

ARM Cortex[®]-M0

32-位 微控制器

NuMicro[®] 家族

NM1120 系列

技术参考手册

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

目录

1 概述	14
2 特性	15
3 缩写表	19
4 器件信息和管脚定义	20
4.1 NuMicro® NM1120 命名规则	20
4.2 NuMicro® NM1120 系列产品选型指南	21
4.3 管脚配置	22
4.3.1 TSSOP 28-Pin	22
4.3.2 TSSOP 20-Pin	23
4.3.3 QFN 33-Pin	24
4.3.4 QFN 20-Pin	25
4.4 管脚描述	26
4.4.1 NM1120 系列管脚描述	26
4.4.2 GPIO 多功能管脚列表	38
5 方框图	41
5.1 NuMicro® NM1120 方框图	41
6 功能描述	42
6.1 ARM® Cortex®-M0 内核	42
6.1.1 概述	42
6.1.2 特性	42
6.2 系统管理	44
6.2.1 概述	44
6.2.2 系统复位	44
6.2.3 电源模式和唤醒源	50
6.2.4 系统电源结构	53
6.2.5 系统内存映射	54
6.2.6 寄存器保护	56
6.2.7 内存组织	58
6.2.8 寄存器映射	61
6.2.9 寄存器描述	62
6.2.10 系统定时器 (SysTick)	82
6.2.11 嵌套向量中断控制器 (NVIC)	87
6.2.12 系统控制寄存器	107
6.3 时钟控制器	116
6.3.1 概述	116
6.3.2 自动校准	118
6.3.3 系统时钟和 SysTick 时钟	118
6.3.4 外设时钟源选择	119

6.3.5	掉电模式时钟	121
6.3.6	分频器输出	121
6.3.7	寄存器映射	123
6.3.8	寄存器描述	124
6.4	存储控制器 (FMC)	136
6.4.1	概述	136
6.4.2	特性	136
6.4.3	框图	136
6.4.4	功能描述	137
6.4.5	寄存器映射	152
6.4.6	寄存器描述	153
6.5	通用I/O (GPIO)	164
6.5.1	概述	164
6.5.2	特性	164
6.5.3	框图	165
6.5.4	基本配置	165
6.5.5	功能描述	166
6.5.6	GPIO 中断和唤醒功能	168
6.5.7	寄存器映射	169
6.5.8	寄存器描述	172
6.6	定时器控制器(TIMER)	189
6.6.1	概述	189
6.6.2	特性	189
6.6.3	框图	190
6.6.4	基本配置	191
6.6.5	功能描述	191
6.6.6	寄存器映射	197
6.6.7	寄存器描述	198
6.7	增强型输入捕捉定时器 (ECAP)	210
6.7.1	概述	210
6.7.2	特性	210
6.7.3	框图	210
6.7.4	输入噪音滤波器	211
6.7.5	输入捕捉定时器/计数器操作	212
6.7.6	输入捕捉定时器/计数器中断架构	214
6.7.7	寄存器映射	215
6.7.8	寄存器描述	216
6.8	增强型PWM 发生器(EPWM)	226
6.8.1	概述	226
6.8.2	特性	226
6.8.3	框图	227
6.8.4	基本配置	229

6.8.5	功能描述	229
6.8.6	寄存器映射	248
6.8.7	寄存器描述	249
6.9	基本PWM 发生器 (BPWM)	277
6.9.1	概述	277
6.9.2	特性	277
6.9.3	框图	277
6.9.4	PWM 计数器操作	278
6.9.5	寄存器映射	286
6.9.6	寄存器描述	287
6.10	看门狗定时器 (WDT)	297
6.10.1	概述	297
6.10.2	特征	297
6.10.3	框图	297
6.10.4	时钟控制	298
6.10.5	基本配置	298
6.10.6	功能描述	298
6.10.7	寄存器映射	300
6.10.8	寄存器描述	301
6.11	USCI – 通用串行控制接口控制器	303
6.11.1	概述	303
6.11.2	特征	303
6.11.3	框图	303
6.11.4	功能描述	303
6.12	USCI – UART 模式	314
6.12.1	概述	314
6.12.2	特性	314
6.12.3	框图	314
6.12.4	基本配置	315
6.12.5	功能描述	315
6.12.6	寄存器映射	324
6.12.7	寄存器描述	325
6.13	USCI – SPI 模式	344
6.13.1	概述	344
6.13.2	特性	344
6.13.3	框图	345
6.13.4	基本配置	345
6.13.5	功能描述	346
6.13.6	寄存器映射	358
6.13.7	寄存器描述	359
6.14	USCI – I ² C 模式	381

6.14.1	概述	381
6.14.2	特征	381
6.14.3	框图	381
6.14.4	基本配置	382
6.14.5	功能描述	382
6.14.6	寄存器映射	398
6.14.7	寄存器描述	399
6.15	硬件除法器 (HDIV)	417
6.15.1	概述	417
6.15.2	特征	417
6.15.3	基本配置	417
6.15.4	功能描述	418
6.15.5	寄存器映射	419
6.15.6	寄存器描述	420
6.16	模数转换器 (ADC)	425
1.1.1	概述	425
1.1.2	特性	425
1.1.3	方框图.....	425
6.16.1	基本配置	426
6.16.2	功能描述	426
6.16.3	寄存器映射	430
6.16.4	寄存器描述	431
6.17	模拟比较器 (ACMP).....	447
6.17.1	概述	447
6.17.2	特性	447
6.17.3	方框图.....	448
6.17.4	基本配置	448
6.17.5	功能描述	449
6.17.6	比较器参考电压 (CRV)	450
6.17.7	寄存器映射	451
6.17.8	寄存器描述	452
6.18	可编程增益放大器 (PGA).....	461
6.18.1	概述	461
6.18.2	特性	461
6.18.3	方框图.....	461
6.18.4	寄存器映射	461
6.18.5	寄存器 描述	462
7	应用电路	463
8	电气特性	464
8.1	绝对最大额定值	464
8.2	DC 电气特性	465

8.3	AC 电气特性	470
8.3.1	外部高速晶振	470
8.3.2	外部 4~24 MHz 高速晶振 (HXT)	470
8.3.3	外部 32.768 kHz XTAL 晶振 (LXT)	470
8.3.4	外部晶振的典型应用电路	470
8.3.5	48 MHz 内部高速 RC 振荡器 (HIRC)	471
8.3.6	内部低速 10 kHz RC 振荡器 (LIRC)	471
8.4	模拟量特性	472
8.4.1	12-bit SAR ADC	472
8.4.2	LDO & 电源管理	473
8.4.3	低电压复位	473
8.4.4	欠压检测	474
8.4.5	上电复位	474
8.4.6	比较器规格	475
8.4.7	PGA	475
8.5	ESD 特性	477
8.6	EFT 特性	477
8.7	Flash DC 电气特性	478
9	封装尺寸	479
9.1	28-Pin TSSOP (4.4x9.7x1.0 mm)	479
9.2	20-Pin TSSOP (4.4x6.5x0.9 mm)	480
9.3	20-pin *QFN20 (4 mm x 4 mm)	481
9.4	33-pin QFN33 (4x4x0.8 mm)	482
10	修订历史	483

图集

图 4.1-1 NuMicro® NM1120 系列产品命名规则.....	20
图 5.1-1 NuMicro® NM1120 方框图.....	41
图 6.1-1 功能框图.....	42
图 6.2-1 系统复位源.....	45
图 6.2-2 nRESET 复位波形.....	47
图 6.2-3 上电复位 (POR) 波形.....	47
图 6.2-4 低电复位 (LVR) 波形.....	48
图 6.2-5 欠压检测 (BOD) 复位波形.....	49
图 6.2-6 电源模式状态机.....	50
图 6.2-7 NuMicro® NM1120 系列电源架构图.....	53
图 6.2-8 NuMicro® NM1120 Flash, Security and Configuration 映射图.....	58
图 6.2-9 SRAM 方框图.....	60
图 6.3-1 时钟发生器框图.....	116
图 6.3-2 时钟发生器全局框图.....	117
图 6.3-3 系统时钟框图.....	118
图 6.3-4 SysTick 时钟控制框图.....	119
图 6.3-5 PCLK外设总线时钟源选择.....	120
图 6.3-6 分频器的时钟源框图.....	122
图 6.3-7 分频器框图.....	122
图 6.4-1 存储控制器框图.....	136
图 6.4-2 Flash 存储器组织.....	138
图 6.4-3 数据 Flash 与APROM共享存储空间.....	139
图 6.4-4 SPROM 安全锁模式.....	141
图 6.4-5 上电后启动项 (BS) 示意图.....	146
图 6.4-6 Flash 存储器的映射通过设置CONFIG0 (CBS).....	147
图 6.4-7 使能IAP功能代码可执行区域.....	148
图 6.4-8 当CBS[0] = 1时启动选择的流程例图.....	149
图 6.4-9 ISP 流程例图.....	150
图 6.5-1 I/O 引脚方框图.....	164
图 6.5-2 GPIO 控制器框图.....	165
图 6.5-3 推挽输出.....	166
图 6.5-4 开漏输出.....	167

图 6.5-5 准双向I/O模式.....	167
图 6.6-1 定时器控制器框图.....	190
图 6.6-2 定时器控制器时钟源	190
图 6.6-3 连续计数模式	192
图 6.6-4 Free-Counting 捕捉 模式	193
表 6.6-5 外部复位计数模式.....	194
图 6.6-6 连续捕捉模式框图.....	195
图 6.6-7 连续捕捉模式时序.....	196
图 6.7-1 增强型输入捕捉定时器/计数器框图	210
图 6.7-2 增强型输入捕捉定时器/计数器时钟源控制	211
图 6.7-3 噪音过滤采样时钟选项	211
图 6.7-4 噪音滤波器.....	211
图 6.7-5 增强型输入捕捉定时器/计数器功能方框图	213
图 6.7-6 增强型输入捕捉定时器/计数器中断架构图	214
图 6.8-1 EPWM 时钟源	227
图 6.8-2 EPWM 方框图	227
图 6.8-3 EPWM 发生器 CH0/1 架构图	228
图 6.8-4 EPWM发生器 CH2/3 架构图	228
图 6.8-5 EPWM 发生器 CH4/5 架构图	229
图 6.8-6 EPWM 边沿对齐波形输出	230
图 6.8-7 EPWM 边沿对齐模式操作时序	231
图 6.8-8 EPWM 边沿对齐中断图	232
图 6.8-9 EPWM 边沿对齐流程图	233
图 6.8-10 EPWM 比较器输出说明	234
图 6.8-11 EPWM-Timer 工作时序	234
图 6.8-12 EPWM 中心对齐模式.....	235
图 6.8-13 EPWM 中心对齐模式操作时序	236
图 6.8-14 EPWM 中心对齐模式波形输出	237
图 6.8-15 EPWM 中心对齐模式中断图.....	237
图 6.8-16 EPWM 中心对齐模式流程图.....	238
图 6.8-17 EPWM 插入死区时间.....	240
图 6.8-18 EPWM 不对称模式时序图	241
图 6.8-19 EPWM One-Shot 模式架构	241

图 6.8-20 上升沿死区EPWM极性控制初始化状态	242
图 6.8-21 EPWM中断架构	243
图 6.8-22 EPWM 刹车架构	244
图 6.8-23 EPWM 3-相位监测屏蔽图	245
图 6.8-24 EPWM 3-相位马达屏蔽例1	246
图 6.8-25 EPWM 3-相位马达屏蔽例2	246
图 6.9-1 PWM 时钟源控制	278
图 6.9-2 PWM 结构图	278
图 6.9-3 PWM内部比较计数器说明	279
图 6.9-4 PWM计数器操作时序	280
图 6.9-5 PWM 边沿对齐中断产生的时序波形	280
图 6.9-6 中心对齐类型的输出波形	281
图 6.9-7 PWM 中心对齐中断产生的时序波形	282
图 6.9-8 PWM 双缓存说明	283
图 6.9-9 PWM 控制器输入占空比率	283
图 6.9-10 一对带死区插入功能的PWM输出	284
图 6.9-11 PWM 中断架构框图	284
图 6.10-1 看门狗定时器框图	297
图 6.10-2 看门狗定时器时钟控制框图	298
图 6.10-3 看门狗定时器超时间隔和复位周期时序	299
图 6.11-1 USCI 框图	303
图 6.11-2 USCIx_DAT[1:0] 和 USCIx_CTL[1:0]输入架构	305
图 6.11-3 USCIx_CLK输入架构	305
图 6.11-4 数据缓存框图	306
图 6.11-5 数据存取架构	307
图 6.11-6 发送数据路径	307
图 6.11-7 接收数据路径	308
图 6.11-8 协议时钟产生器	309
图 6.11-9 基本时钟除频计数器	310
图 6.11-10 时序测量计数器框图	310
图 6.11-11 采样时间计数器	311
图 6.11-12 事件和中断架构	312
图 6.12-1 USCI - UART 模式框图	314

图 6.12-2 UART 全双工通信连接	315
图 6.12-3 UART 标准帧格式.....	316
图 6.12-4 UART 位时序 (数据采样个数).....	318
图 6.12-5 UART 自动波特率控制	320
图 6.12-6 接收收据唤醒	321
图 6.13-1 SPI 主机模式应用框图 (x=0, 1)	344
图 6.13-2 SPI 从机模式应用框图 (x=0, 1)	344
图 6.13-3 USCI - SPI 模式框图	345
图 6.13-4 4线全双工 SPI 通讯信号 (主机模式).....	346
图 6.13-5 4线全双工SPI 通讯信号 (从机模式).....	347
图 6.13-6 SPI 通讯时钟配置 (SCLKMODE=0x0)	348
图 6.13-7 SPI 通讯时钟配置(SCLKMODE=0x1)	348
图 6.13-8 SPI 通讯时钟配置 (SCLKMODE=0x2)	349
图 6.13-9 SPI通讯时钟配置 (SCLKMODE=0x3)	349
图 6.13-10 MSB优先的16位字长传输.....	350
图 6.13-11两字传输间的暂停时间间隔	351
图 6.13-12 自动从机选择 (SUSPITV \geq 0x3)	351
图 6.13-13 自动从机选择 (SUSPITV < 0x3)	352
图 6.13-14 SPI 主机模式时序	354
图 6.13-15 SPI 主机模式时序 (时钟信号相位可选).....	354
图 6.13-16 SPI 从机模式时序	355
图 6.13-17 SPI 从机模式时序 (时钟信号相位可选).....	355
图 6.14-1 I ² C 总线时序	381
图 6.14-2 USCI - I ² C 模式框图	382
图 6.14-3 I ² C 协议	383
图 6.14-4 开始、重复开始和停止信号波形	383
图 6.14-5 I ² C 总线位传输.....	384
图 6.14-6 I ² C 总线应答	385
图 6.14-7 仲裁丢失	386
图 6.14-8 根据当前I ² C 状态控制I ² C总线	388
图 6.14-9主机通过7位地址发送数据到从机	389
图 6.14-10主机通过7位地址读从机数据	389
图 6.14-11主机通过10位地址发送数据到从机.....	389

图 6.14-12 主机通过10位地址读从机数据	390
图 6.14-13 7位地址主机传输模式控制流程	390
图 6.14-14 主机7位地址接收模式控制流程	391
图 6.14-15 从机7位地址模式控制流程	392
图 6.14-16 7位地址广播模式	393
图 6.14-17 设置时间错误	395
图 6.14-18 保持时间错误	395
图 6.14-19 I ² C 超时计数框图	395
图 6.14-20 EEPROM 随机读	396
图 6.14-21 EEPROM的协议随机读	397
图 6.15-1 硬件除法器操作流程	418
图 6.16-1 ADC控制方框图	425
图 6.16-2 ADC外设时钟控制	426
图 6.16-3 单周期模式转换时序框图	427
图 6.16-4 ADC硬件触发源	428
图 6.16-5 独立采样模式转换时序图	428
图 6.16-6 同时采样模式转换采样时序图	429
图 6.16-7 同时连续4R模式转换时序图	429
图 6.17-1 模拟比较器方框图	448
图 6.17-2 模拟比较器控制器中断源	449
图 6.17-3 比较器迟滞功能	449
图 6.17-4 比较器参考电压方框图	450
图 6.18-1 OP放大器方框图	461
图 8.3-1 NM1120 典型晶振应用电路	471

表集

表 3-1 缩写词列表.....	19
表 4.2-1 NuMicro® NM1120 系列产品选型指南.....	21
表 4.4-1 TSSOP28 管脚描述.....	29
表 4.4-2 TSSOP20 管脚描述.....	33
表 4.4-3 QFN33 管脚描述	37
表 4.4-4 TSSOP20 多功能管脚列表	40
表 6.2-1 寄存器复位值.....	46
表 6.2-2 电源模式差异表	50
表 6.2-3 不同电源模式下的时钟情况	51
表 6.2-4 外设再次进入掉电模式的条件.....	52
表 6.2-5 内存映射表.....	54
表 6.2-6 保护寄存器.....	57
表 6.2-7 片上模块地址空间分配	59
表 6.2-8 异常模式	88
表 6.2-9 系统中断映射向量表	89
表 6.2-10 向量表格式	89
表 6.3-1 外设时钟源选择表.....	121
表 6.3-2 掉电模式控制表	126
表 6.4-1 Flash存储器的地址映射	137
表 6.4-2 数据Flash 表格	138
表 6.4-3 启动选项	146
表格 6.4-4 启动选择和支持功能	147
表 6.4-5 ISP 命令表.....	151
表 6.6-1 输入捕捉模式操作流程	195
表 6.10-1 看门狗定时器超时时间间隔周期选择	299
表 6.11-1 不同协议的输入信号	304
表 6.11-2 不同协议的输出信号	306
表 6.11-3 数据发送事件和中断处理.....	312
表 6.11-4 协议事件及中断处理	313
表 6.12-1 UART 协议输入信号	316
表 6.12-2 不同协议的输出信号	316
表 6.12-3 波特率与晶振的关系	319

表 6.13-1 SPI 通讯信号 (x=0, 1) 346

表 6.13-2 串行总线时钟配置..... 347

表 6.14-1 I²C 波特率与PCLK之间的关系..... 394

1 概述

NuMicro® NM1120 系列32位微控制器内嵌ARM® Cortex®-M0内核，适用于追求高性能、高集成度、低成本的工业应用。Cortex®-M0是最新的ARM®32位嵌入式处理器，它拥有32位处理器的性能，与传统8位机相匹敌的价格。

NM1120系列最高运行的频率可达48MHz，工作电压为2.1V ~ 5.5V，工作温度-40℃ ~ 105℃，适用于各种需要高CPU性能的工业控制产品。NM1120内嵌29.5KB的程序存储器，大小可配置的数据存储器（共享程序存储器空间），同时拥有2KB的ISP存储器和1.5KB用于安全加密的SPROM以及4KB的SRAM。

集成丰富的系统级外设，如I/O口、定时器、串口、SPI、I2C、PWM、ADC、看门狗定时器、模拟比较器、欠压检测器，这些都被集成到NM1120内部以减少外部元件数量、节省电路板空间从而降低产品成本。这些丰富的外设使NM1120适用于各种不同的应用。

此外，NM1120 系列还支持ISP（在系统编程）和ICP（在线电路编程）功能，允许用户不用将芯片从实际产品上取下来就可以更新程序。

2 特性

- 内核
 - ARM® Cortex®-M0 内核，主频最高可运行到48MHz
 - 一个 24位系统定时器
 - 支持低功耗空闲模式
 - 单指令周期32位硬件乘法器
 - 32个可嵌套NVIC中断源，每个中断有4个优先级
 - 支持串行调试接口（SWD），支持2个观察点/4个断点
- 集成LDO，支持宽工作电压：2.1V to 5.5V
- 存储器
 - 29.5 KB 程序代码存储器(APROM)
 - 可将程序存储器配置成数据存储器使用(Data Flash)
 - 2 KB启动代码存储器(LDROM)
 - 3个 0.5 KB安全加密存储器(SPROM)
 - 4 KB 高速静态随机存储器 (SRAM)
- 时钟控制
 - 可编程系统时钟源
 - ◆ 可通过软件实时切换系统时钟源
 - 支持外部4 ~ 24 MHz高速晶振输入(HXT)
 - 支持外部32.768 kHz低速晶振(LXT)可用做空闲模式唤醒和系统运行时钟源
 - 内置48 MHz 高速RC振荡器(HIRC) (在25⁰C、5V环境下，精度为±1%)
 - ◆ 可通过外部32.768K晶体振荡器(LXT)动态校准HIRC，使其在-40℃到105℃环境范围内精度可达到48MHz ±1%
 - 内置10 kHz 低速RC振荡器(LIRC)用于看门狗定时器和空闲唤醒操作
- I/O 口
 - QFN-33封装的型号有多达22个通用I/O口(GPIO)和1个复位引脚
 - 四种 I/O 模式:
 - ◆ 准双向输入/输出
 - ◆ 推挽输出
 - ◆ 漏极开路输出
 - ◆ 高阻态输入
 - 可配置成施密特触发输入
 - I/O口可配置成边沿/电平触发的中断源

- I/O口支持高驱动输出和高灌电流输入模式
- 支持软件可选的转换速率控制
- I/O口内置上拉和下拉电阻
- 定时器
 - 提供两通道的32位定时器，包含一个8位预分频器和一个24位向上计数器
 - 每个定时器有独立的时钟源
 - 支持单次、周期、触发、连续的工作模式
 - 24位的向上计数值可通过CNT（定时计数寄存器）来读取
 - 支持通过外部捕捉脚或内部比较器来触发定时计数器复位的功能
 - 支持事件计数功能
 - 支持触发输出模式
 - 支持从空闲或休眠模式下唤醒功能
- 连续捕捉
 - 定时器0和1以及系统滴答定时器支持在一个信号上最多连续捕捉4个边沿
- 增强型输入捕捉
 - 一个24位输入捕捉计数器
 - 捕捉源:
 - ◆ I/O 口输入: ECAP0, ECAP1 和 ECAP2
 - ◆ PWM 触发
 - ◆ ADC 触发
- WDT (看门狗定时器)
 - 可编程选择时钟源和设定超时周期
 - 支持从掉电或空闲模式下唤醒的功能
 - 可选择看门狗超时后产生中断或是系统复位
- PWM
 - 内置一个16位的PWM时钟发生器，提供6路单相PWM输出或3组互补PWM输出
 - 共用一个时钟源、分频器、周期寄存器和死区发生器
 - 支持分组/同步/独立/互补模式
 - 支持单次或自动加载模式
 - 支持边沿对其和中心对其模式
 - 支持非对称模式
 - 支持互补通道之间的可编程死区嵌入功能
 - 可单独设置每个输出通道的极性

- 支持硬件和软件出错停止保护功能
- 支持上升、下降、中心、周期和出错停止时触发中断
- 支持 duty/period 触发 A/D 转换功能
- 支持定时器比较匹配事件触发PWM改变相位功能
- 支持比较器事件触发PWM在当前的周期输出低电平
- 支持中断累加计数功能
- USCI (通用串行接口控制器)
 - 2个 USCI 模块
 - 可单独配置成串口、SPI、I2C
 - 支持可编程的波特率发生器
- ADC (模数转换器)
 - 12位ADC，转换速度可达700 kSPS
 - 支持2个采样/保持
 - 多达8通道单端输入和1通道内部band-gap输入
 - 可通过软件触发，PWM触发，模拟比较器触发或外部中断脚触发等方式开启转换
 - 支持测量芯片温度功能
 - 最大支持4个通道同时连续的采样
- 可编程增益运算放大器 (PGA)
 - 支持8个增益等级1, 2, 3, 5, 7, 9, 11 和 13
 - 增益频宽高达 8 MHz
- 模拟比较器
 - 2个16级内部可编程参考电压的模拟比较器
 - 集成 CRV (比较器参考电压)
 - 支持迟滞功能
 - 支持比较结果改变时产生中断
- 硬件除法器
 - 有符号数（2的补码）整形计算
 - 16位除数32位被除数计算能力
 - 32位商和32位余数输出（有符号扩展的16位余数）
 - 除零警告标记
 - 一个计算周期需要6个CPU时钟
 - 读取商和余数时自动等待计算结束
- 支持ISP (在系统编程) 和 ICP (在电路编程)

- BOD (欠压检测)
 - 8 个可编程门限电压: 4.3V/4.0V/3.7V/3.0V/2.7V/2.4V/2.2V/2.0V
 - 可设定欠压时产生中断或系统复位
- 96位唯一识别码
- LVR (低压复位)
- 工作温度: -40°C~105°C
- 可靠性: EFT > ± 4 KV, ESD HBM 可达 4KV
- 封装:
 - 无铅 (RoHS)
 - 20脚 TSSOP, 28脚 TSSOP, 33脚 QFN

3 缩写表

Acronym	Description
ACMP	Analog Comparator Controller
ADC	Analog-to-Digital Converter
AHB	Advanced High-Performance Bus
APB	Advanced Peripheral Bus
BOD	Brown-out Detection
BPWM	Basic Pulse Width Modulation
DAP	Debug Access Port
EPWM	Enhanced Pulse Width Modulation
FIFO	First In, First Out
FMC	Flash Memory Controller
GPIO	General-Purpose Input/Output
HCLK	The Clock of Advanced High-Performance Bus
HIRC	48 MHz Internal High Speed RC Oscillator
HXT	4~24 MHz External High Speed Crystal Oscillator
ICP	In Circuit Programming
ISP	In System Programming
ISR	Interrupt Service Routine
LDO	Low Dropout Regulator
LIRC	10 kHz internal low speed RC oscillator (LIRC)
LXT	32.768 kHz External Low Speed Crystal Oscillator
NVIC	Nested Vectored Interrupt Controller
PCLK	The Clock of Advanced Peripheral Bus
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
SPS	Samples per Second
TMR	Timer Controller
UART	Universal Asynchronous Receiver/Transmitter
UCID	Unique Customer ID
WDT	Watchdog Timer

表 3-1 缩写词列表

4 器件信息和管脚定义

4.1 NuMicro® NM1120 命名规则

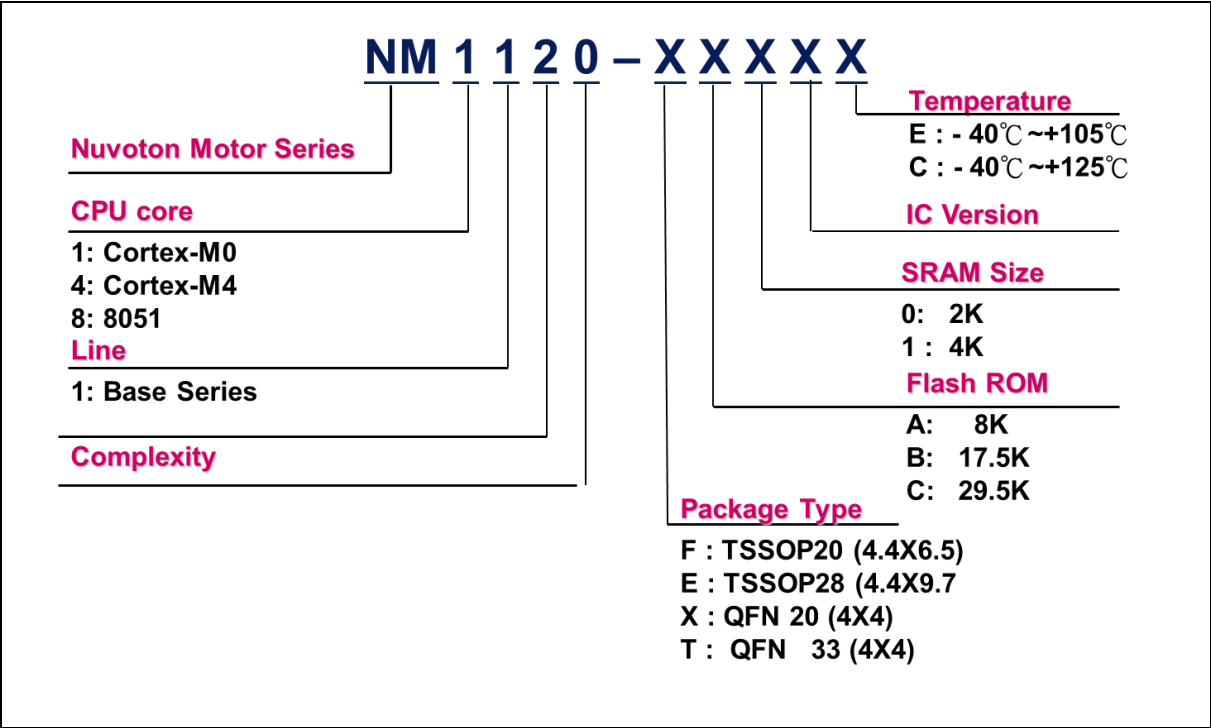


图 4.1-1 NuMicro® NM1120 系列产品命名规则

4.2 NuMicro® NM1120 系列产品选型指南

* USCI可配置成UART, SPI或I2C

Part Number	Flash (KB)	SRAM (KB)	ISP Loader ROM (KB)	Data Flash	I/O	Timer	Connectivity			IRC 48 MHz*	BOD	PWM	Analog Comp.	PGA	ADC (12-bit)	Temperature Sensor	ICP/ISP/AP	Package
							USCI											
							UART*	I ² C	SPI									
NM1120FC1AE	29.5	4	2	✓	18	2	2	2	2	1	1	6	2	1	8x12bit	1	✓	TSSOP20
NM1120EC1AE	29.5	4	2	✓	22	2	2	2	2	1	1	6	2	1	8x12bit	1	✓	TSSOP28
NM1120XC1AE	29.5	4	2	✓	18	2	2	2	2	1	1	6	2	1	8x12bit	1	✓	QFN20
NM1120ZC1AE	29.5	4	2	✓	22	2	2	2	2	1	1	6	2	1	8x12bit	1	✓	QFN33

表 4.2-1 NuMicro® NM1120 系列产品选型指南

4.3 管脚配置

4.3.1 TSSOP 28-Pin

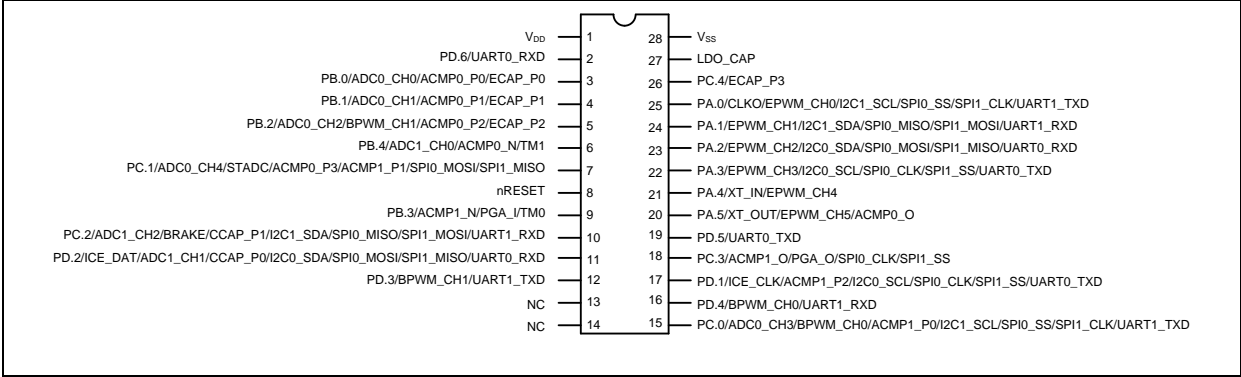


图 4.3-1NuMicro® NM1120 系列 TSSOP 28-pin 多功能管脚图

4.3.2 TSSOP 20-Pin

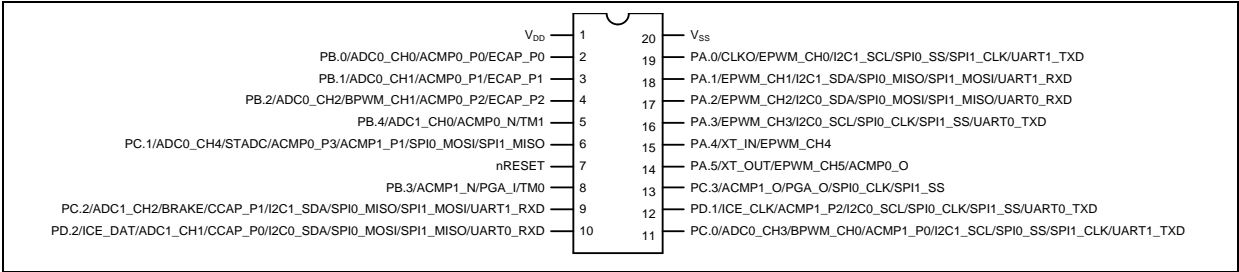


图 4.3-2 NuMicro® NM1120 系列 TSSOP 20-pin 多功能管脚图

4.3.3 QFN 33-Pin

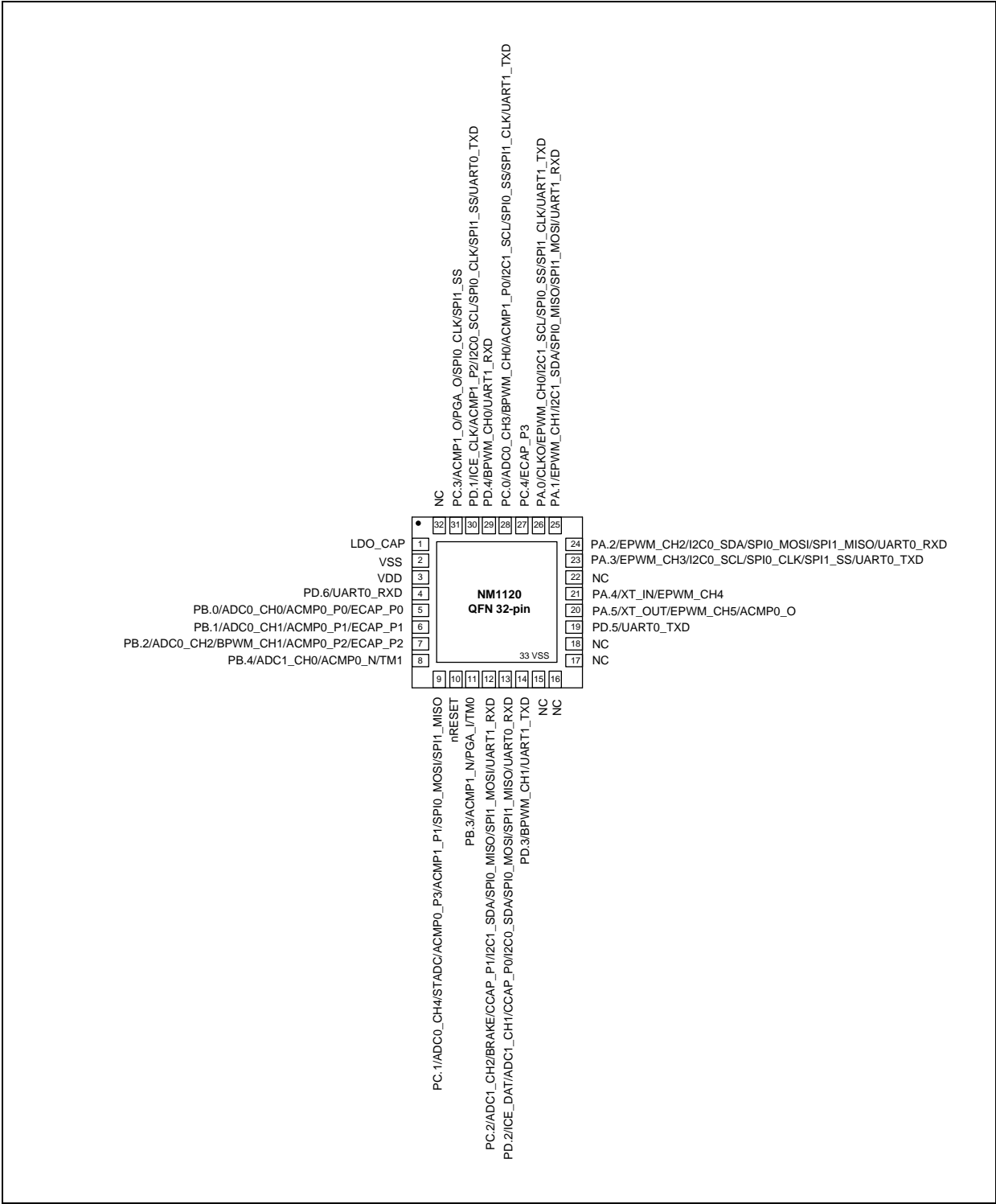


图 4.3-3 NuMicro® NM1120 Series QFN 33-pin 管脚图

4.3.4 QFN 20-Pin

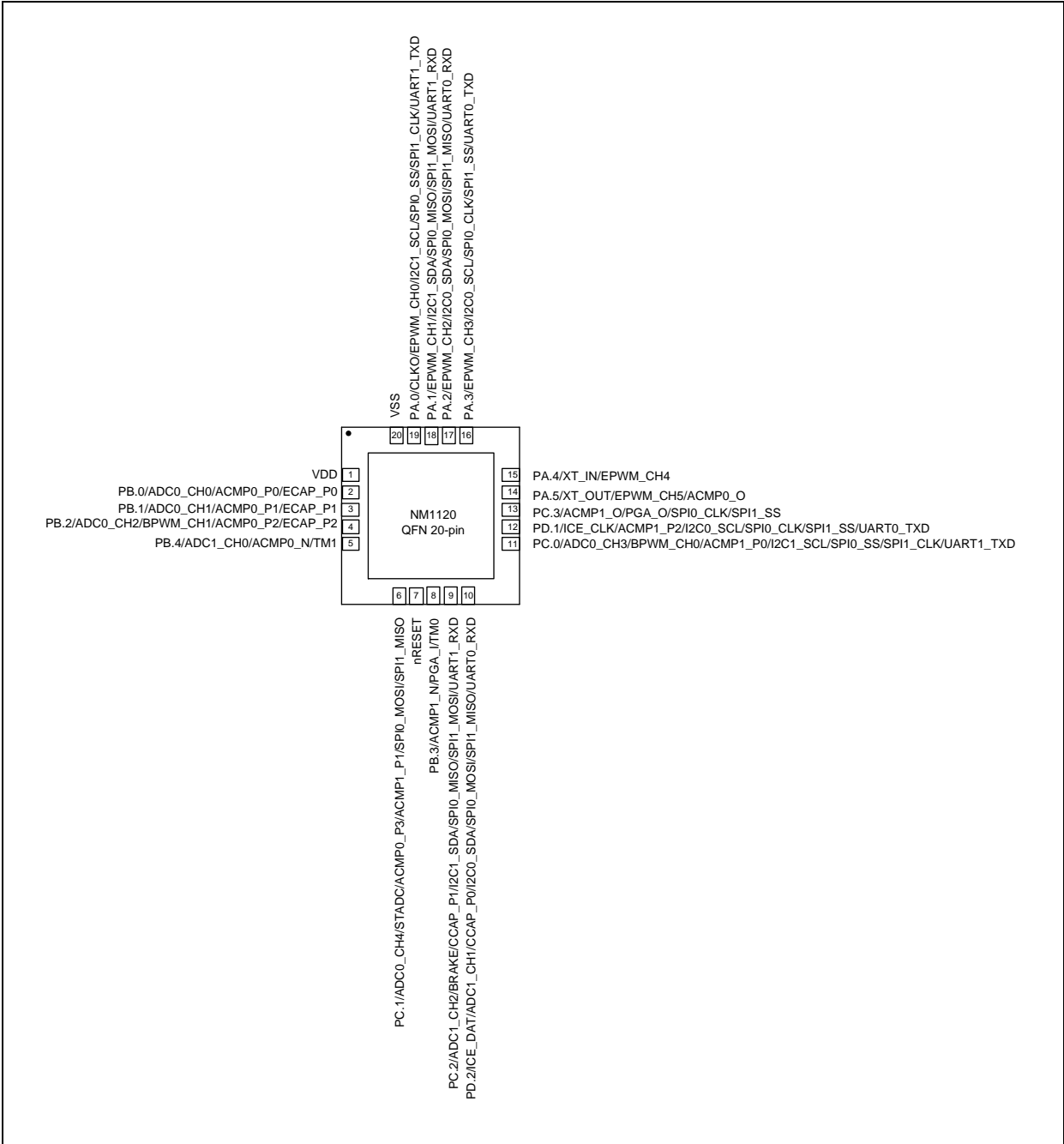


图 4.3-4 NuMicro® NM1120 系列 QFN 20-pin 多功能管脚图

4.4 管脚描述

4.4.1 NM1120 系列管脚描述

MFP* = 多功能管脚. (参考 SYS_GP_x_MFP)

PA.0 MFP0 等同于 SYS_GPA_MFP[3:0]=0x0.

PA.4 MFP5 等同于 SYS_GPA_MFP[19:16]=0x5.

MFP 仅配置PAD的输出数据或输入数据; PAD的方向在PMD配置.

同一个多功能MFP的优先级为: GPA > GPB > GPC > GPD.

多功能的类型A需要配置成输入口.

4.4.1.1 NM1120 系列 TSSOP28 管脚描述

管脚号	管脚名	类型	MFP*	描述
1	V _{DD}	A	MFP0	I/O口供电电压及内部PLL和数字模块供电的LDO输入电压.
2	PD.6	I/O	MFP0	通用的数字I/O口.
	UART0_RXD	I	MFPB	串口0的数据接收输入脚.
3	PB.0	I/O	MFP0	通用的数字I/O口.
	ADC0_CH0	A	MFP2	ADC0 通道0 模拟输入.
	ACMP0_P0	A	MFP4	模拟比较器0 正极输入脚.
	ECAP_P0	I	MFP7	增强型输入捕捉的输入脚
4	PB.1	I/O	MFP0	通用的数字I/O口.
	ADC0_CH1	A	MFP2	ADC0 通道1 模拟输入.
	ACMP0_P1	A	MFP4	模拟比较器0 正极输入脚.
	ECAP_P1	I	MFP7	增强型输入捕捉的输入脚
5	PB.2	I/O	MFP0	通用的数字I/O口.
	ADC0_CH2	A	MFP2	ADC0 通道2 模拟输入.
	BPWM_CH1	I/O	MFP3	PWM 通道1 输出脚 或 捕捉输入脚.
	ACMP0_P2	A	MFP4	模拟比较器0 正极输入脚.
	ECAP_P2	I	MFP7	增强型输入捕捉的输入脚
6	PB.4	I/O	MFP0	通用的数字I/O口.
	ADC1_CH0	A	MFP2	ADC1 通道0 模拟输入.
	ACMP0_N	A	MFP4	模拟比较器0 负极输入脚.
	TM1	I/O	MFP7	定时器1 事件计数输入脚 或 触发输出脚
7	PC.1	I/O	MFP0	通用的数字I/O口.
	ADC0_CH4	A	MFP2	ADC0 通道4 模拟输入.

管脚号	管脚名	类型	MFP*	描述
	STADC	I	MFP3	ADC 外部触发输入脚.
	ACMP0_P3	A	MFP4	模拟比较器0 正极输入脚.
	ACMP1_P1	A	MFP5	模拟比较器1 正极输入脚.
	SPI0_MOSI	I/O	MFP9	SPI0 1st MOSI(主机输出, 从机输入)脚.
	SPI1_MISO	I/O	MFP4	SPI1 MISO(主机输入, 从机输出)脚.
8	nRESET	I	MFP0	外部复位脚: 低电平有效, 内部有上拉电阻, 拉低这个脚将使MCU复位.
9	PB.3	I/O	MFP0	通用的数字I/O口.
	ACMP1_N	A	MFP5	模拟比较器1 负极输入脚.
	PGA_I	A	MFP6	可编程增益放大器 输入脚
	TM0	I/O	MFP7	定时器0 事件计数器输入脚 或 触发输出脚
10	PC.2	I/O	MFP0	通用的数字I/O口.
	ADC1_CH2	A	MFP2	ADC1 通道2 模拟输入脚.
	BRAKE	I	MFP3	EPWM 故障停止输入脚.
	CCAP_P1	I	MFP7	定时器连续捕捉输入脚
	I2C1_SDA	I/O	MFP8	I2C1 数据脚.
	SPI0_MISO	I/O	MFP9	SPI0 1st MISO (主机输入, 从机输出)脚.
	SPI1_MOSI	I/O	MFP4	SPI1 MOSI (主机输出, 从机输入)脚.
	UART1_RXD	I	MFPB	串口1数据接收输入脚.
11	PD.2	I/O	MFP0	通用的数字I/O口.
	ICE_DAT	I/O	MFP1	串行调试器 数据脚
	ADC1_CH1	A	MFP2	ADC1 通道1 模拟输入.
	CCAP_P0	I	MFP7	定时器连续捕捉输入脚
	I2C0_SDA	I/O	MFP8	I2C0 数据脚.
	SPI0_MOSI	I/O	MFP9	SPI0 1st MOSI(主机输出, 从机输入)脚.
	SPI1_MISO	I/O	MFP4	SPI1 MISO(主机输入, 从机输出)脚.
	UART0_RXD	I	MFPB	串口0的数据接收输入脚.
12	PD.3	I/O	MFP0	通用的数字I/O口.
	BPWM_CH1	I/O	MFP3	PWM 通道1 输出脚 或 捕捉输入脚.
	UART1_TXD	O	MFPB	串口1数据发送输出脚.
13	NC			空脚
14	NC			空脚

管脚号	管脚名	类型	MFP*	描述
15	PC.0	I/O	MFP0	通用的数字I/O口.
	ADC0_CH3	A	MFP2	ADC0 通道3 模拟输入.
	BPWM_CH0	I/O	MFP3	PWM 通道0 输出脚 或 捕捉输入脚
	ACMP1_P0	A	MFP5	模拟比较器1 正极输入脚.
	I2C1_SCL	I/O	MFP8	I2C1 时钟脚.
	SPI0_SS	I/O	MFP9	SPI0 从机选择脚.
	SPI1_CLK	I/O	MFPA	SPI1 串行时钟脚
	UART1_TXD	O	MFPB	串口1数据发送输出脚.
16	PD.4	I/O	MFP0	通用的数字I/O口.
	BPWM_CH0	I/O	MFP3	PWM 通道0 输出脚 或 捕捉输入脚
	UART1_RXD	I	MFPB	串口1数据接收输入脚.
17	PD.1	I/O	MFP0	通用的数字I/O口.
	ICE_CLK	I	MFP1	串行调试器时钟脚
	ACMP1_P2	A	MFP5	模拟比较器1 正极输入脚.
	I2C0_SCL	I/O	MFP8	I2C0 时钟脚.
	SPI0_CLK	I/O	MFP9	SPI0 串行时钟脚.
	SPI1_SS	I/O	MFPA	SPI1 从机选择脚
	UART0_TXD	O	MFPB	串口0 数据发送输出脚.
18	PC.3	I/O	MFP0	通用的数字I/O口.
	ACMP1_O	O	MFP5	模拟比较器1输出脚.
	PGA_O	A	MFP6	可编程增益放大器输出脚
	SPI0_CLK	I/O	MFP9	SPI0 串行时钟脚.
	SPI1_SS	I/O	MFPA	SPI1 从机选择脚
19	PD.5	I/O	MFP0	通用的数字I/O口.
	UART0_TXD	O	MFPB	串口0 数据发送输出脚.
20	PA.5	I/O	MFP0	通用的数字I/O口.
	XT_OUT	O	MFP1	4~24 MHz (高速) 外部晶振输出脚.
	EPWM_CH5	I/O	MFP3	PWM 通道5 输出脚 或 捕捉输入脚.
	ACMP0_O	O	MFP4	模拟比较器0输出脚.
21	PA.4	I/O	MFP0	通用的数字I/O口.
	XT_IN	I	MFP1	4~24 MHz (高速) 外部晶振输入脚.

管脚号	管脚名	类型	MFP*	描述
	EPWM_CH4	I/O	MFP3	PWM 通道4 输出脚 或 捕捉输入脚.
22	PA.3	I/O	MFP0	通用的数字I/O口.
	EPWM_CH3	I/O	MFP3	PWM 通道3 输出脚 或 捕捉输入脚.
	I2C0_SCL	I/O	MFP8	I2C0 时钟脚.
	SPI0_CLK	I/O	MFP9	SPI0 串行时钟脚.
	SPI1_SS	I/O	MFPA	SPI1 从机选择脚
	UART0_TXD	O	MFPB	串口0 数据发送输出脚.
23	PA.2	I/O	MFP0	通用的数字I/O口.
	EPWM_CH2	I/O	MFP3	PWM 通道2 输出脚 或 捕捉输入脚.
	I2C0_SDA	I/O	MFP8	I2C0 数据脚.
	SPI0_MOSI	I/O	MFP9	SPI0 1st MOSI(主机输出, 从机输入)脚.
	SPI1_MISO	I/O	MFPA	SPI1 MISO(主机输入, 从机输出)脚.
	UART0_RXD	I	MFPB	串口0的数据接收输入脚.
24	PA.1	I/O	MFP0	通用的数字I/O口.
	EPWM_CH1	I/O	MFP3	PWM 通道1 输出脚 或 捕捉输入脚.
	I2C1_SDA	I/O	MFP8	I2C1 数据脚.
	SPI0_MISO	I/O	MFP9	SPI0 1st MISO (主机输入, 从机输出) 脚.
	SPI1_MOSI	I/O	MFPA	SPI1 MOSI (主机输出, 从机输入) 脚.
	UART1_RXD	I	MFPB	串口1数据接收输入脚.
25	PA.0	I/O	MFP0	通用的数字I/O口.
	CLKO	O	MFP1	时钟输出脚
	EPWM_CH0	I/O	MFP3	PWM 通道0 输出脚 或 捕捉输入脚
	I2C1_SCL	I/O	MFP8	I2C1 时钟脚.
	SPI0_SS	I/O	MFP9	SPI0 从机选择脚.
	SPI1_CLK	I/O	MFPA	SPI1 串行时钟脚
	UART1_TXD	O	MFPB	串口1数据发送输出脚.
26	PC.4	I/O	MFP0	通用的数字I/O口.
	ECAP_P3	I	MFP7	增强型输入捕捉的输入脚
27	LDO_CAP	A	MFP0	LDO 输出脚.
28	V _{SS}	A	MFP0	数字地.

表 4.4-1 TSSOP28 管脚描述

4.4.1.2 NM1120 系列 TSSOP20 管脚描述

管脚号	管脚名	类型	MFP*	描述
1	V _{DD}	A	MFP0	I/O口供电电压及内部PLL和数字模块供电的LDO输入电压.
2	PB.0	I/O	MFP0	通用的数字I/O口.
	ADC0_CH0	A	MFP2	ADC0 通道0 模拟输入.
	ACMP0_P0	A	MFP4	模拟比较器0 正极输入脚.
	ECAP_P0	I	MFP7	增强型输入捕捉的输入脚
3	PB.1	I/O	MFP0	通用的数字I/O口.
	ADC0_CH1	A	MFP2	ADC0 通道1 模拟输入.
	ACMP0_P1	A	MFP4	模拟比较器0 正极输入脚.
	ECAP_P1	I	MFP7	增强型输入捕捉的输入脚
4	PB.2	I/O	MFP0	通用的数字I/O口.
	ADC0_CH2	A	MFP2	ADC0 通道2 模拟输入.
	BPWM_CH1	I/O	MFP3	PWM 通道1 输出脚 或 捕捉输入脚.
	ACMP0_P2	A	MFP4	模拟比较器0 正极输入脚.
	ECAP_P2	I	MFP7	增强型输入捕捉的输入脚
5	PB.4	I/O	MFP0	通用的数字I/O口.
	ADC1_CH0	A	MFP2	ADC1 通道0 模拟输入.
	ACMP0_N	A	MFP4	模拟比较器0 负极输入脚.
	TM1	I/O	MFP7	定时器1 事件计数输入脚 或 触发输出脚
6	PC.1	I/O	MFP0	通用的数字I/O口.
	ADC0_CH4	A	MFP2	ADC0 通道4 模拟输入.
	STADC	I	MFP3	ADC 外部触发输入脚.
	ACMP0_P3	A	MFP4	模拟比较器0 正极输入脚.
	ACMP1_P1	A	MFP5	模拟比较器1 正极输入脚.
	SPI0_MOSI	I/O	MFP9	SPI0 1st MOSI(主机输出, 从机输入)脚.
	SPI1_MISO	I/O	MFP4	SPI1 MISO(主机输入, 从机输出)脚.
7	nRESET	I	MFP0	外部复位脚: 低电平有效, 内部有上拉电阻, 拉低这个脚将使MCU复位.
8	PB.3	I/O	MFP0	通用的数字I/O口.
	ACMP1_N	A	MFP5	模拟比较器1 负极输入脚.
	PGA_I	A	MFP6	可编程增益放大器 输入脚
	TM0	I/O	MFP7	定时器0 事件计数器输入脚 或 触发输出脚

管脚号	管脚名	类型	MFP*	描述
9	PC.2	I/O	MFP0	通用的数字I/O口.
	ADC1_CH2	A	MFP2	ADC1 通道2 模拟输入脚.
	BRAKE	I	MFP3	EPWM 故障停止输入脚.
	CCAP_P1	I	MFP7	定时器连续捕捉输入脚
	I2C1_SDA	I/O	MFP8	I2C1 数据脚.
	SPI0_MISO	I/O	MFP9	SPI0 1st MISO (主机输入, 从机输出) 脚.
	SPI1_MOSI	I/O	MFP9	SPI1 MOSI (主机输出, 从机输入) 脚.
	UART1_RXD	I	MFPB	串口1数据接收输入脚.
10	PD.2	I/O	MFP0	通用的数字I/O口.
	ICE_DAT	I/O	MFP1	串行调试器 数据脚
	ADC1_CH1	A	MFP2	ADC1 通道1 模拟输入.
	CCAP_P0	I	MFP7	定时器连续捕捉输入脚
	I2C0_SDA	I/O	MFP8	I2C0 数据脚.
	SPI0_MOSI	I/O	MFP9	SPI0 1st MOSI(主机输出, 从机输入)脚.
	SPI1_MISO	I/O	MFP9	SPI1 MISO(主机输入, 从机输出)脚.
	UART0_RXD	I	MFPB	串口0的数据接收输入脚.
11	PC.0	I/O	MFP0	通用的数字I/O口.
	ADC0_CH3	A	MFP2	ADC0 通道3 模拟输入.
	BPWM_CH0	I/O	MFP3	PWM 通道0 输出脚 或 捕捉输入脚
	ACMP1_P0	A	MFP5	模拟比较器1 正极输入脚.
	I2C1_SCL	I/O	MFP8	I2C1 时钟脚.
	SPI0_SS	I/O	MFP9	SPI0 从机选择脚.
	SPI1_CLK	I/O	MFP9	SPI1 串行时钟脚
	UART1_TXD	O	MFPB	串口1数据发送输出脚.
12	PD.1	I/O	MFP0	通用的数字I/O口.
	ICE_CLK	I	MFP1	串行调试器时钟脚
	ACMP1_P2	A	MFP5	模拟比较器1 正极输入脚.
	I2C0_SCL	I/O	MFP8	I2C0 时钟脚.
	SPI0_CLK	I/O	MFP9	SPI0 串行时钟脚.
	SPI1_SS	I/O	MFP9	SPI1 从机选择脚
	UART0_TXD	O	MFPB	串口0 数据发送输出脚.

管脚号	管脚名	类型	MFP*	描述
13	PC.3	I/O	MFP0	通用的数字I/O口。
	ACMP1_O	O	MFP5	模拟比较器1输出脚。
	PGA_O	A	MFP6	可编程增益放大器输出脚
	SPI0_CLK	I/O	MFP9	SPI0 串行时钟脚。
	SPI1_SS	I/O	MFP4	SPI1 从机选择脚
14	PA.5	I/O	MFP0	通用的数字I/O口。
	XT_OUT	O	MFP1	4~24 MHz (高速) 外部晶振输出脚。
	EPWM_CH5	I/O	MFP3	PWM 通道5 输出脚 或 捕捉输入脚。
	ACMP0_O	O	MFP4	模拟比较器0输出脚。
15	PA.4	I/O	MFP0	通用的数字I/O口。
	XT_IN	I	MFP1	4~24 MHz (高速) 外部晶振输入脚。
	EPWM_CH4	I/O	MFP3	PWM 通道4 输出脚 或 捕捉输入脚。
16	PA.3	I/O	MFP0	通用的数字I/O口。
	EPWM_CH3	I/O	MFP3	PWM 通道3 输出脚 或 捕捉输入脚。
	I2C0_SCL	I/O	MFP8	I2C0 时钟脚。
	SPI0_CLK	I/O	MFP9	SPI0 串行时钟脚。
	SPI1_SS	I/O	MFP4	SPI1 从机选择脚
	UART0_TXD	O	MFPB	串口0 数据发送输出脚。
17	PA.2	I/O	MFP0	通用的数字I/O口。
	EPWM_CH2	I/O	MFP3	PWM 通道2 输出脚 或 捕捉输入脚。
	I2C0_SDA	I/O	MFP8	I2C0 数据脚。
	SPI0_MOSI	I/O	MFP9	SPI0 1st MOSI(主机输出，从机输入)脚。
	SPI1_MISO	I/O	MFP4	SPI1 MISO(主机输入，从机输出)脚。
	UART0_RXD	I	MFPB	串口0的数据接收输入脚。
18	PA.1	I/O	MFP0	通用的数字I/O口。
	EPWM_CH1	I/O	MFP3	PWM 通道1 输出脚 或 捕捉输入脚。
	I2C1_SDA	I/O	MFP8	I2C1 数据脚。
	SPI0_MISO	I/O	MFP9	SPI0 1st MISO（主机输入，从机输出）脚。
	SPI1_MOSI	I/O	MFP4	SPI1 MOSI（主机输出，从机输入）脚。
	UART1_RXD	I	MFPB	串口1数据接收输入脚。
19	PA.0	I/O	MFP0	通用的数字I/O口。

管脚号	管脚名	类型	MFP*	描述
	CLKO	O	MFP1	时钟输出脚
	EPWM_CH0	I/O	MFP3	PWM 通道0 输出脚 或 捕捉输入脚
	I2C1_SCL	I/O	MFP8	I2C1 时钟脚.
	SPI0_SS	I/O	MFP9	SPI0 从机选择脚.
	SPI1_CLK	I/O	MFPA	SPI1 串行时钟脚
	UART1_TXD	O	MFPB	串口1数据发送输出脚.
20	V _{SS}	A	MFP0	数字地.

表 4.4-2 TSSOP20 管脚描述

4.4.1.3 NM1120 系列 QFN33 管脚描述

QFN33 管脚号	管脚名	类型	MFP*	描述
1	LDO_CAP	A	MFP0	LDO 输出脚.
2	V _{SS}	A	MFP0	数字地.
33				
3	V _{DD}	A	MFP0	I/O 口供电电压及内部 PLL 和数字模块供电的 LDO 输入电压.
4	PD.6	I/O	MFP0	通用的数字 I/O 口.
	UART0_RXD	I	MFPB	串口 0 的数据接收输入脚.
5	PB.0	I/O	MFP0	通用的数字 I/O 口.
	ADC0_CH0	A	MFP2	ADC0 通道 0 模拟输入.
	ACMP0_P0	A	MFP4	模拟比较器 0 正极输入脚.
	ECAP_P0	I	MFP7	增强型输入捕捉的输入脚
6	PB.1	I/O	MFP0	通用的数字 I/O 口.
	ADC0_CH1	A	MFP2	ADC0 通道 1 模拟输入.
	ACMP0_P1	A	MFP4	模拟比较器 0 正极输入脚.
	ECAP_P1	I	MFP7	增强型输入捕捉的输入脚
7	PB.2	I/O	MFP0	通用的数字 I/O 口.
	ADC0_CH2	A	MFP2	ADC0 通道 2 模拟输入.
	BPWM_CH1	I/O	MFP3	PWM 通道 1 输出脚 或 捕捉输入脚.
	ACMP0_P2	A	MFP4	模拟比较器 0 正极输入脚.
	ECAP_P2	I	MFP7	增强型输入捕捉的输入脚
8	PB.4	I/O	MFP0	通用的数字 I/O 口.
	ADC1_CH0	A	MFP2	ADC1 通道 0 模拟输入.
	ACMP0_N	A	MFP4	模拟比较器 0 负极输入脚.
	TM1	I/O	MFP7	定时器 1 事件计数输入脚 或 触发输出脚
9	PC.1	I/O	MFP0	通用的数字 I/O 口.
	ADC0_CH4	A	MFP2	ADC0 通道 4 模拟输入.
	STADC	I	MFP3	ADC 外部触发输入脚.
	ACMP0_P3	A	MFP4	模拟比较器 0 正极输入脚.
	ACMP1_P1	A	MFP5	模拟比较器 1 正极输入脚.
	SPI0_MOSI	I/O	MFP9	SPI0 1st MOSI(主机输出, 从机输入)脚.

	SPI1_MISO	I/O	MFP0	SPI1 MISO(主机输入，从机输出)脚。
10	nRESET	I	MFP0	外部复位脚：低电平有效，内部有上拉电阻，拉低这个脚将使 MCU 复位。
11	PB.3	I/O	MFP0	通用的数字 I/O 口。
	ACMP1_N	A	MFP5	模拟比较器 1 负极输入脚。
	PGA_I	A	MFP6	可编程增益放大器 输入脚
	TM0	I/O	MFP7	定时器 0 事件计数器输入脚 或 触发输出脚
12	PC.2	I/O	MFP0	通用的数字 I/O 口。
	ADC1_CH2	A	MFP2	ADC1 通道 2 模拟输入脚。
	BRAKE	I	MFP3	EPWM 故障停止输入脚。
	CCAP_P1	I	MFP7	定时器连续捕捉输入脚
	I2C1_SDA	I/O	MFP8	I2C1 数据脚。
	SPI0_MISO	I/O	MFP9	SPI0 1st MISO（主机输入，从机输出）脚。
	SPI1_MOSI	I/O	MFP0	SPI1 MOSI（主机输出，从机输入）脚。
	UART1_RXD	I	MFPB	串口 1 数据接收输入脚。
13	PD.2	I/O	MFP0	通用的数字 I/O 口。
	ICE_DAT	I/O	MFP1	串行调试器 数据脚
	ADC1_CH1	A	MFP2	ADC1 通道 1 模拟输入。
	CCAP_P0	I	MFP7	定时器连续捕捉输入脚
	I2C0_SDA	I/O	MFP8	I2C0 数据脚。
	SPI0_MOSI	I/O	MFP9	SPI0 1st MOSI(主机输出，从机输入)脚。
	SPI1_MISO	I/O	MFP0	SPI1 MISO(主机输入，从机输出)脚。
	UART0_RXD	I	MFPB	串口 0 的数据接收输入脚。
14	PD.3	I/O	MFP0	通用的数字 I/O 口。
	BPWM_CH1	I/O	MFP3	PWM 通道 1 输出脚 或 捕捉输入脚。
	UART1_TXD	O	MFPB	串口 1 数据发送输出脚。
15	NC			空脚
16	NC			空脚
17	NC			空脚
18	NC			空脚
19	PD.5	I/O	MFP0	通用的数字 I/O 口。
	UART0_TXD	O	MFPB	串口 0 数据发送输出脚。
20	PA.5	I/O	MFP0	通用的数字 I/O 口。

	XT_OUT	O	MFP1	4~24 MHz (高速) 外部晶振输出脚.
	EPWM_CH5	I/O	MFP3	PWM 通道 5 输出脚 或 捕捉输入脚.
	ACMP0_O	O	MFP4	模拟比较器 0 输出脚.
21	PA.4	I/O	MFP0	通用的数字 I/O 口.
	XT_IN	I	MFP1	4~24 MHz (高速) 外部晶振输入脚.
	EPWM_CH4	I/O	MFP3	PWM 通道 4 输出脚 或 捕捉输入脚.
22	NC			空脚
23	PA.3	I/O	MFP0	通用的数字 I/O 口.
	EPWM_CH3	I/O	MFP3	PWM 通道 3 输出脚 或 捕捉输入脚.
	I2C0_SCL	I/O	MFP8	I2C0 时钟脚.
	SPI0_CLK	I/O	MFP9	SPI0 串行时钟脚.
	SPI1_SS	I/O	MFP4	SPI1 从机选择脚
	UART0_TXD	O	MFPB	串口 0 数据发送输出脚.
24	PA.2	I/O	MFP0	通用的数字 I/O 口.
	EPWM_CH2	I/O	MFP3	PWM 通道 2 输出脚 或 捕捉输入脚.
	I2C0_SDA	I/O	MFP8	I2C0 数据脚.
	SPI0_MOSI	I/O	MFP9	SPI0 1st MOSI(主机输出, 从机输入)脚.
	SPI1_MISO	I/O	MFP4	SPI1 MISO(主机输入, 从机输出)脚.
	UART0_RXD	I	MFPB	串口 0 的数据接收输入脚.
25	PA.1	I/O	MFP0	通用的数字 I/O 口.
	EPWM_CH1	I/O	MFP3	PWM 通道 1 输出脚 或 捕捉输入脚.
	I2C1_SDA	I/O	MFP8	I2C1 数据脚.
	SPI0_MISO	I/O	MFP9	SPI0 1st MISO (主机输入, 从机输出) 脚.
	SPI1_MOSI	I/O	MFP4	SPI1 MOSI (主机输出, 从机输入) 脚.
	UART1_RXD	I	MFPB	串口 1 数据接收输入脚.
26	PA.0	I/O	MFP0	通用的数字 I/O 口.
	CLKO	O	MFP1	时钟输出脚
	EPWM_CH0	I/O	MFP3	PWM 通道 0 输出脚 或 捕捉输入脚
	I2C1_SCL	I/O	MFP8	I2C1 时钟脚.
	SPI0_SS	I/O	MFP9	SPI0 从机选择脚.
	SPI1_CLK	I/O	MFP4	SPI1 串行时钟脚
	UART1_TXD	O	MFPB	串口 1 数据发送输出脚.

27	PC.4	I/O	MFP0	通用的数字 I/O 口。
	ECAP_P3	I	MFP7	增强型输入捕捉的输入脚
28	PC.0	I/O	MFP0	通用的数字 I/O 口。
	ADC0_CH3	A	MFP2	ADC0 通道 3 模拟输入。
	BPWM_CH0	I/O	MFP3	PWM 通道 0 输出脚 或 捕捉输入脚
	ACMP1_P0	A	MFP5	模拟比较器 1 正极输入脚。
	I2C1_SCL	I/O	MFP8	I2C1 时钟脚。
	SPI0_SS	I/O	MFP9	SPI0 从机选择脚。
	SPI1_CLK	I/O	MFPA	SPI1 串行时钟脚
	UART1_TXD	O	MFPB	串口 1 数据发送输出脚。
29	PD.4	I/O	MFP0	通用的数字 I/O 口。
	BPWM_CH0	I/O	MFP3	PWM 通道 0 输出脚 或 捕捉输入脚
	UART1_RXD	I	MFPB	串口 1 数据接收输入脚。
30	PD.1	I/O	MFP0	通用的数字 I/O 口。
	ICE_CLK	I	MFP1	串行调试器时钟脚
	ACMP1_P2	A	MFP5	模拟比较器 1 正极输入脚。
	I2C0_SCL	I/O	MFP8	I2C0 时钟脚。
	SPI0_CLK	I/O	MFP9	SPI0 串行时钟脚。
	SPI1_SS	I/O	MFPA	SPI1 从机选择脚
	UART0_TXD	O	MFPB	串口 0 数据发送输出脚。
31	PC.3	I/O	MFP0	通用的数字 I/O 口。
	ACMP1_O	O	MFP5	模拟比较器 1 输出脚。
	PGA_O	A	MFP6	可编程增益放大器输出脚
	SPI0_CLK	I/O	MFP9	SPI0 串行时钟脚。
	SPI1_SS	I/O	MFPA	SPI1 从机选择脚
32	NC			空脚

表 4.4-3 QFN33 管脚描述

4.4.2 GPIO 多功能管脚列表

MFP* = 多功能管脚. (参考SYS_GP_x_MFP)

PA.0 MFP0 等同于 SYS_GPA_MFP[3:0]=0x0.

PA.4 MFP5 等同于 SYS_GPA_MFP[19:16]=0x5.

组	管脚名称	GPIO	MFP*	类型	描述
ACMP0	ACMP0_P0	PB.0	MFP4	A	比较器 0 正极输入脚.
	ACMP0_P1	PB.1	MFP4	A	比较器 0 正极输入脚.
	ACMP0_P2	PB.2	MFP4	A	比较器 0 正极输入脚.
	ACMP0_N	PB.4	MFP4	A	比较器 0 负极输入脚.
	ACMP0_P3	PC.1	MFP4	A	比较器 0 正极输入脚.
	ACMP0_O	PA.5	MFP4	O	比较器 0 输出脚.
ACMP1	ACMP1_P1	PC.1	MFP5	A	比较器 1 正极输入脚.
	ACMP1_N	PB.3	MFP5	A	比较器 1 负极输入脚.
	ACMP1_O	PC.3	MFP5	O	比较器 1 输出脚.
	ACMP1_P2	PD.1	MFP5	A	比较器 1 正极输入脚.
	ACMP1_P0	PC.0	MFP5	A	比较器 1 正极输入脚.
ADC0	ADC0_CH0	PB.0	MFP2	A	ADC0 通道 0 模拟输入.
	ADC0_CH1	PB.1	MFP2	A	ADC0 通道 1 模拟输入.
	ADC0_CH2	PB.2	MFP2	A	ADC0 通道 2 模拟输入
	ADC0_CH4	PC.1	MFP2	A	ADC0 通道 4 模拟输入
	ADC0_CH3	PC.0	MFP2	A	ADC0 通道 3 模拟输入
ADC1	ADC1_CH0	PB.4	MFP2	A	ADC1 通道 0 模拟输入
	ADC1_CH2	PC.2	MFP2	A	ADC1 通道 2 模拟输入
	ADC1_CH1	PD.2	MFP2	A	ADC1 通道 1 模拟输入.
BPWM	BPWM_CH1	PB.2	MFP3	O	基础型 PWM 通道 1 输出
	BPWM_CH0	PC.0	MFP3	O	基础型 PWM 通道 0 输出
	BPWM_CH1	PD.3	MFP3	O	基础型 PWM 通道 1 输出
	BPWM_CH0	PD.4	MFP3	O	基础型 PWM 通道 0 输出
CCAP	CCAP_P1	PC.2	MFP7	I	连续捕捉输入
	CCAP_P0	PD.2	MFP7	I	连续捕捉输入
CLKO	CLKO	PA.0	MFP1	O	时钟输出脚.
ECAP	ECAP_P0	PB.0	MFP7	I	输入捕捉通道 0

	ECAP_P1	PB.1	MFP7	I	输入捕捉通道 1
	ECAP_P2	PB.2	MFP7	I	输入捕捉通道 2
EPWM	BRAKE	PC.2	MFP3	I	EPWM 停止输入脚.
	EPWM_CH5	PA.5	MFP3	O	增强型 PWM 输出脚.
	EPWM_CH4	PA.4	MFP3	O	增强型 PWM 输出脚.
	EPWM_CH3	PA.3	MFP3	O	增强型 PWM 输出脚.
	EPWM_CH2	PA.2	MFP3	O	增强型 PWM 输出脚
	EPWM_CH1	PA.1	MFP3	O	增强型 PWM 输出脚
	EPWM_CH0	PA.0	MFP3	O	增强型 PWM 输出脚.
I ² C	I2C1_SDA	PC.2	MFP8	I/O	I ² C1 数据脚.
	I2C0_SDA	PD.2	MFP8	I/O	I ² C0 数据脚.
	I2C0_SCL	PD.1	MFP8	I/O	I2C0 时钟脚.
	I2C1_SCL	PC.0	MFP8	I/O	I2C1 时钟脚.
	I2C0_SCL	PA.3	MFP8	I/O	I2C0 时钟脚.
	I2C0_SDA	PA.2	MFP8	I/O	I ² C0 数据脚.
	I2C1_SDA	PA.1	MFP8	I/O	I ² C1 数据脚.
	I2C1_SCL	PA.0	MFP8	I/O	I2C1 时钟脚.
ICE	ICE_DAT	PD.2	MFP1	I/O	串行调试器 数据脚
	ICE_CLK	PD.1	MFP1	I	串行调试器 时钟脚
nRESET	nRESET			I	外部复位脚, 内部上拉
PGA	PGA_I	PB.3	MFP6	A	可编程增益放大器输入脚
	PGA_O	PC.3	MFP6	A	可编程增益放大器输入脚
SPI0	SPI0_MOSI	PC.1	MFP9	I/O	SPI0 MOSI (主机输出, 从机输入) 管脚
	SPI0_MISO	PC.2	MFP9	I/O	SPI0 MISO (主机输入, 从机输出) 管脚.
	SPI0_MOSI	PD.2	MFP9	I/O	SPI0 MOSI (主机输出, 从机输入) 管脚
	SPI0_CLK	PC.3	MFP9	I/O	SPI0 时钟脚.
	SPI0_CLK	PD.1	MFP9	I/O	SPI0 时钟脚.
	SPI0_SS	PC.0	MFP9	I	SPI0 从机选择脚.
	SPI0_CLK	PA.3	MFP9	I/O	SPI0 时钟脚.
	SPI0_MOSI	PA.2	MFP9	I/O	SPI0 MOSI (主机输出, 从机输入) 管脚
	SPI0_MISO	PA.1	MFP9	I/O	SPI0 MISO (主机输入, 从机输出) 管脚.
	SPI0_SS	PA.0	MFP9	I	SPI0 从机选择脚.

SPI1	SPI1_MISO	PC.1	MFPA	I/O	SPI1 MISO(主机输入，从机输出)脚
	SPI1_MOSI	PC.2	MFPA	I/O	SPI1 MOSI（主机输出，从机输入）脚.
	SPI1_MISO	PD.2	MFPA	I/O	SPI1 MISO(主机输入，从机输出)脚
	SPI1_SS	PC.3	MFPA	I/O	SPI1 从机选择脚
	SPI1_SS	PD.1	MFPA	I/O	SPI1 从机选择脚
	SPI1_CLK	PC.0	MFPA	I/O	SPI1 时钟脚.
	SPI1_SS	PA.3	MFPA	I	SPI1 从机选择脚.
	SPI1_MISO	PA.2	MFPA	I/O	SPI1 MISO(主机输入，从机输出)脚.
	SPI1_MOSI	PA.1	MFPA	I/O	SPI1 MOSI（主机输出，从机输入）脚.
	SPI1_CLK	PA.0	MFPA	I/O	SPI1 时钟脚.
STADC	STADC	PC.1	MFP3	I	ADC 外部触发输入脚.
TM0	TM0	PB.3	MFP7	I	定时器 0 事件计数器输入脚 或 触发输出脚
TM1	TM1	PB.4	MFP7	I	定时器 1 事件计数输入脚 或 触发输出脚
UART0	UART0_RXD	PD.2	MFPB	I	串口 0 数据接收输入脚
	UART0_TXD	PD.1	MFPB	O	串口 0 数据接收输出脚
	UART0_TXD	PA.3	MFPB	O	串口 0 数据接收输出脚
	UART0_RXD	PA.2	MFPB	I	串口 0 数据接收输入脚
	UART0_TXD	PD.5	MFPB	O	串口 0 数据接收输出脚
	UART0_RXD	PD.6	MFPB	I	串口 0 数据接收输入脚
UART1	UART1_RXD	PC.2	MFPB	I	串口 1 数据接收输入脚
	UART1_TXD	PC.0	MFPB	O	串口 1 数据接收输出脚
	UART1_RXD	PA.1	MFPB	I	串口 1 数据接收输入脚
	UART1_TXD	PA.0	MFPB	O	串口 1 数据接收输出脚
	UART1_TXD	PD.3	MFPB	O	串口 1 数据接收输出脚
	UART1_RXD	PD.4	MFPB	I	串口 1 数据接收输入脚
XT	XT_OUT	PA.5	MPF1	A	外部晶振输出脚
	XT_IN	PA.4	MFP1	A	外部晶振输入脚

表 4.4-4 TSSOP20 多功能管脚列表

5 方框图

5.1 NuMicro® NM1120 方框图

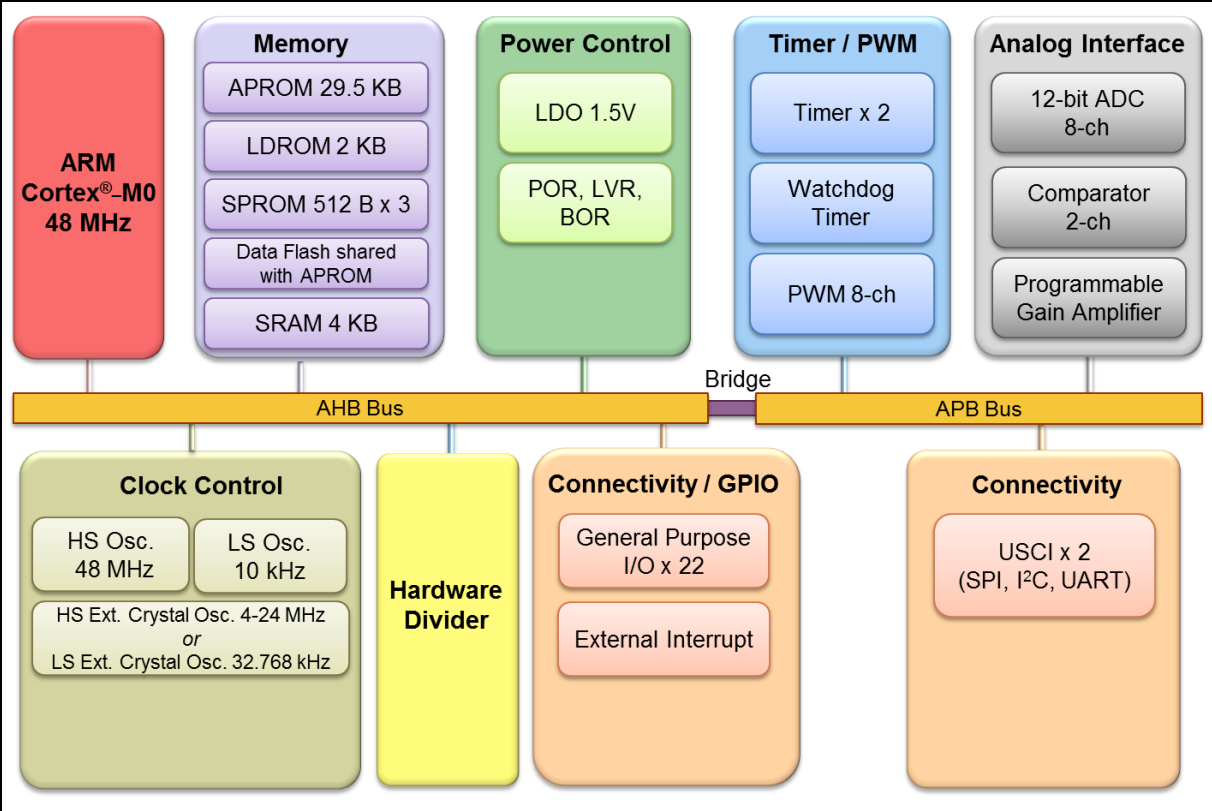


图 5.1-1 NuMicro® NM1120 方框图

6 功能描述

6.1 ARM® Cortex®-M0 内核

6.1.1 概述

Cortex®-M0是一个可配置的，具有多级流水处理能力的32位精简指令处理器。它内部有一个AMBA AHB-Lite接口和一个嵌套向量中断控制器(NVIC)。同时有一个可选配的硬件调试器。它可以执行Thumb指令并且兼容其他的Cortex®-M处理器。该处理器支持2种模式- Thread模式和Handler模式。CPU异常时进入Handler模式。异常只能在Handler模式下退出。系统复位和异常退出后可进入Thread模式。图6.1-1是处理器的功能框图。

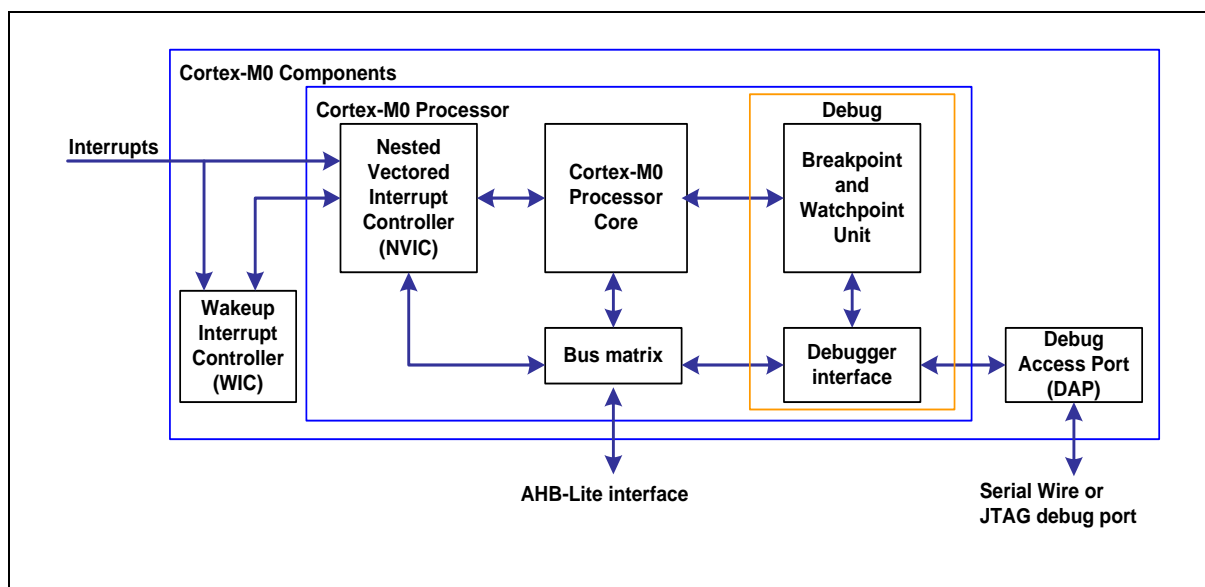


图 6.1-1 功能框图

6.1.2 特性

该处理器包含：

- 低门数处理器：
 - ARMv6-M Thumb® 指令集
 - Thumb-2 技术
 - 兼容ARMv6-M的24位系统定时器
 - 一个32位的硬件乘法器
 - 系统接口支持小端数据访问
 - 精准而及时的中断处理能力
 - 加载/存储多个数据和多周期乘法指令可被终止，然后重新开始，从而实现快速中断处理

- C 应用程序二进制接口异常兼容模式. ARMv6-M的C应用程序二进制接口(C-ABI)异常兼容模式允许用户在中断处理中使用纯C函数。
- 运行WFI指令，WFE指令或者退出即睡眠特性可以进入低功耗睡眠模式
- NVIC:
 - 32个外部中断，每个中断有4个优先级
 - 专用的不可屏蔽中断 (NMI)
 - 同时支持电平和脉冲触发中断
 - 支持中断唤醒控制器(WIC) ， 提供极低功耗空闲模式
- 调试支持:
 - 四个硬件断点
 - 两个观察点
 - 用于非侵入代码分析的程序计数采样寄存器(PCSR)
 - 单步和向量捕捉能力
- 总线接口:
 - 为所有的系统接口及存储器提供简单集成的单一的32位AMBA-3 AHB-Lite 系统接口
 - 支持DAP（调试使用端口）单一的32位从机端口

6.2 系统管理

6.2.1 概述

系统管理包含以下几个部分:

- 系统复位
- 系统电源分配
- 系统内存映射
- 用于产品ID、芯片复位、片上控制器复位和多功能管脚控制的系统管理寄存器
- 系统定时器(SysTick)
- 嵌套中断向量控制器(NVIC)
- 系统控制寄存器

6.2.2 系统复位

系统复位可以由下面列出的任一事件触发。这些复位事件标志可以通过读取SYS_RSTSTS 寄存器来判断复位源。硬件复位可以通过外部的复位信号来复位芯片。软件复位则是通过控制寄存器来触发复位。

- 硬件复位源
 - 上电复位 (POR)
 - nRESET 引脚低电平复位
 - 看门狗定时器溢出复位 (WDT)
 - 低电压复位 (LVR)
 - 欠压检测复位 (BOD Reset)
- 软件复位源
 - 写1到CHIPRST位(SYS_IPRST0[0])会复位整个芯片
 - 写1到SYSRESETREQ(SCS_AIRCR[2])会触发MCU复位重启，但不影响APROM和LDROM的启动选项
 - 仅当写1到CPURST(SYS_IPRST0[1])才会触发Cortex®-M0 CPU复位

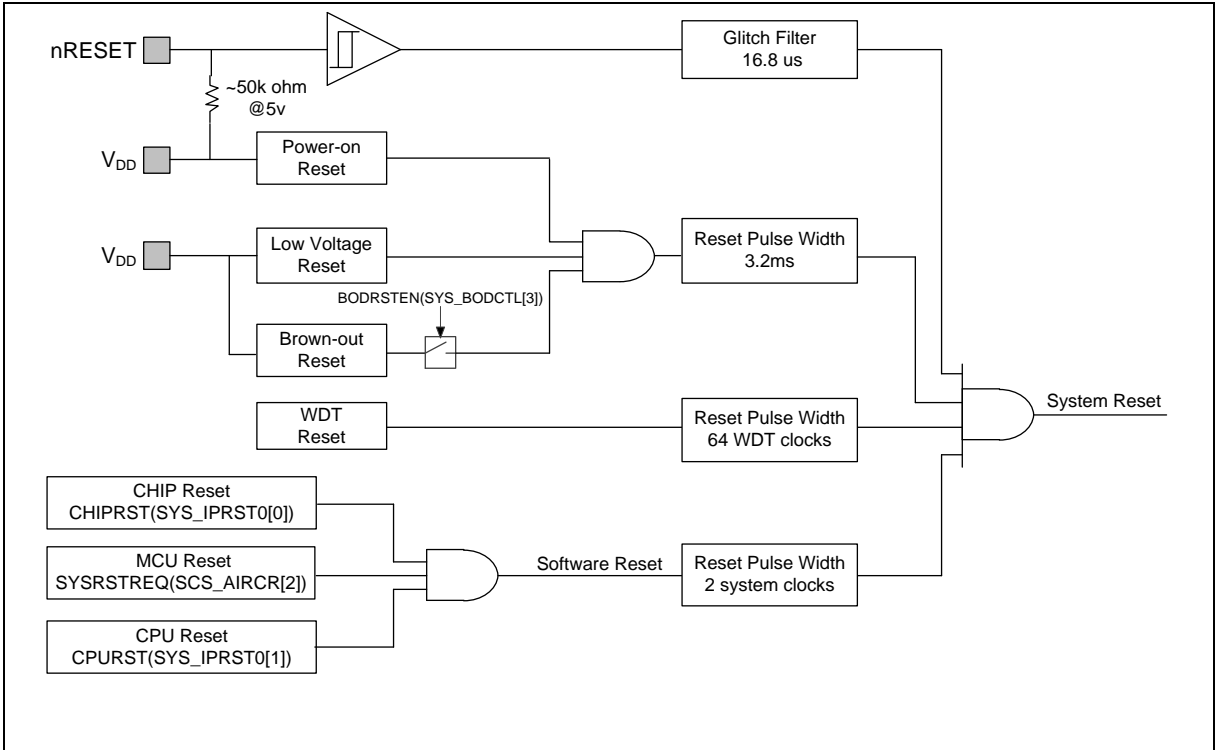


图 6.2-1 系统复位源

NuMicro®系列单片机总共有9个复位源。通常，CPU复位仅用于复位Cortex®-M0内核；其他复位源则会复位Cortex®-M0内核及所有外设。其中，每种复位之间仍有细微的差别，具体可以参见表 6.2-5。

复位源 寄存器	POR	nRESET	WDT	LVR	BOD	CHIP	MCU	CPU
SYS_RSTSTS	0x001	Bit 1 = 1	Bit 2 = 1	0x001	Bit 4 = 1	Bit 0 = 1	Bit 5 = 1	Bit 7 = 1
CHIPRST (SYS_IPRST0[0])	0x0	-	-	-	-	-	-	-
BODEN (SYS_BODCTL[0])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	Reload from CONFIG0	-
BODVL (SYS_BODCTL[2:1])								
BODRSTEN (SYS_BODCTL[3])								
XTLEN (CLK_PWRCTL[1:0])	0x0	0x0	0x0	0x0	0x0	0x0	0x0	
WDTCKEN	0x1	-	0x1	-	-	0x1	-	-

(CLK_APBCLK0[0])								
HCLKSEL (CLK_CLKSEL0[1:0])	0x8	0x8	0x8	0x8	0x8	0x8	0x8	-
WDTSEL (CLK_CLKSEL1[1:0])	0x3	0x3	-	-	-	-	-	-
XLTSTB (CLK_STATUS[0])	0x0	-	-	-	-	-	-	-
LIRCSTB (CLK_STATUS[3])	0x0							
HIRCSTB (CLK_STATUS[4])	0x0	-	-	-	-	-	-	-
CLKSFAIL (CLK_STATUS[7])	0x0	0x0	-	-	-	-	-	-
WDT_CTL	0x0700	0x0700	0x0700	0x0700	0x0700	0x0700	-	-
WDT_ALTCTL	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	-	-
WWDT_RLDCNT	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	-	-
WWDT_CTL	0x3F0800	0x3F0800	0x3F0800	0x3F0800	0x3F0800	0x3F0800	-	-
WWDT_STATUS	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	-	-
WWDT_CNT	0x3F	0x3F	0x3F	0x3F	0x3F	0x3F	-	-
BS (FMC_ISPCTL[1])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	-
ISPEN (FMC_ISPCTL[16])								
FMC_DFBA	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	-	-
CBS (FMC_ISPSTS[2:1])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	-
VECMAP (FMC_ISPSTS[20:9])	Reload base on CONFIG0	Reload base on CONFIG0	Reload base on CONFIG0	Reload base on CONFIG0	Reload base on CONFIG0	Reload base on CONFIG0	-	-
Other Peripheral Registers	Reset Value							
FMC Registers	Reset Value							
Note: '-' means that the value of register keeps original setting.								

表 6.2-1 寄存器复位值

6.2.2.1 nRESET 复位

nRESET复位是指拉低nRESET引脚产生的复位。nRESET复位是一个异步的复位，可以在任何时间用于系统复位。当nRESET引脚电压低于0.2VDD超过16.8us(消抖时间)，芯片将会复位。

nRESET复位将会保持芯片的复位状态直到nRESET引脚电压上升超过 $0.7V_{DD}$ 且时长超过 $36\mu s$ (消抖时间)。当上一个复位源是nRESET复位时，寄存器PINRF(SYS_RSTSTS[1])会被置1。如图 6.2-2 是nRESET复位的波形。

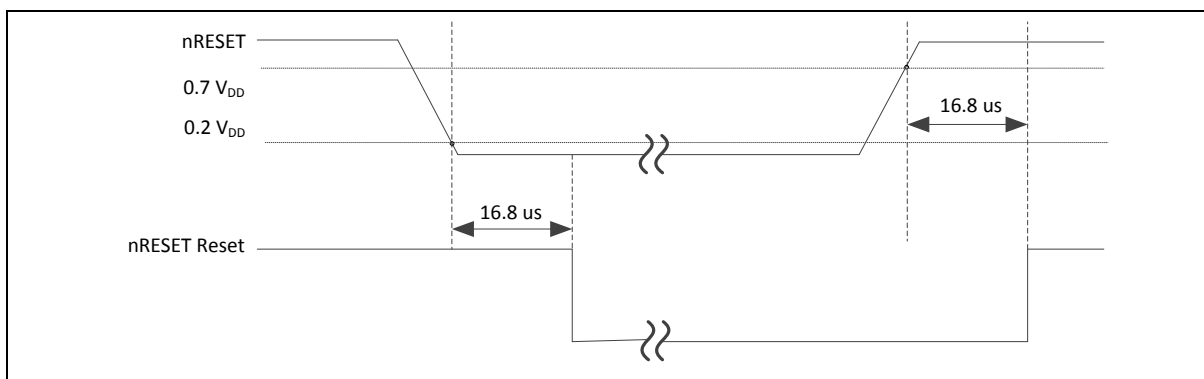


图 6.2-2 nRESET 复位波形

6.2.2.2 上电复位 (POR)

上电复位可以产生可靠的系统复位信号从而强制系统复位，以避免单片机出现异常行为。单片机上电时，POR模块会检测上升电压，当电压达到单片机工作电压时，上电复位模块会产生一个复位信号来复位系统。上电复位时，寄存器PORF(SYS_RSTSTS[0])会被置1以显示有一个上电复位事件。向寄存器PORF(SYS_RSTSTS[0])写1可以清除该位。如图 6.2-3 是上电复位时的波形。

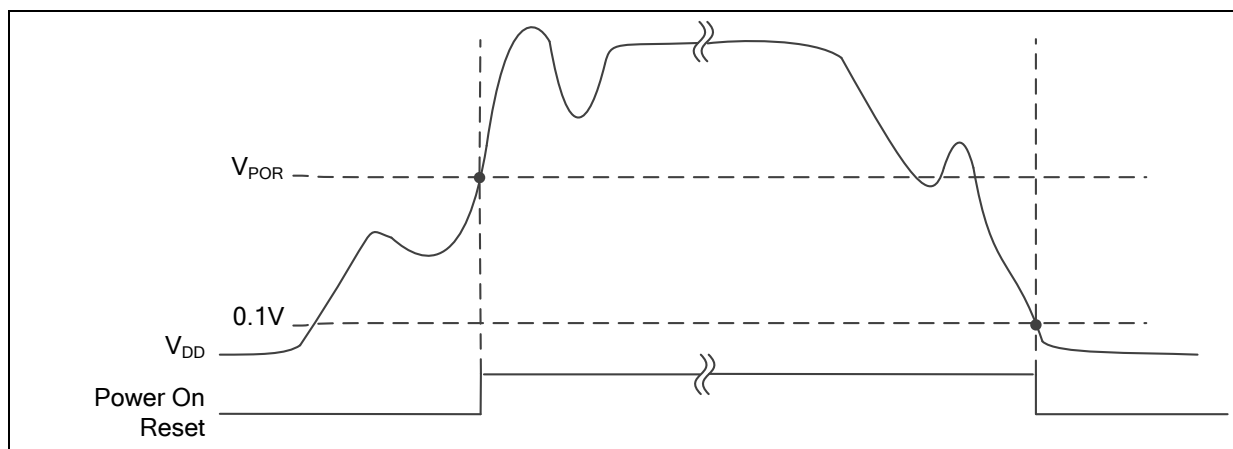


图 6.2-3 上电复位 (POR) 波形

6.2.2.3 低电复位 (LVR)

系统工作时低电复位模块监测AVDD的电压，当AVDD的电压低于VLVR且该状态持续超过消抖时长(16个系统时钟周期(HCLK))，芯片将会复位。低电复位将会保持芯片的复位状态直到AVDD的电压上升超过VLVR且该状态持续超过消抖时长。当上一个复位源是低电复位时，寄存器PINRF

(SYS_RSTSTS[1])将会被置1。如图 6.2-4 是低电复位的波形图。

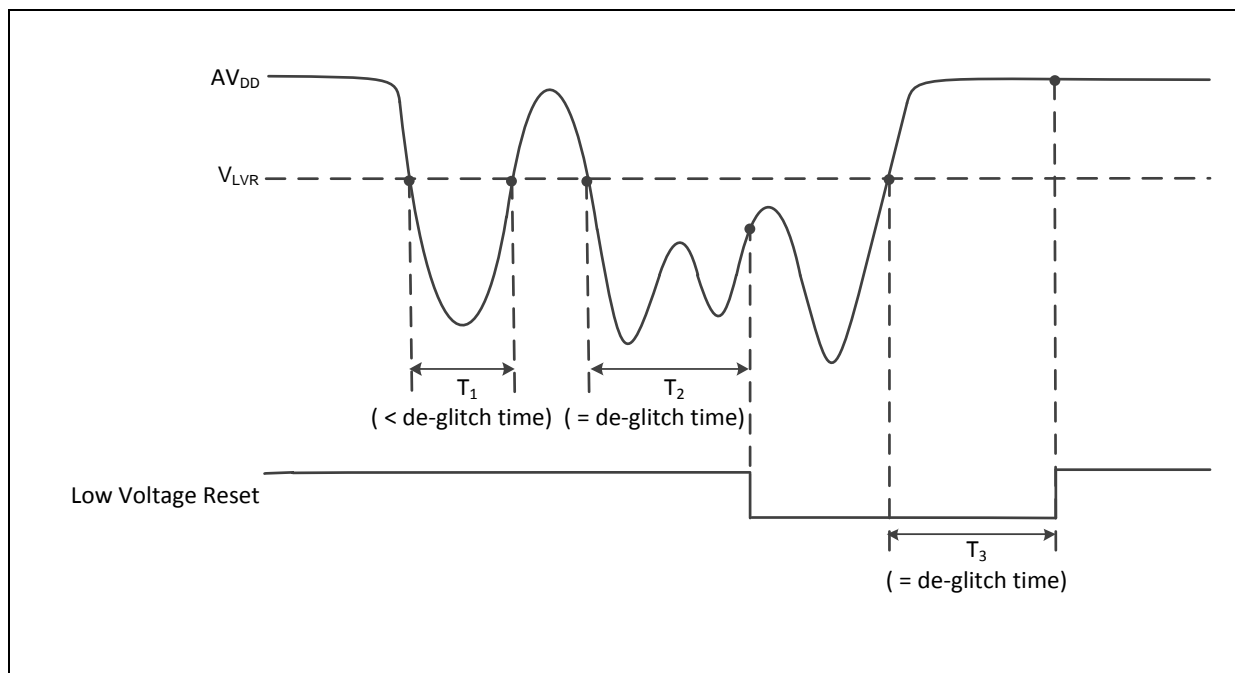


图 6.2-4 低电复位 (LVR) 波形

6.2.2.4 欠压检测复位 (BOD Reset)

设置欠电压检测使能位BODEN(SYS_BODCTL[0])打开欠电压检测功能后，系统运行过程中BOD模块会一直检测AVDD电压。当AVDD电压低于预设的VBOD[由寄存器SYS_BODCTL[0]]及BODVL(SYS_BODCTL[2:1])决定]且持续超过消抖时间(20个系统时钟(HCLK)周期时长、1个内部低速晶振(LIRC)周期时长间的最大值)，芯片会被复位。BOD复位将会保持芯片复位状态直到AVDD的电压上升超过VBOD并且该状态持续超过消抖时间。BODEN、BODVL和BODRSTEN三个寄存器的默认值分别由存储控制器里的用户配置寄存器CBOVEXT (CONFIG0[23]), CBOV (CONFIG0[22:21]) 和 CBORST (CONFIG0[20])决定。如图 6.2-5 是BOD复位的波形图。

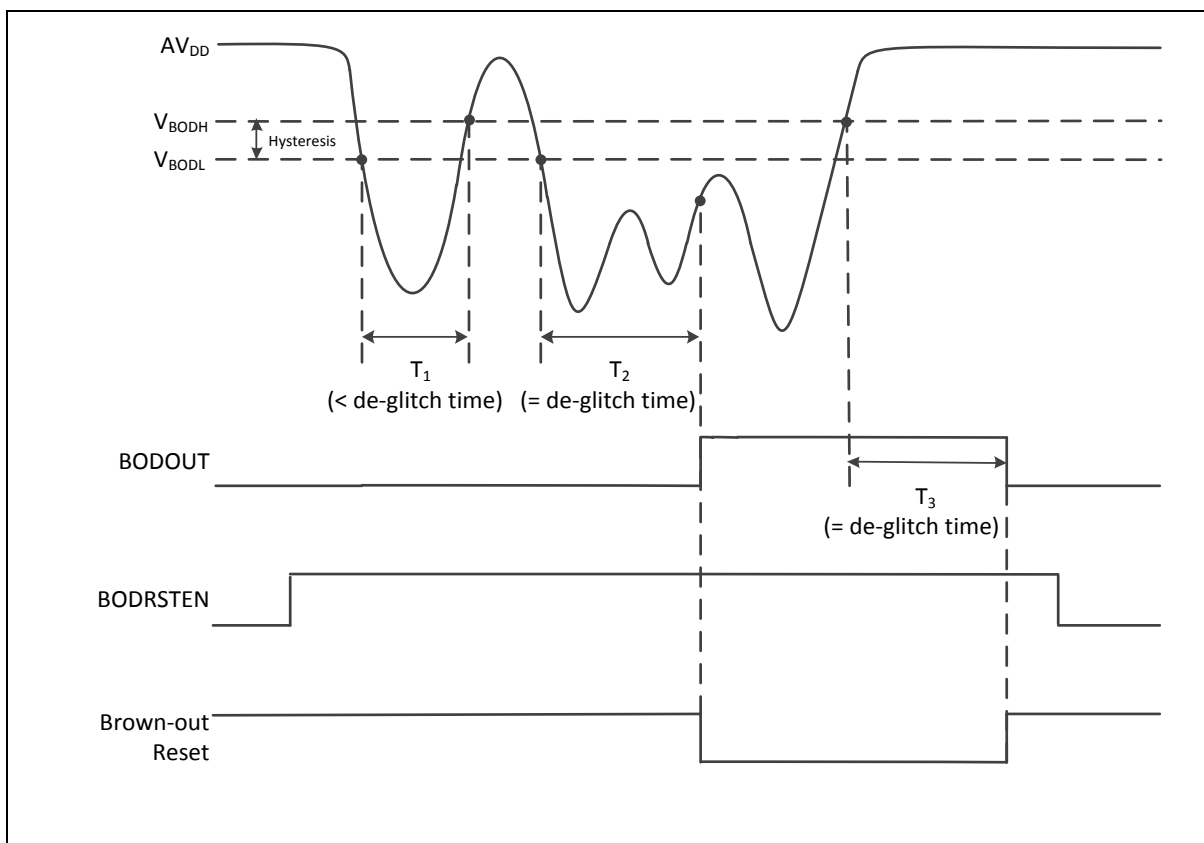


图 6.2-5 欠压检测 (BOD) 复位波形

6.2.2.5 看门狗定时器复位

在大部分工业应用中，系统可靠性是非常重要的。让单片机能从故障状态自动恢复是一个常见的提高系统可靠性的方法。看门狗定时器(WDT)被广泛用来检测系统是否正常工作。当单片机崩溃或者失控时将会引起看门狗定时器溢出。看门狗定时器溢出后，用户可以使能系统复位以使系统恢复正常，并在复位后针对之前的崩溃/失控状态做相应处理。

软件可以通过寄存器WDTRF (SYS_RSTSTS[2])来判断复位是否由看门狗定时器溢出触发，并在复位后针对单片机的错误做相应操作。

6.2.2.6 CPU 复位, 芯片复位 and 系统复位

CPU复位是指仅Cortex®-M0内核复位，单片机其他外设的状态在CPU复位之后保持不变。用户可以通过设置寄存器CPURST (SYS_IPRST0[1])位为1来触发CPU复位信号。

芯片(CHIP)复位与上电复位一致。CPU内核和所有外设都会复位，寄存器BS(FMC_ISPCTL[1])会自动装载用户配置里的值。用户可以通过设置寄存器CHIPRST (SYS_IPRST0[0])位为1来触发CHIP复位信号。

单片机(MCU)复位与芯片(CHIP)复位类似。两者的不同点是单片机(MCU)复位中寄存器BS(FMC_ISPCTL[1])不会自动装载用户配置里的值，而会保持之前软件里设定的从APROM或是从LDROM启动。用户可以通过设置寄存器SYSRESETREQ (SCS_AIRCR[2])位为1来触发单片机(MCU)复位信号。

6.2.3 电源模式和唤醒源

当单片机处于空闲模式或者掉电模式下可以由多种唤醒源来唤醒单片机。表 6.2-2 列出了不同电源模式下可用的时钟。

电源模式	正常模式	空闲模式	掉电模式
定义	CPU 处于激活的状态	CPU 处于休眠的状态	CPU处于休眠的状态。除 LXT 、 LIRC和SRAM 保留，所有时钟停止.
进入条件	系统复位完成之后芯片将会处于正常模式	CPU 执行WFI命令.	CPU使能睡眠模式和掉电模式后执行WFI指令
唤醒源	N/A	所有中断	WDT, I ² C, 定时器(Timer), UART, SPI, ACMP, BOD 和 GPIO
可用的时钟	所有时钟	除了CPU时钟外的所有时钟	外部低速晶振(LXT)和 内部低速振荡器(LIRC)
唤醒后	N/A	CPU 恢复到正常模式	CPU恢复到正常模式

表 6.2-2 电源模式差异表

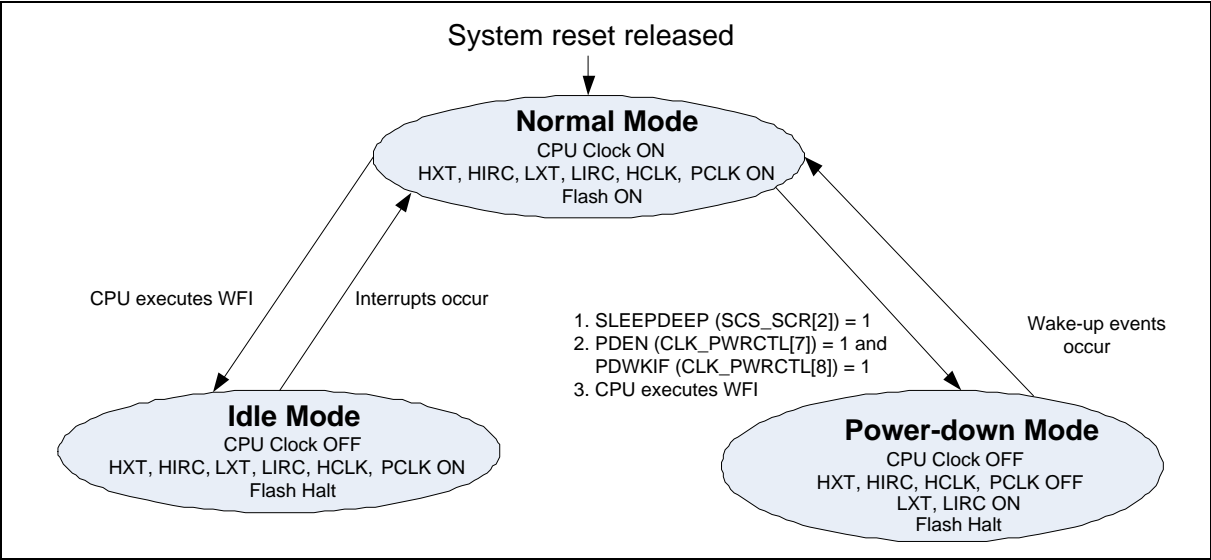


图 6.2-6 电源模式状态机

- 1. LXT (32768 Hz XTL) ON 或 OFF 取决于软件设置的运行模式.
- 2. LIRC (10 kHz OSC) ON or OFF 取决于软件设置的运行模式.
- 3.如果TIMER的时钟源是LIRC/LXT那么LIRC/LXT ON.
- 4.如果WDT的时钟源是LIRC那么LIRC ON.

	正常模式	空闲模式	掉电模式
HXT (4~20 MHz XTL)	ON	ON	Halt

HIRC (12/16 MHz OSC)	ON	ON	Halt
LXT (32768 Hz XTL)	ON	ON	ON/OFF ¹
LIRC (10 kHz OSC)	ON	ON	ON/OFF ²
LDO	ON	ON	ON
CPU	ON	Halt	Halt
HCLK/PCLK	ON	ON	Halt
SRAM retention	ON	ON	ON
FLASH	ON	ON	Halt
GPIO	ON	ON	Halt
TIMER	ON	ON	ON/OFF ³
BPWM	ON	ON	Halt
EPWM	ON	ON	Halt
WDT	ON	ON	ON/OFF ⁴
USCI	ON	ON	Halt
ADC	ON	ON	Halt
ACMP	ON	ON	Halt
ECAP	ON	ON	Halt
HDIV	ON	ON	Halt
PGA	ON	ON	Halt

表 6.2-3 不同电源模式下的时钟情况

掉电模式下的唤醒源:

WDT, I²C, Timer, UART, SPI, BOD, ACMP 及 GPIO

芯片进入掉电模式后以下的唤醒源可以唤醒芯片到正常模式。表 6.2-4列出了各外设再次进入掉电模式的条件。

*在设置 PDEN (CLK_PWRCTL[7])位和执行 WFI 以进入掉电模式前，用户需要等待这些条件完成。

唤醒源	唤醒条件	系统再次进入掉电模式的条件*
BOD	Brown-Out Detector 中断	软件写 1 清除 SYS_BODCTL[BODIF]位后
GPIO	GPIO 中断	软件写 1 清除Px_INTSRC[n] 位后
TIMER	Timer 中断	软件写 1 清除 TWKF (TIMERx_INTSTS[1]) 和 TIF (TIMERx_INTSTS[0])位后
WDT	WDT 中断	软件写 1 清除WKF (WDT_CTL[5])位后(写保护位)
USCI UART	接收数据唤醒	软件写 1 清除 WKF (UUART_WKSTS[0])位后.
USCI SPI	从机选择信号唤醒	软件写 1 清除 WKF (USPI_WKSTS[0])位后.

USCI I ² C	数据翻转	软件写 1 清除 WKF (UI2C_WKSTS[0])位后.
	地址匹配	软件写 1 清除 WKAKDONE (UI2C_PROTSTS[16]位后, 接着写 1 清除WKF (UI2C_WKSTS[0]).
ACMP	比较器掉电唤醒中断	软件写 1 清除 ACMPF0 (ACMP_STATUS[0]) 和 写1清除 ACMPF1 (ACMP_STATUS[1])位后.

表 6.2-4 外设再次进入掉电模式的条件

6.2.4 系统电源结构

本芯片的电源分为三个部分。

- 由AVDD和AVSS提供的模拟电源，为芯片模拟部分提供工作电压。AVDD 与VDD 电压值必须相等，以防止电流漏电现象。
- 由VDD和VSS提供的数字电源，为I/O管脚及用于数字运算的1.5V内部线性电源提供工作电压
- 内置一个电容用于内部稳压器

内部稳压器的输出，LDO不需要外接电容也没有外部引脚。模拟电压(AVDD)必须和数字电压(VDD)有相同的电压。

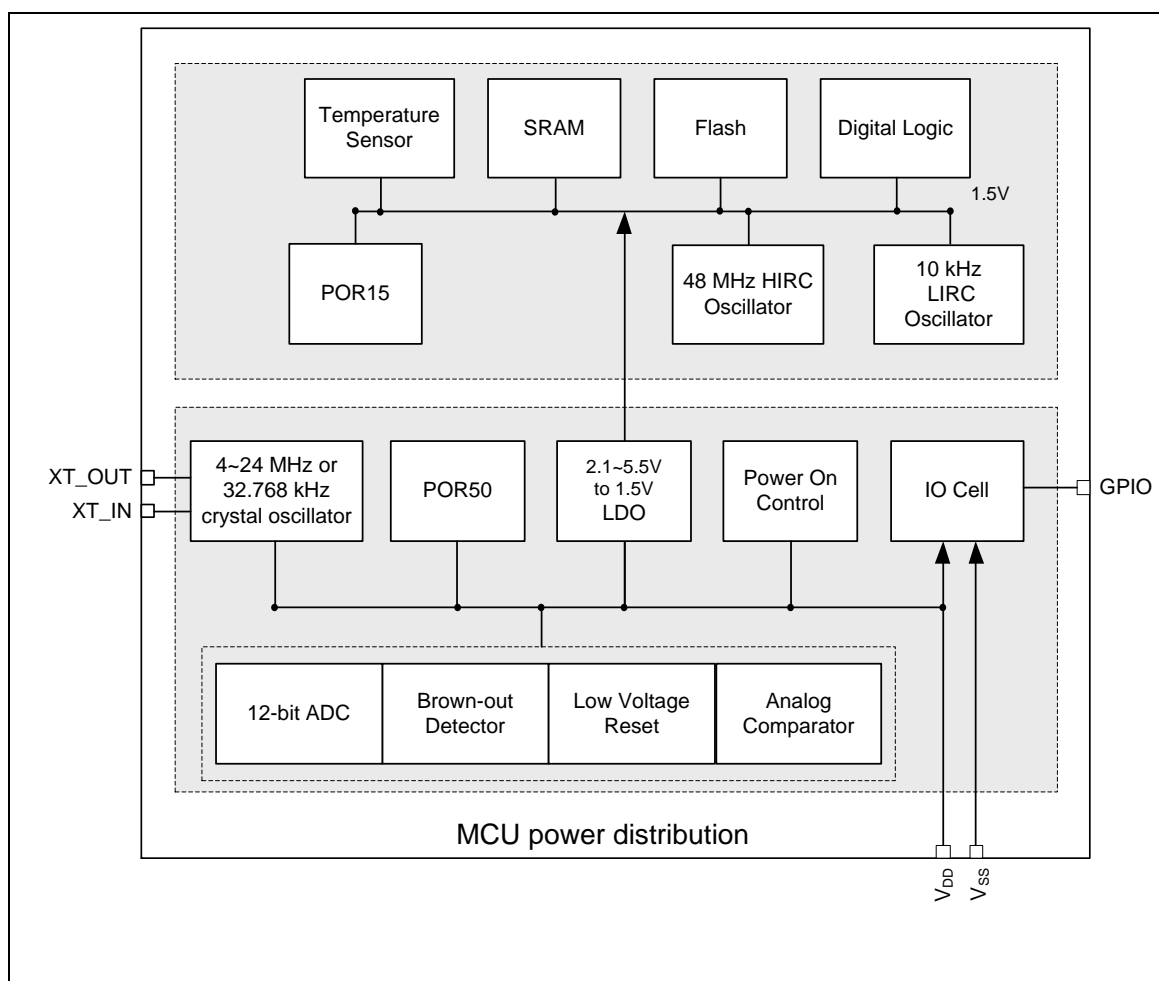


图 6.2-7 NuMicro® NM1120 系列电源架构图

6.2.5 系统内存映射

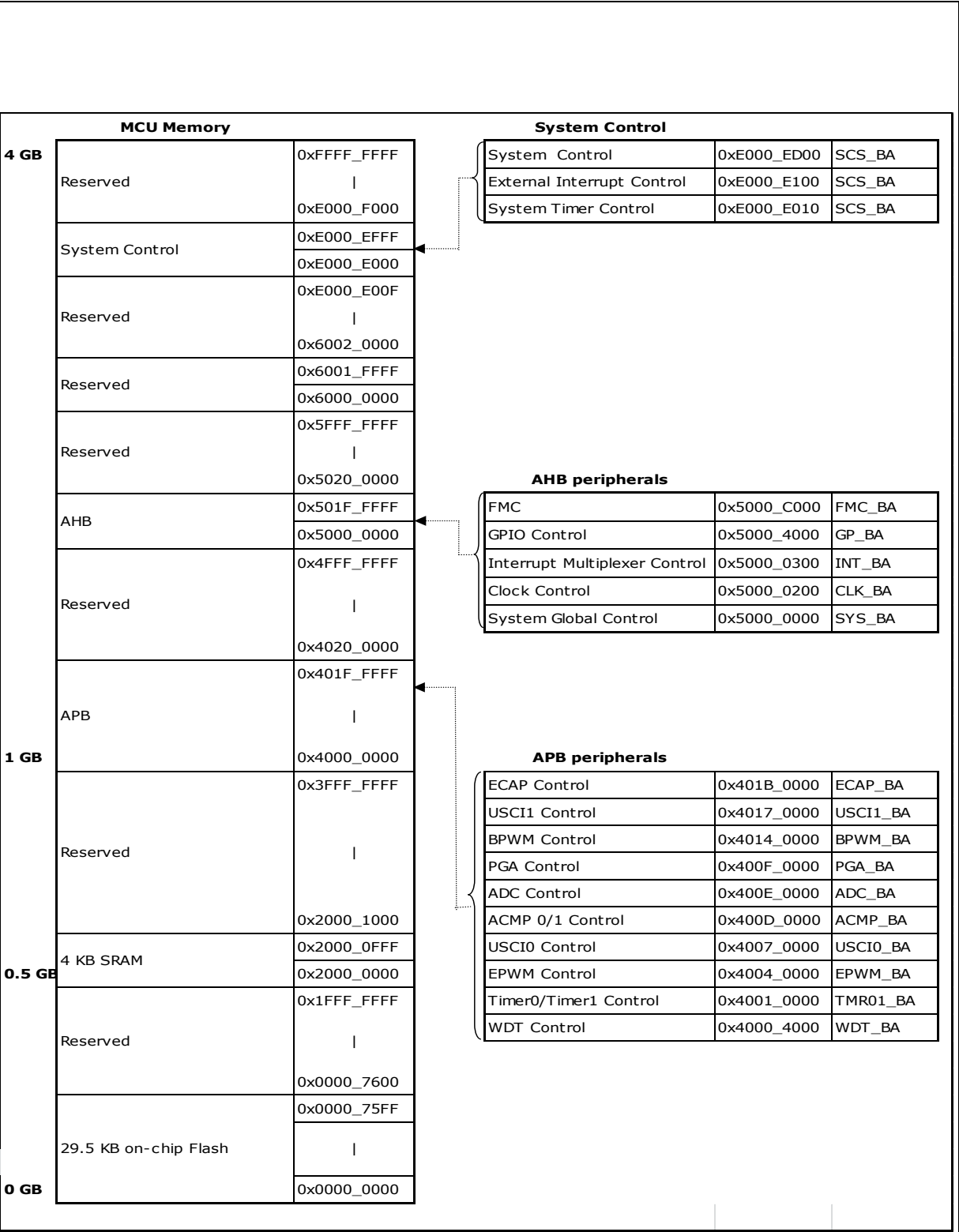


表 6.2-5 内存映射表

6.2.6 寄存器保护

部分系统控制寄存器必须被保护起来用于避免因为非人为改写造成芯片工作异常。在芯片复位后这些系统控制寄存器默认是开启保护的直到使用者关闭保护功能。用户在对这些保护寄存器编程之前需通过特殊的指令来解锁。这些特殊的解锁指令是连续写“59h”，“16h”“88h”到寄存器SYS_REGLCTL。写入其他的任何数据或者按照不同的顺序来写入以及写到其它的地址都会终止解锁。

在解锁之后，用户可以通过读取寄存器REGLCTL (SYS_REGLCTL [0])来检查是否解锁。“1”代表已解锁，“0”代表未解锁。这个时候用户可以更新对应保护寄存器的值，最后写任意数据到寄存器REGLCTL (SYS_REGLCTL [0])以开启保护功能。

表 6.2-6是保护寄存列表

寄存器	位	描述
SYS_IPRST0	[1] CPURST	CPU复位 (写保护)
	[0] CHIPRST	芯片复位 (写保护)
SYS_BODCTL	[15] LVREN	低电压复位使能 (写保护)
	[6] BODLPM	欠压检测低功耗模式 (写保护)
	[4] BODRSTEN	欠压复位使能 (写保护)
	[3:1] BODVL	欠压检测门限值选择 (写保护)
	[0] BODEN	欠压检测使能 (写保护)
SYS_PORCTL	[15:0] POROFF	上电复位使能控制 (写保护)
INT_NMICTL	[8] NMISELEN	不可屏蔽中断使能控制 (写保护)
CLK_PWRCTL	[11:10] HXTGAIN	外部高速晶振增益控制 (写保护)
	[7] PDEN	系统掉电休眠使能 (写保护)
	[5] PDWKIEN	掉电模式唤醒中断使能 (写保护)
	[4] PDWKDLY	唤醒延时计数器使能控制 (写保护)
	[3] LIRCEN	内部低速RC振荡器使能控制 (写保护)
	[2] HIRCEN	内部高速RC振荡器使能控制 (写保护)
	[1:0] XTLEN	外部晶振使能控制 (写保护)
CLK_APBCLK	[0] WDTCKEN	看门狗定时器时钟使能控制 (写保护)
CLK_CLKSEL0	[4:3] STCLKSEL	Cortex®-M0 系统钟摆时钟源选择 (写保护)
	[1:0] HCLKSEL	高速时钟源选择 (写保护)
CLK_CLKSEL1	[1:0] WDTSEL	看门狗定时器时钟源选择 (写保护)
FMC_ISPCTL	[6] ISPPFF	ISP 失败标志位 (写保护)
	[5] LDUEN	LDROM 更新使能控制 (写保护)

	[4] CFGUEN	配置为更新使能控制 (写保护)
	[3] APUEN	APROM 更新使能控制 (写保护)
	[2] SPUEN	SPROM 更新使能控制 (写保护)
	[1] BS	启动选择 (写保护)
	[0] ISPEN	ISP 使能控制 (写保护)
FMC_ISPTRG	[0] ISPGO	ISP 开始触发 (写保护)
FMC_ISPSTS	[6] ISPPF	ISP 失败标记位 (写保护)
TIMER0_CTL	[31] ICEDEBUG	ICE 调试模式握手禁止控制 (写保护)
TIMER1_CTL	[31] ICEDEBUG	ICE 调试模式握手禁止控制 (写保护)
WDT_CTL	[31] ICEDEBUG	ICE 调试模式握手禁止控制 (写保护)
	[7] WDTEN	看门狗使能控制 (写保护)
	[6] INTEN	看门狗超时中断使能控制 (写保护)
	[4] WKEN	看门狗超时唤醒功能控制 (写保护)
	[1] RSTEN	看门狗超时复位使能控制 (写保护)
	[0] RSTCNT	复位看门狗定时计数器 (写保护)

表 6.2-6 保护寄存器

6.2.7.1 概述

[illegible]图 6.2-8 NuMicro[®] NM1120 Flash, Security and Configuration 映射图

NuMicro® NM1120 系列提供4GB地址空间，分配给片上控制器的地址空间如图6.2-7所示。每个片上外设详细的寄存器定义、地址空间和编程细节将在接下来的章节中描述。NM1120系列只支持小端数据格式。

分配给片上控制器的地址空间如图6.2-7所示

地址空间	Token	控制器
Flash and SRAM Memory Space		
0x0000_0000 – 0x0000_75FF	FLASH_BA	FLASH 存储空间 (29.5KB)
0x0010_0000 – 0x0010_07FF	LD_BA	引导存储器空间 (2 KB)
0x0020_0000 – 0x0020_01FF	SP0_BA	安全保护存储器空间0 (0.5 KB)
0x0024_0000 – 0x0024_01FF	SP1_BA	安全保护存储器空间1 (0.5 KB)
0x0028_0000 – 0x0028_01FF	SP2_BA	安全保护存储器空间2 (0.5 KB)
0x2000_0000 – 0x2000_0FFF	SRAM_BA	SRAM 存储器空间 (4 KB)
AHB Modules Space (0x5000_0000 – 0x501F_FFFF)		
0x5000_0000 – 0x5000_01FF	SYS_BA	系统控制寄存器
0x5000_0200 – 0x5000_02FF	CLK_BA	时钟控制寄存器
0x5000_0300 – 0x5000_03FF	INT_BA	中断多路选择控制寄存器
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	硬件除法器控制寄存器
APB Controllers Space (0x4000_0000 ~ 0x401F_FFFF)		
0x4000_4000 – 0x4000_7FFF	WDT_BA	看门狗定时器控制寄存器
0x4001_0000 – 0x4001_3FFF	TMR01_BA	Timer0/Timer1 控制寄存器
0x4004_0000 – 0x4004_3FFF	EPWM_BA	增强型PWM 控制寄存器
0x4007_0000 – 0x4007_3FFF	USCI0_BA	USCI0 控制寄存器
0x400D_0000 – 0x400D_3FFF	ACMP_BA	模拟比较器 0/1 控制寄存器
0x400E_0000 – 0x400E_3FFF	ADC_BA	ADC 控制寄存器
0x400F_0000 – 0x400F_3FFF	PGA_BA	可编程增益放大器控制寄存器
0x4014_0000 – 0x4014_3FFF	BPWM_BA	基础型 PWM 控制寄存器
0x4017_0000 – 0x4017_3FFF	USCI1_BA	USCI1 控制寄存器
0x401B_0000 – 0x401B_3FFF	ECAP_BA	增强型输入捕捉定时计数器
System Controllers Space (0xE000_E000 ~ 0xE000_EFFF)		
0xE000_E010 – 0xE000_E0FF	SCS_BA	系统钟摆定时器控制寄存器
0xE000_E100 – 0xE000_ECFF	SCS_BA	外部中断控制寄存器
0xE000_ED00 – 0xE000_ED8F	SCS_BA	系统控制寄存器

表 6.2-7 片上模块地址空间分配

6.2.7.3 SRAM 存储器组织

NM1120支持4KB的嵌入式SRAM。

- 支持 4KB 的嵌入式 SRAM
- 支持 字节/双字节/四字节 写操作
- 支持超长度错误回馈

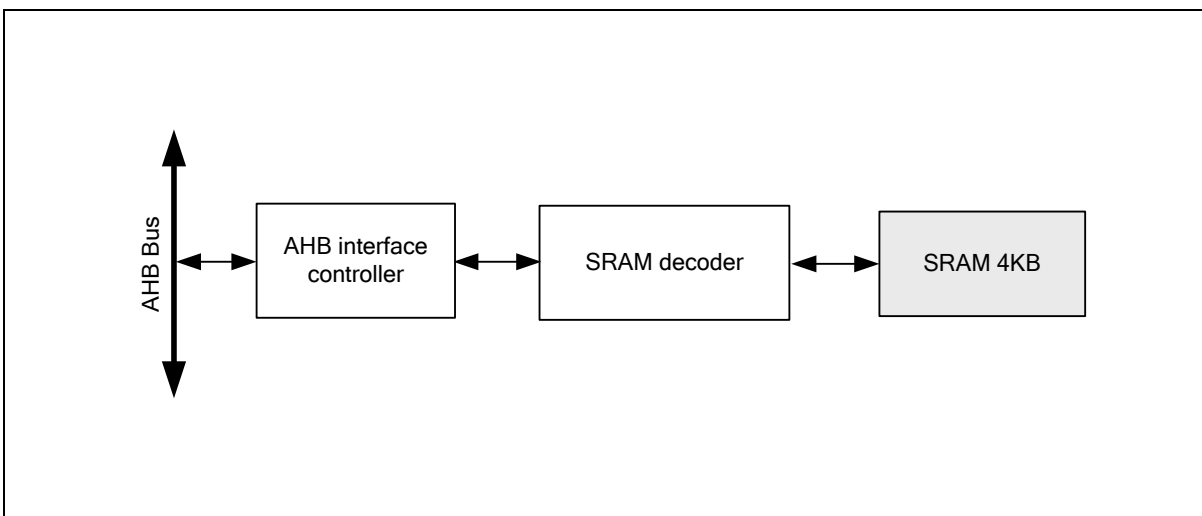


图 6.2-9 SRAM 方框图

6.2.8 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
SYS 基地址: SYS_BA = 0x5000_0000				
SYS_PDID	SYS_BA+0x00	R	器件ID寄存器	0XXXXX_XXXX ^[1]
SYS_RSTSTS	SYS_BA+0x04	R/W	系统复位状态寄存器	0x0000_00XX
SYS_IPRST0	SYS_BA+0x08	R/W	外设复位控制寄存器0	0x0000_0000
SYS_IPRST1	SYS_BA+0x0C	R/W	外设复位控制寄存器1	0x0000_0000
SYS_WAIT	SYS_BA+0x10	R/W	高速时钟等待状况循环控制寄存器	0x0000_0001
SYS_BODCTL	SYS_BA+0x18	R/W	掉电检测控制寄存器	0x0000_80XX
SYS_IVSCTL	SYS_BA+0x1C	R/W	内部电压源控制寄存器	0x0000_0000
SYS_PORCTL	SYS_BA+0x24	R/W	上电复位控制寄存器	0x0000_00XX
SYS_GPA_MFP	SYS_BA+0x30	R/W	GPIOA 复用功能和输入类型控制寄存器	0x0000_0000
SYS_GPB_MFP	SYS_BA+0x34	R/W	GPIOB 复用功能和输入类型控制寄存器	0x0000_0000
SYS_GPC_MFP	SYS_BA+0x38	R/W	GPIOC 复用功能和输入类型控制寄存器	0x0000_0000
SYS_GPD_MFP	SYS_BA+0x3C	R/W	GPIOD 复用功能和输入类型控制寄存器	0x0000_0111
SYS_IRCTCTL	SYS_BA+0x80	R/W	HIRC校准控制寄存器	0x0000_0030
SYS_IRCTIEN	SYS_BA+0x84	R/W	HIRC校准中断使能寄存器	0x0000_0000
SYS_IRCTISTS	SYS_BA+0x88	R/W	HIRC校准中断状态寄存器	0x0000_0000
SYS_REGLCTL	SYS_BA+0x100	R/W	寄存器写保护控制寄存器	0x0000_0000
SYS_TSOFFSET	SYS_BA+0x114	R	温度传感器偏移寄存器	0x0XXX_0XXX

6.2.9 寄存器描述

器件ID寄存器 (SYS_PDID)

寄存器	偏移地址	R/W	描述	复位值
SYS_PDID	SYS_BA+0x00	R	器件ID寄存器	0xFFFF_FFFF ^[1]

[1]每个器件具有一个独一无二的默认复位值

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 (只读) 该寄存器为设备的器件号码。软件可以通过读取该寄存器来识别所使用的器件。

NuMicro® NM1120 系列	设备器件识别码
NM1120FC1AE	0x00B01420
NM1120EC1AE	0x00B01440
NM1120XC1AE	0x00B01450
NM1120ZC1AE	0x00B01470

表 6.2-1 设备器件识别码

系统复位状态寄存器 (SYS_RSTSTS)

该寄存器提供一些信息用于软件识别上次引起芯片复位的复位源。

寄存器	偏移地址	R/W	描述	复位值
SYS_RSTSTS	SYS_BA+0x04	R/W	系统复位状态寄存器	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
CPURF	Reserved	SYSRF	BODRF	LVRF	WDTRF	PINRF	PORF

位	描述	
[31:10]	Reserved	Reserved.
[9:8]	Reserved	Reserved.
[7]	CPURF	CPU复位标志 如果软件写1到CPURST (SYS_IPRST0[1]) 来复位Cortex®-M0内核和Flash内存控制器(FMC)，硬件会自动置位该CPU复位标志位。 0 = CPU无复位 1 = Cortex®-M0 内核和FMC因为软件置CPURS为1而复位。 注: 软件可向该位写1来清0。
[6]	Reserved	保留
[5]	SYSRF	CPU复位标志 The system reset flag is set by the “Reset Signal” from the Cortex®-M0 Core to indicate the previous reset source. 如果软件写1到CPURST (SYS_IPRST0[1]) 来复位Cortex®-M0内核和Flash内存控制器(FMC)，硬件会自动置位该CPU复位标志位。 0 = CPU无复位 1 = Cortex®-M0 内核和FMC因为软件置CPURS为1而复位。 注: 软件可向该位写1来清0。
[4]	BODRF	BOD复位标志 BOD复位标志由欠压检测模块的“复位信号”置位，用于表示导致之前复位的信号源。 0 = BOD无复位。 1 = BOD发出复位信号使系统复位。 注: 软件可向该位写1来清0。

位	描述	
[3]	LVRF	<p>LVR Reset Flag 低压复位标志</p> <p>LVR低压复位标志位受低压复位控制器的控制用于记录上一次的复位源。</p> <p>0 = LVR无复位</p> <p>1 = LVR控制器已经产生复位信号使系统复位</p> <p>注：写1将此位清零</p>
[2]	WDTRF	<p>WDT复位标志</p> <p>WDT标志由看门狗定时器或窗口看门狗定时器的“复位信号”置位，用于表示导致之前复位的复位源。</p> <p>0 = 看门狗定时器或窗口看门狗定时器无复位。</p> <p>1 = 看门狗定时器或窗口看门狗定时器发出复位信号使系统复位。</p> <p>注1：软件可向该位写1来清0。</p> <p>注2：如果系统被WDT看门狗定时器溢出信号复位，看门狗定时寄存器RSTF(WDT_CTL[2])将会被置1.如果系统被WWDT窗口看门狗定时器溢出信号复位，窗口看门狗定时寄存器WWDT_STATUS[1]将会被置1.</p>
[1]	PINRF	<p>nRESET管脚复位标志</p> <p>nRESET管脚复位标志由nRESET管脚的“复位信号”置位，用于表示导致之前复位的复位源。</p> <p>0 = nRESET管脚无复位</p> <p>1 = nRESET管脚发出复位信号使系统复位</p> <p>注：软件可向该位写1来清0。</p>
[0]	PORF	<p>POR复位标志</p> <p>POR复位标志由上电复位 (POR) 控制器或CHIPRST (SYS_IPRST0[0])位发出“复位信号”置位，用于表示导致之前复位的复位源。</p> <p>0 = POR或CHIPRST无复位</p> <p>1 = 上电复位(POR)或CHIPRST发出复位信号使系统复位</p> <p>注：软件可向该位写1来清0。</p>

外设复位控制寄存器 0 (SYS_IPRST0)

寄存器	偏移地址	R/W	描述	复位值
SYS_IPRST0	SYS_BA+0x08	R/W	外设复位控制寄存器 0	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						CPURST	CHIPRST

位	描述	
[31:2]	Reserved	保留
[1]	CPURST	<p>处理器内核一次性复位(写保护)</p> <p>置该位为1将只复位处理器内核和Flash内存控制器(FMC)，该位将在两个时钟周期后自动清零。</p> <p>0 = 处理器内核正常运行</p> <p>1 = 处理器内核一次性复位</p> <p>注: 该位为写保护位，请参考SYS_REGLCTL寄存器。</p>
[0]	CHIPRST	<p>芯片一次性复位 (写保护)</p> <p>置该位为1将复位整个芯片，包括处理器内核和所有外设，该位将在2个时钟周期后自动清零。</p> <p>CHIPRST与POR复位一样，所有芯片控制器都复位，芯片设置从flash重新加载。</p> <p>有关CHIPRST 和 SYSRESETREQ(AIRCR[2])的差异，请参考6.2.2章节。</p> <p>0 = 芯片正常运行</p> <p>1 = 芯片一次性复位</p> <p>注: 该位为写保护位，请参考SYS_REGLCTL寄存器。</p>

外设复位控制寄存器 1 (SYS_IPRST1)

设置这些位为1将产生异步复位信号给相应的模块控制器，用户需要设置这些位为0，才能将相应模块控制器从复位状态中释放。

寄存器	偏移地址	R/W	描述	复位值
SYS_IPRST1	SYS_BA+0x0C	R/W	外设复位控制寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	ACMPRST	Reserved	ADCRST	Reserved		USCI1RST	USCI0RST
23	22	21	20	19	18	17	16
Reserved		EPWMRST		Reserved		BPWMRST	
15	14	13	12	11	10	9	8
Reserved			PGARST	Reserved		CAPRST	
7	6	5	4	3	2	1	0
Reserved				TMR1RST	TMR0RST	GPIORST	Reserved

位	描述	
[31]	Reserved	保留.
[30]	ACMPRST	ACMP 控制器复位 0 = ACMP 控制器正常运行. 1 = ACMP 控制器复位.
[29]	Reserved	保留.
[28]	ADCRST	ADC 控制器复位 0 = ADC 控制器正常运行. 1 = ADC 控制器复位.
[27:26]	Reserved	保留.
[25]	USCI1RST	USCI1 控制器复位 0 = USCI1 控制器正常运行. 1 = USCI1 控制器复位.
[24]	USCI0RST	USCI0 控制器复位 0 = USCI0控制器正常运行. 1 = USCI0控制器复位.
[23:21]	Reserved	保留.
[20]	EPWMRST	增强型PWM控制器复位 0 = EPWM控制器正常运行. 1 = EPWM控制器复位.

[19:17]	Reserved	保留.
[16]	BPWMRST	基础型 PWM 控制器复位 0 = BPWM控制器正常运行. 1 = BPWM控制器复位.
[15:13]	Reserved	保留.
[12]	PGARST	PGA控制器复位 0 = PGA控制器正常运行. 1 = PGA控制器复位.
[11:9]	Reserved	保留.
[8]	CAPRST	ECAP控制器复位 0 = ECAP控制器正常运行. 1 = ECAP控制器复位.
[7:4]	Reserved	保留.
[3]	TMR1RST	Timer1控制器复位 0 = Timer1控制器正常运行. 1 = Timer1控制器复位.
[2]	TMR0RST	Timer0控制器复位 0 = Timer0控制器正常运行. 1 = Timer0控制器复位.
[1]	GPIORST	GPIO控制器复位 0 = GPIO控制器正常运行. 1 = GPIO控制器复位.
[0]	Reserved	保留.

HCLK（高速时钟）等待状况循环控制寄存器 (SYS_WAIT)

寄存器	偏移地址	R/W	描述	复位值
SYS_WAIT	SYS_BA+0x10	R/W	HCLK（高速时钟）等待状况循环控制寄存器	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							HCLKWS

位	描述	
[31:1]	Reserved	保留.
[0]	HCLKWS	<p>HCLK（高速时钟）等待状况循环控制位</p> <p>这一位用来开启或关闭在读写Flash时HCLK（高速时钟）是否做等待延时处理。</p> <p>0 = 不需要等待.</p> <p>1 = 在读写Flash时需等待一个时钟周期.</p> <p>注：强烈建议在HCLK（高速时钟）频率高于48MHz时开启等待延时功能。</p>

欠压检测器控制寄存器 (SYS_BODCTL)

SYS_BODCTL控制寄存器部分位的值由flash配置初始化，部分位具有写保护功能。

寄存器	偏移地址	R/W	描述	复位值
SYS_BODCTL	SYS_BA+0x18	R/W	欠压检测器控制寄存器	0x0000_80XX

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
LVREN	Reserved						
7	6	5	4	3	2	1	0
BODOUT	BODLPM	BODIF	BODRSTEN	BODVL			BODEN

位	描述	
[31:16]	Reserved	Reserved.
[15]	LVREN	<p>低压复位使能位 (写保护)</p> <p>当电源电压低于LVR (低压复位) 电路设定值时LVR (低压复位) 功能将会使系统复位。</p> <p>LVR (低压复位) 功能默认是使能的。</p> <p>0 = 关闭低压复位功能。</p> <p>1 = 打开低压复位功能。</p> <p>注: 这是一个写保护位, 请参考SYS_REGLCTL寄存器</p>
[14:8]	Reserved	保留.
[7]	BODOUT	<p>欠压检测器输出状态</p> <p>0 = 欠压检测器输出状态为0.表示检测到的电压比BODVL设定的值高或者BODEN设置成0</p> <p>1 = 欠压检测器输出状态为1.表示检测到的电压比BODVL设定的值低。如果BODEN设定成0, 并且BOD功能关闭时, 这一位将总是读为0000.</p>
[6]	BODLPM	<p>欠压检测器低功耗模式 (写保护)</p> <p>0 = BOD运行在正常模式 (默认)</p> <p>1 = BOD低功耗模式使能</p> <p>注1: BOD在正常模式消耗差不多100uA电流, 低功耗模式将大约减少电流到1/10, 但将减慢BOD的响应速度。</p> <p>注2: 这是一个写保护位, 请参考SYS_REGLCTL寄存器.</p>

位	描述	
[5]	BODIF	<p>欠压检测器中断标志位</p> <p>0 = 欠压检测器没有检测到VDD 曾经下降或上升到BODVL设定的值</p> <p>1 = 当检测到VDD 下降或者上升到BODVL 设定的值，该位将置1，如果欠压中断使能，则欠压中断将发生</p> <p>注：写1可将此位清零</p>
[4]	BODRSTEN	<p>欠压复位使能位（写保护）</p> <p>用户可以通过Flash控制器修改配置寄存器的CBORST(CONFIG0[12])位来设定默认值。</p> <p>0 = 欠压“中断”功能使能。</p> <p>1 = 欠压“复位”功能使能。</p> <p>注1：</p> <p>当使能欠压检测器功能（BODEN为高）并且使能BOD复位功能（BODRSTEN为高），此时当检测电压比门限电压（BODOUT为高）低时BOD将会产生复位信号使芯片复位。</p> <p>当使能BOD功能（BODEN为高）并且使能BOD中断功能（BODRSTEN为低），如果BODOUT为高那么将会产生BOD中断。BOD将会一直保持中断状态直到BODEN被清零。</p> <p>关闭BOD中断可通过NVIC关闭BOD中断或者关闭BOD功能（BODEN设为低）来实现。在进入掉电模式BODOUT输出为高时BOD将会使CPU唤醒。</p> <p>注2：此位是写保护位，详情请参考SYS_REGLCTL寄存器。</p>
[3:1]	BODVL	<p>欠压检测器门限电压选择 (写保护)</p> <p>用户可以通过Flash控制器修改配置寄存器CBOV (CONFIG0 [15:13])来设定默认值。</p> <p>000 = 欠压检测器门限电压为2.0V。</p> <p>001 = 欠压检测器门限电压为2.2V。</p> <p>010 = 欠压检测器门限电压为2.4V。</p> <p>011 = 欠压检测器门限电压为2.7V。</p> <p>100 = 欠压检测器门限电压为3.0V。</p> <p>101 = 欠压检测器门限电压为3.7V。</p> <p>110 = 欠压检测器门限电压为4.0V。</p> <p>111 = 欠压检测器门限电压为4.3V。</p> <p>注：此位是写保护位，详情请参考SYS_REGLCTL寄存器。</p>
[0]	BODEN	<p>欠压检测器使能位（写保护）</p> <p>用户可以通过Flash控制器修改配置寄存器CBODEN (CONFIG0 [12])来设定默认值。</p> <p>0 = 关闭欠压检测器功能</p> <p>1 = 打开欠压检测器功能</p> <p>注：此位是写保护位，详情请参考SYS_REGLCTL寄存器。</p>

内置参考电压源控制寄存器 (SYS_IVSCTL)

寄存器	偏移地址	R/W	描述	复位值
SYS_IVSCTL	SYS_BA+0x1C	R/W	内置参考电压源控制寄存器	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							VTEMPEN

位	描述	
[31:1]	Reserved	保留。
[0]	VTEMPEN	<p>温度传感器使能位</p> <p>这一位用于打开或关闭温度传感器功能。</p> <p>0 = 关闭温度传感器功能（默认值）。</p> <p>1 = 打开温度传感器功能。</p> <p>注：将这位置1后，可通过A/D转换的结果来获取温度传感器输出的温度值。详情请参考ADC功能章节。</p>

上电复位控制寄存器 (SYS_PORCTL)

寄存器	偏移地址	R/W	描述	复位值
SYS_PORCTL	SYS_BA+0x24	R/W	上电复位控制寄存器	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
POROFF							
7	6	5	4	3	2	1	0
POROFF							

位	描述	
[31:16]	Reserved	保留.
[15:0]	POROFF	<p>上电复位使能位（写保护）</p> <p>当系统上电时，内部的POR（上电复位）电路将会产生一个复位信号使整个芯片复位，但是电源上的噪声可能会使芯片多次进入POR（上电复位）状态。用户可通过对此区域写0x5AA5来关闭内部POR（上电复位）电路，这样可避免不可预料的噪声导致芯片复位。</p> <p>对此区域写入其他的任意值可重新打开POR（上电复位）功能。或者芯片被以下的其他复位源复位：</p> <p>nRESET（PIN脚外部复位），看门狗复位，低压复位，欠压复位，ICE（调试）复位命令，软件芯片复位。</p> <p>注：此位是写保护位，详情请参考SYS_REGLCTL寄存器。</p>

GPIOA 复用功能控制寄存器 (SYS_GPA_MFP)

请参考 GPIO 管脚复用功能列表

寄存器	偏移地址	R/W	描述	复位值
SYS_GPA_MFP	SYS_BA+0x30	R/W	GPIOA 复用功能控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
PA5MFP				PA4MFP			
15	14	13	12	11	10	9	8
PA3MFP				PA2MFP			
7	6	5	4	3	2	1	0
PA1MFP				PA0MFP			

位	描述	
[31:24]	Reserved	保留.
[23:20]	PA5MFP	PA.5 管脚复用功能选择
[19:16]	PA4MFP	PA.4 管脚复用功能选择
[15:12]	PA3MFP	PA.3 管脚复用功能选择
[11:8]	PA2MFP	PA.2 管脚复用功能选择
[7:4]	PA1MFP	PA.1 管脚复用功能选择
[3:0]	PA0MFP	PA.0 管脚复用功能选择

GPIOB 复用功能控制寄存器 (SYS_GPB_MFP)

请参考GPIO管脚复用功能列表

寄存器	偏移地址	R/W	描述	复位值
SYS_GPB_MFP	SYS_BA+0x34	R/W	GPIOB 复用功能控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				PB4MFP			
15	14	13	12	11	10	9	8
PB3MFP				PB2MFP			
7	6	5	4	3	2	1	0
PB1MFP				PB0MFP			

位	描述	
[31:20]	Reserved	保留.
[19:16]	PB4MFP	PB.4 管脚复用功能选择
[15:12]	PB3MFP	PB.3 管脚复用功能选择
[11:8]	PB2MFP	PB.2 管脚复用功能选择
[7:4]	PB1MFP	PB.1 管脚复用功能选择
[3:0]	PB0MFP	PB.0 管脚复用功能选择

GPIOC 复用功能控制寄存器 (SYS_GPC_MFP)

请参考GPIO管脚复用功能列表

寄存器	偏移地址	R/W	描述	复位值
SYS_GPC_MFP	SYS_BA+0x38	R/W	GPIOC 复用功能控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				PC4MFP			
15	14	13	12	11	10	9	8
PC3MFP				PC2MFP			
7	6	5	4	3	2	1	0
PC1MFP				PC0MFP			

位	描述	
[31:20]	Reserved	保留.
[19:16]	PC4MFP	PC.4 管脚复用功能选择
[15:12]	PC3MFP	PC.3 管脚复用功能选择
[11:8]	PC2MFP	PC.2 管脚复用功能选择
[7:4]	PC1MFP	PC.1 管脚复用功能选择
[3:0]	PC0MFP	PC.0 管脚复用功能选择

GPIO 复用功能控制寄存器 (SYS_GPD_MFP)

请参考 GPIO 管脚复用功能列表

寄存器	偏移地址	R/W	描述	复位值
SYS_GPD_MFP	SYS_BA+0x3C	R/W	GPIO 复用功能控制寄存器	0x0000_0111

31	30	29	28	27	26	25	24
Reserved				PD6MFP			
23	22	21	20	19	18	17	16
PD5MFP				PD4MFP			
15	14	13	12	11	10	9	8
PD3MFP				PD2MFP			
7	6	5	4	3	2	1	0
PD1MFP				Reserved			

位	描述	
[31:28]	Reserved	保留.
[27:24]	PD6MFP	PD.6 管脚复用功能选择
[23:20]	PD5MFP	PD.5 管脚复用功能选择
[19:16]	PD4MFP	PD.4 管脚复用功能选择
[15:12]	PD3MFP	PD.3 管脚复用功能选择
[11:8]	PD2MFP	PD.2 管脚复用功能选择
[7:4]	PD1MFP	PD.1 管脚复用功能选择
[3:0]	Reserved	保留.

HIRC 校准控制寄存器 (SYS_IRCTCTL)

寄存器	偏移地址	R/W	描述	复位值
SYS_IRCTCTL	SYS_BA+0x80	R/W	HIRC 校准控制寄存器	0x0000_0030

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
RETRYCNT		LOOPSEL		Reserved			FREQSEL

位	描述	
[31:8]	Reserved	保留。
[7:6]	RETRYCNT	限制校准重复的次数 这个区域设定在HIRC（内部高速RC振荡器）锁频之前允许校准电路自动校准的最大次数。 HIRC锁频后，内部的校准更新计数寄存器将被清零。 如果校准更新计数器的值达到限制的次数后HIRC还没有锁频，那么自动校准功能将会被关闭，并且FREQSEL寄存器也会被清零。 00 = 重复校准次数限制为64次 01 = 重复校准次数限制为128次 10 = 重复校准次数限制为256次 11 = 重复校准次数限制为512次
[5:4]	LOOPSEL	校准计算周期选择 这个区域设定校准的计算基于多少个32.768 kHz时钟周期。 00 = 校准结果取4个32.768KHz时钟周期的平均值。 01 = 校准结果取8个32.768KHz时钟周期的平均值。 10 = 校准结果取16个32.768KHz时钟周期的平均值。 11 = 校准结果取32个32.768KHz时钟周期的平均值。 注： 例如，如果LOPPSEL设置成00，自动校准电路将会取4个32.768KHz周期的平均值作为校准值。
[3:1]	Reserved	保留。
[0]	FREQSEL	校准频率选择 这个区域设定是否打开内部48MHz高速RC振荡器的自动校准功能。 0 = 关闭HIRC自动校准功能。 1 = 打开内部48MHz HIRC自动校准功能。

HIRC 校准中断使能寄存器 (SYS_IRCTIEN)

寄存器	偏移地址	R/W	描述	复位值
SYS_IRCTIEN	SYS_BA+0x84	R/W	HIRC 校准中断使能寄存器	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							
7	6	5	4	3	2	1	0
Reserved					CLKEIEN	TFALIEN	Reserved

位	描述	
[31:3]	Reserved	保留.
[2]	CLKEIEN	时钟错误中断使能位 这位设定在自动校准后时钟不准时是否产生中断信号。 如果在自动校准时这个位被置1，并且CLKERRIF(SYS_IRCTSTS[2])也被置1，那么将会通过一个中断的方式来告诉用户时钟频率已经不准。 0 = 关闭寄存器CLKERRIF(SYS_IRCTSTS[2]) 状态触发中断功能。 1 = 打开寄存器CLKERRIF(SYS_IRCTSTS[2]) 状态触发中断功能。
[1]	TFALIEN	校准失败中断使能位 这位设定在 HIRC 校准更新计数器达到限制次数后 HIRC 频率未达到 FREQSEL (SYS_IRCTCTL[0])寄存器所设定的目标频率时是否触发中断。 在自动校准时如果这个位为高并且TFAILIF(SYS_IRCTSTS[1])也为高，那么将会产生一个中断来通知用户HIRC校准更新次数已到达限定值。 0 = 关闭TFAILIF(SYS_IRCTSTS[1])状态触发中断功能。 1 = 打开TFAILIF(SYS_IRCTSTS[1])状态触发中断功能。
[0]	Reserved	保留.

HIRC 校准中断状态寄存器 (SYS_IRCTISTS)

寄存器	偏移地址	R/W	描述	复位值
SYS_IRCTISTS	SYS_BA+0x88	R/W	HIRC 校准中断状态寄存器	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					CLKERRIF	TFAILIF	FREQLOCK

位	描述	
[31:3]	Reserved	保留。
[2]	CLKERRIF	时钟错误中断状态 当外部的32.768KHz低速晶振或内部48Mhz高速HIRC时钟频率的偏差超过合理范围时这一位将会被置1用来指示时钟频率不准确。 如果这位被置1，并且CLKEIEN(SYS_IRCTIEN[2])为高，将会产生一个中断来通知用户频率不准确。对此位写1可将其清零。 0 = 时钟准确 1 = 时钟不准确
[1]	TFAILIF	校准失败中断状态 这一位用于指示HIRC校准计数值达到限制次数后频率仍然没有被锁定。一旦这位被置1，自动校准将会被停止并且FREQSEL(SYS_IRCTCTL[1:0])将会被硬件自动清零。 如果这一位被置1，并且TFAILIEN(SYS_IRCTIEN[1])为高，将会产生一个中断来通知用户HIRC校准的计算值已经达到限制的次数。对此位写1可将其清零。 0 = 校准更新计数值未到达限定次数。 1 = 校准更新计数值达到限制次数并且HIRC频率未锁定。
[0]	FREQLOCK	HIRC频率锁定状态 这位指示HIRC频率已经被锁定。 这只是一个状态位，不产生任何中断。 0 = 内部的高速振荡器还没有锁定到48MHz 1 = 内部的高速振荡器已经锁定到48Mhz。

写保护寄存器控制寄存器 (SYS_REGLCTL)

有些寄存器需要被保护起来，以防止误写而影响芯片运行。这些系统寄存器在上电复位后就受保护，直到用户关闭寄存器保护。用户需要编程这些受保护寄存器的话，需要一个解锁时序。解锁时序是依次连续的往地址为0x5000_0100的控制器写入0x59、0x16、0x88。在写以上三个数据的过程中，写入任何其他数据、不同顺序写、或往别的地址写入数据都将终止解锁时序。

写保护功能失能后，用户可以查看保护失能位（地址0x5000_0100位0），1表示保护失能，0表示保护使能。写保护失能后，用户可以更新目标保护寄存器的值，往地址0x5000_0100写入任何值将使能寄存器保护。

可通过写这个寄存器来关闭或打开写保护功能，可通过读这个寄存器来获取REGLCTL的状态。

寄存器	偏移地址	R/W	描述	复位值
SYS_REGLCTL	SYS_BA+0x100	R/W	写保护寄存器控制寄存器	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
REGLCTL							

位	描述	
[31:8]	Reserved	Reserved.
[7:1]	REGPROTDIS	写保护代码寄存器(只写) 一些寄存器具有写保护功能。写这些寄存器前必须按照顺序依次往寄存器写入0x59、0x16、0x88来失能写保护功能。在这个时序结束后，SYS_REGLCTL位会被置1，写保护寄存器可以正常写入数据。
[0]	REGLCTL	写保护寄存器失能标志 (只读) 0 = 写保护寄存器的写保护功能使能，任何写入写保护寄存器的数据将被忽略 1 = 写保护寄存器的写保护功能失能 详情请参考6.2.6章节 注: 写保护位在其描述旁边会标注“(写保护)”。

温度传感器偏移寄存器 (SYS_TSOFFSET)

寄存器	偏移地址	R/W	描述	复位值
SYS_TSOFFSET	SYS_BA+0x114	R	温度传感器偏移寄存器	0x0XXX_0XXX

31	30	29	28	27	26	25	24
Reserved			VTEMP1				
23	22	21	20	19	18	17	16
VTEMP1							
15	14	13	12	11	10	9	8
Reserved			VTEMP0				
7	6	5	4	3	2	1	0
VTEMP0							

位	描述	
[31:28]	Reserved	保留。
[27:16]	VTEMP1	温度传感器偏移值 这个区域的数值反映温度传感器输出的电压在125°C 偏移。
[15:12]	Reserved	保留。
[11:0]	VTEMP0	温度传感器偏移值 这个区域的数值反映温度传感器输出的电压在25°C 偏移。

6.2.10 系统定时器 (SysTick)

Cortex®-M0内置一个系统定时器SysTick，SysTick提供一个简单的24位写清零、递减、自动装载，同时拥有灵活控制机制的寄存器。这个定时器有几种不同的用途，例如：

用于RTOS（实时操作系统）作为频率可编程（例如100Hz）滴答定时器并且有一个SysTick程序。

用于内核时钟的高速闹钟定时器

时间可调的闹钟或者单一定时器- 持续时间范围取决于使用的参考时钟和计数器的动态范围。

用做一个简单的计数器，软件用它来测量任务完成的时间。

用做基于时间间隔控制的内部时钟源，控制寄存器的COUNTFLAG位和状态寄存器可以用来判断一个任务是否在规定的时间内完成，把它作为动态时钟管理控制循环的一部分。

当系统定时器使能后，将从SysTick当前值寄存器(SYST_CVR)的值向下计数到0，并在下一个时钟边缘，重新加载SysTick重新加载值寄存器(SYST_RVR)的值，然后再随时钟递减。当计数器递减到0，COUNTFLAG 状态位就会被设置，读COUNTFLAG 位使其清零。

SYST_CVR的值在复位后是未知的，使能前，软件应该向寄存器写入值来清零。这样确保定时器使能时以 SYST_RVR的值开始计数，而非任意值。

如果SYST_RVR的值为0，定时器在重新加载后将保持值为0。这种机制可以在计数器使能后，独立地禁用计数功能。

详情请参考“ARM® Cortex®-M0技术参考手册”和“ARM® v6-M架构参考手册”。

6.2.10.1 系统定时器控制寄存器映射

R: 只读, W: 只写, R/W: 读/写, W&C: 写1清零

寄存器	偏移地址	R/W	描述	复位值
SCS 基地址: SCS_BA = 0xE000_E000				
SYST_CTL	SCS_BA+0x10	R/W	SysTick 控制和状态寄存器	0x0000_0004
SYST_RVR	SCS_BA+0x14	R/W	SysTick 重新加载值寄存器	0xFFFF_FFFF
SYST_CVR	SCS_BA+0x18	R/W	SysTick 当前值寄存器	0xFFFF_FFFF

6.2.10.2 系统定时器控制寄存器描述

SysTick 控制和状态寄存器 (SYST_CTL)

寄存器	偏移地址	R/W	描述	复位值
SYST_CTL	SCS_BA+0x10	R/W	SysTick 控制和状态寄存器	0x0000_0004

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	Reserved.
[16]	COUNTFLAG	系统滴答计数器标志位 从上次读取该寄存器后，如果定时器计数到0，则返回1 0 = 读取的时候对COUNTFLAG清零或者写当前值寄存器对COUNTFLAG清零 1 = 计数从1到0转变时，COUNTFLAG置1
[15:3]	Reserved	保留.
[2]	CLKSRC	系统滴答时钟源选择位 0 = 时钟源是可选的，参考STCLKSEL 1 = 内核时钟用于SysTick 定时器
[1]	TICKINT	系统滴答中断使能位 0 = 向下计数到0不会引起SysTick异常挂起。用户可以根据COUNTFLAG标志位来确定是否已经发生计数到0的情况。 1 = 向下计数到0将引起SysTick异常挂起。软件清除SysTick当前值寄存器值将不会导致SysTick挂起。
[0]	ENABLE	系统滴答计数器使能位 0 = 计数器失能 1 = 计数器使能并运行于multi-shot方式

SysTick 重加载值寄存器 (SYST_RVR)

寄存器	偏移地址	R/W	描述	复位值
SYST_RVR	SCS_BA+0x14	R/W	SysTick 重加载值寄存器	0xFFFF_FFFF

31	30	29	28	27	26	25	24
Reserved							
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	系统滴答重加载值 当计数器计数到0时, 该值将加载到当前值寄存器

SysTick 当前值寄存器（SYST_CVR）

寄存器	偏移地址	R/W	描述	复位值
SYST_CVR	SCS_BA+0x18	R/W	SysTick 当前值寄存器	0xFFFF_FFFF

31	30	29	28	27	26	25	24
Reserved							
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	系统滴答当前值 当前计数值。该值为采样时刻的计数器的值。计数器不提供读修改写保护功能。该寄存器有写清除功能，软件写入任何值到该寄存器将清除该寄存器值为0。

6.2.11 嵌套向量中断控制器 (NVIC)

6.2.11.1 概述

Cortex®-M0 CPU提供一个中断控制器用于异常模式，称之为“嵌套向量中断控制器(NVIC)”。NVIC与处理器内核紧密相连，它提供以下特征：

6.2.11.2 特性

- 支持嵌套向量中断
- 自动保存和恢复处理器状态
- 动态改变优先级
- 简化和确定的中断时间

NVIC按照优先级处理所有支持的异常。所有的异常在“Handler模式”处理。NVIC架构支持32个 (IRQ[31:0])离散中断，每个中断支持4级中断优先级。所有的中断和大部分系统异常可以配置成不同的优先级等级。当一个中断发生时，NVIC将比较新中断与当前中断的优先级，如果新中断优先级高，则立即处理新中断。

当一个中断接受后，中断服务程序(ISR)的开始地址可从内存中的向量表取得。软件不需要决定哪个中断被响应，也不用分配相关ISR的开始地址。当开始地址取得后，NVIC将自动保存包含寄存器“PC、PSR、LR、R0~R3、R12”值的处理器状态到栈中。在ISR结束后，NVIC将从栈中恢复相关寄存器的值，并运行在正常状态。因此花费少量且确定的时间处理中断请求。

NVIC支持“末尾连锁”，可以有效的处理背对背中断，即无需保存和恢复当前状态，从而减少结束当前ISR切换到挂起ISR的延迟时间。NVIC还支持“Late Arrival”，因此可以提高并发中断的效率。当较高优先级中断请求发生在当前ISR开始执行之前（保存处理器状态和获取起始地址阶段），NVIC将立即处理更高优先级的中断，从而提高实时性。

更多详细信息，请参考“ARM® Cortex®-M0 技术参考手册”和“ARM® v6-M 架构参考手册”。

在进入异常中断时处理器将会自动把状态寄存器的数据保存到堆栈里，退出异常时自动释放堆栈并恢复状态，不需要消耗任何指令。这提供低延时的异常处理机制。

6.2.11.3 异常模式和系统中断映射

表6.2-8列出了NuMicro® NM1120系列支持的异常模式。与所有中断一样，软件可以为其中一些异常设置4级优先级。最高用户可配置优先级为0，最低优先级为3。所有用户可配置中断的默认优先级为0。注：优先级0在系统为第4优先级，排在“Reset”、“NMI”和“Hard Fault”三个系统异常之后。

异常名称	向量号	优先级
------	-----	-----

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

表 6.2-8 异常模式

异常号	中断号 (中断寄存器对应位)	中断名称	中断描述
0 ~ 15	-	-	系统异常
16	0	BOD_OUT	欠压检测中断
17	1	WDTPINT	看门狗定时器中断
18	2	USCI0	通用串行控制接口0中断
19	3	USCI1	通用串行控制接口1中断
20	4	GP_INT	GPA ~ GPD 管脚外部中断
21	5	EPWM_INT	EPWM 中断
22	6	BRAKE0_INT	来自 PWM0 或 PWM_BRAKE 管脚的EPWM打断暂停中断
23	7	BRAKE1_INT	来自 PWM1的EPWM打断暂停中断
24	8	BPWM0_INT	BPWM0 中断
25	9	BPWM1_INT	BPWM1 中断
26	10	Reserved	保留
27	11	Reserved	保留
28	12	Reserved	保留
29	13	Reserved	保留
30	14	Reserved	保留
31	15	ECAP_INT	增强型输入捕捉中断
32	16	CCAP_INT	连续输入捕捉中断

33	17	Reserved	保留
34	18	Reserved	保留
35	19	Reserved	保留
36	20	Reserved	保留
37	21	HIRCTRIM_INT	HIRC校准中断
38	22	TMR0_INT	定时器0中断
39	23	TMR1_INT	定时器1中断
40	24	Reserved	保留
41	25	Reserved	保留
42	26	ACMP_INT	模拟比较器 0 或 计较器 1 中断
43	27	Reserved	保留
44	28	PWRWU_INT	芯片从掉电模式唤醒中断
45	29	ADC0_INT	ADC0 中断
46	30	ADC1_INT	ADC1 中断
47	31	ADCWCMP_INT	ADC 窗口比较中断

表 6.2-9 系统中断映射向量表

6.2.11.4 向量表

当一个中断被接受时，处理器会自动从内存中的向量表取出中断服务例程(ISR)的起始地址。对于 ARMv6-M，向量表基地址固定为0x00000000。向量表包含复位后的栈指针初始值和所有异常处理程序的入口地址。前一页的向量号表示前部分处理异常的先后顺序。

向量表字偏移地址 (字节)	描述
0x00	初始化堆栈指针值
异常号 * 0x04	异常入口指针所对应的异常号

表 6.2-10 向量表格式

6.2.11.5 向量描述

NVIC中断可以通过写使能中断或清使能中断寄存器相关位来使能或失能。这些寄存器使用写1使能和写1清零机制，通过读取这些寄存器可以读取当前相应中断的使能状态。当中断禁用后，中断声明将使中断挂起，因此中断不被激活。如果在禁用中断时中断被激活，它将保持激活状态直到被复位或异常返回清除。清使能位可以阻止相应中断的新中断被激活。

NVIC中断可以使用互补寄存器对来挂起/取消挂起来使能/禁用中断，这些寄存器分别为Set-

Pending寄存器 和Clear-Pending寄存器。这些寄存器使用写1使能和写1清零机制，通过读取这些寄存器可以读取当前相应中断的挂起状态。Clear-Pending寄存器不影响激活中断的执行状态。

NVIC中断的优先级可以通过更新32位寄存器的8个位段来设置（每个寄存器支持4个中断）。

与NVIC相关的通用寄存器在系统控制空间的一块内存中设置，下一节将做出描述。

6.2.11.6 NVIC 控制寄存器映射

R: 只读, W: 只写, R/W: 读/写

寄存器	偏移地址	R/W	描述	复位值
SCS 基地址: SCS_BA = 0xE000_E000				
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0 ~ IRQ31 设置使能控制寄存器	0x0000_0000
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0 ~ IRQ31 清除使能控制寄存器	0x0000_0000
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0 ~ IRQ31 设置挂起控制寄存器	0x0000_0000
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0 ~ IRQ31 清除挂起控制寄存器	0x0000_0000
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0 ~ IRQ3 中断优先级控制寄存器	0x0000_0000
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4 ~ IRQ7 中断优先级控制寄存器	0x0000_0000
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8 ~ IRQ11 中断优先级控制寄存器	0x0000_0000
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12 ~ IRQ15 中断优先级控制寄存器	0x0000_0000
NVIC_IPR4	SCS_BA+0x410	R/W	IRQ16 ~ IRQ19 中断优先级控制寄存器	0x0000_0000
NVIC_IPR5	SCS_BA+0x414	R/W	IRQ20 ~ IRQ23 中断优先级控制寄存器	0x0000_0000
NVIC_IPR6	SCS_BA+0x418	R/W	IRQ24 ~ IRQ27 中断优先级控制寄存器	0x0000_0000
NVIC_IPR7	SCS_BA+0x41C	R/W	IRQ28 ~ IRQ31 中断优先级控制寄存器	0x0000_0000

IRQ0 ~ IRQ31 设置使能控制寄存器 (NVIC_ISER)

寄存器	偏移地址	R/W	描述	复位值
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0 ~ IRQ31 设置使能控制寄存器	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	<p>中断使能位</p> <p>使能一个或多个中断。每一位代表一个从IRQ0~IRQ31的中断（向量号从16~47）。</p> <p>写操作:</p> <p>0 = 无效</p> <p>1 = 写1使能相关中断</p> <p>读操作:</p> <p>0 = 相关中断状态禁止</p> <p>1 = 相关中断状态使能</p> <p>注: 读该寄存器值表明当前使能状态</p>

IRQ0 ~ IRQ31 清除使能控制寄存器 (NVIC_ICER)

寄存器	偏移地址	R/W	描述	复位值
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0 ~ IRQ31 清除使能控制寄存器	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	<p>中断禁用位</p> <p>禁用一个或多个中断。每一位表示一个从IRQ0~IRQ31的中断（向量号从16~47）。</p> <p>写操作:</p> <p>0 = 无效</p> <p>1 = 写1禁止相关中断</p> <p>读操作:</p> <p>0 = 相关中断状态禁止</p> <p>1 = 相关中断状态使能</p> <p>注: 读该寄存器值表明当前使能状态</p>

IRQ0 ~ IRQ31 设置挂起控制寄存器 (NVIC_ISPR)

寄存器	偏移地址	R/W	描述	复位值
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0 ~ IRQ31 设置挂起控制寄存器	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]	<p>SETPEND</p> <p>设置中断挂起位</p> <p>写操作:</p> <p>0 = 无效</p> <p>1 = 写1置挂起状态。每一位表示一个从IRQ0 ~ IRQ31的中断（向量号从16~47）。</p> <p>读操作:</p> <p>0 = 相关中断不在挂起状态</p> <p>1 = 相关中断在挂起状态</p> <p>注: 读该寄存器值表明当前挂起状态</p>

IRQ0 ~ IRQ31 清除挂起控制寄存器 (NVIC_ICPR)

寄存器	偏移地址	R/W	描述	复位值
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0 ~ IRQ31 清除挂起控制寄存器	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	<p>清中断挂起位</p> <p>写操作:</p> <p>0 = 无效</p> <p>1 = 写1清挂起状态。每一位表示一个从IRQ0 ~ IRQ31的中断（向量号从16~47）。</p> <p>读操作:</p> <p>0 = 相关中断不在挂起状态</p> <p>1 = 相关中断在挂起状态</p> <p>注: 读该寄存器值表明当前挂起状态</p>

IRQ0 ~ IRQ3 中断优先级寄存器 (NVIC_IPR0)

寄存器	偏移地址	R/W	描述	复位值
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0 ~ IRQ3 中断优先级控制寄存器	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优先级 0表示最高优先级, 3表示最低优先级
[29:24]	Reserved	保留.
[23:22]	PRI_2	IRQ2优先级 0表示最高优先级, 3表示最低优先级
[21:16]	Reserved	保留.
[15:14]	PRI_1	IRQ1优先级 0表示最高优先级, 3表示最低优先级
[13:8]	Reserved	保留.
[7:6]	PRI_0	IRQ0优先级 0表示最高优先级, 3表示最低优先级
[5:0]	Reserved	保留.

IRQ4 ~ IRQ7 中断优先级寄存器 (NVIC_IPR1)

寄存器	偏移地址	R/W	描述	复位值
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4 ~ IRQ7 中断优先级控制寄存器	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优先级 0表示最高优先级, 3表示最低优先级
[29:24]	Reserved	保留.
[23:22]	PRI_6	IRQ6优先级 0表示最高优先级, 3表示最低优先级
[21:16]	Reserved	保留.
[15:14]	PRI_5	IRQ5优先级 0表示最高优先级, 3表示最低优先级
[13:8]	Reserved	保留.
[7:6]	PRI_4	IRQ4优先级 0表示最高优先级, 3表示最低优先级
[5:0]	Reserved	保留.

IRQ8 ~ IRQ11 中断优先级寄存器 (NVIC_IPR2)

寄存器	偏移地址	R/W	描述	复位值
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8 ~ IRQ11 中断优先级控制寄存器	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优先级 0表示最高优先级，3表示最低优先级
[29:24]	Reserved	保留。
[23:22]	PRI_10	IRQ10优先级 0表示最高优先级，3表示最低优先级
[21:16]	Reserved	保留。
[15:14]	PRI_9	IRQ9优先级 0表示最高优先级，3表示最低优先级
[13:8]	Reserved	保留。
[7:6]	PRI_8	IRQ8优先级 0表示最高优先级，3表示最低优先级
[5:0]	Reserved	保留。

IRQ12 ~ IRQ15 中断优先级寄存器 (NVIC_IPR3)

寄存器	偏移地址	R/W	描述	复位值
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12 ~ IRQ15 中断优先级控制寄存器	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优先级 0表示最高优先级，3表示最低优先级
[29:24]	Reserved	保留。
[23:22]	PRI_14	IRQ14优先级 0表示最高优先级，3表示最低优先级
[21:16]	Reserved	保留。
[15:14]	PRI_13	IRQ13优先级 0表示最高优先级，3表示最低优先级
[13:8]	Reserved	保留。
[7:6]	PRI_12	IRQ12优先级 0表示最高优先级，3表示最低优先级
[5:0]	Reserved	保留。

IRQ16 ~ IRQ19 中断优先级寄存器 (NVIC_IPR4)

寄存器	偏移地址	R/W	描述	复位值
NVIC_IPR4	SCS_BA+0x410	R/W	IRQ16 ~ IRQ19 中断优先级控制寄存器	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优先级 0表示最高优先级, 3表示最低优先级
[29:24]	Reserved	保留.
[23:22]	PRI_18	IRQ18优先级 0表示最高优先级, 3表示最低优先级
[21:16]	Reserved	保留.
[15:14]	PRI_17	IRQ17优先级 0表示最高优先级, 3表示最低优先级
[13:8]	Reserved	保留.
[7:6]	PRI_16	IRQ16优先级 0表示最高优先级, 3表示最低优先级
[5:0]	Reserved	保留.

IRQ20 ~ IRQ23 中断优先级寄存器 (NVIC_IPR5)

寄存器	偏移地址	R/W	描述	复位值
NVIC_IPR5	SCS_BA+0x414	R/W	IRQ20 ~ IRQ23 中断优先级控制寄存器	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优先级 0表示最高优先级, 3表示最低优先级
[29:24]	Reserved	保留.
[23:22]	PRI_22	IRQ22优先级 0表示最高优先级, 3表示最低优先级
[21:16]	Reserved	保留.
[15:14]	PRI_21	IRQ21优先级 0表示最高优先级, 3表示最低优先级
[13:8]	Reserved	保留.
[7:6]	PRI_20	IRQ20优先级 0表示最高优先级, 3表示最低优先级
[5:0]	Reserved	保留.

IRQ24 ~ IRQ27 中断优先级寄存器 (NVIC_IPR6)

寄存器	偏移地址	R/W	描述	复位值
NVIC_IPR6	SCS_BA+0x418	R/W	IRQ24 ~ IRQ27 中断优先级控制寄存器	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优先级 0表示最高优先级，3表示最低优先级
[29:24]	Reserved	保留。
[23:22]	PRI_26	IRQ26优先级 0表示最高优先级，3表示最低优先级
[21:16]	Reserved	保留。
[15:14]	PRI_25	IRQ25优先级 0表示最高优先级，3表示最低优先级
[13:8]	Reserved	保留。
[7:6]	PRI_24	IRQ24优先级 0表示最高优先级，3表示最低优先级
[5:0]	Reserved	保留。

IRQ28 ~ IRQ31 中断优先级寄存器 (NVIC_IPR7)

寄存器	偏移地址	R/W	描述	复位值
NVIC_IPR7	SCS_BA+0x41C	R/W	IRQ28 ~ IRQ31 中断优先级控制寄存器	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优先级 0表示最高优先级，3表示最低优先级
[29:24]	Reserved	保留。
[23:22]	PRI_30	IRQ30优先级 0表示最高优先级，3表示最低优先级
[21:16]	Reserved	保留。
[15:14]	PRI_29	IRQ29优先级 0表示最高优先级，3表示最低优先级
[13:8]	Reserved	保留。
[7:6]	PRI_28	IRQ28优先级 0表示最高优先级，3表示最低优先级
[5:0]	Reserved	保留。

6.2.11.7 中断源控制寄存器

除了与NVIC相关的中断控制寄存器外，NuMicro® NM1120 系列也提供了一些特殊的控制寄存器来增强中断功能，包含“NMI源选择”和“IRQ（中断请求）号身份识别”，描述如下。

R: 只读, W: 只写, R/W: 读/写

寄存器	偏移地址	R/W	描述	复位值
INT 基地址: INT_BA = 0x5000_0300				
INT_NMICTL	INT_BA+0x80	R/W	NMI中断源选择控制寄存器	0x0000_0000
INT_IRQSTS	INT_BA+0x84	R/W	MCU中断请求号身份识别寄存器	0x0000_0000

NMI 中断源选择控制寄存器 (INT_NMICTL)

寄存器	偏移地址	R/W	描述	复位值
INT_NMICTL	INT_BA+0x80	R/W	NMI 中断源选择控制寄存器	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							NMISELEN
7	6	5	4	3	2	1	0
Reserved				NMISEL			

位	描述	
[31:9]	Reserved	保留.
[8]	NMISELEN	NMI中断使能位（写保护） 0 = 关闭NMI中断 1 = 打开NMI中断 注： 这是一个保护位，对它编程之前需对地址0x5000_0100写0x59, 0x16, 和 0x88先把它解锁，请参考地址为SYS_BA+0x100的SYS_REGLCTL寄存器
[7:5]	Reserved	保留.
[4:0]	NMISEL	NMI中断源选择 Cortex®-M0的NMI中断可以通过设定NMTSEL寄存器从外部中断当中选择。

MCU中断请求源寄存器 (INT_IRQSTS)

寄存器	偏移地址	R/W	描述	复位值
INT_IRQSTS	INT_BA+0x84	R/W	MCU IRQ 号识别寄存器	0x0000_0000

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

Bits	Description	
[31:0]	IRQ	<p>MCU IRQ源寄存器</p> <p>IRQ收集来自外设的所有中断，然后形成到Cortex®-M0内核的同步中断。有一种使Cortex®-M0产生中断的模式叫做正常模式。</p> <p>IRQ收集来自每个外设的所有中断、并使他们同步，然后中断Cortex®-M0。</p> <p>当IRQ[n]为0时，设置IRQ[n]为1将生成一个到Cortex®-M0 NVIC[n]的中断。</p> <p>当IRQ[n]为1（例如一个中断被声明）时，设置MCU_bit[n]为1将清除中断，设置IRQ[n] 为0无影响。</p>

6.2.12 系统控制寄存器

Cortex®-M0关键的控制和状态等特性由系统控制模块在系统控制寄存器里集中的管理。

更多的详细信息请参考“ARM® Cortex®-M0技术参考手册”和“ARM® v6-M架构参考手册”

6.2.12.1 系统控制寄存器内存映射

R: 只读, W: 只写, R/W: 读/写

寄存器	偏移地址	R/W	描述	复位值
SCS 基地址: SCS_BA = 0xE000_E000				
SCS_CPUID	SCS_BA+0xD00	R	CPUID 寄存器	0x410C_C200
SCS_ICSR	SCS_BA+0xD04	R/W	中断控制状态寄存器	0x0000_0000
SCS_AIRCR	SCS_BA+0xD0C	R/W	应用中断和复位控制寄存器	0xFA05_0000
SCS_SCR	SCS_BA+0xD10	R/W	系统控制寄存器	0x0000_0000
SCS_SHPR2	SCS_BA+0xD1C	R/W	系统处理器优先级寄存器2	0x0000_0000
SCS_SHPR3	SCS_BA+0xD20	R/W	系统处理器优先级寄存器3	0x0000_0000

6.2.12.2 系统控制寄存器描述

CPUID 寄存器 (CPUID)

寄存器	偏移地址	R/W	描述	复位值
SCS_CPUID	SCS_BA+0xD00	R	CPUID 寄存器	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	实施代码 实施代码由 ARM® 分配(ARM® = 0x41)
[23:20]	Reserved	保留.
[19:16]	PART	处理器的架构 ARMv6-M读出来的值为0xC
[15:4]	PARTNO	处理器产品编号 读出来的值为0xC20
[3:0]	REVISION	修订号 读出来的值为0x0

中断控制状态寄存器 (ICSR)

寄存器	偏移地址	R/W	描述	复位值
SCS_ICSR	SCS_BA+0xD04	R/W	中断控制状态寄存器	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
ISRPREEMPT	ISRPENDING	Reserved	VECTPENDING				
15	14	13	12	11	10	9	8
VECTPENDING				Reserved			VECTACTIVE
7	6	5	4	3	2	1	0
VECTACTIVE							

位	描述	
[31]	NMIPENDSET	<p>NMI 设置挂起位</p> <p>写操作:</p> <p>0 = 无效</p> <p>1 = 将NMI异常挂起</p> <p>读操作:</p> <p>0 = NMI异常没有挂起</p> <p>1 = NMI异常挂起</p> <p>注: 由于NMI是最高优先级异常, 所以通常处理器一检测到该位写1就会进入NMI异常处理。进入异常处理后, 处理器会将该位清零。这意味着只有当处理器正在执行NMI异常处理程序时再次产生NMI信号, NMI异常处理程序读取这一位才返回1。</p>
[30:29]	Reserved	保留。
[28]	PENDSVSET	<p>PendSV设置挂起位</p> <p>写操作:</p> <p>0 = 无效</p> <p>1 = 将PendSV异常挂起</p> <p>读操作:</p> <p>0 = PendSV异常没有挂起</p> <p>1 = PendSV异常挂起</p> <p>注: 设置该位为1是设置PendSV 异常挂起的唯一方法。</p>
[27]	PENDSVCLR	<p>PendSV 清除挂起位</p> <p>写操作:</p> <p>0 = 无效</p> <p>1 = 清除PendSV异常挂起状态</p> <p>该位为只写位.为了清除PENDSV位,你必须同时往PENDSVSET写0,往PENDSVCLR写1。</p>

[26]	PENDSTSET	SysTick异常设置挂起位 写操作: 0 = 无效 1 = 将SysTick异常挂起 读操作: 0 = SysTick异常没有挂起 1 = SysTick异常挂起
[25]	PENDSTCLR	SysTick异常清除挂起位 写操作: 0 = 无效 1 = 清除SysTick异常挂起状态 注: 该位为只读位。当你要清除PENDST位时, 必须同时往PENDSTSET写0, 往PENDSTCLR写1。
[24]	Reserved	保留.
[23]	ISRPREEMPT	中断抢先占有位 (只读) 如果该位设置, 一个挂起的异常将会从调试停止状态退出并进入服务。
[22]	ISRPENDING	中断挂起标志, 不包括 NMI和Faults (只读) 0 = 中断没有挂起 1 = 中断挂起
[21]	Reserved	保留.
[20:12]	VECTPENDING	允许挂起的最高优先级异常的异常编号 0 = 没有挂起的异常 非0 = 允许挂起的最高优先级异常的异常编号。
[11:9]	Reserved	保留.
[8:0]	VECTACTIVE	包含当前执行异常号 0 = Thread模式 非0 = 当前执行异常的异常号

应用中断和复位控制寄存器 (AIRC_R)

寄存器	偏移地址	R/W	描述	复位值
SCS_AIRC_R	SCS_BA+0xD0C	R/W	应用中断和复位控制寄存器	0xFA05_0000

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

Bits	Description	
[31:16]	VECTORKEY	<p>寄存器访问密钥</p> <p>写操作: 当要写这个寄存器时, VECTORKEY 域必须设置为0x05FA, 否则写操作将被忽略。 VECTORKEY 域用于防止系统复位或者清除异常状态时误写该寄存器。</p> <p>读操作: 读出来的值为0xFA05.</p>
[15:3]	Reserved	保留.
[2]	SYSRESETREQ	<p>系统复位请求</p> <p>往该位写1将引起一个复位信号给芯片, 表明有复位请求。 该位为只写位, 复位后自动清零。</p>
[1]	VECTCLRACTIVE	<p>异常有效状态清除位</p> <p>保留给调试使用。当写这个寄存器时, 用户必须往该位写0, 否则将出现不可预知的情况。</p>
[0]	Reserved	保留.

系统控制寄存器 (SCR)

寄存器	偏移地址	R/W	描述	复位值
SCS_SCR	SCS_BA+0xD10	R/W	System Control Register	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

Bits	Description
[31:5]	Reserved 保留.
[4]	SEVONPEND 挂起时发送事件 0 = 只有使能中断或事件可以唤醒处理器，不包括禁用中断在内。 1 = 使能事件和所有中断（包括禁用的中断），可以唤醒处理器。 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。如果处理器没在等待事件，事件将会被注册并影响下一个WFE。 执行SEV指令或外部事件同样会唤醒处理器。
[3]	Reserved 保留.
[2]	SLEEPDEEP 处理器深度睡眠和睡眠模式选择 控制处理器在低电模式时使用休眠还是深度休眠模式。 0 = 休眠模式 1 = 深度休眠模式
[1]	SLEEPONEXIT Sleep-On-Exit使能位 该位表明当从Handler 模式返回到Thread模式时，是否退出休眠 0 = 当从Thread 模式返回时，不休眠 1 = 从ISR返回到Thread模式时，进入休眠或深度休眠 置该位为1使能一个中断驱动应用，从而避免返回到一个空的主函数应用。
[0]	Reserved 保留.

系统处理器优先级寄存器 2 (SHPR2)

寄存器	偏移地址	R/W	描述	复位值
SCS_SHPR2	SCS_BA+0xD1C	R/W	系统处理器优先级寄存器 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	系统处理器11优先级 – SVCall 0表示最高优先级，3表示最低优先级
[29:0]	Reserved	保留.

系统处理器优先级寄存器 3 (SHPR3)

寄存器	偏移地址	R/W	描述	复位值
SCS_SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 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	系统处理器15优先级– SysTick 0表示最高优先级，3表示最低优先级
[29:24]	Reserved	保留.
[23:22]	PRI_14	系统处理器14优先级– PendSV 0表示最高优先级，3表示最低优先级
[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。在CPU使能低功耗PDEN(CLK_PWRCTL[7])位和Cortex®-M0内核执行WFI指令后，芯片才能进入低功耗模式。直到唤醒中断发生，芯片才会退出低功耗模式。在低功耗模式下，时钟控制器会关闭外部4~24MHz高速晶振和内部48MHz高速RC振荡器，以降低整个系统功耗。图6.3-2所示各模块时钟发生器和时钟源的简图。

时钟发生器由如下4个时钟源组成：

- 32.768 kHz 外部低速晶振(LXT)
- 4~24 MHz 外部高速晶振(HXT)
- 48 MHz 内部高速 RC 振荡器(HIRC)
- 10 kHz 内部低速 RC 振荡器 (LIRC)

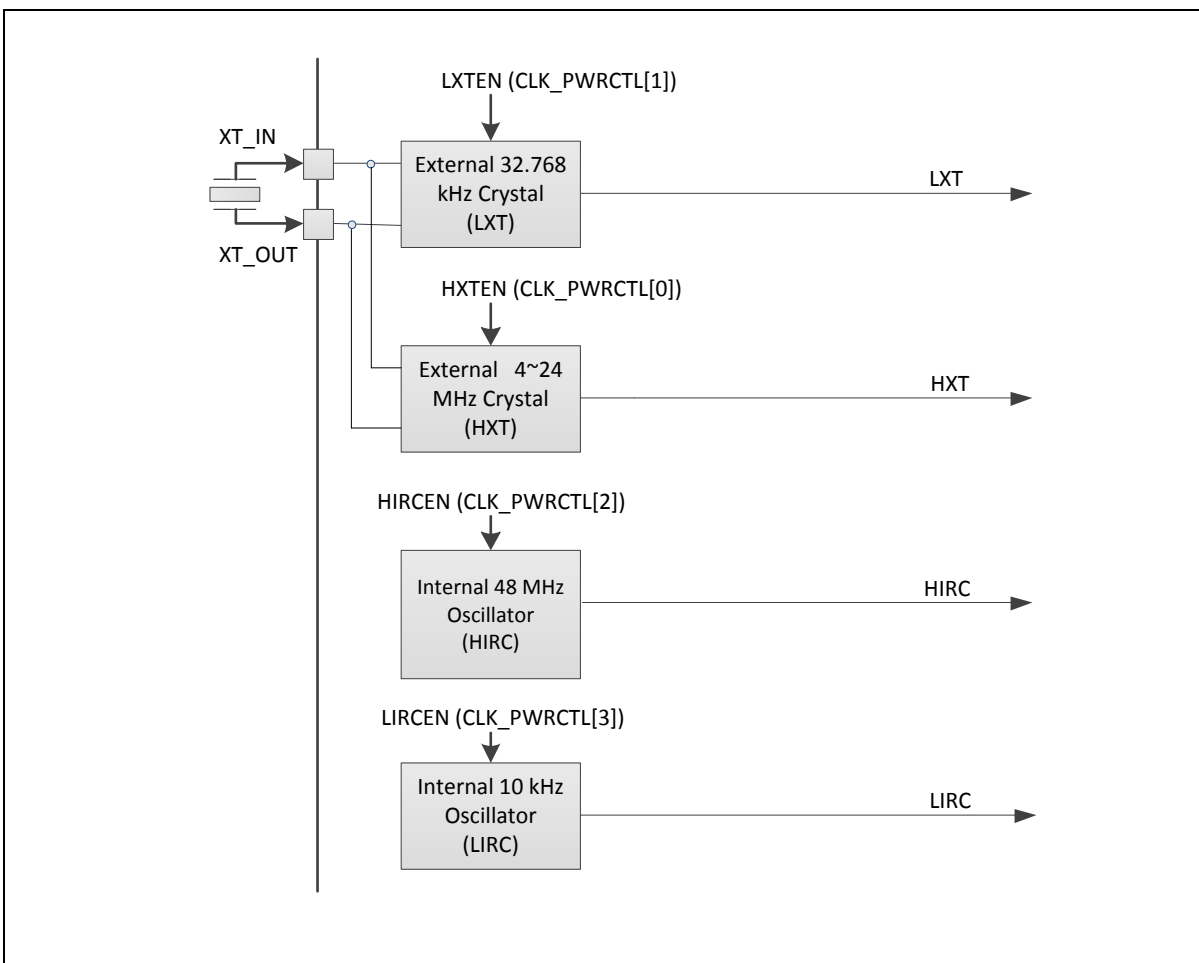


图 6.3-1 时钟发生器框图

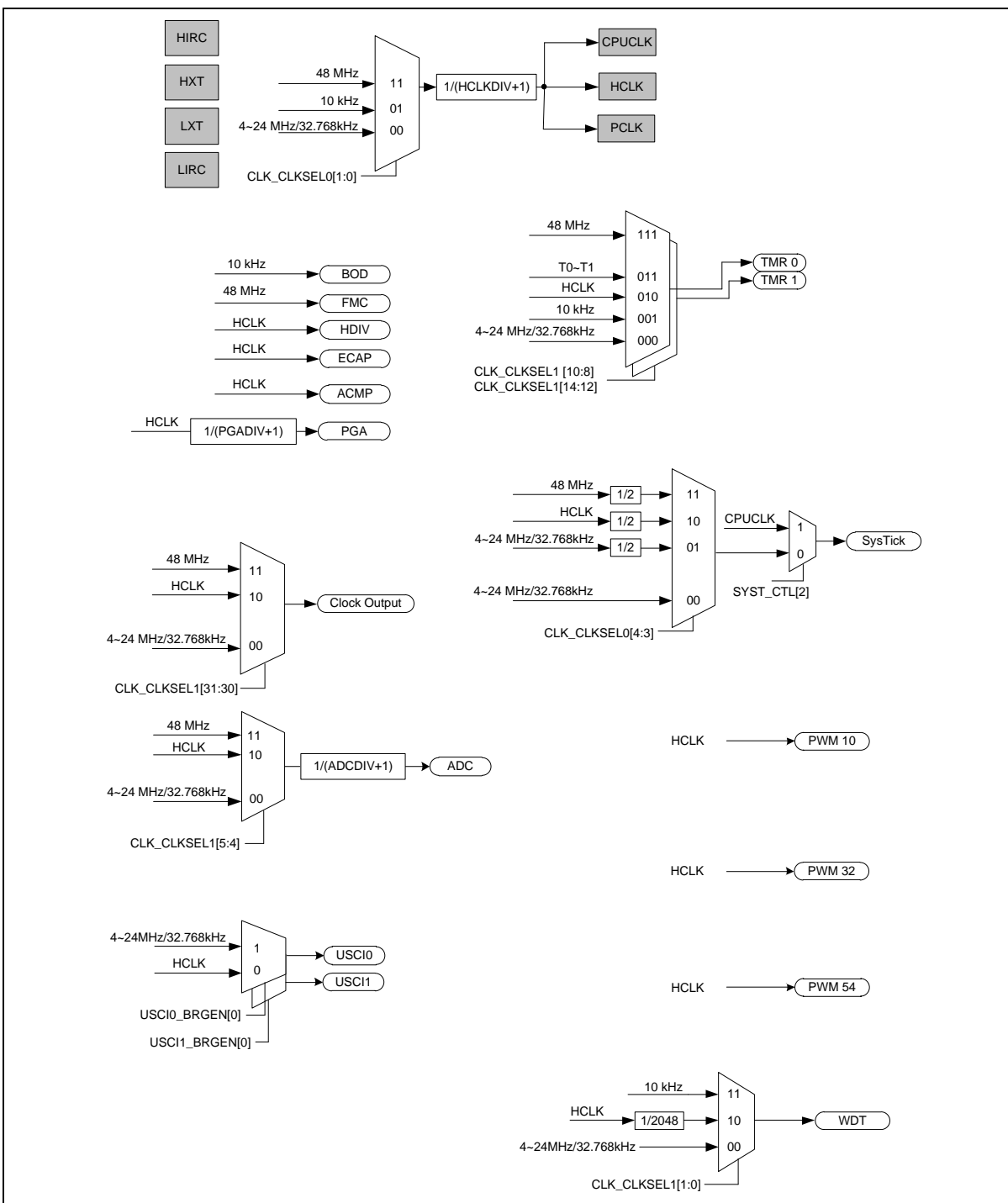


图 6.3-2 时钟发生器全局框图

6.3.2 自动校准

该芯片支持自动频率校准功能：HIRC（内部 48MHz RC 振荡器）可以通过精确的LXT（外部32.768 kHz晶体振荡器）自动调整输出精确的HIRC频率，在全温度范围里，误差在0.25%内。

例如：系统需要一个精确的48MHz时钟，但用户不想使用PLL作为系统时钟源，他们可以在系统上焊接一个32.768 kHz晶振，然后设置FREQSEL (SYS_IRCTCTL[0] 频率校准选择) 为“1”，这样自动频率校准功能就会被使能。中断状态标志位FREQLOCK (SYS_IRCTISTS[0] HIRC频率锁定状态)为“1”表明HIRC输出频率误差在0.25%内。为了取得更好的效果，推荐同时设置LOOPSEL (SYS_IRCTCTL[5:4] 频率校准计算循环) 和RETRYCNT (SYS_IRCTCTL[7:6]校准值更新限制计数) 为“11”。

6.3.3 系统时钟和 SysTick 时钟

系统时钟有 3 个可选时钟源，由时钟发生器产生。时钟源切换取决于寄存器 HCLKSEL(CLK_CLKSEL0[1:0])。其框图如图 6.3-3所示。

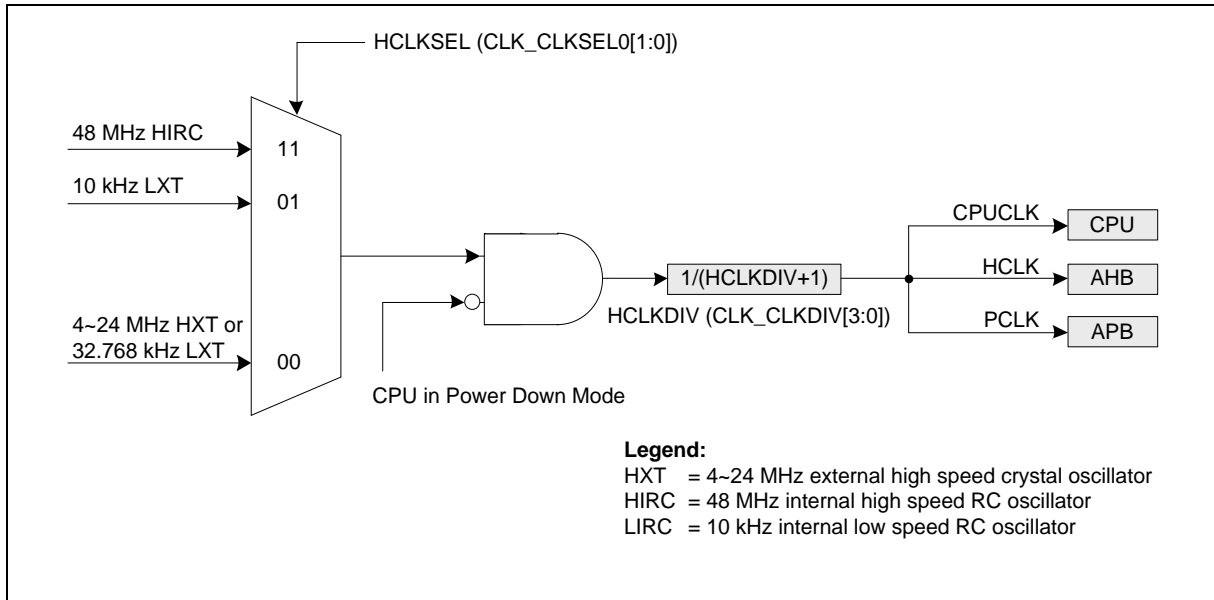


图 6.3-3 系统时钟框图

Cortex®-M0内核的SysTick 时钟源可以选择CPU时钟或外部时钟CLKSRC(SYST_CTL[2])。如果使用外部时钟，SysTick 时钟 (STCLK) 有 4 个可选时钟源。时钟源切换取决于寄存器STCLKSEL (CLK_CLKSEL0[4:3])。其框图如图 6.3-4所示：

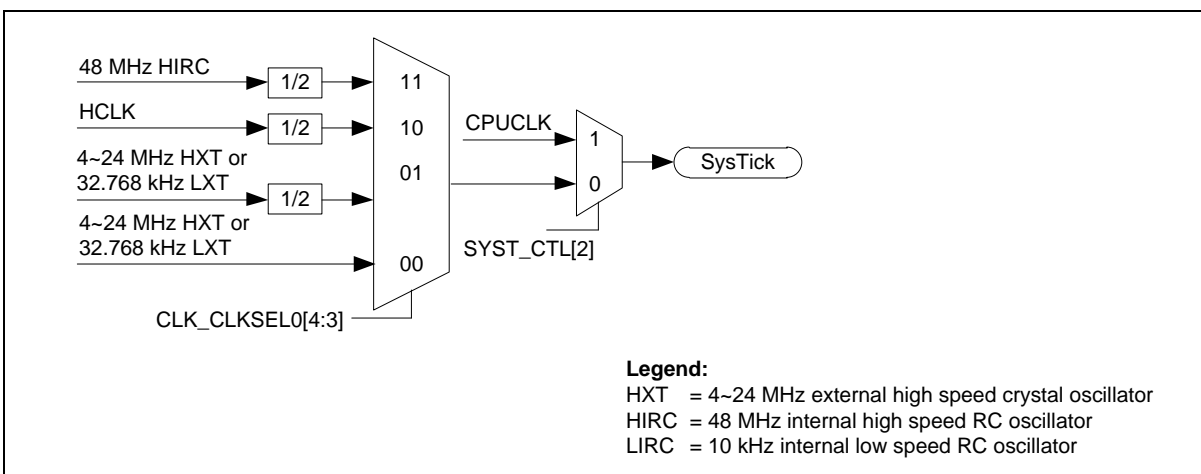


图 6.3-4 SysTick 时钟控制框图

6.3.4 外设时钟源选择

外设时钟可以有不同的时钟源做切换设置。主要取决于不同的外设。请参看寄存器CLK_CLKSEL1和CLK_APBCLK。在**Error! Reference source not found.**章节有详细描述。

请注意, 用户必须等到两个时钟源都处于稳定状态之后, 才可从一个时钟源切换到另一个时钟源。

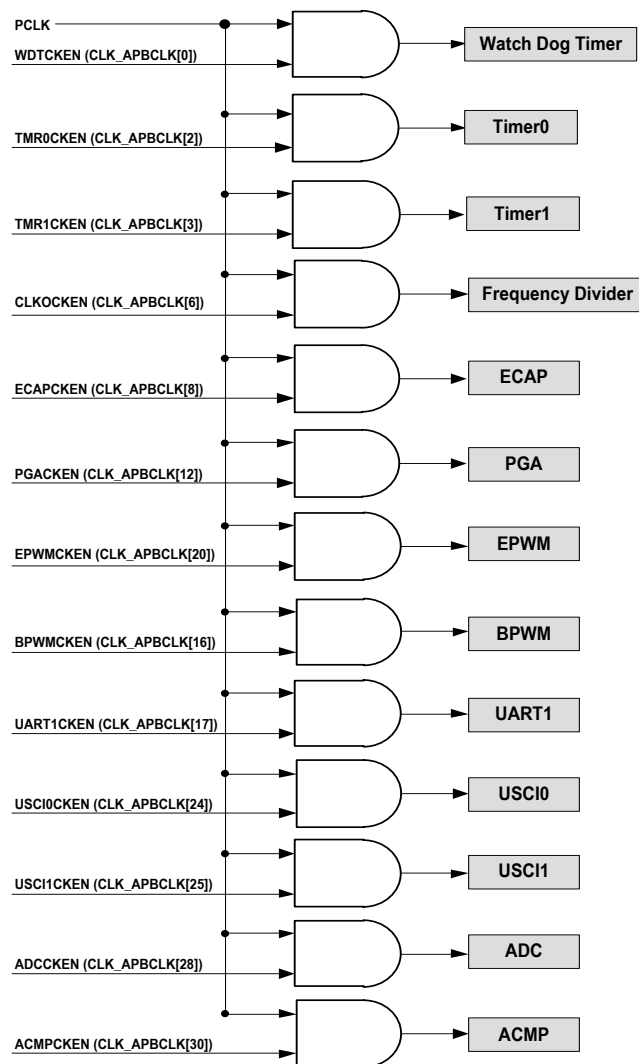


图 6.3-5 PCLK 外设总线时钟源选择

	外设时钟源选择表	Ext. CLK (HXT Or LXT)	HIRC	LIRC	HCLK
WDT	Yes	Yes	No	Yes	Yes
WWDT	Yes	Yes	No	Yes	Yes
Timer0	Yes	Yes	Yes	Yes	Yes
Timer1	Yes	Yes	Yes	Yes	Yes
USCI0	Yes	Yes	Yes	Yes	Yes
USCI1	Yes	Yes	Yes	Yes	Yes
ADC	Yes	Yes	Yes	No	Yes
ACMP	No	No	No	No	Yes
ECAP	No	No	No	No	Yes
EBWM	No	No	No	No	Yes
BPWM	No	No	No	No	Yes
HDIV	No	No	No	No	Yes

表 6.3-1 外设时钟源选择表

注意:对于那些时钟源不可选择的外设，它们的时钟源固定为PCLK。

6.3.5 掉电模式时钟

当芯片进入掉电模式时，系统时钟和一些时钟源以及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。

如下时钟仍在工作：

- 时钟发生器
- 10 kHz 内部低速振荡器时钟 (LIRC)
- 32.768 kHz 外部低速晶振时钟 (LXT) (当 PDLXT = 1 且 XTLEN[1:0] = 10)
- 外设时钟 (当模块的时钟源来自10kHz内部低速振荡器)
 - ◆ 看门狗时钟
 - ◆ 定时器 0/1 时钟

6.3.6 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。其中哪一级的值被输出，由一个16选1的多路转换器决定，该多路转换器的输出连接到CKO管脚上。因此共有16种时钟分频选择，分别从 $F_{in}/2^1$ 到 $F_{in}/2^{16}$ ，此处 F_{in} 是时钟输入频率。

输出公式： $F_{out} = F_{in}/2^{(N+1)}$ ，其中 F_{in} 为输入时钟频率， F_{out} 为通过时钟分频器的输出频率，N为

FREQSEL(CLK_CLKOCTL[3:0])的值。

将CLKOEN (CLK_CLKOCTL[4])置1，计数器开始计数。将CLKOEN (CLK_CLKOCTL[4])置0，计数器保持计数状态，直到分频时钟达到低电平并保持在低电平状态。

若将DIV1EN (CLK_CLKOCTL[5])置1，分频时钟（FRQDIV_CLK）将绕过分频器，使得分频器的时钟在CLKO管脚上直接输出。

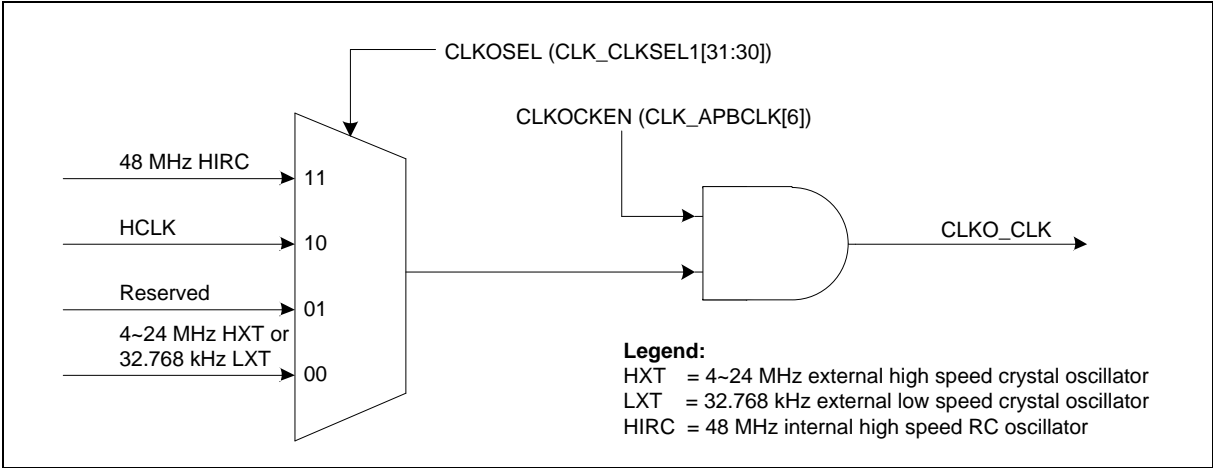


图 6.3-6 分频器的时钟源框图

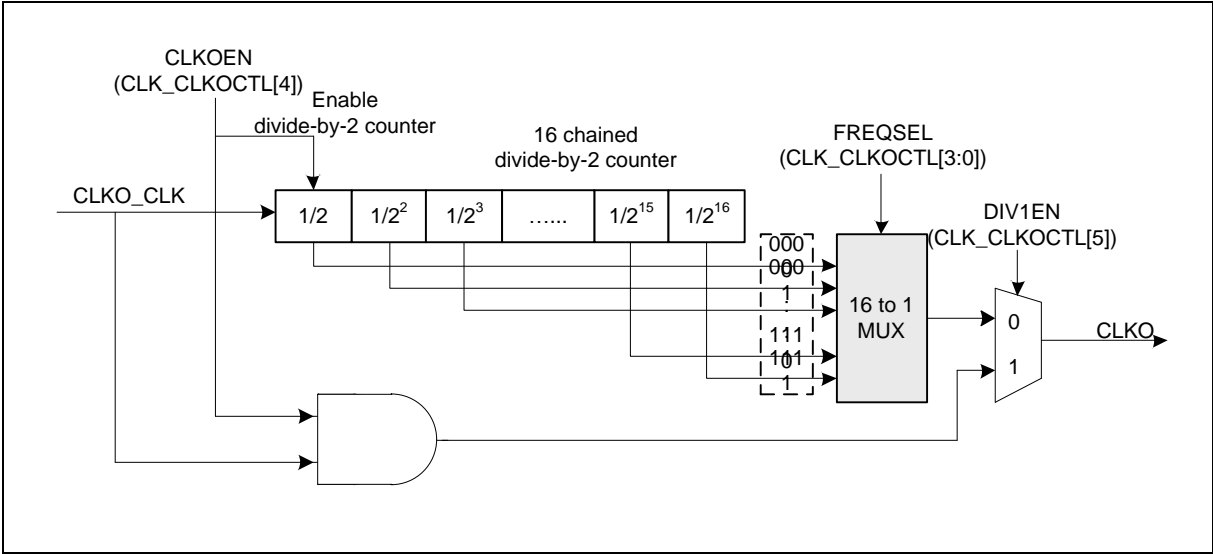


图 6.3-7 分频器框图

6.3.7 寄存器映射

R: 只读, W: 只写, R/W: 读/写

寄存器	偏移地址	读/写	描述	复位值
CLK 基地址: CLK_BA = 0x5000_0200				
CLK_PWRCTL	CLK_BA+0x00	R/W	系统掉电控制寄存器	0x0000_001C
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB 设备时钟使能控制寄存器	0x0000_0014
CLK_APBCLK	CLK_BA+0x08	R/W	APB 设备时钟使能控制寄存器	0x0000_0001
CLK_CLKSEL0	CLK_BA+0x10	R/W	时钟源选择控制寄存器0	0x0000_001B
CLK_CLKSEL1	CLK_BA+0x14	R/W	时钟源选择控制寄存器1	0xC307_7733
CLK_CLKDIV	CLK_BA+0x20	R/W	时钟分频寄存器	0x0000_0000
CLK_STATUS	CLK_BA+0x50	R	时钟状态监测寄存器	0x0000_00XX
CLK_CLKOCTL	CLK_BA+0x60	R/W	时钟输出控制寄存器	0x0000_0000

6.3.8 寄存器描述

系统掉电控制寄存器 (CLK_PWRCTL)

除BIT[6]外,其他位都是写保护的,在编程这些位之前需将0x59, 0x16, 0x88写入到地址0x5000_0100,以解锁寄存器状态。详细内容请参考 SYS_REGLCTL 寄存器,其地址为SYS_BA + 0x100。

寄存器	偏移地址	读/写	描述	复位值
CLK_PWRCTL	CLK_BA+0x00	R/W	系统掉电控制寄存器	0x0000_001C

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				HXTGAIN		PDLXT	Reserved
7	6	5	4	3	2	1	0
PDEN	PDWKIF	PDWKIEN	PDWKDLY	LIRCEN	HIRCEN	XTLEN	

位	描述	
[31:12]	Reserved	保留
[11:10]	HXTGAIN	HXT 增益控制寄存器 (写保护) 这是一个写保护的寄存器。编辑之前需先开锁。 增益控制用于增大晶振的增益以保证晶振正常工作。如果使能增益控制,晶振将消耗更多功耗。 00 = HXT 频率低于 8 MHz. 01 = HXT 频率是 8 MHz 到12 MHz之间. 10 = HXT 频率是 12 MHz 到16 MHz之间. 11 = HXT 频率高于 16 MHz. 注意: 该位是写保护. 参考 SYS_REGLCTL 寄存器.
[9]	PDLXT	掉电模式下使能 LXT 0 = 进入掉电模式后 LXT 将会自动关闭. 1 = 当 XTLEN[1:0] 是 0x2 时, LXT 在掉电模式下依旧有效.
[8]	Reserved	保留
[7]	PDEN	系统掉电模式使能位 (写保护) 当该位置1, 使能掉电模式。 当芯片从掉电模式唤醒时, 该位由硬件清除, 用户需要重新设置该位以便进入下一次的掉电模式。 在掉电模式下, 外部高速晶振(HXT)与内部高速振荡器 (HIRC) 将被禁用, 但是LXT和LIRC 依旧有效。

		<p>在掉电模式下，系统时钟将被禁用，忽略时钟源选择。如果外设时钟源为LXT或LIRC，则外设的时钟不受掉电模式的控制。</p> <p>0 = 执行 WFI/WFE命令，芯片工作于正常模式或者进入 idle 模式</p> <p>1 = 芯片进入掉电模式当 CPU 执行WFI/WFE睡眠命令</p> <p>注意: 该位是写保护. 参考 SYS_REGLCTL 寄存器.</p>
[6]	PDWKIF	<p>芯片掉电模式唤醒中断状态</p> <p>该位由掉电唤醒事件置位，表示从掉电模式中恢复。</p> <p>如果GPIO, USCI01, WDT, ACMP01, BOD, TMR01唤醒事件发生时，该标志位置1。</p> <p>注意1: 该位写1清0。</p> <p>注意 2: 只有在PDWKEN (CLK_PWRCTL[5]) 被置 1 的时候，该位置1。</p>
[5]	PDWKIEN	<p>掉电模式唤醒中断使能位 (写保护)</p> <p>0 = 禁用掉电模式唤醒中断</p> <p>1 = 使能掉电模式唤醒中断</p> <p>注意1: 当PDWKIF 和 PDWKIEN都为1时，该中断将产生。</p> <p>注意2: 该位是写保护. 参考 SYS_REGLCTL 寄存器.</p>
[4]	PDWKDLY	<p>唤醒延时计数器使能位 (写保护)</p> <p>当芯片从掉电模式中唤醒时，时钟控制将会延迟一定的时钟周期以保证系统时钟稳定。</p> <p>当芯片工作在外部4~24 MHz高速晶振时，延迟时钟周期为4096 个时钟周期；当芯片工作在内部48 MHz 内部高速振荡器 (HIRC) 时，延迟时钟周期为 256个时钟周期。</p> <p>0 = 禁用时钟周期的延迟</p> <p>1 = 使能时钟周期的延迟</p> <p>注意: 该位是写保护. 参考 SYS_REGLCTL 寄存器.</p>
[3]	LIRCEN	<p>LIRC 使能位 (写保护)</p> <p>0 = 禁用10 kHz 内部低速 RC 振荡器(LIRC)</p> <p>1 = 使能10 kHz内部低速 RC 振荡器(LIRC)</p> <p>注意: 该位是写保护. 参考 SYS_REGLCTL 寄存器.</p>
[2]	HIRCEN	<p>HIRC 使能位 (写保护)</p> <p>0 = 禁用48 MHz 内部高速RC振荡器(HIRC)</p> <p>1 = 使能48 MHz 内部高速RC振荡器(HIRC)</p> <p>注意: 该位是写保护. 参考 SYS_REGLCTL 寄存器.</p>
[1:0]	XTLEN	<p>XTL 使能位 (写保护)</p> <p>默认值为“00”，且XT1_IN、XT1_OUT管脚默认为GPIO功能。</p> <p>00 = XT_IN和XT_OUT为GPIO, 禁止LXT&HXT(默认值)。</p> <p>01 = HXT使能。</p> <p>10 = LXT使能。</p> <p>11 = XT_IN为外部时钟输入管脚、XT_OUT为GPIO管脚。</p> <p>注意: 该位是写保护. 参考 SYS_REGLCTL 寄存器.</p>

寄存器模式/指令模式	SLEEPDEEP (SCS_SCR[2])	PDEN (CLK_PWRCTL [7])	CPU 运行 WFI 指令	时钟禁用
正常运行模式	0	0	NO	所有时钟皆可被禁用
空闲模式 (CPU 进入休眠模式)	0	0	YES	仅禁用CPU时钟
掉电模式 (CPU 进入深度休眠模式)	1	1	YES	当时钟源为LIRC/LXT时，大部分 时钟被禁用，仅LIRC/LXT时钟源 和RTC/WDT/Timer外设时钟能工 作。

表 6.3-2 掉电模式控制表

当芯片进入掉电模式时，用户可以通过一些中断源唤醒芯片。用户必须在设置 PDEN 位 (CLK_PWRCTL[7])之前，使能相关的中断源及相应的NVIC IRQ 使能位(NVIC_ISER) 以保证芯片能进入掉电模式并能够成功被唤醒。

AHB设备时钟使能控制寄存器 (CLK_AHBCLK)

该寄存器的各位用于使能/禁用系统时钟，AHB总线设备时钟。

寄存器	偏移地址	读/写	描述	复位值
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB设备时钟使能控制寄存器	0x0000_0014

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			HDIVCKEN	Reserved	ISPCKEN	Reserved	

位	描述	
[31:5]	Reserved	保留
[4]	HDIVCKEN	硬件除法器时钟控制使能位 0 = HDIV 外设时钟禁用. 1 = HDIV 外设时钟使能
[3]	Reserved	保留.
[2]	ISPCKEN	Flash ISP 时钟控制使能位 0 = Flash ISP 外设时钟禁用. 1 = Flash ISP 外设时钟使能.
[1:0]	Reserved	保留.

APB 设备时钟使能控制寄存器 (CLK_APBCLK)

该寄存器用于使能/禁用外设控制器时钟

寄存器	偏移地址	读/写	描述	复位值
CLK_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	ACMPCKEN	Reserved	ADCCKEN	Reserved		USCI1CKEN	USCI0CKEN
23	22	21	20	19	18	17	16
BPWMCKEN	Reserved		EPWMCKEN	Reserved			
15	14	13	12	11	10	9	8
Reserved			PGACKEN	Reserved		ECAPCKEN	
7	6	5	4	3	2	1	0
Reserved	CLKOCKEN	Reserved		TMR1CKEN	TMR0CKEN	Reserved	WDTCKEN

位	描述	
[31]	Reserved	保留。
[30]	ACMPCKEN	模拟比较器时钟使能位 0 = 禁用模拟比较器时钟 1 = 使能模拟比较器时钟
[29]	Reserved	保留。
[28]	ADCCKEN	模拟数字转换器(ADC) 时钟使能位 0 = 禁用ADC外设时钟 1 = 使能ADC外设时钟
[27:26]	Reserved	保留。
[25]	USCI1CKEN	USCI1 时钟使能位 0 = 禁用USCI1 时钟 1 = 使能USCI1 时钟
[24]	USCI0CKEN	USCI0 时钟使能位 0 = 禁用 USCI0 时钟 1 = 使能USCI0 时钟
[23]	BPWMCKEN	基本 PWM 0/1 通道时钟使能位 0 = 禁用 BPWM 0/1 通道时钟 1 = 使能 BPWM 0/1 通道时钟
[22:21]	Reserved	保留。
[20]	EPWMCKEN	加强型PWM时钟使能位

		0 = 禁用EPWM 时钟. 1 = 使能EPWM 时钟.
[19:13]	Reserved	保留.
[12]	PGACKEN	PGA 时钟使能位 0 = 禁用 PGA 时钟. 1 = 使能PGA 时钟.
[11:9]	Reserved	保留.
[8]	ECAPCKEN	输入捕获时钟使能位 0 = 禁用 ECAP 时钟. 1 = 使能ECAP 时钟.
[7]	Reserved	保留
[6]	CLKOCKEN	CLKO 时钟使能位 0 = 禁用CLKO 时钟. 1 = 使能CLKO 时钟.
[5:4]	Reserved	保留.
[3]	TMR1CKEN	Timer1时钟使能位 0 = 禁用Timer1 时钟. 1 = 使能Timer1 时钟.
[2]	TMR0CKEN	Timer0时钟使能位 0 = 禁用Timer0 时钟. 1 = 使能Timer0 时钟.
[1]	Reserved	保留.
[0]	WDTCKEN	Watchdog 定时器时钟使能位 (写保护) 0 = 禁用Watchdog 时钟. 1 =使能 Watchdog 时钟. 注意: 该位是写保护. 参考 SYS_REGLCTL 寄存器.

时钟源选择控制寄存器0 (CLK_CLKSEL0)

寄存器	偏移地址	读/写	描述	复位值
CLK_CLKSEL0	CLK_BA+0x10	R/W	时钟源选择控制寄存器0	0x0000_001B

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			STCLKSEL		Reserved	HCLKSEL	

位	描述	
[31:5]	Reserved	保留.
[4:3]	STCLKSEL	<p>Cortex®-M0 SysTick 时钟源选择 (写保护)</p> <p>如果 SYST_CTL[2]=0, SysTick 时钟源配置如下:</p> <p>00 = 时钟源为 HXT/LXT.</p> <p>01 = 时钟源为(HXT 或 LXT)/2.</p> <p>10 = 时钟源为 HCLK/2.</p> <p>11 = 时钟源为 HIRC/2.</p> <p>其它 = 保留.</p> <p>注意: 如果 SysTick 时钟源不是来自 HCLK (例如 SYST_CTL[2] = 0),SysTick 时钟源必须小于或等于HCLK/2。</p> <p>注意: 该位是写保护. 参考 SYS_REGLCTL 寄存器.</p>
[2]	Reserved	保留.
[1:0]	HCLKSEL	<p>HCLK 时钟源选择 (写保护)</p> <p>在时钟切换前, 相关时钟源 (预选和新选) 必须打开并保持稳定。</p> <p>00 = 时钟源为 HXT/LXT.</p> <p>01 = 时钟源为 LIRC.</p> <p>11= 时钟源为 HIRC.</p> <p>其它 = 保留.</p> <p>注意: 该位是写保护. 参考 SYS_REGLCTL 寄存器.</p>

时钟源选择控制寄存器1 (CLK_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	读/写	描述	复位值
CLK_CLKSEL1	CLK_BA+0x14	R/W	时钟源选择控制寄存器1	0xC307_7733

31	30	29	28	27	26	25	24
CLKOSEL		Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		TMR1SEL		Reserved		TMR0SEL	
7	6	5	4	3	2	1	0
Reserved		ADCSEL		Reserved		WDTSEL	

位	描述	
[31:30]	CLKOSEL	时钟分频器时钟源选择位 00 = 时钟源为外部晶振 (HXT 或 LXT). 01 = 保留. 10 = 时钟源为HCLK. 11 = 时钟源为 48 MHz 内部高速RC振荡器 (HIRC).
[29:15]	Reserved	保留.
[14:12]	TMR1SEL	TIMER1 时钟源选择 000 = 时钟源为外部晶振 (HXT 或 LXT). 001 = 时钟源为 10 kHz 内部低速RC振荡器 (LIRC). 010 = 时钟源为 HCLK. 011 = 时钟源为外部时钟管脚 T1 111 = 时钟源为 48 MHz 内部高速RC振荡器 (HIRC). 其它 = 保留.
[11]	Reserved	保留.
[10:8]	TMR0SEL	TIMER0 时钟源选择 000 =时钟源为外部晶振 (HXT 或 LXT). 001 =时钟源为 10 kHz 内部低速RC振荡器 (LIRC). 010 =时钟源为 HCLK. 011 =时钟源为外部时钟管脚 T0 111 =时钟源为 48 MHz 内部高速RC振荡器 (HIRC). 其它 = 保留.

[7:6]	Reserved	保留.
[5:4]	ADCSEL	ADC 外设时钟源选择 00 =时钟源为外部晶振 (HXT 或 LXT). 01 = 保留. 10 =时钟源为 HCLK. 11 =时钟源为 48 MHz 内部高速RC振荡器 (HIRC).
[3:2]	Reserved	保留.
[1:0]	WDTSEL	Watchdog 定时器时钟源选择 (写保护) 00 =时钟源为外部晶振 (HXT 或 LXT). 01 = 保留. 10 = 时钟源为 HCLK/2048. 11 =时钟源为 10 kHz 内部低速RC振荡器 (LIRC). 注意: 该位是写保护. 参考 SYS_REGLCTL 寄存器.

时钟分频寄存器 (CLK_CLKDIV)

寄存器	偏移地址	读/写	描述	复位值
CLK_CLKDIV	CLK_BA+0x20	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
ADCDIV							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				HCLKDIV			

位	描述	
[31:24]	Reserved	保留.
[23:16]	ADCDIV	ADC 时钟分频值 来自 ADC 时钟源 ADC 时钟频率 = (ADC 时钟源频率) / (ADCDIV + 1).
[15:4]	Reserved	保留.
[3:0]	HCLKDIV	HCLK 时钟分频值来自 HCLK 时钟源 HCLK 时钟频率 = (HCLK 时钟源频率) / (HCLKDIV + 1).

时钟状态寄存器 (CLK_STATUS)

该寄存器用于监控芯片时钟源是否稳定，以及时钟源切换是否失败。

寄存器	偏移地址	读/写	描述	复位值
CLK_STATUS	CLK_BA+0x50	R	时钟状态监测寄存器	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
CLKSFALL	Reserved		HIRCSTB	LIRCSTB	Reserved		XTLSTB

位	描述	
[31:8]	Reserved	保留。
[7]	CLKSFALL	<p>时钟切换失败标志（只读）</p> <p>当软件切换系统时钟源时，该位被更新。如果切换目标时钟稳定，该位将被置0。如果切换目标时钟不稳定，该位将被设置为1。</p> <p>0 = 时钟切换成功</p> <p>1 = 时钟切换失败</p> <p>注意：该位写1清0。</p>
[6:5]	Reserved	保留。
[4]	HIRCSTB	<p>HIRC 时钟源稳定标志（只读）</p> <p>0 = 48MHz内部高速RC振荡器（HIRC）时钟不稳定或者禁用</p> <p>1 = 48MHz内部高速RC振荡器（HIRC）时钟使能并稳定</p>
[3]	LIRCSTB	<p>LRC 时钟源稳定标志（只读）</p> <p>0 = LIRC 时钟不稳定或者禁用</p> <p>1 = LIRC 时钟使能并稳定</p>
[2:1]	Reserved	保留。
[0]	XTLSTB	<p>XTL 时钟源稳定标志（只读）</p> <p>0 = HXT 或 LXT 时钟不稳定或者禁用</p> <p>1 = HXT 或 LXT 时钟使能并稳定</p>

时钟输出控制寄存器 (CLK_CLKOCTL)

寄存器	偏移地址	读/写	描述	复位值
CLK_CLKOCTL	CLK_BA+0x60	R/W	时钟输出控制寄存器	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		DIV1EN	CLKOEN	FREQSEL			

位	描述	
[31:6]	Reserved	保留。
[5]	DIV1EN	时钟输出分频器1使能位 0 = 分频器的输出频率由FREQSEL的值决定 1 = 分频器输出频率与输入时钟频率相同
[4]	CLKOEN	时钟输出使能位 0 = 禁用时钟输出功能 1 = 使能时钟输出功能
[3:0]	FREQSEL	时钟输出频率选择位 输出频率的公式： $F_{out} = F_{in} / 2^{(N+1)}$ F _{in} 为输入时钟频率 F _{out} 为分频器输出时钟频率 N为FREQSEL[3:0]的值。

6.4 存储控制器 (FMC)

6.4.1 概述

NM1120系列具有29.5K字节片上Flash，应用程序存储器（APROM），可以通过ISP更新程序。当芯片已经贴完片后，用户可以通过在系统编辑功能（ISP）来更新程序存储器。芯片上电后，Cortex®-M0 CPU通过config0（CBS）确定启动项后决定从APROM或LDROM获取代码。NM1120系列提供Data Flash区域，支持与APROM共享空间，Data Flash大小可根据应用需求通过config1配置。安全保护存储器（SPROM）是保护用户编程代码在SPROM里。

6.4.2 特性

- 对于不连续的地址访问，48MHz需要等待状态，24MHz无需等待状态
- 支持29.5K字节应用程序存储空间（APROM）
- 支持2K引导存储器（LDROM）
- 大小可配置的数据Flash，支持512字节页擦除
- 支持3个512字节可编程加密存储器（SPROM）
- 支持在系统编程（ISP）/在应用编程（IAP）来更新片上Flash

6.4.3 框图

存储控制器包括AHB Slave接口，ISP逻辑控制，writer接口和Flash接口时序逻辑控制。存储控制器的框图如图 6.4-1所示。

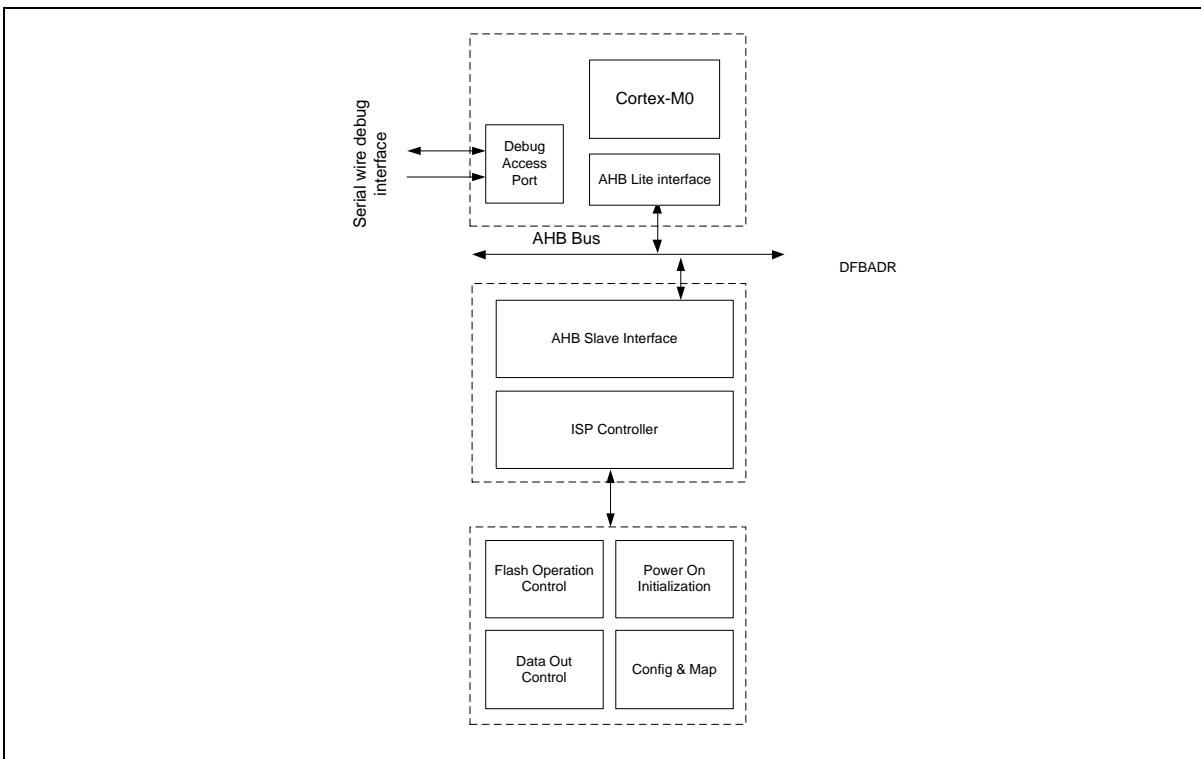


图 6.4-1 存储控制器框图

6.4.4 功能描述

6.4.4.1 存储器组织

NM1120 存储器由APROM, Data Flash, LDROM和用户配置区组成。

用户应用可编程存储器称为APROM。用户可以读写APROM和设置系统从APROM启动。

ISP程序存储器是为加载引导程序而设计的，通过在系统编程功能实现。LDROM与APROM是独立的并且系统也可以设置开机从LDROM启动。因此，当APROM中代码损坏时用户可以通过LDROM启动，避免系统启动失败。

Data Flash用于用户存储数据。它可以通过ISP或存储器来读取并通过ISP寄存器编程。每个擦除单元的大小为512字节。Data Flash与应用程序存储器共享，大小和起始地址由用户根据应用需求来设置。

用户配置提供多个字节来控制系统，如Flash安全锁、启动选择、欠压电平设置和数据flash基地址设置等...用户配置类似于保险丝在上电时的工作原理，在上电的时候，用户配置区设定被加载到相应的控制寄存器。

在NuMicro®家族，flash存储器映射有别于系统存储器映射。当CPU从FMC存储器中获取代码或数据时，使用系统存储器映射。当用户用ISP功能去读取、编程、或者擦除FMC存储器时，使用flash存储器映射。例如，当系统设置启动项为LDROM通过CBS [1:0] = 1设置，CPU将获取的指令地址在LDROM的0x0000 ~ 0x07ff位置。然而，如果用户想通过ISP读LDROM，需要访问LDROM的地址0x0010_0000 ~ 0x0010_07ff。

表 6.4-1 展示了APROM、LDROM、Data Flash和用户配置区的地址映射信息。

区域名称	DFEN	大小	起始地址	结束地址
APROM	0	(29.5-0.5*N) Kbytes	0x0000_0000	DFBA-1
APROM	1	29.5 Kbytes	0x0000_0000	0x0000_75FF
Data Flash	0	0.5*N Kbytes	DFBA	0x0000_75FF
Data Flash	1	N/A	N/A	N/A
LDROM	x	2 Kbytes	0x0010_0000	0x0010_07FF
SPROM0	x	0.5 Kbytes	0x0020_0000	0x0020_01FF
SPROM1	x	0.5 Kbytes	0x0024_0000	0x0024_01FF
SPROM2	x	0.5 Kbytes	0x0028_0000	0x0028_01FF
用户配置区	x	2 words	0x0030_0000	0x0030_0004

表 6.4-1 Flash 存储器的地址映射

Flash存储器组织如图 6.4-2所示:

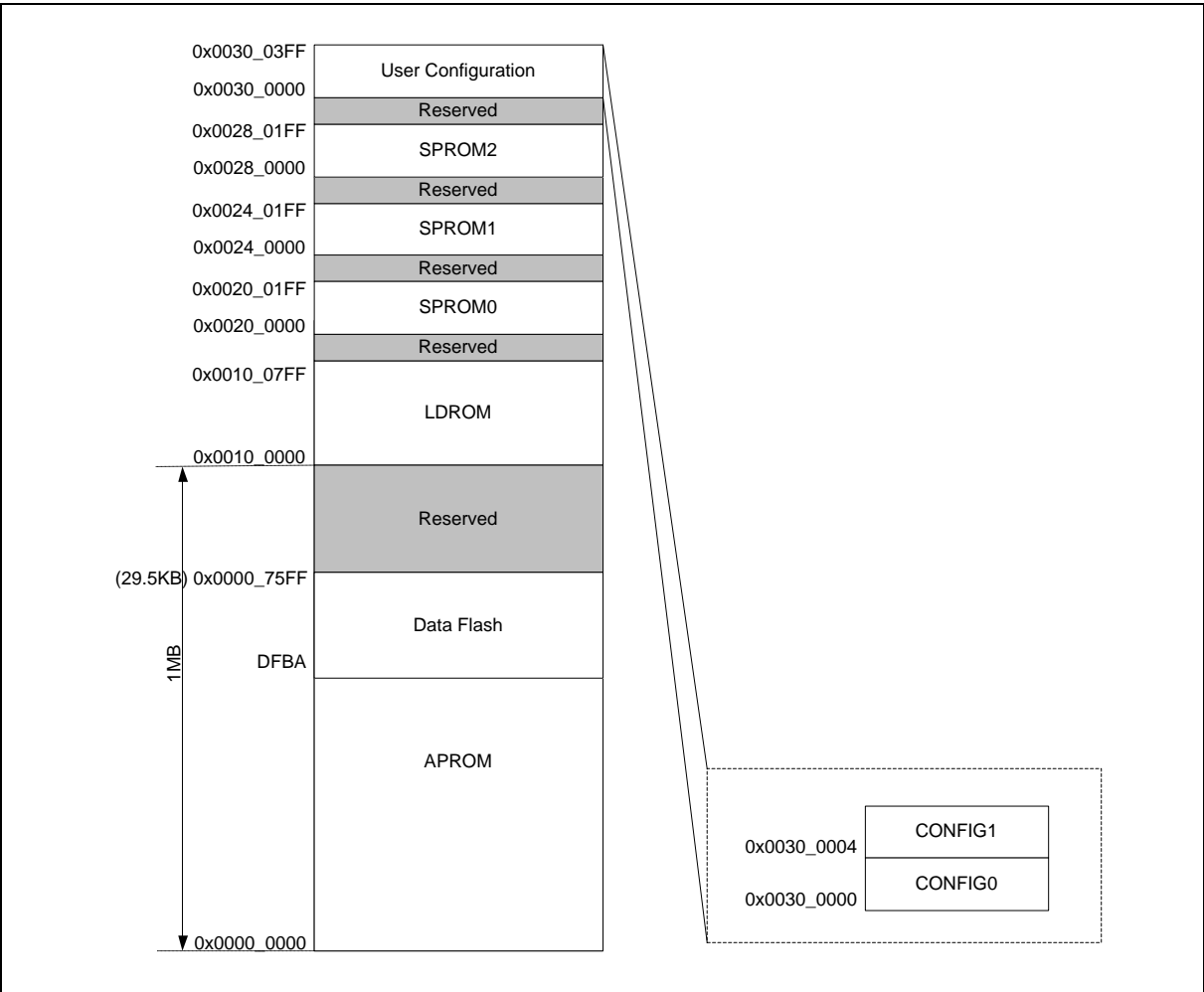


图 6.4-2 Flash 存储器组织

6.4.4.2 数据 Flash

NM1120系列提供了数据Flash。擦除单元为512字节，当一个字被改变时，所有128个字都需要提前复制到另一个页或SRAM中。当DFEN（Config0）被使能后数据Flash的基地址由寄存器DFBA（CONFIG1[13:0]）设定。例如，对于4K/2K/1K/0KB数据Flash的DFBA设定值，参考表 6.4-2 所示。

Data Flash APROM	4KB (DFEN=0)	2KB (DFEN=0)	1KB (DFEN=0)	0KB (DFEN=1)
29.5K Flash	DFBA=0x0000_6600	DFBA=0x0000_6E00	DFBA=0x0000_7200	DFEN=1

表 6.4-2 数据 Flash 表格

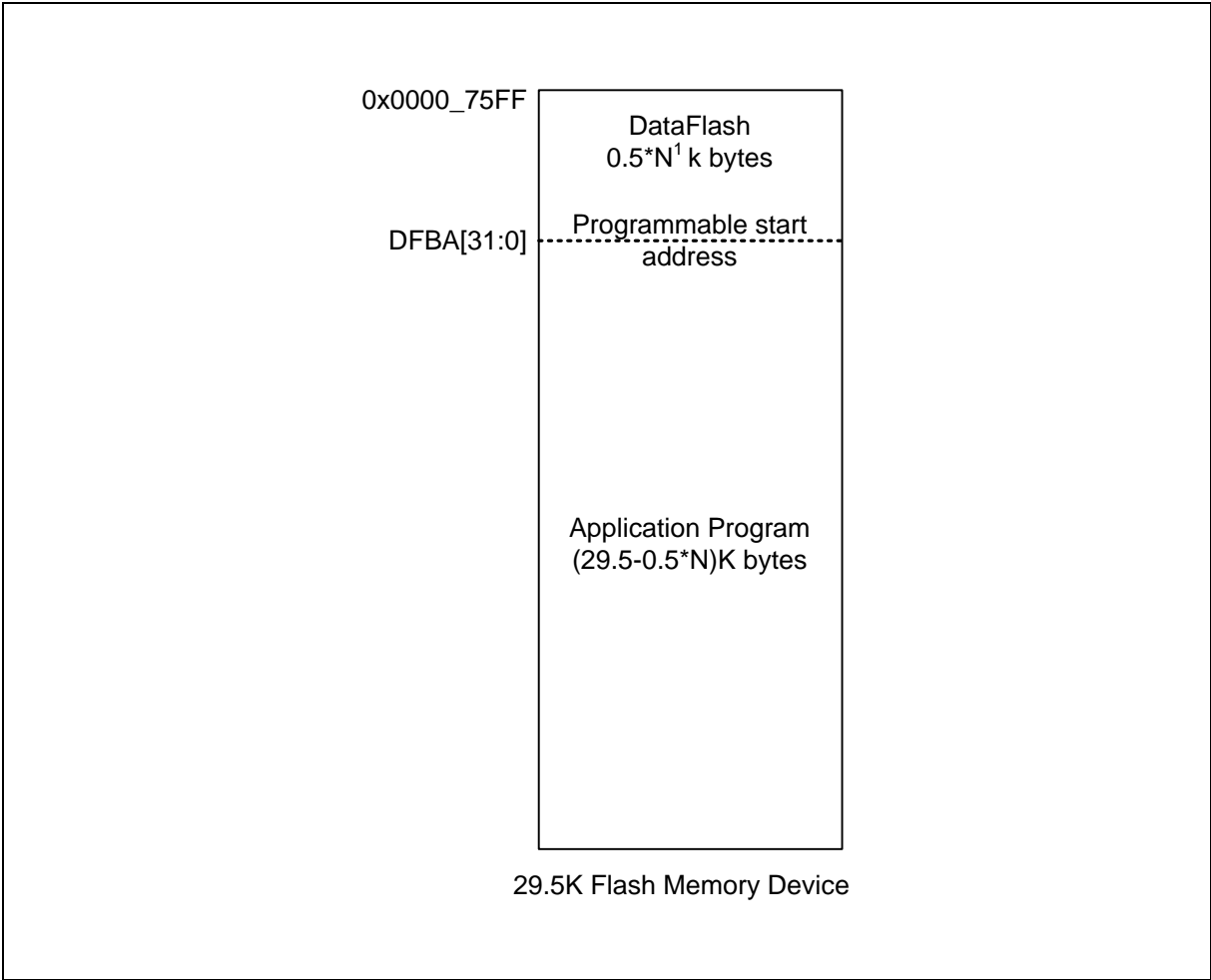


图 6.4-3 数据 Flash 与 APROM 共享存储空间

6.4.4.3 安全保护存储器 (SPROM)

NM1120系列提供安全保护存储器（SPROM）用于存储安全指令，是通过ISP和ICE进行读写，不支持整片擦除命令，但是支持页擦除命令。SPROM通过最后一字节来确定代码是处于安全模式或非安全模式。请参考表格 6.4-4所示，这表明安全保护存储器只允许CPU在执行安全代码时执行指令读取和页面擦除操作。

- （最后一字节= 0xFF): 非安全模式

	ICE Debug	ISP/IAP	CPU Data	CPU Instruction
整个芯片擦除	-	-	-	-
页擦除	-	√	-	-
编程	-	√	-	-
读	√	√	√	√

- (最后一字节=其它): 安全模式

	ICE Debug	ISP/IAP	CPU Data	CPU Instruction
整个芯片擦除	-	-	-	-
页擦除	-	√	-	-
编辑	-	-	-	-
读	00h	00h	00h	CPU Instruction

	SPROM0/1/2			
	0x200000 ~ 0x2001FF/0x240000 ~ 0x2401FF/0x280000 ~ 0x2801FF			
	ISP/IAP/ICP/Writer		ICE	
	Lock	unLock	Lock	unLock
整个芯片擦除	-	-	-	-
页擦除	√	√	-	-
编辑	-	√	-	-
读指令	√	√	√	√
读数据	00h	√	00h	√

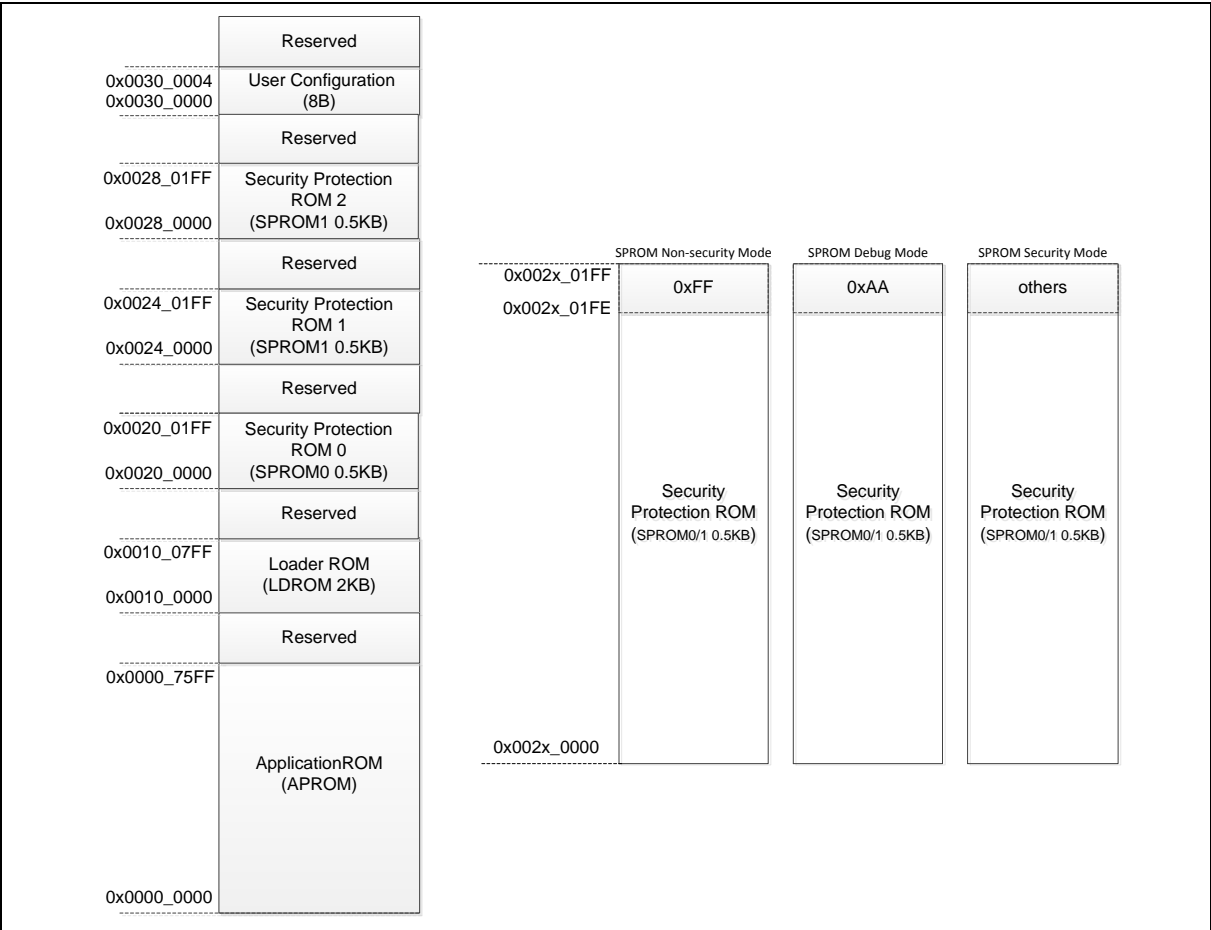


图 6.4-4 SPROM 安全锁模式

6.4.4.4 用户配置区

用户配置区是一个内部可编程的配置区域，用于启动选项。用户配置区位于地址0x300000，有2个32位的寄存器。用户更改后的配置将在系统重启后生效。

Config0 (地址 = 0x0030_0000)

31	30	29	28	27	26	25	24
Reserved				GPA5RINI		GPA4RINI	
23	22	21	20	19	18	17	16
GPA3RINI		GPA2RINI		GPA1RINI		GPA0RINI	
15	14	13	12	11	10	9	8
CBOV			CBORST	CBODEN	CIOINI	Reserved	
7	6	5	4	3	2	1	0
CBS		Reserved				LOCK	DFEN

Config0	地址 = 0x0030_0000	
位	描述	
[31:28]	Reserved	保留.
[27:26]	GPA5RINI	上电上拉电阻初始状态选择 00 = 使能下拉电阻 01 = 使能上拉电阻 1x = 禁用上拉/下拉电阻 上电后GPA5 设置该模式
[25:24]	GPA4RINI	上电上拉电阻初始状态选择 00 = 使能下拉电阻 01 = 使能上拉电阻 1x = 禁用上拉/下拉电阻 上电后GPA4 设置该模式
[23:22]	GPA3RINI	上电上拉电阻初始状态选择 00 = 使能下拉电阻 01 = 使能上拉电阻 1x = 禁用上拉/下拉电阻 上电后GPA3 设置该模式
[21:20]	GPA2RINI	上电上拉电阻初始状态选择 00 = 使能下拉电阻 01 = 使能上拉电阻 1x = 禁用上拉/下拉电阻 上电后GPA2 设置该模式
[19:18]	GPA1RINI	上电上拉电阻初始状态选择 00 = 使能下拉电阻 01 = 使能上拉电阻 1x = 禁用上拉/下拉电阻 上电后GPA1 设置该模式

Config0	地址 = 0x0030_0000	
位	描述	
[17:16]	GPA0RINI	<p>上电上拉电阻初始状态选择</p> <p>00 = 使能下拉电阻</p> <p>01 = 使能上拉电阻</p> <p>1x = 禁用上拉/下拉电阻</p> <p>上电后GPA0 设置该模式</p>
[15:13]	CBOV	<p>欠压选择</p> <p>欠压选择如下:</p> <p>000 = 2.0V.</p> <p>001 = 2.2V.</p> <p>010 = 2.4V.</p> <p>011 = 2.7V.</p> <p>100 = 3.0V.</p> <p>101 = 3.7V.</p> <p>110 = 4.0V.</p> <p>111 = 4.3V.</p>
[12]	CBORST	<p>欠压复位使能位</p> <p>0 = 上电后, 使能欠压复位</p> <p>1 = 上电后, 禁用欠压复位</p>
[11]	CBODEN	<p>欠压使能位</p> <p>0 = 使能欠压检测</p> <p>1 = 禁用欠压检测</p>
[10]	CIOINI	<p>I/O初始状态选择</p> <p>0 = 上电模式</p> <p>1 = 三态输入模式</p> <p>注意: 上电后所有IO口的模式选择</p>
[9:8]	Reserved	保留.
[7:6]	CBS	<p>芯片启动选择</p> <p>00=由LDROM启动支持IAP功能</p> <p>01=由LDROM启动不支持IAP功能</p> <p>10=由APROM启动支持IAP功能</p> <p>11=由APROM启动不支持IAP功能</p> <p>NM1120系列, 在IAP模式下, 当CBS[0] = 0, LDROM的基地址映射到0x100000, APROM的基地址映射到0x0. 用户既可以访问APROM也可以访问LDROM, 无需启动切换。换句话说, 如果是IAP模式, 存放在LDROM与APROM的代码可以互相调用。</p> <p>注意1: 当CBS[0] = 1, BS (FMC_ISPCTL)只用于启动切换的控制。</p> <p>注意2: 当CBS[0] = 0, VECMAP只用于地址0x0~0x1ff重新映射。</p>
[5:2]	Reserved	保留.
[1]	LOCK	<p>安全锁</p> <p>0 = 锁定Flash存储器内容</p>

Config0	地址 = 0x0030_0000	
位	描述	
		1 = 解锁Flash存储器内容 当Flash数据被锁，通过ICP串行调试接口或烧录器仅设备ID，UID，CRC校验和用户配置能被读，其它数据都是被锁为0xFFFFFFFF。不管LOCK锁值，ISP可以读取任何位置的数据
[0]	DFEN	数据 Flash 使能位 0 = 使能数据Flash 1 = 禁用数据Flash

注意: 用户配置区的保留位应保持为 “1”。

Config1 (地址 = 0x0030_0004)

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		DFBA					
7	6	5	4	3	2	1	0
DFBA							

Config1	地址 = 0x0030_0004	
位	描述	
[31:14]	Reserved	保留
[13:0]	DFBA	数据Flash 基地址 数据Flash基地址由用户定义。因为片上FLASH擦除单位为512字节，所以强制DFBA的第8-0位置为0。

6.4.4.5 欠压检测

NM1120带有欠压检测功能为了监控V_{DD} 引脚的电压情况。如果V_{DD} 电压低于CBOV设置的电压值，当BOD功能使能后BOD事件会被触发。用户可以通过CBORST使能BOD复位或者通过NVIC使能BOD中断。。因为当V_{DD}电压低于CBOV设定的电平时BOD复位会发生，BOD复位使能后用户需确认CBOV的设定值避免BOD复位被重复启用。

6.4.4.6 启动选择

NM1120提供在系统编程功能，当PCB板贴完片后可通过ISP更新代码。2Kbytes的LDROM用于存储ISP引导程序。用户可通过CBS（CONFIG0）选择从APROM还是LDROM启动。



图 6.4-5 上电后启动项 (BS) 示意图

CBS[1:0]	启动项
00	LDROM 带 IAP 功能 从LDROM启动，程序可执行的范围是SPROM、LDROM和APROM（APROM的第一页区域除外）。 LDROM的地址映射是0x0010_0000 ~ 0x0010_07FF，同时LDROM的首页512字节地址映射到0x0000_0000 ~ 0x0000_01FF。 在该模式下APROM和LDROM都可以被编程，无论当前运行在APROM还是LDROM区域。
01	LDROM 不带 IAP 功能 从LDROM启动，通过ISP命令程序可执行范围是SPROM、LDROM和APROM。 在该模式下LDROM写保护。
10	APROM 带 IAP 功能 从 APROM 启动，程序可执行的范围包括 SPROM、LDROM 和 APROM。LDROM 地址映射到 0x0010_0000~0x0010_07FF。 在该模式下APROM和LDROM都可以被编程，无论当前运行在APROM还是LDROM区域。
11	APROM 不带 IAP 功能 从APROM启动，通过ISP命令程序可执行范围是SPROM、LDROM和APROM。 在该模式下APROM写保护。

表 6.4-3 启动选项

CBS[1:0]	启动项	向量映射	运行在LDROM写APROM	运行在APROM写LDROM	运行在LDROM写LDROM	运行在APROM写APROM
00	LDROM	Yes	Yes	-	Yes	-
01	LDROM	-	Yes	-	Yes	-
10	APROM	Yes	-	Yes	-	Yes
11	APROM	-	-	Yes	-	Yes

表格 6.4-4 启动选择和支持功能

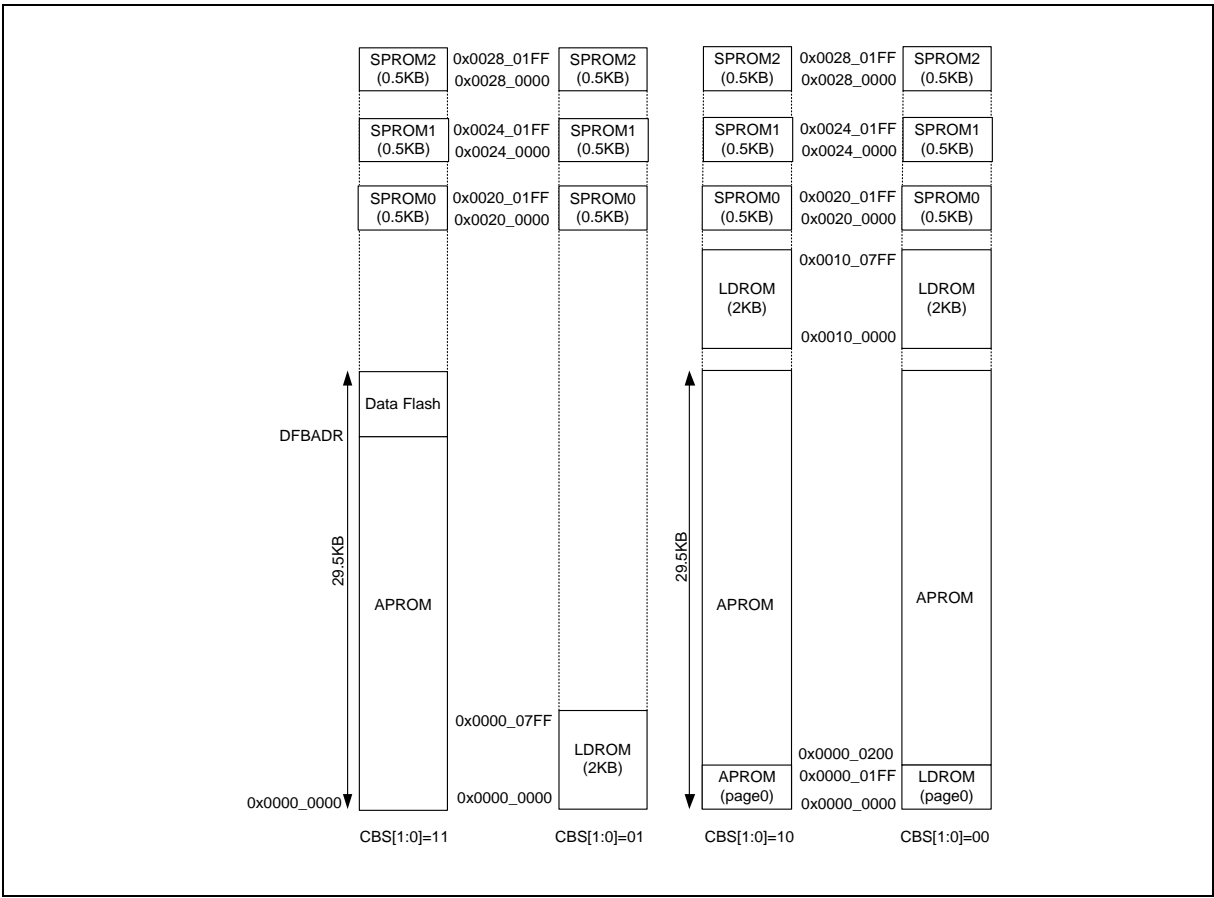


图 6.4-6 Flash 存储器的映射通过设置 CONFIG0 (CBS)

6.4.4.7 在应用编程

NM1120系列提供了在应用编程(IAP)功能，用户可以切换APROM和LDROM之间的代码无需复位。用户可以使能IAP功能，通过设定芯片的启动位寄存器Config0 (CBS[1:0])等于10 或00。

当使能IAP功能从APROM启动(CBS[1:0] = 10'b),代码可运行在LDROM和APROM。APROM的地址空间为原始大小，但是LDROM的2KB地址空间映射到0x0010_0000~ 0x0010_07FF。

芯片从LDROM启动带IAP功能(CBS[1:0] = 00'b)，代码可执行的范围包括LDROM和APROM，除了APROM的第一页。用户不能访问APROM的第一页是因为LDROM的首页默认要映射到APROM第一页。同时，LDROM的2KB地址空间映射到0x0010_0000~ 0x0010_07FF。

请参考 图 6.4-7 带IAP功能的地址映射

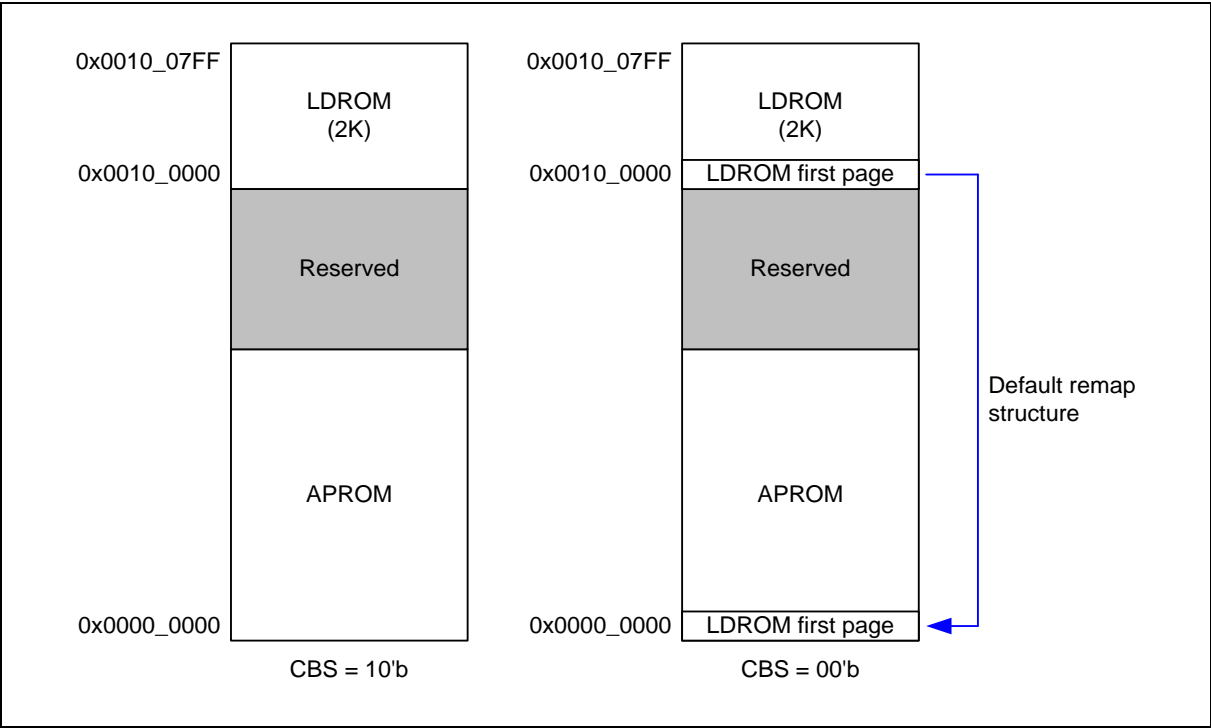


图 6.4-7 使能 IAP 功能代码可执行区域

当支持IAP功能的芯片启动模式使能，任何可以执行的代码都可以随时映射到系统内存向量(0x0000_0000~0x0000_01FF)。用户可以改变重映射地址到FMC_ISPADDR，然后用”向量重映射”命令触发ISP流程。改变向量地址后，用户可通过读FMC_ISPSTS (VECMAP) 检查是否成功更改。

6.4.4.8 在系统编程 (ISP)

NM1120系列支持在系统编程，当下载或编程失败时，在软件的控制下允许设备被重新编程避免了系统失败的风险。此外，更新应用程序固件的能力被广泛的应用成为可能。

支持在系统编程，NM1120包含LDROM和ISP控制器。用户可以在LDROM中存放ISP引导程序，并且可通过ISP寄存器编辑应用代码（APROM）。使用ISP功能可以通过软件控制在目标板上直接编程。通过ISP方式加载，让硬件接口接收新的代码变得更加容易。执行ISP最常见的方法是通过UART方式需在LDROM中装载ISP引导程序。一般来说，PC通过串口传输新的APROM代码。此时ISP Loader接收它并通过ISP命令重新编辑到APROM中。

ISP 寄存器控制步骤

NM1120支持从APROM或LDROM启动，在初始化时通过用户配置来定义。用户配置的更改需要重启系统后才会生效。如果用户想在APROM或LDROM之间切换而不改变用户配置（CBS[0] = 1），需要控制FMC_ISPCTL寄存器的BS位，然后通过SYS_IPRST1寄存器复位CPU。通过BS位切换启动项流程图如图 6.4-8所示，通过BS位切换启动项仅在CBS[0] = 1时有效。

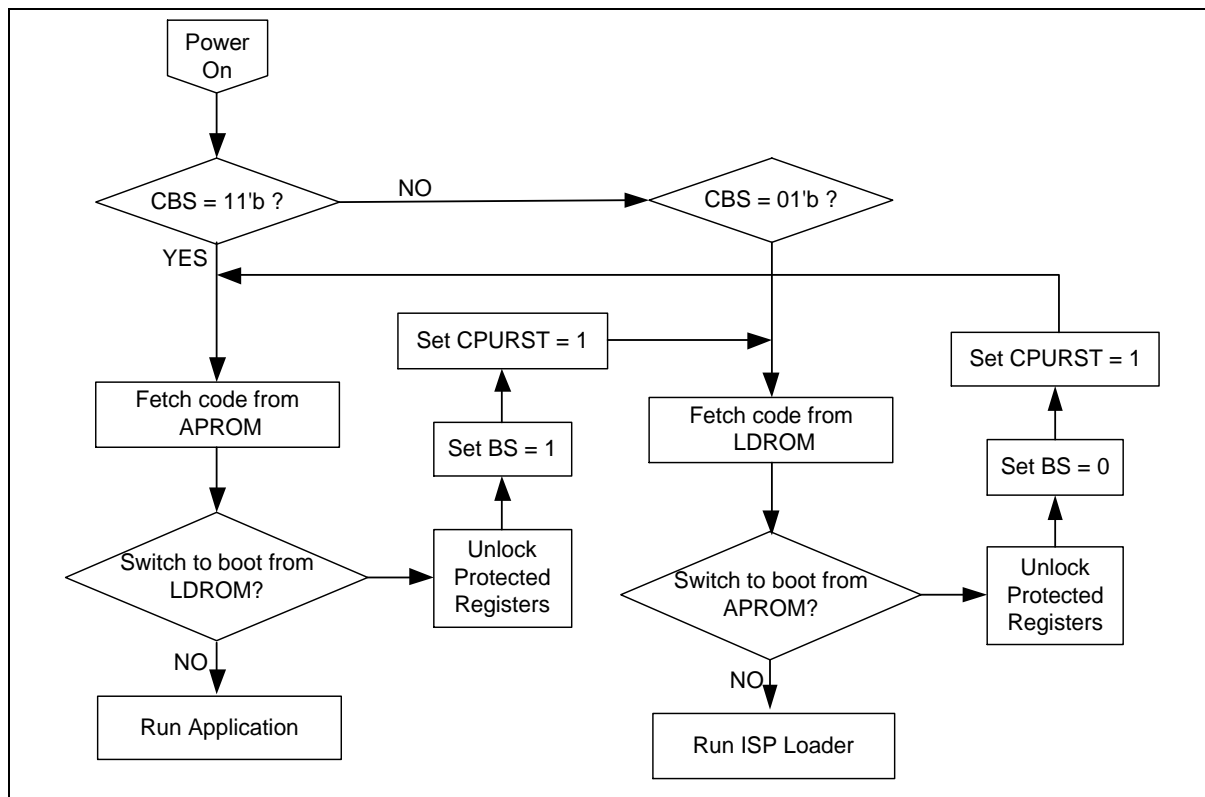


图 6.4-8 当 CBS[0] = 1 时启动选择的流程例图

在LDROM通过软件更新APROM或在APROM通过软件更新LDROM能够避免由于更新失败而导致系统错误。

ISP控制器支持对Flash存储器的读取、擦除和编程。ISP控制器的几个控制位是写保护的，因此在设置它们之前必须进行解锁。解锁保护寄存器位，软件需要按顺序写0x59，0x16和0x88到REGWRPROT。如果寄存器解锁成功，REGWRPROT的值将置1。解锁顺序不能被中断，否则可

能无法解锁。

写保护的寄存器位解锁后，用户需通过设置FMC_ISPCTL寄存器来决定更新LDROM、用户配置或APROM，然后使能ISP功能。

一旦FMC_ISPCTL寄存器设置正确，用户可以设置FMC_ISPCMD寄存器完成擦除、读取或编辑操作。通过ISPADDR作为Flash内存的目标地址。FMC_ISPDAT可以作为设定数据来编程或作为读寄存器命令FMC_ISPCMD的返回数据

最终，设定ISPGO (FMC_ISPTRG) 寄存器来执行相应的ISP功能。当ISP功能完成以后，ISPGO(FMC_ISPTRG) 位自动清除。为确保ISP功能在CPU继续运行之前已经完成，在ISPGO(FMC_ISPTRG)设置完成后应立即使用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失败标志会被置位。ISPFF标志只能由软件清除。就算ISPFF保持1，也可以开启下一个ISP流程，所以建议在ISP操作完成后检查ISPFF位，如果已经置1，则将其清0。

当ISPGO置位，CPU将一直等到ISP操作完成。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ISPBUSY位将被硬件自动清0。用户可以通过ISPBUSY位来检查ISP操作是否完成。用户应该在ISPGO置1之后加ISP指令，确保ISP操作之后的指令正确执行。

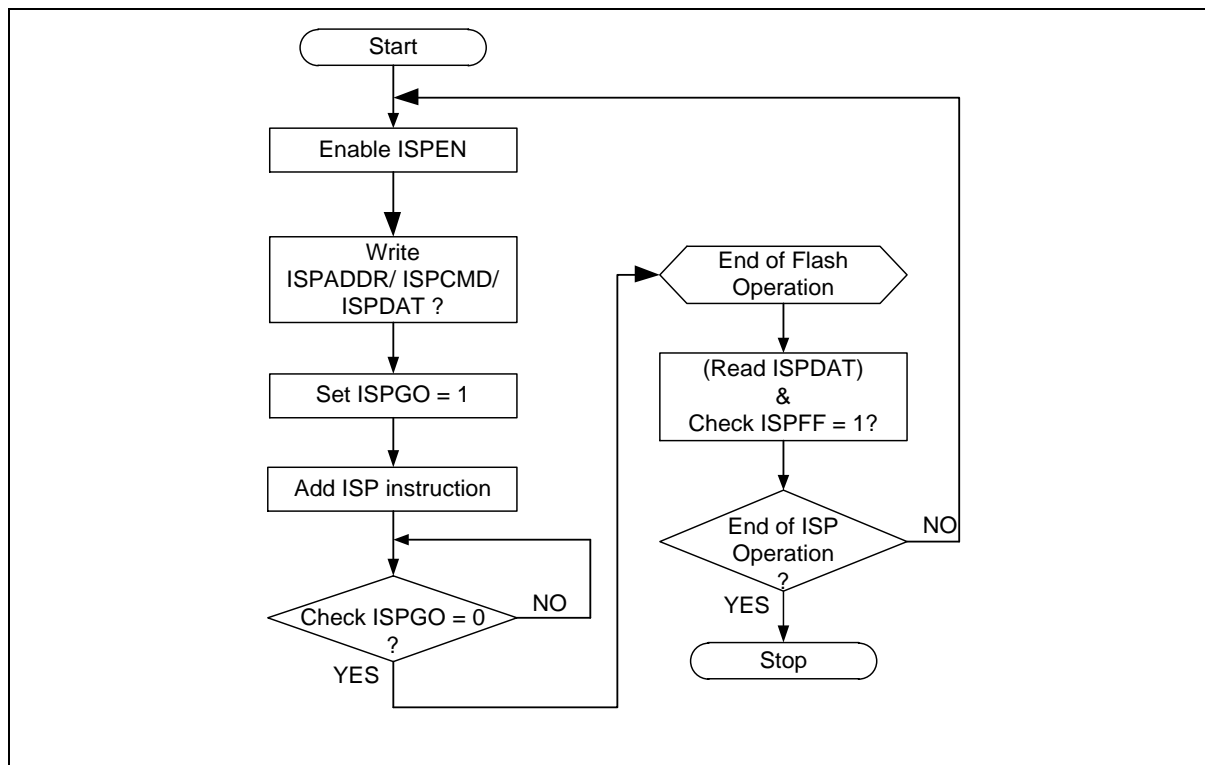


图 6.4-9 ISP 流程例图

表 6.4-5 NM1120支持的 ISP 命令列表

ISP 命令	FMC_ISPCMD	FMC_ISPADDR	FMC_ISPDAT
FLASH 页擦除	0x22	FLASH存储器的有效地址 它必须由512字节地址对齐	N/A
SPROM 页擦除	0x22	FLASH存储器的有效地址 它必须由512字节地址对齐	0x0055AA03
FLASH 编程	0x21	FLASH存储器的有效地址	编程数据
FLASH 读	0x00	FLASH存储器的有效地址	返回数据
读UID	0x04	0x0000_0000	Unique ID 字 0
		0x0000_0004	Unique ID 字 1
		0x0000_0008	Unique ID 字 2
读公司 ID	0x0B	N/A	公司 ID (0xDA)
向量重映射	0x2E	APROM或 LDROM中的有效地址,必须为 512字节对齐	N/A
CRC 计算	0x2D	CRC计算的起始地址	CRC 计算范围
读 CRC	0xD	0x0000_0000	CRC 值

表 6.4-5 ISP 命令表

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	读/写	描述	复位值
FMC 基地址: FMC_BA = 0x5000_C000				
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP 控制寄存器	0x0000_0000
FMC_ISPADDR	FMC_BA+0x04	R/W	ISP 地址寄存器	0x0000_0000
FMC_ISPDAT	FMC_BA+0x08	R/W	ISP 数据寄存器	0x0000_0000
FMC_ISPCMD	FMC_BA+0x0C	R/W	ISP 命令寄存器	0x0000_0000
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP 触发控制寄存器	0x0000_0000
FMC_DFBA	FMC_BA+0x14	R	数据 Flash 基地址	0x0000_3800
FMC_ISPSTS	FMC_BA+0x40	R/W	ISP 状态寄存器	0xFFFF_FFFF
FMC_CRCSEED	FMC_BA+0x50	R/W	ISP CRC 寄存器	0xFFFF_FFFF
FMC_CRCCV	FMC_BA+0x54	R	ISP CRC 当前值寄存器	0xFFFF_FFFF

6.4.6 寄存器描述

ISP 控制寄存器(FMC ISPCTL)

寄存器	偏移地址	读/写	描述	复位值
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP 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	ISPFF	LDUEN	CFGUEN	APUEN	SPUEN	BS	ISPEN

位	描述	
[31:7]	Reserved	保留.
[6]	ISPFF	ISP 失败标志 (写保护位) 当已启动的ISP符合下面的任何一个条件时, 该位由硬件置1: (1) APUEN等于0时, APROM 写APROM. (2) LDUEN等于0时, LDROM 写LDROM. (3) SPUEN等于0时, SPROM写SPROM. (4) CFGUEN等于0时, CONFIG 被擦除或编程. (5)定义地址无效, 比如超出正常范围 该位需要通过写1清0。
[5]	LDUEN	LDROM 更新使能位(写保护位) 0 = 禁止LDROM更新 1 = 当MCU运行在APROM时, LDROM 可以更新
[4]	CFGUEN	配置更新 使能 位(写保护位) 不管代码运行在APROM或LDROM, 该位写1 使能软件更新CONFIG值通过ISP寄存器。 0 = 禁止更新配置 1 = 使能更新配置
[3]	APUEN	APROM 更新使能 位(写保护位) 0 = 当芯片在APROM中运行时APROM 不能被更新. 1 = 当芯片在APROM中运行时APROM 可以被更新.

位	描述	
[2]	SPUEN	SPROM 更新使能位 (写保护位) 0 = SPROM 不能被更新. 1 = 当MCU运行在APROM, SPROM 能被更新.
[1]	BS	启动选择 (写保护位) 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.除了CPU 复位(CPURF = 1)及system 复位 (SYSRF)外的任何复位发生时, 该位由CBS (CONFIG0) 的反转值初始化。 0 = APROM启动 1 = LDROM启动
[0]	ISPEN	ISP 使能 位(写保护位) 设置该位可以使能ISP功能. 0 = 禁用 ISP 功能 1 = 使能 ISP 功能

ISP 地址寄存器 (FMC_ISPADDR)

寄存器	偏移地址	读/写	描述	复位值
FMC_ISPADDR	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 NM1120仅支持32位编程，在ISP操作中ISPADR[1:0]必须为00

FMC_ISPDAT (ISP 数据寄存器)

寄存器	偏移地址	读/写	描述	复位值
FMC_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 ISP写操作前，写数据到该寄存器 ISP读操作后，从该寄存器读数据

ISP 命令寄存器(FMC_ISPCMD)

寄存器	偏移地址	读/写	描述	复位值
FMC_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 ISP 命令表如下: 0x00 = 读Flash 0x04 = 读 UID. 0x0B = 读公司 ID (0xDA). 0x0D = 读 CRC32 Checksum 值. 0x21 = 编程. 0x22 = 页擦除. 0x2D = 执行 CRC32 Checksum 计算. 0x2E = 向量重映射

ISP 触发控制寄存器 (FMC_ISPTRG)

寄存器	偏移地址	读/写	描述	复位值
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP 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
Reserved							
7	6	5	4	3	2	1	0
Reserved							ISPGO

位	描述	
[31:1]	Reserved	保留.
[0]	ISPGO	ISP 触发寄存器 (写保护) 写 1 开始ISP操作, 当ISP操作结束后, 该位由硬件自动清零. 0 = ISP 操作结束 1 = ISP 正在执行

数据Flash起始地址寄存器 (FMC_DFBA)

寄存器	偏移地址	读/写	描述	复位值
FMC_DFBA	FMC_BA+0x14	R	数据Flash起始地址寄存器	0x0000_3800

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

位	描述	
[31:0]	DFBA	<p>数据Flash起始地址</p> <p>该寄存器为数据FLASH开始地址寄存器，只读。</p> <p>数据Flash起始地址由用户定义。因为片上FLASH擦除单位为512字节，所以强制DFBA的第8-0位置为0。</p>

例如:

Data Flash	4KB (DFEN=0)	2KB (DFEN=0)	1KB (DFEN=0)	0KB (DFEN=1)
17.5K Flash	DFBA=0x0000_3600	DFBA=0x0000_3E00	DFBA=0x0000_4200	DFEN=1
8K Flash	DFBA=0x0000_1000	DFBA=0x0000_1800	DFBA=0x0000_1C00	DFEN=1
4K Flash	Forbidden	DFBA=0x0000_0800	DFBA=0x0000_0C00	DFEN=1

ISP 状态寄存器 (FMC_ISPSTS)

寄存器	偏移地址	读/写	描述	复位值
FMC_ISPSTS	FMC_BA+0x40	R/W	ISP 状态寄存器	0xFFFF_FFFF

31	30	29	28	27	26	25	24
SCODE			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		ISPBUSY

位	描述	
[31:29]	SCODE	<p>安全代码有效标志位</p> <p>在flash初始化时，如果检测到SPROM安全代码有效，则该位由硬件置1。或由软件写1以确 保安全代码有效,该位仅可由SPROM页擦除操作清零。</p> <p>000 = SPROM0/1/2 安全代码无效 001 = SPROM0 安全代码有效 010 = SPROM1 安全代码有效 100 = SPROM2 安全代码有效 111 = SPROM0/1/2 安全代码有效</p>
[28:21]	Reserved	保留。
[20:9]	VECMAP	<p>向量页映射地址（只读）</p> <p>当前Flash地址空间0x0000_0000~0x0000_01FF映射到地址{VECMAP[11:0], 9'h000} ~ {VECMAP[11:0], 9'h1FF}</p>
[8:7]	Reserved	保留。
[6]	ISPFF	<p>ISP 失败标志（写保护位）</p> <p>当已启动的ISP符合下面的任何一个条件时，该位由硬件置1：</p> <p>(1) APUEN等于0时，APROM 写APROM. (2) LDUEN等于0时，LDROM 写LDROM. (3) SPUEN等于0时，SPROM写SPROM. (4) CFGUEN等于0时，CONFIG 被擦除或编程。 (5)定义地址无效，比如超出正常范围 该位需要通过写1清0。</p>
[5:3]	Reserved	保留。

[2:1]	CBS	<p>启动配置（只读）</p> <p>该位是CONFIG0（CBS）的镜像</p>
[0]	ISPBUSY	<p>ISP 忙标志位（只读）</p> <p>该位为1 表示ISP正在执行，当ISP操作结束后该位由硬件自动清0.</p> <p>0 = ISP 操作结束</p> <p>1 = ISP 正在执行</p> <p>注意：该位与FMC_ISPTRG[0]位相同</p>

ISP CRC 种子寄存器 (FMC_CRCSEED)

寄存器	偏移地址	读/写	描述	复位值
FMC_CRCSEED	FMC_BA+0x50	R/W	ISP CRC 种子寄存器	0xFFFF_FFFF

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

位	描述	
[31:0]	CRCSEED	CRC 种子数据 该寄存器提供CRC操作的初始值。 在ISP CRC操作之前写数据到该寄存器。 在ISP CRC读操作后从该寄存器读数据。

ISP CRC 当前值寄存器 (FMC_CRCCV)

寄存器	偏移地址	读/写	描述	复位值
FMC_CRCCV	FMC_BA+0x54	R	ISP CRC 当前值寄存器	0xFFFF_FFFF

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

位	描述	
[31:0]	CRCCV	CRC 当前值 该寄存器提供CRC二次计算的当前值。

6.5 通用 I/O (GPIO)

6.5.1 概述

NM1120系列有多达22个通用I/O管脚，这些管脚可以通过配置芯片和其它功能管脚共享。这22个管脚分配在PA, PB, PC和PD四个端口上。每个管脚都是独立的，都有相应的寄存器位来控制管脚的功能模式与数据。

每个I/O管脚的状态可由软件配置为输入，推挽输出，开漏输出或准双向模式。芯片复位之后，所有管脚状态根据CIOIN (CONFIG0[10])设置。每一个I/O管脚都有一个阻值为110K~300K的弱上拉电阻接到VDD上，VDD范围为从5.0 V 到2.5 V。

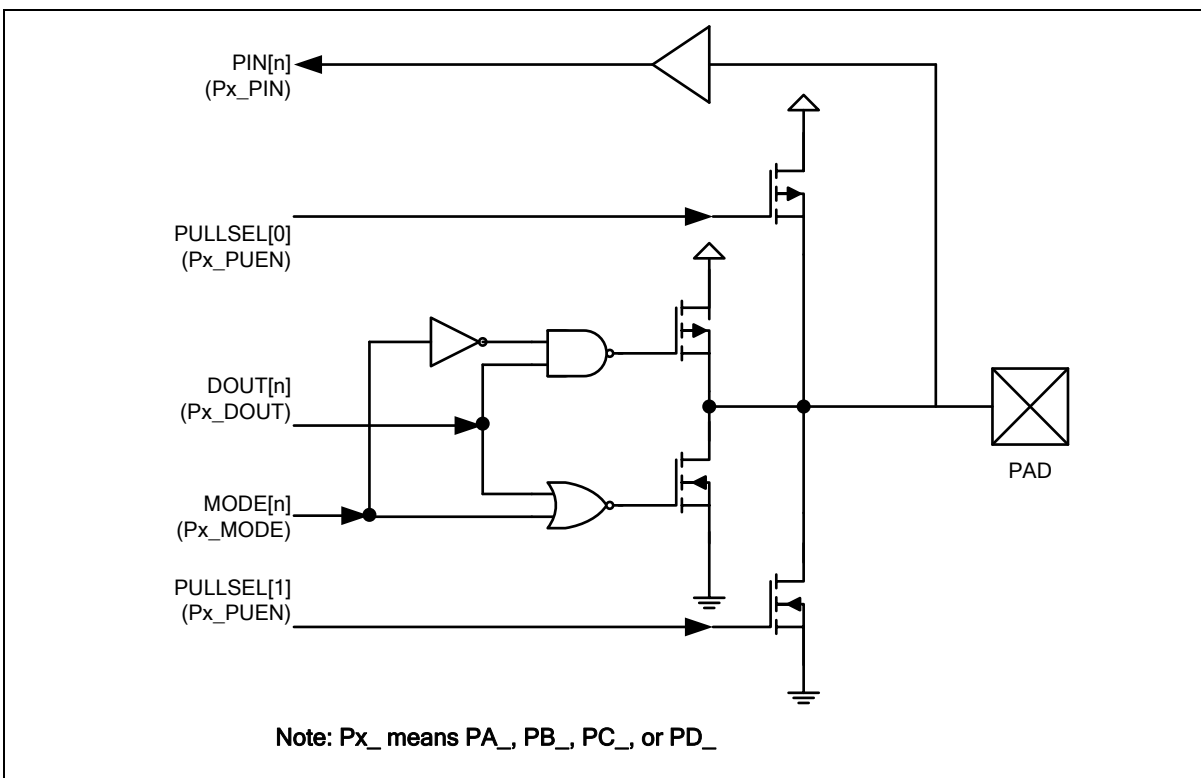


图 6.5-1 I/O 引脚方框图

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
- 可选择 TTL/Schmitt 触发输入模式

- I/O可以配置成边沿/电平触发的中断源
- 支持高驱动及高灌电流的I/O模式
- 支持软件可选择的斜率控制
- 通过CIOINI (CONFIG0[10])复位, 可配置所有I/O的默认模式
 - ◆ 如果CIOIN = 0, 复位后所有的GPIO管脚是准双向模式
 - ◆ 如果CIOIN = 1, 复位后所有的GPIO管脚是三态输入模式
- GPIOA 在四种I/O模式中支持使能上拉和下拉电阻
- GPIOB、GPIOC和GPIOD仅在准双向模式下, I/O管脚内部上拉电阻才使能
- 使能管脚中断功能将同时使能GPIO唤醒功能

6.5.3 框图

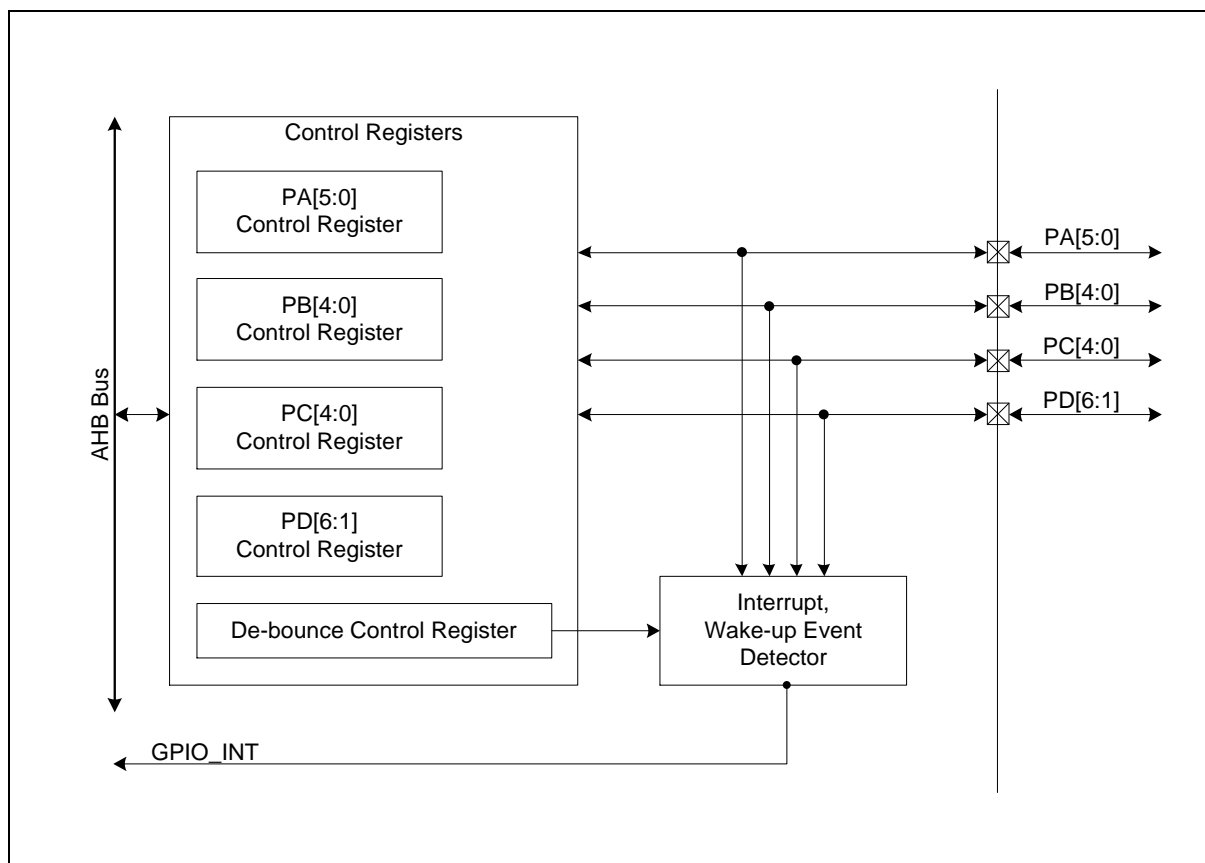


图 6.5-2 GPIO 控制器框图

6.5.4 基本配置

GPIO 管脚功能通过SYS_GPA_MFP, SYS_GPB_MFP, SYS_GPC_MFP, 和 SYS_GPD_MFP 寄存器进行配置.

6.5.5 功能描述

6.5.5.1 输入模式

设置MODEn (Px_MODE[2n+1:2n]) 为00, Px.n管脚为输入模式, I/O管脚为三态 (高阻), 没有输出驱动能力。管脚(Px_PIN[n])的值反映相应端口的状态。

6.5.5.2 推挽输出模式

设置 MODEn (Px_MODE[2n+1:2n]) 为01, Px.n 管脚为推挽输出模式, I/O支持数字输出功能, 有拉/灌电流能力。DOUT (Px_DOUT[n])相应位bit[n]的值被送到相应管脚上。

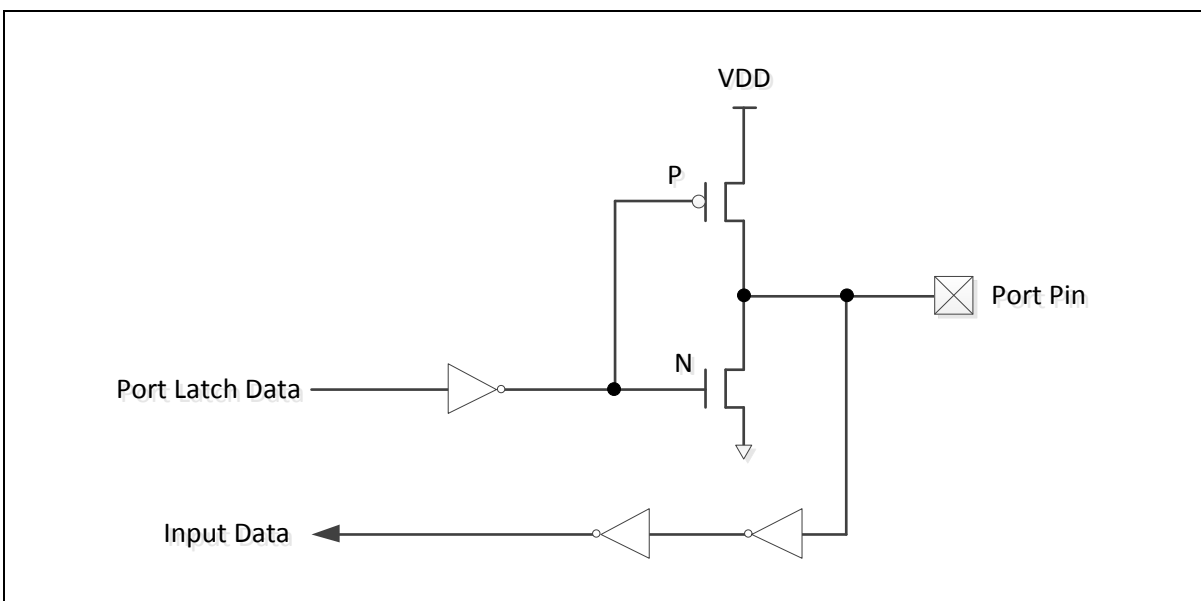


图 6.5-3 推挽输出

6.5.5.3 开漏输出模式

设置MODEn (Px_MODE[2n+1:2n]) 为10, Px.n 管脚为开漏模式, I/O管脚数字输出功能仅支持灌电流, 驱动高电平时需要外加上拉电阻。如果DOUT (Px_DOUT[n]) 相应位为 '0', 管脚上输出低电平. 如果DOUT(Px_DOUT[n]) 相应位为 '1', 该管脚输出为高电平时, 由外部上拉电阻控制。

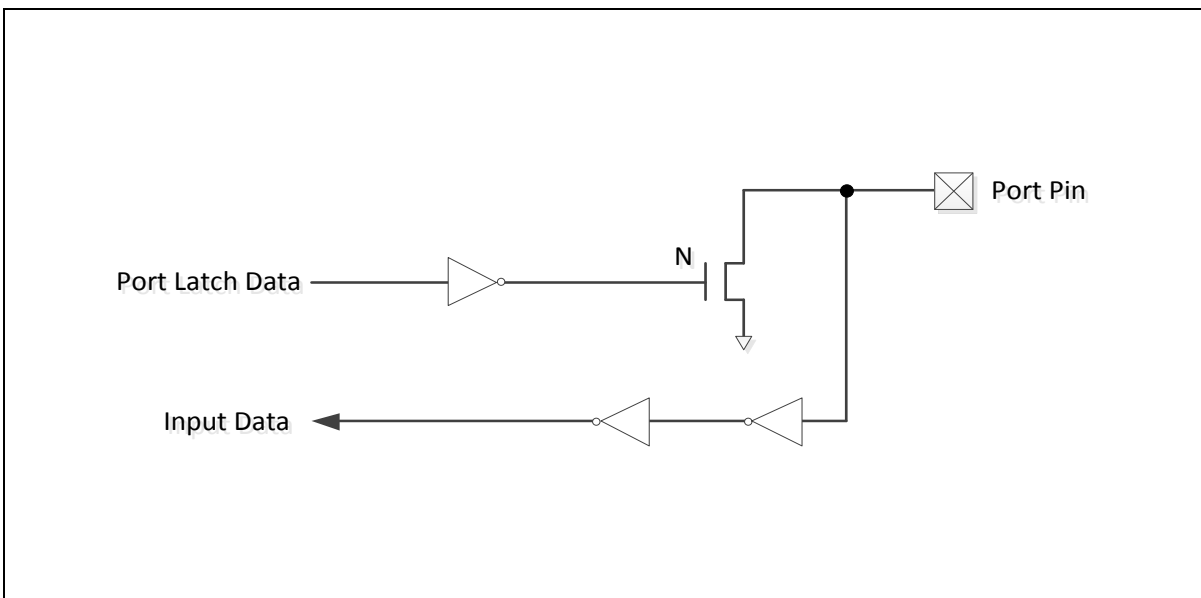


图 6.5-4 开漏输出

6.5.5.4 准双向模式

设置MODEn (Px_MODE[2n+1:2n])为11, Px.n 管脚为准双向模式, 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DOUT (Px_DOUT[n])相应位置1。准双向输出是80C51 及其派生产品常见的模式。若DOUT (Px_DOUT[n])相应位bit[n]为 ‘0’, 管脚上输出为“低”。若DOUT (Px_DOUT[n])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, 在该管脚上将有强输出驱动长度为2个时钟周期的高电平, 然后关闭强输出驱动, 其后管脚状态由内部上拉电阻控制。注意: 准双向模式的源电流大小仅有200uA到30 uA(相应VDD的电压从5.0 V到2.5 V)。

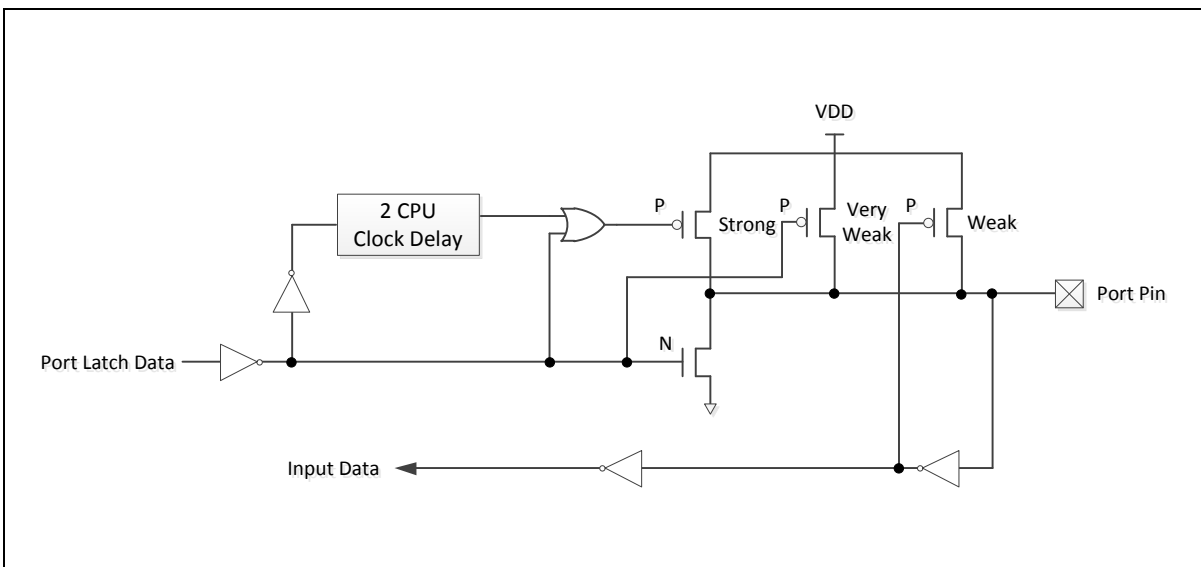


图 6.5-5 准双向 I/O 模式

6.5.6 GPIO 中断和唤醒功能

每个 GPIO 管脚都可以通过 RHIE (Px_INTEN[n+16])/FLIE (Px_INTEN[n]) 位和 TYPE (Px_INTTYPE[n]) 设置成芯片的中断源。有五种中断触发条件可以设置：低电平触发、高电平触发、下降沿触发和上升沿触发以及上升与下降沿同时触发。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 DBCLKSRC (GPIO_DBCTL[4]) 和 DBCLKSEL (GPIO_DBCTL[3:0]) 寄存器来设置。

当芯片进入 Idle/Power-down 模式时，GPIO 也可以唤醒系统。设置 GPIO 为唤醒触发的条件与 GPIO 中断触发的条件相同。

1. 进入 Idle/Power-down 模式之前确认 I/O 口的状态

当使用切换 GPIO 唤醒系统，根据相应的唤醒设置，用户进入空闲/掉电模式前必须确保 I/O 状态。

例如，如果配置唤醒事件发生由 I/O 上升沿/高电平触发，用户必须确保指定的引脚 I/O 状态在进入空闲/掉电模式前是低电平；如果配置 I/O 触发唤醒事件是下降沿/低电平触发，用户必须确保进入掉电模式前指定引脚 I/O 状态是高电平。

2. 在进入空闲/关机模式之前禁用 I/O 去抖动功能

如果通过 I/O 引脚唤醒并使能该引脚输入信号的去抖功能，系统唤醒是通过该 GPIO 引脚，系统会产生两个 GPIO 中断事件。一个中断事件是由唤醒功能引起的，另一个是由 I/O 输入去抖动功能引起的。用户应在进入空闲/关机模式之前禁用去抖功能，以避免系统唤醒后发生第二次中断事件。

6.5.7 寄存器映射

R: 只读, W: 只写, R/W: 读写

寄存器	偏移地址	读/写	描述	复位值
GPIO 基地址: GPIO_BA = 0x5000_4000				
PA_MODE	GPIO_BA+0x000	R/W	PA I/O 模式控制	0x0000_0XXX
PA_DINOFF	GPIO_BA+0x004	R/W	PA 关闭数字输入通路	0x0000_0000
PA_DOUT	GPIO_BA+0x008	R/W	PA 数据输出	0x0000_003F
PA_DATMSK	GPIO_BA+0x00C	R/W	PA 数据输出写屏蔽	0x0000_0000
PA_PIN	GPIO_BA+0x010	R	PA 管脚数值	0x0000_00XX
PA_DBEN	GPIO_BA+0x014	R/W	PA 去抖动使能控制	0x0000_0000
PA_INTTYPE	GPIO_BA+0x018	R/W	PA 中断模式控制	0x0000_0000
PA_INTEN	GPIO_BA+0x01C	R/W	PA 中断使能控制	0x0000_0000
PA_INTSRC	GPIO_BA+0x020	R/W	PA 中断源标志	0x0000_00XX
PA_SMTEN	GPIO_BA+0x024	R/W	PA Schmitt 输入触发使能寄存器	0x0000_0000
PA_SLEWCTL	GPIO_BA+0x028	R/W	PA 高斜率控制寄存器	0x0000_0000
PA_PLEN	GPIO_BA+0x02C	R/W	PA 下拉控制寄存器	0x0000_0000
PA_PHEN	GPIO_BA+0x030	R/W	PA 上拉控制寄存器	0x0000_003F
PB_MODE	GPIO_BA+0x040	R/W	PB I/O 模式控制	0x0000_0XXX
PB_DINOFF	GPIO_BA+0x044	R/W	PB 关闭数字输入通道	0x0000_0000
PB_DOUT	GPIO_BA+0x048	R/W	PB 数据输出	0x0000_001F
PB_DATMSK	GPIO_BA+0x04C	R/W	PB 数据输出写屏蔽	0x0000_0000
PB_PIN	GPIO_BA+0x050	R	PB 管脚数值	0x0000_00XX
PB_DBEN	GPIO_BA+0x054	R/W	PB 去抖动使能控制寄存器	0x0000_0000
PB_INTTYPE	GPIO_BA+0x058	R/W	PB 中断模式控制	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	PB 中断使能控制寄存器	0x0000_0000
PB_INTSRC	GPIO_BA+0x060	R/W	PB 中断源标志	0x0000_00XX
PB_SMTEN	GPIO_BA+0x064	R/W	PB Schmitt 输入触发使能寄存器	0x0000_0000
PB_SLEWCTL	GPIO_BA+0x068	R/W	PB 高斜率控制寄存器	0x0000_0000
PB_PLEN	GPIO_BA+0x06C	R/W	PB 下拉控制寄存器	0x0000_0000

PB_PHEN	GPIO_BA+0x070	R/W	PB 上拉控制寄存器	0x0000_001F
PC_MODE	GPIO_BA+0x080	R/W	PC I/O 模式控制	0x0000_0XXX
PC_DINOFF	GPIO_BA+0x084	R/W	PC 关闭数字输入通道	0x0000_0000
PC_DOUT	GPIO_BA+0x088	R/W	PC 数据输出	0x0000_001F
PC_DATMSK	GPIO_BA+0x08C	R/W	PC 数据输出写屏蔽	0x0000_0000
PC_PIN	GPIO_BA+0x090	R	PC 管脚数值	0x0000_00XX
PC_DBEN	GPIO_BA+0x094	R/W	PC 去抖动使能控制寄存器	0x0000_0000
PC_INTTYPE	GPIO_BA+0x098	R/W	PC 中断模式控制	0x0000_0000
PC_INTEN	GPIO_BA+0x09C	R/W	PC 中断使能控制寄存器	0x0000_0000
PC_INTSRC	GPIO_BA+0x0A0	R/W	PC 中断源标志	0x0000_00XX
PC_SMTEN	GPIO_BA+0x0A4	R/W	PC Schmitt 输入触发使能寄存器	0x0000_0000
PC_SLEWCTL	GPIO_BA+0x0A8	R/W	PC 高斜率控制寄存器	0x0000_0000
PC_PLEN	GPIO_BA+0x0AC	R/W	PC 下拉控制寄存器	0x0000_0000
PC_PHEN	GPIO_BA+0x0B0	R/W	PC 上拉控制寄存器	0x0000_001F
PD_MODE	GPIO_BA+0x0C0	R/W	PD I/O 模式控制	0x0000_00XX
PD_DINOFF	GPIO_BA+0x0C4	R/W	PD 关闭数字输入通道	0x0000_0000
PD_DOUT	GPIO_BA+0x0C8	R/W	PD 数据输出	0x0000_007F
PD_DATMSK	GPIO_BA+0x0CC	R/W	PD 数据输出写屏蔽	0x0000_0000
PD_PIN	GPIO_BA+0x0D0	R	PD 管脚数值	0x0000_00XX
PD_DBEN	GPIO_BA+0x0D4	R/W	PD 去抖动使能控制寄存器	0x0000_0000
PD_INTTYPE	GPIO_BA+0x0D8	R/W	PD 中断模式控制	0x0000_0000
PD_INTEN	GPIO_BA+0x0DC	R/W	PD 中断使能控制寄存器	0x0000_0000
PD_INTSRC	GPIO_BA+0x0E0	R/W	PD 中断源标志	0x0000_00XX
PD_SMTEN	GPIO_BA+0x0E4	R/W	PD Schmitt 输入触发使能寄存器	0x0000_0000
PD_SLEWCTL	GPIO_BA+0x0E8	R/W	PD 高斜率控制寄存器	0x0000_0000
PD_PLEN	GPIO_BA+0x0EC	R/W	PD 下拉控制寄存器	0x0000_0000
PD_PHEN	GPIO_BA+0x0F0	R/W	PD 上拉控制寄存器	0x0000_007F
GPIO_DBCTL	GPIO_BA+0x440	R/W	去抖周期控制寄存器	0x0000_0020
PAn_PDIO n=0,1..5	GPIO_BA+0x800+(0x04 * n)	R/W	GPIO PA.n 端口数据输入/输出	0x0000_000X

PBn_PDIO n=0,1..4	GPIO_BA+0x840+(0x04 * n)	R/W	GPIO PB.n 端口数据输入/输出	0x0000_000X
PCn_PDIO n=0,1..4	GPIO_BA+0x880+(0x04 * n)	R/W	GPIO PC.n 端口数据输入/输出	0x0000_000X
PDn_PDIO n=0,1..6	GPIO_BA+0x8C0+(0x04 * n)	R/W	GPIO PD.n 端口数据输入/输出	0x0000_000X

6.5.8 寄存器描述

端口 A-D I/O 模式控制 (Px MODE)

寄存器	偏移地址	读/写	描述	复位值
PA_MODE	GPIO_BA+0x000	R/W	PA I/O 模式控制	0x0000_0XXX
PB_MODE	GPIO_BA+0x040	R/W	PB I/O 模式控制	0x0000_0XXX
PC_MODE	GPIO_BA+0x080	R/W	PC I/O 模式控制	0x0000_0XXX
PD_MODE	GPIO_BA+0x0C0	R/W	PD I/O 模式控制	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
MODE7		MODE6		MODE5		MODE4	
7	6	5	4	3	2	1	0
MODE3		MODE2		MODE1		MODE0	

位	描述	
[31:16]	Reserved	保留.
[2n+1:2n] n=0,1..7	MODEn	<p>端口 A-d I/O Pin[n] 模式控制 决定Px.n引脚的每个I/O 类型. 00 = Px.n管脚为输入模式 01 = Px.n管脚为推挽输出模式 10 = Px.n 管脚为开漏输出模式 11 = Px.n管脚为准双向模式 注意1: 初始值由CIOINI (CONFIG0 [10])决定, CIOINI置0时默认值是0xFFFF_FFFF, 上电后所有端口是准双向模式。 CIOINI置1时默认值是0x0000_0000, 上电后所有端口是输入模式。 注意2: 端口A: Max. n=5 端口B: Max. n=4 端口C: Max. n=4 端口D: Max. n=6, n=0保留</p>

Port A-D 管脚关闭数字通路寄存器 (Px_DINOFF)

寄存器	偏移地址	读/写	描述	复位值
PA_DINOFF	GPIO_BA+0x004	R/W	PA 关闭数字通路使能	0x0000_0000
PB_DINOFF	GPIO_BA+0x044	R/W	PB 关闭数字通路使能	0x0000_0000
PC_DINOFF	GPIO_BA+0x084	R/W	PC 关闭数字通路使能	0x0000_0000
PD_DINOFF	GPIO_BA+0x0C4	R/W	PD 关闭数字通路使能	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
DINOFFn							
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.
[n+16] n=0,1..7	DINOFFn	<p>端口A-d Pin[n]关闭数字输入通道使能</p> <p>用于控制Px.n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Px.n数字输入通道,防止漏电.</p> <p>0 = 使能IO 数字输入通道</p> <p>1 = 关闭IO 数字输入通道(数字输入拉低)</p> <p>注意:</p> <p>端口A: Max. n=5</p> <p>端口B: Max. n=4</p> <p>端口C: Max. n=4</p> <p>端口D: Max. n=6,n=0保留</p>
[15:0]	Reserved	保留.

端口 A-D 数据输出值 (Px DOUT)

寄存器	偏移地址	读/写	描述	复位值
PA_DOUT	GPIO_BA+0x008	R/W	PA 数据输出值	0x0000_003F
PB_DOUT	GPIO_BA+0x048	R/W	PB 数据输出值	0x0000_001F
PC_DOUT	GPIO_BA+0x088	R/W	PC 数据输出值	0x0000_001F
PD_DOUT	GPIO_BA+0x0C8	R/W	PD 数据输出值	0x0000_007F

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
DOUTn							

位	描述	
[31:8]	Reserved	保留.
[n] n=0,1..7	DOUTn	<p>端口 A-d Pin[n] 输出值</p> <p>在 Px.n 配置成推挽输出, 开漏输出或准双向模式时, 控制 Px.n 相应管脚的状态.</p> <p>0 = GPIO 配置成推挽输出, 开漏或准双向模式时, Px.n 为低</p> <p>1 = GPIO 配置成推挽输出, 开漏或准双向模式时, Px.n 为高</p> <p>注意 :</p> <p>端口 A: Max. n=5</p> <p>端口 B: Max. n=4</p> <p>端口 C: Max. n=4</p> <p>端口 D: Max. n=6,n=0 保留</p>

端口 A-D 数据输出写屏蔽 (Px_DATMSK)

寄存器	偏移地址	读/写	描述	复位值
PA_DATMSK	GPIO_BA+0x00C	R/W	PA 数据输出写屏蔽	0x0000_0000
PB_DATMSK	GPIO_BA+0x04C	R/W	PB 数据输出写屏蔽	0x0000_0000
PC_DATMSK	GPIO_BA+0x08C	R/W	PC 数据输出写屏蔽	0x0000_0000
PD_DATMSK	GPIO_BA+0x0CC	R/W	PD 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
DATMSKn							

位	描述	
[31:8]	Reserved	保留.
[n] n=0,1..7	DATMSKn	<p>端口A-d Pin[n] 数据输出写屏蔽</p> <p>用于保护相应寄存器DOUT (Px_DOUT[n]). 当设置DATMSK (Px_DATMSK[n])为 '1'时, 相应DOUT (Px_DOUT[n])位被保护, 写信号被屏蔽时, 不能向保护位写数据。</p> <p>0 =相应的DOUT (Px_DOUT[n])位可以被更新</p> <p>1 =保护相应的DOUT (Px_DOUT[n])位</p> <p>注意1: 该功能只保护相应的DOUT (Px_DOUT[n])位, 不保护(PDIO (Pxn_PDIO[0]))相应位。</p> <p>注意2 :</p> <p>端口A: Max. n=5</p> <p>端口B: Max. n=4</p> <p>端口C: Max. n=4</p> <p>端口D: Max. n=6,n=0保留</p>

端口 A-D 管脚数据 (Px PIN)

寄存器	偏移地址	读/写	描述	复位值
PA_PIN	GPIO_BA+0x010	R	PA 管脚数据	0x0000_00XX
PB_PIN	GPIO_BA+0x050	R	PB 管脚数据	0x0000_00XX
PC_PIN	GPIO_BA+0x090	R	PC 管脚数据	0x0000_00XX
PD_PIN	GPIO_BA+0x0D0	R	PD 管脚数据	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
PINn							

位	描述	
[31:8]	Reserved	保留.
[n] n=0,1..7	PINn	端口A-d Pin[n] 管脚数据 寄存器的每一位的值反映出各个Px.n管脚真实状态。如果值为 ‘1’，表示相应管脚状态为高，否则为低 注意： 端口A: Max. n=5 端口B: Max. n=4 端口C: Max. n=4 端口D: Max. n=6,n=0保留

端口 A-D 去抖动使能寄存器 (Px DBEN)

寄存器	偏移地址	读/写	描述	复位值
PA_DBEN	GPIO_BA+0x014	R/W	PA 去抖动使能控制	0x0000_0000
PB_DBEN	GPIO_BA+0x054	R/W	PB 去抖动使能控制	0x0000_0000
PC_DBEN	GPIO_BA+0x094	R/W	PC 去抖动使能控制	0x0000_0000
PD_DBEN	GPIO_BA+0x0D4	R/W	PD 去抖动使能控制	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
DBENn							

位	描述	
[31:8]	Reserved	保留.
[n] n=0,1..7	DBENn	<p>端口A-d Pin[n]] 输入信号去抖动使能位</p> <p>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, 则被视为信号抖动, 从而不触发中断. 去抖动时钟源由DBCLKSRC (GPIO_DBCTL [4])控制, 一个去抖动采样周期由DBCLKSEL (GPIO_DBCTL [3:0])控制。</p> <p>0 = 禁用Px.n去抖动功能</p> <p>1 = 使能Px.n去抖动功能</p> <p>去抖动功能仅对边沿触发中断有效, 对于电平触发中断模式, 去抖动功能使能位不起作用.</p> <p>注意:</p> <p>端口A: Max. n=5</p> <p>端口B: Max. n=4</p> <p>端口C: Max. n=4</p> <p>端口D: Max. n=6,n=0保留</p>

端口 A-D 中断模式控制 (Px_INTTYPE)

寄存器	偏移地址	读/写	描述	复位值
PA_INTTYPE	GPIO_BA+0x018	R/W	PA 中断模式控制	0x0000_0000
PB_INTTYPE	GPIO_BA+0x058	R/W	PB 中断模式控制	0x0000_0000
PC_INTTYPE	GPIO_BA+0x098	R/W	PC 中断模式控制	0x0000_0000
PD_INTTYPE	GPIO_BA+0x0D8	R/W	PD 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
TYPE _n							

位	描述	
[31:8]	Reserved	保留.
[n] n=0,1..7	TYPE _n	<p>端口A-d Pin[n]边沿或电平检测中断控制</p> <p>TYPE (Px_INTTYPE[n])位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。</p> <p>0 = 边沿触发中断 1 = 电平触发中断</p> <p>如果设置管脚为电平触发模式, 则在寄存器RHIE_n (Px_INTEN[n+16])/FLIE_n (Px_INTEN[n])中, 只能设置一个电平; 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。</p> <p>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。</p> <p>注意:</p> <p>端口A: Max. n=5 端口B: Max. n=4 端口C: Max. n=4 端口D: Max. n=6,n=0保留</p>

端口 A-D 中断使能控制寄存器 (Px_INTEN)

寄存器	偏移地址	读/写	描述	复位值
PA_INTEN	GPIO_BA+0x01C	R/W	PA 中断使能控制寄存器	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	PB 中断使能控制寄存器	0x0000_0000
PC_INTEN	GPIO_BA+0x09C	R/W	PC 中断使能控制寄存器	0x0000_0000
PD_INTEN	GPIO_BA+0x0DC	R/W	PD 中断使能控制寄存器	0x0000_0000

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

位	描述	
[31:24]	Reserved	保留.
[n+16] n=0,1..7	RHIENn	<p>端口A-d Pin[n]输入上升沿或输入高电平中断使能位</p> <p>RHIEN (Px_INTEN[n+16])用于使能相应Px.n输入引脚中断。置‘1’也可以使能管脚唤醒功能。</p> <p>当设置RHIEN (Px_INTEN[n+16])位为‘1’</p> <p>如果中断是电平触发模式(TYPE (Px_INTTYPE[n])=1), 输入Px.n的状态为高电平时, 产生中断。</p> <p>如果中断是边沿触发模式(TYPE (Px_INTTYPE[n])=0), 输入Px.n的状态由低电平到高电平变化时, 产生中断。</p> <p>0 = 禁用Px.n高电平或由低电平到高电平变化的中断</p> <p>1 = 使能Px.n高电平或由低电平到高电平变化的中断</p> <p>注意:</p> <p>端口A: Max. n=5</p> <p>端口B: Max. n=4</p> <p>端口C: Max. n=4</p> <p>端口D: Max. n=6,n=0保留</p>
[15:8]	Reserved	保留.
[n] n=0,1..7	FLIENn	<p>端口A-d Pin[n]输入下降沿或输入低电平中断使能位</p> <p>FLIEN (Px_INTEN[n])用于使能相应 Px.n引脚的输入中断。置‘1’也可以使能管脚唤醒功能。</p>

		<p>当设置FLIEN (Px_INTEN[n])位为 '1'。</p> <p>如果中断是电平触发模式(TYPE (Px_INTTYPE[n])=1),，输入Px.n的状态为低电平时，产生中断.</p> <p>如果中断是边沿触发模式(TYPE (Px_INTTYPE[n])= 0),，输入Px.n的状态由高电平到低电平变化时，产生中断.</p> <p>0 = 禁用Px.n低电平或由高电平到低电平变化的中断</p> <p>1 = 使能Px.n低电平或由高电平到低电平变化的中断</p> <p>注意:</p> <p>端口A: Max. n=5</p> <p>端口B: Max. n=4</p> <p>端口C: Max. n=4</p> <p>端口D: Max. n=6,n=0保留</p>
--	--	--

端口 A-D 中断源标志 (Px_INTSRC)

寄存器	偏移地址	读/写	描述	复位值
PA_INTSRC	GPIO_BA+0x020	R/W	PA 中断源标志	0x0000_00XX
PB_INTSRC	GPIO_BA+0x060	R/W	PB 中断源标志	0x0000_00XX
PC_INTSRC	GPIO_BA+0x0A0	R/W	PC 中断源标志	0x0000_00XX
PD_INTSRC	GPIO_BA+0x0E0	R/W	PD 中断源标志	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
INTSRCn							

位	描述	
[31:8]	Reserved	保留.
[n] n=0,1..7	INTSRCn	<p>端口A-d Pin[n]中断触发源标志</p> <p>写操作：</p> <p>0= 无动作</p> <p>1= 清相应的未处理中断标志</p> <p>读操作：</p> <p>0 = Px.n没有中断</p> <p>1 = Px.n产生中断</p> <p>注意：</p> <p>端口A: Max. n=5</p> <p>端口B: Max. n=4</p> <p>端口C: Max. n=4</p> <p>端口D: Max. n=6,n=0保留</p>

端口 A-D Schmitt 输入触发使能寄存器 (Px SMTEN)

寄存器	偏移地址	读/写	描述	复位值
PA_SMTEN	GPIO_BA+0x024	R/W	PA Schmitt 输入触发使能寄存器	0x0000_0000
PB_SMTEN	GPIO_BA+0x064	R/W	PB Schmitt 输入触发使能寄存器	0x0000_0000
PC_SMTEN	GPIO_BA+0x0A4	R/W	PC Schmitt 输入触发使能寄存器	0x0000_0000
PD_SMTEN	GPIO_BA+0x0E4	R/W	PD Schmitt 输入触发使能寄存器	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
SMTENn							

位	描述	
[31:8]	Reserved	保留.
[n] n=0,1..7	SMTENn	<p>端口A-d Pin[n] Schmitt 输入触发使能寄存器</p> <p>0 = Px.n 禁用Schmitt 输入触发使能寄存器</p> <p>1 = Px.n 使能Schmitt 输入触发使能寄存器</p> <p>注意:</p> <p>端口A: Max. n=5</p> <p>端口B: Max. n=4</p> <p>端口C: Max. n=4</p> <p>端口D: Max. n=6,n=0保留</p>

端口 A-D 高斜率控制寄存器 (Px SLEWCTL)

寄存器	偏移地址	读/写	描述	复位值
PA_SLEWCTL	GPIO_BA+0x028	R/W	PA 高斜率控制寄存器	0x0000_0000
PB_SLEWCTL	GPIO_BA+0x068	R/W	PB 高斜率控制寄存器	0x0000_0000
PC_SLEWCTL	GPIO_BA+0x0A8	R/W	PC 高斜率控制寄存器	0x0000_0000
PD_SLEWCTL	GPIO_BA+0x0E8	R/W	PD 高斜率控制寄存器	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
HSRENn							

位	描述	
[31:8]	Reserved	保留.
[n] n=0,1..7	HSRENn	端口A-d Pin[n] 高斜率控制 0 = Px.n 普通斜率输出 1 = Px.n 高斜率输出 注意: 端口A: Max. n=5 端口B: Max. n=4 端口C: Max. n=4 端口D: Max. n=6,n=0保留

端口下拉电阻控制寄存器 (Px_PLEN)

寄存器	偏移地址	读/写	描述	复位值
PA_PLEN	GPIO_BA+0x02C	R/W	PA 下拉控制寄存器	0x0000_0000
PB_PLEN	GPIO_BA+0x06C	R/W	PB 下拉控制寄存器	0x0000_0000
PC_PLEN	GPIO_BA+0x0AC	R/W	PC 下拉控制寄存器	0x0000_0000
PD_PLEN	GPIO_BA+0x0EC	R/W	PD 下拉控制寄存器	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
PLENn							

位	描述	
[31:8]	Reserved	保留.
[n] n=0,1..7	PLENn	<p>端口A-d Pin[n]下拉电阻控制</p> <p>0 = 禁用下拉电阻</p> <p>1 = 使能下拉电阻</p> <p>注意：通过GPA_n_RINI (CONFIG0[27:16])定义PA_PLEN的初值，芯片上电后被选的引脚会被配置</p> <p>注意：</p> <p>端口A: Max. n=5</p> <p>端口B: Max. n=4</p> <p>端口C: Max. n=4</p> <p>端口D: Max. n=6,n=0保留</p>

端口上拉电阻控制寄存器 (Px_PHEN)

寄存器	偏移地址	读/写	描述	复位值
PA_PHEN	GPIO_BA+0x030	R/W	PA 上拉控制寄存器	0x0000_003F
PB_PHEN	GPIO_BA+0x070	R/W	PB 上拉控制寄存器	0x0000_001F
PC_PHEN	GPIO_BA+0x0B0	R/W	PC 上拉控制寄存器	0x0000_001F
PD_PHEN	GPIO_BA+0x0F0	R/W	PD 上拉控制寄存器	0x0000_007F

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
PHENn							

位	描述	
[31:8]	Reserved	保留.
[n] n=0,1..7	PHENn	<p>端口A-d Pin[n]上拉电阻控制</p> <p>0 = 使能上拉电阻</p> <p>1 = 禁用上拉电阻</p> <p>注意: 通过GPA_n_RINI (CONFIG0[27:16])定义PA_ PHEN的初值, 芯片上电后被选的引脚会被配置</p> <p>注意:</p> <p>端口A: Max. n=5</p> <p>端口B: Max. n=4</p> <p>端口C: Max. n=4</p> <p>端口D: Max. n=6,n=0保留</p>

中断去抖周期控制寄存器 (GPIO_DBCTL)

寄存器	偏移地址	读/写	描述	复位值
GPIO_DBCTL	GPIO_BA+0x440	R/W	中断去抖周期控制寄存器	0x0000_0020

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		ICLKON	DBCLKSRC	DBCLKSEL			

位	描述	
[31:6]	Reserved	保留.
[5]	ICLKON	中断时钟 On 模式 0 = 边沿侦测电路仅在 I/O管脚对应的RHIEN (Px_INTEN[n+16])/FLIEN (Px_INTEN[n])位置1有效. 1 = 复位之后所有 I/O边沿侦测电路使能. 注意: 如果没有相关的特定应用, 建议关掉该时钟以减少耗电
[4]	DBCLKSRC	去抖动计数器时钟源选择 0 = 去抖动计数器时钟源为 HCLK 1 = 去抖动计数器时钟源为内部 10 KHz(LIRC)时钟

位	描述	
[3:0]	DBCLKSEL	<p>去抖动采样周期选择</p> <p>0000 = 采样中断信号, 每1 clocks一次</p> <p>0001 = 采样中断信号, 每2 clocks一次</p> <p>0010 = 采样中断信号, 每4 clocks一次</p> <p>0011 = 采样中断信号, 每8 clocks一次</p> <p>0100 = 采样中断信号, 每16 clocks一次</p> <p>0101 = 采样中断信号, 每32 clocks一次</p> <p>0110 = 采样中断信号, 每64 clocks一次</p> <p>0111 = 采样中断信号, 每128 clocks一次</p> <p>1000 = 采样中断信号, 每256 clocks一次</p> <p>1001 = 采样中断信号, 每2*256 clocks一次.</p> <p>1010 = 采样中断信号, 每4*256 clocks一次.</p> <p>1011 = 采样中断信号, 每8*256 clocks一次</p> <p>1100 = 采样中断信号, 每16*256 clocks一次</p> <p>1101 = 采样中断信号, 每32*256 clocks一次</p> <p>1110 = 采样中断信号, 每64*256 clocks一次</p> <p>1111 = 采样中断信号, 每128*256 clocks一次</p>

GPIO Px.n 管脚数据输入输出寄存器 (Pxn_PDIO)

寄存器	偏移地址	读/写	描述	复位值
PAn_PDIO n=0,1..5	GPIO_BA+0x800+(0x04 * n)	R/W	GPIO PA.n 管脚数据输入/输出寄存器	0x0000_000X
PBn_PDIO n=0,1..4	GPIO_BA+0x840+(0x04 * n)	R/W	GPIO PB.n 管脚数据输入/输出寄存器	0x0000_000X
PCn_PDIO n=0,1..4	GPIO_BA+0x880+(0x04 * n)	R/W	GPIO PC.n 管脚数据输入/输出寄存器	0x0000_000X
PDn_PDIO n=0,1..6	GPIO_BA+0x8C0+(0x04 * n)	R/W	GPIO PD.n 管脚数据输入/输出寄存器	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							PDIO

位	描述	
[31:1]	Reserved	保留.
[0]	PDIO	<p>GPIO Px.N 管脚数据输入/ 输出</p> <p>写该位可以控制一个GPIO管脚的输出值</p> <p>0 = 设置相应GPIO管脚为低</p> <p>1 = 设置相应GPIO管脚为高</p> <p>读该寄存器得到GPIO管脚状态</p> <p>例如: 写PA0_PDIO即把值写到DOUT (Px_DOUT[0])位上, 读PA0_PDIO即读取PIN (PA_PIN[0])的值.</p> <p>注意1: 写操作不受DATMSK (Px_DATMSK[n])影响</p> <p>注意2:</p> <p>端口A: Max. n=5</p> <p>端口B: Max. n=4</p> <p>端口C: Max. n=4</p> <p>端口D: Max. n=6,n=0保留</p>

6.6 定时器控制器(TIMER)

6.6.1 概述

定时器控制器包含 2 组 32-位定时器，TIMER0 和 TIMER1，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

- 支持2组 32-位定时器，包含一个24位向上计数器和一个8位的预分频计数器
- 每个定时器(TMR0_CLK, TMR1_CLK)都有独立的时钟源
- 提供 one-shot, periodic, toggle 和 continuous 四种计数操作模式
- $\text{Time-out 周期} = (\text{输入定时器时钟周期}) * (8\text{位预分频计数器} + 1) * (24\text{位 CMPDAT})$
- $\text{支持最大计数周期时间} = (1 / T \text{ MHz}) * (2^8) * (2^{24})$; T表示定时器输入时钟周期
- 通过TIMERx_CNT (定时器数据寄存器)可读取内部 24 位向上计数器的值
- 支持引脚 (TM0, TM1) 事件计数功能
- 支持内部捕捉触发当内部ACMP输出信号变化
- 当定时器中断信号产生，支持芯片从Idle/Power-down唤醒

6.6.3 框图

每个定时器都有一个8位预分频计数器，一个24位向上计数器，一个24位比较寄存器和一个中断响应信号。如图 6.6-1。每个定时器有5个时钟源可选。如图 6.6-2说明时钟源控制功能。

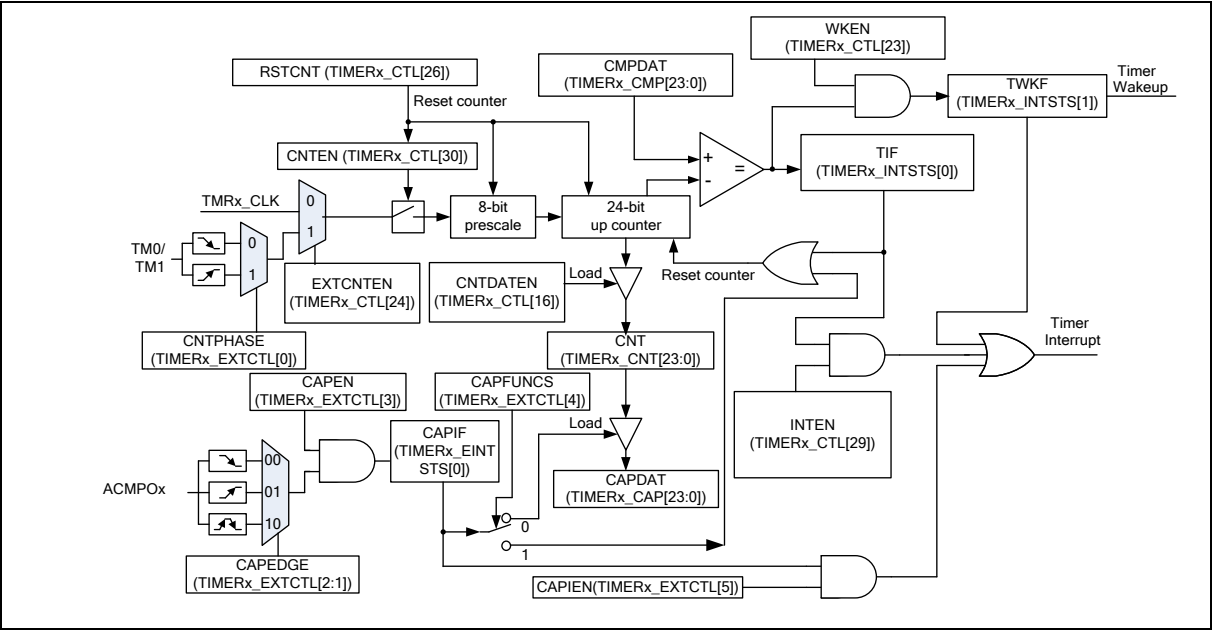


图 6.6-1 定时器控制器框图

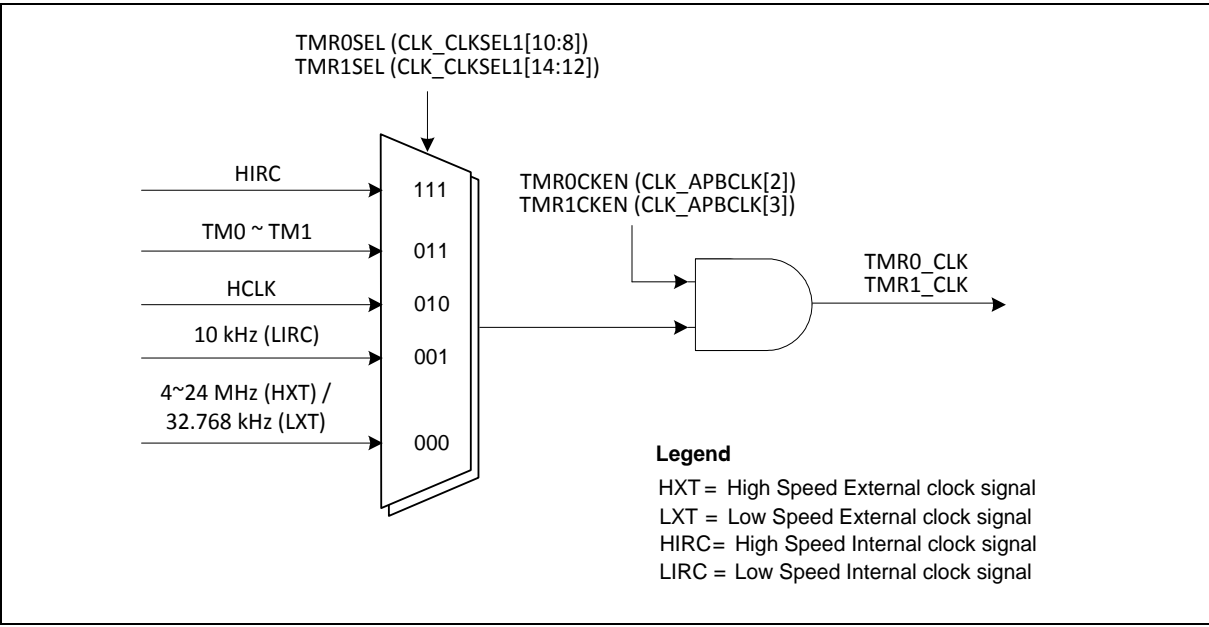


图 6.6-2 定时器控制器时钟源

6.6.4 基本配置

定时器Timer0和Timer1的时钟源可以在寄存器TMRxCKEN (CLK_APBCLK[3:2])使能和在寄存器TMR0SEL (CLK_CLKSEL1[10:8]) (Timer0), TMR1SEL (CLK_CLKSEL1[14:12]) (Timer1)选择不同时钟。

6.6.5 功能描述

定时器控制提供One-shot, Period, Toggle 和 Continuous Counting操作模式。事件计数功能也支持计数事件/计数通过外部引脚, 外部引脚捕获功能可以用来测量间隔或者重启定时器计数。每个操作功能模式如下:

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 (TIMERx_INTSTS[0])标志, 该标志在当定时器计数器值CNT (TIMERx_CNT[23:0])与定时器比较值CMPDAT (TIMERx_CMP[23:0])相匹配时置位, 另一个是CAPIF (TIMERx_EINTSTS[0])标志, 该标志在当ACMP0x(ACMP_STATUS[3] 和ACMP_STATUS[2])的变化与CAPEDGE (TIMERx_EXTCTL[2:1])的设置一致时置位。

6.6.5.2 定时器计数模式

定时器控制器提供四种定时器计数模式: one-shot, periodic, toggle-output 和 continuous counting 计数模式。

6.6.5.3 One-shot 模式

如果定时器工作在单周期 (one-shot) 模式OPMODE(TIMERx_CTL[28:27] 为00) 且 CNTEN (TIMERx_CTL[30])置1, 则定时器的计数器开始计数。一旦CNT (TIMERx_CNT[23:0])达到CMPDAT (TIMERx_CMP[23:0]) 的值时, TIF (TIMERx_INTSTS[0]) 标志将变为1, CNT(TIMERx_CNT[23:0])的值和 CNTEN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INTEN (TIMERx_CTL[29])位使能, 则定时器中断信号产生并送到 NVIC通知CPU。

6.6.5.4 Periodic 模式

如果定时器工作在周期 (periodic) 模式(OPMODE (TIMERx_CTL[28:27] 为01) 且 CNTEN (TIMERx_CTL[30])置1), 则定时器的计数器开始向上计数。一旦CNT (TIMERx_CNT[23:0])计数器的值达到CMPDAT (TIMERx_CMP[23:0])的值时, TIF标志将变为1, CNT的值将由定时器控制器自动清零, 然后定时器重新计数。与此同时, 如果INTEN (TIMERx_CTL[29])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CMPDAT的值比较, 直到CNTEN位由用户软件清0。

6.6.5.5 Toggle-Output 模式

如果定时器工作在触发输出 (toggle-out) 模式 (OPMODE (TIMERx_CTL[28:27] 为10)) , 且 CNTEN (TIMERx_CTL[30]) 位置1, 则定时器的计数器开始计数。toggle-out 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被置位时, 有相关的管脚会输出信号。因此, 管脚上的

触发输出信号以 50% 的占空周期反复改变。

6.6.5.6 Continuous Counting 模式

如果定时器工作在连续计数模式（OPMODE (TIMERx_CTL[28:27] 为 11)），且 CNTEN (TIMERx_CTL[30]) 位置 1，则定时器的计数器开始计数。一旦 CNT (TIMERx_CNT[23:0]) 的值达到 CMPDAT (TIMERx_CMP[23:0]) 的值时，TIF 标志将变为 1，但 CNT 的值继续保持向上计数。与此同时，如果 INTEN 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 CMPDAT 值，而不需要停止定时器计数和重新开始定时器计数。

例如，CMPDAT 的值设置为 80。当 CNT 达到 80 时，TIF 标志将被置 1，定时器计数器继续计数，而且 CNT 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 CMPDAT 的值为 200 并且清除 TIF 标志位，当 CNT 的值达到 200 时，TIF 标志将再次变为 1。最后，软件改变 CMPDAT 的值为 500 并且清除 TIF 标志，当 CNT 的值达到 500 时，TIF 标志将再次变为 1。这种模式就是定时器连续计数模式。

这种模式下，定时器不停计数，因此，这种操作模式叫为连续计数模式。

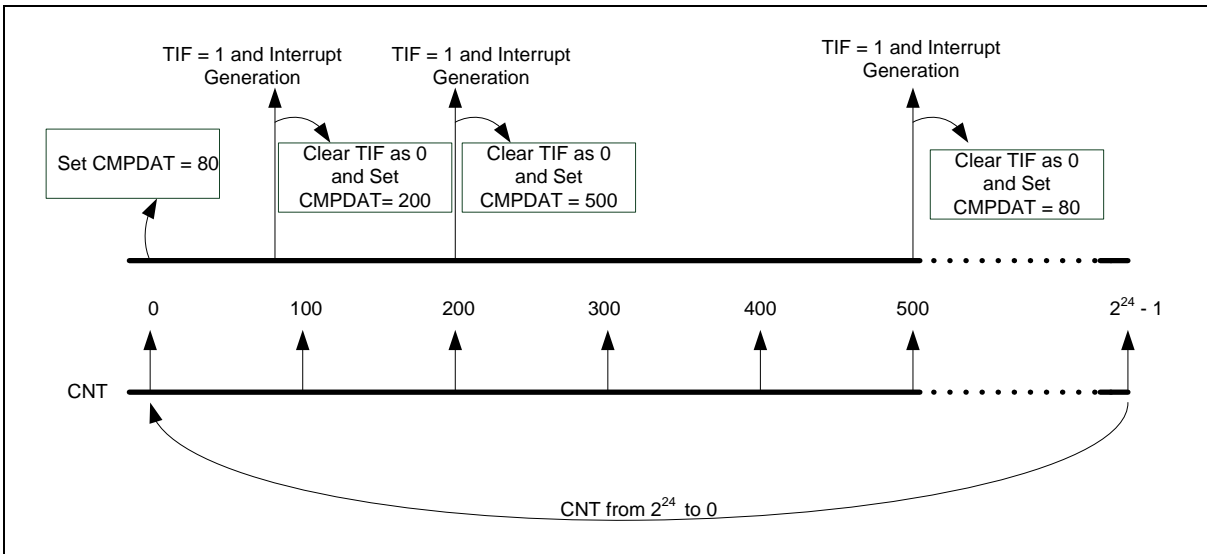


图 6.6-3 连续计数模式

6.6.5.7 事件计数 模式

定时器控制器也提供这样的应用，能对来自管脚 TMx (x= 0~1) 输入事件计数并将事件的次数反应到 CNT (TIMERx_CNT[23:0]) 的值。也可以称为事件计数功能。该功能下，EXTCNTEN (TIMERx_CTL[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 ECNTDBEN (TIMERx_EXTCTL[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 $1/3$ HCLK，如果消抖电路打开，输入事件的频率须小于 $1/8$ HCLK，以保证 CNT 的值是正确的。软件也可以通过设置 CNTPHASE (TIMERx_EXTCTL[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 one-shot, 周期 和 连续计数模式，通过 CNT 寄存器来计算来自 TMx 管脚的输入事件。

6.6.5.8 捕捉功能

输入捕捉或复位功能是用来捕捉或复位计数器的值。捕捉功能通过 CAPMODE (TIMERx_EXTCTL[8]) 寄存器，可配置为 free-counting 和 trigger-counting 捕捉模式。free-counting 捕捉模式,复位模式, trigger-counting 捕捉模式详细描述如下。

6.6.5.9 Free-Counting 捕捉模式

事件捕捉功能是当检测到 ACMPOx (x= 0~1)管脚边沿电平有变化时，CNT (TIMERx_CNT[23:0])会送到 CAPDAT (TIMERx_CAP[23:0])。在该模式下，需把 CAPMODE (TIMERx_EXTCTL[8]) 和 CAPFUNCS (TIMERx_EXTCTL[4])位设置为0，用来选择ACMPOx (x= 0~1)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。

这种模式下能通过设置CAPEDGE (TIMERx_EXTCTL[2:1])来选择ACMPOx (x= 0~1)边沿变化检测方式。

在Free-Counting捕捉模式，用户不用考虑定时器计数器工作模式的选择，只有当检测到ACMPOx (x= 0~1)管脚有边沿变化时捕捉事件才会发生。

如果CPU不清除 CAPIF (TIMERx_EINTSTS[0]) 状态标志，用户应知道此时的定时器会保持 TIMERx_CAP 寄存器的值不变，会丢掉新的捕捉值。这种操作模式如表 6.6-1.所示。

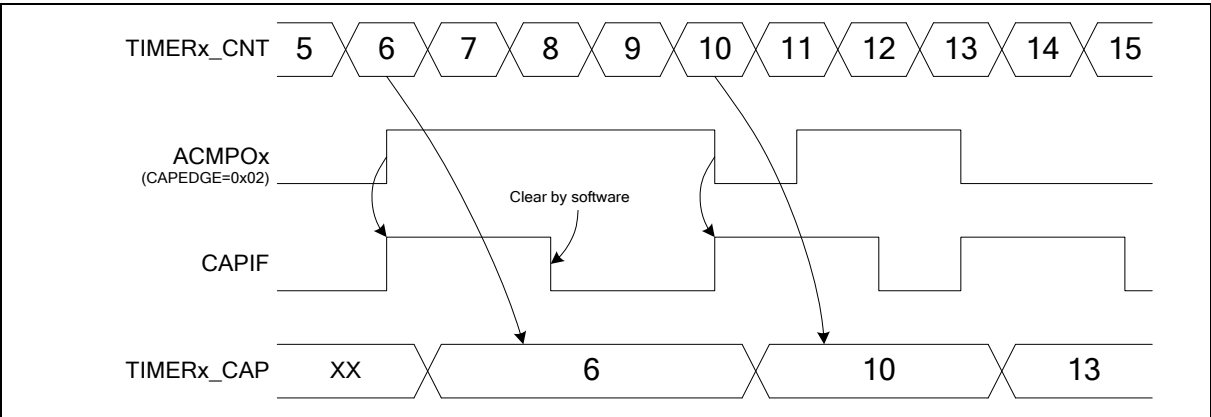


图 6.6-4 Free-Counting 捕捉 模式

6.6.5.10 复位计数器模式

当检测到 ACMPOx (x= 0~1)有边沿转变时，定时器同样提供复位计数器功能来复位 CNT (TIMERx_CNT[23:0]) 的值。在该模式，大部分设置与 Free-Counting 捕捉功能相同，除了 CAPFUNCS (TIMERx_EXTCTL[4])位必须设置为1来选择ACMPOx (x= 0~1)变化时用来触发复位计数器的值。这种工作模式如表 6.6-1所示。

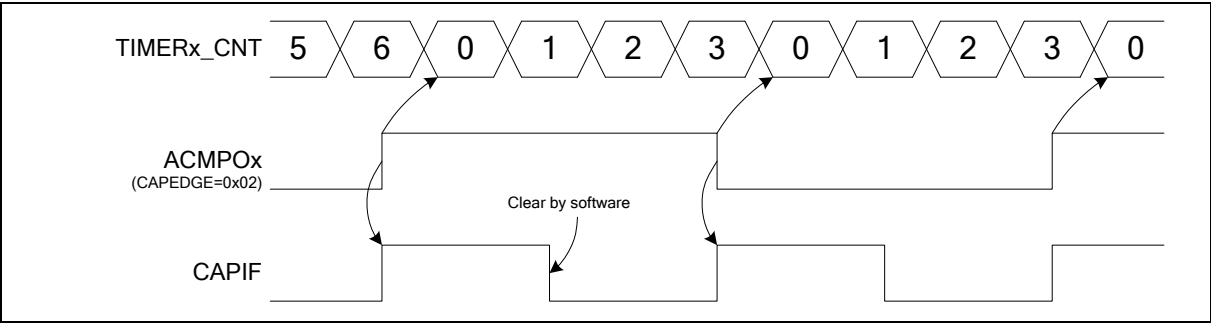


表 6.6-5 外部复位计数模式

6.6.5.11 Trigger-Counting 捕捉模式

如果CAPMODE (TIMERx_EXTCTL[8])位和CAPEN (TIMERx_EXTCTL[3])被置1，CAPFUNCS (TIMERx_EXTCTL[4])被置0，CNT将会被清零，然后当ACMPOx (x= 0~1)管脚触发条件产生时捕捉到数据放入CAPDAT寄存器。ACMPOx触发边沿可以通过CAPEDGE (TIMERx_EXTCTL[2:1])位来设置。具体的操作办法请参考表 6.6-1。当ACMPOx (x= 0~1)触发产生后，CAPIF (TIMERx_EINTSTS[0])被置1，如CAPIEN (TIMERx_EXTCTL[5])是被置1，中断信号产生，并被发送到NVIC 来告知CPU。

功能	捕捉模式 (TIMERx_EXTCTL[8])	捕捉功能 (TIMERx_EXTCTL[4])	捕捉边沿 (TIMERx_EXTCTL[2:1])	操作描述
Free-counting 捕捉模式	0	0	00	ACMPOx (x= 0~1) 管脚检测到由1变0，CNT 的值被捕捉到CAPDAT.
	0	0	01	ACMPOx (x= 0~1) 管脚检测到由0变1，CNT 的值被捕捉到CAPDAT.
	0	0	10	ACMPOx (x= 0~1) 管脚检测到由1变0或由0变1，CNT 的值被捕捉到CAPDAT.
	0	0	11	保留
外部复位计数模式	0	1	00	ACMPOx (x= 0~1)管脚检测到由1变0. CNT 被清0.
	0	1	01	ACMPOx (x= 0~1)管脚检测到由0变1. CNT 被清0.
	0	1	10	ACMPOx (x= 0~1)管脚检测到由1变0或由0变1. CNT 被清0.
	0	1	11	保留
Trigger-Counting 捕捉模式	1	0	00	下降沿触发: ACMPOx (x= 0~1) 管脚第一个由1变0，将使CNT 清0，然后开始计数，当第2个由1变0产生，计数停止。

	1	0	01	上升沿触发: ACMPOx (x= 0~1) 管脚第一个由0变1, 将使CNT清0, 然后开始计数, 当第2个由0变1产生, 计数停止.
	1	0	10	电平切换触发: ACMPOx (x= 0~1) 脚检测到由1变0, 将使CNT清0, 然后开始计数, 当由0变1停止计数.
	1	0	11	电平切换触发: ACMPOx (x= 0~1) 脚检测到由0变1, 将使CNT清0然后开始计数, 当由1变0停止计数.

表 6.6-1 输入捕捉模式操作流程

6.6.5.12 连续捕捉模式

定时器0/1提供连续捕捉模式，当输入捕获信号边沿变化能自动获得定时器计数值

设置 CCAPEN (TIMER_CCAPCTL[0]) 为 1，连续捕捉模式被使能。CNTSEL (TIMER_CCAPCTL[3:2])为00，意味着当输入捕捉信号边沿改变，定时器0计数器值将被锁存。CNTSEL是1意味着定时器1计数值被锁存。

CCAPEN使能后，如果输入捕捉信号边沿改变，第一个上升沿信号将锁存定时器计数器值到TIMER_CCAP0，同时设置CAPR1F (TIMER_CCAPCTL[8])为高。当CAPR1F为高时，第一个下降沿将锁存定时器计数器值到TIMER_CCAP1，同时设置CAPF1F (TIMER_CCAPCTL[9])为高。当CAPF1F为高时，第二个上升沿信号将锁存定时器计数器值到TIMER_CCAP2，同时设置CAPR2F (TIMER_CCAPCTL[10])为高。当CAPR2F为高时，第二个下降沿将锁存定时器计数器值到TIMER_CCAP3，同时设置CAPF2F (TIMER_CCAPCTL[11])为高。

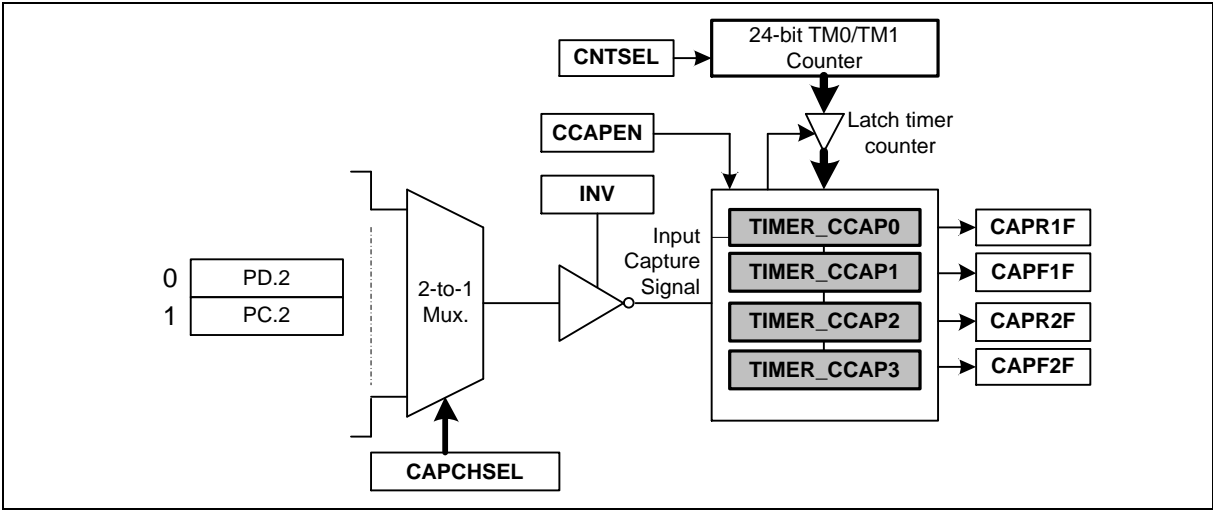


图 6.6-6 连续捕捉模式框图

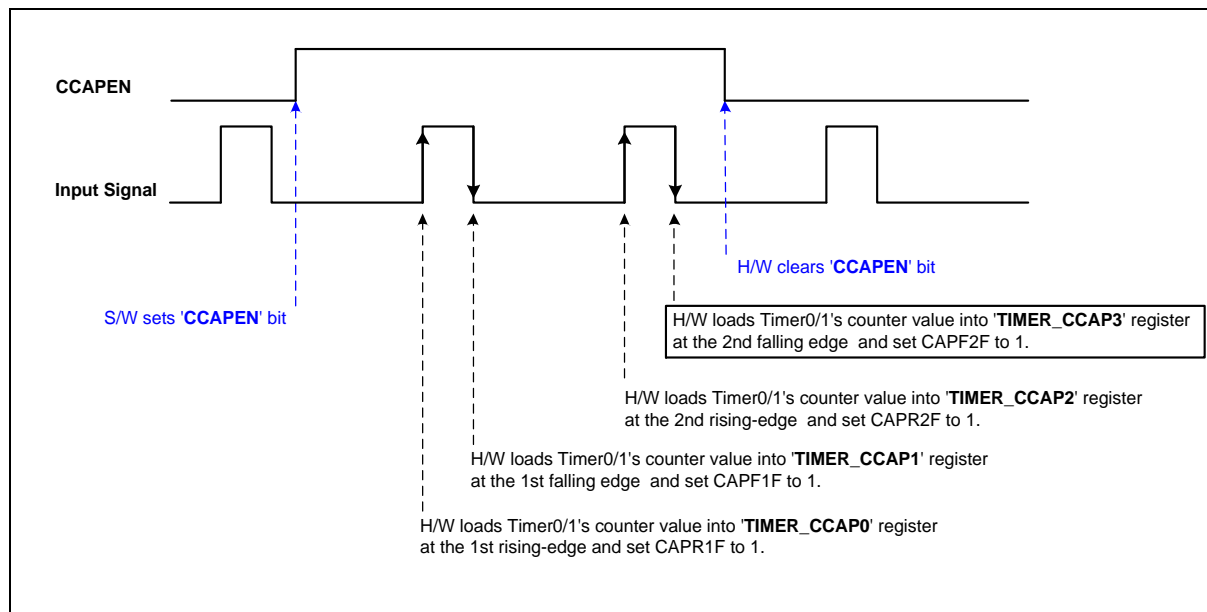


图 6.6-7 连续捕捉模式时序

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 TMR_BA = 0x4001_0000				
TIMER0_CTL	TMR_BA+0x00	R/W	Timer0 控制和状态寄存器	0x0000_0005
TIMER0_CMP	TMR_BA+0x04	R/W	Timer0比较寄存器	0x0000_0000
TIMER0_INTSTS	TMR_BA+0x08	R/W	Timer0中断状态寄存器	0x0000_0000
TIMER0_CNT	TMR_BA+0x0C	R	Timer0数据寄存器	0x0000_0000
TIMER0_CAP	TMR_BA+0x10	R	Timer0捕捉数据寄存器	0x0000_0000
TIMER0_EXTCTL	TMR_BA+0x14	R/W	Timer0外部事件控制寄存器	0x0000_0000
TIMER0_EINTSTS	TMR_BA+0x18	R/W	Timer0外部事件中断状态寄存器	0x0000_0000
TIMER1_CTL	TMR_BA+0x20	R/W	Timer1控制和状态寄存器	0x0000_0005
TIMER1_CMP	TMR_BA+0x24	R/W	Timer1比较寄存器	0x0000_0000
TIMER1_INTSTS	TMR_BA+0x28	R/W	Timer1中断状态寄存器	0x0000_0000
TIMER1_CNT	TMR_BA+0x2C	R	Timer1数据寄存器	0x0000_0000
TIMER1_CAP	TMR_BA+0x30	R	Timer1捕捉数据寄存器	0x0000_0000
TIMER1_EXTCTL	TMR_BA+0x34	R/W	Timer1外部事件控制寄存器	0x0000_0000
TIMER1_EINTSTS	TMR_BA+0x38	R/W	Timer1外部事件中断状态寄存器	0x0000_0000
TIMER_CCAPCTL	TMR_BA+0x40	R/W	定时器连续捕捉控制寄存器	0x0000_0000
TIMER_CCAP0	TMR_BA+0x44	R	定时器连续捕捉数据寄存器0	0x0000_0000
TIMER_CCAP1	TMR_BA+0x48	R	定时器连续捕捉数据寄存器1	0x0000_0000
TIMER_CCAP2	TMR_BA+0x4C	R	定时器连续捕捉数据寄存器2	0x0000_0000
TIMER_CCAP3	TMR_BA+0x50	R	定时器连续捕捉数据寄存器3	0x0000_0000

6.6.7 寄存器描述

定时器控制寄存器((TIMERx CTL)

寄存器	偏移地址	R/W	描述	复位值
TIMER0_CTL	TMR_BA+0x00	R/W	Timer0控制和状态寄存器	0x0000_0005
TIMER1_CTL	TMR_BA+0x20	R/W	Timer1控制和状态寄存器	0x0000_0005

31	30	29	28	27	26	25	24
ICEDEBUG	CNTEN	INTEN	OPMODE		RSTCNT	ACTSTS	EXTCNTEN
23	22	21	20	19	18	17	16
WKEN	Reserved					CMPCTL	CNTDATEN
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PSC							

位	描述	
[31]	ICEDEBUG	<p>仿真器 (ICE) 调试模式响应禁止位 (写保护位)</p> <p>0 = 仿真器调试模式响应影响定时器计数。</p> <p>当调试器调试模式响应时, 定时器计数器将被固定住。</p> <p>1 = 仿真器调试模式响应禁用。</p> <p>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。</p>
[30]	CNTEN	<p>定时器计数使能位</p> <p>0 = 停止/暂停计数</p> <p>1 = 开始计数</p> <p>注意1: 在停止状态, 设置CNTEN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。</p> <p>注意2: 在 one-shot 模式下(TIMERx_CTL[28:27] = 00), 当定时中断标志TIF 产生时, 该位由硬件自动清零。</p>
[29]	INTEN	<p>定时器中断使能位</p> <p>0 = 禁止定时器中断</p> <p>1 = 使能定时器中断</p> <p>注意: 如果该位使能, 当定时器中断标志TIF置 1, 定时器中断信号产生并通知CPU。</p>
[28:27]	OPMODE	<p>定时器操作模式</p> <p>00 = 定时器工作在单触发模式 (one-shot)。如果INTEN被使能, 相应的中断信号将产生一次, CNTEN将自动被硬件清除。</p> <p>01 = 定时器工作在周期模式 (Periodic)。如果INTEN被使能, 相应的中断信号将周期性产生</p> <p>10 = 定时器工作在Toggle 模式。如果INTEN被使能, 相应的中断信号将周期性产生。相应的信号脚(tout)将前后变化输出50%占空比周期信号。</p>

位	描述	
		11 = 定时器工作在连续计数模式(Continuous Counting)。如果INTEN使能, 相应的中断信号将产生当TIMERx_CNT = TIMERx_CMP。同时, 24位向上计数器不停连续向上计数。请参考6.12.5.2关于连续计数模式操作的详细描述。
[26]	RSTCNT	定时器复位 0 = 该位写 0 无效 1 = 如果ACTSTS为1, 复位8-位预分频计数器, 内部24-位向上计数器的值和CNTEN位
[25]	ACTSTS	定时器激活状态位 (只读) 该位表示24位向上计数器的状态。 0 = 定时器计数器未激活状态 1 = 定时器计数器激活状态
[24]	EXTCNTEN	计数器模式使能位 该位用来使能外部计数管脚功能。当定时器用作事件计数器, 该位需要设置成1.并选择HCLK作为定时器时钟源。详细请参考章节事件计数模式。 0 = 禁用事件计数器模式 1 = 使能事件计数器模式
[23]	WKEN	唤醒功能使能位 如果WKEN置1, 当标志TIF或CAPIF被置1, 定时器中断信号将产生一个唤醒触发事件给CPU 0 = 如果定时器中断信号产生, 唤醒功能禁止 1 = 如果定时器中断信号产生, 唤醒功能使能
[22:18]	Reserved	保留。
[17]	CMPCTL	定时器比较模式选择控制位 0 = 在One-shot或Periodic模式下, 当写新的CMPDAT, 定时器计数器将复位。 1 = 在One-shot或Periodic模式下, 当写新的CMPDAT, 如果新的CMPDAT > CNT (TIMERx_CNT[23:0])(当前计数值), 定时器计数器将继续保持计数不会复位, 如果新的CMPDAT <= CNT (TIMERx_CNT[23:0])(当前计数值), 定时器计数器将复位。
[16]	CNTDATEN	数据载入使能位 当CNTDATEN置位, CNT (TIMERx_CNT[23:0]) (定时器数据寄存器)将不断随着24位向上定时器的计数值更新 0 = 禁止定时器数据寄存器更新。 1 = 使能定时器数据寄存器更新当定时器计数器有效。
[15:8]	Reserved	保留。
[7:0]	PSC	预分频计数器 定时器的输入时钟源被(PSC+1)预分频, 然后再输入到定时器。如果 该位为0(PSC = 0), 表示未分频。

定时器比较寄存器(TIMERx_CMP)

寄存器	偏移地址	R/W	描述	复位值
TIMER0_CMP	TMR_BA+0x04	R/W	Timer0比较寄存器	0x0000_0000
TIMER1_CMP	TMR_BA+0x24	R/W	Timer1比较寄存器	0x0000_0000

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

位	描述	
[31:24]	Reserved	保留.
[23:0]	CMPDAT	<p>定时器比较值</p> <p>CMPDAT是24-位比较寄存器。当内部 24-位向上计数器的值等于CMPDAT的值时，TIF将被置1。</p> <p>超时溢出周期 = (定时器输入时钟源周期) * (8位 PSC + 1) * (24位 CMPDAT)</p> <p>注意1: 不能向CMPDAT里写 0x0 或 0x1，否则内核将运行到未知状态。</p> <p>注意2: 当定时器工作在 continuous counting 模式，即使软件写一个新的值到CMPDAT，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除M05xxDN/DE的周期模式外，如果软件写一个新的值到CMPDAT，定时器复位当前计数值，开始重新计数，并与新的比较值做比较。</p>

定时器中断状态寄存器(TIMERx_INTSTS)

寄存器	偏移地址	R/W	描述	复位值
TIMER0_INTSTS	TMR_BA+0x08	R/W	Timer0中断状态寄存器	0x0000_0000
TIMER1_INTSTS	TMR_BA+0x28	R/W	Timer1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						TWKF	TIF

位	描述	
[31:2]	Reserved	保留.
[1]	TWKF	<p>定时器唤醒标志</p> <p>该位表示定时器的中断唤醒标志状态。</p> <p>0 = 定时器不会引起CPU 唤醒</p> <p>1 = 如果定时器中断信号产生, CPU 从空闲或掉电模式唤醒</p> <p>注意: 该位必须通过软件写1清0</p>
[0]	TIF	<p>定时器中断标志</p> <p>当CNT (TIMERx_CNT[23:0])的值与定时器比较值CMPDAT匹配时, 该位指示其中断状态。</p> <p>0 = 无影响</p> <p>1 = CNT的值与CMPDAT的值相匹配</p> <p>注意: 该位写 1 清零。</p>

定时器数据寄存器((TIMERx_CNT)

寄存器	偏移地址	R/W	描述	复位值
TIMER0_CNT	TMR_BA+0x0C	R	Timer0数据寄存器	0x0000_0000
TIMER1_CNT	TMR_BA+0x2C	R	Timer1数据寄存器	0x0000_0000

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

位	描述	
[31:24]	Reserved	保留.
[23:0]	CNT	定时器数据寄存器 如果CNTDATEN被置1，CNT寄存器值将不停更新来反映24位向上计数器值

定时器捕捉数据寄存器(TIMERx_CAP)

寄存器	偏移地址	R/W	描述	复位值
TIMER0_CAP	TMR_BA+0x10	R	Timer0捕捉数据寄存器	0x0000_0000
TIMER1_CAP	TMR_BA+0x30	R	Timer1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
CAPDAT							
15	14	13	12	11	10	9	8
CAPDAT							
7	6	5	4	3	2	1	0
CAPDAT							

位	描述	
[31:24]	Reserved	保留.
[23:0]	CAPDAT	定时器捕捉数据寄存器 当CAPEN (TIMERx_EXTCTL[3])位被置1, CAPFUNCS (TIMERx_EXTCTL[4])位为 0, ACMPOx管脚的变化与 CAPEEDGE (TIMERx_EXTCTL[2:1]) 设定相匹配时, CAPIF (TIMERx_EINTSTS[0])将被置1且当前定时器计数器CNT (TIMERx_CNT[23:0])的值将被自动载入到CAPDAT.

定时器外部事件控制寄存器 (TIMERx_EXTCTL)

寄存器	偏移地址	R/W	描述	复位值
TIMER0_EXTCTL	TMR_BA+0x14	R/W	Timer0外部事件控制寄存器	0x0000_0000
TIMER1_EXTCTL	TMR_BA+0x34	R/W	Timer1外部事件控制寄存器	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							CAPMODE
7	6	5	4	3	2	1	0
ECNTDBEN	Reserved	CAPIEN	CAPFUNCS	CAPEN	CAPEDGE		CNTPHASE

位	描述	
[31:9]	Reserved	保留.
[8]	CAPMODE	<p>捕捉模式选择位</p> <p>0 = 定时器计数复位功能或定时器捕捉功能free-counting 模式.</p> <p>1 = 定时器捕捉功能Trigger-counting模式</p>
[7]	ECNTDBEN	<p>定时器外部计数器输入管脚防抖动使能位</p> <p>0 = TMx (x= 0~1)脚防抖禁用</p> <p>1 = TMx (x= 0~1)管脚防抖使能</p> <p>注意: 如果该位使能, TMx(x = 0~1)管脚边沿检测带防抖动电路。</p>
[6]	Reserved	保留.
[5]	CAPIEN	<p>定时器捕捉中断使能位</p> <p>0 = 定时器捕捉中断禁止.</p> <p>1 = 定时器捕捉中断使能</p> <p>注意: CAPIEN 用于使能定时器捕捉中断。如果 CAPIEN 使能, 当 CAPIF (TIMERx_EINTSTS[0])标志设为1时, 定时器会产生一个中断信号。</p> <p>例如, 当 CAPIEN = 1, CAPEN = 1, 且 CAPEDGE = 00, ACMPOx 管脚上的由1变0变化, 将导致CAPIF 被置位, 然后中断信号产生并发送到NVIC并告知CPU.</p>
[4]	CAPFUNCS	<p>捕捉功能选择位</p> <p>0 =外部捕捉模式使能.</p> <p>1 =复位模式使能.</p> <p>注意1: 当 CAPFUNCS为 0, ACMPOx管脚上的变化用于保存 24-位定时器计数值到CAPDAT.</p> <p>注意2: 当 CAPFUNCS为 1, ACMPOx管脚上的变化用于复位24-位定时器计数值</p>

位	描述	
[3]	CAPEN	定时器捕捉使能 该位使能定时器捕捉功能 0 = 定时器捕捉功能禁止。 1 = 定时器捕捉功能使能。
[2:1]	CAPEDGE	定时器外部捕捉管脚边沿检测选择 00 = ACMPOx管脚上 下降沿将被检测。 01 = ACMPOx管脚上 上升沿将被检测。 10 = ACMPOx管脚上 上升沿或下降沿将被检测。 11 = 保留。
[0]	CNTPHASE	定时器外部计数管脚相位检测选择 该位表示TMx (x= 0~1)脚相位检测。 0 = TMx (x = 0~1)管脚的下降沿将被计数。 1 = TMx (x = 0~1)管脚的上升沿将被计数。

定时器外部事件中断状态寄存器 (TIMERx_EINTSTS)

寄存器	偏移地址	R/W	描述	复位值
TIMER0_EINTSTS	TMR_BA+0x18	R/W	Timer0外部事件中断状态寄存器	0x0000_0000
TIMER1_EINTSTS	TMR_BA+0x38	R/W	Timer1外部事件中断状态寄存器	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							CAPIF

位	描述	
[31:1]	Reserved	保留.
[0]	CAPIF	<p>定时器捕捉中断标志位 该位表示定时器外部捕捉中断标志状态。 0 = 定时器捕捉中断没有发生。 1 = 定时器捕捉中断发生。 注意1: 该位写 1 清零。 注意2: 当CAPEN (TIMERx_EXTCTL[3]) 位被置1, CAPFUNCS (TIMERx_EXTCTL[4]) 位为 0, ACMPOx管脚的变化与CAPEDGE (TIMERx_EXTCTL[2:1])设定匹配时, 该位将被硬件置1。 注意3: CPU 需要清 CAPIF 状态在检测到一个新的捕捉事件前。如果未清CAPIF, 定时器将保持TIMERx_CAP的值不变并丢掉新的捕捉值。</p>

定时器连续捕捉控制寄存器 (TIMER_CCAPCTL)

寄存器	偏移地址	R/W	描述	复位值
TIMER_CCAPCTL	TMR_BA+0x40	R/W	定时器连续捕捉控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						CCAPIEN	
15	14	13	12	11	10	9	8
Reserved				CAPF2F	CAPR2F	CAPF1F	CAPR1F
7	6	5	4	3	2	1	0
Reserved			CAPCHSEL	CNTSEL		INV	CCAPEN

位	描述	
[31:18]	Reserved	保留。
[17:16]	CCAPIEN	捕捉中断使能位 00 = 中断禁止 01 = 捕捉第一个上升沿和第一个下降沿中断使能 10 = 捕捉第一个上升沿，第一个下降沿中断和第二个上升沿使能 11 = 捕捉第一个上升沿，第一个下降沿中断，第二个上升沿和第二个下降沿使能
[15:12]	Reserved	保留。
[11]	CAPF2F	捕捉第二个下降沿标志 第二个下降沿已经捕捉，该位置1 0 = 没有。 1 = CAPDAT(TIMER_CCAP3[23:0]) 数据准备好可读 注意： 写1到此位硬件自动清除
[10]	CAPR2F	捕捉第二个上升沿标志 第二个上升沿已经捕捉，该位置1 0 = 没有。 1 = CAPDAT(TIMER_CCAP2[23:0]) 数据准备好可读 注意： 写1到此位硬件自动清除
[9]	CAPF1F	捕捉第一个下降沿标志 第一个下降沿已经捕捉，该位置1 0 = 没有。 1 = CAPDAT(TIMER_CCAP1[23:0]) 数据准备好可读 注意： 写1到此位硬件自动清除

位	描述	
[8]	CAPR1F	<p>捕捉第一个上升沿标志</p> <p>第一个上升沿已经捕捉，该位置1</p> <p>0 = 没有.</p> <p>1 = CAPDAT(TIMER_CCAP0[23:0]) 数据准备好可读</p> <p>注意： 写1到此位硬件自动清除</p>
[7:5]	Reserved	保留.
[4]	CAPCHSEL	<p>捕捉定时器通道选择</p> <p>选择连续捕捉数据通道。</p> <p>0 = PD.2.</p> <p>1 = PC.2.</p>
[3:2]	CNTSEL	<p>捕捉定时器选择</p> <p>选择连续捕捉输入信号的定时器</p> <p>00 = 定时器0.</p> <p>01 = 定时器1.</p> <p>10 = SysTick.</p> <p>11 = 保留.</p>
[1]	INV	<p>输入信号翻转</p> <p>翻转输入的捕捉信号</p> <p>0 = 没有.</p> <p>1 = 翻转.</p>
[0]	CCAPEN	<p>连续捕捉使能位</p> <p>此位使能连续捕捉功能</p> <p>0 = 连续捕捉功能禁止</p> <p>1 = 连续捕捉功能使能</p> <p>注意： 写0到此位清除此位或者当捕捉操作完成，硬件自动清除此位</p>

定时器连续捕捉寄存器0-3 (TIMER_CCAP0-3)

寄存器	偏移地址	R/W	描述	复位值
TIMER_CCAP0	TMR_BA+0x44	R	定时器连续捕捉数据寄存器0	0x0000_0000
TIMER_CCAP1	TMR_BA+0x48	R	定时器连续捕捉数据寄存器1	0x0000_0000
TIMER_CCAP2	TMR_BA+0x4C	R	定时器连续捕捉数据寄存器2	0x0000_0000
TIMER_CCAP3	TMR_BA+0x50	R	定时器连续捕捉数据寄存器3	0x0000_0000

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

位	描述	
[31:24]	Reserved	保留.
[23:0]	CAPDAT	定时器连续捕捉数据寄存器 TIMER_CCAP0 保存第一个上升沿定时器计数器值 TIMER_CCAP1保存第一个下降沿定时器计数器值 TIMER_CCAP2保存第二个上升沿定时器计数器值 TIMER_CCAP3保存第二个下降沿定时器计数器值

6.7 增强型输入捕捉定时器 (ECAP)

6.7.1 概述

这个外设提供输入捕捉定时器/计数器，能够检测通道输入信号的数字边沿变化。这个模块有3个输入捕捉通道。定时器/计数器配备有向上计数，重载和比较匹配功能。

6.7.2 特性

- 24 位输入捕捉向上计数定时器/计数器
- 3 个输入通道都有独立的捕捉计数保持寄存器。
- 噪音滤波器在输入端口前端
- 三选项边沿检测：
 - ◆ 上升沿检测
 - ◆ 下降沿检测
 - ◆ 上升/下降沿检测
- 作为输入源支持 ADC 比较输出和 ACMP 输出
- 捕捉到事件后复位或者重载捕捉计数器
- 支持比较匹配功能
- 支持中断功能

6.7.3 框图

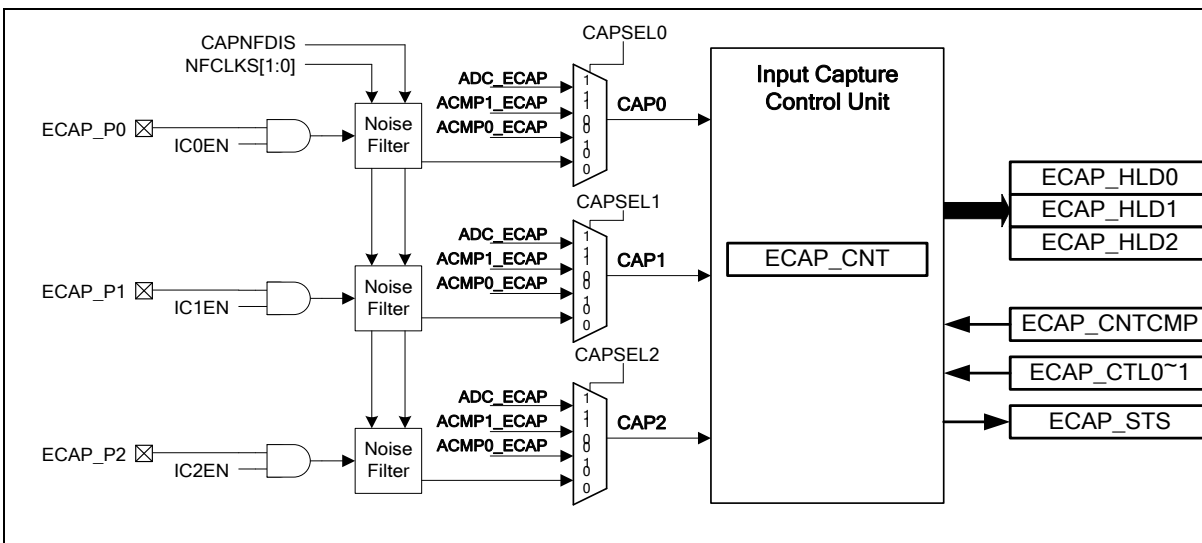


图 6.7-1 增强型输入捕捉定时器/计数器框图

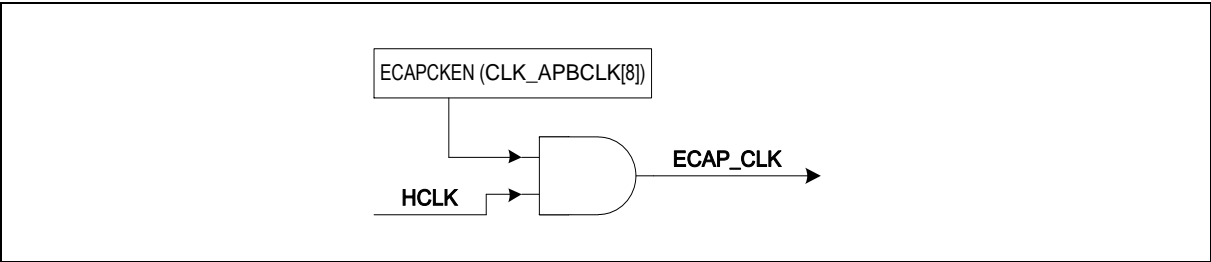


图 6.7-2 增强型输入捕捉定时器/计数器时钟源控制

图 6.7-1 为输入捕捉框图。输入捕捉定时器/计数器有3个可编程输入信号源的输入通道。引脚 ECAP_P0 到 ECAP_P2能用作输入捕捉单元可以配置噪音滤波器或不配置 (CAPNFDIS = 1)。此外，通过设置 ECAP_CTL0(CAPSEL0~ CAPSEL 2)，模拟比较器输出 (ACMPn_ECAPH)，和 ADC 比较输出 (ADC_ECAPH) 也能被内部连接用作捕捉输入。

6.7.4 输入噪音滤波器

图 6.7-3为4个采样率的噪音滤波器的架构图。

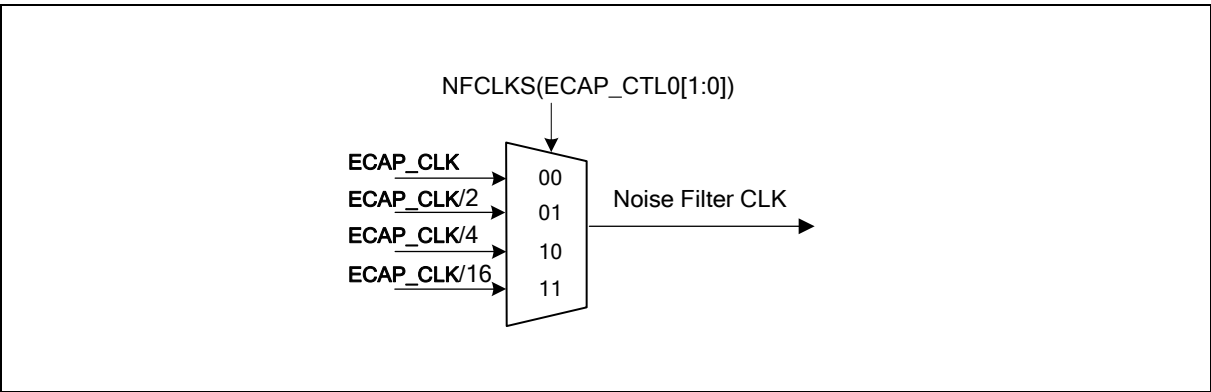


图 6.7-3 噪音过滤采样时钟选项

如果使能，捕捉逻辑要求连续采样4次同一个捕捉输入值以此来识别边沿用作捕捉事件。图 6.7-4. 为数字噪音滤波器的一个例子。

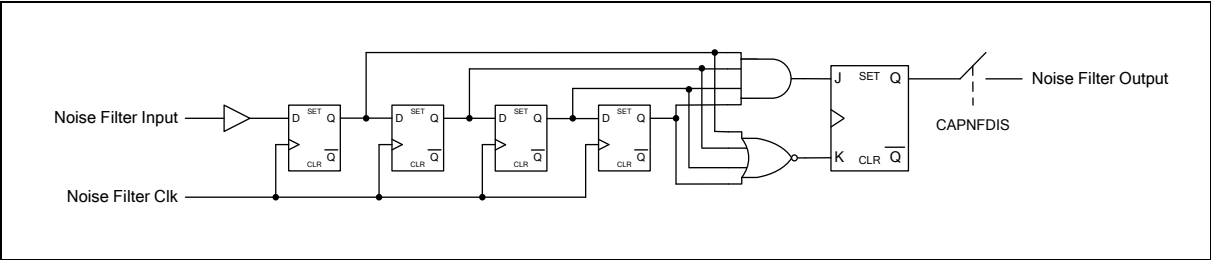


图 6.7-4 噪音滤波器

6.7.5 输入捕捉定时器/计数器操作

输入捕捉定时器/计数器包含2个主要功能模块，捕捉模块和操作模块。捕捉模块有3个输入捕捉单元作为3个输入通道使用。

捕捉单元用于检测和测量脉冲宽度和方波的周期。在输入捕捉模块，输入通道0到2在输入捕捉模块都有自己的边沿检测器，但在操作模块，它们共用一个捕捉定时器/计数器，ECAP_CNT。边沿检测可通过CAPEDG (ECAP_CTL1[5,4], [3,2], [1,0]) 寄存器编程，支持上升沿，下降沿和上升/下降沿触发。每个捕捉单元包含一个使能位，IC0EN ~ IC2EN (ECAP_CTL0[6:4]) 来使能/禁止每个输入通道，和一个状态位CAP0 ~ CAP2 (ECAP_STATUS[10:8]) 以便软件监测每个通道当前状态。

输入捕捉支持重载模式和比较模式。两种模式来说，捕捉计数器 (ECAP_CNT) 作为一个24位的向上计数器，它的时钟由时钟分频器输出和CPTST控制。时钟分频器的时钟源可以通过CAPDIV[2:0] 来分频为1, 4, 16, 32, 64, 96, 112 和 128，可通过编程CNTSRC[1:0] 配置时钟源来自系统时钟源，ECAP_CLK 或输入通道CAP0 ~ CAP2。在重载模式下，ECAP_CNTCMP作为一个重载寄存器，而在比较模式下，ECAP_CNTCMP作为一个比较寄存器。输入捕捉定时器/计数器使能位 (CAPEN) 必须被置1来使能输入捕捉定时器/计数器功能。更多详细操作在后面描述。

6.7.5.1 捕捉功能

当捕捉输入检测到一个有效的边沿变化，它会触发一个有效的捕捉事件 (CAPTE0~2)，这样24位捕捉计数器ECAP_CNT内容将被捕捉/传输到相应被触发通道的捕捉保持寄存器ECAP_HLD0~2。这个事件也会导致相应的标志 CAPTFx (ECAP_STS[2:0]) 被置位，如果相应的中断使能位 CAPTFxIEN (ECAP_CTL0[18:16]) 被置1，将会产生中断。触发标志由硬件置位，软件来清除。软件能读寄存器ECAP_STS来得到标志的状态，写1到ECAP_STS相应位用来清除标志。

另外，设置CPTCLR (ECAP_CTL0[26]) 当捕捉事件发生，将允许硬件自动的复位捕捉计数器

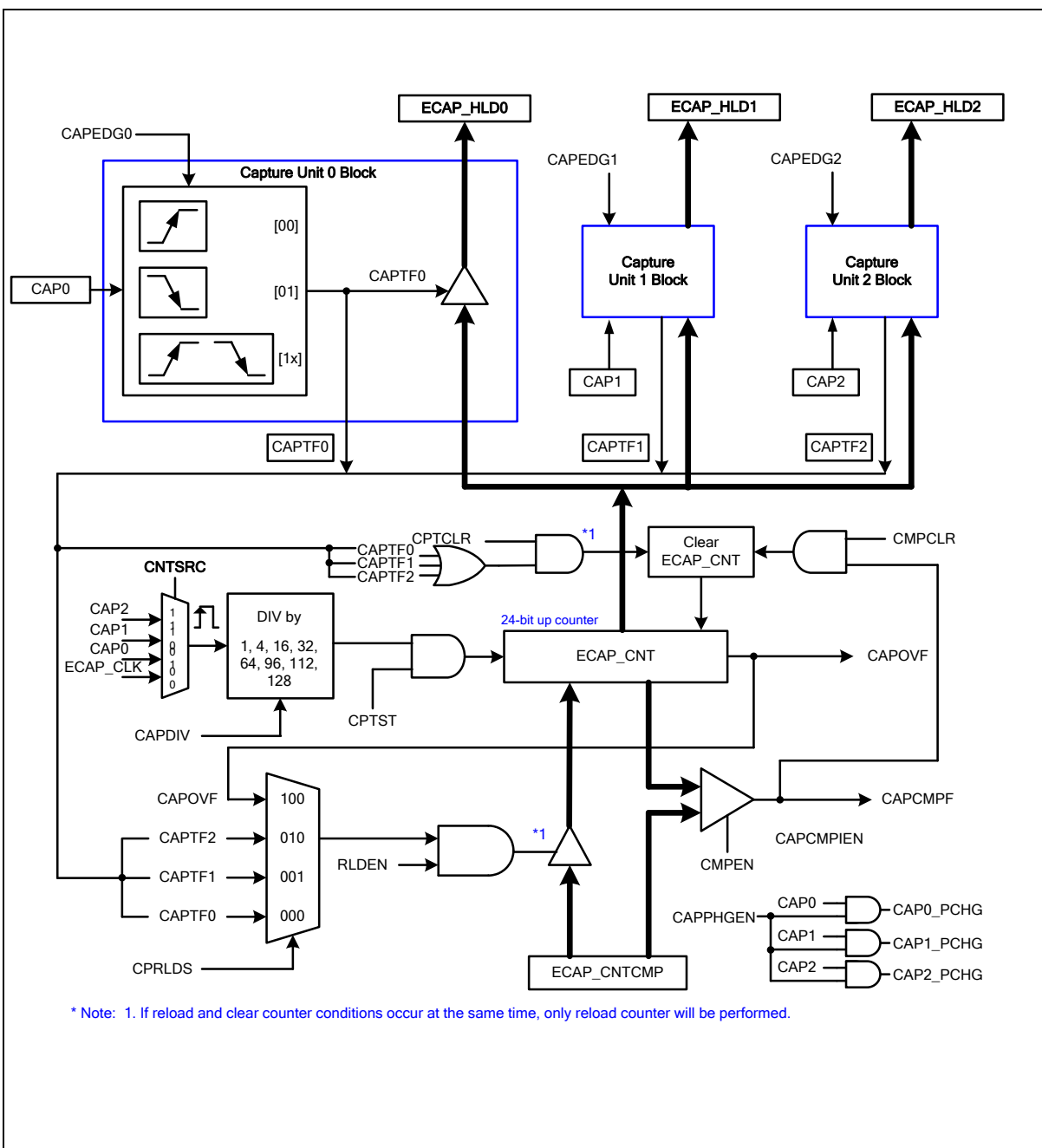


图 6.7-5 增强型输入捕捉定时器/计数器功能方框图

6.7.5.2 比较模式

通过设定CMPEN (ECAP_CTL0[28])为1来使能比较功能，ECAP_CNTCMP将作为比较寄存器。当ECAP_CNT向上计数到达ECAP_CNTCMP的值，CAPCMPF (ECAP_STS[4])将被置1，如果设置CAPCMPIEN (ECAP_CTL0[21])使能比较中断，那么将产生一个中断 (CMP_INT)。

此外，置位CMPCLR (ECAP_CTL0[25])，在一个比较匹配事件发生后，将允许硬件自动清除捕捉计

计数器为0。

6.7.5.3 重载模式

输入捕捉定时器/计数器也能配置为重载模式。设置RLDEN (ECAP_CTL0[27])为1将使能重载功能。

重载模式下, ECAP_CNTCMP将作为重载寄存器。通过配置CPRLDS (ECAP_CTL1[10:8]), 当ECAP_CNT溢出, CAPTF2, CAPTF1 或 CAPTF0被置位后, 一个重载事件的产生将使寄存器CAP_CNTCMP载入到寄存器ECAP_CNT。

值得注意的是, 如果CPTCLR 也和 RLDEN一样被置位, 当一个有效的触发事件 (CAPTFx) 发生, 仅仅重载功能被执行。

6.7.6 输入捕捉定时器/计数器中断架构

图 6.7-6是输入捕捉定时器/计数器中断模块结构的事例。在一个输入捕捉单元, 有5个中断源 (OVF_INT, CMP_INT, CAPTF0_INT~CAPTF2_INT), 它们逻辑或在一起。每一个中断源都有中断标志 (CAPOVF, CAPCMPF, CAPTF0~CAPTF2), 当打开中断使能位 (CAPOVIEN, CAPCMPIEN, CAPTF0IEN ~ CAPTF2IEN), 就能触发中断 (ECAP_INT)。

必须注意所有的中断标志都是由硬件置位, 必须由软件写1到相应的标志 (ECAP_STS[5:4], [2:0]) 来清除。

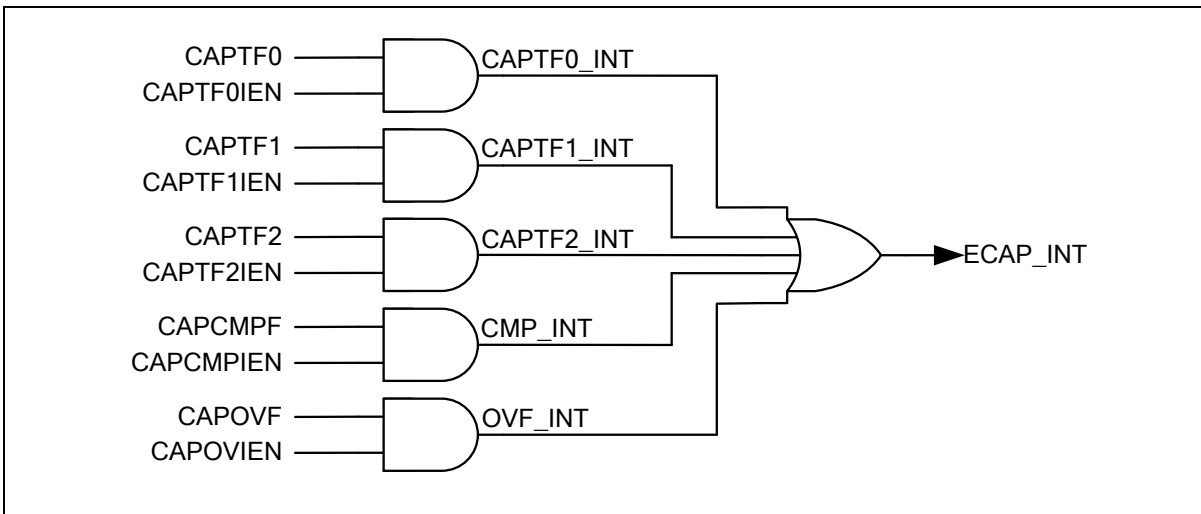


图 6.7-6 增强型输入捕捉定时器/计数器中断架构图

6.7.7 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
ECAP基地址: ECAP_BA = 0x401B_0000				
ECAP_CNT	ECAP_BA+0x00	R/W	输入捕捉计数器	0x0000_0000
ECAP_HLD0	ECAP_BA+0x04	R/W	输入捕捉计数器保持寄存器 0	0x0000_0000
ECAP_HLD1	ECAP_BA+0x08	R/W	输入捕捉计数器保持寄存器 1	0x0000_0000
ECAP_HLD2	ECAP_BA+0x0C	R/W	输入捕捉计数器保持寄存器 2	0x0000_0000
ECAP_CNTCMP	ECAP_BA+0x10	R/W	输入捕捉计数器比较寄存器	0x0000_0000
ECAP_CTL0	ECAP_BA+0x14	R/W	输入捕捉控制寄存器 0	0x0000_0000
ECAP_CTL1	ECAP_BA+0x18	R/W	输入捕捉控制寄存器 1	0x0000_0000
ECAP_STS	ECAP_BA+0x1C	R/W	输入捕捉状态寄存器	0x0000_0000

6.7.8 寄存器描述

输入捕捉计数器(ECAP_CNT)

寄存器	偏移量	R/W	描述	复位值
ECAP_CNT	ECAP_BA+0x00	R/W	输入捕捉计数器	0x0000_0000

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

位	描述	
[31:24]	Reserved	保留.
[23:0]	CNT	输入捕捉定时器/计数器 输入捕捉定时器/计数器时一个24位向上计数器。计数的时钟源来自时钟分频器

输入捕捉计数器保持寄存器 0-2 (ECAP_HLD0-2)

寄存器	偏移量	R/W	描述	复位值
ECAP_HLD0	ECAP_BA+0x04	R/W	输入捕捉计数器保持寄存器 0	0x0000_0000
ECAP_HLD1	ECAP_BA+0x08	R/W	输入捕捉计数器保持寄存器 1	0x0000_0000
ECAP_HLD2	ECAP_BA+0x0C	R/W	输入捕捉计数器保持寄存器 2	0x0000_0000

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

位	描述	
[31:24]	Reserved	保留.
[23:0]	HOLD	输入捕捉计数器保持寄存器 当激活的输入捕捉通道探测到一个有效的信号边沿改变, ECAP_CNT的值将载入相应的保持寄存器。每个输入通道都有自己的保持寄存器ECAP_HLDx(x为0到1), 用来指示CAP0到CAP2的输入。

输入捕捉计数器比较寄存器(ECAP_CNTCMP)

寄存器	偏移量	R/W	描述	复位值
ECAP_CNTCMP	ECAP_BA+0x10	R/W	输入捕捉计数器比较寄存器	0x0000_0000

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

位	描述	
[31:24]	Reserved	保留.
[23:0]	CNTCMP	输入捕捉计数器比较寄存器 如果比较功能使能 (CMPEN = 1)，寄存器 (ECAP_CNTCMP) 用作来同捕捉寄存器 (ECAP_CNT) 相比较。 如果重载控制使能 (RLDEN = 1)，一个溢出事件或捕捉事件将触发硬件来装载寄存器 (ECAP_CNTCMP) 的值到 ECAP_CNT。

输入捕捉定时器/计数器控制寄存器0(ECAP_CTL0)

寄存器	偏移量	R/W	描述	复位值
ECAP_CTL0	ECAP_BA+0x14	R/W	输入捕捉控制寄存器0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	CAPPHGEN	CAPEN	CMPEN	RLDEN	CPTCLR	CMPCLR	CPTST
23	22	21	20	19	18	17	16
Reserved		CAPCMPIEN	CAPOVIEN	Reserved	CAPTF2IEN	CAPTF1IEN	CAPTF0IEN
15	14	13	12	11	10	9	8
Reserved		CAPSEL2		CAPSEL1		CAPSEL0	
7	6	5	4	3	2	1	0
Reserved	IC2EN	IC1EN	IC0EN	CAPNFDIS	Reserved	NFCLKS	

位	描述	
[31]	Reserved	保留.
[30]	CAPPHGEN	输入捕捉标志来触发PWM相位改变功能使能位 0 = CAPTF2, CAPTF1 或 CAPTF0 触发PWM相位改变功能禁止 1 = CAPTF2, CAPTF1 或 CAPTF0触发PWM相位改变功能使能
[29]	CAPEN	输入捕捉定时器/计数器使能位 0 = 输入捕捉功能禁止 1 = 输入捕捉功能使能
[28]	CMPEN	比较功能使能位 输入捕捉定时器/计数器用来作为动态计数寄存器ECAP_CNT和寄存器ECAP_CNTCMP比较, 如果ECAP_CNT值等于ECAP_CNTCMP, 标志CAPCMPF将置1. 0 = 比较功能禁止 1 = 比较功能使能
[27]	RLDEN	重载功能使能位 设置此位使能重载功能。如果重载控制被使能, 一个溢出事件(CAPOVF) 或捕捉事件(CAPTFx)就被硬件触发来重载ECAP_CNTCMP 到 ECAP_CNT。 0 = 禁止重载功能 1 = 使能重载功能
[26]	CPTCLR	输入捕捉计数器清除通过捕捉事件控制 如此位置1, 当任何一个捕捉事件捕捉事件(CAPTF0~3)发生, 计数器(ECAP_CNT)将被清0 0 = 禁止捕捉事件(CAPTF0~3)能清除捕捉计数器 1 = 使能捕捉事件(CAPTF0~3)能清除捕捉计数器

位	描述	
[25]	CMPCLR	<p>输入捕捉计数器清除通过比较匹配控制</p> <p>如此位置1, 当比较匹配事件(CAPCMPF = 1)发生, 计数器(ECAP_CNT)将被清0</p> <p>0 = 禁止比较匹配事件(CAPCMPF)能清除捕捉计数器(ECAP_CNT)</p> <p>1 = 使能比较匹配事件(CAPCMPF)能清除捕捉计数器(ECAP_CNT)</p>
[24]	CPTST	<p>输入捕捉计数器开始位</p> <p>设置此位为1, 捕捉计数器(ECAP_CNT)同步捕捉时钟输入(CAP_CLK) 开始向上计数</p> <p>0 = ECAP_CNT停止计数</p> <p>1 = ECAP_CNT开始向上计数</p>
[23:22]	Reserved	保留.
[21]	CAPCMPIEN	<p>使能标志CAPCMPF触发输入捕捉中断</p> <p>0 = 禁止标志CAPCMPF 来触发输入捕捉中断</p> <p>1 = 使能标志CAPCMPF 来触发输入捕捉中断</p>
[20]	CAPOVIEN	<p>使能标志CAPOVF触发输入捕捉中断</p> <p>0 = 禁止标志CAPOVF来触发输入捕捉中断</p> <p>1 = 使能标志CAPOVF来触发输入捕捉中断</p>
[19]	Reserved	保留.
[18]	CAPTF2IEN	<p>使能输入捕捉通道2中断</p> <p>0 = 禁止标志CAPTF2触发输入捕捉中断</p> <p>1 = 使能标志CAPTF2触发输入捕捉中断</p>
[17]	CAPTF1IEN	<p>使能输入捕捉通道1中断</p> <p>0 = 禁止标志CAPTF1触发输入捕捉中断</p> <p>1 = 使能标志CAPTF1触发输入捕捉中断</p>
[16]	CAPTF0IEN	<p>使能输入捕捉通道0中断</p> <p>0 = 禁止标志CAPTF0触发输入捕捉中断</p> <p>1 = 使能标志CAPTF0触发输入捕捉中断</p>
[15:14]	Reserved	保留.
[13:12]	CAPSEL2	<p>CAP2输入源选项</p> <p>00 = CAP2 输入来自引脚ECAP_P2.</p> <p>01 = CAP2 输入来自信号ACMP0_O (模拟比较器 0 输出).</p> <p>10 = CAP2输入来自信号ACMP1_O (模拟比较器 1 输出).</p> <p>11 = CAP2输入来自 信号ADC_CPR (ADC 比较输出).</p>
[11:10]	CAPSEL1	<p>CAP1输入源选项</p> <p>00 = CAP1 输入来自引脚ECAP_P1.</p> <p>01 = CAP1 输入来自信号ACMP0_O (模拟比较器 0 输出).</p> <p>10 = CAP1输入来自信号ACMP1_O (模拟比较器 1 输出).</p> <p>11 = CAP1输入来自 信号ADC_CPR (ADC 比较输出).</p>

位	描述	
[9:8]	CAPSEL0	CAP0输入源选项 00 = CAP0 输入来自引脚ECAP_P0. 01 = CAP0 输入来自信号ACMP0_O (模拟比较器 0 输出). 10 = CAP0输入来自信号ACMP1_O (模拟比较器 1 输出). 11 = CAP0输入来自 信号ADC_CPR (ADC 比较输出).
[7]	Reserved	保留.
[6]	IC2EN	使能引脚IC2作为捕捉单元输入 0 = IC2 作为输入捕捉单元禁止 1 = IC2 作为输入捕捉单元使能
[5]	IC1EN	使能引脚IC1作为捕捉单元输入 0 = IC1 作为输入捕捉单元禁止 1 = IC1 作为输入捕捉单元使能
[4]	IC0EN	使能引脚IC0作为捕捉单元输入 0 = IC0 作为输入捕捉单元禁止 1 = IC0 作为输入捕捉单元使能
[3]	CAPNFDIS	禁止输入捕捉噪音滤波器 0 =输入捕捉的噪音滤波器使能 1 =输入捕捉的噪音滤波器禁止
[2]	Reserved	保留.
[1:0]	NFCLKS	噪音滤波器时钟预分频选项 决定噪音滤波器时钟的采样频率 00 = ECAP_CLK. 01 = ECAP_CLK / 2. 10 = ECAP_CLK / 4. 11 = ECAP_CLK / 16.

输入捕捉定时器/计数器控制寄存器1 (ECAP_CTL1)

寄存器	偏移量	R/W	描述	复位值
ECAP_CTL1	ECAP_BA+0x18	R/W	输入捕捉控制寄存器1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						CNTSRC	
15	14	13	12	11	10	9	8
Reserved	CAPDIV			Reserved	CPRLDS		
7	6	5	4	3	2	1	0
Reserved		CAPEDG2		CAPEDG1		CAPEDG0	

位	描述	
[31:18]	Reserved	保留.
[17:16]	CNTSRC	捕捉定时器/计数器时钟源选项 选择捕捉定时器/计数器时钟源 00 = ECAP_CLK (默认). 01 = CAP0. 10 = CAP1. 11 = CAP2.
[15]	Reserved	保留.
[14:12]	CAPDIV	捕捉定时器时钟分频选项 捕捉定时器时钟通过CAPDIV[2:0]来控制8个预分频选项 000 = CAPCLK / 1. 001 = CAPCLK / 4. 010 = CAPCLK / 16. 011 = CAPCLK / 32. 100 = CAPCLK / 64. 101 = CAPCLK / 96. 110 = CAPCLK / 112. 111 = CAPCLK / 128.
[11]	Reserved	保留.

位	描述	
[10:8]	CPRLDS	重载触发源选项 当重载功能使能 (RLDEN = 1)，当一个重载事件来到，ECAP_CNT 将载入 ECAP_CNTCMP。 CPRLDS[2:0] 决定 ECAP_CNT 重载触发源 000 = CAPTF0. 001 = CAPTF1. 010 = CAPTF2. 100 = CAPOVF. Other = 保留.
[7:6]	Reserved	保留.
[5:4]	CAPEDG2	通道2捕捉边沿选项 输入捕捉可以检测下降沿，上升沿和下降沿/上升沿 00 = 检测上升沿 01 = 检测下降沿 1x = 检测上升沿或下降沿
[3:2]	CAPEDG1	通道1捉边沿选项 输入捕捉可以检测下降沿，上升沿和下降沿/上升沿 00 = 检测上升沿 01 = 检测下降沿 1x = 检测上升沿或下降沿
[1:0]	CAPEDG0	通道0捉边沿选项 输入捕捉可以检测下降沿，上升沿和下降沿/上升沿 00 = 检测上升沿 01 = 检测下降沿 1x = 检测上升沿或下降沿

输入捕捉定时器/计数器状态寄存器 (ECAP_STS)

寄存器	偏移量	R/W	描述	复位值
ECAP_STS	ECAP_BA+0x1C	R/W	输入捕捉状态寄存器	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					CAP2	CAP1	CAP0
7	6	5	4	3	2	1	0
Reserved		CAPOVF	CAPCMPF	Reserved	CAPTF2	CAPTF1	CAPTF0

位	描述	
[31:11]	Reserved	保留.
[10]	CAP2	输入捕捉引脚2状态 (只读) 输入捕捉引脚2(ECAP_P2)状态 (此位只读, 写无效)
[9]	CAP1	输入捕捉引脚1状态 (只读) 输入捕捉引脚1(ECAP_P1)状态 (此位只读, 写无效)
[8]	CAP0	输入捕捉引脚0状态 (只读) 输入捕捉引脚0(ECAP_P0)状态 (此位只读, 写无效)
[7:6]	Reserved	保留.
[5]	CAPOVF	输入捕捉计数器溢出标志 当计数器(ECAP_CNT)从0x00FF_FFFF 到 0溢出, 标志硬件置位。 0 = 上一次清除后没有溢出事件发生。 1 = 上一次清除后有溢出事件发生。 注意:此位写1清0
[4]	CAPCMPF	输入捕捉比较匹配溢出标志 如输入捕捉比较功能使能, 当捕捉计数器(ECAP_CNT)向上计数等于ECAP_CNTCMP值, 标志硬件置位。 0 = 上一次清除后没有发生ECAP_CNT等于ECAP_CNTCMP。 1 = 上一次清除后有发生ECAP_CNT等于ECAP_CNTCMP。 注意:此位写1清0

位	描述	
[3]	Reserved	保留.
[2]	CAPTF2	<p>输入捕捉通道2捕捉标志</p> <p>当输入捕捉通道2检测到在CAP2输入信号有有效的边沿变化，将置标志CAPTF2为1.</p> <p>0 = 上一次清除后CAP2输入信号没有有效的边沿变化被检测到</p> <p>1 = 上一次清除后CAP2输入信号有有效的边沿变化被检测到</p> <p>注意:此位写1清0</p>
[1]	CAPTF1	<p>输入捕捉通道1捕捉标志</p> <p>当输入捕捉通道1检测到在CAP1输入信号有有效的边沿变化，将置标志CAPTF1为1.</p> <p>0 = 上一次清除后CAP1输入信号没有有效的边沿变化被检测到</p> <p>1 = 上一次清除后CAP1输入信号有有效的边沿变化被检测到</p> <p>注意:此位写1清0</p>
[0]	CAPTF0	<p>输入捕捉通道0捕捉标志</p> <p>当输入捕捉通道0检测到在CAP0输入信号有有效的边沿变化，将置标志CAPTF0为1.</p> <p>0 = 上一次清除后CAP0输入信号没有有效的边沿变化被检测到</p> <p>1 = 上一次清除后CAP0输入信号有有效的边沿变化被检测到</p> <p>注意:此位写1清0</p>

6.8 增强型 PWM 发生器(EPWM)

6.8.1 概述

NM1120系列微控制器内置了针对马达驱动应用的一个PWM单元。PWM单元支持6路PWM发生器，可以配置为相互独立的6路PWM输出，PWM0~PWM5，或配置成3对分别带有可编程死区发生器的互补PWM(PWM0, PWM1)，(PWM2, PWM3) 和 (PWM4, PWM5)。

每一对互补PWM共用一个预分频器，分别提供9种分频系数(1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256)。每一路PWM输出有独立的16位计数器用以PWM周期控制，和16位的比较器用以调节占空比。6组发生器提供14个独立PWM中断标志，当相关PWM通道的周期和占空比如果与计数器相符，PWM中断将会被硬件置1。每一路PWM中断各有中断使能位。每个PWM发生器可以配置成单次模式来产生一个PWM信号周期或者自动重载模式来连续输出PWM波形。

为了防止PWM输出引脚输出不稳定波形，16位周期向上计数器和比较器带有双缓冲区。当用户向计数器和比较器寄存器写数据时，所写数据会在这个周期结束时加载。这种双缓存的特质保证了PWM稳定平滑的输出。

除了PWM，电机控制还需要Timer、ACMP和ADC共同工作。为了更精确的控制电机，我们提供一些寄存器，不仅可以配置PWM同时也可以配置Timer、ADC和ACMP。这样做，可以节省更多的CPU时间且便于控制电机，尤其是在直流无刷电机中。

6.8.2 特性

- 支持HCLK时钟输入和9级分频(1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256)
- 支持6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：
 - ◆ 6路独立PWM输出 - PWM0, PWM1, PWM2, PWM3, PWM4, 和 PWM5
 - ◆ 三组互补PWM输出，每组的一根引脚与另外一个引脚互补，且可以插入可编程的死区时间 - (PWM0, PWM1)，(PWM2, PWM3) 和 (PWM4, PWM5)
 - ◆ 三组同步PWM输出，每组PWM的引脚相位同步 - (PWM0, PWM1)，(PWM2, PWM3) 和 (PWM4, PWM5)
- 支持群组控制
- 支持单次模式（只支持边沿对齐类型）或者自动装载模式
- 支持16位精度
- 支持边沿对齐和中心对齐模式
- 支持互补PWM中可编程死区发生器
- 支持硬件故障刹车保护
 - ◆ 两个中断源类型：
 - ◆ 一种类型直接刹车，一种是能成刹车状态恢复。
 - ◆ 故障刹车源：
 - BRK0: ACMP0, ACMP1, EADC 和 外部引脚 (BRAKE).
 - BRK1: ACMP0, ACMP1, EADC 和外部引脚(BRAKE).

- PWM信号在极性控制阶段之前定义为正逻辑，PWM口的有效状态由极性控制寄存器来控制
- 支持独立的下降值CMPDAT比较，中心值CMPDAT比较（中心对齐模式），上升值CMPDAT比较（中心对齐模式），周期值比较，用来触发EADC转换。
- 支持ACMP输出事件触发PWM强制输出一个周期的低电平，该特性可用于步进电机驱动
- 支持中断累加功能

6.8.3 框图

图 6.8-1 表示EPWM时钟源。

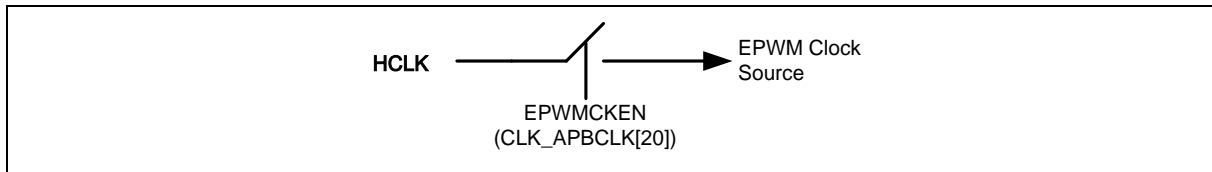


图 6.8-1 EPWM 时钟源

图 6.8-2表示EPWM模块全部功能。

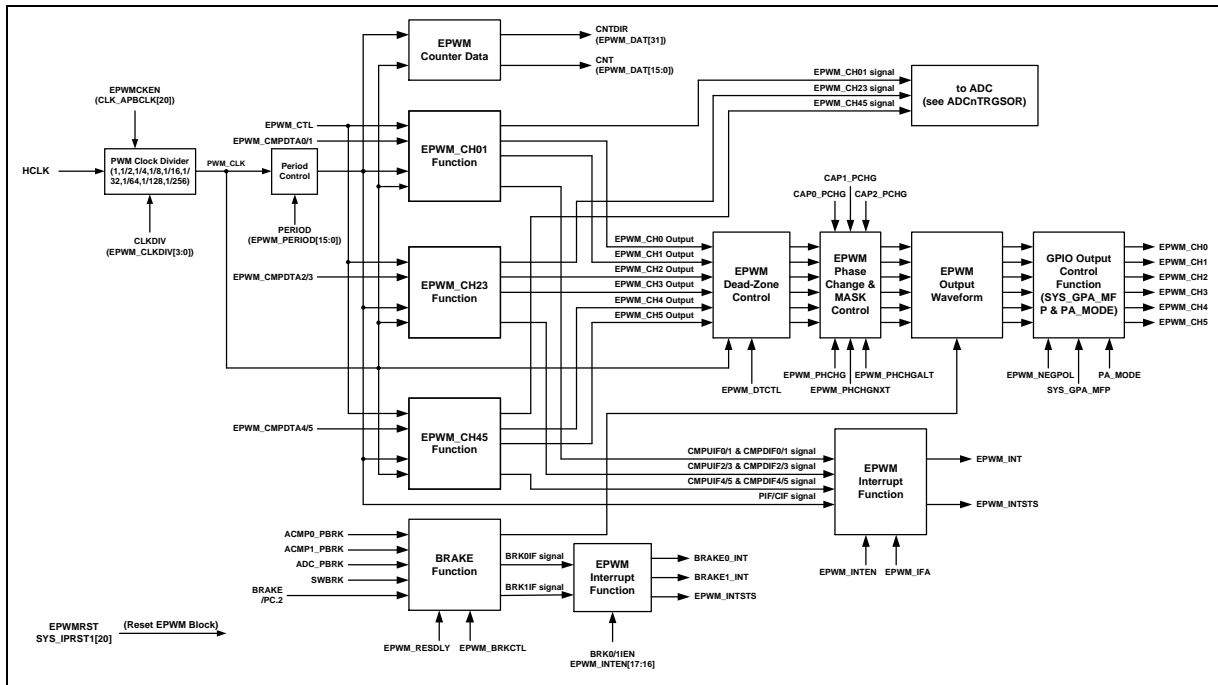


图 6.8-2 EPWM 方框图

图 6.8-3可以看出PWM的组架构，如通道0/1组成一组PWM，通道2/3是另一组和通道4/5为一组。

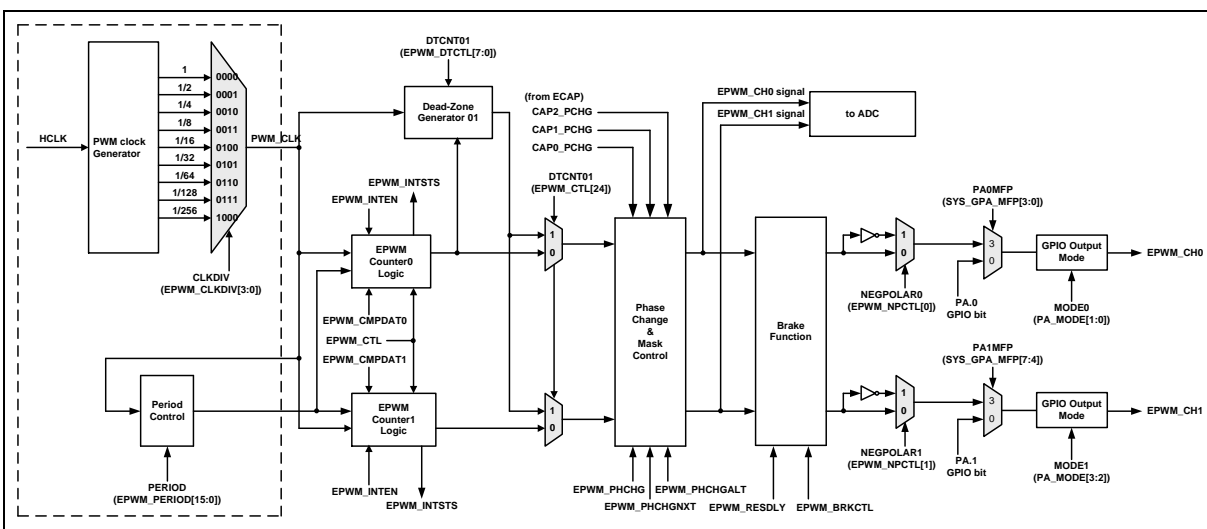


图 6.8-3 EPWM 发生器 CH0/1 架构图

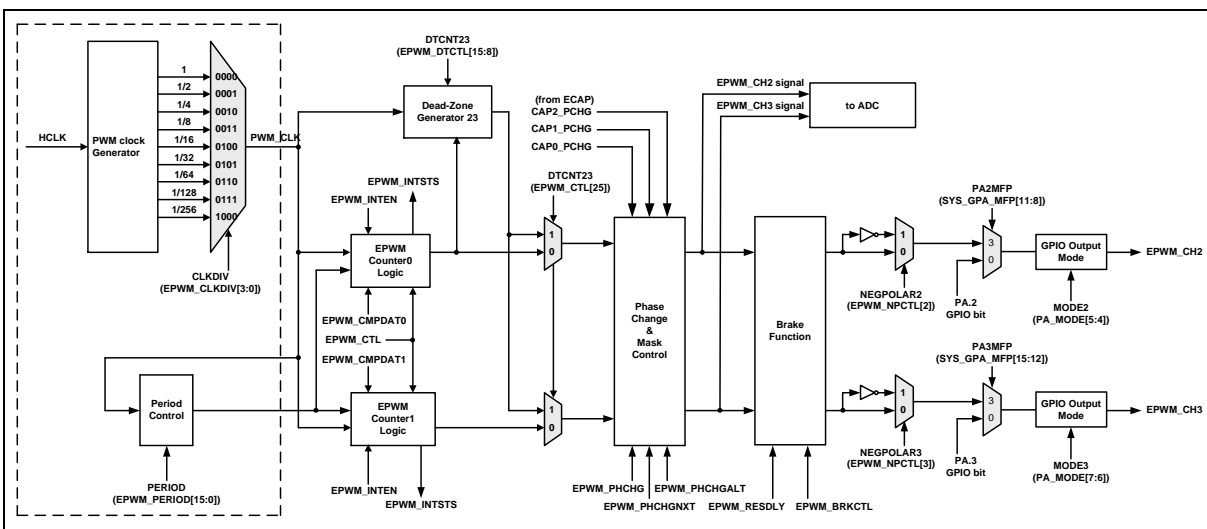


图 6.8-4 EPWM 发生器 CH2/3 架构图

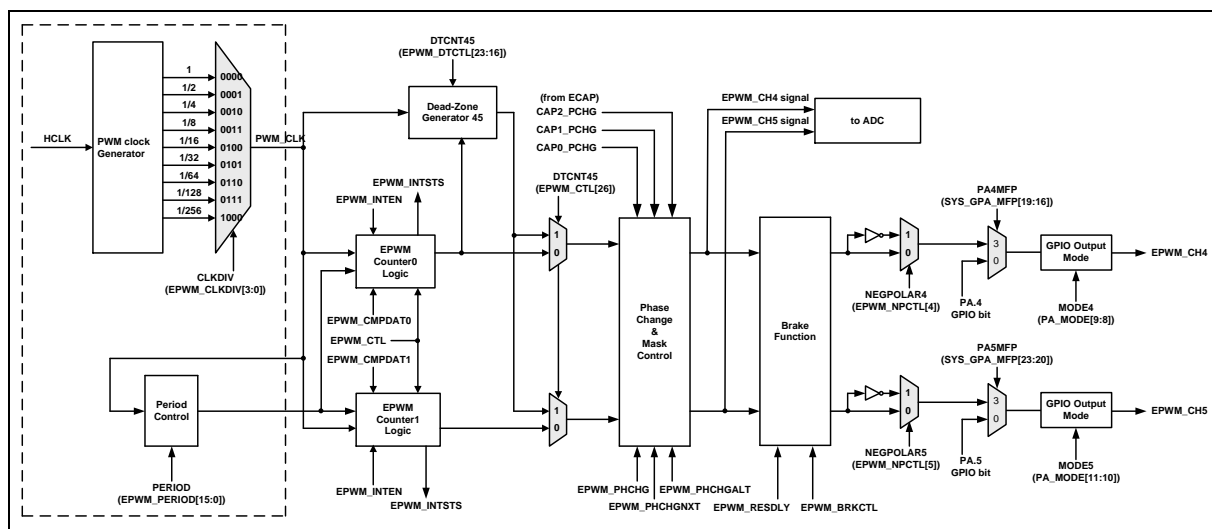


图 6.8-5 EPWM 发生器 CH4/5 架构图

6.8.4 基本配置

PWM的管脚功能配置在SYS_GPA_MFP 寄存器中

PWM时钟使能位为CLK_APBCLK[20]。PWM时钟源为HCLK

6.8.5 功能描述

6.8.5.1 PWM计数器工作模式

支持2种类型：边沿对齐，中心对齐

以下是每个PWM操作模式的周期和占空比的计算公式。

边沿对齐（向下计数）：

占空比 = $\text{CMPDAT} / (\text{PERIOD} + 1)$

高电平时间 = $\text{CMPDAT} * (\text{时钟周期})$

周期 = $(\text{PERIOD} + 1) * (\text{时钟周期})$

中心对齐（上下计数）：

占空比 = $2 * (\text{CMPDAT} / (\text{PERIOD} + 1))$

高电平时间 = $2 * \text{CMPDAT} * (\text{时钟周期})$

周期 = $2 * (\text{PERIOD} + 1) * (\text{时钟周期})$

边沿对齐PWM类型（向下计数）

在边沿对齐PWM输出模式时，16位PWM计数器会在每个周期开始向下计数，与占空比值CMPDATn（旧）

进行比较，当比较值相符就会翻转通道EPWM0_CHn输出为高电平。计数器会继续向下计数至0，这样会翻转EPWM0_CHn输出为低电平，当前CMPDATn（新）和 PERIOD（新）值在CNTMODEn=1情况下会更新，并响应PWM中断如果使能（EPWM_INTEN）。

图 6.8-6 为边沿对齐PWM时序和操作流程

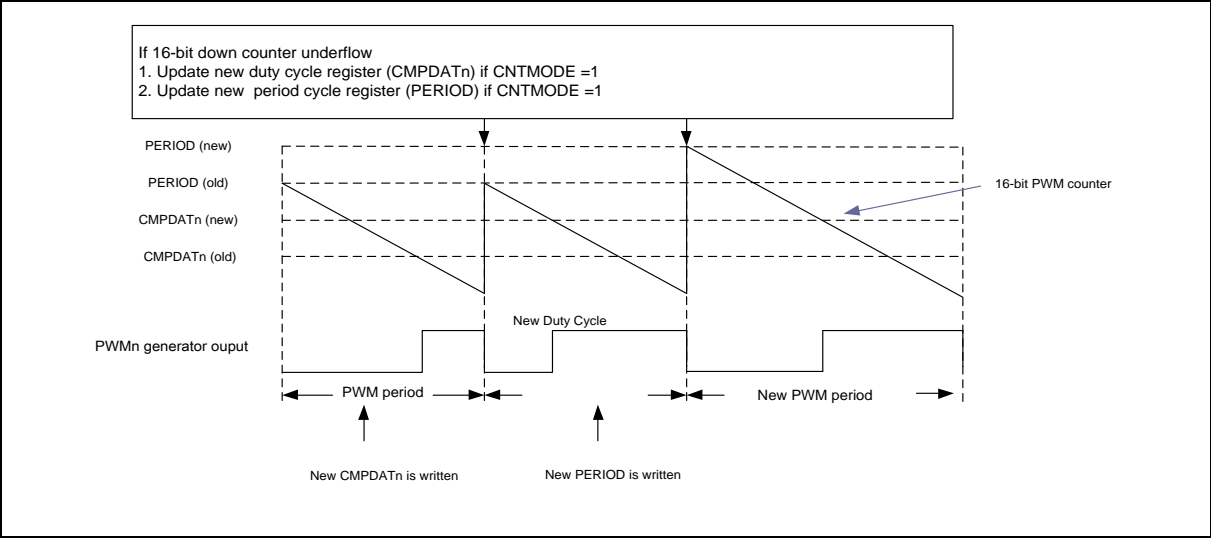


图 6.8-6 EPWM 边沿对齐类型

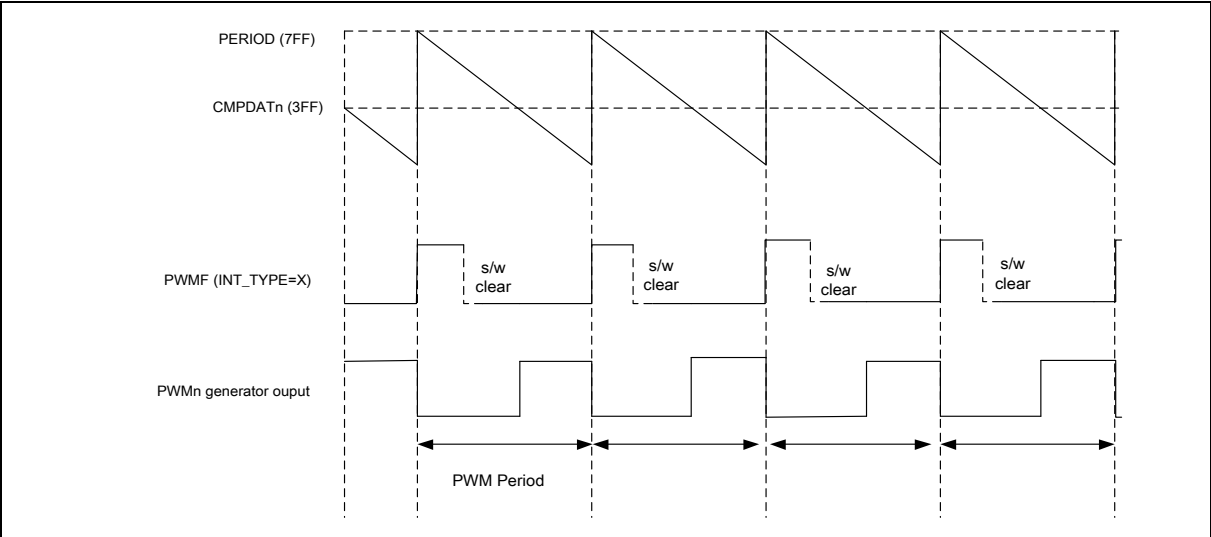


图 6.8-6 EPWM 边沿对齐波形输出

Edge Aligned mode :
 Period = (PERIOD + 1) * clock cycle
 Duty ratio = CMPDAT / (PERIOD + 1)
 CMRDAT > PERIOD --> always High
 CMRDAT = 0 --> always Low

Edge Aligned mode
 PERIOD = 99
 Period = PERIOD + 1 = 99 + 1 = 100

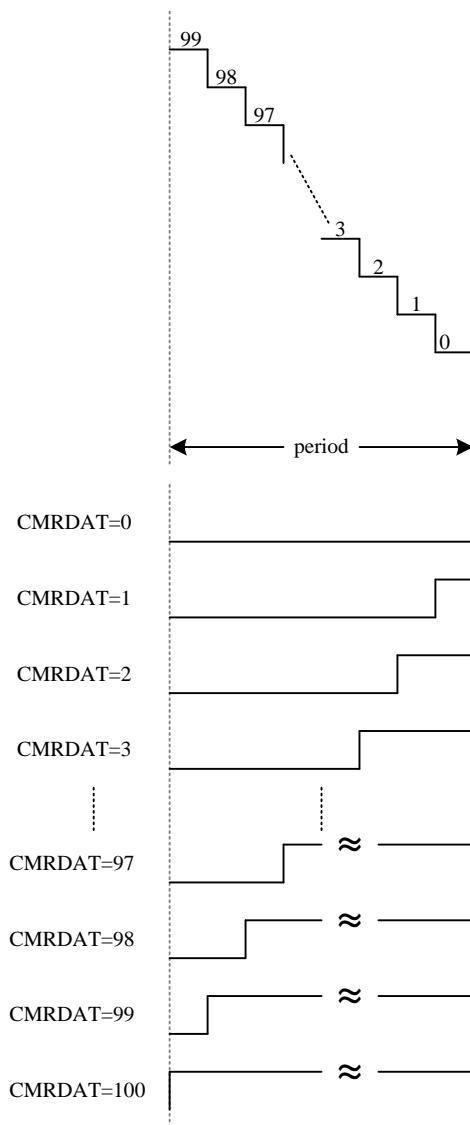


图 6.8-7 EPWM 边沿对齐模式操作时序

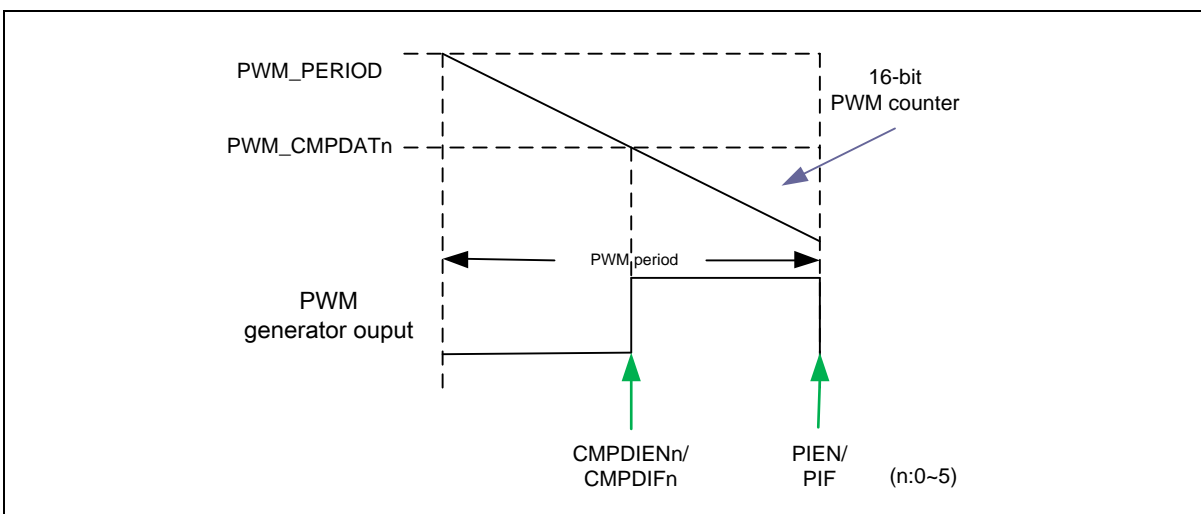


图 6.8-8 EPWM 边沿对齐中断图

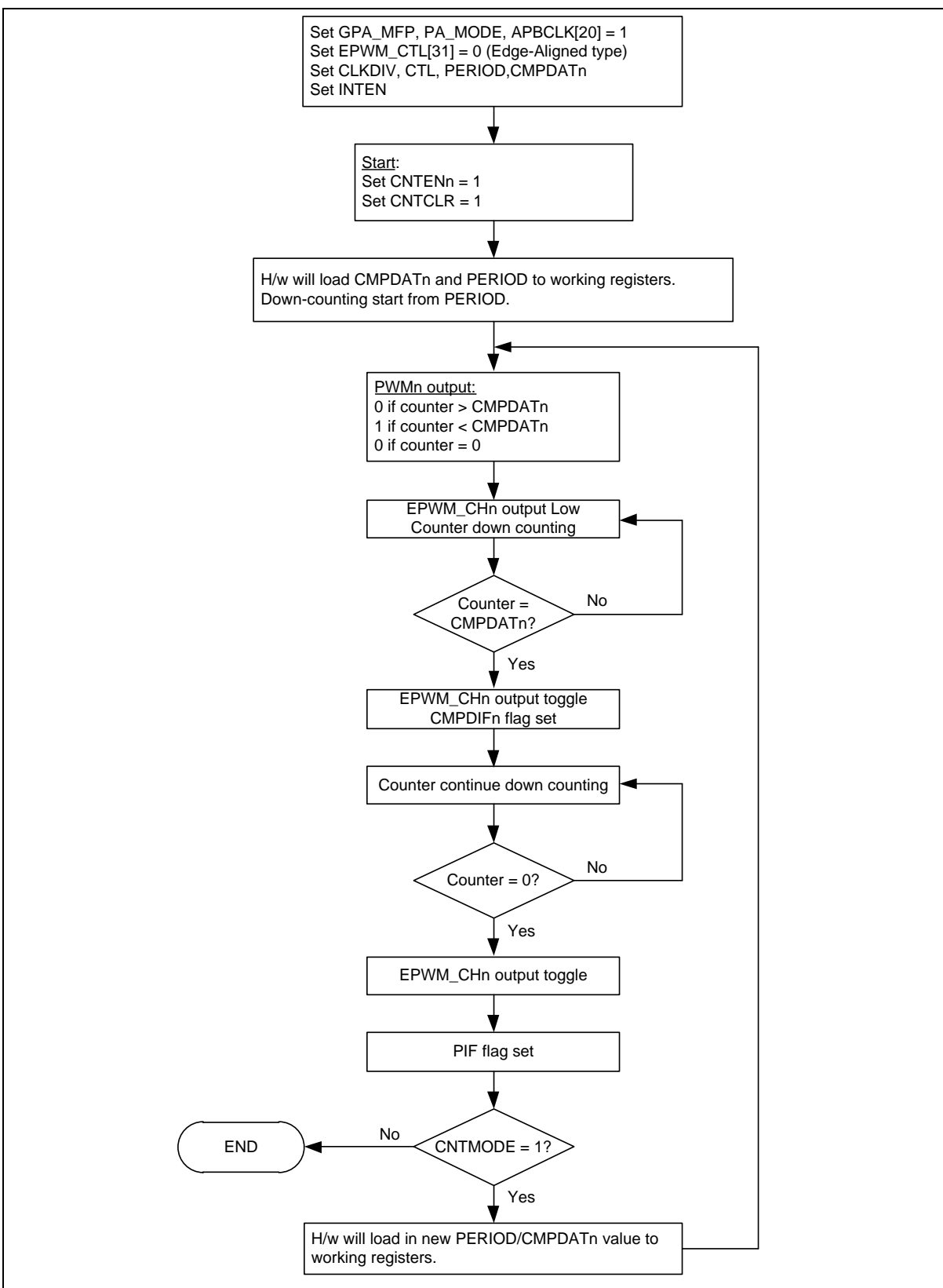


图 6.8-9 EPWM 边沿对齐流程图

EPWM的周期和占空比由向下计数器寄存器PERIOD 和PWM比较寄存器CMPDATn决定。PWM定时器工作时序见图 6.8-11 EPWM-Timer。脉冲宽度调制遵循以下公式，PWM比较器工作原理见图 6.8-10 EPWM 比较器输出说明。注意相关的GPIO脚需要配置成PWM功能。

PWM 频率 = $HCLK / (\text{时钟分频}) / (PERIOD+1)$

周期 = $(PERIOD + 1) \text{ unit}$

占空比 = $CMPDAT / (PERIOD+1)$

$CMPDAT > PERIOD$: PWM 输出总为高

$CMPDAT \leq PERIOD$: PWM 输出高占空比 = $(CMPDAT) \text{ unit}$

$CMPDAT = 0$: PWM 总为低

注意: 1. Unit = 一个 PWM 时钟周期.

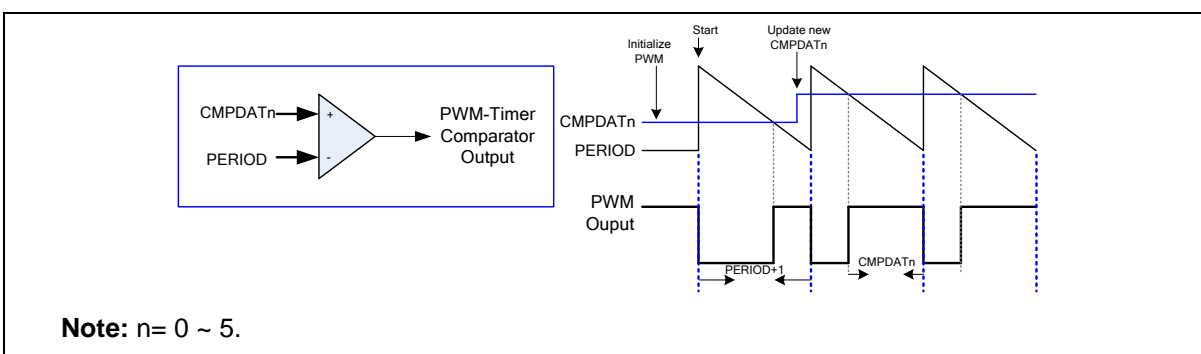


图 6.8-10 EPWM 比较器输出说明

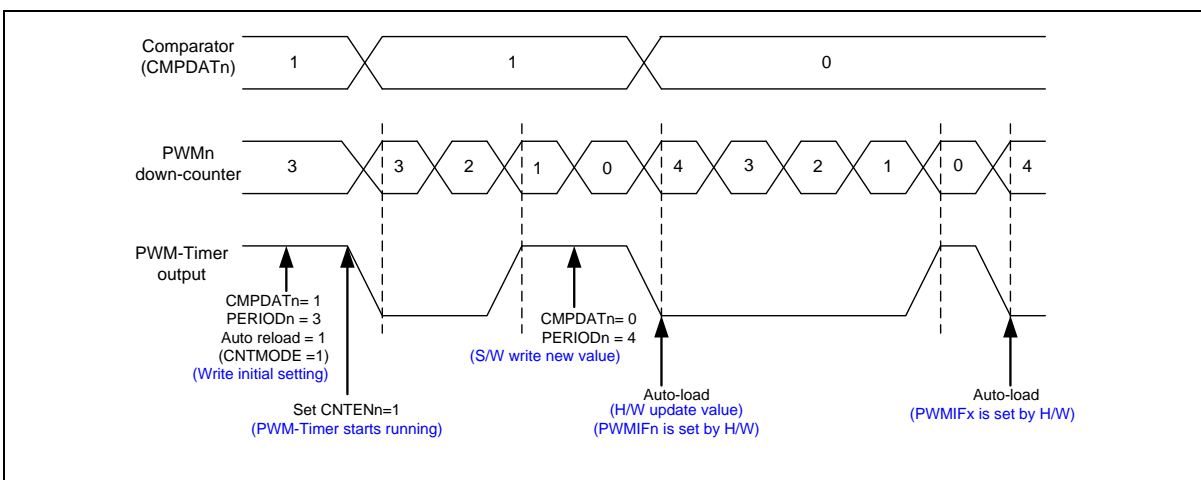


图 6.8-11 EPWM-Timer 工作时序

PWM中心对齐模式 (上/下 计数器)

中心对齐模式，PWM的上/下计数模式时间参数配置好以后将会产生相应的PWM波形。当PWM首先是向下计数模式，计数器的值从PERIOD周期值每次减一计到CMPDATn (旧)，这样会翻转EPWM_CHn产生输出为高。计数器继续计数到0，到达这个状态计数器自动配置为向上计数，当EPWM计数器又一次计数到CMPDATn (旧)，PWMn输出翻转位低。一旦EPWM计数器溢出，在CNTMODE = 1，将更新EPWM周期寄存器PERIOD (新)和占空比周期寄存器CMPDATn (新)。

中心对齐模式，EPWM有4种类型中断，周期中断 (PIF)，向上匹配中断 (CMPUIF)，中心点中断 (CIF)，和向下匹配中断 (CMPDIF)。

PWM 频率 = $HCLK / (\text{时钟分频}) / (2 * (\text{PERIOD} + 1))$

周期 = $2 \times (\text{PERIOD} + 1) \text{ unit}$

占空比 = $2 \times (\text{CMPDAT} / (\text{PERIOD} + 1))$

$\text{CMPDAT} > \text{PERIOD}$: PWM 输出总为高

$\text{CMPDAT} \leq \text{PERIOD}$: PWM 输出高占空比 = $(2 \times \text{CMPDAT}) \text{ unit}$

$\text{CMPDAT} = 0$: PWM 总为低

注意: 1. Unit = 一个 PWM时钟周期

图 6.8-12 为中心对齐模式PWM时序和操作流程

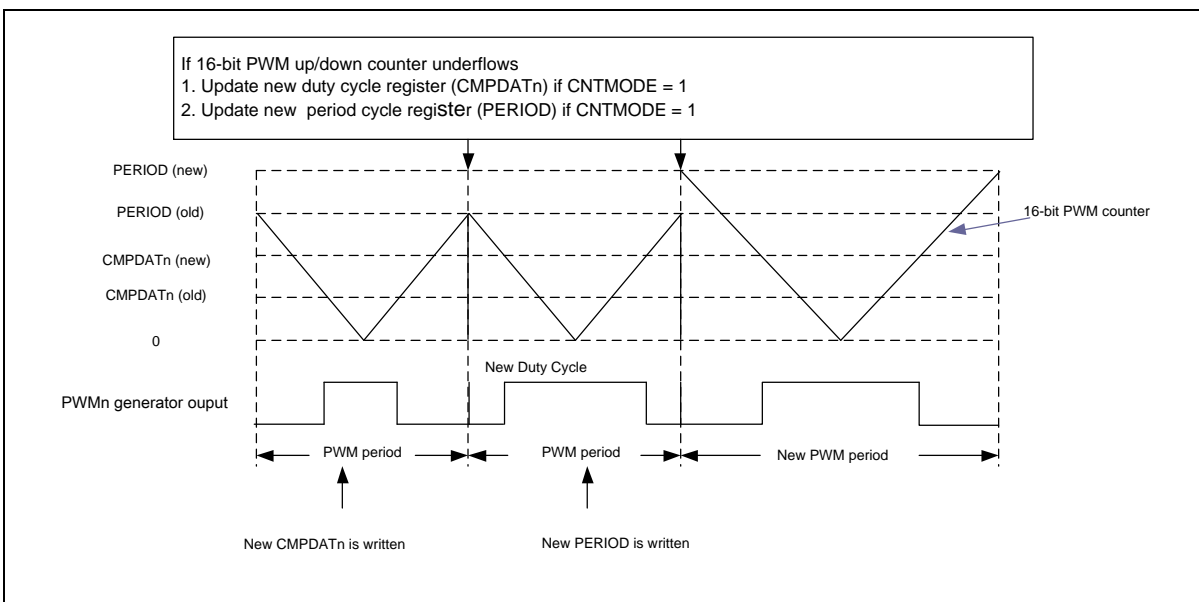


图 6.8-12 EPWM 中心对齐模式

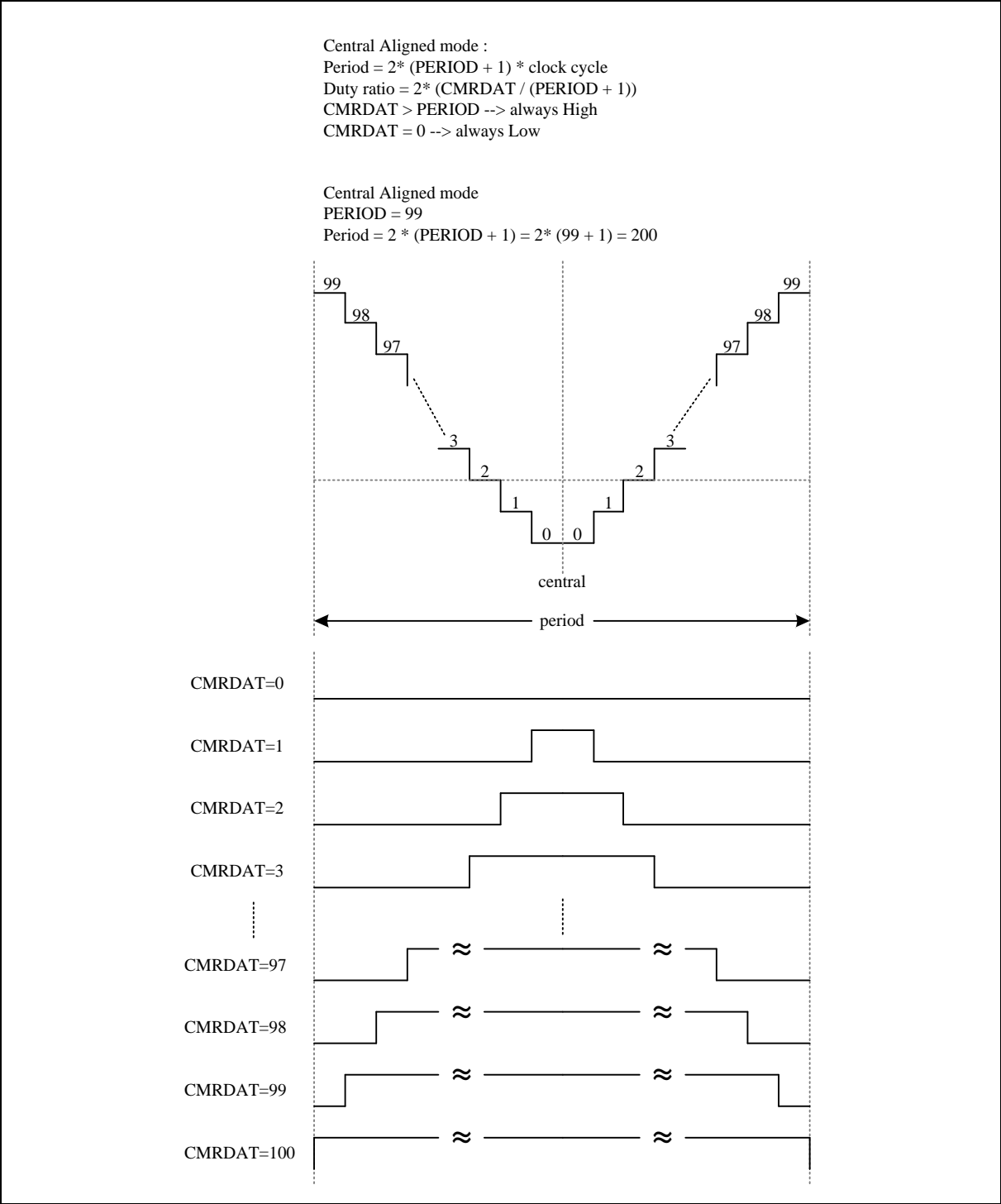


图 6.8-13 EPWM 中心对齐模式操作时序

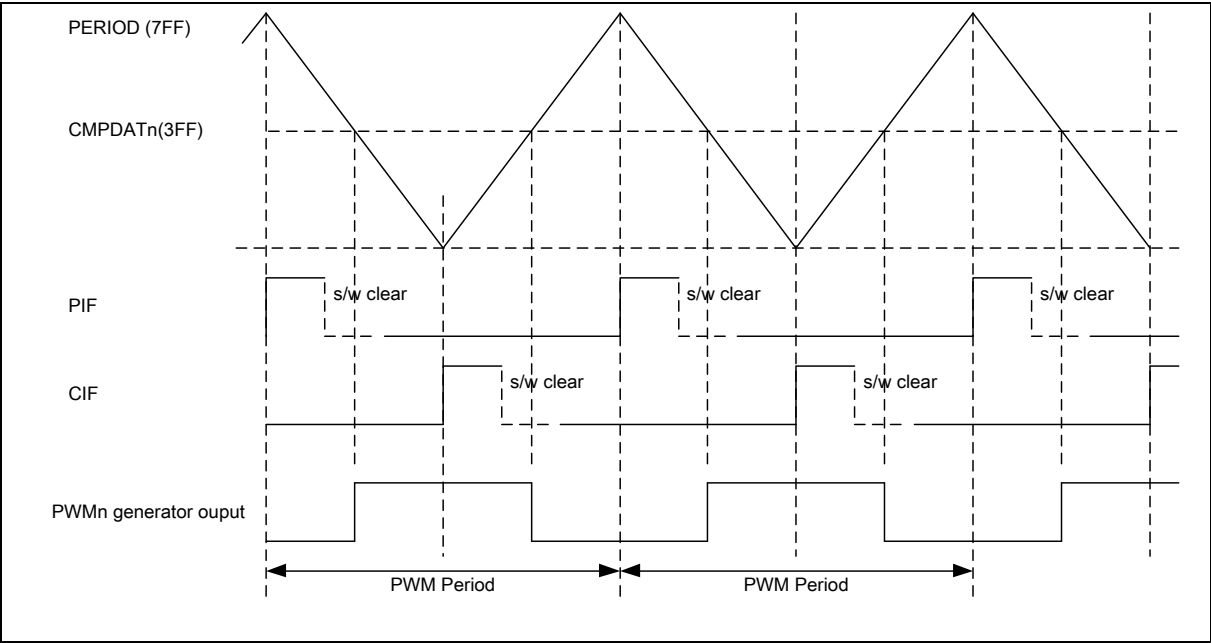


图 6.8-14 EPWM 中心对齐模式波形输出

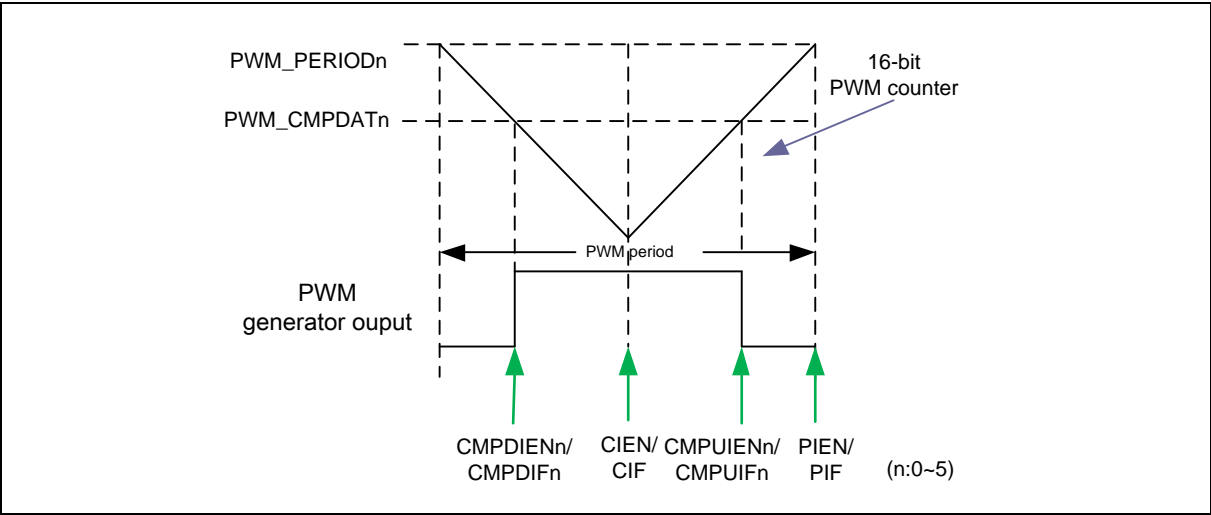


图 6.8-15 EPWM 中心对齐模式中断图

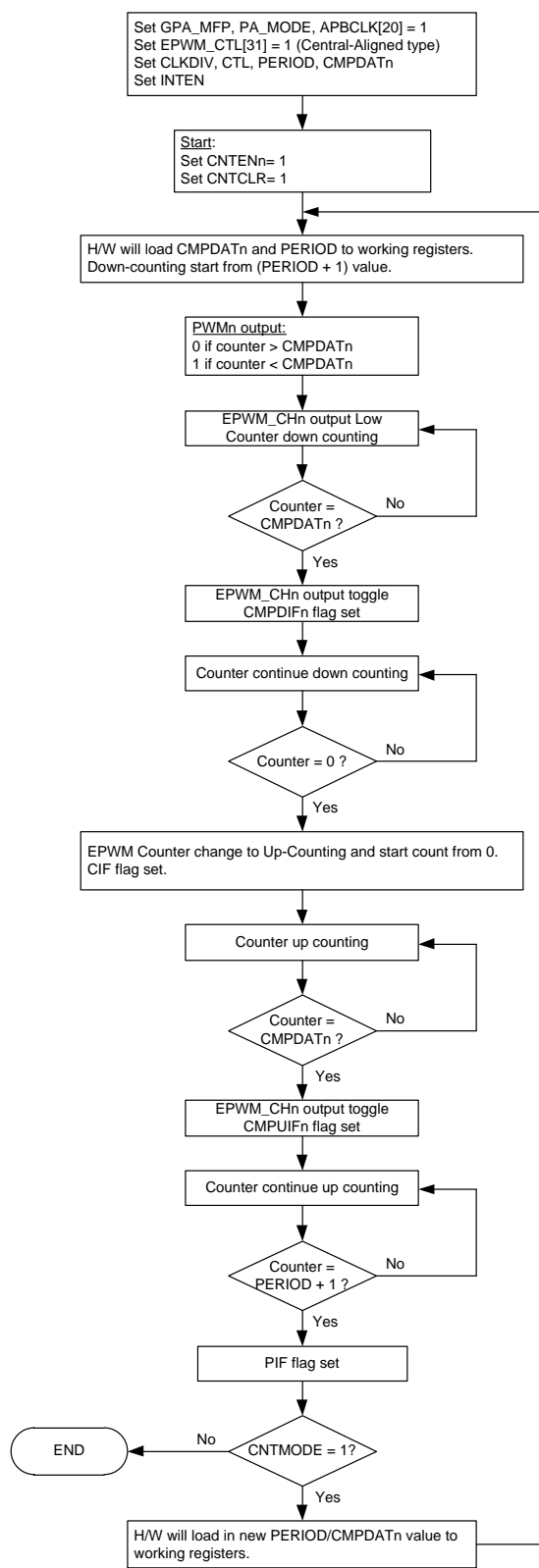


图 6.8-16 EPWM 中心对齐模式流程图

6.8.5.2 EPWM 端口输出控制

EPWM 有6个输出引脚，为PA.0~PA.5。

如图 6.8-10所示，在任何复位动作后，通过设置PA_MODE寄存器，PWM输出端口能初始化为三态类型或其它类型。

6.8.5.3 独立模式

当MODE (EPWM_CTL[29:28]) = 00的时候，EPWM设置为独立模式，有6个PWM通道输出，每个通道有自己的占空比周期。

6.8.5.4 互补模式

MODE (EPWM_CTL[29:28]) = 01时启用互补模式。

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EPWM0_CHn与相应的偶数PWM通道互补，如EPWM0_CH0与EPWM0_CH1互补，EPWM0_CH3与EPWM0_CH2互补，EPWM0_CH5与EPWM0_CH4互补。

6.8.5.5 同步模式

MODE (EPWM_CTL[29:28]) = 10时启用互补模式。

这种模式下，通过设置有3个PWM通道周期，EPWM_CH0，EPWM_CH2，和 EPWM_CH4。6个输出通道根据奇偶分组。在同步模式下，PWM奇数通道(1/3/5)信号一直同步相应的偶数通道(0/2/4)信号。

6.8.5.6 插入死区时间

死区发生器插入一段“关”周期叫死区时间，发生在一组互补的PWM中，把一个管脚的波形从关闭状态到相应互补的另一个引脚打开状态之间的偏移时间。这样能对外接的电源切换设备起到保护作用。一对PWM输出通道有一个8位向下计数器产生死区时间。死区输出维持到计数器向下计数到0。

死区时间按照以下函数计算：

死区时间 = PWM_CLK * (DTCNTnm+1)。这里 nm，可以是01，23，45

图 6.8-17示一对通道打开死区时间功能时序图。

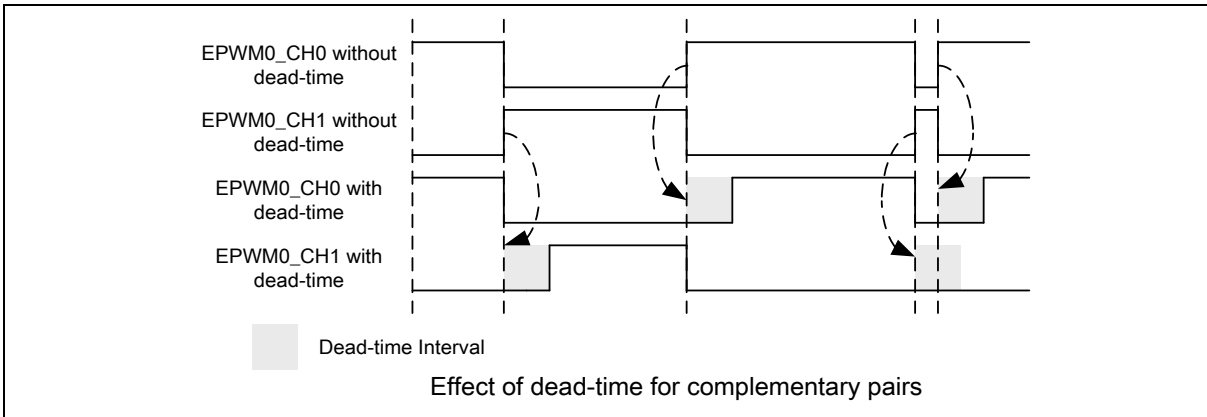


图 6.8-17 EPWM 插入死区时间

6.8.5.7 群组模式

GROUPEN (EPWM_CTL[30]) = 1时，使能组模式

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EPWM0_CH0占空比寄存器控制。

如果 GROUPEN = 1，那么 (EPWM0_CH2, EPWM0_CH3) 和 (EPWM0_CH4, EPWM0_CH5) 对会跟随 (EPWM0_CH0, EPWM0_CH1)，意味着：

$EPWM0_CH4 = EPWM0_CH2 = EPWM0_CH0$;

当 MODE (EPWM_CTL[29:28]) = 01，如果互补模式被使能的话， $EPWM0_CH5 = EPWM0_CH3 = EPWM0_CH1 = (EPWM0_CH0)$ 的反向波形。

注意：对应用来说，不要同时用组模式和同步模式，这样同步模式实现不了。

6.8.5.8 不对称模式

不对称模式只在中心对齐模式中起作用。ASYMEN (EPWM_CTL[20]) = 1时，不对称模式使能。在这个模式下，当向上计数时EPWM计数器将与另外一个比较值CMPU (EPWM_CMPDATn[31:16])比较。如果CMPU不等于CMR，EPWM会产生不对称波形并把相应通道n标志CMPUIFn (EPWM_INTSTS[13:8])置位。

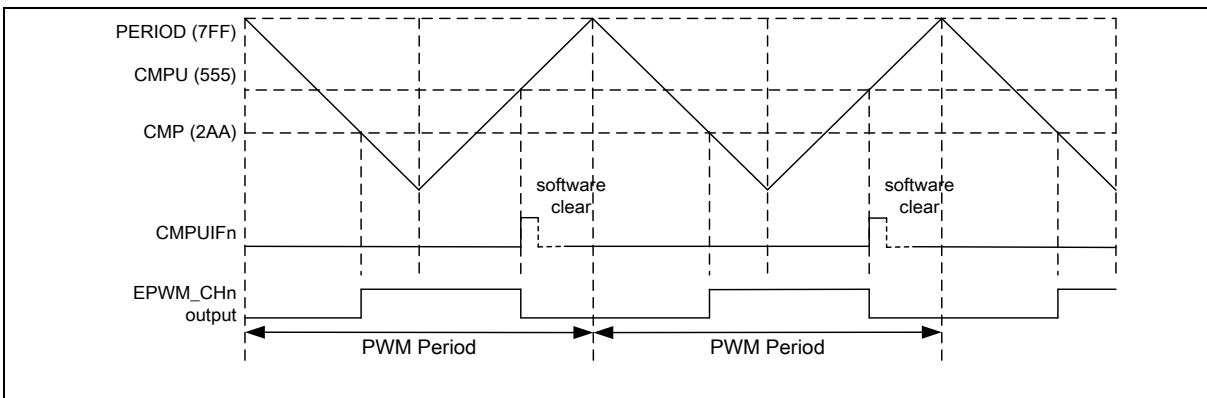


图 6.8-18 EPWM 不对称模式时序图

6.8.5.9 One-Shot 模式

当CNTMODE (EPWM_CTL[8]) = 0时，EPWM设置为one-shot模式。EPWM运行这个设置将输出一个脉冲周期。图 6.8-19为one-shot模式状态。

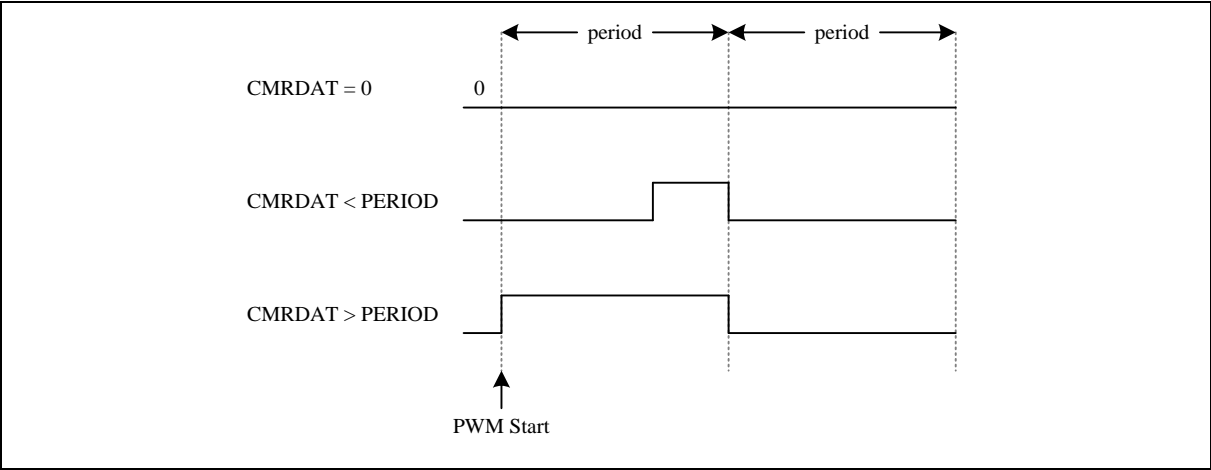


图 6.8-19 EPWM One-Shot 模式架构

6.8.5.10 极性控制

每个PWM通道从PWM0_CH0 到 PWM0_CH5都有独立的极性控制，默认极性是输出高电平。

图 6.8-20可以看出不同极性的PWM初始化状态。

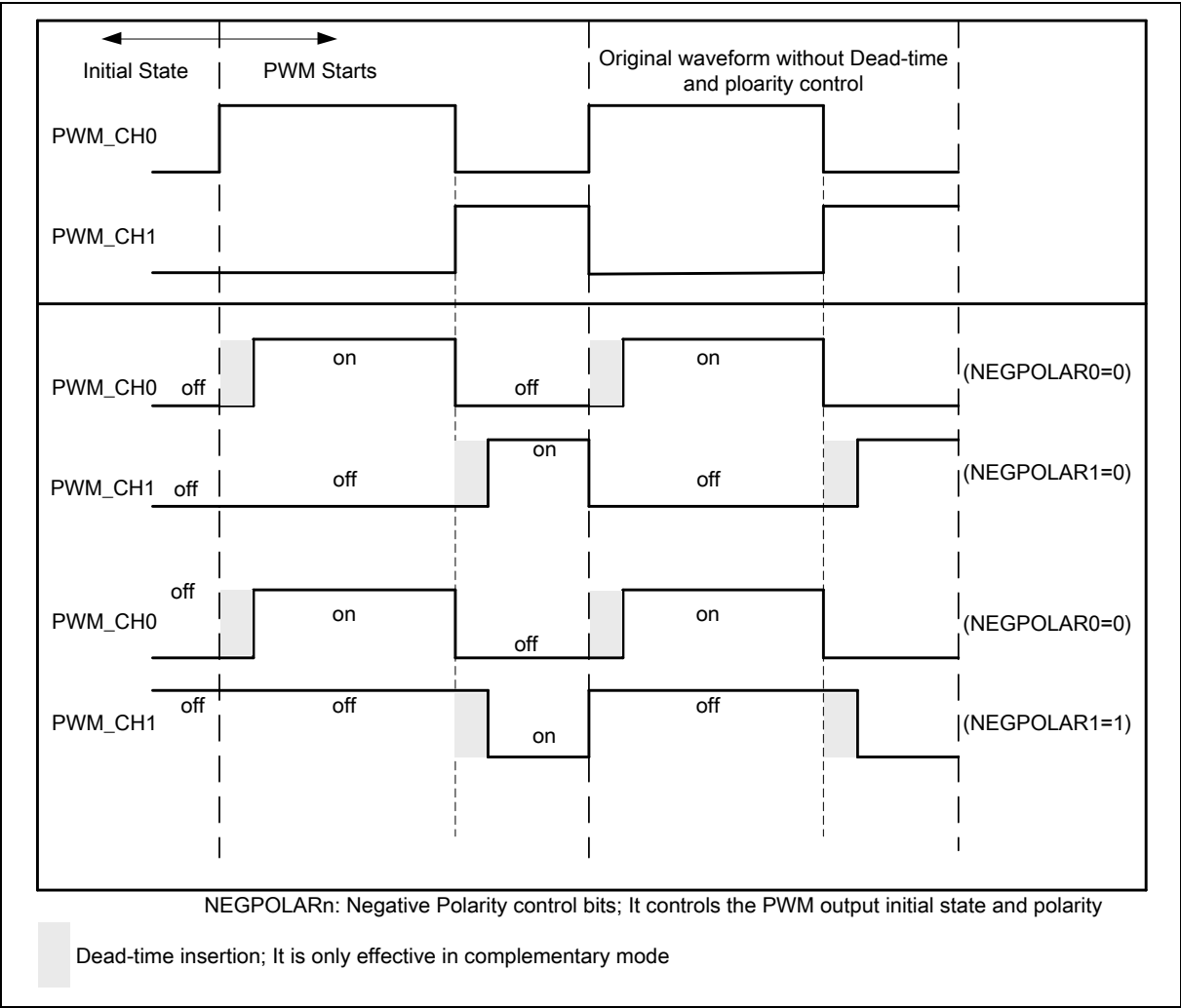


图 6.8-20 上升沿死区 EPWM 极性控制初始化状态

6.8.5.11 中断架构

EPWM单元有6个中断源，PIF (EPWM_INTSTS[0])PWM计数器计数到周期点时产生的中断标志。CIF (EPWM_INTSTS[18])PWM计数器在中心对齐模式计数到中心点的中断标志。CMPDIFn (EPWM_INTSTS[29:24])PWM计数器向下计数到CMPn (EPWM_CMPDATn[15:0])的中断标志。CMPUIFn (EPWM_INTSTS[13:8])PWM计数器向上计数到CMPUn (EPWM_CMPDATn[31:16])的中断标志，如果操作在非对称模式，这是在向上计数时计数到CMPUn (PWM_CMPDATn[31:16])的标志。BRKIF0 (PWM_INTSTS[16])Brake0中断标志, BRKIF1 (PWM_INTSTS[17])Brake1中断标志 (BRKIF1)。

BRK0IEN (EPWM_INTEN[16]) 和 BRK1IEN (EPWM_INTEN[17]) 位控制刹车中断使能。PIEN (EPWM_INTEN[0]) 控制 PIF 中断使能。CIEN (EPWM_INTEN[18]) 控制 CIF 中断使能。CMPUIENn (EPWM_INTEN[13:8]) 控制 CMPUIFn 中断使能。CMPDIENn (EPWM_INTEN[29:24]) 控制 CMPDIFn 中断使能。注意所有的中断标志都是硬件置位，必须软件清除。

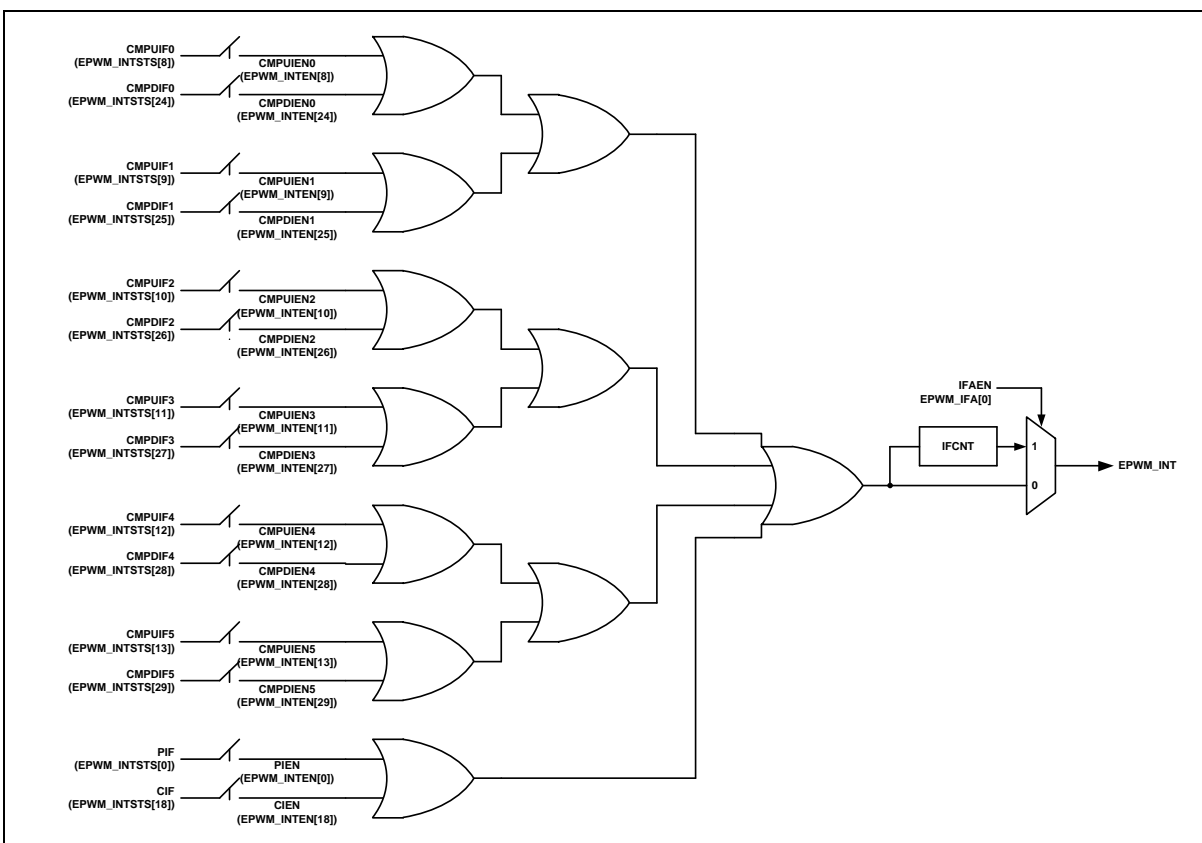


图 6.8-21 EPWM 中断架构

注意: BRKnIF的中断架构, 看 图 6.8-22.

6.8.5.12 EPWM 刹车

支持两路刹车BRK0和BRK1检测，每路有4个刹车信号。一个外部刹车引脚（BRAKE连接到BRK0和BRK1都可），2个模拟比较器输出和一个ADC输出。外部的刹车引脚有数字滤波器。刹车功能的控制通过设置寄存器EPWM_BRKCTL。

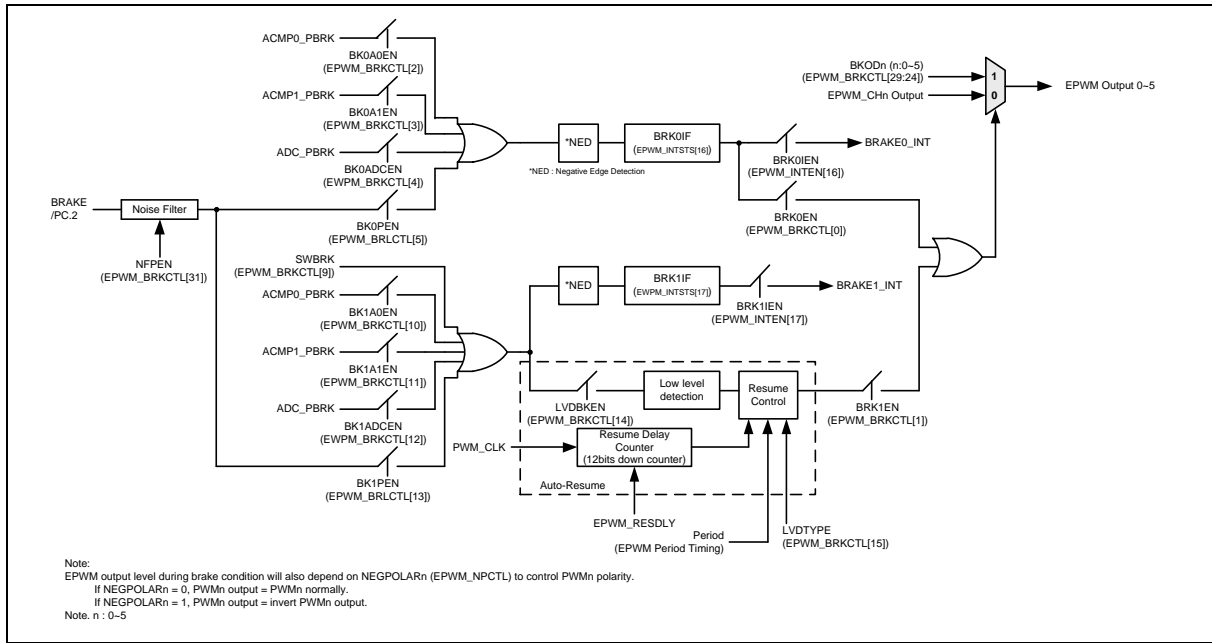


图 6.8-22 EPWM 刹车架构

BRK0刹车将保持EPWM输出为刹车定义的值（通过设置EPWM_BRKCTL[29:24]），需要再次初始化EPWM功能来解除EPWM刹车信号。

BRK1有刹车恢复功能，当BRK1刹车激活，它可以恢复通过自己的12位延时计数器。

由于两个刹车条件被实现，都将自动导致BRKnIF (n:0,1)标志被置位。用户可以轮询这些刹车标志位或者使能EPWM的刹车中断(EPWM_INTEN)来实现哪个条件导致刹车事件发生。

6.8.5.13 EPWM 相位换相功能

换相功能可以通过用定时器和ACMP触发PWM模块，通过选择TRGSEL (EPWM_PHCHG[22:20]) 寄存器。使用定时器(或ACMP)触发EPWM, 通过配置EPWM_PHCHG和EPWM_PHCHGNXT寄存器。每次超时事件到来时，EPWM_PHCHG的值会被EPWM_PHCHG_NXT的值自动更新，EPWM_PHCHG的位域也是由EPWM_PHCHGNXT的位域定义。每一次EPWM_PHCHG更新，相关功能也将改变。

除了触发EPWM，换相寄存器有相关的屏蔽控制位去改变PWM输出相位。将MSKENn (EPWM_PHCHG[13:8]) 相关寄存器位置1去使能通道屏蔽功能，相关通道会输出MSKDATn (EPWM_PHCHG[5:0])值。

1.1.1.1.1 EPWM 换相屏蔽输出功能

换相功能，每个EPWM输出能手动屏蔽输出通过设定EPWM屏蔽使能功能（MSKENn, n:0~5）和EPWM屏蔽数据寄存器(MSKDATn, n:0~5)，并依靠独立的占空比周期比较器单元，来驱动EPWM引脚输出特定的逻辑电平。MSKENn寄存器包括6个位，MSKEN[5:0] (EPWM_PHCHG[13:8] / EPWM_PHCHGNXT[13:8])决定哪路PWM i/o引脚将屏蔽输出。复位MSKENn为00H。MSKDATn寄存器包含6个位，MSKDAT[5:0] (EPWM_PHCHG[5:0] / EPWM_PHCHGNXT[5:0])决定当特定的

输出通过MSKDATn位屏蔽，PWM i/o引脚输出状态。复位MSKDATn是00H。当MSKEN[5:0]位被设置，相应的MSKDAT[5:0]位将对PWM通道有效。

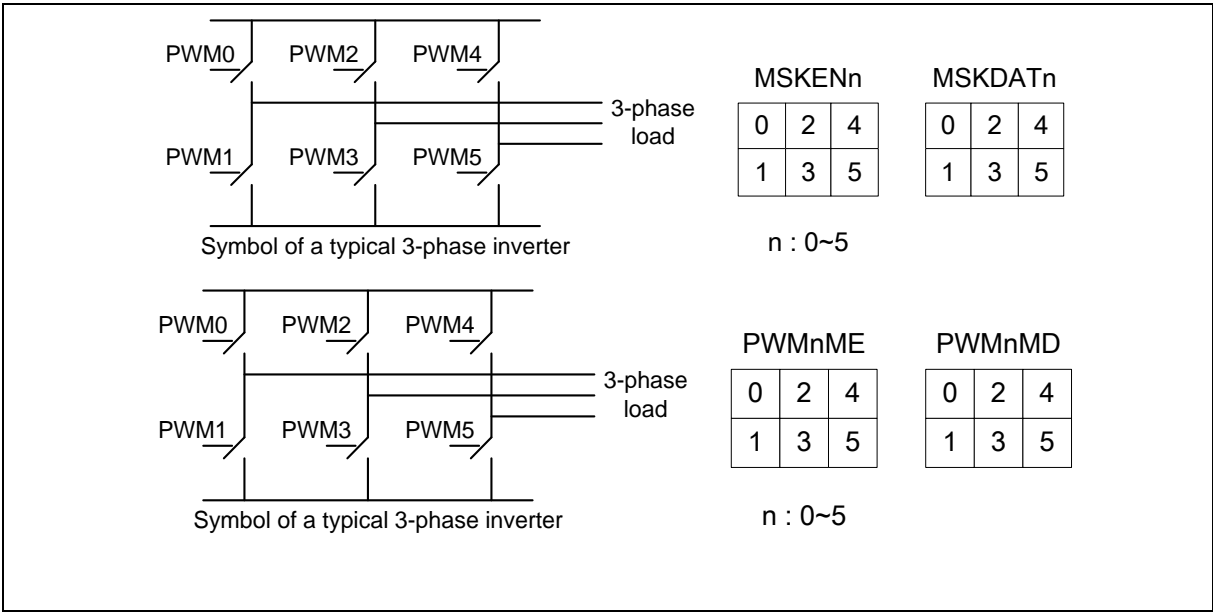


图 6.8-23 EPWM 3-相位监测屏蔽图

如例 1,马达激活路径是路径0连接路径3，路径0用EPWM0，路径3为开（短路）。

- PWM 通道 0 输出为PWM 发生器信号
- PWM 通道 1-5 被屏蔽通过 MSKENn
- PWM通道1-5 输出状态被 MSKDATn 位决定

开关 0 (开/关)	被EPWM0控制(EPWM0 频率/占空比发生器)
开关1 (关)	MSKDAT1 = 0
开关2 (关)	MSKDAT2 = 0
开关3 (开)	MSKDAT3 = 1
开关4 (关)	MSKDAT4 = 0
开关5 (关)	MSKDAT5 = 0

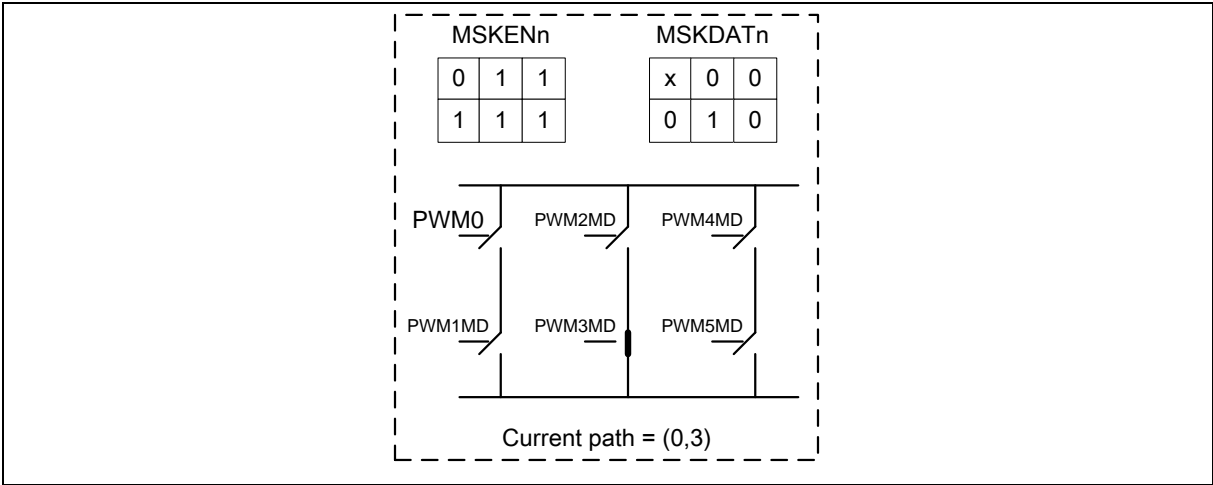


图 6.8-24 EPWM 3-相位马达屏蔽例 1

如例 2， 马达激活路径是路径2连接路径5， 路径2用EPWM2， 路径5为开（短路）。

PWM通道 2 输出为PWM 发生器信号

PWM通道0-1, 3-5被屏蔽通过 MSKENn

PWM通道0-1, 3-5输出状态被 MSKDATn 位决定

开关0 (关)	MSKDAT0 = 0
开关1 (关)	MSKDAT1 = 0
开关2 (开/关)	被EPWM2控制(EPWM2 频率/占空比发生器)
开关3 (关)	MSKDAT3 = 0
开关4 (关)	MSKDAT4 = 0
开关5 (开)	MSKDAT5 = 1

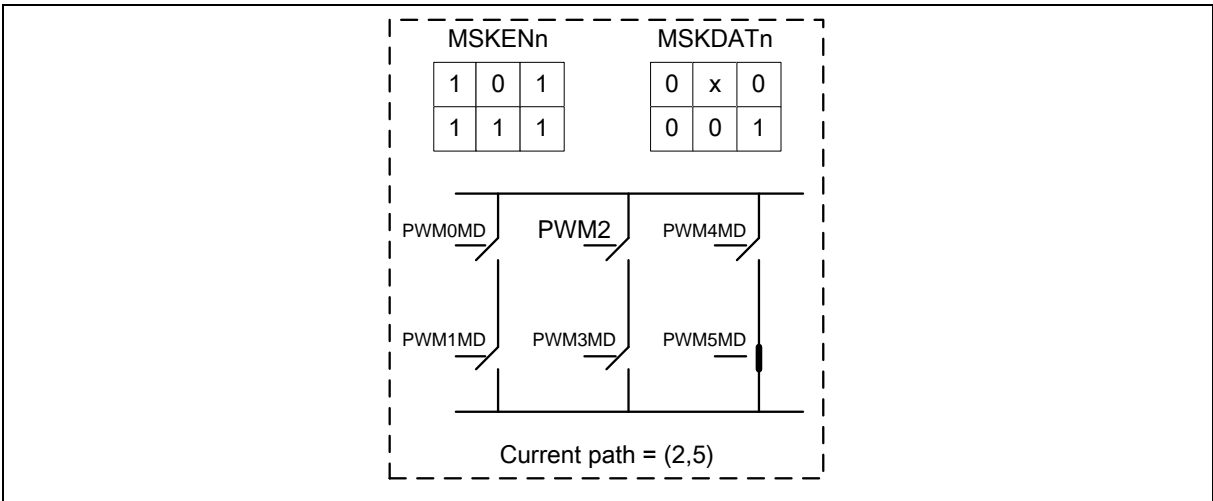


图 6.8-25 EPWM 3-相位马达屏蔽例 2

1. 1. 1. 1. 2EPWM 相位换相霍尔传感器模式

EPWM能直接检测霍尔传感器状态来改变马达相位。当TRGSEL = 011b (EPWM_PHCHGNXT[22:20]) 来通过下个霍尔状态触发，换相控制器将检测CAPn_PCHG (n:0, 2) 状态。如果匹配HALLSTS (EPWM_PHCHGNXT[18:16]) 设置的值，那么EPWM输出数据将从EPWM_PHCHGNEXT拷贝到EPWM_PHCHG，同时改变马达的状态。

6.8.6 寄存器映射

R: 只读, W: 只写, R/W: 读/写, C: 仅能写入0值

寄存器	偏移地址	R/W	描述	复位值
EPWM 基地址: EPWM_BA = 0x4004_0000				
EPWM_NPCTL	EPWM_BA+0x00	R/W	EPWM 负极性控制寄存器	0x0000_0000
EPWM_CLKDIV	EPWM_BA+0x04	R/W	EPWM 时钟选择寄存器	0x0000_0000
EPWM_CTL	EPWM_BA+0x08	R/W	EPWM 控制寄存器	0x0000_0000
EPWM_PERIOD	EPWM_BA+0x0C	R/W	EPWM 周期计数器寄存器	0x0000_0000
EPWM_CMPDAT0	EPWM_BA+0x24	R/W	EPWM 比较寄存器0	0x0000_0000
EPWM_CMPDAT1	EPWM_BA+0x28	R/W	EPWM 比较寄存器1	0x0000_0000
EPWM_CMPDAT2	EPWM_BA+0x2C	R/W	EPWM 比较寄存器2	0x0000_0000
EPWM_CMPDAT3	EPWM_BA+0x30	R/W	EPWM 比较寄存器 3	0x0000_0000
EPWM_CMPDAT4	EPWM_BA+0x34	R/W	EPWM 比较寄存器4	0x0000_0000
EPWM_CMPDAT5	EPWM_BA+0x38	R/W	EPWM比较寄存器5	0x0000_0000
EPWM_CNT	EPWM_BA+0x3C	R	EPWM 数据寄存器	0x0000_0000
EPWM_INTEN	EPWM_BA+0x54	R/W	EPWM中断使能寄存器	0x0000_0000
EPWM_INTSTS	EPWM_BA+0x58	R/W	EPWM 中断状态寄存器	0x0000_0000
EPWM_RESPLY	EPWM_BA+0x5C	R/W	EPWM 刹车低电平检测恢复延时寄存器	0x0000_0000
EPWM_BRKCTL	EPWM_BA+0x60	R/W	EPWM 故障刹车控制寄存器	0x0000_0000
EPWM_DTCTL	EPWM_BA+0x64	R/W	EPWM 死区控制寄存器	0x0000_0000
EPWM_PHCHG	EPWM_BA+0x78	R/W	EPWM 相位换相寄存器	0x0000_0000
EPWM_PHCHGNXT	EPWM_BA+0x7C	R/W	EPWM下一相位修改寄存器	0x0000_0000
EPWM_PHCHGALT	EPWM_BA+0x80	R/W	EPWM 相位换相屏蔽控制寄存器	0x0000_0000
EPWM_IFA	EPWM_BA+0x84	R/W	EPWM 周期中断累加控制寄存器	0x0000_00F0

6.8.7 寄存器描述

EPWM负极性控制寄存器(EPWM_NPCTL)

寄存器	偏移地址	R/W	描述	复位值
EPWM_NPCTL	EPWM_BA+0x00	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		NEGPOLAR5	NEGPOLAR4	NEGPOLAR3	NEGPOLAR2	NEGPOLAR1	NEGPOLAR0

位	描述	
[31:6]	Reserved	保留.
[5]	NEGPOLAR5	PWM5 负极性控制 此位控制PWM输出的极性 有效状态 0 = PWM 输出高有效 1 = PWM输出低有效
[4]	NEGPOLAR4	PWM4 负极性控制 此位控制PWM输出的极性有效状态 0 = PWM 输出高有效 1 = PWM输出低有效
[3]	NEGPOLAR3	PWM3 负极性控制 此位控制PWM输出的极性 有效状态 0 = PWM 输出高有效 1 = PWM输出低有效
[2]	NEGPOLAR2	PWM2 负极性控制 此位控制PWM输出的极性 有效状态 0 = PWM 输出高有效 1 = PWM输出低有效

位	描述	
[1]	NEGPOLAR1	PWM1 负极性控制 此位控制PWM输出的极性有效状态 0 = PWM 输出高有效 1 = PWM输出低有效
[0]	NEGPOLAR0	PWM0 负极性控制 此位控制PWM输出的极性 有效状态 0 = PWM 输出高有效 1 = PWM输出低有效

EPWM 时钟选择寄存器 (EPWM_CLKDIV)

寄存器	偏移地址	R/W	描述	复位值
EPWM_CLKDIV	EPWM_BA+0x04	R/W	EPWM 时钟选择寄存器	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				CLKDIV			

位	描述	
[31:4]	Reserved	保留.
[3:0]	CLKDIV	EPWM 时钟分频器 (9 级分频) 选择PWM时钟输入分频。 0000 = 1 (HCLK / 2 ⁰). 0001 = 1/2 (HCLK / 2 ¹). 0010 = 1/4 (HCLK / 2 ²). 0011 = 1/8 (HCLK / 2 ³). 0100 = 1/16 (HCLK / 2 ⁴). 0101 = 1/32 (HCLK / 2 ⁵). 0110 = 1/64 (HCLK / 2 ⁶). 0111 = 1/128 (HCLK / 2 ⁷). 1000 = 1/256 (HCLK / 2 ⁸). 1001~ 1111 = 保留.

EPWM 控制寄存器 (EPWM_CTL)

寄存器	偏移地址	R/W	描述	复位值
EPWM_CTL	EPWM_BA+0x08	R/W	EPWM 控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
CNTTYPE	GROUPEN	MODE		CNTCLR	DTCNT45	DTCNT23	DTCNT01
23	22	21	20	19	18	17	16
DBGTRIOFF	Reserved		ASYMEN	Reserved		HCUPDT	
15	14	13	12	11	10	9	8
Reserved							CNTMODE
7	6	5	4	3	2	1	0
Reserved		CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0

位	描述	
[31]	CNTTYPE	PWM 对齐类型选择 0 = 边沿对齐类型 1 = 中心对齐类型
[30]	GROUPEN	群组控制位 0 = 所有PWM通道信号时序时独立的。 1 = 统一PWM0, PWM2 and PWM4信号时序,并由PWM0控制。
[29:28]	MODE	PWM 工作模式选择位 00 = 独立模式 01 = 互补模式 10 = 保留。 11 = 保留。
[27]	CNTCLR	清除 PWM 计数器控制位 0 = 不清PWM计数器 1 = 所有 16-bit PWM 计数器清 0x0000。 注意: 该位有硬件自动清除
[26]	DTCNT45	死区4 发生器使能位 (PWM4 和 PWM5 为 PWM 组的一对) 0 = 死区4 发生器禁止。 1 = 死区4 发生器使能。 注意: 当死区发生器被使能, PWM4 和 PWM5 对变成PWM 组的互补对
[25]	DTCNT23	死区2 发生器使能位 (PWM2 和 PWM3 为 PWM 组的一对) 0 = 死区2 发生器禁止。 1 = 死区2发生器使能。 注意: 当死区发生器被使能, PWM2 和 PWM3 对变成PWM 组的互补对

位	描述	
[24]	DTCNT01	<p>死区0发生器使能位 (PWM1 和 PWM1 为 PWM 组的一对)</p> <p>0 = 死区0 发生器禁止.</p> <p>1 = 死区0 发生器使能.</p> <p>注意: 当死区发生器被使能, PWM0和 PWM1 对变成PWM 组的互补对</p>
[23]	DBGTRIOFF	<p>PWM仿真模式配置位 (仅仿真模式有效)</p> <p>0 = 安全模式: 定时器停止, PWM输出关闭, 反向器处于安全状态。计数器可以重新从停止处开始计数器。</p> <p>1 = 正常模式: 在某些特殊情况下, 定时器继续正常运行可能非常危险, 因为有一个固定占空比的波形输出到反向器 (没有中断服务程序)</p>
[22:21]	Reserved	保留.
[20]	ASYMEN	<p>中心对齐类型的非对称模式</p> <p>0 = 中心对齐类型的对称模式</p> <p>1 = 中心对齐类型的非对称模式</p>
[19:18]	Reserved	保留.
[17:16]	HCUPDT	<p>中心对齐模式的半周期更新使能位</p> <p>00= 更新 PERIOD & CMP 在 pwm_counter = PERIOD (周期).</p> <p>01 =更新PERIOD & CMP在pwm_counter = 0.</p> <p>10 =更新PERIOD & CMP在半周期 (counter = 0 & PERIOD, 都更新).</p> <p>11 =更新PERIOD & CMP在pwm_counter = PERIOD (周期).</p>
[15:9]	Reserved	保留.
[8]	CNTMODE	<p>PWM自动装载/One-Shot 模式</p> <p>0 = One-shot 模式.</p> <p>1 = 自动装载模式</p>
[7:6]	Reserved	保留.
[5]	CNTEN5	<p>PWM 5使能开始运行</p> <p>0 = 相应的 PWM停止.</p> <p>1 = 相应的 PWM开始运行.</p>
[4]	CNTEN4	<p>PWM 4使能开始运行</p> <p>0 = 相应的 PWM停止.</p> <p>1 = 相应的 PWM开始运行.</p>
[3]	CNTEN3	<p>PWM 3使能开始运行</p> <p>0 = 相应的 PWM停止.</p> <p>1 = 相应的 PWM开始运行.</p>
[2]	CNTEN2	<p>PWM 2使能开始运行</p> <p>0 = 相应的 PWM停止.</p> <p>1 = 相应的 PWM开始运行.</p>

位	描述	
[1]	CNTEN1	PWM 1使能开始运行 0 = 相应的 PWM停止. 1 = 相应的 PWM开始运行.
[0]	CNTEN0	PWM 0使能开始运行 0 = 相应的 PWM停止. 1 = 相应的 PWM开始运行.

EPWM 周期计数器寄存器 (EPWM_PERIOD)

寄存器	偏移地址	R/W	描述	复位值
EPWM_PERIOD	EPWM_BA+0x0C	R/W	EPWM 周期计数器寄存器	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
PERIOD							
7	6	5	4	3	2	1	0
PERIOD							

位	描述	
[31:16]	Reserved	保留.
[15:0]	PERIOD	<p>PWM 计数器/定时器装载值</p> <p>PERIODn 决定PWM周期</p> <p>边沿对齐模式: 这里xy, 根据所选择PWM通道可能是 01, 23, 45 。</p> <p>PWM 频率 = $PWM_{xy_CLK}[HCLK]/\text{时钟分频}[EPWM_CLKDIV]$.</p> <p>PWM 时钟周期 = $1 / \text{PWM频率}$</p> <p>周期 = PWM 时钟周期 * (PERIOD+1).</p> <p>高占空比 = PWM 时钟周期 * CMPn.</p> <p>占空比 = $CMPn / (PERIOD+1)$.</p> <p>CMPn >= PERIOD: PWM输出总为高</p> <p>CMPn < PERIOD: PWM low 宽度 = (PERIODn-CMPn) unit; PWM high宽度= (CMP) unit.</p> <p>CMPn = 0: PWM 输出总为低</p> <p>中心对齐模式: 这里xy, 根据所选择PWM通道可能是 01, 23, 45 。</p> <p>周期 = $1 / (PWM_{xy_CLK}[HCLK]/\text{时钟分频}[EPWM_CLKDIV] / (2 * (PERIOD+1)))$.</p> <p>占空比 = $2 * (CMPn / (PERIOD+1))$.</p> <p>CMPn >= PERIOD: PWM输出总为高</p> <p>CMPn < PERIOD: PWM low宽度= (PERIOD - CMPn) x 2 unit; PWM high宽度= CMP x 2 unit.</p> <p>CMPn = 0: PWM输出总为低</p> <p>(Unit = 一个 PWM 时钟周期).</p> <p>注意: PERIODn的任何写入都将在下个PWM时钟有效。</p>

EPWM 比较寄存器 0-5 (EPWM_CMPDAT0-5)

寄存器	偏移地址	R/W	描述	复位值
EPWM_CMPDAT0	EPWM_BA+0x24	R/W	EPWM 比较寄存器 0	0x0000_0000
EPWM_CMPDAT1	EPWM_BA+0x28	R/W	EPWM比较寄存器1	0x0000_0000
EPWM_CMPDAT2	EPWM_BA+0x2C	R/W	EPWM比较寄存器2	0x0000_0000
EPWM_CMPDAT3	EPWM_BA+0x30	R/W	EPWM比较寄存器3	0x0000_0000
EPWM_CMPDAT4	EPWM_BA+0x34	R/W	EPWM比较寄存器4	0x0000_0000
EPWM_CMPDAT5	EPWM_BA+0x38	R/W	EPWM比较寄存器5	0x0000_0000

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

位	描述	
[31:16]	CMPU	<p>在中心对齐非对称模式时，向上计数时的PWM比较寄存器</p> <p>CMPU > PERIOD: @ 向上计数 PWM 输出为高电平</p> <p>CMPU <= PERIOD: (CMPUn + CMPn) unit.</p> <p>CMP <= PERIOD: PWM 输出高占空比= (CMP + CMPU) unit.</p> <p>其它: PWM输出总为低</p> <p>(Unit = 一个PWM时钟周期).</p>

位	描述	
[15:0]	CMP	<p>PWM 比较寄存器</p> <p>CMP 决定PWM的占空比.</p> <p>边沿对齐模式： 这里xy，根据所选择PWM通道可能是 01, 23, 45 。</p> <p>周期 = (PERIOD + 1) unit.</p> <p>占空比 = CMP / (PERIOD+1).</p> <p>CMP > PERIOD: PWM输出总为高</p> <p>CMP <= PERIOD: PWM输出高占空比= (CMP) unit.</p> <p>CMP = 0: PWM输出总为低</p> <p>中心对齐模式： 这里xy，根据所选择PWM通道可能是 01, 23, 45 。</p> <p>周期 = 2 x (PERIOD + 1) unit.</p> <p>占空比 = 2 x (CMP / PERIOD).</p> <p>CMP > PERIOD: PWM 输出总为高</p> <p>CMP <= PERIOD: PWM 输出高占空比 = (2 x CMP) unit.</p> <p>CMP = 0: PWM 输出总为低.</p> <p>(Unit = 一个PWM时钟周期.)</p> <p>注意： 任何写CMPn 值都将在下一个PWM周期有效</p>

EPWM 数据寄存器 (EPWM_CNT)

寄存器	偏移地址	R/W	描述	复位值
EPWM_CNT	EPWM_BA+0x3C	R	EPWM数据寄存器	0x0000_0000

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

位	描述	
[31]	CNTDIR	PWM 计数器（向上/向下）方向标志 0 = PWM 计数器时向下计数. 1 = PWM计数器时向上计数.
[30:16]	Reserved	保留.
[15:0]	CNT	PWM 数据 用户能读取CNT来知道16位计数器的当前值

EPWM 中断使能寄存器 (EPWM_INTEN)

寄存器	偏移地址	R/W	描述	复位值
EPWM_INTEN	EPWM_BA+0x54	R/W	EPWM 中断使能寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CMPDIEN5	CMPDIEN4	CMPDIEN3	CMPDIEN2	CMPDIEN1	CMPDIEN0
23	22	21	20	19	18	17	16
Reserved					CIEN	BRK1IEN	BRK0IEN
15	14	13	12	11	10	9	8
Reserved		CMPUIEN5	CMPUIEN4	CMPUIEN3	CMPUIEN2	CMPUIEN1	CMPUIEN0
7	6	5	4	3	2	1	0
Reserved							PIEN

位	描述	
[31:30]	Reserved	保留。
[29]	CMPDIEN5	PWM通道5向下中断使能位 边沿对齐和中心对齐向下计数 0 =比较中断禁止 1 =比较中断使能。当EPWM_CH5向下计数等于EPWM_CMPDAT5，进入中断
[28]	CMPDIEN4	PWM通道4向下中断使能位 边沿对齐和中心对齐向下计数 0 =比较中断禁止 1 =比较中断使能。当EPWM_CH4向下计数等于EPWM_CMPDAT4，进入中断
[27]	CMPDIEN3	PWM通道3向下中断使能位 边沿对齐和中心对齐向下计数 0 =比较中断禁止 1 =比较中断使能。当EPWM_CH3向下计数等于EPWM_CMPDAT3，进入中断
[26]	CMPDIEN2	PWM通道2向下中断使能位 边沿对齐和中心对齐向下计数 0 =比较中断禁止 1 =比较中断使能。当EPWM_CH2向下计数等于EPWM_CMPDAT2，进入中断
[25]	CMPDIEN1	PWM通道1向下中断使能位 边沿对齐和中心对齐向下计数 0 =比较中断禁止 1 =比较中断使能。当EPWM_CH1向下计数等于EPWM_CMPDAT1，进入中断

位	描述	
[24]	CMPDIEN0	PWM通道0向下中断使能位 边沿对齐和中心对齐向下计数 0 = 比较中断禁止 1 = 比较中断使能。当EPWM_CH0向下计数等于EPWM_CMPDAT0，进入中断
[23:19]	Reserved	保留。
[18]	CIEN	PWM中心点中断使能位 仅中心对齐模式 0 = 中心点中断禁止 1 = 中心点中断使能。
[17]	BRK1IEN	故障刹车1中断使能位 0 = BRK1IF 触发PWM中断禁止。 1 = BRK1IF 触发PWM中断使能。
[16]	BRK0IEN	故障刹车0中断使能位 0 = BRK0IF 触发PWM中断禁止。 1 = BRK0IF 触发PWM中断使能。
[15:14]	Reserved	保留。
[13]	CMPUIEN5	PWM通道5向上中断使能位 仅中心对齐模式 0 = PWM通道5向上计数比较中断禁止 1 = 比较中断使能。当EPWM_CH5向上计数等于EPWM_CMPDAT5，进入中断
[12]	CMPUIEN4	PWM通道4向上中断使能位 仅中心对齐模式 0 = PWM通道4向上计数比较中断禁止 1 = 比较中断使能。当EPWM_CH4向上计数等于EPWM_CMPDAT4，进入中断
[11]	CMPUIEN3	PWM通道3向上中断使能位 仅中心对齐模式 0 = PWM通道3向上计数比较中断禁止 1 = 比较中断使能。当EPWM_CH3向上计数等于EPWM_CMPDAT3，进入中断
[10]	CMPUIEN2	PWM通道2向上中断使能位 仅中心对齐模式 0 = PWM通道2向上计数比较中断禁止 1 = 比较中断使能。当EPWM_CH2向上计数等于EPWM_CMPDAT2，进入中断
[9]	CMPUIEN1	PWM通道1向上中断使能位 仅中心对齐模式 0 = PWM通道1向上计数比较中断禁止 1 = 比较中断使能。当EPWM_CH1向上计数等于EPWM_CMPDAT1，进入中断

位	描述	
[8]	CMPUIEN0	PWM通道0向上中断使能位 仅中心对齐模式 0 =PWM通道0向上计数比较中断禁止 1 =比较中断使能。当EPWM_CH0向上计数等于EPWM_CMPDAT0，进入中断
[7:1]	Reserved	保留。
[0]	PIEN	PWM通道0周期中断使能位 边沿对齐模式和中心对齐模式都适用 0 = EPWM 周期中断禁止。 1 = EPWM 周期中断使能。

EPWM 中断状态寄存器 (EPWM_INTSTS)

寄存器	偏移地址	R/W	描述	复位值
EPWM_INTSTS	EPWM_BA+0x58	R/W	EPWM中断状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CMPDIF5	CMPDIF4	CMPDIF3	CMPDIF2	CMPDIF1	CMPDIF0
23	22	21	20	19	18	17	16
Reserved					CIF	BRK1IF	BRK0IF
15	14	13	12	11	10	9	8
Reserved		CMPUIF5	CMPUIF4	CMPUIF3	CMPUIF2	CMPUIF1	CMPUIF0
7	6	5	4	3	2	1	0
Reserved							PIF

位	描述	
[31:30]	Reserved	保留。
[29]	CMPDIF5	PWM通道5向下计数中断标志 当PWM通道5向下计数值等于EPWM_CMPDAT5，硬件置标志。软件写1清除此位。
[28]	CMPDIF4	PWM通道4向下计数中断标志 当PWM通道4向下计数值等于EPWM_CMPDAT4，硬件置标志。软件写1清除此位。
[27]	CMPDIF3	PWM通道3向下计数中断标志 当PWM通道3向下计数值等于EPWM_CMPDAT3，硬件置标志。软件写1清除此位。
[26]	CMPDIF2	PWM通道2向下计数中断标志 当PWM通道2向下计数值等于EPWM_CMPDAT2，硬件置标志。软件写1清除此位。
[25]	CMPDIF1	PWM通道1向下计数中断标志 当PWM通道1向下计数值等于EPWM_CMPDAT1，硬件置标志。软件写1清除此位。
[24]	CMPDIF0	PWM通道0向下计数中断标志 当PWM通道0向下计数值等于EPWM_CMPDAT0，硬件置标志。软件写1清除此位。
[23:19]	Reserved	保留。
[18]	CIF	PWM通道0中心点中断标志 当PWM通道向下计数值等0，硬件置标志。软件写1清除此位。
[17]	BRK1IF	PWM 刹车1标志 0 = 在BKP1，PWM 刹车没有检测到故障信号 1 = 在BKP1，PWM 刹车检测到故障信号，这位置高 注意： 软件写1清除此位。

位	描述	
[16]	BRK0IF	PWM 刹车0标志 0 =在BKP0, PWM 刹车没有检测到故障信号 1 =在BKP0, PWM 刹车检测到故障信号, 这位置高 注意: 软件写1清除此位。
[15:14]	Reserved	保留。
[13]	CMPUIF5	PWM通道5向上计数中断标志 当PWM通道5向上计数值等于PWM_CMPDAT5, 硬件置标志。软件写1清除此位。
[12]	CMPUIF4	PWM通道4向上计数中断标志 当PWM通道4向上计数值等于PWM_CMPDAT4, 硬件置标志。软件写1清除此位。
[11]	CMPUIF3	PWM通道3向上计数中断标志 当PWM通道3向上计数值等于PWM_CMPDAT3, 硬件置标志。软件写1清除此位。
[10]	CMPUIF2	PWM通道2向上计数中断标志 当PWM通道2向上计数值等于PWM_CMPDAT2, 硬件置标志。软件写1清除此位。
[9]	CMPUIF1	PWM通道1向上计数中断标志 当PWM通道1向上计数值等于PWM_CMPDAT1, 硬件置标志。软件写1清除此位。
[8]	CMPUIF0	PWM通道0向上计数中断标志 当PWM通道0向上计数值等于PWM_CMPDAT0, 硬件置标志。软件写1清除此位。
[7:1]	Reserved	保留。
[0]	PIF	PWM通道0周期中断标志 边沿对齐模式: 当PWM向下计数到0时, 标志被硬件置位 中心对齐模式: 当PWM向下计数到0然后向上计数到EPWM_PERIOD, 标志被硬件置位。 软件写1清除此位。

EPWM刹车低电平检测恢复延时寄存器 (EPWM RESDLY)

寄存器	偏移地址	R/W	描述	复位值
EPWM_RESDLY	EPWM_BA+0x5C	R/W	EPWM刹车低电平检测恢复延时寄存器	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
Reserved				DELAY			
7	6	5	4	3	2	1	0
DELAY							

位	描述	
[31:12]	Reserved	保留.
[11:0]	DELAY	PWM刹车低电平检测恢复延时 12 位向下计数器

EPWM故障刹车控制寄存器(EPWM_BRKCTL)

寄存器	偏移地址	R/W	描述	复位值
EPWM_BRKCTL	EPWM_BA+0x60	R/W	EPWM故障刹车控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
NFPEN	Reserved	BKOD5	BKOD4	BKOD3	BKOD2	BKOD1	BKOD0
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
LVDTYPE	LVDBKEN	BRK1PEN	BK1ADCEN	BRK1A1EN	BRK1A0EN	SWBRK	Reserved
7	6	5	4	3	2	1	0
Reserved		BRK0PEN	BK0ADCEN	BRK0A1EN	BRK0A0EN	BRK1EN	BRK0EN

位	描述	
[31]	NFPEN	外部刹车输入引脚(BRAKE)噪音滤波使能位 0 = 外部刹车输入引脚(BRAKE)噪音滤波禁止. 1 = 外部刹车输入引脚(BRAKE)噪音滤波使能.
[30]	Reserved	保留.
[29]	BKOD5	PWM 通道5 刹车输出选择 0 = 当故障刹车条件检测到, PWM输出低 1 = 当故障刹车条件检测到, PWM输出高
[28]	BKOD4	PWM 通道4 刹车输出选择 0 = 当故障刹车条件检测到, PWM输出低 1 = 当故障刹车条件检测到, PWM输出高
[27]	BKOD3	PWM 通道3 刹车输出选择 0 = 当故障刹车条件检测到, PWM输出低 1 = 当故障刹车条件检测到, PWM输出高
[26]	BKOD2	PWM 通道2 刹车输出选择 0 = 当故障刹车条件检测到, PWM输出低 1 = 当故障刹车条件检测到, PWM输出高
[25]	BKOD1	PWM 通道1 刹车输出选择 0 = 当故障刹车条件检测到, PWM输出低 1 = 当故障刹车条件检测到, PWM输出高
[24]	BKOD0	PWM 通道0 刹车输出选择 0 = 当故障刹车条件检测到, PWM输出低 1 = 当故障刹车条件检测到, PWM输出高

位	描述	
[23:16]	Reserved	保留.
[15]	LVDTYPE	低电平检测恢复类型 0 =在BRK恢复延时计数器计数到0后, 刹车恢复。 1 =在下一个周期刹车恢复。
[14]	LVDBKEN	低电平检测触发PWM刹车1功能使能位 0 =刹车功能1触发通过低电平检测禁止 1 =刹车功能1触发通过低电平检测使能
[13]	BRK1PEN	外部引脚作为刹车BRK1源使能位 0 = 外部引脚作为刹车BRK1源禁止 1 =外部引脚作为刹车BRK1源使能
[12]	BRK1ADCEN	ADC作为刹车BRK1源使能位 0 =ADC作为刹车BRK1源禁止 1 = ADC作为刹车BRK1源使能
[11]	BRK1A1EN	ACMP1作为刹车BRK1源使能位 0 = ACMP1作为刹车BRK1源禁止 1 = ACMP1作为刹车BRK1源使能
[10]	BRK1A0EN	ACMP0作为刹车BRK1源使能位 0 = ACMP0作为刹车BRK1源禁止 1 = ACMP0作为刹车BRK1源使能
[9]	SWBRK	软件刹车 0 = 禁止PWM软件刹车并回到正常PWM功能。 1 = 立即产生PWM刹车.
[8:6]	Reserved	保留.
[5]	BRK0PEN	外部引脚作为刹车BRK0源使能位 0 = 外部引脚作为刹车BRK0源禁止 1 =外部引脚作为刹车BRK0源使能
[4]	BRK0ADCEN	ADC作为刹车BRK0源使能位 0 =ADC作为刹车BRK0源禁止 1 = ADC作为刹车BRK0源使能
[3]	BRK0A1EN	ACMP1作为刹车BRK0源使能位 0 = ACMP1作为刹车BRK0源禁止 1 = ACMP1作为刹车BRK0源使能
[2]	BRK0A0EN	ACMP0作为刹车BRK0源使能位 0 = ACMP0作为刹车BRK0源禁止 1 = ACMP0作为刹车BRK0源使能

位	描述	
[1]	BRK1EN	刹车1功能使能位 0 =刹车1检测功能禁止 1 =刹车1检测功能使能
[0]	BRK0EN	刹车0功能使能位 0 =刹车0检测功能禁止 1 =刹车0检测功能使能

EPWM 死区间隔寄存器(EPWM_DTCTL)

寄存器	偏移地址	R/W	描述	复位值
EPWM_DTCTL	EPWM_BA+0x64	R/W	EPWM 死区间隔寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
DTCNT45							
15	14	13	12	11	10	9	8
DTCNT23							
7	6	5	4	3	2	1	0
DTCNT01							

位	描述	
[31:24]	Reserved	保留.
[23:16]	DTCNT45	通道4 和通道5 对(PWM4 和 PWM5 对)的死区间隔寄存器 该 8 位决定了死区长度. 死区长度的单位时间是由EPWM_CLKDIV 位所决定.
[15:8]	DTCNT23	通道2 和通道3 对(PWM4 和 PWM5 对)的死区间隔寄存器 该 8 位决定了死区长度. 死区长度的单位时间是由EPWM_CLKDIV 位所决定.
[7:0]	DTCNT01	通道0 和通道1 对(PWM4 和 PWM5 对)的死区间隔寄存器 该 8 位决定了死区长度. 死区长度的单位时间是由EPWM_CLKDIV 位所决定.

EPWM相位换相寄存器(EPWM_PHCHG)

寄存器	偏移地址	R/W	描述	复位值
EPWM_PHCHG	EPWM_BA+0x78	R/W	EPWM相位修改寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		ACMP1TEN	ACMP0TEN	A1POSSEL		A0POSSEL	
23	22	21	20	19	18	17	16
Reserved		TRGSEL		Reserved			
15	14	13	12	11	10	9	8
Reserved		MSKEN5	MSKEN4	MSKEN3	MSKEN2	MSKEN1	MSKEN0
7	6	5	4	3	2	1	0
Reserved		MSKDAT5	MSKDAT4	MSKDAT3	MSKDAT2	MSKDAT1	MSKDAT0

位	描述	
[31:30]	Reserved	保留.
[29]	ACMP1TEN	ACMP1 触发功能使能位 0 = ACMP1 触发PWM功能禁止. 1 = ACMP1 触发PWM功能使能.
[28]	ACMP0TEN	ACMP0 触发功能使能位 0 = ACMP0触发PWM功能禁止. 1 = ACMP0 触发PWM功能使能.
[27:26]	A1POSSEL	选择比较器1正极性输入源 选择ACMP1的正极性输入源 00 = 选择 ACMP1_P0 (PC.0) 作为ACMP1的输入 01 =选择ACMP1_P1 (PC.1) 作为ACMP1的输入 10 =选择ACMP1_P2 (PD.1) 作为ACMP1的输入 11 = 保留.
[25:24]	A0POSSEL	选择比较器0正极性输入源 选择ACMP0的正极性输入源 00 = 选择 ACMP0_P0 (PB.0) 作为ACMP0的输入 01 =选择ACMP0_P1 (PB.1) 作为ACMP0的输入 10 =选择ACMP0_P2 (PB.2) 作为ACMP0的输入 11 = 保留.
[23]	Reserved	保留.

位	描述	
[22:20]	TRGSEL	<p>换相触发选择</p> <p>选择触发条件从PHCHG_NXT装载到PHCHG</p> <p>当触发条件发生，将装载PHCHG和PHCHG_NXT</p> <p>相位改变：PWM输入将屏蔽，根据定义的PHCHG中MSKENn 和 MSKDATn。</p> <p>000 = 定时器0事件触发</p> <p>001 = 定时器1事件触发</p> <p>010 = 定时器2事件触发</p> <p>011 = HALLSTS (EPWM_PHCHGNXT[18:16])匹配霍尔传感器状态触发</p> <p>100 = ACMP0事件触发</p> <p>101 = ACMP1事件触发</p> <p>110 = 保留.</p> <p>111 = 自动换相功能禁止.</p>
[19:14]	Reserved	保留.
[13]	MSKEN5	<p>使能PWM5屏蔽功能</p> <p>0 = PWM5 屏蔽功能禁止</p> <p>1 = PWM5 屏蔽功能使能</p>
[12]	MSKEN4	<p>使能PWM4屏蔽功能</p> <p>0 = PWM4 屏蔽功能禁止</p> <p>1 = PWM4 屏蔽功能使能</p>
[11]	MSKEN3	<p>使能PWM3屏蔽功能</p> <p>0 = PWM3 屏蔽功能禁止</p> <p>1 = PWM3 屏蔽功能使能</p>
[10]	MSKEN2	<p>使能PWM2屏蔽功能</p> <p>0 = PWM2 屏蔽功能禁止</p> <p>1 = PWM2 屏蔽功能使能</p>
[9]	MSKEN1	<p>使能PWM1屏蔽功能</p> <p>0 = PWM1 屏蔽功能禁止</p> <p>1 = PWM1 屏蔽功能使能</p>
[8]	MSKEN0	<p>使能PWM0屏蔽功能</p> <p>0 = PWM0 屏蔽功能禁止</p> <p>1 = PWM0 屏蔽功能使能</p>
[7:6]	Reserved	Reserved.
[5]	MSKDAT5	<p>使能PWM5屏蔽输出数据</p> <p>0 = PWM5屏蔽输出状态为0</p> <p>1 = PWM5屏蔽输出状态为1</p>
[4]	MSKDAT4	<p>使能PWM4屏蔽输出数据</p> <p>0 = PWM4屏蔽输出状态为0</p> <p>1 = PWM4屏蔽输出状态为1</p>

位	描述	
[3]	MSKDAT3	使能PWM3屏蔽输出数据 0 = PWM3屏蔽输出状态为0 1 = PWM3屏蔽输出状态为1
[2]	MSKDAT2	使能PWM2屏蔽输出数据 0 = PWM2屏蔽输出状态为0 1 = PWM2屏蔽输出状态为1
[1]	MSKDAT1	使能PWM1屏蔽输出数据 0 = PWM1屏蔽输出状态为0 1 = PWM1屏蔽输出状态为1
[0]	MSKDAT0	使能PWM0屏蔽输出数据 0 = PWM0屏蔽输出状态为0 1 = PWM0屏蔽输出状态为1

EPWM下一相位换相寄存器 (EPWM_PHCHGNXT)

寄存器	偏移地址	R/W	描述	复位值
EPWM_PHCHGNXT	EPWM_BA+0x7C	R/W	EPWM下一相位修改寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		ACMP1TEN	ACMP0TEN	A1POSSEL		A0POSSEL	
23	22	21	20	19	18	17	16
Reserved		TRGSEL		Reserved		HALLSTS	
15	14	13	12	11	10	9	8
Reserved		MSKEN5	MSKEN4	MSKEN3	MSKEN2	MSKEN1	MSKEN0
7	6	5	4	3	2	1	0
Reserved		MSKDAT5	MSKDAT4	MSKDAT3	MSKDAT2	MSKDAT1	MSKDAT0

位	描述	
[31:30]	Reserved	保留.
[29]	ACMP1TEN	ACMP1 触发功能控制预设位 当装载触发条件发生, 此位将装载到PHCHG寄存器中ACMP1TEN位 详细请参考寄存器EPWM_PHCHG说明
[28]	ACMP0TEN	ACMP0 触发功能控制预设位 当装载触发条件发生, 此位将装载到PHCHG寄存器中ACMP0TEN位 详细请参考寄存器EPWM_PHCHG说明
[27:26]	A1POSSEL	选择比较器1正极性输入源预设位 当装载触发条件发生, 此位将装载到PHCHG寄存器中A1POSSEL位 详细请参考寄存器EPWM_PHCHG说明
[25:24]	A0POSSEL	选择比较器0正极性输入源预设位 当装载触发条件发生, 此位将装载到PHCHG寄存器中A0POSSEL位 详细请参考寄存器EPWM_PHCHG说明
[23]	Reserved	保留.

位	描述	
[22:20]	TRGSEL	<p>换相触发器选项预设位</p> <p>这个位域将被载入PHCHG的TRGSEL位域当载入触发条件发生。</p> <p>000 = 定时器0事件触发。</p> <p>001 = 定时器1事件触发。</p> <p>010 = 定时器2事件触发。</p> <p>011 = HALLSTS (EPWM_PHCHGNXT[18:16]) 匹配霍尔传感器状态触发。</p> <p>100 = ACMP0事件触发。</p> <p>101 = ACMP1事件触发。</p> <p>110 = 保留。</p> <p>111 = 自动相位换相功能禁止。</p> <p>详细描述请参考EPWM_PHCHG。</p>
[19]	Reserved	保留。
[18:16]	HALLSTS	<p>预测下一个霍尔状态</p> <p>该位指示预测下个换相的霍尔状态</p> <p>如果 TRGSEL (EPWM_PHCHG[22:20]) = 0x3,.</p> <p>当任何霍尔状态改变发生，硬件将在定时器2种比较位 (CAP2, CAP1, CAP0) 和 HALLSTS[2: 0]。</p> <p>如果比较匹配，将触发换相功能。</p>
[15:14]	Reserved	保留。
[13]	MSKEN5	<p>使能PWM5屏蔽功能预设位</p> <p>当载入触发条件发生，这个位将载入PHCHG中MSKEN5</p> <p>详细描述请参考EPWM_PHCHG。</p>
[12]	MSKEN4	<p>使能PWM5屏蔽功能预设位</p> <p>当载入触发条件发生，这个位将载入PHCHG中MSKEN4</p> <p>详细描述请参考EPWM_PHCHG。</p>
[11]	MSKEN3	<p>使能PWM5屏蔽功能预设位</p> <p>当载入触发条件发生，这个位将载入PHCHG中MSKEN3</p> <p>详细描述请参考EPWM_PHCHG。</p>
[10]	MSKEN2	<p>使能PWM5屏蔽功能预设位</p> <p>当载入触发条件发生，这个位将载入PHCHG中MSKEN2</p> <p>详细描述请参考EPWM_PHCHG。</p>
[9]	MSKEN1	<p>使能PWM5屏蔽功能预设位</p> <p>当载入触发条件发生，这个位将载入PHCHG中MSKEN1</p> <p>详细描述请参考EPWM_PHCHG。</p>
[8]	MSKEN0	<p>使能PWM5屏蔽功能预设位</p> <p>当载入触发条件发生，这个位将载入PHCHG中MSKEN0</p> <p>详细描述请参考EPWM_PHCHG。</p>
[7:6]	Reserved	保留。

位	描述	
[5]	MSKDAT5	使能PWM5屏蔽功能数据输出预设位 当载入触发条件发生，这个位将载入PHCHG中MSKDAT5 详细描述请参考EPWM_PHCHG。
[4]	MSKDAT4	使能PWM5屏蔽功能数据输出预设位 当载入触发条件发生，这个位将载入PHCHG中MSKDAT4 详细描述请参考EPWM_PHCHG。
[3]	MSKDAT3	使能PWM5屏蔽功能数据输出预设位 当载入触发条件发生，这个位将载入PHCHG中MSKDAT3 详细描述请参考EPWM_PHCHG。
[2]	MSKDAT2	使能PWM5屏蔽功能数据输出预设位 当载入触发条件发生，这个位将载入PHCHG中MSKDAT2 详细描述请参考EPWM_PHCHG。
[1]	MSKDAT1	使能PWM5屏蔽功能数据输出预设位 当载入触发条件发生，这个位将载入PHCHG中MSKDAT1 详细描述请参考EPWM_PHCHG。
[0]	MSKDAT0	使能PWM5屏蔽功能数据输出预设位 当载入触发条件发生，这个位将载入PHCHG中MSKDAT0 详细描述请参考EPWM_PHCHG。

EPWM换相选择控制寄存器 (EPWM_PHCHGALT)

寄存器	偏移地址	R/W	描述	复位值
EPWM_PHCHGALT	EPWM_BA+0x80	R/W	EPWM 换相选择控制寄存器	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						POSCTL1	POSCTL0

位	描述	
[31:2]	Reserved	保留。
[1]	POSCTL1	ACMP1正极性输入控制 0 = ACMP1的输入控制通过ACMP_CTL1 1 = ACMP1的输入控制通过PHCHG的A1POSSEL。 注意: 寄存器 ACMP_CTL1 的描述在比较器控制章节
[0]	POSCTL0	ACMP0正极性输入控制 0 = ACMP1的输入控制通过ACMP_CTL0 1 = ACMP1的输入控制通过PHCHG的A0POSSEL。 注意: 寄存器 ACMP_CTL0 的描述在比较器控制章节

EPWM 周期中断累加控制寄存器 (EPWM_IFA)

寄存器	偏移地址	R/W	描述	复位值
EPWM_IFA	EPWM_BA+0x84	R/W	EPWM 周期中断累加控制寄存器	0x0000_00F0

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
IFDAT				Reserved			
7	6	5	4	3	2	1	0
IFCNT				Reserved			IFAEN

位	描述	
[31:16]	Reserved	保留.
[15:12]	IFDAT	周期中断向下计数器事件寄存器（只读） 当IFAEN被设置，当有PWM中断标志被置1后IFDAT将减1，当IFDAT到0时，PWM中断将会发生，IFCNT值将重载到IFDAT.
[11:8]	Reserved	保留.
[7:4]	IFCNT	周期中断累加计数值设置寄存器（只写） 16级向下计数器值设定寄存器 当IFAEN被设置，IFCNT值将载入IFDAT并逐步减小。
[3:1]	Reserved	保留.
[0]	IFAEN	使能周期中断累加功能 0 = 周期中断累加功能禁止。 1 = 周期中断累加功能使能。

6.9 基本 PWM 发生器 (BPWM)

6.9.1 概述

NM1120 系列有 1 组 BPWM，每组 BPWM 发生器可配置成 2 个独立的 PWM 输出，BPWM CH0~BPWM CH1，或者可配置成 1 对互补的 PWM，(BPWM CH0, BPWM CH1) 带可编程的死区发生器。

PWM 发生器由一个 8 位的预分频器，一个时钟除频器可进行 5 种分频 (1, 1/2, 1/4, 1/8, 1/16)，两个 PWM 计时器包含两种时钟源，两个 16 位 PWM 向下计数器用于 PWM 周期控制，两路 16 位比较器用于 PWM 占空比控制和一个死区发生器。PWM 发生器提供两个独立的 PWM 中断标志，当相应的 PWM 周期向下计数器计数到 0 时，它们会通过硬件置位。每个 PWM 中断有其相对应的使能位，可以让 CPU 去请求相应的 PWM 中断。PWM 发生器也可以配置成单次模式，用于处理只有一次 PWM 周期的信号或者自动加载模式，用于连续的输出 PWM 波形。

当 DTCNT01 (BPWM_CTL[4]) 被置位，BPWM CH0 和 BPWM CH1 执行互补模式；这对 PWM 的时序，周期，占空比和死区时间都由 PWM0 的计数器和死区发生器决定。参考图 6.9-1 PWM 时钟源控制

为了避免 PWM 驱动输出管脚导致故障，16 位的周期向下计数器和 16 位的比较器带有双缓存。当用户写数据到计数器或者比较器的寄存器时，16 位向下计数器/比较器更新的数据只有在向下计数到 0 时才被装载。这个双缓存的特性可以避免在 PWM 输出的时候出现故障。

当 16 位的周期向下计数器计数到 0 时，中断请求将会产生。如果 PWM 定时器设置为自动加载模式，当向下计数到 0 时，它将自动重载 PWM 计数寄存器 (BPWM_PERIODx, x=0,1) 中的值，然后开始递减，如此重复。如果 PWM 定时器设置为单次模式，那么向下计数器在计数到 0 后将会停止并产生一个中断请求。

PWM 计数比较器的值被用于高脉冲宽度的调制。当向下计数器的值与比较寄存器的值相同时，这个计数器的控制逻辑就变输出为高电位。

6.9.2 特性

- 一个 PWM 发生器有一个 8 位的预分频器，一个时钟除频器，两个 PWM 计数器（向下计数），一个死区发生器和两个 PWM 输出。
- 支持 16 位的解析度
- 支持 PWM 中断请求与 PWM 周期同步。
- 支持单周期和自动加载模式
- 支持边沿对齐和中心对齐的操作模式

6.9.3 框图

图 6.9-1 展示了 PWM 时钟源控制和图 6.9-2 阐明了 PWM 的结构。

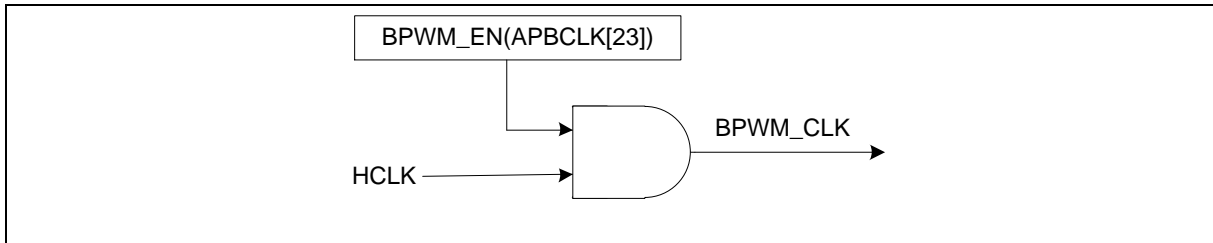


图 6.9-1 PWM 时钟源控制

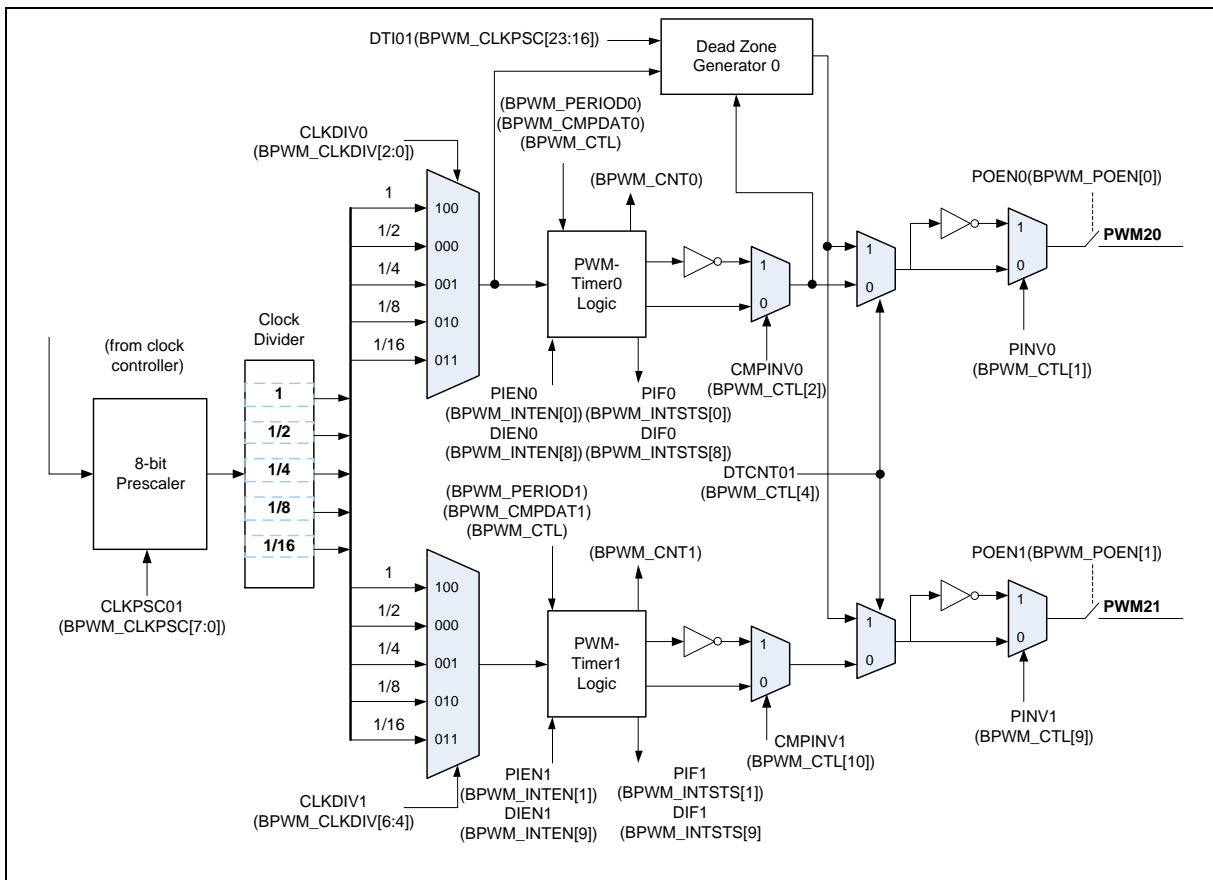


图 6.9-2 PWM 结构图

6.9.4 PWM 计数器操作

这个PWM控制器支持两种操作类型：边沿对齐和中心对齐。

6.9.4.1 边沿对齐的PWM (向下计数)

在边沿对齐的 PWM 输出模式，这个 16 位的 PWM 计数器将开始向下计数从 PERIOD(BPWM_PERIOD0-1[15:0])值向下减到占空比的值CMP(BPWM_CMPDAT0-1[15:0]),此时将触发PWMn发生器输出低电位. 计数器将继续向下计数到0，此时PWMn发生器将输出高电位并且

CMP 和 PERIOD 在 CNTMODEN=1 的时候将会更新，如果 PWM 的中断被使能 BPWM_INTEN(PWM_INTEN.n = 1),那么BPWM将产生中断请求。

PWM 的周期和占空比的控制是通过BPWM 计数寄存器 (BPWM_PERIOD0-1) 和BPWM 比较寄存器 (BPWM_CMPDAT0-1) 来配置.PWM计数器的时序操作如图 6.9-4.脉冲宽度的调制遵循下面的公式，PWM计数比较器说明如图 6.9-3.注意，当使能了BPWM_POEN后相应的GPIO管脚必须被配置为BPWM功能用于相应的BPWM通道。

- PWM 频率 = $\text{BPWM_CLK} / [(\text{prescale} + 1) * (\text{clock divider}) * (\text{PERIOD} + 1)]$.
- 占空比 = $(\text{CMP} + 1) / (\text{PERIOD} + 1)$
- $\text{CMP} \geq \text{PERIOD}$: PWM 输出总是高电位
- $\text{CMP} < \text{PERIOD}$: PWM 低脉宽 = $(\text{PERIOD} - \text{CMP})$ 单位[1]; PWM 高脉宽 = $(\text{CMP} + 1)$ 单位
- $\text{CMP} = 0$: PWM 低脉宽 = (PERIOD) 单位; PWM 高脉宽 = 1 单位

注: [1] 单位 = 一个PWM时钟周期.

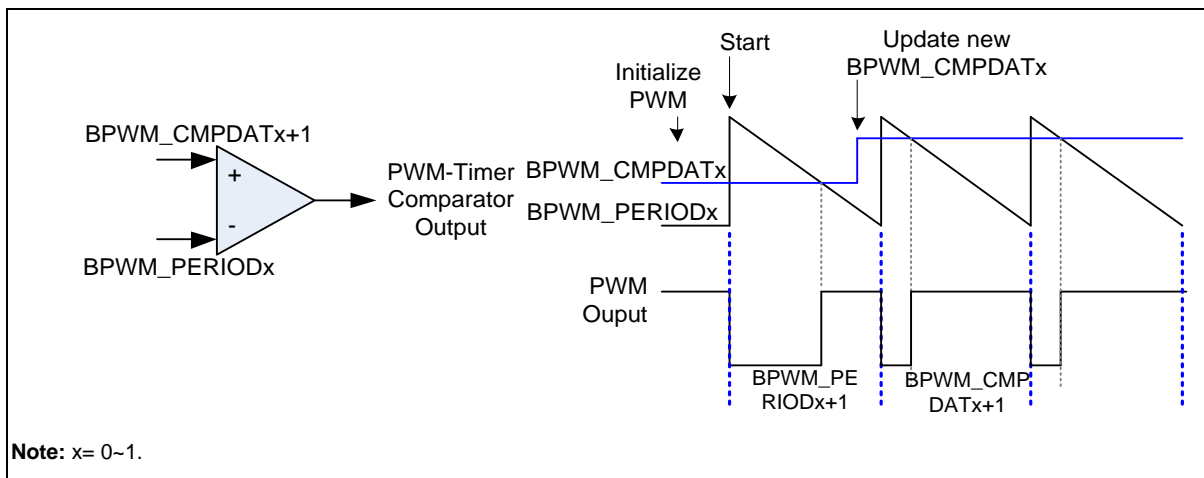


图 6.9-3 PWM 内部比较计数器说明

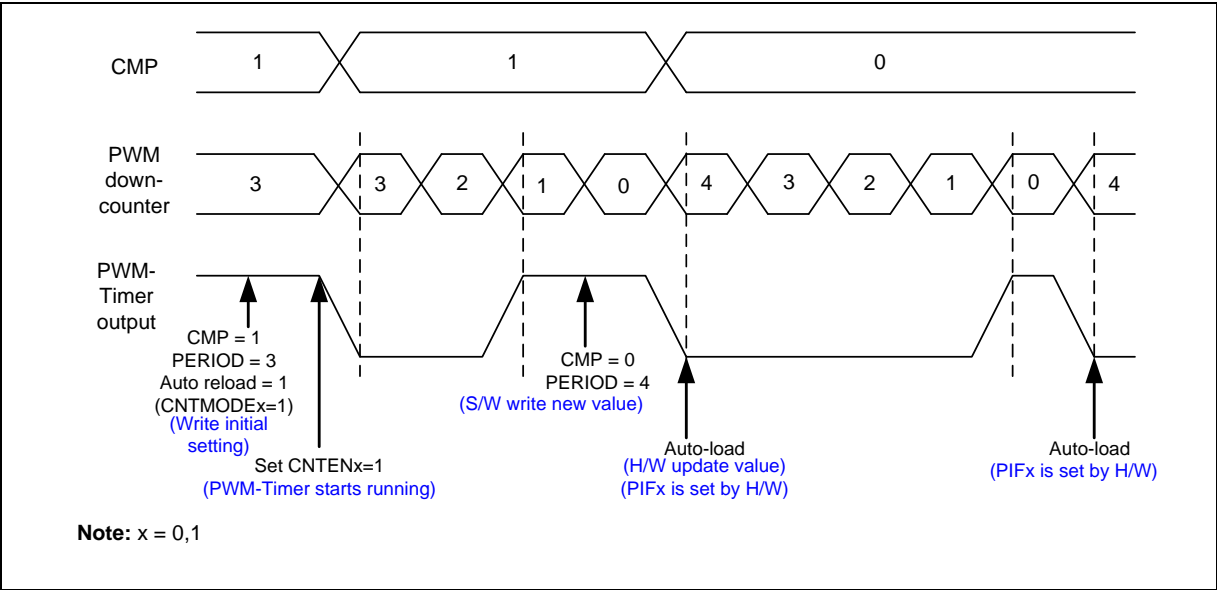


图 6.9-4 PWM 计数器操作时序

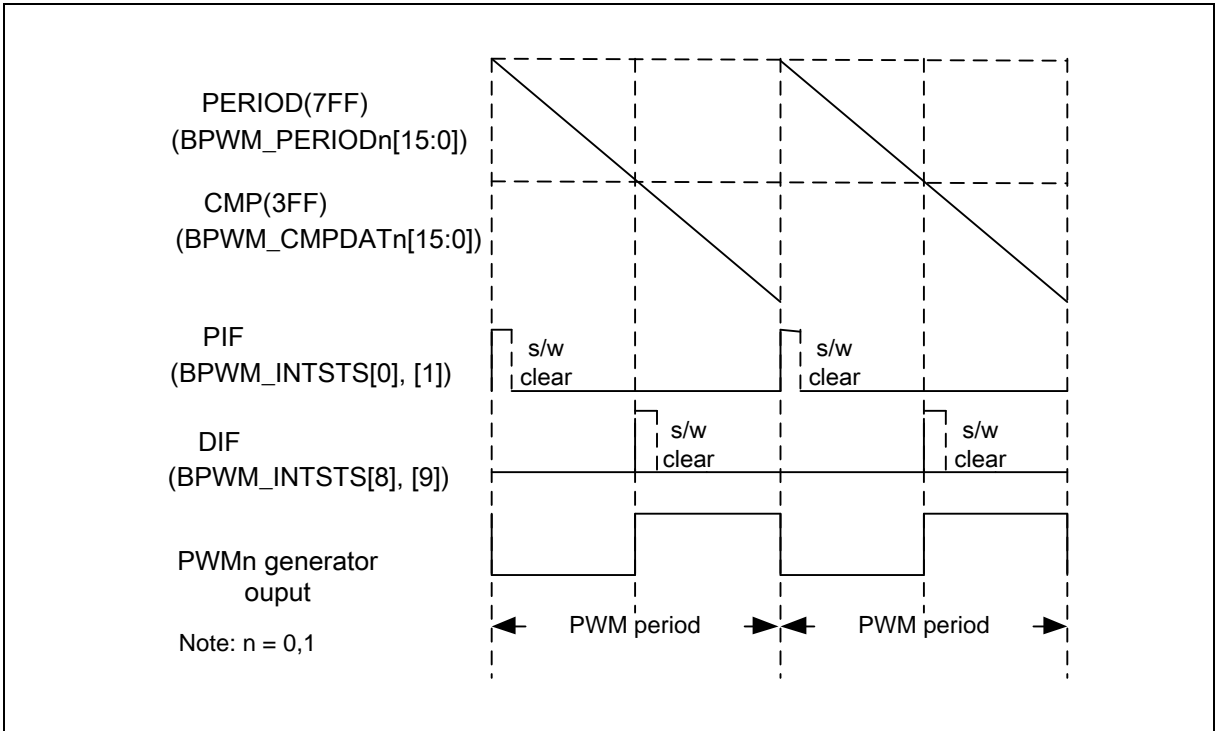


图 6.9-5 PWM 边沿对齐中断产生的时序波形

6.9.4.2 中心对齐的PWM (向上/向下计数)

当PWM的定时计数被配置成向上/向下计数模式时，中心对齐的PWM信号将通过这个模组产生。

PWM计数器将从0开始向上计数到CMP(BPWM_cmpdat0-1[15:0]);此时PWMn 发生器将输出低, 这个计数器会继续向上计数一直到PERIOD(BPWM_PERIOD0-1[15:0]).计数器到达配置的值时会自动向下计数, 当PWM计数器再次计数到CMP的值时PWMn将输出高。一旦PWM计数器计数到0时就可以更新 PWM 计数寄存器的 PERIOD 和 BPWM 比较寄存器 (0-1) 的 CMP 值用 CNTMODEn=1,n=0,1。

在中心对齐类型, PWM周期中断的产生, 如果PINTTYPE(BPWM_INTEN[16])=0, 那么在向下计数到0的时候就会产生中断, 比如在每一个PWM周期的开始时候和结束的时候会产生中断, 如果PINTTYPE(BPWM_INTEN[16]) = 1, 那么在向上计数到PERIOD时会产生中断, 比如在PWM周期的中心点位置。

- PWM 频率 = $\text{BPWM_CLK} / [(\text{prescale} + 1) * (\text{clock divider}) * 2 * (\text{PERIOD} + 1)]$.
- 占空比 = $[(2 \times \text{CMP}) + 1] / [2 \times (\text{PERIOD} + 1)]$
- $\text{CMP} > \text{PERIOD}$: PWM 输出总是高
- $\text{CMP} \leq \text{PERIOD}$: PWM 低脉宽 = $2 \times (\text{PERIOD} - \text{CMP}) + 1$ 单位[1]; PWM 高脉宽 = $(2 \times \text{CMP}) + 1$ 单位
- $\text{CMP} = 0$: PWM 低脉宽 = $2 \times \text{PERIOD} + 1$ 单位; PWM 高脉宽 = 1 单位

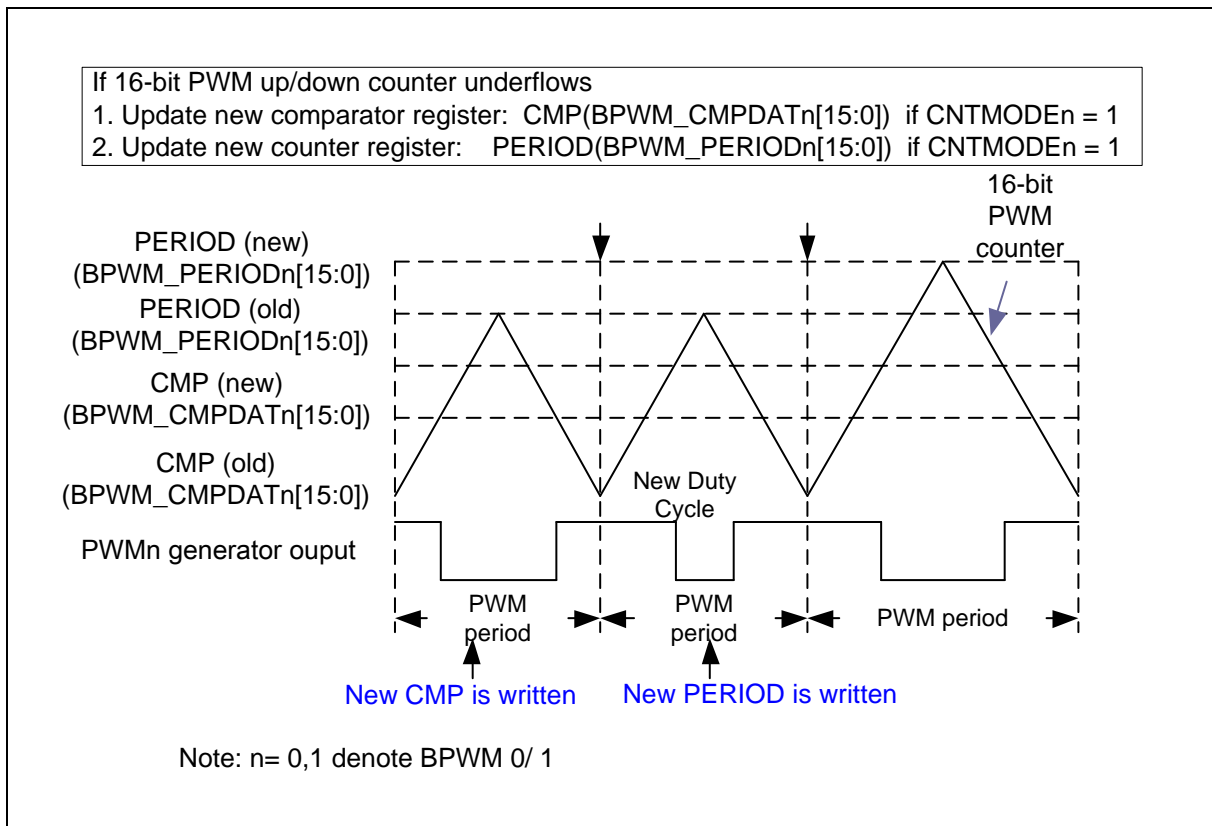


图 6.9-6 中心对齐类型的输出波形

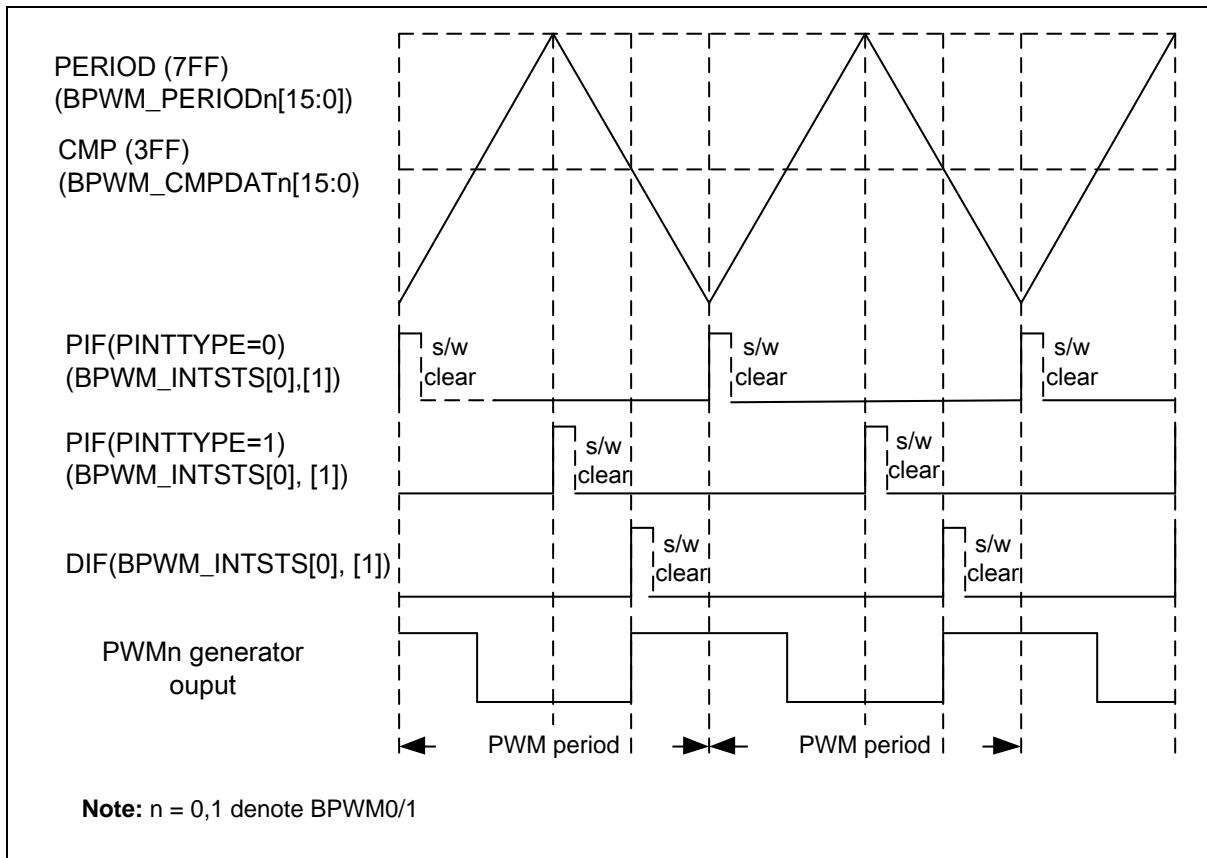


图 6.9-7 PWM 中心对齐中断产生的时序波形

6.9.4.3 PWM 双缓存, 自动加载和单次操作

PWM 计数器有双缓存功能, 新加载的数据会在下一个周期开始的时候才更新, 这样不会影响到当前的计数操作。PWM 计数器的值会写到 PERIOD(BPWM_PERIOD0-1) 中, PWM 当前计数值会从 CNTx(BPWM_CNT0-1[15:0]) 中读出。

如果 CNTMODE0 位被置 0, PWM0 将为单次操作模式, 如果 CNTMODE0 位被置 1, PWM0 将为自动装载模式。推荐在置 CNTEN0 位为 1 使 PWM0 计数器开始运行之前设置好 PWM0 的操作模式, 因为当 PWM0 的操作模式被改变的时候会将 BPWM_PERIOD0 和 BPWM_CMPDAT0 中的数据清 0 然后重新设置 PWM0 的周期和占空比设定。假如 PWM0 是单次操作模式, BPWM_CMPDAT0 和 BPWM_PERIOD0 将首先写入值, 然后再设置 CNTEN0 位为 1 去使能 PWM0 计数器开始运行。当 PWM0 的计数器从 BPWM_PERIOD0 的值向下计数到 0 时, BPWM_CMPDAT0 和 BPWM_PERIOD0 的值将会被硬件清 0 并且 PWM 计数器将挂起。需要软件写入新的 BPWM_CMPDAT0 和 BPWM_PERIOD0 的值去设定下一个单次的周期和占空比。当重新开始下一个单次操作时, BPWM_CMPDAT0 将被首先写入, 因为当 BPWM_PERIOD0 被写入非 0 值时, PWM0 计数器将自动重新开始计数。假如 PWM0 为自动装载模式, BPWM_CMPDAT0 和 BPWM_PERIOD0 将被首先写入, 然后再设置 CNTEN0 位为 1 使能 PWM0 计数器开始运行。当 BPWM_PERIOD0 的值向下计数到 0 时会自动重新装载该值。如果 BPWM_PERIOD0 的值被设置为 0 那么 PWM0 将挂起。PWM1 的操作和 PWM0 相同。

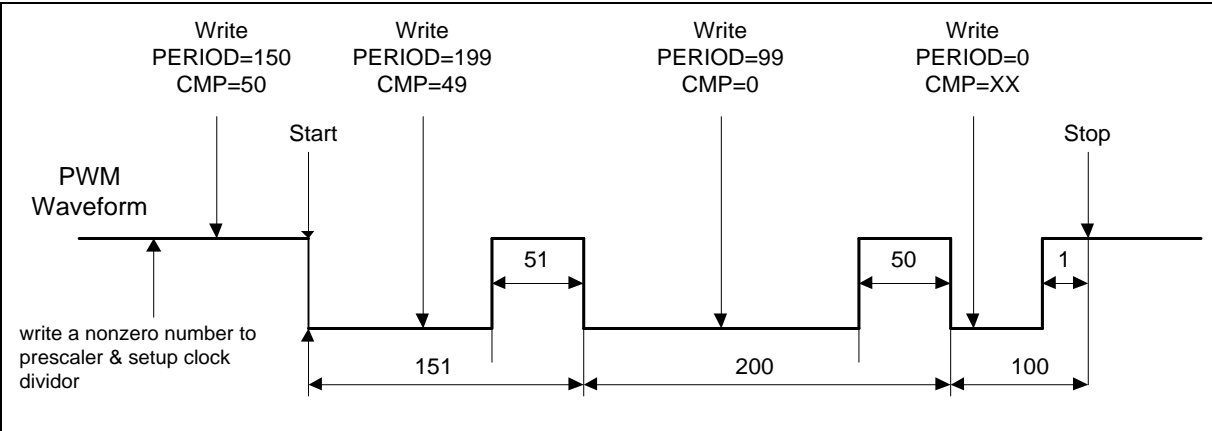


图 6.9-8 PWM 双缓存说明

6.9.4.4 调制占空比率

双缓存功能允许在当前周期的任何一个点写入CMP。该值将在下一个周期起效。

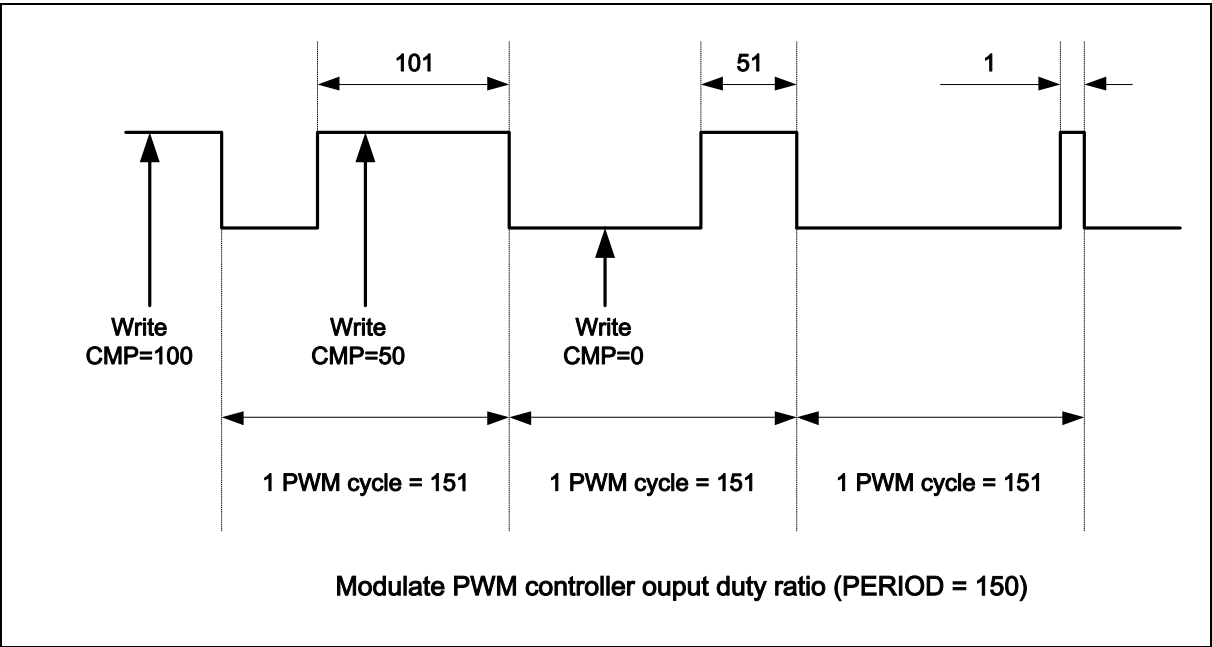


图 6.9-9 PWM 控制器输入占空比率

6.9.4.5 死区发生器

PWM 控制器带有死区发生器。它主要用于电源设备的保护。它的功能是产生一个可编程的时间间隔用于延迟PWM上升沿的输出。用户可以编程DTI01(BPWM_CLKPSC)确定死区间隔。

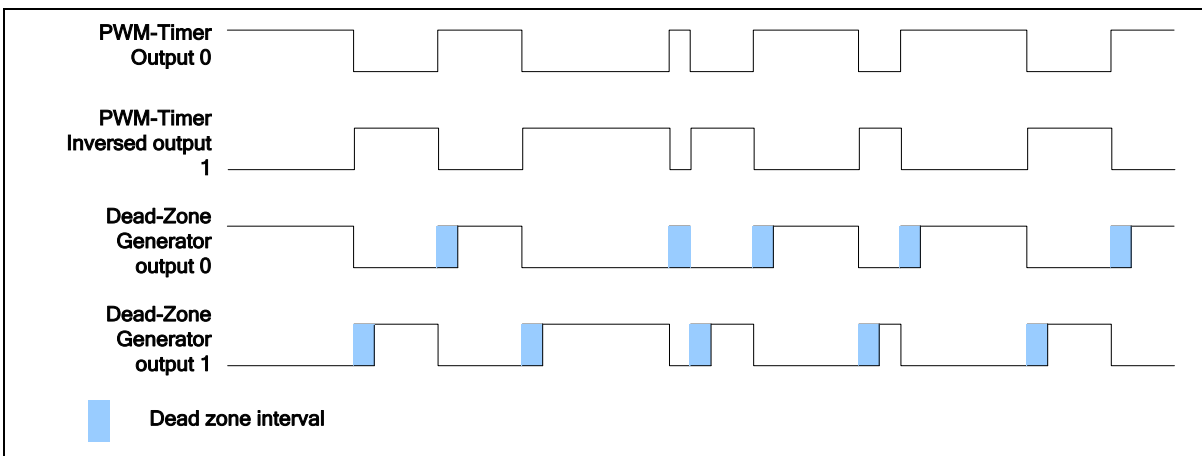


图 6.9-10 一对带死区插入功能的 PWM 输出

6.9.4.6 PWM-Timer 中断架构

有两个PWM中断，BPWM0_INT和BPWM1_INT.图 6.9-11 展示了PWM计数器的中断架构

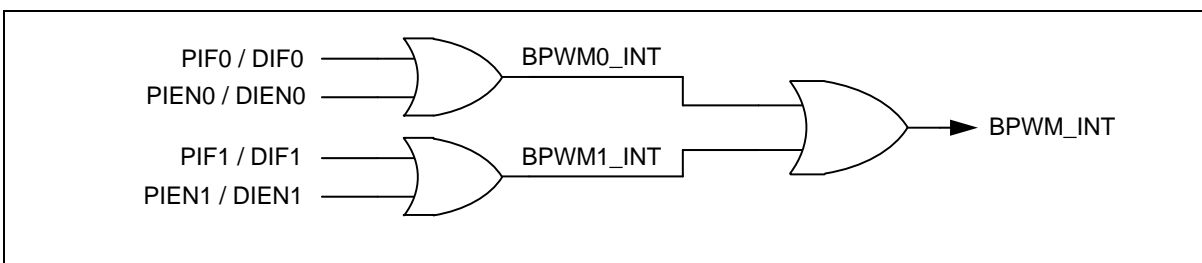


图 6.9-11 PWM 中断架构框图

6.9.4.7 PWM计数器 开始流程

推荐如下流程用于PWM的操作

1. 设置时钟源分频选择寄存器 (BPWM_CLKDIV)
2. 设置预分频 (BPWM_CLKPSC)
3. 设置极性翻转 ON/OFF ,死区发生器 ON/OFF,自动装载/单次模式和停止 PWM 计数 (BPWM_CTL) 器
4. 设置比较寄存器 (BPWM_CMPDAT) 用于设定PWM占空比
5. 设置PWM向下计数器寄存器 (BPWM_PERIOD) 用于设定PWM周期。
6. 设置中断使能寄存器 (BPWM_INTEN) (可选)
7. 设置相应的GPIO管脚作为PWM功能 (使能BPWM_POEN) 用于相应的PWM通道
8. 使能PWM计时器开始运行 (设置BPWM_CTL中的CNTENx =1, x=0或1)

6.9.4.8 PWM 计时器单次模式重新开始流程

在PWM单次模式PWM波形产生后，PWM计时器将自动停止。推荐下面的操作流程来重新开始产

生一次PWM单次波形。

- 设置比较寄存器（BPWM_CMPDAT）用于PWM占空比。
- 设置PWM 向下计数器寄存器（BPWM_PERIOD）用于设定PWM周期。在设定完PERIOD后，PWM波形将会产生

6.9.4.9 PWM 计数器停止流程

方式1:

设置16位计数器（PERIOD）为0，监视CNT(16位向下计数器的当前值)。当CNT到达0，关闭PWM计数器（BPWM_CTL中CNTENx，x=0或1）**(推荐)**

方式 2:

设置16位计数器（PERIOD）为0. 当中断请求发生，关闭PWM计数器（BPWM_CTL中CNTENx，x=0或1）**(推荐)**

方式3:

直接关闭PWM计数器（BPWM_CTL中CNTENx，x=0或1）**(不推荐)**

不推荐方式3的原因是关闭CNTENx将立即停止PWM的输出信号，导致PWM输出的占空比改变，这样有可能会损害电机的控制线路。

6.9.5 寄存器映射

R: 只读, W: 只写, R/W: 读写, C: 只能写0

寄存器	偏移	R/W	描述	复位值
BPWM 基地址: BPWM_BA = 0x4014_0000				
BPWM_CLKPSC	BPWM_BA+0x00	R/W	基本 PWM 预分频寄存器	0x0000_0000
BPWM_CLKDIV	BPWM_BA+0x04	R/W	基本PWM 时钟源除频寄存器	0x0000_0000
BPWM_CTL	BPWM_BA+0x08	R/W	基本PWM 控制寄存器	0x0000_0000
BPWM_PERIOD0	BPWM_BA+0x0C	R/W	基本PWM 周期计数器寄存器0	0x0000_0000
BPWM_CMPDAT0	BPWM_BA+0x10	R/W	基本PWM 比较寄存器0	0x0000_0000
BPWM_CNT0	BPWM_BA+0x14	R	基本PWM 数据寄存器0	0x0000_0000
BPWM_PERIOD1	BPWM_BA+0x18	R/W	基本PWM 周期计数器寄存器1	0x0000_0000
BPWM_CMPDAT1	BPWM_BA+0x1C	R/W	基本PWM 比较寄存器1	0x0000_0000
BPWM_CNT1	BPWM_BA+0x20	R	基本PWM 数据寄存器1	0x0000_0000
BPWM_INTEN	BPWM_BA+0x40	R/W	基本PWM 中断使能寄存器	0x0000_0000
BPWM_INTSTS	BPWM_BA+0x44	R/W	基本PWM 中断标志寄存器	0x0000_0000
BPWM_POEN	BPWM_BA+0x7C	R/W	基本PWM 输出使能	0x0000_0000

6.9.6 寄存器描述

BPWM 预分频寄存器 (BPWM_CLKPSC)

寄存器	偏移	R/W	描述	复位值
BPWM_CLKPSC	BPWM_BA+0x00	R/W	基本 PWM 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
DTI01							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CLKPSC01							

位	描述	
[31:24]	保留	保留.
[23:16]	DTI01	通道0和通道1成对的死区间隔 8位死区时间长度 死区长度的时间单位 = $[(\text{预分频}+1) \times (\text{时钟源除频器})] / \text{BPWM_CLK}$.
[15:8]	保留	保留.
[7:0]	CLKPSC01	时钟预分频 输入的时钟被除以 $(\text{CLKPSC01} + 1)$ 再送给相应的PWM计数器 如果 CLKPSC01=0, 那么时钟预分频器0输出时钟将会停止, 那么相应的PWM计数器也将停止

BPWM 时钟源除频选择寄存器 (BPWM_CLKDIV)

寄存器	偏移	R/W	描述	复位值
BPWM_CLKDIV	BPWM_BA+0x04	R/W	基本 PWM 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	CLKDIV1			Reserved	CLKDIV0		

位	描述	
[31:7]	保留	保留.
[6:4]	CLKDIV1	PWM计数器1 时钟源除频选择 选择PWM计数器1的时钟源除频. 000 = 1/2. 001 = 1/4. 010 = 1/8. 011 = 1/16. 100 = 1.
[3]	Reserved	保留.
[2:0]	CLKDIV0	PWM计数器0时钟源除频选择 选择PWM计数器0的时钟源除频. (同 CLKDIV1的一样)

BPWM 控制寄存器 (BPWM_CTL)

寄存器	偏移	R/W	描述	复位值
BPWM_CTL	BPWM_BA+0x08	R/W	基本 PWM 控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	CNTTYPE01	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				CNTMODE1	CMPINV1	PINV1	CNTEN1
7	6	5	4	3	2	1	0
Reserved			DTCNT01	CNTMODE0	CMPINV0	PINV0	CNTEN0

位	描述	
[31]	Reserved	保留。
[30]	CNTTYPE01	PWM01 对齐类型选择 0 = 边沿对齐类型。 1 = 中心对齐类型。
[29:12]	Reserved	保留。
[11]	CNTMODE1	PWM计数器 1 自动加载/单次模式 0 = 单次模式。 1 = 自动加载。 注: 如果该位改变, 将导致BPWM_PERIOD1和BPWM_CMPDAT1被清掉。
[10]	CMPINV1	PWM计数器1 输出翻转使能位 0 = 翻转无效。 1 = 翻转使能。
[9]	PINV1	PWM计数器 1 输出极性翻转使能位 0 = PWM1 输出极性翻转无效。 1 = PWM1 输出极性翻转使能。
[8]	CNTEN1	PWM计数器 1使能位 0 = 相应的PWM计数器停止。 1 = 相应的PWM计数器开始运行。
[7:5]	Reserved	保留。
[4]	DTCNT01	死区0发生器使能位 0 = 死区0发生器无效。 1 = 死区0发生器使能。

		注: 当死区发生器使能, PWM0和PWM1是一组互补。
[3]	CNTMODE0	PWM计数器0 自动加载/单次模式 0 = 单次模式. 1 = 自动加载. 注: 如果该位有改变, 那么将导致BPWM_PERIOD0和BPWM_CMPDAT0被清0.
[2]	CMPINV0	PWM计数器0 输出翻转使能位 0 = 翻转无效. 1 = 翻转使能.
[1]	PINV0	PWM计数器0 输出极性翻转使能位 0 = PWM0 输出极性翻转无效. 1 = PWM0 输出极性翻转使能.
[0]	CNTEN0	PWM计数器0 使能位 0 = 相应的PWM计数器停止运行. 1 = 相应的PWM计数器开始运行.

BPWM 计数器寄存器0-1 (BPWM_PERIOD0-1)

寄存器	偏移	R/W	描述	复位值
BPWM_PERIOD0	BPWM_BA+0x0C	R/W	基本 PWM 周期计数器寄存器0	0x0000_0000
BPWM_PERIOD1	BPWM_BA+0x18	R/W	基本 PWM 周期计数器寄存器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
PERIOD							
7	6	5	4	3	2	1	0
PERIOD							

位	描述	
[31:16]	Reserved	保留.
[15:0]	PERIOD	<p>基本 PWM 周期计数器寄存器</p> <p>PERIOD 数据决定PWM的周期.</p> <p>对于边沿对齐的类型: :</p> <p>$PWM \text{ 频率} = BPWM_CLK / [(预分频+1) * (时钟除频) * (PERIOD+1)]$.</p> <ul style="list-style-type: none"> 占空比率 = $(CMP+1) / (PERIOD+1)$. $CMP \geq PERIOD$: PWM 输出总是高电平. $CMP < PERIOD$: PWM 低脉宽 = $(PERIOD - CMP)$ 单位; PWM 高脉宽 = $(CMP+1)$ 单位. $CMP = 0$: PWM 低脉宽 = $(PERIOD)$ 单位; PWM 高脉宽 = 1 单位. <p>对于中心对齐类型:</p> <p>$PWM \text{ 频率} = BPWM_CLK / [(prescale+1) * (clock \ divider) * 2 * (PERIOD+1)]$.</p> <ul style="list-style-type: none"> 占空比率 = $[(2 \times CMP) + 1] / [2 \times (PERIOD+1)]$. $CMP > PERIOD$: PWM 输出总是高电平. $CMP \leq PERIOD$: PWM 低脉宽 = $2 \times (PERIOD - CMP) + 1$ 单位; PWM 高脉宽 = $(2 \times CMP) + 1$ 单位. $CMP = 0$: PWM 低脉宽 = $2 \times PERIOD + 1$ 单位; PWM 高脉宽 = 1 单位. <p>(单位 = 一个PWM 时钟周期).</p> <p>注: 任何时候写值到PERIOD都将在下一个PWM周期起效.</p> <p>注: 当PWM在中心对齐模式下, PERIOD 的值应该在0x0000 到0xFFFFE 之间, 如果PERIOD 等于0xFFFF, PWM 将工作在不可预期的情况下.</p> <p>注: 当 PERIOD 值设为0, PWM输出总是高电平.</p>

BPWM 比较寄存器0-1 (BPWM_CMPDAT0-1)

寄存器	偏移	R/W	描述	复位值
BPWM_CMPDAT0	BPWM_BA+0x10	R/W	基本 PWM 比较寄存器0	0x0000_0000
BPWM_CMPDAT1	BPWM_BA+0x1C	R/W	基本 PWM 比较寄存器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
CMP							
7	6	5	4	3	2	1	0
CMP							

位	描述	
[31:16]	Reserved	保留.
[15:0]	CMP	<p>PWM 比较寄存器</p> <p>CMP 决定PWM 的占空比.</p> <p>PWM 频率 = $BPWM_CLK / [(prescale+1) * (clock\ divider) * (PERIOD+1)]$.</p> <p>对于边沿对齐类型:</p> <p>PWM 频率 = $BPWM_CLK / [(prescale+1) * (clock\ divider) * (PERIOD+1)]$.</p> <ul style="list-style-type: none"> 占空比率 = $(CMP+1) / (PERIOD+1)$. CMP \geq PERIOD: PWM 输出总是高电平. CMP < PERIOD: PWM 低脉宽 = (PERIOD-CMP) 单位; PWM 高脉宽 = (CMP+1) 单位. CMP = 0: PWM 低脉宽 = (PERIOD) 单位; PWM 高脉宽 = 1 单位. <p>对于中心对齐:</p> <p>PWM 频率 = $BPWM_CLK / [(prescale+1) * (clock\ divider) * 2 * (PERIOD+1)]$.</p> <ul style="list-style-type: none"> 占空比率 = $[(2 * CMP) + 1] / [2 * (PERIOD+1)]$. CMP > PERIOD: PWM 输出总是高电平. CMP \leq PERIOD: PWM 低脉宽 = $2 * (PERIOD - CMP) + 1$ 单位; PWM 高脉宽 = $(2 * CMP) + 1$ 单位. CMP = 0: PWM 低脉宽 = $2 * PERIOD + 1$ 单位; PWM 高脉宽 = 1 单位. <p>(单位 = 一个PWM 时钟周期).</p> <p>注: 任何时候写值到PERIOD都将在下一个PWM周期起效</p>

BPWM 数据寄存器0-1 (BPWM_CNT0-1)

寄存器	偏移	R/W	描述	复位值
BPWM_CNT0	BPWM_BA+0x14	R	基本 PWM 数据寄存器 0	0x0000_0000
BPWM_CNT1	BPWM_BA+0x20	R	基本 PWM 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
CNT							
7	6	5	4	3	2	1	0
CNT							

位	描述	
[31:16]	Reserved	保留.
[15:0]	CNT	PWM 数据寄存器 16位计数器可以让用户监控CNT当前值.

BPWM 中断使能寄存器 (BPWM_INTEN)

寄存器	偏移	R/W	描述	复位值
BPWM_INTEN	BPWM_BA+0x40	R/W	基本 PWM 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							PINTTYPE
15	14	13	12	11	10	9	8
Reserved						DIEN1	DIEN0
7	6	5	4	3	2	1	0
Reserved						PIEN1	PIEN0

位	描述	
[31:17]	Reserved	保留.
[16]	PINTTYPE	BPWM 中断周期类型选择 0 = 如果 BPWM 向下计数溢出, PIFn 将被置位. 1 = 如果 BPWM 计数值与 PERIODn 寄存器的值相同, PIFn 将置位 . 注: 该位只有在 BPWM 中心对齐的时候有效.
[15:10]	Reserved	保留.
[9]	DIEN1	BPWM 通道1 工作中断使能位 0 = BPWM 通道1 工作中断禁用. 1 = BPWM 通道1 工作中断使能.
[8]	DIEN0	BPWM 通道0 工作中断使能位 0 = BPWM 通道0 工作中断禁用. 1 = BPWM 通道0 工作中断使能.
[7:2]	Reserved	保留.
[1]	PIEN1	BPWM 通道1 周期中断使能位 0 = BPWM 通道1 周期中断禁用. 1 = BPWM 通道1 周期中断使能.
[0]	PIEN0	BPWM 通道0 周期中断使能位 0 = BPWM 通道0 周期中断禁用. 1 = BPWM 通道0 周期中断使能.

BPWM 中断标识寄存器 (BPWM_INTSTS)

寄存器	偏移	R/W	描述	复位值
BPWM_INTSTS	BPWM_BA+0x44	R/W	基本 PWM 中断标识寄存器	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						DIF1	DIF0
7	6	5	4	3	2	1	0
Reserved						PIF1	PIF0

位	描述	
[31:10]	Reserved	保留。
[9]	DIF1	BPWM 通道1工作中断标志 当BPWM计数器向下计数到BPWM_CMPDAT1的值是，该标志被硬件置位，软件可以将该标志清除，通过向该位写1。 注: 如果CMP等于PERIOD,在边沿对齐类型下，该标志不起作用。
[8]	DIF0	BPWM 通道0工作中断标志 当BPWM计数器向下计数到BPWM_CMPDAT0的值是，该标志被硬件置位，软件可以将该标志清除，通过向该位写1。 注: 如果CMP等于PERIOD,在边沿对齐类型下，该标志不起作用。
[7:2]	Reserved	保留。
[1]	PIF1	BPWM 通道1 周期中断状态 当BPWM1计数器到达中断要求时（依据PWM_INTEN寄存器的PINTTYPE位），该位被硬件置位，可以通过软件向该位写1将该位清0。
[0]	PIF0	BPWM 通道0周期中断状态 当BPWM0计数器到达中断要求时（依据PWM_INTEN寄存器的PINTTYPE位），该位被硬件置位，可以通过软件向该位写1将该位清0。

注: 用户可以通过向BPWM_INTSTS相应位写1来清掉每个中断标志

BPWM 输出使能寄存器 (BPWM_POEN)

寄存器	偏移	R/W	描述	复位值
BPWM_POEN	BPWM_BA+0x7C	R/W	基本 PWM 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						POEN1	POEN0

位	描述	
[31:2]	Reserved	保留.
[1]	POEN1	通道1 输出使能寄存器 0 = BPWM 通道1输出到管脚禁用. 1 = BPWM 通道1输出到管脚使能 注: 相应的GPIO管脚必须设定为BPWM功能
[0]	POEN0	通道0输出使能寄存器 0 = BPWM 通道0输出到管脚禁用. 1 = BPWM 通道0输出到管脚使能 注: 相应的GPIO管脚必须设定为BPWM功能

6.10 看门狗定时器 (WDT)

6.10.1 概述

当系统运行到未知状态时，看门狗定时器用于执行系统复位。可以预防系统无限制的挂机。另外看门狗定时器也可以用于将系统从空闲/掉电模式下唤醒。

6.10.2 特征

- 18位自由运行的计数器用于看门狗超时间隔
- 超时间隔可选 (24 ~ 218) WDT_CLK 周期，超时时间范围在104ms~26.3168ms，如果 WDT_CLK = 10kHz 。
- 系统维持在复位状态时间为 $(1 / \text{WDT_CLK}) * 63$
- 如果看门狗时钟源选择位10kHz时，支持看门狗唤醒功能

6.10.3 框图

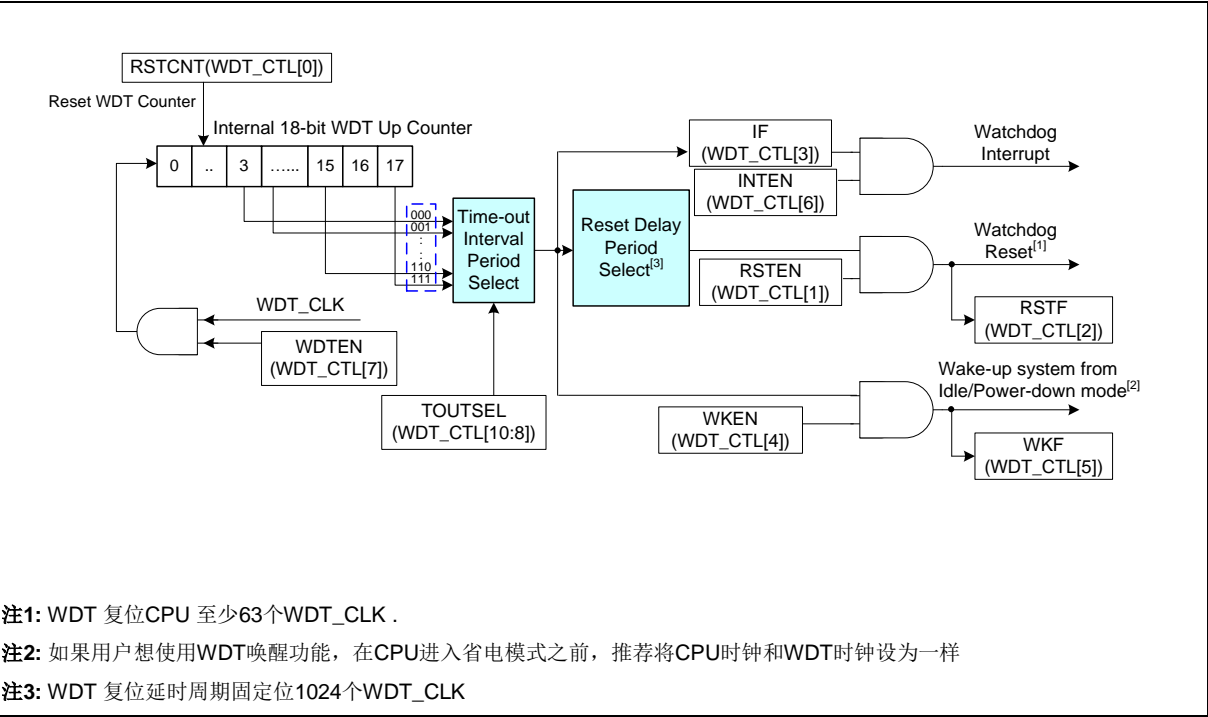


图 6.10-1 看门狗定时器框图

6.10.4 时钟控制

看门狗定时器时钟和框图如下所示.

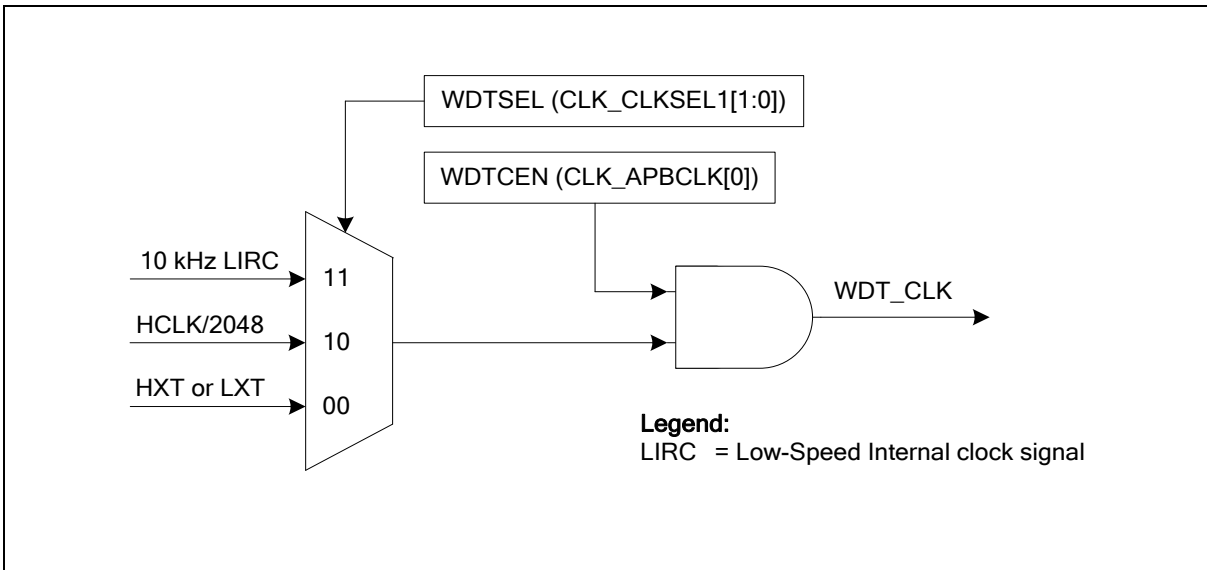


图 6.10-2 看门狗定时器时钟控制框图

6.10.5 基本配置

WDT 外设时钟使能在 WDTKEN(CLK_APBCLK[0]), 时钟源选择在 WDTSEL(CLK_CLKSEL1[1:0]).

6.10.6 功能描述

看门狗定时器 (WDT) 包含一个18位自由运行的向上计数器, 超时间隔时间可编程控制。图 6.10-3 显示WDT 超时间隔和复位周期时序。

6.10.6.1 WDT 超时中断

置 WDTEN(WDTCR[7]) 为 1 将使能 WDT 功能, WDT 计数器开始向上计数。通过设置 TOUTSEL(WDT_CTL[10:9]), 有 8 个超时时间间隔周期可以选择。当 WDT 向上计数到达 TOUTSEL(WDT_CTL[10:9]) 设定的值, WDT 超时中断将产生并且 IF(WDT_CTL[3]) 将立即置 1。

6.10.6.2 WDT 复位延迟周期和复位系统

在 IF(WDT_CTL[3]) 标志置 1 后有一个指定的 T_{RSTD} 延时周期。用户必须在 T_{RSTD} 延时周期之内使能 RSTCNT (WDT_CTL[0]) 位去复位 18 位 WDT 向上计数器的值, 以避免产生 WDT 超时复位信号。如果 WDT 向上计数器的值在指定的 T_{RSTD} 延时周期内没有被清掉, 如果 RSTEN(WDT_CTL[1]) 位被使能, WDT 将置 RSTF(WDT_CTL[2]) 标志为 1。然后芯片将立即进入复位状态。参考图 6.10-3, T_{RST} 复位周期将持续 63 个 WDT 时钟然后芯片重新开始从复位向量 (0x0000_0000) 开始执行程序。在 WDT 超时复位芯片后 RSTF(WDT_CTL[2]) 标志将保持为 1, 用户可以通过软件检查 RSTF(WDT_CTL[2]) 标志来判断系统是否有被 WDT 复位。

6.10.6.3 WDT 唤醒

在WDT超时中断信号产生且WKEN(WDT_CTL[4])位使能，如果WDT时钟源选择10kHz,系统可以从掉电模式下唤醒。同时WKF(WDT_CTL[5])标志将被自动置1.用户可以通过软件检查WFK(WDT_CTL[5])标志来判断系统是否有被WDT唤醒过。

TOUTSEL	超时间隔周期 T_{TIS}	复位延时周期 T_{RSTD}
000	$2^4 * T_{WDT}$	$1024 * T_{WDT}$
001	$2^6 * T_{WDT}$	$1024 * T_{WDT}$
010	$2^8 * T_{WDT}$	$1024 * T_{WDT}$
011	$2^{10} * T_{WDT}$	$1024 * T_{WDT}$
100	$2^{12} * T_{WDT}$	$1024 * T_{WDT}$
101	$2^{14} * T_{WDT}$	$1024 * T_{WDT}$
110	$2^{16} * T_{WDT}$	$1024 * T_{WDT}$
111	$2^{18} * T_{WDT}$	$1024 * T_{WDT}$

表 6.10-1 看门狗定时器超时间隔周期选择

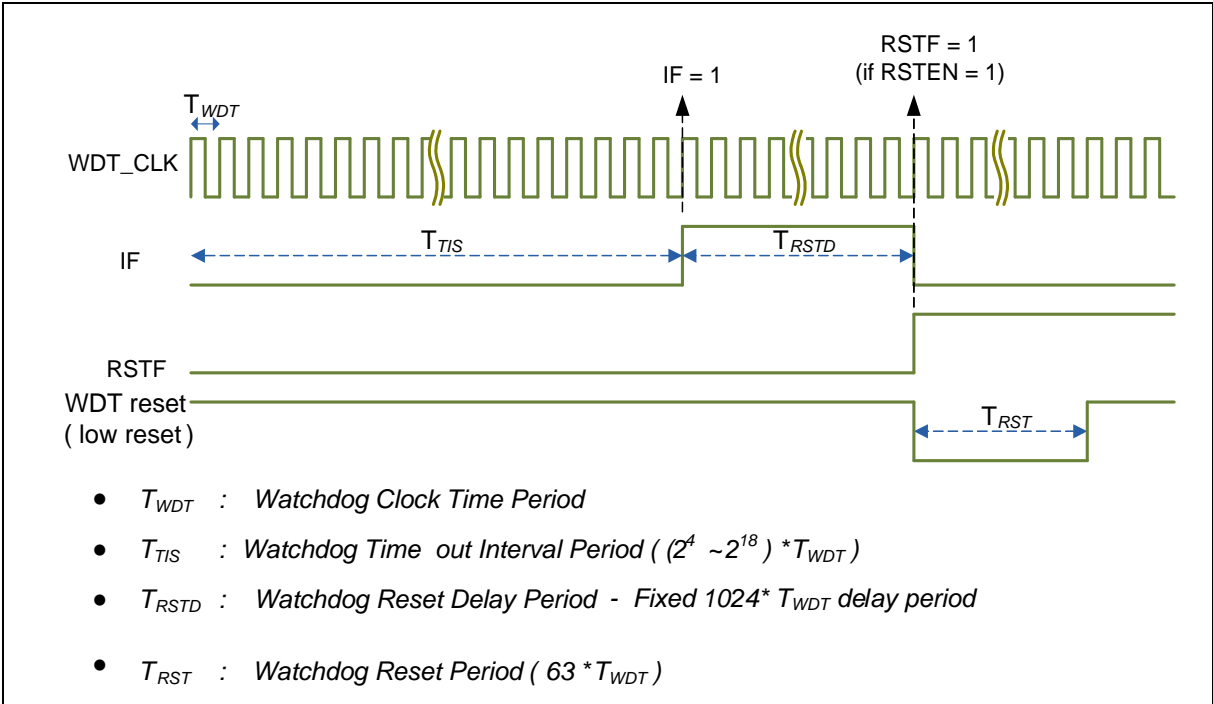


图 6.10-3 看门狗定时器超时间隔和复位周期时序

6.10.7 寄存器映射

R: 只读, W: 只写, R/W: 读写

寄存器	偏移	R/W	描述	复位值
WDT 基地址: WDT_BA = 0x4000_4000				
WDT_CTL	WDT_BA+0x00	R/W	看门狗定时器控制寄存器	0x0000_0700

6.10.8 寄存器描述

看门狗定时器控制寄存器 (WDT_CTL)

寄存器	偏移	R/W	描述	复位值
WDT_CTL	WDT_BA+0x00	R/W	看门狗定时器控制寄存器	0x0000_0700

注：该寄存器所有位都是写保护。要改写它需要通过向寄存器REGWRPROT地址SYS_BA+0x100写入解锁时序0x59,0x16,0x88解锁。

31	30	29	28	27	26	25	24
ICEDEBUG	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					TOUTSEL		
7	6	5	4	3	2	1	0
WDTEN	INTEN	WKF	WKEN	IF	RSTF	RSTEN	RSTCNT

位	描述	
[31]	ICEDEBUG	ICE 调试模式应答禁止位（写保护） 0 = ICE 调试模式应答影响WDT计数。 在CPU通过ICE挂起，WDT向上计数器将保持不变。 1 = ICE 调试模式应答禁止。 无论CPU是否被ICE挂起，WDT向上计数器都将继续计数。
[30:11]	Reserved	保留。
[10:8]	TOUTSEL	看门狗定时器超时时间间隔选择 3个位用于看门狗定时器超时时间间隔选择。 $000 = 2^4 * T_{WDT}$. $001 = 2^6 * T_{WDT}$. $010 = 2^8 * T_{WDT}$. $011 = 2^{10} * T_{WDT}$. $100 = 2^{12} * T_{WDT}$. $101 = 2^{14} * T_{WDT}$. $110 = 2^{16} * T_{WDT}$. $111 = 2^{18} * T_{WDT}$.
[7]	WDTEN	看门狗定时器使能位（写保护） 0 = WDT 禁止. (该动作将复位内部向上计数器的值.) 1 = WDT 使能。

位	描述	
[6]	INTEN	<p>看门狗定时器超时中断使能位（写保护）</p> <p>如果该位使能，WDT超时中断信号将产生并通知CPU</p> <p>0 = WDT 超时中断禁止</p> <p>1 = WDT 超时中断使能。</p>
[5]	WKF	<p>看门狗定时器超时唤醒标志</p> <p>该位表示WDT中断唤醒的标志状态</p> <p>0 = WDT 没有让芯片唤醒。</p> <p>1 = WDT超时中断信号产生，芯片从空闲或掉电模式下唤醒</p> <p>注: 该位写1清0</p>
[4]	WKEN	<p>看门狗定时器超时唤醒功能控制（写保护）</p> <p>如果该位置1，在IF被置1且INTEN被使能，WDT超时中断信号将产生一个唤醒触发事件给芯片。</p> <p>0 =如果WDT超时中断信号产生，唤醒触发事件禁止。</p> <p>1 = 如果WDT超时中断信号产生，唤醒触发事件使能。</p> <p>注: 只有在WDT时钟源为10kHz晶振时，芯片才能被WDT超时中断信号唤醒</p>
[3]	IF	<p>看门狗定时器超时中断标志</p> <p>当WDT向上计数器计数到选择的WDT超时时间间隔值时该位将被置1</p> <p>0 = WDT 超时中断没产生。</p> <p>1 = WDT 超时中断产生。</p> <p>注: 该位写1清0</p>
[2]	RSTF	<p>看门狗定时器超时复位标志</p> <p>该位表示系统是否曾经被WDT超时复位过</p> <p>0 = WDT 超时复位没有发生。</p> <p>1 = WDT 超时复位发生。</p> <p>注:该位写1清0</p>
[1]	RSTEN	<p>看门狗定时器超时复位使能位（写保护）</p> <p>如果WDT向上计数器的值在规定的WDT复位延迟周期($1024 * T_{WDT}$)内没有被清0，该位使能将使能WDT超时复位功能</p> <p>0 = WDT 超时复位功能禁止</p> <p>1 = WDT 超时复位功能使能</p>
[0]	RSTCNT	<p>复位看门狗定时器向上计数器（写保护）</p> <p>0 = 无效。</p> <p>1 =复位内部18位WDT向上计数器的值。</p> <p>注: 该位硬件自动清0。</p>

6.11 USCI – 通用串行控制接口控制器

6.11.1 概述

通用串行控制接口（USCI）是一个包含了几种通讯协议的灵活的接口模块。用户可以配置这个控制器用作UART,SPI,I2C功能的协议。

注: USCI中UART,I2C和SPI信息的具体细节, 请参看6.12,6.13,6.14章节

6.11.2 特征

根据应用需要该控制器可以独立的配置成如下几种协议:

- UART
- SPI
- I²C

为了增加可读性, USCI寄存器根据不同的功能有不同的别名。例如, USCI_CTL寄存器用于UART协议时别名为UUART_CTL,用于SPI协议时别名为USPI_CTL,用于I2C协议时别名为UI2C_CTL。

6.11.3 框图

USCI 的基本配置如图 6.11-1.

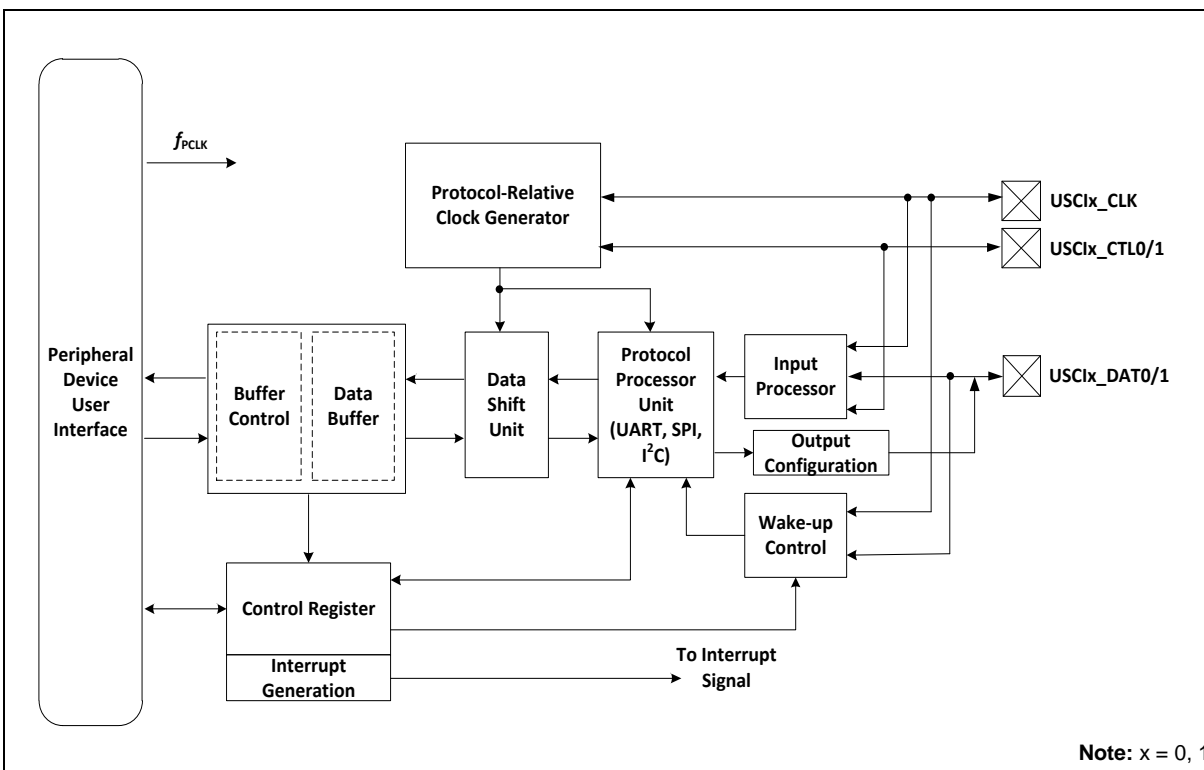


图 6.11-1 USCI 框图

6.11.4 功能描述

通用串行控制接口（USCI）的结构如图 6.11-1. 输入信号连接到输入处理器, 数据缓存和数据移位

单元实现数据传输。协议处理单元负责每种协议特定的协议功能。由协议时钟产生器负责特定协议的时序和时间事件。中断产生器单元负责所有协议的中断事件处理。唤醒控制单元负责特定协议的唤醒功能。

USCI 支持三种协议包括UART,SPI,I2C.它们通过寄存器FUNMODE(USCI_CTL[2:0])选择。注意，在改变协议类型之前FUNMODE必须先置0。

6.11.4.1 I/O 处理器

输入信号

所有输入信号都有相似的特征，所有协议公用这些信号。

表 6.11-1 列出了每种协议相应的输入信号。输入处理器会根据信号的属性处理每一种输入信号。比如信号反向选择控制或数字输入滤波

选择协议		UART	SPI	I ² C
串行总线时钟输	USCIx_CLK	-	SPIx_CLK	I2Cx_SCL
控制输入	USCIx_CTL0	-	SPIx_SS	-
	USCIx_CTL1	-	-	-
数据输入	USCIx_DAT0	UARTx_RX	SPIx_MOSI	I2Cx_SDA
	USCIx_DAT1	-	SPxI_MISO	-

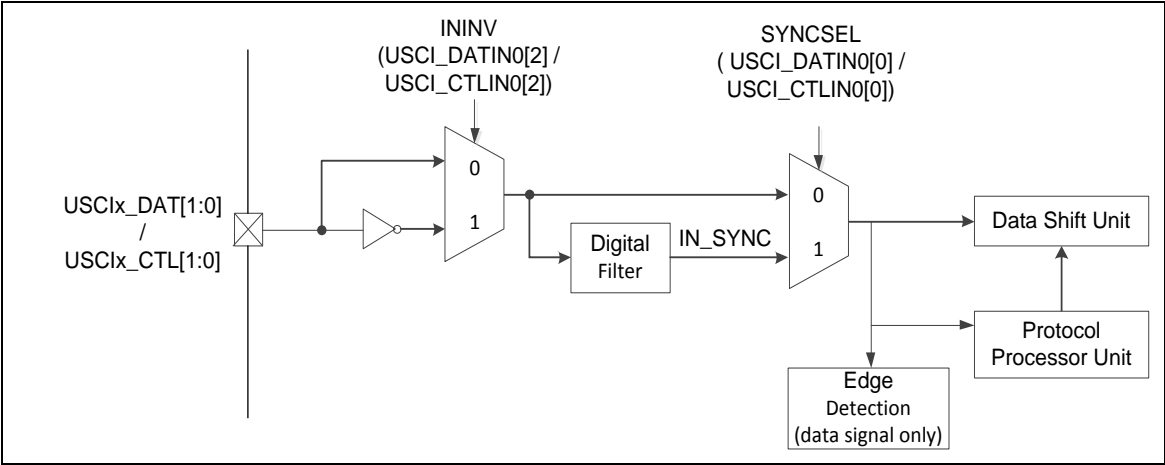
表 6.11-1 不同协议的输入信号

注 1: x = 0, 1

注2: 特定协议的描述在后面相应章节介绍

通用输入架构

数据和控制信号的输入架构包括反向器，数字滤波器和边沿检测（仅数据信号）



注: x = 0, 1

图 6.11-2 USCIx_DAT[1:0] 和 USCIx_CTL[1:0]输入架构

USCIx_CLK 输入架构和USCIx_CTL[1:0] 相似, 除了不支持反向功能.

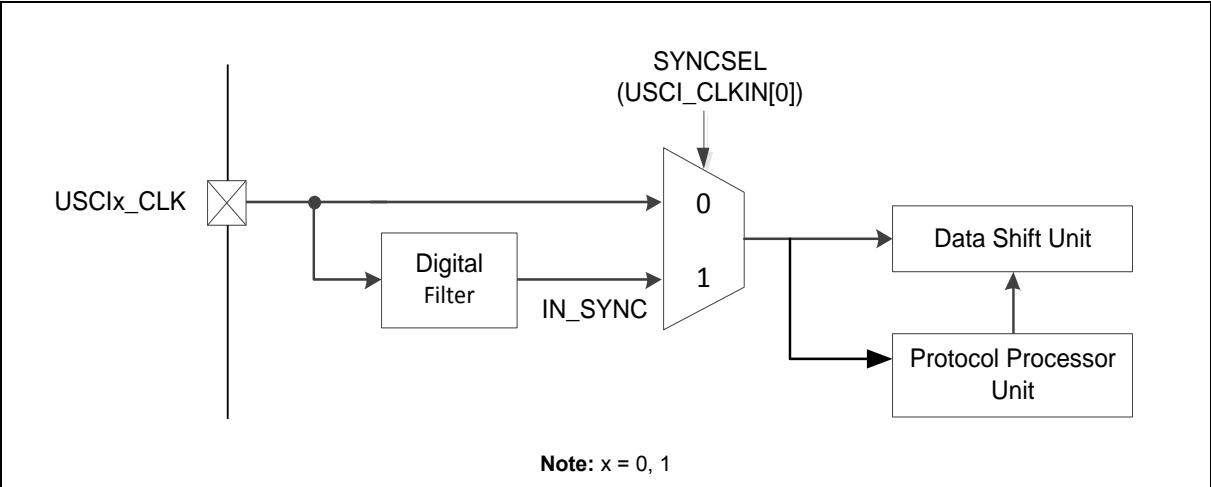


图 6.11-3 USCIx_CLK 输入架构

所有控制，时钟和数据输入架构的配置都分别在寄存器USCI_CTLIN0, USCI_CLKIN 和 USCI_DATIN0 中. EDGEDET (USCI_DATIN0[4:3]) 用来选择边沿检测条件. 注意，在UART模式下 USCI_DATIN0 的EDGEDET 必须设定为 2'b10. 设置可编程的边沿检测，可以根据起效的触发信号来表明某个事件发生。

设置ININV (USCI_DATIN0[2] / USCI_CTLIN0[2]) 寄存器可以翻转输入信号的极性，以适应数据移位单元和协议状态机的需要。

如果SYNCSEL (USCI_DATIN0[0] / USCI_CTLIN0[0] / USCI_CLKIN[0]) 被置0，输入信号不会有任何的延迟或滤波。如果输入信号上的噪声，也可能造成信号同步(IN_SYNC 被f_{PCLK}同步). 如果 SYNCSEL = 1，输入信号会有一个2-3倍f_{PCLK} 时钟周期的延迟滤波。

输出信号

表 6.11-2 表示每种协议相应的输出信号. 具体用到的输出信号取决于所选择的协议及分类.

选择的协议		UART	SPI	I ² C
总线时钟输出	USCIx_CLK	-	SPIx_CLK	I2Cx_SCL
控制输出	USCIx_CTL0	-	SPIx_SS	-
	USCIx_CTL1	-	-	-
数据输出	USCIx_DAT0	-	SPIx_MOSI	I2Cx_SDA
	USCIx_DAT1	UARTx_TX	SPIx_MISO	-

表 6.11-2 不同协议的输出信号

注1: $x = 0, 1$

注2: 协议的相关描述在相应章节

6.11.4.2 数据缓存

USCI控制器的数据处理是基于数据移位单元（DSU）和一个缓存寄存器，数据移位及缓存寄存器都是16位宽的。数据移位单元的输入包括移位数据，串行时钟，移位控制，根据不同的协议，发送输出管脚可以是USCIx_DAT0 管脚或者 USCIx_DAT1 管脚

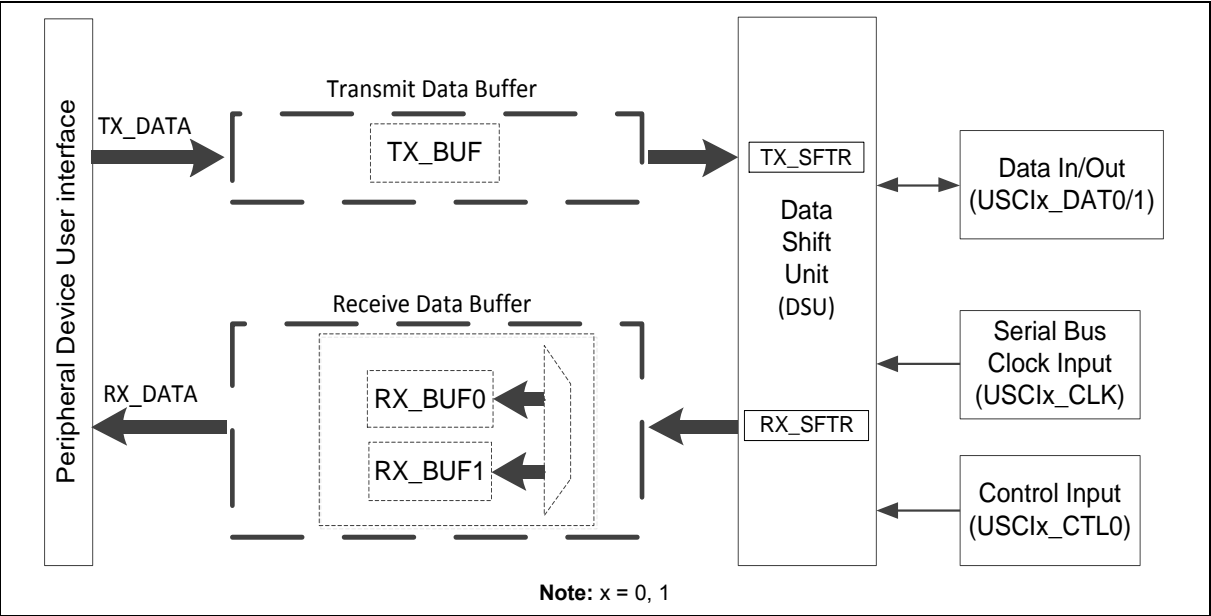


图 6.11-4 数据缓存框图

数据处理操作包括:

- 外设用户接口（APB）用于处理数据，中断，状态和控制信息。
- 发送器包括发送移位寄存器（TX_SFTR）和发送数据缓存（TX_BUF）。TXFULL (USCI_BUFSTS[9], TXEMPTY (USCI_BUFSTS[8]) 和 TXENDIF (USCI_PROTSTS[2]) 显示发送器的状态。
- 接收器包括接收移位寄存器（RX_SFTR）和双接收缓存架构（RX_BUF0, RX_BUF1）。在双缓存接收架构中，用户不需要考虑接收的时序，并且如果用户没有及时的读走USCI_RXDAT寄存器中的数据，会有两个数据被保存。

数据存取架构

数据存取架构包括读写数据，接收的数据保存在RX_BUF0和RX_BUF1接收缓存中，用户无需关心接收时序，接收缓存可以通过USCI_RXDAT寄存器读取，第一个数据被读出后第二个数据才可以

发送数据可以通过写寄存器USCI_TXDAT装载到TX_BUF中.

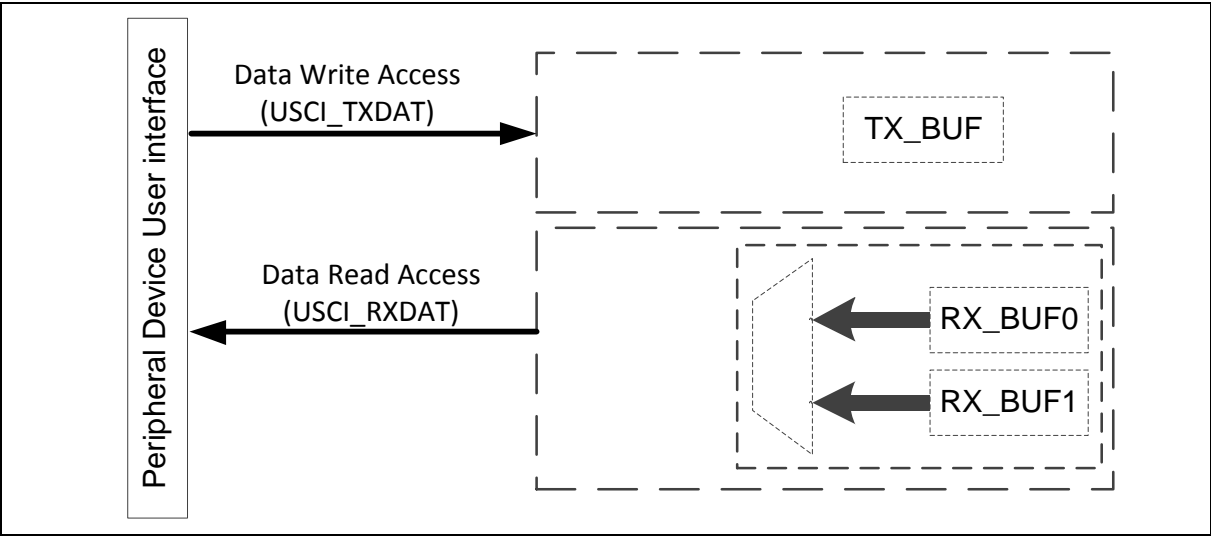


图 6.11-5 数据存取架构

发送数据路径

发送数据路径基于16位宽的发送移位寄存器（TX_SFTR）以及发送缓冲TX_BUF，发送数据的参数例如数据长度，是由共同的收发控制寄存器USCI_LINECTL决定的

发送缓冲

用户不能直接操作发送移位寄存器，它是自动从发送缓冲寄存器（TX_BUF）中更新过来的，当前数据发送完成，并且新的发送数据有效时才会更新

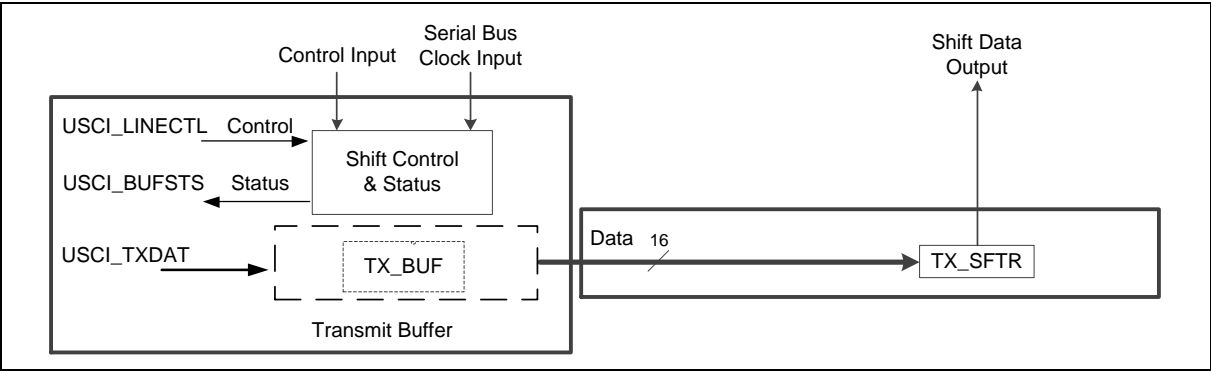


图 6.11-6 发送数据路径

发送数据有效确认

TXEMPTY (USCI_BUFSTS[8]) 的状态显示了发送缓冲(TX_BUF)中的数据是否有效，TXSTIF (USCI_PROTSTS[1])表示了每个数据的开始状态

- 如果USCI 控制器作主机，只有当发送缓冲（TX_BUF）有有效数据时才会触发，在这种情况下，发送移位寄存器加载发送缓冲中的数据

注: 主机决定开始传输数据.

- 如果USCI作从机，数据传输是应主机请求开始的，与发送缓冲（TX_BUF）的状态无关。如果主机要求发送数据，发送移位寄存器根据不同的协议加载数据

注：从机不能主动开始传输数据，只能响应。

- 从发送缓冲加载数据到数据移位单元的时序由协议配置决定

UART: 在正常工作模式下，如果TXEMPTY = 0 发送缓冲中的数据开始发送

SPI: 在主机模式下，TXEMPTY(USCI_BUFSTS[8])=0开始数据传输；在从机模式下，只有当从机选择信号有效且USCIx_CLK管脚有时钟信号时才开始传输数据。

I²C: 如果TXEMPTY=0，发送缓冲中的数据开始传输。

- 如果TXEMPTY(USCI_BUFSTS[8])=0，发送缓冲中的数据开始传送，发送缓冲中的内容（TX_BUF）在传送开始后就不能被重新写了，如果用户一定要改变TX_BUF中的内容，可以设置 TXRST(USCI_BUFCTL[16]) 为 1 清除 TX_BUF 中的内容，此外，TXEMPTY(USCI_BUFSTS[8])将在TX_BUF加载新的数据后自动清除。当一个传输正在进行时，TX_BUF可以加载新的数据，用户必须在新的传输开始请更新TX_BUF中的内容

接收数据路径

接收数据路径基于16位宽的接收移位寄存器RX_SFTR及接收缓冲RX_BUF0, RX_BUF1. 数据传输的参数，例如数据长度，移位方向都由共同的收发控制寄存器USCI_LINECTL控制，寄存器USCI_BUFSTS监控USCI_RXDAT中的数据有效性

接收缓冲

用户不能直接存取接收移位寄存器，但是当接收到一个完整的数据或者帧结束后其内容会自动加载到接收缓存。接收缓存中的数据可以从USCI_RXDAT寄存器读取

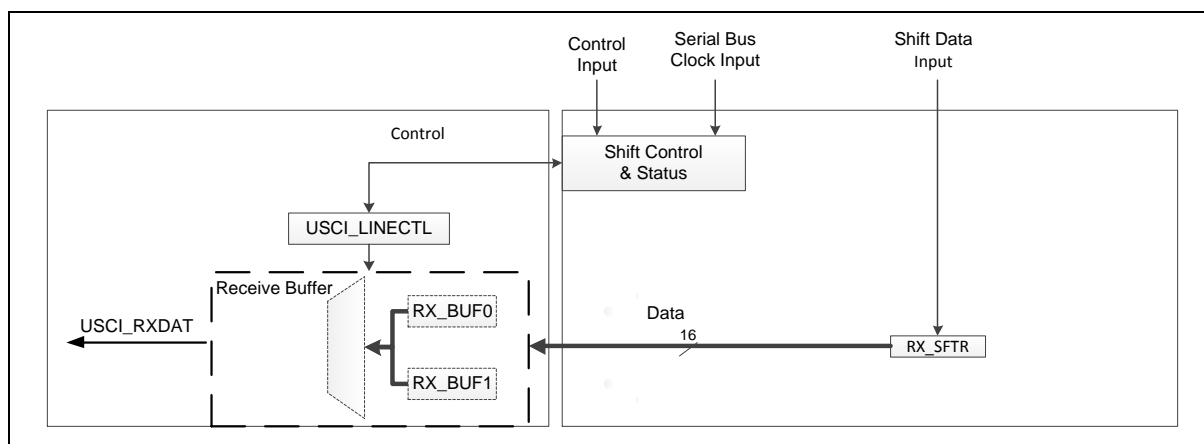


图 6.11-7 接收数据路径

6.11.4.3 协议控制和状态

协议控制及状态信息位于协议控制寄存器USCI_PROTCTL及协议状态寄存器USCI_PROTSTS，所有协议共用这些寄存器，所以这些寄存器中的位在不同的协议中代表不同的意义，详细信息请参考每个协议相关的寄存器描述

6.11.4.4 协议时钟产生器

USCI控制器包含一个协议时钟产生器，通过USCI_BRGEN来控制，写USCI_BRGEN寄存器复位该时钟产生器，协议时钟产生器的架构如图 6.11-8.

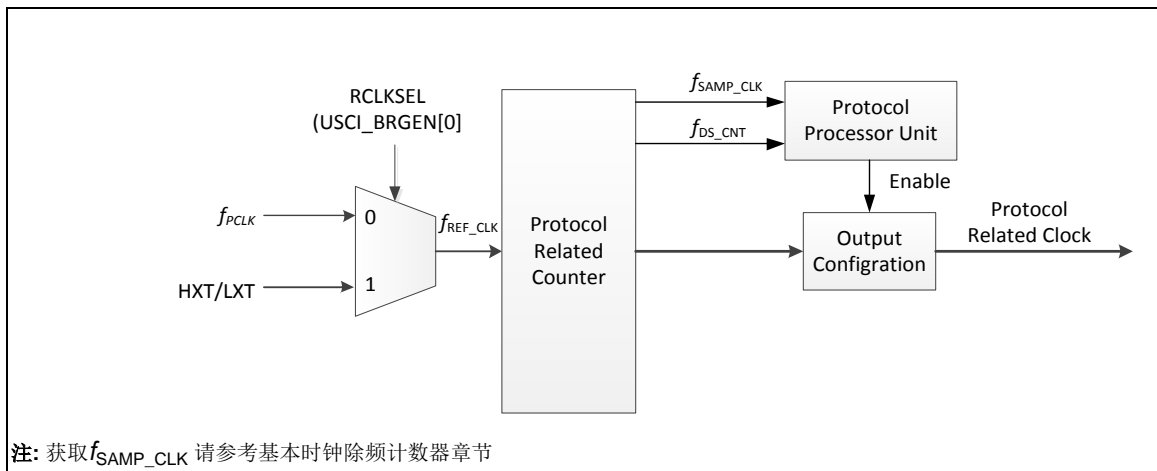


图 6.11-8 协议时钟产生器

协议时钟产生器包括基本时钟分频器及时序测量计数器，它基于一个除频器提供不同协议所需的频率，包含：

- 外部时钟输入用于产生输入频率 $f_{REF_CLK} = f_{ECLK}$ 给外部信号一个波特率。注意，外部时钟是半个系统时钟，因为外部时钟是通过系统时钟采样。
- 基本时钟除频器提供协议相关的时钟信号和其它协议相关的信号 (f_{SAMP_CLK} and f_{DS_CLK})。
- 时序测量计数器用于内部时间间隔测量，比如UART协议的波特率。
- 协议相关的时钟发生器输出信号可以在相应的管脚上获得（比如 SPI的USCIx_CLK）

基本时钟除频计数器

基本时钟除频计数器是用整除的方式产生 f_{REF_CLK2} , f_{REF_CLK} , f_{DIV_CLK} , f_{SCLK} , 和 f_{SAMP_CLK} 。除频器输出的频率是由 PTCLKSEL (USCI_BRGEN [1]), CLKDIV (USCI_BRGEN [25:16]), SPCLKSEL (USCI_BRGEN [3:2])控制。

基本时钟除频计数器用于产生相应协议的时序信号

$$f_{DIV_CLK} = f_{REF_CLK} \times \frac{1}{CLKDIV + 1} \text{ if PTCLKSEL} = 0$$

$$f_{DIV_CLK} = f_{REF_CLK} \times \frac{1}{(CLKDIV + 1) \times 2} \text{ if PTCLKSEL} = 1$$

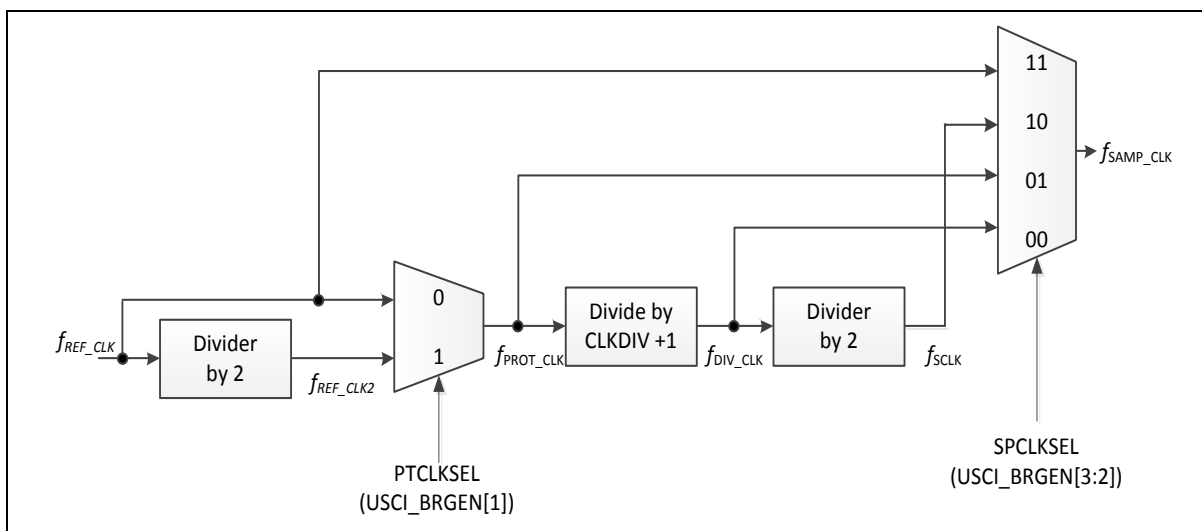


图 6.11-9 基本时钟除频计数器

时序测量计数器

时序测量计数器用来量测时间间隔，通过TMCNTEN (USCI_BRGEN [4]) = 1使能。当TMCNTSRC (USCI_BRGEN [5]) =1，计数器主频为 f_{DIV_CLK} ，否则其主频为 f_{PROT_CLK} ，串口数据收发的时候，计数器也在进行时序测量。计数器依靠 f_{PROT_CLK} 或 f_{DIV_CLK} 计数相关协议信号的长度，由此实现数据的连续收发，当计数器达到用户设定的值就会停止计数

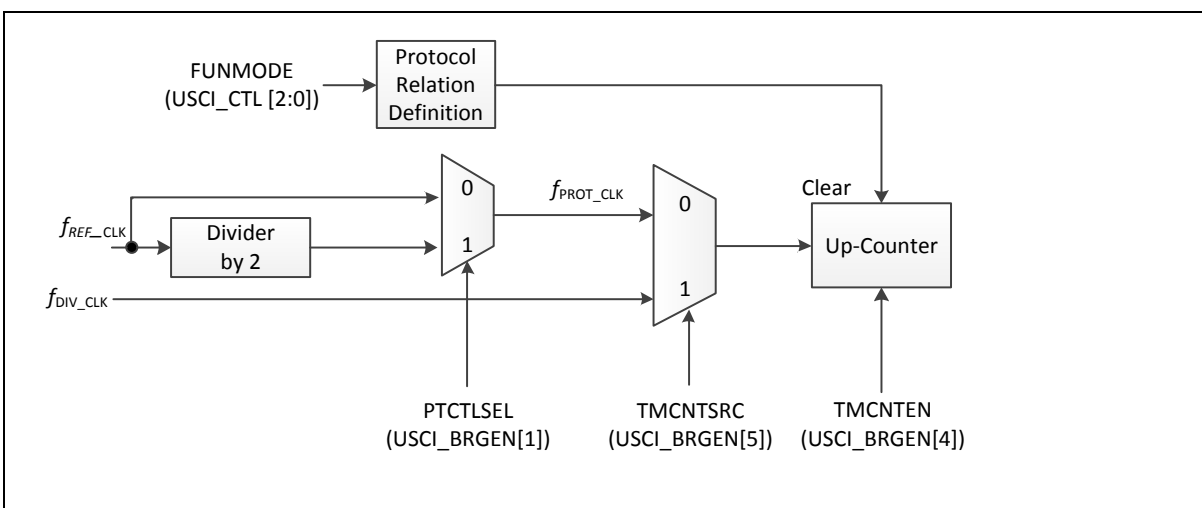


图 6.11-10 时序测量计数器框图

时序测量计数器用于超时功能或者波特率自动识别机制，其功能取决于如下选择的协议：

- **UART:** 时序测量计数器用于波特率自动检测
- **SPI:** 时序测量计数器用于计算从机超时周期
- **I²C:** 时序测量计数器用于提示时钟超时周期

采样时间计数器

采样计数器基于输入频率 f_{SAMP_CLK} ，协同协议相关计数器定义协议的时序，例如移位控制信号或时序，采样计数器可以为特定协议产生时间间隔。采样计数器的周期 f_{PDS_CNT} 取决于输入频率 f_{SAMP_CLK} 及可以编程的预分频值(PDSCNT (USCI_BRGEN [9:8]))。采样定时器的意义取决于所选的协议，详细请参考协议相关章节

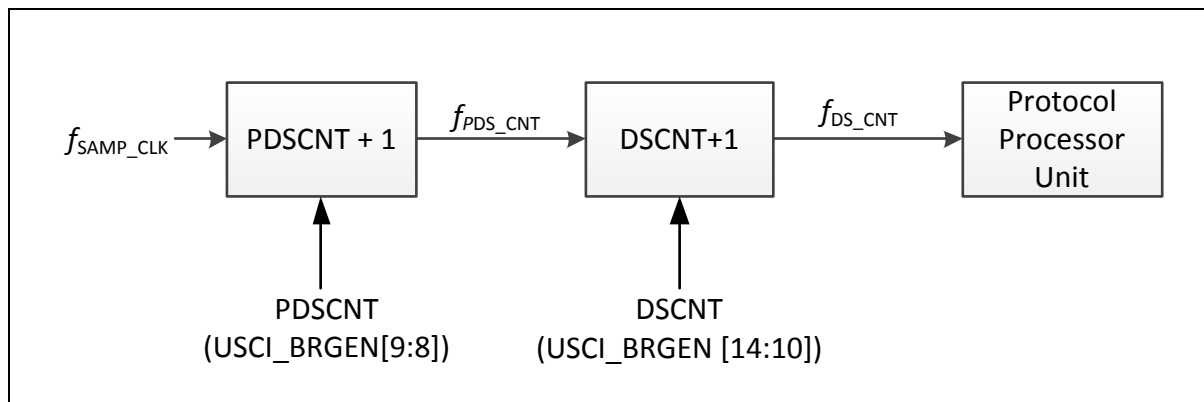


图 6.11-11 采样时间计数器

6.11.4.5 数据传输时间及中断

数据传送事件是基于发送或接收一个数据的事件，相关的指示标志位于USCI_PROTSTS。所有事件可以单独使能产生中断，如果FUNMODE (USCI_CTL [2:0])=0，USCI被禁止，当FUNMODE (USCI_CTL [2:0])设置为一种协议接口，其内部状态将被所选择协议的硬件逻辑控制

- 发送开始中断事件表示一个数据开始被发送了：

当数据被加载到发送移位寄存器后会产生发送开始中断事件，通过TXSTIF (USCI_PROTSTS [1])标志事件产生，如果中断使能，将会触发发送开始中断

- 发送结束中断事件表示一个数据发送完成:

当前移位寄存器中的数据发送完成后触发“发送完成中断”，由标志TXENDIF (USCI_PROTSTS [2])表示该中断，如果中断使能，将会触发发送结束中断，当前数据传送的移位控制设定（数据长度，移位方向等）内部“冻结”时，也会触发该事件。在UART及I2C模式下，数据发送的有效性，可以通过TXEMPTY (USCI_BUFSTS [8])标志及相关发送结束中断来判断

- 接收开始事件表示一个数据开始被接收:

当接收使能并且接收到第一个数据位时，接收开始事件被触发，RXSTIF(USCI_PROTSTS [3])标志该事件，如果中断使能则会触发相应接收开始中断

- 接收事件表示一个数据完成接收:

如果接收缓存中收到一个新的数据，接收事件被触发，通过RXENDIF (USCI_PROTSTS [4])标志该事件，如果相应中断被使能则会触发中断

- 数据丢失事件表示最新接收的数据被丢失了:

如果寄存器USCI_RXDAT (RX_BUF0 或 RX_BUF1上次接收的数据)中的数据没有被及时读走，新的数据将被丢失，并触发该事件，RXOVIF (USCI_BUFSTS[3])标志该事件，如果中断被使能则会触发相应中断

事件和中断的架构如图 6.11-12

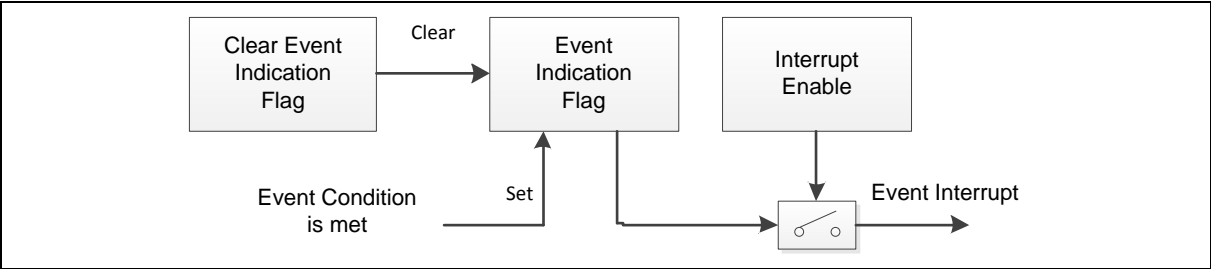


图 6.11-12 事件和中断架构

中断可以通过寄存器RXENDIEN, RXSTIEN, TXENDIEN, 及USCI_INTEN [4:1]的TXSTIEN位使能，这些事件包括接收结束中断事件，接收开始中断事件，发送结束中断事件，发送开始中断事件，协议相关的中断在各协议中断使能寄存器中定义

事件在满足特定条件时被触发，事件指示标志自动置位直到被软件清除，如果相应中断被使能则该事件会触发相应中断

USCI寄存器、位及位域表示数据传输事件并控制相应中断，如下表 6.11-3

事件	指示标志	清除指示标志	中断使能
发送开始中断事件	TXSTIF (USCI_PROTSTS [1])	通过向USCI_PROTSTS相应的中断位写1清0	TXSTIEN (USCI_INTEN [1])
发送结束中断事件	TXENDIF (USCI_PROTSTS [2])		TXENDIEN (USCI_INTEN [2])
接收开始中断事件	RXSTIF (USCI_PROTSTS [3])		RXSTIEN (USCI_INTEN [3])
接收结束中断事件	RXENDIF (USCI_PROTSTS [4])		RXENDIEN (USCI_INTEN [4])

表 6.11-3 数据发送事件和中断处理

6.11.4.6 协议事件及中断

与协议相关的事件在各协议章节描述，相应的指示标志在寄存器USCI_PROTSTS，所有事件可以单独使能并触发中断

事件	指示标志	清除指示标志	中断使能
UART 模式协议事件	USCI_PROTSTS [17:16] 和 USCI_PROTSTS [11:5]	软件写1到寄存器USCI_PROTSTS相应位清除中断.	USCI_PROTIEN[2:1]
SPI 模式协议事件	USCI_PROTSTS [9:8],		USCI_PROTIEN

	USCI_PROTSTS [6:5]		[3:0]
I2C 模式协议事件	USCI_PROTSTS [13:8], USCI_PROTSTS [5]		USCI_PROTIEN [6:0]

表 6.11-4 协议事件及中断处理

6.11.4.7 唤醒

协议相关唤醒功能的信息位于唤醒控制寄存器(USCI_WKCTL)及唤醒状态寄存器(USCI_WKSTS)，所有协议共用这些寄存器，所以这些寄存器中的位在不同协议中代表不同的意思

6.12 USCI – UART 模式

6.12.1 概述

UART功能包括接收/发送异步数据帧，接收数据时UART是串并转换，发送数据时是并串转换功能，接收器和发送器是独立的，接收和发送数据帧可以在不同时间开始。

UART控制器也提供LIN功能，数据输入可以唤醒系统。

6.12.2 特性

- 支持两个数据接收缓冲和一个数据发送缓冲
- 支持可编程波特率发生器
- 支持9位数据发送
- 支持LIN 功能
- 通过波特率产生器内置的捕获功能可以实现波特率检测
- 支持唤醒功能

6.12.3 框图

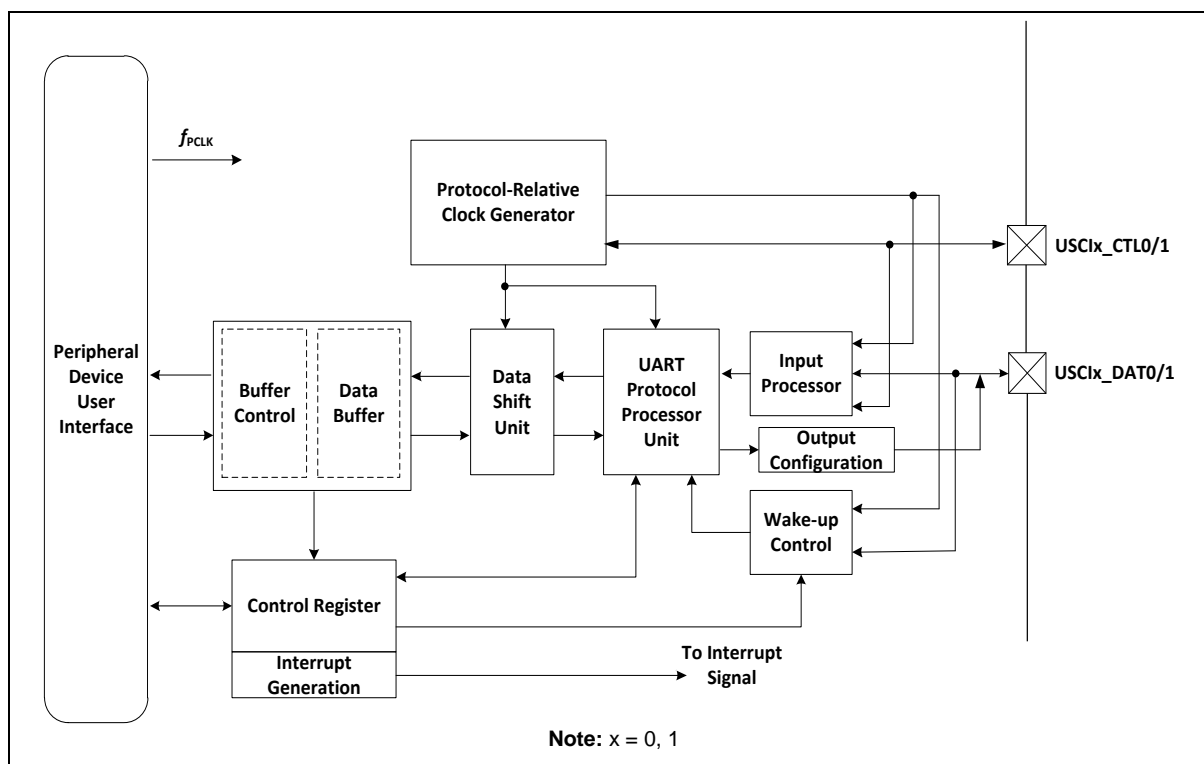


图 6.12-1 USCI - UART 模式框图

注: nCTS 和 nRTS 功能不支持.

6.12.4 基本配置

USC10 用于UART模式的基本配置如下：

- SYS_GPA_MFP, SYS_GPC_MFP, SYS_GPD_MFP 寄存器里配置USC10管脚。
- USC10CKEN (CLK_APBCLK[24])使能USC10外设时钟
- USC10RST (SYS_IPRST1[24])复位USC10控制器。

USC11 用于UART模式的基本配置如下：

- SYS_GPA_MFP, SYS_GPC_MFP, SYS_GPD_MFP 寄存器里配置USC11管脚。
- USC11CKEN (CLK_APBCLK[25])使能USC11外设时钟
- USC11RST (SYS_IPRST1[25])复位USC11控制器

6.12.5 功能描述

6.12.5.1 USC1 通用功能描述

请参看6.11.4章节的详细描述

6.12.5.2 信号描述

UART的连接是收/发端的TX, RX分别相连；接收信号（RXD）对应到USC1x_DAT0寄存器，发送信号(TXD)对应到USC1x_DAT1寄存器

全双工通信要求收发都要有独立的连接 如图 6.12-2 所示，UART模块A和UART模块B全双工连接

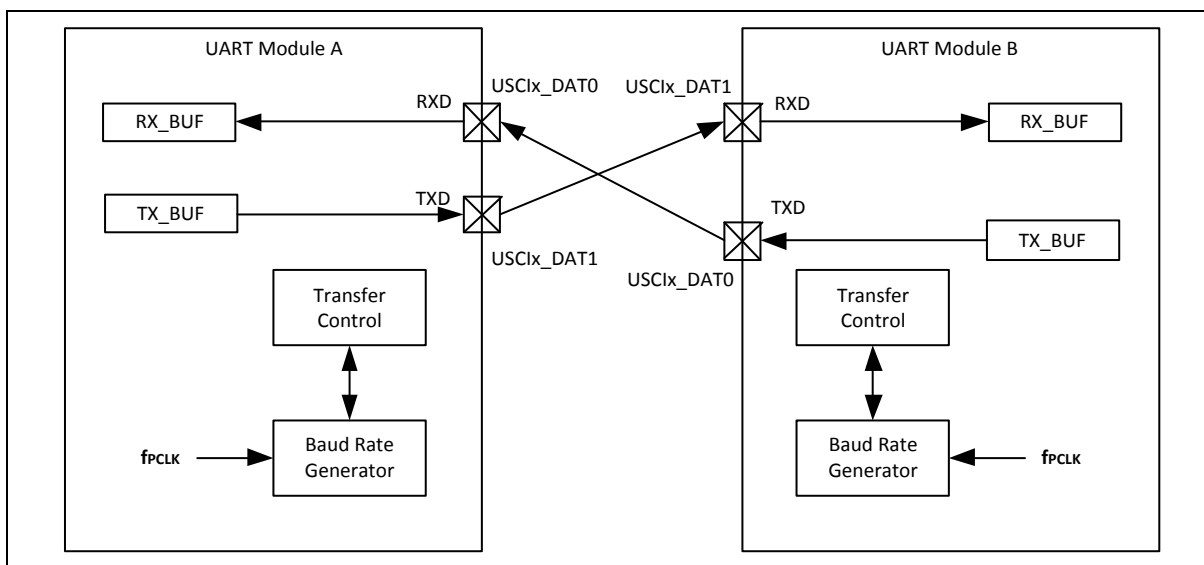


图 6.12-2 UART 全双工通信连接

输入信号

对于UART协议，表 6.12-1. 列出了输入信号，每个输入信号都有输入处理器处理，例如信号反向选择控制，数字输入滤波等

协议选择		UART
Control Input	USCIx_CTL0	X
	USCIx_CTL1	X
Data Input(s)	USCIx_DAT0	RX
	USCIx_DAT1	X

表 6.12-1 UART 协议输入信号

输出信号

对于UART 协议相关的每个输出信号都是有效的。实际输出的信号依据协议的选择，根据协议的意义来分类，如下表。

协议选择		UART
控制输出	USCI_CTL0	X
	USCI_CTL1	X
数据输出	USCI_DAT0	X
	USCI_DAT1	TX

表 6.12-2 不同协议的输出信号

6.12.5.3 帧格式

一个标准的UART帧如图 6.12-3. 它的组成如下:

- 电平为高的空闲时间.
- 电平为0的帧起始位 (SOF).
- 6~13 数据位
- 一个校验位 (P),可编程设置奇，偶校验或无校验功能
- 电平为高的1位或2位停止位.

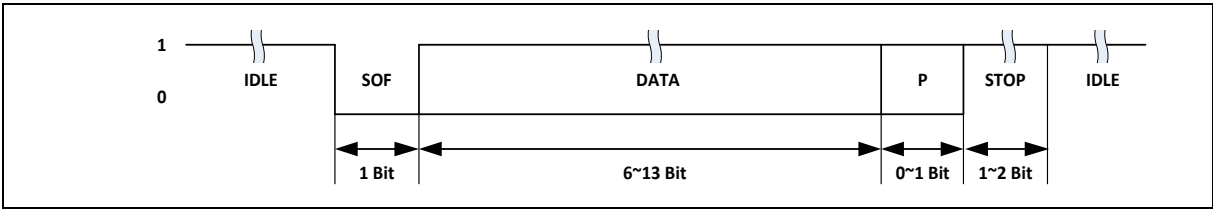


图 6.12-3 UART 标准帧格式

UART协议状态机会自动处理协议位(SOF, P, STOP)，不会显示在接收或发送数据流中

起始位

接收器在空闲状态或上一个数据停止位采样完成后，会检测USCIx_DAT0信号的下降沿，一旦检测到下降沿就会产生SOF信号，为了增强抗干扰能力，检测到SOF位后如果在采样点检测到高电平，那么此SOF认为是干扰，接收器重新进入空闲模式

数据域

数据域的长度可以通过寄存器DWIDTH (UART_LINECTL[11:8])设定，可以设置为6-13位

注意：UART模式中数据传输顺序是LSB优先，可以通过LSB (UART_LINECTL[0])=1 设定

校验位

UART基于数据帧结构，支持收发数据帧的校验位功能，校验位可以通过PARITYEN (UART_PROTCTL[1])使能，通过EVENPARITY (UART_PROTCTL[2])设置奇偶校验类型，校验对收发数据同时有效，所有通信参与方都要遵循相同的校验类型

如果校验功能使能，数据最后一个位发送完成后，发送器自动计算并发送校验位，接收器会将收到的校验位跟自己计算的校验位作比较。协议状态寄存器(UART_PROTSTS)会监控校验结果及帧检测（停止位），该寄存器位(FRMERR, PARITYERR)代表协议相关状态及错误

停止位

每个UART数据帧包含1或2个高电平的停止位（与空闲状态的电平相同），停止位的数量可以通过STOPB (UART_PROTCTL[0])设定，停止位后可以开始一个新的数据发送

传输状态指示

RXBUSY (UART_PROTSTS[10]) 显示了接收器的状态。

接收器的状态可以通过RXBUSY位来监控，RXBUSY位在从起始位到结束位的整帧数据接收过程中保持为1

6.12.5.4 操作模式

操作UART协议，下列问题必须考虑：

设置UART 模式

设置FUNMODEOE (UART_CTL[2:0]) =0x2，PROTEN (UART_PROTCTL [31])= 1 使能UART模式；注意：改变协议前FUNMODE必须设置为0，建议在配置好所有参数后再使能UART协议

管脚连接

USCI_DAT0管脚用来接收UART数据（Rx），通过UART_DATIN0配置输入信号的特性，建议设置EDGEDET (UART_DATIN0[4:3]) = 0x2 检测开始位

USCI_DAT1管脚用于发送UART数据（Tx），通过UART_LINECTL设置输出信号的特性

位时序配置

选定所需的波特率后就确定了波特率发生器及位时序，详情请参考章节6.12.5.

帧格式配置

根据应用的需要，通过UART_LINECTL 及 UART_PROTCTL寄存器来设定数据长度，停止位数，校验模式；数据输入输出信号可以被反转，数据发送顺序LSB 优先，需要设定LSB (UART_LINECTL[0]) =1

6.12.5.5 位时序

在UART模式下，每个数据位被分成若干采样时间，以便根据应用需要来调整采样点，每个位的采样时间个数通过DSCNT (UART_BRGEN[14:10])来设定，每个位的采样时间长度通过PDSCNT (UART_BRGEN[9:8])设定

在图 6.12-4所示的情况下，一个位被分为16个采样时间DSCNT(UART_BRGEN[14:10]) = 15，建议每个数据位的采样时间个数大于4

每个位的采样点固定在1/2采样时间个数，采样数据会取平均

发送接收会用同样的位时序设定（采样时间个数），因为他们使用相同的硬件电路

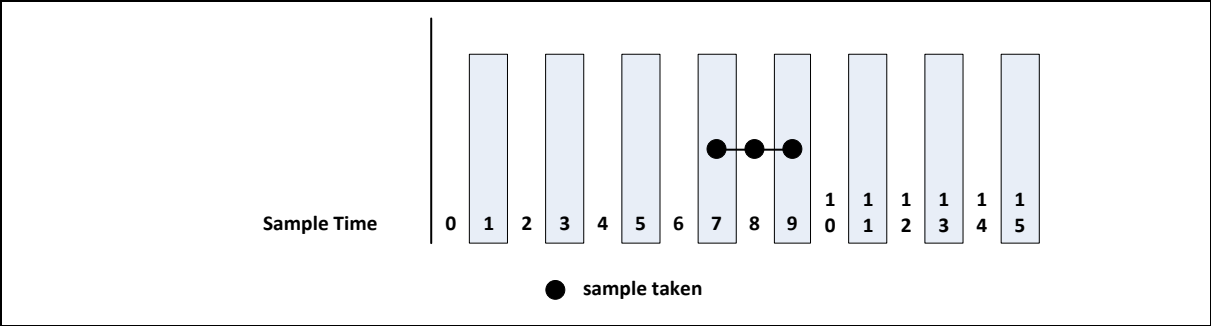


图 6.12-4 UART 位时序 (数据采样个数)

6.12.5.6 波特率产生

UART的波特率 f_{UART} 取决于每个数据位中有多少个采样时间，波特率的设定只能在收发器处于空闲状态时改变，位RCLKSEL, SPCLKSEL, PDSCNT, 及 DSCNT定义了波特率设置：

RCLKSEL (UART_BRGEN [0])

定义了输入时钟 f_{REF_CLK}

SPCLKSEL (UART_BRGEN[3:2])

定义了多个采样时钟源 f_{SAMP_CLK}

PDSCNT (UART_BRGEN [9:8])

定义了一个采样时间的长度(由 f_{REF_CLK} 除以 1, 2, 3, 或 4获得)

DSCNT (UART_BRGEN [14:10])

定义一个数据位的采样时间个数

如果RCLKSEL = 0 (fREF_CLK = fPCLK), PTCLKSEL = 0 (fPROT_CLK = fREF_CLK) 和 SPCLKSEL = 0x0 (fSAMP_CLK = fDIV_CLK)那么波特率为:

f_{UART} = f_{REF_CLK} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}

要想产生慢一些的频率，通过PTCLKSEL = 1 (fPROT_CLK = fREF_CLK2)可以使时钟除频，得到:

f_{UART} = \frac{f_{REF_CLK}}{2} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}

如果SPCLKSEL = 0x2 (fSAMP_CLK = fSCLK), RCLKSEL = 0 (fREF_CLK = fPCLK), PTCLKSEL = 0 (fPROT_CLK = fREF_CLK), 那么波特率为:

f_{UART} = f_{REF_CLK} \times \frac{1}{CLKDIV + 1} \times \frac{1}{2} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}

UART波特率设定会有一些误差，表 6.12-3 列出了使用不同晶振的误差百分比:

HCLK 时钟源	PCLK 时钟源	预期波特率	CLKDIV (UART_BRGEN[25:16])	DSCNT (UART_BRGEN[14:10])	PDSCNT	实际波特率	误差率
HXT12M	12M (HCLK)	115200	0xC	0x7	0x0	115384	1.6%
HIRC/2, 24M	24M (HCLK)	9600	0xF9	0x9	0x0	9600	0%
HIRC48M	48M (HCLK)	115200	0xC	0xF	0x1	115384	1.6%

表 6.12-3 波特率与晶振的关系

注: SPCLKSEL = 0x0, PTCLKSEL = 0, RCLKSEL = 0

6.12.5.7 自动波特率检测

UART支持自动波特率检测功能，可以识别输入信号(USC1x_DAT0)的波特率，并自动调整波特率时钟寄存器CLKDIV (UART_BRGEN[25:16])，以匹配输入信号的时序。根据时序测量计数器部分的描述，时序测量计数器会测量输入信号(USC1x_DAT0)并把计数值在每个信号下降沿保存到BRDETIV (UART_PROTCTL [24:16])中

当ABREN (PROTOCOL[6])被使能，模板数据0x55用来做波特率自动识别的信号，输入信号下降沿触发波特率计数开始，在下一个下降沿将计数的值存储到BRDETIV (UART_PROTCTL [24:16])。建议使用fDIV_CLK (TMCNTSRC (UART_BRGENC[5]) =1)作为计数时钟源

波特率自动检测完成时（输入信号0x55的第四个下降沿），CLKDIV (UART_BRGEN[25:16])会根据BRDETIV (UART_PROTCTL [25:16])的值自动修正，如果用户希望正确接收下一个数据帧，最好将CLKDIV (UART_BRGEN[25:16]) 和 DSCNT (UART_BRGEN[14:10])设为相同的值，该值应该处于0xF到0x5之间，因为DSCNT定义了每个位的采样次数而此时PDSCNT (UART_BRGEN[9:8])=0x0

波特率自动检测过程中，ABRDIF (UART_PROTSTS[9])和BRDETIV (UART_PROTCTL [24:16])在输入信号的每个下降沿都会更新，模板数据0x55不会被接收到输入缓冲中。ABREN位将在输入信号的第四个下降沿被硬件清零，用户可以通过读ABREN的状态了解自动波特率检测是否

完成。

如果CLKDIV 和 DSCNT在自动波特率检测过程中没有被设置为相同的值， 需要用户根据BRDETIV 和 CLKDIV去计算合适的值

如果输入信号的波特率很慢， 时序测量计数器不能计算正确的信号周期， ABERRSTS (UART_PROTSTS[11])会显示一个错误状态， 这时用户需要修改CLKDIV的值并要求主机重新发送0x55模板数据

根据时序测量计数器的限制， BRDETIV能够支持的最大计数值为0x1FE

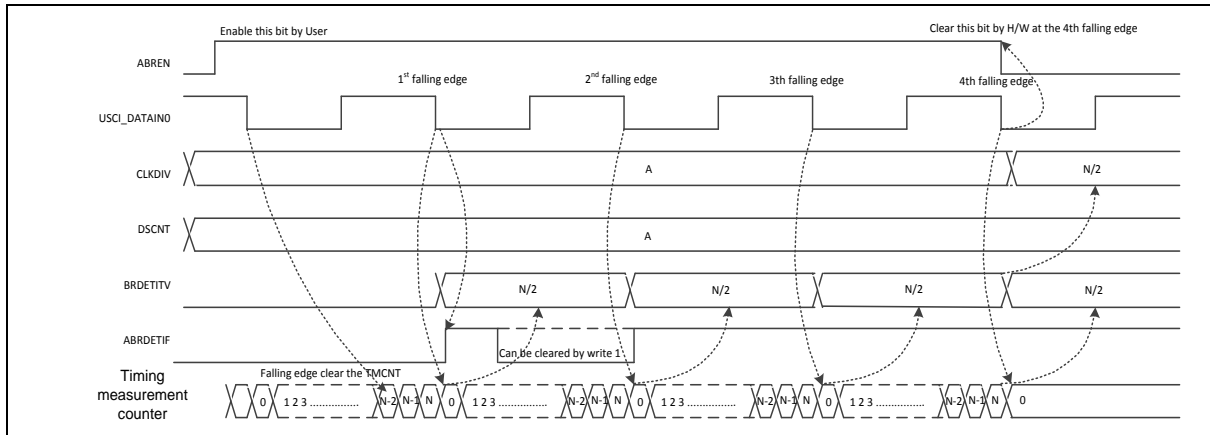


图 6.12-5 UART 自动波特率控制

6.12.5.8 支持硬件LIN

支持LIN(内部局域网)协议，主机时LINBRKEN(UART_PROTCTL[7])位置1，从机该位清0，固定8位数据位长度设置（DWIDTH=8）。两者同时使用时，校验功能需关闭(PARITYEN(UART_PROTCTL[1])=0),数据传输顺序为LSB优先(LSB(UART_LINECTL[0])=1)和1个停止位(STOPB(UART_PROTCTL[0])=0)。

在UART模式下LIN数据交换协议包含的几个特征都可以处理。每一个LIN特征信号都代表一个UART帧。LIN总线是单主机信号和多从机信号的主-从总线系统（更多的定义请参考LIN官方规格书）

一个完整的LIN帧包含如下特征标志：

同步断点

在开始一个新帧时主机发送一个同步断点信号，它包含连续13个位时间的低电位，接着至少一个位时间的高电位（相应的1个停止位）。因此，LINBRKEN(UART_PROTCTL[7])被置1（SOF帧接在12低电平的数据位后面）

从机通过同步断点检测将测到连续的11个位的低电平。如果检测到那么BRKEDTIF(UART_PROTSTS[8])置1，并且产生中断。另外，接收缓冲中数据为0并产生数据格式错误信号。

如果需要从机适配主机波特率，那么在下一个标志开始前置ABREN (UART_PROTCTL[6])= 1使

能下降沿波特率测量。

同步位

在主机写0x55到UUART_TXDAT中后主机会发送一个同步位标志。从机收到这个标志不会有任何进一步的动作（会丢掉它），或者用下降沿来测量波特率。**ABREN=1**时收到同步位被用于捕获一个定时计数器的值。在一帧完成后可以读出捕获的有效值。这个标志之后，波特率测量可以取消（**ABREN=0**），然后修正波特率时钟除频器CLKDIV(UUART_BRGEN[25:16])。更多细节描述请参考6.12.5.6 章节。

其他标志

LIN 帧的其他标志可以用UART数据帧处理不需要特殊的动作 。
请注意在UART接收端进行波特率测量的时候，UART发送端仍然在进行发送。

6.12.5.9 唤醒功能

UART模式下的USCI控制器支持唤醒功能，唤醒源包括接收到数据唤醒，下面详细描述接收数据唤醒源：

接收数据唤醒

当系统进入掉电模式，并且WKEN (UUART_WKCTL [0]) 和 DATWKEN (UUART_PROTCTL[9])都被置位，数据输入管脚的电平变化会唤醒系统。如果希望唤醒后能够收到这个唤醒数据，必须设置WAKECNT (UUART_PROTCTL[14:11])寄存器，它表示系统唤醒时，数据起始位期间有多少个fPDS_CLK时钟周期

- 注意1: WKF (UUART_WKSTS[0])标志清零的那一刻，硬件计数器的值会加载到WAKECNT中，所以用户在系统唤醒后要立刻清除该标志，以保证计数值最接近系统唤醒时间
- 注意2: 如果想接收到唤醒数据，需要考虑所选择时钟的稳定时间与波特率的关系（例如：HXT的稳定时间是4096个时钟周期）

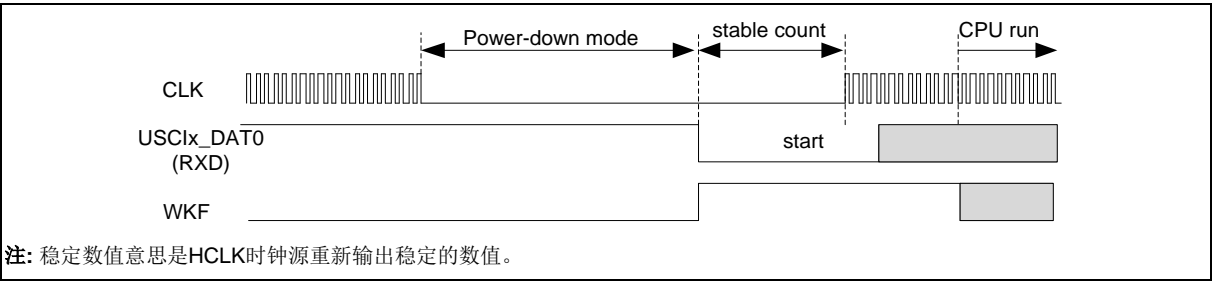


图 6.12-6 接收收据唤醒

6.12.5.10 中断事件

协议中断事件

UART模式下会产生下列协议相关事件，并会导致产生中断

请注意UUART_PROTSTS中的标志位不会自动清零，要求用户软件清零，才能接收下一个事件

接收线状态

协议相关的错误标志FRMERR (UUART_PROTSTS[6]) 及 PARITYERR (UUART_PROTSTS[5])，在每次接收数据时都会在其状态寄存器中更新

UART模式下，奇偶校验错误标志为零，表示没有奇偶校验错误，帧错误标志为零表示没有帧错误，这些标志在每个数据帧都会更新

在收到的数据，校验位，停止位都为零时，Break错误标志(BREAK (UUART_PROTSTS[7]))置位

当产生校验错误，帧错误及break错误标志时（BREAK, FRMERR, PARITYERR (UUART_PROTSTS[7:5])），如果RLSIEN (UUART_PROTIEN[2])=1，则会产生相关错误中断

自动波特率检测

自动波特率中断标志ABRDETIF (UUART_PROTSTS [9])=1，表示时序测量计数器捕获到2个数据位时长的计数，如果ABRIEN (UUART_PROTIEN[1])=1 控制器会产生中断

自动波特率检测功能，在接收信号的的第一个下降沿触发，并开始计数一直计到下一个下降沿一帧数据传结束。在一帧数据传输结束后，用户将捕获的数据除以2得到的就是每个数据位周期内的采样时间个数。用户可以读出BRDETITV (UUART_PROTCTL[24:16]) 的值然后写入波特率发生器寄存器CLKDIV (UUART_BRGEN[25:16])中。

同步断点检测

在LIN 网络中该中断表明收到了一个同步断点特征标志（开始一个LIN 帧）。如果BRKIEN(UUART_PROTIEN[0])也被置1，控制器将产生一个中断。

数据传输中断处理

数据传输中断是与UART帧处理有关的事件

发送开始中断

发送数据的起始位发出后TXSTIF (UUART_PROTSTS[1])置1，在缓存模式，这是可以向UUART_TXDAT写入数据的最早时间点，如果此时TXSTIEN (UUART_INTEN[1])=1，控制器将会产生中断

发送结束中断

该中断表示发送缓存中的数据完全发送完毕了，数据结束位发送完成后TXENDIF (UUART_PROTSTS[2])=1，如果此时TXENDIEN (UUART_INTEN[2])=1，将会产生中断

接收开始中断

当控制器接收到数据起始位的采样点，RXSTIF (UUART_PROTSTS[3])置位，如果此时RXSTIEN (UUART_INTEN[3])=1 控制器将产生中断

帧接收结束中断

该中断表示接收器完整接收到一个数据帧，RXENDIF (UART_PROTSTS[4])置位，如果此时RXENDIEN (UART_INTEN[4])=1，控制器会产生中断

6.12.5.11 编程示例

下列步骤用来配置UART协议及数据传输：

1. 置 FUNMODE (UART_CTL[2:0]) 为 0x2 选择UART 协议.
2. 写波特率发生器寄存器UART_BRGEN 选择需要的波特率.
 1. 置 SPCLKSEL (UART_BRGEN[3:2]), PTCLKSEL (UART_BRGEN[1]) 和 RCLKSEL (UART_BRGEN[0]) 选择时钟源.
 2. 配置 CLKDIV (UART_BRGEN[25:16]), DSCNT (UART_BRGEN[14:10]) 和 PDSCNT (UART_BRGEN[9:8]) 设定波特率除频.
3. 写线性控制器 UART_LINECTL 和协议控制寄存器UART_PROTCTL 配置传输数据的格式和UART 协议设定.
 1. 设定 DWIDTH (UART_LINECTL[11:8])数据域的长度.
 2. 通过设定EVENPARITY (UART_PROTCTL[2]) 和 PARITYEN (UART_PROTCTL[1]) 使能校验位和选择校验类型 .
 3. 通过设定 STOPB (UART_PROTCTL[0])配置停止位长度.
 4. 使能 LSB (UART_LINECTL[0]) 选择UART协议LSB优先传输
 5. 设置 EDGEDET (UART_DATIN0[4:3]) 为 0x2 用于选择下降沿作为起始为检测
4. 设置 PROTEN (UART_PROTCTL[31]) 为 1 使能UART 协议
5. 发送和接收数据.
 - 写发送数据寄存器UART_TXDAT 用于发送数据.
 - 等待 TXSTIF(UART_PROTSTS[1]) 置 1 后，用户才可以写下一笔数据到 UART_TXDAT.
 - 通过设定 EVENPARITY (UART_PROTCTL[2]) 和 PARITYEN (UART_PROTCTL[1]) 使能校验位和选择校验类型.
 - 当 TXENDIF(UART_PROTSTS[2])置1时，发送缓冲为空，最后一笔数据的停止位发送完毕
 - 如果RXENDIF(UART_PROTSTS[4]) 置1，接收完一帧数据。用户可以通过读接收数据寄存器UART_RXDAT 来获取数据。

6.12.6 寄存器映射

R: 只读, W: 只写, R/W: 读写

寄存器	偏移	R/W	描述	复位值
UUART_CTL x = 0, 1	UUARTx_BA+0x00	R/W	USCI 控制寄存器	0x0000_0000
UUART_INTEN x = 0, 1	UUARTx_BA+0x04	R/W	USCI 中断使能寄存器	0x0000_0000
UUART_BRGEN x = 0, 1	UUARTx_BA+0x08	R/W	USCI 波特率发生器寄存器	0x0000_3C00
UUART_DATIN0 x = 0, 1	UUARTx_BA+0x10	R/W	USCI 输入数据信号配置寄存器0	0x0000_0000
UUART_CTLIN0 x = 0, 1	UUARTx_BA+0x20	R/W	USCI 输入控制信号配置寄存器0	0x0000_0000
UUART_CLKIN x = 0, 1	UUARTx_BA+0x28	R/W	USCI 输入时钟信号配置寄存器	0x0000_0000
UUART_LINECTL x = 0, 1	UUARTx_BA+0x2C	R/W	USCI 线性控制寄存器	0x0000_0000
UUART_TXDAT x = 0, 1	UUARTx_BA+0x30	W	USCI 发送数据寄存器	0x0000_0000
UUART_RXDAT x = 0, 1	UUARTx_BA+0x34	R	USCI 接收数据寄存器	0x0000_0000
UUART_BUFCTL x = 0, 1	UUARTx_BA+0x38	R/W	USCI 发送/接收缓冲控制寄存器	0x0000_0000
UUART_BUFSTS x = 0, 1	UUARTx_BA+0x3C	R	USCI 发送/接收缓冲状态寄存器	0x0000_0101
UUART_WKCTL x = 0, 1	UUARTx_BA+0x54	R/W	USCI 唤醒控制寄存器	0x0000_0000
UUART_WKSTS x = 0, 1	UUARTx_BA+0x58	R/W	USCI 唤醒状态寄存器	0x0000_0000
UUART_PROTCTL x = 0, 1	UUARTx_BA+0x5C	R/W	USCI 协议控制寄存器	0x0000_0000
UUART_PROTIEN x = 0, 1	UUARTx_BA+0x60	R/W	USCI 协议中断使能寄存器	0x0000_0000
UUART_PROTSTS x = 0, 1	UUARTx_BA+0x64	R/W	USCI 协议状态寄存器	0x0000_0000

6.12.7 寄存器描述

USCI 控制寄存器(UUART_CTL)

寄存器	偏移	R/W	描述	复位值
UUART_CTL x = 0, 1	UUARTx_BA+0x00	R/W	USCI 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					FUNMODE		

位	描述	
[31:3]	Reserved	保留.
[2:0]	FUNMODE	<p>功能模式</p> <p>此位域选择USCI控制器的协议，非法的设置会关闭USCI功能。两种协议切换时，需要先关闭USCI功能再切换新的协议，写0x0到FUNMODE复位USCI</p> <p>0x0 = USCI 禁用. 所有协议状态控制机为空闲状态</p> <p>0x1 = SPI 协议</p> <p>0x2 = UART 协议</p> <p>0x4 = I2C 协议</p> <p>注: 其它位组合保留</p>

USCI 中断使能寄存器(UUART_INTEN)

寄存器	偏移	R/W	描述	复位值
UUART_INTEN x = 0, 1	UUARTx_BA+0x04	R/W	USCI 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							
7	6	5	4	3	2	1	0
Reserved			RXENDIEN	RXSTIEN	TXENDIEN	TXSTIEN	Reserved

位	描述	
[31:5]	Reserved	保留
[4]	RXENDIEN	接收结束中断使能位 该位使能接收数据完成中断 0 = 接收结束中断禁用 1 = 接收结束中断使能
[3]	RXSTIEN	接收开始中断使能位 该位使能接收开始中断 0 = 接收开始中断禁用 1 = 接收开始中断使能
[2]	TXENDIEN	发送结束中断使能位 该位使能发送结束中断 0 = 发送完成中断禁用 1 = 发送完成中断使能
[1]	TXSTIEN	发送开始中断使能位 该位使能发送开始中断 0 = 发送开始中断禁用 1 = 发送开始中断使能
[0]	Reserved	保留。

USCI 波特率发生器寄存器(UUART_BRGEN)

寄存器	偏移	R/W	描述	复位值
UUART_BRGEN x = 0, 1	UUARTx_BA+0x08	R/W	USCI 波特率发生器寄存器	0x0000_3C00

31	30	29	28	27	26	25	24
Reserved						CLKDIV	
23	22	21	20	19	18	17	16
CLKDIV							
15	14	13	12	11	10	9	8
Reserved	DSCNT					PDSCNT	
7	6	5	4	3	2	1	0
Reserved		TMCNTSRC	TMCNTEN	SPCLKSEL		PTCLKSEL	RCLKSEL

位	描述	
[31:26]	Reserved	保留
[25:16]	CLKDIV	时钟除频 此位定义了 $f_{\text{PROT_CLK}}$ 和 $f_{\text{DIV_CLK}}$ 之间的比例关系($f_{\text{DIV_CLK}} = f_{\text{PROT_CLK}} / (\text{CLKDIV} + 1)$) 注意: 在UART模式下使能自动波特率检测(ABREN(UUART_PROTCTL[6])), 硬件会在输入模板数据0x55的第四个下降沿自动更新此位域, 更新后的值是第5位, 第6位的平均时间, 用户可以使用更新后的CLKDIV及BRDETIV (UUART_PROTCTL[24:16])值来计算精确的波特率
[15]	Reserved	保留.
[14:10]	DSCNT	采样计数器分母 此位域定义采样时钟($f_{\text{SAMP_CLK}}$)的除频率. $f_{\text{DS_CNT}} = f_{\text{PDS_CNT}} / (\text{DSCNT} + 1)$. 注意: UART模式下DSCNT的值最大为0xF, 建议设置大于4的值, 以保证采样数据的准确
[9:8]	PDSCNT	采样计数器预分频 此位域定义了采样时钟 $f_{\text{SAMP_CLK}}$ 的除频比例 $f_{\text{PDS_CNT}} = f_{\text{SAMP_CLK}} / (\text{PDSCNT} + 1)$
[7:6]	Reserved	保留.
[5]	TMCNTSRC	时序测量计数器时钟源选择 0 = 时序测量计数器时钟源为 $f_{\text{PROT_CLK}}$. 1 = 时序测量计数器时钟源为 $f_{\text{DIV_CLK}}$.
[4]	TMCNTEN	时序测量计数器使能位 该位使能10位时序测量计数器. 0 = 时序测量计数器禁用. 1 = 时序测量计数器使能.

[3:2]	SPCLKSEL	采样时钟源选择 该位用于协议处理器的采样时钟源($f_{\text{SAMP_CLK}}$)的选择. $00 = f_{\text{SAMP_CLK}} = f_{\text{DIV_CLK}}$. $01 = f_{\text{SAMP_CLK}} = f_{\text{PROT_CLK}}$. $10 = f_{\text{SAMP_CLK}} = f_{\text{SCLK}}$. $11 = f_{\text{SAMP_CLK}} = f_{\text{REF_CLK}}$.
[1]	PTCLKSEL	协议时钟源选择 该位用于选择协议时钟($f_{\text{PROT_CLK}}$)的选择. $0 =$ 参考时钟源 $f_{\text{REF_CLK}}$. $1 = f_{\text{REF_CLK}2}$ ($f_{\text{REF_CLK}}$ 除以2).
[0]	RCLKSEL	参考时钟源选择 该位用于选择参考时钟源($f_{\text{REF_CLK}}$). $0 =$ 外设时钟 f_{PCLK} . $1 =$ HXT/LXT.

USCI 输入数据信号配置(UUART_DATIN0)

寄存器	偏移	R/W	描述	复位值
UUART_DATIN0 x = 0, 1	UUARTx_BA+0x10	R/W	USCI 输入数据信号配置寄存器0	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			EDGEDET		ININV	Reserved	SYNCSEL

位	描述	
[31:5]	Reserved	保留.
[4:3]	EDGEDET	输入信号边沿检测模式 该位用于决定输入数据信号的上升沿还是下降沿触发事件 00 = 触发事件禁用. 01 = 输入数据信号的上升沿触发事件 10 = 输入数据信号的下降沿触发事件 11 = 输入数据信号的上升下降沿都触发事件 注: 在UART 模式下, 建议该位设置为10
[2]	ININV	输入信号反向选择 该位定义非同步输入信号的是否反转 0 = 非同步输入信号不反转 1 = 非同步输入信号反转
[1]	Reserved	保留.
[0]	SYNCSEL	输入信号同步选择 此位决定非同步信号 (可设置反向) 或者同步信号 (可设置滤波) 作为数据移位单元的输入 0 = 非同步信号作为数据移位单元的输入 1 = 同步信号作为数据移位单元的输入

USCI 输入控制信号配置(UUART_CTLIN0)

寄存器	偏移	R/W	描述	复位值
UUART_CTLIN0 x = 0, 1	UUARTx_BA+0x20	R/W	USCI 输入控制信号配置寄存器0	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					ININV	Reserved	SYNCSEL

位	描述	
[31:3]	Reserved	保留.
[2]	ININV	输入信号反向选择 该位设置非同步输入信号反转是否使能 0 = 非同步输入信号不反转 1 = 非同步输入信号反转
[1]	Reserved	保留.
[0]	SYNCSEL	输入同步信号选择 此位决定非同步信号（可设置反向）或者同步信号（可设置滤波）作为数据移位单元的输入 0 = 非同步信号作为数据移位单元的输入 1 = 同步信号作为数据移位单元的输入

USCI 输入时钟信号配置(UUART_CLKIN)

寄存器	偏移	R/W	描述	复位值
UUART_CLKIN x = 0, 1	UUARTx_BA+0x28	R/W	USCI 输入时钟信号配置寄存器	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							SYNCSEL

位	描述	
[31:1]	Reserved	保留.
[0]	SYNCSEL	输入同步信号选择 此位决定非同步信号（可设置反向）或者同步信号（可设置滤波）作为数据移位单元的输入 0 = 非同步信号作为数据移位单元的输入 1 = 同步信号作为数据移位单元的输入

USCI 线性控制寄存器(UUART_LINECTL)

寄存器	偏移	R/W	描述	复位值
UUART_LINECTL x = 0, 1	UUARTx_BA+0x2C	R/W	USCI 线性控制寄存器	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				DWIDTH			
7	6	5	4	3	2	1	0
CTLOINV	Reserved	DATOINV	Reserved				LSB

位	描述	
[31:12]	Reserved	保留.
[11:8]	DWIDTH	<p>传输数据字长 此位定义收发数据的字长（以位为单位），数据右对齐，USCI支持4到16位字长 0x0: 数据字长16位[15:0]. 0x1: 保留. 0x2: 保留. 0x3: 保留. 0x4: 数据字长4位，位于[3:0]. 0x5: 数据字长5位，位于[4:0]. ... 0xF: 数据字长15位，位于[14:0]. 注意: UART协议洗数据字长可配置为6~13 位</p>
[7]	CTLOINV	<p>控制信号输出反向选择 此位定义了内部控制信号与输出控制信号的关系 0 = 不反转 1 = 控制信号反向后再发出去 注意:在UART协议中，控制信号指的是nRTS</p>
[6]	Reserved	保留.
[5]	DATOINV	<p>数据输出反向选择 0 = USCIx_DAT1的值与移位寄存器中的值相同 1 = USCIx_DAT1是移位寄存器的值取反</p>
[4:1]	Reserved	保留.
[0]	LSB	<p>LSB优先发送选择 0 = MSB优先 1 = LSB优先</p>

USCI 发送数据寄存器(UUART_TXDAT)

寄存器	偏移	R/W	描述	复位值
UUART_TXDAT x = 0, 1	UUARTx_BA+0x30	W	USCI 发送数据寄存器	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
TXDAT							
7	6	5	4	3	2	1	0
TXDAT							

位	描述	
[31:16]	Reserved	保留.
[15:0]	TXDAT	发送数据 软件将要发送的16位数据写入此寄存器

USCI 接收数据寄存器(UUART_RXDAT)

寄存器	偏移	R/W	描述	复位值
UUART_RXDAT x = 0, 1	UUARTx_BA+0x34	R	USCI 接收数据寄存器	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
RXDAT							
7	6	5	4	3	2	1	0
RXDAT							

位	描述	
[31:16]	Reserved	保留
[15:0]	RXDAT	接收数据 此寄存器保存收到的数据 注意: RXDAT[15:13]显示与BREAK, FRMERR 及 PARITYERR (UUART_PROTSTS[7:5]). 同样的状态

USCI 发送/接收缓冲控制寄存器 (UART_BUFCTL)

寄存器	偏移	R/W	描述	复位值
UART_BUFCTLx = 0, 1	UARTx_BA+0x38	R/W	USCI 发送/接收缓冲控制寄存器	0x0000_0000

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

位	描述	
[31:18]	Reserved	保留
[17]	RXRST	接收复位 0 = 无动作. 1 = 复位与接收相关的计数器, 状态机及接收移位寄存器和缓存中的数据 注 1: 一个PCLK时钟后自动清除 注 2: 设置此位前建议先确认RXBUSY (UART_PROTSTS[10]) 是否正在接收数据
[16]	TXRST	发送复位 0 = 无动作. 1 = 复位与发送相关的计数器, 状态机及发送移位寄存器和缓存中的数据 注: 一个PCLK时钟后自动清除
[15]	RXCLR	清除接收缓存 0 = 无动作 1 = 接收缓存清零, 只有当缓存没有参与数据通信时使用 注: 一个PCLK时钟后自动清除
[14]	RXOVLEN	接收缓存溢出中断使能控制 0 = 接收缓存溢出中断禁止 1 = 接收缓存溢出中断使能
[13:8]	Reserved	保留.
[7]	TXCLR	清除发送缓存 0 = 无动作 1 = 发送缓存清零, 只有当缓存没有参与数据通信时使用 注: 一个PCLK时钟后自动清除
[6:0]	Reserved	保留.

USCI 发送/接收缓冲状态寄存器(UUART BUFSTS)

寄存器	偏移	R/W	描述	复位值
UUART_BUFSTS x = 0, 1	UUARTx_BA+0x3C	R	USCI 发送/接收缓冲状态寄存器	0x0000_0101

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						TXFULL	TXEMPTY
7	6	5	4	3	2	1	0
Reserved				RXOVIF	Reserved	RXFULL	RXEMPTY

位	描述	
[31:10]	Reserved	保留.
[10]	Reserved	保留.
[9]	TXFULL	发送缓冲满指示 0 = 发送缓冲未满. 1 = 发送缓冲满.
[8]	TXEMPTY	发送缓冲空指示 0 = 发送缓冲非空 1 = 发送缓冲空
[7:4]	Reserved	保留.
[3]	RXOVIF	接收缓存溢出标志 此位表示检测到接收缓存溢出事件, 如果RXOVIEN (UUART_BUFCTL[14])使能, 相应中断会产生, 此位软件写1清零 0 = 没有检测到接收缓存溢出事件 1 = 检测到接收缓存溢出事件
[2]	Reserved	保留.
[1]	RXFULL	接收缓冲满指示 0 = 接收缓冲未满 1 = 接收缓冲满

USCI 唤醒控制寄存器(UUART_WKCTL)

寄存器	偏移	R/W	描述	复位值
UUART_WKCTL x = 0, 1	UUARTx_BA+0x54	R/W	USCI 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					PDBOPT	Reserved	WKEN

位	描述	
[31:3]	Reserved	保留
[2]	PDBOPT	掉电模式选项 0 = 如果数据正在收发，MCU运行WFI指令进入掉电模式，那么MCU将立即停止传输并进入掉电模式 1 = 如果数据正在收发，MCU运行WFI指令进入掉电模式，此时MCU不会停止传输并且会进入空闲模式
[1]	Reserved	保留.
[0]	WKEN	唤醒使能位 0 =唤醒功能禁用 1 = 唤醒功能使能

USCI 唤醒状态寄存器(UUART_WKSTS)

寄存器	偏移	R/W	描述	复位值
UUART_WKSTS x = 0, 1	UUARTx_BA+0x58	R/W	USCI 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							WKF

位	描述	
[31:1]	Reserved	保留.
[0]	WKF	唤醒标志 当芯片从掉电状态唤醒此位置1, 软件写1清零

USCI 协议控制寄存器– UART (UART_PROTCTL)

寄存器	偏移	R/W	描述	复位值
UART_PROTCTL x = 0, 1	UARTx_BA+0x5C	R/W	USCI 协议控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PROTEN	Reserved	BCEN	Reserved	Reserved	STICKEN	Reserved	BRDETITV
23	22	21	20	19	18	17	16
BRDETITV							
15	14	13	12	11	10	9	8
Reserved	WAKECNT				Reserved	DATWKEN	LINRXEN
7	6	5	4	3	2	1	0
LINBRKEN	ABREN	Reserved			EVENPARITY	PARITYEN	STOPB

位	描述	
[31]	PROTEN	UART 协议使能位 0 = UART 协议禁用 1 = UART 协议使能
[30]	Reserved	保留.
[29]	BCEN	发送break控制使能位 0 = 发送break控制禁止 1 = 发送break控制使能 注意: 当此位为1, 串行数据输出(Tx)被强制输出低电平。
[27]	Reserved	保留.
[26]	STICKEN	校验使能位 0 = 不使能校验位. 1 = 使能校验位.
[25]	Reserved	保留.
[24:16]	BRDETITV	波特率检测间隔 此位表示从机在计算波特率时一个位有多少个时钟周期, 这里的时钟由TMCNTSRC (UART_BRGEN [5])选择. 接收到的数据应该是1,0交替的信号 (例如模板数据0x55), 用户可以在ABRDITIF (UART_PROTSTS[9])置位时读取BRDETITV的值来获取当前输入信号的波特率 注: 此位域可以写零清除
[15]	Reserved	保留.
[14:11]	WAKECNT	唤醒计数器 此位域决定了唤醒时从机在信号起始位可以获得多少个fPDS_CNT时钟周期
[10]	Reserved	保留.

[9]	DATWKEN	数据唤醒模式使能位 0 = 数据唤醒模式禁止 1 = 数据唤醒模式使能
[8]	LINRXEN	LIN RX 双工模式使能位 0 = LIN RX 双工模式禁用 1 = LIN RX 双工模式使能 .LIN 能做为从机接收LIN 帧。 注: 当LIN操作有效时，该位用于检测接收数据持续时间。
[7]	LINBRKEN	LIN TX 断点模式使能位 0 = LIN TX 断点模式禁用。 1 = LIN TX 断点模式使能。 注1: 当TX 断点域的传输结束时，该位自动清0 注2: 在第一个数据传输之前，将送出13个位的低电平和1个位的高电平
[6]	ABREN	自动波特率检测使能位 0 = 自动波特率检测功能禁止 1 = 自动波特率检测功能使能 注: 当自动波特率检测操作完成，硬件清除此位，如果ARBIEN (UART_PROTIEN [1])=1，ABRDETIF (UART_PROTST[9])中断会产生
[5:4]	Reserved	保留。
[3]	Reserved	保留。
[2]	EVENPARITY	偶校验使能位 0 = 收发数据中有奇数个1 1 = 收发数据中有偶数个1 注: 此位只有当PARITYEN 设置后有效
[1]	PARITYEN	校验使能位 此位定义UART数据中使能校验功能。 0 = 校验功能禁止 1 = 校验功能使能
[0]	STOPB	停止位 此位定义了UART数据帧中有几个停止位 0 = 1个停止位 1 = 2个停止位

USCI 协议中断使能寄存器– UART (UUART_PROTIEN)

寄存器	偏移	R/W	描述	复位值
UUART_PROTIEN x = 0, 1	UUARTx_BA+0x60	R/W	USCI 协议中断使能寄存器	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							
7	6	5	4	3	2	1	0
Reserved					RLSIEN	ABRIEN	BRKIEN

位	描述	
[31:3]	Reserved	保留
[2]	RLSIEN	接收总线状态中断使能 0 = 接收总线状态中断禁止 1 = 接收总线状态中断使能 注: UUART_PROTSTS[7:5]表示当前接收总线的中断状态
[1]	ABRIEN	自动波特率中断使能 0 = 自动波特率中断禁止 1 = 自动波特率中断使能.
[0]	BRKIEN	LIN Break 检测中断使能位 0 = LIN 断点检测中断禁用. 1 = LIN 断点检测中断使能.

USCI 协议状态寄存器- UART (UUART_PROTSTS)

寄存器	偏移	R/W	描述	复位值
UUART_PROTSTS x = 0, 1	UUARTx_BA+0x64	R/W	USCI 协议状态寄存器	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				ABERRSTS	RXBUSY	ABRDETIF	BRKDETIF
7	6	5	4	3	2	1	0
BREAK	FRMERR	PARITYERR	RXENDIF	RXSTIF	TXENDIF	TXSTIF	Reserved

位	描述	
[31:17]	Reserved	保留.
[16:12]	Reserved	保留.
[11]	ABERRSTS	<p>自动波特率错误状态</p> <p>当自动波特率检测计数器溢出时，此位置位，用户需要修订 CLKDIV (UUART_BRGEN[25:16]) 的值并使能 ABREN (UUART_PROTCTL[6]) 重新检测波特率</p> <p>0 = 自动波特率检测计数器没有溢出</p> <p>1 = 自动波特率检测计数器有溢出</p> <p>注 1: 此位与 ABRDETIF 同时置位.</p> <p>注 2: 此位可以通过写 1 到 ABRDETIF 或 ABERRSTS 清零.</p>
[10]	RXBUSY	<p>RX 总线状态标志 (只读)</p> <p>此位显示接收器的总线状态</p> <p>0 = 接收器空闲</p> <p>1 = 接收器繁忙</p>
[9]	ABRDETIF	<p>自动波特率中断标志</p> <p>当波特率自动检测完成时此位置位，如果 ABRIEN (UUART_PROTCTL[6])=1，自动波特率中断将会产生，此位在模板数据 0x55 接收过程中会被置位 4 次</p> <p>0 = 自动波特率检测功能未完成</p> <p>1 = 自动波特率检测完成</p> <p>注: 此位写 1 清零</p>
[8]	BRKDETIF	<p>LIN 断点检测中断标志 (只读)</p> <p>在 LIN 模式时，每当数据接收端 (RX) 持续超过 12 位传输时间的“空白状态”（逻辑 0），该位就置 1.</p> <p>0 = LIN 断点没有检测到</p> <p>1 = LIN 断点检测到</p>

		注: 该位只读, 但可以写1清0
[7]	BREAK	<p>断点标志</p> <p>当接收数据输入 (Rx) 收到低电平, 并保持一个全字发送时间 (包括: 起始位+数据位+校验位+停止位), 此位置1</p> <p>0 = 断点没有产生</p> <p>1 = 断点产生.</p> <p>注:此位可以通过写1到BREAK, FRMERR 及 PARITYERR清零</p>
[6]	FRMERR	<p>帧错误标志</p> <p>当接收到的数据没有一个有效的停止位 (停止位检测到0), 此位置1</p> <p>0 = 没有帧错误产生</p> <p>1 = 有帧错误产生</p> <p>注:此位可以通过写1到BREAK, FRMERR 及 PARITYERR清零</p>
[5]	PARITYERR	<p>校验错误标志</p> <p>当接收数据没有一个合格的校验位时, 该位置1</p> <p>0 = 没有校验位错误</p> <p>1 = 有校验错误</p> <p>注:此位可以通过写1到BREAK, FRMERR 及 PARITYERR清零</p>
[4]	RXENDIF	<p>接收结束中断标志</p> <p>0 = 没有产生接收结束中断</p> <p>1 = 有接收结束中断产生</p> <p>注: 软件写1清零</p>
[3]	RXSTIF	<p>接收开始中断标志</p> <p>0 = 没有接收开始中断</p> <p>1 = 有接收开始中断</p> <p>注: 软件写1清零</p>
[2]	TXENDIF	<p>发送结束中断标志</p> <p>0 = 没有发送结束中断</p> <p>1 = 有发送结束中断发生</p> <p>注: 软件写1清零</p>
[1]	TXSTIF	<p>发送开始中断标志</p> <p>0 = 没有发送开始中断</p> <p>1 = 有发送开始中断产生</p> <p>注 1: 软件写1清零</p> <p>注 2:用户可以通过此标志知道发送缓存空了, 可以写入新的数据</p>
[0]	Reserved	保留.

6.13 USCI – SPI 模式

6.13.1 概述

USCI控制器的SPI模式就是同步串行数据通信，支持全双工数据传输，支持主从机模式，支持4线双向接口。USCI的SPI功能是从外设接收数据时实现串并转换，发送数据到外设时实现并串数据转换，通过FUNMODE (USPI_CTL[2:0]) = 0x1选择SPI模式

设置SLAVE (USPI_PROTCTL[0])决定SPI协议处在主机/从机模式，图 6.13-1 和 图 6.13-2展示了在主机和从机模式的应用框图。

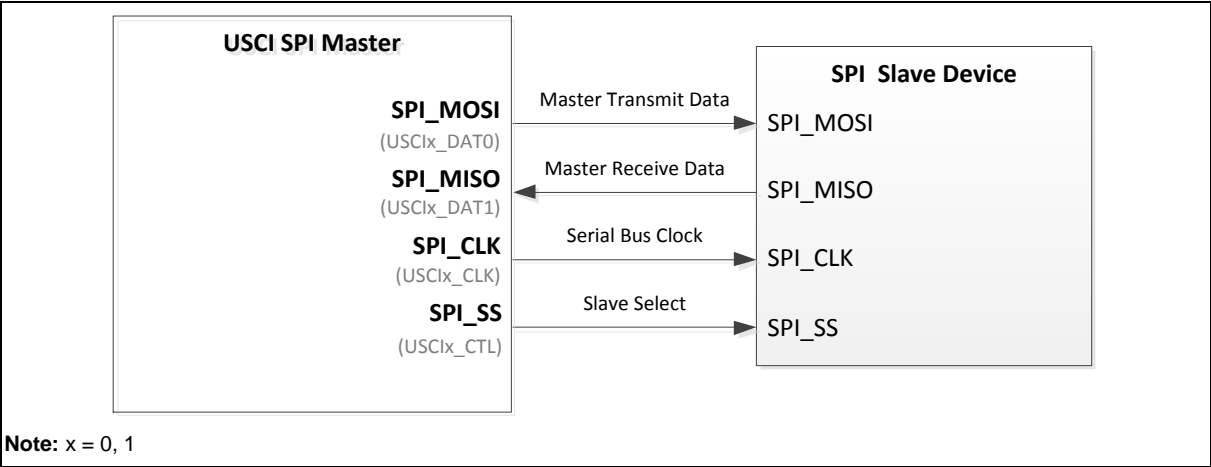


图 6.13-1 SPI 主机模式应用框图 (x=0, 1)

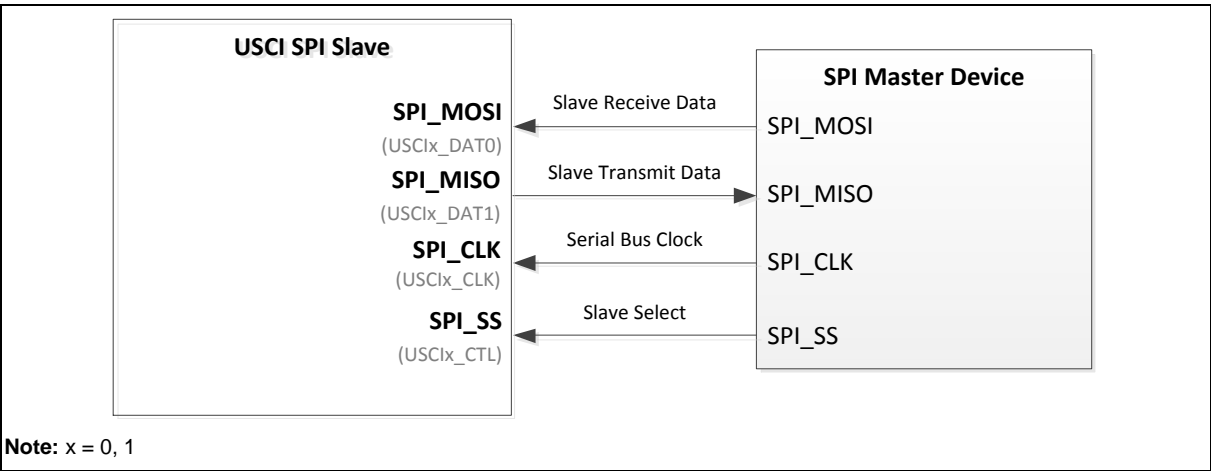


图 6.13-2 SPI 从机模式应用框图 (x=0, 1)

6.13.2 特性

- 支持主机或从机模式操作 (最高频率—主机 = $f_{PCLK}/2$, 从机 < $f_{PCLK}/5$)
- 可配置传输数据的字长4 到 16位

- 支持一个发送缓存和两个接收缓存
- 可设定MSB或LSB优先
- 支持传输暂停功能
- 支持3线双向数据接口，无从机选择信号
- 从机模式支持从机选择信号唤醒

6.13.3 框图

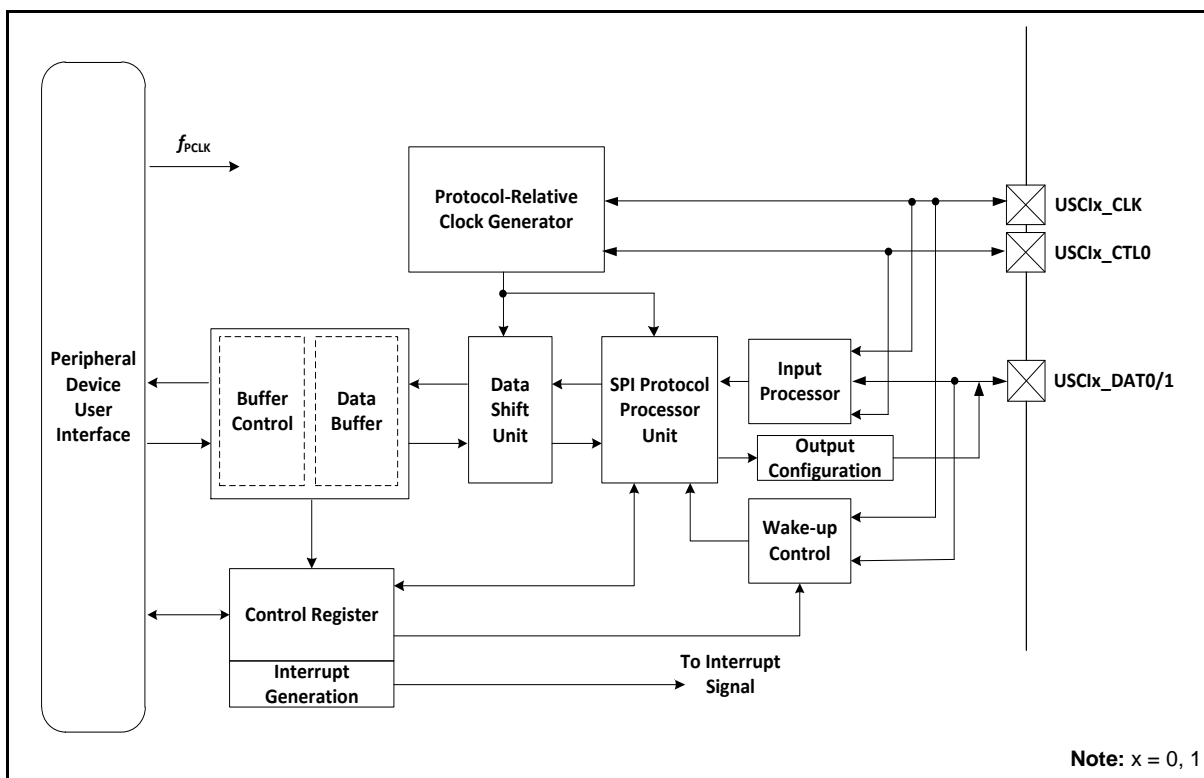


图 6.13-3 USCI - SPI 模式框图

6.13.4 基本配置

USC10 的SPI模式基本配置如下:

- USCI0管脚通过SYS_GPA_MFP, SYS_GPC_MFP及SYS_GPD_MFP寄存器配置。
- USCI0CKEN (CLK_APBCLK[24])使能USCI0外设时钟。
- 通过USCI0RST (SYS_IPRST1[24])复位USCI0控制器。

USCI1 的SPI模式基本配置如下:

- USCI1 管脚通过SYS_GPA_MFP, SYS_GPC_MFP, 及SYS_GPD_MFP 寄存器配置.
- USCI1CKEN (CLK_APBCLK[25]) 使能USCI1外设时钟.
- 通过 USCI1RST (SYS_IPRST1[25]) 复位 USCI1控制器.

6.13.5 功能描述

6.13.5.1 USCI 通用功能描述

详细信息请参考章节 6.11.4.

6.13.5.2 信号描述

主机控制数据传输的开始和结束，主机产生SPI总线时钟和从机选择信号。从机选择信号指示了数据传输的开始和结束，主机可以通过它启动从机的收发操作。从机设备接收主机发送的时钟信号及一个可选的从机选择信号。SPI通讯信号如表 6.13-1所示

SPI 模式	接收数据	发送数据	串行总线时钟	从机选择
全双工 SPI 主机	SPI_MISO (USCIx_DAT1)	SPI_MOSI (USCIx_DAT0)	SPI_CLK (USCIx_CLK)	SPI_SS (USCIx_CTL0)
全双工 SPI 从机	SPI_MOSI (USCIx_DAT0)	SPI_MISO (USCIx_DAT1)	SPI_CLK (USCIx_CLK)	SPI_SS (USCIx_CTL0)

表 6.13-1 SPI 通讯信号 (x=0, 1)

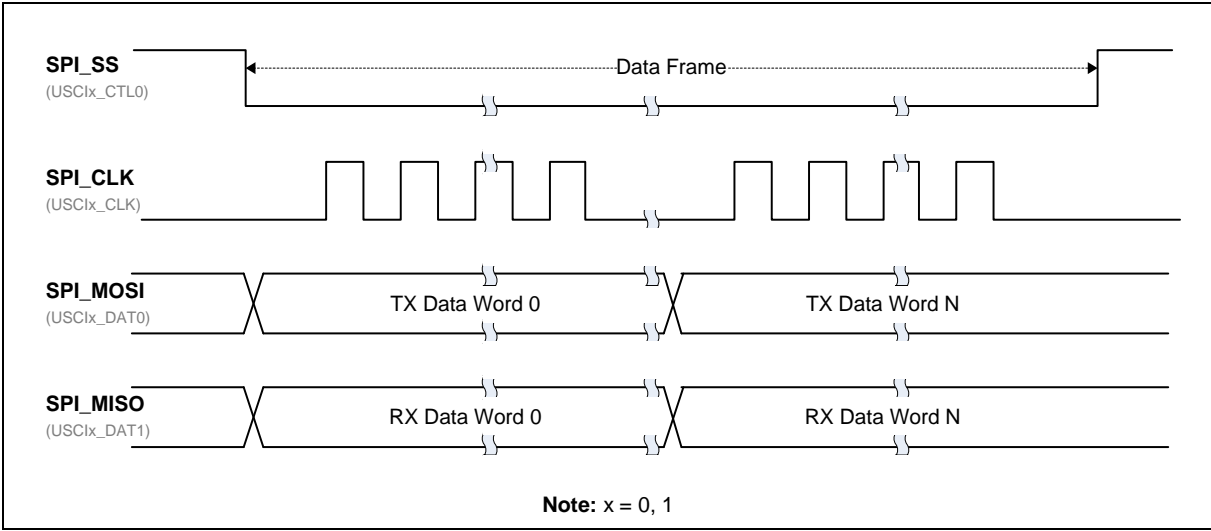


图 6.13-4 4 线全双工 SPI 通讯信号 (主机模式)

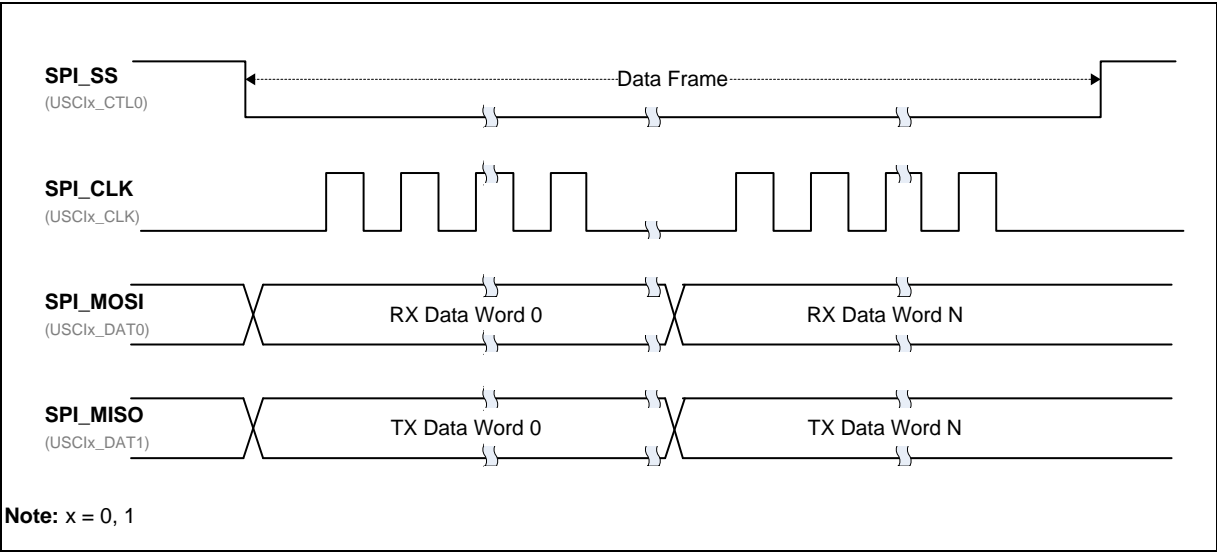


图 6.13-5 4线全双工SPI 通讯信号 (从机模式)

6.13.5.3 串行总线时钟配置

USCI控制器需要外设时钟驱动USCI逻辑单元实现数据收发，外设时钟为PCLK。

在主机模式下，SPI总线时钟取决于协议相关时钟产生器，通常SPI总线时钟表示为SPI时钟，SPI时钟频率是 f_{SAMP_CLK} 的一半，可以通过SPCLKSEL (USPI_BRGEN[3:2])选择，关于时钟产生器详细信息请参照章节6.11.4.4。

在从机模式下，SPI总线时钟由外部主机提供，外设时钟频率 f_{PCLK} 必须是SPI总线时钟频率的5倍以上（即SPI总线时钟频率 $<1/5$ 外设 f_{PCLK} 时钟频率）。

在SPI模式下，SCLKMODE (USPI_PROTCTL[7:6]) 定义了总线的空闲状态及收发数据时时钟边沿的变化，主从机要有相同的SCLKMODE设定，SPI的四种总线配置如表 6.13-2，和时序如图 6.13-6、图 6.13-7、图 6.13-8 和 图 6.13-9。

SCLKMODE [1:0]	SPI 时钟空闲状态	发送时序	接收时序
0x0	低	下降沿	上升沿
0x1	低	上升沿	下降沿
0x2	高	上升沿	下降沿
0x3	高	下降沿	上升沿

表 6.13-2 串行总线时钟配置

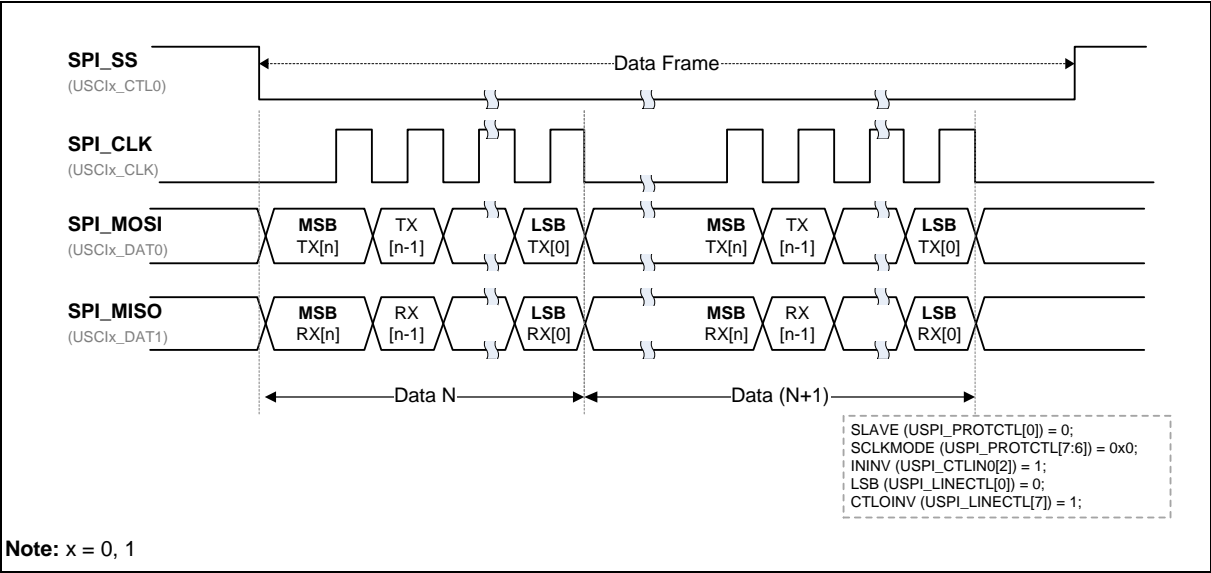


图 6.13-6 SPI 通讯时钟配置 (SCLKMODE=0x0)

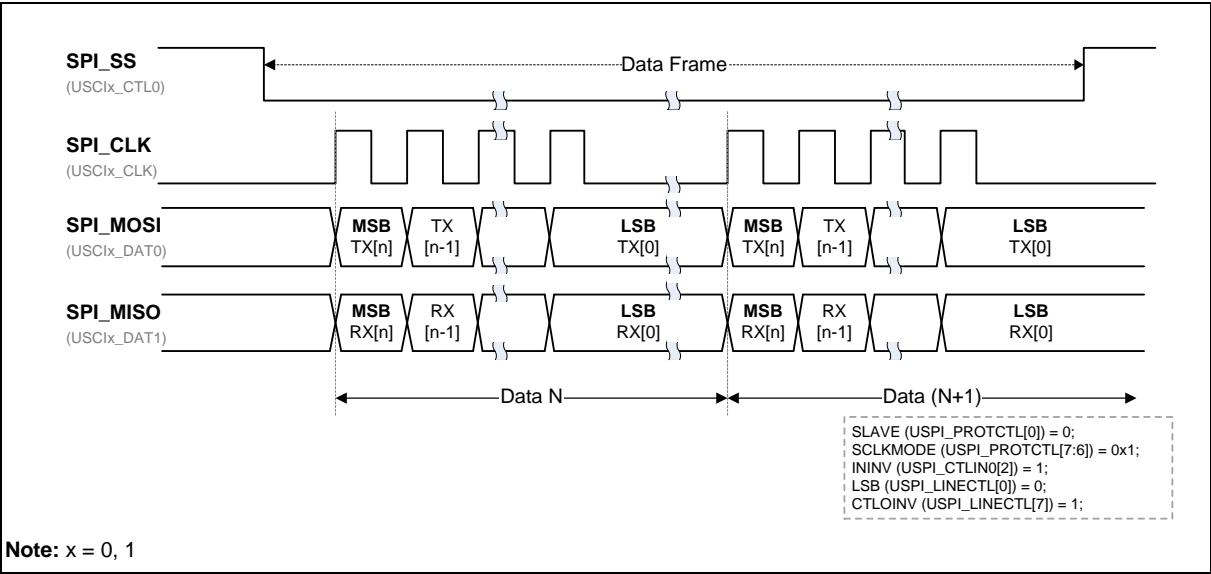


图 6.13-7 SPI 通讯时钟配置(SCLKMODE=0x1)

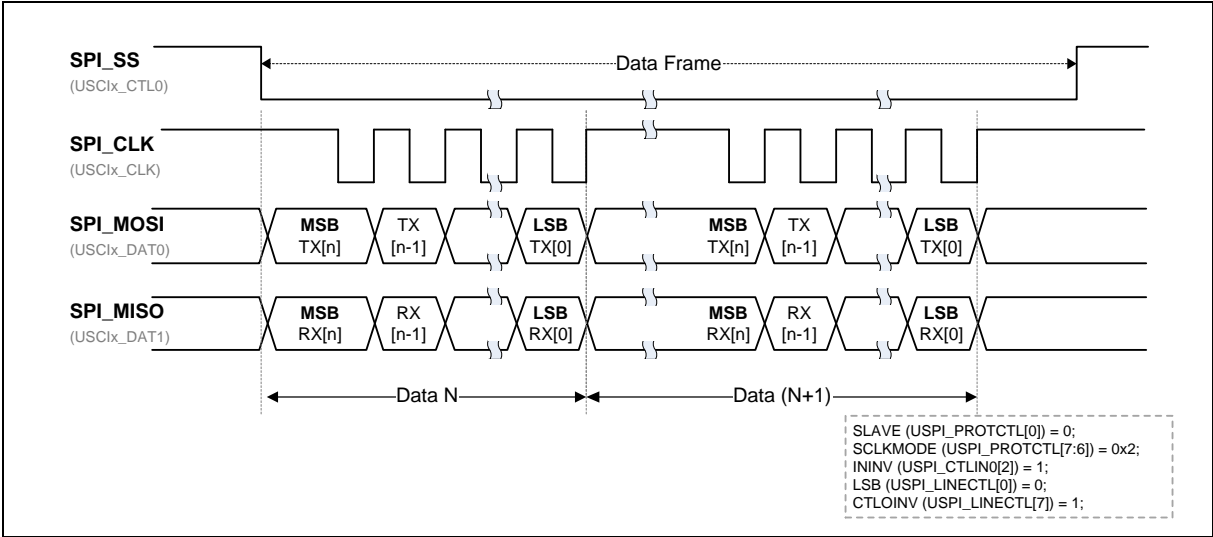


图 6.13-8 SPI 通讯时钟配置 (SCLKMODE=0x2)

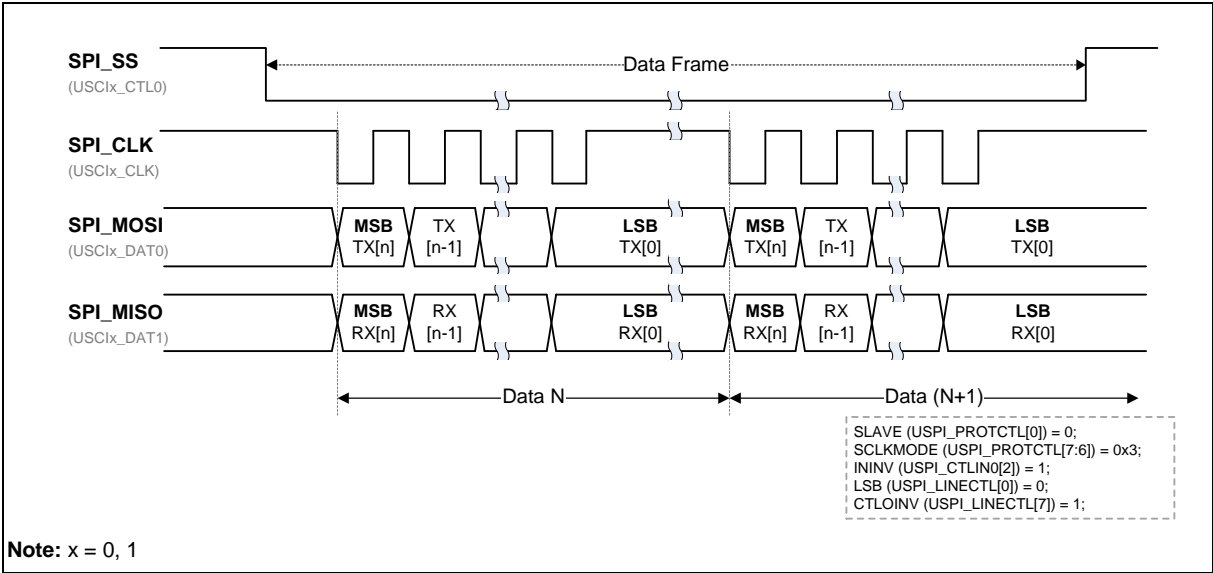


图 6.13-9 SPI 通讯时钟配置 (SCLKMODE=0x3)

6.13.5.4 从机选择型号

SPI协议中从机选择信号默认是高有效，在SPI主机模式，USCI控制器可以通过从机选择信号SPI_SS (USCIx_CTL0)驱动SPI从机，在SPI从机模式，输入的从机选择信号可以通过ININV (USPI_CTLIN0[2])设置输出反向信号。

如果外部主机SPI设备的从机选择信号是低有效，从机设备输入反向控制信号，需设置ININV (USPI_CTLIN0[2])位为1。如果USCI工作在SPI主机模式，为了提供给外部SPI从机设备低电平有效的从机选择信号，需要通过设置CTLOINV (USPI_LINECTL[7])为1输出从机选择反向信号。

从从机选择信号的有效边沿到第一个SPI时钟输入边沿至少要有2个USCI外设时钟周期。

两个连续发送的数据帧之间，从机选择信号至少保持2个USCI的无效从机选择电平，以保证检测到正确的数据帧结束。

6.13.5.5 发送和接收数据

USCI控制器的SPI协议的收发数据字长可以通过DWIDTH (USPI_LINECTL[11:8])设定，在SPI通讯可以配置4 位到16位字长发送和接收数据。

LSB位USPI_LINECTL[0]定义发送数据位LSB优先还是MSB优先，如果LSB位设置为1，传输数据LSB位优先传输，如果LSB位设置为0，传输数据MSB优先传输

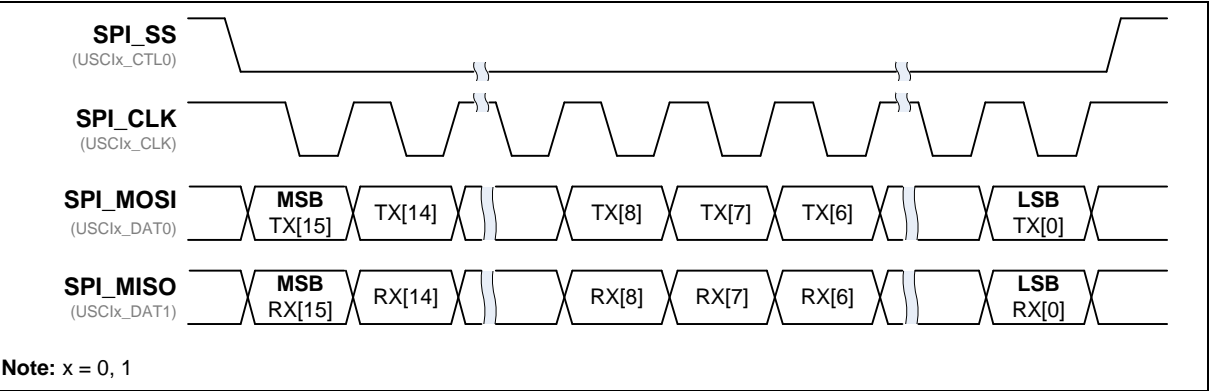


图 6.13-10 MSB 优先的 16 位字长传输

6.13.5.6 字传输暂停

主机在发送两个连续的数据字时，可以通过SUSPITV (USPI_PROTCTL[11:8]) 配置两次传输间0.5-15.5个SPI时钟周期的暂停间隔，暂停间隔的定义指的是前一个数据最后一个时钟信号边沿到下一个数据的第一个时钟信号边沿，其默认值是SUSPITV (USPI_PROTCTL[11:8]) = 0x3（3.5个SPI时钟周期）。

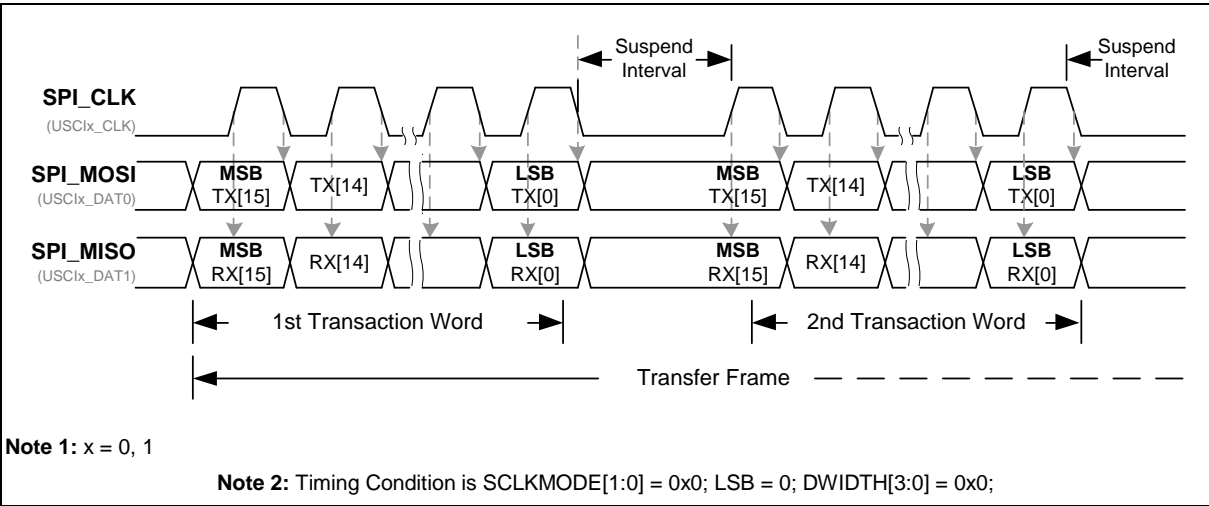


图 6.13-11 两字传输间的暂停时间间隔

6.13.5.7 自动从机选择功能

AUTOSS (USPI_PROTCTL[3]) 用于 SPI 主机模式下的自动从机地址选择功能，如果 AUTOSS(USPI_PROTCTL[3]) =1，那么从机选择信号将会自动产生(通过 USCIX_CTL0 线)，软件设置 SS (USPI_PROTCTL[2])是不影响从机选择信号输出；也就是说从机选择信号在USCI控制器发送数据时自动生效，数据发送完成自动失效，如果SUSPITV (USPI_PROTCTL[11:8])大于等于 3，两个连续发送的数据中间也会自动失效。

如果AUTOSS (USPI_PROTCTL[3])=0，USCIX_CTL0管脚的从机选择信号需要通过设定 SS (USPI_PROTCTL[2])完成。从机选择信号内部高有效，通过设置CTLOINV (USPI_LINECTL[7])可以将信号反转。

在 SPI 主机模式 如果 SUSPITV (USPI_PROTCTL[11:8]) 的值 小于 3，并且 AUTOSS (USPI_PROTCTL[3])=1，那么从机选择信号在两次连续发送时将一直保持有效状态。

在SPI从机模式，为了确保从机能够读到从机选择信号的无效状态，此无效信号周期必须大于2个外设时钟周期。

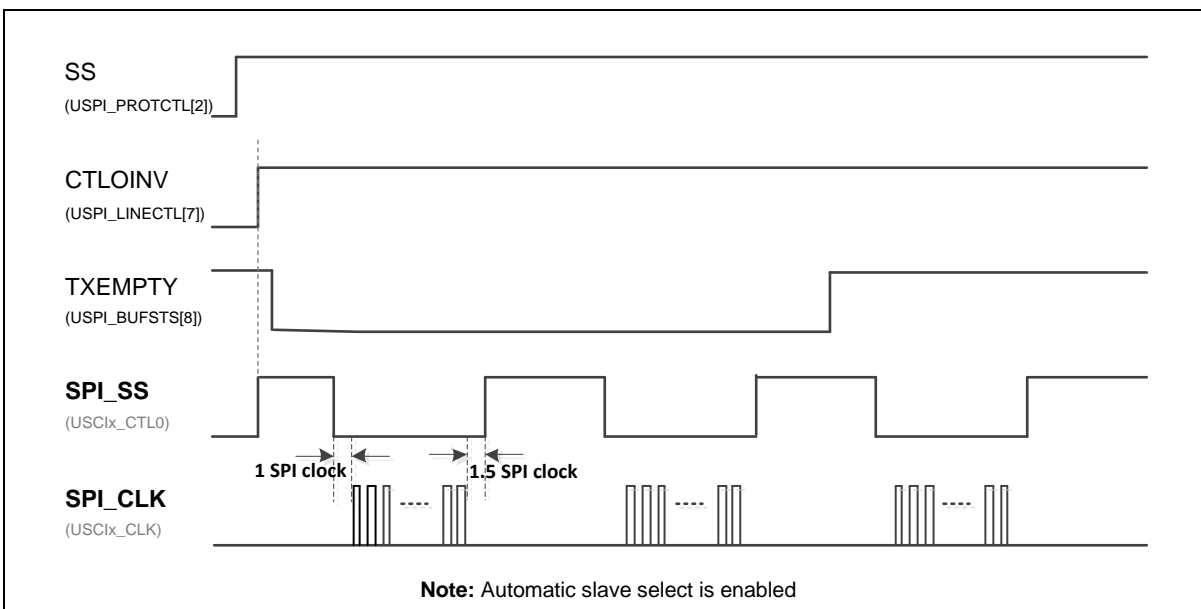


图 6.13-12 自动从机选择 (SUSPITV ≥ 0x3)

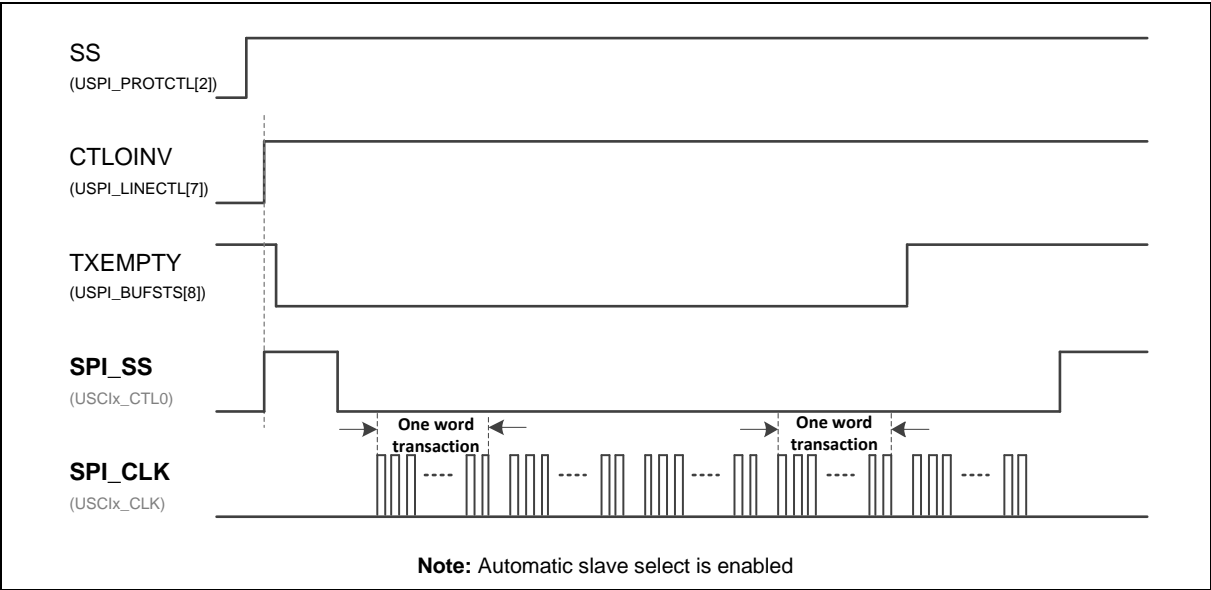


图 6.13-13 自动从机选择 (SUSPITV < 0x3)

6.13.5.8 从机3-线模式

当软件设置SLV3WIRE (USPI_PROTCTL[1])为1，使能从机3线模式，USCI SPI通信可以工作在无从机选择信号的从机模式，SLV3WIRE (USPI_PROTCTL[1])仅在从机模式下有效，只需要将SPI_CLK (USCIx_CLK线), SPI_MOSI (USCIx_DAT0 线), 及 SPI_MISO (USCIx_DAT1线)三条线连接到主机就可以通信；当SLV3WIRE (USPI_PROTCTL[1]) 设置为1，SPI从机在FUNMODE(USPI_CTL [2:0])=0x1后就已经准备好数据的收发。

6.13.5.9 数据传输模式

USCI控制器支持全双工SPI传输模式。

在全双工模式下，有两条数据线一条用于数据发送，另一条用于数据接收，这样数据收发就可以同时进行。

SCLKMODE (USPI_PROTCTL[7:6])定义了USCIx_DAT0管脚上数据输出信号的发送时序；根据寄存器的设定，传输数据会在SPI时钟上升沿或从机选择信号的有效边沿；数据脚（USCIx_DAT0）输出的电平会保持在最后一个数据的最后一位状态，直到下一个数据时钟沿变化时才会更改。

6.13.5.10 中断

数据传输中断

发送开始中断

发送缓存中数据字的第一个数据位被传出后中断事件TXSTIF (USPI_PROTSTS[1])置位，软件写1清零。如果TXSTIEN (USPI_INTEN[1])位使能，控制器将产生中断。

发送结束中断

发送缓存中最后一个数据位被传出后中断事件TXENDIF (USPI_PROTSTS[2])置位，软件写1清零。如果TENDIEN (USPI_INTEN[2])位使能，控制器将产生中断。

接收开始中断

接收到数据字的第一个数据位后中断事件RXSTIF (USPI_PROTSTS[3])置位，软件写1清零。如果RXSTIEN (USPI_INTEN[3])位使能，控制器将产生中断。

接收结束中断

接收到一个数据字的最后一位后中断事件RXENDIF (USPI_PROTSTS[4])置1，软件写1清零。如果RXENDIEN (USPI_INTEN[4])位使能，控制器将产生中断。

协议相关中断

SPI从机选择中断

SPI从机模式有从机选择有效及无效中断标志，SSACTIF (USPI_PROTSTS[9]) 及 SSINAIF (USPI_PROTSTS[8])，当SLAVE (USPI_PROTCTL [0])=1和从机选择信号有效或无效，如果SSINAIF (USPI_PROTIEN[0]) 或 SSACTIEN (USPI_PROTIEN[1]) 置位，相应中断会产生；SPI内部从机选择信号是高有效，用户可以通过设置ININV (USPI_CTLIN0[2])反转从机选择信号的极性满足低电平有效的设备

从机超时中断

在SPI从机模式下，从机超时功能可以使用户知道数据传输没有完成突然没有了时钟信号的错误；此功能通过设置SLVTOCNT (USPI_PROTCTL[25:16])，使用时序测量计数器来测量超时周期，时序测量计数器的频率可以通过TMCNTSRC (USPI_BRGEN[5])选择。

SPI从机模式下，TMCNTEN (USPI_BRGEN[4])=1 且SLVTOCNT (USPI_PROTCTL[25:16]) 不为0，时序测量计数器使能；此计数器在接收数据的第一个时钟开始计数，收到下一个时钟计数器清零并重新计数，最终计数器将在数据接收完成后清零并停止，如果一个字数据传输完成前计数值等于或大于SLVTOCNT (USPI_PROTCTL[25:16]) 设定的值，就会产生超时中断事件SLVTOIF (USPI_PROTSTS[5])=1

缓存相关中断

如果USCI控制器有收发缓存，则会有缓存相关中断。

接收缓存覆盖中断

如果接收缓存发生数据覆盖事件，RXOVIF (USPI_BUFSTS[3])将被置1，此位通过软件写1清零。如果RXOVIEN (USPI_BUFCTL[14]) 设置为1，控制器将产生中断。

发送缓存欠载中断

如果发生发送缓存数据欠载事件，TXUDRIF (USPI_BUFSTS[11]) 将被置1。此位通过软件写1清零。如果TXUDRIEN (USPI_BUFCTL[6]) 设置为1，控制器将产生中断。

6.13.5.11 时序图

USCI SPI协议的从机选择信号默认为高有效，可以通过CTLOINV (USPI_LINECTL[7])设置反转

通过SCLKMODE (USPI_PROTCTL[7:6])可以设置SPI时钟空闲时的状态及收发数据的时钟沿，发送数据的长度由DWIDTH (USPI_LINECTL[11:8])定义，LSB (USPI_LINECTL[0])决定数据位发送顺序，四种SPI主从操作的时序图相关设置如图 6.13-14, 图 6.13-15, 图 6.13-16 和 图 6.13-17所示。

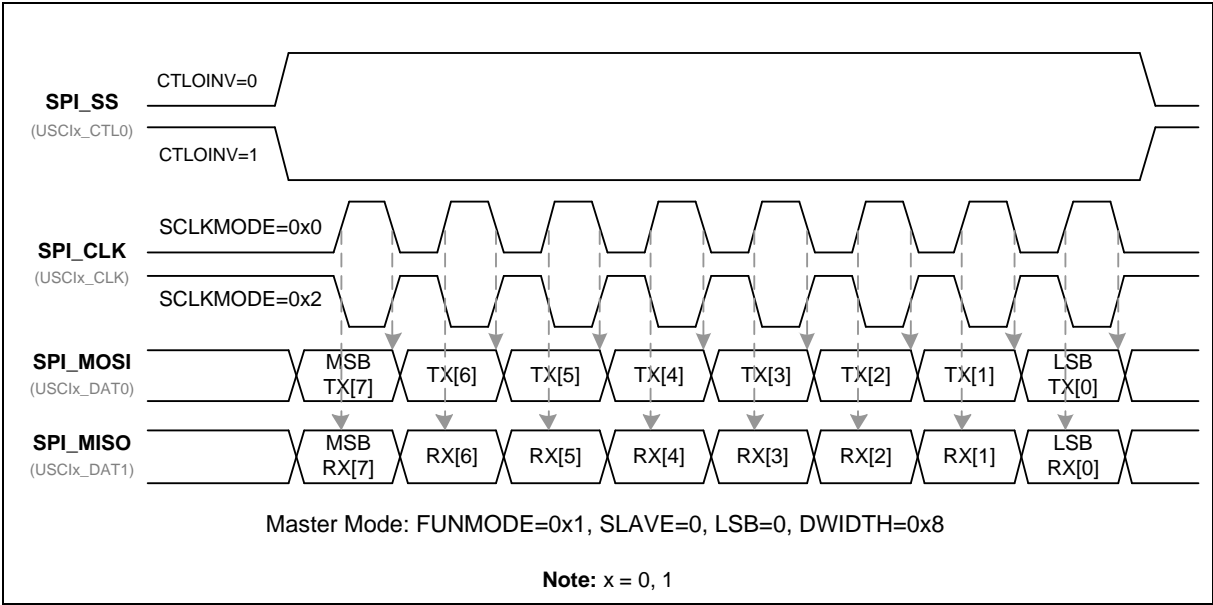


图 6.13-14 SPI 主机模式时序

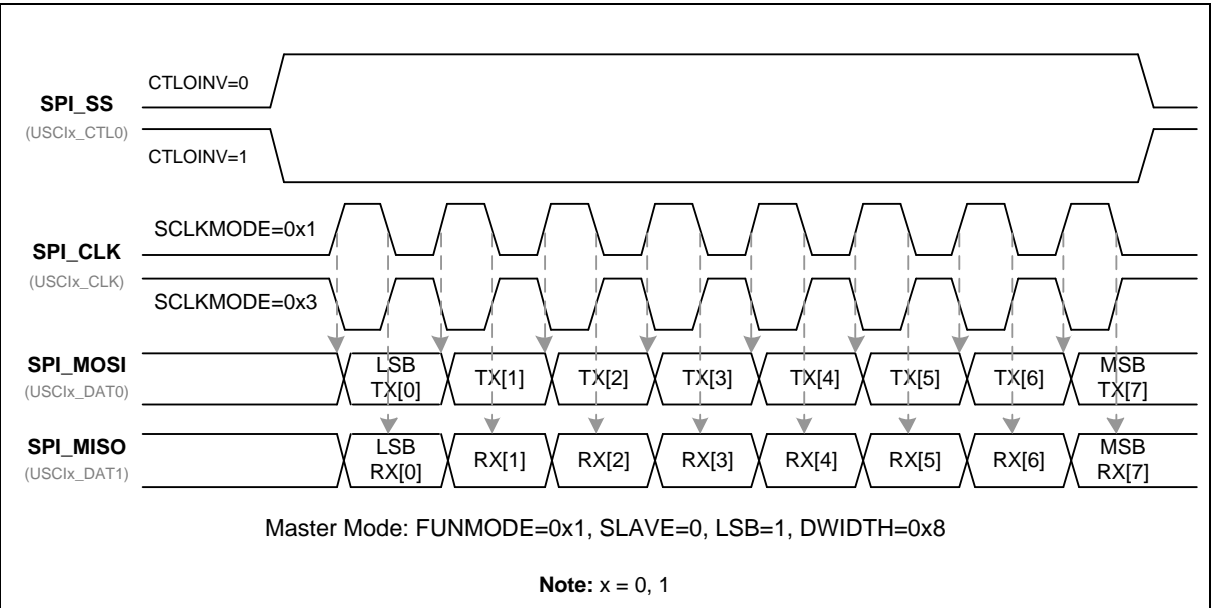


图 6.13-15 SPI 主机模式时序 (时钟信号相位可选)

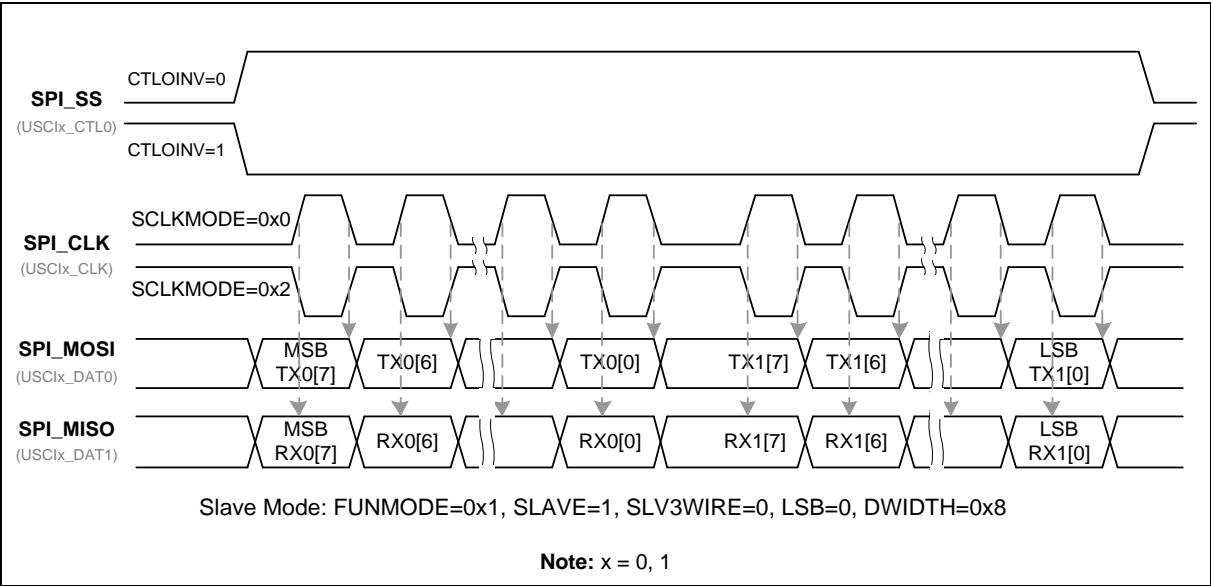


图 6.13-16 SPI 从机模式时序

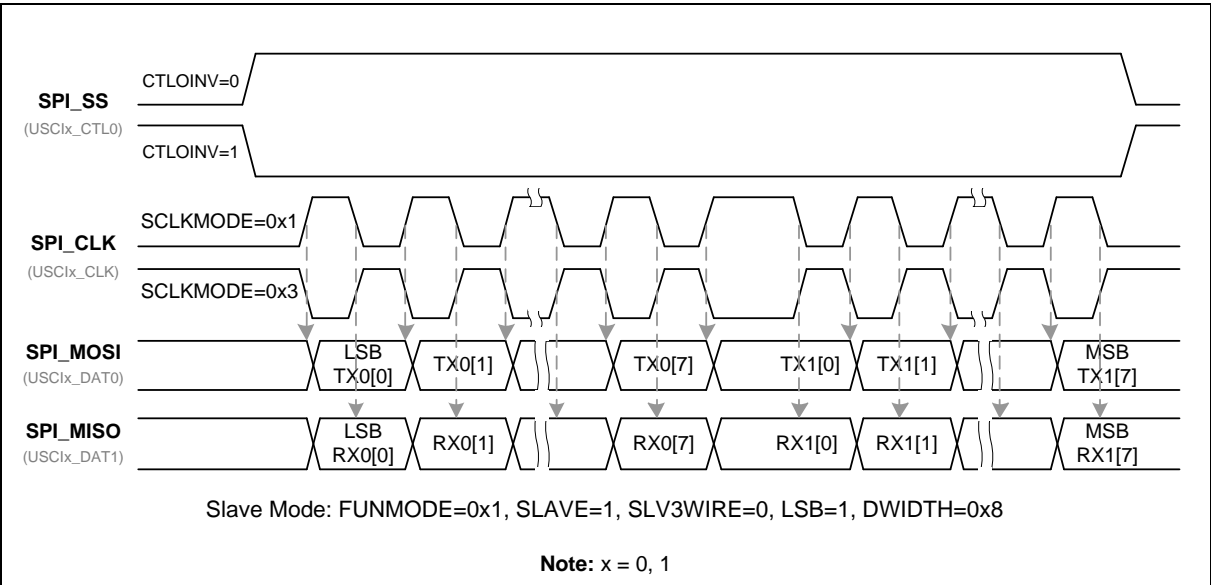


图 6.13-17 SPI 从机模式时序 (时钟信号相位可选)

6.13.5.12 编程流程

这章节描述USCI SPI 数据传输编程流程。

主机模式:

1. 设置寄存器 CLK_APBCLK 使能USCI外设时钟。
2. 通过功能复用寄存器配置USCI功能管脚。

3. 设置FUNMODE (USPI_CTL[2:0]) = 0x1 选择SPI模式
4. 设置寄存器 USPI_BRGEN 配置SPI 总线时钟频率.
5. 根据用户具体应用配置下列寄存器.
 - CTLOINV (USPI_LINECTL[7]): 如果从机选择信号为低有效,则此位设为1反之为0.
 - DWIDTH (USPI_LINECTL[11:8]): 数据宽度设定.
 - LSB (USPI_LINECTL[0]): LSB 优先或MSB优先.
 - TSMSEL (USPI_PROTCTL[14:12]): 全双工SPI 传输.
 - SCLKMODE (USPI_PROTCTL[7:6]): 设置时钟时序.
 - AUTOSS (USPI_PROTCTL[3]): 使能自动从机选择功能
 - SLAVE (USPI_PROTCTL[0]): 主机模式设置为0.
6. 设置 PROTEN (USPI_PROTCTL[31]) 为 1使能SPI协议.
7. 如果自动从机选择功能禁止(AUTOSS=0), 数据传输前设置SS (USPI_PROTCTL[2]) 为1; 通过用户应用设置SS 为0 使从机选择信号无效.
8. 写数据到 USPI_TXDAT 寄存器发送数据.
9. 用户可以通过USPI_RXDAT读取数据, 一直读到RXEMPTY (USPI_BUFSTS[0]) = 0。 当TXFULL (USPI_BUFSTS[9]) = 0时, 可以写数据到USPI_TXDAT触发SPI数据发送.

从机模式:

1. 设置寄存器 CLK_APBCLK 使能USCI外设时钟.
2. 通过功能复用寄存器配置USCI功能管脚.
3. 设置FUNMODE (USPI_CTL[2:0]) = 0x1 选择SPI模式
4. 根据用户具体应用配置下列寄存器:
 - ININV (USPI_CTLIN0[2]): 如果从机选择信号为低有效则此位设为1反之为0.
 - DWIDTH (USPI_LINECTL[11:8]): 数据宽度设定.
 - LSB (USPI_LINECTL[0]): LSB 优先或MSB优先.
 - TSMSEL (USPI_PROTCTL[14:12]): 全双工SPI 传输
 - SCLKMODE (USPI_PROTCTL[7:6]): 设置时钟时序.
 - SLAVE (USPI_PROTCTL[0]): 从机模式设置为1.
5. 设置 PROTEN (USPI_PROTCTL[31]) 为 1使能SPI协议.
6. 写数据到 USPI_TXDAT 寄存器发送数据.
7. 用户可以通过USPI_RXDAT读取数据, 一直读到RXEMPTY (USPI_BUFSTS[0]) = 0。 当TXFULL (USPI_BUFSTS[9]) = 0时, 可以写数据到USPI_TXDAT触发SPI数据发送.

6.13.5.13 唤醒功能

USCI控制器在SPI模式支持唤醒系统功能，唤醒源是从机选择信号的变化。

6.13.6 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
USPI基地址: USPIx_BA = 0x4007_0000 + (0x10_0000 * x) x = 0, 1				
USPI_CTL x = 0, 1	USPIx_BA+0x00	R/W	USCI 控制寄存器	0x0000_0000
USPI_INTEN x = 0, 1	USPIx_BA+0x04	R/W	USCI 中断使能寄存器	0x0000_0000
USPI_BRGEN x = 0, 1	USPIx_BA+0x08	R/W	USCI 波特率产生寄存器	0x0000_3C00
USPI_DATIN0 x = 0, 1	USPIx_BA+0x10	R/W	USCI 输入数据配置寄存器 0	0x0000_0000
USPI_CTLIN0 x = 0, 1	USPIx_BA+0x20	R/W	USCI 输入控制配置寄存器 0	0x0000_0000
USPI_CLKIN x = 0, 1	USPIx_BA+0x28	R/W	USCI 输入时钟信号配置寄存器	0x0000_0000
USPI_LINECTL x = 0, 1	USPIx_BA+0x2C	R/W	USCI 线控寄存器	0x0000_0000
USPI_TXDAT x = 0, 1	USPIx_BA+0x30	W	USCI 发送数据寄存器	0x0000_0000
USPI_RXDAT x = 0, 1	USPIx_BA+0x34	R	USCI 接收数据寄存器	0x0000_0000
USPI_BUFCTL x = 0, 1	USPIx_BA+0x38	R/W	USCI 发送/接收缓冲控制寄存器	0x0000_0000
USPI_BUFSTS x = 0, 1	USPIx_BA+0x3C	R	USCI 发送/接收缓冲状态寄存器	0x0000_0101
USPI_WKCTL x = 0, 1	USPIx_BA+0x54	R/W	USCI 唤醒控制寄存器	0x0000_0000
USPI_WKSTS x = 0, 1	USPIx_BA+0x58	R/W	USCI 唤醒状态寄存器	0x0000_0000
USPI_PROTCTL x = 0, 1	USPIx_BA+0x5C	R/W	USCI 协议控制寄存器	0x0000_0300
USPI_PROTIEN x = 0, 1	USPIx_BA+0x60	R/W	USCI 协议中断使能寄存器	0x0000_0000
USPI_PROTSTS x = 0, 1	USPIx_BA+0x64	R/W	USCI 协议状态寄存器	0x0000_0000

6.13.7 寄存器描述

USCI 控制寄存器 (USPI_CTL)

寄存器	偏移地址	读/写	描述	复位值
USPI_CTL x = 0, 1	USPIx_BA+0x00	R/W	USCI 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					FUNMODE		

位	描述	
[31:3]	Reserved	保留.
[2:0]	FUNMODE	<p>功能模式</p> <p>此位域选择USCI控制器的协议，非法的设置会关闭USCI功能。两种协议切换时，需要先关闭USCI功能再切换新的协议，写0x0到FUNMODE复位USCI</p> <p>0x0 = 关闭USCI功能，所有协议状态机进入空闲状态</p> <p>0x1 = 选择SPI协议</p> <p>0x2 = 选择UART协议</p> <p>0x4 = 选择 I²C 协议</p> <p>注意: 其他组合保留</p>

USCI 中断使能寄存器 (USPI_INTEN)

寄存器	偏移地址	读/写	描述	复位值
USPI_INTEN x = 0, 1	USPIx_BA+0x04	R/W	USCI 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							
7	6	5	4	3	2	1	0
Reserved			RXENDIEN	RXSTIEN	TXENDIEN	TXSTIEN	Reserved

位	描述	
[31:5]	Reserved	保留.
[4]	RXENDIEN	接收结束中断使能位 该位使能接收数据完成中断 0 = 接收结束中断禁止 1 = 接收结束中断使能
[3]	RXSTIEN	接收开始中断使能位 此位使能接收开始事件中断 0 = 禁止接收开始事件中断. 1 = 使能接收开始事件中断
[2]	TXENDIEN	发送结束中断使能位 此位使能发送结束中断. 0 = 发送结束中断禁止 1 = 发送结束中断使能
[1]	TXSTIEN	发送开始中断使能位 此位使能发送开始中断 0 = 发送开始中断禁止 1 = 发送开始中断使能
[0]	Reserved	保留.

USCI 波特率产生寄存器 (USPI BRGEN)

寄存器	偏移地址	读/写	描述	复位值
USPI_BRGEN x = 0, 1	USPIx_BA+0x08	R/W	USCI 波特率产生寄存器	0x0000_3C00

31	30	29	28	27	26	25	24
Reserved						CLKDIV	
23	22	21	20	19	18	17	16
CLKDIV							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		TMCNTSRC	TMCNTEN	SPCLKSEL		PTCLKSEL	RCLKSEL

位	描述	
[31:26]	Reserved	保留.
[25:16]	CLKDIV	时钟除频器 该位域定义了 $f_{\text{PROT_CLK}}$ 和 $f_{\text{DIV_CLK}}$ 之间的比例关系 ($f_{\text{DIV_CLK}} = f_{\text{PROT_CLK}} / (\text{CLKDIV} + 1)$)
[15:10]	Reserved	保留.
[9:6]	Reserved	保留.
[5]	TMCNTSRC	时序测量计数器时钟选择 0 = 时序测量计数器时钟为: $f_{\text{PROT_CLK}}$. 1 = 时序测量计数器时钟为: $f_{\text{DIV_CLK}}$.
[4]	TMCNTEN	时序测量计数器使能位 该位使能10位时序测量计数器 0 = 时序测量计数器禁止 1 = 时序测量计数器使能
[3:2]	SPCLKSEL	采样时钟源选择 该位域用来选择采样时钟 ($f_{\text{SAMP_CLK}}$) 的时钟源. 00 = $f_{\text{DIV_CLK}}$. 01 = $f_{\text{PROT_CLK}}$. 10 = f_{SCLK} . 11 = $f_{\text{REF_CLK}}$.
[1]	PTCLKSEL	协议时钟源选择 该位选择协议时钟源 ($f_{\text{PROT_CLK}}$). 0 = 参考时钟 $f_{\text{REF_CLK}}$. 1 = $f_{\text{REF_CLK}2}$ ($f_{\text{REF_CLK}}$ 时钟的一半).

[0]	RCLKSEL	<p>参考时钟源选择</p> <p>该位选择参考时钟源 (f_{REF_CLK}).</p> <p>0 = 外设时钟 f_{PCLK}.</p> <p>1 = HXT/LXT.</p>
-----	---------	---

USCI 输入数据信号配置寄存器0(USPI_DATIN0)

寄存器	偏移地址	读/写	描述	复位值
USPI_DATIN0 x = 0, 1	USPIx_BA+0x10	R/W	USCI 输入数据信号配置寄存器 0	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					ININV	Reserved	SYNCSEL

位	描述	
[31:0]	Reserved	保留.
[2]	ININV	输入信号反向选择 该位定义了输入信号是否反转 0 = 信号不会反转 1 = 信号反转 注意: 建议SPI协议设置此位为0
[1]	Reserved	保留.
[0]	SYNCSEL	输入信号同步选择 该位决定非同步信号（可设置反向）或者同步信号（可设置滤波）作为数据移位单元的输入 0 = 非同步信号作为数据移位单元的输入 1 = 同步信号作为数据移位单元的输入 注意: 建议SPI协议设置此位为0

USCI 输入控制信号配置寄存器0 (USPI_CTLIN0)

寄存器	偏移地址	读/写	描述	复位值
USPI_CTLIN0 x = 0, 1	USPIx_BA+0x20	R/W	USCI 输入控制信号配置寄存器 0	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					ININV	Reserved	SYNCSEL

位	描述	
[31:3]	Reserved	保留.
[2]	ININV	输入信号反向选择 此位定义了输入信号是否反转 0 = 信号不会反转 1 = 信号反转
[1]	Reserved	保留.
[0]	SYNCSEL	输入同步信号选择 此位决定非同步信号（可设置反向）或者同步信号（可设置滤波）作为数据移位单元的输入 0 = 非同步信号作为数据移位单元的输入 1 = 同步信号作为数据移位单元的输入 注意: SPI协议, 建议该位设置为0.

USCI 输入时钟信号配置寄存器 (USPI_CLKIN)

寄存器	偏移地址	读/写	描述	复位值
USPI_CLKIN x = 0, 1	USPIx_BA+0x28	R/W	USCI 输入时钟信号配置寄存器	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							SYNCSEL

位	描述	
[31:1]	Reserved	保留.
[0]	SYNCSEL	<p>输入同步信号选择</p> <p>此位决定非同步信号（可设置反向）或者同步信号（可设置滤波）作为数据移位单元的输入</p> <p>0 = 非同步信号作为数据移位单元的输入</p> <p>1 = 同步信号作为数据移位单元的输入</p> <p>注意: SPI协议, 建议该位设置为0.</p>

USCI 线控寄存器 (USPI_LINECTL)

寄存器	偏移地址	读/写	描述	复位值
USPI_LINECTL x = 0, 1	USPIx_BA+0x2C	R/W	USCI 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				DWIDTH			
7	6	5	4	3	2	1	0
CTLOINV	Reserved	DATOINV	Reserved				LSB

位	描述	
[31:12]	Reserved	保留.
[11:8]	DWIDTH	传输数据字长 此位定义收发数据的字长（以位为单位），数据右对齐，USCI支持4到16位字长 0x0: 数据字长16位[15:0]. 0x1: 保留 0x2: 保留 0x3: 保留 0x4: 数据字长4位，位于[3:0]. 0x5: 数据字长5位，位于[4:0]. ... 0xF: 数据字长15位，位于[14:0].
[7]	CTLOINV	控制信号输出反向选择 此位定义了内部控制信号与输出控制信号的关系 0 = 不反转 1 = 控制信号反向后再输出 注意: 在不同协议中控制信号有不同的定义. SPI协议中,控制信号作为从机选择信号.
[6]	Reserved	保留.
[5]	DATOINV	数据输出反向选择 此位定义内部移位数据与USCIx_DAT0/1管脚输出信号的关系 0 = 数据输出不反转 1 = 数据输出反转
[4:1]	Reserved	保留.

[0]	LSB	LSB 优先发送选择 0 = MSB优先, 根据发送/接收数据缓冲的数据位宽度决定优先发送/接收位. 1 = LSB优先, 发送/接收数据缓冲位0优先发送/接收.
-----	-----	---

USCI 发送数据寄存器 (USPI_TXDAT)

寄存器	偏移地址	读/写	描述	复位值
USPI_TXDAT x = 0, 1	USPIx_BA+0x30	W	USCI 发送数据寄存器	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
TXDAT							
7	6	5	4	3	2	1	0
TXDAT							

位	描述	
[31:16]	Reserved	保留.
[15:0]	TXDAT	发送数据 用户写要发送的16位数据到此位域，为了防止覆写传送输入，用户需要在写数据前先检测TXEMPTY (USPI_BUFSTS[8])位状态

USCI 接收数据寄存器 (USPI_RXDAT)

寄存器	偏移地址	读/写	描述	复位值
USPI_RXDAT x = 0, 1	USPIx_BA+0x34	R	USCI 接收数据寄存器	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
RXDAT							
7	6	5	4	3	2	1	0
RXDAT							

位	描述	
[31:16]	Reserved	保留.
[15:0]	RXDAT	接收数据 此寄存器保存收到的数据.

USCI 发送/接收缓冲控制寄存器 (USPI_BUFCTL)

寄存器	偏移地址	读/写	描述	复位值
USPI_BUFCTL x = 0, 1	USPIx_BA+0x38	R/W	USCI 发送/接收缓冲控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						RXRST	TXRST
15	14	13	12	11	10	9	8
RXCLR	RXOVLEN	Reserved					
7	6	5	4	3	2	1	0
TXCLR	TXUDRIEN	Reserved					

位	描述	
[31:18]	Reserved	保留。
[17]	RXRST	接收复位 0 = 无动作。 1 = 复位与接收相关的计数器，状态机及接收移位寄存器和缓存中的数据 注意：一个 PCLK 周期后，该位自动清零。
[16]	TXRST	发送复位 0 = 无动作。 1 = 复位与接收相关的计数器，状态机及接收移位寄存器和缓存中的数据 注意：一个 PCLK 周期后，该位自动清零。
[15]	RXCLR	清除接收缓存 0 = 无动作 1 = 接收缓存清零，只有当缓存没有参与数据通信时才使用置位清零 注意：一个 PCLK 周期后，该位自动清零。
[14]	RXOVLEN	接收缓存溢出中断使能位 0 = 接收缓存溢出中断禁止 1 = 接收缓存溢出中断使能
[13:8]	Reserved	保留。
[7]	TXCLR	清除发送缓存 0 = 无动作 1 = 接收缓存清零，只有当缓存没有参与数据通信时才使用置位清零 注意：一个 PCLK 周期后，该位自动清零。

[6]	TXUDRIEN	从机发送欠载中断使能 0 = 发送欠载中断禁止 1 = 发送欠载中断使能
[5:0]	Reserved	保留.

USCI 发送/接收缓冲状态寄存器 (USPI_BUFSTS)

寄存器	偏移地址	读/写	描述	复位值
USPI_BUFSTS x = 0, 1	USPIx_BA+0x3C	R	USCI 发送/接收缓冲状态寄存器	0x0000_0101

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				TXUDRIF	Reserved	TXFULL	TXEMPTY
7	6	5	4	3	2	1	0
Reserved				RXOVIF	Reserved	RXFULL	RXEMPTY

位	描述	
[31:12]	Reserved	保留。
[11]	TXUDRIF	发送缓存欠载中断状态 此位表示发送缓存欠载事件发生，如果TXUDRIEN (USPI_BUFCTL[6])使能，将产生中断，此位软件写1清零 0 = 没有检测到发送缓存欠载事件 1 = 检测到发送缓存欠载事件
[10]	Reserved	保留。
[9]	TXFULL	发送缓存满指示 0 = 发送缓存未满 1 = 发送缓存已满
[8]	TXEMPTY	发送缓存空指示 0 = 发送缓存不为空。 1 = 发送缓存为空，可以装载下次发送数据
[7:4]	Reserved	保留。
[3]	RXOVIF	接收缓存溢出标志 此位表示检测到接收缓存溢出事件，如果RXOVIEN (UUART_BUFCTL[14])使能，相应中断会产生，此位软件写1清零 0 = 没有检测到接收缓存溢出事件 1 = 检测到接收缓存溢出事件
[2]	Reserved	保留。
[1]	RXFULL	接收缓存满标志

		0 = 接收缓存未满 1 = 接收缓存满
--	--	-------------------------

USCI 唤醒控制寄存器 (USPI_WKCTL)

寄存器	偏移地址	读/写	描述	复位值
USPI_WKCTL x = 0, 1	USPIx_BA+0x54	R/W	USCI 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					PDBOPT	Reserved	WKEN

位	描述	
[31:3]	Reserved	保留.
[2]	PDBOPT	掉电模式选项 0 = 如果数据正在收发, MCU运行WFI指令进入掉电模式, 那么MCU将立即停止传输并进入掉电模式 1 = 如果数据正在收发, MCU运行WFI指令进入掉电模式, 此时MCU不会停止传输并且会进入空闲模式
[1]	Reserved	保留.
[0]	WKEN	唤醒使能位 0 = 唤醒功能禁止 1 = 唤醒功能使能

USCI 唤醒状态寄存器 (USPI WKSTS)

寄存器	偏移地址	读/写	描述	复位值
USPI_WKSTS x = 0, 1	USPIx_BA+0x58	R/W	USCI 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							WKF

位	描述	
[31:1]	Reserved	保留.
[0]	WKF	唤醒标志 当芯片从掉电状态唤醒此位置1, 软件写1清零

USCI 协议控制寄存器 – SPI (USPI_PROTCTL)

寄存器	偏移地址	读/写	描述	复位值
USPI_PROTCTL x = 0, 1	USPIx_BA+0x5C	R/W	USCI 协议控制寄存器	0x0000_0300

31	30	29	28	27	26	25	24
PROTEN	Reserved		TXUDRPOL	Reserved		SLVTOCNT	
23	22	21	20	19	18	17	16
SLVTOCNT							
15	14	13	12	11	10	9	8
Reserved	TSMSEL			SUSPITV			
7	6	5	4	3	2	1	0
SCLKMODE		Reserved		AUTOSS	SS	SLV3WIRE	SLAVE

位	描述	
[31]	PROTEN	SPI 协议使能位 0 = SPI 协议禁止 1 = SPI 协议使能
[30:29]	Reserved	保留.
[28]	TXUDRPOL	发送欠载数据极性 (仅从机) 此位定义了没有数据要发送时输出电平 0 = TX欠载后输出低电平 1 = TX欠载后输出高电平.
[27:26]	Reserved	保留.
[25:16]	SLVTOCNT	从机模式超时周期(仅从机) 从机模式下, 此位域定义了从机超时的周期, 即两个输入SCK时钟信号之间最大的时钟 (由TMCNTSRC, USPI_BRGEN[5]选择时钟源) 个数, 超过这个数字将产生从机超时事件; 此位域写0禁止从机超时功能 例如: 设SLVTOCNT=0x0A 且 TMCNTSRC (USPI_BRGEN[5]) = 1, 意味着如果SPI时钟信号在传输过程中超过 (10+1) 个fDIV_CLK时钟周期没有变化, 就会产生超时事件
[15]	Reserved	保留.
[14:12]	TSMSEL	发送数据模式选择 此位域描述数据收发的模式 TSMSEL = 0: 全双工 SPI. 其他值保留. 注意: 改变此位域的设置将导致TXRST 及 RXRST自动清除收发数据缓存
[11:8]	SUSPITV	传输暂停 间隔(仅主机)

		<p>此位域可以设置两个连续的收发数据间的暂停时间，暂停间隔的定义指的是前一个数据最后一个时钟信号边沿到下一个数据的第一个时钟信号边沿，其默认值是 SUSPITV (USPI_PROTCTL[11:8]) = 0x3 (3.5个SPI时钟周期)；暂停时间设置如下：</p> <p>(SUSPITV[3:0] + 0.5) * SPI_CLK 时钟周期</p> <p>例如：</p> <p>SUSPITV = 0x0 ... 0.5 SPI_CLK 时钟周期.</p> <p>SUSPITV = 0x1 ... 1.5 SPI_CLK 时钟周期.</p> <p>.....</p> <p>SUSPITV = 0xE ... 14.5 SPI_CLK 时钟周期.</p> <p>SUSPITV = 0xF ... 15.5 SPI_CLK 时钟周期.</p>
[7:6]	SCLKMODE	<p>串行总线时钟模式</p> <p>此位域定义时钟信号在空闲及收发数据时的状态</p> <p>MODE0 = 时钟信号在空闲时为低，下降沿发送数据，上升沿接收数据</p> <p>MODE1 = 时钟信号在空闲时为低，上升沿发送数据，下降沿接收数据</p> <p>MODE2 = 时钟信号在空闲时为高，上升沿发送数据，下降沿接收数据</p> <p>MODE3 = 时钟信号在空闲时为高，下降沿发送数据，上升沿接收数据</p>
[5:4]	Reserved	保留.
[3]	AUTOSS	<p>自动从机选择功能使能（仅主机）</p> <p>0 = 通过设置SS (USPI_PROTCTL[2])控制从机选择信号</p> <p>1 = SPI控制器在收发数据时自动产生从机选择信号，收发完成后信号自动失效</p>
[2]	SS	<p>从机选择控制 (仅主机)</p> <p>如果AUTOSS=0，设置此位为1将使能从机选择信号，为0则不使能从机选择信号</p> <p>如果AUTOSS=1，设置此位对从机选择信号没有影响</p> <p>注意: SPI协议下，内部从机选择信号高有效</p>
[1]	SLV3WIRE	<p>从机3线模式选择(仅从机)</p> <p>从机模式下该位置1，SPI 3线工作模式（没有从机选择信号）</p> <p>0 = 4线双向接口</p> <p>1 = 3线双线接口</p>
[0]	SLAVE	<p>从机模式选择</p> <p>0 = 主机模式</p> <p>1 = 从机模式</p>

USCI 协议中断使能寄存器 – SPI (USPI_PROTIEN)

寄存器	偏移地址	读/写	描述	复位值
USPI_PROTIEN x = 0, 1	USPIx_BA+0x60	R/W	USCI 协议中断使能寄存器	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							
7	6	5	4	3	2	1	0
Reserved				SLVBEIEN	SLVTOIEN	SSACTIEN	SSINAIEN

位	描述	
[31:4]	Reserved	保留。
[3]	SLVBEIEN	<p>从机模式位计数错误中断使能位</p> <p>从机模式下，如果发送/接收的数据位数与DWIDTH (USPI_LINECTL[11:8]) 设定不一致（如果通过从机超时或选择信号无效结束数据传输），会触发该中断</p> <p>0 = 从机模式位计数错误中断禁止</p> <p>1 = 从机模式位计数错误中断使能</p>
[2]	SLVTOIEN	<p>从机超时中断使能位</p> <p>SPI协议下，此位使能从机超时中断</p> <p>0 = 从机超时中断禁止</p> <p>1 = 从机超时中断使能</p>
[1]	SSACTIEN	<p>从机选择有效中断使能位</p> <p>此位使能/禁止从机选择信号有效中断</p> <p>0 = 从机选择信号有效中断禁止</p> <p>1 = 从机选择信号有效中断使能</p>
[0]	SSINAIEN	<p>从机选择无效中断使能位</p> <p>此位使能/禁止从机选择信号无效中断</p> <p>0 = 从机选择无效中断禁止</p> <p>1 = 从机选择无效中断使能</p>

USCI 协议状态寄存器 – SPI (USPI_PROTSTS)

寄存器	偏移地址	读/写	描述	复位值
USPI_PROTSTS x = 0, 1	USPIx_BA+0x64	R/W	USCI 协议状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					SLVUDR	BUSY	SSLINE
15	14	13	12	11	10	9	8
Reserved						SSACTIF	SSINAIF
7	6	5	4	3	2	1	0
Reserved	SLVBEIF	SLVTOIF	RXENDIF	RXSTIF	TXENDIF	TXSTIF	Reserved

位	描述	
[31:19]	Reserved	保留.
[18]	SLVUDR	<p>从机模式发送欠载状态(只读)</p> <p>从机模式下, 如果发送缓存没有数据, 却收到主机的时钟信号, 此标志位置1; 此位也标志着移位寄存器是否发送TXUDRPOL (USPI_PROTCTL[28])设定的数据</p> <p>0 = 从机发送欠载事件没有发生</p> <p>1 = 发生了从机发送欠载事件</p>
[17]	BUSY	<p>总线状态 (只读)</p> <p>0 = SPI 在空闲状态</p> <p>1 = SPI 在繁忙状态</p> <p>总线繁忙状态如下:</p> <p>a. USPI_PROTCTL[31] = 1且 TXEMPTY = 0.</p> <p>b. SPI主机模式下, TXEMPTY = 1 但是当前传输还没结束</p> <p>c. SPI从机模式下, USPI_PROTCTL[31] = 1并且从机选择信号有效, 有时钟输入SPI内核逻辑</p> <p>d. SPI从机模式下, USPI_PROTCTL[31] = 1 , 即便从机选择无效但是发送缓冲或发送移位寄存器不为空时</p>
[16]	SSLINE	<p>从机选择信号状态(只读)</p> <p>此位只在从机模式下有效, 用来监控从机选择信号是否有效</p> <p>0 = 从机选择信号状态为0</p> <p>1 = 从机选择信号状态为1.</p>
[15:10]	Reserved	保留.
[9]	SSACTIF	从机选择信号有效中断标志(仅从机)

		<p>此位显示从机选择信号变为有效，软件写1清零</p> <p>0 = 从机选择信号没有变为有效</p> <p>1 = 从机选择信号变为有效</p> <p>注意: 内部从机选择信号高电平有效</p>
[8]	SSINAIF	<p>从机选择信号无效中断标志(仅从机)</p> <p>此位显示从机选择信号变为无效，软件写1清零</p> <p>0 = 从机选择信号没有变为无效</p> <p>1 = 从机选择信号变为无效</p> <p>注意: 内部从机选择信号高电平有效</p>
[7]	Reserved	保留.
[6]	SLVBEIF	<p>从机位计数错误中断标志(仅从机)</p> <p>0 = 从机位计数错误未发生.</p> <p>1 = 从机位计数错误发生了</p> <p>注 意: 通过软件写1清零该位</p>
[5]	SLVTOIF	<p>从机超时中断标志(仅从机)</p> <p>0 = 从机超时事件没有发生</p> <p>1 = 从机超时事件发生了</p> <p>注 意: 通过软件写1清零该位</p>
[4]	RXENDIF	<p>接收结束中断标志</p> <p>0 = 没有接收结束中断</p> <p>1 = 有接收结束中断产生</p> <p>注 意: 通过软件写1清零该位</p>
[3]	RXSTIF	<p>接收开始中断标志</p> <p>0 = 没有接收开始中断</p> <p>1 = 有接收开始中断产生</p> <p>注 意: 通过软件写1清零该位</p>
[2]	TXENDIF	<p>发送结束中断标志</p> <p>0 = 没有发送结束中断</p> <p>1 = 有发送结束中断产生</p> <p>注 意: 通过软件写1清零该位</p>
[1]	TXSTIF	<p>发送开始中断标志</p> <p>0 = 没有发送开始中断</p> <p>1 = 有发送开始中断产生</p> <p>注 意: 通过软件写1清零该位</p>
[0]	Reserved	保留.

6.14 USCI – I²C 模式

6.14.1 概述

I²C总线主从通信时，数据与时钟信号在SCL, SDA线上同步变化，数据字节长度为8位，MSB优先发送，每个数据位对应一个SCL时钟信号，数据字节发送完后跟一个响应信号位。SCL为高电平时数据位采样，所以SCL为低电平时才能改变数据线（SDA）的信号，SCL为高电平时数据信号必须保持稳定；SCL为高电平时SDA信号的变化被认为是命令（开始或停止），I²C总线的详细时序请参考图 6.14-1。

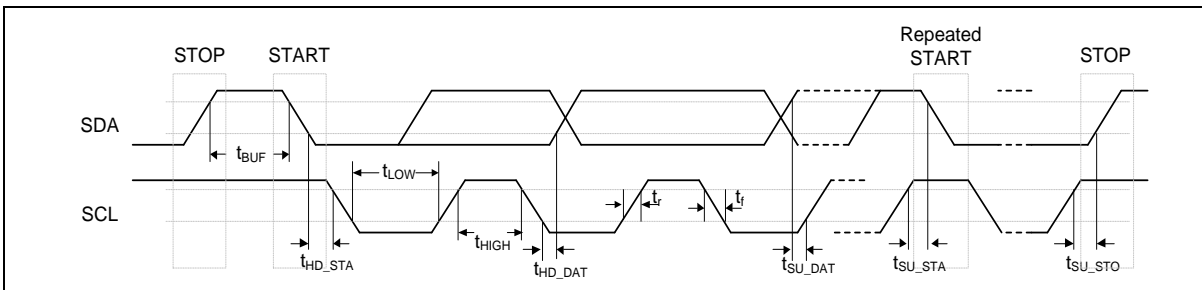


图 6.14-1 I²C 总线时序

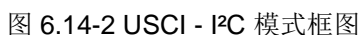
片上I²C模块提供满足标准I²C协议的接口，自动处理数据的传输；选择I²C模式需要设置FUNMODE (UI2C_CTL [2:0]) = 100B，当此I²C端口使能，USCI接口用于I²C总线管脚：SDA和SCL。I/O管脚用于I²C端口前用户必须先设置管脚功能为I²C。

注意：当USCI配置为I²C 模式后SDA,SCL信号处于开漏输出状态，必须外接上拉电阻。

6.14.2 特征

- 支持主从机功能
- 支持7位地址及10位地址
- 支持标准模式（100 kBit/s）或快速模式(最高到400 kBit/s)通讯
- 支持多主机模式
- 支持一个发送缓存和两个接收缓存
- 支持10位超时功能
- 支持地址匹配或数据信号变化唤醒功能
- 支持设置/保持时间可编程配置

6.14.3 框图



USC10用于I2C基本配置如下:

- USCI1 用于 I²C 基本配置如下:

- #### 6.14.5.1 USCI 通用功能描述

Rev.1.00

6.14.5.2 开始或重复开始信号

图 6.14-3展示了典型的I²C 协议，标准通信包括4部分：

- 产生开始或重复开始信号
- 从机地址及读/写位的传输
- 数据传输
- 产生停止信号

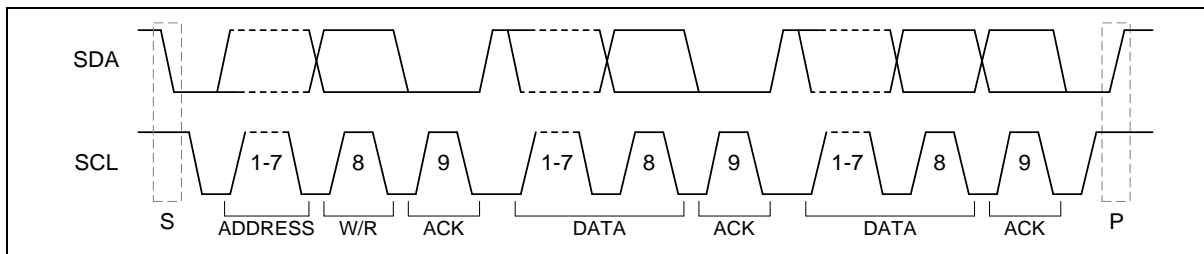


图 6.14-3 I²C 协议

总线空闲指的是没有主机占用总线（SCL和SDA都是高电平），在此状态下主机可以发送开始信号启动一次传输，开始信号也称作“S”位，定义为当SCL为高电平时SDA线上由高到低的变化。开始信号是一次新传输的开始。

开始信号到结束信号之间可以有重复开始信号，一般表示为“Sr”位,主机用这种方式启动与其他从机的通信或者启动同一个从机不同方向的传输（例如，从写设备到从设备读），而不释放总线的空闲标志。

6.14.5.3 停止信号

主机可以发送停止信号终结通信，停止信号一般表示为“P”位，指的是SCL为高电平时SDA线上由低到高的电平变化，停止信号到开始信号之间叫做总线空闲。图 6.14-4展现了开始、重复开始及停止信号的波形。

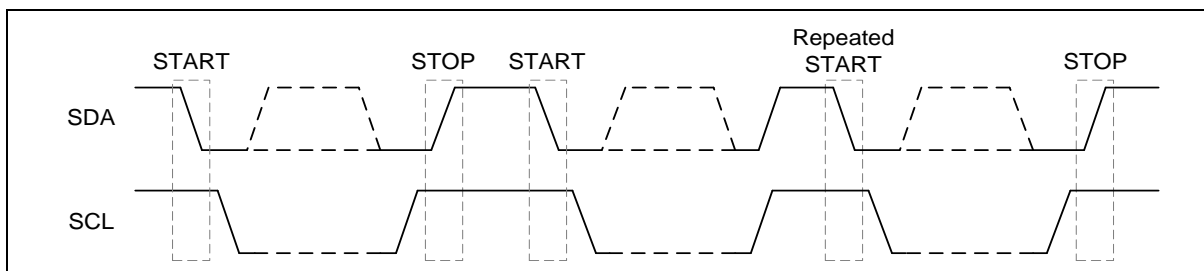


图 6.14-4 开始、重复开始和停止信号波形

6.14.5.4 从机地址传输

开始或重复开始信号后，主机发送从机地址以表示要跟对应从设备通讯。从机地址由一个或两个字节组成（7位或10位地址机制），从机检测到发送的地址并做出响应。

从机被设置地址后会将收到的地址数据跟自己的地址作比对，如果相同那么从机会回复响应信号（SDA=0），没有被寻址的从机不响应（SDA=1）。如果被寻址的从机支持10位地址模式，另外一个地址数据也会回复响应信号。地址00H表示广播地址。

为了对不同的地址作选择性响应，实施了下列控制机制：

- GCFUNC 位 (UI2C_PROTCTL [0])设置I²C硬件响应广播地址（00H），清除GC位禁止广播功能
- I²C端口有一个从机地址寄存器UI2C_DEVADDR0，7位地址模式时，收到的地址数据的前7位与编程设定的从机地址(UI2C_DEVADDR0 [6:0])作比对，如果匹配从机发送响应信号。
- 此外，当ADDR10EN (UI2C_PROTCTL [4])=1（使能10位地址模式），如果收到的从机地址为1111 0XXB，xx位将与位UI2C_DEVADDR0 [9:8]作比对，如果匹配则从机回复响应信号，然后等待第二个地址字节，并与UI2C_DEVADDR0 [7:0]作比对，匹配后回复响应信号。用户要小心保留地址（详细描述请参照I2C规格书），仅仅支持地址1111 0XXB。当地址匹配后位SLASEL (UI2C_PROTSTS [14])将被置1，此位将被（重复）开始或停止信号清零。
- I²C端口有一个地址位屏蔽寄存器I2C_ADDRMSKn (n = 0)，支持多地址识别功能，当屏蔽地址位设置为1表示接收到的响应地址位“不用关心”，如果屏蔽位为0表示接收到的数据必须与地址寄存器中设定的值相同才能被识别。

6.14.5.5 数据传输

当从机接收到一个包含读/写位的正确地址，接下来数据将根据读/写位定义的方向传输。每个传输字节的第9个时钟接收方都发出响应信号，如果从机发出不响应(NACK)信号，主机可以产生停止信号停止传输，或者产生重新开始信号开始一个新的传输过程。

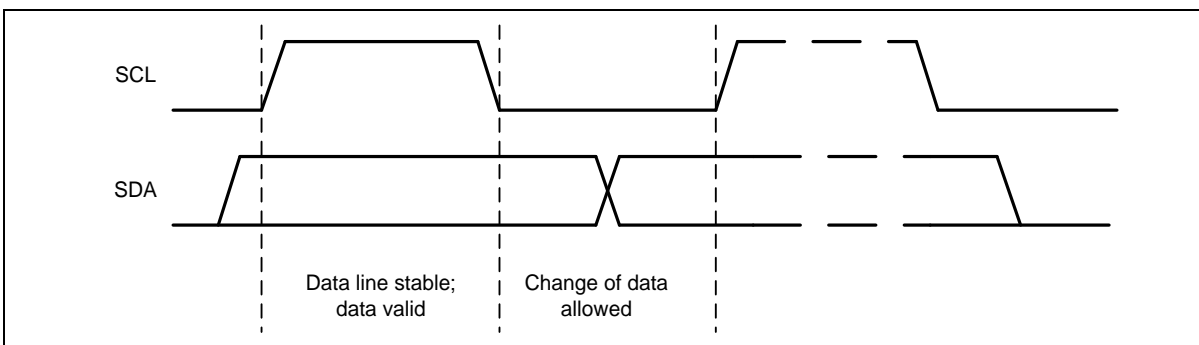


图 6.14-5 I²C 总线位传输

如果主机接收数据不发响应信号（NACK）给从机，从机将释放SDA线给主机以便主机发送停止或重新开始信号。

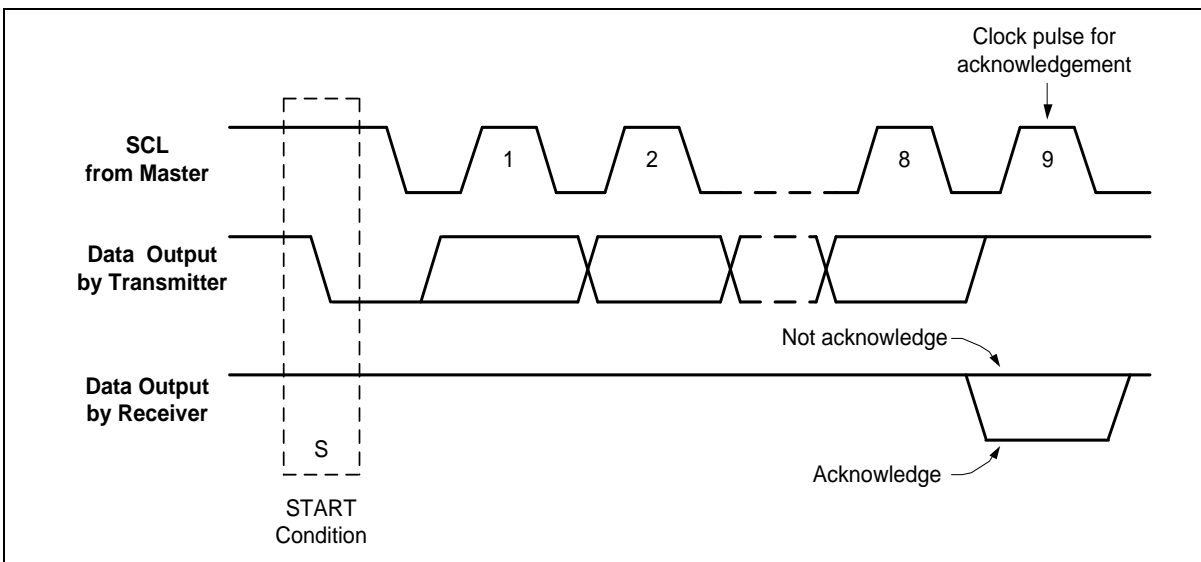


图 6.14-6 I²C 总线应答

6.14.5.6 时钟波特率位

此部分请参考 图 6.11-9. 当I²C在主机模式时其波特率由寄存器UI2C_BRGEN决定, 从机模式与寄存器UI2C_BRGEN无关, 从机模式时I²C会自动与任何波特率主机同步, 位RCLKSEL, SPCLKSEL, PDSCNT, DSCNT定义了波特率的设定:

- RCLKSEL (UI2C_BRGEN [0])

定义输入频率 f_{REF_CLK}

- SPCLKSEL (UI2C_BRGEN[3:2])

选择采样时钟 f_{SAMP_CLK} 的时钟源

- PDSCNT (UI2C_BRGEN [9:8])

定义一个采样时间的长度(f_{REF_CLK} 除以 1, 2, 3, 或 4)

- DSCNT (UI2C_BRGEN [14:10])

定义一个数据位的采样时间个数

标准设定如下: RCLKSEL = 0 ($f_{REF_CLK} = f_{PCLK}$), PTCLKSEL = 0 ($f_{PROT_CLK} = f_{REF_CLK}$) 和 SPCLKSEL = 2'b00 ($f_{SAMP_CLK} = f_{DIV_CLK}$). 在此条件下波特率计算如下:

$$f_{I2C} = \frac{f_{REF_CLK}}{2} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

为了产生慢一些的频率, 可以通过设置PTCLKSEL = 1 ($f_{PROT_CLK} = f_{REF_CLK2}$) 将所选时钟频率除以2:

$$f_{I2C} = \frac{f_{REF_CLK}}{4} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

如果SPCLKSEL = 2'b10 ($f_{SAMP_CLK} = f_{SCLK}$), 并且 RCLKSEL = 0 ($f_{REF_CLK} = f_{PCLK}$), PTCLKSEL = 0 ($f_{PROT_CLK} = f_{REF_CLK}$). 那么波特率计算如下:

$$f_{I2C} = \frac{f_{REF_CLK}}{2} \times \frac{1}{CLKDIV + 1} \times \frac{1}{2} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

6.14.5.7 字节拉长

如果I²C设备要发送数据，但TXDAT中还没准备好要发送的数据，此时发送设备可以在上次传输的响应位将SCL拉低为0，数据接收方将会等待，一直到发送方写1到寄存器PTRG (UI2C_PROTCTL [5])解除拉长状态。

6.14.5.8 主机仲裁

一些应用中会有2个或2个以上主机在同一条I²C总线上与从机通信，主机可能同时发送数据，I²C支持冲突检测及仲裁功能，以防止通信数据丢失。

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并做相应的处理。图 6.14-7描述了DATA1 和 DATA2的仲裁。

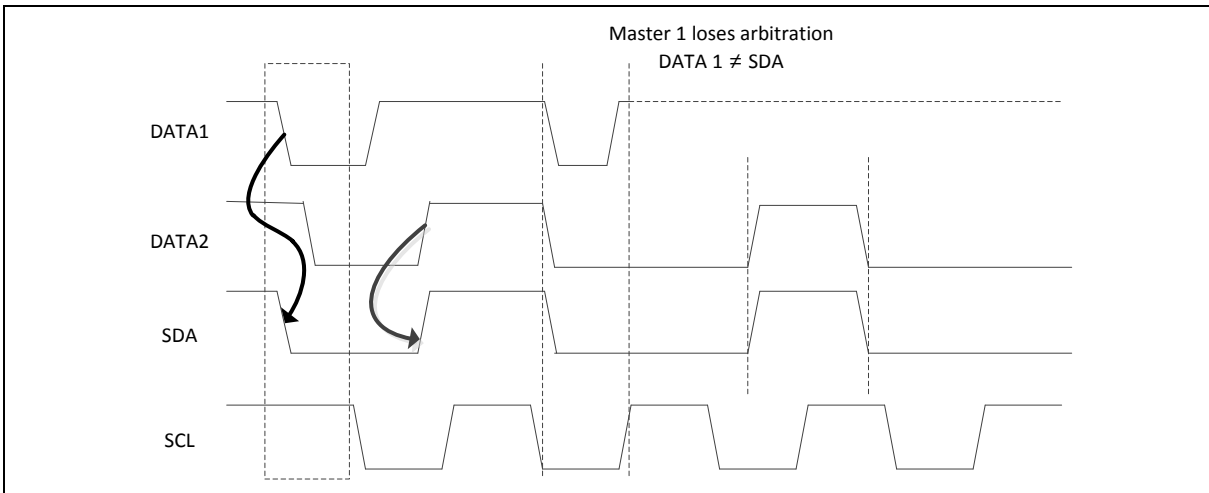


图 6.14-7 仲裁丢失

地址和数据发送过程中，主机发送器会在SCL的上升沿检测SDA线上的数据是否跟自己发出去的一致，如果不一致（发送1，读到0）那么该主机仲裁丢失，中断标志ARBLOIF (UI2C_PROTSTS [11])会显示这种情况，如果相应中断使能ARBLOIEN (UI2C_PROTIEN [4])则会发生中断。

发生仲裁丢失后，软件必须重新初始化完整数据帧，包括开始信号及地址字节。响应位（ACK）也会发生仲裁，如果主机仲裁丢失并发现地址匹配事件，主机将变为从机。

6.14.5.9 发送链

I²C总线协议在仲裁或从机发送数据阶段要求一种位反应的机制，这种发送链的延迟效果限制了可以达到的最高波特率，并且跟总线的状态相关（总线负载，模块频率等等）

主机产生移位时钟SCL经过输入阶段，输入滤波，检测到边沿信号然后产生SDA信号，SDA信号经过输出阶段到达主机接收部分，再经过输入阶段，输入滤波最后采样。

这个完整的循环（包括所有时间设置以获得稳定信号）必须在SCL变化前完成，这些延时在以 f_{PCLK}

或 $f_{\text{PROT_CLK}}$ 时钟计算波特率时必须考虑在内，我们建议采用 f_{PCLK} 时钟

6.14.5.10 无应答和错误条件

如果发生无响应（NACKIF (UI2C_PROTSTS [10])）或错误状态（ERRIF (UI2C_PROTSTS [12])），后续传输将停止，用户软件需要重新配置传输（写数据到TXDAT）以应对上次的错误。

I²C 协议中断事件

I²C模式可以产生下列协议相关事件并产生中断

请注意，寄存器UI2C_PROTSTS位不能被硬件自动清零，为了能够随时监控发生的事件，需要用用户软件清除寄存器该位状态。

- 数据帧正确位置接收到起始信号 (STARIF (UI2C_PROTSTS [8]))
- 数据帧正确位置接收到停止信号 (STORIF (UI2C_PROTSTS [9]))
- 主机仲裁丢失 (ARBLOIF (UI2C_PROTSTS [11]))
- 收到从机读请求 (SLAREAD (UI2C_PROTSTS [15]))
- 收到响应信号 (ACKIF (UI2C_PROTSTS [13]))
- 收到不响应信号 (NACKIF (UI2C_PROTSTS [10]))
- 开始信号不在正确位置 (ERRIF (UI2C_PROTSTS [12]))
- 停止信号不在正确位置 (ERRIF (UI2C_PROTSTS [12]))

6.14.5.11 I²C操作

操作I²C必须考虑下列问题:

选择 I²C 模式

建议用户在FUNMODE (UI2C_CTL [2:0]) = 000B时配置好所有I²C在运行时不会改变的参数，以避免意外引起的干扰信号，然后再设置FUNMODE (UI2C_CTL [2:0]) = 100b使能I²C模式。

-步骤 1. 设置 FUNMODE (UI2C_CTL [2:0]) = 000B

-步骤 2. 设置 FUNMODE (UI2C_CTL [2:0]) = 100B

管脚连接

SDA,SCL所用的管脚需设置为开漏输出模式，USCI控制器支持“线与”架构的I²C总线。

注意: I²C模式使能后再去使能不同端口的输出功能，以防止出现干扰信号。

位时序配置

标准模式下（100 kBit/s）I²C模块需要的最低频率为2MHz，高速模式（400 kBit/s）最低频率为10MHz，此外如果要使用数字滤波除掉50ns的尖峰信号，滤波器需要20MHz的时钟。如果另外一个I²C设备参与到总线通信并拉长了SCL低电平时间，SCL为高电平时或许会有一个最长 $1/f_{\text{PROT_CLK}}$ 的不确定状态。请注意SCL最高频率为SAMP_CLK/2 并且 SPCLKSEL (UI2C_BRGEN [3:2]) = 0

选择 $f_{\text{SAMP_CLK}} = f_{\text{DIV_CLK}}$

数据格式配置

数据格式必须配置为8位(DWIDTH (UI2C_LINECTL [11:8]) = 8), MSB优先(LSB (UI2C_LINECTL [0]) = 0), 所以必须配置UI2C_LINECTL=0x800.

控制流

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 端口可以作为主机或从机. 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UI2C_PROTSTS的当前状态码来设置UI2C_PROTCTL, UI2C_PROTIEN, TXDAT寄存器. 换句话说, 每个I2C总线动作都要检查UI2C_PROTSTS寄存器的当前状态, 然后再设置UI2C_PROTCTL, UI2C_PROTIEN, TXDAT寄存器执行总线动作. 最后, 通过UI2C_PROTSTS检查响应状态.

在中断标志被清除后, 寄存器UI2C_PROTCTL的这些位STA, STO 和 AA 用来控制I2C硬件的下一个状态. 当完成一个新的动作, UI2C_PROTSTS的状态代码将被更新, 如果I2C中断控制位UI2C_PROTIEN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.

图 6.14-8表示当前I2C状态被硬件设置为STARIF (UI2C_PROTSTS [8])=1, 然后通过设置TXDAT = SLA+W (从机地址 + 写), (PTRG, STA, STO, AA) = (1, 0, 0, x)发送地址到I2C总线, 写1到STARIF (UI2C_PROTSTS [8])清除状态标志. 如果总线上有从机匹配相应的地址且返回ACK, UI2C_PROTSTS状态码更新为ACKIF (UI2C_PROTSTS [13]).

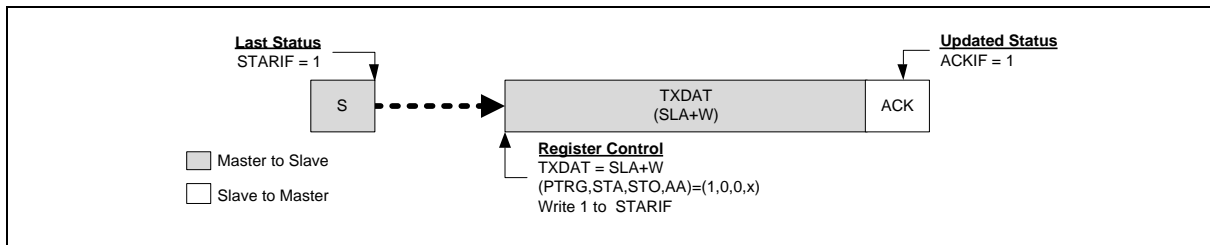


图 6.14-8 根据当前 I²C 状态控制 I²C 总线

I²C 总线上数据传输

图 6.14-9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。.

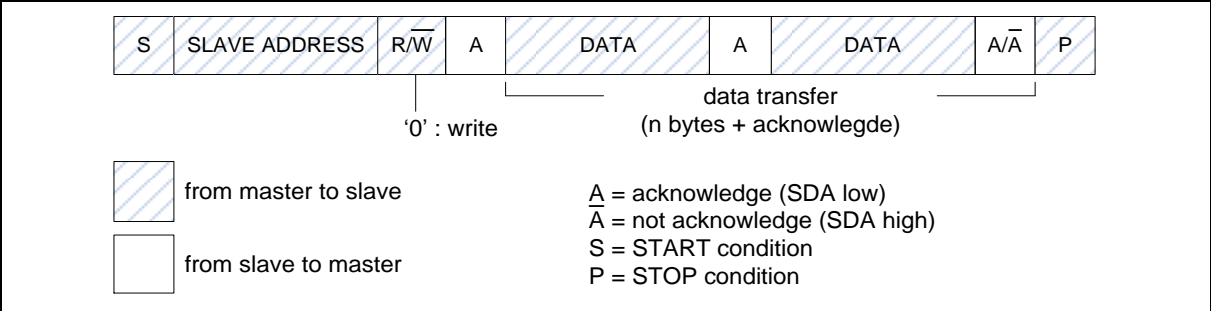


图 6.14-9 主机通过 7 位地址发送数据到从机

图 6.14-10表示主机读取从机数据。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。

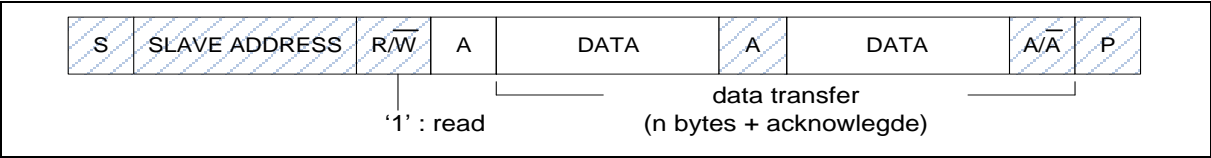


图 6.14-10 主机通过 7 位地址读从机数据

图 6.14-11表示主机通过10位地址发送数据到从机。主机通过10位地址寻址从机: 第一个字节包含10位地址指示 (5'b11110), 2位地址及写指示, 第二个字节包含剩余的8位地址, 主机在第二个字节结束后继续传送数据。注意: 7位地址和10位地址设备可以同时工作在同一总线上。

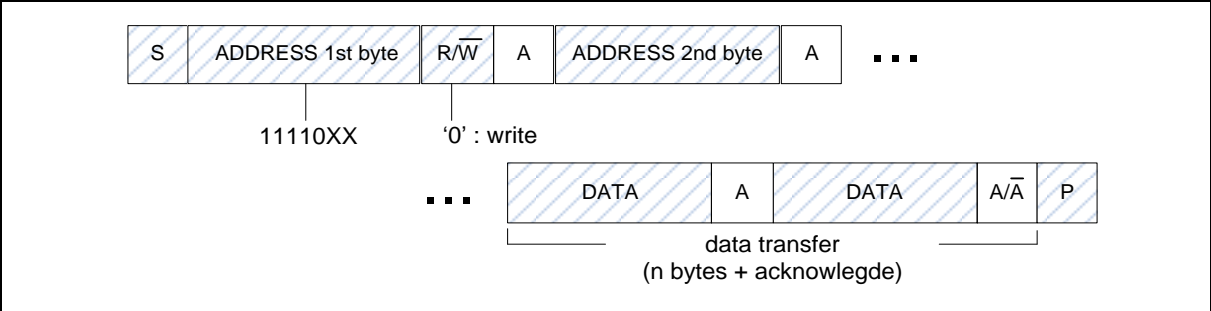


图 6.14-11 主机通过 10 位地址发送数据到从机

图 6.14-12表示主机通过10位地址读从机。主机通过10位地址寻址从机: 第一个字节包含10位地址指示 (5'b11110), 2位地址及写指示, 第二个字节包含剩余的8位地址; 主机发送重新开始信号并发送第一个地址字节包含10位地址指示 (5'b11110), 2位地址及读指示, 从机应答后开始传送数据给主机。

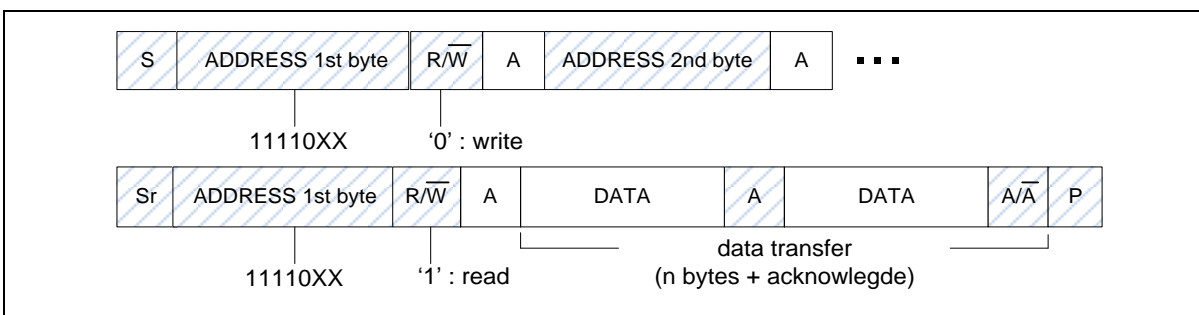


图 6.14-12 主机通过 10 位地址读从机数据

主机模式

图 6.14-13. 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总线将设置成主机传输模式(图 6.14-13) 或者主机接收模式(图 6.14-14) 起始信号设置成功后新的状态码STARIF (UI2C_PROTSTS [8])将被置1。起始信号后，用户可以发送从机地址，读/写位，数据和重新开始，停止来执行I2C协议。

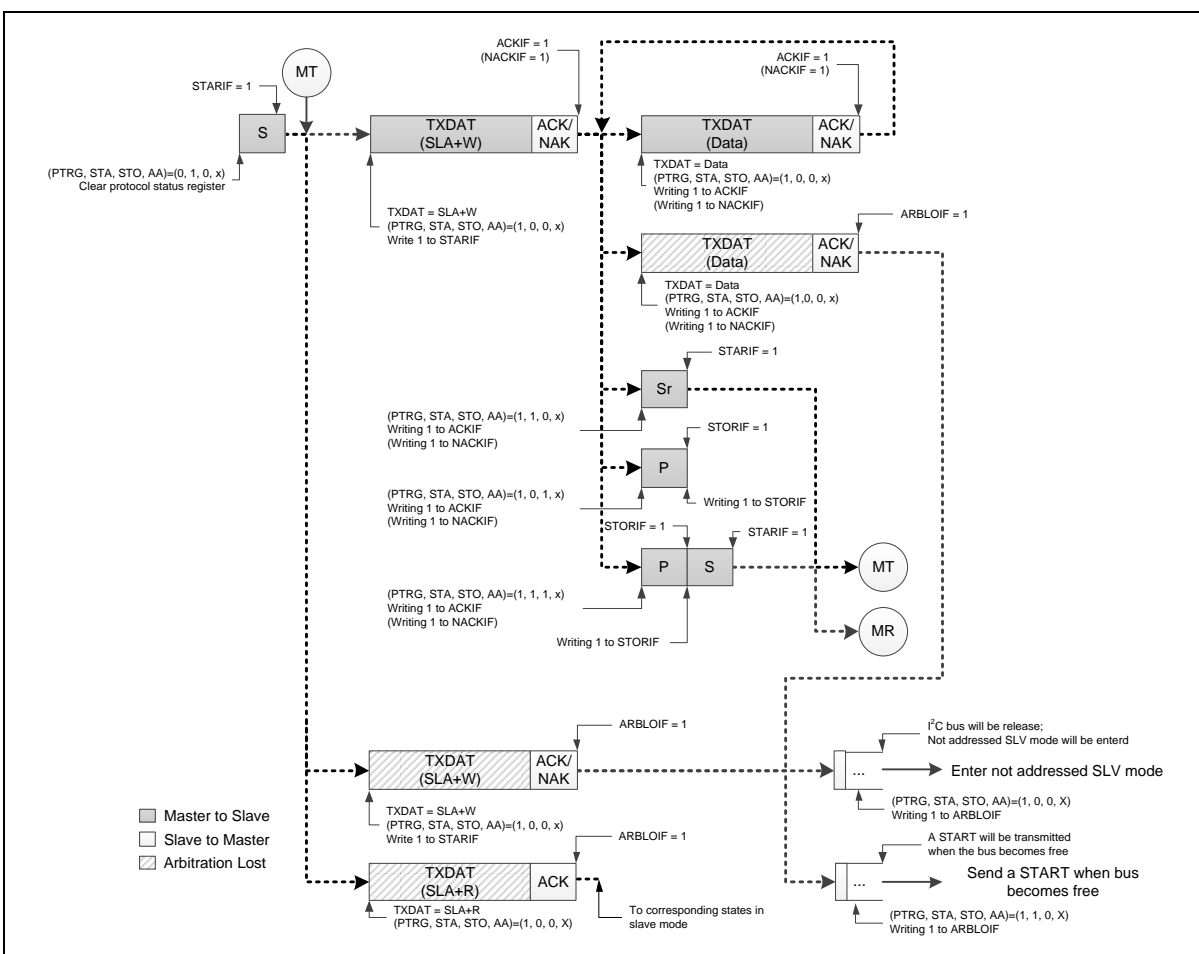


图 6.14-13 7 位地址主机传输模式控制流程

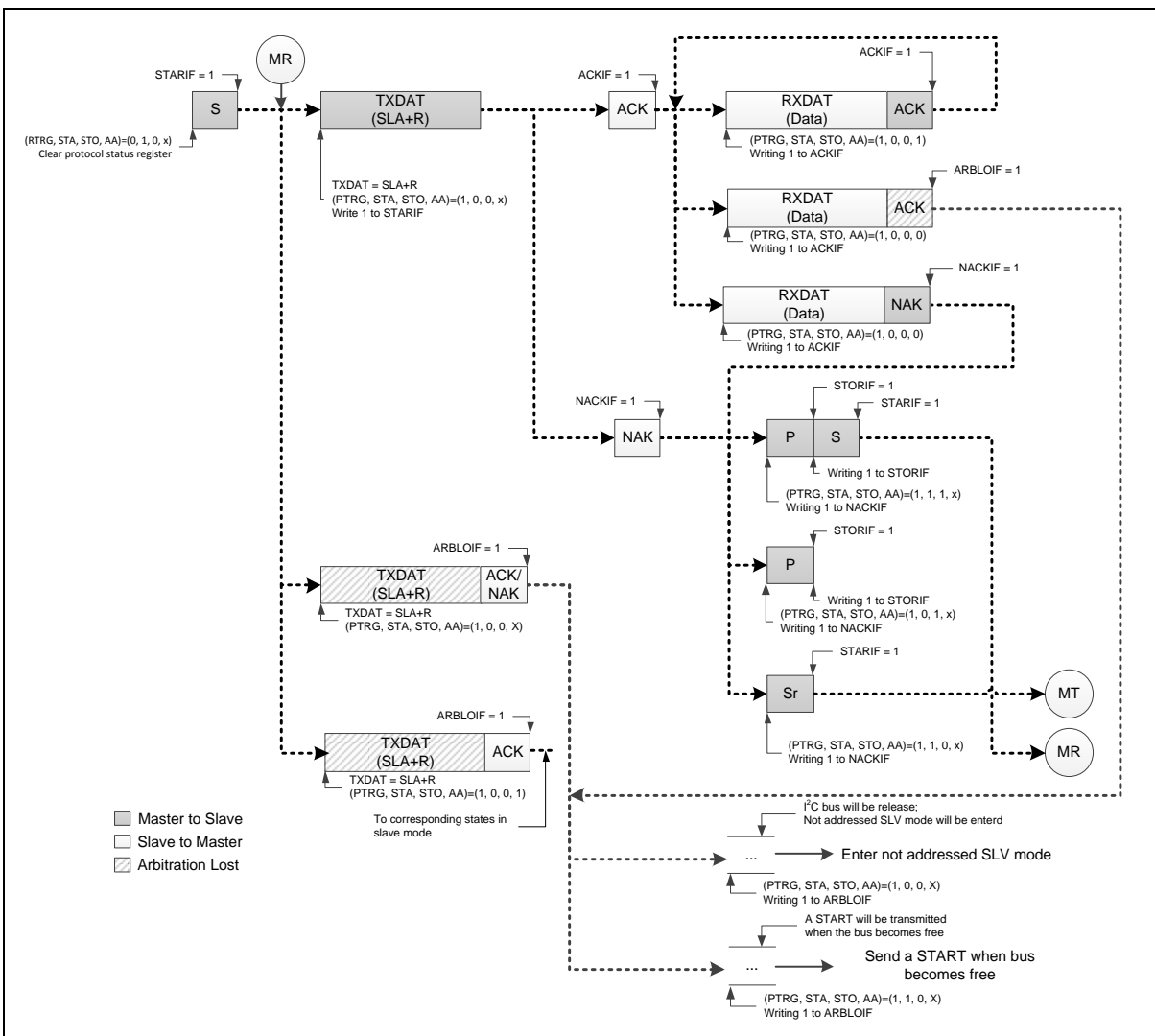


图 6.14-14 主机 7 位地址接收模式控制流程

如果I2C在主机模式并且仲裁丢失，状态码ARBLOIF (UI2C_PROTSTS [11])=1。当总线空闲时，用户可以设置ARBLOIF (UI2C_PROTSTS [11]) =1，PTRG, STA, STO, AA) = (1, 1, 0, X)发送起始信号来重新开始主机操作。否则，用户可以设置ARBLOIF (UI2C_PROTSTS [11]) =1，(PTRG, STA, STO, AA) = (1, 0, 0, X)来释放总线，并进入无地址从机模式。

从机模式

复位后默认情况下，I2C不会被寻址，并且不会识别I2C总线上的地址。用户可以通过UI2C_DEVADDR0设置从机地址和设置(PTRG, STA, STO, AA) = (1, 0, 0, 1)来让I2C识别主机发送的地址。图 6.14-15从机7位地址模式控制流程所示为I2C从机模式的所有可能的流程。用户需要遵循相应流程 (如图 6.14-15从机7位地址模式控制流程的流程来实现他们的I2C协议)。

如果在主机模式总线仲裁丢失，I2C端口立即切换到从机模式，并且在本次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）或SLA+R（主机向从机读数据），ARBLOIF将被置1。

注意: 在I²C从机模式通信中，写1到PTRG (UI2C_PROTCTL [5])，SCL时钟将被释放。

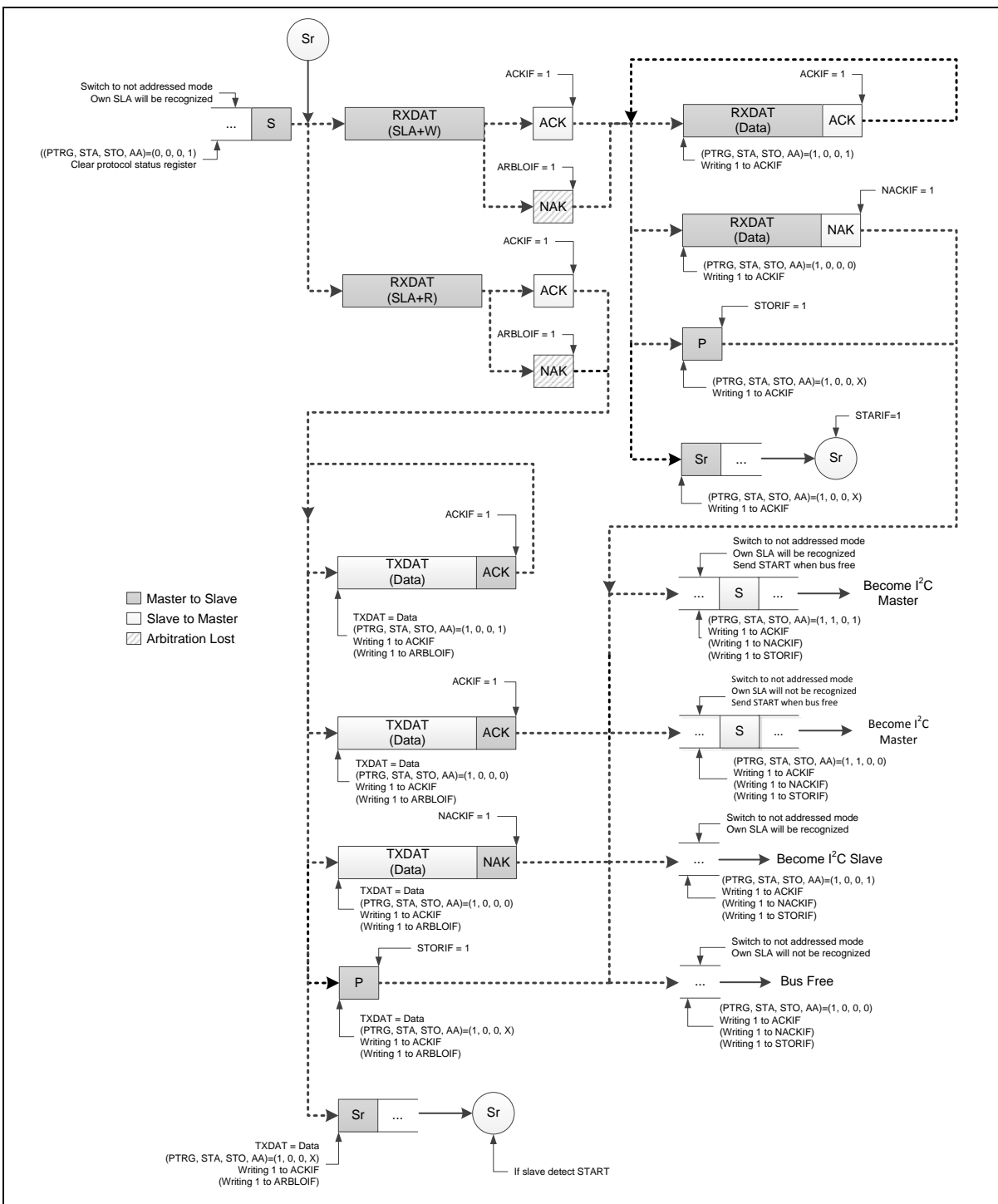


图 6.14-15 从机 7 位地址模式控制流程

如果I2C在从机模式传输数据时却收到停止或重复起始信号，寄存器STORIF (UI2C_PROTSTS [9]) 或 STARIF (UI2C_PROTSTS [8])将被置1。当STARIF (UI2C_PROTSTS [8])置1，用户可以遵循如图 6.14-15 NACKIF (UI2C_PROTSTS [10])的流程处理。

注意：从机如果收到NACKIF (UI2C_PROTSTS [10])，STARIF (UI2C_PROTSTS [8]) 及 STORIF

(UI2C_PROTSTS [9])中断标志，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，需要设置FUNMODE (UI2C_CTL [2:0]) = 000B复位I2C模块才能离开这种状态。

广播 (GC) 模式

如果GCFUNC 位 (UI2C_PROTCTL [0])被设置，I2C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I2C在从机模式时且GC 位被设置，可以接收主机发送的广播地址(0x00)呼叫，将遵循广播模式状态

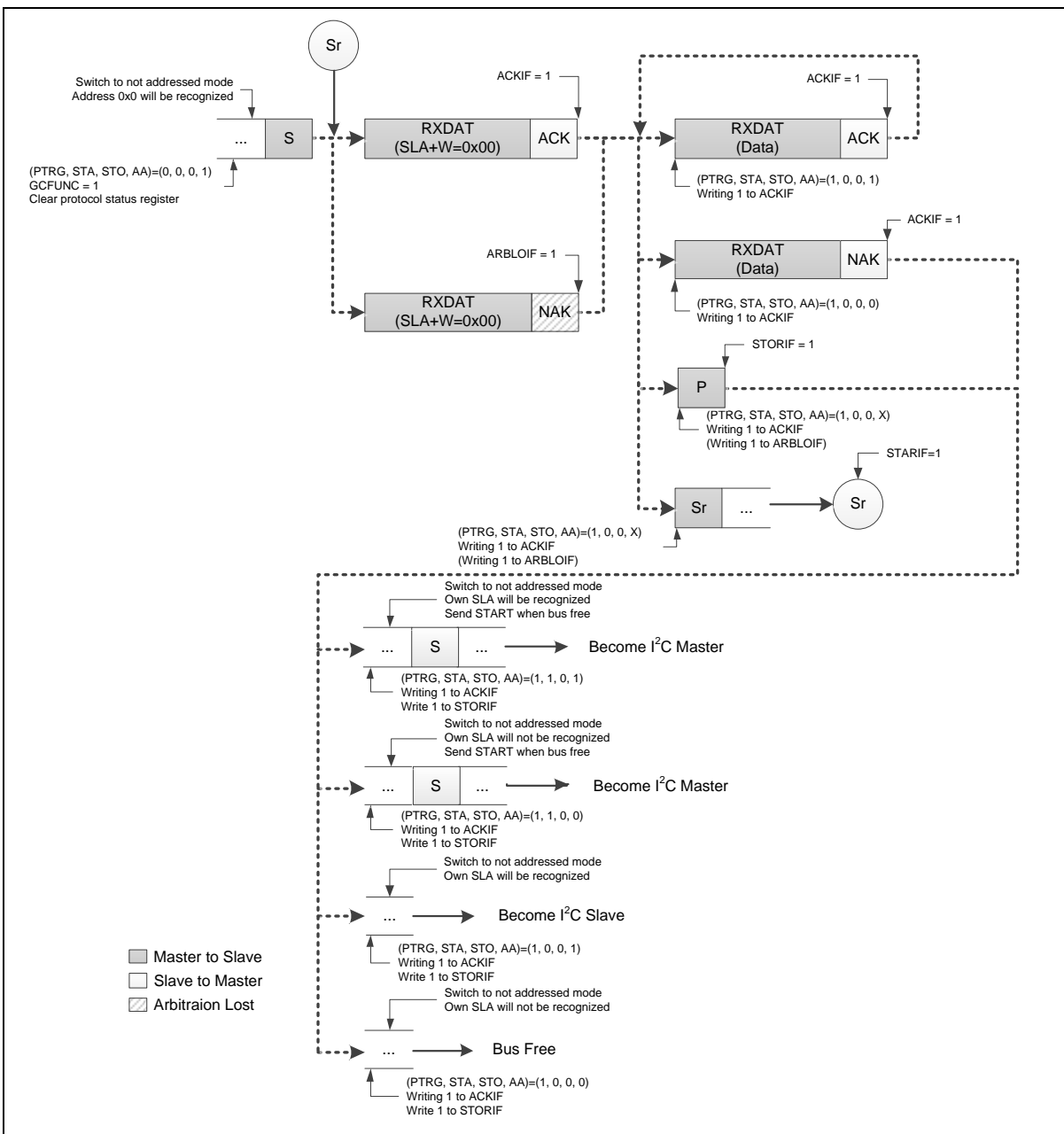


图 6.14-16 7 位地址广播模式

如果I2C在广播呼叫模式接收数据时，却收到停止或重新开始信号，STORIF (UI2C_PROTSTS [9])

或 STARIF (UI2C_PROTSTS [8])将被置1，当 STORIF (UI2C_PROTSTS [9]) 或 STARIF (UI2C_PROTSTS [8])被置1，用户可以遵循如图 6.14-16状态NACKIF (UI2C_PROTSTS [10])的流程处理。

注意：从机收到NACKIF (UI2C_PROTSTS [10])， STARIF (UI2C_PROTSTS [8])， STORIF (UI2C_PROTSTS [9])标志后，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。在这时，需要设置FUNMODE (UI2C_CTL [2:0]) = 000B复位I2C模块才能离开这种状态。

协议功能描述

可编程设置及保持时间

为了保证正确的数据设置及保持时间，必须通过HTCTL [5:0] (UI2C_TMCTL[11:6])设置“保持时间”， STCTL [5:0] (UI2C_TMCTL[5:0]) 配置“设置时间”。

延迟时间参考外设时钟(PCLK)，当设备拉伸主机时钟时，配置的“设置”及“保持”时间不会被拉伸影响。

用户要注意“设置”及“保持”时间配置的限制，时序设置必须遵循I²C协议，如果“设置”时间配置超过设计限制，也就是“设置”时间配置使得SCL输出小于3个PCLK，I²C控制器将无法正常工作，因为SCL必须采样3次；如果“保持”时间配置超过I²C时钟限制将产生总线错误。建议客户根据所用的波特率及协议计算适合的时序，表 6.14-1 I2C 波特率与PCLK表示I²C 波特率与 PCLK之间的关系，表中的数字表示一个时钟周期包含多少个PCLK。“设置”及“保持”时间配置需要用户根据设计需要及I²C协议标准去设定

I ² C PCLK与波特率	100k	200k	400k
12MHz	120	60	30
24MHz	240	120	60
48MHz	480	240	120

表 6.14-1 I²C 波特率与PCLK之间的关系

作为“设置”时间出错的例子，我们假设一个 SCL 周期包含 10 个 PCLK，STCTL [5:0] (UI2C_TMCTL[5:0]) = 3，“设置”时间拉伸 3 个 PCLK；“设置”时间限制：ST_{limit} = (UI2C_BRGEN[25:16]+1) – 6，例如：如果用户设定 PCLK = 12MHz，波特率 =100k，则 UI2C_BRGEN[25:16] 必须等于 59，并且STCTL [5:0] 最大值 54。

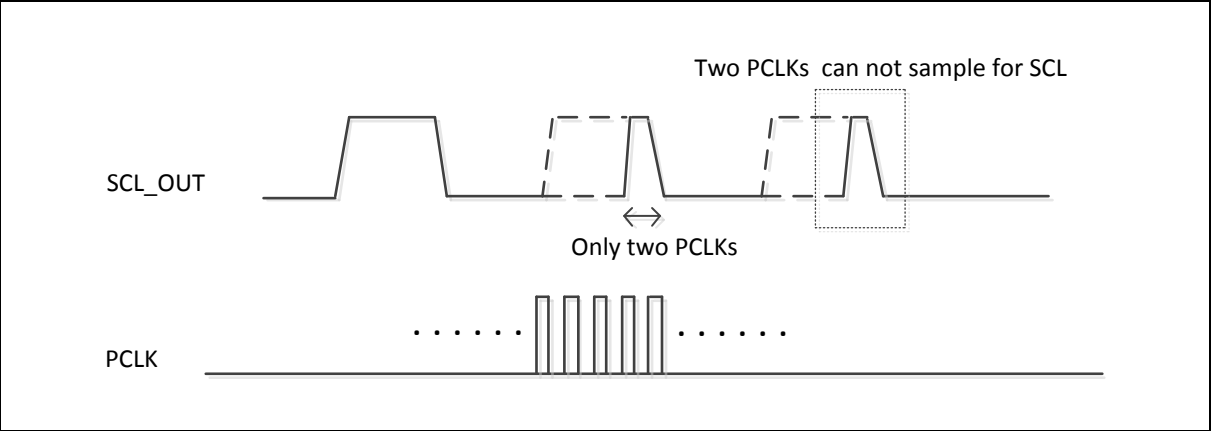


图 6.14-17 设置时间错误

作为“保持”时间出错的例子，我们设 I^2C 波特率为400K，PCLK=48MHz，SCL高/低占空比为60 PCLK，当我们设置HTCTL [5:0] (UI2C_TMCTL[11:6]) = 63，STCTL [5:0] (UI2C_TMCTL[5:0]) = 0，SDA输出延时将覆盖SCL高电平造成总线错误。“保持”时间设置限制： $HT_{limit} = (UI2C_BRGEN[25:16]+1) - 9$ ；例如，如果用户选择PCLK=12M，波特率=100k，UI2C_BRGEN[25:16]必须为59，HTCTL [5:0]最大值51。

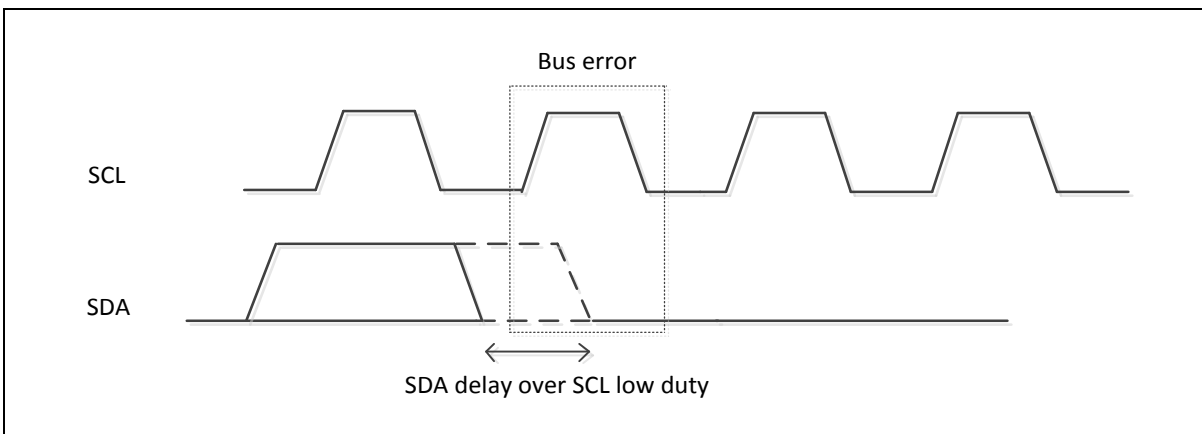


图 6.14-18 保持时间错误

I^2C 超时功能

超时计数器TOCNT (UI2C_PROTCTL [25:16])用来处理 I^2C 总线锁死事件，一旦计数器使能，会一直计数到TOCNT (UI2C_PROTCTL [25:16]) 设定的值并产生中断，用户可以设置TOIEN (UI2C_PROTIEN [0]) = 0或写1清中断标志(ACKIF, ERRIF, ARBLOIF, NACKIF, STORIF, STARIF) 停止超时计数器。用户写1清零TOIF(UI2C_PROTSTS[5])，当超时计数器使能，写1到TOIF，中断标志清除后计数器复位并且重新开始计数。参照图 6.14-19 为超时计数器TOCNT (UI2C_PROTCTL [25:16])。 $T_{TOCNT} = (TOCNT (UI2C_PROTCTL [25:16]) + 1) \times 32 (5\text{-位}) \times T_{PCLK}$ 。注意：计数器时钟源TMCNTSRC (USCI_BRGEN [5]) 需设为0。

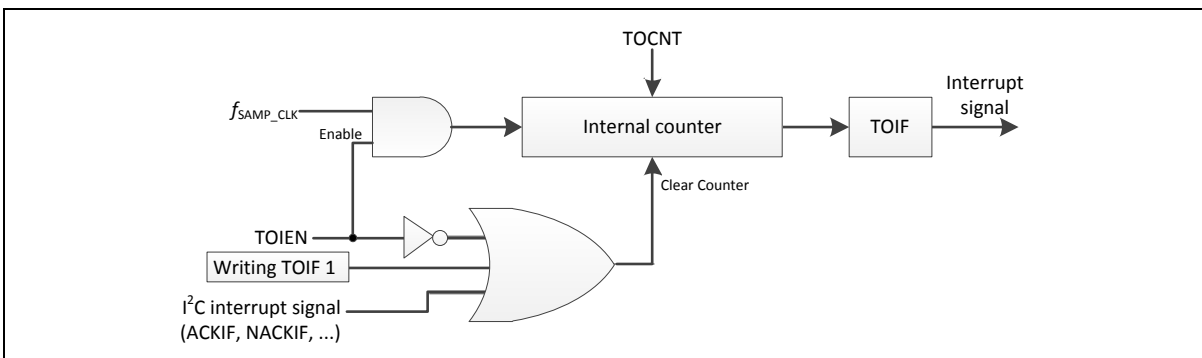


图 6.14-19 I^2C 超时计数框图

唤醒功能

设置WKEN (WKCTL[0]) = 1，芯片进入掉电模式，其他I²C主机可以通过地址唤醒芯片，进入掉电模式前用户要先设置好相应寄存器，地址匹配的ACK位周期都是在掉电模式下完成的。当地址匹配和应答周期完成，控制器将SCL低电平拉长直到用户清零WKAKDONE (UI2C_PROTSTS [16])。如果SCL频率是低速的并且系统已经从地址匹配唤醒，用户根据该位确认数据帧已经传输完成然后去处理唤醒过程。因此当芯片通过地址寄存器UI2C_DEVADDR0地址匹配唤醒，用户通过WKAKDONE (UI2C_PROTSTS [16])位来确定地址唤醒帧是否传输完成。WKAKDONE位表示在掉电模式下地址匹配帧应答位周期已经完成。注意用户必须清除WKAKDONE位后立即清除WKUPIF位。

WR_STATUS (UI2C_PROTSTS [17])位记录地址匹配唤醒后的读/写命令，用户根据此位的状态准备下一个要传送的数据(WR_STATUS = 0)或等待要接收的数据(WR_STATUS = 1)，以保证通过地址匹配唤醒后能够及时处理数据。

当系统被其他I²C主机唤醒，WKUPIF将被置1，用户需要写“1”清除该位。

I²C 也支持数据信号变化唤醒，设置WKEN (UI2C_WKCTL [0]) =1 及 WKADDREN (UI2C_WKCTL[1]) = 0，当数据信号有变化时MCU可以从低功耗下唤醒。

随机读EEPROM的范例

下列步骤用来配置USC10_I²C相关寄存器，通过I²C协议读EEPROM。

- 1. 通过SYS_GPA_MFP, SYS_GPC_MFP, SYS_GPD_MFP寄存器设置USC10_I²C多功能管脚作为SCL和SDA管脚。
- 2. 通过寄存器 “CLK_APBCLK[24]” USC10CKEN =1，使能USC10 APB 时钟。
- 3. 设置“SYS_IPRST1[24]”寄存器USC10RST=1去复位USC1控制器然后USC10RST=0 设置USC1控制器为正常操作。
- 4. 设置“UI2C_CTL” 寄存器中FUNMODE =100 使能USC10_I²C 控制器。
- 5. 通过寄存器“UI2C_BRGEN”设置USC10_I²C控制器的USC10_I²C时钟分频值
- 6. 通过设置寄存器 “NVIC_ISR” 中SETENA =0x00400000设置USC1_IRQ。
- 7. 在寄存器“UI2C_PROTIEN”中设置ACKIEN, ERRIEN, ARBLOIEN, NACKIEN, STORIEN, STARIEN, 和 TOIEN 使能中断。
- 8. 设置 USC1 地址寄存器 “USC1_DEVADDR0”。

随机读操作时一种存取EEPROM的方式，可以让主机读取EEPROM的任何地址。图 6.14-20 表示EEPROM随机读操作

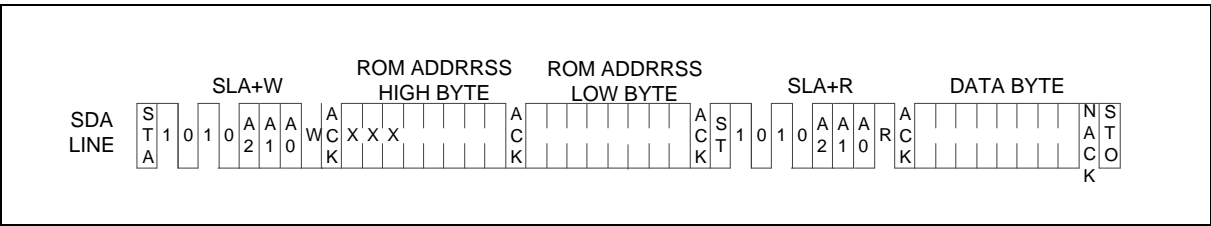


图 6.14-20 EEPROM 随机读

图 6.14-21 展示怎样用I²C 控制器实现EEPROM随机读操作.

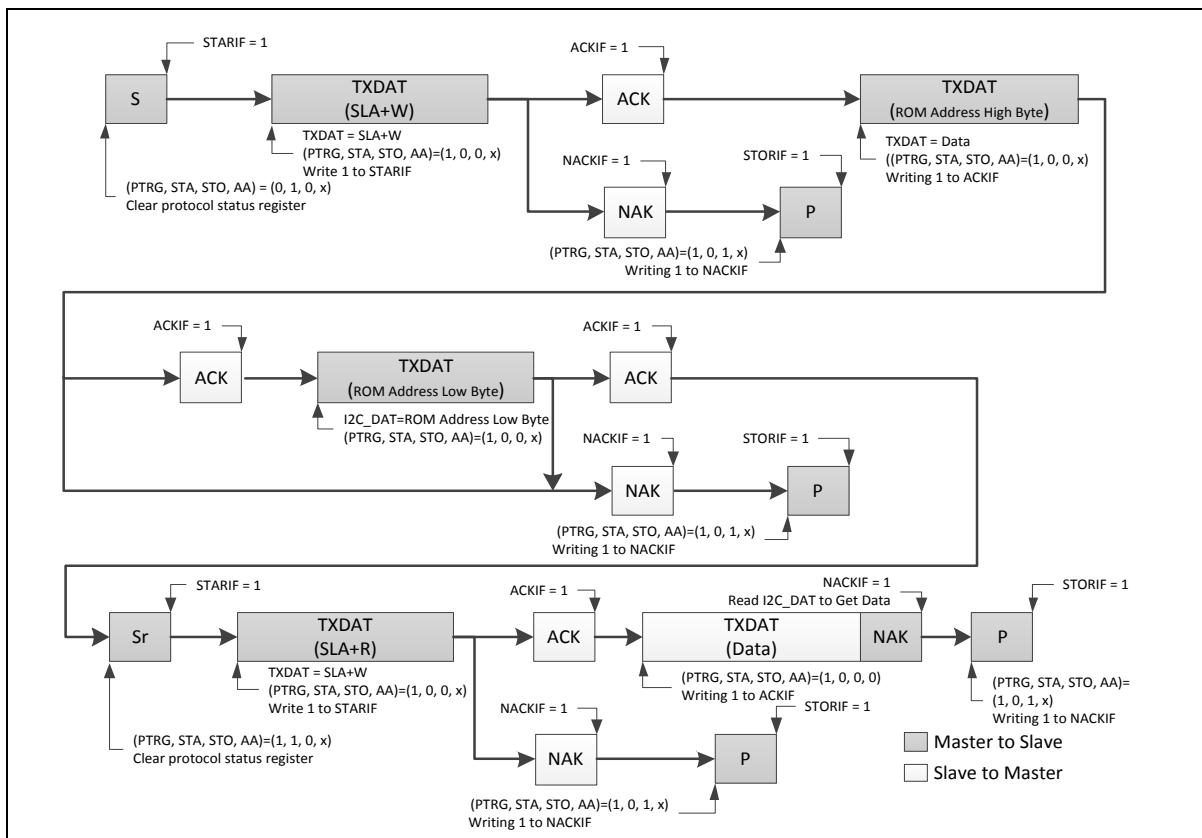


图 6.14-21 EEPROM 的协议随机读

I²C主控制器发送START信号，然后发送SLA+W (从机地址 + 写命令) 到 EEPROM接着是2字节地址，最后发送重新开始和SLA+R信号读EEPROM的数据。

6.14.6 寄存器映射

R: 只读, W: 只写, R/W: 读和写

寄存器	偏移地址	读 / 写	描述	复位值
UI2C 基地址: $UI2Cx_BA = 0x4007_0000 + (0x10_0000 * x)$ $x = 0, 1$				
UI2C_CTL	UI2Cx_BA+0x00	R/W	USCI 控制寄存器	0x0000_0000
UI2C_BRGEN	UI2Cx_BA+0x08	R/W	USCI 波特率产生寄存器	0x0000_3C00
UI2C_LINECTL	UI2Cx_BA+0x2C	R/W	USCI 线控寄存器	0x0000_0000
UI2C_TXDAT	UI2Cx_BA+0x30	W	USCI 发送数据寄存器	0x0000_0000
UI2C_RXDAT	UI2Cx_BA+0x34	R	USCI 接收数据寄存器	0x0000_0000
UI2C_DEVADDR0	UI2Cx_BA+0x44	R/W	USCI 设备地址寄存器 0	0x0000_0000
UI2C_ADDRMSK0	UI2Cx_BA+0x4C	R/W	USCI 设备地址掩码寄存器 0	0x0000_0000
UI2C_WKCTL	UI2Cx_BA+0x54	R/W	USCI 唤醒控制寄存器	0x0000_0000
UI2C_WKSTS	UI2Cx_BA+0x58	R/W	USCI 唤醒状态寄存器	0x0000_0000
UI2C_PROTCTL	UI2Cx_BA+0x5C	R/W	USCI 协议控制寄存器	0x0000_0000
UI2C_PROTIEN	UI2Cx_BA+0x60	R/W	USCI 协议中断使能寄存器	0x0000_0000
UI2C_PROTSTS	UI2Cx_BA+0x64	R/W	USCI 协议状态寄存器	0x0000_0000
UI2C_TMCTL	UI2Cx_BA+0x8C	R/W	I ² C 时序配置控制寄存器	0x0000_0000

6.14.7 寄存器描述

USCI 控制寄存器 (UI2C_CTL)

寄存器	偏移地址	读/写	描述	复位值
UI2C_CTL	UI2Cx_BA+0x00	R/W	USCI 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					FUNMODE		

位	描述	
[31:3]	Reserved	保留。
[2:0]	FUNMODE	<p>功能模式</p> <p>此位域选择USCI 控制器的协议，非法的设置会关闭USCI功能。两种协议切换时，需要先关闭USCI功能再切换新的协议，写0x0到FUNMODE复位USCI</p> <p>000 =关闭USCI功能，所有协议状态机进入空闲状态。</p> <p>001 =选择SPI协议。</p> <p>010 =选择UART协议。</p> <p>100 =选择 I²C 协议。</p> <p>注意： 其他位组合保留。</p>

USCI 波特率产生寄存器 (UI2C BRGEN)

寄存器	偏移地址	读/写	描述	复位值
UI2C_BRGEN	UI2Cx_BA+0x08	R/W	USCI 波特率产生寄存器	0x0000_3C00

31	30	29	28	27	26	25	24
Reserved						CLKDIV	
23	22	21	20	19	18	17	16
CLKDIV							
15	14	13	12	11	10	9	8
Reserved		DSCNT				PDSCNT	
7	6	5	4	3	2	1	0
Reserved		TMCNTSRC	TMCNTEN	SPCLKSEL		PTCLKSEL	RCLKSEL

位	描述	
[31:26]	Reserved	保留.
[25:16]	CLKDIV	<p>时钟除频器</p> <p>此位定义了$f_{\text{PROT_CLK}}$ 和 $f_{\text{DIV_CLK}}$ ($f_{\text{DIV_CLK}} = f_{\text{PROT_CLK}} / (\text{CLKDIV}+1)$) 的比例关系.</p> <p>注意: 在UART模式, 当自动波特率功能 (ABREN (UI2C_PROTCTL[6])) 使能时, 该区域可以被硬件通过输入数据0x55的第4个下降沿自动更新. 改进的值是第5和第6位的平均位时间. 用户可以通过CLKDIV改进值和新的BRDETIV (UI2C_PROTCTL[24:16]) 值计算出准确的波特率.</p>
[15]	Reserved	保留.
[14:10]	DSCNT	<p>采样计数器分母</p> <p>此位域定义采样时钟$f_{\text{SAMP_CLK}}$的除频率.</p> <p>除频率 $f_{\text{DS_CNT}} = f_{\text{PDS_CNT}} / (\text{DSCNT}+1)$.</p> <p>注意: UART模式下DSCNT 的最大值是 0xF, 建议设置大于4的值, 以保证采样数据的准确</p>
[9:8]	PDSCNT	<p>采样计数器预分频</p> <p>此位域定义了采样时钟$f_{\text{SAMP_CLK}}$的除频比例。 $f_{\text{PDS_CNT}} = f_{\text{SAMP_CLK}} / (\text{PDSCNT}+1)$.</p>
[7:6]	Reserved	保留.
[5]	TMCNTSRC	<p>时间测量计数器时钟源选择</p> <p>0 = 时间测量计数器时钟为 $f_{\text{PROT_CLK}}$.</p> <p>1 = 时间测量计数器时钟为$f_{\text{DIV_CLK}}$.</p>
[4]	TMCNTEN	<p>时间测量计数器使能位</p> <p>该位使能10bit时间测量计数器.</p> <p>0 = 时间测量计数器禁止.</p> <p>1 = 时间测量计数器使能.</p>
[3:2]	SPCLKSEL	采样时钟源选择

		<p>该位域用来选择采样时钟 (f_{SAMP_CLK}) 的时钟源.</p> <p>00 = $f_{SAMP_CLK} = f_{DIV_CLK}$.</p> <p>01 = $f_{SAMP_CLK} = f_{PROT_CLK}$.</p> <p>10 = $f_{SAMP_CLK} = f_{SCLK}$.</p> <p>11 = $f_{SAMP_CLK} = f_{REF_CLK}$.</p>
[1]	PTCLKSEL	<p>协议时钟源选择</p> <p>该位用于选择协议时钟 (f_{PROT_CLK}) 的时钟源.</p> <p>0 = 参考时钟源 f_{REF_CLK}.</p> <p>1 = f_{REF_CLK2} (频率为 参考时钟源 f_{REF_CLK} 一半).</p>
[0]	RCLKSEL	<p>参考时钟源选择</p> <p>该位用于选择参考时钟 (f_{REF_CLK}) 的时钟源.</p> <p>0 = 外设时钟 f_{PCLK}.</p> <p>1 = HXT/LXT.</p>

USCI 线控寄存器 (UI2C_LINECTL)

寄存器	偏移地址	读/写	描述	复位值
UI2C_LINECTL	UI2Cx_BA+0x2C	R/W	USCI 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				DWIDTH			
7	6	5	4	3	2	1	0
Reserved							LSB

位	描述	
[31:12]	Reserved	保留。
[11:8]	DWIDTH	<p>传输数据字长</p> <p>此位定义收发数据的字长（以位为单位），数据右对齐，USCI支持4到16位字长。</p> <p>0x0: 数据字长16位[15:0].</p> <p>0x1: 保留.</p> <p>0x2: 保留.</p> <p>0x3: 保留.</p> <p>0x4: 数据字长4位，位于[3:0].</p> <p>0x5: 数据字长5位，位于[4:0].</p> <p>...</p> <p>0xF: 数据字长15位，位于[14:0].</p> <p>注意: I²C协议, 协议下数据字长必须配置为8位.</p>
[7:1]	Reserved	保留。
[4:1]	Reserved	保留。
[0]	LSB	<p>LSB 优先发送选择</p> <p>0 = MSB优先发送, 根据设置的数据位宽度DWIDTH决定数据缓冲中优先发送 /接收位.</p> <p>1 = LSB优先发送, 数据缓冲的位0, 将被优先发送/接收.</p>

USCI 发送数据寄存器 (UI2C_TXDAT)

寄存器	偏移地址	读/写	描述	复位值
UI2C_TXDAT	UI2Cx_BA+0x30	W	USCI 发送数据寄存器	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
TXDAT							
7	6	5	4	3	2	1	0
TXDAT							

位	描述	
[31:8]	Reserved	保留.
[7:0]	TXDAT	发送数据 软件将要发送的8位数据写入此寄存器.

USCI 接收数据寄存器 (UI2C_RXDAT)

寄存器	偏移地址	读/写	描述	复位值
UI2C_RXDAT	UI2Cx_BA+0x34	R	USCI 接收数据寄存器	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
RXDAT							
7	6	5	4	3	2	1	0
RXDAT							

位	描述	
[31:8]	Reserved	保留
[7:0]	RXDAT	接收数据 此位域保存接收到的数据。 注意 1: I ² C 协议, 仅用 RXDAT[7:0].

USCI 设备地址寄存器 (UI2C DEVADDR)

寄存器	偏移地址	读/写	描述	复位值
UI2C_DEVADDR0	UI2Cx_BA+0x44	R/W	USCI 设备地址寄存器 0	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						DEVADDR	
7	6	5	4	3	2	1	0
DEVADDR							

位	描述	
[31:10]	Reserved	保留.
[9:0]	DEVADDR	<p>设备地址</p> <p>I²C 协议, , 此位域包含可编程设置的从机地址, 如果接收到的第一个地址字节是: b1111 0AAX, 那么AA位会跟 DEVADDR[9:8]比较是否地址匹配, 这里X 是读/写位, 然后第二个地址字节会跟 DEVADDR[7:0]比较是否匹配</p> <p>注意: 当 I²C 用 7-位地址模式, 仅仅用 DEVADDR[6:0]</p>

USCI 设备地址掩码寄存器 (UI2C_ADDRMSK) – 仅仅用于I²C

寄存器	偏移地址	读/写	描述	复位值
UI2C_ADDRMSK0	UI2Cx_BA+0x4C	R/W	USCI 设备地址掩码寄存器 0	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						ADDRMSK	
7	6	5	4	3	2	1	0
ADDRMSK							

位	描述	
[31:10]	Reserved	保留.
[9:0]	ADDRMSK	<p>USCI 设备地址掩码</p> <p>0 =屏蔽禁止（接收到的地址位必须与地址寄存器一致）</p> <p>1 =屏蔽使能（接收到的地址位不用关心）</p> <p>USCI支持两个地址屏蔽寄存器下的多地址识别。当地址屏蔽寄存器位设为1，意味着不用关心收到的地址位；如果地址屏蔽寄存器位设为0，意味着接收到的地址位必须与地址寄存器中的值完全一致才可能产生地址匹配</p>

USCI 唤醒控制寄存器 (UI2C_WKCTL)

寄存器	偏移地址	读/写	描述	复位值
UI2C_WKCTL	UI2Cx_BA+0x54	R/W	USCI 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						WKADDREN	WKEN

位	描述	
[31:2]	Reserved	保留.
[1]	WKADDREN	唤醒地址匹配使能 0 = 芯片被数据信号变化唤醒 1 = 芯片被地址匹配唤醒
[0]	WKEN	唤醒使能位 0 = 唤醒功能禁止. 1 = 唤醒功能使能.

USCI 唤醒状态寄存器 (UI2C WKSTS)

寄存器	偏移地址	读/写	描述	复位值
UI2C_WKSTS	UI2Cx_BA+0x58	R/W	USCI 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							WKF

位	描述	
[31:1]	Reserved	保留
[0]	WKF	唤醒标志 当芯片从掉电模式唤醒，此位置1，软件写1清零

USCI 协议控制寄存器 – I²C (UI2C_PROTCTL)

寄存器	偏移地址	读/写	描述	复位值
UI2C_PROTCTL	UI2Cx_BA+0x5C	读/写	USCI 协议控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PROTEN	Reserved					TOCNT	
23	22	21	20	19	18	17	16
TOCNT							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		PTRG	ADDR10EN	STA	STO	AA	GCFUNC

位	描述	
[31]	PROTEN	I ² C 协议使能位 0 = I ² C 协议禁止. 1 = I ² C 协议使能.
[30:26]	Reserved	保留.
[25:16]	TOCNT	超时时钟计数 此位域定义超时计数器计数多少个时钟周期会产生超时中断, 时钟周期由TMCNTSRC (UI2C_BRGEN [5])所选,TOCNT大于0超时功能使能. 注意: TMCNTSRC (UI2C_BRGEN [5]) 在I ² C 模式下必须设为0.
[15:6]	Reserved	保留.
[5]	PTRG	I ² C 协议触发 当一个新的状态出现在UI2C_PROTSTS寄存器中, 如果相应中断使能, 将会触发I ² C中断, 相应中断标志置位后软件必须写1到此寄存器, I ² C协议功能将会继续向前直到STOP信号有效或者PROTEN被禁止 0 = I ² C拉伸禁止, 协议功能将继续向前 1 = I ² C 拉伸使能
[4]	ADDR10EN	10位地址功能使能 0 = 10位地址功能禁止 1 = 10位地址功能使能
[3]	STA	I ² C 开始控制 STA设为1进入主机模式, 当总线空闲时I ² C硬件发送开始或重复开始信号

[2]	STO	I²C 停止控制 主机模式下设置STO=1发送停止信号，I ² C硬件会检测总线状态，如果检测到停止信号，此位将被硬件清零；从机模式下，当总线错误时(UI2C_PROTSTS.ERRIF = 1)，设置STO=1复位I ² C硬件为无地址从机模式
[1]	AA	应答发送控制 接收到地址或数据之前设置AA=1，下列情况下应答位（SDA低电平）将被自动发送： 1）从机应答主机发送的地址； 2）接收器应答发送器； 如果接收到地址或数据之前设置AA=0，NAK（SDA高电平）将被自动发送
[0]	GCFUNC	广播功能 0 = 广播模式禁止 1 = 广播模式使能

USCI 协议中断使能寄存器 – I²C (UI2C_PROTIEN)

寄存器	偏移地址	读/写	描述	复位值
UI2C_PROTIEN	UI2Cx_BA+0x60	R/W	USCI 协议中断使能寄存器	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							
7	6	5	4	3	2	1	0
Reserved	ACKIEN	ERRIEN	ARBLOIEN	NACKIEN	STORIEN	STARIEN	TOIEN

位	描述	
[31:7]	Reserved	保留.
[6]	ACKIEN	ACK中断使能 此位使能主机检测到应答时产生中断 0 = 应答中断禁止 1 = 应答中断使能
[5]	ERRIEN	错误中断使能控制 此位使能I ² C总线错误中断，总线错误由ERR (UI2C_PROTSTS [16])显示 0 = 总线错误中断禁止 1 = 总线错误中断使能
[4]	ARBLOIEN	仲裁失败中断使能 此位使能仲裁失败事件中断 0 = 禁止仲裁失败中断 1 = 使能仲裁失败中断
[3]	NACKIEN	无应答中断使能 此位使能主机检测到无应答中断 0 = 无应答中断禁止 1 = 无应答中断使能
[2]	STORIEN	停止信号中断使能 此位使能停止信号中断 0 = 停止信号中断禁止 1 = 停止信号中断使能
[1]	STARIEN	开始信号中断使能

		此位使能检测到开始信号产生中断 0 = 开始信号中断禁止 1 = 开始信号中断使能
[0]	TOIEN	超时中断使能 I ² C协议下，此位使能超时事件产生中断 0 = 超时中断禁止 1 = 超时中断使能

USCI 协议状态寄存器 – I²C (UI2C_PROTSTS)

寄存器	偏移地址	读/写	描述	复位值
UI2C_PROTSTS	UI2Cx_BA+0x64	R/W	USCI 协议状态寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					BUSHANG	WRSTSWK	WKAKDONE
15	14	13	12	11	10	9	8
SLAREAD	SLASEL	ACKIF	ERRIF	ARBLOIF	NACKIF	STORIF	STARIF
7	6	5	4	3	2	1	0
Reserved	ONBUSY	TOIF	Reserved				

位	描述	
[31:19]	Reserved	保留.
[18]	BUSHANG	<p>错误仲裁丢失</p> <p>该位表示总线仲裁丢失状态。当SCL保持高电平，参照f_{SAMP_CLK}，4位计数器计数，当SDA为低电平，仲裁计数器溢出并且置位该位。计数器将在SCL信号下降沿被复位。</p> <p>0 = 总线处于正常状态。</p> <p>1 = 总线错误仲裁丢失。</p> <p>注意：该位无中断信号，它会被硬件自动清零。</p>
[17]	WRSTSWK	<p>地址匹配唤醒帧的读/写状态位</p> <p>0 = 地址匹配唤醒帧写命令</p> <p>1 = 地址匹配唤醒帧读命令</p>
[16]	WKAKDONE	<p>唤醒帧地址应答位完成</p> <p>0 = 唤醒帧地址应答位未完成</p> <p>1 = 唤醒帧地址应答位在掉电模式下已经完成</p> <p>注意：当WKUPIF位被置1时该位不能被释放。</p>
[15]	SLAREAD	<p>从机读请求状态</p> <p>此位表示检测到了从机读请求</p> <p>0 = 没有检测到从机读请求</p> <p>1 = 检测到从机读请求</p> <p>注意：该位没有中断信号，硬件会自动清零</p>
[14]	SLASEL	<p>从机选择状态</p> <p>该位表示本设备被选为从机</p> <p>0 = 本设备没有被选为从机</p> <p>1 = 本设备被选为从机</p>

		注意： 该位没有中断信号，硬件会自动清零
[13]	ACKIF	应答接收中断标志 该位表示主机模式下收到了应答信号，在从机模式该位无效。如果UI2C_PROTCTL.ACKIEN = 1将会产生中断 0 = 没有收到应答 1 = 接收到应答 通过软件写1清零该位
[12]	ERRIF	错误中断标志 此位表示总线错误，也就是说在错误的位置检测到STOP,START信号，错误位置出现在发传输地址字节、数据字节或者应答位。如果UI2C_PROTCTL.ERRIEN = 1，将会产生中断 0 = 没有检测到I ² C总线错误 1 = 检测到I ² C总线错误 通过软件写1清零 注意: 从机模式下该位置位，用户必须写1到STO寄存器，使从机进入未寻址从机模式
[11]	ARBLOIF	仲裁失败标志 此位表示仲裁失败，如果UI2C_PROTCTL.ARBLOIEN = 1将会产生中断 0 = 没有仲裁失败 1 = 仲裁失败 通过软件写1清零
[10]	NACKIF	NAK中断标志 此位表示主机模式下收到NAK信号，如果UI2C_PROTCTL.NACKIEN = 1将会产生中断 0 = 没有收到NAK信号 1 = 收到NAK信号 通过软件写1清零
[9]	STORIF	STOP中断标志 此位表示I ² C检测到STOP信号，如果UI2C_PROTCTL.STORIEN = 1将会产生中断 0 = 没有检测到STOP信号 1 = 检测到STOP信号 通过软件写1清零
[8]	STARIF	START中断标志 此位表示主机接收到开始或重新开始信号，从机模式下表示接收到重新开始信号。 如果UI2C_PROTCTL.STARIEN = 1将会进入中断 0 = 没有检测到开始信号 1 = 检测到开始信号 通过软件写1清零
[7]	Reserved	保留。
[6]	ONBUSY	总线繁忙标志 此位表示总线正在通信，硬件在检测到开始信号后此位置1，检测到停止信号后清零 0 = 总线空闲 (SCLK 和SDA 都为高). 1 = 总线繁忙
[5]	TOIF	超时中断标志

		0 = 没有发生超时中断 1 = 发生超时中断 注意: 通过软件写清除该位
[4:3]	Reserved	保留.
[2:0]	Reserved	保留.
[0]	Reserved	保留.

USCI 时序配置控制寄存器 (UI2C_TMCTL)

寄存器	偏移地址	读/写	描述	复位值
UI2C_TMCTL	UI2Cx_BA+0x8C	R/W	I ² C 时序配置控制寄存器	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				HTCTL			
7	6	5	4	3	2	1	0
HTCTL		STCTL					

位	描述	
[31:8]	Reserved	保留.
[11:6]	HTCTL	保持时间配置寄存器 该位域配置发送模式下SCL下降沿与SDA电平变化之间的延时 保持时间延时以外设时钟个数HTCTL x f _{PLK} 计算
[5:0]	STCTL	设置时间配置寄存器 该位域配置发送模式下SDA电平变化与SCL上升沿之间的延时 设置时间延时以外设时钟个数STCTL x f _{PLK} 计算

6.15 硬件除法器 (HDIV)

6.15.1 概述

硬件除法器(HDIV) 在高性能应用中很有用，该硬件除法器是一个有符号的整数除法器，其输出商和余数。

6.15.2 特征

- 有符号（二进制补码）整数计算
- 32位被除数、16位除数的计算能力
- 输出32位商和32位余数(16位余数随着符号扩展到32位)
- 除以0警告标志
- 一个计算周期消耗6HCLK 时钟
- 写除数触发计算
- 当读商和余数时，将自动准备好等待下次计算

6.15.3 基本配置

在使用硬件除法器之前，硬件除法器的时钟必须使能。为了使能硬件除法器，HDIVCKEN (CLK_AHBCLK[4]) 要设为1.

6.15.4 功能描述

使用硬件除法器，首先要设置被除数，然后设置除数。除数设置了之后，硬件除法器将自动触发计算。包含商和余数的计算结果可以从QUOTIENT (HDIV_QUOTIENT[31:0])和REM (HDIV_REM[31:0])寄存器中读到。如果CPU在硬件除法器计算完成之前读QUOTIENT或者REM，CPU将保持直到硬件除法器完成计算。因此，CPU在触发一次硬件除法器计算后，没有软件延时，也可以始终获取有效的结果。

如果除数为 0，DIVBYZERO (HDIV_STATUS[1])将被置1。

被除数是32位有符号整数，除数是16位有符号整数。商是32位有符号整数，余数是16位有符号整数，要注意的是如果被除数是-1，由于溢出，商会设为最小的负值，余数为0，这是唯一的两个负数相除商不是正数的特例

硬件除法操作流程如图 6.15-1 所示。 计算 X / Y ，CPU 需要写 X 到 DIVIDEND (HDIV_DIVIDEND[31:0]) 寄存器，然后写 Y 到DIVISOR (HDIV_DIVISOR[15:0])寄存器。 DIVISOR 被写后，CPU 可以读QUOTIENT 和 REM 寄存器获取计算结果。

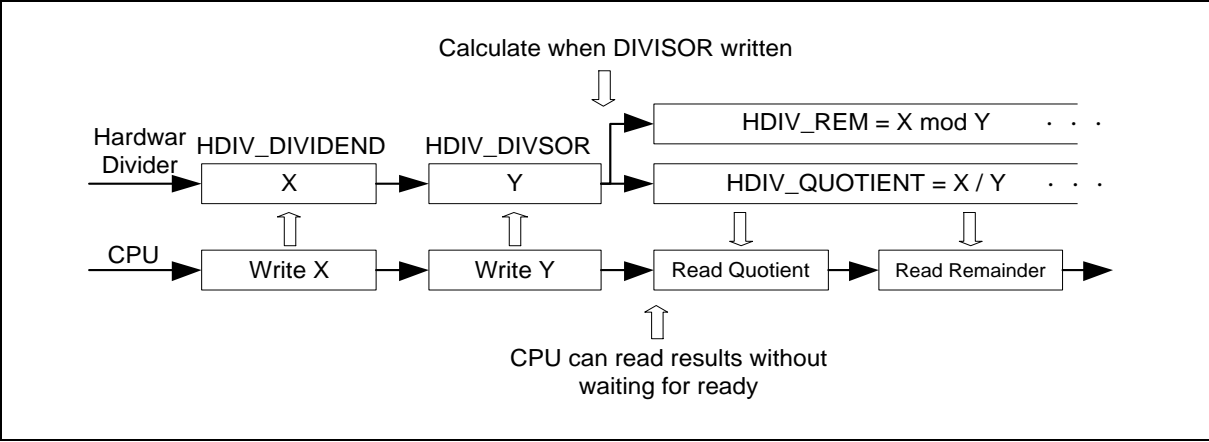


图 6.15-1 硬件除法器操作流程

6.15.5 寄存器映射

R: 只读, W: 只写, R/W: 读和写

寄存器	偏移地址	读 / 写	描述	复位值
HDIV 基地址: HDIV_BA = 0x5001_4000				
HDIV_DIVIDEND	HDIV_BA+0x00	R/W	被除数源寄存器	0x0000_0000
HDIV_DIVISOR	HDIV_BA+0x04	R/W	除数源寄存器	0x0000_FFFF
HDIV_QUOTIENT	HDIV_BA+0x08	R/W	商结果寄存器	0x0000_0000
HDIV_REM	HDIV_BA+0x0C	R/W	余数结果寄存器	0x0000_0000
HDIV_STATUS	HDIV_BA+0x10	R	除法器状态寄存器	0x0000_0001

6.15.6 寄存器描述

被除数源寄存器 (HDIV_DIVIDEND)

寄存器	偏移地址	读/写	描述	复位值
HDIV_DIVIDEND	HDIV_BA+0x00	R/W	被除数源寄存器	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	被除数源 该寄存器要在除法器计算开始之前写入值

除数源寄存器 (HDIV_DIVISOR)

寄存器	偏移地址	读/写	描述	复位值
HDIV_DIVISOR	HDIV_BA+0x04	R/W	除数源寄存器	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	<p>除数源</p> <p>该寄存器要在除法器计算开始之前写入值。</p> <p>注意: 当寄存器被写入之后, 硬件除法器将开始计算。</p>

商结果寄存器 (HDIV_QUOTIENT)

寄存器	偏移地址	读/写	描述	复位值
HDIV_QUOTIENT	HDIV_BA+0x08	R/W	商结果寄存器	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	商结果 该寄存器在除法器计算完成之后保存商的值。

余数结果寄存器 (HDIV_REM)

寄存器	偏移地址	读/写	描述	复位值
HDIV_REM	HDIV_BA+0x0C	R/W	余数结果寄存器	0x0000_0000

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

位	描述	
[31:0]	REM	<p>余数结果</p> <p>硬件除法器的余数是16位有符号整数(REM[15:0])，符号扩充(REM[31:16])到32位整数。</p>

除法器状态寄存器 (HDIV_STATUS)

寄存器	偏移地址	读/写	描述	复位值
HDIV_STATUS	HDIV_BA+0x10	R	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						DIVBYZERO	Reserved

位	描述	
[31:2]	Reserved	保留.
[1]	DIVBYZERO	<p>除数0警告 (只读)</p> <p>0 = 除数不是0. 1 = 除数是0.</p> <p>注意: DIVBYZERO 标志用于表示除以0的情况, 并且当HDIV_DIVISOR被写入后就更新. 该位只读.</p>
[0]	Reserved	保留.

6.16 模数转换器 (ADC)

1.1.1 概述

NM1120系列包含一个12位8通道逐次逼近型模数转换器(SAR A/D 转换器). A/D转换可通过软件、外部引脚(STADC/PC.1)或PWM触发.

1.1.2 特性

- 模拟输入电压范围: $0 \sim V_{DD}$.
- 12位分辨率, 保证10位精度.
- 多达8个模拟输入通道.
- ADC 时钟频率高达16MHz.
- 可配置ADC内部采样时间.

1.1.3 方框图

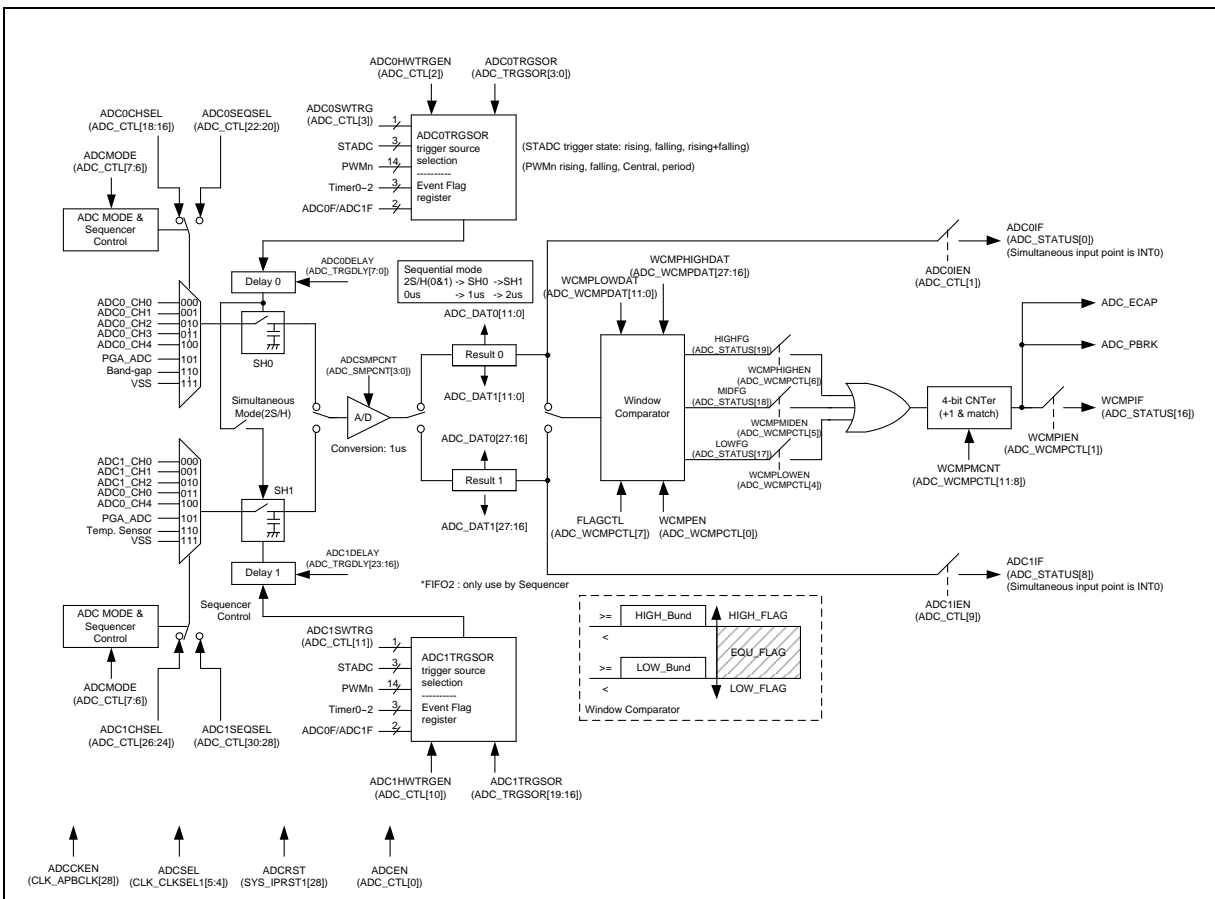


图 6.16-1 ADC 控制方框图

6.16.1 基本配置

NM1120系列有两路采样和保持(S/H)电路用于同时采样两路ADC输入通道.针对BLDC马达用途，支持四种操作模式.

ADC引脚功能通过SYS_PB_MFP, SYS_PC_MFP 和 SYS_PD_MFP寄存器配置. 针对模拟输入引脚，建议禁用数字输入通道，防止漏电.用户可以通过配置 PB_DINOFF, PC_DINOFF 和 PD_DINOFF 寄存器禁用数字输入通道.

ADC外设时钟可以通过ADCCKEN (CLK_APBCLK[28])使能. ADC外设时钟源通过ADCSEL (CLK_CLKSEL1[5:4])选择. 时钟分频由ADCDIV (CLK_CLKDIV[23:16])决定.

6.16.2 功能描述

该A/D转换器采用逐次逼近转换方式，支持12位分辨率.当要改变模拟输入通道时，为了防止错误操作，软件必须清除ADC_CTL寄存器的ADCnSWTRG位.当ADCnSWTRG位清除时，A/D转换器将立即丢弃当前转换结果，并进入空闲状态.

6.16.2.1 ADC外设时钟发生器

ADC有四种时钟源，可通过ADCSEL (CLK_CLKSEL1)选择. 选择HXT或LXT必须通过CLK_PWRCTL使能相应时钟. ADC外设时钟频率通过一个8位的预分频器，按下述公式进行分频：

$$\text{ADC 外设时钟频率} = (\text{ADC 外设时钟源频率}) / (\text{ADCDIV} + 1);$$

ADCDIV 位于寄存器CLK_CLKDIV.

通常，软件可以设置ADCSEL 和 ADCDIV 取得16MHz或少于16MHz的ADC外设时钟频率.

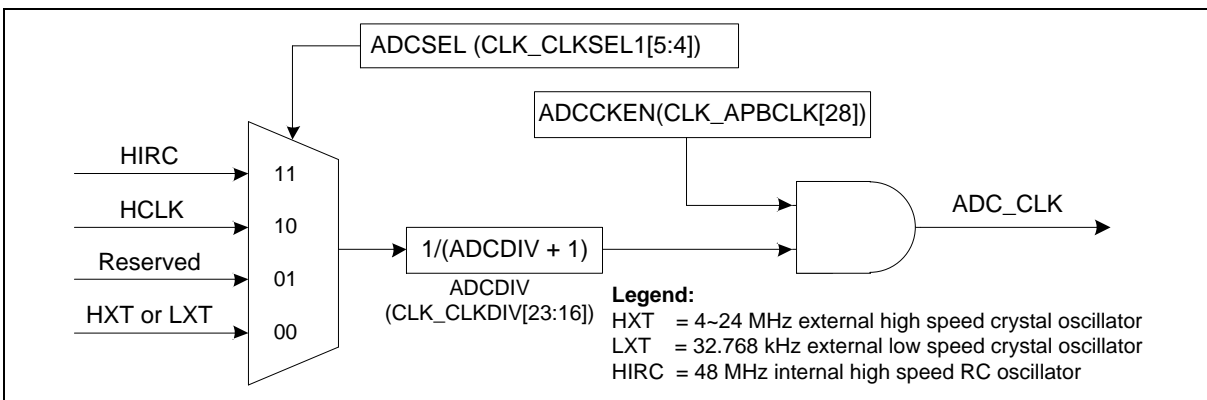


图 6.16-2 ADC 外设时钟控制

6.16.2.2 ADC操作

A/D转换器仅对指定的一个通道进行一次转换，操作步骤如下：

1. 当ADC_CTL的ADCnSWTRG位通过软件或硬件触发输入设置为1，A/D转换开始。
2. 当A/D转换结束，转换结果储存在A/D数据寄存器中。
3. A/D转换完成后，ADC_STATUS寄存器的ADCnIF位将被置1，如果ADC_CTL寄存器的ADCnIEN位置1，将产生ADC中断。
4. 在A/D转换过程中，ADCnSWTRG位保持为1。当转换结束，ADCnSWTRG将自动清0，A/D转换器进入空闲状态。

单次模式采样时序框图如图 6.16-3.

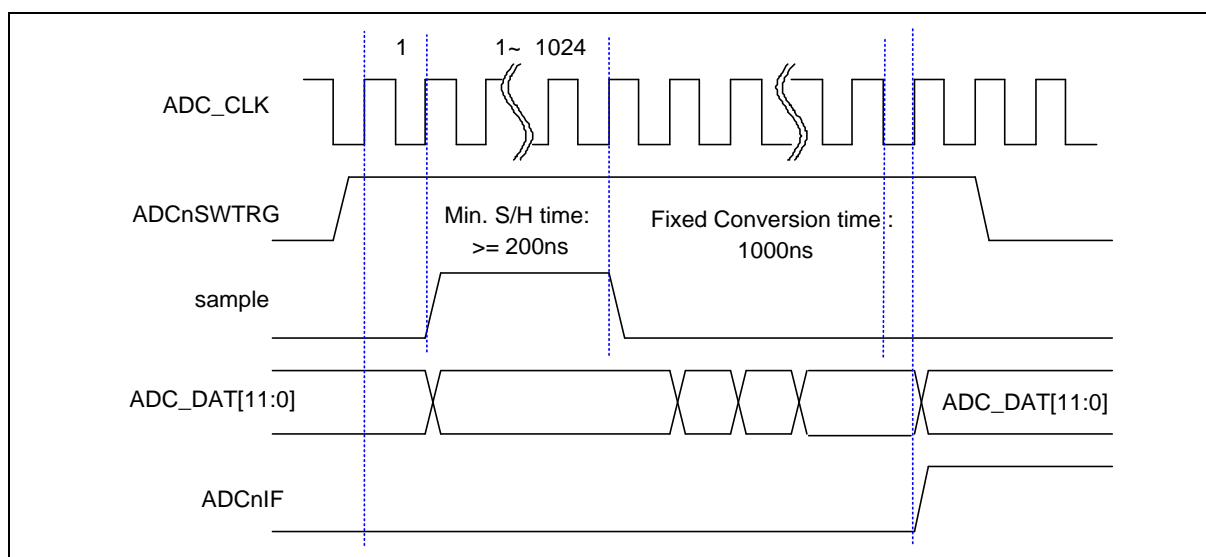


图 6.16-3 单周期模式转换时序框图

6.16.2.3 硬件触发输入采样和A/D转换时间

A/D转换可以通过硬件触发，当ADC_CTL寄存器的ADCnHWTRGEN位置1使能ADC硬件触发功能，设置ADCnTRGSOR位为0000b选择来自STADC引脚的外部触发输入，软件可以设置ADCnSTADCSEL选择下降沿还是上升沿触发。如果选择边沿触发功能，高或低状态必须至少保持4个PLCK，低于该值的脉冲将被忽略。

6.16.2.4 PWM触发

A/D转换也可以通过PWM触发，当ADCnHWTRGEN设置为高使能ADC硬件触发功能，设置ADCnTRGSOR可以选择硬件触发输入源来自PWM触发。使能PWM触发时，设置ADC_TRGDLY可以在PWM触发和ADC开始转换之间插入一段延时。

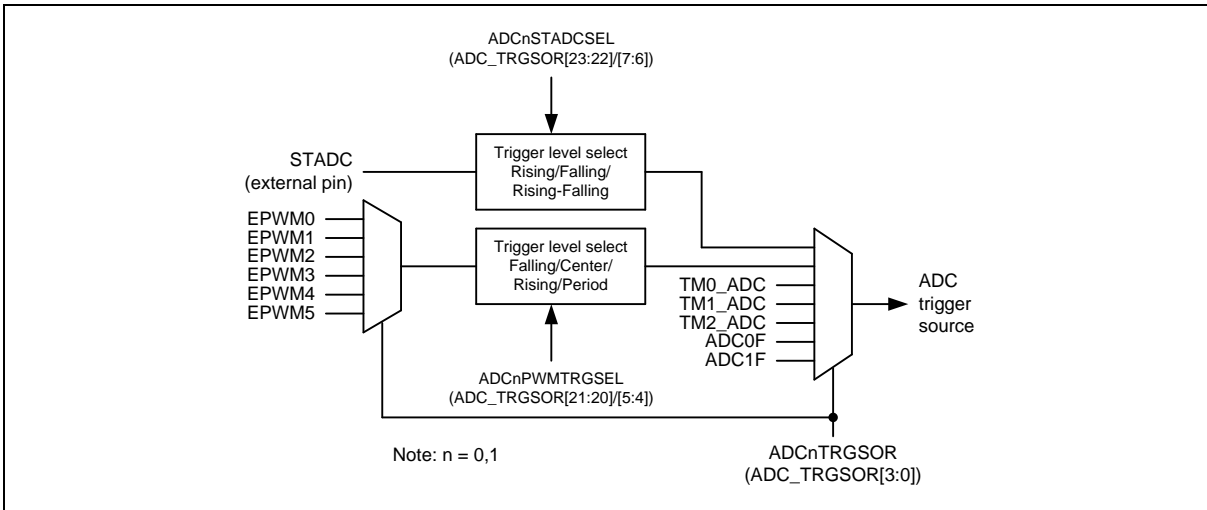


图 6.16-4 ADC 硬件触发源

6.16.2.5 通过窗口比较模式功能侦测转换结果

NM1120系列提供一个窗口比较器功能，软件可以写ADC_WCMPDAT寄存器设置低或高范围来检测三段ADC值。

当转换结果与设置的匹配时，比较匹配计数器将加1，否则比较匹配计数将清0。当匹配计数器到达设置的值（WCMPCNT+1），WCMPIF将置1，如果WCMPIEN和WCPMEN位置1，将产生ADCINT中断。软件可以使用该功能检测外部模拟输入引脚电压转变。

6.16.2.6 中断源

ADC中断的中断源有三个，分别是：ADC0中断、ADC1中断、ADC窗口比较器中断。

6.16.2.7 独立采样模式

NM1120有两路采样保持电路(S/H)和一个A/D转换模块。当ADCMODE设置为00b (ADC_CTL[7:6])，可以设置为独立采样、转换和独立产生中断。其可以看作使用两路ADC功能。

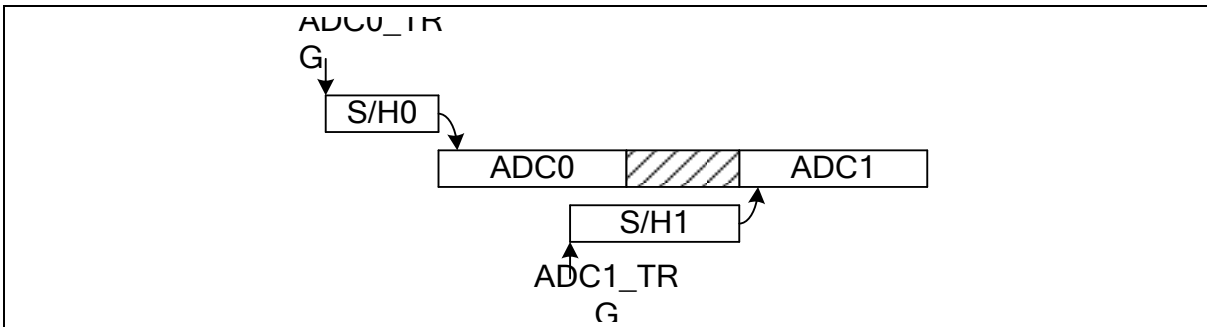


图 6.16-5 独立采样模式转换时序图

6.16.2.8 独立2SH模式

当ADCMODE设置为01b (ADC_CTL[7:6])，NM1120同样可以被设置为独立采样模式，但是它需要

连续转换两次(S/H0和S/H1)，且只产生ADC0IF中断。

例如，ADC0触发源设置为PWM0，ADC1触发源设置为PWM2，如果PWM0首先触发ADC0，当ADC0转换结束，不会产生中断，需要等ADC1转换，当ADC1转换完成，才会进入产生ADC0中断。也就是连续转换ADC0和ADC1，然后才产生中断。

6.16.2.9 同时采样模式

当ADCMODE设置为10b (ADC_CTL[7:6])，NM1120可以设置通过一次ADC0触发源，执行两次S/H转换，产生ADC0IF中断。

图 6.16-6 展示一次S/H0和S/H1同时采样和保持，然后通过A/D转换器连续转换。

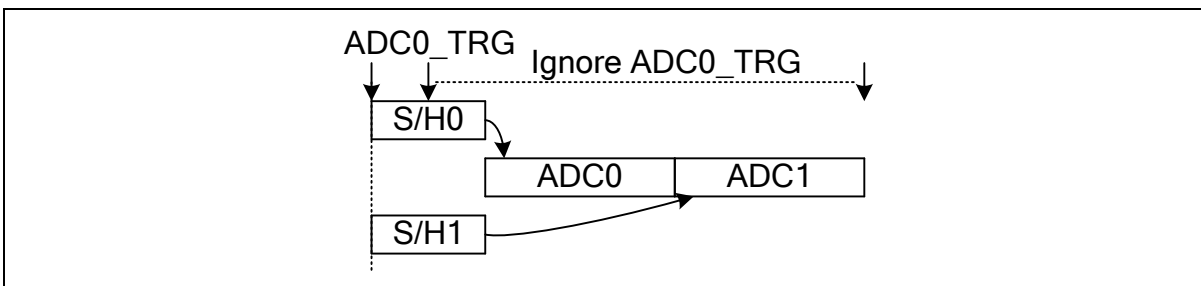


图 6.16-6 同时采样模式转换采样时序图

6.16.2.10 同时连续4R模式

当ADCMODE设置为11b (ADC_CTL[7:6])，NM1120可以设置通过ADC0触发源，执行两次S/H转换，然后通过ADC连续转换四次，产生ADC0IF中断。

图 6.16-7 展示S/H0 和 S/H1同时采样保持，然后顺序转换ADC0和ADC1四次。当ADC0转换结束，ADC0 S/H将被再次采样；当AD1转换结束，ADC1 S/H将被再次采样。

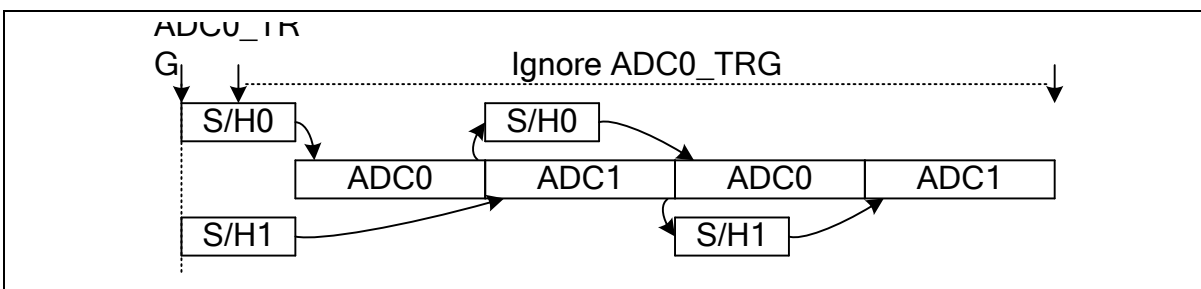


图 6.16-7 同时连续 4R 模式转换时序图

6.16.3 寄存器映射

R: 只读, W: 只写, R/W: 读/写

寄存器	偏移	R/W	描述	复位值
ADC基地址: ADC_BA = 0x400E_0000				
ADC_DAT0	ADC_BA+0x00	R	ADC数据寄存器 0	0x0000_0000
ADC_DAT1	ADC_BA+0x04	R	ADC数据寄存器 1	0x0000_0000
ADC_CTL	ADC_BA+0x20	R/W	ADC控制寄存器	0x0000_0000
ADC_TRGSOR	ADC_BA+0x24	R/W	ADC硬件触发源控制寄存器	0x0000_0000
ADC_TRGDLY	ADC_BA+0x28	R/W	ADC触发延时控制寄存器	0x0000_0000
ADC_SMPCNT	ADC_BA+0x2C	R/W	ADC采样时间计数器寄存器	0x0000_0005
ADC_STATUS	ADC_BA+0x30	R/W	ADC状态寄存器	0x0000_0000
ADC_WCMPCTL	ADC_BA+0x34	R/W	ADC窗口比较器控制寄存器	0x0000_0000
ADC_WCMPDAT	ADC_BA+0x38	R/W	ADC窗口比较器数据寄存器	0x0000_0000

6.16.4 寄存器描述

ADC数据寄存器 0 (ADC_DAT0)

寄存器	偏移	R/W	描述	复位值
ADC_DAT0	ADC_BA+0x00	R	ADC数据寄存器 0	0x0000_0000

31	30	29	28	27	26	25	24
ADC1VALID	ADC1OV	Reserved		ADC1DAT0			
23	22	21	20	19	18	17	16
ADC1DAT0							
15	14	13	12	11	10	9	8
ADC0VALID	ADC0OV	Reserved		ADC0DAT0			
7	6	5	4	3	2	1	0
ADC0DAT0							

位	描述	
[31]	ADC1VALID	ADC1有效标志 0 = 在ADC1DAT0[27:16]位的数据无效。 1 = 在ADC1DAT0[27:16]位的数据有效。 注: 当A/D转换完成之后, 该位自动置1。在读ADC_DAT0寄存器后, 该位由硬件自动清零。
[30]	ADC1OV	ADC1覆盖标志 0 = ADC1DAT0[27:16]的数据为当前转换结果。 1 = ADC1DAT0[27:16]的数据被覆盖。 注1: 如果ADC1DAT0[27:16]的转换数据还没被读走, 新的转换结果加载到该寄存器, OV将置1。 注2: 在ADC_DAT0寄存器读出来之后, 该位由硬件自动清零。
[29:28]	Reserved	保留
[27:16]	ADC1DAT0	ADC1转换结果 该域包含ADC转换结果。
[15]	ADC0VALID	ADC0有效标志 0 = 在ADC0DAT0[11:0]位的数据无效。 1 = 在ADC0DAT0[11:0]位的数据有效。 注: 当A/D转换完成之后, 该位自动置1。在读ADC_DAT0寄存器后, 该位由硬件自动清零。

位	描述	
[14]	ADC0OV	<p>ADC0覆盖标志</p> <p>0 = ADC0DAT0[11:0] 的数据为当前转换结果.</p> <p>1 = ADC0DAT0[11:0] 的数据被覆盖.</p> <p>注1: 如果ADC0DAT0[11:0]的转换数据还没被读走, 新的转换结果加载到该寄存器, OV将置1.</p> <p>注2: 在ADC_DAT0寄存器读出来之后, 该位由硬件自动清零.</p>
[13:12]	Reserved	保留
[11:0]	ADC0DAT0	<p>ADC0转换结果</p> <p>该域包含ADC转换结果.</p>

ADC数据寄存器 1 (ADC_DAT1)

寄存器	偏移	R/W	描述	复位值
ADC_DAT1	ADC_BA+0x04	R	ADC数据寄存器 1	0x0000_0000

31	30	29	28	27	26	25	24
ADC1VALID	ADC1OV	Reserved		ADC1DAT1			
23	22	21	20	19	18	17	16
ADC1DAT1							
15	14	13	12	11	10	9	8
ADC0VALID	ADC0OV	Reserved		ADC0DAT1			
7	6	5	4	3	2	1	0
ADC0DAT1							

位	描述	
[31]	ADC1VALID	ADC1有效标志 0 = 在ADC1DAT1[27:16]位的数据无效。 1 = 在ADC1DAT1[27:16]位的数据有效。 注: 当A/D转换完成之后, 该位自动置1。在读ADC_DAT1寄存器后, 该位由硬件自动清零。
[30]	ADC1OV	ADC1超载标志 0 = ADC1DAT1[27:16]的数据为当前转换结果。 1 = ADC1DAT1[27:16]的数据被覆盖。 注1: 如果ADC1DAT1[27:16]的转换数据还没被读走, 新的转换结果加载到该寄存器, OV将置1。 注2: 在ADC_DAT1寄存器读出来之后, 该位由硬件自动清零。
[29:28]	Reserved	保留
[27:16]	ADC1DAT1	ADC1转换结果 该域包含ADC转换结果。
[15]	ADC0VALID	ADC0有效标志 0 = 在ADC0DAT1[11:0]位的数据无效。 1 = 在ADC0DAT1[11:0]位的数据有效。 注: 当A/D转换完成之后, 该位自动置1。在读ADC_DAT1寄存器后, 该位由硬件自动清零。
[14]	ADC0OV	ADC0超载标志 0 = ADC0DAT1[11:0]的数据为当前转换结果。 1 = ADC0DAT1[11:0]的数据被覆盖。 注1: 如果ADC0DAT1[11:0]的转换数据还没被读走, 新的转换结果加载到该寄存器, OV将置1。 注2: 在ADC_DAT1寄存器读出来之后, 该位由硬件自动清零。

位	描述	
[13:12]	Reserved	保留
[11:0]	ADC0DAT1	ADC0转换结果 该域包含ADC转换结果.

ADC控制寄存器 (ADC_CTL)

寄存器	偏移	R/W	描述	复位值
ADC_CTL	ADC_BA+0x20	R/W	ADC控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	ADC1SEQSEL			Reserved	ADC1CHSEL		
23	22	21	20	19	18	17	16
Reserved	ADC0SEQSEL			Reserved	ADC0CHSEL		
15	14	13	12	11	10	9	8
Reserved				ADC1SWTRG	ADC1HWTRG EN	ADC1IEN	Reserved
7	6	5	4	3	2	1	0
ADCMODE		ADCSS3R	Reserved	ADC0SWTRG	ADC0HWTRG EN	ADC0IEN	ADCEN

位	描述	
[31]	Reserved	保留
[30:28]	ADC1SEQSEL	ADC1输入引脚选择 (第二个输入) 000 = ADC1_CH0. 001 = ADC1_CH1. 010 = ADC1_CH2. 011 = ADC0_CH0. 100 = ADC0_CH4. 101 = PGA_ADC. 110 = 温度传感器. 111 = VSS.
[27]	Reserved	保留
[26:24]	ADC1CHSEL	ADC1 通道选择 000 = ADC1_CH0. 001 = ADC1_CH1. 010 = ADC1_CH2. 011 = ADC0_CH0. 100 = ADC0_CH4. 101 = PGA_ADC. 110 = 温度传感器. 111 = VSS.
[23]	Reserved	保留

位	描述	
[22:20]	ADC0SEQSEL	ADC0输入引脚选择 000 = ADC0_CH0. 001 = ADC0_CH1. 010 = ADC0_CH2. 011 = ADC0_CH3. 100 = ADC0_CH4. 101 = PGA_ADC. 110 = BAND_GAP. 111 = VSS.
[19]	Reserved	保留
[18:16]	ADC0CHSEL	ADC0 通道选择 000 = ADC0_CH0. 001 = ADC0_CH1. 010 = ADC0_CH2. 011 = ADC0_CH3. 100 = ADC0_CH4. 101 = PGA_ADC. 110 = BAND_GAP. 111 = VSS.
[15:12]	Reserved	保留
[11]	ADC1SWTRG	ADC1 开始转换 0 = 停止转换，A/D转换器进入空闲模式。 1 = 开始转换。 注: ADC1SWTRG可以通过软件或外部STADC引脚置1，该位会被硬件自动清零。
[10]	ADC1HWTRGEN	硬件触发ADC转换使能位 使能或禁用通过硬件（PWM, Timer, ADC自身）触发A/D转换 0= 硬件触发ADC转换禁用。 1= 硬件触发ADC转换使能
[9]	ADC1IEN	ADC1中断使能位 0 = ADC1 中断功能禁用。 1 = ADC1 中断功能使能 注: 如果ADC1IEN位置1，A/D转换结束，将产生中断请求
[7:6]	ADCMODE	A/D 转换模式 00 = 独立采样; 各自有独立功能和独立中断。 01 = 独立2SH; 独立触发功能，ADC0和ADC1都转换完成，然后只产生ADC0IF 中断。 10 = 同时采样; 通过ADC0同时触发，ADC0、ADC1都转换完成后，产生ADC0IF 中断。 11 = 同时连续采样; 通过ADC0同时触发，该模式转换顺序为ADC0 -> ADC1 ->ADC0 -> ADC1 4次,然后产生 ADC0IF中断。

位	描述	
[5]	ADCSS3R	<p>ADC在ADCMODE = 11时，同时顺序采样3数据。</p> <p>0 = 转换顺序为 ADC0 -> ADC1 -> ADC0 -> ADC1, 在ADCMODE=11时，同时顺序采样四个数据.</p> <p>1 = 转换顺序为 ADC0 -> ADC1 -> ADC0, 在ADCMODE=11时，同时顺序采样三个数据.</p>
[4]	Reserved	保留
[3]	ADC0SWTRG	<p>ADC0转换开始</p> <p>0 = 停止转换，A/D转换器进入空闲模式.</p> <p>1 = 开始转换t.</p> <p>注: ADC0SWTRG可以通过软件或外部STADC引脚置1，该位会被硬件自动清零.</p>
[2]	ADC0HWTRGEN	<p>硬件触发ADC转换使能</p> <p>使能或禁用通过硬件（PWM, Timer, ADC自身）触发A/D转换</p> <p>0= 禁用.</p> <p>1= 使能.</p>
[1]	ADC0IEN	<p>ADC0中断使能</p> <p>0 = ADC0中断功能禁用.</p> <p>1 = ADC0中断功能使能.</p> <p>注: 如果ADC0IEN位置1，A/D转换结束，将产生中断请求</p>
[0]	ADCEN	<p>ADC转换器使能</p> <p>0 = ADC转换器禁用</p> <p>1 = ADC转换器使能</p> <p>注: 开始A/D转换之前，该位必须置1。清零该位，禁用A/D转换器模拟电路，减少功耗</p>

ADC硬件触发源控制寄存器 (ADC_TRGSOR)

寄存器	偏移	R/W	描述	复位值
ADC_TRGSOR	ADC_BA+0x24	R/W	ADC硬件触发源控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
ADC1STADCSEL		ADC1PWMTRGSEL		ADC1TRGSOR			
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
ADC0STADCSEL		ADC0PWMTRGSEL		ADC0TRGSOR			

位	描述	
[31:24]	Reserved	保留
[23:22]	ADC1STADCSEL	ADC1外部触发引脚(STADC) 触发选择 00 = 上升沿. 01 = 下降沿. 10 = 上升沿或下降沿. 11 = 保留
[21:20]	ADC1PWMTRGSEL	ADC1 PWM触发选择 00 = EPWM 信号下降沿. 01 = EPWM 计数器中间值. 10 = EPWM 信号上升沿. 11 = 周期.

位	描述	
[19:16]	ADC1TRGSOR	ADC1 触发源 0000 = STADC. 0001 = PWM0. 0010 = PWM1. 0011 = PWM2. 0100 = PWM3. 0101 = PWM4. 0110 = PWM5. 0111 = TMR0. 1000 = TMR1. 1001 = TMR2. 1010 = ADC0IF. 1011 = ADC1IF. 1100~1111 = 保留
[15:8]	Reserved	保留
[7:6]	ADC0STADCSEL	ADC0外部触发引脚(STADC) 触发选择 00 = 上升沿. 01 = 下降沿. 10 = 上升沿或下降沿. 11 = 保留
[5:4]	ADC0PWMTRGSEL	ADC0 PWM触发选择 00 = EPWM 信号下降沿. 01 = EPWM 计数器中间值. 10 = EPWM 信号上升沿. 11 = 周期.
[3:0]	ADC0TRGSOR	ADC0 触发源 0000 = STADC. 0001 = PWM0. 0010 = PWM1. 0011 = PWM2. 0100 = PWM3. 0101 = PWM4. 0110 = PWM5. 0111 = TMR0. 1000 = TMR1. 1001 = TMR2. 1010 = ADC0IF. 1011 = ADC1IF. 1100~1111 = 保留

ADC触发延时控制寄存器 (ADC_TRGDLY)

寄存器	偏移	R/W	描述	复位值
ADC_TRGDLY	ADC_BA+0x28	R/W	ADC触发延时控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
ADC1DELAY							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
ADC0DELAY							

位	描述	
[31:24]	Reserved	保留
[23:16]	ADC1DELAY	ADC1触发延时定时器 设置该域，在ADCxTRGCTL (x:0/1)触发到来之后，将延时ADC开始转换时间。 延时时间为 $(4 * \text{ADC1DELAY}) * \text{系统时钟}$
[15:8]	Reserved	保留
[7:0]	ADC0DELAY	ADC0触发延时定时器 设置该域，在ADCxTRGCTL (x:0/1)触发到来之后，将延时ADC开始转换时间。 延时时间为 $(4 * \text{ADC0DELAY}) * \text{系统时钟}$

ADC采样时间计数器寄存器 (ADC_SMPCNT)

寄存器	偏移	R/W	描述	复位值
ADC_SMPCNT	ADC_BA+0x2C	R/W	ADC采样时间计数器寄存器	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				ADCSMPCNT			

位	描述	
[31:4]	Reserved	保留
[3:0]	ADCSMPCNT	ADC 采样计数器 ADC 采样计数器建议为6个ADC时钟 0 = 1 * ADC 时钟. 1 = 2 * ADC 时钟. 2 = 3 * ADC 时钟. 3 = 4 * ADC 时钟. 4 = 5 * ADC 时钟. 5 = 6 * ADC 时钟. 6 = 7 * ADC 时钟. 7 = 8 * ADC 时钟. 8 = 16 * ADC 时钟. 9 = 32 * ADC 时钟. 10 = 64 * ADC 时钟. 11 = 128 * ADC 时钟. 12 = 256 * ADC 时钟. 13 = 512 * ADC 时钟. 14 = 1024 * ADC 时钟. 15 = 1024 * ADC 时钟.

ADC状态寄存器 (ADC_STATUS)

寄存器	偏移	R/W	描述	复位值
ADC_STATUS	ADC_BA+0x30	R/W	ADC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				HIGHFG	MIDFG	LOWFG	WCMPIF
15	14	13	12	11	10	9	8
ADC1CH				ADC1BUSY	Reserved	ADC1OV	ADC1IF
7	6	5	4	3	2	1	0
ADC0CH				ADC0BUSY	Reserved	ADC0OV	ADC0IF

位	描述	
[31:20]	Reserved	保留
[19]	HIGHFG	窗口比较器高范围标志 当A/D转换结果高于高范围(WCMPHIGHDAT)设置的条件，该位置1。 该位写1清0。 0 = 在ADC_DAT1的转换结果与WCMPHIGHDAT设置的不匹配。 1 = 在ADC_DAT1的转换结果与WCMPHIGHDAT设置的匹配。
[18]	MIDFG	窗口比较器中范围标志 当A/D转换结果在高范围 (WCMPHIGHDAT)和低范围(WCMPLOWDAT)之间, 该位置1。 该位写1清0。 0 = 在ADC_DAT1的转换结果不在高范围 (WCMPHIGHDAT) 与低范围(WCMPLOWDAT)之间。 1 = 在ADC_DAT1的转换结果在高范围 (WCMPHIGHDAT) 与低范围(WCMPLOWDAT)之间。
[17]	LOWFG	窗口比较器低范围标志 当A/D转换结果低于低范围(WCMPLOWDAT)设置的条件，该位置1。 该位写1清0。 0 = 在ADC_DAT1的转换结果与WCMPLOWDAT设置的不匹配。 1 = 在ADC_DAT1的转换结果与WCMPLOWDAT设置的匹配。
[16]	WCMPIF	窗口比较器中断标志 当窗口比较器输出结果，该位置1。 该位写1清0。 0 = 在ADC_DAT1的转换结果与WCMPLOWDAT设置的不匹配。 1 = 在ADC_DAT1的转换结果与WCMPLOWDAT设置的匹配。

位	描述	
[15:12]	ADC1CH	<p>当前转换通道</p> <p>当ADC1BUSY =1时，该域反映当前转换通道.</p> <p>当ADC1BUSY =0时，该域反映下一个转换通道.</p> <p>该位只读.</p>
[11]	ADC1BUSY	<p>BUSY/IDLE</p> <p>0 = A/D转换器处于空闲状态.</p> <p>1 = A/D转换器忙于转换中.</p> <p>该位为ADCR的ADST位的映像.</p>
[10]	Reserved	保留
[9]	ADC1OV	<p>超载标志</p> <p>该位为ADDR的OV位的镜像</p>
[8]	ADC1IF	<p>ADC1转换结束标志</p> <p>表明A/D转换完成的状态标志</p> <p>当A/D转换结束，ADF置1.</p> <p>该位写1清0</p>
[7:4]	ADC0CH	<p>当前转换通道</p> <p>当ADC0BUSY =1时，该域反映当前转换通道.</p> <p>当ADC0BUSY =0时，该域反映下一个转换通道.</p> <p>该位只读.</p>
[3]	ADC0BUSY	<p>BUSY/IDLE</p> <p>0 = A/D转换器处于空闲状态.</p> <p>1 = A/D转换器忙于转换中</p> <p>该位为ADCR的ADST位的镜像</p>
[2]	Reserved	保留
[1]	ADC0OV	<p>超载标志</p> <p>该位为ADDR的OV位的镜像</p>
[0]	ADC0IF	<p>A/D转换结束标志</p> <p>表明A/D转换完成的状态标志</p> <p>当A/D转换结束，ADF置1.</p> <p>该位写1清0</p>

ADC窗口比较器控制寄存器 (ADC_WCMPCTL)

寄存器	偏移	R/W	描述	复位值
ADC_WCMPCTL	ADC_BA+0x34	R/W	ADC窗口比较器控制寄存器	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				WCMPMCNT			
7	6	5	4	3	2	1	0
WFLAGCTL	WCMPHIGHEN	WCMPMIDEN	WCMPLOWEN	Reserved		WCMPIEN	WCPEN

位	描述	
[31:12]	Reserved	保留
[11:8]	WCMPMCNT	窗口比较器匹配计数 当A/D转换结果与CMPUPEN, CMPEQUEN, CMPLOWEN 和 WCFLAGCTL设置的比较条件匹配时, 内部匹配计数器加1. 当内部计数器的值达到WCMPMCNT, CMPIF位将置1 注:如果WCMPMCNT = 0,计数器将计数16次.
[7]	WFLAGCTL	窗口比较器标志控制 当A/D转换结果与比较条件匹配时 0 = 自动更新. 1 = 无.
[6]	WCMPHIGHEN	窗口比较器高标志使能位 设置A/D比较结果高于比较条件高范围 0 = 窗口比较器高标志禁用 1 = 窗口比较器高标志使能
[5]	WCMPMIDEN	窗口比较器中标志使能位 设置A/D转换结果在比较条件的低和高范围之间 0 = 窗口比较器中标志禁用. 1 = 窗口比较器中标志使能.
[4]	WCMPLOWEN	窗口比较器低标志使能位 设置A/D转换结果低于比较条件低范围 0 = 窗口比较器低标志禁用 1 = 窗口比较器低标志使能

位	描述	
[3:2]	Reserved	保留
[1]	WCMPEN	窗口比较器中断使能位 0 = 窗口比较器中断禁用. 1 = 窗口比较器中断使能
[0]	WCMPEN	窗口比较器使能位 0 = 窗口比较器禁用. 1 = 窗口比较器使能.

ADC窗口比较器数据寄存器 (ADC_WCMPDAT)

寄存器	偏移	R/W	描述	复位值
ADC_WCMPDAT	ADC_BA+0x38	R/W	ADC窗口比较器数据寄存器	0x0000_0000

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

位	描述	
[31:28]	Reserved	保留
[27:16]	WCMPHIGHDAT	窗口比较器高范围值
[15:12]	Reserved	保留
[11:0]	WCMPLOWDAT	窗口比较器低范围值

6.17 模拟比较器 (ACMP)

6.17.1 概述

NM1120系列包含两个比较器，通过不同配置可用于多种场合。当正输入大于负输入，比较器输出1，否则输出0。每个比较器可以配置为当比较器输出值变化时产生中断。

6.17.2 特性

- 模拟输入电压范围: $0 \sim V_{DD}$
- 支持迟滞功能
- 每个比较器负输入支持可选的内部参考电压源

6.17.3 方框图

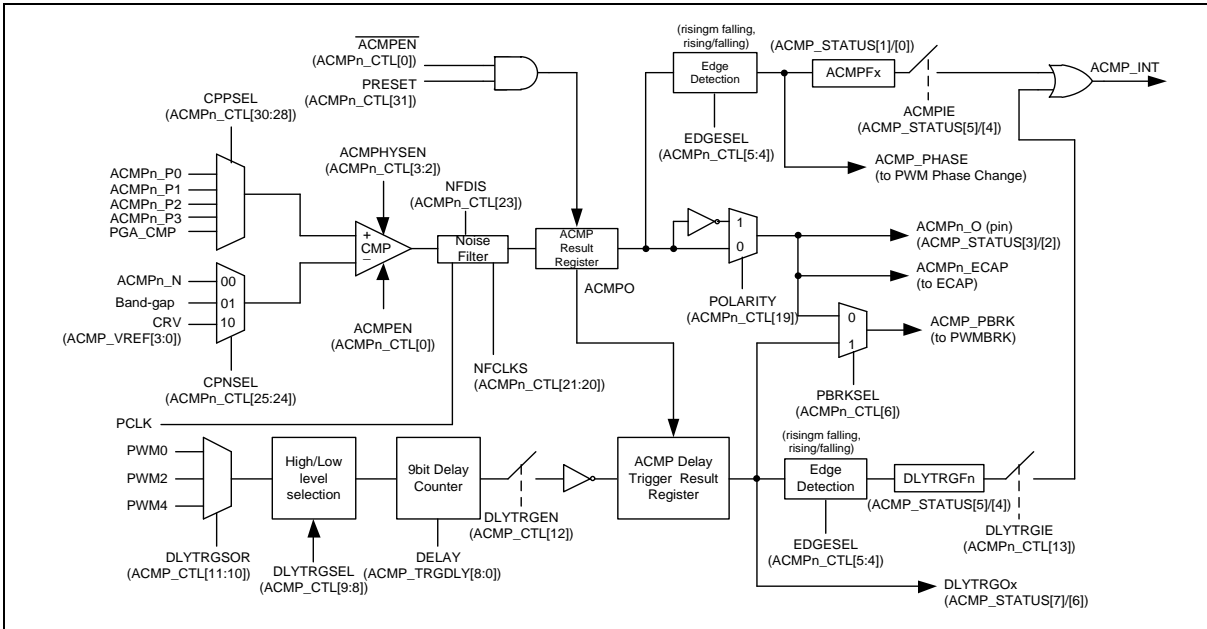


图 6.17-1 模拟比较器方框图

6.17.4 基本配置

ACMP引脚功能在SYS_GPB_MFP, SYS_GPC_MFP 和 SYS_GPD_MFP寄存器中配置，推荐禁用模拟输入引脚的数字输入通道，避免漏电。数字输入通道可以通过PB_DINOFF, PC_DINOFF 和 PD_DINOFF寄存器禁用。如果GPIO引脚配置为ACMP输入引脚，该引脚不能通过Px_MODE寄存器设置为推挽模式，输入模式是最安全的配置。如果配置为开漏模式、输出模式或准双向模式，不要输出0。GPIO默认输出1。PB_MODE, PC_MODE 和 PD_MODE的默认设置取决于用户配置，可配置为输入或准双向模式。

ACMP 外设时钟可以通过设置ACMPCKEN (CLK_APBCLK [30])为1来使能。

6.17.5 功能描述

6.17.5.1 中断源

比较器输出通过PCLK采样，结果反馈在ACMPOx(ACMP_STATUS[3] 和 ACMP_STATUS[2])。如果ACMPIE(ACMP_CTLx[1])设置为1，将使能比较器中断。如果比较器的输出状态改变，比较器将产生中断，相应的标志ACMPF_x(ACMP_STATUS[1] 和 ACMP_STATUS[0])将置1。软件可以通过写1来清0该位。

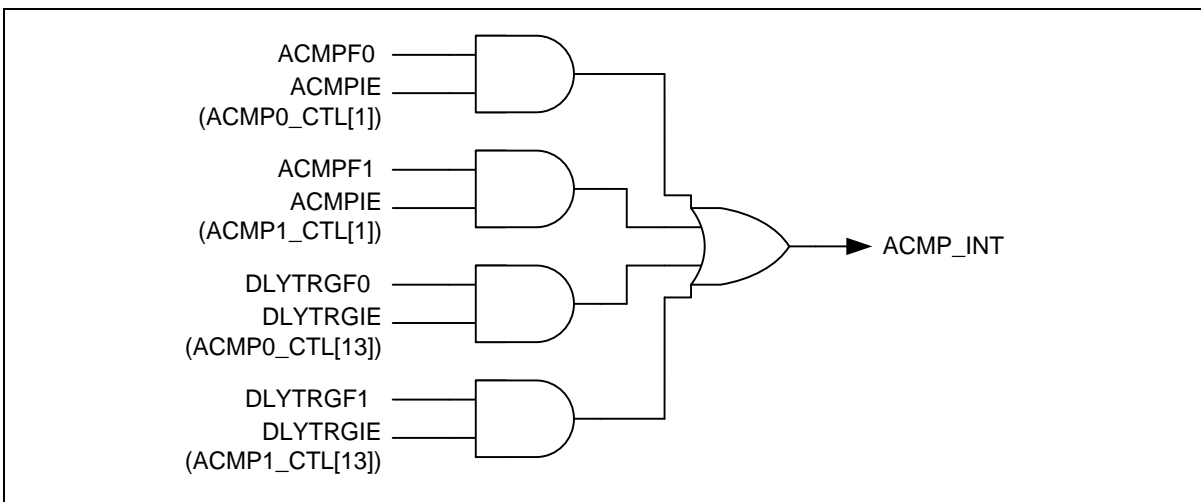


图 6.17-2 模拟比较器控制器中断源

6.17.5.2 迟滞功能

模拟比较器提供迟滞功能来促使比较器输出更平稳。如果输出0，它将不会输出1，直到正输入电压超过负输入电压一个正迟滞电压。同样，如果比较器输出1，它将不会输出0，直到正输入电压下降到负输入电压一个负迟滞电压。

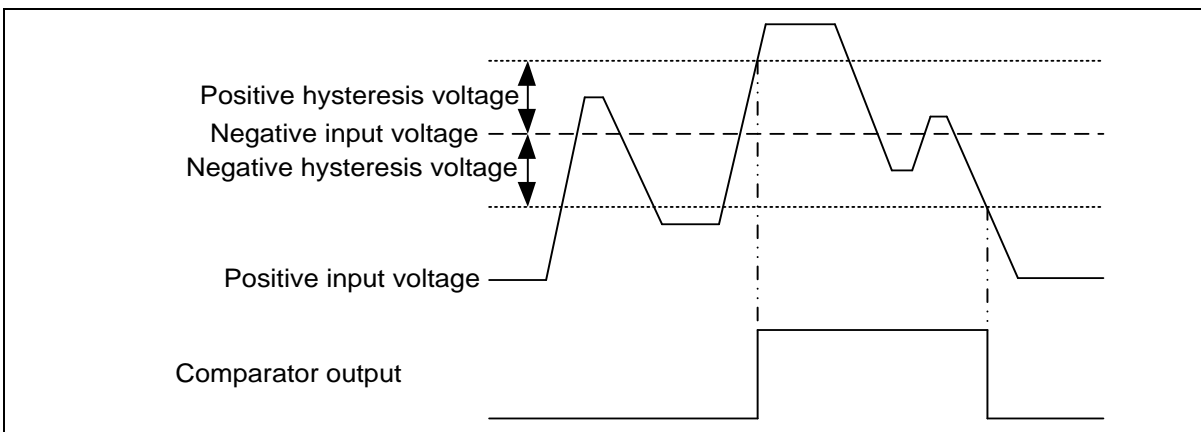


图 6.17-3 比较器迟滞功能

6.17.6 比较器参考电压 (CRV)

6.17.6.1 介绍

比较器参考电压(CRV)模块负责产生比较器的参考电压。CRV模块包含阶梯电阻和模拟开关，用户可以使用 CRVCTL(ACMP_VREF[3:0]) 设置 CRV 输出电压，通过设置 CPNSEL (ACMP_CTL[25:24])选择ACMP的参考电压。

6.17.6.2 特性

- 用户通过设置CRVCTL(ACMP_VREF [3:0])选择参考电压
- 当设置CPNSEL (ACMP_CTL[25:24]) = 01b (选择Band-gap电压源)时，将自动禁用阶梯电阻，减小功耗。

CRV模块方框图如图 6.17-4.

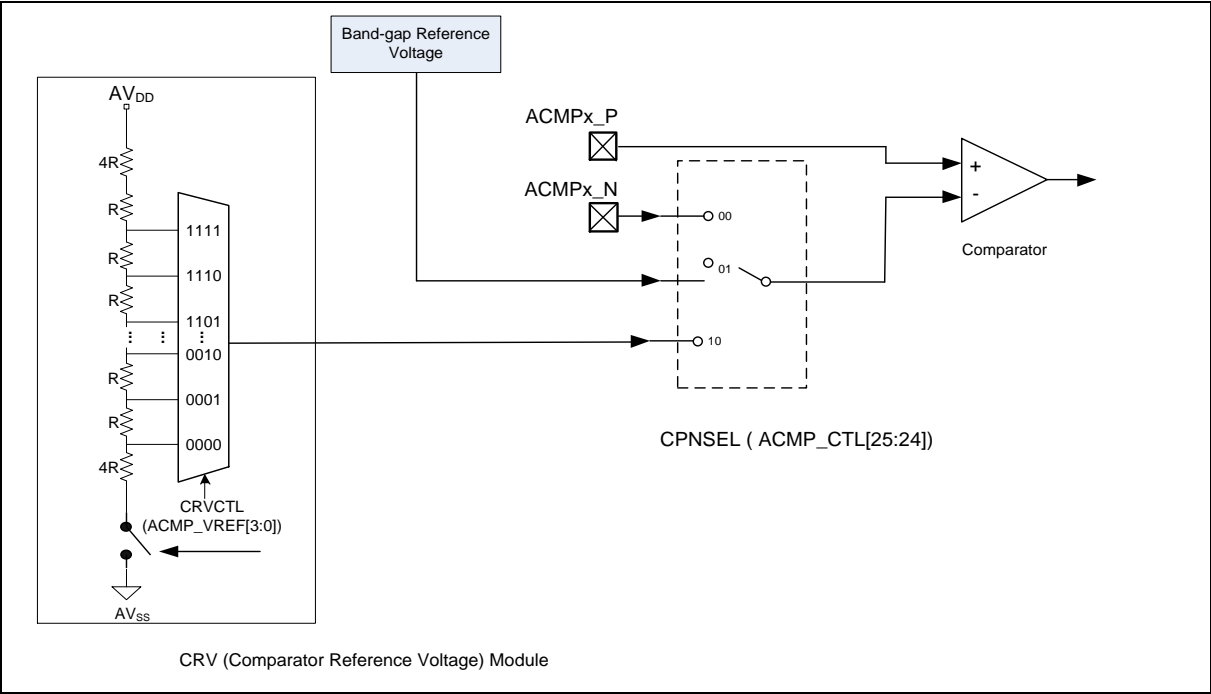


图 6.17-4 比较器参考电压方框图

6.17.7 寄存器映射

R: 只读, W: 只写, R/W: 读/写

寄存器	偏移	R/W	描述	复位值
ACMP基地址: ACMP_BA = 0x400D_0000				
ACMP_CTL0	ACMP_BA+0x00	R/W	模拟比较器0控制寄存器	0x0000_0000
ACMP_CTL1	ACMP_BA+0x04	R/W	模拟比较器1控制寄存器	0x0000_0000
ACMP_STATUS	ACMP_BA+0x08	R/W	模拟比较器状态寄存器	0x0000_0000
ACMP_VREF	ACMP_BA+0x0C	R/W	模拟比较器参考电压控制寄存器	0x0000_0000
ACMP_TRGDLY	ACMP_BA+0x10	R/W	模拟比较器延时触发模式延时寄存器	0x0000_0000

6.17.8 寄存器描述

模拟比较器0控制寄存器 (ACMP_CTL0)

寄存器	偏移	R/W	描述	复位值
ACMP_CTL0	ACMP_BA+0x00	R/W	模拟比较器0控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRESET	CPPSEL			Reserved		CPNSEL	
23	22	21	20	19	18	17	16
NFDIS	Reserved	NFCLKS		POLARITY	Reserved		
15	14	13	12	11	10	9	8
Reserved		DLYTRGIE	DLYTRGEN	DLYTRGSOR		DLYTRGSEL	
7	6	5	4	3	2	1	0
Reserved	PBRKSEL	EDGESEL		ACMPHYSEN		ACMPIE	ACMPEN

位	描述	
[31]	PRESET	比较器结果预先调整值 0 = 预先调整值为0. 1 = 预先调整值为1.
[30:28]	CPPSEL	比较器正输入选择 000 = ACMP0_P0 (PB.0). 001 = ACMP0_P1 (PB.1). 010 = ACMP0_P2 (PB.2). 011 = ACMP0_P3 (PC.1). 100 = PGA_CMP.
[27:26]	Reserved	保留
[25:24]	CPNSEL	比较器负输入选择 00 = ACMP0_N (PB.4). 01 = Band_Gap. 10 = CRV. 11 = 保留
[23]	NFDIS	禁用比较器噪声滤波器 0 = 噪声滤波器使能. 1 = 噪声滤波器禁用.
[22]	Reserved	保留

位	描述	
[21:20]	NFCLKS	噪音滤波器时钟预分频选择 决定噪音滤波器时钟的采样频率 00 = PCLK. 01 = PCLK / 2. 10 = PCLK / 4. 11 = PCLK / 16.
[19]	POLARITY	模拟比较器极性控制 0 = 模拟比较器正常输出. 1 = 模拟比较器反转输出.
[18:14]	Reserved	保留
[13]	DLYTRGIE	模拟比较器延时触发模式中断使能位 0 = 模拟比较器延时触发模式中断禁用. 1 = 模拟比较器延时触发模式中断使能.
[12]	DLYTRGEN	模拟比较器延时触发模式使能位 0 = 模拟比较器延时触发模式禁用. 1 = 模拟比较器延时触发模式使能.
[11:10]	DLYTRGSOR	模拟比较器延时触发模式触发源选择 00 = PWM0. 01 = PWM2. 10 = PWM4. 11 = 保留
[9:8]	DLYTRGSEL	模拟比较器延时触发模式触发边沿选择 00 = 模拟比较器延时触发模式触发禁用. 01 = 上升沿. 10 = 下降沿. 11 = 上升沿/下降沿.
[7]	Reserved	保留
[6]	PBRKSEL	ACMP作为EPWM刹车选择 0 = ACMP结果直接输出. 1 = ACMP延时触发结果输出.
[5:4]	EDGESEL	中断标志触发边沿选择检测 00 = 中断标志触发边沿禁用. 01 = 上升沿. 10 = 下降沿. 11 = 上升沿/下降沿.

位	描述	
[3:2]	ACMPHYSEN	<p>比较器0迟滞使能位 (只能 20mV)</p> <p>00 = ACMP0 迟滞功能禁用 (默认).</p> <p>01 = ACMP0 迟滞功能使能, 典型范围值10mV.</p> <p>10 = ACMP0 迟滞功能使能, 典型范围值90mV.</p> <p>11 = ACMP0 迟滞功能禁用.</p>
[1]	ACMPIE	<p>比较器中断使能位</p> <p>0 = ACMP中断功能禁用.</p> <p>1 = ACMP中断功能使能.</p> <p>注1: 如果ACMPIE位置1, 在ACMP转换完成后, 中断将产生.</p> <p>注2: ACMP中断会将CPU从Power-down模式唤醒.</p>
[0]	ACMPEN	<p>比较器使能位</p> <p>0 = 比较器禁用.</p> <p>1 = 比较器使能.</p> <p>注: 比较器输出在ACMPEN设置之后, 必须等待2us稳定时间.</p>

模拟比较器1控制寄存器 (ACMP_CTL1)

寄存器	偏移	R/W	描述	复位值
ACMP_CTL1	ACMP_BA+0x04	R/W	模拟比较器1控制寄存器	0x0000_0000

31	30	29	28	27	26	25	24
PRESET	CPPSEL			Reserved		CPNSEL	
23	22	21	20	19	18	17	16
NFDIS	Reserved	NFCLKS		POLARITY	Reserved		
15	14	13	12	11	10	9	8
Reserved		DLYTRGIE	DLYTRGEN	DLYTRGSOR		DLYTRGSEL	
7	6	5	4	3	2	1	0
Reserved	PBRKSEL	EDGESEL		ACMPHYSEN		ACMPIE	ACMPEN

位	描述	
[31]	PRESET	比较器结果预先调整值 0 = 预先调整值为0. 1 = 预先调整值为1.
[30:28]	CPPSEL	比较器正输入选择 000 = ACMP1_P0 (PC.0). 001 = ACMP1_P1 (PC.1). 010 = ACMP1_P2 (PD.1). 011 = PGA_CMP.
[27:26]	Reserved	保留
[25:24]	CPNSEL	比较器负输入选择 00 = ACMP1_N (PB.3). 01 = Band_Gap. 10 = CRV. 11 = 保留
[23]	NFDIS	禁用比较器噪音滤波器 0 = 噪音滤波器使能. 1 = 噪音滤波器禁用.
[22]	Reserved	保留

位	描述	
[21:20]	NFCLKS	噪音滤波器时钟预分频选择 决定噪音滤波器时钟的采样频率 00 = PCLK. 01 = PCLK / 2. 10 = PCLK / 4. 11 = PCLK / 16.
[19]	POLARITY	模拟比较器极性控制 0 = 模拟比较器正常输出. 1 = 模拟比较器反转输出.
[18:14]	Reserved	保留
[13]	DLYTRGIE	模拟比较器延时触发模式中断使能 0 = 模拟比较器延时触发模式中断禁用. 1 = 模拟比较器延时触发模式中断使能.
[12]	DLYTRGEN	模拟比较器延时触发模式使能 0 = 模拟比较器延时触发模式禁用. 1 = 模拟比较器延时触发模式使能.
[11:10]	DLYTRGSOR	模拟比较器延时触发模式触发源选择 00 = PWM0. 01 = PWM2. 10 = PWM4. 11 = 保留
[9:8]	DLYTRGSEL	模拟比较器延时触发模式触发边沿选择 00 = 模拟比较器延时触发模式触发禁用. 01 = 上升沿. 10 = 下降沿. 11 = 上升沿/下降沿.
[7]	Reserved	保留
[6]	PBRKSEL	ACMP作为EPWM 刹车选择 0 = ACMP结果直接输出. 1 = ACMP延时触发结果输出.
[5:4]	EDGESEL	中断标志触发边沿检测 00 = 中断标志触发边沿检测禁用. 01 = 上升沿. 10 = 下降沿. 11 = 上升沿/下降沿.

位	描述	
[3:2]	ACMPHYSEN	<p>比较器1迟滞使能位 (只有 20mV)</p> <p>00 = ACMP0迟滞功能禁用 (默认).</p> <p>01 = ACMP0迟滞功能使能, 典型范围10mV.</p> <p>10 = ACMP0迟滞功能使能, 典型范围90mV.</p> <p>11 = ACMP0迟滞功能禁用.</p>
[1]	ACMPIE	<p>比较器中断使能位</p> <p>0 = ACMP中断功能禁用.</p> <p>1 = ACMP中断功能使能.</p> <p>注1: 如果ACMPIE位置1, 在ACMP转换完成后, 中断将产生.</p> <p>注2: ACMP中断会将CPU从Power-down模式唤醒.</p>
[0]	ACMPEN	<p>比较器使能位</p> <p>0 = 比较器禁用.</p> <p>1 = 比较器使能.</p> <p>注: 比较器输出在ACMPEN设置之后, 必须等待2us稳定时间.</p>

模拟比较器状态寄存器 (ACMP_STATUS)

寄存器	偏移	R/W	描述	复位值
ACMP_STATUS	ACMP_BA+0x08	R/W	模拟比较器状态寄存器	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
DLYTRGO1	DLYTRGO0	DLYTRGF1	DLYTRGF0	ACMPO1	ACMPO0	ACMPF1	ACMPF0

位	描述	
[31:8]	Reserved	保留
[7]	DLYTRGO1	模拟比较器延时触发模式比较器输出 同步APB时钟来允许通过软件读，当比较器禁用(DLYTRGEN = 0)时，清除。
[6]	DLYTRGO0	模拟比较器0延时触发模式比较器输出 同步APB时钟来允许通过软件读，当比较器禁用(DLYTRGEN = 0)时，清除。
[5]	DLYTRGF1	比较器1标志 当比较器1输出状态改变时，该位由硬件置1。如果DLYTRGIEN置1，该位为1将触发中断。 注: 该位写1清0。
[4]	DLYTRGF0	比较器0标志 当比较器0输出状态改变时，该位由硬件置1。如果DLYTRGIEN置1，该位为1将触发中断。 注: 该位写1清0。
[3]	ACMPO1	比较器1输出 同步APB时钟来允许通过软件读，当比较器禁用(ACMPEN = 0)时，清除。
[2]	ACMPO0	比较器0输出 同步APB时钟来允许通过软件读，当比较器禁用(ACMPEN = 0)时，清除。
[1]	ACMPF1	比较器1标志 当比较器1输出状态改变时，该位由硬件置1。如果ACMPIE置1，该位为1将触发中断。 注: 该位写1清0。
[0]	ACMPF0	比较器0标志 当比较器0输出状态改变时，该位由硬件置1。如果ACMPIE置1，该位为1将触发中断。 注: 该位写1清0。

模拟比较器参考电压控制寄存器 (ACMP_VREF)

寄存器	偏移	R/W	描述	复位值
ACMP_VREF	ACMP_BA+0x0C	R/W	模拟比较器参考电压控制寄存器	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				CRVCTL			

位	描述	
[31:4]	Reserved	保留
[3:0]	CRVCTL	比较器参考电压设置 $CRVS = AV_{DD} \times (1/6 + CRV[3:0]/24)$.

模拟比较器延时触发模式延时寄存器 (ACMP_TRGDLY)

寄存器	偏移	R/W	描述	复位值
ACMP_TRGDLY	ACMP_BA+0x10	R/W	模拟比较器延时触发模式延时寄存器	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
Reserved							DELAY
7	6	5	4	3	2	1	0
DELAY							

位	描述	
[31:9]	Reserved	保留
[8:0]	DELAY	模拟比较器延时触发模式延时周期

6.18 可编程增益放大器 (PGA)

6.18.1 概述

NM1120系列包含一个可编程增益放大器，可通过PGAEN位使能.用户可以通过内部的A/D转换通道测量可编程增益放大器输出电压.此外，用户可以调整增益为1, 2, 3, 5, 7, 9, 11, 或 13.

注: 模拟输入端口引脚必须在PGA功能使能之前配置为输入类型.

6.18.2 特性

- 支持模拟输入电压范围: 0~ V_{DD}.
- 支持可编程增益: 1,2, 3,5,7,9,11,13
- 支持PGA输出作为ADC或ACMP的输入

6.18.3 方框图

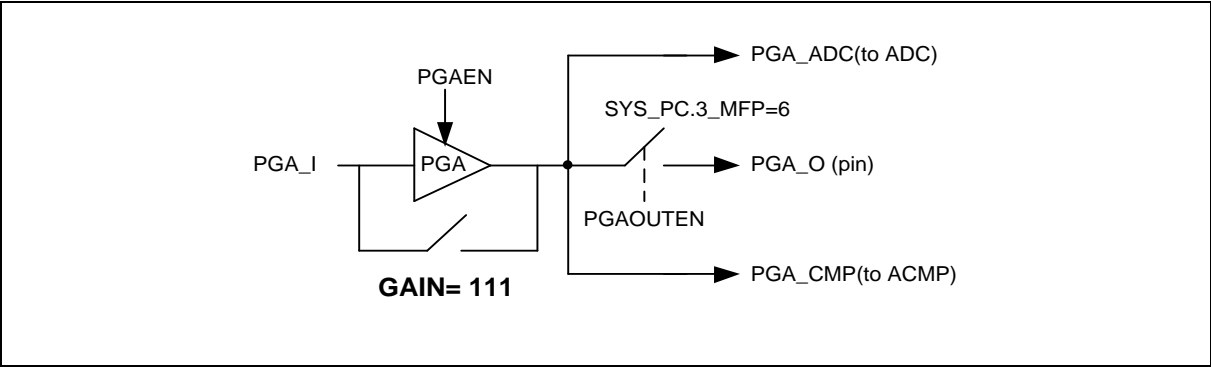


图 6.18-1 OP 放大器方框图

6.18.4 寄存器映射

R: 只读, W: 只写, R/W: 读/写

寄存器	偏移	R/W	描述	复位值
PGA基地址: PGA_BA = 0x400F_0000				
PGA_CTL	PGA_BA+0x00	R/W	可编程增益放大器控制寄存器	0x0000_0000

6.18.5 寄存器 描述

PGA控制寄存器 (PGA_CTL)

寄存器	偏移	R/W	描述	复位值
PGA_CTL	PGA_BA+0x00	R/W	可编程增益放大器控制寄存器	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	GAIN			Reserved			PGAEN

位	描述	
[31:7]	Reserved	保留
[6:4]	GAIN	PGA增益选择 000 = 2. 001 = 3. 010 = 5. 011 = 7. 100 = 9. 101 = 11. 110 = 13. 111 = 1. (*看注)
[3:1]	Reserved	保留
[0]	PGAEN	可编程增益放大器使能位 0 = 可编程增益放大器禁用. 1 = 可编程增益放大器使能. 注: PGA输出在PGAEN首次设置之后, 必须等待20us稳定时间.

注:: GAIN =111; PGA使用旁路通道; 软件必须写PGA_EN禁用.

7 应用电路

8 电气特性

8.1 绝对最大额定值

符号	参数	最小值	最大值	单位
$V_{DD}-V_{SS}$	直流电源电压	-0.3	+7.0	V
V_{IN}	输入电压	$V_{SS}-0.3$	$V_{DD}+0.3$	V
$1/t_{CLCL}$	晶振频率	4	24	MHz
T_A	工作温度	-40	+105	°C
T_{ST}	贮存温度	-55	+150	°C
I_{DD}	V_{DD} 最大流入电流	-	120	mA
I_{SS}	V_{SS} 最大流出电流	-	120	mA
I_{IO}	单一管脚最大灌电流	-	35	mA
	单一管脚最大流出电流	-	35	mA
	所有管脚最大灌电流总和	-	100	mA
	所有管脚最大输出电流总和	-	100	mA

注：上表所列的条件中，其极限值可能对设备的稳定有反作用

8.2 DC 电气特性

($V_{DD} - V_{SS} = 2.1 \sim 5.5 \text{ V}$, $T_A = 25^\circ\text{C}$)

符号	参数	最小值	典型值	最大值	单位	测试条件			
V_{DD}	工作电压	2.1	-	5.5	V	$V_{DD} = 2.1\text{V} \sim 5.5\text{V}$ up to 48 MHz			
V_{SS} / AV_{SS}	电源地	-0.3	-	-	V				
V_{LDO}	LDO 输出电压		1.5		V				
V_{BG}	带隙电压	1.14	1.2	1.24	V	$V_{DD} = 3.0\text{V} \sim 5.5\text{V}$, $T_A = -40^\circ\text{C} \sim 105^\circ\text{C}$			
I_{DD5}	正常运行模式下的工作电流 HCLK = 48 MHz while(1){ 运行在 Flash 中	-	9.7	-	mA	V_{DD}	HXT	HIRC	All Digital Modules
						5.5V	X	48 MHz	V
I_{DD6}		-	7.4	-	mA	5.5V	X	48 MHz	X
I_{DD7}		-	9.7	-	mA	3V	X	48 MHz	V
I_{DD8}		-	7.4	-	mA	3V	X	48 MHz	X
I_{DD1}	正常运行模式下的工作电流 HCLK = 24 MHz while(1){ 运行在 Flash 中	-	5.4	-	mA	V_{DD}	HXT	HIRC	All Digital Modules
						5.5V	24 MHz	X	V
I_{DD2}		-	4.4	-	mA	5.5V	24 MHz	X	X
I_{DD3}		-	5.4	-	mA	3V	24 MHz	X	V
I_{DD4}		-	4.4	-	mA	3V	24 MHz	X	X
I_{DD9}	正常运行模式下的工作电流 HCLK = 16 MHz while(1){ 运行在 Flash 中	-	3.7	-	mA	V_{DD}	HXT	HIRC	All Digital Modules
						5.5V	16 MHz	X	V
I_{DD10}		-	3.0	-	mA	5.5V	16 MHz	X	X
I_{DD11}		-	3.7	-	mA	3V	16 MHz	X	V
I_{DD12}		-	3.1	-	mA	3V	16 MHz	X	X
I_{DD9}	正常运行模式下的工作电流 HCLK = 12 MHz	-	2.8	-	mA	V_{DD}	HXT	HIRC	All Digital Modules
						5.5V	12 MHz	X	V

I _{DD10}	while(1){ 运行在 Flash 中	-	2.3	-	mA	5.5V	12 MHz	X	X
I _{DD11}		-	2.8	-	mA	3V	12 MHz	X	V
I _{DD12}		-	2.3	-	mA	3V	12 MHz	X	X
I _{DD13}	正常运行模式下的工作电流	-	1.2	-	mA	V _{DD}	HXT	HIRC	All Digital Modules
						5.5V	4 MHz	X	V
I _{DD14}	HCLK = 4 MHz while(1){	-	1.0	-	mA	5.5V	4 MHz	X	X
I _{DD15}	运行在 Flash 中	-	1.2	-	mA	3V	4 MHz	X	V
I _{DD16}		-	1.0	-	mA	3V	4 MHz	X	X
I _{DD17}	正常运行模式下的工作电流	-	291.7	-	μA	V _{DD}	LXT	LIRC	All Digital Modules
						5.5V	32 kHz	V	V ^[1]
I _{DD18}	HCLK = 32 kHz while(1){	-	290.7	-	μA	5.5V	32 kHz	V	X
I _{DD19}	运行在 Flash 中	-	280.8	-	μA	3V	32 kHz	V	V ^[1]
I _{DD20}		-	281.4	-	μA	3V	32 kHz	V	X
I _{DD17}	正常运行模式下的工作电流	-	248.0	-	μA	V _{DD}	HXT	LIRC	All Digital Modules
						5.5V	X	10 kHz	V ^[2]
I _{DD18}	HCLK = 10 kHz while(1){	-	247.7	-	μA	5.5V	X	10 kHz	X
I _{DD19}	运行在 Flash 中	-	237.9	-	μA	3V	X	10 kHz	V ^[2]
I _{DD20}		-	237.5	-	μA	3V	X	10 kHz	X
I _{IDLE5}	空闲模式下的工作电流	-	4.9	-	mA	V _{DD}	HXT	HIRC	All Digital Modules
						5.5V	X	V	V
I _{IDLE6}	HCLK = 48 MHz	-	2.6	-	mA	5.5V	X	V	X
I _{IDLE7}		-	4.9	-	mA	3V	X	V	V
I _{IDLE8}		-	2.6	-	mA	3V	X	V	X
I _{IDLE1}	空闲模式下的工作电流 HCLK = 24 MHz	-	2.8	-	mA	V _{DD}	HXT	HIRC	All Digital Modules
						5.5V	24 MHz	X	V

I _{IDLE2}		-	1.9	-	mA	5.5V	24 MHz	X	X
I _{IDLE3}		-	2.8	-	mA	3V	24 MHz	X	V
I _{IDLE4}		-	1.9	-	mA	3V	24 MHz	X	X
I _{IDLE9}	空闲模式下的工作电流 HCLK = 16 MHz	-	2.0	-	mA	V _{DD}	HXT	HIRC	All Digital Modules
						5.5V	V	X	V
I _{IDLE10}		-	1.3	-	mA	5.5V	V	X	X
I _{IDLE11}		-	2.0	-	mA	3V	V	X	V
I _{IDLE12}		-	1.4	-	mA	3V	V	X	X
I _{IDLE9}	空闲模式下的工作电流 HCLK = 12 MHz	-	1.5	-	mA	V _{DD}	HXT	HIRC	All Digital Modules
						5.5V	V	X	V
I _{IDLE10}		-	1.0	-	mA	5.5V	V	X	X
I _{IDLE11}		-	1.5	-	mA	3V	V	X	V
I _{IDLE12}		-	1.0	-	mA	3V	V	X	X
I _{IDLE13}	空闲模式下的工作电流 HCLK = 4 MHz	-	0.8	-	mA	V _{DD}	HXT	HIRC	All Digital Modules
						5.5V	V	X	V
I _{IDLE14}		-	0.6	-	mA	5.5V	V	X	X
I _{IDLE15}		-	0.7	-	mA	3V	V	X	V
I _{IDLE16}		-	0.6	-	mA	3V	V	X	X
I _{DD17}	空闲模式下的工作电流 HCLK = 32 kHz	-	274.3	-	μA	V _{DD}	HXT	LIRC	All Digital Modules
						5.5V	X	V	V ^[1]
I _{DD18}		-	273.0	-	μA	5.5V	X	V	X
I _{DD19}		-	265.0	-	μA	3V	X	V	V ^[1]
I _{DD20}		-	263.9	-	μA	3V	X	V	X
I _{DD17}	空闲模式下的工作电流 HCLK = 10 kHz	-	232.6	-	μA	V _{DD}	HXT	LIRC	All Digital Modules
						5.5V	X	V	V ^[2]

I _{DD18}		-	232.2	-	μA	5.5V	X	V	X
I _{DD19}		-	222.5	-	μA	3V	X	V	V ^[2]
I _{DD20}		-	222.1	-	μA	3V	X	V	X
I _{PWD1}	掉电模式下的待机电流 (深度睡眠模式)	-	1.9	-	μA	V _{DD} = 5.5 V, 所有振荡器和模拟模块关闭			
I _{PWD2}		-	1.7	-	μA	V _{DD} = 3 V, 所有振荡器和模拟模块关闭			
I _{LK}	PA/PB/PC/PD 输入漏电流	-1	-	+1	μA	V _{DD} = 5.5 V, 0 < V _{IN} < V _{DD} 开漏或输入模式			
V _{IL1}	PA/PB/PC/PD 输入低电压 (TTL 输入)	-0.3	1.33		V	V _{DD} = 5.5 V			
		-0.3	1			V _{DD} = 3.3 V			
V _{IH1}	PA/PB/PC/PD 输入高电压 (TTL 输入)		1.47	V _{DD} + 0.3	V	V _{DD} = 5.5 V			
			1.08	V _{DD} + 0.3		V _{DD} = 3.3 V			
V _{ILS}	nRESET 负向阈值电压 (Schmitt 输入)	-	-	0.3V _{DD}	V	-			
V _{IHS}	nRESET 正向阈值电压 (Schmitt 输入)	0.7V _{DD}	-	-	V	-			
R _{RST}	nRESET 脚内部上拉电阻	48		148	kΩ	V _{DD} = 2.1 V ~ 5.5V			
V _{ILS}	PA/PB/PC/PD 负向阈值电压 (Schmitt 输入)	-	-	0.3V _{DD}	V	-			
V _{IHS}	PA/PB/PC/PD 正向阈值电压 (Schmitt 输入)	0.7V _{DD}	-	-	V	-			
I _{IL}	PA/PB/PC/PD 逻辑 0 输入 电流 (准双向模式)	-	-63.65		μA	V _{DD} = 5.5 V, V _{IN} = 0V			
I _{TL}	PA/PB/PC/PD 逻辑 1 到 0 的转换电流	-	-566.7	-	μA	V _{DD} = 5.5 V			
I _{SR11}	PA/PB/PC/PD 源电流 (准 双向模式)	-	-372	-	μA	V _{DD} = 4.5 V, V _{IN} = 2.4 V			
I _{SR12}		-	-76.8	-	μA	V _{DD} = 2.7 V, V _{IN} = 2.2 V			
I _{SR13}		-	-37.3	-	μA	V _{DD} = 2.1 V, V _{IN} = 1.8 V			
I _{SR21}	PA/PB/PC/PD 源电流 (推 挽模式)	-	-19.2	-	mA	V _{DD} = 4.5 V, V _{IN} = 2.4 V			
I _{SR22}		-	-4	-	mA	V _{DD} = 2.7 V, V _{IN} = 2.2 V			
I _{SR23}		-	-2	-	mA	V _{DD} = 2.1 V, V _{IN} = 1.8 V			
I _{SK11}	PA/PB/PC/PD 灌电流 (准 双向、开漏和推挽模式)	-	12.8	-	mA	V _{DD} = 4.5 V, V _{IN} = 0.4 V			
I _{SK12}		-	8.1	-	mA	V _{DD} = 2.7 V, V _{IN} = 0.4 V			

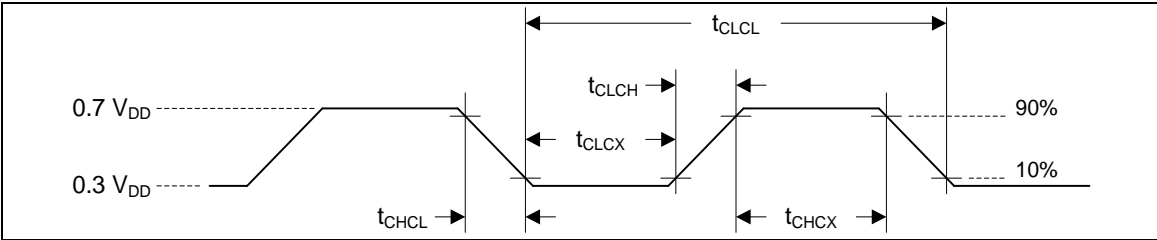
I_{SK13}		-	6	-	mA	$V_{DD} = 2.1\text{ V}, V_{IN} = 0.4\text{ V}$
------------	--	---	---	---	----	--

注:

- 1. 只使能支持32 kHz LIRC时钟源的IP
- 2. 只使能支持10 kHz LIRC时钟源的IP

8.3 AC 电气特性

8.3.1 外部高速晶振



注意: 占空比为 50%.

符号	参数	最小值	典型值	最大值	单位	测试条件
t _{CHCX}	时钟高电平时间	10	-	-	ns	-
t _{CLCX}	时钟低电平时间	10	-	-	ns	-
t _{CLCH}	时钟上升沿时间	2	-	15	ns	-
t _{CHCL}	时钟下降沿时间	2	-	15	ns	-

8.3.2 外部 4~24 MHz 高速晶振 (HXT)

符号	参数	最小值	典型值	最大值	单位	测试条件
V _{HXT}	工作电压	2.1	-	5.5	V	-
T _A	温度	-40	-	105	°C	-
I _{HXT}	工作电流	-	414	-	uA	12 MHz, V _{DD} = 5.5V
		-	407	-	uA	12 MHz, V _{DD} = 3.3V
f _{HXT}	输入时钟频率	4	-	24	MHz	-

8.3.3 外部 32.768 kHz XTAL 晶振 (LXT)

符号	参数	规格				测试条件
		最小值	典型值	最大值	单位	
f _{LXTAL}	晶振频率		32.768		kHz	
T _{LXTAL}	温度	-40		105	°C	
I _{LXTAL}	工作电流		17		μA	V _{DD} =2.1V

8.3.4 外部晶振的典型应用电路

晶振	C1	C2
----	----	----

4 MHz ~ 24 MHz	20 pF	20 pF
32.768 kHz	20 pF	20 pF

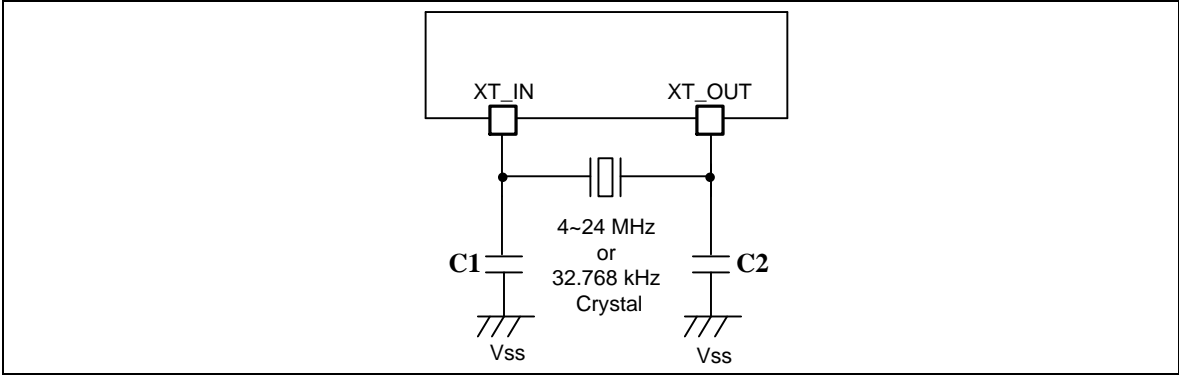


图 8.3-1 NM1120 典型晶振应用电路

8.3.5 48 MHz 内部高速 RC 振荡器 (HIRC)

符号	参数	最小值	典型值	最大值	单位	测试条件
V_{HIRC}	工作电压	-	1.5	-	V	-
f_{HIRC}	中心频率	-	48	-	MHz	-
	内部振荡器频率校准	-1	-	+1	%	$T_A = 25\text{ }^{\circ}\text{C}$ $V_{DD} = 5.5\text{ V}$
		-2		2	%	$T_A = -40^{\circ}\text{C} \sim 105^{\circ}\text{C}$ $V_{DD} = 2.1\text{ V} \sim 5.5\text{ V}$
I_{HIRC}	工作电流	-	1.1	-	mA	$T_A = 25\text{ }^{\circ}\text{C}, V_{DD} = 5\text{ V}$

8.3.6 内部低速10 kHz RC振荡器(LIRC)

符号	参数	最小值	典型值	最大值	单位	测试条件
V_{LRC}	支持电压	-	1.5V	-	V	-
f_{LRC}	中心频率	-	10	-	kHz	-
	振荡器频率	-50 ^[1]	-	+50 ^[1]	%	$V_{DD} = 2.1\text{ V} \sim 5.5\text{ V}$ $T_A = -40^{\circ}\text{C} \sim +105^{\circ}\text{C}$
I_{LRC}	工作电流	-	0.3	0.5	μA	$T_A = 25\text{ }^{\circ}\text{C}, V_{DD} = 5\text{ V}$

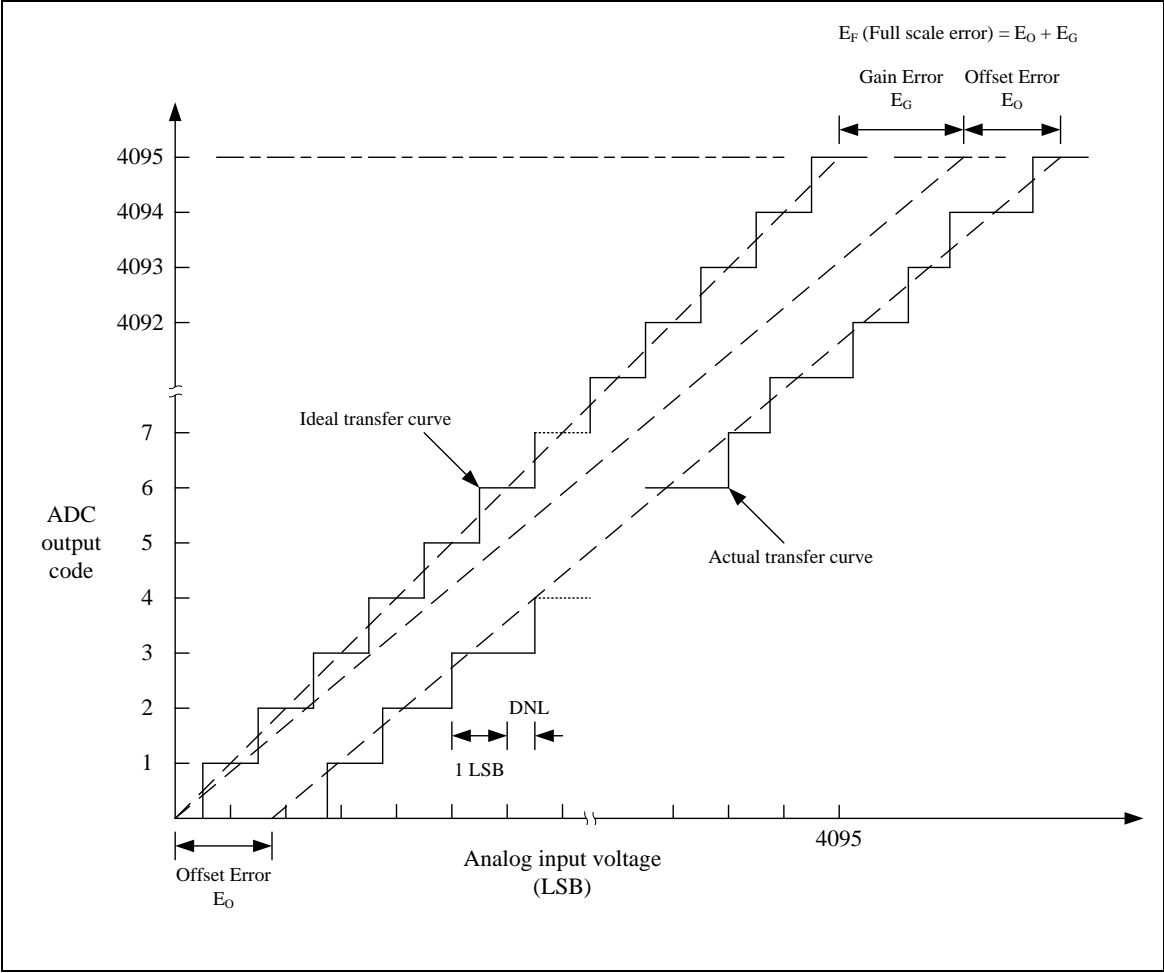
注： Guaranteed by design, not test in production.

8.4 模拟量特性

8.4.1 12-bit SAR ADC

符号	参数	最小值	典型值	最大值	单位	测试条件
-	分辨率	-	-	12	Bit	-
DNL	非线性差分误差	-	2	-	LSB	$V_{DD} = 3.0 \sim 5.5 \text{ V}$
INL	非线性整形误差	-	± 2	-	LSB	$V_{DD} = 3.0 \sim 5.5 \text{ V}$
E_O	偏移误差	-	± 1	-	LSB	$V_{DD} = 3.0 \sim 5.5 \text{ V}$
E_G	增益误差 (传输增益)	-	-1	-	LSB	$V_{DD} = 3.0 \sim 5.5 \text{ V}$
E_A	绝对误差	-	± 3	-	LSB	$V_{DD} = 3.0 \sim 5.5 \text{ V}$
-	一致性	保证			-	-
F_{ADC}	ADC 时钟频率		12	16	MHz	$V_{DD} = 3.0 \sim 5.5 \text{ V}$
F_S	采样率 (F_{ADC}/T_{CONV})			700	kSPS	$V_{DD} = 3.0 \sim 5.5 \text{ V}$
T_{ACQ}	采样时间 (阶段采样)	N+1			1/ F_{ADC}	$V_{DD} = 3.0 \sim 5.5 \text{ V}$ N is 采样计数, N=1~1024
		200			ns	$V_{DD} = 3.0 \sim 5.5 \text{ V}$
T_{CONV}	转换时间		1000	1050	ns	$V_{DD} = 3.0 \sim 5.5 \text{ V}$
V_{DD}	支持电压	3.0	-	5.5	V	-
I_{DDA}	支持电流 (平均.)	-	1	-	mA	$V_{DD} = 5.5 \text{ V}$
V_{IN}	输入电压范围	0	-	AV_{DD}	V	-
C_{IN}	输入电容	-	1.6	-	pF	-
R_{IN}	输入负载	-	2.5	-	k Ω	-

注: ADC 参考电压与 V_{DD} 一样



8.4.2 LDO & 电源管理

符号	参数	最小值	典型值	最大值	单位	测试条件
V_{DD}	输入电压	2.1	-	5.5	V	-
V_{LDO}	输出电压		1.5		V	-
T_A	温度	-40	25	105	°C	

注：建议接一颗 0.1μF 旁路电容在 VDD 引脚与最近的 VSS 引脚之间

8.4.3 低电压复位

符号	参数	最小值	典型值	最大值	单位	测试条件
AV_{DD}	操作电压	2.1	-	5.5	V	-
T_A	温度	-40	25	105	°C	-

I_{LVR}	静态电流			1	μA	$T_A=25^{\circ}C$
V_{LVR}	阈值电压	1.8	1.9	2.0	V	$T_A = -40^{\circ}C \sim +105^{\circ}C$

8.4.4 欠压检测

符号	参数	最小值	典型值	最大值	单位	测试条件
AV_{DD}	操作电压	0	-	5.5	V	-
T_A	温度	-40	25	105	$^{\circ}C$	-
I_{BOD}	静态电流	-	100	-	μA	$AV_{DD}=5.5V$
V_{BOD}	欠压迟滞电压	4.33	4.3	4.39	V	BOV_VL [2:0] = 3
		4.03	4.0	4.10	V	BOV_VL [2:0] = 2
		3.73	3.7	3.79	V	BOV_VL [2:0] = 7
		3.02	3.0	3.09	V	BOV_VL [2:0] = 1
		2.72	2.7	2.79	V	BOV_VL [2:0] = 6
		2.42	2.4	2.49	V	BOV_VL [2:0] = 0
		2.22	2.2	2.30	V	BOV_VL [2:0] = 5
		2.02	2.0	2.09	V	BOV_VL [2:0] = 4
V_{BOD}	欠压检测		4.3		V	BOV_VL [2:0] = 3
			4.0		V	BOV_VL [2:0] = 2
			3.7		V	BOV_VL [2:0] = 7
			3.0		V	BOV_VL [2:0] = 1
			2.7		V	BOV_VL [2:0] = 6
			2.4		V	BOV_VL [2:0] = 0
			2.2		V	BOV_VL [2:0] = 5
			2.0		V	BOV_VL [2:0] = 4

8.4.5 上电复位

符号	参数	最小值	典型值	最大值	单位	测试条件
T_A	温度	-40	25	105	$^{\circ}C$	-
V_{POR}	阈值电压		1.75		V	$V_{DD} = 5.0 V$
V_{POR}	确保上电复位的启动电压 V_{DD}		TBD		mV	
RR_{VDD}	确保上电复位电压 V_{DD} 上升率		TBD		V/ms	
t_{POR}	确保上电复位需要电压 V_{DD} 保持的最短时间 V_{POR}		TBD		ms	

8.4.6 比较器规格

符号	参数	最小值	典型值	最大值	单位	测试条件
V_{CMP}	工作电压	2.1	-	5.5	V	
T_A	温度	-40	25	105	°C	-
I_{CMP}	工作电流	-	46		μA	$V_{DD}=3.3V$
V_{OFF}	输入偏移电压		±10		mV	-
V_{SW}	输出摆幅	0	-	V_{DD}	V	-
V_{COM}	输入范围	0.1	-	$AV_{DD} - 0.1$	V	-
-	DC 增益 ^[1]	-	60	-	dB	-
T_{PGD}	传输延迟	-	225	-	ns	
V_{HYS}	迟滞	-	10	-	mV	ACMPPHYSEN = 01
V_{HYS}	迟滞	-	90	-	mV	ACMPPHYSEN = 10
T_{STB}	稳定时间	-	1.06	-	μs	

注: Guaranteed by design, not test in production.

8.4.7 PGA

符号	参数	最小值	典型值	最大值	单位	测试条件
	工作电压范围	2.5	3.3	5.5	V	
	工作电流			5	mA	$V_{DD}=5V, T=125^{\circ}C$
	工作温度	-40	25	125	°C	
	校准后的输入偏移 temp=25, VCM=AVDD/2			+2	mV	
	输入平均偏移			3.5	μV/°C	
	输出摆幅	0.1		$V_{DD} - 0.1$	V	
	PGA 增益精度	-1		+1	%	
	普通输入模式范围	0		$V_{DD} - 1.5$	V	
	DC 增益	50	80		dB	
	总增益频率	7		8.2	MHz	$V_{DD} = 5V$
	相位容限	50°			°	
	PSRR+	49	90		dB	$V_{DD} = 5V$
	CMRR	69	90		dB	$V_{DD} = 5V$
	Slew Rate+		6.0	7.5	V/μs	$V_{DD}=5V, R_{Load}=1.3K, C_{Load}=100p$

	唤醒时间			20	us	
--	------	--	--	----	----	--

注： Guaranteed by design, not test in production.

8.5 ESD 特性

符号	等级	条件	封装	最大值	单位
V _{ESD}	静电放电 (人体模式)	TA = + 25 °C	TSSOP 20 TSSOP 28 QFN33	7000	V
	静电放电 (机器模式)			300	V
	静电放电 (带电设备模式)			750	V

8.6 EFT 特性

符号	条件	封装	可通过等级	单位
	Fsys			
	HIRC	TSSOP 20 TSSOP 28 QFN33	± 4400	V

8.7 Flash DC 电气特性

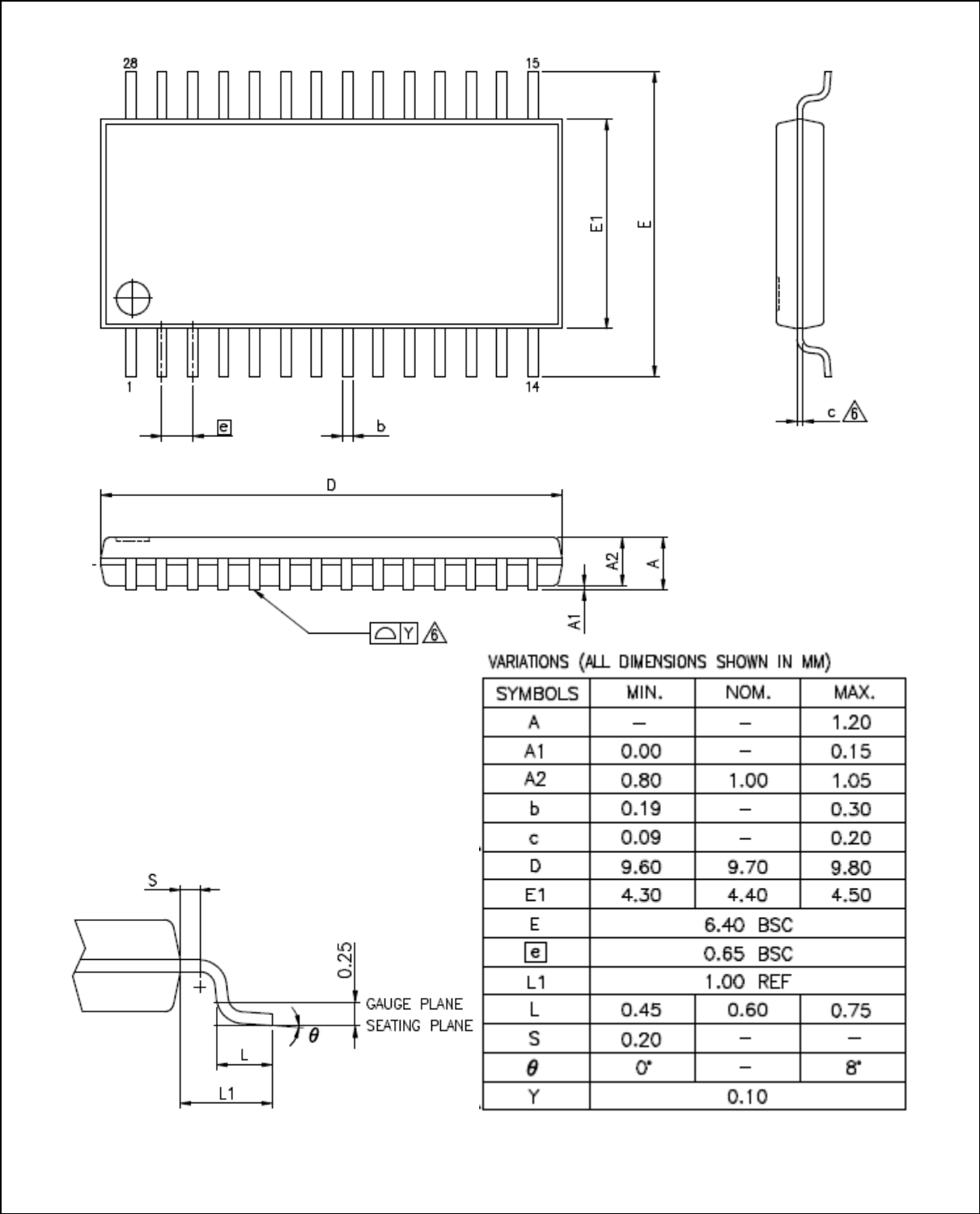
符号	参数	最小值	典型值	最大值	单位	测试条件
$V_{FLA}^{[2]}$	工作电压	1.35	1.5	1.65	V	
N_{ENDUR}	持久度	100,000	-	-	cycles ^[1]	
T_{RET}	数据保留	20	-	-	year	$T_A = 125^{\circ}C$
T_{ERASE}	页擦除时间	-		5	ms	
T_{PROG}	编程时间	-	-	7.5	us	
I_{DD1}	读电流	-	3	4.5	mA	@33 MHz
I_{DD2}	编程电流	-	-	4	mA	
I_{DD3}	擦除电流	-	-	2	mA	

注:

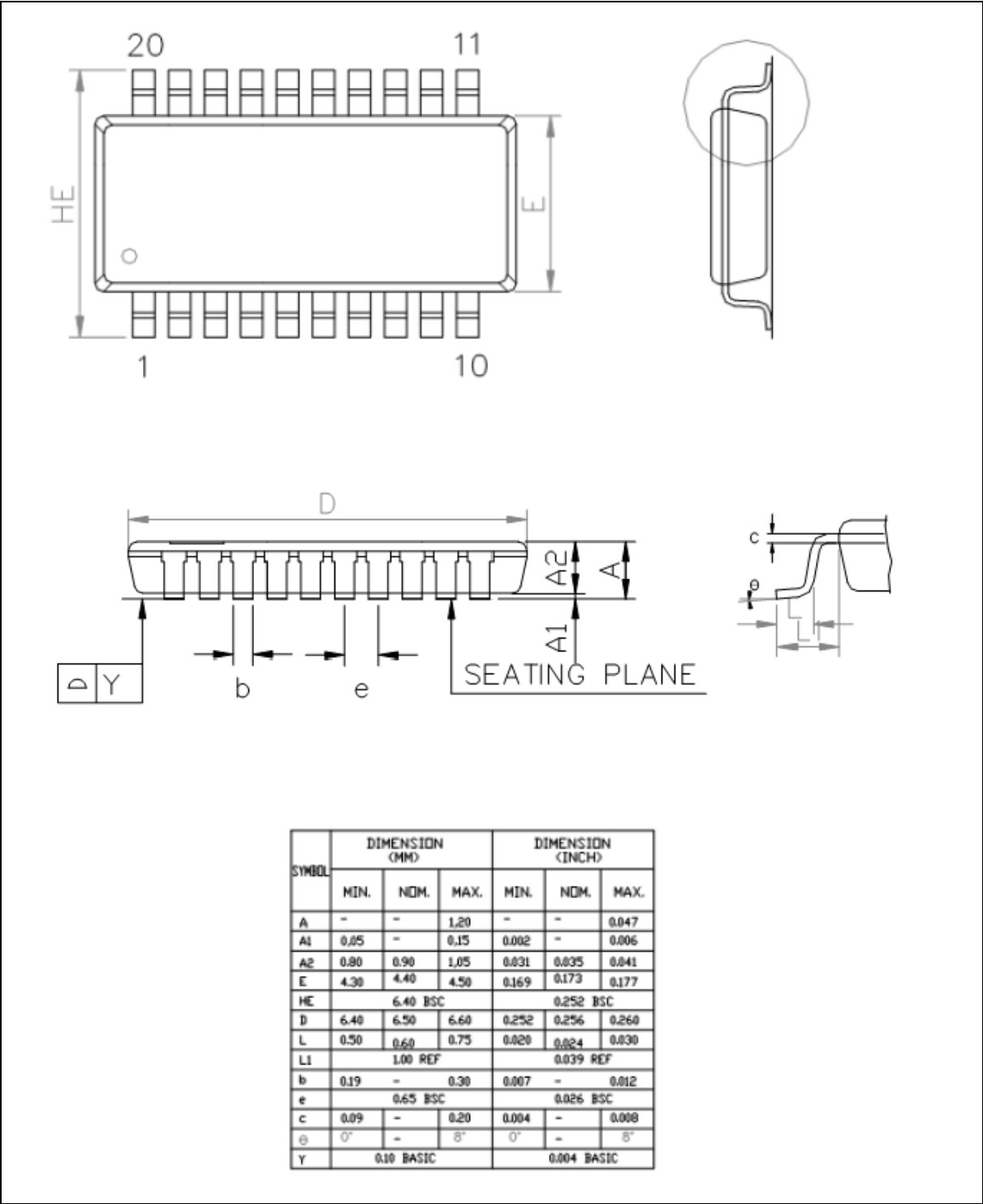
1. 编程/擦除的次数
2. V_{FLA} 是芯片 LDO 的输出电压源
Guaranteed by design, not test in production.

9 封装尺寸

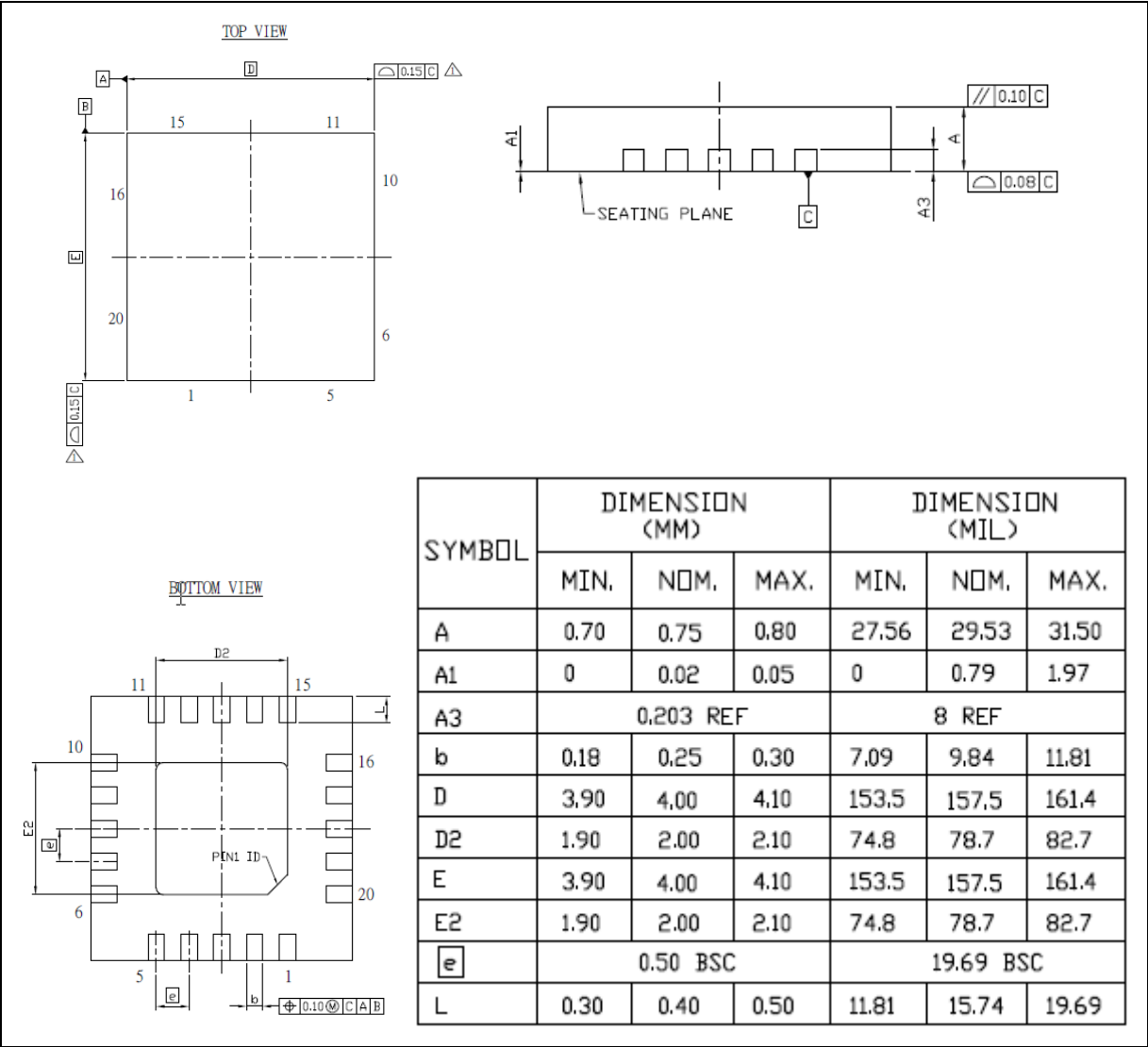
9.1 28-Pin TSSOP (4.4x9.7x1.0 mm)



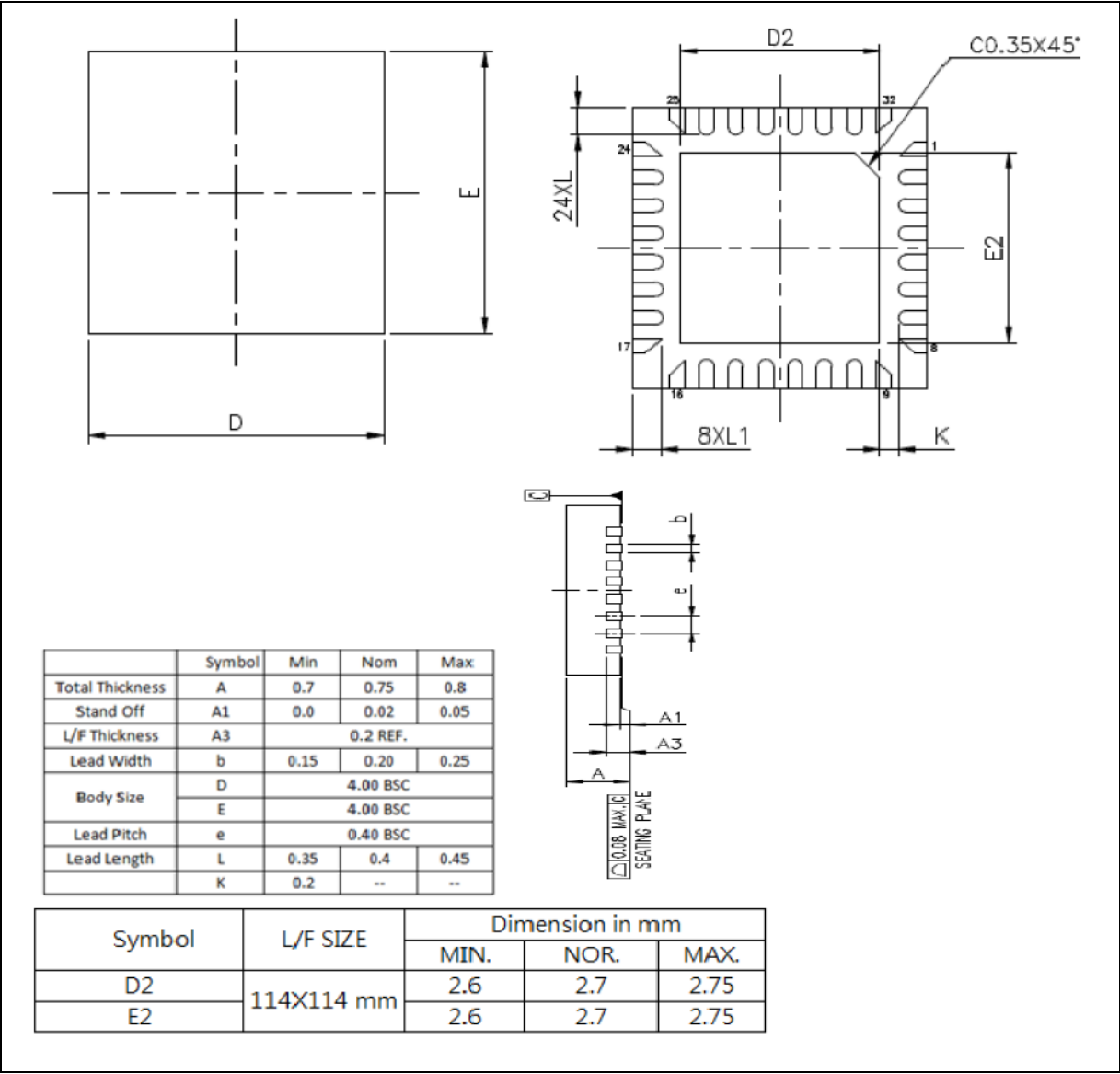
9.2 20-Pin TSSOP (4.4x6.5x0.9 mm)



9.3 20-pin *QFN20 (4 mm x 4 mm)



9.4 33-pin QFN33 (4x4x0.8 mm)



10 修订历史

日期	版本	描述
2017.12.28	1.00	初始版本.

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.*