

简介

CN0816/1616/3217 采用最新技术与灵活的低功耗架构，具有事件系统、高精度模拟特性以及独立于内核的外设（Core Independent Peripheral, CIP）。集成的外设触摸控制器（Peripheral Touch Controller, PTC）支持采用增强型驱动屏蔽和升压模式技术的电容式触摸接口。

特性

- CPU
 - 运行速度最高为 20 MHz
 - 单周期 I/O 访问
 - 两级中断控制器
 - 双周期硬件乘法器
- 存储器
 - 8/16/32 KB 在线自编程闪存
 - 128/256B EEPROM
 - 512B/2 KB SRAM
 - 耐擦写次数：
 - 闪存为 10,000 次
 - EEPROM 为 100,000 次
 - 数据保存时间：
 - 40 年（55°C 使用条件下）
- 系统
 - 上电复位（Power-on Reset, POR）
 - 欠压检测器（Brown-out Detector, BOD）
 - 时钟选项：
 - 16/20 MHz 低功耗内部 RC 振荡器
 - 32.768 kHz 超低功耗（Ultra Low-Power, ULP）内部 RC 振荡器
 - 32.768 kHz 外部晶振
 - 外部时钟输入
 - 单引脚统一编程和调试接口（Unified Program and Debug Interface, UPDI）
 - 三种休眠模式：
 - 空闲模式：所有外设均运行，可立即唤醒
 - 待机模式：
 - 可配置所选外设运行与否
 - 掉电模式：可保存全部数据
- 外设
 - 一个 16 位 A 型定时器/计数器（Timer/Counter Type A, TCA），具有专用周期寄存器和三个比较通道

- 最多两个 16 位 B 型定时器/计数器 (Timer/Counters type B, TCB)，具有输入捕捉功能
- 一个 12 位 D 型定时器/计数器 (Timer/Counter type D, TCD)，针对控制应用进行了优化
- 一个 16 位实时计数器 (Real-Time Counter, RTC)，由外部晶振、外部时钟或内部 RC 振荡器驱动运行
- 支持窗口模式的看门狗定时器 (Watchdog Timer, WDT)，具有独立的片上振荡器
- 一个 USART，具有小数波特率发生器、自动波特率和帧起始检测功能
- 一个主/从串行外设接口 (Serial Peripheral Interface, SPI)
- 一个双线接口 (Two-Wire Interface, TWI)，支持双地址匹配
 - 兼容 Philips I²C
 - 标准模式 (Standard mode, Sm, 100 kHz)
 - 快速模式 (Fast mode, Fm, 400 kHz)
 - 增强型快速模式 (Fast mode plus, Fm+, 1 MHz)
- 最多三个模拟比较器 (Analog Comparator, AC)，具有低传播延时
- 最多两个 10 位 115 ksp/s 模数转换器 (Analog-to-Digital Converter, ADC)
- 最多三个 8 位数模转换器 (Digital-to-Analog Converter, DAC)，每个 DAC 各有一个外部输出
- 多个参考电压 (V_{REF})：
 - 0.55V
 - 1.1V
 - 1.5V
 - 2.5V
 - 4.3V
- 事件系统 (EVSYS)，独立于 CPU 在外设间进行可预测的信号传输
- 可配置定制逻辑 (Configurable Custom Logic, CCL)，具有两个可编程查找表
- 自动 CRC 存储器扫描
- 外设触摸控制器 (PTC) ⁽¹⁾
 - 电容式触摸按钮、滑动条、滚轮和 2D 表面
 - 触摸时唤醒
 - 驱动屏蔽，可提高防潮和抗噪性能
 - 最多 14 个自电容通道
 - 最多 49 个互电容通道
- 所有通用引脚上均具有外部中断
- I/O 和封装：
 - 最多 22 条可编程 I/O 线
 - 20 引脚 VQFN 3x3 mm
 - 24 引脚 VQFN 4x4 mm
- 速度等级：
 - 0-5 MHz @ 1.8V - 5.5V
 - 0-10 MHz @ 2.7V - 5.5V
 - 0-20 MHz @ 4.5V - 5.5V

目录

简介.....	1
特性.....	1
1. CN0816/1616/3217 概述.....	5
2. 框图.....	7
3. 引脚分配.....	8
4. I/O 复用和注意事项.....	10
5. 存储器.....	11
6. 外设和架构.....	32
7. CPU.....	37
8. NVMCTRL——非易失性存储器控制器.....	47
9. CLKCTRL——时钟控制器.....	62
10. SLPCTRL——休眠控制器.....	77
11. RSTCTRL——复位控制器.....	83
12. CPUINT——CPU 中断控制器.....	91
13. EVSYS——事件系统.....	102
14. PORTMUX——端口多路开关.....	113
15. PORT——I/O 引脚配置.....	119
16. BOD——欠压检测器.....	141
17. VREF——参考电压.....	151
18. WDT——看门狗定时器.....	157
19. TCA——A 型 16 位定时器/计数器.....	165
20. TCB——16 位 B 型定时器/计数器.....	213
21. TCD——12 位 D 型定时器/计数器.....	234
22. RTC——实时计数器.....	283
23. USART——通用同步/异步收发器.....	304
24. SPI——串行外设接口.....	337
25. TWI——双线接口.....	352
26. CRCSCAN——循环冗余校验存储器扫描.....	382

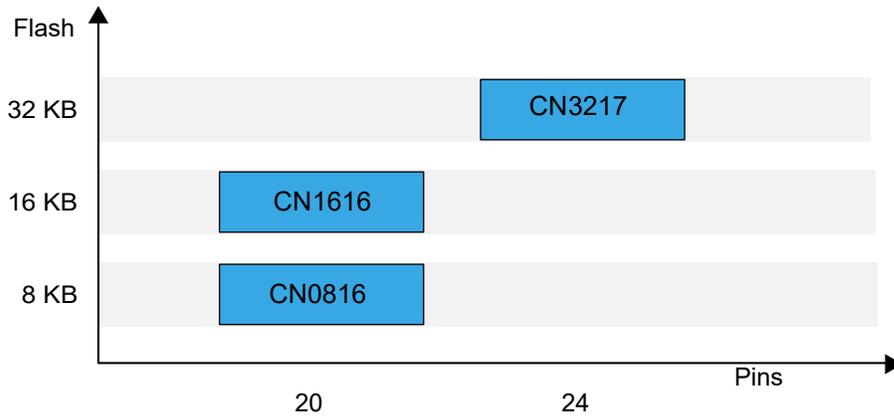
27. CCL——可配置定制逻辑.....	390
28. AC——模拟比较器.....	407
29. ADC——模数转换器.....	416
30. DAC——数模转换器.....	443
31. PTC——外设触摸控制器.....	449
32. UPDI——统一编程和调试接口.....	453
33. 约定.....	486
34. 电气特性.....	490
35. 散热注意事项.....	509
36. 产品信息.....	510
37. 封装图.....	511
产品标识体系.....	517
制造商信息.....	518

1. CN0816/1616/3217 概述

下图所示为本文档中的器件，注明了不同的引脚数与存储器大小：

- 垂直向上移植无需修改代码，因为这些器件的引脚彼此兼容，可提供相同甚至更多的功能。而向下移植可能需要修改代码，因为某些外设的可用数量更少。
- 水平向左移植会使引脚数变少，因此可用的功能也会减少

图 1-1. CN0816/1616/3217 概述



具有不同闪存大小的器件通常也具有不同的 SRAM 和 EEPROM 大小。

1.1. 配置汇总

1.1.1. 外设汇总

表 1-1. 外设汇总

	CN0816	CN1616	CN3217
引脚数	20	20	24
SRAM	512B	2 KB	2 KB
闪存	8 KB	16 KB	32 KB
EEPROM	128B	256B	256B
最大频率 (MHz)	20	20	20
16 位 A 型定时器/计数器 (TCA)	1	1	1
16 位 B 型定时器/计数器 (TCB)	1	2	2
12 位 D 型定时器/计数器 (TCD)	1	1	1
实时计数器 (RTC)	1	1	1

表 1-1. 外设汇总 (续)

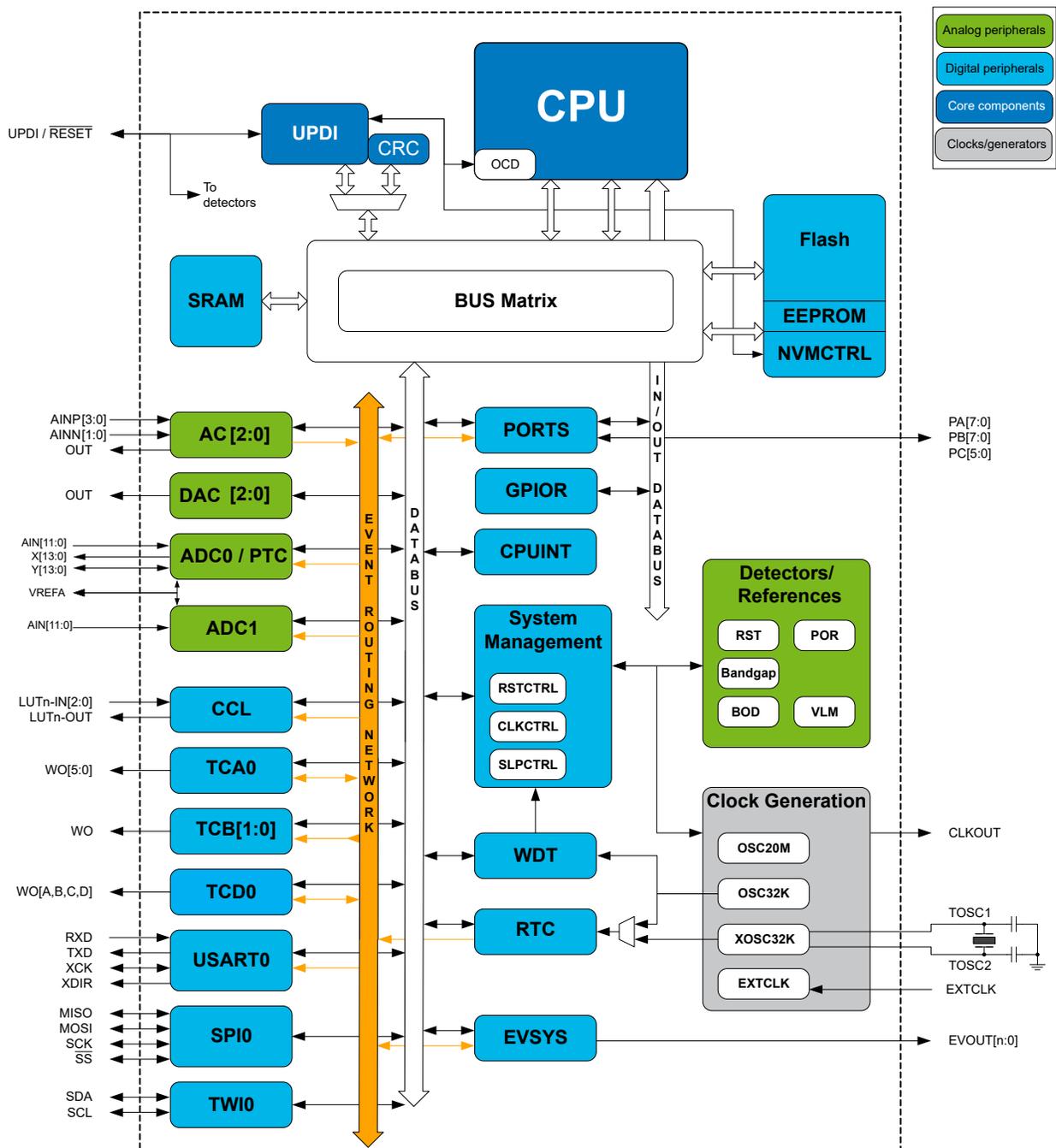
	CN0816	CN1616	CN3217
USART	1	1	1
SPI	1	1	1
TWI (I ² C)	1	1	1
ADC	1	2	2
ADC 通道数	12	12+8	12+12
DAC	1	3	3
AC	1	3	3
AC 输入	2p/2n	3p/2n+ 4p/1n+ 3p/2n (6p/3n)	4p/2n+ 4p/2n+ 4p/2n
外设触摸控制器 (PTC) ⁽¹⁾	1	1	1
PTC 自电容通道数	6	12	14
PTC 互电容通道数	9	36	49
可配置定制逻辑	1	1	1
窗口看门狗	1	1	1
事件系统通道数	6	6	6
通用 I/O	18	18	22
外部中断	18	18	22
CRCSCAN	1	1	1

注:

1. 使用 PTC 时由 PTC 控制 ADC0。
2. 对于 CN0816/1616/3217 器件, 如果没有专门支持, 请在开发工具中选择 ATtiny816/1616/3217。

2. 框图

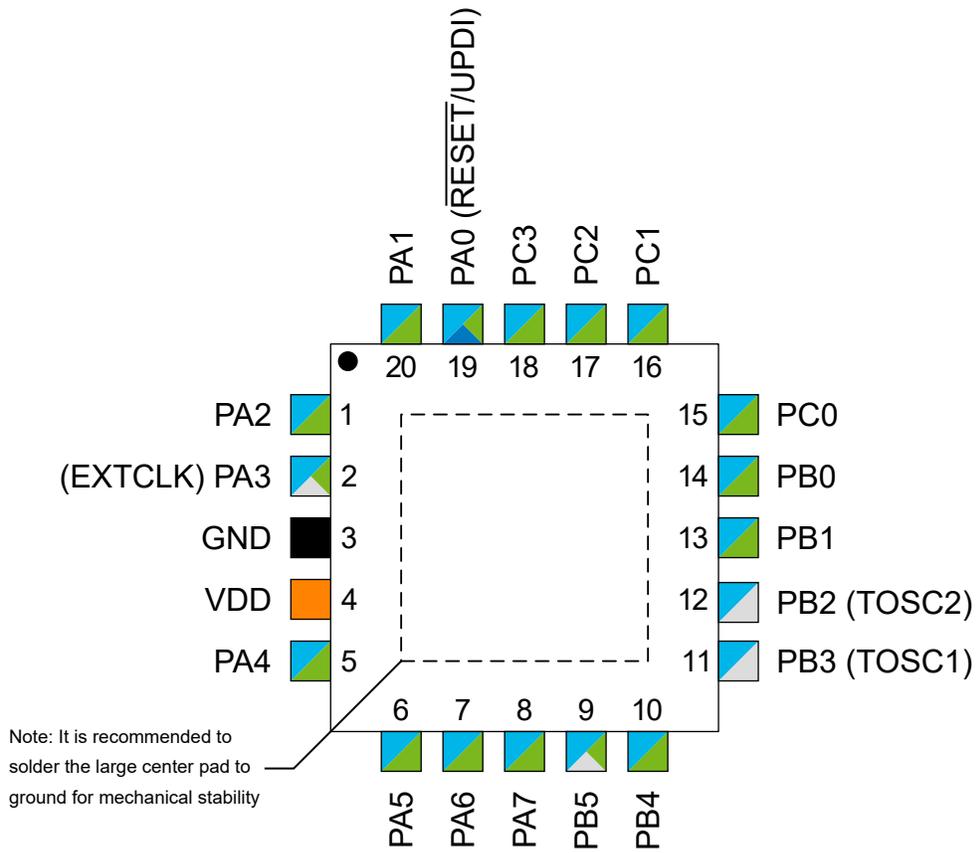
图 2-1. CN0816/1616/3217 框图



注：框图所示为 CN0816/1616/3217 中规格最高的器件（无论是引脚数还是闪存大小）。有关本数据手册中具体器件特性的概述，请参见配置汇总和 I/O 复用和注意事项章节。

3. 引脚分配

3.1. 20 引脚 VQFN



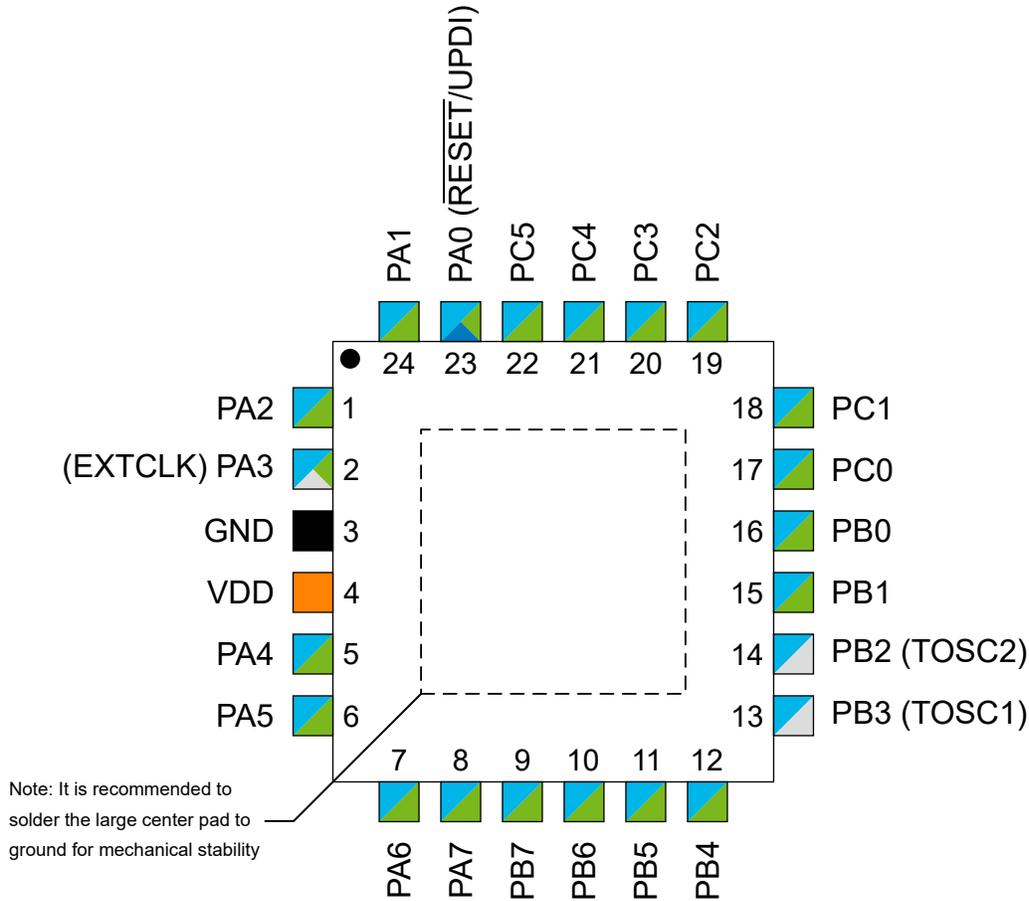
Power

-  Power Supply
-  Ground
-  Pin on VDD Power Domain

Functionality

-  Programming/Debug
-  Clock/Crystal
-  Digital Function Only
-  Analog Function

3.2. 24 引脚 VQFN



Power

-  Power Supply
-  Ground
-  Pin on VDD Power Domain

Functionality

-  Programming/Debug
-  Clock/Crystal
-  Digital Function Only
-  Analog Function

4. I/O 复用和注意事项

4.1. 复用信号

表 4-1. 端口功能复用

24 引脚 VQFN	20 引脚 VQFN	引脚名称 ^(1,2)	其他/专用	ADC0	ADC1 ⁽⁴⁾	PTC ⁽⁵⁾	AC0	AC1 ⁽⁴⁾	AC2 ⁽⁴⁾	DAC0	USART0	SPI0	TWI0	TCA0	TCBn ⁽⁶⁾	TCDO	CCL
23	19	PA0	RESET7 UPDI	AIN0													LUTO- IN0
24	20	PA1		AIN1							TxD ⁽³⁾	MOSI	SDA ⁽³⁾				LUTO- IN1
1	1	PA2	EVOUT0	AIN2							RxD ⁽³⁾	MISO	SCL ⁽³⁾				LUTO- IN2
2	2	PA3	EXTCLK	AIN3							XCK ⁽³⁾	SCK		WO3	TCB1 WO ⁽⁶⁾		
3	3	GND															
4	4	V _{DD}															
5	5	PA4		AIN4	AIN0	X0/Y0					XDIR ⁽³⁾	SS		WO4		WOA	LUTO- OUT
6	6	PA5	VREFA ⁽⁴⁾	AIN5	AIN1	X1/Y1	OUT	AINN0						WO5	TCB0 WO	WOB	
7	7	PA6		AIN6	AIN2	X2/Y2	AINN0	AINP1	AINP0	OUT							
8	8	PA7		AIN7	AIN3	X3/Y3	AINP0	AINP0	AINN0								LUT1- OUT
9		PB7		AIN4				AINN1	AINP3								
10		PB6		AIN5			AINP3 ⁽⁴⁾	AINN1									
11	9	PB5	CLKOUT	AIN8		X12/ Y12 ⁽⁴⁾	AINP1		AINP2					WO2 ⁽³⁾			
12	10	PB4		AIN9		X13/ Y13 ⁽⁴⁾	AINN1	AINP3						WO1 ⁽³⁾			LUTO- OUT ⁽³⁾
13	11	PB3	TOSC1					OUT			RxD			WO0 ⁽³⁾			
14	12	PB2	TOSC2, EVOUT1						OUT		TxD			WO2			
15	13	PB1		AIN10		X4/Y4	AINP2 ⁽⁴⁾				XCK		SDA	WO1			
16	14	PB0		AIN11		X5/Y5	AINP2	AINP1			XDIR		SCL	WO0			
17	15	PC0		AIN6		X6/Y6						SCK ⁽³⁾			TCB0 WO ⁽³⁾	WOC	
18	16	PC1		AIN7		X7/Y7 ⁽⁴⁾						MISO ⁽³⁾				WOD	LUT1- OUT ⁽³⁾
19	17	PC2	EVOUT2	AIN8		X8/Y8 ⁽⁴⁾						MOSI ⁽³⁾					
20	18	PC3		AIN9		X9/Y9 ⁽⁴⁾						SS ⁽³⁾		WO3 ⁽³⁾			LUT1- IN0
21		PC4		AIN10		X10/ Y10 ⁽⁴⁾								WO4 ⁽³⁾	TCB1 WO ^(3,6)		LUT1- IN1
22		PC5		AIN11		X11/ Y11 ⁽⁴⁾								WO5 ⁽³⁾			LUT1- IN2

注:

1. 引脚名称为 Pxn 形式，其中 x 表示端口实例（A 或 B），n 表示引脚编号。信号的符号为 PORTx_PINn。所有引脚均可以用作事件输入。
2. 所有引脚均可用于外部中断，其中每个端口的引脚 Px2 和 Px6 均支持全异步检测。
3. 备用引脚位置。有关备用位置的选择，请参见 [PORTMUX——端口多路开关](#) 一章。
4. CN0816 未提供。
5. 每条 PTC 线路都可以配置为 X 线路或 Y 线路。
6. CN0816 只有 TCB0。

5. 存储器

5.1. 概述

主存储器包括 SRAM 数据存储器、EEPROM 数据存储器和闪存程序存储器。此外，外设寄存器位于 I/O 存储空间。

表 5-1. 闪存的物理属性

属性	CN0816	CN1616	CN3217
大小	8 KB	16 KB	32 KB
页大小	64B	64B	128B
页数	128	256	256
起始地址	0x8000	0x8000	0x8000

表 5-2. SRAM 的物理属性

属性	CN0816	CN1616/CN3217
大小	512B	2 KB
起始地址	0x3E00	0x3800

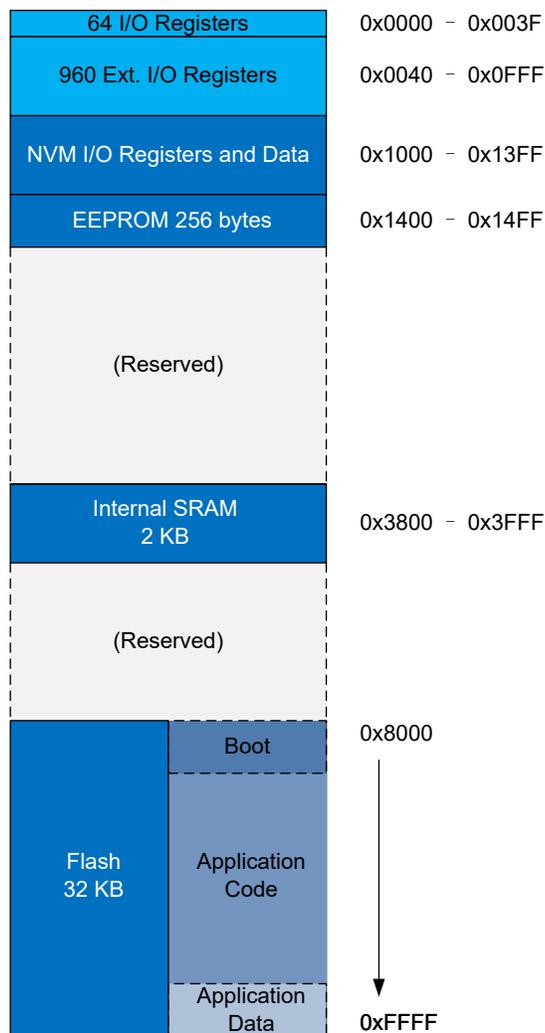
表 5-3. EEPROM 的物理属性

属性	CN0816	CN1616	CN3217
大小	128B	256B	256B
页大小	32B	32B	64B
页数	4	8	4
起始地址	0x1400	0x1400	0x1400

5.2. 存储器映射

下图显示了 CN3217 器件的存储器映射。CN0816/CN1616 器件中各存储段的大小会有所不同。

图 5-1. 存储器映射



5.3. 系统内可重复编程的闪存程序存储器

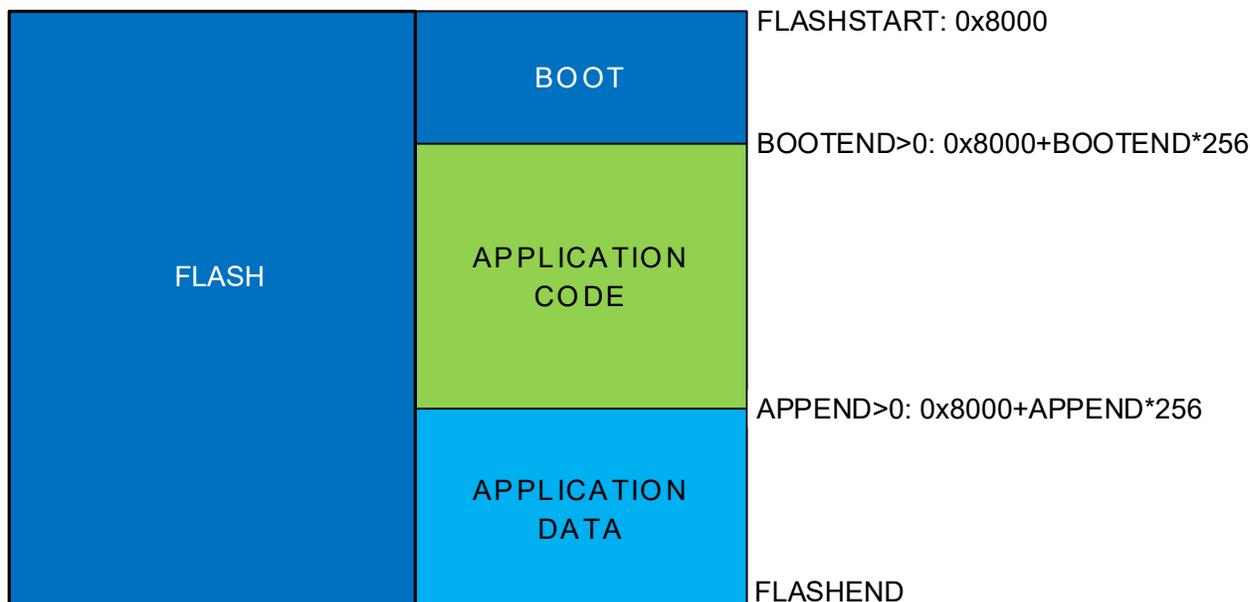
CN0816/1616/3217 包含一个用于存储程序的 8/16/32 KB 片上闪存程序存储器，可在系统内重复编程。由于所有指令均为 16 或 32 位宽，因此闪存采用 16 位数据宽度的存储结构。关于写保护，闪存程序存储空间可划分为三个段（见下图）：自举程序段、应用程序代码段和应用程序数据段，这三个段的访问权限均存在一定的限制。

有关写入闪存步骤的详细信息，请参见非易失性存储器控制器（NVMCTRL）外设文档。

整个闪存映射到存储空间，可通过常规 LD/ST 指令和 LPM 指令访问。对于 LD/ST 指令，闪存的起始地址为 0x8000。对于 LPM 指令，闪存的起始地址为 0x0000。

CN0816/1616/3217 还有一个 CRC 外设，它是总线上的主器件。

图 5-2. 闪存及其三个段



5.4. SRAM 数据存储器

512B/2 KB 的 SRAM 用于数据存储和堆栈。

5.5. EEPROM 数据存储器

CN0816/1616/3217 具有 128/256 字节的 EEPROM 数据存储器。另请参见[存储器映射](#)一节。EEPROM 存储器支持单字节读写。EEPROM 由非易失性存储器控制器（NVMCTRL）控制。

5.6. 用户行

除 EEPROM 外，CN0816/1616/3217 额外有一个可用于存储固件设置的 EEPROM 存储器页——用户行（USERROW）。与正常 EEPROM 一样，该存储器支持单字节读写操作，CPU 可以对该存储器进行读写。如果器件解锁，UPDI 也可以对其进行读写；如果器件锁定，UPDI 就只能进行写操作。USERROW 不受全片擦除的影响。

5.7. 签名字节

CN0816/1616/3217 单片机具有一个 3 字节的签名代码，用于标识器件。这三个字节位于独立的地址空间中。对于此器件，签名字节已在下表中列出。

注：当器件锁定时，只能访问系统信息块（System Information Block，SIB）。

表 5-4. 器件 ID

设备名称	签名字节地址		
	0x00	0x01	0x02
CN0816	0x1E	0x93	0x21
CN1616	0x1E	0x94	0x21
CN3217	0x1E	0x95	0x22

5.8. I/O 存储器

所有 CN0816/1616/3217 I/O 和外设均位于 I/O 存储空间。I/O 地址范围为 0x00 至 0x3F，使用 IN 和 OUT 指令可在单个周期内对该范围进行访问。扩展 I/O 存储空间 0x0040 至 0x0FFF 可通过 LD/LDS/LDD 和 ST/STS/STD 指令进行访问，数据在 32 个通用工作寄存器和 I/O 存储空间之间传输。

可使用 SBI 和 CBI 指令直接访问 0x00 至 0x1F 地址范围内的 I/O 寄存器位。在这些寄存器中，单个位的值可使用 SBIS 和 SBIC 指令来检查。更多详细信息，请参见“指令集”部分。

为了与未来的器件兼容，在访问保留位时必须写入 0。保留的 I/O 空间地址始终不得写入。

一些中断标志可通过向其写入 1 来清零。在 CN0816/1616/3217 器件中，CBI 和 SBI 指令仅适用于指定位，因此可用于包含此类中断标志的寄存器。CBI 和 SBI 指令仅适用于寄存器 0x00 至 0x1F。

通用 I/O 寄存器

CN0816/1616/3217 器件提供四个通用 I/O 寄存器。这些寄存器可用于存储任何信息，特别适合用来存储全局变量和中断标志。可使用 SBI、CBI、SBIS 和 SBIC 指令直接访问位于 0x1C 至 0x1F 地址范围内的通用 I/O 寄存器位。

5.8.1. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	GPIOR0	7:0								GPIOR[7:0]
0x01	GPIOR1	7:0								GPIOR[7:0]
0x02	GPIOR2	7:0								GPIOR[7:0]
0x03	GPIOR3	7:0								GPIOR[7:0]

5.8.2. 寄存器说明

5.8.2.1. 通用 I/O 寄存器 n

名称: GPIORn
 偏移量: $0x00 + n * 0x01$ [n=0..3]
 复位: 0x00
 属性: -

这些寄存器是通用寄存器，可用于将存储数据（如全局变量和标志）存储在可访问位的 I/O 存储空间中。

位	7	6	5	4	3	2	1	0
	GPIOR[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 – GPIOR[7:0] 通用 I/O 寄存器字节

5.9. 器件锁定时 CPU 和 UPDI 对存储段的访问

通过锁定器件，可禁止 UPDI 读取存储器。锁定操作可以保护闪存（所有引导段、应用程序代码段和应用程序数据段）、SRAM 以及包含 FUSE 数据的 EEPROM。这样就无法通过调试器接口读取应用程序数据或代码，但应用程序内部的常规存储器访问不受影响。

向 FUSE.LOCKBIT 中的 LOCKBIT 位域写入无效密钥即可锁定器件。

表 5-5. 存储器访问解锁（FUSE.LOCKBIT 有效密钥）⁽¹⁾

存储段	CPU 访问		UPDI 访问	
	读	写	读	写
SRAM	支持	支持	支持	支持
寄存器	支持	支持	支持	支持
闪存	支持	支持	支持	支持
EEPROM	支持	支持	支持	支持
USERROW	支持	支持	支持	支持
SIGROW	支持	不支持	支持	不支持
其他熔丝	支持	不支持	支持	支持

表 5-6. 存储器访问锁定（FUSE.LOCKBIT 无效密钥）⁽¹⁾

存储段	CPU 访问		UPDI 访问	
	读	写	读	写
SRAM	支持	支持	不支持	不支持
寄存器	支持	支持	不支持	不支持
闪存	支持	支持	不支持	不支持
EEPROM	支持	支持	不支持	不支持
USERROW	支持	支持	不支持	支持 ⁽²⁾
SIGROW	支持	不支持	不支持	不支持
其他熔丝	支持	不支持	不支持	不支持

注:

1. 表中标记为不支持的读操作在执行时可能看似成功，但实际上数据无效。因此，尝试通过 UPDI 对这些存储段进行代码验证时将以失败告终。
2. 在锁定模式下，可以使用 Fuse Write 命令写入 USERROW，但无法读出当前的 USERROW 值。



重要：解锁器件的惟一方式是通过 CHIPERASE。不保留任何应用程序数据。

5.10. 配置和用户熔丝（FUSE）

熔丝是非易失性存储器的一部分，用于保存器件配置。熔丝在器件上电后即可使用。通过 CPU 或 UPDI 可以读取熔丝，但只能通过 UPDI 编程或清零熔丝。存储在熔丝中的配置值会在启动序列结束时写入各自的目标寄存器。

外设配置熔丝（FUSE）已经过预编程，但用户可以进行更改。配置熔丝中发生更改的值需在复位后才能生效。

注：写入熔丝时，必须向所有保留位中写入 1。

5.10.1. 签名行汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	DEVICEID0	7:0								DEVICEID[7:0]
0x01	DEVICEID1	7:0								DEVICEID[7:0]
0x02	DEVICEID2	7:0								DEVICEID[7:0]
0x03	SERNUM0	7:0								SERNUM[7:0]
0x04	SERNUM1	7:0								SERNUM[7:0]
0x05	SERNUM2	7:0								SERNUM[7:0]
0x06	SERNUM3	7:0								SERNUM[7:0]
0x07	SERNUM4	7:0								SERNUM[7:0]
0x08	SERNUM5	7:0								SERNUM[7:0]
0x09	SERNUM6	7:0								SERNUM[7:0]
0x0A	SERNUM7	7:0								SERNUM[7:0]
0x0B	SERNUM8	7:0								SERNUM[7:0]
0x0C	SERNUM9	7:0								SERNUM[7:0]
0x0D										
...	保留									
0x1F										
0x20	TEMPSENSE0	7:0								TEMPSENSE[7:0]
0x21	TEMPSENSE1	7:0								TEMPSENSE[7:0]

5.10.2. 签名行说明

5.10.2.1. 器件 ID n

名称: DEVICEIDn
 偏移量: $0x00 + n*0x01$ [n=0..2]
 默认值: [器件 ID]
 属性: -

每个器件都有一个用于标识器件及其属性（例如，存储器大小、引脚数和管芯版本）的器件 ID。该 ID 可用于标识器件以及可通过软件实现的功能。器件 ID 由三个字节组成：SIGROW.DEVICEID[2:0]。

位	7	6	5	4	3	2	1	0
	DEVICEID[7:0]							
访问	R	R	R	R	R	R	R	R
默认值	x	x	x	x	x	x	x	x

Bit 7:0 - DEVICEID[7:0] 器件 ID 的字节 n

5.10.2.2. 序列号字节 n

名称: SERNUMn
 偏移量: $0x03 + n*0x01$ [n=0..9]
 默认值: [器件序列号]
 属性: -

每个器件都有一个单独的序列号，表示唯一 ID。该 ID 可用于在现场标识特定器件。序列号由十个字节组成：SIGROW.SERNUM[9:0]。

位	7	6	5	4	3	2	1	0
	SERNUM[7:0]							
访问	R	R	R	R	R	R	R	R
默认值	x	x	x	x	x	x	x	x

Bit 7:0 - SERNUM[7:0] 序列号字节 n

5.10.2.3. 温度传感器校准 n

名称: TEMPSENSEn
 偏移量: $0x20 + n*0x01$ [n=0..1]
 默认值: [温度传感器校准值]
 属性: -

温度传感器校准寄存器包含校正因子，用于片上传感器的温度测量。ADC.SIGROW.TEMPSENSE0 为增益/斜率（无符号）的校正因子，SIGROW.TEMPSENSE1 为失调（有符号）的校正因子。

位	7	6	5	4	3	2	1	0
	TEMPSENSE[7:0]							
访问	R	R	R	R	R	R	R	R
默认值	x	x	x	x	x	x	x	x

Bit 7:0 – TEMPSENSE[7:0] 温度传感器校准字节 n

有关如何使用该寄存器的说明，请参见 ADC 一章。

5.10.3. 熔丝汇总——FUSE

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	WDTCFG	7:0	WINDOW[3:0]				PERIOD[3:0]			
0x01	BODCFG	7:0	LVL[2:0]			SAMPFREQ	ACTIVE[1:0]		SLEEP[1:0]	
0x02	OSCCFG	7:0	OSCLOCK						FREQSEL[1:0]	
0x03	保留									
0x04	TCD0CFG	7:0	CMPDEN	CMPDEN	CMPDEN	CMPDEN	CMPDEN	CMPDEN	CMPDEN	CMPDEN
0x05	SYSCFG0	7:0	CRCSRC[1:0]				RSTPINCFG[1:0]			EESAVE
0x06	SYSCFG1	7:0						SUT[2:0]		
0x07	APPEND	7:0	APPEND[7:0]							
0x08	BOOTEND	7:0	BOOTEND[7:0]							
0x09	保留									
0x0A	LOCKBIT	7:0	LOCKBIT[7:0]							

5.10.4. 熔丝说明

5.10.4.1. 看门狗配置

名称: WDTCFG
 偏移量: 0x00
 默认值: 0x00
 属性: -

本熔丝说明中给出的默认值是出厂编程值，不要误认为是复位值。

位	7	6	5	4	3	2	1	0
	WINDOW[3:0]				PERIOD[3:0]			
访问	R	R	R	R	R	R	R	R
默认值	0	0	0	0	0	0	0	0

Bit 7:4 - WINDOW[3:0] 看门狗窗口超时周期

在复位期间，该值装入看门狗控制 A (WDT.CTRLA) 寄存器的 WINDOW 位域。

Bit 3:0 - PERIOD[3:0] 看门狗超时周期

在复位期间，该值装入看门狗控制 A (WDT.CTRLA) 寄存器的 PERIOD 位域。

5.10.4.2. BOD 配置

名称: BODCFG
 偏移量: 0x01
 默认值: 0x00
 属性: -

本熔丝说明中给出的默认值是出厂编程值，不要误认为是复位值。

该熔丝寄存器的位值在启动时写入相应的 BOD 配置寄存器。

位	7	6	5	4	3	2	1	0
	LVL[2:0]			SAMPFREQ	ACTIVE[1:0]		SLEEP[1:0]	
访问	R	R	R	R	R	R	R	R
默认值	0	0	0	0	0	0	0	0

Bit 7:5 - LVL[2:0] BOD 电压

在复位期间，该值被装入 BOD 控制 B (BOD.CTRLB) 寄存器的 LVL 位域。

值	名称	说明
0x0	BODLEVEL0	1.8V
0x2	BODLEVEL2	2.6V
0x7	BODLEVEL7	4.2V

注:

- 说明中的值为典型值
- 有关最大值和最小值的信息，请参见 *电气特性* 中的 *BOD 和 POR 特性*

Bit 4 - SAMPFREQ BOD 采样频率

在复位期间，该值装入 BOD 控制 A (BOD.CTRLA) 寄存器的 SAMPFREQ 位。

值	说明
0x0	采样频率为 1 kHz
0x1	采样频率为 125 Hz

Bit 3:2 - ACTIVE[1:0] 工作和空闲状态下的 BOD 工作模式

在复位期间，该值装入 BOD 控制 A (BOD.CTRLA) 寄存器的 ACTIVE 位域。

值	说明
0x0	禁止
0x1	使能
0x2	采样
0x3	使能，唤醒功能暂停，直至 BOD 就绪

Bit 1:0 - SLEEP[1:0] 休眠状态下的 BOD 工作模式

在复位期间，该值装入 BOD 控制 A (BOD.CTRLA) 寄存器的 SLEEP 位域。

值	说明
0x0	禁止
0x1	使能
0x2	采样
0x3	保留

5.10.4.3. 振荡器配置

名称: OSCCFG
 偏移量: 0x02
 默认值: 0x02
 属性: -

本熔丝说明中给出的默认值是出厂编程值，不要误认为是复位值。

位	7	6	5	4	3	2	1	0
	OSCLOCK						FREQSEL[1:0]	
访问	R						R	R
默认值	0						1	0

Bit 7 - OSCLOCK 振荡器锁定

在复位期间，该熔丝位装载到 CLKCTRL.OSC20MCALIBB 中的 LOCK。

值	说明
0	可访问 OSC20M 振荡器的校准寄存器
1	锁定 OSC20M 振荡器的校准寄存器

Bit 1:0 - FREQSEL[1:0] 频率选择

该位域用于选择 16/20 MHz 内部振荡器（OSC20M）的工作频率，并确定写入 CLKCTRL.OSC20MCALIBA 中的 CAL20M 和 CLKCTRL.OSC20MCALIBB 中的 TEMPCAL20M 的出厂校准值。

值	说明
0x1	以 16 MHz 运行并采用相应的出厂校准值
0x2	以 20 MHz 运行并采用相应的出厂校准值
其他	保留

5.10.4.4. D 型定时器/计数器配置

名称: TCD0CFG
 偏移量: 0x04
 默认值: 0x00
 属性: -

本熔丝说明中给出的默认值是出厂编程值，不要误认为是复位值。

该熔丝寄存器的位值在启动时写入 TCD0 的 TCD.FAULTCTRL 寄存器中的相应位。

位	7	6	5	4	3	2	1	0
	CMPDEN	CMPDEN	CMPBEN	CMPAEN	CMPD	CMPC	CMPB	CMPA
访问	R	R	R	R	R	R	R	R
默认值	0	0	0	0	0	0	0	0

Bit 4, 5, 6, 7 - CMPEN 比较 x 使能

值	说明
0	禁止引脚上的比较 x 输出
1	使能引脚上的比较 x 输出

Bit 0, 1, 2, 3 - CMP 比较 x

该位用于选择比较 x 在复位后的默认状态，或者在 FAULTDET 为 1 的条件下进入调试模式时的默认状态。

值	说明
0	比较 x 默认状态为 0
1	比较 x 默认状态为 1

5.10.4.5. 系统配置 0

名称: SYSCFG0
 偏移量: 0x05
 默认值: 0xF6
 属性: -

本熔丝说明中给出的默认值是出厂编程值，不要误认为是复位值。

位	7	6	5	4	3	2	1	0
	CRCSRC[1:0]				RSTPINCFG[1:0]			EESAVE
访问	R	R			R	R		R
默认值	1	1			0	1		0

Bit 7:6 - CRCSRC[1:0] CRC 源

该位域用于控制在复位初始化期间 CRCSCAN 外设将检查闪存的哪个段。

值	名称	说明
0x0	闪存	整个闪存（引导、应用程序代码和应用程序数据）的 CRC
0x1	BOOT	引导段的 CRC
0x2	BOOTAPP	应用程序代码段和引导段的 CRC
0x3	NOCRC	无 CRC

Bit 3:2 - RSTPINCFG[1:0] 复位引脚配置

该位域用于选择复位/UPDI 引脚配置。

值	说明
0x0	GPIO
0x1	UPDI
0x2	RESET
其他	保留

注：将 RESET 引脚配置为 GPIO 时，在 GPIO 主动驱动输出与高电压 UPDI 使能序列启动之间可能存在冲突。为避免这一冲突，GPIO 输出驱动器将在系统复位后禁止 768 个 OSC32K 周期。待这段时间过后，才能为该引脚允许任何中断。

Bit 0 - EESAVE 全片擦除期间保存 EEPROM

注：如果器件被锁定，该位将不起作用，全片擦除期间始终会擦除 EEPROM。

值	说明
0	全片擦除期间擦除 EEPROM
1	全片擦除期间不擦除 EEPROM

5.10.4.6. 系统配置 1

名称: SYSCFG1
 偏移量: 0x06
 默认值: 0x07
 属性: -

本熔丝说明中给出的默认值是出厂编程值，不要误认为是复位值。

位	7	6	5	4	3	2	1	0	
							SUT[2:0]		
访问						R	R	R	
默认值						1	1	1	

Bit 2:0 - SUT[2:0] 启动时间设置

该位域用于选择上电和代码执行之间的启动时间。

值	说明
0x0	0 ms
0x1	1 ms
0x2	2 ms
0x3	4 ms
0x4	8 ms
0x5	16 ms
0x6	32 ms
0x7	64 ms

5.10.4.7. 应用程序代码末尾

名称: APPEND
 偏移量: 0x07
 默认值: 0x00
 属性: -

本熔丝说明中给出的默认值是出厂编程值，不要误认为是复位值。

位	7	6	5	4	3	2	1	0
	APPEND[7:0]							
访问	R	R	R	R	R	R	R	R
默认值	0	0	0	0	0	0	0	0

Bit 7:0 - APPEND[7:0] 应用程序代码段末尾

该位域以 256 字节的块为单位设置应用程序代码段的末尾。应用程序代码段的末尾应设置为 (BOOT 大小) + (应用程序代码大小)。剩余的闪存段将是应用程序数据段。值 0x00 将 $BOOTEND * 256$ 至闪存末尾的闪存段定义为应用程序代码段。当 FUSE.APPEND 和 FUSE.BOOTEND 均为 0x00 时，整个闪存都是 BOOT 段。

5.10.4.8. 引导末尾

名称: BOOTEND
 偏移量: 0x08
 默认值: 0x00
 属性: -

本熔丝说明中给出的默认值是出厂编程值，不要误认为是复位值。

位	7	6	5	4	3	2	1	0
	BOOTEND[7:0]							
访问	R	R	R	R	R	R	R	R
默认值	0	0	0	0	0	0	0	0

Bit 7:0 - BOOTEND[7:0] 引导段末尾

该位域以 256 字节的块为单位设置引导段的末尾。值 0x00 将整个闪存定义为 BOOT 段。当 FUSE.APPEND 和 FUSE.BOOTEND 均为 0x00 时，整个闪存都是 BOOT 段。

5.10.4.9. 锁定位

名称: LOCKBIT
 偏移量: 0x0A
 默认值: 0xC5
 属性: -

本熔丝说明中给出的默认值是出厂编程值，不要误认为是复位值。

位	7	6	5	4	3	2	1	0
	LOCKBIT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
默认值	1	1	0	0	0	1	0	1

Bit 7:0 - LOCKBIT[7:0] 锁定位

当器件被锁定时，UPDI 无法访问系统总线，因此无法读取除系统信息块（SIB）之外的任何数据。

值	说明
0xC5	有效密钥——存储器访问解锁
其他	无效密钥——存储器访问锁定

6. 外设和架构

6.1. 外设地址映射

地址映射表列出了各外设的基址。有关各外设的完整寄存器说明和汇总，请参见相应的章节。

表 6-1. 外设地址映射

基址	名称	说明
0x0000	VPORТА	虚拟端口 A
0x0004	VPORТB	虚拟端口 B
0x0008	VPORТC	虚拟端口 C
0x001C	GPIO	通用 I/O 寄存器
0x0030	CPU	CPU
0x0040	RSTCTRL	复位控制器
0x0050	SLPCTRL	休眠控制器
0x0060	CLKCTRL	时钟控制器
0x0080	BOD	欠压检测器
0x00A0	VREF	参考电压
0x0100	WDT	看门狗定时器
0x0110	CPUINT	中断控制器
0x0120	CRCSCAN	循环冗余校验存储器扫描
0x0140	RTC	实时计数器
0x0180	EVSYS	事件系统
0x01C0	CCL	可配置定制逻辑
0x0200	PORTMUX	端口多路开关
0x0400	PORTA	端口 A 配置
0x0420	PORTB	端口 B 配置
0x0440	PORTC	端口 C 配置
0x0600	ADC0	模数转换器 0/外设触摸控制器
0x0640	ADC1	模数转换器 1 ⁽¹⁾
0x0670	AC0	模拟比较器 0 ⁽²⁾
0x0680	DAC0	数模转换器 0 ⁽²⁾
0x0680	AC0	模拟比较器 0 ⁽¹⁾
0x0688	AC1	模拟比较器 1 ⁽¹⁾
0x0690	AC2	模拟比较器 2 ⁽¹⁾
0x06A0	DAC0	数模转换器 0 ⁽¹⁾
0x06A8	DAC1	数模转换器 1 ⁽¹⁾
0x06B0	DAC2	数模转换器 2 ⁽¹⁾
0x0800	USART0	通用同步/异步收发器 0
0x0810	ТWIO	双线接口 0
0x0820	SPI0	串行外设接口 0
0x0A00	TCA0	A 型定时器/计数器 0
0x0A40	TCB0	B 型定时器/计数器 0
0x0A50	TCB1	B 型定时器/计数器 1 ⁽¹⁾
0x0A80	TCD0	D 型定时器/计数器 0
0x0F00	SYSCFG	系统配置
0x1000	NVMCTRL	非易失性存储器控制器

表 6-1. 外设地址映射（续）

基址	名称	说明
0x1100	SIGROW	签名行
0x1280	FUSES	器件特定熔丝
0x1300	USERROW	用户行

注：

1. 不适用于 CN0816。
2. 仅适用于 CN0816。

6.2. 中断向量映射

每个中断向量都与一个外设实例相连，如下表所示。一个外设可以有一个或多个中断源，有关可用中断源的更多详细信息，请参见相应外设功能说明的中断部分。

发生中断条件时，外设的中断标志（*外设.INTFLAGS*）寄存器中的中断标志（*nameIF*）置 1。

可以通过写入外设的中断控制（*外设.INTCTRL*）寄存器中相应的中断允许（*nameIE*）位来允许或禁止中断。

在一些外设中，寄存器的命名方式可能略有不同。

当相应的中断被允许且中断标志置 1 时，将生成中断请求。中断请求保持有效，直至中断标志清零。有关如何清零中断标志的详细信息，请参见外设的 *INTFLAGS* 寄存器。

必须全局允许中断才能生成中断请求。

表 6-2. 中断向量映射

向量编号	程序地址（字） CN0816	程序地址（字） CN1616/3217	外设源 （名称）	说明
0	0x00	0x00	RESET	
1	0x01	0x02	CRCSCAN_NMI	CRCSCAN 可用的不可屏蔽中断
2	0x02	0x04	BOD_VLM	电压监视器中断
3	0x03	0x06	PORTA_PORT	端口 A 中断
4	0x04	0x08	PORTB_PORT	端口 B 中断
5	0x05	0x0A	PORTC_PORT	端口 C 中断
6	0x06	0x0C	RTC_CNT	实时计数器中断
7	0x07	0x0E	RTC_PIT	周期性中断定时器中断（在 RTC 外设中）
8	0x08	0x10	TCA0_OVF TCA0_LUNF	正常模式：A 型定时器/计数器上溢中断。 拆分模式：A 型定时器/计数器低字节下溢中断。
9	0x09	0x12	TCA0_HUNF	正常模式：未使用。 拆分模式：A 型定时器/计数器高字节下溢中断。
10	0x0A	0x14	TCA0_CMP0 TCA0_LCMP0	正常模式：A 型定时器/计数器比较通道 0 中断。 拆分模式：A 型定时器/计数器低字节比较通道 0 中断。
11	0x0B	0x16	TCA0_CMP1 TCA0_LCMP1	正常模式：A 型定时器/计数器比较通道 1 中断。 拆分模式：A 型定时器/计数器低字节比较通道 1 中断。

表 6-2. 中断向量映射（续）

向量编号	程序地址（字） CN0816	程序地址（字） CN1616/3217	外设源 （名称）	说明
12	0x0C	0x18	TCA0_CMP2 TCA0_LCMP2	正常模式：A 型定时器/计数器比较通道 2 中断。 拆分模式：A 型定时器/计数器低字节比较通道 2 中断。
13	0x0D	0x1A	TCB0_INT	B 型定时器/计数器捕捉中断
14	—	0x1C	TCB1_INT	B 型定时器/计数器捕捉中断 ⁽¹⁾
14	0x0E	—	TCD0_OVF	D 型定时器/计数器上溢中断 ⁽²⁾
15	—	0x1E	TCD0_OVF	D 型定时器/计数器上溢中断
15	0x0F	—	TCD0_TRIG	D 型定时器/计数器触发中断 ⁽²⁾
16	—	0x20	TCD0_TRIG	D 型定时器/计数器触发中断
16	0x10	—	AC0_AC	模拟比较器中断 ⁽²⁾
17	—	0x22	AC0_AC	模拟比较器中断 ⁽¹⁾
17	0x11	—	ADC0_RESRDY	模数转换器结果就绪中断 ⁽²⁾
18	—	0x24	AC1_AC	模拟比较器中断 ⁽¹⁾
18	0x12	—	ADC0_WCOMP	模数转换器窗口比较中断 ⁽²⁾
19	—	0x26	AC2_AC	模拟比较器中断 ⁽¹⁾
19	0x13	—	TWI0_TWIS	双线接口/I ² C 从器件中断 ⁽²⁾
20	—	0x28	ADC0_RESRDY	模数转换器结果就绪中断 ⁽¹⁾
20	0x14	—	TWI0_TWIM	双线接口/I ² C 主器件中断 ⁽²⁾
21	—	0x2A	ADC0_WCOMP	模数转换器窗口比较中断 ⁽¹⁾
21	0x15	—	SPI0_INT	串行外设接口中断 ⁽²⁾
22	—	0x2C	ADC1_RESRDY	模数转换器结果就绪中断 ⁽¹⁾
22	0x16	—	USART0_RXC	通用同步/异步收发器接收完成中断 ⁽²⁾
23	—	0x2E	ADC1_WCOMP	模数转换器窗口比较中断 ⁽¹⁾
23	0x17	—	USART0_DRE	通用同步/异步收发器数据就绪中断 ⁽²⁾
24	—	0x30	TWI0_TWIS	双线接口/I ² C 从器件中断 ⁽¹⁾
24	0x18	—	USART0_TXC	通用同步/异步收发器发送完成中断 ⁽²⁾
25	—	0x32	TWI0_TWIM	双线接口/I ² C 主器件中断 ⁽¹⁾
25	0x19	—	NVMCTRL_EE	非易失性存储器 EEPROM 就绪中断 ⁽²⁾
26	—	0x34	SPI0_INT	串行外设接口中断 ⁽¹⁾
27	—	0x36	USART0_RXC	通用同步/异步收发器接收完成中断 ⁽¹⁾
28	—	0x38	USART0_DRE	通用同步/异步收发器数据就绪中断 ⁽¹⁾
29	—	0x3A	USART0_TXC	通用同步/异步收发器发送完成中断 ⁽¹⁾
30	—	0x3C	NVMCTRL_EE	非易失性存储器 EEPROM 就绪中断 ⁽¹⁾

注：

1. 不适用于 CN0816。
2. 仅适用于 CN0816。

6.3. 系统配置（SYSCFG）

系统配置包含器件的版本 ID。版本 ID 可从 CPU 中读取，方便用户在器件改版时对应用做出调整。

6.3.1. 寄存器汇总

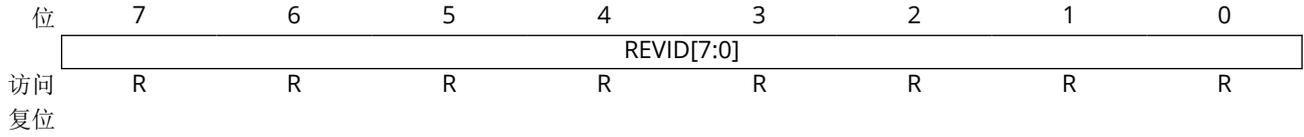
偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	保留									
0x01	REVID	7:0	REVID[7:0]							

6.3.2. 寄存器说明

6.3.2.1. 器件版本 ID 寄存器

名称: REVID
 偏移量: 0x01
 复位: [版本 ID]
 属性: -

该寄存器是只读的，显示器件版本 ID。



Bit 7:0 - REVID[7:0] 版本 ID

该位域包含器件版本。0x00 = A, 0x01 = B, 依此类推。

7. CPU

7.1. 特性

- 8 位高性能 RISC CPU:
 - 135 条指令
 - 硬件乘法器
- 32 个 8 位寄存器，直接连接到算术逻辑单元（Arithmetic Logic Unit, ALU）
- RAM 中的堆栈
- 可在 I/O 存储空间中访问堆栈指针
- 直接寻址最高 64 KB 的统一存储器
- 有效支持 8 位、16 位和 32 位算术运算
- 针对系统关键特性的配置更改保护
- 原生支持片上调试（On-Chip Debugging, OCD）：
 - 两个硬件断点
 - 程序流、中断和软件断点有变化
 - 运行时读取堆栈指针（Stack Pointer, SP）寄存器、程序计数器（Program Counter, PC）和状态寄存器（SREG）
 - 在停止模式下可读写寄存器文件

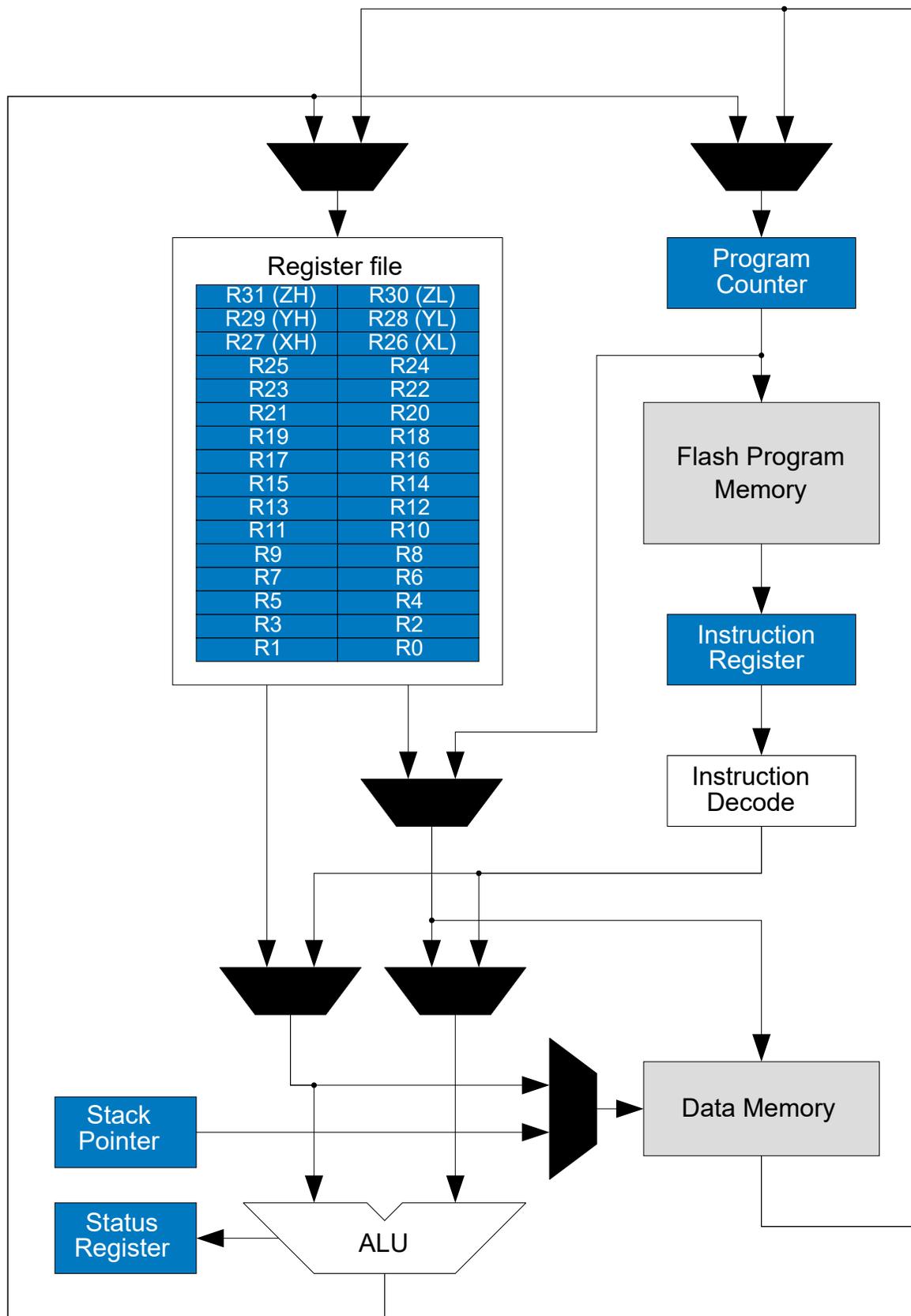
7.2. 概述

CPU 可以访问存储器、执行计算、控制外设、执行程序存储器中的指令，以及处理中断。

7.3. 架构

为了最大程度地发挥性能和并行能力，CPU 采用哈佛架构，为程序和数据提供单独的总线。程序存储器中的指令采用单级流水线的形式执行。在执行一条指令的同时，会从程序存储器中预取出下一条指令。这样可以在每个时钟周期内都执行指令。

图 7-1. CPU 架构



7.4. 算术逻辑单元（ALU）

算术逻辑单元（Arithmetic Logic Unit, ALU）支持在工作寄存器之间或者在常量与工作寄存器之间进行算术和逻辑运算。此外，也可以执行单寄存器运算。

ALU 运算直接与时钟文件中的所有 32 个通用工作寄存器相连。工作寄存器之间或者工作寄存器与立即操作数之间的算术运算在一个时钟周期内执行，结果存储在寄存器文件中。算术或逻辑运算完成后，状态寄存器（CPU.SREG）会进行更新以反映运算结果的相关信息。

ALU 运算分为三个主要类别——算术、逻辑和位函数。ALU 支持 8 位和 16 位算术运算，并且指令集可以有效地实现 32 位算术运算。硬件乘法器支持有符号和无符号的乘法以及小数格式。

7.4.1. 硬件乘法器

乘法器可以将两个 8 位数相乘得到一个 16 位结果。硬件乘法器支持多种形式的有符号和无符号整数和小数：

- 有符号/无符号整数的乘法
- 有符号/无符号小数的乘法
- 有符号整数与无符号整数的乘法
- 有符号小数与无符号小数的乘法

一次乘法运算需要两个 CPU 时钟周期。

7.5. 功能说明

7.5.1. 程序流

复位后，CPU 将从闪存程序存储器中的最低地址 0x0000 开始执行指令。程序计数器（PC）寻址下一条要获取的指令。

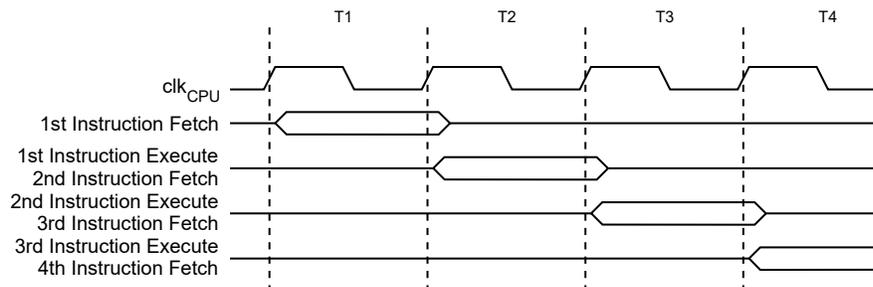
CPU 支持可有条件或无条件地更改程序流并且能够直接寻址整个地址空间的指令。大多数指令使用 16 位字格式，少数使用 32 位格式。

在中断和子程序调用期间，返回地址 PC 会作为字指针存储在堆栈中。堆栈在通用数据 SRAM 中进行分配，因此，堆栈大小仅受 SRAM 总大小及其使用率的限制。堆栈指针（SP）复位后指向内部 SRAM 的最高地址。可在 I/O 存储空间中对 SP 进行读/写访问，从而轻松实现多个堆栈或堆栈区域。通过 CPU 支持的五种不同寻址模式，可以轻松访问数据 SRAM。

7.5.2. 指令执行时序

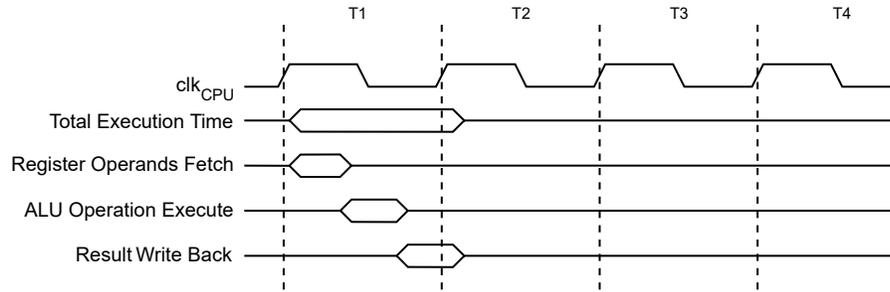
CPU 的时钟由 CPU 时钟 CLK_CPU 提供，不采用内部时钟分频。下图显示了通过哈佛架构和快速访问寄存器文件概念实现的并行取指和执行。这是一种基本的流水线概念，可实现最高 1 MIPS/MHz 的高性能。

图 7-2. 并行取指和执行



下图给出了寄存器文件的内部时序概念。在一个时钟周期内，系统执行了使用两个寄存器操作数的 ALU 操作，并将结果存储在目标寄存器中。

图 7-3. 单个周期的 ALU 操作



7.5.3. 状态寄存器

状态寄存器（CPU.SREG）中包含最近执行的算术或逻辑指令的结果的相关信息。该信息可用于更改程序流以执行条件运算。

CPU.SREG 在所有 ALU 运算后更新，如 *指令集汇总* 一章中所述，在许多情况下，这将无需使用专用比较指令，从而可实现更快速、更紧凑的代码。进入中断服务程序（Interrupt Service Routine, ISR）或从该程序返回时，CPU.SREG 不会自动存储或恢复。因此，必须通过用户定义的软件在现场切换时保持状态寄存器。CPU.SREG 可以在 I/O 存储空间中访问。

7.5.4. 堆栈和堆栈指针

堆栈用于在中断和子程序调用后存储返回地址。此外，也可用于存储临时数据。堆栈指针（SP）始终指向堆栈顶部。SP 指向的地址存储在堆栈指针（CPU.SP）寄存器中。CPU.SP 以两个 8 位寄存器的形式实现，这两个寄存器在 I/O 存储空间中可以访问。

使用表 7-1 中给出的指令或者执行中断可以将数据压入堆栈或从堆栈中弹出。堆栈从高存储单元向低存储单元增长。这意味着将数据压入堆栈时，SP 将会减小；而从堆栈中弹出数据时，SP 将会增加。SP 在复位后将自动设置为内部 SRAM 的最高地址。如果更改了堆栈，则必须将 SP 设置为指向 SRAM 起始地址以上的位置（有关 SRAM 起始地址，请参见 *存储器* 一章中的 *SRAM 数据存储* 主题），并且必须在执行任何子程序调用前及允许中断前对其进行定义。有关 SP 的详细信息，请参见下表。

表 7-1. 堆栈指针指令

指令	堆栈指针	说明
PUSH	递减 1	数据压入堆栈
CALL ICALL RCALL	递减 2	返回地址通过子程序调用或中断压入堆栈
POP	递增 1	数据从堆栈中弹出
RET RETI	递增 2	返回地址通过从子程序返回或从中断返回而从堆栈中弹出

在中断或子程序调用期间，返回地址将作为一个字被自动压入堆栈，SP 递减 2。返回地址由两个字节组成，最低有效字节（Least Significant Byte, LSB）先被压入堆栈（较高的地址）。例如，字节指针返回地址 0x0006 在堆栈中保存为 0x0003（右移 1 位），指向程序存储器中的第四个 16 位指令字。返回地址通过 RETI（从中断返回时）和 RET（从子程序调用返回时）从堆栈中弹出，SP 递增 2。

数据通过 PUSH 指令压入堆栈时，SP 递减 1，通过 POP 指令从堆栈中弹出时，SP 递增 1。

为了防止从软件更新 SP 时发生数据损坏，写入 SPL 将自动禁止中断，持续时间最长为四条指令或直到下一次 I/O 存储器写操作（以先出现者为准）。

7.5.5. 寄存器文件

寄存器文件由 CPU 使用的 32 个 8 位通用工作寄存器组成。寄存器文件与数据存储器位于不同的地址空间中。

所有对工作寄存器进行操作的 CPU 指令都可以直接对寄存器文件进行单周期访问。某些指令可访问的工作寄存器会受到一定限制，例如常量算术和逻辑指令 SBCI、SUBI、CPI、ANDI、ORI 和 LDI。这些指令适用于寄存器文件中工作寄存器的后半部分，即 R16 至 R31。

图 7-4. CPU 通用工作寄存器

7	0	Addr.	
R0		0x00	
R1		0x01	
R2		0x02	
...			
R13		0x0D	
R14		0x0E	
R15		0x0F	
R16		0x10	
R17		0x11	
...			
R26		0x1A	X-register Low Byte
R27		0x1B	X-register High Byte
R28		0x1C	Y-register Low Byte
R29		0x1D	Y-register High Byte
R30		0x1E	Z-register Low Byte
R31		0x1F	Z-register High Byte

7.5.5.1. X、Y 和 Z 寄存器

除一般用途外，工作寄存器 R26 至 R31 还有其他功能。

这些寄存器可以形成 16 位地址指针，用于对数据存储器进行间接寻址。这三个地址寄存器称为 X 寄存器、Y 寄存器和 Z 寄存器。Z 寄存器还可用作程序存储器的地址指针。

图 7-5. X、Y 和 Z 寄存器

Bit (individually)	7	R27	0	7	R26	0
X-register	XH		XL			
Bit (X-register)	15		8	7		0
Bit (individually)	7	R29	0	7	R28	0
Y-register	YH		YL			
Bit (Y-register)	15		8	7		0
Bit (individually)	7	R31	0	7	R30	0
Z-register	ZH		ZL			
Bit (Z-register)	15		8	7		0

最低寄存器地址保存最低有效字节（LSB），最高寄存器地址保存最高有效字节（MSB）。通过不同的 LD*/ST* 指令，这些地址寄存器可按照固定位移、自动递增和自动递减的方式工作。有关详细信息，请参见指令集汇总一章。

7.5.6. 访问 16 位寄存器

CN0816/1616/3217 器件的大多数寄存器都是 8 位寄存器，但也有一些 16 位寄存器。由于数据总线宽度为 8 位，因此访问 16 位需要两次读或写操作。CN0816/1616/3217 器件的所有 16 位寄存器都通过临时（TEMP）寄存器连接到 8 位总线。

对于 16 位写操作，必须先写入 16 位寄存器的低字节寄存器（如 DATAL），然后再写入高字节寄存器（如 DATAH）。写入低字节寄存器时，实际上将写入临时（TEMP）寄存器而非低字节寄存器。写入 16 位寄存器的高字节寄存器时，TEMP 的内容将在同一个时钟周期内复制到 16 位寄存器的低字节中。

对于 16 位读操作，必须先读取 16 位寄存器的低字节寄存器（如 DATAL），然后再读取高字节寄存器（如 DATAH）。读取低字节寄存器时，16 位寄存器的高字节寄存器的内容将在同一个时钟周期内复制到临时（TEMP）寄存器中。读取高字节寄存器时，实际上将读取 TEMP 而非高字节寄存器。

上述机制可确保在读取或写入寄存器时始终同时访问 16 位寄存器的低字节和高字节。

如果在 16 位读/写操作期间触发了中断并在中断服务程序中访问了同一外设内的 16 位寄存器，则中断可能会破坏时间顺序。为了防止这种情况，可以在写入或读取 16 位寄存器时禁止中断。或者，也可以在中断服务程序中的 16 位访问之前读取临时寄存器，并在访问之后恢复临时寄存器。

7.5.6.1. 访问 24 位寄存器

对于 24 位寄存器，除了有两个临时寄存器用于 24 位寄存器外，读写方式与 16 位寄存器相同。写入寄存器时必须最后写入最高有效字节，读取寄存器时必须先读取最低有效字节。

7.5.7. 配置更改保护（CCP）

系统关键 I/O 寄存器设置不受意外修改的影响。闪存自编程（通过存储到 NVM 控制器）不受意外执行的影响。这些都由配置更改保护（CCP）寄存器进行全局处理。

只有在 CPU 向 CCP 寄存器写入签名后，才能对受保护的 I/O 寄存器或位进行更改或执行受保护的指令。CCP 寄存器（CPU.CCP）的说明中列出了不同的签名。

操作模式分为两种：一种用于受保护的 I/O 寄存器，另一种用于受保护的自编程。

7.5.7.1. 对受配置更改保护的 I/O 寄存器的写操作顺序

为了写入受 CCP 保护的寄存器，需要执行以下步骤：

1. 软件将允许更改受保护 I/O 寄存器的签名写入 CPU.CCP 寄存器中的 CCP 位域。
2. 在四条指令内，软件必须将相应的数据写入受保护的寄存器。
大多数受保护的寄存器还包含一个写使能/更改使能/锁定位。该位必须在写入数据的同一操作中写入 1。

如果 CPU 对 I/O 寄存器或数据存储执行写操作、对闪存、NVMCTRL 和 EEPROM 执行装载或存储访问，或者执行 SLEEP 指令，则受保护的更改将被立即禁止。

7.5.7.2. 自编程的执行顺序

为了执行自编程（执行对 NVM 控制器的命令寄存器的写操作），需要执行以下步骤：

1. 软件通过向 CCP 寄存器（CPU.CCP）写入 SPM 签名来临时使能自编程功能。
2. 在四条指令内，软件必须执行相应的指令。如果 CPU 执行对闪存、NVMCTRL 或 EEPROM 的访问，或者执行 SLEEP 指令，则受保护的更改将被立即禁止。

CPU 写入正确的签名后，中断将在配置更改使能期间被忽略。CCP 期间的任何中断请求（包括不可屏蔽中断）都会将相应的中断标志正常置 1，并且请求将保持待处理状态。CCP 结束后，所有待处理的中断都将根据其级别和优先级执行。

7.6. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	保留									
...										
0x03										
0x04	CCP	7:0	CCP[7:0]							
0x05	保留									
...										
0x0C										
0x0D	SP	7:0	SP[7:0]							
		15:8	SP[15:8]							
0x0F	SREG	7:0	I	T	H	S	V	N	Z	C

7.7. 寄存器说明

7.7.1. 配置更改保护

名称: CCP
偏移量: 0x04
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	CCP[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - CCP[7:0] 配置更改保护

向此位域写入正确的签名即可更改受保护的 I/O 寄存器，或者可在执行的后续四条 CPU 指令中执行受保护的指令。

在这些周期内将忽略所有中断。这些周期完成之后，CPU 将自动处理中断，并将根据其级别和优先级执行所有待处理中断。

写入受保护的 I/O 寄存器签名时，只要使能 CCP 功能，CCP[0]就将读为 1。

写入受保护的自编程签名时，只要使能 CCP 功能，CCP[1]就将读为 1。

CCP[7:2]将始终读为 0。

值	名称	说明
0x9D	SPM	允许自编程
0xD8	IOREG	解锁受保护的 I/O 寄存器

7.7.2. 堆栈指针

名称: SP
偏移量: 0x0D
复位: 栈顶
属性: -

CPU.SP 寄存器保存指向栈顶的堆栈指针 (SP)。复位后, SP 指向最高的内部 SRAM 地址。

对于每个器件, 仅实现寻址可用数据存储 (包括外部存储器, 最大 64 KB) 所需的位数。未使用的位将始终读为 0。

CPU.SPL 和 CPU.SPH 寄存器代表 16 位值 CPU.SP。低字节[7:0] (后缀 L) 可通过原始偏移地址来访问。高字节[15:8] (后缀 H) 可通过偏移地址 + 0x01 来访问。

为了防止从软件更新 SP 时发生数据损坏, 写入 CPU.SPL 将自动禁止中断, 持续时间为接下来的四条指令或直到下一次 I/O 存储器写操作 (以先出现者为准)。

位	15	14	13	12	11	10	9	8
	SP[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-							
位	7	6	5	4	3	2	1	0
	SP[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	-							

Bit 15:8 - SP[15:8] 堆栈指针高字节
这些位保存 16 位寄存器的 MSB。

Bit 7:0 - SP[7:0] 堆栈指针低字节
这些位保存 16 位寄存器的 LSB。

7.7.3. 状态寄存器

名称: SREG
 偏移量: 0x0F
 复位: 0x00
 属性: -

状态寄存器包含最近执行的算术或逻辑指令结果的相关信息。有关该寄存器中的位以及它们如何受不同指令影响的详细信息，请参见指令集汇总部分。

位	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
访问	R/W							
复位	0	0	0	0	0	0	0	0

Bit 7 - I 全局中断允许位

向该位写入 1 可允许器件的中断。

向该位写入 0 可禁止器件的中断，它不受外设的中断允许设置的影响。

该位既不会在进入中断服务程序（ISR）时由硬件清零，也不会执行 RETI 指令时置 1。

该位可通过软件使用 SEI 和 CLI 指令置 1 和清零。

通过 I/O 寄存器更改 I 位会导致访问时出现一个周期的等待状态。

Bit 6 - T 传输位

位装入（BLD）和位存储（BST）这两个位复制指令使用 T 位作为操作位的源或目标。

Bit 5 - H 半进位标志

该标志在支持半进位的算术运算中发生半进位时置 1，未发生半进位时清零。半进位在 BCD 算术中很有用。

Bit 4 - S 符号标志

该标志始终为负标志（N）与二进制补码溢出标志（V）之间的异或（XOR）运算结果。

Bit 3 - V 二进制补码溢出标志

该标志在支持溢出的算术运算中发生溢出时置 1，未发生溢出时清零。

Bit 2 - N 负标志

该标志在算术或逻辑运算结果为负时置 1，不为负时清零。

Bit 1 - Z 零标志

该标志在算术或逻辑运算结果为零时置 1，不为零时清零。

Bit 0 - C 进位标志

该标志在算术或逻辑运算中发生进位时置 1，未发生进位时清零。

8. NVMCTRL——非易失性存储器控制器

8.1. 特性

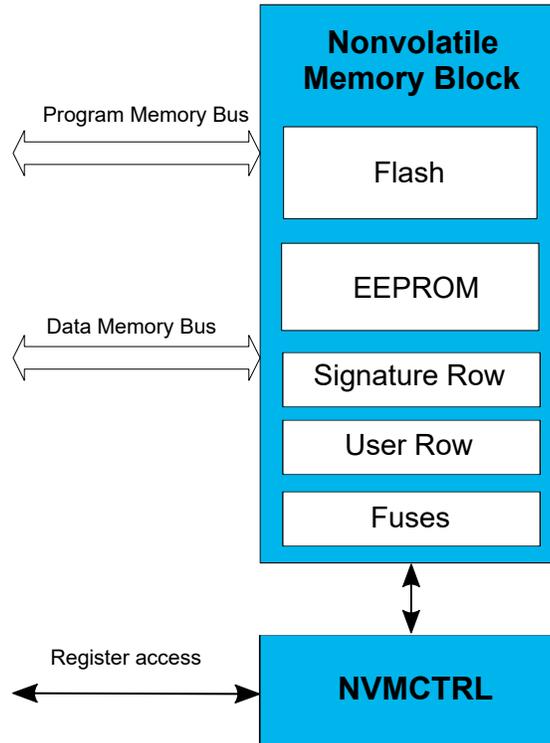
- 统一存储器
- 系统内可编程
- 自编程和自举程序支持
- 可配置写保护的段：
 - 用于存储自举程序代码或应用程序代码的引导段
 - 用于存储应用程序代码的应用程序代码段
 - 用于存储应用程序代码或数据的应用程序数据段
- 用于出厂编程数据的签名行：
 - 每种器件类型的 ID
 - 每个器件的序列号
 - 出厂校准外设的校准字节
- 用于应用程序数据的用户行：
 - 可通过软件读写
 - 在锁定的器件上可通过 UPDI 写入
 - 此内容在全片擦除后仍会保留

8.2. 概述

NVM 控制器（NVMCTRL）是 CPU 与非易失性存储器（闪存、EEPROM、签名行、用户行和熔丝）之间的接口。这些是可重新编程的存储器块，即使在未供电的情况下也能保留其值。闪存主要用于存储程序，也可用于存储数据，而 EEPROM、签名行、用户行和熔丝用于存储数据。

8.2.1. 框图

图 8-1. NVMCTRL 框图



8.3. 功能说明

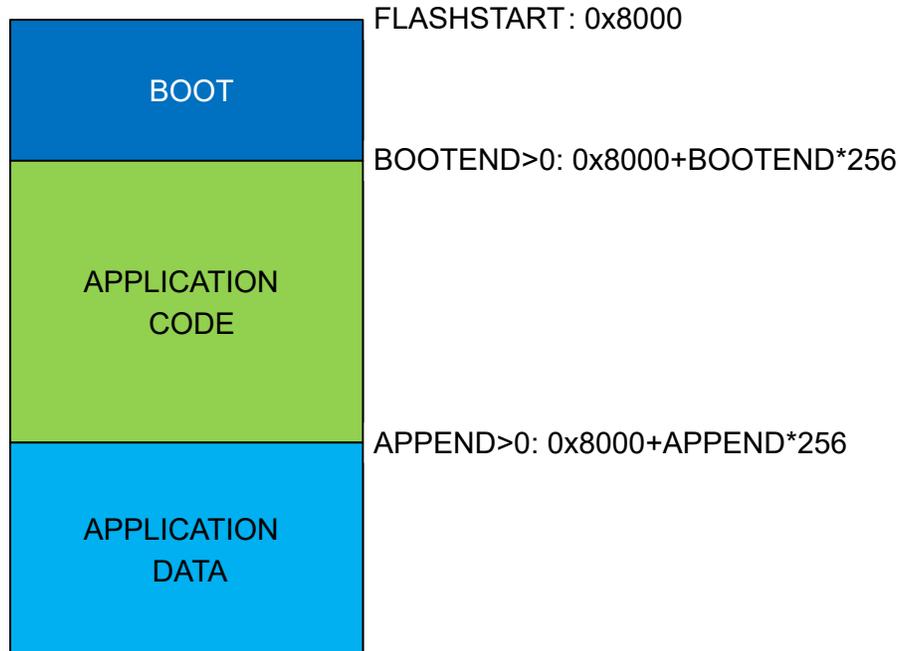
8.3.1. 存储器构成

8.3.1.1. 闪存

闪存分为一组页。页是编程闪存时寻址的基本单元。只能一次写入或擦除整页。一页由多个字组成。

为实现不同的安全级别，可以 256 字节的块为单位将闪存分为三个段。这三个不同的段分别是引导段（BOOT）、应用程序代码段（APPCODE）和应用程序数据段（APPDATA）。

图 8-2. 闪存段



段大小

这些段的大小由引导段末尾（FUSE.BOOTEND）熔丝和应用程序代码段末尾（FUSE.APPEND）熔丝设置。

熔丝将选择段大小（以 256 字节的块为单位）。BOOT 段的范围为闪存开头到 BOOTEND。APPCODE 段的范围为 BOOTEND 到 APPEND。剩余区域是 APPDATA 段。

表 8-1. 设置闪存段

BOOTEND	APPEND	BOOT 段	APPCODE 段	APPDATA 段
0	—	0 至 FLASHEND	—	—
> 0	0	0 至 256*BOOTEND	256*BOOTEND 至 FLASHEND	—
> 0	≤ BOOTEND	0 至 256*BOOTEND	—	256*BOOTEND 至 FLASHEND
> 0	> BOOTEND	0 至 256*BOOTEND	256*BOOTEND 至 256*APPEND	256*APPEND 至 FLASHEND

如果 BOOTEND 写入 0，则会将整个闪存视为 BOOT 段。如果 APPEND 写入 0 且 BOOTEND > 0，则 APPCODE 段的范围为 BOOTEND 到闪存末尾（无 APPDATA 段）。当 APPEND ≤ BOOTEND 时，移除 APPCODE 段，APPDATA 从 BOOTEND 一直到闪存末尾。当 APPEND > BOOTEND 时，APPCODE 段从 BOOTEND 一直延伸至 APPEND。剩余区域是 APPDATA 段。

如果没有自举程序软件，建议使用 BOOT 段存储应用程序代码。

注：

- 复位后，默认的向量表位于 APPCODE 段的开头。通过将中断向量表重定位到 BOOT 段开头，可以在该段内运行的代码中使用外设中断。方法是将 CPUINT.CTRLA 寄存器中的 IVSEL 位置 1。有关详细信息，请参见 CPUINT 部分。
- 如果 BOOTEND/APPEND（通过 BOOTEND 和 APPEND 熔丝设置生成）超出器件的 FLASHEND，则忽略相应的熔丝设置并使用默认值。关于默认值，请参见存储器部分的“熔丝”。

例 8-1. 闪存段的大小

如果 FUSE.BOOTEND 写入 0x04 并且 FUSE.APPEND 写入 0x08，则前 4*256 字节是 BOOT，接下来的 4*256 字节是 APPCODE，剩余的闪存部分是 APPDATA。

段间写保护

三个闪存段之间已实现定向写保护。

- BOOT 段中的代码可以写入 APPCODE 和 APPDATA
- APPCODE 段中的代码可以写入 APPDATA
- APPDATA 段中的代码可以写入闪存或 EEPROM

引导段锁定和应用程序代码段写保护

除了段间写保护之外，NVMCTRL 还提供了安全机制以防止对闪存段进行不必要的访问。虽然 CPU 永远无法写入 BOOT 段，但控制 B (NVMCTRL.CTRLB) 寄存器中提供了引导段锁定 (BOOTLOCK) 位来防止从 BOOT 段读取和执行代码。该位只能通过在 BOOT 段中执行的代码进行设置，并且仅在退出 BOOT 段时生效。

将控制 B (NVMCTRL.CTRLB) 寄存器中的应用程序代码段写保护 (APCWP) 位置 1 可以防止对 APPCODE 段进行进一步更新。

8.3.1.2. EEPROM

EEPROM 分为多个页，一页包含多个字节。EEPROM 在擦除/写入操作时使用字节粒度。在一页内，只有标记为更新的字节才会被擦除/写入。通过向相应地址单元的页缓冲区写入新值来标记字节。

8.3.1.3. 用户行

用户行是 EEPROM 的一个额外页。该页可用于存储各种数据，例如校准/配置数据和序列号。该页不会被全片擦除操作擦除。用户行的写入方式与正常 EEPROM 一样，此外还可以在锁定的器件上通过 UPDI 进行写操作。

8.3.2. 存储器访问**8.3.2.1. 读取**

闪存和 EEPROM 通过存储器映射的地址和装入指令来读取。在写入或擦除过程中读取任何阵列都将导致总线进入等待状态，并且指令将被暂停，直到所进行的操作完成。

8.3.2.2. 页缓冲区装入

通过直接写入存储器映射中定义的存储器来装入页缓冲区。闪存、EEPROM 和用户行共用同一页缓冲区，因此一次只能编程一个段。地址的最低有效位 (Least Significant bit, LSb) 用于选择页缓冲区中写入数据的位置。得到的数据是页缓冲区的新内容和原内容进行与运算后的二进制结果。页缓冲区将在以下事件后自动擦除 (所有位置 1)：

- 器件复位
- 任何页写入或擦除操作
- 清零页缓冲区命令
- 器件从任何休眠模式中唤醒

8.3.2.3. 编程

对于页编程而言，填充页缓冲区以及将页缓冲区的内容写入闪存、用户行和 EEPROM 是两个单独的操作。

在使用页缓冲区的数据编程闪存页之前，必须先擦除闪存页。当器件进入休眠模式时，会同时擦除页缓冲区。编程未擦除的闪存页将损坏其内容。

写入闪存时，既可以单独使用擦除命令和写入命令，也可以使用擦除/写入命令：

备选方案 1:

1. 填充页缓冲区。
2. 使用擦除/写入页（ERWP）命令将页缓冲区的内容写入闪存。

备选方案 2:

1. 写入页上的一个存储单元以设置地址。
2. 执行擦除页（Erase Page, ER）命令。
3. 填充页缓冲区。
4. 执行写入页（Write Page, WP）命令。

NVM 命令集既支持擦除后再写入操作，也支持独立的页擦除（ER）和页写入（WP）命令。这种独立命令可缩短每个命令的编程时间，并且擦除操作可以在非时间关键编程操作执行期间完成。

EEPROM 的编程类似，但只有页缓冲区中更新的字节才会在 EEPROM 中写入或擦除。

8.3.2.4. 命令

可通过正常的装入/存储指令来执行对闪存/EEPROM 的读操作和对页缓冲区的写操作。其他操作（例如写入和擦除存储器阵列）可在 NVM 中通过命令执行。

要在 NVM 中执行命令：

1. 通过读取 NVMCTRL.STATUS 寄存器中的繁忙（EEBUSY 和 FBUSY）标志确认之前的操作是否已完成。
2. 向配置更改保护（CPU.CCP）寄存器中写入相应的密钥，以解锁 NVM 控制 A（NVMCTRL.CTRLA）寄存器。
3. 在接下来的四条指令内，将所需的命令值写入控制 A（NVMCTRL.CTRLA）寄存器中的 CMD 位域。

8.3.2.4.1. 写入页命令

闪存控制器的写入页（WP）命令可将页缓冲区的内容写入闪存或 EEPROM。

写入闪存时，只要闪存忙于处理写操作，CPU 就会停止执行代码。写入 EEPROM 时，CPU 可以在操作执行过程中继续执行代码。

操作完成后，页缓冲区将自动清除。

8.3.2.4.2. 擦除页命令

擦除页（ER）命令用于擦除当前页。必须在页缓冲区中写入一个字节才能使擦除页（ER）命令生效。

要擦除闪存，首先要在所需的页中写入一个地址，然后执行命令。闪存中的整页随后将被擦除。擦除过程中 CPU 将处于停止状态。

对于 EEPROM，只有在执行该命令时才会擦除页缓冲区中写入的字节。要擦除特定字节，可在执行命令前写入相应的地址。要擦除整页，必须在执行命令前对页缓冲区中的所有字节进行更新。CPU 可以在操作执行过程中继续运行代码。

操作完成后，页缓冲区将自动清除。

8.3.2.4.3. 擦除/写入页命令

擦除/写入页（Erase and Write Page, ERWP）命令是擦除页和写入页命令的结合，但在擦除页命令后不会清除页缓冲区：擦除/写入操作会先擦除所选页，然后将页缓冲区的内容写入这一页。

在闪存上执行擦除-写入操作时，CPU 将会停止。在 EEPROM 上执行该操作时，CPU 可以继续执行代码。

操作完成后，页缓冲区将自动清除。

8.3.2.4.4. 页缓冲区清除命令

页缓冲区清除（Page Buffer Clear, PBC）命令用于清除页缓冲区。页缓冲区的内容将在操作完成后全部变为 1。操作执行时，CPU 将处于停止状态（7 个 CPU 周期）。

8.3.2.4.5. 全片擦除命令

全片擦除（CHER）命令用于擦除闪存和 EEPROM。如果 FUSE.SYSCFG0 中的全片擦除期间保存 EEPROM（EESAVE）熔丝置 1，则 EEPROM 保持不变。闪存不受 NVMCTRL.CTRLB 中的引导段锁定（BOOTLOCK）位或应用程序代码段写保护（APCWP）位的保护。操作完成后，存储器的内容将全部变为 1。

8.3.2.4.6. EEPROM 擦除命令

EEPROM 擦除（EEPROM Erase, EEER）命令用于擦除 EEPROM。操作完成后，EEPROM 的内容将全部变为 1。擦除 EEPROM 时，CPU 将处于停止状态。

8.3.2.4.7. 写入熔丝命令

写入熔丝（Write Fuse, WFU）命令用于写入熔丝。只能通过 UPDI 使用该命令；CPU 无法启动该命令。

要使用写入熔丝命令，请按照以下步骤操作：

1. 将熔丝的地址写入地址（NVMCTRL.ADDR）寄存器。
2. 将要写入熔丝的数据写入数据（NVMCTRL.DATA）寄存器。
3. 执行写入熔丝命令。
4. 写入熔丝后，需要复位才能使更新的值生效。

要读取熔丝，可对存储单元使用常规读操作。

8.3.2.5. 复位后的写访问

上电复位（POR）后，NVMCTRL 会在一定时间内拒绝任何尝试写入 NVM 的操作。在此期间，NVMCTRL.STATUS 寄存器中的闪存繁忙（FBUSY）和 EEPROM 繁忙（EEBUSY）位域将读为 1。EEBUSY 和 FBUSY 位域必须读为 0，才能填充页缓冲区或发出 NVM 命令。

通过向系统配置 0（FUSE.SYSCFG0）熔丝中的超时禁止位（TOUTDIS）写入 0 或通过 FUSE.SYSCFG0 熔丝中的 RSTPINCFG 位域配置为 UPDI，可以禁止该超时周期。

8.3.3. 防止闪存/EEPROM 损坏

在低 V_{DD} 期间，如果 CPU 和闪存/EEPROM 因电源电压过低而无法正常工作，则闪存程序或 EEPROM 数据可能会损坏。使用闪存/EEPROM 的电路板级系统也同样面临着这些问题，可应用相同的设计解决方案。

当电压过低时，以下两种情况可能造成闪存/EEPROM 损坏：

1. 对闪存的常规写序列，需要最小电压才能正常工作。
2. 当电源电压过低时，CPU 本身可能会错误地执行指令。

有关最大频率与 V_{DD} 的关系，请参见 *电气特性* 一章。



注意：可以采取以下措施避免闪存/EEPROM 损坏：

1. 在电源电压不足期间，将器件保持在复位状态。该操作可以通过使能内部欠压检测器（BOD）来完成。
2. BOD 中的电压监视器可用于防止接近 BOD 电压时启动 EEPROM 写操作。
3. 如果内部 BOD 的检测电压与所需的检测电压不匹配，则可以使用外部低 V_{DD} 复位保护电路。如果在写操作执行过程中发生复位，则写操作将会中止。

8.3.4. 中断

表 8-2. 可用中断向量和源

失调电压	名称	向量说明	条件
0x00	EEREADY	NVM	EEPROM 已准备好执行新的写入/擦除操作。

发生中断条件时，中断标志（NVMCTRL.INTFLAGS）寄存器中的相应中断标志将置 1。

可以通过写入中断控制（NVMCTRL.INTCTRL）寄存器中的相应位来允许或禁止中断源。

当相应的中断源被允许且中断标志置 1 时，将生成中断请求。中断请求保持有效，直到中断标志清零。有关如何清零中断标志的详细信息，请参见 NVMCTRL.INTFLAGS 寄存器。

8.3.5. 休眠模式操作

如果系统进入休眠模式时未在进行写操作，则 NVMCTRL 将进入休眠模式。

如果系统进入休眠模式时正在进行写操作，则 NVM 模块、NVM 控制器和系统时钟将保持开启，直到写操作完成。这适用于所有休眠模式，包括掉电休眠模式。

EEPROM 就绪中断只会在空闲休眠模式下唤醒器件。

从休眠状态唤醒时，页缓冲区将清零。

8.3.6. 配置更改保护

此外设的一些寄存器具有配置更改保护（CCP）。要写入这些寄存器，必须先将特定密钥写入 CPU.CCP 寄存器，然后在 4 条 CPU 指令内对受保护位进行写访问。

必须遵循相应的 CCP 解锁序列，否则无法更改受保护的寄存器。

以下寄存器具有 CCP：

表 8-3. NVMCTRL——配置更改保护下的寄存器

寄存器	密钥
NVMCTRL.CTRLA	SPM
NVMCTRL.CTRLB	IOREG

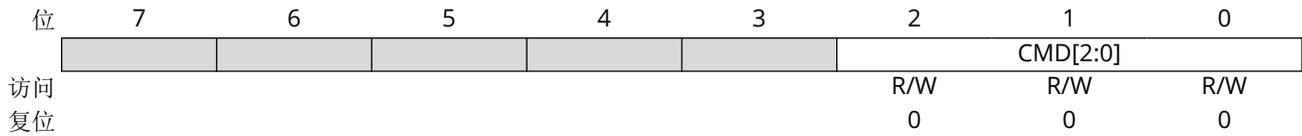
8.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0							CMD[2:0]	
0x01	CTRLB	7:0							BOOTLOCK	APCWP
0x02	STATUS	7:0						WRERROR	EEBUSY	FBUSY
0x03	INTCTRL	7:0								EEREADY
0x04	INTFLAGS	7:0								EEREADY
0x05	保留									
0x06	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
0x08	ADDR	7:0	ADDR[7:0]							
		15:8	ADDR[15:8]							

8.5. 寄存器说明

8.5.1. 控制 A

名称: CTRLA
 偏移量: 0x00
 复位: 0x00
 属性: 配置更改保护



Bit 2:0 - CMD[2:0] 命令

对该位域进行写操作以发出命令。在进行此写操作之前，必须在四条指令内写入用于自编程（SPM）的配置更改保护密钥。

值	名称	说明
0x0	-	无命令
0x1	WP	将页缓冲区写入存储器（通过 NVMCTRL.ADDR 选择存储器）
0x2	ER	擦除页（通过 NVMCTRL.ADDR 选择存储器）
0x3	ERWP	擦除和写入页（通过 NVMCTRL.ADDR 选择存储器）
0x4	PBC	页缓冲区清除
0x5	CHER	全片擦除：擦除闪存和 EEPROM（除非 FUSE.SYSCFG 中的 EESAVE 为 1）
0x6	EEER	EEPROM 擦除
0x7	WFU	写入熔丝（只能通过 UPDI 访问）

8.5.2. 控制 B

名称: CTRLB
 偏移量: 0x01
 复位: 0x00
 属性: 配置更改保护

位	7	6	5	4	3	2	1	0
							BOOTLOCK	APCWP
访问							R/W	R/W
复位							0	0

Bit 1 - BOOTLOCK 引导段锁定

向该位写入 1 会将 BOOT 段锁定，防止从中读取数据和取指令。

如果该位为 1，从 BOOT 段读取数据将返回 0。从 BOOT 段取指令也将返回 0。

该位只能通过 BOOT 段写入，并且只能通过复位清零。

第一次写入该位后，必须退出 BOOT 段才会生效。

Bit 0 - APCWP 应用程序代码段写保护

向该位写入 1 可防止对应用程序代码段进行进一步更新。

该位只能写入 1，并且只能通过复位清零。

8.5.3. 状态

名称: STATUS
 偏移量: 0x02
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
						WRERROR	EEBUSY	FBUSY
访问						R	R	R
复位						0	0	0

Bit 2 - WRERROR 写入错误

发生写入错误时，该位将读为 1。写入错误可能包括：在进行页写操作之前写入不同的段，或写入受保护区域。该位适用于上一次操作。

Bit 1 - EEBUSY EEPROM 繁忙

EEPROM 忙于处理命令时，该位将读为 1。

Bit 0 - FBUSY 闪存繁忙

闪存忙于处理命令时，该位将读为 1。

8.5.4. 中断控制

名称: INTCTRL
 偏移量: 0x03
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
访问								EEREADY
复位								R/W 0

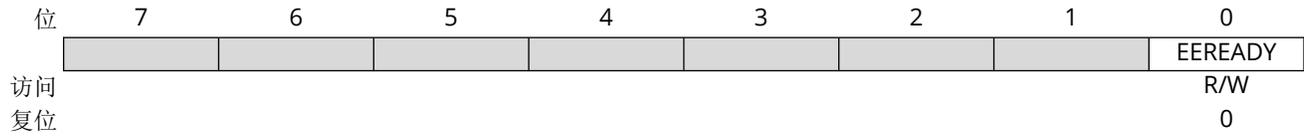
Bit 0 - EEREADY EEPROM 就绪中断

向该位写入 1 会允许中断，这表示 EEPROM 已准备好进行新的写操作/擦除操作。

这是仅在 INTFLAGS 寄存器中的 EEREADY 标志设置为 0 时才会触发的优先级中断。因此，在触发 NVM 命令之前不得允许该中断，因为在发出 NVM 命令之前 EEREADY 标志不会置 1。可在中断处理程序中禁止该中断。

8.5.5. 中断标志

名称: INTFLAGS
 偏移量: 0x04
 复位: 0x00
 属性: -



Bit 0 - EEREADY EEREADY 中断标志

只要 EEPROM 不忙，该标志就会持续置 1。该标志通过写入 1 来清零。

8.5.6. 数据

名称: DATA
 偏移量: 0x06
 复位: 0x00
 属性: -

NVMCTRL.DATAL 和 NVMCTRL.DATAH 寄存器对代表 16 位值 NVMCTRL.DATA。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

位	15	14	13	12	11	10	9	8
	DATA[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	DATA[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:0 – DATA[15:0] 数据寄存器

UPDI 使用该寄存器进行熔丝写操作。

8.5.7. 地址

名称: ADDR
 偏移量: 0x08
 复位: 0x00
 属性: -

NVMCTRL.ADDRL 和 NVMCTRL.ADDRH 寄存器对代表 16 位值 NVMCTRL.ADDR。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

位	15	14	13	12	11	10	9	8
	ADDR[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	ADDR[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:0 - ADDR[15:0] 地址

地址寄存器包含已更新的最后一个存储单元的地址。

9. CLKCTRL——时钟控制器

9.1. 特性

- 所有时钟和时钟源均可应外设请求自动使能。
- 内部振荡器：
 - 16/20 MHz 振荡器 (OSC20M)
 - 32.768 kHz 超低功耗振荡器 (OSCULP32K)
- 外部时钟选项：
 - 32.768 kHz 晶振 (XOSC32K)
 - 外部时钟
- 主时钟特性：
 - 安全运行时切换
 - 支持 12 种分频设置 (1x 至 64x) 的预分频器

9.2. 概述

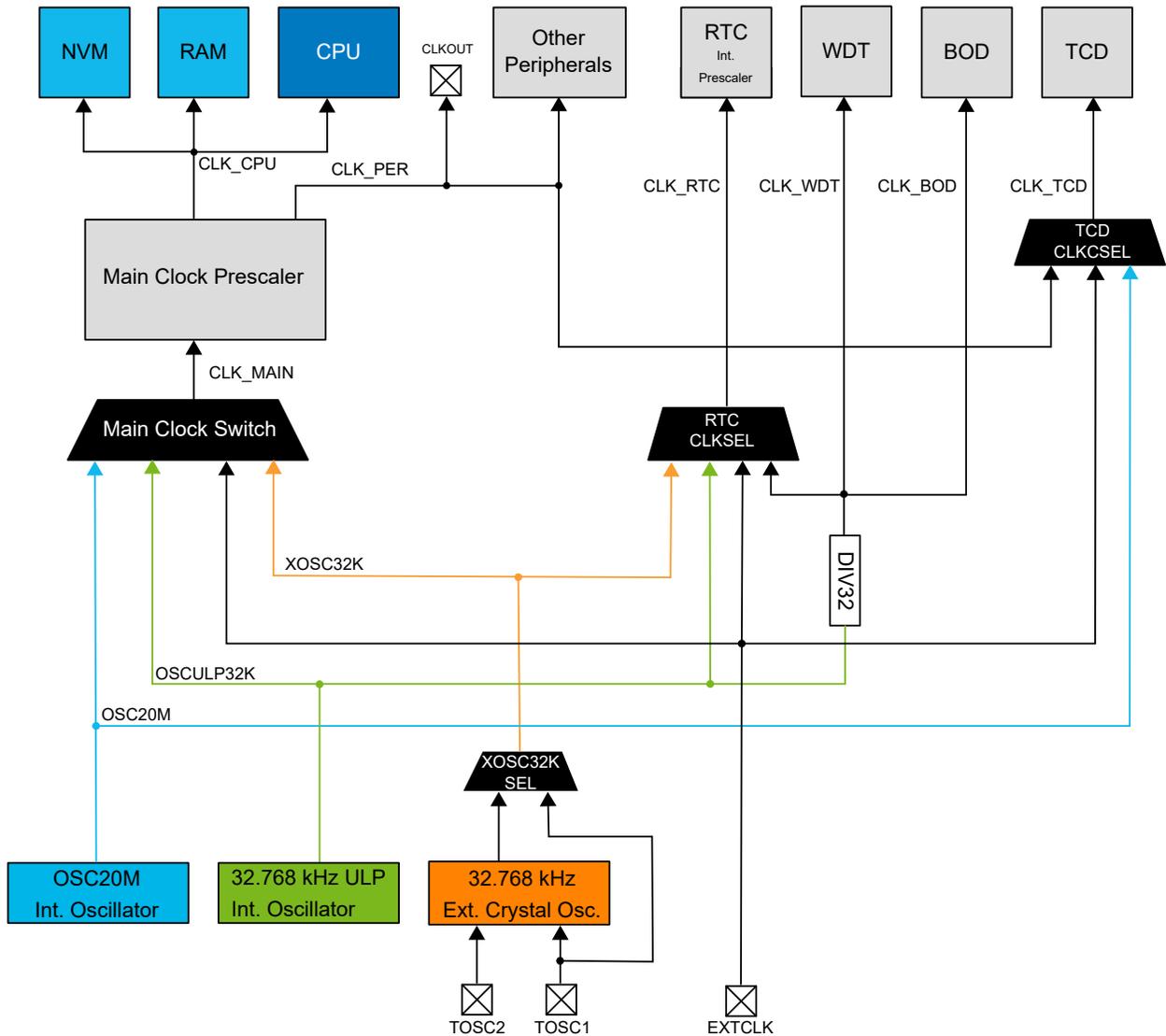
时钟控制器 (CLKCTRL) 外设用于对来自可用振荡器的时钟信号进行控制、分配和预分频。CLKCTRL 支持内部和外部时钟源。

CLKCTRL 基于自动时钟请求系统 (在器件的所有外设中均已实现)。外设将自动请求所需的时钟。如果有多个时钟源, 则会将请求路由到正确的时钟源。

主时钟 (CLK_MAIN) 供 CPU、RAM 和 I/O 总线使用。主时钟源不但可以选择, 而且可以进行预分频。某些外设可与主时钟共用相同的时钟源, 但也可选择与主时钟域异步运行。

9.2.1. 框图——CLKCTRL

图 9-1. CLKCTRL 框图



时钟系统由主时钟和其他异步时钟组成：

- 主时钟
该时钟供 CPU、RAM、闪存、I/O 总线以及连接到 I/O 总线的所有外设使用。在工作模式和空闲休眠模式下，主时钟将保持运行。在待机休眠模式下，主时钟将在外设请求时运行。
主时钟 CLK_MAIN 由时钟控制器预分频和分配：
 - CLK_CPU 供 CPU、SRAM 和 NVMCTRL 外设用来访问非易失性存储器
 - CLK_PER 供未在异步时钟下列出的所有外设使用
- 与主时钟域异步运行的时钟：
 - CLK_RTC 供 RTC/PIT 使用。当使能 RTC/PIT 时，将请求该时钟。仅当禁止外设时，才可更改 CLK_RTC 的时钟源。
 - CLK_WDT 供 WDT 使用。当使能 WDT 时，将请求该时钟。
 - CLK_BOD 供 BOD 使用。当采样模式下使能 BOD 时，将请求该时钟。

- CLK_TCD 供 TCD 使用。当使能 TCD 时，将请求该时钟。仅当禁止外设时，才可更改时钟源。

通过写入主时钟控制 A（CLKCTRL.MCLKCTRLA）寄存器中的时钟选择（CLKSEL）位来配置主时钟域的时钟源。异步时钟源由相应外设中的寄存器配置。

9.2.2. 信号说明

信号	类型	说明
CLKOUT	数字输出	CLK_PER 输出

9.3. 功能说明

9.3.1. 休眠模式操作

未使用/请求时钟源时，时钟源将关闭。向相应振荡器的控制 A（CLKCTRL.[osc]CTRLA）寄存器中的待机运行（RUNSTDBY）位写入 1 可以直接请求时钟源。这将导致振荡器持续运行，但掉电休眠模式下除外。此外，如果该位写入 1，则当外设请求时钟源时，振荡器起振时间会被消除。

在工作模式和空闲休眠模式下，主时钟将始终运行。在待机休眠模式下，主时钟仅在外设请求时，或在相应振荡器的控制 A（CLKCTRL.[osc]CTRLA）寄存器中的待机运行（RUNSTDBY）位写入 1 时运行。

在掉电休眠模式下，主时钟将在所有 NVM 操作完成后停止。

9.3.2. 主时钟选择和预分频器

所有内部振荡器均可用作 CLK_MAIN 的主时钟源。主时钟源可通过软件选择，并可在正常工作期间安全地切换。

系统内置硬件保护功能，可确保时钟切换的安全性。

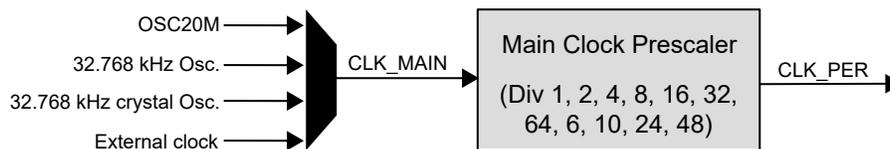
在选择外部时钟源时，只有在检测到相应边沿时，才会切换到所选择的时钟源。在检测到足够数量的时钟边沿之前，不会进行切换，并且必须执行复位才能再次切换为另一个时钟源。

正在进行的时钟源切换通过主时钟状态（CLKCTRL.MCLKSTATUS）寄存器中的系统振荡器切换（SOSC）标志来指示。外部时钟源的稳定性通过相应的状态（CLKCTRL.MCLKSTATUS 中的 EXTS 和 XOSC32KS）标志来指示。

CAUTION 如果外部时钟源在用作 CLK_MAIN 源时发生故障，则只有 WDT 可以通过系统复位切换回之前的时钟源。

CLK_MAIN 馈入预分频器，然后再提供给器件中的外设（CLK_PER）使用。预分频器将使用 1 到 64 的系数对 CLK_MAIN 进行分频。

图 9-2. 主时钟和预分频器



主时钟和预分频器配置（CLKCTRL.MCLKCTRLA 和 CLKCTRL.MCLKCTRLB）寄存器受配置更改保护机制的保护，需采用定时写入程序来更改这些寄存器。

9.3.3. 复位后的主时钟

发生复位后，CLK_MAIN 将由 16/20 MHz 振荡器（OSC20M）进行 6 分频后提供。由于 OSC20M 的实际频率由振荡器配置（FUSE.OSCCFG）熔丝的频率选择（FREQSEL）位决定，复位后可提供以下频率：

表 9-1. 复位后的外设时钟频率

CLK_MAIN (由 FUSE.OSCCFG 中的 FREQSEL 决定)	最终得到的 CLK_PER
16 MHz	2.67 MHz
20 MHz	3.33 MHz

更多详细信息，请参见 OSC20M 说明。

9.3.4. 时钟源

所有内部时钟源在外设请求时自动使能。基于外部晶振的晶振必须先通过向 32.768 kHz 晶振控制 A (CLKCTRL.XOSC32KCTRLA) 寄存器中的 ENABLE 位写入 1 使能，才能用作时钟源。

主时钟状态 (CLKCTRL.MCLKSTATUS) 寄存器中相应的振荡器状态位指示时钟源是否正在运行且保持稳定。

9.3.4.1. 内部振荡器

内部振荡器不需要依靠任何外部元件即可运行。

9.3.4.1.1. 16/20 MHz 振荡器 (OSC20M)

该振荡器可工作在多个频率下，具体由振荡器配置 (FUSE.OSCCFG) 熔丝中的频率选择 (FREQSEL) 位值进行选择。中心频率如下：

- 16 MHz
- 20 MHz

系统复位后，FUSE.OSCCFG 确定 CLK_MAIN 的初始频率。

复位期间，OSC20M 的校准值从熔丝装入。有两个不同的校准位域：

- 校准 A (CLKCTRL.OSC20MCALIBA) 寄存器中的校准 (CAL20M) 位域可围绕当前中心频率进行校准
- 校准 B (CLKCTRL.OSC20MCALIBB) 寄存器中的振荡器温度系数校准 (TEMPCAL20M) 位域可调节温度漂移补偿的斜率

如果应用要求更精确的频率设置 (相比于振荡器校准所能提供的频率)，可使用出厂校准的频率误差。

振荡器校准可通过振荡器锁定 (OSCLOCK) 熔丝 (FUSE.OSCCFG) 锁定。该熔丝为 1 时，无法更改校准。如果将该振荡器用作主时钟源，校准将被锁定，控制 B (CLKCTRL.OSC20MCALIBB) 寄存器中的锁定使能 (LOCKEN) 位将为 1。

校准位受配置更改保护机制保护，需要定时写入程序来更改主时钟和预分频器设置。

该振荡器的起振时间等于模拟起振时间加上四个振荡器周期。有关起振时间的信息，请参见 *电气特性* 一章。

更改振荡器校准值时，频率可能会过冲。如果振荡器用作主时钟 (CLK_MAIN)，建议更改主时钟预分频器，以使主时钟频率不超过主时钟最大工作频率的 $\frac{1}{4}$ ，如“一般工作额定值”一节所述。在更新振荡器校准值之后，可更改系统时钟预分频器。

9.3.4.1.2. 32.768 kHz 振荡器 (OSCULP32K)

32.768 kHz 振荡器针对超低功耗 (ULP) 操作进行了优化。与外部晶振相比，功耗得以降低，但代价是精度也随之降低。

该振荡器为实时计数器 (RTC)、看门狗定时器 (WDT) 和欠压检测器 (BOD) 提供 1.024 kHz 信号。

该振荡器的起振时间等于振荡器起振时间加上四个振荡器周期。有关起振时间的信息，请参见“电气特性”一章。

9.3.4.2. 外部时钟源

可使用以下外部时钟源：

- 来自引脚 EXTCLK 的外部时钟
- TOSC1 和 TOSC2 引脚上的 32.768 kHz 晶振
- TOSC1 引脚上的 32.768 kHz 外部时钟

9.3.4.2.1. 外部时钟 (EXTCLK)

EXTCLK 直接取自引脚。该 GPIO 引脚会在任何外设请求该时钟时自动配置为 EXTCLK。

首次请求该时钟源时有两个周期的启动时间。

9.3.4.2.2. 32.768 kHz 晶振 (XOSC32K)

该振荡器支持两种输入选项：将晶振连接到引脚 TOSC1 和 TOSC2，或者将以 32.768 kHz 运行的外部时钟连接到 TOSC1。必须通过写入 XOSC32K 控制 A (CLKCTRL.XOSC32KCTRLA) 寄存器中的时钟源选择 (SEL) 位来配置输入选项。

通过向 CLKCTRL.XOSC32KCTRLA 中的 ENABLE 位写入 1 来使能 XOSC32K。使能后，XOSC32K 使用的 GPIO 引脚配置被改写为 TOSC1 和 TOSC2 引脚。要使振荡器能够应请求开始运行，需要将 ENABLE 位置 1。

通过写入 CLKCTRL.XOSC32KCTRLA 中的晶振起振时间 (CSUT) 位，可以调节给定晶振的起振时间。

当 XOSC32K 配置为在 TOSC1 上使用外部时钟时，起振时间固定为两个周期。

9.3.5. 配置更改保护

该外设的一些寄存器具有配置更改保护 (CCP)。要写入这些寄存器，必须先将给定密钥写入 CPU.CCP 寄存器，然后在 4 条 CPU 指令内对受保护位进行写访问。

必须遵循相应的 CCP 解锁序列，否则无法更改受保护的寄存器。

以下寄存器具有 CCP：

表 9-2. CLKCTRL——配置更改保护下的寄存器

寄存器	密钥
CLKCTRL.MCLKCTRLB	IOREG
CLKCTRL.MCLKLOCK	IOREG
CLKCTRL.XOSC32KCTRLA	IOREG
CLKCTRL.MCLKCTRLA	IOREG
CLKCTRL.OSC20MCTRLA	IOREG
CLKCTRL.OSC20MCALIBA	IOREG
CLKCTRL.OSC20MCALIBB	IOREG
CLKCTRL.OSC32KCTRLA	IOREG

9.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	MCLKCTRLA	7:0	CLKOUT						CLKSEL[1:0]	
0x01	MCLKCTRLB	7:0				PDIV[3:0]				PEN
0x02	MCLKLOCK	7:0								LOCKEN
0x03	MCLKSTATUS	7:0	EXTS	XOSC32KS	OSC32KS	OSC20MS				SOSC
0x04	...									
0x0F	保留									
0x10	OSC20MCTRLA	7:0							RUNSTDBY	
0x11	OSC20MCALIBA	7:0			CAL20M[5:0]					
0x12	OSC20MCALIBB	7:0	LOCK				TEMPCAL20M[3:0]			
0x13	...									
0x17	保留									
0x18	OSC32KCTRLA	7:0							RUNSTDBY	
0x19	...									
0x1B	保留									
0x1C	XOSC32KCTRLA	7:0			CSUT[1:0]			SEL	RUNSTDBY	ENABLE

9.5. 寄存器说明

9.5.1. 主时钟控制 A

名称: MCLKCTRLA
偏移量: 0x00
复位: 0x00
属性: 配置更改保护

位	7	6	5	4	3	2	1	0
	CLKOUT						CLKSEL[1:0]	
访问	R/W						R/W	R/W
复位	0						0	0

Bit 7 - CLKOUT 系统时钟输出

向该位写入 1 时，系统时钟会输出到 CLKOUT 引脚。
器件处于休眠模式时，除非外设使用系统时钟，否则无时钟输出。

Bit 1:0 - CLKSEL[1:0] 时钟选择

该位域选择主时钟（CLK_MAIN）的源。

值	名称	说明
0x0	OSC20M	16/20 MHz 内部振荡器
0x1	OSCULP32K	32.768 kHz 内部超低功耗振荡器
0x2	XOSC32K	32.768 kHz 外部晶振
0x3	EXTCLK	外部时钟

9.5.2. 主时钟控制 B

名称: MCLKCTRLB
偏移量: 0x01
复位: 0x11
属性: 配置更改保护

位	7	6	5	4	3	2	1	0
				PDIV[3:0]				PEN
访问				R/W	R/W	R/W	R/W	R/W
复位				1	0	0	0	1

Bit 4:1 - PDIV[3:0] 预分频器分频

如果向预分频器使能（PEN）位写入 1，该位域将定义主时钟预分频器的分频比。

可以在运行期间写入该位域，以更改系统的时钟频率，进而满足应用要求。

用户软件必须确保正确配置输入频率（CLK_MAIN）和预分频器设置，确保得到的 CLK_PER 频率不会超过允许的最大值（见电气特性）。

值	说明
值	除法
0x0	2
0x1	4
0x2	8
0x3	16
0x4	32
0x5	64
0x8	6
0x9	10
0xA	12
0xB	24
0xC	48
其他	保留

Bit 0 - PEN 预分频器使能

要使能预分频器，必须向该位写入 1。使能后，分频比由 PDIV 位域选择。

向此位写入 0 时，无论 PDIV 的值如何，都不会对主时钟进行分频（CLK_PER = CLK_MAIN）。

9.5.3. 主时钟锁定

名称: MCLKLOCK
 偏移量: 0x02
 复位: 基于 FUSE.OSCCFG 中的 OSCLOCK
 属性: 配置更改保护

位	7	6	5	4	3	2	1	0
访问								LOCKEN
复位								R/W x

Bit 0 - LOCKEN 锁定使能

向该位写入 1 将锁定 CLKCTRL.MCLKCTRLA 和 CLKCTRL.MCLKCTRLB 寄存器，以及后续软件更新中提供的当前主时钟源的校准设置（如适用）。锁定后，CLKCTRL.MCLKLOCK 寄存器将无法访问，直到下一次硬件复位。

这可以防止软件意外修改 CLKCTRL.MCLKCTRLA 和 CLKCTRL.MCLKCTRLB 寄存器以及主时钟源的校准设置。

复位时，将根据 FUSE.OSCCFG 中的 OSCLOCK 位装载 LOCKEN 位。

9.5.4. 主时钟状态

名称: MCLKSTATUS
偏移量: 0x03
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	EXTS	XOSC32KS	OSC32KS	OSC20MS				SOSC
访问	R	R	R	R				R
复位	0	0	0	0				0

Bit 7 - EXTS 外部时钟状态

值	说明
0	EXTCLK 尚未启动
1	EXTCLK 已启动

Bit 6 - XOSC32KS XOSC32K 状态

仅当该时钟源被请求作为主时钟或由其他模块请求时，状态位才可用。如果振荡器 RUNSTDBY 位置 1 但未使用/未请求振荡器，该位将为 0。

值	说明
0	XOSC32K 不稳定
1	XOSC32K 稳定

Bit 5 - OSC32KS OSCULP32K 状态

仅当该时钟源被请求作为主时钟或由其他模块请求时，状态位才可用。如果振荡器 RUNSTDBY 位置 1 但未使用/未请求振荡器，该位将为 0。

值	说明
0	OSCULP32K 不稳定
1	OSCULP32K 稳定

Bit 4 - OSC20MS OSC20M 状态

仅当该时钟源被请求作为主时钟或由其他模块请求时，状态位才可用。如果振荡器 RUNSTDBY 位置 1 但未使用/未请求振荡器，该位将为 0。

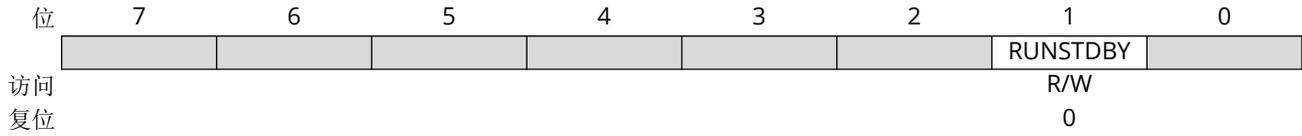
值	说明
0	OSC20M 不稳定
1	OSC20M 稳定

Bit 0 - SOSC 主时钟振荡器更改

值	说明
0	CLK_MAIN 的时钟源未进行切换
1	CLK_MAIN 的时钟源正在进行切换，新时钟源稳定后，将立即发生更改

9.5.5. 16/20 MHz 振荡器控制 A

名称: OSC20MCTRLA
偏移量: 0x10
复位: 0x00
属性: 配置更改保护



Bit 1 - RUNSTDBY 在待机模式下运行

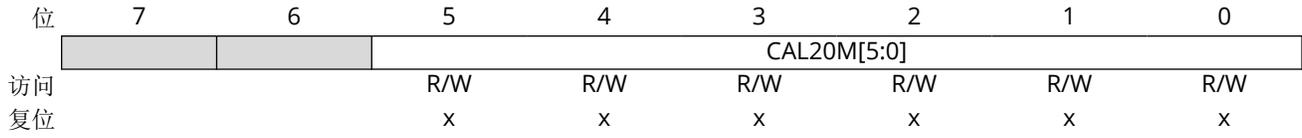
该位强制振荡器在所有模式下均开启，即使系统未使用也是如此。在待机休眠模式下，这可用于确保立即唤醒，无需等待振荡器起振时间。

外设未请求时，不提供振荡器输出。

在发出请求之后需要四个振荡器周期来打开时钟门，但是当该位置 1 时，将移除振荡器模拟起振时间。

9.5.6. 16/20 MHz 振荡器校准 A

名称: OSC20MCALIBA
 偏移量: 0x11
 复位: 基于 FUSE.OSCCFG 中的 FREQSEL
 属性: 配置更改保护



Bit 5:0 - CAL20M[5:0] 校准

该位域可围绕 OSC20M 的当前中心频率更改频率，以完成微调。
 复位时，根据 FUSE.OSCCFG 中的 FREQSEL 位装入出厂校准值。

9.5.7. 16/20 MHz 振荡器校准 B

名称: OSC20MCALIBB
偏移量: 0x12
复位: 基于 FUSE.OSCCFG
属性: 配置更改保护

位	7	6	5	4	3	2	1	0
	LOCK				TEMPCAL20M[3:0]			
访问	R				R/W	R/W	R/W	R/W
复位	x				x	x	x	x

Bit 7 - LOCK 由熔丝锁定振荡器校准

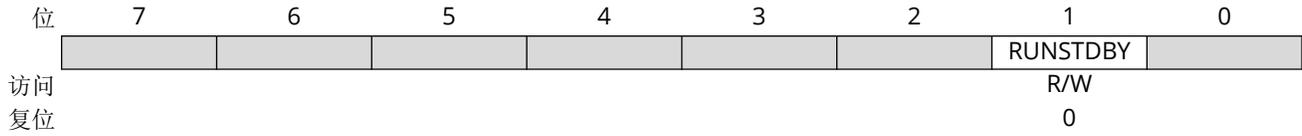
该位置 1 时，不能更改 CLKCTRL.OSC20MCALIBA 和 CLKCTRL.OSC20MCALIBB 中的校准设置。
复位时，该值从振荡器配置（FUSE.OSCCFG）熔丝中的 OSCLOCK 位装入。

Bit 3:0 - TEMPCAL20M[3:0] 振荡器温度系数校准

该位域调节温度补偿的斜率。
复位时，根据 FUSE.OSCCFG 中的 FREQSEL 位装入出厂校准值。

9.5.8. 32.768 kHz 晶振控制 A

名称: OSC32KCTRLA
偏移量: 0x18
复位: 0x00
属性: 配置更改保护



Bit 1 - RUNSTDBY 在待机模式下运行

该位强制振荡器在所有模式下均开启，即使系统未使用也是如此。在待机休眠模式下，这可用于确保立即唤醒，无需等待振荡器起振时间。

外设未请求时，不提供振荡器输出。

在发出请求之后需要四个振荡器周期来打开时钟门，但是当该位置 1 时，将移除振荡器模拟起振时间。

9.5.9. 32.768 kHz 晶振控制 A

名称: XOSC32KCTRLA
偏移量: 0x1C
复位: 0x00
属性: 配置更改保护

只要 ENABLE 位置 1 或 CLKCTRL.MCLKSTATUS 中的 XOSC32K 稳定 (XOSC32KS) 位为高电平, 就无法更改 SEL 和 CSUT 位。
要安全地更改设置, 需向 ENABLE 位写入 0, 并等待至 XOSC32KS 为 0 后使用新设置重新使能 XOSC32K。

位	7	6	5	4	3	2	1	0
			CSUT[1:0]			SEL	RUNSTDBY	ENABLE
访问			R/W	R/W		R/W	R/W	R/W
复位			0	0		0	0	0

Bit 5:4 - CSUT[1:0] 晶振起振时间

该位域选择 XOSC32K 的起振时间。当振荡器使能时 (ENABLE = 1), 该位域受到写保护。
如果 SEL = 1, 则不会应用起振时间。

值	名称	说明
0x0	1K	1k 周期
0x1	16K	16k 周期
0x2	32K	32k 周期
0x3	64K	64k 周期

Bit 2 - SEL 源选择

该位选择外部源类型。当振荡器使能时 (ENABLE = 1), 该位域受到写保护。

值	说明
0	外部晶振
1	TOSC1 引脚上的外部时钟

Bit 1 - RUNSTDBY 在待机模式下运行

向该位写入 1 可启动晶振并强制振荡器在所有模式下开启, 即使在 ENABLE 位置 1 时系统不使用的情况下也是如此。在待机休眠模式下, 这可用于确保立即唤醒而无需等待振荡器起振时间。当该位为 0 时, 晶振仅在收到请求且 ENABLE 位置 1 时运行。

除非一个或多个外设请求, 否则 XOSC32K 的输出不会发送到其他外设。

当 RUNSTDBY 位置 1 时, 如果初始晶振起振时间已经完成, 从发出请求到接收到振荡器输出之间只有两个到三个晶振周期的延时。

根据 RUNSTDBY 位, 振荡器将在器件处于工作、空闲或待机休眠模式时一直开启, 或者仅在请求时使能。该位受 I/O 保护, 可防止意外使能振荡器。

Bit 0 - ENABLE 使能

向该位写入 1 时, 相应输入引脚的配置将改写为 TOSC1 和 TOSC2。此外, 时钟源选择 (SEL) 位和晶振起振时间 (CSUT) 位变为只读。

该位受 I/O 保护, 可防止意外使能振荡器。

10. SLPCTRL——休眠控制器

10.1. 特性

- 功耗管理，可调节功耗和功能
- 三种休眠模式：
 - 空闲
 - 待机
 - 掉电
- 可配置的待机模式：外设可配置为开启或关闭

10.2. 概述

休眠模式用于关闭器件中的外设和时钟域，以节省功耗。休眠控制器（SLPCTRL）负责控制和处理在工作模式和休眠模式之间的转换。

支持四种模式：一种工作模式（执行软件）和三种休眠模式。支持的休眠模式包括空闲、待机和掉电。

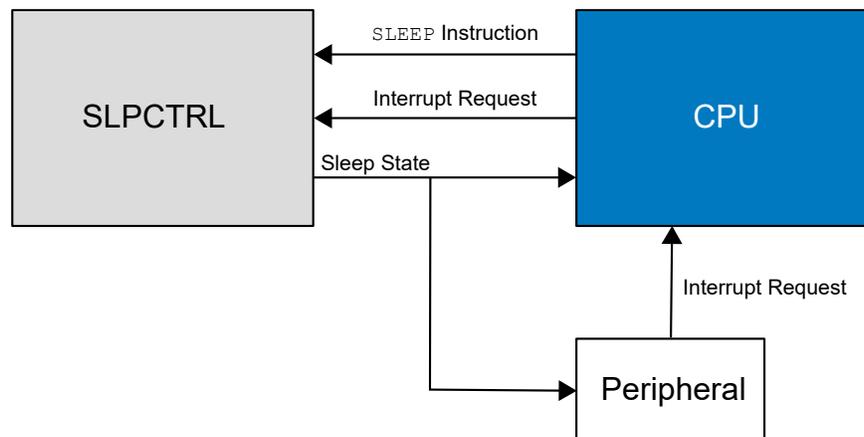
所有休眠模式均可用，也均可从工作模式切换进入。在工作模式下，CPU 执行应用程序代码。当器件进入休眠模式时，程序执行将停止。应用程序代码决定进入哪种休眠模式以及何时进入。

中断用于将器件从休眠状态唤醒。可用的中断唤醒源取决于配置的休眠模式。当发生中断时，从 SLEEP 指令后的第一条指令起，器件将唤醒并执行中断服务程序，然后继续执行正常的程序。任何复位都将使器件退出休眠模式。

休眠期间保留寄存器文件、SRAM 和寄存器的内容。如果在休眠期间发生复位，器件将从复位向量复位、启动和执行程序。

10.2.1. 框图

图 10-1. SLPCTRL 框图



10.3. 功能说明

10.3.1. 初始化

要将器件置于休眠模式，请按照以下步骤操作：

1. 配置并允许能够将器件从休眠模式唤醒的中断。
同时允许全局中断。



如果在进入休眠模式时没有允许中断，则器件无法再次唤醒。只有复位才能使器件继续工作。

- 向控制 A (SLPCTRL.CTRLA) 寄存器中的休眠模式 (SMODE) 位域和使能 (SEN) 位写入，以选择要进入的休眠模式并使能休眠控制器。

要使器件进入休眠模式，必须执行 SLEEP 指令。

10.3.2. 操作

10.3.2.1. 休眠模式

可使能三种不同的休眠模式以降低功耗。

空闲 CPU 停止执行代码，最终降低功耗。

所有外设都运行，所有中断源都可以唤醒器件。

待机 除了在待机休眠模式下使能运行的外设或时钟之外，所有高频时钟都停止运行。这通过向相应的 RUNSTDBY 位中写入 1 来使能。功耗取决于使能的功能。

部分中断源可以唤醒器件⁽¹⁾。

掉电 所有高频时钟都停止运行，最终功耗低于空闲休眠模式。

部分外设运行，部分中断源可以唤醒器件⁽¹⁾。

注：

- 更多信息，请参见 *休眠模式活动表*。

有关不同休眠模式如何影响唤醒时间的信息，请参见 *唤醒时间* 一节。

表 10-1. 外设的休眠模式活动概览

外设	可在休眠模式下工作		
	空闲	待机	掉电
CPU	-	-	-
RTC	X	X ^(1,2)	X ⁽²⁾
WDT	X	X	X
BOD	X	X	X
EVSYS	X	X	X
CCL	X	X ⁽¹⁾	-
ACn	X	X ⁽¹⁾	-
ADCn/PTC ⁽³⁾	X	X ⁽¹⁾	-
DACn	X	X ⁽¹⁾	-
TCBn	X	X ⁽¹⁾	-
所有其他外设	X	-	-

注：

- 要使外设 in 待机休眠模式下运行，必须将相应外设的 RUNSTDBY 位置 1。
- 在待机休眠模式下，只有 RTC 功能需要将 RUNSTDBY 位置 1。
在掉电休眠模式下，只有 PIT 功能可用。

表 10-2. 时钟源的休眠模式活动概览

时钟源	可在休眠模式下工作		
	空闲	待机	掉电
主时钟源	X	X ⁽¹⁾	—

表 10-2. 时钟源的休眠模式活动概览（续）

时钟源	可在休眠模式下工作		
	空闲	待机	掉电
RTC 时钟源	X	X ^(1,2)	X ⁽²⁾
WDT 振荡器	X	X	X
BOD 振荡器 ⁽³⁾	X	X	X
CCL 时钟源	X	X ⁽¹⁾	—
TCD 时钟源	X	—	—

注：

1. 要使用时钟源在待机休眠模式下运行，必须将请求外设的 RUNSTDBY 位置 1。
2. 在待机休眠模式下，只有 RTC 功能需要将 RUNSTDBY 位置 1。
在掉电休眠模式下，只有 PIT 功能可用。
3. 仅限采样模式。

表 10-3. 休眠模式唤醒源

唤醒源	可在休眠模式下工作		
	空闲	待机	掉电
PORT 引脚中断	X	X	X ⁽¹⁾
BOD VLM 中断	X	X	X
RTC 中断	X	X ^(2,3)	X ⁽³⁾
TWIn 地址匹配中断	X	X	X
USARTn 帧起始中断	—	X	—
ADCn/PTC ⁽⁴⁾ 中断	X	X	—
ACn 中断	X	X ⁽⁵⁾	—
TCBn 捕捉中断	X	X	—
所有其他中断	X	—	—

注：

1. 必须根据 PORT 一章中的 *异步检测引脚属性* 一节配置 I/O 引脚。
2. 要使外设处于待机休眠模式下运行，必须将相应外设的 RUNSTDBY 位置 1。
3. 在待机休眠模式下，只有 RTC 功能需要将 RUNSTDBY 位置 1。
在掉电休眠模式下，只有 PIT 功能可用。
4. 当 RUNSTDY 位置 1 时，AC 将在不更新其状态寄存器或触发中断的情况下运行。如果其他外设已请求 CLK_PER，AC 将使用该时钟更新状态寄存器并触发中断。

10.3.2.2. 唤醒时间

器件的常规唤醒时间为六个主时钟周期（CLK_PER）加上启动主时钟源和稳压器（如果已关闭）所需的时间：

- 在空闲休眠模式下，主时钟源保持运行，因此无需额外的唤醒时间
- 在待机休眠模式下，主时钟可能运行，具体取决于外设置
- 在掉电休眠模式下，只有内部 32.768 kHz 振荡器和实时时钟（RTC）时钟源可能运行，供欠压检测器（BOD）、看门狗定时器（WDT）或周期性中断定时器（PIT）使用。所有其他时钟源均将关闭。

表 10-4. 休眠模式和启动时间

休眠模式	启动时间
空闲	6 个时钟周期

表 10-4. 休眠模式和启动时间（续）

休眠模式	启动时间
待机	6 个时钟周期 + 1 个 OSC 启动时间
掉电	6 个时钟周期 + 1 个 OSC 启动时间

CLKCTRL——时钟控制器部分介绍了不同时钟源的启动时间。

除了常规唤醒时间之外，还可使器件等待至 BOD 就绪后再执行代码。这可以通过向 BOD 配置（FUSE.BODCFG）熔丝中的工作模式和空闲模式下的 BOD 工作模式（ACTIVE）位域写入 0x3 来完成。如果 BOD 在常规唤醒时间之前就绪，则总唤醒时间将相同。如果 BOD 需要的时间超过常规唤醒时间，则唤醒时间将延长至 BOD 就绪。这样，无论何时执行代码都能确保提供正确的电源电压。

10.3.3. 调试操作

在运行时调试期间，该外设将继续正常工作。SLPCTRL 仅受调试操作中的中断影响：如果发生中断时 SLPCTRL 处于休眠模式，则器件将唤醒并且 SLPCTRL 将进入工作模式，即使没有待处理的中断请求也是如此。

如果外设配置为需要由 CPU 通过中断或类似操作进行定期控制，则在调试期间停止 CPU 可能会产生不正确的操作或丢失数据。

10.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0						SMODE[1:0]		SEN

10.5. 寄存器说明

10.5.1. 控制 A

名称: CTRLA
偏移量: 0x00
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
						SMODE[1:0]		SEN
访问	R	R	R	R	R	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 2:1 - SMODE[1:0] 休眠模式

通过写入这些位，可选择当休眠使能（SEN）位写入 1 且执行 SLEEP 指令时将进入的休眠模式。

值	名称	说明
0x0	IDLE	使能空闲休眠模式
0x1	STANDBY	使能待机休眠模式
0x2	PDOWN	使能掉电休眠模式
其他	-	保留

Bit 0 - SEN 休眠使能

在执行 SLEEP 指令使 MCU 进入所选休眠模式之前，该位必须写入 1。

11. RSTCTRL——复位控制器

11.1. 特性

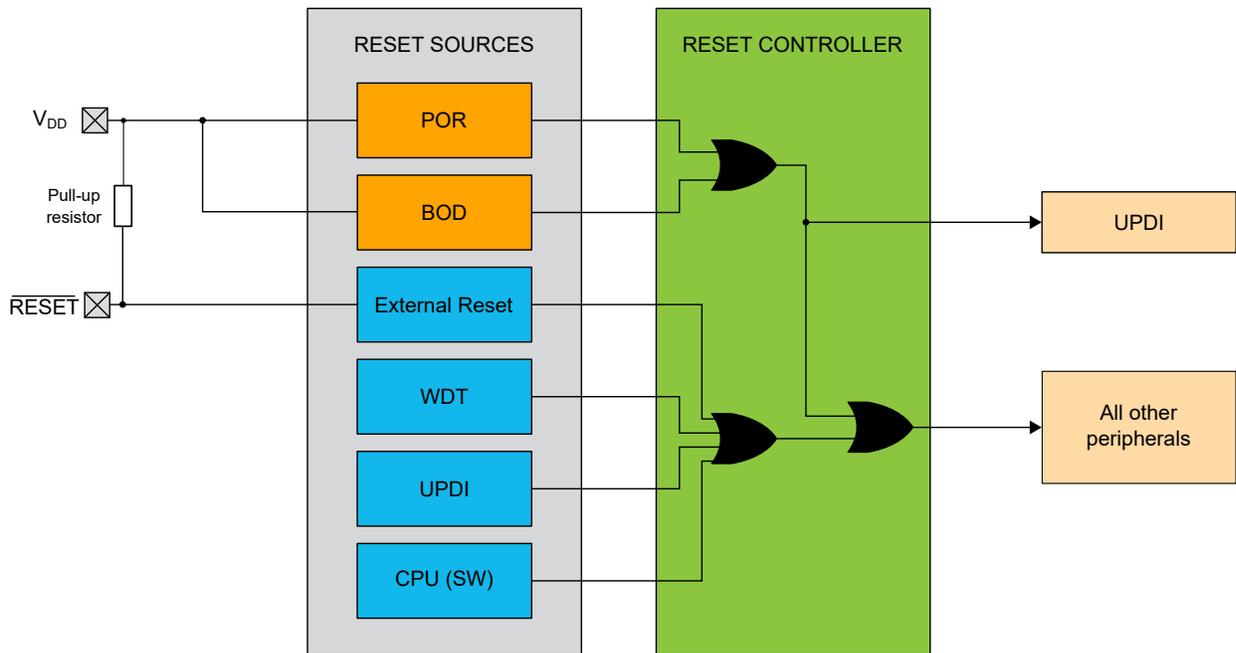
- 复位后将器件返回至初始状态
- 指示前一个复位源
- 电源复位源：
 - 上电复位（POR）
 - 欠压检测器（BOD）复位
- 用户复位源：
 - 外部复位（ $\overline{\text{RESET}}$ ）
 - 看门狗定时器（WDT）复位
 - 软件复位（Software Reset, SWRST）
 - 统一编程和调试接口（UPDI）复位

11.2. 概述

复位控制器（RSTCTRL）用于管理器件的复位。该控制器会发出器件复位信号、将器件设置为初始状态，并允许通过软件识别复位源。

11.2.1. 框图

图 11-1. 复位系统概览



11.2.2. 信号说明

信号	说明	类型
RESET	外部复位（低电平有效）	数字输入

11.3. 功能说明

11.3.1. 初始化

RSTCTRL 始终使能，但某些复位源必须先单独使能（通过熔丝或软件）才能请求复位。

通过任何复位源复位后，通过熔丝或签名行自动装入的器件中的寄存器都将更新。

11.3.2. 操作

11.3.2.1. 复位源

发生任何复位后，复位标志（RSTCTRL.RSTFR）寄存器中都可找到导致复位的复位源。用户可以通过在软件应用程序中读取该寄存器来识别先前的复位源。

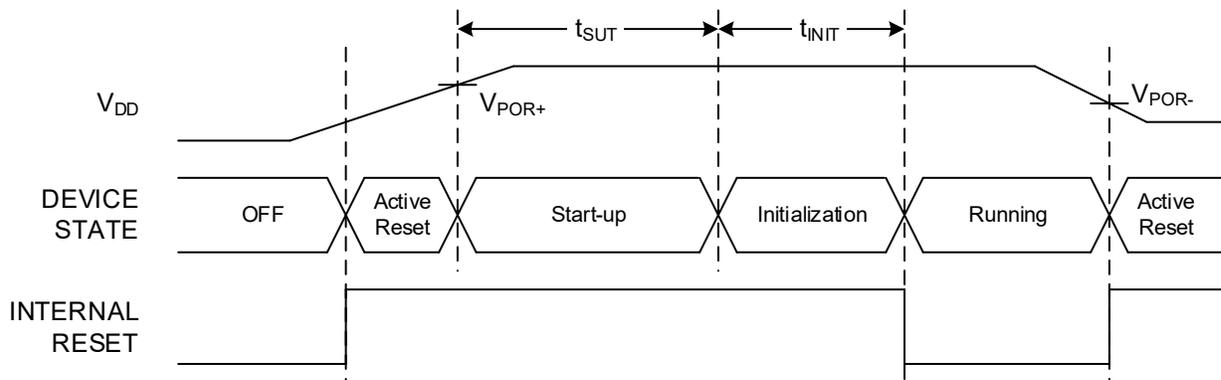
基于复位源的复位有以下两种：

- 电源复位源：
 - 上电复位（POR）
 - 欠压检测器（BOD）复位
- 用户复位源：
 - 外部复位（ $\overline{\text{RESET}}$ ）
 - 看门狗定时器（Watchdog Timer, WDT）复位
 - 软件复位（SWRST）
 - 统一编程和调试接口（UPDI）复位

11.3.2.1.1. 上电复位（POR）

上电复位（POR）旨在确保逻辑和存储器安全启动。该复位由片上检测电路生成，并且始终处于使能状态。当 V_{DD} 上升时，POR 激活，只要 V_{DD} 低于 POR 阈值电压（ $V_{\text{POR}+}$ ）就会产生有效复位。复位将持续到启动和复位初始化序列完成为止。启动时间（Start-Up Time, SUT）由熔丝决定。当 V_{DD} 降至检测电压（ $V_{\text{POR}-}$ ）以下时，复位会立即再次激活，没有任何延时。

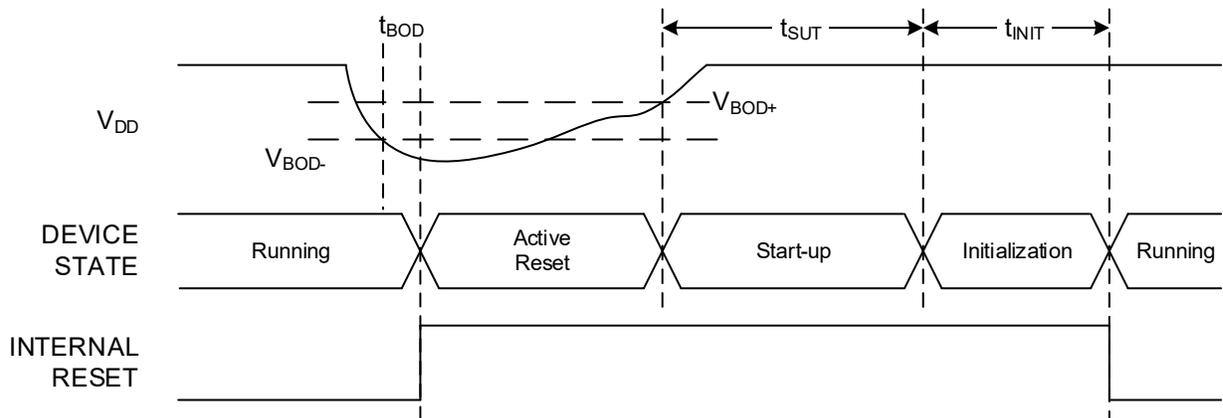
图 11-2. MCU 启动， $\overline{\text{RESET}}$ 连接至 V_{DD}



11.3.2.1.2. 欠压检测器（BOD）复位

片上欠压检测器（BOD）电路将监视工作期间的 V_{DD} ，实现方法是将该电压与固定触发电压进行比较。可以通过熔丝选择 BOD 的触发电压。如果应用中未使用 BOD，则会强制设为最低电压，以确保在内部复位和全片擦除期间安全工作。

图 11-3. 欠压检测器复位

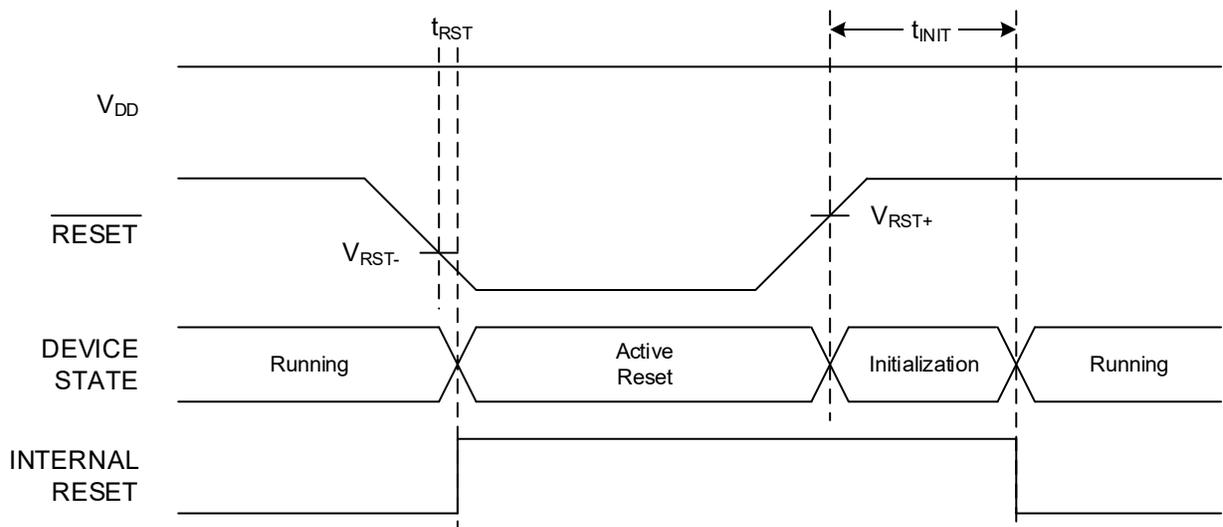


11.3.2.1.3. 外部复位

外部复位通过熔丝使能，请参见 FUSE.SYSCFG0 中的 RSTPINCFG 位域。

使能后，只要 $\overline{\text{RESET}}$ 引脚为低电平，外部复位便会请求复位。器件将保持复位状态，直至 $\overline{\text{RESET}}$ 再次变为高电平。

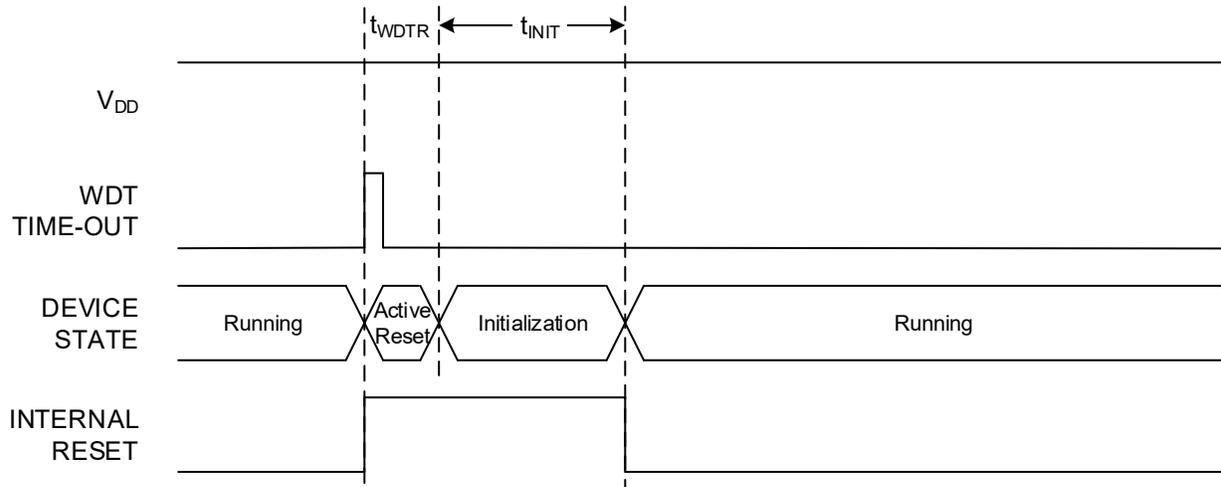
图 11-4. 外部复位特性



11.3.2.1.4. 看门狗复位

看门狗定时器 (WDT) 是一种用于监视程序是否正常运行的系统功能。如果 WDT 未在编程的超时周期内通过软件复位，则将发起看门狗复位。更多详细信息，请参见 WDT——看门狗定时器一章。

图 11-5. 看门狗复位



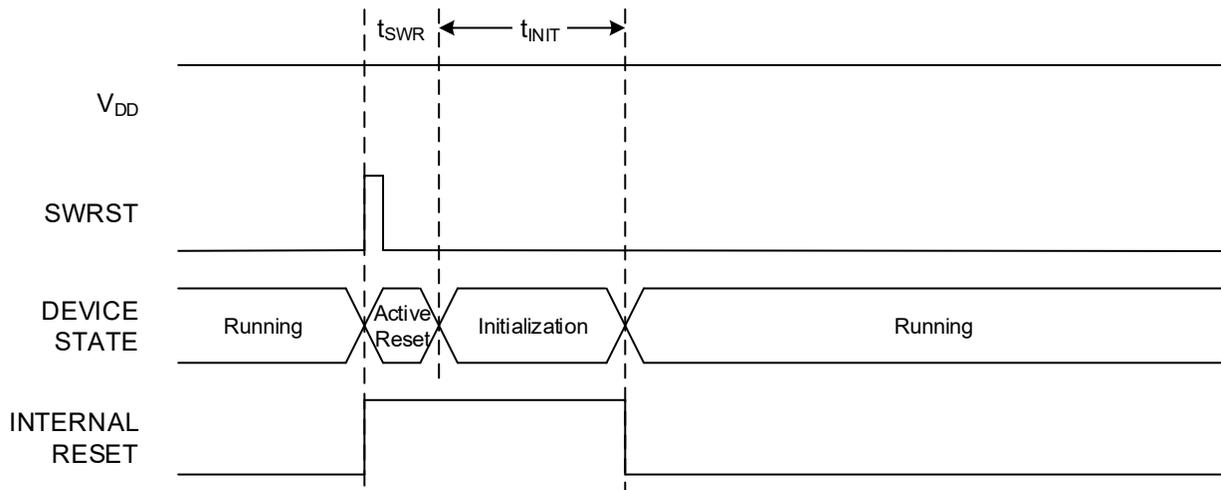
注：时间 t_{WDTR} 约为 50 ns。

11.3.2.1.5. 软件复位

通过软件发起系统复位。通过向软件复位（RSTCTRL.SWRR）寄存器中的软件复位使能（SWRE）位写入 1 来产生复位。

写入此位后将立即进行复位，器件将保持复位状态，直至复位序列完成。

图 11-6. 软件复位



注：时间 t_{SWR} 约为 50 ns。

11.3.2.1.6. 统一编程和调试接口（UPDI）复位

统一编程和调试接口（UPDI）包含一个单独的复位源，用于在外部编程和调试期间复位器件。复位源只能从外部调试器和编程器访问。有关更多详细信息，请参见 *UPDI——统一编程和调试接口* 一章。

11.3.2.1.7. 受复位影响的域

以下逻辑域受各种复位的影响：

表 11-1. 受各种复位影响的逻辑域

复位类型	重新装载熔丝	TCD 引脚改写功能可用	复位 TCD 引脚改写设置	复位 UPDI	复位其他易失性逻辑
POR	X		X	X	X
BOD	X	X		X	X
软件复位	X	X			X
外部复位	X	X			X
看门狗复位	X	X			X
UPDI 复位	X	X			X

11.3.2.2. 复位时间

复位时间可以分为两部分。

第一部分是当任何复位源工作时。这部分取决于复位源的输入。只要 $\overline{\text{RESET}}$ 引脚为低电平，外部复位便会工作。只要电源电压低于复位源阈值，上电复位（POR）和欠压检测器（BOD）便会工作。

第二部分是当所有复位源均释放且器件的内部复位初始化完成时。当电源复位源引起复位时，该时间将随着系统配置 1（FUSE.SYSCFG1）熔丝中的启动时间设置（SUT）位域给出的启动时间而增加。如果循环冗余校验存储器扫描（CRCSCAN）配置为在启动时运行，则内部复位初始化时间也会增加。该配置可以通过系统配置 0（FUSE.SYSCFG0）熔丝中的 CRC 源（CRCSRC）位域进行更改。

11.3.3. 休眠模式操作

RSTCTRL 在工作模式和所有休眠模式下工作。

11.3.4. 配置更改保护

此外设的一些寄存器具有配置更改保护（CCP）。要写入这些寄存器，必须先将特定密钥写入 CPU.CCP 寄存器，然后在 4 条 CPU 指令内对受保护位进行写访问。

必须遵循相应的 CCP 解锁序列，否则无法更改受保护的寄存器。

以下寄存器具有 CCP：

表 11-2. RSTCTRL——受配置更改保护影响的寄存器

寄存器	密钥
RSTCTRL.SWRR	IOREG

11.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	RSTFR	7:0			UPDIRF	SWRF	WDRF	EXTRF	BORF	PORF
0x01	SWRR	7:0								SWRE

11.5. 寄存器说明

11.5.1. 复位标志寄存器

名称: RSTFR
偏移量: 0x00
复位: 0xXX
属性: -

所有标志均可通过写入 1 的方式清零，也可通过上电复位（POR）清零，但上电复位标志（Power-On Reset Flag, PORF）除外。

位	7	6	5	4	3	2	1	0
访问			R/W	R/W	R/W	R/W	R/W	R/W
复位			x	x	x	x	x	x

Bit 5 - UPDIRF UPDI 复位标志

如果发生 UPDI 复位，则该位置 1。

Bit 4 - SWRF 软件复位标志

如果发生软件复位，则该位置 1。

Bit 3 - WDRF 看门狗复位标志

如果发生看门狗复位，则该位置 1。

Bit 2 - EXTRF 外部复位标志

如果发生外部复位，则该位置 1。

Bit 1 - BORF 欠压复位标志

如果发生欠压复位，则该位置 1。

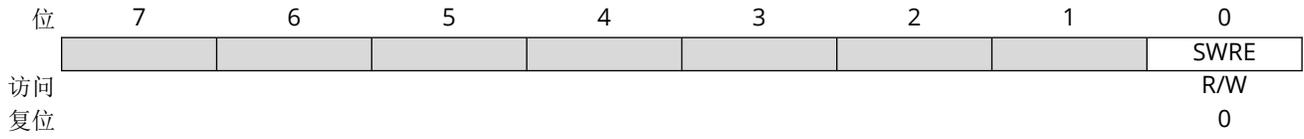
Bit 0 - PORF 上电复位标志

如果发生 POR，则该位置 1。

在 POR 之后，仅 POR 标志置 1，所有其他标志均清零。在 POR 之后，任何其他标志均无法在运行完整系统引导之前置 1。

11.5.2. 软件复位寄存器

名称: SWRR
偏移量: 0x01
复位: 0x00
属性: 配置更改保护



Bit 0 - SWRE 软件复位使能
该位写入 1 时，将发生软件复位。
该位将始终读为 0。

12. CPUINT——CPU 中断控制器

12.1. 特性

- 中断响应时间短且可预测
- 每个中断有单独的中断配置和向量地址
- 按优先级和向量地址对中断进行优先级排序
- 关键功能的不可屏蔽中断（Non-Maskable Interrupt, NMI）
- 两个中断优先级：0（正常优先级）和 1（高优先级）：
 - 可选择将其中一个中断请求的优先级指定为 1
 - 可选择为优先级为 0 的中断使能循环优先级方案
- 可选择将中断向量置于应用程序段或自举程序段中
- 可选的紧凑向量表（Compact Vector Table, CVT）

12.2. 概述

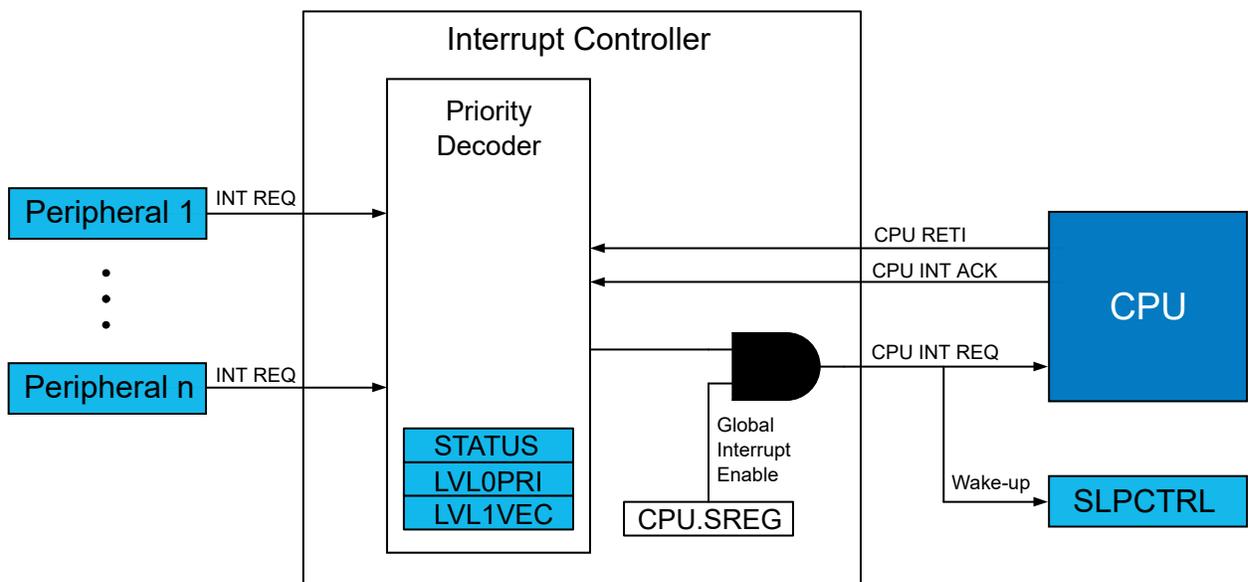
中断请求指示外设内部的状态变化，可用于改变程序执行。外设可具有一个或多个中断。所有中断均单独允许和配置。允许并配置中断后，出现中断条件时便会生成中断请求。

CPU 中断控制器（CPUINT）处理中断请求并确定其优先级。当允许中断且发生中断条件时，CPUINT 将接收中断请求。根据中断的优先级和任何进行中的中断的优先级，中断请求将得到应答，或者保持待处理状态直至优先级较高时得到应答。从中断处理程序返回后，程序将从发生中断之前所处的位置继续执行，并且在处理任何待处理的中断之前都会执行一条指令。

CPUINT 提供关键功能的 NMI、一个可选的高优先级中断和一个针对正常优先级中断的可选循环调度方案。该循环调度过程可确保在一定时间内处理所有中断。

12.2.1. 框图

图 12-1. CPUINT 框图



12.3. 功能说明

12.3.1. 初始化

必须按以下顺序初始化中断：

1. 如果默认配置不足，则配置 CPUINT（可选）：
 - 对控制 A（CPUINT.CTRLA）寄存器中的相应位（IVSEL 和 CVT）进行写操作，可配置向量处理。
 - 向 CPUINT.CTRLA 中的轮转优先级使能（LVLORR）位写入 1，以使能向量轮转优先级处理方案。
 - 选择优先级为 1 的向量，方法是将中断向量编号写入优先级为 1 的中断向量（CPUINT.LVL1VEC）寄存器。
2. 配置外设内的中断条件，并允许外设的中断。
3. 向 CPU 状态（CPU.SREG）寄存器中的全局中断允许（I）位写入 1，全局允许中断。

12.3.2. 工作模式

12.3.2.1. 使能、禁止和复位

向 CPU 状态（CPU.SREG）寄存器中的全局中断允许位（I）写入 1，全局允许中断。要全局禁止中断，请向 CPU.SREG 中的 I 位写入 0。

此外，还必须通过写入外设的中断控制（*peripheral.INTCTRL*）寄存器在相应的外设中允许所需的中断线。

中断执行完毕后，中断标志不会自动清零。相应的 INTFLAGS 寄存器说明提供了有关如何清零特定标志的信息。

12.3.2.2. 中断向量位置

中断向量的放置取决于控制 A（CPUINT.CTRLA）寄存器中的中断向量选择（IVSEL）位的值。有关可能位置的信息，请参见 CPUINT.CTRLA 中的 IVSEL 说明。

如果程序始终不允许中断源，则不使用中断向量，并且可以在这些位置放置常规程序代码。

12.3.2.3. 中断响应时间

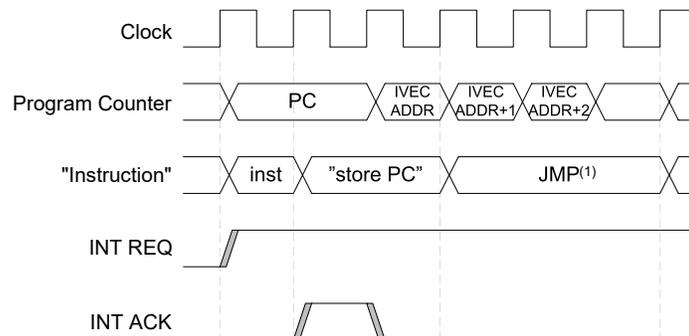
最小中断响应时间如下表所示。

表 12-1. 最小中断响应时间

	闪存大小 > 8 KB	闪存大小 ≤ 8 KB
完成正在进行的指令	一个周期	一个周期
将 PC 存储到堆栈	两个周期	两个周期
跳转到中断处理程序	三个周期（jmp）	两个周期（rjmp）

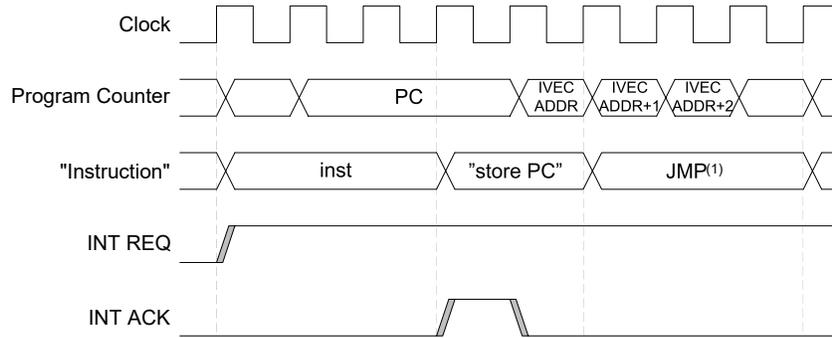
将程序计数器压入堆栈后，将执行中断的程序向量。请参见下图。

图 12-2. 单周期指令的中断执行



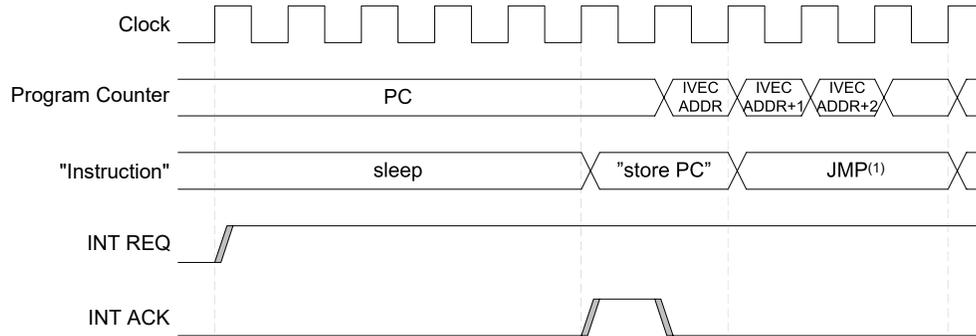
如果在执行多周期指令期间发生中断，则先完成指令再处理中断，如下图所示。

图 12-3. 多周期指令的中断执行



如果在器件处于休眠模式时发生中断，则中断执行响应时间会增加 5 个时钟周期，如下图所示。此外，从所选休眠模式启动的时间也会使响应时间有所增加。

图 12-4. 从休眠模式执行中断



从中断处理程序返回需要 4 到 5 个时钟周期，具体取决于程序计数器的大小。在这些时钟周期内，程序计数器从堆栈中弹出，堆栈指针将递增。

注：

1. 闪存大小不超过 8 KB 的器件使用 RJMP（而非 JMP），仅需两个时钟周期。

12.3.2.4. 中断优先级

所有中断向量都会被指定三种优先级中的一种，如下表所示。来自高优先级源的中断请求将中断来自正常优先级源的任何正在进行的中断处理程序。从高优先级中断处理程序返回时，将恢复执行正常优先级中断处理程序。

表 12-2. 中断优先级

优先级	级别	源
最高	不可屏蔽中断	与器件相关，静态分配
...	级别 1（高优先级）	可由用户选作级别 1 的向量
最低	级别 0（正常优先级）	其余中断向量

12.3.2.4.1. 不可屏蔽中断

无论 CPU.SREG 中的 I 位如何设置，都将执行不可屏蔽中断（NMI）。NMI 永远不会更改 I 位。其他中断均无法中断 NMI 处理程序。如果同时请求多个 NMI，则优先级为静态，具体取决于中断向量地址，地址越低，优先级越高。

哪些中断是不可屏蔽中断取决于器件，不受配置限制。必须先允许不可屏蔽中断才能使用。有关可用 NMI 源的信息，请参见器件的 *中断向量映射表*。

12.3.2.4.2. 高优先级中断

通过将中断向量编号写入 CPUINT.LVL1VEC 寄存器，可以将相应的中断请求指定为级别 1（高优先级）。该中断请求的优先级将高于其他（正常优先级）中断请求。优先级 1 中断将中断级别 0 中断处理程序。

12.3.2.4.3. 正常优先级中断

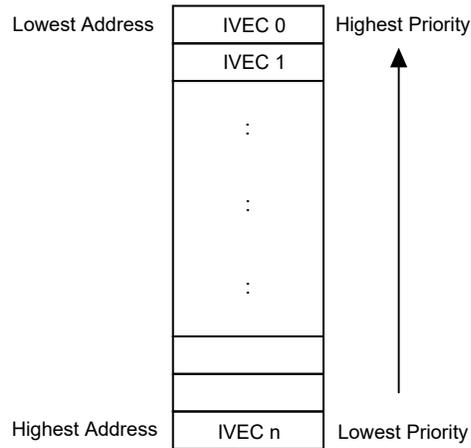
默认情况下，除 NMI 之外的所有中断向量均被指定为优先级 0（正常）。用户可以通过将这些向量之一指定为高优先级向量来进行覆盖。器件将有許多正常优先级向量，其中一些可能同时处于待处理状态。有两种不同的调度方案可用于选择首先处理哪个待处理的正常优先级中断：静态或轮转。

IVEC 是中断向量映射，具体在 *外设和架构* 一章中列出。以下部分使用 IVEC 来解释调度方案。IVEC0 是复位向量，IVEC1 是 NMI 向量，依此类推。在具有 n+1 个元素的向量表中，向量编号最高的向量用 IVECn 表示。复位、不可屏蔽中断和高级中断包含在 IVEC 映射中，但始终优先于正常优先级中断。

静态调度

如果多个级别 0 中断请求同时处于待处理状态，则会安排先执行优先级最高的中断请求。下图给出了默认配置，其中中断向量的地址越低，优先级越高。

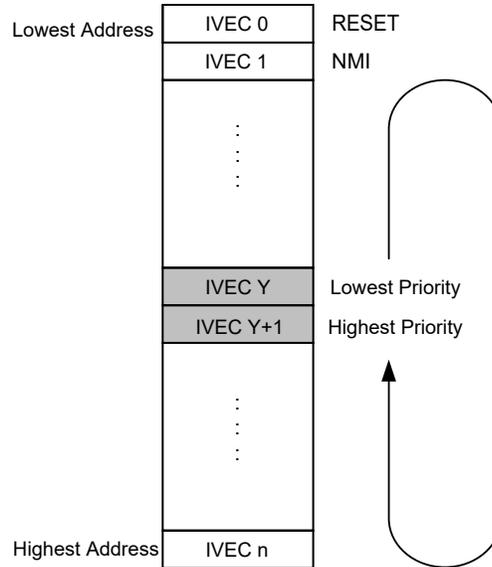
图 12-5. 默认静态调度



修改后的静态调度

可以通过向 CPUINT.LVL0PRI 寄存器写入向量编号来更改默认优先级。该向量编号将被指定为最低优先级。IVEC 中下一个中断向量的优先级在所有 LVLO 中断之中最高，如下图所示。

图 12-6. CPUINT.LVLOPRI 不为零时的静态调度



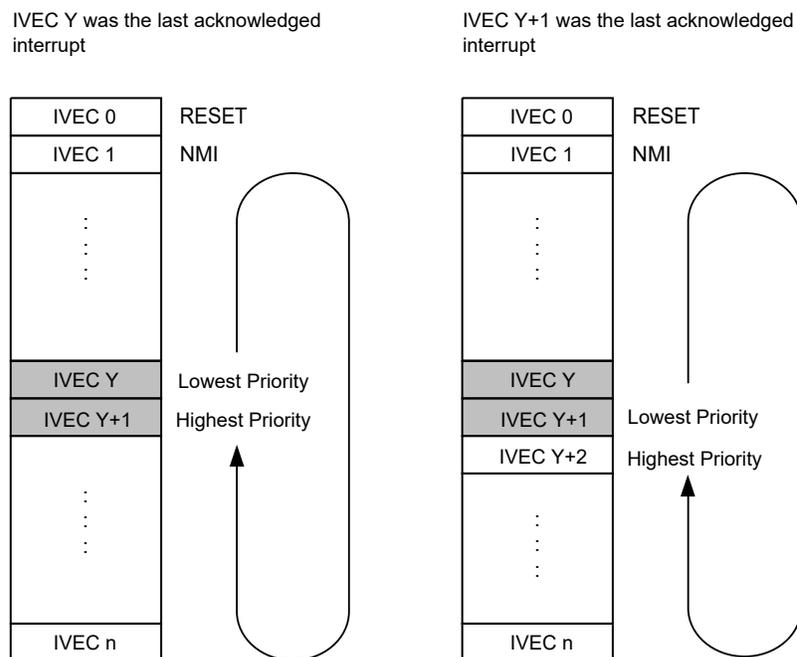
此处，值 Y 已写入 CPUINT.LVLOPRI，因此中断向量 Y+1 的优先级最高。请注意，在这种情况下，优先级将绕回，因此最低地址不再具有最高优先级。这包括 RESET 和 NMI，它们始终具有最高优先级。

有关可用中断请求及其中断向量编号，请参见器件的中断向量映射。

循环调度

静态调度可能会阻止处理某些中断请求。为避免出现这种情况，CPUINT 为正常优先级（LVLO）中断提供了循环调度。在循环调度中，CPUINT.LVLOPRI 寄存器存储最后确认的中断向量编号。该寄存器确保最后确认的中断向量获得最低优先级，并由硬件自动更新。下图给出了确认 IVEC Y 后和确认 IVEC Y+1 后的优先级顺序。

图 12-7. 循环调度



通过向控制 A (CPUINT.CTRLA) 寄存器中的循环优先级使能 (LVL0RR) 位写入 1, 可启用 LVL0 中断请求的循环调度。

12.3.2.5. 精简向量表

精简向量表 (CVT) 功能允许所有级别 0 中断共用同一中断向量编号, 以便能够编写紧凑的代码。因此, 这些中断共用同一中断服务程序 (ISR)。这样能够减少中断处理程序的数量, 从而释放可用于存储应用程序代码的存储器。

通过向控制 A (CPUINT.CTRLA) 寄存器中的 CVT 位写入 1 来使能 CVT 后, 该向量表包含以下三个中断向量:

1. 向量地址为 1 的不可屏蔽中断 (NMI)。
2. 向量地址为 2 的优先级 1 (LVL1) 中断。
3. 向量地址为 3 的所有优先级 0 (LVL0) 中断。

该功能非常适合存储空间有限的器件和使用少量中断生成器的应用。

12.3.3. 调试操作

使用 1 级优先级中断时, 确保正确配置中断服务程序非常重要, 因为这可能会导致应用程序卡在具有 1 级优先级的中断循环中。

通过读取 CPUINT STATUS (CPUINT.STATUS) 寄存器, 可以查看应用程序是否执行了正确的 RETI (中断返回) 指令。CPUINT.STATUS 寄存器包含状态信息, 可确保 CPUINT 在中断处理程序结束后 RETI 指令执行时返回到正确的中断级别。从中断返回后, CPUINT 将返回到进入中断之前的状态。

12.3.4. 配置更改保护

此外设的一些寄存器具有配置更改保护 (CCP)。要写入这些寄存器, 必须先将特定密钥写入 CPU.CCP 寄存器, 然后在 4 条 CPU 指令内对受保护位进行写访问。

必须遵循相应的 CCP 解锁序列, 否则无法更改受保护的寄存器。

以下寄存器具有 CCP:

表 12-3. CPUINT——受配置更改保护影响的寄存器

寄存器	密钥
CPUINT.CTRLA 中的 IVSEL	IOREG
CPUINT.CTRLA 中的 CVT	IOREG

12.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0		IVSEL	CVT					LVLORR
0x01	STATUS	7:0	NMIEX						LVL1EX	LVL0EX
0x02	LVL0PRI	7:0	LVL0PRI[7:0]							
0x03	LVL1VEC	7:0	LVL1VEC[7:0]							

12.5. 寄存器说明

12.5.1. 控制 A

名称: CTRLA
 偏移量: 0x00
 复位: 0x00
 属性: 配置更改保护

位	7	6	5	4	3	2	1	0
访问		IVSEL	CVT					LVLORR
复位		R/W	R/W					R/W
		0	0					0

Bit 6 - IVSEL 中断向量选择

此位受配置更改保护机制保护。

值	说明
0	中断向量位于闪存 BOOT 段的后面 ⁽¹⁾
1	中断向量位于闪存 BOOT 段的开头

注:

- 当整个闪存配置为 BOOT 段时，将忽略该位。

Bit 5 - CVT 精简向量表

此位受配置更改保护机制保护。

值	说明
0	禁止精简向量表功能
1	使能精简向量表功能

Bit 0 - LVLORR 循环优先级使能

此位不受配置更改保护机制保护。

值	说明
0	优先级固定为优先级 0 中断请求：最低中断向量地址具有最高优先级。
1	为优先级 0 中断请求使能循环优先级方案

12.5.2. 状态

名称: STATUS
 偏移量: 0x01
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	NMIEX						LVL1EX	LVL0EX
访问	R						R	R
复位	0						0	0

Bit 7 - NMIEX 正在执行不可屏蔽中断

如果正在执行不可屏蔽中断，则该标志置 1。从中断处理程序返回（RETI）时，该标志清零。

Bit 1 - LVL1EX 正在执行优先级 1 中断

执行优先级 1 中断或中断处理程序被 NMI 中断时，该标志置 1。从中断处理程序返回（RETI）时，该标志清零。

Bit 0 - LVL0EX 正在执行优先级 0 中断

执行优先级 0 中断或中断处理程序被优先级 1 中断或 NMI 中断时，该标志置 1。从中断处理程序返回（RETI）时，该标志清零。

12.5.3. 中断优先级 0

名称: LVL0PRI
 偏移量: 0x02
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	LVL0PRI[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - LVL0PRI[7:0] 中断优先级 0

该寄存器用于修改 LVLO 中断的优先级。有关更多信息，请参见[正常优先级中断](#)部分。

12.5.4. 优先级为 1 的中断向量

名称: LVL1VEC
 偏移量: 0x03
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	LVL1VEC[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - LVL1VEC[7:0] 优先级为 1 的中断向量

该位域包含优先级升为 1（LVL1）的单个向量的编号。如果该位域的值为 0x00，则没有向量具有 LVL1。因此，会禁止 LVL1 中断。

13. EVSYS——事件系统

13.1. 特性

- 用于在外设之间直接传输信号的系統
- 外设可以直接生成、使用和响应外设事件
- 响应时间短
- 最多可提供 4 个并行异步事件通道
- 最多可提供 2 个并行同步事件通道
- 可以将通道配置为具有一个触发外设操作和多个外设用户
- 外设可以直接触发和响应来自其他外设的事件
- 可以通过大多数外设以及软件发送和/或接收事件
- 在工作模式和待机休眠模式下工作

13.2. 概述

事件系统（EVSYS）支持在外设之间直接传输信号。一个外设（事件生成器）中的变化可通过事件系统的事件通道来触发其他外设（事件用户）中的操作，无需使用 CPU。该系统经过专门设计，可在外设之间提供短暂且可预测的响应时间，从而实现自主外设控制和交互，以及多个外设模块操作之间的时序同步。因此，该功能非常强大，可降低软件的复杂度和规模，并缩短执行时间。

事件生成器的状态变化被称为事件，通常与外设的其中一个中断条件相对应。事件可通过专用的事件路由网络直接转发到其他外设。每个通道的路由均在软件中配置，包括事件的生成和使用。

每个通道上只能路由一个来自事件生成器外设的触发信号，但多个通道可使用同一个生成器源。多个外设可以使用来自同一通道的事件。

通道路径既可与主时钟同步，也可与主时钟异步，具体取决于应用要求。

事件系统可直接连接模拟和数字转换器、模拟比较器、I/O 端口引脚、实时计数器、定时器/计数器以及可配置定制逻辑外设。此外，也可从软件和外设时钟生成事件。

13.2.1. 框图

图 13-1. 框图

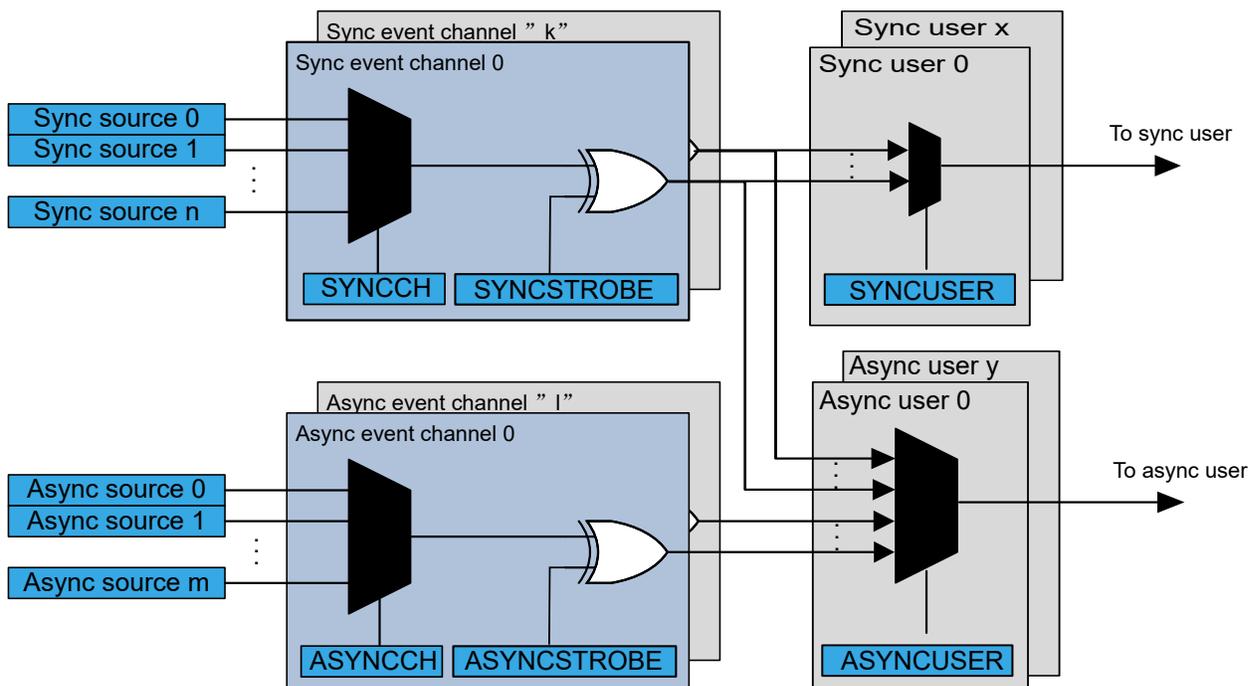
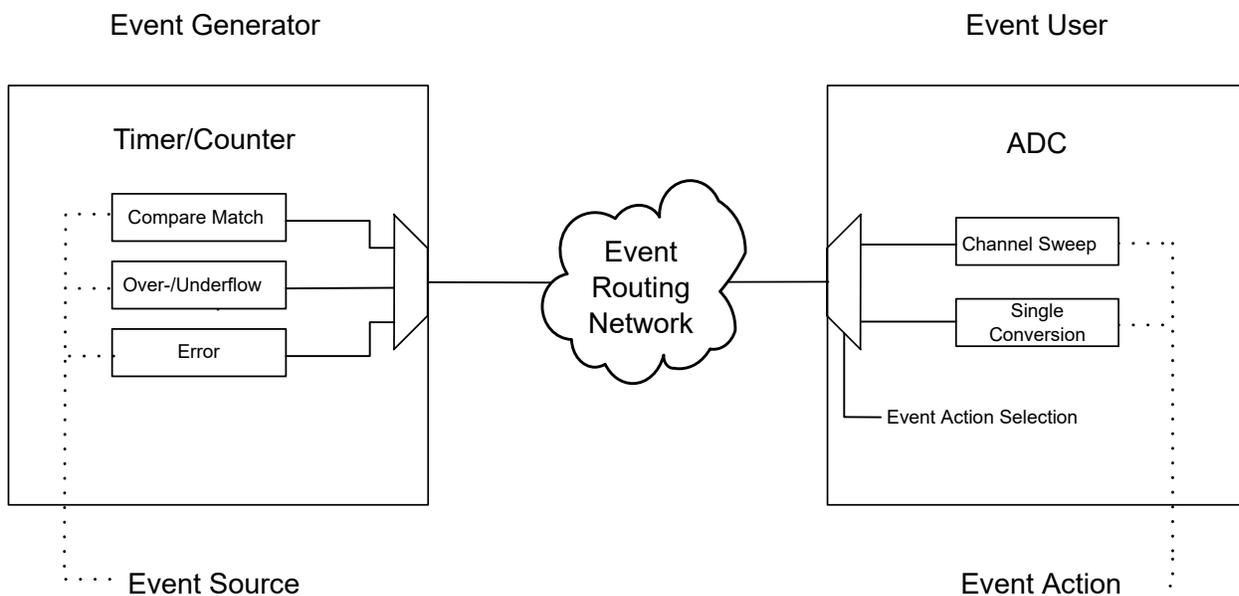


图 13-2. 事件源、生成器、用户和操作示例



注:

1. 有关支持事件的外设概述，请参见器件框图。
2. 有关事件生成器的列表，请参见通道 n 生成器选择（EVSYS.SYNCCH 和 EVSYS.ASYNCCH）寄存器。
3. 有关事件用户的列表，请参见用户通道 n 输入选择（EVSYS.SYNCUSER 和 EVSYS.ASYNCUSER）寄存器。

13.2.2. 信号说明

内部事件信号传输

事件信号传输可以与主时钟（CLK_MAIN）同步或异步发生。

根据基础事件的不同，事件信号可以是单个持续一个时钟周期的脉冲，或者是一个电平信号（类似于状态标志）。

引脚的事件输出

信号	类型	说明
EVOUT[2:0]	数字输出	事件输出

13.2.3. 系统相关性

要使用此外设，必须按如下所述正确配置系统的其他部分。

表 13-1. EVSYS 系统相关性

相关性	适用	外设
时钟	是	CLKCTRL
I/O 线路和连接	是	PORTMUX
中断	否	-
事件	是	EVSYS
调试	是	UPDI

13.2.3.1. 时钟

EVSYS 将外设时钟用于 I/O 寄存器和软件事件。正确设置时，路由网络也可用于休眠模式，而无需任何时钟。软件事件不会在外设时钟停止的休眠模式下工作。

13.2.3.2. I/O 线

EVSYS 可在引脚上异步输出三个事件通道。输出信号称为 EVOUT[2:0]。

1. 通过分别对 EVSYS.ASYNCUSER10、EVSYS.ASYNCUSER9 或 EVSYS.ASYNCUSER8 进行写操作，配置要输出的事件通道以及在哪个 EVOUTn 位上输出（SYNCCH[1:0]或 ASYNCCH[3:0]）。
2. 可选：使用端口外设配置引脚属性。
3. 向 PORTMUX 外设的控制 A（PORTMUX.CTRLA）寄存器中相应的 EVOUTn 位写入 1，使能引脚输出。

13.3. 功能说明

13.3.1. 初始化

在器件中使能事件之前，必须先配置事件用户多路开关和事件通道。

13.3.2. 工作原理

13.3.2.1. 事件用户多路开关设置

事件用户多路开关为事件用户选择通道。每个事件用户都有一个专用事件用户多路开关。每个多路开关连接到多个支持的事件通道输出，经配置后可选择其中一个通道。

支持异步事件的事件用户也支持同步事件。也有仅支持同步事件的事件用户。

通过对相应寄存器进行写操作来配置事件用户多路开关：

- 通过对相应的异步用户通道输入选择 n（EVSYS.ASYNCUSERn）寄存器进行写操作来配置同时支持同步事件和异步事件的事件用户。
- 通过对相应的同步用户通道输入选择 n（EVSYS.SYNCUSERn）寄存器进行写操作来配置仅支持同步事件的事件用户。

所有用户多路开关的默认设置均为关闭。

13.3.2.2. 事件系统通道

事件通道可连接到其中一个事件生成器。事件通道支持异步生成器或同步生成器。

通过对相应的异步通道 n 输入选择 (EVSYS.ASYNCCH n) 寄存器进行写操作来配置每个异步事件通道的事件源。

通过对相应的同步通道 n 输入选择 (EVSYS.SYNCCH n) 寄存器进行写操作来配置每个同步事件通道的事件源。

13.3.2.3. 事件生成器

每个事件通道均可从多个事件生成器接收事件。有关事件生成的详细信息，请参见相应外设的文档。

每个事件通道有多个可选的事件生成器，但每次只能选择其中一个。通过对相应的通道寄存器 (EVSYS.ASYNCCH n 和 EVSYS.SYNCCH n) 进行写操作来为每个通道选择事件生成器触发信号。默认情况下，事件通道不连接任何事件生成器。

13.3.2.4. 软件事件

在软件事件中，CPU 将通过反转当前值并持续一个系统时钟周期来“选通”事件通道。

通过向相应通道选通寄存器中的相应选通位写入 1，在通道上触发软件事件：

- 通过向异步通道选通 (EVSYS.ASYNCSTROBE) 寄存器中的 ASYNCSTROBE[l]位写入 1，可启动异步通道 l 上的软件事件。
- 通过向同步通道选通 (EVSYS.SYNCSTROBE) 寄存器中的 SYNCSTROBE[k]位写入 1，可启动同步通道 k 上的软件事件。

对于事件用户而言，软件事件与事件生成器外设生成的事件没有区别：向相应位写入 1 时，将在相应通道上生成事件，并由事件用户接收和处理。

13.3.3. 休眠模式操作

经过配置后，事件系统将在所有休眠模式下工作，但需要系统时钟的软件事件除外。

13.3.4. 调试操作

进入调试模式不会影响此外设。

13.3.5. 同步

异步事件由与之兼容的事件用户进行同步和处理。与异步事件不兼容的事件用户外设只能配置为侦听同步事件通道。

13.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	ASYNCSTROBE	7:0	ASYNCSTROBE[7:0]							
0x01	SYNCSTROBE	7:0	SYNCSTROBE[7:0]							
0x02	ASYNCCH0	7:0	ASYNCCH[7:0]							
0x03	ASYNCCH1	7:0	ASYNCCH[7:0]							
0x04	ASYNCCH2	7:0	ASYNCCH[7:0]							
0x05	ASYNCCH3	7:0	ASYNCCH[7:0]							
0x06	保留									
...										
0x09										
0x0A	SYNCCH0	7:0	SYNCCH[7:0]							
0x0B	SYNCCH1	7:0	SYNCCH[7:0]							
0x0C	保留									
...										
0x11										
0x12	ASYNCUSER0	7:0	ASYNCUSER[7:0]							
...										
0x1E	ASYNCUSER12	7:0	ASYNCUSER[7:0]							
0x1F	保留									
...										
0x21										
0x22	SYNCUSER0	7:0	SYNCUSER[7:0]							
0x23	SYNCUSER1	7:0	SYNCUSER[7:0]							

13.5. 寄存器说明

13.5.1. 异步通道选通

名称: ASYNCSTROBE
偏移量: 0x00
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	ASYNCSTROBE[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - ASYNCSTROBE[7:0] 异步通道选通

如果写入选通寄存器存储单元，则每个事件通道将被反转并持续一个系统时钟周期（即，生成单个事件）。

13.5.2. 同步通道选通

名称: SYNCSTROBE
偏移量: 0x01
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	SYNCSTROBE[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - SYNCSTROBE[7:0] 同步通道选通

如果写入选通寄存器存储单元，则每个事件通道将被反转并持续一个系统时钟周期（即，生成单个事件）。

13.5.3. 异步通道 n 生成器选择

名称: ASYNCCHn
偏移量: 0x02 + n*0x01 [n=0..3]
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	ASYNCCH[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - ASYNCCH[7:0] 异步通道生成器选择

值	ASYNCCH0	ASYNCCH1	ASYNCCH2	ASYNCCH3
0x00	关闭	关闭	关闭	关闭
0x01			CCL_LUT0	
0x02			CCL_LUT1	
0x03			AC0_OUT	
0x04			TCD0_CMPBCLR	
0x05			TCD0_CMPASET	
0x06			TCD0_CMPBSET	
0x07			TCD0_PROGEV	
0x08			RTC_OVF	
0x09			RTC_CMP	
0x0A	PORTA_PIN0	PORTB_PIN0	PORTC_PIN0	PIT_DIV8192
0x0B	PORTA_PIN1	PORTB_PIN1	PORTC_PIN1	PIT_DIV4096
0x0C	PORTA_PIN2	PORTB_PIN2	PORTC_PIN2	PIT_DIV2048
0x0D	PORTA_PIN3	PORTB_PIN3	PORTC_PIN3	PIT_DIV1024
0x0E	PORTA_PIN4	PORTB_PIN4	PORTC_PIN4	PIT_DIV512
0x0F	PORTA_PIN5	PORTB_PIN5	PORTC_PIN5	PIT_DIV256
0x10	PORTA_PIN6	PORTB_PIN6	AC1_OUT ⁽¹⁾	PIT_DIV128
0x11	PORTA_PIN7	PORTB_PIN7	AC2_OUT ⁽¹⁾	PIT_DIV64
0x12	UPDI	AC1_OUT ⁽¹⁾	—	AC1_OUT ⁽¹⁾
0x13	AC1_OUT ⁽¹⁾	AC2_OUT ⁽¹⁾	—	AC2_OUT ⁽¹⁾
0x14	AC2_OUT ⁽¹⁾	—	—	—
其他	—	—	—	—

注:

1. 不适用于 CN0816。

注: 低引脚数器件上只提供部分端口引脚。有关详细信息，请查看引脚图和/或 I/O 复用表。

13.5.4. 同步通道 n 生成器选择

名称: SYNCCHn
偏移量: 0x0A + n*0x01 [n=0..1]
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	SYNCCH[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 – SYNCCH[7:0] 同步通道生成器选择

值	SYNCCH0	SYNCCH1
0x00		OFF
0x01		TCB0
0x02		TCA0_OVF_LUNF
0x03		TCA0_HUNF
0x04		TCA0_CMP0
0x05		TCA0_CMP1
0x06		TCA0_CMP2
0x07	PORTC_PIN0	—
0x08	PORTC_PIN1	PORTB_PIN0
0x09	PORTC_PIN2	PORTB_PIN1
0x0A	PORTC_PIN3	PORTB_PIN2
0x0B	PORTC_PIN4	PORTB_PIN3
0x0C	PORTC_PIN5	PORTB_PIN4
0x0D	PORTA_PIN0	PORTB_PIN5
0x0E	PORTA_PIN1	PORTB_PIN6
0x0F	PORTA_PIN2	PORTB_PIN7
0x10	PORTA_PIN3	TCB1 ⁽¹⁾
0x11	PORTA_PIN4	—
0x12	PORTA_PIN5	—
0x13	PORTA_PIN6	—
0x14	PORTA_PIN7	—
0x15	TCB1	—
其他	—	—

注:

1. 不适用于 CN0816。

注: 低引脚数器件上只提供部分端口引脚。有关详细信息，请查看引脚图和/或 I/O 复用表。

13.5.5. 异步用户通道 n 输入选择

名称: ASYNCUSERn
 偏移量: $0x12 + n*0x01$ [n=0..12]
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	ASYNCUSER[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 – ASYNCUSER[7:0] 异步用户通道选择

ASYNCUSERn	用户多路开关	说明
0	TCB0	定时器/计数器 B 0
1	ADC0	ADC 0
2	CCL_LUT0EV0	CCL LUT0 事件 0
3	CCL_LUT1EV0	CCL LUT1 事件 0
4	CCL_LUT0EV1	CCL LUT0 事件 1
5	CCL_LUT1EV1	CCL LUT1 事件 1
6	TCD0_EV0	定时器计数器 D 0 事件 0
7	TCD0_EV1	定时器计数器 D 0 事件 1
8	EVOU0	事件输出 0
9	EVOU1	事件输出 1
10	EVOU2	事件输出 2
11	TCB1 ⁽¹⁾	B 型定时器/计数器 1
12	ADC1 ⁽¹⁾	ADC 1

注:

1. 不适用于 CN0816。

值	名称
0x0	OFF
0x1	SYNCCH0
0x2	SYNCCH1
0x3	ASYNCCH0
0x4	ASYNCCH1
0x5	ASYNCCH2
0x6	ASYNCCH3
其他	—

13.5.6. 同步用户通道 n 输入选择

名称: SYNCUSERn
 偏移量: $0x22 + n*0x01$ [n=0..1]
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	SYNCUSER[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - SYNCUSER[7:0] 同步用户通道选择

SYNCUSERn	用户多路开关	说明
0	TCA0	定时器/计数器 A
1	USART0	USART

值	名称
0x0	OFF
0x1	SYNCCH0
0x2	SYNCCH1
其他	—

14. PORTMUX——端口多路开关

14.1. 概述

端口多路开关（PORTMUX）可用于使能或禁止引脚功能，或者在默认引脚位置和备用引脚位置之间切换，具体取决于实际的引脚和属性。有关详细信息，请参见 PORTMUX 寄存器映射。

有关可用引脚和功能，请参见 [I/O 复用和注意事项](#)。

14.2. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0			LUT1	LUT0		EVOUT2	EVOUT1	EVOUT0
0x01	CTRLB	7:0				TWI0		SPI0		USART0
0x02	CTRLC	7:0			TCA05	TCA04	TCA03	TCA02	TCA01	TCA00
0x03	CTRLD	7:0							TCB1	TCB0

14.3. 寄存器说明

14.3.1. 控制 A

名称: CTRLA
偏移量: 0x00
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
			LUT1	LUT0		EVOUT2	EVOUT1	EVOUT0
访问			R/W	R/W		R/W	R/W	R/W
复位			0	0		0	0	0

Bit 5 - LUT1 CCL LUT 1 输出

向该位写入 1 可为 CCL LUT 1 选择备用引脚位置。

Bit 4 - LUT0 CCL LUT 0 输出

向该位写入 1 可为 CCL LUT 0 选择备用引脚位置。

Bit 2 - EVOUT2 事件输出 2

向该位写入 1 可使能事件输出 2。

Bit 1 - EVOUT1 事件输出 1

向该位写入 1 可使能事件输出 1。

Bit 0 - EVOUT0 事件输出 0

向该位写入 1 可使能事件输出 0。

14.3.2. 控制 B

名称: CTRLB
偏移量: 0x01
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
访问				TWI0		SPI0		USART0
复位				R/W		R/W		R/W
				0		0		0

Bit 4 - TWI0 TWI 0 通信

向该位写入 1 可为 TWI 0 选择备用通信引脚。

Bit 2 - SPI0 SPI 0 通信

向该位写入 1 可为 SPI 0 选择备用通信引脚。

Bit 0 - USART0 USART 0 通信

向该位写入 1 可为 USART 0 选择备用通信引脚。

14.3.3. 控制 C

名称: CTRLC
偏移量: 0x02
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
访问			TCA05	TCA04	TCA03	TCA02	TCA01	TCA00
复位			R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0

Bit 5 - TCA05 TCA0 波形输出 5

向该位写入 1 可在拆分模式下为 TCA0 波形输出 5 选择备用输出引脚。
在 TCA 处于正常模式时不适用。

Bit 4 - TCA04 TCA0 波形输出 4

向该位写入 1 可在拆分模式下为 TCA0 波形输出 4 选择备用输出引脚。
在 TCA 处于正常模式时不适用。

Bit 3 - TCA03 TCA0 波形输出 3

向该位写入 1 可在拆分模式下为 TCA0 波形输出 3 选择备用输出引脚。
在 TCA 处于正常模式时不适用。

Bit 2 - TCA02 TCA0 波形输出 2

向该位写入 1 可为 TCA0 波形输出 2 选择备用输出引脚。
在拆分模式下，该位控制低字节比较通道 2 的输出。

Bit 1 - TCA01 TCA0 波形输出 1

向该位写入 1 可为 TCA0 波形输出 1 选择备用输出引脚。
在拆分模式下，该位控制低字节比较通道 1 的输出。

Bit 0 - TCA00 TCA0 波形输出 0

向该位写入 1 可为 TCA0 波形输出 0 选择备用输出引脚。
在拆分模式下，该位控制低字节比较通道 0 的输出。

14.3.4. 控制 D

名称: CTRLD
偏移量: 0x03
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
							TCB1	TCB0
访问							R/W	R/W
复位							0	0

Bit 1 - TCB1 TCB1 输出

向该位写入 1 可为 16 位 B 类定时器/计数器 1 选择备用输出引脚。

注: CN0816 上未提供该位。

Bit 0 - TCB0 TCB0 输出

向该位写入 1 可为 16 位 B 类定时器/计数器 0 选择备用输出引脚。

15. PORT——I/O 引脚配置

15.1. 特性

- 支持独立配置的通用输入和输出引脚：
 - 上拉
 - 反相 I/O
- 中断和事件：
 - 检测两个边沿
 - 检测上升沿
 - 检测下降沿
 - 检测低电平
- 异步引脚变化检测，可将器件从所有休眠模式中唤醒
- 高效安全地访问端口引脚
 - 通过专用翻转/清零/置 1 寄存器实现硬件读-修改-写（Read-Modify-Write, RMW）操作
 - 将常用的 PORT 寄存器映射到位可访问的 I/O 存储空间（虚拟端口）

15.2. 概述

器件的 I/O 引脚由端口外设寄存器的实例控制。每个端口实例最多包含 8 个 I/O 引脚。这些端口分别命名为 PORTA、PORTB 和 PORTC 等。有关哪个端口实例控制哪些引脚的信息，请参见 *I/O 复用和注意事项* 一章。有关端口实例和相应虚拟端口实例的基址，请参见 *外设和架构* 一章。

每个端口引脚在数据方向（PORTx.DIR）和数据输出值（PORTx.OUT）寄存器中都具有对应的位，分别用于将该引脚使能为输出和定义输出状态。例如，引脚 PA3 由 PORTA 实例的 DIR[3]和 OUT[3]控制。

端口引脚的输入值与外设时钟（CLK_PER）同步，然后可作为数据输入值（PORTx.IN）进行访问。无论引脚配置为输入还是输出，都可以读取该引脚的值。

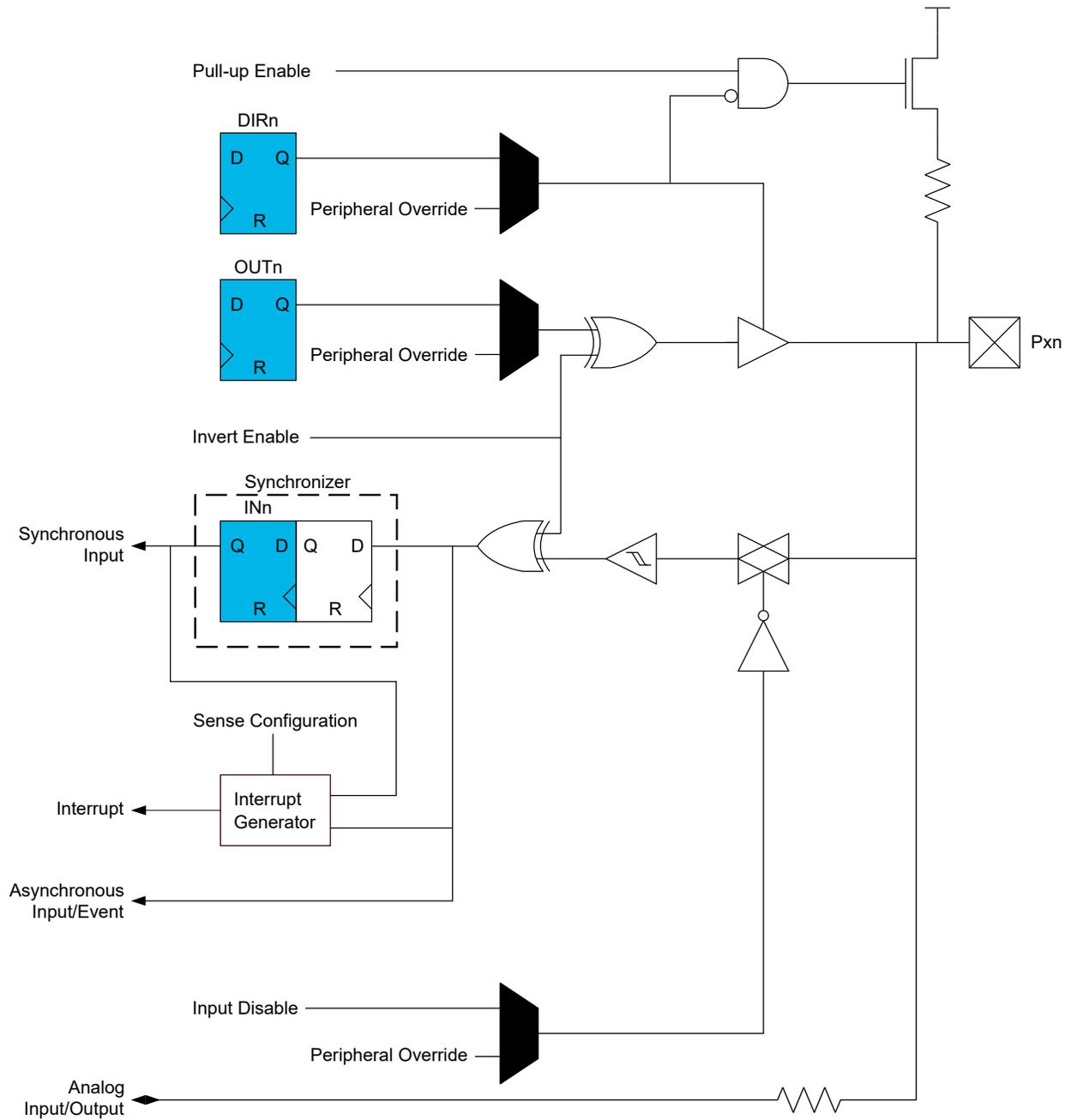
端口还支持异步输入检测，并且可在出现引脚变化条件（可选）时生成中断和事件。异步引脚变化检测是指引脚变化可触发中断并将器件从休眠模式唤醒，其中包括 CLK_PER 处于停止状态的休眠模式。

所有引脚功能均可按引脚单独配置。引脚具有硬件读-修改-写功能，可确保安全且正确地更改驱动值和/或输入和检测配置。

端口引脚配置用于控制其他器件功能的输入和输出选择。

15.2.1. 框图

图 15-1. PORT 框图



15.2.2. 信号说明

信号	类型	说明
Pxn	I/O 引脚	PORTx 上的 I/O 引脚 n

15.3. 功能说明

15.3.1. 初始化

复位后，即使没有时钟运行，所有输出也均为三态，并会使能数字输入缓冲器。

初始化端口操作时，以下所有步骤均可供选择：

- 分别向 PORTx.DIRSET 或 PORTx.DIRCLR 寄存器中的 bit n 写入 1，以使能或禁止引脚 Pxn 的输出驱动器
- 向 PORTx.OUTSET 或 PORTx.OUTCLR 寄存器中的 bit n 位写入 1，可将引脚 Pxn 的输出驱动器分别设置为高电平或低电平
- 通过读取 PORTx.IN 寄存器中的 bit n 来读取引脚 Pxn 的输入
- 在 PORTx.PINnCTRL 中为引脚 Pxn 配置各个引脚配置和中断控制



重要：为了实现最低功耗，可禁止未用引脚以及用作模拟输入或输出的引脚的数字输入缓冲器。

对于特定引脚（例如用于连接调试器的引脚），可以根据其特殊功能的要求进行不同的配置。

15.3.2. 工作模式

15.3.2.1. 基本功能

每个引脚组 x 都有自己的一组 PORT 寄存器。I/O 引脚 Pxn 可由 PORTx 中的寄存器控制。

要将引脚 n 用作输出，请向 PORTx.DIR 寄存器的 bit n 写入 1。这可以通过向 PORTx.DIRSET 寄存器中的 bit n 写入 1 来实现，同时可避免干扰该组中其他引脚的配置。必须向 PORTx.OUT 寄存器中的 bit n 写入所需输出值。

同样，向 PORTx.OUTSET 位写入 1 会将 PORTx.OUT 寄存器中的相应位置 1。向 PORTx.OUTCLR 中的某个位写入 1 会将 PORTx.OUT 中的该位清 0。向 PORTx.OUTTGL 或 PORTx.IN 中的某个位写入 1 将翻转 PORTx.OUT 中的该位。

要将引脚 n 用作输入，必须向 PORTx.DIR 寄存器中的 bit n 写入 0，以禁止输出驱动器。这可以通过向 PORTx.DIRCLR 寄存器中的 bit n 写入 1 来实现，同时可避免干扰该组中其他引脚的配置。只要 ISC 位未设置为 INPUT_DISABLE，即可从 PORTx.IN 寄存器中的 bit n 读取输入值。

向 PORTx.DIRTGL 中的某个位写入 1 将翻转 PORTx.DIR 中的该位并切换相应引脚的方向。

15.3.2.2. 引脚配置

引脚 n 控制（PORTx.PINnCTRL）寄存器用于配置引脚的反相 I/O、上拉和输入检测。引脚 n 的控制寄存器位于字节地址 $+(PORTx, +(0x10, n))$ 。

通过向 PORTx.PINnCTRL 中的反相 I/O 使能位（INVEN）位写入 1，可将相应引脚 n 上的所有输入和输出反相。当 INVEN 为 1 时，PORTx.IN/OUT/OUTSET/OUTTGL 寄存器将对该引脚进行反相操作。

翻转 INVEN 位会在引脚上产生一个边沿，随后可被使用该引脚的所有外设检测到，如果已允许中断或使能事件，则还会检测到中断或事件。

通过向 PORTx.PINnCTRL 中的上拉使能（PULLUPEN）位写入 1 来使能引脚 n 的输入上拉。即使 PULLUPEN 为 1，当该引脚配置为输出时，上拉也会断开连接。

通过写入 PORTx.PINnCTRL 中的输入/检测配置（Input/Sense Configuration, ISC）位域，可以为引脚 n 允许引脚中断。有关更多详细信息，请参见 [中断](#)。

通过将 INPUT_DISABLE 设置写入 ISC，可以禁止引脚 n 的数字输入缓冲器。这样，当该引脚用作模拟输入时，可以降低功耗和噪声。当配置为 INPUT_DISABLE 时，由于输入同步器被禁止，PORTx.IN 中的 bit n 不会改变。

15.3.2.3. 虚拟端口

虚拟 PORT 寄存器通过单周期位访问将最常用的常规 PORT 寄存器映射到 I/O 寄存器空间。访问虚拟 PORT 寄存器的结果与访问常规寄存器的结果相同，但前者支持存储器特定的指令（例如位操作指令），这些指令不能在常规 PORT 寄存器所在的扩展 I/O 寄存器空间中使用。下表列出了 PORT 和 VPORT 寄存器之间的映射。

表 15-1. 虚拟端口映射

常规 PORT 寄存器	映射到虚拟 PORT 寄存器
PORTx.DIR	VPORTx.DIR
PORTx.OUT	VPORTx.OUT
PORTx.IN	VPORTx.IN
PORTx.INTFLAGS	VPORTx.INTFLAGS

15.3.2.4. 外设改写

USART、ADC 和定时器等外设可以连接到 I/O 引脚。此类外设通常有一个主要 I/O 引脚连接以及一个或多个（可选）备用 I/O 引脚连接，具体可通过 PORTMUX 或外设内部的多路开关进行选择。通过配置和使能此类外设，通常由 PORT 控制的通用 I/O 引脚行为将依赖于外设的方式改写。一些外设可能不会改写所有 PORT 寄存器，而是让 PORT 模块控制 I/O 引脚操作的某些方面。

有关外设改写的信息，请参见每个外设的说明。端口中未被外设改写的任何引脚都将作为通用 I/O 引脚工作。

15.3.3. 中断

表 15-2. 可用中断向量和中断源

名称	向量说明	条件
PORTx	端口中断	根据 PORTx.PINnCTRL 中的输入/检测配置（ISC）位的配置，PORTx.INTFLAGS 中的 INTn 置 1

每个端口引脚 n 都可以配置为中断源。可通过对 PORTx.PINnCTRL 中的 ISC 进行写操作来单独允许或禁止每个中断。

发生中断条件时，外设的中断标志寄存器（外设.INTFLAGS）中的相应中断标志将置 1。

当相应的中断源被允许且中断标志置 1 时，将生成中断请求。中断请求保持有效，直到中断标志清零。有关如何清零中断标志的详细信息，请参见外设的 INTFLAGS 寄存器。

设置或更改中断设置时，需考虑以下几点：

- 如果在更改 ISC 时所处的周期内反相 I/O 使能（INVEN）位翻转，则因反相切换所产生的边沿可能不会引发中断请求
- 如果在同步中断时通过写入 ISC 禁止了输入，即便使用不同的中断设置重新使能输入，仍可能会请求原中断
- 如果在同步中断时通过写入 ISC 更改了中断设置，则可能不会请求该中断

15.3.3.1. 异步检测引脚属性

所有 PORT 引脚都支持异步输入检测，并且可针对可选的引脚变化条件产生中断。完全异步引脚变化检测可触发中断并将器件从所有休眠模式（包括外设时钟（CLK_PER）停止运行的模式）唤醒，而部分异步引脚变化检测则受到下表所示的限制。有关哪些引脚支持完全异步引脚变化检测的更多详细信息，请参见 I/O 复用和注意事项一章。

表 15-3. 两种检测引脚的行为比较

属性	部分异步引脚	完全异步引脚
将器件从 CLK_PER 运行的休眠模式唤醒	通过所有中断检测配置	通过所有中断检测配置
将器件从 CLK_PER 停止运行的休眠模式唤醒	仅通过 BOTHEDGES 或 LEVEL 中断检测配置	
CLK_PER 运行时触发中断的最小脉冲宽度	至少一个 CLK_PER 周期	短于一个 CLK_PER 周期
CLK_PER 停止运行时触发中断的最小脉冲宽度	引脚值必须保持到 CLK_PER 重新启动 ⁽¹⁾	
中断“死区”	前一个中断后的三个 CLK_PER 周期内没有新中断	

注:

1. 如果使用部分异步输入引脚将器件从 CLK_PER 停止运行的休眠状态唤醒，所需电压必须保持足够长的时间，以便 MCU 完成唤醒过程以触发中断。如果所需电压消失，则 MCU 唤醒时可能不会产生任何中断。

15.3.4. 事件

PORT 可生成以下事件:

表 15-4. PORTx 中的事件生成器

生成器名称		说明	事件类型	生成时钟域	事件时长
外设	事件				
PORTx	PINn	引脚电平	电平	异步	由引脚电平给出

所有 PORT 引脚均是异步事件系统生成器。PORT 的事件生成器与器件中的 PORT 引脚一样多。如果使能数字输入缓冲器，则 PORT 的每个事件系统输出均是相应引脚上的值。如果禁止引脚输入缓冲器，则相应的事件系统输出为 0。

端口没有事件输入。有关事件类型和事件系统配置的更多详细信息，请参见 *事件系统 (EVSYS)* 一章。

15.3.5. 休眠模式操作

除中断和输入同步外，所有引脚配置均与休眠模式无关。所有引脚都可以将器件从休眠状态唤醒。更多信息，请参见端口中断部分。

连接到端口的外设会受休眠模式的影响，如相应外设的数据手册部分中所述。



重要： 端口将始终使用外设时钟 (CLK_PER)。当该时钟停止时，输入同步将暂停。

15.3.6. 调试操作

当 CPU 在调试模式下停止时，PORT 将继续正常工作。如果 PORT 配置为需要由 CPU 通过中断或类似操作进行定期控制，则调试期间可能会产生不正确的操作或丢失数据。

15.4. 寄存器汇总——PORTx

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	DIR	7:0	DIR[7:0]							
0x01	DIRSET	7:0	DIRSET[7:0]							
0x02	DIRCLR	7:0	DIRCLR[7:0]							
0x03	DIRTGL	7:0	DIRTGL[7:0]							
0x04	OUT	7:0	OUT[7:0]							
0x05	OUTSET	7:0	OUTSET[7:0]							
0x06	OUTCLR	7:0	OUTCLR[7:0]							
0x07	OUTTGL	7:0	OUTTGL[7:0]							
0x08	IN	7:0	IN[7:0]							
0x09	INTFLAGS	7:0	INT[7:0]							
0x0A	保留									
...										
0x0F										
0x10	PIN0CTRL	7:0	INVEN				PULLUPEN		ISC[2:0]	
0x11	PIN1CTRL	7:0	INVEN				PULLUPEN		ISC[2:0]	
0x12	PIN2CTRL	7:0	INVEN				PULLUPEN		ISC[2:0]	
0x13	PIN3CTRL	7:0	INVEN				PULLUPEN		ISC[2:0]	
0x14	PIN4CTRL	7:0	INVEN				PULLUPEN		ISC[2:0]	
0x15	PIN5CTRL	7:0	INVEN				PULLUPEN		ISC[2:0]	
0x16	PIN6CTRL	7:0	INVEN				PULLUPEN		ISC[2:0]	
0x17	PIN7CTRL	7:0	INVEN				PULLUPEN		ISC[2:0]	

15.5. 寄存器说明——PORTx

15.5.1. 数据方向

名称: DIR
偏移量: 0x00
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	DIR[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 – DIR[7:0] 数据方向

该位域控制每个 PORTx 引脚的输出驱动器。

该位域不控制数字输入缓冲器。可以在引脚 n 控制 (PORTx.PINnCTRL) 寄存器的输入/检测配置 (ISC) 位域中配置引脚 n (Pxn) 的数字输入缓冲器。该位域中每个 bit n 的可用配置如下表所示。

值	说明
0	Pxn 配置为仅输入引脚，禁止输出驱动器
1	Pxn 配置为输出引脚，使能输出驱动器

15.5.2. 数据方向置 1

名称: DIRSET
 偏移量: 0x01
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	DIRSET[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - DIRSET[7:0] 数据方向置 1

该位域控制每个 PORTx 引脚的输出驱动器，而无需使用读-修改-写操作。向该位域中的 bit n 写入 0 没有任何作用。向该位域中的 bit n 写入 1 会将 PORTx.DIR 中的相应位置 1，从而将引脚 n (Pxn) 配置为输出引脚并使能输出驱动器。读取该位域将返回 PORTx.DIR 的值。

15.5.3. 数据方向清零

名称: DIRCLR
 偏移量: 0x02
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	DIRCLR[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - DIRCLR[7:0] 数据方向清零

该位域控制每个 PORTx 引脚的输出驱动器，而无需使用读-修改-写操作。向该位域中的 bit n 写入 0 没有任何作用。向该位域中的 bit n 写入 1 会将 PORTx.DIR 中的相应位清零，从而将引脚 n (Pxn) 配置为仅输入引脚并禁止输出驱动器。读取该位域将返回 PORTx.DIR 的值。

15.5.4. 数据方向翻转

名称: DIRTGL
 偏移量: 0x03
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	DIRTGL[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - DIRTGL[7:0] 数据方向翻转

该位域控制每个 PORTx 引脚的输出驱动器，而无需使用读-修改-写操作。
 向该位域中的 bit n 写入 0 没有任何作用。向该位域中的 bit n 写入 1 会将 PORTx.DIR 中的相应位翻转。
 读取该位域将返回 PORTx.DIR 的值。

15.5.5. 输出值

名称: OUT
偏移量: 0x04
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	OUT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 – OUT[7:0] 输出值

该位域控制每个 PORTx 引脚的输出驱动器电平。

该配置仅在相应引脚使能输出驱动器 (PORTx.DIR) 时起作用。该位域中每个 bit n 的可用配置如下表所示。

值	说明
0	引脚 n (Pxn) 输出驱动为低电平
1	Pxn 输出驱动为高电平

15.5.6. 输出值置 1

名称: OUTSET
 偏移量: 0x05
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	OUTSET[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - OUTSET[7:0] 输出值置 1

该位域控制每个 PORTx 引脚的输出驱动器电平，而无需使用读-修改-写操作。向该位域中的 bit n 写入 0 没有任何作用。向该位域中的 bit n 写入 1 会将 PORTx.OUT 中的相应位置 1，从而将引脚 n (Pxn) 的输出配置为以高电平驱动。读取该位域将返回 PORTx.OUT 的值。

15.5.7. 输出值清零

名称: OUTCLR
 偏移量: 0x06
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	OUTCLR[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - OUTCLR[7:0] 输出值清零

该位域控制每个 PORTx 引脚的输出驱动器电平，而无需使用读-修改-写操作。向该位域中的 bit n 写入 0 没有任何作用。向该位域中的 bit n 写入 1 会将 PORTx.OUT 中的相应位清零，从而将引脚 n (Pxn) 的输出配置为以低电平驱动。读取该位域将返回 PORTx.OUT 的值。

15.5.8. 输出值翻转

名称: OUTTGL
 偏移量: 0x07
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	OUTTGL[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - OUTTGL[7:0] 输出值翻转

该位域控制每个 PORTx 引脚的输出驱动器电平，而无需使用读-修改-写操作。
 向该位域中的 bit n 写入 0 没有任何作用。向该位域中的 bit n 写入 1 会将 PORTx.OUT 中的相应位翻转。
 读取该位域将返回 PORTx.OUT 的值。

15.5.9. 输入值

名称: IN
偏移量: 0x08
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	IN[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - IN[7:0] 输入值

当使能数字输入缓冲器时，该位域显示 PORTx 引脚的状态。

向该位域中的 bit n 写入 0 没有任何作用。向该位域中的 bit n 写入 1 会将 PORTx.OUT 中的相应位翻转。

如果禁止数字输入缓冲器，则不会对输入进行采样，并且位值不会更改。

可以在引脚 n 控制 (PORTx.PINnCTRL) 寄存器的输入/检测配置 (ISC) 位域中配置引脚 n (Pxn) 的数字输入缓冲器。该位域中每个 bit n 的可用状态如下表所示。

值	说明
0	Pxn 为低电平
1	Pxn 为高电平

15.5.10. 中断标志

名称: INTFLAGS
 偏移量: 0x09
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	INT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - INT[7:0] 引脚中断标志

引脚中断标志 n 通过写入 1 清零。

当引脚 n (Pxn) 的变化或状态与 PORTx.PINnCTRL 中引脚的输入/检测配置 (ISC) 匹配时, 引脚中断标志 n 置 1。

向该位域中的 bit n 写入 0 没有任何作用。

向该位域中的 bit n 写入 1 会将引脚中断标志 n 清零。

15.5.11. 引脚 n 控制

名称: PINnCTRL
偏移量: $0x10 + n*0x01$ [$n=0..7$]
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	INVEN				PULLUPEN		ISC[2:0]	
访问	R/W				R/W	R/W	R/W	R/W
复位	0				0	0	0	0

Bit 7 - INVEN 反相 I/O 使能

该位控制引脚 n 的输入和输出是否反相。

值	说明
0	输入和输出值不反相
1	输入和输出值反相

Bit 3 - PULLUPEN 上拉使能

当引脚配置为仅输入时，该位控制是否使能引脚 n 的内部上拉。

值	说明
0	禁止上拉
1	使能上拉

Bit 2:0 - ISC[2:0] 输入/检测配置

该位域控制引脚 n 的输入和检测配置。检测配置决定如何触发端口中断。

值	名称	说明
0x0	INTDISABLE	禁止中断但使能输入缓冲器
0x1	BOTHEDGES	允许中断及双边沿检测
0x2	RISING	允许中断及上升沿检测
0x3	FALLING	允许中断及下降沿检测
0x4	INPUT_DISABLE	中断和数字输入缓冲器禁止 ⁽¹⁾
0x5	LEVEL	允许中断及低电平检测
其他	—	保留

注:

1. 如果禁止引脚 n 的数字输入缓冲器，则将不会更新输入值 (PORTx.IN) 寄存器中的 bit n。

15.6. 寄存器汇总——VPORTx

偏移量	名称	位位置	7	6	5	4	3	2	1	0	
0x00	DIR	7:0	DIR[7:0]								
0x01	OUT	7:0	OUT[7:0]								
0x02	IN	7:0	IN[7:0]								
0x03	INTFLAGS	7:0	INT[7:0]								

15.7. 寄存器说明——VPORTx

15.7.1. 数据方向

名称: DIR
 偏移量: 0x00
 复位: 0x00
 属性: -

访问虚拟 PORT 寄存器的结果与访问常规寄存器的结果相同，但前者支持存储器特定的指令（例如位操作指令），这些指令不能在常规 PORT 寄存器所在的扩展 I/O 寄存器空间中使用。

位	7	6	5	4	3	2	1	0
	DIR[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - DIR[7:0] 数据方向

该位域控制每个 PORTx 引脚的输出驱动器。

该位域不控制数字输入缓冲器。可以在引脚 n 控制（PORTx.PINnCTRL）寄存器的输入/检测配置（ISC）位域中配置引脚 n（Pxn）的数字输入缓冲器。该位域中每个 bit n 的可用配置如下表所示。

值	说明
0	Pxn 配置为仅输入引脚，禁止输出驱动器
1	Pxn 配置为输出引脚，使能输出驱动器

15.7.2. 输出值

名称: OUT
偏移量: 0x01
复位: 0x00
属性: -

访问虚拟 PORT 寄存器的结果与访问常规寄存器的结果相同，但前者支持存储器特定的指令（例如位操作指令），这些指令不能在常规 PORT 寄存器所在的扩展 I/O 寄存器空间中使用。

位	7	6	5	4	3	2	1	0
	OUT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - OUT[7:0] 输出值

该位域控制每个 PORTx 引脚的输出驱动器电平。

该配置仅在相应引脚使能输出驱动器（PORTx.DIR）时起作用。该位域中每个 bit n 的可用配置如下表所示。

值	说明
0	引脚 n (Pxn) 输出驱动为低电平
1	Pxn 输出驱动为高电平

15.7.3. 输入值

名称: IN
 偏移量: 0x02
 复位: 0x00
 属性: -

访问虚拟 PORT 寄存器的结果与访问常规寄存器的结果相同，但前者支持存储器特定的指令（例如位操作指令），这些指令不能在常规 PORT 寄存器所在的扩展 I/O 寄存器空间中使用。

位	7	6	5	4	3	2	1	0
	IN[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - IN[7:0] 输入值

当使能数字输入缓冲器时，该位域显示 PORTx 引脚的状态。

向该位域中的 bit n 写入 0 没有任何作用。向该位域中的 bit n 写入 1 会将 PORTx.OUT 中的相应位翻转。

如果禁止数字输入缓冲器，则不会对输入进行采样，并且位值不会更改。

可以在引脚 n 控制（PORTx.PINnCTRL）寄存器的输入/检测配置（ISC）位域中配置引脚 n（Pxn）的数字输入缓冲器。该位域中每个 bit n 的可用状态如下表所示。

值	说明
0	Pxn 为低电平
1	Pxn 为高电平

15.7.4. 中断标志

名称: INTFLAGS
 偏移量: 0x03
 复位: 0x00
 属性: -

访问虚拟 PORT 寄存器的结果与访问常规寄存器的结果相同，但前者支持存储器特定的指令（例如位操作指令），这些指令不能在常规 PORT 寄存器所在的扩展 I/O 寄存器空间中使用。

位	7	6	5	4	3	2	1	0
	INT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - INT[7:0] 引脚中断标志

引脚中断标志 n 通过写入 1 清零。

当引脚 n (Pxn) 的变化或状态与 PORTx.PINnCTRL 中引脚的输入/检测配置 (ISC) 匹配时，引脚中断标志 n 置 1。

向该位域中的 bit n 写入 0 没有任何作用。

向该位域中的 bit n 写入 1 会将引脚中断标志 n 清零。

16. BOD——欠压检测器

16.1. 特性

- 通过欠压检测器监视电源，防止其低于阈值电压（可编程）
- 三种可用模式：
 - 使能模式（持续工作）
 - 采样模式
 - 禁止模式
- 可分别在器件处于工作模式和休眠模式时选择 BOD 工作模式
- 具有中断功能的电压监视器（VLM）
- 可编程 VLM 电压（相对于 BOD 电压）

16.2. 概述

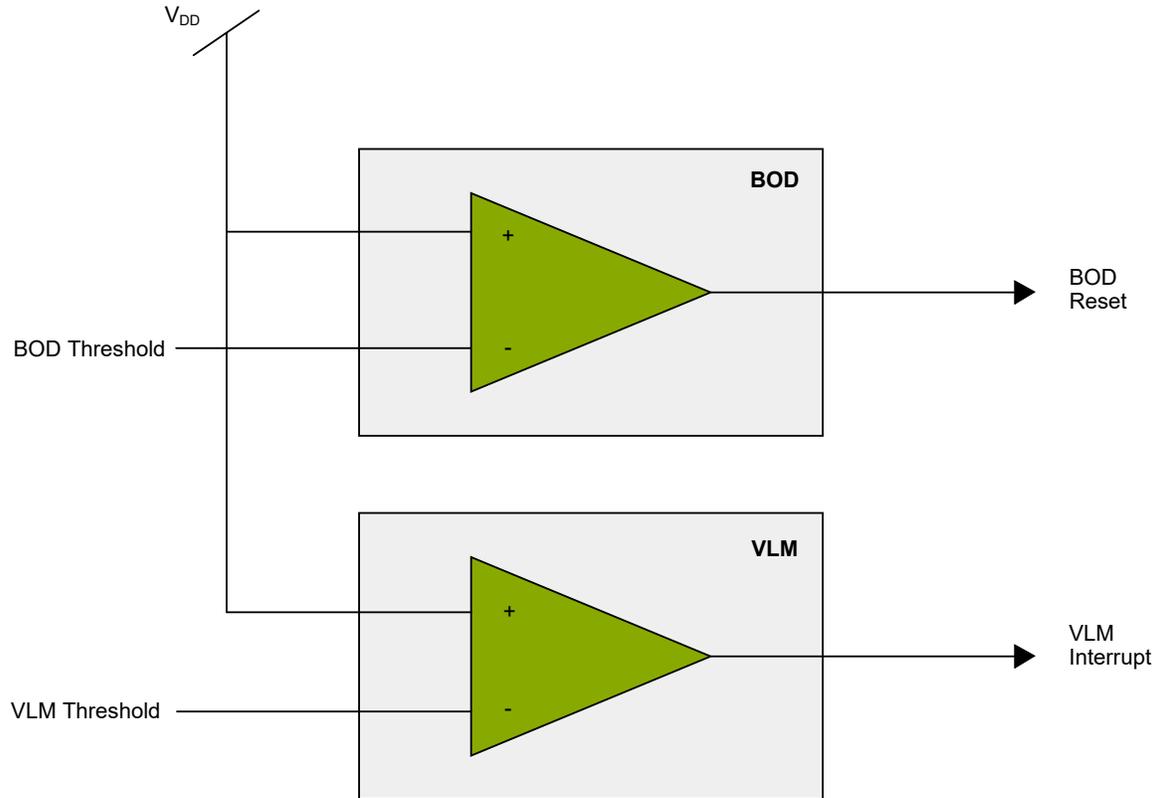
欠压检测器（BOD）用于监视电源并将电源电压与可编程欠压阈值电压进行比较。欠压阈值电压定义何时产生系统复位。电压监视器（VLM）用于监视电源并将其与高于 BOD 阈值的阈值进行比较。当电源电压接近 BOD 阈值时，VLM 可产生中断作为“预警信号”。VLM 阈值电压以超出 BOD 阈值电压的百分比的形式表示。

BOD 主要由熔丝控制，必须由用户使能。在正常程序执行期间，可更改在待机休眠模式和掉电休眠模式下使用的模式。VLM 还受到 I/O 寄存器控制。

激活后，BOD 既可工作在使能模式下（BOD 持续工作）；或者工作在采样模式下（BOD 在给定时间内短暂激活，以检查电源电压）。

16.2.1. 框图

图 16-1. BOD 框图



16.3. 功能说明

16.3.1. 初始化

BOD 设置在复位期间通过熔丝装入。工作模式和空闲休眠模式下的 BOD 电压和工作模式由熔丝设置，不可通过软件更改。待机模式和掉电休眠模式下的工作模式设置通过熔丝装入，可通过软件更改。

通过向中断控制（BOD.INTCTRL）寄存器中的 VLM 中断允许（VLMIE）位写入 1，可使能电压监视器功能。通过写入 BOD.INTCTRL 中的 VLM 配置（VLMCFG）位，可配置 VLM 中断。当电源电压从上方、下方或任何方向超过 VLM 阈值时，可请求中断。

VLM 功能将随 BOD 模式变化。如果禁止 BOD，即使 VLMIE 为 1，也不会使能 VLM。如果 BOD 使用采样模式，VLM 也将被采样。允许 VLM 中断时，如果电压超过 VLM 电压，将根据 VLMCFG 来置 1 中断标志。

通过写入控制 A（BOD.VLMCTRLA）寄存器中的 VLM 电压（VLMLVL）位，可定义 VLM 阈值。

16.3.2. 中断

表 16-1. 可用中断向量和中断源

名称	向量说明	条件
VLM	电压监视器	电源电压超过由中断控制（BOD.INTCTRL）寄存器中的 VLM 配置（VLMCFG）位域配置的 VLM 阈值

如果 CPU 在调试模式下停止，则不会执行 VLM 中断。

发生中断条件时，外设的中断标志（外设.INTFLAGS）寄存器中的相应中断标志将置 1。

可以通过写入外设的中断控制（外设.INTCTRL）寄存器中的相应允许位来允许或禁止中断源。

当相应的中断源被允许且中断标志置 1 时，将生成中断请求。中断请求保持有效，直到中断标志清零。有关如何清零中断标志的详细信息，请参见外设的 INTFLAGS 寄存器。

16.3.3. 休眠模式操作

不同休眠模式下的 BOD 配置由熔丝定义。在工作模式和空闲休眠模式下所使用的模式由 FUSE.BODCFG 中的 ACTIVE 熔丝定义，该模式会装入控制 A（BOD.CTRLA）寄存器中的 ACTIVE 位域。在待机休眠模式和掉电休眠模式下所使用的模式由 FUSE.BODCFG 中的 SLEEP 熔丝定义，该模式会装入控制 A（BOD.CTRLA）寄存器中的 SLEEP 位域。

工作模式和空闲休眠模式（即 BOD.CTRLA 中的 ACTIVE）下的工作模式无法通过软件更改。对控制 A（BOD.CTRLA）寄存器中的 SLEEP 位域进行写操作，可以更改在待机休眠模式和掉电休眠模式下的工作模式。

当器件进入待机或掉电休眠模式时，BOD 将根据 BOD.CTRLA 中 SLEEP 的定义更改工作模式。当器件从待机或掉电休眠模式中唤醒时，BOD 将在控制 A（BOD.CTRLA）寄存器中 ACTIVE 位域所定义的模式下工作。

16.3.4. 配置更改保护

此外设的一些寄存器具有配置更改保护（CCP）。要写入这些寄存器，必须先将特定密钥写入 CPU.CCP 寄存器，然后在 4 条 CPU 指令内对受保护位进行写访问。

必须遵循相应的 CCP 解锁序列，否则无法更改受保护的寄存器。

以下寄存器具有 CCP：

表 16-2. 配置更改保护下的寄存器

寄存器	密钥
BOD.CTRLA 中的 SLEEP	IOREG

16.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0				SAMPFREQ		ACTIVE[1:0]		SLEEP[1:0]
0x01	CTRLB	7:0								LVL[2:0]
0x02	保留									
...										
0x07										
0x08	VLMCTRLA	7:0								VLMLVL[1:0]
0x09	INTCTRL	7:0							VLMCFG[1:0]	VLMIE
0x0A	INTFLAGS	7:0								VLMIF
0x0B	STATUS	7:0								VLMS

16.5. 寄存器说明

16.5.1. 控制 A

名称: CTRLA
偏移量: 0x00
复位: 从熔丝装入
属性: 配置更改保护

位	7	6	5	4	3	2	1	0
				SAMPFREQ	ACTIVE[1:0]		SLEEP[1:0]	
访问				R	R	R	R/W	R/W
复位				x	x	x	x	x

Bit 4 – SAMPFREQ 采样频率

该位控制 BOD 采样频率。
从 FUSE.BODCFG 中的 SAMPFREQ 位装入复位值。
该位受配置更改保护（CCP）影响。

值	说明
0x0	采样频率为 1 kHz
0x1	采样频率为 125 Hz

Bit 3:2 – ACTIVE[1:0] 工作

在器件处于活动或空闲模式时，这些位选择 BOD 工作模式。
从 FUSE.BODCFG 中的 ACTIVE 位域装入复位值。
该位域不受配置更改保护（CCP）的控制。

值	名称	说明
0x0	DIS	禁止
0x1	ENABLED	在连续模式下使能
0x2	SAMPLED	在采样模式下使能
0x3	ENWAKE	在连续模式下使能。在唤醒时暂停执行，直到 BOD 运行

Bit 1:0 – SLEEP[1:0] 休眠

在器件处于待机或掉电休眠模式时，这些位选择 BOD 工作模式。从 FUSE.BODCFG 中的 SLEEP 位域装入复位值。

值	名称	说明
0x0	DIS	禁止
0x1	ENABLED	在连续模式下使能
0x2	SAMPLED	在采样模式下使能
0x3	-	保留

16.5.2. 控制 B

名称: CTRLB
偏移量: 0x01
复位: 从熔丝装入
属性: -

位	7	6	5	4	3	2	1	0
							LVL[2:0]	
访问	R	R	R	R	R	R	R	R
复位	0	0	0	0	0	x	x	x

Bit 2:0 - LVL[2:0] BOD 电压

该位域控制 BOD 阈值电压。

从 BOD 配置熔丝 (FUSE.BODCFG) 中的 BOD 电压 (LVL) 位装入复位值。

值	名称	说明
0x0	BODLEVEL0	1.8V
0x2	BODLEVEL2	2.6V
0x7	BODLEVEL7	4.2V

注:

- 更多详细信息, 请参见 *电气特性* 中的 *BOD* 和 *POR* 特性。
- 说明中的值为典型值

16.5.3. VLM 控制 A

名称: VLMCTRLA
偏移量: 0x08
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
							VLMLVL[1:0]	
访问							R/W	R/W
复位							0	0

Bit 1:0 - VLMLVL[1:0] VLM 电压

这些位选择相对于 BOD 阈值 (BOD.CTRLB 中的 LVL) 的 VLM 阈值。

值	说明
0x0	VLM 阈值比 BOD 阈值高 5%
0x1	VLM 阈值比 BOD 阈值高 15%
0x2	VLM 阈值比 BOD 阈值高 25%
其他	保留

16.5.4. 中断控制

名称: INTCTRL
偏移量: 0x09
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
						VLMCFG[1:0]		VLMIE
访问						R/W	R/W	R/W
复位						0	0	0

Bit 2:1 - VLMCFG[1:0] VLM 配置

这些位选择将触发 VLM 中断的事件。

值	名称	说明
0x0	BELOW	V _{DD} 降至 VLM 阈值以下
0x1	ABOVE	V _{DD} 升至 VLM 阈值以上
0x2	CROSS	V _{DD} 超过 VLM 阈值
其他	-	保留

Bit 0 - VLMIE VLM 中断允许

向该位写入 1 可允许 VLM 中断。

16.5.5. VLM 中断标志

名称: INTFLAGS
偏移量: 0x0A
复位: 0x00
属性: -



Bit 0 - VLMIF VLM 中断标志

出现来自 VLM 的触发信号时，根据 BOD.INTCTRL 寄存器中的 VLMCFG 位配置，该标志置 1。该标志仅在 BOD 使能时更新。

16.5.6. VLM 状态

名称: STATUS
偏移量: 0x0B
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
								VLMS
访问								R
复位								0

Bit 0 - VLMS VLM 状态

该位仅在 BOD 使能后才有效。

值	说明
0	电压高于 VLM 阈值
1	电压低于 VLM 阈值

17. VREF——参考电压

17.1. 特性

- 可编程参考电压源：
 - 每个 ADC 外设各有一个
 - 每个 AC 和 DAC 外设各有一个
- 每个参考源支持五种不同的电压：
 - 0.55V
 - 1.1V
 - 1.5V
 - 2.5V
 - 4.3V

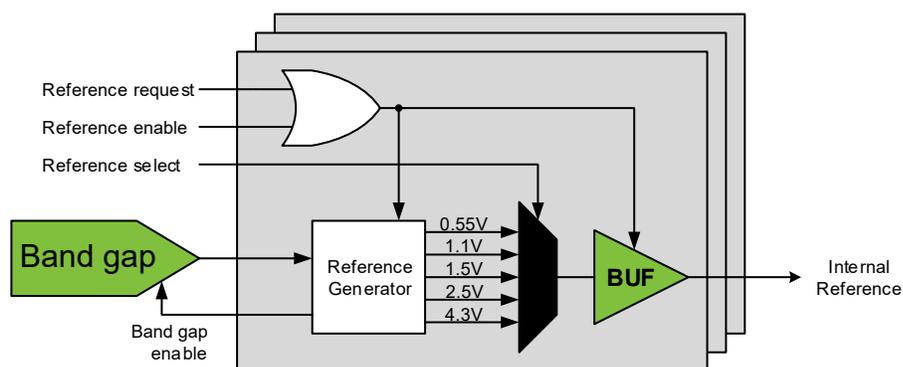
17.2. 概述

参考电压（VREF）外设为多个外设使用的参考电压源提供控制寄存器。用户可通过写入控制 x（VREF.CTRLx）寄存器中的 ADCn 参考选择（ADCnREFSEL）位域来选择 ADCn 的参考电压，通过写入控制 x（VREF.CTRLx）寄存器中的 DACn 参考选择（DACnREFSEL）位域来选择 ACn 和 DACn 的参考电压。

参考电压源可应外设请求自动使能。用户可以通过写入控制 B（VREF.CTRLB）寄存器中相应的强制使能（ADCnREFEN 和 DACnREFEN）位来使能参考电压源（从而取消自动禁止未使用的源）。这样可以缩短启动时间，但代价是功耗会增加。

17.2.1. 框图

图 17-1. VREF 框图



17.3. 功能说明

17.3.1. 初始化

当 ADCn、ACn 或 DACn 请求参考电压时，默认配置将使能相应的源。默认参考电压为 0.55V，但可通过写入控制（VREF.CTRLx）寄存器中相应的参考选择（ADCnREFSEL 和 DACnREFSEL）位域进行配置。

17.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0			ADCOREFSEL[2:0]				DACOREFSEL[2:0]	
0x01	CTRLB	7:0			DAC2REFEN	ADC1REFEN	DAC1REFEN		ADC0REFEN	DAC0REFEN
0x02	CTRLC	7:0			ADC1REFSEL[2:0]				DAC1REFSEL[2:0]	
0x03	CTRLD	7:0							DAC2REFSEL[2:0]	

17.5. 寄存器说明

17.5.1. 控制 A

名称: CTRLA
 偏移量: 0x00
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	ADCCOREFSEL[2:0]			DACOREFSEL[2:0]				
访问		R/W	R/W	R/W		R/W	R/W	R/W
复位		0	0	0		0	0	0

Bit 6:4 - ADCCOREFSEL[2:0] ADC0 参考电压选择

该位域用于选择 ADC0 的参考电压。

值	说明
0x0	0.55V
0x1	1.1V
0x2	2.5V
0x3	4.3V
0x4	1.5V
其他	保留

Bit 2:0 - DACOREFSEL[2:0] DAC0 和 AC0 参考电压选择

该位域用于选择 DAC0 和 AC0 的参考电压。

值	说明
0x0	0.55V
0x1	1.1V
0x2	2.5V
0x3	4.3V
0x4	1.5V
其他	保留

17.5.2. 控制 B

名称: CTRLB
偏移量: 0x01
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
访问			DAC2REFEN	ADC1REFEN	DAC1REFEN		ADC0REFEN	DAC0REFEN
复位			R/W	R/W	R/W		R/W	R/W
			0	0	0		0	0

Bit 5 - DAC2REFEN DAC2 和 AC2 参考电压强制使能

向该位写入 1 会强制使能 DAC2 和 AC2 的参考电压，即使未请求也会使能。

向该位写入 0 可在未请求时自动使能/禁止参考源。

注：该位不适用于 CN0816。

Bit 4 - ADC1REFEN ADC1 参考电压强制使能

向该位写入 1 会强制使能 ADC1 的参考电压，即使未请求也会使能。

向该位写入 0 可在未请求时自动使能/禁止参考源。

注：当 ADC 使用外部参考电压（ADC.CTRLA 中的 REFSEL 位域）时，不要强制使能内部参考电压（VREF.CTRLB 中的 ADCnREFEN = 1）。

注：该位不适用于 CN0816。

Bit 3 - DAC1REFEN DAC1 和 AC1 参考电压强制使能

向该位写入 1 会强制使能 DAC1 和 AC1 的参考电压，即使未请求也会使能。

向该位写入 0 可在未请求时自动使能/禁止参考源。

注：该位不适用于 CN0816。

Bit 1 - ADC0REFEN ADC0 参考电压强制使能

向该位写入 1 会强制使能 ADC0 的参考电压，即使未请求也会使能。

向该位写入 0 可由外设自动使能/禁止参考源。

注：当 ADC 使用外部参考电压（ADC.CTRLA 中的 REFSEL 位域）时，不要强制使能内部参考电压（VREF.CTRLB 中的 ADCnREFEN = 1）。

Bit 0 - DAC0REFEN DAC0 和 AC0 参考电压强制使能

向该位写入 1 可强制使能 DAC0 和 AC0 的参考电压，即使未请求也会使能。

向该位写入 0 可由外设自动使能/禁止参考源。

17.5.3. 控制 C

名称: CTRLC
 偏移量: 0x02
 复位: 0x00
 属性: -

注: CN0816 上未提供该寄存器

位	7	6	5	4	3	2	1	0
		ADC1REFSEL[2:0]				DAC1REFSEL[2:0]		
访问		R/W	R/W	R/W		R/W	R/W	R/W
复位		0	0	0		0	0	0

Bit 6:4 – ADC1REFSEL[2:0] ADC1 参考电压选择

该位域用于选择 ADC1 的参考电压。

值	说明
0x0	0.55V
0x1	1.1V
0x2	2.5V
0x3	4.3V
0x4	1.5V
其他	保留

Bit 2:0 – DAC1REFSEL[2:0] DAC1 和 AC1 参考电压选择

该位域用于选择 DAC1 和 AC1 的参考电压。

值	说明
0x0	0.55V
0x1	1.1V
0x2	2.5V
0x3	4.3V
0x4	1.5V
其他	保留

17.5.4. 控制 D

名称: CTRLD
 偏移量: 0x03
 复位: 0x00
 属性: -

注: CN0816 上未提供该寄存器。

位	7	6	5	4	3	2	1	0
						DAC2REFSEL[2:0]		
访问						R/W	R/W	R/W
复位						0	0	0

Bit 2:0 - DAC2REFSEL[2:0] DAC2 和 AC2 参考电压选择

该位域用于选择 DAC2 和 AC2 的参考电压。

值	说明
0x0	0.55V
0x1	1.1V
0x2	2.5V
0x3	4.3V
0x4	1.5V
其他	保留

18. WDT——看门狗定时器

18.1. 特性

- 如果看门狗定时器在超时周期结束之前未清零，则会发出系统复位信号
- 使用独立振荡器与系统时钟异步工作
- 使用 32.768 kHz 超低功耗振荡器（OSCULP32K）的 1.024 kHz 输出
- 11 种超时周期（8 ms 到 8s）可供选择
- 两种工作模式：
 - 正常模式
 - 窗口模式
- 可锁定配置，防止意外更改
- 可在第一条 WDT 指令后激活关闭周期定时器以便于设置

18.2. 概述

看门狗定时器（WDT）是一种用于监视程序是否正常运行的系统功能。它允许系统通过发出复位信号从失控或死锁代码等情况中恢复。使能后，WDT 是一个具有预定义超时周期的持续运行定时器。如果 WDT 没有在超时周期内复位，则将发出系统复位。通过在软件中执行看门狗定时器复位（WDR）指令来复位 WDT。

WDT 有两种工作模式：正常模式和窗口模式。控制 A（WDT.CTRLA）寄存器中的设置决定了工作模式。

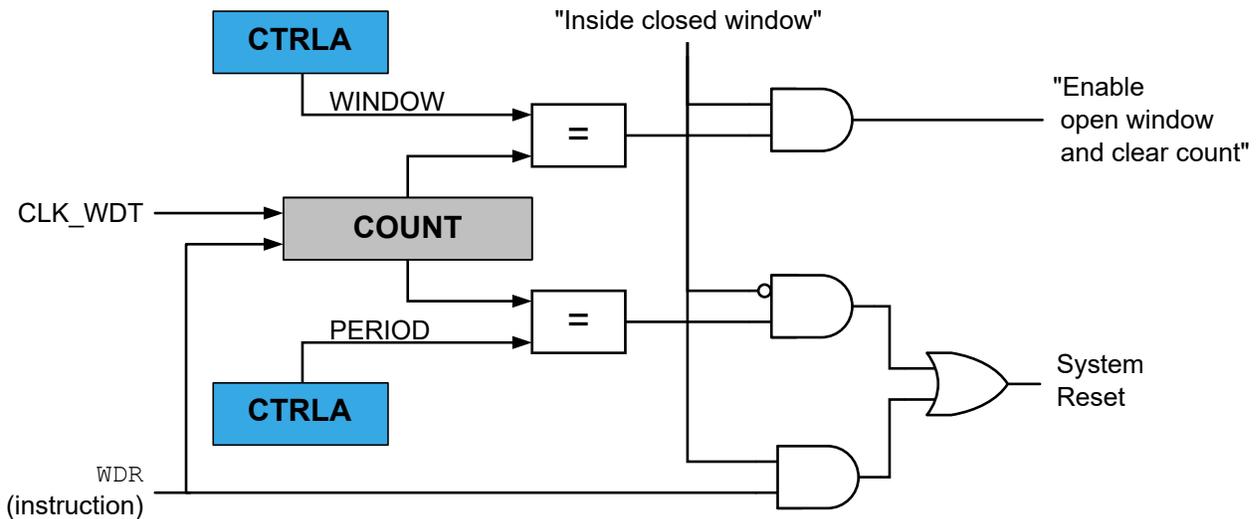
窗口模式在超时周期内定义了一个时隙或窗口，在此期间 WDT 必须复位。如果 WDT 在此窗口外复位，无论太早还是太晚，都会发出系统复位指令。相比于正常工作模式，窗口模式可以捕捉到代码错误导致常量 WDR 执行的情况。

使能后，WDT 将在工作模式和所有休眠模式下运行。WDT 是异步的（即，由独立于 CPU 的时钟源驱动）。因此，即使主时钟出现故障，WDT 也将继续运行并能够发出系统复位。

CCP 机制确保了 WDT 设置不会遭到意外更改。为了提高安全性，可通过配置来锁定 WDT 设置。

18.2.1. 框图

图 18-1. WDT 框图



18.3. 功能说明

18.3.1. 初始化

- 向控制 A (WDT.CTRLA) 寄存器中的周期 (PERIOD) 位写入非零值时, WDT 使能。
- 可选: 向 WDT.CTRLA 中的窗口 (WINDOW) 位写入非零值以使能窗口模式操作。

控制 A 寄存器中的所有位和 STATUS (WDT.STATUS) 寄存器中的锁定 (LOCK) 位均由配置更改保护机制进行写保护。

WDT.CTRLA 的复位值由熔丝 (FUSE.WDTCFG) 定义, 因此可以在启动时使能 WDT。如果是这种情况, WDT.STATUS 中的 LOCK 位在启动时置 1。

18.3.2. 时钟

1.024 kHz 振荡器时钟 (CLK_WDT_OSC) 来自内部超低功耗振荡器 OSCULP32K。由于采用超低功耗设计, 振荡器的精度不是很高, 因此确切的超时周期可能因器件而异。在设计使用 WDT 的软件时, 必须牢记这种差异, 以确保所用的超时周期对所有器件都有效。

1.024 kHz 振荡器时钟 CLK_WDT_OSC 与系统时钟异步。由于这种异步性, 写入 WDT 控制寄存器时需要时钟域之间同步。

18.3.3. 工作模式

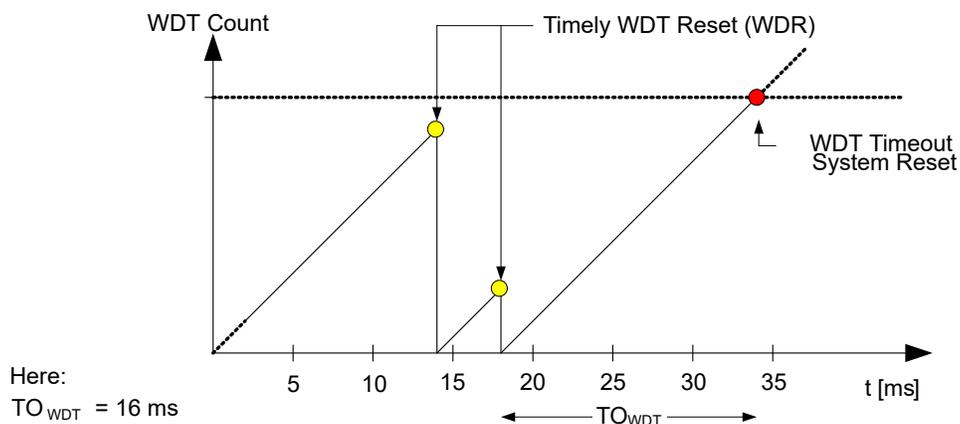
18.3.3.1. 正常模式

在正常模式下, 将为 WDT 设置一个超时周期。在发生超时之前, 如果未使用 WDR 通过软件将 WDT 复位, WDT 将发出系统复位。

每次通过 WDR 复位 WDT 时, 都会启动一个新的 WDT 超时周期。

通过对控制 A (WDT.CTRLA) 寄存器中的周期 (PERIOD) 位域进行写操作, 可从 8 ms 到 8s 的范围内选择 WDT 超时周期 (TO_{WDT}), 共有 11 种。

图 18-2. 正常模式操作



只要控制 A (WDT.CTRLA) 寄存器中的 WINDOW 位域为 0x0, 就会使能正常模式。

18.3.3.2. 窗口模式

在窗口模式下, WDT 使用两个不同的超时周期: 关闭窗口超时周期 (TO_{WDTW}) 和正常超时周期 (TO_{WDT}):

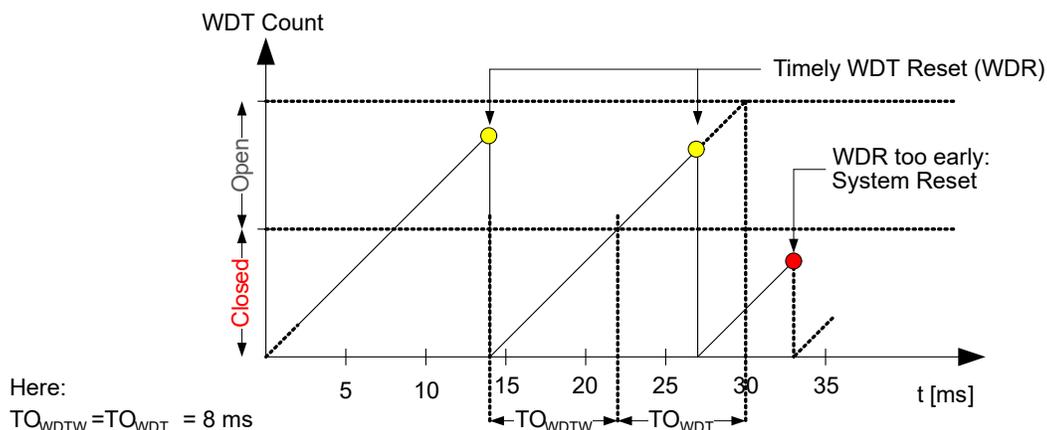
- 关闭窗口超时周期定义了一段介于 8 ms 到 8s 范围内的持续时间, 在此期间 WDT 无法复位。如果 WDT 在此期间发生复位, 则将发出系统复位。

- 正常 WDT 超时周期同样定义了一段介于 8 ms 到 8s 范围内的持续时间（称为打开周期），在此期间 WDT 可以（并且需要）复位。打开周期将始终跟在关闭周期之后，因此超时周期的总持续时间为关闭窗口超时周期和打开窗口超时周期之和。

使能窗口模式或退出调试模式时，第一个关闭周期在第一个 WDR 指令之后激活。

如果在同步前一个 WDR 时发出第二个 WDR，则将忽略第二个指令。

图 18-3. 窗口模式操作



通过向控制 A（WDT.CTRLA）寄存器中的 WINDOW 位域写入非零值来使能窗口模式，通过向该位写入 0x0 来禁止该模式。

18.3.3.3. 配置保护和锁定

WDT 提供两种安全机制来避免 WDT 设置遭到意外更改。

第一种机制是配置更改保护机制，采用定时写入程序来更改 WDT 控制寄存器。

第二种机制通过向 STATUS（WDT.STATUS）寄存器中的 LOCK 位写入 1 来锁定配置。当该位为 1 时，无法更改控制 A（WDT.CTRLA）寄存器。因此，无法通过软件禁止 WDT。

WDT.STATUS 中的 LOCK 只能写入 1。该位只能在调试模式下清零。

如果通过熔丝装入 WDT 配置，则 WDT.STATUS 中的 LOCK 会自动置 1。

18.3.4. 休眠模式操作

在源时钟处于活动状态的任何休眠模式下，WDT 都将继续工作。

18.3.5. 调试操作

在运行时调试时，此外设将继续正常工作。如果在调试模式下暂停 CPU，外设也将暂停正常工作。

在调试模式下停止 CPU 时，WDT 计数器将复位。

当再次启动 CPU 且 WDT 在窗口模式下工作时，将禁止第一个关闭窗口超时周期，并执行正常模式超时周期。

18.3.6. 同步

由于主时钟域和外设时钟域不同步，因此在写入控制 A（WDT.CTRLA）寄存器时需要进行同步。STATUS（WDT.STATUS）寄存器中的同步繁忙（SYNCBUSY）标志指示是否正在进行同步。

不允许在 SYNCBUSY = 1 时对 WDT.CTRLA 进行写操作。

写入以下寄存器时会进行同步：

- 控制 A (WDT.CTRLA) 寄存器中的 PERIOD 位
- WDT.CTRLA 中的窗口周期 (WINDOW) 位

WDR 指令需要两到三个 WDT 时钟周期才能实现同步。如果在一条 WDR 指令正在同步时发出一条新的 WDR 指令，则会忽略后者。

18.3.7. 配置更改保护

此外设的一些寄存器具有配置更改保护 (CCP)。要写入这些寄存器，必须先将特定密钥写入 CPU.CCP 寄存器，然后在 4 条 CPU 指令内对受保护位进行写访问。

必须遵循相应的 CCP 解锁序列，否则无法更改受保护的寄存器。

以下寄存器具有 CCP：

表 18-1. WDT——配置更改保护下的寄存器

寄存器	密钥
WDT.CTRLA	IOREG
WDT.STATUS 中的 LOCK 位	IOREG

受 CCP 保护的位/寄存器的列表：

- 控制 A 寄存器中的周期位 (CTRLA.PERIOD)
- 控制 A 寄存器中的窗口周期位 (CTRLA.WINDOW)
- STATUS 寄存器中的 LOCK 位 (STATUS.LOCK)

18.4. 寄存器汇总——WDT

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	WINDOW[3:0]				PERIOD[3:0]			
0x01	STATUS	7:0	LOCK							SYNCBUSY

18.5. 寄存器说明

18.5.1. 控制 A

名称: CTRLA
偏移量: 0x00
复位: 通过 FUSE.WDTCFG
属性: 配置更改保护

位	7	6	5	4	3	2	1	0
	WINDOW[3:0]				PERIOD[3:0]			
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	x	x	x	x	x	x	x	x

Bit 7:4 - WINDOW[3:0] 窗口

向这些位写入非零值可使能窗口模式，并相应地选择关闭周期的持续时间。
可选择对这些位进行锁定保护：

- 如果 WDT.STATUS 中的 LOCK 位为 1，则所有位均受到更改保护（访问 = R）
- 如果 WDT.STATUS 中的 LOCK 位为 0，则所有位均可更改（访问 = R/W）

值	名称	说明
0x0	OFF	-
0x1	8CLK	0.008s
0x2	16CLK	0.016s
0x3	32CLK	0.031s
0x4	64CLK	0.063s
0x5	128CLK	0.125s
0x6	256CLK	0.25s
0x7	512CLK	0.5s
0x8	1KCLK	1s
0x9	2KCLK	2s
0xA	4KCLK	4s
0xB	8KCLK	8s
其他	-	保留

Bit 3:0 - PERIOD[3:0] 周期

向该位写入非零值可使能 WDT，并相应地选择正常模式下的超时周期。在窗口模式下，这些位选择打开窗口的持续时间。

可选择对这些位进行锁定保护：

- 如果 WDT.STATUS 中的 LOCK 位为 1，则所有位均受到更改保护（访问 = R）
- 如果 WDT.STATUS 中的 LOCK 位为 0，则所有位均可更改（访问 = R/W）

值	名称	说明
0x0	OFF	-
0x1	8CLK	0.008s
0x2	16CLK	0.016s
0x3	32CLK	0.031s
0x4	64CLK	0.063s
0x5	128CLK	0.125s
0x6	256CLK	0.25s
0x7	512CLK	0.5s
0x8	1KCLK	1s
0x9	2KCLK	2s
0xA	4KCLK	4s
0xB	8KCLK	8s

值	名称	说明
其他	-	保留

18.5.2. 状态

名称: STATUS
偏移量: 0x01
复位: 0x00
属性: 配置更改保护

位	7	6	5	4	3	2	1	0
	LOCK							SYNCBUSY
访问	R/W							R
复位	0							0

Bit 7 - LOCK Lock

向该位写入 1 可对 WDT.CTRLA 寄存器进行写保护。
只能向该位写入 1。该位只能在调试模式下清零。
如果引导代码后 WDT.CTRLA 中的 PERIOD 位不为 0，则锁定位将自动置 1。
该位受 CCP 影响。

Bit 0 - SYNCBUSY 同步繁忙

当数据从系统时钟域同步到 WDT 时钟域时，该位在写入 WDT.CTRLA 寄存器后置 1。
同步完成后，系统会将该位清零。
该位不受 CCP 影响。

19. TCA——A 型 16 位定时器/计数器

19.1. 特性

- 16 位定时器/计数器
- 三个比较通道
- 双缓冲定时器周期设置
- 双缓冲比较通道
- 波形生成：
 - 频率生成
 - 单斜率 PWM（脉宽调制）
 - 双斜率 PWM
- 基于事件进行计数
- 定时器溢出中断/事件
- 每个比较通道一个比较匹配
- 拆分模式下为两个 8 位定时器/计数器

19.2. 概述

16 位 A 型 PWM 定时器/计数器（TCA）非常灵活，可提供精确的程序执行时序，或者用于生成频率和波形以及执行命令。

TCA 由一个基本计数器和一组比较通道组成。基本计数器可用于对时钟周期或事件进行计数或者让事件控制其对时钟周期的计数方式。它具有可用于计时的方向控制和周期设置。比较通道可与基本计数器一起使用，以进行比较匹配控制、频率生成和脉宽波形调制。

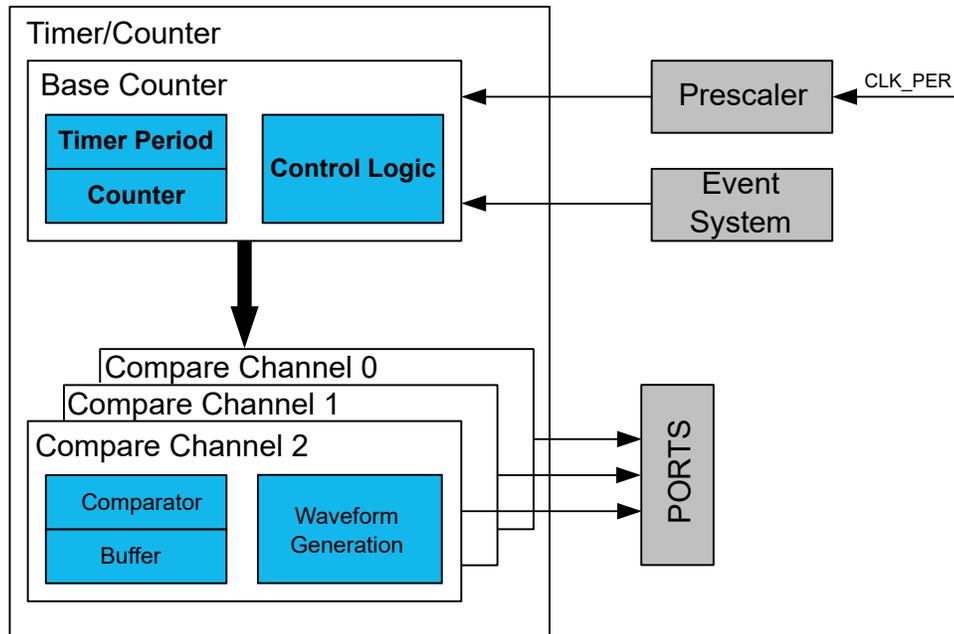
每当输入定时器/计数器时钟或事件时，计数器均会清零、重载、递增或递减，具体取决于工作模式。

定时器/计数器可由外设时钟（可选预分频）或事件系统提供时钟和时序。事件系统还可用于方向控制或同步操作。

默认情况下，TCA 是一个 16 位定时器/计数器。该定时器/计数器具有拆分模式功能，可将自身拆分为两个 8 位定时器/计数器，每个定时器/计数器有三个比较通道。

下图给出了 16 位定时器/计数器及与其密切相关的外设模块（灰色）的框图。

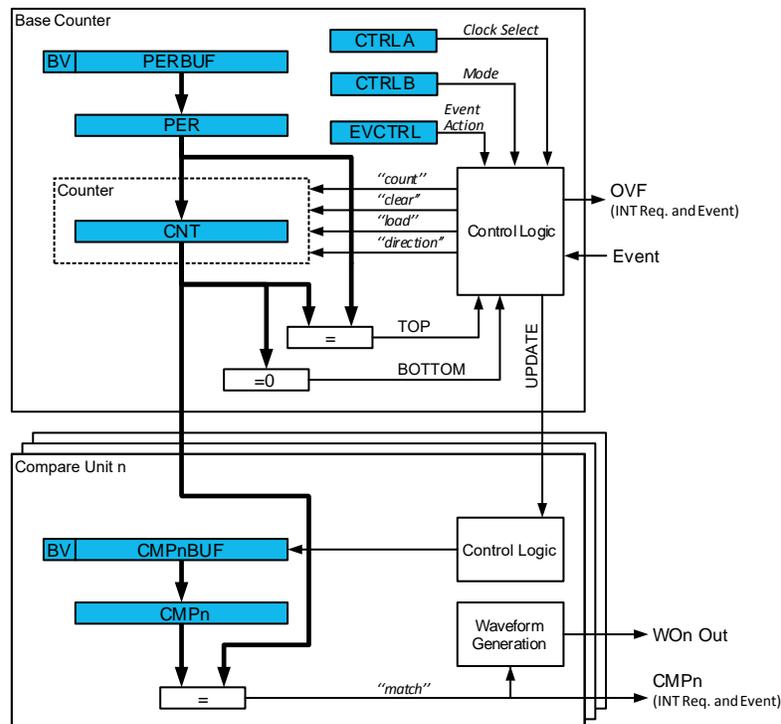
图 19-1. 16 位定时器/计数器及其密切相关的外设



19.2.1. 框图

下图显示了详细的定时器/计数器框图。

图 19-2. 定时器/计数器框图



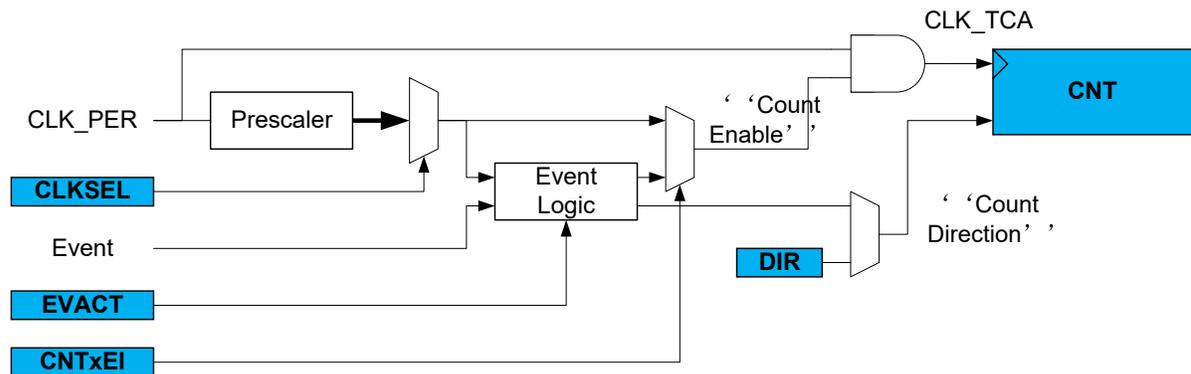
计数器 (TCAn.CNT) 寄存器、周期和比较 (TCAn.PER 和 TCAn.CMPn) 寄存器及其对应的缓冲寄存器 (TCAn.PERBUF 和 TCAn.CMPnBUF) 均为 16 位寄存器。所有缓冲寄存器都有一个缓冲区有效 (BV) 标志, 用于指示缓冲区何时包含新值。

正常工作期间, 计数器值不断与零和周期 (PER) 值进行比较, 以确定计数器是否已达到 TOP 或 BOTTOM。计数器值也可与 TCAn.CMPn 寄存器进行比较。

计数器 (TCAn.CNT) 寄存器达到 TOP、BOTTOM 或 CMPn 从而触发定时器/计数器后, 定时器/计数器会产生中断请求、事件或更改波形输出。中断请求、事件或波形输出变化将在触发后的下一个 CLK_TCA 周期发生。

CLK_TCA 是预分频的外设时钟或事件系统生成的事件, 如下图所示。

图 19-3. 定时器/计数器时钟逻辑



19.2.2. 信号说明

信号	说明	类型
WOn	数字输出	波形输出

19.3. 功能说明

19.3.1. 定义

以下定义通篇适用:

表 19-1. 定时器/计数器定义

名称	说明
BOTTOM	计数器在变为 0x0000 时达到 BOTTOM
MAX	计数器在变为全 1 时达到最大值
TOP	计数器在等于计数序列中的最大值时达到 TOP
UPDATE	定时器/计数器达到 BOTTOM 或 TOP 时满足更新条件, 具体取决于波形发生器模式。除非 TCAn.CTRLE 寄存器中的锁定更新 (LUPD) 位置 1, 否则将更新包含有效缓冲值的缓冲寄存器。
CNT	计数器寄存器值
CMP	比较寄存器值
PER	周期寄存器值

通常, 当定时器/计数器对周期性时钟节拍进行计数时, 使用时期定时器。当输入信号具有偶发或不规则节拍时, 使用时期计数器。事件计数的情况属于后者。

19.3.2. 初始化

要在基本模式下开始使用定时器/计数器，请按照以下步骤操作：

1. 向周期（TCAn.PER）寄存器写入 TOP 值。
2. 向控制 A（TCAn.CTRLA）寄存器中的 ENABLE 位写入 1，使能外设。
计数器将根据 TCAn.CTRLA 寄存器中时钟选择（CLKSEL）位域的预分频比设置开始计数时钟节拍。
3. 可选：向事件控制（TCAn.EVCTRL）寄存器中的事件输入计数使能（CNTEI）位写入 1，以对事件（而非时钟节拍）进行计数。
4. 可以从计数器（TCAn.CNT）寄存器中的计数器（CNT）位域读取计数器值。

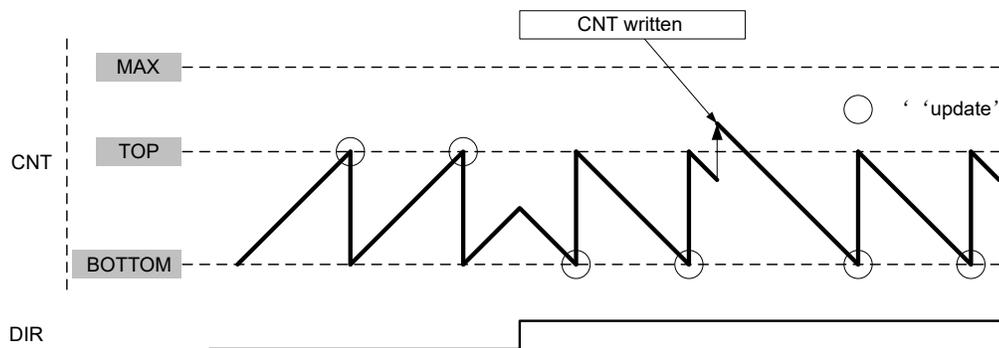
19.3.3. 操作

19.3.3.1. 正常操作

在正常操作时，计数器按照控制 E（TCAn.CTRLE）寄存器中方向（DIR）位选择的方向来计数时钟节拍，直至达到 TOP 或 BOTTOM。时钟节拍由外设时钟（CLK_PER）给出，根据控制 A（TCAn.CTRLA）寄存器中的时钟选择（CLKSEL）位域进行预分频。

向上计数达到 TOP 时，计数器将在下一个时钟节拍折回至 0。向下计数达到 BOTTOM 时，计数器将重新装入周期（TCAn.PER）寄存器值。

图 19-4. 正常操作



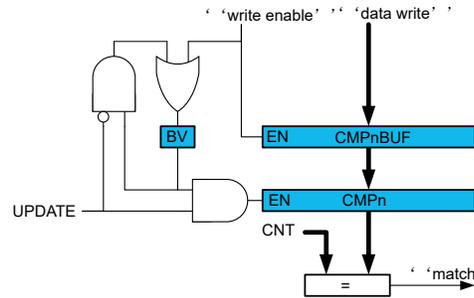
计数器运行时，可以更改计数器（TCAn.CNT）寄存器中的计数器值。对 TCAn.CNT 寄存器的写访问优先于计数、清零或重载，将立即执行。在正常工作期间，通过写入控制 E（TCAn.CTRLE）寄存器中的方向（DIR）位也可以更改计数器的方向。

19.3.3.2. 双缓冲

周期（TCAn.PER）寄存器值和比较 n（TCAn.CMPn）寄存器值都具有双缓冲（TCAn.PERBUF 和 TCAn.CMPnBUF）。

每个缓冲寄存器都在控制 F（TCAn.CTRLF）寄存器中有一个缓冲区有效（BV）标志（PERBV 和 CMPnBV），表示缓冲寄存器中包含一个有效值（即新值），该值可以复制到相应的周期或比较寄存器中。当周期寄存器和比较 n 寄存器用于比较操作时，BV 标志在数据写入缓冲寄存器时置 1，并在 UPDATE 条件下清零。比较（CMPn）寄存器的这一过程如下所示。

图 19-5. 周期和比较双缓冲



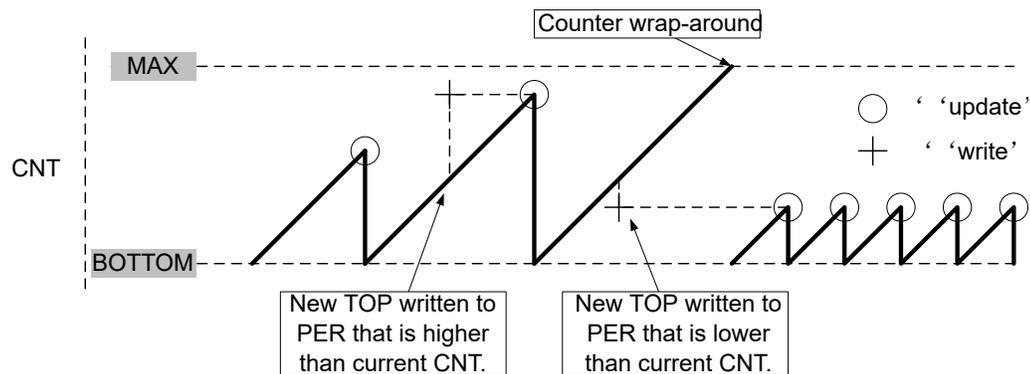
TCA n .CMP n 和 TCA n .CMP n BUF 寄存器都可用作 I/O 寄存器。这样便允许初始化并旁路缓冲寄存器，以及双缓冲功能。

19.3.3.3. 更改周期

将新的 TOP 值写入周期（TCA n .PER）寄存器可以更改计数器周期。

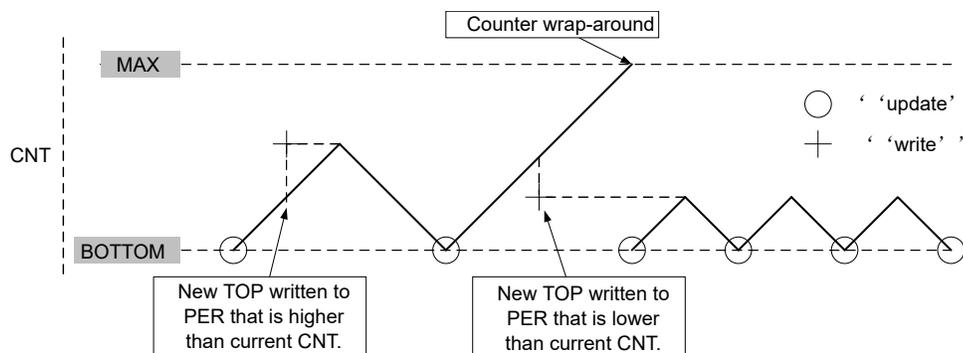
无缓冲：如果不使用双缓冲，则会立即更新周期。

图 19-6. 不使用缓冲更改周期



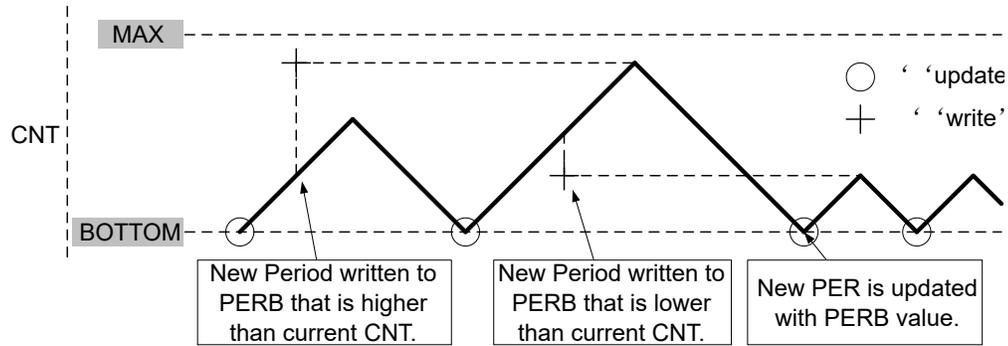
不使用缓冲向上计数时，在任何操作模式下都可能发生计数器折回，这是因为对 TCA n .CNT 和 TCA n .PER 寄存器进行连续比较。如果将新的 TOP 值写入 TCA n .PER 而该值小于当前的 TCA n .CNT，则计数器将在比较匹配发生之前先折回。

图 19-7. 未经缓冲的双斜率操作



使用缓冲：当使用双缓冲时，缓冲区可以随时写入并且仍然保持正常工作。TCA n .PER 始终在 UPDATE 条件下更新，如下图所示的双斜率操作。这可以防止折回和产生奇数个波形。

图 19-8. 使用缓冲更改周期



注：除非另行说明，否则缓冲将用于对 TCA 操作进行说明的图中。

19.3.3.4. 比较通道

每个比较通道 n 都会连续比较计数器值 (TCAn.CNT) 和比较 n (TCAn.CMPn) 寄存器。如果 TCAn.CNT 等于 TCAn.CMPn, 则比较器 n 会发出匹配信号。匹配信号将在下一个定时器时钟周期将比较通道的中断标志置 1, 并产生可选中断。

比较 n 缓冲 (TCAn.CMPnBUF) 寄存器提供与周期缓冲区相当的双缓冲功能。双缓冲功能会根据 UPDATE 条件将 TCAn.CMPn 寄存器更新和缓冲区值与计数序列的 TOP 或 BOTTOM 同步。同步操作可防止出现奇数长度的非对称脉冲, 从而实现无毛刺输出。

CMPnBUF 中的值在 UPDATE 条件下移至 CMPn, 并与下一次计数的计数器值 (TCAn.CNT) 进行比较。

19.3.3.4.1. 波形生成

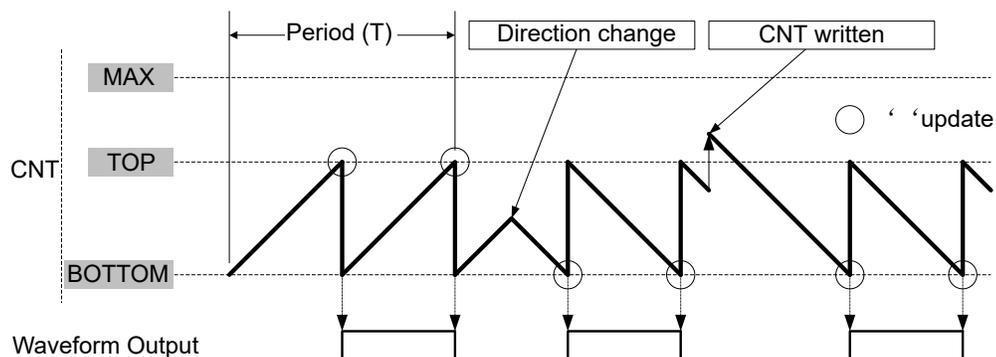
比较通道可用于在相应端口引脚上生成波形。必须满足以下要求才能使波形在连接的端口引脚上可见:

1. 必须通过写入 TCAn.CTRLB 寄存器中的波形生成模式 (WGMode) 位域来选择波形生成模式。
2. 必须使能所使用的比较通道 (TCAn.CTRLB 中的 CMPnEN = 1)。这将改写相应引脚的输出值。可通过配置端口多路开关 (PORTMUX) 选择备用引脚。有关详细信息, 请参见 PORTMUX 章节。
3. 必须将端口外设中相关端口引脚 n 的方向配置为输出。
4. 可选: 为相关端口引脚 n 使能反相波形输出。有关详细信息, 请参见 PORT 章节。

19.3.3.4.2. 频率 (FRQ) 波形生成

对于频率生成, 周期时间 (T) 由 TCAn.CMP0 寄存器而不是周期 (TCAn.PER) 寄存器控制。每次 TCAn.CNT 与 TCAn.CMPn 寄存器之间发生比较匹配时都会翻转相应的波形发生器输出。

图 19-9. 频率波形生成



以下公式定义了波形频率 (f_{FRQ}):

$$f_{FRQ} = \frac{f_{CLK_PER}}{2N(CMP0+1)}$$

其中 N 表示使用的预分频器分频比（见 TCA n .CTRLA 寄存器中的 CLKSEL 位域）， f_{CLK_PER} 表示外设时钟频率。

当 TCA n .CMP0 写入 0x0000 且不使用预分频（ $N = 1$ ，TCA n .CTRLA 中的 CLKSEL = 0x0）时，所生成波形的最大频率为外设时钟频率的一半（ $f_{CLK_PER}/2$ ）。

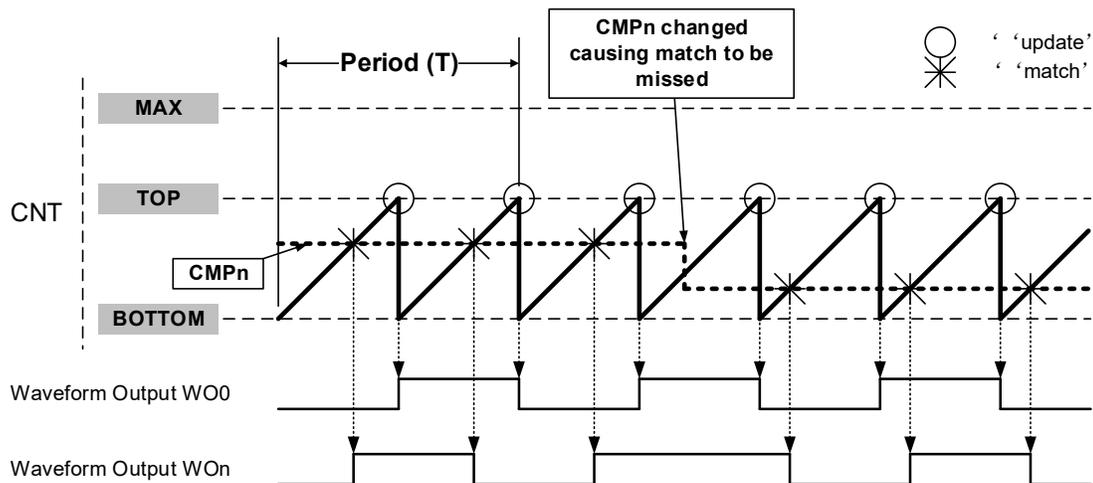
使用 TCA n .CMP1 和 TCA n .CMP2 寄存器来获取额外的波形输出 WOn。波形 WOn 可能与 WO0 相同，也可能存在偏移。偏移量会受 TCA n .CMPn、TCA n .CNT 和计数方向的影响。可使用下表中的两个公式计算以秒为单位的偏移量 t_{Offset} 。这两个公式仅在 $CMPn < CMP0$ 时适用。

表 19-2. 偏移公式概览

公式	计数方向	CMPn 与 CNT 的状态	偏移量
$t_{Offset} = \left(\frac{CMP0 - CMPn}{CMP0 + 1} \right) \left(\frac{T}{2} \right)$	UP	$CMPn \geq CNT$	WOn 在 WO0 之前
	DOWN	$CMP0 \leq CNT$ $CMP0 > CNT$ 且 $CMPn > CNT$	WOn 在 WO0 之后
$t_{Offset} = \left(\frac{CMPn + 1}{CMP0 + 1} \right) \left(\frac{T}{2} \right)$	UP	$CMPn < CNT$	WOn 在 WO0 之后
	DOWN	$CMPn \leq CNT$	WOn 在 WO0 之前

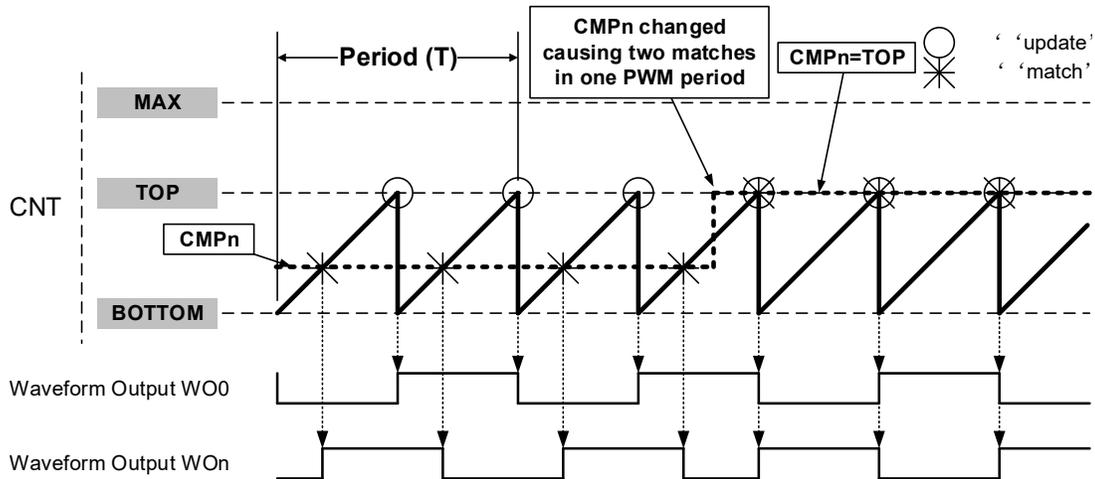
下图给出了 WOn 的前导和尾随偏移，其中两个公式都可以使用。正确的公式由计数方向以及定时器使能或 CMPn 更改时 CMPn 与 CNT 的状态决定。

图 19-10. 向上计数时的偏移



下图显示了如何通过更改 CMPn 来将波形反相。

图 19-11. 将波形输出反相

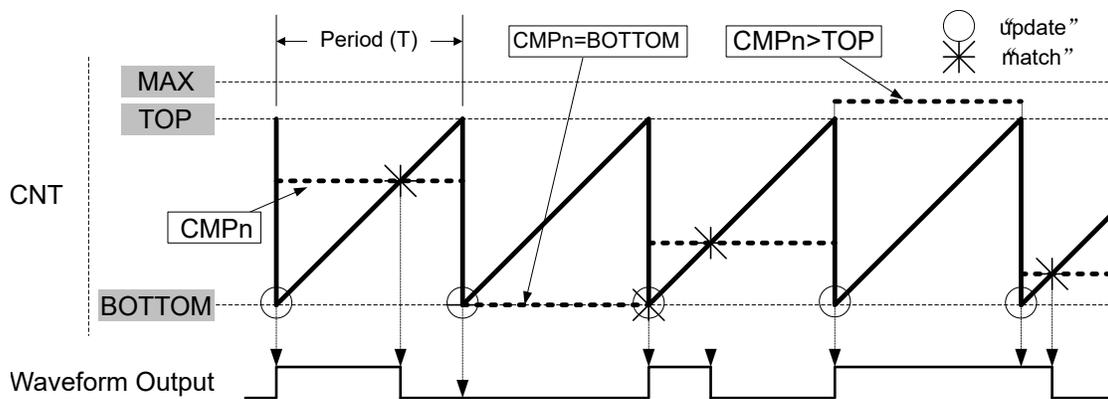


19.3.3.4.3. 单斜率 PWM 生成

单斜率脉宽调制 (Pulse-Width Modulation, PWM) 的生成周期 (T) 由 TCA_n.PER 寄存器控制, 而 TCA_n.CMP_n 寄存器的值控制所生成波形的占空比。下图显示了计数器如何从 BOTTOM 计数到 TOP, 然后再从 BOTTOM 重新开始计数。波形发生器输出在达到 BOTTOM 时置 1, 并在 TCA_n.CNT 和 TCA_n.CMP_n 寄存器之间发生比较匹配时清零。

CMP_n = BOTTOM 将在 WOn 上生成静态低电平信号, 而 CMP_n > TOP 将在 WOn 上生成静态高电平信号。

图 19-12. 单斜率脉宽调制



注:

1. 上图中的表示在使用 CMP_nBUF 更新 CMP_n 时有效。
2. 对于单斜率脉宽调制 (PWM) 生成, 不支持计数器从 TOP 计数到 BOTTOM。

TCA_n.PER 寄存器用于定义 PWM 分辨率。最小分辨率为 2 位 (TCA_n.PER = 0x0003, 最大分辨率为 16 位 (TCA_n.PER = MAX)。

以下公式可计算精确的单斜率 PWM 分辨率 ($R_{\text{PWM_SS}}$, 以位为单位) :

$$R_{\text{PWM_SS}} = \frac{\log(\text{PER}+1)}{\log(2)}$$

单斜率 PWM 频率 ($f_{\text{PWM_SS}}$) 取决于周期设置 (TCA_n.PER)、外设时钟频率 $f_{\text{CLK_PER}}$ 和 TCA 预分频比 (TCA_n.CTRLA 寄存器中的 CLKSEL 位域)。该值可通过以下公式计算, 其中 N 表示使用的预分频比:

$$f_{\text{PWM_SS}} = \frac{f_{\text{CLK_PER}}}{N(\text{PER}+1)}$$

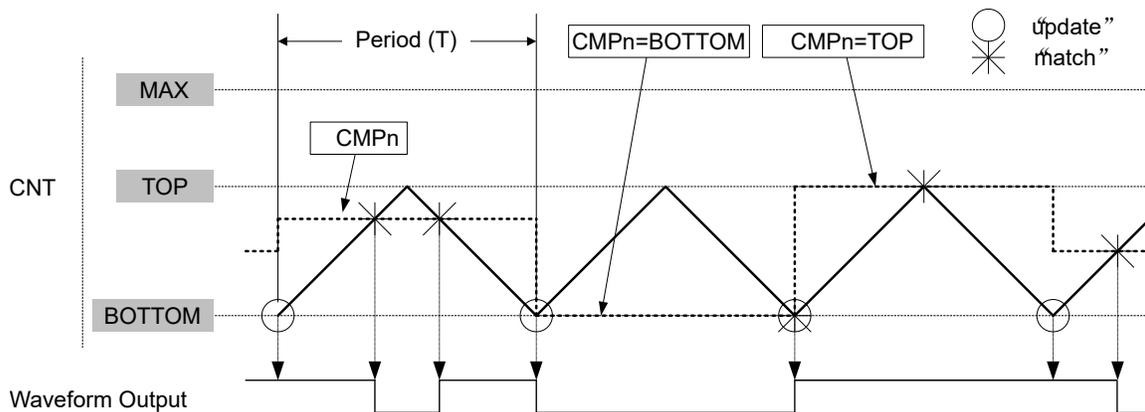
19.3.3.4.4. 双斜率 PWM

双斜率 PWM 的生成周期 (T) 由 TCA_n.PER 控制, 而 TCA_n.CMP_n 的值控制 WG 输出的占空比。

下图显示了双斜率 PWM 计数器如何反复从 BOTTOM 计数到 TOP, 然后再从 TOP 计数到 BOTTOM。波形发生器输出在达到 BOTTOM 时置 1, 在向上计数期间发生比较匹配时清零, 在向下计数期间发生比较匹配时置 1。

CMP_n = BOTTOM 将在 WO_n 上生成静态低电平信号, 而 CMP_n = TOP 将在 WO_n 上生成静态高电平信号。

图 19-13. 双斜率脉宽调制



注: 上图中的表示在使用 CMP_nBUF 更新 CMP_n 时有效。

周期 (TCA_n.PER) 寄存器用于定义 PWM 分辨率。最小分辨率为 2 位 (TCA_n.PER = 0x0003, 最大分辨率为 16 位 (TCA_n.PER = MAX)。

以下公式可计算精确的双斜率 PWM 分辨率 (R_{PWM_DS}, 以位为单位):

$$R_{\text{PWM_DS}} = \frac{\log(\text{PER}+1)}{\log(2)}$$

PWM 频率取决于 TCA_n.PER 寄存器中的周期设置、外设时钟频率 (f_{CLK_PER}) 和通过 TCA_n.CTRLA 寄存器中的 CLKSEL 位域选择的预分频比。该值可通过以下公式计算:

$$f_{\text{PWM_DS}} = \frac{f_{\text{CLK_PER}}}{2N \cdot \text{PER}}$$

N 表示使用的预分频比。

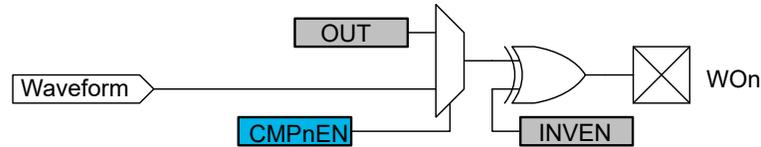
与单斜率 PWM 操作相比, 使用双斜率 PWM 会导致最大工作频率大约减半, 因为每个周期的定时器递增次数加倍。

19.3.3.4.5. 用于波形生成的端口改写

要在端口引脚上生成波形, 必须将相应的端口引脚方向设置为输出 (PORT_x.DIR[n] = 1)。使能比较通道 (TCA_n.CTRLB 寄存器中的 CMP_nEN = 1) 并选择了波形生成模式时, TCA 将改写端口引脚值。

下图显示了 TCA 的端口改写功能。定时器/计数器比较通道将改写相应端口引脚上的端口引脚输出值 (PORT_x.OUT)。在端口引脚上使能反相 I/O (PORT_x.PIN_nCTRL 寄存器中的 INVEN = 1) 可以将相应的 WG 输出反相。

图 19-14. A 型定时器/计数器的端口改写功能



19.3.3.5. 定时器/计数器命令

软件可以发出一组命令来立即更改外设的状态。这些命令可以直接控制 UPDATE、RESTART 和 RESET 信号。向控制 E (TCAn.CTRLSET) 寄存器中的命令 (CMD) 位域写入相应的值可以发出命令。

UPDATE 命令与 UPDATE 条件发生时的效果相同，但 UPDATE 命令不受控制 E (TCAn.CTRLE) 寄存器中的锁定更新 (LUPD) 位状态的影响。

软件可以通过发出 RESTART 命令强制重启当前波形周期。在这种情况下，计数器、方向和所有比较输出都设置为 0。

RESET 命令会将所有定时器/计数器寄存器设置为其初始值。只有在定时器/计数器未运行 (TCAn.CTRLA 寄存器中的 ENABLE = 0) 时才能发出 RESET 命令。

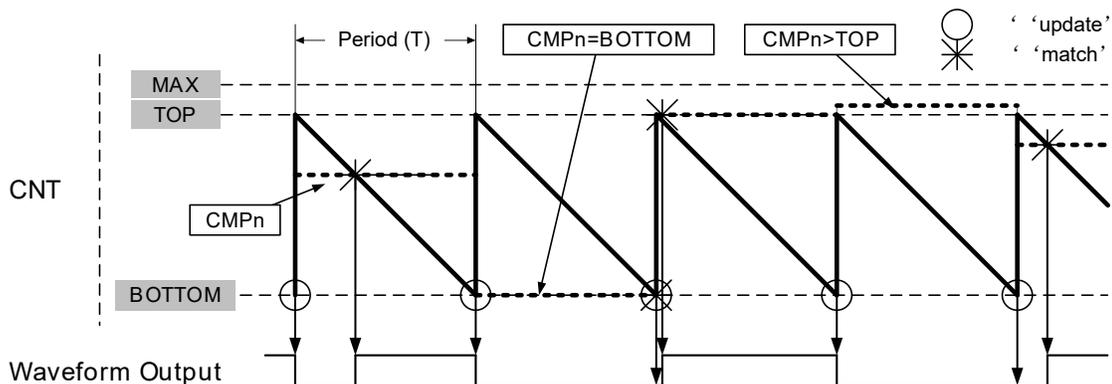
19.3.3.6. 拆分模式——两个 8 位定时器/计数器

拆分模式概述

为了使 TCA 中的定时器和 PWM 通道数量加倍，提供了拆分模式。在该拆分模式下，16 位定时器/计数器可用作两个独立的 8 位定时器，每个定时器都有三个比较通道可用于 PWM 生成。拆分模式仅适用于单斜率向下计数。拆分模式下不支持事件控制的操作。

下图显示了拆分模式下的单斜率 PWM 生成。波形发生器输出在达到 BOTTOM 时清零，并在计数器值 (TCAn.CNT) 和比较 n (TCAn.CMPn) 寄存器之间发生比较匹配时置 1。

图 19-15. 拆分模式下的单斜率脉宽调制



注：波形输出的最大占空比为 $TOP/(TOP+1)$

激活拆分模式会导致一些寄存器和寄存器位的功能发生变化。相关修改在单独的寄存器映射中介绍（见[寄存器汇总——拆分模式](#)）。

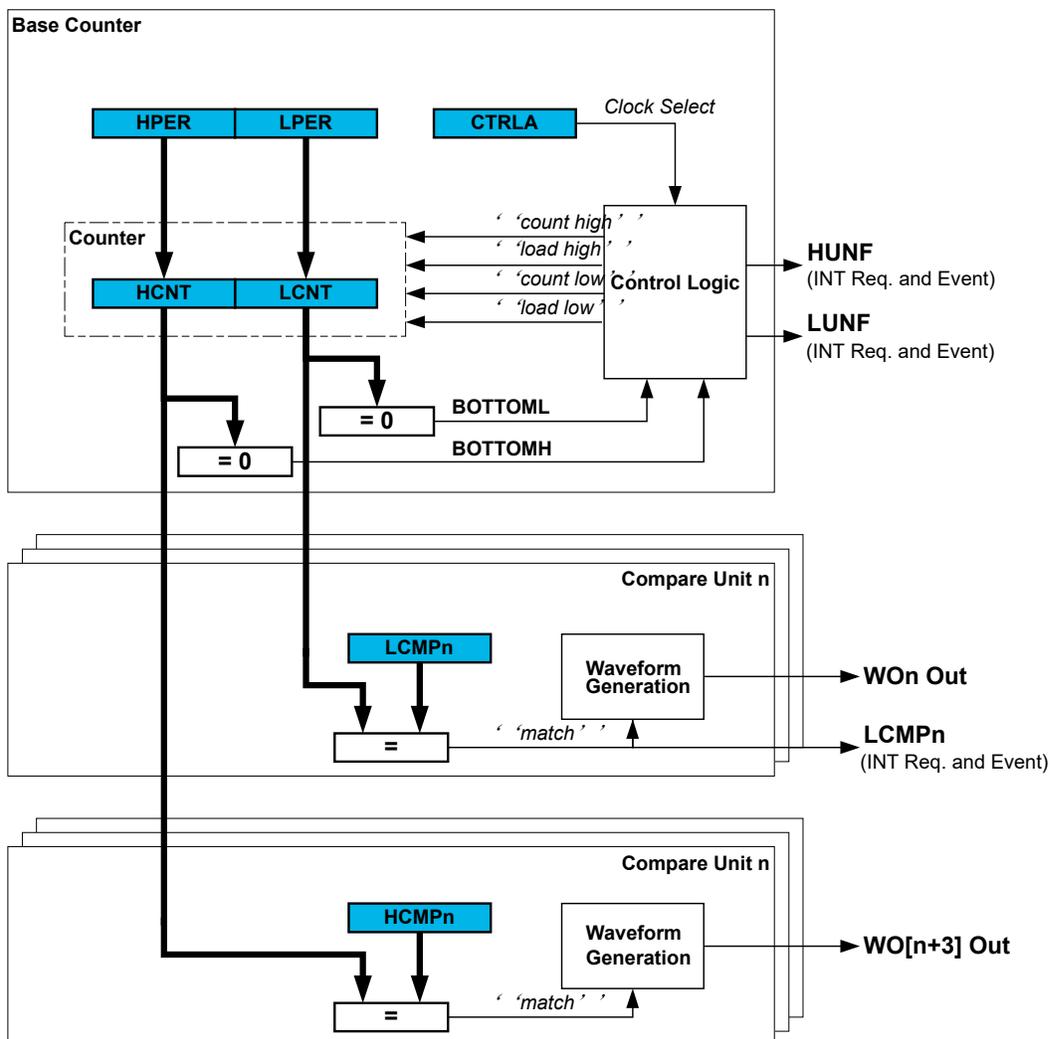
拆分模式与正常模式的差别

- 计数：
 - 仅适用于向下计数
 - 低字节定时器/计数器 (TCAn.LCNT) 寄存器和高字节定时器/计数器 (TCAn.HCNT) 寄存器相互独立

- 波形生成：
 - 仅适用于单斜率 PWM（TCAn.CTRLB 寄存器中的 WGMODE = SINGLESLOPE）
- 中断：
 - 低字节定时器/计数器（TCAn.LCNT）寄存器无变化
 - 高字节定时器/计数器（TCAn.HCNT）寄存器下溢中断
 - 高字节比较 n（TCAn.HCMPn）寄存器无比较中断或标志
- 事件动作：不兼容
- 缓冲寄存器和缓冲区有效标志：未使用
- 寄存器访问：所有寄存器均支持字节访问

框图

图 19-16. 定时器/计数器框图拆分模式



拆分模式初始化

在正常模式和拆分模式之间切换时，一些寄存器和位的功能会发生变化，但它们的值不变。因此，建议在更改模式时禁止外设（TCAn.CTRLA 寄存器中的 ENABLE = 0）并执行硬复位（TCAn.CTRLESET 寄存器中的 CMD = RESET）以避免意外行为。

要在硬复位后在基本拆分模式下开始使用定时器/计数器，请执行以下步骤：

1. 向控制 D（TCAn.CTRLD）寄存器中的拆分模式使能（SPLITM）位写入 1，使能拆分模式。
2. 向周期（TCAn.PER）寄存器写入 TOP 值。
3. 向控制 A（TCAn.CTRLA）寄存器中的使能（ENABLE）位写入 1，使能外设。
计数器将根据 TCAn.CTRLA 寄存器中的时钟选择（CLKSEL）位域的预分频比设置开始计数时钟节拍。
4. 可以从计数器（TCAn.CNT）寄存器中的计数器位域读取计数器值。

19.3.4. 事件

TCA 可生成下表中所述的事件。除 TCAn_HUNF 外的所有事件生成器在正常模式与分离模式操作之间共用。

表 19-3. TCA 中的事件生成器

生成器名称		说明	事件类型	生成时钟域	事件长度
外设	事件				
TCAn	OVF_LUNF	正常模式：上溢 拆分模式：低字节定时器下溢	脉冲	CLK_PER	一个 CLK_PER 周期
	HUNF	正常模式：不可用 拆分模式：高字节定时器下溢	脉冲	CLK_PER	一个 CLK_PER 周期
	CMP0	正常模式：比较通道 0 匹配 拆分模式：低字节定时器比较通道 0 匹配	脉冲	CLK_PER	一个 CLK_PER 周期
	CMP1	正常模式：比较通道 1 匹配 拆分模式：低字节定时器比较通道 1 匹配	脉冲	CLK_PER	一个 CLK_PER 周期
	CMP2	正常模式：比较通道 2 匹配 拆分模式：低字节定时器比较通道 2 匹配	脉冲	CLK_PER	一个 CLK_PER 周期

注：在正常模式和拆分模式下，生成事件的条件与将 TCAn.INTFLAGS 寄存器中的相应中断标志置 1 的条件相同。

TCA 具有一个事件用户，用于检测和响应输入事件。下表对事件用户及相关功能进行了说明。

表 19-4. TCA 中的事件用户

用户名称		说明	输入检测	异步/同步
外设	输入			
TCAn	CNT	在事件正边沿计数	边沿	同步
		在任何事件边沿计数	边沿	同步
		在事件信号为高电平时计数	电平	同步
		事件电平控制计数方向，低电平时递增计数，高电平时递减计数	电平	同步

注：

1. 拆分模式下不使用事件输入。
2. 基于电平输入检测的事件动作仅在事件频率低于定时器频率时才能可靠工作。

通过写入事件控制（TCAn.EVCTRL）寄存器中的事件动作（EVACT）位域选择上表中所述的特定动作。通过向事件控制（TCAn.EVCTRL）寄存器中的事件输入计数使能（CNTEI）位写入 1 使能输入事件。

有关事件类型和事件系统配置的更多详细信息，请参见 *事件系统 (EVSYS)* 一章。

19.3.5. 中断

表 19-5. 正常模式下的可用中断向量和中断源

名称	向量说明	条件
OVF	上溢或下溢中断	计数器已达到 TOP 或 BOTTOM
CMP0	比较通道 0 中断	计数器值和比较 0 寄存器之间匹配
CMP1	比较通道 1 中断	计数器值和比较 1 寄存器之间匹配
CMP2	比较通道 2 中断	计数器值和比较 2 寄存器之间匹配

表 19-6. 拆分模式下的可用中断向量和中断源

名称	向量说明	条件
LUNF	低字节下溢中断	低字节定时器达到其最小值
HUNF	高字节下溢中断	高字节定时器达到其最小值
LCMP0	比较通道 0 中断	计数器值和比较 0 寄存器的低字节之间匹配
LCMP1	比较通道 1 中断	计数器值和比较 1 寄存器的低字节之间匹配
LCMP2	比较通道 2 中断	计数器值和比较 2 寄存器的低字节之间匹配

发生中断条件时，外设的中断标志（*外设.INTFLAGS*）寄存器中的相应中断标志将置 1。

可以通过写入外设的中断控制（*外设.INTCTRL*）寄存器中的相应允许位来允许或禁止中断源。

当相应的中断源被允许且中断标志置 1 时，将生成中断请求。中断请求保持有效，直到中断标志清零。有关如何清零中断标志的详细信息，请参见外设的 *INTFLAGS* 寄存器。

19.3.6. 休眠模式操作

定时器/计数器将在空闲休眠模式下继续工作。

19.4. 寄存器汇总——正常模式

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0						CLKSEL[2:0]		ENABLE
0x01	CTRLB	7:0		CMP2EN	CMP1EN	CMP0EN	ALUPD		WGMODE[2:0]	
0x02	CTRLC	7:0						CMP2OV	CMP1OV	CMP0OV
0x03	CTRLD	7:0								SPLITM
0x04	CTRLECLR	7:0						CMD[1:0]	LUPD	DIR
0x05	CTRLESET	7:0						CMD[1:0]	LUPD	DIR
0x06	CTRLFCLR	7:0						CMP2BV	CMP1BV	CMP0BV
0x07	CTRLFSET	7:0						CMP2BV	CMP1BV	CMP0BV
0x08	保留									
0x09	EVCTRL	7:0						EVACT[2:0]		CNTEI
0x0A	INTCTRL	7:0		CMP2	CMP1	CMP0				OVF
0x0B	INTFLAGS	7:0		CMP2	CMP1	CMP0				OVF
0x0C	保留									
...	保留									
0x0D	保留									
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	TEMP	7:0					TEMP[7:0]			
0x10	保留									
...	保留									
0x1F	保留									
0x20	CNT	7:0					CNT[7:0]			
		15:8					CNT[15:8]			
0x22	保留									
...	保留									
0x25	保留									
0x26	PER	7:0					PER[7:0]			
		15:8					PER[15:8]			
0x28	CMP0	7:0					CMP[7:0]			
		15:8					CMP[15:8]			
0x2A	CMP1	7:0					CMP[7:0]			
		15:8					CMP[15:8]			
0x2C	CMP2	7:0					CMP[7:0]			
		15:8					CMP[15:8]			
0x2E	保留									
...	保留									
0x35	保留									
0x36	PERBUF	7:0					PERBUF[7:0]			
		15:8					PERBUF[15:8]			
0x38	CMP0BUF	7:0					CMPBUF[7:0]			
		15:8					CMPBUF[15:8]			
0x3A	CMP1BUF	7:0					CMPBUF[7:0]			
		15:8					CMPBUF[15:8]			
0x3C	CMP2BUF	7:0					CMPBUF[7:0]			
		15:8					CMPBUF[15:8]			

19.5. 寄存器说明——正常模式

19.5.1. 控制 A——正常模式

名称: CTRLA
 偏移量: 0x00
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
					CLKSEL[2:0]			ENABLE
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3:1 - CLKSEL[2:0] 时钟选择

该位域选择定时器/计数器的时钟频率。

值	名称	说明
0x0	DIV1	$f_{TCA} = f_{CLK_PER}$
0x1	DIV2	$f_{TCA} = f_{CLK_PER}/2$
0x2	DIV4	$f_{TCA} = f_{CLK_PER}/4$
0x3	DIV8	$f_{TCA} = f_{CLK_PER}/8$
0x4	DIV16	$f_{TCA} = f_{CLK_PER}/16$
0x5	DIV64	$f_{TCA} = f_{CLK_PER}/64$
0x6	DIV256	$f_{TCA} = f_{CLK_PER}/256$
0x7	DIV1024	$f_{TCA} = f_{CLK_PER}/1024$

Bit 0 - ENABLE 使能

值	说明
0	禁止外设
1	使能外设

19.5.2. 控制 B——正常模式

名称: CTRLB
 偏移量: 0x01
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
		CMP2EN	CMP1EN	CMP0EN	ALUPD	WGMODE[2:0]		
访问		R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位		0	0	0	0	0	0	0

Bit 4, 5, 6 - CMPnEN 比较 n 使能

在 FRQ 和 PWM 波形生成模式下，比较 n 使能（CMPnEN）位将在 WOn 对应的引脚上产生波形输出，从而改写相应 PORT 输出寄存器中的值。必须在 PORT 外设中将相应的引脚方向配置为输出。

值	说明
0	相应引脚上不产生波形输出 WOn
1	波形输出 WOn 将改写相应引脚的输出值

Bit 3 - ALUPD 自动锁定更新

自动锁定更新位控制 TCAn.CTRLE 寄存器中的锁定更新（LUPD）位。当 ALUPD 写入 1 时，LUPD 位将置 1，直到所有已使能比较通道的缓冲区有效（CMPnBV）位为 1。该条件会将 LUPD 位清零，清零状态将保持到下一个 UPDATE 条件，其中缓冲区的值将传送到 CMPn 寄存器，LUPD 位将再次置 1。这可确保在写入所有已使能比较缓冲区之后才将 CMPnBUF 寄存器值传送到 CMPn 寄存器。

值	说明
0	TCAn.CTRLE 寄存器中的 LUPD 位不由系统更改
1	TCAn.CTRLE 寄存器中的 LUPD 位自动置 1 和清零

Bit 2:0 - WGMODE[2:0] 波形生成模式

该位域选择波形生成模式并控制计数器的计数序列、TOP 值、UPDATE 条件、中断条件和所生成波形的类型。

在正常工作模式下不执行波形生成。对于所有其他模式，波形发生器输出将仅指向端口引脚（如果已将相应 CMPnEN 位置 1）。端口引脚方向必须设置为输出。

表 19-7. 定时器波形生成模式

值	组配置	工作模式	TOP	UPDATE	OVF
0x0	NORMAL	正常	PER	TOP ⁽¹⁾	TOP ⁽¹⁾
0x1	FRQ	频率	CMPO	TOP ⁽¹⁾	TOP ⁽¹⁾
0x2	-	保留	-	-	-
0x3	SINGLESLOPE	单斜率 PWM	PER	BOTTOM	BOTTOM
0x4	-	保留	-	-	-
0x5	DSTOP	双斜率 PWM	PER	BOTTOM	TOP
0x6	DSBOTH	双斜率 PWM	PER	BOTTOM	TOP 和 BOTTOM
0x7	DSBOTTOM	双斜率 PWM	PER	BOTTOM	BOTTOM

注:

1. 向上计数时。

19.5.3. 控制 C——正常模式

名称: CTRLC
 偏移量: 0x02
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
						CMP2OV	CMP1OV	CMP0OV
访问						R/W	R/W	R/W
复位						0	0	0

Bit 2 - CMP2OV 比较输出值 2
 请参见 CMP0OV。

Bit 1 - CMP1OV 比较输出值 1
 请参见 CMP0OV。

Bit 0 - CMP0OV 比较输出值 0
 当定时器/计数器未使能时，CMPnOV 位允许直接访问波形发生器的输出比较值。这用于在定时器/计数器未运行时将 WG 输出值置 1 或清零。

注：当输出连接到焊盘时，除非 TCA_n.CTRLB 寄存器中的 CMPnEN 位已置 1，否则改写这些位将不起作用。如果输出连接到 CCL，将旁路 TCA_n.CTRLB 寄存器中的 CMPnEN 位。

19.5.4. 控制 D——正常模式

名称: CTRLD
 偏移量: 0x03
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
访问								SPLITM
复位								R/W 0

Bit 0 - SPLITM 使能拆分模式

该位在拆分模式操作下设置定时器/计数器，并将用作两个 8 位定时器/计数器。与正常的 16 位模式相比，寄存器映射将发生变化。

19.5.5. 控制寄存器 E 清零——正常模式

名称: CTRLCLR
 偏移量: 0x04
 复位: 0x00
 属性: -

使用该寄存器代替读-修改-写 (RMW) 操作, 通过向各个位的位存储单元写入 1 将其清零。

位	7	6	5	4	3	2	1	0
					CMD[1:0]		LUPD	DIR
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3:2 - CMD[1:0] 命令

该位域用于通过软件控制定时器/计数器的更新、重启和复位。命令位域始终读为 0。

值	名称	说明
0x0	NONE	无命令
0x1	UPDATE	强制更新
0x2	RESTART	强制重启
0x3	RESET	强制硬复位 (如果使能定时器/计数器则忽略)

Bit 1 - LUPD 锁定更新

锁定更新位可用于确保在执行更新之前所有缓冲区都有效。

值	说明
0	一旦出现 UPDATE 条件, 就会更新缓冲的寄存器。
1	即使出现了更新条件, 也不会更新缓冲寄存器。此设置不会阻止命令位域发出的更新。

Bit 0 - DIR 计数器方向

通常, 该位由波形生成模式或事件动作通过硬件控制, 但也可通过软件更改。

值	说明
0	计数器向上计数 (递增)
1	计数器向下计数 (递减)

19.5.6. 控制寄存器 E 置 1——正常模式

名称: CTRLRESET
 偏移量: 0x05
 复位: 0x00
 属性: -

使用该寄存器代替读-修改-写 (RMW) 操作, 通过向各个位的位存储单元写入 1 将其置 1。

位	7	6	5	4	3	2	1	0
					CMD[1:0]		LUPD	DIR
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3:2 - CMD[1:0] 命令

该位域用于通过软件控制定时器/计数器的更新、重启和复位。命令位域始终读为 0。

值	名称	说明
0x0	NONE	无命令
0x1	UPDATE	强制更新
0x2	RESTART	强制重启
0x3	RESET	强制硬复位 (如果使能定时器/计数器则忽略)

Bit 1 - LUPD 锁定更新

锁定更新位可确保在执行更新之前所有缓冲区都有效。

值	说明
0	一旦出现 UPDATE 条件, 就会更新缓冲的寄存器。
1	即使出现了更新条件, 也不会更新缓冲寄存器。此设置不会阻止命令位域发出的更新。

Bit 0 - DIR 计数器方向

通常, 该位由波形生成模式或事件动作通过硬件控制, 但也可通过软件更改。

值	说明
0	计数器向上计数 (递增)
1	计数器向下计数 (递减)

19.5.7. 控制寄存器 F 清零

名称: CTRLFCLR
 偏移量: 0x06
 复位: 0x00
 属性: -

使用该寄存器代替读-修改-写 (RMW) 操作，通过向各个位的位存储单元写入 1 将其清零。

位	7	6	5	4	3	2	1	0
访问					CMP2BV	CMP1BV	CMP0BV	PERBV
复位					R/W	R/W	R/W	R/W
					0	0	0	0

Bit 3 - CMP2BV 比较 2 缓冲区有效

请参见 CMP0BV。

Bit 2 - CMP1BV 比较 1 缓冲区有效

请参见 CMP0BV。

Bit 1 - CMP0BV 比较 0 缓冲区有效

当新值写入相应的 TCA_n.CMP_nBUF 寄存器时，CMP_nBV 位置 1。这些位在 UPDATE 条件下自动清零。

Bit 0 - PERBV 周期缓冲区有效

当新值写入 TCA_n.PERBUF 寄存器时，该位置 1。该位在 UPDATE 条件下自动清零。

19.5.8. 控制寄存器 F 置 1

名称: CTRLFSET
 偏移量: 0x07
 复位: 0x00
 属性: -

使用该寄存器代替读-修改-写 (RMW) 操作，通过向各个位的位存储单元写入 1 将其置 1。

位	7	6	5	4	3	2	1	0
访问					CMP2BV	CMP1BV	CMP0BV	PERBV
复位					R/W	R/W	R/W	R/W
					0	0	0	0

Bit 3 - CMP2BV 比较 2 缓冲区有效

请参见 CMP0BV。

Bit 2 - CMP1BV 比较 1 缓冲区有效

请参见 CMP0BV。

Bit 1 - CMP0BV 比较 0 缓冲区有效

当新值写入相应的 TCA_n.CMP_nBUF 寄存器时，CMP_nBV 位置 1。这些位在 UPDATE 条件下自动清零。

Bit 0 - PERBV 周期缓冲区有效

当新值写入 TCA_n.PERBUF 寄存器时，该位置 1。该位在 UPDATE 条件下自动清零。

19.5.9. 事件控制

名称: EVCTRL
 偏移量: 0x09
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
					EVACT[2:0]			CNTEI
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3:1 - EVACT[2:0] 事件动作

该位域定义计数器在特定事件条件下将采取的动作。

值	名称	说明
0x0	EVACT_POSEDGE	在事件正边沿计数
0x1	EVACT_ANYEDGE	在任何事件边沿计数
0x2	EVACT_HIGHLVL	在事件信号为高电平时对预分频时钟周期进行计数
0x3	EVACT_UPDOWN	对预分频时钟周期进行计数。事件信号控制计数方向，低电平时递增计数，高电平时递减计数。计数器计数时会锁存方向。
其他		保留

Bit 0 - CNTEI 事件输入计数使能

值	说明
0	禁止对事件输入进行计数
1	使能对事件输入进行计数（根据 EVACT 位域）

19.5.10. 中断控制寄存器——正常模式

名称: INTCTRL
 偏移量: 0x0A
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
		CMP2	CMP1	CMP0				OVF
访问		R/W	R/W	R/W				R/W
复位		0	0	0				0

Bit 6 - CMP2 比较通道 2 中断允许
 请参见 CMP0。

Bit 5 - CMP1 比较通道 1 中断允许
 请参见 CMP0。

Bit 4 - CMP0 比较通道 0 中断允许
 向 CMPn 位写入 1 可允许比较通道 n 的中断。

Bit 0 - OVF 定时器上溢/下溢中断允许
 向 OVF 位写入 1 可允许上溢/下溢中断。

19.5.11. 中断标志寄存器——正常模式

名称: INTFLAGS
 偏移量: 0x0B
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
		CMP2	CMP1	CMP0				OVF
访问		R/W	R/W	R/W				R/W
复位		0	0	0				0

Bit 6 - CMP2 比较通道 2 中断标志
 请参见 CMP0 标志说明。

Bit 5 - CMP1 比较通道 1 中断标志
 请参见 CMP0 标志说明。

Bit 4 - CMP0 比较通道 0 中断标志

比较中断 (CMPn) 标志在相应比较通道发生比较匹配时置 1。

对于所有工作模式而言, 当计数 (TCAn.CNT) 寄存器与相应的比较 n (TCAn.CMPn) 寄存器之间发生比较匹配时, CMPn 标志将置 1。CMPn 标志不会自动清零, 只能通过向其位存储单元写入 1 的方式清零。

Bit 0 - OVF 上溢/下溢中断标志

该标志在 TOP (上溢) 或 BOTTOM (下溢) 条件下置 1, 具体取决于 WGMODE 设置。OVF 标志不会自动清零, 只能通过向其位存储单元写入 1 的方式清零。

19.5.12. 调试控制寄存器——正常模式

名称: DBGCTRL
 偏移量: 0x0E
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
								DBGRUN
访问								R/W
复位								0

Bit 0 - DBGRUN 在调试模式下运行

值	说明
0	外设在中断调试模式下停止并忽略事件
1	当 CPU 停止时，外设将继续运行在中断调试模式下

19.5.13. 用于 16 位访问的临时位

名称: TEMP
 偏移量: 0x0F
 复位: 0x00
 属性: -

临时寄存器供 CPU 用于对该外设的 16 位寄存器进行单周期 16 位访问。该寄存器由该外设的所有 16 位寄存器共用，可通过软件进行读写。有关读写 16 位寄存器的更多详细信息，请参见 *访问 16 位寄存器*。

位	7	6	5	4	3	2	1	0
	TEMP[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - TEMP[7:0] 用于 16 位访问的临时位

19.5.14. 计数器寄存器——正常模式

名称: CNT
 偏移量: 0x20
 复位: 0x00
 属性: -

TCA_n.CNTL 和 TCA_n.CNTH 寄存器对代表 16 位值 TCA_n.CNT。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

位	15	14	13	12	11	10	9	8
	CNT[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	CNT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:8 - CNT[15:8] 计数器高字节

该位域保存 16 位计数器寄存器的 MSB。

Bit 7:0 - CNT[7:0] 计数器低字节

该位域保存 16 位计数器寄存器的 LSB。

19.5.15. 周期寄存器——正常模式

名称: PER
 偏移量: 0x26
 复位: 0xFFFF
 属性: -

TCA_n.PER 寄存器包含除频率波形生成 (FRQ) 外的所有其他工作模式下定时器/计数器中的 16 位 TOP 值。

TCA_n.PERL 和 TCA_n.PERH 寄存器对代表 16 位值 TCA_n.PER。低字节[7:0] (后缀 L) 可通过原始偏移地址来访问。高字节[15:8] (后缀 H) 可通过偏移地址 + 0x01 来访问。

位	15	14	13	12	11	10	9	8
	PER[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1
位	7	6	5	4	3	2	1	0
	PER[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1

Bit 15:8 – PER[15:8] 周期高字节

该位域保存 16 位周期寄存器的 MSB。

Bit 7:0 – PER[7:0] 周期低字节

该位域保存 16 位周期寄存器的 LSB。

19.5.16. 比较 n 寄存器——正常模式

名称: CMPn
 偏移量: $0x28 + n*0x02$ [n=0..2]
 复位: 0x00
 属性: -

该寄存器不断与计数器值进行比较。通常，比较器的输出用于生成波形。

出现 UPDATE 条件时，TCAn.CMPn 寄存器将使用相应 TCAn.CMPnBUF 寄存器中的缓冲区值进行更新。

TCAn.CMPnL 和 TCAn.CMPnH 寄存器对代表 16 位值 TCAn.CMPn。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

位	15	14	13	12	11	10	9	8
	CMP[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	CMP[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:8 - CMP[15:8] 比较高字节

该位域保存 16 位比较寄存器的 MSB。

Bit 7:0 - CMP[7:0] 比较低字节

该位域保存 16 位比较寄存器的 LSB。

19.5.17. 周期缓冲寄存器

名称: PERBUF
 偏移量: 0x36
 复位: 0xFFFF
 属性: -

该寄存器用作周期 (TCAn.PER) 寄存器的缓冲区。从 CPU 或 UPDI 写入该寄存器会将 TCAn.CTRLF 寄存器中的周期缓冲区有效 (PERBV) 位置 1。

TCAn.PERBUFL 和 TCAn.PERBUFH 寄存器对代表 16 位值 TCAn.PERBUF。低字节[7:0] (后缀 L) 可通过原始偏移地址来访问。高字节[15:8] (后缀 H) 可通过偏移地址 + 0x01 来访问。

位	15	14	13	12	11	10	9	8
	PERBUF[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1
位	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1

Bit 15:8 - PERBUF[15:8] 周期缓冲区高字节
 该位域保存 16 位周期缓冲寄存器的 MSB。

Bit 7:0 - PERBUF[7:0] 周期缓冲区低字节
 该位域保存 16 位周期缓冲寄存器的 LSB。

19.5.18. 比较 n 缓冲寄存器

名称: CMPnBUF
 偏移量: $0x38 + n*0x02$ [$n=0..2$]
 复位: 0x00
 属性: -

该寄存器用作相关比较 n (TCAn.CMPn) 寄存器的缓冲区。从 CPU 或 UPDI 写入该寄存器会将 TCAn.CTRLF 寄存器中的比较缓冲区有效 (CMPnBV) 位置 1。

TCAn.CMPnBUFL 和 TCAn.CMPnBUFH 寄存器对代表 16 位值 TCAn.CMPnBUF。低字节[7:0] (后缀 L) 可通过原始偏移地址来访问。高字节[15:8] (后缀 H) 可通过偏移地址 + $0x01$ 来访问。

位	15	14	13	12	11	10	9	8
	CMPBUF[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	CMPBUF[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:8 - CMPBUF[15:8] 比较高字节

该位域保存 16 位比较缓冲寄存器的 MSB。

Bit 7:0 - CMPBUF[7:0] 比较低字节

该位域保存 16 位比较缓冲寄存器的 LSB。

19.6. 寄存器汇总——拆分模式

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0						CLKSEL[2:0]		ENABLE
0x01	CTRLB	7:0		HCMP2EN	HCMP1EN	HCMP0EN		LCMP2EN	LCMP1EN	LCMP0EN
0x02	CTRLC	7:0		HCMP2OV	HCMP1OV	HCMP0OV		LCMP2OV	LCMP1OV	LCMP0OV
0x03	CTRLD	7:0								SPLITM
0x04	CTRLECLR	7:0					CMD[1:0]		CMDEN[1:0]	
0x05	CTRLESET	7:0					CMD[1:0]		CMDEN[1:0]	
0x06	...									
0x09	保留									
0x0A	INTCTRL	7:0		LCMP2	LCMP1	LCMP0			HUNF	LUNF
0x0B	INTFLAGS	7:0		LCMP2	LCMP1	LCMP0			HUNF	LUNF
0x0C	...									
0x0D	保留									
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	...									
0x1F	保留									
0x20	LCNT	7:0					LCNT[7:0]			
0x21	HCNT	7:0					HCNT[7:0]			
0x22	...									
0x25	保留									
0x26	LPER	7:0					LPER[7:0]			
0x27	HPER	7:0					HPER[7:0]			
0x28	LCMP0	7:0					LCMP[7:0]			
0x29	HCMP0	7:0					HCMP[7:0]			
0x2A	LCMP1	7:0					LCMP[7:0]			
0x2B	HCMP1	7:0					HCMP[7:0]			
0x2C	LCMP2	7:0					LCMP[7:0]			
0x2D	HCMP2	7:0					HCMP[7:0]			

19.7. 寄存器说明——拆分模式

19.7.1. 控制 A——拆分模式

名称: CTRLA
 偏移量: 0x00
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
					CLKSEL[2:0]			ENABLE
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3:1 - CLKSEL[2:0] 时钟选择

该位域选择定时器/计数器的时钟频率。

值	名称	说明
0x0	DIV1	$f_{TCA} = f_{CLK_PER}$
0x1	DIV2	$f_{TCA} = f_{CLK_PER}/2$
0x2	DIV4	$f_{TCA} = f_{CLK_PER}/4$
0x3	DIV8	$f_{TCA} = f_{CLK_PER}/8$
0x4	DIV16	$f_{TCA} = f_{CLK_PER}/16$
0x5	DIV64	$f_{TCA} = f_{CLK_PER}/64$
0x6	DIV256	$f_{TCA} = f_{CLK_PER}/256$
0x7	DIV1024	$f_{TCA} = f_{CLK_PER}/1024$

Bit 0 - ENABLE 使能

值	说明
0	禁止外设
1	使能外设

19.7.2. 控制 B——拆分模式

名称: CTRLB
 偏移量: 0x01
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
		HCMP2EN	HCMP1EN	HCMP0EN		LCMP2EN	LCMP1EN	LCMP0EN
访问		R/W	R/W	R/W		R/W	R/W	R/W
复位		0	0	0		0	0	0

Bit 6 - HCMP2EN 高字节比较 2 使能
 请参见 HCMP0EN。

Bit 5 - HCMP1EN 高字节比较 1 使能
 请参见 HCMP0EN。

Bit 4 - HCMP0EN 高字节比较 0 使能
 在 FRQ 或 PWM 波形生成工作模式下将 HCMPnEN 位置 1 将改写相应 WO[n+3] 引脚的端口输出寄存器。

Bit 2 - LCMP2EN 低字节比较 2 使能
 请参见 LCMP0EN。

Bit 1 - LCMP1EN 低字节比较 1 使能
 请参见 LCMP0EN。

Bit 0 - LCMP0EN 低字节比较 0 使能
 在 FRQ 或 PWM 波形生成工作模式下将 LCMPnEN 位置 1 将改写相应 WOn 引脚的端口输出寄存器。

19.7.3. 控制 C——拆分模式

名称: CTRLC
 偏移量: 0x02
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
		HCMP2OV	HCMP1OV	HCMP0OV		LCMP2OV	LCMP1OV	LCMP0OV
访问		R/W	R/W	R/W		R/W	R/W	R/W
复位		0	0	0		0	0	0

Bit 6 - HCMP2OV 高字节比较 2 输出值
 请参见 HCMP0OV。

Bit 5 - HCMP1OV 高字节比较 1 输出值
 请参见 HCMP0OV。

Bit 4 - HCMP0OV 高字节比较 0 输出值
 当定时器/计数器未使能时，HCMPnOV 位允许直接访问波形发生器的输出比较值。这用于在定时器/计数器未运行时将 WO[n+3]输出值置 1 或清零。

Bit 2 - LCMP2OV 低字节比较 2 输出值
 请参见 LCMP0OV。

Bit 1 - LCMP1OV 低字节比较 1 输出值
 请参见 LCMP0OV。

Bit 0 - LCMP0OV 低字节比较 0 输出值
 当定时器/计数器未使能时，LCMPnOV 位允许直接访问波形发生器的输出比较值。这用于在定时器/计数器未运行时将 WOn 输出值置 1 或清零。

注: 当输出连接到焊盘时，除非 TCA_n.CTRLB 寄存器中的 xCMPnEN 位已置 1，否则改写这些位将不起作用。如果输出连接到 CCL，将旁路 TCA_n.CTRLB 寄存器中的 xCMPnEN 位。

19.7.4. 控制 D——拆分模式

名称: CTRLD
 偏移量: 0x03
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
访问								SPLITM
复位								R/W 0

Bit 0 - SPLITM 使能拆分模式

该位在拆分模式操作下设置定时器/计数器，并将用作两个 8 位定时器/计数器。与正常的 16 位模式相比，寄存器映射将发生变化。

19.7.5. 控制寄存器 E 清零——拆分模式

名称: CTRLCLR
 偏移量: 0x04
 复位: 0x00
 属性: -

使用该寄存器代替读-修改-写 (RMW) 操作, 通过向各个位的位存储单元写入 1 将其清零。

位	7	6	5	4	3	2	1	0
					CMD[1:0]		CMDEN[1:0]	
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3:2 - CMD[1:0] 命令

该位域用于通过软件控制定时器/计数器的重启和复位。命令位域始终读为 0。

值	名称	说明
0x0	NONE	无命令
0x1	-	保留
0x2	RESTART	强制重启
0x3	RESET	强制硬复位 (如果使能定时器/计数器则忽略)

Bit 1:0 - CMDEN[1:0] 命令使能

该位域配置 CMD 位给出的命令将应用于哪些定时器/计数器。

值	名称	说明
0x0	NONE	无
0x1	-	保留
0x2	-	保留
0x3	BOTH	命令 (CMD) 将同时应用于低字节和高字节定时器/计数器

19.7.6. 控制寄存器 E 置 1——拆分模式

名称: CTRLRESET
 偏移量: 0x05
 复位: 0x00
 属性: -

使用该寄存器代替读-修改-写 (RMW) 操作，通过向各个位的位存储单元写入 1 将其置 1。

位	7	6	5	4	3	2	1	0
					CMD[1:0]		CMDEN[1:0]	
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3:2 - CMD[1:0] 命令

该位域用于通过软件控制定时器/计数器的重启和复位。命令位域始终读为 0。CMD 位域必须与命令使能 (CMDEN) 位配合使用。使用 RESET 命令需要通过 CMDEN 同时选择低字节和高字节定时器/计数器。

值	名称	说明
0x0	NONE	无命令
0x1	-	保留
0x2	RESTART	强制重启
0x3	RESET	强制硬复位 (如果使能定时器/计数器则忽略)

Bit 1:0 - CMDEN[1:0] 命令使能

该位域配置 CMD 位给出的命令将应用于哪些定时器/计数器。

值	名称	说明
0x0	NONE	无
0x1	-	保留
0x2	-	保留
0x3	BOTH	命令 (CMD) 将同时应用于低字节和高字节定时器/计数器

19.7.7. 中断控制寄存器——拆分模式

名称: INTCTRL
 偏移量: 0x0A
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
		LCMP2	LCMP1	LCMP0			HUNF	LUNF
访问		R/W	R/W	R/W			R/W	R/W
复位		0	0	0			0	0

Bit 6 - LCMP2 低字节比较通道 2 中断允许
 请参见 LCMP0。

Bit 5 - LCMP1 低字节比较通道 1 中断允许
 请参见 LCMP0。

Bit 4 - LCMP0 低字节比较通道 0 中断允许
 向 LCMPn 位写入 1 可允许低字节比较通道 n 中断。

Bit 1 - HUNF 高字节下溢中断允许
 向 HUNF 位写入 1 可允许高字节下溢中断。

Bit 0 - LUNF 低字节下溢中断允许
 向 LUNF 位写入 1 可允许低字节下溢中断。

19.7.8. 中断标志寄存器——拆分模式

名称: INTFLAGS
 偏移量: 0x0B
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
		LCMP2	LCMP1	LCMP0			HUNF	LUNF
访问		R/W	R/W	R/W			R/W	R/W
复位		0	0	0			0	0

Bit 6 - LCMP2 低字节比较通道 2 中断标志
 请参见 LCMP0 标志说明。

Bit 5 - LCMP1 低字节比较通道 1 中断标志
 请参见 LCMP0 标志说明。

Bit 4 - LCMP0 低字节比较通道 0 中断标志
 低字节比较中断 (LCMPn) 标志在低字节定时器中的相应比较通道发生比较匹配时置 1。
 对于所有工作模式而言, 当低字节定时器计数器 (TCAn.LCNT) 寄存器与相应的比较 n (TCAn.LCMPn) 寄存器之间发生比较匹配时, LCMPn 标志将置 1。LCMPn 标志不会自动清零, 必须通过软件清零。这通过向其位存储单元写入 1 的方式完成。

Bit 1 - HUNF 高字节下溢中断标志
 该标志在高字节定时器 BOTTOM (下溢) 条件下置 1。HUNF 不会自动清零, 需要通过软件清零。这通过向其位存储单元写入 1 的方式完成。

Bit 0 - LUNF 低字节下溢中断标志
 该标志在低字节定时器 BOTTOM (下溢) 条件下置 1。LUNF 不会自动清零, 需要通过软件清零。这通过向其位存储单元写入 1 的方式完成。

19.7.9. 调试控制寄存器——拆分模式

名称: DBGCTRL
 偏移量: 0x0E
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
								DBGRUN
访问								R/W
复位								0

Bit 0 - DBGRUN 在调试模式下运行

值	说明
0	外设在中断调试模式下停止并忽略事件
1	当 CPU 停止时，外设将继续运行在中断调试模式下

19.7.10. 低字节定时器计数器寄存器——拆分模式

名称: LCNT
 偏移量: 0x20
 复位: 0x00
 属性: -

TCA_n.LCNT 寄存器包含低字节定时器的计数器值。CPU 和 UPDI 写访问优先于计数器的计数、清零或重载。

位	7	6	5	4	3	2	1	0
	LCNT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - LCNT[7:0] 低字节定时器的计数器值
 该位域定义低字节定时器的计数器值。

19.7.11. 高字节定时器计数器寄存器——拆分模式

名称: HCNT
 偏移量: 0x21
 复位: 0x00
 属性: -

TCA_n.HCNT 寄存器包含高字节定时器的计数器值。CPU 和 UPDI 写访问优先于计数器的计数、清零或重载。

位	7	6	5	4	3	2	1	0
	HCNT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - HCNT[7:0] 高字节定时器的计数器值
 该位域定义高字节定时器中的计数器值。

19.7.12. 低字节定时器周期寄存器——拆分模式

名称: LPER
 偏移量: 0x26
 复位: 0xFF
 属性: -

TCA_n.LPER 寄存器包含低字节定时器的 TOP 值。

位	7	6	5	4	3	2	1	0
	LPER[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1

Bit 7:0 - LPER[7:0] 低字节定时器周期值
 该位域保存低字节定时器的 TOP 值。

19.7.13. 高字节周期寄存器——拆分模式

名称: HPER
 偏移量: 0x27
 复位: 0xFF
 属性: -

TCA_n.HPER 寄存器包含高字节定时器的 TOP 值。

位	7	6	5	4	3	2	1	0
	HPER[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1

Bit 7:0 - HPER[7:0] 高字节定时器周期值
 该位域保存高字节定时器的 TOP 值。

19.7.14. 低字节定时器的比较寄存器 n——拆分模式

名称: LCMPn
 偏移量: $0x28 + n*0x02$ [$n=0..2$]
 复位: 0x00
 属性: -

TCA_n.LCMP_n 寄存器表示低字节定时器的比较通道 n 的比较值。该寄存器不断与低字节定时器 TCA_n.LCNT 的计数器值进行比较。然后比较器的输出通常用于生成波形。

位	7	6	5	4	3	2	1	0
	LCMP[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - LCMP[7:0] 通道 n 的比较值

该位域保存与 TCA_n.LCNT 进行比较的通道 n 的比较值。

19.7.15. 高字节比较寄存器 n——拆分模式

名称: HCMPn
 偏移量: $0x29 + n*0x02$ [$n=0..2$]
 复位: 0x00
 属性: -

TCAn.HCMPn 寄存器表示高字节定时器的比较通道 n 的比较值。该寄存器不断与高字节定时器 TCAn.HCNT 的计数器值进行比较。然后比较器的输出通常用于生成波形。

位	7	6	5	4	3	2	1	0
	HCMP[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - HCMP[7:0] 通道 n 的比较值

该位域保存与 TCAn.HCNT 进行比较的通道 n 的比较值。

20. TCB——16 位 B 型定时器/计数器

20.1. 特性

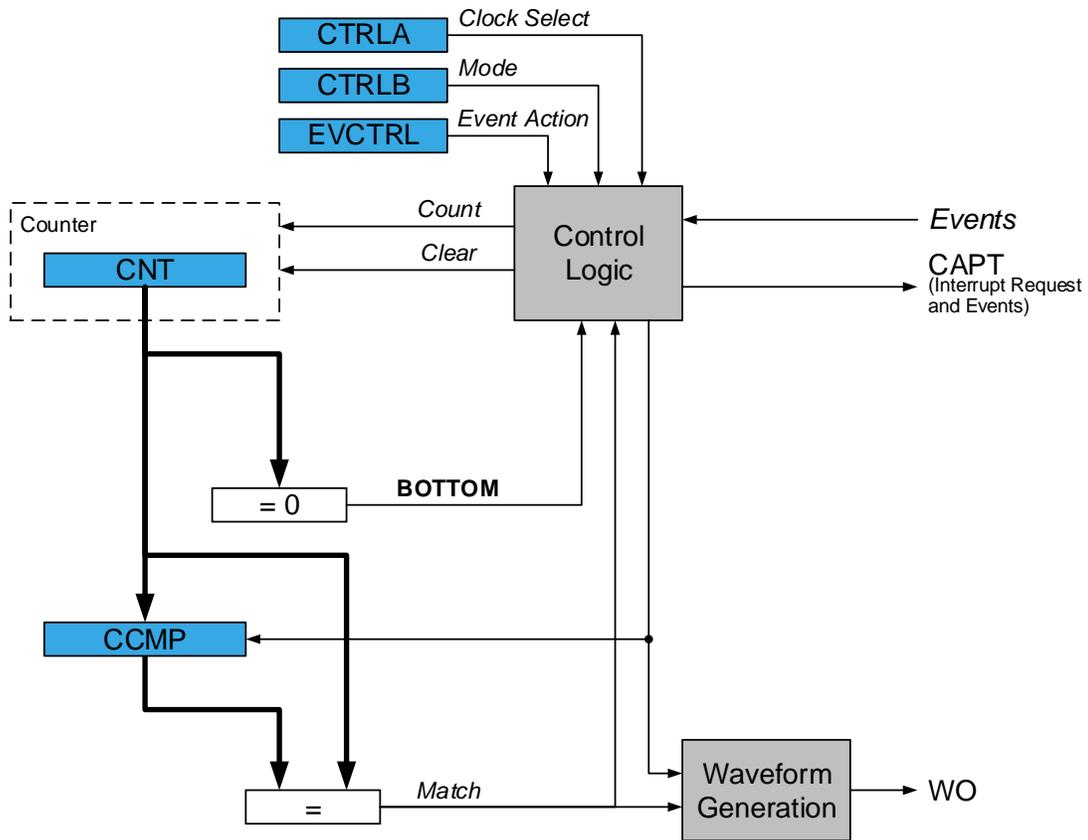
- 16 位计数器工作模式：
 - 周期性中断
 - 超时检查
 - 输入捕捉
 - 基于事件
 - 频率测量
 - 脉宽测量
 - 频率和脉宽测量
 - 单触发
 - 8 位脉宽调制 (PWM)
- 针对事件输入的噪声消除器
- 与 TCAn 同步操作

20.2. 概述

16 位 B 型定时器/计数器 (TCB) 的功能包括生成频率和波形以及针对事件的输入捕捉以及数字信号的时间和频率测量。TCB 由基本计数器和控制逻辑组成，可设置为八种不同的工作模式，每一种工作模式均可提供独特的功能。基本计数器的时钟由外设时钟（可选预分频）提供。

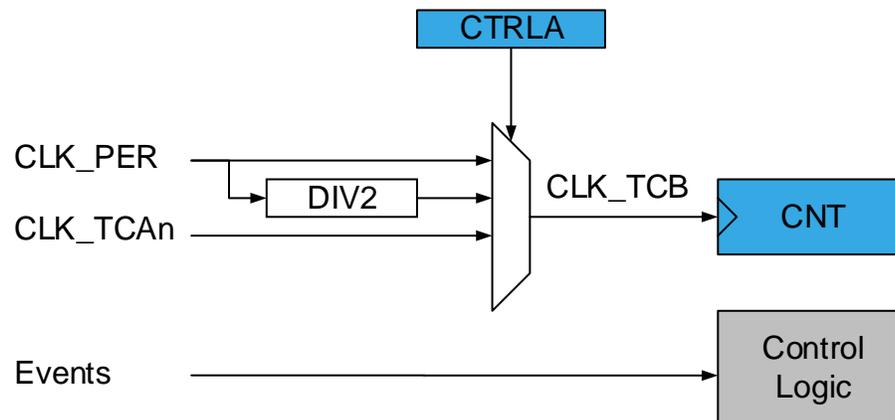
20.2.1. 框图

图 20-1. B 型定时器/计数器框图



该定时器/计数器可由外设时钟（CLK_PER）或 16 位 A 型定时器/计数器（CLK_TCA_n）提供时钟。

图 20-2. 定时器/计数器时钟逻辑



控制 A (TCBn.CTRLA) 寄存器中的时钟选择 (CLKSEL) 位域可直接选择预分频器输出之一作为时钟 (CLK_TCB) 输入。

将定时器/计数器设置为使用来自 TCAn 的时钟, 可使定时器/计数器与该 TCAn 同步运行。

通过使用 EVSYS, 任何事件源 (如任何 I/O 引脚上的外部时钟信号) 均可用作控制逻辑输入。使用事件动作控制的操作时, 必须将时钟选择设置为使用事件通道作为计数器输入。

20.2.2. 信号说明

信号	说明	类型
WO	数字异步输出	波形输出

20.3. 功能说明

20.3.1. 定义

以下定义通篇适用:

表 20-1. 定时器/计数器定义

名称	说明
BOTTOM	计数器在变为 0x0000 时达到 BOTTOM
MAX	计数器在变为 0xFFFF 时达到最大值
TOP	计数器在等于计数序列中的最大值时达到 TOP
CNT	计数 (TCBn.CNT) 寄存器值
CCMP	捕捉/比较 (TCBn.CCMP) 寄存器值

注: 通常, 当定时器/计数器对周期性时钟节拍进行计数时, 使用时期定时器。当输入信号具有偶发或不规则节拍时, 使用时期计数器。

20.3.2. 初始化

默认情况下, TCB 处于周期性中断模式。要开始使用 TCB, 请按照以下步骤操作:

1. 向比较/捕捉 (TCBn.CCMP) 寄存器写入 TOP 值。
2. 可选: 向控制 B (TCBn.CTRLB) 寄存器中的比较/捕捉输出使能 (CCMPEN) 位中写入 1。这样将在相应的引脚上提供波形输出, 从而改写相应 PORT 输出寄存器中的值。
3. 向控制 A (TCBn.CTRLA) 寄存器中的 ENABLE 位写入 1, 使能计数器。
计数器将根据控制 A (TCBn.CTRLA) 寄存器中的时钟选择 (CLKSEL) 位域中的预分频比设置开始计数时钟节拍。
4. 计数器值可从计数 (TCBn.CNT) 寄存器中读取。当 CNT 值达到 TOP 时, 外设将产生 CAPT 中断和事件。
 - a. 如果比较/捕捉寄存器被修改为低于当前计数寄存器的值, 外设将计数到 MAX 后折回。

20.3.3. 操作

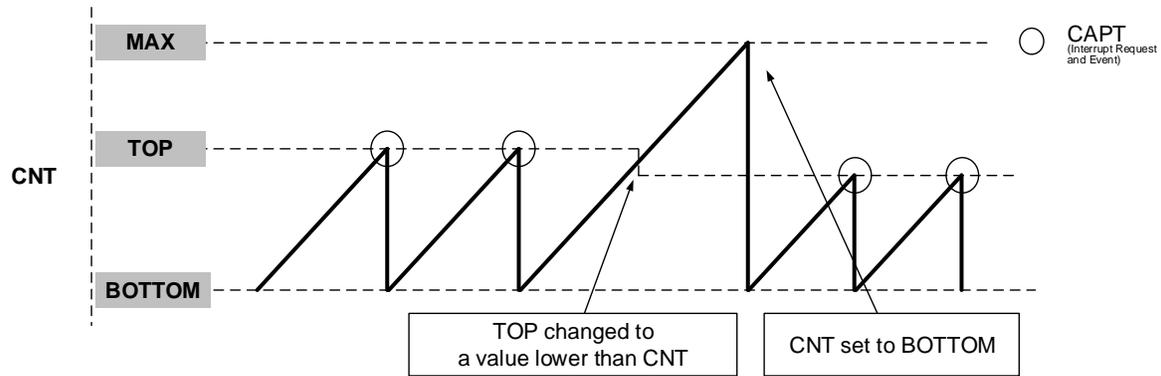
20.3.3.1. 模式

定时器可配置为以下各节中所述的八种不同模式。事件脉冲的时间需大于一个系统时钟周期, 以确保边沿检测。

20.3.3.1.1. 周期性中断模式

在周期性中断模式下, 计数器计数至捕捉值并从 BOTTOM 重新开始计数。当计数器等于 TOP 时, 将产生 CAPT 中断和事件。如果在达到 MAX 时 TOP 更新为小于计数的值, 则计数器将从 BOTTOM 重新开始计数。

图 20-3. 周期性中断模式

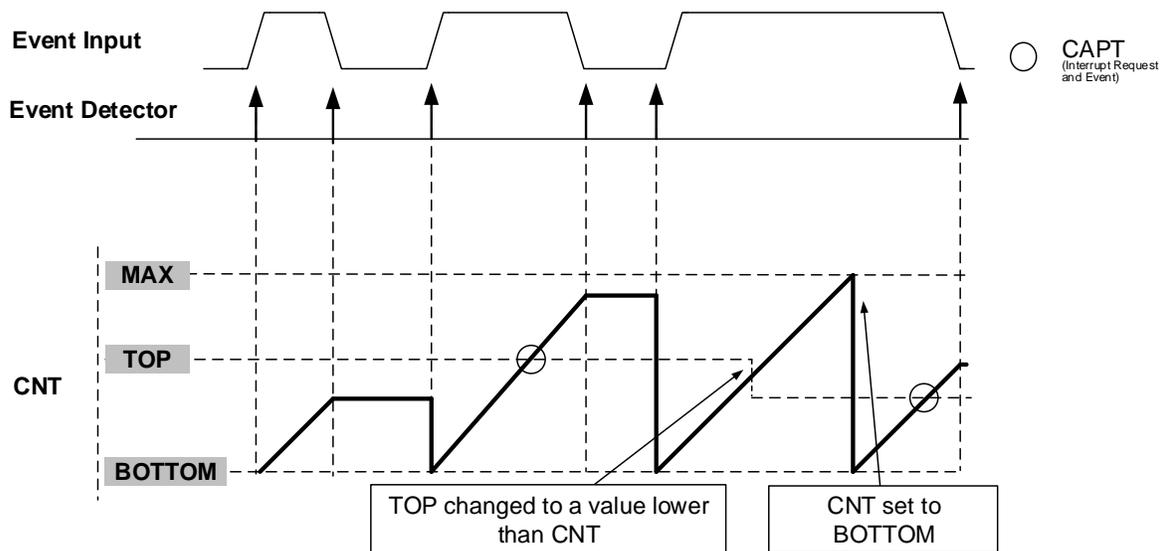


20.3.3.1.2. 超时检查模式

在超时检查模式下，外设从第一个信号边沿开始计数，并在事件输入通道上检测到的下一个信号边沿停止计数。启动或停止边沿由事件控制（TCBn.EVCTRL）寄存器中的事件边沿（EDGE）位决定。如果计数（TCBn.CNT）寄存器在第二个边沿之前达到 TOP，将产生 CAPT 中断和事件。在冻结状态下，检测到停止边沿后，计数器将在新的启动边沿重新开始计数。如果在达到 MAX 时 TOP 被更新为低于计数（TCBn.CNT）寄存器的值，则计数器将从 BOTTOM 重新开始计数。在冻结状态下读取计数（TCBn.CNT）寄存器或比较/捕捉（TCBn.CCMP）寄存器，或者写入状态（TCBn.STATUS）寄存器中的运行（RUN）位将不会产生任何作用。

该模式需要将 TCB 配置为事件用户，具体说明请参见“事件”一节。

图 20-4. 超时检查模式



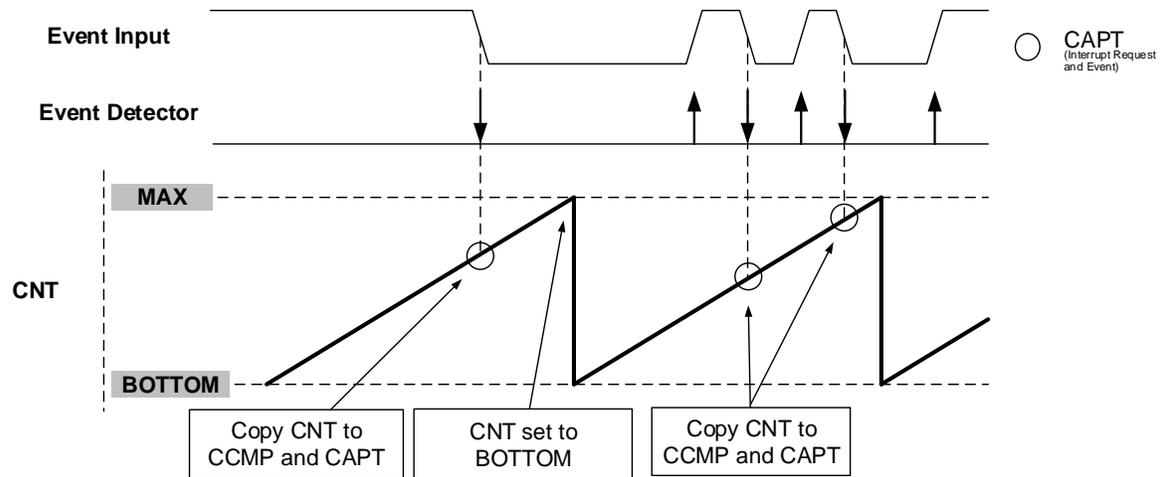
20.3.3.1.3. 输入捕捉事件模式

在输入捕捉事件模式下，计数器将从 BOTTOM 连续计数至 MAX。当检测到事件时，计数（TCBn.CNT）寄存器值将被传送到捕捉/比较（TCBn.CCMP）寄存器，同时产生 CAPT 中断和事件。事件边沿检测器可配置为在上升沿或下降沿触发捕捉。

该模式需要将 TCB 配置为事件用户，具体说明请参见“事件”一节。

下图给出了配置为在事件输入信号的下降沿进行捕捉的输入捕捉单元。读取比较/捕捉（TCBn.CCMP）寄存器的低字节后，CAPT 中断标志将自动清零。

图 20-5. 输入捕捉事件



重要：当从任何其他模式进入该模式时，建议向计数（TCBn.CNT）寄存器写入 0x0000。

20.3.3.1.4. 输入捕捉频率测量模式

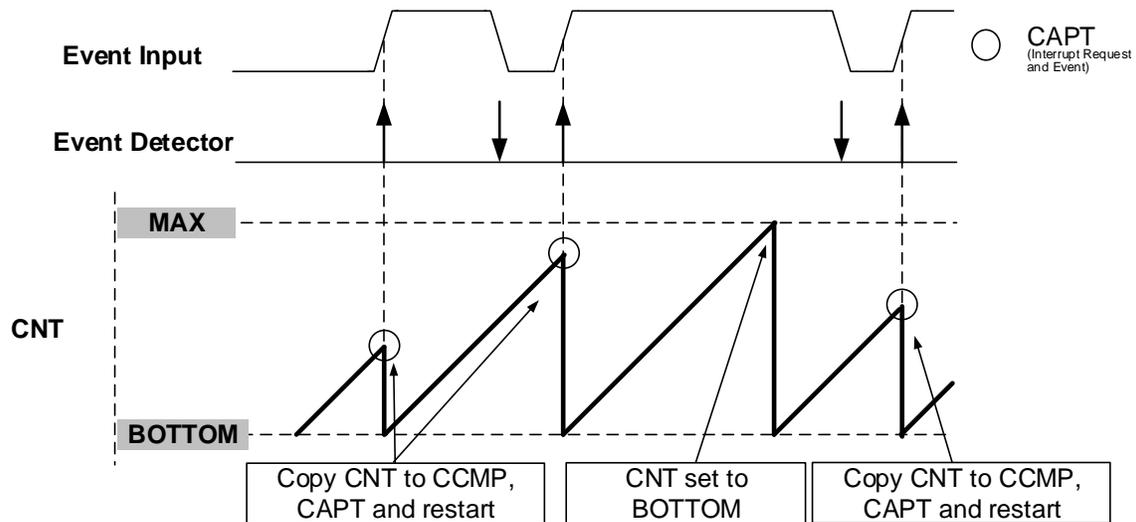
在输入捕捉频率测量模式下，TCB 会在事件输入信号的正边沿或负边沿捕捉计数器值并重新启动。

读取比较/捕捉（TCBn.CCMP）寄存器的低字节后，CAPT 中断标志将自动清零。

下图显示的是该模式在配置为在上升沿工作时的情况。

该模式需要将 TCB 配置为事件用户，具体说明请参见“事件”一节。

图 20-6. 输入捕捉频率测量

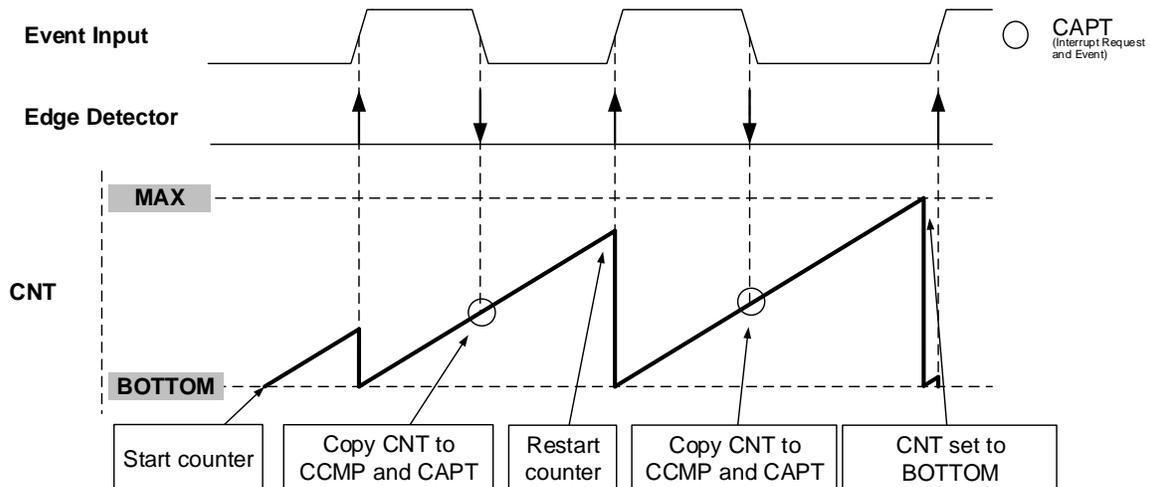


20.3.3.1.5. 输入捕捉脉宽测量模式

在输入捕捉脉宽测量模式下，将在正边沿重启输入捕捉脉宽测量并在下一个下降沿进行捕捉，然后产生中断请求。读取比较/捕捉（TCBn.CCMP）寄存器的低字节后，CAPT 中断标志将自动清零。定时器将在上升沿和下降沿检测之间自动切换，但需要两个时钟周期的最小边沿间隔以确保行为正确。

该模式需要将 TCB 配置为事件用户，具体说明请参见“事件”一节。

图 20-7. 输入捕捉脉宽测量



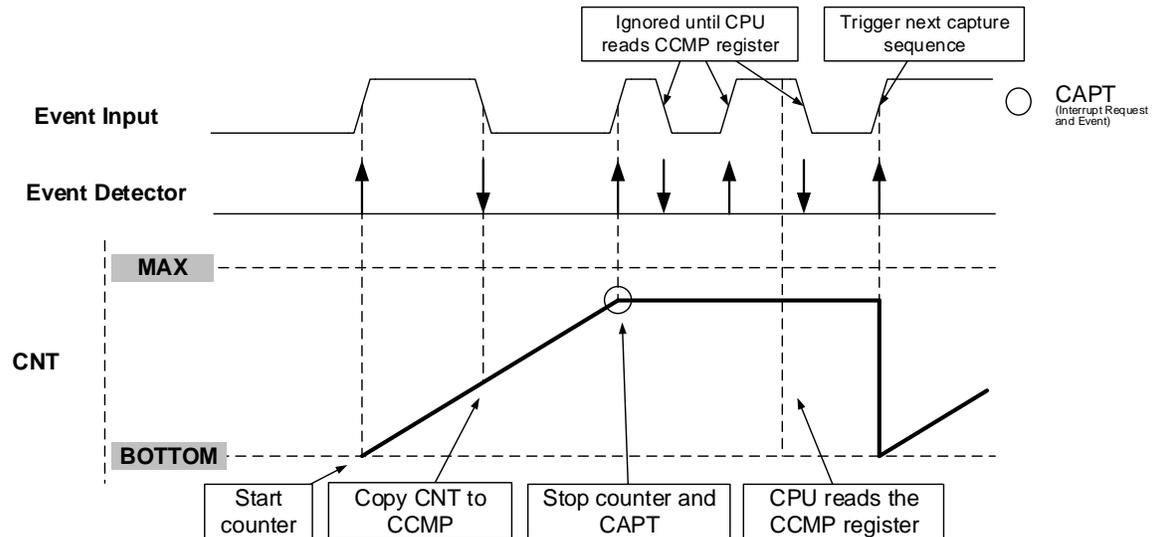
20.3.3.1.6. 输入捕捉频率和脉宽测量模式

在输入捕捉频率和脉宽测量模式下，当在事件输入信号上检测到正边沿时，定时器将开始计数。计数值在随后的下降沿被捕捉。当检测到事件输入信号的第二个上升沿时，计数器停止计数，这会将中断标志置 1。

读取比较/捕捉 (TCBn.CCMP) 寄存器的低字节后, CAPT 中断标志将自动清零, 定时器/计数器准备好进行新的捕捉序列。因此, 必须先读取计数 (TCBn.CNT) 寄存器再读取比较/捕捉 (TCBn.CCMP) 寄存器, 否则前一个寄存器会在事件输入信号的下一个正边沿复位为 BOTTOM。

该模式需要将 TCB 配置为事件用户, 具体说明请参见“事件”一节。

图 20-8. 输入捕捉频率和脉宽测量



20.3.3.1.7. 单触发模式

单触发模式可用于每次在连接的事件通道上观察到上升沿或下降沿时生成一个持续时间由比较 (TCBn.CCMP) 寄存器定义的脉冲。

当计数器停止时, 输出引脚被驱动为低电平。如果在连接的事件通道上检测到事件, 则定时器将复位并从 BOTTOM 开始向 TOP 计数, 同时将其输出驱动为高电平。通过读取状态 (TCBn.STATUS) 寄存器中的 RUN 位可以确定计数器是否正在计数。当计数器寄存器达到 CCMP 寄存器值时, 计数器将停止计数, 输出引脚将变为低电平并持续至少一个预分频器周期。在此期间出现的新事件将被忽略。从接收到事件到输出被设置为高电平之间有两个时钟周期的延时。向 TCB.EVCTRL 寄存器的 EDGE 位写入 1 时, 任何边沿都可以触发计数器启动。如果 EDGE 位为 0, 则只有正边沿可以触发启动。

计数器模块在使能后会立即开始计数, 即使没有触发事件。将 TOP 写入计数器寄存器可避免这种情况。如果在模块已使能时事件控制 (TCBn.EVCTRL) 寄存器中的事件边沿 (EDGE) 位为 1, 则会出现类似的行为。将 TOP 写入计数器寄存器也可以避免出现这种行为。

如果向控制 B (TCBn.CTRLB) 中的事件异步 (ASYNC) 位写入 1, 则定时器将异步响应传入事件。事件的边沿将立即使输出信号置 1。接收到事件后, 计数器仍会开始计数两个时钟周期。

该模式需要将 TCB 配置为事件用户, 具体说明请参见“事件”一节。

向 TCBn.CTRLB 中的比较/捕捉输出使能 (CCMPEN) 位写入 1 可启用波形输出，这样就能在相应的引脚上提供波形输出，从而改写相应 PORT 输出寄存器中的值。

下表列出了不同的配置及其对输出的影响。

表 20-2. 输出配置

CCMPEN	CNTMODE	ASYNC	输出
1	单触发模式	0	计数器启动时输出高电平，计数器停止时输出低电平
		1	事件到达时输出高电平，计数器停止时输出低电平
	8 位 PWM 模式	不适用	8 位 PWM 模式
	其他模式	不适用	输出初始电平设置 TCBn.CTRLB 寄存器中的 CCMPINIT 位
0	不适用	不适用	无输出

建议不要在外设使能时更改模式，否则会产生不可预测的输出。在配置定时器期间，可能会设置中断标志。建议在配置外设后清零定时器/计数器中断标志 (TCBn.INTFLAGS) 寄存器。

20.3.3.3. 噪声消除器

噪声消除器通过使用简单的数字滤波器方案来提高抗噪声能力。使能事件控制 (TCBn.EVCTRL) 寄存器中的噪声滤波器 (FILTER) 位时，外设会监视事件通道并保留最后四个观察样本的记录。如果四个连续样本相等，则输入被视为稳定，信号将馈入边沿检测器。

使能时，噪声消除器会在输入端发生变化到输入比较寄存器进行更新之间增加四个系统时钟周期的延时。

噪声消除器使用系统时钟，因此不受预分频器的影响。

20.3.3.4. 与 A 型定时器/计数器同步

写入控制 A (TCBn.CTRLA) 寄存器中的时钟选择 (CLKSEL) 位域可以将 TCB 配置为使用 A 型定时器/计数器 (TCAn) 的时钟 (CLK_TCA)。在该设置中，TCB 将使用与 TCAn 中选择的相同时钟源计数。

向控制 A (TCBn.CTRLA) 寄存器中的同步更新 (SYNCUPD) 位写入 1 时，TCB 计数器将在 TCAn 计数器重启时重启。

20.3.4. 事件

TCB 可生成下表中所述的事件：

表 20-3. TCB 中的事件生成器

生成器名称		说明	事件类型	生成时钟域	事件长度
外设	事件				
TCBn	CAPT	CAPT 标志置 1	脉冲	CLK_PER	一个 CLK_PER 周期

生成 CAPT 事件的条件与将定时器/计数器中断标志 (TCBn.INTFLAGS) 寄存器中的相应中断标志置 1 的条件相同。有关事件用户和事件系统配置的更多详细信息，请参见 [事件系统一章](#)。

TCB 可接收下表中所述的事件：

表 20-4. TCB 中的事件用户和可用事件动作

用户名称		说明	输入检测	异步/同步
外设	输入			
TCBn	CAPT	超时检查计数模式	边沿	同步
		输入捕捉事件计数模式		
		输入捕捉频率测量计数模式		
		输入捕捉脉宽测量计数模式		
		输入捕捉频率和脉宽测量计数模式		两者兼具
单触发计数模式				

如果事件控制（TCBn.EVCTRL）寄存器中的捕捉事件输入使能（CAPTEI）位写入 1，则传入事件将触发由事件控制（TCBn.EVCTRL）寄存器中的事件边沿（EDGE）位和控制 B（TCBn.CTRLB）寄存器中的定时器模式（CNTMODE）位域定义的事件动作。事件需要持续至少一个 CLK_PER 周期才能被识别到。

如果为单触发模式使能了异步模式，则事件在边沿触发，并将捕捉短于一个系统时钟周期的事件输入变化。

20.3.5. 中断

表 20-5. 可用中断向量和中断源

名称	向量说明	条件
CAPT	TCB 中断	具体取决于工作模式。请参见 TCBn.INTFLAG 寄存器中的 CAPT 位的说明。

发生中断条件时，外设的中断标志（外设.INTFLAGS）寄存器中的相应中断标志将置 1。

可以通过写入外设的中断控制（外设.INTCTRL）寄存器中的相应允许位来允许或禁止中断源。

当相应的中断源被允许且中断标志置 1 时，将生成中断请求。中断请求保持有效，直到中断标志清零。有关如何清零中断标志的详细信息，请参见外设的 INTFLAGS 寄存器。

20.3.6. 休眠模式操作

在待机休眠模式下，TCBn 默认处于禁止状态。一旦进入休眠模式，该模块将立即停止运行。

如果 TCBn.CTRLA 寄存器中的运行待机（RUNSTDBY）位写入 1，则该模块会在待机休眠模式下保持完全运行状态。

在掉电休眠模式下，所有操作都将停止。

20.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0		RUNSTDBY		SYNCUPD		CLKSEL[1:0]		ENABLE
0x01	CTRLB	7:0		ASYNC	CCMPINIT	CCMPEN		CNTMODE[2:0]		
0x02	保留									
0x03										
0x04	EVCTRL	7:0		FILTER		EDGE				CAPTEI
0x05	INTCTRL	7:0								CAPT
0x06	INTFLAGS	7:0								CAPT
0x07	STATUS	7:0								RUN
0x08	DBGCTRL	7:0								DBGRUN
0x09	TEMP	7:0	TEMP[7:0]							
0x0A	CNT	7:0	CNT[7:0]							
		15:8	CNT[15:8]							
0x0C	CCMP	7:0	CCMP[7:0]							
		15:8	CCMP[15:8]							

20.5. 寄存器说明

20.5.1. 控制 A

名称: CTRLA
 偏移量: 0x00
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
		RUNSTDBY		SYNCUPD		CLKSEL[1:0]		ENABLE
访问		R/W		R/W		R/W	R/W	R/W
复位		0		0		0	0	0

Bit 6 – RUNSTDBY 在待机模式下运行

向该位写入 1 将使外设处于待机休眠模式下运行。不适用于 CLKSEL 设置为 0x2 (CLK_TCA) 的情况。

Bit 4 – SYNCUPD 同步更新

向该位写入 1 后，只要 TCA0 计数器重新启动或溢出，TCB 便会重新启动。这可用于将捕捉与 PWM 周期同步。不适用于 CLKSEL 设置为 0x1 (CLK_PER/2) 的情况。

Bit 2:1 – CLKSEL[1:0] 时钟选择

写入这些位可选择该外设的时钟源。

值	名称	说明
0x0	CLKDIV1	CLK_PER
0x1	CLKDIV2	CLK_PER/DIV2
0x3	CLKTCA	使用来自 TCA0 的 TCA_CLK
0x4	-	保留

Bit 0 – ENABLE 使能

向该位写入 1 可使能 B 型定时器/计数器外设。

20.5.2. 控制 B

名称: CTRLB
 偏移量: 0x01
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
		ASYNC	CCMPINIT	CCMPEN		CNTMODE[2:0]		
访问		R/W	R/W	R/W		R/W	R/W	R/W
复位		0	0	0		0	0	0

Bit 6 - ASYNC 异步使能

向该位写入 1 可允许在单触发模式下异步更新 TCB 输出信号。

值	说明
0	当计数器在同步后启动时，输出将变为高电平
1	发生事件时，输出将变为高电平

Bit 5 - CCMPINIT 比较/捕捉引脚初始值

该位用于在使用引脚输出时设置引脚的初始输出值。该位不影响 8 位 PWM 模式和单触发模式。

值	说明
0	初始引脚状态为低电平
1	初始引脚状态为高电平

Bit 4 - CCMPEN 比较/捕捉输出使能

向该位写入 1 可使能波形输出并在相应的引脚上提供波形输出，从而改写相应 PORT 输出寄存器中的值。在 PORT 外设中将相应的引脚方向配置为输出。

值	说明
0	相应引脚上不使能波形输出
1	波形输出将改写相应引脚的输出值

Bit 2:0 - CNTMODE[2:0] 定时器模式

写入这些位可选择定时器模式。

值	名称	说明
0x0	INT	周期性中断模式
0x1	TIMEOUT	超时检查模式
0x2	CAPT	输入捕捉事件模式
0x3	FRQ	输入捕捉频率测量模式
0x4	PW	输入捕捉脉宽测量模式
0x5	FRQPW	输入捕捉频率和脉宽测量模式
0x6	SINGLE	单触发模式
0x7	PWM8	8 位 PWM 模式

20.5.3. 事件控制

名称: EVCTRL
 偏移量: 0x04
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
		FILTER		EDGE				CAPTEI
访问		R/W		R/W				R/W
复位		0		0				0

Bit 6 - FILTER 输入捕捉噪声消除滤波器
 向该位写入 1 可启用输入捕捉噪声消除单元。

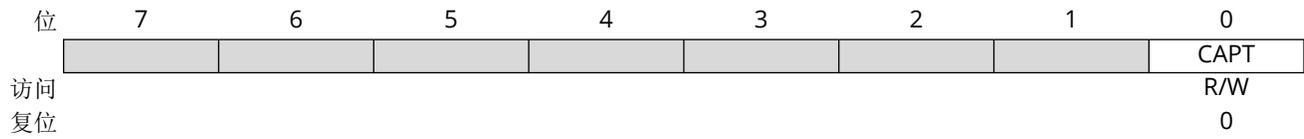
Bit 4 - EDGE 事件边沿
 该位用于选择事件边沿。该位的作用取决于 TCBn.CTRLB 中选择的计数模式（CNTMODE）位域。“—”表示事件或边沿不影响该模式。

计数模式	EDGE	正边沿	负边沿
周期性中断模式	0	—	—
	1	—	—
超时检查模式	0	启动计数器	停止计数器
	1	停止计数器	启动计数器
输入捕捉事件模式	0	输入捕捉, 中断	—
	1	—	输入捕捉, 中断
输入捕捉频率测量模式	0	输入捕捉, 清零并重启计数器, 中断	—
	1	—	输入捕捉, 清零并重启计数器, 中断
输入捕捉脉宽测量模式	0	清零并重启计数器	输入捕捉, 中断
	1	输入捕捉, 中断	清零并重启计数器
输入捕捉频率和脉宽测量模式	0	<ul style="list-style-type: none"> 在第 1 个正边沿: 清零并重启计数器 在接下来的负边沿: 输入捕捉 在第 2 个正边沿: 停止计数器, 中断 	—
	1	<ul style="list-style-type: none"> 在第 1 个负边沿: 清零并重启计数器 在接下来的正边沿: 输入捕捉 在第 2 个负边沿: 停止计数器, 中断 	—
单触发模式	0	启动计数器	—
	1	启动计数器	启动计数器
8 位 PWM 模式	0	—	—
	1	—	—

Bit 0 - CAPTEI 捕捉事件输入使能
 向该位写入 1 可启用输入捕捉事件。

20.5.4. 中断控制

名称: INTCTRL
 偏移量: 0x05
 复位: 0x00
 属性: -



Bit 0 - CAPT 捕捉中断允许

向该位写入 1 可允许捕捉中断。

20.5.5. 中断标志

名称: INTFLAGS
 偏移量: 0x06
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
								CAPT
访问								R/W
复位								0

Bit 0 - CAPT 捕捉中断标志

该位在发生捕捉中断时置 1。中断条件取决于控制 B (TCBn.CTRLB) 寄存器中的计数器模式 (CNTMODE) 位域。

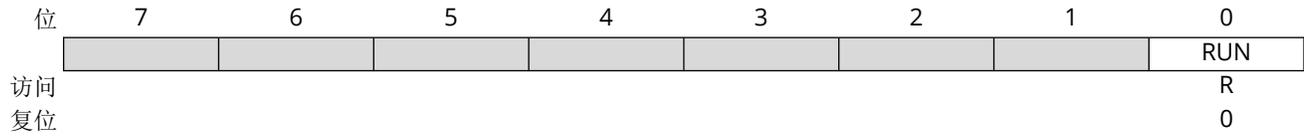
通过向该位写入 1 或在捕捉模式下读取捕捉寄存器来将该位清零。

表 20-6. 不同计数器模式的中断源设置条件

计数器模式	中断设置条件	TOP 值	CAPT
周期性中断模式	计数器达到 TOP 时置 1		
超时检查模式	计数器达到 TOP 时置 1	CCMP	CNT == TOP
单触发模式	计数器达到 TOP 时置 1		
输入捕捉频率测量模式	装载捕捉寄存器时在边沿置 1 并且计数器重启; 读取捕捉寄存器时标志清零		发生事件时将 CNT 复制到 CCMP 并重新开始计数 (CNT == BOTTOM)
输入捕捉事件模式	发生事件时置 1 并装载捕捉寄存器; 读取捕捉寄存器时标志清零	--	
输入捕捉脉宽测量模式	装载捕捉寄存器时在边沿置 1 并在前一个边沿初始化计数; 读取捕捉寄存器时标志清零		发生事件时将 CNT 复制到 CCMP 并继续计数
输入捕捉频率和脉宽测量模式	计数器停止时在第二个边沿 (正边沿或负边沿) 置 1, 读取捕捉寄存器时标志清零		
8 位 PWM 模式	计数器达到 CCML 时置 1	CCML	CNT == CCML

20.5.6. 状态

名称: STATUS
 偏移量: 0x07
 复位: 0x00
 属性: -



Bit 0 - RUN 运行

计数器运行时，该位置 1。计数器停止时，该位清零。
 该位是只读位，不能由 UPDI 置 1。

20.5.7. 调试控制

名称: DBGCTRL
 偏移量: 0x08
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
								DBGRUN
访问								R/W
复位								0

Bit 0 - DBGRUN 调试运行

值	说明
0	外设在中断调试模式下停止并忽略事件
1	当 CPU 停止时，外设将继续运行在中断调试模式下

20.5.8. 临时值

名称: TEMP
 偏移量: 0x09
 复位: 0x00
 属性: -

临时寄存器供 CPU 用于对该外设的 16 位寄存器进行单周期 16 位访问。该寄存器由该外设的所有 16 位寄存器共用，可通过软件进行读写。有关读写 16 位寄存器的更多详细信息，请参见 [访问 16 位寄存器](#)。

位	7	6	5	4	3	2	1	0
	TEMP[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - TEMP[7:0] 临时值

20.5.9. 计数

名称: CNT
 偏移量: 0x0A
 复位: 0x00
 属性: -

TCBn.CNTL 和 TCBn.CNTH 寄存器对代表 16 位值 TCBn.CNT。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

CPU 和 UPDI 写访问优先于寄存器的内部更新。

位	15	14	13	12	11	10	9	8
	CNT[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	CNT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:8 - CNT[15:8] 计数值高字节

这些位保存 16 位计数器寄存器的 MSB。

Bit 7:0 - CNT[7:0] 计数值低字节

这些位保存 16 位计数器寄存器的 LSB。

20.5.10. 捕捉/比较

名称: CCMP
 偏移量: 0x0C
 复位: 0x00
 属性: -

TCBn.CCMPL 和 TCBn.CCMPH 寄存器对代表 16 位值 TCBn.CCMP。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

该寄存器具有不同的功能，具体取决于工作模式：

- 对于捕捉操作，该寄存器包含发生捕捉时捕捉到的计数器值
- 在周期性中断/超时和单触发模式下，该寄存器用作 TOP 值
- 在 8 位 PWM 模式下，TCBn.CCMPL 和 TCBn.CCMPH 用作两个独立的寄存器: CCMPH 控制波形的周期，CCMPL 控制占空比。

位	15	14	13	12	11	10	9	8
	CCMP[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	CCMP[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:8 – CCMP[15:8] 捕捉/比较值高字节

这些位保存 16 位比较、捕捉和最高值的 MSB。

Bit 7:0 – CCMP[7:0] 捕捉/比较值低字节

这些位保存 16 位比较、捕捉和最高值的 LSB。

21. TCD——12 位 D 型定时器/计数器

21.1. 特性

- 12 位定时器/计数器
- 可编程预分频器
- 双缓冲比较寄存器
- 波形生成：
 - 单斜坡模式
 - 双斜坡模式
 - 四斜坡模式
 - 双斜率模式
- 两个单独的输入通道
- 软件捕捉和基于输入的捕捉
- 针对输入事件的可编程滤波器
- 针对外部事件的条件波形生成：
 - 故障处理
 - 输入消隐
 - 过载保护
 - 通过硬件快速紧急停止
- 半桥和全桥输出支持

21.2. 概述

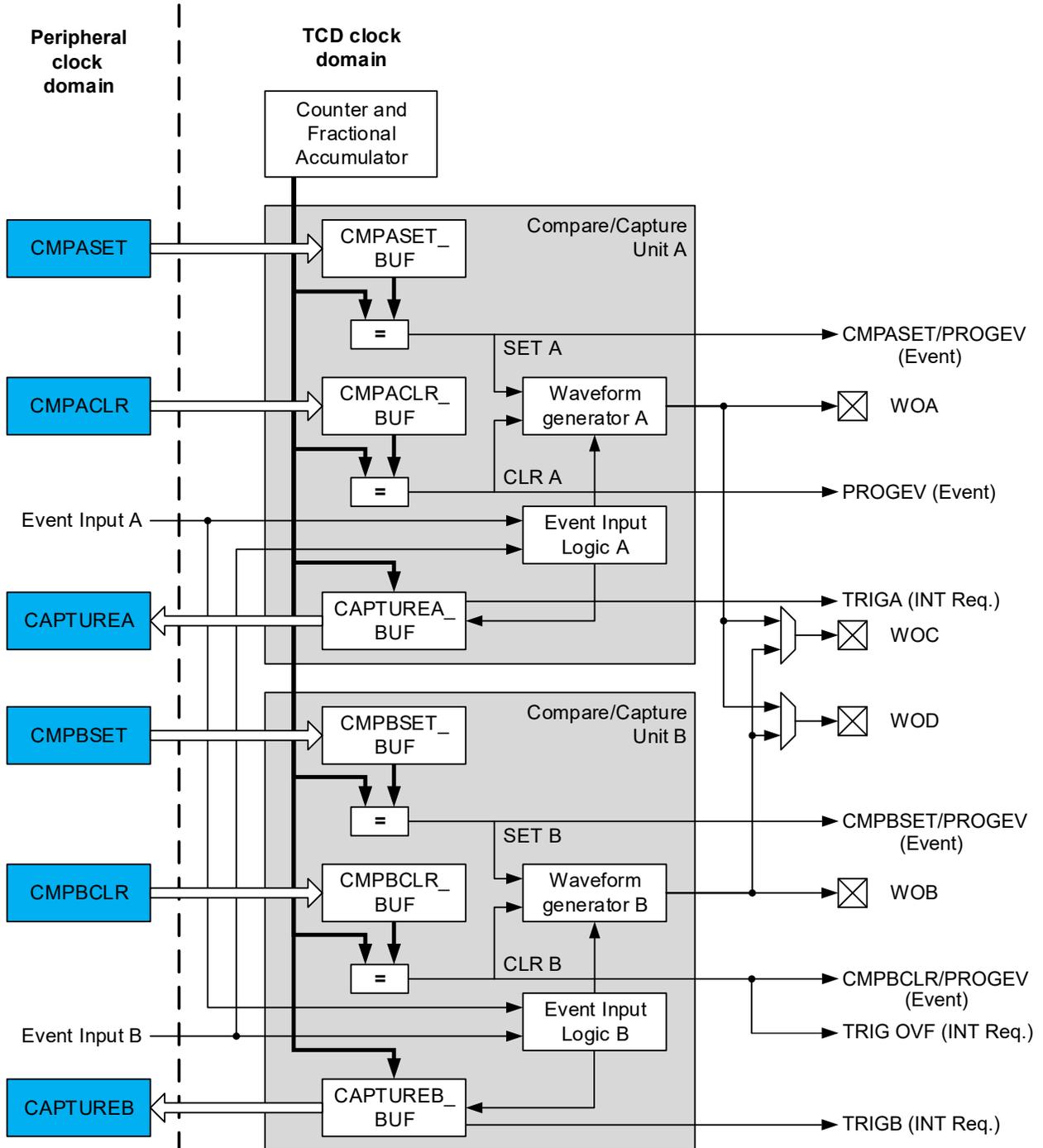
D 型定时器/计数器（TCD）是高性能波形发生器，由异步计数器、预分频器以及比较、捕捉和控制逻辑组成。

TCD 包含一个计数器，可基于与外设时钟异步的时钟运行。它包含比较逻辑，可生成两个具有可选死区的独立输出。该 TCD 连接到事件系统以进行捕捉和确定性故障控制。定时器/计数器可在发生比较匹配和溢出时生成中断和事件。

该器件提供一个 TCD 外设实例，即 TCDO。

21.2.1. 框图

图 21-1. 定时器/计数器框图



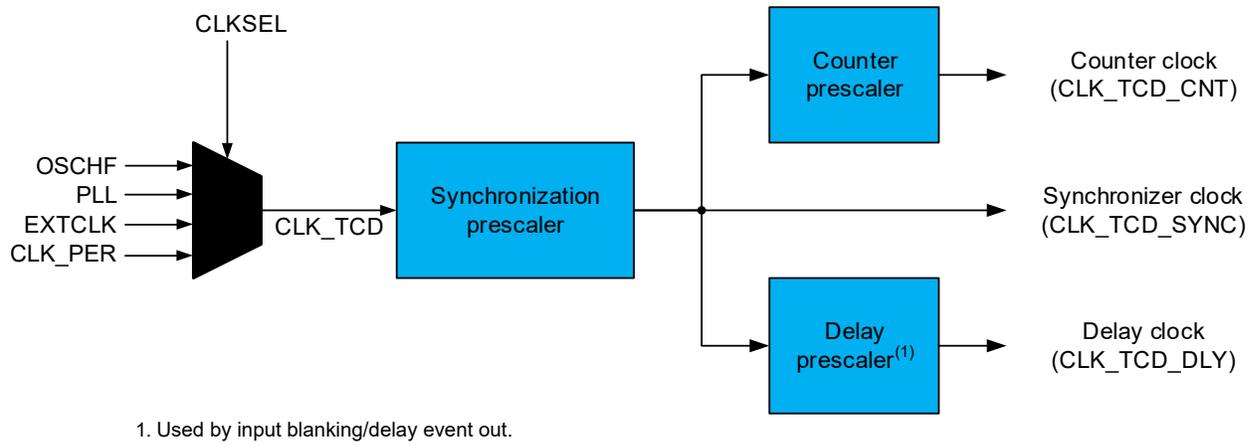
TCD 内核与外设时钟异步。定时器/计数器由两个比较/捕捉单元组成，每个单元都具有单独的波形输出。还有两个额外的波形输出，可与其中一个单元的输出相等。每个比较/捕捉单元都有一对比较寄存器，存储在各自的外设寄存器（TCDn.CMPASET、TCDn.CMPACLR、TCDn.CMPBSET 和 TCDn.CMPBCLR）中。

在正常工作期间，计数器值不断与比较寄存器的值进行比较，用于生成中断和事件。

TCD 可在 10 种不同输入模式下使用输入事件（单独为两个输入事件选择输入模式）。输入模式定义输入事件对输出的影响方式，以及在 TCD 周期中事件发生时计数器的行为。

TCD 可以在四个可预分频的不同时钟源之间进行选择。有三个可单独控制的不同预分频器，如下所示。

图 21-2. 时钟选择和预分频器概览



TCD 同步器时钟独立于其他模块时钟，因此可提高 TCD 域与 I/O 域之间的同步速度。

计数器的总预分频比为：

$$\text{SYNCPRESC_division_factor} \times \text{CNTPRESC_division_factor}$$

对于用于输入消隐/延时事件输出功能的时钟，可以使用延时预分频器进行预分频。预分频器可以独立配置，从而允许对计数器功能进行单独的范围和精度设置。同步预分频器和计数器预分频器可通过控制 A（TCDn.CTRLA）寄存器进行配置，而延时预分频器可通过延时控制（TCDn.DLYCTRL）寄存器进行配置。

21.2.2. 信号说明

信号	说明	类型
WOA	TCD 波形输出 A	数字输出
WOB	TCD 波形输出 B	数字输出
WOC	TCD 波形输出 C	数字输出
WOD	TCD 波形输出 D	数字输出

21.3. 功能说明

21.3.1. 定义

以下定义通篇适用：

表 21-1. 定时器/计数器定义

名称	说明
TCD 周期	计数器返回到同一位置之前需经历的四状态序列
输入消隐	在 TCD 周期的可选部分中，可在一段可编程的时间内忽略事件输入。
异步输出控制	允许事件在发生时立即改写输出。它用于处理不可恢复的故障
单斜坡	在 TCD 周期内，计数器复位为零一次
双斜坡	在 TCD 周期内，计数器复位为零两次
四斜坡	在 TCD 周期内，计数器复位为零四次
双向斜坡	在 TCD 周期内，计数器在零和所选最高值之间递增和递减计数

表 21-1. 定时器/计数器定义（续）

名称	说明
输入模式	一种预定义设置，会根据给定的输入事件更改输出特性

21.3.2. 初始化

要初始化 TCD：

1. 通过控制 A（TCDn.CTRLA）寄存器选择时钟源和预分频比。
2. 通过控制 B（TCDn.CTRLB）寄存器选择波形生成模式。
3. 可选：将其他静态寄存器配置为所需的功能。
4. 将初始值写入比较（TCDn.CMPxSET/CLR）寄存器。
5. 可选：将所需值写入其他双缓冲寄存器。
6. 确保状态（TCDn.STATUS）寄存器中的使能就绪（ENRDY）位置 1。
7. 向控制 A（TCDn.CTRLA）寄存器中的 ENABLE 位写入 1，以使能 TCD。

21.3.3. 工作模式

21.3.3.1. 寄存器同步类别

大多数 I/O 寄存器需要与 TCD 内核时钟域同步。对于不同的寄存器类别，同步方式会有所不同。

表 21-2. 寄存器的类别

使能和命令寄存器	双缓冲寄存器	静态寄存器	只读寄存器	普通 I/O 寄存器
TCDn.CTRLA (ENABLE 位)	TCDn.DLYCTRL	TCDn.CTRLA ⁽¹⁾ (除 ENABLE 位外的所有位)	TCDn.STATUS	TCDn.INTCTRL
TCDn.CTRLE	TCDn.DLYVAL	TCDn.CTRLB	TCDn.CAPTUREA	TCDn.INTFLAGS
	TCDn.DITCTRL	TCDn.CTRLC	TCDn.CAPTUREB	
	TCDn.DITVAL	TCDn.CTRLD		
	TCDn.DBGCTRL	TCDn.EVCTRLA		
	TCDn.CMPASET	TCDn.EVCTRLB		
	TCDn.CMPACLAR	TCDn.INPUTCTRLA		
	TCDn.CMPBSET	TCDn.INPUTCTRLB		
	TCDn.CMPBCLR	TCDn.FAULTCTRL ⁽²⁾		

注：

1. 控制 A（TCDn.CTRLA）寄存器中的位受使能保护，但 ENABLE 位除外。只有在先向 ENABLE 写入 0 时才能写入这些位。
2. 该寄存器受配置更改保护机制保护，需要通过定时写入过程来更改其值设置。

使能和命令寄存器

由于时钟域之间的同步，只有在状态（TCDn.STATUS）寄存器中的使能就绪（ENRDY）位为 1 时才能更改控制 A（TCDn.CTRLA）寄存器中的 ENABLE 位。

当 TCD 使能时，只要没有正在进行同步，控制 E（TCDn.CTRLE）寄存器就可以自动同步到 TCD 内核域。检查 TCDn.STATUS 寄存器中的命令就绪（CCMDRDY）位是否为 1 以确保可以发出新的命令。TCDn.CTRLE 为选通寄存器，命令发出时将自行清零。

双缓冲寄存器

当 TCD 使能并且两个时钟域之间没有正在进行同步时，可以通过正常的 I/O 写操作来更新双缓冲寄存器。检查 TCDn.STATUS 寄存器中的 CMDRDY 位是否为 1 以确保可以更新双缓冲 I/O 寄存器。当发送同步命令或使能 TCD 时，这些值将与 TCD 内核域同步。

表 21-3. 发出同步命令

同步发出位	双寄存器更新
CTRLC.AUPDATE	每次写入 TCDn.CMPBCLR 寄存器时，同步会在 TCD 周期结束时发生
CTRLC.SYNC ⁽¹⁾	在 SYNC 位与 TDC 域同步后立即发生一次
CTRLC.SYNCEOC ⁽¹⁾	在下一个 TCD 周期结束时发生一次

注：

1. 如果正在进行同步，该操作不起作用。

静态寄存器

使能 TCD 后无法更新静态寄存器。因此，必须在使能 TCD 之前配置这些寄存器。要查看 TCD 是否使能，可检查 TCDn.CTRLA 寄存器中的 ENABLE 位是否读为 1。

普通 I/O 和只读寄存器

普通 I/O 和只读寄存器不受域之间任何同步的限制。只读寄存器会通知同步状态以及从内核域同步的值。

21.3.3.2. 波形生成模式

TCD 提供四种不同的波形生成模式，它们由控制 B（TCDn.CTRLB）寄存器中的波形生成模式（WGMODE）位域控制。波形生成模式包括：

- 单斜坡模式
- 双斜坡模式
- 四斜坡模式
- 双斜率模式

波形生成模式决定了计数器在 TCD 周期内如何计数，以及比较值如何影响波形。TCD 周期分为以下几种状态：

- 死区时间 WOA（DTA）
- 导通时间 WOA（OTA）
- 死区时间 WOB（DTB）
- 导通时间 WOB（OTB）

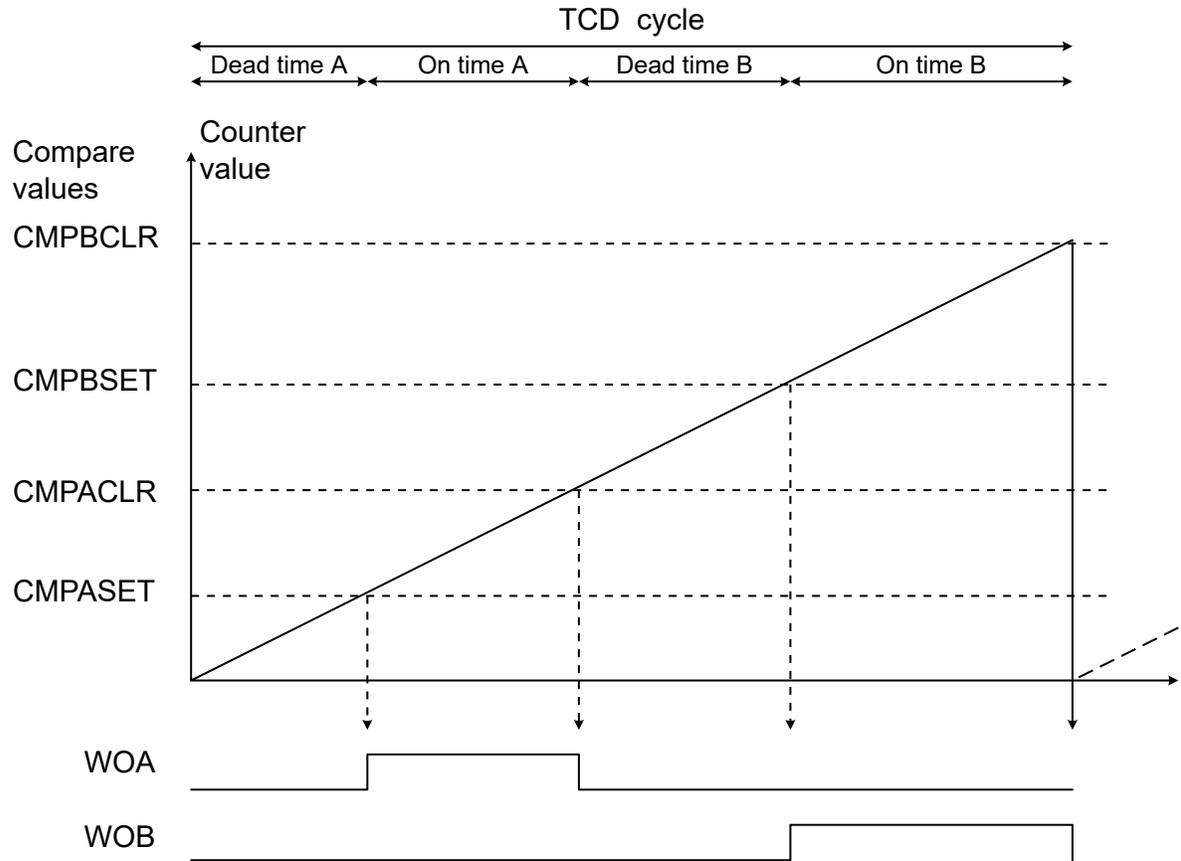
比较值“比较 A 置 1”（CMPASET）、“比较 A 清零”（CMPACLRL）、“比较 B 置 1”（CMPBSET）和“比较 B 清零”（CMPBCLR）定义了各状态的结束时间和下一个状态的开始时间。

21.3.3.2.1. 单斜坡模式

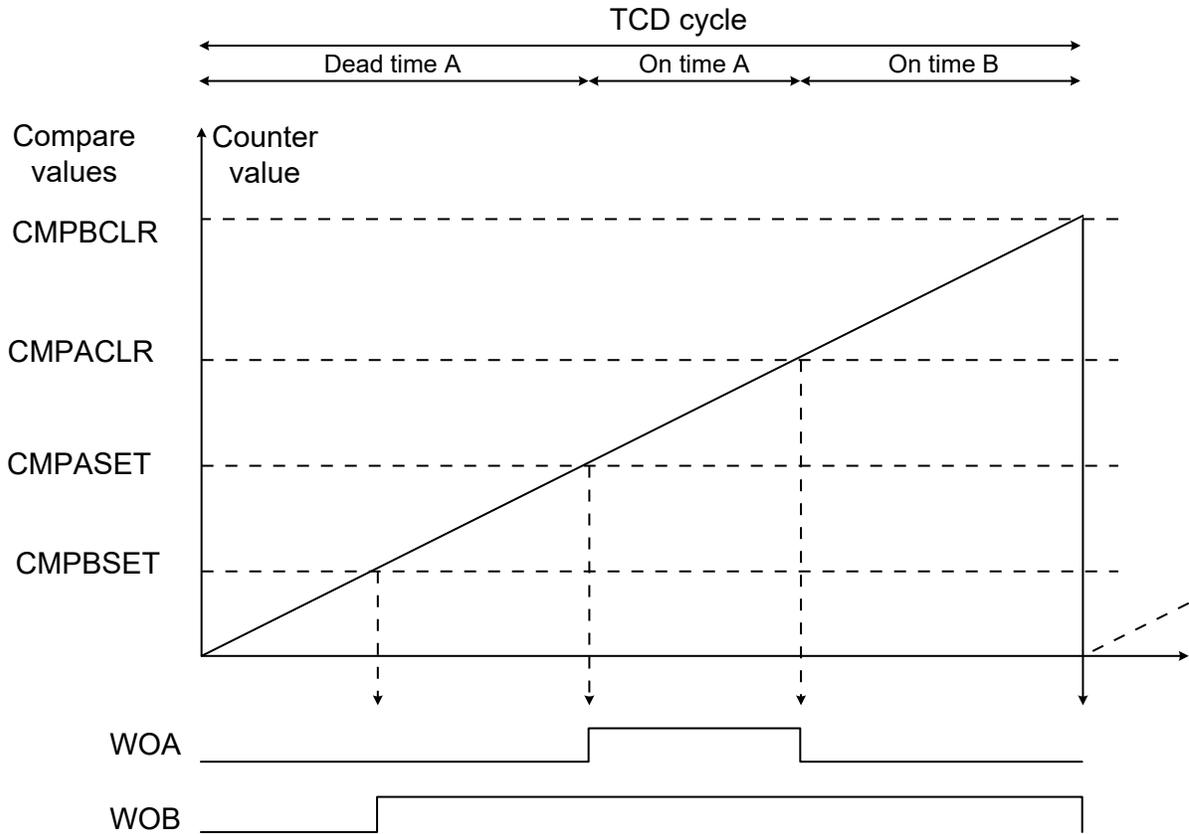
在单斜坡模式下，TCD 计数器向上计数直至达到 CMPBCLR 值。随后，TCD 周期完成并且计数器从 0x000 重新启动，开始新的 TCD 周期。TCD 周期的计算公式如下：

$$T_{\text{TCD_cycle}} = \frac{(\text{CMPBCLR} + 1)}{f_{\text{CLK_TCD_CNT}}}$$

图 21-3. 单斜坡模式



在上图中， $CMPASET < CMPACLR < CMPBSET < CMPBCLR$ 。在单斜坡模式下，为避免导通期间输出重叠，必须符合此要求。下图是 $CMPBSET < CMPASET < CMPACLR < CMPBCLR$ 导致导通期间输出重叠的示例。

图 21-4. $CMPBSET < CMPASET$ 时的单斜坡模式

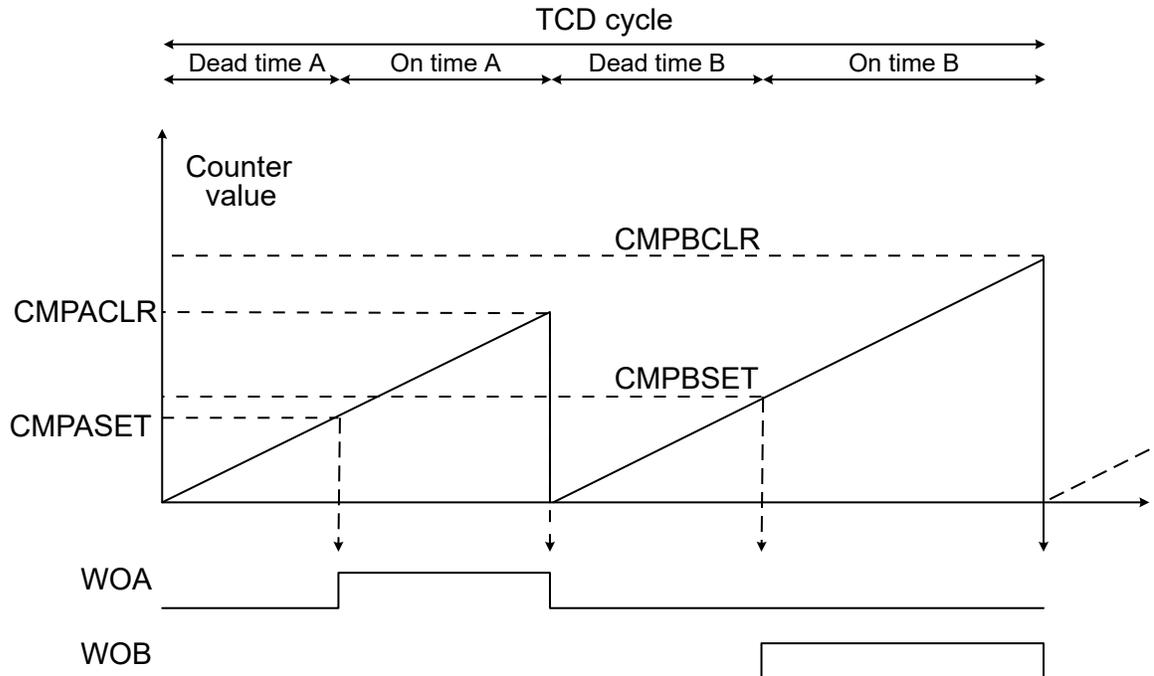
与 $CMPBCLR$ 匹配始终将导致所有输出清零。如果任意其他比较值大于 $CMPBCLR$ ，则其始终没有相关作用。如果 $CMPACLR$ 小于 $CMPASET$ 值，则清零值没有任何作用。

21.3.3.2.2. 双斜坡模式

在双斜坡模式下，TCD 计数器向上计数直至达到 $CMPACLR$ 值，然后复位并向上计数直至达到 $CMPBCLR$ 值。随后，TCD 周期完成并且计数器从 0×000 重新启动，开始新的 TCD 周期。TCD 周期的计算公式如下：

$$T_{TCD_cycle} = \frac{(CMPACLR + 1 + CMPBCLR + 1)}{f_{CLK_TCD_CNT}}$$

图 21-5. 双斜坡模式



在上图中， $CMPASET < CMPACLR$ 且 $CMPBSET < CMPBCLR$ 。这会引引起输出变为高电平。与 $CMPBSET$ 和 $CMPBCLR$ 值相比， $CMPASET$ 和 $CMPACLR$ 没有任何限制。

在双斜坡模式下，不会出现重叠输出（不使用改写功能）。即使 $CMPASET/CMPBSET > CMPACLR/CMPBCLR$ ，计数器也会在达到 $CMPACLR/CMPBCLR$ 时复位，始终不会达到 $CMPASET/CMPBSET$ 。

21.3.3.2.3. 四斜坡模式

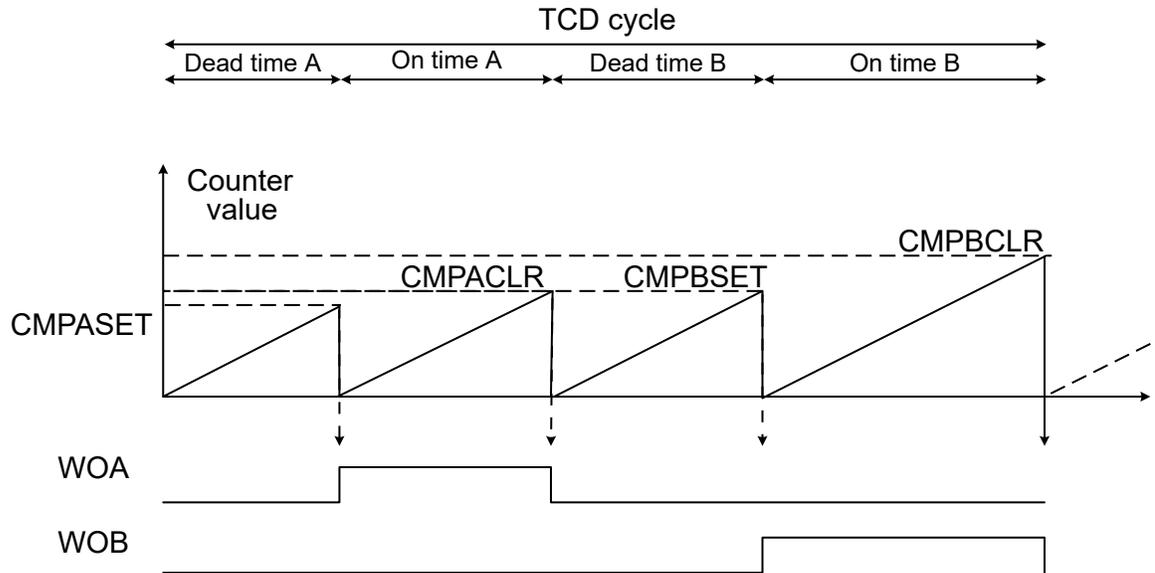
在四斜坡模式下，TCD 周期遵循以下模式：

1. TCD 周期始于 TCD 计数器从零开始向上计数，直至达到 $CMPASET$ 值，然后复位为零。
2. 计数器向上计数，直至达到 $CMPACLR$ 值，然后复位为零。
3. 计数器向上计数，直至达到 $CMPBSET$ 值，然后复位为零。
4. 计数器向上计数，直至达到 $CMPBCLR$ 值，然后复位为零以结束 TCD 周期。

TCD 周期的计算公式如下：

$$T_{TCD_cycle} = \frac{(CMPASET + 1) + (CMPACLR + 1) + (CMPBSET + 1) + (CMPBCLR + 1)}{f_{CLK_TCD_CNT}}$$

图 21-6. 四斜率模式



比较值没有限制，因为它们之间没有相关性。

在四斜率模式下，不会出现重叠输出（不使用改写功能）。

21.3.3.2.4. 双斜率模式

在双斜率模式下，TCD 周期包含 TCD 计数器从 CMPBCLR 值开始向下计数至零，再向上计数至 CMPBCLR 值的时间。TCD 周期的计算公式如下：

$$T_{\text{TCD_cycle}} = \frac{2 \times (\text{CMPBCLR} + 1)}{f_{\text{CLK_TCD_CNT}}}$$

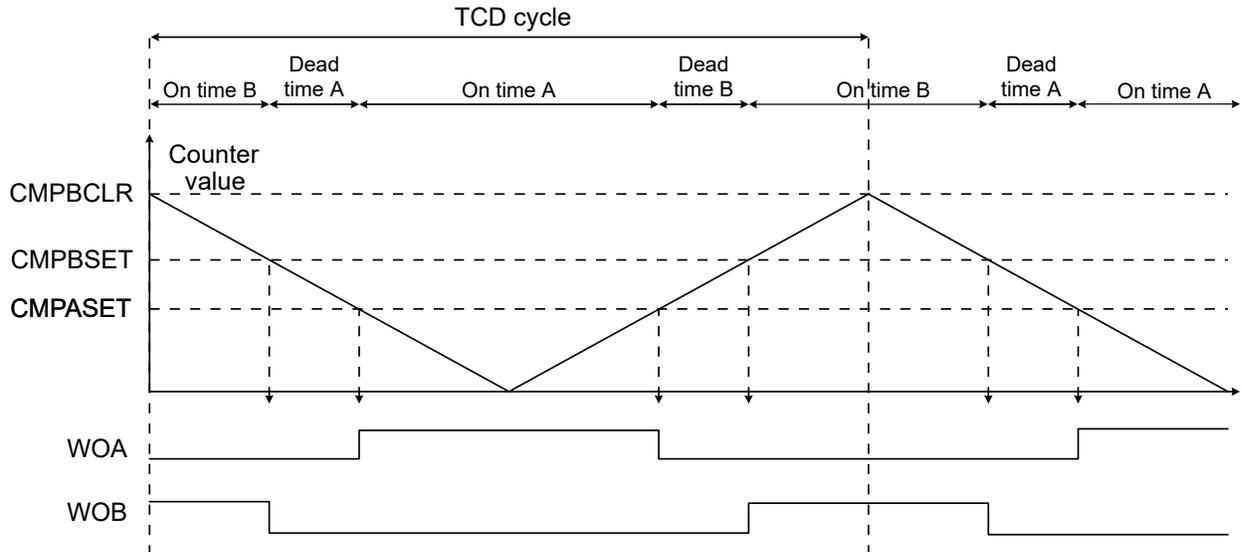
当 TCD 计数器向下计数并且与 CMPASET 值匹配时，WOA 输出置 1。当 TCD 计数器向上计数并且与 CMPASET 值匹配时，WOA 清零。

当 TCD 计数器向上计数并且与 CMPBSET 值匹配时，WOB 输出置 1。当 TCD 计数器向下计数并且与 CMPBSET 值匹配时，WOB 清零。

如果 $\text{CMPASET} > \text{CMPBSET}$ ，则输出将发生重叠。

双斜率模式不使用 CMPACL R。向 CMPACL R 写入值没有任何作用。

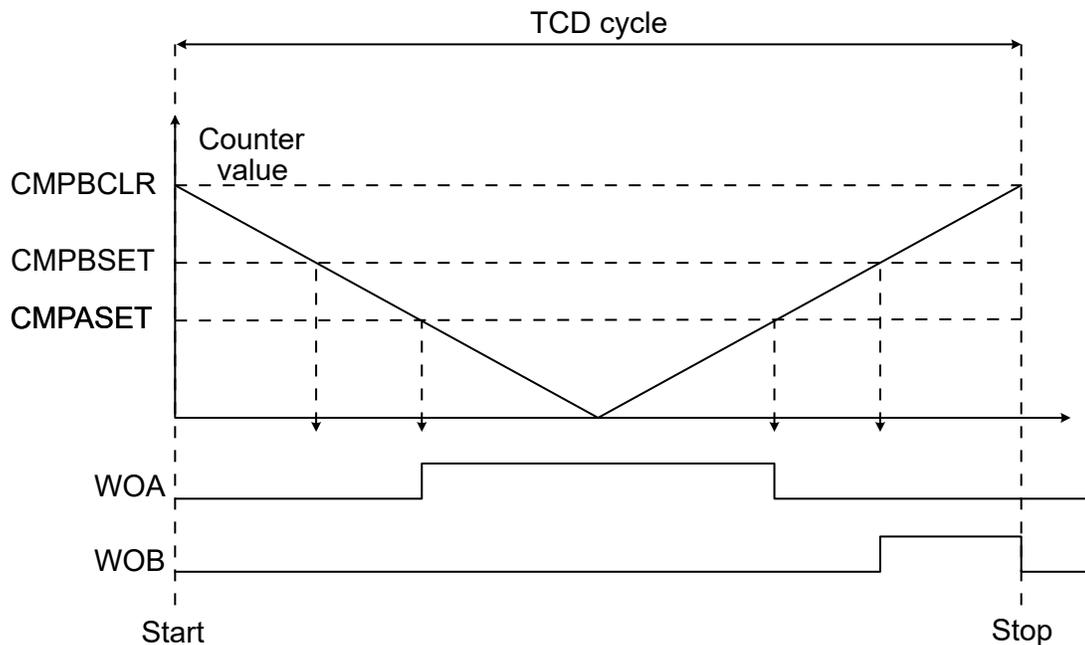
图 21-7. 双斜率模式



在双斜率模式下启动 TCD 时，TCD 计数器从 CMPBCLR 值开始向下计数。在第一个周期中，当向上计数时，在 TCD 计数器与 CMPBSET 值匹配之前，WOB 都不会置 1。

当控制 E (TCDn.CTRLE) 寄存器中的周期选通信号结束时禁止 (DISEOC) 位置 1 时，将在 TCD 周期结束时自动禁止 TCD。

图 21-8. 双斜率模式的开始和结束



21.3.3.3. 禁止 TCD

可以通过两种不同的方式禁止 TCD：

1. 向控制 A (TCDn.CTRLA) 寄存器中的 ENABLE 位写入 0。该操作将在同步至 TCD 内核域时立即禁止 TCD。

- 向控制 E (TCDn.CTRLE) 寄存器中的周期选通信号结束时禁止 (DISEOC) 位写入 1。该操作将在 TCD 周期结束时禁止 TCD。

21.3.3.4. TCD 输入

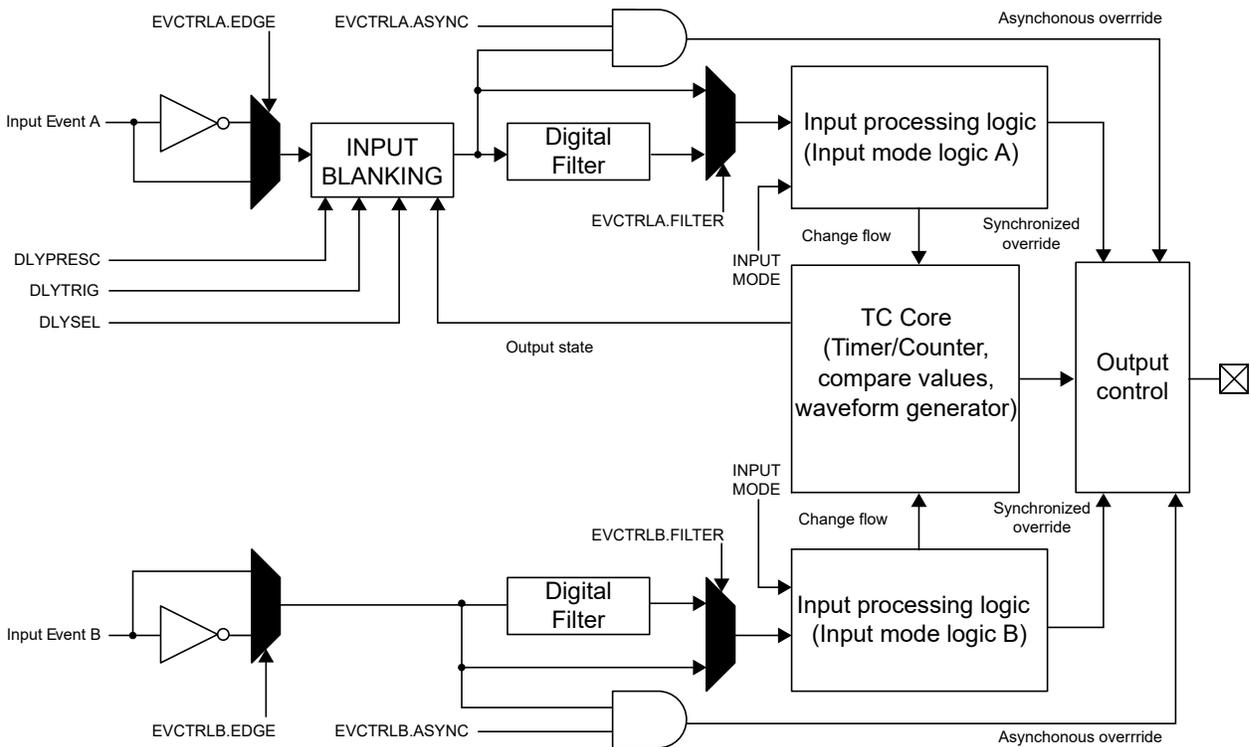
TCD 有两个与事件系统相连的输入，输入 A 和输入 B。每个输入都具有连接到相应输出 (WOA 和 WOB) 的功能。该功能由事件控制 (TCDn.EVCTRLA 和 TCDn.EVCTRLB) 寄存器和输入控制 (TCDn.INPUTCTRLA 和 TCDn.INPUTCTRLB) 寄存器控制。

要启用输入事件，可以向相应事件控制 (TCDn.EVCTRLA 或 TCDn.EVCTRLB) 寄存器中的触发事件输入使能 (TRIGEI) 位写入 1。默认情况下，输入将用作故障检测信号，但也可用作捕捉触发信号。要启用捕捉触发，可以向相应事件控制 (TCDn.EVCTRLA 或 TCDn.EVCTRLB) 寄存器中的 ACTION 位写入 1。要禁止故障检测，必须向相应输入控制 (TCDn.INPUTCTRLA 或 TCDn.INPUTCTRLB) 寄存器中的 INPUTMODE 位域写入 0。

故障检测有十种不同的输入模式。两个输入具有相同的功能，但输入消隐除外，只有输入 A 支持输入消隐。输入消隐由延时控制 (TCDn.DLYCTRL) 寄存器和延时值 (TCDn.DLYVAL) 寄存器配置。

输入与事件系统相连。必须在事件系统中配置事件源和 TCD 输入之间的连接。

图 21-9. TCD 输入概览



在接收输入事件、处理该事件和改写输出之间，TCD 同步器时钟有 2/3 个时钟周期的延时。如果使用异步事件检测，则输出将在输入处理之外立即改写。

21.3.3.4.1. 输入消隐

输入消隐功能用于在 TCD 周期的可选部分将输入事件屏蔽掉一段可编程的时间。输入消隐可用于屏蔽在输出发生变化后立即触发的“错误”输入事件。

可通过配置延时控制 (TCDn.DLYCTRL) 寄存器中的延时选择 (DLYSEL) 位域来使能输入消隐。触发源由 TCDn.DLYCTRL 中的延时触发 (DLYTRIG) 位域选择。

输入消隐使用延时时钟。触发后，计数器向上计数，直到达到延时值（TCDn.DLYVAL）寄存器中的延时值（DLYVAL）位域。随后，输入消隐被关闭。TCD 延时时钟是同步器时钟（CLK_TCD_SYNC）的预分频版本。分频系数由延时控制（TCDn.DLYCTRL）寄存器中的延时预分频器（DLYPRESC）位域设置。输入消隐持续时间的计算公式如下：

$$t_{\text{BLANK}} = \frac{\text{DLYPRESC_division_factor} \times \text{DLYVAL}}{f_{\text{CLK_TCD_SYNC}}}$$

输入消隐使用与可编程输出事件相同的逻辑。因此，二者无法同时使用。

21.3.3.4.2. 数字滤波器

通过向相应事件控制（TCDn.EVCTRLA 或 TCDn.EVCTRLB）寄存器中的 FILTER 位写入 1 来使能事件输入 x 的数字滤波器。当数字滤波器被使能时，任何持续时间少于四个计数器时钟周期的脉冲都将被滤除。因此，对传入事件的任何更改都将需要四个计数器时钟周期才能影响输入处理逻辑。

21.3.3.4.3. 异步事件检测

要对输入事件使能异步事件检测，必须相应地配置相应事件控制（TCDn.EVCTRLA 或 TCDn.EVCTRLB）寄存器中的事件配置（CFG）位域。

异步事件检测可以在发生输入事件时异步改写输出。输入事件的作用取决于输入模式。当事件与同步器时钟（CLK_TCD_SYNC）同步时，输出会直接改写，计数器流程也将发生变化。

无法同时使用异步事件检测和数字滤波器。

21.3.3.4.4. 软件命令

下表列出了 TCD 模块的命令。

表 21-4. 软件命令

触发信号	软件命令
TCDn.CTRLA 寄存器中的 SYNCEOC 位	在 TCD 周期结束时更新双缓冲寄存器
TCDn.CTRLA 寄存器中的 SYNC 位	更新双缓冲寄存器
TCDn.CTRLA 寄存器中的 RESTART 位	重启 TCD 计数器
TCDn.CTRLA 寄存器中的 SCAPTUREA 位	捕捉到捕捉 A（TCDn.CAPTUREA/H）寄存器
TCDn.CTRLA 寄存器中的 SCAPTUREB 位	捕捉到捕捉 B（TCDn.CAPTUREB/H）寄存器

21.3.3.4.5. 输入模式

用户可以在 10 种输入模式之间进行选择，具体通过写入输入控制（TCDn.INPUTCTRLA 和 TCDn.INPUTCTRLB）寄存器中的输入模式（INPUTMODE）位域来完成。

输入模式有效性

并非所有输入模式下都可以使用所有波形生成模式。下表列出了不同输入模式下的有效波形生成模式。

表 21-5. 输入模式有效性

INPUTMODE	单斜坡模式	双斜坡模式	四斜坡模式	双斜率模式
0	有效	有效	有效	有效
1	有效	有效	有效	不要使用
2	不要使用	有效	有效	不要使用
3	不要使用	有效	有效	不要使用
4	有效	有效	有效	有效
5	不要使用	有效	有效	不要使用
6	不要使用	有效	有效	不要使用
7	有效	有效	有效	有效
8	有效	有效	有效	不要使用
9	有效	有效	有效	不要使用

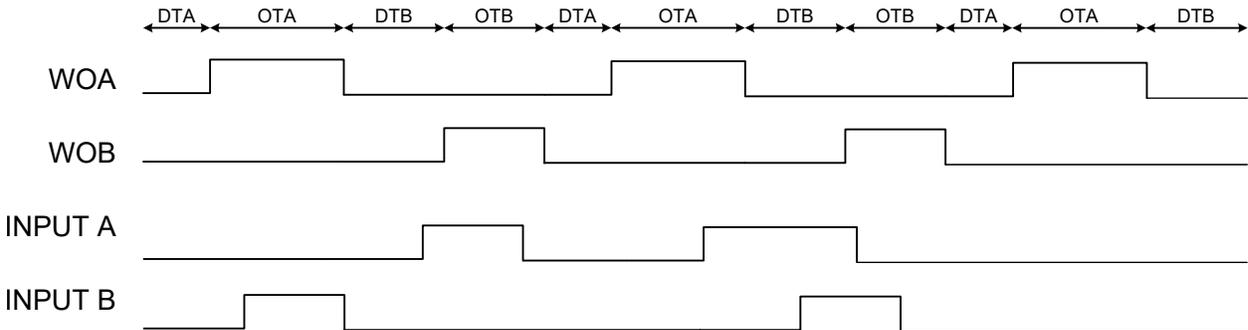
表 21-5. 输入模式有效性（续）

INPUTMODE	单斜坡模式	双斜坡模式	四斜坡模式	双斜率模式
10	有效	有效	有效	不要使用

输入模式 0：输入无动作

在输入模式 0 下，输入不会影响输出，但仍可触发捕捉和中断（如果已允许）。

图 21-10. 输入模式 0

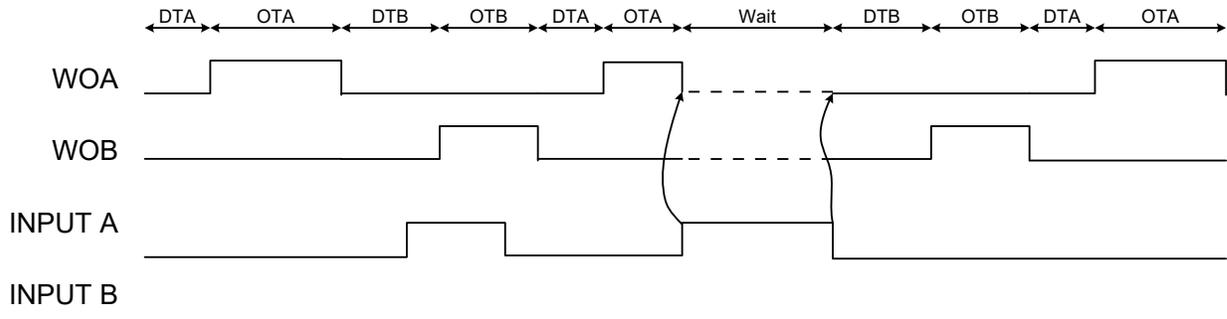


输入模式 1：停止输出，跳转到相反的比较周期，然后等待

输入模式 1 下的输入事件将停止输出信号，跳转到相反的死区时间，并等到输入事件变为低电平，然后 TCD 计数器继续工作。

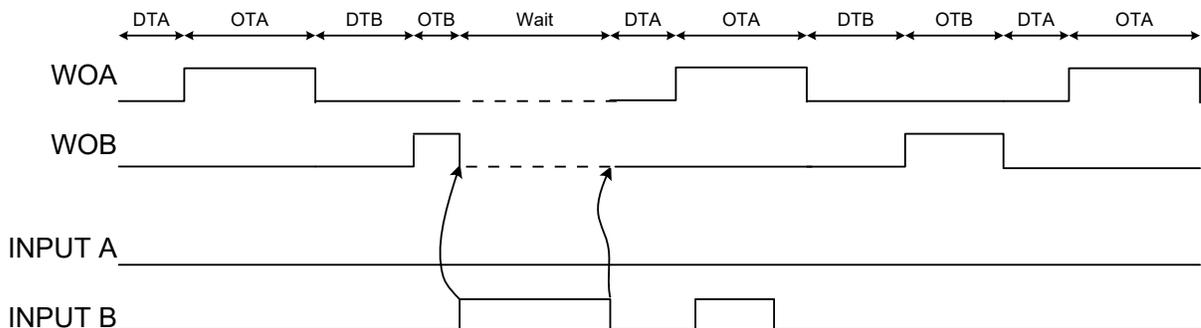
如果对输入 A 使用输入模式 1，则只有当 TCD 处于死区时间 A 或导通时间 A 时，事件才会生效，并且事件只会影响 WOA 输出。当事件完成时，TCD 计数器从死区时间 B 开始工作。

图 21-11. 对输入 A 使用输入模式 1



如果对输入 B 使用输入模式 1，则只有当 TCD 处于死区时间 B 或导通时间 B 时，事件才会生效，并且事件只会影响 WOB 输出。当事件完成时，TCD 计数器从死区时间 A 开始工作。

图 21-12. 对输入 B 使用输入模式 1

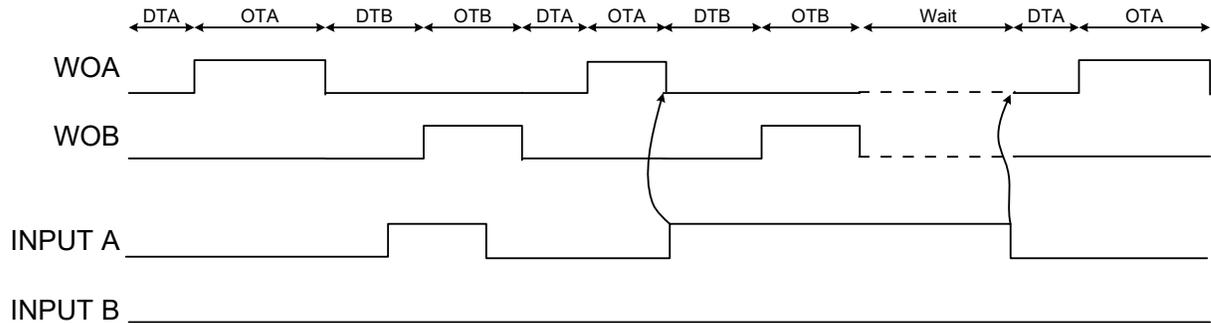


输入模式 2：停止输出，执行相反的比较周期，然后等待

输入模式 2 下的输入事件将停止输出信号，执行到相反的死区时间和导通时间，并等到输入事件变为低电平，然后 TCD 计数器继续工作。如果输入在相反的死区时间和导通时间结束之前完成，则无需等待，但相反的死区时间和导通时间将继续。

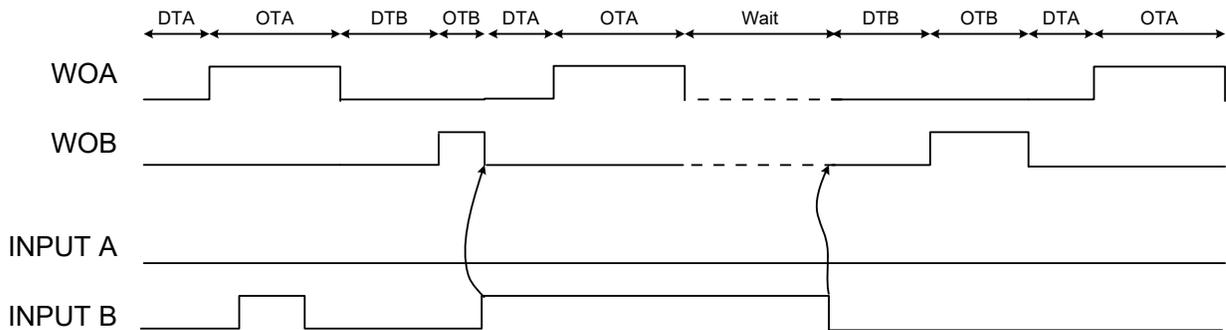
如果对输入 A 使用输入模式 2，则只有当 TCD 处于死区时间 A 或导通时间 A 时，事件才会生效，并且事件只会影响 WOA 输出。

图 21-13. 对输入 A 使用输入模式 2



如果对输入 B 使用输入模式 2，则只有当 TCD 处于死区时间 B 或导通时间 B 时，事件才会生效，并且事件只会影响 WOB 输出。

图 21-14. 对输入 B 使用输入模式 2

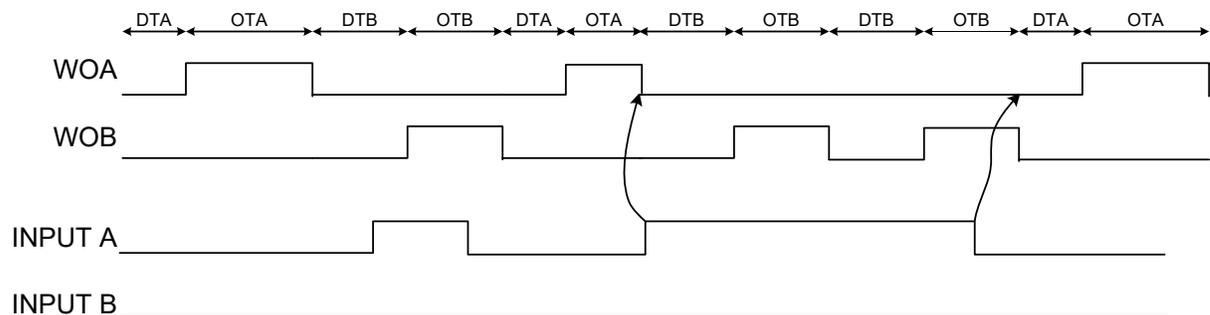


输入模式 3：停止输出，在故障有效时执行相反的比较周期

只要故障/输入有效，输入模式 3 下的输入事件就会停止输出信号并开始重复执行相反的死区时间和导通时间。当输入释放时，进行中的死区时间和/或导通时间将结束，之后正常流程将开始。

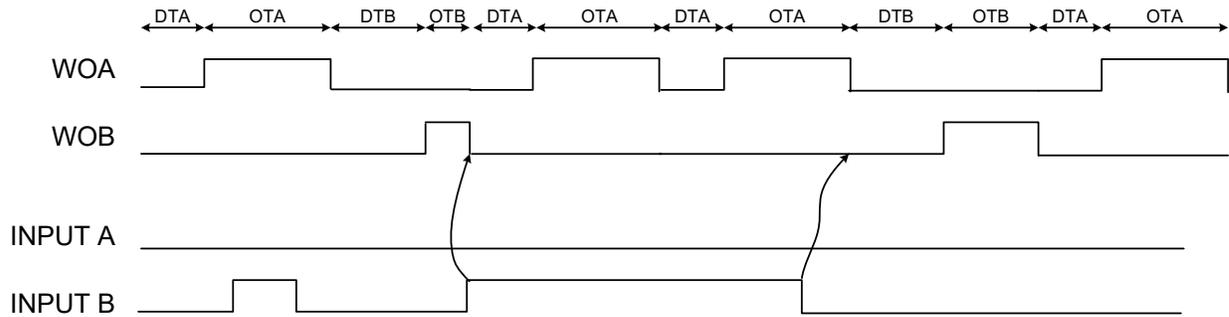
如果对输入 A 使用输入模式 3，则只有当 TCD 处于死区时间 A 或导通时间 A 时，事件才会生效。

图 21-15. 对输入 A 使用输入模式 3



如果对输入 B 使用输入模式 3，则只有当 TCD 处于死区时间 B 或导通时间 B 时，事件才会生效。

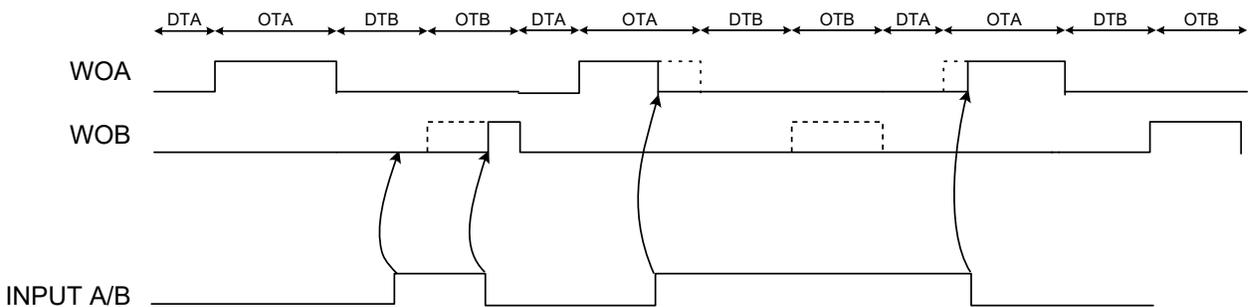
图 21-16. 对输入 B 使用输入模式 3



输入模式 4：停止所有输出，保持频率不变

当使用输入模式 4 时，输入 A 和输入 B 将提供相同的功能。只要输入事件有效，就会禁止输出。在该输入模式下，TCD 计数器不受事件的影响。

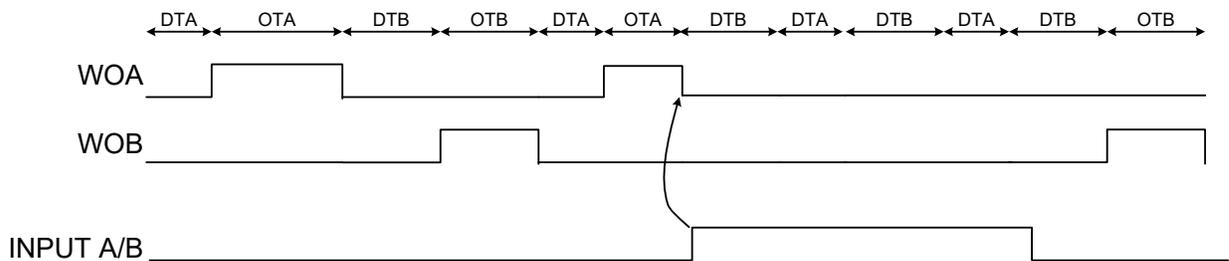
图 21-17. 输入模式 4



输入模式 5：停止所有输出，在故障有效时执行死区时间

当使用输入模式 5 时，输入 A 和输入 B 提供相同的功能。如果输入事件发生在导通时间内，则会停止输出并在相反的死区时间开始。如果事件发生在死区内，则死区时间会持续到下一段导通时间开始时。不过，如果输入仍有效，则周期将继续另一段死区时间。只要输入事件有效，就会出现备用死区时间。当输入事件停止时，进行中的死区时间将结束，下一段导通时间将以正常流程继续。

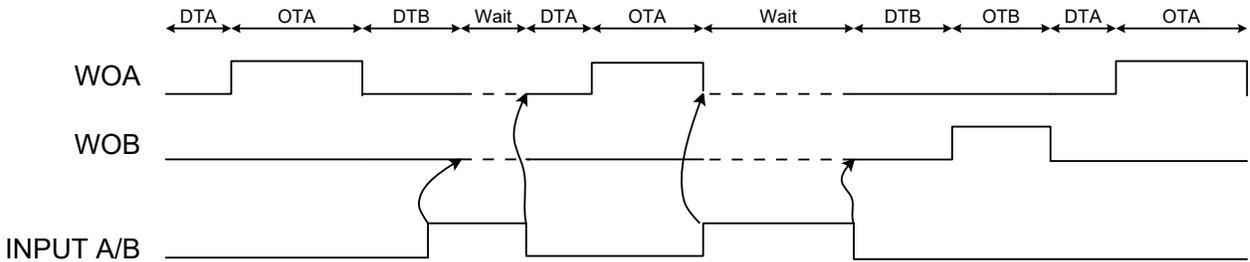
图 21-18. 输入模式 5



输入模式 6：停止所有输出，跳转到下一个比较周期，然后等待

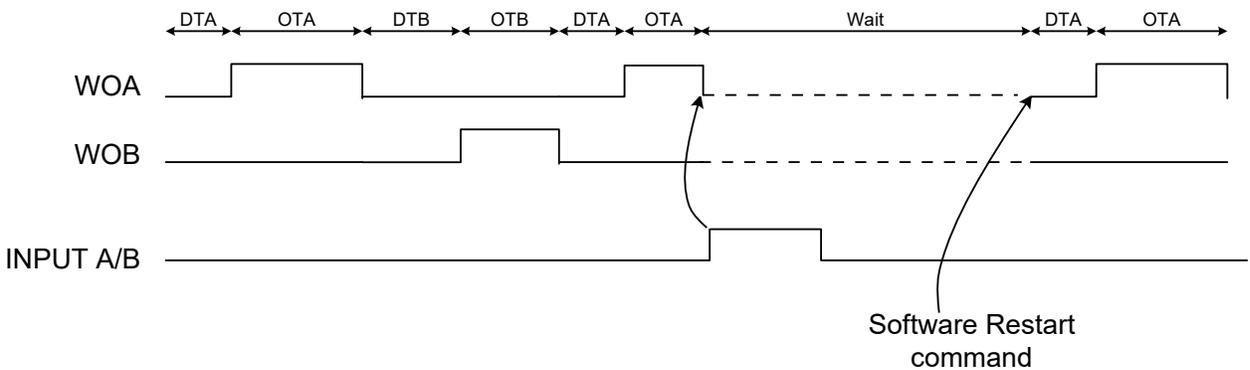
当使用输入模式 6 时，输入 A 和输入 B 将提供相同的功能。如果输入事件发生在导通时间内，则会停止输出并跳转到相反的死区时间。如果事件发生在死区内，则死区时间会持续到下一段导通时间开始时。只要输入事件有效，TCD 计数器就会等待。当输入事件停止时，下一段死区时间将开始，正常流程将继续。

图 21-19. 输入模式 6

**输入模式 7：停止所有输出，等待软件操作**

当使用输入模式 7 时，输入 A 和输入 B 将提供相同的功能。输入事件将停止输出和 TCD 计数器，一直持续到发出重启命令。如果发出重启命令（TCDn.CTRL 寄存器中的 RESTART 位）时输入事件仍为高电平，将再次停止。当 TCD 计数器重启时，它将始终在死区时间 A 开始。

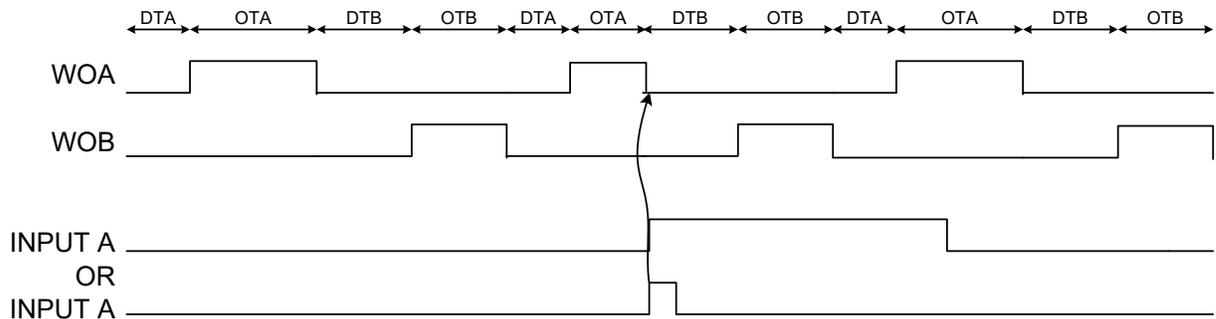
图 21-20. 输入模式 7

**输入模式 8：在出现相应边沿时停止输出，跳转到下一个比较周期**

在输入模式 8 下，如果相应输出导通时输入事件出现正边沿，则将导致输出停止，TCD 计数器跳转到相反的死区时间。

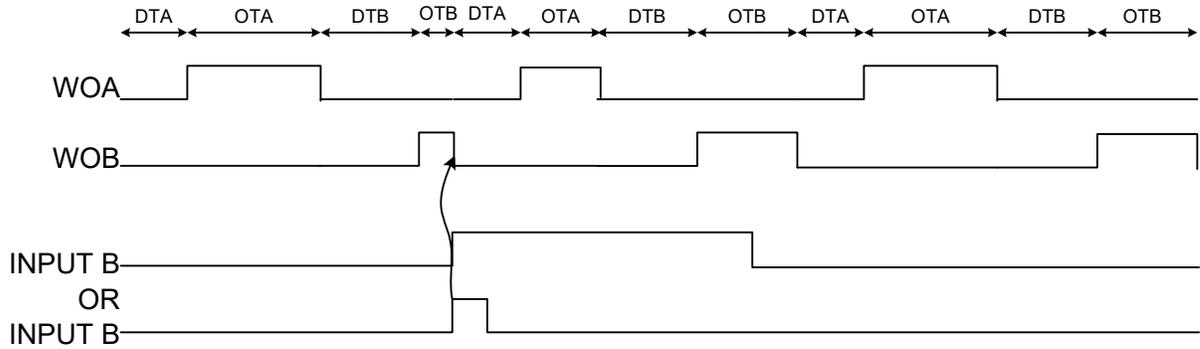
如果对输入 A 使用输入模式 8 并且在导通时间 A 内输入事件出现正边沿，则 TCD 计数器将跳转到死区时间 B。

图 21-21. 对输入 A 使用输入模式 8



如果对输入 B 使用输入模式 8 并且在导通时间 B 内输入事件出现正边沿，则 TCD 计数器将跳转到死区时间 A。

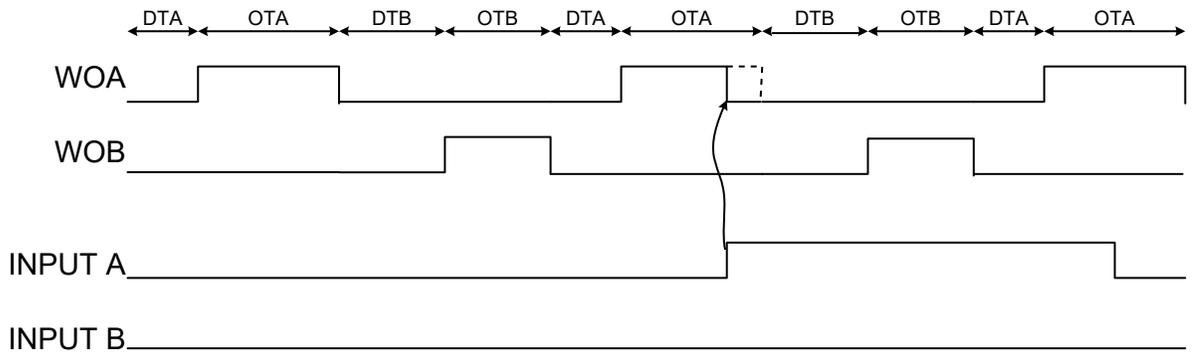
图 21-22. 对输入 B 使用输入模式 8

**输入模式 9：在出现相应边沿时停止输出，保持频率不变**

在输入模式 9 下，如果相应输出导通时输入事件出现正边沿，则将导致输出在剩余导通时间内停止。TCD 计数器不受事件的影响，只有输出受影响。

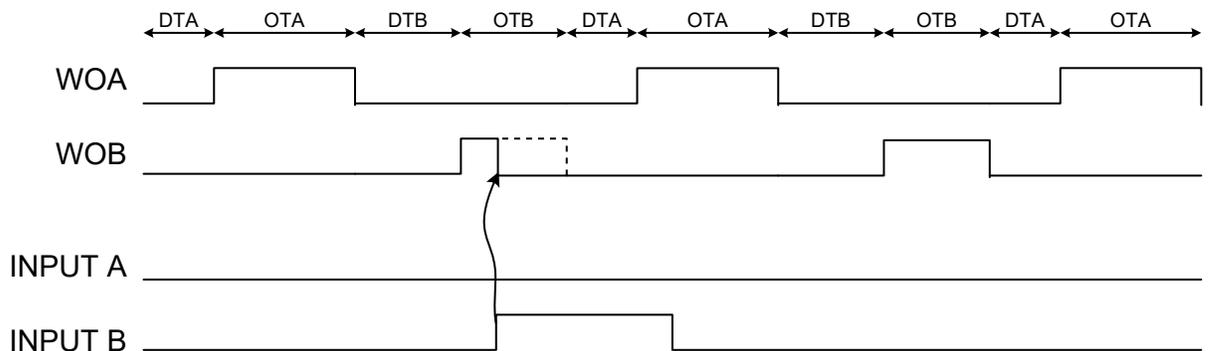
如果对输入 A 使用输入模式 9 并且在导通时间 A 内输入事件出现正边沿，则输出将在剩余导通时间内关断。

图 21-23. 对输入 A 使用输入模式 9



如果对输入 B 使用输入模式 9 并且在导通时间 B 内输入事件出现正边沿，则输出将在剩余导通时间内关断。

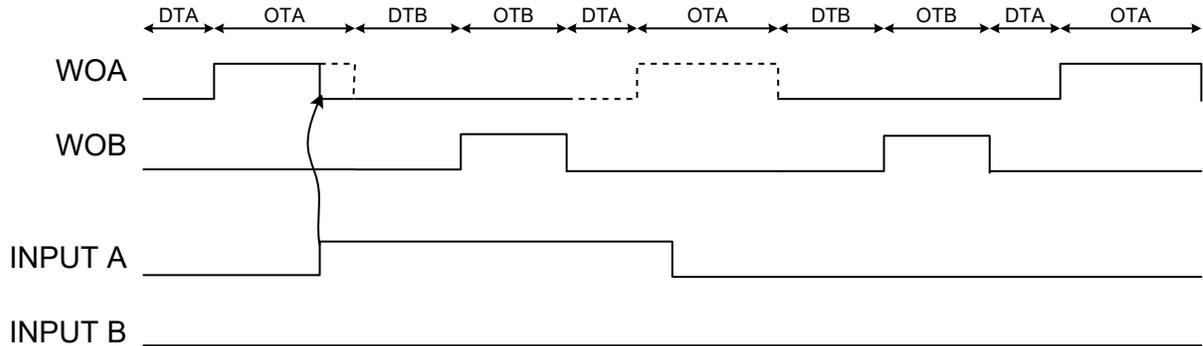
图 21-24. 对输入 B 使用输入模式 9

**输入模式 10：在出现相应电平时停止输出，保持频率不变**

在输入模式 10 下，只要输入有效，输入事件就会使相应的输出停止。如果相应输出上必须有一段导通时间时输入变为低电平，则输出将在剩余导通时间内禁止。TCD 计数器不受事件的影响，只有输出受影响。

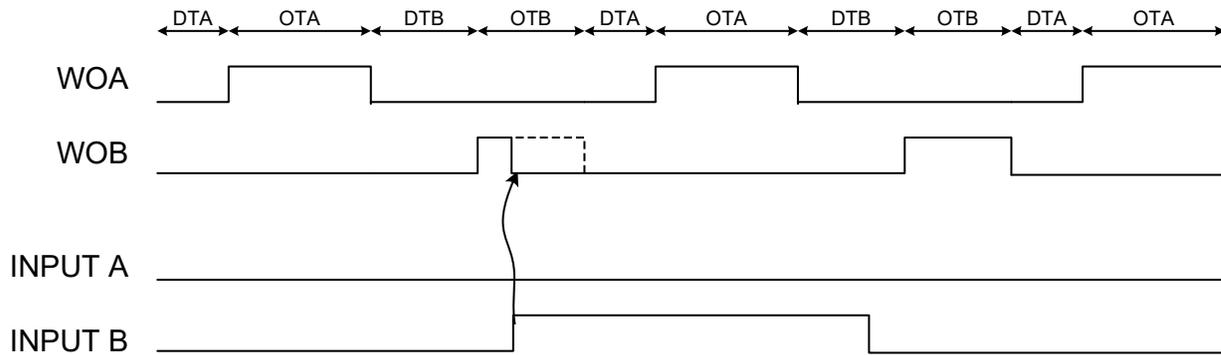
如果对输入 A 使用输入模式 10 并且发生输入事件，则只要事件持续，WOA 就会关断。如果在导通时间内释放，则它将在剩余导通时间关断。

图 21-25. 对输入 A 使用输入模式 10



如果对输入 B 使用输入模式 10 并且发生输入事件，则只要事件持续，WOB 就会关断。如果在导通时间内释放，则它将在剩余导通时间关断。

图 21-26. 对输入 B 使用输入模式 10



输入模式汇总

表 21-6 总结了前面章节时序图中说明的条件。

表 21-6. 输入模式汇总

INPUTMODE	触发→受影响的输出	故障开启/有效	故障释放/无效
0	-	无动作	无动作
1	输入 A→WOA	结束电流导通时间，然后等待	开始死区时间，以进行其他比较
	输入 B→WOB		
2	输入 A→WOA	结束电流导通时间，执行其他比较周期，然后等待	开始死区时间，以进行电流比较
	输入 B→WOB		
3	输入 A→WOA	执行电流导通时间，然后重复执行其他比较周期	重新使能电流比较周期
	输入 B→WOB		
4	输入 A→{WOA, WOB}	禁止输出	
	输入 B→{WOA, WOB}		
5	输入 A→{WOA, WOB}	仅执行死区时间	
	输入 B→{WOA, WOB}		
6	输入 A→{WOA, WOB}	结束导通时间，然后等待	开始死区时间，以进行其他比较
	输入 B→{WOA, WOB}		

表 21-6. 输入模式汇总（续）

INPUTMODE	触发→受影响的输出	故障开启/有效	故障释放/无效
7	输入 A→{WOA, WOB}	结束导通时间, 然后等待软件操作	开始死区时间, 以进行电流比较
	输入 B→{WOA, WOB}		
8	输入 A→WOA	结束电流导通时间, 然后继续其他关断时间	
	输入 B→WOB		
9	输入 A→WOA	阻止电流导通时间, 然后继续序列	
	输入 B→WOB		
10	输入 A→WOA	当触发有效时, 禁止导通时间直到序列结束	
	输入 B→WOB		
other	-	-	-

注: 在每个事件输入上使用不同模式时, 需考虑可能发生的冲突 (请记住 TCD 有一个计数器), 以避免出现意外结果。

21.3.3.5. 抖动

如果由于预分频器/周期选择限制而无法实现所需频率, 则可以使用抖动来近似估计所需频率并减少波形漂移。

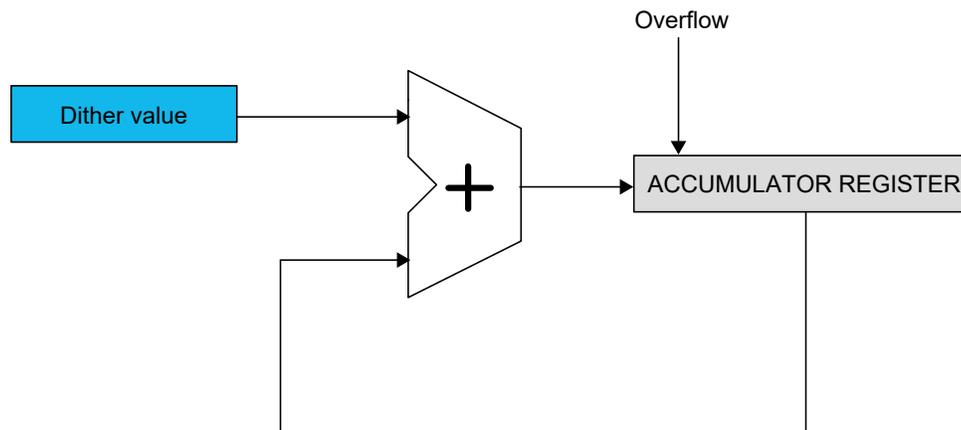
抖动累加每个周期的计数器时钟的小数误差。当小数误差溢出时, 会在 TCD 周期的选定部分增加一个额外的时钟周期。

例 21-1. 使用 10 MHz 时钟生成 75 kHz 输出

如果定时器时钟频率为 10 MHz, 它将为定时器提供 100 ns 的分辨率。所需的输出频率为 75 kHz, 这意味着输出周期为 13333 ns。该输出周期无法以 100 ns 分辨率实现, 因为它需要 133.33 个时钟周期。可以将输出周期设置为 133 个时钟周期 (75.188 kHz) 或 134 个时钟周期 (74.626 kHz)。

可以在固件中手动更改两个频率之间的周期, 以获得 75 kHz 的平均输出频率 (每隔两个输出周期后, 更改为 134 个时钟周期)。抖动可以通过累加误差 (0.33 个周期) 自动完成。累加器会计算累加误差何时大于一个时钟周期, 当发生这种情况时, 它会在定时器周期中增加一个额外的时钟周期。

图 21-27. 抖动逻辑



用户可以通过写入抖动控制 (TCDn.DITCTRL) 寄存器中的抖动选择 (DITHERSEL) 位来选择在 TCD 周期中增加抖动的位置:

- 导通时间 B
- 导通时间 A 和 B
- 死区时间 B
- 死区时间 A 和 B

抖动对 TCD 周期时间的影响取决于使用的波形生成模式（见表 21-7）。双斜率模式不支持抖动。

表 21-7. 不同模式下向 TCD 周期增加的抖动

WAVEGEN	TCDn.DITCTRL 中的 DITHERSEL	向 TCD 周期增加的 TCD 时钟周期
单斜坡模式	导通时间 B	1
	导通时间 A 和 B	1
	死区时间 B	0
	死区时间 A 和 B	0
双斜坡模式	导通时间 B	1
	导通时间 A 和 B	2
	死区时间 B	0
	死区时间 A 和 B	0
四斜坡模式	导通时间 B	1
	导通时间 A 和 B	2
	死区时间 B	1
	死区时间 A 和 B	2
双斜率模式	导通时间 B	不支持
	导通时间 A 和 B	不支持
	死区时间 B	不支持
	死区时间 A 和 B	不支持

增加到 TCD 周期的 TCD 时钟周期数会随着 TCD 周期使用的比较值数量而变化。例如，在单斜坡模式下，只有 CMPBCLR 影响 TCD 周期时间。

对于 TCD 周期未增加额外周期的 DITHERSEL 配置，通过缩短以下输出状态来达到补偿效果。

例 21-2. 单斜坡模式下的 DITHERSEL

在 DITHERSEL 选择死区时间 B 的单斜坡模式下，当发生抖动溢出时，死区时间 B 将增加一个周期，这会使导通时间 B 减少一个周期。

21.3.3.6. TCD 计数器捕捉

TCD 计数器与外设时钟异步，因此无法直接读出计数器值。可以通过两种方式来捕捉与 I/O 时钟域同步的 TCD 计数器值：

- 捕捉输入事件的值
- 软件捕捉

捕捉逻辑包含两个独立的捕捉块 CAPTUREA 和 CAPTUREB，它们可以捕捉 TCD 计数器值并将其与 I/O 时钟域同步。CAPTUREA/B 可以由输入事件 A/B 或软件触发。

可以通过先读取 TCDn.CAPTUREAL/TCDn.CAPTUREBL 寄存器然后读取 TCDn.CAPTUREAH/TCDn.CAPTUREBH 寄存器来获取捕捉值。

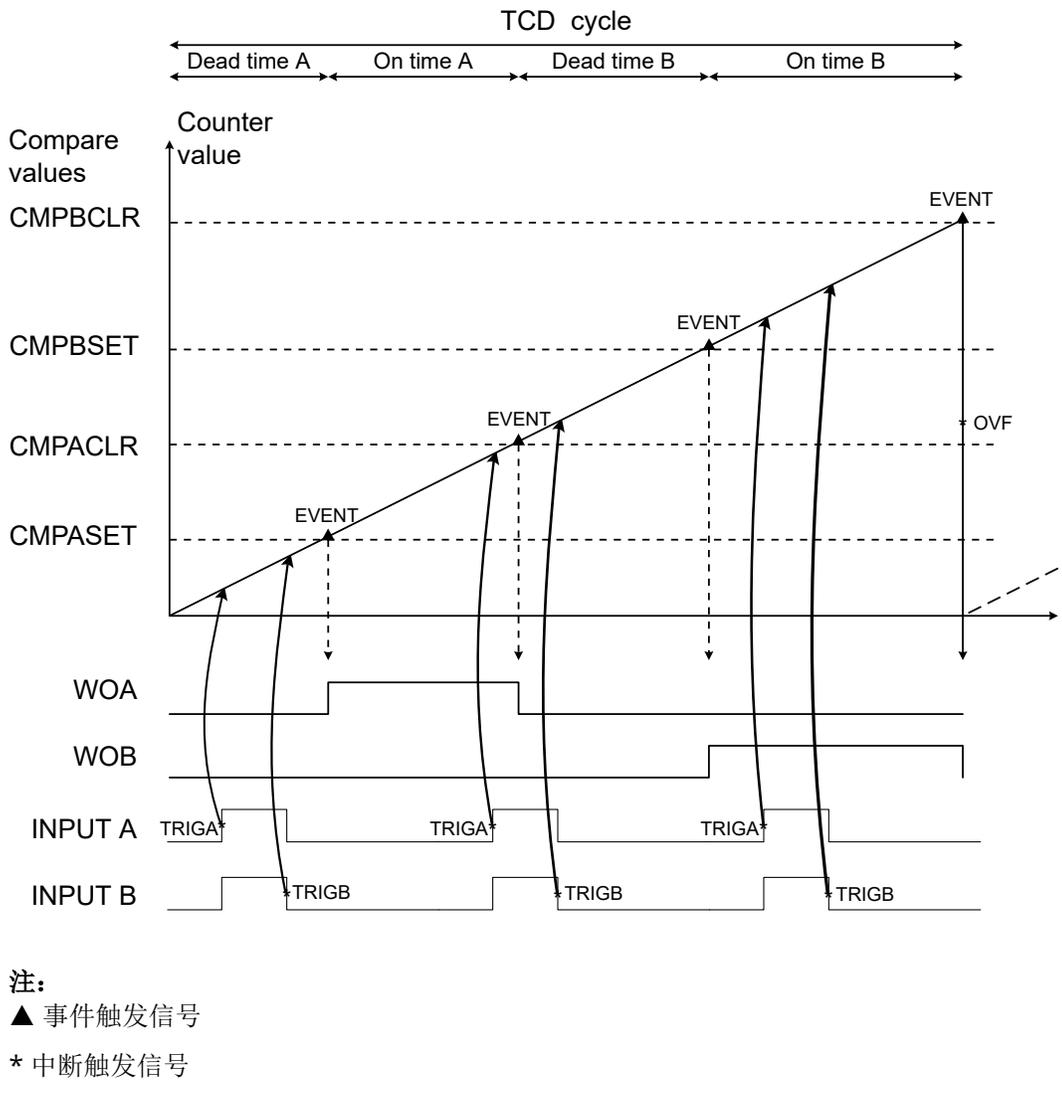
由输入事件触发的捕捉

要使能捕捉输入事件，在配置事件输入时将 1 写入相应事件控制（TCDn.EVCTRLA 或 TCDn.EVCTRLB）寄存器中的 ACTION 位。

发生捕捉时，中断标志（TCDn.INTFLAGS）寄存器中的 TRIGA/B 标志将置 1。可通过向中断控制（TCDn.INTCTRL）寄存器中相应的触发中断允许（TRIGA 或 TRIGB）位写入 1 来允许相应的 TRIGA/B 中断。通过在 TCDn.INTFLAGS 中轮询 TRIGA 或 TRIGB，用户可了解 CAPTURE 值是否可用，并且可以通过首先读取 TCDn.CAPTUREAL 或 TCDn.CAPTUREBL 寄存器然后读取 TCDn.CAPTUREAH 或 TCDn.CAPTUREBH 寄存器来读出该值。

例 21-3. PWM 捕捉

要执行 PWM 捕捉，将事件 A 和事件 B 连接到包含 PWM 信号的同一异步事件通道。要获得有关 PWM 信号的信息，将一个事件输入配置为捕捉信号的上升沿，将另一个事件输入配置为捕捉信号的下降沿。



由软件触发的捕捉

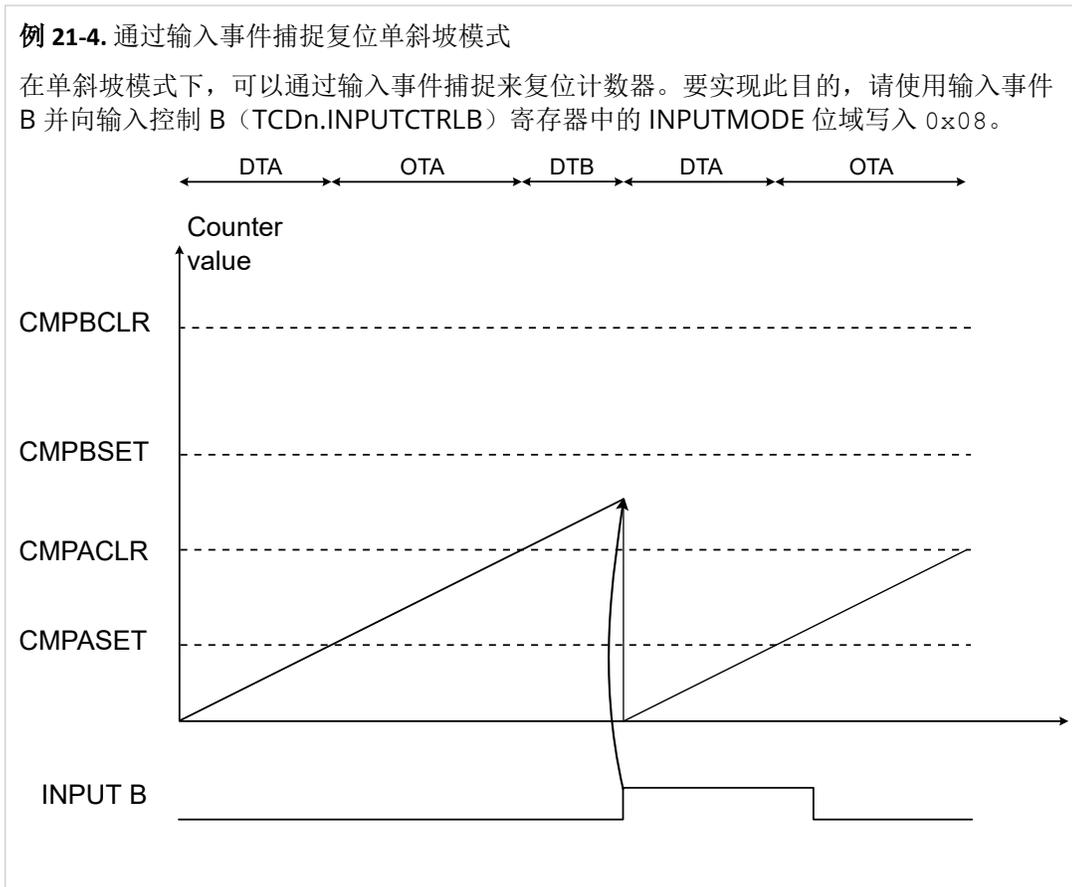
软件可以通过向控制 E（TCDn.CTRLE）寄存器中相应的软件捕捉 A/B 选通（SCAPTUREx）位写入 1 来捕捉 TCD 值。执行此命令且状态（TCDn.STATUS）寄存器中的命令就绪（CMDRDY）位再次读为 1 时，就可以获得 CAPTUREA/B 值。现在可通过先读取 TCDn.CAPTUREAL 或 TCDn.CAPTUREBL 寄存器再读取 TCDn.CAPTUREAH 或 TCDn.CAPTUREBH 寄存器来读取该值。

将捕捉与输入模式结合使用

捕捉功能可与输入模式结合使用。然后，同一事件将捕捉计数器值并触发计数器流程发生变化，具体取决于所选的输入模式。

例 21-4. 通过输入事件捕捉复位单斜坡模式

在单斜坡模式下，可以通过输入事件捕捉来复位计数器。要实现此目的，请使用输入事件 B 并向输入控制 B（TCDn.INPUTCTRLB）寄存器中的 INPUTMODE 位域写入 0x08。



21.3.3.7. 输出控制

通过写入故障控制（TCDn.FAULTCTRL）寄存器来配置输出。TCDn.FAULTCTRL 寄存器在 POR 复位后才会复位为 0。在任何复位后的复位序列期间，TCDn.FAULTCTRL 寄存器将从 TCD（FUSE.TCDCFG）熔丝获取值。

TCDn.FAULTCTRL 寄存器中的比较 x 使能（CMPxEN）位域用于使能不同的输出。TCDn.FAULTCTRL 寄存器中的 CMPx 位域用于设置触发故障时的输出值。

TCD 本身产生两个不同的输出 WOA 和 WOB。可以通过软件将两个附加输出 WOC 和 WOD 配置为连接到 WOA 或 WOB，具体方法为写入控制 C（TCDn.CTRL C）寄存器中的比较 C/D 输出选择（CMPCSEL 和 CMPDSEL）位。

通过向控制 C（TCDn.CTRL C）寄存器中的比较输出值改写（CMPOVR）位写入 1，用户可以根据 TCD 计数器状态改写输出。随后，用户可以通过写入控制 D（TCDn.CTRL D）寄存器中的比较值（CMPAVAL 和 CMPBVAL）位域，在不同的死区和导通时间之中选择输出值。

当在单斜坡模式下使用时，WOA 将仅使用死区时间 A（DTA）和导通时间 A（OTA）的设置来设置输出。WOB 将仅使用死区时间 B（DTB）和导通时间 B（OTB）的值来设置输出。

当改写功能与故障检测（输入模式）结合使用时，TCDn.FAULTCTRL 寄存器中的 CMPA（和 CMPC/D，如果 WOC/D 等于 WOA）位必须等于 TCDn.CTRL D 寄存器中的 CMPAVAL[0]和[2]。如果两者不相等，则检测到故障后的第一个周期的输出极性将会不正确。这同样适用于 TCDn.FAULTCTRL 中的 CMPB（和 CMPC/D，如果 WOC/D 等于 WOB）位，它们必须等于 TCDn.CTRL D 寄存器中的 CMPBVAL[0]和[2]。

由于 TCD 的异步特性，输入事件会立即影响输出信号，当引脚上没有任何负载时，输出有可能会出现纳秒级尖峰。在输入模式不为 0 且触发输入事件时会出现这种情况。尖峰值的方向始终为 TCDn.FAULTCTRL 寄存器给出的 CMPx 值的方向。

21.3.4. 事件

TCD 可生成下表所述的事件：

表 21-8. TCD 中的事件生成器

生成器名称		说明	事件类型	生成时钟域	事件时长
外设	事件				
TCDn	CMPBCLR	计数器与 CMPBCLR 匹配	脉冲	CLK_TCD	1 个 CLK_TCD_CNT 周期
	CMPASET	计数器与 CMPASET 匹配			
	CMPBSET	计数器与 CMPBSET 匹配			
	PROGEV	可编程事件输出 ⁽¹⁾			1 个 CLK_TCD_SYNC 周期

注：

1. 用户可选择触发信号和所有比较匹配（包括 CMPACL R）。此外，可以将输出事件延迟 0 到 255 个 TCD 延时周期。

基于计数器匹配的三个事件可直接生成事件选通信号，信号的持续时间为 TCD 计数器时钟的一个时钟周期。而可编程输出事件所生成的事件选通信号的持续时间为 TCD 同步器时钟的一个时钟周期。

TCD 可接收下表所述的事件：

表 21-9. TCD 中的事件用户和可用事件操作

用户名		说明	输入检测	异步/同步
外设	输入			
TCDn	输入 A/输入 B	停止输出，跳转到相反的比较周期，然后等待。	电平	两者兼具
		停止输出，执行相反的比较周期，然后等待。		
		停止输出，在故障有效时执行相反的比较周期。		
		停止所有输出，保持频率不变。		
		停止所有输出，在故障有效时执行死区时间。		
		停止所有输出，跳转到下一个比较周期，然后等待。		
		停止所有输出，等待软件操作。		
		在出现相应边沿时停止输出，跳转到下一个比较周期。	边沿	
		在出现相应边沿时停止输出，保持频率不变。		
		在出现相应电平时停止输出，保持频率不变。	电平	

输入 A 和输入 B 是 TCD 事件用户，用于检测输入事件并在检测到事件时执行相应操作。有关输入事件及其配置方法的更多信息，请参见章节 [TCD 输入](#)。有关事件类型和事件系统配置的更多详细信息，请参见“事件系统（EVSYS）”一章。

21.3.4.1. 可编程输出事件

可编程输出事件（PROGEV）使用与输入消隐相同的逻辑进行触发选择和延时。因此，此功能无法独立配置。如果使用输入消隐功能，则不能延迟输出事件，并且用于输入消隐的触发将也可用于输出事件。

PROGEV 在 TCDn.DLYCTRL 和 TCDn.DLYVAL 寄存器中配置。可以将输出事件延迟 0 到 255 个 TCD 延时时钟周期。延时输出事件功能使用 TCD 延时时钟，并在触发信号作为事件发送之前进行计数，直至达到 DLYVAL 值。TCD 延时时钟是 TCD 同步器时钟（CLK_TCD_SYNC）的预分频版本，分频系数由 TCDn.DLYCTRL 寄存器中的 DLYPRESC 位设置。输出事件的延时时间为 TCD 时钟周期 x DLYPRESC 分频系数 x DLYVAL。

21.3.5. 中断

表 21-10. 可用中断向量和源

名称	向量说明	条件
OVF	溢出中断	TCD 完成一个 TCD 周期
TRIG	触发中断	<ul style="list-style-type: none"> • TRIGA: 发生事件输入 A 时 • TRIGB: 发生事件输入 B 时

发生中断条件时，中断标志（TCDn.INTFLAGS）寄存器中的相应中断标志将置 1。

可以通过写入中断控制（TCDn.INTCTRL）寄存器中的相应允许位来允许或禁止中断源。

当相应的中断源被允许且中断标志置 1 时，将生成中断请求。中断请求保持有效，直到中断标志清零。有关如何清零中断标志的详细信息，请参见外设的 INTFLAGS 寄存器。

如果中断向量支持多个中断请求条件，将对各个中断请求执行逻辑或运算，将合并生成的一个组合中断请求发送到中断控制器。用户必须读取外设的 INTFLAGS 寄存器来确定存在哪些中断条件。

21.3.6. 休眠模式操作

TCD 在空闲休眠模式下工作，在进入待机休眠模式和掉电休眠模式时停止。

21.3.7. 调试操作

如果在调试模式下暂停 CPU，外设也将暂停正常的工作。可通过向调试控制（TCDn.DBGCTRL）寄存器中的调试运行（DBGRUN）位写入 1，强制此外设在 CPU 暂停的情况下工作。

向 TCDn.DBGCTRL 中的故障检测（FAULTDET）位写入 1 并且 CPU 在调试模式下暂停时，两个输入事件通道上都会发生事件/故障。这些事件/故障的持续时间与中断相当，可用作调试模式下的安全措施（例如通过强制外部元件关闭）。

如果外设配置为需要由 CPU 通过中断或类似操作进行定期控制，则在调试期间停止 CPU 可能会产生不正确的操作或丢失数据。

21.3.8. 配置更改保护

此外设的一些寄存器具有配置更改保护（CCP）。要写入这些寄存器，必须先将特定密钥写入 CPU.CCP 寄存器，然后在 4 条 CPU 指令内对受保护位进行写访问。

必须遵循相应的 CCP 解锁序列，否则无法更改受保护的寄存器。

以下寄存器具有 CCP：

表 21-11. TCD 中受配置更改保护影响的寄存器

寄存器	密钥
TCDn.FAULTCTRL	IOREG

21.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0		CLKSEL[1:0]		CNTPRES[1:0]		SYNCPRES[1:0]		ENABLE
0x01	CTRLB	7:0							WGMODE[1:0]	
0x02	CTRLC	7:0	CMPDSEL	CMPCSEL			FIFTY		AUPDATE	CMPOVR
0x03	CTRLD	7:0	CMP1VAL[3:0]			CMPA 和 BVAL[3:0]				
0x04	CTRLF	7:0	DISEOC			SCAPTUREB	SCAPTUREA	RESTART	SYNC	SYNCEOC
0x05	...									
0x07	保留									
0x08	EVCTRLA	7:0	CFG[1:0]			EDGE		ACTION		TRIGEI
0x09	EVCTRLB	7:0	CFG[1:0]			EDGE		ACTION		TRIGEI
0x0A	...									
0x0B	保留									
0x0C	INTCTRL	7:0					TRIGB0	TRIGA0		OVF
0x0D	INTFLAGS	7:0					TRIGB0	TRIGA0		OVF
0x0E	STATUS	7:0	PWMACTB0	PWMACTA0					CMDRDY	ENRDY
0x0F	保留									
0x10	INPUTCTRLA	7:0					INPUTMODE[3:0]			
0x11	INPUTCTRLB	7:0					INPUTMODE[3:0]			
0x12	FAULTCTRL	7:0	CMPDEN	CMPCEN	CMPBEN	CMPAEN	CMPD	CMPC	CMPB	CMPA
0x13	保留									
0x14	DLYCTRL	7:0			DLYPRESC[1:0]		DLYTRIG[1:0]		DLYSEL[1:0]	
0x15	DLYVAL	7:0	DLYVAL[7:0]							
0x16	...									
0x17	保留									
0x18	DITCTRL	7:0							DITHERSEL[1:0]	
0x19	DITVAL	7:0					DITHER[3:0]			
0x1A	...									
0x1D	保留									
0x1E	DBGCTRL	7:0						FAULTDET		DBGRUN
0x1F	...									
0x21	保留									
0x22	CAPTUREA	7:0	CAPTUREA[7:0]							
		15:8	CAPTUREA[11:8]							
0x24	CAPTUREB	7:0	CAPTUREB[7:0]							
		15:8	CAPTUREB[11:8]							
0x26	...									
0x27	保留									
0x28	CMPASET	7:0	CMPASET[7:0]							
		15:8	CMPASET[11:8]							
0x2A	CMPACLR	7:0	CMPACLR[7:0]							
		15:8	CMPACLR[11:8]							
0x2C	CMPBSET	7:0	CMPBSET[7:0]							
		15:8	CMPBSET[11:8]							
0x2E	CMPBCLR	7:0	CMPBCLR[7:0]							
		15:8	CMPBCLR[11:8]							

21.5. 寄存器说明

21.5.1. 控制 A

名称: CTRLA
 偏移量: 0x00
 复位: 0x00
 属性: 使能保护

位	7	6	5	4	3	2	1	0
	CLKSEL[1:0]		CNTPRES[1:0]		SYNCPRES[1:0]		ENABLE	
访问		R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位		0	0	0	0	0	0	0

Bit 6:5 – CLKSEL[1:0] 时钟选择

时钟选择位域用于选择 TCD 时钟的时钟源。

值	名称	说明
0x0	20MHZ	内部 16/20 MHz 振荡器 (OSC20M)
0x1	-	保留
0x2	EXTCLK	外部时钟
0x3	SYSCLK	系统时钟

Bit 4:3 – CNTPRES[1:0] 计数器预分频器

计数器预分频器位域用于选择 TCD 计数器时钟的分频系数。

值	名称	说明
0x0	DIV1	分频系数 1
0x1	DIV4	分频系数 4
0x2	DIV32	分频系数 32
0x3	-	保留

Bit 2:1 – SYNCPRES[1:0] 同步预分频器

同步预分频器位域用于选择 TCD 时钟的分频系数。

值	名称	说明
0x0	DIV1	分频系数 1
0x1	DIV2	分频系数 2
0x2	DIV4	分频系数 4
0x3	DIV8	分频系数 8

Bit 0 – ENABLE 使能

写入该位时，将自动同步到 TCD 时钟域。

只要该位未在进行同步操作，就可以更改。请参见状态 (TCDn.STATUS) 寄存器中的使能就绪 (ENRDY) 位。

该位不受使能保护。

值	名称	说明
0	NO	禁止 TCD。
1	YES	使能 TCD 并持续运行。

21.5.2. 控制 B

名称: CTRLB
 偏移量: 0x01
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
							WGMODE[1:0]	
访问							R/W	R/W
复位							0	0

Bit 1:0 - WGMODE[1:0] 波形生成模式
 该位域用于选择波形生成模式。

值	名称	说明
0x0	ONERAMP	单斜坡模式
0x1	TWORAMP	双斜坡模式
0x2	FOURRAMP	四斜坡模式
0x3	DS	双斜率模式

21.5.3. 控制 C

名称: CTRLC
 偏移量: 0x02
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	CMPDSEL	CMPCSEL			FIFTY		AUPDATE	CMPOVR
访问	R/W	R/W			R/W		R/W	R/W
复位	0	0			0		0	0

Bit 7 - CMPDSEL 比较 D 输出选择

该位选择将连接到输出 D 的波形。

值	名称	说明
0	PWMA	波形 A
1	PWMB	波形 B

Bit 6 - CMPCSEL 比较 C 输出选择

该位选择将连接到输出 C 的波形。

值	名称	说明
0	PWMA	波形 A
1	PWMB	波形 B

Bit 3 - FIFTY 半波形

如果两个波形具有相同的特性，则该位可写入 1。随后会将写入 TCDn.CMPBSET/TCDn.CLR 寄存器的值也写入 TCDn.CMPASET/TCDn.CLR 寄存器。

Bit 1 - AUPDATE 自动更新

如果该位写入 1，在对比较 B 清零高字节（TCDn.CMPBCLR）寄存器进行写操作后，将自动请求在 TCD 周期结束时执行同步。

如果使能了半波形（通过将该寄存器中的 FIFTY 位置 1），当 AUPDATE 位置 1 时，对“比较 A 清零高字节”寄存器进行写操作后，也会请求在 TCD 周期结束时执行同步。

Bit 0 - CMPOVR 比较输出值改写

当该位写入 1 时，波形输出 A 和 B 的默认值会被写入控制 D 寄存器的工作状态位域中比较 x 值的值改写。有关更多详细信息，请参见 CTRLD 寄存器说明。

21.5.4. 控制 D

名称: CTRLD
 偏移量: 0x03
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	CMP1VAL[3:0]				CMPA 和 BVAL[3:0]			
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 0:3, 4:7 - CMPVAL 比较 x 值（处于工作状态）

该位域设置 TCD 周期中相应状态的 PWMx 信号的逻辑值。

这些设置仅在控制 C（TCDn.CTRLC）寄存器中的比较输出值改写（CMPOVR）位写入 1 时才有效。

表 21-12. 双斜坡和四斜坡模式

CMPxVAL	DTA	OTA	DTB	OTB
PWMA	CMPAVAL[0]	CMPAVAL[1]	CMPAVAL[2]	CMPAVAL[3]
PWMB	CMPBVAL[0]	CMPBVAL[1]	CMPBVAL[2]	CMPBVAL[3]

当在单斜坡模式下使用时，WOA 将仅使用死区时间 A（DTA）和导通时间 A（OTA）的设置来设置输出。

WOB 将仅使用死区时间 B（DTB）和导通时间 B（OTB）的值来设置输出。

表 21-13. 单斜坡模式

CMPxVAL	DTA	OTA	DTB	OTB
PWMA	CMPAVAL[1]	CMPAVAL[0]	-	-
PWMB	-	-	CMPBVAL[3]	CMPBVAL[2]

21.5.5. 控制 E

名称: CTRL E
 偏移量: 0x04
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	DISEOC			SCAPTUREB	SCAPTUREA	RESTART	SYNC	SYNCEOC
访问	R/W			R/W	R/W	R/W	R/W	R/W
复位	0			0	0	0	0	0

Bit 7 - DISEOC 在 TCD 周期结束时禁止选通信号

该位写入 1 时，TCD 将在 TCD 周期结束时自动禁止。

请注意，TCDn.STATUS 中的 ENRDY 将保持低电平，直到 TCD 被禁止。

只有在 TCDn.CTRLA 中的 ENABLE 值未与 TCD 域进行同步时，写入该位才有效。另请参见 TCDn.STATUS 中的 ENRDY 位。

Bit 4 - SCAPTUREB 软件捕捉 B 选通信号

该位写入 1 时，只要执行与 TCD 域的同步操作，就会触发对捕捉 B（TCDn.CAPTUREBL/H）寄存器的软件捕捉。

只有在未进行命令同步时，写入该位才有效。另请参见 TCDn.STATUS 中的 CMDRDY 位。

Bit 3 - SCAPTUREA 软件捕捉 A 选通信号

该位写入 1 时，只要执行与 TCD 域的同步操作，就会触发对捕捉 A（TCDn.CAPTUREAL/H）寄存器的软件捕捉。

只有在未进行命令同步时，写入该位才有效。另请参见 TCDn.STATUS 中的 CMDRDY 位。

Bit 2 - RESTART 重启选通信号

该位写入 1 时，只要该位同步到 TCD 域，就会对 TCD 计数器执行重启。

只有在未进行命令同步时，写入该位才有效。另请参见 TCDn.STATUS 中的 CMDRDY 位。

Bit 1 - SYNC 同步选通信号

该位写入 1 时，只要该位同步到 TCD 域，双缓冲寄存器就会被装入 TCD 域。

只有在未进行命令同步时，写入该位才有效。另请参见 TCDn.STATUS 中的 CMDRDY 位。

Bit 0 - SYNCEOC 在 TCD 周期结束时同步选通信号

该位写入 1 时，双缓冲寄存器将在下一个 TCD 周期结束时被装入 TCD 域。

只有在未进行命令同步时，写入该位才有效。另请参见 TCDn.STATUS 中的 CMDRDY 位。

21.5.6. 事件控制 A

名称: EVCTRLA
 偏移量: 0x08
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	CFG[1:0]			EDGE		ACTION		TRIGEI
访问	R/W	R/W		R/W		R/W		R/W
复位	0	0		0		0		0

Bit 7:6 - CFG[1:0] 事件配置

当激活输入捕捉噪声消除器（FILTERON）时，事件输入将被滤除。滤波器功能需要触发引脚的四个连续等值采样来改变其输出。因此，在使能噪声消除器（FILTERON）时，输入捕捉会延时四个时钟周期。使能异步事件（ASYNCON）时，事件输入将直接影响输出。

值	名称	说明
0x0	NEITHER	禁止滤波器和异步事件。
0x1	FILTERON	使能输入捕捉噪声消除滤波器。
0x2	ASYNCON	使能异步事件输出限定条件。
其他	-	保留。

Bit 4 - EDGE 边沿选择

该位用于选择事件输入的有效边沿或电平。

值	名称	说明
0	FALL_LOW	事件输入的下降沿或低电平会触发捕捉或故障操作。
1	RISE_HIGH	事件输入的上升沿或高电平会触发捕捉或故障操作。

Bit 2 - ACTION 事件动作

该位用于使能事件输入捕捉。在默认情况下，输入将触发故障，具体取决于输入控制寄存器的输入模式。此外，也可以触发事件输入捕捉。

值	名称	说明
0	FAULT	事件触发故障。
1	CAPTURE	事件触发故障和捕捉。

Bit 0 - TRIGEI 触发事件输入使能

向该位写入 1 可将事件作为输入 A 的触发信号。

21.5.7. 事件控制 B

名称: EVCTRLB
 偏移量: 0x09
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	CFG[1:0]			EDGE		ACTION		TRIGEI
访问	R/W	R/W		R/W		R/W		R/W
复位	0	0		0		0		0

Bit 7:6 - CFG[1:0] 事件配置

当激活输入捕捉噪声消除器（FILTERON）时，事件输入将被滤除。滤波器功能需要触发引脚的四个连续等值采样来改变其输出。因此，在使能噪声消除器（FILTERON）时，输入捕捉会延时四个时钟周期。使能异步事件（ASYNCON）时，事件输入将直接影响输出。

值	名称	说明
0x0	NEITHER	禁止滤波器和异步事件。
0x1	FILTERON	使能输入捕捉噪声消除滤波器。
0x2	ASYNCON	使能异步事件输出限定条件。
其他	-	保留。

Bit 4 - EDGE 边沿选择

该位用于选择事件输入的有效边沿或电平。

值	名称	说明
0	FALL_LOW	事件输入的下降沿或低电平会触发捕捉或故障操作。
1	RISE_HIGH	事件输入的上升沿或高电平会触发捕捉或故障操作。

Bit 2 - ACTION 事件动作

该位用于使能事件输入捕捉。在默认情况下，输入将触发故障，具体取决于输入控制寄存器的输入模式。此外，也可以触发事件输入捕捉。

值	名称	说明
0	FAULT	事件触发故障。
1	CAPTURE	事件触发故障和捕捉。

Bit 0 - TRIGEI 触发事件输入使能

向该位写入 1 可将事件作为输入 B 的触发信号。

21.5.8. 中断控制

名称: INTCTRL
 偏移量: 0x0C
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
					TRIGB0	TRIGA0		OVF
访问					R/W	R/W		R/W
复位					0	0		0

Bit 3 - TRIGB 触发 B 中断允许

向该位写入 1 将允许在接收到触发输入 B 时中断。

Bit 2 - TRIGA 触发 A 中断允许

向该位写入 1 将允许在接收到触发输入 A 时中断。

Bit 0 - OVF 计数器溢出

向该位写入 1 将允许序列重启中断或溢出中断。

21.5.9. 中断标志

名称: INTFLAGS
 偏移量: 0x0D
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
					TRIGB0	TRIGA0		OVF
访问					R/W	R/W		R/W
复位					0	0		0

Bit 3 - TRIGB 触发 B 中断标志

出现触发 B 信号或捕捉 B 条件时，触发 B 中断（TRIGB）标志置 1。向该标志的位存储单元写入 1 可将其清零。

Bit 2 - TRIGA 触发 A 中断标志

出现触发 A 信号或捕捉 A 条件时，触发 A 中断（TRIGA）标志置 1。向该标志的位存储单元写入 1 可将其清零。

Bit 0 - OVF 溢出中断标志

溢出标志（OVF）在 TCD 周期结束时置 1。向该标志的位存储单元写入 1 可将其清零。

21.5.10. 状态

名称: STATUS
 偏移量: 0x0E
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	PWMACTB0	PWMACTA0					CMDRDY	ENRDY
访问	R/W	R/W					R	R
复位	0	0					0	0

Bit 7 - PWMACTB B 上的 PWM 活动

每次 WOB 输出从 0 翻转为 1 或从 1 翻转为 0 时，该位都由硬件置 1。
 在检测到新的 PWM 活动之前，必须通过软件向该位写入 1 来清零该状态位。

Bit 6 - PWMACTA A 上的 PWM 活动

每次 WOA 输出从 0 翻转为 1 或从 1 翻转为 0 时，该位都由硬件置 1。
 在检测到新的 PWM 活动之前，必须通过软件向该位写入 1 来清零该状态位。

Bit 1 - CMDRDY 命令就绪

该状态位用于指示命令已同步到 TCD 域并且系统已准备好接收新命令。
 可通过以下操作清零 CMDRDY 位：

1. TCDn.CTRL0 SYNCEOC 选通信号。
2. TCDn.CTRL0 SYNC 选通信号。
3. TCDn.CTRL0 RESTART 选通信号。
4. TCDn.CTRL0 SCAPTUREA 捕捉 A 选通信号。
5. TCDn.CTRL0 SCAPTUREB 捕捉 B 选通信号。
6. TCDn.CTRL0 AUPDATE 写入 1 并写入 TCDn.CMPBCLR0 寄存器。

Bit 0 - ENRDY 使能就绪

该状态位用于指示 TCDn.CTRLA 中的 ENABLE 值已同步到 TCD 域并准备好再次写入。
 可通过以下操作清零 ENRDY 位：

1. 写入 TCDn.CTRLA 中的 ENABLE 位。
2. TCDn.CTRL0 DISEOC 选通信号。
3. 当 TCDn.DBGCTRL 中的调试运行 (DBGCTRL) 位为 0 时，通过片上调试 (OCD) 会话进入中断。

21.5.11. 输入控制 A

名称: INPUTCTRLA
 偏移量: 0x10
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
					INPUTMODE[3:0]			
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3:0 - INPUTMODE[3:0] 输入模式

值	名称	说明
0x0	NONE	输入无动作
0x1	JMPWAIT	停止输出, 跳转到相反的比较周期, 然后等待
0x2	EXECWAIT	停止输出, 执行相反的比较周期, 然后等待
0x3	EXECFAULT	停止输出, 在故障有效时执行相反的比较周期
0x4	FREQ	停止所有输出, 保持频率不变
0x5	EXECDT	停止所有输出, 在故障有效时执行死区时间
0x6	WAIT	停止所有输出, 跳转到下一个比较周期, 然后等待
0x7	WAITSW	停止所有输出, 等待软件操作
0x8	EDGETRIG	在出现相应边沿时停止输出, 跳转到下一个比较周期
0x9	EDGETRIGFREQ	在出现相应边沿时停止输出, 保持频率不变
0xA	LVLTRIGFREQ	在出现相应电平时停止输出, 保持频率不变

21.5.12. 输入控制 B

名称: INPUTCTRLB
 偏移量: 0x11
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
					INPUTMODE[3:0]			
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3:0 - INPUTMODE[3:0] 输入模式

值	名称	说明
0x0	NONE	输入无动作
0x1	JMPWAIT	停止输出, 跳转到相反的比较周期, 然后等待
0x2	EXECWAIT	停止输出, 执行相反的比较周期, 然后等待
0x3	EXECFAULT	停止输出, 在故障有效时执行相反的比较周期
0x4	FREQ	停止所有输出, 保持频率不变
0x5	EXECDT	停止所有输出, 在故障有效时执行死区时间
0x6	WAIT	停止所有输出, 跳转到下一个比较周期, 然后等待
0x7	WAITSW	停止所有输出, 等待软件操作
0x8	EDGETRIG	在出现相应边沿时停止输出, 跳转到下一个比较周期
0x9	EDGETRIGFREQ	在出现相应边沿时停止输出, 保持频率不变
0xA	LVLTRIGFREQ	在出现相应电平时停止输出, 保持频率不变

21.5.13. 故障控制

名称: FAULTCTRL
 偏移量: 0x12
 复位: 0x00
 属性: 配置更改保护

位	7	6	5	4	3	2	1	0
	CMPDEN	CMPEN	CMPBEN	CMPEEN	CMPD	CMPC	CMPB	CMPEA
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 4, 5, 6, 7 - CMPEN 比较 x 使能

该位域使能比较作为引脚上的输出。该位域在上电复位后复位为 0。发生其他复位时，内容保持不变，从 TCD 配置（FUSE.TCD CFG）熔丝装载的复位序列除外。

Bit 0, 1, 2, 3 - CMP 比较 x 值

该位域设置复位后或在输入事件触发导致输出发生变化的故障时的默认状态。该位域在上电复位后复位为 0。发生其他复位时，内容保持不变，从 TCD 配置（FUSE.TCD CFG）熔丝装载的复位序列除外。

21.5.14. 延时控制

名称: DLYCTRL
 偏移量: 0x14
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
			DLYPRESC[1:0]		DLYTRIG[1:0]		DLYSEL[1:0]	
访问			R/W	R/W	R/W	R/W	R/W	R/W
复位			0	0	0	0	0	0

Bit 5:4 - DLYPRESC[1:0] 延时预分频器

该位域用于控制消隐或输出事件延时的预分频器设置。

值	名称	说明
0x0	DIV1	预分频器分频系数 1
0x1	DIV2	预分频器分频系数 2
0x2	DIV4	预分频器分频系数 4
0x3	DIV8	预分频器分频系数 8

Bit 3:2 - DLYTRIG[1:0] 延时触发

该位域用于控制消隐或输出事件延时的触发信号。

值	名称	说明
0x0	CMPASET	CMPASET 触发延时
0x1	CMPACLR	CMPACLR 触发延时
0x2	CMPBSET	CMPBSET 触发延时
0x3	CMPBCLR	CMPASET 触发延时（周期结束）

Bit 1:0 - DLYSEL[1:0] 延时选择

该位域用于控制触发延时、消隐或输出事件延时时必须使用的功能。

值	名称	说明
0x0	OFF	不使用延时功能
0x1	INBLANK	使能输入消隐
0x2	EVENT	使能事件延时
0x3	-	保留

21.5.15. 延时值

名称: DLYVAL
 偏移量: 0x15
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	DLYVAL[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - DLYVAL[7:0] 延时值

该位域用于配置消隐/输出事件延时或事件输出同步延时（用预分频 TCD 周期数表示）。

21.5.16. 抖动控制

名称: DITCTRL
 偏移量: 0x18
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
							DITHERSEL[1:0]	
访问							R/W	R/W
复位							0	0

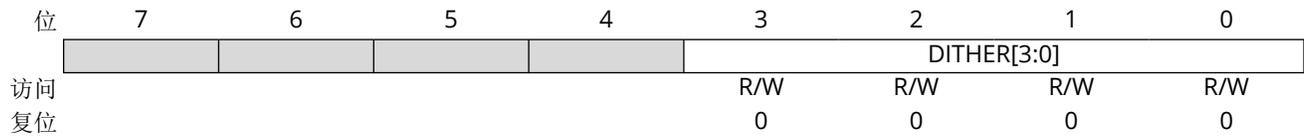
Bit 1:0 - DITHERSEL[1:0] 抖动选择

该位域选择 TCD 周期的哪个状态将从抖动功能中受益。请参见章节[抖动](#)。

值	名称	说明
0x0	ONTIMEB	导通时间斜坡 B
0x1	ONTIMEAB	导通时间斜坡 A 和 B
0x2	DEADTIMEB	死区时间斜坡 B
0x3	DEADTIMEAB	死区时间斜坡 A 和 B

21.5.17. 抖动值

名称: DITVAL
 偏移量: 0x19
 复位: 0x00
 属性: -



Bit 3:0 - DITHER[3:0] 抖动值

该位域根据抖动控制（TCDn.DITCTRL）寄存器中的抖动选择（DITHERSEL）位对导通或关断时间的小数调整进行配置。DITHER 值在每个 TCD 周期结束时被累加到 4 位累加器中。当累加器溢出时，将发生频率调整。

DITHER 位域具有双缓冲，因此会在出现更新条件时复制新值。

21.5.18. 调试控制

名称: DBGCTRL
 偏移量: 0x1E
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
						FAULTDET		DBGRUN
访问						R/W		R/W
复位						0		0

Bit 2 - FAULTDET 故障检测

该位用于定义在调试模式下停止时外设的行为方式。

值	名称	说明
0	NONE	如果 TCD 在调试模式下停止, 则不会产生故障
1	FAULT	如果 TCD 在调试模式下停止, 则会产生故障并且两个触发标志都将置 1

Bit 0 - DBGRUN 调试运行

如果写入 1, 则在 CPU 暂停时, 外设将继续工作在调试模式下。

值	说明
0	外设在中断调试模式下停止并忽略事件
1	当 CPU 停止时, 外设将继续运行在中断调试模式下

21.5.19. 捕捉 A

名称: CAPTUREA
 偏移量: 0x22
 复位: 0x00
 属性: -

TCDn.CAPTUREAL 和 TCDn.CAPTUREAH 寄存器对代表 12 位 TCDn.CAPTUREA 值。

对于捕捉操作，这些寄存器构成了 CPU 的第二个缓冲级和访问点。出现更新条件时，TCDn.CAPTUREA 寄存器将使用缓冲区值进行更新。出现触发 A 信号或发生软件捕捉 A 时，CAPTURE A 寄存器包含 TCD 计数器值。

TCD 计数器值通过软件或事件与 CAPTUREA 同步。

捕捉寄存器会被阻止更新新的捕捉数据，直到读取该寄存器的高字节。

位	15	14	13	12	11	10	9	8
	CAPTUREA[11:8]							
访问					R	R	R	R
复位					0	0	0	0
位	7	6	5	4	3	2	1	0
	CAPTUREA[7:0]							
访问	R	R	R	R	R	R	R	R
复位	0	0	0	0	0	0	0	0

Bit 11:0 - CAPTUREA[11:0] 捕捉 A 值

21.5.20. 捕捉 B

名称: CAPTUREB
 偏移量: 0x24
 复位: 0x00
 属性: -

TCDn.CAPTUREBL 和 TCDn.CAPTUREBH 寄存器对代表 12 位 TCDn.CAPTUREB 值。

对于捕捉操作，这些寄存器构成了 CPU 的第二个缓冲级和访问点。出现更新条件时，TCDn.CAPTUREB 寄存器将使用缓冲区值进行更新。出现触发 B 信号或发生软件捕捉 B 时，CAPTURE B 寄存器包含 TCD 计数器值。

TCD 计数器值通过软件或事件与 CAPTUREB 同步。

捕捉寄存器会被阻止更新新的捕捉数据，直到读取该寄存器的高字节。

位	15	14	13	12	11	10	9	8
	CAPTUREB[11:8]							
访问					R	R	R	R
复位					0	0	0	0
位	7	6	5	4	3	2	1	0
	CAPTUREB[7:0]							
访问	R	R	R	R	R	R	R	R
复位	0	0	0	0	0	0	0	0

Bit 11:0 - CAPTUREB[11:0] 捕捉 B 值

21.5.21. 比较置 1 A

名称: CMPASET
 偏移量: 0x28
 复位: 0x00
 属性: -

TCDn.CMPASET_L 和 TCDn.CMPASET_H 寄存器对代表 12 位 TCDn.CMPASET 值。该寄存器不断与计数器值进行比较。然后比较器的输出将用于生成波形。

位	15	14	13	12	11	10	9	8
	CMPASET[11:8]							
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0
位	7	6	5	4	3	2	1	0
	CMPASET[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 11:0 – CMPASET[11:0] 比较 A 置 1

该位域保存比较寄存器值。

21.5.22. 比较置 1 B

名称: CMPBSET
 偏移量: 0x2C
 复位: 0x00
 属性: -

TCDn.CMPBSETL 和 TCDn.CMPBSETH 寄存器对代表 12 位 TCDn.CMPBSET 值。该寄存器不断与计数器值进行比较。然后比较器的输出将用于生成波形。

位	15	14	13	12	11	10	9	8
					CMPBSET[11:8]			
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0
位	7	6	5	4	3	2	1	0
	CMPBSET[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 11:0 – CMPBSET[11:0] 比较 B 置 1

该位域保存比较寄存器值。

21.5.23. 比较清零 A

名称: CMPACLR
 偏移量: 0x2A
 复位: 0x00
 属性: -

TCDn.CMPACLRH 和 TCDn.CMPACLRH 寄存器对代表 12 位 TCDn.CMPACLR 值。该寄存器不断与计数器值进行比较。然后比较器的输出将用于生成波形。

位	15	14	13	12	11	10	9	8
	CMPACLR[11:8]							
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0
位	7	6	5	4	3	2	1	0
	CMPACLR[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 11:0 – CMPACLR[11:0] 比较 A 清零

该位域保存比较寄存器值。

21.5.24. 比较清零 B

名称: CMPBCLR
 偏移量: 0x2E
 复位: 0x00
 属性: -

TCDn.CMPBCLRL 和 TCDn.CMPBCLRH 寄存器对代表 12 位 TCDn.CMPBCLR 值。该寄存器不断与计数器值进行比较。然后比较器的输出将用于生成波形。

位	15	14	13	12	11	10	9	8
					CMPBCLR[11:8]			
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0
位	7	6	5	4	3	2	1	0
	CMPBCLR[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 11:0 – CMPBCLR[11:0] 比较 B 清零

该位域保存比较寄存器值。

22. RTC——实时计数器

22.1. 特性

- 16 位分辨率
- 可选择的时钟源
- 可编程 15 位时钟预分频
- 一个比较寄存器
- 一个周期寄存器
- 周期溢出时清零定时器
- 可选择在发生溢出和比较匹配时生成中断/事件
- 周期性中断和事件

22.2. 概述

RTC 外设提供两种定时功能：实时计数器（RTC）和周期性中断定时器（Periodic Interrupt Timer, PIT）。

PIT 功能可以独立使能，与 RTC 功能无关。

RTC——实时计数器

RTC 对计数器寄存器中的（预分频）时钟周期进行计数，并将计数器寄存器的内容与周期寄存器和比较寄存器进行比较。

RTC 可以在比较匹配或溢出时产生中断和事件。它将在计数器值等于比较寄存器值后的第一个计数生成比较中断和/或事件，在计数器值等于周期寄存器值后的第一个计数生成溢出中断和/或事件。溢出时会将计数器值复位为零。

RTC 外设通常连续运行（包括在低功耗休眠模式下时）以跟踪时间。它可以将器件从休眠模式唤醒和/或以固定时间间隔中断器件。

参考时钟通常是外部晶振的 32.768 kHz 输出。RTC 也可以由外部时钟信号、32.768 kHz 内部超低功耗振荡器（OSCULP32K）或 32 分频的 OSCULP32K 来提供时钟。

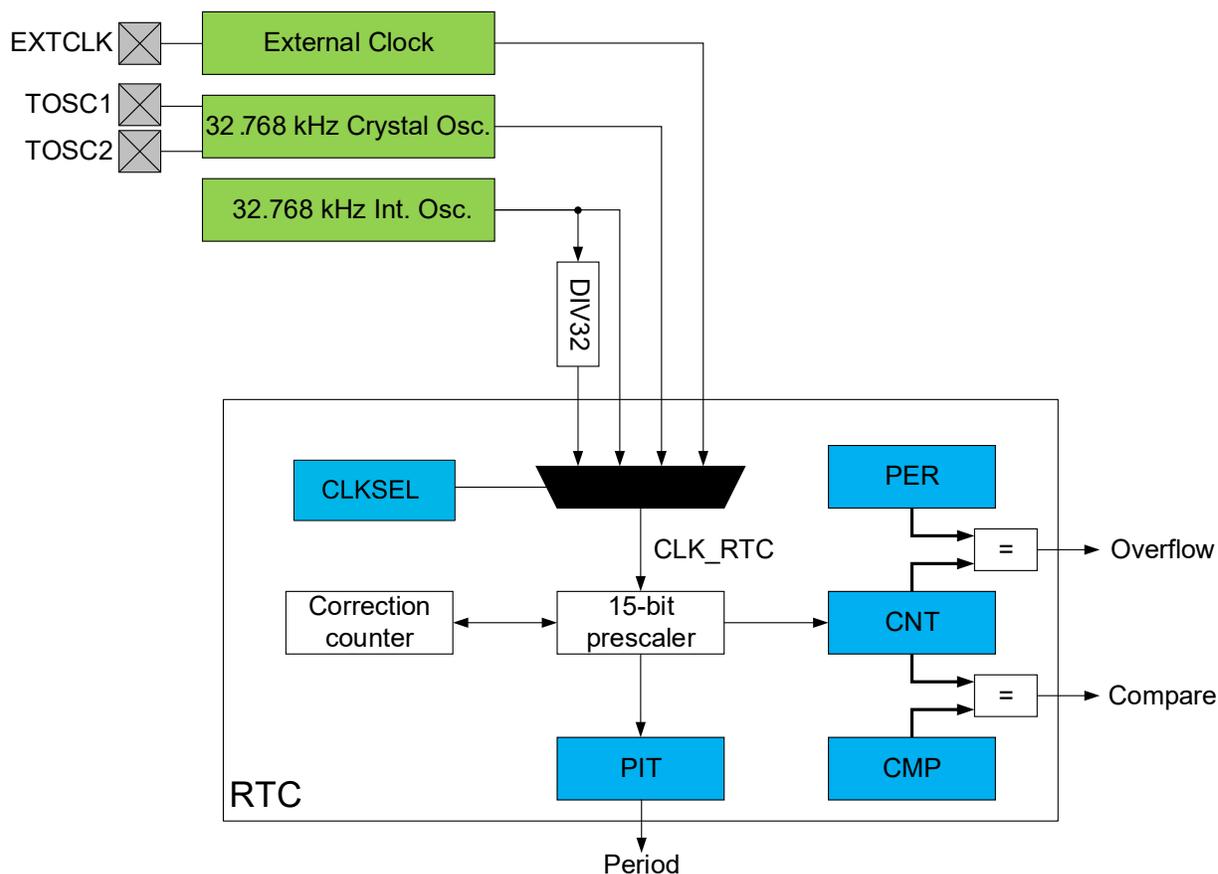
RTC 外设包含一个 15 位可编程预分频器，该预分频器可将参考时钟频率调低之后再送入计数器。RTC 支持多种分辨率和超时周期配置。对于 32.768 kHz 的时钟源，最大分辨率为 30.5 μ s，超时周期最长为 2 秒。分辨率为 1s 时，最大超时周期超过 18 小时（65536 秒）。

PIT——周期性中断定时器

PIT 使用的时钟源（CLK_RTC）与 RTC 功能相同，可以每隔 n 个时钟周期产生一次中断请求或电平事件。对于中断， n 可从{4, 8, 16,...32768}范围内选择；对于事件，则可从{64, 128, 256,...8192}范围内选择。

22.2.1. 框图

图 22-1. RTC 框图



22.3. 时钟

要读取计数器值，外设时钟（CLK_PER）的频率需要至少达到 RTC 时钟（CLK_RTC）的四倍，无论预分频比设置如何。

可以在 TOSC1 或 TOSC2 引脚上连接 32.768 kHz 晶振，以及所需的任何负载电容。也可以在 TOSC1 引脚上连接外部数字时钟。

22.4. RTC 功能说明

RTC 外设提供两种定时功能：实时计数器（RTC）和周期性中断定时器（PIT）。本节介绍 RTC。

22.4.1. 初始化

在使能 RTC 外设和所需操作（中断请求和输出事件）之前，必须先配置 RTC 计数器的源时钟，以使 RTC 可以正常工作。

22.4.1.1. 配置时钟 CLK_RTC

要配置 CLK_RTC，请按照以下步骤操作：

1. 在时钟控制器（CLKCTRL）外设中根据需要配置所需的振荡器。
2. 相应地写入时钟选择（RTC.CLKSEL）寄存器中的时钟选择（CLKSEL）位域。

RTC 和 PIT 功能均可使用 CLK_RTC 时钟配置。

22.4.1.2. 配置 RTC

要使 RTC 正常工作，请执行以下步骤：

1. 在比较（RTC.CMP）寄存器中设置比较值，并且/或者在周期（RTC.PER）寄存器中设置溢出值。
2. 通过写入中断控制（RTC.INTCTRL）寄存器中相应的中断允许位（CMP、OVF）来允许所需的中断。
3. 向控制 A（RTC.CTRLA）寄存器中的预分频器（PRESCALER）位域写入所需值来配置 RTC 内部预分频器。
4. 向 RTC.CTRLA 寄存器中的 RTC 外设使能（RTCCEN）位写入 1 来使能 RTC。

注：器件启动期间，RTC 外设在使用。始终检查状态（RTC.STATUS）和周期性中断定时器状态（RTC.PITSTATUS）寄存器中的同步繁忙位，在初始配置时也是如此。

22.4.2. 操作——RTC

22.4.2.1. 使能和禁止

通过向控制 A（RTC.CTRLA）寄存器中的 RTC 外设使能（RTCCEN）位写入 1 来使能 RTC。通过向 RTC.CTRLA 中的 RTC 外设使能（RTCCEN）位写入 0 来禁止 RTC。

22.5. PIT 功能说明

RTC 外设提供两种定时功能：实时计数器（RTC）和周期性中断定时器（PIT）。本节介绍 PIT。

22.5.1. 初始化

要使 PIT 正常工作，请按照以下步骤操作：

1. 按配置时钟 CLK_RTC 一节所述，配置 RTC 时钟 CLK_RTC。
2. 向 PIT 中断控制（RTC.PITINTCTRL）寄存器中的周期性中断（PI）位写入 1，以允许中断。
3. 向周期性中断定时器控制 A（RTC.PITCTRLA）寄存器中的周期（PERIOD）位域写入所需值，以选择中断周期。
4. 向 RTC.PITCTRLA 寄存器中的周期性中断定时器使能（PITEN）位写入 1，以使能 PIT。

注：器件启动期间，RTC 外设在使用。始终检查 RTC.STATUS 和 RTC.PITSTATUS 寄存器中的同步繁忙位，在初始配置时也是如此。

22.5.2. 操作——PIT

22.5.2.1. 使能和禁止

通过向周期性中断定时器控制 A（RTC.PITCTRLA）寄存器中的周期性中断定时器使能（PITEN）位写入 1 来使能 PIT。通过向 RTC.PITCTRLA 中的周期性中断定时器使能（PITEN）位写入 0 来禁 PIT。

22.5.2.2. PIT 中断时序

第一个中断的时序

PIT 功能和 RTC 功能在预分频器内的同一个计数器上运行，可按下文所述进行配置：

- 通过写入周期（RTC.PER）寄存器来配置 RTC 中断周期
- 通过写入周期性中断定时器控制 A（RTC.PITCTRLA）寄存器中的周期（PERIOD）位域来配置 PIT 中断周期

当两个功能都关闭时（RTC.CTRLA 中的 RTC 外设使能（RTCCEN）位和 RTC.PITCTRLA 中的周期性中断定时器使能（PITEN）位为 0），预分频器关闭，但当任一功能使能时，预分频器运行（即其内部计数器正在计数）。因此，第一个 PIT 中断和第一个 RTC 计数节拍的时序将是未知的（在使能和完整周期之间的任何时间）。

连续操作

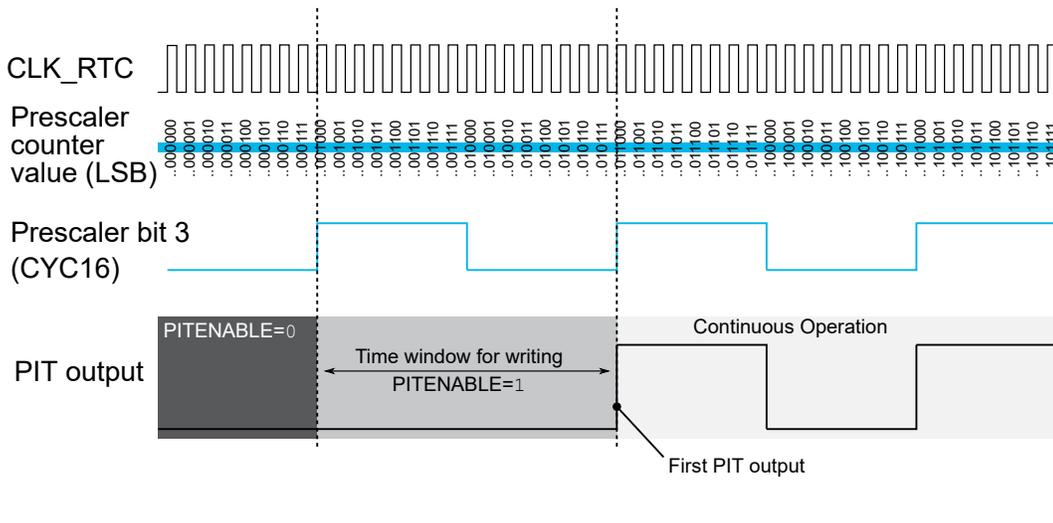
在第一个中断之后，PIT 将继续每 1/2 个 PIT 周期翻转一次，从而产生完整 PIT 周期信号。

例 22-1. PERIOD=CYC16 时的 PIT 时序图

当 RTC.PITCTRLA 中的 PERIOD=CYC16 时，PIT 输出有效地遵循预分频器计数器 bit 3 的状态，因此产生的中断输出的周期为 16 个 CLK_RTC 周期。

向 PITEN 写入 1 和第一个 PIT 中断之间的延时可能在约等于 0 和 16 个 CLK_RTC 周期的完整 PIT 周期之间变化。使能 PIT 及其第一个输出之间的精确延时取决于预分频器的计数阶段：下图所示的第一个中断是通过在前导时间窗口内的任意时间向 PITEN 写入 1 产生的。

图 22-2. PIT 使能和第一个中断间的时序



22.6. 事件

RTC 可生成下表所述的事件：

表 22-1. RTC 事件生成器

生成器名称	说明	事件类型	生成时钟域	事件长度	
RTC 模块	OVF	溢出	脉冲	CLK_RTC	一个 CLK_RTC 周期
	CMP	比较匹配			一个 CLK_RTC 周期
	PIT_DIV8192	预分频的 RTC 时钟进行 8192 分频	电平		由预分频的 RTC 时钟进行 8192 分频后给出
	PIT_DIV4096	预分频的 RTC 时钟进行 4096 分频			由预分频的 RTC 时钟进行 4096 分频后给出
	PIT_DIV2048	预分频的 RTC 时钟进行 2048 分频			由预分频的 RTC 时钟进行 2048 分频后给出
	PIT_DIV1024	预分频的 RTC 时钟进行 1024 分频			由预分频的 RTC 时钟进行 1024 分频后给出
	PIT_DIV512	预分频的 RTC 时钟进行 512 分频			由预分频的 RTC 时钟进行 512 分频后给出
	PIT_DIV256	预分频的 RTC 时钟进行 256 分频			由预分频的 RTC 时钟进行 256 分频后给出
	PIT_DIV128	预分频的 RTC 时钟进行 128 分频			由预分频的 RTC 时钟进行 128 分频后给出
	PIT_DIV64	预分频的 RTC 时钟进行 64 分频			由预分频的 RTC 时钟进行 64 分频后给出

生成 OVF 和 CMP 事件的条件与 RTC.INTFLAGS 寄存器中的相应中断标志置 1 的条件相同。

有关事件用户和事件系统配置的更多详细信息，请参见 [事件系统 \(EVSYS\)](#) 一章。

22.7. 中断

表 22-2. 可用中断向量和中断源

名称	向量说明	条件
RTC	实时计数器溢出和比较匹配中断	<ul style="list-style-type: none"> 溢出 (OVF)：计数器已达到 RTC.PER 寄存器中的值并计满返回到 0。 比较 (CMP)：计数器 (RTC.CNT) 寄存器的值与比较 (RTC.CMP) 寄存器的值匹配。
PIT	周期性中断定时器中断	已经过由 RTC_PITCTRLA 中的 PERIOD 位域配置的时间段。

发生中断条件时，外设的中断标志（*外设*.INTFLAGS）寄存器中的相应中断标志将置 1。

可以通过写入外设的中断控制（*外设*.INTCTRL）寄存器中的相应允许位来允许或禁止中断源。

当相应的中断源被允许且中断标志置 1 时，将生成中断请求。中断请求保持有效，直到中断标志清零。有关如何清零中断标志的详细信息，请参见外设的 INTFLAGS 寄存器。

请注意：

- RTC 具有 2 个 INTFLAGS 寄存器：RTC.INTFLAGS 和 RTC.PITINTFLAGS。
- RTC 具有 2 个 INTCTRL 寄存器：RTC.INTCTRL 和 RTC.PITINTCTRL。

22.8. 休眠模式操作

在空闲休眠模式下，RTC 将继续工作。如果 RTC.CTRLA 中的待机运行 (RUNSTDBY) 位置 1，则 RTC 将在待机休眠模式下运行。

在任何休眠模式下，PIT 都将继续工作。

22.9. 同步

RTC 和 PIT 都是异步的，基于独立于外设时钟 (CLK_PER) 的不同时钟源 (CLK_RTC) 工作。对于控制和计数寄存器更新，需要花费一些 RTC 和/或外设时钟周期，才能在寄存器中提供更新后的寄存器值或者让配置更改分别影响 RTC 或 PIT。有关每个寄存器的这段同步时间，将在 *寄存器说明* 部分进行说明。

对于某些 RTC 寄存器，状态 (RTC.STATUS) 寄存器中提供了同步繁忙标志 (CMPBUSY、PERBUSY、CNTBUSY 和 CTRLABUSY)。

对于 RTC.PITCTRLA 寄存器，周期中断定时器状态 (RTC.PITSTATUS) 寄存器中提供了同步繁忙标志 (CTRLBUSY)。

在写入上述寄存器之前，应检查这些标志。

22.10. 调试操作

如果调试控制 (RTC.DBGCTRL) 寄存器中的调试运行 (DBGRUN) 位为 1，则 RTC 将继续正常工作。如果 DBGRUN 为 0 且 CPU 暂停，则 RTC 也将暂停工作并忽略所有传入事件。

如果周期性中断定时器调试控制 (RTC.PITDBGCTRL) 寄存器中的调试运行 (DBGRUN) 位为 1，则 PIT 将继续正常工作。如果 DBGRUN 在调试模式下为 0 且 CPU 暂停，则 PIT 输出将为低电平。如果此时 PIT 输出为高电平，则在从中断重新启动时会出现一个新的正边沿将中断标志置 1。结果是在正常工作期间不会发生额外的 PIT 中断。如果中断时 PIT 输出为低电平，则 PIT 将恢复为低电平，同时不会产生额外的中断。

22.11. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0	
0x00	CTRLA	7:0	RUNSTDBY	PRESCALER[3:0]							RTCEN
0x01	STATUS	7:0					CMPBUSY	PERBUSY	CNTBUSY	CTRLABUSY	
0x02	INTCTRL	7:0							CMP	OVF	
0x03	INTFLAGS	7:0							CMP	OVF	
0x04	TEMP	7:0	TEMP[7:0]								
0x05	DBGCTRL	7:0								DBGRUN	
0x06	保留										
0x07	CLKSEL	7:0							CLKSEL[1:0]		
0x08	CNT	7:0	CNT[7:0]								
		15:8	CNT[15:8]								
0x0A	PER	7:0	PER[7:0]								
		15:8	PER[15:8]								
0x0C	CMP	7:0	CMP[7:0]								
		15:8	CMP[15:8]								
0x0E	保留										
...											
0x0F	保留										
0x10	PITCTRLA	7:0		PERIOD[3:0]							PITEN
0x11	PITSTATUS	7:0								CTRLBUSY	
0x12	PITINTCTRL	7:0								PI	
0x13	PITINTFLAGS	7:0								PI	
0x14	保留										
0x15	PITDBGCTRL	7:0								DBGRUN	

22.12. 寄存器说明

22.12.1. 控制 A

名称: CTRLA
偏移量: 0x00
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	RUNSTDBY	PRESCALER[3:0]						RTCEN
访问	R/W	R/W	R/W	R/W	R/W			R/W
复位	0	0	0	0	0			0

Bit 7 – RUNSTDBY 在待机模式下运行

值	说明
0	在待机休眠模式下禁止 RTC
1	在待机休眠模式下使能 RTC

Bit 6:3 – PRESCALER[3:0] 预分频比

这些位定义 CLK_RTC 时钟信号的预分频。由于 RTC 时钟与外设时钟之间的同步，从更新寄存器到更新生效之间有一段两个 RTC 时钟周期的延时。应用软件需要检查 RTC.STATUS 寄存器中的 CTRLABUSY 标志是否清零，然后才能写入该寄存器。

值	名称	说明
0x0	DIV1	RTC 时钟/1 (无预分频)
0x1	DIV2	RTC 时钟/2
0x2	DIV4	RTC 时钟/4
0x3	DIV8	RTC 时钟/8
0x4	DIV16	RTC 时钟/16
0x5	DIV32	RTC 时钟/32
0x6	DIV64	RTC 时钟/64
0x7	DIV128	RTC 时钟/128
0x8	DIV256	RTC 时钟/256
0x9	DIV512	RTC 时钟/512
0xA	DIV1024	RTC 时钟/1024
0xB	DIV2048	RTC 时钟/2048
0xC	DIV4096	RTC 时钟/4096
0xD	DIV8192	RTC 时钟/8192
0xE	DIV16384	RTC 时钟/16384
0xF	DIV32768	RTC 时钟/32768

Bit 0 – RTCEN RTC 外设使能

值	说明
0	禁止 RTC 外设
1	使能 RTC 外设

22.12.2. 状态

名称: STATUS
偏移量: 0x01
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
					CMPBUSY	PERBUSY	CNTBUSY	CTRLABUSY
访问					R	R	R	R
复位					0	0	0	0

Bit 3 - CMPBUSY 比较同步繁忙

当 RTC 忙于同步 RTC 时钟域中的比较 (RTC.CMP) 寄存器时, 该位为 1。

Bit 2 - PERBUSY 周期同步繁忙

当 RTC 忙于同步 RTC 时钟域中的周期 (RTC.PER) 寄存器时, 该位为 1。

Bit 1 - CNTBUSY 计数器同步繁忙

当 RTC 忙于同步 RTC 时钟域中的计数 (RTC.CNT) 寄存器时, 该位为 1。

Bit 0 - CTRLABUSY 控制 A 同步繁忙

当 RTC 忙于同步 RTC 时钟域中的控制 A (RTC.CTRLA) 寄存器时, 该位为 1。

22.12.3. 中断控制

名称: INTCTRL
偏移量: 0x02
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
							CMP	OVF
访问							R/W	R/W
复位							0	0

Bit 1 - CMP 比较匹配中断允许

允许比较匹配中断（即，当计数（RTC.CNT）寄存器中的值与比较（RTC.CMP）寄存器中的值匹配时）。

值	说明
0	禁止比较匹配中断
1	允许比较匹配中断

Bit 0 - OVF 溢出中断允许

允许计数器溢出中断（即，当计数（RTC.CNT）寄存器中的值与周期（RTC.PER）寄存器中的值匹配并折回到零时）。

值	说明
0	禁止溢出中断
1	允许溢出中断

22.12.4. 中断标志

名称: INTFLAGS
偏移量: 0x03
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
							CMP	OVF
访问							R/W	R/W
复位							0	0

Bit 1 - CMP 比较匹配中断标志

当计数（RTC.CNT）寄存器中的值与比较（RTC.CMP）寄存器中的值匹配时，该标志置 1。
向该位写入 1 可将标志清零。

Bit 0 - OVF 溢出中断标志

当计数（RTC.CNT）寄存器中的值达到周期（RTC.PER）寄存器中的值并折回到零时，该标志置 1。
向该位写入 1 可将标志清零。

22.12.5. 临时

名称: TEMP
偏移量: 0x4
复位: 0x00
属性: -

临时寄存器供 CPU 用于对该外设的 16 位寄存器进行单周期 16 位访问。该寄存器由该外设的所有 16 位寄存器共用，可通过软件进行读写。有关读写 16 位寄存器的更多详细信息，请参见 [访问 16 位寄存器](#)。

位	7	6	5	4	3	2	1	0
	TEMP[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - TEMP[7:0] 临时

用于对 16 位寄存器执行读/写操作的临时寄存器。

22.12.6. 调试控制

名称: DBGCTRL
偏移量: 0x05
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
								DBGRUN
访问								R/W
复位								0

Bit 0 - DBGRUN 调试运行

值	说明
0	外设在中断调试模式下停止并忽略事件
1	当 CPU 停止时，外设将继续运行在中断调试模式下

22.12.7. 时钟选择

名称: CLKSEL
偏移量: 0x07
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
							CLKSEL[1:0]	
访问							R/W	R/W
复位							0	0

Bit 1:0 - CLKSEL[1:0] 时钟选择

写入这些位可选择 RTC 时钟源 (CLK_RTC)。

值	名称	说明
0x0	INT32K	32.768 kHz (来自 OSCULP32K)
0x1	INT1K	1.024 kHz (来自 OSCULP32K)
0x2	TOSC32K	32.768 kHz (来自 XOSC32K 或 TOSC1 的外部时钟)
0x3	EXTCLK	EXTCLK 引脚的外部时钟

22.12.8. 计数

名称: CNT
偏移量: 0x08
复位: 0x0000
属性: -

RTC.CNTL 和 RTC.CNTH 寄存器对代表 16 位值 RTC.CNT。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

由于 RTC 时钟和主时钟域之间的同步，从更新寄存器到更新生效之间有一段两个 RTC 时钟周期的延时。应用软件需要检查 RTC.STATUS 中的 CNTBUSY 标志是否清零，然后才能写入此寄存器。

位	15	14	13	12	11	10	9	8
	CNT[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	CNT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:8 - CNT[15:8] 计数器高字节

这些位保存 16 位计数器寄存器的 MSB。

Bit 7:0 - CNT[7:0] 计数器低字节

这些位保存 16 位计数器寄存器的 LSB。

22.12.9. 周期

名称: PER
偏移量: 0x0A
复位: 0xFFFF
属性: -

RTC.PERL 和 RTC.PERH 寄存器对代表 16 位值 RTC.PER。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

由于 RTC 时钟和主时钟域之间的同步，从更新寄存器到更新生效之间有一段两个 RTC 时钟周期的延时。应用软件需要检查 RTC.STATUS 中的 PERBUSY 标志是否清零，然后才能写入此寄存器。

位	15	14	13	12	11	10	9	8
	PER[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1
位	7	6	5	4	3	2	1	0
	PER[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1

Bit 15:8 - PER[15:8] 周期高字节

这些位保存 16 位周期寄存器的 MSB。

Bit 7:0 - PER[7:0] 周期低字节

这些位保存 16 位周期寄存器的 LSB。

22.12.10. 比较

名称: CMP
偏移量: 0x0C
复位: 0x0000
属性: -

RTC.CMPL 和 RTC.CMPH 寄存器对代表 16 位值 RTC.CMP。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

位	15	14	13	12	11	10	9	8
	CMP[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	CMP[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:8 – CMP[15:8] 比较高字节

这些位保存 16 位比较寄存器的 MSB。

Bit 7:0 – CMP[7:0] 比较低字节

这些位保存 16 位比较寄存器的 LSB。

22.12.11. 周期性中断定时器控制 A

名称: PITCTRLA
偏移量: 0x10
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	PERIOD[3:0]							PITEN
访问		R/W	R/W	R/W	R/W			R/W
复位		0	0	0	0			0

Bit 6:3 - PERIOD[3:0] 周期

写入该位域可选择各中断之间的 RTC 时钟周期数。

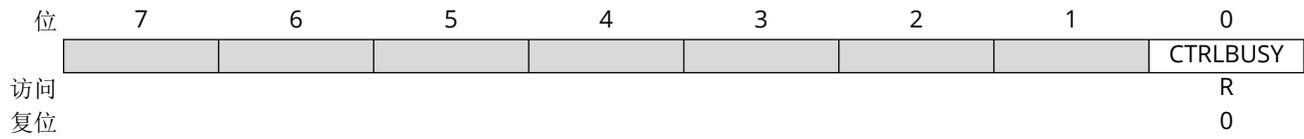
值	名称	说明
0x0	OFF	不产生中断
0x1	CYC4	4 个周期
0x2	CYC8	8 个周期
0x3	CYC16	16 个周期
0x4	CYC32	32 个周期
0x5	CYC64	64 个周期
0x6	CYC128	128 个周期
0x7	CYC256	256 个周期
0x8	CYC512	512 个周期
0x9	CYC1024	1024 个周期
0xA	CYC2048	2048 个周期
0xB	CYC4096	4096 个周期
0xC	CYC8192	8192 个周期
0xD	CYC16384	16384 个周期
0xE	CYC32768	32768 个周期
0xF	-	保留

Bit 0 - PITEN 周期性中断定时器使能

值	说明
0	禁止周期性中断定时器
1	使能周期性中断定时器

22.12.12. 周期性中断定时器状态

名称: PITSTATUS
偏移量: 0x11
复位: 0x00
属性: -

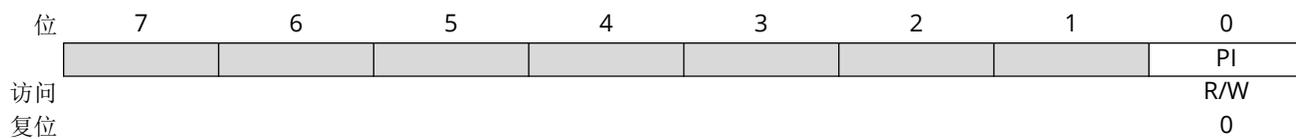


Bit 0 - CTRLBUSY PITCTRLA 同步繁忙

当 RTC 忙于同步 RTC 时钟域中的周期性中断定时器控制 A (RTC.PITCTRLA) 寄存器时, 该位为 1。

22.12.13. PIT 中断控制

名称: PITINTCTRL
偏移量: 0x12
复位: 0x00
属性: -

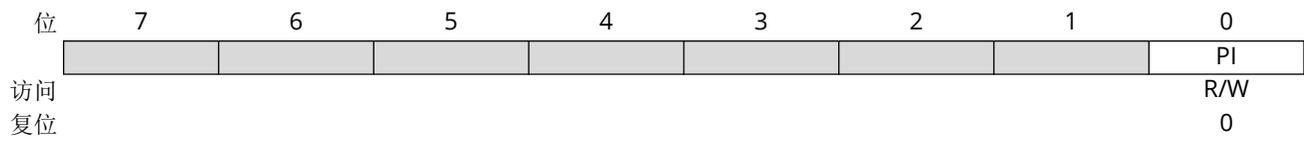


Bit 0 - PI 周期性中断

值	说明
0	禁止周期性中断
1	允许周期性中断

22.12.14. PIT 中断标志

名称: PITINTFLAGS
偏移量: 0x13
复位: 0x00
属性: -



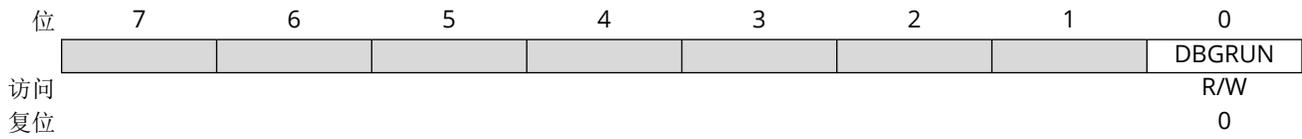
Bit 0 - PI 周期性中断标志

发出周期性中断时，该标志置 1。

写入 1 可将标志清零。

22.12.15. 周期性中断定时器调试控制

名称: PITDBGCTRL
 偏移量: 0x15
 复位: 0x00
 属性: -



Bit 0 - DBGRUN 调试运行

值	说明
0	外设在中断调试模式下停止并忽略事件
1	当 CPU 停止时，外设将继续运行在中断调试模式下

23. USART——通用同步/异步收发器

23.1. 特性

- 全双工操作
- 半双工操作：
 - 单线模式
 - RS-485 模式
- 异步或同步操作
- 支持包含五、六、七、八或九个数据位以及一个或两个停止位的串行帧
- 小数波特率发生器：
 - 可利用任何外设时钟频率生成所需的波特率
 - 无需使用外部振荡器
- 内置错误检测和校正方案：
 - 奇偶校验生成和奇偶校验
 - 缓冲区溢出和帧错误检测
 - 噪声滤波包括虚假启动位检测和数字低通滤波
- 通过单独中断指示以下事件：
 - 发送完成
 - 发送数据寄存器为空
 - 接收完成
- 主 SPI 模式
- 多处理器通信模式
- 帧起始检测
- IRCOM 模块，适用于符合 IrDA[®]标准的脉冲调制/解调
- LIN 从器件支持

23.2. 概述

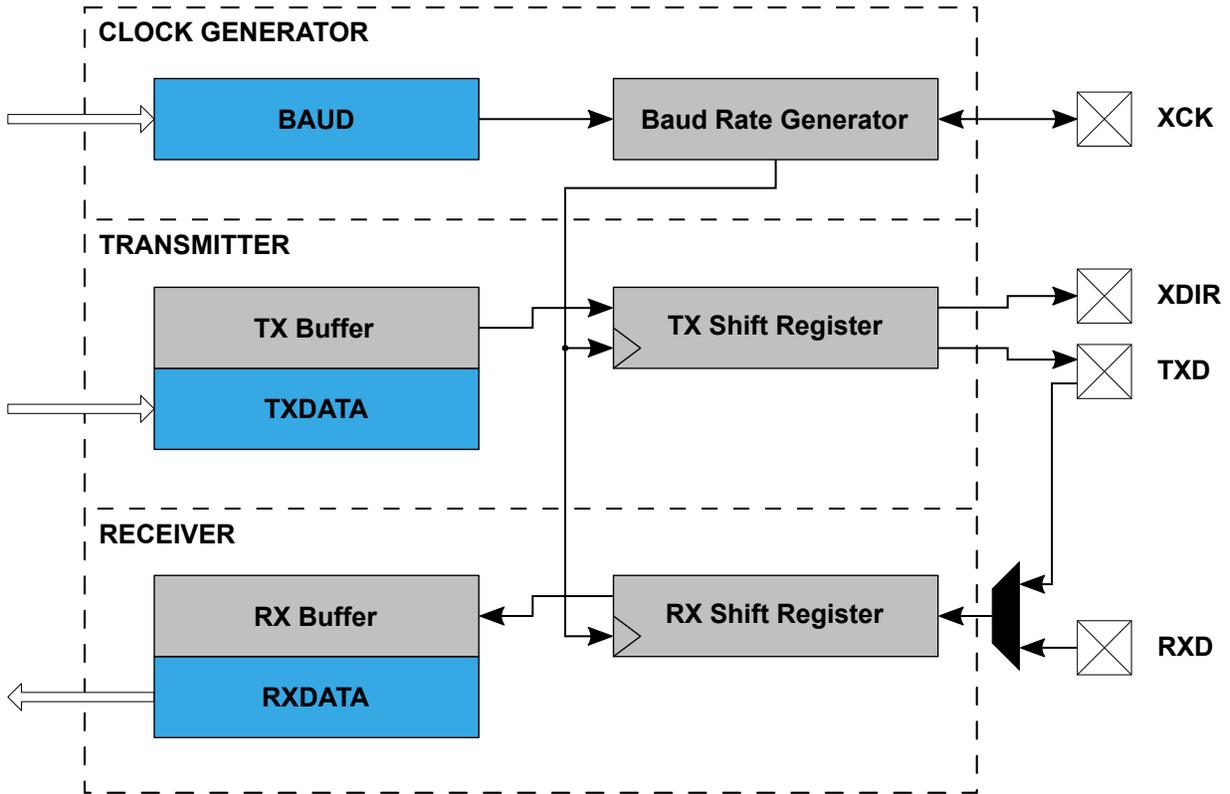
通用同步/异步串行收发器（Universal Synchronous and Asynchronous serial Receiver and Transmitter, USART）是一种快速灵活的串行通信外设。USART 支持多种不同的工作模式，可适应多种类型的应用和通信器件。例如，当需要低引脚数应用时，单线半双工模式非常有用。通信是基于帧的，可以自定义帧格式以支持不同的标准。

USART 在两个方向上进行缓冲，因此能够以零帧间延时连续地发送数据。完成接收和发送时会生成单独的中断，因此允许进行完全中断驱动通信。

发送器由两级写缓冲器、移位寄存器和不同帧格式的控制逻辑组成。接收器由一个两级接收缓冲器和一个移位寄存器组成。所接收数据的状态信息可用于错误校验。数据和时钟恢复单元可确保异步数据接收期间的稳健同步和噪声滤波。

23.2.1. 框图

图 23-1. USART 框图



23.2.2. 信号说明

信号	类型	说明
XCK	输出/输入	同步操作的时钟
XDIR	输出	RS-485 的发送使能
TxD	输出/输入	发送线（和接收线，采用单线模式）
RxD	输入	接收线

23.3. 功能说明

23.3.1. 初始化

全双工模式:

1. 设置波特率 (USARTn.BAUD)。
2. 设置帧格式和工作模式 (USARTn.CTRLA)。
3. 将 TXD 引脚配置为输出。
4. 使能发送器和接收器 (USARTn.CTRLB)。

注:

- 对于中断驱动的 USART 操作，必须在初始化期间禁止全局中断
- 在对已更改的波特率或帧格式重新执行初始化之前，确保寄存器在更改期间没有传输数据

单线半双工模式:

1. 在内部将 TXD 连接到 USART 接收器（USARTn.CTRLA 寄存器中的 LBME 位）。
2. 使能 RX/TX 引脚的内部上拉（PORTx.PINnCTRL 寄存器中的 PULLUPEN 位）
3. 使能漏极开路模式（USARTn.CTRLB 寄存器中的 ODME 位）。
4. 设置波特率（USARTn.BAUD）。
5. 设置帧格式和工作模式（USARTn.CTRLC）。
6. 使能发送器和接收器（USARTn.CTRLB）。

注：

- 使能漏极开路模式时，硬件会将 TXD 引脚自动设置为输出
- 对于中断驱动的 USART 操作，必须在初始化期间禁止全局中断
- 在对已更改的波特率或帧格式重新执行初始化之前，确保寄存器在更改期间没有传输数据

23.3.2. 操作

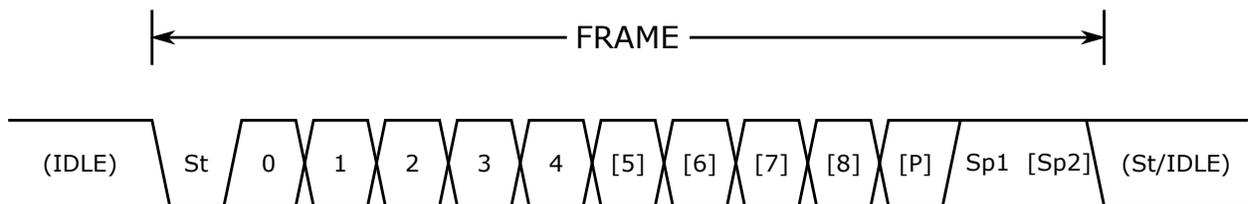
23.3.2.1. 帧格式

USART 数据传输基于帧。帧从启动位开始，后跟一个数据位字符。奇偶校验位在使能后插到数据位之后，第一个停止位之前。帧的停止位后可以直接跟下一个帧，或者通信线路可以返回到空闲（高电平）状态。USART 将以下所有组合视为有效帧格式：

- 1 个启动位
- 5、6、7、8 或 9 个数据位
- 无奇偶校验位、偶校验位或奇校验位
- 1 或 2 个停止位

下图显示了帧格式的可能组合。括号内的位为可选位。

图 23-2. 帧格式



St Start bit, always low

(n) Data bits (0 to 8)

P Parity bit, may be odd or even

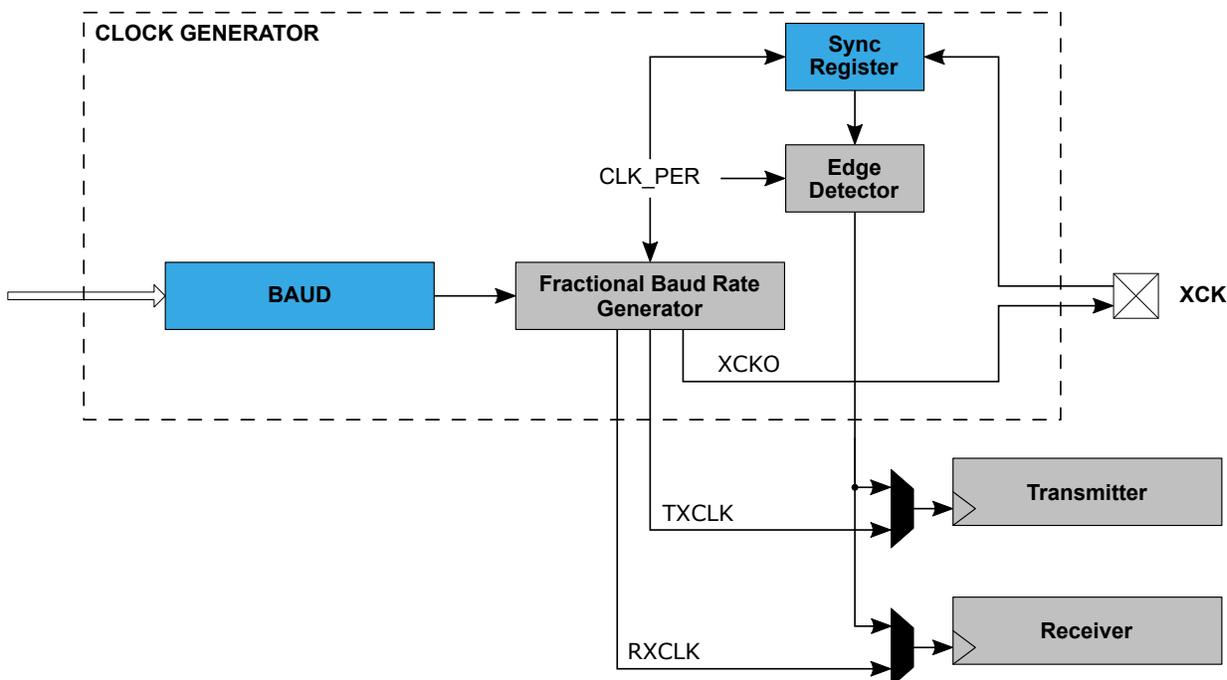
Sp Stop bit, always high

IDLE No transfer on the communication line (RxD or TxD). The Idle state is always high.

23.3.2.2. 时钟生成

用于移位和采样数据位的时钟由小数波特率发生器在内部生成，或者由传输时钟（XCK）引脚在外部生成。

图 23-3. 时钟生成逻辑框图



23.3.2.2.1. 小数波特率发生器

在 USART 不使用 XCK 输入作为时钟源的模式中，小数波特率发生器用于生成时钟。波特率以每秒位数（bps）的形式提供，并通过写入 USARTn.BAUD 寄存器进行配置。波特率（ f_{BAUD} ）是通过将外设时钟（ $f_{\text{CLK_PER}}$ ）除以由 BAUD 寄存器决定的分频系数而生成的。

在小数波特率发生器的硬件中，会出现 $f_{\text{CLK_PER}}$ 不能被 f_{BAUD} 整除的情况。这种情况通常会造成舍入误差。小数波特率生成器要求 BAUD 寄存器包含所需的左移 6 位的分频系数，如表 23-1 中的公式所示。六个最低有效位（Least Significant bit, LSB）将保存所需分频系数的小数部分。使用 BAUD 寄存器的小数部分动态调整 f_{BAUD} ，以实现更接近所需波特率的近似值。

由于波特率不能高于 $f_{\text{CLK_PER}}$ ，BAUD 寄存器的整数部分至少应为 1。由于结果左移 6 位，BAUD 寄存器相应的最小值为 64。因此，值的有效范围为 64-65535。

在同步模式下，只有 BAUD 寄存器的 10 位整数部分（BAUD[15:6]）才能确定波特率，因此必须向小数部分（BAUD[5:0]）写入零。

下表列出了将波特率转换为 BAUD 寄存器的输入值的公式。这些公式考虑了小数部分，因此用这些公式计算的 BAUD 值可以直接写入到 USARTn.BAUD 中，无需任何额外的变换。

表 23-1. 波特率寄存器设置的计算公式

工作模式	条件	波特率（每秒位数）	USART.BAUD 寄存器值计算
异步	$f_{\text{BAUD}} \leq \frac{f_{\text{CLK_PER}}}{S}$ $\text{USART.BAUD} \geq 64$	$f_{\text{BAUD}} = \frac{64 \times f_{\text{CLK_PER}}}{S \times \text{BAUD}}$	$\text{BAUD} = \frac{64 \times f_{\text{CLK_PER}}}{S \times f_{\text{BAUD}}}$
同步主模式	$f_{\text{BAUD}} \leq \frac{f_{\text{CLK_PER}}}{S}$ $\text{USART.BAUD} \geq 64$	$f_{\text{BAUD}} = \frac{f_{\text{CLK_PER}}}{S \times \text{BAUD}[15:6]}$	$\text{BAUD}[15:6] = \frac{f_{\text{CLK_PER}}}{S \times f_{\text{BAUD}}}$

S 为每位的采样数

- 异步正常模式：S = 16

- 异步双速模式：S = 8
- 同步模式：S = 2

23.3.2.3. 数据发送

USART 发送器通过定期将传输线驱动为低电平来发送数据。通过将要发送的数据装入发送数据（USARTn.TXDATAL 和 USARTn.TXDATAH）寄存器来启动数据发送。一旦发送缓冲区为空并准备发送新帧时，发送数据寄存器中的数据便会移入其中，并随后继续传输至移位寄存器。当移位寄存器装入数据后，将发送数据帧。

当移位寄存器中的整个帧移出并且发送数据寄存器或发送缓冲区中没有新数据时，发送完成中断标志（USARTn.STATUS 寄存器中的 TXCIF 位）置 1 并产生中断（如果已允许）。

只能在数据寄存器为空中断标志（USARTn.STATUS 寄存器中的 DREIF 位）置 1（表示该寄存器为空并准备好接收新数据）时写入发送数据寄存器。

当使用少于 8 位的帧时，将忽略写入发送数据寄存器的高位（MSb）。当控制 C（USARTn.CTRL0）寄存器中的字符大小（CHSIZE）位域配置为 9 位（低字节优先）时，必须先写入发送数据寄存器低字节（TXDATAL），然后再写入发送数据寄存器高字节（TXDATAH）。当 CHSIZE 配置为 9 位（高字节优先）时，必须先写入 TXDATAH，然后再写入 TXDATAL。

23.3.2.3.1. 禁止发送器

禁止发送器时，需等到正在进行和待处理的发送完成之后（即，发送移位寄存器、发送数据（USARTn.TXDATAL 和 USARTn.TXDATAH）寄存器和发送缓冲区寄存器不包含要发送的数据时）才会生效。当禁止发送器时，它将不再改写 TXD 引脚，PORT 模块将重新获得对该引脚的控制。无论该引脚之前的设置如何，都会被硬件自动配置为输入。该引脚现在可用作常规 I/O 引脚，USART 不会对其端口进行改写。

23.3.2.4. 数据接收

USART 接收器对接收线进行采样以检测和解释接收到的数据。因此，必须通过向数据方向（PORTx.DIR）寄存器中的相应位写入 0 来将引脚方向配置为输入。

接收器在检测到有效的启动位时接受数据。启动位之后的每一位将以波特率或 XCK 时钟进行采样并移入接收移位寄存器，直到接收到帧的第一个停止位为止。接收器将忽略第二个停止位。当接收到第一个停止位并且接收移位寄存器中存在完整的串行帧时，移位寄存器的内容将移入接收缓冲区。接收完成中断标志（USARTn.STATUS 寄存器中的 RXCIF 位）将置 1 并产生中断（如果已允许）。

RXDATA 寄存器是双缓冲接收缓冲区的一部分，该缓冲区可在 RXCIF 置 1 时供应用软件读取。如果只接收到一个帧，则该帧的数据和状态位会直接推送至 RXDATA 寄存器。如果接收缓冲区中有两个帧，则 RXDATA 寄存器包含较旧帧的数据。

缓冲区在读取 RXDATAL 或 RXDATAH（具体取决于配置）时移出数据。必须先读取不会导致数据移出的寄存器，以便在移出前读取到两个字节。当控制 C（USARTn.CTRL0）寄存器中的字符大小（CHSIZE）位域配置为 9 位（低字节优先）时，读取 RXDATAH 会移出接收缓冲区内容。否则，读取 RXDATAL 会移出接收缓冲区内容。

23.3.2.4.1. 接收器错误标志

USART 接收器具有错误检测机制，可发现传输受损的情况。具体机制包括：

- 帧错误检测——检查接收到的帧是否有效
- 缓冲区溢出检测——指示在接收器缓冲区已满的情况下新数据覆盖旧数据而导致的数据丢失
- 奇偶校验错误检测——通过计算传入帧的奇偶校验并将其与奇偶校验位进行比较来检查传入帧的有效性

每项错误检测机制控制一个可在 RXDATAH 寄存器中读取的错误标志：

- 帧错误（FERR）
- 缓冲区溢出（BUFOVF）

- 奇偶校验错误（PERR）

错误标志及其相应的帧位于 RX 缓冲区中。必须先读取包含错误标志的 RXDATAH 寄存器然后再读取 RXDATAL 寄存器，因为读取 RXDATAL 寄存器将导致 RX 缓冲区移出 RXDATA 字节。

注：如果控制 C（USARTn.CTRLC）寄存器中的字符大小（CHSIZE）位域设置为 9 位、低字节优先（9BITL），则 RXDATAH 寄存器（非 RXDATAL 寄存器）将导致 RX 缓冲区移出 RXDATA 字节。在这种情况下，必须先读取 RXDATAL 寄存器，再读取 RXDATAH 寄存器。

23.3.2.4.2. 禁止接收器

禁止接收器的操作会立即执行。接收器缓冲区将刷新，正在接收的数据将丢失。

23.3.2.4.3. 刷新接收缓冲区

如果必须在正常工作期间刷新接收缓冲区，则重复读取 DATA 位置（USARTn.RXDATAH 和 USARTn.RXDATAL 寄存器），直到接收完成中断标志（USARTn.RXDATAH 寄存器中的 RXCIF 位）清零。

23.3.3. 通信模式

USART 是一种灵活的外设，支持多种不同的通信协议。可用的工作模式可以分为两组：同步和异步通信。

同步通信依靠总线上的一个器件（主器件），通过 XCK 引脚向其余器件提供时钟信号。所有器件均使用此公共时钟信号进行发送和接收，无需额外的同步机制。

器件可配置为在同步总线上作为主器件或从器件运行。

异步通信不使用公共时钟信号，而是依靠采用相同波特率配置的通信器件。接收传输内容时，使用硬件同步机制将传入的传输内容与接收器件外设时钟对齐。

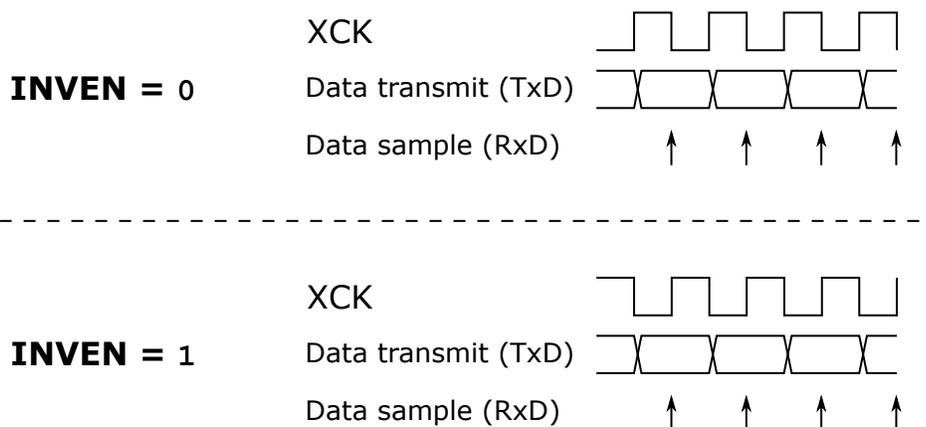
异步通信可以使用四种不同的接收模式。其中一种模式在接收传输内容时速度可以是正常速度的两倍，每位仅采样 8 次，而不是正常情况下的 16 次。其他三种工作模式使用同步逻辑的变体，均以正常速度进行接收。

23.3.3.1. 同步操作

23.3.3.1.1. 时钟操作

XCK 引脚方向控制传输时钟是输入（从模式）还是输出（主模式）。相应的端口引脚方向必须设置为输出（主模式）或输入（从模式）（PORTx.DIRn）。在 XCK 时钟边沿处（与 TXD 上数据发送时的边沿相反）对数据输入（RXD 上）进行采样，如下图所示。

图 23-4. 同步模式 XCK 时序



向端口外设的引脚 n 控制寄存器（PORTx.PINnCTRL）中的反相 I/O 使能（INVEN）位写入 1，可以将 I/O 引脚反相。使用相应 XCK 端口引脚的反相 I/O 设置，可以选择用于在 RxD 上进行采样和在 TxD 上进行发送的 XCK 时钟边沿。如果禁止反相 I/O（INVEN = 0），则 XCK 时钟上升沿表示新数据位的开始，并将在

XCK 时钟下降沿对接收的数据进行采样。如果使能反相 I/O (INVEN = 1)，则 XCK 时钟下降沿表示新数据位的开始，并将在 XCK 时钟上升沿对接收的数据进行采样。

23.3.3.1.2. 外部时钟限制

当在同步从模式下配置 USART 时，必须由主器件从外部提供 XCK 信号。由于时钟是从外部提供的，因此配置 BAUD 寄存器不会影响传输速度。要成功恢复时钟，需要对时钟信号的每个上升沿和下降沿至少采样两次。因此，同步工作模式下的最大 XCK 速度 f_{Client_XCK} 受以下限制：

$$f_{Client_XCK} < \frac{f_{CLK_PER}}{4}$$

如果 XCK 时钟有抖动，或者高/低周期占空比不是 50/50，则必须相应地降低最大 XCK 时钟速度，以确保每个边沿最少采样两次 XCK。

23.3.3.1.3. 主 SPI 模式下的 USART

USART 可以配置为与多个不同的通信接口协同工作，包括串行外设接口 (SPI)，在该接口上可以作为主器件。SPI 是一个四线接口，可允许主器件与一个或多个从器件通信。

帧格式

主 SPI 模式下的 USART 的串行帧始终包含 8 个数据位。向控制 C (USARTn.CTRLC) 寄存器中的数据顺序 (UDORD) 位中写入，可将数据位配置为先使用 LSB 或 MSB 传输。

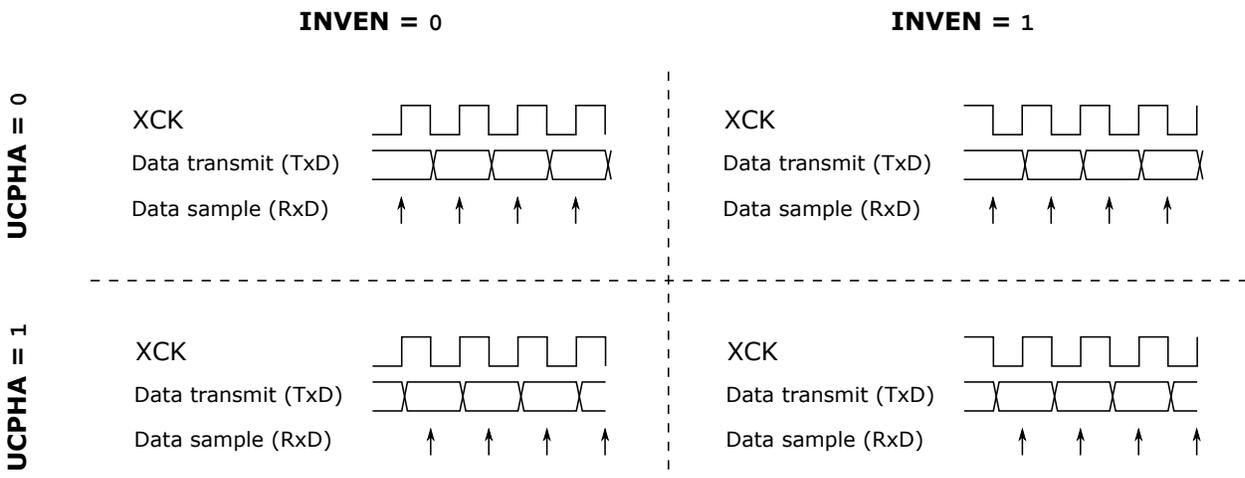
SPI 不使用启动位、停止位或奇偶校验位，因此传输帧只能包含数据位。

时钟生成

作为同步通信接口中的主器件，主 SPI 模式下的 USART 必须生成与从器件共用的接口时钟。接口时钟通过小数波特率发生器生成，如 [小数波特率发生器](#) 中所述。

每个数据位的发送方式为，将数据线拉至高电平或低电平并保持一个完整的时钟周期。接收器将在发送器保持周期的中间位置对位进行采样，如下图所示。图中还展示了如何使用 PORTx.PINnCTRL 寄存器中的反相 I/O 使能 (INVEN) 位和 USARTn.CTRLC 寄存器中的 USART 时钟相位 (UCPHA) 位配置时序方案。

图 23-5. 数据传输时序图



下表对上图进行了补充说明。

表 23-2. INVEN 和 UCPHA 位的功能

INVEN	UCPHA	前沿 ⁽¹⁾	后沿 ⁽¹⁾
0	0	上升沿, 采样	下降沿, 发送
0	1	上升沿, 发送	下降沿, 采样

表 23-2. INVEN 和 UCPHA 位的功能（续）

INVEN	UCPHA	前沿 ⁽¹⁾	后沿 ⁽¹⁾
1	0	下降沿, 采样	上升沿, 发送
1	1	下降沿, 发送	上升沿, 采样

注:

1. 前沿是时钟周期的第一个时钟边沿。后沿是时钟周期的最后一个时钟边沿。

数据发送

主 SPI 模式下的数据发送在功能上与操作部分中所述的一般 USART 操作相同。发送器中断标志以及相应的 USART 中断也是相同的。更多说明, 请参见[数据发送](#)。

数据接收

主 SPI 模式下的数据接收在功能上与操作部分中所述的一般 USART 操作相同。接收器中断标志以及相应的 USART 中断也是相同的, 接收器错误标志除外, 该标志并未使用, 始终读为 0。更多说明, 请参见[数据接收](#)。

主 SPI 模式下的 USART 与 SPI

主 SPI 模式下的 USART 与独立 SPI 外设完全兼容。二者的数据帧和时序配置相同。但主 SPI 模式下的 USART 不支持某些 SPI 特有的功能:

- 写冲突标志保护
- 双速模式
- 多主器件支持

下表对比了主 SPI 模式下的 USART 与 SPI 所用的引脚。

表 23-3. 主 SPI 模式下的 USART 与 SPI 引脚的比较

USART	SPI	备注
TXD	MOSI	主器件输出
RXD	MISO	主器件输入
XCK	SCK	功能相同
-	SS	主 SPI 模式下的 USART 不支持 ⁽¹⁾

注:

1. 对于独立的 SPI 外设, 此引脚与多主器件功能配合使用, 或用作专用的从选择引脚。主 SPI 模式下的 USART 不支持多主器件功能, 并且没有专用的从选择引脚。

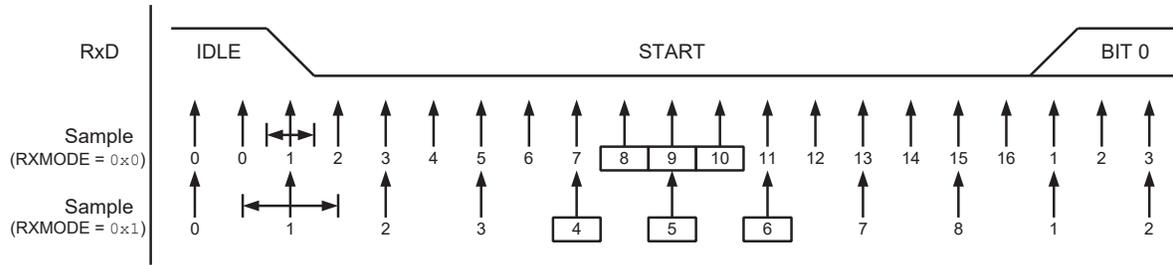
23.3.3.2. 异步操作

23.3.3.2.1. 时钟恢复

由于使用异步模式时没有公共时钟信号, 因此每个通信器件会生成单独的时钟信号。这些时钟信号必须配置为以相同的波特率运行, 以便进行通信。因此, 这些器件以相同的速度运行, 但它们的时序存在相对的偏差。为了支持这种情形, USART 提供了一个硬件时钟恢复单元, 用于将传入的异步串行帧与内部生成的波特率时钟同步。

下图说明了传入帧启动位的采样过程: 图中显示了正常模式和双速模式的时序方案 (USARTn.CTRLB 寄存器中的 RXMODE 位域分别配置为 0x00 和 0x01)。正常模式的采样率是波特率的 16 倍, 双速模式的采样率是波特率的 8 倍 (参见[双速操作](#)了解更多详细信息)。水平箭头给出了最大同步误差。请注意, 双速模式下的最大同步误差更大。

图 23-6. 启动位采样

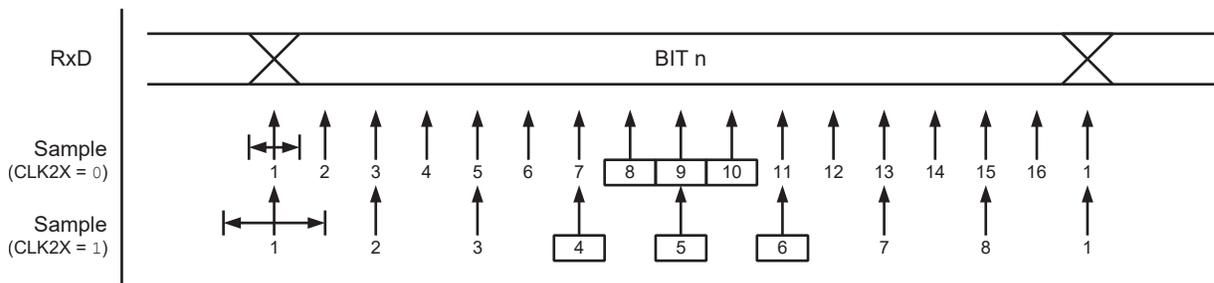


当时钟恢复逻辑检测到从空闲（高电平）状态到启动位（低电平）的下降沿时，启动位检测序列将启动。在上图中，样本 1 表示第一个读为 0 的样本。然后，时钟恢复逻辑使用三个后续样本（正常模式下为样本 8、9 和 10，双速模式下为样本 4、5 和 6）来确定是否接收到有效的启动位。如果两个或三个样本读为 0，则接受启动位。时钟恢复单元同步，数据恢复可以开始。如果不到两个样本读为 0，则拒绝启动位。每个启动位均重复该过程。

23.3.3.2.2. 数据恢复

与时钟恢复一样，数据恢复单元的采样速率是波特率的 8 倍或 16 倍，具体取决于在双速模式还是正常模式下运行。下图显示了在接收帧中读取一个位的采样过程。

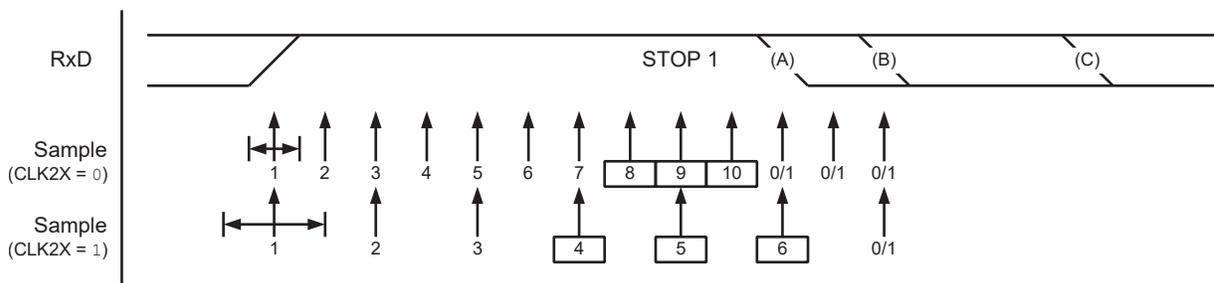
图 23-7. 数据和奇偶校验位的采样



在三个中心样本上使用与时钟恢复相同的多数票决技术来确定接收位的逻辑电平。对每个位重复该过程，直到接收到完整帧为止。

数据恢复单元将只接收第一个停止位，如果有更多停止位则会忽略。如果采样的停止位读为 0，则帧错误标志将置 1。下图显示了停止位的采样，以及下一帧启动位可能的最早开始时间。

图 23-8. 停止位和下一个启动位采样



指示新帧启动位的高电平到低电平的新跳变在用于多数票决的最后一个位之后。对于正常速度模式，第一个低电平样本位于上图中的点 (A) 处。对于双速模式，第一个低电平必须延迟到点 (B)。点 (C) 以标称波特率标记一个全长的停止位。

23.3.3.2.3. 容差

内部生成的波特率和外部接收的数据速率必须相同，但由于时钟源本身固有的误差，二者通常不会完全相同。USART 容许此类误差的存在，这一容差限度有时被称为工作范围。

下表列出了 USART 的工作范围，即可以容许的最大接收器波特率误差。相比双速模式，正常速度模式容许更大范围的波特率变化。

表 23-4. 推荐用于正常速度模式的最大接收器波特率误差

D	R _{slow} [%]	R _{fast} [%]	最大总误差[%]	建议的最大接收器误差[%]
5	93.20	106.67	-6.80/+6.67	±3.0
6	94.12	105.79	-5.88/+5.79	±2.5
7	94.81	105.11	-5.19/+5.11	±2.0
8	95.36	104.58	-4.54/+4.58	±2.0
9	95.81	104.14	-4.19/+4.14	±1.5
10	96.17	103.78	-3.83/+3.78	±1.5

注：

- D: 字符大小和奇偶校验大小的总和 (D = 5 至 10 位)
- R_{SLOW}: 可接受的最慢传入数据速率与接收器波特率之比
- R_{FAST}: 可接受的最快传入数据速率与接收器波特率之比

表 23-5. 推荐用于双速模式的最大接收器波特率误差

D	R _{slow} [%]	R _{fast} [%]	最大总误差[%]	建议的最大接收器误差[%]
5	94.12	105.66	-5.88/+5.66	±2.5
6	94.92	104.92	-5.08/+4.92	±2.0
7	95.52	104.35	-4.48/+4.35	±1.5
8	96.00	103.90	-4.00/+3.90	±1.5
9	96.39	103.53	-3.61/+3.53	±1.5
10	96.70	103.23	-3.30/+3.23	±1.0

注：

- D: 字符大小和奇偶校验大小的总和 (D = 5 至 10 位)
- R_{SLOW}: 可接受的最慢传入数据速率与接收器波特率之比
- R_{FAST}: 可接受的最快传入数据速率与接收器波特率之比

推荐的最大接收器波特率误差基于接收器和发送器等分最大总误差这一假设。

以下公式用于计算传入数据速率与内部接收器波特率的最大比值。

$$R_{SLOW} = \frac{S(D+1)}{S(D+1) + S_F - 1} \quad R_{FAST} = \frac{S(D+2)}{S(D+1) + S_M}$$

- D: 字符大小和奇偶校验大小的总和 (D = 5 至 10 位)
- S: 每位的采样数。对于正常速度模式，S = 16；对于双速模式，S = 8。
- S_F: 用于多数票决的前段采样数。对于正常速度模式，S_F = 8；对于双速模式，S_F = 4。
- S_M: 用于多数票决的中段采样数。对于正常速度模式，S_M = 9；对于双速模式，S_M = 5。
- R_{SLOW}: 可接受的最慢传入数据速率与接收器波特率之比
- R_{FAST}: 可接受的最快传入数据速率与接收器波特率之比

23.3.3.2.4. 双速操作

双速操作允许在异步操作下以更低的外设时钟频率实现更高的波特率。通过向控制 B (USARTn.CTRLB) 寄存器中的 RXMODE 位域写入 0x01, 可以使能该工作模式。

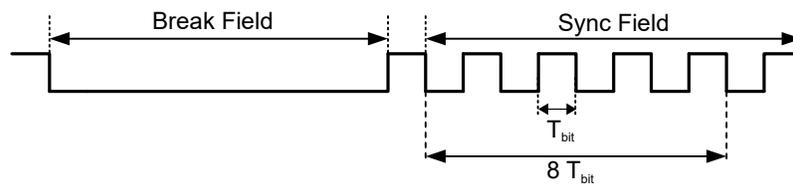
使能后, 给定异步波特率设置的波特率将加倍, 如**小数波特率发生器**中的公式所示。在这种模式下, 接收器将使用一半的采样数 (从 16 减少到 8) 进行数据采样和时钟恢复。这需要更准确的波特率设置和外设时钟。更多详细信息, 请参见[容差](#)。

23.3.3.2.5. 自动波特率

自动波特率功能允许 USART 根据来自通信器件的输入配置其 BAUD 寄存器, 这允许该器件与多个以不同波特率通信的器件进行自主通信。USART 外设支持两种自动波特率模式: 通用自动波特率模式和 LIN 约束自动波特率模式。

两种自动波特率模式都必须接收自动波特率帧, 如下图所示。

图 23-9. 自动波特率时序



当采样 12 个或更多连续的低电平周期时检测到间隔域, 并通知 USART 即将接收同步域。检测到间隔域后, 当检测到同步域的启动位时, 以外设时钟速度运行的计数器启动。随后, 对于同步域的接下来八个 T_{bit} , 计数器将递增。对全部八个位完成采样后, 计数器停止。相应计数器值将作为新的 BAUD 寄存器值。

当 USART 接收模式设置为 GENAUTO 模式时 (USARTn.CTRLB 寄存器中的 RXMODE 位域), 使能通用自动波特率模式。在此模式下, 可将 USARTn.STATUS 寄存器中的等待间隔 (WFB) 位置 1, 以支持检测任意长度的间隔域 (即同样支持少于 12 个周期的情况)。这样, 便可在不了解当前波特率的情况下设置任意新的波特率。如果测量的同步域产生有效的 BAUD 值 (0x0064-0xFFFF), 则 BAUD 寄存器将被更新。

当 USART 接收模式设置为 LINAUTO 模式时 (USARTn.CTRLB 寄存器中的 RXMODE 位域), 它遵循 LIN 格式。通用自动波特率模式的 WFB 功能与 LIN 约束自动波特率模式不兼容, 这意味着接收到的信号必须在 12 个外设时钟周期或更长时间内处于低电平, 间隔域才有效。当检测到间隔域时, USART 期望随后的同步域字符为 0x55。如果接收的同步域字符不是 0x55, 则不一致同步域错误标志 (USARTn.STATUS 寄存器中的 ISFIF 位) 将置 1 并且波特率不变。

23.3.3.2.6. 半双工操作

半双工通信类型允许两个或多个器件相互通信, 但只能同时进行一个通信。USART 可配置为以下列半双工模式工作:

- 单线模式
- RS-485 模式

单线模式

通过将 USARTn.CTRLA 寄存器中的环回模式使能 (LBME) 位置 1, 可以使能单线模式。这样即可在 TXD 引脚与 USART 接收器之间建立内部连接, 使 TXD 引脚成为 TxD/RxD 组合线。RXD 引脚将与 USART 接收器断开连接, 可由其他外设控制。

在单线模式下, 多个器件可同时操控 TxD/RxD 线。如果一个器件将引脚驱动为逻辑高电平 (V_{CC}), 另一个器件将线路拉至低电平 (GND), 则会发生短路。为了应对这种情形, USART 提供漏极开路模式 (USARTn.CTRLB 寄存器中的 ODME 位), 此模式可防止发送器将引脚驱动为逻辑高电平, 从而将其限制为只能拉至低电平。结合此功能与内部上拉功能 (PORTx.PINnCTRL 寄存器中的 PULLUPEN 位), 线路将通过上拉电阻保持高电平, 从而允许任何器件将其拉至低电平。当线路被拉至低电平时, 从 V_{CC} 到 GND 的电流将受到上拉电阻的限制。当使能漏极开路模式时, TXD 引脚由硬件自动设置为输出。

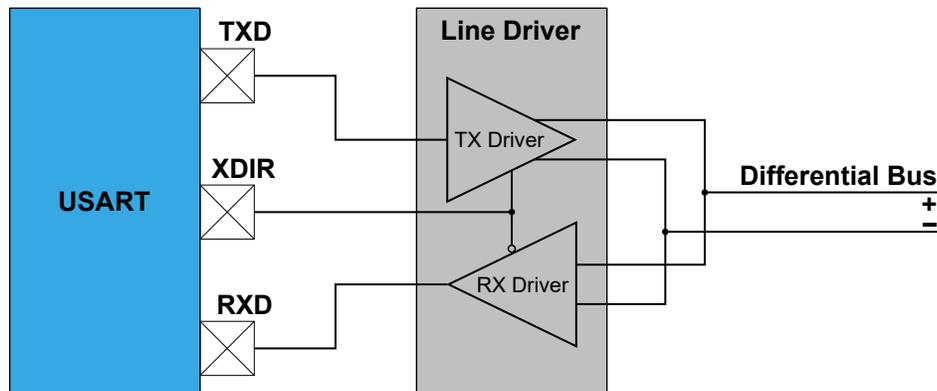
USART 向 TxD/RxD 线发送内容时，也会接收其发送内容。因此，可通过检查接收的数据是否与发送的数据相同来检测是否存在重叠传输。

RS-485 模式

RS-485 是 USART 外设支持的通信标准。这是一种定义通信电路设置的物理接口。数据使用差分信号进行发送，以使通信具备稳健的抗噪性。RS-485 通过写入 USARTn.CTRLA 寄存器中的 RS485 位域来使能。

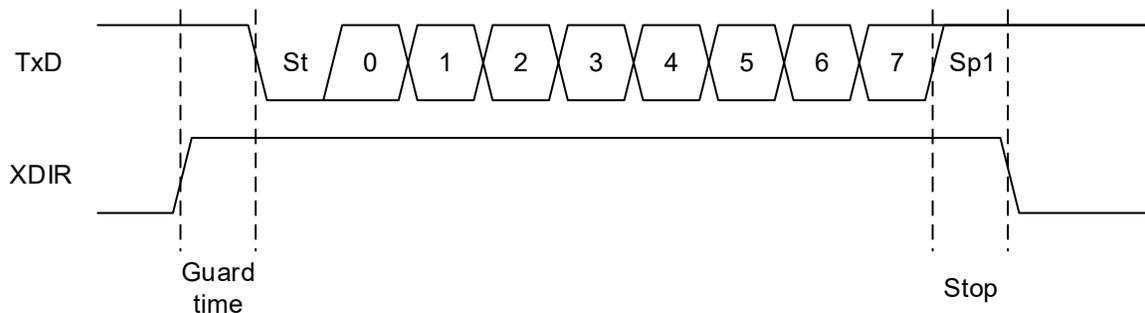
RS-485 模式支持外部线路驱动器件，这些器件将单端 USART 传输转换为相应的差分对信号。向 RS485[0] 中写入 1 可使能 XDIR 引脚的自动控制，该引脚可用于使能线路驱动器件的发送或接收。USART 在发送时自动将 XDIR 引脚驱动为高电平，并在发送完成后将其拉为低电平。相关电路示例如下图所示。

图 23-10. RS-485 总线连接



在数据移出之前，XDIR 引脚在一个波特率时钟周期内变为高电平，以获得一些保护时间来使能外部线路驱动器。对于包括停止位的整个帧，XDIR 引脚将保持高电平。

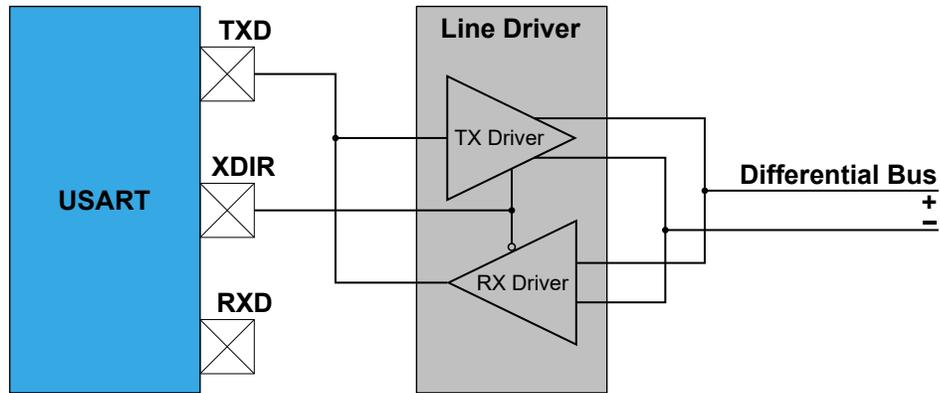
图 23-11. XDIR 驱动时序



向 RS485[1] 中写入 1 可使能 RS-485 模式，以便在开始传输前一个时钟周期自动将 TXD 引脚设置为输出，并在传输完成时再次将该引脚设置为输入。

RS-485 模式与单线模式兼容。单线模式可实现 TXD 引脚与 USART 接收器之间的内部连接，使 TXD 引脚成为 TxD/RxD 组合线路。RXD 引脚将与 USART 接收器断开连接，并可能由其他外设控制。相关电路示例如下图所示。

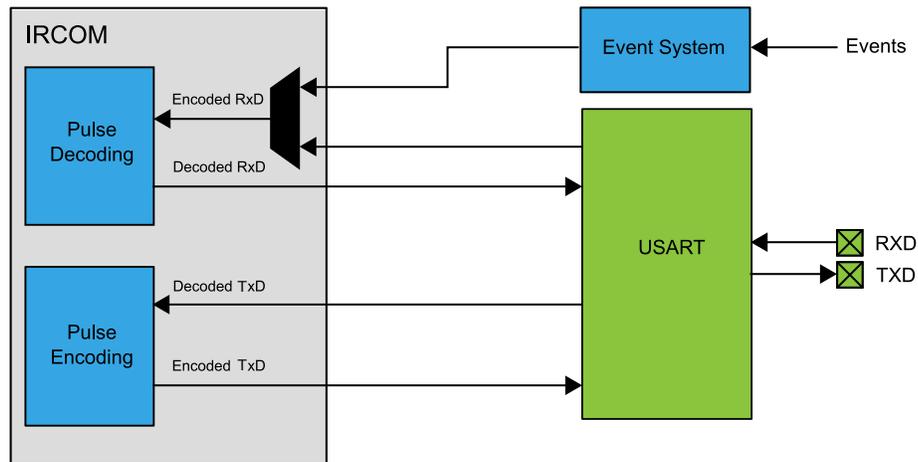
图 23-12. 采用环回模式的 RS-485 连接



23.3.3.2.7. IRCOM 工作模式

USART 外设可配置为红外通信模式（IRCOM），该模式兼容 IrDA® 1.4，波特率最高 115.2 kbps。使能后，IRCOM 模式可为 USART 使能红外脉冲编码/解码。

图 23-13. 框图



可通过向 USARTn.CTRLA 寄存器中的 CMODE 位域写入 0x02 将 USART 设置为 IRCOM 模式。TXD/RXD 引脚上的数据是发送/接收红外脉冲的取反值。此外，也可以从事件系统中选择一个事件通道作为 IRCOM 接收器的输入。这样 IRCOM 就能从除对应的 RXD 引脚以外的 I/O 引脚或源接收输入，同时禁止来自 USART 引脚的 RxD 输入。

对于发送，有三种脉冲调制方案可供使用：

- 3/16 波特率周期
- 固定可编程脉冲时间（基于外设时钟频率）
- 禁止脉冲调制

接收时使用固定可编程最小高电平脉宽，将脉冲解码为逻辑 0。然后将丢弃较短的脉冲，并将该位解码为逻辑 1，就像没有接收到脉冲一样。

当使能 IRCOM 模式时，双速模式不能用于 USART。

23.3.4. 其他特性

23.3.4.1. 奇偶校验

USART 可以使用奇偶校验位来检查数据帧的有效性。奇偶校验位由发送器根据发送内容中值为 1 的位的数量来设置，在接收时则由接收器控制。如果奇偶校验位与发送帧不一致，则接收器会认为数据帧已损坏。

向 USARTn.CTRLC 寄存器中的奇偶校验模式 (PMODE) 位域写入，可以选择偶校验或奇校验以进行错误校验。如果选择偶校验，则当值为 1 的数据位的数量为奇数时，将奇偶校验位置 1 (使值为 1 的位的总数为偶数)。如果选择奇校验，则当值为 1 的数据位的数量为偶数时，将奇偶校验位置 1 (使值为 1 的位的总数为奇数)。

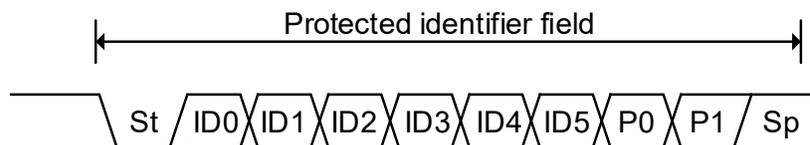
使能时，奇偶校验器计算传入帧中数据位的奇偶校验，并将结果与相应帧的奇偶校验位进行比较。如果检测到奇偶校验错误，则将奇偶校验错误标志 (USARTn.RXDATAH 寄存器中的 PERR 位) 置 1。

如果使能 LIN 约束自动波特率模式 (USARTn.CTRLB 寄存器中的 RXMODE = 0x03)，则仅对受保护的标识符字段执行奇偶校验。如果以下任意一个等式不为真，则将检测到奇偶校验错误，这会将奇偶校验错误标志置 1。

$$P0 = ID0 \text{ XOR } ID1 \text{ XOR } ID2 \text{ XOR } ID4$$

$$P1 = \text{NOT} (ID1 \text{ XOR } ID3 \text{ XOR } ID4 \text{ XOR } ID5)$$

图 23-14. 受保护的标识符字段以及标识符和奇偶校验位的映射



23.3.4.2. 帧起始检测

帧起始检测功能可在接收数据时将 USART 从待机休眠模式唤醒。

在 RXD 引脚上检测到高电平到低电平的跳变时，振荡器将上电，并使能 USART 外设时钟。启动后，只要波特率相对于振荡器起振时间而言足够慢，就可以接收其余的数据帧。振荡器的起振时间因电源电压和温度而异。有关振荡器起振时间特性的详细信息，请参见 *电气特性* 一章。

如果检测到虚假启动位，则器件将在未触发其他唤醒源的情况下返回待机休眠模式。

帧起始检测仅在异步模式下有效。该功能通过写入 USARTn.CTRLB 寄存器中的帧起始检测使能 (SFDEN) 位来使能。如果在器件处于待机休眠模式时检测到启动位，则 USART 接收启动中断标志 (RXSIF) 位置 1。

USART 接收完成中断标志 (RXCIF) 位和 RXSIF 位共用同一中断线路，但二者都有专用的中断设置。下表列出了 USART 起始帧检测模式，具体取决于中断设置。

表 23-6. USART 起始帧检测模式

SFDEN	RXSIF 中断	RXCIF 中断	备注
0	x	x	标准模式
1	禁止	禁止	在帧接收期间只有振荡器通电。如果禁止中断并忽略缓冲区溢出，则所有传入帧都将丢失。
1	禁止	使能	系统/所有时钟被接收完成中断唤醒
1	使能	x	系统/所有时钟在检测到启动位时被唤醒

注：如果正在进行通信，休眠指令将不会关闭振荡器。

23.3.4.3. 多处理器通信

在多个单片机通过同一串行总线进行通信的系统中，多处理器通信模式（Multiprocessor Communication Mode, MPCM）有效减少了接收器必须处理的传入帧数。通过向控制 B（USARTn.CTRLB）寄存器中的 MPCM 位写入 1 来使能此模式。在此模式下，帧中有一个专用位用于指示帧类型是地址还是数据。

如果接收器设置为接收包含 5 到 8 个数据位的帧，则第一个停止位用于指示帧类型。如果接收器设置为接收包含 9 个数据位的帧，则第 9 位用于指示帧类型。当帧类型位为 1 时，帧包含地址。当帧类型位为 0 时，帧是数据帧。如果使用 5 到 8 位字符帧，则必须将发送器设置为使用两个停止位，因为第一个停止位用于指示帧类型。

如果已经寻址了特定的从 MCU，它将像往常一样接收后续数据帧，而其他从 MCU 将忽略这些数据帧，直至接收到另一个地址帧。

23.3.4.3.1. 使用多处理器通信

使用以下过程在多处理器通信模式（MPCM）下交换数据：

1. 所有从 MCU 均处于多处理器通信模式。
2. 主 MCU 发送地址帧，所有从器件接收和读取此帧。
3. 每个从 MCU 确定自身是否被选中。
4. 寻址的 MCU 将禁止 MPCM 并接收所有数据帧。其他从 MCU 将忽略数据帧。
5. 当寻址的 MCU 接收到最后一个数据帧时，它必须再次使能 MPCM 并等待主器件发送新的地址帧。

然后，从步骤 2 开始重复上述过程。

23.3.5. 事件

USART 可生成下表所述的事件。

表 23-7. USART 中的事件生成器

生成器名称		说明	事件类型	生成时钟域	事件时长
外设	事件				
USARTn	XCK	SPI 主模式和同步 USART 主模式下的时钟信号	脉冲	CLK_PER	1 个 XCK 周期

下表说明了事件用户及其相关功能。

表 23-8. USART 中的事件用户

用户名		说明	输入检测	异步/同步
外设	输入			
USARTn	IREI	USARTn IrDA 事件输入	脉冲	同步

23.3.6. 中断

表 23-9. 可用中断向量和中断源

名称	向量说明	条件
RXC	接收完成中断	<ul style="list-style-type: none"> 接收缓冲区中有未读数据（RXCIE） 检测到接收帧起始条件（RXSIE） 自动波特率错误/ISFIF 标志置 1（ABEIE）
DRE	数据寄存器为空中断	发送缓冲区为空/准备接收新数据（DREIE）
TXC	发送完成中断	发送移位寄存器中的整个帧已移出，发送缓冲区中没有新数据（TXCIE）

发生中断条件时，STATUS（USARTn.STATUS）寄存器中的相应中断标志将置 1。

可以通过写入控制 A（USARTn.CTRLA）寄存器中的相应位来允许或禁止中断源。

当相应的中断源被允许且中断标志置 1 时，将产生中断请求。中断请求保持有效，直至中断标志清零。有关如何清零中断标志的详细信息，请参见 USARTn.STATUS 寄存器。

23.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	RXDATAL	7:0	DATA[7:0]							
0x01	RXDATAH	7:0	RXCIF	BUFOVF				FERR	PERR	DATA[8]
0x02	TXDATAL	7:0	DATA[7:0]							
0x03	TXDATAH	7:0								DATA[8]
0x04	STATUS	7:0	RXCIF	TXCIF	DREIF	RXSIF	ISFIF		BDF	WFB
0x05	CTRLA	7:0	RXCIE	TXCIE	DREIE	RXSIE	LBME	ABEIE	RS485[1:0]	
0x06	CTRLB	7:0	RXEN	TXEN		SFDEN	ODME	RXMODE[1:0]		MPCM
0x07	CTRLC	7:0	CMODE[1:0]		PMODE[1:0]		SBMODE	CHSIZE[2:0]		
0x07	CTRLC	7:0	CMODE[1:0]					UDORD	UCPHA	
0x08	BAUD	7:0	BAUD[7:0]							
		15:8	BAUD[15:8]							
0x0A	保留									
0x0B	DBGCTRL	7:0								DBGRUN
0x0C	EVCTRL	7:0								IREI
0x0D	TXPLCTRL	7:0	TXPL[7:0]							
0x0E	RXPLCTRL	7:0	RXPL[6:0]							

23.5. 寄存器说明

23.5.1. 接收器数据寄存器低字节

名称: RXDATAL
 偏移量: 0x00
 复位: 0x00
 属性: -

该寄存器包含 USART 接收器接收数据的低 8 位。USART 接收器是双缓冲的，该寄存器始终代表最早接收帧的数据。如果接收缓冲区中只有一帧的数据，则该寄存器包含该数据。

当读取 RXDATAL 或 RXDATAH 时，缓冲区移出数据，具体取决于配置。必须先读取不会导致数据移位的寄存器，才能在移位前读取两个字节。

当控制 C (USARTn.CTRL) 寄存器中的字符大小 (CHSIZE) 位域配置为 9 位 (低字节优先) 时，RXDATAH 的读操作会对接收缓冲区进行移位，否则，RXDATAL 会对接收缓冲区进行移位。

位	7	6	5	4	3	2	1	0
	DATA[7:0]							
访问	R	R	R	R	R	R	R	R
复位	0	0	0	0	0	0	0	0

Bit 7:0 - DATA[7:0] 接收器数据寄存器

23.5.2. 接收器数据寄存器高字节

名称: RXDATAH
偏移量: 0x01
复位: 0x00
属性: -

该寄存器包含 USART 接收器接收数据的 MSb，以及反映接收数据帧状态的状态位。USART 接收器是双缓冲的，该寄存器始终代表最早接收帧的数据和状态位。如果接收缓冲区中仅存在一帧的数据和状态位，则该寄存器包含该数据。

当读取 RXDATAL 或 RXDATAH 时，缓冲区移出数据，具体取决于配置。必须先读取不会导致数据移位的寄存器，才能在移位前读取两个字节。

当控制 C (USARTn.CTRLC) 寄存器中的字符大小 (CHSIZE) 位配置为 9 位 (低字节优先) 时，RXDATAH 的读操作会对接收缓冲区进行移位，否则，RXDATAL 会对接收缓冲区进行移位。

位	7	6	5	4	3	2	1	0
	RXCIF	BUFOVF				FERR	PERR	DATA[8]
访问	R	R				R	R	R
复位	0	0				0	0	0

Bit 7 - RXCIF USART 接收完成中断标志

当接收缓冲区中有未读数据时，该标志置 1；当接收缓冲区为空时，该标志清零。

Bit 6 - BUFOVF 缓冲区溢出

如果检测到缓冲区溢出，该标志置 1。当接收缓冲区已满时，将发生缓冲区溢出，新帧将在接收移位寄存器中等待，并将检测到新启动位。当读取接收器数据 (USARTn.RXDATAL 和 USARTn.RXDATAH) 寄存器时，该标志清零。

该标志不用于主 SPI 工作模式。

Bit 2 - FERR 帧错误

如果第一个停止位为 0，该标志置 1；如果第一个停止位正确读为 1，该标志清零。

该标志不用于主 SPI 工作模式。

Bit 1 - PERR 奇偶校验错误

如果使能奇偶校验并且接收到的数据有奇偶校验错误，则该标志置 1，否则该标志清零。有关奇偶校验计算的详细信息，请参见 [奇偶校验](#)。

该标志不用于主 SPI 工作模式。

Bit 0 - DATA[8] 接收器数据寄存器

使用 9 位帧大小时，该位保存接收数据的第 9 位 (MSb)。

当控制 B (USARTn.CTRLB) 寄存器中的接收器模式 (RXMODE) 位域配置为 LIN 约束自动波特率 (LINAUTO) 模式时，该位指示接收到的数据是否在 LIN 帧的响应空间内。如果接收到的数据处于受保护的标识符字段中，该位清零，否则置 1。

23.5.3. 发送数据寄存器低字节

名称: TXDATAL
 偏移量: 0x02
 复位: 0x00
 属性: -

写入该寄存器的数据会自动装入发送缓冲区并移至专用移位寄存器。移位寄存器以串行方式将每个位输出到 TXD 引脚。

当使用 9 位帧大小时，必须将第 9 位（MSb）写入发送数据寄存器高字节（USARTn.TXDATAH）。在这种情况下，缓冲区在写入发送数据寄存器低字节（USARTn.TXDATAL）或发送数据寄存器高字节（USARTn.TXDATAH）时移入数据，具体取决于配置。必须先写入不会导致数据移入的寄存器，这样才能在移入前写入两个寄存器。

当控制 C（USARTn.CTRLA）寄存器中的字符大小（CHSIZE）位域配置为 9 位（低字节优先）时，写入发送数据寄存器高字节时会移入发送缓冲区内容。在其他情况下，写入发送数据寄存器低字节时会移入缓冲区内容。

只能在状态（USARTn.STATUS）寄存器中的数据寄存器为空中断标志（DREIF）置 1 时写入该寄存器。

位	7	6	5	4	3	2	1	0
	DATA[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - DATA[7:0] 发送数据寄存器低字节

23.5.4. 发送数据寄存器高字节

名称: TXDATAH
 偏移量: 0x03
 复位: 0x00
 属性: -

写入该寄存器的数据会自动装入发送缓冲区并移至专用移位寄存器。移位寄存器以串行方式将每个位输出到 TXD 引脚。

当使用 9 位帧大小时，必须将第 9 位（MSb）写入发送数据寄存器高字节（USARTn.TXDATAH）。在这种情况下，缓冲区在写入发送数据寄存器低字节（USARTn.TXDATAL）或发送数据寄存器高字节（USARTn.TXDATAH）时移入数据，具体取决于配置。必须先写入不会导致数据移入的寄存器，这样才能在移入前写入两个寄存器。

当控制 C（USARTn.CTRLA）寄存器中的字符大小（CHSIZE）位域配置为 9 位（低字节优先）时，写入发送数据寄存器高字节时会移入发送缓冲区内容。在其他情况下，写入发送数据寄存器低字节时会移入缓冲区内容。

只能在状态（USARTn.STATUS）寄存器中的数据寄存器为空中断标志（DREIF）置 1 时写入该寄存器。

位	7	6	5	4	3	2	1	0
								DATA[8]
访问								R/W
复位								0

Bit 0 - DATA[8] 发送数据寄存器高字节

23.5.5. USART 状态寄存器

名称: STATUS
 偏移量: 0x04
 复位: 0x20
 属性: -

位	7	6	5	4	3	2	1	0
	RXCIF	TXCIF	DREIF	RXSIF	ISFIF		BDF	WFB
访问	R	R/W	R	R/W	R/W		R/W	W
复位	0	0	1	0	0		0	0

Bit 7 - RXCIF USART 接收完成中断标志

当接收缓冲区中有未读数据时，该标志置 1；当接收缓冲区为空时，该标志清零。

Bit 6 - TXCIF USART 发送完成中断标志

发送移位寄存器中的整个帧已移出且发送缓冲区（TXDATAL 和 TXDATAH）寄存器中没有新数据时，该标志置 1。该标志通过写入 1 清零。

Bit 5 - DREIF USART 数据寄存器为空中断标志

当发送数据（USARTn.TXDATAL 和 USARTn.TXDATAH）寄存器为空时，该标志置 1；当这两个寄存器包含尚未移入发送移位寄存器的数据时，该标志清零。

Bit 4 - RXSIF USART 接收启动中断标志

当帧起始检测使能、器件处于待机休眠模式且检测到有效的启动位时，该标志置 1。该标志通过写入 1 清零。
 该标志不适用于主 SPI 工作模式。

Bit 3 - ISFIF 不一致同步域中断标志

当自动波特率模式使能且同步域过短或过长而无法给出有效的波特率设置时，该标志置 1。当 USART 设置为 LINAUTO 模式且 SYNC 字符不为数据值 0x55 时，该标志也将置 1。向该标志写入 1 可将其清零。更多信息，请参见 *自动波特率* 一节。

Bit 1 - BDF 检测到断开标志

当使能自动波特率模式且检测到有效的断开和同步字符时，该标志置 1；当接收到下一个数据时，该标志清零。此外，该标志也可通过写入 1 清零。更多信息，请参见 *自动波特率* 一节。

Bit 0 - WFB 等待中断

将该位置 1 将为下一个传入帧使能等待中断功能。在该帧之后，将自动禁止该功能。

23.5.6. 控制 A

名称: CTRLA
 偏移量: 0x05
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	DREIE	RXSIE	LBME	ABEIE	RS485[1:0]	
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7 – RXCIE 接收完成中断允许

该位控制是否允许接收完成中断。如果允许，则 USARTn.STATUS 寄存器中的 RXCIF 位置 1 时将触发中断。

值	说明
0	禁止接收完成中断
1	允许接收完成中断

Bit 6 – TXCIE 发送完成中断允许

该位控制是否允许发送完成中断。如果允许，则 USARTn.STATUS 寄存器中的 TXCIF 位置 1 时将触发中断。

值	说明
0	禁止发送完成中断
1	允许发送完成中断

Bit 5 – DREIE 数据寄存器为空中断允许

该位控制是否允许数据寄存器为空中断。如果允许，则 USARTn.STATUS 寄存器中的 DREIF 位置 1 时将触发中断。

值	说明
0	禁止数据寄存器为空中断
1	允许数据寄存器为空中断

Bit 4 – RXSIE 接收器起始帧中断允许

该位控制是否允许接收器起始帧中断。如果允许，则 USARTn.STATUS 寄存器中的 RXSIF 位置 1 时将触发中断。

值	说明
0	禁止接收器起始帧中断
1	允许接收器起始帧中断

Bit 3 – LBME 环回模式使能

该位控制是否使能环回模式。使能时，将在 TXD 引脚与 USART 接收器之间建立内部连接，并断开 RXD 引脚到 USART 接收器的输入。

值	说明
0	禁止环回模式
1	使能环回模式

Bit 2 - ABEIE 自动波特率错误中断允许

该位控制是否允许自动波特率错误中断。如果允许，则 USARTn.STATUS 寄存器中的 ISFIF 位置 1 时将触发中断。

值	说明
0	禁止自动波特率错误中断
1	允许自动波特率错误中断

Bit 1:0 - RS485[1:0] RS-485 模式

该位域可使能 RS-485 和选择工作模式。向 RS485[0]中写入 1 可使能 RS-485 模式，以便在开始传输的前一个时钟周期自动将 XDIR 引脚驱动为高电平，并在传输完成时再次将该引脚拉为低电平。向 RS485[1]中写入 1 可使能 RS-485 模式，以便在开始传输的前一个时钟周期自动将 TXD 引脚设置为输出，并在传输完成时再次将该引脚设置为输入。

23.5.7. 控制 B

名称: CTRLB
偏移量: 0x06
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	RXEN	TXEN		SFDEN	ODME	RXMODE[1:0]		MPCM
访问	R/W	R/W		R/W	R/W	R/W	R/W	R/W
复位	0	0		0	0	0	0	0

Bit 7 – RXEN 接收器使能

该位控制是否使能 USART 接收器。更多信息，请参见[禁止接收器](#)。

值	说明
0	禁止 USART 接收器
1	使能 USART 接收器

Bit 6 – TXEN 发送器使能

该位控制是否使能 USART 发送器。更多信息，请参见[禁止发送器](#)。

值	说明
0	禁止 USART 发送器
1	使能 USART 发送器

Bit 4 – SFDEN 帧起始检测使能

该位控制是否使能 USART 帧起始检测模式。更多信息，请参见[帧起始检测](#)。

值	说明
0	禁止 USART 帧起始检测模式
1	使能 USART 帧起始检测模式

Bit 3 – ODME 漏极开路模式使能

该位控制是否使能漏极开路模式。更多信息，请参见[单线模式](#)部分。

值	说明
0	禁止漏极开路模式
1	使能漏极开路模式

Bit 2:1 – RXMODE[1:0] 接收器模式

写入该位域可选择 USART 的接收器模式。

- 向这些位写入 0x00 可使能正常速度（NORMAL）模式。当控制 C（USARTn.CTRLC）寄存器中的 USART 通信模式（CMODE）位域配置为异步 USART（ASYNCHRONOUS）或红外通信（IRCOM）时，始终向 RXMODE 位域写入 0x00。
- 向这些位写入 0x01 可使能双速（CLK2X）模式。更多信息，请参见[双速操作](#)。
- 将这些位写入 0x02 可使能通用自动波特率（GENAUTO）模式。有关更多信息，请参见[自动波特率](#)一节。
- 向这些位写入 0x03 可使能 LIN 约束自动波特率（LINAUTO）模式。有关更多信息，请参见[自动波特率](#)一节。

值	名称	说明
0x00	NORMAL	正常速度模式
0x01	CLK2X	双速模式
0x02	GENAUTO	通用自动波特率模式
0x03	LINAUTO	LIN 约束自动波特率模式

Bit 0 – MPCM 多处理器通信模式

该位控制是否使能多处理器通信模式。更多信息，请参见[多处理器通信](#)。

值	说明
0	禁止多处理器通信模式
1	使能多处理器通信模式

23.5.8. 控制 C——正常模式

名称: CTRLC
 偏移量: 0x07
 复位: 0x03
 属性: -

该寄存器说明适用于除主 SPI 模式以外的所有模式。当该寄存器中的 USART 通信模式 (CMODE) 位域写入“MSPI”时, 请参见 [CTRLC——主 SPI 模式](#) 了解正确的说明。

位	7	6	5	4	3	2	1	0
	CMODE[1:0]		PMODE[1:0]		SBMODE	CHSIZE[2:0]		
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	1	1

Bit 7:6 - CMODE[1:0] USART 通信模式

该位域选择 USART 的通信模式。

向这些位写入 0x03 将改变该寄存器中的可用位域。请参见 [CTRLC——主 SPI 模式](#)。

值	名称	说明
0x00	ASYNCHRONOUS	异步 USART
0x01	SYNCHRONOUS	同步 USART
0x02	IRCOM	红外通信
0x03	MSPI	主 SPI

Bit 5:4 - PMODE[1:0] 奇偶校验模式

该位域可使能奇偶校验生成并选择奇偶校验生成的类型。更多信息, 请参见 [奇偶校验](#)。

值	名称	说明
0x0	DISABLED	禁止
0x1	-	保留
0x2	EVEN	使能, 偶校验
0x3	ODD	使能, 奇校验

Bit 3 - SBMODE 停止位模式

该位可选择发送器要插入的停止位数。

接收器将忽略该设置。

值	说明
0	1 个停止位
1	2 个停止位

Bit 2:0 - CHSIZE[2:0] 字符大小

该位域可选择帧中的数据位数。接收器和发送器使用相同的设置。对于 9BIT 字符大小, 可配置读/写顺序, 即先读/写哪个字节 (RXDATA 或 TXDATA 的低字节或高字节)。

值	名称	说明
0x00	5BIT	5 位
0x01	6BIT	6 位
0x02	7BIT	7 位
0x03	8BIT	8 位
0x04	-	保留
0x05	-	保留
0x06	9BITL	9 位 (低字节优先)
0x07	9BITH	9 位 (高字节优先)

23.5.9. 控制 C——主 SPI 模式

名称: CTRLC
 偏移量: 0x07
 复位: 0x02
 属性: -

该寄存器说明仅适用于 USART 处于主 SPI 模式的情况（将 CMODE 写入 MSPI）。有关其他 CMODE 值的信息，请参见 [CTRLC——正常模式](#)。

有关主 SPI 模式操作的完整说明，请参见[主 SPI 模式下的 USART](#)。

位	7	6	5	4	3	2	1	0
	CMODE[1:0]					UDORD	UCPHA	
访问	R/W	R/W				R/W	R/W	
复位	0	0				0	1	

Bit 7:6 - CMODE[1:0] USART 通信模式

该位域选择 USART 的通信模式。

向这些位写入 0x03 以外的值将改变该寄存器中的可用位域。请参见 [CTRLC——正常模式](#)。

值	名称	说明
0x00	ASYNCHRONOUS	异步 USART
0x01	SYNCHRONOUS	同步 USART
0x02	IRCOM	红外通信
0x03	MSPI	主 SPI

Bit 2 - UDORD USART 数据顺序

该位控制帧格式。接收器和发送器使用相同的设置。更改 UDORD 位的设置将破坏接收器和发送器所有正在进行的通信。

值	说明
0	先发送数据字的 MSb
1	先发送数据字的 LSb

Bit 1 - UCPHA USART 时钟相位

该位控制内部时钟的相位。更多信息，请参见[时钟生成部分](#)。

值	说明
0	在前沿（第一个边沿）进行数据采样
1	在后沿（最后一个边沿）进行数据采样

23.5.10. 波特率寄存器

名称: BAUD
 偏移量: 0x08
 复位: 0x00
 属性: -

USARTn.BAUDL 和 USARTn.BAUDH 寄存器对代表 16 位值 USARTn.BAUD。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

如果波特率发生变化，则发送器和接收器正在进行的传输将被破坏。写入该寄存器将导致波特率预分频器立即更新。有关如何设置波特率的更多信息，请参见表 23-1。

位	15	14	13	12	11	10	9	8
	BAUD[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	BAUD[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:8 - BAUD[15:8] USART 波特率高字节
 该位域保存 16 位波特率寄存器的 MSB。

Bit 7:0 - BAUD[7:0] USART 波特率低字节
 该位域保存 16 位波特率寄存器的 LSB。

23.5.11. 调试控制寄存器

名称: DBGCTRL
 偏移量: 0x0B
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
								DBGRUN
访问								R/W
复位								0

Bit 0 - DBGRUN 调试运行

值	说明
0	外设在中断调试模式下停止并忽略事件
1	当 CPU 停止时，外设将继续运行在中断调试模式下

23.5.12. IrDA 控制寄存器

名称: EVCTRL
 偏移量: 0x0C
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
								IREI
访问								R/W
复位								0

Bit 0 - IREI IrDA 事件输入使能

该位控制是否使能 IrDA 事件输入。更多信息，请参见 [IRCOM 工作模式](#)。

值	说明
0	禁止 IrDA 事件输入
1	使能 IrDA 事件输入

23.5.13. IRCOM 发送器脉冲长度控制寄存器

名称: TXPLCTRL
 偏移量: 0x0D
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	TXPL[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - TXPL[7:0] 发送器脉冲长度

此 8 位值设置发送器的脉冲调制方案。只有在 USART 选择了 IRCOM 模式后，设置此寄存器才起作用，并且必须在使能 USART 发送器（TXEN）之前进行配置。

值	说明
0x00	使用 3/16 的波特率周期脉冲调制
0x01-0xFE	采用固定脉冲长度编码。8 位值设置脉冲的外设时钟周期数。脉冲的开始将与波特率时钟的上升沿同步。
0xFF	禁止脉冲编码。RX 和 TX 信号通过 IRCOM 模块且保持不变。这样便可以通过 IRCOM 模块使能其他功能，例如半双工 USART、环回测试和来自事件通道的 USART RX 输入。

23.5.14. IRCOM 接收器脉冲长度控制寄存器

名称: RXPLCTRL
 偏移量: 0x0E
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	RXPL[6:0]							
访问		R/W						
复位		0	0	0	0	0	0	0

Bit 6:0 - RXPL[6:0] 接收器脉冲长度

此 7 位值用于设置 IRCOM 收发器的滤波系数。只有在 USART 选择了 IRCOM 模式后，设置此寄存器才起作用，并且必须在使能 USART 接收器（RXEN）之前进行配置。

值	说明
0x00	禁止滤波
0x01-0x7F	使能滤波。RXPL+1 的值表示要接受接收的脉冲所需的采样数。

24. SPI——串行外设接口

24.1. 特性

- 全双工三线同步数据传输
- 主模式或从模式操作
- LSb 在前或 MSb 在前的数据传输
- 七种可编程的比特率
- 传输结束中断标志
- 写冲突标志保护
- 从空闲模式唤醒
- 双速 (CK/2) 主 SPI 模式

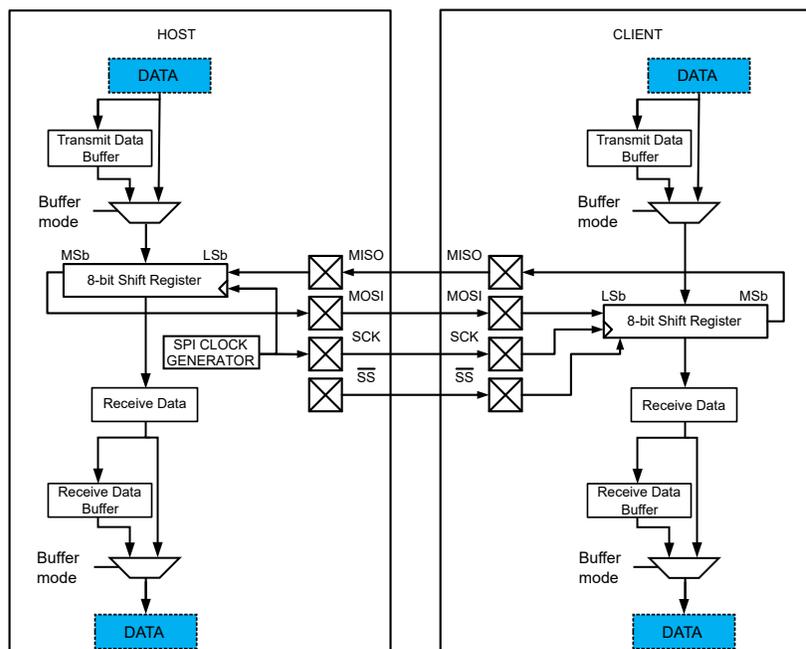
24.2. 概述

串行外设接口 (SPI) 是一个使用三个或四个引脚的高速同步数据传输接口。它支持在一个器件和多个外设器件之间或者在多个单片机之间进行全双工通信。SPI 外设可以配置为主器件或从器件。主器件负责启动并控制所有数据事务。

SPI 主器件和从器件之间的连接如框图所示。系统由两个移位寄存器和一个服务器时钟发生器组成。SPI 主器件通过将所需从器件的从选择 (\overline{SS}) 信号下拉为低电平来启动通信周期。主器件和从器件分别在各自的移位寄存器中准备要发送的数据，待主器件在 SCK 线上生成所需的时钟脉冲后，便可交换数据。在主输出从输入 (Master Output, Slave Input, MOSI) 线上，数据总是从主器件移入从器件；在主输入从输出 (Master Input, Slave Output, MISO) 线上，数据总是从从器件移入主器件。

24.2.1. 框图

图 24-1. SPI 框图



SPI 围绕一个 8 位移位寄存器构建，该寄存器可同时移入和移出数据。发送数据寄存器和接收数据寄存器不是物理寄存器，但在被写入或读取时会映射到其他寄存器：如果对发送数据 (SPIn.DATA) 寄存器进行

写操作，在正常模式下会写入移位寄存器，而在缓冲模式下则会写入发送缓冲寄存器。如果对接收数据（SPI_{IN}.DATA）寄存器进行读操作，在正常模式下会读取接收数据寄存器，而在缓冲模式下则会读取接收数据缓冲区。

在主模式下，SPI 通过时钟发生器来生成 SCK 时钟。在从模式下，对接收的 SCK 时钟进行同步和采样，以触发移位寄存器中的数据移位。

24.2.2. 信号说明

表 24-1. 主模式和从模式下的信号

信号	说明	引脚配置	
		主机模式	从模式
MOSI	主输出从输入	用户定义 ⁽¹⁾	输入
MISO	主输入从输出	输入	用户定义 ^(1,2)
SCK	串行时钟	用户定义 ⁽¹⁾	输入
\overline{SS}	从选择	用户定义 ⁽¹⁾	输入

注：

1. 如果引脚数据方向配置为输出，则引脚电平由 SPI 控制。
2. 如果 SPI 处于从模式且 MISO 引脚数据方向配置为输出，则 \overline{SS} 引脚将通过以下方式控制 MISO 引脚输出：
 - 如果 \overline{SS} 引脚驱动为低电平，则 MISO 引脚由 SPI 控制
 - 如果 \overline{SS} 引脚驱动为高电平，则 MISO 引脚为三态

当使能 SPI 模块时，表 24-1 中标记为“输入”的信号的引脚数据方向将被改写。

24.3. 功能说明

24.3.1. 初始化

按照以下步骤将 SPI 初始化为基本功能状态：

1. 配置端口外设中的 \overline{SS} 引脚。
2. 向控制 A（SPI_{IN}.CTRLA）寄存器中的主/从选择（MASTER）位写入，以选择 SPI 主/从操作。
3. 在主模式下，向 SPI_{IN}.CTRLA 中的预分频比（PRESC）位和时钟加倍（CLK2X）位写入，以选择时钟速度。
4. 可选：向控制 B（SPI_{IN}.CTRLB）寄存器中的 MODE 位写入，以选择数据传输模式。
5. 可选：向 SPI_{IN}.CTRLA 中的数据顺序（DORD）位写入。
6. 可选：向控制 B（SPI_{IN}.CTRLB）寄存器中的 BUFEN 和 BUFWR 位写入，以设置缓冲模式。
7. 可选：要在主模式下禁止多主器件支持，向 SPI_{IN}.CTRLB 中的从选择禁止（SSD）位写入 1。
8. 向 SPI_{IN}.CTRLA 中的 ENABLE 位写入 1，使能 SPI。

24.3.2. 操作

24.3.2.1. 主机模式操作

SPI 配置为主模式时，对 SPI_{IN}.DATA 寄存器进行写操作将启动新的传输。主 SPI 有两种工作模式（正常模式和缓冲模式），如下所述。

24.3.2.1.1. 正常模式

在正常模式下，系统在发送方向上为单缓冲形式，在接收方向上为双缓冲形式。这对于数据处理的影响有以下几种形式：

1. 在整个传输过程完成之前，不能向 DATA (SPIn.DATA) 寄存器写入要发送的新字节。过早写入将导致发送的数据损坏，同时 SPIn.INTFLAGS 中的写冲突 (WRCOL) 标志将置 1。
2. 发送完成后，会立即将接收到的字节写入接收数据缓冲寄存器。
3. 必须在下一次发送完成之前读取接收数据缓冲寄存器，否则数据将丢失。通过读取 SPIn.DATA 来读取该寄存器。
4. 在正常模式下，不使用发送数据缓冲寄存器和接收数据缓冲寄存器。

传输完成后，中断标志 (SPIn.INTFLAGS) 寄存器中的中断标志 (IF) 将置 1。如果允许该中断和全局中断，则将导致执行相应的中断。将中断控制 (SPIn.INTCTRL) 寄存器中的中断允许 (IE) 位置 1 将允许中断。

24.3.2.1.2. 缓冲模式

向 SPIn.CTRLB 寄存器中的 BUFEN 位写入 1 可使能缓冲模式。SPIn.CTRLB 中的 BUFWR 位不影响主模式。在缓冲模式下，系统在发送方向上为双缓冲形式，在接收方向上为三重缓冲形式。这对于数据处理的影响有以下几种形式：

1. 只要中断标志 (SPIn.INTFLAGS) 寄存器中的数据寄存器为空中断标志 (DREIF) 置 1，即可将新字节写入数据 (SPIn.DATA) 寄存器。第一次写入的数据将被立即发送，随后写入的数据将进入发送数据缓冲寄存器。
2. 在发送操作完成后，接收到的字节将立即置于由接收数据寄存器和接收数据缓冲寄存器组成的双条目接收先进先出 (Receive First-In, First-Out, RX FIFO) 队列中。
3. 数据寄存器用于从 RX FIFO 读取数据。必须至少每隔一次传输就读取一次 RX FIFO，以避免任何数据丢失。

如果移位寄存器和发送数据缓冲寄存器均为空，则中断标志 (SPIn.INTFLAGS) 寄存器中的传输完成中断标志 (TXCIF) 将置 1。如果允许该中断和全局中断，则将导致执行相应的中断。将中断控制 (SPIn.INTCTRL) 寄存器中的传输完成中断允许 (TXCIE) 位置 1 时会允许传输完成中断。

24.3.2.1.3. \overline{SS} 引脚功能 (主模式下) ——多主器件支持

在主模式下，控制 B (SPIn.CTRLB) 寄存器中的从选择禁止 (SSD) 位控制 SPI 如何使用 \overline{SS} 引脚。

- 如果 SPIn.CTRLB 中的 SSD 为 0，则 SPI 可使用 \overline{SS} 引脚从主模式切换到从模式。这允许同一 SPI 总线上有多个 SPI 主器件。
- 如果 SPIn.CTRLB 中的 SSD 为 0，则 \overline{SS} 引脚配置为输出引脚，该引脚可用作常规 I/O 引脚，也可供其他外设模块使用，不会影响 SPI 系统
- 如果 SPIn.CTRLB 中的 SSD 为 1，则 SPI 不使用 \overline{SS} 引脚，该引脚可用作常规 I/O 引脚，也可供其他外设模块使用。

如果 SPIn.CTRLB 中的 SSD 位为 0，且 \overline{SS} 配置为输入引脚，则 \overline{SS} 引脚必须保持高电平以确保主 SPI 操作。低电平将被解析为另一个主器件正试图控制总线。这会将 SPI 切换到从模式，SPI 的硬件将执行以下操作：

1. SPI 控制 A (SPIn.CTRLA) 寄存器中的主模式 (MASTER) 位清零，SPI 系统变为从模式。当满足表 24-2 中的条件时，SPI 引脚的方向将发生切换。
2. 中断标志 (SPIn.INTFLAGS) 寄存器中的中断标志 (IF) 位将置 1。如果允许该中断和全局中断，则将执行相应的中断程序。

表 24-2. SPIn.CTRLB 中的 SSD 位为 0 时的 \overline{SS} 引脚功能概览

SS 配置	SS 引脚电平	说明
输入	高电平	激活 (选择) 主模式
	低电平	禁止主模式，切换到从模式
输出	高电平	激活 (选择) 主模式
	低电平	

注：如果器件处于主模式，则无法确保 \overline{SS} 引脚在两次发送之间保持高电平，在写入新字节之前必须检查 SPIn.CTRLA 中的主模式（MASTER）位的状态。在主模式位通过 \overline{SS} 线的低电平清零后，必须通过应用程序将其置 1 才能重新使能 SPI 主模式。

24.3.2.2. 从模式

在从模式下，SPI 外设从主器件接收 SPI 时钟和从选择信号。从模式支持三种工作模式：一种正常模式和两种配置的缓冲模式。在从模式下，控制逻辑将对 SCK 引脚上的传入信号进行采样。为确保对该时钟信号进行正确采样，最短低电平和高电平周期均必须超过两个外设时钟周期。

24.3.2.2.1. 正常模式

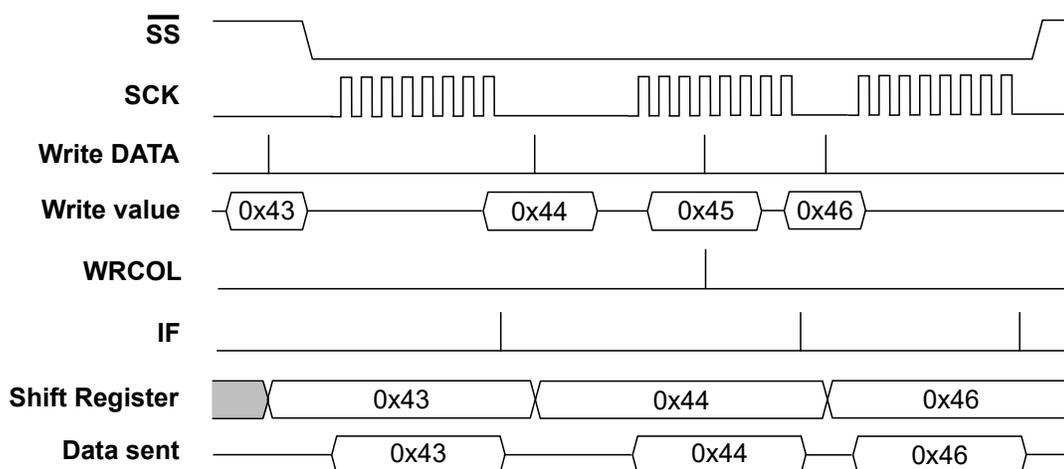
在正常模式下，只要 \overline{SS} 引脚驱动为高电平，SPI 外设便会始终保持空闲状态。在该状态下，软件可能会更新 DATA 寄存器的内容，但在 \overline{SS} 引脚驱动为低电平之前，数据不会随 SCK 引脚上的传入时钟脉冲移出。如果 \overline{SS} 引脚驱动为低电平，从器件将在第一个 SCK 时钟脉冲开始移出数据。完全移出一个字节后，SPIn.INTFLAGS 中的 SPI 中断标志（IF）置 1。

在读取传入数据之前，用户应用程序可以继续将要发送的新数据置于 DATA 寄存器中。在整个传输过程完成之前，不能向 DATA 寄存器写入要发送的新字节。过早写入将被忽略，同时硬件会将 SPIn.INTFLAGS 中的写冲突（WRCOL）标志置 1。

当 \overline{SS} 引脚驱动为高电平时，SPI 逻辑暂停，SPI 从器件不会接收任何新数据。移位寄存器中接收到的任何不完整数据包都将丢失。

图 24-2 给出了正常模式下的发送序列。请注意值 0x45 是如何写入 DATA 寄存器但从未发送的。

图 24-2. 正常模式下的 SPI 时序图（未使能缓冲模式）



上图显示了 SPI 忙于传输时对 DATA 寄存器进行的三次传输和一次写操作。该写操作将被忽略，SPIn.INTFLAGS 中的写冲突（WRCOL）标志置 1。

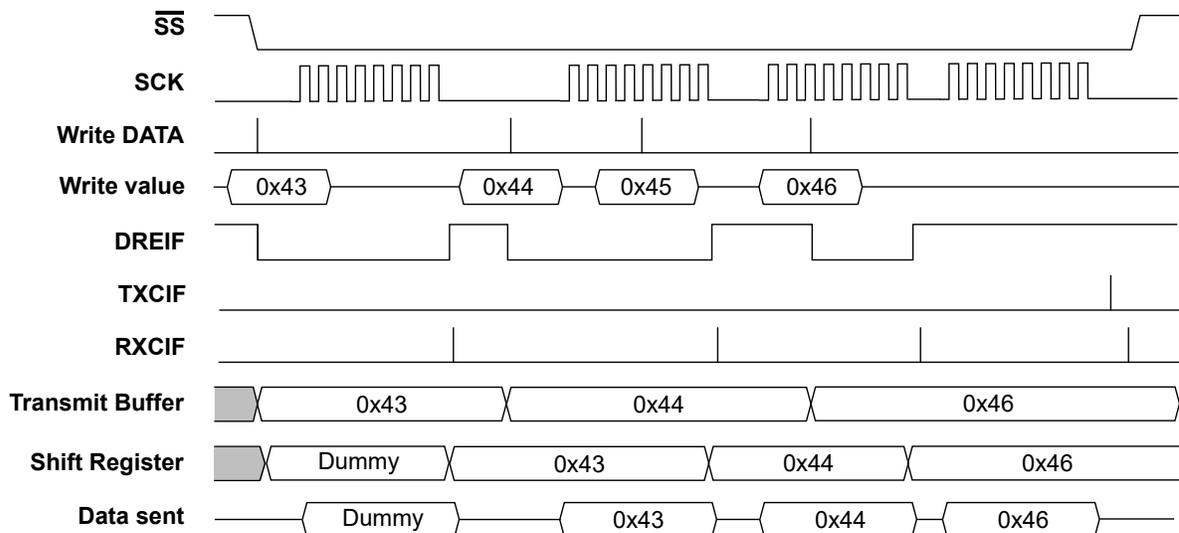
24.3.2.2.2. 缓冲模式

为避免数据冲突，可通过向控制 B（SPIn.CTRLB）寄存器中的缓冲模式使能（BUFEN）位写入 1，将 SPI 外设配置为缓冲模式。在该模式下，SPI 具有额外的中断标志和额外的缓冲区。额外的缓冲区如图 24-1 所示。在缓冲模式下，有两种不同的工作模式，具体可使用缓冲模式等待接收（BUFWR）位来选择。下文以时序图的形式对这两种不同的工作模式进行了介绍。

等待接收位中写入 0 的从缓冲模式

在从模式下，如果 SPIn.CTRLB 中的缓冲模式等待接收（BUFWR）位中写入 0，则在开始发送用户数据前将发送一个空字节。图 24-3 给出了此配置的发送序列。请注意值 0x45 是如何写入数据（SPIn.DATA）寄存器但从未发送的。

图 24-3. SPIn.CTRLB 中的 BUFWR 写入 0 时缓冲模式下的 SPI 时序图



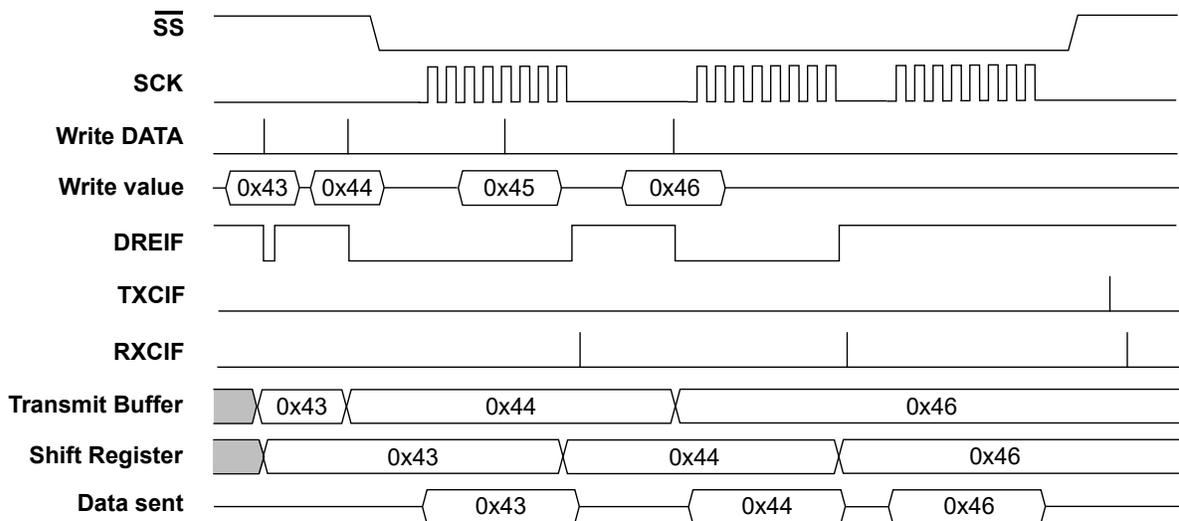
向 SPIn.CTRLB 中的等待接收 (BUFWR) 位写入 0 时, 所有写入数据 (SPIn.DATA) 寄存器中的数据都会进入发送数据缓冲寄存器。由上图可知, 值 0x43 写入到数据 (SPIn.DATA) 寄存器中, 但并未立即传送到移位寄存器, 因此发送的第一个字节是一个空字节。空字节的值等于移位寄存器中的当前值。传送完第一个空字节后, 值 0x43 传送到移位寄存器。然后 0x44 写入到数据 (SPIn.DATA) 寄存器中并进入发送数据缓冲寄存器。随后将开始新的传输, 将发送 0x43。值 0x45 写入到数据 (SPIn.DATA) 寄存器中, 但由于发送数据缓冲寄存器已满 (包含 0x44) 而并未更新, SPIn.INTFLAGS 中的数据寄存器为空中断标志 (DREIF) 为低电平。值 0x45 将丢失。传输结束后, 值 0x44 将移至移位寄存器。在下次传输期间, 0x46 写入到数据 (SPIn.DATA) 寄存器中, 0x44 被发出。传输完成后, 0x46 被复制到移位寄存器中并在下次传送时发出。

每次向发送数据缓冲寄存器写入时, DREIF 都变为低电平, 并在完成传送 (即发送数据缓冲寄存器中的前一个值复制到移位寄存器) 后变为高电平。在 DREIF 变为高电平后, SPIn.INTFLAGS 中的接收完成中断标志 (RXCIF) 置 1 一个周期。在移位寄存器中的值和发送数据缓冲寄存器中的值均已发送的情况下, 当接收完成中断标志置 1 后, 传输完成中断标志将置 1 一个周期。

等待接收位中写入 1 的从缓冲模式

在从模式下, 如果 SPIn.CTRLB 中的缓冲模式等待接收 (BUFWR) 位写入 1, 当 \overline{SS} 引脚被驱动为低电平时, 将立即开始发送用户数据。图 24-4 给出了此配置的发送序列。请注意值 0x45 是如何写入数据 (SPIn.DATA) 寄存器但从未发送的。

图 24-4. CTRLB.BUFWR 写入 1 时缓冲模式下的 SPI 时序图



所有写入到数据 (SPIIn.DATA) 寄存器中的数据都会进入发送数据缓冲寄存器。由上图可知, 值 0x43 被写入数据 (SPIIn.DATA) 寄存器, 由于 \overline{SS} 引脚为高电平, 因此在下一个周期会被复制到移位寄存器。然后下一次写入的数据 (0x44) 将进入发送数据缓冲寄存器。在首次传送期间, 值 0x43 将被移出。在上图中, 值 0x45 被写入数据 (SPIIn.DATA) 寄存器, 但由于 DREIF 为低电平, 发送数据缓冲寄存器并未更新。传输完成后, 发送数据缓冲寄存器中的值 0x44 被复制到移位寄存器。值 0x46 被写入到发送数据缓冲寄存器中。在接下来的两次传输中, 0x44 和 0x46 被移出。标志的行为与 SPIIn.CTRLB 中的缓冲模式等待接收 (BUFWR) 位置 0 时的行为相同。

24.3.2.2.3. \overline{SS} 引脚功能 (从模式下)

从选择 (\overline{SS}) 引脚在 SPI 的操作中起着核心作用。根据 SPI 模式和该引脚的配置, 可使用该引脚激活或禁止器件。 \overline{SS} 引脚用作片选引脚。

在从模式下, \overline{SS} 、MOSI 和 SCK 始终为输入。MISO 引脚的行为取决于端口外设中的引脚配置的数据方向和 \overline{SS} 的值。当 \overline{SS} 引脚驱动为低电平时, SPI 将被激活, 并且如果用户已将 MISO 引脚的数据方向配置为输出, 则 SPI 会通过使数据随时钟移出 MISO 的方式来响应接收的 SCK 脉冲。当 \overline{SS} 引脚驱动为高电平时, SPI 将被禁止, 这意味着其不会接收传入数据。如果 MISO 引脚数据方向配置为输出, 则 MISO 引脚将处于三态。表 24-3 给出了 \overline{SS} 引脚功能的概览。

表 24-3. \overline{SS} 引脚功能的概览

\overline{SS} 配置	\overline{SS} 引脚电平	说明	MISO 引脚模式	
			端口方向 = 输出	端口方向 = 输入
始终为输入	高电平	禁止 (取消选择) 从模式	三态	输入
	低电平	激活 (选择) 从模式	输出	输入

注: 在从模式下, SPI 状态机将在 \overline{SS} 引脚驱动为高电平时复位。如果 \overline{SS} 引脚在发送过程中驱动为高电平, 则 SPI 会立即停止发送和接收数据, 接收的数据和发送的数据都必须视为丢失。由于 \overline{SS} 引脚用于指示传输的开始和结束, 因此可用于实现数据包/字节同步以及使从模式位计数器与主时钟发生器保持同步。

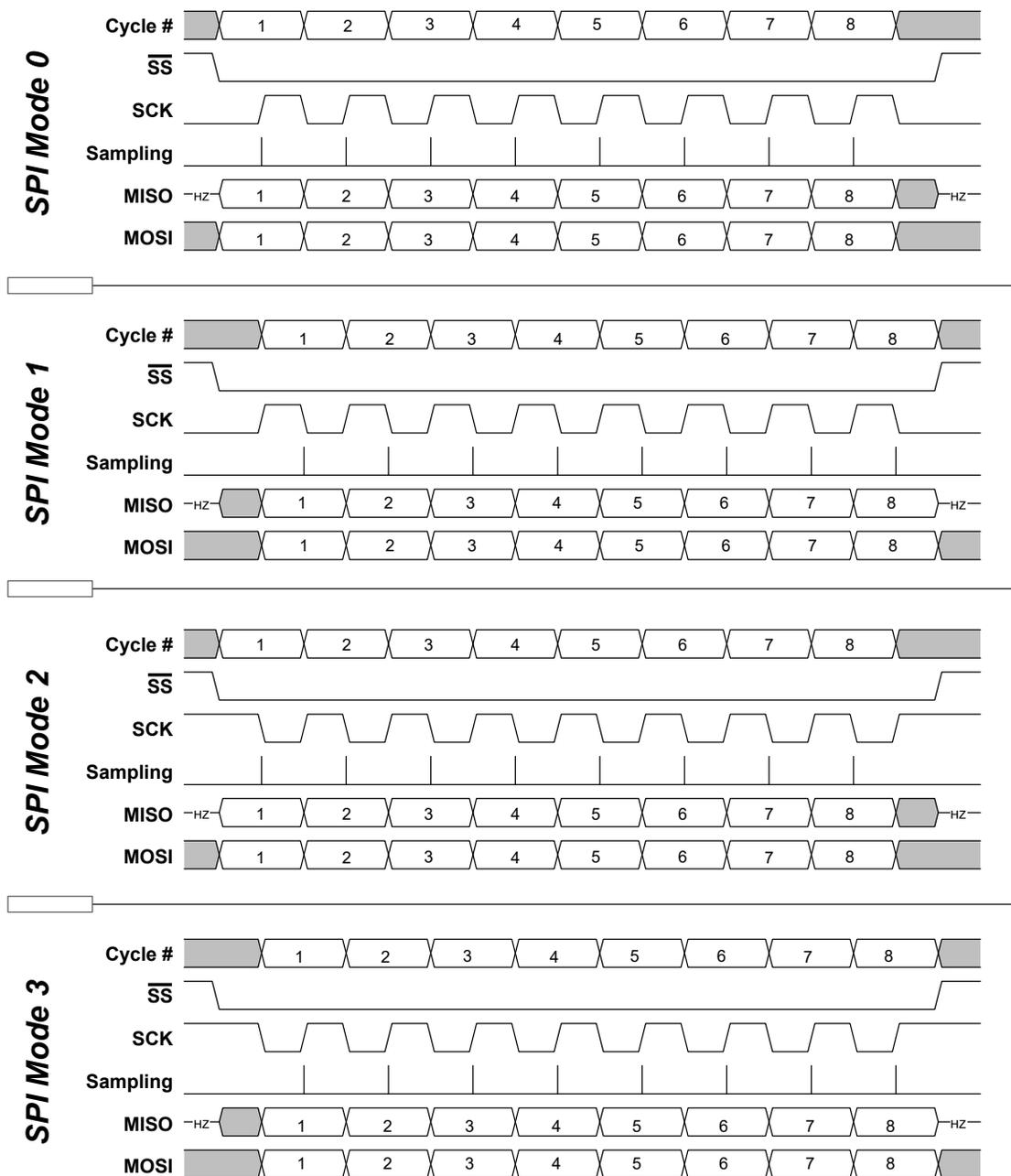
24.3.2.3. 数据模式

串行数据的 SCK 相位和极性有四种组合。通过对控制 B (SPIIn.CTRLB) 寄存器中的 MODE 位进行写操作来选择所需组合。

SPI 数据传输格式如下图所示。数据位分别在 SCK 信号的两种相反边沿进行移出和锁存, 以确保数据信号有充足的时间达到稳定状态。

前沿是时钟周期的第一个时钟边沿。后沿是时钟周期的最后一个时钟边沿。

图 24-5. SPI 数据传输模式



24.3.2.4. 事件

SPI 可生成以下事件：

表 24-4. SPI 中的事件生成器

生成器名称		说明	事件类型	生成时钟域	事件时长
模块	事件				
SPIin	SCK	SPI 主时钟	电平	CLK_PER	最短 2 个 CLK_PER 周期

SPI 没有事件用户。

有关事件类型和事件系统配置的更多详细信息，请参见 *事件系统* 一章。

24.3.2.5. 中断

表 24-5. 可用中断向量和中断源

名称	向量说明	条件	
		正常模式	缓冲模式
SPI _n	SPI 中断	<ul style="list-style-type: none"> IF: 中断标志中断 WRCOL: 写冲突中断 	<ul style="list-style-type: none"> SSI: 从选择触发中断 DRE: 数据寄存器为空中断 TXC: 发送完成中断 RXC: 接收完成中断

发生中断条件时，外设的中断标志（*外设*.INTFLAGS）寄存器中的相应中断标志将置 1。

可以通过写入外设的中断控制（*外设*.INTCTRL）寄存器中的相应允许位来允许或禁止中断源。

当相应的中断源被允许且中断标志置 1 时，将生成中断请求。中断请求保持有效，直到中断标志清零。有关如何清零中断标志的详细信息，请参见外设的 INTFLAGS 寄存器。

24.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0		DORD	MASTER	CLK2X		PRESC[1:0]		ENABLE
0x01	CTRLB	7:0	BUFEN	BUFWR				SSD	MODE[1:0]	
0x02	INTCTRL	7:0	RXCIE	TXCIE	DREIE	SSIE				IE
0x03	INTFLAGS	7:0	IF	WRCOL						
0x03	INTFLAGS	7:0	RXCIF	TXCIF	DREIF	SSIF				BUFOVF
0x04	DATA	7:0	DATA[7:0]							

24.5. 寄存器说明

24.5.1. 控制 A

名称: CTRLA
偏移量: 0x00
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
		DORD	MASTER	CLK2X		PRESC[1:0]		ENABLE
访问		R/W	R/W	R/W		R/W	R/W	R/W
复位		0	0	0		0	0	0

Bit 6 – DORD 数据顺序

值	说明
0	先发送数据字的 MSb
1	先发送数据字的 LSb

Bit 5 – MASTER 主/从选择

该位用于选择所需的 SPI 模式。

如果 SS 在该位为 1 时配置为输入并驱动为低电平，则该位清零且 SPIn.INTFLAGS 中的 IF 置 1。用户必须再次写入 MASTER = 1 以重新使能 SPI 主模式。

此行为由 SPIn.CTRLB 中的从选择禁止（SSD）位控制。

值	说明
0	选择 SPI 从模式
1	选择 SPI 主模式

Bit 4 – CLK2X 时钟加倍

向该位写入 1 时，主模式下的 SPI 速度（内部预分频器后的 SCK 频率）加倍。

值	说明
0	SPI 速度（SCK 频率）不加倍
1	主模式下的 SPI 速度（SCK 频率）加倍

Bit 2:1 – PRESC[1:0] 预分频器

该位域控制主模式下配置的 SPI 时钟速率。这些位在从模式下不起作用。SCK 与外设时钟频率（ f_{CLK_PER} ）之间的关系如下所示。

向 CLK2X 位写入 1，可将 SPI 预分频器的输出加倍。

值	名称	说明
0x0	DIV4	$CLK_PER/4$
0x1	DIV16	$CLK_PER/16$
0x2	DIV64	$CLK_PER/64$
0x3	DIV128	$CLK_PER/128$

Bit 0 – ENABLE SPI 使能

值	说明
0	禁止 SPI
1	使能 SPI

24.5.2. 控制 B

名称: CTRLB
偏移量: 0x01
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	BUFEN	BUFWR				SSD	MODE[1:0]	
访问	R/W	R/W				R/W	R/W	R/W
复位	0	0				0	0	0

Bit 7 - BUFEN 缓冲模式使能

向该位写入 1 可使能缓冲模式。该操作将使能两个接收缓冲区和一个发送缓冲区。发送完成和接收完成均有单独的中断标志。

Bit 6 - BUFWR 缓冲模式等待接收

向该位写入 0 时，传输的第一个数据将为空字节样本。

值	说明
0	在将数据复制到移位寄存器之前，必须完成一次 SPI 传输。
1	如果在 SPI 使能且 \overline{SS} 为高电平时写入数据寄存器，则第一次写入的内容将直接进入移位寄存器

Bit 2 - SSD 从选择禁止

当该位置 1 且以 SPI 主模式 (SPIn.CTRLA 中的 MASTER = 1) 工作时， \overline{SS} 不会禁止主模式。

值	说明
0	以 SPI 主模式工作时，使能从选择线
1	以 SPI 主模式工作时，禁止从选择线

Bit 1:0 - MODE[1:0] 模式

这些位用于选择传输模式。串行数据的 SCK 相位和极性的四种组合如下表所示。这些位决定时钟周期的第一个边沿（前沿）是上升沿还是下降沿以及前沿或后沿上是否发生数据建立和采样。当前沿为上升沿时，SCK 信号在空闲时为低电平；当前沿为下降沿时，SCK 信号在空闲时为高电平。

值	名称	说明
0x0	0	前沿：上升沿，采样 后沿：下降沿，建立
0x1	1	前沿：上升沿，建立 后沿：下降沿，采样
0x2	2	前沿：下降沿，采样 后沿：上升沿，建立
0x3	3	前沿：下降沿，建立 后沿：上升沿，采样

24.5.3. 中断控制

名称: INTCTRL
 偏移量: 0x02
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	DREIE	SSIE				IE
访问	R/W	R/W	R/W	R/W				R/W
复位	0	0	0	0				0

Bit 7 - RXCIE 接收完成中断允许

在缓冲模式下，该位可允许接收完成中断。允许的中断将在 SPIn.INTFLAGS 寄存器中的 RXCIF 置 1 时触发。在非缓冲模式下，该位为 0。

Bit 6 - TXCIE 发送完成中断允许

在缓冲模式下，该位可允许传输完成中断。允许的中断将在 SPIn.INTFLAGS 寄存器中的 TXCIF 置 1 时触发。在非缓冲模式下，该位为 0。

Bit 5 - DREIE 数据寄存器为空中断允许

在缓冲模式下，该位可允许数据寄存器为空中断。允许的中断将在 SPIn.INTFLAGS 寄存器中的 DREIF 置 1 时触发。在非缓冲模式下，该位为 0。

Bit 4 - SSIE 从选择触发中断允许

在缓冲模式下，该位可允许从选择中断。允许的中断将在 SPIn.INTFLAGS 寄存器中的 SSIF 置 1 时触发。在非缓冲模式下，该位为 0。

Bit 0 - IE 中断允许

当 SPI 未处于缓冲模式时，该位可允许 SPI 中断。允许的中断将在 SPIn.INTFLAGS 寄存器中的 RXCIF/IF 置 1 时触发。

24.5.4. 中断标志——正常模式

名称: INTFLAGS
 偏移量: 0x03
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	IF	WRCOL						
访问	R/W	R/W						
复位	0	0						

Bit 7 - IF 中断标志

当串行传输完成且一个字节完全移入/移出 SPIn.DATA 寄存器时，该标志置 1。如果 \overline{SS} 在 SPI 处于主模式时配置为输入并驱动为低电平，也会将该标志置 1。可在执行相应的中断向量时由硬件将 IF 清零。也可通过先读取 SPIn.INTFLAGS 寄存器（IF 置 1 时）再访问 SPIn.DATA 寄存器的方式将 IF 清零。

Bit 6 - WRCOL 写冲突

如果在完整字节移出之前写入 SPIn.DATA 寄存器，则 WRCOL 标志置 1。可通过先读取 SPIn.INTFLAGS 寄存器（WRCOL 置 1 时）再访问 SPIn.DATA 寄存器的方式将该标志清零。

24.5.5. 中断标志——缓冲模式

名称: INTFLAGS
偏移量: 0x03
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	RXCIF	TXCIF	DREIF	SSIF				BUFOVF
访问	R/W	R/W	R/W	R/W				R/W
复位	0	0	0	0				0

Bit 7 - RXCIF 接收完成中断标志

当接收数据缓冲寄存器中有未读数据时，该标志置 1；当接收数据缓冲寄存器为空时（即不包含任何未读数据），该标志清零。

当使用中断驱动的数据接收时，接收完成中断程序必须从数据寄存器读取接收的数据，以将 RXCIF 清零。否则，从当前中断返回后将直接发生新中断。向该标志的位存储单元写入 1 可将其清零。

Bit 6 - TXCIF 传输完成中断标志

发送移位寄存器中的全部数据已移出且发送缓冲区（SPIn.DATA）中没有新数据时，该标志置 1。向该标志的位存储单元写入 1 可将其清零。

Bit 5 - DREIF 数据寄存器为空中断标志

该标志指示发送数据缓冲寄存器是否准备好接收新数据。当发送缓冲区为空时，该标志置 1；当发送缓冲区包含尚未移入移位寄存器的待发送数据时，该标志为 0。DREIF 在复位后清零，表示发送器已准备就绪。

对数据寄存器进行写操作可将 DREIF 清零。当使用中断驱动的数据发送时，数据寄存器为空中断程序必须向数据寄存器写入新数据以将 DREIF 清零，或禁止数据寄存器为空中断。否则，从当前中断返回后将直接发生新中断。

Bit 4 - SSIF 从选择触发中断标志

该标志指示 SPI 已处于主模式且 \overline{SS} 引脚已从外部拉低，因此 SPI 现在以从模式工作。仅当从选择禁止（SSD）位不为 1 时，该标志才置 1。向该标志的位存储单元写入 1 可将其清零。

Bit 0 - BUFOVF 缓冲区溢出

该标志指示由于接收数据缓冲区已满而导致数据丢失。如果检测到缓冲区溢出情况，则该标志置 1。当接收缓冲区已满（两个字节）且移位寄存器中已接收到第三个字节时，发生缓冲区溢出。如果没有发送数据，则在新的串行传输开始之前缓冲区溢出标志不会置 1。读取数据寄存器或向该标志的位存储单元写入 1 即可将该标志清零。

24.5.6. 数据

名称: DATA
 偏移量: 0x04
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	DATA[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - DATA[7:0] SPI 数据

数据寄存器用于发送和接收数据。在主模式下写入寄存器将启动数据发送，在从模式下写入寄存器将准备好要发送的数据。启动事务时，写入寄存器的字节将在 SPI 输出线上移出。

SPI_{IN}.DATA 寄存器不是物理寄存器。根据配置的模式，该寄存器被映射到其他寄存器，如下所述。

- 正常模式：
 - 写入数据寄存器时将写入移位寄存器
 - 读取数据寄存器时将读取接收数据寄存器
- 缓冲模式：
 - 写入数据寄存器时将写入发送数据缓冲寄存器。
 - 读取数据寄存器时将读取接收数据缓冲寄存器。随后，接收数据寄存器的内容将移至接收数据缓冲寄存器。

25. TWI——双线接口

25.1. 特性

- 双线通信接口
- 兼容 Philips I²C
 - 标准模式
 - 快速模式
 - 增强型快速模式
- 兼容系统管理总线（System Management Bus, SMBus）2.0
 - 支持在启动/重复启动条件和数据位之间仲裁
 - 从器件仲裁允许通过软件为地址解析协议（Address Resolution Protocol, ARP）提供支持
 - 可在硬件中配置 SMBus 第 1 层超时
- 主器件和从器件操作相互独立
 - 组合模式（相同引脚）
 - 支持单主器件或多主器件总线操作，提供全面的仲裁支持
- 针对从器件地址匹配的硬件支持
 - 可在所有休眠模式下工作
 - 7 位地址识别
 - 广播呼叫地址识别
 - 支持地址范围掩码或第二个地址匹配
- 用于抑制总线噪声的输入滤波器
- 支持智能模式

25.2. 概述

双线接口（TWI）是双向双线通信接口（总线），包含一条串行数据线（Serial Data Line, SDA）和一条串行时钟线（Serial Clock Line, SCL）。

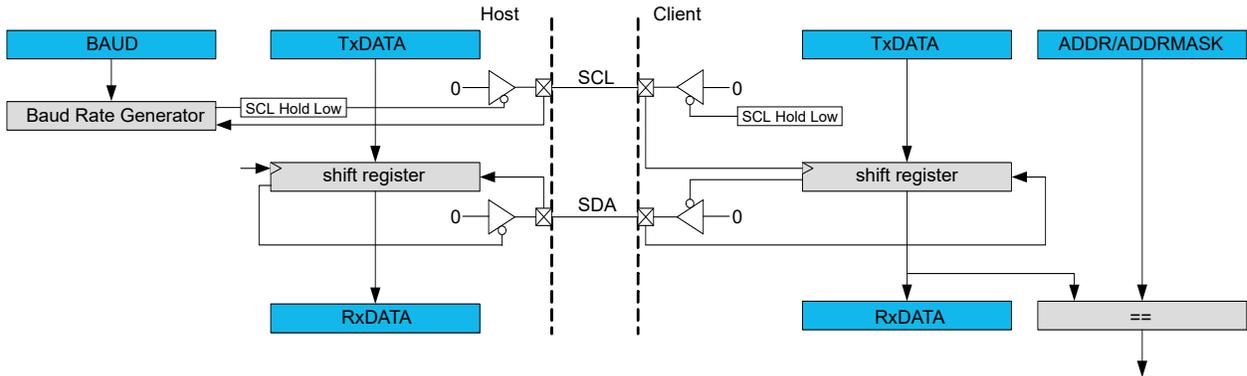
TWI 总线可将一个或多个从器件连接到一个或多个主器件。连接到总线的任何器件都可以充当主器件和/或从器件。主器件使用波特率发生器（Baud Rate Generator, BRG）生成 SCL，并通过寻址一个从器件并告知是要发送还是接收数据来启动数据事务。BRG 能够生成标准模式（Sm）和快速模式（Fm 和 Fm+）总线频率，范围从 100 kHz 到 1 MHz。

TWI 将检测启动和停止条件、总线冲突以及总线错误。此外，无论在主模式还是从模式下，都将检测仲裁失败、错误、冲突和时钟保持情况，并通过可用的独立状态标志进行指示。

TWI 支持多主器件总线操作和仲裁。仲裁方案用于处理多个主器件同时尝试发送数据的情况。此外，TWI 还支持智能模式，可自动触发操作，从降低软件复杂度。TWI 支持快速命令模式，允许主机在不交换数据的情况下寻址从器件。

25.2.1. 框图

图 25-1. TWI 框图



25.2.2. 信号说明

信号	说明	类型
SCL	串行时钟线	数字 I/O
SDA	串行数据线	数字 I/O

25.3. 功能说明

25.3.1. 一般 TWI 总线概念

TWI 提供简单的双向双线通信总线，其中包括：

- 用于数据包传输的串行数据线（SDA）
- 用于总线时钟的串行时钟线（SCL）

这两条线均为集电极开路线（线与）。

TWI 总线拓扑可轻松高效地实现在串行总线上互连多个器件。连接到该总线的器件可以是主器件，也可以是从器件。只有主器件可以控制总线和总线通信。

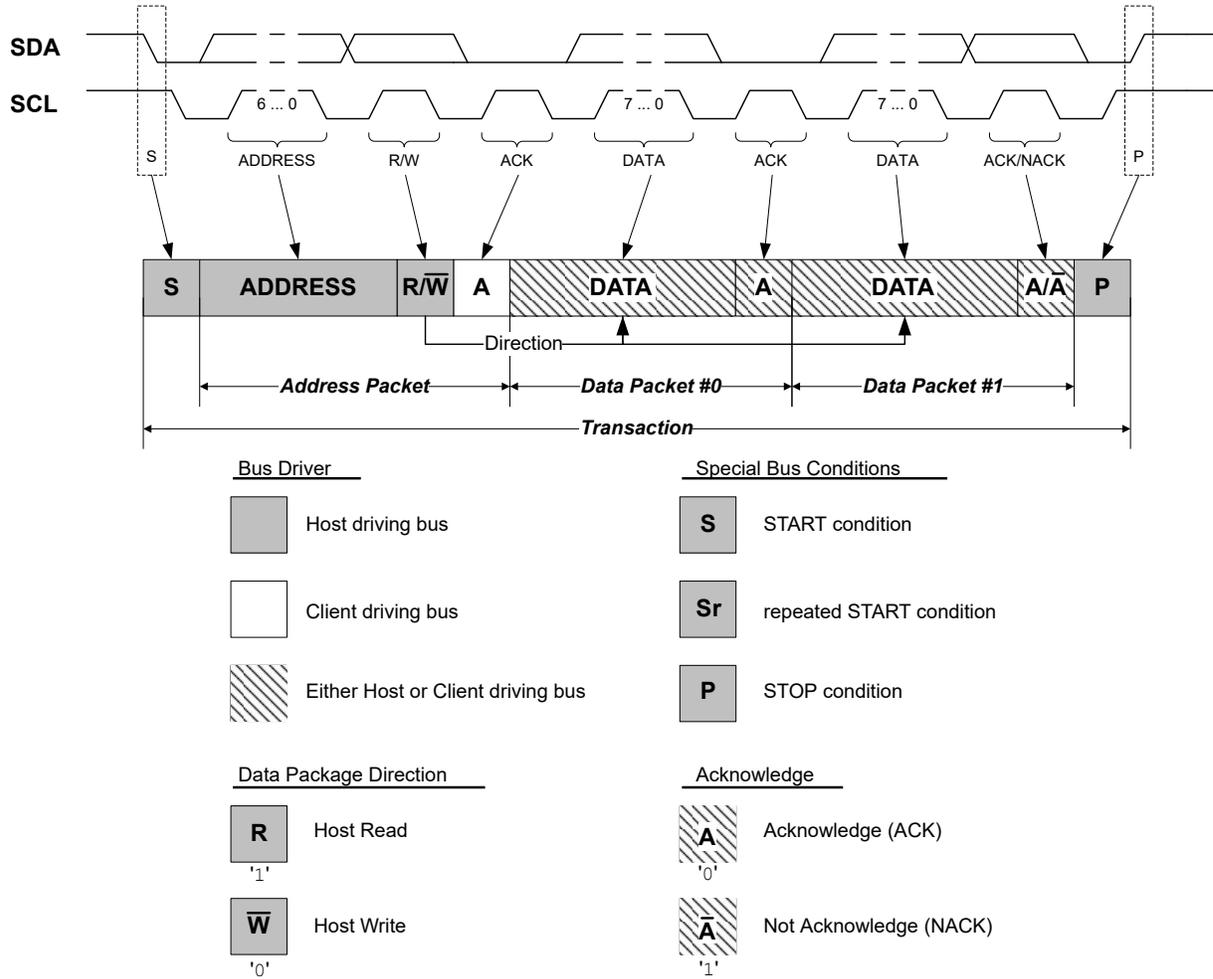
每个连接到总线的从器件均会获得一个惟一地址，主器件将使用该地址来控制从器件并启动事务。多个主器件可连接到同一总线，这称为多主器件环境。由于在任意给定时间只有一个主器件可拥有总线，因此我们提供仲裁机制来解决主器件之间的总线所有权问题。

主器件通过在总线上发出启动条件（S）来指示事务已开始。主器件为事务提供时钟信号。随后发送包含 7 位从器件地址（ADDRESS）和方向位的地址包，表示主器件是要读取还是写入数据（R/W）。

然后，已寻址的 I²C 从器件将确认（ACK）地址，数据包事务可以开始。每个 9 位数据包中包含 8 个数据位，后跟一个回复位，指示接收器是否确认数据。

传送完所有数据包（DATA）后，主器件会在总线上发出停止条件（P）以结束事务。

图 25-2.7 7 位地址总线的基本 TWI 事务图拓扑



25.3.2. TWI 基本操作

25.3.2.1. 初始化

在使能 TWI 器件之前，必须配置以下位（如使用）：

- 控制 A (TWIn.CTRLA) 寄存器中的 SDA 保持时间 (SDAHOLD) 位域
- 控制 A (TWIn.CTRLA) 寄存器中的 FM+使能 (FMPEN) 位

25.3.2.1.1. 主器件初始化

向主波特率 (TWIn.MBAUD) 寄存器写入可以使 TWI 总线时钟频率有效的值。向主器件控制 A (TWIn.MCTRLA) 寄存器中的使能 TWI 主器件 (ENABLE) 位写入 1 将使能 TWI 主器件。主器件状态 (TWIn.MSTATUS) 寄存器的总线状态 (BUSSTATE) 位域必须设为 0x1 才能强制总线进入空闲状态。

25.3.2.1.2. 从器件初始化

要初始化从器件，请按照以下步骤操作：

1. 使能 TWI 器件之前，配置控制 A (TWIn.CTRLA) 寄存器中的 SDA 建立时间 (SDASETUP) 位。
2. 将从器件地址写入从器件地址 (TWIn.SADDR) 寄存器。
3. 向从器件控制 A (TWIn.SCTRLA) 寄存器中的使能 TWI 从器件 (ENABLE) 位写入 1 以使能 TWI 从器件。

从器件现在将等待主器件发出启动条件和匹配的从器件地址。

25.3.2.2. TWI 主器件操作

TWI 主器件面向字节，每个字节后均有一个可选中断。主器件写和读操作分别具有单独的中断标志。中断标志还可用于轮询操作。此外，还有专用于指示收到 ACK/NACK、总线错误、仲裁失败、时钟保持和总线状态的状态标志。

中断标志设置 1 时，SCL 被强制为低电平。这会为主器件留出响应时间或数据处理时间，在大多数情况下需要软件交互。将中断标志清零会释放 SCL。通过自动处理大多数条件，可使生成的中断数保持为最小值。

25.3.2.2.1. 时钟生成

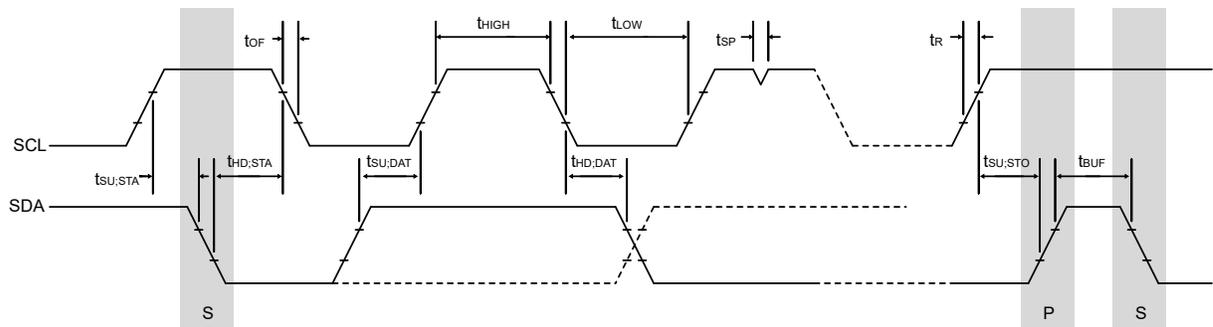
TWI 支持具备不同频率限制的多种发送模式：

- 标准模式（Sm），最高 100 kHz
- 快速模式（Fm），最高 400 kHz
- 增强型快速模式（Fm+），最高 1 MHz

向主波特率（TWIn.MBAUD）寄存器写入可以使 TWI 总线时钟频率小于或等于频率限制的值，具体取决于发送模式。

低电平（ t_{LOW} ）和高电平（ t_{HIGH} ）时间由主波特率（TWIn.MBAUD）寄存器决定，而上升（ t_R ）和下降（ t_{OF} ）时间则由总线拓扑决定。

图 25-3. SCL 时序



- t_{LOW} 为 SCL 时钟的低电平周期
- t_{HIGH} 为 SCL 时钟的高电平周期
- t_R 由总线阻抗决定，表示内部上拉。有关详细信息，请参见 *电气特性* 一章。
- t_{OF} 为输出下降时间，由漏极开路电流限制和总线阻抗决定。有关详细信息，请参见 *电气特性* 一章。

SCL 时钟的属性

SCL 频率的计算公式如下：

公式 25-1. SCL 频率

$$f_{SCL} = \frac{1}{t_{LOW} + t_{HIGH} + t_{OF} + t_R} [\text{Hz}]$$

SCL 时钟采用具有 50/50 占空比的设计，其中占空比的低电平周期由 t_{OF} 和 t_{LOW} 组成。只有检测到 SCL 为高电平时， t_{HIGH} 才会启动。TWIn.MBAUD 寄存器中的 BAUD 位域与 SCL 频率通过以下公式关联：

公式 25-2. SCL 频率

$$f_{SCL} = \frac{f_{CLK_PER}}{10 + 2 \times BAUD + f_{CLK_PER} \times t_R}$$

公式 25-2 可进行转换以表示 BAUD:

公式 25-3. BAUD

$$BAUD = \frac{f_{CLK_PER}}{2 \times f_{SCL}} - \left(5 + \frac{f_{CLK_PER} \times t_R}{2} \right)$$

计算 BAUD 值

为确保在所需速度模式（Sm、Fm、Fm+）的规范内工作，请执行以下步骤:

1. 使用公式 25-3 计算 BAUD 位域的值。
2. 使用步骤 1 中的 BAUD 值计算 t_{LOW} :

公式 25-4. t_{LOW}

$$t_{LOW} = \frac{BAUD + 5}{f_{CLK_PER}} - t_{OF}$$

3. 检查从公式 25-4 得出的 t_{LOW} 是否高于所需模式的指定最小值（ $t_{LOW_Sm} = 4700$ ns, $t_{LOW_Fm} = 1300$ ns, $t_{LOW_Fm+} = 500$ ns）。
 - 如果计算出的 t_{LOW} 高于限值，请使用从公式 27-3 得出的 BAUD 值 公式 25-3
 - 如果不符合限制条件，请使用下面的公式 25-5 计算新的 BAUD 值，其中 t_{LOW_mode} 为模式规范中给出的 t_{LOW_Sm} 、 t_{LOW_Fm} 或 t_{LOW_Fm+} :

公式 25-5. BAUD

$$BAUD = f_{CLK_PER} \times (t_{LOW_mode} + t_{OF}) - 5$$

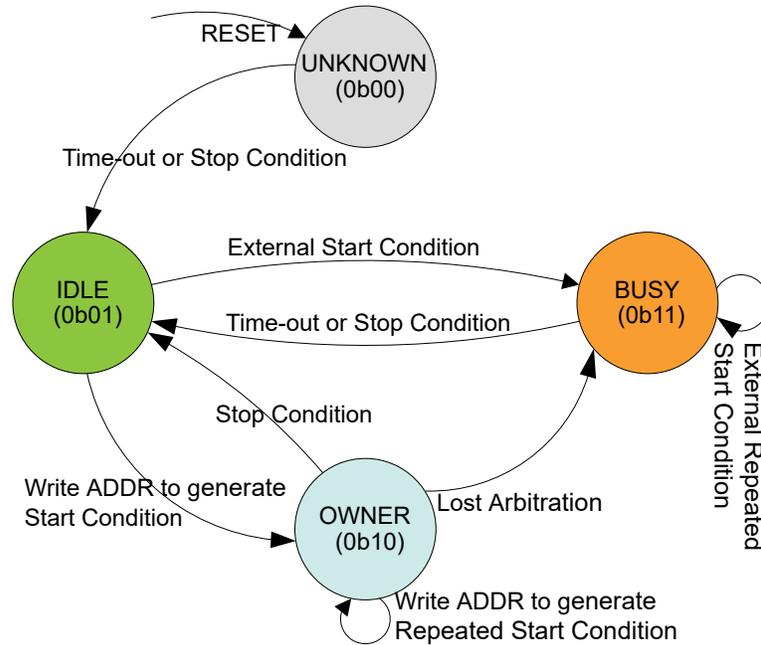
25.3.2.2.2. TWI 总线状态逻辑

如果使能主器件，则总线状态逻辑会持续监视 TWI 总线上的活动。它在所有休眠模式（包括掉电模式）下均会继续工作。

总线状态逻辑包括启动和停止条件检测器、冲突检测、非活动总线超时检测和位计数器。这一切均用于确定总线状态。软件可通过读取主器件状态（TWIn.MSTATUS）寄存器中的总线状态（BUSSTATE）位域来获取当前总线状态。

总线状态可以是未知、空闲、繁忙或所有者，具体根据下面所示的状态图进行确定。

图 25-4. 总线状态图



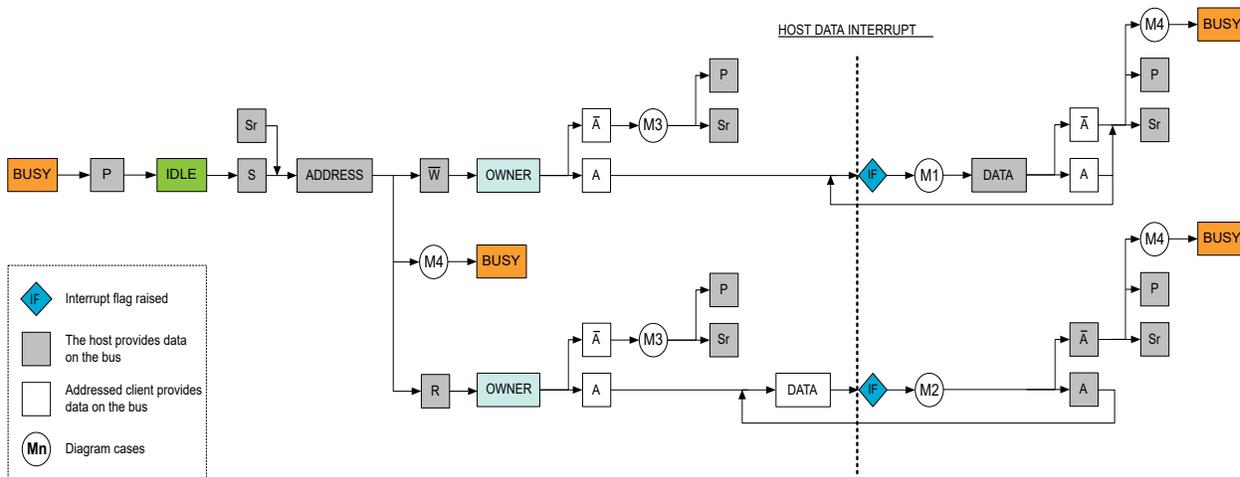
1. **未知**：使能 TWI 主器件时，总线状态机激活。使能 TWI 主器件后，总线状态未知。执行系统复位或禁止 TWI 主器件后，总线状态也会设置为未知。
2. **空闲**：通过向总线状态（BUSSTATE）位域写入 0x1，可以强制总线状态机进入空闲状态。不能强制总线状态逻辑进入任何其他状态。如果应用软件未设置状态，则总线状态将在检测到第一个停止条件时变为空闲状态。如果主器件控制 A（TWIn.MCTRLA）寄存器中的非活动总线超时（TIMEOUT）位域配置为非零值，则在发生超时总线状态将变为空闲。总线空闲时，即准备好进行新的事务。
3. **繁忙**：如果在总线空闲时检测到外部生成的启动条件，则总线变为繁忙状态。当检测到停止条件或设置了超时（如果已配置）时，总线变回空闲状态。
4. **所有者**：如果在总线空闲的状态下在内部生成启动条件，则总线变为所有者状态。如果在无干扰的情况下执行了完整事务，则主器件将发出停止条件，总线状态将切换回空闲。如果检测到冲突，则假定仲裁失败并且总线状态变为繁忙，直到检测到停止条件。

25.3.2.2.3. 发送地址包

如果向主器件地址（TWIn.MADDR）寄存器写入从器件地址和 R/\bar{W} 方向位，主器件将开始执行总线事务。MADDR 寄存器的值随后被复制到主器件数据（TWIn.MDATA）寄存器。如果总线为繁忙状态，TWI 主器件将等到总线变为空闲状态后再发出启动条件。TWI 将发出启动条件，移位寄存会在总线上执行字节发送操作。

根据仲裁和 R/\bar{W} 方向位，在发送地址包之后会出现四种情况之一（M1 到 M4）。

图 25-5. TWI 主器件操作



情况 M1：地址包发送完成——方向位设为 0

如果从器件通过 ACK 来响应地址包，写中断标志（WIF）设置为 1，接收应答（RXACK）标志设置为 0，时钟保持（CLKHOLD）标志设置为 1。WIF、RXACK 和 CLKHOLD 标志位于主器件状态（TWIn.MSTATUS）寄存器中。

此时，时钟保持功能激活，强制将 SCL 设为低电平。该操作将延长时钟的低电平周期，从而降低整体时钟频率，强制生成处理数据所需的延时，并防止总线上发生进一步活动。

软件可准备：

- 向从器件发送数据包

情况 M2：地址包发送完成——方向位设为 1

如果从器件通过 ACK 来响应地址包，RXACK 标志设置为 0，则因为此时从器件具备总线的所有权，所以可以开始向主器件发送数据而无任何延时。此时，时钟保持功能激活，强制将 SCL 设为低电平。

软件可准备：

- 从从器件读取接收的数据包

情况 M3：地址包发送完成——从器件未确认地址

如果没有从器件响应地址包，则 WIF 和 RXACK 标志将置 1。此时，时钟保持功能激活，强制将 SCL 设为低电平。

如果 ACK 响应缺失，说明 I²C 从器件可能在忙于其他任务，或处于休眠模式而无法做出应答。

软件可准备执行以下操作之一：

- 重新发送地址包
- 通过在主器件控制 B（TWIn.MCTRLB）寄存器中的命令（MCMD）位域中发出停止条件来完成事务（建议操作）

情况 M4：仲裁失败或总线错误

如果仲裁失败，主器件状态（TWIn.MSTATUS）寄存器中的 WIF 和仲裁失败（ARBLOST）标志都将置 1。SDA 被禁止，SCL 被释放。总线状态变为繁忙，在总线状态变回空闲状态之前，不再允许主器件在总线上执行任何操作。

总线错误与仲裁失败的情况相似。在这种情况下，除 WIF 和 ARBLOST 标志外，主器件状态（TWIn.MSTATUS）寄存器中的总线错误（BUSERR）标志也将置 1。

软件可准备：

- 中止操作并等待总线变回空闲状态（通过读取主器件状态（TWIn.MSTATUS）寄存器中的总线状态（BUSSTATE）位域来判断）。

25.3.2.2.4. 发送数据包

假设出现上述的 M1 情况，TWI 主器件可通过对主器件数据（TWIn.MDATA）寄存器进行写操作来启动数据发送，该写操作还可以将写中断标志（WIF）清零。在数据传输期间，主器件会持续监控总线是否存在冲突和错误。数据包传输完成后，WIF 标志将设置为 1。

如果发送成功并且主器件收到来自从器件的 ACK 位，则接收应答（RXACK）标志将设置为 0，表示从器件已准备好接收新的数据包。

软件可准备执行以下操作之一：

- 发送新的数据包
- 发送新的地址包
- 通过在主器件控制 B（TWIn.MCTRLB）寄存器中的命令（MCMD）位域中发出停止条件来完成事务

如果发送成功并且主器件收到来自从器件的 NACK 位，则 RXACK 标志将设置为 1，表示从器件无法或不需要接收更多数据。

软件可准备执行以下操作之一：

- 发送新的地址包
- 通过在主器件控制 B（TWIn.MCTRLB）寄存器中的命令（MCMD）位域中发出停止条件来完成事务

只有当 WIF 标志设置为 1 并且仲裁失败（ARBLOST）和总线错误（BUSERR）标志设置为 0 时，RXACK 状态才有效。

如果检测到冲突，发送可能会失败。随后，主器件仲裁失败，仲裁失败（ARBLOST）标志将设置为 1，总线变为繁忙状态。数据包发送期间仲裁失败的情况与上述 M4 情况的处理方式相同。

WIF、ARBLOST、BUSERR 和 RXACK 标志均位于主器件状态（TWIn.MSTATUS）寄存器中。

25.3.2.2.5. 接收数据包

假设出现上述的 M2 情况，时钟会释放一个字节，以允许从器件在总线上放置一个字节的的数据。主器件将接收来自从器件的一个字节的数据，读中断标志（RIF）和时钟保持（CLKHOLD）标志将设置为 1。向 TWIn.MCTRLB 寄存器中的命令（MCMD）位域写入命令时，通过主器件控制 B（TWIn.MCTRLB）寄存器中的应答动作（ACKACT）位选择的动作将在总线上自动发送。

软件可准备执行以下操作之一：

- 通过向 TWIn.MCTRLB 寄存器中的 ACKACT 位写入 0 以使用 ACK 进行响应，并准备接收新的数据包
- 通过向 ACKACT 位写入 1 以使用 NACK 进行响应，然后发送新的地址包
- 通过向 ACKACT 位写入 1 以使用 NACK 进行响应，然后通过向 TWIn.MCTRLB 寄存器的 MCMD 位域发出停止条件来完成事务

NACK 响应可能无法成功执行，因为发送期间仲裁可能失败。如果检测到冲突，主器件仲裁将失败，仲裁失败（ARBLOST）标志将设置为 1，总线变为繁忙状态。如果在发送 NACK 时仲裁失败，或者发送过程中发生总线错误，则主器件写中断标志（WIF）将置 1。数据包发送期间仲裁失败的情况与上述 M4 情况的处理方式相同。

RIF、CLKHOLD、ARBLOST 和 WIF 标志均位于主器件状态（TWIn.MSTATUS）寄存器中。

注：RIF 和 WIF 标志是互斥关系，不能同时置 1。

25.3.2.3. TWI 从器件操作

TWI 从器件面向字节，每个字节后均有可选中断。从器件数据和地址/停止识别分别使用不同的中断标志。中断标志还可用于轮询操作。提供专用状态标志，用于指示 ACK/NACK 已收到、时钟保持、冲突、总线错误以及 R/W 方向位，主器件将开始执行总线事务。

中断标志设置 1 时，SCL 被强制为低电平。这会为从器件留出响应时间或数据处理时间，在大多数情况下需要软件交互。通过自动处理大多数条件，可使生成的中断数保持为最小值。

可以配置从器件控制 A (TWIn.SCTRLA) 寄存器中的地址识别模式 (PMEN) 位，使从器件能够响应所有接收的地址。

25.3.2.3.1. 接收地址包

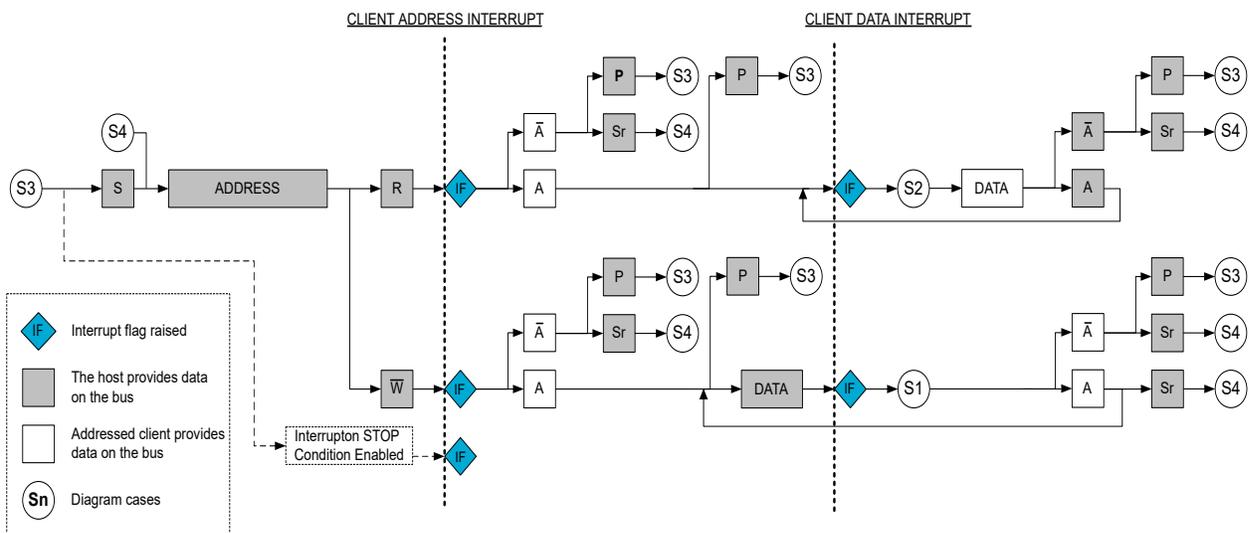
TWI 配置为从器件时，会一直等待，直到检测到启动条件。出现该条件时，将接收连续的地址数据包并通过地址匹配逻辑进行检查。从器件将对正确的地址进行确认并将该地址存储在从器件数据 (TWIn.SDATA) 寄存器中。如果接收到的地址不匹配，从器件将不会确认或存储该地址，而是会等待新的启动条件。

当启动条件后跟以下几项之一时，从器件状态 (TWIn.SSTATUS) 寄存器中的地址或停止中断标志 (APIF) 将置 1:

- 与从器件地址 (TWIn.SADDR) 寄存器中的地址 (ADDR[7:1]) 位域中存储的地址有效匹配
- 广播呼叫地址 (0x00)，并且从器件地址 (TWIn.SADDR) 寄存器中的地址 (ADDR[0]) 位置 1
- 与地址掩码 (ADDRMASK) 位域中存储的第二个地址有效匹配，并且从器件地址掩码 (TWIn.SADDRMASK) 寄存器中的地址掩码使能 (ADDREN) 位置 1
- 任何地址，但前提是从器件控制 A (TWIn.SCTRLA) 寄存器中的地址识别模式 (PMEN) 位置 1

根据从器件状态 (TWIn.SSTATUS) 寄存器中的读/写方向 (DIR) 位和总线条件，在接收地址数据包后会出现四种不同情况 (S1 至 S4) 之一。

图 25-6. TWI 从器件操作



情况 S1: 已接受地址包——方向位设为 0

如果在接收到地址包后从器件发送 ACK 信号，并且从器件状态 (TWIn.SSTATUS) 寄存器中的读/写方向 (DIR) 位设为 0，则主器件会指示写操作。

此时，时钟保持功能激活，强制将 SCL 设为低电平。该操作将延长时钟的低电平周期，从而降低整体时钟频率，强制生成处理数据所需的延时，并防止总线上发生进一步活动。

软件可准备:

- 从主器件读取接收的数据包

情况 S2: 已接受地址包——方向位设为 1

如果在接收到地址包后从器件发送 ACK 信号，并且 DIR 位置 1，主器件会指示读操作，从器件状态 (TWIn.SSTATUS) 寄存器中的数据中断标志 (DIF) 将置 1。

此时，时钟保持功能激活，强制将 SCL 设为低电平。

软件可准备：

- 向主器件发送数据包

情况 S3：收到停止条件

收到停止条件时，地址或停止（AP）标志将设为 0，以指示停止条件（而非地址匹配）激活了地址或停止中断标志（APIF）。

AP 和 APIF 标志位于从器件状态（TWIn.SSTATUS）寄存器中。

软件可准备：

- 等待直到对新的地址包寻址到该处

情况 S4：冲突

如果从器件无法发送高电平数据位或 NACK，则从器件状态（TWIn.SSTATUS）寄存器中的冲突（COLL）位置 1。从器件将正常开始运行，但低电平值不会移出到 SDA 上。将禁止来自从器件逻辑的数据和应答输出。时钟保持功能将被释放。将接受启动或重复启动条件。

COLL 位适用于采用地址解析协议（ARP）的系统。在非 ARP 情况下检测到的冲突表明存在协议违规，必须视为总线错误。

25.3.2.3.2. 接收数据包

假设出现上述的 S1 情况，从器件必须准备好接收数据。接收到数据包时，从器件状态（TWIn.SSTATUS）寄存器中的数据中断标志（DIF）设置为 1。向 TWIn.SCTRLB 寄存器中的命令（SCMD）位域写入命令时，通过从器件控制 B（TWIn.SCTRLB）寄存器中的应答动作（ACKACT）位选择的动作将在总线上自动发送。

软件可准备执行以下操作之一：

- 通过向 TWIn.SCTRLB 寄存器中的 ACKACT 位写入 0 以使用 ACK 进行响应，表示从器件已准备好接收更多数据
- 通过向 ACKACT 位写入 1 以使用 NACK 进行响应，表示从器件无法接收更多数据，主器件必须发出停止或重复启动条件

25.3.2.3.3. 发送数据包

假设出现上述的 S2 情况，从器件可通过对从器件数据（TWIn.SDATA）寄存器进行写操作来启动数据发送。完成数据包发送时，从器件状态（TWIn.SSTATUS）寄存器中的数据中断标志（DIF）设置为 1。

软件可准备执行以下操作之一：

- 通过读取来自从器件状态（TWIn.SSTATUS）寄存器的接收应答（RXACK）位来检查主器件是否通过 ACK 做出响应，并开始发送新的数据包
- 通过读取 RXACK 来检查主器件是否通过 NACK 做出响应，并停止发送数据包。发出 NACK 响应后，主器件必须发送停止或重复启动条件。

25.3.3. 其他特性

25.3.3.1. SMBus

如果在 SMBus 环境中使用 TWI，必须配置主器件控制 A（TWIn.MCTRLA）寄存器中的非活动总线超时（TIMEOUT）位域。建议在设置超时之前对主波特率（TWIn.MBAUD）寄存器执行写操作，因为超时取决于波特率设置。

SMBus 环境可以使用 100 kHz 的频率。对于标准模式（Sm）和快速模式（Fm），工作频率下的输出压摆率受限，增强型快速模式（Fm+）则具有 10 倍的输出驱动能力。

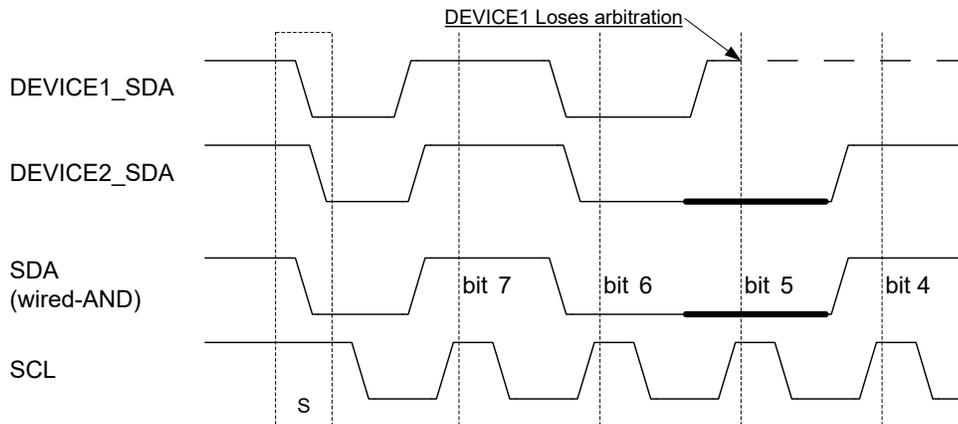
TWI 还支持在控制 A（TWIn.CTRLA）寄存器的 SDA 保持时间（SDAHOLD）位域中配置与 SMBus 兼容的 SDA 保持时间。

25.3.3.2. 多主机

只有在检测到总线处于空闲状态时，主器件才能启动总线事务。由于 TWI 总线为多主器件总线，因此可能发生多个器件尝试同时启动事务的情况。这会导致多个主器件同时拥有总线。TWI 使用仲裁方案来解决这一问题，这种方案可让无法在 SDA 上发送高电平数据位的主器件失去对总线的控制权，主器件状态（TWIn.MSTATUS）寄存器中的总线状态（BUSSTATE）位域将变为繁忙状态。仲裁失败的主器件必须等待至总线空闲之后，才能尝试重新获取总线所有权。

两个器件都能发出启动条件，但如果 DEVICE1 在 DEVICE2 发送低电平时尝试发送高电平（bit 5），则仲裁会失败。

图 25-7. TWI 仲裁



25.3.3.3. 智能模式

TWI 接口支持智能模式，可简化应用程序代码并最大限度地减少因遵守 I²C 协议而需要的用户交互。

对于 TWI 主器件，智能模式将在读取主器件数据（TWIn.MDATA）寄存器时立即自动发送 ACK 操作。该功能仅适用于主器件控制 B（TWIn.MCTRLB）寄存器中的应答动作（ACKACT）位设为 ACK 的情况。如果将 ACKACT 位设为 NACK，则在读取 MDATA 寄存器后，TWI 主器件不会生成 NACK。主器件控制 A（TWIn.MCTRLA）寄存器中的智能模式使能（SMEN）位置 1 时使能该功能。

对于 TWI 从器件，智能模式将在读取从器件数据（TWIn.SDATA）寄存器时立即自动发送 ACK 操作。如果对 TWIn.SDATA 寄存器进行读或写操作，智能模式会自动将从器件状态（TWIn.SSTATUS）寄存器中的数据中断标志（DIF）设为 0。从器件控制 A（TWIn.SCTRLA）寄存器中的智能模式使能（SMEN）位置 1 时使能该功能。

25.3.3.4. 快速命令模式

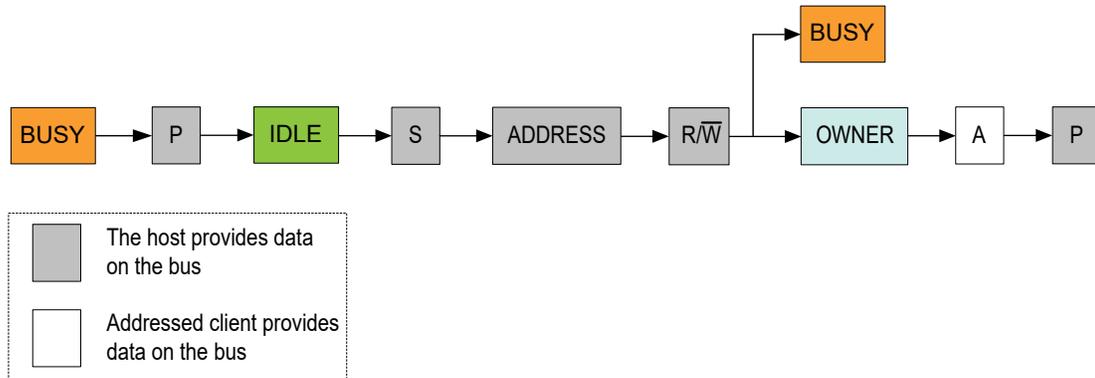
在快速命令模式下，地址包的 R/\bar{W} 位表示该命令。通过向主器件控制 A（TWIn.MCTRLA）寄存器中的快速命令使能（QCEN）位写入 1，可以使能该模式。该模式不会发送或接收数据。

快速命令模式是 SMBus 所特有的，其中 R/\bar{W} 位可用于开启/关闭器件功能，或使能/禁止低功耗待机模式。可使能该模式来自动触发操作并降低软件复杂度。

主器件收到来自从器件的 ACK 后，读中断标志（RIF）或写中断标志（WIF）将置 1，具体取决于 R/\bar{W} 位的值。在发出快速命令后 RIF 或 WIF 标志置 1 时，TWI 将通过对主器件控制 B（TWIn.MCTRLB）寄存器中的命令（MCMD）位域进行写操作来接受停止命令。

RIF 标志、WIF 标志连同最后一个接收应答（RXACK）标志的值均位于主器件状态（TWIn.MSTATUS）寄存器中。

图 25-8. 快速命令帧格式



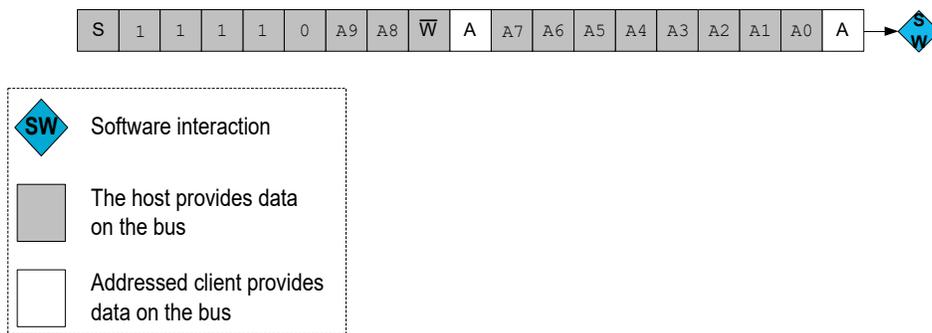
25.3.3.5. 10 位地址

无论事务是读还是写，主器件必须通过发送 $\overline{R/W}$ 方向位设为 0 的 10 位地址，来启动该事务。

从器件地址匹配逻辑支持识别 7 位地址和广播呼叫地址。从器件地址匹配逻辑使用从器件地址（TWIn.SADDR）寄存器来确定主器件是否已寻址 TWI 从器件。

TWI 从器件地址匹配逻辑仅支持识别 10 位地址的第一个字节，第二个字节必须由软件处理。如果从器件地址（TWIn.SADDR）寄存器的高五位为 0b11110，将识别 10 位地址的第一个字节。因此，第一个字节将包含五个指示位、10 位地址的两个最高有效位（MSb）以及 $\overline{R/W}$ 方向位，主器件将开始执行总线事务。沿袭自主器件的地址最低有效字节（LSB）将以数据包的形式出现。

图 25-9. 10 位地址发送



25.3.4. 中断

表 25-1. 可用中断向量和中断源

名称	向量说明	条件
客户端	TWI 从器件中断	<ul style="list-style-type: none"> DIF: TWIn.SSTATUS 中的数据中断标志设置为 1 APIF: TWIn.SSTATUS 中的地址或停止中断标志设置为 1
主器件	TWI 主器件中断	<ul style="list-style-type: none"> RIF: TWIn.MSTATUS 中的读中断标志设置为 1 WIF: TWIn.MSTATUS 中的写中断标志设置为 1

发生中断条件时，主器件状态（TWIn.MSTATUS）寄存器或从器件状态（TWIn.SSTATUS）寄存器中的相应中断标志将置 1。

如果中断向量支持多个中断请求条件，将对各个中断请求执行逻辑或运算，将合并生成的一个组合中断请求发送到中断控制器。用户必须读取 TWIn.MSTATUS 寄存器或 TWIn.SSTATUS 寄存器的中断标志以确定存在哪些中断条件。

25.3.5. 休眠模式操作

总线状态逻辑和地址识别硬件在所有休眠模式下都继续工作。如果从器件处于休眠模式且检测到启动条件后跟从器件的地址，则在唤醒期间激活时钟延长，直到主时钟可用为止。在所有休眠模式下，主器件都将停止工作。

25.3.6. 调试操作

在运行时调试期间，TWI 将继续正常工作。如果在调试模式下暂停 CPU，TWI 也将暂停正常工作。可通过向调试控制（TWIn.DBGCTRL）寄存器中的调试运行（DBGRUN）位写入 1，强制 TWI 在 CPU 暂停的情况下工作。当 CPU 在调试模式下暂停并且 DBGRUN 位为 1 时，读取或写入主数据（TWIn.MDATA）寄存器或从数据（TWIn.SDATA）寄存器既不会触发总线操作，也不会导致发送和清零标志。如果 TWI 配置为需要由 CPU 通过中断或类似操作进行定期控制，则在调试期间暂停 CPU 可能会产生不正确的操作或丢失数据。

25.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0	
0x00	CTRLA	7:0				SDASETUP	SDAHOLD[1:0]		FMPEN		
0x01	保留										
0x02	DBGCTRL	7:0								DBGRUN	
0x03	MCTRLA	7:0	RIEN	WIEN		QCEN	TIMEOUT[1:0]		SMEN	ENABLE	
0x04	MCTRLB	7:0					FLUSH	ACKACT	MCMMD[1:0]		
0x05	MSTATUS	7:0	RIF	WIF	CLKHOLD	RXACK	ARBLOST	BUSERR	BUSSTATE[1:0]		
0x06	MBAUD	7:0	BAUD[7:0]								
0x07	MADDR	7:0	ADDR[7:0]								
0x08	MDATA	7:0	DATA[7:0]								
0x09	SCTRLA	7:0	DIEN	APIEN	PIEN			PMEN	SMEN	ENABLE	
0x0A	SCTRLB	7:0						ACKACT	SCMD[1:0]		
0x0B	SSTATUS	7:0	DIF	APIF	CLKHOLD	RXACK	COLL	BUSERR	DIR	AP	
0x0C	SADDR	7:0	ADDR[7:0]								
0x0D	SDATA	7:0	DATA[7:0]								
0x0E	SADDRMASK	7:0	ADDRMASK[6:0]								ADDREN

25.5. 寄存器说明

25.5.1. 控制 A

名称: CTRLA
 偏移量: 0x00
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
				SDASETUP	SDAHOLD[1:0]		FMPEN	
访问				R/W	R/W	R/W	R/W	
复位				0	0	0	0	

Bit 4 – SDASETUP SDA 建立时间

该位用于在 TWI 从器件模式下选择时钟保持时间并确保 SDA 输出信号的建立时间最短。

值	名称	说明
0	4CYC	SDA 建立时间为四个时钟周期
1	8CYC	SDA 建立时间为八个时钟周期

Bit 3:2 – SDAHOLD[1:0] SDA 保持时间

该位域用于为 TWI 选择 SDA 保持时间。有关详细信息，请参见 *电气特性* 一章。

值	名称	说明
0x0	OFF	保持时间关闭
0x1	50NS	短保持时间
0x2	300NS	在典型条件下符合 SMBus 2.0 规范
0x3	500NS	在所有条件下均符合 SMBus 2.0 规范

Bit 1 – FMPEN FM+使能

向该位写入 1 可为 TWI（默认配置下）选择 1 MHz 总线速度。

值	名称	说明
0	OFF	在标准模式或快速模式下工作
1	ON	在增强型快速模式下工作

25.5.2. 调试控制

名称: DBGCTRL
 偏移量: 0x02
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
								DBGRUN
访问								R/W
复位								0

Bit 0 - DBGRUN 调试运行

有关详细信息，请参见 *调试操作* 一节。

值	说明
0	TWI 在中断调试模式下暂停并忽略事件
1	当 CPU 暂停时，TWI 将继续在中断调试模式下运行

25.5.3. 主器件控制 A

名称: MCTRLA
 偏移量: 0x03
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	RIEN	WIEN		QCEN	TIMEOUT[1:0]		SMEN	ENABLE
访问	R/W	R/W		R/W	R/W	R/W	R/W	R/W
复位	0	0		0	0	0	0	0

Bit 7 - RIEN 读中断允许

仅当该位和状态（CPU.SREG）寄存器中的全局中断允许（I）位置 1 时，才会产生 TWI 主器件读中断。向该位写入 1 可允许与主器件状态（TWIn.MSTATUS）寄存器中的读中断标志（RIF）相关的中断。发生主器件读中断时，RIF 标志置 1。

Bit 6 - WIEN 写中断允许

仅当该位和状态（CPU.SREG）寄存器中的全局中断允许（I）位置 1 时，才会产生 TWI 主器件写中断。向该位写入 1 可允许与主器件状态（TWIn.MSTATUS）寄存器中的写中断标志（WIF）相关的中断。发生主器件写中断时，WIF 标志置 1。

Bit 4 - QCEN 快速命令使能

向该位写入 1 可启用快速命令模式。如果使能快速命令模式并且从器件确认了地址，相应的读中断标志（RIF）或写中断标志（WIF）将置 1，具体取决于 R/W 位的值。软件必须通过对主器件控制 B（TWIn.MCTRLB）寄存器的命令（MCMD）位域进行写操作来发出停止命令。

Bit 3:2 - TIMEOUT[1:0] 非活动总线超时

将该位域设置为非零值将使能非活动总线超时监控器。如果总线处于非活动状态的时间超过 TIMEOUT 设置，则总线状态逻辑将进入空闲状态。

值	名称	说明
0x0	DISABLED	禁止总线超时——I ² C
0x1	50US	50 μs——SMBus（假设波特率设置为 100 kHz）
0x2	100US	100 μs（假设波特率设置为 100 kHz）
0x3	200US	200 μs（假设波特率设置为 100 kHz）

Bit 1 - SMEN 智能模式使能

向该位写入 1 可启用主器件智能模式。启用智能模式后，将在读取主器件数据（TWIn.MDATA）寄存器后立即发送主器件控制 B（TWIn.MCTRLB）寄存器中应答动作（ACKACT）位的现有值。

Bit 0 - ENABLE 使能 TWI 主器件

向该位写入 1 可启用 TWI 作为主器件。

25.5.4. 主器件控制 B

名称: MCTRLB
 偏移量: 0x04
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
					FLUSH	ACKACT	MCMD[1:0]	
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3 – FLUSH 刷新

该位用于清除主器件的内部状态，总线状态变为空闲。如果先对主器件数据（TWIn.MDATA）寄存器进行写操作，再对主器件地址（TWIn.MADDR）寄存器进行写操作，TWI 将发送无效数据。如果刷新后写入主器件地址（TWIn.MADDR）和主器件数据（TWIn.MDATA），则只要硬件检测到 SCL 总线空闲，事务就会开始。

向该位写入 1 会生成持续一个时钟周期的选通信号，以便先禁止主器件，然后再重新使能主器件。向该位写入 0 没有任何作用。

Bit 2 – ACKACT 应答动作

ACKACT⁽¹⁾位表示在总线状态和软件交互所定义的特定条件下主模式的行为。如果主器件控制 A（TWIn.MCTRLA）寄存器中的智能模式使能（SMEN）位置 1，则在读取主器件数据（TWIn.MDATA）寄存器时将执行应答动作，否则必须向主器件控制 B（TWIn.MCTRLB）寄存器中的命令（MCMD）位域写入一个命令。

由于主器件正在发送数据，对主器件数据（TWIn.MDATA）寄存器进行写操作时不会执行应答动作。

值	名称	说明
0	ACK	发送 ACK
1	NACK	发送 NACK

Bit 1:0 – MCMD[1:0] 命令

MCMD⁽¹⁾位域为选通位域。该位域始终读为 0。

写入该位域会触发主器件操作，如下表所定义。

表 25-2. 命令设置

MCMD[1:0]	组配置	DIR	说明
0x0	NOACT	X	保留
0x1	REPSTART	X	执行应答动作，接着发出重复启动条件
0x2	RECVTRANS	\bar{W}	执行应答动作（无动作），接着执行字节写操作 ⁽²⁾
		R	执行应答动作，接着执行字节读操作
0x3	STOP	X	执行应答动作，接着发出停止条件

注:

1. 可以同时 ACKACT 位和 MCMD 位域进行写操作。
2. 对于主器件写操作，TWI 将等待新数据写入主器件数据（TWIn.MDATA）寄存器。

25.5.5. 主器件状态

名称: MSTATUS
 偏移量: 0x05
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	RIF	WIF	CLKHOLD	RXACK	ARBLOST	BUSERR	BUSSTATE[1:0]	
访问	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7 - RIF 读中断标志

当主器件字节读取操作完成时，该标志置 1。

RIF 标志可用于主器件读中断。更多信息，请参见主器件控制 A (TWIn.MCTRLA) 寄存器中的读中断允许 (RIEN) 位。

当访问其他几个 TWI 寄存器时，该标志会自动清零。选择以下方法之一可将 RIF 标志清零：

1. 向其写入 1。
2. 写入主器件地址 (TWIn.MADDR) 寄存器。
3. 读/写主器件数据 (TWIn.MDATA) 寄存器。
4. 对主器件控制 B (TWIn.MCTRLB) 寄存器中的命令 (MCMD) 位域执行写操作。

Bit 6 - WIF 写中断标志

该标志在主器件发送地址或字节写操作完成时置 1，与是否发生总线错误或仲裁失败无关。

WIF 标志可用于主器件写中断。更多信息，请参见主器件控制 A (TWIn.MCTRLA) 寄存器中的写中断允许 (WIEN) 位。

选择针对 RIF 标志的下述方法之一可将该标志清零。

Bit 5 - CLKHOLD 时钟保持

如果该位读为 1，则指示主器件当前正在将 SCL 保持为低电平，从而延长 TWI 时钟周期。

选择针对 RIF 标志的下述方法之一可将该位清零。

Bit 4 - RXACK 接收的应答

当该标志读为 0 时，表示从从器件接收的最新应答位为 ACK，从器件已准备好接收更多数据。

当该标志读为 1 时，表示从从器件接收的最新应答位为 NACK，从器件无法或不需接收更多数据。

Bit 3 - ARBLOST 仲裁失败

如果该位读为 1，则表示主器件仲裁失败。可能发生在以下某种情况下：

1. 发送高电平数据位时。
2. 发送 NACK 位时。
3. 发出启动条件 (S) 时。
4. 发出重复启动条件 (Sr) 时。

选择针对 RIF 标志的下述方法之一可将该标志清零。

Bit 2 – BUSERR 总线错误

BUSERR 标志表示出现了非法总线操作。如果在 TWI 总线上检测到违反协议的启动条件 (S)、重复启动条件 (Sr) 或停止条件 (P)，就代表检测到非法总线操作。比如，在启动条件后紧跟停止条件就会违反协议。

选择以下方法之一可将 BUSERR 标志清零：

1. 向其写入 1。
2. 写入主器件地址 (TWIn.MADDR) 寄存器。

TWI 总线错误检测器是 TWI 主电路的一部分。要检测总线错误，必须使能 TWI 主模式 (TWIn.MCTRLA 中的 ENABLE 位为 1)，并且主时钟频率必须至少是 SCL 频率的四倍。

Bit 1:0 – BUSSTATE[1:0] 总线状态

该位域指示当前 TWI 总线状态。向该位域写入 0x1 将强制总线状态变为空闲。所有其他值都将被忽略。

值	名称	说明
0x0	UNKNOWN	未知总线状态
0x1	IDLE	空闲总线状态
0x2	OWNER	该 TWI 控制总线
0x3	BUSY	繁忙总线状态

25.5.6. 主器件波特率

名称: MBAUD
 偏移量: 0x06
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	BAUD[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - BAUD[7:0] 波特率

利用该位域可以得到 SCL 高电平和低电平时间，禁止主器件时必须对该位域进行写操作。通过向主器件控制 A (TWIn.MCTRLA) 寄存器中的使能 TWI 主器件 (ENABLE) 位写入 0，可以禁止主器件。

有关如何计算 SCL 频率的更多信息，请参见*时钟生成*一节。

25.5.7. 主机地址

名称: MADDR
 偏移量: 0x07
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	ADDR[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - ADDR[7:0] 地址

该寄存器包含外部从器件的地址。写入该位域时，TWI 将发出启动条件，移位寄存器根据总线状态在总线上执行字节发送操作。

由于读访问不会导致主逻辑执行任何与总线协议相关的操作，因此可随时读取该寄存器，且不会干扰正在进行的总线活动。

主器件控制逻辑使用该寄存器的 bit 0 作为 R/\overline{W} 方向位。

25.5.8. 主器件数据

名称: MDATA
 偏移量: 0x08
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	DATA[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - DATA[7:0] 数据

该位域支持直接访问主器件的物理移位寄存器，该寄存器用于将数据移出到总线（发送）以及移入从总线接收的数据（接收）。直接访问意味着在字节传送期间无法访问 MDATA 寄存器。

只有 CLKHOLD 位读为 1 或发生中断时，才能成功读取有效数据或写入待发送数据。

对 MDATA 寄存器进行写访问时，将命令主器件在总线上执行字节发送操作，紧接着从从器件接收应答位。该过程与主器件控制 B (TWIn.MCTRLB) 寄存器中的应答动作 (ACKACT) 位无关。无论仲裁成功还是失败，都会在写中断标志 (WIF) 置 1 前执行写操作。

如果主器件控制 A (TWIn.MCTRLA) 寄存器中的智能模式使能 (SMEN) 位置 1，对 MDATA 寄存器进行读访问将命令主器件执行应答动作，具体取决于主器件控制 B (TWIn.MCTRLB) 寄存器中的应答动作 (ACKACT) 位的设置。

注:

1. 如果在 ACKACT 置 1 时读取 MDATA 寄存器，WIF 和 RIF 标志将自动清零。
2. ARBLOST 和 BUSEER 标志保持不变。
3. WIF、RIF、ARBLOST、BUSERR 标志连同时钟保持 (CLKHOLD) 位均位于主器件状态 (TWIn.MSTATUS) 寄存器中。

25.5.9. 从器件控制 A

名称: SCTRLA
 偏移量: 0x09
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	DIEN	APIEN	PIEN			PMEN	SMEN	ENABLE
访问	R/W	R/W	R/W			R/W	R/W	R/W
复位	0	0	0			0	0	0

Bit 7 - DIEN 数据中断允许

向该位写入 1 可允许与从器件状态 (TWIn.SSTATUS) 寄存器中的数据中断标志 (DIF) 相关的中断。仅当该位、DIF 标志和状态 (CPU.SREG) 寄存器中的全局中断允许 (I) 位均置 1 时, 才会产生 TWI 从器件数据中断。

Bit 6 - APIEN 地址或停止中断允许

向该位写入 1 可允许与从器件状态 (TWIn.SSTATUS) 寄存器中的地址或停止中断标志 (APIF) 相关的中断。仅当该位、APIF 标志和状态 (CPU.SREG) 寄存器中的全局中断允许 (I) 位均置 1 时, 才会产生 TWI 从器件地址或停止中断。

注:

1. 从器件停止中断与从器件地址中断共用中断标志和向量。
2. 必须向从器件控制 A (TWIn.SCTRLA) 寄存器中的停止中断允许 (PIEN) 位写入 1 才能在出现停止条件时将 APIF 置 1。
3. 当中断发生时, 从器件状态 (TWIn.SSTATUS) 寄存器中的地址或停止 (AP) 位将确定中断原因是地址匹配还是停止条件。

Bit 5 - PIEN 停止中断允许

向该位写入 1 可在出现停止条件时将从器件状态 (TWIn.SSTATUS) 寄存器中的地址或停止中断标志 (APIF) 置 1。要使用该功能, 主时钟频率必须至少是 SCL 频率的 4 倍。

Bit 2 - PMEN 地址识别模式

如果向该位写入 1, 则从器件地址匹配逻辑将响应所有接收地址。

如果向该位写入 0, 则地址匹配逻辑将使用从器件地址 (TWIn.SADDR) 寄存器来确定将哪个地址识别为从器件的地址。

Bit 1 - SMEN 智能模式使能

向该位写入 1 可使能从器件智能模式。使能智能模式时, 可以通过对从器件控制 B (TWIn.SCTRLB) 寄存器的命令 (SCMD) 位域进行写操作而发出命令, 或者访问从器件数据 (TWIn.SDATA) 寄存器将中断复位, 操作将继续。如果禁止智能模式, 则从器件将始终等待新的从器件命令, 收到该命令后再继续操作。

Bit 0 - ENABLE 使能 TWI 从器件

向该位写入 1 可使能 TWI 从器件。

25.5.10. 从器件控制 B

名称: SCTRLB
 偏移量: 0x0A
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
						ACKACT	SCMD[1:0]	
访问						R/W	R/W	R/W
复位						0	0	0

Bit 2 - ACKACT 应答动作

ACKACT⁽¹⁾位表示在总线协议状态和软件交互所定义的特定条件下从器件的行为。如果从器件控制 A (TWIn.SCTRLA) 寄存器中的智能模式使能 (SMEN) 位置 1, 则在读取从器件数据 (TWIn.SDATA) 寄存器时执行应答动作, 否则必须向从器件控制 B (TWIn.SCTRLB) 寄存器中的命令 (SCMD) 位域写入命令。

由于从器件正在发送数据, 因此对从器件数据 (TWIn.SDATA) 寄存器进行写操作时不会执行应答动作。

值	名称	说明
0	ACK	发送 ACK
1	NACK	发送 NACK

Bit 1:0 - SCMD[1:0] 命令

SCMD⁽¹⁾位域为选通位域。该位域始终读为 0。

写入该位域会触发从器件操作, 如下表所定义。

表 25-3. 命令设置

值	名称	DIR	说明
0x0	NOACT	X	无动作
0x1	—	X	保留
0x2	COMPTRANS	W	执行应答动作, 接着等待启动 (S/Sr) 条件
		R	等待启动 (S/Sr) 条件
0x3	RESPONSE	W	执行应答动作, 接着接收下一个字节
		R	用于响应地址中断 (APIF): 执行应答动作, 接着执行从器件数据中断。 用于响应数据中断 (DIF): 执行字节读操作, 接着执行应答动作。

注: 1. 可以同时写入 ACKACT 位和 SCMD 位域。ACKACT 将在触发命令前更新。

25.5.11. 从器件状态

名称: SSTATUS
 偏移量: 0x0B
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	DIF	APIF	CLKHOLD	RXACK	COLL	BUSERR	DIR	AP
访问	R/W	R/W	R	R	R/W	R/W	R	R
复位	0	0	0	0	0	0	0	0

Bit 7 - DIF 数据中断标志

当从器件字节发送或接收操作完成且没有任何总线错误时，该标志置 1。如果冲突检测时发现事务未成功完成，则该标志置 1。更多信息，可参见冲突（COLL）位说明。

DIF 标志可用于生成从器件数据中断。更多信息，可参见从器件控制 A（TWIn.SCTRLA）寄存器中的数据中断允许（DIEN）位。

当访问其他几个 TWI 寄存器时，该标志会自动清零。可选择以下方法之一来清零 DIF 标志：

1. 写入/读取从器件数据（TWIn.SDATA）寄存器。
2. 写入从器件控制 B（TWIn.SCTRLB）寄存器中的命令（SCMD）位域。

Bit 6 - APIF 地址或停止中断标志

收到从器件地址或出现停止条件时，该标志置 1。

APIF 标志可用于生成从器件地址或停止中断。更多信息，可参见从器件控制 A（TWIn.SCTRLA）寄存器中的地址或停止中断允许（APIEN）位。

该标志的清零方法与 DIF 标志相同。

Bit 5 - CLKHOLD 时钟保持

当该位读为 1 时，表示从器件当前正在使 SCL 保持低电平，延长 TWI 时钟周期。

当发生地址或数据中断时，该位置 1。复位相应的中断将间接地清零该位。

Bit 4 - RXACK 接收的应答

当该标志读为 0 时，表示来自主器件的最新应答位是 ACK。

当该标志读为 1 时，表示来自主器件的最新应答位是 NACK。

Bit 3 - COLL 冲突

当该位读为 1 时，表示从器件尚无法执行以下操作之一：

1. 在 SDA 上发送高位。数据中断标志（DIF）将在内部完成不成功的事务后置 1。
2. 发送 NACK 位。由于发生从器件地址匹配并导致 APIF 标志置 1，因此发生了冲突。

向该位写入 1 会将 COLL 标志清零。如果检测到任何启动条件（S/Sr），该标志将自动清零。

注： APIF 和 DIF 标志生成的中断只供处理程序用于检查冲突。

Bit 2 - BUSERR 总线错误

BUSERR 标志表示出现了非法总线条件。如果在 TWI 总线上检测到违反协议的启动条件（S）、重复启动条件（Sr）或停止条件（P），就代表检测到非法总线操作。比如，在启动条件后紧跟停止条件就会违反协议。

向该位写入 1 会将 BUSERR 标志清零。

TWI 总线错误检测器是 TWI 主电路的一部分。为了使从器件能够检测到总线错误，必须使能 TWI 主器件，并且主时钟频率必须至少是 SCL 频率的四倍。通过向 TWIn.MCTRLA 寄存器中的 ENABLE 位写入 1 可使能 TWI 主器件。

Bit 1 - DIR 读/写方向

该位用于指示当前的 TWI 总线方向。DIR 位反映从 TWI 主器件接收的最后一个地址数据包中的方向位值。

当该位读为 1 时，表示主器件正在进行读操作。

当该位读为 0 时，表示主器件正在进行写操作。

Bit 0 - AP 地址或停止

当 TWI 从器件地址或停止中断标志 (APIF) 置 1 时，该位用于确定中断是由地址检测还是停止条件引起。

值	名称	说明
0	STOP	停止条件产生的中断触发了 APIF 标志
1	ADR	地址检测产生的中断触发了 APIF 标志

25.5.12. 从器件地址

名称: SADDR
 偏移量: 0x0C
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	ADDR[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - ADDR[7:0] 地址

从器件地址匹配逻辑使用从器件地址 (TWIn.SADDR) 寄存器来确定主器件是否已寻址 TWI 从器件。如果接收到地址包, 从器件状态 (TWIn.SSTATUS) 寄存器中的地址或停止中断标志 (APIF) 和地址或停止 (AP) 位置 1。

TWIn.SADDR 寄存器的高 7 位 (ADDR[7:1]) 代表主从器件地址。

TWIn.SADDR 寄存器的最低有效位 (ADDR[0]) 用于识别 I²C 协议的广播呼叫地址 (0x00)。该位置 1 时使能该功能。

25.5.13. 从器件数据

名称: SDATA
 偏移量: 0x0D
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	DATA[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - DATA[7:0] 数据

可通过该位域访问从器件数据寄存器。

只有在 SCL 由从器件保持为低电平（即从器件 CLKHOLD 位置 1）时，才能读取有效数据或写入待发送数据。但如果软件一直通过使用中断或观察中断标志的方式来跟踪当前协议状态，则在访问 SDATA 寄存器前无需通过软件检查从器件状态（TWIn.SSTATUS）寄存器中的时钟保持（CLKHOLD）位。

如果从器件控制 A（TWIn.SCTRLA）寄存器中的智能模式使能（SMEN）位置 1，则当时钟保持功能激活时，对 SDATA 寄存器进行读访问将自动触发总线操作，并命令从器件执行应答动作，具体取决于从器件控制 B（TWIn.SCTRLB）寄存器中的应答动作（ACKACT）位的设置。

25.5.14. 从器件地址掩码

名称: SADDRMASK
 偏移量: 0x0E
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	ADDRMASK[6:0]							ADDREN
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:1 - ADDRMASK[6:0] 地址掩码

ADDRMASK 位域用作第二个地址匹配或地址掩码寄存器，具体取决于 ADDREN 位。

如果向 ADDREN 位写入 0，则 ADDRMASK 位域可以装入 7 位从器件地址掩码。从器件地址掩码

(TWIn.SADDRMASK) 寄存器中的每个位都可以屏蔽（禁止）TWI 从器件地址 (TWIn.SADDR) 寄存器中的相应地址位。如果向一个掩码位写入 1，则地址匹配逻辑将忽略传入地址位与从器件地址

(TWIn.SADDR) 寄存器中相应位之间的比较。换言之，被掩码的位将始终匹配，从而可以识别地址值范围。

如果向 ADDREN 位写入 1，则除了从器件地址 (TWIn.SADDR) 寄存器外，从器件地址掩码

(TWIn.SADDRMASK) 寄存器也可以装入第二个从器件地址。在该模式下，从器件将有两个惟一的地址，一个位于从器件地址 (TWIn.SADDR) 寄存器中，另一个位于从器件地址掩码 (TWIn.SADDRMASK) 寄存器中。

Bit 0 - ADDREN 地址掩码使能

如果向该位写入 0，则 TWIn.SADDRMASK 寄存器用作 TWIn.SADDR 寄存器的屏蔽寄存器。

如果向该位写入 1，则从器件地址匹配逻辑将响应从器件 TWIn.SADDR 和 TWIn.SADDRMASK 寄存器中的两个惟一地址。

26. CRCSCAN——循环冗余校验存储器扫描

26.1. 特性

- CRC-16-CCITT
- 检查整个闪存段、应用程序代码和/或引导段
- 可选择在出现故障时触发 NMI
- 可在内部复位初始化期间进行用户可配置校验

26.2. 概述

循环冗余校验（Cyclic Redundancy Check, CRC）从 NVM（整个闪存、仅引导段，或者引导段和应用程序代码段）获取字节形式的数据流，并生成校验和。CRC 外设（CRCSCAN）可用于检测程序存储器中的错误。

在要校验的段中，最后一个存储单元必须包含预先计算好用于比较的正确 16 位校验和。如果 CRCSCAN 计算出的校验和与预先计算的校验和匹配，则状态位置 1。如果不匹配，状态寄存器（CRCSCAN.STATUS）将指示匹配失败。用户可选择在校验和不匹配时让 CRCSCAN 生成不可屏蔽中断（NMI）。

应用于任意长度数据块的 n 位 CRC 将检测长度达 n 位的单一变化（错误突发）。对于更长的错误突发，将检测到小数 $1-2^{-n}$ 。

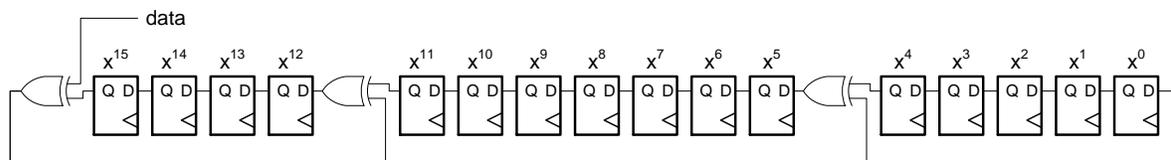
CRC 生成器支持 CRC-16-CCITT。

多项式：

- CRC-16-CCITT: $x^{16} + x^{12} + x^5 + 1$

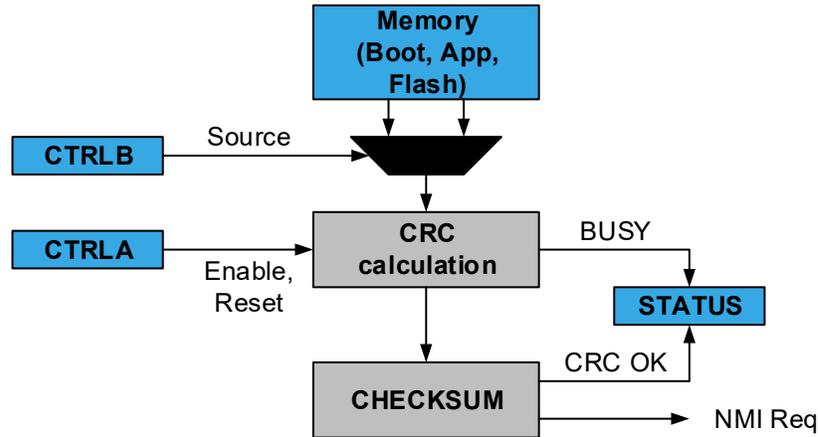
CRC 从字节 0 开始逐字节读取已设为待校验段的内容，并针对每个字节生成一个新的校验和。字节按照下图所示进行发送，从最高有效位开始。如果段中的最后几个字节包含正确的校验和，则 CRC 成功。有关如何放置校验和的信息，请参见[校验和](#)。校验和寄存器的初始值为 0xFFFF。

图 26-1. CRC 实现说明



26.2.1. 框图

图 26-2. 循环冗余校验框图



26.3. 功能说明

26.3.1. 初始化

要在软件中（或通过调试器）使能 CRC，请按照以下步骤操作：

1. 写入控制 B 寄存器（CRCSCAN.CTRLB）的源（SRC）位域，选择所需模式和源设置。
2. 向控制 A 寄存器（CRCSCAN.CTRLA）中的 ENABLE 位写入 1，使能 CRCSCAN。
3. CRC 将在三个周期后启动。CPU 将在这三个周期内继续执行。

CRCSCAN 可配置为在器件退出复位模式之前执行代码存储器扫描。如果该校验失败，将不允许 CPU 开始执行正常代码。该功能由 FUSE.SYSCFG0 中的 CRCSRC 位域使能和控制，更多信息请参见熔丝一章。

如果使能该功能，CRC 校验成功时将得到以下结果：

- 开始执行正常代码
- CRCSCAN.CTRLA 中的 ENABLE 位将为 1
- CRCSCAN.CTRLB 中的 SRC 位域将反映已校验的段
- CRCSCAN.STATUS 中的 OK 标志将为 1

如果使能该功能，CRC 校验失败时将得到以下结果：

- 不会开始执行正常代码，CPU 将挂起，不执行任何代码
- CRCSCAN.CTRLA 中的 ENABLE 位将为 1
- CRCSCAN.CTRLB 中的 SRC 位域将反映已校验的段
- CRCSCAN.STATUS 中的 OK 标志将为 0
- 可使用调试接口观察该条件

26.3.2. 工作原理

当 CRC 在优先模式下工作时，CRC 外设具有对闪存的优先访问权限，在访问完成之前将使 CPU 停止运行。

在优先模式下，CRC 每三个主时钟周期获取一个新字（16 位），或者在 CRC 外设配置为从启动开始扫描时获取一个新字（16 位）。

26.3.2.1. 校验和

预先计算的校验和必须位于待校验段的最后一个存储单元中。如果要校验引导段，则必须将校验和保存在引导段的末尾字节中，这同样适用于应用程序段和整个闪存。表 26-1 说明了如何为不同的段存储校验和。此外，若要了解如何配置要校验的段及 BOOTEND 和 APPEND 熔丝，请参见 CRCSCAN.CTRLB 寄存器说明和器件熔丝说明。

表 26-1. 预先计算的校验和在闪存中的位置

要校验的段	CHECKSUM[15:8]	CHECKSUM[7:0]
BOOT	FUSE_BOOTEND*256-2	FUSE_BOOTEND*256-1
BOOT 和 APPLICATION	FUSE_APPEND*256-2	FUSE_APPEND*256-1
整个闪存	FLASHEND-1	FLASHEND

26.3.3. 中断

表 26-2. 可用中断向量和中断源

名称	向量说明	条件
NMI	不可屏蔽中断	CRC 故障

发生中断条件时，状态（CRCSCAN.STATUS）寄存器中的 OK 标志将清除为 0。

通过将 1 写入控制 A（CRCSCAN.CTRLA）寄存器中的相应允许（NMIEN）位来允许不可屏蔽中断（NMI），但只能通过系统复位来禁止该中断。当 CRCSCAN.STATUS 中的 OK 标志清零且 NMIEN 位为 1 时，将生成 NMI。NMI 请求在系统复位前保持有效，无法禁止。

即使未全局允许中断，也可以触发 NMI。

26.3.4. 休眠模式操作

CRCSCAN 在所有休眠模式下都会停止。在所有 CPU 休眠模式下，CRCSCAN 外设都会停止，并在 CPU 唤醒时恢复工作。

写入 CRCSCAN.CTRLA 中的 EN 位后，CRCSCAN 开始工作三个周期。在这三个周期内，可以进入休眠模式。在这种情况下：

1. 只有当 CPU 被唤醒时，CRCSCAN 才会启动。
2. CRCSCAN 完成后，任意中断处理程序将执行。

26.3.5. 调试操作

每当调试器对外设或存储单元执行读或写时，CRCSCAN 就会被禁止。

如果调试器访问器件时 CRCSCAN 处于繁忙状态，则当调试器访问内部寄存器或调试器连接断开时，CRCSCAN 将重新开始正在进行的操作。

如果 CRCSCAN 在调试器导致其禁止时处于繁忙状态，则状态（CRCSCAN.STATUS）寄存器中的 BUSY 位将读为 1，但只要调试器将 CRCSCAN 禁止，它就不会主动校验任何部分。调试器内部寄存器空间中存在同步 CRC 状态位，这些位可由调试器读取而无需禁止 CRCSCAN。读取调试器的内部 CRC 状态位将确保使能 CRCSCAN。

可以直接通过调试器对 CRCSCAN.STATUS 寄存器进行写操作：

- CRCSCAN.STATUS 中的 BUSY 位：
 - 向 BUSY 位写入 0 将停止正在进行的 CRC 操作（这样 CRCSCAN 就不会在调试器允许时重新启动其操作）。
 - 向 BUSY 位写入 1 将使 CRC 通过控制 B（CRCSCAN.CTRLB）寄存器中的设置启动单次校验，但直到调试器允许才会执行。

只要 CRCSCAN.STATUS 中的 BUSY 位为 1，就不能更改控制 A（CRCSCAN.CTRLA）寄存器中的 CRCSCAN.CTRLB 和不可屏蔽中断允许（NMIEN）位。

- CRCSCAN.STATUS 中的 OK 位：
 - 如果 CRCSCAN.CTRLA 中的 NMIEN 位为 1，则向 OK 位写入 0 会触发不可屏蔽中断（NMI）。如果已触发 NMI，则不允许对 CRCSCAN 执行写操作。
 - 如果当 CRCSCAN.STATUS 中的 BUSY 位为 0 时向 OK 位写入 1，则 OK 位将读为 1。

通过调试器向 CRCSCAN.CTRLA 和 CRCSCAN.CTRLB 中写入与通过 CPU 写入的处理方式相同。

26.4. 寄存器汇总——CRCSCAN

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RESET						NMIEN	ENABLE
0x01	CTRLB	7:0			MODE[1:0]				SRC[1:0]	
0x02	STATUS	7:0							OK	BUSY

26.5. 寄存器说明

26.5.1. 控制 A

名称: CTRLA
 偏移量: 0x00
 复位: 0x00
 属性: -

如果 NMI 已触发，则该寄存器不可写。

位	7	6	5	4	3	2	1	0
	RESET						NMIEN	ENABLE
访问	R/W						R/W	R/W
复位	0						0	0

Bit 7 - RESET 复位 CRCSCAN

向该位写入 1 将复位 CRCSCAN 外设。CRCSCAN 控制寄存器和状态寄存器（CRCSCAN.CTRLA、CRCSCAN.CTRLB 和 CRCSCAN.STATUS）将在向 RESET 位写入 1 后的一个时钟周期内清零。

如果 NMIEN 为 0，则当 CRCSCAN 繁忙（CRCSCAN.STATUS 中的 BUSY 位为 1）以及不忙（BUSY 位为 0）时，该位均可写，并将立即生效。

如果 NMIEN 为 1，则只有在 CRCSCAN 不忙（CRCSCAN.STATUS 中的 BUSY 位为 0）时，该位才可写。RESET 位是选通位。

Bit 1 - NMIEN 使能 NMI 触发

向该位写入 1 时，任何 CRC 故障都会触发 NMI。

该位只能通过系统复位来清零——对 RESET 位进行写操作无法清零。

只有在 CRCSCAN 不忙（CRCSCAN.STATUS 中的 BUSY 位为 0）时，才可向该位写入 1。

Bit 0 - ENABLE 使能 CRCSCAN

向该位写入 1 可使能采用当前设置的 CRCSCAN 外设。即使完成 CRC 校验后该位也将保持为 1，但再次向其写入 1 将开始新的校验。

向该位写入 0 将不起作用。

可将 CRCSCAN 配置为在 MCU 启动序列期间运行扫描，以便先验证闪存段再让 CPU 开始执行正常代码（见[初始化](#)一节）。如果使能该功能，则开始执行正常代码时，ENABLE 位将读为 1。

要明确 CRCSCAN 外设是否正忙于正在进行的校验，请轮询状态寄存器（CRCSCAN.STATUS）中的 BUSY 位。

26.5.2. 控制 B

名称: CTRLB
 偏移量: 0x01
 复位: 0x00
 属性: -

CRCSCAN.CTRLB 寄存器包含 CRC 的模式和源设置。当 CRC 繁忙或触发 NMI 时，该寄存器不可写。

位	7	6	5	4	3	2	1	0
			MODE[1:0]				SRC[1:0]	
访问			R/W	R/W			R/W	R/W
复位			0	0			0	0

Bit 5:4 – MODE[1:0] CRC 闪存访问模式

内部复位初始化期间可使能 CRC，以在 CPU 启动之前先验证闪存段（见器件数据手册熔丝说明）。如果在内部复位初始化期间使能 CRC，则开始执行正常代码时，MODE 位域将读为非零值。为确保在执行代码时 CRC 正常工作，再次向 MODE 位写入 0x0。

值	名称	说明
0x0	PRIORITY	CRC 模块运行一次优先访问闪存的校验。CPU 停止，直到 CRC 完成。
其他值	-	保留

Bit 1:0 – SRC[1:0] CRC 源

SRC 位域选择 CRC 模块应校验的闪存段。要设置该段的大小，请参见 *配置和用户熔丝 (FUSE)* 一节。

内部复位初始化期间可使能 CRC，以便在 CPU 启动之前先验证闪存段（见熔丝一章）。如果在内部复位初始化期间 CRC 使能，则会在开始执行正常代码时将 SRC 位域读为 FLASH、BOOTAPP 或 BOOT（具体取决于配置）。

值	名称	说明
0x0	闪存	对整个闪存（引导段、应用程序代码段和应用程序数据段）执行 CRC。
0x1	BOOTAPP	对闪存的引导段和应用程序代码段执行 CRC。
0x2	BOOT	对闪存的引导段执行 CRC。
0x3	-	保留。

26.5.3. 状态

名称: STATUS
 偏移量: 0x02
 复位: 0x02
 属性: -

位	7	6	5	4	3	2	1	0
							OK	BUSY
访问							R	R
复位							1	0

Bit 1 - OK CRC 正常

此位读为 1 时，表示前一个 CRC 成功完成。CRC 扫描运行前该位默认置 1。除非 BUSY 位为 0，否则该位无效。

Bit 0 - BUSY CRC 繁忙

该位读为 1 时，表示 CRCSCAN 繁忙。只要模块繁忙，对控制寄存器的访问就会受限。

27. CCL——可配置定制逻辑

27.1. 特性

- 适用于通用 PCB 设计的胶连逻辑
- 两个可编程查找表（LUT）
- 组合逻辑函数：任何逻辑表达式，最多三个输入的函数
- 定序器逻辑函数：
 - 门控 D 型触发器
 - JK 型触发器
 - 门控 D 型锁存器
 - RS 锁存器
- 可灵活选择 LUT 输入：
 - I/O
 - 事件
 - 后续 LUT 输出
 - 内部外设，例如：
 - 模拟比较器
 - 定时器/计数器
 - USART
 - SPI
- 由系统时钟或其他外设提供时钟
- 输出可连接至 I/O 引脚或事件系统
- 每个 LUT 输出端均可选配同步器、滤波器或边沿检测器

27.2. 概述

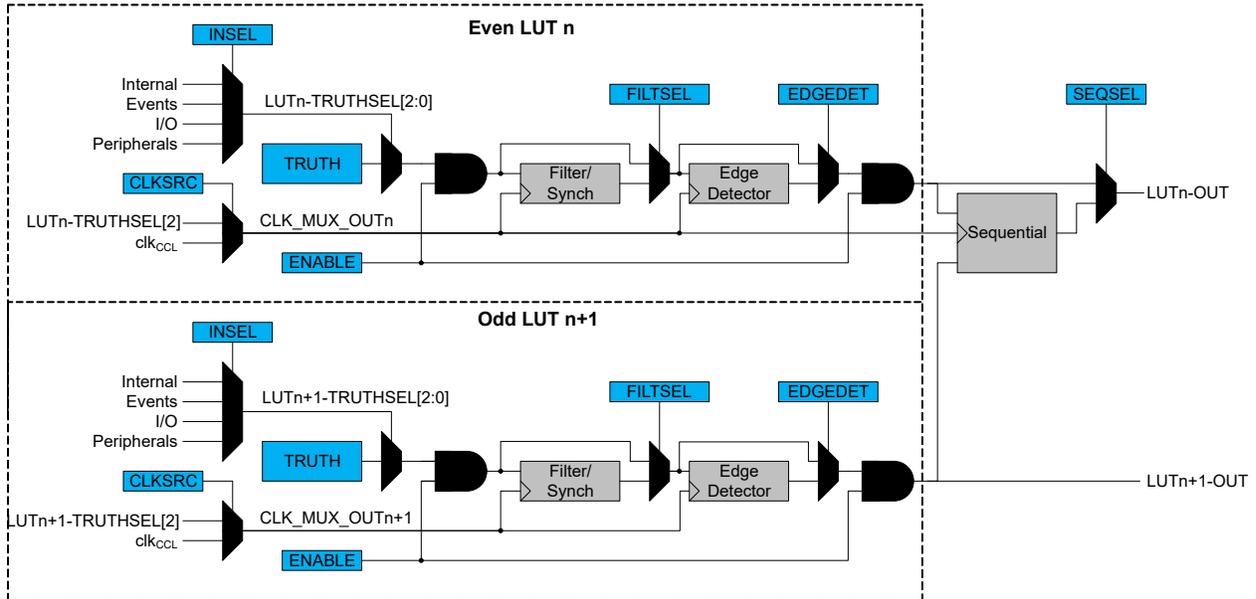
可配置定制逻辑（CCL）是一种可编程逻辑外设，可以连接到器件引脚、事件或其他内部外设。CCL 可以用作器件外设与外部器件之间的“胶连逻辑”。使用 CCL 不但可消除对外部逻辑元件的需求，而且还可组合使用多个独立于内核的外设来独立于 CPU 处理应用中的大多数时间关键部分，从而帮助设计人员克服实时约束。

CCL 外设提供多个 LUT。每个 LUT 由三个输入、一个真值表、一个同步器/滤波器和一个边沿检测器组成。每个 LUT 均可生成一个输出，作为用户可编程的逻辑表达式（具有三个输入）。三个输入均可单独屏蔽。

输出由各个输入使用组合逻辑生成，并且可进行滤波以消除尖峰。通过组合相邻的 LUT 来执行特定操作。通过使用定序器生成复杂波形。

27.2.1. 框图

图 27-1. CCL 框图



27.2.2. 信号说明

引脚名称	类型	说明
LUTn-OUT	数字输出	LUT 的输出
LUTn-IN[2:0]	数字输入	LUT 的输入

有关该外设引脚映射的详细信息，请参见 *I/O 复用和注意事项* 一章。可将一个信号映射到多个引脚。

27.2.3. 系统相关性

要使用此外设，必须按如下所述正确配置系统的其他部分。

表 27-1. CCL 系统相关性

相关性	适用	外设
时钟	是	CLKCTRL
I/O 线路和连接	是	PORT
中断	否	-
事件	是	EVSYS
调试	是	UPDI

27.2.3.1. 时钟

默认情况下，滤波器、边沿检测器和定序器由外设时钟（CLK_PER）驱动。此外，也可以使用其他时钟输入（CLK_MUX_OUTn）为这些模块提供时钟。这通过写入 LUT 控制 A 寄存器中的时钟源选择（CLKSRC）位域来配置。

当时钟源选择（CLKSRC）位写入 1 时，LUTn-TRUTHSEL[2] 用于为相应的滤波器和边沿检测器（CLK_MUX_OUTn）提供时钟。定序器由 LUT 对中的偶数 LUT 的 CLK_MUX_OUTn 提供时钟。当 CLKSRC 写入 1 时，LUTn-TRUTHSEL[2] 在真值表中视为关闭（低电平）。

更改时钟源时必须禁止 CCL 外设，从而避免该外设生成未定义的输出。

27.2.3.2. I/O 线

CCL 可通过 I/O 引脚获取输入并生成输出。为使其正常工作，必须将 I/O 引脚配置为供 LUT 使用。

27.2.3.3. 调试操作

当 CPU 在调试模式下停止时，CCL 将继续正常工作，但无法停止。如果 CCL 被配置为需要 CPU 定期对其进行处理，则调试期间可能会产生不正确的操作或丢失数据。

27.3. 功能说明

27.3.1. 初始化

LUT 和定序器的配置受使能保护，这意味着只能在禁止相应的偶数 LUT（LUT n 控制 A（CCL.LUTnCTRLA）寄存器中的 ENABLE = 0）时进行配置。这种机制可在（重新）配置过程中防止 CCL 产生不需要的输出。

以下位和寄存器受使能保护：

- 定序器控制 n（CCL.SEQCTRLn）寄存器中的定序器选择（SEQSEL）位
- LUT n 控制 x（CCL.LUTnCTRLx）寄存器，CCL.LUTnCTRLA 寄存器中的 ENABLE 位除外
- TRUTHn（CCL.TRUTHn）寄存器

CCL.LUTnCTRLx 寄存器中的使能保护位可在 CCL.LUTnCTRLA 寄存器中的 ENABLE 位写入 1 时写入，但不可在 ENABLE 位写入 0 时写入。

使能保护由寄存器说明中的使能保护属性表示。

27.3.2. 工作原理

27.3.2.1. 使能、禁止和复位

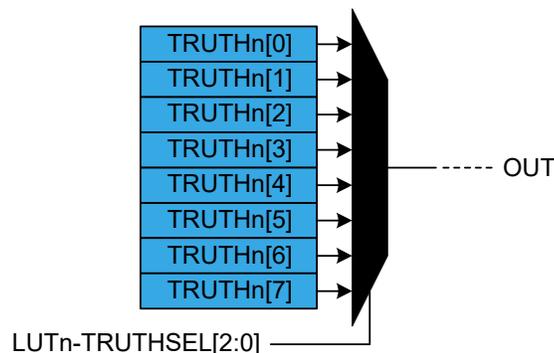
向控制 A（CCL.CTRLA）寄存器中的 ENABLE 位写入 1，使能 CCL。向该 ENABLE 位写入 0 可禁止 CCL。

各个 LUT 均通过向 CCL.LUTnCTRLA 寄存器中的 LUT 使能（ENABLE）位写入 1 来使能。各个 LUT 均通过向 CCL.LUTnCTRLA 寄存器中的 ENABLE 位写入 0 来禁止。

27.3.2.2. 真值表逻辑

每个 LUT 单元中的真值表均可生成一个组合逻辑输出，作为最多三个输入（LUTn-TRUTHSEL[2:0]）的函数。使用一个 LUT 可以实现任何 3 输入布尔逻辑功能。

图 27-2. LUT 真值表输出值选择



通过写入 LUT 控制寄存器中的输入源选择位域配置真值表输入(LUTn-TRUTHSEL[2:0])：

- CCL.LUTnCTRLB 中的 INSEL0
- CCL.LUTnCTRLB 中的 INSEL1

- CCL.LUTnCTRLC 中的 INSEL2

输入位（LUTn-TRUTHSEL[2:0]）的每种组合都对应于 CCL.TRUTHn 寄存器中的一个位，如下表所示：

表 27-2. LUT 真值表

LUTn-TRUTHSEL[2]	LUTn-TRUTHSEL[1]	LUTn-TRUTHSEL[0]	OUT
0	0	0	TRUTHn[0]
0	0	1	TRUTHn[1]
0	1	0	TRUTHn[2]
0	1	1	TRUTHn[3]
1	0	0	TRUTHn[4]
1	0	1	TRUTHn[5]
1	1	0	TRUTHn[6]
1	1	1	TRUTHn[7]



重要： 创建逻辑功能时，应将未使用的输入关闭（连接至低电平）。

例 27-1. CCL.TRUTHn = 0x42 时的 LUT 输出

当 CCL.TRUTHn 配置为 0x42 时，如果输入为 'b001 或 'b110，则 LUT 输出将为 1；如果输入为任何其他组合，则 LUT 输出将为 0。

27.3.2.3. 真值表输入选择

输入概述

输入可单独：

- 掩码
- 由外设驱动：
 - 模拟比较器（AC）输出
 - 定时器/计数器（TC）波形输出
- 由来自事件系统的内部事件驱动
- 由其他 CCL 子模块驱动

内部反馈输入（FEEDBACK）

选择该项（CCL.LUTnCTRLx 中的 INSELy = FEEDBACK）时，将定序器（SEQ）输出用作相应 LUT 的输入。

内部定序器的输出可用作 LUT 的输入源。有关 LUT0 和 LUT1 的示例，请参见下图。每个 LUT 的顺序选择均遵循以下公式：

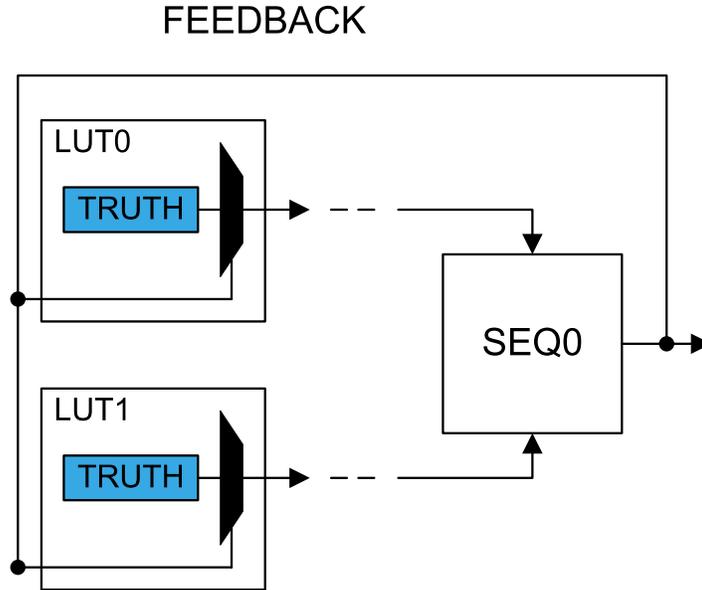
$$IN[2N][i] = SEQ[N]$$

$$IN[2N+1][i] = SEQ[N]$$

N 表示定序器编号； $i=0、1$ 或 2 ，表示 LUT 输入索引。

有关更多详细信息，请参见[定序器逻辑](#)。

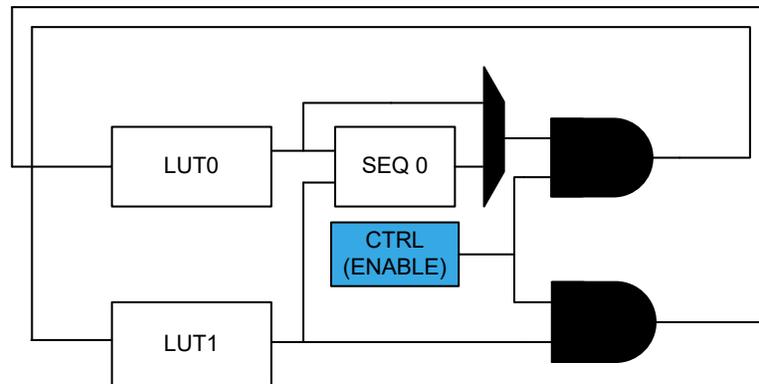
图 27-3. 反馈输入选择



链接 LUT (LINK)

选择 LINK 输入选项时，下一个 LUT 的直接输出用作 LUT 输入。通常，LUT[n+1]链接至 LUT[n]的输入。LUT0 链接至最后一个 LUT 的输入。

图 27-4. 链接 LUT 输入选择



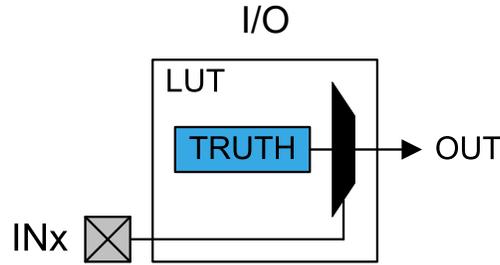
内部事件输入选择 (EVENT)

来自事件系统的异步事件可用作 LUT 的输入。事件输入线有两条 (EVENT0 和 EVENT1)，并可选作 LUT 输入。在选择 EVENT 输入选项之前配置事件系统。

I/O 引脚输入 (I/O)

选择 I/O 选项时，LUT 输入将连接到相应的 I/O 引脚。有关 LUTnIny 具体位置的更多详细信息，请参见 I/O 复用和注意事项一章。

图 27-5. I/O 引脚输入选择



外设

通过对 LUT n 控制 B 和 C 寄存器中的相应 LUT n 输入 y 位域进行写操作，可选择每个 LUT 的三条输入线上的不同外设：

- CCL.LUTnCTRLB 中的 INSEL0
- CCL.LUTnCTRLB 中的 INSEL1
- CCL.LUTnCTRLC 中的 INSEL2

27.3.2.4. 滤波器

默认情况下，LUT 输出是 LUT 输入的组合函数。当输入值发生更改时，这可能会导致出现一些短暂毛刺。如果应用需要，可通过滤波器随时钟消除这些毛刺。

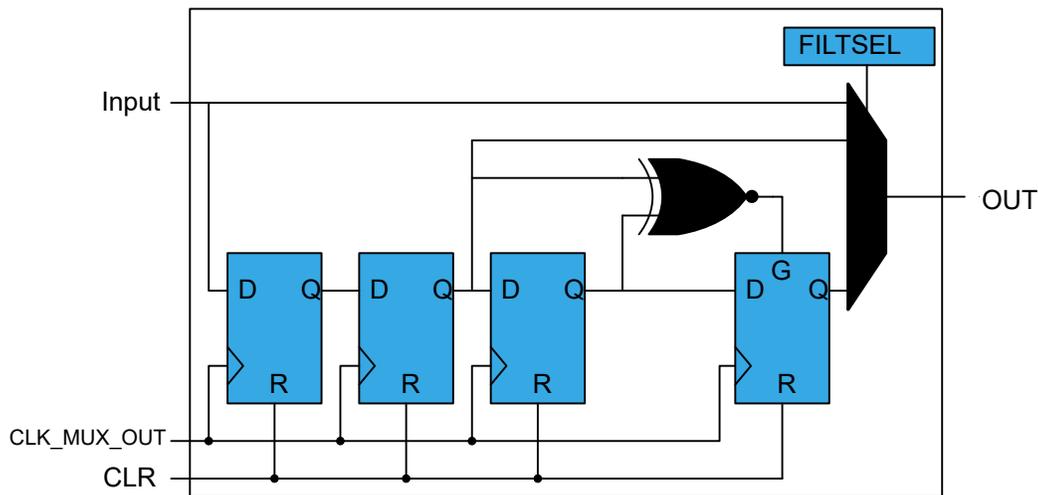
LUT n 控制 A (CCL.LUTnCTRLA) 寄存器中的滤波器选择 (FILTSEL) 位用于定义数字滤波器选项。

当 FILTSEL = SYNCH 时，输出与 CLK_MUX_OUTn 同步。输出将延迟两个 CLK_MUX_OUTn 正边沿。

当 FILTSEL = FILTER 时，只有持续时间超过两个 CLK_MUX_OUTn 正边沿的输入才会通过门控触发器传递到输出。输出将延迟四个 CLK_MUX_OUTn 正边沿。

禁止相应 LUT 后，再经过一个时钟周期之后会清零所有内部滤波器逻辑。

图 27-6. 滤波器



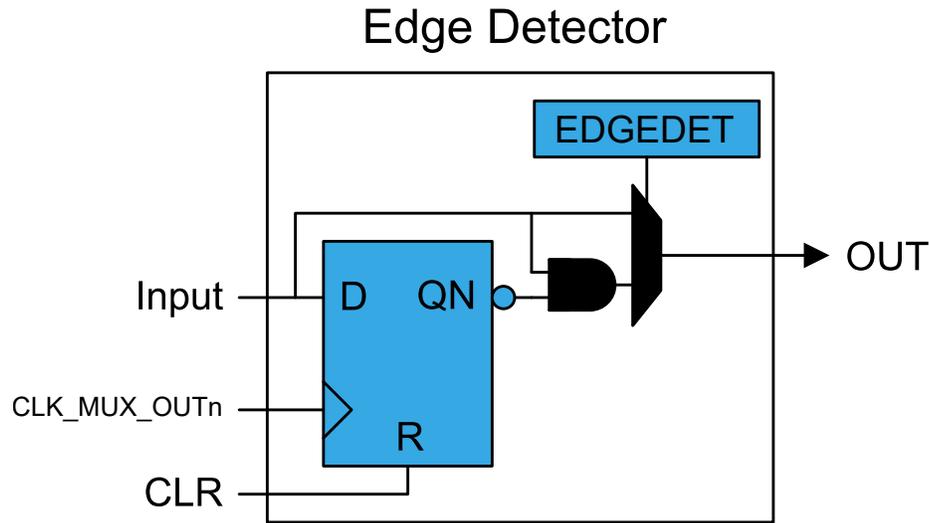
27.3.2.5. 边沿检测器

边沿检测器用于在其输入端检测到上升沿时生成脉冲。通过编程真值表提供反相输出来检测下降沿。

通过向 CCL.LUTnCTRLA 寄存器中的边沿选择 (EDGEDET) 位写入 1 来使能边沿检测器。为避免发生不可预测的行为，还必须使能有效的滤波器选项。

通过向 CCL.LUTnCTRLA 寄存器中的 EDGEDET 位写入 0 来禁止边沿检测。禁止 LUT 后，将在稍后的一个时钟周期内清零相应的内部边沿检测器逻辑。

图 27-7. 边沿检测器



27.3.2.6. 定序器逻辑

每个 LUT 对均可连接到定序器。定序器可用作 D 型触发器、JK 型触发器、门控 D 型锁存器或 RS 锁存器，具体通过对定序器控制（CCL.SEQCTRLn）寄存器中的定序器选择（SEQSEL）位进行写操作来选择。

定序器接收来自 LUT、滤波器或边沿检测器的输入，具体取决于配置。

定序器的时钟与对应偶数 LUT 的时钟相同。时钟源通过 CCL.LUTnCTRLA 寄存器中的时钟源（CLKSRC）位域来选择。

触发器输出（OUT）在时钟的上升沿刷新。禁止偶数 LUT 时，会异步清零锁存器。触发器复位信号（R）保持使能状态并持续一个时钟周期。

门控 D 型触发器（DFF）

D 输入由偶数 LUT 输出驱动，G 输入由奇数 LUT 输出驱动。

图 27-8. D 型触发器

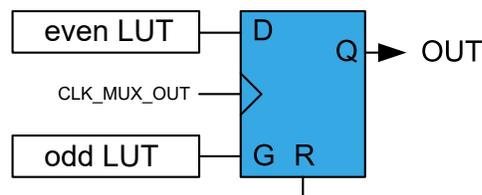


表 27-3. DFF 特性

R	G	D	OUT
1	X	X	清零
0	1	1	置 1
		0	清零
		X	保持状态（无变化）

JK 型触发器（JK）

J 输入由偶数 LUT 输出驱动，K 输入由奇数 LUT 输出驱动。

图 27-9. JK 型触发器

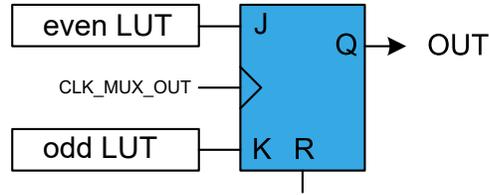


表 27-4. JK 特性

R	J	K	OUT
1	X	X	清零
0	0	0	保持状态（无变化）
0	0	1	清零
0	1	0	置 1
0	1	1	翻转

门控 D 型锁存器（DLATCH）

D 输入由偶数 LUT 输出驱动，G 输入由奇数 LUT 输出驱动。

图 27-10. D 型锁存器

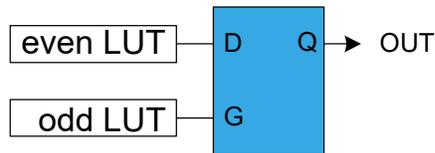


表 27-5. D 型锁存器特性

G	D	OUT
0	X	保持状态（无变化）
1	0	清零
1	1	置 1

RS 锁存器（RS）

S 输入由偶数 LUT 输出驱动，R 输入由奇数 LUT 输出驱动。

图 27-11. RS 锁存器

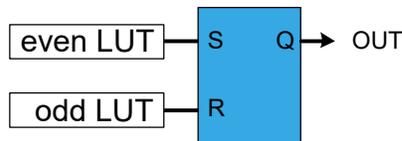


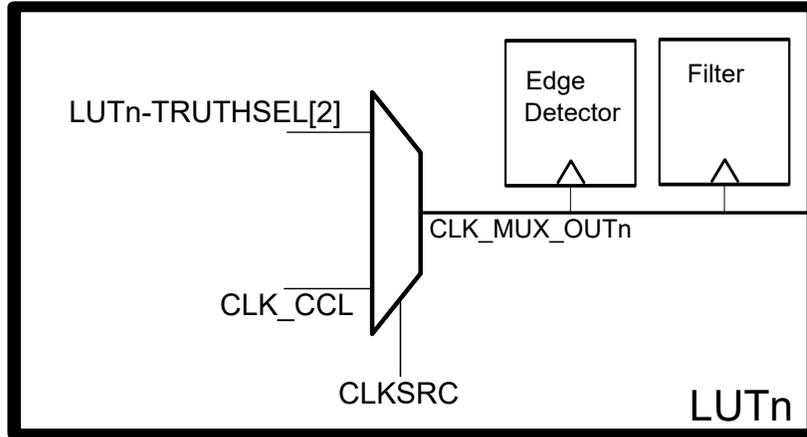
表 27-6. RS 锁存器特性

S	R	OUT
0	0	保持状态（无变化）
0	1	清零
1	0	置 1
1	1	禁止状态

27.3.2.7. 时钟源设置

默认情况下，滤波器、边沿检测器和定序器由外设时钟（CLK_PER）驱动。此外，也可以使用其他时钟输入（CLK_MUX_OUTn）为这些模块提供时钟。该配置通过写入 LUT 控制 A 寄存器中的 CLKSRC 位来实现。

图 27-12. 时钟源设置



当 CLKSRC 位写入 0x1 时，LUTn-TRUTHSEL[2]用于为相应的滤波器和边沿检测器（CLK_MUX_OUTn）提供时钟。定序器由 LUT 对中的偶数 LUT 的 CLK_MUX_OUTn 提供时钟。当 CLKSRC 写入 0x1 时，LUTn-TRUTHSEL[2]在真值表中视为关闭（低电平）。

更改时钟源时必须禁止 CCL 外设，从而避免该外设生成不确定的输出。

27.3.3. 事件

CCL 可生成以下输出事件：

- LUTnOUT：查找表输出值

CCL 可对输入事件执行以下操作：

- INx：事件用作真值表的输入

27.3.4. 休眠模式操作

向 CCL.CTRLA 寄存器中的待机运行（RUNSTDBY）位写入 1 可以在待机休眠模式下使能所选的时钟。

如果 RUNSTDBY 为 0，则外设时钟将在待机休眠模式下处于禁止状态。如果使能了滤波器、边沿检测器和/或定序器，则在待机休眠模式下 LUT 输出将强制为 0。在空闲休眠模式下，无论 RUNSTDBY 位如何，真值表解码器都将继续工作并相应地刷新 LUT 输出。

如果向 CCL.LUTnCTRLA 寄存器中的 CLKSRC 位写入 1，则 LUTn-TRUTHSEL[2]将始终为滤波器、边沿检测器和定序器模块提供时钟。在休眠模式下，LUTn-TRUTHSEL[2]时钟的可用性将取决于所用外设的休眠设置。

27.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0		RUNSTDBY						ENABLE
0x01	SEQCTRL0	7:0					SEQSEL0[3:0]			
0x02	保留									
...										
0x04										
0x05	LUT0CTRLA	7:0	EDGEDET	CLKSRC	FILTSEL[1:0]		OUTEN			ENABLE
0x06	LUT0CTRLB	7:0	INSEL1[3:0]				INSEL0[3:0]			
0x07	LUT0CTRLC	7:0					INSEL2[3:0]			
0x08	TRUTH0	7:0	TRUTH0[7:0]							
0x09	LUT1CTRLA	7:0	EDGEDET	CLKSRC	FILTSEL[1:0]		OUTEN			ENABLE
0x0A	LUT1CTRLB	7:0	INSEL1[3:0]				INSEL0[3:0]			
0x0B	LUT1CTRLC	7:0					INSEL2[3:0]			
0x0C	TRUTH1	7:0	TRUTH1[7:0]							

27.5. 寄存器说明

27.5.1. 控制 A

名称: CTRLA
偏移量: 0x00
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
		RUNSTDBY						ENABLE
访问		R/W						R/W
复位		0						0

Bit 6 – RUNSTDBY 在待机模式下运行

该位指示外设时钟（CLK_PER）在待机休眠模式下是否保持运行。对于无需 CLK_PER 的配置，将忽略该设置。

值	说明
0	待机休眠模式下无需系统时钟
1	待机休眠模式下需要系统时钟

Bit 0 – ENABLE 使能

值	说明
0	禁止外设
1	使能外设

27.5.2. 定序器控制 0

名称: SEQCTRL0
偏移量: 0x01
复位: 0x00
属性: 使能保护

位	7	6	5	4	3	2	1	0
					SEQSELO[3:0]			
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3:0 - SEQSELO[3:0] 定序器选择

该位域选择 LUT0 和 LUT1 的定序器配置。

值	名称	说明
0x0	DISABLE	禁止定序器逻辑
0x1	DFF	D 型触发器
0x2	JK	JK 型触发器
0x3	LATCH	D 型锁存器
0x4	RS	RS 锁存器
其他	-	保留

27.5.3. LUT n 控制 A

名称: LUTnCTRLA
偏移量: 0x05 + n*0x04 [n=0..1]
复位: 0x00
属性: 使能保护

位	7	6	5	4	3	2	1	0
	EDGEDET	CLKSRC	FILTSEL[1:0]		OUTEN			ENABLE
访问	R/W	R/W	R/W	R/W	R/W			R/W
复位	0	0	0	0	0			0

Bit 7 – EDGEDET 边沿检测

值	说明
0	禁止边沿检测器
1	使能边沿检测器

Bit 6 – CLKSRC 时钟源选择

该位选择是将外设时钟（CLK_PER）还是由 LUTnCTRLC 寄存器（LUTn-TRUTHSEL[2]）的 INSEL2 位域选择的源用作 LUT 的时钟（CLK_MUX_OUTn）。

偶数 LUT 的 CLK_MUX_OUTn 用于为 LUT 对的定序器模块提供时钟。

值	说明
0	CLK_PER 为 LUTn 提供时钟
1	LUTn-TRUTHSEL[2]为 LUTn 提供时钟

Bit 5:4 – FILTSEL[1:0] 滤波器选择

该位域用于选择 LUT 输出滤波器选项。

值	名称	说明
0x0	DISABLE	禁止滤波器
0x1	SYNCH	使能同步器
0x2	FILTER	使能滤波器
0x3	-	保留

Bit 3 – OUTEN 输出使能

该位用于使能 LUTnOUT 引脚的 LUT 输出。写入 1 时，将改写 PORT I/O 控制器的引脚配置。

值	说明
0	禁止输出到引脚
1	使能输出到引脚

Bit 0 – ENABLE LUT 使能

值	说明
0	禁止 LUT
1	使能 LUT

27.5.4. LUT n 控制 B

名称: LUTnCTRLB
偏移量: 0x06 + n*0x04 [n=0..1]
复位: 0x00
属性: 使能保护

注:

- 与 CCL 的 SPI 连接仅适用于主 SPI 模式。
- 仅当 USART 处于以下一种模式时，才可使用 CCL 的 USART 连接：
 - 异步 USART
 - 同步 USART 主模式

位	7	6	5	4	3	2	1	0
	INSEL1[3:0]				INSEL0[3:0]			
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:4 - INSEL1[3:0] LUT n 输入 1 源选择

该位域用于选择 LUT n 输入 1 的源。

值	名称	说明
0x0	MASK	屏蔽输入
0x1	FEEDBACK	反馈输入
0x2	LINK	链接其他 LUT 作为输入源
0x3	EVENT0	事件输入源 0
0x4	EVENT1	事件输入源 1
0x5	IO	I/O 引脚 LUTn-IN1 输入源
0x6	AC0	AC0 OUT 输入源
0x7	TCB0	TCB0 WO 输入源
0x8	TCA0	TCA0 WO1 输入源
0x9	TCD0	TCD0 WOB 输入源
0xA	USART0	USART0 TXD 输入源
0xB	SPI0	SPI0 MOSI 输入源
0xC	AC1 ⁽¹⁾	AC1 OUT 输入源
0xD	TCB1 ⁽¹⁾	TCB1 WO 输入源
0xE	AC2 ⁽¹⁾	AC2 OUT 输入源
其他	-	保留

注:

- 不适用于 CN0816。

Bit 3:0 - INSEL0[3:0] LUT n 输入 0 源选择

该位域用于选择 LUT n 输入 0 的源。

值	名称	说明
0x0	MASK	屏蔽输入
0x1	FEEDBACK	反馈输入
0x2	LINK	链接其他 LUT 作为输入源
0x3	EVENT0	事件输入源 0
0x4	EVENT1	事件输入源 1
0x5	IO	I/O 引脚 LUTn-IN0 输入源
0x6	AC0	AC0 OUT 输入源

INSELO (续)		
值	名称	说明
0x7	TCB0	TCB0 WO 输入源
0x8	TCA0	TCA0 W00 输入源
0x9	TCD0	TCD0 W0A 输入源
0xA	USART0	USART0 XCK 输入源
0xB	SPI0	SPI0 SCK 输入源
0xC	AC1 ⁽¹⁾	AC1 OUT 输入源
0xD	TCB1 ⁽¹⁾	TCB1 WO 输入源
0xE	AC2 ⁽¹⁾	AC2 OUT 输入源
其他	-	保留

注:

1. 不适用于 CN0816。

27.5.5. LUT n 控制 C

名称: LUTnCTRLC
偏移量: 0x07 + n*0x04 [n=0..1]
复位: 0x00
属性: 使能保护

位	7	6	5	4	3	2	1	0
					INSEL2[3:0]			
访问					R/W	R/W	R/W	R/W
复位					0	0	0	0

Bit 3:0 - INSEL2[3:0] LUT n 输入 2 源选择
这些位用于选择 LUT n 输入 2 的源。

值	名称	说明
0x0	MASK	屏蔽输入
0x1	FEEDBACK	反馈输入
0x2	LINK	链接其他 LUT 作为输入源
0x3	EVENT0	事件输入源 0
0x4	EVENT1	事件输入源 1
0x5	IO	I/O 引脚 LUTn-IN2 输入源
0x6	AC0	AC0 OUT 输入源
0x7	TCB0	TCB0 WO 输入源
0x8	TCA0	TCA0 WO2 输入源
0x9	TCD0	TCD0 WOA 输入源
0xA	-	保留
0xB	SPI0	SPI0 MISO 输入源
0xC	AC1(1)	AC1 OUT 输入源
0xD	TCB1(1)	TCB1 WO 输入源
0xE	AC2(1)	AC2 OUT 输入源
其他	-	保留

注:

1. 不适用于 CN0816。

27.5.6. TRUTHn

名称: TRUTHn
偏移量: $0x08 + n * 0x04$ [n=0..1]
复位: 0x00
属性: 使能保护

位	7	6	5	4	3	2	1	0
	TRUTHn[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - TRUTHn[7:0] 真值表

这些位根据 LUTn-TRUTHSEL[2:0]输入确定 LUTn 的输出。

位名称	值	说明
TRUTHn[0]	0	当输入为'b000时, LUTn 的输出为 0
	1	当输入为'b000时, LUTn 的输出为 1
TRUTHn[1]	0	当输入为'b001时, LUTn 的输出为 0
	1	当输入为'b001时, LUTn 的输出为 1
TRUTHn[2]	0	当输入为'b010时, LUTn 的输出为 0
	1	当输入为'b010时, LUTn 的输出为 1
TRUTHn[3]	0	当输入为'b011时, LUTn 的输出为 0
	1	当输入为'b011时, LUTn 的输出为 1
TRUTHn[4]	0	当输入为'b100时, LUTn 的输出为 0
	1	当输入为'b100时, LUTn 的输出为 1
TRUTHn[5]	0	当输入为'b101时, LUTn 的输出为 0
	1	当输入为'b101时, LUTn 的输出为 1
TRUTHn[6]	0	当输入为'b110时, LUTn 的输出为 0
	1	当输入为'b110时, LUTn 的输出为 1
TRUTHn[7]	0	当输入为'b111时, LUTn 的输出为 0
	1	当输入为'b111时, LUTn 的输出为 1

28. AC——模拟比较器

28.1. 特性

- 电源电压高于 2.7V 时的响应时间为 50 ns
- 过零检测
- 可选滞后：
 - 无
 - 10 mV
 - 25 mV
 - 50 mV
- 引脚上提供模拟比较器输出
- 可实现比较器输出反相
- 可灵活选择输入：
 - 最多四个同相引脚
 - 最多两个反相引脚
 - DAC 的输出
 - 内部参考电压
- 在出现以下边沿时生成中断：
 - 上升沿
 - 下降沿
 - 上升沿和下降沿
- 生成事件：
 - 比较器输出

28.2. 概述

模拟比较器（AC）比较两个输入端的电压大小，并根据此比较结果提供数字输出。AC 可配置为根据输入变化的几种不同组合产生中断请求和/或事件。

可通过滞后特性调整 AC 的动态行为。可针对每个应用自定义滞后，从而优化操作。

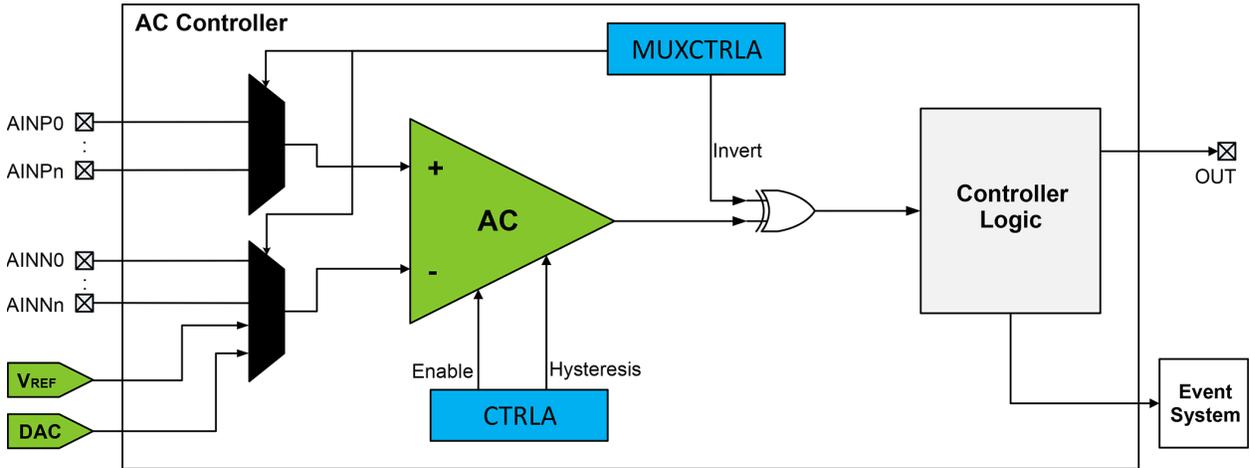
输入选择包括模拟端口引脚、DAC 输出和内部参考和内部参考电压。模拟比较器输出状态也可在引脚上输出，以供外部器件使用。

AC 具有一个同相输入和一个反相输入。同相输入源为其中一个模拟输入引脚。反相输入从模拟输入引脚或者内部输入中选择（例如内部参考电压）。

同相输入电压和反相输入电压之差为正时，比较器的数字输出为 1，否则为 0。

28.2.1. 框图

图 28-1. 模拟比较器



注：有关 AINN 和 AINP 的数量，请参见特性。

28.2.2. 信号说明

信号	说明	类型
AINNn	反相输入 n	模拟
AINPn	同相输入 n	模拟
OUT	AC 的比较器输出	数字

28.2.3. 系统相关性

要使用此外设，必须按如下所述正确配置系统的其他部分。

表 28-1. AC 系统相关性

相关性	适用	外设
时钟	是	CLKCTRL
I/O 线路和连接	是	PORT
中断	是	CPUINT
事件	是	EVSYS
调试	是	UPDI

28.2.3.1. 时钟

该外设取决于外设时钟。

28.2.3.2. I/O 线路和连接

I/O 引脚 AINNn 和 AINPn 均为 AC 的模拟输入。

为了正常工作，必须在端口和端口多路复用外设中配置相应引脚。

建议在使用 AC 时禁止 GPIO 输入。

28.2.3.3. 中断

使用此外设的中断需要先配置中断控制器。

28.2.3.4. 事件

此外设的事件连接至事件系统。

28.2.3.5. 调试操作

进入调试模式不会影响此外设。

如果外设配置为需要由 CPU 通过中断或类似操作进行定期控制，则在调试期间停止 CPU 可能会产生不正确的操作或丢失数据。

28.3. 功能说明

28.3.1. 初始化

要进行基本操作，请按照以下步骤操作：

1. 配置端口外设中所需的输入引脚。
2. 写入 MUX 控制 A (ACn.MUXCTRLA) 寄存器中的同相和反相输入 MUX 选择 (MUXPOS 和 MUXNEG) 位域，选择同相和反相输入源。
3. 可选：向控制 A (ACn.CTRLA) 寄存器中的输出焊盘使能 (OUTEN) 位写入 1，使能引脚输出。
4. 向 ACn.CTRLA 寄存器中的 ENABLE 位写入 1，使能 AC。

在使能 AC 后的启动时间内，AC 的输出可能无效。

AC 自身的启动时间最长为 2.5 μs 。如果使用内部参考，则参考启动时间通常比 AC 启动时间长。VREF 启动时间最长为 60 μs 。

28.3.2. 工作原理

28.3.2.1. 输入滞后

当受噪声影响的输入信号彼此接近时，应用输入滞后有助于防止不断切换输出。

输入滞后可禁止，也可设为三个级别之一，具体通过对控制 A (ACn.CTRLA) 寄存器中的滞后模式选择 (HYSMODE) 位域进行写操作来配置。

28.3.2.2. 输入源

AC 具有一个同相输入和一个反相输入。这两个输入可以是引脚和内部源，例如参考电压。

通过对 MUX 控制 A (ACn.MUXCTRLA) 寄存器中的同相和反相输入 MUX 选择 (MUXPOS 和 MUXNEG) 位域进行写操作来选择每个输入。

28.3.2.2.1. 引脚输入

可选择端口上的以下模拟输入引脚作为模拟比较器的输入：

- AINN0
- AINN1⁽¹⁾
- AINP0
- AINP1⁽¹⁾
- AINP2⁽¹⁾
- AINP3⁽¹⁾

注：

1. 低引脚数器件上只提供部分端口引脚。有关详细信息，请查看引脚图和/或 I/O 复用表。

28.3.2.2.2. 内部输入

AC 具有以下内部输入：

- DAC 的输出
- AC 参考电压

28.3.2.3. 低功耗模式

对于功耗敏感型应用，AC 通过提供低功耗模式来降低功耗，但会延长传播延时。

通过向控制 A（ACn.CTRLA）寄存器中的低功耗模式（LPMODE）位写入 1 来使能该模式。

28.3.3. 事件

使能 AC 后，AC 将自动生成以下事件：

- AC 的数字输出（框图中的 OUT）可用作事件系统源。来自 AC 的事件与器件中的任何时钟都是异步的。

AC 没有事件输入。

28.3.4. 中断

表 28-2. 可用中断向量和源

失调电压	名称	向量说明	条件
0x00	COMP0	模拟比较器中断	AC 输出根据 ACn.CTRLA 中的 INTMODE 的配置翻转

发生中断条件时，STATUS（ACn.STATUS）寄存器中的相应中断标志将置 1。

可以通过写入外设的中断控制（ACn.INTCTRL）寄存器中的相应位来允许或禁止中断源。

当相应的中断源被允许且中断标志置 1 时，将产生中断请求。中断请求保持有效，直至中断标志清零。有关如何清零中断标志的详细信息，请参见 ACn.STATUS 寄存器说明。

28.3.5. 休眠模式操作

在空闲休眠模式下，AC 将继续正常工作。

在待机休眠模式下，AC 默认处于禁止状态。如果控制 A（ACn.CTRLA）寄存器中的待机休眠模式（RUNSTDBY）位写入 1，则 AC 将继续工作，但状态寄存器将不会更新，如果没有其他模块请求 CLK_PER，则不会产生中断，但会更新事件和焊盘输出。

在掉电休眠模式下，AC 和到焊盘的输出都处于禁止状态。

28.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY	OUTEN	INTMODE[1:0]		LPMODE	HYSMODE[1:0]		ENABLE
0x01	保留									
0x02	MUXCTRLA	7:0	INVERT			MUXPOS[1:0]			MUXNEG[1:0]	
0x03	保留									
...										
0x05	保留									
0x06	INTCTRL	7:0								CMP
0x07	STATUS	7:0				STATE				CMP

28.5. 寄存器说明

28.5.1. 控制 A

名称: CTRLA
偏移量: 0x00
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	RUNSTDBY	OUTEN	INTMODE[1:0]		LPMODE	HYSMODE[1:0]		ENABLE
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7 - RUNSTDBY 在待机模式下运行

向该位写入 1 可允许 AC 在待机休眠模式下继续工作。由于时钟停止，中断和状态标志不会更新。

值	说明
0	在待机休眠模式下，外设停止
1	在待机休眠模式下，外设继续工作

Bit 6 - OUTEN 模拟比较器输出焊盘使能

向该位写入 1 可使引脚上的 OUT 信号可用。

Bit 5:4 - INTMODE[1:0] 中断模式

写入该位域可选择 AC 输出的哪些边沿会触发中断请求。

值	名称	说明
0x0	BOTHEDGE	负边沿和正边沿
0x1	-	保留
0x2	NEGEDGE	负边沿
0x3	POSEDGE	正边沿

Bit 3 - LPMODE 低功耗模式

向该位写入 1 会减小流过比较器的电流。这会降低功耗，但会增加 AC 的响应时间。

值	说明
0	禁止低功耗模式
1	使能低功耗模式

Bit 2:1 - HYSMODE[1:0] 滞后模式选择

写入该位域可选择 AC 输入的滞后模式。

值	名称	说明
0x0	OFF	关闭
0x1	10	±10 mV
0x2	25	±25 mV
0x3	50	±50 mV

Bit 0 - ENABLE 使能 AC

向该位写入 1 可使能 AC。

28.5.2. MUX 控制 A

名称: MUXCTRLA
偏移量: 0x02
复位: 0x00
属性: -

ACn.MUXCTRLA 控制模拟比较器多路开关。

位	7	6	5	4	3	2	1	0
	INVERT			MUXPOS[1:0]			MUXNEG[1:0]	
访问	R/W			R/W	R/W		R/W	R/W
复位	0			0	0		0	0

Bit 7 - INVERT 将 AC 输出反相

向该位写入 1 可将 AC 的输出反相。这能够有效地将信号连接的所有外设的输入反相并影响内部状态信号。

Bit 4:3 - MUXPOS[1:0] 同相输入多路开关选择

写入该位域可选择 AC 的同相输入的输入信号。

值	名称	说明
0x0	AINP0	同相引脚 0
0x1	AINP1 ⁽¹⁾	同相引脚 1
0x2	AINP2 ⁽¹⁾	同相引脚 2
0x3	AINP3 ⁽¹⁾	同相引脚 3

注:

1. 低引脚数器件上只提供部分端口引脚。有关详细信息，请查看引脚图和/或 I/O 复用表。

Bit 1:0 - MUXNEG[1:0] 反相输入多路开关选择

写入该位域可选择 AC 的反相输入的输入信号。

值	名称	说明
0x0	AINN0	反相引脚 0
0x1	AINN1 ⁽¹⁾	反相引脚 1
0x2	VREF	参考电压
0x3	DAC	DAC 输出

注:

1. 低引脚数器件上只提供部分端口引脚。有关详细信息，请查看引脚图和/或 I/O 复用表。

28.5.3. 中断控制

名称: INTCTRL
 偏移量: 0x06
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
访问								CMP
复位								R/W 0

Bit 0 - CMP 模拟比较器中断允许

向该位写入 1 可允许模拟比较器中断。

28.5.4. 状态

名称: STATUS
 偏移量: 0x07
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
				STATE				CMP
访问				R				R/W
复位				0				0

Bit 4 - STATE 模拟比较器状态

该位将显示来自 AC 的 OUT 信号的当前状态。这将会在 I/O 寄存器中更新同步器延时（三个周期）。

Bit 0 - CMP 模拟比较器中断标志

这是 AC 的中断标志。向该位写入 1 会将中断标志清零。

29. ADC——模数转换器

29.1. 特性

- 10 位分辨率
- 输入电压范围为 0V 至 V_{DD}
- 多个内部 ADC 参考电压
- 外部参考电压输入
- 自由运行和单次转换模式
- 转换完成时可产生中断
- 转换结果就绪时产生中断（可选）
- 温度传感器输入通道
- 通过事件触发转换（可选）
- 用于精确监视或定义阈值的窗口比较器功能
- 每次转换最多可累加 64 个采样

29.2. 概述

模数转换器（ADC）外设生成 10 位结果。ADC 输入既可为内部输入（例如，参考电压），也可为外部输入（通过模拟输入引脚实现）。ADC 连接到一个模拟多路开关，有多个单端电压输入可供选择。单端电压输入以 0V（GND）为参考。

ADC 支持突发采样，经配置后可将一定数量的转换结果累加到单个 ADC 结果中（采样累加）。此外，可通过配置采样延时来调整与单次突发相关联的 ADC 采样频率，使其远离会在突发采样器期间与采样频率发生混淆的谐波噪声，避免影响采样信号。此外，还可使用自动采样延时变化功能对该延时进行随机化处理，以此略微改变采样间隔。

ADC 输入信号通过一个采样保持电路馈送，可确保 ADC 的输入电压在采样期间保持在恒定水平。

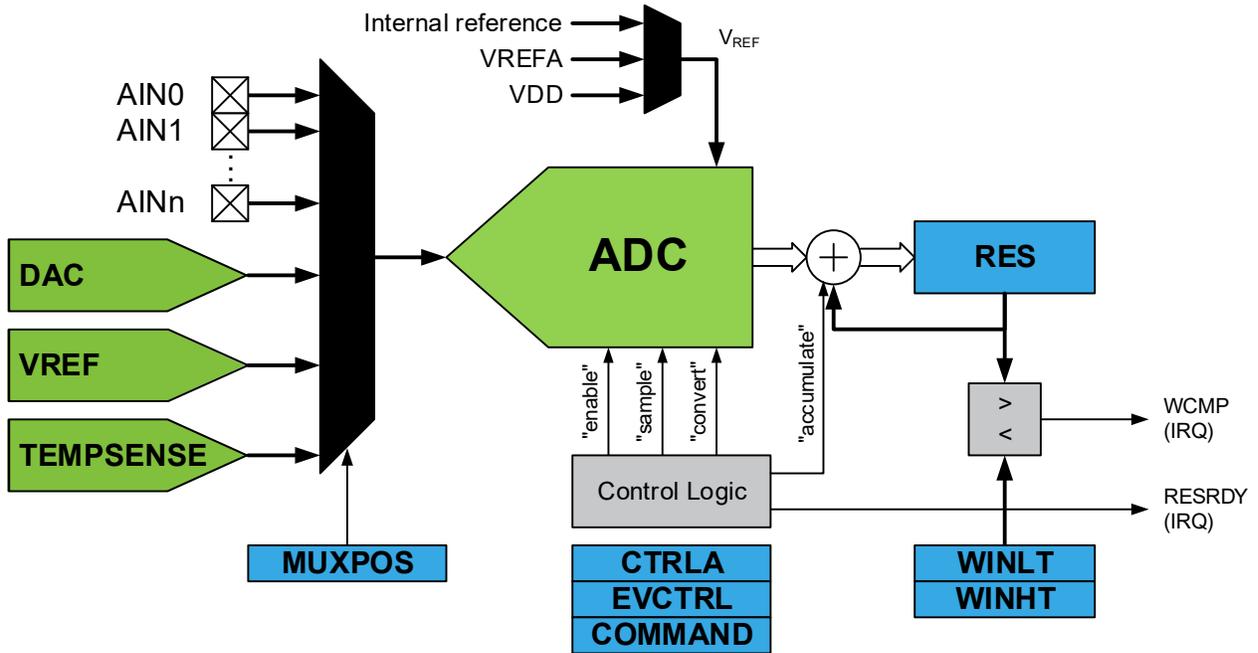
可选择内部参考电压（VREF）外设、外部 VREF 引脚（VREFA）或 VDD 电源电压作为参考电压。

窗口比较功能可用于监视输入信号，经配置后可仅在达到用户自定义阈值（低于窗口、高于窗口、位于窗口内部或位于窗口外部）时触发中断，几乎无需软件干预。

使能外设触摸控制器（PTC）时，ADC0 完全由 PTC 外设控制。

29.2.1. 框图

图 29-1. ADC 框图



注：VREFA 不适用于 CN0816。

通过写入 MUXPOS (ADCn.MUXPOS) 寄存器中的 MUXPOS 位域选择模拟输入通道。ADC 的任何输入引脚、GND、内部参考电压 (V_{REF}) 或温度传感器都可以被选作 ADC 的单端输入。通过向控制 A (ADCn.CTRLA) 寄存器中的 ENABLE 位写入 1 来使能 ADC。参考电压和输入通道选择在 ADC 使能之后才会生效。当 ADCn.CTRLA 寄存器中的 ENABLE 位为 0 时，ADC 不消耗功率。

ADC 会生成一个 10 位结果 (可从结果 (ADCn.RES) 寄存器中读取)，并以右对齐的形式呈现。

29.2.2. 信号说明

引脚名称	类型	说明
AIN[n:0]	模拟输入	模拟输入引脚
VREFA	模拟输入	外部参考电压引脚

29.3. 功能说明

29.3.1. 初始化

建议按照以下步骤初始化 ADC 操作：

1. 写入控制 A (ADCn.CTRLA) 寄存器中的分辨率选择 (RESSEL) 位，配置分辨率。
2. 可选：向 ADCn.CTRLA 寄存器中的自由运行 (FREERUN) 位写入 1，使能自由运行模式。
3. 可选：写入控制 B (ADCn.CTRLB) 寄存器中的采样累加数选择 (SAMPNUM) 位，配置每次转换要累加的采样数。
4. 写入控制 C (ADCn.CTRLC) 寄存器中的参考电压选择 (REFSEL) 位，配置参考电压。默认值是器件的内部参考电压 (相应位置配置的 V_{REF})。
5. 写入控制 C (ADCn.CTRLC) 寄存器中的预分频比 (PRESC) 位域，配置 ADC 时钟 (CLK_ADC)。
6. 写入 MUXPOS (ADCn.MUXPOS) 寄存器中的 MUXPOS 位域，配置输入。

7. 可选：向事件控制（ADCn.EVCTRL）寄存器中的启动事件输入（STARTEI）位写入 1，使能启动事件输入。相应地配置事件系统。
8. 向 ADCn.CTRLA 寄存器中的 ENABLE 位写入 1，使能 ADC。

执行上述步骤后即已完成 ADC 的基本测量初始化，随后可通过事件（如果已配置）或通过向命令（ADCn.COMMAND）寄存器中的启动转换（STCONV）位写入 1 来触发基本测量。

29.3.1.1. I/O 线路和连接

I/O 引脚 AINx 和 VREF 通过 I/O 引脚控制器端口配置。

应在用作 ADC 输入的引脚上禁止数字输入缓冲器，以断开数字域与模拟域的连接，从而尽可能获得最佳的 ADC 结果。这通过 PORT 外设置。

29.3.2. 工作原理

29.3.2.1. 启动转换

通过对 MUXPOS（ADCn.MUXPOS）寄存器进行写操作来选择输入通道，然后向命令（ADCn.COMMAND）寄存器中的 ADC 启动转换（STCONV）位写入 1 来触发转换。只要转换正在进行，该位就为 1。在单次转换模式下，STCONV 会在转换完成后由硬件清零。

如果在转换过程中选择了另一输入通道，则 ADC 会在完成当前转换之后再切换通道。

根据累加器设置，转换结果可能来自单次检测操作或累加采样序列。触发的操作完成后，中断标志（ADCn.INTFLAGS）寄存器中的结果就绪（RESRDY）标志会置 1。如果中断控制（ADCn.INTCTRL）寄存器中的结果就绪中断允许（RESRDY）位为 1 且全局中断允许位为 1，则执行相应的中断向量。

通过向 ADCn.COMMAND 寄存器中的 STCONV 位写入 1 可启动单次转换。STCONV 位可用于确定是否正在进行转换。STCONV 位在转换期间将置 1，在转换完成后将清零。

即使禁止特定中断，ADCn.INTFLAGS 寄存器中的 RESRDY 中断标志也将置 1，随后软件可通过轮询该标志检查转换是否完成。因此，可以在不引起中断的情况下触发转换。

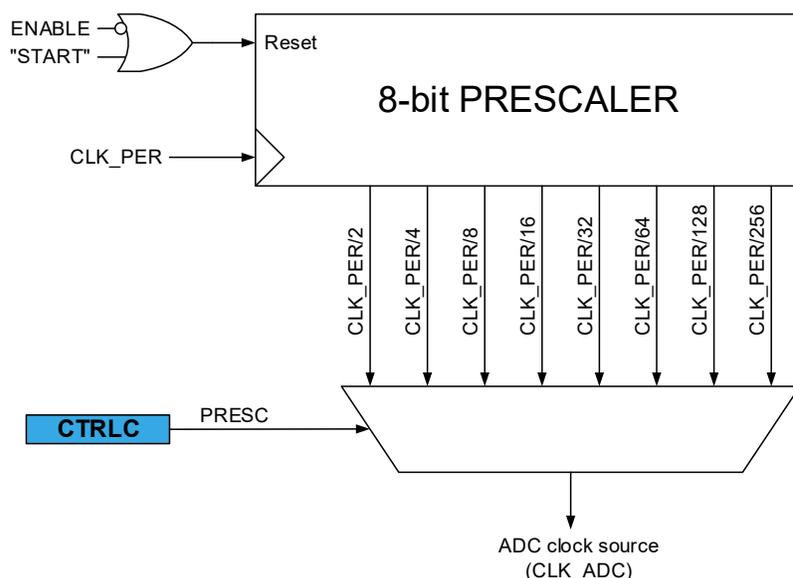
或者，也可通过事件触发转换。通过向事件控制（ADCn.EVCTRL）寄存器中的启动事件输入（STARTEI）位写入 1 来使能该功能。通过事件系统（EVSYS）传输到 ADC 的传入事件将触发 ADC 转换。通过这种方法，就能以可预测的间隔或在特定条件下启动转换。

事件触发输入为边沿敏感型。发生相关事件时，ADCn.COMMAND 寄存器中的 STCONV 置 1。转换完成后，STCONV 位将清零。

在自由运行模式下，通过向 ADCn.COMMAND 寄存器中的 STCONV 位写入 1 来启动第一次转换。在上一个转换周期完成后，会立即启动新的转换周期。转换完成后，会将 ADCn.INTFLAGS 中的 RESRDY 标志置 1。

29.3.2.2. 时钟生成

图 29-2. ADC 预分频器



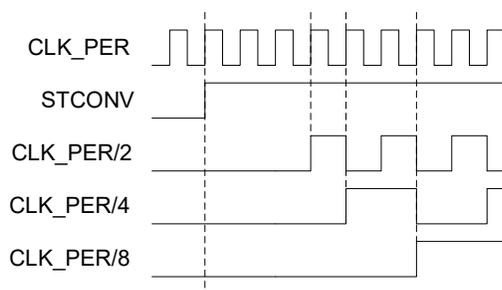
ADC 需要一个介于 50 kHz 和 1.5 MHz 之间的输入时钟频率来获得最大分辨率。如果选择的分辨率低于 10 位，ADC 的输入时钟频率可能高于 1.5 MHz，此时可获得更高的采样率。

ADC 模块包含一个预分频器，其作用是通过超过 100 kHz 的外设时钟（CLK_PER）生成 ADC 时钟（CLK_ADC）。通过写入控制 C（ADCn.CTRLC）寄存器中的预分频比（PRESC）位来选择预分频。预分频器在 ADC 开启时开始计数，ADC 的开启方式是向 ADCn.CTRLA 寄存器中的 ENABLE 位写入 1。只要 ENABLE 位为 1，预分频器就会保持运行。ENABLE 位为 0 时，预分频器计数器复位为零。

通过向命令（ADCn.COMMAND）寄存器中的启动转换（STCONV）位写入 1 或通过事件来启动转换时，转换在一个 CLK_PER 周期后启动。只要没有正在进行的转换，预分频器就会保持复位。这可确保从触发转换到实际开始转换之间存在一个固定延时（最长 2 个 CLK_PER 周期），其计算公式如下：

$$\text{StartDelay} = \frac{\text{PRESC}_{\text{factor}}}{2} + 2$$

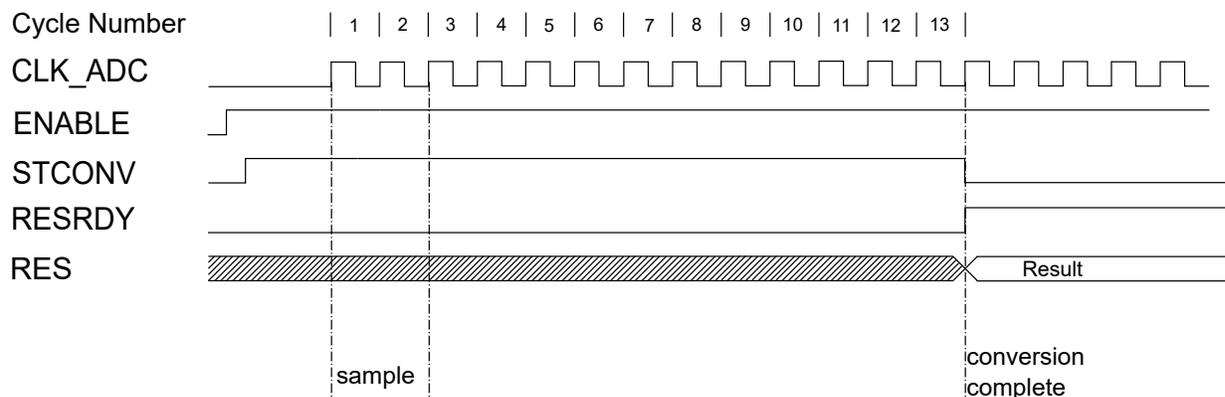
图 29-3. 启动转换和时钟生成



29.3.2.3. 转换时序

正常转换需要 13 个 CLK_ADC 周期。实际采样保持在开始转换的两个 CLK_ADC 周期后进行。通过向 ADCn.COMMAND 中的 STCONV 位写入 1 来启动转换。转换完成后，结果（ADCn.RES）寄存器中将提供结果，中断标志（ADCn.INTFLAGS）寄存器中的结果就绪（RESRDY）中断标志将置 1。从结果（ADCn.RES）寄存器读取结果时将清零此中断标志，或者也可通过向 ADCn.INTFLAGS 寄存器中的 RESRDY 位写入 1 来清零。

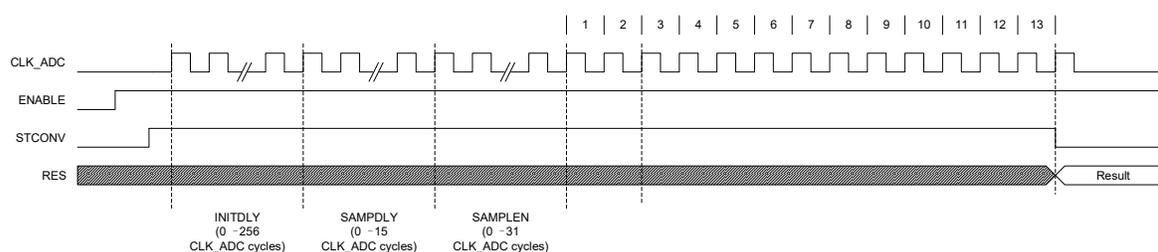
图 29-4. ADC 时序图——单次转换



通过使用控制 D (ADCn.CTRLD) 寄存器中的采样延时 (SAMPDLY) 位域以及采样控制 (ADCn.SAMPCTRL) 寄存器中的采样长度 (SAMPLN) 位域, 可以调整采样时间和采样长度。这两个位域控制 ADC 的采样时间 (若干个 CLK_ADC 周期)。这样可以在不影响转换速度的情况下对高阻抗源进行采样。更多信息, 请参见寄存器说明。总采样时间的计算公式如下:

$$T_{\text{采样}} = \frac{(2 + \text{SAMPDLY} + \text{SAMPLN})}{f_{\text{CLK_ADC}}}$$

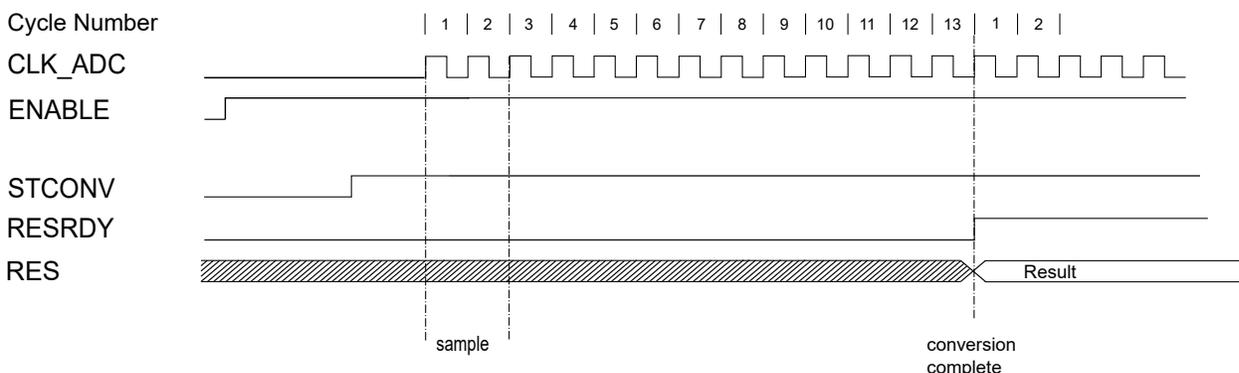
图 29-5. ADC 时序图——具有延时的单次转换



在自由运行模式下, 如果 STCONV 位为 1, 则在转换完成后会立即启动新的转换。自由运行模式下的采样率 R_S 的计算公式如下:

$$R_S = \frac{f_{\text{CLK_ADC}}}{(13 + \text{SAMPDLY} + \text{SAMPLN})}$$

图 29-6. ADC 时序图——自由运行转换



29.3.2.4. 更改通道或参考选择

ADCn.MUXPOS 寄存器中的 MUXPOS 位域以及 ADCn.CTRLA 寄存器中的 REFSEL 位域均通过 CPU 随机访问的临时寄存器进行缓冲。这样可确保仅在转换期间的安全点进行通道和参考电压选择。在启动转换之前，通道和参考电压选择会不断更新。

一旦转换开始，通道和参考电压选择即锁定，以确保 ADC 有足够的采样时间。在转换完成之前的最后一个 CLK_ADC 时钟周期内将恢复连续更新（ADCn.INTFLAGS 寄存器中的 RESRDY 位置 1）。在向 STCONV 位写入 1 后的下一个 CLK_ADC 时钟上升沿开始转换。

29.3.2.4.1. ADC 输入通道

更改通道选择时，用户必须遵循以下准则，以确保选择正确的通道：

在单次转换模式下：应在开始转换之前选择通道。可在向 STCONV 位写入 1 后的一个 ADC 时钟周期内更改通道选择。

在自由运行模式下：应在开始第一次转换之前选择通道。可在向 STCONV 位写入 1 后的一个 ADC 时钟周期内更改通道选择。由于下一次转换已自动启动，因此下一个结果将反映上一个通道选择。后续转换将反映新的通道选择。

切换输入通道后，ADC 需要一段稳定时间——有关详细信息，请参见 *电气特性* 一章。

29.3.2.4.2. ADC 参考电压

ADC 的参考电压 (V_{REF}) 控制 ADC 的转换范围。超出所选 V_{REF} 的输入电压将转换为 ADC 的最大结果值。对于理想的 10 位 ADC，该值为 $0 \times 3FF$ 。

通过向控制 C (ADCn.CTRLA) 寄存器中的参考电压选择 (REFSEL) 位域进行写操作，可选择 V_{DD} 、外部参考电压 V_{REFA} 或来自 VREF 外设的内部参考电压作为 V_{REF} 。 V_{DD} 通过无源开关连接到 ADC。

使用外部参考电压 V_{REFA} 时，将相应参考电压 (VREF.CTRLn) 寄存器中的 ADCnREFSEL[0:2] 配置为最接近但高于所施加参考电压的值。如果外部参考电压高于 4.3V，请使用 $ADCnREFSEL[0:2] = 0 \times 3$ 。

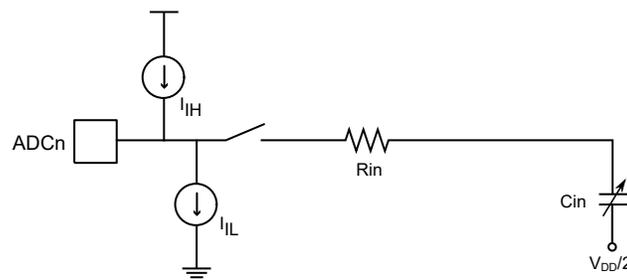
内部参考电压由内部带隙参考电压通过内部放大器生成（通过参考电压 (VREF) 外设控制）。

29.3.2.4.3. 模拟输入电路

模拟输入电路如图 29-7 所示。施加到 ADCn 的模拟源受引脚电容和该引脚的输入泄漏电流（用 I_{IH} 和 I_{IL} 表示）影响，而与是否将相应通道选作 ADC 的输入无关。选择通道时，此模拟源必须通过串联电阻（输入路径中的组合电阻）驱动采样保持电容。

ADC 通过一个约为 $10\text{ k}\Omega$ 或更低的输出阻抗针对模拟信号进行了优化。如果使用此类源，则采样时间将忽略不计。如果使用阻抗较高的模拟源，采样时间将取决于模拟源需要为采样保持电容充电多长时间，因此会存在较大差异。

图 29-7. 模拟输入原理图



29.3.2.5. ADC 转换结果

转换完成后（RESRDY 为 1），转换结果 RES 将在 ADC 结果 (ADCn.RES) 寄存器中提供。10 位转换结果的计算公式如下：

$$RES = \frac{1023 \times V_{IN}}{V_{REF}}$$

其中， V_{IN} 是所选输入引脚上的电压， V_{REF} 是所选参考电压（见 ADCn.CTRLC 和 ADCn.MUXPOS 寄存器中的 REFSEL 说明）。

29.3.2.6. 温度测量

温度测量基于片上温度传感器实现。要进行温度测量，请按照下列步骤操作：

1. 通过配置 VREF 外设将内部参考电压配置为 1.1V。
2. 通过向 ADCn.CTRLC 寄存器中的 REFSEL 位域写入 0x0 来选择内部参考电压。
3. 通过配置 MUXPOS (ADCn.MUXPOS) 寄存器来选择 ADC 温度传感器通道。这将使能温度传感器。
4. 在 ADCn.CTRLD 中，选择 $INITDLY \geq 32\mu s \times f_{CLK_ADC}$ 。
5. 在 ADCn.SAMPCTRL 中，选择 $SAMPLEN \geq 32\mu s \times f_{CLK_ADC}$ 。
6. 在 ADCn.CTRLA 中，选择 $SAMPCAP = 1$ 。
7. 通过启动转换获取温度传感器输出电压。
8. 按如下所述处理测量结果。

测量电压与温度呈线性关系。由于工艺差异，相同温度下的温度传感器输出电压也会因器件而异。各个补偿因子在生产测试期间确定并保存在签名行中：

- SIGROW.TEMPESENSE0 为增益/斜率校正
- SIGROW.TEMPESENSE1 为失调校正

为获得准确的结果，必须使用出厂校准值通过应用软件来处理温度传感器测量的结果。温度（单位为开尔文）按照以下规则进行计算：

```
Temp = (((RESH << 8) | RESL) - TEMPESENSE1) * TEMPESENSE0 >> 8
```

RESH 和 RESL 分别表示结果 (ADCn.RES) 寄存器的高字节和低字节，TEMPESENSEn 为签名行的相应值。

建议在用户代码中执行以下步骤：

```
int8_t sigrow_offset = SIGROW.TEMPESENSE1; // Read signed value from signature row
uint8_t sigrow_gain = SIGROW.TEMPESENSE0; // Read unsigned value from signature row
uint16_t adc_reading = ADCn.RES; // ADC conversion result with 1.1 V internal reference

uint32_t temp = adc_reading - sigrow_offset;
temp *= sigrow_gain; // Result might overflow 16 bit variable (10bit+8bit)
temp += 0x80; // Add 1/2 to get correct rounding on division below
temp >>= 8; // Divide result to get Kelvin
uint16_t temperature_in_K = temp;
```

29.3.2.7. 窗口比较器模式

当转换结果高于和/或低于特定阈值时，ADC 可将中断和标志 (ADCn.INTFLAGS) 寄存器中的 WCMP 标志置 1 并请求中断 (WCMP)。支持以下模式：

- 结果低于阈值
- 结果高于阈值
- 结果位于窗口内部（高于下限阈值，但低于上限阈值）
- 结果位于窗口外部（低于下限阈值或高于上限阈值）

通过对窗口比较器阈值 (ADCn.WINLT 和 ADCn.WINHT) 寄存器进行写操作来定义阈值。通过对控制 E (ADCn.CTRLE) 寄存器中的窗口比较器模式 (WINCM) 位域进行写操作来选择标志置 1 和/或请求中断的条件。

假设 ADC 已配置为运行，按照以下步骤使用窗口比较器模式：

1. 选择要使用的窗口比较器（见 ADCn.CTRLE 寄存器中的 WINCM 说明），并通过对 ADCn.WINLT 和/或 ADCn.WINHT 寄存器进行写操作来设置所需阈值。
2. 可选：通过向中断控制（ADCn.INTCTRL）寄存器中的窗口比较器中断允许（WCMP）位写入 1 来允许中断请求。
3. 通过向 ADCn.CTRLE 寄存器中的 WINCM 位域写入非零值来使能窗口比较器以及选择模式。

累加多个采样时，将在获取最后一个采样之后进行结果和阈值之间的比较。因此，标志仅在获取最后一个累加采样之后置 1 一次。

29.3.2.8. PTC 操作

外设触摸控制器（PTC）使能时将完全控制 ADC0。

PTC 禁止时，ADC0 可用作常规 ADC。

更多信息，请参见 *外设触摸控制器（PTC）* 部分。

29.3.3. 事件

如果事件控制（ADCn.EVCTRL）寄存器中的启动事件输入（STARTEI）位写入 1，则可以通过事件输入自动触发 ADC 转换。

当可从结果（ADCn.RES）寄存器读取新结果时，ADC 将生成结果就绪事件。该事件是一个长度为 1 个时钟周期的脉冲，由事件系统（EVSYS）处理。使能 ADC 时，将始终生成 ADC 结果就绪事件。

另请参见事件系统中的异步用户通道 n 输入选择（EVSYS.ASYNCUSERn）的说明。

29.3.4. 中断

表 29-1. 可用中断向量和中断源

名称	向量说明	条件
RESRDY	结果就绪中断	转换结果位于结果寄存器（ADCn.RES）中。
WCMP	窗口比较器中断	由 ADCn.CTRLE 中的 WINCM 定义

发生中断条件时，外设的中断标志（*外设.INTFLAGS*）寄存器中的相应中断标志将置 1。

可以通过写入外设的中断控制（*外设.INTCTRL*）寄存器中的相应允许位来允许或禁止中断源。

当相应的中断源被允许且中断标志置 1 时，将生成中断请求。中断请求保持有效，直到中断标志清零。有关如何清零中断标志的详细信息，请参见外设的 *INTFLAGS* 寄存器。

29.3.5. 休眠模式操作

在待机休眠模式下，ADC 默认处于禁止状态。

如果控制 A（ADCn.CTRLA）寄存器中的待机运行（RUNSTDBY）位写入 1，则 ADC 会在待机休眠模式下保持完全运行状态。

如果器件在 RUNSTDBY 为 1 时进入待机休眠模式，ADC 将保持工作状态，因此所有正在进行的转换都将完成，中断将按配置执行。

在待机休眠模式下，必须通过事件系统（EVSYS）触发 ADC 转换，或者 ADC 必须处于自由运行模式，并且在进入休眠状态前通过软件触发第一次转换。必要时，可以请求外设时钟，并在转换完成后关闭该时钟。

当输入事件触发信号出现时，将检测到正边沿，命令（ADCn.COMMAND）寄存器中的启动转换（STCONV）位将被置 1，转换随即启动。转换完成后，中断标志（ADCn.INTFLAGS）寄存器中的结果就绪（RESRDY）标志置 1，ADCn.COMMAND 中的 STCONV 位清零。

在待机休眠模式下激活参考电压源和电源基础结构时，需要经过一段时间才能达到稳定。将非零值写入控制 D（ADCn.CTRLD）寄存器中的初始延时（INITDLY）位可以配置第一次转换的启动延时。

在掉电休眠模式下，无法进行任何转换。所有正在进行的转换都将暂停，并在退出休眠模式时恢复。转换结束时，结果就绪（RESRDY）标志置 1，但由于 ADC 在转换过程中处于停止状态，因此结果（ADCn.RES）寄存器的内容无效。

29.4. 寄存器汇总——ADCn

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTBY					RESSEL	FREERUN	ENABLE
0x01	CTRLB	7:0						SAMPNUM[2:0]		
0x02	CTRLC	7:0		SAMPCAP	REFSEL[1:0]			PRESC[2:0]		
0x03	CTRLD	7:0		INITDLY[2:0]		ASDV		SAMPDLY[3:0]		
0x04	CTRL E	7:0						WINCM[2:0]		
0x05	SAMPCTRL	7:0					SAMPLEN[4:0]			
0x06	MUXPOS	7:0					MUXPOS[4:0]			
0x07	保留									
0x08	COMMAND	7:0								STCONV
0x09	EVCTRL	7:0								STARTEI
0x0A	INTCTRL	7:0							WCMP	RESRDY
0x0B	INTFLAGS	7:0							WCMP	RESRDY
0x0C	DBGCTRL	7:0								DBGRUN
0x0D	TEMP	7:0	TEMP[7:0]							
0x0E	保留									
...										
0x0F										
0x10	RES	7:0	RES[7:0]							
		15:8	RES[15:8]							
0x12	WINLT	7:0	WINLT[7:0]							
		15:8	WINLT[15:8]							
0x14	WINHT	7:0	WINHT[7:0]							
		15:8	WINHT[15:8]							
0x16	CALIB	7:0								DUTYCYC

29.5. 寄存器说明

29.5.1. 控制 A

名称: CTRLA
偏移量: 0x00
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	RUNSTBY					RESSEL	FREERUN	ENABLE
访问	R/W					R/W	R/W	R/W
复位	0					0	0	0

Bit 7 – RUNSTBY 在待机模式下运行

该位决定当芯片处于待机休眠模式时 ADC 是否需要运行。

Bit 2 – RESSEL 分辨率选择

该位选择 ADC 分辨率。

值	说明
0	完整的 10 位分辨率。10 位 ADC 结果累加或存储到 ADC 结果 (ADCn.RES) 寄存器中。
1	8 位分辨率。转换结果累加或存储到 ADC 结果 (ADCn.RES) 寄存器之前会截断为 8 位 (MSb)。低 2 位 (LSb) 将被丢弃。

Bit 1 – FREERUN 自由运行

向该位写入 1 将使能自由运行模式以进行数据采集。通过向 ADCn.COMMAND 中的 STCONV 位写入 1 来启动第一次转换。在自由运行模式下，新的转换周期在上一个转换周期完成之后立即启动。这由 ADCn.INTFLAGS 寄存器中的 RESRDY 标志指示。

Bit 0 – ENABLE ADC 使能

值	说明
0	禁止 ADC
1	使能 ADC

29.5.2. 控制 B

名称: CTRLB
偏移量: 0x01
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
						SAMPNUM[2:0]		
访问						R/W	R/W	R/W
复位						0	0	0

Bit 2:0 - SAMPNUM[2:0] 采样累加数选择

该位域选择自动累加多少个连续的 ADC 采样结果。当该位写入大于 0x0 的值时，会将相应数量的连续 ADC 采样结果在一次完整转换中累加到 ADC 结果（ADCn.RES）寄存器中。

值	名称	说明
0x0	NONE	无累加。
0x1	ACC2	累加 2 个结果
0x2	ACC4	累加 4 个结果
0x3	ACC8	累加 8 个结果
0x4	ACC16	累加 16 个结果
0x5	ACC32	累加 32 个结果
0x6	ACC64	累加 64 个结果
0x7	-	保留

29.5.3. 控制 C

名称: CTRLC
偏移量: 0x02
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
		SAMPCAP	REFSEL[1:0]			PRESC[2:0]		
访问	R	R/W	R/W	R/W	R	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 6 – SAMPCAP 采样电容选择

该位选择采样电容，从而选择输入阻抗。最佳值取决于参考电压和应用的电气特性。

值	说明
0	推荐用于 1V 以下的参考电压值
1	减小采样电容的大小。推荐用于较高的参考电压。

Bit 5:4 – REFSEL[1:0] 参考电压选择

该位域选择 ADC 的参考电压。

值	名称	说明
0x0	INTERNAL	内部参考电压
0x1	VDD	V _{DD}
0x2	VREFA	外部参考电压 V _{REFA}
其他	-	保留

Bit 2:0 – PRESC[2:0] 预分频比

该位域定义从外设时钟（CLK_PER）到 ADC 时钟（CLK_ADC）的分频系数。

值	名称	说明
0x0	DIV2	CLK_PER 进行 2 分频
0x1	DIV4	CLK_PER 进行 4 分频
0x2	DIV8	CLK_PER 进行 8 分频
0x3	DIV16	CLK_PER 进行 16 分频
0x4	DIV32	CLK_PER 进行 32 分频
0x5	DIV64	CLK_PER 进行 64 分频
0x6	DIV128	CLK_PER 进行 128 分频
0x7	DIV256	CLK_PER 进行 256 分频

29.5.4. 控制 D

名称: CTRLD
偏移量: 0x03
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	INITDLY[2:0]			ASDV	SAMPDLY[3:0]			
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:5 - INITDLY[2:0] 初始化延时

该位域定义使能 ADC 或切换为内部参考电压时第一次采样之前的初始化/启动延时。设置该延时将确保在启动第一次转换之前参考电压和多路开关等已就绪。从深度休眠模式唤醒进行测量时，也会出现初始化延时。

延时用多个 CLK_ADC 周期来表示。

值	名称	说明
0x0	DLY0	0 个 CLK_ADC 周期的延时
0x1	DLY16	16 个 CLK_ADC 周期的延时
0x2	DLY32	32 个 CLK_ADC 周期的延时
0x3	DLY64	64 个 CLK_ADC 周期的延时
0x4	DLY128	128 个 CLK_ADC 周期的延时
0x5	DLY256	256 个 CLK_ADC 周期的延时
其他	-	保留

Bit 4 - ASDV 自动采样延时变化

向该位写入 1 可在 ADC 转换之间使能自动采样延时变化。改变采样时刻的目的是使采样时刻随机化，从而避免频谱中出现驻波频率分量。SAMPDLY 位域的值在每次采样后自动递增 1。

当自动采样延时变化使能并且 SAMPDLY 值达到 0xF 时，它将折回到 0x0。

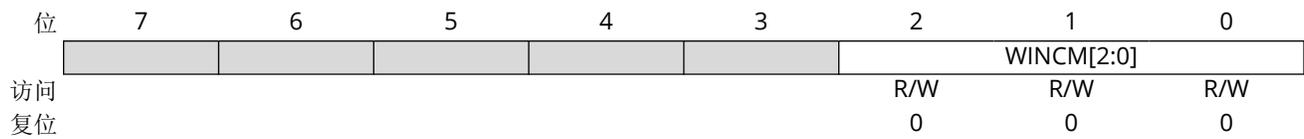
值	名称	说明
0	ASVOFF	禁止自动采样延时变化
1	ASVON	使能自动采样延时变化

Bit 3:0 - SAMPDLY[3:0] 采样延时选择

该位域定义连续 ADC 采样之间的延时。可编程采样延时允许修改硬件累加期间的采样频率，以抑制可能干扰采样的周期性噪声源。此外，也可通过置 1 ASDV 位将 SAMPDLY 位域从一个采样周期自动修改为其他值。延时用 CLK_ADC 周期表示，并通过位域设置直接给出。采样电容在延时期间保持断开。

29.5.5. 控制 E

名称: CTRL E
偏移量: 0x4
复位: 0x00
属性: -



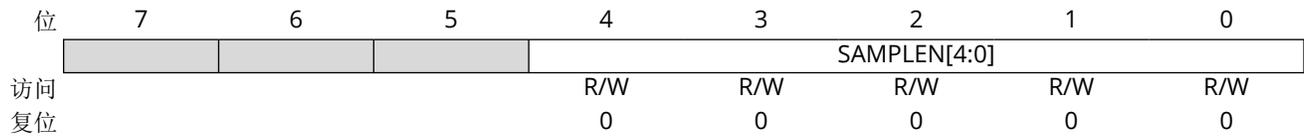
Bit 2:0 - WINCM[2:0] 窗口比较器模式

该位域用于允许并定义在窗口比较器模式下中断标志何时置 1。RESULT 是 16 位累加器结果。WINLT 和 WINHT 分别是 16 位下限阈值和 16 位上限阈值。

值	名称	说明
0x0	NONE	无窗口比较（默认）
0x1	BELOW	$RESULT < WINLT$
0x2	ABOVE	$RESULT > WINHT$
0x3	INSIDE	$WINLT < RESULT < WINHT$
0x4	OUTSIDE	$RESULT < WINLT$ 或 $RESULT > WINHT$
其他	-	保留

29.5.6. 采样控制

名称: SAMPCTRL
偏移量: 0x5
复位: 0x00
属性: -



Bit 4:0 - SAMPLEN[4:0] 采样长度

该位域可扩展 ADC 采样长度（多个 CLK_ADC 周期）。默认情况下，采样时间为两个 CLK_ADC 周期。增加采样长度可支持更高阻抗的采样源。总转换时间随选定的采样长度而增加。

29.5.7. MUXPOS

名称: MUXPOS
偏移量: 0x06
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	MUXPOS[4:0]							
访问				R/W	R/W	R/W	R/W	R/W
复位				0	0	0	0	0

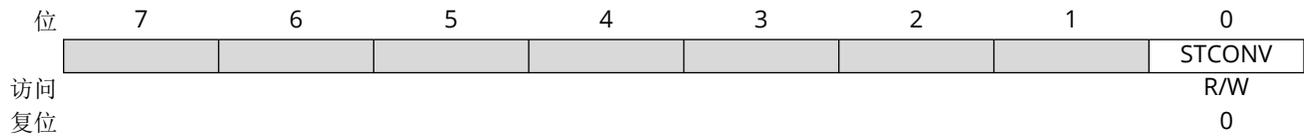
Bit 4:0 - MUXPOS[4:0] MUXPOS

该位域选择将哪个单端模拟输入连接到 ADC。如果这些位在转换期间发生更改，则更改将在此转换完成后生效。

值	名称	说明
0x00	AIN0	ADC 输入引脚 0
0x01	AIN1	ADC 输入引脚 1
0x02	AIN2	ADC 输入引脚 2
0x03	AIN3	ADC 输入引脚 3
0x04	AIN4	ADC 输入引脚 4
0x05	AIN5	ADC 输入引脚 5
0x06	AIN6	ADC 输入引脚 6
0x07	AIN7	ADC 输入引脚 7
0x08	AIN8	ADC 输入引脚 8
0x09	AIN9	ADC 输入引脚 9
0x0A	AIN10	ADC 输入引脚 10
0x0B	AIN11	ADC 输入引脚 11
0x1C	DAC0	DAC0
0x1D	INTREF	内部参考 (来自 VREF 外设)
0x1E	TEMPSENSE	温度传感器
0x1F	GND	0V (GND)
其他	-	保留

29.5.8. 命令

名称: COMMAND
偏移量: 0x08
复位: 0x00
属性: -

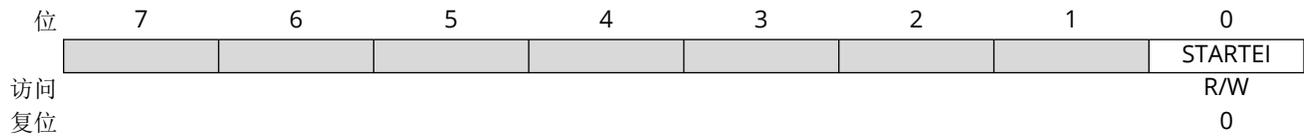


Bit 0 - STCONV 启动转换

向该位写 1 将启动单次测量。如果处于自由运行模式，这将启动第一次转换。只要转换正在进行，STCONV 就会读为 1。转换完成后，该位自动清零。

29.5.9. 事件控制

名称: EVCTRL
偏移量: 0x09
复位: 0x00
属性: -



Bit 0 - STARTEI 启动事件输入
通过该位，可以将事件输入作为启动转换的触发信号。

29.5.10. 中断控制

名称: INTCTRL
偏移量: 0x0A
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
							WCMP	RESRDY
访问							R/W	R/W
复位							0	0

Bit 1 - WCMP 窗口比较器中断允许
向该位写入 1 可允许窗口比较器中断。

Bit 0 - RESRDY 结果就绪中断允许
向该位写入 1 可允许结果就绪中断。

29.5.11. 中断标志

名称: INTFLAGS
偏移量: 0x0B
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
							WCMP	RESRDY
访问							R/W	R/W
复位							0	0

Bit 1 - WCMP 窗口比较器中断标志

当测量完成并且结果与 WINCM (ADCn.CTRLE) 定义的所选窗口比较器模式匹配时, 该窗口比较器标志置 1。比较在转换结束时完成。向位存储单元写入 1 或读取结果 (ADCn.RES) 寄存器均可将该标志清零。向该位写入 0 将不起作用。

Bit 0 - RESRDY 结果就绪中断标志

当测量完成并且新结果就绪时, 结果就绪中断标志置 1。向位存储单元写入 1 或读取结果 (ADCn.RES) 寄存器均可将该标志清零。向该位写入 0 将不起作用。

29.5.12. 调试运行

名称: DBGCTRL
偏移量: 0x0C
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
								DBGRUN
访问								R/W
复位								0

Bit 0 - DBGRUN 调试运行

值	说明
0	外设在中断调试模式下停止并忽略事件
1	当 CPU 停止时，外设将继续运行在中断调试模式下

29.5.13. 临时

名称: TEMP
偏移量: 0x0D
复位: 0x00
属性: -

临时寄存器供 CPU 用于对该外设的 16 位寄存器进行单周期 16 位访问。该寄存器由该外设的所有 16 位寄存器共用，可通过软件进行读写。有关读写 16 位寄存器的更多详细信息，请参见 [访问 16 位寄存器](#)。

位	7	6	5	4	3	2	1	0
	TEMP[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - TEMP[7:0] 临时

用于对 16 位寄存器执行读/写操作的临时寄存器。

29.5.14. 结果

名称: RES
偏移量: 0x10
复位: 0x00
属性: -

ADCn.RESL 和 ADCn.RESH 寄存器对代表 16 位值 ADCn.RES。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

如果模拟输入高于 ADC 的参考电压，则 10 位 ADC 结果将等于最大值 0x3FF。同样，如果输入低于 0V，ADC 结果将为 0x000。由于 ADC 无法产生高于值 0x3FF 的结果，即使在达到允许的最多累加次数 64 次之后，累加值也不会超过 0xFFC0。

位	15	14	13	12	11	10	9	8
	RES[15:8]							
访问	R	R	R	R	R	R	R	R
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	RES[7:0]							
访问	R	R	R	R	R	R	R	R
复位	0	0	0	0	0	0	0	0

Bit 15:8 - RES[15:8] 结果高字节

该位域构成 ADCn.RES 寄存器的 MSB，其中 MSb 为 RES[15]。ADC 本身具有 10 位输出 ADC[9:0]，其中 MSb 为 ADC[9]。ADC 和数字累加中的数据格式是二进制补码，其中 0x0000 表示零，0xFFFF 表示最大数字（满量程）。

Bit 7:0 - RES[7:0] 结果低字节

该位域构成 ADC/累加器结果（ADCn.RES）寄存器的 LSB。ADC 和数字累加中的数据格式是二进制补码，其中 0x0000 表示零，0xFFFF 表示最大数字（满量程）。

29.5.15. 窗口比较器下限阈值

名称: WINLT
偏移量: 0x12
复位: 0x00
属性: -

该寄存器是监视 ADCn.RES 寄存器的数字比较器的 16 位下限阈值。ADC 本身具有 10 位输出 RES[9:0]，其中 MSb 为 RES[9]。ADC 和数字累加中的数据格式是二进制补码，其中 0x0000 表示零，0xFFFF 表示最大数字（满量程）。

ADCn.WINLTH 和 ADCn.WINLTL 寄存器对代表 16 位值 ADCn.WINLT。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

累加采样时，窗口比较器阈值应用于累加值而不是每个采样。

位	15	14	13	12	11	10	9	8
	WINLT[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	WINLT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:8 - WINLT[15:8] 窗口比较器下限阈值高字节
该位域保存 16 位寄存器的 MSB。

Bit 7:0 - WINLT[7:0] 窗口比较器下限阈值低字节
该位域保存 16 位寄存器的 LSB。

29.5.16. 窗口比较器上限阈值

名称: WINHT
偏移量: 0x14
复位: 0x00
属性: -

该寄存器是监视 ADCn.RES 寄存器的数字比较器的 16 位上限阈值。ADC 本身具有 10 位输出 RES[9:0]，其中 MSb 为 RES[9]。ADC 和数字累加中的数据格式是二进制补码，其中 0x0000 表示零，0xFFFF 表示最大数字（满量程）。

ADCn.WINHHTH 和 ADCn.WINHTL 寄存器对代表 16 位值 ADCn.WINHT。低字节[7:0]（后缀 L）可通过原始偏移地址来访问。高字节[15:8]（后缀 H）可通过偏移地址 + 0x01 来访问。

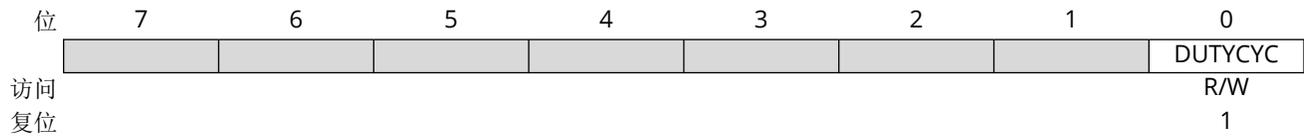
位	15	14	13	12	11	10	9	8
	WINHT[15:8]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0
位	7	6	5	4	3	2	1	0
	WINHT[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 15:8 - WINHT[15:8] 窗口比较器上限阈值高字节
该位域保存 16 位寄存器的 MSB。

Bit 7:0 - WINHT[7:0] 窗口比较器上限阈值低字节
该位域保存 16 位寄存器的 LSB。

29.5.17. 校准

名称: CALIB
偏移量: 0x16
复位: 0x01
属性: -



Bit 0 - DUTYCYC 占空比

该位决定 ADC 时钟的占空比。

ADC_{clk} > 1.5 MHz 时要求最低工作电压为 2.7V。

值	说明
0	如果 ADC _{clk} > 1.5 MHz, 必须使用 50% 的占空比
1	如果 ADC _{clk} ≤ 1.5 MHz, 必须使用 25% 的占空比 (高电平占 25%, 低电平占 75%)

30. DAC——数模转换器

30.1. 特性

- 8 位分辨率
- 最高 350 ksp/s 的转换速率
- 高驱动能力 (DAC0)
- 用作模拟比较器 (AC) 或模数转换器 (ADC) 的输入

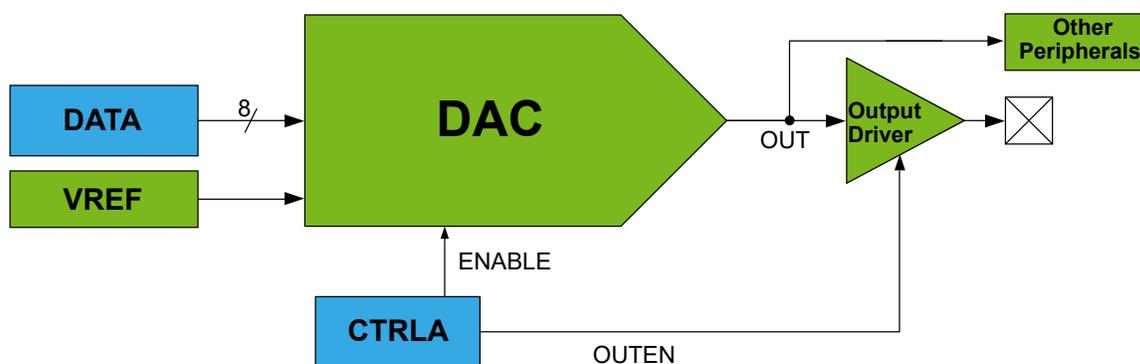
30.2. 概述

数模转换器 (DAC) 用于将写入数据 (DACn.DATA) 寄存器的数字值转换为模拟电压。转换范围介于 GND 与参考电压 (VREF) 外设中所选的参考电压之间。

DAC 为 8 位电阻串型 DAC，每秒可转换 350,000 个采样 (350 ksp/s)。DAC 使用内部参考电压 (VREF) 作为转换上限，并且具有一个驱动能力较高的连续时间输出，能够驱动 5 k Ω 和/或 30 pF 负载。通过写入数据 (DACn.DATA) 寄存器，可从应用程序启动 DAC 转换。

30.2.1. 框图

图 30-1. DAC 框图



注：仅 DAC0 具有一个用于外部引脚的输出驱动器。

30.2.2. 信号说明

信号	说明	类型
OUT	DAC 输出	模拟

30.2.3. 系统相关性

要使用此外设，必须按如下所述正确配置系统的其他部分。

表 30-1. DAC 系统相关性

相关性	适用	外设
时钟	是	CLKCTRL
I/O 线路和连接	是	PORT
中断	否	-
事件	否	-
调试	是	UPDI

30.2.3.1. 时钟

该外设取决于外设时钟。

30.2.3.2. I/O 线路和连接

使用外设的 I/O 线路需要配置 I/O 引脚。

表 30-2. I/O 线

实例	信号	I/O 线路	外设功能
DAC0	OUT	PA6	A

DAC0 具有一个模拟输出引脚（OUT），必须先配置才能使用。

DAC 还在内部连接到 AC 和 ADC。要将此内部 OUT 用作输入，输出和输入必须在其相应寄存器中进行配置。

30.2.3.3. 调试操作

进入调试模式不会影响此外设。

如果外配置为需要由 CPU 通过中断或类似操作进行定期控制，则在调试期间停止 CPU 可能会产生不正确的操作或丢失数据。

30.3. 功能说明

30.3.1. 初始化

要使 DAC 正常工作，请按照以下步骤操作：

1. 写入控制 x（VREF.CTRLx）寄存器中的 DAC 和 AC 参考电压选择（DACnREFSEL）位域，选择参考电压（VREF）外设中的 DAC 参考电压。
2. 转换范围介于 GND 与参考电压（VREF）外设中所选的参考电压之间。
3. 通过配置进一步使用 DAC 输出：
 - a. 配置内部外设（例如 AC 和 ADC）以使用 DAC 输出。请参见相应外设的文档。
 - b. 向控制 A（DACn.CTRLA）寄存器中的输出使能（OUTEN）位写入 1，使能引脚输出。这需要配置端口外设。

对于 DAC0，两个选项中至少有一个有效。DAC 的其他实例仅支持内部信号传输。

4. 向数据（DACn.DATA）寄存器中写入初始数字值。
5. 向 DACn.CTRLA 寄存器中的 ENABLE 位写入 1，以使能 DAC。

30.3.2. 工作原理

30.3.2.1. 使能、禁止和复位

通过向控制 A（DACn.CTRLA）寄存器中的 ENABLE 位写入 1 来使能 DAC，通过向该位写入 0 来禁止 DAC。

通过写入 DACn.CTRLA 寄存器中的输出使能（OUTEN）位来使能引脚的 OUT 输出。

30.3.2.2. 启动转换

使能 DAC（DACn.CTRLA 中的 ENABLE = 1）时，只要写入数据（DACn.DATA）寄存器，就会启动转换。

禁止 DAC（DACn.CTRLA 中的 ENABLE = 0）时，对 DACn.DATA 寄存器进行写操作不会触发转换，而向 DACn.CTRLA 寄存器中的 ENABLE 位写入 1 才会启动转换。

30.3.2.3. DAC 作为内部外设的输入源

DAC 的模拟输出在内部连接到 AC 和 ADC，当 DAC 使能（DACn.CTRLA 中的 ENABLE = 1）时，可用于这些外设。DAC 模拟输出仅在内部使用时，无需使能引脚输出驱动器（即，可将 DACn.CTRLA 中的 OUTEN = 0）。

30.3.3. 休眠模式操作

如果向控制 A（DACn.CTRLA）寄存器中的待机运行（RUNSTDBY）位写入 1 且 CLK_PER 可用，则 DAC 将在待机休眠模式下继续工作。如果 RUNSTDBY 位为 0，则 DAC 将在待机休眠模式下停止转换。

如果转换在待机休眠模式下停止，则 DAC 和输出缓冲器将被禁止以降低功耗。当器件退出待机休眠模式时，DAC 和输出缓冲器（如果已在 DACn.CTRLA 中配置 OUTEN = 1）将再次使能。因此，需要一定的启动时间才能启动新的转换。

在掉电休眠模式下，DAC 和输出缓冲器会被禁止以降低功耗。

30.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY	OUTEN						ENABLE
0x01	DATA	7:0	DATA[7:0]							

30.5. 寄存器说明

30.5.1. 控制 A

名称: CTRLA
 偏移量: 0x00
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	RUNSTDBY	OUTEN						ENABLE
访问	R/W	R/W						R/W
复位	0	0						0

Bit 7 - RUNSTDBY 在待机模式下运行

如果向该位写入 1，则当器件进入待机休眠模式时，DAC 或输出缓冲器不会自动禁止。

Bit 6 - OUTEN 输出缓冲器使能

向该位写入 1 会使能输出缓冲器并将 OUT 信号发送到引脚。

Bit 0 - ENABLE DAC 使能

向该位写入 1 会使能 DAC。

30.5.2. DATA

名称: DATA
 偏移量: 0x01
 复位: 0x00
 属性: -

位	7	6	5	4	3	2	1	0
	DATA[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - DATA[7:0] 数据

该位域包含数字数据，此类数据将转换为模拟电压。

31. PTC——外设触摸控制器

31.1. 概述

外设触摸控制器（PTC）通过采集信号来检测电容式传感器上的触摸。外部电容式触摸传感器通常内置在印刷电路板（PCB）布线上，传感器电极通过器件中的 I/O 引脚连接到 PTC 的模拟前端。PTC 支持自电容和互电容传感器。

在互电容模式下，使用各种 X-Y 配置的电容式触摸矩阵进行检测，其中包括氧化铟锡（Indium Tin Oxide, ITO）传感器网络。PTC 的每条 X 线路和 Y 线路都需要一个引脚。

在自电容模式下，PTC 的每个触摸传感器只需要一个引脚（Y 线路）。

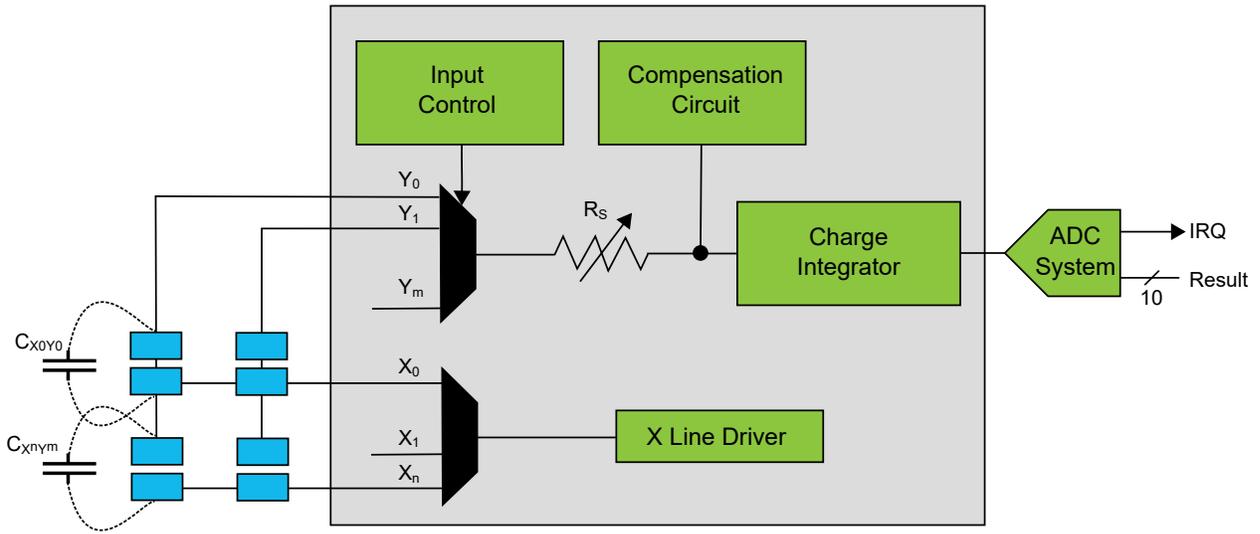
可用引脚数以及 X 线路和 Y 线路的分配取决于封装类型和器件配置。更多详细信息，请参见 [配置汇总](#) 一节和 [I/O 复用和注意事项](#) 一章。

31.2. 特性

- 稳健的电容式触摸按钮、滑动条和滚轮，低功耗、高灵敏度且不受环境影响
- 支持在待机休眠模式下通过触摸唤醒
- 支持互电容和自电容传感
 - 混合和匹配互电容和自电容传感器
- 每个电极一个引脚——无外部元件
- 负载补偿电荷检测
 - 寄生电容补偿和可调增益，可实现出色的灵敏度
- 整个温度和 V_{DD} 范围内的漂移均为零
 - 传感器自动校准和重新校准
- 单触发和自由运行电荷测量
- 硬件噪声滤波和噪声信号去同步化，可提高传导抗扰度
- 驱动屏蔽，可改善抗噪性和耐湿性
 - 任何 PTC X/Y 线路均可用于驱动屏蔽
 - 所有使能的传感器驱动时的电位与被扫描的传感器相同
- 可选通道变化延时，允许根据需要选择新通道的稳定时间
- 通过命令或自动触发功能触发采集-启动
- 采集完成时产生中断以减少 CPU 使用
- 使用 ADC 外设进行信号转换和采集

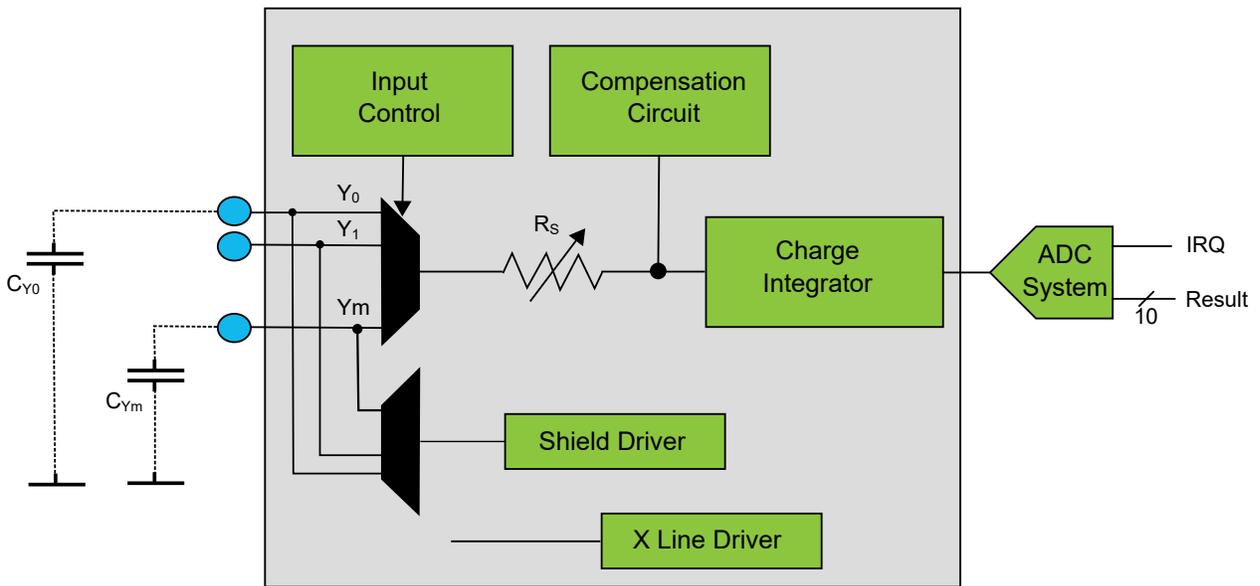
31.3. 框图

图 31-1. PTC 框图（互电容）



注：对于 CN0816/1616/3217， $R_s = 0、20、50、70、100$ 和 $200\text{ k}\Omega$ 。

图 31-2. PTC 框图（自电容）



31.4. 信号说明

表 31-1. PTC 的信号说明

名称	类型	说明
$Y[m:0]$	模拟	Y 线路（输入/输出）
$X[n:0]$	数字	X 线路（输出）

注：X 线路和 Y 线路的数量取决于器件。有关详细信息，请参见配置汇总。

有关该外设引脚映射的详细信息，请参见 I/O 复用和注意事项。可将一个信号映射到多个引脚上。

31.5. 系统相关性

要使用此外设，请按照以下各节中的说明配置系统的其他组件。

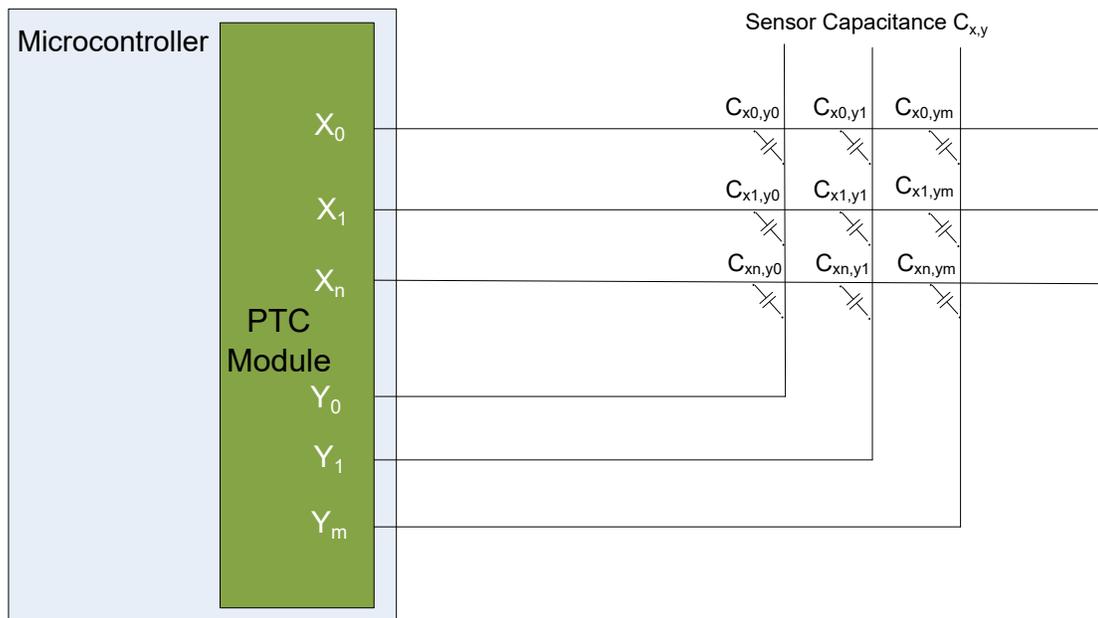
31.5.1. I/O 线

用于模拟 X 线路和 Y 线路的 I/O 线路必须连接到外部电容式触摸传感器电极。正常工作期间不需要外部元件。但是，为提高 EMC 性能，可在 X 线路和 Y 线路上串联一个不低于 $1\text{ k}\Omega$ 的电阻。

31.5.1.1. 互电容传感器配置

互电容传感器内置在印刷电路板（PCB）布线上的两个 I/O 线路之间——X 电极用于发送，Y 电极用于检测。X 和 Y 电极之间的互电容通过外设触摸控制器测量。

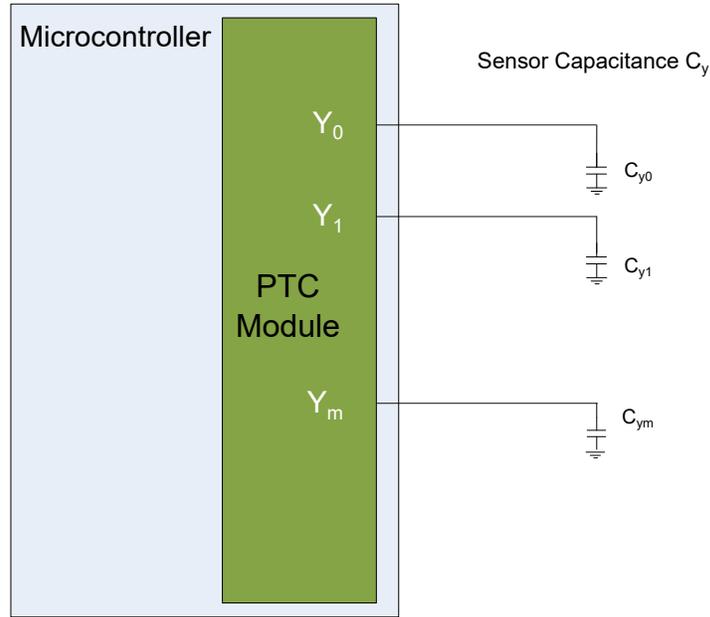
图 31-3. 互电容传感器配置



31.5.1.2. 自电容传感器配置

自电容传感器通过 Y 电极（用于检测信号）连接到外设触摸控制器上的单个引脚。检测电极电容由外设触摸控制器测量。

图 31-4. 自电容传感器配置



31.5.2. 时钟

PTC 的时钟源是 CLK_PER 时钟。有关配置 CLK_PER 的详细信息，请参见*时钟控制器 (CLKCTRL)* 一章。

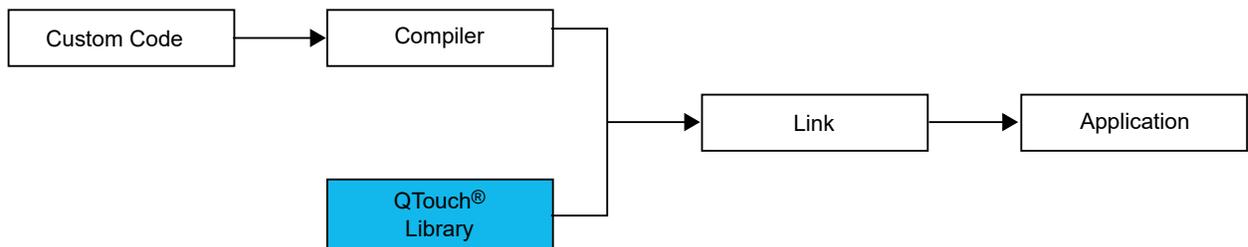
31.5.3. 模数转换器 (ADC)

PTC 使用 ADC 进行信号转换和采集。必须使能并正确配置 ADC，以确保 PTC 正常工作。更多详细信息，请参见*模数转换器 (ADC)* 一章。

31.6. 功能说明

要访问 PTC，用户必须使用 Atmel START QTouch®配置器来配置 QTouch 库固件并将其链接到应用软件。使用 QTouch 库，可以在单个界面中实现按钮、滑动条和滚轮的各种组合。

图 31-5. QTouch®库的使用



32. UPDI——统一编程和调试接口

32.1. 特性

- 用于外部编程和片上调试（OCD）的 UPDI 单线接口
 - 通过高电压或熔丝使能编程
 - 使用器件的 $\overline{\text{RESET}}$ 引脚进行编程
 - 操作期间未占用 GPIO 引脚
 - 为编程器应用异步半双工 UART 协议
- 编程：
 - 内置错误检测，可生成错误签名
 - 支持改写响应生成以加快编程速度
- 调试：
 - 针对器件地址空间（NVM、RAM 和 I/O）的存储器映射访问
 - 对器件时钟频率没有限制
 - 用户程序断点的数量不限
 - 两个硬件断点
 - 支持高级 OCD 特性
 - 运行时读取 CPU 程序计数器（PC）、堆栈指针（SP）和状态寄存器（SREG），以进行代码性能分析
 - 检测和指示 CPU 中的中断/停止条件
 - 针对运行、停止和复位调试指令的程序流控制
 - 无需访问系统寄存器即可进行非侵入式运行时芯片监视
 - 提供接口，用于读取被锁定器件上闪存的 CRC 校验结果

32.2. 概述

统一编程和调试接口（UPDI）为专用接口，用于器件的外部编程和 OCD。

UPDI 支持对非易失性存储器（Nonvolatile Memory, NVM）空间、闪存、EEPROM、熔丝、锁定位和用户行进行编程。某些映射到存储器的寄存器仅在使能正确的访问权限（密钥或锁定位）时在 OCD 停止模式或特定编程模式下才能访问。这些模式通过向 UPDI 发送正确的密钥来解锁。有关通过 NVM 控制器进行编程以及执行 NVM 控制器命令的信息，请参见 *NVMCTRL——非易失性存储器控制器* 一章。

UPDI 分为三个单独的协议层：UPDI 物理（PHY）层、UPDI 数据链路（DL）层和 UPDI 访问（ACC）层。默认 PHY 层通过 UPDI 引脚线路处理与连接的编程器/调试器之间的双向 UART 通信。此外，还在单线通信模式下对传入的数据帧提供数据恢复和时钟恢复。接收到的指令和相应的数据由 DL 层处理，该层根据解码后的指令与 ACC 层建立通信。通过 ACC 层授予对系统总线和映射到存储器的寄存器的访问权限。

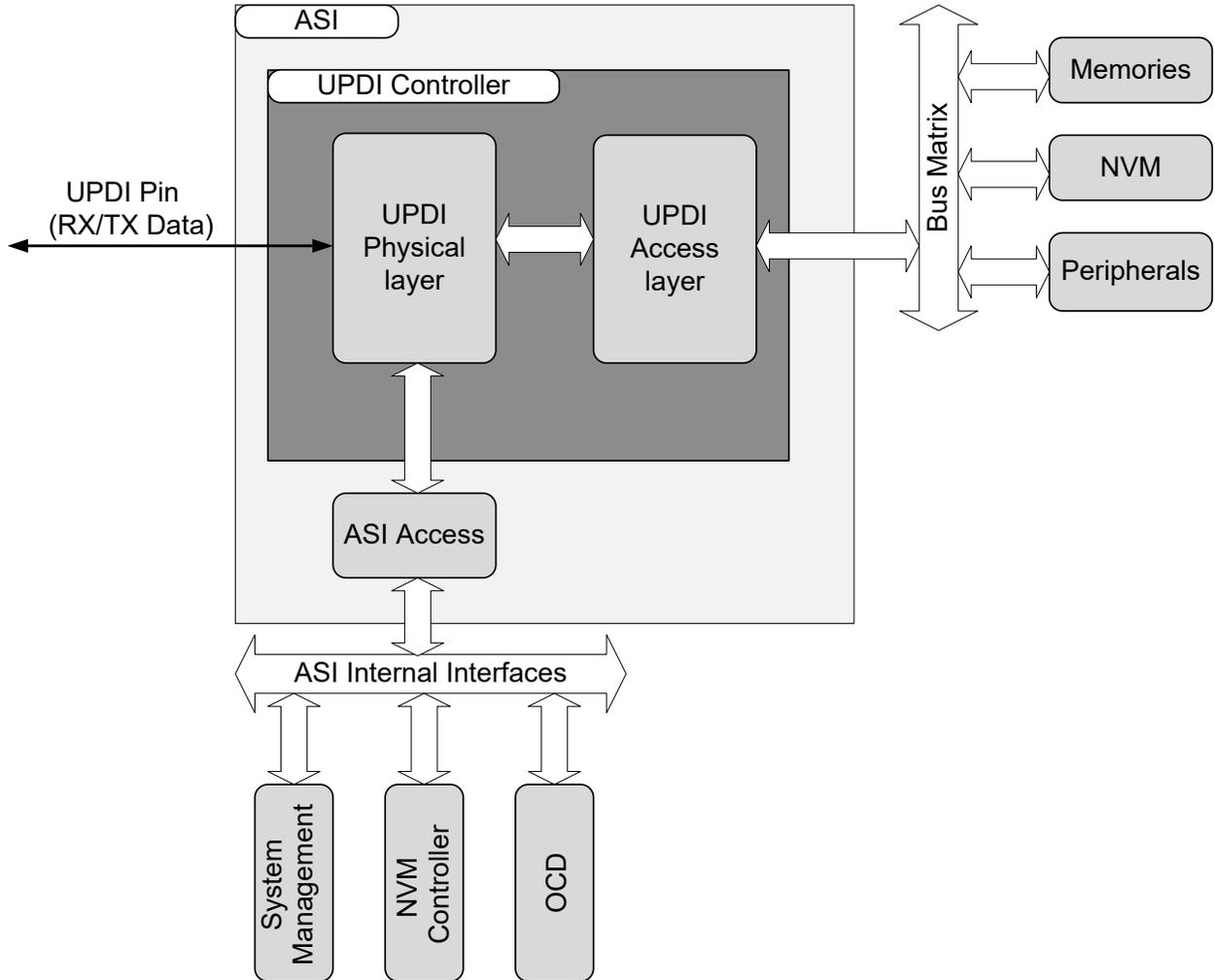
编程和调试通过 PHY 层完成，该层为基于半双工接口的单线 UART，使用 $\overline{\text{RESET}}$ 引脚进行数据接收和发送。专用内部振荡器为 PHY 层提供时钟。

ACC 层是 UPDI 与连接的总线矩阵之间的接口。该层通过 UPDI 接口授予对总线矩阵的访问权限，其中包含面向系统块（例如，存储器、NVM 和外设）的存储器映射访问。

异步系统接口（Asynchronous System Interface, ASI）允许直接访问 OCD、NVM 和系统管理系统。这样，调试器便可直接访问系统信息，而无需请求访问总线。

32.2.1. 框图

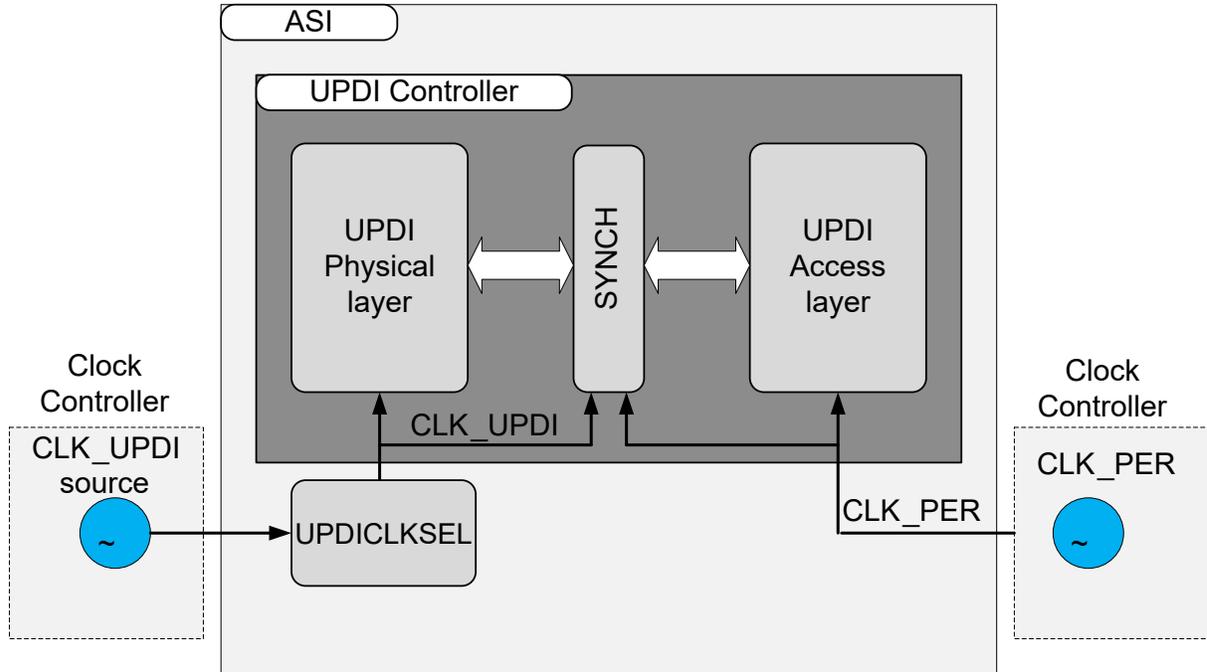
图 32-1. UPDI 框图



32.2.2. 时钟

PHY 层和 ACC 层可以在不同的时钟域上工作。PHY 层时钟源自专用内部振荡器，而 ACC 层时钟与外设时钟相同。PHY 和 ACC 层之间有同步边界，可确保时钟域之间正确工作。通过 ASI 选择 UPDI 时钟输出频率，使能或复位 UPDI 后，UPDI 时钟的默认启动频率为 4 MHz。可以通过写入 ASI 控制 A (UPDI.ASI_CTRLA) 寄存器中的 UPDI 时钟分频比选择 (UPDICKSEL) 位域来更改 UPDI 时钟频率。

图 32-2. UPDI 时钟域



32.2.3. 物理层

PHY 层是所连编程器/调试器与器件之间的通信接口。PHY 层的主要特性可以概括如下：

- 支持 UPDI 单线模式，在 UPDI 引脚上使用异步、半双工 UART 通信
- UART 帧上的内部波特率检测、时钟和数据恢复
- 错误检测（奇偶校验、时钟恢复、帧、系统错误）
- 生成发送响应（ACK）
- 在工作期间生成错误签名
- 保护时间控制

32.2.4. I/O 线路和连接

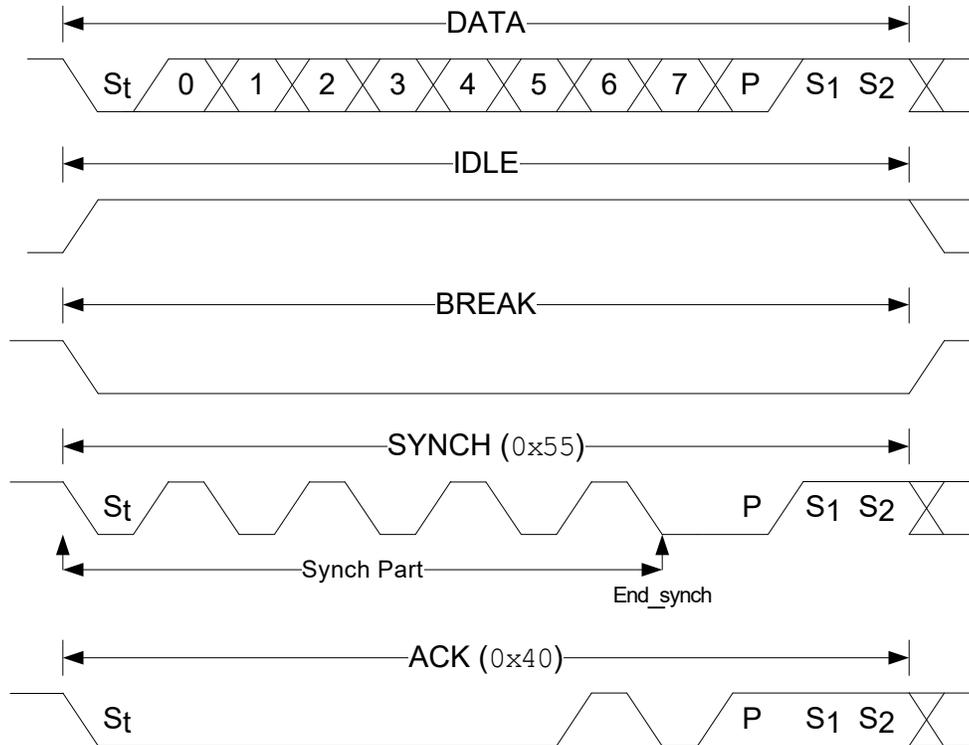
要使 UPDI 正常工作，必须将 $\overline{\text{RESET}}$ 引脚设置为 UPDI 模式。此模式并非像常规 I/O 引脚那样通过端口 I/O 引脚配置来实现，而是通过将 FUSE.SYSCFG0 中的 $\overline{\text{RESET}}$ 引脚配置（RSTPINCFG）位置 1（见[通过对 RESET 引脚应用熔丝改写脉冲来使能 UPDI](#)），或者执行[通过对 RESET 引脚施加高电压改写脉冲来使能 UPDI](#) 中的 UPDI 高电压使能序列来完成。上拉/下拉使能、输入使能和输出使能设置通过 UPDI（激活时）自动控制。

32.3. 功能说明

32.3.1. 工作原理

通过 UPDI 进行的通信基于标准 UART 通信，其采用固定帧格式，并支持自动波特率检测，以用于时钟和数据恢复。除数据帧外，以下几个控制帧对通信也非常重要：DATA、IDLE、BREAK、SYNCH 和 ACK。

图 32-3. 支持的 UPDI 帧格式



帧	说明
DATA	DATA 帧由一个始终为低电平的启动 (St) 位、八个数据位、一个用于偶校验的奇偶校验 (P) 位和两个始终为高电平的停止 (S1 和 S2) 位组成。如果奇偶校验位或停止位的值不正确, UPDI 将检测到错误并发出信号。通过写入控制 A (UPDI.CTRLA) 寄存器中的奇偶校验禁止 (PARD) 位, 可以禁止 UPDI 的奇偶校验位检查, 在这种情况下, 可以忽略通过调试器生成的奇偶校验位。
IDLE	IDLE 是由至少 12 个高电平位组成的特殊帧, 其作用与将传输线保持在空闲状态相同。
BREAK	BREAK 是由至少 12 个低电平位组成的特殊帧。该帧用于将 UPDI 复位回其默认状态, 其通常用于错误恢复。
SYNCH	波特率发生器使用 SYNCH 帧来设置后续发送操作的波特率。在成功发送 BREAK 之后, 在每个新指令之前, UPDI 始终需要一个 SYNCH 字符。
ACK	只要 ST 或 STS 指令成功越过同步边界并获得总线访问权限, 便会从 UPDI 发送 ACK 帧。当调试器接收到 ACK 时, 可启动下一次发送。

32.3.1.1. UPDI UART

通信由调试器/编程器端发起。每次传输必须以 SYNCH 字符开始, UPDI 可基于该字符恢复传输波特率, 并存储该设置以供传入数据使用。SYNCH 字符设置的波特率将用于发送和接收后续的指令和数据字节。有关指令流中何时出现下一个 SYNCH 字符的详细信息, 请参见 *UPDI 指令集* 一节。

UPDI 中没有可写入的波特率寄存器, 因此对数据字节进行采样时, 从 SYNCH 字符采样的波特率用于数据恢复。

PHY 层的传输波特率与所选的 UPDI 时钟有关, 该时钟可通过写入 ASI 控制 A (UPDI.ASI_CTRLA) 寄存器中的 UPDI 时钟分频比选择 (UPDICKSEL) 位域进行调整。在自动波特率的精度范围内, 接收波特率与发送波特率始终相同。最大 UPDI 时钟频率取决于器件的电源电压 V_{DD} , 更多详细信息, 请参见 ASI 控制 A (UPDI.ASI_CTRLA) 寄存器。

表 32-1. 基于 UPDICKSEL 设置建议的 UART 波特率

UPDICKSEL[1:0]	建议的最大波特率	建议的最小波特率
0x1 (16 MHz)	0.9 Mbps	0.300 kbps

表 32-1. 基于 UPDICKSEL 设置建议的 UART 波特率（续）

UPDICKSEL[1:0]	建议的最大波特率	建议的最小波特率
0x2 (8 MHz)	450 kbps	0.150 kbps
0x3 (4 MHz) ——默认值	225 kbps	0.075 kbps

UPDI 波特率发生器利用小数波特率计数来最大程度地减小传输误差。表 32-2 列出了 UPDI 使用固定帧格式时的最大误差限值和推荐的接收器传输误差限值。

表 32-2. 接收器波特率误差

数据 + 奇偶校验位	R _{slow}	R _{fast}	最大总误差[%]	建议的最大 RX 误差[%]
9	96.39	104.76	+4.76/-3.61	+1.5/-1.5

32.3.1.2. BREAK 字符

BREAK 字符用于将 UPDI 的内部状态复位为默认设置。如果 UPDI 由于通信错误而进入错误状态，或者在调试器与 UPDI 之间的同步丢失时进入错误状态，则此功能非常有用。

为了确保在所有情况下 UPDI 都能成功接收到 BREAK，调试器必须发送两个连续的 BREAK 字符。如果 UPDI 处于空闲状态，则会检测到第一个 BREAK 字符；如果发送第一个 BREAK 字符时 UPDI 正在执行接收或发送操作（以非常低的波特率），则不会检测到。但是，这将导致接收（RX）出现帧错误或发送（TX）出现争用错误，进而中止正在进行的操作。UPDI 随后将成功检测到下一个 BREAK 字符。

接收到 BREAK 字符后，ASI 控制 A（UPDI.ASI_CTRLA）寄存器中的 UPDI 振荡器设置会复位为 4 MHz（默认 UPDI 时钟选择），UPDI 的波特率范围基于上面的 *UPDICKSEL 设置建议的 UART 波特率表* 进行更改。

32.3.1.2.1. 单线模式下的 BREAK

在单线模式下，编程器/调试器和 UPDI 可能完全不同步，因此需要最差情况长度的 BREAK 字符来确保 UPDI 可以检测到它。假设 UPDI 最低时钟速度为 4 MHz（250 ns），则 16 位中可以包含的 8 位 SYNCH 模式值的最大长度为

$$65535 \times 250\text{ns} = 16.4\text{ms}/\text{字} \div 8 = 2.05\text{ms}/\text{位}。$$

对于最低预分频比设置，上式给出了最差情况下的 BREAK 帧持续时间： $2.05\text{ms} \times 12\text{bits} \approx 24.6\text{ms}$ 。已知预分频比设置时，可以根据表 32-3 中的值放宽 BREAK 帧的时间。

表 32-3. 建议的 BREAK 字符持续时间

UPDICKSEL[1:0]	建议的 BREAK 字符持续时间
0x1 (16 MHz)	6.15 ms
0x2 (8 MHz)	12.30 ms
0x3 (4 MHz)	24.60 ms

32.3.1.3. SYNCH 字符

SYNCH 字符共有 8 位，遵循常规的 UPDI 帧格式。其值固定为 0x55。SYNCH 字符有两个主要用途：

1. 在禁止 UPDI 后充当 UPDI 的使能字符。
2. 供波特率发生器用于设置后续传输的波特率。如果发送的 SYNCH 字符无效，则将无法对下一次传输进行正确采样。

32.3.1.3.1. 单线模式下的 SYNCH

在每条新指令之前都会使用 SYNCH 字符。使用 REPEAT 指令时，SYNCH 字符只会出现在 REPEAT 之后的第一条指令前。

SYNCH 是一个已知字符，其具有针对每个位切换的属性，这允许 UPDI 测量采样 8 位 SYNCH 模式时需要花费的 UPDI 时钟周期数。通过采样获取的信息可以在接收时提供异步时钟恢复和异步数据恢复，并可用于在执行发送操作时保持所连编程器的波特率。

32.3.2. 工作原理

必须先使能 UPDI，然后才能启动 UART 通信。

32.3.2.1. UPDI 使能

UPDI 的使能序列与器件无关，具体内容如以下各节所述。

32.3.2.1.1. 单线使能

UPDI 引脚具有内部上拉电阻，如果将 UPDI 引脚驱动为低电平的时间超过 200 ns，则连接的编程器将会发起启动序列。

负边沿跳变会导致边沿检测器（如果是在多电压系统中，则边沿检测器位于高压域）开始将 UPDI 引脚驱动为低电平，因此当编程器释放线路时，该引脚将保持低电平，直到请求的 UPDI 振荡器就绪。时钟的预期到达时间将取决于振荡器的精度、过冲和校准读数。对于多电压系统，线路将被驱动为低电平，直到稳压器处于正确的电压值，系统将在所选的振荡器就绪且稳定时上电。编程器必须在第一次释放 UPDI 引脚后对其进行轮询，以检测该引脚何时再次变为高电平。这种跳变意味着边沿检测器已释放引脚（上拉），并且 UPDI 可以接收 SYNCH 字符。成功检测到 SYNCH 字符后，UPDI 被使能并准备接收第一条指令。

使能发送序列如图 32-4 所示，其中包括编程器和边沿检测器的主动驱动周期。“UPDI 引脚”波形显示了任意给定时间的引脚值。

边沿检测器主动驱动周期的延时为典型启动时间（等待 256 个 32 MHz 振荡器周期 + 校准读数）。有关预期启动时间的详细信息，请参见 *电气特性* 一章。

注：在初始使能 SYNCH 后发出的第一条指令不需要发送额外的 SYNCH，因为使能序列 SYNCH 会为第一条指令设置波特率发生器。

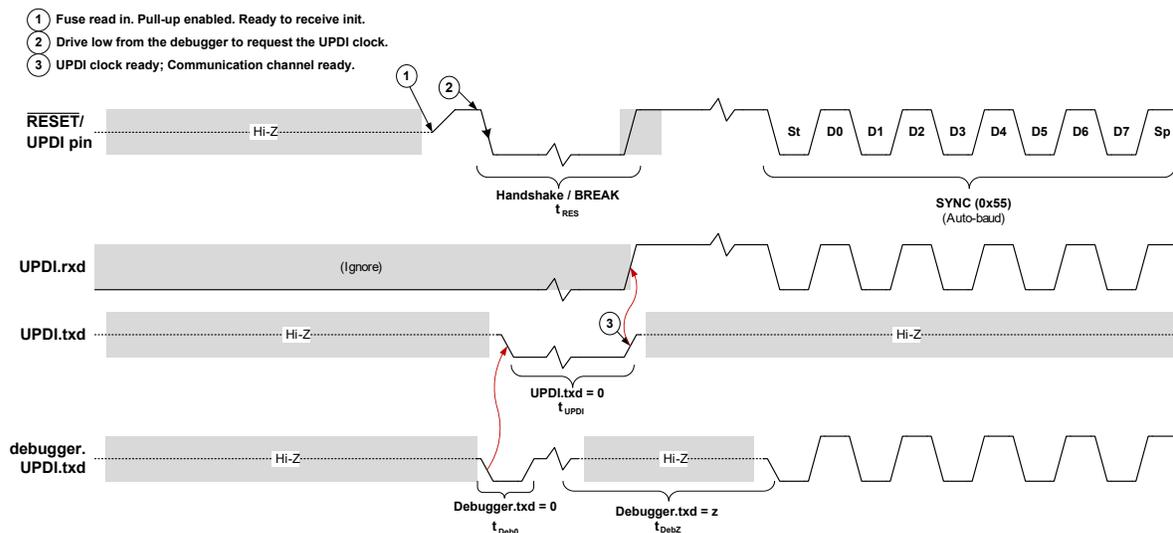
为避免 UPDI 在边沿检测器意外触发时保持使能状态，UPDI 将自动地自行禁止并降低其时钟请求的优先级。有关更多详细信息，请参见 *启动期间禁止* 一节。

通过对 $\overline{\text{RESET}}$ 引脚应用熔丝改写脉冲来使能 UPDI

当 FUSE.SYSCFG0 中的 $\overline{\text{RESET}}$ 引脚配置 (RSTPINCFG) 位为 0x1 时， $\overline{\text{RESET}}$ 引脚将被改写，UPDI 将控制引脚并将其配置为具有上拉的输入。

检测到上拉时，调试器会通过将线路驱动为低电平并持续 t_{Deb0} 的时间来启动使能序列，如图 32-4 所示：

图 32-4. 通过熔丝使能 UPDI 焊盘时的 UPDI 使能序列



当检测到负边沿时，UPDI 时钟启动。UPDI 会继续将线路驱动为低电平，直到时钟稳定，可供 UPDI 使用。使能 UPDI 时， t_{UPDI} 的持续时间将根据振荡器的状态变化。在这段持续时间之后，数据线将由 UPDI 释放并拉为高电平。

当调试器检测到线路为高电平时，必须发送初始 SYNCH 字符 0x55 以同步 UPDI 通信数据速率。如果 SYNCH 字符的启动位未在最大 t_{DebZ} 时间内发出，UPDI 将自行禁止，此时必须重新启动 UPDI 使能序列。如果违反该时序，则将禁止 UPDI 以避免其意外使能。

在成功发送同步字符后，便可以发送第一个指令帧。

通过对 \overline{RESET} 引脚施加高电压改写脉冲来使能 UPDI

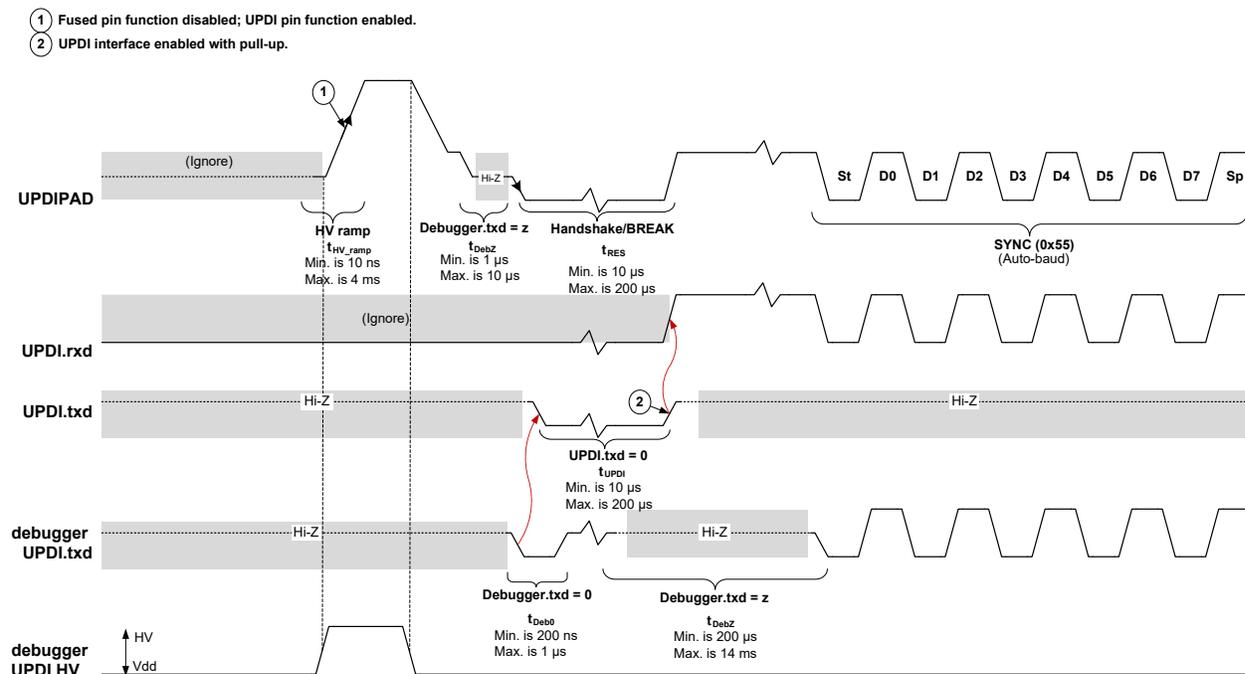
可通过高电压（HV）编程功能将 \overline{RESET} 引脚的 GPIO 或复位功能改写为 UPDI。对 \overline{RESET} 引脚施加 HV 脉冲会将引脚功能切换为 UPDI。这与 FUSE.SYSCFG0 中的 \overline{RESET} 引脚配置（RSTPINCFG）无关。要改写引脚功能，请按照以下步骤操作：

1. **推荐：**在启动 HV 使能序列之前复位器件。
2. 施加 HV 信号，如图 32-5 所述。
3. 在第一个 SYNC 字符后使用 key 指令发送 NVMPROG 密钥以开始编程。锁定的器件只接受 CHIPERASE 密钥。另请参见全片擦除一节。
4. 完成编程后，通过使用 STCS 指令向控制 B（UPDI.CTRLB）寄存器中的 UPDI 禁止（UPDIDIS）位写入 1 来复位 UPDI。

上电期间，必须先释放 \overline{RESET} 信号才能施加 HV 脉冲。在三态之前，建议脉冲持续时间在 100 μs 至 1 ms 的范围内。

当施加 HV 脉冲的上升沿时，UPDI 将复位。在三态之后，UPDI 将保持复位状态，直到 \overline{RESET} 引脚被调试器驱动为低电平。这将释放 UPDI 复位并启动相同的使能序列，具体参见[通过对 \$\overline{RESET}\$ 引脚应用熔丝改写脉冲来使能 UPDI](#)。

图 32-5. 通过高电压（HV）编程实现的 UPDI 使能序列



通过 HV 脉冲使能时，只有 POR 会禁止 $\overline{\text{RESET}}$ 引脚的 UPDI 配置并恢复默认设置。如果通过 UPDI.CTRLB 中的 UPDIDIS 位发出 UPDI 禁止命令，UPDI 将被复位并且时钟请求将被取消，但是 $\overline{\text{RESET}}$ 引脚仍保持 UPDI 配置。

注：

1. 如果向 UPDI 引脚添加的外部保护不足，则 ESD 脉冲可能被器件解释为高电压改写脉冲，然后使能 UPDI。
2. UPDI HV 激活的实际阈值电压取决于 V_{DD} 。更多详细信息，请参见 *电气特性* 一章。

GPIO 配置的输出使能定时器保护

当 FUSE.SYSCFG0 中的 $\overline{\text{RESET}}$ 引脚配置 (RSTPINCFG) 位为 0x0 时， $\overline{\text{RESET}}$ 引脚配置为 GPIO。为避免在 GPIO 主动驱动输出与 UPDI 高电压 (HV) 使能序列启动之间发生潜在的冲突，GPIO 输出驱动器将在系统复位后至少禁止 8.8 ms。

始终建议在进入 HV 编程序列之前发出上电复位 (POR)。

32.3.2.2. UPDI 禁止

32.3.2.2.1. 启动期间禁止

在使能序列期间，UPDI 可以在使能序列无效的情况下自行禁止。可通过两种机制来复位 UPDI 向电源管理模块发出的任何请求，并将 UPDI 设置为禁止状态。然后，必须启动新的使能序列来使能 UPDI。

超时禁止

当启动负边沿检测器在 UPDI 接收到其时钟后释放引脚时，或者当多电压系统中的稳压器稳定且系统已上电时，默认上拉会将 UPDI 引脚驱动为高电平。如果编程器未检测到引脚为高电平并且未在 UPDI 释放引脚后的 16.4 ms 内 (4 MHz UPDI 时钟) 发送 SYNCH 字符，则 UPDI 将自行禁止。

注：启动振荡器频率取决于器件。UPDI 将在发出超时前计数 65536 个 UPDI 时钟周期。

错误的 SYNCH 模式

如果 SYNCH 字符的长度大于 UPDI 波特率寄存器中可包含的采样数 (溢出) 或者小于针对每个位的采样长度可以处理的最小小数计数，则将检测到错误的 SYNCH 模式。只要检测到上述任一错误，UPDI 便会自行禁止。

32.3.2.2.2. UPDI 常规禁止

断开编程器后，任何无需通过 UPDI 进行特定操作的编程或调试会话都必须通过写入控制 B (UPDI.CTRLB) 寄存器中的 UPDI 禁止 (UPDIDIS) 位来终止，之后 UPDI 将发出系统复位并自行禁止。复位会将 CPU 恢复到运行状态，不受先前的状态影响。它同时也会降低对系统的 UPDI 时钟请求的优先级，并复位所有 UPDI 密钥和设置。

如果不执行禁止操作，则 UPDI 和振荡器请求将保持使能状态，这会导致应用的功耗增加。

32.3.2.3. UPDI 通信错误处理

UPDI 包含一个全面的错误检测系统，可在从错误情况中恢复时向调试器提供信息。错误检测包括检测物理传输错误 (如奇偶校验错误、争用错误和帧错误)，以及更高级别的错误 (如访问超时错误)。有关可用错误签名的概述，请参见状态 B (UPDI.STATUSB) 寄存器中的 UPDI 错误签名 (PESIG) 位域。

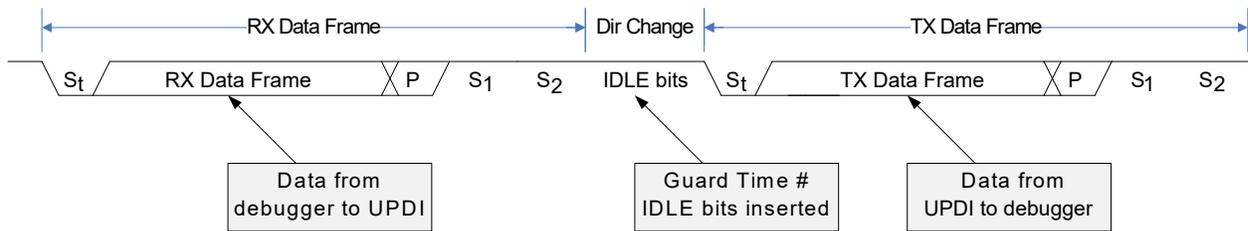
每当 UPDI 检测到错误时，将立即进入内部错误状态以避免发生不必要的系统通信。在错误状态下，UPDI 将忽略所有传入的数据请求 (接收 BREAK 字符除外)。从错误状态恢复时，必须始终遵循以下步骤。

1. 发送 BREAK 字符。有关建议的 BREAK 字符处理方法，请参见 *BREAK 字符* 一节。
2. 以下一次数据传输所需的波特率发送 SYNCH 字符。
3. 执行载入控制状态 (LDCS) 指令以读取状态 B (UPDI.STATUSB) 寄存器中的 UPDI 错误签名 (PESIG) 位域，并获取有关所发生错误的信息。
4. UPDI 现在从错误状态恢复，并准备接收下一个 SYNCH 字符和指令。

32.3.2.4. 方向改变

为了确保半双工 UART 操作采用正确的时序，UPDI 内置了保护时间机制，以便在方向从 RX 模式变为 TX 模式时放宽时序。保护时间用传输第一个响应字节的下一个启动位之前插入的空闲位数量表示。可通过控制 A (UPDI.CTRLA) 寄存器中的保护时间值 (GTVAL) 位域配置空闲位的数量。每个空闲位的持续时间由当前传输使用的波特率指定。

图 32-6. 通过插入空闲位改变 UPDI 方向



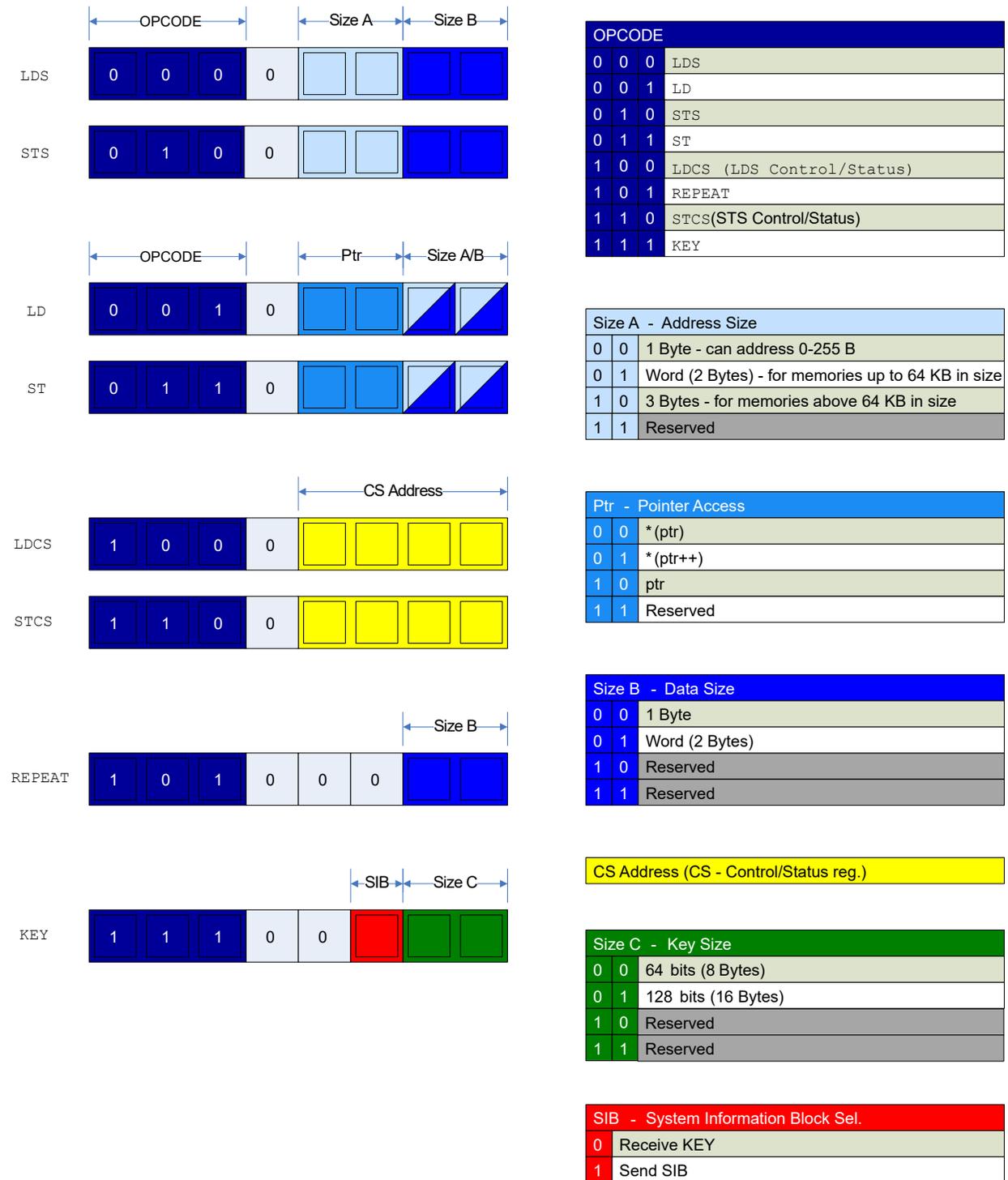
UPDI 保护时间是所连调试器在等待来自 UPDI 的数据时将经历的最短空闲时间。最长空闲时间与超时相同。当同步时间与数据总线访问时间之和大于保护时间时，传输前的空闲时间将大于预期保护时间。

建议始终在 UPDI 端插入至少两个保护时间位，并从调试器端插入一个保护时间周期。

32.3.3. UPDI 指令集

通过 UPDI 进行的通信基于精简指令集来实现。这些指令是 UPDI 数据链路 (DL) 层的一部分。指令用于访问 UPDI 寄存器，因为它们均已映射到称为“ASI 控制和状态 (CS) 空间”的内部存储空间以及存储器映射的系统空间。所有指令都是字节指令，必须在前面加上一个 SYNCH 字符以确定通信的波特率。有关设置传输波特率的信息，请参见 *UPDI UART* 一节。图 32-7 给出了 UPDI 指令集概览。

图 32-7. UPDI 指令集概览

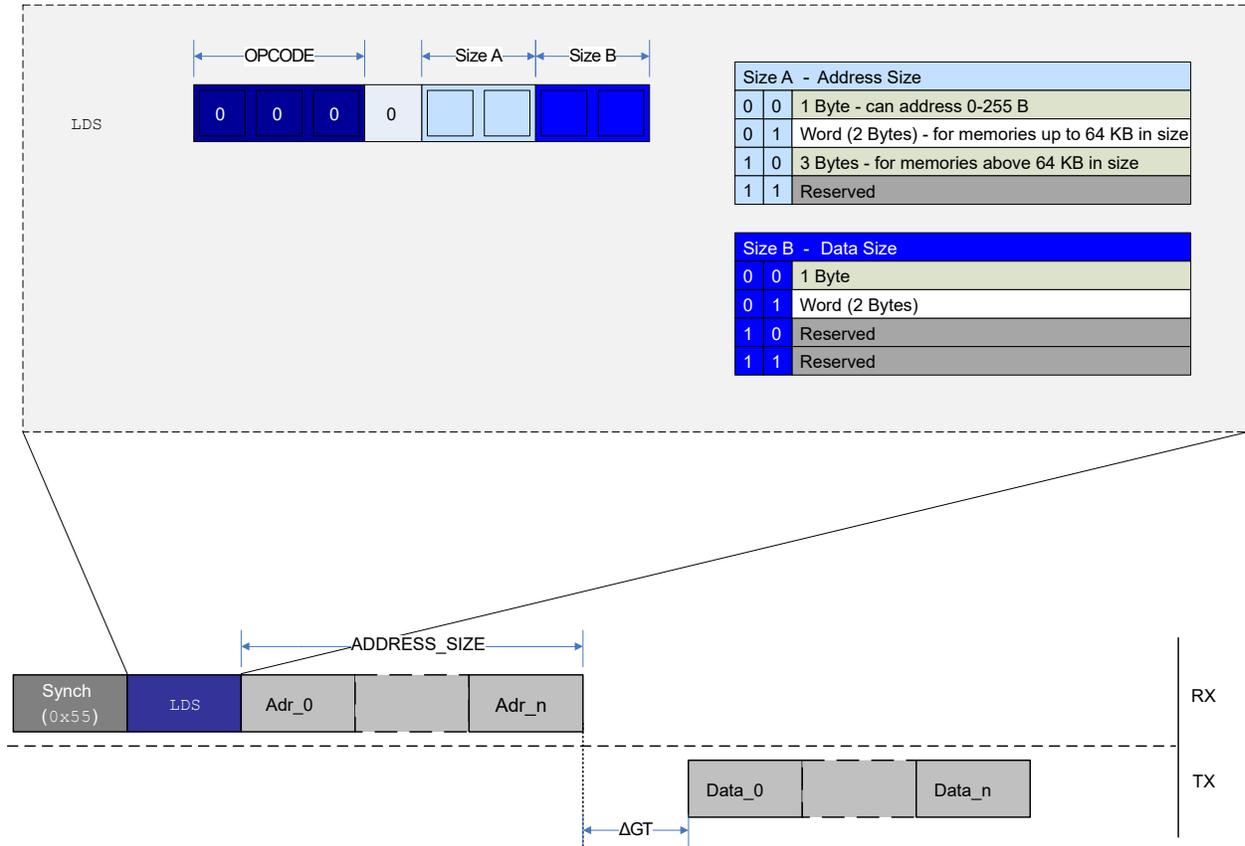


32.3.3.1. LDS——使用直接寻址从数据空间装入数据

LDS 指令用于将数据从系统总线装入 PHY 层移位寄存器中以进行串行读取。LDS 指令基于直接寻址，并且地址必须作为操作数提供给指令才能启动数据传输。地址和数据支持的最大大小为 32 位。LDS 指令与 REPEAT 指令配合使用时支持重复存储器访问。

发出 LDS 指令后，必须发送由 Size A 字段指示的所需地址字节数，然后发送由 Size B 字段选择的输出数据大小。输出数据在指定的保护时间（Guard Time, GT）后发出。当与 REPEAT 指令配合使用时，必须在重复的每次迭代中发送地址，这意味着在每次输出数据采样完成后发送地址。将 REPEAT 和 LDS 配合使用时，没有自动地址递增的情况，因为采用直接寻址协议。

图 32-8. LDS 指令操作



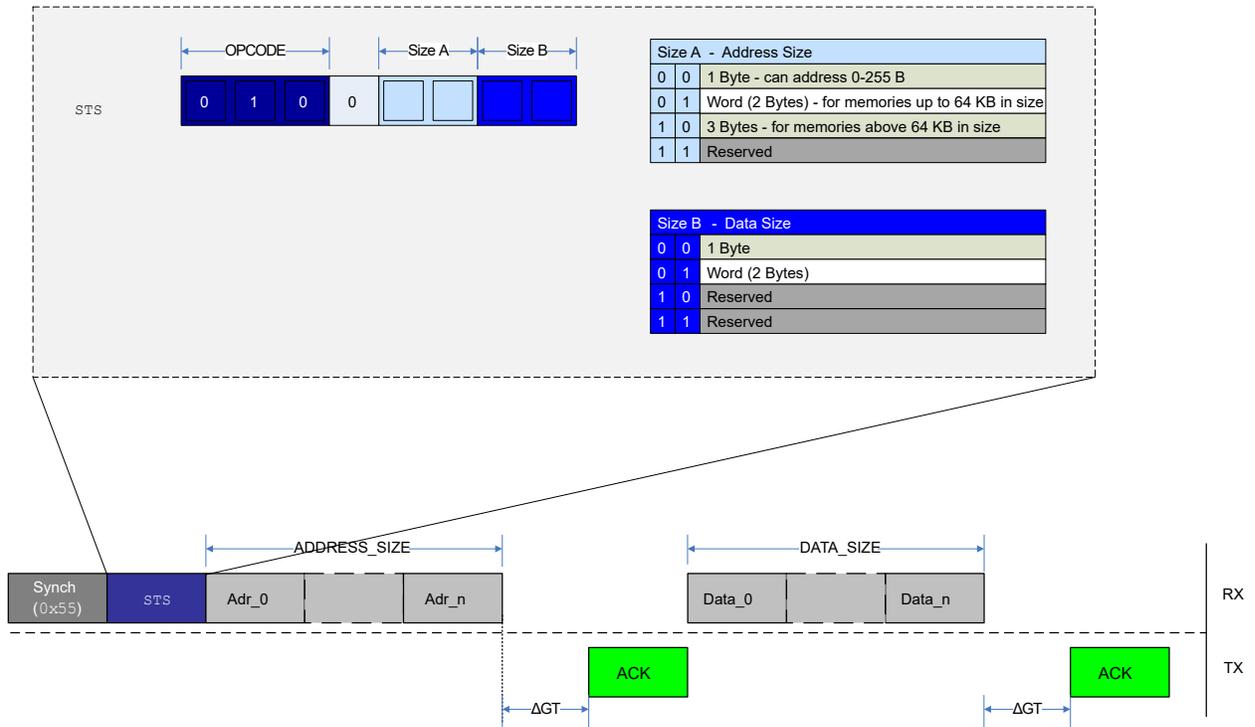
在解码指令并按照解码后的指令指示接收地址字节时，DL 层会将全部所需的信息同步到 ACC 层。ACC 层将会处理总线请求并将总线缓冲的数据再次同步回 DL 层，此过程将产生同步延时，从 UPDI 接收数据时必须考虑这一点。

32.3.3.2. STS——使用直接寻址将数据存储到数据空间

STS 指令用于将以串行方式移入 PHY 层移位寄存器中的数据存储到系统总线地址空间。STS 指令基于直接寻址，并且地址必须作为操作数提供给指令才能启动数据传输。地址是第一组操作数，数据是第二组。地址和数据操作数的大小由图 32-9 所示的大小字段给出。地址和数据的最大大小为 32 位。

STS 与 REPEAT 指令配合使用时支持重复存储器访问。

图 32-9. STS 指令操作



STS 指令的传输协议也在图 32-9 中进行了说明，遵循以下序列：

1. 发送地址。
2. 如果传输成功，则从 UPDI 发回应答（ACK）。
3. 发送 STS 指令中指定的字节数。
4. 数据传输成功后收到新的 ACK。

32.3.3.3. LD——使用间接寻址从数据空间装入数据

LD 指令用于将数据从数据空间装入 PHY 层移位寄存器中以进行串行读取。LD 指令基于间接寻址，这意味着 UPDI 中的地址指针需要在数据空间读取访问之前写入。支持自动指针后递增操作，当 LD 指令与 REPEAT 指令配合使用时，该操作十分有用。也可以对 UPDI 指针寄存器执行 LD。地址和数据装入支持的最大大小为 32 位。

图 32-10. LD 指令操作

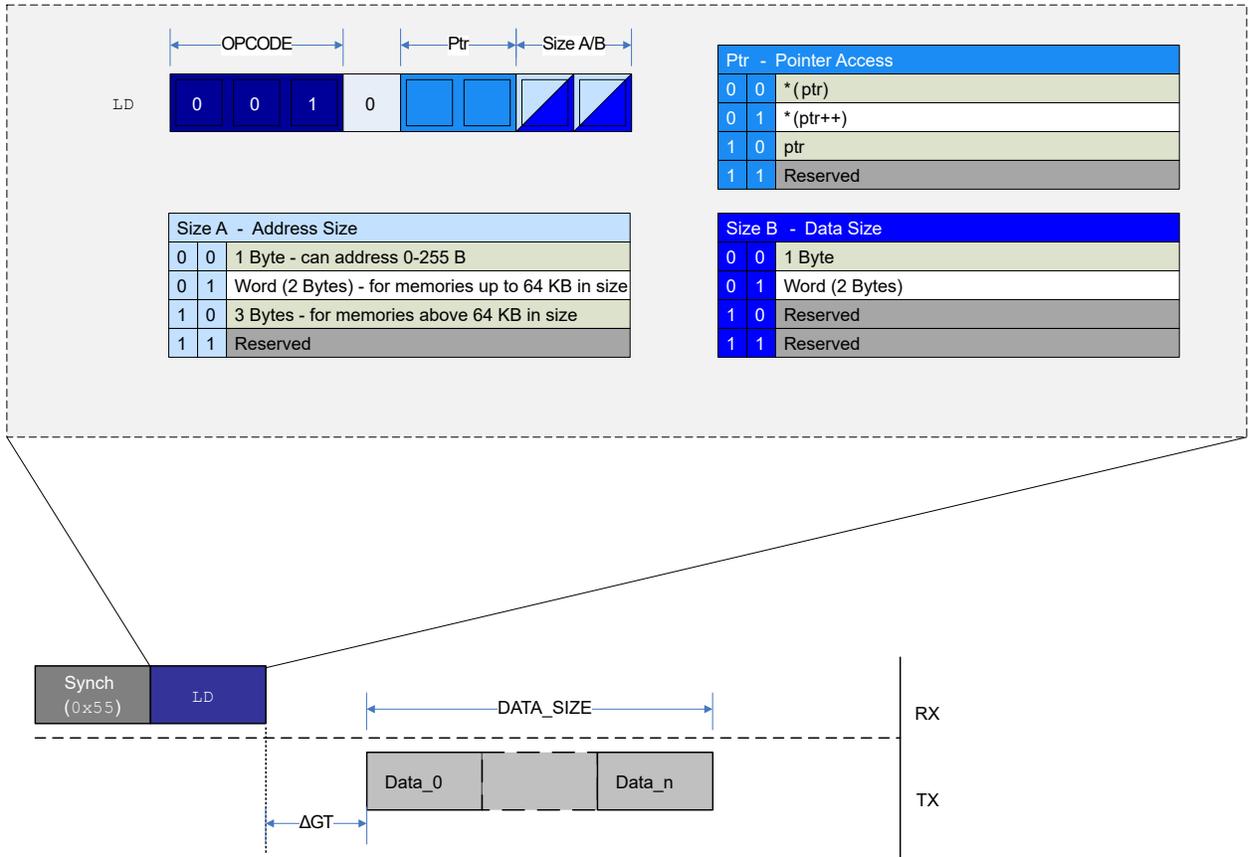


图 32-10 给出了典型 LD 序列的示例，其中的数据在保护时间（GT）段后接收。从 UPDI 指针寄存器装入数据遵循相同的传输协议。

对于来自数据空间的 LD 指令，需对 UPDI 指针寄存器使用 ST 指令以设置指针寄存器。如果在成功写入指针寄存器后接收到 ACK，则应使用所需的 DATA SIZE 操作数来设置 LD 指令。可以直接通过 LD 指令对 UPDI 指针寄存器执行 LD。

32.3.3.4. ST——使用间接寻址将数据从 UPDI 存储到数据空间

ST 指令用于将数据从 UPDI PHY 移位寄存器存储到数据空间。ST 指令用于存储以串行方式移入 PHY 层的数据。ST 指令基于间接寻址，这意味着 UPDI 中的地址指针需要在数据空间之前写入。支持自动指针后递增操作，当 ST 指令与 REPEAT 指令配合使用时，该操作十分有用。ST 指令还用于将 UPDI 地址指针存储到指针寄存器。地址和数据存储支持的最大大小为 32 位。

图 32-11. ST 指令操作

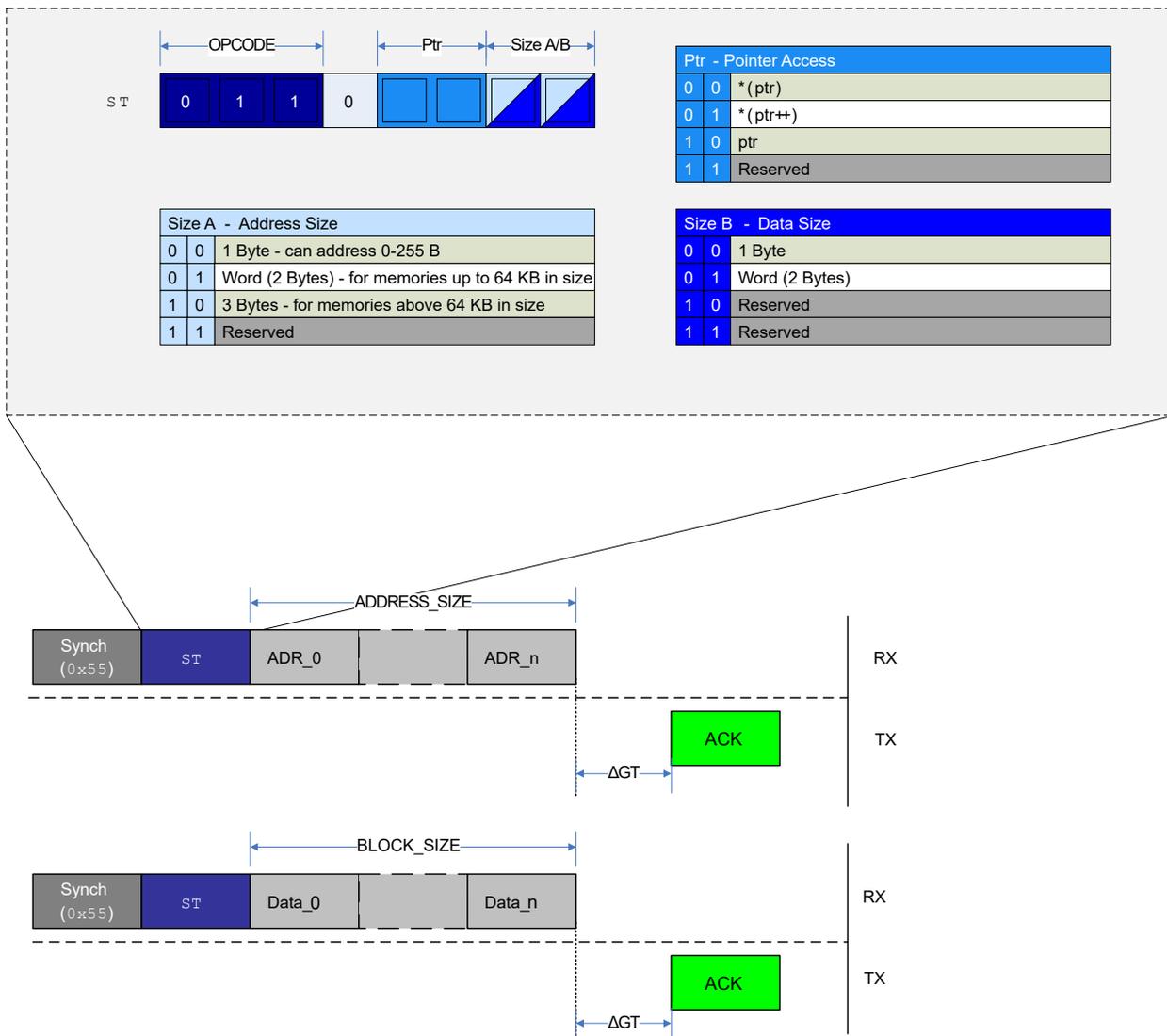


图 32-11 给出了面向 UPDI 指针寄存器的 ST 指令以及常规数据存储的示例。在每条指令之前发送 SYNCH 字符。在这两种情况下，如果成功执行 ST 指令，则 UPDI 将发回应答（ACK）。

按照以下步骤写入 UPDI 指针寄存器：

1. 将 ST 指令中的 PTR 字段设置为签名 0x2。
2. 将地址大小（Size A）字段设置为所需地址大小。
3. 发出 ST 指令后，发送 Size A 字节的地址数据。
4. 等待表示成功写入地址寄存器的 ACK 字符。

写入地址寄存器后，以类似的方式发送数据：

1. 将 ST 指令中的 PTR 字段设置为签名 0x0 以写入 UPDI 指针寄存器指定的地址。如果 PTR 字段设置为 0x1，则 UPDI 指针将在写操作执行完毕后根据指令的数据大小 Size B 字段自动更新到下一个地址。
2. 将指令中的 Size B 字段设置为所需数据大小。
3. 发送 ST 指令后，发送 Size B 字节的数据。

4. 等待表示成功写入总线矩阵的 ACK 字符。

当与 REPEAT 指令配合使用时，建议使用待写入块的起始地址设置地址寄存器，并使用指针后递增寄存器自动增加每个重复周期的地址。使用 REPEAT 指令时，可在接收到每个 ACK 后发送 Size B 数据字节的数据帧。

32.3.3.5. LDCS——从控制和状态寄存器空间装入数据

LDCS 指令用于将串行读取数据从位于 DL 层的 UPDI 控制和状态寄存器空间装入 PHY 层移位寄存器。LDCS 指令基于直接寻址，其中的地址是指令操作数的一部分。LDCS 指令只能访问 UPDI CS 寄存器空间。该指令仅支持字节访问，数据大小不可配置。

图 32-12. LDCS 指令操作

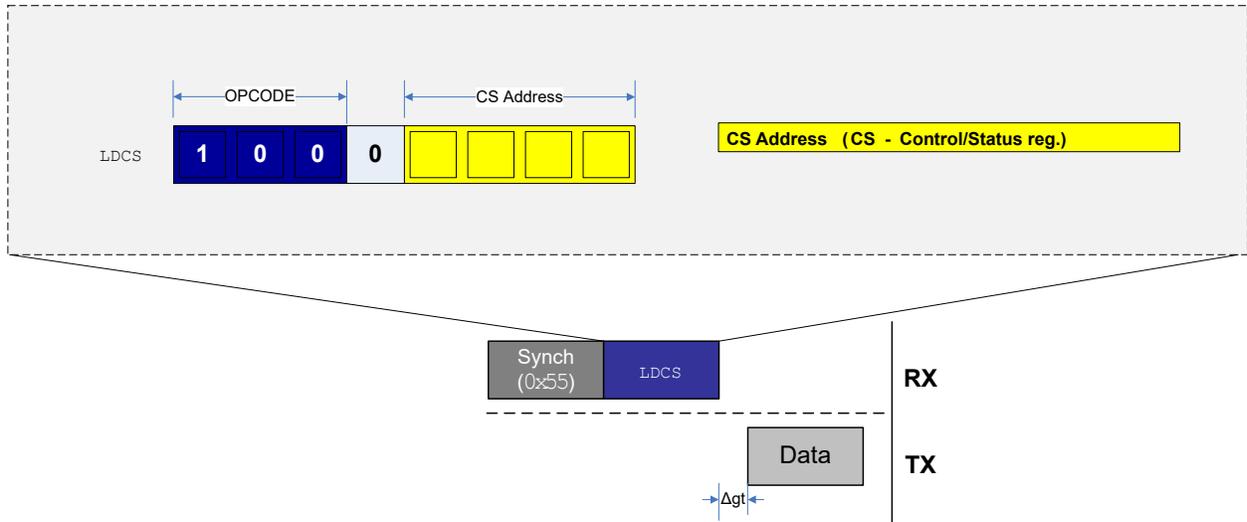


图 32-12 给出了 LDCS 数据传输的典型示例。保护时间完成后，将通过 UPDI 发送 LDCS 的数据字节。

32.3.3.6. STCS——将数据存储到控制和状态寄存器空间

STCS 指令用于将数据存储到 UPDI 控制和状态寄存器空间。数据以串行方式移入 PHY 层移位寄存器，并作为整个字节写入所选的 CS 寄存器。STCS 指令基于直接寻址，其中的地址是指令操作数的一部分。STCS 指令只能访问内部 UPDI 寄存器空间。该指令仅支持字节访问，数据大小不可配置。

图 32-13. STCS 指令操作

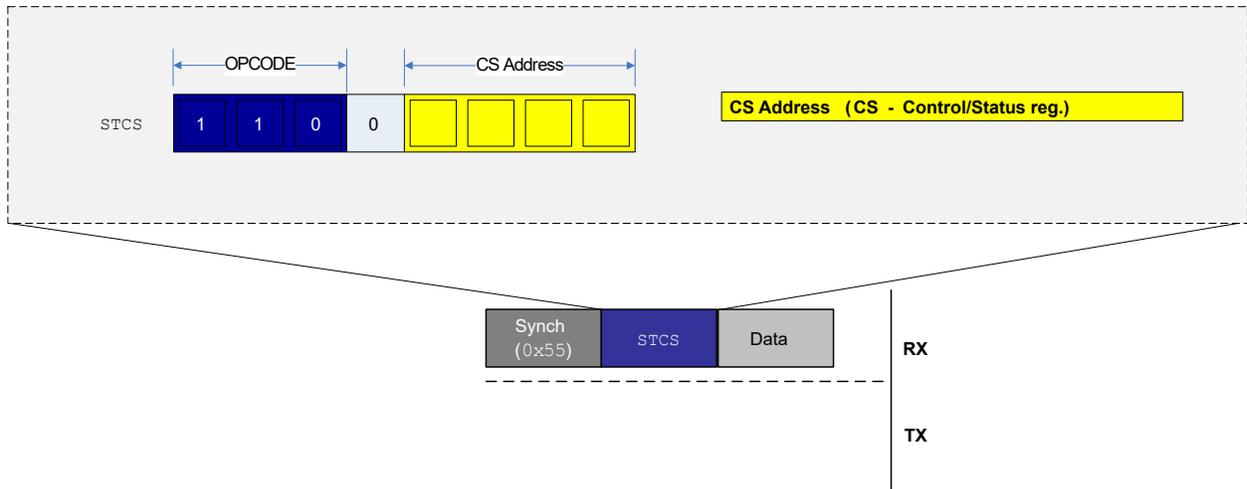


图 32-13 给出了在 SYNCH 字符和指令帧之后传输的数据帧。STCS 指令字节可以紧跟在数据字节之后。STCS 指令不生成任何响应，ST 和 STS 指令也如此。

32.3.3.7. REPEAT——设置指令重复计数器

REPEAT 指令用于将重复计数值存储到 DL 层的 UPDI 重复计数器寄存器中。当指令与 REPEAT 配合使用时，除 REPEAT 发出后的第一条指令外的所有指令都可以忽略 SYNCH 和指令帧的协议开销。REPEAT 对存储器指令（LD、ST、LDS 和 STS）最有用，但除 REPEAT 指令自身以外的所有指令均可以重复。

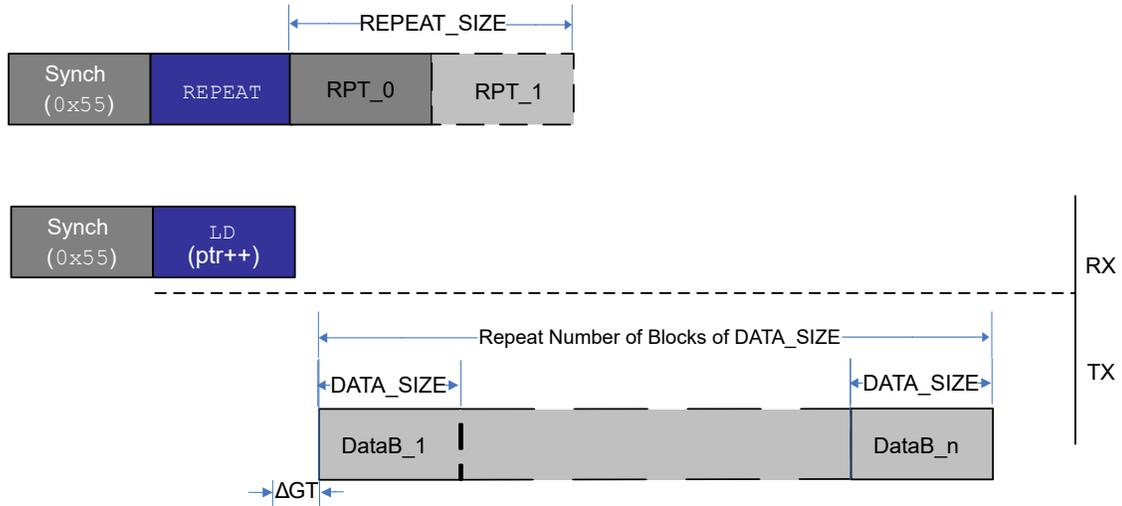
DATA_SIZE 操作数字段指的是重复值的大小。最多仅支持重复 255 次。REPEAT 指令后直接装入的指令将重复 RPT_0 + 1 次。如果重复计数器寄存器为 0，该指令将只运行一次。进行中的重复操作只能通过发送 BREAK 字符中止。

图 32-14. REPEAT 指令操作（与 ST 指令一起使用）



图 32-14 所示为使用指针后递增操作重复 ST 指令的示例。在发送 RPT_0 = n 的 REPEAT 指令后，将发出包含 SYNCH 和指令帧的第一条 ST 指令。接下来的 n 条 ST 指令只需发送符合 ST 操作数 DATA_SIZE 的数据字节并保持应答（ACK）握手协议即可执行。

图 32-15. REPEAT 指令操作（与 LD 指令一起使用）



对于 LD，数据将在 LD 指令之后连续发出。请注意第一个数据块上的保护时间。

如果使用间接寻址指令（LD/ST），在与 REPEAT 配合使用时，建议始终使用指针后递增选项。ST/LD 指令只能位于第一个数据块（具体的数据字节数取决于 DATA_SIZE）的前面。否则，将在所有重复访问操作中访问相同的地址。对于直接寻址指令（LDS/STS），在接收（LDS）或发送（STS）数据之前，必须始终按照指令协议的规定发送地址。

32.3.3.8. KEY——设置激活密钥或发送系统信息块

KEY 指令用于将密钥字节发送到 UPDI 或将系统信息块（System Information Block, SIB）提供给编程器，以开放执行器件上受保护的功能。有关密钥所激活功能的概述，请参见使能密钥受保护接口一节中的密钥激活概述表。KEY 指令仅支持 64 位密钥大小。SIB 支持的最大大小为 128 位。

图 32-16. KEY 指令操作

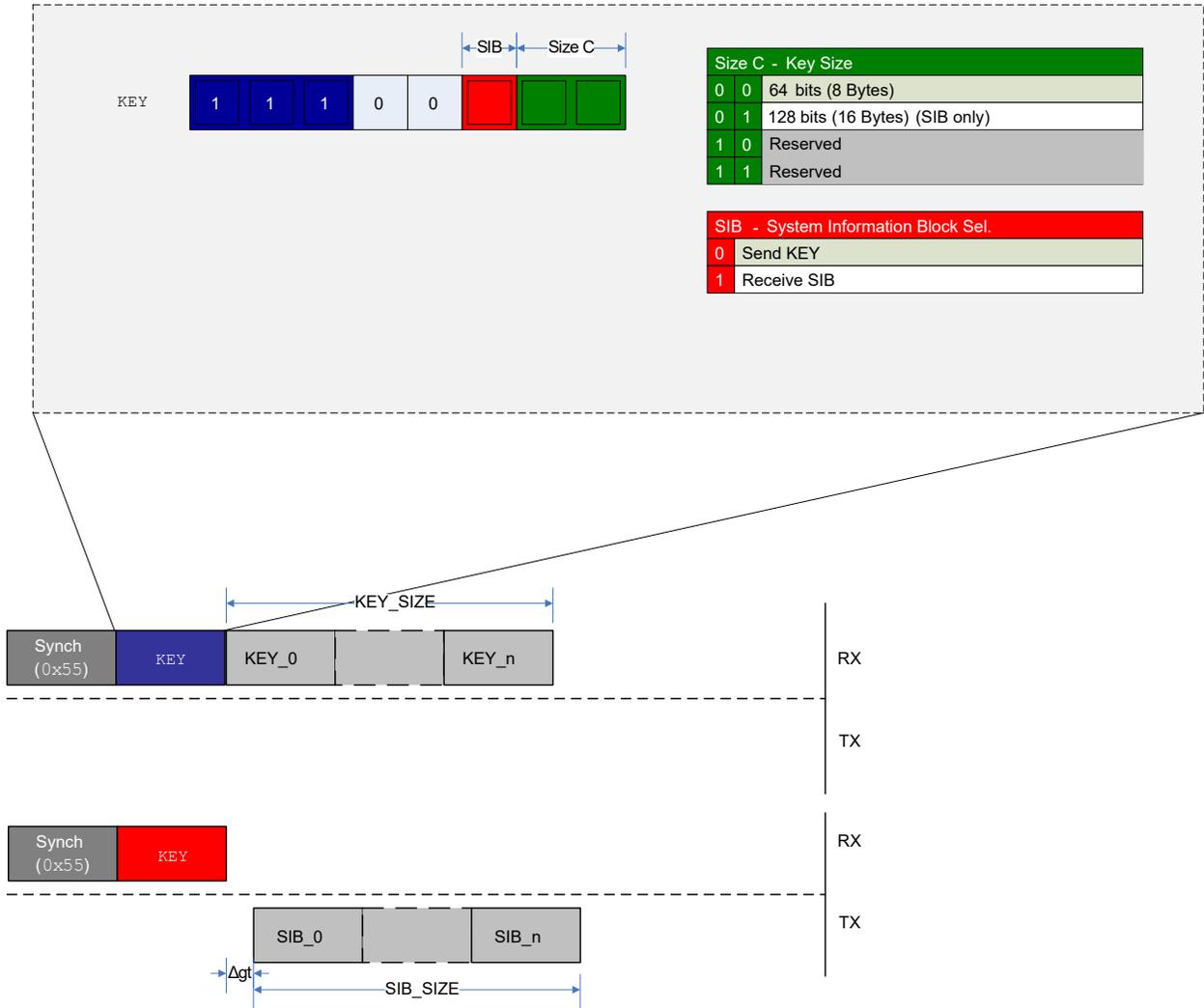


图 32-16 给出了密钥的发送过程和 SIB 的接收过程。在这两种情况下，均由操作数中的 Size C (SIZE_C) 字段决定发送或接收的帧数。将 KEY 发送到 UPDI 后没有响应。当请求 SIB 时，UPDI 将根据当前保护时间设置发送数据。

32.3.4. 启动期间闪存的 CRC 校验

部分器件支持在启动过程中对闪存内容运行 CRC 校验。即使器件被锁定，也可以执行该校验。该 CRC 校验的结果可从 ASI_CRC_STATUS 寄存器读取。有关该功能的更多信息，请参见器件数据手册中的 *CRCSCAN*——循环冗余校验存储器扫描一章。

32.3.5. 使用 UPDI 测量系统时钟

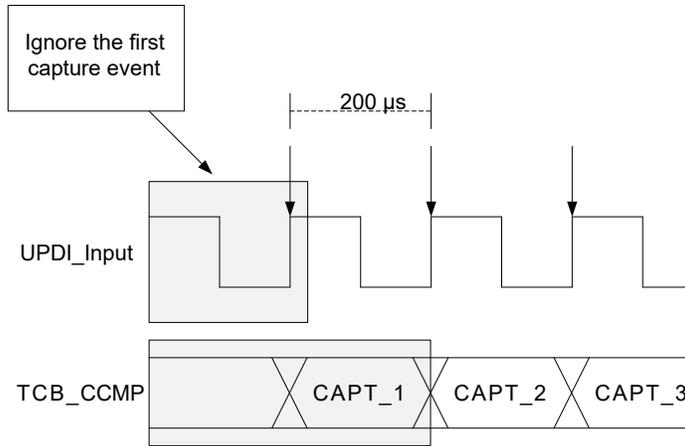
通过将 UPDI 事件连接到带输入捕捉功能的 TCB，可使用 UPDI 准确测量系统时钟频率。该功能的建议设置流程如下：

- 在 TCBn.CTRLB 中设置 CNTMODE = 0x3，即输入捕捉频率测量模式
- 在 TCBn.EVCTRL 中写入 CAPTEI = 1 以允许事件中断。保持 TCBn.EVCTRL 中的 EDGE = 0
- 配置事件系统以将 UPDI SYNCH 事件（生成器）路由到 TCB（用户）
- 对于用于生成 UPDI 事件的 SYNCH 字符，建议使用 10 kbps-50 kbps 范围内的慢速波特率，以便更准确地测量定时器在 UPDI 事件之间所捕捉的值。需要特别注意的是：如果捕捉设置为触发中断，则必须

忽略第一个捕捉值。必须使用基于输入事件的第二个捕捉值进行测量。有关使用 10 kbps UPDI SYNCH 字符脉冲（使定时器捕捉窗口为 200 μ s）的示例，请参见图 32-17。

- 可通过读取 TCBn.CCMP 寄存器以在 SYNCH 字符后直接读取捕捉值，捕捉完成后，CPU 即可将该值写入存储器。更多详细信息，请参见 *TCB——16 位 B 型定时器/计数器* 一章。

图 32-17. UPDI 系统时钟测量事件



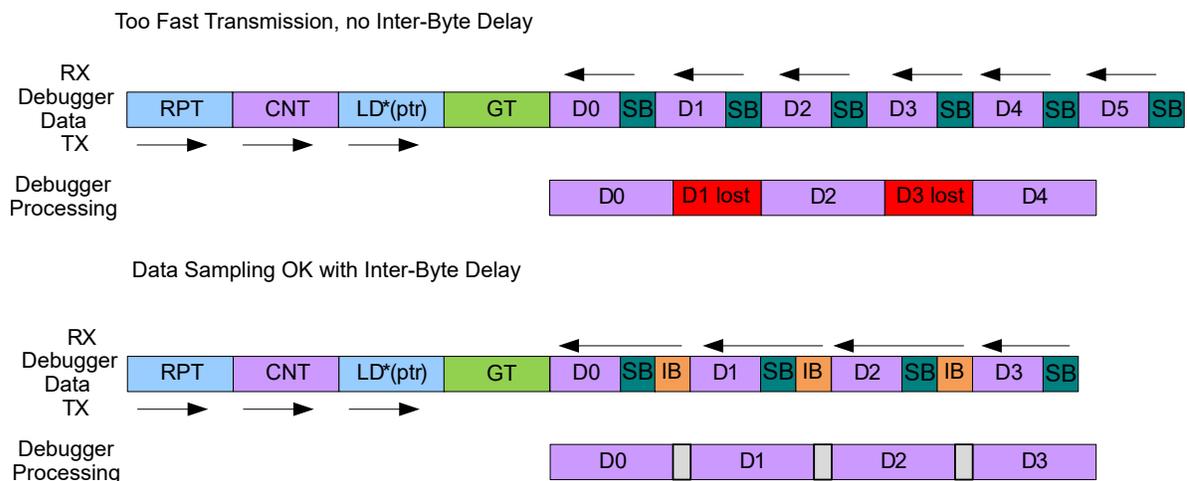
32.3.6. 字节间延时

在执行多字节传输（LD 与 REPEAT 配合使用）或读取系统信息块（SIB）时，输出数据将以连续流的形式输出。根据不同的应用，接收器端的数据可能会出现得过快，进而导致在下一个启动位到达之前可能没有足够的时间来处理数据。

字节间延时的的工作原理是针对多字节传输插入固定数量的空闲位。添加字节间延时的原因是当所有数据都在同一方向进行传输时不会插入保护时间。

通过向控制 A（UPDI.CTRLA）寄存器中的字节间延时使能（IBDLY）位写入 1，可启用字节间延时功能。结果是，将在相邻字节之间插入两个额外的空闲位以放宽调试器的采样时间。

图 32-18. 包含 LD 和 RPT 的字节间延时示例



注:

1. GT 表示插入的保护时间。
2. SB 表示停止位。
3. IB 表示插入的字节间延时。
4. 帧的其余部分为数据和指令。

32.3.7. 系统信息块

根据 *KEY——设置激活密钥或发送系统信息块* 一节所述的 KEY 指令将 SIB 位置 1，可以随时读取系统信息块 (SIB)。无论锁定位的设置如何，调试器始终可以访问 SIB，并且 SIB 能够以一种简洁的方式为调试器提供有关器件和系统参数的信息。这些信息对于识别和设置与器件之间的正确通信通道至关重要。SIB 的输出被解析为 ASCII 符号。密钥大小字段必须在读取完整 SIB 时设置为 16 字节，使用 8 字节大小只能读取 Family_ID。有关 SIB 格式说明，以及支持不同读取大小的数据的信息，请参见图 32-19。

图 32-19. 系统信息块格式

16	8	[Byte][Bits]	Field Name
		[6:0][55:0]	Family_ID
		[7][7:0]	Reserved
		[10:8][23:0]	NVM_VERSION
		[13:11][23:0]	OCD_VERSION
		[14][7:0]	RESERVED
		[15][7:0]	DBG_OSC_FREQ

32.3.8. 使能密钥受保护接口

通过 UPDI 密钥机制保护对某些内部接口和功能的访问。要激活密钥，必须使用 *KEY——设置激活密钥或发送系统信息块* 一节所述的 KEY 指令传输正确的密钥数据。表 32-4 说明了可用的密钥，以及在执行密钥处于激活状态的相关操作时所需的条件。

表 32-4. 密钥激活概述

密钥名称	说明	操作要求	密钥失效条件
全片擦除	启动 NVM 全片擦除。清零锁定位	-	UPDI 禁止/UPDI 复位
NVMPROG	激活 NVM 编程	锁定位清零。ASI_SYS_STATUS.NVMPROG 置 1	编程完成/UPDI 复位
USERROW 写入	对锁定器件上的用户行进行编程	锁定位置 1。 ASI_SYS_STATUS.UROWPROG 置 1	写入密钥状态位/UPDI 复位

表 32-5 概述了要激活接口必须移入的可用密钥签名。

表 32-5. 密钥激活签名

密钥名称	密钥签名 (先写入 LSB)	大小
全片擦除	0x4E564D4572617365	64 位
NVMPROG	0x4E564D50726F6720	64 位
USERROW 写入	0x4E564D5573267465	64 位

32.3.8.1. 全片擦除

应按照以下步骤发起全片擦除:

1. 通过 KEY 指令输入全片擦除密钥。有关 CHIPERASE 签名的信息，请参见表 32-5。
2. 通过 KEY 指令输入 NVM 编程密钥。有关 NVMPROG 签名的信息，请参见表 32-5。这将防止新擦除的器件无法通过 CRC (激活时) 校验。

3. 读取 ASI 密钥状态 (UPDI.ASI_KEY_STATUS) 寄存器, 以验证全片擦除密钥状态 (CHER) 位和 NVM 编程密钥状态 (NVMPROG) 位是否已置 1。
4. 将签名写入 ASI 复位请求 (UPDI.ASI_RESET_REQ) 寄存器中的复位请求 (RSTREQ) 位。这将发出系统复位。
5. 将 0x00 写入 ASI 复位请求 (UPDI.ASI_RESET_REQ) 寄存器以清除系统复位。
6. 读取 ASI 系统状态 (UPDI.ASI_SYS_STATUS) 寄存器中的 NVM 锁定状态 (LOCKSTATUS) 位。
7. 当 LOCKSTATUS 位为 0 时, 表示已完成全片擦除。如果 LOCKSTATUS 位为 1, 则返回至步骤 5。

全片擦除成功后, 锁定位将清零, UPDI 将能够完全访问系统。在锁定位清零之前, UPDI 无法访问系统总线, 只能执行 CS 空间操作。



在全片擦除期间, BOD 强制设为 ON 状态 (通过写入控制 A (BOD.CTRLA) 寄存器中的工作状态 (ACTIVE) 位域) 并使用 BOD 配置 (FUSE.BODCFG) 熔丝中的 BOD 电压 (LVL) 位域和控制 B (BOD.CTRLB) 寄存器中的 BOD 电压 (LVL) 位域。如果电源电压 V_{DD} 低于阈值电压, 则在 V_{DD} 升至足够高之前, 器件无法工作。更多详细信息, 请参见 *BOD* 一章。

32.3.8.2. NVM 编程

如果器件已解锁, 则可以使用 UPDI 直接写入 NVM 控制器。在这种情况下, 如果 CPU 在 NVM 编程期间工作, 则将导致代码执行不可预测。为避免这种情况, 应执行以下 NVM 编程序列:

1. 按照全片擦除一节所述的全片擦除步骤进行操作。如果器件已解锁, 则跳过这一步。
2. 通过使用 KEY 指令输入 NVMPROG 密钥。有关 NVMPROG 签名的信息, 请参见表 32-5。
3. **可选:** 读取 ASI 密钥状态 (UPDI.KEY_STATUS) 寄存器中的 NVM 编程密钥状态 (NVMPROG) 位, 以查看密钥是否已激活。
4. 将签名写入 ASI 复位请求 (UPDI.ASI_RESET_REQ) 寄存器中的复位请求 (RSTREQ) 位。这将发出系统复位。
5. 将 0x00 写入 ASI 复位请求 (UPDI.ASI_RESET_REQ) 寄存器以清除系统复位。
6. 读取 ASI 系统状态 (UPDI.ASI_SYS_STATUS) 寄存器中的 NVM 编程密钥状态 (NVMPROG) 位。
7. NVM 编程可在 NVMPROG 为 1 时启动。如果 NVMPROG 位为 0, 则返回到步骤 6。
8. 通过 UPDI 将数据写入 NVM。
9. 将签名写入 ASI 复位请求 (UPDI.ASI_RESET_REQ) 寄存器中的复位请求 (RSTREQ) 位。这将发出系统复位。
10. 将 0x00 写入 ASI 复位请求 (UPDI.ASI_RESET_REQ) 寄存器以清除系统复位。
11. 编程完成。

32.3.8.3. 用户行编程

用户行编程功能允许将新值编程到锁定器件上的用户行 (USERROW)。按照以下步骤使能此功能进行编程:

1. 通过使用 KEY 指令输入表 32-5 中的 USERROW 写入密钥。有关 USERROW 写入签名的信息, 请参见表 32-5。
2. **可选:** 读取 ASI 密钥状态 (UPDI.ASI_KEY_STATUS) 寄存器中的用户行写密钥状态 (UROWWRITE) 位, 以查看密钥是否已激活。
3. 将签名写入 ASI 复位请求 (UPDI.ASI_RESET_REQ) 寄存器中的复位请求 (RSTREQ) 位。这将发出系统复位。

4. 将 0x00 写入 ASI 复位请求 (UPDI.ASI_RESET_REQ) 寄存器以清除系统复位。
5. 读取 ASI 系统状态 (UPDI.ASI_SYS_STATUS) 寄存器中的启动用户行编程 (UROWPROG) 位。
6. 用户行编程可在 UROWPROG 位为 1 时启动。如果 UROWPROG 为 0, 则返回到步骤 5。
7. 要写入用户行的数据必须先写入 RAM 缓冲区。RAM 中可写区域的大小为 32/64 字节, 只能将用户行数据写入 RAM 的前 32/64 个字节地址。在该存储器范围之外寻址将导致不会执行写操作。在完成编程序列后将数据复制到用户行时, 数据将以 1:1 的比例与用户行空间进行映射。
8. 当所有用户行数据都写入 RAM 后, 写入 ASI 系统控制 A (UPDI.ASI_SYS_CTRLA) 寄存器中的用户行编程完成 (UROWDONE) 位。
9. 读取 ASI 系统状态 (UPDI.ASI_SYS_STATUS) 寄存器中的启动用户行编程 (UROWPROG) 位。
10. 用户行编程在 UROWPROG 位为 0 时完成。如果 UROWPROG 位为 1, 则返回到步骤 9。
11. 写入 ASI 密钥状态 (UPDI.ASI_KEY_STATUS) 寄存器中的用户行写密钥状态 (UROWWRITE) 位。
12. 将签名写入 ASI 复位请求 (UPDI.ASI_RESET_REQ) 寄存器中的复位请求 (RSTREQ) 位。这将发出系统复位。
13. 将 0x00 写入 ASI 复位请求 (UPDI.ASI_RESET_REQ) 寄存器以清除系统复位。
14. 用户行编程完成。

在该模式下, 无法从 RAM 回读数据。只允许写入 RAM 的前 32/64 个字节。

32.3.9. 事件

UPDI 可生成以下事件:

表 32-6. UPDI 中的事件生成器

生成器名称		说明	事件类型	生成时钟域	事件长度
模块	事件				
UPDI	SYNCH	SYNCH 字符	电平	CLK_UPDI	与 CLK_UPDI 同步的 UPDI 引脚上的 SYNCH 字符

该事件基于 SYNCH 字符中检测到的每个正边沿的 UPDI 时钟设置, 并且无法通过 UPDI 禁止该事件。

UPDI 没有事件用户。

有关事件类型和事件系统配置的更多详细信息, 请参见 *EVSYS——事件系统* 一章。

32.3.10. 休眠模式操作

UPDI PHY 层独立于所有休眠模式运行, 连接的调试器始终可以访问 UPDI, 而与器件的休眠状态无关。如果系统进入关闭系统时钟的休眠模式, 则 UPDI 无法访问系统总线及读取存储器和外设。使能后, UPDI 将请求系统时钟, 以便 UPDI 可以与器件的其余部分始终保持通信。因此, UPDI PHY 层时钟不受休眠模式设置的影响。通过读取 ASI 系统状态 (UPDI.ASI_SYS_STATUS) 寄存器中的系统域休眠 (INSLEEP) 位, 可以监视系统域是否处于休眠模式。

通过对 ASI 系统控制 A (UPDI.ASI_SYS_CTRLA) 寄存器中的请求系统时钟 (CLKREQ) 位进行写操作, 可以防止进入休眠模式时系统时钟停止。如果该位置 1, 则会模拟系统休眠模式状态, 即使处于最深度的休眠模式, UPDI 也可以访问系统总线及读取外设寄存器。

使能 UPDI 时, CLKREQ 位默认为 1, 这意味着休眠模式期间的默认操作是保持系统时钟处于开启状态。

32.4. 寄存器汇总

偏移量	名称	位位置	7	6	5	4	3	2	1	0
0x00	STATUSA	7:0	UPDIREV[3:0]							
0x01	STATUSB	7:0						PESIG[2:0]		
0x02	CTRLA	7:0	IBDLY		PARD	DTD	RSD	GTVAL[2:0]		
0x03	CTRLB	7:0				NACKDIS	CCDETDIS	UPDIDIS		
0x04	保留									
...										
0x06										
0x07	ASI_KEY_STATUS	7:0			UROWWRITE	NVMPROG	CHIPERASE			
0x08	ASI_RESET_REQ	7:0	RSTREQ[7:0]							
0x09	ASI_CTRLA	7:0							UPDICKSEL[1:0]	
0x0A	ASI_SYS_CTRLA	7:0						UROWWRITE_FINAL	CLKREQ	
0x0B	ASI_SYS_STATUS	7:0			RSTSYS	INSLEEP	NVMPROG	UROWPROG		LOCKSTATUS
0x0C	ASI_CRC_STATUS	7:0						CRC_STATUS[2:0]		

32.5. 寄存器说明

这些寄存器只能使用特殊指令通过 UPDI 读取，不能通过 CPU 读取。

32.5.1. 状态 A

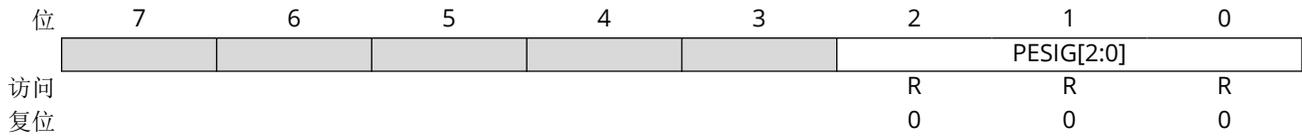
名称: STATUSA
 偏移量: 0x00
 复位: 0x20
 属性: -

位	7	6	5	4	3	2	1	0
	UPDIREV[3:0]							
访问	R	R	R	R				
复位	0	0	1	0				

Bit 7:4 - UPDIREV[3:0] UPDI 版本
 该位域包含当前 UPDI 实现的版本。

32.5.2. 状态 B

名称: STATUSB
偏移量: 0x01
复位: 0x00
属性: -



Bit 2:0 - PESIG[2:0] UPDI 错误签名

PESIG 位域用于指示 UPDI 错误签名，在发生内部 UPDI 错误条件时置 1。该位域在通过调试器读取后清零。

表 32-7. 有效错误签名

PESIG[2:0]	错误类型	错误描述
0x0	无错误	未检测到错误（默认）
0x1	奇偶校验错误	奇偶校验位采样错误
0x2	帧错误	停止位采样错误
0x3	访问层超时错误	UPDI 无法从访问层获取数据或响应
0x4	时钟恢复错误	启动位采样错误
0x5	-	保留
0x6	总线错误	地址错误或访问权限错误
0x7	争用错误	UPDI 引脚上存在驱动争用

32.5.3. 控制 A

名称: CTRLA
偏移量: 0x02
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
	IBDLY		PARD	DTD	RSD	GTVAL[2:0]		
访问	R/W		R/W	R/W	R/W	R/W	R/W	R/W
复位	0		0	0	0	0	0	0

Bit 7 - IBDLY 字节间延时使能

向该位写入 1 会在执行多字节 LD(S)时在 UPDI 发送的各个数据字节之间使能固定长度的字节间延时。固定长度为两个空闲位。

Bit 5 - PARD 奇偶校验禁止

向该位写入 1 将通过忽略奇偶校验位禁止 UPDI 中的奇偶校验检测。建议仅在测试期间使用此功能。

Bit 4 - DTD 禁止超时检测

向该位写入 1 将禁止 PHY 层上的超时检测，这会请求 ACC 层在指定时间（65536 个 UPDI 时钟周期）内发出响应。

Bit 3 - RSD 响应签名禁止

向该位写入 1 将禁止 UPDI 生成的任何响应签名，这样就可以在将大块数据写入 NVM 空间时将协议开销降至最低。当访问系统总线时，UPDI 可能会经历延时。如果该延时可预测，则可以禁止响应签名。否则，可能会发生数据丢失。

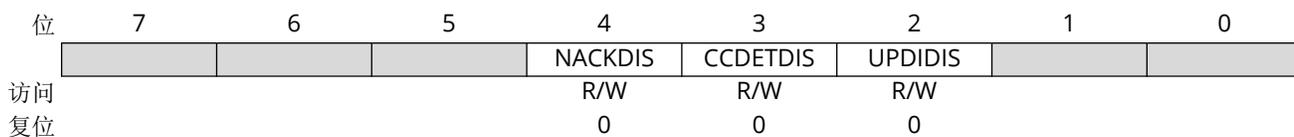
Bit 2:0 - GTVAL[2:0] 保护时间值

该位域选择当传输方向从 RX 切换为 TX 时 UPDI 所使用的保护时间值。

值	说明
0x0	UPDI 保护时间: 128 个周期 (默认)
0x1	UPDI 保护时间: 64 个周期
0x2	UPDI 保护时间: 32 个周期
0x3	UPDI 保护时间: 16 个周期
0x4	UPDI 保护时间: 8 个周期
0x5	UPDI 保护时间: 4 个周期
0x6	UPDI 保护时间: 2 个周期
0x7	保留

32.5.4. 控制 B

名称: CTRLB
 偏移量: 0x03
 复位: 0x00
 属性: -



Bit 4 - NACKDIS 禁止 NACK 响应

如果在进行 LD(S)和 ST(S)操作期间发出系统复位，则向该位写入 1 将禁止 UPDI 发送的 NACK 签名。

Bit 3 - CCDETDIS 冲突和争用检测禁止

向该位写入 1 将禁止争用检测。向该位写入 0 将使能争用检测。

Bit 2 - UPDIDIS UPDI 禁止

向该位写入 1 将禁止 UPDI PHY 接口。UPDI 时钟请求的优先级会降低，并且 UPDI 会复位。当 UPDI 禁止时，所有 UPDI PHY 配置和密钥都将复位。

32.5.5. ASI 密钥状态

名称: ASI_KEY_STATUS
偏移量: 0x07
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
			UROWWRITE	NVMPROG	CHIPERASE			
访问			R/W	R	R			
复位			0	0	0			

Bit 5 - UROWWRITE 用户行写密钥状态

如果 UROWWRITE 密钥成功解码，则该位置 1。该位必须作为用户行写入程序的最后一部分写入，以便正确复位编程会话。

Bit 4 - NVMPROG NVM 编程密钥状态

如果 NVMPROG 密钥成功解码，则该位置 1。该位在 NVM 编程序列启动时清零，此时 ASI_SYS_STATUS 中的 NVMPROG 位置 1。

Bit 3 - CHIPERASE 片上擦除密钥状态

如果全片擦除密钥成功解码，则该位置 1。该位通过全片擦除一节所述的全片擦除序列中发出的复位请求清零。

32.5.6. ASI 复位请求

名称: ASI_RESET_REQ
偏移量: 0x08
复位: 0x00
属性: -

将复位签名写入该寄存器时，会向系统发出复位信号。

位	7	6	5	4	3	2	1	0
	RSTREQ[7:0]							
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 7:0 - RSTREQ[7:0] 复位请求

从该寄存器发出系统复位时，不会复位 UPDI。

值	名称	说明
0x00	RUN	清除复位条件
0x59	RESET	正常复位
其他	-	保留的

32.5.7. ASI 控制 A

名称: ASI_CTRLA
 偏移量: 0x09
 复位: 0x03
 属性: -

位	7	6	5	4	3	2	1	0
							UPDICKSEL[1:0]	
访问							R/W	R/W
复位							1	1

Bit 1:0 - UPDICKSEL[1:0] UPDI 时钟分频比选择

写入这些位可选择 UPDI 时钟输出频率。UPDI 最大时钟频率取决于器件的电源电压 V_{DD} 。

值	说明
0x0	保留
0x1	16 MHz UPDI 时钟 (建议 V_{DD} 高于 2.7V)
0x2	8 MHz UPDI 时钟 (建议 V_{DD} 高于 2.2V)
0x3	4 MHz UPDI 时钟 (默认设置)

32.5.8. ASI 系统控制 A

名称: ASI_SYS_CTRLA
偏移量: 0x0A
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
							UROWWRITE_FINAL	CLKREQ
访问	R	R	R	R	R	R	R/W	R/W
复位	0	0	0	0	0	0	0	0

Bit 1 - UROWWRITE_FINAL 用户行编程完成

当用户行数据写入 RAM 后，必须写入该位。向该位写入 1 将启动将用户行数据编程到闪存的过程。
如果 UPDI 在将用户行数据写入 RAM 之前写入该位，则 CPU 继续操作，但不会写入数据。
只有成功解码 USERROW 写入密钥时，该位才可写。

Bit 0 - CLKREQ 请求系统时钟

如果该位写入 1，则 ASI 将请求系统时钟，而与系统休眠模式无关。这样一来，即使系统处于休眠模式，UPDI 也可以访问 ACC 层。
向该位写入 0 将降低时钟请求的优先级。
禁止 UPDI 时，该位将复位。
在任何编程模式（熔丝或高电压）下使能 UPDI 时，该位默认置 1。

32.5.9. ASI 系统状态

名称: ASI_SYS_STATUS
偏移量: 0x0B
复位: 0x01
属性: -

位	7	6	5	4	3	2	1	0
			RSTSYS	INSLEEP	NVMPROG	UROWPROG		LOCKSTATUS
访问			R	R	R	R		R
复位			0	0	0	0		1

Bit 5 - RSTSYS 系统复位有效

当该位置 1 时，系统域上存在有效复位。当该位清零时，系统未处于复位状态。
该位在进行读操作时清零。
ASI_RESET_REQ 寄存器保持的复位也会影响该位。

Bit 4 - INSLEEP 系统域休眠

当该位置 1 时，系统域处于空闲或深度休眠模式。当该位清零时，系统未处于任何休眠模式。

Bit 3 - NVMPROG 启动 NVM 编程

当该位置 1 时，NVM 编程可从 UPDI 开始。
编程完成后，通过 UPDI 复位寄存器复位系统。

Bit 2 - UROWPROG 启动用户行编程

当该位置 1 时，用户行编程可从 UPDI 开始。
当用户行数据已写入 RAM 时，必须写入 ASI_SYS_CTRLA 寄存器中的 UROWDONE 位。

Bit 0 - LOCKSTATUS NVM 锁定状态

当该位置 1 时，器件处于锁定状态。如果执行了全片擦除且锁定位设置为 0，则该位将读为 0。

32.5.10. ASI CRC 状态

名称: ASI_CRC_STATUS
偏移量: 0x0C
复位: 0x00
属性: -

位	7	6	5	4	3	2	1	0
						CRC_STATUS[2:0]		
访问						R	R	R
复位						0	0	0

Bit 2:0 - CRC_STATUS[2:0] CRC 执行状态

该位域表示 CRC 转换的状态。该位域采用独热编码。

值	说明
0x0	不使能
0x1	CRC 使能, 繁忙
0x2	CRC 使能, 使用 OK 签名完成
0x4	CRC 使能, 使用 FAILED 签名完成
其他	保留

33. 约定

33.1. 数字符号

表 33-1. 数字符号

符号	说明
165	十进制数
0b0101	二进制数
0101	除非明确说明，否则二进制数没有前缀
0x3B24	十六进制数
X	表示未知值或任意值
Z	表示信号或总线为高阻抗（悬空）状态

33.2. 存储器大小和类型

表 33-2. 存储器大小和比特率

符号	说明
KB	千字节 ($2^{10}\text{B} = 1024\text{B}$)
MB	兆字节 ($2^{20}\text{B} = 1024\text{KB}$)
GB	千兆字节 ($2^{30}\text{B} = 1024\text{MB}$)
b	位（二进制 0 或 1）
B	字节（8 位）
1 kbps	1,000 bps 的速率
1 Mbps	1,000,000 bps 的速率
1 Gbps	1,000,000,000 bps 的速率
字	16 位

33.3. 频率和时间

表 33-3. 频率和时间

符号	说明
kHz	1 kHz = 10^3 Hz = 1,000 Hz
MHz	1 MHz = 10^6 Hz = 1,000,000 Hz
GHz	1 GHz = 10^9 Hz = 1,000,000,000 Hz
ms	1 ms = $10^{-3}\text{s} = 0.001\text{s}$
μs	1 $\mu\text{s} = 10^{-6}\text{s} = 0.000001\text{s}$
ns	1 ns = $10^{-9}\text{s} = 0.000000001\text{s}$

33.4. 寄存器和位

表 33-4. 寄存器和位助记符

符号	说明
R/W	可读/写寄存器位。用户可以读取和写入该位。
R	只读寄存器位。用户只能读取该位。写操作将被忽略。
W	只写寄存器位。用户只能写入该位。读取该位将返回未定义的值。
BITFIELD	位域名称以大写字母显示。示例：INTMODE。
BITFIELD[n:m]	从 bit n 到 bit m 的一组位。示例：PINA[3:0] = {PINA3, PINA2, PINA1, PINA0}。

表 33-4. 寄存器和位助记符（续）

符号	说明
保留	保留的位、位域和位域值不使用，保留供将来使用。为了与将来的器件兼容，在写入寄存器时始终向保留位写入 0。读取保留位时始终返回 0。
PERIPHERALn	如果存在多个外设实例，则在外设名称末尾追加一个数字来标识一个实例。示例：USARTn 是 USART 模块的所有实例的集合，USART3 是 USART 模块的一个特定实例。
PERIPHERALx	如果存在多个外设实例，则在外设名称末尾追加一个大写字母（A-Z）来标识一个实例。示例：PORTx 是 PORT 模块的所有实例的集合，PORTB 是 PORT 模块的一个特定实例。
复位	上电复位后寄存器的值。这也是对外设执行软件复位后外设中寄存器的值，但调试控制寄存器除外。
SET/CLR/TGL	具有 SET/CLR/TGL 后缀的寄存器允许用户在不进行读-修改-写操作的情况下将寄存器中的位清零和置 1。每个 SET/CLR/TGL 寄存器都与其影响的寄存器配对。读取寄存器对中的两个寄存器时将返回相同的值。 示例：在 PORT 外设中，OUT 寄存器与 OUTSET 寄存器可构成这样的寄存器对。写入 OUTSET 时将修改 OUT 的内容。读取 OUT 和 OUTSET 时将返回相同的值。 向 CLR 寄存器中的某个位写入 1 会同时将两个寄存器中的相应位清零。 向 SET 寄存器中的某个位写入 1 会同时将两个寄存器中的相应位置 1。 向 TGL 寄存器中的某个位写入 1 会同时将两个寄存器中的相应位翻转。

33.4.1. 在头文件中寻址寄存器

要在提供的 C 头文件中寻址寄存器，需遵循以下规则：

1. 通过<peripheral_instance_name>.<register_name>标识寄存器，例如 CPU.SREG、USART2.CTRLA 或 PORTB.DIR。
2. 关于外设名称，请参见“外设和架构”一章中的“外设地址映射”一节。
3. 将外设名称中的 n 或 x 替换为正确的实例标识符以获得<peripheral_instance_name>。
4. 为外设寄存器分配预定义值时，需按照以下规则构建：
<peripheral_name>_<bit_field_name>_<bit_field_value>_gc
将<peripheral_instance_name>中的实例标识符移除后就是<peripheral_name>。
关于<bit_field_value>，可参见“寄存器说明”章节中描述外设寄存器位域的表中的“名称”列。

例 33-1. 寄存器赋值

```
// EVSYS channel 0 is driven by TCB3 OVF event
EVSYS.CHANNEL0 = EVSYS_CHANNEL0_TCB3_OVF_gc;

// USART0 RXMODE uses Double Transmission Speed
USART0.CTRLB = USART_RXMODE_CLK2X_gc;
```

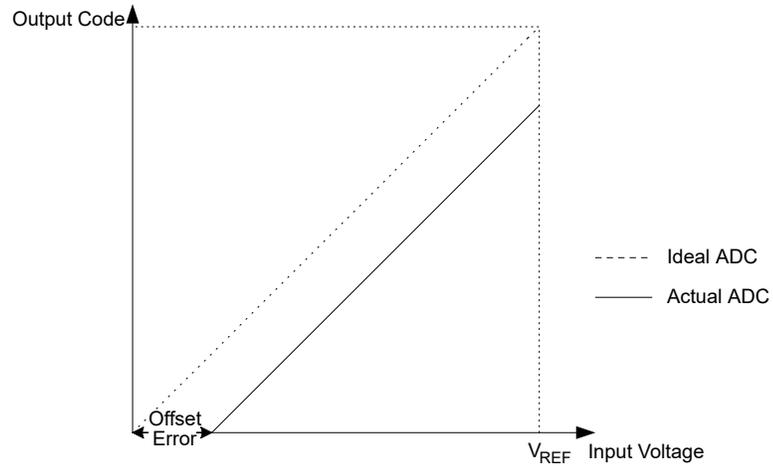
注：对于在不同模式下具有不同寄存器集的外设，必须在<peripheral_instance_name>和<peripheral_name>后面追加模式名称。

33.5. ADC 参数定义

理想的 n 位单端 ADC 会通过 2^n 步（LSb）线性转换 GND 与 V_{REF} 之间的电压。最低编码读为 0，最高编码读为 2^n-1 。下面几个参数描述了与理想行为的偏差：

失调误差 第一次转换（0x000 至 0x001）相比于理想转换（0.5 LSb 处）的偏差。理想值：0 LSb。

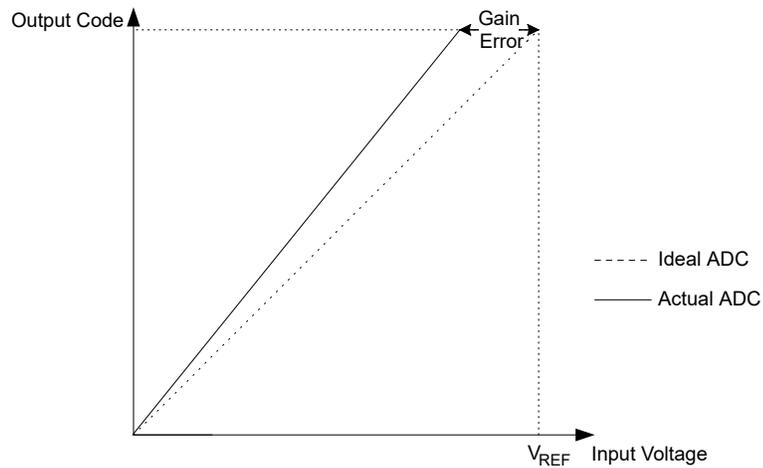
图 33-1. 失调误差



增益误差

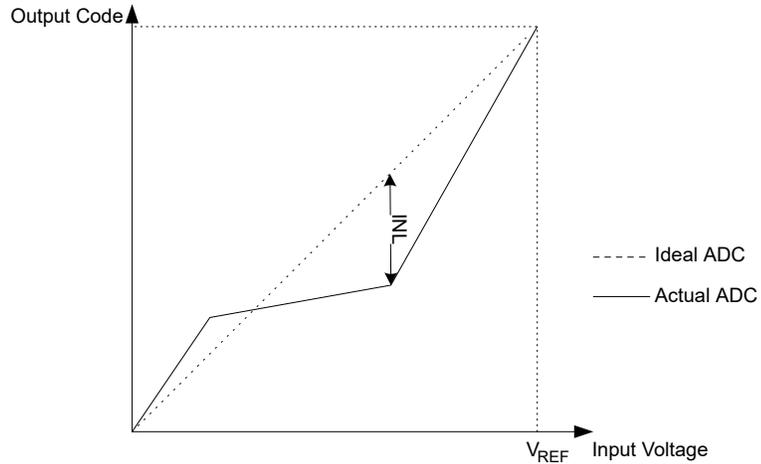
调整失调误差后，会出现增益误差，即最后一次转换（例如，对于 10 位 ADC，为 0x3FE 至 0x3FF）相比于理想转换（低于最大值 1.5 LSB 处）的偏差。理想值：0 LSB。

图 33-2. 增益误差

积分非线性
(Integral
Nonlinearity,
INL)

调整失调和增益误差后，还有 INL，即实际转换相比于任意编码的理想转换的最大偏差。理想值：0 LSB。

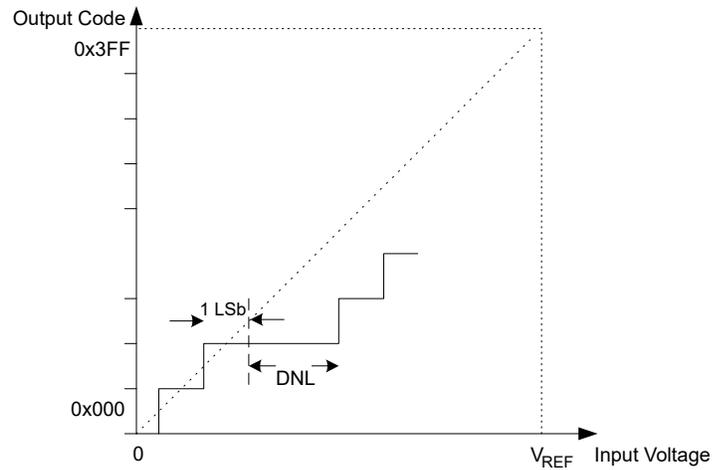
图 33-3. 积分非线性



微分非线性
(Differential Nonlinearity, DNL)

实际编码宽度（相邻两次转换的间隔）与理想编码宽度（1 LSB）的最大偏差。理想值：0 LSB。

图 33-4. 微分非线性



量化误差

由于输入电压会量化为有限数量的编码，因此输入电压范围（1 LSB 宽）将编码为相同值。始终为 ± 0.5 LSB。

绝对精度

实际（未调整的）转换相比于任意编码的理想转换的最大偏差。此为上述所有误差的复合结果。理想值： ± 0.5 LSB。

34. 电气特性

34.1. 免责声明

除非另外说明，否则所有典型值均在 $T = 25^{\circ}\text{C}$ 且 $V_{\text{DD}} = 3\text{V}$ 的条件下测得。除非另外说明，否则所有最小值和最大值均适用于整个工作温度和电压范围。

34.2. 绝对最大额定值

如果器件的工作条件超过下述绝对最大额定值，可能对器件造成永久性损坏。下述值仅代表极限工作条件，不建议在处于或超出下述极限值的情况下工作。器件长时间工作在最大值条件下，其可靠性可能受到影响。

表 34-1. 绝对最大额定值

符号	说明	条件	最小值	最大值	单位
V_{DD}	电源电压		-0.5	6	V
I_{VDD}	VDD 引脚的输入电流	$T = [-40, 85]^{\circ}\text{C}$	-	200	mA
		$T = [85, 105]^{\circ}\text{C}$	-	100	mA
I_{GND}	GND 引脚的输出电流	$T = [-40, 85]^{\circ}\text{C}$	-	200	mA
		$T = [85, 105]^{\circ}\text{C}$	-	100	mA
V_{RST}	RESET 引脚相对于 GND 的电压		-0.5	13	V
V_{PIN}	引脚相对于 GND 的电压		-0.5	$V_{\text{DD}}+0.5$	V
I_{PIN}	I/O 引脚灌/拉电流		-40	40	mA
$I_{\text{C1}}^{(1)}$	除 RESET 引脚外的 I/O 引脚注入电流	$V_{\text{PIN}} < \text{GND} - 0.6\text{V}$ 或 $5.5\text{V} < V_{\text{PIN}} \leq 6.1\text{V}$ $4.9\text{V} < V_{\text{DD}} \leq 5.5\text{V}$	-1	1	mA
$I_{\text{C2}}^{(1)}$	除 RESET 引脚外的 I/O 引脚注入电流	$V_{\text{PIN}} < \text{GND} - 0.6\text{V}$ 或 $V_{\text{PIN}} \leq 5.5\text{V}$ $V_{\text{DD}} \leq 4.9\text{V}$	-15	15	mA
I_{ctot}	除 RESET 引脚外的 I/O 引脚注入电流之和		-45	45	mA
T_{storage}	储存温度		-65	150	$^{\circ}\text{C}$

注:

- 如果 V_{PIN} 低于 $\text{GND} - 0.6\text{V}$ ，则需要一个限流电阻。负直流注入限流电阻的计算公式为 $R = (\text{GND} - 0.6\text{V} - V_{\text{PIN}})/I_{\text{Cn}}$ 。
如果 V_{PIN} 高于 $V_{\text{DD}}+0.6\text{V}$ ，则需要一个限流电阻。正直流注入限流电阻的计算公式为 $R = (V_{\text{PIN}} - (V_{\text{DD}} + 0.6\text{V}))/I_{\text{Cn}}$ 。



$V_{\text{RSTMAX}} = 13\text{V}$

将 RESET 引脚连接到 12V 电源时，应注意避免过冲（过电压）。使引脚工作在超过绝对最大额定值的电压下会激活引脚的 ESD 保护电路，该电路将保持激活状态，直到电压降至约 10V 以下。如果由于过电压条件而激活 ESD 保护，12V 驱动器会使 ESD 保护保持激活状态，并通过该电路来驱动电流，这可能会对器件造成永久性损坏。

34.3. 一般工作额定值

器件必须在本节列出的额定值范围内工作，才能保证器件的所有其他电气特性和典型特性有效。

表 34-2. 一般工作条件

符号	说明	条件	最小值	最大值	单位
V_{DD}	工作电源电压		1.8 ⁽²⁾	5.5	V
T	工作温度范围 ⁽¹⁾	标准温度范围	-40	85	$^{\circ}\text{C}$

注:

1. 有关器件温度范围的信息，请参见器件订购代码。
2. 对于一些器件而言， V_{BOD} 可能低于最小工作电源电压。鉴于这种情况，我们会在生产期间将 V_{DD} 降至 V_{BOD} ，然后对器件进行测试。
 - 在全片擦除期间，BOD 强制开启。如果电源电压 V_{DD} 低于配置的 V_{BOD} ，全片擦除将失败。请参见全片擦除。

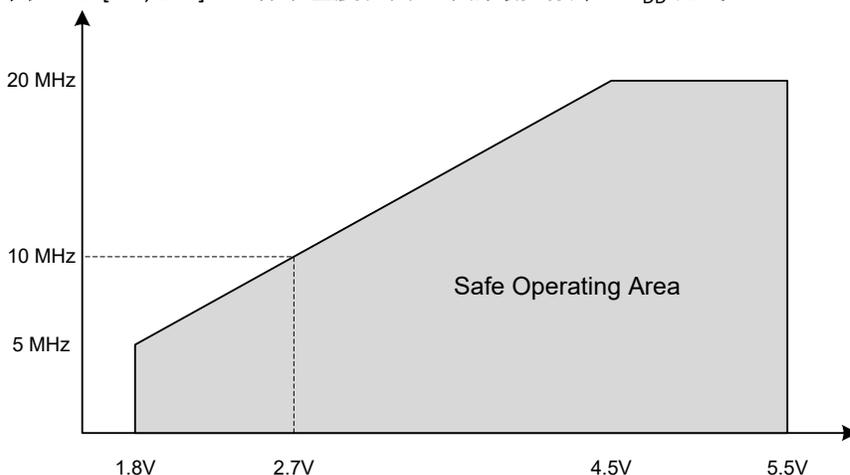
表 34-3. 工作电压和频率

符号	说明	条件	最小值	最大值	单位
CLK_CPU	工作系统时钟频率	$V_{DD} = [1.8, 5.5]V$ $T = [-40, 105]^{\circ}C^{(1)}$	0	5	MHz
		$V_{DD} = [2.7, 5.5]V$ $T = [-40, 105]^{\circ}C^{(2)}$	0	10	
		$V_{DD} = [4.5, 5.5]V$ $T = [-40, 105]^{\circ}C^{(3)}$	0	20	

注:

1. 电压下降到 BOD 触发电压 V_{BOD} (BODLEVEL0) 时可确保正常工作。
2. 电压下降到 BOD 触发电压 V_{BOD} (BODLEVEL2) 时可确保正常工作。
3. 电压下降到 BOD 触发电压 V_{BOD} (BODLEVEL7) 时可确保正常工作。

最大 CPU 时钟频率取决于 V_{DD} 。如下图所示，最大频率— V_{DD} 曲线在 $1.8V < V_{DD} < 2.7V$ 和 $2.7V < V_{DD} < 4.5V$ 范围下呈线性。

图 34-1. $[-40, 105]^{\circ}C$ (标准温度范围) 下的最大频率— V_{DD} 曲线

34.4. 功耗

功耗的相关值在以下条件下测得 (另外说明的情况除外)：

- $V_{DD} = 3V$
- $T = 25^{\circ}C$
- 除非另有说明，否则将 OSC20M 用作系统时钟源
- 在禁止外设且没有 I/O 驱动器的情况下测量的系统功耗

表 34-4. 工作和空闲模式下的功耗

模式	说明	条件	典型值	最大值	单位	
工作	工作功耗	CLK_CPU = 20 MHz (OSC20M)	V _{DD} = 5V	10	-	mA
		CLK_CPU = 10 MHz (OSC20M 2 分频)	V _{DD} = 5V	5	-	mA
			V _{DD} = 3V	3	-	mA
		CLK_CPU = 5 MHz (OSC20M 4 分频)	V _{DD} = 5V	3	-	mA
			V _{DD} = 3V	2	-	mA
			V _{DD} = 2V	1	-	mA
		CLK_CPU = 32.768 kHz (OSCUPLP32K)	V _{DD} = 5V	19	-	μA
			V _{DD} = 3V	11	-	μA
			V _{DD} = 2V	7	-	μA
		空闲	空闲功耗	CLK_CPU = 20 MHz (OSC20M)	V _{DD} = 5V	4
CLK_CPU = 10 MHz (OSC20M 2 分频)	V _{DD} = 5V			2	-	mA
	V _{DD} = 3V			1	-	mA
CLK_CPU = 5 MHz (OSC20M 4 分频)	V _{DD} = 5V			1	-	mA
	V _{DD} = 3V			1	-	mA
	V _{DD} = 2V			1	-	mA
CLK_CPU = 32.768 kHz (OSCUPLP32K)	V _{DD} = 5V			8	-	μA
	V _{DD} = 3V			4	-	μA
	V _{DD} = 2V			3	-	μA

表 34-5. 掉电、待机和复位模式下的功耗

模式	说明	条件	典型值 25°C	最大值 25°C	最大值 85°C	单位	
待机	待机功耗	RTC 通过外部 XOSC32K (CL=7.5 pF) 以 1.024 kHz 运行	V _{DD} = 3V	0.69	-	-	μA
		RTC 通过内部 OSCUPLP32K 以 1.024 kHz 运行	V _{DD} = 3V	0.71	-	-	μA
掉电/待机	所有外设均停止时， 掉电/待机功耗相同	所有外设均停止	V _{DD} = 3V	0.1	-	-	μA
复位	复位功耗	复位线下拉	V _{DD} = 3V	100	-	-	μA

34.5. 唤醒时间

从休眠模式唤醒的时间是指从唤醒信号的边沿到执行第一条指令的时间。

工作条件：

- V_{DD} = 3V
- T = 25°C
- 除非另外说明，否则 OSC20M 作为系统时钟源

表 34-6. OSC20M 的启动、复位和唤醒时间

符号	说明	条件	最小值	典型值	最大值	单位
t _{wakeup}	从任何复位释放开始的启动时间		-	200	-	μs
	从空闲模式唤醒的时间	OSC20M @ 20 MHz V _{DD} = 5V	-	1	-	
		OSC20M @ 10 MHz V _{DD} = 3V	-	2	-	
		OSC20M @ 5 MHz V _{DD} = 2V	-	4	-	
	从待机和掉电休眠模式唤醒的时间		-	10	-	

34.6. 外设功耗

下表用于计算各种工作模式下不同 I/O 外设的额外电流消耗。

工作条件:

- V_{DD} = 3V
- T = 25°C
- 1 MHz 的 OSC20M 用作系统时钟源（另外说明的情况除外）
- 空闲休眠模式下（另外说明的情况除外）

表 34-7. 外设功耗

外设	条件	典型值 ⁽¹⁾	单位
BOD	连续	19	μA
	采样（1 kHz）	1	
TCA	16 位计数（1 MHz）	13	μA
TCB	16 位计数（1 MHz）	7.5	μA
RTC	16 位计数（32.768 kHz）（OSCULP32K）	1	μA
WDT（包括 OSCULP32K）		1	μA
OSC20M		125	μA
AC	快速模式 ⁽²⁾	92	μA
	低功耗模式 ⁽²⁾	45	
ADC	50 ksps	325	μA
	100 ksps	340	
XOSC32K	C _L = 7.5 pF	0.5	μA
OSCULP32K		0.5	μA
USART	使能（9600 波特）	13	μA
SPI（主）	使能（100 kHz）	2	μA
TWI（主）	使能（100 kHz）	24	μA
TWI（从）	使能（100 kHz）	17	μA
闪存编程	擦除操作	1.5	mA
	写操作	3.0	

注:

1. 仅为模块的电流消耗。要计算系统的总功耗，需将该值与功耗中列出的基本功耗相加。
2. CPU 处于待机休眠模式。

34.7. BOD 和 POR 特性

表 34-8. 电源特性

符号	说明	条件	最小值	典型值	最大值	单位
SRON	上电斜率		-	-	100	V/ms

表 34-9. 上电复位 (POR) 特性

符号	说明	条件	最小值	典型值	最大值	单位
V _{POR}	V _{DD} 下降时的 POR 阈值电压	V _{DD} 以 0.5 V/ms 或更慢的斜率下降/上升	0.8	1.5	-	V
	V _{DD} 上升时的 POR 阈值电压		1.4	1.5	-	

表 34-10. 欠压检测 (BOD) 特性

符号	说明	条件	最小值	典型值	最大值	单位
V _{BOD}	BOD 触发电压 (下降/上升)	BODLEVEL7	-	4.2	-	V
		BODLEVEL2	-	2.6	-	
		BODLEVEL0	-	1.8	-	
V _{VLM}	相对于 BOD 触发电压的 VLM 阈值	BOD.VLMLVL = 0x0	-	4	-	%
		BOD.VLMLVL = 0x1	-	13	-	
		BOD.VLMLVL = 0x2	-	25	-	
V _{HYS}	滞后	BODLEVEL7	-	80	-	mV
		BODLEVEL2	-	40	-	
		BODLEVEL0	-	25	-	
T _{BOD}	检测时间	连续	-	7	-	μs
		采样, 1 kHz	-	1	-	ms
		采样, 125 Hz	-	8	-	
T _{Start}	启动时间	从使能到就绪的时间	-	40	-	μs

34.8. 外部复位特性

表 34-11. 外部复位特性

符号	说明	条件	最小值	典型值	最大值	单位
V _{HVRST}	确保检测到高电压复位的电压		11.5	-	12.5	V
V _{RST_VIH}	RESET 的输入高电压		0.8 × V _{DD}	-	V _{DD} + 0.2	
V _{RST_VIL}	RESET 的输入低电压		-0.2	-	0.2 × V _{DD}	
t _{RST}	RESET 引脚的最小脉冲宽度		-	-	2.5	μs
R _{RST}	RESET 上拉电阻	V _{Reset} = 0V	-	36	-	kΩ

34.9. 振荡器和时钟

工作条件:

- V_{DD} = 3V (另外说明的情况除外)

表 34-12. 内部振荡器 (OSC20M) 特性

符号	说明	条件	最小值	典型值	最大值	单位	
f _{OSC20M}	选择 16 MHz 和 20 MHz 频率时的精度	出厂时经过校准	T = 25°C, 3.0V	-	< ±3	-	%
			整个工作范围 ⁽³⁾	-	< ±5	-	
f _{CAL}	用户校准范围	OSC20M ⁽²⁾ = 16 MHz	-	16	-	MHz	
		OSC20M ⁽²⁾ = 20 MHz	-	20	-		

表 34-12. 内部振荡器 (OSC20M) 特性 (续)

符号	说明	条件	最小值	典型值	最大值	单位
%CAL	校准步长		-	1.5	-	%
DC	占空比		-	50	-	%
T _{Start}	启动时间	精度在 2%以内	-	8	-	μs

注:

1. 有关校准的信息, 请参见 OSC20M 的说明。
2. 如果振荡器频率高于速度规范, 必须对其进行分频以确保 CPU 时钟始终符合规范。
3. 这些值仅供设计参考, 不在生产测试限值范围内。

表 34-13. 32.768 kHz 内部振荡器 (OSCULP32K) 特性

符号	说明	条件	最小值	典型值	最大值	单位
f _{OSCULP32K}	精度	出厂时经过校准				
		T = 25°C, 3.0V	-	< ±3	--	%
		整个工作范围 ⁽¹⁾	-	< ±30	-	
DC	占空比		-	50	-	%
T _{Start}	启动时间		-	250	-	μs

注:

1. 这些值仅供设计参考, 不在生产测试限值范围内。

表 34-14. 32.768 kHz 外部晶振 (XOSC32K) 特性

符号	说明	条件	最小值	典型值	最大值	单位
F _{out}	频率		-	32.768	-	kHz
T _{Start}	启动时间	C _L = 7.5 pF	-	300	-	ms
C _L	晶振负载电容		-	< 12.5 ⁽¹⁾	-	pF
C _{TOSC1}	寄生电容负载		-	5.5	-	pF
C _{TOSC2}			-	5.5	-	pF
ESR	等效串联电阻——安全系数 = 3	C _L = 7.5 pF	-	-	80	kΩ
		C _L = 12.5 pF	-	-	40	

注:

1. 不建议使用所需负载电容高于表中所示值的晶振。表中给出的值仅供设计参考。

图 34-2. 外部时钟波形特性

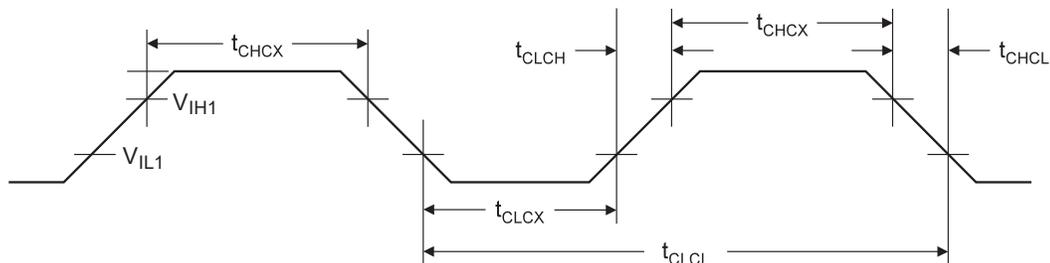


表 34-15. 外部时钟特性

符号	说明	条件	V _{DD} = [1.8, 5.5]V		V _{DD} = [2.7, 5.5]V		V _{DD} = [4.5, 5.5]V		单位
			最小值	最大值	最小值	最大值	最小值	最大值	
f _{CLCL}	频率		0	5.0	0.0	10.0	0.0	20.0	MHz

表 34-15. 外部时钟特性 (续)

符号	说明	条件	$V_{DD} = [1.8, 5.5]V$		$V_{DD} = [2.7, 5.5]V$		$V_{DD} = [4.5, 5.5]V$		单位
			最小值	最大值	最小值	最大值	最小值	最大值	
t_{CLCL}	时钟周期		200	-	100	-	50	-	ns
t_{CHCX}	高电平时间		80	-	40	-	20	-	ns
t_{CLCX}	低电平时间		80	-	40	-	20	-	ns

34.10. I/O 引脚特性

表 34-16. I/O 引脚特性 (除非另外说明, 否则 $T_A = [-40, 105]^{\circ}C$, $V_{DD} = [1.8, 5.5]V$)

符号	说明	条件	最小值	典型值	最大值	单位
V_{IL}	除用作 I/O 的 RESET 引脚外的 I/O 引脚输入低电压		-0.2	-	$0.3 \times V_{DD}$	V
V_{IH}	除用作 I/O 的 RESET 引脚外的 I/O 引脚输入高电压		$0.7 \times V_{DD}$	-	$V_{DD} + 0.2V$	V
I_{IH}/I_{IL}	除用作 I/O 的 RESET 引脚外的 I/O 引脚输入泄漏电流	$V_{DD} = 5.5V$, 引脚为高电平	-	< 0.05	-	μA
		$V_{DD} = 5.5V$, 引脚为低电平	-	< 0.05	-	
V_{OL}	I/O 引脚驱动能力	$V_{DD} = 1.8V$, $I_{OL} = 1.5 mA$	-	<0.36	-	V
		$V_{DD} = 3.0V$, $I_{OL} = 7.5 mA$	-	<0.6	-	
		$V_{DD} = 5.0V$, $I_{OL} = 15 mA$	-	<1	-	
V_{OH}	I/O 引脚驱动能力	$V_{DD} = 1.8V$, $I_{OH} = 1.5 mA$	-	>1.44	-	V
		$V_{DD} = 3.0V$, $I_{OH} = 7.5 mA$	-	>2.4	-	
		$V_{DD} = 5.0V$, $I_{OH} = 15 mA$	-	>4	-	
I_{total}	每个引脚组的最大组合 I/O 灌电流 ⁽¹⁾		-	-	100	mA
	每个引脚组的最大组合 I/O 拉电流 ⁽¹⁾		-	-	100	
V_{IL2}	用作 I/O 的 RESET 引脚的输入低电压		-0.2	-	$0.3 \times V_{DD}$	V
V_{IH2}	用作 I/O 的 RESET 引脚的输入高电压		$0.7 \times V_{DD}$	-	$V_{DD} + 0.2V$	V
V_{OL2}	用作 I/O 的 RESET 引脚的 I/O 引脚驱动能力	$V_{DD} = 1.8V$, $I_{OL} = 0.1 mA$	-	<0.36	-	V
		$V_{DD} = 3.0V$, $I_{OL} = 0.25 mA$	-	<0.6	-	
		$V_{DD} = 5.0V$, $I_{OL} = 0.5 mA$	-	<1	-	

表 34-16. I/O 引脚特性（除非另外说明，否则 $T_A = [-40, 105]^{\circ}\text{C}$ ， $V_{DD} = [1.8, 5.5]\text{V}$ ）（续）

符号	说明	条件	最小值	典型值	最大值	单位
V_{OH2}	用作 I/O 的 $\overline{\text{RESET}}$ 引脚的 I/O 引脚驱动能力	$V_{DD} = 1.8\text{V}$, $I_{OH} = 0.1\text{ mA}$	-	>1.44	-	V
		$V_{DD} = 3.0\text{V}$, $I_{OH} = 0.25\text{ mA}$	-	>2.4	-	
		$V_{DD} = 5.0\text{V}$, $I_{OH} = 0.5\text{ mA}$	-	>4	-	
t_{RISE}	上升时间	$V_{DD} = 3.0\text{V}$, 负载 = 20 pF	-	2.5	-	ns
		$V_{DD} = 5.0\text{V}$, 负载 = 20 pF	-	1.5	-	
t_{FALL}	下降时间	$V_{DD} = 3.0\text{V}$, 负载 = 20 pF	-	2.0	-	ns
		$V_{DD} = 5.0\text{V}$, 负载 = 20 pF	-	1.3	-	
C_{PIN}	除 TOSC 和 TWI 引脚外的 I/O 引脚电容		-	3	-	pF
C_{PIN}	TOSC 引脚的 I/O 引脚电容		-	5.5	-	pF
C_{PIN}	TWI 引脚的 I/O 引脚电容		-	10	-	pF
R_p	上拉电阻		-	35	-	k Ω

注:

1. 引脚组 x (Px[7:0])。所有 I/O 端口的组合连续灌/拉电流不得超过相应限值。

34.11. TCD

工作条件:

- 如果 CLK_TCD 频率高于最大 CLK_TCD_SYNC，则必须使用同步预分频比（TCDn.CTRLA 中的 SYNCRES）进行预分频，以使同步器时钟满足这些规范

表 34-17. 定时器/计数器 D 最大频率⁽¹⁾

符号	说明	条件	最大值	单位
$f_{\text{CLK_TCD_SYNC}}$	CLK_TCD_SYNC 最大频率	$V_{DD} = [1.8, 5.5]\text{V}$	8	MHz
		$V_{DD} = [2.7, 5.5]\text{V}$	16	
		$V_{DD} = [4.5, 5.5]\text{V}$	32	

注:

1. 这些参数仅供设计参考，不在生产测试限值范围内。

34.12. USART

图 34-3. USART 处于 SPI 模式——主模式下的时序要求

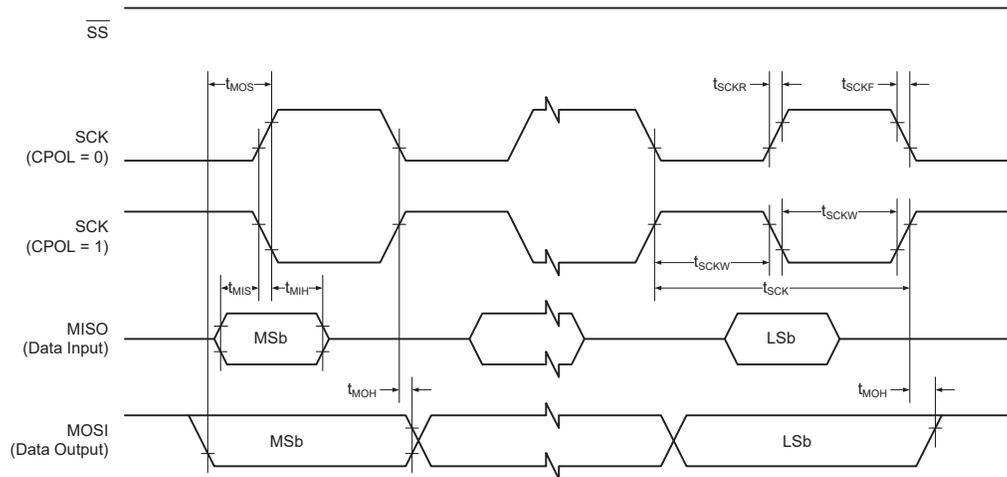


图 34-4. USART 处于 SPI 模式——从模式下的时序要求

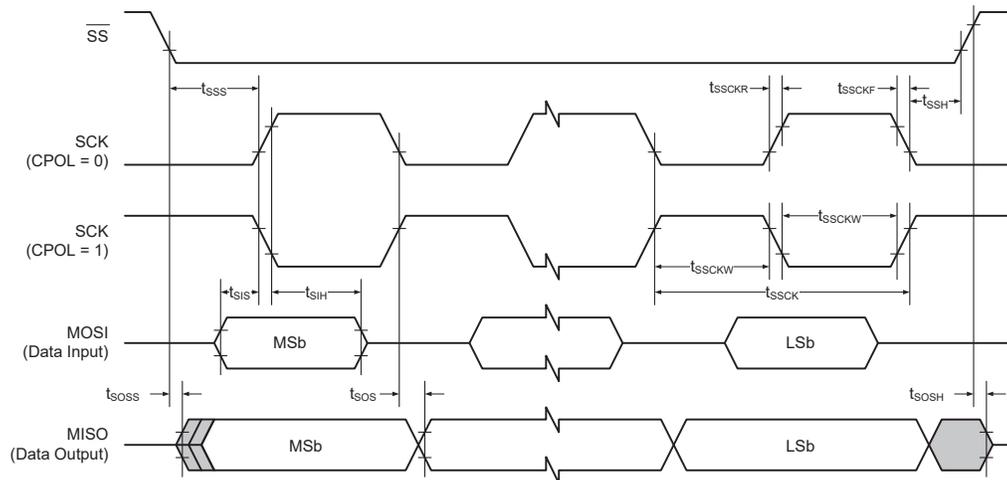


表 34-18. USART 处于 SPI 主模式——时序特性

符号	说明	条件	最小值	典型值	最大值	单位
f_{SCK}	SCK 时钟频率	主模式	-	-	10	MHz
t_{SCK}	SCK 周期	主模式	100	-	-	ns
t_{SCKW}	SCK 高/低电平宽度	主模式	-	$0.5 \times t_{SCK}$	-	ns
t_{SCKR}	SCK 上升时间	主模式	-	2.7	-	ns
t_{SCKF}	SCK 下降时间	主模式	-	2.7	-	ns
t_{MIS}	从 MISO 建立到 SCK 的时间	主模式	-	10	-	ns
t_{MIH}	SCK 后的 MISO 保持时间	主模式	-	10	-	ns
t_{MOS}	从 MOSI 建立到 SCK 的时间	主模式	-	$0.5 \times t_{SCK}$	-	ns
t_{MOH}	SCK 后的 MOSI 保持时间	主模式	-	1.0	-	ns

表 34-19. SPI——时序特性（续）

符号	说明	条件	最小值	典型值	最大值	单位
t_{SSCKR}	SCK 上升时间	从模式	-	-	1600	ns
t_{SSCKF}	SCK 下降时间	从模式	-	-	1600	ns
t_{SIS}	从 MOSI 建立到 SCK 的时间	从模式	3.0	-	-	ns
t_{SIH}	SCK 后的 MOSI 保持时间	从模式	t_{CLK_PER}	-	-	ns
t_{SSS}	SS 建立到 SCK 的时间	从模式	21	-	-	ns
t_{SSH}	SCK 后的 SS 保持时间	从模式	20	-	-	ns
t_{SOS}	从 MISO 建立到 SCK 的时间	从模式	-	8.0	-	ns
t_{SOH}	SCK 后的 MISO 保持时间	从模式	-	13	-	ns
t_{SOSS}	SS 低电平后的 MISO 建立时间	从模式	-	11	-	ns
t_{SOSh}	SS 低电平后的 MISO 保持时间	从模式	-	8.0	-	ns

34.14. TWI

图 34-7. TWI——时序要求

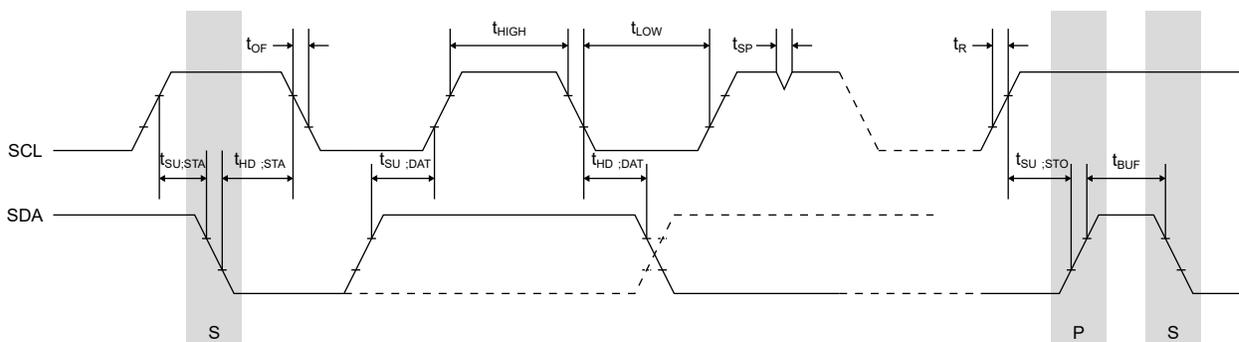


表 34-20. TWI——时序特性

符号	说明	条件	最小值	典型值	最大值	单位	
f_{SCL}	SCL 时钟频率	最大频率需要以 10 MHz 运行的系统时钟，而该时钟又需要 $V_{DD} = [2.7, 5.5]V$, $T = [-40, 105]^{\circ}C$	0	-	400	kHz	
V_{IH}	输入高电压		$0.7 \times V_{DD}$	-	-	V	
V_{IL}	输入低电压		-	-	$0.3 \times V_{DD}$	V	
V_{HYS}	施密特触发器输入滞后		$0.1 \times V_{DD}$	-	$0.4 \times V_{DD}$	V	
V_{OL}	输出低电压	$I_{load} = 3\text{ mA}$, 正常模式, $V_{DD} > 2V$	-	-	0.4	V	
		$I_{load} = 3\text{ mA}$, 正常模式, $V_{DD} \leq 2V$	-	-	$0.2 \times V_{DD}$	V	
I_{OL}	低电平输出电流	$f_{SCL} \leq 400\text{ kHz}$, $V_{OL} = 0.4V$	3	-	-	mA	
C_B	每条总线的容性负载	$f_{SCL} \leq 100\text{ kHz}$	-	-	400	pF	
		$f_{SCL} \leq 400\text{ kHz}$	-	-	400	pF	
t_R	SDA 和 SCL 的上升时间	$f_{SCL} \leq 100\text{ kHz}$	-	-	1000	ns	
		$f_{SCL} \leq 400\text{ kHz}$	20	-	300	ns	
t_{OF}	V_{IHmin} 到 V_{ILmax} 的输出下降时间	$10\text{ pF} < \text{总线电容} < 400\text{ pF}$	$f_{SCL} \leq 400\text{ kHz}$	$20 \times (V_{DD}/5.5V)$	-	250	ns
t_{SP}	输入滤波器抑制的尖峰时间		0	-	50	ns	
I_L	每个 I/O 引脚的输入电流	$0.1 \times V_{DD} < V_I < 0.9 \times V_{DD}$	-	-	1	μA	
C_I	每个 I/O 引脚的电容		-	-	10	pF	

表 34-20. TWI——时序特性 (续)

符号	说明	条件	最小值	典型值	最大值	单位
R _p	上拉电阻的值	f _{SCL} ≤ 100 kHz	(V _{DD} - V _{OL(max)}) / I _{OL}	-	1000 ns / (0.8473 × C _B)	Ω
		f _{SCL} ≤ 400 kHz	-	-	300 ns / (0.8473 × C _B)	
t _{HD;STA}	(重复) 启动条件的保持时间	f _{SCL} ≤ 100 kHz	4.0	-	-	μs
		f _{SCL} ≤ 400 kHz	0.6	-	-	
		启动	-	2.1	-	T _{SCL}
		重复启动	-	3.1	-	T _{SCL}
t _{LOW}	SCL 时钟的低电平周期	f _{SCL} ≤ 100 kHz	4.7	-	-	μs
		f _{SCL} ≤ 400 kHz	1.3	-	-	
t _{HIGH}	SCL 时钟的高电平周期	f _{SCL} ≤ 100 kHz	4.0	-	-	μs
		f _{SCL} ≤ 400 kHz	0.6	-	-	
t _{SU;STA}	重复启动条件的建立时间	f _{SCL} ≤ 100 kHz	4.7	-	-	μs
		f _{SCL} ≤ 400 kHz	0.6	-	-	
		-	-	3	-	T _{SCL}
t _{HD;DAT}	数据保持时间	f _{SCL} ≤ 100 kHz	0	-	3.45	μs
		f _{SCL} ≤ 400 kHz	0	-	0.9	
t _{SU;DAT}	数据建立时间	f _{SCL} ≤ 100 kHz	250	-	-	ns
		f _{SCL} ≤ 400 kHz	100	-	-	
t _{SU;STO}	停止条件的建立时间	f _{SCL} ≤ 100 kHz	4	-	-	μs
		f _{SCL} ≤ 400 kHz	0.6	-	-	
		-	-	2	-	T _{SCL}
t _{BUF}	停止条件和启动条件之间的总线空闲时间	f _{SCL} ≤ 100 kHz	4.7	-	-	μs
		f _{SCL} ≤ 400 kHz	1.3	-	-	
		-	-	2	-	T _{SCL}

表 34-21. SDA 保持时间^(1,2)

符号	说明	条件	最小值	典型值	最大值	单位		
t _{HD;DAT}	数据保持时间	主模式 ⁽³⁾	f _{CLK_PER} = 5 MHz	SDAHOLD = 0x00	-	800	-	ns
				SDAHOLD = 0x01	-	850	-	
				SDAHOLD = 0x02	-	850	-	
				SDAHOLD = 0x03	-	850	-	
			f _{CLK_PER} = 10 MHz	SDAHOLD = 0x00	-	400	-	
				SDAHOLD = 0x01	-	450	-	
		SDAHOLD = 0x02		-	450	-		
		SDAHOLD = 0x03		-	550	-		
		f _{CLK_PER} = 20 MHz	SDAHOLD = 0x00	-	200	-		
			SDAHOLD = 0x01	-	250	-		
			SDAHOLD = 0x02	-	450	-		
			SDAHOLD = 0x03	-	600	-		
t _{HD;DAT}	数据保持时间	从模式 ⁽⁴⁾	所有频率	SDAHOLD = 0x00	-	150	-	ns
				SDAHOLD = 0x01	-	200	-	
				SDAHOLD = 0x02	-	400	-	
				SDAHOLD = 0x03	-	550	-	

注:

1. 这些参数仅供设计参考，不在生产测试限值范围内。
2. SDAHOLD 是 SCL 信号检测为低电平后的数据保持时间。因此，实际保持时间长于配置的保持时间。
3. 对于主模式，数据保持时间取以下两项中的较大值：
 - $4 \times t_{CLK_PER} + 50 \text{ ns}$ （典型值）
 - SDAHOLD 配置 + SCL 滤波器延时
4. 对于从模式，保持时间如下：
 - SDAHOLD 配置 + SCL 滤波器延时

34.15. VREF**表 34-22. 内部参考电压特性**

符号	说明	最小值	典型值	最大值	单位
T_{Start}	启动时间	-	25	-	μs
$V_{DDINT055V}$	INT055V 的电源电压范围	1.8	-	5.5	V
$V_{DDINT11V}$	INT11V 的电源电压范围	1.8	-	5.5	
$V_{DDINT15V}$	INT15V 的电源电压范围	1.9	-	5.5	
$V_{DDINT25V}$	INT25V 的电源电压范围	2.9	-	5.5	
$V_{DDINT43V}$	INT43V 的电源电压范围	4.75	-	5.5	

表 34-23. ADC 内部参考电压特性⁽¹⁾

符号 ⁽²⁾	说明	条件	最小值	典型值	最大值	单位
INT11V	内部参考电压	$V_{DD} = [1.8V, 3.6V]$ $T = [0, 105]^\circ\text{C}$	-	$< \pm 2$	-	%
INT055V INT15V INT25V	内部参考电压	$V_{DD} = [1.8V, 3.6V]$ $T = [0, 105]^\circ\text{C}$	-	$< \pm 3$	-	
INT055V INT11V INT15V INT25V INT43V	内部参考电压	$V_{DD} = [1.8V, 5.5V]$ $T = [-40, 105]^\circ\text{C}$	-	$< \pm 5$	-	

注:

1. 这些值仅供设计参考，不在生产测试限值范围内。
2. 符号 INTxxV 指代 VREF.CTRLA 寄存器中 ADCOREFSEL 和 DACOREFSEL 位域的相应值。

表 34-24. DAC 和 AC 内部参考电压特性⁽¹⁾

符号 ⁽²⁾	说明	条件	最小值	典型值	最大值	单位
INT055V INT11V INT15V INT25V	内部参考电压	$V_{DD} = [1.8V, 3.6V]$ $T = [0, 105]^\circ\text{C}$	-	$< \pm 3$	-	%
INT055V INT11V INT15V INT25V INT43V	内部参考电压	$V_{DD} = [1.8V, 5.5V]$ $T = [-40, 105]^\circ\text{C}$	-	$< \pm 5$	-	

注:

1. 这些值仅供设计参考，不在生产测试限值范围内。
2. 符号 INTxxV 指代 VREF.CTRLA 寄存器中 ADCOREFSEL 和 DACOREFSEL 位域的相应值。

34.16. ADC

工作条件:

- $V_{DD} = 1.8V$ 至 $5.5V$
- 温度范围 = $-40^{\circ}C$ 至 $105^{\circ}C$
- DUTYCYC = 25%
- $CLK_{ADC} = 13 \times f_{ADC}$
- 对于 $0.55V$ 参考电压，SAMPCAP 为 10 pF ；如果 $V_{REF} \geq 1.1V$ ，SAMPCAP 设为 5 pF
- 除非另外说明，否则适用于所有允许的 V_{REF} 选择和采样率组合

表 34-25. 电源、参考和输入范围

符号	说明	条件	最小值	典型值	最大值	单位
V_{DD}	电源电压		1.8	-	5.5	V
V_{REF}	参考电压	REFSEL = 内部参考	0.55	-	$V_{DD} - 0.4$	V
		REFSEL = V_{DD}	1.8	-	5.5	
C_{IN}	输入电容	SAMPCAP = 5 pF	-	5	-	pF
		SAMPCAP = 10 pF	-	10	-	
R_{IN}	输入电阻		-	14	-	k Ω
V_{IN}	输入电压范围		0	-	V_{REF}	V
I_{BAND}	输入带宽	$1.1V \leq V_{REF}$	-	-	57.5	kHz

表 34-26. 时钟和时序特性

符号	说明	条件	最小值	典型值	最大值	单位
f_{ADC}	采样率	$1.1V \leq V_{REF}$	15	-	115	ksps
		$1.1V \leq V_{REF}$ (8 位分辨率)	15	-	150	
		$V_{REF} = 0.55V$ (10 位)	7.5	-	20	
CLK_{ADC}	时钟频率	$V_{REF} = 0.55V$ (10 位)	100	-	260	kHz
		$1.1V \leq V_{REF}$ (10 位)	200	-	1500	
		$1.1V \leq V_{REF}$ (8 位分辨率)	200	-	2000 ⁽¹⁾	
T_S	采样时间		2	2	33	CLK_{ADC} 周期
T_{CONV}	转换时间 (延时)	采样时间 = $2 CLK_{ADC}$	8.7	-	50	μs
T_{START}	启动时间	内部 V_{REF}	-	22	-	μs

注:

1. 对于超过 1500 kHz 的时钟频率，需要 50% 的占空比。

表 34-27. 精度特性⁽²⁾

符号	说明	条件	最小值	典型值	最大值	单位
RES	分辨率		-	10	-	位

表 34-27. 精度特性⁽²⁾ (续)

符号	说明	条件	最小值	典型值	最大值	单位	
INL	积分非线性	REFSEL = INTERNAL V _{REF} = 0.55V	f _{ADC} = 7.7 ksps	-	1.0	-	LSb
		REFSEL = INTERNAL 或 V _{DD}	f _{ADC} = 15 ksps	-	1.0	-	
		REFSEL = INTERNAL 或 V _{DD}	f _{ADC} = 77 ksps	-	1.0	-	
		1.1V ≤ V _{REF}	f _{ADC} = 115 ksps	-	1.2	-	
DNL ⁽¹⁾	微分非线性	REFSEL = INTERNAL V _{REF} = 0.55V	f _{ADC} = 7.7 ksps	-	0.6	-	LSb
		REFSEL = INTERNAL 或 V _{DD}	f _{ADC} = 15 ksps	-	0.4	-	
		REFSEL = INTERNAL 或 V _{DD}	f _{ADC} = 77 ksps	-	0.4	-	
		1.1V ≤ V _{REF}	f _{ADC} = 115 ksps	-	0.6	-	
		REFSEL = V _{DD}	f _{ADC} = 115 ksps	-	0.6	-	
EABS	绝对精度	REFSEL = INTERNAL V _{REF} = 1.1V	T = [0, 105]°C V _{DD} = [1.8V, 3.6V] V _{DD} = [1.8V, 3.6V]	-	3	-	LSb
		REFSEL = V _{DD}		-	2	-	
		REFSEL = INTERNAL		-	3	-	
				-	3	-	
EGAIN	增益误差	REFSEL = INTERNAL V _{REF} = 1.1V	T = [0, 105]°C V _{DD} = [1.8V, 3.6V] V _{DD} = [1.8V, 3.6V]	-	5	-	LSb
		REFSEL = V _{DD}		-	2	-	
		REFSEL = INTERNAL		-	5	-	
				-	5	-	
E0FF	失调误差			-	-0.5	-	LSb

注:

1. DNL 误差 ≤ 1 LSb 可确保单调传递函数，并且不会丢失编码。
2. 这些值基于特性，不在生产测试限值范围内。

34.17. TEMPSENSE

表 34-28. 温度传感器精度特性

符号	说明	条件	最小值	典型值	最大值	单位
V _{DD}	电源电压		1.8	-	5.5	V
T _{ACC}	传感器精度 ⁽¹⁾	T _A = 25°C	-	±3	-	°C
T _{RES}	转换分辨率	10 位	-	0.55	-	°C
t _{CONV}	转换时间	1 MHz ADC 时钟	-	13	-	μs

注:

1. 这些值仅供设计参考，不在生产测试限值范围内。

34.18. DAC

除非另外声明，否则 V_{DD} = 3V。

精度特性基于 DAC 的 5%到 95%范围计算。

表 34-29. 电源和输出特性

符号	说明	最小值	典型值	最大值	单位
V_{DD}	电源电压 ⁽¹⁾	1.8	3	5.5	V
R_{Load}	阻性外部负载	-	>5	-	k Ω
C_{Load}	容性外部负载	-	<30	-	pF
V_{OUT}	输出电压范围	0.2	-	$V_{DD}-0.2$	V
I_{OUT}	输出灌/拉电流	-	1	-	mA

注：1.电源电压必须满足 V_{DD} 规范，以便将 V_{REF} 用作 DAC 参考。

表 34-30. 时钟和时序特性

符号	说明	条件	最小值	典型值	最大值	单位
f_{DAC}	最大转换速率	$0.55V \leq V_{REF} \leq 2.5V$	-	350	-	ksps
		$V_{REF} = 4.3V$	-	270	-	ksps

表 34-31. 精度特性⁽³⁾

符号	说明	条件	最小值	典型值	最大值	单位
RES	分辨率		-		8	位
INL	积分非线性	$0.55V \leq V_{REF} \leq 4.3V$	-	0.3	-	LSb
DNL	微分非线性	$0.55V \leq V_{REF} \leq 4.3V$	-	0.25	-	LSb
E _{OFF} ⁽¹⁾	失调误差	$0.55V \leq V_{REF} \leq 1.5V$	-	-3	-	mV
		$V_{REF} = 2.5V$	-	-6	-	
		$V_{REF} = 4.3V$	-	-10	-	
E _{GAIN} ⁽²⁾	增益误差	$V_{REF} = 1.1V, V_{DD} = 3.0V, T = 25^{\circ}C$	-	± 1	-	LSb
		$0.55V \leq V_{REF} \leq 4.3V$	-	-1	-	

注：

1. 该失调包含 DAC 输出缓冲器失调，在 DAC 输出引脚处测得。
2. V_{REF} 精度包含在增益精度规范中。
3. 这些值仅供设计参考，不在生产测试限值范围内。

34.19. AC

表 34-32. 模拟比较器特性（禁止低功耗模式）

符号	说明	条件	最小值	典型值	最大值	单位
V_{IN}	输入电压		-0.2	-	V_{DD}	V
C_{IN}	输入引脚电容	PA6	-	9	-	pF
		PA7、PB5 和 PB4	-	5	-	
V_{OFF}	输入失调电压	$0.7V < V_{IN} < (V_{DD} - 0.7V)$	-	± 5	-	mV
		$V_{IN} = [-0.2V, V_{DD}]$	-	± 20	-	
I_L	输入泄漏电流		-	5	-	nA
T_{START}	启动时间		-	1.3	-	μs
V_{HYS}	滞后	HYSMODE = 0x0	-	0	-	mV
		HYSMODE = 0x1	-	10	-	
		HYSMODE = 0x2	-	30	-	
		HYSMODE = 0x3	-	55	-	
t_{PD}	传播延时	25 mV 过驱动电压, $V_{DD} \geq 2.7V$, 禁止低功耗模式	-	50	-	ns

表 34-33. 模拟比较器特性（使能低功耗模式）

符号	说明	条件	最小值	典型值	最大值	单位
V_{IN}	输入电压		0	-	V_{DD}	V
C_{IN}	输入引脚电容	PA6	-	9	-	pF
		PA7、PB5 和 PB4	-	5	-	
V_{OFF}	输入失调电压	$0.7V < V_{IN} < (V_{DD} - 0.7V)$	-	± 10	-	mV
		$V_{IN} = [0V, V_{DD}]$	-	± 30	-	
I_L	输入泄漏电流		-	5	-	nA
T_{START}	启动时间		-	1.3	-	μs
V_{HYS}	滞后	HYSMODE = 0x0	-	0	-	mV
		HYSMODE = 0x1	-	10	-	
		HYSMODE = 0x2	-	30	-	
		HYSMODE = 0x3	-	55	-	
t_{PD}	传播延时	25 mV 过驱动电压, $V_{DD} \geq 2.7V$	-	150	-	ns

34.20. PTC

表 34-34. 外设触摸控制器特性——工作额定值

符号	说明	条件	最小值	典型值	最大值	单位
C_{LOAD}	最大负载		-	48	-	pF
C_{INT}	积分电容的最大大小		-	30	-	pF
C_{DS}	驱动屏蔽容性驱动		-	300	-	pF
CLK_{ADC}	支持的 ADC 时钟频率	25% 占空比	200	-	1500	kHz
		50% 占空比	200	-	2000	

表 34-35. 外设触摸控制器特性——焊盘电容

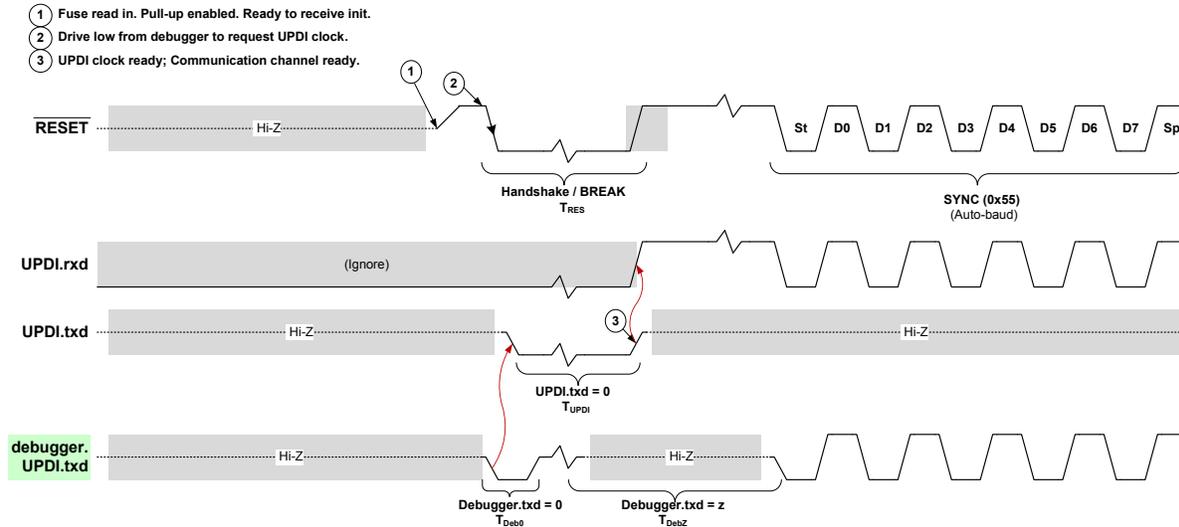
符号	说明	条件	最小值	典型值	最大值	单位
$C_{X/Y}$	焊盘电容 X/Y 线路	PA4, X0/Y0	-	4	-	pF
		PA5, X1/Y1	-	24	-	
		PA6, X2/Y2	-	9	-	
		PA7, X3/Y3	-	6	-	
		PB5, X12/Y12	-	4	-	
		PB4, X13/Y13	-	4	-	
		PB1, X4/Y4	-	13	-	
		PB0, X5/Y5	-	13	-	
		PC0, X6/Y6	-	6	-	
		PC1, X7/Y7	-	6	-	
		PC2, X8/Y8	-	6	-	
		PC3, X9/Y9	-	6	-	
		PC4, X10/Y10	-	6	-	
		PC5, X11/Y11	-	6	-	

注:

1. 低引脚数器件上只提供部分端口引脚。有关详细信息，请查看引脚图和/或 I/O 复用表。

34.21. UPDI 时序

通过熔丝使能 UPDI 焊盘时的 UPDI 使能序列⁽¹⁾

表 34-36. UPDI 时序特性⁽¹⁾

符号	说明	最小值	最大值	单位
T_{RES}	RESET 引脚上握手/断路的持续时间	10	200	μs
T_{UPDI}	UPDI.txd = 0 的持续时间	10	200	μs
T_{Deb0}	Debugger.txd = 0 的持续时间	0.2	1	μs
T_{DebZ}	Debugger.txd = z 的持续时间	200	14000	μs

注:

1. 这些参数仅供设计参考, 不在生产测试限值范围内。

表 34-37. UPDI 最大比特率与 V_{DD} 的关系⁽¹⁾

符号	说明	条件	最大值	单位
f_{UPDI}	UPDI 波特率	$V_{DD} = [1.8, 5.5]\text{V}$ $T_A = [0, 50]^\circ\text{C}$	225	kbps
		$V_{DD} = [2.2, 5.5]\text{V}$ $T_A = [0, 50]^\circ\text{C}$	450	kbps
		$V_{DD} = [2.7, 5.5]\text{V}$ $T_A = [0, 50]^\circ\text{C}$	0.9	Mbps

注:

1. 这些参数仅供设计参考, 不在生产测试限值范围内。

34.22. 编程时间

有关闪存和 EEPROM 的典型编程时间, 请参见下表。

表 34-38. 编程时间

符号	典型编程时间
页缓冲区清除 (PBC)	7 个 CLK_CPU 周期
页写 (WP)	2 ms
页擦除 (ER)	2 ms
页擦除-写 (ERWP)	4 ms
EEPROM 擦除 (EER)	4 ms

表 34-38. 编程时间（续）

符号	典型编程时间
通过 UPDI 进行全片擦除	80 ms ⁽¹⁾
	150 ms ⁽²⁾
	280 ms ⁽³⁾

注:

1. 仅适用于 CN0816。
2. 仅适用于 CN1616。
3. 仅适用于 CN3217。

35. 散热注意事项

35.1. 热阻数据

下表总结了与封装相关的热阻数据。

表 35-1. 热阻数据

引脚数	封装类型	θ_{JA} [°C/W]	θ_{JC} [°C/W]
20	VQFN	79.7	36
24	VQFN	60.6	25

35.2. 结温

可通过以下公式计算芯片平均结温 T_J （单位为°C）：

- 公式 1: $T_J = T_A + (P_D \times \theta_{JA})$
- 公式 2: $T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$

其中：

- θ_{JA} = 结到环境的封装热阻（°C/W），请参见 表 35-1
- θ_{JC} = 结到外壳的封装热阻（°C/W），请参见 表 35-1
- $\theta_{HEATSINK}$ = 外部冷却器的热阻（°C/W）规范
- P_D = 器件功耗（W）
- T_A = 环境温度（°C）

根据第一个公式，用户可以推导出芯片的预计使用寿命，进而决定是否需要冷却器。如果必须在芯片上安装冷却器，则应使用第二个公式计算最终的芯片平均结温 T_J （单位为°C）。

可通过将系统功耗和 I/O 模块功耗相加来计算功耗。一个具有容性负载的引脚所消耗的电流可按照以下方式估算：

$$I_{cp} \approx V_{DD} * C_{load} * f_{sw}$$

其中， C_{load} = 引脚负载电容， f_{sw} = I/O 引脚的平均开关频率。

36. 产品信息

订购代码	闪存/SRAM	引脚数	最大 CPU 速度	电源电压	封装类型 ⁽⁴⁾	温度范围
CN0816-MNR ⁽¹⁾	8 KB/512B	20	20 MHz	1.8-5.5V	VQFN	-40°C 至+105°C
CN1616-MNR ⁽²⁾	16 KB/512B	20	20 MHz	1.8-5.5V	VQFN	-40°C 至+105°C
CN3217-MNR ⁽³⁾	32 KB/512B	24	20 MHz	1.8-5.5V	VQFN	-40°C 至+105°C

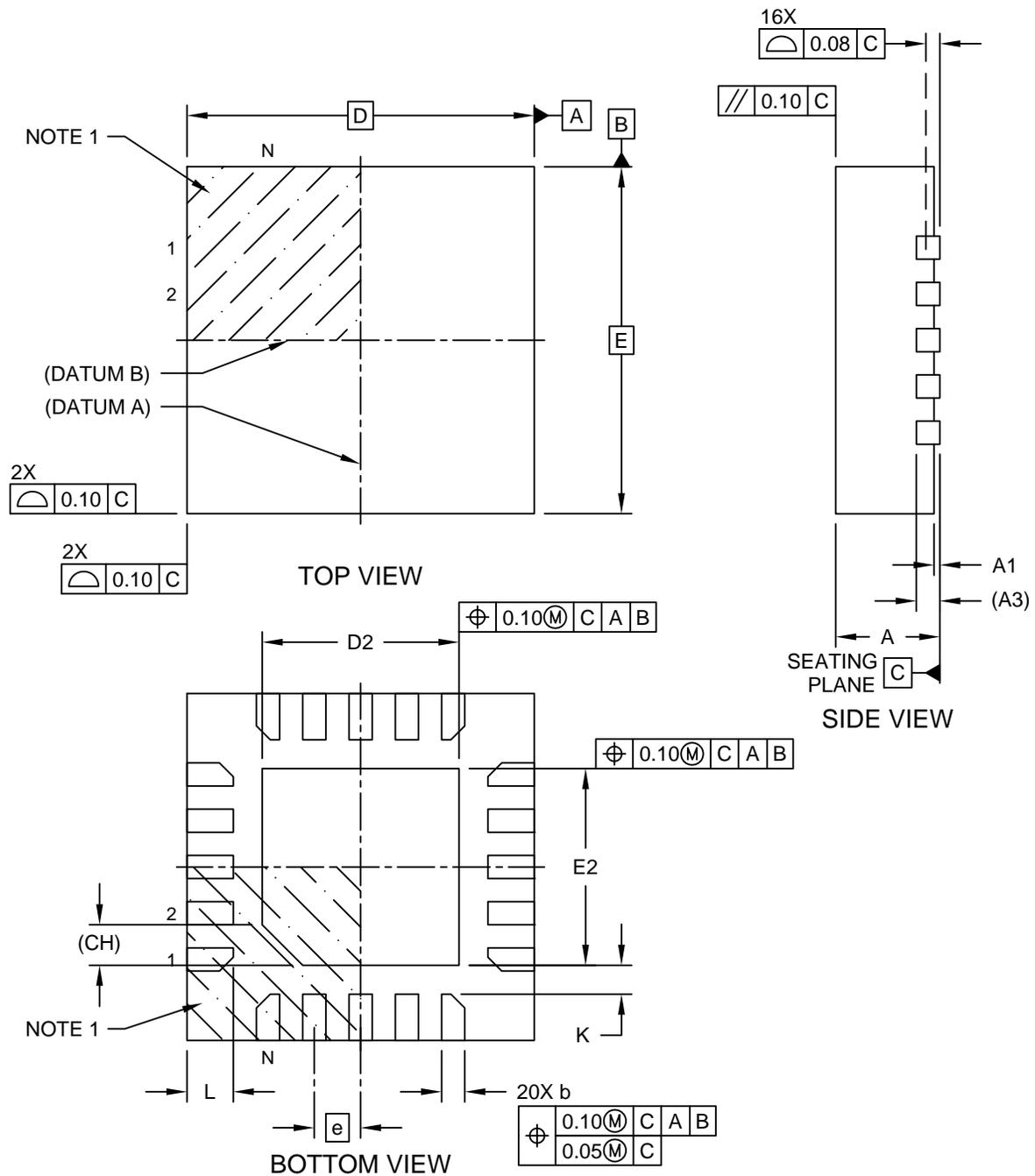
注:

1. 要对 CN0816 进行编程，请选择 ATtiny816。
2. 要对 CN1616 进行编程，请选择 ATtiny1616。
3. 要对 CN3217 进行编程，请选择 ATtiny3217。
4. 提供卷带式包装。

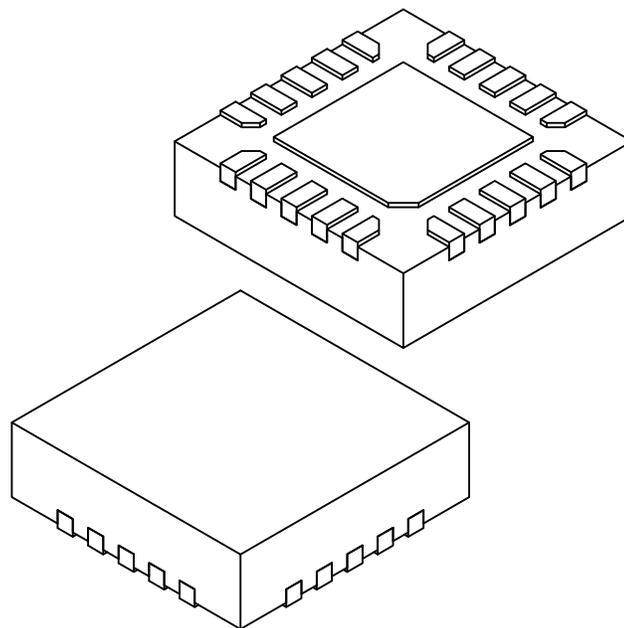
37. 封装图

37.1. 20 引脚 VQFN

20-Lead Very Thin Plastic Quad Flat, No Lead Package (REB) - 3x3 mm Body [VQFN]
With 1.7 mm Exposed Pad



20-Lead Very Thin Plastic Quad Flat, No Lead Package (REB) - 3x3 mm Body [VQFN] With 1.7 mm Exposed Pad

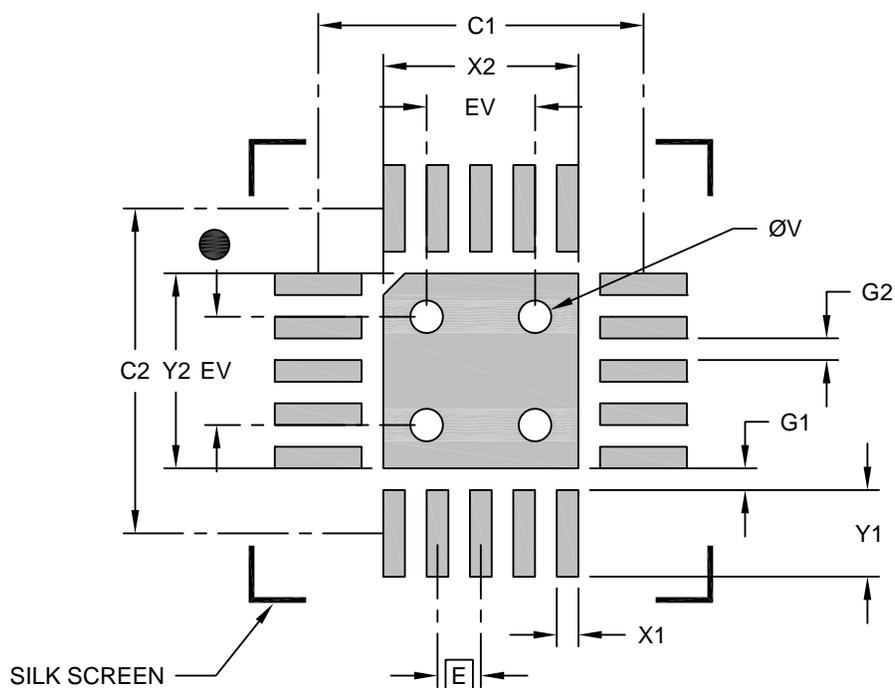


Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Terminals	N	20		
Pitch	e	0.40 BSC		
Overall Height	A	0.80	0.85	0.90
Standoff	A1	0.00	0.035	0.05
Terminal Thickness	A3	0.203 REF		
Overall Length	D	3.00 BSC		
Exposed Pad Length	D2	1.60	1.70	1.80
Overall Width	E	3.00 BSC		
Exposed Pad Width	E2	1.60	1.70	1.80
Terminal Width	b	0.15	0.20	0.25
Terminal Length	L	0.35	0.40	0.45
Terminal-to-Exposed-Pad	K	0.20	-	-
Pin 1 Index Chamfer	CH	0.35 REF		

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated
- Dimensioning and tolerancing per ASME Y14.5M
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 REF: Reference Dimension, usually without tolerance, for information purposes only.

20-Lead Very Thin Plastic Quad Flat, No Lead Package (REB) - 3x3 mm Body [VQFN] With 1.7 mm Exposed Pad



RECOMMENDED LAND PATTERN

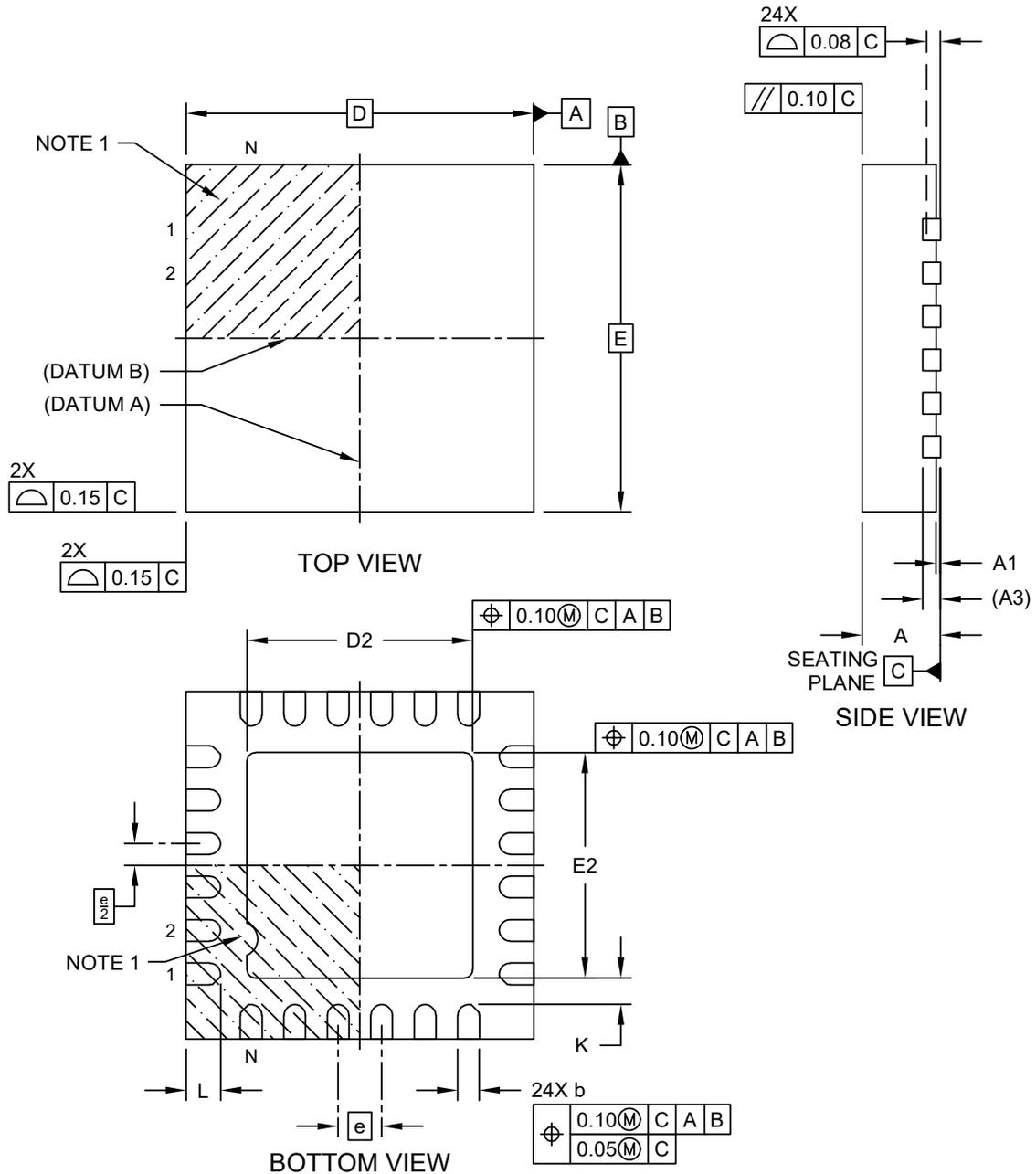
Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.40 BSC		
Optional Center Pad Width	X2			1.80
Optional Center Pad Length	Y2			1.80
Contact Pad Spacing	C1		3.00	
Contact Pad Spacing	C2		3.00	
Contact Pad Width (X20)	X1			0.20
Contact Pad Length (X20)	Y1			0.80
Contact Pad to Center Pad (X20)	G1	0.20		
Contact Pad to Contact Pad (X16)	G2	0.20		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

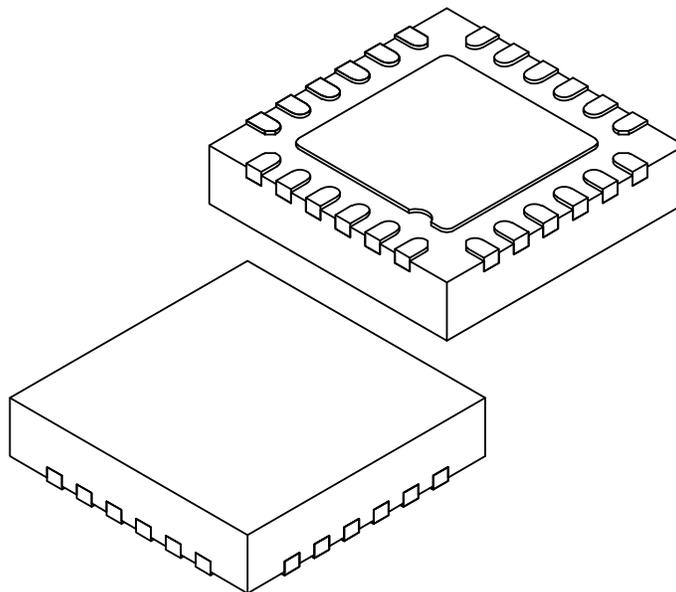
- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
- For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process.

37.2. 24 引脚 VQFN

24-Lead Very Thin Plastic Quad Flat, No Lead Package (RLB) - 4x4 mm Body [VQFN]



24-Lead Very Thin Plastic Quad Flat, No Lead Package (RLB) - 4x4 mm Body [VQFN]

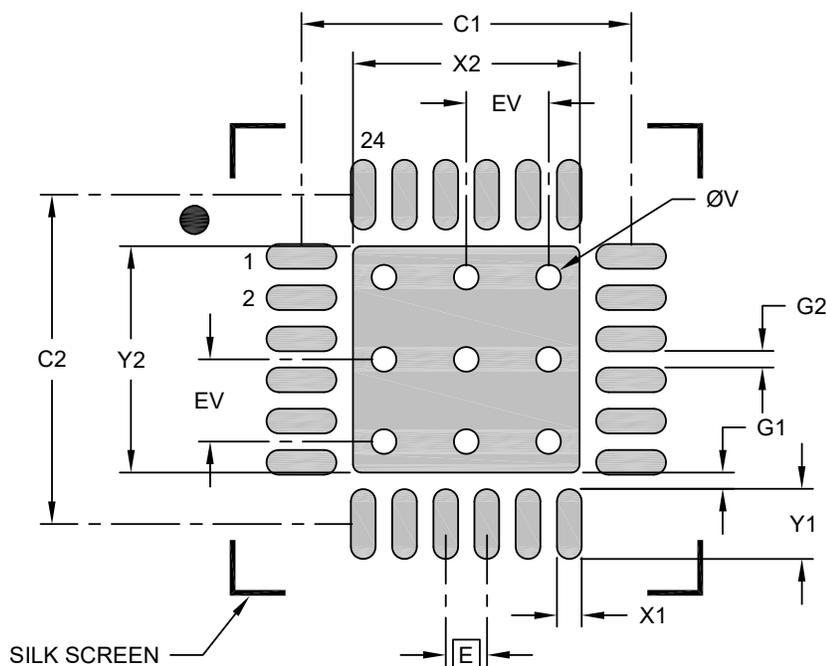


Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Terminals	N	24		
Pitch	e	0.50 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	-	0.05
Terminal Thickness	A3	0.203 REF		
Overall Length	D	4.00 BSC		
Exposed Pad Length	D2	2.45	2.60	2.75
Overall Width	E	4.00 BSC		
Exposed Pad Width	E2	2.45	2.60	2.75
Terminal Width	b	0.18	0.25	0.30
Terminal Length	L	0.35	0.40	0.45
Terminal-to-Exposed-Pad	K	0.20	-	-

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated
- Dimensioning and tolerancing per ASME Y14.5M
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 REF: Reference Dimension, usually without tolerance, for information purposes only.

24-Lead Very Thin Plastic Quad Flat, No Lead Package (RLB) - 4x4 mm Body [VQFN]



RECOMMENDED LAND PATTERN

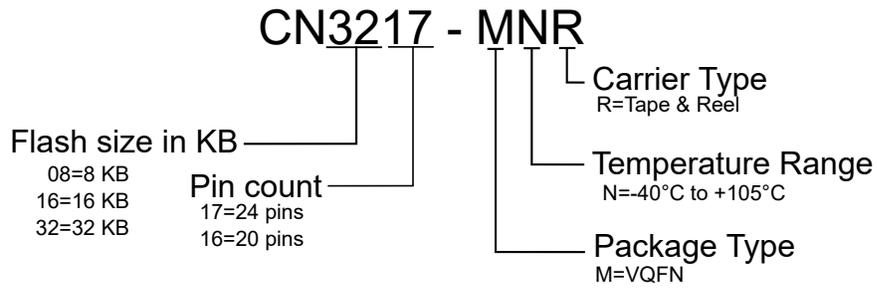
Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Optional Center Pad Width	X2			2.75
Optional Center Pad Length	Y2			2.75
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X24)	X1			0.30
Contact Pad Length (X24)	Y1			0.85
Contact Pad to Center Pad (X24)	G1	0.20		
Contact Pad to Contact Pad (X20)	G2	0.20		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
- For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process.

产品标识体系

欲订货或获取信息，请访问 www.weixinsemi.com。



制造商信息

商标

本文档中的名称、徽标和品牌均为制造商或其关联公司和/或子公司在中国和/或其他国家或地区的注册商标或商标。

法律声明

本出版物仅适用于制造商的产品，包括设计、测试以及将制造商的产品集成到用户的应用中。以其他任何方式使用这些信息都将被视为违反条款。

不涉及任何制造商知识产权的使用许可。

如果将制造商的器件用于生命维持和/或生命安全应用，一切风险由买方自负。

器件应用的详细信息仅供参考，内容可能随时更新。用户须自行确保应用符合规范。如需支持，请通过 www.weixinsemi.com 联系制造商。

用户须遵守所有适用的出口管制与经济制裁规定。

本文档中的信息“按原样”提供。制造商对这些信息不作任何形式的担保，包括但不限于针对非侵权性、适销性和特定用途的适用性的担保。除法律强制要求外，对于因这些信息或使用这些信息而产生的任何损失，制造商概不承担任何责任。在法律允许的最大范围内，制造商概不承担任何间接或附带损害赔偿。制造商在任何情况下所承担的全部责任均不超出用户为获得这些信息而向制造商支付的金额（如有）。

制造商的器件代码保护功能

请注意以下有关制造商产品的代码保护功能的要点：

- 制造商的产品均达到制造商数据手册中所述的技术规范。
- 制造商确信：在正常使用且符合工作规范的情况下，其产品非常安全。
- 制造商注重并积极保护其知识产权。严禁任何试图破坏制造商的代码保护功能的行为。
- 制造商或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着产品是“牢不可破”的。代码保护功能处于持续发展中。制造商承诺将不断改进产品的代码保护功能。

中国销售及服务

如需获取更多信息或支持，请通过以下方式联系我们：

邮箱：sales@weixinsemi.com

网址：www.weixinsemi.com