1/4, 1/8, 1/16)的时钟除频器、2组BPWM定时器且有两路时钟选择、2组16位下数计数器用于BPWM周期控制、2组16位比较器用于调节BPWM的占空比和死区产生。BPWM发生器提供2个独立的BPWM中断标志，当BPWM周期下数计数器减到0时该标志位被硬件置位。

每一路BPWM中断源对应的使能位都可以让CPU响应BPWM中断。BPWM发生器可以配置成单次触发模式（BPWM产生一次脉冲信号）和自动加载模式（连续不断的输出波形）。BPWM可以用来触发工作在中心对齐模式下的EADC。

### 6.7.2 特性

#### *6.7.2.1 BPWM 功能:*

- 多达1组BPWM且支持2个BPWM通道或1对互补BPWM通道
- 支持1到255的8位预分频
- 支持16位分辨率定时器
- PWM定时器支持边沿对齐和中心对齐工作类型
- 支持单次和自动装载模式
- PWM中断请求与BPWM周期或是占空比同步
- BPWM互补模式下支持死区发生且分辨率为8位
- 支持触发EADC

#### *6.7.2.2 捕获功能:*

- 支持2路捕获输入和2路BPWM输出通道共享
- 支持上升沿或下降沿捕获
- 支持上升沿或是下降沿捕获中断

### 6.7.3 框图

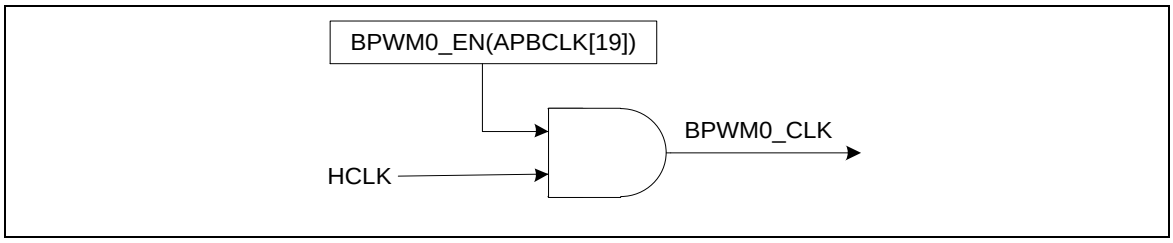



图 6-26 BPWM 时钟源控制

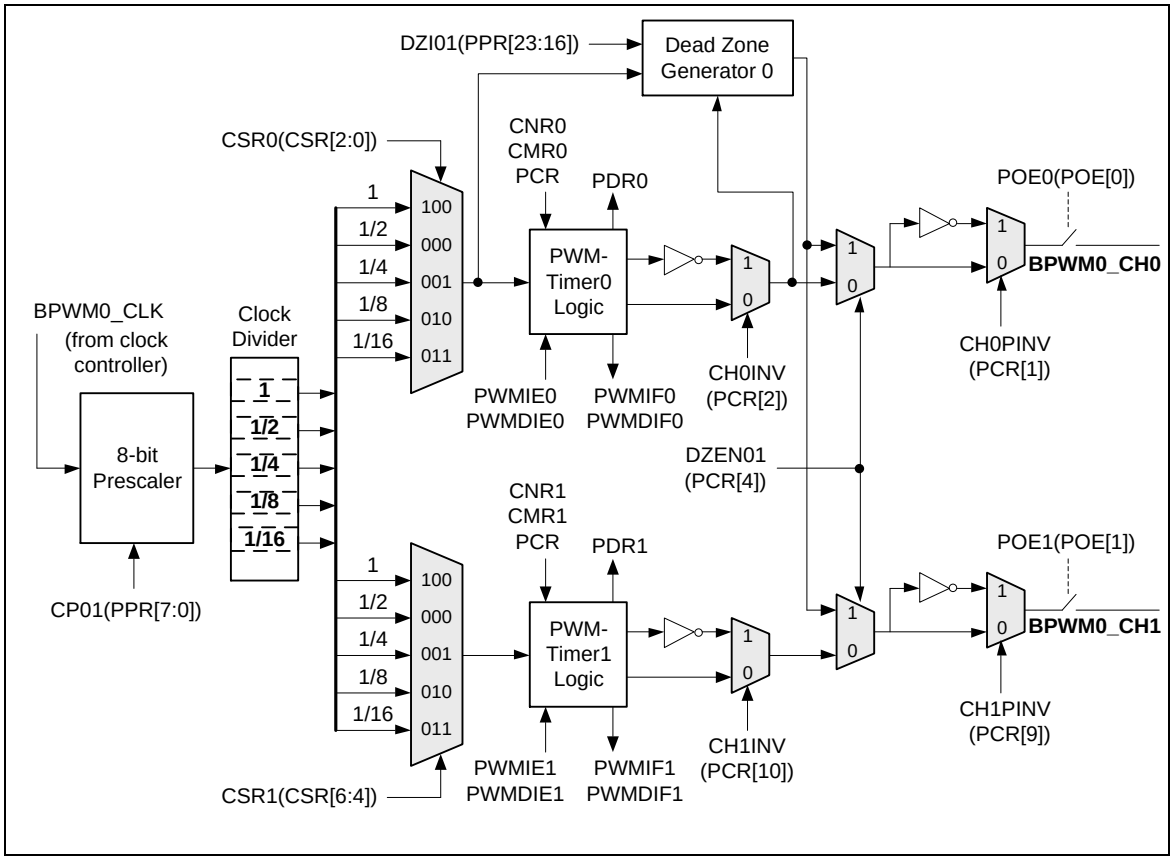



图 6-27 BPWM 结构图

### 6.7.4 基本配置

BPWM管脚功能通过配置P5_MFP[7:6]多功能引脚寄存器开启。BPWM外设时钟源通过设置BPWM0_EN (APBCLK[19])打开。配置PWMPOEN[4] PWM端口输出使能寄存器控制BPWM端口输出驱动模式

### 6.7.5 功能描述

BPWM控制器支持2种工作模式：边沿对齐和中心对齐模式

#### 6.7.5.1 *边沿对齐 BPWM (下数计数器)*

边沿对齐BPWM输出模式下，16位BPWM计数器从周期值(CNRn 计数器)开始下数到0完成一个BPWM周期，如果自动装载模式开启（置PCR寄存器的CHnMOD为1）计数器会重新装载周期值继续下数到0。BPWM计数器的值会和CMRn值比较，用来控制BPWM发生器的电平输出。当BPWM计数器的值大于CMRn值时BPWM发生器输出低电平，当BPWM计数器的值小于或等于CMRn值时BPWM发生器输出高电平。

设置BPWMPIEn (PIER[1:0])为1将触发BPWM周期中断(BPWMIFn (PIIR[1:0]))，置BPWMDIEn (PIER[9:8])为1将触发BPWM占空比中断(BPWMDIFn (PIIR[9:8]))。

通过配置下数计数器寄存器CNR控制BPWM周期，通过配置比较器寄存器CMR控制BPWM占空比。新的周期和比较值会在设置后的下一个周期生效。BPWM定时器工作时序如图 6-29。脉冲宽度调节依据下面的公式，BPWM定时器比较器说明如图 6-28。注对于相应的BPWM通道与其对应的GPIO管脚必须配置成BPWM功能（使能POE，禁用CAPENR）。

- BPWM 频率 = BPWM0_CLK/[(prescale+1)*(clock divider)*(CNR+1)].
- 占空比 = (CMR+1)/(CNR+1)
- CMR >= CNR: BPWM 输出总为高
- CMR < CNR: BPWM 低电平宽度= (CNR-CMR) 单元[1]; BPWM 高电平宽度 = (CMR+1) 单元
- CMR = 0: BPWM低电平宽度= (CNR) 单元; BPWM高电平宽度= 1单元

**注:** [1] 单元 =一个BPWM时钟周期

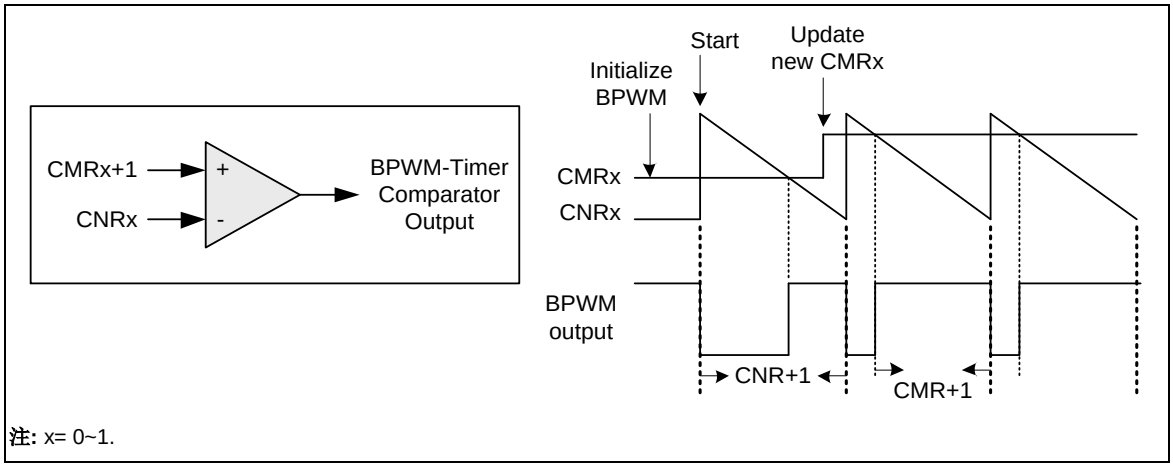



图 6-28 BPWM定时器内部比较器输出图形说明

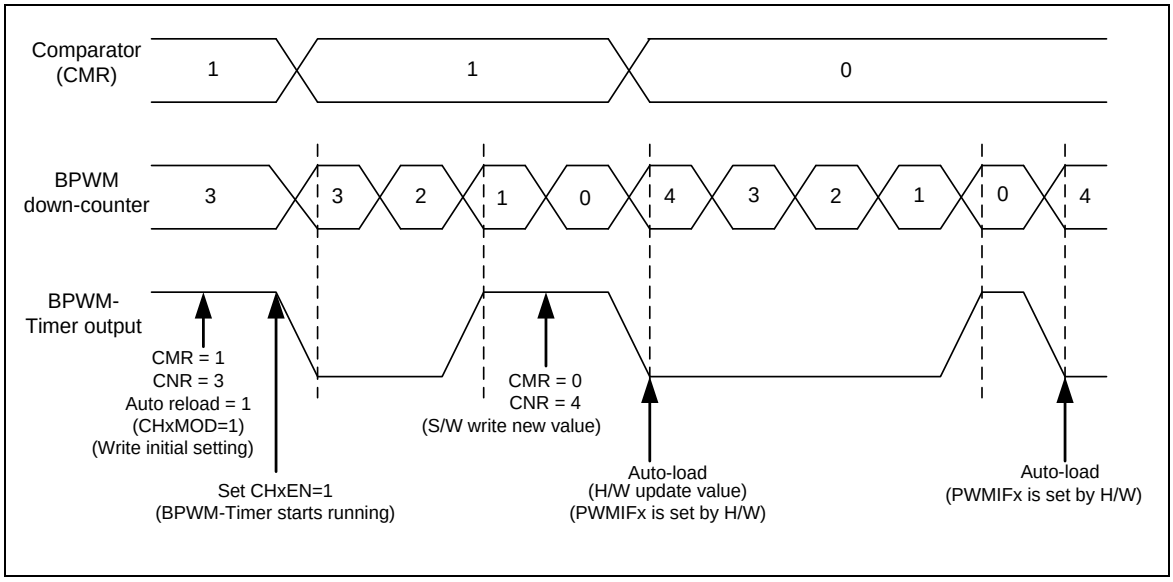



图 6-29 BPWM定时器工作时序

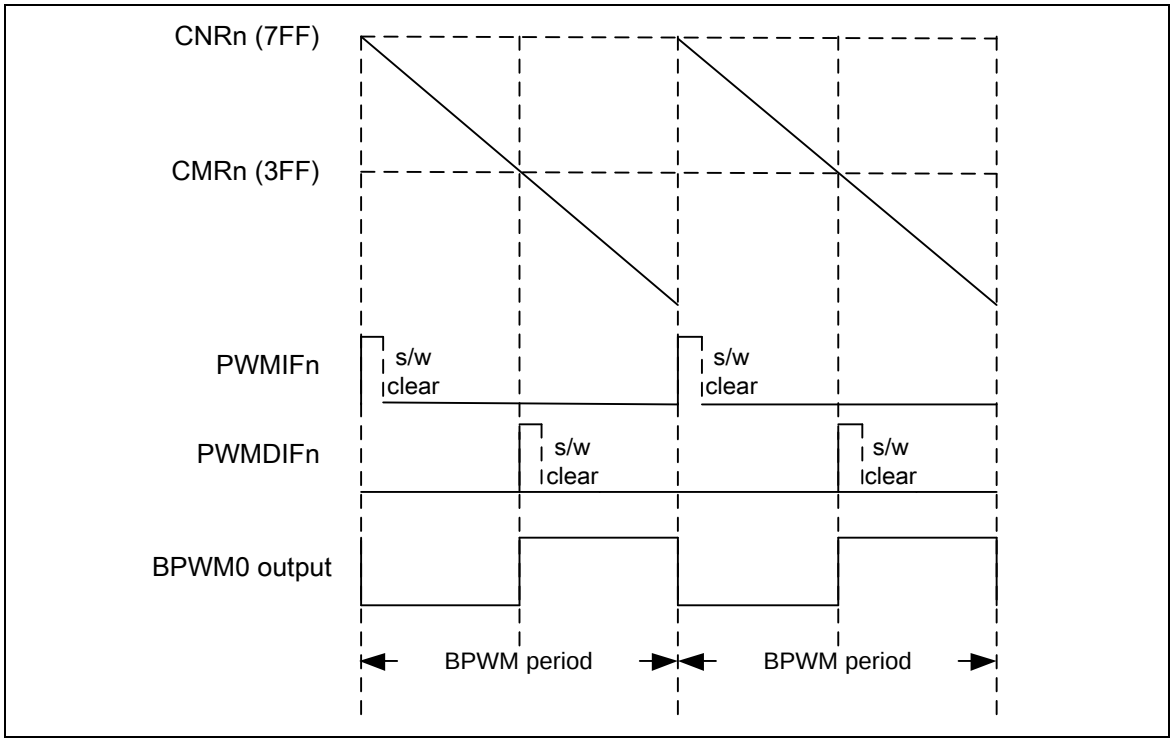



图 6-30 BPWM 边沿对齐中断产生时序波形

### 6.7.5.2 *中心对齐 BPWM (上/下数计数器)*

当BPWM定时器配置为上/下数型计数器模式时，中心对齐BPWM信号将产生。 BPWM计数器从0开始上数到与CNRn的值匹配然后开始下数到0完成一次BPWM周期，如果自动装载模式开启（置PCR寄存器的CHnMOD为1）将会重新执行下一次BPWM周期。BPWM计数器的值会和CMRn值比较，用来控制BPWM发生器的电平输出。当BPWM计数器的值大于CMRn值时BPWM发生器输出低

电平，当BPWM计数器的值小于或等于CMRn值时BPWM发生器输出高电平。在自动加载模式下，一旦BPWM 计数器下溢，新的BPWM周期和比较值将会生效.

中心对齐模式下, 如果INTTYPE (PIER[16]) =0，当计数器下溢时(也就是每个周期的开始和结束)BPWM周期中断将发生; 如果INTTYPE (PIER[16]) =1，当上数计数器与CNRn的值匹配时(也就是PWM周期的中点)PWM周期中断将发生.

- BPWM 频率 = BPWM0_CLK/[(prescale+1)*(clock divider)*(CNR+1)].
- Duty ratio = (CMR+1)/(CNR+1)
- CMR >= CNR: BPWM 输出总为高
- CMR < CNR: BPWM 低电平宽度= (CNR-CMR) 单元[1]; BPWM 高电平宽度 = (CMR+1) 单元
- CMR = 0: BPWM低电平宽度= (CNR) 单元; BPWM高电平宽度= 1单元

**注:** [1] 单元 =一个BPWM时钟周期

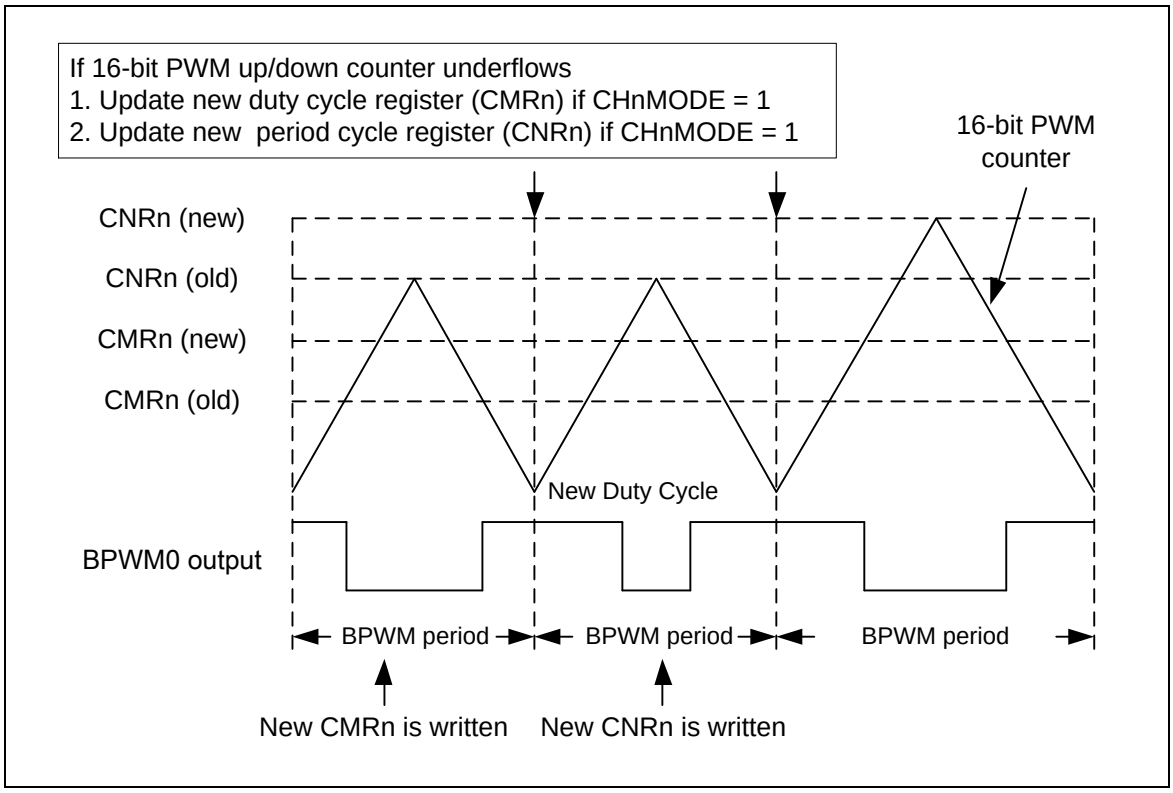



图 6-31 中心对齐输出波形

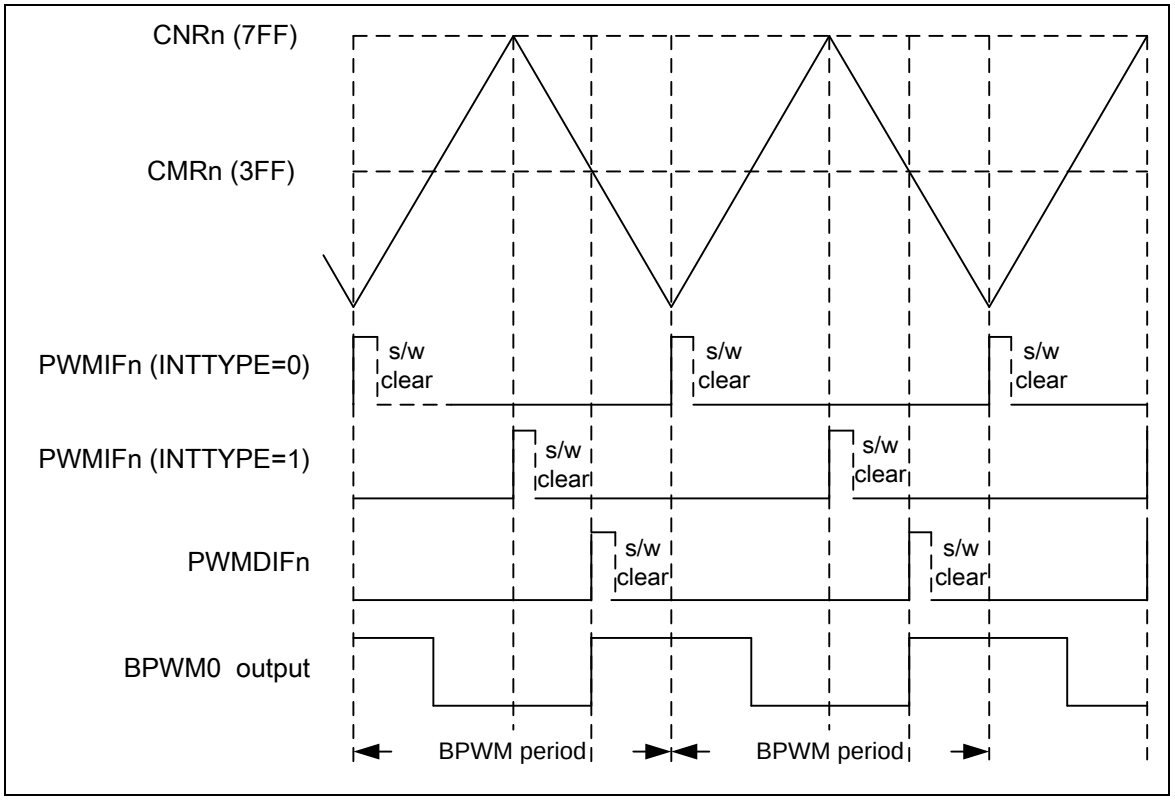



图 6-32 BPWM中心对齐中断产生时序波形

### *6.7.5.3 BPWM 双缓冲，自动加载 和单次触发操作*

为了避免BPWM驱动输出不稳定的波形，16位的周期下数计数器和16位比较器运行时都带有双缓冲。当用户写数据到计数器/比较器的缓冲寄存器时，更新的值将在下一个周期起作用不会影响当前定时器工作，这样就避免BPWM输出干扰脉冲。BPWM计数器值被写入CNRn，当前的BPWM计数器值可以从PDRn读出。

如果CH0MOD 位清0，BPWM0_CH0将会工作在One-shot模式下。如果CH0MOD 位置1，BPWM0_CH0将会工作在自动加载模式下。建议在CH0EN位置1（使能BPWM0_CH0计数器开始计数）前切换BPWM0_CH0的工作模式，因为在模式切换时CNR0 和 CMR0里的值会被清0用来复位BPWM0_CH0周期和占空比的配置。

BPWM0_CH0工作在单次触发模式下时，首先写入CNR0 和 CMR0然后置CH0EN位为1启动BPWM0_CH0计数器。在BPWM0_CH0计数器从CNR0值下数到0时，CNR0 和 CMR0会被硬件清0,并且BPWM计数器会被保持住。软件需要写新的CNR0 和 CMR0值用来设置下次单次触发的周期和占空比。当重新下次单次触发工作时，应该首先写入CMR0的值，因为当CNR0写入一个非0值时BPWM0_CH0计数器会自动重新计数。

BPWM0_CH0工作在自动加载模式下时，首先写入CNR0 和 CMR0然后置CH0EN位为1启动BPWM0_CH0计数器。在BPWM0_CH0计数器从CNR0值下数到0时，CNR0 的值会重新被加载到BPWM0_CH0计数器里。如果CNR0被设置为0，BPWM0_CH0计数器会被保持住。

BPWM0_CH1的功能与BPWM0_CH0相同。

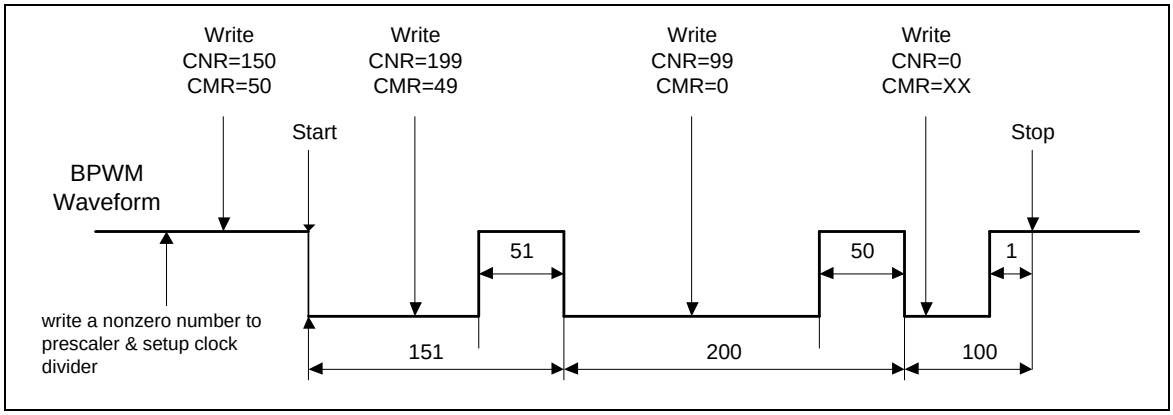



图 6-33 BPWM双缓冲说明

### 6.7.5.4 *占空比调制*

双缓冲功能允许在任何时间更新CMRn.新加载的值在下个周期起作用.

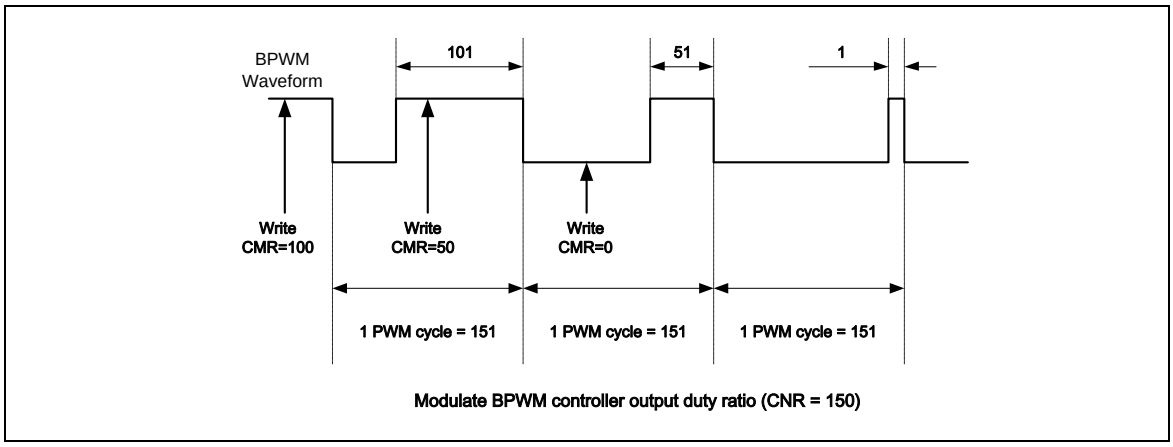



图 6-34 BPWM 控制输出占空比

### 6.7.5.5 *死区发生器*

互补模式下，死区发生器在关闭两个成对的引脚之中的一个到打开另外一个互补的引脚之间插入称为 “死区”的“off” 周期，这可以防止电源切换时设备被破坏。互补对输出模式有一个8位的下数计数器DZI01 (PPR[23:16])用来插入死区时间。互补输出一直延迟到计数器减到0.

死区时间可以按照下面的公式来计算:

Dead-Zone = BPWM0_CLK * (DZI01 (PPR[23:16]) +1).

下面的时序图说明一对BPWM信号的死区插入

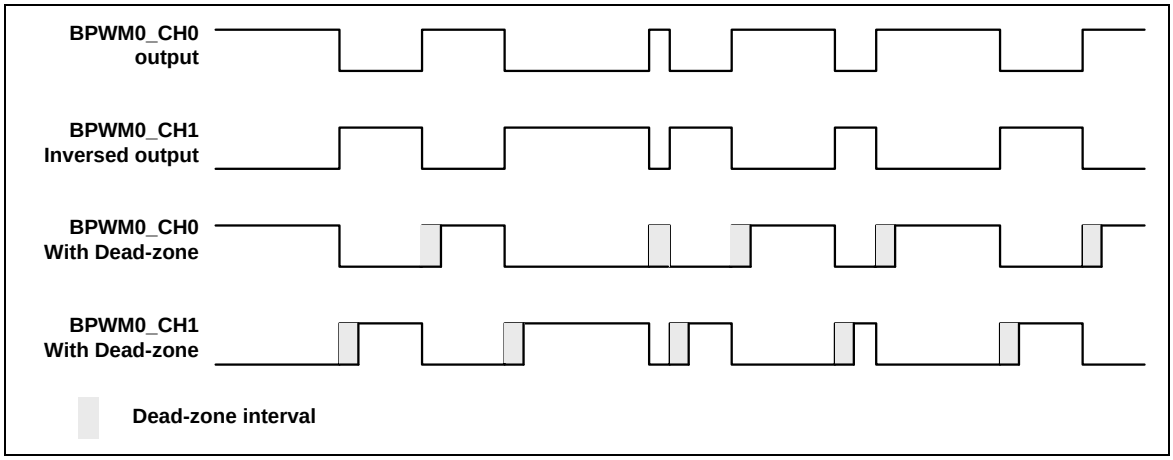



图 6-35 带有死区产生的BPWM对输出

在逆变器应用中，死区插入可以避免半整流桥的上下臂同时导通。因而死区控制对系统的正常运转至关重要。一个互补对中，在关闭一个BPWM输出而打开另一个时，必须等待一定的时间而不能同时切换.

#### *6.7.5.6* 极性控制

BPWM0_CH0 和BPWM0_CH1都有独立的极性控制来配置BPWM输出激活时的极性.预设情况下,BPWM输出高

下图显示了不同极性设定下BPWM的初始状态

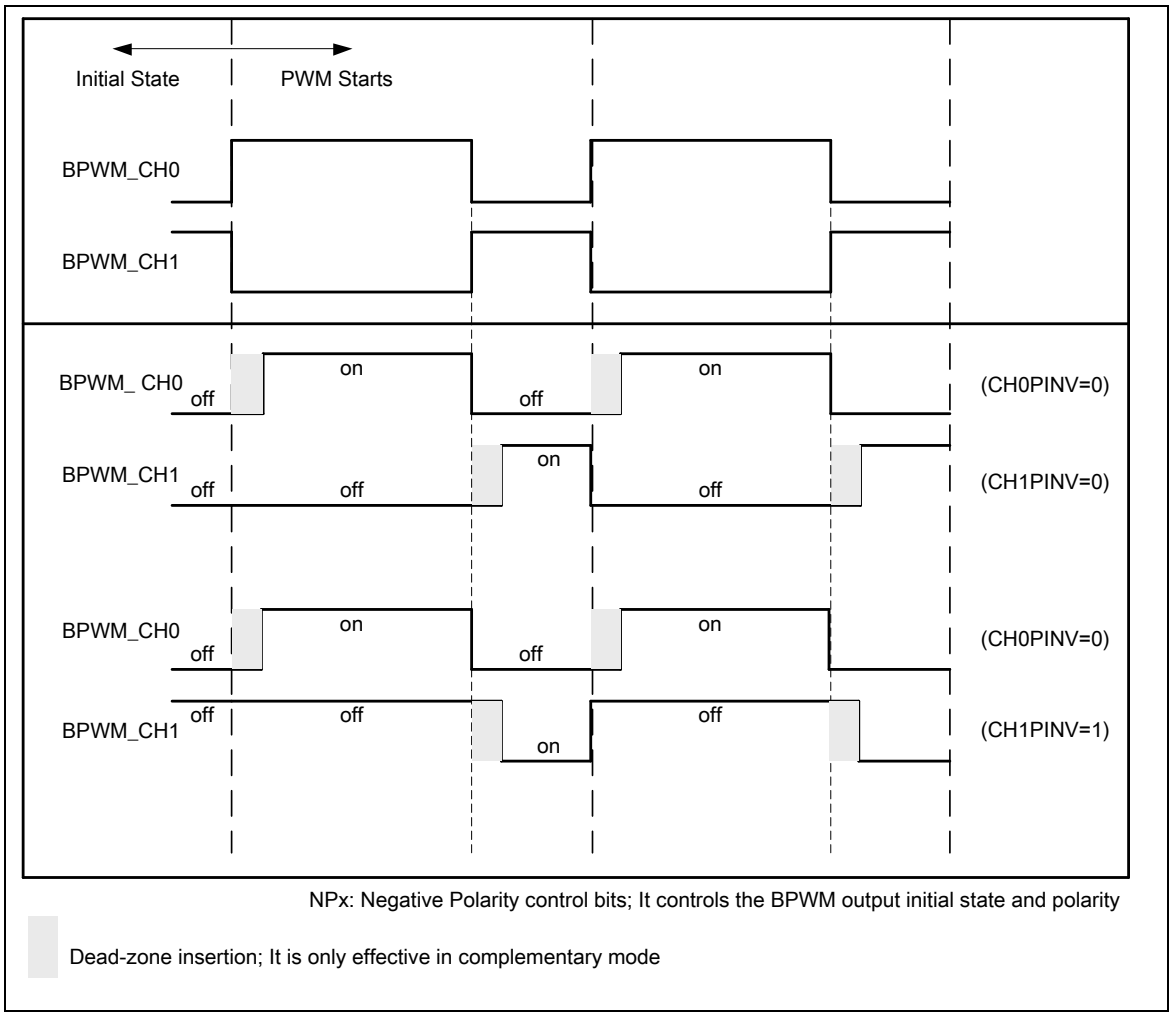



图 6-36 初始状态和极性控制并在上升沿插入死区

### *6.7.5.7* BPWM 触发 EADC 功能

当BPWM计数器在输出上升沿、下降沿、边沿对齐和中心对齐一个周期结束、中心对齐输出周期的中间都可以触发EADC转换。

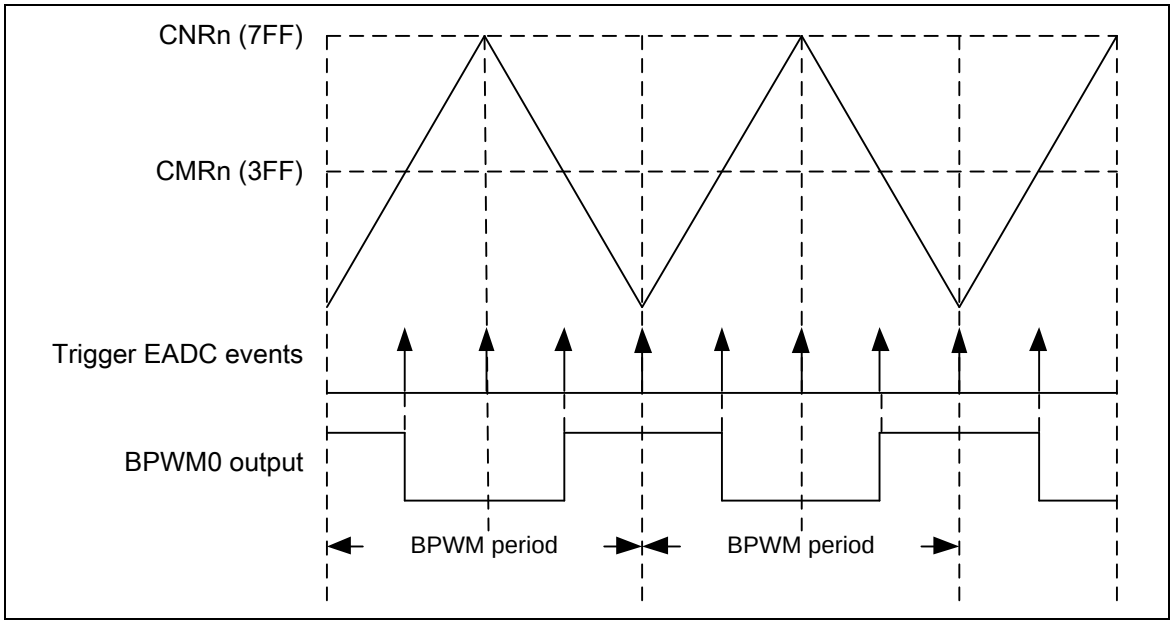



图 6-37 BPWM 触发 EADC 时序图

#### *6.7.5.8 捕获操作*

BPWM定时器另一个特征就是数字输入捕获功能。捕获输入通道0和输出通道0共享一个定时器和管脚(BPWM0_CH0)；捕获输入通道1和输出通道1共享一个定时器和管脚(BPWM0_CH1)。因此BPWM功能和捕获功能在同一个通道时是不能同时使用的。在使用捕获功能前用户必须先设置BPWM定时器。捕获功能开启后，当输入通道有一个上升沿转变时就会将当前的BPWM计数器的值锁存到捕获上升沿寄存器CRLRn中，同样当输入通道有一个下降沿转变时就会将当前的BPWM计数器的值锁存到捕获下降沿寄存器CFLRn中。捕获通道0中断可以同设置CRL_IE0 (CCR[1])（上升沿中断使能）和CFL_IE0 (CCR[2]) （下降沿中断使能）来决定中断产生的情况。捕获通道1中断配置位为CCR[17] 和 CCR[18]。每当捕获控制器发生捕获中断时，相应的BPWM计数器会重新加载CNRn的值。注对应捕获通道的GPIO管脚必须配置成捕获功能（禁用POE，使能CAPENR）。

捕获的最大频率受到捕获中断的延时限制。当捕获中断发生时，软件至少要执行三个步骤：读PIIR获取中断源、读CRLRn/CFLRn (n=0, 1)获取捕获的值、写1到PIIR清中断标志。如果中断延时花费T0时间完成，在这T0间隔时间内捕获信号不能发生转变，也就是说最大捕获频率不能超过1/T0。

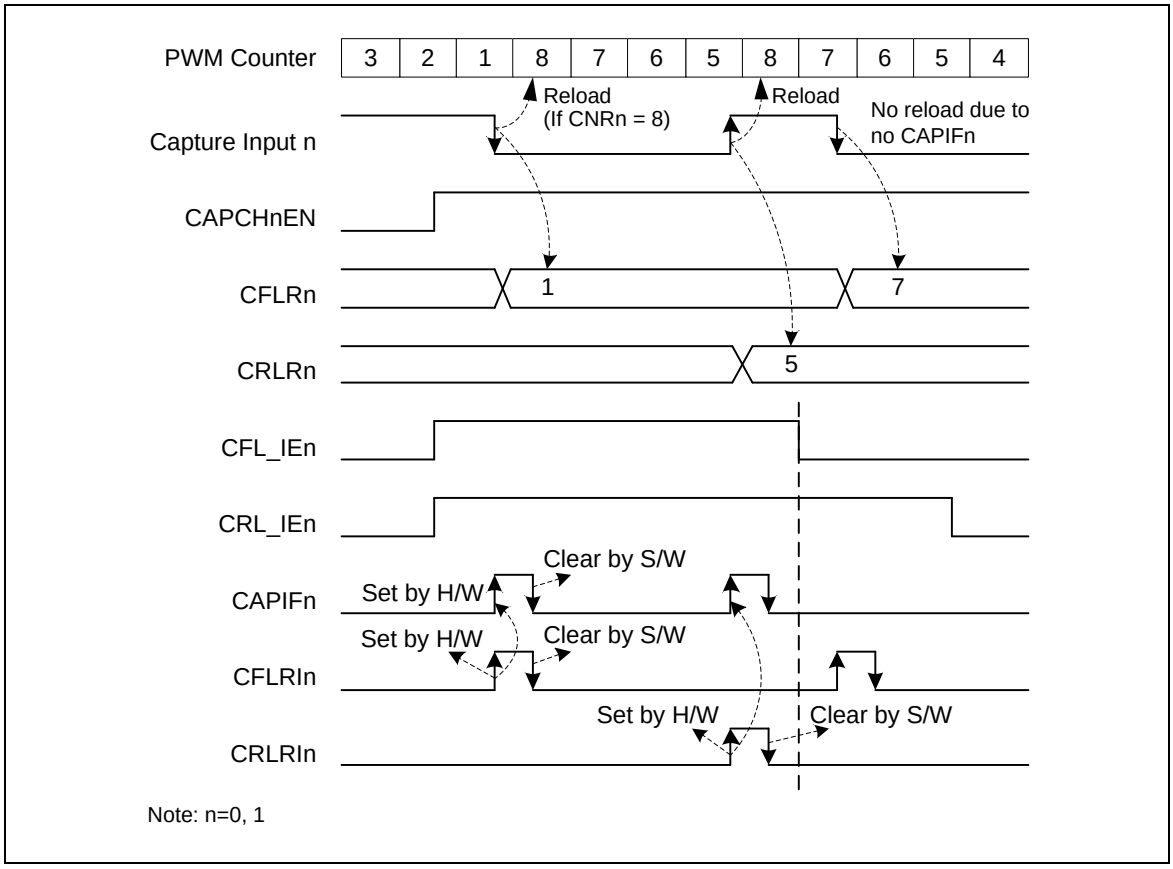



图 6-38 捕获操作时序

上图中 CNR 为 8:

1. 当捕获中断标志CAPIFn被置位时，BPWM计数器重新加载CNRn的值
2. 低电平宽度 (CNR + 1 - CRLR)
3. 高电平宽度 (CNR + 1 - CFLR)

### *6.7.5.9 BPWM定时器中断结构*

BPWM有2个中断，分别是BPCH0_INT和BPCH1_INT。BPWM输出功能0和捕获输入功能0共享一个中断；BPWM输出功能1和捕获输入功能1共享一个中断. BPWM功能和捕获功能在同一个通道时是不能同时使用的。图 6-39说明了BPWM定时器中断的结构。

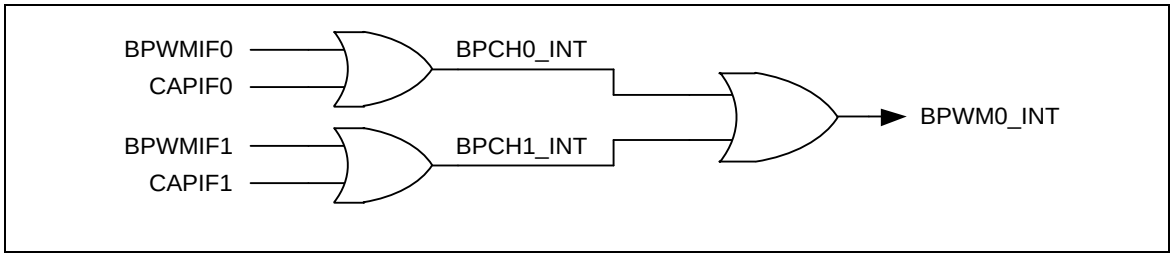



图 6-39 BPWM 中断结构框图

#### *6.7.5.10 BPWM定时器启动步骤*

建议按照下面步骤启动BPWM驱动

1. 设置时钟源除频选择寄存器(CSR)
2. 设置预分频寄存器 (PPR)
3. 设置BPWM控制寄存器(PCR)来配置反转 开/关、死区发生器开/关、自动加载/ 单次触发模式和停止BPWM定时器(PCR)
4. 设置比较器寄存器(CMR)调节占空比
5. 设置BPWM下数计数器寄存器(CNR)设置BPWM周期
6. 设置中断使能寄存器 (PIER) (可选)
7. 设置BPWM通道的GPIO管脚为BPWM功能（HZ_BPWM (PWMPOEN[4]) = 0，使能POE，禁用CAPENR）
8. 启动 BPWM 定时器 (向PCR中的 CHnEN位 写1)

#### *6.7.5.11 单次触发模式下BPWM定时器重启步骤*

单次触发模式下BPWM产生一次波形后，BPWM定时器会自动停止。再重新启动BPWM 单次触发输出波形请按照下面步骤

- 设置比较器寄存器(CMR)调节BPWM占空比
- 设置BPWM下数计数器寄存器(CNR)设置BPWM周期。设置CNR完成后BPWM会自动输出波形

#### *6.7.5.12 BPWM定时器 停止 步骤*

**方法 1:**

设置16位计数器(CNR) 为0，监视PDR当PDR下数到0时禁用BPWM定时器(PCR中的CHnEN位). ***(推荐)***

**方法 2:**

设置16位计数器(CNR) 为0，当中断请求发生时禁用BPWM定时器(PCR中的CHnEN位) ***(推荐)***

**方法 3:**

直接禁用BPWM定时器(PCR中的CHnEN位) ***(不建议)***

不推荐方法3的原因是通过直接禁用CHnEN来停止BPWM输出信号，这样会改变了BPWM的输出占空比，以至于会损坏到马达控制电路

*6.7.5.13 启动捕获功能步骤*

1. 设置时钟源除频选择寄存器(CSR)
2. 设置预分频寄存器 (PPR)
3. 打开通道、使能上升沿/下降沿中断、设置输入信号反转开/关(CCR)
4. 设置自动加载模式、边沿对齐类型和停止BPWM定时器（PCR）
5. 设置BPWM下数计数器 (CNR)
6. 启动 BPWM 定时器 (向PCR中的 CHnEN位 写1)
7. 设置BPWM通道的GPIO管脚为捕获功能（HZ_BPWM (PWMPOEN[4]) = 0，禁用POE，使能CAPENR）

### 6.7.6 寄存器映射

**R**: 只读, **W**: 只写, **R/W**: 读/写

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **BPWM0 基地址:**<br>**BPWM0_BA = 0x4004_0000** | | | | |
| **PPR** | BPWM0_BA+0x00 | R/W | BPWM0 预分频寄存器 | 0x0000_0000 |
| **CSR** | BPWM0_BA+0x04 | R/W | BPWM0时钟源除频选择寄存器 | 0x0000_0000 |
| **PCR** | BPWM0_BA+0x08 | R/W | BPWM0控制寄存器 | 0x0000_0000 |
| **CNR0** | BPWM0_BA+0x0C | R/W | BPWM0计数器寄存器0 | 0x0000_0000 |
| **CMR0** | BPWM0_BA+0x10 | R/W | BPWM0比较器寄存器0 | 0x0000_0000 |
| **PDR0** | BPWM0_BA+0x14 | R | BPWM0数据寄存器0 | 0x0000_0000 |
| **CNR1** | BPWM0_BA+0x18 | R/W | BPWM0计数器寄存器1 | 0x0000_0000 |
| **CMR1** | BPWM0_BA+0x1C | R/W | BPWM0比较器寄存器1 | 0x0000_0000 |
| **PDR1** | BPWM0_BA+0x20 | R | BPWM0数据寄存器1 | 0x0000_0000 |
| **PIER** | BPWM0_BA+0x40 | R/W | BPWM0中断使能寄存器 | 0x0000_0000 |
| **PIIR** | BPWM0_BA+0x44 | R/W | BPWM0中断指示寄存器 | 0x0000_0000 |
| **CCR** | BPWM0_BA+0x50 | R/W | BPWM0捕获控制寄存器 | 0x0000_0000 |
| **CRLR0** | BPWM0_BA+0x58 | R | BPWM0捕获上升沿锁存寄存器 (通道 0) | 0x0000_0000 |
| **CFLR0** | BPWM0_BA+0x5C | R | BPWM0捕获下降沿锁存寄存器(通道 0) | 0x0000_0000 |
| **CRLR1** | BPWM0_BA+0x60 | R | BPWM0捕获上升沿锁存寄存器(通道 1) | 0x0000_0000 |
| **CFLR1** | BPWM0_BA+0x64 | R | BPWM0捕获下降沿锁存寄存器(通道 1) | 0x0000_0000 |
| **CAPENR** | BPWM0_BA+0x78 | R/W | BPWM0捕获输入使能寄存器 | 0x0000_0000 |
| **POE** | BPWM0_BA+0x7C | R/W | BPWM0 输出使能 | 0x0000_0000 |

### 6.7.7 寄存器描述

**BPWM0预分频寄存器(PPR)**

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PPR** | BPWM0_BA+0x00 | R/W | BPWM0预分频寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **DZI01** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **CP01** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23:16] | **DZI01** | **通道0和通道1对的死区间隔**<br>这8位决定了死区的长度<br>死区单位时间长度 = [(prescale+1)*(clock source divider)] / BPWM0_CLK. |
| [15:8] | **Reserved** | 保留 |
| [7:0] | **CP01** | **时钟预分频**<br>时钟在送给BPWM计数器之前除以(CP01 + 1)<br>如果 CP01=0, 时钟预分频器 0时钟输出将停止. 所以相应的PWM计数器也将停止. |

## BPWM0时钟源除频选择寄存器(CSR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CSR** | BPWM0_BA+0x04 | R/W | BPWM0时钟源除频选择寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | **CSR1** | | | **Reserved** | **CSR0** | | |

| 位 | 描述 | |
|---|---|---|
| [31:7] | **Reserved** | 保留 |
| [6:4] | **CSR1** | **BPWM 定时器 1 时钟源除频选择**<br>选择BPWM定时器1的时钟源除频<br>000 =输入时钟被2除频<br>001 =输入时钟被4除频<br>010 =输入时钟被8除频<br>011 =输入时钟被16除频<br>100 =输入时钟被1除频 |
| [3] | **Reserved** | 保留 |
| [2:0] | **CSR0** | **BPWM 定时器 0 时钟源除频选择**<br>选择BPWM定时器0的时钟源除频，请参考CSR1 |

## BPWM0控制寄存器(PCR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PCR** | BPWM0_BA+0x08 | R/W | BPWM0控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | **PWM01TYPE** | **Reserved** | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | **CH1MOD** | **CH1INV** | **CH1PINV** | **CH1EN** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | **DZEN01** | **CH0MOD** | **CH0INV** | **CH0PINV** | **CH0EN** |

| 位 | 描述 | |
|---|---|---|
| [31] | **Reserved** | 保留 |
| [30] | **PWM01TYPE** | **BPWM0_CH0/1对齐类型选择**<br>0 = 边沿对齐类型<br>1 = 中心对齐类型 |
| [29:12] | **Reserved** | 保留 |
| [11] | **CH1MOD** | **BPWM定时器 1 自动加载/单次触发模式**<br>0 = 单次触发模式<br>1 = 自动加载模式<br>**注**: 如果该位有改变, 将导致CNR1 和 CMR1被清除. |
| [10] | **CH1INV** | **BPWM定时器 1 输出反转使能位**<br>0 =禁用反转<br>1 =使能反转 |
| [9] | **CH1PINV** | **BPWM定时器 1 输出极性 反转使能位**<br>0 = 禁用BPWM0_CH1输出极性反转<br>1 = 使能BPWM0_CH1输出极性反转 |
| [8] | **CH1EN** | **BPWM定时器 1 使能位**<br>0 = 停止 BPWM定时器计数<br>1 =使能 BPWM定时器开始计数 |
| [7:5] | **Reserved** | 保留 |
| [4] | **DZEN01** | **死区0发生器使能**<br>0 =禁用死区0发生器<br>1 =使能死区0发生器<br>**注**当死区发生器使能时, BPWM0_CH0 和BPWM0_CH1 变成互补对 |

| | | |
|---|---|---|
| [3] | **CH0MOD** | **BPWM定时器 0 自动加载/单次触发模式**<br>0 = 单次触发模式<br>1 = 自动加载模式<br>注: 如果该位有改变, 将导致CNR0 和 CMR0被清除. |
| [2] | **CH0INV** | **BPWM定时器 0 输出反转使能位**<br>0 =禁用反转<br>1 =使能反转 |
| [1] | **CH0PINV** | **BPWM定时器 0 输出极性 反转使能位**<br>0 = 禁用BPWM0_CH0输出极性反转<br>1 = 使能BPWM0_CH0输出极性反转 |
| [0] | **CH0EN** | **BPWM定时器 0使能位**<br>0 = 停止 BPWM定时器计数<br>1 =使能 BPWM定时器开始计数 |

## BPWM0计数器寄存器0-1 (CNR0-1)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CNR0** | BPWM0_BA+0x0C | R/W | BPWM0计数器寄存器0 | 0x0000_0000 |
| **CNR1** | BPWM0_BA+0x18 | R/W | BPWM0计数器寄存器1 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留 |
| [15:0] | **CNR** | **BPWM定时器加载值**<br>CNRn 决定BPWM的周期.<br>BPWM 频率 = BPWM0_CLK/[(prescale+1)*(clock divider)*(CNR+1)].<br>边沿对齐类型:<br>● 占空比 = (CMR+1)/(CNR+1).<br>● CMR >= CNR: BPWM 输出总为高<br>● CMR < CNR: BPWM 低电平宽度 = (CNR-CMR) unit; BPWM 高电平宽度 = (CMR+1) unit.<br>● CMR = 0: BPWM低电平宽度= (CNR) unit; BPWM高电平宽度= 1 unit.<br>中心对齐类型:<br>● 占空比 = [(2 x CMR) + 1]/[2 x (CNR+1)].<br>● CMR > CNR: BPWM输出总为高<br>● CMR <= CNR: BPWM低电平宽度= 2 x (CNR-CMR) + 1 unit; BPWM 高电平宽度= (2 x CMR) + 1 unit.<br>● CMR = 0: BPWM低电平宽度= 2 x CNR + 1 unit; BPWM 高电平宽度= 1 unit.<br>(Unit =一个BPWM时钟周期).<br>**注:**写CNR将在BPWM下一个周期生效<br>**注:** 当BPWM工作在中心对齐类型时，CNR的设置范围在0x0000 到0xFFFE。如果CNR等于0xFFFF,BPWM工作不可预测<br>**注:**当CNR为0时，BPWM输出总为高 |

## BPWM0比较器寄存器0-1 (CMR0-1)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CMR0** | BPWM0_BA+0x10 | R/W | BPWM0比较器寄存器0 | 0x0000_0000 |
| **CMR1** | BPWM0_BA+0x1C | R/W | BPWM0比较器寄存器1 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留 |
| [15:0] | **CMR** | **BPWM比较器寄存器**<br>CMR决定BPWM的占空比<br>BPWM 频率 = BPWM0_CLK/[(prescale+1)*(clock divider)*(CNR+1)].<br>边沿对齐类型:<br>● 占空比 = (CMR+1)/(CNR+1).<br>● CMR >= CNR: BPWM 输出总为高<br>● CMR < CNR: BPWM 低电平宽度 = (CNR-CMR) unit; BPWM 高电平宽度 = (CMR+1) unit.<br>● CMR = 0: BPWM低电平宽度= (CNR) unit; BPWM高电平宽度= 1 unit.<br>中心对齐类型:<br>● 占空比 = [(2 x CMR) + 1]/[2 x (CNR+1)].<br>● CMR > CNR: BPWM输出总为高<br>● CMR <= CNR: BPWM低电平宽度= 2 x (CNR-CMR) + 1 unit; BPWM 高电平宽度= (2 x CMR) + 1 unit.<br>● CMR = 0: BPWM低电平宽度= 2 x CNR + 1 unit; BPWM 高电平宽度= 1 unit.<br>(Unit =一个BPWM时钟周期).<br>**注:**写CMR将在BPWM下一个周期生效 |

## BPWM0 数据寄存器 0-1 (PDR 0-1)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PDR0** | BPWM0_BA+0x14 | R | BPWM0数据寄存器0 | 0x0000_0000 |
| **PDR1** | BPWM0_BA+0x20 | R | BPWM0数据寄存器1 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留 |
| [15:0] | **PDR** | **BPWM 数据寄存器**<br>用户可以监控PDR了解当前16位计数器的值 |

## BPWM0中断使能寄存器(PIER)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PIER** | BPWM0_BA+0x40 | R/W | BPWM0中断使能寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | **INTTYPE** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | **BPWMDIE1** | **BPWMDIE0** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | **BPWMPIE1** | **BPWMPIE0** |

| 位 | 描述 | |
|---|---|---|
| [31:17] | **Reserved** | 保留 |
| [16] | **INTTYPE** | **BPWM中断类型选择**<br>0 = 当BPWM计数器下溢时BPWMPIFn 将被置位.<br>1 = 当BPWM计数器与CNRn寄存器的值匹配时BPWMPIFn 将被置位.<br>注: 当BPWM工作在中心对齐模式时该比特才起作用. |
| [15:10] | **Reserved** | 保留 |
| [9] | **BPWMDIE1** | **BPWM 通道 1 占空比中断使能位**<br>0 =禁用BPWM0_CH1 占空比中断<br>1 =使能BPWM0_CH1 占空比中断 |
| [8] | **BPWMDIE0** | **BPWM 通道 0占空比中断使能位**<br>0 =禁用BPWM0_CH0 占空比中断<br>1 =使能BPWM0_CH0占空比中断 |
| [7:2] | **Reserved** | 保留 |
| [1] | **BPWMPIE1** | **BPWM 通道 1 周期中断使能位**<br>0 =禁用BPWM0_CH1 周期中断<br>1 =使能BPWM0_CH1 周期中断 |
| [0] | **BPWMPIE0** | **BPWM 通道 1 周期中断使能位**<br>0 =禁用BPWM0_CH0 周期中断<br>1 =使能BPWM0_CH0 周期中断 |

## BPWM0中断指示寄存器(PIIR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PIIR** | BPWM0_BA+0x44 | R/W | BPWM0中断指示寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | **BPWMDIF1** | **BPWMDIF0** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | **BPWMIF1** | **BPWMIF0** |

| 位 | 描述 | |
|---|---|---|
| [31:10] | **Reserved** | 保留 |
| [9] | **BPWMDIF1** | **BPWM 通道 1 占空比中断标志**<br>当BPWM0_CH1计数器下数并达到CMR1的值时该标志位将被置位,软件可以写 “1”清除<br>**注:** 如果CMR等于CNR时，在边沿对齐类型下该标志位不会工作 |
| [8] | **BPWMDIF0** | **BPWM 通道 0 占空比中断标志**<br>当BPWM0_CH0计数器下数并达到CMR0的值时该标志位将被置位,软件可以写 “1”清除<br>**注:** 如果CMR等于CNR时，在边沿对齐类型下该标志位不会工作 |
| [7:2] | **Reserved** | 保留 |
| [1] | **BPWMIF1** | **BPWM通道1周期中断标志**<br>当BPWM0_CH1计数器到达需求中断条件时（取决于INTTYPE和PIER），该位被硬件置位<br>**注:**该位写1可被清除 |
| [0] | **BPWMIF0** | **BPWM通道0周期中断标志**<br>当BPWM0_CH0计数器到达需求中断条件时（取决于INTTYPE和PIER），该位被硬件置位<br>**注:**该位写1可被清除 |

## BWM0捕获控制寄存器(CCR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| CCR | BPWM0_BA+0x50 | R/W | BPWM0捕获控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CFLRI1 | CRLRI1 | Reserved | CAPIF1 | CAPCH1EN | CFL_IE1 | CRL_IE1 | INV1 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CFLRI0 | CRLRI0 | Reserved | CAPIF0 | CAPCH0EN | CFL_IE0 | CRL_IE0 | INV0 |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留 |
| [23] | **CFLRI1** | **CFLR1锁存指示位**<br>BPWM0组输入通道1有下降沿转变时，BPWM0向下计数器的值锁存到CFLR1寄存器，同时该位由硬件置位。<br>**注：**该位写1清0. |
| [22] | **CRLRI1** | **CRLR1锁存指示位**<br>BPWM0组输入通道1有上升沿转变时，BPWM0向下计数器的值锁存到CRLR1寄存器，同时该位由硬件置位。<br>**注：**该位写1清0. |
| [5] | **Reserved** | 保留 |
| [20] | **CAPIF1** | **捕捉通道1中断标志**<br>如果BPWM0_CH1上升沿锁存中断使能(CRL_IE1=1), BPWM0_CH1有上升沿转变将使CAPIF1为高； 同样, 如果下降沿锁存中断使能(CFL_IE1=1)，下降沿转变将使CAPIF1 为高.<br>注：该位写1清0. |
| [19] | **CAPCH1EN** | **捕捉器通道1使能控制**<br>0 = 禁用BPWM0_CH1的捕捉功能<br>1 = 使能BPWM0_CH1的捕捉功能.<br>使能时，捕捉锁存BPWM0_CH1计数器并保存到CRLR（上升沿锁定）和CFLR（上升沿锁定）<br>禁用时，捕捉器不更新CRLR和CFLR，并禁用BPWM0_CH1捕获中断. |
| [18] | **CFL_IE1** | **通道1下降沿锁存中断使能控制**<br>0 = 禁用下降沿锁存中断<br>1 = 使能下降沿锁存中断<br>注：使能时，如果捕捉器检测到BPWM0_CH1有下降沿转变，捕捉器产生中断. |
| [17] | **CRL_IE1** | **通道1上升沿锁存中断使能控制** |

| | | |
|---|---|---|
| | | 0 = 禁用上升沿锁存中断<br>1 = 使能上升沿锁存中断<br>注：使能时，如果捕捉器检测到BPWM0_CH1有上升沿转变，捕捉器产生中断 |
| [16] | **INV1** | **通道1反向使能控制**<br>0 = 反向关闭<br>1 = 反向打开。I/O脚上的信号输入到捕获器前反转再输入。 |
| [15:8] | **Reserved** | 保留 |
| [7] | **CFLRI0** | **CFLR0锁存指示位**<br>BPWM0组输入通道0有下降沿转变时，BPWM0向下计数器的值锁存到CFLR0寄存器，同时该位由硬件置位。<br>**注：**该位写1清0. |
| [6] | **CRLRI0** | **CRLR0锁存指示位**<br>BPWM0组输入通道0有上升沿转变时，BPWM0向下计数器的值锁存到CRLR0寄存器，同时该位由硬件置位。<br>**注：**该位写1清0. |
| [5] | **Reserved** | 保留 |
| [4] | **CAPIF0** | **捕捉通道0中断标志**<br>如果BPWM0_CH0上升沿锁存中断使能(CRL_IE0=1), BPWM0_CH0有上升沿转变将使CAPIF0为高； 同样, 如果下降沿锁存中断使能(CFL_IE0=1)，下降沿转变将使CAPIF0 为高.<br>注：该位写1清0. |
| [3] | **CAPCH0EN** | **捕捉器通道0使能控制**<br>0 = 禁用BPWM0_CH0的捕捉功能<br>1 = 使能BPWM0_CH0的捕捉功能.<br>使能时，捕捉锁存BPWM0_CH0计数器并保存到CRLR（上升沿锁定）和CFLR（上升沿锁定）<br>禁用时，捕捉器不更新CRLR和CFLR，并禁用BPWM0_CH0捕获中断. |
| [2] | **CFL_IE0** | **通道0下降沿锁存中断使能控制**<br>0 = 禁用下降沿锁存中断<br>1 = 使能下降沿锁存中断<br>注：使能时，如果捕捉器检测到BPWM0_CH0有下降沿转变，捕捉器产生中断. |
| [1] | **CRL_IE0** | **通道0上升沿锁存中断使能控制**<br>0 = 禁用上升沿锁存中断<br>1 = 使能上升沿锁存中断<br>注：使能时，如果捕捉器检测到BPWM0_CH0有上升沿转变，捕捉器产生中断 |
| [0] | **INV0** | **通道0反向使能控制**<br>0 = 反向关闭<br>1 = 反向打开。I/O脚上的信号输入到捕获器前反转再输入。 |

## BPWM0捕获上升沿锁存寄存器0-1 (CRLR0-1)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CRLR0** | BPWM0_BA+0x58 | R | BPWM0捕获上升沿锁存寄存器(通道 0) | 0x0000_0000 |
| **CRLR1** | BPWM0_BA+0x60 | R | BPWM0捕获上升沿锁存寄存器(通道 1) | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留 |
| [15:0] | **CRLR** | **捕获上升沿锁存寄存器**<br>当通道0/1有上升沿转变时，锁存BPWM0计数器的值到这个寄存器 |

## BPWM0捕获下降沿锁存寄存器0-1 (CFLR0-1)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CFLR0** | BPWM0_BA+0x5C | R | BPWM0捕获下降沿锁存寄存器(通道 0) | 0x0000_0000 |
| **CFLR1** | BPWM0_BA+0x64 | R | BPWM0捕获下降沿锁存寄存器(通道 1) | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留 |
| [15:0] | **CFLR** | **捕获下降沿锁存寄存器**<br>当通道0/1有下降沿转变时，锁存BPWM0计数器的值到这个寄存器. |

## BPWM0捕获输入使能寄存器(CAPENR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **CAPENR** | BPWM0_BA+0x78 | R/W | BPWM0捕获输入使能寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | **CINEN1** | **CINEN0** |

| 位 | 描述 | |
|---|---|---|
| [31:2] | **Reserved** | 保留 |
| [1] | **CINEN1** | **通道1捕获输入使能位**<br>0 =禁用BPWM0_CH1捕获输入通道。BPWM0_CH1输入捕获功能总被当作0.<br>1 = 使能BPWM0_CH1捕获输入通道。如果GPIO多功能设置为BPWM0_CH1，那么BPWM0_CH1输入捕获功能来自对应的多功能管脚。 |
| [0] | **CINEN0** | **通道0捕获输入使能位**<br>0 =禁用BPWM0_CH0捕获输入通道。BPWM0_CH0输入捕获功能总被当作0.<br>1 = 使能BPWM0_CH0捕获输入通道。如果GPIO多功能设置为BPWM0_CH0，那么BPWM0_CH0输入捕获功能来自对应的多功能管脚。 |

## BPM0 输出使能寄存器 (POE)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **POE** | BPWM0_BA+0x7C | R/W | BPWM0输出使能 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | **POE1** | **POE0** |

| 位 | 描述 | |
|---|---|---|
| [31:2] | **Reserved** | 保留 |
| [1] | **POE1** | **通道 1 输出使能位**<br>0 =禁用 BPWM0_CH1输出到引脚上<br>1 =使能 BPWM0_CH1输出到引脚上<br>**注:** 相应的GPIO脚也必须切成BPWM0功能 |
| [0] | **POE0** | **通道 0 输出使能位**<br>0 =禁用 BPWM0_CH0输出到引脚上<br>1 =使能 BPWM0_CH0输出到引脚上<br>**注:** 相应的GPIO脚也必须切成BPWM0功能 |

## 6.8 增强型PWM发生器 (EPWM)

### 6.8.1 概述

M0519内置了两组针对马达驱动应用设计的PWM单元。

### 6.8.2 特性

每个PWM单元都支持如下的特性：

- 3个独立的16位PWM占空比控制单元控制最多6个PWM输出管脚：
  - 3路独立的PWM输出：
    PWM单元0：EPWM0_CH0, EPWM0_CH2 和 EPWM0_CH4
    PWM单元1：EPWM1_CH0, EPWM1_CH2 和 EPWM1_CH4
  - 3组互补PWM对，每组PWM对的一根引脚与另一根引脚互补，且可以插入可编程的死区时间：
    (EPWMx_CH0, EPWMx_CH1), (PWMx_CH2, EPWMx_CH3) 和(EPWMx_CH4, EPWMx_CH5)，其中 x=0~1
  - 3组同步PWM对，每组PWM对的引脚相位同步：
    (EPWMx_CH0, EPWMx_CH1), (EPWMx_CH2, EPWMx_CH3) 和 (EPWMx_CH4, EPWMx_CH5)，其中 x=0~1
- 支持组控制：
  EPWMx_CH2、 EPWMx_CH4 与 EPWMx_CH0 同步
- 支持边沿对齐模式和中心对齐模式
- 支持在互补PWM对间插入可编程死区时间
- PWM0_CH0到PWM0_CH5每路PWM管脚都有独立的极性控制
- 在马达电流换相操作时支持输出屏蔽功能
- 在复位和刹车状态时三态输出
- 硬件刹车保护
- 两个中断源：
  - 向下计数器比较匹配（边沿对齐和中心对齐模式下）或者下溢（边沿对齐模式）的时候，发生中断请求
  - 外部刹车引脚信号发生中断请求
- PWM预设状态下为正逻辑PWM，PWM口的有效状态由极性控制寄存器来控制。
- 高灌/拉电流
- 支持触发EADC

### 6.8.3 方块图

CPU复位后，每个PWM通道的默认输出取决于极性的设置。连续输出的间隔由16位的向上/下计数器控制，该16位计数器使用软件可选的时钟源，由内置的时钟分频器分频后作为其输入时钟。PWM计数时钟的时钟频率和时钟源为$F_{PWM}$ = EPWMx_CLK/预分频值; 这里 EPWMx_CLK 与CPU时钟 HCLK同步。

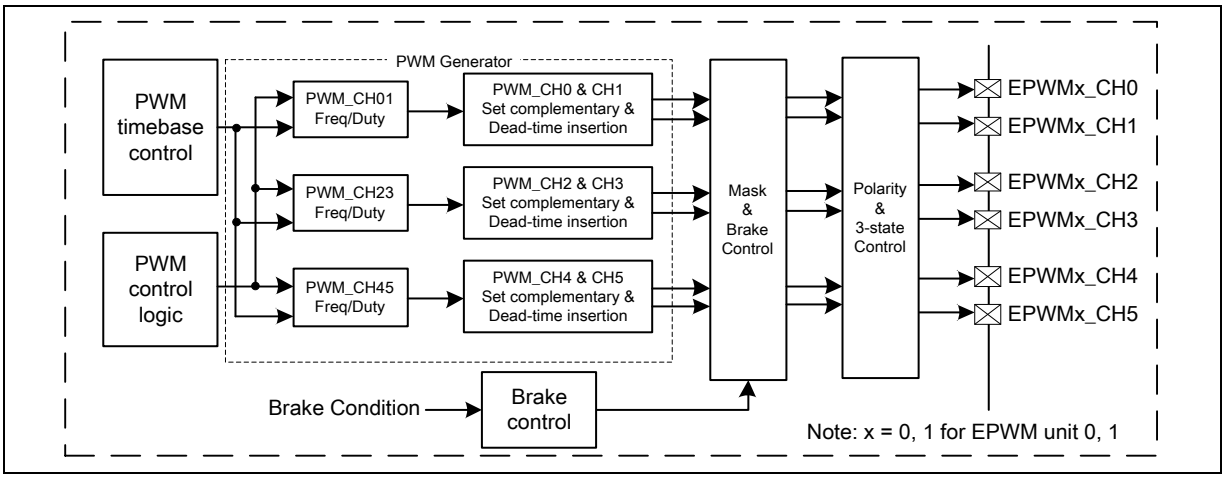



图 6-40 EPWM 方块图

### 6.8.4 基本配置

EPWM 管脚功能选择由P0_MFP和P1_MFP两个寄存器配置。

EPWM的外设时钟源可以通过EPWM0_EN (APBCLK[20]) 和 EPWM1_EN (APBCLK[21])进行使能。

### 6.8.5 功能描述

图 6-41是 EPWM 时钟源控制框图和 PWM 时基发生器框图。

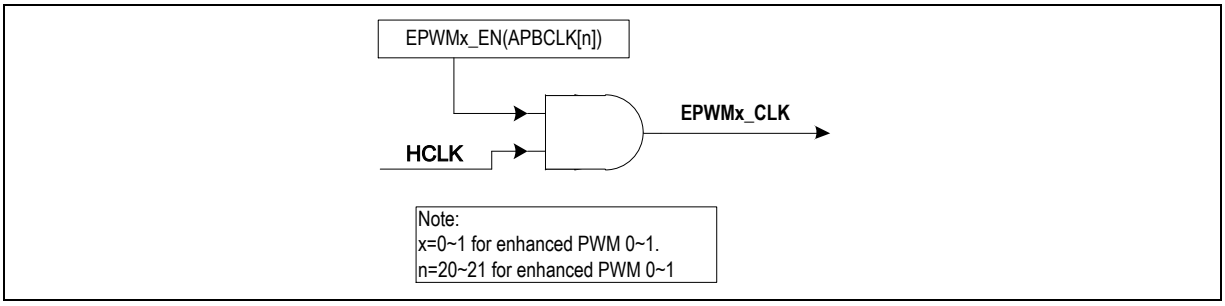



图 6-41 EPWM 时钟源控制

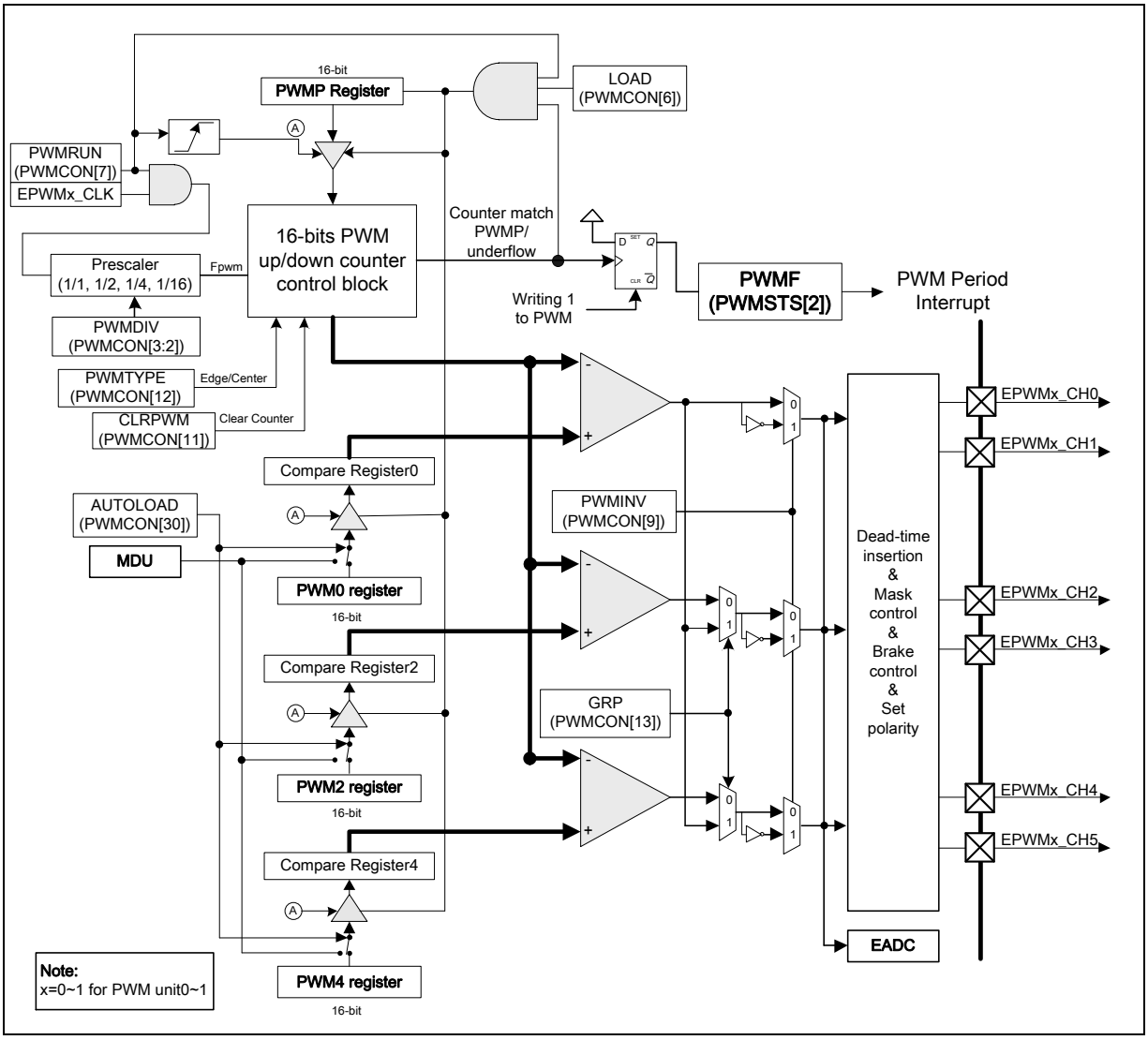



图 6-42 PWM 时基发生器

PWM模块所有的功能都是由寄存器PWMCONx0, PWMB 和 PWMCON里的内容来控制。大部分控制位的操作都很直接。PWMRUN (PWMCON[7]) 控制PWM处于工作状态或者空闲状态。数据从PWMP寄存器到16位PWM计数器的传送发生在：PWM的上升沿或LOAD(PWMCON[6])位使能且PWM计数器到达计数值或下溢时。数据从PWMn寄存器到比较寄存器则由LOAD(PWMCON[6])位控制(发生条件为，PWMRUN=1且PWM计数器到达计数值或溢出时)。

**注意:**

1. 比较值高于计数器重载值会导致PWM一直处在输出高的状态。另外还有两个特殊情况。当比较寄存器设为0时，PWM_CHn将持续输出低；而比较寄存器设为0xFFFF时，PWM_CHn将持续输出高直到比较寄存器里的值被修改。[n = 0-5]。
2. 在 ICP, ISP 和 ICE 模式下, PWM 管脚为三态模式.。PWM操作会停止同时模块会重启。离开ICP, ISP和ICE模式后，PWM管脚将会按设置寄存器的设置工作。
3. 在ICE模式下，一旦CPU停止或者暂停运行，PWM_CHn 输出管脚会强制切换成三态模式。CPU运行时，PWM_CHn管脚将会按设置寄存器的设置工作。

软件写PWM周期值到PWMP寄存器后，该值会先被保存在保持寄存器里，然后保持寄存器里的值

会在满足LOAD (PWMCON[6]) = 1, PWMRUN (PWMCON[7]) = 1 且 PWM 计数值到达/下溢条件时重载为当前PWM周期。每个PWM输出脉冲宽度由对应的比较寄存器里的值决定。每个16位宽的PWM寄存器PWM0、PWM2、PWM4决定了PWM的周期和各个通道的占空比。如果PWMINV (PWMCON[9])( PWM比较输出反向)寄存器设为高，PWM比较输出信号将会翻转，此时 PWM占空比寄存器表示PWM0/2/4关断时间，PWM实际占空比将会是1- Duty。

注意占空比寄存器PWM0/2/4和周期寄存器PWMP都是双缓冲的寄存器，以使占空比和周期调节各自独立。其中，用户写数据到缓冲区1而缓冲区2则保存当前周期使用的比较值。当PWM 计数值到达/下溢时，LOAD位一定要被置1以保证缓冲区1的值能载入缓冲区2。

#### *6.8.5.1 PWM工作模式*

支持两个模式：边沿对齐模式和中心对齐模式。

以下是各模式下占空比和周期的计算公式：

<u>边沿对齐</u>:

周期 = (PWMP +1) * EPWMx_CLK period/pre-scalar

占空比 = (duty + 1) * EPWMx_CLK period /pre-scalar

<u>中心对齐</u>:

周期 = (PWMP* 2) * EPWMx_CLK period/pre-scalar

占空比 = (duty*2 + 1) * EPWMx_CLK period/pre-scalar

**主要:** “占空比” 参照 PWM0/2/4/6 寄存器值.

<u>**边沿对齐的PWM (向上计数)**</u>

边沿对齐的PWM输出模式，16位的PWM计数器从0开始计数到PWMP寄存器里的周期值，完成一个PWM周期，然后再从0开始重新计数到周期值。PWM计数器的值与占空比值(PWMn寄存器)比较来决定PWM发生器的输出状态。当PWM计数器高于占空比值时，PWM发生器输出低；当PWM计数器小于等于占空比值时，PWM发生器输出高。

PWMI_En (PWMCON[4])置1时会触发PWM周期中断(PWMF (PWMSTS[2]))， PWMnEI_EN (PWMEIC[2:0]) 置1则会触发PWM边沿中断(PWMnEF (PWMSTS[6:4]))。

PWM周期和占空比通过PWMP和PWMn寄存器配置。如果LOAD (PWMCON[6])值为1，新设置的周期和占空比将会在下一个周期开始时生效。

图 6-43, 图 6-44和 图 6-45 是边沿对齐模式下PWM的时序图和工作流程。

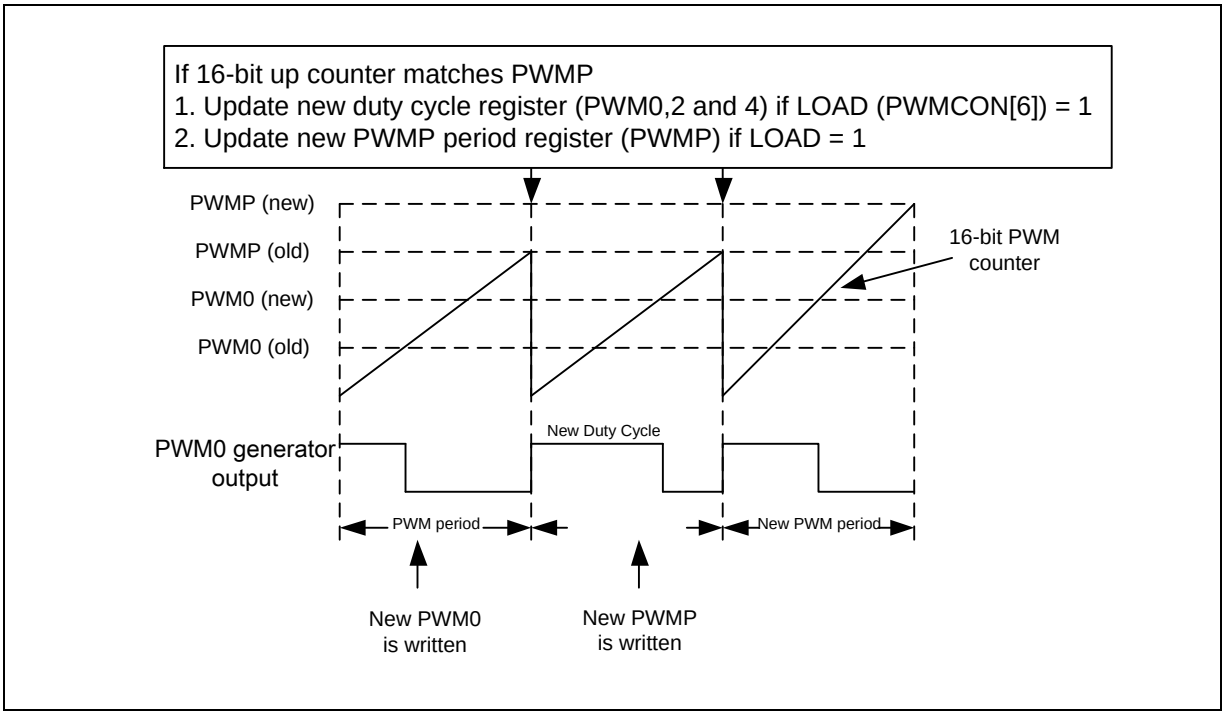



图 6-43 边沿对齐模式PWM

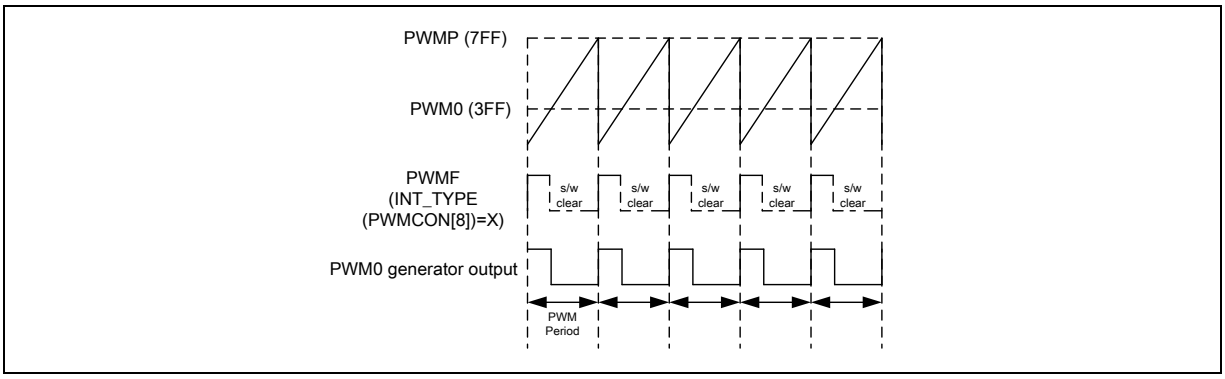



图 6-44 PWM0 边沿对齐下的输出波形

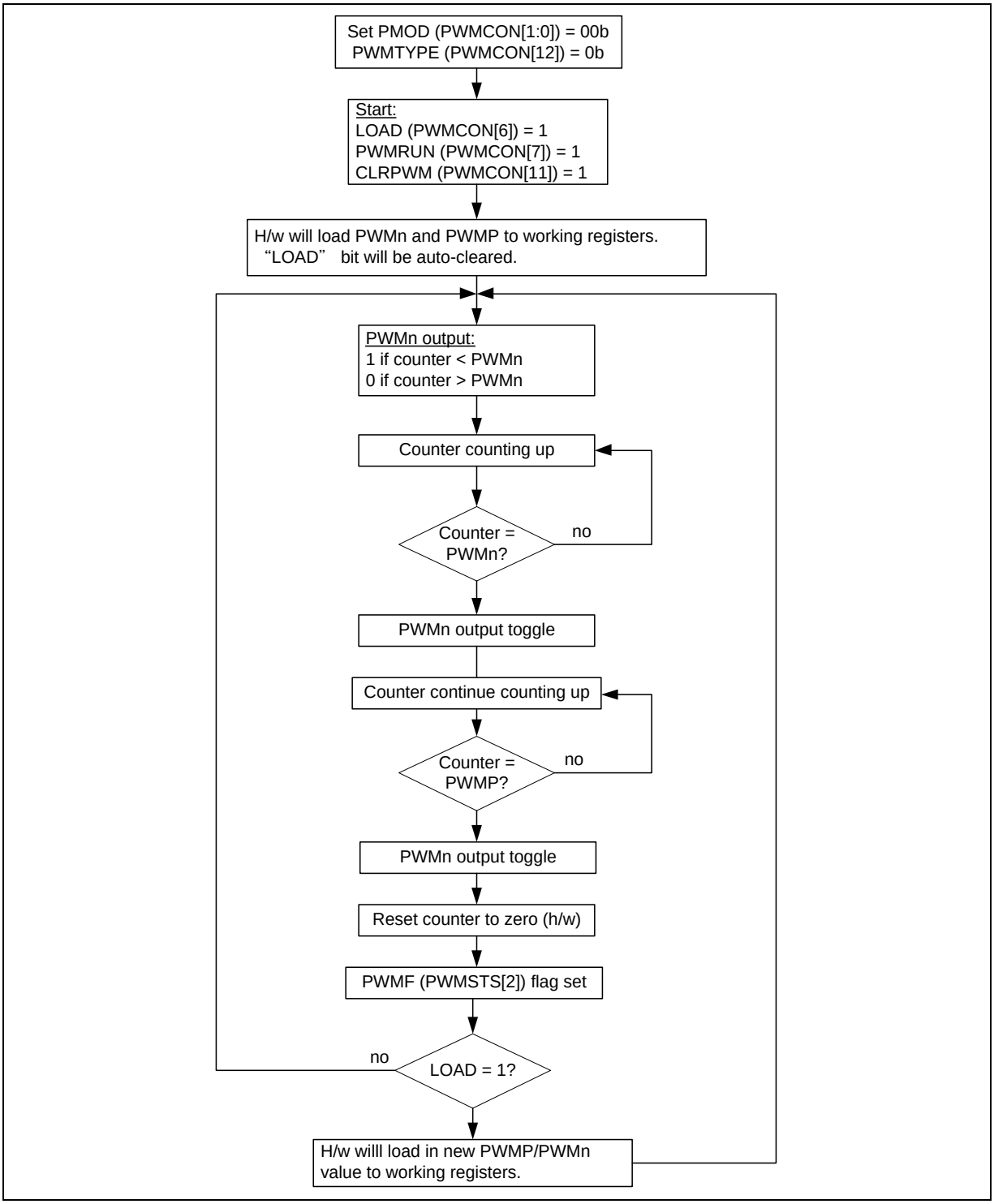



图 6-45 边沿对齐流程图

**中心对齐模式PWM (向上/向下计数)**

PWM时钟基准设定为向上/向下计数模式时，模块会产生中心对齐的PWM信号。PWM计数器从0开始向上计数到PWMP寄存器的值，然后向下计数到0，完成一个PWM周期，再重新开始下一个PWM周期计数。PWM计数器的值与占空比值(PWMn寄存器)比较来决定PWM发生器的输出状态。当PWM计数器高于占空比值时，PWM发生器输出低；当PWM计数器小于等于占空比值时，PWM发生器输出高。如果LOAD (PWMCON[6])值为1，一旦PWM计数值下溢新设置的周期和占空比将会立即生效。

中心对齐模式下，INT_TYPE (PWMCON0[8]) =0，PWM中断请求发生在向下计数下溢时，例如，在每个PWM周期的开始(结尾)；或者当INT_TYPE (PWMCON0[0]) =1时，PWM中断请求发生在向上计数到PWMP的值。

图 6-46, 图 6-47 和 图 6-48是中心对齐模式下PWM的时序图和工作流程。

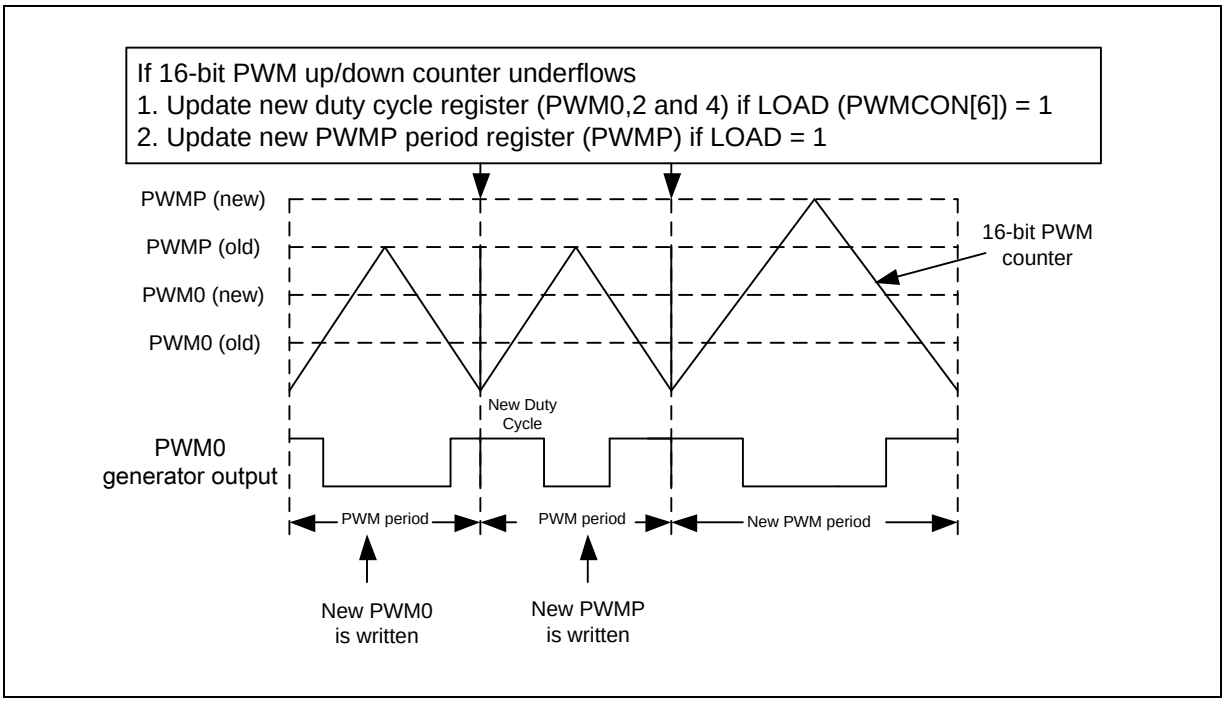



图 6-46 中心对齐模式

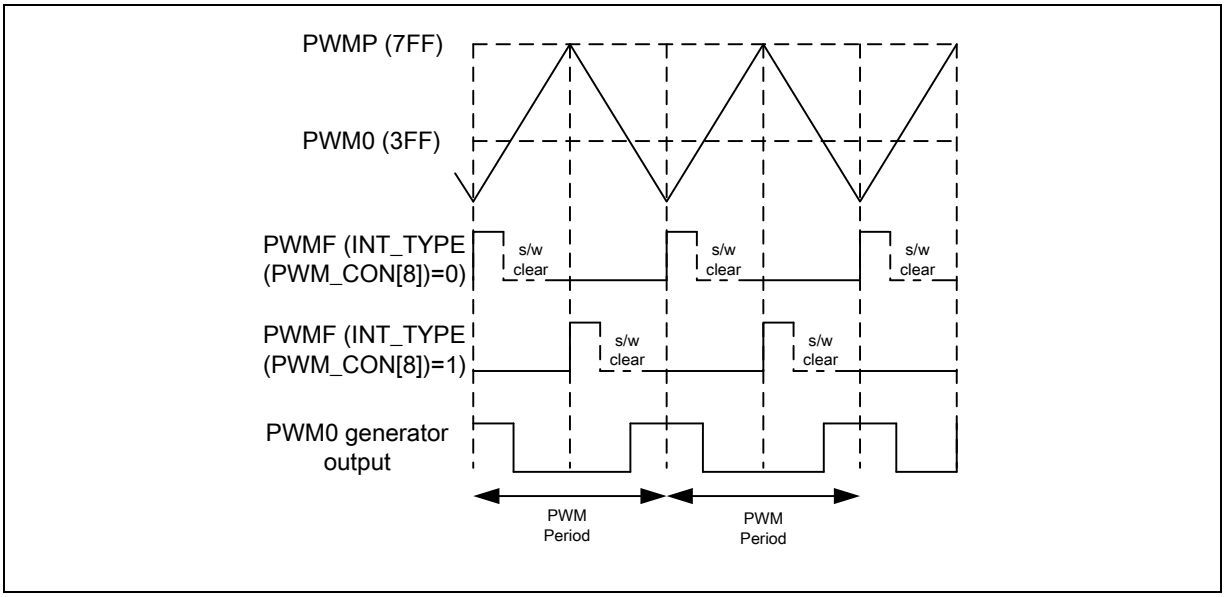



图 6-47 PWM0 中心对齐下的输出波形

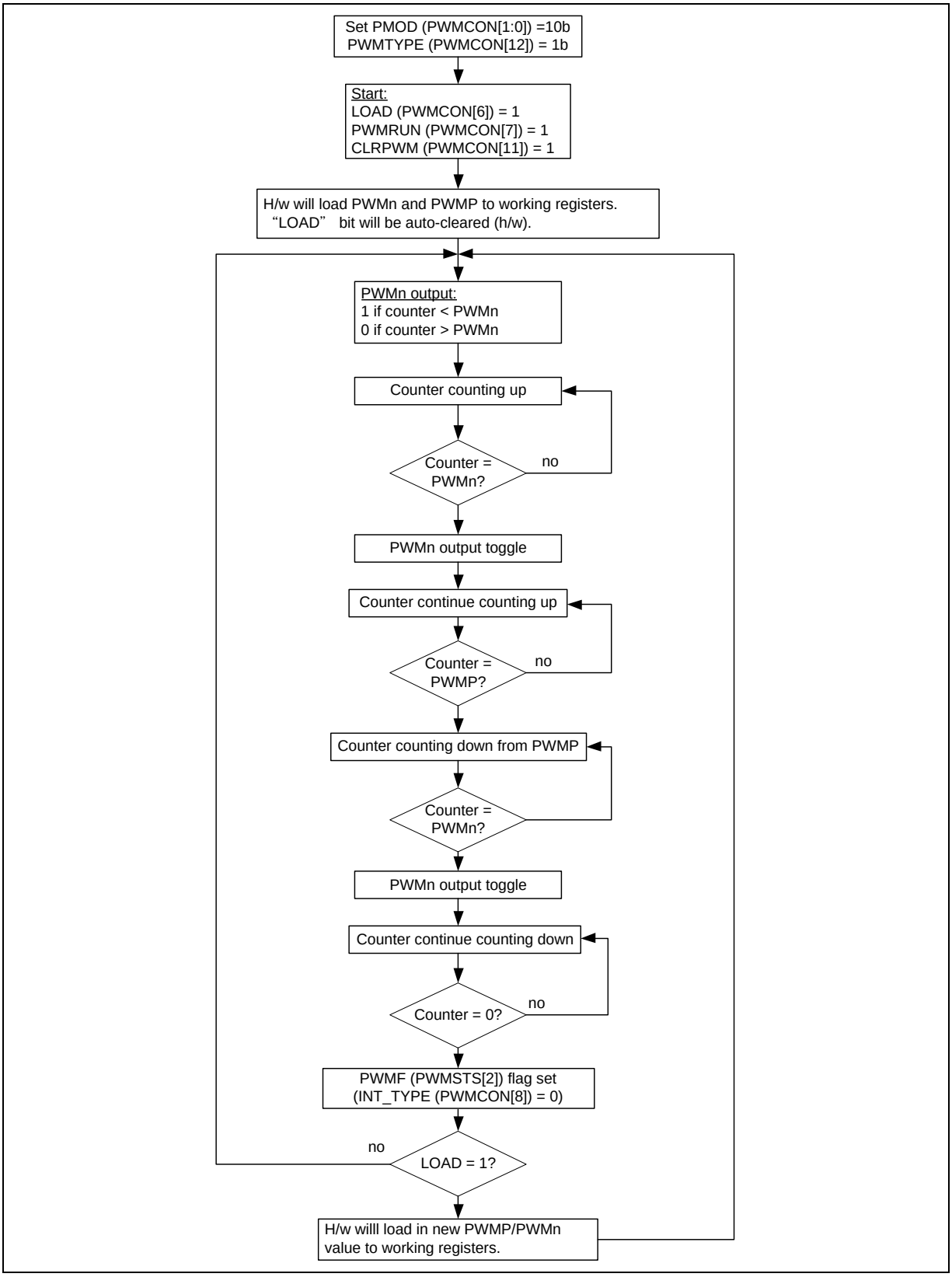



图 6-48 中心对齐流程图 (INT_TYPE = 0)

### 6.8.5.2 PWM 刹车

M0519支持两组刹车检测模块，BK0和BK1。每组刹车模块有4个刹车信号，1个外部刹车管脚(EPWMx_BRAKE0 对应 BK0 和 EPWMx_BRAKE1 对应 BK1)和3个模拟比较强输出。外部刹车管脚内置4档数字滤波，用户可以通过BKxFILT (PWMCON[23:20]) (x=0 、1分别是 BK0 和 BK1)来调节。刹车功能由PWMCON寄存器控制。

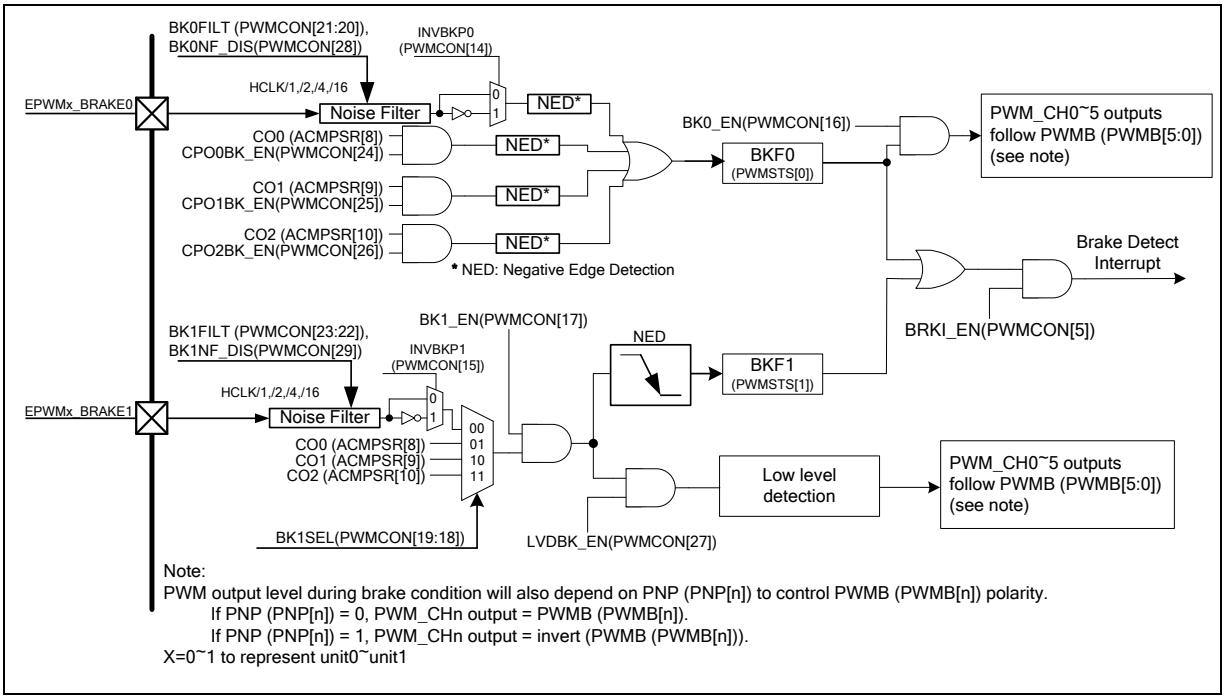



图 6-49 PWM刹车功能

表 6-9 总结了每个刹车管脚的效果. 图 6-50 和 图 6-51 则描绘了刹车信号和PWM的工作情况。

| 刹车源 | 刹车触发调节 | 动作 |
|---|---|---|
| BKP0<br>(I/O 口);<br>CO0 (ACMPSR[8])<br>(比较强0输出);<br>CO1 (ACMPSR[9])<br>(比较强1输出);<br>CO2 (ACMPSR[10])<br>(比较强2输出) | 刹车0边沿检测检测到下降沿 | ● 刹车0边沿检测检测到下降沿，BKF0 (PWMSTS[0]) 和 BKLK0 (PWMSTS[8]) 标志位置1；可通过软件清除。<br>● PWM_CH0~5 输出跟随 PWMB (PWMB[n]). PNP (PNP[n]) 也可以控制PWMB的极性.<br>若 PNP (PNP[n])=0, PWMn 管脚输出 = PWMB (PWMB[n]).<br>若 PNP (PNP[n])=1, PWMn 管脚输出 = PWMB (PWMB[n])反向.<br>● 软件清BKLK0 (PWMSTS[8])位前，PWM_CH0~5 输出会保持 PWMB (PWMB[n]) 状态.<br>● PWMRUN (PWMCON[7]) 位保持置位以保证PWM发生器的运行<br>● 如果BKLK0 (PWMSTS[8]) 位被清楚，刹车状态将会在下一个PWM中期被释放。.<br>● 若果在清除BKLK0 (PWMSTS[8]) 之前清除PWMRUN (PWMCON[7])，PWM_CH0~5 输出将维持PWMB (PWMB[5:0])的状态。 |
| BKP1<br>(I/O 口);<br>CO0 (ACMPSR[8]) | 刹车0边沿检测检测到下降沿；<br>低电平 | 1. 刹车1边沿检测检测到下降沿，BKF1 (PWMSTS[1])标志位置1；可通过软件清除。<br>2. 如果LVDBK_EN (PWMCON[27])=1 且刹车1信号检测为低， |

| | | |
|---|---|---|
| (比较强0输出);<br>CO1 (ACMPSR[9])<br>(比较强1输出);<br>CO2 (ACMPSR[10])<br>(比较强2输出) | | PWM0~5 管脚输出跟随PWMB (PWMB[5:0]). 否则, PWM_CH0~5 输出继续跟随PWM发生器的输出。这两种情况下， PNP (PNP[5:0])都可以控制管脚的极性.<br>3. PWMRUN (PWMCON[7]) 位保持置位以保证PWM发生器的运行<br>4. EPWMx_BRAKE1 (x=0~1)管脚变为高电平，PWM_CH0~5 恢复。 下一个PWM周期，PWM_CH0~5 会恢复。 |

表 6-9 刹车源，刹车条件和动作

**注意:** 为了达到以上的效果，BK0_EN 和 BK1_EN两个刹车使能位一定要使能。

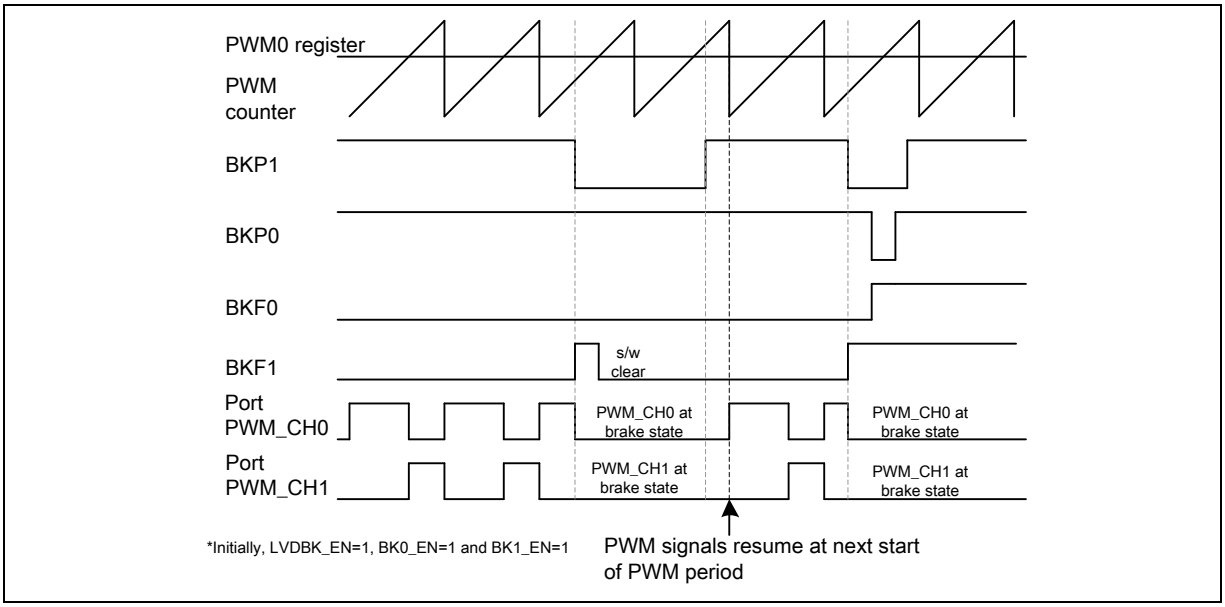



图 6-50 PWM 刹车条件 (边沿对齐模式)

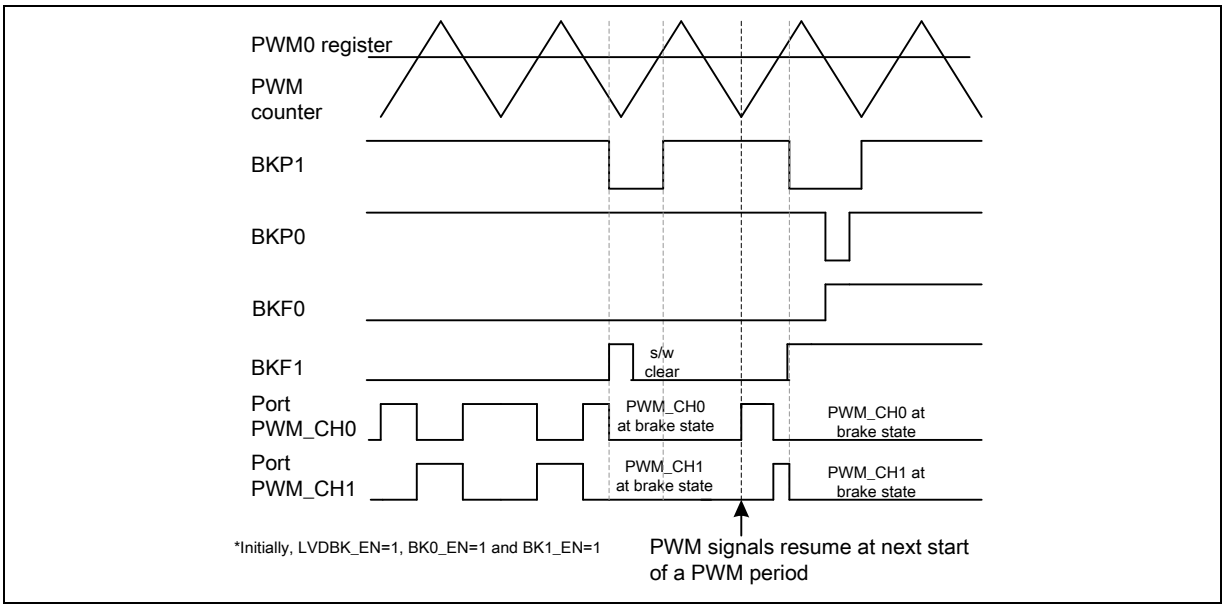



图 6-51 PWM 刹车条件 (中心对齐模式)

由于刹车条件发生时，硬件会自动置位BKFn，用户程序可以读取这些刹车标志位或者使能PWM刹车中断来判断是哪一个条件触发了刹车。

#### 6.8.5.3 *PWM 管脚输出驱动控制*

M0519有两个EPWM单元，每一个单元有6个输出管脚。单元0的PWM输出管脚为P0.0~P0.5，单元1的为P1.0~P1.5。

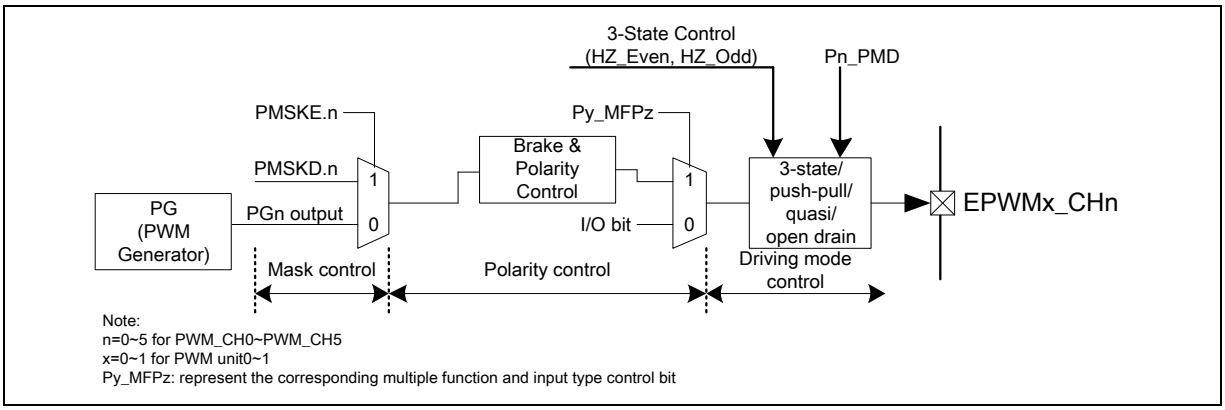



图 6-52 PWM输出驱动控制

复位后PWM输出管脚的驱动状态通过Pn_PMD 寄存器可以初始化成三态或者其他状态。如图所示，PWM输出结构由配置位，HZ_Even0/1 (PWMPOEN[0, 2]), Hz_Odd0/1 (PWMPOEN[1, 3] 和 Pn_PMD 控制。

| HZ_Even/HZ_Odd<br>(寄存器 PWMPOEN 内) | Even/Odd PWM 输出驱动模式 |
|---|---|
| 0 | 取决于 Pn_PMD |
| 1 | 驱动模式为 Hi-Z. |

表 6-10 Even/Odd 输出驱动模式

**注意:** 复位之后，寄存器HZ_Even 和 HZ_Odd的值为config0的锁存值。

#### 6.8.5.4 *PWM 模式*

我们有着功能强劲的PWM单元：支持独立模式，可应用于直流或无刷直流电机系统；支持互补模式，辅以死区功能可用于交流感应电机和同步电机的应用；支持同步模式，可以使每组的两个引脚相位同相。此外，组模式可以强制使PWM0和PWM2与PWM4发生器与PWM0同步，可以在直流和无刷直流电机应用中简化更新占空比操作。

##### 6.8.5.4.1 *独立模式*

当 PMOD (PWMCON[1:0]) = 00b时，独立模式使能。

默认情况下，PWM工作在独立模式，偶数的PWM通道输出：PWM0，PWM2，PWM4。每个通道运行在各自的占空比周期发生模块。在PWM输出屏蔽功能关闭时(PMSKE (PMSKE[5:0]) =0x00)，PWM1，PWM3，PWM5默认复位为0。

#### *6.8.5.4.2 互补模式*

当 PMOD (PWMCON[1:0]) = 01b时，互补模式使能。

在这个模块下有三个专为互补模式提供的占空比-周期发生器， 共有三对PWM输出。 6个输出通道根据奇偶分组。互补模式下，奇数PWM通道PGx与相应的偶数PWM通道互补，如PG1与PG0互补，PG3与PG2互补，PG5与PG4互补。 PWM的时间基准由其自身的16位计数器提供，它也有可选的预分频设置。

### *6.8.5.5 插入死区时间*

死区发生器在一组互补的PWM中，把一个管脚的波形从关闭状态到打开状态之间的偏移时间叫做“死区时间”。这样能对外接的电源切换设备起到保护作用。一对互补PWM输出通道有一个11位向下计数器产生死去时间。死区输出维持到计数器向下计数到0。

死区时间计算公式如下：

死区时间 = EPWMx_CLK * (DTCNT (PDTC[10:0])+1).

图 6-53 是一组PWM信号间插入死区时间的时序图。

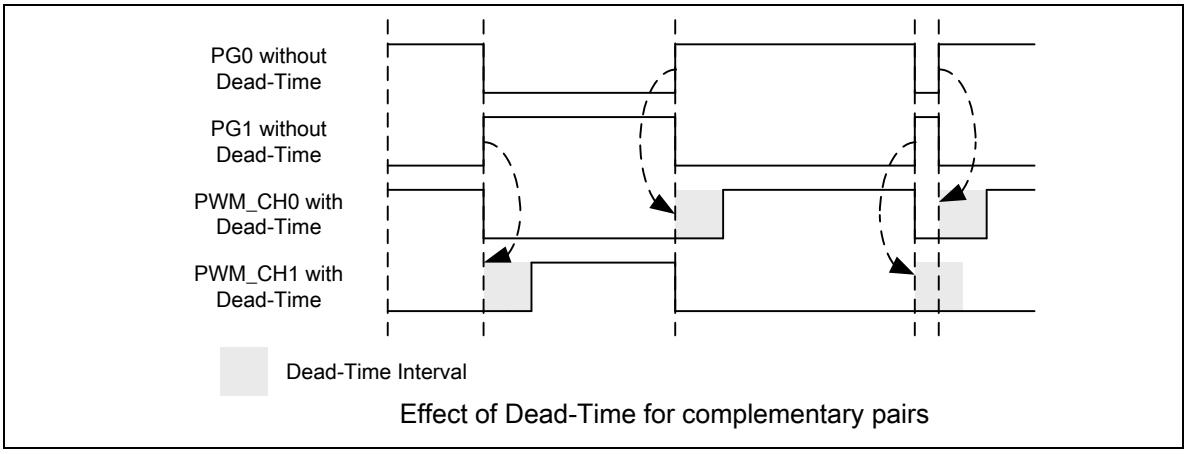



图 6-53 死区时间插入

PWM死区控制寄存器，PDTC是写保护寄存器。

在电源翻转的应用中，死区时间插入避免上下半桥开关同时打开的情况。所以插入死区时间控制对系统正常工作是至关重要的。通道打开和关闭不能实现立即切换，所以在切换过程中需要留有一些时间，死区功能就恰好实现了这个延时。

### *6.8.5.6 同步模式*

当 PMOD (PWMCON[1:0]) = 10b时，同步模式使能。

同步模式中，来自于同一PWM发生器的PWM信号对处于相位相同。

PG1=PG0, PG3=PG2 和 PG5=PG4.。

### 6.8.5.7 组模式

当GRP (PWMCON[13]) = 1，组模式使能。

支持组模式控制，此模式中所有偶数PWM通道输出占空比都由PWM0_CH0占空比寄存器控制。

如果 GRP = 1, 那么 (PG2, PG3) 和 (PG4, PG5) 都会跟随 (PG0, PG1), 即：

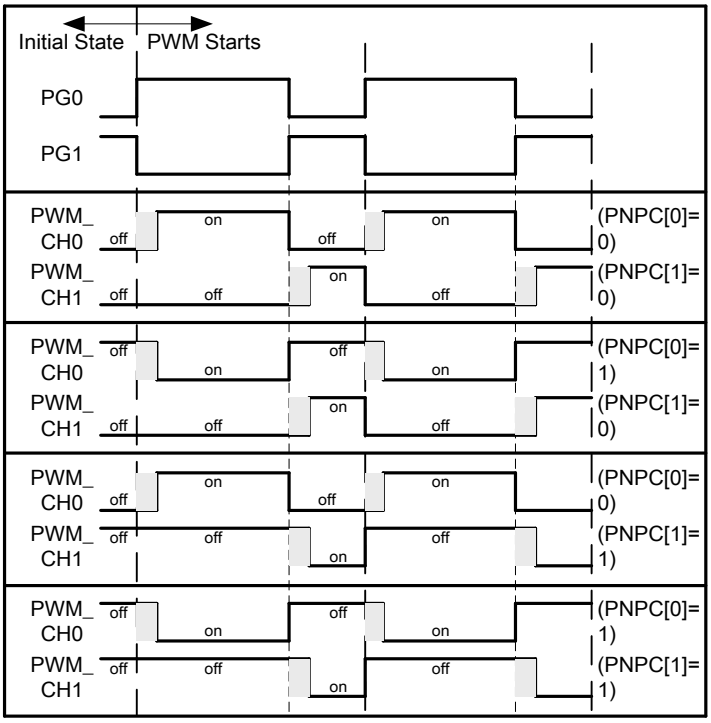
PG4 = PG2 = PG0；

如果互补模式使能(PMOD (PWMCON[1:0])=01b)，PG5 = PG3 = PG1 = 反向 (PG0)

### 6.8.5.8 极性控制

从PWM_CH0 到PWM_CH5，每个PWM口都有一个独立的极性控制模组来配置PWM输出的有效状态。 PWM默认输出高有效。这意味着PWM关状态是低，开状态是高。但对于每个独立的PWM通道而言，通过设定PWM负极性控制寄存器(PNPC)，这种设定是可更改的。

图 6-54描述了PWM在不同极性设定时启动的初始状态。




Dead-time insertion; It is only effective in complementary mode

PNPC[x]: Negative Polarity control bits; It controls the PWM output initial state and polarity

PWM output with dead-time insertion and polarity control

图 6-54 上升沿死区时间插入时的初始状态和极性控制

### 6.8.5.9 PWM 屏蔽输出功能

每个PWM输出通道都可以手动配置。通过设置对应在PWM 屏蔽使能控制寄存器(PMSKE)和PWM屏蔽数据寄存器(PMSKD)中的设定位，在每个独立比较单元区间都可驱动BPWM通道输出指定逻辑电平。 当控制不同类型的电子换向电机(ECM)，比如BLDC马达， PWM 屏蔽位将会非常有用。PMSKE寄存器包含六个比特位，PMSKE[5:0]决定哪些PWM通道输出将被屏蔽（覆盖）。复位后PMSKD默认为0x00H。PMSKE[5:0]为高有效。当PMSKE[5:0]被置位后，对应的PMSKD[5:0]位会相应的作用于PWM通道。当PMSKE 某一位被置位后，PMSKD位和极性控制位决定了对应PWM I/O管脚的输出。

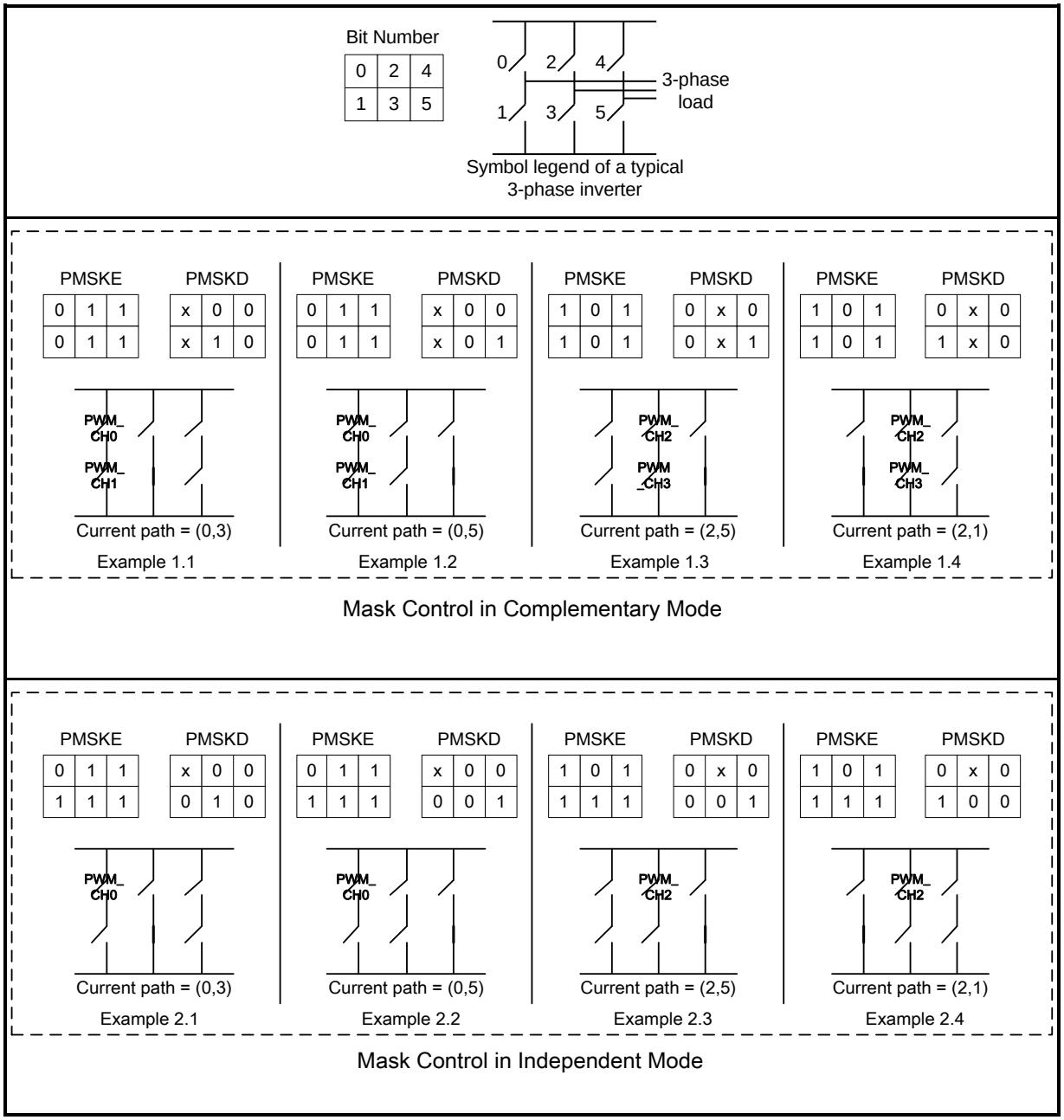



图 6-55 图解屏蔽控制

上图举例解释了PWM屏蔽控制是怎么实现屏蔽功能的。

***例 1.1： PMSKE[5:0] = 11 1100b, PMSKD[5:0] = 0010xxb (互补模式)***

PWM 通道 2-5 被PWM 频率/占空比 发生器屏蔽

PWM 通道2-5 输出由PMSKD状态决定

PWM 通道0和通道1跟随PWM发生器

Switch 0 (On/Off) : 由PWM_CH0控制 (PWM0 频率/占空比 发生器).

Switch 1 (On/Off) : 由PWM_CH1控制(互补模式，PWM0的反向).

Switch 2 (Off) : PMSKD[2] = 0.

Switch 3 (On) : PMSKD[3] = 1.

Switch 4 (Off) : PMSKD[4] = 0.

Switch 5 (Off) : PMSKD[5] = 0.

***例 1.3： PMSKE[5:0] = 11 0011b, PMSKD[5:0] = 10xx00b (互补模式)***

PWM 通道 0, 1, 4 和 5 被PWM 频率/占空比 发生器屏蔽

PWM 通道 0, 1, 4 和 5 输出由PMSKD状态决定

PWM 通道 2 and 3 跟随PWM 发生器.

Switch 0 (Off) : PMSKD[0] = 0.

Switch 1 (Off) : PMSKD[1] = 0.

Switch 2 (On/Off) : 由PWM_CH2控制 (PWM2 频率/占空比 发生器).

Switch 3 (On/Off) : 由PWM_CH3控制(互补模式，PWM2的反向).

Switch 4 (Off) : PMSKD[4] = 0.

Switch 5 (On) : PMSKD[5] = 1.

***例 2.1; PMSKE[5:0] = 11 1110b, PMSKD[5:0] = 00100xb (独立模式)***

PWM 通道 1-5 are 被PWM 频率/占空比 发生器屏蔽.

PWM 通道 1-5 输出由PMSKD状态决定.

PWM channel 0 跟随PWM 发生器.

Switch 0 (On/Off) : 由PWM_CH0控制 (PWM0 频率/占空比 发生器).

Switch 1 (Off) : PMSKD[1] = 0.

Switch 2 (Off) : PMSKD[2] = 0.

Switch 3 (On) : PMSKD[3] = 1.

Switch 4 (Off) : PMSKD[4] = 0.

Switch 5 (Off) : PMSKD[5] = 0.

***例 2.3; PMSKE[5:0] = 11 1011b, PMSKD[5:0] = 100x00b (独立模式)***

PWM 通道 0,1,3,4 和 5 被PWM 频率/占空比 发生器屏蔽.

PWM 通道 0,1,3,4 和 5 输出由PMSKD状态决定.

PWM channel 2 跟随PWM 发生器.

Switch 0 (Off) : PMSKD[0] = 0.

Switch 1 (Off) : PMSKD[1] = 0.

Switch 2 (On/Off) : 由PWM_CH2控制 (PWM2 频率/占空比 发生器).

Switch 3 (Off) : PMSKD[3] = 0.

Switch 4 (Off) : PMSKD[4] = 0.

Switch 5 (On) : PMSKD[5] = 1.

#### *6.8.5.10 EPWM触发 EADC*

如图 图 6-56，当PWM通道输出上升或下降时，EPWM可以产生触发信号到EADC。

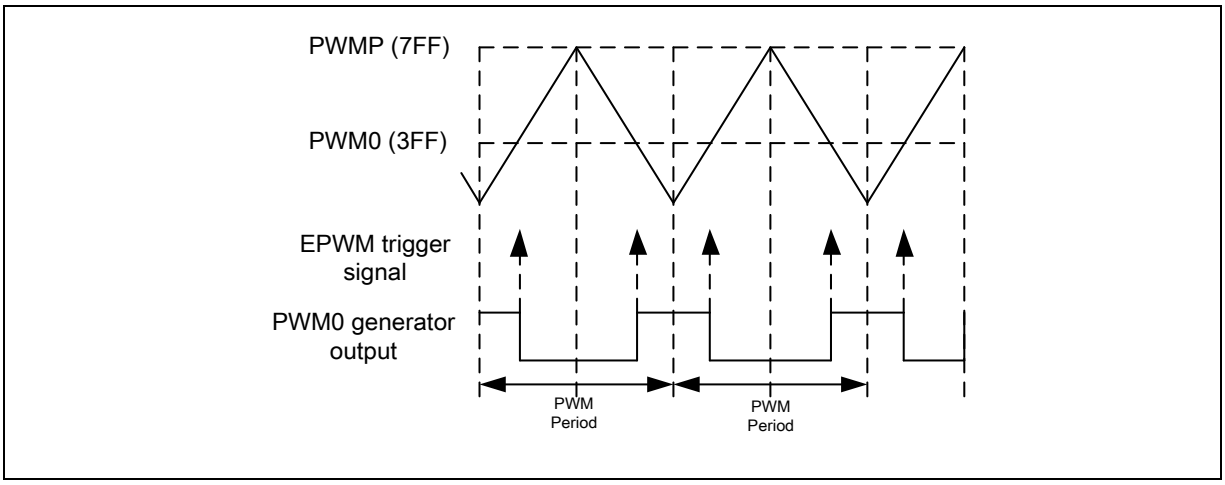



图 6-56 EPWM 触发EADC时序图

*6.8.5.11 EPWM的中断结构*

每个PWM单元都有6个中断源，包括PWM周期标志(PWMF)、刹车0标志(BKF0)、刹车1(BKF1)、PWM0边沿标志、PWM2边沿标志、PWM4边沿标志。BRKI_EN (PWMCON[5])位控制刹车中断使能；PWMI_EN (PWMCON[4])位控制PWM周期中断使能；PWM0EI_EN (PWMEIC[0])位控制PWM0边沿中断使能；PWM2EI_EN (PWMEIC[1]) 控制PWM2边沿中断使能；PWM4EI_EN (PWMEIC[2]) 控制PWM4边沿中断使能。

**注意: 所有中断标志由硬件自动置位；清除时必须通过软件向标志位写1清0。**

如图 6-57 所示是EPWM中断的结构图。

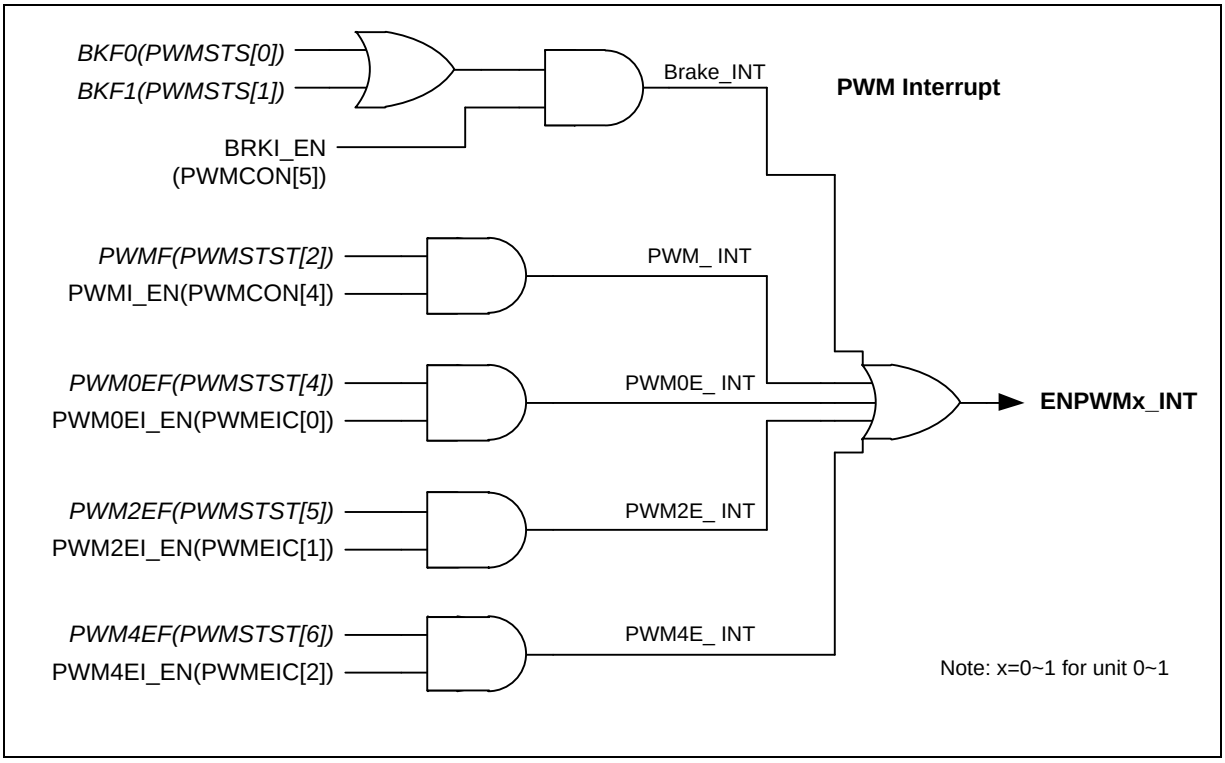



图 6-57 EPWM中断的结构图

### 6.8.6 寄存器映射

**R: 只读, W: 只写, R/W: 读/写**

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **EPWM 基地址:**<br>**EPWMx_BA = 0x4019_0000 + (0x4000 * x)**<br>**x = 0, 1** | | | | |
| **PWMCON** | EPWMx_BA+0x00 | R/W | EPWM 控制寄存器 | 0x0000_0000 |
| **PWMSTS** | EPWMx_BA+0x04 | R/W | EPWM 状态寄存器 | 0x0000_0000 |
| **PWMP** | EPWMx_BA+0x08 | R/W | EPWM 周期寄存器 | 0x0000_0000 |
| **PWM0** | EPWMx_BA+0x0C | R/W | EPWM PWM0 占空比寄存器 | 0x0000_0000 |
| **PWM2** | EPWMx_BA+0x10 | R/W | EPWM PWM2 占空比寄存器 | 0x0000_0000 |
| **PWM4** | EPWMx_BA+0x14 | R/W | EPWM PWM4 占空比寄存器 | 0x0000_0000 |
| **PMSKE** | EPWMx_BA+0x18 | R/W | EPWM 屏蔽模式使能寄存器 | 0x0000_0000 |
| **PMSKD** | EPWMx_BA+0x1C | R/W | EPWM 屏蔽模式数据寄存器 | 0x0000_0000 |
| **PDTC** | EPWMx_BA+0x2C | R/W | EPWM 死区控制寄存器 | 0x0000_0000 |
| **PWMB** | EPWMx_BA+0x30 | R/W | EPWM 刹车输出寄存器 | 0x0000_0000 |
| **PNPC** | EPWMx_BA+0x34 | R/W | EPWM 负极性控制 | 0x0000_0000 |
| **PWMFCNT** | EPWMx_BA+0x3C | R/W | EPWMF 比较计数寄存器 | 0x0000_0000 |
| **PWMEIC** | EPWMx_BA+0x40 | R/W | EPWM 边沿中断控制寄存器 | 0x0000_0000 |

### 6.8.7 寄存器描述

**EPWM 控制寄存器(PWMCON)**

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PWMCON** | EPWMx_BA+0x00 | R/W | EPWM 控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **CLDMD** | **Reserved** | **BK1NF_DIS** | **BK0NF_DIS** | **LVDBK_EN** | **CPO2BK_EN** | **CPO1BK_EN** | **CPO0BK_EN** |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **BK1FILT** | | **BK0FILT** | | **BK1SEL** | | **BKEN1** | **BKEN0** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **INVBKP1** | **INVBKP0** | **GRP** | **PWMTYPE** | **CLRPWM** | **Reserved** | **PWMINV** | **INT_TYPE** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PWMRUN** | **LOAD** | **BRKI_EN** | **PWMI_EN** | **PWMDIV** | | **PWMMOD** | |

| 位 | 描述 | |
|---|---|---|
| [31] | **CLDMD** | **中心重载模式使能位**<br>0 = PWM计数到周期点，EPWM重载占空比寄存器<br>1 = PWM计数到中心点，EPWM重载占空比寄存器<br>该位仅在EPWM处于中心对齐时有效 |
| [30] | **Reserved** | 保留. |
| [29] | **BK1NF_DIS** | **PWM 刹车1噪声滤波禁止位**<br>0 = PWM 刹车1噪声滤波使能<br>1 = PWM 刹车1噪声滤波禁止 |
| [28] | **BK0NF_DIS** | **PWM 刹车0噪声滤波禁止位**<br>0 = PWM 刹车0噪声滤波使能<br>1 = PWM 刹车0噪声滤波禁止 |
| [27] | **LVDBK_EN** | **低电检测触发PWM刹车功能1使能位**<br>0 = 低电检测触发PWM刹车功能1禁止<br>1 = 低电检测触发PWM刹车功能1使能 |
| [26] | **CPO2BK_EN** | **模拟比较器2数字输出作为刹车0刹车源使能位**<br>0 = CO2 (ACMPSR[10]) 作为刹车0刹车源禁止<br>1 = CO2 (ACMPSR[10]) 作为刹车0刹车源使能 |
| [25] | **CPO1BK_EN** | **模拟比较器1数字输出作为刹车0刹车源使能位**<br>0 = CO1 (ACMPSR[9]) 作为刹车0刹车源禁止<br>1 = CO1 (ACMPSR[9]) 作为刹车0刹车源使能 |

| 位 | 描述 | |
|---|---|---|
| [24] | **CPO0BK_EN** | **模拟比较器0数字输出作为刹车0刹车源使能位**<br>0 = CO0 (ACMPSR[8]) 作为刹车0刹车源禁止<br>1 = CO0 (ACMPSR[8]) 作为刹车0刹车源使能 |
| [23:22] | **BK1FILT** | **刹车1(BKPx1 管脚)边沿检测滤波时钟选择**<br>00 = 滤波时钟为 HCLK.<br>01 = 滤波时钟为 HCLK/2.<br>10 = 滤波时钟为 HCLK/4.<br>11 = 滤波时钟为 HCLK/16. |
| [21:20] | **BK0FILT** | **刹车0(BKPx0 管脚)边沿检测滤波时钟选择**<br>00 = 滤波时钟为 HCLK.<br>01 = 滤波时钟为 HCLK/2.<br>10 = 滤波时钟为 HCLK/4.<br>11 = 滤波时钟为 HCLK/16. |
| [19:18] | **BK1SEL** | **刹车功能1源选择**<br>00 = 刹车信号来自外部管脚EPWMx_BRAKE1 (x=0~1 对应PWM单元0~1).<br>01 = 刹车信号来自比较器0输出CO0 ACMPSR[8]).<br>10 = 刹车信号来自比较器1输出CO1 (ACMPSR[9]).<br>11 = 刹车信号来自比较器2输出CO2 (ACMPSR[10]). |
| [17] | **BKEN1** | **刹车1管脚触发刹车功能1使能位**<br>0 = PWMx刹车功能1禁止<br>1 = PWMx刹车功能1使能<br>**注意:** x=0~1 对应 PWM 单元0~1. |
| [16] | **BKEN0** | **刹车0管脚触发刹车功能0使能位**<br>0 = PWMx刹车功能0禁止<br>1 = PWMx刹车功能0使能<br>**注意:** x=0~1 对应 PWM 单元0~1. |
| [15] | **INVBKP1** | **反转刹车1管脚状态**<br>0 = EPWMx_BRAKE1 管脚状态直接送到负边沿检测模块<br>1 = EPWMx_BRAKE1 管脚状态反转后送到负边沿检测模块 |
| [14] | **INVBKP0** | **反转刹车0管脚状态**<br>0 = EPWMx_BRAKE0 管脚状态直接送到负边沿检测模块<br>1 = EPWMx_BRAKE0 管脚状态反转后送到负边沿检测模块 |
| [13] | **GRP** | **组模式位**<br>0 = PWM_CH0, PWM_CH2 和PWM_CH4的信号时序相独立<br>1 = 统一PWM_CH0, PWM_CH2 和PWM_CH4的信号时序到与PWM_CH0相同的相位 |
| [12] | **PWMTYPE** | **PWM 对齐类型选择位**<br>0 = 边沿对齐<br>1 = 中心对齐 |

| 位 | 描述 | |
|---|---|---|
| [11] | **CLRPWM** | **PWM计数器清除控制位**<br>0 = 忽略.<br>1 = 清除16位的PWM计数器到0000H.<br>**注意:** 该位由硬件自动清0. |
| [10] | **Reserved** | 保留. |
| [9] | **PWMINV** | **反转PWM比较器输出**<br>PWMINV 为高时，PWM比较器输出信号会被反转。因此，在PWMINV置1前，PWM占空比要改为1-duty<br>0 = 不反转PWM比较器输出.<br>1 = 反转PWM比较器输出. |
| [8] | **INT_TYPE** | **PWM中断类型选择位**<br>0 = PWM计数器下溢时，PWMF置位.<br>1 = PWM计数器计数到PWMP寄存器值时，PWMF置位.<br>**注意: 该位仅在中心对齐模式下有效** |
| [7] | **PWMRUN** | **PWMRUN启动 控制位**<br>0 = PWM停止运行.<br>1 = PWM计数器开始工作. |
| [6] | **LOAD** | **重载PWM周期寄存器(PWMP)和PWM占空比(PWM0~4)控制位**<br>0 = 写0无动作. PWM周期寄存器(PWMP)和PWM占空比(PWM0~4)的值不会重载到PWM计数器和比较强寄存器.<br>1 = 边沿对齐模式和中心对齐模式下PWM计数器计数到PWMP时，或者中心对齐模式下PWM计数器向下计数到下溢时，硬件将会更新PWM周期寄存器(PWMP)和PWM占空比(PWM0~4)的值到PWM计数器和比较强寄存器.<br>**注意:** 该位由软件置位，硬件清0；读该位始终为0. |
| [5] | **BRKI_EN** | **刹车0和刹车1中断使能位**<br>0 = 禁止BKF0 (PWMSTS[0]) 和 BKF1 (PWMSTS[1]) 标志位触发PWM中断.<br>1 = 允许BKF0 (PWMSTS[0]) 和 BKF1 (PWMSTS[1]) 标志位触发PWM中断. |
| [4] | **PWMI_EN** | **PWM 中断使能位**<br>0 = 禁止PWMF (PWMSTS[2]) 标志位触发PWM中断.<br>1 = 允许PWMF (PWMSTS[2]) 标志位触发PWM中断. |
| [3:2] | **PWMDIV** | **PWM 时钟预分频选择**<br>00 = PWM 时钟为 EPWMx_CLK.<br>01 = PWM 时钟为 EPWMx_CLK/2.<br>10 = PWM 时钟为 EPWMx_CLK/4.<br>11 = PWM 时钟为 EPWMx_CLK/16. |
| [1:0] | **PWMMOD** | **PWM 模式选择**<br>00 = 独立模式.<br>01 = 互补模式.<br>10 = 同步模式.<br>11 = 保留. |

## EPWM 状态寄存器 (PWMSTS)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PWMSTS** | EPWMx_BA+0x04 | R/W | EPWM 状态寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | BK1STS | BK0STS |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| Reserved | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| Reserved | | | | | | | BKLK0 |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Reserved | PWM4EF | PWM2EF | PWM0EF | Reserved | PWMF | BKF1 | BKF0 |

| 位 | 描述 | |
|---|---|---|
| [31:26] | **Reserved** | 保留. |
| [25] | **BK1STS** | **刹车1状态 (只读)**<br>0 = PWM 不在刹车1状态.<br>1 = PWM 处于刹车1状态. |
| [24] | **BK0STS** | **刹车0状态 (只读)**<br>0 = PWM 不在刹车1状态.<br>1 = PWM 处于刹车1状态. |
| [23:9] | **Reserved** | 保留. |
| [8] | **BKLK0** | **PWM 刹车0锁定**<br>0 = 刹车0释放.<br>1 = 当PWM刹车检测在EPWMx_BRAKE0, x=0, 1上检测到下降沿信号，这个标志位会被置高，即刹车0处于锁定状态.<br>**注意:** 该位必须通过软件写1清0. |
| [7] | **Reserved** | 保留. |
| [6] | **PWM4EF** | **PWM通道4边沿标志**<br>0 = PWM_CH4 没有边沿信号.<br>1 = PWM_CH4上升或下降时硬件会 置位该位.如果EINT4_TYPE (PWMEIC[10]) = 0, 检测到PWM_CH4下降沿时，置位该位.如果 EINT4_TYPE (PWMEIC[10]) = 1, 检测到PWM_CH4上降沿时.<br>**注意:** 该位必须通过软件写1清0. |

| | | |
|---|---|---|
| [5] | **PWM2EF** | **PWM通道2边沿标志**<br>0 = PWM_CH2没有边沿信号.<br>1 = PWM_CH2上升或下降时硬件会 置位该位.如果EINT2_TYPE (PWMEIC[9]) = 0, 检测到PWM_CH4下降沿时，置位该位.如果 EINT2_TYPE (PWMEIC[9]) = 1, 检测到PWM_CH2上降沿时.<br>**注意:** 该位必须通过软件写1清0. |
| [4] | **PWM0EF** | **PWM通道0边沿标志**<br>0 = PWM_CH0没有边沿信号.<br>1 = PWM_CH0上升或下降时硬件会 置位该位.如果EINT0_TYPE (PWMEIC[8]) = 0, 检测到PWM_CH0下降沿时，置位该位.如果 EINT0_TYPE (PWMEIC[8]) = 1, 检测到PWM_CH0上降沿时.<br>**注意:** 该位必须通过软件写1清0. |
| [3] | **Reserved** | 保留. |
| [2] | **PWMF** | **PWM 周期标志**<br>0 = PWM计数器向上计数还未达PWMP的值或者向下计数未到下溢<br>1 =边沿对齐模式和中心对齐模式下PWM计数器计数到PWMP时，或者中心对齐模式下PWM计数器向下计数到下溢时，该位置1.<br>**注意:** 该位必须由软件写1清0. |
| [1] | **BKF1** | **PWM 刹车1标志**<br>0 = PWM刹车1轮询EPWMx_BRAKE1上的下降沿信号，但没有检测到对应信号, x=0, 1.<br>1 = PWM刹车1在EPWMx_BRAKE1上检测到下降沿信号，该位置1. x=0, 1.<br>**注意:** 该位必须由软件写1清0. |
| [0] | **BKF0** | **PWM 刹车0标志**<br>0 = PWM刹车0轮询EPWMx_BRAKE0上的下降沿信号，但没有检测到对应信号, x=0, 1.<br>1 = PWM刹车0在EPWMx_BRAKE0上检测到下降沿信号，该位置1. x=0, 1.<br>**注意:** 该位必须由软件写1清0. |

## EPWM 周期寄存器 (PWMP)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PWMP** | EPWMx_BA+0x08 | R/W | EPWM 周期寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留. |
| [15:0] | **PWMP** | **PWM 周期寄存器**<br>**边沿对齐模式**:<br>周期 = (PWMP +1) * EPWMx_CLK的周期/预分频值.<br>占空比 = (Duty + 1) * EPWMx_CLK的周期/预分频值.<br>**中心对齐模式**:<br>周期 = (PWMP* 2) * EPWMx_CLK的周期/预分频值.<br>占空比 = (Duty*2 + 1) * EPWMx_CLK的周期/预分频值. |

## EPWM 占空比寄存器 (PWM0/2/4)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PWM0** | EPWMx_BA+0x0C | R/W | EPWM PWM0占空比寄存器 | 0x0000_0000 |
| **PWM2** | EPWMx_BA+0x10 | R/W | EPWM PWM2占空比寄存器 | 0x0000_0000 |
| **PWM4** | EPWMx_BA+0x14 | R/W | EPWM PWM4占空比寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留. |
| [15:0] | **PWM_Duty** | **PWM 占空比寄存器**<br>**边沿对齐模式**:<br>周期 = (PWMP +1) * EPWMx_CLK的周期/预分频值.<br>占空比 = (Duty + 1) * EPWMx_CLK的周期/预分频值.<br>**中心对齐模式**:<br>周期 = (PWMP* 2) * EPWMx_CLK的周期/预分频值.<br>占空比 = (Duty*2 + 1) * EPWMx_CLK的周期/预分频值. |

## EPWM 屏蔽模式使能寄存器(PMSKE)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PMSKE** | EPWMx_BA+0x18 | R/W | EPWM 屏蔽模式使能寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:6] | **Reserved** | 保留. |
| [5:0] | **PMSKE** | **PWM 屏蔽使能位**<br>使能该位会屏蔽PWM发生器的信号. 对于PWM通道n会按PMSKD[n]的值进行输出.<br>0 = 下一阶段, PWM发生器信号正常输出.<br>1 = 下一阶段, PWM发生器信号被屏蔽，按按PMSKD[n]的值进行输出, n = 0~5. |

## EPWM 屏蔽模式数据寄存器 (PMSKD)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PMSKD** | EPWMx_BA+0x1C | R/W | EPWM 屏蔽模式数据寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:6] | **Reserved** | 保留. |
| [5:0] | **PMSKD** | **PWM 屏蔽数据位**<br>如果PMSKE[n] = 1, n = 0~5, 对应的PWM_CHn按该位数据进行输出<br>0 = PWM_CHn输出低.<br>1 = PWM_CHn输出高. |

## EPWM 死区控制寄存器 (PDTC)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PDTC** | EPWMx_BA+0x2C | R/W | EPWM 死区控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | **DTEN4** | **DTEN2** | **DTEN0** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | **DTCNT** | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **DTCNT** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:19] | **Reserved** | 保留. |
| [18] | **DTEN4** | **使能PWMx 对 (PWM_CH4, PWM_CH5)死区时间插入**<br>只有当互补的PWM对使能，死区时间插入才会生效. 如果死区时间插入没有生效，互补输出的管脚对的输出将没有延时.<br>0 = 禁止管脚对(PWM_CH4, PWM_CH5)的死区时间插入.<br>1 = 使能管脚对(PWM_CH4, PWM_CH5)的死区时间插入.<br>**注意:** x=0~1 对应 PWM 单元0~1. |
| [17] | **DTEN2** | **使能PWMx 对 (PWM_CH2, PWM_CH3)死区时间插入**<br>只有当互补的PWM对使能，死区时间插入才会生效. 如果死区时间插入没有生效，互补输出的管脚对的输出将没有延时.<br>0 = 禁止管脚对(PWM_CH2, PWM_CH3)的死区时间插入.<br>1 = 使能管脚对(PWM_CH2, PWM_CH3)的死区时间插入.<br>**注意:** x=0~1 对应 PWM 单元0~1. |
| [16] | **DTEN0** | **使能PWMx 对 (PWM_CH0, PWM_CH1)死区时间插入**<br>只有当互补的PWM对使能，死区时间插入才会生效. 如果死区时间插入没有生效，互补输出的管脚对的输出将没有延时.<br>0 = 禁止管脚对(PWM_CH0, PWM_CH1)的死区时间插入.<br>1 = 使能管脚对(PWM_CH0, PWM_CH1)的死区时间插入.<br>**注意:** x=0~1 对应 PWM 单元0~1. |
| [15:11] | **Reserved** | 保留. |
| [10:0] | **DTCNT** | **死区时间计数器**<br>按下面的公式计算死区时间:<br>死区时间 = EPWMx_CLK * (DTCNT+1). |

## EPWM 刹车输出寄存器 (PWMB)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PWMB** | EPWMx_BA+0x30 | R/W | EPWM 刹车输出寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:6] | **Reserved** | 保留. |
| [5:0] | **PWMB** | **PWM 刹车输出**<br>发生PWM刹车时,没有极性控制的PWM_CHn输出状态跟随PWMB[n]的设置, n = 0~5.<br>0 = 发生PWM刹车时,没有极性控制的PWM_CHn输出低.<br>1 = 发生PWM刹车时,没有极性控制的PWM_CHn输出高. |

## EPWM 负极性控制寄存器 (PNPC)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PNPC** | EPWMx_BA+0x34 | R/W | EPWM 负极性控制寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:6] | **Reserved** | 保留. |
| [5:0] | **PNP** | **PWM 负极性控制**<br>该寄存器位控制实际PWM_CHn 的输出极性/有效状态, n = 0~5.<br>0 = PWM_CHn 输出高电平有效.<br>1 = PWM_CHn 输出低电平有效. |

## EPWMF 比较计数寄存器 (PWMFCNT)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PWMFCNT** | EPWMx_BA+0x3C | R/W | EPWMF 比较计数寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:4] | **Reserved** | 保留. |
| [3:0] | **PWMFCNT** | **PWMF 比较计数器**<br>该寄存器通过计算PWM周期发生次数来置位PWMF (PWMSTS[2])从而请求PWM周期中断<br>PWMF (PWMSTS[2])会在PWM发生的周期次数或者是中心点 次数（由PWMCON[8]的INT_TYPE决定）达到PWMFCNT[3:0] + 1时被置位。 |

## EPWM 边沿中断控制寄存器 (PWMEIC)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **PWMEIC** | EPWMx_BA+0x40 | R/W | EPWM 边沿中断控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | **EINT4_TYPE** | **EINT2_TYPE** | **EINT0_TYPE** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | **PWM4EI_EN** | **PWM2EI_EN** | **PWM0EI_EN** |

| 位 | 描述 | |
|---|---|---|
| [31:11] | **Reserved** | 保留. |
| [10] | **EINT4_TYPE** | **PWM 通道4边沿中断类型**<br>0 = 检测到PWM_CH4上的下降沿时会清0PWM4EF.<br>1 = 检测到PWM_CH4上的上升沿时会置1PWM4EF. |
| [9] | **EINT2_TYPE** | **PWM 通道2边沿中断类型**<br>0 = 检测到PWM_CH2上的下降沿时会清0PWM2EF.<br>1 = 检测到PWM_CH2上的上升沿时会置1PWM2EF. |
| [8] | **EINT0_TYPE** | **PWM 通道0边沿中断类型**<br>0 = 检测到PWM_CH0上的下降沿时会清0PWM0EF.<br>1 = 检测到PWM_CH0上的上升沿时会置1PWM0EF. |
| [7:3] | **Reserved** | 保留. |
| [2] | **PWM4EI_EN** | **使能PWM通道4边沿中断**<br>0 = 禁止PWM4EF 标志触发PWM中断.<br>1 = 使能PWM4EF 标志触发PWM中断. |
| [1] | **PWM2EI_EN** | **使能PWM通道2边沿中断**<br>0 = 禁止PWM2EF 标志触发PWM中断.<br>1 = 使能PWM2EF 标志触发PWM中断. |
| [0] | **PWM0EI_EN** | **使能PWM通道0边沿中断**<br>0 = 禁止PWM0EF 标志触发PWM中断.<br>1 = 使能PWM0EF 标志触发PWM中断. |

## 6.9 增强型输入捕获定时器 (ECAP)

### 6.9.1 概述

该设备提供两个输入捕获定时器/计数器，它带捕获功能，可以侦测数字边沿的跳变。每一个单元有三个输入捕获通道。该定时器/计数器配备上数，重载和比较匹配模式。

### 6.9.2 特性

- 最大到两个输入捕获定时器/计数器单元：输入捕获0和输入捕获1。
- 每个单元有自己的中断向量
- 24位输入捕获上数定时器/计数器
- 输入端口的前端有噪音过滤器
- 带三个选项的边沿检查
  - 上升沿检测
  - 下降沿检测
  - 上升沿和下降沿检测
- 每一个输入通道支持一个捕获计数器保持寄存器
- 可选捕获事件复位/重载捕获计数器
- 支持比较匹配功能

### 6.9.3 输入捕获定时器/计数器架构

每一个输入捕获定时器/计数器单元支持3个输入通道，带三个可编程的输入信号源。引脚ECAPx_IC0 到 ECAPx_IC2 可以反馈到捕获单元的输入、模拟比较器输出(COx)、运放数字输出(OPDOx)和ADC比较输出(ADCMPOx)。也可以通过软件配置内部路径到捕获输入。图 6-58 说明了输入捕获的架构。

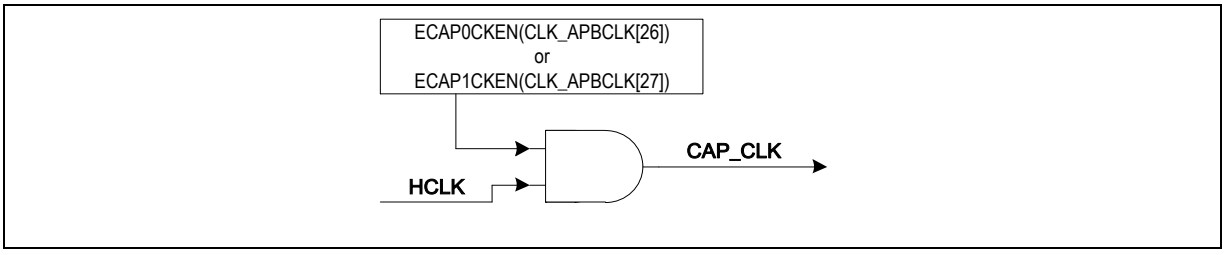



图 6-58 输入捕获定时器/计数器时钟源控制

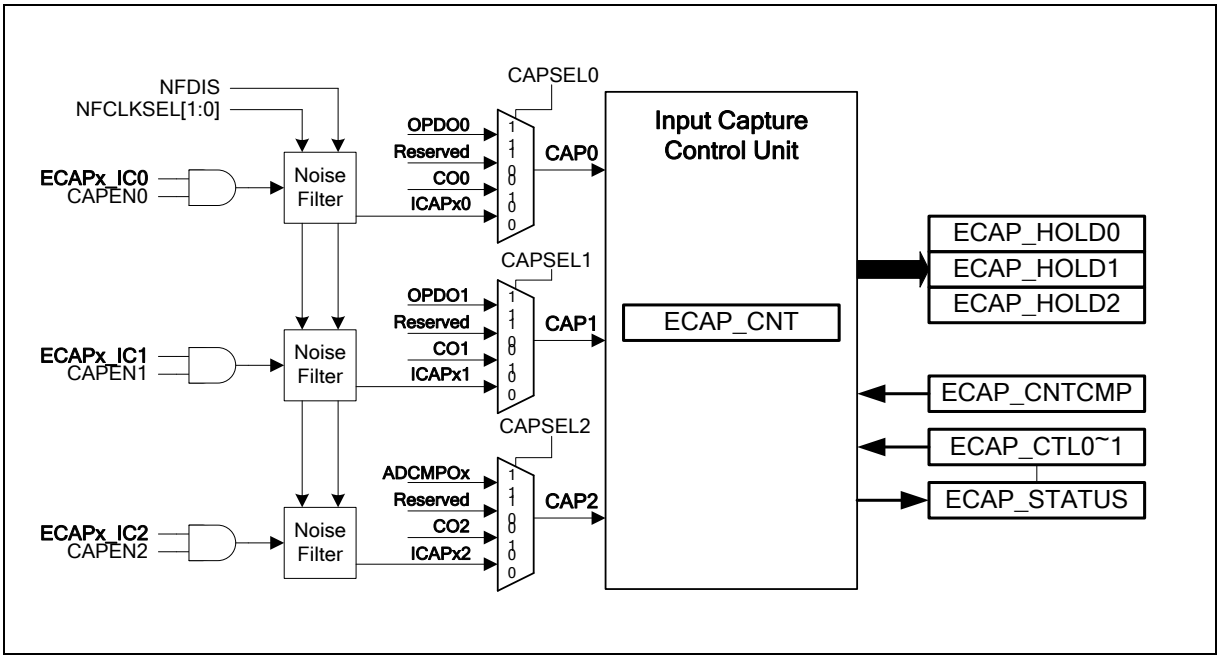



图 6-59 输入捕获定时器/计数器架构

### 6.9.4 输入噪音过滤器

每一个ECAP输入都配备了噪音过滤器，可以过滤掉不想要的噪音。ECAPx_IC0, ECAPx_IC1 和 ECAPx_IC2噪音过滤器可以通过位CAPNF_DIS (ECAP_CTLx[3])禁止。如果使能，捕获逻辑需要采样4个连续相同输入值来识别跳变沿作为捕获事件。两个需要捕获的脉冲之间的间隔需要大于等于4个捕获时钟的宽度。任何小于或等于3个捕获时钟的脉冲宽度都不会触发。ICAPx0, ICAPx1 和 ICAPx2分别是ECAPx_IC0、ECAPx_IC1 和 ECAPx_IC2 经过噪音过滤器和极性控制之后的输出，如图 6-60。如果噪音过滤器被禁止，ECAPx_IC0, ECAPx_IC1 和 ECAPx_IC2的输入信号被分别送到内部信号ICAPx0, ICAPx1 和 ICAPx2，没有任何延时。

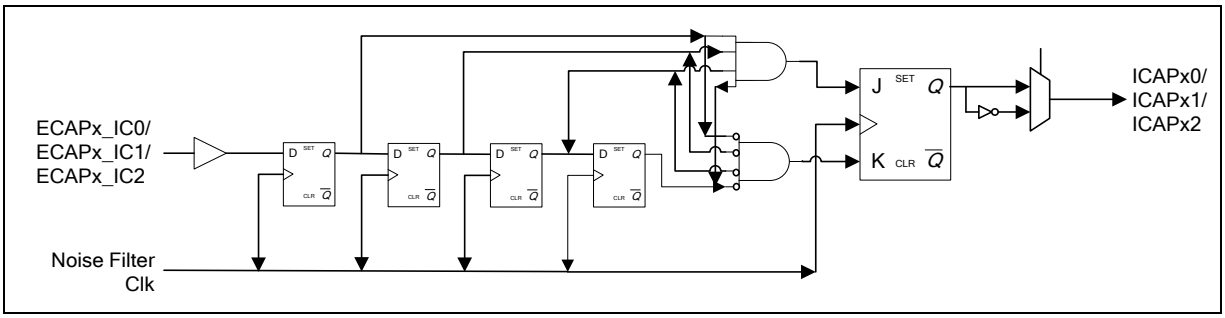



图 6-60 噪音过滤器架构

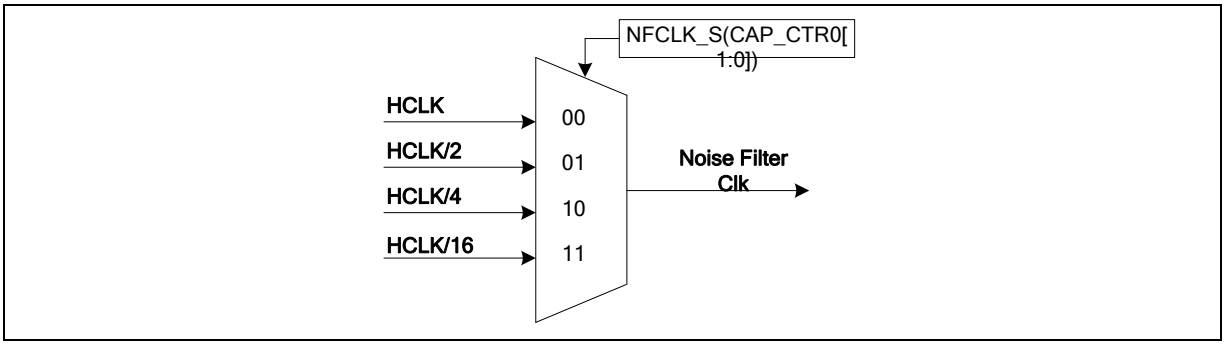



图 6-61 噪音过滤器采样时钟选择

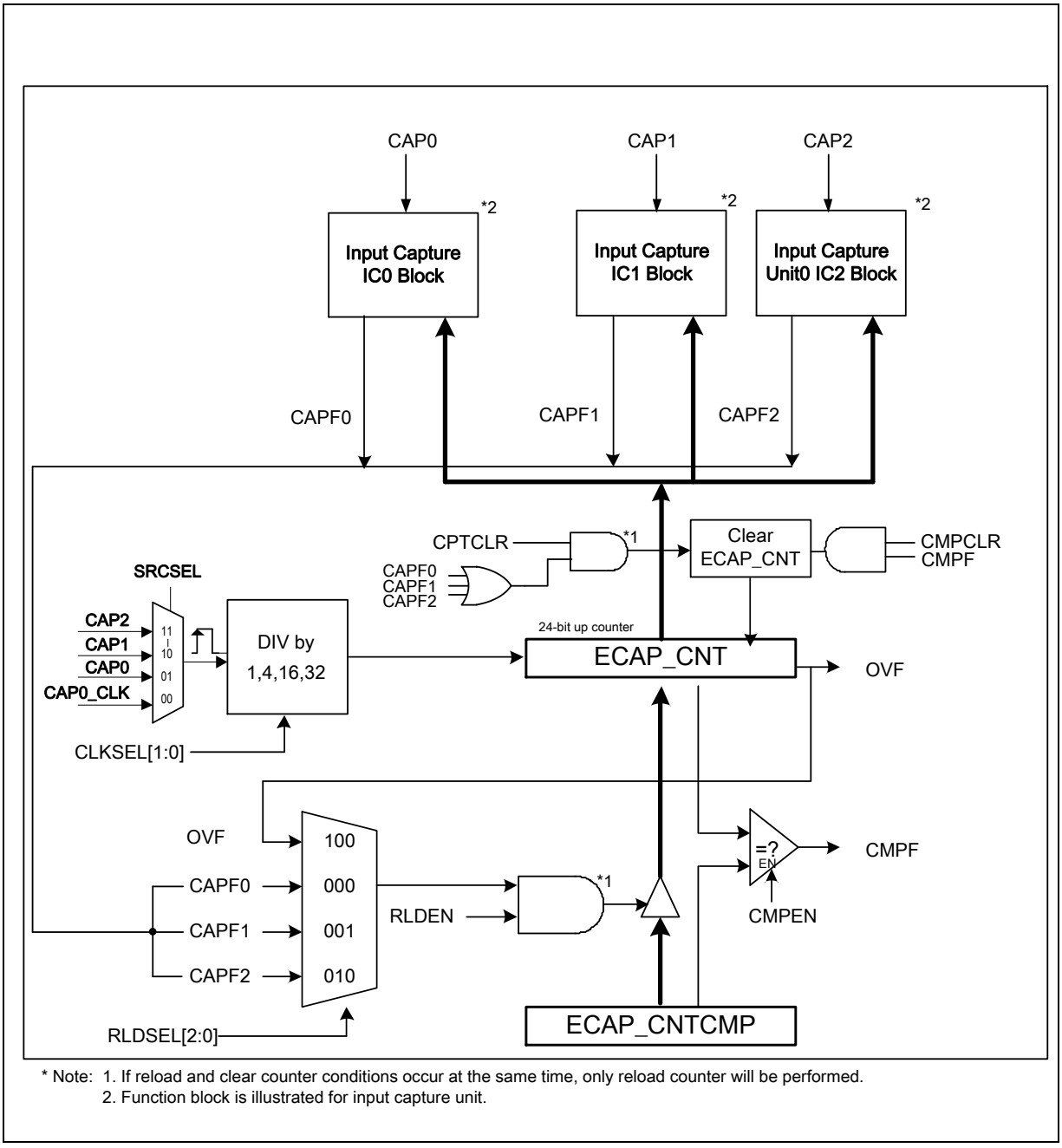
### 6.9.5 输入捕获定时器/计数器的操作

捕获模块功能用于检查和测量出脉冲的宽度和波形的周期。输入通道0到2有他们自己的边沿侦测器但是共享一个捕获定时器/计数器，也就是CAP_CNT。触发选择是可通过CAP_CTR1寄存器中的CAPEDG选择。它支持上升沿触发、下降沿触发和上升下降沿都触发。每一个捕获模块有一个使能控制位IC0_EN 到 IC2_EN。捕获计数器(CAP_CNT)是一个24位的上数计数器。它支持重载和比较模式。输入捕获定时器/计数器使能位(CAP_EN)必须设置为使能输入捕获定时器/计数器功能。更多细节在下面部分描述。

#### *6.9.5.1 捕获功能*

每一次捕获输入触发被识别，24位自由计数捕获计数器ECAP_CNT会被捕获下来并传输到捕获保持寄存器ECAP_HOLD0~2。保存到哪个寄存器取决于哪个通道触发。这个动作也会引起ECAP_STATUS 中的CAPF标志置位，如果CAPIENx (ECAP_CTL0[16]使能，会产生中断。CAPF0~2标志是逻辑或到中断模块，标志由硬件置位，通过软件清零。

设置CPTCLR (ECAP_CTL0[26])，硬件会复位捕获计数器(ECAP_CNT)。ECAP_CNT的值会自动被捕获。在捕获计数器的值到捕获寄存器之后会立即复位计数器。




* Note: 1. If reload and clear counter conditions occur at the same time, only reload counter will be performed.
2. Function block is illustrated for input capture unit.

图 6-62 输入捕获定时器/计数器功能模块

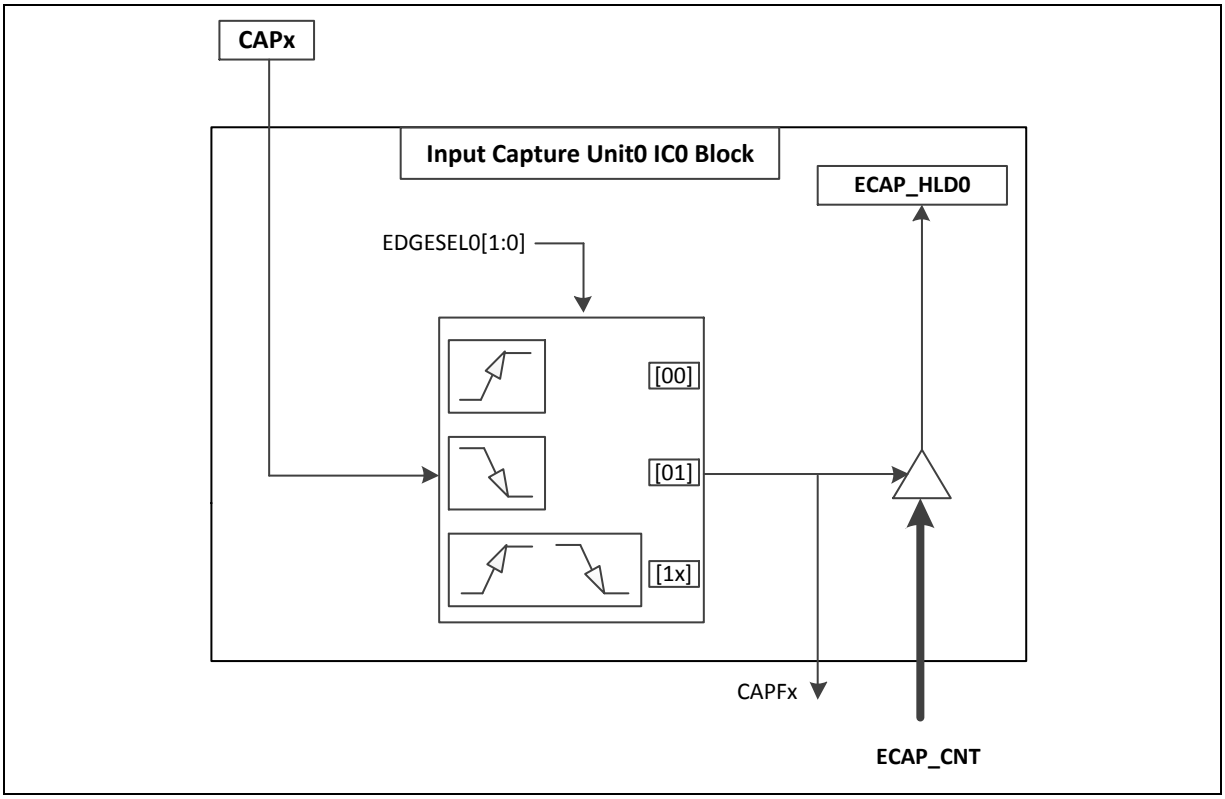



图 6-63 输入捕获单元

### 6.9.5.2 *比较模式*

比较功能通过设置CMPEN (ECAP_CTL0[28]) 为1使能。ECAP_CNTCMP作为比较寄存器，当ECAP_CNT向上计数与ECAP_CNTCMP值匹配的时候CMPF (ECAP_STATUS[4])会置位，如果捕获比较中断使能CMPIEN (ECAP_CTL0[21])位置位，会产生中断请求。然后定时器重载初值0并重新开始计数。

设置CMPCLR (ECAP_CTL0[25]),当匹配发生，硬会件自动复位捕获计数器。

### 6.9.5.3 *重载模式*

输入捕获定时器/计数器也可以配置成重载模式。重载功能通过设置RLDEN (ECAP_CTL0[27])为1使能。在该模式下ECAP_CNTCMP作为重载寄存器。如果RLDEN置位，当ECAP_CNT溢出，ECAP_CNTCMP寄存器的值重载到ECAP_CNT寄存器。然而如果RLDEN = 0，ECAP_CNT会重载0，然后再次向上计数。

作为选择，通过配置RLDSEL (ECAP_CTL1[10:8])，可以选择其他的重载源。

### 6.9.5.4 *输入捕获定时器/计数器中断架构*

对于一个输入捕获单元有五个中断源。每一个有一个中断标志和使能控制位，可以触发输入捕获定时器/计数器中断。注意中断标志由硬件置位必须通过软件清零。

图 6-64 表示输入捕获定时器/计数器的中断架构

.

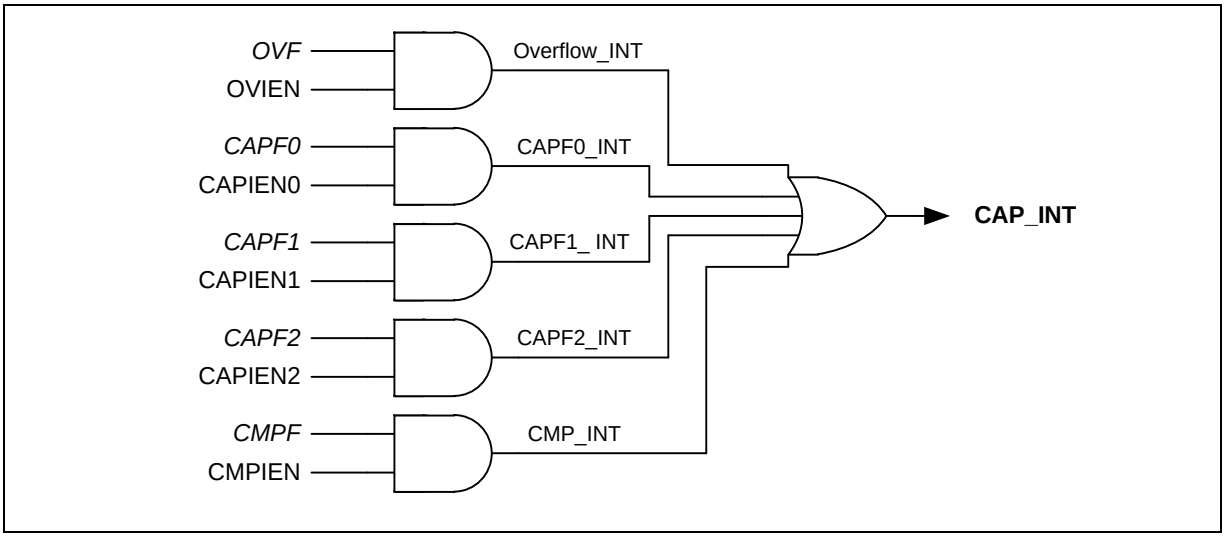



图 6-64 输入捕获定时器/计数器的中断架构框图

### 6.9.6 寄存器映射

**R:** 只读, **W:** 只写, **R/W:** 可读写

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ECAP 基地址:**<br>**ECAPn_BA = 0x401B_0000 + n*0x4000**<br>**n=0,1** | | | | |
| **ECAP_CNT** | ECAPn_BA+0x00 | R/W | 输入捕获计数器 (24-bit 上数计数器) | 0x0000_0000 |
| **ECAP_HOLD0** | ECAPn_BA+0x04 | R/W | 输入捕获计数器保持寄存器0 | 0x0000_0000 |
| **ECAP_HOLD1** | ECAPn_BA+0x08 | R/W | 输入捕获计数器保持寄存器1 | 0x0000_0000 |
| **ECAP_HOLD2** | ECAPn_BA+0x0C | R/W | 输入捕获计数器保持寄存器2 | 0x0000_0000 |
| **ECAP_CNTCMP** | ECAPn_BA+0x10 | R/W | 输入捕获计数器比较寄存器 | 0x0000_0000 |
| **ECAP_CTL0** | ECAPn_BA+0x14 | R/W | 输入捕获控制寄存器0 | 0x0000_0000 |
| **ECAP_CTL1** | ECAPn_BA+0x18 | R/W | 输入捕获控制寄存器1 | 0x0000_0000 |
| **ECAP_STATUS** | ECAPn_BA+0x1C | R/W | 输入捕获状态寄存器 | 0x0000_0000 |

### 6.9.7 寄存器描述

**输入捕获计数器 (CAP_CNT)**

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ECAP_CNT** | ECAPn_BA+0x00 | R/W | 输入捕获计数器 (24-bit 上数计数器) | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留. |
| [23:0] | **VAL** | **输入捕获定时器/计数器**<br>输入捕获定时器/计数器是一个24位上数计数器。它的时钟源来自时钟分频器的输出。可选择. 1,4,16 或 32分频 |

## 输入捕获计数器保持寄存器 (ECAP_HOLD0~2)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ECAP_HOLD0** | ECAPn_BA+0x04 | R/W | 输入捕获计数器保持寄存器0 | 0x0000_0000 |
| **ECAP_HOLD1** | ECAPn_BA+0x08 | R/W | 输入捕获计数器保持寄存器1 | 0x0000_0000 |
| **ECAP_HOLD2** | ECAPn_BA+0x0C | R/W | 输入捕获计数器保持寄存器2 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留. |
| [23:0] | **VAL** | **输入捕获计数器保持寄存器**<br>当一个有效的输入捕获通道侦测到一个有效的边沿信号改变，ECAP_CNT值被锁存到对应的保持寄存器。每一个输入通道有它自己的保持寄存器，命名为ECAP_HOLDx，其中的x从0到2分别表示ECAPx_IC0 到 ECAPx_IC2。 |

## 输入捕获计数器比较寄存器 (ECAP_CNTCMP)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ECAP_CNTCMP** | ECAPn_BA+0x10 | R/W | 输入捕获计数器比较寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **Reserved** | 保留. |
| [23:0] | **VAL** | **输入捕获计数器比较寄存器**<br>如果比较功能使能(CMPEN(ECAP_CTL0[28]) = 1),比较寄存器加载该值，用于和捕获计数器(ECAP_CNT) 比较。<br>如果重载控制使能(RLDEN = 1), 一个溢出事件或捕获事件会触发硬件重载ECAP_CNTCMP到 ECAP_CNT. |

## 输入捕获定时器/计数器控制寄存器 (ECAP_CTL0)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ECAP_CTL0** | ECAPn_BA+0x14 | R/W | 输入捕获控制寄存器0 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | CAPEN | CMPEN | RLDEN | CPTCLR | CMPCLR | CNTEN |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | CMPIEN | OVIEN | Reserved | CAPIEN2 | CAPIEN1 | CAPIEN0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | CAPSEL2 | | CAPSEL1 | | CAPSEL0 | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | CAPEN2 | CAPEN1 | CAPEN0 | CAPNF_DIS | Reserved | NFDIS | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **Reserved** | 保留. |
| [29] | **CAPEN** | **输入捕获定时器/计数器使能位**<br>0 =输入捕获功能禁止.<br>1 =输入捕获功能使能 |
| [28] | **CMPEN** | **比较功能使能位**<br>输入捕获定时器/计数器的比较功能可以在动态计数中将ECAP_CNT值与比较寄存器ECAP_CNTCMP比较 。如果ECAP_CNT值达到ECAP_CNTCMP的值，标志位CMPF位置位。<br>0 =.比较功能禁止<br>1 = 比较功能使能 |
| [27] | **RLDEN** | **重载功能使能位**<br>设置该位使能重载功能。如果重载控制使能，一个溢出事件(OVF)或捕获事件(CAPFx)会触发硬件重载ECAP_CNTCMP 到 ECAP_CNT.<br>0 =.重载功能禁止<br>1 =.重载功能使能 |
| [26] | **CPTCLR** | **输入捕获计数器通过捕获事件控制清零**<br>如果该位置1，当一个捕获事件 (CAPF0~3) 发生，捕获计数器(ECAP_CNT)会被清0。<br>0 = 捕获事件 (CAPF0~3) 清除捕获计数器 (ECAP_CNT) 禁止.<br>1 = 捕获事件(CAPF0~3) 清除捕获计数器(ECAP_CNT) 使能. |
| [25] | **CMPCLR** | **输入捕获计数器通过比较匹配控制清零**<br>如果该位置1， 当一个比较匹配事件 (CAMCMPF = 1) 发生，捕获计数器(ECAP_CNT)会被清0。<br>0 = 比较匹配事件(CAMCMPF) 清除捕获计数器(ECAP_CNT) 禁止.<br>1 =比较匹配事件(CAMCMPF) 清除捕获计数器(ECAP_CNT)使能 |

| 位 | 描述 | |
|---|---|---|
| [24] | **CNTEN** | **输入捕获计数器开始**<br>该位置1，捕获计数器(ECAP_CNT)开始根据捕获时钟(CAP_CLK).输入同步向上计数<br>0 = ECAP_CNT 停止计数<br>1 = ECAP_CNT 开始向上计数 |
| [23:22] | **Reserved** | 保留. |
| [21] | **CMPIEN** | **CMPF 触发输入捕获中断使能位**<br>0 = CMPF 触发输入捕获中断禁止.<br>1 = CMPF触发输入捕获中断使能 |
| [20] | **OVIEN** | **OVF 触发输入捕获中断使能位**<br>0 = OVUNF触发输入捕获中断禁止<br>1 = OVUNF 触发输入捕获中断使能 |
| [19] | **Reserved** | 保留. |
| [18] | **CAPIEN2** | **输入捕获通道2中断使能位**<br>0 = CAPF2 触发输入捕获中断禁止<br>1 = CAPF2触发输入捕获中断使能 |
| [17] | **CAPIEN1** | **输入捕获通道1中断使能位**<br>0 = CAPF1 触发输入捕获中断禁止<br>1 = CAPF1触发输入捕获中断使能 |
| [16] | **CAPIEN0** | **输入捕获通道0中断使能位**<br>0 = CAPF0 触发输入捕获中断禁止<br>1 = CAPF0触发输入捕获中断使能 |
| [15:14] | **Reserved** | 保留. |
| [13:12] | **CAPSEL2** | **CAP2 输入源选择**<br>00 = CAP2 输入来自引脚 RCAPx_IC2.<br>01 = CAP2 输入来自 CO2 (ACMPSR[10]).<br>10 = 保留.<br>10 = 保留.<br>11 = CAP2输入来自信号ADCMPOx (ADC 比较输出 x).<br>**注意:** 输入捕获单元n匹配比较器单元x，这里的x = 0~1。. |
| [11:10] | **CAPSEL1** | **CAP1 输入源选择**<br>00 = CAP1 输入来自引脚 RCAPx_IC1.<br>01 = CAP1 输入来自 CO1 (ACMPSR[9]).<br>10 = 保留.<br>10 = 保留.<br>11 = CAP1输入来自信号OPDO1 (OPASR[1]).<br>**注意:** 输入捕获单元n匹配比较器单元x，这里的x = 0~1。. |

| 位 | 描述 | |
|---|---|---|
| [9:8] | **CAPSEL0** | **CAP0 输入源选择**<br>00 = CAP0 输入来自引脚 RCAPx_IC0.<br>01 = CAP0 输入来自 CO0 (ACMPSR[8]).<br>10 = 保留.<br>10 = 保留.<br>11 = CAP0输入来自信号OPDO0 (OPASR[0]).<br>**注意:** 输入捕获单元n匹配比较器单元x，这里的x = 0~1。. |
| [7] | **Reserved** | 保留. |
| [6] | **CAPEN2** | **引脚 ECAPx_IC2 输入到输入捕获单元使能位**<br>0 = ECAPx_IC2 输入到输入捕获单元禁止<br>1 = ECAPx_IC2输入到输入捕获单元使能 |
| [5] | **CAPEN1** | **引脚 ECAPx_IC1 输入到输入捕获单元使能位**<br>0 = ECAPx_IC1输入到输入捕获单元禁止<br>1 = ECAPx_IC1输入到输入捕获单元使能 |
| [4] | **CAPEN0** | **引脚 ECAPx_IC0 输入到输入捕获单元使能位**<br>0 = ECAPx_IC0 输入到输入捕获单元禁止<br>1 = ECAPx_IC0输入到输入捕获单元使能 |
| [3] | **CAPNF_DIS** | **输入捕获噪声过滤器禁止位**<br>0 = 输入捕获的噪音过滤器使能.<br>1 = 输入捕获的噪音过滤器禁止. |
| [2] | **Reserved** | 保留. |
| [1:0] | **NFDIS** | **噪音过滤器时钟预分频选择**<br>分频决定了噪音过滤器时钟的采样频率<br>00 = CAP_CLK.<br>01 = CAP_CLK/2.<br>10 = CAP_CLK/4.<br>11 = CAP_CLK/16. |

## 输入捕获定时器/计数器控制寄存器 (ECAP_CTL1)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ECAP_CTL1** | ECAPn_BA+0x18 | R/W | 输入捕获控制寄存器1 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | **SRCSEL** | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | **CLKSEL** | | | **Reserved** | **RLDSEL** | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | **EDGESEL2** | | **EDGESEL1** | | **EDGESEL0** | |

| 位 | 描述 | |
|---|---|---|
| [31:18] | **Reserved** | 保留. |
| [17:16] | **SRCSEL** | **捕获定时器/计数器时钟源选择**<br>选择捕获定时器/计数器时钟源<br>00 = CAP_CLK (默认).<br>01 = CAP0.<br>10 = CAP1.<br>11 = CAP2. |
| [15] | **Reserved** | 保留. |
| [14:12] | **CLKSEL** | **捕获定时器时钟分频选择**<br>捕获定时器时钟有一个预分频器，有下面8个分频选项。<br>000 = CAP_CLK/1.<br>001 = CAP_CLK/4.<br>010 = CAP_CLK/16.<br>011 = CAP_CLK/32.<br>100 = CAP_CLK/64.<br>101 = CAP_CLK/96.<br>110 = CAP_CLK/112.<br>111 = CAP_CLK/128. |
| [11] | **Reserved** | 保留. |

| 位 | 描述 | |
|---|---|---|
| [10:8] | **RLDSEL** | **ECAP_CNT 重载触发源选择**<br>如果RLDEN (ECAP_CTL0[27]) = 1重载功能使能，当一个重载触发事件到来，ECAP_CNT会重载ECAP_CNTCMP的值。<br>RLDSEL[2:0] 决定了 ECAP_CNT 重载触发源<br>000 = CAPF0.<br>001 = CAPF1.<br>010 = CAPF2.<br>100 = OVF.<br>其他 = 保留. |
| [7:6] | **Reserved** | 保留. |
| [5:4] | **EDGESEL2** | **通道 2 捕获边沿选择**<br>输入捕获可以只侦测下降沿或上升沿，或上升沿和下降沿都侦测。<br>00 = 侦测上升沿.<br>01 = 侦测下降沿.<br>1x = 上升沿和下降沿都侦测 |
| [3:2] | **EDGESEL1** | **通道 1 捕获边沿选择**<br>输入捕获可以只侦测下降沿或上升沿，或上升沿和下降沿都侦测。<br>00 = 侦测上升沿.<br>01 = 侦测下降沿.<br>1x = 上升沿和下降沿都侦测 |
| [1:0] | **EDGESEL0** | **通道 0 捕获边沿选择**<br>输入捕获可以只侦测下降沿或上升沿，或上升沿和下降沿都侦测。<br>00 = 侦测上升沿.<br>01 = 侦测下降沿.<br>1x = 上升沿和下降沿都侦测 |

## 输入捕获定时器/计数器状态寄存器 (ECAP_STATUS)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ECAP_STATUS** | ECAPn_BA+0x1C | R/W | 输入捕获状态寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | **OVF** | **CMPF** | **Reserved** | **CAPF2** | **CAPF1** | **CAPF0** |

| 位 | 描述 | |
|---|---|---|
| [31:6] | **Reserved** | 保留. |
| [5] | **OVF** | **输入捕获计数器溢出标志**<br>标志由硬件置位，当输入捕获上数计数器(ECAP_CNT)从0x00FF_FFFF溢出到0<br>0 = ECAP_CNT.没有溢出事件发生<br>1 = ECAP_CNT 溢出.<br>**注意:** 该位仅通过写 1 清零 |
| [4] | **CMPF** | **输入捕获比较匹配标志**<br>如果输入捕获比较功能使能，当捕获计数器(ECAP_CNT)上数且到达ECAP_CNTCMP的值，标志由硬件置位。<br>0 = ECAP_CNT 没有和ECAP_CNTCMP 值匹配.<br>1 = ECAP_CNT 数到和 ECAP_CNTCMP 值相同.<br>**注意:** 该位仅通过写 1 清零 |
| [3] | **Reserved** | 保留. |
| [2] | **CAPF2** | **输入捕获通道2捕获标志**<br>当输入捕获通道2侦测到CAP2输入有一个有效的跳变沿，会置位CAPF2为高。<br>0 = 在 CAP2 输入没有有效的跳变沿.<br>1 = 在 CAP2 输入侦测到有效的跳变沿.<br>**注意:** 该位仅通过写 1 清零 |
| [1] | **CAPF1** | **输入捕获通道1捕获标志**<br>当输入捕获通道1侦测到CAP1输入有一个有效的跳变沿，会置位CAPF1为高。<br>0 = 在 CAP1输入没有有效的跳变沿.<br>1 = 在 CAP1 输入侦测到有效的跳变沿.<br>**注意:** 该位仅通过写 1 清零 |

| 位 | 描述 | |
|---|---|---|
| [0] | **CAPF0** | **输入捕获通道0捕获标志**<br>当输入捕获通道0侦测到CAP0输入有一个有效的跳变沿，会置位CAPF0为高。<br>0 = 在 CAP0输入没有有效的跳变沿.<br>1 = 在 CAP0输入侦测到有效的跳变沿.<br>**注意:** 该位仅通过写 1 清零 |

## 6.10 看门狗定时器 (WDT)

### 6.10.1 概述

设计看门狗定时器的目的是，当系统运行到一个未知状态时，通过它来使系统复位。这种做法可以预防系统进入到无限期的死循环。此外，该看门狗定时器支持系统从Idle/Power-down模式唤醒功能。

### 6.10.2 特性

- 18 位的看门狗定时器可满足用户溢出时间间隔要求
- 可选的超时时间间隔 (24 ~ 218) WDT_CLK时钟周期，如WDT_CLK = 10 kHz，那么溢出时间间隔是1.6 ms ~ 26.214 s
- 系统复位保持时间(1 / WDT_CLK) * 63
- 可选的复位延时周期包括(1026、130、18 或 3) * WDT_CLK个复位延时周期
- 当CWDTEN (CONFIG0[31] 看门狗使能)位被置为0，支持芯片上电或复位条件下看门狗强制打开。
- 支持看门狗定时器溢出唤醒功能，此时时钟源必须选择内部低速 10k 时钟源

### 6.10.3 框图

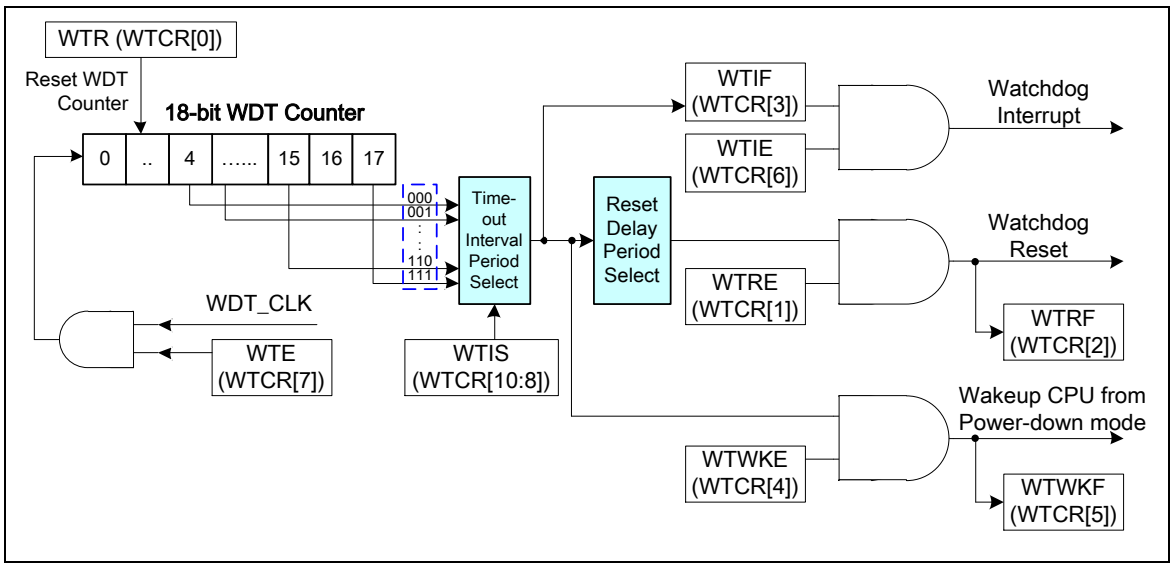



图 6-65 看门狗定时器框图

**注意1:** WDT 复位 CPU 将保持 63个 WDT_CLK.

**注意2:** 如果时钟源选择内部10K，看门狗超时中断信号产生可以唤醒芯片。

**注意3:** WDT 复位延时周期可以选择为3/18/130/1026 个WDT_CLK.

### 6.10.4 时钟控制

WDT 时钟控制如下 图 6-66.

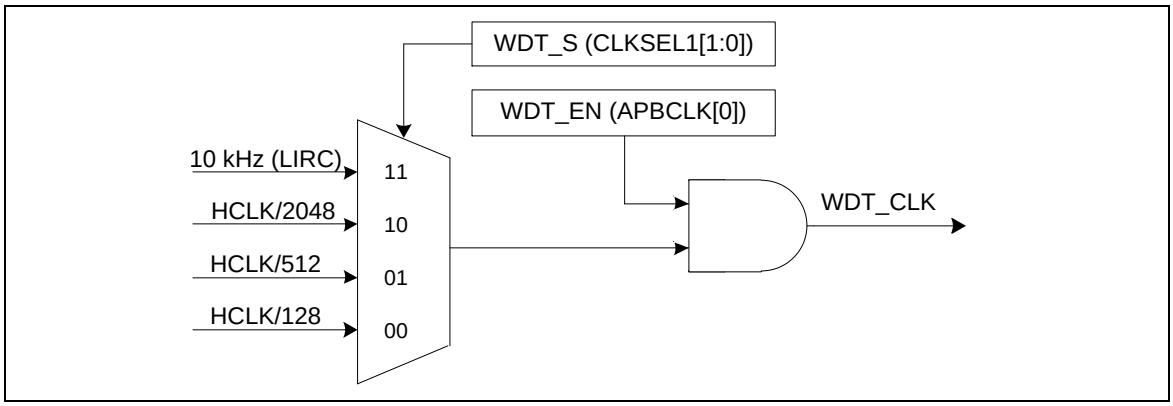



图 6-66 看门狗定时器时钟

### 6.10.5 基本配置

看门狗定时器的时钟在WDT_EN (APBCLK[0])位使能，时钟源选择在WDT_S (CLKSEL1[1:0])设置。或者用户也可以把CWDTEN (Config0[31])配置为0，从而在芯片上电或RESET后强制使能看门狗定时器，并使看门狗定时器以10k时钟为时钟源工作。

### 6.10.6 功能描述

看门狗定时器(WDT)包含一个18位的可设置溢出时间间隔的向上计数器。表 6-11显示了WDT溢出时间间隔选择， 图 6-67显示了WDT溢出时间间隔和复位周期时序。

#### *6.10.6.1 WDT 超时中断*

设置 WTE (WTCR[7]) 为 1 使能WDT功能。然后WDT计数器向上计数。通过设置WTIS (WTCR[10:8])，可以选择8个不同的溢出时间间隔。当WDT计数器计到WTIS (WTCR[10:8])设定值，WDT产生溢出中断，然后WTIF (WTCR[3])标志立即被置1.

### 6.10.6.2 WDT 复位延时周期和复位系统

WTIF标志被置位后，还会有一个固定的延时周期TRSTD，用户应在TRSTD延时计完之前对WTR置1来复位18位的WDT向上计数器的值，以防止产生WDT时间溢出复位信号。如TRSTD时间计满以后，WDT向上计数器的值仍没有被清除，若WTRE位为1，则WTRF标志会被置1，然后芯片立即复位。请参考图 6-67,看门狗定时器时间溢出间隔和复位周期时序，TRST复位周期将保持至少63个WDT时钟周期，然后芯片从复位向量地址(0x0000_0000)重新开始执行程序。芯片由WDT定时溢出复位后，WTRF标志将会保持1。用户可以通过检查该标志来确认系统是否因WDT定时溢出复位。

| WTIS | 超时间隔周期 $T_{TIS}$ | 复位延时周期 $T_{RSTD}$ |
|---|---|---|
| 000 | $2^4 * T_{WDT}$ | (3/18/130/1026) * $T_{WDT}$ |
| 001 | $2^6 * T_{WDT}$ | (3/18/130/1026) * $T_{WDT}$ |
| 010 | $2^8 * T_{WDT}$ | (3/18/130/1026) * $T_{WDT}$ |
| 011 | $2^{10} * T_{WDT}$ | (3/18/130/1026) * $T_{WDT}$ |
| 100 | $2^{12} * T_{WDT}$ | (3/18/130/1026) * $T_{WDT}$ |
| 101 | $2^{14} * T_{WDT}$ | (3/18/130/1026) * $T_{WDT}$ |
| 110 | $2^{16} * T_{WDT}$ | (3/18/130/1026) * $T_{WDT}$ |
| 111 | $2^{18} * T_{WDT}$ | (3/18/130/1026) * $T_{WDT}$ |

表 6-11 看门狗定时器超时间隔周期选择

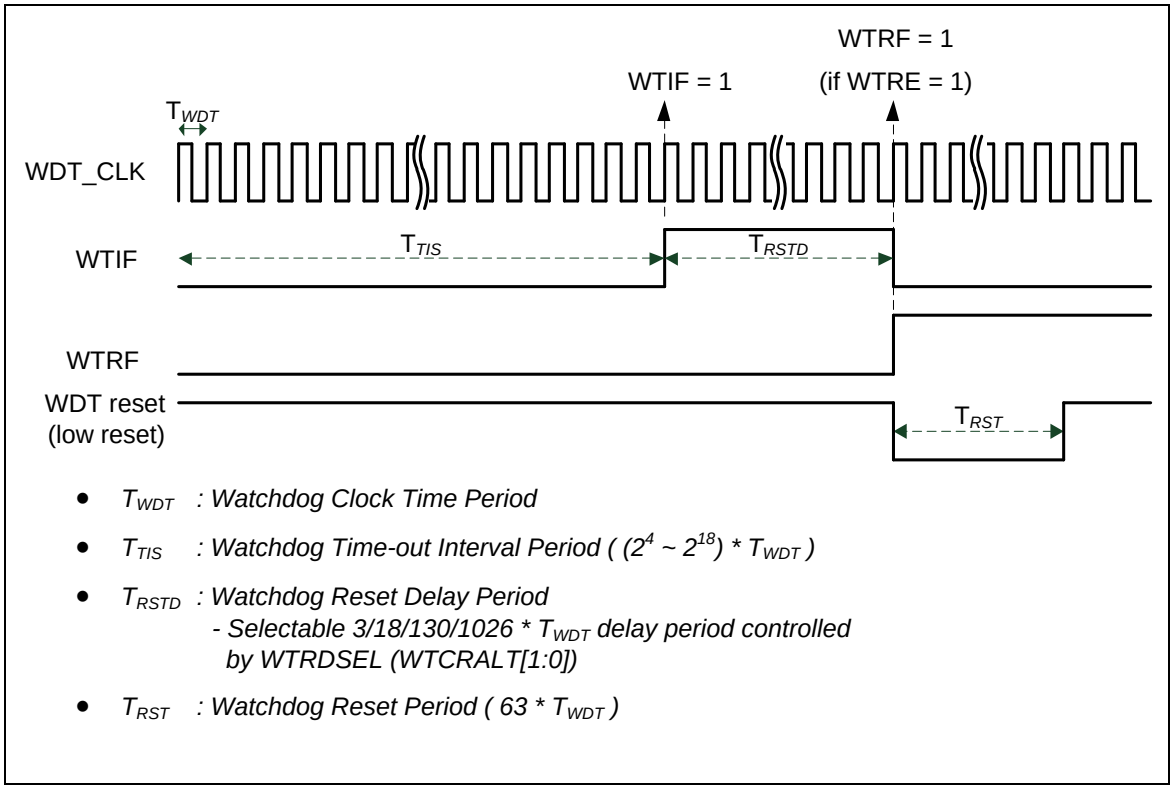



图 6-67 看门狗定时器超时间隔和复位周期时序

### *6.10.6.3 WDT 唤醒*

如果看门狗时钟源选择为内部10k，如果WTWKE (WTCR[4])位被使能，看门狗定时溢出中断产生后，系统能从Power-down模式下被唤醒。同时，WTWKF标志会被自动置1。用户可以通过查询WTWKF标志知道系统是否是被看门狗定时溢出中断唤醒。

### 6.10.7 寄存器映射

**R**: 只读, **W**: 只写, **R/W**: 可读写

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **WDT 基地址:**<br>**WDT_BA = 0x4000_4000** | | | | |
| **WTCR** | WDT_BA+0x00 | R/W | WDT 控制寄存器 | 0x0000_0700 |
| **WTCRALT** | WDT_BA+0x04 | R/W | WDT 选择控制寄存器 | 0x0000_0000 |

### 6.10.8 寄存器描述

**WDT 控制寄存器 (WTCR)**

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **WTCR** | WDT_BA+0x00 | R/W | 看门狗定时器控制寄存器 | 0x0000_0700 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **DBGACK_WDT** | **Reserved** | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | **WTIS** | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **WTE** | **WTIE** | **WTWKF** | **WTWKE** | **WTIF** | **WTRF** | **WTRE** | **WTR** |

| 位 | 描述 | |
|---|---|---|
| [31] | **DBGACK_WDT** | **ICE调试模式下看门狗计数控制位(写保护)**<br>0 = ICE 调试模式影响看门狗计数，当ICE让CPU运行停止，看门狗计数器也停止.<br>1 = ICE 调试模式不影响看门狗计数。不论CPU受ICE停止还是运行，看门狗继续计数<br>**注意:**. 该位是写保护位，请参考REGWRPROT 寄存器. |
| [30:11] | **Reserved** | 保留. |
| [10:8] | **WTIS** | **WDT 超时间隔选择 (写保护)**<br>以下三个位用于选择看门狗定时溢出周期<br>000 = $2^4$ * WDT_CLK.<br>001 = $2^6$ * WDT_CLK.<br>010 = $2^8$ * WDT_CLK.<br>011 = $2^{10}$ * WDT_CLK.<br>100 = $2^{12}$ * WDT_CLK.<br>101 = $2^{14}$ * WDT_CLK.<br>110 = $2^{16}$ * WDT_CLK.<br>111 = $2^{18}$ * WDT_CLK.<br>**注意:**. 该位是写保护位，请参考REGWRPROT 寄存器. |
| [7] | **WTE** | **WDT 使能位 (写保护)**<br>0 = WDT 禁止 (该操作会恢复看门狗计数器的值.)<br>1 = WDT 使能.<br>**注意1:**. 该位是写保护位，请参考REGWRPROT 寄存器.<br>**注意2:** 如果 CWDTEN (Config0[31])位配置为0，该位会强制为1，用户不需要改变该位为0。 |
| [6] | **WTIE** | **WDT 超时中断使能位 (写保护)** |

| | | |
|---|---|---|
| | | CPU. 如果该位使能，看门狗超时中断信号会产生，并通知CPU。<br>0 = WDT 超时中断禁止<br>1 = WDT 超时中断使能<br>**注意:**. 该位是写保护位，请参考REGWRPROT 寄存器. |
| [5] | **WTWKF** | **WDT 超时唤醒标志 (写保护)**<br>该位表明看门狗定时器中断唤醒标志的状态<br>0 =看门狗定时器没有导致芯片唤醒.<br>1 =芯片从Idle或Power-down 模式被WDT超时中断信号唤醒<br>**注意1:**. 该位是写保护位，请参考REGWRPROT 寄存器<br>**注意2:** 该位写1清零 |
| [4] | **WTWKE** | **WDT 超时唤醒功能控制 (写保护)**<br>如果此位被置1, 当WTIF被置1且WTIE被使能，看门狗定时溢出中断信号将会触发唤醒MCU.<br>0 =唤醒触发事件禁止，即便看门狗定时器定时溢出中断发生<br>1 =唤醒触发事件使能，如果看门狗定时溢出中断产生，将唤醒MCU<br>**注意1:**. 该位是写保护位，请参考REGWRPROT 寄存器..<br>**注意2:** MCU能被看门狗定时器定时溢出中断信号唤醒的条件是看门狗定时器时钟源必须为内部10k时钟 |
| [3] | **WTIF** | **WDT 超时中断标志**<br>当看门狗定时器计数器的值达到溢出时间间隔，该位将被置1<br>0 = 没有发生看门狗定时溢出中断<br>1 = 有看门狗定时溢出中断发生.<br>**注意:** 该位写1清零 |
| [2] | **WTRF** | **看门狗定时器复位标志**<br>该位用来指示系统是否因看门狗定时器定时溢出发生复位<br>0 = 没有发生看门狗定时溢出复位<br>1 = 发生了看门狗定时溢出复位.<br>**注意**: 该位写1清零. |
| [1] | **WTRE** | **WDT 超时复位使能位 (写保护)**<br>设置该位会使能WDT超时复位功能，如果WDT上数计数器的值在指定的WDT复位延时周期之后还没有清零，会复位芯片。<br>0 = WDT 超时复位功能禁止<br>1 = WDT 超时复位功能使能<br>**注意:**. 该位是写保护位，请参考REGWRPROT 寄存器. |
| [0] | **WTR** | **复位 WDT 上数计数器 (写保护)**<br>0 = 没影响.<br>1 = 复位内部 18-bit WDT 上数计数器的值<br>**注意1:**. 该位是写保护位，请参考REGWRPROT 寄存器<br>**注意2:** 该位由硬件自动清零 |

## WDT 选择控制寄存器 (WTCRALT)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **WTCRALT** | WDT_BA+0x04 | R/W | WDT 选择控制寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:2] | **Reserved** | 保留. |
| [1:0] | **WTRDSEL** | **看门狗定时器复位延时选择 (写保护)**<br>当看门狗定时器发生溢出, 在看门狗复位延时的时段内可以将看门狗计数器清零，以防止看门狗溢出复位. 对不同的看门狗溢出时间，用户也可以选择适当的看门狗复位延时时间<br>00 = 看门狗定时器复位延时时间是1026 * WDT_CLK.<br>01 =看门狗定时器复位延时时间是130 * WDT_CLK.<br>10 =看门狗定时器复位延时时间是18 * WDT_CLK.<br>11 =看门狗定时器复位延时时间是3 * WDT_CLK.<br>**注意1:**. 该位是写保护位，请参考REGWRPROT 寄存器<br>**注意2:** 如果看门狗定时器超时复位发生，该寄存器的值将被清零 |

## 6.11 窗口看门狗定时器 (WWDT)

### 6.11.1 概述

窗口看门狗定时器用于在一个窗口时间内执行系统复位，以防止程序在不可预知条件下跑到一个不可控的状态。 通常是外部的干扰或意外的逻辑条件导致。

窗口功能被用来调整看门狗的行为动作，以完美地匹配应用。软件必须在超时之前重置计数器。

### 6.11.2 特性

- 6-bit 下数计数器值 WWDTCVAL (WWDTCVR[5:0]) 和 6-bit 比较值 WINCMP (WWDTCR[21:16]) 来使WWDT 超时窗口周期灵活可调。
- 支持4-bit 值 PERIODSEL (WWDTCR[11:8]) 来选择预分频，预分频计数器最大可达11位。
- 在Idle/Power-down 模式，WWDT 计数器停止。

### 6.11.3 框图

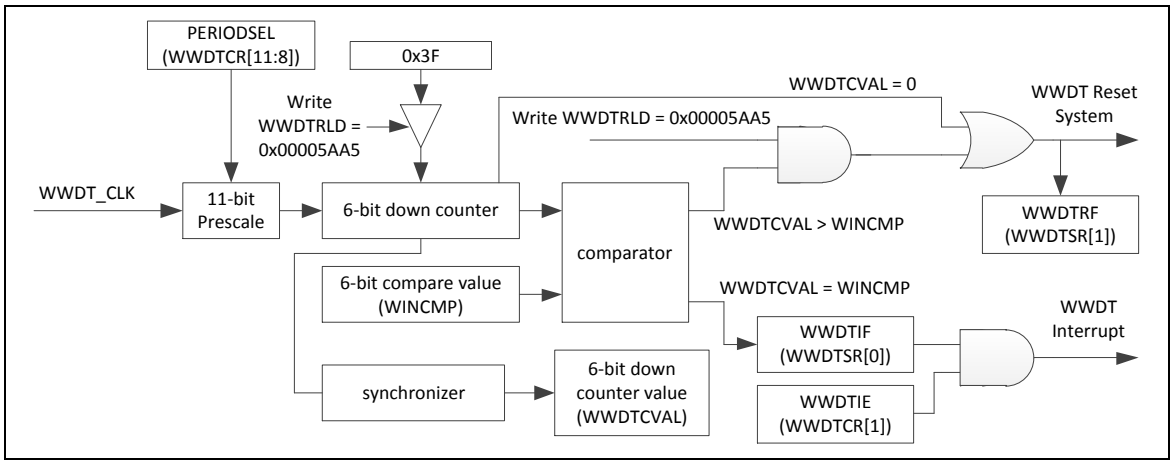



图 6-68 窗口看门狗定时器框图

### 6.11.4 时钟框图

窗口看门狗定时器框图如图 6-69.所示

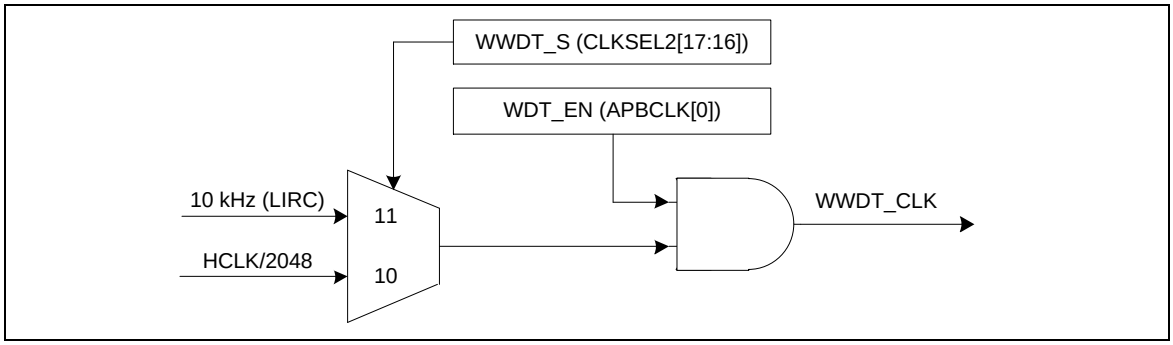



图 6-69 窗口看门狗定时器时钟框图

### 6.11.5 基本配置

WWDT 外设时钟在WDT_EN (APBCLK[0])使能，时钟源在 WWDT_S (CLKSEL2[17:16])选择。

### 6.11.6 功能描述

窗口看门狗定时器(WWDT)是一个 6位向下计数器，该计数器带一个可选择预分频值，不同的预分频值对应不同的看门狗定时溢出时间。 6位窗口看门狗定时器的时钟源可以是系统时钟经2048 (HCLK/2048)分频的时钟，也可以是内部10k低速RC振荡时钟源，看门狗的时钟源带一个可选择的11位预分频值，该值可通过PERIODSEL (WWDTCRL[11:8]) 位来设置选择，对应预分频值如下表 6-12.

| PERIODSEL | 分频值 | 超时周期 | 超时间隔 (WWDT_CLK=10 KHz) |
|---|---|---|---|
| 0000 | 1 | 1 * 64 * $T_{WWDT}$ | 6.4 ms |
| 0001 | 2 | 2 * 64 * $T_{WWDT}$ | 12.8 ms |
| 0010 | 4 | 4 * 64 * $T_{WWDT}$ | 25.6 ms |
| 0011 | 8 | 8 * 64 * $T_{WWDT}$ | 51.2 ms |
| 0100 | 16 | 16 * 64 * $T_{WWDT}$ | 102.4 ms |
| 0101 | 32 | 32 * 64 * $T_{WWDT}$ | 204.8 ms |
| 0110 | 64 | 64 * 64 * $T_{WWDT}$ | 409.6 ms |
| 0111 | 128 | 128 * 64 * $T_{WWDT}$ | 819.2 ms |
| 1000 | 192 | 192 * 64 * $T_{WWDT}$ | 1.2288 s |
| 1001 | 256 | 256 * 64 * $T_{WWDT}$ | 1.6384 s |
| 1010 | 384 | 384 * 64 * $T_{WWDT}$ | 2.4576 s |
| 1011 | 512 | 512 * 64 * $T_{WWDT}$ | 3.2768 s |
| 1100 | 768 | 768 * 64 * $T_{WWDT}$ | 4.9152 s |
| 1101 | 1024 | 1024 * 64 * $T_{WWDT}$ | 6.5536 s |
| 1110 | 1536 | 1536 * 64 * $T_{WWDT}$ | 9.8304 s |

| | | | |
|---|---|---|---|
| 1111 | 2048 | 2048 * 64 * $T_{WWDT}$ | 13.1072 s |

表 6-12 WWDT 分频值选择

#### 6.11.6.1 *WWDT 计数*

当WWDTEN位被使能，窗口看门狗向下计数器将会从0x3F向下递减计数到0. 为了防止程序在非用户指定位置关闭窗口看门狗定时器，窗口看门狗定时器控制寄存器WWDTCR在芯片上电或复位后仅可写一次。当WWDTEN (WWDTCR[0])位被软件使能以后，用户不能禁止窗口看门狗定时器(WWDTEN), 不能修改计数器预分频周期(PERIODSEL)或窗口比较值 (WINCMP)，除非芯片复位。

#### 6.11.6.2 *WWDT 比较匹配中断*

窗口看门狗定时器向下计数过程中，当窗口看门狗定时器计数值(WWDTCVAL) 等于比较值WINCMP， WWDTIF会被置1，并且WWDTIF 可以被软件清零;如果WWDTIE位被使能，当WWDTIF 位被硬件置1， 就会产生窗口看门狗比较匹配中断。

#### 6.11.6.3 *WWDT 复位系统*

当WWDTIF产生后，用户必须通过对WWDTRLD写0x00005AA5进行重载回到初始值0x3F，同时内部预分频计数器值也变为0。这可以防止窗口看门狗定时器的值继续向下计数到0，从而产生窗口看门狗定时器复位系统信号通知系统复位。

如果寄存器WWDTCVAL 的当前值大于比较寄存器WINCMP的值， 用户对WWDTRLD寄存器写0x00005AA5，窗口看门狗定时器复位系统信号将立刻产生，并导致芯片复位。

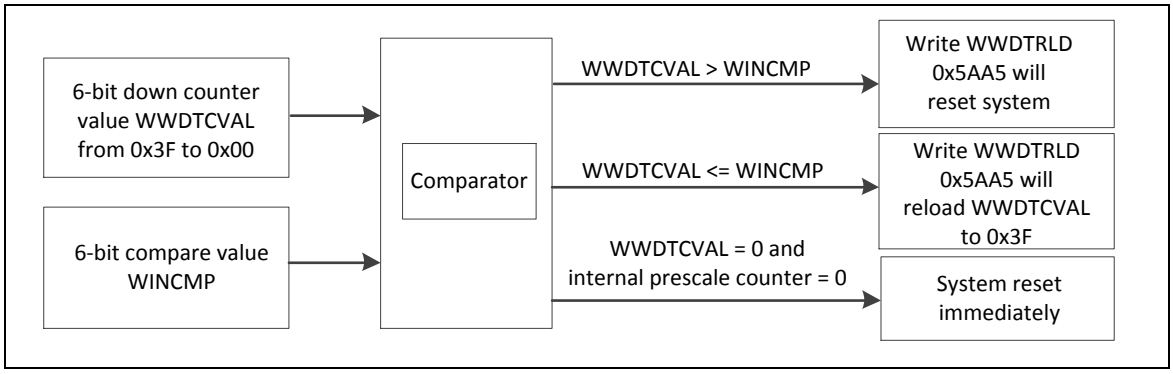



图 6-70 WWDT 复位和重载行为

#### 6.11.6.4 WWDT 窗口设置限制

当用户对WWDTRLD寄存器写0x00005AA5来重载WWDT的值到0x3F的时候，大概需要3个窗口看门狗定时器时钟来同步重载命令到实际执行重载操作。这意味着如果用户设置PERIODSEL的值为0000，即计数器预分频值设置为1，WINCMP寄存器的值必须大于2；否则，当WWDTIF标志产生后，写WWDTRLD来重载窗口看门狗定时器的值为0x3F操作无效，将会一直出现窗口看门狗定时器引起的系统复位。

| PERIODSEL | 预分频值 | 有效的 WINCMP 值 |
|---|---|---|
| 0000 | 1 | 0x3 ~ 0x3F |
| 0001 | 2 | 0x2 ~ 0x3F |
| 其他 | 其他 | 0x0 ~ 0x3F |

表 6-13 WWDT WINCMP 的设置限制

### 6.11.7 寄存器映射

**R**: 只读, **W**: 只写, **R/W**: 可读写

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **WWDT 基地址:**<br>**WWDT_BA = 0x4000_4100** | | | | |
| **WWDTRLD** | WWDT_BA+0x00 | W | WWDT 重载计数器寄存器 | 0x0000_0000 |
| **WWDTCR** | WWDT_BA+0x04 | R/W | WWDT 控制寄存器 | 0x003F_0800 |
| **WWDTSR** | WWDT_BA+0x08 | R/W | WWDT 状态寄存器 | 0x0000_0000 |
| **WWDTCVR** | WWDT_BA+0x0C | R | WWDT 计数器值寄存器 | 0x0000_003F |

### 6.11.8 寄存器描述

**WWDT 重载计数器寄存器 (WWDTRLD)**

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **WWDTRLD** | WWDT_BA+0x00 | W | WWDT 重载计数器寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **WWDTRLD** | **WWDT 重载计数器寄存器**<br>对此寄存器写0x00005AA5 将会重载窗口看门狗定时器计数器的值为0x3F.<br>**注意:**用户只能当窗口看门狗计数器的值在0到WINCMP之间才可以写WWDTRLD寄存器来重载窗口看门狗计数器。如果窗口看门狗计数器的值大于WINCMP时写WWDTRLD寄存器，窗口看门狗定时器复位信号仍会立即产生。 |

## WWDT 控制寄存器 (WWDTCR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **WWDTCR** | WWDT_BA+0x04 | R/W | WWDT 控制寄存器 | 0x003F_0800 |

**注意:** 此寄存器仅当芯片上电或复位后写一次

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **DBGACK_WWDT** | **Reserved** | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | **WINCMP** | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | **PERIODSEL** | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | **WWDTIE** | **WWDTEN** |

| 位 | 描述 | |
|---|---|---|
| [31] | **DBGACK_WWDT** | **ICE 调试模式下窗口看门狗计数控制位**<br>0 =ICE 调试模式下影响窗口看门狗定时器计数，当CPU被ICE暂停后，窗口看门狗向下计数器计数值将保持不变<br>1 = ICE 调试模式下不影响窗口看门狗定时器计数，窗口看门狗定时器向下计数器将会保持继续计数，不管CPU是否被ICE暂停。 |
| [30:22] | **Reserved** | 保留. |
| [21:16] | **WINCMP** | **窗口看门狗定时器窗口比较寄存器**<br>设置此寄存器来调整有效重载窗口<br>**注意**：当前窗口看门狗定时器计数器的值在0到WINCMP之间时，用户才可以写WWDTRLD来重载窗口看门狗定时器计数器的值。如果当前窗口看门狗定时器计数器的值大于WINCMP时用户写WWDTRLD寄存器，窗口看门狗定时器复位信号仍会立即产生。 |
| [15:12] | **Reserved** | 保留. |
| [11:8] | **PERIODSEL** | **窗口看门狗定时器计数器预分频周期选择**<br>0000 =预分频为1; 最大定时溢出周期是1 * 64 * WWDT_CLK.<br>0001 =预分频为2; 最大定时溢出周期是2 * 64 * WWDT_CLK.<br>0010 =预分频为4; 最大定时溢出周期是4 * 64 * WWDT_CLK.<br>0011 =预分频为8; 最大定时溢出周期是8 * 64 * WWDT_CLK.<br>0100 =预分频为16; 最大定时溢出周期是16 * 64 * WWDT_CLK.<br>0101 =预分频为32, 最大定时溢出周期是32 * 64 * WWDT_CLK.<br>0110 =预分频为64; 最大定时溢出周期是64 * 64 * WWDT_CLK.<br>0111 =预分频为128; 最大定时溢出周期是128 * 64 * WWDT_CLK.<br>1000 =预分频为192; 最大定时溢出周期是192 * 64 * WWDT_CLK.<br>1001 =预分频为256; 最大定时溢出周期是256 * 64 * WWDT_CLK.<br>1010 =预分频为384; 最大定时溢出周期是384 * 64 * WWDT_CLK.<br>1011 =预分频为512; 最大定时溢出周期是512 * 64 * WWDT_CLK. |

| | | |
|---|---|---|
| | | 1100 =预分频为768; 最大定时溢出周期是768 * 64 * WWDT_CLK.<br>1101 =预分频为1024; 最大定时溢出周期是1024 * 64 * WWDT_CLK.<br>1110 =预分频为1536; 最大定时溢出周期是1536 * 64 * WWDT_CLK.<br>1111 =预分频为2048; 最大定时溢出周期是2048 * 64 * WWDT_CLK. |
| [7:2] | **Reserved** | 保留. |
| [1] | **WWDTIE** | 窗口看门狗定时器中断使能位<br>如果该位被使能，如果有窗口看门狗定时器计数器比较匹配中断信号产生，就会通知CPU<br>0 =窗口看门狗定时器计数器比较匹配中断禁止<br>1 =窗口看门狗定时器计数器比较匹配中断使能 |
| [0] | **WWDTEN** | 窗口看门狗定时器使能位<br>设置该位来使能窗口看门狗定时器计数器计数<br>0 =窗口看门狗定时器计数器停止.<br>1 =窗口看门狗定时器计数器计数开始 |

## WWDT 状态 寄存器 (WWDTSR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **WWDTSR** | WWDT_BA+0x08 | R/W | WWDT 状态寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:2] | **Reserved** | 保留. |
| [1] | **WWDTRF** | **窗口看门狗定时器溢出复位标志**<br>该位指示系统是否已被窗口看门狗定时器溢出复位<br>0 =窗口看门狗定时器没有发生复位<br>1 =窗口看门狗定时器发生了复位<br>**注意**：该位是写1清零 |
| [0] | **WWDTIF** | **窗口看门狗定时器比较匹配中断标志**<br>该位指示窗口看门狗定时器的比较匹配中断状态标志<br>0 =窗口看门狗定时器计数器的值与WINCMP寄存器的值不匹配<br>1 =窗口看门狗定时器计数器的值与WINCMP寄存器的值匹配<br>**注意**：该位是写1清零 |

## WWDT 计数器值寄存器 (WWDTCVR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **WWDTCVR** | WWDT_BA+0x0C | R | WWDT 计数器值寄存器 | 0x0000_003F |

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

| 位 | 描述 | |
|---|---|---|
| [31:6] | **Reserved** | 保留. |
| [5:0] | **WWDTCVAL** | **WWDT 计数器值**<br>WWDTCVAL 将会被持续更新，以指示6位向下计数器的值 |

## 6.12 通用异步收发器 (UART)

### 6.12.1 概述

NuMicro M0519系列提供2个通用异步串行接口(UART)。UART控制器支持正常速度的串口和支持流控功能。UART控制器的接收过程是把外设的串行数据转为并行数据，发送过程是把CPU的并行数据转成串行数据发送出去。UART控制器还支持IrDA SIR, RS-485 和 LIN功能。

### 6.12.2 特性

- 全双工，异步通信
- 独立接收/发送16 字节FIFO，用于数据装载。
- 支持硬件自动流控制/流控制功能 (nCTS, nRTS) 和可编程的nRTS流控制触发电平
- 可编程的接收缓冲触发阀值
- 每个通道独立的可编程的波特率发生器
- 支持nCTS唤醒功能
- 支持 8-位接收缓存定时溢出检测功能
- 在上一次的停止位与下一次的开始位之间通过设置寄存器DLY (UA_TOR [15:8])可编程发送数据延迟时间
- 支持 break error, frame error, parity error 和 receive/transmit 缓存溢出检测功能。
- 完全可编程的串口特性
  - 可编程为 5-, 6-, 7-, 8-位的数据位
  - 可编程的奇偶校验位, even, odd, no parity 或 stick parity bit 产生和侦测
  - 可编程为 1, 1.5, 或 2 位的停止位
- 支持 IrDA SIR 功能模式
  - 普通模式下支持 3-/16-位时间
- 支持 LIN 功能模式
  - 支持 LIN 主机/从机模式
  - 发送器支持可编程的 break 产生功能
  - 接收器支持 break 侦测功能
- 支持 RS-485 功能模式
  - 支持 RS-485 9-位模式
  - 由nRTS管脚提供支持硬件或软件直接使能控制

### 6.12.3 框图

UART 时钟控制和框图分别如图 6-71 和 图 6-72 所示.

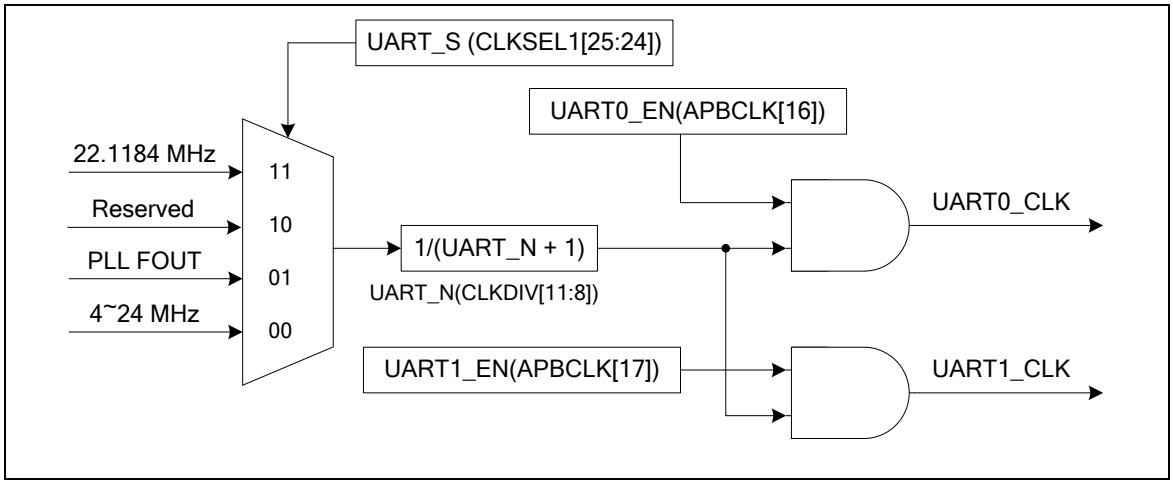



图 6-71 UART 时钟控制框图

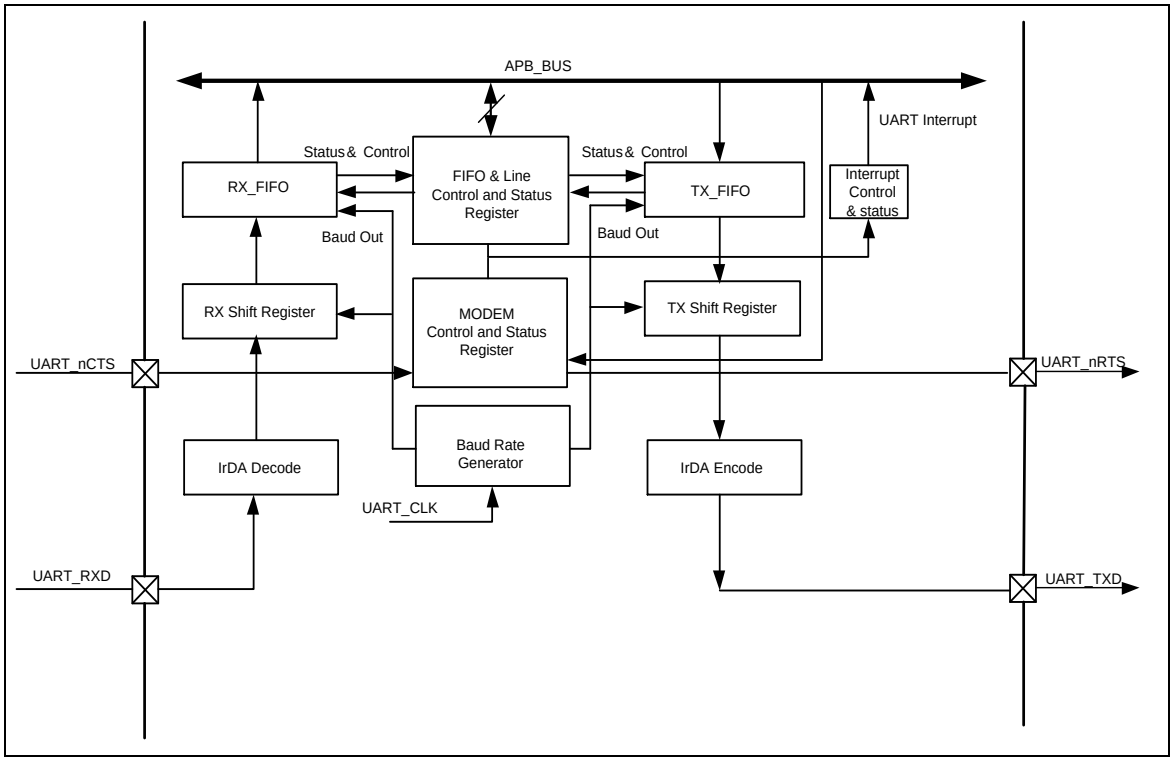



图 6-72 UART 框图

**TX_FIFO**
发送器用一个16 字节的 FIFO 缓存来降低 CPU 的中断数量。

**RX_FIFO**
接收器用一个16 字节的 FIFO （每个字节加 3 个比特的纠错位BIF (UA_FSR[6]), FEF (UA_FSR[5]), PEF (UA_FSR[4])）缓存来降低 CPU 的中断数量

**TX 移位寄存器**
移位发送数据到数据串行输出控制模块

**RX 移位寄存器**
控制将接收到的数据移入串行移位模块

**Modem 控制寄存器**
这个寄存器控制 到 MODEM 的接口或数据集（一个模拟 MODEM 外设）

**波特率发生器**
将外部时钟除以一个除数来产生期望的波特率时钟。参考波特率公式。

**IrDA 编码**
IrDA 编码控制模块。

**IrDA 解码**
IrDA 解码控制模块

**FIFO & 线 控制和状态寄存器**

此寄存器设定，包括 FIFO 控制寄存器 (UA_FCR)，FIFO 状态寄存器 (UA_FSR) 和线控制寄存器 (UA_LCR) 用于发送和接收。时间超时控制寄存器 (UA_TOR) 应用于识别时间超时中断。该寄存器也包括中断使能寄存器 (UA_IER) 和中断状态寄存器 (UA_ISR) 来使能或禁用中断响应并且识别发生的中断。

**中断控制和状态寄存器**

有 七种类型的中断，发送器 FIFO 空中断 (THRE_INT)，接收器达到阈值中断 (RDA_INT)，线状态中断 (parity 错误 或 framing 错误 或 break 错误) (RLS_INT)，定时溢出中断 (TOUT_INT)，MODEM/唤醒状态中断 (MODEM_INT)，缓存错误中断 (BUF_ERR_INT) 和 LIN 总线中断 (LIN_INT)。

### 6.12.4 基本配置

UART0的基本配置如下:

- UART0 管脚在 P3_MFP 和 P5_MFP 寄存器配置.
- 在UART0_EN (APBCLK[16])中使能 UART0 外设时钟
- 在UART0_RST (IPRSTC2[16])中复位 UART0 控制器

UART1的基本配置如下:

- UART1管脚在P2_MFP 和 PA_MFP寄存器配置.
- 在UART1_EN (APBCLK[17])中使能 UART1 外设时钟
- 在UART1_RST (IPRSTC2[17]).中复位 UART1控制器

### 6.12.5 功能描述

UART控制器支持四种功能模式，包括UART, IrDA, LIN和RS-485模式。用户可以通过对UA_FUN_SEL设置选择功能。四种功能模式将在下面的章节中描述。

#### *6.12.5.1 UART 控制器波特率产生器*

UART控制器包含一个可编程波特率发生器，其通过分频器对输入时钟源分频而得到收发数据所需的串行时钟。下表列出了各种条件下的UART波特率计算公式和参数的设置。通过设置相应的参数和寄存器可以得到零误差的波特率。IrDA功能模式，波特率发生器必须是模式0。寄存器表UART_BAUD 一栏中将更详细地介绍寄存器内容。共有三种设定模式：设定UART_BAUD[29:28]=00 为模式 0，设定 UART_BAUD[29:28]=10 为模式 1，设定UART_BAUD[29:28]=11为模式2。

| 模式 | DIV_X_EN | DIV_X_ONE | 波特率公式 |
|---|---|---|---|
| Mode 0 | 0 | 0 | UART_CLK / [16 * (BRD+2)] |
| Mode 1 | 1 | 0 | UART_CLK / [(DIVIDER _X+1) * (BRD+2)], DIVIDER_X 必须 >= 8 |
| Mode 2 | 1 | 1 | UART_CLK / (BRD+2),<br>如果 UART_CLK <= HCLK, BRD 必须 >=9.<br>如果 HCLK < UART_CLK <= 2*HCLK, BRD 必须 >=15.<br>如果 2*HCLK < UART_CLK <= 3*HCLK, BRD 必须 >=21.<br>如果 UART_CLK > 3*HCLK, 不支持. |

表 6-14 波特率公式表

| UART 外设时钟 = 22.1184 MHz | | | |
|---|---|---|---|
| 波特率 | 模式 0 | 模式 1 | 模式 2 |
| 921600 | 不支持 | BRD=0, DIVIDER_X=11 | BRD=22 |

| | | | |
|---|---|---|---|
| 460800 | BRD=1 | BRD=1, DIVIDER_X =15<br>BRD=2, DIVIDER_X =11 | BRD=46 |
| 230400 | BRD =4 | BRD =4, DIVIDER_X =15<br>BRD =6, DIVIDER_X =11 | BRD =94 |
| 115200 | BRD =10 | BRD =10, DIVIDER_X =15<br>BRD =14, DIVIDER_X =11 | BRD =190 |
| 57600 | BRD =22 | BRD =22, DIVIDER_X =15<br>BRD =30, DIVIDER_X =11 | BRD =382 |
| 38400 | BRD =34 | BRD =62, DIVIDER_X =8<br>BRD =46, DIVIDER_X =11<br>BRD =34, E DIVIDER_X =15 | BRD =574 |
| 19200 | BRD =70 | BRD =126, DIVIDER_X =8<br>BRD =94, DIVIDER_X =11<br>BRD =70, DIVIDER_X =15 | BRD =1150 |
| 9600 | BRD =142 | BRD =254, DIVIDER_X =8<br>BRD =190, DIVIDER_X =11<br>BRD =142, DIVIDER_X =15 | BRD =2302 |
| 4800 | BRD =286 | BRD =510, DIVIDER_X =8<br>BRD =382, DIVIDER_X =11<br>BRD =286, DIVIDER_X =15 | BRD =4606 |

表 6-15 UART 控制器波特率参数设置示例表

| **UART 外设 时钟 = 22.1184 MHz** | | | |
|---|---|---|---|
| **波特率** | **UA_BAUD 值** | | |
| | **模式 0** | **模式 1** | **模式 2** |
| 921600 | Not support | 0x2B00_0000 | 0x3000_0016 |
| 460800 | 0x0000_0001 | 0x2F00_0001<br>0x2B00_0002 | 0x3000_002E |
| 230400 | 0x0000_0004 | 0x2F00_0004<br>0x2B00_0006 | 0x3000_005E |
| 115200 | 0x0000_000A | 0x2F00_000A<br>0x2B00_000E | 0x3000_00BE |
| 57600 | 0x0000_0016 | 0x2F00_0016<br>0x2B00_001E | 0x3000_017E |
| 38400 | 0x0000_0022 | 0x2800_003E<br>0x2B00_002E<br>0x2F00_0022 | 0x3000_023E |
| 19200 | 0x0000_0046 | 0x2800_007E<br>0x2B00_005E<br>0x2F00_0046 | 0x3000_047E |
| 9600 | 0x0000_008E | 0x2800_00FE<br>0x2B00_00BE<br>0x2F00_008E | 0x3000_08FE |

| | | | |
|---|---|---|---|
| 4800 | 0x0000_011E | 0x2800_01FE<br>0x2B00_017E<br>0x2F00_011E | 0x3000_11FE |

表 6-16 UART 控制器波特率寄存器设置示例表

#### *6.12.5.2 UART 控制器发送延时时间值*

UART 控制器可以通过设置DLY (UART_TOUT [15:8])来控制传输过程两个数据帧之间即上一个数据帧的停止位和下一个数据帧的起始位之间的间隔。单位是波特。延时操作如下图所示图 6-73.

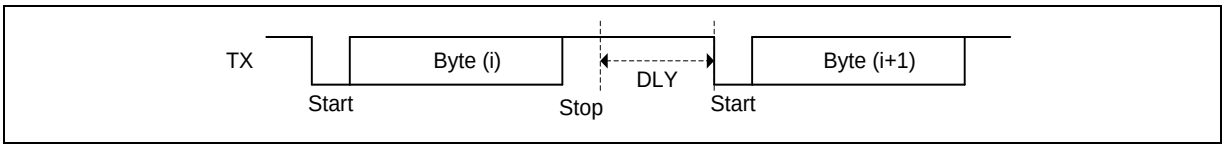



图 6-73 发送延时操作

#### *6.12.5.3 UART 控制器FIFO控制和状态*

UART控制器内置一个16字节的发送FIFO (TX_FIFO)和一个16字节的接收FIFO (RX_FIFO), 通讯中,使用这些收发FIFO可以减少对CPU的中断次数。CPU在操作过程中任何时间都可以读取UART的状态，包括3个错误状态（校验错误，帧错误，间隔错误）如果接收的数据出现校验错误、帧或间隔错误。FIFO控制和状态也支持所有的功能模式，包括UART, IrDA, LIN和RS-485模式。

如果在发送或接收FIFO中有任何的溢出事件，缓存错误标志BUF_ERR_IF (UA_ISR[5])会自动置位。

#### *6.12.5.4 UART 控制器唤醒功能*

UART控制器支持系统唤醒功能。唤醒功能包括nCTS唤醒。 当芯片在Power-down模式下，nCTS脚可以唤醒系统。图 6-74 列举了唤醒功能。

nCTS 唤醒情况 1 (nCTS 从低到高变化)

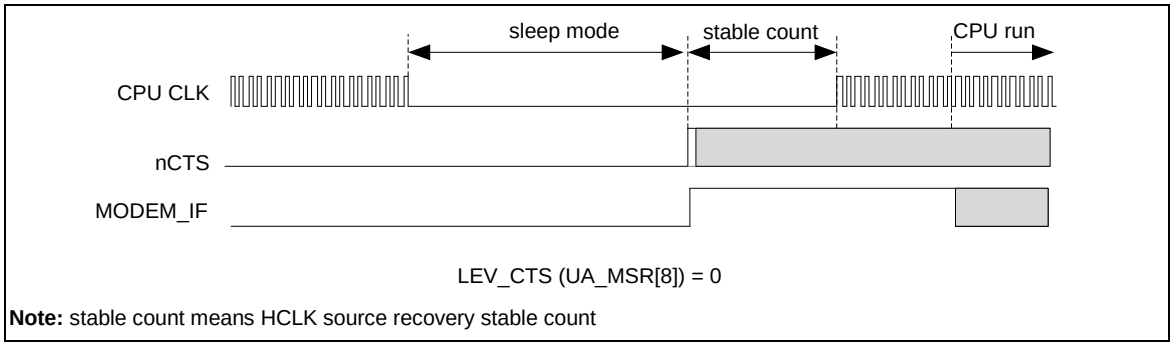
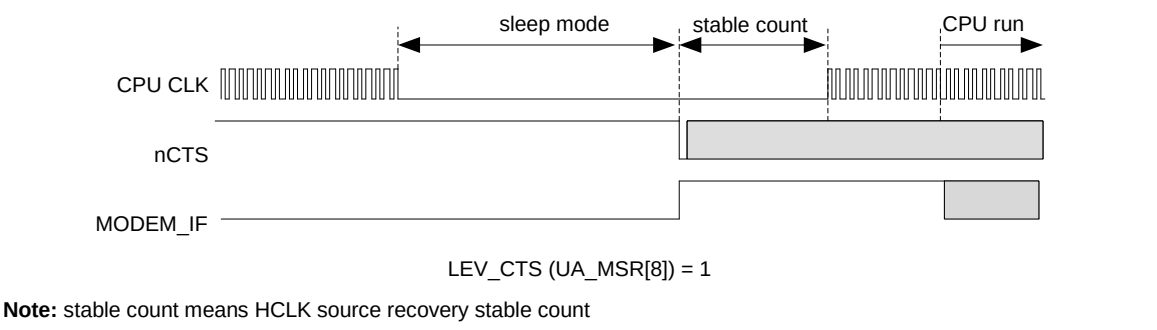



图 6-74 UART nCTS 唤醒情况1

nCTS 唤醒情况 2 (nCTS 从高到低变化)




**Note:** stable count means HCLK source recovery stable count

图 6-75 UART nCTS 唤醒情况2

### *6.12.5.5 UART 控制器中断和状态*

每个UART控制器支持7种类型的中断，包括：

- 达到接收阈值水平的中断(RDA_INT)
- 发送FIFO空中断(THRE_INT)
- Line状态中断（校验错误，帧错误，间隔错误）(RLS_INT)
- MODEM状态中断(MODEM_INT)
- 接收缓冲区定时溢出中断(TOUT_INT)
- 缓冲区错误中断(BUF_ERR_INT)
- LIN总线中断(LIN_INT)

表 6-17描述了中断源和中断标志。当中断使能且有中断标志时就会产生中断。用户必须在中断后清中断标志。

| UART 中断源 | 中断使能位 | 中断指示 | 中断标志 | 标志清除 |
|---|---|---|---|---|
| LIN 中断 | LIN_IEN | LIN_INT | **LIN_IF** | 写 '1' 到 LINS_HDET_F/LIN_BKDET_F/BIT_ERR_F/LINS_IDPERR_F/LINS_HERR_F |
| 缓存错误中断 | BUF_ERR_IEN | BUF_ERR_INT | **BUF_ERR_IF** = TX_OVER_IF | 写 '1'到 TX_OVER_IF |
| | | | **BUF_ERR_IF** = RX_OVER_IF | 写 '1' 到 RX_OVER_IF |
| 接收缓存超时中断 | TOUT_IEN | TOUT_INT | **TOUT_IF** | 读 UA_RBR |
| Modem 状态中断 | MODEM_IEN | MODEM_INT | **MODEM_IF** = DCTSF | 写 '1' 到 DCTSF |
| 接收线状态中断 | RLS_IEN | RLS_INT | **RLS_IF** = BIF | 写 '1' 到 BIF |
| | | | **RLS_IF** = FEF | 写 '1' 到 FEF |
| | | | **RLS_IF** = PEF | 写 '1' 到 PEF |
| | | | **RLS_IF** = RS485_ADD_DETF | 写 '1' 到 |

| | | | | |
|---|---|---|---|---|
| | | | | RS485_ADD_DETF |
| 发送保持寄存器空中断 | THRE_IEN | THRE_INT | **THRE_IF** | 写 UA_THR |
| 接收数据有效中断 | RDA_IEN | RDA_INT | **RDA_IF** | 读 UA_RBR |

表 6-17 UART控制器中断源和标志列表

### *6.12.5.6 UART 功能模式*

UART控制器提供UART功能(设置 FUN_SEL (UA_FUN_SEL [1:0]) 为 00 使能 UART 功能 模式)，UART 波特率最高速度是1 Mbps

UART为全双工异步通讯接口。收发各包含一个16字节的FIFO缓冲区。用户可以设置接收时的FIFO触发阈值以及定时溢出检测时间。

UART支持硬件自动流控功能，且nRTS流控触发电平可设。如果RX FIFO中数据字节数大于或等于RTS_TRI_LEV (UA_FCR[19:16])，nRTS将被释放。

**UART线控功能**

UART控制器通过设置UA_LCR寄存器支持串行接口全部特性。可以通过对UA_LCR寄存器设置数据位和停止位长度以及校验位。下表列出了UART数据位和停止位长度的设置以及UART校验位的设置。

| NSB (UA_LCR[2]) | WLS (UA_LCR[1:0]) | 数据位长度 (Bit) | 停止位长度 (Bit) |
|---|---|---|---|
| 0 | 00 | 5 | 1 |
| 0 | 01 | 6 | 1 |
| 0 | 10 | 7 | 1 |
| 0 | 11 | 8 | 1 |
| 1 | 00 | 5 | 1.5 |
| 1 | 01 | 6 | 2 |
| 1 | 10 | 7 | 2 |
| 1 | 11 | 8 | 2 |

表 6-18 UART 线控的数据位和停止位长度设置

| 校验类型 | SPE (UA_LCR[5]) | EPE (UA_LCR[4]) | PBE (UA_LCR[3]) | 描述 |
|---|---|---|---|---|
| 无校验 | x | x | 0 | 无奇偶校验位输出 |
| 奇校验 | 0 | 0 | 1 | 奇校验位的计算方法是把数据流中的所有的1相加，使得包括校验位在内,1的总数为奇数个. |
| 偶校验 | 0 | 1 | 1 | 偶校验位的计算方法是把数据流中的所有的1相加，使得包括校验位在内,1的总数为偶数个 |

| | | | | |
|---|---|---|---|---|
| 奇偶校验位强制置1 | 1 | 0 | 1 | 奇偶校验位总是逻辑1<br>不管数据位中1的个数是奇数还是偶数，奇偶校验位永远都是逻辑1 |
| 奇偶校验位强制为0 | 1 | 1 | 1 | 奇偶校验位总是逻辑0<br>不管数据位中1的个数是奇数还是偶数，奇偶校验位永远都是逻辑0 |

表 6-19 UART 线控校验设置

**UART 自动流控功能**

UART支持自动流控功能，该功能用到两根信号线nCTS (清零发送) 和nRTS (请求发送)来控制UART与外部设备（如Modem）间的数据传输。当自动流控使能后，只有等到UART对外部设备发出有效的nRTS信号后才允许接收数据，否则不接收。当RX FIFO接收到数据的数量达等于RTS_TRI_LEV (UA_FCR[19:16])位的值后，nRTS信号会被取消。当UART检测到外部设备给nCTS信号脚有效信号后，UART开始发送数据，否则UART不会发送数据。

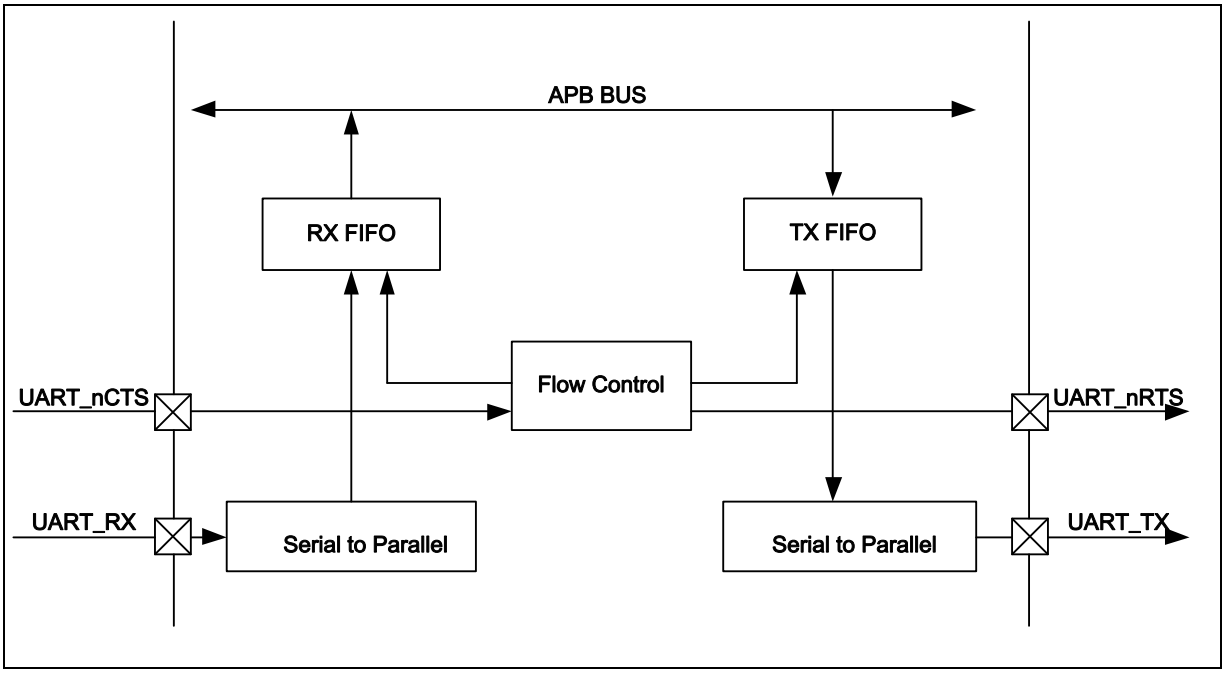



图 6-76 自动流控框图

图 6-77 展示了UART nCTS自动流控功能的框图。用户必须要先设置AUTO_CTS_EN (UA_IER[13])以使能nCTS自动流控功能。LEV_CTS (UA_MSR[8])位可以设置nCTS脚输入的有效状态。当nCTS脚上任何电平变化将导致DCTSF (UA_MSR[0])位被置1，然后TX FIFO将自动将数据发送到TX脚,并传送出去。

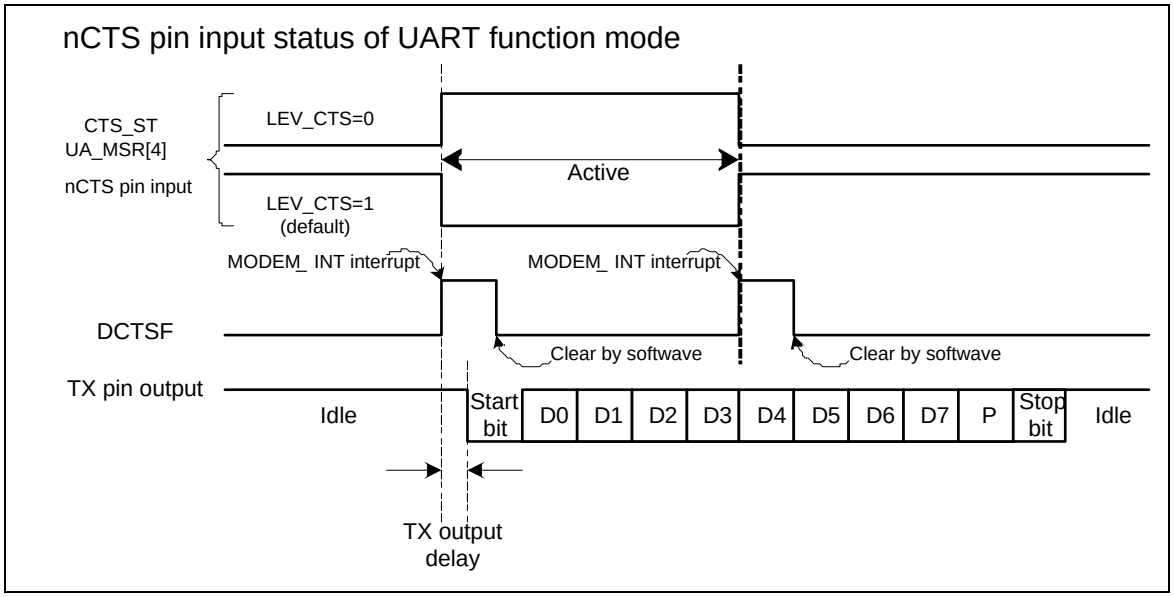



图 6-77 UART nCTS自动流控使能

如图 6-78,所示，UART nRTS自动流控模式(AUTO_RTS_EN (UA_IER[12])=1)中，nRTS的触发阈值由UA_FCR控制寄存器的RTS_TRI_LEV (UA_FCR[19:16])控制 。

设置LEV_RTS(UA_MCR[9])可以控制nRTS 脚输出反向或非反向。用户可以读RTS_ST (UA_MCR[13])来获取nRTS脚输出电压的真实逻辑状态。

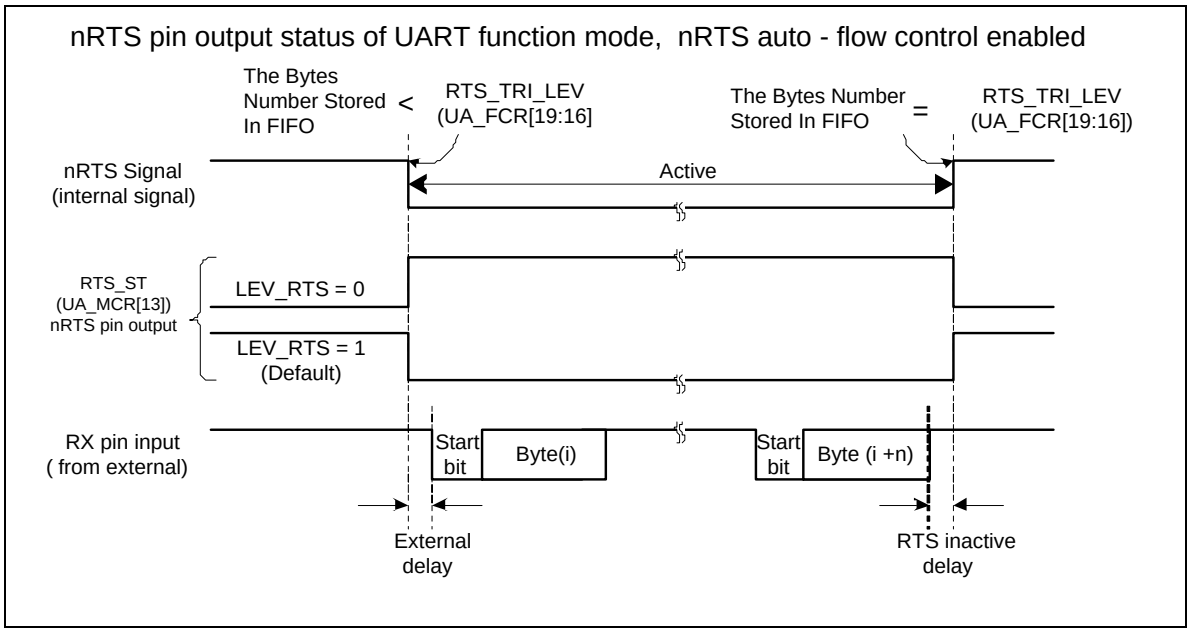



图 6-78 UART nRTS自动流控使能

如下图，在软件模式下(AUTO_RTS_EN (UA_IER[12])=0)，软件改动RTS(UA_MCR[1])来实现

nRTS流控。

设置LEV_RTS (UA_MCR[9])可以控制nRTS输出脚状态与RTS(UA_MCR[1])控制状态是同向还是反向。用户可以读RTS_ST(UA_MCR[13])位来获取nRTS脚真实输出电平状态。

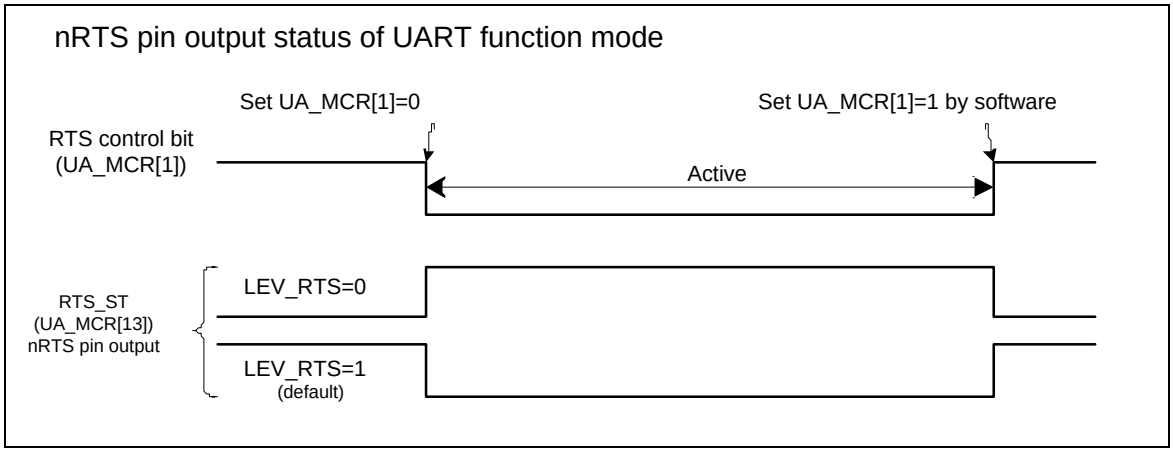



图 6-79 UART nRTS软件方式下的自动流控

### 6.12.5.7 *IrDA 功能模式*

UART控制器也提供Serial IrDA (SIR, 串行红外)功能(用户必须设置FUN_SEL (UA_FUN_SEL[1:0])为'10')来使能IrDA功能。SIR规范定义了一个短距离红外异步串行传输模式，它包括一个起始位，8个数据位，一个停止位。最大速率115.2kbps。 IrDA SIR模块包含一个IrDA SIR协议编/解码器。IrDA SIR协议是半双工工作模式，所以它不能同时收发数据。IrDA SIR物理层规定了发送与接收数据的时间上至少10ms的时间间隔，该延迟特性需通过软件来完成。

IrDA 模式下，DIV_X_EN (UA_BAUD[29])需为0。

波特率= Clock / (16 * BRD +2)，这里BRD是UA_BAUD寄存器中定义的波特率分频器。

以下框图展示了IrDA控制模块框图：

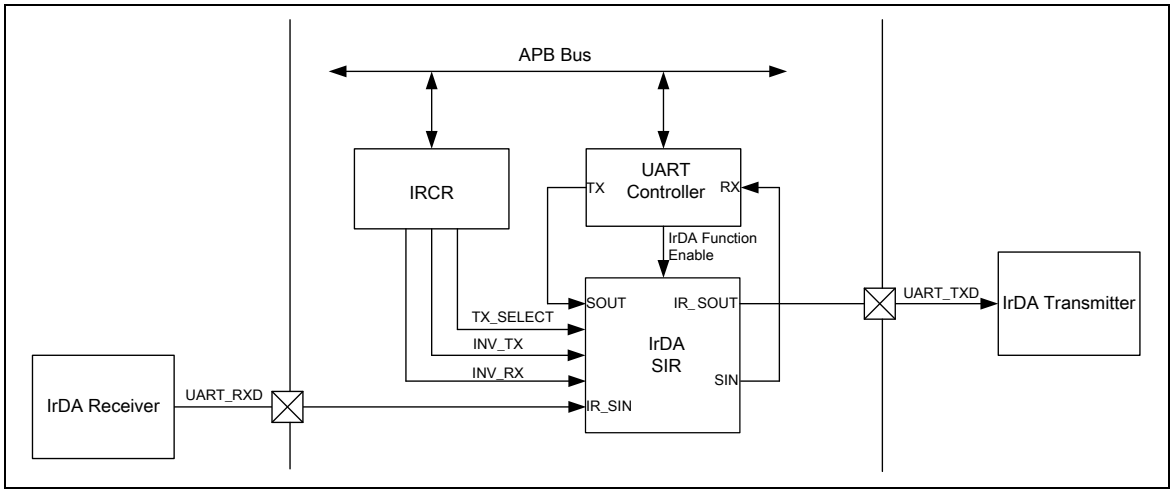



图 6-80 IrDA 框图

#### *6.12.5.7.1 IrDA SIR 发送编码器*

IrDA SIR 传送编码调制采用 Non-Return-to Zero (NRZ)编码，数据流编码后从UART 接口输出。IrDA SIR 物理层指定使用归零反向调制编码 (Return-to-Zero, Inverted (RZI))，用一个红外光脉冲代表逻辑 0，被调整的脉冲输出到外部输出驱动器和红外线发光二极管。

在正常模式下，传输脉冲的宽度为3/16 波特率周期。

#### *6.12.5.7.2 IrDA SIR 接收解码器*

IrDA SIR 接收解码器对输入管脚的(Return-to-Zero, Inverted (RZI))串行位流进行解调，并输出NRZ串行位流到 UART接收数据输入端。当解码器输入端为低时，表明接收到一个起始位。在空闲状态里，解码器输入端通常为高。正常操作时，INV_RX (UA_IRCR[6])设置为1，INV_TX (UA_IRCR[5])设置为0。

#### *6.12.5.7.3 IrDA SIR 操作*

IrDA SIR 编码/解码提供 UART 数据流和半双工串行 SIR 之间的转换。IrDA编码/解码波形图如下：

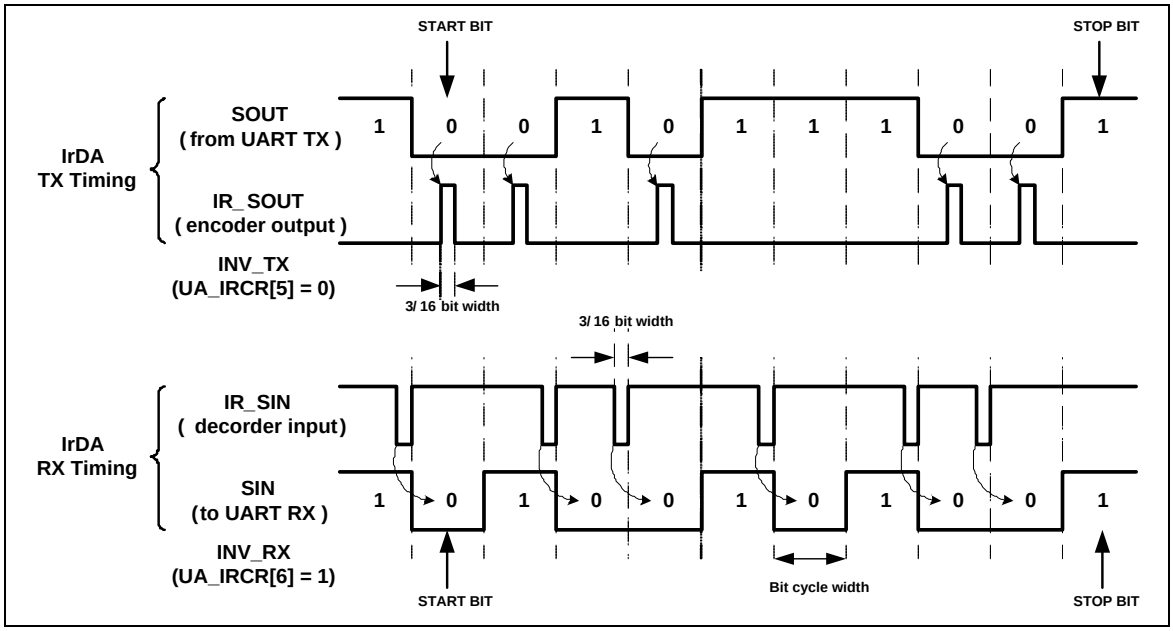



图 6-81 IrDA 时序框图

#### 6.12.5.8 RS-485 功能模式

UART控制器另一个可选择的功能是RS-485功能（用户必须设置FUN_SEL (UA_FUN_SEL[1:0]) = '11'来使能RS-485功能），方向控制则由异步串口的nRTS脚来控制。RS-485收发器的驱动控制是通过nRTS控制信号来驱动的。RS-485模式下的RX和TX大多数特性与UART相同。

RS-485模式，控制器可以配置成 RS-485 可寻址的从机模式，RS-485 主机发送可通过设置检验位 ($9^{th}$ bit) 为 1标识地址特性。对于数据字符，检验位设置为 0。设置寄存器UA_LCR控制第9位（UA_LCR[3]), EPE (UA_LCR[4]) 和 SPE (UA_LCR[5])置位时，第9位发送 0；PBE (UA_LCR[3]) 和 SPE (UA_LCR[5])置位，EPE (UA_LCR[4])清零时，第 9 位发送1）。

该控制器支持三种操作模式：RS-485 普通多点操作模式 (NMM)，RS-485 自动地址识别模式 (AAD) 和 RS-485 自动方向控制模式 (AUD)，可通过UA_ALT_CSR寄存器的设置选择其中一种工作模式，通过设置DLY (UA_TOR[15:8])可以设置上一个停止位与下一个开始位之间的延迟时间。

该控制器支持三种操作模式是

**RS-485 普通多点操作模式 (NMM)**

RS-485 普通多点操作模式(RS485NMM (RS485_NMM (UA_ALT_CSR[8]) = 1)，首先，软件决定在检测到地址字节之前的数据是否存储到 RX-FIFO。如果软件想忽略在检测到地址之前的任何数据，流程是设置RX_DIS (UA_FCR[8])，然后使能RS485_NMM (UA_ALT_CSR[8])，接收器将会忽略数据，直到检测到地址字节 (bit9 =1) 并且地址字节数据存储到RX-FIFO。如果软件想接收在检测到地址字节之前的任何数据，流程是禁用RX_DIS (UA_FCR[8])，然后使能RS485_NMM (UA_ALT_CSR[8])，接收器将接收任何数据。

如果检测到地址字节 (bit9 =1)，会产生一个中断到 CPU，软件可以通过设置RX_DIS (UA_FCR[8]) ，决定是否使能或禁用接收器接收数据。如果使能接收器，就会接收所有字节数据并存储到 RX-FIFO。如果设置RX_DIS (UA_FCR[8])禁用接收器，会忽略所有接收到的字节数据，直到检测到下一个地址字节。当检测到地址字节后，控制器将清除RX_DIS (UA_FCR[8])且地址字节数据将存储到 RX-FIFO。

**RS-485 自动地址识别操作模式 (AAD)**

RS-485 自动地址识别模式(RS485_AAD (UA_ALT_CSR[9]) = 1)，接收器在检测到地址字节 (bit9=1) 并且地址字节数据与ADDR_MATCH (UA_ALT_CSR [31:24])的值相匹配之前将忽略所有数据。地址字节数据将存储在 RX-FIFO。所有接收字节数据将被接收并存储于 RX-FIFO 直到地址字节或数据字节与ADDR_MATCH (UA_ALT_CSR[31:24])的值不匹配。

**RS-485 自动方向控制模式 (AUD)**

RS-485控制器的另一个功能是RS-485 自动方向控制功能(RS485_AUD (UA_ALT_CSR[10] = 1)。RS-485 通过nRTS 驱动控制异步串口的控制信号，使能RS-485 驱动器。nRTS 连接到RS-485 驱动器，设置nRTS线为高（逻辑1）使能RS-485 驱动器。设置nRTS 为低（逻辑0），使驱动器进入 tri-state 状态。用户通过设置LEV_RTS (UA_MCR[9])改变nRTS 驱动电平。

以下框图展示了AUD模式下RS-485 nRTS驱动电平。nRTS脚在TX数据发送阶段自动驱动收发器。

设置LEV_RTS (UA_MCR[9])可以控制nRTS脚的输出电平。用户可以通过读RTS_ST(UA_MCR[13]) 来获取nRTS脚上实际的输出逻辑电平。

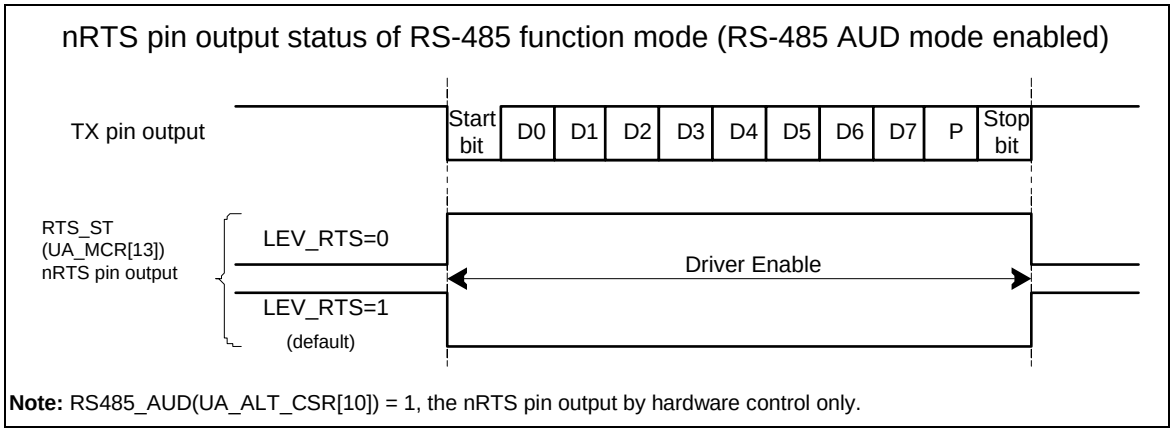



图 6-82 RS-485在自动方向模式下nRTS的驱动电平

下图展示了通过软件控制(RS485_AUD (UA_ALT_CSR[10])=0)模式下，RS-485 nRTS脚的驱动电平。nRTS驱动电平通过设置RTS(UA_MCR[1])来控制。

设置LEV_RTS (UA_MCR[9])可以控制nRTS脚的输出与RTS (UA_MCR[1])设置值是否相反。用户可以读RTS_ST (UA_MCR[13])来获取nRTS脚上实际的逻辑电平。

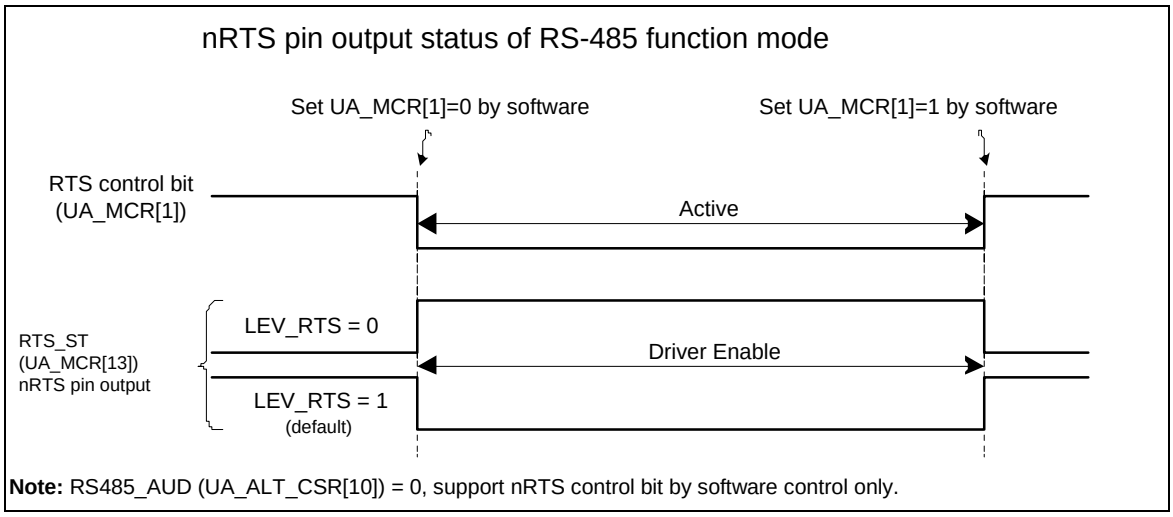
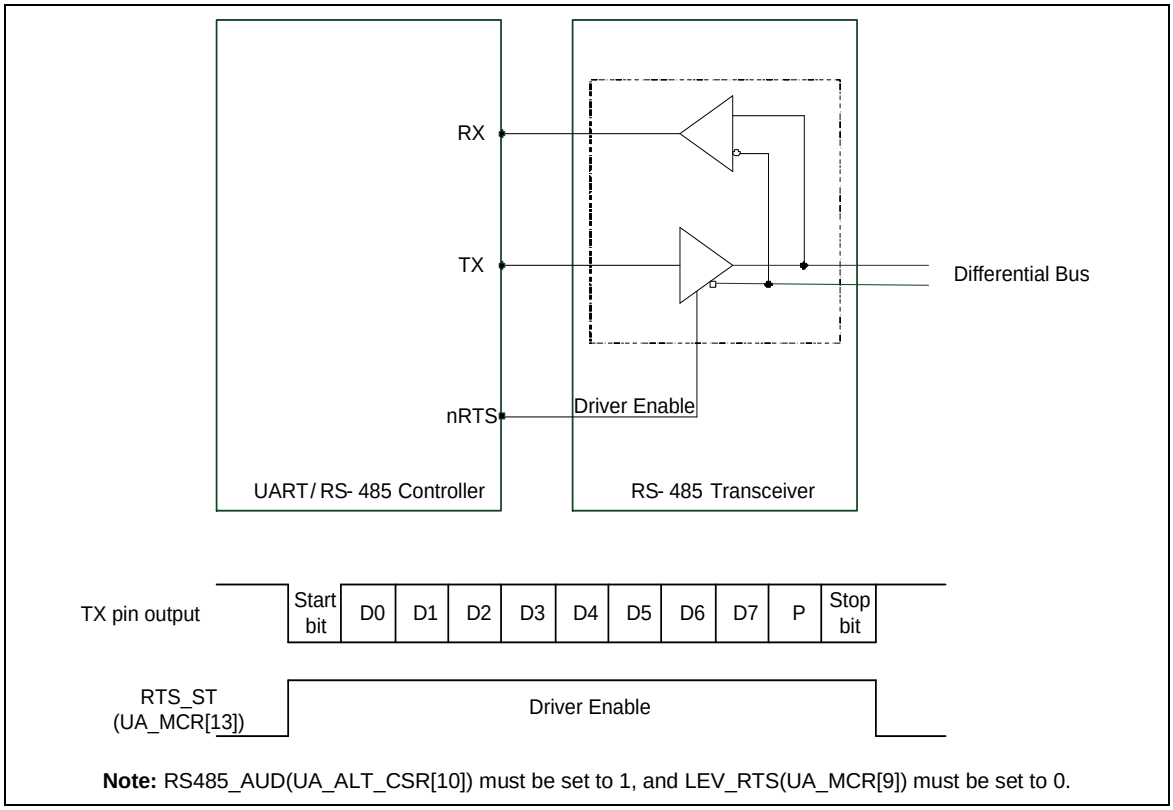



图 6-83 RS-485 在软件控制模式下nRTS的驱动电平

编程流程示例:

1. 设置UA_FUN_SEL中的 FUN_SEL 选择 RS-485 功能
2. 设置RX_DIS (UA_FCR[8])，使能或禁用 RS-485 接收器.
3. 设置RS485_NMM (UA_ALT_CSR[8])或RS485_AAD (UA_ALT_CSR[9])模式
4. 如果选择RS485_AAD (UA_ALT_CSR[9])模式，ADDR_MATCH (UA_ALT_CSR[31:24])需设置自动地址匹配值
5. 设置RS485_AUD (UA_ALT_CSR[10])来决定是否为自动方向控制




**Note:** RS485_AUD(UA_ALT_CSR[10]) must be set to 1, and LEV_RTS(UA_MCR[9]) must be set to 0.

图 6-84 RS-485 帧结构

#### 6.12.5.9 LIN (本地互连网络) 模式

UART支持LIN功能，通过设置FUN_SEL (UA_FUN_SEL[1:0])为 '01'可以将UART设定为LIN模式。在主机模式，UART支持LIN break/delimiter的产生以及break/delimiter的侦测，在LIN从机模式下，支持报头侦测和自动重新同步。

##### 6.12.5.9.1 LIN 帧结构

根据LIN协议，所有的传输信息被打包为帧。一个帧由一个报头（主机任务提供）和一个紧跟其后的应答（从机任务提供）组成。报头（主机任务提供）由一个break域和一个sync（同步）域再跟一个帧识别码(frame ID)组成。帧ID仅作为定义帧的用途。从机任务负责回应相关的帧ID。响应由一个数据域和一个校验域组成。下图是LIN帧的结构

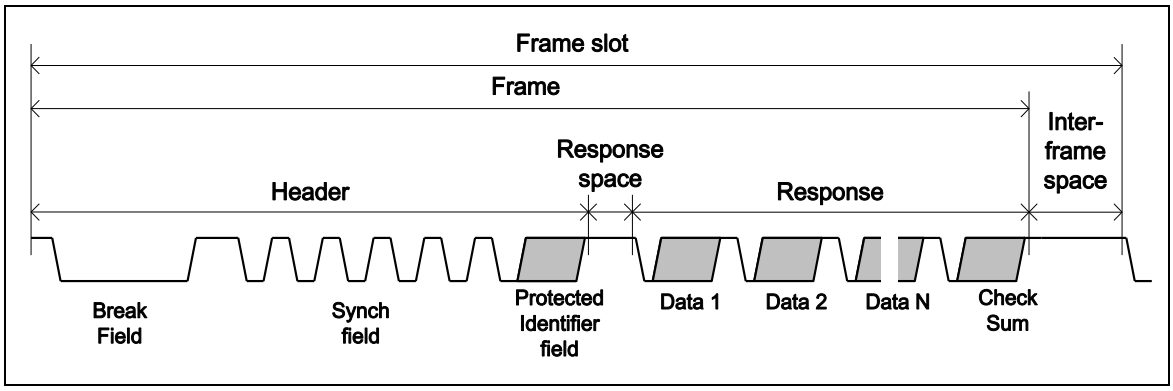



图 6-85 LIN 帧结构

##### 6.12.5.9.2 LIN 字节结构

在LIN模式，根据LIN的标准，每个字节由值为0（显性）的START位开始，接着是8位数据位，没有校验位，LSB优先，由一个值为1（NSB (UA_LCR[2]) = 0）（隐性）的STOP位结束。字节的结构如下：

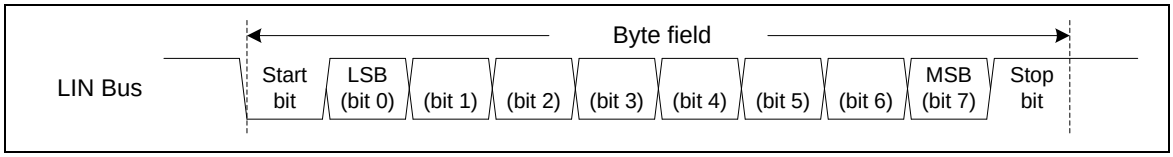



图 6-86 LIN 字节结构

##### 6.12.5.9.3 LIN 主机模式

通过设置UA_FUN_SEL寄存器，UART控制器支持LIN主机模式，使能并初始化LIN主机模式需要如下步骤：

1. 通过设置 UA_BAUD寄存器设定波特率.
2. 设置WLS (UA_LCR[1:0]) ='11'配置数据长度为8位，PBE (UA_LCR[3])=0禁止奇偶校验，NSB (UA_LCR[2])=0配置1位stop位。
3. 设置UA_FUN_SEL 寄存器选择LIN功能模式

一个完整的报头由一个break域和同步域再跟一个帧标识符(帧ID)组成。UART控制器可以选择三种报头发送模式，通过设置LIN_HEAD_SEL (UA_LIN_CTL[23:22]) 可以选择“break 域” 或 “break 域和同步域” 或 “break 域, 同步域和 帧 ID 域”作为发送的报头。如果选择的报头是“break 域”， 发送一个完整的报头到总线，软件必须依次填同步数据(0x55)和帧ID数据到UA_THR 寄存器。如果选择的报头是“break 域和同步域”，软件必须填帧ID数据到UA_THR 寄存器，按顺列来发送一个完整的报头到总线。如果选择的报头是“break 域, 同步域和 帧 ID 域”，硬件会自动控制报头发送顺序，但是软件必须填帧ID数据到LIN_PID (UA_LIN_CTL[31:24])。当选择的报头模式是“break 域, 同步域和 帧 ID 域”时，帧ID校验位可以由软件或硬件来计算，这取决于LIN_IDPEN (UA_LIN_CTL[9])位是否置位。

| LIN_HEAD_SEL | Break 域 | Sync 域 | ID 域 |
|---|---|---|---|
| 0 | 由H/W产生 | 由S/W处理 | 由SW处理 |
| 1 | 由H/W产生 | 由H/W产生 | 由SW处理 |
| 2 | 由H/W产生 | 由H/W产生 | 由 H/W 产生(但是软件需要先填 ID 到 LIN_PID (UA_LIN_CTL[31:24])) |

表 6-20 LIN主模式时报头选择

当UART 工作于LIN数据传输模式时，LIN总线传输状态可以由硬件或软件监控。通过设置BIT_ERR_EN (UA_LIN_CTL[12])为 “1”使能硬件监控。如果在LIN发送状态输入管脚(SIN) 状态不同于输出管脚(SOUT) 状态，硬件会产生一个中断到CPU。软件也能通过读回UA_RBR寄存器数据监视LIN总线传输状态。以下为编程顺序示例。

在主机模式下，没有软件错误监控的步骤

1. 填保密标识符ID到LIN_PID (UA_LIN_CTL[31:24]).
2. 通过设置LIN_HEAD_SEL (UA_LIN_CTL[23:22]) = 10，选择硬件传输的报头域包括“break 域 + 同步域 +保密标识符ID域”
3. 通过设置LIN_SHD (UA_LIN_CTL[8])为1选择硬件传输报头.
4. 等待LIN_SHD (UA_LIN_CTL[8])被硬件清零.
5. 等待TE_FLAG (UA_FSR[28])被硬件置位.

**注意1:** break域 的预设值是13 个显性位 (break 域) 和1 个隐性位 (break/sync 分隔符). 软件可以通过设定LIN_BKFL (UA_LIN_CTL[19:16]) 和 LIN_BS_LEN (UA_LIN_CTL[21:20]),来改变break域的长度和break/sync分隔符的长度.

**注意2:** break/sync分隔符预设长度是1比特时间，字节间隔预设也是1个比特时间。软件可以通过设定LIN_BS_LEN (UA_LIN_CTL[21:20]) 和 DLY (UA_TOR[15:8])来改变它们的间隔.

**注意3:** 如果报头包含“break 域, sync 域和帧ID 域”, 在触发硬件发送报头之前(设定LIN_SHD (UA_LIN_CTL[8]))，软件必须填帧 ID 到LIN_PID (UA_LIN_CTL[31:24])。根据LIN_IDPEN (UA_LIN_CTL[9])的设定，帧 ID 校验可以由硬件或者软件产生。如果校验由软件产生LIN_IDPEN (UA_LIN_CTL[9])为0, 软件必须填8 位数据 (包括 2 个校验位)到帧ID域；如果校验位由硬件产生 (LIN_IDPEN (UA_LIN_CTL[9])=1), 软件填 ID0~ID5就好，硬件负责计算P0和 P1。

在主机模式下，有软件错误监控的步骤：

1. 设置LIN_HEAD_SEL (UA_LIN_CTL[23:22]) = 0x00，选择硬件传输的报头域，其内容只包括“break 域”
2. 设置LIN_BKDET_EN (UA_LIN_CTL[10])使能 break 域侦测功能
3. 设置LIN_SHD (UA_LIN_CTL[8])请求硬件传输 break + break/同步 分隔符
4. 等待，直到LIN_BKDET_F (UA_LIN_SR[8])标志被硬件置为 “1”.
5. 写0x55到UA_THR寄存器发送 sync 域
6. 等待，直到RDA_IF (UA_ISR[0])标志被硬件置为 “1” ，然后读回UA_RBR寄存器的值.
7. 写protected identifier 到UA_THR寄存器，发送帧ID 域
8. 等待，直到RDA_IF (UA_ISR[0])标志被硬件置为 “1”， 然后读回UA_RBR寄存器的值

**LIN break 和分隔符侦测**

当软件通过设定LIN_BKDET_EN (UA_LIN_CTL[10])使能break 域侦测功能时， break域侦测功能被激活。break 域侦测电路和UART接收电路是完全独立的。

当break 域侦测功能使能时,电路侦测SIN引脚的起始信号。如果UART LIN控制器侦测到显性连续大于11个比特并跟随1个隐性比特 (分隔符)，在break域结束时LIN_BKDET_F (UA_LIN_SR[8])标志将被置位，如果LIN_IEN (UA_IER[8]) =1,中断将发生，break 侦测和break 标志如下图所示。

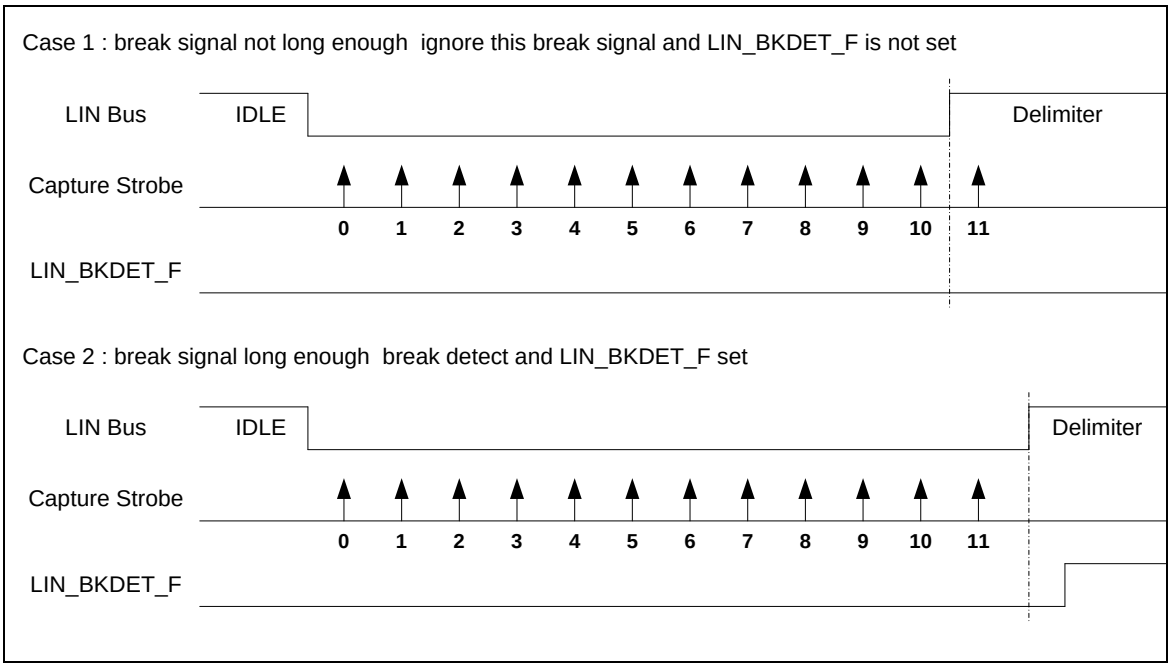



图 6-87 LIN 模式Break侦测

*6.12.5.9.4 LIN 从机模式*

通过设置LINS_EN (UA_LIN_CTL[0]). UART控制器支持LIN从模式。使能和初始化LIN从模式的必要步骤如下：

1. 设置UART_BAUD 寄存器设定波特率
2. 设定WLS (UA_LCR[1:0]) = 11设定数据长度为8 比特，清PBE (UA_LCR[3])禁止校验，清NSB (UA_LCR[2])]设定停止位为1比特
3. 设定UA_FUN_SEL 寄存器选择 LIN 功能
4. 设定LINS_EN (UA_LIN_CTL[0])为1，使能LIN 从机模式

**LIN 报头接收**

根据LIN 协议, 从节点必须等待从主节点接收一个有效的报头，然后应用程序可以采取以下动作 (根据主报头帧ID 的值)

- 接收响应.
- 发送响应.
- 忽略响应，等待下一个报头.

LIN从模式下，通过设定LINS_HDET_EN (UA_LIN_CTL[1])，使能从机报头检测功能来侦测完整的报头(接收 “break 域”, “sync 域” 和 “frame ID 域”)。接收到一个 LIN 报头后，LINS_HDET_F (UA_LIN_SR[0])标志将被置位。如果LIN_IEN (UA_IER[8]), 中断将发生。 通过设定LIN_IDPEN (UA_LIN_CTL[9])，使能帧ID校验检查功能。如果收到的帧ID校验位不正确时 (break 和 sync 域正确), LIN_IDPERR_F (UA_LIN_SR[2])标志将被置1。如果LINIEN(UART_INTEN[8]) =1, 中断将发生，LINS_HDET_F (UA_LIN_SR[0])会被置1。通过设定LIN_MUTE_EN (UA_LIN_CTL[4])=1，用户也可以将 LIN 设为mute 模式。该模式只允许检测报头(break + sync + frame ID) ，不允许接收任何其它字符。为了避免比特率偏差, 控制器支持自动重同步功能，避免时钟误差错误，通过设定LINS_ARS_EN (UA_LIN_CTL[2])使能该特性。

**LIN发送响应**

LIN 从机模式可以发送响应和接收响应。当从机节点是响应的发送者时，通过将数据填到UA_THR寄存器发送响应；如果从机节点是响应的接收者时，从接节点从LIN总线接收数据

**LIN 报头超时错误**

LIN 从机控制器包含一个报头超时计数器。如果整个报头没有在57比特的最大时间限制内收到，报头错误标志LINS_HERR_F (UR_LIN_SR[1])将被置位。超时计数器在每个break侦测沿使能，在下列条件下将停止：

- LIN 帧 ID 域已经收到了
- 报头错误标志被置起
- 写 1 到LINS_SYNC_F (UR_LIN_SR[3])，重新侦测新的报头

**LIN Mute 模式和从mute模式退出的条件**

Mute模式下, LIN 从机节点需满足一定的条件时才能接收其它数据。此模式只允许侦测报头，禁止接收任何其它数据。通过设定LIN_MUTE_EN (UA_LIN_CTL[4])使能Mute 模式，设置LIN_HEAD_SEL (UA_LIN_CTL[23:22])来退出Mute模式。

**注意:** 建议校验字节发送之后，将LIN从机节点设定为Mute模式

LIN 从机控制器从Mute模式退出描述如下：如果LIN_HEAD_SEL (UA_LIN_CTL[23:22])设为“break 域”， 当 LIN 从机控制器检测到一个有效的LIN break + 分隔符时, 控制器将使能接收器 (退出Mute 模式) ，随后的数据(sync 数据, 帧ID , 响应数据) 将被收到 RX-FIFO中。

如果LIN_HEAD_SEL (UA_LIN_CTL[23:22])设为 “break 域 和 sync 域”, 当 LIN 从机控制器检测到一个有效的LIN break + 分隔符，后面跟一个有效的同步域并且没有帧错误时，控制器将使能接收器 (退出Mute 模式)， 随后的数据(帧 ID, 响应数据) 将被收到 RX-FIFO中。

如果LIN_HEAD_SEL (UA_LIN_CTL[23:22])设为“break 域, sync 域和 ID 域”, 当 LIN 从机控制器检测到一个有效的LIN break + 分隔符和有效的同步域，并且没有帧错误后面跟一个有效的帧ID域并且没有帧错误，并且收到的帧ID和LIN_PID (UA_LIN_CTL[31:24])中的值匹配时，控制器将使能接收器 (退出Mute 模式) 随后的数据(响应数据) 将被收到 RX-FIFO中。

**非自动重新同步从机模式(NAR)**

用户可以关闭自动重新同步功能来固定通信波特率。当工作在非自动重新同步模式时，软件需要一些初始化过程，初始化过程如下所示：

1. 设定UA_BAUD 寄存器设定波特率
2. 设定UA_FUN_SEL寄存器选择 LIN 功能
3. 设定LINS_ARS_EN (UA_LIN_CTL[2]) = 0关闭自动重新同步功能
4. 设定LINS_EN (UA_LIN_CTL[0]).为1使能 LIN 从机模式.

**自动重新同步从机模式(AR)**

自动重新同步模式下，每次收到同步域时，控制器将调整比特率发生器。软件需要一些初始化过程，初始化过程如下所示

1. 设定UA_BAUD 寄存器设定波特率
2. 设定UA_FUN_SEL寄存器选择 LIN 功能模式
3. 设定LINS_ARS_EN (UA_LIN_CTL[2]) = 1使能自动重新同步功能
4. 设定LINS_EN (UA_LIN_CTL[0]).为‘1’，使能LIN从机模式

当自动重新同步功能使能后, 每个LIN break 域后面, 用LIN的工作时钟持续采样5个下降沿的时间，测量的结果储存在内部一个13-bit 寄存器中，在第5个下降沿结束，UA_BAUD 寄存器的值将被自动更新。如果在5个下降沿之前，测量计数器(13-bit) 溢出, 报头错误标志LINS_HERR_F (UA_LIN_SR[1])将被置位。

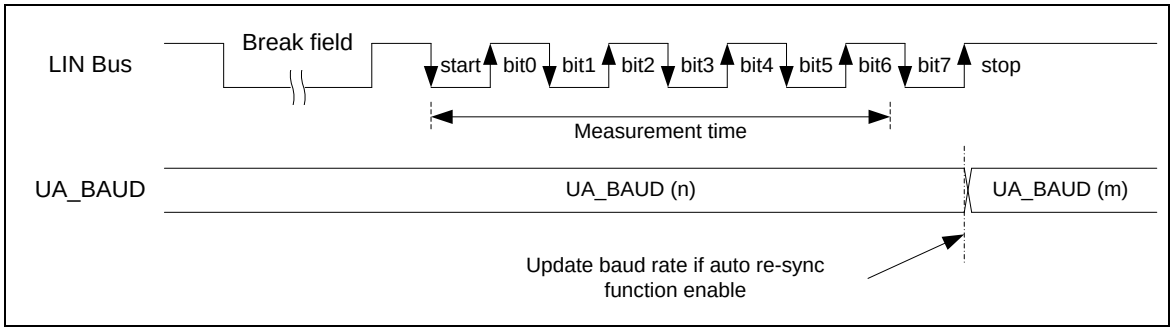



图 6-88 LIN同步域测量

自动重新同步模式(AR)下，软件必须通过设定UA_BAUD 寄存器设定希望的波特率，硬件将保存到内部的TEMP_REG 寄存器, 每次LIN break 域之后, 5个下降沿被采样，测量结果保存到内部 13-bit 寄存器 (BAUD_LIN) ，这个结果将自动更新到UA_BAUD 寄存器.

为了保证传输波特率,每个新的break域收到之前，波特率发生器必须重新加载初始值。初始值在初始化的时候由应用程序设定（TEMP_REG）。用户可以设定LINS_DUM_EN (UA_LIN_CTL[3])来使能自动加载初始波特率功能。如果LINS_DUM_EN (UA_LIN_CTL[3]) = 1, 当前帧结束，收到下一个字符之前，硬件将自动加载初始值到UA_BAUD寄存器，UA_BAUD 被更新之后, LINS_DUM_EN (UA_LIN_CTL[3])将被自动清0。LIN波特率的更新方法如下图所示

**注1:** 收到检验字符之前，推荐设定LINS_DUM_EN (UA_LIN_CTL[3])比特为1.

**注2:** 侦测到报头错误时, 用户需要对LINS_SYNC_F (UA_LIN_SR[3])写1来重新检测报头. 当写1到该位时, 硬件将重载初始波特率(TEMP_REG) 并重新搜索新的报头.

**注3:** 自动重新同步模式下，波特率必须设定为mode2 (DIV_X_EN (UA_BAUD[29]) 和 DIV_X_ONE (UA_BAUD[28]) 必须都为1)

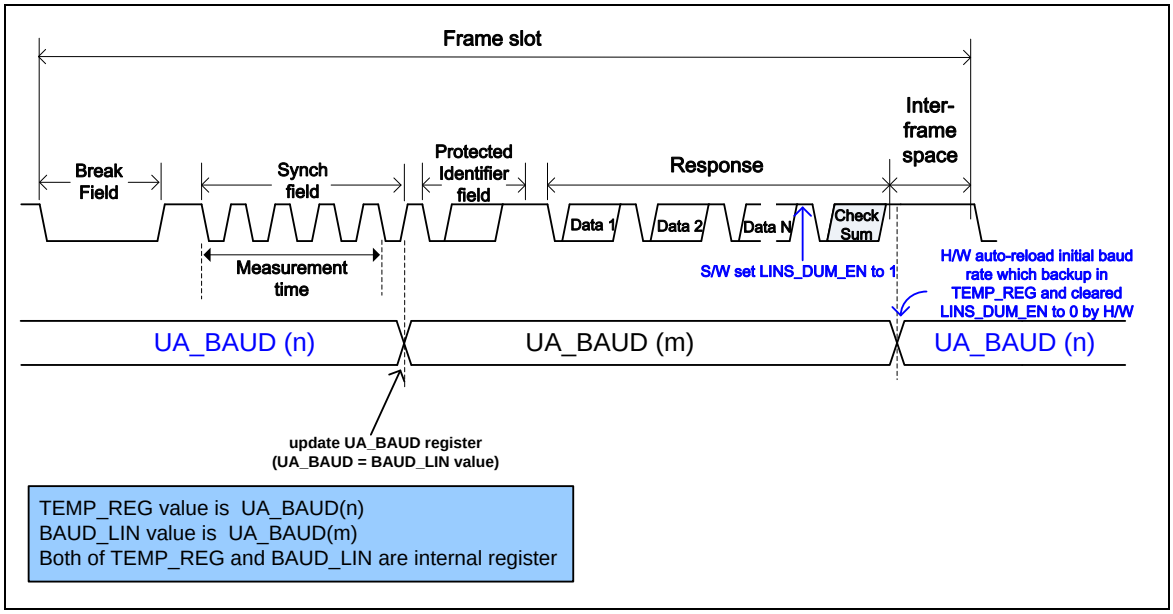



图 6-89 当LINS_DUM_EN (UA_LIN_CTL[3]) = 1，在自动重同步模式下UA_BAUD 更新顺序

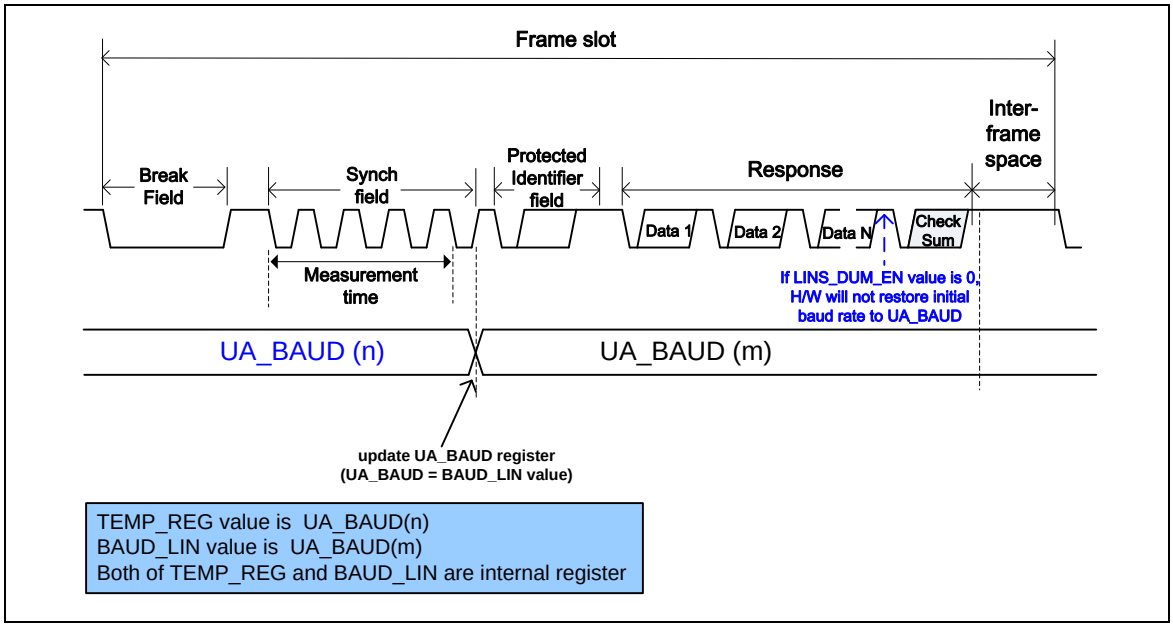



图 6-90 当LINS_DUM_EN (UA_LIN_CTL[3]) = 0，在自动重同步模式下UA_BAUD 更新顺序

**同步域误差错误**

自动重新同步模式下，控制器将检测同步域的误差错误。误差错误检测比较当前波特率和接收到的同步域的波特率。两个检测被同步执行.

检查1:根据同步域的第一个下降沿和最后一个下降沿的测量值

- 如果误差大于14.84%, 报头错误标志LINS_HERR_F (UA_LIN_SR[1])将被置位
- 如果误差小于14.06%, 报头错误标志LINS_HERR_F (UA_LIN_SR[1])不会被置位.
- 如果误差在14.84% 和 14.06%之间, 报头错误标志LINS_HERR_F (UA_LIN_SR[1])可能被置位也可能没有被置位 (取决于数据失相)

检查2: 根据同步域的每一个下降沿的测量值.

- 如果误差大于18.75%,报头错误标志LINS_HERR_F (UA_LIN_SR[1])将被置位
- 如果误差小于15.62%,报头错误标志LINS_HERR_F (UA_LIN_SR[1])不会被置位
- 如果误差在18.75% 和 15.62%之间, 报头错误标志LINS_HERR_F (UA_LIN_SR[1])可能被置位也可能没有被置位(取决于数据失相)

**注意:** 误差检测基于当前波特率时钟。因而，为了保证误差检测的正确性，通过设定LINS_DUM_EN (UA_LIN_CTL[3])，新的break域收到之前，波特率必须重新加载为初始值(每个checksum收到之前，推荐设定LINS_DUM_EN (UA_LIN_CTL[3])为1)

**LIN 报头错误侦测**

LIN 从机模式下, 当用户通过设定LINS_HDET_EN (UA_LIN_CTL[1])使能报头检测功能时, 硬件将处理报头检测流程。如果报头有错误，LIN 报头错误标志LINS_HERR_F (UA_LIN_SR[1])将被置位，如果LIN_IEN (UA_IER[8]) =1，中断将发生。报头错误被检测到时，用户必须写1到LINS_SYNC_F (UA_LIN_SR[3])来复位检测电路以重新检测新的报头

如果下列一个条件发生，LIN 报头错误标志LINS_HERR_F (UA_LIN_SR[1])将被置位

- Break分隔符太短 (小于 0.5 比特时间).
- 同步域或者帧ID域帧错误
- 同步域数据不是0x55 (非自动重新同步模式)
- 同步域误差错误(自动重新同步模式).
- 同步域测量超时 (自动重新同步模式)
- LIN报头接收超时

### 6.12.6 寄存器映射

**R**: 只读, **W**: 只写, **R/W**: 可读写

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UART 基地址:**<br>**UARTx_BA = 0x4005_0000 + (0x10_0000 * x)**<br>**x = 0, 1** | | | | |
| **UA_RBR** | UARTx_BA+0x00 | R | UART 接收缓存寄存器 | Undefined |

| | | | | |
|---|---|---|---|---|
| **UA_THR** | UARTx_BA+0x00 | W | UART 发送缓存寄存器 | Undefined |
| **UA_IER** | UARTx_BA+0x04 | R/W | UART 中断使能寄存器 | 0x0000_0000 |
| **UA_FCR** | UARTx_BA+0x08 | R/W | UART FIFO 控制寄存器 | 0x0000_0000 |
| **UA_LCR** | UARTx_BA+0x0C | R/W | UART线控寄存器 | 0x0000_0000 |
| **UA_MCR** | UARTx_BA+0x10 | R/W | UART Modem 控制寄存器 | 0x0000_0200 |
| **UA_MSR** | UARTx_BA+0x14 | R/W | UART Modem 状态寄存器 | 0x0000_0110 |
| **UA_FSR** | UARTx_BA+0x18 | R/W | UART FIFO状态寄存器 | 0x1040_4000 |
| **UA_ISR** | UARTx_BA+0x1C | R/W | UART 中断状态寄存器 | 0x0000_0002 |
| **UA_TOR** | UARTx_BA+0x20 | R/W | UART 定时溢出寄存器 | 0x0000_0000 |
| **UA_BAUD** | UARTx_BA+0x24 | R/W | UART波特率分频寄存器 | 0x0F00_0000 |
| **UA_IRCR** | UARTx_BA+0x28 | R/W | UART IrDA控制寄存器 | 0x0000_0040 |
| **UA_ALT_CSR** | UARTx_BA+0x2C | R/W | UART选择控制/状态寄存器 | 0x0000_000C |
| **UA_FUN_SEL** | UARTx_BA+0x30 | R/W | UART功能选择寄存器 | 0x0000_0000 |
| **UA_LIN_CTL** | UARTx_BA+0x34 | R/W | UART LIN 控制寄存器 | 0x000C_0000 |
| **UA_LIN_SR** | UARTx_BA+0x38 | R/W | UART LIN 状态寄存器 | 0x0000_0000 |

### 6.12.7 寄存器描述

**接收缓存寄存器 (UA_RBR)**

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_RBR** | UARTx_BA+0x00 | R | UART 接收缓存寄存器 | 未定义 |

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

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留. |
| [7:0] | **RBR** | **接收缓存寄存器 (只读)**<br>通过读该寄存器， UART将返回从UART_RXD引脚接收到的8位数据(LSB 优先). |

## 发送保持寄存器 (UA_THR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_THR** | UARTx_BA+0x00 | W | UART 发送保持寄存器 | 未定义 |

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

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留. |
| [7:0] | **THR** | **发送保持寄存器**<br>写数据到该寄存器, 数据将会保存到发送FIFO. UART控制器将会通过TX脚把存放在FIFO中最前面的数据通过UART_TXD引脚发送出去(LSB 优先). |

## 中断使能寄存器 (UA_IER)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_IER** | UARTx_BA+0x04 | R/W | UART 中断使能寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | **AUTO_CTS_EN** | **AUTO_RTS_EN** | **TIME_OUT_EN** | **Reserved** | | **LIN_IEN** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | **WAKE_EN** | **BUF_ERR_IEN** | **TOUT_IEN** | **MODEM_IEN** | **RLS_IEN** | **THRE_IEN** | **RDA_IEN** |

| 位 | 描述 | |
|---|---|---|
| [31:14] | **Reserved** | 保留. |
| [13] | **AUTO_CTS_EN** | **nCTS自动流控使能位**<br>0 = nCTS自动流控禁止.<br>1 = nCTS自动流控使能.<br>当nCTS 自动流控使能后，当nCTS输入有效， UART 会发送数据到外部设备（UART将不会发送数据到外部设备直到nCTS有效）。 |
| [12] | **AUTO_RTS_EN** | **nRTS 自动流控使能位**<br>0 = nRTS 自动流控禁止<br>1 = nRTS 自动流控使能<br>当nRTS 自动流控使能后，如果RX FIFO中字节的数量等于RTS_TRI_LEV (UA_FCR[19:16])，UART会自动禁止nRTS信号 |
| [11] | **TIME_OUT_EN** | **接收缓存超时计数器使能位**<br>0 =接收缓存超时计数器禁止<br>1 = 接收缓存超时计数器使能. |
| [10:9] | **Reserved** | 保留. |
| [8] | **LIN_IEN** | **LIN 总线中断使能位**<br>0 = LIN 总线中断禁止<br>1 = LIN 总线中断使能<br>**注:** 该位用于LIN 功能模式 |
| [7] | **Reserved** | 保留. |
| [6] | **WAKE_EN** | **UART 唤醒功能使能位**<br>0 = UART 唤醒功能禁止 |

| | | |
|---|---|---|
| | | 1 = UART 唤醒功能使能，当芯片在掉电模式，外部nCTS 引脚电平改变可以唤醒芯片 |
| [5] | **BUF_ERR_IEN** | **缓存错误中断使能**<br>0 = 缓冲区错误中断禁止<br>1 = 缓冲区错误中断使能 |
| [4] | **TOUT_IEN** | **RX超时中断使能位**<br>0 = RX 超时中断禁止<br>1 = RX 超时中断使能 |
| [3] | **MODEM_IEN** | **Modem 状态中断使能位**<br>0 = 关闭MODEM状态中断.<br>1 = 使能MODEM状态中断 |
| [2] | **RLS_IEN** | **接收 线状态中断使能位**<br>0 = 关闭接收线状态中断.<br>1 = 使能接收线状态中断. |
| [1] | **THRE_IEN** | **发送保持寄存器空中断使能位**<br>0 = 关闭发送数据寄存器为空时中断<br>1 = 使能发送数据寄存器为空时中断. |
| [0] | **RDA_IEN** | **接收数据有效中断使能位**<br>0 = 接收数据有效中断关闭<br>1 = 接收数据有效中断使能. |

## FIFO 控制寄存器 (UA_FCR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_FCR** | UARTx_BA+0x08 | R/W | UART FIFO 控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | **RTS_TRI_LEV** | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | **RX_DIS** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **RFITL** | | | | **Reserved** | **TFR** | **RFR** | **Reserved** |

| 位 | 描述 | |
|---|---|---|
| [31:20] | **Reserved** | 保留. |
| [19:16] | **RTS_TRI_LEV** | **nRTS 自动流控触发阈值**<br>0000 = nRTS 触发阈值为1 byte.<br>0001 = nRTS触发阈值为4 bytes.<br>0010 = nRTS触发阈值为8 bytes.<br>0011 = nRTS触发阈值为14 bytes.<br>其他 = 保留.<br>**注：** 该区域用于自动nRTS 流控制 |
| [15:9] | **Reserved** | 保留. |
| [8] | **RX_DIS** | **接收器禁止设置位**<br>是否禁用接收器（置 1 禁用接收器）<br>0 =接收器使能<br>1 =接收器禁止<br>**注：** 该位用于 RS-485 普通多点模式,需要在配置RS485_NMM (UA_ALT_CSR [8]) 之前设置. |
| [7:4] | **RFITL** | **RX FIFO 中断 (INT_RDA) 触发级别**<br>当FIFO 接收字节数等于 RFITL 后，RDA_IF 将被置位（如果RDA_IEN (UA_IER [0])使能，将产生中断）。<br>0000 = RX FIFO 触发中断阈值为 1 byte.<br>0001 = RX FIFO 触发中断阈值为4 bytes.<br>0010 = RX FIFO 触发中断阈值为8 bytes.<br>0011 = RX FIFO 触发中断阈值为14 bytes.<br>其它 = 保留. |
| [3] | **Reserved** | 保留. |

| | | |
|---|---|---|
| [2] | **TFR** | **TX 域软件复位**<br>当TFR置位， 发送 FIFO和 TX 内部状态机中的所有数据将被清除。<br>0 =不影响.<br>1 =该位写 1 将复位 TX 内部状态机和指针<br>**注：**该位至少 3个 UART 时钟周期后会自动清零 |
| [1] | **RFR** | **RX 域软件复位**<br>当RFR被置位， 接收 FIFO和 RX 内部状态机中的所有数据将被清除。<br>0 = 不影响.<br>1 =该位写 1 将复位 RX 内部状态机和指针.<br>**注：**该位至少 3个 UART 时钟周期后会自动清零。 |
| [0] | **Reserved** | 保留. |

## 线控寄存器(UA_LCR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_LCR** | UARTx_BA+0x0C | R/W | UART 线控寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | **BCB** | **SPE** | **EPE** | **PBE** | **NSB** | **WLS** | |

| 位 | 描述 | |
|---|---|---|
| [31:7] | **Reserved** | 保留. |
| [6] | **BCB** | **Break 控制位**<br>0 = Break 控制禁止.<br>1 = Break 控制使能.<br>**注:** 当该位被置逻辑 1，串行数据输出 (TX) 将强制到 Spacing 状态 (逻辑 0)。该位仅作用于 TX ，对传输逻辑不起作用。 |
| [5] | **SPE** | **Stick校验位使能**<br>0 = 禁用 Stick 校验<br>1 = 使能 Stick 校验<br>注: 如果PBE (UA_LCR[3]) 和 EBE (UA_LCR[4])为逻辑 1，校验位发送和检验值为逻辑 0。如果PBE (UA_LCR[3])是1 ，EPE (UA_LCR[4])是 0，则校验位发送和检验值为 1。 |
| [4] | **EPE** | **偶校验使能**<br>0 = 奇数个逻辑 1在每个字节中被发送和检验<br>1 = 偶数个逻辑 1在每个字中被发送和检验<br>该位只在PBE (UA_LCR[3])置位时有效。 |
| [3] | **PBE** | **校验位使能**<br>0 = 禁止生成校验位<br>1 = 使能生成校验位<br>注: 每一个发送字符中都产生校验位，对每一个传进来的数据进行校验位检测 |
| [2] | **NSB** | **“停止位”个数**<br>0= 当发送数据时，产生 1 个“ STOP bit”<br>1= 当发送数据，选择 5-位 字长度时，产生 1.5 “ STOP bit” 当选择 6-, 7- 和 8-位字长度时，产生 2 个“STOP bit” |
| [1:0] | **WLS** | **字长度选择** |

| | | |
|---|---|---|
| | | 00 =字长是 5-bit.<br>01 =字长是6-bit.<br>10 =字长是7-bit.<br>11 =字长是8-bit. |

## MODEM 控制寄存器 (UA_MCR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_MCR** | UARTx_BA+0x10 | R/W | UART Modem 控制寄存器 | 0x0000_0200 |

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

| 位 | 描述 | |
|---|---|---|
| [31:14] | **Reserved** | 保留. |
| [13] | **RTS_ST** | **nRTS 脚状态(只读)**<br>该位的值对应于nRTS脚的输出电平<br>0 = nRTS脚为低电平逻辑状态<br>1 = nRTS脚为高电平逻辑状态. |
| [12:10] | **Reserved** | 保留. |
| [9] | **LEV_RTS** | **nRTS 脚的有效电平**<br>该位定义了nRTS输出脚的有效电平<br>0 = nRTS 脚输出为高电平有效.<br>1 = nRTS 脚输出为低电平有效.（默认）<br>**注意1:** UART功能模式请参考图 6-78和图 6-79<br>**注意2:** RS-485 功能模式请参考图 6-82 和 图 6-83 |
| [8:2] | **Reserved** | 保留. |
| [1] | **RTS** | **nRTS (请求发送) 信号控制**<br>该位直接控制内部nRTS脚信号是否有效，然后使用LEV_RTS位的配置驱动nRTS脚输出<br>0 = nRTS 信号有效.<br>1 = nRTS 信号无效.<br>**注1:** UART 功能模式下，当nRTS自动流控被使能后，nRTS信号控制位无效<br>**注2:** RS-485模式下，当RS-485自动方向模式（AUD）被使能后，nRTS信号控制位无效 |
| [0] | **Reserved** | 保留. |

## Modem 状态寄存器 (UA_MSR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_MSR** | UARTx_BA+0x14 | R/W | UART Modem 状态寄存器 | 0x0000_0110 |

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

| 位 | 描述 | |
|---|---|---|
| [31:9] | **Reserved** | 保留. |
| [8] | **LEV_CTS** | **nCTS 脚有效电平**<br>该位定义了nCTS输入脚的有效电平<br>0 = nCTS输入脚高电平有效<br>1 = nCTS输入脚低电平有效.(默认设置) |
| [7:5] | **Reserved** | 保留. |
| [4] | **CTS_ST** | **nCTS 脚状态(只读)**<br>该位对应于nCTS脚输入逻辑状态<br>0 = nCTS脚输入状态为低电平<br>1 = nCTS脚输入状态为高电平<br>**注意:**当UART控制器外设时钟被使能，且nCTS管脚功能被使能，该位才有效。 |
| [3:1] | **Reserved** | 保留. |
| [0] | **DCTSF** | **检测到nCTS 引脚电平改变标志**<br>如果nCTS输入脚上有电平变化，该位将被置位，如果MODEM_IEN (UA_IER[3])位被置位，将会产生Modem中断到CPU。<br>0 = nCTS 输入脚没有电平变化.<br>1 = nCTS输入脚有电平变化.<br>**注意:**写 “1” 清零. |

## FIFO 状态寄存器 (UA_FSR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_FSR** | UARTx_BA+0x18 | R/W | UART FIFO 状态寄存器 | 0x1040_4000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | TE_FLAG | Reserved | | | TX_OVER_IF |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| TX_FULL | TX_EMPTY | TX_POINTER | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| RX_FULL | RX_EMPTY | RX_POINTER | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Reserved | BIF | FEF | PEF | RS485_ADD_DETF | Reserved | | RX_OVER_IF |

| 位 | 描述 | |
|---|---|---|
| [31:29] | **Reserved** | 保留. |
| [28] | **TE_FLAG** | **发送空标志(只读)**<br>当TX FIFO (UA_THR)为空,并且最后一个字节的STOP位也已经被发送完毕，该位被硬件置1.<br>0 = TX FIFO 不为空或最后一个字节的STOP位还没有发送.<br>1 = TX FIFO 为空而且最后一个字节的STOP位已经发送.<br>**注**：当TX FIFO不为空或最后一个字节的STOP位还没有被发送完毕，那么该位将被自动清零. |
| [27:25] | **Reserved** | 保留. |
| [24] | **TX_OVER_IF** | **TX溢出错误中断标志**<br>如果TX FIFO (UA_THR)满，如果再向UA_THR写入数据，将会导致此位被置1.<br>0 = TX FIFO 未溢出<br>1 = TX FIFO 已溢出.<br>**注**:写1可清除该位 |
| [23] | **TX_FULL** | **发送FIFO满(只读)**<br>此位用于指示TX FIFO是否已满.<br>0 = TX FIFO 未满<br>1 = TX FIFO 已满<br>**注**:当TX FIFO 缓存中使用的字节数量等于16,此位将被置1.否则被硬件清零. |
| [22] | **TX_EMPTY** | **发送FIFO 空(只读)**<br>此位指示TX FIFO是否为空。<br>0 = TX FIFO不为空.<br>1 = TX FIFO为空.<br>**注意**：当TX FIFO中的最后一个字节被发送到发送移位寄存器，硬件将把此位置1.当写入 |

| | | |
|---|---|---|
| | | 数据到UA_THR中, (TX FIFO不为空)，此位被清零 |
| [21:16] | **TX_POINTER** | **TX FIFO 指针(只读)**<br>此位指示TX FIFO缓冲区指针位置。当CPU写一个字节到UA_THR寄存器，TX_POINTER将累加1.当TX FIFO发送一个字节到发送移位寄存器中，TX_POINTER指针将减1.<br>TX_POINTER显示的最大值是15。当TX FIFO缓冲区所填充数据数量达到16，TXFULL将被置1，TX_POINTER被清零。如果TX FIFO中发送一个字节到发送移位寄存器，TXFULL位将被清零，TX_POINTER显示15。 |
| [15] | **RX_FULL** | **接收 FIFO 满(只读)**<br>该位指示RX FIFO 是否已满。<br>0 = RX FIFO 未满<br>1 = RX FIFO 已满<br>**注意：**当RX FIFO缓冲区中的数据数量等于16后，此位将被置1，否则被硬件清零。 |
| [14] | **RX_EMPTY** | **接收 FIFO空(只读)**<br>此位指示RX FIFO是否为空。<br>0 = RX FIFO不为空.<br>1 = RX FIFO为空.<br>**注意：**当RX FIFO中最后一个字节被CPU读取后，硬件将对此位置1，UART接收到新数据后此位将被清零。 |
| [13:8] | **RX_POINTER** | **RX FIFO指针(只读)**<br>此位指示RX FIFO缓冲区指针。当UART从外部设备接收到一个字节，RX_POINTER将累加1.当RX FIFO的数据被CPU读取一个字节，RX_POINTER将递减1.<br>RX_POINTER显示的最大值是15。当RX FIFO的数据达到16，RXFULL位将被置1，RX_POINTER显示0，RX FIFO 当中的数据被CPU读取一个后，RXFULL 将被清零，RX_POINTER显示15。 |
| [7] | **Reserved** | 保留. |
| [6] | **BIF** | **Break 错误中断标志位**<br>每当接收到数据输入 (RX) 维持在 “spacing 状态” (逻辑 0) 的时间长于一个全字的传输时间（即“start bit” + data bits + parity + stop bits 的总时间），该位置 1。<br>0 =没有Break错误中断产生.<br>1 =有Break错误中断产生.<br>**注意：**此位写1清零 |
| [5] | **FEF** | **帧错误标志**<br>每当接收到的字符没有有效的 “停止位”（即跟在最后的数据位后或奇偶校验位后的数据为 0）该位置 1。<br>0 =没有帧错误产生<br>1 =有帧错误产生.<br>**注：**该位写 1 清零。 |
| [4] | **PEF** | **奇偶校验错误标志**<br>每当接收到的字符没有有效的奇偶校验位，该位置 1。<br>0 =没有奇偶校验错误产生<br>1 =有奇偶校验错误产生<br>**注：**该位写 1 清零。 |
| [3] | **RS485_ADD_DETF** | **RS-485 地址字节检测标志**<br>0 =接收到的数据没有地址位标记 (bit 9 =’0’). |

| | | |
|---|---|---|
| | | 1 =接收到的数据有地址位标记(bit 9 ='1').<br>**注1**：此位用于RS-485功能模式，且RS485_ADD_EN (UA_ALT_CSR[15])位被置1使能地址检测模式<br>**注2**：该位写 1 清零。 |
| [2:1] | **Reserved** | 保留. |
| [0] | **RX_OVER_IF** | **RX 溢出错误中断标志**<br>当RX FIFO溢出时被置1<br>如果接收到的数据数量大于RX_FIFO (UA_RBR) 16 字节，该位将被置位.<br>0 = RX FIFO未溢出.<br>1 = RX FIFO溢出<br>**注意**：该位写 1 清零 |

## 中断状态寄存器 (UA_ISR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_ISR** | UARTx_BA+0x1C | R/W | UART 中断状态寄存器 | 0x0000_0002 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **LIN_INT** | **Reserved** | **BUF_ERR_INT** | **TOUT_INT** | **MODEM_INT** | **RLS_INT** | **THRE_INT** | **RDA_INT** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **LIN_IF** | **Reserved** | **BUF_ERR_IF** | **TOUT_IF** | **MODEM_IF** | **RLS_IF** | **THRE_IF** | **RDA_IF** |

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留. |
| [15] | **LIN_INT** | **LIN总线中断提示(只读)**<br>如果 LIN_IEN (UA_IER[8]) 和 LIN _IF (UA_ISR[7])都被置1，该位置 1<br>0 =没有LIN总线中断产生<br>1 =有LIN总线中断产生. |
| [14] | **Reserved** | 保留. |
| [13] | **BUF_ERR_INT** | **Buffer错误中断提示(只读)**<br>如果BUF_ERR_IEN (UA_IER[5]) 和 BUF_ERR_IF (UA_ISR[5])都被置1，该位置 1<br>0 =没有buffer错误中断产生.<br>1 =有buffer错误中断产生 |
| [12] | **TOUT_INT** | **定时溢出中断提示(只读)**<br>如果TOUT_IEN (UA_IER[4]) 和 TOUT_IF (UA_ISR[4])都被置1，该位置 1<br>0 =没有定时溢出中断产生<br>1 =有定时溢出中断产生. |
| [11] | **MODEM_INT** | **MODEM状态中断提示(只读)**<br>如果MODEM_IEN (UA_IER[3]) 和 MODEM_IF (UA_ISR[3])都被置1，该位置 1<br>0 =没有Modem 中断产生.<br>1 =有Modem 中断产生.. |
| [10] | **RLS_INT** | **接收线状态中断提示(只读)**<br>如果RLS_IEN (UA_IER[2] 和 RLS_IF (UA_ISR[2])都被置1，该位置 1<br>0 =没有RLS中断产生<br>1 =有RLS中断产生. |
| [9] | **THRE_INT** | **发送保持寄存器空中断提示(只读)**<br>如果THRE_IEN (UA_IER[1]) 和 THRE_IF (UA_ISR[1])都被置1，该位置 1 |

| | | |
|---|---|---|
| | | 0 =没有THRE中断产生<br>1 =有THRE中断产生 |
| [8] | **RDA_INT** | **接收有效数据中断提示(只读)**<br>如果RDA_IEN (UA_IER[0]) 和 RDA_IF (UA_ISR[0])都被置1，该位置 1<br>0 =没有RDA中断产生.<br>1 =有RDA中断产生. |
| [7] | **LIN_IF** | **LIN总线标志(只读)**<br>当LIN从机报头侦测(LINS_HDET_F (UA_LIN_SR[0]) =1)，LIN break 侦测(LIN_BKDET_F (UA_LIN_SR[8]) = 1)，位错误侦测(BIT_ERR_F (UA_LIN_SR[9]) = 1)，LIN从机ID校验错误(LINS_IDPERR_F (UA_LIN_SR[2]) = 1)或LIN从机报头错误侦测(LINS_HERR_F (UA_LIN_SR[1]) = 1)，该位置1。如果LIN_ IEN (UA_IER [8])被使能，LIN中断产生。<br>0 = LINS_HDET_F, LIN_BKDET_F, BIT_ERR_F, LINS_IDPERR_F 和 LINS_HERR_F中没有任何一个标志产生<br>1 = LINS_HDET_F, LIN_BKDET_F, BIT_ERR_F, LINS_IDPERR_F 和 LINS_HERR_F中至少有一个标志产生<br>**注:**该位只读。当LINS_HDET_F (UA_LIN_SR[0]), LIN_BKDET_F (UA_LIN_SR[8]), BIT_ERR_F (UA_LIN_SR[9]), LINS_IDPENR_F (UA_LIN_SR[2]) 和 LINS_HERR_F (UA_LIN_SR[1])都被清0时，该位清0。 |
| [6] | **Reserved** | 保留. |
| [5] | **BUF_ERR_IF** | **Buffer 错误中断标志 (只读)**<br>当TX FIFO或RX FIFO溢出(TX_OVER_IF (UA_FSR[24]) 或 RX_OVER_IF (UA_FSR[0]) 为1)该位置1. 当BUF_ERR_IF (UA_ISR[5])被置位, 传输出错.如果BUF_ERR_IEN (UA_IER [5])被使能, buffer 错误中断将产生。<br>0 = 没有buffer 错误中断标志产生<br>1 = 有buffer 错误中断标志产生<br>**注**：当写1到RX_OVER_IF (UA_FSR[0]) 和 TX_OVER_IF (UA_FSR[24])，将 RX_OVER_IF (UA_FSR[0]) 和 TX_OVER_IF (UA_FSR[24])清零时，该位也将被清零。 |
| [4] | **TOUT_IF** | **定时溢出中断标志 (只读)**<br>当 RX FIFO非空而且 RX FIFO无新的数据，定时溢出计数器等于TOIC (UA_TOR[7:0]). 时，该位置位。如果TOUT_IEN (UA_IER [4])使能，定时溢出中断将产生。<br>0 =没有定时溢出中断标志产生.<br>1 =有定时溢出中断标志产生<br>**注**：用户可以读UART_ RBR（RX处于活动状态）清除该位 |
| [3] | **MODEM_IF** | **MODEM 中断标志(只读)**<br>当n CTS管脚状态有改变(DCTSF (UA_MSR[0]) = 1)，该位置位。如果MODEM_IEN (UA_IER [3])使能，Modem 中断将产生。<br>0 =没有Modem 中断标志产生.<br>1 =有Modem 中断标志产生.<br>**注**：当向DCTSF (UA_MSR[0]).写1将DCTSF (UA_MSR[0]).清零后，该位也复位为0。 |
| [2] | **RLS_IF** | **接收线中断标志(只读)**<br>当 RX 接收数据发生校验错误, 帧错误 或 break 错误（至少BIF (UA_FSR[6]), FEF (UA_FSR[5]) 和 PEF (UA_FSR[4]),三位中有一位被置位）该位置位。如果RLS_IEN (UA_IER [2])使能，RLS 中断将产生。<br>0 =没有RLS中断标志产生<br>1 =有RLS中断标志产生.<br>**注意1:** 在 RS-485功能模式，当“接收检测到任一的地址字节字符 (bit9 = ‘1’) 位”时，该位也会置1，同时，RS485_ADD_DETF (UA_FSR[3])位也被置1 |

| | | |
|---|---|---|
| | | **注意2:** 该位只读，当BIF (UA_FSR[6]), FEF (UA_FSR[5]) 和 PEF (UA_FSR[4])三位都被清除后，该位重置为 0<br>**注意3:** 在RS-485 功能模式下,该位只读，且当BIF (UA_FSR[6]) , FEF(UA_FSR[5]) 和 PEF(UA_FSR[4]) 和 RS485_ADD_DETF (UA_FSR[3])所有位被清零，该位清零。 |
| [1] | **THRE_IF** | **发送保持寄存器空中断标志(只读)**<br>当TX FIFO中最后数据传输到发送移位寄存器时，该位置位。如果THRE_IEN (UA_IER [1])使能，THRE 中断将产生。<br>0 =没有THRE中断标志产生<br>1 =有THRE中断标志产生.<br>**注意:** 该位只读，当写数据到THR（TX FIFO 非空）时，该位将被清除 |
| [0] | **RDA_IF** | **接收可用数据中断标志(只读)**<br>当 RX FIFO 中的数据数量等于RFITL (UA_FCR[7:4])时，RDA_IF (UA_ISR[0])将被置位。如果RDA_IEN (UA_IER [0])使能，RDA 中断将产生。<br>0 =没有RDA中断标志产生.<br>1 =有RDA中断标志产生.<br>**注**：该位只读，当 RX FIFO 的未读数据低于阈值RFITL (UA_FCR[7:4]).时，该位将被清除 |

## 超时寄存器 (UA_TOR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_TOR** | UARTx_BA+0x20 | R/W | UART 超时寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留. |
| [15:8] | **DLY** | **TX延迟时间值**<br>该域用于设置上一次停止位和下一次开始位之间的传输延迟时间 |
| [7:0] | **TOIC** | **定时溢出中断比较器**<br>如果设置TIME_OUT_EN (UA_IER[11])使能定时溢出计数器，当RX FIFO接收到一个新的数据字，定时溢出计数器重置并开始计数 (计数时钟 = 波特率)。一旦超时计数器的内容等于超时中断比较器(TOIC (UART_TOUT[7:0]))，如果此时TOUT_IEN (UA_IER [4])位使能，则接收超时中断(TOUT_INT (UA_ISR[12]))产生。接收到新的数据或RX FIFO为空将把TOUT_IF (UA_ISR[4]).清零。为了避免接收超时中断在接收到一个字符就立即产生，TOIC的值必须设置在 40 到 255 之间。例如，如果TOIC为 40，当 UART 传输设置为1 位停止位且无奇偶校验位时，在4个字符时间长度后还没收到数据，超时中断将产生。 |

## 波特率分频寄存器(UA_BAUD)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_BAUD** | UARTx_BA+0x24 | R/W | UART波特率分频寄存器 | 0x0F00_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | **DIV_X_EN** | **DIV_X_ONE** | **DIVIDER_X** | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **BRD** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **BRD** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:30] | **Reserved** | 保留. |
| [29] | **DIV_X_EN** | **分频器 X 使能位**<br>BRD = 波特率分频器，波特率公式是：<br>波特率 = Clock / [M * (BRD + 2)]; M 默认值是 16.<br>0 = 分频器 X 禁止 (公式中 M = 16).<br>1 = 分频器 X 使能 (公式中 M = X+1, 但是 DIVIDER_X [27:24] 必须 >= 8).<br>**注意1:** 具体的描述如**UART 控制器波特率产生** 部分所述<br>**注意2:** 在 IrDA 模式, 该位必须禁止. |
| [28] | **DIV_X_ONE** | **分频器 X 等于 1**<br>0 = 分频器 M = X+1 (公式中 M = X+1, 但是 DIVIDER_X[27:24] 必须 >= 8).<br>1 = 分频器 M = 1.<br>**注意:** 具体的描述如**UART 控制器波特率产生** 部分所述. |
| [27:24] | **DIVIDER_X** | **分频器 X**<br>波特率分频器 M = X+1.<br>**注意 1:** T 该域用于模式1的波特率计算，对于模式0和模式2是无效的。.<br>**注意 2:** 具体的描述如**UART 控制器波特率产生** 部分所述 |
| [23:16] | **Reserved** | 保留. |
| [15:0] | **BRD** | **波特率分频器**.<br>该域是表示波特率分频器，该域用于波特率计算。<br>**注意:** 具体的描述如**UART 控制器波特率产生** 部分所述. |

## IrDA 控制寄存器 (UA_IRCR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_IRCR** | UARTx_BA+0x28 | R/W | UART IrDA 控制寄存器 | 0x0000_0040 |

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

| 位 | 描述 | |
|---|---|---|
| [31:7] | **Reserved** | 保留. |
| [6] | **INV_RX** | **IrDA接收输入信号反向**<br>0 =接收输入信号不反向.<br>1 =接收输入信号反向（默认）. |
| [5] | **INV_TX** | **IrDA发送信号反向**<br>0 =发送信号不反向（默认）.<br>1 =发送信号反向 |
| [4:2] | **Reserved** | 保留. |
| [1] | **TX_SELECT** | **IrDA 接收/发送选择使能位**<br>0 = IrDA发送禁止，接收使能（默认）<br>1 = IrDA发送使能，接收禁止 |
| [0] | **Reserved** | 保留. |

## UART复用控制/状态寄存器(UA_ALT_CSR)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_ALT_CSR** | UARTx_BA+0x2C | R/W | UART复用控制/状态寄存器 | 0x0000_000C |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **ADDR_MATCH** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **RS485_ADD_EN** | **Reserved** | | | | **RS485_AUD** | **RS485_AAD** | **RS485_NMM** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **LIN_TX_EN** | **LIN_RX_EN** | **Reserved** | | **LIN_BKFL** | | | |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **ADDR_MATCH** | **地址匹配值**<br>该位包括 RS-485 地址匹配值.<br>**注：**该域用于 RS-485 自动地址检测模式 |
| [23:16] | **Reserved** | 保留. |
| [15] | **RS485_ADD_EN** | **RS-485 地址检测使能位**<br>该位用于使能 RS-485 地址检测模式.<br>0 =地址检测模式禁止.<br>1 =地址检测模式使能.<br>**注：**该位适用于 各种RS-485 操作模式 |
| [14:11] | **Reserved** | 保留. |
| [10] | **RS485_AUD** | **RS-485 自动方向模式 (AUD)**<br>0 =RS-485自动方向操作模式（AUD)禁止.<br>1 =RS-485自动方向操作模式（AUD)使能<br>**注：**仅在RS-485_AAD 或 RS-485_NMM 操作模式有效 |
| [9] | **RS485_AAD** | **RS-485 自动地址检测操作模式 (AAD)**<br>0 = RS-485自动地址方向操作模式 (AAD)禁止.<br>1 =RS-485自动地址方向操作模式 (AAD)使能.<br>**注：**在 RS-485_NMM 操作模式下无效 |
| [8] | **RS485_NMM** | **RS-485 普通多点操作模式 (NMM)**<br>0 = RS-485 普通多点操作模式 (NMM)禁止.<br>1 = RS-485 普通多点操作模式 (NMM)使能<br>**注：**在 RS-485_AAD 操作模式下无效. |

| | | |
|---|---|---|
| [7] | **LIN_TX_EN** | **LIN TX Break 模式使能**<br>0 = LIN TX Break 模式禁止<br>1 = LIN TX Break 模式使能.<br>**注**：当 TX break 域传输操作完成后，该位将被自动清除. |
| [6] | **LIN_RX_EN** | **LIN RX 使能**<br>0 =LIN RX 模式禁止.<br>1 = LIN RX 模式使能 |
| [5:4] | **Reserved** | 保留. |
| [3:0] | **LIN_BKFL** | **UART LIN Break 域长度**<br>该域表示一个 4-位 LIN TX break 域数量<br>**注1:** 该break 域长度为UA_LIN_BKFL + 1.<br>**注2:** 根据LIN 规范, 复位值是0xC (break 域长度= 13). |

## UART 功能选择寄存器(UA_FUN_SEL)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_FUN_SEL** | UARTx_BA+0x30 | R/W | UART功能选择寄存器 | 0x0000_0000 |

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

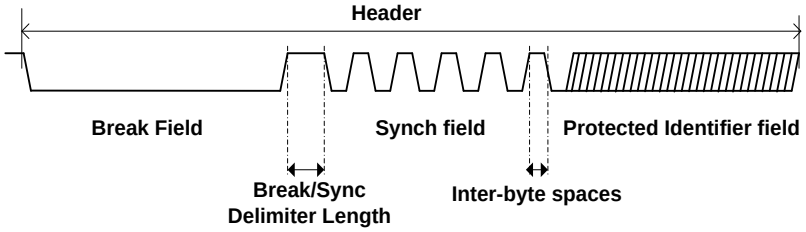
| 位 | 描述 | |
|---|---|---|
| [31:2] | **Reserved** | 保留. |
| [1:0] | **FUN_SEL** | **功能选择**<br>00 = UART功能使能.<br>01 = LIN 功能使能<br>10 = IrDA功能使能<br>11 = RS-485功能使能 |

## UART LIN控制寄存器(UA_LIN_CTL)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_LIN_CTL** | UARTx_BA+0x34 | R/W | UART LIN 控制寄存器 | 0x000C_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| LIN_PID | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| LIN_HEAD_SEL | | LIN_BS_LEN | | LIN_BKFL | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| Reserved | | | BIT_ERR_EN | LIN_RX_DIS | LIN_BKDET_EN | LIN_IDPEN | LIN_SHD |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Reserved | | | LIN_MUTE_EN | LINS_DUM_EN | LINS_ARS_EN | LINS_HDET_EN | LINS_EN |

| 位 | 描述 | |
|---|---|---|
| [31:24] | **LIN_PID** | **LIN PID 寄存器**<br>在LIN功能模式下，该域包含LIN帧ID值。帧ID的校验可以由软件或硬件产生，这取决于LIN_IDPEN (UA_LIN_CTL[9])的设置。<br>如果校验由硬件产生， 用户填ID0~ID5, (LIN_PID[24:29]，硬件会计算P0(LIN_PID[30]) 和P1(LIN_PID[31])，否则用户必须填帧ID和校验位<br>PID: Start \| ID0 \| ID1 \| ID2 \| ID3 \| ID4 \| ID5 \| P0 \| P1<br>P0 = ID0 xor ID1 xor ID2 xor ID4<br>P1 = ~(ID1 xor ID3 xor ID4 xor ID5)<br>**注1:** 用户可以填任何 8-bit 值到该域，位 24 表示 ID0 (LSB 优先)<br>**注2:** 该域可以用在 LIN 主机模式或从机模式 |
| [23:22] | **LIN_HEAD_SEL** | **LIN 帧报头选择**<br>00 = LIN帧报头包括 “break 域”<br>01 = LIN 帧报头包括 “break 域” 和 “同步域””.<br>10 = LIN 帧报头 包括 “break域”, “同步域” 域“帧ID域”.<br>11 = 保留<br>**注意：** LIN 主机模式时，该位用于发送帧报头域LIN_SHD (UA_LIN_CTL [8]) = 1)，而在LIN从机模式时，用于指示从mute模式退出条件(LIN_MUTE_EN (UA_LIN_CTL[4] = 1)。 |
| [21:20] | **LIN_BS_LEN** | **LIN Break/同步分隔符长度**<br>00 = LIN break/同步分隔符长度为1 bit时间.<br>10 = LIN break/同步分隔符长度为2 bit时间.<br>10 = LIN break/同步分隔符长度为3 bit时间.<br>11 = LIN break/同步分隔符长度为4 bit时间. |

| | | |
|---|---|---|
| | |  <br>**注:** 该位用于LIN 主机发送帧报头域 |
| [19:16] | **LIN_BKFL** | **LIN Break 域长度**<br>该域表示一个4-bit LIN TX break域数量<br>**注1:** 这个寄存器是LIN_BKFL (UA_ALT_CSR[19:16])的影子寄存器，用户可以通过设置LIN_BKFL (UA_ALT_CSR[3:0])或LIN_BKFL (UA_LIN_CTL[19:16])对break域的长度进行读/写。<br>**注2**: 该 break 域长度为LIN_BKFL + 1<br>**注3:** 根据LIN 规范，复位值是0XC (break域长度= 13). |
| [15:13] | **Reserved** | 保留. |
| [12] | **BIT_ERR_EN** | **位错误侦测使能**<br>0 =位错误侦测功能禁止.<br>1 =位错误侦测使能.<br>**注:** 在 LIN功能模式下，当位错误发生时，BIT_ERR_F (UA_LIN_SR[9])标志会被置位。如果LIN_IEN (UA_IER[8]) = 1，会产生一个中断 |
| [11] | **LIN_RX_DIS** | **LIN接收器禁止位**<br>如果接收器使能(LIN_RX_DIS (UA_LIN_CTL[11]) = 0)，所有接收到的数据字节会被接受并存储在RX-FIFO，如果接收器禁止(LIN_RX_DIS (UA_LIN_CTL[11]) = 1)，接收到的所有数据会被忽略<br>0 = LIN 接收器禁止.<br>1 = LIN 接收器使能.<br>**注:** 该位仅工作在 LIN功能模式下有效(FUN_SEL (UA_FUN_SEL[1:0]) = 01). |
| [10] | **LIN_BKDET_EN** | **LIN Break 域检测使能位**<br>当检测到显性位长度连续超过11位，后跟一个分隔符，在break域结束时LIN_BKDET_F (UA_LIN_SR[8])标志被置位。如果LIN_IEN (UA_IER[8]) =1，将会产生中断。<br>0 = LIN break 域检测禁止<br>1 = LIN break 域检测使能. |
| [9] | **LIN_IDPEN** | **LIN ID 校验使能位**<br>0 = LIN 帧ID校验禁止.<br>1 = LIN 帧ID校验使能.<br>**注1:** 该位可以用于LIN主机发送帧头域(LIN_SHD (UA_LIN_CTL[8]) = 1 和 LIN_HEAD_SEL (UA_LIN_CTL[23:22]) = 10)或用来使能 LIN 从机对接收到的帧 ID进行奇偶校验检查。<br>**注2:**该位仅在发送器报头是在LIN_HEAD_SEL (UA_LIN_CTL[23:22]) = 10时有用。 |
| [8] | **LIN_SHD** | **LIN TX 发送帧报头使能位**<br>LIN TX 帧报头可以是“break 域” 或 “break 和同步域” 或 “break, 同步和帧 ID 域”，这取决于LIN_HEAD_SEL (UA_LIN_CTL[23:22]).的设置<br>0 = LIN TX 帧报头发送禁止.<br>1 = LIN TX 帧报头发送使能.<br>**注1:** 这些 寄存器是LIN_TX_EN (UA_ALT_CSR[7])的影子寄存器;用户可以通过LIN_TX_EN |

| | | |
|---|---|---|
| | | (UA_ALT_CSR[7])或LIN_SHD (UA_LIN_CTL[8]).进行读写设置<br>**注2:** 当发送器帧头域 (可能是 “break” 或 “break + 同步” 或 “break + 同步 + 帧 ID”通过 LIN_HEAD_SEL (UA_LIN_CTL[23:22])选择) 传输操作完成, 该位会自动清0. |
| [7:5] | **Reserved** | 保留. |
| [4] | **LIN_MUTE_EN** | **LIN Mute 模式使能位**<br>0 = LIN Mute 模式禁止<br>1 = LIN Mute 模式使能.<br>**注意:** 退出Mute模式条件，该域的控制和相互影响在LIN从机模式中有详细解释 |
| [3] | **LINS_DUM_EN** | **LIN从机分频器更新方式使能**<br>0 = UA_BAUD 被软件更新(如果同时没有自动重同步发生)<br>1 = UA_BAUD在下次接收到字符的时候更新。用户必须在接收校验之前设置该位.<br>**注1:** 该位仅在 LIN 从模式有效(LINS_EN (UA_LIN_CTL[0]) = 1).<br>**注2**: 该位用于LIN 从机自动重同步模式. (对于非自动重同步模式，该位需维持清零)<br>**注3:** 该域的控制和相互作用在章节6.12.5.9.4中解释(带自动重同步的从机模式) |
| [2] | **LINS_ARS_EN** | **LIN 从机自动重同步模式使能位**<br>0 = LIN从机自动重同步模式禁止<br>1 = LIN从机自动重同步模式使能.<br>**注意1:** 该位仅在 LIN 从机模式(LINS_EN (UA_LIN_CTL[0]) = 1).有效.<br>**注意2:** 当工作在自动重同步模式，波特率的设置必须是模式2 (DIV_X_EN (UA_BAUD [29]) 和 DIV_X_ONE (UA_BAUD [28])必须是 1).<br>**注意3:** 该域的控制和相互作用在章节6.12.5.9.4带自动重同步的从机模式)中解释 |
| [1] | **LINS_HDET_EN** | **LIN从机报头侦测使能位**<br>0 = LIN 从机报头检测禁止.<br>1 = LIN 从机报头检测使能.<br>**注1:** 该位仅在 LIN 从机模式(LINS_EN (UA_LIN_CTL[0]) = 1).有效.<br>**注2:** 在 LIN功能模式， 当侦测报头域(break + sync + frame ID), LINS_HDET_F (UA_LIN_SR [0])标志会被置位，如果LIN_IEN (UA_IER[8]) = 1，会产生一个中断 |
| [0] | **LINS_EN** | **LIN从模式使能位**<br>0 = LIN 从模式禁止<br>1 = LIN 从模式使能 |

## UART LIN 状态寄存器 (UA_LIN_SR)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **UA_LIN_SR** | UARTx_BA+0x38 | R/W | UART LIN 状态寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | **BIT_ERR_F** | **LIN_BKDET_F** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | **LINS_SYNC_F** | **LINS_IDPERR_F** | **LINS_HERR_F** | **LINS_HDET_F** |

| 位 | 描述 | |
|---|---|---|
| [31:10] | **Reserved** | 保留. |
| [9] | **BIT_ERR_F** | **位错误侦测状态标志**<br>TX传输状态，硬件会监视总线状态，如果输入管脚 (SIN) 状态不等于输出管脚 (SOUT) 状态, BIT_ERR_F (UA_LIN_SR[9])位会被置位。<br>当发生位错误，如果LIN_IEN (UA_IER[8]) = 1，会产生一个中断<br>**注1:**向该位写1清0<br>**注2:** 该位仅当位错误侦测功能使能时有效(BIT_ERR_EN (UA_LIN_CTL[12]) = 1) |
| [8] | **LIN_BKDET_F** | **LIN Break 检测标志**<br>当侦测到一个break，该位由硬件置位。可通过软件写1将该位清零。<br>0 = LIN break 没有被检测到<br>1 = LIN break 被检测到.<br>**注1:**向该位写1清0.<br>**注2:**该位仅当 LIN break侦测功能使能时有效(LIN_BKDET_EN (UA_LIN_CTL[10]) = 1). |
| [7:4] | **Reserved** | 保留. |
| [3] | **LINS_SYNC_F** | **LIN 从机同步域.**<br>该位表示在自动重同步模式， LIN同步域正在被分析。 当接收器报头侦测到一些错误，用户必须通过写1到该位复位内部电路来重新搜索新的帧报头<br>0 =当前字符不在 LIN同步状态.<br>1 =当前字符在 LIN 同步状态.<br>**注1:** 该位仅在 LIN 从机模式有效(LINS_EN (UA_LIN_CTL[0]) = 1).<br>**注2:** 向该位写1清0<br>**注3:** 当向该位写1，硬件会重载初始波特率并重新搜索新的帧报头 |
| [2] | **LINS_IDPERR_F** | **LIN 从机 ID 校验错误标志** |

| | | |
|---|---|---|
| | | 当接收帧ID校验错误，该位由硬件置1<br>0 = 无效.<br>1 = 收到的帧ID校验错误.<br>**注1:**向该位写1清0..<br>**注2:**该位仅在 LIN 从机模式(LINS_EN (UA_LIN_CTL [0]) = 1)，并使能LIN帧ID校验功能(LIN_IDPEN (UA_LIN_CTL [9]) = 1).时 有效 |
| [1] | **LINS_HERR_F** | **LIN从机报头错误标志**<br>在LIN从机模式，当侦测到一个LIN报头错误时，该位由硬件置1，向该位写1清0。报头错误包括“break 间隔符太短（小于0.5位的时间），在同步域或在识别域中帧错误，在非自动重同步模式同步域数据不是0x55”，“自动重同步模式同步域偏离错误”，“自动重同步模式同步域测量超时”和“LIN 报头接收超时’<br>0 = LIN未检测到报头错误<br>1 = LIN检测到报头错误.<br>**注1:**向该位写1清0<br>**注2:** 该位仅在 LIN 从机模式(LINS_EN (UA_LIN_CTL [0]) = 1)，并且使能LIN 从机报头侦测功能(LINS_HDET_EN (UA_LIN_CTL [1]) = 1)时有效。 |
| [0] | **LINS_HDET_F** | **LIN 从机报头侦测标志**<br>在LIN从机模式，当侦测到一个LIN报头，该位由硬件置位，通过软件写1清0<br>0 = LIN 报头没有被侦测到.<br>1 = LIN 报头被侦测到(break + 同步+ 帧ID)<br>**注1:**向该位写1清0<br>**注2:** 该位仅在 LIN 从机模式(LINS_EN (UA_LIN_CTL [0]) = 1)，并且使能LIN 从机报头侦测功能(LINS_HDET_EN (UA_LIN_CTL [1])).时有效。<br>**注3:** 当使能ID 校验(LIN_IDPEN (UA_LIN_CTL [9]) = 1)，如果硬件侦测到完整的报头(“break + sync + frame ID”)，SLVHDETF会被置位，无论帧ID是否正确 |

## 6.13 I²C串行接口控制器(I²C)

### 6.13.1 概述

I²C 为双线，双向串行总线，通过简单有效的连线方式实现器件间的数据交换。I²C标准是多主机总线，包括冲突检测和仲裁，以防止在两个或多个主机同时尝试控制总线时发生数据损坏。I2C控制器支持掉电唤醒功能。

### 6.13.2 特性

I²C通过SDA 及SCL两条线与连接在总线上的器件传输信息，总线的主要特征有：

- 支持主机/从机 模式
- 主从机之间双向数据传输
- 总线支持多主机 (无中心主机)
- 多主机间同时传输数据仲裁，避免总线上串行数据损坏
- 总线采用串行同步时钟，可实现设备之间以不同的速率传输
- 串行时钟同步，可以用于握手机制来暂停或重新开始串行传输。
- 内建14位溢出定时器，当I2C总线中止且定时器溢出，产生I2C中断
- 需要外接上拉电阻用于输出高电平
- 可配置不同时钟以适用于可变速率控制
- 支持7位从地址模式
- I²C 总线控制器支持多地址识别（4组从机地址带mask 选项）

### 6.13.3 基本配置

I²C的基本配置如下:

- I²C0 管脚由寄存器P3_MFP[5:4] 或 P5_MFP[1:0] 或 PA_MFP[1:0] 配置.
- I²C0 中断向量由 NVIC_ISER 寄存器配置.
- 在 APBCLK [8] 寄存器中使能 I²C0 时钟 (I2C_EN).
- 在 IPRSTC2 [8] 寄存器中复位 I²C0 控制器 (I2C_RST).

### 6.13.4 框图

I²C 控制器如图 6-91所示

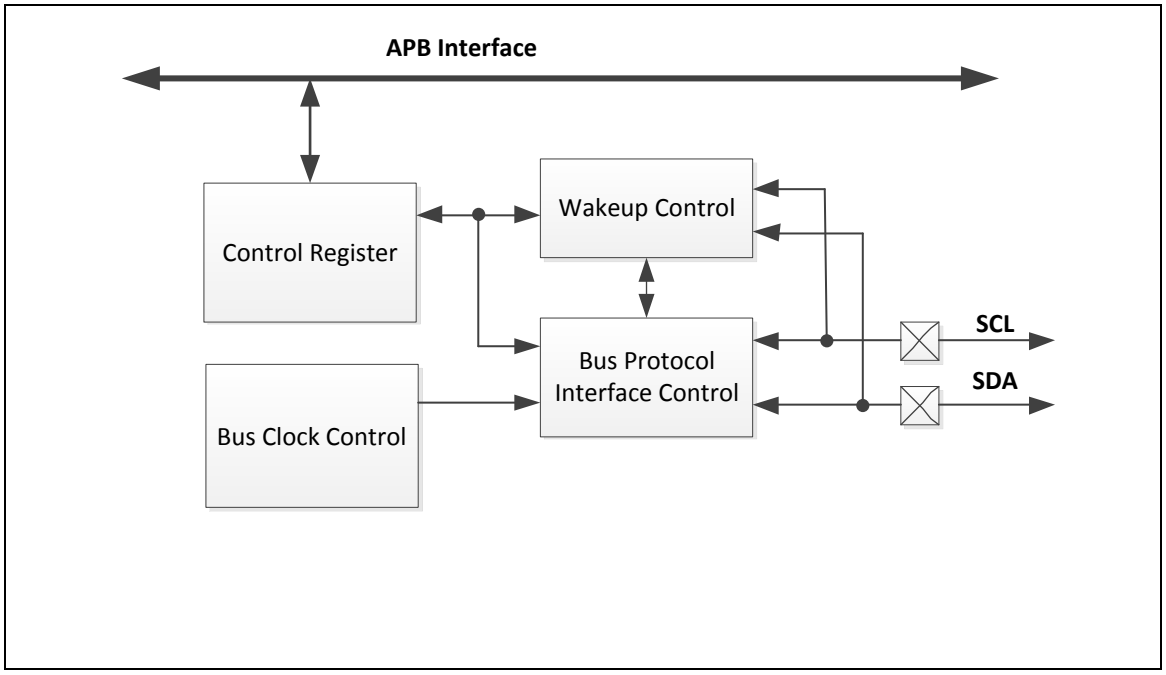



图 6-91 I²C 控制器框图

### 6.13.5 功能描述

在I2C总线上，数据通过时钟线SCL和数据线SDA在主从机间逐一字节同步传送。每个字节数据长度是8位. 一个SCL 时钟脉冲传输一个数据位，数据由最高位MSB 开始传输，每个字节传输后跟随一个应答位，每个位在SCL 为高时采样；因此，SDA 线可能在SCL 为低时改变，但在SCL 为高时必须保持稳定。当SCL 为高时，SDA 线上的跳变视为一个命令(START 或 STOP)。更多关于I2C总线时序的细节请参考 下图。

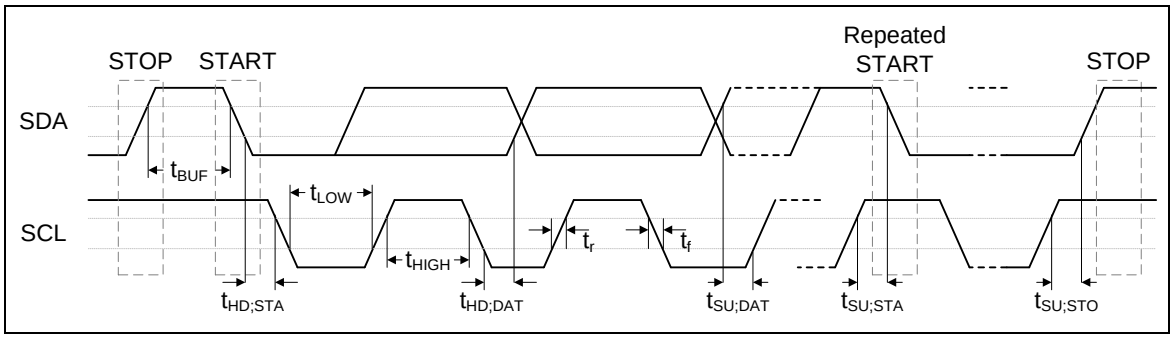



图 6-92 I²C 总线时序

片上I2C外设提供了一个符合I2C总线规范的串行接口。I2C端口自动处理字节传输。通过设置寄存器ENS1 (I2CON[6])为'1'可使能该端口。I2C 硬件接口通过数据线SDA 和时钟线 SCL两个管脚连到I2C总线。当I/O 管脚作为I2C 端口使用时，用户必须事先设定I/O 管脚为I2C功能。

**注意：**SDA 和 SCL两个管脚需要上拉电阻，因为这个两个管脚为开漏脚.

#### 6.13.5.1 I²C 协议

标准I²C 协议如下图，通常标准通讯有以下4部分:

1. 产生START 或 重复 START 信号
2. 从机地址传输和R/W 位传输
3. 数据传输
4. 产生STOP 信号

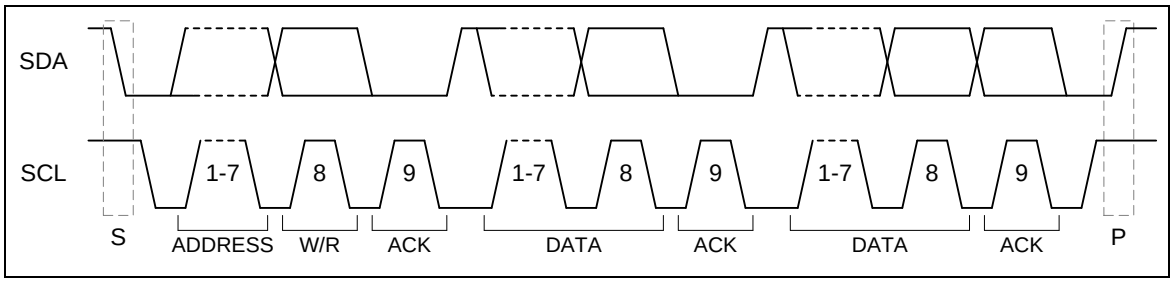



图 6-93 I²C 协议

##### 6.13.5.1.1 START 或重复START 信号

当总线处于释放/空闲状态下，说明没有主机设备占用总线（SDA 与SCL线同时为高），主机可以通过发送起始(START)信号来发起传输过程。起始信号，通常表示为"S"位，当SCL 线为高时，SDA 线上信号由高至低变化，就被定义为起始信号。起始信号表示一个新的数据传输的开始。

主机发送完地址字节（地址和读/写位）后，可以发送任何数量的字节数据并带一个停止信号。也可以用另一个起始信号替代停止信号，随后是地址(包含读写位)和更多的数据。这个起始信号叫做重复起始。这个定义可以用来发送任意个起始信号，它的目的是在不释放总线情况下，对一个或多个设备能读/写操作，而不让操作被打断。用这个方法主机跟其他从机或同一个从机在不同方向传输（例如，写设备到读设备）不用释放总线。

##### 6.13.5.1.2 STOP 信号

主机可以通过产生一个停止信号来终止数据通信。停止信号，通常表示为P位，当SCL 线为高时，SDA线上信号由低至高变化，就被定义为停止信号。

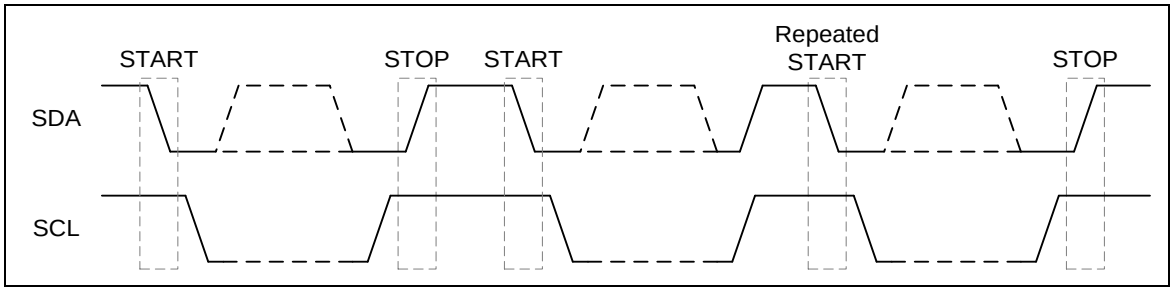



图 6-94 START 和 STOP 条件

#### *6.13.5.1.3 从地址传输*

起始信号后立即传输的第一个字节就是从机地址(SLA)。这是一个7位呼叫地址，后面跟随一个读/写(R/W) 位。R/W 位为从机数据传输方向的信号。系统中从机不能有相同的地址，只有被主机寻址的从机设备才会通过在第9个SCL 时钟周期时将SDA 拉低作为返回应答信号。

#### *6.13.5.1.4 数据传输*

当从机设备地址和R/W位被成功识别到，就可以根据R/W 位所决定的方向按一字节一字节方式进行数据传输。每个字节传输完后，紧接着的第9个SCL 时钟周期会有一个应答信号位。如果从机上产生无应答信号(NACK)，主机可以产生停止信号来中止数据传输或者产生重复起始信号开始新一轮数据传输。

当主机作为接收设备时，发生无应答信号(NACK)，则从机将释放SDA 线，让主机产生停止信号或重复起始信号。

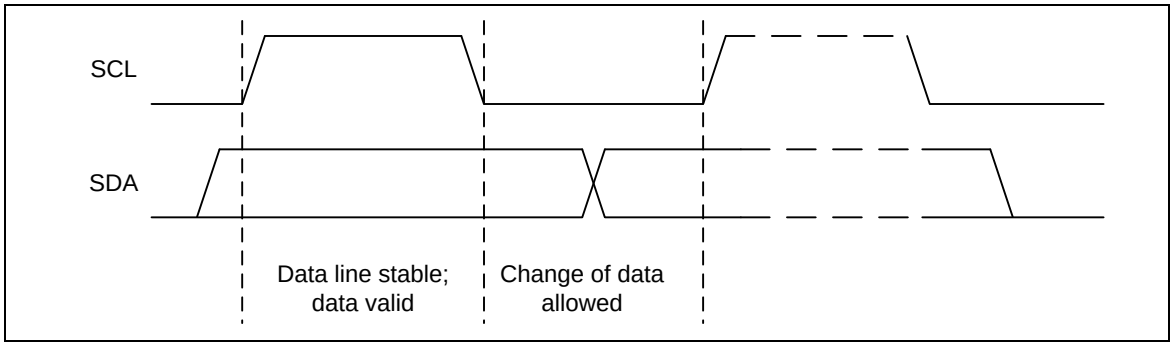



图 6-95 I²C 总线上的位传输

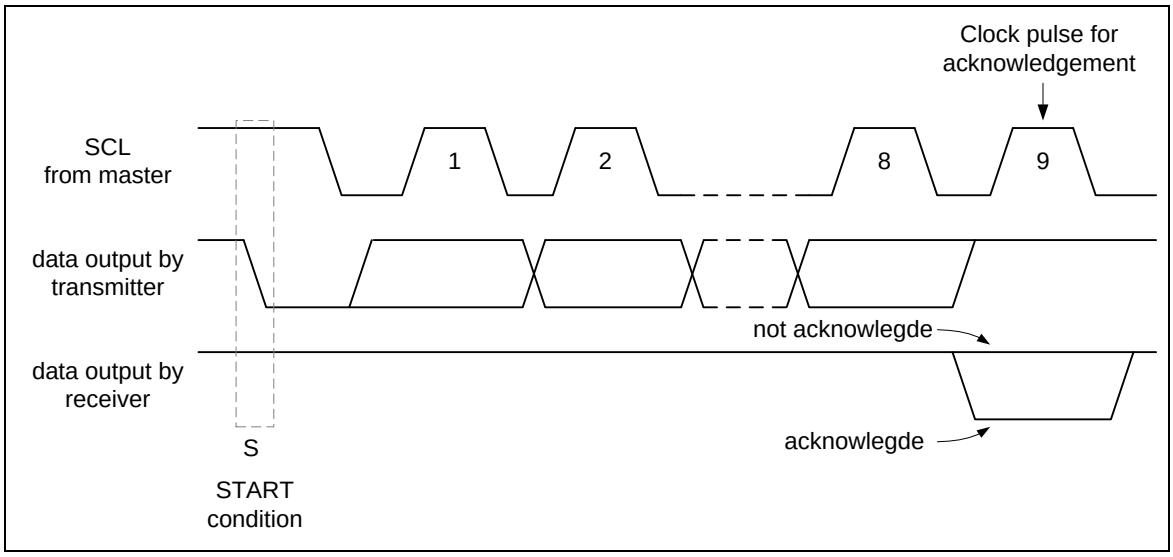



图 6-96 I²C 总线上的应答

*6.13.5.1.5 I²C-总线上的数据传输*

下图表示主机向从机传输数据。主机发出一个7-位地址和1-位写指示,表示主机想要传送数据给从机。从机回复应答给主机之后，主机继续传输数据。

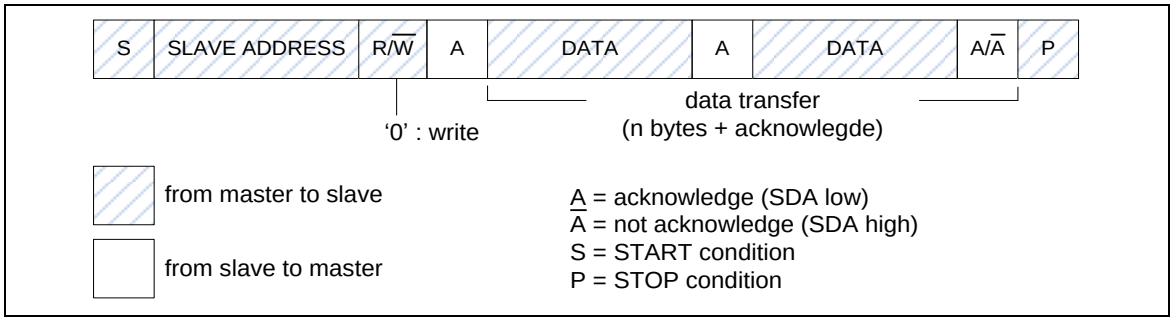



图 6-97主机向从机传输数据

下图表示主机向从机读取数据。主机发7-位地址寻址和1-位读指示，表示主机要向从机读取数据，从机返回应答给主机后，就开始给主机传输数据。

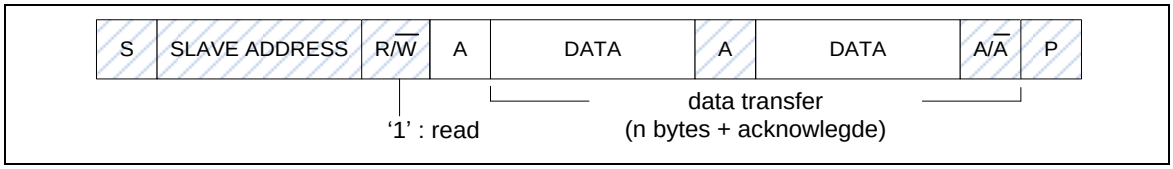



图 6-98 主机向从机读取数据

### 6.13.5.2 *工作模式*

片上I²C端口支持三种工作模式：主机模式，从机模式和广播呼叫模式。

应用中，I²C 端口可以作为主机和从机。在从机模式，I2C端口硬件会查找自身从机地址和广播呼叫地址，如果这两个地址的任一个被检测到，并且从机想要从主机接收或向主机发送数据(通过设置AA(I2CON[2])位)，应答脉冲将会在第9个时钟发出，此时，如果中断使能，则主机和从机设备上都会发生一次中断请求。在微控制器想要成为总线主机时，在进入主机模式之前，硬件需等待到总线空闲，以保证合理的从机动作不会被打断。在主机模式下，如果总线仲裁丢失，I2C端口立即切换到从机模式，并可以在同一次串行传输过程中检测自身的从机地址。

为控制I²C总线的各种模式传输，用户需要按照寄存器I2CSTATUS的当前状态码来设置I2CON, I2CDAT寄存器。换句话说，每个I2C总线动作都要检查I2CSTATUS寄存器的当前状态，然后再设置I2CON, I2CDAT寄存器执行总线动作。最后，通过I2CSTATUS检查响应状态。

在寄存器SI (I2CON[3])的SI标志清除后，这些位STA , STO (I2CON[5:4]) 和 AA (I2CON[2])用来控制I²C硬件的下一个状态。当完成一个新的动作，I2CSTATUS的状态代码将被更新，(I2CON[3])寄存器的SI标志将被设置。如果I²C中断控制位EI (I2CON [7])被设置，新状态代码对应的动作或者软件将在中断服务程序中被执行。

下图显示当前I²C状态码是0x08，然后通过设置DATA=SLA+W和(STA,STO,SI,AA) = (0,0,1,x)发送地址到I²C总线。如果总线上有从机匹配相应的地址且返回ACK，I2CSTATUS状态码更新为0x18。

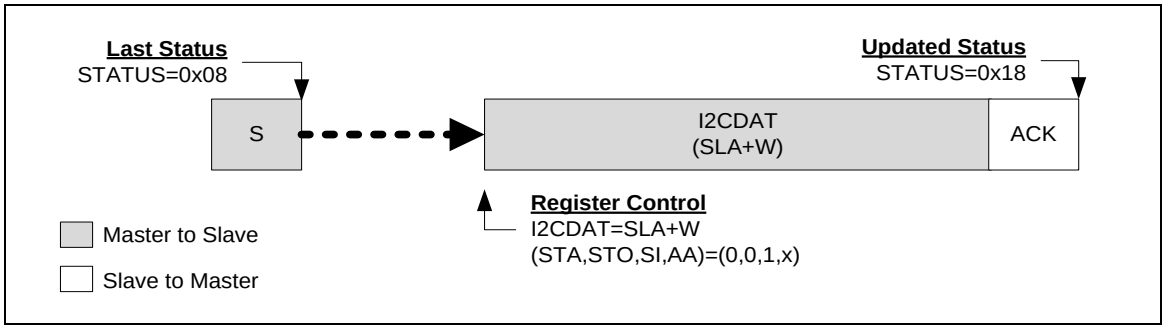



图 6-99 通过当前状态控制I²C总线

### 6.13.5.3 *主机模式*

下图中，是I²C主机模式所有可能的协议。用户需要遵循恰当的流程来实现I²C协议。

换句话说，用户可以发送一个起始信号到总线，I²C总线将变成主机传输模式，或主机接收模式。起始信号设置成功后新的状态码将是0x08。起始信号后，用户可以发送从机地址，读/写位，数据

和重复起始，停止来执行I²C协议。

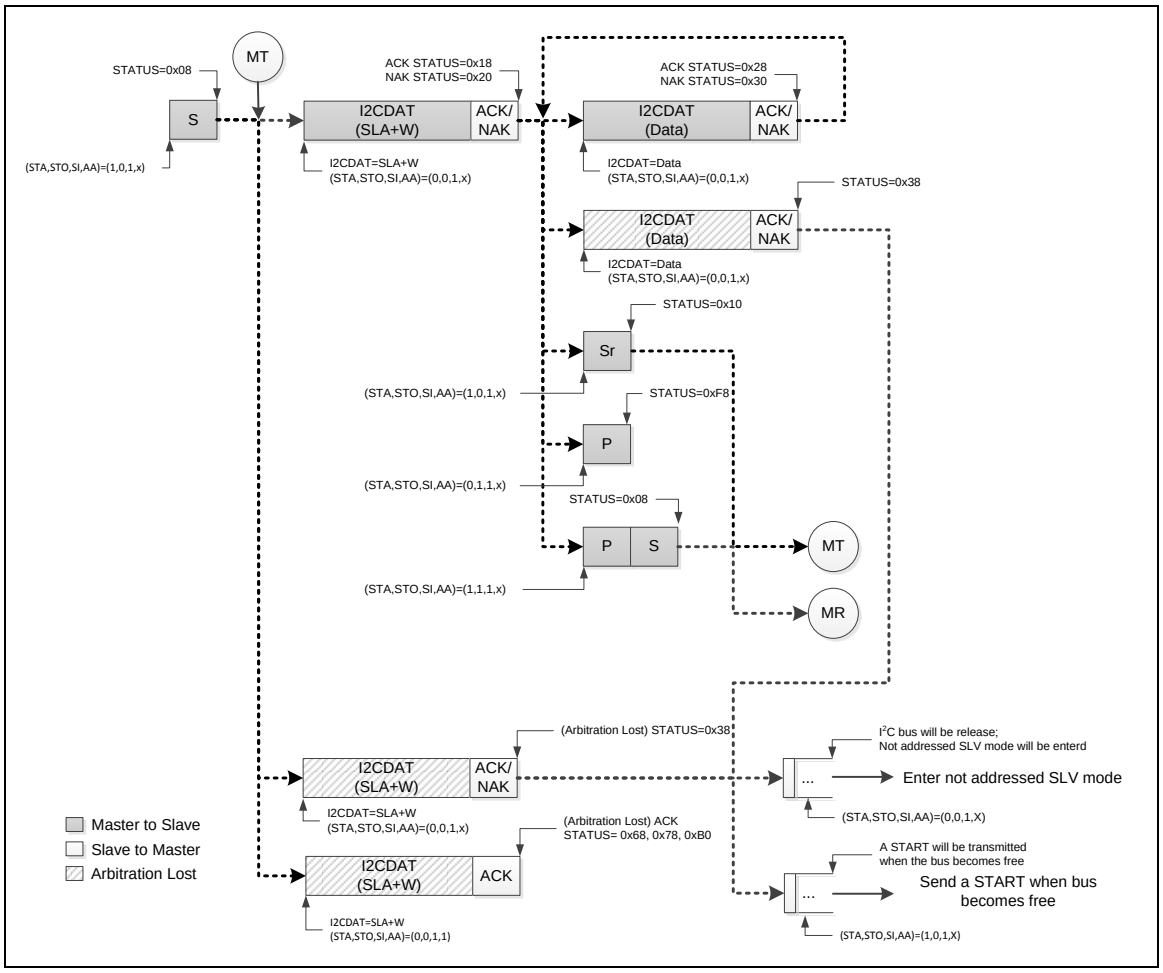



图 6-100 主机发送模式控制流程

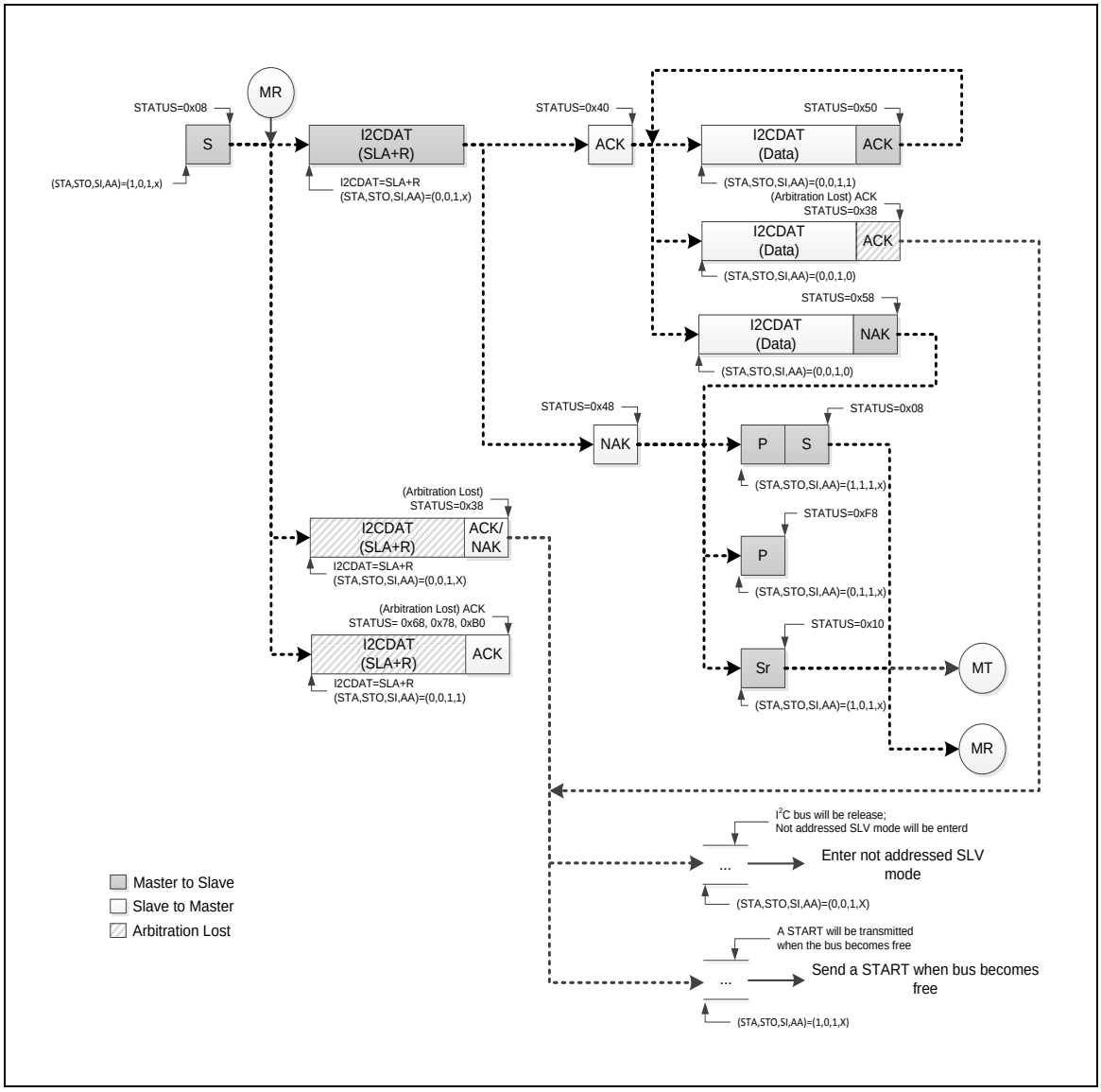



图 6-101 主机接收模式控制流程

如果I2C在主机模式并且仲裁丢失，状态码将置为0x38。在状态为0x38时，当总线空闲时，用户可以设置(STA, STO, SI, AA) = (1, 0, 1, X)发送起始信号来重新开始主机操作。否则，用户可以设置(STA, STO, SI, AA) = (0, 0, 1, X)来释放总线，并进入无地址从机模式。

#### *6.13.5.4 从机模式*

复位后默认情况下，$I^2C$不会被寻址，并且不会识别$I^2C$总线上的地址。用户可以通过I2CADDRx设置从机地址和设置 (STA, STO, SI, AA) = (0, 0, 1, 1)来让$I^2C$识别主机发送的地址。图 6-102所示为I2C从机模式的所有可能的流程。用户需要遵循适当的流程来实现他们的$I^2C$协议。

如果在主机模式总线仲裁丢失，$I^2C$端口立即切换到从机模式，并且在同一串行传输中识别自己的从机地址。如果在仲裁丢失后识别到地址是SLA+W（主机写数据到从机），状态码是0x68。如果在仲裁丢失后识别到地址是SLA+R（主机向从机读数据），状态码是0xB0。

**注意**：$I^2C$通信期间，在从机模式下，当对SI标志写‘1’清除时，SCL 时钟将被释放。

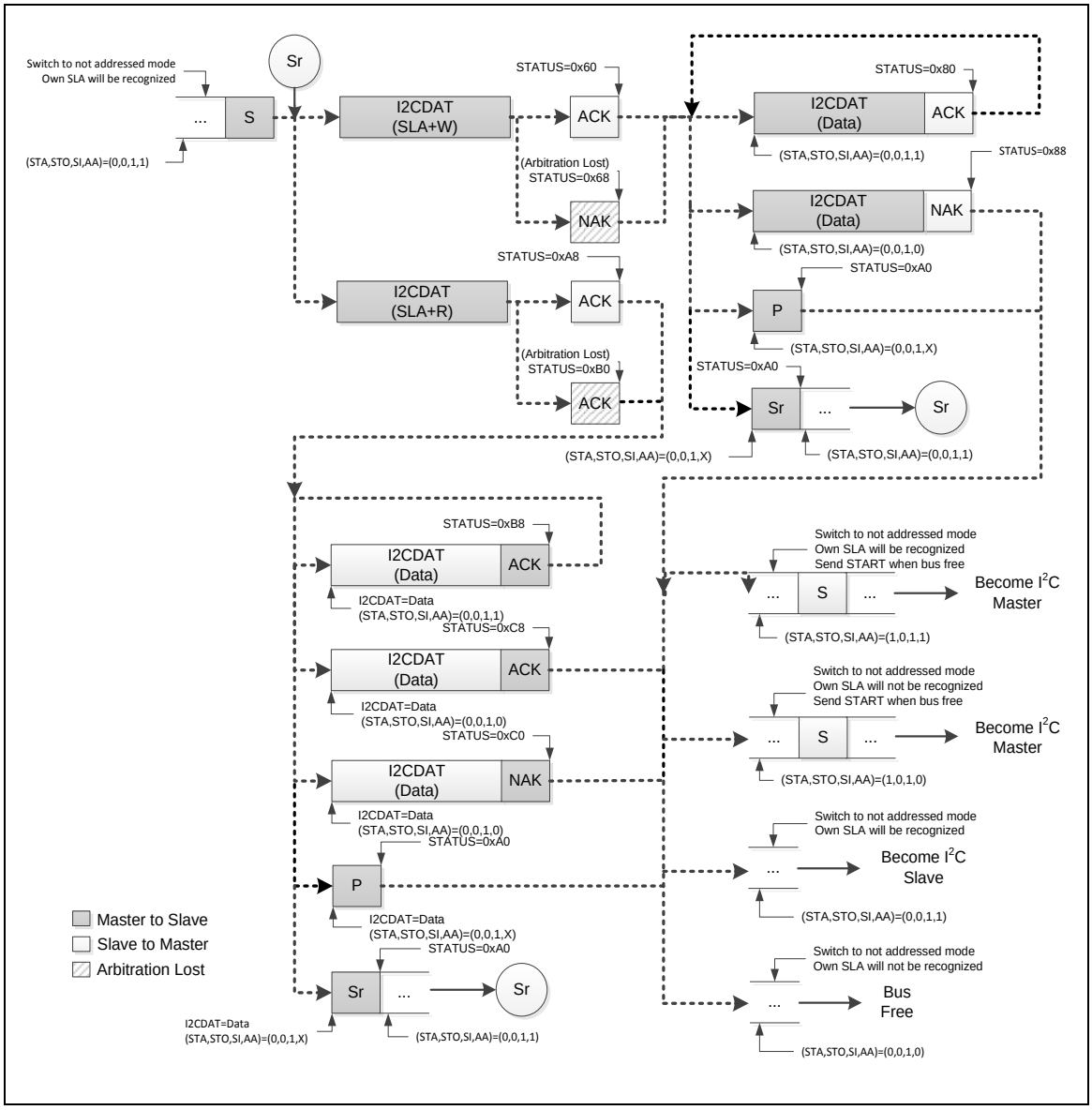



图 6-102 从机模式控制流程

如果I2C在可寻址从机模式接收数据时，收到停止或重复起始信号，状态码是0xA0。当状态码是0xA0时，用户可以遵循如上图状态码是0x88的操作。

如果I2C在可寻址从机模式发送数据时，收到停止或重复起始信号，状态码是0xA0。当状态码是0xA0时，用户可以遵循如上图状态码是0xC8的操作。

**注意：**从机获得0x88, 0xC8, 0xC0 和0xA0状态后,从机可以切换到无地址模式，自身SLA不会被辨识。如果进入这种状态，从机不再接收主机任何信号或地址。在这种状态，需要复位才能离开这种状态。

### *6.13.5.5 广播呼叫 (GC) 模式*

如果 GC位(I2CADDR[0]) 被设, $I^2C$端口硬件将响应广播呼叫地址(0x00).用户可以通过清GC位来禁止广播呼叫功能。当$I^2C$在从机模式时且GC 位被设置，可以接收主机地址(0x00)的广播呼叫，将遵

循广播模式状态。

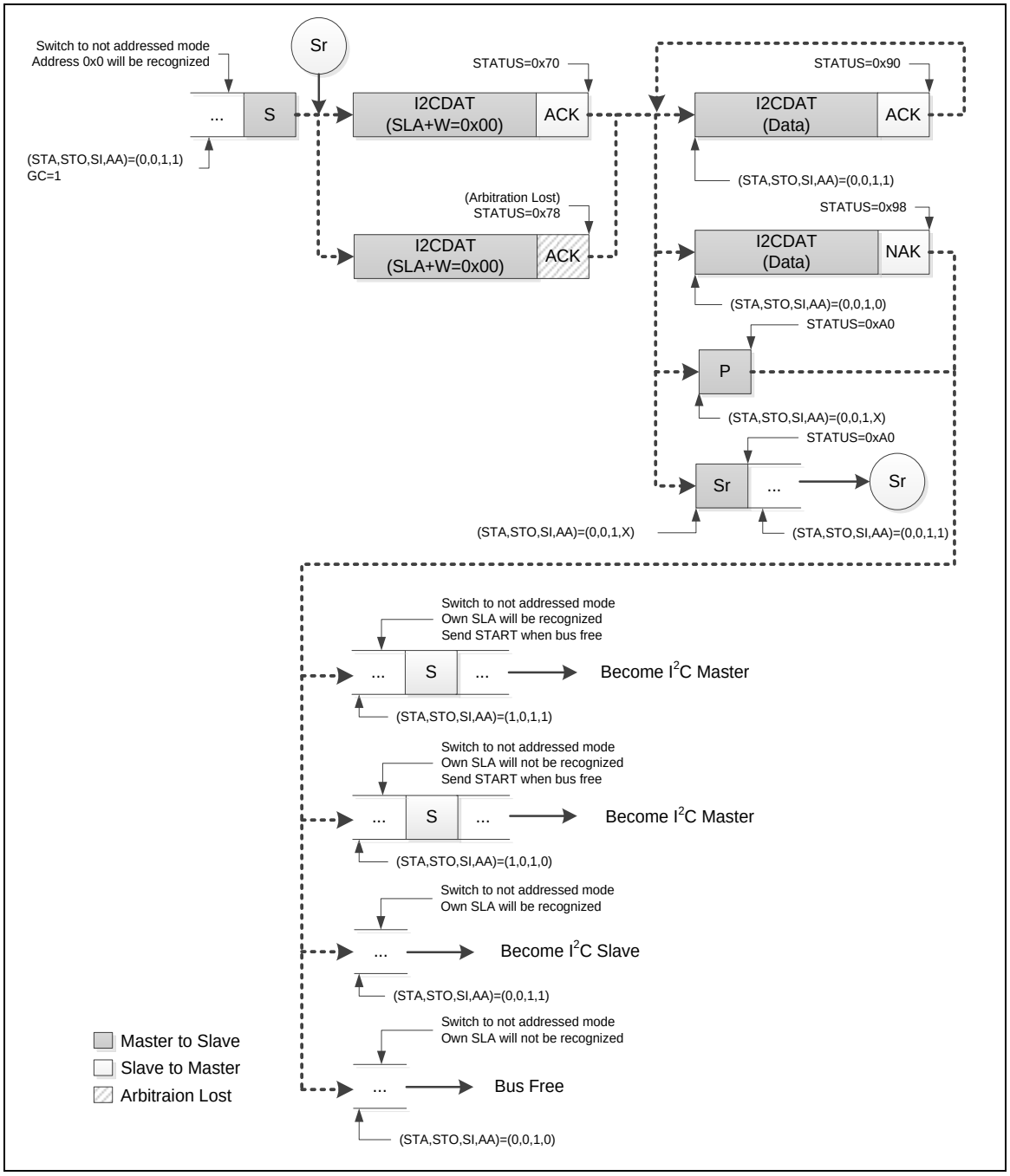



图 6-103 GC 模式

如果I2C在广播呼叫模式接收数据时，却收到停止或重复起始信号，状态码是0xA0。当状态码是0xA0时，用户可以遵循如上图状态码是0x98的流程处理。

**注意：**从机获得0x98 和0xA0 状态后，从机会切换到无地址模式并且自身SLA将不被辨识。如果进入这种状态，从机不再接收主机的任何信号或地址。需要复位才能离开这种状态。.

#### *6.13.5.6 多主机*

在一些应用中，一个$I^2C$总线上有多个主机同时访问从机，并有可能同时在传送数据。$I^2C$是支持多主机模式，并包含有冲突检测和仲裁，防止数据损坏。

如果两个主机同时发命令，通过仲裁来决定哪个优先并继续发命令。仲裁是在SCL为高时在SDA上执行的。每一个主机都会检测总线上的SDA信号是否符合它产生的SDA信号。如果检测到总线上SDA为低，但应该为高，则这个主机将失去仲裁。设备在仲裁丢失后会产生SCL脉冲直到本字节结束，然后释放总线进入从机模式。仲裁可以一直进行到所有数据被传输完。这样意味着多主机系统中主机必须监控总线和做相应的处理。

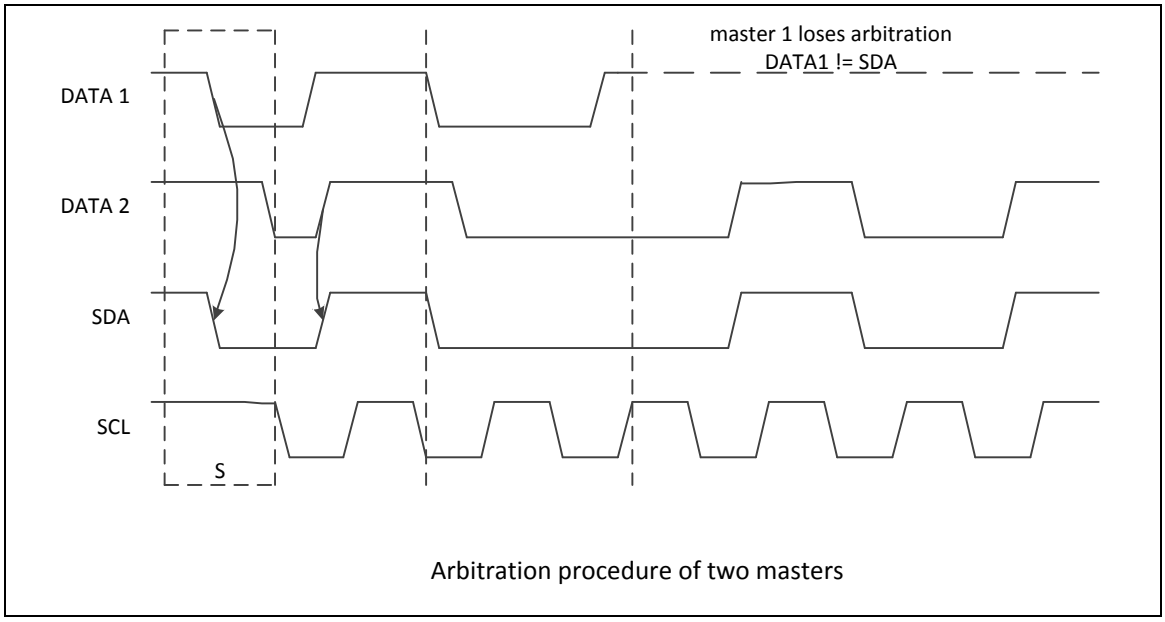



图 6-104 总裁丢失

- 当I2CSTATUS = 0x38代表接收到一个仲裁丢失。仲裁丢失事件可能发生在发送起始位，数据位或者停止位期间。用户可以在总线空闲时设置(STA, STO, SI, AA) = (1, 0, 1, X)来再次发送起始信号或者设置(STA, STO, SI, AA) = (0, 0, 1, X)发送停止信号来返回到无地址从机模式。
- 接收到一个“总线错误”，为了从总线错误中恢复I2C总线，STO应被设置且SI应将被清0，然后对STO清0来释放总线。
  - 设置 (STA, STO, SI, AA) = (0, 1, 1, X) 停止当前传输
  - 设置 (STA, STO, SI, AA) = (0, 0, 1, X) 释放总线

### 6.13.6 协议寄存器

通过下列15个特殊功能寄存器来控制I$^2$C端口：I2CCON（控制寄存器）I2CSTATUS（状态寄存器），I2CDAT（数据寄存器），I2CADDRn（地址寄存器，n=0~3），I2CADMn（地址掩码寄存器，n=0~3），I2CLK（时钟速率寄存器），I2CTOC（超时控制寄存器），I2CWKUPCON (唤醒控制寄存器)，I2CWKUPSTS (唤醒状态寄存器)。

#### *6.13.6.1 地址寄存器 (I2CADDR)*

I$^2$C端口内建4个从机地址寄存器I2CADDRn (n=0~3)。当I2C作为主机时，这四个寄存器的内容不相关。在从机模式下，位字段I2CADDRn[7:1]必须装载芯片自己的从机地址。当I2CADDRn地址与接收到的从机地址符合时I$^2$C硬件会作出反应。

I2C 端口支持“广播呼叫”功能。当GC位(I2CADDRn [0])被设置，I2C端口硬件将应答广播呼叫地址(00H)。清GC 位可禁用“广播呼叫”功能。

当GC 位被置且I2C处于从机模式时，主机发出广播呼叫地址到I2C总线后，从机可以通过地址00H接收广播呼叫地址，然后它将跟随GC 模式的状态。

I$^2$C总线支持多主机地址识别，带四个地址掩码寄存器I2CADMn (n=0~3)。当地址掩码寄存器的位被置1，意味着接收到对应的地址位匹配被忽略。如果该位置0，意味着接收到对应的地址位必须和地址寄存器的一样。

#### *6.13.6.2 从机地址掩码寄存器(I2CADM)*

I$^2$C总线控制器带有四个地址掩码寄存器I2CADMn (n=0~3) 支持多地址识别。当地址掩码寄存器被置1，意味着收到的相应地址位是"无关的".如果置0则收到相应地址位应跟地址寄存器完全一样。

#### *6.13.6.3 数据寄存器 (I2CDAT)*

该寄存器包含一个准备发送或刚接收到的一个字节的串行数据。当不在字节移位处理过程时，CPU可以直接读写访问I2CDAT[7:0]。当I2C处于一个确定的状态并且串行中断标志(SI) 被置1，I2CDAT[7:0]中的数据保持稳定。当数据被移出，总线上的数据同时被移入。I2CDAT[7:0]的内容总是总线上传递的最后一个字节。

应答位由I$^2$C硬件控制，不能被CPU 访问。串行数据在SCL线上串行时钟脉冲的上升沿移入I2CDAT[7:0] 。当一个字节被移入I2CDAT[7:0]，则I2CDAT[7:0] 中的串行数据是可用的，应答位(ACK 或NACK) 在第9个时钟脉冲由控制逻辑返回。为了在发送数据时监控总线的状态，当发送I2CDAT[7:0]到总线时，总线数据将同时被移入I2CDAT[7:0]。在发送数据过程中，串行数据在SCL时钟脉冲的下降沿从I2CDAT[7:0] 移出，在SCL时钟脉冲的上升沿数据移入I2CDAT [7:0]。

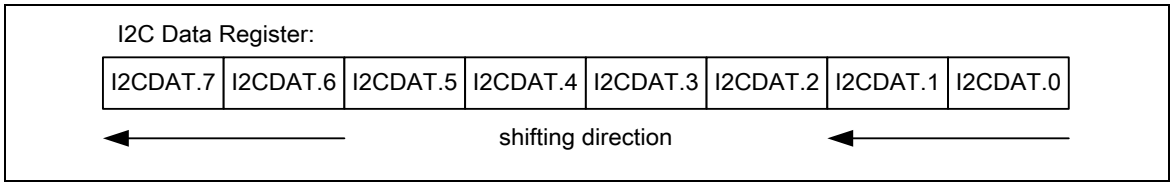



图 6-105 I$^2$C数据移位方向

#### 6.13.6.4 控制寄存器 (I2CON)

CPU 可以直接读或写寄存器I2CON [7:0]，有两个位会受硬件影响：当I2C硬件产生中断时SI被置位，当总线上产生停止信号时，STO位被清零，当ENS1 =0时STO也会被清零。

#### 6.13.6.5 状态寄存器(I2CSTATUS)

I2CSTATUS[7:0] 是一个8-位只读寄存器。I2CSTATUS[7:0]共有26个可能的状态码。所有状态码由下面表所列出。当I2CSTATUS[7:0] 的值为0xF8时，没有串行中断请求。所有其他的I2CSTATUS[7:0]的值对应$I^2C$ 的状态。当进入其中任一状态时，就会产生状态中断请求(SI=1)。在SI 被硬件置位或SI 被软件复位后一个机器周期，有效状态码出现在I2CSTATUS[7:0] 中。

此外，状态0x00表示总线错误，这会出现在起始或停止条件处在不正确的$I^2C$格式帧。总线错误会发生在一个地址字节、一个数据字节或一个应答位的串行传输。从错误总线恢复时STO应被置位，S应被清除，来进入无地址从机模式。然后STO清除释放总线并等待下一个传输。出现总线错误操作期间$I^2C$总线不能识别停止条件。

| 主机模式 | | 从机模式 | |
|---|---|---|---|
| 状态 | 描述 | 状态 | 描述 |
| 0x08 | 起始 | 0xA0 | 从机发送重新起始或停止 |
| 0x10 | 主机重新起始 | 0xA8 | 从机发送地址ACK |
| 0x18 | 主机发送地址ACK | 0xB0 | 从机发送仲裁丢失 |
| 0x20 | 主机发送地址NACK | 0xB8 | 从机发送数据ACK |
| 0x28 | 主机发送数据ACK | 0xC0 | 从机发送数据NACK |
| 0x30 | 主机发送数据NACK | 0xC8 | 从机发送最后数据ACK |
| 0x38 | 主机仲裁丢失 | 0x60 | 从机接收地址 ACK |
| 0x40 | 主机接收地址ACK | 0x68 | 从机接收仲裁丢失 |
| 0x48 | 主机接收地址NACK | 0x80 | 从机接收数据 ACK |
| 0x50 | 主机接收数据ACK | 0x88 | 从机接收数据NACK |
| 0x58 | 主机接收数据NACK | 0x70 | 广播模式地址ACK |
| 0x00 | 总线错误 | 0x78 | 广播模式仲裁丢失 |
| | | 0x90 | 广播模式数据 ACK |
| | | 0x98 | 广播模式数据 NACK |
| 0xF8 | 总线释放<br>**注意**：主/从模式的“0xF8”状态，都不会产生中断 | | |

表 6-21 $I^2C$状态码描述

#### 6.13.6.6 I²C时钟波特率位(I2CLK)

当I²C 在主机模式下，I²C数据的波特率由I2CLK [7:0]寄存器设定。在从机模式下时不需要。在从机模式下，I²C 将自动与主机I²C设备时钟频率同步。

I²C 数据波特率的设定：I²C数据波特率= (系统时钟) / (4x ((I2CLK [7:0] +1))。如果系统时钟 = 16 MHz, I2CLK [7:0] = 40 (0x28), 那么I²C数据波特率= 16 MHz/ (4x (40 +1)) = 97.5 Kbits/sec。

#### 6.13.6.7 I²C超时计数器寄存器(I2CTOC)

系统提供一个14 位超时的计数器来处理当I2C总线锁死时的情况。当超时计数器计数功能使能后，计数器开始计数直至溢出(TIF(I2CTOC[0])=1) 并向CPU 产生I2C中断或者清除ENTI(I2CTOC[2])为0停止计数。当超时计数器使能，对SI 标志置高会使计数器复位，再对SI 清零会重新开始计数。如果I2C总线锁死，会使I2CSTATUS及SI(I2CON[3])标志不再更新，该14-位超时计数器会发生溢出从而产生I2C中断通知CPU。参考下图14-位超时计数器。用户可以写1 清TIF(I2CTOC[0])为0。

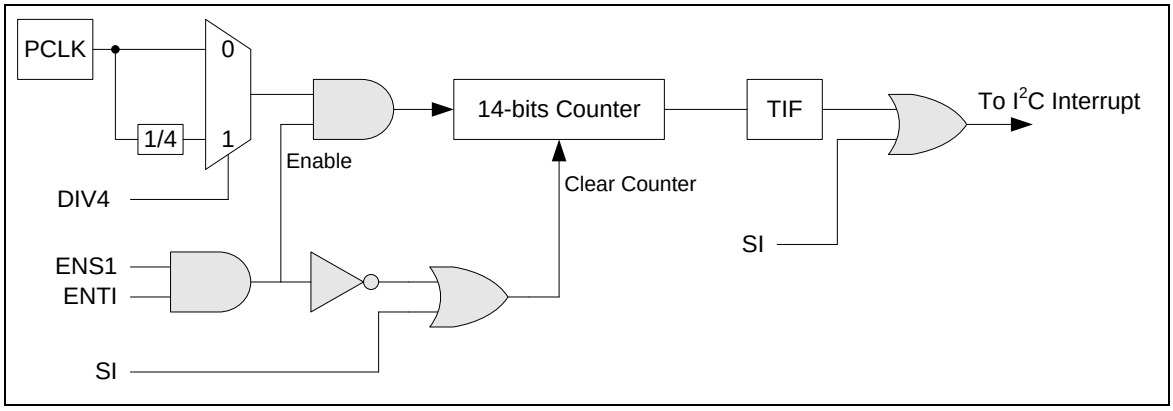



图 6-106 I²C 超时计数框图

#### 6.13.6.8 I²C 唤醒控制寄存器 (I2CWKUPCON)

当系统进入掉电模式后，其他的I²C主机可以通过寻址I²C设备唤醒我们的芯片。用户必须在进入掉电模式之前做相关设置。当地址与四个地址寄存器的其中一个相匹配时芯片唤醒，此时下一个数据将被作废。

#### 6.13.6.9 I²C 唤醒状态寄存器 (I2CWKUPSTS)

当系统被其他的I²C主机设备唤醒时，WKUPIF(I2CWKUPSTS[0])被置位表示该事件发生。用户需要写“1”来清除此位。

#### 6.13.6.10 随即读取 EEPROM的例子

通过下面的步骤来配置I²C0相关寄存器，来使用I²C从EEPROM读取数据。

8. 在 P3_MFP[5:4]寄存器中将多功能管脚设置成 SCL 和 SDA 管脚。
9. 通过设置寄存器 APBCLK0 的 I2C0_EN 为 1 使能 I2C0 APB 时钟。

10. 通过设置I2C0_RST=1来复位I$^2$C0控制器，然后通过I2C0_RST= 0将I$^2$C控制器设置成普通操作。I2C0_RST位在IPRSTC2寄存器中。
11. 通过设置寄存器I2CON中的EI为1使能I$^2$C0控制器。
12. 通过在I2CLK寄存器写I$^2$C0时钟分频值。
13. 在“NVIC_ISER”寄存器中设置SETENA=0x00000040来设置I$^2$C0 IRQ.
14. 设置寄存器I2CON的EI为1使能I$^2$C0中断
15. 设置I$^2$C0地址寄存器“I2CADDR0 ~ I2CADDR3”。

随机读操作是访问EEPROM的其中一种方法。这个方法是允许主机访问EEPROM的任何一个地址。下图显示对EEPROM随机读取操作。

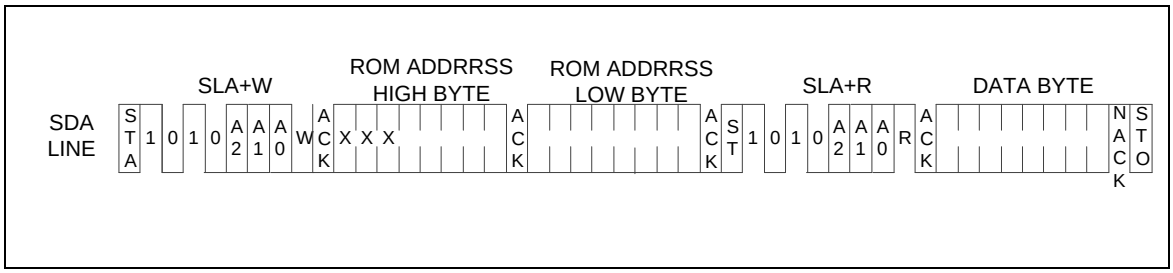



图 6-107 EEPROM 随即读

下图显示怎样使用I2C控制器执行EEPROM随机读取操作协议

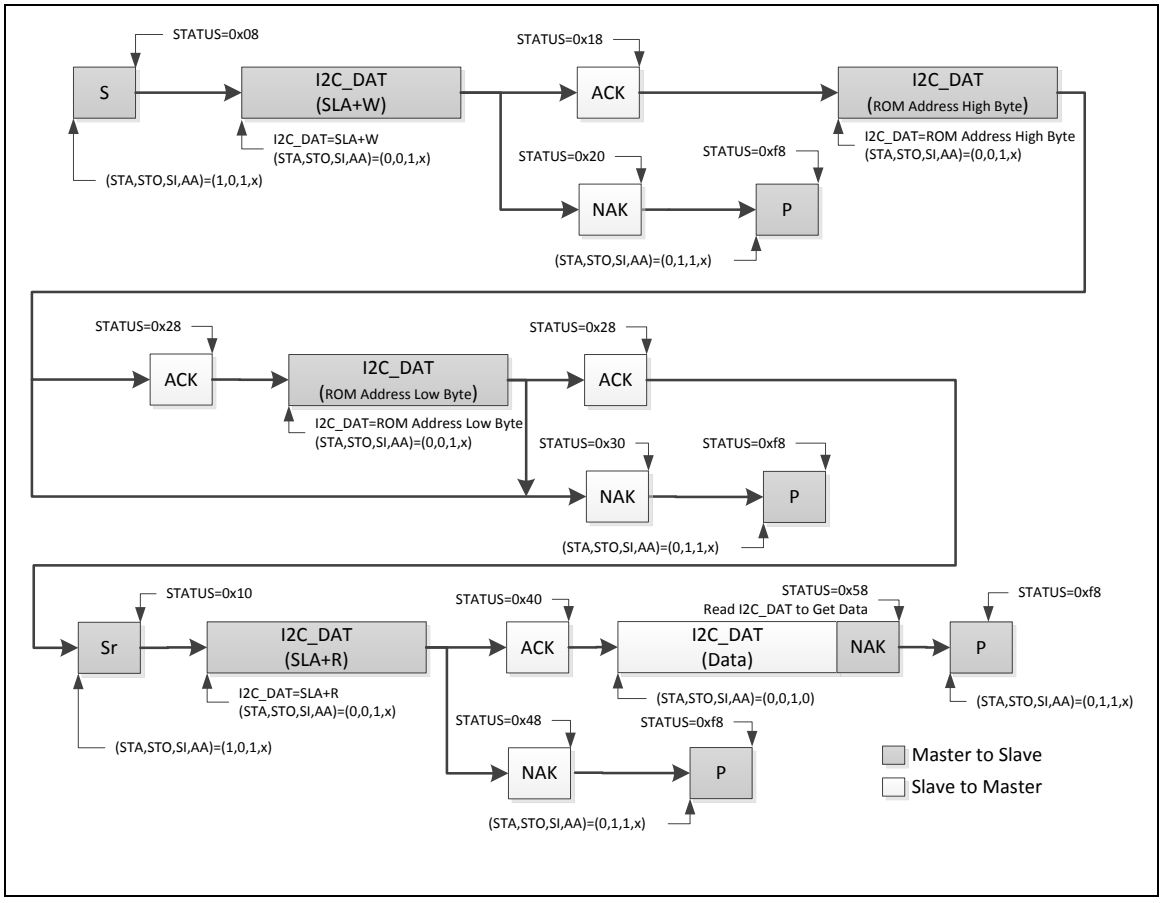


图 6-108 EEPROM 随机读协议

I$^2$C控制器作为主机发送起始信号到总线，然后发送SLA+W (从机地址 + 写 位)到EERPOM，跟着由两个字节数据地址来设置EEPROM被读的地址。最后，重复起始信号跟着SLA+R被发送来向EEPROM 读取数据。

### 6.13.7 寄存器映射

**R**: 只读, **W**: 只写, **R/W**: 可读写

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **I2C 基地址:**<br>**I2C0_BA = 0x4002_0000** | | | | |
| **I2CON** | I2Cx_BA+0x00 | R/W | $I^2C$ 控制寄存器 | 0x0000_0000 |
| **I2CADDR0** | I2Cx_BA+0x04 | R/W | $I^2C$ 从机地址寄存器0 | 0x0000_0000 |
| **I2CDAT** | I2Cx_BA+0x08 | R/W | $I^2C$ 数据寄存器 | 0x0000_0000 |
| **I2CSTATUS** | I2Cx_BA+0x0C | R | $I^2C$ 状态寄存器 | 0x0000_00F8 |
| **I2CLK** | I2Cx_BA+0x10 | R/W | $I^2C$ 时钟分频寄存器 | 0x0000_0000 |
| **I2CTOC** | I2Cx_BA+0x14 | R/W | $I^2C$ 超时计数器寄存器 | 0x0000_0000 |
| **I2CADDR1** | I2Cx_BA+0x18 | R/W | I2C 从机地址寄存器1 | 0x0000_0000 |
| **I2CADDR2** | I2Cx_BA+0x1C | R/W | I2C 从机地址寄存器2 | 0x0000_0000 |
| **I2CADDR3** | I2Cx_BA+0x20 | R/W | I2C 从机地址寄存器3 | 0x0000_0000 |
| **I2CADM0** | I2Cx_BA+0x24 | R/W | I2C 从机地址掩码寄存器0 | 0x0000_0000 |
| **I2CADM1** | I2Cx_BA+0x28 | R/W | I2C 从机地址掩码寄存器1 | 0x0000_0000 |
| **I2CADM2** | I2Cx_BA+0x2C | R/W | I2C 从机地址掩码寄存器2 | 0x0000_0000 |
| **I2CADM3** | I2Cx_BA+0x30 | R/W | I2C 从机地址掩码寄存器3 | 0x0000_0000 |
| **I2CWKUPCON** | I2Cx_BA+0x3C | R/W | $I^2C$ 唤醒控制寄存器 | 0x0000_0000 |
| **I2CWKUPSTS** | I2Cx_BA+0x40 | R/W | $I^2C$ 唤醒状态寄存器 | 0x0000_0000 |

### 6.13.8 寄存器描述

**I²C 控制寄存器 (I2CON)**

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **I2CON** | I2Cx_BA+0x00 | R/W | I²C 控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **EI** | **ENS1** | **STA** | **STO** | **SI** | **AA** | **Reserved** | |

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留. |
| [7] | **EI** | **使能中断**<br>0 = 禁用 I²C 中断<br>1 = 使能 I²C 中断 |
| [6] | **ENS1** | **I²C 控制器使能位**<br>设置使能 I²C 串行功能控制器。当ENS1 =1，I²C 串行功能使能。SDA和SCL对应的多功能管脚功能必须先设置为 I²C 功能。<br>0 = I2C控制器禁用<br>1 = I2C控制器使能 |
| [5] | **STA** | **I²C 起始控制位**<br>设置 STA 为 1，进入主机模式，如果总线处于空闲状态，I²C 硬件会送出 START 或 重复 START 条件。 |
| [4] | **STO** | **I²C 停止控制位**<br>在主机模式，设置 STO 来传送一个 STOP 条件到总线，然后 I²C 硬件将会检查总线状况，如果检测到一个 STOP 状况，这个标志会被硬件自动清除。在 I2C 从机模式，设置 STO 复位 I2C 硬件来定义“无地址”从机模式，这表示在从机接收模式不再接收从主机设备发送的数据。 |
| [3] | **SI** | I2C 中断标志<br>当一个新的 I₂C 状态出现在寄存器 I2CSTATUS 时，SI 标志由硬件置位，并且如果 EI (I2CON [7]) 位被置位，则产生 I₂C 中断请求。SI 必须由软件通过向该位写 '1' 清零。 |
| [2] | **AA** | **应答控制位**<br>当 AA=1 先于地址或数据接收，在以下两种情况：1.) 从机正在应答主机发送的地址。2.) 接收设备正在应答发送设备发送的数据，在SCL线上的应答时钟脉冲期间将返回一个应答（SDA上的低电平）。当 AA = 0 先于地址或数据接收，则在SCL线上的应答时钟脉冲期间将返回一个非应答（SDA上的高电平）。 |

| | | |
|---|---|---|
| [1:0] | **Reserved** | 保留 |

## I²C 数据寄存器 (I2CDAT)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **I2CDAT** | I2Cx_BA+0x08 | R/W | I²C 数据寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **I2CDAT** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留. |
| [7:0] | **I2CDAT** | **I²C 数据寄存器**<br>该区域 8 位存放 I²C 串行端口8位传输/接收数据 |

## I²C 状态寄存器 (I2CSTATUS )

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **I2CSTATUS** | I2Cx_BA+0x0C | R | I²C 状态寄存器 | 0x0000_00F8 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **I2CSTATUS** | | | | | | | |

| 位 | 模式 | |
|---|---|---|
| [31:8] | **Reserved** | 保留. |
| [7:0] | **I2CSTATUS** | **I²C状态寄存器**<br>低3位有效位一直为0。高5位有效位包括状态码。有26种可能的状态码。<br>当I2C_STATUS的内容为0xF8时，没有中断产生。所有其他的I2CSTATUS的值对应 I²C 的状态。当进入其中任一状态时，就会产生状态中断请求 (SI=1)。在 SI 被硬件置位或 SI 被软件复位后一个机器周期，有效状态码出现在I2CSTATUS中。<br>此外，00H 状态表示总线错误。总线错误发生在 START 或 STOP 条件出现在帧结构不正确的位置。不正确的位置比如是在串行传输地址字节、数据字节或应答位期间。 |

## I²C 时钟分频寄存器 (I2CLK)

| 寄存器 | 偏移量 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **I2CLK** | I2Cx_BA+0x10 | R/W | I²C 时钟分频寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **I2CLK** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留. |
| [7:0] | **I2CLK** | **I²C 时钟分频寄存器**<br>I²C 数据波特率 = (系统时钟) / (4x (I2CLK +1)).<br>**注意:** I2CLK的最小值是 4. |

## I²C超时计数器寄存器 (I2CTOC)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **I2CTOC** | I2Cx_BA+0x14 | R/W | I²C超时计数器寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | **ENTI** | **DIV4** | **TIF** |

| 位 | 描述 | |
|---|---|---|
| [31:3] | **Reserved** | 保留. |
| [2] | **ENTI** | **超时计数器使能位**<br>当使能该位，则14-位溢出定时器将会在 SI 清零后开始计数。设置 SI位为高将会复位计数器，在 SI位清零之后，计数器会重新开始计数。<br>0 = 超时计数器禁用<br>1 = 超时计数器使能 |
| [1] | **DIV4** | **溢出定时器输入时钟除以 4**<br>该位使能后，超时时间扩大 4 倍。<br>0 = 溢出定时器输入时钟禁用<br>1 = 溢出定时器输入时钟使能 |
| [0] | **TIF** | **定时溢出标志**<br>当超时发生时，该位由 H/W 置位，如果此时 I²C 的中断使能位（EI）置为1，则可引发 CPU 的中断。<br>**注意**：写 1 清除该位。 |

## I²C 从机地址寄存器 (I2CADDRx)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **I2CADDR0** | I2Cx_BA+0x04 | R/W | I²C 从机地址寄存器0 | 0x0000_0000 |
| **I2CADDR1** | I2Cx_BA+0x18 | R/W | I²C从机地址寄存器1 | 0x0000_0000 |
| **I2CADDR2** | I2Cx_BA+0x1C | R/W | I²C从机地址寄存器2 | 0x0000_0000 |
| **I2CADDR3** | I2Cx_BA+0x20 | R/W | I²C从机地址寄存器3 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **I2CADDR** | | | | | | | **GC** |

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留. |
| [7:1] | **I2CADDR** | **I²C 地址位**<br>主机模式下，该寄存器内容没有意义。从机模式下，高七位作为芯片本身的地址。如果地址符合，I²C 硬件将会自动应答。 |
| [0] | **GC** | **广播呼叫功能**<br>0 = 广播呼叫功能禁用<br>1 = 广播呼叫功能使能 |

## I²C 从机地址掩码寄存器 (I2CADMx)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **I2CADM0** | I2Cx_BA+0x24 | R/W | I²C 从机地址掩码寄存器0 | 0x0000_0000 |
| **I2CADM1** | I2Cx_BA+0x28 | R/W | I²C从机地址掩码寄存器1 | 0x0000_0000 |
| **I2CADM2** | I2Cx_BA+0x2C | R/W | I²C从机地址掩码寄存器2 | 0x0000_0000 |
| **I2CADM3** | I2Cx_BA+0x30 | R/W | I²C从机地址掩码寄存器3 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **I2CADM** | | | | | | | **Reserved** |

| 位 | 描述 | |
|---|---|---|
| [31:8] | **Reserved** | 保留. |
| [7:1] | **I2CADM** | **I²C 地址掩码位**<br>I²C 总线控制器有4个地址掩码寄存器，支持多地址识别。当地址掩码寄存器的某位被置 '1'，表示接收到的地址的相应位可忽略。如果该位被置 '0'，则表示接收到的地址的相应位必须和地址寄存器中的完全一致。<br>0 = 掩码禁用（接收到的相应地址必须完全符合地址寄存器）<br>1 = 掩码使能（接收到的相应地址位不予辨识） |
| [0] | **Reserved** | 保留 |

## I²C 唤醒控制寄存器 (I2CWKUPCON)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **I2CWKUPCON** | I2Cx_BA+0x3C | R/W | I²C 唤醒控制寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留. |
| [0] | **WKUPEN** | **I²C 唤醒功能使能位**<br>0 = I²C 唤醒功能禁止.<br>1 = I²C 唤醒功能使能 |

## I²C 唤醒状态寄存器 (I2CWKUPSTS)

| 寄存器 | 偏移地址 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **I2CWKUPSTS** | I2Cx_BA+0x40 | R/W | I²C 唤醒状态寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:1] | **Reserved** | 保留. |
| [0] | **WKUPIF** | **I²C 唤醒中断标志**<br>当芯片从掉电模式下由I²C唤醒时，该位置为1。<br>**注意：**该位软件写1清0。 |

## 6.14 串行外围设备接口 (SPI)

### 6.14.1 概述

串行外围设备接口 (SPI) 是一种全双工同步串行通讯协议。设备可以工作在主模式或者从模式，通过双向的4线信号传送数据。NuMicro® M0519系列最多提供3个SPI 控制器，当从外部设备接收数据的时候执行串并转换，当向外部设备发送数据的时候执行并串转换。每个SPI控制器都可以工作在主模式或者从模式。

### 6.14.2 特性

- 多达三组SPI控制器
- 支持主模式或从模式
- 一次传输的位数可以配置成 8 到 32-bit
- 提供发送和接收独立的8级深度的 FIFO 缓存
- 支持高位先发或低位先发
- 支持字节重排序功能
- 支持没有片选信号的双向3线模式

### 6.14.3 模块图

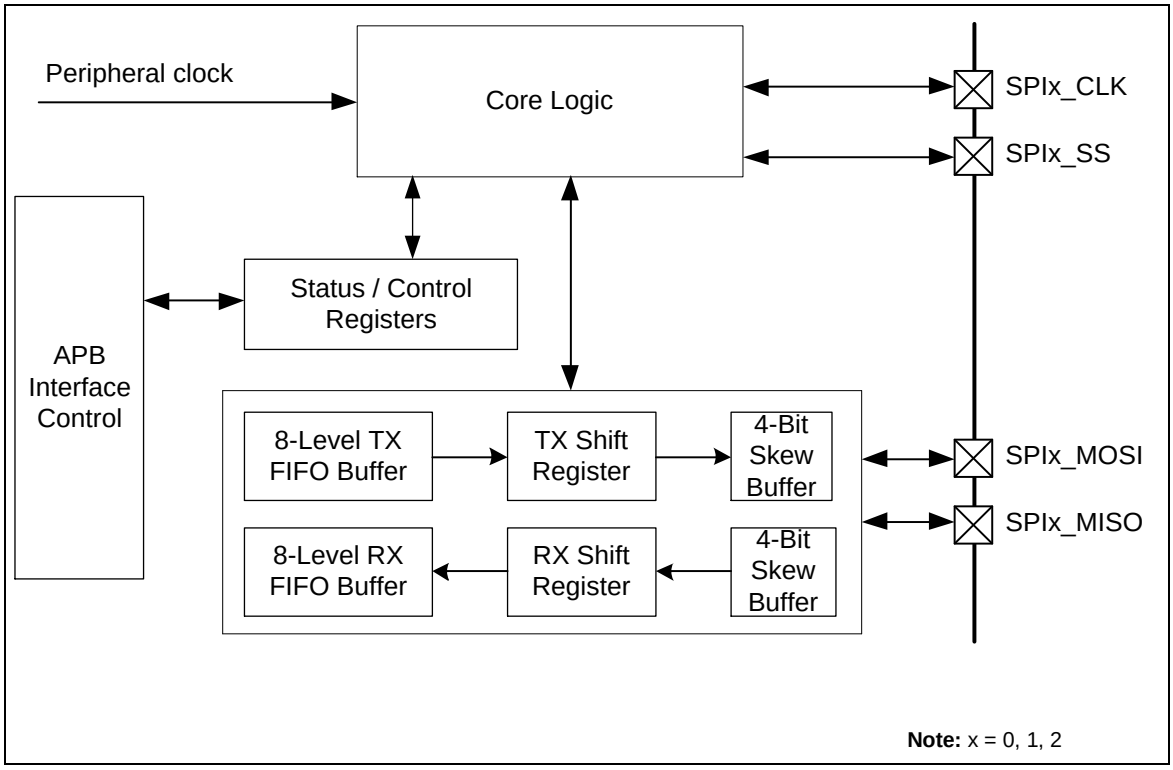



图 6-109 SPI 模块图

**发送 FIFO 缓存:**

发送FIFO缓存是8级深度的,32位宽度，先进先出的缓存。 程序可以通过写SPI_TX 寄存器来写数据到发送缓存。

**接收 FIFO缓存:**

接收 FIFO 缓存也是8级深度，32位宽度，先进先出的缓存。接收逻辑单元会把接收到的数据存到这个缓存里。程序可以通过读SPI_RX 寄存器来读出FIFO里的数据。

**发送移位寄存器:**

发送移位寄存器是一个32位的寄存器缓存。发送的数据从发送FIFO里一位一位的传送到偏离缓存里。

**接收移位寄存器:**

接收移位寄存器也是一个32位的寄存器缓存。当一次接收完成，接收到的数据会一位一位的从偏离缓存移到接收FIFO里。

**偏离缓存:**

偏离缓存是一个4位的缓存。

在发送的时候，从移位寄存器移位到偏离缓存按照外设时钟同步，从偏离寄存器发送出去按照SPI总线的时钟同步。当开始发送的时候，从发送移位寄存器传送来的前3位数据会首先载入偏离缓存。当这些偏离缓存里的数据有数据传送到SPI总线上，发送FIFO会再移位数据传输到偏离缓存的剩下的一位。

在接收的时候，从SPI总线上接收到的数据会写到偏离缓存。时钟按照SPI总线时钟同步。数据会从偏离缓存移位输出到输入FIFO，按照SPI外设时钟同步。

### 6.14.4 基本配置

SPI0 的基本配置如下:

- 通过设置 P2_MFP 和 P5_MFP 寄存器设置相应的GPIO为SPI0管脚。
- 设置 SPI0_S (CLKSEL1[4])寄存器来选择SPI0的时钟源。
- 设定SPI0_EN (APBCLK[12])寄存器使能SPI0外设时钟。
- 设置SPI0_RST (IPRSTC2[12])寄存器来复位SPI0控制器。

SPI1的基本配置如下:

- 通过设置 P9_MFP寄存器来设定响应GPIO为SPI1管脚。
- 设置SPI1_S (CLKSEL1[5])寄存器来选择SPI1的时钟源。
- 设定 SPI1_EN (APBCLK[13])寄存器来使能SPI1外设时钟。
- 设置SPI1_RST (IPRSTC2[13])寄存器来复位SPI1控制器。

SPI2 的基本配置如下:

- 通过设置 P2_MFP 和 P5_MFP 寄存器来设置响应GPIO位SPI2管脚。
- 设置SPI2_S (CLKSEL1[6])寄存器来选择SPI2的时钟源。
- 设置SPI2_EN (APBCLK[14])寄存器来使能SPI2外部时钟
- 设置SPI2_RST (IPRSTC2[14])寄存器来复位SPI2控制器。

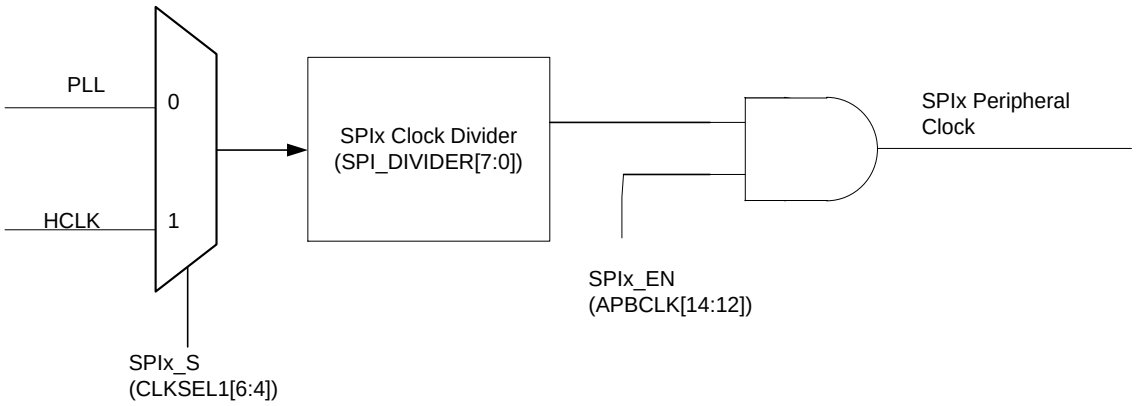
### 6.14.5 功能描述

#### *6.14.5.1 术语*

**SPI外设时钟和 SPI 总线时钟**

SPI控制器需要SPI外设时钟来驱动SPI逻辑单元执行数据传输。SPI外设时钟通过设置SPI时钟源和分频来设置。CLKSEL1寄存器的SPIx_SS位决定SPI外设时钟的时钟源。 时钟源可以选择HCLK或者PLL的输出。SPI_DIVIDER寄存器的DIVIDER设定决定时钟分频。

主模式下，SPI总线的输出时钟与SPI外设时钟一样。从模式下，SPI总线时钟由外部主机提供。从模式下SPI外设时钟要比外部主机提供的SPI总线时钟快。不管在主模式还是从模式下，SPI外设时钟都不能比APB总线的时钟快。




Note1: x = 0, 1, 2

Note2: Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

**主/从模式**

SPI控制器可以通过设置SLAVE 位(SPI_CNTRL[18])来设定工作在主模式或从模式来与外设主设备或从设备来通讯。主模式或从模式的应用框图如图 6-110 和 图 6-111所示

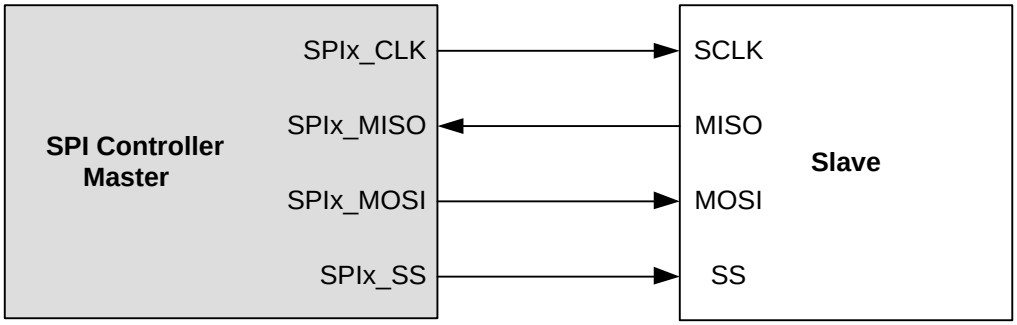



图 6-110 SPI 主模式应用框图

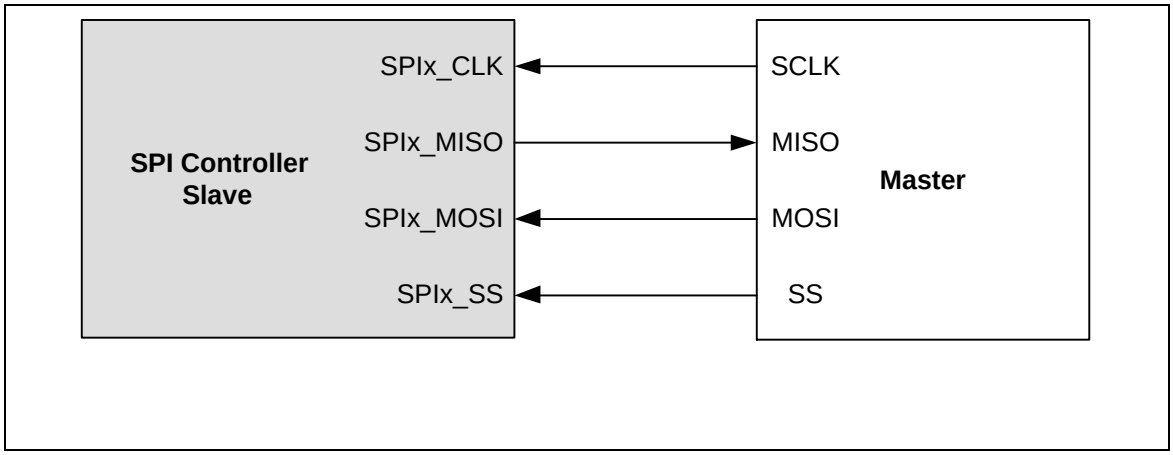



图 6-111 SPI 从模式应用框图

**从机片选**

主模式下，SPI控制器可以通过SPIx_SS 管脚输出从机片选信号驱动外部从设备。从模式下，外部主设备通过SPIx_SS 管脚输入从机片选到SPI控制器。从机片选信号的有效沿到第一个SPI时钟输入之间的间隔至少要3个SPI外设时钟的长度。

在主从模式下，根据外部设备的不同，可以通过设定SS_LVL (SPI_SSR[2])来设置从机片选是高电平有效还是低电平有效。

在从模式下，可以设定SS_LTRIG (SPI_SSR[4]) 来设置SPIx_SS 信号是电平触发或是边沿触发。如果设置为电平触发，LTRIG_FLAG 位 (SPI_SSR[5])用来指示一次传输接收到的数据是否满足TX_BIT_LEN (SPI_CNTRL[7:3])所设定的长度。为了能区分出片选无效电平，在两次成功的传输直接，无效电平至少保持3个SPI外设时钟以上。

**电平触发/边沿触发**

从模式下，从机片选信号可以选择电平触发或变压触发。边沿触发，数据传输从有效沿开始到无效沿结束。单元传输中断标志 IF (SPI_CNTRL[16]) 当检测到片选无效沿后会置1。如果主机没有发出无效沿，那么传输就不会结束，单元传输中断标志也不会置位。电平触发，从机单元传输中断标志会在下面两种情况下置位：第一种情况是传输的长度达到 TX_BIT_LEN (SPI_CNTRL[7:3])所设定的值。第二种情况是片选信号变成无效电平，这会中断目前的传输而不管已经传输了多少位，单元传输中断标志会置位。用户可以读LTRIG_FLAG 位的状态来判断传输是否结束。

**时序设定**

通过 CLKP (SPI_CNTRL[11]) 来设定SPI时钟信号的空闲电平。如果 CLKP = 1，输出SPI总线时钟空闲的时候是高电平。如果 CLKP = 0,空闲的时候是低电平。

TX_NEG (SPI_CNTRL[2]) 设置发送时通过SPI总线时钟的上升沿发送或是下降沿发送。

RX_NEG (SPI_CNTRL[1]) 设置接收数据通过SPI总线时钟的上升沿或下降沿接收。

**注意:** TX_NEG 和 RX_NEG 的设定是互斥的，就是说不要设置发送和接收在一个沿上。

**发送接收位长**

一次传输的位数有TX_BIT_LEN (SPI_CNTRL[7:3])来设置。一次发送或接受最多可以传输32位的数据。

当SPI控制器完成一次传输，如发送或接收的数据达到TX_BIT_LEN (SPI_CNTRL[7:3])所定义的长度，单元传输中断标志会置1.

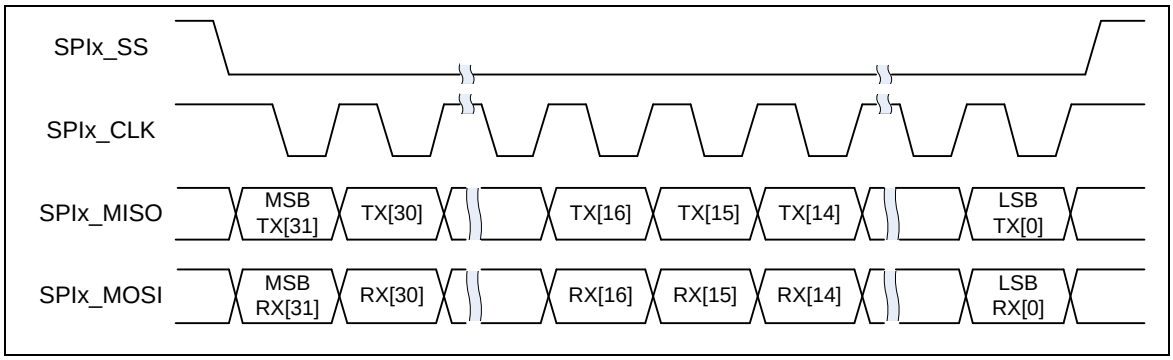



图 6-112 一次传输32位

**高位/低位先发**

LSB (SPI_CNTRL[10]) 设定位传输的顺序。如果 LSB 位是 1，传输顺序是低位先发，bit 0会首先传输。如果LSB 位是 0，传输顺序是高位先发。

### *6.14.5.2 自动片选*

主模式下，如果AUTOSS (SPI_SSR[3])位置1，如果SSR (SPI_SSR[0]) 位使能，会自动产生片选输出到 SPIx_SS 管脚上。当写数据到TX FIFO （使能FIFO模式）或设定GO_BUSY (SPI_CNTRL[0])（禁止FIFO模式）时，片选信号会自动产生。当SPI总线空闲的时候，片选信号会自动变为无效状态。当总线不是空闲，例如当FIFO模式，输出移位寄存器或是输出偏离缓存非空，片选信号会在两次传输直接变为无效状态，如果 SP_CYCLE (SPI_CNTRL[15:12])设定值大于等于3.

主模式下，如果SP_CYCLE[3:0] 的值小于 3 并且 AUTOSS为 1，片选信号在两次成功的传输之间会保持有效状态。

如果AUTOSS 位清0，片选信号的状态通过程序设置SSR (SPI_SSR[0])来决定。片选信号的有效状态由SS_LVL 位 (SPI_SSR[2])来确定。

从模式下，为了区分出无效状态，在两次成功的传输之间，片选无效状态的时间必须大于等于6个SPI外设时钟的长度。

从片选信号的有效沿到SPI总线时钟的第一个时钟沿之间需要有1个SPI总线时钟的延迟。从最后一个SPI总线时钟到片选信号变为无效需要有1.5个SPI总线时钟。.

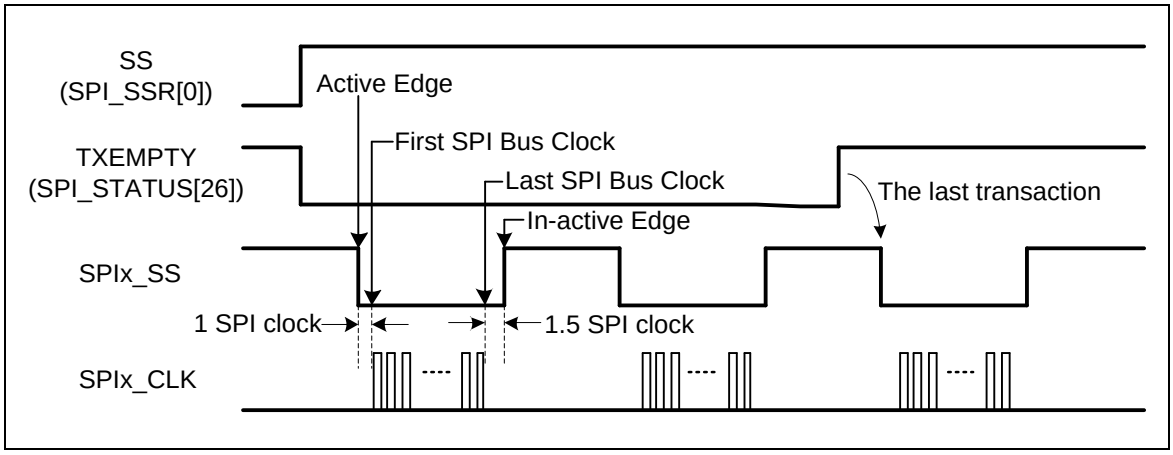



图 6-113 自动片选 (SS_LVL = 0, SP_CYCLE > 0x2)

#### *6.14.5.3 字间隙*

主模式下两次成功的传输之间，SP_CYCLE (SPI_CNTRL[15:12]) 设定间隙的长度， 0.5 ~ 15.5 个串行时钟周期。间隙是指上一次传输的最后一个时钟沿到这次传输的第一个时钟沿。SP_CYCLE 的默认值是0x3 (3.5个串行时钟周期)。如果禁止了FIFO模式，那么SP_CYCLE 的设置是无效的。

#### *6.14.5.4 字节重排序*

如果传输设置为高位先发(LSB (SPI_CNTRL[10])= 0) ，并且 REORDER (SPI_CNTRL[19]) 设为 1，当32位传输模式(TX_BIT_LEN = 0)时，发送缓存和接收缓存里的数据会重新排列为 [Byte0, Byte1, Byte2, Byte3] 。发送或接收的数据顺序为 Byte0, Byte1, Byte2, and then Byte3。 如果 TX_BIT_LEN 设为 24-bit 传输模式， 发送或接收缓存里的数据会重新排列为 [unknown byte, Byte0, Byte1, Byte2]。SPI控制器发送接收数据的顺序改为 Byte0, Byte1 and then Byte2。每个字节发送接收的顺序都是高位先发。 16-bit模式的规则和上面一样。字节重排序功能只当 TX_BIT_LEN (SPI_CNTRL[7:3]) 设为 16, 24, or 32 位时有效。

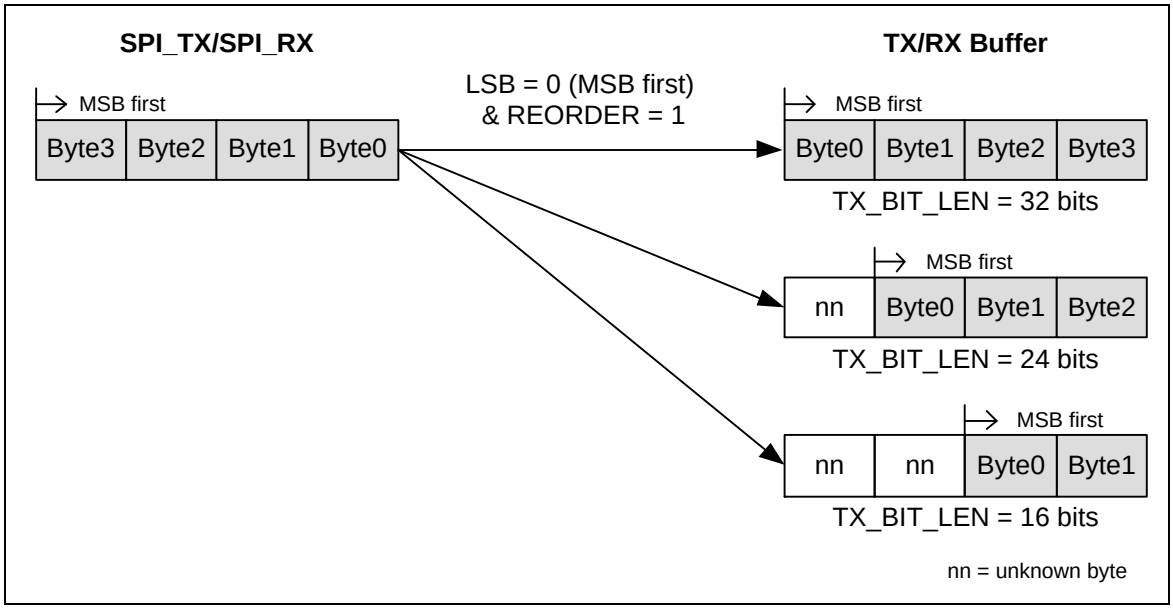



图 6-114 字节重排序功能

#### 6.14.5.5 字节间隙

主模式下，如果REORDER (SPI_CNTRL[19]) 设为1, 在一次传输的每两个成功传输的字节之间会插入0.5 ~ 15.5 个SPI 串行时钟的间隙。字节间隙或字间隙的长度都由SP_CYCLE (SPI_CNTRL[15:12])来决定。

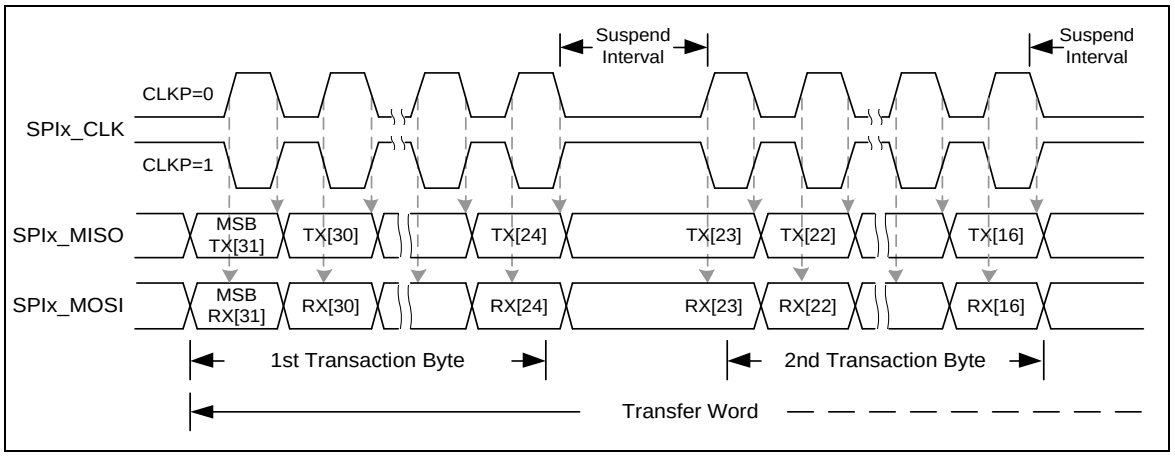



图 6-115 字节间隙时序图

#### 6.14.5.6 从机三线模式

当 NOSLVSEL (SPI_CNTRL2[8]) 设为1就使能从机三线模式，SPI控制器在从模式下可以没有片选信号。NOSLVSEL 位只在从机模式下有效。只有三根管脚SPIx_CLK, SPIx_MISO, 和SPIx_MOSI, 与SPI主机通讯。 SPIx_SS 可以设置为 GPIO。当 NOSLVSEL 和GO_BUSY (SPI_CNTRL[0])都设为1，SPI从机就准备好接收或发送数据。当接收到的数据等于TX_BIT_LEN (SPI_CNTRL[7:3]), 单元传输中断标志 IF (SPI_CNTRL[16]) 会设为 1。

**注意:** 三线模式下, SS_LTRIG, SPI_SSR[4], 需要设为 1.

#### 6.14.5.7 FIFO 模式

SPI 控制器当FIFO (SPI_CNTRL[21])设为1后支持FIFO模式。 SPI 控制器有接收和发送各8个32位宽的缓存。

发送FIFO 缓存是8级深度的，32位宽的，先进先出的缓存。程序可以通过写 SPI_TX寄存器写数据到发送缓存。 发送FIFO里的数据会由发送控制逻辑单元读出。如果8级深度的FIFO 满了, TX_FULL (SPI_STATUS[27]) 位会置 1。如果SPI发送逻辑单元传输出最后一笔数据，8级深度的FIFO就空了， TX_EMPTY (SPI_STATUS[26]) 位会置1。注意当 TX_EMPTY标志 (SPI_STATUS[26])置1最后一笔传输仍在进行。主模式下， GO_BUSY (SPI_CNTRL[0]) 和 TX_EMPTY (SPI_STATUS[26]) 都会确认来确定SPI传输时空闲或非空。

接收 FIFO 缓存也是8级深度的，32位宽的，先进先出的缓存。接收逻辑单元会存数据到这个缓存。程序可以读SPI_RX寄存器从缓存里读数据。 FIFO 相关的状态位, 如 RX_EMPTY (SPI_STATUS[24]) 何 RX_FULL (SPI_STATUS[25]), 指示 FIFO 相关状态。

可以通过设定TX_THRESHOLD (SPI_FIFO_CTL[30:28]) RX_THRESHOLD (SPI_FIFO_CTL[26:24])寄存器来设置接收和发送的阈值。当发送 FIFO里的数据少于 TX_THRESHOLD的设定时, tTX_INTSTS (SPI_STATUS[4]) 会置 1.。当接收FIFO里的有效数据多于RX_THRESHOLD 设定值时, RX_INTSTS (SPI_STATUS[0]) 会置 1.

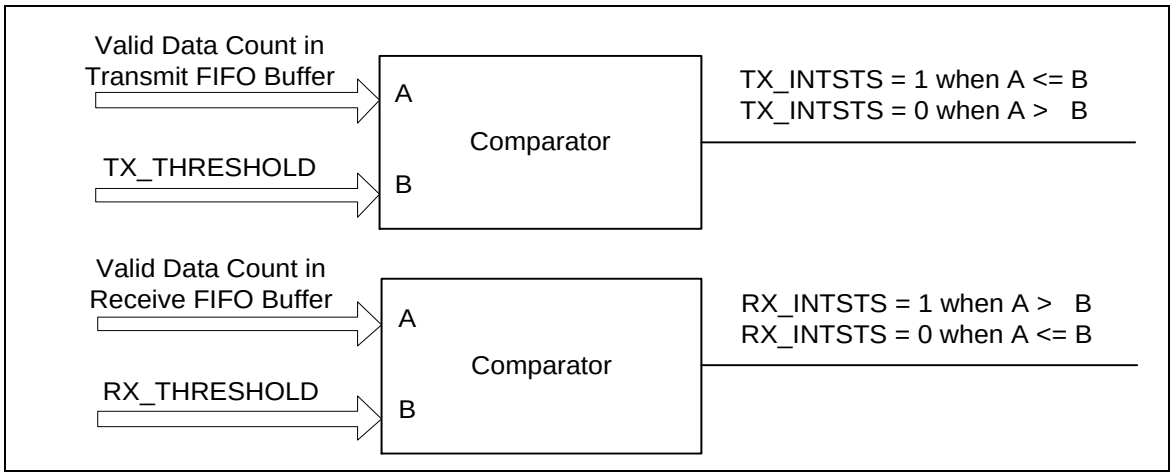



图 6-116 FIFO 阈值判别

FIFO模式下，程序可以连续写8笔数据到发送FIFO里。在FIFO模式下GO_BUSY (SPI_CNTRL[0]) 位由硬件控制。程序不能修改 SPI_CNTRL 寄存器除非清除 FIFO bit (SPI_CNTRL[21]) 位关闭FIFO模式。

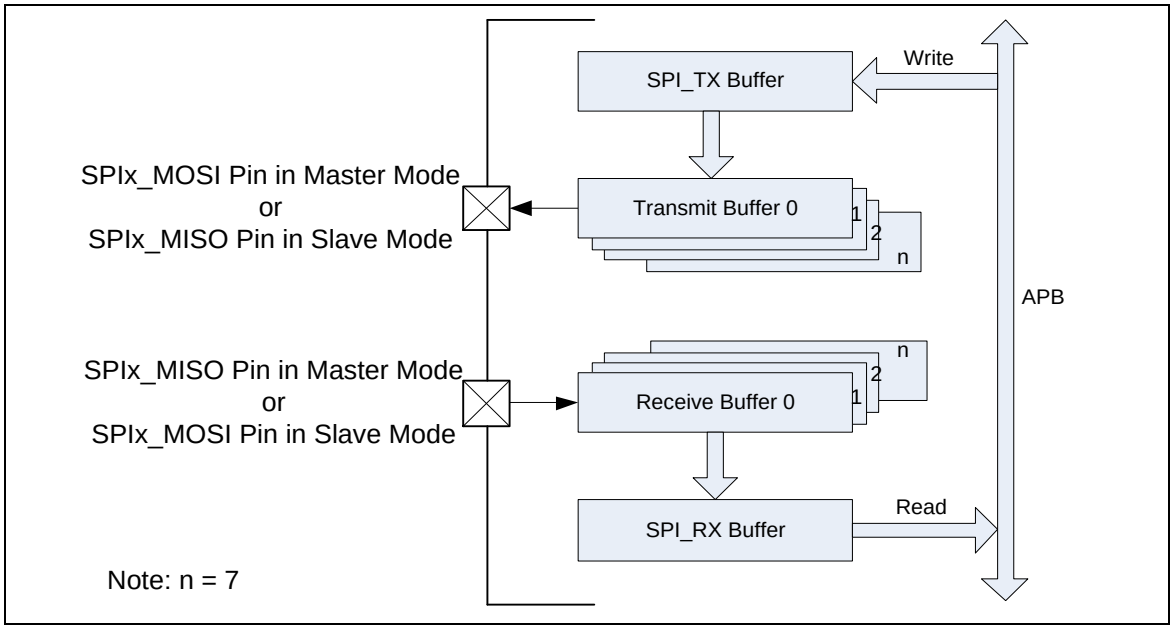



图 6-117 FIFO 模式框图

主模式下，第一笔数据写到SPI_TX寄存器里, TX_EMPTY标志 (SPI_STATUS[16]) 会清零。传输会在一个APB时钟和6个SPI外部时钟后开始。用户可以立即写下一笔数据到SPI_TX寄存器里。SPI控制器会自动在两次成功的传输之间插入时间间隔，时间间隔的长度由 SP_CYCLE (SPI_CNRTL[15:12])决定。 如果 SP_CYCLE (SPI_CNTRL[15:12]) 等于0,SPI控制器可以连续发送。当X_FULL (SPI_STATUS[27])等于0，用户可以一直写数据到SPI_TX寄存器。

如图 6-118的例子1, 展示了TX_EMPTY (SPI_STATUS[26])与FIFO缓存，移位寄存器，偏离缓存之间的关系。当Data0写到FIFO缓存里后，TX_EMPTY (SPI_STATUS[16]) 会清0。 Data 0载入移位寄存器后 TX_EMPTY (SPI_STATS[26]) 会置1. Data 0在移位寄存器里会移位到偏离寄存器里直到传输完成。

例子2中，展示当FIFO缓存里已经有8笔数据，TX_FULL (SPI_STATUS[27]) 会置1.当TX_FULL = 1时第9笔数据不会写进去。

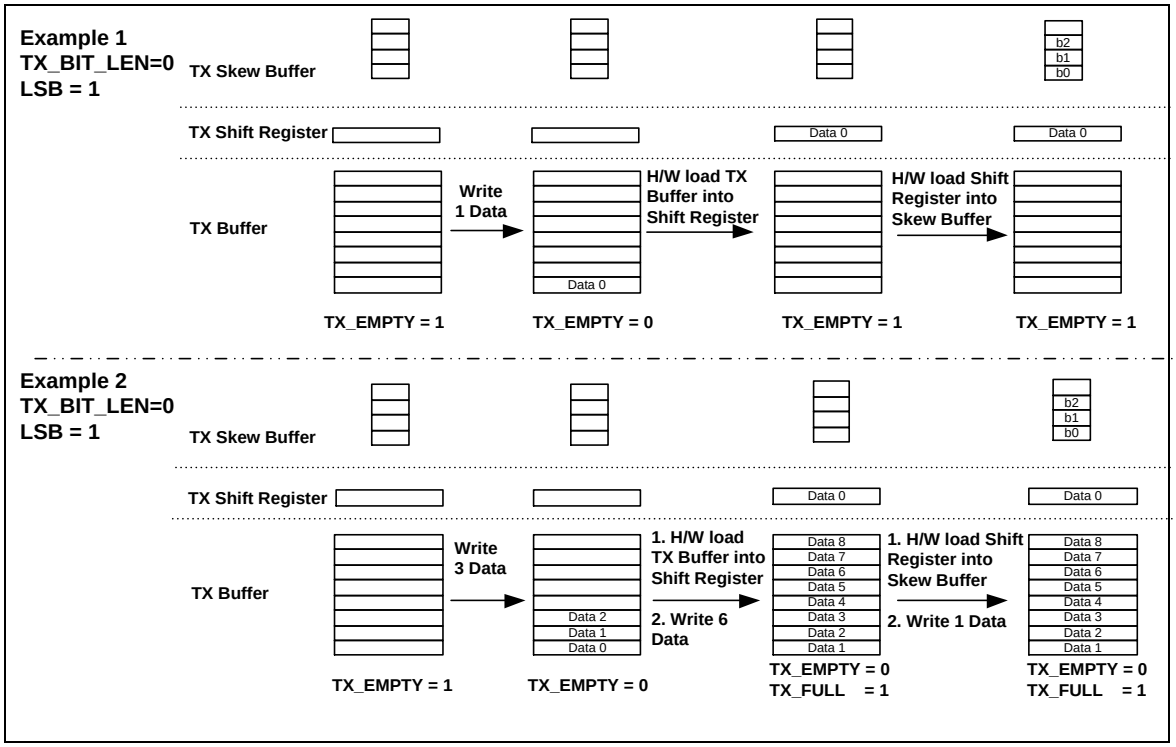



图 6-118 发送 FIFO 缓存实例

如果数据及时写入SPI_TX寄存器，接下来的传输会持续进行。如果再没有数据写入SPI_TX 寄存器了，传输会停止。

主模式下，接收的数据会通过SPIx_MISO 管脚输入，存入接收FIFO缓存。

接收到的数据 (Data 0's b0, b1, …, b31) 同步于SPI串行时钟 (SPIx_CLK) 存入到偏离缓存里，然后一位一位的移到移位寄存器里。当数据长度达到TX_BIT_LEN (SPI_CNTRL[7:3])的设定，内核逻辑单元从移位寄存器把数据载入到FIFO缓存里。当FIFO里有未读数据时，RX_EMPTY (SPI_STATUS[24]) 会清0 (见下图例子1)。程序可以一直通过 SPI_RX 寄存器读数据直到 RX_EMPTY (SPI_STATUS[24]) 为 0。如果接收FIFO里有8个未读数据， RX_FULL (SPI_STATUS[25]) 会置 1 (见下图例子2)

从模式下，如果FIFO 位设为1, GO_BUSY会由硬件自动置1。如果用户想停止SPI数据传输，需要程序清 FIFO 位和 GO_BUSY 位为0。

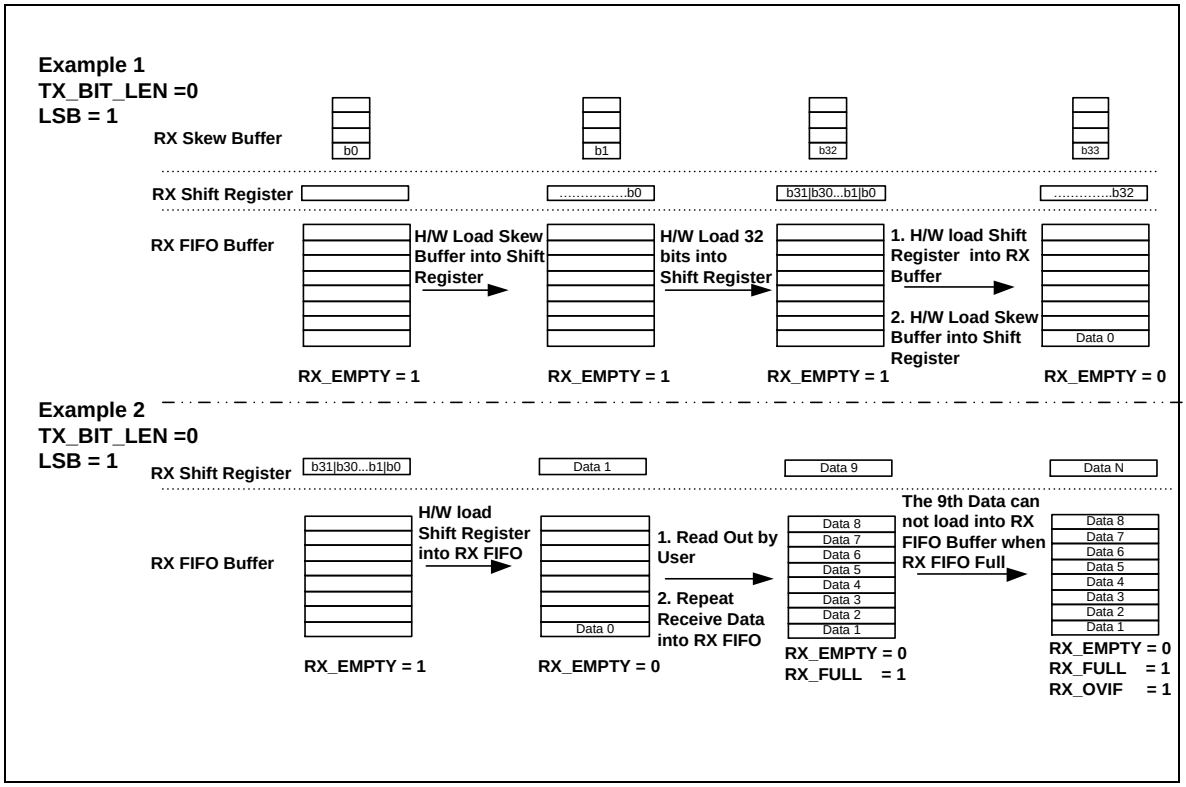



图 6-119 接收FIFO 缓存示例

从模式下，传输过程中，当程序写数据到 SPI_TX 寄存器，数据会写入发送FIFO缓存，并且TX_EMPTY 标志会清0。当主机发来时钟信号数据会发送出去。程序可以在TX_FULL 标志为0时持续向SPI_TX寄存器写入数据。 当所有数据都通过SPI发送逻辑单元传送出去，并且程序没有再更新SPI_TX 寄存器, TX_EMPTY 标志会置 1.

从模式下接收数据，串行数据通过 SPIx_MOSI管脚输入存入SPI_RX 寄存器。接收机制与主模式下的接收流程一样。

*6.14.5.8 中断*

■ SPI 单元传输中断

当SPI控制器完成了一个单元的传输，单元传输中断标志 (SPI_CNTRL[16]) 会置 1。当单元传输中断使能位IE (SPI_CNTRL[17])为1，单元传输中断时间会向cpu请求一次中断。单元传输中断标志只能通过程序写1清0。

■ SPI 从机三线模式开始中断

在三线模式下，当从机收到SPI串行时钟后从机三线模式开中断标志SLV_START_INTSTS (SPI_STATUS[11])会置1。如果SSTA_INTEN (SPI_CNTRL2[10]) 位1，SPI控制器会产生中断。如果接收到的数据比TX_BIT_LEN (SPI_CNTRL[7:3])的设定少，并且在用户设定的时间内再没有SPI串行时钟的输入， 用户可以设定SLV_ABORT (SPI_CNTRL2[9]) 来中断目前的传输。如果SLV_ABORT (SPI_CNTRL2[9])置1，单元传输中断标志 IF, (SPI_STATUS[16]) 会置1。.

■ 接收 FIFO 超时中断

FIFO模式下。有超时机制来提醒用户。 如果FIFO中有数据在64个SPI外设时钟（主模式）或576个SPI外设时钟（从模式）没有读出，如果TIMEOUT_INTEN (FIFO_CTL[21])设为1会产生一个中断。

■ 发送 FIFO 中断

FIFO模式下, 发送FIFO里的有效数据少于TX_THRESHOLD里的设定, 发送FIFO 中断标志会置1。如果TX_INTEN (SPI_FIFO_CTL[3]) 设为1，SPI控制器会产生一个发送FIFO中断。

■ 接收 FIFO 中断

FIFO模式下，如果接收FIFO里的数据多于 RX_THRESHOLD的设定值, 接收 FIFO 中断标志会置1。如果RX_INTEN (SPI_FIFO_CTL[2]) 设为1，SPI控制器会产生一个接收FIFO中断。

### 6.14.6 时序图

从机片选信号的有效状态由SS_LVL 位(SPI_SSR[2]) 和 SS_LTRIG 位 (SPI_SSR[4])设置. SPI时钟空闲时是高电平或低电平由 CLKP 位 (SPI_CNTRL[11])设置. TX_BIT_LEN (SPI_CNTRL[7:3])设置一次传输的位数，LSB 位 (SPI_CNTRL[10]) 设置发送或接收高位先发还是低位先发. TX_NEG/RX_NEG (SPI_CNTRL[2:1])设置发送接收数据的有效沿。4中主从模式和相关设定的时序图详见 图 6-120, 图 6-121, 图 6-122 和 图 6-123.

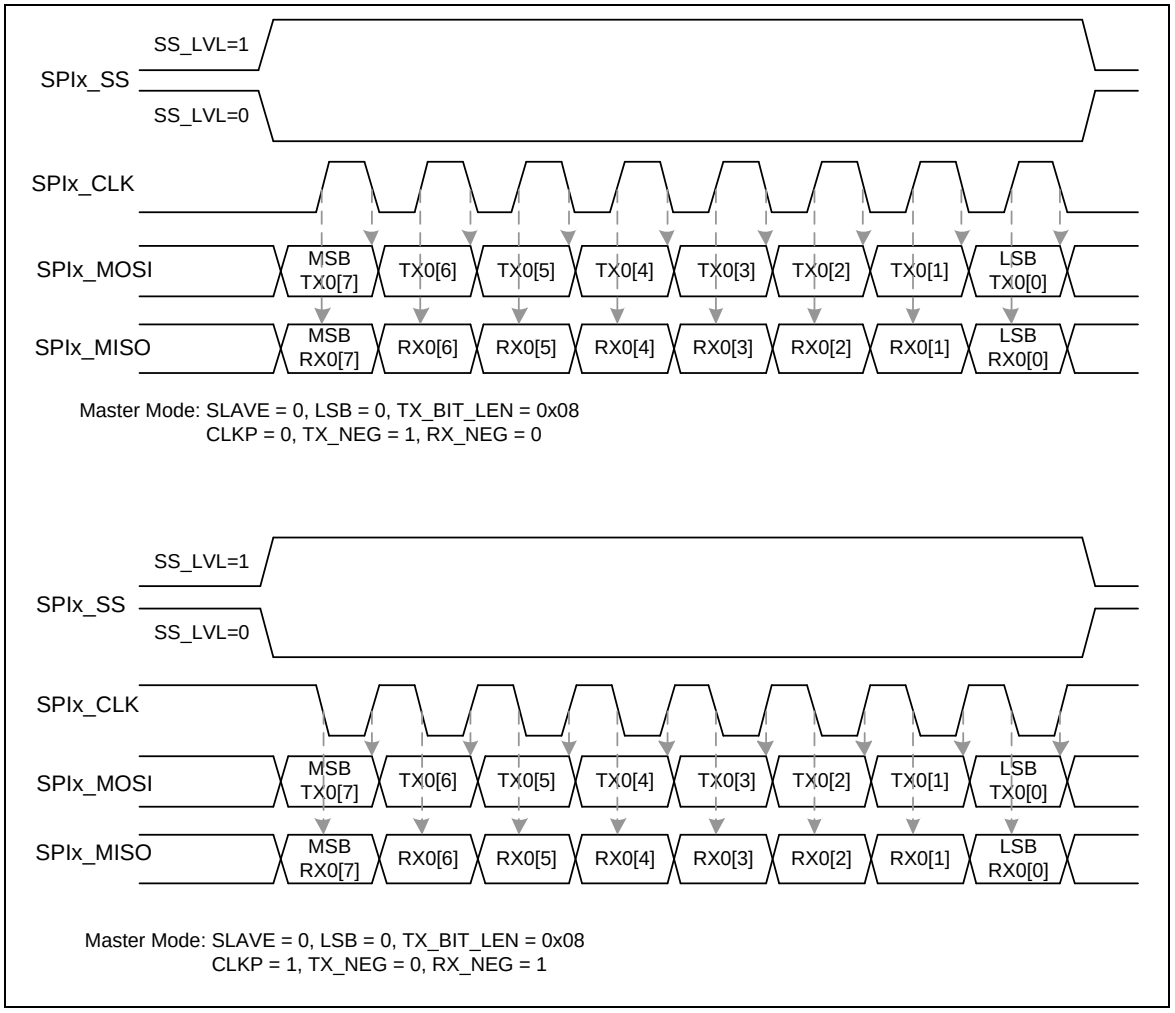



图 6-120 SPI 主模式时序

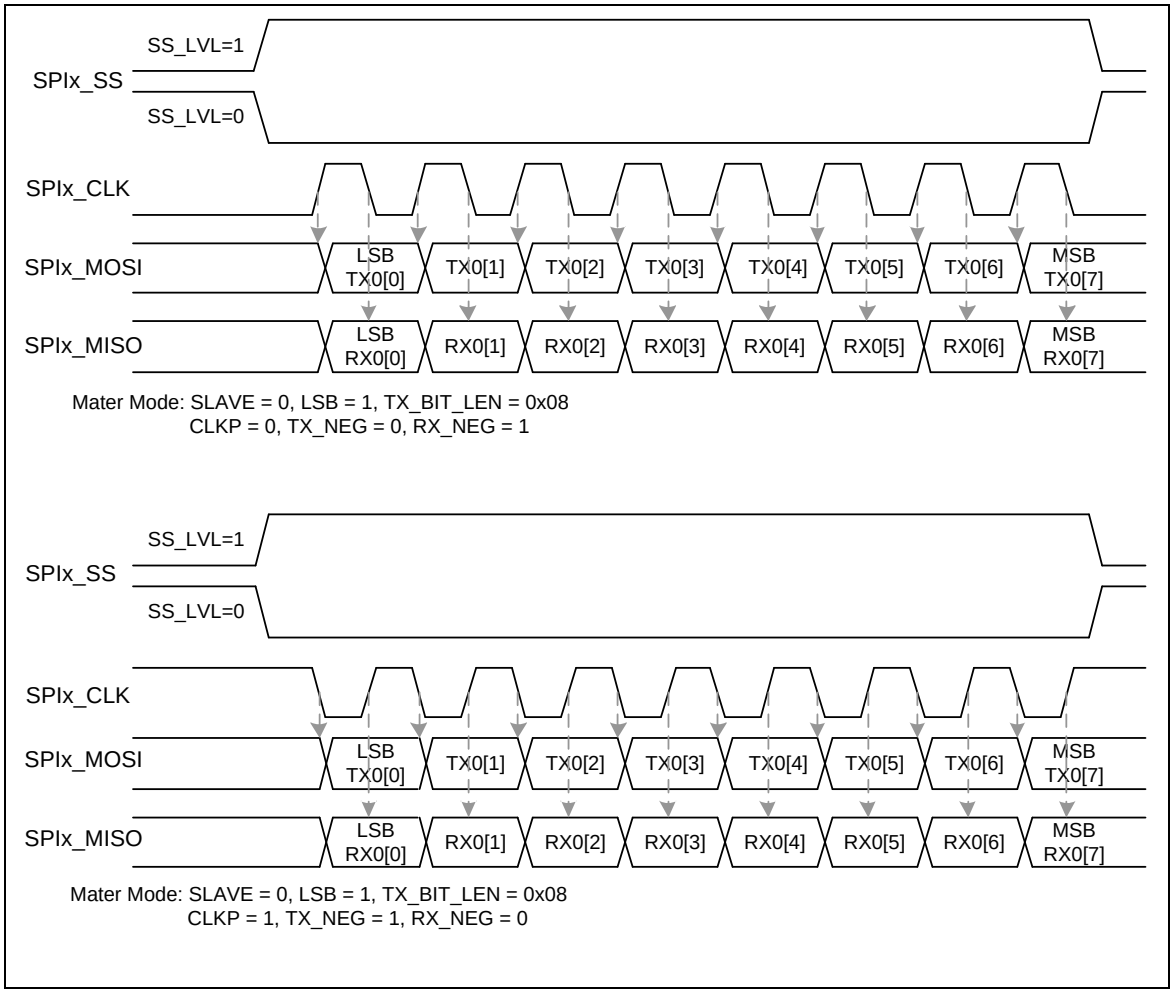



图 6-121 SPI 主模式时序图 (变换SPI_CLK相位)

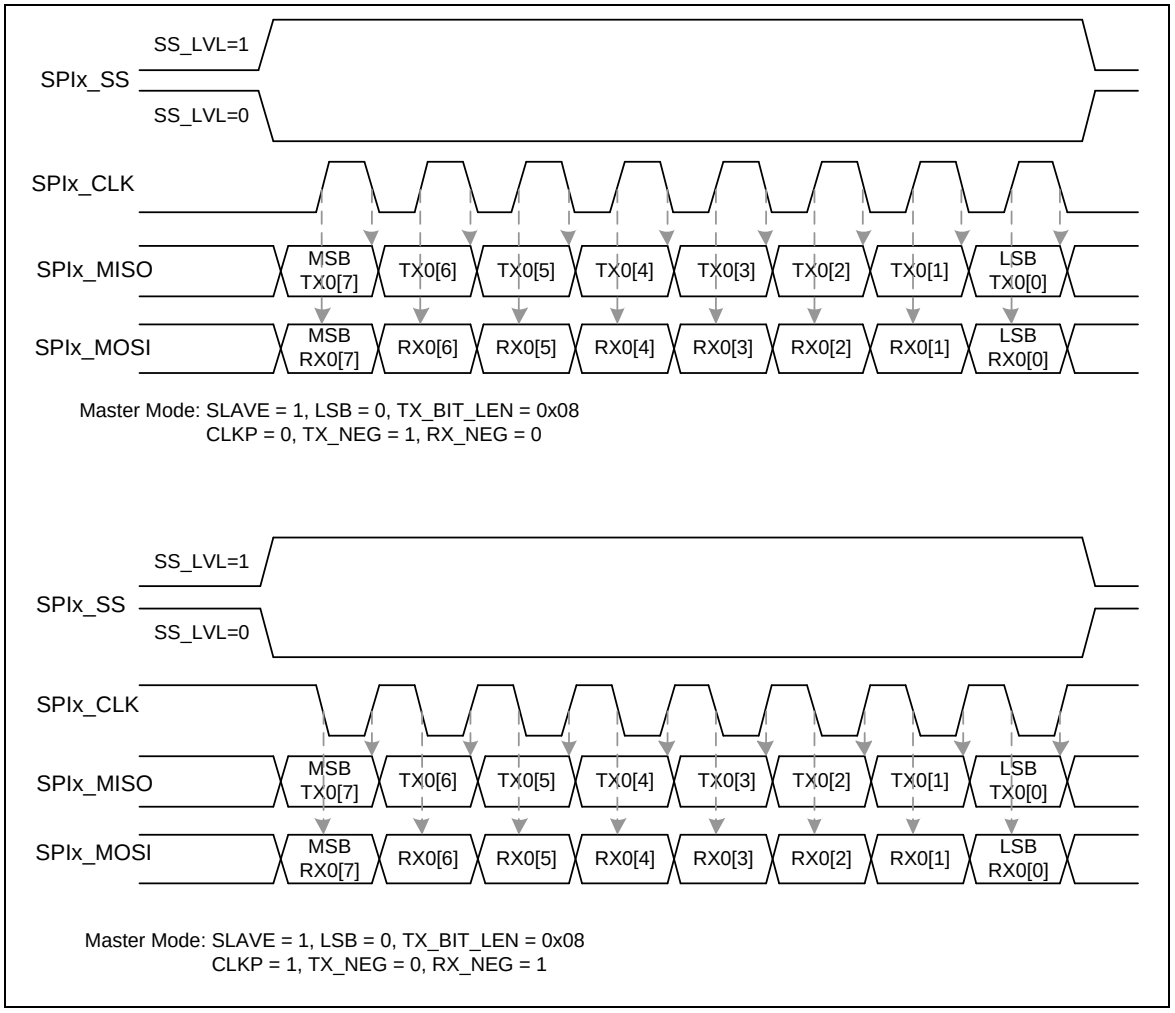



图 6-122 SPI 从模式时序

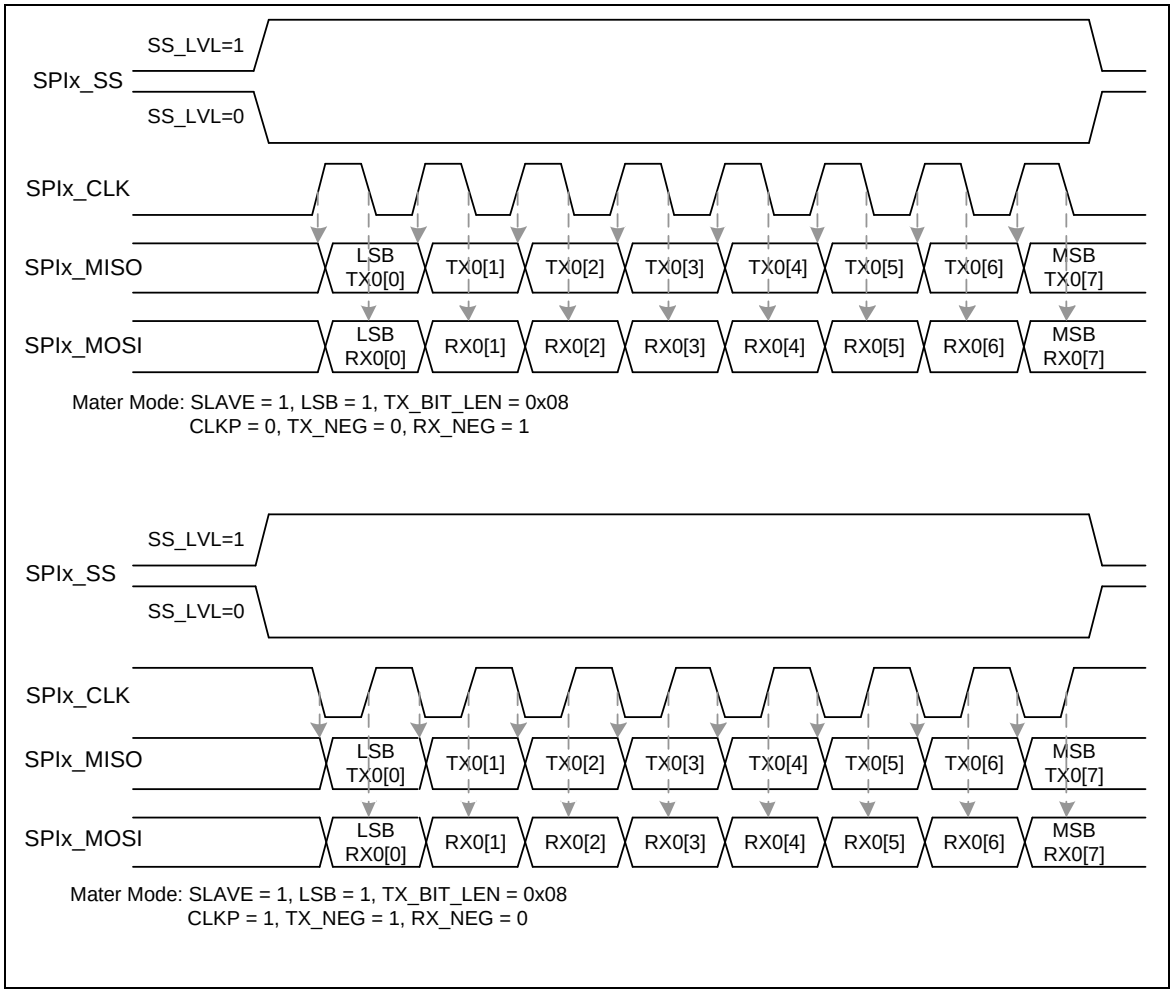



图 6-123 SPI 从模式时序 (变换SPI时钟相位)

### 6.14.7 编程示例

**例子 1:** SPI控制器工作在主模式，依照下面的规范访问外部从设备:

- 数据在 SPI 总线时钟上升沿锁存。
- 数据 SPI总线时钟下降沿发送
- 数据高位先发
- SPI 空闲时为低电平
- 一次发送或接收传输1个字节
- 从机片选信号低有效。

操作流程如下

1) 设定 DIVIDER (SPI_DIVIDER[7:0]) 寄存器设定SPI总线时钟。
2) 写SPI_CNTRL 相应位设置SPI工作状态。

1. 设定 SPI 工作在主机模式. SLAVE (SPI_CNTRL[18]) = 0.
2. 设定 SPI 总线时钟空闲时为低电平. CLKP (SPI_CNTRL[11]) = 0.
3. 在SPI总线时钟下降沿发送数据. TX_NEG (SPI_CNTRL[2]) = 1.
4. 接收数据在SPI总线时钟上升沿. RX_NEG (SPI_CNTRL[1]) = 0.
5. 设定一次传输8位. TX_BIT_LEN (SPI_CNTRL[7:3]) = 0x08.
6. 设定传输顺序高位先发. LSB (SPI_CNTRL[10]) = 0.

3) 写 SPI_SSR 相应位设置状态:
   1. 禁止自动片选功能. AUTOSS (SPI_SSR[3]) 写 0.
   2. 设定从机片选低电平有效. SS_LVL (SPI_SSR[2]) = 0， SS_LTRIG (SPI_SSR[4]) = 1.
   3. SSR (SPI_SSR[0]) 写1使能外部从设备。
4) 当SPI主设备要写数据到外部从设备，写数据到SPI_TX 寄存器。
5) 如果SPI主设备只是要从外部从设备读数据，而不管发送什么数据到从设备，可以不用去更新 SPI_TX 寄存器.
6) 设置 GO_BUSY (SPI_CNTRL[0]) 为1 开始SPI接口数据传输.
7) 当使能了单元传输中断时等待中断发生，或者查询 GO_BUSY 位 (SPI_CNTRL[0]) 直到它自动清0.
8) 从SPI_RX[7:0]寄存器读出数据.
9) 跳到 4) 开始下一次传输 或者设定 SSR 为 0停止外部片选.

**例子 2:** SPI控制器工作在从模式与外部主机通讯。依照下面的规范访问:

- 数据在SPI总线时钟上升沿接收。
- 数据在SPI总线时钟下降沿发送。
- 数据传输低位先发。
- SPI 空闲时为高
- 一次传输一个字节。
- 片选信号高电平有效

操作流程如下：

1) 写合适的值到 SPI_SSR 寄存器设定从模式相关配置

   写SS_LVL (SPI_SSR[2])为1， SS_LTRIG (SPI_SSR[4])为1设置片选信号为高电平触发
2) 写合适的值到SPI_CNTRL寄存器配置相关特性
   1. 设置SPI工作在从模式. SLAVE (SPI_CNTRL[18]) = 1.
   2. 设置SPI时钟空闲时为高 CLKP (SPI_CNTRL[11]) = 1.
   3. 下降沿发送数据. TX_NEG (SPI_CNTRL[2])= 1.
   4. 上升沿接收数据. RX_NEG (SPI_CNTRL[1]) = 0.

5. 设置一次传输8位. TX_BIT_LEN (SPI_CNTRL[7:3])= 0x08.
6. 设置传输顺序低位先发 LSB (SPI_CNTRL[10]) = 1.

3) 如果想发送数据给外部SPI主设备，写SPI_TX 寄存器.
4) 如果只是想从外部SPI主设备读数据，而不管输出的数据是什么，可以不用更新SPI_TX 寄存器.
5) GO_BUSY (SPI_CNTRL[0]) 写1 ，等待片选信号和时钟信号的输入
6) 如果使能了单元传输中断，那么等待中断产生，或者查询GO_BUSY bit (SPI_CNTRL[0]) 直到被硬件自动清0.
7) 从 SPI_RX[7:0]读出接收到的数据.
8) 跳到3) 开始下一次传输，或者清除GO_BUSY位结束传输

### 6.14.8 寄存器列表

**R**: 只读, **W**: 只写, **R/W**: 可读写

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **SPI 基地址:**<br>**SPI0_BA = 0x4003_0000**<br>**SPI1_BA = 0x4003_4000**<br>**SPI2_BA = 0x4013_0000** | | | | |
| **SPI_CNTRL**<br>**x=0,1,2** | SPIx_BA+0x00 | R/W | 控制和状态寄存器 | 0x0500_3004 |
| **SPI_DIVIDER**<br>**x=0,1,2** | SPIx_BA+0x04 | R/W | 时钟分频寄存器 | 0x0000_0000 |
| **SPI_SSR**<br>**x=0,1,2** | SPIx_BA+0x08 | R/W | 片选寄存器 | 0x0000_0000 |
| **SPI_RX**<br>**x=0,1,2** | SPIx_BA+0x10 | R | 接收数据寄存器 | 0x0000_0000 |
| **SPI_TX**<br>**x=0,1,2** | SPIx_BA+0x20 | W | 发送数据寄存器 | 0x0000_0000 |
| **SPI_CNTRL2**<br>**x=0,1,2** | SPIx_BA+0x3C | R/W | 控制和状态寄存器2 | 0x0000_1000 |
| **SPI_FIFO_CTL**<br>**x=0,1,2** | SPIx_BA+0x40 | R/W | SPI FIFO 控制寄存器 | 0x4400_0000 |
| **SPI_STATUS**<br>**x=0,1,2** | SPIx_BA+0x44 | R/W | SPI 状态寄存器 | 0x0500_0000 |

### 6.14.9 寄存器描述

**SPI 控制和状态寄存器 (SPI_CNTRL)**

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **SPI_CNTRL** | SPIx_BA+0x00 | R/W | 控制和状态寄存器 | 0x0500_3004 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | TX_FULL | TX_EMPTY | RX_FULL | RX_EMPTY |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | FIFO | Reserved | REORDER | SLAVE | IE | IF |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SP_CYCLE | | | | CLKP | LSB | Reserved | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX_BIT_LEN | | | | | TX_NEG | RX_NEG | GO_BUSY |

| 位 | 描述 | |
|---|---|---|
| [31:28] | | 保留 |
| [27] | **TX_FULL** | **发送FIFO 满指示 (只读)**<br>SPI_STATUS[27]的镜像位.<br>0 = 发送FIFO缓存不满.<br>1 = 发送 FIFO缓存满了. |
| [26] | **TX_EMPTY** | **发送 FIFO缓存空指示(只读)**<br>SPI_STAUTS[26]的镜像位.<br>0 = 发送tFIFO缓存非空.<br>1 =发送FIFO缓存空了. |
| [25] | **RX_FULL** | **接收 FIFO缓存满指示 (只读)**<br>SPI_STATUS[25]的镜像位.<br>0 = 接收 FIFO缓存没满.<br>1 = 接收FIFO 缓存满了. |
| [24] | **RX_EMPTY** | **接收 FIFO 缓存空指示 (只读)**<br>SPI_STATUS[24]的镜像位.<br>0 =接收 FIFO缓存非空.<br>1 =接收 FIFO缓存空了. |
| [23:22] | | 保留 |
| [21] | **FIFO** | **FIFO 模式使能位**<br>0 = FIFO 模式禁止.<br>1 = FIFO 模式使能.<br>**注意:** |

| | | |
|---|---|---|
| | | 1. 使能FIFO模式之前，其他相关配置需要先配置好。.<br>2. 主模式下，如果是能了FIFO模式，当写数据到FIFO缓存。GO_BUSY位会自动置1.当SPI控制器空闲的时候， GO_BUSY位会自动清0。如果FIFO里所以的数据都发送出去了，TX_EMPTY 位会置 1 GO_BUSY 位会清0. |
| [20] | | 保留 |
| [19] | **REORDER** | **字节重排序使能位**<br>0 =禁止字节重排序.<br>1 =使能字节重排序。每两个字节间会插入字节间隔，间隔长度由 SP_CYCLE.决定<br>**注意:**<br>1. 字节重排序功能只在TX_BIT_LEN 设为 16, 24, 和 32 位时有效.<br>2. 如果从机模式下，电平触发，在字节间隙时间内，片选信号必须保持使能状态。 |
| [18] | **SLAVE** | **从机模式使能位**<br>0 = 主机模式.<br>1 = 从机模式. |
| [17] | **IE** | **单元传输中断使能位**<br>0 = SPI单元传输中断禁止<br>1 = SPI 单元传输中断使能. |
| [16] | **IF** | **单元传输中断标志**<br>0 = 传输没有完成此位为 0.<br>1 = SPI 控制器完成了一单元的传输.<br>此位写1清0. |
| [15:12] | **SP_CYCLE** | **间隔时间 (Master Only)**<br>这4位设置在两次成功的发送或接收之间插入的间隔的长度。间隔长度是指上一笔最后一个时钟源到下笔的第一个时钟源。. 默认值是 0x3. 间隔时间计算方法按下面方法计算：<br>(SP_CYCLE[3:0] + 0.5) * SPI时钟周期<br>示例:<br>SP_CYCLE = 0x0 …. 0.5 SPI 时钟周期.<br>SP_CYCLE = 0x1 …. 1.5 SPI时钟周期.<br>……<br>SP_CYCLE = 0xE …. 14.5 SPI时钟周期.<br>SP_CYCLE = 0xF …. 15.5 SPI时钟周期. |
| [11] | **CLKP** | **时钟极性**<br>0 = SPI 总线时钟空闲时为低.<br>1 = SPI 总线空闲时为高. |
| [10] | **LSB** | **低位先发**<br>0 = TX_BIT_LEN 中所设置的发送或接收中的最高位会首先发送或接收。<br>1 = SPI_TX 发送时位0首先发送出去，第一个接收到的数据会存入SPI_RX 的位0. |
| [9:8] | | 保留. |
| [7:3] | **TX_BIT_LEN** | **传输位数**<br>这几位决定一次发送或接收传输多少位。最少8位，最多32位。<br>TX_BIT_LEN = 0x01~0x07 …. 不能设.<br>TX_BIT_LEN = 0x08 …. 8位. |

| | | |
|---|---|---|
| | | TX_BIT_LEN = 0x09 …. 9 位.<br>……<br>TX_BIT_LEN = 0x1F …. 31 位.<br>TX_BIT_LEN = 0x00 …. 32 位. |
| [2] | **TX_NEG** | **下降沿发送**<br>0 = 在SPI时钟上升沿发送数据.<br>1 = 在SPI时钟下降沿发送数据 |
| [1] | **RX_NEG** | **下降沿接收**<br>0 = 在SPI时钟上升沿接收数据.<br>1 = 在SPI时钟下降沿接收数据 |
| [0] | **GO_BUSY** | **SPI 传输控制和忙状态**<br>如果FIFO模式没有使能，数据传输时这一位保持为1，传输结束后会自动清0。程序可以查询这一位来判断SPI是否在忙状态。<br>在FIFO模式下，这一位由硬件控制，程序不能修改这一位。在从模式下，这一位总是读到1. 主模式下，这一位反映SPI是忙碌或是空闲。<br>0 = 数据传输暂停.<br>1 = 主模式下，写1开始SPI数据传输。从模式下，写1表示已经准备好与主机通讯I<br>**注意:**<br>FIFO模式禁止时，在写1到GO_BUSY 位前，相关配置必须都配置好。 |

## SPI 时钟分频寄存器 (SPI_DIVIDER)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **SPI_DIVIDER** | SPIx_BA+0x04 | R/W | 时钟分频寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:8] | | 保留. |
| [7:0] | **DIVIDER** | **时钟分频器1**<br>这个分频值用来产生SPI外设时钟, $f_{spi_eclk}$, 和主模式下SPI总线时钟。频率值由下面的算式得出<br>$f_{spi_eclk} = \frac{f_{spi_clock_src}}{(DIVIDER+1)*2}$<br>其中<br>$f_{spi_clock_src}$ 是CLKSEL1寄存器定义的SPI外设时钟的时钟源 |

## SPI 片选寄存器 (SPI_SSR)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **SPI_SSR** | SPIx_BA+0x08 | R/W | 片选寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | **LTRIG_FLAG** | **SS_LTRIG** | **AUTOSS** | **SS_LVL** | **Reserved** | **SSR** |

| 位 | 描述 | |
|---|---|---|
| [31:6] | | 保留. |
| [5] | **LTRIG_FLAG** | **电平触发完成标志**<br>从模式下，这一位指示当目前的传输完成后，收到的数据是否达到设置的位数<br>0 = 一次传输发送数据长度没有达到设定的数目<br>1 =传输的数据长度达到 TX_BIT_LEN设定的数目<br>**注意: 这一位只读。**当GO_BUSY 位写1, LTRIG_FLAG会在4个SPI外设时钟和一个系统时钟后清0.在FIFO模式下，这一位没有意义。 |
| [4] | **SS_LTRIG** | **片选电平触发使能位 (仅从机)**<br>0 = 片选信号边沿触发。默认是边沿触发。SS_LVL 位决定是上升沿或是下降沿<br>1 = 片选信号电平触发。SS_LVL决定是低电平或高电平触发。 |
| [3] | **AUTOSS** | **自动片选功能使能(仅主模式)**<br>0 = 如果这一位清0，片选信号的产生或解除由SPI_SSR[0]位的值确定。<br>1 = 如果这一位置1，, SPI_SS 信号由硬件自动产生。SPI_SSR[0]所设定的片选状态会在发送或接收开始的时候自动产生，传输完成后自动解除。 |
| [2] | **SS_LVL** | **片选有效状态**<br>这一位设定片选信号(SPI_SS)的有效状态.<br>0 = 片选信号的有效状态是低电平或下降沿。<br>1 = 片选信号的有效状态是高电平和上升沿。 |
| [1] | | 保留. |
| [0] | **SSR** | **片选控制位(仅主机)**<br>如果AUTOSS 位清0<br>0 = SPIx_SS 输出无效状态.<br>1 = SPIx_SS 片选使能.<br>如果AUTOSS 位置1 |

| | | |
|---|---|---|
| | | 0 = 保持SPIx_SS管脚无效状态.<br>1 =让SPIx_SS 管脚在传输时自动变为有效状态，在其他时间变为无效。有效地状态由SS_LVL寄存器决定 |

## SPI 接收数据寄存器 (SPI_RX)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **SPI_RX** | SPIx_BA+0x10 | R | 接收数据寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **RX** | **接收数据寄存器**<br>接收数据寄存器存放从SPI输入管脚接收到的数据。如果不使能FIFO模式，程序可以从这里读到最新的输入数据。如果FIFO 位时1，并且RX_EMPTY b位, SPI_CNTRL[24] 或 SPI_STATUS[24],不是1, 程序可以从这个寄存器读到接收FIFO的数据。这个寄存器只读。 |

## SPI 发送数据寄存器 (SPI_TX)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **SPI_TX** | SPIx_BA+0x20 | W | 发送数据寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **TX** | **发送数据寄存器**<br>T发送数据寄存器存入下一个等待发送的数据。有效位的个数由SPI_CNTRL 寄存器TX_BIT_LEN 决定.<br>例如, 如果TX_BIT_LEN 设为 0x08, TX[7:0]会作为下次发送的数据。如果 TX_BIT_LEN 设为 0x00, 整个32位作为下次发送的数据.<br>**注意: 如果SPI控制器工作再从模式，并且不使用FIFO模式。如果SPI控制器要发送数据给主设备，发送数据寄存器需要在**GO_BUSY 设为1之前设置。 |

## SPI 控制和状态寄存器 2 (SPI_CNTRL2)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **SPI_CNTRL2** | SPIx_BA+0x3C | R/W | 控制和状态寄存器 2 | 0x0000_1000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | **SS_INT_OPT** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | **SLV_START_INTSTS** | **SSTA_INTEN** | **SLV_ABORT** | **NOSLVSEL** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | | | |

| 位 | 描述 | |
|---|---|---|
| [31:17] | | 保留 |
| [16] | **SS_INT_OPT** | **从机片选无效中断选择**<br>这一位仅当SPI控制器配置为从机片选电平触发时有效。<br>0 = 当片选信号变为无效状态， IF 位不会设为 1.<br>1 = 当片选信号变为无效状态， IF位会变为1. |
| [15:12] | | 保留. |
| [11] | **SLV_START_INTSTS** | **从机3线模式开始中断状态**<br>在3线从机模式下，这一位指示传输开始了。这一位是SPI_STATUS[11]的镜像<br>0 = 当 SSTA_INTEN 位设为1后，SPI从机没有检测到时钟输入.<br>1 = 从机3线模式开始一次传输。这一位在传输完成后自动清0，或者写1清0 |
| [10] | **SSTA_INTEN** | **从机3线模式开始中断使能位**<br>在3线从机模式开始传输时，这一位用来使能中断。如果传输开始后，在设定的时间内没有传输完成中断，用户可以设置SLV_ABORT 来强制传输结束.<br>0 = 禁止传输开始中断.<br>1 = 传输开始中断使能.这一位会在当前传输完成后清0，或者在SLV_START_INTSTS位清除后清0. |
| [9] | **SLV_ABORT** | **从机3线模式中止控制**<br>正常模式下，这一位指示接收到的数据个数满足 TX_BIT_LEN定义的值.<br>如果从机3线模式下，数据传输没达到设定的个数，又没有SPI时钟的输入，用户可以设定这一位中止目前的传输，然后会收到一个传输完成中断事件。<br>0 = 没影响.<br>1 =强制传输中止.<br>**Note:** 这一位写1后会自动清0.. |
| [8] | **NOSLVSEL** | **从机3线模式使能位** |

| | | |
|---|---|---|
| | | 从机3线模式下，SPI控制器只需要3根线 SPI_CLK, SPI_MISO, 和 SPI_MOSI.<br>0 = 4线双向从机模式.<br>1 = 3线双向从机模式. SPI控制器会在GO_BUSY 位设为 1后准备好发送或接收.<br>**注意:从机3线模式下，** SS_LTRIG (SPI_SSR[4]) 会自动置1. |
| [7:0] | | 保留 |

## SPI FIFO 控制寄存器 (SPI_FIFO_CTL)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **SPI_FIFO_CTL** | SPIx_BA+0x40 | R/W | SPI FIFO 控制寄存器 | 0x4400_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | **TX_THRESHOLD** | | | **Reserved** | **RX_THRESHOLD** | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | **TIMEOUT_INTEN** | **Reserved** | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | **RXOV_INTEN** | **Reserved** | | **TX_INTEN** | **RX_INTEN** | **TX_CLR** | **RX_CLR** |

| 位 | 描述 | |
|---|---|---|
| [31] | | 保留. |
| [30:28] | **TX_THRESHOLD** | **发送 FIFO 阈值**<br>如果发送FIFO里的数据数目小于等于TX_THRESHOLD 的设定, TX_INTSTS 位会置 1, 否则TX_INTSTS 位会清 0. |
| [27] | | 保留 |
| [26:24] | **RX_THRESHOLD** | **接收FIFO 阈值**<br>如果接收FIFO里数据多于RX_THRESHOLD的设置, RX_INTSTS 位会置 1, 否则 RX_INTSTS 位会清 0. |
| [23:22] | | 保留. |
| [21] | **TIMEOUT_INTEN** | **接收 FIFO 超时中断使能**<br>0 = 超时中断禁止.<br>1 = 超时中断使能. |
| [20:7] | | 保留 |
| [6] | **RXOV_INTEN** | **接收 FIFO 溢出中断使能位**<br>0 = 接收 FIFO 溢出中断禁止.<br>1 = 接收 FIFO 溢出中断使能. |
| [5:4] | | 保留. |
| [3] | **TX_INTEN** | **发送阈值中断使能位**<br>0 = 发送阈值中断禁止.<br>1 = 发送阈值中断使能. |
| [2] | **RX_INTEN** | **接收阈值中断使能位** |

| | | |
|---|---|---|
| | | 0 = 接收阈值中断禁止.<br>1 = 接收阈值中断使能. |
| [1] | **TX_CLR** | **清空发送 FIFO 缓存**<br>0 = 没影响.<br>1 = 清空发送FIFO缓存. TX_FULL flag 标志清 0， TX_EMPTY 标志置 1. 这一位写1后硬件自动清0. |
| [0] | **RX_CLR** | **清空接收 FIFO 缓存**<br>0 = 没影响.<br>1 =清空接收FIFO缓存. RX_FULL 标志清 0， RX_EMPTY标志置 1. 这一位写1后硬件自动清0. |

## SPI 状态寄存器 (SPI_STATUS)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **SPI_STATUS** | SPIx_BA+0x44 | R/W | SPI 状态寄存器 | 0x0500_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| TX_FIFO_COUNT | | | | TX_FULL | TX_EMPTY | RX_FULL | RX_EMPTY |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| Reserved | | | TIMEOUT | Reserved | | | IF |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| RX_FIFO_COUNT | | | | SLV_START_INTSTS | Reserved | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| Reserved | | | TX_INTSTS | Reserved | RX_OVERRUN | Reserved | RX_INTSTS |

| 位 | 描述 | |
|---|---|---|
| [31:28] | **TX_FIFO_COUNT** | **发送 FIFO 数据个数(只读)**<br>这几位指示目前发送FIFO里的数据个数 |
| [27] | **TX_FULL** | **发送FIFO缓存满指示 (只读)**<br>SPI_CNTRL[27]的镜像.<br>0 = 发送FIFO缓存不满.<br>1 = 发送FIFO缓存满. |
| [26] | **TX_EMPTY** | **发送FIFO缓存空指示 (只读)**<br>SPI_CNTRL[26]的镜像.<br>0 = 发送FIFO缓存非空.<br>1 = 发送FIFO缓存空. |
| [25] | **RX_FULL** | **接收FIFO缓存满指示 (只读)**<br>SPI_CNTRL[25] 的镜像.<br>0 = 接收FIFO缓存非满<br>1 = 接收FIFO缓存满. |
| [24] | **RX_EMPTY** | **接收FIFO缓存空指示(只读)**<br>SPI_CNTRL[24]的镜像.<br>0 = 接收FIFO缓存非空.<br>1 = 接收FIFO缓存空. |
| [23:21] | | 保留. |
| [20] | **TIMEOUT** | **超时中断标志**<br>0 = 没有FIFO超时事件.<br>1 = 接收FIFO非空，在64个SPI时钟（主模式）或576个SPI外设时钟（从模式）内没有读接 |

| | | |
|---|---|---|
| | | 收FIFO。如果程序开始读接收FIFO，这一位自动清除.<br>**注意:** 此位写1清0. |
| [19:17] | | 保留. |
| [16] | **IF** | **SPI 单元传输中断标志**<br>SPI_CNTRL[16]的镜像.<br>0 = 当这位清0后没有传输完成<br>1 = SPI控制器完成了一个单元的传输<br>**注意:** 此位写1清0. |
| [15:12] | **RX_FIFO_COUNT** | **接收 FIFO 数据个数 (只读)**<br>T这一位指示目前接收 FIFO 缓存中的数据个数. |
| [11] | **SLV_START_INTSTS** | **从机片选中断状态**<br>这一位用来指示从机3线模式下传输开始I，是 SPI_CNTRL2[11]的镜像.<br>0 = SSTA_INTEN 设为1后没有检测到SPI时钟输入，传输不会开始.<br>1 = 从机3线模式传输开始。传输完成后自动清0，或者写1清0 |
| [10:5] | | 保留. |
| [4] | **TX_INTSTS** | **发送 FIFO 阈值中断标志 (只读)**<br>0 = 发送FIFO里的有效数据多于 TX_THRESHOLD的设定.<br>1 = 发送FIFO里的有效数据少于TX_THRESHOLD的设定. |
| [3] | | 保留 |
| [2] | **RX_OVERRUN** | **接收 FIFO 溢出标志**<br>当接收FIFO已经满了，接下来的数据会被丢弃，这一位会置1<br>0 = 接收FIFO没有溢出<br>1 = 接收FIFO溢出了.<br>**Note:** 此位写1清0. |
| [1] | | 保留. |
| [0] | **RX_INTSTS** | **接收 FIFO 阈值中断标志 (只读)**<br>0 = 接收FIFO里的有效数据少于RX_THRESHOLD的设定.<br>1 = T接收FIFO里的有效数据多于RX_THRESHOLD的设定 |

## 6.15 硬件除法器 (HDIV)

### 6.15.1 概述

.硬件除法器拥有高性能的表现.硬件触发器是一种有符号的整数数据类型除法器，输出商和余数。

### 6.15.2 特性

- 支持有符号(二进制补码的) 整数数据类型计算。
- 支持32位被除数与16位除数的计算能力。
- 支持32位商和16位余数输出。
- 支持除数0警告指示。
- 7 HCLK时钟为一个计算周期。
- 有完成标志位的软件触发模式。

### 6.15.3 功能描述

使用硬件除法器,需要先设置被除数，然后设置除数，当除数被写入时，除法器将自动触发计算。通过读取DIVQUO和DIVREM寄存器，得到商和余数。假如硬件除法器还没有计算完成的情况下，CPU要读取DIVQUO或DIVREM寄存器,CPU将挂起直到硬件除法器完成计算。因此,即使在没有软件延迟的情况下CPU也可以得到一个有效的结果。

DIV0 (DIVSTS[0]) 如果除数是0 ，DIV0 (DIVSTS[0])标志会置位

被除数是32位带符号的整数类型,除数是16位带符号整数类型，商是32位带符号整数类型，余数是16位带符号整数类型。

如图 6-124 所示，计算X / Y,CPU需要把X写入DIVIDEND寄存器,然后把Y写入DIVISOR寄存器。随后通过读DIVQUO和DIVREM寄存器后得到计算结果。

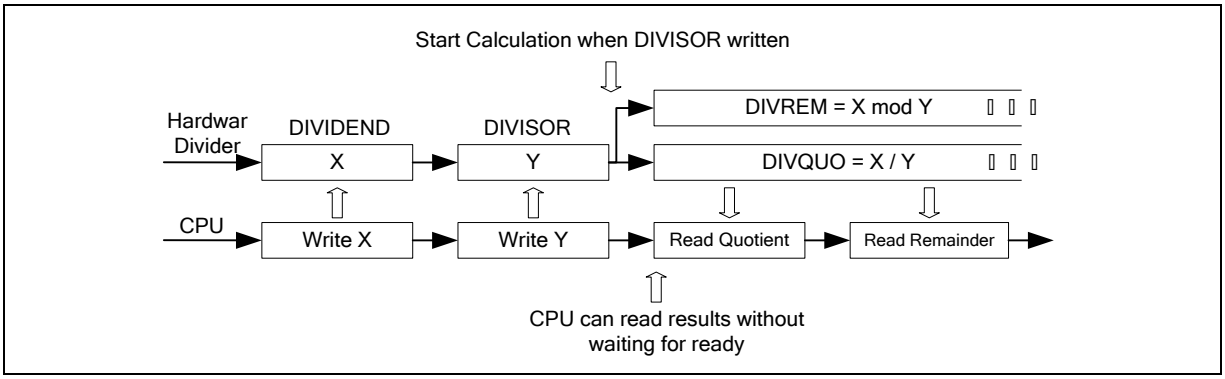



图 6-124 硬件除法器操作流程

### 6.15.4 寄存器列表

**R**: 只读, **W**: 只写, **R/W**: 可读写

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **HDIV 基地址:**<br>**HDIV_BA = 0x5001_4000** | | | | |
| **DIVIDEND** | HDIV_BA+0x04 | R/W | 被除数寄存器 (32位有符号整数类型) | 0x0000_0000 |
| **DIVISOR** | HDIV_BA+0x08 | R/W | 除数寄存器 (32位有符号整数类型) | 0x0000_FFFF |
| **DIVQUO** | HDIV_BA+0x0C | R | 商寄存器 (16位有符号整数类型) | 0x0000_0000 |
| **DIVREM** | HDIV_BA+0x10 | R | 余数寄存器 (16位有符号整数类型) | 0x0000_0000 |
| **DIVSTS** | HDIV_BA+0x14 | R/W | 除法器状态寄存器 | 0x0000_0001 |

### 6.15.5 寄存器描述

被除数寄存器(DIVIDEND)

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **DIVIDEND** | HDIV_BA+0x04 | R/W | )被除数寄存器（32位有符号整数类型） | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **Dividend** | **被除数**<br>被除数（32位有符号整数类型）.需要在除法器计算之前就写入。 |

## 除数寄存器 (DIVISOR)

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **DIVISOR** | HDIV_BA+0x08 | R/W | 除数寄存器（16位有符号整数类型） | 0x0000_FFFF |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留 |
| [15:0] | **Divisor** | 除数寄存器<br>**注意:**只有当此寄存器有写入时，硬件除法器才开始工作。 |

## 商寄存器 (DIVQUO)

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **DIVQUO** | HDIV_BA+0x0C | R | 商寄存器（32位有符号整数类型） | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:0] | **Quotient** | **商寄存器**<br>计算完成保留商的结果，（32位有符号整数类型），等待CPU读取。 |

## 余数寄存器 (DIVREM)

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **DIVREM** | HDIV_BA+0x10 | R | 余数寄存器（16位有符号整数类型） | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留. |
| [15:0] | **Reminder** | **余数**<br>算完成保留余数的结果，（16位有符号整数类型），等待CPU读取。 |

## 除法器状态寄存器(DIVSTS)

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **DIVSTS** | HDIV_BA+0x14 | R/W | 除法器状态寄存器 | 0x0000_0001 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | | **DIVFF** | **DIV0** | **DIV_FINISH** |

| 位 | 描述 | |
|---|---|---|
| [31:3] | **Reserved** | 保留 |
| [2] | **DIVFF** | **除法器操作完成标志位**<br>当计算完成时，硬件自动置1，通过软件写1清0 |
| [1] | **DIV0** | **除0警告指示（只读）**<br>0 = 除数非0<br>1 = 除数为0 |
| [0] | **DIV_FINISH** | **除法器 操作完成指示（只读）**<br>0=除法器计算结果没有完成<br>1=除法器计算结果完成<br>此位只读. |

## 6.16 增强型模数转换器 (EADC)

### 6.16.1 概述

NuMicro® M0519系列包含2个 12位逐次逼近式的模数转换器 (SAR A/D 转换器)，多达16个输入通道 。两个模数转换器ADCA 和 ADCB可以分开独立采样，也可以同步采样。 模数转换可以通过软件触发，PWM触发，定时器0-3事件触发，ADINT0, ADINT的EOC中断以及外部 STADC管脚触发。

**注意:模拟输入管脚必须在使能EADC功能前设置为输入模式**

### 6.16.2 特性

- 模拟电压输入范围: 0~$V_{REF}$(最大 5.0V).
- 12-bit 解析度 10-bit精度.
- 多达16路单端输入通道.
- 两路逐次逼近式ADC转换器.
- 4个EADC中断，各有独立的中断向量.
- 最大EADC工作频率: 16MHz.
- 采样率最大1.6M SPS, 每次ADC转换时间少于1.25μs.
- 两种工作模式
  - 单独采样模式: 2个ADC通道独立工作.
  - 同步采样模式: 让两个ADC转换器同时采样
- 一次模数转换可以这样开始:
  - 程序写1到 ADST(ADSSTR[n]) 位( n = 0~15)
  - 外部输入管脚 STADC
  - 定时器0~3 溢出脉冲触发
  - ADINT0, ADINT1 EOC中断信号触发
  - PWM触发
- 转换的结果放在16个数据寄存器里，并且有覆盖标志.
- SAMPLEA0~7 ADC 控制逻辑模块, 可以配置给ADCA转换通道 AINA0~7 和触发源.
- SAMPLEB0~7 ADC 控制逻辑模块，可以配置给 ADCB转换通道 AINB0~7 和触发源.
- 通道 AINA0 支持两个输入源: 外部模拟电压输入和内部 OP0放大器输出电压.
- 通道 AINB0 支持2个输入源: 外部模拟电压输入或内部 OP1 放大器输出电压.
- 通道 AINA7 支持4个输入源: 外部模拟电压输入, 内部固定带隙电压, 内部温度传感器输出和模拟地.

### 6.16.3 框图

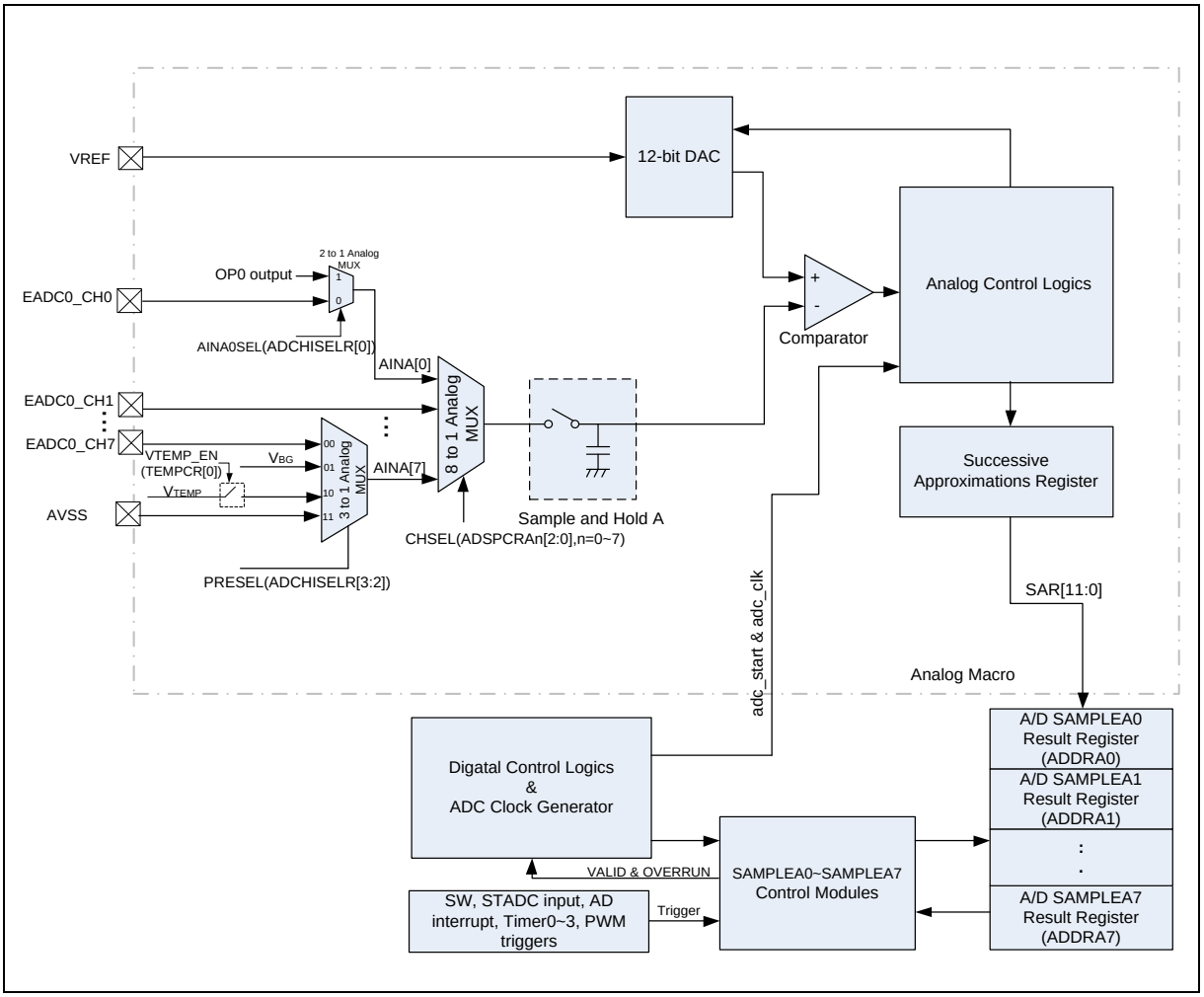



图 6-125 ADCA 转换器框图

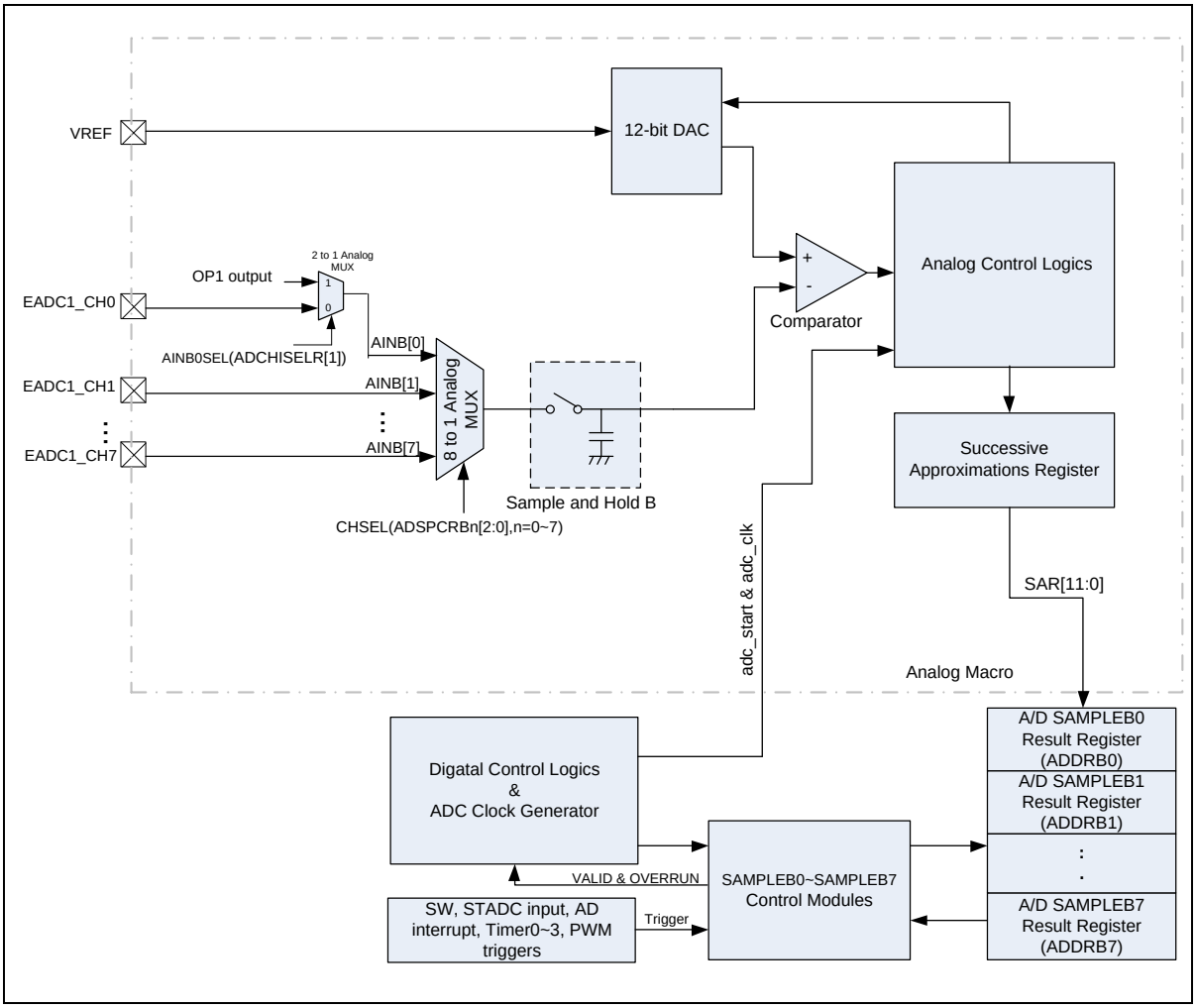



图 6-126 ADCB 转换器框图

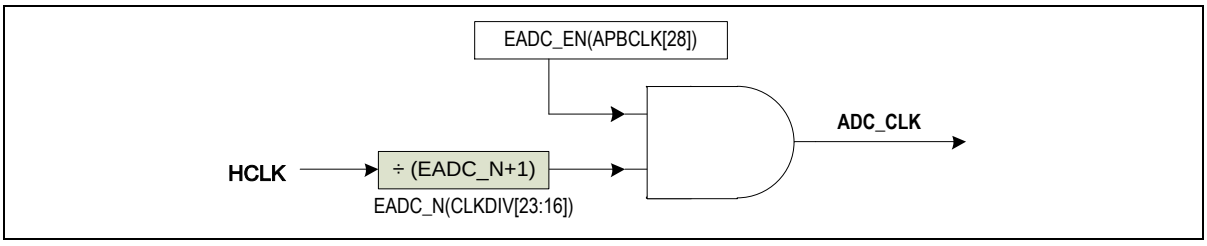



图 6-127 ADC 时钟控制

### 6.16.4 操作流程

共2组ADC 转换器, 每一组有8个选择输入通道以及8个采样控制模组。AD转换器是逐次逼近式的，12位的解析度。

AD转换的操作基于采样模块SAMPLEA0~7(ADCA 转换器) 和 SAMPLEB0~7(ADCB 转换器) 控制逻辑模块。每一个模块都可以单独配置触发源和输入通道。不同的采样模块可以配置到同一个输入通道，相同的触发源。 这样可以很方便的得到过采样的结果。

ADC 转换触发源有如下几种:

- 程序写1到 ADST(ADSSTR[n]) 位 ( n = 0~15)
- 外部管脚 STADC
- 定时器0~3 溢出信号触发
- ADINT0, ADINT1 ADC EOC 中断信号触发
- PWM 触发

ADINT0, ADINT1 中断信号当任意一个采样模块AD转换完成信号（ EOC）产生后生成。 ADINT0, 1 中断可以触发另一次AD转换, 可以方便的实现连续转换.

*6.16.4.1 ADC 时钟生成*

最大的采样率是800 kHz，转换时间少于1.25μs.一次AD装好需要 20个 EADC 时钟. EADC 引擎的时钟源是HCLK, EADC时钟经过一个8位的预分频然后通过下面的公式计算:

EADC 时钟频率 = (HCLK) / (EADC_N+1).

8位设为 EADC_N 在寄存器 CLKDIV[23:16]里.

总的来说，程序可以设定 EADC_N 得到16 MHz 或稍低些的频率.

*6.16.4.2 ADC 单次采样模式*

当ADC转换器通过采样模块SAMPLEx 设定为单词采样模式, 操作流程如下:

- 程序写1到 ADST(ADSSTR[n]) 位或其他触发方式触发ADC转换.
- 当转换完成以后，12位的结果会存入EADC 对应采样通道SAMPLEx的数据寄存器ADDRx 里.
- 转换完成之后，ADSR的 ADFn位会置1 ，如果ADIEn 设为1那么EADC中断 (ADINTn)会产生
- ADST(ADSSTR[n]) 在AD转换期间保持为1.当转换完成ADST(ADSSTR[n]) 位会自动清0，AD转换器可以开始下一个挂起的任务。

**注意: 如果程序和其他触发源使能了多个单次或同步转换，优先级高的采样模块会首先开始转换，其他的会挂起。**

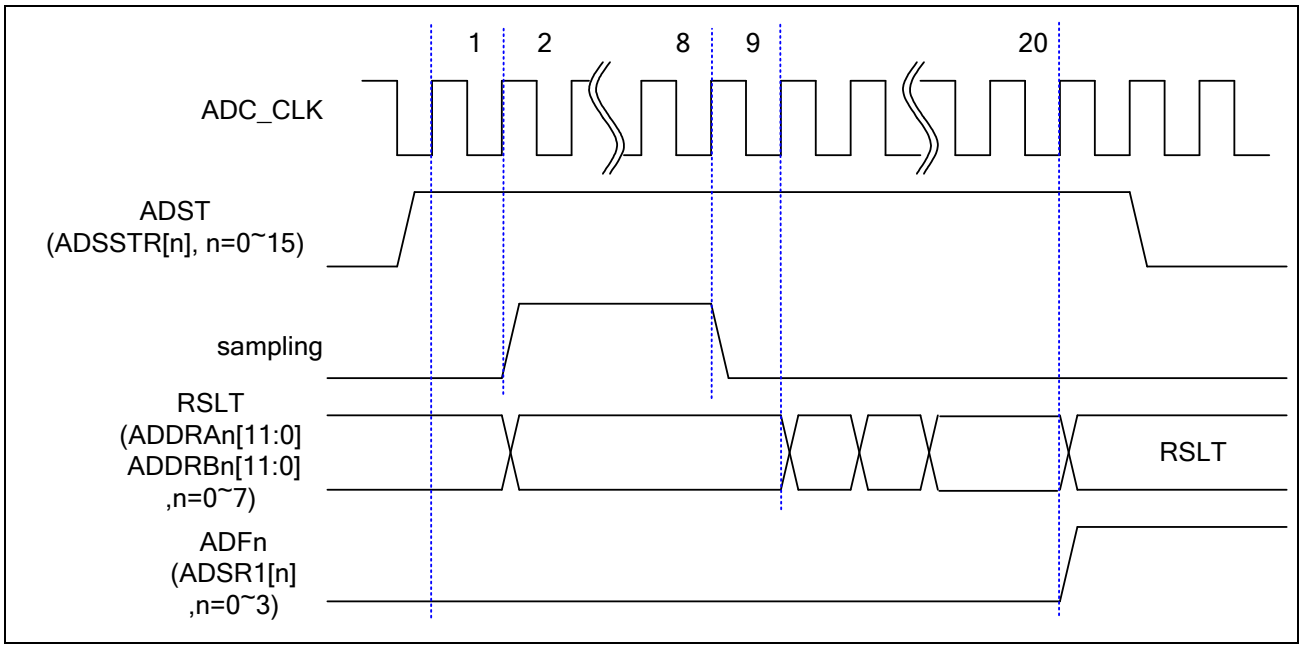



图 6-128 单次采样模式时序图

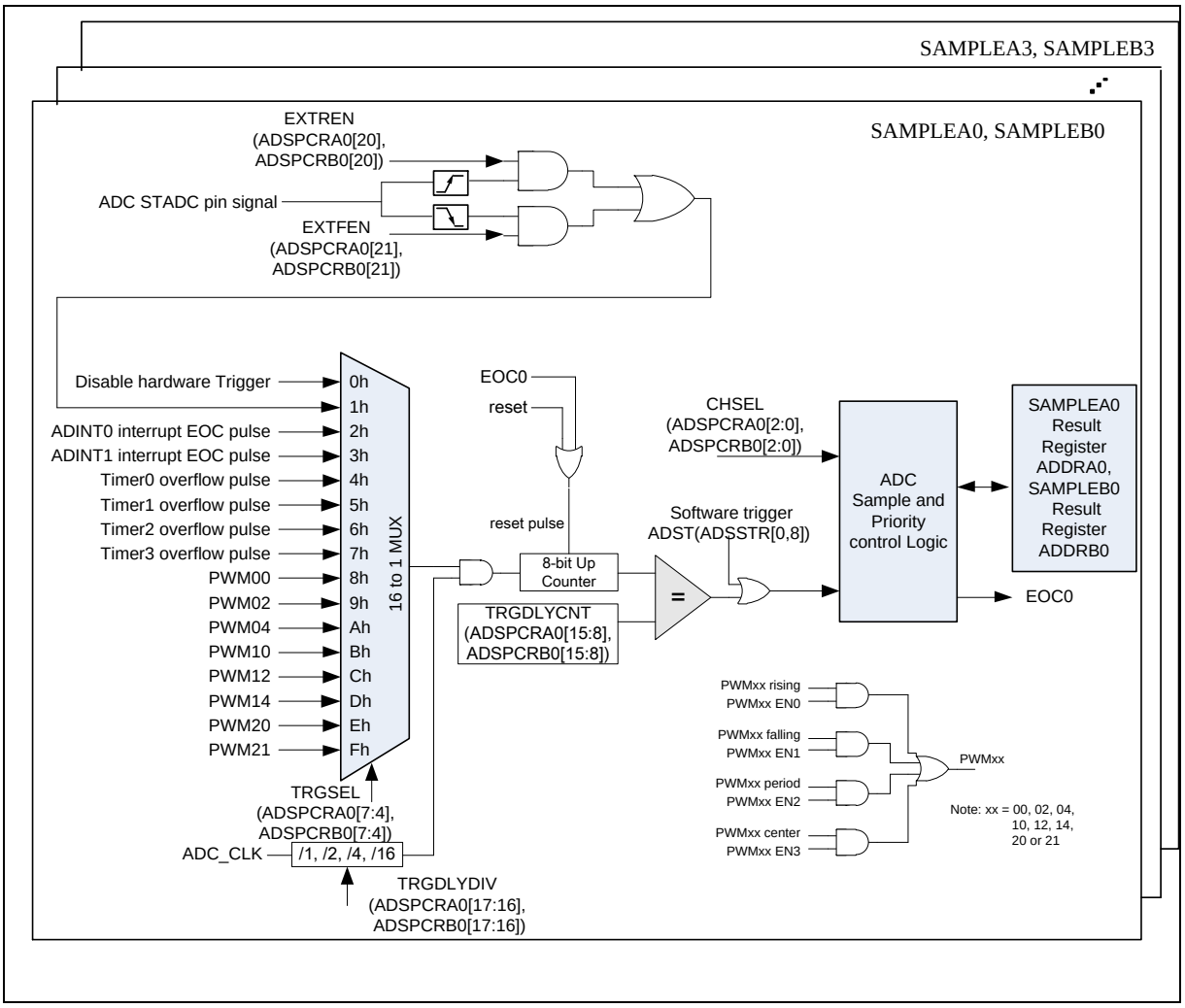



图 6-129 SAMPLEA0~3 和 SAMPLEB0~3 控制框图

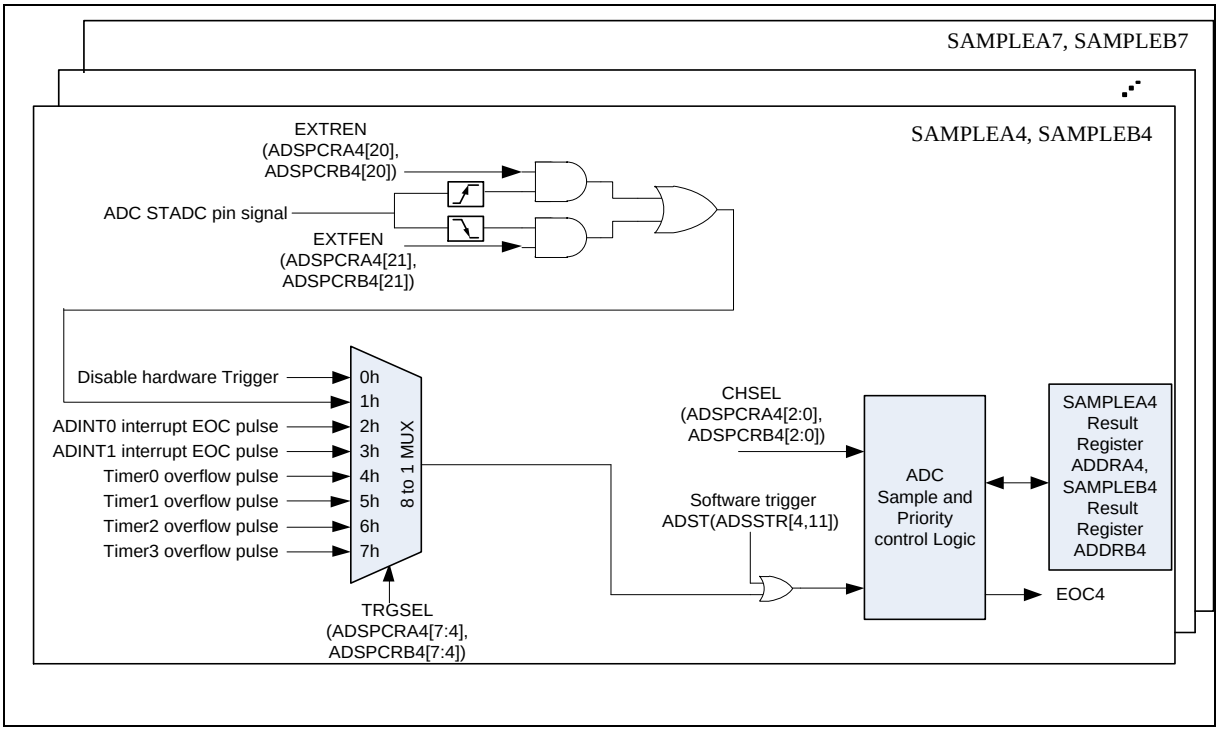



图 6-130 SAMPLEA4~7 和SAMPLEB4~7控制框图

### *6.16.4.3 ADC 同步采样模式*

NuMicro® M0519有两组ADC转换器，允许2个不同的ADC通道同时采样，优先级的规则与单次采样模式相同.

SAMPLEA 和SAMPLEB 相同编号的模块组合在一起，比如SAMPLEA0 和 SAMPLEB0 当同步采样使能位SIMUSEL0(ADSMSELR[0]设为1后组合在一起。

ADC 同步采样模式操作流程如下:

1. 只有 SAMPLEA 来触发一组转换.
2. SAMPLEA 和 SAMPLEB 各一组通道使能同步采样功能， SAMPLEB 的相关触发设置被忽略.
3. A模块的转换结果存入SAMPLEA ADDRA寄存器, B模块的装换结果存入同样编号的SAMPLEB ADDRB 寄存器.

例如:

如 果 SIMUSEL2(ADSMSELR[2]) 位 = 1 并 且 SAMPLEA2 配 置 采 样 EADC0_CH4 (CHSEL(ADSPCRA2[2:0]) = 4), 定义 SAMPLEA2 和同样编号的 SAMPLEB2 组合在一起工作在同步采样模式, 如果 SAMPLEB2 配置为采样 EADC1_CH3 (CHSEL(ADSPCRB2[2:0]) = 3), 这一组采样通道是 EADC0_CH4 和 EADC1_CH3.

当通道对(AINA[4], AINB[3])AD转换完成, 装换的结果相应存入寄存器 ADDRA2 和 ADDRB2.

### *6.16.4.4 ADC 转换优先级*

有2组优先级设定来决定多个采样模组同时触发时的优先顺序，SAMPLEA0~7 是 ADCA 转换器的优先级组, SAMPLEB0~7 ADCB转换器的优先级组.

编号低的采样模块拥有更高的优先级。如果2个采样模组同时触发，编号低的采样模组会首先进行转换。

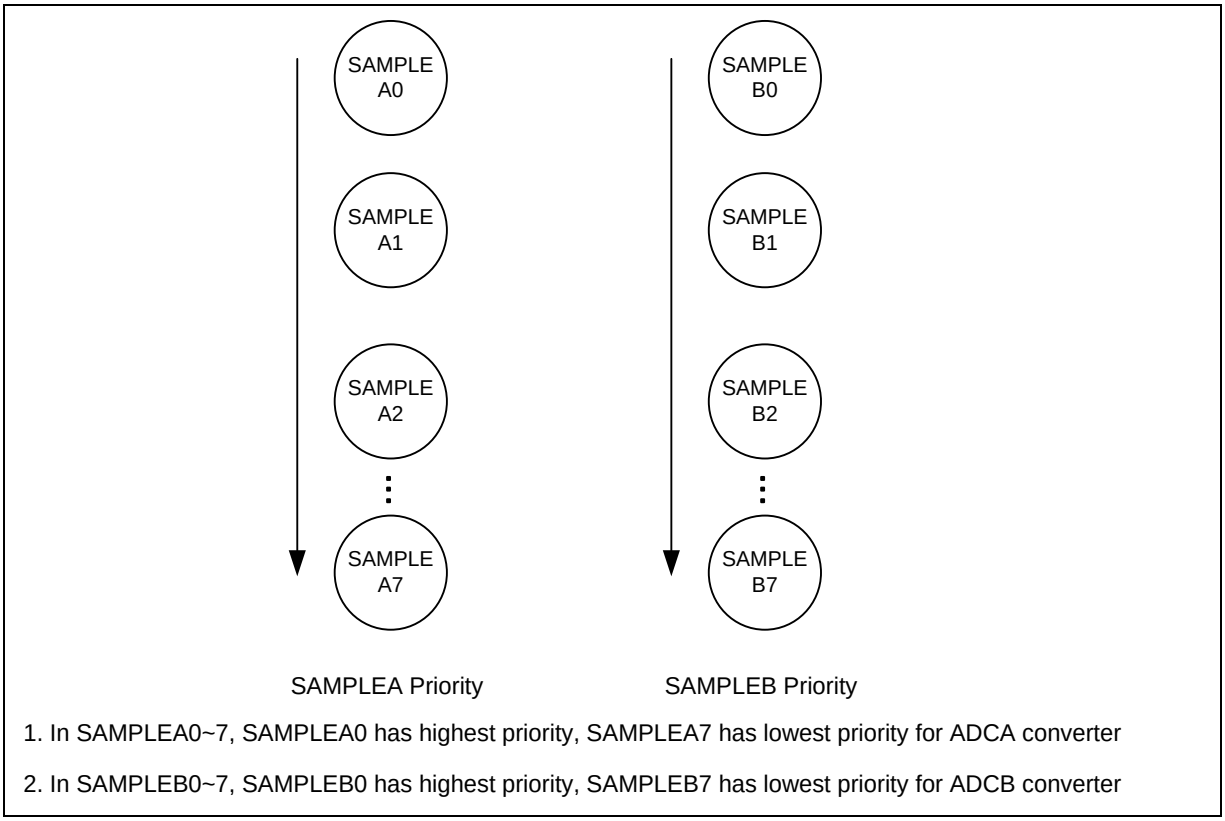



图 6-131 采样模组转换优先级仲裁算法

#### *6.16.4.5 PWM触发ADC转换*

ADC转换除了可以通过写寄存器， ADINT0, 1 中断信号, 外部管脚 STADC, 定时器0~3 溢出信号外,还增加了新的特性：允许PWM通道触发ADC转换。 用户可以设置如下几种PWM触发类型：PWM的上升沿，下降沿，PWM信号周期的中点（仅中心对齐模式）。设备也允许在收到PWM触发信号后延迟一些时间来触发ADC转换。延迟时间在ADSPCRAn 寄存器的TRGDLYCNT 和TRGDLYDIV 位和ADSPCRBn(n=0~3)寄存器中设定. 图 6-132 说明从PWM触发信号到ADC转换开始的延时时间.

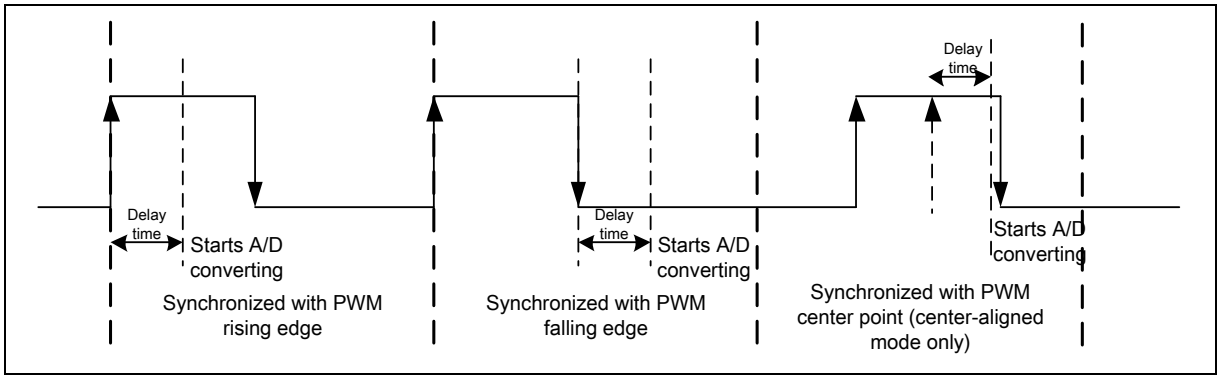



图 6-132 PWM触发ADC转换

#### 6.16.4.6 ADC 采样模块转换完成中断操作

ADC有4个中断 ADINT0~3, 每个终端有独立的中断向量地址。每个中断可以选择任意的采样终端的转换完成信号 (SAMPLEA0~7, SAMPLEB0~7 转换完成信号) 做自己的中断源. 每个中断有自己的中断改写标志ADFOVn(ADIFOVR[n], n=0~3). 当相应的中断被改写为1，这个标志置1.

当ADIE0(ADCR[2]) = 1 且 ADINT0SRCTL[7:0]=0xFF, 所有的A组的采样模块SAMPLEA转换完成信号可以触发ADINT0 中断.

当 ADIE1(ADCR[3]) = 1 且 ADINT1SRCTL[15:8]=0xFF, 所有B组的采样模块 SAMPLEB 转换完成信号可以触发 ADINT1中断.

ADINT0, ADINT1 中断可以有任意一个采样模块的ADC转换完成信号产生。这两个中断也可以作为采样模块的触发源，这样可以设置ADC连续转换模式。

连续转换的例子:

第一步. 如果ADC采样模块A2的转换完成信号EOCA2作为 ADINT0 的中断源(ADINT0SRCTL[7:0] = 0x04)，并且ADINT0 作为采样模块A0, B1, A2的触发源.

第二步. 软件触发 ADST(ADSSTR[2]) 位为1开始采样模块A2 的ADC转换, 转换完成以后，会产生转换完成信号EOCA2 和 ADINT0 中断信号。 ADINT0 中断信号会触发 采样模块A0, B1, A2 开始ADC 转换.

第三步. ADINT0 中断信号会自动重复触发采样模块A0, B1, A2 ADC 转换.

第四步. 如果需要结束转换，TRGSEL(ADSPCRA2[7:4]) 清 0禁止 采样模块A2 的ADINT0 中断信号.

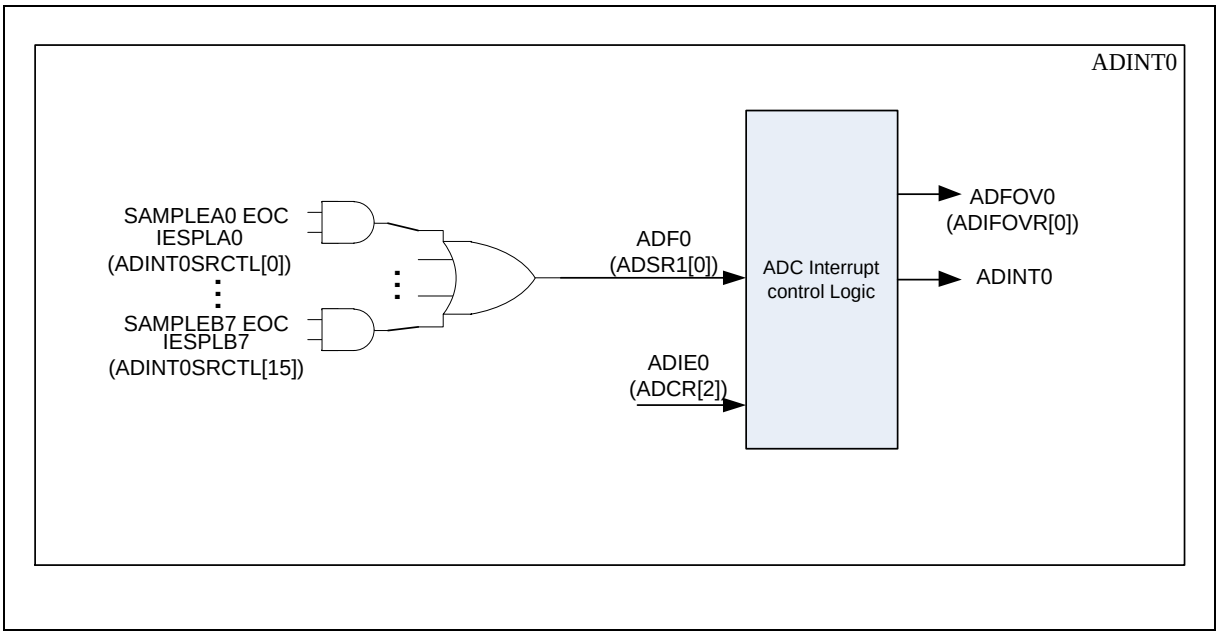



图 6-133 选择特定的采样模块的转换完成信号产生ADINT0中断

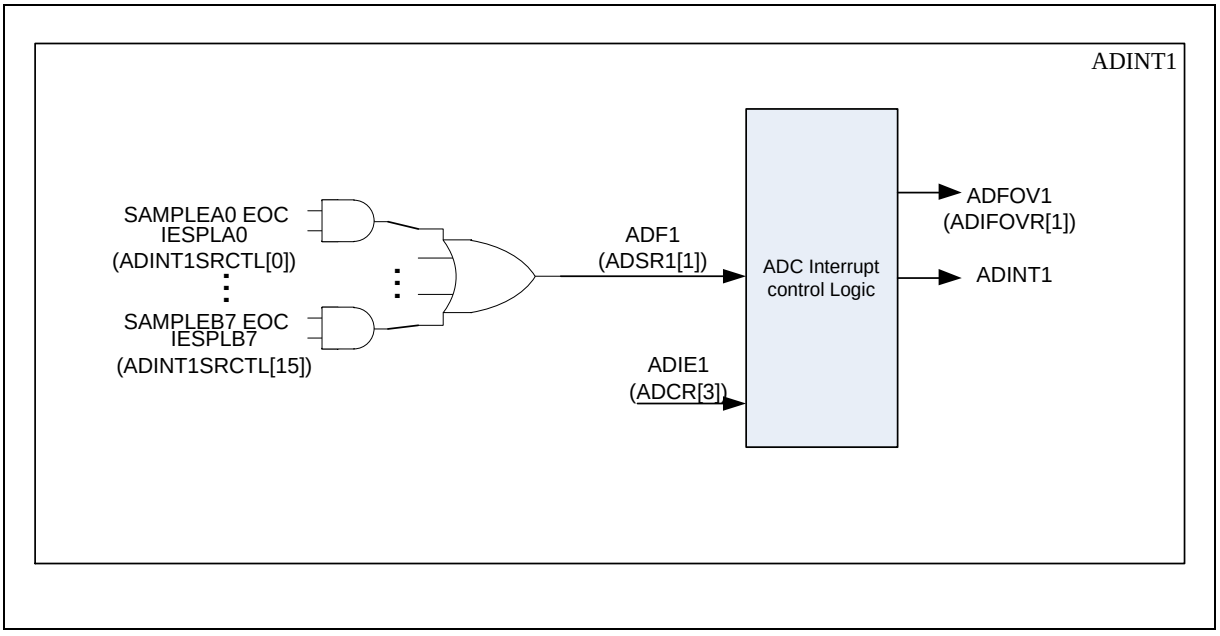



图 6-134 选择特定的采样模块的转换完成信号产生ADINT1 中断

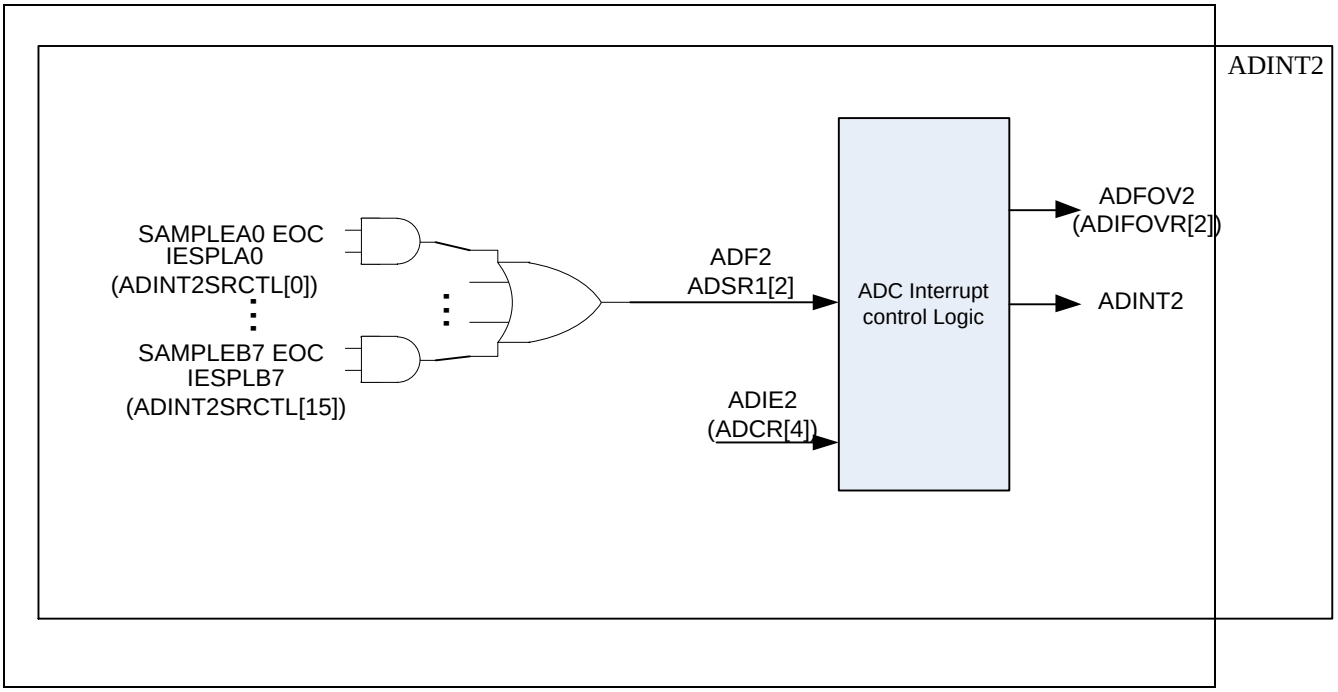



图 6-135 选择特定的采样模块的转换完成信号产生ADINT2 中断

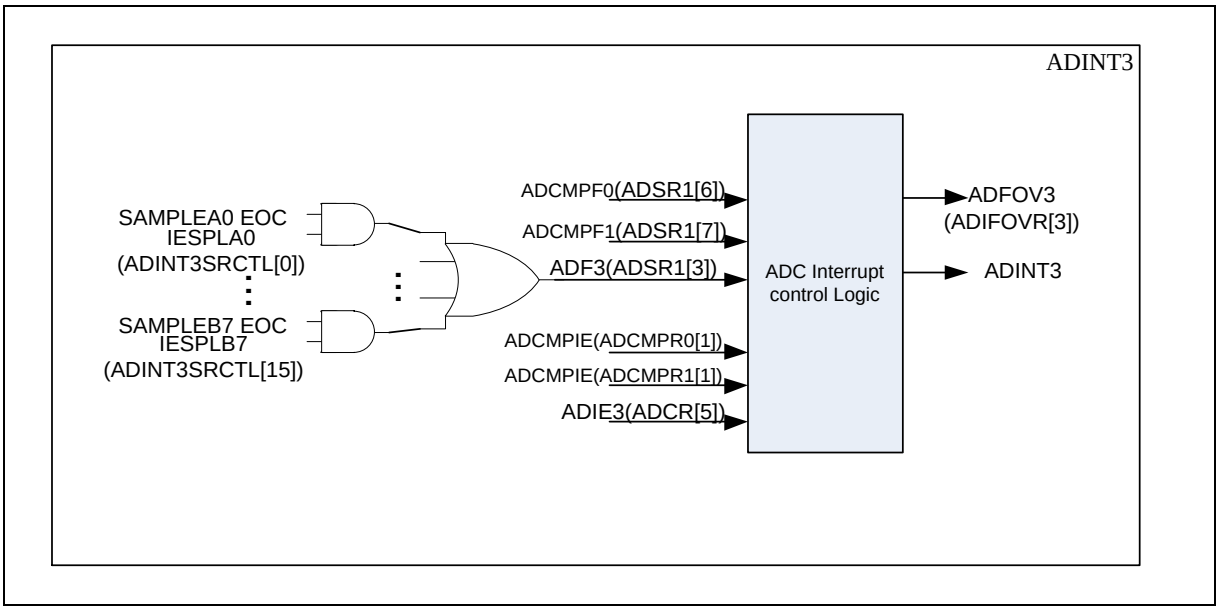



图 6-136 选择特定的采样模块的转换完成信号产生ADINT3 中断

### *6.16.4.7 输入采样 A/D 转换时间*

A/D 转换器当ADSSTR设为1之后等待ADC转换开始延迟时间(Td)后采样模拟信号输入。因为ADC时钟由HCLK除以(EADC_N(CLKDIV[23:16])+1)得到的，从写ADST到ADC开始采样的延时时间最多2个ADC时钟周期. 开始延时如图 图 6-137.

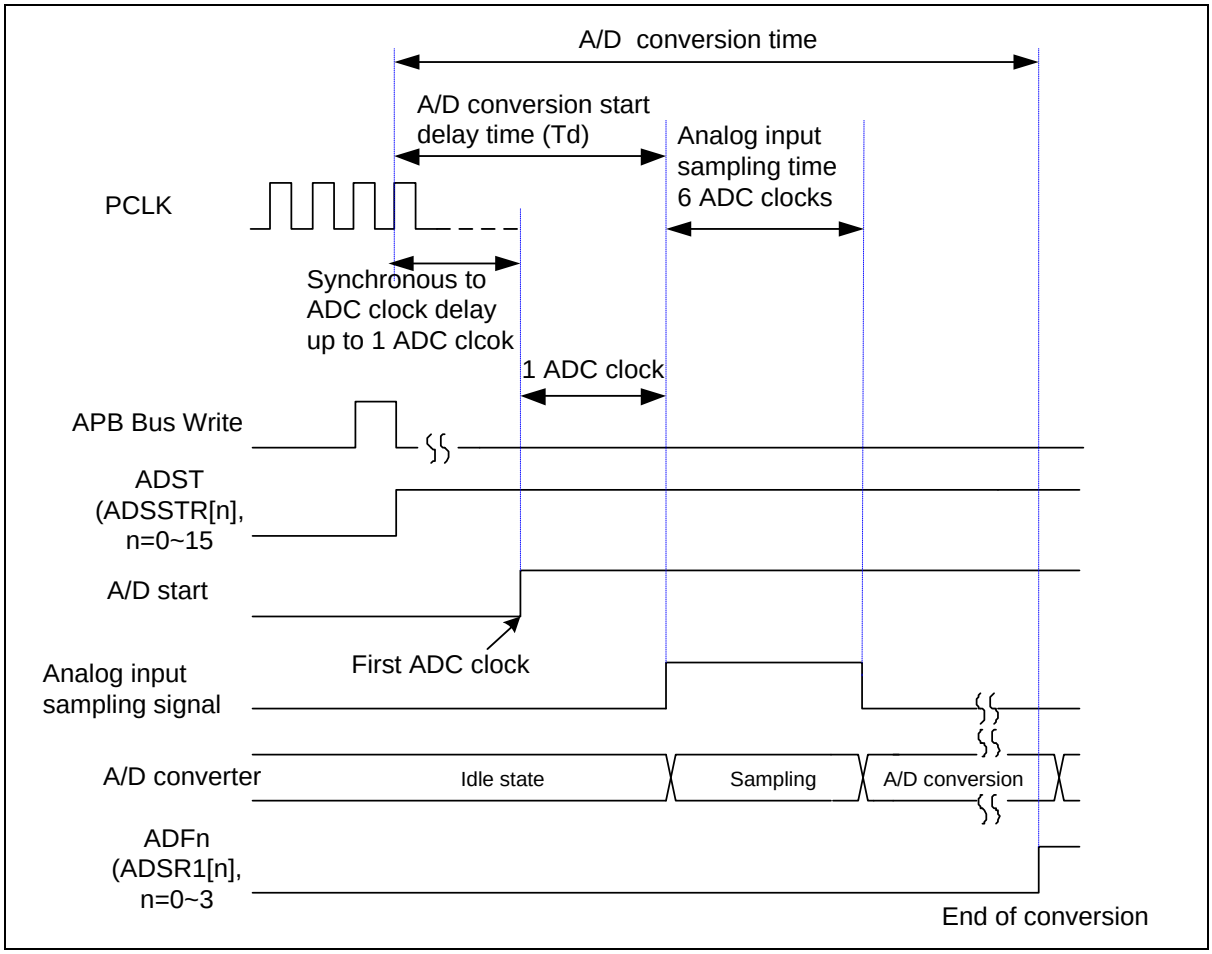



图 6-137 转换开始延迟时间

A/D转换可以通过外部管脚 STADC 触发. 设置 TRGSEL(ADSPCRAm[7:4], ADSPCRBm[7:4], ADSPCRAn[6:4], ADSPCRBn[6:4], m=0~3, n=4~7) 位为1选择从外部管脚 STADC触发.为了抵抗尖峰脉冲设计了一个8位的采样计数器。如果选择了边沿触发，高电平和低电平的时间都至少在4个HCLK的时间以上，短于这个时间的信号会被忽略。

### *6.16.4.8 A/D 增加采样时间*

当ADC的时钟频率比较高，如果输入的模拟信号阻抗很大需要更多的时间充电，那么采样时间可能不够。用户可以通过写ADTCR 寄存器的ADAEST 或ADBEST区域来增加AD采样时间。 扩充的AD采样时间增加在ADC控制器决定哪一路转换到ADC开始转换之间。增加时间的范围从0 ~255 ADC 时钟. 增加采样时间如 图 6-138.

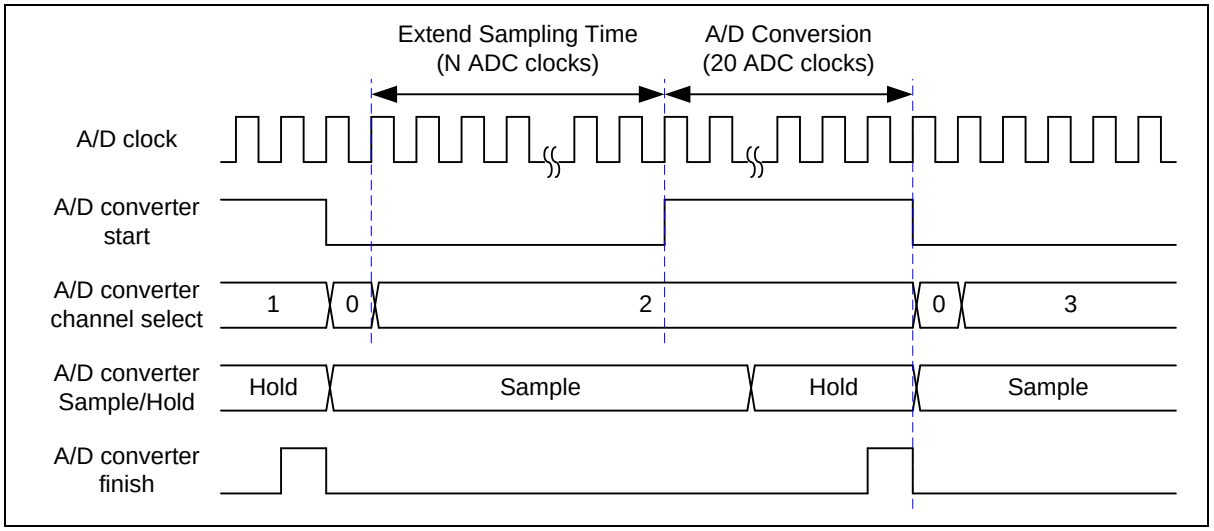



图 6-138 A/D 增加采样时钟

### *6.16.4.9 采样结果比较模式*

NuMicro® M0519 控制器提供了2个比较寄存器ADCMPR0/1 监控最多2组特定的采样模块SAMPLEA0~3, SAMPLEB0~3的转换结果, 如下图所示. 程序可以设定CMPSMPL 寄存器选择监控哪一个采样模块的结果，设定ADCMPR0/1 寄存器的CMPCOND 位选择比较条件，大于等于或小于CMPD(ADCMPRn[27:16], n=0,1)设定的特定值。当CMPSMPL 设定的采样模块转换结束了，比较动作会自动触发。如果满足比较条件，比较结果匹配寄存器会加一。当匹配寄存器的值达到(CMPMATCNT(ADCMPRn[11:8]+1n n=0, 1) 的设定， ADCMPFn(ADSR1[7:6], n=0, 1) 位会置1.如果ADCMPIE(ADCMPRn[1], n=0, 1) 位为1, ADINT3 中断就会产生。用户可以用这个功能来监控模拟电压输入的变化。具体的逻辑框图如 图 6-139.

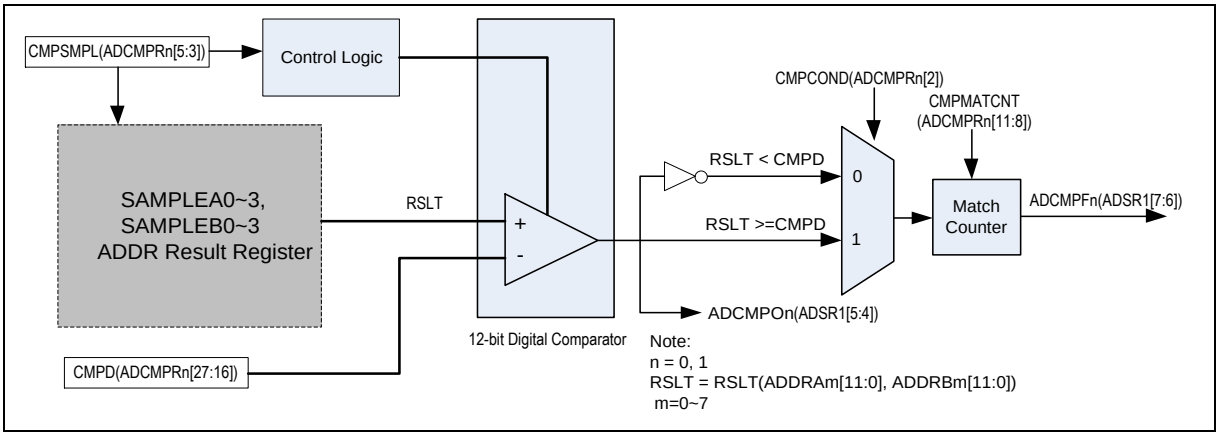



图 6-139 A/D 转换结果监控逻辑单元

### *6.16.4.10 中断源*

当选定的采样模块转换完成后， A/D 转换器触发ADSR1 寄存器的ADFn(n=0~3) 标志。如果ADCR寄存器的ADIEn(n=0~3)位使能, 会产生 ADINTn(n=0~3)中断.

如果ADCMPIE(ADCMPRn[1], n=0, 1) 位使能, 当AD转换的结果满足ADCMPRn(n=0, 1) 寄存器的设置, 监控中断会产生, ADINT3 会产生. 用户可以清楚ADCMPFn(ADSR1[7:6], n=0,1)和 ADFn(ADSR1[3:0], n=0~3)标志阻止中断产生.

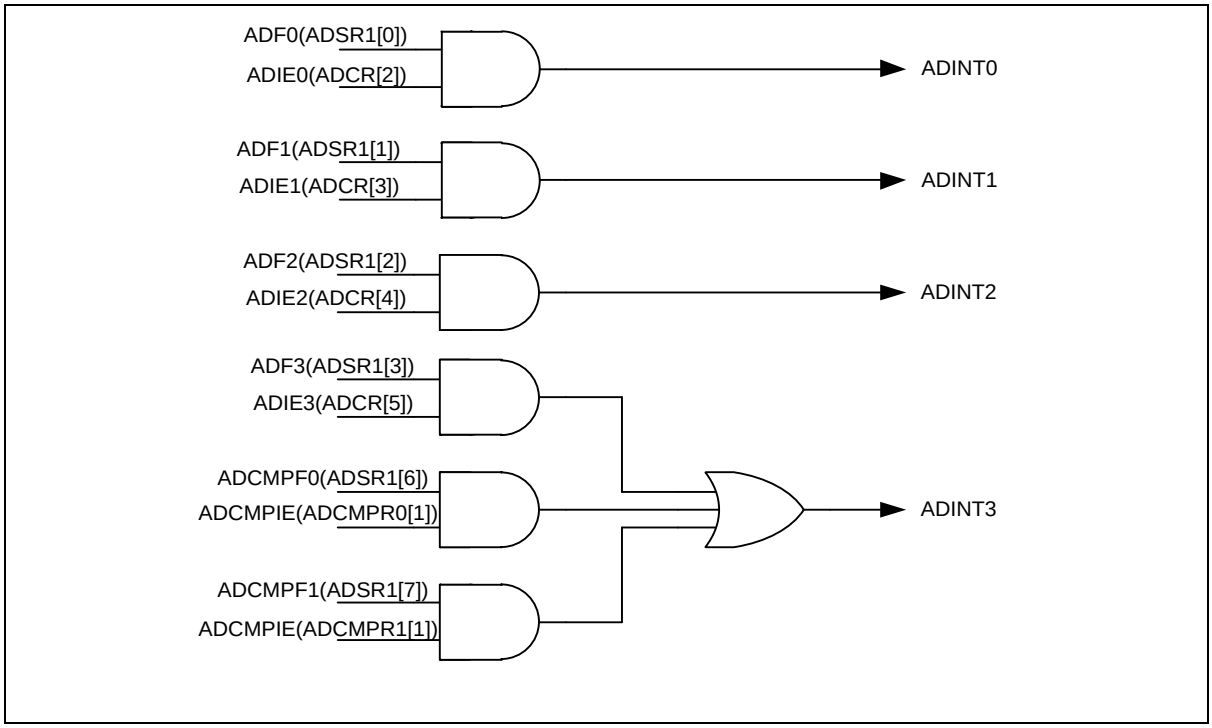



图 6-140 A/D 控制器中断

### 6.16.5 寄存器列表

**R**: 只读, **W**: 只写, **R/W**: 可读写, **C**: 只可写0

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **EADC 基地址:**<br>**EADC_BA = 0x400E_0000** | | | | |
| **ADDRA0** | EADC_BA+0x00 | R | 采样模组A0的AD数据寄存器0 | 0x0000_0000 |
| **ADDRA1** | EADC_BA+0x04 | R | 采样模组A1的AD数据寄存器1 | 0x0000_0000 |
| **ADDRA2** | EADC_BA+0x08 | R | 采样模组A2的AD数据寄存器2 | 0x0000_0000 |
| **ADDRA3** | EADC_BA+0x0C | R | 采样模组A3的AD数据寄存器3 | 0x0000_0000 |
| **ADDRA4** | EADC_BA+0x10 | R | 采样模组A4的AD数据寄存器4 | 0x0000_0000 |
| **ADDRA5** | EADC_BA+0x14 | R | 采样模组A5的AD数据寄存器5 | 0x0000_0000 |
| **ADDRA6** | EADC_BA+0x18 | R | 采样模组A6的AD数据寄存器6 | 0x0000_0000 |
| **ADDRA7** | EADC_BA+0x1C | R | 采样模组A7的AD数据寄存器7 | 0x0000_0000 |
| **ADDRB0** | EADC_BA+0x20 | R | 采样模组B0的AD数据寄存器8 | 0x0000_0000 |
| **ADDRB1** | EADC_BA+0x24 | R | 采样模组B1的AD数据寄存器9 | 0x0000_0000 |
| **ADDRB2** | EADC_BA+0x28 | R | 采样模组B2的AD数据寄存器10 | 0x0000_0000 |
| **ADDRB3** | EADC_BA+0x2C | R | 采样模组B3的AD数据寄存器11 | 0x0000_0000 |
| **ADDRB4** | EADC_BA+0x30 | R | 采样模组B4的AD数据寄存器12 | 0x0000_0000 |
| **ADDRB5** | EADC_BA+0x34 | R | 采样模组B5的AD数据寄存器13 | 0x0000_0000 |
| **ADDRB6** | EADC_BA+0x38 | R | 采样模组B6的AD数据寄存器14 | 0x0000_0000 |
| **ADDRB7** | EADC_BA+0x3C | R | 采样模组B7的AD数据寄存器15 | 0x0000_0000 |
| **ADCR** | EADC_BA+0x40 | R/W | A/D 控制寄存器 | 0x0000_0000 |
| **ADCHISELR** | EADC_BA+0x44 | R/W | A/D 通道源选择寄存器 | 0x0000_0000 |
| **ADSSTR** | EADC_BA+0x48 | W | A/D 采样模块软件触发寄存器 | 0x0000_0000 |
| **ADSTPFR** | EADC_BA+0x4C | R | A/D 采样模块开始转换挂起标志寄存器 | 0x0000_0000 |
| **ADIFOVR** | EADC_BA+0x50 | R/W | A/D ADINT3~0 I中断标志覆盖寄存器 | 0x0000_0000 |
| **ADSPOVFR** | EADC_BA+0x54 | R/W | A/D 采样模块开始转换标志覆盖寄存器 | 0x0000_0000 |
| **ADSPCRA0** | EADC_BA+0x58 | R/W | A/D 采样模块A0 控制寄存器 | 0x0000_0000 |
| **ADSPCRA1** | EADC_BA+0x5C | R/W | A/D 采样模块A1 控制寄存器 | 0x0000_0000 |
| **ADSPCRA2** | EADC_BA+0x60 | R/W | A/D 采样模块A2 控制寄存器 | 0x0000_0000 |
| **ADSPCRA3** | EADC_BA+0x64 | R/W | A/D 采样模块A3 控制寄存器 | 0x0000_0000 |

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **EADC 基地址:**<br>**EADC_BA = 0x400E_0000** | | | | |
| **ADSPCRA4** | EADC_BA+0x68 | R/W | A/D 采样模块A4 控制寄存器r | 0x0000_0000 |
| **ADSPCRA5** | EADC_BA+0x6C | R/W | A/D 采样模块A5 控制寄存器 | 0x0000_0000 |
| **ADSPCRA6** | EADC_BA+0x70 | R/W | A/D 采样模块A6 控制寄存器 | 0x0000_0000 |
| **ADSPCRA7** | EADC_BA+0x74 | R/W | A/D 采样模块A7 控制寄存器 | 0x0000_0000 |
| **ADSPCRB0** | EADC_BA+0x78 | R/W | A/D 采样模块B0 控制寄存器 | 0x0000_0000 |
| **ADSPCRB1** | EADC_BA+0x7C | R/W | A/D 采样模块B1 控制寄存器 | 0x0000_0000 |
| **ADSPCRB2** | EADC_BA+0x80 | R/W | A/D 采样模块B2 控制寄存器 | 0x0000_0000 |
| **ADSPCRB3** | EADC_BA+0x84 | R/W | A/D 采样模块B3 控制寄存器 | 0x0000_0000 |
| **ADSPCRB4** | EADC_BA+0x88 | R/W | A/D 采样模块B4 控制寄存器 | 0x0000_0000 |
| **ADSPCRB5** | EADC_BA+0x8C | R/W | A/D 采样模块B5 控制寄存器 | 0x0000_0000 |
| **ADSPCRB6** | EADC_BA+0x90 | R/W | A/D 采样模块B6 控制寄存器 | 0x0000_0000 |
| **ADSPCRB7** | EADC_BA+0x94 | R/W | A/D 采样模块B7 控制寄存器 | 0x0000_0000 |
| **ADSMSELR** | EADC_BA+0xA4 | R/W | A/D 采样模块同步模式选择寄存器 | 0x0000_0000 |
| **ADCMPR0** | EADC_BA+0xA8 | R/W | A/D结果比较寄存器 0 | 0x0000_0000 |
| **ADCMPR1** | EADC_BA+0xAC | R/W | A/D 结果比较寄存器 1 | 0x0000_0000 |
| **ADSR0** | EADC_BA+0xB0 | R | A/D 状态寄存器 0 | 0x0000_0000 |
| **ADSR1** | EADC_BA+0xB4 | R/W | A/D 状态寄存器 1 | 0x0000_0000 |
| **ADTCR** | EADC_BA+0xB8 | R/W | A/D 时序控制寄存器 | 0x0000_0000 |
| **ADDRDBA0** | EADC_BA+0x100 | R | 采样模块A0 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBA1** | EADC_BA+0x104 | R | 采样模块A1 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBA2** | EADC_BA+0x108 | R | 采样模块A2 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBA3** | EADC_BA+0x10C | R | 采样模块A3 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBB0** | EADC_BA+0x120 | R | 采样模块B0 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBB1** | EADC_BA+0x124 | R | 采样模块B1 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBB2** | EADC_BA+0x128 | R | 采样模块B2 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBB3** | EADC_BA+0x12C | R | 采样模块B3 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDBM** | EADC_BA+0x130 | R/W | A/D 双缓存模式选择 | 0x0000_0000 |
| **ADINT0SRCTL** | EADC_BA+0x134 | R/W | A/D中断0来源使能控制寄存器 | 0x0000_0000 |

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **EADC 基地址:**<br>**EADC_BA = 0x400E_0000** | | | | |
| **ADINT1SRCTL** | EADC_BA+0x138 | R/W | A/D中断1来源使能控制寄存器 | 0x0000_0000 |
| **ADINT2SRCTL** | EADC_BA+0x13C | R/W | A/D中断2来源使能控制寄存器 | 0x0000_0000 |
| **ADINT3SRCTL** | EADC_BA+0x140 | R/W | A/D中断3来源使能控制寄存器. | 0x0000_0000 |
| **SMPTRGA0** | EADC_BA+0x144 | R/W | 采样模块A0 A/D 触发状态 | 0x0000_0000 |
| **SMPTRGA1** | EADC_BA+0x148 | R/W | 采样模块A1 A/D 触发状态 | 0x0000_0000 |
| **SMPTRGA2** | EADC_BA+0x14C | R/W | 采样模块A2 A/D 触发状态 | 0x0000_0000 |
| **SMPTRGA3** | EADC_BA+0x150 | R/W | 采样模块A3 A/D 触发状态 | 0x0000_0000 |
| **SMPTRGB0** | EADC_BA+0x154 | R/W | 采样模块B0 A/D 触发状态 | 0x0000_0000 |
| **SMPTRGB1** | EADC_BA+0x158 | R/W | 采样模块B1 A/D 触发状态 | 0x0000_0000 |
| **SMPTRGB2** | EADC_BA+0x15C | R/W | 采样模块B2 A/D 触发状态 | 0x0000_0000 |
| **SMPTRGB3** | EADC_BA+0x160 | R/W | 采样模块B3 A/D 触发状态 | 0x0000_0000 |

### 6.16.6 寄存器描述

**A/D 数据寄存器 (ADDRA0~7, ADDRB0~7)**

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADDRA0** | EADC_BA+0x00 | R | 采样模组A0的AD数据寄存器0 | 0x0000_0000 |
| **ADDRA1** | EADC_BA+0x04 | R | 采样模组A1的AD数据寄存器1 | 0x0000_0000 |
| **ADDRA2** | EADC_BA+0x08 | R | 采样模组A2的AD数据寄存器2 | 0x0000_0000 |
| **ADDRA3** | EADC_BA+0x0C | R | 采样模组A3的AD数据寄存器3 | 0x0000_0000 |
| **ADDRA4** | EADC_BA+0x10 | R | 采样模组A4的AD数据寄存器4 | 0x0000_0000 |
| **ADDRA5** | EADC_BA+0x14 | R | 采样模组A5的AD数据寄存器5 | 0x0000_0000 |
| **ADDRA6** | EADC_BA+0x18 | R | 采样模组A6的AD数据寄存器6 | 0x0000_0000 |
| **ADDRA7** | EADC_BA+0x1C | R | 采样模组A7的AD数据寄存器7 | 0x0000_0000 |
| **ADDRB0** | EADC_BA+0x20 | R | 采样模组B0的AD数据寄存器8 | 0x0000_0000 |
| **ADDRB1** | EADC_BA+0x24 | R | 采样模组B1的AD数据寄存器9 | 0x0000_0000 |
| **ADDRB2** | EADC_BA+0x28 | R | 采样模组B2的AD数据寄存器10 | 0x0000_0000 |
| **ADDRB3** | EADC_BA+0x2C | R | 采样模组B3的AD数据寄存器11 | 0x0000_0000 |
| **ADDRB4** | EADC_BA+0x30 | R | 采样模组B4的AD数据寄存器12 | 0x0000_0000 |
| **ADDRB5** | EADC_BA+0x34 | R | 采样模组B5的AD数据寄存器13 | 0x0000_0000 |
| **ADDRB6** | EADC_BA+0x38 | R | 采样模组B6的AD数据寄存器14 | 0x0000_0000 |
| **ADDRB7** | EADC_BA+0x3C | R | 采样模组B7的AD数据寄存器15 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:18] | | 保留. |

| | | |
|---|---|---|
| [17] | **VALID** | **有效标志**<br>0 = RSLT[11:0] 里的数据无效.<br>1 = RSLT[11:0] 里的数据有效.<br>这一位当相应的采样模块通道转换完成后置一。当程序读取这一位后硬件自动清0 |
| [16] | **OVERRUN** | **改写标志**<br>0 = RSLT[11:0] 里的数据是最近的结果<br>1 = RSLT[11:0]里的数据被改写了。如果之前RSLT[11:0]里的数据没有被读取，新的数据又写进来， OVERRUN 会置1. 程序读取后硬件自动清0. |
| [15:12] | | 保留. |
| [11:0] | **RSLT** | **A/D 转换结果**<br>这里是12位的转换结果. |

## A/D控制寄存器 (ADCR)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADCR** | EADC_BA+0x40 | R/W | A/D 控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | **ADIE3** | **ADIE2** | **ADIE1** | **ADIE0** | **ADRESET** | **AD_EN** |

| 位 | 描述 | |
|---|---|---|
| [31:6] | | 保留 |
| [5] | **ADIE3** | **选定采样模块AD转换触发ADINT3使能位**<br>0 =禁止选定采样模块AD转换触发ADINT3.<br>1 =使能选定采样模块AD转换触发ADINT3.<br>当选定的采样模块AD转换完成，对应的标志ADF3 置1，如果 ADIE3位使能，那么 中断ADINT3会产生. |
| [4] | **ADIE2** | **选定采样模块AD转换触发ADINT2使能位**<br>0 =禁止选定采样模块AD转换触发ADINT2.<br>1 =使能选定采样模块AD转换触发ADINT2.<br>当选定的采样模块AD转换完成，对应的标志ADF2 置1，如果 ADIE2位使能，那么 中断ADINT2会产生. |
| [3] | **ADIE1** | **选定采样模块AD转换触发ADINT1使能位**<br>0 =禁止选定采样模块AD转换触发ADINT1.<br>1 =使能选定采样模块AD转换触发ADINT1.<br>当选定的采样模块AD转换完成，对应的标志ADF1 置1，如果 ADIE1位使能，那么 中断ADINT1会产生.. |
| [2] | **ADIE0** | **选定采样模块AD转换触发ADINT0使能位**<br>0 =禁止选定采样模块AD转换触发ADINT0.<br>1 =使能选定采样模块AD转换触发ADINT0<br>当选定的采样模块AD转换完成，对应的标志ADF0 置1，如果 ADIE0位使能，那么 中断ADINT0会产生. |
| [1] | **ADRESET** | **ADCA, ADCB A/D 转换控制电路复位**<br>0 = 写0没有作用.<br>1 =写1让ADC控制电路重置到复位状态，但不会改变ADC寄存器的值.<br>**注意:** 在ADC复位期间这一位保持为1，当ADC复位结束， ADRESET 位会自动回到0 |

| 位 | 描述 | |
|---|---|---|
| [0] | **AD_EN** | **A/D 转换使能位**<br>0 = A/D 转换禁止.<br>1 = A/D转换使能.<br>**注意:** 在ADC转换开始之前，这一位要置1.这一位清0可以关闭ADC功能节省功耗。 |

## A/D 通道输入选择寄存器 (ADCHISELR)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADCHISELR** | EADC_BA+0x44 | R/W | A/D 通道输入选择寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | **PRESEL** | | **AINB0SEL** | **AINA0SEL** |

| 位 | 描述 | |
|---|---|---|
| [31:4] | | 保留 |
| [3:2] | **PRESEL** | **A/D 通道 AINA[7] 模拟信号输入选择**<br>00 = 模拟输入通道 AINA7.<br>01 = 带隙电压 (VBG) 输入.<br>10 = VTEMP 内部温度传感器信号输入.<br>11 = 模拟地. |
| [1] | **AINB0SEL** | **A/D 通道 AINB[0] 模拟通道输入选择**<br>0 = AINB[0] 管脚 P7.0E/EADC1_CH0 作为A/D AINB[0] 输入信号.<br>1 = OP 放大器 1 输出作为 A/D AINB[0]输入信号. |
| [0] | **AINA0SEL** | **A/D 通道I AINA[0] 模拟信号输入选择**<br>0 = AINA[0] 管脚P6.0/EADC0_CH0作为 ADC AINA[0] 输入信号.<br>1 = OP 放大器 0输出作为ADC AINA[0] 输入信号. |

## A/D 采样模块软件触发寄存器(ADSSTR)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADSSTR** | EADC_BA+0x48 | W | A/D 采样模块软件触发寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | | 保留. |
| [15:8] | **ADST** | **A/D 采样模块B7~0 软件触发开始ADC 转换寄存器**<br>0 = 没影响.<br>1 = 根据SAMPLEB的优先级开始一次ADC转换. |
| [7:0] | **ADST** | **A/D 采样模块A7~0软件触发开始ADC 转换寄存器**<br>0 = 没影响.<br>1 =根据SAMPLEA的优先级开始一次ADC转换. |

## A/D 采样模块开始转换挂起标志寄存器(ADSTPFR)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADSTPFR** | EADC_BA+0x4C | R | A/D 采样模块开始转换挂起标志寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | | 保留. |
| [15:8] | **STPF** | **A/D 采样模块B7~0 开始转换挂起标志**<br>0 = 采样模块B没有挂起的转换.<br>1 = 对应的采样模块Bn ADC 开始转换但挂起了.<br>在挂起状态下这一位保持为1，如果相应的ADC转换结束了，相对应的STPF 位会自动清0<br>**注意:** n=0~7. |
| [7:0] | **STPF** | **A/D 采样模块A7~0 开始转换挂起标志**<br>0 = 采样模块A没有挂起的转换.<br>1 = 对应的采样模块Bn ADC 开始转换但挂起了.<br>在挂起状态下这一位保持为1，如果相应的ADC转换结束了，相对应的STPF 位会自动清0<br>**注意:** n=0~7. |

## A/D 中断标志覆盖寄存器(ADIFOVR)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADIFOVR** | EADC_BA+0x50 | R/W | A/D ADINT3~0 中断标志覆盖寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | **ADFOV3** | **ADFOV2** | **ADFOV1** | **ADFOV0** |

| 位 | 描述 | |
|---|---|---|
| [31:4] | | 保留. |
| [3] | **ADFOV3** | **A/D ADINT3 中断标志覆盖**<br>0 = ADINT3 中断标志没有被覆盖.<br>1 = 当ADF3为1，接收到ADINT3中断信号<br>**注意:** 此位写1清0 . |
| [2] | **ADFOV2** | **A/D ADINT2 中断标志覆盖**<br>0 = ADINT2 中断标志没有被覆盖.<br>1 = 当ADF2为1，接收到ADINT2中断信号<br>**注意:** 此位写1清0 |
| [1] | **ADFOV1** | **A/D ADINT1 中断标志覆盖**<br>0 = ADINT1 中断标志没有被覆盖.<br>1 = 当ADF1为1，接收到ADINT1中断信号<br>**注意:** 此位写1清0 |
| [0] | **ADFOV0** | **A/D ADINT0 中断标志覆盖**<br>0 = ADINT0 中断标志没有被覆盖.<br>1 = 当ADF0为1，接收到ADINT0中断信号<br>**注意:** 此位写1清0 |

## A/D 采样模块开始转换标志覆盖寄存器 (ADSPOVFR)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADSPOVFR** | EADC_BA+0x54 | R/W | A/D 采样模块开始转换标志覆盖寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:16] | **Reserved** | 保留. |
| [15:8] | **SPOVF** | **A/D 采样模块B0-7 开始转换覆盖标志**<br>0 = 没有采样模块的转换开始标志被覆盖<br>1 = 当过去的转换还在挂起状态，又有新的采样模块Bn 触发了新的转换.<br>如果当上一次触发事件开始的转换还在挂起状态，又有新的事件触发了新的转换，就产生了覆盖事件， SPOVF位会置1.<br>**注意1:** 此位写1清0.<br>**注意2:** n = 0~7. |
| [7:0] | **SPOVF** | **A/D 采样模块A0-7 开始转换覆盖标志**<br>0 = 没有采样模块的转换开始标志被覆盖<br>1 = 当过去的转换还在挂起状态，又有新的采样模块An 触发了新的转换.<br>如果当上一次触发事件开始的转换还在挂起状态，又有新的事件触发了新的转换，就产生了覆盖事件， SPOVF位会置1.<br>**注意1:** 此位写1清0.<br>**注意2:** n = 0~7.. |

## A/D 采样模块A0~3, 采样模块B0~3 控制寄存器 (ADSPCRx0 ~ ADSPCRx3)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADSPCRA0** | EADC_BA+0x58 | R/W | A/D 采样模块A0控制寄存器 | 0x0000_0000 |
| **ADSPCRA1** | EADC_BA+0x5C | R/W | A/D 采样模块A1控制寄存器 | 0x0000_0000 |
| **ADSPCRA2** | EADC_BA+0x60 | R/W | A/D 采样模块A2控制寄存器 | 0x0000_0000 |
| **ADSPCRA3** | EADC_BA+0x64 | R/W | A/D 采样模块A3控制寄存器 | 0x0000_0000 |
| **ADSPCRB0** | EADC_BA+0x78 | R/W | A/D 采样模块B0控制寄存器 | 0x0000_0000 |
| **ADSPCRB1** | EADC_BA+0x7C | R/W | A/D 采样模块B1控制寄存器 | 0x0000_0000 |
| **ADSPCRB2** | EADC_BA+0x80 | R/W | A/D 采样模块B2控制寄存器 | 0x0000_0000 |
| **ADSPCRB3** | EADC_BA+0x84 | R/W | A/D 采样模块B3控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | **EXTFEN** | **EXTREN** | **Reserved** | | **TRGDLYDIV** | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **TRGDLYCNT** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **TRGSEL** | | | | **Reserved** | **CHSEL** | | |

| 位 | 描述 | |
|---|---|---|
| [31:20] | | 保留. |
| [21] | **EXTFEN** | **A/D 外部触发下降沿使能位**<br>0 =当 A/D 转换选择 STADC 做触发源的时候禁止下降沿触发.<br>1 =当 A/D 转换选择 STADC 做触发源的时候使能下降沿触发. |
| [20] | **EXTREN** | **A/D 外部触发上升沿使能位**<br>0 =当 A/D 转换选择 STADC 做触发源的时候禁止上升沿触发.<br>1 =当 A/D 转换选择 STADC 做触发源的时候使能上升沿触发.. |
| [19:18] | | 保留. |
| [17:16] | **TRGDLYDIV** | **A/D采样模块开始转换延时除频选择**<br>触发延时时钟频率:<br>00 = ADC_CLK/1.<br>01 = ADC_CLK/2.<br>10 = ADC_CLK/4.<br>11 = ADC_CLK/16. |

| 位 | 描述 | |
|---|---|---|
| [15:8] | **TRGDLYCNT** | **A/D 采样模块开始转换延时时间**<br>延时时间 =TRGDLYCNT x ADC_CLK x n (n=1, 2, 4, 16 TRGDLYDIV 的设定). |
| [7:4] | **TRGSEL** | **A/D 采样开始转换触发源选择**<br>0000 = 禁止硬件触发<br>0001 = 外部STADC管脚输入.<br>0010 = ADC ADINT0 中断 EOC信号触发.<br>0011 = ADC ADINT1中断 EOC信号触发<br>0100 = Timer0 溢出信号触发.<br>0101 = Timer1溢出信号触发.<br>0110 = Timer2溢出信号触发<br>0111 = Timer3溢出信号触发.<br>1000 = PWM00触发.<br>1001 = PWM02触发<br>1010 = PWM04触发<br>1011 = PWM10触发.<br>1100 = PWM12触发<br>1101 = PWM14触发.<br>1110 = PWM20触发.<br>1111 = PWM21触发 |
| 3 | **Reserved** | 保留 |
| [2:0] | **CHSEL** | **A/D 采样模块A,B 通道选择**<br>000 = AINx[0].<br>001 = AINx[1].<br>010 = AINx[2].<br>011 = AINx[3].<br>100 = AINx[4].<br>101 = AINx[5].<br>110 = AINx[6].<br>111 = AINx[7]. |

## A/D 采样模块A4~7, 采样模块4~7 控制寄存器 (ADSPCRx4 ~ ADSPCRx7)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADSPCRA4** | EADC_BA+0x68 | R/W | A/D 采样模块A4控制寄存器 | 0x0000_0000 |
| **ADSPCRA5** | EADC_BA+0x6C | R/W | A/D 采样模块A5控制寄存器 | 0x0000_0000 |
| **ADSPCRA6** | EADC_BA+0x70 | R/W | A/D 采样模块A6控制寄存器 | 0x0000_0000 |
| **ADSPCRA7** | EADC_BA+0x74 | R/W | A/D 采样模块A7控制寄存器 | 0x0000_0000 |
| **ADSPCRB4** | EADC_BA+0x88 | R/W | A/D 采样模块B4控制寄存器 | 0x0000_0000 |
| **ADSPCRB5** | EADC_BA+0x8C | R/W | A/D 采样模块B5控制寄存器 | 0x0000_0000 |
| **ADSPCRB6** | EADC_BA+0x90 | R/W | A/D 采样模块B6控制寄存器 | 0x0000_0000 |
| **ADSPCRB7** | EADC_BA+0x94 | R/W | A/D 采样模块B7控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | **EXTFEN** | **EXTREN** | **Reserved** | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | **TRGSEL** | | | **Reserved** | **CHSEL** | | |

| 位 | 描述 | |
|---|---|---|
| [31:22] | | 保留 |
| [21] | **EXTFEN** | **A/D 外部触发下降沿使能位**<br>0 =当 A/D 转换选择 STADC 做触发源的时候禁止下降沿触发.<br>1 =当 A/D 转换选择 STADC 做触发源的时候使能下降沿触发. |
| [20] | **EXTREN** | **A/D 外部触发上升沿使能位**<br>0 =当 A/D 转换选择 STADC 做触发源的时候禁止上升沿触发.<br>1 =当 A/D 转换选择 STADC 做触发源的时候使能上升沿触发.. |
| [19:7] | | 保留. |

| 位 | 描述 | |
|---|---|---|
| [6:4] | **TRGSEL** | **A/D 采样开始转换触发源选择**<br>000 = 禁止硬件触发<br>001 = 外部STADC管脚输入.<br>010 = ADC ADINT0 中断 EOC信号触发.<br>011 = ADC ADINT1中断 EOC信号触发<br>100 = Timer0 溢出信号触发.<br>101 = Timer1溢出信号触发.<br>110 = Timer2溢出信号触发<br>111 = Timer3溢出信号触发. |
| [3] | | 保留. |
| [2:0] | **CHSEL** | **A/D 采样模块A,B 通道选择**<br>000 = AINx[0].<br>001 = AINx[1].<br>010 = AINx[2].<br>011 = AINx[3].<br>100 = AINx[4].<br>101 = AINx[5].<br>110 = AINx[6].<br>111 = AINx[7]. |

## A/D 同步采样模式选择寄存器 (ADSMSELR)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADSMSELR** | EADC_BA+0xA4 | R/W | A/D 采样模块同步采样模式选择寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **SIMUSEL7** | **SIMUSEL6** | **SIMUSEL5** | **SIMUSEL4** | **SIMUSEL3** | **SIMUSEL2** | **SIMUSEL1** | **SIMUSEL0** |

| 位 | 描述 | |
|---|---|---|
| [31:8] | | 保留 |
| [7] | **SIMUSEL7** | **A/D 采样模块A7, 采样模块B7 同步采样模式选择**<br>0 = 采样模块A7, 采样模块B7工作在独立采样模式, 采样模块A7 和采样模块B7相应的CHSEL寄存器决定 哪一路ADC通道被转换。<br>1 =采样模块A7, 采样模块B7 工作在同步采样模式, 只有采样模块A7 可以同时触发采样模块A7和采样模块B7的同步采样, 采样模块B7触发选择TRGSEL 被忽略。 如果采样模块A7的CHSEL = 1, 采样模块B7'的 CHSEL = 3, 采样对选择为 AINA[1], AINB[3], 它们会同时开始ADC转换. |
| [6] | **SIMUSEL6** | **A/D 采样模块A6, 采样模块B6 同步采样模式选择**<br>0 = 采样模块A6, 采样模块B6工作在独立采样模式, 采样模块A6 和采样模块B6相应的CHSEL寄存器决定 哪一路ADC通道被转换。<br>1 =采样模块A6, 采样模块B6 工作在同步采样模式, 只有采样模块A6 可以同时触发采样模块A6和采样模块B6的同步采样, 采样模块B6触发选择TRGSEL 被忽略。 如果采样模块A6的CHSEL = 1, 采样模块B6'的 CHSEL = 3, 采样对选择为 AINA[1], AINB[3], 它们会同时开始ADC转换. |
| [5] | **SIMUSEL5** | **A/D 采样模块A5, 采样模块B5 同步采样模式选择**<br>0 = 采样模块A5, 采样模块B5工作在独立采样模式, 采样模块A5 和采样模块B5相应的CHSEL寄存器决定 哪一路ADC通道被转换。<br>1 =采样模块A5, 采样模块B5 工作在同步采样模式, 只有采样模块A5 可以同时触发采样模块A5和采样模块B5的同步采样, 采样模块B5触发选择TRGSEL 被忽略。 如果采样模块A5的CHSEL = 1, 采样模块B5的 CHSEL = 3, 采样对选择为 AINA[1], AINB[3], 它们会同时开始ADC转换. |

| 位 | 描述 | |
|---|---|---|
| [4] | **SIMUSEL4** | **A/D 采样模块A4, 采样模块B4 同步采样模式选择**<br>0 = 采样模块A4, 采样模块B4工作在独立采样模式, 采样模块A4 和采样模块B4相应的CHSEL寄存器决定 哪一路ADC通道被转换。<br>1 =采样模块A4, 采样模块B4 工作在同步采样模式, 只有采样模块A4 可以同时触发采样模块A4和采样模块B4的同步采样, 采样模块B4触发选择TRGSEL 被忽略。 如果采样模块A4的CHSEL = 1, 采样模块B4'的 CHSEL = 3, 采样对选择为 AINA[1], AINB[3], 它们会同时开始ADC转换. |
| [3] | **SIMUSEL3** | **A/D 采样模块A3, 采样模块B3 同步采样模式选择**<br>0 = 采样模块A3, 采样模块B3工作在独立采样模式, 采样模块A3 和采样模块B3相应的CHSEL寄存器决定 哪一路ADC通道被转换。<br>1 =采样模块A3, 采样模块B3 工作在同步采样模式, 只有采样模块A3 可以同时触发采样模块A3和采样模块B3的同步采样, 采样模块B3触发选择TRGSEL 被忽略。 如果采样模块A3的CHSEL = 1, 采样模块B3'的 CHSEL = 3, 采样对选择为 AINA[1], AINB[3], 它们会同时开始ADC转换. |
| [2] | **SIMUSEL2** | **A/D 采样模块A2, 采样模块B2 同步采样模式选择**<br>0 = 采样模块A2, 采样模块B2工作在独立采样模式, 采样模块A2 和采样模块B2相应的CHSEL寄存器决定 哪一路ADC通道被转换。<br>1 =采样模块A2, 采样模块B2 工作在同步采样模式, 只有采样模块A2 可以同时触发采样模块A2和采样模块B2的同步采样, 采样模块B2触发选择TRGSEL 被忽略。 如果采样模块A2的CHSEL = 1, 采样模块B2'的 CHSEL = 3, 采样对选择为 AINA[1], AINB[3], 它们会同时开始ADC转换. |
| [1] | **SIMUSEL1** | **A/D 采样模块A1, 采样模块B1 同步采样模式选择**<br>0 = 采样模块A1, 采样模块B1工作在独立采样模式, 采样模块A1 和采样模块B1相应的CHSEL寄存器决定 哪一路ADC通道被转换。<br>1 =采样模块A1, 采样模块B1 工作在同步采样模式, 只有采样模块A1 可以同时触发采样模块A1和采样模块B1的同步采样, 采样模块B1触发选择TRGSEL 被忽略。 如果采样模块A1的CHSEL = 1, 采样模块B1'的 CHSEL = 3, 采样对选择为 AINA[1], AINB[3], 它们会同时开始ADC转换. |
| [0] | **SIMUSEL0** | **A/D 采样模块A0, 采样模块B0 同步采样模式选择**<br>0 = 采样模块A0, 采样模块B0工作在独立采样模式, 采样模块A0 和采样模块B0相应的CHSEL寄存器决定 哪一路ADC通道被转换。<br>1 =采样模块A0, 采样模块B0 工作在同步采样模式, 只有采样模块A0可以同时触发采样模块A0和采样模块B0的同步采样, 采样模块B0触发选择TRGSEL 被忽略。 如果采样模块A0的CHSEL = 1, 采样模块B0'的 CHSEL = 3, 采样对选择为 AINA[1], AINB[3], 它们会同时开始ADC转换. |

## A/D 结果比较寄存器 0/1 (ADCMPR0/1)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADCMPR0** | EADC_BA+0xA8 | R/W | A/D 结果比较寄存器 0 | 0x0000_0000 |
| **ADCMPR1** | EADC_BA+0xAC | R/W | A/D 结果比较寄存器 1 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | CMPD | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| CMPD | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | CMPMATCNT | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | CMPSMPL | | | CMPCOND | ADCMPIE | ADCMP_EN |

| 位 | 描述 | |
|---|---|---|
| [31:28] | | 保留. |
| [27:16] | **CMPD** | **比较值**<br>这个12位的数据用来和和采样模块的转换结果比较。程序可以用这个功能监控输入模拟信号的情况，而不用一直用软件监控转换结果. |
| [15:12] | | 保留. |
| [11:8] | **CMPMATCNT** | **满足比较条件计数门限**<br>当选定的 A/D 采样通道的转换结果满足CMPCOND所设定的转换条件, 内部满足比较条件计数器会加 1. 当这个内部计数器的值达到 (CMPMATCNT +1), ADCMPFn(ADSR1[7:6], n=0, 1) 位会置1. |
| [7:6] | | 保留 |
| [5:3] | **CMPSMPL** | **比较采样模块选择**<br>000 = 选择采样模块A0 的比较结果ADDRA0来和比较值比较.<br>001 = 选择采样模块A1的比较结果ADDRA1来和比较值比较<br>010 =选择采样模块A2 的比较结果ADDRA2来和比较值比较.<br>011 =选择采样模块A3 的比较结果ADDRA3来和比较值比较.<br>100 =选择采样模块B0 的比较结果ADDRB0来和比较值比较.<br>101 =选择采样模块B1 的比较结果ADDRB1来和比较值比较.<br>110 =选择采样模块B2 的比较结果ADDRB2来和比较值比较.<br>111 =选择采样模块B3 的比较结果ADDRB3来和比较值比较. |

| 位 | 描述 | |
|---|---|---|
| [2] | **CMPCOND** | **比较条件**<br>0= 设定比较条件为，当12位ADC转换结果小于12位寄存器CMPD(ADCMPRn[27:16], n=0, 1)的值时, 内部满足比较条件计数器会加1.<br>1=设定比较条件为，当12位ADC转换结果大于等于12位寄存器CMPD(ADCMPRn[27:16], n=0, 1)的值时, 内部满足比较条件计数器会加1..<br>**注意:** 当内部计数器的值达到 (CMPMATCNT +1), ADCMPFn(ADSR1[7:6], n=0, 1) 位会置1. |
| [1] | **ADCMPIE** | **A/D 结果比较中断使能位**<br>0 = 比较功能中断禁止.<br>1 = 比较功能中断使能.<br>如果使能了比较功能，并且比较结果满足CMPCOND 和 CMPMATCNT的设定, ADCMPFn(ADSR1[7:6], n=0, 1) 位会置1, 同时, 如果ADCMPIE 设为1,会产生一次比较中断 |
| [0] | **ADCMP_EN** | **A/D 结果比较功能使能位**<br>0 = 禁止比较功能.<br>1 = 使能比较功能.<br>这一位写1使能比较功能，比较CMPD的值和选定的采样模块转换完后存入ADDRAn 或 ADDRBn(n=0~3)寄存器的采样结果 |

## A/D 状态寄存器 0 (ADSR0)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADSR0** | EADC_BA+0xB0 | R | A/D 状态寄存器 0 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | **OVERRUN** | **ADDRB7~0 覆盖标志**<br>这一位是采样模块B转换结果寄存器ADDRBn, n=0~7的 OVERRUN位的镜像 |
| [23:16] | **OVERRUN** | **ADDRA7~0覆盖标志**<br>这一位是采样模块A转换结果寄存器ADDRAn, n=0~7的 OVERRUN位的镜像. |
| [15:8] | **VALID** | **ADDRB7~0 数据有效标志**<br>这一位是采样模块B转换结果寄存器ADDRBn, n=0~7的 VALID位的镜像 |
| [7:0] | **VALID** | **ADDRA7~0数据有效标志**<br>这一位是采样模块A转换结果寄存器ADDRAn, n=0~7的 VALID位的镜像 |

## A/D 状态寄存器 1 (ADSR1)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADSR1** | EADC_BA+0xB4 | R/W | A/D 状态寄存器 1 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | **AOVERRUN** | **AVALID** | **ASPOVF** | **AADFOV** |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | **CHANNELB** | | | **Reserved** | | | **BUSYB** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | **CHANNELA** | | | **Reserved** | | | **BUSYA** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **ADCMPF1** | **ADCMPF0** | **ADCMPO1** | **ADCMPO0** | **ADF3** | **ADF2** | **ADF1** | **ADF0** |

| 位 | 描述 | |
|---|---|---|
| [31:28] | | 保留. |
| [27] | **AOVERRUN** | **全体采样模块A/D 转换结果覆盖标志检查**<br>0 = 没有采样模块的数据寄存器的覆盖标志OVERRUNn置为1<br>1 =有采样模块的数据寄存器的覆盖标志OVERRUNn置为1<br>当任意一个OVERRUNn 标志是1此位就为1. OVERRUNn = OVERRUN(ADDRAn[16], ADDRBn[16], n=0~7). |
| [26] | **AVALID** | **全体采样模块A/D 转换结果数据有效标志检查**<br>0 = 没有采样模块的数据寄存器的数据有效标志VALIDn置为1<br>1 =有采样模块的数据寄存器的数据有效标志置VALIDn为1<br>当任意一个VALIDn 标志是1此位就为1. VALIDn = VALID(ADDRAn[17], ADDRBn[17], n=0~7). |
| [25] | **ASPOVF** | **全体采样模块开始转换标志覆盖检查**<br>0 = 没有采样模块的开始转换标志覆盖指示SPOVFn 设为 1.<br>1 =有采样模块的开始转换标志覆盖指示SPOVFn 设为 1.<br>当任意一个SPOVFn标志为1此位就为1. SPOVFn = SPOVF(ADSPOVFR[n], n=0~15). |
| [24] | **AADFOV** | **全体采样模块中断标志覆盖检查**<br>0 = 没有中断标志ADFOVn被覆盖为1.<br>1 =有中断标志ADFOVn被覆盖为1.<br>这一位当任意一个ADFOVn 为1此位为1. ADFOVn = ADFOV(ADIFOVR[n], n=0~3). |
| [23] | **Reserved** | 保留. |

| 位 | 描述 | |
|---|---|---|
| [22:20] | **CHANNELB** | **目前转换通道 (只读)**<br>当 BUSYB=1，这里指示B组ADC目前的转换通道. 当 BUSYB=0, 指示上一次转换的通道.<br>000 = AINB[0].<br>001 = AINB[1].<br>010 = AINB[2].<br>011 = AINB[3].<br>100 = AINB[4].<br>101 = AINB[5].<br>110 = AINB[6].<br>111 = AINB[7]. |
| [19:17] | | 保留. |
| [16] | **BUSYB** | **忙闲指示 (只读)**<br>0 = A/D 转换器 B (ADCB) 空闲.<br>1 = A/D 转换器 B (ADCB) 正在转换. |
| [14:12] | **CHANNELA** | **目前转换通道(只读)**<br>当 BUSYA=1，这里指示A组ADC目前的转换通道. 当 BUSYA=0, 指示上一次转换的通道.<br>000 = AINA[0].<br>001 = AINA[1].<br>010 = AINA[2].<br>011 = AINA[3].<br>100 = AINA[4].<br>101 = AINA[5].<br>110 = AINA[6].<br>111 = AINA[7]. |
| [11:9] | | 保留. |
| [8] | **BUSYA** | **忙闲指示 (只读)**<br>0 = A/D 转换器 A(ADCA) 空闲.<br>1 = A/D 转换器 A (ADCA) 正在转换. |
| [7] | **ADCMPF1** | **ADC 比较器1 标志**<br>当选择的采样模块的AD转换结果符合ADCMPR1寄存器的设定，这一位置1.此位写1清0<br>0 = ADDR中的转换结果不满足ADCMPR1中的设定.<br>1 = ADDR中的转换结果满足ADCMPR1中的设定 |
| [6] | **ADCMPF0** | **ADC 比较器0 标志**<br>当选择的采样模块的AD转换结果符合ADCMPR0寄存器的设定，这一位置1.此位写1清0<br>0 = ADDR中的转换结果不满足ADCMPR0中的设定.<br>1 = ADDR中的转换结果满足ADCMPR0中的设定 |
| [5] | **ADCMPO1** | **ADC 比较器 1 输出状态位**<br>CMPD(ADCMPR1[27:16])里的12位数用来和选定的采样模块的转换结果比较。程序可以监控这一位来了解外部模拟输入信号的状态。<br>0 = ADDR里的转换结果小于 CMPD(ADCMPR1[27:16]) 的设定.<br>1 = ADDR里的转换结果大于等于 CMPD(ADCMPR1[27:16]) 的设定. |

| 位 | 描述 | |
|---|---|---|
| [4] | **ADCMPO0** | **ADC 比较器 0 输出状态位**<br>CMPD(ADCMPR0[27:16])里的12位数用来和选定的采样模块的转换结果比较。程序可以监控这一位来了解外部模拟输入信号的状态。<br>0 = ADDR里的转换结果小于 CMPD(ADCMPR0[27:16]) 的设定.<br>1 = ADDR里的转换结果大于等于 CMPD(ADCMPR0[27:16]) 的设定. |
| [3] | **ADF3** | **A/D ADINT3中断标志**<br>0 = 没收到 ADINT3 中断信号.<br>1 = 收到ADINT3 中断信号.<br>**注意1:** 此位写1清0.<br>**注意2: 此位指示选定的采样模块转换完成了** |
| [2] | **ADF2** | **A/D ADINT2中断标志**<br>0 = 没收到 ADINT2 中断信号.<br>1 = 收到ADINT2 中断信号.<br>**注意1:** 此位写1清0.<br>**注意2: 此位指示选定的采样模块转换完成了** |
| [1] | **ADF1** | **A/D ADINT1中断标志**<br>0 = 没收到 ADINT1 中断信号.<br>1 = 收到ADINT1 中断信号.<br>**注意1:** 此位写1清0.<br>**注意2: 此位指示选定的采样模块转换完成了** |
| [0] | **ADF0** | **A/D ADINT0中断标志**<br>0 = 没收到 ADINT0 中断信号.<br>1 = 收到ADINT0 中断信号.<br>**注意1:** 此位写1清0.<br>**注意2: 此位指示选定的采样模块转换完成了** |

## A/D 时序控制寄存器 (ADTCR)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADTCR** | EADC_BA+0xB8 | R/W | A/D时序控制寄存器 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:24] | | 保留 |
| [23:16] | **ADBEST** | **ADCB 增加采样时间**<br>当AD转换频率较高，如果输入信号内阻很大，采样时间可能不够。用户可以在收到触发信号后增加一段延时来获得足够的AD采样时间。<br>延时的范围是 0~255 ADC 时钟周期. |
| [15:8] | | 保留. |
| [7:0] | **ADAEST** | **ADCA增加采样时间**<br>当AD转换频率较高，如果输入信号内阻很大，采样时间可能不够。用户可以在收到触发信号后增加一段延时来获得足够的AD采样时间。<br>延时的范围是 0~255 ADC 时钟周期. |

## A/D 数据寄存器双缓存寄存器 (ADDRDBA0~3, ADDRDBB0~3)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADDRDBA0** | EADC_BA+0x100 | R | 采样模块A0 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBA1** | EADC_BA+0x104 | R | 采样模块A1 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBA2** | EADC_BA+0x108 | R | 采样模块A2 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBA3** | EADC_BA+0x10C | R | 采样模块A3 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBB0** | EADC_BA+0x120 | R | 采样模块B0 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBB1** | EADC_BA+0x124 | R | 采样模块B1 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBB2** | EADC_BA+0x128 | R | 采样模块B2 A/D 数据寄存器双缓存 | 0x0000_0000 |
| **ADDRDBB3** | EADC_BA+0x12C | R | 采样模块B3 A/D 数据寄存器双缓存 | 0x0000_0000 |

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

| 位 | 描述 | |
|---|---|---|
| [31:17] | | 保留 |
| [16] | **VALID** | **数据有效标志**<br>0 = RSLTDB[11:0]里的双缓存数据无效.<br>1 = RSLTDB[11:0]里的双缓存数据有效.<br>这一位当相应的采样模块数据转换完成后为1，当ADDRDBAn 和ADDRDBBn寄存器读完后清0.(n=0~3). |
| [15:12] | | 保留. |
| [11:0] | **RSLTDB** | **A/D 转换结果**<br>这一位包含12位的转换结果，依照 DBMAn (n=0~3, ADDBM[3:0]) 或 DBMBn(n=0~3, ADDBM[11:8]) 的设定. |

## A/D 双缓存模式选择 (ADDBM)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADDBM** | EADC_BA+0x130 | R/W | A/D 双缓存模式选择 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **Reserved** | | | | **DBMB3** | **DBMB2** | **DBMB1** | **DBMB0** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | | | | **DBMA3** | **DBMA2** | **DBMA1** | **DBMA0** |

| 位 | 描述 | |
|---|---|---|
| [31:12] | | 保留 |
| [11] | **DBMB3** | **采样模块 B3 双缓存模式**<br>0 = 采样模块B3 只有一个结果寄存器. (默认)<br>1 = 采样模块B3 有两个结果寄存器. |
| [10] | **DBMB2** | **采样模块 B2 双缓存模式**<br>0 = 采样模块B2 只有一个结果寄存器. (默认)<br>1 = 采样模块B2 有两个结果寄存器. |
| [9] | **DBMB1** | **采样模块 B1 双缓存模式**<br>0 = 采样模块B1只有一个结果寄存器. (默认)<br>1 = 采样模块B1 有两个结果寄存器. |
| [8] | **DBMB0** | **采样模块 B0 双缓存模式**<br>0 = 采样模块B0 只有一个结果寄存器. (默认)<br>1 = 采样模块B0 有两个结果寄存器. |
| [7:4] | | 保留. |
| [3] | **DBMA3** | **采样模块 A3 双缓存模式**<br>0 = 采样模块A3 只有一个结果寄存器. (默认)<br>1 = 采样模块A3 有两个结果寄存器.. |
| [2] | **DBMA2** | **采样模块 A2 双缓存模式**<br>0 = 采样模块A2 只有一个结果寄存器. (默认)<br>1 = 采样模块A2 有两个结果寄存器. |
| [1] | **DBMA1** | **采样模块 A1 双缓存模式**<br>0 = 采样模块A1 只有一个结果寄存器. (默认)<br>1 = 采样模块A1 有两个结果寄存器 |

| 位 | 描述 | |
|---|---|---|
| [0] | **DBMA0** | **采样模块 A0 双缓存模式**<br>0 = 采样模块A0只有一个结果寄存器. (默认)<br>1 = 采样模块A0 有两个结果寄存器 |

## A/D 中断源使能控制寄存器 (ADINT0SRCTL ~ ADINT3SRCTL)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **ADINT0SRCTL** | EADC_BA+0x134 | R/W | A/D 中断 0 中断源使能控制寄存器 | 0x0000_0000 |
| **ADINT1SRCTL** | EADC_BA+0x138 | R/W | A/D 中断 1 中断源使能控制寄存器 | 0x0000_0000 |
| **ADINT2SRCTL** | EADC_BA+0x13C | R/W | A/D 中断 2 中断源使能控制寄存器 | 0x0000_0000 |
| **ADINT3SRCTL** | EADC_BA+0x140 | R/W | A/D 中断 3 中断源使能控制寄存器. | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **Reserved** | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **Reserved** | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **IESPLB7** | **IESPLB6** | **IESPLB5** | **IESPLB4** | **IESPLB3** | **IESPLB2** | **IESPLB1** | **IESPLB0** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **IESPLA7** | **IESPLA6** | **IESPLA5** | **IESPLA4** | **IESPLA3** | **IESPLA2** | **IESPLA1** | **IESPLA0** |

| 位 | 描述 | |
|---|---|---|
| [31:16] | | 保留. |
| [15] | **IESPLB7** | **采样模块 B7 中断屏蔽使能**<br>0 = 采样模块 B7 中断不屏蔽.<br>1 = 屏蔽采样模块 B7 的中断. |
| [14] | **IESPLB6** | **采样模块 B6 中断屏蔽使能**<br>0 = 采样模块 B6 中断不屏蔽.<br>1 = 屏蔽采样模块 B6 的中断. |
| [13] | **IESPLB5** | **采样模块 B5 中断屏蔽使能**<br>0 = 采样模块 B5 中断不屏蔽.<br>1 = 屏蔽采样模块 B5 的中断 |
| [12] | **IESPLB4** | **采样模块 B4 中断屏蔽使能**<br>0 = 采样模块 B4 中断不屏蔽.<br>1 = 屏蔽采样模块 B4 的中断 |
| [11] | **IESPLB3** | **采样模块 B3 中断屏蔽使能**<br>0 = 采样模块 B3 中断不屏蔽.<br>1 = 屏蔽采样模块 B3 的中断 |
| [10] | **IESPLB2** | **采样模块 B2 中断屏蔽使能**<br>0 = 采样模块 B2 中断不屏蔽.<br>1 = 屏蔽采样模块 B2 的中断. |

| 位 | 描述 | |
|---|---|---|
| [9] | **IESPLB1** | **采样模块 B1 中断屏蔽使能**<br>0 = 采样模块 B1 中断不屏蔽.<br>1 = 屏蔽采样模块 B1 的中断 |
| [8] | **IESPLB0** | **采样模块 B0 中断屏蔽使能**<br>0 = 采样模块 B0 中断不屏蔽.<br>1 = 屏蔽采样模块 B0 的中断 |
| [7] | **IESPLA7** | **采样模块 A7 中断屏蔽使能**<br>0 = 采样模块 A7 中断不屏蔽.<br>1 = 屏蔽采样模块 A7 的中断. |
| [6] | **IESPLA6** | **采样模块 A6 中断屏蔽使能**<br>0 = 采样模块 A6 中断不屏蔽.<br>1 = 屏蔽采样模块 A6 的中断. |
| [5] | **IESPLA5** | **采样模块 A5 中断屏蔽使能**<br>0 = 采样模块 A5 中断不屏蔽.<br>1 = 屏蔽采样模块 A5 的中断 |
| [4] | **IESPLA4** | **采样模块 A4 中断屏蔽使能**<br>0 = 采样模块 A4 中断不屏蔽.<br>1 = 屏蔽采样模块 A4 的中断 |
| [3] | **IESPLA3** | **采样模块 A3 中断屏蔽使能**<br>0 = 采样模块 A3 中断不屏蔽.<br>1 = 屏蔽采样模块 A3 的中断 |
| [2] | **IESPLA2** | **采样模块 A2 中断屏蔽使能**<br>0 = 采样模块 A2 中断不屏蔽.<br>1 = 屏蔽采样模块 A2 的中断. |
| [1] | **IESPLA1** | **采样模块 A1 中断屏蔽使能**<br>0 = 采样模块 A1 中断不屏蔽.<br>1 = 屏蔽采样模块 A1 的中断 |
| [0] | **IESPLA0** | **采样模块 A0 中断屏蔽使能**<br>0 = 采样模块 A0 中断不屏蔽.<br>1 = 屏蔽采样模块 A0 的中断 |

## A/D 采样模块触发条件使能寄存器 (SMPTRGA0~ SMPTRGA3, SMPTRGB0~ SMPTRGB3)

| 寄存器 | 偏移 | 读写 | 描述 | 复位值 |
|---|---|---|---|---|
| **SMPTRGA0** | EADC_BA+0x144 | R/W | 采样模块A0 A/D 触发条件寄存器 | 0x0000_0000 |
| **SMPTRGA1** | EADC_BA+0x148 | R/W | 采样模块A1 A/D 触发条件寄存器 | 0x0000_0000 |
| **SMPTRGA2** | EADC_BA+0x14C | R/W | 采样模块A2 A/D 触发条件寄存器 | 0x0000_0000 |
| **SMPTRGA3** | EADC_BA+0x150 | R/W | 采样模块A3 A/D 触发条件寄存器 | 0x0000_0000 |
| **SMPTRGB0** | EADC_BA+0x154 | R/W | 采样模块B0 A/D 触发条件寄存器 | 0x0000_0000 |
| **SMPTRGB1** | EADC_BA+0x158 | R/W | 采样模块B1 A/D 触发条件寄存器 | 0x0000_0000 |
| **SMPTRGB2** | EADC_BA+0x15C | R/W | 采样模块B2 A/D 触发条件寄存器 | 0x0000_0000 |
| **SMPTRGB3** | EADC_BA+0x160 | R/W | 采样模块B3 A/D 触发条件寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| **PWM21CEN** | **PWM21PEN** | **PWM21FEN** | **PWM21REN** | **PWM20CEN** | **PWM20PEN** | **PWM20FEN** | **PWM20REN** |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| **PWM14CEN** | **PWM14PEN** | **PWM14FEN** | **PWM14REN** | **PWM12CEN** | **PWM12PEN** | **PWM12FEN** | **PWM12REN** |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| **PWM10CEN** | **PWM10PEN** | **PWM10FEN** | **PWM10REN** | **PWM04CEN** | **PWM04PEN** | **PWM04FEN** | **PWM04REN** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PWM02CEN** | **PWM02PEN** | **PWM02FEN** | **PWM02REN** | **PWM00CEN** | **PWM00PEN** | **PWM00FEN** | **PWM00REN** |

| 位 | 描述 | |
|---|---|---|
| [31] | **PWM21CEN** | **BPWM0_CH1 中心触发使能位**<br>0 = 禁止BPWM0_CH1 中心触发.<br>1 = 使能BPWM0_CH1 中心触发. |
| [30] | **PWM21PEN** | **BPWM0_CH1 周期触发使能位**<br>0 = 禁止BPWM0_CH1 周期触发.<br>1 =使能 BPWM0_CH1 周期触发. |
| [29] | **PWM21FEN** | **BPWM0_CH1 下降沿触发使能位**<br>0 =禁止 BPWM0_CH1 下降沿触发.<br>1 = 使能BPWM0_CH1 下降沿触发. |
| [28] | **PWM21REN** | **BPWM0_CH1 上升沿触发使能位**<br>0 = 禁止BPWM0_CH1 上升沿触发.<br>1 = 使能BPWM0_CH1上升沿触发. |

| 位 | 描述 | |
|---|---|---|
| [27] | **PWM20CEN** | **BPWM0_CH0 中心触发使能位**<br>0 = 禁止BPWM0_CH0 中心触发.<br>1 = 使能BPWM0_CH0 中心触发. |
| [26] | **PWM20PEN** | **BPWM0_CH0 周期触发使能位**<br>0 = 禁止BPWM0_CH0 周期触发.<br>1 =使能 BPWM0_CH0 周期触发. |
| [25] | **PWM20FEN** | **BPWM0_CH0 下降沿触发使能位**<br>0 =禁止 BPWM0_CH0下降沿触发.<br>1 = 使能BPWM0_CH0 下降沿触发. |
| [24] | **PWM20REN** | **BPWM0_CH0上升沿触发使能位**<br>0 = 禁止BPWM0_CH0 上升沿触发.<br>1 = 使能BPWM0_CH0上升沿触发. |
| [23] | **PWM14CEN** | **PWM1_CH4中心触发使能位**<br>0 = 禁止PWM1_CH4中心触发<br>1 = 使能PWM1_CH4中心触发 |
| [22] | **PWM14PEN** | **PWM1_CH4周期触发使能位**<br>0 =禁止 PWM1_CH4 周期触发.<br>1 = 使能PWM1_CH4 周期触发. |
| [21] | **PWM14FEN** | **PWM1_CH4 下降沿触发使能位**<br>0 = 禁止PWM1_CH4下降沿触发.<br>1 =使能 PWM1_CH4下降沿触发. |
| [20] | **PWM14REN** | **PWM1_CH4 上升沿触发使能位**<br>0 = 禁止PWM1_CH4上升沿触发.<br>1 = 使能PWM1_CH4 上升沿触发. |
| [19] | **PWM12CEN** | **PWM1_CH2中心触发使能位**<br>0 = 禁止PWM1_CH2中心触发<br>1 = 使能PWM1_CH2中心触发 |
| [18] | **PWM12PEN** | **PWM1_CH2周期触发使能位**<br>0 =禁止 PWM1_CH2 周期触发.<br>1 = 使能PWM1_CH2 周期触发. |
| [17] | **PWM12FEN** | **PWM1_CH2 下降沿触发使能位**<br>0 = 禁止PWM1_CH2下降沿触发.<br>1 =使能 PWM1_CH2下降沿触发. |
| [16] | **PWM12REN** | **PWM1_CH2 上升沿触发使能位**<br>0 = 禁止PWM1_CH2上升沿触发.<br>1 = 使能PWM1_CH2 上升沿触发. |
| [15] | **PWM10CEN** | **PWM1_CH0中心触发使能位**<br>0 = 禁止PWM1_CH0中心触发<br>1 = 使能PWM1_CH0中心触发 |

| 位 | 描述 | |
|---|---|---|
| [14] | **PWM10PEN** | **PWM1_CH0周期触发使能位**<br>0 =禁止 PWM1_CH0 周期触发.<br>1 = 使能PWM1_CH0 周期触发. |
| [13] | **PWM10FEN** | **PWM1_CH0 下降沿触发使能位**<br>0 = 禁止PWM1_CH0下降沿触发.<br>1 =使能 PWM1_CH0下降沿触发. |
| [12] | **PWM10REN** | **PWM1_CH0 上升沿触发使能位**<br>0 = 禁止PWM1_CH0上升沿触发.<br>1 = 使能PWM1_CH0 上升沿触发. |
| [11] | **PWM04CEN** | **PWM0_CH4中心触发使能位**<br>0 = 禁止PWM0_CH4中心触发<br>1 = 使能PWM0_CH4中心触发 |
| [10] | **PWM04PEN** | **PWM0_CH4周期触发使能位**<br>0 =禁止 PWM0_CH4 周期触发.<br>1 = 使能PWM0_CH4 周期触发. |
| [9] | **PWM04FEN** | **PWM0_CH4 下降沿触发使能位**<br>0 = 禁止PWM0_CH4下降沿触发.<br>1 =使能 PWM0_CH4下降沿触发. |
| [8] | **PWM04REN** | **PWM0_CH4 上升沿触发使能位**<br>0 = 禁止PWM0_CH4上升沿触发.<br>1 = 使能PWM0_CH4 上升沿触发. |
| [7] | **PWM02CEN** | **PWM0_CH2中心触发使能位**<br>0 = 禁止PWM0_CH2中心触发<br>1 = 使能PWM0_CH2中心触发 |
| [6] | **PWM02PEN** | **PWM0_CH2周期触发使能位**<br>0 =禁止 PWM0_CH2 周期触发.<br>1 = 使能PWM0_CH2 周期触发. |
| [5] | **PWM02FEN** | **PWM0_CH2 下降沿触发使能位**<br>0 = 禁止PWM0_CH2下降沿触发.<br>1 =使能 PWM0_CH2下降沿触发. |
| [4] | **PWM02REN** | **PWM0_CH2 上升沿触发使能位**<br>0 = 禁止PWM0_CH2上升沿触发.<br>1 = 使能PWM0_CH2 上升沿触发. |
| [3] | **PWM00CEN** | **PWM0_CH0中心触发使能位**<br>0 = 禁止PWM0_CH0中心触发<br>1 = 使能PWM0_CH0中心触发 |
| [2] | **PWM00PEN** | **PWM0_CH0周期触发使能位**<br>0 =禁止 PWM0_CH0 周期触发.<br>1 = 使能PWM0_CH0 周期触发. |

| 位 | 描述 | |
|---|---|---|
| [1] | **PWM00FEN** | **PWM0_CH0 下降沿触发使能位**<br>0 = 禁止PWM0_CH0下降沿触发.<br>1 =使能 PWM0_CH0下降沿触发. |
| [0] | **PWM00REN** | **PWM0_CH0 上升沿触发使能位**<br>0 = 禁止PWM0_CH0上升沿触发.<br>1 = 使能PWM0_CH0 上升沿触发. |

## 6.17 模拟比较器 (ACMP)

### 6.17.1 概述

The NuMicro® M0519系列包含三个比较器，当正极输入大于负极输入电压，比较器输出为逻辑1，反之为逻辑0.当比较器输出值改变时，每一路比较器都可通过配置产生中断。模块系统框图见图 6-141.

### 6.17.2 特性

- 模拟输入电压范围：0~ $AV_{DD}$
- 支持迟滞功能
- 支持唤醒功能
- 支持比较器输出反相功能
- 支持比较器的输出作为EPWM功能的刹车源功能
- ACMP0支持
  - 2路正极源：ACMP0_P 和 OP0_O
  - 2路负极源：ACMP0_N和内部带隙电压($V_{BG}$)
- ACMP1支持
  - 2路正极源：ACMP1_P 和 OP1_O
  - 2路负极源：ACMP1_N和内部带隙电压($V_{BG}$)
- ACMP2支持
  - 1路正极源：ACMP2_P
  - 2路负极源：ACMP2_N和内部带隙电压($V_{BG}$)
- 所有比较器共享1个中断向量

### 6.17.3 模块图

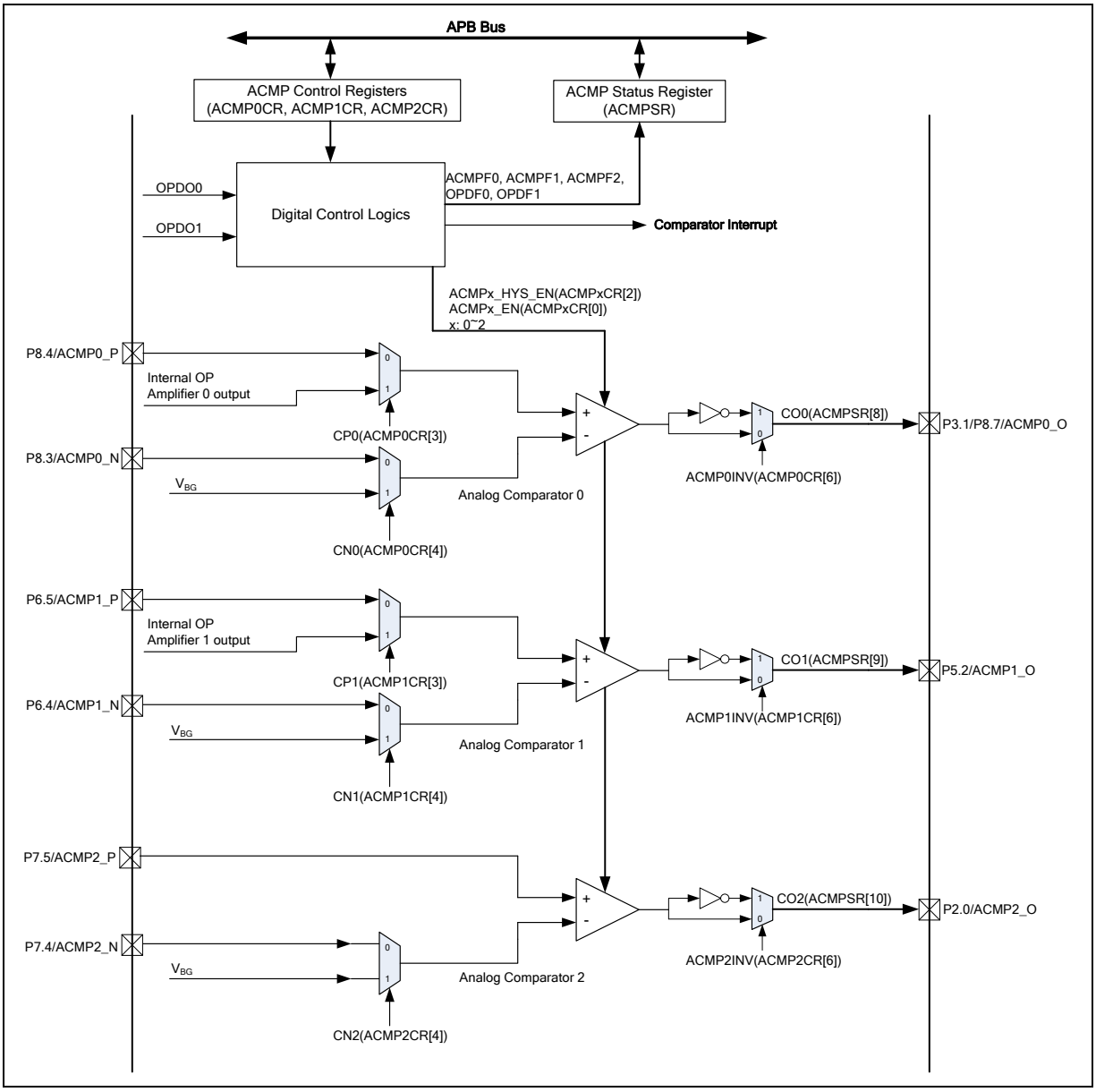



图 6-141 模拟比较器模块图

### 6.17.4 基本配置

ACMP时钟源是PCLK,可以通过设置ACMP EN(APBCLK[22])位来使能。通过设置P2_MFP，P3_MFP,P5_MFP, P6_MFP P7_MFP,P8 MFP寄存器来配置ACMP引脚功能。建议禁用模拟输入引脚的数字输入路径,避免漏电电流。数字输入路径可以通过配置P6 OFFD,P7 OFFD和P8 OFFD寄存器被禁用。

### 6.17.5 功能描述

#### *6.17.5.1 中断源*

比较器的输出采样时钟为PCLK时钟，通过CO0(ACMPSR[8]),CO1(ACMPSR[9]) 和CO2(ACMPSR[10])位反馈给系统。当ACMPIEx(ACMPxCR[1])置1时,比较器将启用中断。比较器的输出状态发生变化时,比较器中断将断言，同时相应的标志位,ACMPFx(ACMPSR[x]),将被设置。软件可以通过置1清除标志位。

当OP0和OP1放大器的施密特触发缓冲器改变输出端状态时,OPDF0(ACMPSR[4]),OPDF1(ACMPSR[5])中断标志将被自动置位。软件可以通过置1清除标志位。

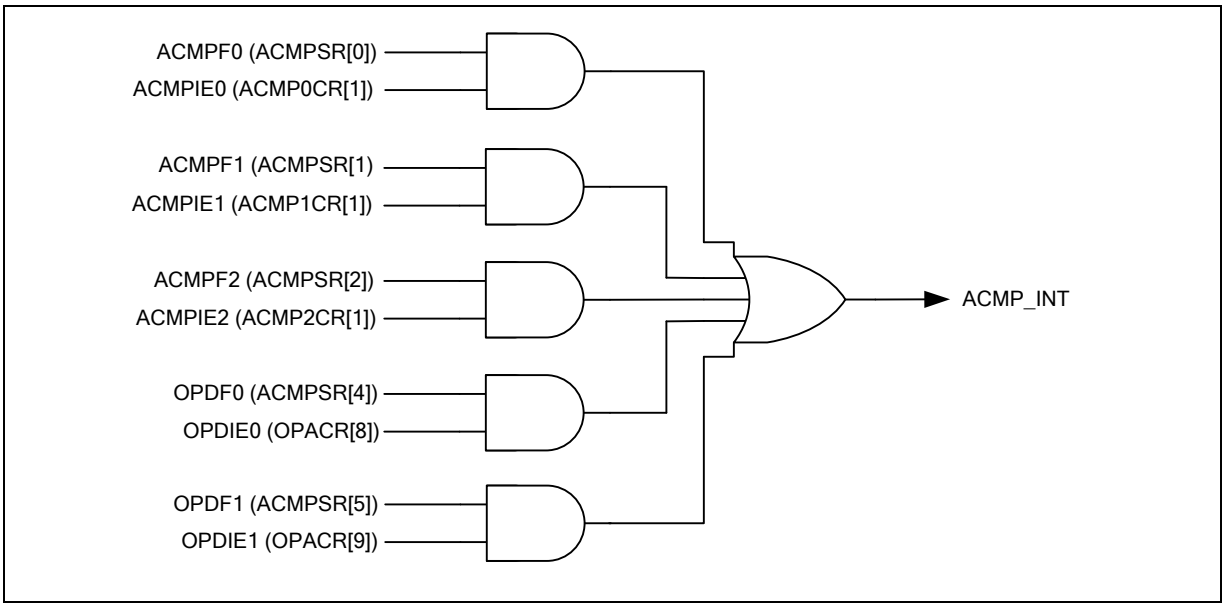



图 6-142 模拟比较器控制器中断

*6.17.5.2 迟滞功能*

.

模拟比较器提供了迟滞功能让比较器输出更加稳定。当比较器输出为0时，直到正极输入电压超过负极输入电压的差值达到阈值时，比较器的输出才会变化为1。同样，当比较器输出为1时，直到正极输入电压低过输入负极电压的差值达到阈值时，比较器的输出才会变化为1。

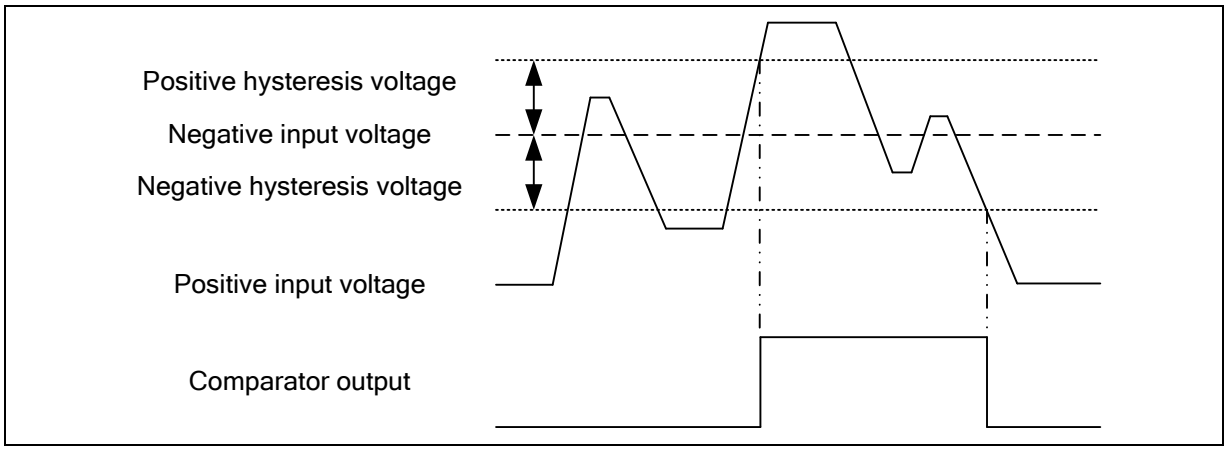



图 6-143 比较器迟滞功能

### 6.17.6 寄存器列表

**R**: 只读, **W**: 只写, **R/W**: 可读写, **C**:只可以写0

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ACMP基地址:**<br>**ACMP_BA = 0x400D_0000** | | | | |
| **ACMP0CR** | ACMP_BA+0x00 | R/W | 模拟比较器0通道控制寄存器 | 0x0000_0000 |
| **ACMP1CR** | ACMP_BA+0x04 | R/W | 模拟比较器1通道控制寄存器 | 0x0000_0000 |
| **ACMP2CR** | ACMP_BA+0x08 | R/W | 模拟比较器2通道控制寄存器 | 0x0000_0000 |
| **ACMPSR** | ACMP_BA+0x0C | R/W | 模拟比较器状态寄存器 | 0x0000_0000 |

### 6.17.7 寄存器描述

**模拟比较器 0控制寄存器(ACMP0CR)**

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ACMP0CR** | ACMP_BA+0x00 | R/W | 模拟比较器0通道控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| 保留 | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| 保留 | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **Reserved** | **ACMP0INV** | **Reserved** | **CN0** | **CP0** | **ACMP0_HYS_EN** | **ACMPIE0** | **ACMP0_EN** |

| 位 | 描述 | |
|---|---|---|
| [31:7] | | 保留 |
| [6] | **ACMP0INV** | **ACMP0通道输出反相功能使能位**<br>0=模拟比较器输出反相功能禁止<br>1=模拟比较器输出反相功能使能 |
| [5] | | 保留 |
| [4] | **CN0** | **ACMP0通道负极输入选择位**<br>0=引脚pin P8.3/ACMP0_N作为为比较器负极输入<br>1=内部带隙电压($V_{BG}$)作为比较器负极输入 |
| [3] | **CP0** | **ACMP0通道正极输入选择位**<br>0=引脚pin P8.4/ACMP0_P被作为比较器正极输入<br>1=内部 OP放大器的0通道输出作为比较器正极输入 |
| [2] | **ACMP0_HYS_EN** | **ACMP0迟滞功能使能位**<br>0=迟滞功能禁止<br>1=迟滞功能使能 |
| [1] | **ACMPIE0** | **ACMP0通道中断使能位**<br>0=中断功能禁止<br>1=中断功能使能 |
| [0] | **ACMP0_EN** | **ACMP0使能位**<br>0=模拟比较器禁止。<br>1=模拟比较器使能。 |

## 模拟比较器 1控制寄存器(ACMP1CR)

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ACMP1CR** | ACMP_BA+0x04 | R/W | 模拟比较器通道1控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| 保留 | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| 保留 | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 保留 | **ACMP1INV** | 保留 | **CN1** | **CP1** | **ACMP1_HYS_EN** | **ACMPIE1** | **ACMP1_EN** |

| 位 | 描述 | |
|---|---|---|
| [31:7] | | 保留 |
| [6] | **ACMP1INV** | **ACMP1通道输出反相功能使能位**<br>0=模拟比较器输出反相功能使能<br>1=模拟比较器输出反相功能使能 |
| [5] | | 保留 |
| [4] | **CN1** | **ACMP1通道负极输入选择位**<br>0=引脚pin P6.4/ACMP1_N作为比较器负极输入<br>1=内部带隙电压($V_{BG}$)作为比较器负极输入 |
| [3] | **CP1** | **ACMP1通道正极输入选择位**<br>0=引脚pin P6.5/ACMP1_P作为比较器负极输入<br>1=内部 OP放大器的通道1输出作为比较器正极输入 |
| [2] | **ACMP1_HYS_EN** | **ACMP1迟滞功能使能位**<br>0=迟滞功能禁止<br>1=迟滞功能使能 |
| [1] | **ACMPIE1** | **ACMP1通道中断使能位**<br>0=中断功能禁止<br>1=中断功能使能 |
| [0] | **ACMP1_EN** | **ACMP1通道使能位**<br>0=模拟比较器禁止。<br>1=模拟比较器使能。 |

## 模拟比较器2控制寄存器(ACMP2CR)

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ACMP2CR** | ACMP_BA+0x08 | R/W | 模拟比较器2通道控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| 保留 | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| 保留 | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 保留 | **ACMP2INV** | **Reserved** | **CN2** | 保留 | **ACMP2_HYS_EN** | **ACMPIE2** | **ACMP2_EN** |

| 位 | 描述 | |
|---|---|---|
| [31:7] | | 保留 |
| [6] | **ACMP2INV** | **ACMP2通道输出反相功能使能位**<br>0=模拟比较器输出反相功能使能<br>1=模拟比较器输出反相功能使能 |
| [5] | | 保留 |
| [4] | **CN2** | **ACMP2通道负极输入选择位**<br>0=引脚pin P7.4/ACMP2_N作为比较器负极输入<br>1=内部带隙电压($V_{BG}$)作为比较器负极输入 |
| [3] | **CP1** | **ACMP2通道正极输入选择位**<br>0=引脚pin P7.5/ACMP2_P作为比较器负极输入<br>1=内部 OP放大器的2通道输出作为比较器正极输入 |
| [2] | **ACMP2_HYS_EN** | **ACMP2迟滞功能使能位**<br>0=迟滞功能禁止<br>1=迟滞功能使能 |
| [1] | **ACMPIE2** | **ACMP2通道中断使能位**<br>0=中断功能禁止<br>1=中断功能使能 |
| [0] | **ACMP2_EN** | **ACMP2通道使能位**<br>0=模拟比较器禁止。<br>1=模拟比较器使能。 |

## 模拟比较器状态寄存器

| 寄存 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **ACMPSR** | ACMP_BA+0x0C | R/W | 模拟比较器状态寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| 保留 | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| 保留 | | | | | **CO2** | **CO1** | **CO0** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 保留 | | **OPDF1** | **OPDF0** | 保留 | **ACMPF2** | **ACMPF1** | **ACMPF0** |

| 位 | 描述 | |
|---|---|---|
| [31:11] | | 保留 |
| [10] | **CO2** | **比较器通道2输出**<br>由软件读取，同步时钟为PCLK.当比较器使能寄存器禁止(ACMP2_EN = 0)时,自动清除该位 |
| [9] | **CO1** | **比较器通道1输出**<br>由软件读取，同步时钟为PCLK.当比较器使能寄存器禁止(ACMP1_EN = 0)时,自动清除该位 |
| [8] | **CO0** | **比较器通道0输出**<br>由软件读取，同步时钟为PCLK.当比较器使能寄存器禁止(ACMP0_EN = 0)时,自动清除该位 |
| [7:6] | | 保留. |
| [5] | **OPDF1** | **运算放大器1施密特触发数字输出中断标志位**<br>当OP放大器通道1的施密特触发非-反相缓冲器改变了数字输出端状态时，OPDF1中断标志被硬件自动置位<br>**注意:**.该位从OPASR[5]重映射，写1到ACMPSR[5] 或 OPASR[5]寄存器可以清除该位 |
| [4] | **OPDF0** | **运算放大器0施密特触发数字输出中断标志**<br>当OP放大器通道0的施密特触发非-反相缓冲器改变了数字输出端状态时，OPDF0中断标志被硬件自动置位<br>**注:**该位从OPASR[4]重映射，写1到ACMPSR[4] 或 OPASR[4]寄存器可以清除该位 |
| [3] | | 保留 |
| [2] | **ACMPF2** | **比较器通道2标志位**.<br>当比较器通道2输出改变状态时，该位被自动配置。当ACMP2CR[1]位被置1时，将引起中断<br>写1清除该位 |

| | | |
|---|---|---|
| [1] | **ACMPF1** | **比较器通道1标志位**<br>当比较器通道1输出改变状态时，该位被自动配置。当ACMP1CR[1]位被置1时，将引起中断<br>写1清除该位 |
| [0] | **ACMPF0** | **比较器通道0标志位**<br>当比较器通道0输出改变状态时，该位被自动配置。当ACMP0CR[1]位被置1时，将引起中断<br>写1清除该位 |

## 6.18 运算放大器 (OPA)

### 6.18.1 概述.

该模块集成两个运算放大器。OP0_EN (OPACR[0]) 和OP1_EN (OPACR[1])的位是该模块的使能位。用户可以通过片内ADC测量运算放大器的输出。

### 6.18.2 特性

- 模拟输入电压范围：0~$AV_{DD}$
- 支持两路模拟运算放大器
- 支持通过ADC测量OP输出电压
- 支持施密特触发器缓冲输出和产生中断
- 运算放大器0通道输出可以选作内部比较器0通道的正极输入
- 运算放大器1通道输出可以选作内部比较器1通道的正极输入

### 6.18.3 模块图

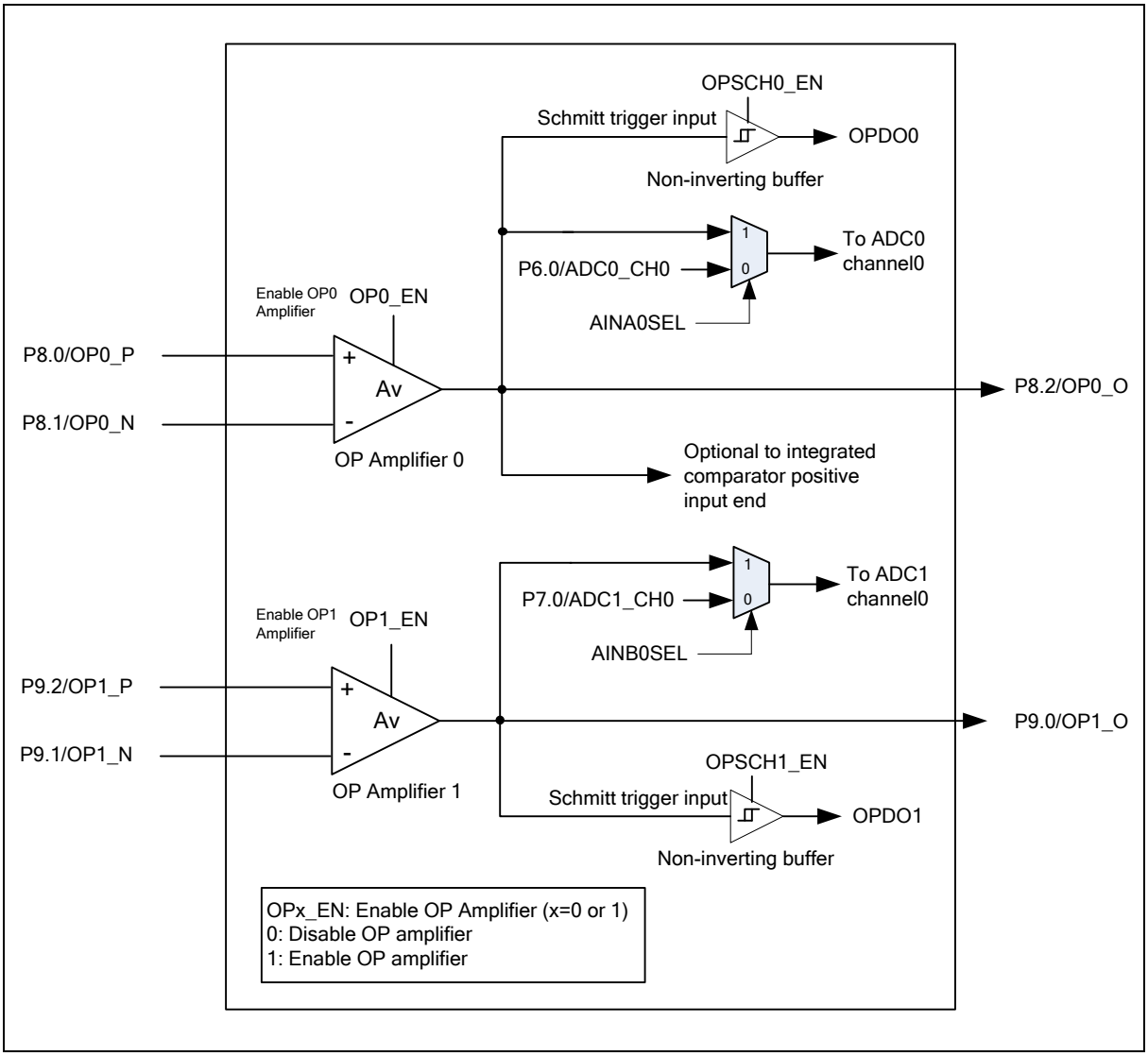



图 6-144 运算放大器模块图

### 6.18.4 基本配置

运算放大器的时钟源是 PCLK，由 OPA_EN (APBCLK[29]配置使能。引脚功能由 P8_MFP 和 P9_MFP registers 配置。建议禁用的模拟输入引脚的数字输入路径,避免漏电电流。通过配置 P8_OFFD 和 P9_OFFD 寄存器可以禁止数字输入路径。

### 6.18.5 中断源

当OP0和OP1放大器的施密特触发缓冲器改变输出端状态时,OPDF0(ACMPSR[4]),OPDF1(ACMPSR[5])中断标志将被自动配置。软件可以通过置1清除标志位

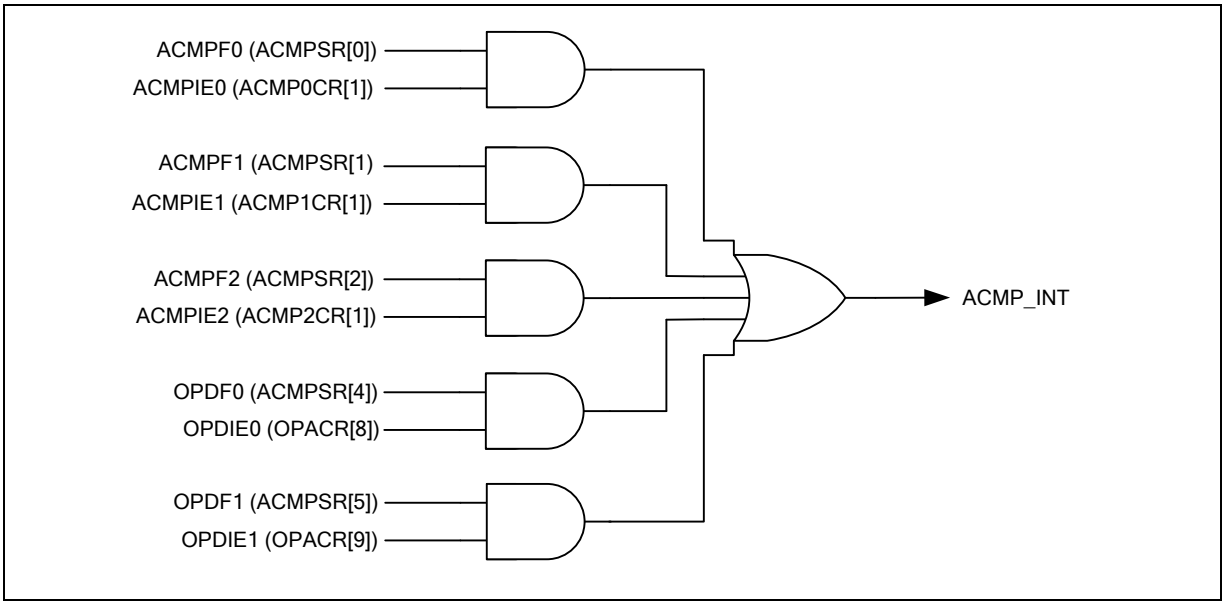



图 6-145 模拟比较器中断设计的运算放大器中断标志位

### 6.18.6 寄存器列表

**R**: 只读, **W**: 只写, **R/W**: 可读写, **C**:只可以写0

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **OPA 基地址:**<br>**OPA_BA = 0x400F_0000** | | | | |
| **OPACR** | OPA_BA+0x00 | R/W | 运算放大器控制寄存器 | 0x0000_0000 |
| **OPASR** | OPA_BA+0x04 | R/W | 运算放大器状态寄存器 | 0x0000_0000 |

### 6.18.7 寄存器描述

**OPA控制寄存器（OPACR）**

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **OPACR** | OPA_BA+0x00 | R/W | 运算放大器控制寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| 保留 | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| 保留 | | | | | | **OPDIE1** | **OPDIE0** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 保留 | | **OPSCH1_EN** | **OPSCH0_EN** | 保留 | | **OP1_EN** | **OP0_EN** |

| 位 | 描述 | |
|---|---|---|
| [31:10] | | 保留 |
| [9] | **OPDIE1** | **运算放大器通道1施密特触发数字输出中断使能位**<br>0=运算放大器通道1数字输出中断功能禁止<br>1=运算放大器通道1数字输出中断功能使能<br>当OP放大器通道1的施密特触发非-反相缓冲器改变了数字输出端状态时，OPDF1中断标志被硬件自动置位，如果此时，OPDIE1被置1，比较器中断请求将产生 |
| [8] | **OPDIE0** | **运算放大器通道0施密特触发数字输出中断使能位**<br>0=运算放大器通道0数字输出中断功能禁止<br>1=运算放大器通道0数字输出中断功能禁止<br>当OP放大器通道0的施密特触发非-反相缓冲器改变了数字输出端状态时，OPDF0中断标志被硬件自动置位，如果 此时，OPDIE0被置1，比较器中断请求将产生 |
| [7:6] | | 保留 |
| [5] | **OPSCH1_EN** | **运算放大器通道1施密特触发非 –反相缓冲器使能位**<br>0=运算放大器通道1施密特触发禁止<br>1=运算放大器通道1施密特触发使能 |
| [4] | **OPSCH0_EN** | **运算放大器通道0施密特触发非 –反相缓冲器使能位**<br>0=运算放大器通道0施密特触发禁止<br>1=运算放大器通道0施密特触发使能 |
| [3:2] | | 保留 |
| [1] | **OP1_EN** | **运算放大器通道1使能位**<br>0=运算放大器通道1禁止<br>1=运算放大器通道1使能 |

| 位 | 描述 | |
|---|---|---|
| [0] | **OP0_EN** | **运算放大器通道0使能位**<br>0=运算放大器通道0禁止<br>1=运算放大器通道0使能 |

## OPA状态寄存器

| 寄存器 | 偏移 | R/W | 描述 | 复位值 |
|---|---|---|---|---|
| **OPASR** | OPA_BA+0x04 | R/W | 运算放大器状态寄存器 | 0x0000_0000 |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| 保留 | | | | | | | |
| **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** |
| 保留 | | | | | | | |
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** |
| 保留 | | | | | | | |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 保留 | | **OPDF1** | **OPDF0** | 保留 | | **OPDO1** | **OPDO0** |

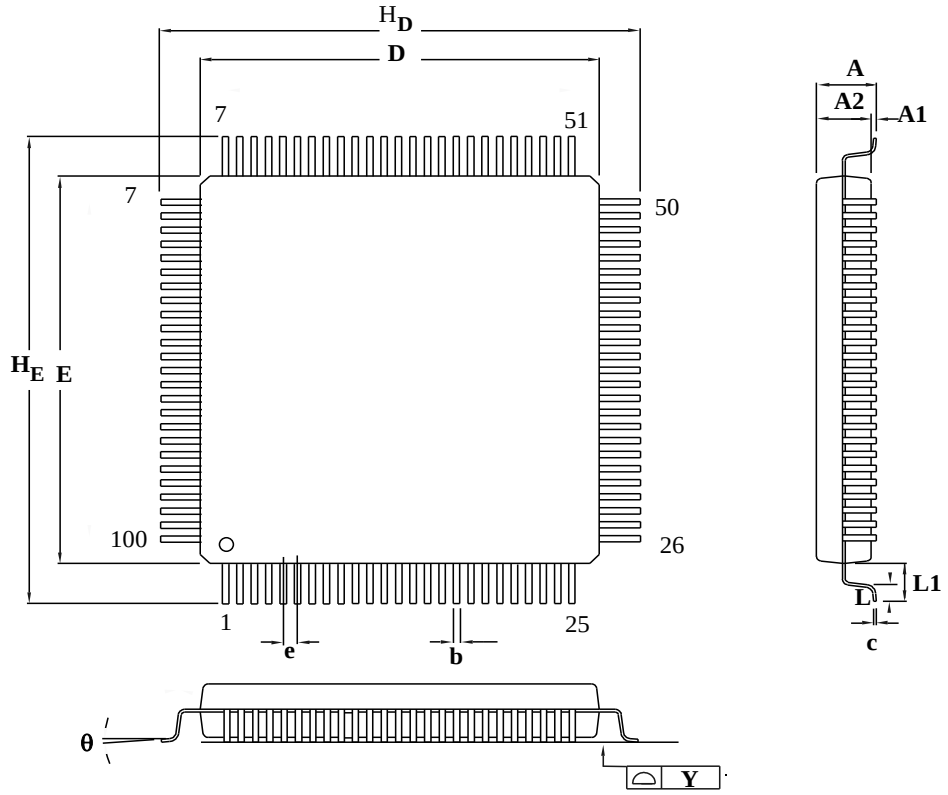
| 位 | 描述 | |
|---|---|---|
| [31:6] | | 保留 |
| [5] | **OPDF1** | **运算放大器通道1施密特触发数字输出中断标志位**<br>当OP放大器通道1的施密特触发非-反相缓冲器改变了数字输出端状态时，OPDF1中断标志被硬件自动置位<br>**注意:**该位写1清除 |
| [4] | **OPDF0** | **运算放大器通道0施密特触发数字输出中断标志位**<br>当OP放大器通道0的施密特触发非-反相缓冲器改变了数字输出端状态时，OPDF0中断标志被硬件自动置位<br>**注意:**该位写1清除 |
| [3:2] | | 保留 |
| [1] | **OPDO1** | **运算放大器通道1数字输出**<br>程序同步于APB时钟读取。当施密特触发缓冲器禁止时自动清0（OPSCH1_EN = 0）. |
| [0] | **OPDO0** | **运算放大器通道0数字输出**<br>程序同步于APB时钟读取。当施密特触发缓冲器禁止时自动清0 (OPSCH0_EN = 0). |

# 7 电气特性

有关 M0519 系列的电气特性, 请参阅 NuMicro® M0519 系列 Datasheet.

# 8 封装尺寸

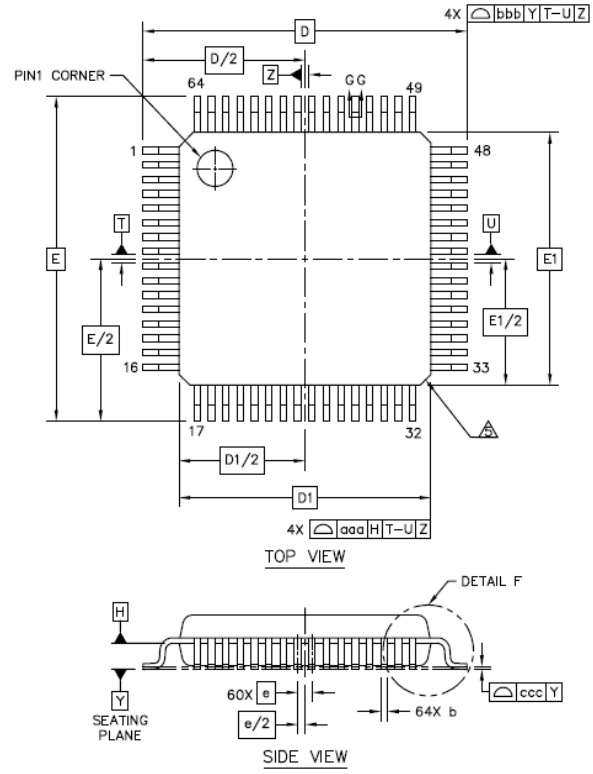
## 8.1 LQFP 100L (14x14x1.4 mm封装2.0mm)



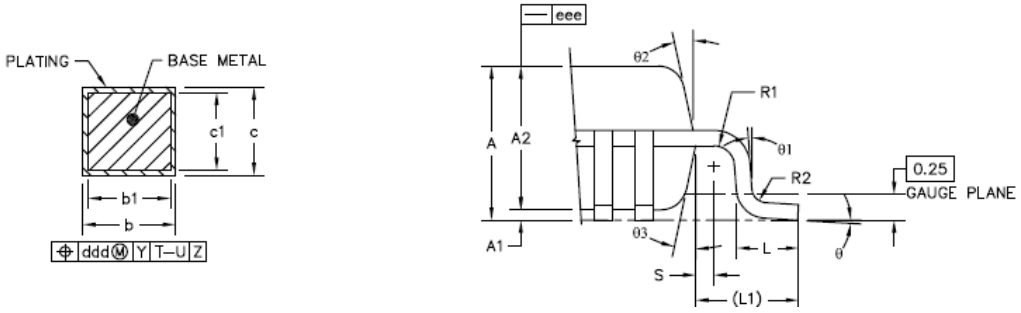

**Controlling Dimension : Millimeters**

| Symbol | Dimension in inch | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min | Nom | Max | Min | Nom | Max |
| A | — | — | 0.063 | — | — | 1.60 |
| A1 | 0.002 | — | — | 0.05 | — | — |
| A | 0.053 | 0.055 | 0.057 | 1.35 | 1.40 | 1.45 |
| b | 0.007 | 0.009 | 0.011 | 0.17 | 0.22 | 0.27 |
| c | 0.004 | 0.006 | 0.008 | 0.10 | 0.15 | 0.20 |
| D | 0.547 | 0.551 | 0.556 | 13.90 | 14.00 | 14.10 |
| E | 0.547 | 0.551 | 0.556 | 13.90 | 14.00 | 14.10 |
| e | — | 0.020 | — | — | 0.50 | — |
| $H_D$ | 0.622 | 0.630 | 0.638 | 15.80 | 16.00 | 16.20 |
| $H_E$ | 0.622 | 0.630 | 0.638 | 15.80 | 16.00 | 16.20 |
| L | 0.018 | 0.024 | 0.030 | 0.45 | 0.60 | 0.75 |
| L1 | — | 0.039 | — | — | 1.00 | — |
| y | — | — | 0.004 | — | — | 0.10 |
| θ | 0° | — | 7° | 0° | — | 7° |

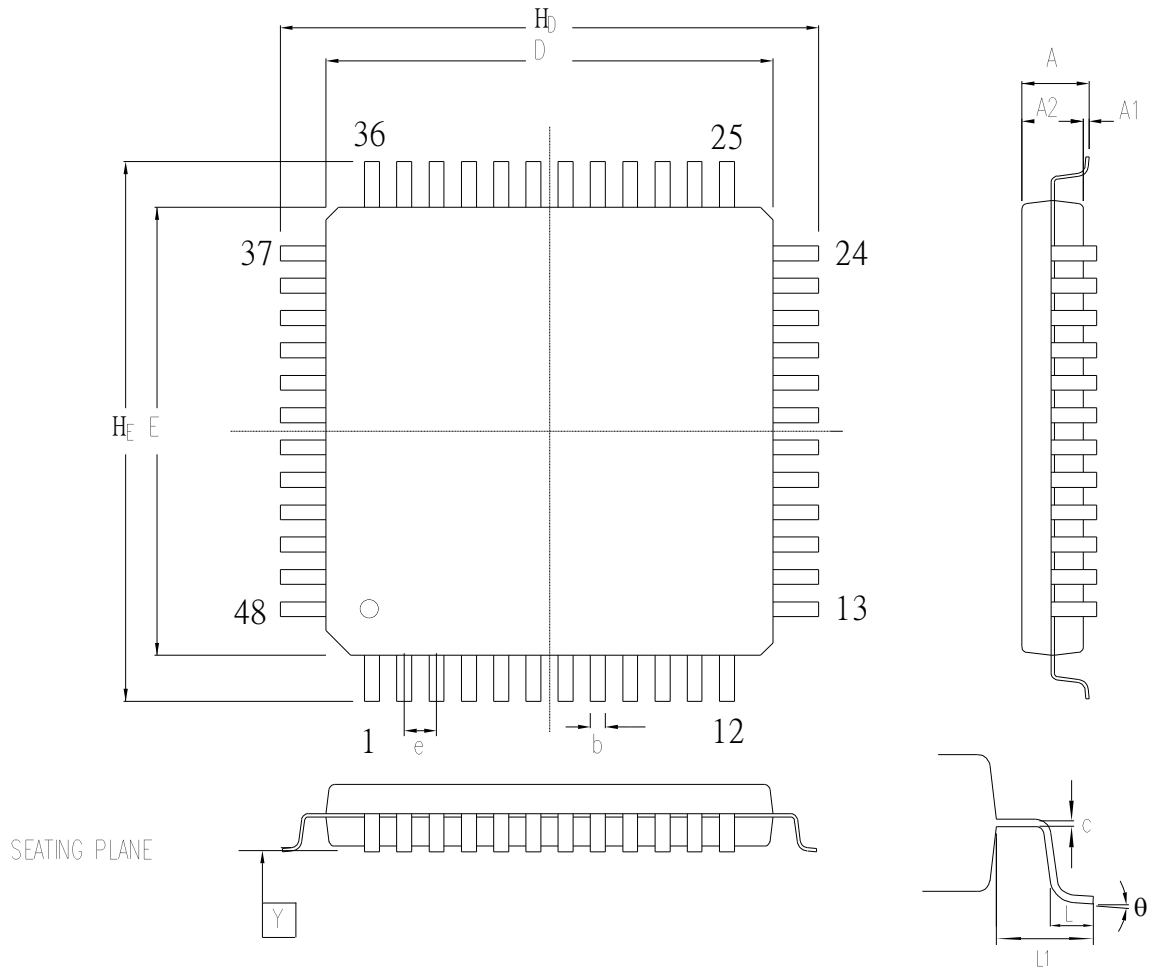
## 8.2 LQFP 64L (7x7x1.4 mm 封装 2.0 mm)




| | | SYMBOL | MIN | NOM | MAX |
|---|---|---|---|---|---|
| TOTAL THICKNESS | | A | --- | --- | 1.6 |
| STAND OFF | | A1 | 0.05 | --- | 0.15 |
| MOLD THICKNESS | | A2 | 1.35 | 1.4 | 1.45 |
| LEAD WIDTH(PLATING) | | b | 0.13 | 0.18 | 0.23 |
| LEAD WIDTH | | b1 | 0.13 | 0.16 | 0.19 |
| L/F THICKNESS(PLATING) | | c | 0.09 | --- | 0.2 |
| L/F THICKNESS | | c1 | 0.09 | --- | 0.16 |
| BODY SIZE | X | D | 9 BSC | | |
| | Y | E | 9 BSC | | |
| | X | D1 | 7 BSC | | |
| | Y | E1 | 7 BSC | | |
| LEAD PITCH | | e | 0.4 BSC | | |
| | | L | 0.45 | 0.6 | 0.75 |
| FOOTPRINT | | L1 | 1 REF | | |
| | | θ | 0° | 3.5° | 7° |
| | | θ1 | 0° | --- | --- |
| | | θ2 | 11° | 12° | 13° |
| | | θ3 | 11° | 12° | 13° |
| | | R1 | 0.08 | --- | --- |
| | | R2 | 0.08 | --- | 0.2 |
| | | S | 0.2 | --- | --- |
| PACKAGE EDGE TOLERANCE | | aaa | 0.2 | | |
| LEAD EDGE TOLERANCE | | bbb | 0.2 | | |
| COPLANARITY | | ccc | 0.08 | | |
| LEAD OFFSET | | ddd | 0.07 | | |
| MOLD FLATNESS | | eee | 0.05 | | |



## 8.3 LQFP 48L (7x7x1.4mm封装2.0mm)




Controlling dimension : Millimeters

| Symbol | Dimension in inch | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min | Nom | Max | Min | Nom | Max |
| A | — | — | — | — | — | — |
| $A_1$ | 0.002 | 0.004 | 0.006 | 0.05 | 0.10 | 0.15 |
| $A_2$ | 0.053 | 0.055 | 0.057 | 1.35 | 1.40 | 1.45 |
| b | 0.006 | 0.008 | 0.010 | 0.15 | 0.20 | 0.25 |
| c | 0.004 | 0.006 | 0.008 | 0.10 | 0.15 | 0.20 |
| D | 0.272 | 0.276 | 0.280 | 6.90 | 7.00 | 7.10 |
| E | 0.272 | 0.276 | 0.280 | 6.90 | 7.00 | 7.10 |
| e | 0.014 | 0.020 | 0.026 | 0.35 | 0.50 | 0.65 |
| $H_D$ | 0.350 | 0.354 | 0.358 | 8.90 | 9.00 | 9.10 |
| $H_E$ | 0.350 | 0.354 | 0.358 | 8.90 | 9.00 | 9.10 |
| L | 0.018 | 0.024 | 0.030 | 0.45 | 0.60 | 0.75 |
| $L_1$ | — | 0.039 | — | — | 1.00 | — |
| Y | — | — | 0.004 | — | — | 0.10 |
| θ | 0 | — | 7 | 0 | — | 7 |

# 9 修改历史

| 日期 | 版本 | 描述 |
|---|---|---|
| 2015.03.30 | 1.00 | 1. 初版. |

## Important Notice

**Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".**

**Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.**

**All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.**

*Please note that all data and specifications are subject to change without notice.*
*All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*