GigaDevice Semiconductor Inc.

GD32L23x
Arm® Cortex®-M23 32-bit MCU

适用于 GD32L233xx 系列

用户手册

1.4 版本

（2023 年 6 月）

目录

图索引

表索引

1. 系统及存储器架构

GD32L23x系列器件是基于Arm® Cortex®-M23处理器的32位通用微控制器。Arm® Cortex®-M23处理器的所有存储访问，根据不同的目的和目标存储空间，都会在AHB总线上执行。存储器的组织采用了ARMv8M结构，预先定义的存储器映射和高达4 GB的存储空间，充分保证了系统的灵活性和可扩展性。

1.1. Arm® Cortex®-M23 处理器

Arm® Cortex®-M23处理器是一个低功耗32位处理器。适用于需要一个区域优化处理器来进行深度嵌入式应用的场景。Arm® Cortex®-M23处理器为开发人员提供了显著的好处，包括：

- 一个简单的体系结构，易于学习和编程；
- 超低功耗、高效节能；
- 优秀的代码密度；
- 确定性、高性能中断处理；
- 向上兼容Cortex-M处理器家族系列。

Arm® Cortex®-M23处理器通过精简强大的指令集和广泛优化的设计提供高效处理能力，提供包括单周期乘法器和17周期分频器的高端处理硬件。

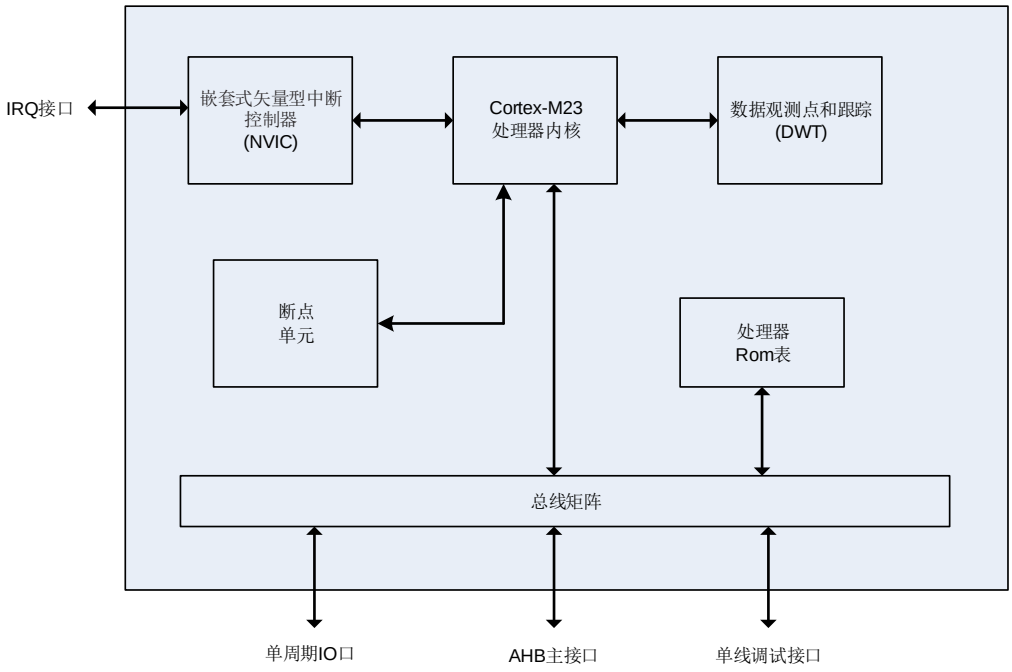
Arm® Cortex®-M23处理器高度集成了一个可配置的嵌套矢量中断控制器（NVIC），以提供业界领先的中断性能。

下面列出由Arm® Cortex®-M23提供的一些系统外设：

- 低延迟，高速外设I/O端口；
- 向量表偏移寄存器；
- 断点单元；
- 数据观测点；
- 串行调试接口。

*图1-1. Arm® Cortex®-M23处理器结构框图*显示了Arm® Cortex®-M23处理器结构框图。欲了解更多信息，请参阅Arm® Cortex®-M23技术参考手册。

图 1-1. Arm® Cortex®-M23 处理器结构框图



1.2. 系统架构

GD32L23x 设备实现了总线矩阵，它被用于管理主机之间的访问仲裁，仲裁采用的是 Round Robin 算法。总线矩阵提供了从主机到从机的访问，即使多个高速外围设备同时工作，也可以进行并发访问和高效运行。GD32L23x 设备采用 32 位多层总线结构，该结构可使系统中的多个主机和从机之间的并行通信成为可能。多层总线结构包括一个 AHB 互联矩阵、两个 AHB 总线。AHB 互联矩阵的互联关系接下来将进行说明。在*表 1-1. AHB 互联矩阵的互联关系列表*，”1”表示相应的主机可以通过 AHB 互联矩阵访问对应的从机，空白的单元格表示相应的主机不可以通过 AHB 互联矩阵访问对应的从机。

表 1-1. AHB 互联矩阵的互联关系列表

	SBUS	DMA
FMC	1	1
SRAM0	1	1
AHB1	1	1
AHB2	1	1
SRAM1	1	1

如上表所示，AHB 互联矩阵共连接两个主机，分别为：SBUS 和 DMA。CPU SBUS 将 Cortex®-M23 内核的系统总线（外设总线）连接到管理内核与 DMA 之间的仲裁的总线矩阵。DMA 总线将 DMA 的 AHB 主接口连接到总线矩阵，该矩阵管理 CPU 和 DMA 对 SRAM，FLASH 和 AHB / APB 外设的访问。

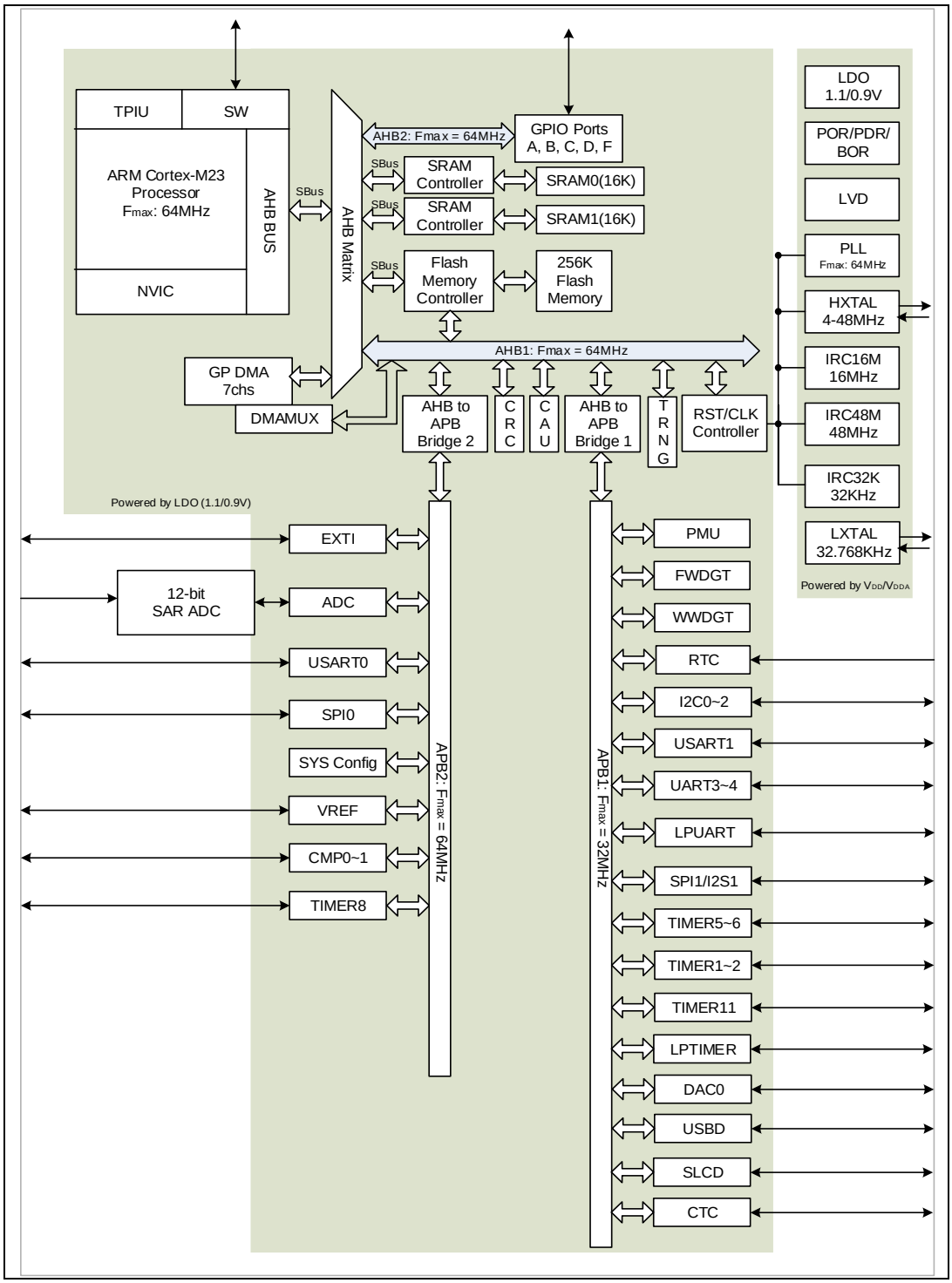
AHB 互联矩阵也连接了一些从机，分别为：FMC、SRAM0、SRAM1、AHB1 和 AHB2。FMC 是闪存存储器控制器的总线。SRAM0~ SRAM1 是片上静态随机存取存储器。AHB1 是连接所

有 AHB 从机和 AHB 到 APB 桥的 AHB 总线，AHB2 是连接 AHB2 从机的 AHB 总线。AHB 到 APB 的桥接器是与所有 APB 从机连接的两条 APB 总线。两条 APB 总线与所有 APB 外设相连。APB1 速度限制为 32MHz，APB2 速度限制为 64MHz。

GD32L23x 系列器件的系统架构如下图所示。该 AHB 矩阵是一个基于 AMBA 5.0 AHB-LITE 的多层总线，这个结构使得系统中的多个主机和从机之间的并行通信成为可能。该 AHB 矩阵中包含属于 Arm® Cortex®-M23 内核的 AHB 总线，以及内核外的 DMA 共 2 个主机。该 AHB 矩阵还连接了 4 个从机，分别为：FMC、内部 SRAM0、内部 SRAM1、AHB1 和 AHB2。

AHB2 连接 GPIO 端口。AHB1 连接 AHB 外设，包括 2 个 AHB-APB 总线桥。AHB-APB 总线桥提供了 AHB1 和两条 APB 总线之间的全同步连接。两条 APB 总线连接了所有的 APB 外设。

图 1-2. GD32L23x 系列器件的系统架构示意图



1.3. 存储器映射

程序存储器，数据存储器，寄存器和 I/O 端口都在同一个线性的 4 GB 的地址空间之内。这是 Arm® Cortex®-M23 的最大地址范围，因为它的地址总线宽度是 32 位。此外，为了降低不同客户在相同应用时的软件复杂度，存储映射是按 Arm® Cortex®-M23 处理器提供的规则预先定义的。同时，一部分地址空间由 Arm® Cortex®-M23 的系统外设所占用。下表显示了 GD32L23x

系列器件的存储器映射，包括代码、SRAM、外设和其他预先定义的区域。几乎每个外设都分配了 1KB 的地址空间，这样可以简化每个外设的地址译码。

表 1-2. GD32L23x 系列器件的存储器映射表

预先定义的地址空间	总线	地址范围	外设
		0xE000 0000 - 0xE00F FFFF	Cortex®-M23 内部外设
外部设备		0xA000 0000 - 0xDFFF FFFF	保留
外部 RAM		0x60000000 - 0x9FFFFFFF	保留
外设	AHB1	0x5006 1000 - 0x5FFF FFFF	保留
		0x5006 0C00 - 0x5006 0FFF	保留
		0x5006 0800 - 0x5006 0BFF	TRNG
		0x5006 0400 - 0x5006 07FF	保留
		0x5006 0000 - 0x5006 03FF	CAU
		0x5005 0400 - 0x5005 FFFF	保留
		0x5005 0000 - 0x5005 03FF	保留
		0x5004 0000 - 0x5004 FFFF	保留
		0x5000 0000 - 0x5003 FFFF	保留
	AHB2	0x4800 1800 - 0x4FFF FFFF	保留
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	保留
		0x4800 0C00 - 0x4800 0FFF	GPIOD
		0x4800 0800 - 0x4800 0BFF	GPIOC
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4002 4400 - 0x47FF FFFF	保留
		0x4002 4000 - 0x4002 43FF	保留
		0x4002 3400 - 0x4002 3FFF	保留
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2400 - 0x4002 2FFF	保留
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1400 - 0x4002 1FFF	保留
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0C00 - 0x4002 0FFF	保留
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	保留
		0x4002 0000 - 0x4002 03FF	DMA
	APB2	0x4001 8000 - 0x4001 FFFF	保留
		0x4001 7C00 - 0x4001 7FFF	CMP
		0x4001 5C00 - 0x4001 7BFF	保留
		0x4001 5800 - 0x4001 5BFF	DBG
		0x4001 5000 - 0x4001 57FF	保留
		0x4001 4C00 - 0x4001 4FFF	TIMER8

预先定义的地址空间	总线	地址范围	外设
		0x4001 3C00 - 0x4001 4BFF	保留
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	保留
		0x4001 3000 - 0x4001 33FF	SPI0
		0x4001 2C00 - 0x4001 2FFF	保留
		0x4001 2800 - 0x4001 2BFF	保留
		0x4001 2400 - 0x4001 27FF	ADC
		0x4001 0800 - 0x4001 23FF	保留
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG + VREF
	APB1	0x4000 CC00 - 0x4000 FFFF	保留
		0x4000 C800 - 0x4000 CBFF	CTC
		0x4000 C400 - 0x4000 C7FF	保留
		0x4000 C000 - 0x4000 C3FF	I2C2
		0x4000 9800 - 0x4000 BFFF	保留
		0x4000 9400 - 0x4000 97FF	LPTIMER
		0x4000 8400 - 0x4000 93FF	保留
		0x4000 8000 - 0x4000 83FF	LPUART
		0x4000 7C00 - 0x4000 7FFF	保留
		0x4000 7800 - 0x4000 7BFF	保留
		0x4000 7400 - 0x4000 77FF	DAC0
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6400 - 0x4000 6FFF	保留
		0x4000 6000 - 0x4000 63FF	USBD RAM (512 bytes)
		0x4000 5C00 - 0x4000 5FFF	USBD
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 5000 - 0x4000 53FF	UART4
		0x4000 4C00 - 0x4000 4FFF	UART3
		0x4000 4800 - 0x4000 4BFF	保留
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	保留
		0x4000 3C00 - 0x4000 3FFF	保留
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 - 0x4000 37FF	保留
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	SLCD
		0x4000 2000 - 0x4000 23FF	保留
		0x4000 1C00 - 0x4000 1FFF	保留

预先定义的 地址空间	总线	地址范围	外设
		0x4000 1800 - 0x4000 1BFF	TIMER11
		0x4000 1400 - 0x4000 17FF	TIMER6
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0800 - 0x4000 0FFF	保留
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
		0x4000 0000 - 0x4000 03FF	保留
SRAM		0x2000 8000 - 0x3FFF FFFF	保留
		0x2000 5000 - 0x2000 7FFF	SRAM1(16KB)
		0x2000 4000 - 0x2000 4FFF	
		0x2000 2000 - 0x2000 3FFF	SRAM0(16KB)
		0x2000 1000 - 0x2000 1FFF	
		0x2000 0000 - 0x2000 0FFF	
代码		0x1FFF F810 - 0x1FFF FFFF	保留
		0x1FFF F800 - 0x1FFF F80F	Option bytes(16B)
		0x1FFF D000- 0x1FFF F7FF	System memory(10KB)
		0x1FFF 7200 - 0x1FFF CFFF	保留
		0x1FFF 7000 - 0x1FFF 71FF	OTP(512B)
		0x1000 0000 - 0x1FFF 6FFF	保留
		0x0804 0000 - 0x0FFF FFFF	保留
		0x0802 0000 - 0x0803 FFFF	Main Flash memory(256KB)
		0x0801 0000 - 0x0801 FFFF	
		0x0800 0000 - 0x0800 FFFF	
		0x0001 0000 - 0x07FF FFFF	保留
		0x0000 0000 - 0x0000 FFFF	Aliased to Flash or system memory

1.3.1. 片上 SRAM 存储器

GD32L23x系列微控制器含有高达32KB的片上SRAM，起始地址为0x2000 0000，支持字节、半字(16比特)和整字(32比特)访问。

1.3.2. 片上闪存

该系列微控制器提供高达256KB的片上闪存。有关闪存模块的组织结构，请参阅*2.3.1*。

读访问支持字节、半字(16比特)和整字(32比特)；写访问（编程）只支持半字、字(16比特)和整字(32比特)。片上闪存的每一页都可以单独被擦除，整个主存模块（除信息块）也可以同时被擦除。

1.4. 引导配置

GD32L23x系列微控制器提供了三种引导源，可以通过BOOT0和BOOT1引脚来进行选择，详细说明见*表1-3. 引导模式*。该两个引脚的电平状态会在复位后的第四个CK_SYS(系统时钟)的上升沿进行锁存。用户可自行选择所需要的引导源，通过设置上电复位和系统复位后的BOOT0和BOOT1的引脚电平。一旦这两个引脚电平被采样，它们可以被释放并用于其他用途。

表 1-3. 引导模式

引导源选择	启动模式选择引脚	
	Boot1	Boot0
主FLASH存储器	x	0
系统存储器	0	1
片上SRAM	1	1

上电序列或系统复位后，Arm® Cortex®-M23处理器先从0x0000 0000地址获取栈顶值，再从0x0000 0004地址获得引导代码的基地址，然后从引导代码的基地址开始执行程序。

根据所选的引导源，片上闪存的主存（开始于0x0800 0000的存储空间）或系统存储器（开始于0x1FFF D000的存储空间）会被映射到引导空间，即从0x0000 0000开始的地址空间。如果片上SRAM（开始于0x2000 0000的存储空间）被选为引导源，用户必须在应用程序初始化代码中通过修改NVIC异常向量表和偏移寄存器将向量表重置到SRAM中。

芯片内嵌的引导装载程序位于系统存储器中，用来对片上闪存的主存进行重编程。该引导装载程序可通过以下接口之一工作：USART0, USART1或USBD。

1.5. 系统配置控制器

系统配置控制器（SYSCFG）的主要用途如下：

- 在某些I / O口上启用/禁用I2C Fast Mode Plus
- 重新映射某些I / O口
- 管理与GPIO的外部中断线连接

1.6. 系统配置寄存器（SYSCFG）

SYSCFG基地址：0x4001 0000

1.6.1. 配置寄存器 0（SYSCFG_CFG0）

地址偏移：0x00
复位值：0x0000 000X (X表示BOOT_MODE[1:0]可能为任意值，取决于复位后BOOT0引脚和BOOT1引脚的电平)

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												PB9_HCCE	PB8_HCCE	PB7_HCCE	PB6_HCCE
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									BOOT0_PD3_RMP	保留	PA11_PA12_RMP	保留		BOOT_MODE[1:0]	
									rw		rw			r	

位/位域	名称	描述
31:20	保留	必须保持复位值。
19	PB9_HCCE	PB9引脚大电流能力使能 当该位为1时, PB9引脚可以直接用来控制红外发光二极管。 0：PB9引脚大电流能力关闭 1：PB9引脚大电流能力开启，同时该引脚的速度控制被忽略
18	PB8_HCCE	PB8引脚大电流能力使能 当该位为1时, PB8引脚可以直接用来控制红外发光二极管。 0：PB8引脚大电流能力关闭 1：PB8引脚大电流能力开启，同时该引脚的速度控制被忽略
17	PB7_HCCE	PB7引脚大电流能力使能 当该位为1时, PB7引脚可以直接用来控制红外发光二极管。 0：PB7引脚大电流能力关闭 1：PB7引脚大电流能力开启，同时该引脚的速度控制被忽略
16	PB6_HCCE	PB6引脚大电流能力使能 当该位为1时, PB6引脚可以直接用来控制红外发光二极管。 0：PB6引脚大电流能力关闭 1：PB6引脚大电流能力开启，同时该引脚的速度控制被忽略
15:7	保留	必须保持复位值。
6	BOOT0_PD3_RMP	BOOT0和PD3重映射使能 它控制BOOT0引脚映射到BOOT0或PD3。

		BOOT0_PD3_RMP置位时，复位后，硬件会将BOOT0功能设定为0。在这种情况下，系统将从主闪存启动，而不考虑BOOT0引脚的输入值。 0：无重映射（BOOT0功能映射在BOOT0引脚上） 1：重新映射（BOOT0功能映射在PD3引脚上）
5	保留	必须保持复位值。
4	PA11_PA12_RMP	小封装PA11/PA12重映射位控制位（28和20pin封装） 该位由软件设置和清除。它控制PA9/10或PA11/12针对在小引脚数封装上的映射。 0：不重映射（PA9/10映射在引脚上） 1：重映射（PA11/12映射而不是PA9/10）
3:2	保留	必须保持复位值。
1:0	BOOT_MODE[1:0]	引导模式(详细请参考第1.4章节 *引导配置*) 位0映射到BOOT0引脚；位1的值映射到BOOT1引脚 x0：从片上闪存的主存引导启动 01：从片上闪存的系统存储器引导启动 11：从片上SRAM引导启动

1.6.2. EXTI 源选择寄存器 0（SYSCFG_EXTISS0）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3_SS [3:0]				EXTI2_SS [3:0]				EXTI1_SS [3:0]				EXTI0_SS [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI3_SS[3:0]	EXTI 3源选择 X000：PA3引脚 X001：PB3引脚 X010：PC3引脚 X011：PD3引脚 X100：保留 X101：保留 X110：保留 X111：保留
11:8	EXTI2_SS[3:0]	EXTI 2源选择

X000：PA2引脚
X001：PB2引脚
X010：PC2引脚
X011：PD2引脚
X100：保留
X101：保留
X110：保留
X111：保留

7:4	EXTI1_SS[3:0]	EXTI 1源选择 X000：PA1引脚 X001：PB1引脚 X010：PC1引脚 X011：PD1引脚 X100：保留 X101：PF1引脚 X110：保留 X111：保留
3:0	EXTI0_SS[3:0]	EXTI 0 源选择 X000：PA0引脚 X001：PB0引脚 X010：PC0引脚 X011：PD0引脚 X100：保留 X101：PF0引脚 X110：保留 X111：保留

1.6.3. EXTI 源选择寄存器 1（SYSCFG_EXTISS1）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7_SS [3:0]				EXTI6_SS [3:0]				EXTI5_SS [3:0]				EXTI4_SS [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。

15:12	EXTI7_SS[3:0]	EXTI 7源选择 X000：PA7引脚 X001：PB7引脚 X010：PC7引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
11:8	EXTI6_SS[3:0]	EXTI 6源选择 X000：PA6引脚 X001：PB6引脚 X010：PC6引脚 X011：PD6引脚 X100：保留 X101：保留 X110：保留 X111：保留
7:4	EXTI5_SS[3:0]	EXTI 5源选择 X000：PA5引脚 X001：PB5引脚 X010：PC5引脚 X011：PD5引脚 X100：保留 X101：保留 X110：保留 X111：保留
3:0	EXTI4_SS[3:0]	EXTI 4源选择 X000：PA4引脚 X001：PB4引脚 X010：PC4引脚 X011：PD4引脚 X100：保留 X101：保留 X110：保留 X111：保留

1.6.4. EXTI 源选择寄存器 2（SYSCFG_EXTISS2）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11_SS [3:0]				EXTI10_SS [3:0]				EXTI9_SS [3:0]				EXTI8_SS [3:0]			
rw				rw				rw				rw			

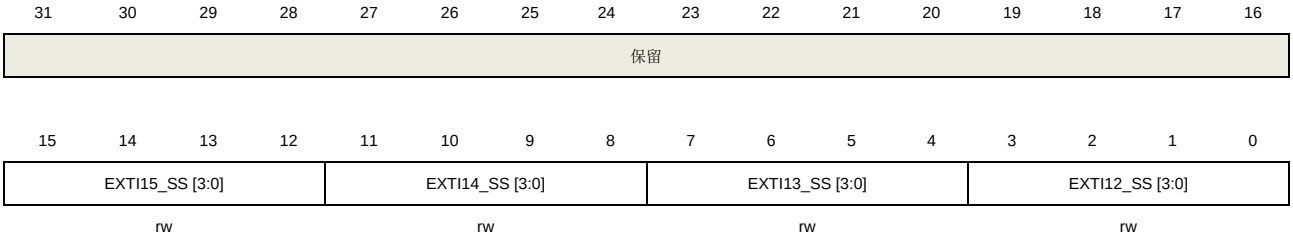
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI11_SS[3:0]	EXTI 11源选择 X000：PA11引脚 X001：PB11引脚 X010：PC11引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
11:8	EXTI10_SS[3:0]	EXTI 10源选择 X000：PA10引脚 X001：PB10引脚 X010：PC10引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
7:4	EXTI9_SS[3:0]	EXTI 9源选择 X000：PA9引脚 X001：PB9引脚 X010：PC9引脚 X011：PD9引脚 X100：保留 X101：保留 X110：保留 X111：保留
3:0	EXTI8_SS[3:0]	EXTI 8源选择 X000：PA8引脚 X001：PB8引脚 X010：PC8引脚 X011：PD8引脚 X100：保留

X101：保留
X110：保留
X111：保留

1.6.5. EXTI 源选择寄存器 3（SYSCFG_EXTISS3）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32位）访问



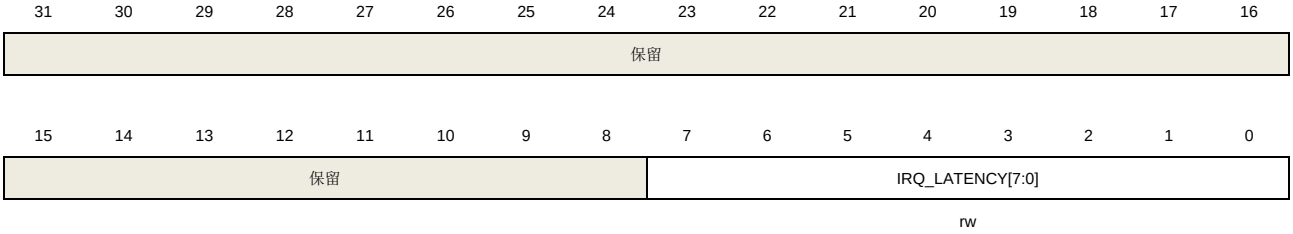
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI15_SS[3:0]	EXTI 15源选择 X000：PA15引脚 X001：PB15引脚 X010：PC15引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
11:8	EXTI14_SS[3:0]	EXTI 14源选择 X000：PA14引脚 X001：PB14引脚 X010：PC14引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留
7:4	EXTI13_SS[3:0]	EXTI 13源选择 X000：PA13引脚 X001：PB13引脚 X010：PC13引脚 X011：保留

		X100：保留 X101：保留 X110：保留 X111：保留
3:0	EXTI12_SS[3:0]	EXTI 12源选择 X000：PA12引脚 X001：PB12引脚 X010：PC12引脚 X011：保留 X100：保留 X101：保留 X110：保留 X111：保留

1.6.6. IRQ 延迟寄存器（SYSCFG_CPU_IRQ_LAT）

地址偏移: 0x100
复位值: 0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	IRQ_LATENCY[7:0]	IRQ_LATENCY 指定在 NVIC 中挂起的中断与在 AHB-Lite 接口上发出的该中断的向量读取之间的最小周期数。 如果将 IRQ_LATENCY 延迟设置为 0，则尽可能快地响应中断。 对于非零值，Cortex-M23 处理器确保在 NVIC 中的中断和正在执行的中断的向量获取之间存在最小的 IRQ_LATENCY + 1 个 hclk 周期。

1.7. 设备电子签名

设备的电子签名中包含的存储容量信息和96位的唯一设备ID。它被存储在片上闪存的信息模块中。96位唯一设备ID对于每颗芯片而言都是唯一的。它可以用作序列号，或安全密钥的一部分，等等。

1.7.1. 存储容量信息

基地址：0x1FFF F7E0
该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM_DENSITY[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLASH_DENSITY[15:0]															
r															

位/位域	名称	描述
31:16	SRAM_DENSITY [15:0]	SRAM存储器容量 该值表明芯片的片上SRAM存储器容量，以Kbytes为单位 例如：0x0008表示8Kbytes。
15:0	FLASH_DENSITY [15:0]	Flash存储器容量 该值表明芯片的片上Flash容量，以Kbytes为单位 例如：0x0020表示32Kbytes。

1.7.2. 设备唯一 ID（96 位/位域）

基地址：0x1FFF F7E8
该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[15:0]															
r															

位/位域	名称	描述
31:0	UNIQUE_ID[31:0]	设备唯一ID

基地址：0x1FFF F7EC
该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[63:48]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

UNIQUE_ID[47:32]

r

位/位域	名称	描述
31:0	UNIQUE_ID[63:32]	设备唯一ID

基地址：0x1FFF F7F0
该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[95:80]															

r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[79:64]															

r

位/位域	名称	描述
31:0	UNIQUE_ID[95:64]	唯一设备ID

2. 闪存控制器（FMC）

2.1. 简介

闪存控制器（FMC），提供了片上闪存需要的所有功能。在闪存的前256K字节空间内，CPU执行指令需要少量等待时间。FMC也提供了页擦除，整片擦除，以及编程操作。

2.2. 主要特征

- 高达256KB的片上闪存可用于存储指令或数据；
- 在闪存的前256K字节空间内，CPU执行指令需要0~3个等待时间；
- 预取缓冲区以加速读操作；
- 根据不同闪存容量大小，闪存页大小为4/2/1KB；
- 支持32位整字编程，页擦除和整片擦除操作；
- 512B OTP块（一次性编程），用于存储用户数据；
- 大小为16字节的选项字节可根据用户需求配置；
- 当系统复位时，选项字节被上载到选项字节控制寄存器；
- 具有安全保护状态，可阻止对代码或数据的非法读访问；
- 具有擦除和编程保护状态，可阻止意外写操作；
- 支持快速编程；
- 支持低功耗模式。

2.3. 功能说明

2.3.1. 闪存结构

存储器包括一个高达256KB的主闪存和一个10KB的用于引导装载程序的信息块。主存储闪存分为64或32页，每页大小为4K或2K或1K。主存储闪存的每页都可以单独擦除。闪存结构细节见下表。

表 2-1. 256KB 闪存基地址和构成

闪存块	名称	地址范围	大小（字节）
主存储闪存块	第0页	0x0800 0000 - 0x0800 0FFF	4KB
	第1页	0x0800 1000 - 0x0800 1FFF	4KB
	第2页	0x0800 2000 - 0x0800 2FFF	4KB
	.		
	第63页	0x0803 E000 - 0x0803 FFFF	4KB

闪存块	名称	地址范围	大小（字节）
信息块	引导装载程序	0x1FFF D000- 0x1FFF F7FF	10KB
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F80F	16B
一次性编程块	OTP字节	0x1FFF_7000~0x1FFF_71FF	512B

注意: 信息块存储了引导装载程序（boot loader），不能被用户编程或擦除。

表 2-2. 128KB 闪存基地址和构成

闪存块	名称	地址范围	大小（字节）
主存储闪存块	第0页	0x0800 0000 - 0x0800 07FF	2KB
	第1页	0x0800 0800 - 0x0800 0FFF	2KB
	第2页	0x0800 1000 - 0x0800 17FF	2KB
	.		
	第63页	0x0801 F800 - 0x0801 FFFF	2KB
信息块	引导装载程序	0x1FFF D000- 0x1FFF F7FF	10KB
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F80F	16B
一次性编程块	OTP字节	0x1FFF_7000~0x1FFF_71FF	512B

注意: 信息块存储了引导装载程序（boot loader），不能被用户编程或擦除。

表 2-3. 64KB 闪存基地址和构成

闪存块	名称	地址范围	大小（字节）
主存储闪存块	第0页	0x0800 0000 - 0x0800 03FF	1KB
	第1页	0x0800 0400 - 0x0800 07FF	1KB
	第2页	0x0800 0800 - 0x0800 0BFF	1KB
	.		
	第63页	0x0800 FC00 - 0x0800 FFFF	1KB
信息块	引导装载程序	0x1FFF D000- 0x1FFF F7FF	10KB
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F80F	16B
一次性编程块	OTP字节	0x1FFF_7000~0x1FFF_71FF	512B

注意: 信息块存储了引导装载程序（boot loader），不能被用户编程或擦除。

表 2-4. 32KB 闪存基地址和构成

闪存块	名称	地址范围	大小（字节）
主存储闪存块	第0页	0x0800 0000 - 0x0800 03FF	1KB
	第1页	0x0800 0400 - 0x0800 07FF	1KB
	第2页	0x0800 0800 - 0x0800 0BFF	1KB
	.		
	第31页	0x08007C00 - 0x0800 7FFF	1KB
信息块	引导装载程序	0x1FFF D000- 0x1FFF F7FF	10KB
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F80F	16B
一次性编程块	OTP字节	0x1FFF_7000~0x1FFF_71FF	512B

注意: 信息块存储了引导装载程序（boot loader），不能被用户编程或擦除。

2.3.2. 读操作

闪存可以像普通存储空间一样直接寻址访问。对闪存取指令和取数据分别使用CPU的SBUS总线。

增加等待状态：

根据AHB时钟频率，读闪存时需正确配置FMC_WS寄存器中的WSCNT位。WSCNT位和AHB时钟频率的对应关系见*表2-5. WSCNT与AHB时钟频率对应关系*。

表 2-5. WSCNT 与 AHB 时钟频率对应关系（LDO= 1.1V）

AHB时钟频率	WSCNT配置
<= 36MHz	0（0等待状态增加）
<= 64MHz	1（1等待状态增加）

表 2-6. WSCNT 与 AHB 时钟频率对应关系（LDO= 0.9V）

AHB时钟频率	WSCNT配置
<= 16MHz	0（0等待状态增加）
<= 32MHz	1（1等待状态增加）
<= 48MHz	2（2等待状态增加）
<= 64MHz	3（3等待状态增加）

如果发生系统复位，AHB时钟频率为16MHz，此时WSCNT置为0。

注意：

1.如果希望增加AHB时钟频率。首先，参考WSCNT位和AHB时钟频率的对应关系表，根据目标AHB时钟频率配置WSCNT位。然后，增加AHB时钟频率至目标频率。禁止在配置WSCNT位之前增加AHB时钟频率。

2.如果希望降低AHB时钟频率。首先，降低AHB时钟频率至目标频率。然后，参考WSCNT位和AHB时钟频率的对应关系表，根据目标AHB时钟频率配置WSCNT位。禁止在降低AHB时钟频率之前配置WSCNT位。

由于添加了等待状态，读效率非常低（例如：LDO = 1.1V，64MHz时需添加1个等待状态）。为了加速读操作，需要用到一些功能。

当前缓存区：

当前缓存区总是被使能的。每次从闪存中读取数据时，当前缓存区可以缓存64位数据。因为CPU每次读操作只需要32位或16位数据。因此在顺序代码下，CPU所需数据可以从当前缓存区获取而不必重复从闪存中获取。

预取缓存区：

置位FMC_WS寄存器中PFEN位来使能预取缓存区。在顺序代码下，当CPU执行来自当前缓存区的数据时（64位），按32位执行时需要至少2个时钟周期，按16位执行时需要至少4个时钟周期。在这种情况下，从flash闪存中预取下一个双字地址的数据并存储在预取缓存区。当CPU执行完当前缓存区的数据时，预取缓存区提供下次需要执行的数据。

2.3.3. FMC_CTL 寄存器解锁

复位后，FMC_CTL寄存器进入写操作锁定状态，LK位复位后置为1。通过先后向FMC_KEY寄存器写入0x45670123和0xCDEF89AB，可以使得FMC_CTL寄存器解锁。两次写操作后，FMC_CTL寄存器的LK位被硬件清0。可以通过软件设置FMC_CTL寄存器的LK位为1再次锁定FMC_CTL寄存器。任何对FMC_KEY寄存器的错误操作都会将LK位置1，从而锁定FMC_CTL寄存器，并引发一个总线错误。

FMC_CTL寄存器的OBPG位和OBER位在FMC_CTL寄存器解锁后，仍然被保护。解锁序列是向FMC_OBKEY寄存器先后写入0x45670123和0xCDEF89AB，然后硬件会将FMC_CTL寄存器的OBWEN位置1。软件可以将FMC_CTL的OBWEN位清0来锁定FMC_CTL的OBPG位和OBER位。

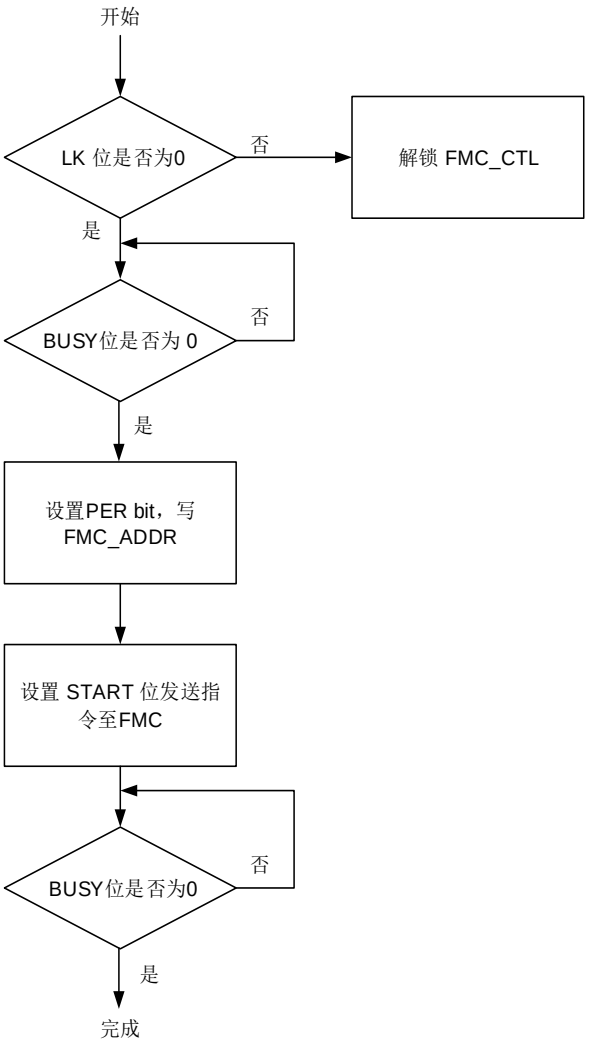
2.3.4. 页擦除

FMC的页擦除功能使得主存储闪存的页内容初始化为高电平。每一页都可以被独立擦除，而不影响其他页内容。页擦除页操作，寄存器设置具体步骤如下：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 置位FMC_CTL寄存器的PER位；
- 将待擦除页的绝对地址（0x08XX XXXX）写到FMC_ADDR寄存器；
- 通过将FMC_CTL寄存器的START位置1来发送页擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用SBUS读并验证该页是否擦除成功。

当页擦除成功执行，FMC_STAT寄存器的ENDF位将置位。若FMC_CTL寄存器的ENDIE位被置1，则FMC将触发一个中断。需要注意的是，用户需确保写入的是正确的擦除地址。否则当待擦除页的地址被用来取指令或访问数据时，软件将会“跑飞”。该情况下，FMC不会提供任何出错通知。另一方面，对擦写保护的页进行擦除操作将无效。如果FMC_CTL寄存器的ERRIE位被置位，该操作将触发操作出错中断。中断服务程序可通过检测FMC_STAT寄存器的WPERR位来判断该中断是否发生。下图显示了页擦除操作流程。

图 2-1. 页擦除操作流程



2.3.5. 整片擦除

FMC提供了整片擦除功能可以初始化主存储闪存块的内容。当设置FMC_CTL寄存器中MER为1时，擦除过程作用于整片闪存。整片擦除操作，寄存器设置具体步骤如下：

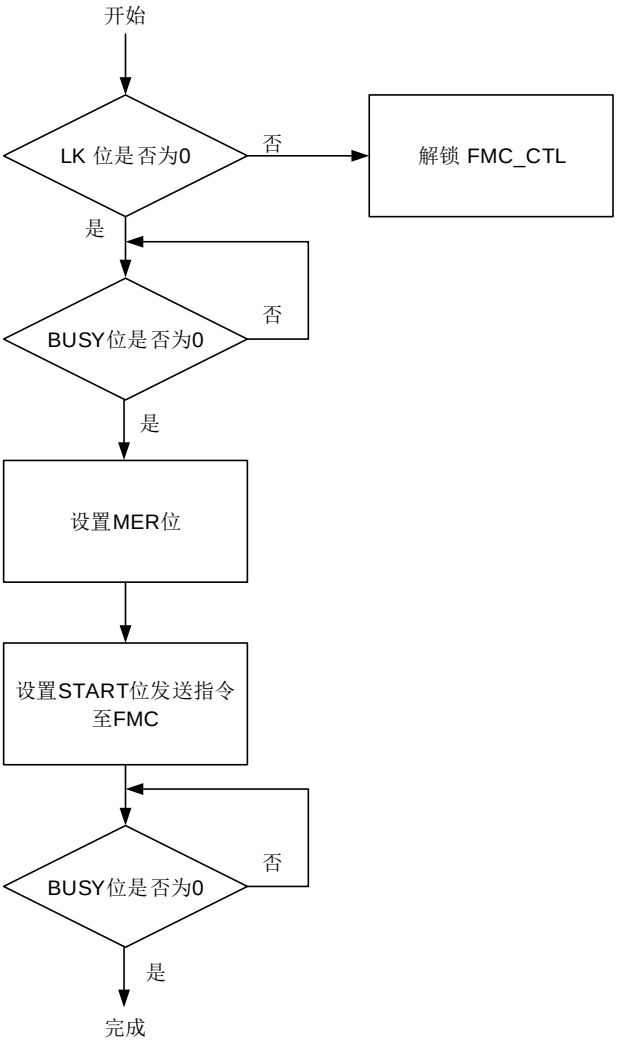
- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 如果整片擦除闪存，置位FMC_CTL寄存器的MER位；
- 通过将FMC_CTL寄存器的START位置1来发送整片擦除命令到FMC；

- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用SBUS读并验证是否擦除成功。

当整片擦除成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。由于所有的闪存数据都将被复位为0xFFFF_FFFF，可以通过运行在SRAM中的程序或使用调试工具直接访问FMC寄存器来实现整片擦除操作。此外，如果任何闪存页处于擦除/编程保护下，整片擦除操作会被忽略。在这种情况下，如果FMC_CTL寄存器的ERRIE位被置位，该操作将触发操作出错中断。在中断服务程序中，软件可以通过检查FMC_STAT寄存器中的WPERR位来检测这种情况。

下图显示了整片擦除操作流程。

图 2-2. 整片擦除操作流程



2.3.6. 主存储闪存块编程

FMC提供了一个通过SBUS修改主存储闪存内容的32位整字/16位半字编程功能。实际上，主存储闪存编程为32位。

编程操作，寄存器设置具体步骤如下：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 置位FMC_CTL寄存器的PG位；
- SBUS写一个32位整字/16位半字到目的绝对地址（0x08XX XXXX）；
  如果SBUS按32位字编程，SBUS写1次即可将数据编程入闪存。待编程数据必须字对齐。
  如果SBUS按16位字编程，SBUS写2次即可将数据编程入闪存。待编程数据必须字对齐。
  为了缩短编程时间，建议SBUS采用按32位编程。
- 等待编程指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用SBUS读并验证是否编程成功。

当主存储块编程成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。有一些编程错误需要注意：

编程操作时需要检查目的地址是否已经被擦除。如果该地址没有被擦除，FMC_STAT寄存器的PGERR位也将被置1。每个字在擦除后和下次擦除前只能编程一次。注意PG位必须在字/半字编程操作前被置位。

此外，在正被擦除/保护页上的编程操作会被忽略，FMC_STAT寄存器中的WPERR位会被置位。

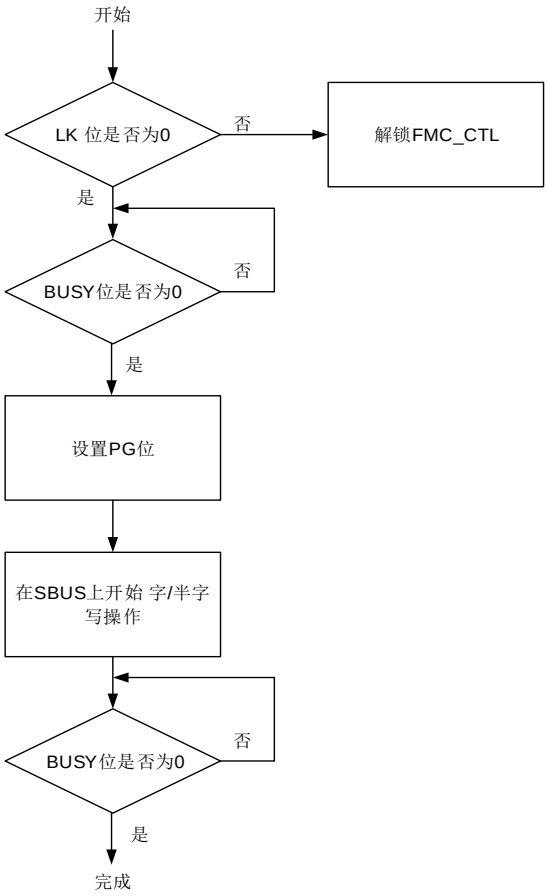
在下列情况下，FMC_STAT寄存器中的PGAERR位会被置位。

- SBUS按字节编程（非32位或16位）
- SBUS混合编程。不允许混合32位和16位编程。
- SBUS编程未对齐。

注意：如果编程数据未能写满32位，这些数据不会被编程入闪存，并且不会有任何提示。

在这些情况下，如果FMC_CTL寄存器的ERRIE位被置1，FMC将触发一次闪存操作错误中断。在中断服务程序中，软件可以通过检查FMC_STAT寄存器中的PGERR位，PGAERR位或者WPERR位来检测发生了哪种错误。下图显示了主存储块字编程操作流程。

图 2-3. 字编程操作流程



注意：避免在同一个bank中既进行读操作，又进行擦除/编程操作。当CPU进入省电模式时，对闪存的操作将失败。

2.3.7. 主存储闪存块快速编程

FMC提供了一个允许对一行（32个双字）快速编程的功能。通过消除编程前对闪存单元的验证，从而缩短了页编程时间。同时针对每个字避免了高电压的上升和下降时间。快速编程功能只能编程主存储器。

快速编程操作，寄存器设置具体步骤如下：

- 检查闪存的一行（32个双字），确保为全FF；
- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 置位FMC_CTL寄存器的FSTPG位；
- BUS写一行数据（32个双字）到目的绝对地址（0x08XX XXXX）；
- 等待编程指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用BUS读并验证是否编程成功。

当主存储块快速编程成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的

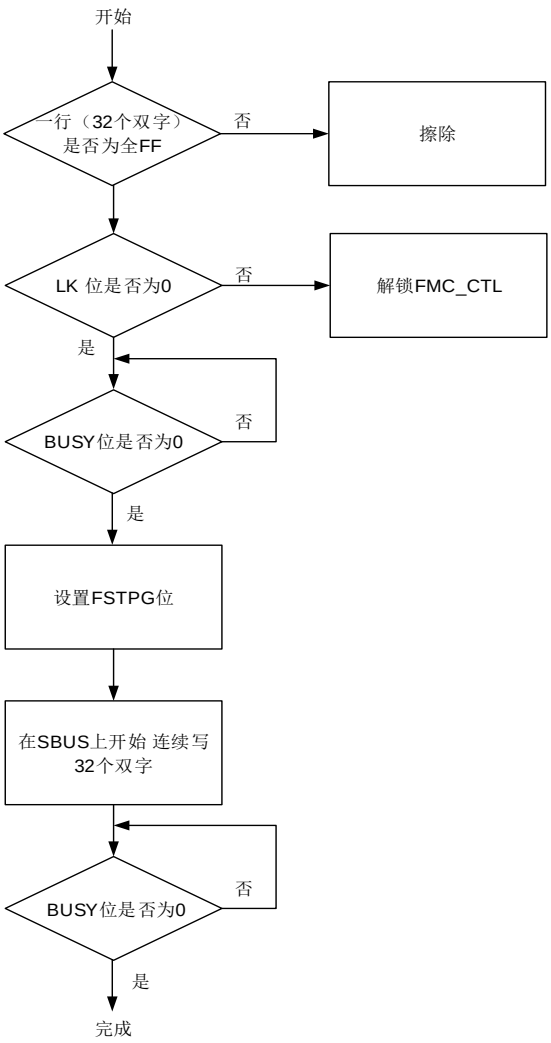
ENDIE位被置1，FMC将触发一个中断。有一些编程错误需要注意：

编程操作时需要检查目的地址是否已经被擦除。如果该地址没有被擦除，FMC_STAT寄存器的PGERR位也将被置1。每个字在擦除后和下次擦除前只能编程一次。注意PG位必须在字/半字编程操作前被置位。

此外，在正被擦除/保护页上的编程操作会被忽略，FMC_STAT寄存器中的WPERR位会被置位。

在这些情况下，如果FMC_CTL寄存器的ERRIE位被置1，FMC将触发一次闪存操作错误中断。在中断服务程序中，软件可以通过检查FMC_STAT寄存器中的PGERR位，PGAERR位或者WPERR位来检测发生了哪种错误。下图显示了主存储块字编程操作流程。

图 2-4. 快速编程操作流程



注意：

1. 32个双字必须对齐。
2. 如果在快速编程模式进行过程中尝试读取闪存操作，则快速编程将被中止，并且FMC_STAT寄存器中的END位置位。

3. 当闪存接口收到第一个字时，编程自动启动。当对第一个字施加高电压时，BUSY位置1；对最后一个字编程完成时，BUSY位被清除。

4. 必须连续写入32个双字。对于所有编程，闪存均保持高电压。两个字的写请求之间的最长时间为编程时间（大约8.8us）。如果第二个字在此编程时间后到达，则快速编程被中断，并且END位将置位，BUSY位将被清除。此时首先需要清除END位，然后写入下一个字以继续快速编程。

5. 两次擦除之间的整行高压持续时间不得超过3毫秒。这可由连续写入的32个双字序列保证。

6. 由于快速编程无法通过硬件检查闪存是否为FF，因此软件必须首先检查闪存是否为FF，并且两次擦除之间不得对一行进行两次或多次编程。如果在两次擦除之间对一行编程两次或更多次，则可能会发生不可预测的结果。

2.3.8. OTP 编程

OTP编程方法与主储存闪存编程相同。OTP块只能被编程一次并且不能被擦除。

注意：必须确保在OTP编程操作时不会发生任何意外中断，例如系统复位或掉电。如果发生意外中断，闪存中的数据有很小可能性会出错。

2.3.9. 选项字节擦除

FMC提供了一个擦除功能用来初始化闪存中的选项字节。选项字节擦除过程如下所示。

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 解锁FMC_CTL寄存器的选项字节操作位；
- 等待FMC_CTL寄存器的OBWEN位置1；
- 置位FMC_CTL寄存器的OBER位；
- 通过将FMC_CTL寄存器的START位置1来发送选项字节擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用SBUS读并验证是否擦除成功。

当选项字节擦除成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。

2.3.10. 选项字节编程

FMC提供了编程功能，可用来修改选项字节内容。选项字节共有8对选项字节。每对选项字节的高字节是低字节的补字节。当选项字节被修改时，FMC自动生成该选项字节的高字节。擦除操作过程如下。

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 解锁FMC_CTL寄存器的选项字节操作位；

- 等待FMC_CTL寄存器的OBWEN位置1；
- 置位FMC_CTL寄存器的OBPG位；
- SBUS写一个32位整字/16位半字到目的地址。写操作方法与主存储闪存编程操作类似；
- 等待编程指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用SBUS读并验证是否编程成功。

当选项字节编程成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。注意可能发生编程错误，类似主存储闪存编程操作，PGERR位和PGAERR位可能被置位。

选项字节修改后只有在系统复位后才生效。

2.3.11. 选项字节说明

每次系统复位后，选项字节被重加载到FMC_OBSTAT和FMC_WP寄存器后，选项字节生效。选项字节的补字节具体为选项字节取反。当选项字节被重装载时，如果选项字节的补字节和选项字节不匹配，FMC_OBSTAT寄存器的OBERR位将被置1，选项字节被强制设置为0xFF。若选项字节和其补字节同为0xFF，则OBERR位不置位。选项字节详情见下表。

表 2-7. 选项字节

地址	名称	说明
0x1fff f800	SPC	选项字节安全保护值 0xA5：未保护状态 除0xA5和0xCC外的任何值：保护级别低 0xCC：保护级别高
0x1fff f801	SPC_N	SPC补字节
0x1fff f802	USER	[7:5]：BOR_TH（BOR复位阈值） 000/101/110/111：BOR复位阈值0（1.6V） 001：BOR复位阈值1（2.0V） 010：BOR复位阈值2（2.2V） 011：BOR复位阈值3（2.5V） 100：BOR复位阈值4（2.8V） [4:3]：保留 [2]：nRST_STDBY 0：设置待机模式时产生复位而不是进入待机模式 1：设置待机模式时进入待机模式而不产生复位 [1]：nRST_DPSLP 0：设置深度睡眠模式时产生复位而不进入深度睡眠模式 1：设置深度睡眠模式时进入深度睡眠模式而不产生复位 [0]：nWDG_HW 0：硬件使能独立看门狗功能 1：软件使能独立看门狗功能
0x1fff f803	USER_N	USER补字节值
0x1fff f804	DATA[7:0]	用户定义数据7到0位
0x1fff f805	DATA_N[7:0]	DATA补字节值的7到0位

地址	名称	说明
0x1fff f806	DATA[15:8]	用户定义数据15到8位
0x1fff f807	DATA_N[15:8]	DATA补字节值的15到8位
0x1fff f808	WP[7:0]	页擦除/编程保护值的7到0位 0：保护生效 1：未保护
0x1fff f809	WP_N[7:0]	WP补字节值的7到0位
0x1fff f80a	WP[15:8]	页擦除/编程的保护值的15到8位
0x1fff f80b	WP_N[15:8]	WP补字节值的15到8位
0x1fff f80c	WP[23:16]	页擦除/编程的保护值的23到16位
0x1fff f80d	WP_N[23:16]	WP补字节值的23到16位
0x1fff f80e	WP[31:24]	页擦除/编程的保护值的31到24位 WP[30:24]：每个bit可设置4KB闪存的保护状态。第0位设置前4KB闪存的保护状态，以此类推。这31位总计可设置前124KB的闪存保护状态。 WP[31]：第31位可设置闪存剩下部分的保护状态。
0x1fff f80f	WP_N[31:24]	WP补字节值的31到24位

2.3.12. 页擦除/编程保护

FMC的页擦除/编程保护功能可以阻止对闪存的意外操作。当FMC对被保护页进行页擦除或编程操作时，操作本身无效且FMC_STAT寄存器的WPERR位将被置1。如果WPERR位被置1且FMC_CTL寄存器的ERRIE位也被置1来使能相应的中断，FMC将触发闪存操作出错中断，等待CPU处理。配置选项字节的WP [31:0]某位为0可以单独使能某几页的保护功能。如果在选项字节块执行了擦除操作，所有的闪存页擦除和编程保护功能都将失效。当选项字节的WP被改变时，需要系统复位使之生效。

2.3.13. 安全保护

FMC提供了一个安全保护功能来阻止非法读取闪存。此功能可以很好地保护软件和固件免受非法的用户操作。

未保护状态：当将SPC字节和它的补字节被设置为0x5AA5，系统复位以后，闪存将处于非安全保护状态。主存储块和选项字节可以被所有操作模式访问。

保护等级低：当设置SPC字节值为任何除0xA5或0xCC外的值，系统复位以后，低安全保护状态生效。需要注意的是，若该修改过程中，MCU的调试模块依然和外部JTAG/SWD设备相连，需要用上电复位代替系统复位以使得修改后的保护状态生效。在低安全保护状态下，主存储闪存块仅能被用户代码访问。在调试模式下，或从SRAM中启动时，以及从boot loader区启动时，这些模式下对主存储块的操作都被禁止。如果在这些模式下读主存储块，将产生总线错误。如果在这些模式下，对主存储块进行编程或擦除操作，FMC_STAT寄存器的WPERR位将被置1。但这些模式下都可以对选项字节进行操作，从而可以通过该方式失能安全保护功能。如果将SPC字节和它的补字节设置为0x5AA5，安全保护功能将失效，并自动触发一次整片擦除操作。

保护等级高：当设置SPC字节为0xCC，激活高安全保护等级。当编程选择该保护等级时，调

试模式，从SRAM中启动，或者从boot loader启动都被禁止。主存储闪存块可由用户代码的所有操作进行访问。SPC字节禁止再次编程。所以，如果高保护等级被激活，将不能再降回到低保护等级或无保护等级。

2.3.14. LVE 序列

如果要将内核电压从1.1V更改为0.9V。首先将设LVE置为1，并使用Read_LV时序（设置LDO为0.9V时的对应的等待状态）进行读取。然后再将内核电压从1.1V更改为0.9V（设置PMU章节LDOVS位为0）。

如果要将内核电压从0.9V更改为1.1V。首先将内核电压从0.9V实际更改为1.1V（设置PMU章节LDOVS位为1）。然后将LVE设置为0，并使用Read_HV时序（设置LDO为1.1V时的对应的等待状态）进行读取。

2.4. FMC 寄存器

FMC基地址：0x4002 2000

2.4.1. 等待状态寄存器（FMC_WS）

地址偏移：0x00
复位值：0x0000 0630

该寄存器只能按字（32位）访问。

位/位域	名称	描述
31:15	保留	必须保持复位值
14	SLEEP_SLP	该位确定当 MCU 进入待机模式或 RUN_SLP 位置位时，闪存进入睡眠模式还是掉电模式。 0：闪存进入掉电模式 1：闪存进入睡眠模式
13	RUN_SLP	该位确定当 MCU 正常或低功耗运行时，闪存进入睡眠/掉电模式（具体是睡眠还是掉电模式由 SLEEP_SLP 决定）还是空闲模式。 只有从 RAM 处执行代码时，闪存才可处于睡眠或掉电模式。 该位由 FMC_SLPKEY 寄存器写保护。 0：闪存进入空闲模式或从睡眠或掉电模式中唤醒（大约 5us） 1：闪存进入睡眠或掉电模式（当没有任何闪存上的操作时）
12:8	保留	必须保持复位值
7	LVE	低功耗模式使能 内核为 0.9V 或 1.1V 1：内核电压为 0.9V 0：内核电压为 1.1V
6:5	保留	必须保持复位值
4	PFEN	预取功能使能位 0：失能预取功能 1：使能预取功能
3	保留	必须保持复位值

2:0	WSCNT[2:0]	等待状态计数寄存器 软件置 1 和清 0。 000：不增加等待状态 001：增加 1 个等待状态 010：增加 2 个等待状态 011：增加 3 个等待状态 100 ~ 111：保留

2.4.2. 解锁寄存器（FMC_KEY）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31:0	KEY[31:0]	FMC_CTL 解锁寄存器 这些位仅能被软件写。 写解锁值到KEY[31:0]可以解锁 FMC_CTL寄存器。

2.4.3. 选项字节操作解锁寄存器（FMC_OBKEY）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31:0	OBKEY[31:0]	FMC_CTL 选项字节操作解锁寄存器 这些位仅能被软件写 写解锁值到OBKEY[31:0]解锁FMC_CTL寄存器的选项字节命令。

2.4.4. 状态寄存器（FMC_STAT）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSTAT	保留									ENDF	WPERR	PGAERR	PGERR	保留	BUSY
rc_w1										rc_w1	rc_w1	rc_w1	rc_w1		r

位/位域	名称	描述
31:17	保留	必须保持复位值
15	FSTAT	闪存状态位 1：闪存处于睡眠或掉电模式 0：闪存处于空闲模式
14:6	保留	必须保持复位值
5	ENDF	操作结束标志位 操作成功执行后，此位被硬件置1。软件写1清0。
4	WPERR	擦除/编程保护错误标志位 在受保护的页上擦除/编程操作时，此位被硬件置 1。软件写 1 清 0。
3	PGAERR	编程对齐错误标志位 当 SBUS 写数据不对齐时，此位被硬件置 1。软件写 1 清 0。
2	PGERR	编程错误标志位 当被编程区域状态不为 0xFFFF 时，对闪存编程，此位被硬件置 1。软件写 1 清 0。
1	保留	必须保持复位值
0	BUSY	闪存忙标志 当闪存操作正在进行时，此位被置 1。当操作结束或者出错，此位被清 0。

2.4.5. 控制寄存器（FMC_CTL）

地址偏移：0x10

复位值：0x0000 0080

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

保留	ENDIE	保留	ERRIE	OBWEN	FSTPG	LK	START	OBER	OBPG	保留	MER	PER	PG
	rw		rw	rw	rw	rs	rs	rw	rw		rw	rw	rw

位/位域	名称	描述
31:13	保留	必须保持复位值
12	ENDIE	操作结束中断使能位 软件置 1 和清 0 0：无硬件中断产生 1：使能操作结束中断
11	保留	必须保持复位值
10	ERRIE	出错中断使能位 软件置 1 和清 0 0：无硬件中断产生. 1：使能出错中断
9	OBWEN	选项字节擦除/编程使能位 当正确的序列写入 FMC_OBKEY 寄存器，此位由硬件置 1。此位可以被软件清 0。
8	FSTPG	主存储块快速编程命令位 软件置 1 和清 0 0：无作用 1：主存储块快速编程命令
7	LK	FMC_CTL0 寄存器锁定标志位 当正确的序列写入 FMC_KEY0 寄存器，此位由硬件清 0。此位可以由软件置 1。
6	START	向 FMC 发送擦除命令位 软件置 1 可以发送擦除命令到 FMC。当 BUSY 位被清 0 时，此位由硬件清 0。
5	OBER	选项字节擦除命令位 软件置 1 和清 0 0：无作用 1：选项字节擦除命令
4	OBPG	选项字节编程命令位 软件置 1 和清 0 0：无作用 1：选项字节编程命令
3	保留	必须保持复位值
2	MER	主存储块整片擦除命令位 软件置 1 和清 0 0：无作用 1：主存储块整片擦除命令
1	PER	主存储块页擦除命令位

软件置 1 和清 0
0：无作用
1：主存储块页擦除命令

0	PG	主存储块编程命令位 软件置 1 和清 0 0：无作用 1：主存储块编程命令

注意：当相应闪存操作完成后，该寄存器需处于复位状态。

2.4.6. 地址寄存器（FMC_ADDR）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[15:0]															
w															

位/位域	名称	描述
31:0	ADDR[31:0]	闪存擦除或编程地址 该位通过软件设置。 ADDR 位是闪存擦除/编程命令的地址

2.4.7. 选项字节状态寄存器（FMC_OBSTAT）

地址偏移：0x1C
复位值：0x0XXX XXXX

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						DATA[15:6]									
						r									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[5:0]						USER[7:0]								SPC	OBERR
r						r								r	r

位/位域	名称	描述
31:26	保留	必须保持复位值

25:10	DATA[15:0]	系统复位后保存选项字节的 DATA[15:0]部分
9:2	USER[7:0]	系统复位后保存选项字节块的 USER 字节
1	SPC	安全保护状态 0：未保护 1：已保护
0	OBERR	选项字节读错误位 当选项字节和它的补字节不匹配时此位由硬件置 1，选项字节被强制设置为 0xFF。

2.4.8. 擦除/编程保护寄存器（FMC_WP）

地址偏移：0x20
复位值：0xXXXX XXXX

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WP[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WP[15:0]															
r															

位/位域	名称	描述
31:0	WP[31:0]	系统复位后保存选项字节块的 WP[31:0]部分

2.4.9. 睡眠或掉电模式解锁寄存器(FMC_SLPKEY)

地址偏移：0x24
复位值：0xXXXX XXXX

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLPKEY[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLPKEY[15:0]															
w															

位/位域	名称	描述
31:0	SLPKEY[31:0]	RUN_SLP 位操作解锁寄存器 这些位仅能被软件写 写解锁值 SLPKEY1 和 SLPKEY2 到 SLPKEY[31:0]解锁 FMC_WS 寄存器的 RUN_SLP 位。 SLPKEY1 为 0x04152637

SLPKEY2 为 0xBCAD9E8F

2.4.10. 产品 ID 寄存器（FMC_PID）

地址偏移：0x100

复位值：0xXXXX XXXX

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31:0	PID[31:0]	产品保留 ID 寄存器 该寄存器为只读 上电后这些位始终不会改变，该寄存器在生产过程中被一次性编程。

3. 电源管理单元（PMU）

3.1. 简介

功耗设计是 GD32L23x 系列产品比较注重的问题之一。电源管理单元提供了十种省电模式，包括运行模式，运行模式 1，运行模式 2，睡眠模式，睡眠模式 1，睡眠模式 2，深度睡眠模式，深度睡眠模式 1，深度睡眠模式 2 和待机模式。这些模式能减少电源能耗，且使得应用程序可以在 CPU 运行时间要求、速度和功耗的相互冲突中获得最佳折衷。如*图 3-1. 电源域概览*所示，GD32L23x 系列设备有三个电源域，包括 V_{DD} / V_{DDA} 域，1.1V 域和备份域。V_{DD} / V_{DDA} 域由电源直接供电。在 V_{DD} / V_{DDA} 域中嵌入了一个 LDO，用来为 1.1V 域供电。在备份域中有一个电源切换器，当 V_{DD} 电源关闭时，电源切换器可以将备份域的电源切换到 V_{BAT} 引脚，此时备份域由 V_{BAT} 引脚（电池）供电。

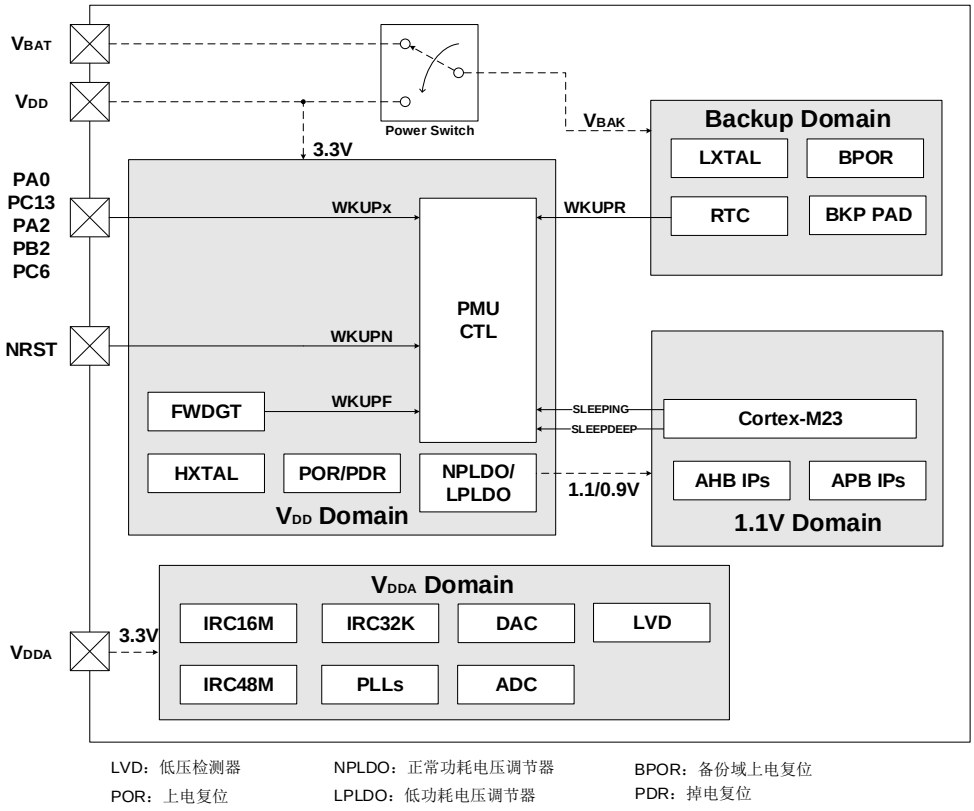
3.2. 主要特征

- 三个电源域：备份域、V_{DD} / V_{DDA}域和1.1V电源域。
- 十种省电模式：运行模式、运行模式1、运行模式2、睡眠模式、睡眠模式1、睡眠模式2、深度睡眠模式、深度睡眠模式1、深度睡眠模式2和待机模式。
- 内部电压调节器（LDO）为1.1V电源域提供1.1V电源或0.9V电源。
- 提供低电压检测器，当电压低于所设定的阈值时能发出中断或事件。
- 当V_{DD}供电关闭时，由V_{BAT}（电池）为备份域供电。
- LDO输出电压用于节约能耗。
- 低驱动模式用于在深入睡眠模式 / 深度睡眠模式1 / 深度睡眠模式2下超低功耗。
- SRAM1可单独断电。
- CAU可单独断电。

3.3. 功能说明

*图 3-1. 电源域概览*提供了 PMU 及相关电源域的内部结构框图。

图3-1. 电源域概览



3.3.1. 电池备份域

电池备份域由内部电源切换器来选择 V_{DD} 供电或 V_{BAT}（电池）供电，然后由 V_{BAK} 为备份域供电，该备份域包含 RTC（实时时钟）、LXTAL（低速外部晶体振荡器）和 BPOR（备份域上电复位），以及 PC13 至 PC15 共 3 个 PAD。为了确保备份域中寄存器的内容及 RTC 正常工作，当 V_{DD} 关闭时，V_{BAT} 引脚可以连接至电池或其他备份电源供电。电源切换器是由 V_{DD} / V_{DDA} 域掉电复位电路控制的。对于没有外部电池的应用，建议将 V_{BAT} 引脚通过 100nF 的外部陶瓷去耦电容连接到 V_{DD} 引脚上。

备份域的复位源包括备份域上电复位和备份域软件复位。在 V_{BAK} 没有完全上电前，BPOR 信号强制设备处于复位状态。应用软件可以通过设置 RCU_BDCTL 寄存器 BKPRST 位来触发备份域软件复位。

RTC的时钟源可以是低速内部32KHz RC振荡器（IRC32K）或低速外部晶体振荡器（LXTAL），或高速外部晶体振荡器（HXTAL）时钟32分频。当V_{DD}被关闭时，RTC只能选择LXTAL作为时钟源。在通过WFI / WFE指令进入省电模式之前，Cortex®-M23需要通过RTC寄存器预期的唤醒时间并启用唤醒功能，以实现RTC定时器唤醒事件。进入省电模式一定时间之后，当经过的时间与预设的唤醒时间匹配时，RTC将唤醒设备。RTC的配置和操作的细节将在*实时时钟（RTC）*来描述。

当备份域由V_{DD}供电（V_{BAK}连接至V_{DD}）时，以下功能可用：

- PC13可以作为通用I/O口或RTC功能引脚（参见*实时时钟（RTC）*）；
- PC14和PC15可以作为通用I/O口或LXTAL晶振引脚。

当备份域由V_{BAT}电源供电时（V_{BAK}连接至V_{BAT}），以下功能可用：

- PC13仅可以作为RTC功能引脚（参见*实时时钟（RTC）*）；
- PC14和PC15仅可作为LXTAL晶振引脚。

注意：由于 PC13 至 PC15 引脚是通过电源切换器供电的，电源切换器仅可通过小电流，因此当 PC13 至 PC15 的 GPIO 口在输出模式时，其工作的速度不能超过 2MHz(最大负载为 30pF)。

V_{DD} 可以通过一个内部电阻给外部电池充电。通过配置 PMU_CTL0 寄存器中 VCRSEL 位，可以选择内部电阻 5K 欧姆或 1.5K 欧姆用于外部 V_{BAT} 电池充电。将 PMU_CTL0 寄存器中 VCEN 位置 1 可以使能 V_{BAT} 电池充电。在 BKP_ONLY 模式，V_{BAT} 电池充电不可用。

注意：在 BKP_ONLY 模式下，V_{DD} 掉电，备份域由 V_{BAT} 引脚供电。

3.3.2. V_{DD} / V_{DDA} 电源域

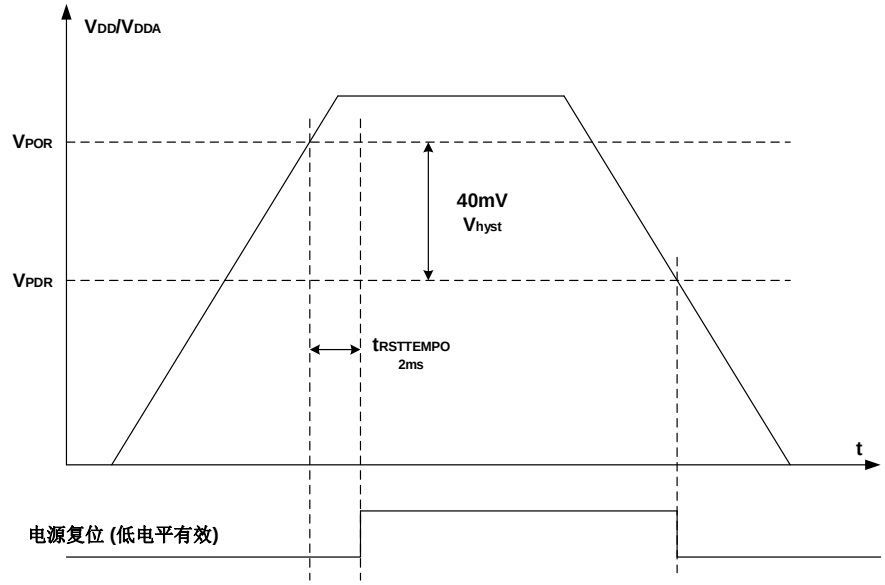
V_{DD} / V_{DDA} 域包括 V_{DD} 域和 V_{DDA} 域两部分。V_{DD} 域包括 HXTAL（高速外部晶体振荡器）、LDO（电压调节器）、POR / PDR（上电 / 掉电复位）、FWDGT（独立看门狗定时器）和除 PC13、PC14 和 PC15 之外的所有 PAD 等等。V_{DDA} 域包括 ADC / DAC（AD / DA 转换器）、IRC16M（内部 16MHz RC 振荡器）、IRC48M（内部 48M RC 振荡器）、IRC32K（内部 32KHz RC 振荡器）PLLs（锁相环）和 LVD（低电压检测器）等等。

V_{DD} 域

为 1.1V 域供电的 LDO（电压调节器），其复位后保持使能。可以被配置为不同的工作状态：包括睡眠模式（1.1V 全供电状态、0.9V 全供电状态和低功耗状态）、深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2（全供电或低功耗状态）和待机模式（关闭状态）。

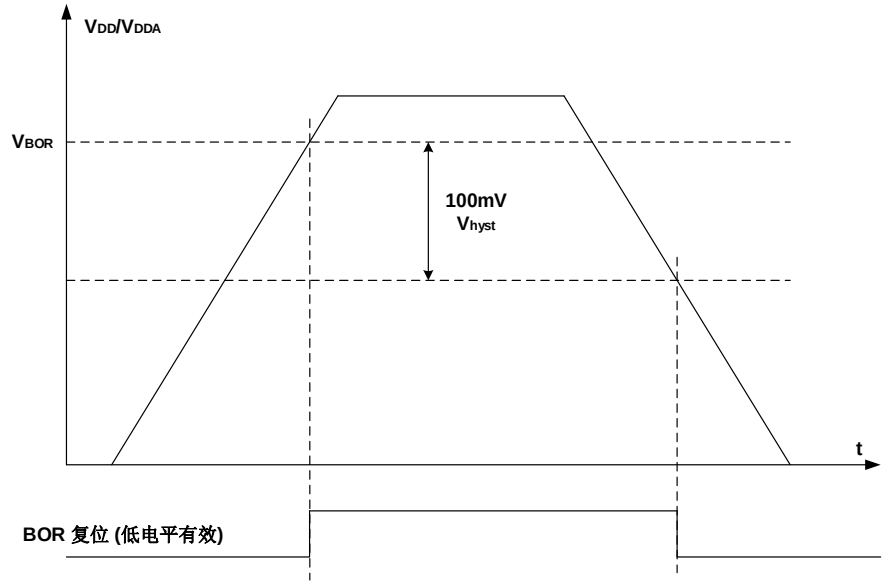
POR / PDR（上电 / 掉电复位）电路检测 V_{DD} / V_{DDA} 并在电压低于特定阈值时产生电源复位信号复位除备份域之外的整个芯片。*图 3-2. 上电 / 掉电复位波形图*显示了供电电压和电源复位信号之间的关系。V_{POR} 表示上电复位的阈值电压，典型值约为 1.60V，V_{PDR} 表示掉电复位的阈值电压，典型值约为 1.56V。迟滞电压 V_{hyst} 值约为 40mV。

图3-2. 上电 / 掉电复位波形图



BOR 电路检测 V_{DD} / V_{DDA} 并在电压低于选项字节的 BOR_TH 定义的阈值时产生电源复位信号复位除备份域之外的整个芯片。注意 POR / PDR（上电 / 掉电复位）电路总是处于检测状态。*图 3-3. BOR 波形图*显示了供电电压和 BOR 复位信号之间的关系。V_{BOR} 表示 BOR 复位的阈值电压，该值在选项字节 BOR_TH 中定义。迟滞电压 V_{hyst} 值为 100mV。

图3-3. BOR波形图



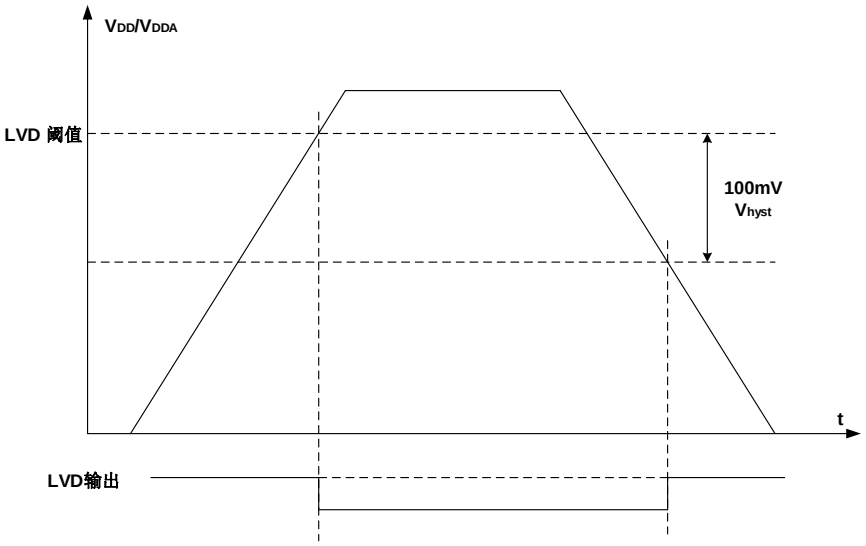
V_{DDA} 域

LVD 的功能是检测 V_{DD} / V_{DDA} 供电电压是否低于低电压检测阈值，该阈值由电源控制寄存器 0（PMU_CTL0）中的 LVDT[2:0]位进行配置。LVD 通过 LVDEN 置位使能，位于电源控制状态寄存器（PMU_CS）中的 LVDF 位表示低电压事件是否出现，该事件连接至 EXTI 的第 16 线，用户可以通过配置 EXTI 的第 16 线产生相应的中断。*图 3-4. LVD 阈值波形图*显示了 V_{DD} / V_{DDA} 供电电压和 LVD 输出信号的关系。(LVD 中断信号依赖于 EXTI 第 16 线的上升或下降沿配置)。

迟滞电压 V_{hyst} 值为 100mV。

注意：当 LVDT[2:0]位配置为“111”时，PB7 引脚上的输入电压与 0.8v 进行比较，LVDF 位表示输入电压高于或低于 0.8v。

图 3-4. LVD 阈值波形图



一般来说，数字电路由 V_{DD} 供电，而大多数的模拟电路由 V_{DDA} 供电。为了提高 ADC 和 DAC 的转换精度，为 V_{DDA} 独立供电可使模拟电路达到更好的特性。为避免噪声，V_{DDA} 通过外部滤波电路连接至 V_{DD}，相应的 V_{SSA} 通过特定电路连接至 V_{SS}。否则，当 V_{DD} 和 V_{DDA} 不是同一个电源提供时，在上电和运行过程中 V_{DD} 与 V_{DDA} 差值不超过 0.3V。

3.3.3. 1.1V 电源域

主要功能包括 Cortex®-M23 内核逻辑、AHB / APB 外设、备份域和 V_{DD} / V_{DDA} 域的 APB 接口等。当 1.1V 电压上电后，POR 将在 1.1V 域中产生一个复位序列，复位完成后，如果要进入指定的省电模式，须先配置相关的控制位，之后一旦执行 WFI 或 WFE 指令，设备便进入该省电模式。详细内容将在以下章节予以说明。

SRAM1 电源域

SRAM1（0x20004000~0x20007FFF）可独立断电。SRAM1 在系统复位后默认是上电的。为了降低运行模式 / 运行模式 1 / 运行模式 2 的功耗，可以将 SRAM1 断电。为了进一步降低低功耗模式（睡眠模式 / 睡眠模式 1 / 睡眠模式 2 / 深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2）的功耗，在进入低功耗模式之前可以将 SRAM1 断电。

COREOFF1 电源域

COREOFF1 域可单独断电。COREOFF1 域在系统复位后默认是断电的。在使用 COREOFF1 域中模块时需要将 COREOFF1 域上电。为了降低运行模式 / 运行模式 1 / 运行模式 2 的功耗，可以将 COREOFF1 域断电。为了进一步降低低功耗模式（睡眠模式 / 睡眠模式 1 / 睡眠模式 2 / 深度睡眠模式 / 深度睡眠模式 1 / 深度睡眠模式 2）的功耗，在进入低功耗模式之前可以将 COREOFF1 域断电。

COREOFF1 电源域包含 CAU 模块。

COREOFF0 电源域

在进入深度睡眠模式 2 时，COREOFF0 电源域断电，在退出深度睡眠模式 2 时，COREOFF0 电源域上电。

COREOFF0 电源域包含以下模块：

CPU / BUS / ADC / CMP / CRC / CTC / DAC / DMA / I2C0 / I2C1 / SLCD / TRNG / SPI0/ SPI1 / TIMER1 / TIMER2 / TIMER5 / TIMER6 / TIMER8 / TIMER11 / USART0 / USART1 / USART3 / USART4 / USBD。

注意：可以通过配置 PMU_CTL1 寄存器中 NRRD2 位来选择在进入/退出深度睡眠模式 2 时 CPU 寄存器的值是否保留。

3.3.4. 省电模式

系统复位或电源复位后，GD32L23x MCU 处于全功能状态且电源域全部处于供电状态。实现较低的功耗的方法有三种：减慢系统时钟（HCLK，PCLK1 和 PCLK2），关闭未使用的外设的时钟。此外，十种省电模式可以实现更低的功耗，它们是运行模式，运行模式 1，运行模式 2，睡眠模式，睡眠模式 1，睡眠模式 2，深度睡眠模式，深度睡眠模式 1，深度睡眠模式 2 和待机模式。

运行模式

在系统复位、上电复位或从待机模式唤醒产生复位后，MCU 进入运行模式，NPLDO（正常功耗 LDO）工作在 1.1V 模式。

运行模式 1

在运行模式下，NPLDO 必须通过配置 PMU_CTL0 寄存器中的 LDOVS 位来选择工作在 0.9V 模式。在该模式下，系统时钟频率不可以超过 16MHz。

运行模式 2

在运行模式或运行模式 1 下，NPLDO 必须通过配置 PMU_CTL0 寄存器中的 LDOVS 位来选择工作在 0.9V 模式，同时必须通过配置 PMU_CTL0 寄存器中的 LDNP 位来选择低驱动模式。在该模式下，系统时钟频率不可以超过 2MHz。

睡眠模式

睡眠模式与 Cortex®-M23 的 SLEEPING 模式相对应。在睡眠模式下，仅关闭 Cortex®-M23 的时钟。如需进入睡眠模式，只要清除 Cortex®-M23 系统控制寄存器中的 SLEEPDEEP 位，并执行一条 WFI 或 WFE 指令即可。如果睡眠模式是通过执行 WFI 指令进入的，任何中断都可以唤醒系统。如果睡眠模式是通过执行 WFE 指令进入的，任何唤醒事件都可以唤醒系统（如果 SEVONPEND 为 1，任何中断都可以唤醒系统，请参考 Cortex®-M23 技术手册）。由于无需在进入或退出中断上消耗时间，该模式所需的唤醒时间最短。

根据 Cortex®-M23 中 SCR（系统控制寄存器）的 SLEEPONEXIT 位，有两种睡眠进入机制可选：

- Sleep-now：如果SLEEPONEXIT位被清零，一旦执行WFI或WFE指令，MCU立即进入睡眠模式；
- Sleep-on-exit：如果SLEEPONEXIT位被置位，当系统从最低优先级的中断处理程序离开后，MCU立即进入睡眠模式。

睡眠模式 1

睡眠模式 1 对应于运行模式 1 的 Cortex®-M23 的 SLEEPING 模式。NPLDO 必须通过配置 PMU_CTL0 寄存器中的 LDOVS 位来选择工作在 0.9V。

睡眠模式 2

睡眠模式 2 对应于运行模式 2 的 Cortex®-M23 的 SLEEPING 模式。NPLDO 必须通过配置 PMU_CTL0 寄存器中的 LDOVS 位来选择工作在 0.9V。同时必须通过配置 PMU_CTL0 寄存器中的 LDNP 位来选择低驱动模式。

深度睡眠模式

深度睡眠模式与 Cortex®-M23 的 SLEEPDEEP 模式相对应。在深度睡眠模式下，1.1V 域中的所有时钟全部关闭，IRC16M、IRC48M、HXTAL 及 PLLs 也全部被禁用。SRAM 和寄存器中的内容被保留。根据 PMU_CTL0 寄存器的 LDNPDSP 位的配置，可控制 NPLDO 工作在正常模式或低功耗模式。进入深度睡眠模式之前，先将 Cortex®-M23 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL0 寄存器的 LPMOD 位域配置为“00”，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式。如果睡眠模式是通过执行 WFI 指令进入的，任何来自 EXTI 的中断可以将系统从深度睡眠模式中唤醒。如果睡眠模式是通过执行 WFE 指令进入的，任何来自 EXTI 的事件可以将系统从深度睡眠模式中唤醒（如果 SEVONPEND 为 1，任何来自 EXTI 的中断都可以唤醒系统，请参考 Cortex®-M23 技术手册）。刚退出深度睡眠模式时，IRC16M 被选中作为系统时钟。请注意，如果 LDO 工作在低驱动模式，那么唤醒时需额外的延时时间。

注意：为了顺利进入深度睡眠模式，所有 EXTI 线上的挂起状态（在 EXTI_PD 寄存器中）和相关外设标志位必须被复位，参考*表 6-3. EXTI 触发源*。否则，程序将直接跳过深度睡眠模式进入过程而继续执行下面的程序。

深度睡眠模式 1

深度睡眠模式 1 与 Cortex®-M23 的 SLEEPDEEP 模式相对应。在深度睡眠模式 1 下，1.1V 域中的所有时钟全部关闭，IRC16M、IRC48M、HXTAL 及 PLLs 也全部被禁用。LPLDO（低功耗 LDO）可以替代 NPLDO 正常工作。进入深度睡眠模式 1 之前，先将 Cortex®-M23 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL0 寄存器的 LPMOD 位域配置为“01”,然后执行 WFI 或 WFE 指令即可进入深度睡眠模式 1。如果睡眠模式是通过执行 WFI 指令进入的，任何来自 EXTI 的中断可以将系统从深度睡眠模式 1 中唤醒。如果睡眠模式是通过执行 WFE 指令进入的，任何来自 EXTI 的事件可以将系统从深度睡眠模式 1 中唤醒（如果 SEVONPEND 为 1,任何来自 EXTI 的中断都可以唤醒系统,请参考 Cortex®-M23 技术手册）。刚退出深度睡眠模式 1 时，IRC16M 被选中作为系统时钟。从深度睡眠模式 1 中唤醒需要额外

的延迟来唤醒 NPLDO。

注意：如果上电或者从待机模式唤醒，在进入深度睡眠模式 1 之前需要等待至少 600us。

深度睡眠模式 2

深度睡眠模式 2 与 Cortex®-M23 的 SLEEPDEEP 模式相对应。在深度睡眠模式 2 下，1.1V 域中的所有时钟全部关闭，IRC16M、IRC48M、HXTAL 及 PLLs 也全部被禁用。COREOFF0 / SRAM1 / COREOFF1 域停止供电。COREOFF0 / SRAM1 / COREOFF1 域寄存器中的内容全部丢失。LPLDO 可以替代 NPLDO 正常工作。进入深度睡眠模式 2 之前，先将 Cortex®-M23 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL0 寄存器的 LPMOD 位域配置为“10”，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式 2。如果睡眠模式是通过执行 WFI 指令进入的，任何来自 EXTI 的中断可以将系统从深度睡眠模式 2 中唤醒。如果睡眠模式是通过执行 WFE 指令进入的，任何来自 EXTI 的事件可以将系统从深度睡眠模式 2 中唤醒（如果 SEVONPEND 为 1，任何来自 EXTI 的中断都可以唤醒系统，请参考 Cortex®-M23 技术手册）。刚退出深度睡眠模式 2 时，IRC16M 被选中作为系统时钟。从深度睡眠模式 2 中唤醒需要额外的延迟来唤醒 NPLDO。

注意：如果上电或者从待机模式唤醒，在进入深度睡眠模式 2 之前需要等待至少 600us。

待机模式

待机模式是基于 Cortex®-M23 的 SLEEPDEEP 模式实现的。在待机模式下，整个 1.1V 域全部停止供电，NPLDO / LPLDO 关闭，同时包括 IRC16M、IRC48M、HXTAL 和 PLLs 也会被关闭。进入待机模式前，先将 Cortex®-M23 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL0 寄存器的 LPMOD 位域配置为“11”，再清除 PMU_CS 寄存器的 WUF 位，然后执行 WFI 或 WFE 指令，系统进入待机模式，PMU_CS 寄存器的 STBF 位状态表示 MCU 是否已进入待机模式。待机模式有四个唤醒源，包括来自 NRST 引脚的外部复位，RTC 闹钟 / 时间戳 / 侵入 / 自动唤醒事件，FWDGT 复位，WKUP 引脚的上升沿。待机模式可以达到最低的功耗，但唤醒时间最长。另外，一旦进入待机模式，SRAM 和 1.1V 电源域寄存器的内容都会丢失。退出待机模式时，会发生上电复位，复位之后 Cortex®-M23 将从 0x00000000 地址开始执行指令代码。

表3-1. 节电模式总结

模式	描述	LDO 状态	进入指令	唤醒	唤醒后模式	唤醒延时
运行	对所有时钟无影响，全部开启	NPLDO 开启 LPLDO 开启	系统 / 上电复位或从待机模式唤醒	-	-	-
运行 1	系统时钟 <= 16Mhz	NPLDO 开启 LPLDO 开启	LDOVS 配置为 0.9V	清除 LDVOS	-	-
运行 2	系统时钟 <= 2Mhz	NPLDO 工作在低驱动模式 LPLDO 开启	LDOVS 配置为 0.9V 且 LDNP 置 1	清除 LDVOS 和 LDNP	-	-
睡眠	仅关闭 CPU 时钟	NPLDO 开启 LPLDO 开启	SLEEPDEEP = 0，在运行模式下执行 WFI 或	若通过 WFI 进入，则任何中断均可唤醒； 若通过 WFE 进入，则	运行模式	-

模式	描述	LDO 状态	进入指令	唤醒	唤醒后模式	唤醒延时
			WFE	任何事件（或 SEVONPEND=1 时的中断）均可唤醒		
睡眠 1	仅关闭 CPU 时钟	NPLDO 开启 LPLDO 开启	SLEEPDEEP = 0，在运行模式 1 下执行 WFI 或 WFE	若通过 WFI 进入，则任何中断均可唤醒； 若通过 WFE 进入，则任何事件（或 SEVONPEND=1 时的中断）均可唤醒	运行模式 1	-
睡眠 2	仅关闭 CPU 时钟	NPLDO 工作在低驱动模式 LPLDO 开启	SLEEPDEEP = 0，在运行模式 2 下执行 WFI 或 WFE	若通过 WFI 进入，则任何中断均可唤醒； 若通过 WFE 进入，则任何事件（或 SEVONPEND=1 时的中断）均可唤醒	运行模式 2	-
深度睡眠	1、关闭 1.1V 电源域的所有时钟 2、关闭 IRC16M、IRC48M、HXTAL 和 PLLs	NPLDO 工作在低驱动模式或正常驱动模式 LPLDO 开启	SLEEPDEEP = 1，LPMOD = 00，执行 WFI 或 WFE	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒；若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND=1 时的中断）可唤醒	运行模式 / 运行模式 1 / 运行模式 2	IRC16M 唤醒时间+ Flash 唤醒时间
深度睡眠 1	1、关闭 1.1V 电源域的所有时钟 2、关闭 IRC16M、IRC48M、HXTAL 和 PLLs 3、LPLDO 代替 NPLDO	NPLDO 关闭 LPLDO 开启	SLEEPDEEP = 1，LPMOD = 01，执行 WFI 或 WFE	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒；若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND=1 时的中断）可唤醒	运行模式 / 运行模式 1 / 运行模式 2	IRC16M 唤醒时间+NPLDO 唤醒时间+ Flash 唤醒时间
深度睡眠 2	1、关闭 1.1V 电源域的所有时钟 2、关闭 IRC16M、IRC48M、HXTAL 和 PLLs 3、LPLDO 代替 NPLDO 4、COREOFF0 / SRAM1 / COREOFF1 掉电	NPLDO 关闭 LPLDO 开启	SLEEPDEEP = 1，LPMOD = 10，执行 WFI 或 WFE	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒；若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND=1 时的中断）可唤醒	运行模式 / 运行模式 1 / 运行模式 2	IRC16M 唤醒时间+NPLDO 唤醒时间+ Flash 唤醒时间
待机	1、关闭 1.1V 电源域的所有时钟 2、关闭 IRC16M、IRC48M、HXTAL 和	NPLDO 关闭 LPLDO 关闭	SLEEPDEEP = 1，LPMOD = 11，执行 WFI 或 WFE	1、NRST 引脚 2、WKUP 引脚 3、FWDGT 复位 4、RTC 闹钟	运行模式	IRC16M 唤醒时间+NPLDO 唤醒时间+ Flash 唤醒时间

模式	描述	LDO 状态	进入指令	唤醒	唤醒后模式	唤醒延时
	PLLs					
BKP_ONLY	V_{DD} 域 / 1.1V 域全部掉电	NPLDO 关闭 LPLDO 关闭	V_{DD} 关闭	V_{DD} 开启	运行模式	V_{DD} 上电序列

注意：在待机模式下，除了 NRST 引脚，配置为 RTC 功能的 PC13，用作 LXTAL 晶振引脚的 PC14 和 PC15，使能的 WKUPx 引脚，其他所有 I/O 都处于高阻态。

3.4. PMU 寄存器

PMU 基地址：0x4000 7000

3.4.1. 控制寄存器 0（PMU_CTL0）

地址偏移：0x00
复位值：0x0000 C000（从待机模式唤醒后复位）

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LDOVS[1:0]		VCRSEL	VCEN	LDNP	LDNPDSP	保留	BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	LPMOD[1:0]	
rw		rw	rw	rw	rw		rw	rw			rw	rc_w1	rc_w1	rw	

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:14	LDOVS[1:0]	选择 LDO 输出 在 PLL 关闭时，这些位由软件配置。在主 PLL 使能后，LDOVS 选择的电压生效。 如果主 PLL 关闭，LDO 输出低电压模式被选中（该位的值不变）。 0x：LDO 输出低电压模式（0.9V）。 1x：LDO 输出高电压模式（1.1V）。
13	VCRSEL	V_{BAT} 电池充电电阻的选择 0：5 k 欧姆电阻用于 V_{BAT} 电池充电。 1：1.5 k 欧姆电阻用于 V_{BAT} 电池充电。
12	VCEN	V_{BAT} 电池充电使能 0：禁能 V_{BAT} 电池充电。 1：使能 V_{BAT} 电池充电。
11	LDNP	在运行 / 睡眠模式下使用 NPLDO 时，工作在低驱动模式 0：使用 NPLDO 时，工作在正常驱动模式。 1：使用 NPLDO 时，低驱动模式被使能。
10	LDNPDSP	在深度睡眠模式下使用 NPLDO 时，工作在低驱动模式 0：使用 NPLDO 时，工作在正常驱动模式。 1：使用 NPLDO 时，低驱动模式被使能。
9	保留	必须保持复位值。
8	BKPWEN	备份域写使能 0：禁止对备份域寄存器的写访问。 1：允许对备份域寄存器的写访问。

		复位之后，任何对备份域寄存器的写访问都将被禁止。如需对备份域寄存器做写访问，需先将该位置 1。
7:5	LVDT[2:0]	低电压检测器阈值 000：2.1V 001：2.3V 010：2.4V 011：2.6V 100：2.7V 101：2.9V 110：3.0V 111：PB7 输入模拟电压（与 0.8V 进行比较）
4	LVDEN	低电压检测器使能 0：关闭低电压检测器。 1：开启低电压检测器。
3	STBRST	待机标志复位 0：无影响 1：复位待机标志 读该位，始终返回 0。
2	WURST	唤醒标志复位 0：无影响 1：复位唤醒标志 读该位，始终返回 0。
1:0	LPMOD[1:0]	选择 Cortex®-M23 进入 SLEEPDEEP 模式，MCU 进入的低功耗模式 00：深度睡眠模式 01：深度睡眠模式 1 10：深度睡眠模式 2 11：待机模式

3.4.2. 电源控制和状态寄存器（PMU_CS）

地址偏移：0x04
复位值：0x0000 0000（从待机模式唤醒后不复位）

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															NPRDY
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	LDOVSRF	保留	WUPEN4	WUPEN3	WUPEN2	WUPEN1	WUPEN0	保留					LVDF	STBF	WUF
	r		rw	rw	rw	rw	rw						r	r	r

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	NPRDY	NPLDO 就绪标志 0：NPLDO 未就绪。 1：NPLDO 就绪。
15	保留	必须保持复位值。
14	LDOVSRF	LDO 电压选择就绪标志 0：LDO 电压选择未就绪。 1：LDO 电压选择就绪。
13	保留	必须保持复位值。
12	WUPEN4	WKUP 引脚 4（PC6）唤醒使能 0：禁能 WKUP 引脚 4 唤醒功能。 1：使能 WKUP 引脚 4 唤醒功能。 如果 WUPEN4 在进入省电模式之前置 1，WKUP 引脚 4 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 4 为高电平有效，WKUP 引脚 4 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
11	WUPEN3	WKUP 引脚 3（PB2）唤醒使能 0：禁能 WKUP 引脚 3 唤醒功能。 1：使能 WKUP 引脚 3 唤醒功能。 如果 WUPEN4 在进入省电模式之前置 1，WKUP 引脚 3 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 3 为高电平有效，WKUP 引脚 3 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
10	WUPEN2	WKUP 引脚 2（PA2）唤醒使能 0：关闭 WKUP 引脚 2 唤醒功能。 1：开启 WKUP 引脚 2 唤醒功能。 如果 WUPEN2 在进入省电模式之前置 1，WKUP 引脚 2 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 2 为高电平有效，WKUP 引脚 2 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
9	WUPEN1	WKUP 引脚 1（PC13）唤醒使能 0：关闭 WKUP 引脚 1 唤醒功能。 1：开启 WKUP 引脚 1 唤醒功能。 如果 WUPEN1 在进入省电模式之前置 1，WKUP 引脚 1 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 1 为高电平有效，WKUP 引脚 1 内部被配置为输入下拉模式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
8	WUPEN0	WKUP 引脚 0（PA0）唤醒使能 0：关闭 WKUP 引脚 0 唤醒功能。 1：开启 WKUP 引脚 0 唤醒功能。 如果 WUPEN0 在进入省电模式之前置 1，WKUP 引脚 0 的上升沿会将系统从省电模式唤醒。由于 WKUP 引脚 0 为高电平有效，WKUP 引脚 0 内部被配置为输入下拉模

		式。当在输入已经为高的时候置位该控制位，将会触发一个唤醒事件。
7:3	保留	必须保持复位值。
2	LVDF	低电压状态标志 0：低电压事件没出现（V_{DD} 高于设定的 LVD 阈值）。 1：低电压事件出现（V_{DD} 等于或低于 LVD 阈值）。 **注意：**LVD 功能在待机模式被禁用。
1	STBF	待机标志 0：设备没进入过待机模式。 1：设备曾进入过待机模式。 该位只能由 POR / PDR 或通过置位 PMU_CTL0 寄存器的 STBRST 位来清零。
0	WUF	唤醒标志 0：没有收到唤醒事件。 1：唤醒事件由 WKUP 引脚或 RTC 事件包括 RTC 闹钟事件，时间戳事件，侵入事件和自动唤醒事件触发。 该位只能由 POR / PDR 或通过设置 PMU_CTL0 寄存器的 WURST 位来清零。

3.4.3. 控制寄存器 1（PMU_CTL1）

地址偏移：0x08
复位值：0x0000 0000（从待机模式唤醒后复位）

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														SRAM1PD2	NRRD2
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										CORE1WAKE	CORE1SLEEP	保留		SRAM1PWAKE	SRAM1PSLEEP
										rw	rw			rw	rw

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	SRAM1PD2	进入深度睡眠模式 2 时，SRAM1 电源状态 0：SRAM1 掉电。 1：SRAM1 电源状态与运行 / 运行 1 / 运行模式 2 一样。 **注意：**当从深度睡眠 2 模式唤醒时，SRAM1 电源状态与进入深度睡眠 2 模式之前一致。
16	NRRD2	在深度睡眠 2 模式下没有保留寄存器 0：CPU 有保留寄存器。 1：没有保留寄存器。

15:6	保留	必须保持复位值。
5	CORE1WAKE	COREOFF1 域唤醒 当 MCU 处于运行 / 运行 1 / 运行模式 2 下，并且 COREOFF1 处于睡眠模式，该位可以由软件置 1。 该位由硬件清零。
4	CORE1SLEEP	COREOFF1 域掉电 当 MCU 处于运行 / 运行 1 / 运行模式 2 下，并且 COREOFF1 处于运行模式，该位可以由软件置 1。 该位由硬件清零。
3:2	保留	必须保持复位值。
1	SRAM1PWAKE	SRAM1 唤醒 当 MCU 处于运行 / 运行 1 / 运行模式 2 下，并且 SRAM1 处于睡眠模式，该位可以由软件置 1。 该位由硬件清零。
0	SRAM1PSLEEP	SRAM1 掉电 当 MCU 处于运行 / 运行 1 / 运行模式 2 下，并且 SRAM1 处于运行模式，该位可以由软件置 1。 该位由硬件清零。

3.4.4. 状态寄存器（PMU_STAT）

地址偏移：0x0C
复位值：0x0000 0018（从待机模式唤醒后不复位）

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
								r		r				r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										CORE1PS_ACTIVE	CORE1PS_SLEEP	SRAM1PS_ACTIVE	SRAM1PS_SLEEP	DPF2	保留
										r	r	r	r	rc_w0	

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	CORE1PS_ACTIVE	COREOFF1 域处于运行状态。
4	CORE1PS_SLEEP	COREOFF1 域处于睡眠状态。
3	SRAM1PS_ACTIVE	SRAM1 处于运行状态。

2	SRAM1PS_SLEEP	SRAM1 处于睡眠状态。
1	DPF2	深度睡眠模式 2 状态标志位。当进入深度睡眠模式 2 时该位由硬件置位。软件写 0 清除该位。
0	保留	必须保持复位值。

3.4.5. 参数寄存器（PMU_PAR）

地址偏移：0x10
复位值：0x040A 2064

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TWKEN	TWKSRAM1EN	TWKCORE1EN	TWK_CORE1[7:0]								TSW_IRC16MCNT[4:0]				
rw	rw	rw	rw										rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TWK_SRAM1[7:0]								TWK_CORE0[7:0]							
rw								rw							

位/位域	名称	描述
31	TWKEN	唤醒深度睡眠模式 2 时是否使用软件值 0：在唤醒深度睡眠模式 2 时，使用硬件应答信号。 1：在唤醒深度睡眠模式 2 时，使用软件设定值，该值由 TWK_CORE0[7:0]来设定。
30	TWKSRAM1EN	唤醒 SRAM1 电源域时是否使用软件值 0：唤醒 SRAM1 时，使用硬件应答信号。 1：唤醒 SRAM1 时，使用软件设定值，该值由 TWK_SRAM1[7:0]来设定。
29	TWKCORE1EN	唤醒 COREOFF1 时是否使用软件值 0：唤醒 COREOFF1 时，使用硬件应答信号。 1：唤醒 COREOFF1 时，使用软件设定值，该值由 TWK_CORE1[7:0]来设定。
28:21	TWK_CORE1[7:0]	COREOFF1 域电源开关唤醒时间。步长为 4 个时钟，最大 64us。
20:16	TSW_IRC16MCNT[4:0]	当进入深度睡眠模式时，切换到 IRC16M 时钟。等待 IRC16M 计数后设置深度睡眠状态。默认值为 10 个 IRC16M 时钟。
15:8	TWK_SRAM1[7:0]	SRAM1 域电源开关的唤醒时间。步长为 4 个 IRC16M 时钟，最大 64us。
7:0	TWK_CORE0[7:0]	COREOFF0 域电源开关的唤醒时间。步长为 2 个 IRC16M 时钟，最大 32us。

4. 复位和时钟单元（RCU）

4.1. 复位控制单元（RCTL）

4.1.1. 简介

GD32L23x复位控制包括三种复位控制：电源复位、系统复位和备份域复位。电源复位又称为冷复位，电源启动时复位除了备份域的所有系统。除了SW-DP控制器和备份域，系统复位将复位处理器内核和外设IP部分。备份域复位复位备份区域。复位被外部信号、内部事件和复位发生器触发。接下来的章节将详细介绍这些复位。

4.1.2. 功能说明

电源复位

当以下事件中之一发生时，产生电源复位：1、上电/掉电复位（POR/PDR复位）；2、从待机模式中返回后由内部复位发生器产生。电源复位复位所有的寄存器除了备份域。电源复位为低电平有效，当内部LDO电源基准准备好提供1.1V电压给GD32L23x产品时，电源复位电平将变为无效。复位入口向量被固定在存储器映射地址0x0000_0004。

系统复位

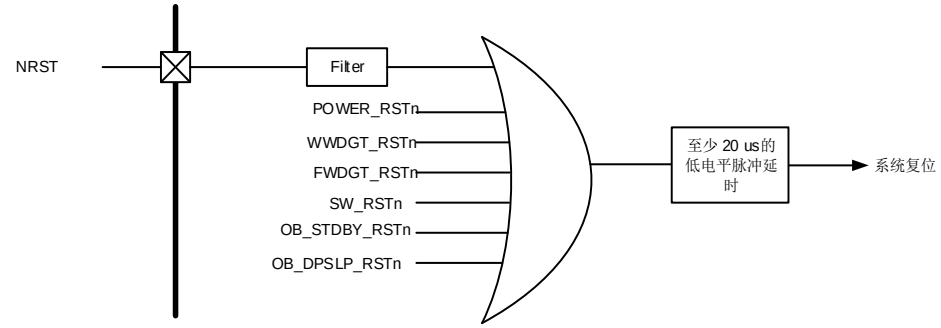
当发生以下任一事件时，产生一个系统复位：

- 电源复位（POWER_RSTn）；
- 外部引脚复位（NRST）；
- 窗口看门狗定时器计数终止（WWDGT_RSTn）；
- 独立看门狗定时器计数终止（FWDGT_RSTn）；
- Cortex®-M23的中断应用和复位控制寄存器中的SYSRESETREQ位置‘1’（SW_RSTn）；
- 用户选择字节寄存器nRST_STDBY设置为0，并且进入待机模式时（OB_STDBY_RSTn）；
- 用户选择字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时（OB_DPSLP_RSTn）。

除了SW-DP控制器和备份域，系统复位将复位处理器内核和外设IP部分。

系统复位脉冲发生器保证每一个复位源（外部或内部）都能有至少20μs的低电平脉冲延时。

图 4-1. 系统复位电路



备份域复位

当以下事件之一发生时，产生备份域复位。1、设置备份域控制寄存器中的BKPRST位为‘1’；2、备份域电源上电复位（V_{DD}重新上电）。

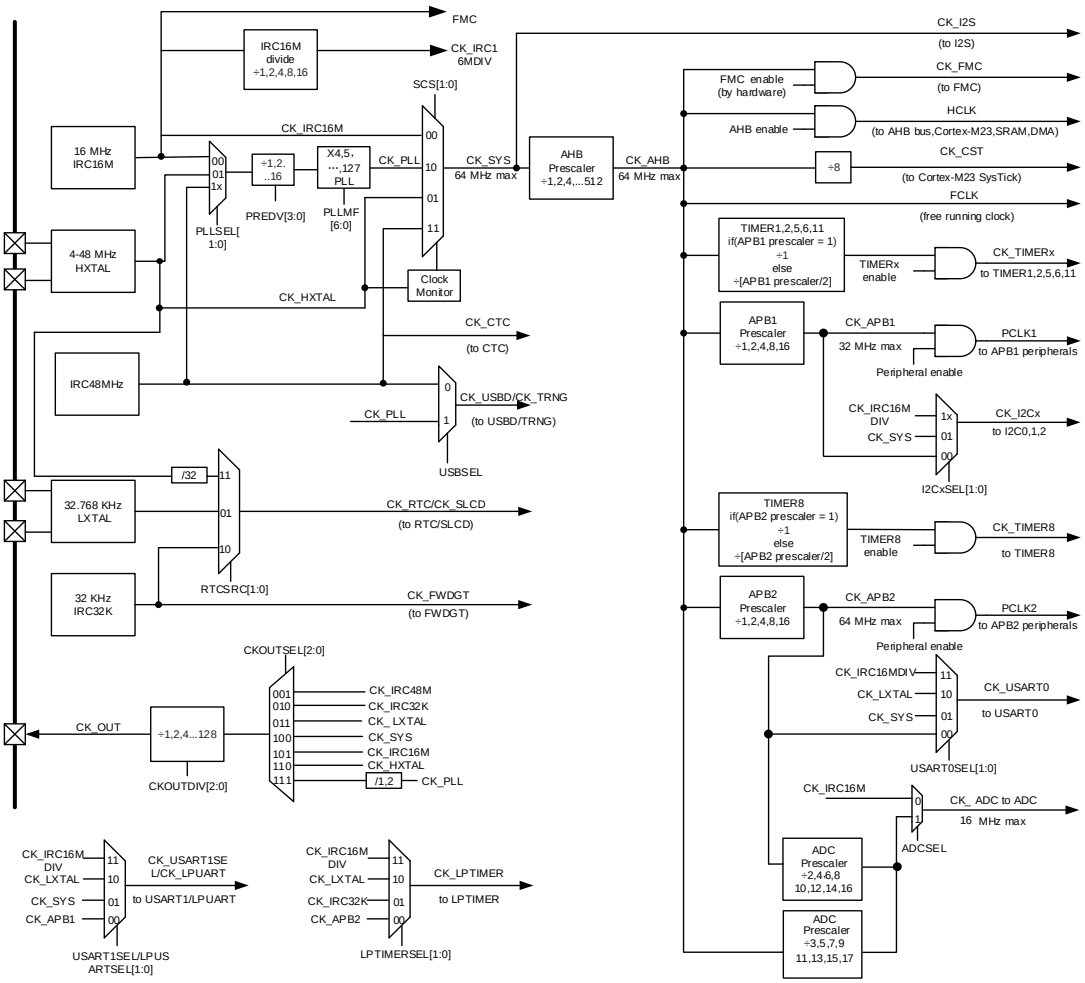
4.2. 时钟控制单元（CCTL）

4.2.1. 简介

时钟控制单元提供了一系列频率和时钟功能，包括一个内部16M RC振荡器时钟（IRC16M）、一个内部48M RC振荡器时钟（IRC48M）、一个外部高速晶体振荡器时钟（HXTAL）、一个内部低速RC振荡器时钟（IRC32K）、一个外部低速晶体振荡器时钟（LXTAL）、一个锁相环（PLL）、一个HXTAL时钟监视器、时钟预分频器、时钟多路复用器和时钟选通电路。

AHB、APB和Cortex®-M23时钟都源自系统时钟（CK_SYS），系统时钟的时钟源为IRC16M、HXTAL或PLL。系统时钟的最大运行时钟频率可以达到64MHz。

图4-2. 时钟树



预分频器可以配置AHB、APB2和APB1域的时钟频率。AHB、APB2和APB1域的最高时钟频率分别为64MHz、64MHz和32MHz。RCU通过AHB时钟（HCLK）8分频后作为Cortex系统定时器（SysTick）的外部时钟。通过对SysTick控制与状态寄存器的设置，可选择上述时钟或APB（HCLK）时钟作为SysTick时钟。

在GD32L23x产品中ADC时钟由APB2时钟经2、4、6、8、10、12、14、16分频或由AHB时钟经3、5、7、9、11、13、15、17分频或IRC16M获得，它们是通过设置配置寄存器2（RCU_CFG2）的ADCSEL位来选择ADC时钟源的。

USART0的时钟可以选择IRC16MDIV时钟、LXTAL时钟、系统时钟或APB2时钟，通过设置配置寄存器2（RCU_CFG2）的USART0SEL位域来选择。

USART1的时钟可以选择IRC16MDIV时钟、LXTAL时钟、系统时钟或APB1时钟，通过设置配置寄存器2（RCU_CFG2）的USART1SEL位域来选择。

LPUART的时钟可以选择IRC16MDIV时钟、LXTAL时钟、系统时钟或APB1时钟，通过设置配置寄存器2（RCU_CFG2）的LPUARTSEL位域来选择。

I2Cx（x = 0, 1, 2）的时钟可以选择IRC16MDIV时钟、系统时钟或APB1时钟，通过设置配置寄存器2（RCU_CFG2）的I2CxSEL（x = 0, 1, 2）位域来选择。

RTC/SLCD时钟可以选择LXTAL时钟、IRC32K时钟或HXTAL时钟32分频，通过设置备用域控

制寄存器（RCU_BDCTL）的RTCSRC位域来选择。

FWDGT时钟可以选择IRC32K时钟，当FWDGT启动时强制选择。

LPTIEMR的时钟可以选择IRC16MDIV时钟、LXTAL时钟、系统时钟或APB2时钟，通过设置配置寄存器2（RCU_CFG2）的LPTIMERSEL位域来选择。

如果APB时钟分频系数为1，定时器的时钟频率与所在AHB总线频率一致。否则，定时器的时钟频率被设为与其相连的APB总线频率的2倍。

4.2.2. 主要特征

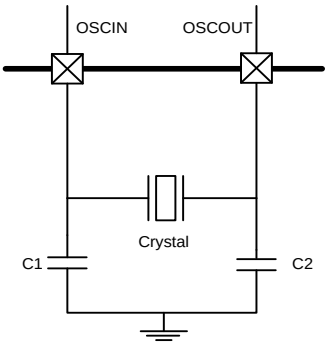
- 4到48 MHz外部高速晶体振荡器（HXTAL）；
- 16 MHz内部高速RC振荡器（IRC16M）；
- 48 MHz内部高速RC振荡器（IRC48M）；
- 32,768 Hz外部低速晶体振荡器（LXTAL）；
- 32 kHz内部低速RC振荡器（IRC32K）；
- PLL时钟源可以是HXTAL，IRC16M或IRC48M；
- HXTAL和LXTAL时钟监视。

4.2.3. 功能说明

高速外部晶体振荡器时钟（HXTAL）

4到48MHz的外部振荡器可为系统提供更为精确的主时钟。带有特定频率的晶体必须靠近两个HXTAL的引脚。和晶体连接的外部电阻和电容必须根据所选择的振荡器来调整。

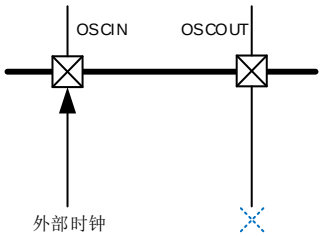
图4-3. HXTAL时钟源



HXTAL晶体可以通过设置时钟控制寄存器RCU_CTL的HXTALEN位来启动或关闭，在时钟控制寄存器RCU_CTL中的HXTALSTB位用来指示高速外部振荡器是否已稳定。在启动时，直到这一位被硬件置‘1’，时钟才被释放出来。这个特定的延迟时间又称启动时间。当HXTAL时钟稳定后，如果在时钟中断寄存器RCU_INT中的相应中断使能位HXTALSTBIE位被置‘1’，将会产生相应中断。在这一点上，HXTAL时钟可以被直接用作系统时钟源或者PLL输入时钟。

将控制寄存器RCU_CTL的HXTALBPS和HXTALEN位置‘1’可以设置外部时钟旁路模式。旁路输入时，信号接至OSCIN，OSCOUT保持悬空状态，如*图4-4. 旁路模式下HXTAL时钟源*所示。此时，CK_HXTAL等于驱动OSCIN管脚的外部时钟。

图4-4. 旁路模式下HXTAL时钟源



高速内部 16MHz RC 振荡器时钟（IRC16M）

高速内部16MHz RC振荡器时钟，简称IRC16M时钟，拥有16MHz的固定频率，设备上电后CPU默认选择的时钟源就是IRC16M时钟。IRC16M RC振荡器能够在不需要任何外部器件的条件下提供更低成本类型的时钟源。IRC16M晶体可以通过设置时钟控制寄存器（RCU_CTL）中的IRC16MEN位被启动和关闭。时钟控制寄存器RCU_CTL中的IRC16MSTB位用来指示IRC16M内部RC振荡器是否稳定。IRC16M振荡器的启动时间比HXTAL晶体振荡器要更短。如果时钟中断寄存器RCU_INT中的相应中断使能位IRC16MSTBIE被置‘1’，在IRC16M稳定以后，将产生一个中断。IRC16M时钟也可用作PLL输入时钟。

工厂会校准IRC16M时钟频率的精度，但是它的精度仍然比HXTAL时钟要差。用户需求、环境条件和成本将决定选择哪个时钟作为系统时钟源。

如果HXTAL或者PLL是系统时钟源，为了最大程度减小系统从深度睡眠模式启动的时间，系统从深度睡眠模式初始唤醒的时候硬件强制IRC16M时钟作为系统时钟。

在深度睡眠模式下，可以通过LPUART / USART0 / USART1 / I2C0 / I2C1 / I2C2打开IRC16M。如果IRC16M在深度睡眠状态下打开，则应禁用未工作的外围设备以节省电源。

内部 48M RC 振荡器时钟（IRC48M）

内部48MHz RC振荡器时钟，简称IRC48M时钟，拥有48MHz的固定频率，当使用USBD模块时，IRC48M振荡器在不需要任何外部器件的条件下为用户提供了一种成本更低的时钟源选择。IRC48M RC振荡器可以通过设置RCU_CTL寄存器中的IRC48MEN位被启动和关闭。RCU_ADDCTL寄存器中的IRC48MSTB位用来指示内部48MHz RC振荡器是否稳定。如果RCU_ADDINT寄存器中的相应中断使能位IRC48MSTBIE被置‘1’，在IRC48M稳定以后，将产生一个中断。IRC48M时钟可做为USBD的系统时钟。

工厂会校准IRC48M时钟频率的精度，但是它的精度仍然不够精准。因为USB模块需要的时钟频率必须满足48MHz（500ppm）。CTC单元提供了一种硬件自动执行动态调整的功能将IRC48M时钟调整到需要的频率。

锁相环（PLL）

内部锁相环PLL通过对输入参考频率为4 ~ 48MHz的时钟基准2 ~ 64倍频，可以提供16 ~ 64 MHz的时钟输出。

PLL可以通过设置时钟控制寄存器（RCU_CTL）中的PLLEN位被启动和关闭。时钟控制寄存器RCU_CTL中的PLLSTB位用来指示PLL时钟是否稳定。如果时钟中断寄存器RCU_INT中的

相应中断使能位PLLSTBIE被置‘1’，在PLL稳定以后，将产生一个中断。

低速外部晶体振荡器时钟（LXTAL）

LXTAL晶体是一个32.768kHz的低速外部晶体或陶瓷谐振器。它为实时时钟电路提供一个低功耗且精确的时钟源。LXTAL时钟可以通过设置备份域控制寄存器（RCU_BDCTL）中的LXTALEN位被启动和关闭。备份域控制寄存器RCU_BDCTL中的LXTALSTB位用来指示LXTAL时钟是否稳定。如果时钟中断寄存器RCU_INT中的相应中断使能位LXTALSTBIE被置‘1’，在LXTAL稳定以后，将产生一个中断。

将备份域控制寄存器RCU_BDCTL的LXTALBPS和LXTALEN位置‘1’可以选择外部时钟旁路模式。CK_LXTAL与连到OSC32IN脚上外部时钟信号一致。

当LPUART / USART0 / USART1使用LXTAL作为功能时钟时，可以打开LXTAL。

低速内部 RC 振荡器时钟（IRC32K）

IRC32K RC振荡器时钟担当一个低功耗时钟源的角色，它的时钟频率大约32kHz，为独立看门狗定时器和实时时钟电路提供时钟。IRC32K提供低成本的时钟源，因为不需要外部器件。IRC32K RC振荡器可以通过设置复位源/时钟寄存器RCU_RSTSCK中的IRC32KEN位被启动和关闭。复位源/时钟寄存器RCU_RSTSCK中的IRC32KSTB位用来指示IRC32K时钟是否已稳定。如果时钟中断寄存器RCU_INT中的相应中断使能位IRC32KSTBIE被置‘1’，在IRC32K稳定以后，将产生一个中断。

系统时钟（CK_SYS）选择

系统复位后，IRC16M时钟被选为系统时钟，改变时钟配置寄存器RCU_CFG0中的系统时钟变换位SCS可以切换系统时钟源为HXTAL或PLL。当SCS的值改变，系统时钟将使用原来的时钟源继续运行直到转换的目标时钟源稳定。当一个时钟源被直接或通过PLL间接作为系统时钟时，它将不能被停止。

HXTAL 时钟监视器（CKM）

设置时钟控制寄存器RCU_CTL中的HXTAL时钟监视使能位CKMEN，HXTAL可以使能时钟监视功能。该功能必须在HXTAL启动延迟完毕后使能，在HXTAL停止后禁止。一旦监测到HXTAL故障，HXTAL将自动被禁止，时钟中断寄存器RCU_INT中的HXTAL时钟阻塞标志位CKMIF将被置‘1’，产生HXTAL故障事件。这个故障引发的中断和Cortex-M23的不可屏蔽中断相连。如果HXTAL被选作系统或PLL的时钟源，HXTAL故障将促使选择IRC16M为系统时钟源且PLL将被自动禁止。

LXTAL 时钟监视器（LCKM）

设置时钟控制寄存器RCU_CTL中的LXTAL时钟监视使能位LXTALCKMEN，LXTAL可以使能时钟监视功能。该功能必须在LXTAL启动延迟完毕和IRC32K使能后使能。

LXTAL上的时钟监视器在除VBAT以外的所有模式下工作。如果在外部32 kHz振荡器上检测到故障，可以向CPU发送中断。

然后，软件必须禁用LXTALCKMEN位，停止有缺陷的32 kHz振荡器，并更改RTC时钟源，或采取任何必要的措施来保护应用程序。

当LXTALCKMEN启用时，一个4位加一个计数器将在IRC32K域工作。如果LXTAL时钟卡在0/1错误或减慢约20KHz，计数器将溢出。将发现LXTAL时钟故障。

时钟输出功能

时钟输出功能输出从32kHz到64MHz的时钟。通过设置时钟配置寄存器RCU_CFG0中的CK_OUT时钟源选择位CKOUTSEL能够选择不同的时钟信号。相应的GPIO引脚应该被配置成复用功能I/O（AFIO）模式来输出选择的时钟信号。

表 4-1. 时钟源的选择

时钟源选择位	时钟源
000	无时钟
001	CK_IRC48M
010	CK_IRC32K
011	CK_LXTAL
100	CK_SYS
101	CK_IRC16M
110	CK_HXTAL
111	CK_PLL 或 CK_PLL/2

通过配置时钟配置寄存器RCU_CFG0的CKOUTDIV[2:0]位，可以将输出时钟按比例分频，进而降低CK_OUT频率。

深度睡眠 1/2 模式时钟控制

当MCU处于深度睡眠1 / 2模式时，LPUART / USART0 / USART1外设时钟由LXTA提供且LXTAL时钟使能时，则LPUART / USART0 / USART1外设可以唤醒MCU。

如果LPUART / USART0 / USART1时钟选择IRC16M_DIV处于深度睡眠1 / 2模式时，则它们能够打开IRC16M时钟或关闭IRC16M时钟，从而使LPUART / USART0 / USART1 / I2C0 / I2C1 / I2C2从深度睡眠模式唤醒。

如果LPUART / USART0 / USART1时钟选择LXTAL处于深度睡眠1 / 2模式时，则它们能够打开LXTAL时钟或关闭LXTAL时钟（如果LXTAL由软件打开，则LPUART / USART0 / USART1不能关闭LXTAL），从而使LPUART从深度睡眠1/2模式唤醒。

如果I2C0 / I2C1 / I2C2选择IRC16M_DIV作为时钟源并处于深度睡眠1/2模式，则它们能够打开或关闭IRC16M时钟，从而使I2C0 / I2C1 / I2C2从深度睡眠1/2模式唤醒。

如果FMC和PMU在深度睡眠1/2模式下工作时，可以打开或关闭IRC16M时钟。

为了在深度睡眠1/2模式下省电，如果FMC/ LPUART / USART0 / USART1未在深度睡眠1/2模式下工作，则它们的时钟可以单独选通。但I2C0 / I2C1 / I2C2、ADC、LPTIMER、PMU功能时钟不能由硬件选通，可以由软件禁用。

4.3. RCU 寄存器

RCU基地址：0x4002 1000

4.3.1. 控制寄存器（RCU_CTL）

地址偏移：0x00
复位值：0x0000 XX83 X表示未定义。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						PLLSTB	PLLEN	LXTALCKMD	LXTALCKMEN	IRC48MSTB	IRC48MEN	CKMEN	HXTALBPS	HXTALSTB	HXTALEN
						r	rw	r	rw	r	rw	rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC16MCALIB[7:0]								IRC16MADJ[4:0]					保留.	IRC16MSTB	IRC16MEN
r								rw						r	rw

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	PLLSTB	PLL时钟稳定标志位 硬件置‘1’来指示PLL输出时钟是否稳定待用。 0：PLL没稳定 1：PLL稳定
24	PLLEN	PLL使能 软件置位或复位。如果PLL时钟作为系统时钟的时候该位不能被复位。进入深度睡眠或待机模式时硬件自动复位。 0：PLL被关闭 1：PLL被打开
23	LXTALCKMD	LXTAL时钟故障检测 由硬件置位，当外部32 kHz振荡器（LXTAL）上的时钟安全系统检测到故障。 0：LXTAL（32 kHz振荡器）上未检测到故障 1：在LXTAL（32 kHz振荡器）上检测到故障
22	LXTALCKMEN	LXTAL时钟监视使能 0：禁止LXTAL时钟监视器 1：使能LXTAL时钟监视器 通过软件设置，启用LXTAL（32 kHz振荡器）上的时钟安全系统。LXTALCKMEN必须在LXTAL已启用（LXTALEN位已启用）和就绪（LXTALSTB标志由硬件设置）
21	IRC48MSTB	内部 48MHz RC 振荡器时钟稳定标志位 硬件置‘1’来指示 IRC48M 振荡器时钟是否稳定待用 0：IRC48M 未稳定

		1：IRC48M已稳定
20	IRC48MEN	内部 48MHz RC 振荡器使能 由软件置位和复位。当进入深度睡眠或待机模式后由硬件复位 0：关闭 IRC48M 时钟 1：打开IRC48M时钟
19	CKMEN	HXTAL时钟监视使能 0：禁止外部4 ~ 48 MHz晶体振荡器（HXTAL）时钟监视器 1：使能外部4 ~ 48 MHz晶体振荡器（HXTAL）时钟监视器 当硬件监测到HXTAL时钟一直停留在低或者高的状态，内部硬件将切换系统时钟到IRC16M RC时钟。恢复原来系统时钟的方式有以下几种：外部复位，上电复位，软件清CKMIF位。 **注意：**使能HXTAL时钟监视器以后，硬件无视控制位IRC16MEN的状态，自动使能IRC16M时钟。
18	HXTALBPS	外部晶体振荡器（HXTAL）时钟旁路模式使能 只有在HXTALEN位为0时，HXTALBPS位才可写。 0：禁止HXTAL旁路模式 1：使能HXTAL旁路模式，HXTAL输出时钟等于输入时钟
17	HXTALSTB	外部晶体振荡器（HXTAL）时钟稳定状态标志位 硬件置‘1’来指示HXTAL振荡器时钟是否稳定待用。 0：HXTAL振荡器未稳定 1：HXTAL振荡器已稳定
16	HXTALEN	外部高速振荡器时钟使能 软件置‘1’或清‘0’。如果HXTAL时钟或者PLL输入时钟作为系统时钟，该位不能被复位。进入深度睡眠或待机模式时硬件自动复位。 0：禁止外部4 ~ 48 MHz晶体振荡器 1：使能外部4 ~ 48 MHz晶体振荡器
15:8	IRC16MCALIB[7:0]	高速内部振荡器校准值寄存器 上电时自动加载这些位
7:3	IRC16MADJ[4:0]	高速内部振荡器时钟调整值 这些位由软件置位，最终调整值为IRC16MADJ当前值加上IRC16MCALIB[7:0]位的值。最终调整值应该调整IRC16M到16 MHz ± 1%。
2	保留	必须保持复位值。.
1	IRC16MSTB	高速内部（IRC16M）时钟稳定状态标志位 硬件置‘1’来指示IRC16M振荡器时钟是否稳定待用。 0：IRC16M振荡器未稳定 1：IRC16M振荡器已稳定
0	IRC16MEN	高速内部振荡器使能 软件复位置位。如果IRC16M时钟用作系统时钟时该位不能被复位。当从待机或深度睡眠模式返回或在HXTALCKM置位的情况下用作系统时钟的HXTAL振荡器发生

故障时，该位由硬件置1来启动IRC16M振荡器。

0：内部16 MHz RC振荡器关闭

1：内部16 MHz RC振荡器开启

4.3.2. 配置寄存器 0（RCU_CFG0）

地址偏移：0x04

复位值：0x003C 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLDV	CKOUTDIV[2:0]			PLLMF[6]	CKOUTSEL[2:0]			PLLMF[5:0]						PLLSEL	
rw	rw			rw	rw			rw						rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPSC[1:0]		APB2PSC[2:0]			APB1PSC[2:0]			AHBPSC[3:0]				SCSS[1:0]		SCS[1:0]	
rw		rw			rw			rw				r		rw	

位/位域	名称	描述
31	PLLDV	CK_PLL 1或2分频来用作CK_OUT 0：CK_PLL 2分频用作CK_OUT 1：CK_PLL用作CK_OUT
30:28	CKOUTDIV[2:0]	CK_OUT分频器，来降低CK_OUT频率 CK_OUT的选择参考RCU_CFG0的26:24位。 000：CK_OUT不分频 001：CK_OUT 2分频 010：CK_OUT 4分频 011：CK_OUT 8分频 100：CK_OUT 16分频 101：CK_OUT 32分频 110：CK_OUT 64分频 111：CK_OUT 128分频
27	PLLMF[6]	PLLMF的位6 参考RCU_CFG0的位23:18
26:24	CKOUTSEL[2:0]	CK_OUT时钟源选择 软件置位或清零。 000：没有时钟被选择 001：选择内部48M RC振荡器时钟 010：选择内部32K RC振荡器时钟 011：选择外部低速振荡器时钟 100：选择系统时钟 101：选择内部16M RC振荡器时钟 110：选择外部高速振荡器时钟

		111：依赖于PLLDV选择（CK_PLL / 2）或CK_PLL
23:18	PLLMF[5:0]	PLL倍频因子 软件写这些位包括RCU_CFG0的27位来确定PLL的倍频因子。 0000000~0000001：保留 0000010~0001110：（PLL 时钟源 x （PLLMF[6:0]+2）） 0001111~1111110:：（PLL 时钟源 x （PLLMF[6:0]+1）） 1111111：保留 **注意：**PLL 输出频率不能超过 64MHz。
17:16	PLLSEL	PLL时钟源选择 软件置1或清0来控制PLL时钟源 00：选择IRC16M二分频为PLL时钟源 01：选择HXTAL为PLL时钟源 1x：选择IRC48M为PLL时钟源
15:14	ADCPSC[1:0]	ADC时钟预分频选择 软件写两位包括RCU_CFG2的31位和30位来确定ADC时钟分频。 软件清0和置1。 0000：选择APB2时钟2分频 0001：选择APB2时钟4分频 0010：选择APB2时钟6分频 0011：选择APB2时钟8分频 0100：选择APB2时钟10分频 0101：选择APB2时钟12分频 0110：选择APB2时钟14分频 0111：选择APB2时钟16分频 1000：选择AHB时钟3分频 1001：选择AHB时钟5分频 1010：选择AHB时钟7分频 1011：选择AHB时钟9分频 1100：选择AHB时钟11分频 1101：选择AHB时钟13分频 1110：选择AHB时钟15分频 1111：选择AHB时钟17分频
13:11	APB2PSC[2:0]	APB2预分频选择 软件置1和清0来控制APB2时钟分频因子。 0xx：选择AHB时钟不分频 100：选择AHB时钟2分频 101：选择AHB时钟4分频 110：选择AHB时钟8分频 111：选择AHB时钟16分频
10:8	APB1PSC[2:0]	APB1预分频选择 软件设置和清除来控制APB1时钟分频因子。 0xx：选择AHB时钟不分频

100：选择AHB时钟2分频
101：选择AHB时钟4分频
110：选择AHB时钟8分频
111：选择AHB时钟16分频

7:4	AHBPSC[3:0]	AHB预分频选择 软件设置和清除来控制AHB时钟分频因子。 0xxx：选择CK_SYS系统时钟不分频 1000：选择CK_SYS系统时钟2分频 1001：选择CK_SYS系统时钟4分频 1010：选择CK_SYS系统时钟8分频 1011：选择CK_SYS系统时钟16分频 1100：选择CK_SYS系统时钟64分频 1101：选择CK_SYS系统时钟128分频 1110：选择CK_SYS系统时钟256分频 1111：选择CK_SYS系统时钟512分频
3:2	SCSS[1:0]	系统时钟转换状态 硬件设置和清除指示系统当前时钟源 00：选择CK_IRC16M作为CK_SYS系统时钟源 01：选择CK_HXTAL作为CK_SYS系统时钟源 10：选择CK_PLL作为CK_SYS系统时钟源 11：选择CK_IRC48M作为CK_SYS系统时钟源
1:0	SCS[1:0]	系统时钟转换 软件设置选择系统时钟源。由于CK_SYS的改变有固有的延迟，需要软件读SCSS位来确保转换是否结束。在从深度睡眠或待机模式中返回时，或作为系统时钟或PLL时钟源的HXTAL出现故障时，强制选择IRC16M作为系统时钟或PLL时钟。 00：选择IRC16M时钟作为CK_SYS系统时钟源 01：选择HXTAL时钟作为CK_SYS系统时钟源 10：选择PLL作为CK_SYS系统时钟源 11：选择CK_IRC48M作为CK_SYS系统时钟源

4.3.3. 中断寄存器（RCU_INT）

地址偏移：0x08
复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								CKMIC	LXTALCKMIC	IRC48MSTBIC	PLLSTBIC	HXTALSTBIC	IRC16MSTBIC	LXTALSTBIC	IRC32KSTBIC
								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	LXTALCKMIE	IRC48MSTBIE	PLLSTBIE	HXTALSTBIE	IRC16MSTBIE	LXTALSTBIE	IRC32KSTBIE	CKMIF	LXTALCKMIF	IRC48MSTBIF	PLLSTBIF	HXTALSTBIF	IRC16MSTBIF	LXTALSTBIF	IRC32KSTBIF

	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	CKMIC	HXTAL时钟阻塞中断清除 软件写1复位CKMIF标志位。 0：不复位CKMIF标志位 1：复位CKMIF标志位
22	LXTALCKMIC	LXTAL时钟阻塞中断清除 软件写1复位LXTALCKMIF标志位。 0：不复位LXTALCKMIF标志位 1：复位LXTALCKMIF标志位
21	IRC48MSTBIC	IRC48M时钟稳定中断清除 软件写1复位IRC48MSTBIF标志位。 0：不复位IRC48MSTBIF标志位 1：复位IRC48MSTBIF标志位
20	PLLSTBIC	PLL稳定中断清除 软件写1复位PLLSTBIF标志位。 0：不复位PLLSTBIF标志位 1：复位PLLSTBIF标志位
19	HXTALSTBIC	HXTAL时钟稳定中断清除 软件写1复位HXTALSTBIF标志位。 0：不复位HXTALSTBIF标志位 1：复位HXTALSTBIF标志位
18	IRC16MSTBIC	IRC16M时钟稳定中断清除 软件写1复位IRC16MSTBIF标志位。 0：不复位IRC16MSTBIF标志位 1：复位IRC16MSTBIF标志位
17	LXTALSTBIC	LXTAL时钟稳定中断清除 软件写1复位LXTALSTBIF标志位。 0：不复位LXTALSTBIF标志位 1：复位LXTALRDYF标志位
16	IRC32KSTBIC	IRC32K时钟稳定中断清除 软件写1复位IRC32KSTBIF标志位。 0：不复位IRC32KSTBIF标志位 1：复位IRC32KSTBIF标志位
15	保留	必须保持复位值。
14	LXTALCKMIE	LXTAL时钟阻塞中断使能 软件置1和清0来使能/禁止LXTAL时钟阻塞中断。

0：禁止LXTAL时钟阻塞中断
1：使能LXTAL时钟阻塞中断

13	IRC48MSTBIE	IRC48M时钟稳定中断使能 软件置1和清0来使能/禁止IRC48M时钟稳定中断。 0：禁止IRC48M时钟稳定中断 1：使能IRC48M时钟稳定中断
12	PLLSTBIE	PLL时钟稳定中断使能 软件置1和清0来使能/禁止PLL时钟稳定中断。 0：禁止PLL时钟稳定中断 1：使能PLL时钟稳定中断
11	HXTALSTBIE	HXTAL时钟稳定中断使能 软件置1和清0来使能/禁止HXTAL时钟稳定中断。 0：禁止HXTAL时钟稳定中断 1：使能HXTAL时钟稳定中断
10	IRC16MSTBIE	IRC16M时钟稳定中断使能 软件置1和清0来使能/禁止IRC16M时钟稳定中断。 0：禁止IRC16M时钟稳定中断 1：使能IRC16M时钟稳定中断
9	LXTALSTBIE	LXTAL时钟稳定中断使能 LXTAL时钟稳定中断使能/禁止控制。 0：禁止LXTAL时钟稳定中断 1：使能LXTAL时钟稳定中断
8	IRC32KSTBIE	IRC32K时钟稳定中断使能 IRC32K时钟稳定中断使能/禁止控制。 0：禁止IRC32K时钟稳定中断 1：使能IRC32K时钟稳定中断
7	CKMIF	HXTAL时钟阻塞中断标志位 当HXTAL时钟阻塞时硬件置1。 软件置位CKMIC时清除该位。 0：时钟运行正常 1：HXTAL时钟阻塞
6	LXTALCKMIF	LXTAL时钟阻塞中断标志位 当LXTAL时钟阻塞由硬件置1。 软件置位LXTALCKMIC时清除该位。 0：LXTAL时钟运行正常 1：LXTAL时钟阻塞
5	IRC48MSTBIF	IRC48M时钟稳定中断标志位 当IRC48M时钟稳定且IRC48MSTBIE位被置1时由硬件置1。 软件置IRC48MSTBIC=1时清除该位。 0：无IRC48M时钟稳定中断产生

1：IRC48M时钟稳定中断发生

4	PLLSTBIF	PLL时钟稳定中断标志位 当PLL时钟稳定且PLLSTBIE位被置1时由硬件置1。 软件置PLLSTBIC=1时清除该位。 0：无PLL时钟稳定中断产生 1：产生PLL时钟稳定中断
3	HXTALSTBIF	HXTAL时钟稳定中断标志位 当外部4 ~ 48 MHz晶体振荡器时钟稳定且HXTALSTBIE位被置1时由硬件置1。 软件置HXTALSTBIC=1时清除该位。 0：无HXTAL时钟稳定中断发生 1：发生HXTAL时钟稳定中断
2	IRC16MSTBIF	IRC16M时钟稳定中断标志位 当内部8 MHz RC振荡器时钟稳定且IRC16MSTBIE位被置1时由硬件置1。 软件置IRC16MSTBIC=1时清除该位。 0：无IRC16M时钟稳定中断产生 1：产生IRC16M时钟稳定中断
1	LXTALSTBIF	LXTAL时钟稳定中断标志位 当外部32.768KHz晶体振荡器时钟稳定且LXTALSTBIE为被置1时由硬件置1。 软件置LXTALSTBIC=1时清除该位。 0：无LXTAL时钟稳定中断发生 1：发生LXTAL时钟稳定中断
0	IRC32KSTBIF	IRC32K时钟稳定中断标志位 当内部32kHz RC振荡器时钟稳定且IRC32KSTBIE位被置1时由硬件置1。 软件置IRC32KSTBIC =1时清除该位。 0：无IRC32K时钟稳定中断产生 1：产生IRC32K时钟稳定中断

4.3.4. APB2 复位寄存器（RCU_APB2RST）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART0RST	保留.	SPI0RST	TIMER8RST	保留.	ADCRST	保留							CMPRST	SYSCFGRST
	rw		rw	rw		rw								rw	rw

位/位域	名称	描述

31:15	保留	必须保持复位值。
14	USART0RST	USART0复位 由软件置1或清0。 0：无复位 1：复位USART0
13	保留	必须保持复位值。
12	SPI0RST	SPI0复位 由软件置1或清0。 0：无复位 1：复位SPI0
11	TIMER8RST	TIMER8定时器复位 由软件置1或清0。 0：无复位 1：复位TIMER8定时器
10	保留	必须保持复位值。
9	ADCRST	ADC复位 由软件置1或清0。 0：无复位 1：复位ADC
8:2	保留	必须保持复位值。
1	CMPRST	比较器复位 由软件置1或清0。 0：无复位 1：复位比较器模块
0	SYSCFGRST	系统配置复位 由软件置1或清0。 0：无复位 1：复位系统配置和比较器模块

4.3.5. APB1 复位寄存器（RCU_APB1RST）

地址偏移：0x10

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	CTCRST	DACRST	PMURST	保留			I2C2RST	USBDRST	I2C1RST	I2C0RST	UART4RST	UART3RST	LPUARTRST	USART1RST	保留
	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

保留	SPI1RST	保留	WWDGTRST	SLCDRST	LPTIMERRST	TIMER11RST	保留	TIMER6RST	TIMER5RST	保留	TIMER2RST	TIMER1RST
	rw		rw	rw	rw	rw		rw	rw		rw	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30	CTCRST	CTC复位 由软件置1或清0。 0：无复位 1：复位CTC
29	DACRST	DAC复位 由软件置1或清0。 0：无复位 1：复位DAC
28	PMURST	电源控制复位 由软件置1或清0。 0：无复位 1：复位电源控制单元
27:25	保留	必须保持复位值。
24	I2C2RST	I2C2复位 由软件置1或清0。 0：无复位 1：复位I2C2
23	USBDRST	USBD复位 由软件置1或清0。 0：无复位 1：复位USBD
22	I2C1RST	I2C1复位 由软件置1或清0。 0：无复位 1：复位I2C1
21	I2C0RST	I2C0复位 由软件置1或清0。 0：无复位 1：复位I2C0
20	UART4RST	UART4复位 由软件置1或清0。 0：无复位 1：复位UART4

19	UART3RST	UART3复位 由软件置1或清0。 0：无复位 1：复位UART3
18	LPUARTRST	LPUART复位 由软件置1或清0。 0：无复位 1：复位LPUART
17	USART1RST	USART1复位 由软件置1或清0。 0：无复位 1：复位USART1
16:15	保留	必须保持复位值。
14	SPI1RST	SPI1复位 由软件置1或清0。 0：无复位 1：复位SPI1
13:12	保留	必须保持复位值。
11	WWDGTRST	窗口看门狗定时器复位 由软件置1或清0。 0：无复位 1：复位窗口看门狗定时器
10	SLCDRST	SLCD复位 由软件置1或清0。 0：无复位 1：复位SLCD
9	LPTIMERRST	LPTIMER定时器复位 由软件置1或清0。 0：无复位 1：复位LPTIMER定时器
8	TIMER11RST	TIMER11定时器复位 由软件置1或清0。 0：无复位 1：复位TIMER11定时器
7:6	保留	必须保持复位值。
5	TIMER6RST	TIMER6定时器复位 由软件置1或清0。 0：无复位 1：复位TIMER6定时器

4	TIMER5RST	TIMER5定时器复位 由软件置1或清0。 0：无复位 1：复位TIMER5定时器
3:2	保留	必须保持复位值。
1	TIMER2RST	TIMER2定时器复位 由软件置1或清0。 0：无复位 1：复位TIMER2定时器
0	TIMER1RST	TIMER1定时器复位 由软件置1或清0。 0：无复位 1：复位TIMER1定时器

4.3.6. AHB 使能寄存器（RCU_AHBEN）

地址偏移：0x14

复位值：0x0000 0014

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									PFEN	保留	PDEN	PCEN	PBEN	PAEN	保留
									rw		rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								SRAM1SPEN	CRCEN	保留	FMCSPEN	保留	SRAM0SPEN	保留	DMAEN
								rw	rw		rw		rw		rw

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PFEN	GPIOF时钟使能 由软件置1或清0。 0：GPIOF时钟关闭 1：GPIOF时钟开启
21	保留	必须保持复位值。
20	PDEN	GPIOD时钟使能 由软件置1或清0。 0：GPIOD时钟关闭 1：GPIOD时钟开启
19	PCEN	GPIOC时钟使能 由软件置1或清0。

		0：GPIOC时钟关闭 1：GPIOC时钟开启
18	PBEN	GPIOB时钟使能 由软件置1或清0。 0：GPIOB时钟关闭 1：GPIOB时钟开启
17	PAEN	GPIOA时钟使能 由软件置1或清0。 0：GPIOA时钟关闭 1：GPIOA时钟开启
16:8	保留	必须保持复位值。
7	SRAM1SPEN	SRAM1接口时钟使能 由软件置1或清0来开启/关闭在睡眠模式下的SRAM1时钟。 0：关闭睡眠模式下的SRAM1接口时钟 1：开启睡眠模式下的SRAM1接口时钟
6	CRCEN	CRC时钟使能 由软件置1或清0。 0：CRC时钟关闭 1：CRC时钟开启
5	保留	必须保持复位值。
4	FMCSPEN	FMC时钟使能 由软件置1或清0来开启/关闭在睡眠模式下的FMC时钟。 0：关闭睡眠模式下的FMC时钟 1：开启睡眠模式下的FMC时钟
3	保留	必须保持复位值。
2	SRAM0SPEN	SRAM0接口时钟使能 由软件置1或清0来开启/关闭在睡眠模式下的SRAM0时钟。 0：关闭睡眠模式下的SRAM0接口时钟 1：开启睡眠模式下的SRAM0接口时钟
1	保留	必须保持复位值。
0	DMAEN	DMA时钟使能 由软件置1或清0。 0：关闭DMA时钟 1：开启DMA时钟

4.3.7. APB2 使能寄存器（RCU_APB2EN）

地址偏移：0x18

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									DBGMCUEN	保留					
									rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART0EN	保留	SPI0EN	TIMER8EN	保留	ADCEN	保留							CMPEN	SYSCFGEN
	rw		rw	rw		rw								rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	DBGMCUEN	DBGMCU时钟使能 由软件置1或清0。 0：关闭DBGMCU时钟 1：开启DBGMCU时钟
21:15	保留	必须保持复位值。
14	USART0EN	USART0时钟使能 由软件置1或清0。 0：关闭USART0时钟 1：开启USART0时钟
13	保留	必须保持复位值。
12	SPI0EN	SPI0时钟使能 由软件置1或清0。 0：关闭SPI0时钟 1：开启SPI0时钟
11	TIMER8EN	TIMER8定时器时钟使能 由软件置1或清0。 0：关闭TIMER8定时器时钟 1：开启TIMER8定时器时钟
10	保留	必须保持复位值。
9	ADCEN	ADC接口时钟使能 由软件置1或清0。 0：关闭ADC接口时钟 1：开启ADC接口时钟
8:2	保留	必须保持复位值。
1	CMPEN	CMP模块时钟使能 由软件置1或清0。 0：关闭CMP模块时钟

		1：开启CMP模块时钟
0	SYSCFGEN	系统配置时钟使能 由软件置1或清0。 0：关闭系统配置时钟 1：开启系统配置时钟

4.3.8. APB1 使能寄存器（RCU_APB1EN）

地址偏移：0x1C

复位值：0x1000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BKPEN	CTCEN	DACEN	PMUEN	保留			I2C2EN	USBDEN	I2C1EN	I2C0EN	UART4EN	UART3EN	LPUARTEN	USART1EN	保留
rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SPI1EN	保留		WWDGTEN	SLCDEN	LPTIMEREN	TIMER11EN	保留		TIMER6EN	TIMER5EN	保留		TIMER2EN	TIMER1EN
	rw			rw	rw	rw	rw			rw	rw			rw	rw

位/位域	名称	描述
31	BKPEN	BKP（RTC）时钟使能 由软件置1或清0。 0：关闭BKT（RTC）时钟 1：开启BKP（RTC）时钟
30	CTCEN	CTC时钟使能 由软件置1或清0。 0：关闭CTC时钟 1：开启CTC时钟
29	DACEN	DAC时钟使能 由软件置1或清0。 0：关闭DAC时钟 1：开启DAC时钟
28	PMUEN	电源接口时钟使能 由软件置1或清0。 0：关闭电源接口时钟 1：开启电源接口时钟
27:25	保留	必须保持复位值。
24	I2C2EN	I2C2时钟使能 由软件置1或清0。

		0：关闭I2C2时钟 1：开启I2C2时钟
23	USBDEN	USBD时钟使能 由软件置1或清0。 0：关闭USBD时钟 1：开启USBD时钟
22	I2C1EN	I2C1时钟使能 由软件置1或清0。 0：关闭I2C1时钟 1：开启I2C1时钟
21	I2C0EN	I2C0时钟使能 由软件置1或清0。 0：关闭I2C0时钟 1：开启I2C0时钟
20	UART4EN	UART4时钟使能 由软件置1或清0。 0：关闭UART4时钟 1：开启UART4时钟
19	UART3EN	UART3时钟使能 由软件置1或清0。 0：关闭UART3时钟 1：开启UART3时钟
18	LPUARTEN	LPUART时钟使能 由软件置1或清0。 0：关闭LPUART时钟 1：开启LPUART时钟
17	USART1EN	USART1时钟使能 由软件置1或清0。 0：关闭USART1时钟 1：开启USART1时钟
16:15	保留	必须保持复位值。
14	SPI1EN	SPI1时钟使能 由软件置1或清0。 0：关闭SPI1时钟 1：开启SPI1时钟
13:12	保留	必须保持复位值。
11	WWDGTEN	窗口看门狗定时器时钟使能 由软件置1或清0。 0：关闭窗口看门狗定时器时钟 1：开启窗口看门狗定时器时钟

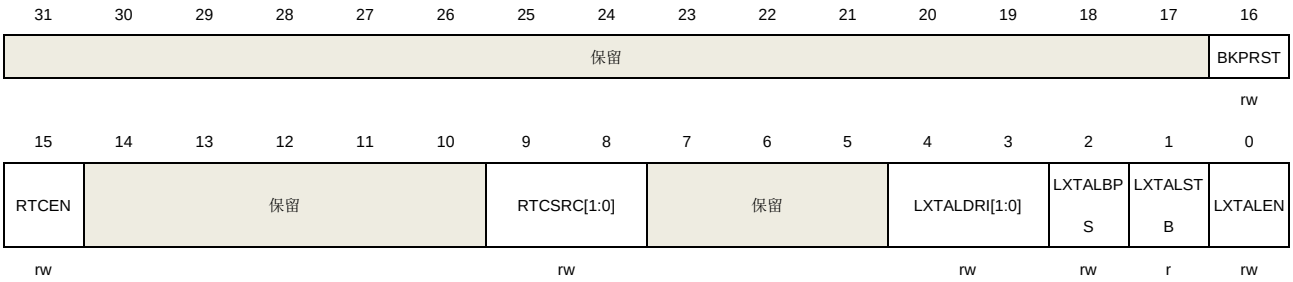
10	SLCDEN	SLCD时钟使能 由软件置1或清0。 0：关闭SLCD时钟 1：开启SLCD时钟
9	LPTIMEREN	LPTIMER时钟使能 由软件置1或清0。 0：关闭LPTIMER时钟 1：开启LPTIMER时钟
8	TIMER11EN	TIMER11定时器时钟使能 由软件置1或清0。 0：关闭TIMER11定时器时钟 1：开启TIMER11定时器时钟
7:6	保留	必须保持复位值。
5	TIMER6EN	TIMER6定时器时钟使能 由软件置1或清0。 0：关闭TIMER6定时器时钟 1：开启TIMER6定时器时钟
4	TIMER5EN	TIMER5定时器时钟使能 由软件置1或清0。 0：关闭TIMER5定时器时钟 1：开启TIMER5定时器时钟
3:2	保留	必须保持复位值。
1	TIMER2EN	TIMER2定时器时钟使能 由软件置1或清0。 0：关闭TIMER2定时器时钟 1：开启TIMER2定时器时钟
0	TIMER1EN	TIMER1定时器时钟使能 由软件置1或清0。 0：关闭TIMER1定时器时钟 1：开启TIMER1定时器时钟

4.3.9. 备份域控制寄存器（RCU_BDCTL）

地址偏移：0x20
复位值：0x0000 0018，由备份域复位电路复位

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

注意：备份域控制寄存器（BDCTL）的LXTALEN, LXTALBPS, RTCSRC和RTCEN位仅在备份域复位后才清0。只有在电源控制寄存器（PMU_CTL）中的BKPWEN位置1后才能对这些位进行改动。



位/位域	名称	描述
31:17	保留	必须保持复位值。
16	BKPRST	备份域复位 由软件置1或清0。 0：无复位 1：复位备份域
15	RTCEN	RTC时钟使能 由软件置1或清0。 0：关闭RTC时钟 1：开启RTC时钟
14:10	保留	必须保持复位值。
9:8	RTCSRC[1:0]	RTC / SLCD时钟入口选择 软件置位或清除来控制RTC / SLCD时钟源。 00：没有时钟 01：选择LXTAL时钟作为RTC / SLCD时钟源 10：选择IRC32K时钟作为RTC / SLCD时钟源 11：选择HXTAL时钟32分频作为RTC / SLCD时钟源
7:5	保留	必须保持复位值。
4:3	LXTALDRI[1:0]	LXTAL驱动能力 软件置位或清除。当复位备份域时，会重装载缺省值。 00：弱驱动能力 01：中低驱动能力 10：中高驱动能力 11：强驱动能力 （复位后的缺省值） **注意：**LXTALDRI在旁路模式下无效
2	LXTALBPS	LXTAL旁路模式使能 软件置1和清0。 0：禁止LXTAL旁路模式 1：使能LXTAL旁路模式
1	LXTALSTB	外部低速振荡器稳定状态位 硬件置1来指示LXTAL输出时钟是否稳定待用。 0：LXTAL未稳定

		1：LXTAL已稳定
0	LXTALEN	LXTAL使能 软件置1和清0。 0：关闭LXTAL 1：开启LXTAL

4.3.10. 复位源/时钟寄存器（RCU_RSTSCK）

地址偏移：0x24

复位值：0x0C80 0000，除复位标志外由系统复位清除，复位标志只能由电源复位清除。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPRSTF	WWDGTRSTF	FWDGTRSTF	SWRSTF	PORRSTF	EPRSTF	保留	RSTFC	V11RSTF	保留						
r	r	r	r	r	r		rw	r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														IRC32KSTB	IRC32KEN
														r	rw

位/位域	名称	描述
31	LPRSTF	低功耗复位标志位 深度睡眠/待机复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0：无低功耗管理复位发生 1：发生低功耗管理复位
30	WWDGTRSTF	窗口看门狗定时器复位标志位 窗口看门狗定时器复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0：无窗口看门狗定时器复位发生 1：发生窗口看门狗定时器复位
29	FWDGTRSTF	独立看门狗定时器复位标志位 独立看门狗复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0：无独立看门狗定时器复位发生 1：发生独立看门狗定时器复位
28	SWRSTF	软件复位标志位 软件复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0：无软件复位发生 1：发生软件复位

27	PORRSTF	电源复位标志位 电源复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0：无电源复位发生 1：发生电源复位
26	EPRSTF	外部引脚复位标志位 当有外部引脚复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0：无外部引脚复位发生 1：发生外部引脚复位
25	保留	必须保持复位值。.
24	RSTFC	清除复位标志位 由软件置1来清除所有复位标志位。 0：无作用 1：清除复位标志位
23	V11RSTF	1.1V域电源复位标志位 当有1.1V域电源复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0：无1.1V域电源复位发生 1：发生1.1V域电源复位
22:2	保留	必须保持复位值。.
1	IRC32KSTB	IRC32K时钟稳定状态位 该位由硬件置1指示IRC32K输出时钟是否稳定待用。 0：IRC32K时钟未稳定 1：IRC32K时钟已稳定
0	IRC32KEN	IRC32K时钟使能 软件置1和清0。 0：关闭IRC32K时钟 1：开启IRC32K时钟

4.3.11. AHB 复位寄存器（RCU_AHBRST）

地址偏移：0x28

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									PFRST	保留	PDRST	PCRST	PBRST	PARST	保留
									rw		rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									CRCRST	保留					

rw

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PFRST	GPIOF复位 由软件置1或清0。 0：无作用 1：复位GPIOF口
21	保留	必须保持复位值。
20	PDRST	GPIOD复位 由软件置1或清0。 0：无作用 1：复位GPIOD口
19	PCRST	GPIOC复位 由软件置1或清0。 0：无作用 1：复位GPIOC口
18	PBRST	GPIOB复位 由软件置1或清0。 0：无作用 1：复位GPIOB口
17	PARST	GPIOA复位 由软件置1或清0。 0：无作用 1：复位GPIOA口
16:7	保留	必须保持复位值。
6	CRCRST	CRC复位 由软件置1或清0。 0：无作用 1：复位CRC
5:0	保留	必须保持复位值。

4.3.12. 配置寄存器 1（RCU_CFG1）

地址偏移：0x2C

复位值：0x0000 0007

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												PREDV[3:0]			
												rw			

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	PREDV[3:0]	PLL输入源分频因子 由软件置1或清0。这些位仅能在PLL关闭时改写。时钟分频因子为 （PREDV + 1）。 0000：PLL的输入，不分频 0001：PLL的输入2分频 0010：PLL的输入3分频 0011：PLL的输入4分频 0100：PLL的输入5分频 0101：PLL的输入6分频 0110：PLL的输入7分频 0111：PLL的输入8分频 1000：PLL的输入9分频 1001：PLL的输入10分频 1010：PLL的输入11分频 1011：PLL的输入12分频 1100：PLL的输入13分频 1101：PLL的输入14分频 1110：PLL的输入15分频 1111：PLL的输入16分频

4.3.13. 配置寄存器 2（RCU_CFG2）

地址偏移：0x30

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCPSC[3:2]		保留										IRC16MDIV		USART1SEL[1:0]	
rw												rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		USBDSEL	LPUART[1:0]		LPTIMERSEL[1:0]		ADCSEL	I2C2SEL[1:0]		I2C1SEL[1:0]		I2C0SEL[1:0]		USART0SEL[1:0]	
		rw	rw		rw		rw	rw		rw		rw		rw	

位/位域	名称	描述
31:30	ADCPSC[3:2]	ADCPSC的位3和位2

		参考RCU_CFG0的位15:14
29:21	保留	必须保持复位值。
20:18	IRC16MDIV	CK_IRC16M时钟分频选择作为CK_IRC16MDIV时钟 0xx：选择 CK_IRC16M 作为 CK_IRC16MDIV 时钟 100：选择 CK_IRC16M/2 作为 CK_IRC16MDIV 时钟 101：选择 CK_IRC16M/4 作为 CK_IRC16MDIV 时钟 110：选择 CK_IRC16M/8 作为 CK_IRC16MDIV 时钟 111：选择CK_IRC16M/16作为CK_IRC16MDIV时钟
17:16	USART1SEL[1:0]	USART1时钟源选择 由软件置1或清0。 00：USART1时钟选择CK_APB1 01：USART1时钟选择CK_SYS 10：USART1钟选择CK_LXTAL 11：USART1钟选择CK_IRC16M
15:14	保留	必须保持复位值。
13	USBDSEL	USBD时钟源选择 由软件置1或清0。 0：USBD时钟选择IRC48M 1：USBD时钟选择CK_PLL
12:11	LPUARTSEL[1:0]	LPUART时钟源选择 由软件置1或清0。 00：LPUART时钟选择CK_APB1 01：LPUART时钟选择CK_SYS 10：LPUART钟选择CK_LXTAL 11：LPUART钟选择CK_IRC16MDIV
10:9	LPTIMERSEL[1:0]	LPTIMER时钟源选择 由软件置1或清0。 00：LPTIMER时钟选择CK_APB2 01：LPTIMER时钟选择CK_IRC32K 10：LPTIMER钟选择CK_LXTAL 11：LPTIMER钟选择CK_IRC16MDIV
8	ADCSEL	ADC时钟源选择 由软件置1或清0。 0：ADC时钟源选择IRC16M时钟 1：ADC时钟源选择由APB2时钟经2、4、6、8、10、12、14、16分频或由AHB时钟经3、5、7、9、11、13、15、17分频
7:6	I2C2SEL[1:0]	CK_I2C2时钟源选择 00：CK_I2C2时钟选择CK_APB1 01：CK_I2C2时钟选择CK_SYS 10/11：CK_I2C2时钟选择CK_IRC16MDIV

5:4	I2C1SEL[1:0]	CK_I2C1时钟源选择 00：CK_I2C1时钟选择CK_APB1 01：CK_I2C1时钟选择CK_SYS 10/11：CK_I2C1时钟选择CK_IRC16MDIV
3:2	I2C0SEL[1:0]	CK_I2C0时钟源选择 00：CK_I2C0时钟选择CK_APB1 01：CK_I2C0时钟选择CK_SYS 10/11：CK_I2C0时钟选择CK_IRC16MDIV
1:0	USART0SEL[1:0]	USART0时钟源选择 由软件置1或清0。 00：USART0时钟选择APB2时钟 01：USART0时钟选择CK_SYS 10：USART0时钟选择LXTAL时钟 11：USART0时钟选择IRC16MDIV时钟

4.3.14. AHB2 使能寄存器（RCU_AHB2EN）

地址偏移：0x34

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												TRNGEN	保留	CAUEN	保留
												rw		rw	

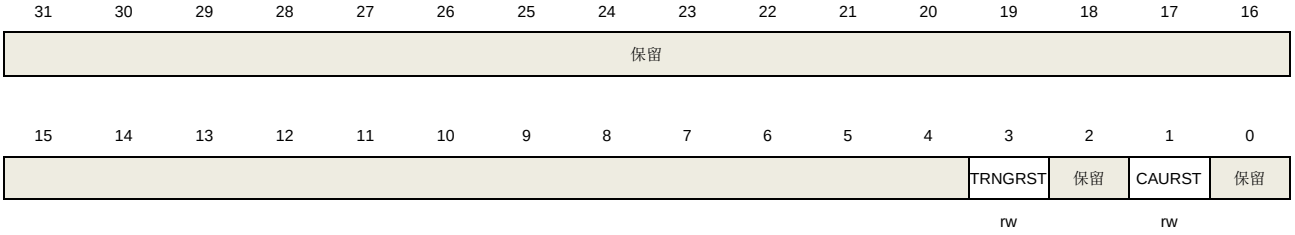
位/位域	名称	描述
31:4	保留	必须保持复位值。
3	TRNGEN	TRNG时钟使能 由软件置1或清0。 0：TRNG时钟关闭 1：TRNG时钟开启
2	保留	必须保持复位值。
1	CAUEN	CAU时钟使能 由软件置1或清0。 0：CAU时钟关闭 1：CAU时钟开启
0	保留	必须保持复位值。

4.3.15. AHB2 复位寄存器（RCU_AHB2RST）

地址偏移：0x38

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



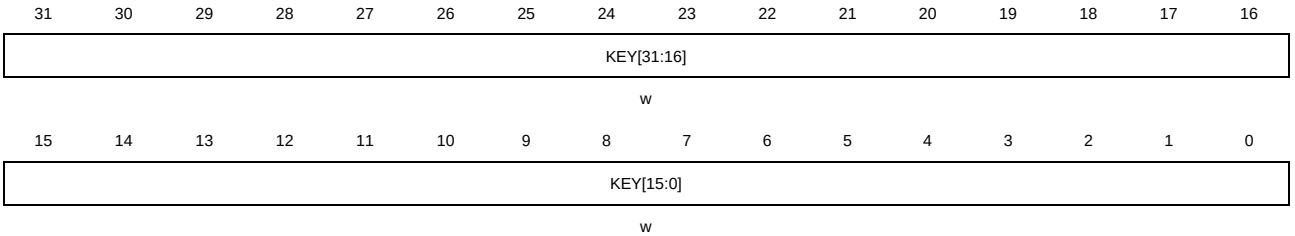
位/位域	名称	描述
31:4	保留	必须保持复位值。
3	TRNGRST	TRNG复位 由软件置1或清0。 0：无作用 1：复位TRNG
2	保留	必须保持复位值。
1	CAURST	CAU复位 由软件置1或清0。 0：无作用 1：复位CAU
0	保留	必须保持复位值。

4.3.16. 电源解锁寄存器（RCU_VKEY）

地址偏移：0x100

复位值：0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



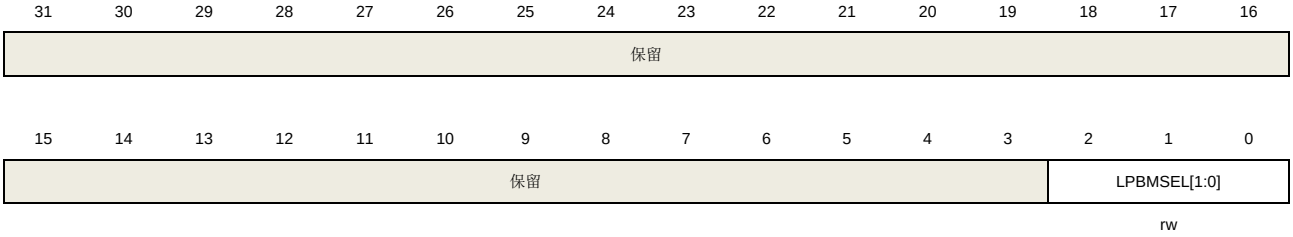
位/位域	名称	描述
31:0	KEY[31:0]	RCU_DSV寄存器解锁 这些位只能被软件写，读的话全是0。只有在向RCU_VKEY寄存器写0x1A2B3C4D后，RCU_DSV, RCU_CPC, RCU_LPLDO, RCU_LPB寄存器才能被写。

4.3.17. 低功耗模式寄存器（RCU_LPB）

地址偏移：0x12C

复位值：0x0000 0007

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。



位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	LPBMSEL[1:0]	低功耗模式选择信号。控制采样保持电路保持相的时间长度。 011：保持相的时间长度为典型值3.2ms，32个时钟周期 010：保持相的时间长度为典型值6.4ms，64个时钟周期 001：保持相的时间长度为典型值12.8ms，128个时钟周期 000：保持相的时间长度为典型值25.6ms，256个时钟周期 111：保持相的时间长度为典型值51.2ms，512个时钟周期 110：保持相的时间长度为典型值102.4ms，1024个时钟周期 101：保持相的时间长度为典型值204.8ms，2048个时钟周期 100：保持相的时间长度为典型值204.8ms，2048个时钟周期

5. 时钟校准控制器（CTC）

5.1. 简介

时钟校准控制器（CTC）采用硬件的方式，自动校准内部48MHz RC晶振（IRC48M）。当USBD / USBFS模块使用IRC48M时钟作为时钟源时，必须要求IRC48M时钟频率在48MHz±500ppm的范围内，但是没有经过校准的内部晶振无法满足这么高的精度。CTC模块基于外部高精度的参考信号源来校准IRC48M的时钟频率，通过自动的或手动的调整校准值，以得到一个精准的IRC48M时钟。

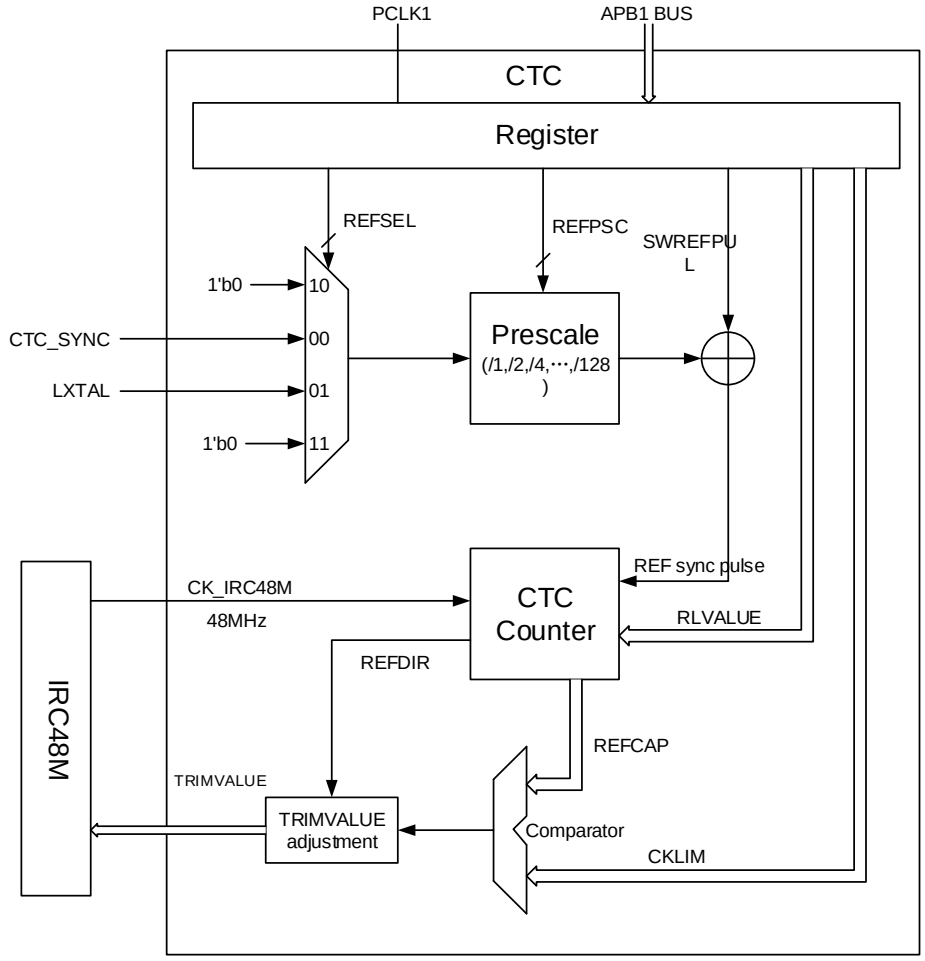
5.2. 主要特征

- 两个外部参考信号源：GPIO，LXTAL时钟；
- 提供软件参考同步脉冲；
- 硬件自动校准，无需软件操作；
- 具有参考信号源捕获和重载功能的16 bits校准计数器；
- 用于频率评估和自动校准的8 bits时钟校准基值；
- 标志位和中断，用于指示时钟校准的状态：校准成功状态（CKOKIF），警告状态（CKWARNIF）和错误状态（ERRIF）。

5.3. 功能说明

CTC模块的内部结构图如*图5-1. CTC简介*。

图 5-1. CTC 简介



5.3.1. REF 同步脉冲发生器

首先，通过设置CTC_CTL1寄存器（CTC控制寄存器1）中的REFSEL位来选择参考信号源：GPIO或者LXTAL时钟输出。

然后，可以通过设置CTC_CTL1寄存器中的REFPOL位来配置参考信号源同步时的信号极性，通过设置CTC_CTL1寄存器中的REFPSC位来产生一个合适的同步时钟频率信号。

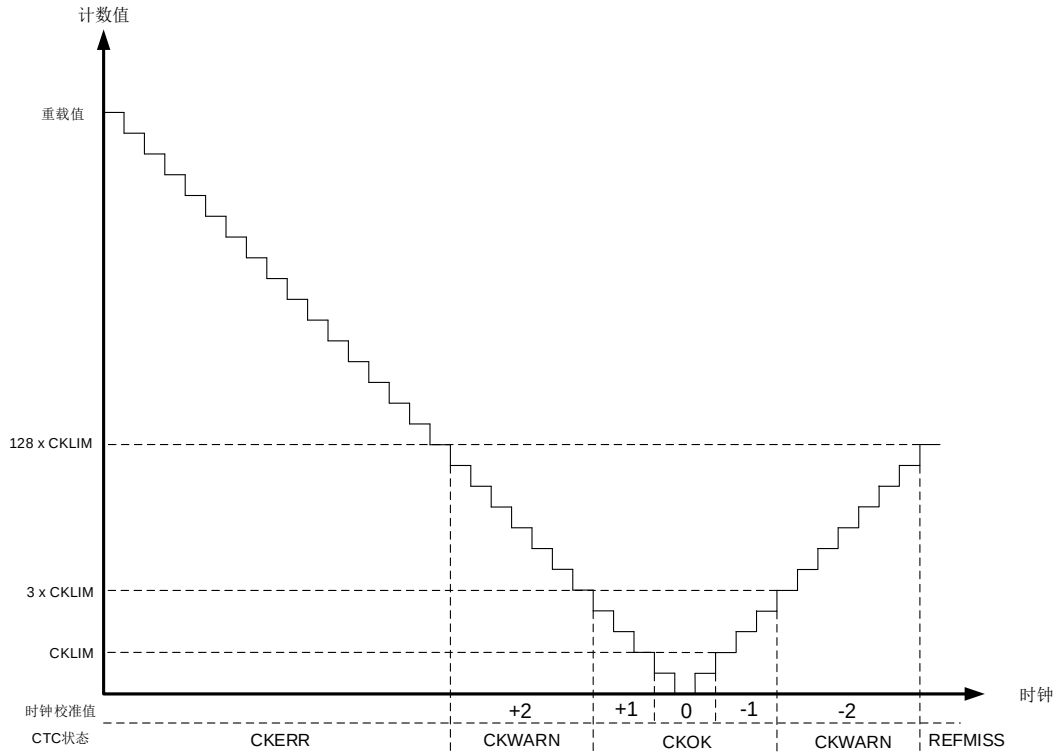
如果需要使用软件参考脉冲信号，则需要设置CTC_CTL0寄存器（CTC控制寄存器0）中的SWREFPUL位为1。软件参考脉冲信号与外部参考脉冲信号最后进行逻辑或操作。

5.3.2. CTC 校准计数器

CTC时钟校准计数器由CK_IRC48M提供时钟。在置位CTC_CTL0寄存器中的CNTEN位后，当检测到第一个REF同步脉冲信号，计数器开始从RLVALUE值（RLVALUE在CTC_CTL1寄存器中定义）向下计数。每次检测到REF同步脉冲信号时，计数器重载RLVALUE值，同时重新开始向下计数。如果始终检测不到REF同步脉冲信号，计数器会向下计数到零，然后再向上计数到128 x CKLIM（CKLIM在CTC_CTL1中定义），最后停止，直到检测到下一个REF同步脉冲信号。一旦检测到REF同步脉冲信号，当前CTC校准计数器的计数值被捕获存入CTC_STAT（CTC

状态寄存器）中的REFCAP位，同时，当前计数器的计数方向被存入CTC_STAT中的REFDIR位。详细内容如*图5-2. CTC校准计数器*所示。

图 5-2. CTC 校准计数器



5.3.3. 频率评估和自动校准过程

当REF同步脉冲信号出现时，时钟频率评估功能开始执行。如果REF同步脉冲信号出现在计数器向下计数的过程中，说明当前时钟频率比期望时钟频率（频率为48M）慢，需要增大CTC_CTL0中的TRIMVALUE值（时钟校准值）。如果REF同步脉冲信号出现在计数器向上计数的过程中，说明当前时钟频率比期望时钟频率快，需要减小TRIMVALUE值。CTC_STAT中的CKOKIF位，CKWARNIF位，CKERR位和REFMISS位反映了频率评估的状态。

如果CTC_CTL0中的AUTOTRIM（硬件自动校准模式）位置1，硬件自动校准模式使能。在这个模式中，如果REF同步脉冲信号出现在计数器向下计数的过程中，说明当前时钟频率比期望时钟频率慢，CTC_CTL0中的TRIMVALUE值会自动增大，来提高当前的时钟频率。反之，如果REF同步脉冲信号出现在计数器向上计数的过程中，说明当前时钟频率比期望时钟频率快，TRIMVALUE值会自动减小，从而减小当前的时钟频率。

- Counter < CKLIM时，检测到REF同步脉冲信号；

  CTC_STAT中的CKOKIF位（时钟校准成功标志位）被置位，同时，如果CTC_CTL0中的CKOKIE位（时钟校准完成中断使能位）置1，将会产生一个中断。
  如果CTC_CTL0中的AUTOTRIM置1，CTC_CTL0中的TRIMVALUE值不变。

- CKLIM ≤ Counter < 3 x CKLIM时，检测到REF同步脉冲信号；

  CTC_STAT中的CKOKIF位被置位，同时，如果CTC_CTL0中的CKOKIE位置1，将会产生

一个中断。
如果CTC_CTL0中的AUTOTRIM位置1，在计数器向下计数过程中，CTC_CTL0中的TRIMVALUE值将加1，而在向上计数过程中将减1。

- 3 x CKLIM ≤ Counter < 128 x CKLIM时，检测到REF同步脉冲信号；

  CTC_STAT中的CKWARNIF位（时钟校准警告中断位）被置位，同时，如果CTC_CTL0中的CKWARNIE位（时钟校准警告中断使能位）置1，将会产生一个中断。
  如果CTC_CTL0中的AUTOTRIM位置1，在计数器向下计数过程中，CTC_CTL0中的TRIMVALUE值将加2，而在向上计数过程中将减2。

- Counter ≥ 128 x CKLIM，计数器在向下计数过程中，检测到REF同步脉冲信号；

  CTC_STAT中的CKERR位（时钟校准错误位）被置位，同时，如果CTC_CTL0中的ERRIE位（错误中断使能位）置1，将会产生一个中断。
  CTC_CTL0中的TRIMVALUE值不变。

- Counter = 128 x CKLIM，计数器在向上计数过程中；

  CTC_STAT中的REFMISS位（REF同步脉冲丢失位）被置位，同时，如果CTC_CTL0中的ERRIE位置1，将会产生一个中断。
  CTC_CTL0中的TRIMVALUE值不变。

如果CTC_CTL0中的TRIMVALUE的校准值大于127，将会发生上溢事件，同时，若TRIMVALUE的校准值小于0，将会发生下溢事件。TRIMVALUE的取值范围为0~127（上溢事件发生时，TRIMVALUE值为127；下溢事件发生时，TRIMVALUE值为0）。然后，CTC_STAT中的TRIMERR位（校准值错误位）将会被置位，如果CTC_CTL0中的ERRIE位置1，将会产生一个中断。

5.3.4. 软件编程指南

CTC_CTL1中RLVALUE位和CKLIM位是时钟频率评估和硬件自动校准的关键。它们的数值由期望时钟的频率（IRC48M：48 MHz）和REF同步脉冲信号的频率计算得到。理想状态是REF同步脉冲信号在CTC计数器计数到零时出现，所以RLVALUE的值为：

$$RLVALUE=(F_{clock} \div F_{REF})-1 \quad (式5\text{-}1)$$

CKLIM的值由用户根据时钟的精度来设置，一般建议设为步长的一半，所以CKLIM的值为：

$$CKLIM=(F_{clock} \div F_{REF}) \times 0.12\% \div 2 \quad (式5\text{-}2)$$

典型的步长值是0.12%，F_{clock}是期望时钟的频率（IRC48M），F_{REF}是REF同步脉冲信号的频率。

5.4. CTC 寄存器

CTC 基地址：0x4000 C800

5.4.1. 控制寄存器 0 (CTC_CTL0)

地址偏移：0x00
复位值：0x0000 4000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRIMVALUE[6:0]							SWREFPUL	AUTOTRIM	CNTEN	保留	EREFIE	ERRIE	CKWARNIE	CKOKIE
	rw							w	rw	rw		rw	rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:8	TRIMVALUE[6:0]	IRC48M 校准值 当 CTC_CTL0 中的 AUTOTRIM 值为 0 时，该位由软件置位和清除，该模式用于软件校准过程。 当 CTC_CTL0 中的 AUTOTRIM 值为 1 时，该位只读，由硬件自动修改，该模式用于硬件校准过程。 TRIMVALUE 的中间值是 64，当 TRIMVALUE 值加 1 时，IRC48M 时钟频率增加大约 57KHz。当 TRIMVALUE 值减 1 时，IRC48M 时钟频率的减少大约 57KHz。
7	SWREFPUL	软件生成同步参考信号脉冲 该位由软件置位，并为 CTC 计数器提供一个同步参考脉冲信号。该位由硬件自动清除，读操作时返回 0。 0：没有影响 1：软件产生一个同步参考脉冲信号
6	AUTOTRIM	硬件自动校准模式 该位由软件置位或清除。当该位置 1 时，硬件自动校准模式使能，通过硬件不断的自动修改 CTC_CTL0 中的 TRIMVALUE 值，直到 IRC48M 的时钟频率达到 48MHz。 0：禁止硬件自动校准模式 1：使能硬件自动校准模式
5	CNTEN	CTC 计数器使能 该位由软件置位或清除，用于使能或禁止 CTC 计数器。当该位置 1 时，不能修改 CTC_CTL1 的值。 0：禁止 CTC 计数器

		1：使能 CTC 计数器
4	保留	必须保持复位值。
3	EREFIE	期望参考信号中断使能 0：禁止期望参考信号产生中断 1：使能期望参考信号产生中断
2	ERRIE	错误中断使能 0：禁止错误中断 1：使能错误中断
1	CKWARNIE	时钟校准警告中断使能 0：禁止时钟校准警告中断 1：使能时钟校准警告中断
0	CKOKIE	时钟校准完成中断使能 0：禁止时钟校准完成中断 1：使能时钟校准完成中断

5.4.2. 控制寄存器 1 (CTC_CTL1)

地址偏移：0x04

复位值：0x2022 BB7F

该寄存器只能按字（32 位）访问。当 CNTEN 为 1 时，不能修改该寄存器的值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REFPOL	保留	REFSEL[1:0]		保留	REFPSC[2:0]			CKLIM[7:0]							
rw		rw			rw			rw							
5	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RLVALUE[15:0]															
rw															

位/位域	名称	描述
31	REFPOL	参考信号源极性 该位由软件置位或清除，用于选择参考信号源的同步极性 0：选择上升沿 1：选择下降沿
30	保留	必须保持复位值。
29:28	REFSEL[1:0]	参考信号源选择 该位由软件置位或清除，用于选择参考信号源 00：选择 GPIO 输入信号 01：选择 LXTAL 时钟 10：保留，选择 0

		11：保留，选择 0
27	保留	必须保持复位值。
26:24	REFPSC[2:0]	参考信号源预分频 该位由软件置位或清除 000：参考信号不分频 001：参考信号 2 分频 010：参考信号 4 分频 011：参考信号 8 分频 100：参考信号 16 分频 101：参考信号 32 分频 110：参考信号 64 分频 111：参考信号 128 分频
23:16	CKLIM[7:0]	时钟校准时基限值 该位由软件置位或清除，用于定义时钟校准时基限值。该位用于频率评估和自动校准过程，详细情况请参考"*频率评估和自动校准过程*"。
15:0	RLVALUE[15:0]	CTC 计数器重载值 该位由软件置位或清除，用于定义 CTC 计数器的重载值，当检测到一个同步参考脉冲时，该值将重载到 CTC 校准计数器中。

5.4.3. 状态寄存器 (CTC_STAT)

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REFCAP[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REFDIR	保留				TRIM ERR	REF MISS	CKERR	保留				EREFIF	ERRIF	CKWARN IF	CKOK IF
r					r	r	r					r	r	r	r

位/位域	名称	描述
31:16	REFCAP[15:0]	CTC 计数器捕获值 当检测到一个同步参考脉冲信号时，CTC 校准计数器中的计数值被存入到 REFCAP 位中。
15	REFDIR	CTC 校准时钟计数方向 当检测到一个同步参考脉冲信号时，CTC 校准计数器的计数方向被存入 REFDIR 位中。 0：向上计数

		1：向下计数
14:11	保留	必须保持复位值。
10	TRIMERR	校准值错误位 当 CTC_CTL0 中的 TRIMVALUE 值发生上溢或下溢时，该位由硬件置位。若 CTC_CTL0 中的 ERRIE 位置 1，则会产生一个中断。通过写 1 到 CTC_INTC 中的 ERRIC 位，可以将 TRIMERR 位清零。 0：无校准值错误发生 1：发生校准值错误
9	REFMISS	同步参考脉冲信号丢失 当同步参考脉冲信号丢失时，该位由硬件置位。当 CTC 校准计数器在增计数的过程中计数到 128 x CKLIM 都没有检测到同步参考脉冲信号时，REFMISS 位置位。说明当前时钟太快，无法校准到期望频率值，或者有其他错误产生。通过写 1 到 CTC_INTC 中的 ERRIC 位，可以将 REFMISS 位清零。 0：无同步参考脉冲信号丢失 1：同步参考脉冲信号丢失
8	CKERR	时钟校准错误位 当时钟校准错误产生时，该位由硬件置位。当 CTC 校准计数器计数值在减计数的过程中大于或等于 128 x CKLIM，并检测到同步参考脉冲信号时，CKERR 置位，说明当前时钟太慢，无法校准到期望频率值。当 CTC_CTL0 中的 ERRIE 置 1 时，产生一个中断。通过写 1 到 CTC_INTC 中的 ERRIC 位，可以将 CKERR 位清零。 0：无时钟校准错误发生 1：发生时钟校准错误
7:4	保留	必须保持复位值。
3	EREFIF	期望参考中断标志位 当 CTC 校准时钟计数器计数到 0 时，该位由硬件置位。当 CTC_CTL0 中的 EREFIE 置 1 时，产生一个中断。通过写 1 到 CTC_INTC 中的 EREFIC 位，可以将 EREFIF 位清零。 0：无期望参考信号产生 1：期望参考信号产生
2	ERRIF	错误中断标志位 当发生一个错误时，该位由硬件置位。只要有 TRIMERR，REFMISS 或者 CKERR 错误发生时，该位置位。当 CTC_CTL0 中的 ERRIE 置位时，产生一个中断。通过写 1 到 CTC_INTC 中的 ERRIC 位，可以将 ERRIF 位清零。 0：无错误发生 1：发生错误
1	CKWARNIF	时钟校准警告中断标志位 当时钟校准警告产生时，该位由硬件置位。当 CTC 校准计数器计数值大于或等于 3 x CKLIM 且小于 128 x CKLIM，并检测到同步参考脉冲信号时，CKWARNIF 置位。这说明当前时钟频率太慢或者太快，但可以通过校准达到期望频率值。当时钟校准警告产生时，TRIMVALUE 值加 2 或者减 2。当 CTC_CTL0 中的 CKWARNIE

		置 1 时，产生一个中断。通过写 1 到 CTC_INTC 中的 CKWARNIC 位，可以将 CKWARNIF 位清零。 0：无时钟校准警告发生 1：有时钟校准警告发生
0	CKOKIF	时钟校准成功中断标志位 当时钟校准成功时，该位由硬件置位。若在 CTC 校准计数器计数值小于 3 x CKLIM 时，检测当同步参考脉冲信号，CKOKIF 置位。说明当前时钟频率正常，可以使用，不需要通过 TRIMVALUE 值进行时钟校准。当 CTC_CTL0 中的 CKOKIE 置 1 时，产生一个中断。通过写 1 到 CTC_INTC 中的 CKOKIC 位，可以将 CKOKIF 位清零。 0：时钟校准未成功 1：时钟校准成功

5.4.4. 中断清除寄存器 (CTC_INTC)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												EREFIC	ERRIC	CKWARNIC	CKOKIC
												w	w	w	w

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	EREFIC	EREFIF 中断清除位 该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 EREFIF 位，写 0 没影响。
2	ERRIC	ERRIF 中断清除位 该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 ERRIF 位，TRIMERR 位，REFMISS 位和 CKERR 位，写 0 没影响。
1	CKWARNIC	CKWARNIF 中断清除位 该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 CKWARNIF 位，写 0 没影响。
0	CKOKIC	CKOKIF 中断清除位 该位只能由软件写，读操作返回 0。写 1 可以清除 CTC_STAT 中的 CKOKIF 位，写 0 没影响。

6. 中断/事件控制器（EXTI）

6.1. 简介

Cortex®-M23集成了嵌套式矢量型中断控制器（Nested Vectored Interrupt Controller（NVIC））来实现高效的异常和中断处理。NVIC实现了低延迟的异常和中断处理，以及电源管理控制。它和内核是紧密耦合的。更多关于NVIC的说明请参考《Cortex®-M23技术参考手册》。

EXTI（中断/事件控制器）包括30个相互独立的边沿检测电路并且能够向处理器内核产生中断请求或唤醒事件。EXTI有三种触发类型：上升沿触发、下降沿触发和任意沿触发。EXTI中的每一个边沿检测电路都可以独立配置和屏蔽。

6.2. 主要特征

- Cortex®-M23系统异常；
- 69种可屏蔽的外设中断；
- 2位中断优先级配置位—4个中断优先等级；
- 高效的中断处理；
- 支持异常抢占和咬尾中断；
- 将系统从省电模式唤醒；
- EXTI中有30个相互独立的边沿检测电路；
- 3种触发类型：上升沿触发，下降沿触发和任意沿触发；
- 软件中断或事件触发；
- 可配置的触发源。

6.3. 中断功能描述

ARM® Cortex®-M23处理器和嵌套式矢量型中断控制器（NVIC）在处理（Handler）模式下对所有异常进行优先级区分以及处理。当异常发生时，系统自动将当前处理器工作状态压栈，在执行完中断服务子程序（ISR）后自动将其出栈。

取向量是和当前工作态压栈并行进行的，从而提高了中断入口效率。处理器支持咬尾中断，可实现背靠背中断，大大削减了反复切换工作态所带来的开销。*表6-1. Cortex®-M23中的NVIC异常类型*和*表6-2. 中断向量表*列出了所有的异常类型。

表 6-1. Cortex®-M23 中的 NVIC 异常类型

异常类型	向量编号	优先级（a）	向量地址	描述
-	0	-	0x0000_0000	保留
复位	1	-3	0x0000_0004	复位
NMI	2	-2	0x0000_0008	不可屏蔽中断
硬件故障	3	-1	0x0000_000C	各种硬件级别的故障

异常类型	向量编号	优先级（a）	向量地址	描述
-	4-10	-	0x0000_0010 - 0x0000_002B	保留
SVCall 服务调用	11	可编程设置	0x0000_002C	通过 SWI 指令实现系统服务调用
-	12-13	-	0x0000_0030 - 0x0000_0034	保留
PendSV 挂起服务	14	可编程设置	0x0000_0038	可挂起的系统服务请求
系统节拍	15	可编程设置	0x0000_003C	系统节拍定时器

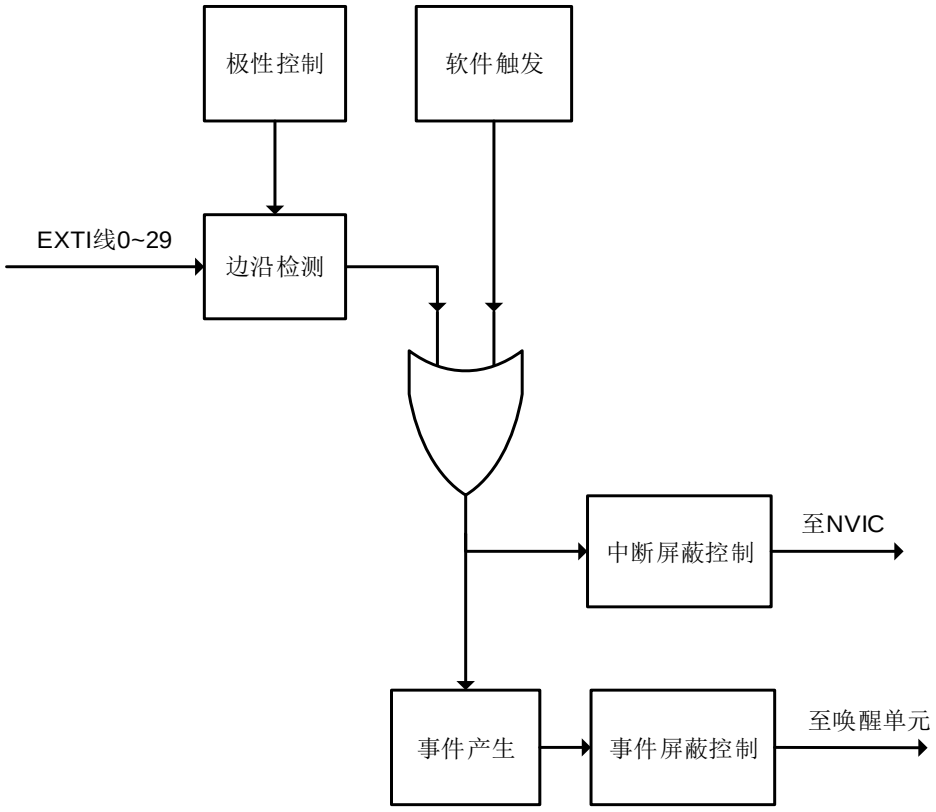
表 6-2. 中断向量表

中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 LVD 中断	0x0000_0044
IRQ 2	18	连接到EXTI线的RTC侵入和时间戳中断	0x0000_0048
IRQ 3	19	连接到EXTI线的RTC唤醒中断	0x0000_004C
IRQ 4	20	FMC 全局中断	0x0000_0050
IRQ 5	21	RCU 或 CTC 全局中断	0x0000_0054
IRQ 6	22	EXTI 线 0 中断	0x0000_0058
IRQ 7	23	EXTI 线 1 中断	0x0000_005C
IRQ 8	24	EXTI 线 2 中断	0x0000_0060
IRQ 9	25	EXTI 线 3 中断	0x0000_0064
IRQ 10	26	EXTI 线 4 中断	0x0000_0068
IRQ 11	27	DMA 通道 0 全局中断	0x0000_006C
IRQ 12	28	DMA 通道 1 全局中断	0x0000_0070
IRQ 13	29	DMA 通道 2 全局中断	0x0000_0074
IRQ 14	30	DMA 通道 3 全局中断	0x0000_0078
IRQ 15	31	DMA 通道 4 全局中断	0x0000_007C
IRQ 16	32	DMA 通道 5 全局中断	0x0000_0080
IRQ 17	33	DMA 通道 6 全局中断	0x0000_0084
IRQ 18	34	ADC 中断	0x0000_0088
IRQ 19	35	USBD 高优先级中断	0x0000_008C
IRQ 20	36	USBD 低优先级中断	0x0000_0090
IRQ 21	37	TIMER1 全局中断	0x0000_0094
IRQ 22	38	TIMER2 全局中断	0x0000_0098
IRQ 23	39	TIMER8 全局中断	0x0000_009C
IRQ 24	40	TIMER11 全局中断	0x0000_00A0
IRQ 25	41	TIMER5 全局中断	0x0000_00A4
IRQ 26	42	TIMER6 全局中断	0x0000_00A8
IRQ 27	43	USART0 全局中断	0x0000_00AC
IRQ 28	44	USART1 全局中断	0x0000_00B0
IRQ 29	45	UART3 全局中断	0x0000_00B4

中断编号	向量编号	外设中断描述	向量地址
IRQ 30	46	UART4 全局中断	0x0000_00B8
IRQ 31	47	I2C0 事件中断	0x0000_00BC
IRQ 32	48	I2C0 错误中断	0x0000_00C0
IRQ 33	49	I2C1 事件中断	0x0000_00C4
IRQ 34	50	I2C1 错误中断	0x0000_00C8
IRQ 35	51	SPI0 全局中断	0x0000_00CC
IRQ 36	52	SPI1 全局中断	0x0000_00D0
IRQ 37	53	DAC 中断	0x0000_00D4
IRQ 38	54	保留	0x0000_00D8
IRQ 39	55	I2C2 事件中断	0x0000_00DC
IRQ 40	56	I2C2 错误中断	0x0000_00E0
IRQ 41	57	连接到 EXTI 线的 RTC 闹钟中断	0x0000_00E4
IRQ 42	58	连接到 EXTI 线的 USBD 唤醒中断	0x0000_00E8
IRQ 43	59	EXTI线[9:5]中断	0x0000_00EC
IRQ 44-46	60-62	保留	0x0000_00F0-0x0000_00F8
IRQ 47	63	EXTI 线[15:10]中断	0x0000_00FC
IRQ 48-54	64-70	保留	0x0000_0100-0x0000_0118
IRQ 55	71	DMA MUX 中断	0x0000_011C
IRQ56	72	连接到 EXTI 线的 CMP0 输出中断	0x0000_0120
IRQ57	73	连接到 EXTI 线的 CMP1 输出中断	0x0000_0124
IRQ 58	74	连接到 EXTI 线的 I2C0 唤醒中断	0x0000_0128
IRQ 59	75	连接到 EXTI 线的 I2C2 唤醒中断	0x0000_012C
IRQ 60	76	连接到 EXTI 中断线的 USART0 唤醒中断	0x0000_0130
IRQ 61	77	LPUART 全局中断	0x0000_0134
IRQ 62	78	CAU 全局中断	0x0000_0138
IRQ 63	79	TRNG 全局中断	0x0000_013C
IRQ 64	80	SLCD 全局中断	0x0000_0140
IRQ 65	81	连接到 EXTI 线的 USART1 唤醒中断	0x0000_0144
IRQ 66	82	连接到 EXTI 线的 I2C1 唤醒中断	0x0000_0148
IRQ 67	83	连接到 EXTI 线的 LPUART 唤醒中断	0x0000_014C
IRQ 68	84	LPTIMER 全局中断	0x0000_0150

6.4. 外部中断及事件(EXTI)结构框图

图 6-1. EXTI 结构框图



6.5. 外部中断及事件功能概述

EXTI包含30个相互独立的边沿检测电路并且可以向处理器产生中断请求或事件唤醒。EXTI提供3种触发类型：上升沿触发，下降沿触发和任意沿触发。EXTI中每个边沿检测电路都可以分别予以配置或屏蔽。

EXTI触发源包括来自I/O管脚的16根线以及来自内部模块的14根线，具体细节参考*表6-3. EXTI触发源*。通过配置SYSCFG模块的SYSCFG_EXTISSx寄存器，所有的GPIO管脚都可以被选作EXTI的触发源，具体细节请参考*系统配置寄存器（SYSCFG）*。

除了中断，EXTI还可以向处理器提供事件信号。Cortex®-M23内核完全支持等待中断（WFI），等待事件（WFE）和发送事件（SEV）指令。唤醒中断控制器（WIC）可以让处理器和NVIC进入功耗极低的省电模式，由WIC来识别中断和事件以及判断优先级。当某些预期的事件发生时，EXTI能唤醒处理器及整个系统，例如一个特定的I/O管脚电平翻转或者RTC闹钟。

表 6-3. EXTI 触发源

EXTI 线编号	触发源
0	PA0 / PB0 / PC0 / PD0 / PF0
1	PA1 / PB1 / PC1 / PD1 / PF1
2	PA2 / PB2 / PC2 / PD2

EXTI 线编号	触发源
3	PA3 / PB3 / PC3 / PD3
4	PA4 / PB4 / PC4 / PD4
5	PA5 / PB5 / PC5 / PD5
6	PA6 / PB6 / PC6 / PD6
7	PA7 / PB7 / PC7
8	PA8 / PB8 / PC8 / PD8
9	PA9 / PB9 / PC9 / PD9
10	PA10 / PB10 / PC10
11	PA11 / PB11 / PC11
12	PA12 / PB12 / PC12
13	PA13 / PB13 / PC13
14	PA14 / PB14 / PC14
15	PA15 / PB15 / PC15
16	LVD
17	RTC 闹钟
18	USBD 唤醒
19	RTC 干预和时间戳
20	RTC 唤醒
21	CMP0 输出
22	CMP1 输出
23	I2C0 唤醒
24	I2C2 唤醒
25	USART0 唤醒
26	USART1 唤醒
27	I2C1 唤醒
28	LPUART 唤醒
29	LPTIMER 唤醒

6.6. EXTI 寄存器

EXTI基地址: 0x4001 0400

6.6.1. 中断使能寄存器（EXTI_INTEN）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		INTEN29	INTEN28	INTEN27	INTEN26	INTEN25	INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
		rw	rw	rw	rw	rw	Rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN15	INTEN14	INTEN13	INTEN12	INTEN11	INTEN10	INTEN9	INTEN8	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0
rw	rw	rw	rw	rw	rw	rw	Rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:0	INTENx	中断使能位x（x = 0..29） 0：第x中断被禁止 1：第x中断被使能

6.6.2. 事件使能寄存器（EXTI_EVEN）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		EVEN29	EVEN28	EVEN27	EVEN26	EVEN25	EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVEN15	EVEN14	EVEN13	EVEN12	EVEN11	EVEN10	EVEN9	EVEN8	EVEN7	EVEN6	EVEN5	EVEN4	EVEN3	EVEN2	EVEN1	EVEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:0	EVENx	事件使能位x（x = 0..29） 0：第x事件被禁止 1：第x事件被使能

6.6.3. 上升沿触发使能寄存器（EXTI_RTEN）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		RTEN29	RTEN28	RTEN27	RTEN26	RTEN25	RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTEN15	RTEN14	RTEN13	RTEN12	RTEN11	RTEN10	RTEN9	RTEN8	RTEN7	RTEN6	RTEN5	RTEN4	RTEN3	RTEN2	RTEN1	RTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:0	RTENx	上升沿触发使能（x = 0..29） 0：第x线上升沿触发无效 1：第x线上升沿触发有效（中断/事件请求）

6.6.4. 下降沿触发使能寄存器（EXTI_FTEN）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		FTEN29	FTEN28	FTEN27	FTEN26	FTEN25	FTEN24	FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTEN15	FTEN14	FTEN13	FTEN12	FTEN11	FTEN10	FTEN9	FTEN8	FTEN7	FTEN6	FTEN5	FTEN4	FTEN3	FTEN2	FTEN1	FTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:0	FTENx	下降沿触发使能（x = 0..29） 0：第x线下降沿触发无效 1：第x线下降沿触发有效（中断/事件请求）

6.6.5. 软件中断事件寄存器（EXTI_SWIEV）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		SWIEV29	SWIEV28	SWIEV27	SWIEV26	SWIEV25	SWIEV24	SWIEV23	SWIEV22	SWIEV21	SWIEV21	SWIEV19	SWIEV18	SWIEV17	SWIEV16
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIEV15	SWIEV14	SWIEV13	SWIEV12	SWIEV11	SWIEV10	SWIEV9	SWIEV8	SWIEV7	SWIEV6	SWIEV5	SWIEV4	SWIEV3	SWIEV2	SWIEV1	SWIEV0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:0	SWIEVx	中断/事件软件触发（x = 0..29） 0：禁用EXTI线x软件中断/事件请求 1：激活EXTI线x软件中断/事件请求

6.6.6. 挂起寄存器（EXTI_PD）

地址偏移：0x14

复位值：0xXXXX XXXX，X表示未定义。

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		PD29	PD28	PD27	PD26	PD25	PD24	PD23	PD22	PD21	PD21	PD19	PD19	PD17	PD16
		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rw	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:0	PDx	中断挂起状态（x = 0..29） 0：EXTI线x没有被触发 1：EXTI线x被触发 对这些位写1，可将其清0。

7. 通用和备用输入/输出接口（GPIO 和 AFIO）

7.1. 简介

最多可支持 59 个通用 I/O 引脚（GPIO），分别为 PA0 ~ PA15，PB0 ~ PB15，PC0 ~ PC15，PD0 ~ PD6，PD8 ~ PD9，PF0 ~ PF1。各片上设备用其来实现逻辑输入/输出功能。每个 GPIO 端口有相关的控制和配置寄存器以满足特定应用的需求。片上设备 GPIO 引脚的外部中断由 EXTI 模块的寄存器控制和配置。

GPIO 端口和其他的备用功能（AFs）备用引脚，在特定的封装下获得最大的灵活性。GPIO 引脚通过配置相关的寄存器可以用作备用功能引脚，备用功能输入/输出都可以。

每个 GPIO 引脚可以由软件配置为输出（推挽或开漏）、输入、外设备用功能或者模拟模式。每个 GPIO 引脚都可以配置为上拉、下拉或无上拉/下拉。除模拟模式外，所有的 GPIO 引脚都具备大电流驱动能力。

7.2. 主要特征

- 输入/输出方向控制；
- 施密特触发输入功能使能控制；
- 每个引脚都具有弱上拉/下拉功能；
- 推挽/开漏输出使能控制；
- 置位/复位输出使能；
- 可编程的边沿触发外部中断-由 EXTI 寄存器配置；
- 模拟输入/输出配置；
- 备用功能输入/输出配置；
- 端口锁定配置；
- 单周期输出翻转功能。

7.3. 功能说明

每个通用I/O端口都可以通过32位控制寄存器（GPIOx_CTL）配置为GPIO输入，GPIO输出，AF功能或模拟模式。引脚AFIO输入/输出是通过AFIO功能使能来选择。当端口配置为输出（GPIO输出或AFIO输出）时，可以通过GPIO输出模式寄存器（GPIOx_OMODE）配置为推挽或开漏模式。输出端口的最大速度可以通过GPIO输出速度寄存器（GPIOx_OSPD）配置。每个端口可以通过GPIO上/下拉寄存器（GPIOx_PUD）配置为浮空（无上拉或下拉），上拉或下拉功能。

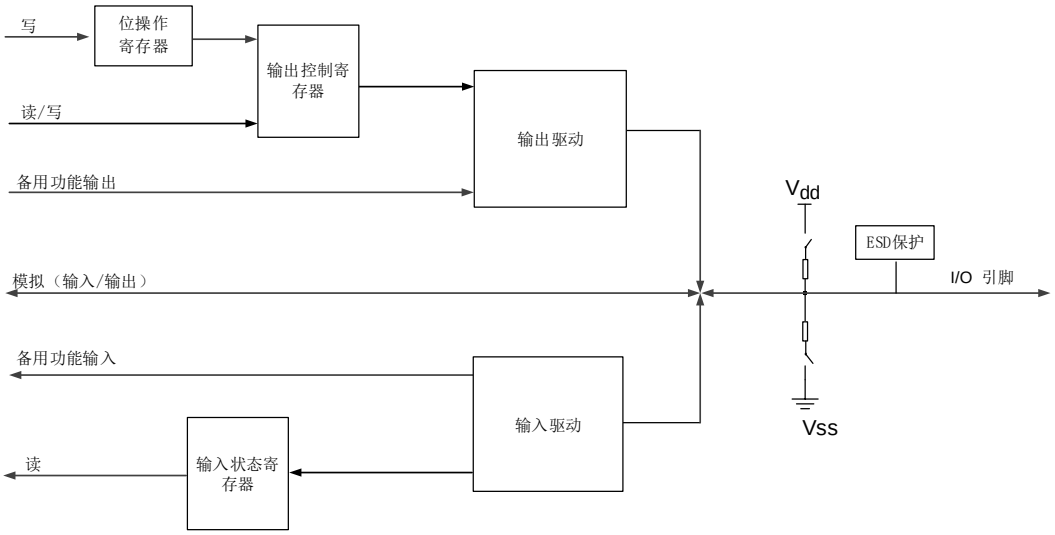
表 7-1. GPIO 配置表

PAD TYPE			CTLy	OMy	PUDy
GPIO 输入	X	悬空	00	X	00
		上拉			01
		下拉			10

PAD TYPE			CTLy	OMy	PUDy
GPIO 输出	推挽	悬空	01	0	00
		上拉			01
		下拉			10
	开漏	悬空		1	00
		上拉			01
		下拉			10
AFIO 输入	X	悬空	10	X	00
		上拉			01
		下拉			10
AFIO 输出	推挽	悬空	10	0	00
		上拉			01
		下拉			10
	开漏	悬空		1	00
		上拉			01
		下拉			10
ANALOG	X	X	11	X	XX

*图7-1. GPIO端口位的基本结构*为标准I/O端口位的基本结构图。

图 7-1. GPIO 端口位的基本结构



7.3.1. GPIO 引脚配置

在复位期间或复位之后，备用功能并未激活，所有 GPIO 端口都被配置成输入浮空模式，这种输入模式禁用上拉(PU)/下拉(PD)电阻。但是复位后，串行线调试为输入 PU/PD 模式。

PA14：SWCLK为PD下拉模式
PA13：SWDIO为PU上拉模式

GPIO管脚可以配置为输入或输出。并且所有的GPIO管脚都有一个内部的弱上拉和弱下拉可以选择。当GPIO管脚可配置为输入管脚时，外部管脚上的数据在每个AHB时钟周期时都会装载

到端口输入状态寄存器（GPIOx_ISTAT）。

当GPIO引脚配置为输出引脚，用户可以配置端口的输出速度和选择输出驱动模式：推挽或开漏模式。端口输出控制寄存器（GPIOx_OCTL）的值将会从相应I/O引脚上输出。

当需要对GPIOx_OCTL进行按位写操作时不需关中断，用户可以通过写‘1’到位操作寄存器（GPIOx_BOP，或用于清0的GPIOx_BC，或用于翻转操作的GPIOx_TG）修改一位或几位，该过程仅需要一个最小的AHB写访问周期，而其他位不受影响。

7.3.2. 外部中断及事件

所有的端口都有外部中断的能力，如果想使用端口的外部中断功能，需要配置为输入模式。

7.3.3. 备用功能（AF）

当端口配置为AFIO（设置GPIOx_CTL寄存器中的CTLy值为“0b10”）时，该端口用作外设备用功能。通过配置GPIO备用功能选择寄存器（GPIOx_AFSELy(y=0..1)），每个端口可以配置16个备用功能。端口备用功能分配的详细介绍见芯片数据手册。

7.3.4. 附加功能

有些引脚具有附加功能，它们优先于标准GPIO寄存器中的配置。当用作ADC，DAC，CMP或附加功能时，引脚必须配置成模拟模式。当引脚用作RTC、WKUPx和振荡器附加功能时，端口类型通过相关的RTC、PMU和RCU寄存器自动设置。当附加功能禁用时，这些端口可用作普通GPIO。

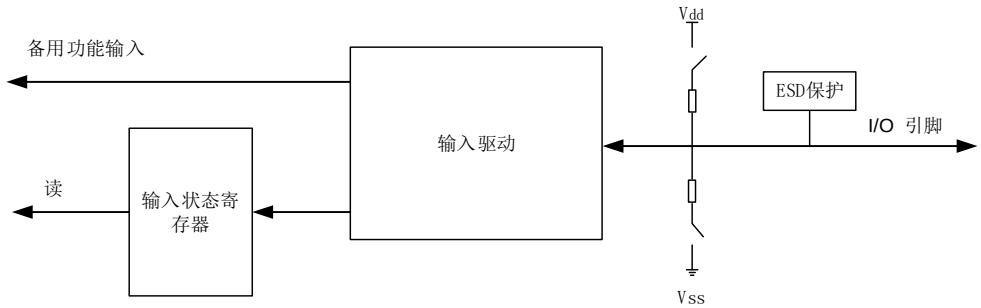
7.3.5. 输入配置

当GPIO引脚配置为输入时：

- 施密特触发输入使能；
- 可选择的弱上拉和下拉电阻；
- 当前I/O引脚上的数据在每个AHB时钟周期都会被采样并存入端口输入状态寄存器；
- 输出缓冲器禁用。

*图7-2. 输入配置的基本结构*是I/O引脚的输入配置。

图 7-2. 输入配置的基本结构



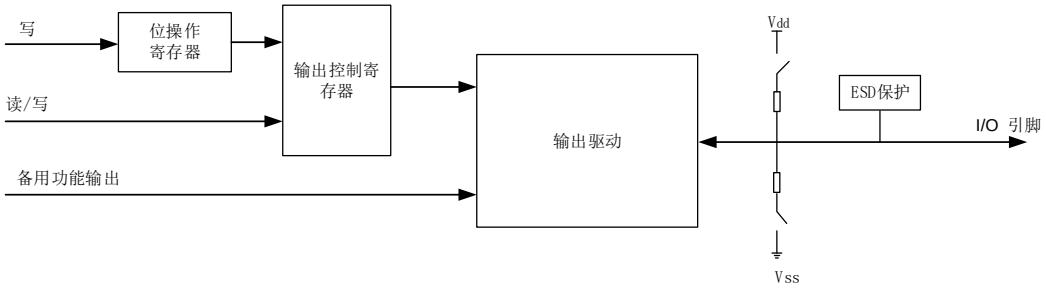
7.3.6. 输出配置

当GPIO配置为输出时：

- 施密特触发输入使能；
- 可选择的弱上拉和下拉电阻；
- 开漏模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；输出控制寄存器设置为“1”，相应管脚处于高阻状态；
- 推挽模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；输出控制寄存器设置为“1”，相应引脚输出高电平；
- 在推挽模式下，对端口输出控制寄存器的读访问将返回上次写入的值；
- 在开漏模式下，对端口输入状态寄存器的读访问将返回I/O的状态。

*图 7-3. 输出配置的基本结构*是 I/O 端口的输出配置。

图 7-3. 输出配置的基本结构



7.3.7. 模拟配置

当GPIO引脚用于模拟模式时：

- 弱上拉和下拉电阻禁用；
- 输出缓冲器禁用；
- 施密特触发输入禁用；
- 读端口输入状态寄存器返回“0”。

*图7-4. 模拟配置的基本结构*是I/O端口的模拟高阻配置。

图 7-4. 模拟配置的基本结构

7.3.8. 备用功能（AF）配置

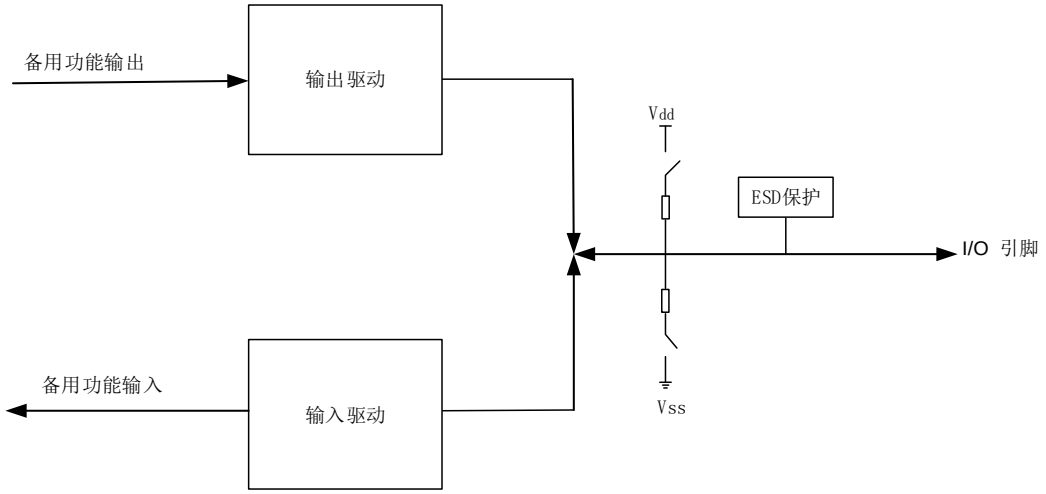
为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

当引脚配置为备用功能时：

- 输出缓冲器启用开漏或者推挽功能；
- 输出缓冲器由外设驱动；
- 施密特触发输入使能；
- 可选择的弱上拉/下拉电阻；
- I/O引脚上的数据在每个AHB时钟周期采样并存入端口输入状态寄存器；
- 对端口输入状态寄存器进行读操作，将获得I/O口的状态；
- 对端口输出控制寄存器进行读操作，将返回上次写入的值。

*图7-5. 备用功能配置的基本结构*是I/O端口备用功能配置图。

图 7-5. 备用功能配置的基本结构



7.3.9. GPIO 锁定功能

GPIO的锁定机制可以保护I/O端口的配置。

被保护的寄存器有：GPIOx_CTL，GPIOx_OMODE，GPIOx_OSPD，GPIOx_PUD和GPIOx_AFSELy(y=0..1)。通过配置32位锁定寄存器（GPIOx_LOCK）可以锁定I/O端口的配置。当特定LOCK序列写到位于GPIOx_LOCK寄存器的LKK位上，并且LKy被置位，那么对应的端口配置直到下一次复位前将不能改变。建议在电源驱动模块驱动的配置时使用锁定功能。

7.3.10. GPIO 单周期输出翻转功能

通过将GPIOx_TG寄存器中对应的位写1，GPIO可以在一个AHB时钟周期内翻转I/O的输出电平。输出信号的频率可以达到AHB时钟的一半。

7.4. GPIO 寄存器

GPIOA基地址：0x4800 0000

GPIOB基地址：0x4800 0400

GPIOC基地址：0x4800 0800

GPIOD基地址：0x4800 0C00

GPIOF基地址：0x4800 1400

7.4.1. 端口控制寄存器（GPIOx_CTL，x=A..D，F）

地址偏移：0x00

复位值：端口 A 0x2800 0000；其他端口 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CTL15[1:0]	CTL14[1:0]	CTL13[1:0]	CTL12[1:0]	CTL11[1:0]	CTL10[1:0]	CTL9[1:0]	CTL8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15 14	**13 12**	**11 10**	**9 8**	**7 6**	**5 4**	**3 2**	**1 0**
CTL7[1:0]	CTL6[1:0]	CTL5[1:0]	CTL4[1:0]	CTL3[1:0]	CTL2[1:0]	CTL1[1:0]	CTL0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	CTL15[1:0]	Pin 15配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
29:28	CTL14[1:0]	Pin 14配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
27:26	CTL13[1:0]	Pin 13配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
25:24	CTL12[1:0]	Pin 12配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
23:22	CTL11[1:0]	Pin 11配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
21:20	CTL10[1:0]	Pin 10配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述

19:18	CTL9[1:0]	Pin 9配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
17:16	CTL8[1:0]	Pin 8配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
15:14	CTL7[1:0]	Pin 7配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
13:12	CTL6[1:0]	Pin 6配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
11:10	CTL5[1:0]	Pin 5配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
9:8	CTL4[1:0]	Pin 4配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
7:6	CTL3[1:0]	Pin 3配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
5:4	CTL2[1:0]	Pin 2配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
3:2	CTL1[1:0]	Pin 1配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
1:0	CTL0[1:0]	Pin 0配置位 该位由软件置位和清除。 00：GPIO输入模式（复位值） 01：GPIO输出模式 10：备用功能模式 11：模拟模式（输入和输出）

7.4.2. 端口输出模式寄存器（GPIOx_OMODE，x=A..D，F）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

保留

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	OM15	Pin 15输出模式位 该位由软件置位和清除。 参考OM0的描述
14	OM14	Pin 14输出模式位 该位由软件置位和清除。 参考OM0的描述
13	OM13	Pin 13输出模式位 该位由软件置位和清除。 参考OM0的描述
12	OM12	Pin 12输出模式位 该位由软件置位和清除。 参考OM0的描述
11	OM11	Pin 11输出模式位 该位由软件置位和清除。 参考OM0的描述
10	OM10	Pin 10输出模式位 该位由软件置位和清除。 参考OM0的描述
9	OM9	Pin 9输出模式位 该位由软件置位和清除。 参考OM0的描述
8	OM8	Pin 8输出模式位 该位由软件置位和清除。 参考OM0的描述
7	OM7	Pin 7输出模式位 该位由软件置位和清除。 参考OM0的描述
6	OM6	Pin 6输出模式位 该位由软件置位和清除。 参考OM0的描述
5	OM5	Pin 5输出模式位

		该位由软件置位和清除。 参考OM0的描述
4	OM4	Pin 4输出模式位 该位由软件置位和清除。 参考OM0的描述
3	OM3	Pin 3输出模式位 该位由软件置位和清除。 参考OM0的描述
2	OM2	Pin 2输出模式位 该位由软件置位和清除。 参考OM0的描述
1	OM1	Pin 1输出模式位 该位由软件置位和清除。 参考OM0的描述
0	OM0	Pin 0输出模式位 该位由软件置位和清除。 0：输出推挽模式（复位值） 1：输出开漏模式

7.4.3. 端口输出速度寄存器（GPIOx_OSPD，x=A..D，F）

地址偏移：0x08

复位值：端口 A 0x0C00 0000；其他端口 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
OSPD15[1:0]	OSPD14[1:0]	OSPD13[1:0]	OSPD12[1:0]	OSPD11[1:0]	OSPD10[1:0]	OSPD9[1:0]	OSPD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15 14	**13 12**	**11 10**	**9 8**	**7 6**	**5 4**	**3 2**	**1 0**
OSPD7[1:0]	OSPD6[1:0]	OSPD5[1:0]	OSPD4[1:0]	OSPD3[1:0]	OSPD2[1:0]	OSPD1[1:0]	OSPD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	OSPD15[1:0]	Pin 15输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
29:28	OSPD14[1:0]	Pin 14输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
27:26	OSPD13[1:0]	Pin 13输出最大速度位 该位由软件置位和清除。

		参考OSPD0[1:0]的描述
25:24	OSPD12[1:0]	Pin 12输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
23:22	OSPD11[1:0]	Pin 11输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
21:20	OSPD10[1:0]	Pin 10输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
19:18	OSPD9[1:0]	Pin 9输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
17:16	OSPD8[1:0]	Pin 8输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
15:14	OSPD7[1:0]	Pin 7输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
13:12	OSPD6[1:0]	Pin 6输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
11:10	OSPD5[1:0]	Pin 5输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
9:8	OSPD4[1:0]	Pin 4输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
7:6	OSPD3[1:0]	Pin 3输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
5:4	OSPD2[1:0]	Pin 2输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
3:2	OSPD1[1:0]	Pin 1输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
1:0	OSPD0[1:0]	Pin 0输出最大速度位

该位由软件置位和清除。
x0：输出最大速度2M（复位值）
01：输出最大速度10M
11：输出最大速度50M

7.4.4. 端口上拉/下拉寄存器（GPIOx_PUD，x=A..D，F）

地址偏移：0x0C
复位值：端口 A 0x2400 0000；其他端口 0x0000 0000
该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
PUD15[1:0]	PUD14[1:0]	PUD13[1:0]	PUD12[1:0]	PUD11[1:0]	PUD10[1:0]	PUD9[1:0]	PUD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PUD7[1:0]	PUD6[1:0]	PUD5[1:0]	PUD4[1:0]	PUD3[1:0]	PUD2[1:0]	PUD1[1:0]	PUD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	PUD15[1:0]	Pin 15上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
29:28	PUD14[1:0]	Pin 14上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
27:26	PUD13[1:0]	Pin 13上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
25:24	PUD12[1:0]	Pin 12上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
23:22	PUD11[1:0]	Pin 11上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
21:20	PUD10[1:0]	Pin 10上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
19:18	PUD9[1:0]	Pin 9上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
17:16	PUD8[1:0]	Pin 8上拉或下拉位

该位由软件置位和清除。
参照PUD0[1:0]的描述

15:14 PUD7[1:0] Pin 7上拉或下拉位
该位由软件置位和清除。
参照PUD0[1:0]的描述

13:12 PUD6[1:0] Pin 6上拉或下拉位
该位由软件置位和清除。
参照PUD0[1:0]的描述

11:10 PUD5[1:0] Pin 5上拉或下拉位
该位由软件置位和清除。
参照PUD0[1:0]的描述

9:8 PUD4[1:0] Pin 4上拉或下拉位
该位由软件置位和清除。
参照PUD0[1:0]的描述

7:6 PUD3[1:0] Pin 3上拉或下拉位
该位由软件置位和清除。
参照PUD0[1:0]的描述

5:4 PUD2[1:0] Pin 2上拉或下拉位
该位由软件置位和清除。
参照PUD0[1:0]的描述

3:2 PUD1[1:0] Pin 1上拉或下拉位
该位由软件置位和清除。
参照PUD0[1:0]的描述

1:0 PUD0[1:0] Pin 0上拉或下拉位
该位由软件置位和清除。
00：悬空模式，无上拉和下拉（复位值）
01：端口上拉模式
10：端口下拉模式
11：保留

7.4.5. 端口输入状态寄存器（GPIOx_ISTAT，x=A..D，F）

地址偏移：0x10
复位值：0x0000 XXXX
该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	ISTAT9	ISTAT8	ISTAT7	ISTAT6	ISTAT5	ISTAT4	ISTAT3	ISTAT2	ISTAT1	ISTAT0

r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	ISTATy	端口输入状态位(y=0..15) 这些位由软件置位和清除。 0：引脚输入信号为低电平 1：引脚输入信号为高电平

7.4.6. 端口输出控制寄存器（GPIOx_OCTL，x=A..D，F）

地址偏移：0x14

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	OCTLy	端口输出控制位(y=0..15) 该位由软件置位和清除。 0：引脚输出低电平 1：引脚输出高电平

7.4.7. 端口位操作寄存器（GPIOx_BOP，x=A..D，F）

地址偏移：0x18

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOP15	BOP14	BOP13	BOP12	BOP11	BOP10	BOP9	BOP8	BOP7	BOP6	BOP5	BOP4	BOP3	BOP2	BOP1	BOP0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	CRy	端口清除位y(y=0..15)

		该位由软件置位和清除。 0：相应的OCTLy位没有改变 1：清除相应的OCTLy位为0
15:0	BOPy[15:0]	端口置位位y(y=0..15) 该位由软件置位和清除。 0：相应的OCTLy位没有改变 1：设置相应的OCTLy位为1

7.4.8. 端口配置锁定寄存器（GPIOx_LOCK，x=A..D，F）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															LKK
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LK15	LK14	LK13	LK12	LK11	LK10	LK9	LK8	LK7	LK6	LK5	LK4	LK3	LK2	LK1	LK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	LKK	锁定键 该位只能通过Lock Key写序列置位，始终可读。 0：GPIOx_LOCK寄存器和端口配置没有锁定 1：直到下一次MCU复位前，GPIOx_LOCK寄存器被锁定 LOCK key写序列： 写1→写0→写1→读0→读1 **注意：**在LOCK Key写序列期间，LK y(y=0..15)的值必须保持。
15:0	LKy	端口锁定位y(y=0..15) 该位由软件置位和清除。 0：端口配置没有锁定 1：端口配置锁定

7.4.9. 备用功能选择寄存器 0（GPIOx_AFSEL0，x=A..D，F）

地址偏移：0x20

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL7[3:0]				SEL6[3:0]				SEL5[3:0]				SEL4[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3[3:0]				SEL2[3:0]				SEL1[3:0]				SEL0[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	SEL7[3:0]	Pin 7选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
27:24	SEL6[3:0]	Pin 6选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
23:20	SEL5[3:0]	Pin 5选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
19:16	SEL4[3:0]	Pin 4选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
15:12	SEL3[3:0]	Pin 3选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
11:8	SEL2[3:0]	Pin 2选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
7:4	SEL1[3:0]	Pin 1选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
3:0	SEL0[3:0]	Pin 0选择备用功能 该位由软件置位和清除。 0000：选择AF0功能（复位值） 0001：选择AF1功能 0010：选择AF2功能 0011：选择AF3功能 0100：选择AF4功能 0101：选择AF5功能 0110：选择AF6功能 0111：选择AF7功能 1000：选择AF8功能 1001：选择AF9功能 1010 ~ 1111：保留

7.4.10. 备用功能选择寄存器 1（GPIOx_AFSEL1，x=A..D，F）

地址偏移：0x24

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	SEL15[3:0]	Pin 15选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
27:24	SEL14[3:0]	Pin 14选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
23:20	SEL13[3:0]	Pin 13选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
19:16	SEL12[3:0]	Pin 12选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
15:12	SEL11[3:0]	Pin 1选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
11:8	SEL10[3:0]	Pin 10选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
7:4	SEL9[3:0]	Pin 9选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
3:0	SEL8[3:0]	Pin 8选择备用功能 该位由软件置位和清除。 0000：选择AF0功能（复位值） 0001：选择AF1功能 0010：选择AF2功能 0011：选择AF3功能 0100：选择AF4功能

0101：选择AF5功能
0110：选择AF6功能
0111：选择AF7功能
1000：选择AF8功能
1001：选择AF9功能

1010 ~ 1111：保留

7.4.11. 位清除寄存器（GPIOx_BC，x=A..D，F）

地址偏移：0x28
复位值：0x0000 0000
该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRy	端口清除位y(y=0..15) 该位由软件置位和清除。 0：相应OCTLy位没有改变 1：清除相应的OCTLy位

7.4.12. 端口位翻转寄存器（GPIOx_TG，x=A..D，F）

地址偏移：0x2C
复位值：0x0000 0000
该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TG15	TG14	TG13	TG12	TG11	TG10	TG9	TG8	TG7	TG6	TG5	TG4	TG3	TG2	TG1	TG0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	TGy	端口翻转位y(y=0..15)

该位由软件置位和清除。

0：相应OCTLy位没有改变t

1：翻转相应的OCTLy位

8. 循环冗余校验管理单元（CRC）

8.1. 简介

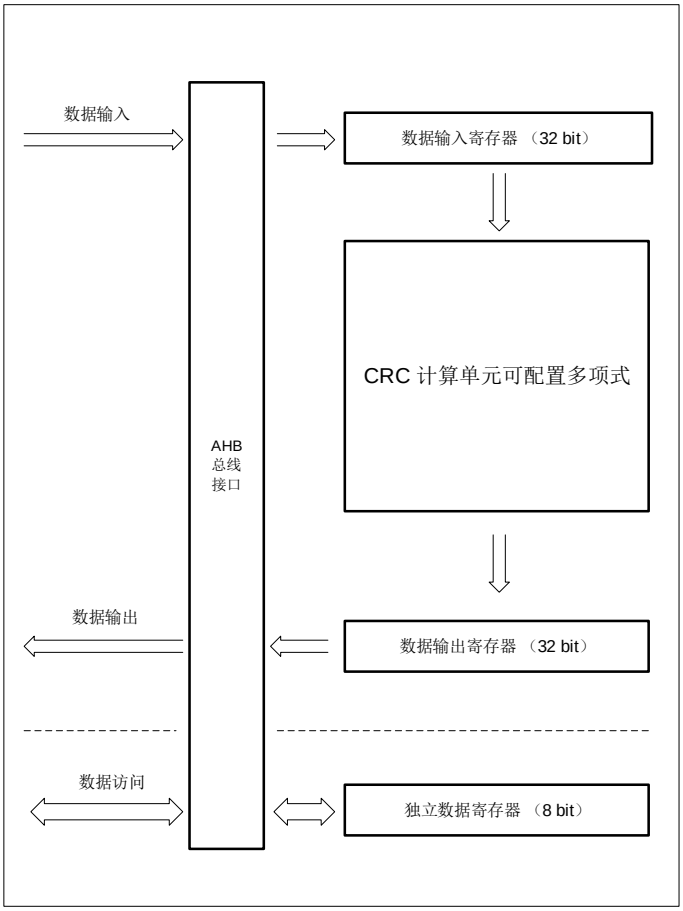
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

CRC 管理单元能用用户配置的多项式来计算 7/8/16/32 位的 CRC 校验码。

8.2. 主要特征

- 支持7/8/16/32位数据输入；
- 对于7（8）/16/32位的输入数据长度，计算周期分别为1/2/4个AHB时钟周期；
- 用户可以配置多项式及多项式长度；
- CRC复位后，用户可以配置计算初值；
- 配有与计算无关的独立8位寄存器，可以供其他任何外设使用。

图 8-1. CRC 计算单元框图



8.3. 功能说明

- CRC计算单元可以用来计算32位的原始数据，CRC_DATA寄存器接收原始数据并存储计算结果。

  如果不通过软件设置CRC_CTL寄存器的方式来清除CRC_DATA寄存器，新输入的原始数据将会基于前一次CRC_DATA寄存器中的结果进行计算。

  对于32/16/8（7）位的数据长度，CRC的计算分别要花费4/2/1个AHB的时钟周期。在此期间，因为32位输入缓存的原因，AHB总线将不会被挂起。

- 此模块提供了一个8位的独立寄存器CRC_FDATA，CRC_FDATA与CRC计算无关，任何时候都可以进行独立的读写操作。

- 逆序功能可以交换输入输出数据的位序。

  输入数据可选择三种逆序形式。

  以原始数据0x3456CDEF为例：

  1）按字节逆序：
  32位数据被分成四组，组内完成颠倒。逆序后的数据为：0x2C6AB3F7

  2）按半字逆序：
  32位数据被分成两组，组内完成颠倒。逆序后的数据为：0x6A2CF7B3

  3）按字逆序：
  32位数据被分成一组，组内完成颠倒，逆序后的数据为：0xF7B36A2C

  对于输出数据来说，逆序形式为按字逆序。

  例如：当REV_O=1，计算结果0x3344CCDD将被逆序成0xBB3322CC。

- 用户可配置的初始计算数据。

  当RST位置位或对CRC_IDATA寄存器进行写操作时，CRC_DATA寄存器将自动初始化为CRC_IDATA寄存器中的值。

- 用户配置多项式。

  通过配置PS[1:0]，用户可以选择有效多项式和输出数据位宽。如果多项式少于32位，那么输入和输出数据的高位无效。当PS[1:0]或多项式改变后，需要复位CRC。

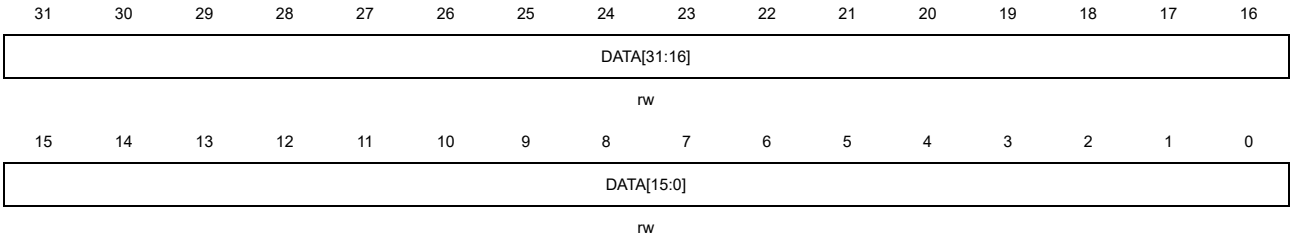
8.4. CRC 寄存器

CRC基地址：0x4002 3000

8.4.1. 数据寄存器（CRC_DATA）

地址偏移：0x00
复位值：0xFFFF FFFF

该寄存器只能按字（32 位）访问。

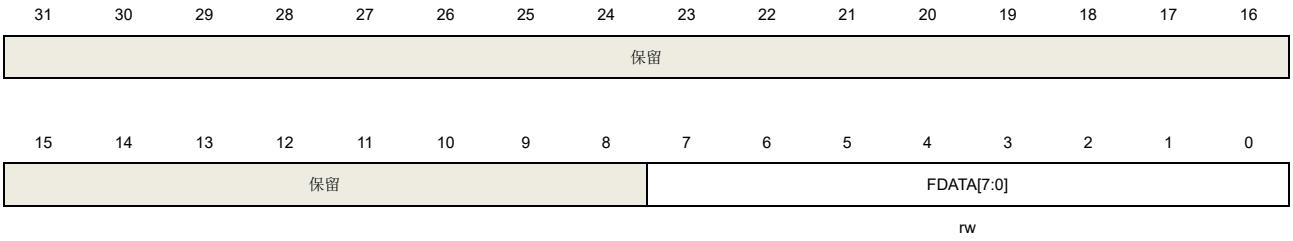


位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次 CRC 计算的结果。

8.4.2. 独立数据寄存器（CRC_FDATA）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



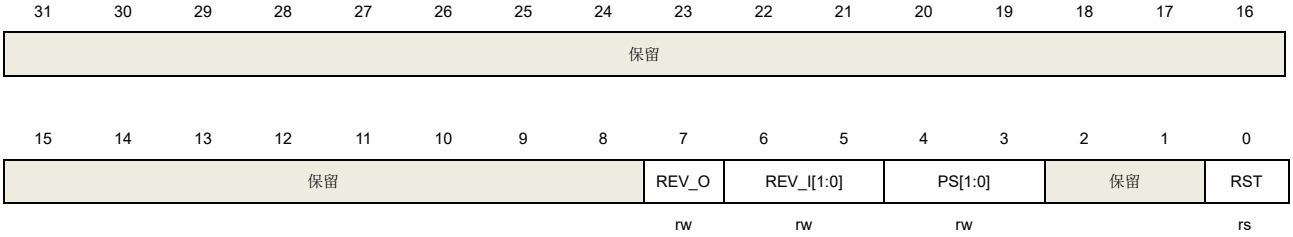
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	FDATA[7:0]	独立数据寄存器位 软件可读可写。 这些位与 CRC 计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受 CRC_CTL 寄存器的影响。

8.4.3. 控制寄存器（CRC_CTL）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



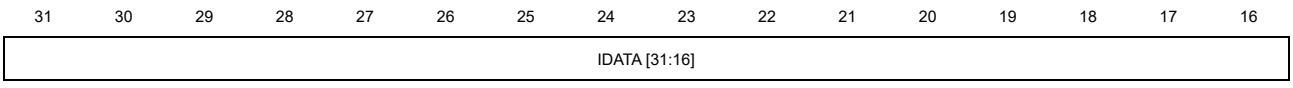
位/位域	名称	描述
31:8	保留	必须保持复位值。
7	REV_O	按位顺序翻转输出数据功能 0：输出数据不翻转 1：输出数据按位顺序翻转
6:5	REV_I[1:0]	翻转输入数据功能 0：输入数据不翻转 1：输入数据按字节翻转 2：输入数据按半字翻转 3：输入数据按字翻转
4:3	PS[1:0]	多项式长度 0：32 位 1：16（POLY[15:0]用于计数）位 2：8（POLY[7:0]用于计数）位 3：7（POLY[6:0]用于计数）位
2:1	保留	必须保持复位值。
0	RST	软件可读写 该位用来复位 CRC_DATA 寄存器。 置位时，CRC_DATA 寄存器的值将自动初始化为 CRC_IDATA 寄存器中的值，然后自动清零。该位对 CRC_FDATA 寄存器没有影响。

8.4.4. 初值寄存器（CRC_IDATA）

地址偏移：0x10

复位值：0xFFFF FFFF

该寄存器只能按字（32 位）访问



rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDATA[15:0]															

rw

位/位域	名称	描述
31:0	IDATA[31:0]	配置 CRC 初值 CRC_CTL 寄存器的 RST 位置位后，CRC_DATA 寄存器的值将被更新为此寄存器的值。

8.4.5. 多项式寄存器（CRC_POLY）

地址偏移：0x14

复位值：0x04C1 1DB7

该寄存器只能按字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POLY [31:16]															

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLY[15:0]															

rw

位/位域	名称	描述
31:0	POLY[31:0]	配置多项式值 配合 PS[1：0]使用。

9. 真随机数生成器（TRNG）

9.1. 简介

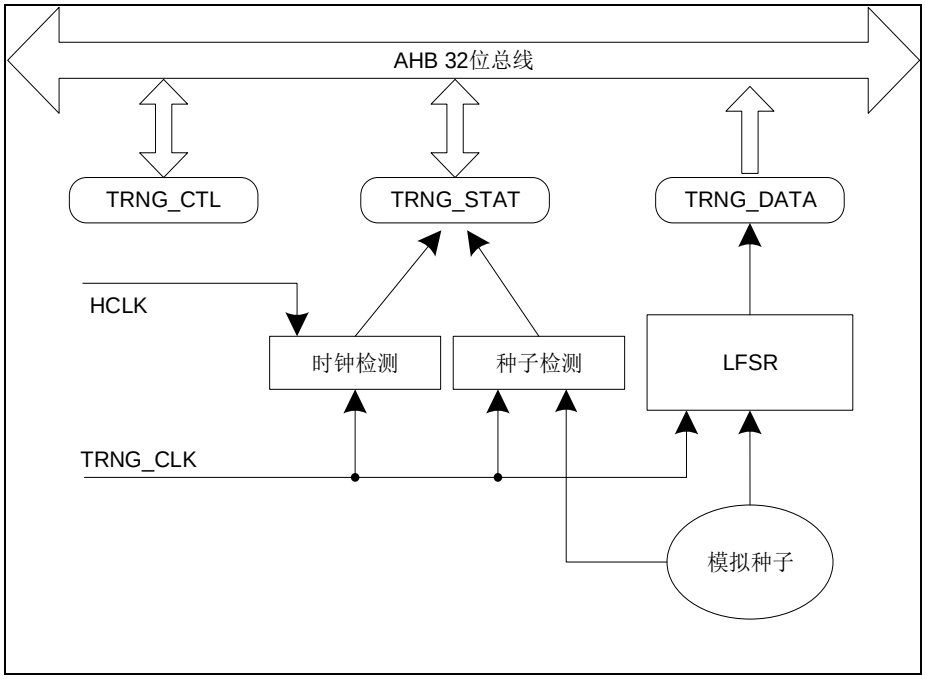
真随机数发生器模块（TRNG）能够通过连续模拟噪声生成一个 32 位的随机数值。

9.2. 主要特征

- 两个连续随机数的间隔大约为40个TRNG_CLK时钟周期；
- 32位随机数的种子是由模拟噪声产生的，因此该随机数是一个真随机数值。

9.3. 功能说明

图 9-1. TRNG 模块框图



随机数种子由模拟电路实现。模拟种子信号输出到一个线性反馈移位寄存器（LFSR）之后在该寄存器中转化成一个 32 位宽度的随机数。

该模拟种子由几个环形振荡器的输出生成。LFSR 由可配置的 TRNG_CLK 时钟（参考 ***RCU*** 相关章节）驱动，因此随机数质量仅与 TRNG_CLK 时钟有关，与 HCLK 频率无关。

当有足够数量的种子被输入 LFSR 之后，LFSR 会输出 32 位数据到 TRNG_DATA 寄存器。同时，系统会监视模拟种子和 TRNG_CLK 时钟。一旦模拟种子发生错误或者时钟产生错误，TRNG_STAT 寄存器的相关状态位将被置 1，如果 TRNG_CTL 寄存器的 TRNGIE 位同时被置 1 还将产生中断。

9.3.1. 操作流程

以下步骤为 TRNG 模块的推荐操作流程：

1）根据需要使能中断，这样当随机数或错误产生时，将会触发一个中断；

2）使能IRC48M时钟，或者选择USBSEL[1:0]为CK_PLL，来使能CK_USBD时钟；

3）使能TRNGEN位；

4）等待中断发生，检测TRGN_STAT寄存器，如果SEIF = 0，CEIF = 0并且DRDY = 1那么数据寄存器中的随机值可以被读取。

按照 FIPS PUB 140-2 的要求，数据寄存器中的第一个随机数需要保留而不是被使用。每一个新生成的随机数应当与之前的随机数相比较。只有当该随机数与前一个随机数不相等时，该数据才可被使用。

9.3.2. 错误标志

（1）时钟错误

当 TRNG_CLK 时钟频率低于 HCLK 频率的 1 / 16 时，CECS 和 CEIF 位将被置 1。此时，软件应当检查 TRNG_CLK 和 HCLK 时钟频率配置并清除 CEIF 位。时钟错误对上一个产生的随机数没有影响。

（2）种子错误

当模拟种子的值在 64 个 TRNG_CLK 时钟周期内不发生变化或连续不断的翻转，SECS 和 SEIF 位将被置位。此时，数据寄存器中的随机数值不应当被使用，并且软件需要清除 SEIF 位。之后将 TRNGEN 位清零并置 1 以便重新启动 TRNG 模块。

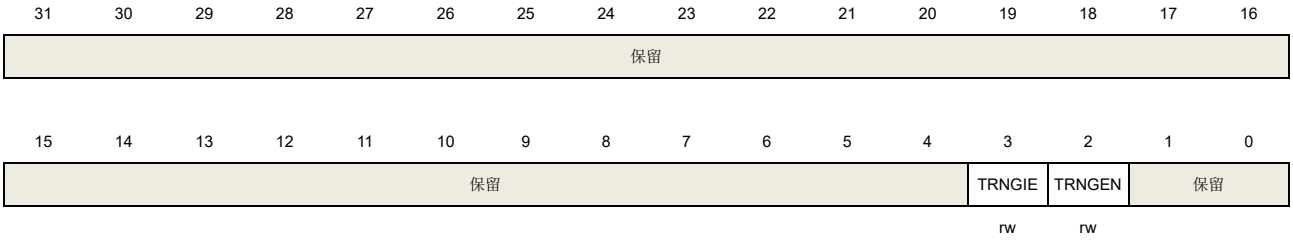
9.4. TRNG 寄存器

TRNG 基地址：0x5006 0800

9.4.1. 控制寄存器（TRNG_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

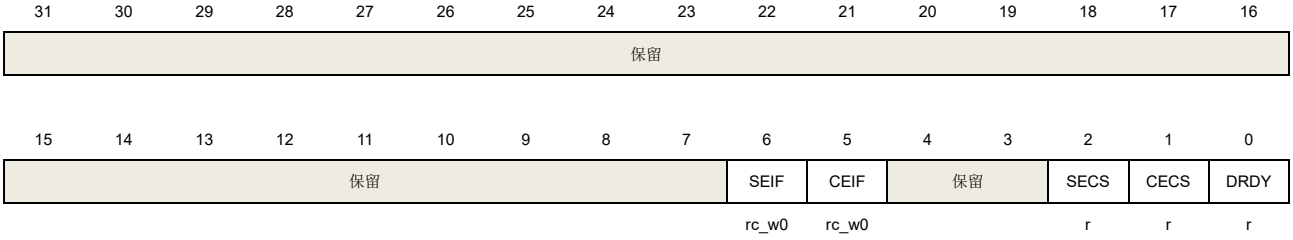


位/位域	名称	描述
31:4	保留	必须保持复位值。
3	TRNGIE	中断使能位，当 DRDY，SEIF 或 CEIF 位被置位时该位控制生成一个中断。 0：禁止 TRNG 中断 1：使能 TRNG 中断
2	TRNGEN	TRNG 使能位 0：禁止 TRNG 模块 1：使能 TRNG 模块
1:0	保留	必须保持复位值。

9.4.2. 状态寄存器（TRNG_STAT）

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。



位/位域	名称	描述
31:7	保留	必须保持复位值。

6	SEIF	种子错误中断标志位 如果超过 64 个连续位具有相同值或超过 32 组连续交替的 0 和 1 被检测到则此位将置 1。 0：未检测到错误 1：检测到种子错误。写 0 将清除该位
5	CEIF	时钟错误中断标志位 如果 TRNG_CLK 时钟频率低于 HCLK 频率的 1 / 16 时该位被置位。 0：未检测到错误 1：检测到时钟错误。写 0 将清除该位
4:3	保留	必须保持复位值。
2	SECS	种子错误当前状态 0：当前未检测到种子错误。如果 SEIF = 1 和 SECS = 0，说明之前已经检测到种子错误但现在已恢复正常。 1：当前检测到种子错误。如果超过 64 个连续位具有相同值或超过 32 组连续交替的 0 和 1 被检测到时，该位置 1。
1	CECS	时钟错误当前状态 0：当前未检测到时钟错误。如果 CEIF = 1 和 CECS = 0，则意味着之前已检测到时钟错误但现在已恢复正常。 1：当前检测到时钟错误。此时 TRNG_CLK 时钟频率低于 1 / 16 HCLK 频率。
0	DRDY	随机数准备状态位 读 TRNG_DATA 寄存器会清零该位，当一个新的随机数产生时被置位。 0：TRNG 数据寄存器的内容无效 1：TRNG 数据寄存器的内容有效

9.4.3. 数据寄存器（TRNG_DATA）

地址偏移：0x08
复位值：0x0000 0000

在读此寄存器之前，软件必须确保 DRDY 位已置 1。

该寄存器只能按字（32 位）访问。

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

位/位域	名称	描述
31:0	TRNGDATA[31:0]	32 位随机数据

10. 直接存储器访问控制器（DMA）

10.1. 简介

DMA 控制器提供了一种硬件传输方式，在外设和存储器之间或者存储器和存储器之间传输数据，而无需 CPU 的介入，从而使 CPU 可以专注在处理其他系统功能上。DMA 控制器有 7 个通道。每个通道都是专门用来处理一个或多个外设的存储器访问请求的。DMA 控制器内部实现了一个仲裁器，用来仲裁多个 DMA 请求的优先级。

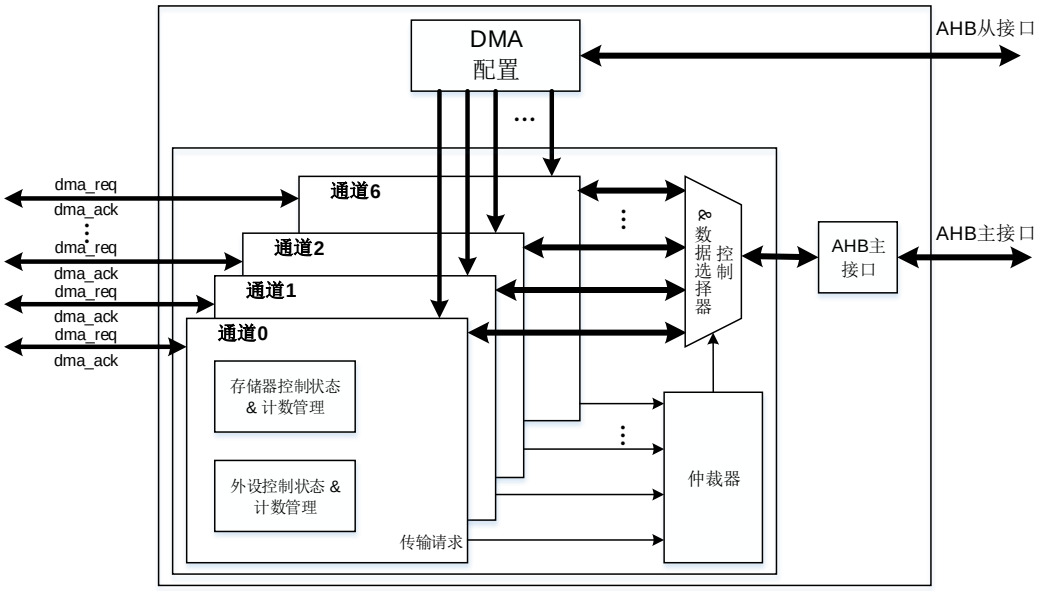
DMA 控制器和 Cortex®-M23 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

10.2. 主要特征

- 传输数据长度可编程配置，最大到 65536；
- 7 个通道，并且每个通道都可配置；
- AHB 和 APB 外设，片上闪存和 SRAM 都可以作为访问的源端和目的端；
- 每个通道连接的 DMA 请求不固定；
- 支持 DMA 软件优先级（低、中、高、极高）和硬件优先级（通道号越低，优先级越高）；
- 存储器和外设的数据传输宽度可配置：字节，半字，字；
- 存储器和外设的数据传输支持固定寻址和增量式寻址；
- 支持循环传输模式；
- 支持外设到存储器，存储器到外设，存储器到存储器的数据传输；
- 每个通道有 3 种类型的事件标志和独立的中断，支持中断的使能和清除；
- 支持中断使能和清除。

10.3. 结构框图

图 10-1. DMA 结构框图



由*图 10-1. DMA 结构框图*所示，DMA 控制器由 4 部分组成：

- AHB 从接口配置 DMA；
- AHB 主接口进行数据传输，用于存储器访问和外设访问；
- 仲裁器进行 DMA 请求的优先级管理；
- 通道管理用于控制数据/地址选择和数据计数。

10.4. 功能说明

10.4.1. DMA 操作

DMA 传输分为两步操作：从源地址读取数据，之后将读取的数据存储到目的地址。DMA 控制器基于 DMA_CHxPADDR、DMA_CHxMADDR、DMA_CHxCTL 寄存器的值计算下一次操作的源/目的地址。DMA_CHxCNT 寄存器用于控制传输的次数。DMA_CHxCTL 寄存器的 PWIDTH 和 MWIDTH 位域决定每次发送和接收的字节数（字节/半字/字）。

假设 DMA_CHxCNT 寄存器的值为 4，并且 PNAGA 和 MNAGA 位均置位。结合 PWIDTH 和 MWIDTH 的各种配置，DMA 传输的操作详见*表 10-1. DMA 传输操作*。

表 10-1. DMA 传输操作

传输宽度		传输操作	
源	目标	源	目标
32 bits	32 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B3B2B1B0[31:0] @0x0 2: Write B7B6B5B4[31:0] @0x4 3: Write BBBAB9B8[31:0] @0x8 4: Write BFBEBDBC[31:0] @0xC
32 bits	16 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B1B0[15:0] @0x0 2: Write B5B4[15:0] @0x2 3: Write B9B8[15:0] @0x4 4: Write BDBC[15:0] @0x6
32 bits	8 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B0[7:0] @0x0 2: Write B4[7:0] @0x1 3: Write B8[7:0] @0x2 4: Write BC[7:0] @0x3
16 bits	32 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write 0000B1B0[31:0] @0x0 2: Write 0000B3B2[31:0] @0x4 3: Write 0000B5B4[31:0] @0x8 4: Write 0000B7B6[31:0] @0xC
16 bits	16 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B1B0[15:0] @0x0 2: Write B3B2[15:0] @0x2 3: Write B5B4[15:0] @0x4 4: Write B7B6[15:0] @0x6
16 bits	8 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B0[7:0] @0x0 2: Write B2[7:0] @0x1 3: Write B4[7:0] @0x2 4: Write B6[7:0] @0x3
8 bits	32 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1: Write 000000B0[31:0] @0x0 2: Write 000000B1[31:0] @0x4 3: Write 000000B2[31:0] @0x8 4: Write 000000B3[31:0] @0xC
8 bits	16 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write 00B0[15:0] @0x0 2, Write 00B1[15:0] @0x2 3, Write 00B2[15:0] @0x4 4, Write 00B3[15:0] @0x6
8 bits	8 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write B0[7:0] @0x0 2, Write B1[7:0] @0x1 3, Write B2[7:0] @0x2 4, Write B3[7:0] @0x3

DMA_CHxCNT寄存器的CNT位域必须在CHEN位置位前被配置，该位域控制传输的次数。在传输过程中，CNT位域的值表示还有多少次数据传输将被执行。

将 DMA_CHxCTL 寄存器的 CHEN 位清零，可以停止 DMA 传输。

- 若 CHEN 位被清零时 DMA 传输还未完成，重新使能 CHEN 位 DMA 传输将分两种情况：
  - 在重新使能 DMA 通道前，未对该通道的相关寄存器进行操作，则 DMA 将继续完成上次的传输；
  - 在重新使能 DMA 通道前，对相应通道的 DMA_CHxCNT、DMA_CHxPADDR 或 DMA_CHxMADDR 寄存器进行了操作，则 DMA 将开始一次新的传输。
- 若清零 CHEN 位时，DMA 传输已经完成，之后未对相应通道的 DMA_CHxCNT、DMA_CHxPADDR 或 DMA_CHxMADDR 寄存器进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

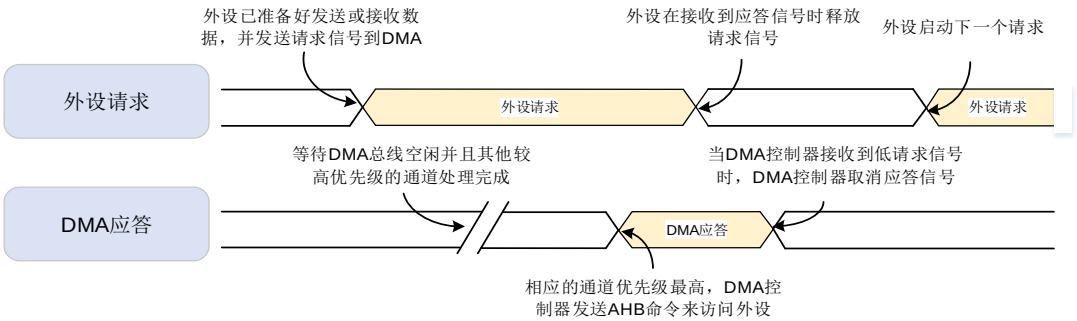
10.4.2. 外设握手

为了保证数据的有效传输，DMA控制器中引入了外设和存储器的握手机制，包括请求信号和应答信号：

- 请求信号：由外设发出，表明外设已经准备好发送或接收数据；
- 应答信号：由 DMA 控制器响应，表明 DMA 控制器已经发送 AHB 命令去访问外设。

*图10-2. 握手机制*中详细描述了DMA控制器与外设之间的握手机制。

图 10-2. 握手机制



10.4.3. 仲裁

当DMA控制器在同一时间接收到多个外设请求时，仲裁器将根据外设请求的优先级来决定响应哪一个外设请求。优先级包括软件优先级和硬件优先级，优先级规则如下：

- 软件优先级：分为4级，低，中，高和极高。可以通过寄存器DMA_CHxCTL的PRIO位域来配置；
- 硬件优先级：当通道具有相同的软件优先级时，编号低的通道优先级高。例：通道0和通道2配置为相同的软件优先级时，通道0的优先级高于通道2。

10.4.4. 地址生成

存储器和外设都独立的支持两种地址生成算法：固定模式和增量模式。寄存器DMA_CHxCTL的PNAGA和MNAGA位分别用来设置存储器和外设的地址生成算法。

在固定模式中，地址一直固定为初始化的基地址（DMA_CHxPADDR，DMA_CHxMADDR）。

在增量模式中，下一次传输数据的地址是当前地址加1（或者2，4），这个值取决于数据传输宽度。

10.4.5. 循环模式

循环模式用来处理连续的外设请求(如ADC扫描模式)。将DMA_CHxCTL寄存器的CMEN位置位可以使能循环模式。

在循环模式中，当每次DMA传输完成后，CNT值会被重新载入，且传输完成标志位会被置1。DMA会一直响应外设的请求，直到通道使能位（DMA_CHxCTL寄存器的CHEN位）被清0。

10.4.6. 存储器到存储器模式

将DMA_CHxCTL寄存器的M2M位置位可以使能存储器到存储器模式。在此模式下，DMA通道传输数据时不依赖外设的请求信号。一旦DMA_CHxCTL寄存器的CHEN位被置1，DMA通道就立即开始传输数据，直到DMA_CHxCNT寄存器达到0，DMA传输才会停止。

10.4.7. 通道配置

要启动一次新的 DMA 数据传输，建议遵循以下步骤进行操作：

1. 读取 CHEN 位，如果为 1（通道已使能），清零该位。当 CHEN 为 0 时，请按照下列步骤配置 DMA 开始新的传输；
2. 配置 DMA_CHxCTL 寄存器的 M2M 及 DIR 位，选择传输模式；
3. 配置 DMA_CHxCTL 寄存器的 CMEN 位，选择是否使能循环模式；
4. 配置 DMA_CHxCTL 寄存器的 PRIO 位域，选择该通道的软件优先级；
5. 通过 DMA_CHxCTL 寄存器配置存储器和外设的传输宽度以及存储器和外设地址生成算法；
6. 通过 DMA_CHxCTL 寄存器配置传输完成中断，半传输完成中断，传输错误中断的使能位；
7. 通过 DMA_CHxPADDR 寄存器配置外设基地址；
8. 通过 DMA_CHxMADDR 寄存器配置存储器基地址；
9. 通过 DMA_CHxCNT 寄存器配置数据传输总量；
10. 将 DMA_CHxCTL 寄存器的 CHEN 位置 1，使能 DMA 通道。

10.4.8. 中断

每个DMA通道都有一个专用的中断。中断事件有三种类型：传输完成，半传输完成和传输错误。

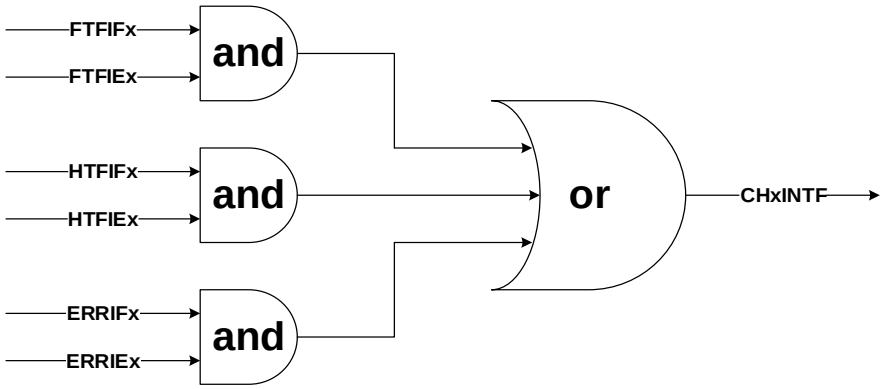
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。*表10-2. 中断事件*描述了其对应关系。

表 10-2. 中断事件

中断事件	标志位	清除位	使能位
	DMA_INTF	DMA_INTC	DMA_CHxCTL
传输完成	FTFIF	FTFIFC	FTFIE
传输半完成	HTFIF	HTFIFC	HTFIE
传输错误	ERRIF	ERRIFC	ERRIE

DMA中断逻辑如*图10-3. DMA中断逻辑图*所示，任何类型中断使能时，产生了相应中断事件均会产生中断。

图 10-3. DMA 中断逻辑图



注意: “x” 表示通道数（对应x=0…6）

10.4.9. DMA 请求映射

每个 DMA 通道的请求都连接至由 DMAMUX 请求复用器的对应通道输出来转发的 AHB/APB 外设请求，参考*表 11-2. DMAMUX 请求路由输入信号映射*。

10.5. DMA 寄存器

DMA 基地址：0x4002 0000

10.5.1. 中断标志位寄存器（DMA_INTF）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				ERRIF6	HTFIF6	FTFIF6	GIF6	ERRIF5	HTFIF5	FTFIF5	GIF5	ERRIF4	HTFIF4	FTFIF4	GIF4
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIF3	HTFIF3	FTFIF3	GIF3	ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	ERRIF0	HTFIF0	FTFIF0	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:28	保留	必须保持复位值。
27/23/19/15 /11/7/3	ERRIFx	通道x错误标志位(x=0…6) 硬件置位，软件写DMA_INTC相应位为1清零 0：通道x未发生传输错误 1：通道x发生传输错误
26/22/18/14 /10/6/2	HTFIFx	通道x半传输完成标志位(x=0…6) 硬件置位，软件写DMA_INTC相应位为1清零 0：通道x半传输未完成 1：通道x半传输完成
25/21/17/13 /9/5/1	FTFIFx	通道x传输完成标志位(x=0…6) 硬件置位，软件写DMA_INTC相应位为1清零 0：通道x传输未完成 1：通道x传输完成
24/20/16/12 /8/4/0	GIFx	通道x全局中断标志位(x=0…6) 硬件置位，软件写DMA_INTC相应位为1清零 0：通道x ERRIF，HTFIF或FTFIF标志位未置位 1：通道x至少发生ERRIF, HTFIF或FTFIF之一置位

10.5.2. 中断标志位清除寄存器（DMA_INTC）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				ERRIFC6	HTFIFC6	FTFIFC6	GIFC6	ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4
				w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIFC3	HTFIFC3	FTFIFC3	GIFC3	ERRIFC2	HTFIC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:28	保留	必须保持复位值。
27/23/19/15 /11/7/3	ERRIFCx	清除通道x(x=0…6)的错误标志位 0：无影响 1：清零DMA_INTF寄存器的ERRIFx位
26/22/18/14 /10/6/2	HTFIFCx	清除通道x(x=0…6)的半传输完成标志位 0：无影响 1：清零DMA_INTF寄存器的HTFIFx位
25/21/17/13 /9/5/1	FTFIFCx	清除通道x(x=0…6)的传输完成标志位 0：无影响 1：清零DMA_INTF寄存器的FTFIFx位
24/20/16/12 /8/4/0	GIFCx	清除通道x(x=0…6)的全局中断标志位 0：无影响 1：清零DMA_INTF寄存器的GIFx，ERRIFx，HTFIFx和FTFIFx位

10.5.3. 通道 x 控制寄存器（DMA_CHxCTL）

x = 0...6, x 为通道序号

地址偏移：0x08 + 0x14 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	M2M	PRIO[1:0]		MWIDTH[1:0]		PWIDTH[1:0]		MNAGA	PNAGA	CMEN	DIR	ERRIE	HTFIE	FTFIE	CHEN
	rw	rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	M2M	存储器到存储器模式 软件置位和清零 0：禁止存储器到存储器模式 1：使能存储器到存储器模式

		CHEN位为1时，该位不能被配置
13:12	PRIO[1:0]	软件优先级 软件置位和清零 00：低 01：中 10：高 11：极高 CHEN位为1时，该位域不能被配置
11:10	MWIDTH[1:0]	存储器的传输数据宽度 软件置位和清零 00：8-bit 01：16-bit 10：32-bit 11：保留 CHEN位为1时，该位域不能被配置
9:8	PWIDTH[1:0]	外设的传输数据宽度 软件置位和清零 00：8-bit 01：16-bit 10：32-bit 11：保留 CHEN位为1时，该位域不能被配置
7	MNAGA	存储器的地址生成算法 软件置位和清零 0：固定地址模式 1：增量地址模式 CHEN位为1时，该位不能被配置
6	PNAGA	外设的地址生成算法 软件置位和清零 0：固定地址模式 1：增量地址模式 CHEN位为1时，该位不能被配置
5	CMEN	循环模式使能 软件置位和清零 0：禁止循环模式 1：使能循环模式 CHEN位为1时，该位不能被配置
4	DIR	传输方向 软件置位和清零 0：从外设读出并写入存储器 1：从存储器读出并写入外设

		CHEN位为1时，该位不能被配置
3	ERRIE	通道错误中断使能位 软件置位和清零 0：禁止通道错误中断 1：使能通道错误中断
2	HTFIE	通道半传输完成中断使能位 软件置位和清零 0：禁止通道半传输完成中断 1：使能通道半传输完成中断
1	FTFIE	通道传输完成中断使能位 软件置位和清零 0：禁止通道传输完成中断 1：使能通道传输完成中断
0	CHEN	通道使能 软件置位和清零 0：禁止该通道 1：使能该通道

10.5.4. 通道 x 计数寄存器（DMA_CHxCNT）

x = 0...6, x 为通道序号

地址偏移：0x0C + 0x14 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	传输计数 CHEN位为1时，该位域不能被配置 该寄存器表明还有多少数据等待被传输。一旦通道使能，该寄存器为只读的，并在每个DMA传输之后值减1。如果该寄存器的值为0，无论通道开启与否，都不会有数据传输。如果该通道工作在循环模式下，一旦通道的传输任务完成，该寄存器会被自动重装载为初始设置值。

10.5.5. 通道 x 外设基地址寄存器（DMA_CHxPADDR）

x = 0...6, x 为通道序号

地址偏移：0x10 + 0x14 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

位/位域	名称	描述
31:0	PADDR[31:0]	外设基地址 CHEN位为1时，该位域不能被配置 当PWIDTH位域的值为01 (16-bit),，PADDR[0]被忽略，访问自动与16位地址对齐。 当PWIDTH位域的值为10 (32-bit)，PADDR [1:0]被忽略，访问自动与32位地址对齐。

10.5.6. 通道 x 存储器基地址寄存器（DMA_CHxMADDR）

x = 0...6, x 为通道序号

地址偏移：0x14 + 0x14 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

位/位域	名称	描述
31:0	MADDR[31:0]	存储器基地址 CHEN位为1时，该位域不能被配置 当MWIDTH位域的值为01 (16-bit)时，MADDR [0]被忽略，访问自动与16位地址对齐。 当MWIDTH位域的值为10 (32-bit)时，MADDR [1:0]被忽略，访问自动与32位地址对齐。

11. DMA 请求多路复用器(DMAMUX)

11.1. 简介

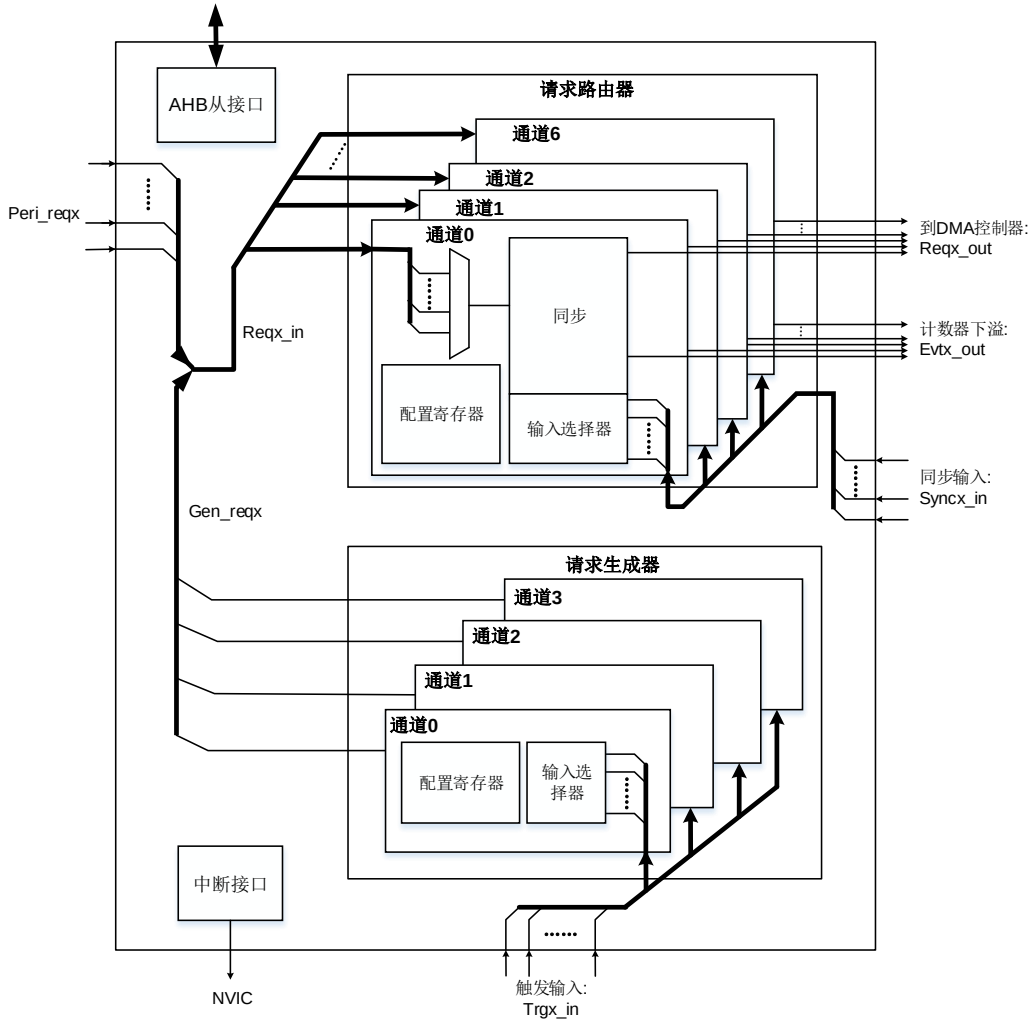
DMAMUX 是 DMA 请求的传输调度器。可编程的 DMA 请求多路复用器 DMAMUX，可在外设和 DMA 控制器之间路由 DMA 请求线路，或者 DMAMUX 也可以将可编程事件连入到输入触发信号上，作为一个 DMAMUX 请求发生器，再由 DMAMUX 请求路由器在 DMAMUX 请求生成器产生的 DMA 请求和 DMA 控制器之间路由 DMA 请求线路。每个 DMAMUX 请求路由通道选择一条唯一的 DMA 请求线路，无条件地或同步地从它的 DMAMUX 同步输入事件。DMA 请求信号会一直挂起，直到 DMA 控制器响应它，并且产生一个 DMA 确认信号，此时相应的 DMA 请求信号被释放。

11.2. 主要特征

- 7 个可配置的 DMAMUX 请求路由输出通道。
- 4 个 DMAMUX 请求生成通道。
- 21 路触发输入信号到 DMAMUX 请求生成器。
- 21 路同步输入信号。
- 每个 DMAMUX 请求生成通道包含一个 DMAMUX 请求触发输入选择器，一个 DMAMUX 请求生成计数器，和一个指示被选中的 DMAMUX 请求触发输入信号的事件溢出标志。
- 每个 DMAMUX 请求路由输出通道包含 41 路外设 DMAMUX 请求输入信号，一个同步输入信号选择器，一条 DMA 请求路由输出线路，一个路由事件输出信号用于 DMA 请求级联，一个 DMAMUX 请求路由计数器，和一个指示被选中的同步输入信号的事件溢出标志。

11.3. 结构框图

图 11-1. DMAMUX 结构框图



11.4. 信号描述

DMAMUX 信号描述如下所示：

- Reqx_in：DMAMUX 请求路由输入信号，来自外设的请求或者 DMAMUX 请求生成器生成的请求。
- Peri_reqx：从外设输入到 DMAMUX 的 DMA 请求线路。
- Gen_reqx：DMAMUX 请求生成器生成输出的 DMA 请求信号。
- Reqx_out：DMAMUX 请求输出信号到 DMA 控制器。
- Trgx_in：DMAMUX 请求触发输入信号到 DMAMUX 请求生成器。
- Syncx_in：DMAMUX 同步输入信号到 DMAMUX 请求路由器。
- Evtx_out：DMAMUX 请求路由计数器下溢事件输出信号。

11.5. 功能说明

如*图 11-1. DMAMUX 结构框图*所示，DMAMUX 包含两个子模块：

- DMAMUX 请求路由器
  DMAMUX 请求路由器输入（Reqx_in）来自两部分：
  – 一部分来自外设请求（Peri_reqx）；
  – 另一部分来自 DMAMUX 请求生成器（Gen_reqx）。
  DMAMUX 请求路由输出到 DMA 控制器对应的通道（Reqx_out）。
  同步输入（Syncx_in）来自内部或外部信号。
- DMAMUX 请求生成器
  DMAMUX 请求触发输入（Trgx_in）来自内部或外部信号。

11.5.1. DMAMUX 请求路由器

DMAMUX 请求路由器可在外设/ DMAMUX 请求生成器，与 DMA 控制器之间路由 DMA 请求线路。DMAMUX 请求路由器由 DMAMUX 请求路由通道组成。DMA 请求输入信号并联至所有的 DMAMUX 请求路由通道。每个 DMAMUX 请求路由通道都有一个同步单元。同步输入信号并联至所有 DMAMUX 请求路由通道的同步单元。每个 DMAMUX 请求路由通道都有一个内部的 DMAMUX 请求路由计数器。

DMAMUX 请求路由通道

DMAMUX 请求路由通道 x 的请求路由输入由 DMAMUX_RM_CHxCFG 寄存器的 MUXID[5:0]位域来配置，请求路由输入可选为外设 DMA 请求，或者 DMAMUX 请求生成器产生的 DMA 请求，参考*表 11-2. DMAMUX 请求路由输入信号映射*。一个 DMAMUX 请求路由通道与对应的 DMA 控制器通道相连接。

注意： 当 MUXID[5:0]值为 0 时，没有 DMA 请求线路被映射到 DMAMUX 请求路由通道上。DMAMUX 不允许将同一个 DMA 请求线路（相同 MUXID[5:0]且非空）映射到两个不同的 DMAMUX 请求路由通道上。

当同步模式禁能时

每当连到 DMAMUX 的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求路由计数器将减 1。当 DMAMUX 请求路由计数器发生下溢时，DMAMUX_RM_CHxCFG 寄存器的 NBR[4:0]值将自动重装载到计数器中。如果将 EVGEN 位置位，使能通道事件输出，则通道事件输出前，DMA 请求数量为 NBR[4:0] + 1。

注意：只有当 DMAMUX 请求路由通道 x 的同步使能位 SYNCEN 位和通道事件输出使能位 EVGEN 位都为 0 时，才能配置其 NBR[4:0]位域。

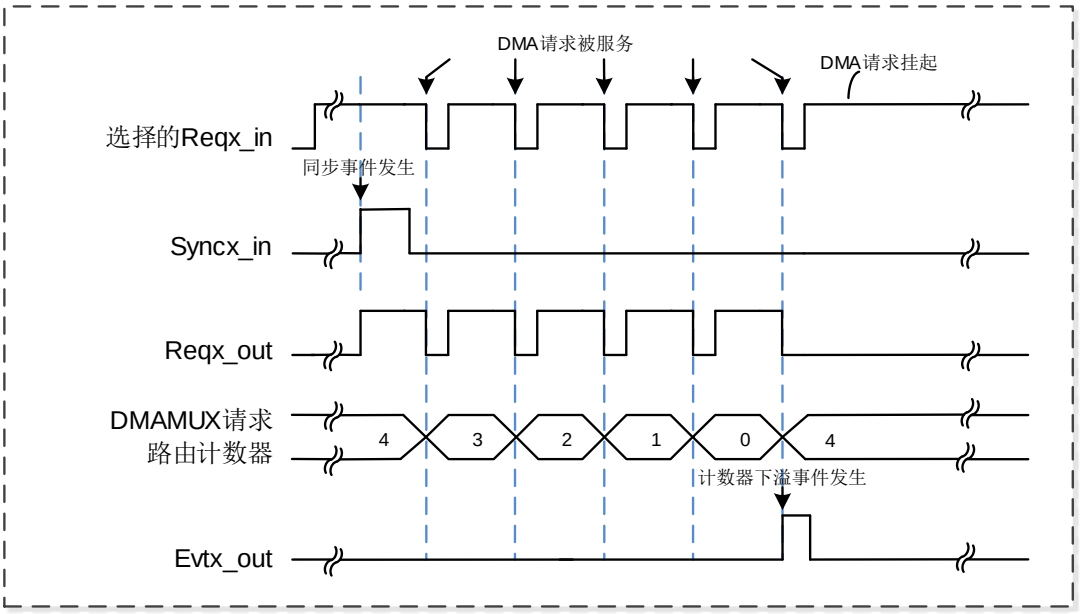
当同步模式使能时

如果 DMAMUX 请求路由通道 x 工作在同步模式下，当检测到选择的同步输入信号的上升沿或者下降沿时，挂起的 DMA 请求将被连到 DMAMUX 请求路由通道 x 的输出。每当连到 DMAMUX 的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求路由计

数器将减 1。当 DMAMUX 请求路由计数器发生下溢时，DMA 请求线路将断开与 DMAMUX 请求路由通道 x 的输出的连接，并且 DMAMUX_RM_CHxCFG 寄存器的 NBR[4:0]值将自动重装载到计数器中。一个同步事件可传输 NBR[4:0] + 1 个 DMA 请求到 DMAMUX 请求路由通道 x 的输出上。

*图 11-2. 同步模式*为当 NBR[4:0]=4，SYNCEN=1，EVGEN=1，SYNCP[1:0]=0b01 时的举例。

图 11-2. 同步模式



置位 DMAMUX_RM_CHxCFG 寄存器的 SYNCEN 位可使能 DMAMUX 请求路由通道 x 的同步模式。同步输入信号可由 DMAMUX_RM_CHxCFG 寄存器的 SYNCID[4:0]位域来配置，参考*表 11-4. 同步输入信号映射*。 同步输入信号的有效边沿由 DMAMUX_RM_CHxCFG 寄存器的 SYNCP[1:0]位域来配置。

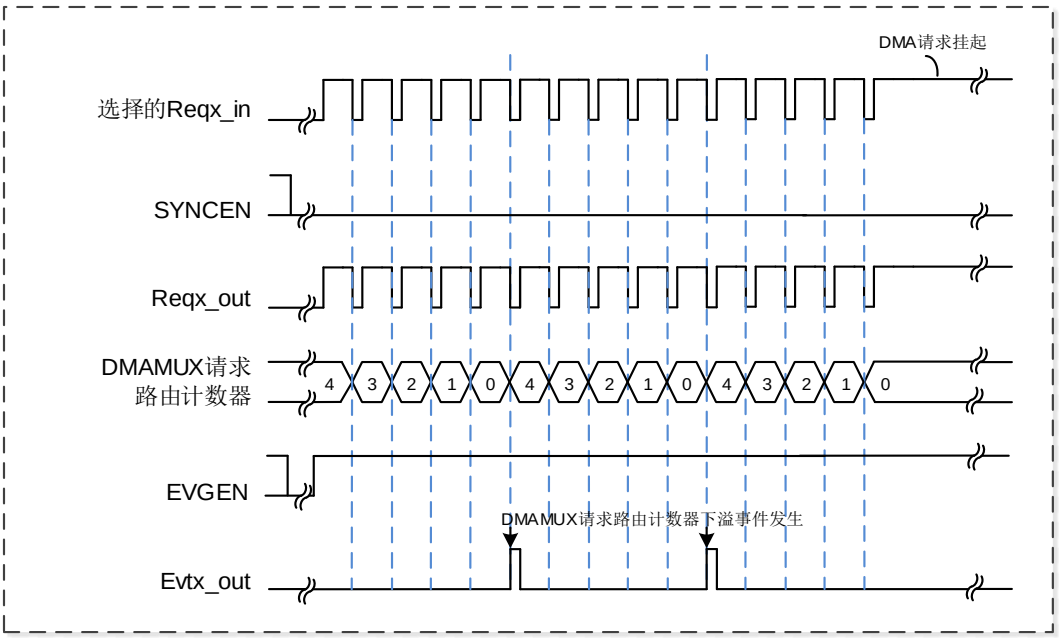
注意： 如果同步输入事件发生时，DMAMUX 输入上没有挂起的 DMA 请求，则这个同步输入事件将被忽略，之后如有 DMA 请求被挂起，它将不会被连接到 DMAMUX 请求路由通道 x 的输出，直到发生下一个同步输入事件。

通道事件输出

每个 DMAMUX 请求路由通道都有一个通道事件输出信号 Evtx_out，用于 DMAMUX 请求路由计数器的下溢事件输出。Evt0_out ~ Evt3_out 信号可用于 DMA 请求级联。如果通过置位 DMAMUX_RM_CHxCFG 寄存器的 EVGEN 位来使能 DMAMUX 请求路由通道 x 的通道事件输出，当 DMAMUX 请求路由计数器自动重装载为 NBR[4:0]值时，发生一个通道事件，输出为一个 AHB 时钟周期脉冲。

*图 11-3. 通道事件输出*为当 NBR[4:0]=4，SYNCEN=0，EVGEN=1 时的举例。

图 11-3. 通道事件输出



注意：如果 EVGEN = 1 且 NBR[4:0] = 0，则每次 DMA 请求被服务时都会输出一个通道事件。

同步溢出

如果在 DMAMUX 请求路由计数器下溢之前又发生了新的同步事件，则 DMAMUX_RM_INTF 寄存器的同步溢出标志位 SOIFx 位将置位。

注意：建议在 DMA 控制器对应通道请求被取消时，配置 DMAMUX_RM_CHxCFG 寄存器的 SYNCEN 位为 0 来禁能 DMAMUX 请求路由通道 x 的同步模式。否则，当又发生一个新的同步事件时，由于接收不到 DMA 的响应信号将会发生同步溢出事件。

11.5.2. DMAMUX 请求生成器

DMAMUX 请求生成器在触发输入事件发生时会产生 DMA 请求。DMAMUX 请求生成器由 DMAMUX 请求生成通道组成。DMA 请求触发输入信号并联至所有 DMAMUX 请求生成通道。每个 DMAMUX 请求生成通道都有一个内部的 DMAMUX 请求生成计数器。

触发输入信号的有效边沿由 DMAMUX_RG_CHxCFG 寄存器的 RGTP[1:0]位域来配置。DMAMUX 请求生成通道 x 的触发输入信号由 DMAMUX_RG_CHxCFG 寄存器的 TID[4:0]位域来配置，参考*表 11-3. 触发输入信号映射*。置位 DMAMUX_RG_CHxCFG 寄存器的 RGEN 位来使能 DMAMUX 请求生成通道 x。

DMAMUX 请求生成通道

当发生触发输入事件时，对应的 DMAMUX 请求生成通道 x 开始产生 DMA 请求到通道的输出上，通道输出连到 DMAMUX 请求路由器的输入上。每当 DMAMUX 生成的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求生成计数器将减 1。当 DMAMUX 请求生成计数器发生下溢时，DMAMUX 请求生成通道将停止产生 DMA 请求，在下

一个触发输入事件发生时，DMAMUX 请求生成计数器将自动重装载为 DMAMUX_RG_CHxCFG 寄存器的 NBRG[4:0]位域值。

注意：触发输入事件后产生的 DMA 请求数量为 NBRG[4:0] + 1。只有当 DMAMUX 请求生成通道 x 的 RGEN 位为 0 时才可以配置 NBRG[4:0]位域。

触发溢出

如果 RGEN 位为 1，DMAMUX 请求生成通道 x 被使能，当一个新的触发输入信号发生了，而此时 DMAMUX 请求生成计数器还未发生下溢，则 DMAMUX_RG_INTF 寄存器的 TOIFx 位将硬件置位以指示发生了触发溢出事件。

注意：建议在 DMA 控制器对应通道请求被取消时，配置 DMAMUX_RG_CHxCFG 寄存器的 RGEN 位为 0 来禁能 DMAMUX 请求生成通道 x。否则，当又发生一个新的触发输入事件时，由于接收不到 DMA 的响应信号将会发生触发溢出事件。

11.5.3. 通道配置

根据以下步骤来配置 DMAMUX 的通道 y 和对应的 DMA 通道 x：

1. 完整配置 DMA 通道 x 相关参数，除了 DMA 通道 x 的使能。
2. 完整配置 DMAMUX 通道 y 相关参数。
3. 设置 DMA_CHxCTL 寄存器的 CHEN 位 1 来使能 DMA 通道 x。

11.5.4. 中断

DMAMUX模块有两种类型的中断事件，包括DMAMUX请求路由通道的同步溢出事件，和DMAMUX请求生成通道的触发溢出事件。

每个中断事件都有一个专用的标志位，专用的清除位和专用的使能位。*表11-1. 中断事件*描述了其对应关系。

表 11-1. 中断事件

中断事件	标志位	清除位	使能位
DMAMUX 请求路由通道 x 上的同步溢出事件	DMAMUX_RM_INTF 寄存器的 SOIFx 位	DMAMUX_RM_INTC 寄存器的 SOIFCx 位	DMAMUX_RM_CHxCFG 寄存器的 SOIE 位
DMAMUX 请求生成通道 y 上的触发溢出事件	DMAMUX_RG_INTF 寄存器的 TOIFy 位	DMAMUX_RG_INTC 寄存器的 TOIFCy 位	DMAMUX_RG_CHxCFG 寄存器的 TOIE 位

触发溢出中断

当 DMAMUX 请求生成触发溢出标志位 TOIFx 置位，并且触发溢出中断使能位 TOIE 位置位，则会产生一个触发溢出中断。写 1 到 DMAMUX_RG_INTC 寄存器的对应触发溢出清除位 TOIFCx 将会清除触发溢出标志位 TOIFx。

同步溢出中断

当 DMAMUX 请求路由同步溢出标志位 SOIFx 置位，并且触发同步溢出中断使能位 SOIE 位置位，则会产生一个同步溢出中断。写 1 到 DMAMUX_RM_INTC 寄存器的对应同步溢出清除位 SOIFCx 将会清除同步溢出标志位 SOIFx。

11.5.5. DMAMUX 映射

DMAMUX 请求路由输入映射

DMAMUX 请求路由输入可来自于外设或者 DMAMUX 请求生成器，参考*表 11-2. DMAMUX 请求路由输入信号映射*，由 DMAMUX_RM_CHxCFG 寄存器的 MUXID[5:0]位域配置 DMAMUX 请求路由通道 x 的输入。

表 11-2. DMAMUX 请求路由输入信号映射

请求路由通道输入标识 MUXID[5:0]	来源
1	Gen_req0
2	Gen_req1
3	Gen_req2
4	Gen_req3
5	ADC
6	DAC
7	保留
8	保留
9	保留
10	I2C0_RX
11	I2C0_TX
12	I2C1_RX
13	I2C1_TX
14	I2C2_RX
15	I2C2_TX
16	SPI0_RX
17	SPI0_TX
18	SPI1_RX
19	SPI1_TX
20	保留
21	保留
22	保留
23	保留
24	保留
25	TIMER1_CH0
26	TIMER1_CH1
27	TIMER1_CH2
28	TIMER1_CH3
29	保留

请求路由通道输入标识 MUXID[5:0]	来源
30	TIMER1_UP
31	保留
32	TIMER2_CH0
33	TIMER2_CH1
34	TIMER2_CH2
35	TIMER2_CH3
36	TIMER2_TRIG
37	TIMER2_UP
38	保留
39	保留
40	保留
41	保留
42	TIMER5_UP
43	TIMER6_UP
44	CAU_IN
45	CAU_OUT
46	保留
47	保留
48	保留
49	保留
50	USART0_RX
51	USART0_TX
52	USART1_RX
53	USART1_TX
54	UART3_RX
55	UART3_TX
56	UART4_RX
57	UART4_TX
58	LPUART_RX
59	LPUART_TX
60	保留
61	保留
62	保留
63	保留

触发输入映射

DMAMUX 请求生成通道 x 的触发输入可由 DMAMUX_RG_CHxCFG 寄存器的 TID[4:0]位域来配置，参考*表 11-3. 触发输入信号映射*。

表 11-3. 触发输入信号映射

触发输入标识TID[4:0]	来源
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evt0_out
17	Evt1_out
18	Evt2_out
19	Evt3_out
20	保留
21	保留
22	TIMER11_CH0_O
23	保留

同步输入映射

同步输入由 DMAMUX_RM_CHxCFG 寄存器的 SYNCID[4:0]位域来配置，参考*表 11-4. 同步输入信号映射*。

表 11-4. 同步输入信号映射

同步输入标识SYNCID[4:0]	来源
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9

同步输入标识SYNCID[4:0]	来源
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evt0_out
17	Evt1_out
18	Evt2_out
19	Evt3_out
20	保留
21	保留
22	TIMER11_CH0_O
23	保留

11.6. DMAMUX 寄存器

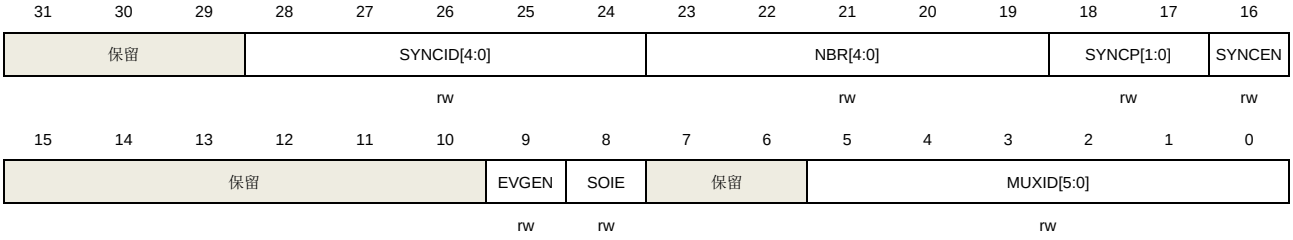
DMAMUX基地址：0x4002 0800

11.6.1. 请求路由通道 x 配置寄存器（DMAMUX_RM_CHxCFG）

x = 0...6，其中 x 为通道序号

地址偏移：0x00 + 0x04 * x
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:29	保留	必须保持复位值。
28:24	SYNCID[4:0]	同步输入标识 选择同步输入源。
23:19	NBR[4:0]	传递的DMA请求数量 在同步输入事件之后，或者通道事件输出之前，将传递到DMA控制器的DMA请求数量为NBR[4:0] + 1。 该位域只能在SYNCEN位和EVGEN位都禁能时才能配置。
18:17	SYNCP[1:0]	同步输入极性 00：不检测事件 01：上升沿 10：下降沿 11：上升和下降沿
16	SYNCEN	同步模式使能 0：禁能同步模式 1：使能同步模式
15:10	保留	必须保持复位值。
9	EVGEN	事件输出使能 0：禁能事件输出 1：使能事件输出
8	SOIE	同步溢出中断使能 0：禁能中断

		1：使能中断
7:6	保留	必须保持复位值。
5:0	MUXID[5:0]	请求路由标识 选择DMAMUX请求路由通道的DMA请求输入源。

11.6.2. 请求路由通道中断标志位寄存器（DMAMUX_RM_INTF）

地址偏移：0x80
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									SOIF6	SOIF5	SOIF4	SOIF3	SOIF2	SOIF1	SOIF0
									r	r	r	r	r	r	r

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	SOIF6	请求路由通道6的同步溢出事件标志位 参照SOIF0的描述。
5	SOIF5	请求路由通道5的同步溢出事件标志位 参照SOIF0的描述。
4	SOIF4	请求路由通道4的同步溢出事件标志位 参照SOIF0的描述。
3	SOIF3	请求路由通道3的同步溢出事件标志位 参照SOIF0的描述。
2	SOIF2	请求路由通道2的同步溢出事件标志位 参照SOIF0的描述。
1	SOIF1	请求路由通道1的同步溢出事件标志位 参照SOIF0的描述。
0	SOIF0	请求路由通道0的同步溢出事件标志位 如果同步输入事件发生时，DMAMUX请求路由计数器值小于NBR[4:0]，则该位置位。 通过对DMAMUX_RM_INTC寄存器的SOIFC0位写1来清除相应通道的同步溢出标志。

11.6.3. 请求路由通道中断标志位清除寄存器（DMAMUX_RM_INTC）

地址偏移：0x084

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									SOIFC6	SOIFC5	SOIFC4	SOIFC3	SOIFC2	SOIFC1	SOIFC0
									w	w	w	w	w	w	w

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	SOIFC6	请求路由通道6的同步溢出事件标志清除位 参照SOIFC0的描述。
5	SOIFC5	请求路由通道5的同步溢出事件标志清除位 参照SOIFC0的描述。
4	SOIFC4	请求路由通道4的同步溢出事件标志清除位 参照SOIFC0的描述。
3	SOIFC3	请求路由通道3的同步溢出事件标志清除位 参照SOIFC0的描述。
2	SOIFC2	请求路由通道2的同步溢出事件标志清除位 参照SOIFC0的描述。
1	SOIFC1	请求路由通道1的同步溢出事件标志清除位 参照SOIFC0的描述。
0	SOIFC0	请求路由通道0的同步溢出事件标志清除位 写1可清除相应通道在DMAMUX_RM_INTF寄存器的同步溢出标志SOIF0。

11.6.4. 请求生成通道 x 配置寄存器（DMAMUX_RG_CHxCFG）

x = 0...3，其中 x 为通道序号

地址偏移：0x100 + 0x04 * x
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								NBRG[4:0]					RGTP[1:0]		RGEN
										rw			rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							TOIE	保留			TID[4:0]				
							rw				rw				

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:19	NBRG[4:0]	待产生的DMA请求数量 在触发输入事件之后，待产生的DMA请求数量为NBRG[4:0] + 1。 **注意：**只有当RGEN位为0时才能写该位域。
18:17	RGTP[1:0]	DMAMUX请求生成触发输入极性 00：不检测事件 01：上升沿 10：下降沿 11：上升沿和下降沿
16	RGEN	DMAMUX请求生成通道x使能 0：禁能DMAMUX请求生成通道x 1：使能DMAMUX请求生成通道x
15:9	保留	必须保持复位值。
8	TOIE	触发溢出中断使能 0：禁能中断 1：使能中断
7:5	保留	必须保持复位值。
4:0	TID[4:0]	触发输入标识 选择DMAMUX请求生成通道的触发输入源。

11.6.5. 请求生成通道中断标志位寄存器（DMAMUX_RG_INTF）

地址偏移：0x140
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												TOIF3	TOIF2	TOIF1	TOIF0
												r	r	r	r

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	TOIF3	DMAMUX请求生成通道3的触发溢出标志位 参照TOIF0的描述。
2	TOIF2	DMAMUX请求生成通道2的触发溢出标志位

参照TOIF0的描述。

1	TOIF1	DMAMUX请求生成通道1的触发溢出标志位 参照TOIF0的描述。
0	TOIF0	DMAMUX请求生成通道0的触发溢出标志位 如果触发输入事件在DMAMUX请求生成计数器下溢之前发生，则该位置位。 通过对DMAMUX_RG_INTC寄存器的TOIFC0位写1来清除相应通道的触发溢出标志。

11.6.6. 请求生成通道中断标志位清除寄存器（DMAMUX_RG_INTC）

地址偏移：0x144
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												TOIFC3	TOIFC2	TOIFC1	TOIFC0
												w	w	w	w

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	TOIFC3	DMAMUX请求生成通道3的触发溢出标志清除位 参照TOIFC0的描述。
2	TOIFC2	DMAMUX请求生成通道2的触发溢出标志清除位 参照TOIFC0的描述。
1	TOIFC1	DMAMUX请求生成通道1的触发溢出标志清除位 参照TOIFC0的描述。
0	TOIFC0	DMAMUX请求生成通道0的触发溢出标志清除位 写1可清除相应通道在DMAMUX_RG_INTF寄存器的触发溢出标志TOIF0。

12. 调试（DBG）

12.1. 简介

GD32L23x系列产品提供了各种各样的调试、跟踪和测试功能。这些功能通过ARM CoreSight组件的标准配置和链状连接的TAP控制器来实现的。调试功能集成在ARM Cortex-M23内核中。调试系统支持串行调试（SWD）和跟踪功能。调试和跟踪功能请参考下列文档：

- Cortex-M23技术参考手册；
- ARM调试接口v5结构规范。

调试系统可以帮助调试者在低功耗模式下调试以及进行一些外设的调试，包括：TIMER、LPTIMER、I2C、RTC、WWDGT和FWDGT。当相应的位被置1，调试系统会在低功耗模式下提供时钟，或者为一些外设保持当前状态，这些外设包括：TIMER、LPTIMER、I2C、RTC、WWDGT和FWDGT。

12.2. 串行调试接口简介

调试工具可以通过串行调试接口（SWD）来访问调试功能。

12.2.1. 引脚分配

串行调试（SWD）提供两个引脚的接口：数据输入输出引脚（SWDIO）和时钟引脚（SWCLK）。

调试引脚分配：

PA14 ： SWCLK

PA13 ： SWDIO

如果 SWD 调试功能没有使用，这两个引脚均释放作为普通 GPIO 功能。两个引脚具体配置请参考*通用和备用输入/输出接口（GPIO 和 AFIO）*。

12.3. 调试保持功能描述

12.3.1. 低功耗模式调试支持

当DBG控制寄存器0（DBG_CTL0）的STB_HOLD位置1并且进入待机模式，AHB总线时钟和系统时钟由CK_IRC16M提供，可以在待机模式下调试。当退出待机模式后，产生系统复位。

当DBG控制寄存器0（DBG_CTL0）的DSLP_HOLD位置1并且进入深度睡眠模式，AHB总线时钟和系统时钟由CK_IRC16M提供，可以在深度睡眠模式下调试。

当DBG控制寄存器0（DBG_CTL0）的SLP_HOLD位置1并且进入睡眠模式，AHB总线时钟没有关闭，可以在睡眠模式下调试。

12.3.2. TIMER，LPTIMER，I2C，RTC，WWDGT 和 FWDGT 外设调试支持

当内核停止，并且DBG控制寄存器0（DBG_CTL0）或DBG控制寄存器1（DBG_CTL1）中的相应位置1。对于不同外设，有不同动作：

对于TIMER和LPTIMER外设，TIMER计数器停止并进行调试；

对于I2C外设，SMBUS保持状态并进行调试；

对于RTC外设，计数器停止并进行调试；

对于WWDGT或者FWDGT外设，计数器时钟停止并进行调试。

12.4. DBG 寄存器

DBG基地址：0x4001 5800

12.4.1. ID 寄存器（DBG_ID）

地址偏移：0x00
只读寄存器

该寄存器只能按字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID_CODE[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID_CODE[15:0]															
r															

位/位域	名称	描述
31:0	ID_CODE[31:0]	DBG ID 寄存器 这些位由软件读取，这些位是不变的常数

12.4.2. 控制寄存器 0（DBG_CTL0）

地址偏移：0x04
复位值：0x0000 0000，仅上电复位

该寄存器只能按字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留.					TIMER11_HOLD	保留.		TIMER8_HOLD	保留	TIMER6_HOLD	TIMER5_HOLD	保留			I2C1_HOLD
					rw			rw		rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C0_HOLD	保留	TIMER2_HOLD	TIMER1_HOLD	保留		WWDGT_HOLD	FWDGT_HOLD	保留					STB_HOLD	DSLP_HOLD	SLP_HOLD
rw		rw	rw			rw	rw						rw	rw	rw

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	TIMER11_HOLD	TIMER11 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 TIMER11 计数器不变，用于调试
25:24	保留	必须保持复位值。
23	TIMER8_HOLD	TIMER8 保持位 该位由软件置位和复位 0：无影响

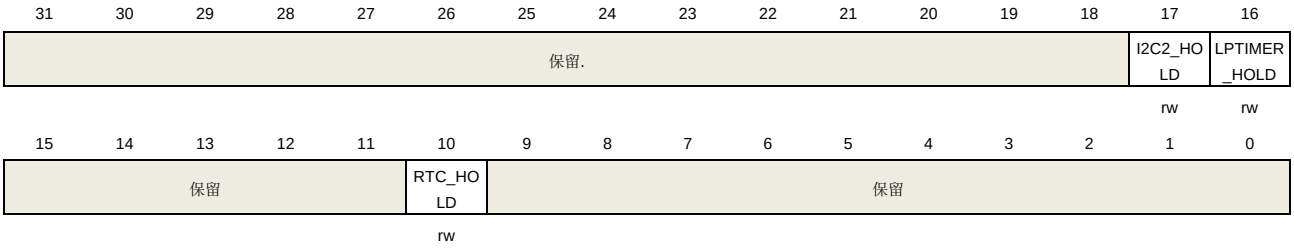
		1：当内核停止时保持 TIMER8 计数器不变，用于调试
22	保留	必须保持复位值。
21	TIMER6_HOLD	TIMER6 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 TIMER6 计数器不变，用于调试
20	TIMER5_HOLD	TIMER5 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 TIMER5 计数器不变，用于调试
19:17	保留	必须保持复位值。
16	I2C1_HOLD	I2C1 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 I2C1 的 SMBUS 状态不变，用于调试
15	I2C0_HOLD	I2C0 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 I2C0 的 SMBUS 状态不变，用于调试
14	保留	必须保持复位值。
13	TIMER2_HOLD	TIMER2 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 TIMER2 计数器不变，用于调试
12	TIMER1_HOLD	TIMER1 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 TIMER1 计数器不变，用于调试
11:10	保留	必须保持复位值。
9	WWDGT_HOLD	WWDGT 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 WWDGT 计数器时钟，用于调试
8	FWDGT_HOLD	FWDGT 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 FWDGT 计数器时钟，用于调试

7:3	保留	必须保持复位值。
2	STB_HOLD	待机模式保持位 该位由软件置位和复位 0：无影响 1：在待机模式下系统时钟和 AHB 时钟由 CK_IRC16M 提供，当退出待机模式时，产生系统复位
1	DSLP_HOLD	深度睡眠模式保持位 该位由软件置位和复位 0：无影响 1：在待机模式下系统时钟和 AHB 时钟由 CK_IRC16M 提供，当退出待机模式时，产生系统复位
0	SLP_HOLD	睡眠模式保持位 该位由软件置位和复位 0：无影响 1：在睡眠模式下, AHB 时钟继续运行

12.4.3. 控制寄存器 1（DBG_CTL1）

地址偏移：0x08
复位值：0x0000 0000，仅上电复位

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:19	保留	必须保持复位值。
18	I2C2_HOLD	I2C2 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 I2C2 的 SMBUS 状态不变，用于调试
16	LPTIMER_HOLD	LPTIMER 保持位 该位由软件置位和复位 0：无影响 1：当内核停止时保持 LPTIMER 计数器不变，用于调试
15:11	保留	必须保持复位值。
10	RTC_HOLD	RTC 保持位

该位由软件置位和复位
0：无影响
1：当内核停止时保持 RTC 计数器不变，用于调试

9:0	保留	必须保持复位值。

13. 模数转换器（ADC）

13.1. 简介

MCU片上集成了12位逐次逼近式模数转换器模块（ADC），可以采样来自于16个外部通道和4个内部通道上的模拟信号。这20个ADC采样通道都支持多种运行模式，采样转换后，转换结果可以按照最低有效位对齐或最高有效位对齐的方式保存在相应的数据寄存器中。片上的硬件过采样机制可以通过减少来自MCU的相关计算负担来提高性能。

13.2. 主要特征

- 高性能：
  - ADC 采样分辨率：12 位、10 位、8 位和 6 位分辨率；
  - 前置校准功能；
  - 可编程的采样时间；
  - 数据存储模式：最高有效位对齐和最低有效位对齐；
  - DMA 请求。
- 双时钟域架构（APB 时钟和 ADC 时钟）。
- 模拟输入通道：
  - 16 个外部模拟输入通道；
  - 1 个内部温度传感通道（V_{SENSE}）；
  - 1 个内部参考电压输入通道（V_{REFINT}）；
  - 1 个监测外部 V_{BAT} 引脚的内部输入通道（V_{BAT}）；
  - 1 个监测 SLCD 电压的内部输入通道（V_{SLCD}）。
- 转换开始的发起：
  - 软件触发；
  - 硬件触发。
- 运行模式：
  - 转换单个通道，或者扫描一组通道；
  - 单次运行模式，每次触发转换一次选择的输入通道；
  - 连续运行模式，连续转换所选择的输入通道；
  - 间断运行模式。
- 中断的产生：
  - 常规序列转换结束；
  - 模拟看门狗事件。
- 模转换结果阈值监测器功能：模拟看门狗。
- 模块供电要求：1.8V 到 3.6V，一般电源电压为 3.3V。
- 过采样：
  - 16 位的数据寄存器；
  - 可调整的过采样率，范围从 2x 到 256x；
  - 高达 8 位的可编程数据移位。
- 通道输入范围：V_{SSA} /V_{SS} ≤V_{IN} ≤V_{DDA}/V_{DD}。

13.3. 引脚和内部信号

*图 13-1. ADC 模块框图*给出了 ADC 模块框图。*表 13-1. ADC 内部输入信号*和*表 13-2. ADC 输入引脚定义* 给出了 ADC 内部信号和引脚定义。

表 13-1. ADC 内部输入信号

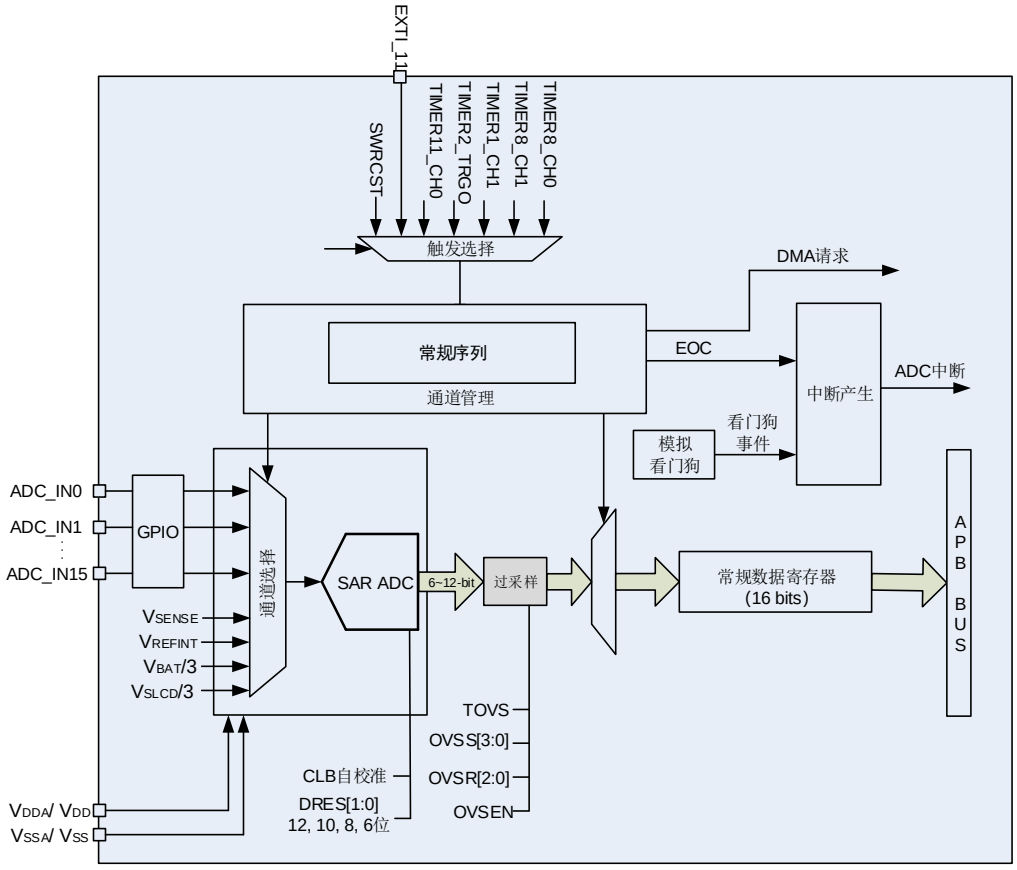
内部信号名称	说明
V_{SENSE}	内部温度传感器电压输出
V_{REFINT}	内部参考电压输出
V_{BAT}	V_{BAT} 引脚上电压除以 3
V_{SLCD}	V_{SLCD} 引脚上电压除以 3

表 13-2. ADC 输入引脚定义

名称	注释
V_{DDA} /V_{DD}	模拟电源输入等于V_{DD}，1.8 V ≤V_{DDA}≤ 3.6 V
V_{SSA} /V_{SS}	模拟地，等于 V_{SS}
ADCx_IN [15:0]	多达 16 路外部通道

13.4. 功能说明

图 13-1. ADC 模块框图



13.4.1. 前置校准功能

在前置校准期间，ADC计算一个校准因子，这个因子应用于ADC的内部，它直到ADC下次掉电才无效。在校准期间，应用程序不能使用ADC，必须等到校准完成。在开始A/D转换前应执行校准操作。通过软件设置CLB=1启动校准。在校准期间CLB位会一直保持1。一旦校准完成，该位由硬件清0。

当ADC运行条件改变（例如供电电压V_{DDA}、温度等）时，建议重新执行一次校准操作。

内部的模拟校准可以通过设置ADC_CTL1寄存器的RSTCLB位来重置。

软件校准过程：

1. 确保 ADCON=1；
2. 延迟 14 个 CK_ADC 以等待 ADC 稳定；
3. 设置 RSTCLB（该步骤是可选的）；
4. 设置 CLB=1；
5. 等待直到 CLB =0。

13.4.2. 双时钟域架构

除了APB接口时钟，ADC的子模块时钟还可以由ADC时钟提供。ADC时钟和APB时钟异步，并独立于APB时钟。

应用程序能够在低功耗运行时，降低PLCK时钟频率，同时ADC仍能保持最佳运行状态。

想要更多ADC时钟产生的信息，可以参考RCU章节的*4.2.1*部分。

13.4.3. ADCON 使能

ADC_CTL1寄存器中的ADCON位是ADC模块的使能开关。如果该位为0，则ADC模块保持复位状态。为了省电，当ADCON位为0时，ADC模拟子模块将会进入掉电模式。ADC使能后需等待t_{US}时间后才能采样，t_{US}数值详见芯片数据手册。

注意： 当ADC使用内部基准VREF（VREFEN=1）时，在使能ADC之前，请确保SYSCFG_VREF_CS寄存器中的VREFDY位置1。

13.4.4. 常规序列

通道管理电路可以将采样通道组织成一个序列：常规序列。常规序列支持最多16个通道，每个通道成为常规通道。

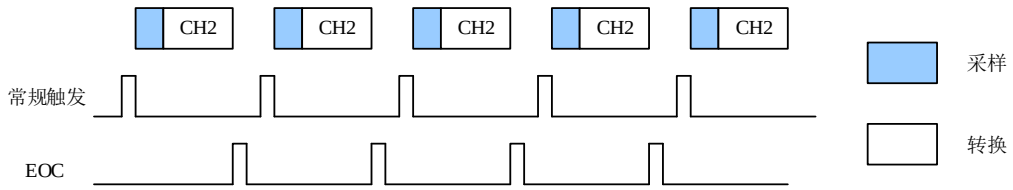
ADC_RSQ0寄存器的RL[3:0]位规定了整个常规序列的长度。ADC_RSQ0~ADC_RSQ2寄存器规定了常规序列的通道选择。

13.4.5. 运行模式

单次运行模式

单次运行模式下，ADC_RSQ2寄存器的RSQ0[4:0]位规定了ADC的转换通道。当ADCON位被置1时，一旦相应软件触发或者外部触发发生，ADC就会采样和转换一个通道。

图 13-2. 单次运行模式



常规通道单次转换结束后，转换数据将被存放于ADC_RDATA寄存器中，EOC将会置1。如果EOCIE位被置1，将产生一个中断。

常规组单次运行模式的软件流程：

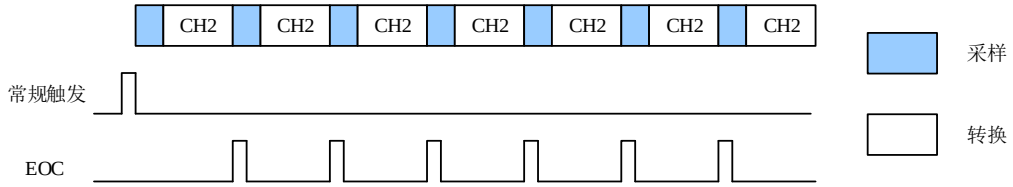
1. 确保ADC_CTL0寄存器的DISRC位和SM位以及ADC_CTL1寄存器中的CTN位为0；
2. 用模拟通道编号来配置RSQ0；
3. 配置ADC_SAMPTx寄存器；

4. 如果有需要，可以配置ADC_CTL1寄存器中的ETERC位和ETSRC位；
5. 设置SWRCST位，或者为常规序列产生一个外部触发信号；
6. 等到EOC置1；
7. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位。

连续运行模式

对ADC_CTL1寄存器中的CTN位置1，可以使能连续运行模式。在此模式下，ADC执行由RSQ0[4:0]规定的转换通道。当ADCON位被置1，一旦相应软件触发或者外部触发产生，ADC就会采样和转换规定的通道。转换数据保存在ADC_RDATA寄存器中。

图 13-3. 连续运行模式



常规序列连续运行模式的软件流程：

1. 设置ADC_CTL1寄存器中的CTN位为1；
2. 根据模拟通道编号来配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 设置SWRCST位，或者给常规序列产生一个外部触发信号；
6. 等待EOC标志位置1；
7. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位；
9. 如果需要进行连续转换，重复步骤6~8。

可以使用DMA来传输转换数据，不需循环查询EOC标志位：

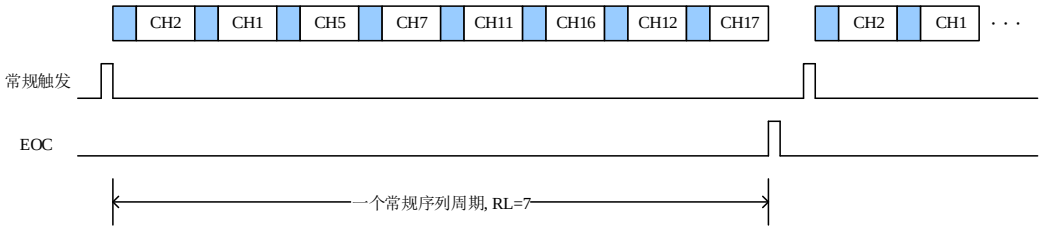
1. 设置ADC_CTL1寄存器的CTN位为1；
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETERC位和ETSRC位；
5. 准备DMA模块，用于传输来自ADC_RDATA寄存器的数据；
6. 设置SWRCST位，或者给常规序列产生一个外部触发。

扫描运行模式

扫描运行模式可以通过将ADC_CTL0寄存器的SM位置1来使能。在此模式下，ADC扫描转换所有被ADC_RSQ0~ADC_RSQ2寄存器选中的所有通道。一旦ADCON位被置1，当相应软件触发或者外部触发产生，ADC就会一个接一个的采样和转换常规序列通道。转换数据存储在ADC_RDATA寄存器中。常规序列转换结束后，EOC位将被置1。如果EOCIE位被置1，将产生中断。当常规序列工作在扫描模式下时，ADC_CTL1寄存器的DMA位必须设置为1。

如果ADC_CTL1寄存器的CTN位也被置1，则在常规序列转换完之后，转换自动重新开始。

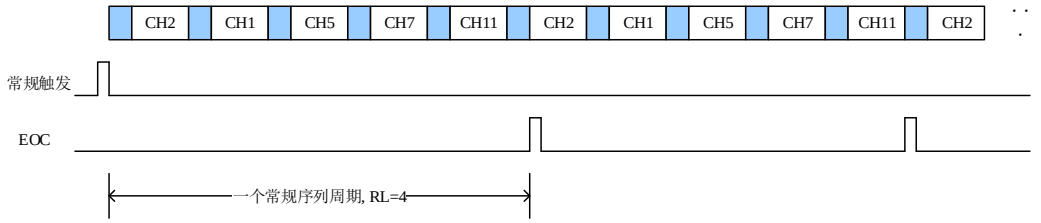
图 13-4. 扫描运行模式，且连续运行模式失能



常规序列扫描运行模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 SM 位和 ADC_CTL1 寄存器的 DMA 位为 1；
2. 配置 ADC_RSQx 和 ADC_SAMPTx 寄存器；
3. 如果有需要，配置 ADC_CTL1 寄存器中的 ETERC 和 ETSRC 位；
4. 准备 DMA 模块，用于传输来自 ADC_RDATA 寄存器的数据；
5. 设置 SWRCST 位，或者给常规序列产生一个外部触发；
6. 等待 EOC 标志位置 1；
7. 写 0 清除 EOC 标志位。

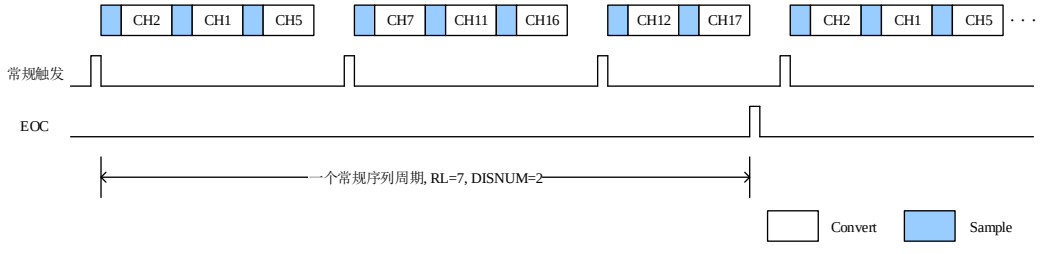
图 13-5. 扫描运行模式，连续运行模式使能



间断运行模式

当 ADC_CTL0 寄存器的 DISRC 位被置 1 时，常规序列使能间断运行模式被使能。该模式下，可以执行一次 n 个通道的短序列转换（n<=8），这个短序列是 ADC_RSQ0~RSQ2 寄存器所选择的转换序列的一部分。数值 n 由 ADC_CTL0 寄存器的 DISCNUM[2:0]位给出。当相应的软件触发或外部触发发生，ADC 就会采样和转换在 ADC_RSQ0~RSQ2 寄存器所选择通道中接下来的 n 个通道，直到常规序列中所有的通道转换完成。每个常规序列转换周期结束后，EOC 位将被置 1。如果 EOCIE 位被置 1 将产生一个中断。

图 13-6. 间断运行模式



常规序列间断运行模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 DISRC 位和 ADC_CTL1 寄存器的 DMA 位为 1；
2. 配置 ADC_CTL0 寄存器的 DISNUM[2:0]位；
3. 配置 ADC_RSQx 和 ADC_SAMPTx 寄存器；
4. 如果有需要，配置 ADC_CTL1 寄存器中的 ETERC 位和 ETSRC 位；

5. 准备 DMA 模块，用于传输来自 ADC_RDATA 寄存器中的数据；
6. 设置 SWRCST 位，或者给常规序列产生一个外部触发；
7. 如果需要，重复步骤 6；
8. 等待 EOC 标志位置 1；
9. 写 0 清除 EOC 标志位。

13.4.6. 转换结果阈值监测功能

ADC_CTL0 寄存器中的 RWDEN 位置 1 时，将使能常规序列的模拟看门狗功能。该功能用于监测转换结果是否超过设定的阈值。如果 ADC 的模拟转换电压低于低阈值或高于高阈值时，ADC_STAT 状态寄存器的 WDE 位将被置 1。如果 WDEIE 位被置 1，将产生中断。ADC_WDHT 和 ADC_WDLT 寄存器用来设定高低阈值。内部数据的比较在对齐之前完成，因此阀值与 ADC_CTL1 寄存器中的 DAL 位确定的对齐方式无关。ADC_CTL0 寄存器的 RWDEN，WDSC 和 WDCHSEL[4:0]位可以用来选择模拟看门狗监控单一通道或多通道。

13.4.7. 数据存储模式

ADC_CTL1 寄存器的 DAL 位确定转换后数据存储的对齐方式。

在最高有效位对齐中，12/10/8 位数据按半字方式对齐，而 6 位数据按照字节的方式对齐的，如下*图 13-7. 12 位数据存储模式*，*图 13-8. 10 位数据存储模式*，*图 13-9. 8 位数据存储模式*和*图 13-10. 6 位数据存储模式*所示。

图 13-7. 12 位数据存储模式

常规通道数据

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

DAL=0

常规通道数据

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

DAL=1

图 13-8. 10 位数据存储模式

常规通道数据

0	0	0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

DAL=0

常规通道数据

D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0

DAL=1

图 13-9. 8 位数据存储模式

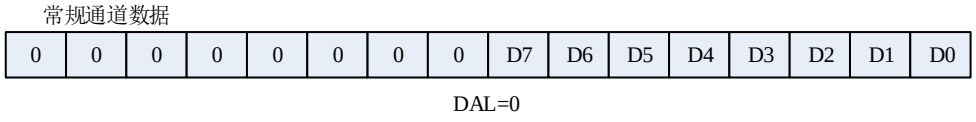
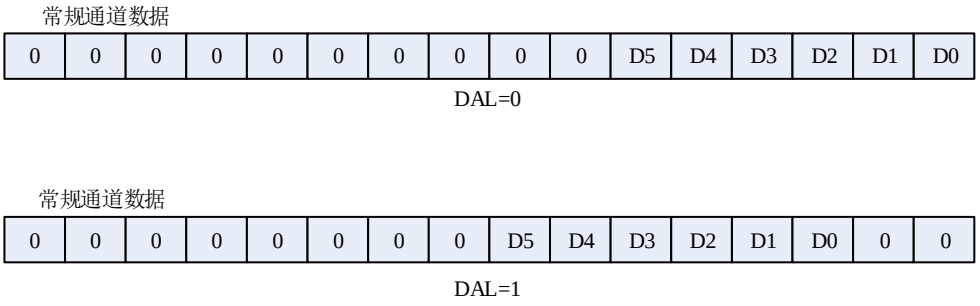


图 13-10. 6 位数据存储模式



13.4.8. 采样时间配置

ADC 使用多个 CK_ADC 周期对输入电压采样，采样周期数目可以通过 ADC_SAMPT0 和 ADC_SAMPT1 寄存器的 SPTn[2:0]位配置。每个通道可以用不同的采样时间。每个通道可以使用不同的时间采样。例如，在 12 位分辨率的情况下，总转换时间=采样时间+12.5 个 CK_ADC 周期。

例如：

CK_ADC = 16MHz，采样时间为 2.5 个周期，那么总的转换时间为：“2.5+12.5”个 CK_ADC 周期，即 0.9375us。

13.4.9. 外部触发配置

外部触发输入的上升沿可以触发常规序列的转换。常规序列的外部触发源由 ADC_CTL1 寄存器的 ETSRC[2:0]位控制。

表 13-3. ADC 的外部触发源

ETSRC[2:0]	触发源	触发类型
000	TIMER8_CH0	硬件触发
001	TIMER8_CH1	
010	保留	
011	TIMER1_CH1	
100	TIMER2_TRGO	
101	TIMER11_CH0	
110	EXTI_11	

ETSRC[2:0]	触发源	触发类型
111	SWRCST	软件触发

13.4.10. DMA 请求

DMA请求，可以通过设置ADC_CTL1寄存器的DMA位来使能，用来传输常规序列多个通道的转换结果。ADC在常规序列一个通道转换结束后产生一个DMA请求，DMA接受到请求后可以将转换的数据从ADC_RDATA寄存器传输到用户指定的目的地址。

13.4.11. ADC 内部通道

将 ADC_CTL1 寄存器的 TSVEN 位置 1，可以使能温度传感器通道（ADC_IN16）。温度传感器可以用来测量器件周围的温度。传感器输出电压能被 ADC 转换成数字量。建议温度传感器的采样时间至少设置为 10μs（ADC 时钟配置不大于 5MHz）。温度传感器不用时，复位 TSVEN 位可以将其置于掉电模式。

温度传感器的输出电压随温度会发生线性变化，由于芯片生产过程的多样化，温度变化曲线的偏移在不同的芯片上会有不同（具体请见数据手册）。

使用温度传感器：

1. 配置温度传感器通道（ADC_IN16）的转换序列和采样时间大于 10μs；
2. 置位 ADC_CTL1 寄存器中的 TSVEN 位，使能温度传感器；
3. 置位 ADC_CTL1 寄存器的 ADCON 位，或者由外部触发启动 ADC 转换；
4. 读取内部温度传感器输出电压 $V_{temperature}$ 并由下面公式计算出实际温度：

$$温度\ (°C) = \{(V_{30} - V_{temperature}) / Avg\_Slope\} + 30$$

$V_{temperature}$：当前温度传感器采样得到的实际温度码值

V_{30}：温度传感器在 30°C 下下的电压，典型值请参考相关型号 datasheet。

Avg_Slope：温度与内部温度传感器电压曲线的均值斜率，典型值请参考相关型号 datasheet。

注意：

1）温度传感器使能后，需等待至少 3 个采样周期，ADC 转换码值才认为有效，前 3 个转换数据应舍弃；

2）可通过硬件过采样或软件求均值的方式提高温度传感器采样精度。

内部电压参考（V_{REFINT}）提供了一个稳定的（带隙基准）电压输出给 ADC 和比较器。V_{REFINT} 内部连接到 ADC_IN17 输入通道。

当 ADC_CTL1 寄存器中的 INREFEN 位置 1 时，可以使能内部参考 V_{REFINT} 通道（ADC_IN17）。内部参考电压 V_{REFINT} 可以为 ADC 和比较器提供稳定的电压（带隙基准），V_{REFINT} 内部连接到到 ADC_IN17。

13.4.12. 电池电压检测

V_{BAT} 通道可以用于监测从 V_{BAT} 引脚过来的备份电池电压。当 ADC_CTL1 寄存器中的 VBATEN 位置 1 时，使能 V_{BAT} 通道（ADC_IN18），同时一个集成在 V_{BAT} 引脚上的 3 分压桥也随之自动使能。由于 V_{BAT} 可能比 VDDA 高，所以使用这个 3 分压桥来确保 ADC 能够正常运行。它

将 ADC_IN18 输入通道连接到 V_{BAT}/3，所以，ADC_IN18 输入通道转换的值是 V_{BAT}/3。为了防止不必要的电池能量消耗，推荐仅在需要时才使能 3 分压桥。

13.4.13. SLCD 电压检测

V_{SLCD} 通道可以用于监测从 V_{SLCD} 引脚上的电压。当 ADC_CTL1 寄存器中的 VSLCDEN 位置 1 时，使能 V_{SLCD} 通道（ADC_IN19），同时一个集成在 V_{SLCD} 引脚上的 3 分压桥也随之自动使能。由于 V_{SLCD} 可能比 VDDA 高，所以使用这个 3 分压桥来确保 ADC 能够正常运行。它将 ADC_IN19 输入通道连接到 V_{SLCD}/3，所以，ADC_IN19 输入通道转换的值是 V_{SLCD} /3。

13.4.14. 可编程分辨率（DRES）

ADC 分辨率可以通过寄存器 ADC_CTL0 中的 DRES[1:0]位进行配置。对于那些不需要高精度数据的应用，可以使用较低的分辨率来实现更快速地转换。只有在 ADCON 位为 0 时，才能修改 DRES[1:0]的值。较低的分辨率能够减少转换时间，如*表 13-4. 不同分辨率对应的 tCONV 时间*所示，较低的分辨率能够减少逐次逼近步骤所需的转换时间 t_{ADC}。

表 13-4. 不同分辨率对应的 t_{CONV} 时间

DRES [1:0] bits	t_{CONV} （ADC 时钟周期）	t_{CONV} （ns）（f_{ADC}=16MHz）	t_{SMPL}（ADC 时钟周期）	t_{ADC}（ADC 时钟周期）	t_{ADC}（ns）（f_{ADC}=16MHz）
12	12.5	781ns	2.5	15	937.5ns
10	10.5	656ns	2.5	13	812.5ns
8	8.5	531ns	2.5	11	687.5ns
6	6.5	406ns	2.5	9	562.5ns

13.4.15. 片上硬件过采样

片上硬件过采样单元执行数据预处理以减轻 CPU 负担。它能够处理多个转换，并将多个转换的结果取平均，得出一个 16 位宽的数据。

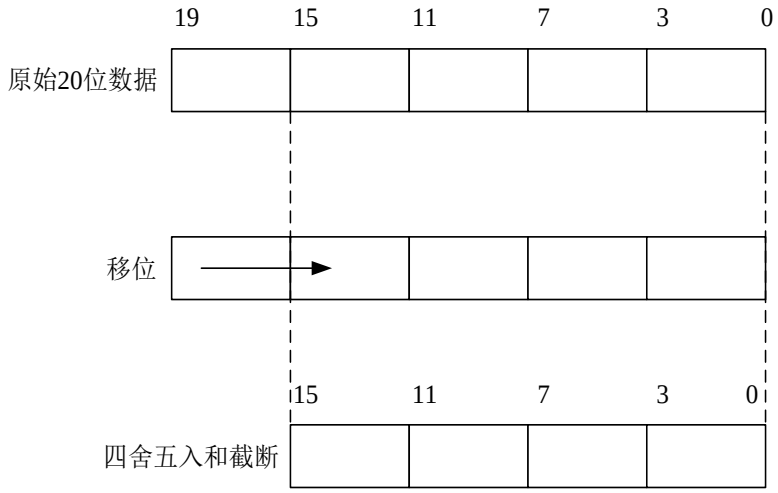
其结果根据如下公式计算得出，其中，N 和 M 的值可以被调整，过采样单元可以通过设置 ADC_OVSAMPCTL 寄存器中的 OVSEN 位来使能，它是以降低数据输出率为代价，换取较高的数据分辨率。$D_{out}(n)$是指 ADC 输出的第 n 个数字信号：

$$\text{Result} = \frac{1}{M} * \sum_{n=0}^{n=N-1} D_{OUT}(n) \qquad (13\text{-}1)$$

片上硬件过采样单元执行两个功能：求和和位右移。过采样率 N 是在 ADC_OVSAMPCTL 寄存器的 OVSR[2:0]位定义，它的取值范围为 2x 到 256x。除法系数 M 定义了一个多达 8 位的右移，它通过 ADC_OVSAMPCTL 寄存器 OVSS[3:0]位进行配置。

求和单元能够生成一个多达 20 位（256*12 位）的值。首先，将这个值进行右移，将移位后剩余的部分再通过取整转化一个近似值，最后将高位截断，仅保留最低 16 位有效位作为最终值传入对应的数据寄存器中。

图 13-11. 20 位到 16 位的结果截断



注意：如果移位后的中间结果还是超过 16 位，那么该结果的高位就会被直接截掉。

*图 13-12. 右移 5 位和取整的数例*描述了一个从原始 20 位的累积数值处理成 16 位结果值的例子。

图 13-12. 右移 5 位和取整的数例

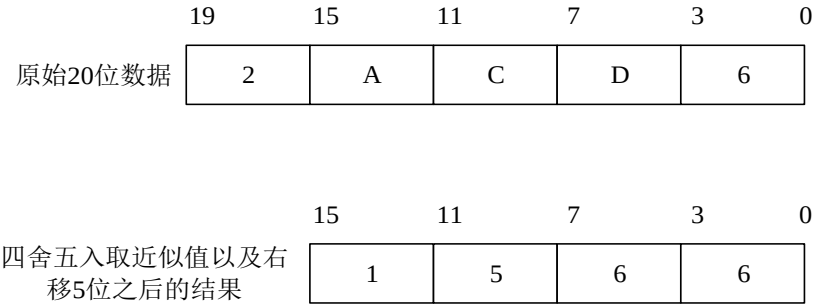


*表 13-5. 不同 N 和 M 组合的最大输出值（灰色值表示截断）*给出了 N 和 M 的各种组合的数据格式，初始转换值为 0xFFF。

表 13-5. 不同 N 和 M 组合的最大输出值（灰色值表示截断）

过采样率	最大原始数据	无移位 OVSS=0000	1 位移位 OVSS=0001	2 位移位 OVSS=0010	3 位移位 OVSS=0011	4 位移位 OVSS=0100	5 位移位 OVSS=0101	6 位移位 OVSS=0110	7 位移位 OVSS=0111	8 位移位 OVSS=1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

和标准的转换模式相比，过采样模式的转换时间不会改变：在整个过采样序列的过程中采样时间仍然保持相等。每 N 个转换就会产生一个新的数据，一个等价的延迟为：

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (13\text{-}2)$$

过采样配合 ADC 工作模式

当过采样使能时，大多数 ADC 工作模式都是可用的。

- 常规序列通道
- 由软件触发或外部触发开始 ADC 转换
- 单次或扫描模式，连续或间断运行模式
- 可编程的采样时间
- 模拟看门狗

只有当 ADCON=0 时，才可以改变过采样的配置，并且要保证在设置 ADCON=1 之前要对过采样进行配置。

13.4.16. ADC 中断

以下任一个事件发生都可以产生中断：

- 常规序列转换结束；
- 模拟看门狗事件；

单独的中断使能位用于灵活设置ADC中断。

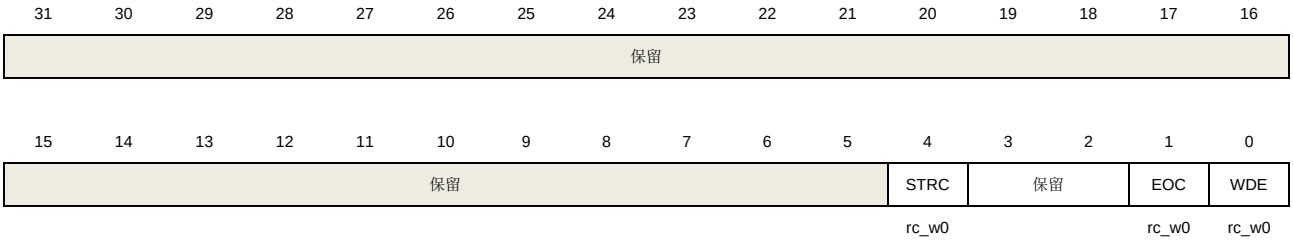
13.5. ADC 寄存器

ADC基地址：0x4001 2400

13.5.1. 状态寄存器（ADC_STAT）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

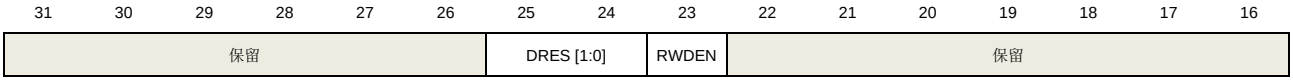


位/位域	名称	描述
31:5	保留	必须保持复位值。
4	STRC	常规序列转换开始标志 0：转换没有开始 1：转换开始 常规序列转换开始时硬件置位，软件写0清除。
3:2	保留	必须保持复位值。
1	EOC	常规序列转换结束标志 0：转换没有结束 1：转换结束 常规序列转换结束时硬件置位，软件写0或读ADC_RDATA寄存器清除。
0	WDE	模拟看门狗事件标志 0：没有模拟看门狗事件 1：产生模拟看门狗事件 转换电压超过ADC_WDLT和ADC_WDHT寄存器中设定的阈值时由硬件置1，软件写0清除。

13.5.2. 控制寄存器 0（ADC_CTL0）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



						rw		rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM [2:0]			保留	DISRC	保留	WDSC	SM	保留	WDEIE	EOCIE	WDCHSEL [4:0]				
rw				rw		rw	rw	rw	rw	rw	rw				

位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	DRES [1:0]	ADC分辨率 00：12位 01：10位 10：8位 11：6位
23	RWDEN	常规序列模拟看门狗使能 0：常规序列模拟看门狗禁止 1：常规序列模拟看门狗使能
22:16	保留	必须保持复位值。
15:13	DISNUM [2:0]	间断模式下的转换转换数目 触发后即将被转换的通道数目将变成DISNUM[2:0]+1。
12	保留	必须保持复位值。
11	DISRC	常规序列间断模式 0：间断运行模式禁止 1：间断运行模式使能
10	保留	必须保持复位值。
9	WDSC	扫描模式下，模拟看门狗在单通道有效 0：模拟看门狗在所有通道有效 1：模拟看门狗在单通道有效
8	SM	扫描模式 0：扫描运行模式禁止 1：扫描运行模式使能
7	保留	必须保持复位值。
6	WDEIE	WDE中断使能 0：中断禁止 1：中断使能
5	EOCIE	EOC中断使能 0：中断禁止 1：中断使能
4:0	WDCHSEL [4:0]	模拟看门狗通道选择 00000：ADC通道0

00001：ADC通道1
00010：ADC通道2
……
01000：ADC通道8
01001：ADC通道9
10000：ADC通道16
10001：ADC通道17
01100: ADC通道18
01101: ADC通道19
注意：ADC的模拟输入通道16、通道17、通道18和通道19内部分别连接到温度传感器、V_{REFINT}、V_{BAT}和V_{SLCD}。

13.5.3. 控制寄存器 1（ADC_CTL1）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					VSLCDEN	VBATEN	INREFEN	TSVEN	SWRCST	保留	ETERC	ETSRC [2:0]			保留
					rw	rw	rw	rw	rw		rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				DAL	保留		DMA	保留				RSTCLB	CLB	CTN	ADCON
				rw			rw					rw	rw	rw	rw

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	VSLCDEN	ADC的通道19（1/3的V_{SLCD}电压）使能 0：ADC的通道19禁止 1：ADC的通道19使能
25	VBATEN	ADC的通道18（1/3外部电池电压）使能 0：ADC的通道18禁止 1：ADC的通道18使能
24	INREFEN	ADC的通道17（内部参考电压）使能 0：ADC的通道17禁止 1：ADC的通道17使能
23	TSVEN	ADC的通道16（温度传感器）使能 0：ADC的通道16禁止 1：ADC的通道16使能
22	SWRCST	软件触发常规序列转换开始 如果ETSRC是111，该位置‘1’开启常规序列转换。该位由软件置位，软件清零或转换开始由硬件清零。

21	保留	必须保持复位值。
20	ETERC	常规序列外部触发使能 0：常规序列外部触发禁止 1：常规序列外部触发使能
19:17	ETSRC [2:0]	常规序列通道外部触发选择 000：TIMER8 CH0 001：TIMER8 CH1 010：保留 011：TIMER1 CH1 100：TIMER2 TRGO 101：TIMER11 CH0 110：中断线11 111：软件触发SWRCST
16:12	保留	必须保持复位值。
11	DAL	数据对齐 0：最低有效位对齐 1：最高有效位对齐
10:9	保留	必须保持复位值
8	DMA	DMA请求使能 0：DMA请求禁止 1：DMA请求使能
7:4	保留	必须保持复位值。
3	RSTCLB	校准复位 软件置位，在校准寄存器初始化后，该位硬件清零。 0：校准寄存器初始化结束. 1：校准寄存器初始化开始
2	CLB	ADC 校准 0：校准结束 1：校准开始
1	CTN	连续模式 0：禁止连续运行模式 1：使能连续运行模式
0	ADCON	开启ADC。该位从‘0’变成‘1’将在稳定时间结束后唤醒ADC。当该位被置位以后，不改变寄存器的其他位仅仅对该位写‘1’，将开启转换。 0：禁能ADC并掉电 1：使能ADC

13.5.4. 采样时间寄存器 0（ADC_SAMPT0）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		SPT19[2:0]			SPT18[2:0]			SPT17[2:0]			SPT16[2:0]			SPT15[2:0]	
		rw			rw			rw			rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT15[0]		SPT14[2:0]			SPT13[2:0]			SPT12[2:0]			SPT11[2:0]			SPT10[2:0]	
rw		rw			rw			rw			rw			rw	

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:27	SPT19[2:0]	参考SPT10[2:0]的描述。
26:24	SPT18[2:0]	参考SPT10[2:0]的描述。
23:21	SPT17[2:0]	参考SPT10[2:0]的描述。
20:18	SPT16[2:0]	参考SPT10[2:0]的描述。
17:15	SPT15[2:0]	参考SPT10[2:0]的描述。
14:12	SPT14[2:0]	参考SPT10[2:0]的描述。
11:9	SPT13[2:0]	参考SPT10[2:0]的描述。
8:6	SPT12[2:0]	参考SPT10[2:0]的描述。
5:3	SPT11[2:0]	参考SPT10[2:0]的描述。
2:0	SPT10[2:0]	通道采样时间 000：通道采样时间为2.5周期 001：通道采样时间为7.5周期 010：通道采样时间为13.5周期 011：通道采样时间为28.5周期 100：通道采样时间为41.5周期 101：通道采样时间为55.5周期 110：通道采样时间为71.5周期 111：通道采样时间为239.5周期

13.5.5. 采样时间寄存器 1（ADC_SAMPT1）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		SPT9[2:0]			SPT8[2:0]			SPT7[2:0]			SPT6[2:0]			SPT5[2:1]	
		rw			rw			rw			rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT5[0]		SPT4[2:0]			SPT3[2:0]			SPT2[2:0]			SPT1[2:0]			SPT0[2:0]	
rw		rw			rw			rw			rw			rw	

位/位域	名称	描述
31:30	保留	必须保持复位值。
29:27	SPT9[2:0]	参考SPT0[2:0]的描述
26:24	SPT8[2:0]	参考SPT0[2:0]的描述
23:21	SPT7[2:0]	参考SPT0[2:0]的描述
20:18	SPT6[2:0]	参考SPT0[2:0]的描述
17:15	SPT5[2:0]	参考SPT0[2:0]的描述
14:12	SPT4[2:0]	参考SPT0[2:0]的描述
11:9	SPT3[2:0]	参考SPT0[2:0]的描述
8:6	SPT2[2:0]	参考SPT0[2:0]的描述
5:3	SPT1[2:0]	参考SPT0[2:0]的描述
2:0	SPT0[2:0]	通道采样时间 000：通道采样时间为2.5周期 001：通道采样时间为7.5周期 010：通道采样时间为13.5周期 011：通道采样时间为28.5周期 100：通道采样时间为41.5周期 101：通道采样时间为55.5周期 110：通道采样时间为71.5周期 111：通道采样时间为239.5周期

13.5.6. 看门狗高阈值寄存器（ADC_WDHT）

地址偏移：0x24

复位值：0x0000 0FFF

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				WDHT [11:0]											

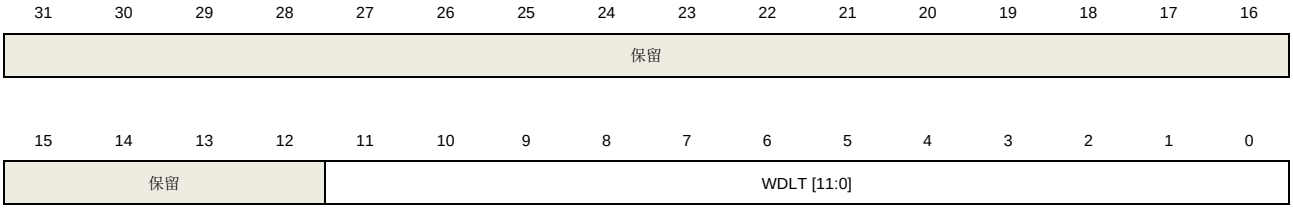
rw

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	WDHT [11:0]	模拟看门狗高侧阈值 这些位定义了模拟看门狗的高侧阈值。

13.5.7. 看门狗低阈值寄存器（ADC_WDLT）

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

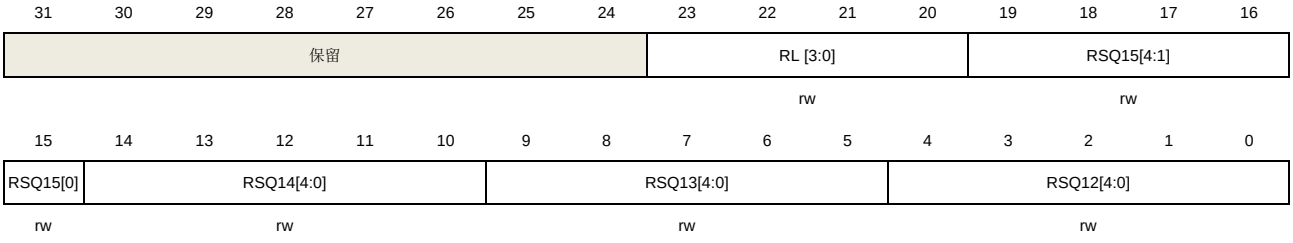


位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	WDLT [11:0]	模拟看门狗低侧阈值 这些位定义了模拟看门狗的低侧阈值。

13.5.8. 常规序列寄存器 0（ADC_RSQ0）

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:24	保留	必须保持复位值。
23:20	RL[3:0]	常规序列长度 常规通道转换序列中的总通道数目为RL[3:0]+1。

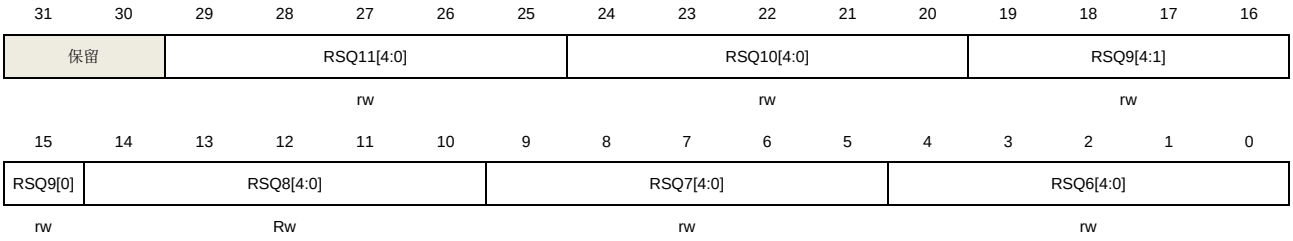
19:15	RSQ15[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ14[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ13[4:0]	参考RSQ0[4:0]的描述
4:0	RSQ12[4:0]	参考RSQ0[4:0]的描述

13.5.9. 常规序列寄存器 1（ADC_RSQ1）

地址偏移：0x30

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



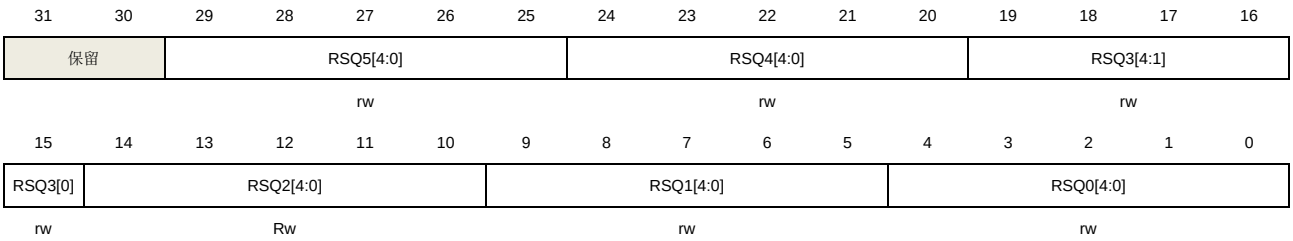
位/位域	名称	描述
31:30	保留	必须保持复位值。
29:25	RSQ11[4:0]	参考RSQ0[4:0]的描述
24:20	RSQ10[4:0]	参考RSQ0[4:0]的描述
19:15	RSQ9[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ8[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ7[4:0]	参考RSQ0[4:0]的描述
4:0	RSQ6[4:0]	参考RSQ0[4:0]的描述

13.5.10. 常规序列寄存器 2（ADC_RSQ2）

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

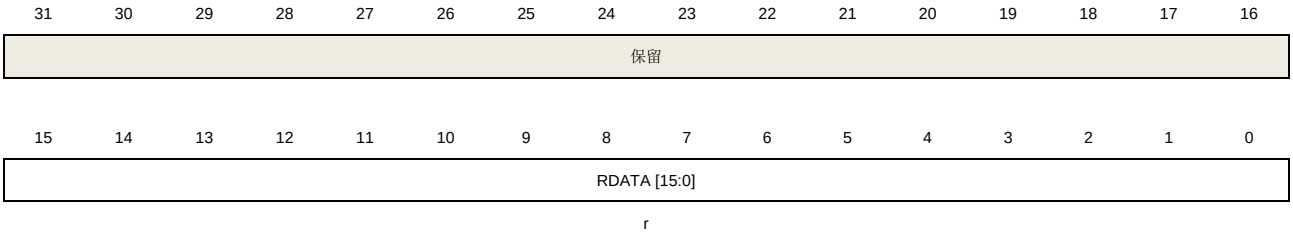


位/位域	名称	描述
31:30	保留	必须保持复位值。
29:25	RSQ5[4:0]	参考RSQ0[4:0]的描述
24:20	RSQ4[4:0]	参考RSQ0[4:0]的描述
19:15	RSQ3[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ2[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ1[4:0]	参考RSQ0[4:0]的描述
4:0	RSQ0[4:0]	通道编号（0..19）写入这些位来选择常规通道的第n个转换的通道

13.5.11. 常规数据寄存器（ADC_RDATA）

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

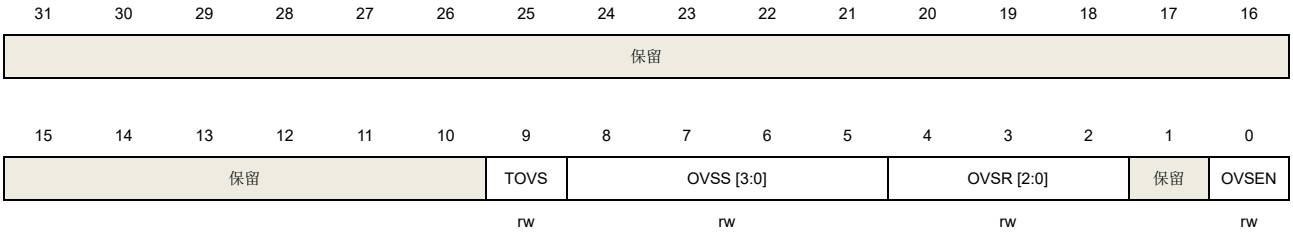


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RDATA [15:0]	常规通道转换数据 这些位包含了常规通道的转换结果，只读。

13.5.12. 过采样控制寄存器（ADC_OVSAMPCTL）

地址偏移：0x80
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述

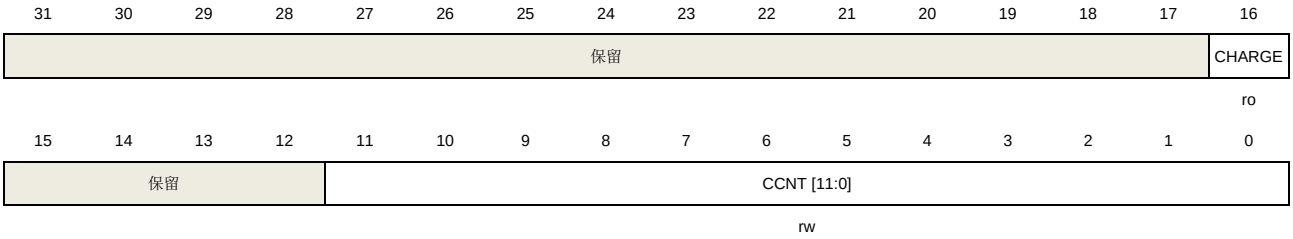
31:10	保留	必须保持复位值。
9	TOVS	过采样触发 该位通过软件设置和清除。 0：在一个触发后，连续执行过采样通道的所有转换 1：对于过采样通道的每次转换都需要一次触发，触发次数由过采样率（OVSR[2:0]）决定 **注意：**当ADCON= 0时软件才允许写该位（确定没有转换正在进行）。
8:5	OVSS [3:0]	过采样移位 该位通过软件设置和清除。 0000：不移位 0001：移1位 0010：移2位 0011：移3位 0100：移4位 0101：移5位 0110：移6位 0111：移7位 1000：移8位 其它位保留 **注意：**只有在ADCON=0的时候，才允许通过软件对该位进行写操作（确保没有转换正在进行）。
4:2	OVSR [2:0]	过采样率 这些位定义了过采样率的大小。 000：2x 001：4x 010：8x 011：16x 100：32x 101：64x 110：128x 111：256x **注意：**只有在ADCON=0的时候，才允许通过软件对该位进行写操作（确保没有转换正在执行）。
1	保留	必须保持复位值。
0	OVSEN	过采样使能 该位通过软件置位和清除。 0：过采样禁止 1：过采样使能 **注意：**只有在ADCON=0的时候，才允许通过软件对该位进行写操作（确保没有转换正在执行）。

13.5.13. 充电控制寄存器（ADC_CCTL）

地址偏移：0x C0

复位值：0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:17	保留	必须保持复位值。
16	CHARGE	ADC充电状态 0：无充电 1：正在充电中 该位由硬件置位和清零。
15:12	保留	必须保持复位值
11:0	CCNT [11:0]	ADC充电脉冲宽度计数器 该位用于控制ADC充电脉冲的宽度，CCNT值与脉冲宽度的关系如下： 脉冲宽度 = 5us = CCNT [11:0] * t_{PCLK2}。 **注意**：只有在ADCON =0（确定没有转换正在进行）时，才能软件写该位。

14. 数模转换器（DAC）

14.1. 简介

数字/模拟转换器可以将 12 位的数字数据转换为外部引脚上的电压输出。数据可以采用 8 位或 12 位模式，左对齐或右对齐模式。当使能了外部触发，DMA 可被用于更新输入端数字数据。在输出电压时，可以利用 DAC 输出缓冲区来获得更高的驱动能力。

14.2. 主要特征

DAC 的主要特征如下：

- 8 位或 12 位分辨率，数据右对齐或左对齐；
- 支持 DMA 功能；
- 同步更新转换；
- 外部事件触发转换；
- 可配置的内部缓冲区；
- 外部参考电压，V_{REF+}；
- 噪声波形(LSFR 噪声模式和三角噪声模式)；

*图 14-1. DAC 结构框图*为 DAC 的结构框图，*表 14-1. DAC I/O 描述*给出了引脚描述。

图 14-1. DAC 结构框图

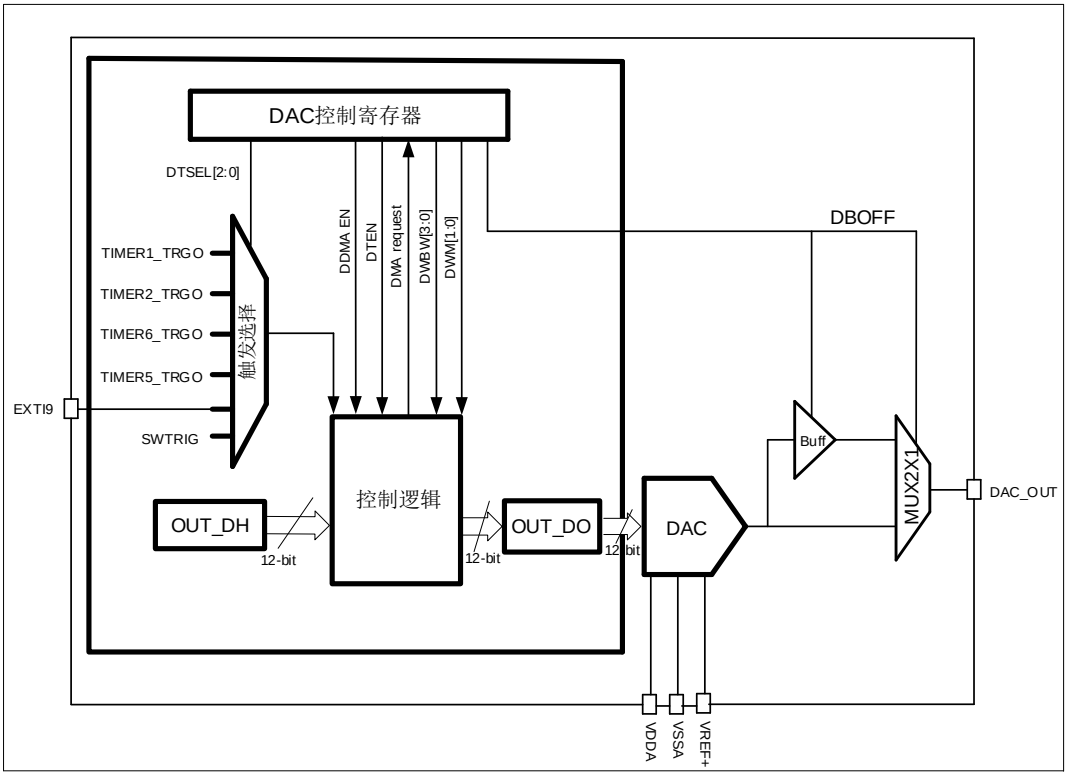


表 14-1. DAC I/O 描述

名称	描述	信号类型

V_{DDA}	模拟电源	电源
V_{SSA}	模拟电源地	电源
V_{REF+}	参考电压	模拟输入
DAC_OUT	DACx 模拟输出	模拟输出

注意：在使能 DAC 模块前，GPIO 口（PA4 对应 DAC_OUT）应配置为模拟模式。

14.3. 功能描述

14.3.1. DAC 使能

将 DAC_CTL0 寄存器中的 DEN 位置 1，可以给 DAC 模块上电，DAC 子模块完全启动需要等待 t_{WAKEUP} 时间。

14.3.2. DAC 输出缓冲

为了降低输出阻抗并驱动外部负载，每个 DAC 模块内部各集成了一个输出缓冲区。

缺省情况下，输出缓冲区是开启的，可以通过设置 DAC_CTL0 寄存器的 DBOFF 位来关闭缓冲区。

14.3.3. DAC 数据配置

对于 12 位的 DAC 保持数据（OUT_DH），可以通过对 OUT_R12DH、OUT_L12DH 和 OUT_R8DH 中的任意一个寄存器写入数据来配置。当数据被加载到 OUT_R8DH 寄存器时，只有 8 位最高有效位是可配置，4 位最低有效位被强制置为 0。

14.3.4. DAC 触发

通过设置 DAC_CTL0 寄存器中 DTEN 位来使能 DAC 外部触发。触发源可以通过 DAC_CTL0 寄存器中 DTSEL 位来进行选择，如*表 14-2. DAC 外部触发*所示。

表 14-2. DAC 外部触发

DTSEL[2:0]	触发源	触发类型
3b'000	TIMER1_TRGO	硬件触发
3b'001	TIMER2_TRGO	
3b'010	保留	
3b'011	保留	
3b'100	TIMER6_TRGO	
3b'101	TIMER5_TRGO	
3b'110	EXTI9	
3b'111	SWTRIG	软件触发

TIMERx_TRGO 信号是由定时器生成的，软件触发是通过设置 DAC_SWT 寄存器的 SWTR 位生成的。

14.3.5. DAC 工作流程

如果使能了外部触发（通过设置 DAC_CTL0 寄存器的 DTEN 位），当已经选择的触发事件发生，DAC 保持数据（OUT_DH）会被转移到 DAC 数据输出寄存器（OUT_DO）。而在外部触发未使能的情况下，DAC 保持数据（OUT_DH）会被自动转移到 DAC 数据输出寄存器（OUT_DO）。

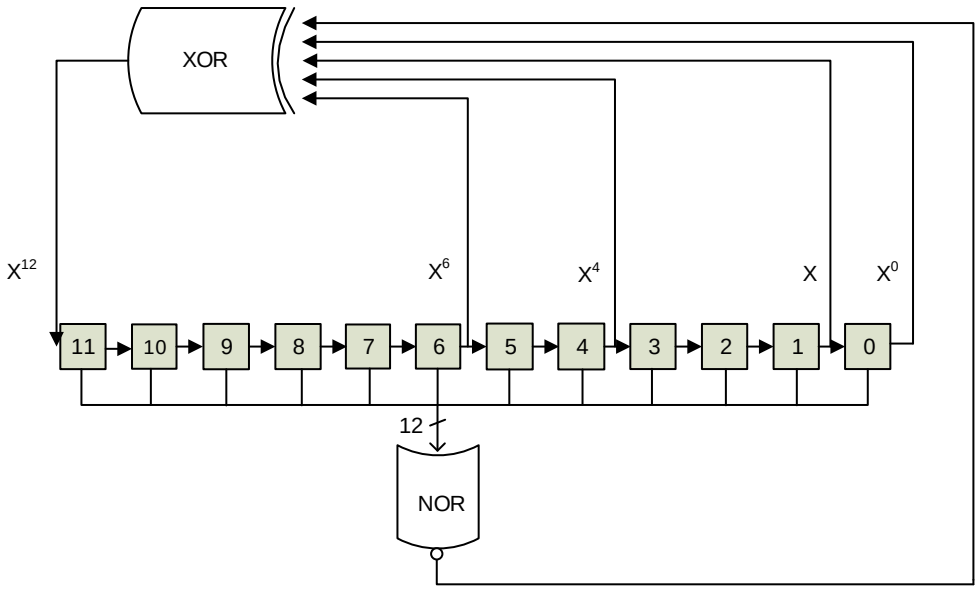
当 DAC 保持数据（OUT_DH）加载到 OUT_DO 寄存器时，再经过 $t_{SETTLING}$ 时间之后，模拟输出变得有效，$t_{SETTLING}$ 的值与电源电压和模拟输出负载有关。

14.3.6. DAC 噪声波

有两种方式可以将噪声波加载到 DAC 输出数据：LFSR 噪声波和三角波。噪声波模式可以通过 DAC_CTL0 寄存器的 DWM 位来进行选择。噪声的幅值可以通过配置 DAC_CTL0 寄存器的 DAC 噪声波位宽（DWBW）位来进行设置。

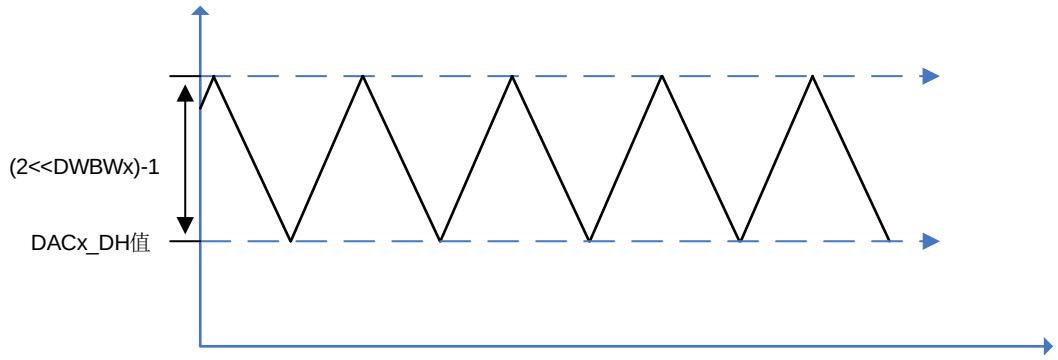
LFSR 噪声模式：在 DAC 控制逻辑中有一个线性反馈移位寄存器（LFSR）。在此模式下，LFSR 的值与 OUT_DH 值相加后，被写入到 DAC 数据输出寄存器（OUT_DO）。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBW 位，DWBW 位决定了不屏蔽 LFSR 的哪些位。

图 14-2. DAC LFSR 算法



三角噪声模式：三角波幅值与 OUT_DH 值相加后，被写入到 DAC 数据输出寄存器（OUT_DO）。三角波幅值的最小值为 0，最大值为(2 << DWBW） - 1。

图 14-3. DAC 三角噪声模式波形



14.3.7. DAC 输出计算

DAC 引脚上的输出电压取决于下面的等式：

$$V_{DACx\_out}=V_{REF+}*OUT\_DO/4096 \quad (14\text{-}1)$$

数字输入被线性地转换成模拟输出电压，输出范围为 0 到 V_{REF+}。

14.3.8. DMA 功能

在外部触发使能的情况下，通过设置 DAC_CTL0 寄存器的 DDMAEN 位来使能 DMA 请求。当有外部硬件触发的时候（不是软件触发），则产生一个 DMA 请求。

如果在前一个请求响应之前第二个外部触发到达，则不响应新到的触发请求，并且发生欠载错误事件。DAC_STAT0 寄存器中的 DDUDR 位置 1，如果 DAC_CTL0 寄存器中的 DDUDRIE 位置 1，则会产生中断。

14.4. DAC 寄存器

DAC 基地址：0x4000 7400

14.4.1. 控制寄存器 (DAC_CTL0)

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DDISC	DDUDR IE	DDMA EN	DWBW[3:0]				DWM[1:0]		DTSEL[2:0]			DTEN	DBOFF	DEN
	rw	rw	rw	rw				rw		rw			rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值
14	DDISC	DAC_OUT 连接 GPIO 选择 0：DAC 连接到外部管脚和片上外设（CMP） 1：当输出缓冲区关闭时（DBOFF=1）仅连接到片上外设（CMP），缓冲区打开时连接到外部管脚和片上外设（CMP）
13	DDUDRIE	DAC_OUT DMA欠载中断使能 0：DAC_OUT DMA欠载中断禁能 1：DAC_OUT DMA 欠载中断使能
12	DDMAEN	DAC_OUT DMA 使能 0：DAC_OUT DMA 模式禁能 1：DAC_OUT DMA 模式使能
11:8	DWBW[3:0]	DAC_OUT 噪声波位宽 这些位指定了 DAC_OUT 的噪声波信号的位宽。LFSR 噪声模式下，这些位表示不屏蔽 LFSR 的位[n-1, 0]；三角噪声模式下，这些位表示三角波幅值为(2<<(n-1))-1。其中，n 为噪声波位宽。 0000：波形信号的位宽为 1 0001：波形信号的位宽为 2 0010：波形信号的位宽为 3 0011：波形信号的位宽为 4 0100：波形信号的位宽为 5 0101：波形信号的位宽为 6 0110：波形信号的位宽为 7 0111：波形信号的位宽为 8

1000：波形信号的位宽为 9
1001：波形信号的位宽为 10
1010：波形信号的位宽为 11
≥1011：波形信号的位宽为 12

7:6	DWM[1:0]	DAC_OUT 噪声波模式 这些位指定了在 DAC_OUT 外部触发使能(DTEN0=1)的情况下，DAC_OUT 的噪声波模式的选择。 00：波形生成禁能 01：LFSR 噪声模式 1x：三角噪声模式
5:3	DTSEL[2:0]	DAC_OUT 触发选择 这些位用于在 DAC_OUT 外部触发使能(DTEN=1)时，DAC_OUT 外部触发的选择。 000：TIMER1 TRGO 001：TIMER2 TRGO 010：保留 011：保留 100：TIMER6 TRGO 101：TIMER5 TRGO 110：外部中断线 9 111：软件触发
2	DTEN	DAC_OUT 触发使能 0：DAC_OUT 触发禁能 1：DAC_OUT 触发使能
1	DBOFF	DAC_OUT 输出缓冲区关闭 0：DAC_OUT 输出缓冲区打开,以降低输出阻抗，提高驱动能力 1：DAC_OUT 输出缓冲区关闭
0	DEN	DAC_OUT 使能 0：DAC_OUT 禁能 1：DAC_OUT 使能

14.4.2. 软件触发寄存器 (DAC_SWT)

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															SWTR

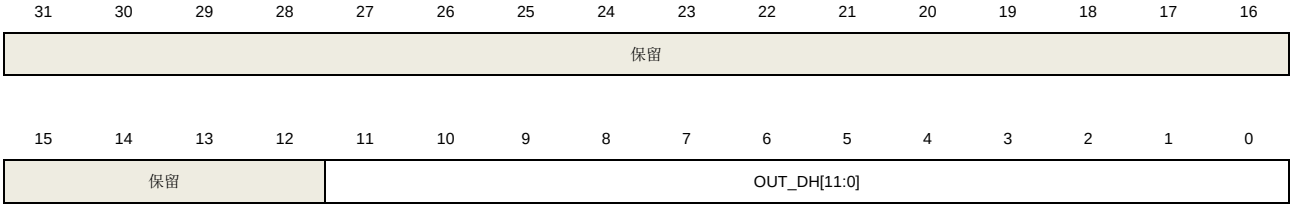
w

位/位域	名称	描述
31:1	保留	必须保持复位值
0	SWTR	DAC_OUT 软件触发，由硬件清除。 0：软件触发禁能 1：软件触发使能

14.4.3. DAC_OUT 12 位右对齐数据保持寄存器 (OUT_R12DH)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



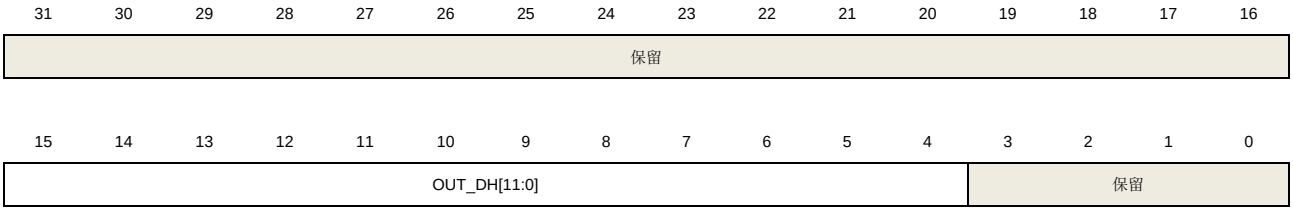
rw

位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT_DH[11:0]	DAC_OUT 12 位右对齐数据 这些位指定了将由 DAC_OUT 转换的数据。

14.4.4. DAC_OUT 12 位左对齐数据保持寄存器 (OUT_L12DH)

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	OUT_DH[11:0]	DAC_OUT 12 位左对齐数据

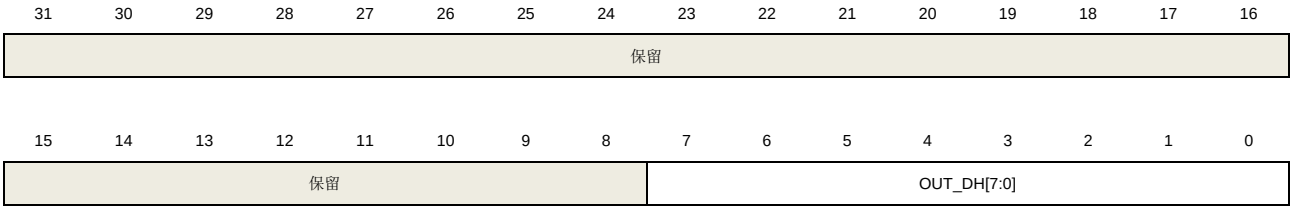
这些位指定了将由 DAC_OUT 转换的数据。

3:0	保留	必须保持复位值

14.4.5. DAC_OUT 8 位右对齐数据保持寄存器 (OUT_R8DH)

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

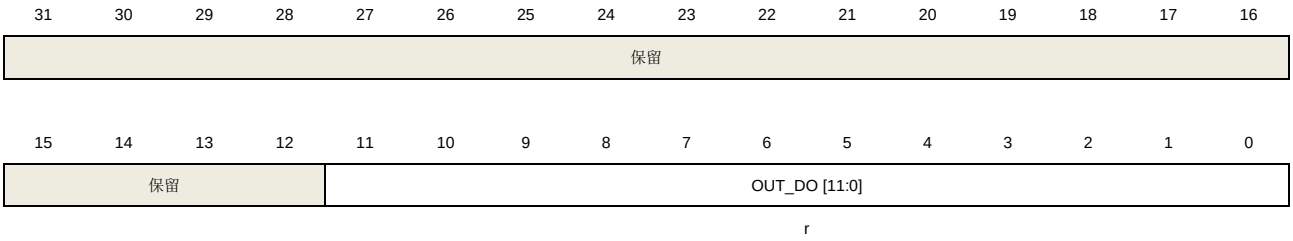


位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT_DH[7:0]	DAC_OUT 8 位右对齐数据 这些位指定了将由 DAC_OUT 转换的数据的最高 8 位有效位。

14.4.6. DAC_OUT 数据输出寄存器 (OUT_DO)

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT_DO [11:0]	DAC_OUT 数据输出 这些位为只读类型，存储由 DAC_OUT 转换的数据。

14.4.7. DAC 状态寄存器 0(DAC_STAT0)

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	13	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		DDUDR	保留												
		rc_w1													

位/位域	名称	描述
31:14	保留	必须保持复位值
13	DDUDR	DAC_OUT DMA 欠载标志位，硬件置位，软件写 1 清零。 0：没有欠载发生。 1：发生欠载（DAC 触发产生速度快于 DMA 传输速度）。
12:0	保留	必须保持复位值

15. 看门狗定时器（WDGT）

看门狗定时器（WDGT）是一个硬件计时电路，用来监测由软件故障导致的系统故障。片上有两个看门狗定时器外设，独立看门狗定时器（FWDGT）和窗口看门狗定时器（WWDGT）。它们使用灵活，并提供了很高的安全水平和精准的时间控制。两个看门狗定时器都是用来解决软件故障问题的。

看门狗定时器在内部计数值达到了预设的门限时，会触发一个复位（对于窗口看门狗定时器来说，会产生一个中断）。当处理器工作在调试模式的时候看门狗定时器定时计数器可以停止计数。

15.1. 独立看门狗定时器（FWDGT）

15.1.1. 简介

独立看门狗定时器（FWDGT）有独立时钟源（IRC32K）。即使主时钟失效，FWDGT依然能保持正常工作状态，适用于需要独立环境且对计时精度要求不高的场合。

当内部向下计数器的计数值达到0，独立看门狗会产生一个系统复位。使能独立看门狗的寄存器写保护功能可以避免寄存器的值被意外的配置篡改。

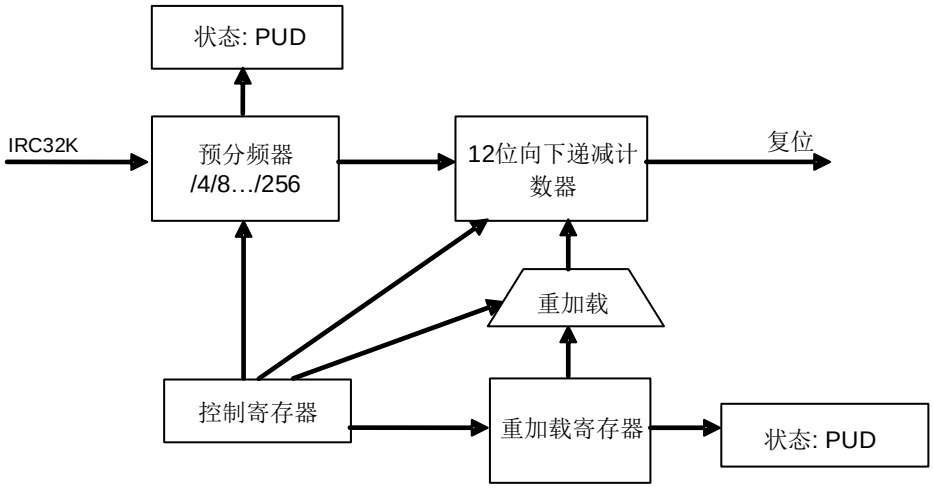
15.1.2. 主要特征

- 自由运行的12位向下计数器；
- 如果看门狗定时器被使能，有以下两种情况下会产生复位：
  - 当计数器到0时产生复位；
  - 当计数器的值大于窗口寄存器的值时，更新计数器会产生复位；
- 独立时钟源，独立看门狗定时器在主时钟故障(例如待机和深度睡眠模式下)时仍能工作；
- 独立看门狗定时器硬件控制位，用来控制是否在上电时自动启动独立看门狗定时器；
- 可以配置独立看门狗定时器在调试模式下选择停止还是继续工作。

15.1.3. 功能说明

独立看门狗定时器带有一个8级预分频器和一个12位的向下递减计数器。参考*图15-1. 独立看门狗定时器框图*为独立看门狗定时器的功能模块。

图 15-1. 独立看门狗定时器框图



向控制寄存器（FWDGT_CTL）中写0xCCCC可以开启独立看门狗定时器，计数器开始向下计数。当计数器记到0x000，产生一次系统复位。

在任何时候向FWDGT_CTL中写0xAAAA都可以重装载计数器，重装载值来源于重装载寄存器（FWDGT_RLD）。软件可以在计数器计数值达到0x000之前可以通过重装载计数器来阻止看门狗定时器产生系统复位。

独立看门狗定时器也能够作为窗口看门狗定时器运行，只要在窗口寄存器（FWDGT_WND）中设置适当的窗口值即可。当重装载操作执行时，如果看门狗定时器计数器的值大于窗口寄存器(FWDGT_WND)中存储的值,将会引起系统复位。FWDGT_WND 的默认值是 0x00000FFF，所以如果没有改写它，那么窗口选项默认是关闭的。窗口值一旦改变，立即就会引起看门狗定时器计数器的一次重加载动作，将向下递减计数器置为 FWDGT_RLD 中的值，并复位预分频计数器。

如果在选项字节中打开了“硬件看门狗定时器”功能，那么在上电的时候看门狗定时器就被自动打开。为了避免系统复位，软件应该在计数器达到0x000之前重装载计数器。

预分频寄存器（FWDGT_PSC）和FWDGT_RLD寄存器都有写保护功能。在写数据到这些寄存器之前，需要写0x5555到FWDGT_CTL中。写其他任何值到FWDGT_CTL中将会再次启动对这些寄存器的写保护。当FWDGT_PSC或者FWDGT_RLD更新时，FWDGT_STAT寄存器相应的状态位被置1。

如果DBG中控制寄存器0（DBG_CTL0）中的FWDGT_HOLD位被清0，即使Cortex™-M23内核停止（调试模式下）独立看门狗定时器依然工作。如果FWDGT_HOLD位被置1，独立看门狗定时器将在调试模式下停止工作。

表 15-1. 独立看门狗定时器在 32kHz（IRC32K）时的最小/最大超时周期

预分频系数	PSC[2:0] 位	最小超时（ms） RLD[11:0]=0x000	最大超时（ms） RLD[11:0]=0xFFF
1 / 4	000	0.03125	511.90625
1 / 8	001	0.03125	1023.78125
1 / 16	010	0.03125	2047.53125
1 / 32	011	0.03125	4095.03125
1 / 64	100	0.03125	8190.03125

预分频系数	PSC[2:0] 位	最小超时（ms） RLD[11:0]=0x000	最大超时（ms） RLD[11:0]=0xFFF
1 / 128	101	0.03125	16380.03125
1 / 256	110或111	0.03125	32760.03125

通过校准IRC32K可以使自由看门狗定时器超时更加精确。

注意：当执行完喂狗reload操作之后，如需要立即进入deepsleep / standby模式时，必须通过软件设置，在reload命令及deepsleep / standby模式命令中间插入（3个以上）IRC32K时钟间隔。

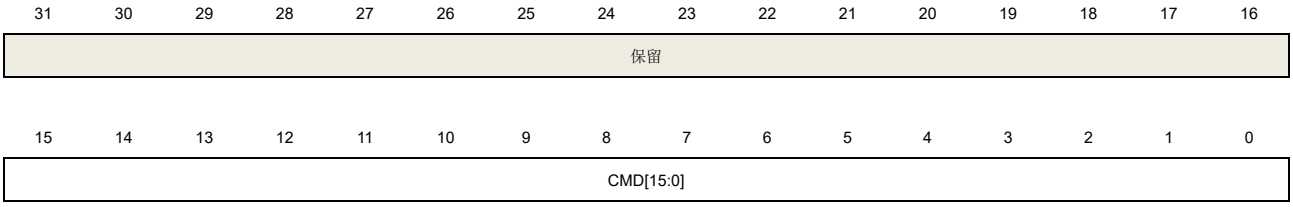
15.1.4. FWDGT 寄存器

FWDGT 基地址：0x4000 3000

控制寄存器（FWDGT_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。

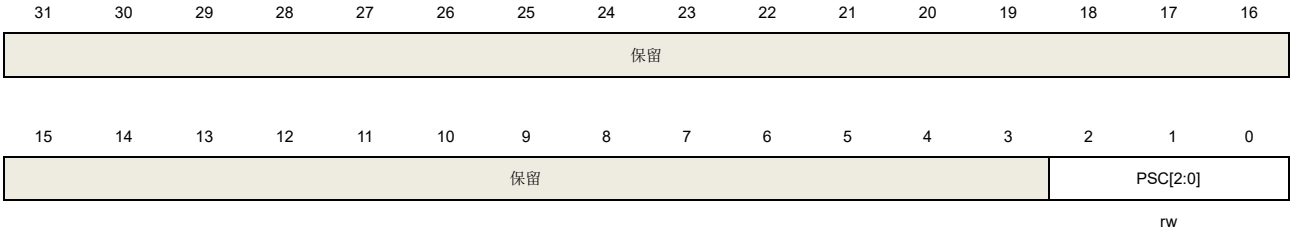


位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	CMD[15:0]	只可写，写入不同的值来产生不同的功能 0x5555：关闭FWDGT_PSC和FWDGT_RLD的写保护 0xCCCC：开启独立看门狗定时器定时计数器。计数减到0时产生中断 0xAAAA：重装载计数器

预分频寄存器（FWDGT_PSC）

地址偏移：0x04
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。



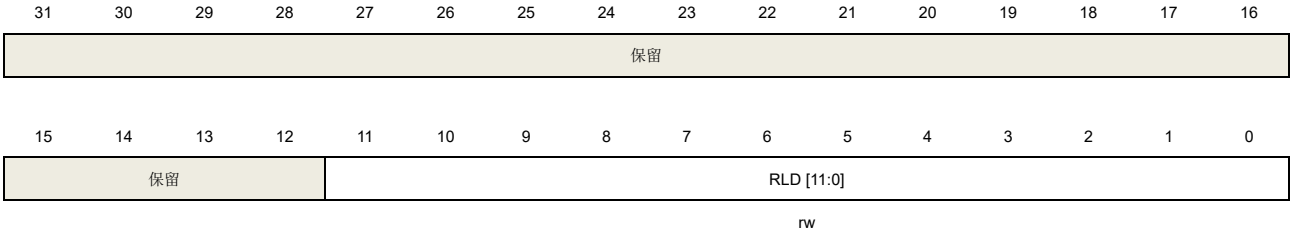
位/位域	名称	说明
31:3	保留	必须保持复位值。
2:0	PSC[2:0]	独立看门狗定时器计时预分频选择。写这些位之前要通过向FWDGT_CTL寄存器写0x5555去除写保护。在改写这个寄存器的过程中，FWDGT_STAT寄存器的PUD位被置1，此时读取此寄存器的值都是无效的。 000：1 / 4 001：1 / 8 010：1 / 16

011：1 / 32
100：1 / 64
101：1 / 128
110：1 / 256
111：1 / 256
如果应用需要使用几个预分频系数，改变预分频值之前必须等到PUD位被清0。更新了预分频寄存器中的值后，在代码持续执行之前不必等待PUD值被清零。

重装载寄存器（FWDGT_RLD）

地址偏移：0x08
复位值：0x0000 0FFF

该寄存器可以按半字（16 位）或字（32 位）访问。

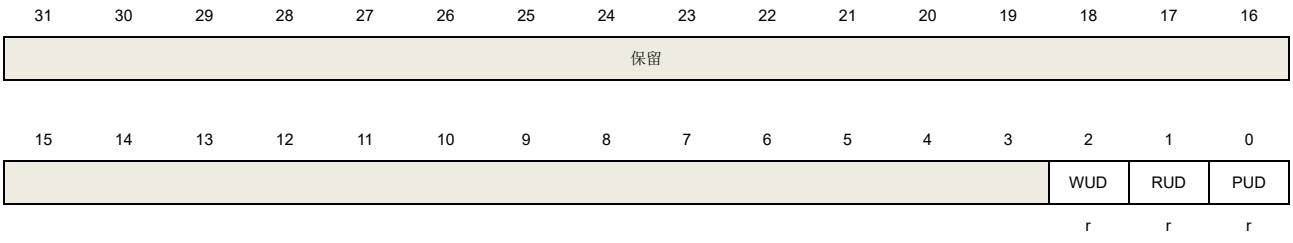


位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	RLD[11:0]	独立看门狗定时器定时计数器重装载值。向FWDGT_CTL寄存器写入0xAAAA的时候，这个值会被更新到看门狗定时器计数器中。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。在改写这个寄存器的过程中，FWDGT_STAT寄存器的RUD位被置1，从此寄存器中读取的任何值都是无效的。 如果应用需要使用几个重装载值，改变重装载值之前必须等到RUD位被清0。更新了重装载寄存器的值后，在代码持续执行之前不必等待RUD值被清零（在进入省电模式前需等待RUD值清零）。

状态寄存器（FWDGT_STAT）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。



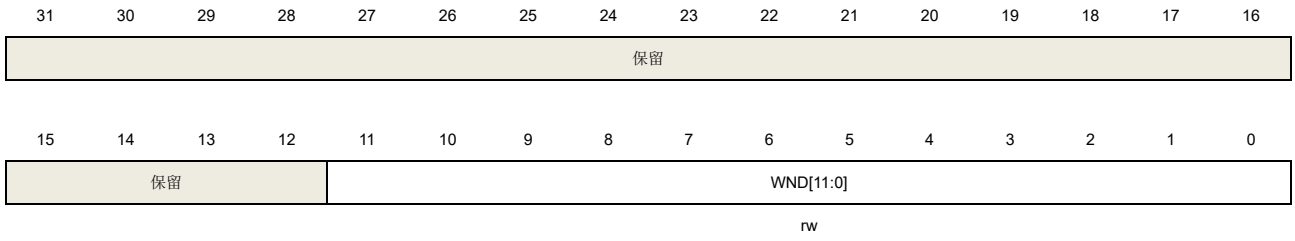
位/位域	名称	说明
31:3	保留	必须保持复位值。
2	WUD	独立看门狗定时器计数器窗口值更新。 FWDGT_WND寄存器写操作时，该位被置1，此时读取FWDGT_WND寄存器的任何值都是无效的。
1	RUD	独立看门狗定时器计数器重装载值更新。 FWDGT_RLD寄存器写操作时，该位被置1，此时读取FWDGT_RLD寄存器的任何值都是无效的。在FWDGT_RLD寄存器更新后，该位由硬件清零。
0	PUD	独立看门狗定时器预分频值更新。 FWDGT_PSC寄存器写操作时，该位被置1，此时读取FWDGT_PSC寄存器的任何值都是无效的。在FWDGT_PSC寄存器更新后，该位由硬件清零。

窗口寄存器（FWDGT_WND）

地址偏移：0x10

复位值：0x0000 0FFF

该寄存器可以按半字（16 位）或字（32 位）访问。



位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WND	独立看门狗定时器计数器窗口值。这些位将用来将窗口值的上限值与向下递减计数器进行比较。当计数值大于WMD[11:0]中值，重装载操作会引起复位，若要改变重装载值，FWDGT_STAT寄存器中的WUD位必须保持复位状态。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。 如果应用需要使用几个窗口值，改变窗口值之前必须等到WUD位被清0。除了在进入低功耗模式下，更新了窗口值后，在代码持续执行之前不必等待WUD值被清零。

15.2. 窗口看门狗定时器（WWDGT）

15.2.1. 简介

窗口看门狗定时器（WWDGT）用来监测由软件故障导致的系统故障。窗口看门狗定时器开启后，7位向下递减计数器值逐渐减小。计数值达到0x3F时会产生系统复位（CNT[6]位被清0）。在计数器计数值达到窗口寄存器值之前，计数器的更新也会产生系统复位。因此软件需要在给定的区间内更新计数器。窗口看门狗定时器在计数器计数值达到0x40，会产生一个提前唤醒标志，如果使能中断将会产生提前唤醒中断。

窗口看门狗定时器时钟是由APB1时钟预分频而来。窗口看门狗定时器适用于需要精确计时的场合。

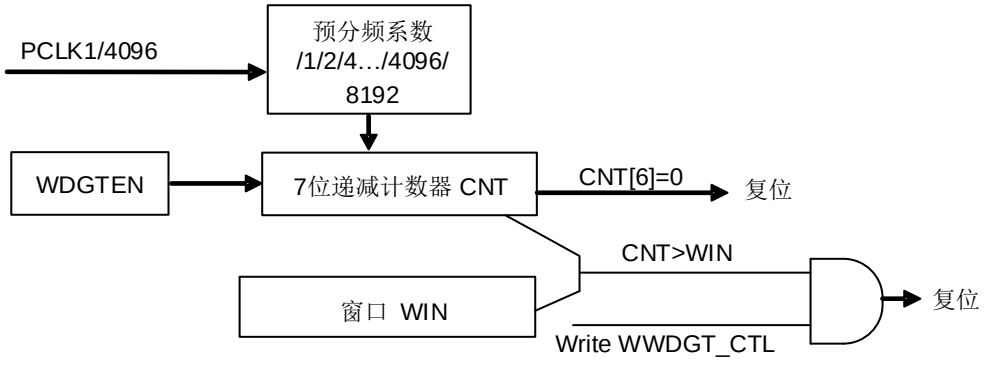
15.2.2. 主要特征

- 可编程的7位自由运行向下递减计数器。
- 当窗口看门狗使能后，有以下两种情况会产生复位：
  - 当计数器达到0x3F时产生复位；
  - 当计数器的值大于窗口寄存器的值时，更新计数器会产生复位。
- 提前唤醒中断(EWI)：如果看门狗定时器打开，中断允许，计数值达到0x40时会产生中断。
- 可以配置窗口看门狗定时器在调试模式下选择停止还是继续工作。

15.2.3. 功能说明

如果窗口看门狗定时器使能(将WWDGT_CTL寄存器的WDGTEN位置1)，计数值达到0x3F的时候产生系统复位(CNT[6]位被清0)。或是在计数值达到窗口寄存器值之前，更新计数器也会产生系统复位。

图 15-2. 窗口看门狗定时器框图



上电复位之后窗口看门狗定时器总是关闭的。软件可以向WWDGT_CTL的WDGTEN写1开启窗口看门狗定时器。窗口看门狗定时器打开后，计数器始终递减计数，计数器配置的值应该大于0x3F，也就是说CNT[6]位应该被置1。CNT[5:0]决定了两次重装载之间的最大间隔时间。计数器的递减速度取决于APB1时钟和预分频器(WWDGT_CFG寄存器的PSC[3:0]位)。

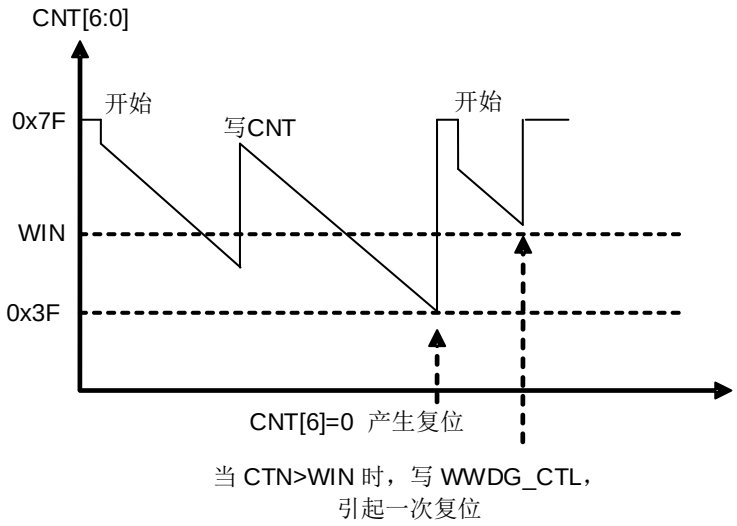
配置寄存器 (WWDGT_CFG)中的WIN[6:0]位用来设定窗口值。当计数器的值小于窗口值，且

大于0x3F的时候，重装载向下计数器可以避免复位，否则在其他时候进行重加载就会引起复位。

对WWDGT_CFG寄存器的EWIE位置1可以使能提前唤醒中断（EWI），当计数值达到0x40的时候该中断产生。同时可以用相应的中断服务程序（ISR）来触发特定的行为（例如通信或数据记录），来分析软件故障的原因以及在器件复位的时候挽救重要数据。此外，在ISR中软件可以重装载计数器来管理软件系统检查等。在这种情况下，窗口看门狗定时器将永远不会复位但是可以用于其他地方。

通过将WWDGT_STAT寄存器的EWIF位写0可以清除EWI中断。

图 15-3. 窗口看门狗定时器时序图



窗口看门狗定时器超时的计算公式如下：

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (13\text{-}1)$$

其中：

t_{WWDGT}：窗口看门狗定时器的超时时间

t_{PCLK1}： APB1以ms为单位的时钟周期

tWWDGT的最大值和最小值请参考*表15-2. 在32MHz（fPCLK1）时的最大/最小超时值*。

表 15-2. 在 32MHz（f_{PCLK1}）时的最大/最小超时值

预分频系数	PSC[3:0]	最小超时 CNT[6:0] =0x40	最大超时 CNT[6:0]=0x7F
1 / 1	0000	128μs	8.192ms
1 / 2	0001	256μs	16.384ms
1 / 4	0010	512μs	32.768ms
1 / 8	0011	1.024ms	65.536ms
1 / 16	0100	2.048ms	131.072ms
1 / 32	0101	4.096ms	262.144ms
1 / 64	0110	8.192ms	524.288ms
1 / 128	0111	16.384ms	1048.576ms
1 / 256	1000	32.768ms	2097.152ms

1 / 512	1001	65.536ms	4194.304ms
1 / 1024	1010	131.072ms	8388.608ms
1 / 2048	1011	262.144ms	16777.216ms
1 / 4096	1100	524.288ms	33554.432ms
1 / 8192	1101	1048.576ms	67108.864ms
1 / 1	1110	128μs	4.096ms
1 / 1	1111	128μs	4.096ms

如果MCU调试模块中的WWDGT_HOLD位被清0，即使Cortex™-M23内核停止工作（调试模式下），窗口看门狗定时器也可以继续工作，当WWDGT_HOLD位被置1时，窗口看门狗定时器在调试模式下停止计数。

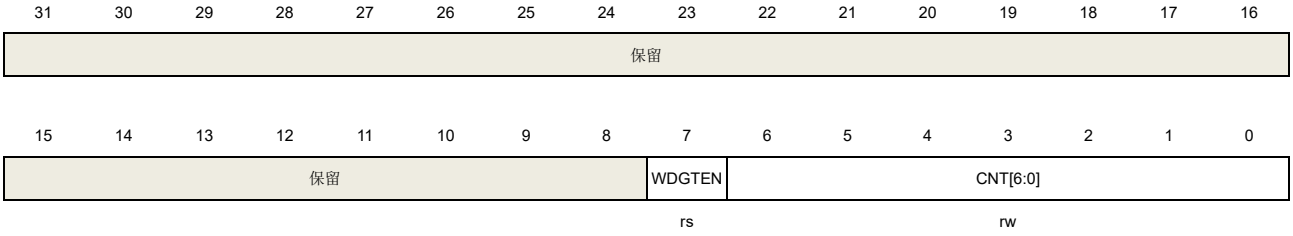
15.2.4. WWDGT 寄存器

WWDGT 基地址：0x4000 2C00

控制寄存器（WWDGT_CTL）

地址偏移：0x00
复位值：0x0000 007F

该寄存器可以按半字（16 位）或字（32 位）访问。

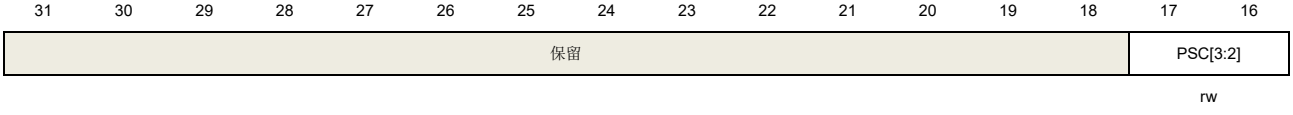


位/位域	名称	说明
31:8	保留	必须保持复位值。
7	WDGTEN	开启窗口看门狗定时器，硬件复位的时候清0，写0无效。 0：关闭窗口看门狗定时器。 1：开启窗口看门狗定时器。
6:0	CNT[6:0]	看门狗定时器计数器的值。当计数值从0x40降到0x3F时，产生看门狗定时器复位。当计数器值高于窗口值的时候，写计数器可以产生看门狗定时器系统复位。

配置寄存器（WWDGT_CFG）

地址偏移：0x04
复位值：0x0000 007F

该寄存器可以按半字（16 位）或字（32 位）访问。



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						EWIE	PSC[1:0]		WIN[6:0]						
						rs	rw		rw						

位/位域	名称	说明
31:18	保留	必须保持复位值。
17:16	PSC[3:2]	预分频器，这些位和PSC[1:0]共同决定看门狗定时器的时间基准。 0000：(PCLK1 / 4096) / 1 0001：(PCLK1 / 4096) / 2 0010：(PCLK1 / 4096) / 4 0011：(PCLK1 / 4096) / 8 0100：(PCLK1 / 4096) / 16 0101：(PCLK1 / 4096) / 32 0110：(PCLK1 / 4096) / 64 0111：(PCLK1 / 4096) / 128 1000：(PCLK1 / 4096) / 256 1001：(PCLK1 / 4096) / 512 1010：(PCLK1 / 4096) / 1024 1011：(PCLK1 / 4096) / 2048 1100：(PCLK1 / 4096) / 4096 1101：(PCLK1 / 4096) / 8192 1110：(PCLK1 / 4096) / 1 1111：(PCLK1 / 4096) / 1
15:10	保留	必须保持复位值。
9	EWIE	提前唤醒中断使能。如果该位被置1，计数值达到0x40时触发中断。该位由硬件复位清0，或通过置位RCU模块的WWDGTRST位进行软件复位。写0没有任何作用。
8:7	PSC[1:0]	预分频器，这些位和PSC[3:2]共同决定看门狗定时器的时间基准。
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器(WWDGT_CTL的CNT位)会产生系统复位。

状态寄存器（WWDGT_STAT）

地址偏移：0x08

复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															EWIF
															rc_w0

位/位域	名称	说明
31:1	保留	必须保持复位值。
0	EWIF	提前唤醒中断标志位。当计数值达到0x40，即使中断没有被使能（WWDGT_CFG中的EWIE位为0）该位也会被硬件置1。这个bit可以通过写0清零，写1无效。

16. 实时时钟（RTC）

16.1. 简介

RTC 模块提供了一个包含日期（年/月/日）和时间（时/分/秒/亚秒）的日历功能。除亚秒用二进制码显示外，时间和日期都以 BCD 码的形式显示。RTC 可以进行夏令时补偿。RTC 可以工作在省电模式下，并通过软件配置来智能唤醒。RTC 支持外接更高精度的低频时钟，用以达到更高的日历精度。

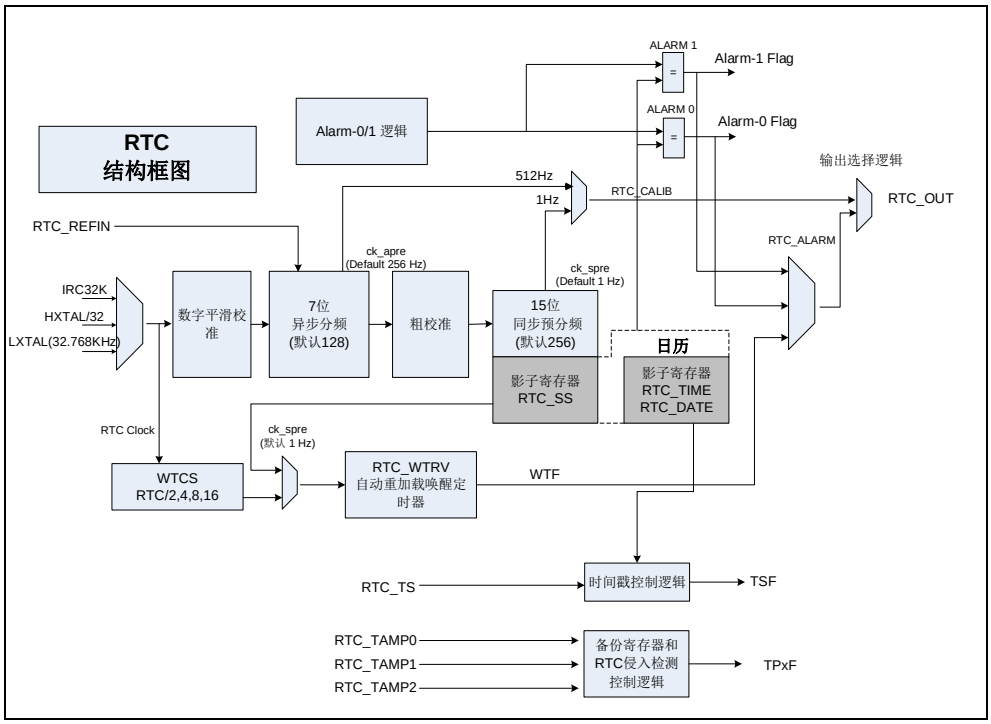
16.2. 主要特征

- 通过软件设置来实现夏令时补偿。
- 参考时钟检测功能：通过外接更高精度的低频率时钟源（50Hz或60Hz）来提高日历精度。
- 数字校准功能：通过调整最小时间单位（最大可调精度0.95ppm）来进行日历校准。
- 通过移位功能进行亚秒级调整。
- 记录事件时间的时间戳功能。
- 三个模式可配置的独立的侵入检测。
- 可编程的日历和一个位域可屏蔽的闹钟。
- 可屏蔽的中断源：
  - 闹钟 0 和闹钟 1；
  - 时间戳检测；
  - 侵入检测；
  - 自动唤醒
- 5个32位（共20字节）通用备份寄存器，能够在省电模式下保存数据。当有外部事件侵入时，备份寄存器将会复位。

16.3. 功能描述

16.3.1. 结构框图

图 16-1. RTC 结构框图



RTC 单元包括：

- 闹钟事件/中断。
- 侵入事件/中断。
- 32位备份寄存器。
- 可选的RTC输出功能：
  - 512Hz（默认预分频值）：RTC_OUT(PC13/PB2/PB14)；
  - 1Hz（默认预分频值）：RTC_OUT(PC13/PB2/PB14)；
  - 闹钟事件（极性可配置）：RTC_OUT(PC13/PB2/PB14)；
  - 自动唤醒事件（极性可配置）：RTC_OUT(PC13/PB2/PB14)。
- 可选的RTC输入功能：
  - 时间戳事件检测：RTC_TS(PC13)；
  - 侵入事件检测 0：RTC_TAMP0(PC13)；
  - 侵入事件检测 1：RTC_TAMP1(PA0)；
  - 侵入事件检测 2：RTC_TAMP2(PA2)；
  - 参考时钟输入：RTC_REFIN(PB15)。

16.3.2. 时钟源和预分频

RTC 单元有三个可选的独立时钟源：LXTAL、IRC32K 和 HXTAL 的 32(由 RCU_CFG 寄存器配置)分频后的时钟。

在 RTC 单元，有两个预分频器用来实现日历功能和其他功能。一个分频器是 7 位异步预分频器，另一个是 15 位同步预分频器。异步分频器主要用来降低功率消耗。如果两个分频器都被使用，建议异步分频器的值尽可能大。

两个预分频器的频率计算公式如下：

$$f_{ck\_apre}=\frac{f_{rtcclk}}{FACTOR\_A+1} \tag{16-1}$$

$$f_{ck\_spre}=\frac{f_{ck\_apre}}{FACTOR\_S+1}=\frac{f_{rtcclk}}{(FACTOR\_A+1)*(FACTOR\_S+1)} \tag{16-2}$$

ck_apre 用于为 RTC_SS 亚秒寄存器自减计数器提供时钟，该寄存器值为二进制，表示到达下一秒时间，该寄存器自减到 0 时，自动加载 FACTOR_S 的值。ck_spre 用于为日历寄存器提供时钟，每个时钟增加一秒。

16.3.3. 影子寄存器

当APB总线访问RTC日历寄存器RTC_DATE、RTC_TIME和RTC_SS时，BPSHAD位决定是访问影子寄存器还是真实日历寄存器。默认情况下BPSHAD为0，APB总线访问影子日历寄存器。每两个RTC时钟，影子日历寄存器值会更新为真实日历寄存器的值，与此同时RSYNF位也会再次置位。在Deep-sleep和Standby模式下，影子寄存器不会更新。退出这两种模式时，软件必须清除RSYNF位。如果想要在BPSHAD=0的情况下读日历寄存器的值，须等待RSYNF置1（最大的等待时间是2个RTC时钟周期）。

注意：在BPSHAD=0下，读日历寄存器（RTC_SS，RTC_TIME，RTC_DATE）的APB时钟的频率（f_{apb}）必须至少是RTC时钟频率（f_{rtcclk}）的七倍。

系统复位将复位影子寄存器。

16.3.4. 位域可屏蔽可配置的闹钟

RTC闹钟功能被划分为多个位域并且每一个位域有一个该域的可屏蔽位。

RTC闹钟功能的使能由RTC_CTL寄存器中的ALRMxEN（x=0,1）位控制。当ALRMxEN=1（x=0,1）并且闹钟所有位域的值与对应的日历时间值匹配，ALRMxF（x=0,1）标志位将会置位。

注意：当秒字段未被屏蔽时(RTC_ALRMxTD 寄存器的 MSKS=0)，为确保正常运行，RTC_PSC 寄存器的同步预分频系数（FACTOR_S）应大于等于 3。

如果一个位域被屏蔽，这个位域被认为在逻辑上匹配的。如果所有的位域被屏蔽，在 ALRMxEN 位被置位 3 个 RTC 时钟周期后，ALRMxF 位将置位。

16.3.5. 可配置周期的自动唤醒定时器

RTC具有一个16位的自动递减计数器用来周期性产生唤醒标志

该功能通过WTEN置1来使能，并且可以工作在省电模式。

自动递减计数器有两种可选的时钟来控制：

1. RTC 时钟的 2/4/8/16 分频：
   如果 RTC 时钟为 LXTAL(32.768 KHz)，则唤醒中断周期在 122us 和 32s 之间，分辨率低至 61us。

2. 内部时钟 ck_spre：
   如果 ck_spre 为 1Hz，则唤醒中断周期在 1s 到 36h 之间，分辨率低至 1s。
   - WTCS[2：1] = 0b10，唤醒中断周期在 1s 到 18 h
   - WTCS[2：1] = 0b11，唤醒中断周期在 18h 到 36 h

该功能使能后，计数器自动递减。当计数器到0时，WTF标志位置1，唤醒计数器自动重载RTC_WUT的值。

当WTF置1后，必须软件清除该标志。

如果WTIE被置位，计数器到0时，会产生唤醒中断，从而使系统退出省电模式。系统复位对该功能没有影响。

WTF标志可以从RTC_ALARM通道输出到RTC_OUT。

16.3.6. RTC 初始化和配置

RTC 寄存器写保护

在默认情况下，PMU_CTL寄存器的BKPWEN位被清0。所以写RTC寄存器前需要软件提前设置BKPWEN位。

上电复位后，大多数RTC寄存器是被写保护的。写入这些寄存器的第一步是解锁这些保护。

通过下面的步骤，可以解锁这些保护：
1. 写‘0xCA’到RTC_WPK寄存器；
2. 写 ‘0x53’到RTC_WPK寄存器。

写一个错误的值到RTC_WPK会使写保护再次生效。

备份域复位后，一些RTC寄存器被写保护：RTC_TIME，RTC_DATE，RTC_CTL，RTC_STAT，RTC_PSC，RTC_WUT，RTC_ALRM0TD, RTC_ALRM1TD，RTC_HRFC，RTC_SHIFTCTL，RTC_ALRM0SS, RTC_ALRM1SS。

日历初始化和配置

通过以下步骤可以设置日历和预分频器的值：
1. 设置 INITM 位为 1 进入初始化模式。等待 INITF 位被置 1。
2. 在 RTC_PSC 寄存器中，设置同步和异步预分频器的分频系数。
3. 在影子寄存器（RTC_TIME 和 RTC_DATE）中写初始的日历值，并且通过设置 RTC_CTL 寄存器的 CS 位来配置时间的格式（12 或 24 小时制）。
4. 清除 INITM 位退出初始化模式。

大约4个RTC时钟周期后，真正的日历寄存器将从影子寄存器载入时间和日期的设定值，同时日历计数器将要重新开始运行。

注意：初始化以后如果要读取日历寄存器（BPSHAD=0），软件应该确保RSYNF位已经置1。

YCM标志表明日历是否完成初始化，该标志会硬件检查日历的年份值。

夏令时

通过S1H，A1H和DSM位配置，RTC模块可以支持夏令时补偿调节功能。

当日历正在运行时，S1H和A1H能使日历减去或加上1小时。S1H和A1H功能可以重复设置，可以软件配置DSM位来记录这个调节操作。设置S1H或A1H位后，减或加1小时将在下一秒钟到来时生效。

闹钟功能操作步骤

为了避免意外的闹钟标记置位和亚稳态，闹钟功能的操作应遵循如下流程：

1. 清除寄存器 RTC_CTL 的 ALRMxEN（x=0,1）位，禁用闹钟；
2. 设置 Alarm 寄存器（RTC_ALRMxTD/RTC_ALRMxSS（x=0,1））；
3. 设置寄存器 RTC_CTL 的 ALRMxEN（x=0,1）位，使能闹钟功能。

16.3.7. 读取日历

当 BPSHAD=0 时，读日历寄存器

当BPSHAD=0，从影子寄存器读日历的值。由于同步机制的存在，正常读取日历需要满足一个基本要求：APB1总线时钟频率必须大于或等于RTC时钟频率的7倍。在任何情况下APB1总线时钟的频率都不能低于RTC的时钟频率。

当APB1总线时钟频率低于7倍RTC时钟频率时，日历的读取应该遵守以下流程：

1. 读取两次日历时间和日期寄存器；
2. 如果两次的值相等，那么这个值就是正确的；
3. 如果这两次的值不相等，应该再读一次；
4. 第三次的值可以认为是正确的。

RSYNF每2个RTC时钟周期被置位一次。在这时，影子日历寄存器会更新为真实的日历时间和日期。

为了确保这3个值（RTC_SS，RTC_TIME，RTC_DATE）为同一时间，硬件上采取了如下一致性机制：

1. 读RTC_SS锁定RTC_TIME和RTC_DATE的更新；
2. 读RTC_TIME锁定RTC_DATE的更新；
3. 读 RTC_DATE 解锁 RTC_TIME 和 RTC_DATE 的更新。

如果想在一个很短的时间间隔内（少于2个RTCCLK）读取日历，应先清除RSYNF位并等待其置位后再读取。

下面几种情况，软件须等待RSYNF置位后才能读日历寄存器（RTC_SS，RTC_TIME，RTC_DATE）：

1. 系统复位之后；

2. 日历初始化之后；
3. 一次移位操作之后。

特别是从低功耗模式唤醒后，软件必须清除RSYNF位并等待RSYNF再次置位后才能读取日历寄存器。

当 BPSHAD=1 时，读日历寄存器

当BPSHAD=1，RSYNF位会被硬件清0，读日历寄存器不需考虑RSYNF位。当前真实的日历寄存器值会被直接读取。如此配置的好处是当从低功耗模式(Deep-sleep/Standby模式)唤醒后，软件可以立即获取当前日历寄存器的值而无需加入任何等待延迟(此延迟最大为2个RTC时钟周期)。

由于没有RSYNF位周期性的置位，如果两次读日历寄存器之间出现ck_apre时钟边沿，不同寄存器（RTC_SS/RTC_TIME/RTC_DATE）的值可能并非同一时刻。

另外，如果日历寄存器的值正在发生变化的时刻被APB总线读取，那么有可能APB总线读取的值是不准确的。

为了确保日历值的正确性和一致性，读取时软件须如下操作：连续读取所有日历寄存器的值两次，如果上两次的值是一样的，那么这个值就是一致的且准确的。

16.3.8. RTC 复位

在RTC单元，有两个复位源可用：系统复位和备份域复位。

当系统复位有效时，日历影子寄存器和RTC_STAT寄存器的某些位将要复位到默认值。

备份域复位将会影响下面的寄存器，但系统复位不会对它们产生影响：

- RTC 真实的日历寄存器；
- RTC 控制寄存器（RTC_CTL）；
- RTC 预分频寄存器（RTC_PSC）；
- RTC 高精度频率补偿寄存器（RTC_HRFC）；
- RTC 移位控制寄存器（RTC_SHIFTCTL）；
- RTC 时间戳寄存器（RTC_SSTS/RTC_TTS/RTC_DTS）；
- RTC 侵入寄存器（RTC_TAMP）；
- RTC 备份寄存器（RTC_BKPx）；
- RTC 闹钟寄存器（RTC_ALRMxSS/RTC_ALRMxTD）。

当系统复位或者进入省电模式的时候，RTC单元将会继续运行。但是如果备份域复位，RTC将会停止计数并且所有的寄存器会复位。

16.3.9. RTC 移位功能

当用户有一个高精度的远程时钟而且RTC1Hz时钟（ck_spre）和远程时钟只有一个亚秒级的偏差，RTC单元提供一个称作移位的功能去消除这个偏差来提高秒钟的精确性。

以二进制格式显示亚秒值，RTC运行时该值是递减计数。因此通过增加RTC_SHIFTCTL寄存器的SFS[14：0]的值到RTC_SS同步预分频器计数器值SSC[15：0])或通过增加SFS[14：0]的

值到同步预分频器计数器SSC[15：0]并且同时置位A1S位，能分别延迟或提前下一秒到达的时间。

RTC_SS的最大值取决于RTC_PSC寄存器的FACTOR_S的值。FACTOR_S越大，调整的精度也就越高。

因为1Hz的时钟（ck_spre）由FACTOR_A和FACTOR_S共同产生，越高的FACTOR_S值就意味着越低的FACTOR_A值，同时越低的FACTOR_A意味着越高的功耗。

注意：在使用移位功能之前，软件必须检查 RTC_SS 中 SSC 的第 15 位(SSC[15])并确保该位为 0。写 RTC_SHIFTCTL 寄存器之后，RTC_STAT 寄存器的 SOPF 位将会再次置位。当同步移位操作完成时，SOPF 位被硬件清 0。系统复位不影响 SOPF 位。当 REFEN=0 时，移位操作才能正确的工作。如果 REFEN=1，软件禁止写入 RTC_SHIFTCTL。

16.3.10. RTC 参考时钟检测

RTC参考时钟是另外一种提高RTC秒级精度的方法。为了使能这项功能，需要有一个相对于LXTAL有更高精度的外部参考时钟源（50Hz或60Hz）。

使能这项功能之后(REFEN=1)，每一个秒更新的时钟(1Hz)边沿将与最近的RTC_REFIN参考时钟沿进行对比。在大多数情况下，这两个时钟沿是对齐的。但当两个时钟沿由于LXTAL准确度的原因没有对齐的时候，RTC参考时钟的检测功能会偏移1Hz时钟沿一点相位，使得下一个1Hz时钟沿和参考时钟沿对齐。

当REFEN=1，每一秒前后都会有一个进行检测的时间窗，处于不同的检测状态，时间窗时长也不同。当检测状态处于检测第一个参考时钟边沿时，使用7个ck_apre时长的时间窗，当检测状态处于边沿对齐操作时，使用3个ck_apre时长的时间窗。

无论使用哪一种时间窗，当参考时钟在时间窗中被检测到的时候，同步预分频计数器会被强制重载。当两个时钟（ck_spre和参考时钟）边沿是对齐的，这个重载操作对1Hz日历更新没有任何影响。但是当两个时钟边沿没有对齐时，这个重载操作将会移动ck_spre时钟边沿，以使得ck_spre（1Hz）时钟边沿和参考时钟边沿对齐。

当参考检测功能正在运行中但外部参考时钟消失（在3个ck_apre时长时间窗内没有发现参考时钟边沿），日历也能通过LXTAL继续自动更新。如果这个参考时钟重新恢复，参考时钟检测功能会先用7个ck_apre时长时间窗口去检测参考时钟，然后用3个ck_apre时长时间窗口去调节ck_spre（1Hz）时钟边沿。

注意：使能参考时钟检测功能之前(REFEN=1)，软件必须配置 FACTOR_A 为 0x7F，FACTOR_S 为 0xFF。

待机模式下，参考时钟检测功能不可用。

16.3.11. RTC 数字平滑校准

RTC平滑校准是一种用于校准RTC频率的方法，该方法通过调整校准周期内的RTC时钟脉冲个数的方式来实现校准。

完成一次这种校准相当于在一次校准周期内，RTC时钟的脉冲个数增加或者减少了一定的数目。

这种校准的分辨率大约为0.954ppm，范围为从-487.1ppm到+488.5ppm。

校准周期的时间可以配置到 $2^{20}/2^{19}/2^{18}$ RTC 时钟周期，如果 RTC 的输入频率是 32.768KHz，这些校准周期时间分别代表 32/16/8 秒。

高精度频率补偿寄存器(RTC_HRFC)指定了在校准周期内要屏蔽的RTC时钟数目，CMSK[8：0]位能屏蔽0到511个RTC时钟，这样RTC的频率最多降低487.1PPM。

为了提高RTC频率可以设置FREQI位。如果FREQI位被置位，将会有512个额外的RTC时钟周期增加到校准周期(32/16/8 秒)时间期间，这意味着每$2^{11}/2^{10}/2^{9}$ RTC时钟插入一个RTC时钟周期。

因此使用FREQI可以使RTC频率增加488.5ppm。

同时使用CMSK和FREQI，每个周期时间可以调整-511到+512个RTC时钟周期。这意味着在0.954ppm分辨率的情况下，调整范围为从-487.1ppm到+488.5ppm。

当数字平滑校准功能正在运行时，按如下公式计算输出校准频率：

$$f_{cal}=f_{rtcclk} \times(1+\frac{FREQI\times512-CMSK}{2^{N}+CMSK-FREQI\times512}) \quad (16\text{-}3)$$

注意： N=20/19 /18 (32/16/8 秒)校准时间周期。

当 FACTOR_A < 3 时校准：

当异步预分频器值(FACTOR_A)被设置小于3时，若要使用校准功能，软件不能将FREQI位设置为1。当FACTOR_A<3，FREQI位设置将会被忽略。

当FACTOR_A小于3时，FACTOR_S值应小于标称值。假设RTC时钟频率是正常的32.768KHz，对应的FACTOR_S应该按下面所示设置：

FACTOR_A = 2：FACTOR_S减少2(8189)
FACTOR_A = 1：FACTOR_S减少4(16379)
FACTOR_A = 0：FACTOR_S减少8(32759)
当FACTOR_A小于3，CMSK为0x100，校准频率公式如下：

$$f_{cal}=f_{rtcclk} \times(1+\frac{256-CMSK}{2^{N}+CMSK-256}) \quad (16\text{-}4)$$

注意： N=20/19 /18 (32/16/8 秒)校准时间周期。

验证 RTC 校准

提供1Hz校准时钟的输出用于协助软件测量并验证RTC的精度。

在有限的测量周期内测量RTC的频率，最高可能发生2个RTCCLK的测量误差。

为了消除这一测量误差，测量周期应该和校准周期一致。

- 校准周期设为32秒(默认配置)
  用准确的32秒周期去测量1Hz校准输出的准确性能保证这个测量误差在0.477ppm（在32秒周期内0.5个RTCCLK）之内。
- 校准周期设为16秒（通过设置CWND16位）
  使用此配置，CMSK[0]被硬件置0。

用准确的16秒周期去测量1Hz校准输出的准确性能保证这个测量误差在0.954ppm（在16秒周期内0.5个RTCCLK）之内。

- 校准周期设为8秒（通过设置CWND8位）
  使用此配置，CMSK[1：0]被硬件置0。
  用准确的8秒周期去测量1Hz校准输出的准确性能保证这个测量误差在1.907ppm(在8秒周期内0.5个RTCCLK)之内。

运行中重校准

当INITF位是0，用下面的步骤，软件可以更新RTC_HRFC：

1). 等待SCPF位置0；
2). 写一个新的值到RTC_HRFC寄存器；
3). 3 个 ck_apre 时钟周期之后，新的校准设置开始生效。

16.3.12. 时间戳功能

时间戳功能由RTC_TS管脚输入，通过配置TSEN位来使能，也可以通过ITSEN使能。

当RTC_TS管脚检测到时间戳事件发生时，会将日历的值保存在时间戳寄存器中（RTC_DTS/RTC_TTS/RTC_SSTS），同时时间戳标志（TSF）也将由硬件置1。如果时间戳中断使能被启用（TSIE），时间戳事件会产生一个中断。

当检测到内部时间戳事件发生时，会将日历的值保存在时间戳寄存器中（RTC_DTS/RTC_TTS/RTC_SSTS），同时时间戳标志（TSF）和内部时间戳标志（ITSF）也将由硬件置1。如果时间戳中断使能被启用（TSIE），时间戳事件会产生一个中断。当切换到V_{BAT}供电时会产生内部时间戳事件。

时间戳寄存器只会在时间戳事件第一次发生的时刻（TSF=0）记录日历时间，而当TSF=1时，时间戳事件的值不会被记录。

RTC模块提供了一个可选的功能特性，来增加时间戳事件的触发源：设置TPTS=1，使得侵入检测功能的侵入事件同时也作为时间戳事件的输入源。

注意：因为同步机制的原因，当时间戳事件发生时，TSF 会延迟 2 个 ck_apre 周期置位。

16.3.13. 侵入检测

RTC_TAMPx管脚可以作为侵入事件检测功能输入管脚，检测模式有两种可供用户选择：边沿检测模式或者是带可配置滤波功能的电平检测模式。

入侵检测的配置可以用于以下目的：

1. 默认配置会擦除RTC备份寄存器
2. 可以从深度睡眠模式和待机模式唤醒并产生中断
3. 为低功耗定时器产生硬件触发

RTC 备份寄存器（RTC_BKPx）

RTC备份寄存器处于VDD备份域中，即使VDD电源被切断，该区域的寄存器的电源还可由VBAT提供。从待机模式唤醒或系统复位操作都不会影响这些寄存器。

只有当被检测到有侵入事件和备份域复位时，这些寄存器复位，除非当 TPxNOERASE 被置位，或者 RTC_TAMP 寄存器的 TPxMASK 置位时，这些寄存器不会复位。

初始化侵入检测功能

TPxEN位可以独立使能对应于不同管脚上的RTC侵入检测功能。使能TPxEN位启动侵入检测功能之前，需要设置好侵入检测的配置。

当TPxMASK =0时：

TPxF标志会在引脚上出现侵入事件后置位，并存在以下延迟：

1. 当FLT不为0x0时（带可配置滤波的电平检测），延迟为3个ck_apre 周期
2. 当 TPTS=1时（入侵事件的时间戳），延迟为3个ck_apre 周期
3. 当 FLT=0x0（边沿检测）且TPTS=0 时，无延迟

在此期间，只要TPxF置1，就无法检测到同一引脚上出现的新入侵。

当TPxMASK =1时：

当在上述延迟期间TPxF置位以及2.5个ck_rtc附加周期内，无法检测到同一引脚上出现的新入侵。

通过将RTC_TAMP寄存器中的TPIE位置1，可在发生入侵检测事件时（当TPxF置1时）生成中断。当一个或多个TPxMASK置1时，不允许将TPIE置1。

将TPIE清零时，可通过相应的TPxIE位置1来分别使能各个入侵引脚事件中断。当相应的TPxMASK置1时，不允许将TPxIE置1。

入侵事件产生触发输出

对于低功耗 TIMER，可以将入侵事件作为触发输入。

为了在同一个引脚检测新的入侵事件，当 TPxMASK =0 时，TPxF 标志需要由软件清零。

当 TPxMASK = 1 时，TPxF 标志会被屏蔽，并在 RTC_STAT 寄存器中保持清零。此配置无需系统唤醒来清零 TPxF，可以在深度睡眠模式下自动触发低功耗 TIMER 定时器。备份寄存器在这种情况下不清零。

侵入事件源的时间戳

使能TPTS位，能让侵入检测功能被用作时间戳功能。如果这位被设置为1，当检测到侵入事件时，TSF也将会被置位，如同使能了时间戳功能。当检测到侵入事件时，无论TPTS位的值如何，TPxF位将置位。

侵入事件检测为边沿检测模式

当FLT位为0x0时，侵入检测被设置成边沿检测模式，TPxEG位决定检测沿是上升沿还是下降沿。当侵入检测配置为边沿检测模式时，侵入检测输入管脚上的上拉电阻将会被禁用。

由于检测侵入事件会复位备份寄存器（RTC_BKPx），因此对备份寄存器写操作时应该确保侵入事件导致的复位和写操作不会同时发生。避免这种情形的推荐方法是先关闭侵入检测功能，在完成写操作后再重新启动该功能。

注意：Tamper 上的侵入检测功能即使 V_{DD} 电源被关掉也依然可以运行。

侵入事件检测为带可配置滤波功能的电平检测模式

当FLT位没有被设置成0x0时，侵入检测被设置成电平检测模式，FLT位决定有效电平需连续采样的次数（2，4或者8）。

当DISPU被设置成0(默认值)，内部的上拉电阻将会在每一次采样前预充电侵入管脚，这样侵入事件的输入管脚上就允许连接更大的电容。预充电的时间可以通过PRCH位来配置。越大的电容，所需的充电时间越长。

电平检测模式下每次采样之间的时间间隔是可配置的。通过调整采样频率(FREQ)，软件能在功耗和检测延迟之间取得一个平衡。

16.3.14. 校准时钟输出

如果COEN位设置为1，RTC_OUT会输出参考校准时钟。

当COS位设置为0（默认值）并且异步预分频器（FACTOR_A）设为0x7F时，RTC_CALIB的频率是f_{rtcclk}/64。因此若RTCCLK的频率为32.768KHz，RTC_CALIB对应的输出为512Hz。因为下降沿存在轻微的抖动，因此推荐使用RTC_CALIB输出的上升沿。

当COS位设置为1时，RTC_CALIB的频率计算公式为：

$$f_{rtc\_calib} = \frac{f_{rtcclk}}{(FACTOR\_A+1)\times(FACTOR\_S+1)} \quad (16\text{-}5)$$

若RTCCLK为32.768KHz，如果预分频器是默认值，那么RTC_CALIB对应的输出是1Hz。

16.3.15. 闹钟输出

当OS控制位不为0x00时，RTC_ALARM复用输出功能被启用。这个功能将直接输出RTC_STAT寄存器的闹钟标志ALRMxF或者自动唤醒标志WTF。

RTC_CTL寄存器中的OPOL位可以配置ALRMxF标志或者WTF标志输出时候的极性，因此RTC_ALARM的输出电平有可能与相应的位值相反。

16.3.16. RTC 引脚配置

RTC_OUT，RTC_TS和RTC_TAMP0都使用同一个PC13引脚。无论PC13的GPIO是什么配置，PC13的功能由RTC控制。PC13的RTC功能可以用于所以低功耗模式和Vbat模式

PC13的输出优先级如表*表16-1. RTC PC13引脚配置*

表 16-1. RTC PC13 引脚配置

功能配置 和引脚功能	OS[1:0] (输出选择)	COEN (校准输出)	TP0EN (侵入检测0使能)	TSEN (时间戳使能)	ALARMOUTTYPE (闹钟输出类型)
闹钟开漏输出	01 或 10 或 11	-	-	-	0
闹钟推挽输出	01 或 10 或 11	-	-	-	1
校准推挽输出	00	1	-	-	-
TAMP0 浮空输入	00	0	1	0	-
时间戳和TAMP0浮空输入	00	0	1	1	-
时间戳浮空输入	00	0	0	1	-
标准 GPIO	00	0	0	0	-

配置RTC_CTL寄存器的OUT2EN位，可以在PB2/PB14引脚输出RTC_OUT信号，但在VBAT模式下不可用。

表 16-2. 所有低功耗模式下 RTC 功能

引脚	RTC功能	除待机模式的所有低功耗模式	待机模式	VBAT模式
PC13	RTC_TAMP0 RTC_TS RTC_OUT	是	是	是
PA0	RTC_TAMP1	是	是	是
PA2	RTC_TAMP2	是	是	是
PB2	RTC_OUT	是	否	否
PB14	RTC_OUT	是	否	否
PB15	RTC_REFIN	是	否	否

16.3.17. RTC 省电模式管理

表 16-3. 省电模式管理

模式	模式下能否工作	退出该模式的方法
睡眠模式	是	RTC中断
睡眠模式1	是	RTC中断
低功耗睡眠模式	是	RTC中断
深度睡眠	当时钟源是LXTAL或IRC32K时可以工作	RTC中断
深度睡眠1	当时钟源是LXTAL或IRC32K时可以工作	RTC中断
深度睡眠2	当时钟源是LXTAL或IRC32K时可以工作	RTC中断
待机模式	当时钟源是LXTAL或IRC32K时可以工作	RTC闹钟/侵入事件/时间戳事件/唤醒

16.3.18. RTC 中断

所有的RTC中断都被连接到EXTI控制器。

如果想使用RTC闹钟/侵入事件/时间戳中断/自动唤醒中断，应按下面步骤操作：

1. 设置并使能对应的 EXTI 中连接到 RTC 闹钟/侵入事件/时间戳/自动唤醒的中断线，然后配置该线为上升沿触发模式；
2. 配置并使能 RTC 闹钟/侵入事件/时间戳/自动唤醒的全局中断；
3. 配置并使能 RTC 闹钟/侵入事件/时间戳功能。

表 16-4. RTC 中断控制

中断	事件标志(	控制位	退出睡眠模式	退出深度睡眠模式和待机模式
闹钟0	ALRM0F	ALRM0IE	Y	Y(1)
闹钟1	ALRM1F	ALRM1IE	Y	Y(1)
唤醒	WTF	WTIE	Y	Y(1)
时间戳	TSF	TSIE	Y	Y(1)
侵入0	TP0F	TPIE	Y	Y(1)
侵入1	TP1F	TPIE	Y	Y(1)
侵入2	TP2F	TPIE	Y	Y(1)

(1). 仅当RTC时钟源为LXTAL或IRC32K时，才可以从深度睡眠和待机模式唤醒。

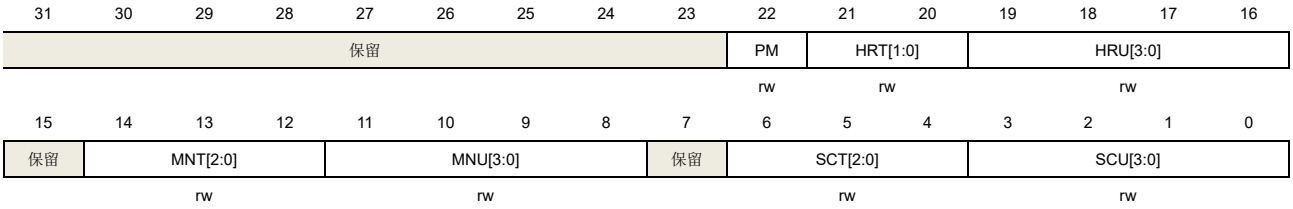
16.4. RTC 寄存器

RTC基地址：0x4000 2800

16.4.1. 时间寄存器（RTC_TIME）

偏移地址：0x00
系统复位值：当BPSHAD = 0，0x0000 0000
当BPSHAD = 1，无影响
写保护寄存器，仅在初始化状态可以进行写操作

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PM	AM/PM 标志 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	保留	必须保持复位值。
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	保留	必须保持复位值。
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

16.4.2. 日期寄存器（RTC_DATE）

偏移地址：0x04
系统复位值：当 BPSHAD = 0，0x0000 2101
当 BPSHAD = 1，无影响
写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								YRT[3:0]				YRU[3:0]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOW[2:0]			MONT	MONU[2:0]				保留		DAYT		DAYU			
rw			rw	rw						rw		rw			

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:20	YRT[3:0]	年份十位值，以 BCD 码形式存储
19:16	YRU[3:0]	年份个位值，以 BCD 码形式存储
15:13	DOW[2:0]	星期 0x0：保留 0x1：星期一 … 0x7：星期日
12	MONT	月份十位值，以 BCD 码形式存储
11:8	MONU[2:0]	月份个位值，以 BCD 码形式存储
7:6	保留	必须保持复位值。
5:4	DAYT[1:0]	日期十位值，以 BCD 码形式存储
3:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储

16.4.3. 控制寄存器（RTC_CTL）

偏移地址：0x08
系统复位：无影响
备份域复位值：0x0000 0000
写保护寄存器

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUT2EN	保留						ITSEN	COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
rw							rw	rw	rw		rw	rw	rw	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WTIE	ALRM1IE	ALRM0IE	TSEN	WTEN	ALRM1EN	ALRM0EN	保留	CS	BPSHAD	REFEN	TSEG	WTCS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		

位/位域	名称	描述
31	OUT2EN	RTC_OUT 引脚选择 0：RTC_OUT 输出到 PC13 1：RTC_OUT 输出到 PB2/PB14

30:25	保留	必须保持复位值。
24	ITSEN	内部时间戳事件使能 0：关闭内部时间戳事件 1：使能内部时间戳事件
23	COEN	校准输出使能 0：关闭校准输出 1：使能校准输出
22:21	OS[1:0]	输出选择 该位用来选择输出的标志源。 0x00：禁用 RTC_ALARM 输出 0x01：启用闹钟 0 标志输出 0x10：启用闹钟 1 标志输出 0x11：启用唤醒标志输出
20	OPOL	输出极性 该位用来反转 RTC_ALARM 输出。 0：禁用反转 RTC_ALARM 输出 1：启用反转 RTC_ALARM 输出
19	COS	校准输出选择 仅当 COEN=1 并且预分频器是默认值时有效。 0：校准输出是 512Hz 1：校准输出是 1Hz
18	DSM	夏令时屏蔽位 该位可以通过软件灵活使用。常用来记录夏令时调整。
17	S1H	减 1 小时(冬季时间变化) 当前时间非零的情况下，将当前时间减去一个小时。 0：没有影响 1：在下一个秒改变时，将减少一个小时
16	A1H	增加 1 小时(夏季时间变化) 将当前时间增加一个小时。 0：没有影响 1：在下一个秒改变时，将增加一个小时
15	TSIE	时间戳中断使能 0：禁用时间戳中断 1：启用时间戳中断
14	WTIE	自动唤醒定时器中断使能 0：禁用自动唤醒定时器中断 1：启用自动唤醒定时器中断
13	ALRM1IE	RTC 闹钟 1 中断使能 0：禁用闹钟中断

		1：启用闹钟中断
12	ALRM0IE	RTC 闹钟 0 中断使能 0：禁用闹钟中断 1：启用闹钟中断
11	TSEN	时间戳功能使能 0：禁用时间戳功能 1：启用时间戳功能
10	WTEN	自动唤醒定时器功能使能 0：禁用自动唤醒定时器功能 1：启用自动唤醒定时器功能
9	ALRM1EN	闹钟 1 功能使能 0：禁用闹钟功能 1：启用闹钟功能
8	ALRM0EN	闹钟 0 功能使能 0：禁用闹钟功能 1：启用闹钟功能
7	保留	必须保持复位值。
6	CS	时间格式 0：24 小时制 1：12 小时制 注意：仅能在初始化状态进行写入
5	BPSHAD	禁止影子寄存器 0：读取的日历的值来自影子日历寄存器 1：读取的日历的值来自真正日历寄存器 注意：如果 APB1 时钟的频率小于 RTCCLK 频率的 7 倍，该位必须设为 1
4	REFEN	参考时钟检测功能使能 0：禁用参考时钟检测功能 1：启用参考时钟检测功能 注意：仅能在初始化状态进行写入并且 FACTOR_S 必须为 0x00FF
3	TSEG	时间戳事件有效检测边沿 0：上升沿是时间戳事件有效检测沿 1：下降沿是时间戳事件有效检测沿
2:0	WTCS[2:0]	自动唤醒定时器时钟选择 0x0：RTC 时钟的 16 分频 0x1：RTC 时钟的 8 分频 0x2：RTC 时钟的 4 分频 0x3：RTC 时钟的 2 分频 0x4，0x5：ck_spre (默认 1Hz)时钟 0x6，0x7：ck_spre (默认 1Hz)时钟并且将唤醒计数器值增加 216

16.4.4. 状态寄存器（RTC_STAT）

偏移地址：0x0C
系统复位：仅INITM，INITF和RSYNF位被置0，其他位无影响。
备份域复位值：0x0000 0007
写保护寄存器

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														ITSF	SCPF
														rc_w0	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP2F	TP1F	TP0F	TSOVRF	TSF	WTF	ALRM1F	ALRM0F	INITM	INITF	RSYNF	YCM	SOPF	WTWF	ALRM1WF	ALRM0WF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

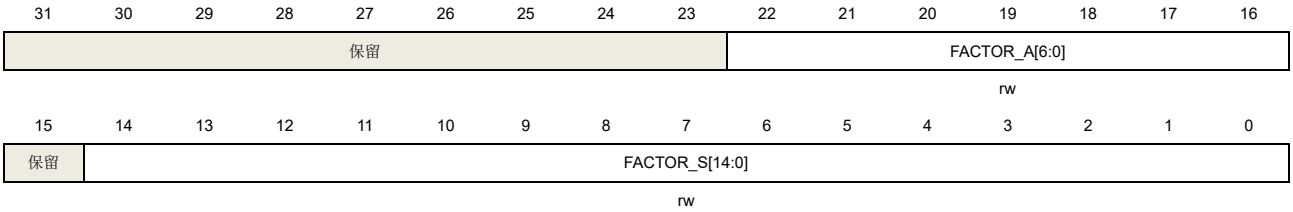
位/位域	名称	描述
31:18	保留	必须保持复位值。
17	ITSF	内部时间戳标志 当检测到内部时间戳事件时，该位硬件置 1。可以通过向该位软件写 0 来清除，并且和 TSF 位一起清零。
16	SCPF	平滑校准挂起标志 在未进入初始化模式时向 RTC_HRFC 进行软件写操作，该位被硬件置 1。当平滑校准设置开始执行后，该位被硬件清零 0。
15	TP2F	RTC_TAMP2 事件标志 当在 tamper2 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。
14	TP1F	RTC_TAMP1 事件标志 当在 tamper1 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。
13	TP0F	RTC_TAMP0 事件标志 当在 tamper0 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。
12	TSOVRF	时间戳事件溢出标志 如果 TSF 位已经置位，当再次检测到时间戳事件时，该位会通过硬件置 1。 可以通过向该位软件写 0 来清除。
11	TSF	时间戳事件标志 当检测到一个时间戳事件时，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
10	WTF	唤醒定时器标志 当唤醒定时器减到 0 时，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。 该标志需要在 WTF 位再次置 1 之前的 1.5 个 RTC 时钟周期前完成软件清除该位。

8	ALRM1F	Alarm1 发生标志 当现在的时间/日期与闹钟 1 设置的时间/日期匹配的时候，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
8	ALRM0F	Alarm0 发生标志 当现在的时间/日期与闹钟 0 设置的时间/日期匹配的时候，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
7	INITM	进入初始化模式 0：自由运行模式 1：进入初始化模式设置时间/日期和预分频，计数器将停止运行
6	INITF	初始化状态标志 该位被硬件置 1，初始化状态时可以设置日历寄存器和预分频器。 0：日历寄存器和预分频器的值不能改变 1：日历寄存器和预分频器的值可以改变
5	RSYNF	寄存器同步标志 每 2 个 RTCCLK 将会由硬件置 1 一次，同时会复制当前日历时间/日期到影子日历寄存器。初始化模式（INITM），移位操作挂起标志（SOPF）或者禁止影子寄存器模式（BPSHAD = 1）会清除该位。该位也可以通过软件写 0 清除。 0：影子寄存器未同步 1：影子寄存器已同步
4	YCM	年份配置标志 当日历寄存器的年份值不为 0 时硬件置 1 0：日历尚未初始化 1：日历已经初始化
3	SOPF	移位功能操作挂起标志 0：移位操作没有挂起 1：移位操作挂起
2	WTWF	唤醒定时器可写标志 0：不允许更新唤醒定时器 1：允许更新唤醒定时器
1	ALRM1WF	Alarm1 配置可写标志 硬件置位和清零。ALRM1EN=0 时，标记 alarm 是否可写。 0：不允许修改 Alarm 寄存器设置 1：允许修改 Alarm 寄存器设置
0	ALRM0WF	Alarm0 配置可写标志 硬件置位和清零。ALRM0EN=0 时，标记 alarm 是否可写。 0：不允许修改 Alarm 寄存器设置 1：允许修改 Alarm 寄存器设置

16.4.5. 预分频寄存器（RTC_PSC）

偏移地址：0x10
系统复位：无影响
备份域复位值：0x007F 00FF
写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32位)访问。

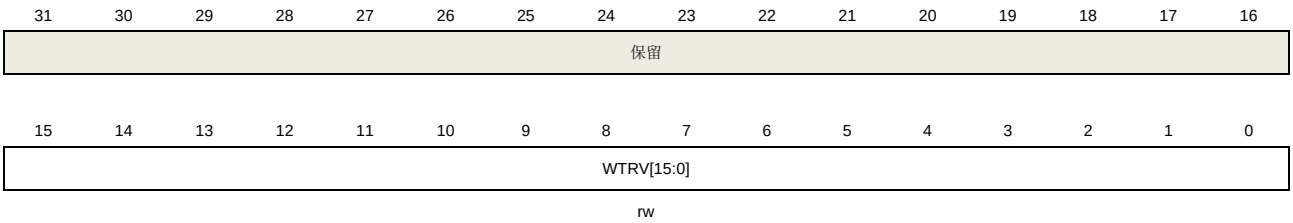


位/位域	名称	描述
31:23	保留	必须保持复位值。
22:16	FACTOR_A[6:0]	异步预分频系数 ck_apre 频率 = RTCCLK 频率/(FACTOR_A+1)
15	保留	必须保持复位值。
14:0	FACTOR_S[14:0]	同步预分频系数 ck_spre 频率 = ck_apre 频率/(FACTOR_S+1)

16.4.6. 唤醒定时器寄存器（RTC_WUT）

偏移地址：0x14
系统复位：无影响
备份域复位值：0x0000 FFFF
写保护寄存器

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	WTRV[15:0]	自动唤醒定时器重载值 当 WTEN 置 1 时，每隔 (WTRV[15：0]+1) 个 ck_wut 周期，WTF 置 1 一次 。 ck_wut 通过 WTCS[2：0] 位选择.

注意： 禁止在 WTCS[2：0]=0b 011 时配置 WTRV=0x0000。
该寄存器仅在 WTWF=1 时才能写操作

16.4.7. 闹钟 0 时间日期寄存器（RTC_ALRM0TD）

偏移地址：0x1C
系统复位：无影响
备份域复位值：0x0000 0000
写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSKD	DOWS	DAYT[1:0]		DAYU[3:0]				MSKH	PM	HRT[1:0]		HRU[3:0]			
rw	rw	rw		rw				rw	rw	rw		rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSKM	MNT[2:0]			MNU[3:0]				MSKS	SCT[2:0]			SCU[3:0]			
rw	rw			rw				rw	rw			rw			

位/位域	名称	描述
31	MSKD	闹钟日期位域屏蔽位 0：不屏蔽日期/天位域 1：屏蔽日期/天位域
30	DOWS	星期选择 0：此时 DAYU[3：0]代表日期个位值 1：此时 DAYU[3：0]代表星期几，此时 DAYT[1：0]无意义
29:28	DAYT[1:0]	日期十位值，以 BCD 码格式存储
27:24	DAYU[3:0]	日期个位值或星期天数，以 BCD 码格式存储
23	MSKH	闹钟小时位域屏蔽位 0：不屏蔽小时位域 1：屏蔽小时位域
22	PM	AM/PM 标志 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	MSKM	闹钟分钟位域屏蔽位 0：不屏蔽分钟位域 1：屏蔽分钟位域
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储

11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	MSKS	闹钟秒位域屏蔽位 0：不屏蔽秒位域 1：屏蔽秒位域
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

16.4.8. 闹钟 1 时间日期寄存器（RTC_ALRM1TD）

偏移地址：0x20
系统复位：无影响
备份域复位值：0x0000 0000
写保护寄存器，仅在初始化状态可以进行写操作。

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSKD	DOWS	DAYT[1:0]		DAYU[3:0]				MSKH	PM	HRT[1:0]		HRU[3:0]			
rw	rw	rw		rw				rw	rw	rw		rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSKM	MNT[2:0]			MNU[3:0]				MSKS	SCT[2:0]			SCU[3:0]			
rw	rw			rw				rw	rw			rw			

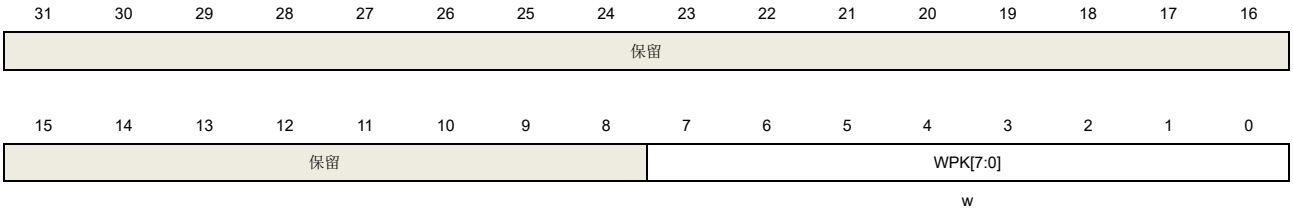
位/位域	名称	描述
31	MSKD	闹钟日期位域屏蔽位 0：不屏蔽日期/天位域 1：屏蔽日期/天位域
30	DOWS	星期选择 0：此时 DAYU[3：0] 代表日期个位值 1：此时 DAYU[3：0] 代表星期几，此时 DAYT[1：0]无意义
29:28	DAYT[1:0]	日期十位值，以 BCD 码格式存储
27:24	DAYU[3:0]	日期个位值或星期天数，以 BCD 码格式存储
23	MSKH	闹钟小时位域屏蔽位 0：不屏蔽小时位域 1：屏蔽小时位域
22	PM	AM/PM 标志 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储

15	MSKM	闹钟分钟位域屏蔽位 0：不屏蔽分钟位域 1：屏蔽分钟位域
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	MSKS	闹钟秒位域屏蔽位 0：不屏蔽秒位域 1：屏蔽秒位域
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

16.4.9. 写保护钥匙寄存器（RTC_WPK）

偏移地址：0x24
复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

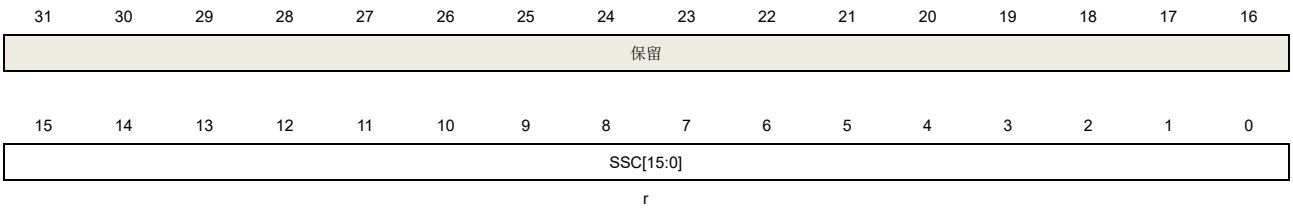


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	WPK[7:0]	写保护的解锁值

16.4.10. 亚秒寄存器（RTC_SS）

偏移地址：0x28
系统复位值：当BPSHAD = 0，0x0000 0000。
当BPSHAD = 1，无影响。

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SSC[15:0]	亚秒值 该位值是同步预分频计数器的值。秒的小数部分由下面公式给出： 秒的小数部分 = (FACTOR_S - SSC)/(FACTOR_S + 1)

16.4.11. 移位控制寄存器（RTC_SHIFTCTL）

偏移地址：0x2C
系统复位：无影响
备份域复位值：0x0000 0000
写保护寄存器，仅当SOPF=0，该寄存器可写。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A1S	保留														
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SFS[14:0]														
	w														

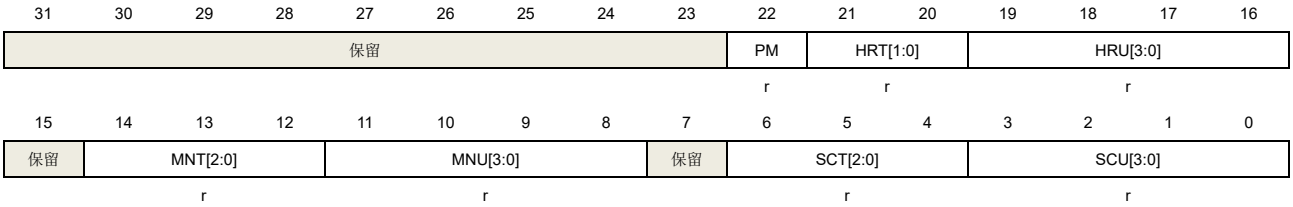
位/位域	名称	描述
31	A1S	增加一秒 0：无影响 1：增加一秒到时钟/日历 该位与 SFS 位一起使用，增加小于一秒到当前时间。
30:15	保留	必须保持复位值。
14:0	SFS[14:0]	减去小于一秒的一段时间 这位的值将增加到同步预分频计数器 当仅用 SFS 时，由于同步预分频器是一个递减计数器，所以时钟将会延迟。 延迟(秒) = SFS/(FACTOR_S + 1) 当 A1S 和 SFS 一起使用时，时钟将会提前 提前(秒) = (1 - (SFS/(FACTOR_S + 1)))

注意：写入此寄存器会导致 RSYNF 位被清 0。

16.4.12. 时间戳时间寄存器（RTC_TTS）

偏移地址：0x30
备份域复位值：0x0000 0000
系统复位：无影响
当TSF被置1，该位用来记录日历时间。
清除TSF位也会清除此寄存器。

该寄存器只能按字(32 位)访问。

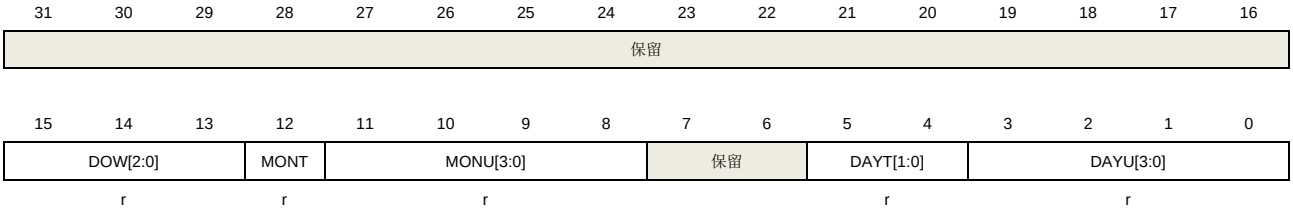


位/位域	名称	描述
31:23	保留	必须保持复位值。
22	PM	AM/PM 标记 0：AM 或 24 小时制 1：PM
21:20	HRT[1:0]	小时十位值，以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值，以 BCD 码形式存储
15	保留	必须保持复位值。
14:12	MNT[2:0]	分钟十位值，以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值，以 BCD 码形式存储
7	保留	必须保持复位值。
6:4	SCT[2:0]	秒钟十位值，以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值，以 BCD 码形式存储

16.4.13. 时间戳日期寄存器（RTC_DTS）

偏移地址：0x34
备份域复位值：0x0000 0000
系统复位：无影响
当TSF被置1，该位用来记录日历日期。
清除TSF位也会清除此寄存器。

该寄存器只能按字(32位)访问。



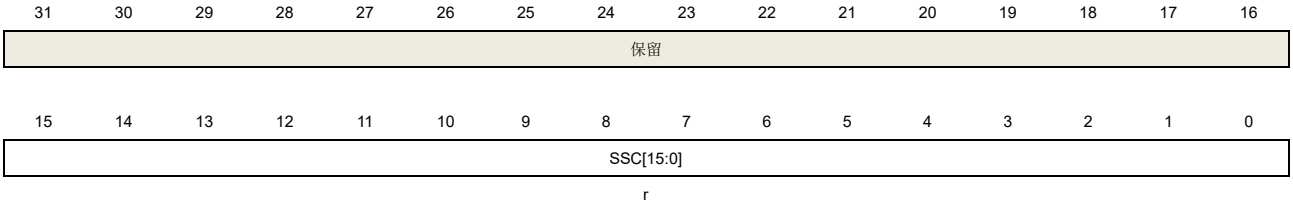
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:13	DOW[2:0]	星期数

12	MONT	月份十位值，以 BCD 码形式存储
11:8	MONU[3:0]	月份个位值，以 BCD 码形式存储
7:6	保留	必须保持复位值。
5:4	DAYT[1:0]	日期十位值，以 BCD 码形式存储
3:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储

16.4.14. 时间戳亚秒寄存器（RTC_SSTS）

偏移地址：0x38
备份域复位：0x0000 0000
系统复位：无影响
当TSF被置1，该位用来记录日历时间。
清除TSF位也会清除此寄存器。

该寄存器只能按字(32 位)访问。

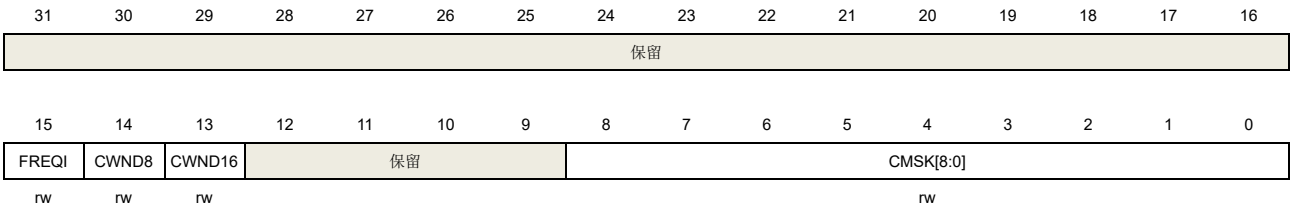


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SSC[15:0]	亚秒值 TSF 置 1 时记录当时的同步预分频计数器的值。

16.4.15. 高精度频率补偿寄存器（RTC_HRFC）

偏移地址：0x3C
备份域复位：0x0000 0000
系统复位：无影响
写保护寄存器。

该寄存器只能按字(32位)访问。



位/位域	名称	描述

31:16	保留	必须保持复位值。
15	FREQI	RTC 频率增加 488.5ppm 0：无影响 1：每 2^{11} 个脉冲增加一个 RTCCLK 脉冲 该位需与 CMSK 位一起使用。如果输入时钟频率是 32.768KHz，在 32s 校准窗期间，增加的 RTCCLK 脉冲数是(512 * FREQI) - CMSK
14	CWND8	采用 8 秒校准周期 0：无影响 1：采用 8 秒校准周期 注意：当 CWND8=1，CMSK[1：0]被锁定在“00”。
13	CWND16	采用 16 秒校准周期 0：无影响 1：采用 16 秒校准周期 注意：当 CWND16=1，CMSK[0] 被锁定在“0”.
12:9	保留	必须保持复位值。
8:0	CMSK[8:0]	校准周期 RTCCLK 脉冲屏蔽数 在 2^{20} 个 RTCCLK 脉冲之内屏蔽的脉冲数 此项功能可以以 0.9537 ppm 的分辨率来降低日历频率

16.4.16. 侵入寄存器（RTC_TAMP）

偏移地址：0x40
备份域复位：0x0000 0000
系统复位：无影响

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		TP2IE	TP1IE	TP0IE	保留	TP2MASK	TP1MASK	TP0MASK	保留	TP2NOERASE	TP1NOERASE	TP0NOERASE	ALRMOUTTYPE	保留	
		rw	rw	rw		rw	rw	rw		rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPU	PRCH[1:0]		FLT[1:0]		FREQ[2:0]			TPTS	TP2EG	TP2EN	TP1EG	TP1EN	TPIE	TP0EG	TP0EN
rw	rw		rw		rw			rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	TP2IE	侵入检测 2 中断使能 0：禁用侵入检测 2 中断 1：启用侵入检测 2 中断
28	TP1IE	侵入检测 1 中断使能 0：禁用侵入检测 1 中断

		1：启用侵入检测 1 中断
27	TP0IE	侵入检测 0 中断使能 0：禁用侵入检测 0 中断 1：启用侵入检测 0 中断
26	保留	必须保持复位值。
25	TP2MASK	侵入检测 2 屏蔽位 0:Tamper 2 引脚产生入侵事件，并且需要软件清除 TP2F 标志才能允许下一次入侵事件检测 1: Tamper 2 引脚产生入侵事件，TP2F 位不会置位，由内部硬件清零，备份域寄存器不会被擦除 注意：当 TP2MASK 为 1 时，Tamper2 中断不能使能
24	TP1MASK	侵入检测 1 屏蔽位 0:Tamper 1 引脚产生入侵事件，并且需要软件清除 TP1F 标志才能允许下一次入侵事件检测 1: Tamper 1 引脚产生入侵事件，TP1F 位不会置位，由内部硬件清零，备份域寄存器不会被擦除 注意：当 TP1MASK 为 1 时，Tamper1 中断不能使能
23	TP0MASK	侵入检测 0 屏蔽位 0:Tamper 0 引脚产生入侵事件，并且需要软件清除 TP0F 标志才能允许下一次入侵事件检测 1: Tamper 0 引脚产生入侵事件，TP0F 位不会置位，由内部硬件清零，备份域寄存器不会被擦除 注意：当 TP0MASK 为 1 时，Tamper0 中断不能使能
22	保留	必须保持复位值。
21	TP2NOERASE	Tamper 2 不擦除备份域寄存器 0：Tamper 2 事件将擦除备份域寄存器 1：Tamper 2 事件不擦除备份域寄存器
20	TP1NOERASE	Tamper 1 不擦除备份域寄存器 0：Tamper 1 事件将擦除备份域寄存器 1：Tamper 1 事件不擦除备份域寄存器
19	TP0NOERASE	Tamper 0 不擦除备份域寄存器 0：Tamper 0 事件将擦除备份域寄存器 1：Tamper 0 事件不擦除备份域寄存器
18	ALRMOUTTYPE	RTC_ALARM 输出类型 0：开漏输出 1：推挽输出
17:16	保留	必须保持复位值。
15	DISPU	RTC_TAMPx 上拉禁用位 0：使能内部 RTC_TAMPx 引脚上的上拉电阻并在采样前进行预充电

		1：禁用预充电功能
14:13	PRCH[1:0]	RTC_TAMPx 的预充电时间 该位设置决定了每次采样前的预充电时间 0x0：1 个 RTC 时钟 0x1：2 个 RTC 时钟 0x2：4 个 RTC 时钟 0x3：8 个 RTC 时钟
12:11	FLT[1:0]	RTC_TAMPx 过滤器计数设置 该位决定了侵入事件检测模式和在电平检测模式下连续采样的次数。 0x0：用边沿模式检测侵入事件，预充电功能被自动禁用。 0x1：用电平模式检测侵入事件。连续采样到 2 个有效电平时认为发生侵入事件 0x2：用电平模式检测侵入事件。连续采样到 4 个有效电平时认为发生侵入事件 0x3：用电平模式检测侵入事件。连续采样到 8 个有效电平时认为发生侵入事件
10:8	FREQ[2:0]	侵入事件电平模式检测的采样频率 0x0：每次采样间隔 32768 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 1Hz) 0x1：每次采样间隔 16384 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 2Hz) 0x2：每次采样间隔 8192 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 4Hz) 0x3：每次采样间隔 4096 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 8Hz) 0x4：每次采样间隔 2048 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 16Hz) 0x5：每次采样间隔 1024 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 32Hz) 0x6：每次采样间隔 512 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 64Hz) 0x7：每次采样间隔 256 个 RTCCLK(若 RTCCLK=32.768KHz，频率为 128Hz)
7	TPTS	侵入事件时触发时间戳 0：无影响 1：当检测到侵入事件时，即使 TSEN=0，TSF 也会被置位
6	TP2EG	TAMP2 输入管脚的侵入事件检测触发沿 如果侵入检测处于边沿模式(FLT =0)： 0：上升沿触发一个侵入检测事件 1：下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLT !=0)： 0：低电平触发一个侵入检测事件 1：高电平触发一个侵入检测事件
5	TP2EN	Tamper2 检测使能位 0：禁用 Tamper2 检测功能 1：启用 Tamper2 检测功能
4	TP1EG	TAMP1 输入管脚的侵入事件检测触发沿 如果侵入检测处于边沿模式(FLT =0)： 0：上升沿触发一个侵入检测事件 1：下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLT !=0)： 0：低电平触发一个侵入检测事件

1：高电平触发一个侵入检测事件

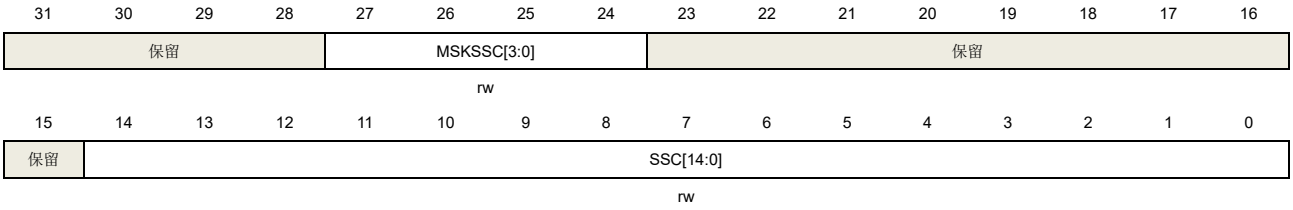
3	TP1EN	Tamper1 检测使能位 0：禁用 Tamper1 检测功能 1：启用 Tamper1 检测功能
2	TPIE	侵入检测中断使能 0：禁用侵入中断 1：启用侵入中断
1	TP0EG	TAMP0 输入管脚的侵入事件检测触发沿 如果侵入检测处于边沿模式(FLT =0)： 0：上升沿触发一个侵入检测事件 1：下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLT !=0)： 0：低电平触发一个侵入检测事件 1：高电平触发一个侵入检测事件
0	TP0EN	Tamper0 检测使能位 0：禁用 Tamper0 检测功能 1：启用 Tamper0 检测功能

注意：强烈建议在改变侵入检测配置之前，应该复位 TPxEN 位。

16.4.17. 闹钟 0 亚秒寄存器（RTC_ALRM0SS）

偏移地址： 0x44
备份域复位：0x0000 0000
系统复位：无影响
写保护寄存器，仅当ALRM0EN=0或INITM=1，可以进行写操作。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	MSKSSC[3:0]	亚秒位域的屏蔽控制位 0x0：屏蔽闹钟亚秒设置。当所有其他的闹钟位域匹配的时候，闹钟将会在每一秒钟到达的时刻置 1。 0x1：SSC[0]位用于时间匹配，其他位被忽略。 0x2：SSC[1：0]位用于时间匹配，其他位被忽略。 0x3：SSC[2：0]位用于时间匹配，其他位被忽略。

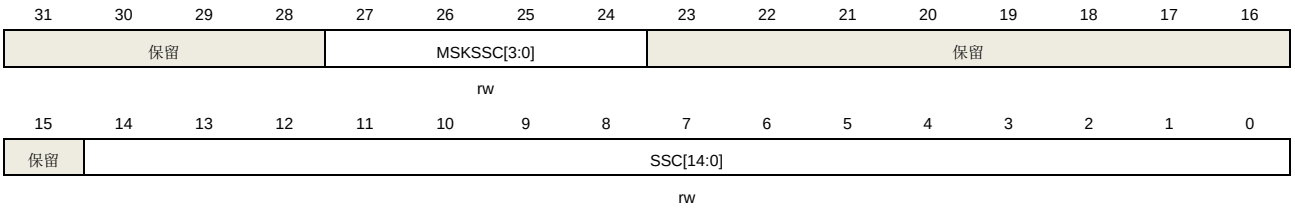
0x4：SSC[3：0]位用于时间匹配，其他位被忽略。
0x5：SSC[4：0]位用于时间匹配，其他位被忽略。
0x6：SSC[5：0]位用于时间匹配，其他位被忽略。
0x7：SSC[6：0]位用于时间匹配，其他位被忽略。
0x8：SSC[7：0]位用于时间匹配，其他位被忽略。
0x9：SSC[8：0]位用于时间匹配，其他位被忽略。
0xA：SSC[9：0]位用于时间匹配，其他位被忽略。
0xB：SSC[10：0]位用于时间匹配，其他位被忽略。
0xC：SSC[11：0]位用于时间匹配，其他位被忽略。
0xD：SSC[12：0]位用于时间匹配，其他位被忽略。
0xE：SSC[13：0]位用于时间匹配，其他位被忽略。
0xF：SSC[14：0]位用于时间匹配，其他位被忽略。
注意：同步预分频计数器的第 15 位(RTC_SS 寄存器中的 SSC[15])从不被匹配。

23:15	保留	必须保持复位值。
14:0	SSC[14:0]	闹钟亚秒值 该值为闹钟亚秒值，用于与同步预分频计数器匹配。 匹配位数由 MSKSSC 位控制。

16.4.18. 闹钟 1 亚秒寄存器（RTC_ALRM1SS）

偏移地址：0x48
备份域复位：0x0000 0000
系统复位：无影响
写保护寄存器，仅当 ALRM1EN=0 或 INITM=1，可以进行写操作。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	MSKSSC[3:0]	亚秒位域的屏蔽控制位 0x0：屏蔽闹钟亚秒设置。当所有其他的闹钟位域匹配的时候，闹钟将会在每一秒钟到达的时刻置 1。 0x1：SSC[0]位用于时间匹配，其他位被忽略。 0x2：SSC[1：0] 位用于时间匹配，其他位被忽略。 0x3：SSC[2：0] 位用于时间匹配，其他位被忽略。 0x4：SSC[3：0] 位用于时间匹配，其他位被忽略。 0x5：SSC[4：0] 位用于时间匹配，其他位被忽略。

0x6：SSC[5：0] 位用于时间匹配，其他位被忽略。
0x7：SSC[6：0] 位用于时间匹配，其他位被忽略。
0x8：SSC[7：0] 位用于时间匹配，其他位被忽略。
0x9：SSC[8：0] 位用于时间匹配，其他位被忽略。
0xA：SSC[9：0] 位用于时间匹配，其他位被忽略。
0xB：SSC[10：0] 位用于时间匹配，其他位被忽略。
0xC：SSC[11：0] 位用于时间匹配，其他位被忽略。
0xD：SSC[12：0] 位用于时间匹配，其他位被忽略。
0xE：SSC[13：0] 位用于时间匹配，其他位被忽略。
0xF：SSC[14：0] 位用于时间匹配，其他位被忽略。
注意：同步预分频计数器的第 15 位(RTC_SS 寄存器中的 SSC[15])从不被匹配。

23:15	保留	必须保持复位值。
14:0	SSC[14:0]	闹钟亚秒值 该值为闹钟亚秒值，用于与同步预分频计数器匹配。 匹配位数由 MSKSSC 位控制。

16.4.19. 备份寄存器（RTC_BKPx）（x=0..4）

偏移地址： 0x50 到 0x64
备份域复位：0x0000 0000
系统复位：无影响

该寄存器只能按字(32位)访问。

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

位/位域	名称	描述
31:0	DATA[31:0]	数据 软件可读写寄存器。由于此寄存器可由 V_{BAT} 供电，因此寄存器值在省电模式下依然保持有效。当侵入检测标志位 TPxF 置 1，这些寄存器会被复位。当 FMC 读保护功能禁用时，这些寄存器会被复位。

17. 定时器（TIMER）

表 17-1. 定时器（TIMERx）分为六种类型

定时器	定时器 1/2	定时器 8/11	定时器 5/6
类型	通用 L0	通用 L1	基本
预分频器	16 位	16 位	16 位
计数器	16 位	16 位	16 位
计数模式	向上，向下，中央对齐	只有向上	只有向上
可重复性	×	×	×
捕获/比较通道数	4	2	0
互补和死区时间	×	×	×
中止输入	×	×	×
单脉冲	●	●	●
正交译码器	●	×	×
主-从管理	●	●	×
内部连接	●(1)	●(2)	TRGO TO DAC(3)
DMA	●	×	●
Debug 模式	●	●	●

(1)

TIMER1	**ITI0:** TIMER2_TRGO	**ITI1:** 1'b0	**ITI2:** 1'b0	**ITI3:** 1'b0
TIMER2	**ITI0:** TIMER1_TRGO	**ITI1:** 1'b0	**ITI2:** 1'b0	**ITI3:** 1'b0

(2)

TIMER8	**ITI0:** TIMER1_TRGO	**ITI1:** TIMER2_TRGO	**ITI2:** 1'b0	**ITI3:** 1'b0
TIMER11	**ITI0:** 1'b0	**ITI1:** TIMER1_TRGO	**ITI2:** TIMER2_TRGO	**ITI3:** 1'b0

(3) 只有更新事件可以产生 DMA 请求。但是定时器 5 和定时 6 中没有 DMA 配置寄存器。

17.1. 通用定时器 L0（TIMERx，x=1，2）

17.1.1. 简介

通用定时器 L0（TIMER1，2）是 4 通道定时器，支持输入捕获，输出比较，产生 PWM 信号控制电机和电源管理。通用定时器 L0 计数器是 16 位无符号计数器。

通用定时器 L0 是可编程的，可以被用来计数，其外部事件可以驱动其他定时器。

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

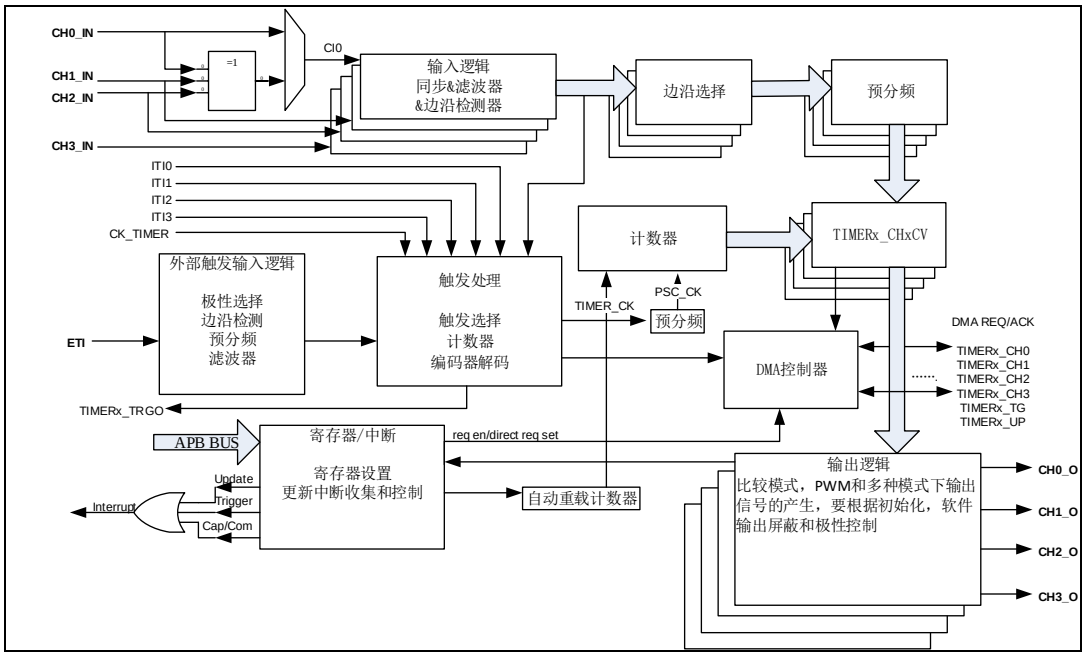
17.1.2. 主要特征

- 总通道数：4；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交译码器接口：被用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口：用来做三相电机控制；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 自动重装载功能；
- 中断输出和DMA请求：更新事件，触发事件，比较/捕获事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

17.1.3. 结构框图

*图 17-1. 通用定时器 L0 结构框图*提供了通用定时器 L0 的内部细节

图 17-1. 通用定时器 L0 结构框图



17.1.4. 功能说明

时钟源配置

通用定时器 L0 可以是内部时钟源 CK_TIMER，或者是由 SMC（TIMERx_SMCFG 寄存器位[2:0]）位确定的时钟源。

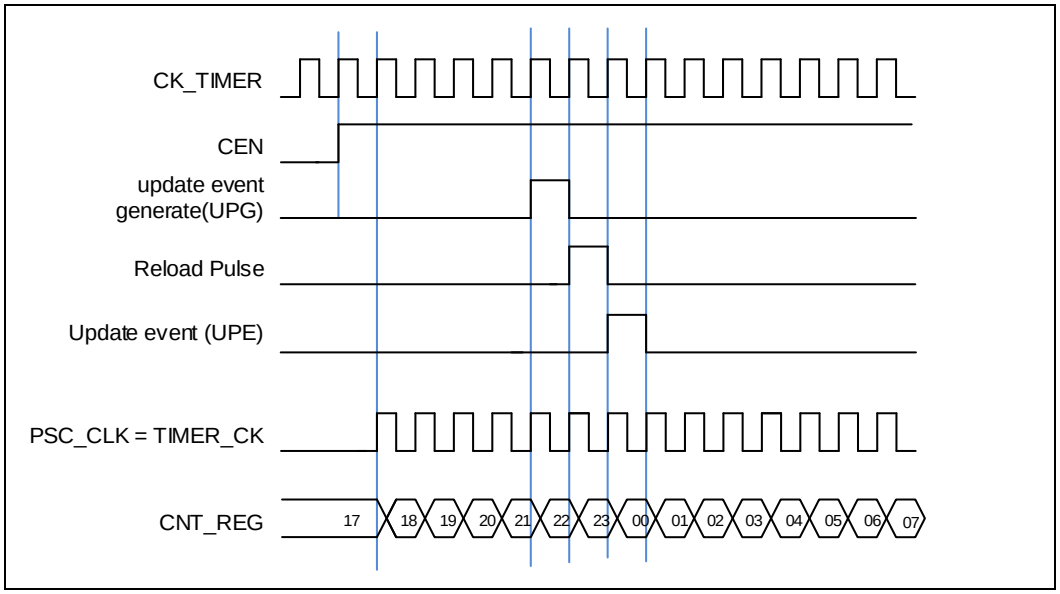
- SMC[2:0]=3’b000，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

如果 SMC[2:0]=3’b000，默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当 CEN 置位，CK_TIMER 经过预分频器（预分频值由 TIMERx_PSC 寄存器确定）产生 PSC_CLK。

这种模式下，驱动预分频器计数的 TIMER_CK 等于来自于 RCU 模块的 CK_TIMER。

如果将 TIMERx_SMCFG 寄存器的 SMC[2:0]设置为 0x1、0x2、0x3 和 0x7，预分频器被其他时钟源(由 TIMERx_SMCFG 寄存器的 TRGS[2:0]区域选择)驱动，在下文说明。当 SMC 位被设置为 0x4、0x5 和 0x6，计数器预分频器时钟源由内部时钟 CK_TIMER 驱动。

图 17-2. 内部时钟分频为 1 时，计数器的时序图



- SMC[2:0]=3'b111（外部时钟模式0），定时器选择外部输入引脚作为时钟源。

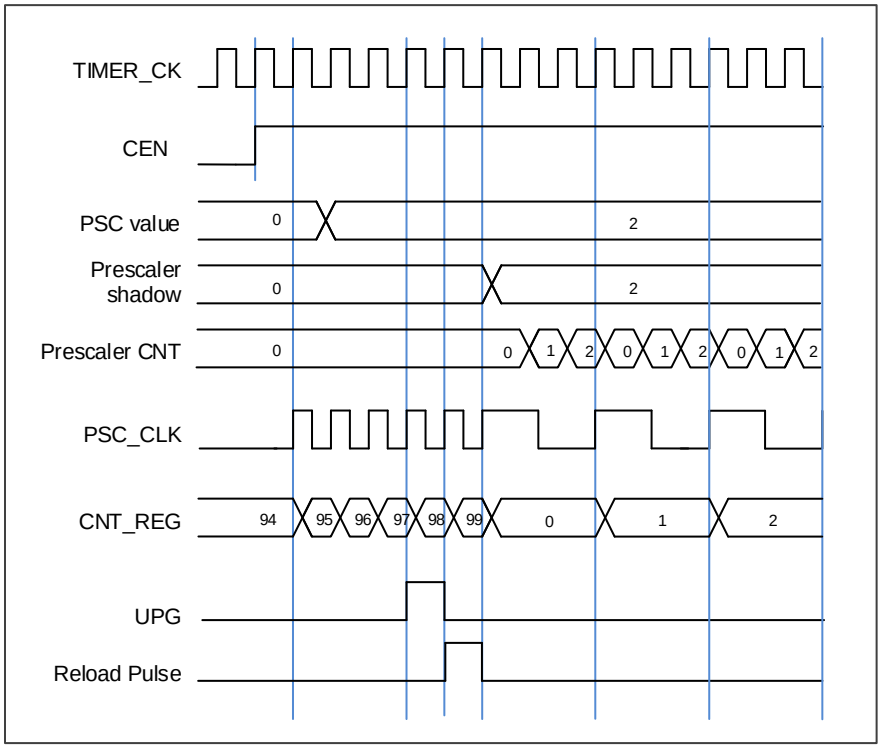
计数器预分频器可以在 TIMERx_CH0/ TIMERx_CH1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 SMC[2:0]为 0x7 同时设置 TRGS[2:0]为 0x4，0x5 或 0x6 来选择。

计数器预分频器也可以在内部触发信号 ITI0/1/2/3 的上升沿计数。这种模式可以通过设置 SMC[2:0]为 0x7 同时设置 TRGS[2:0]为 0x0，0x1，0x2 或者 0x3。

时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 17-3. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值（定义在 TIMERx_CAR 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图 17-4. 向上计数时序图,PSC=0/2*和*图 17-5. 向上计数时序图,在运行时改变 TIMERx_CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 17-4. 向上计数时序图，PSC=0/2

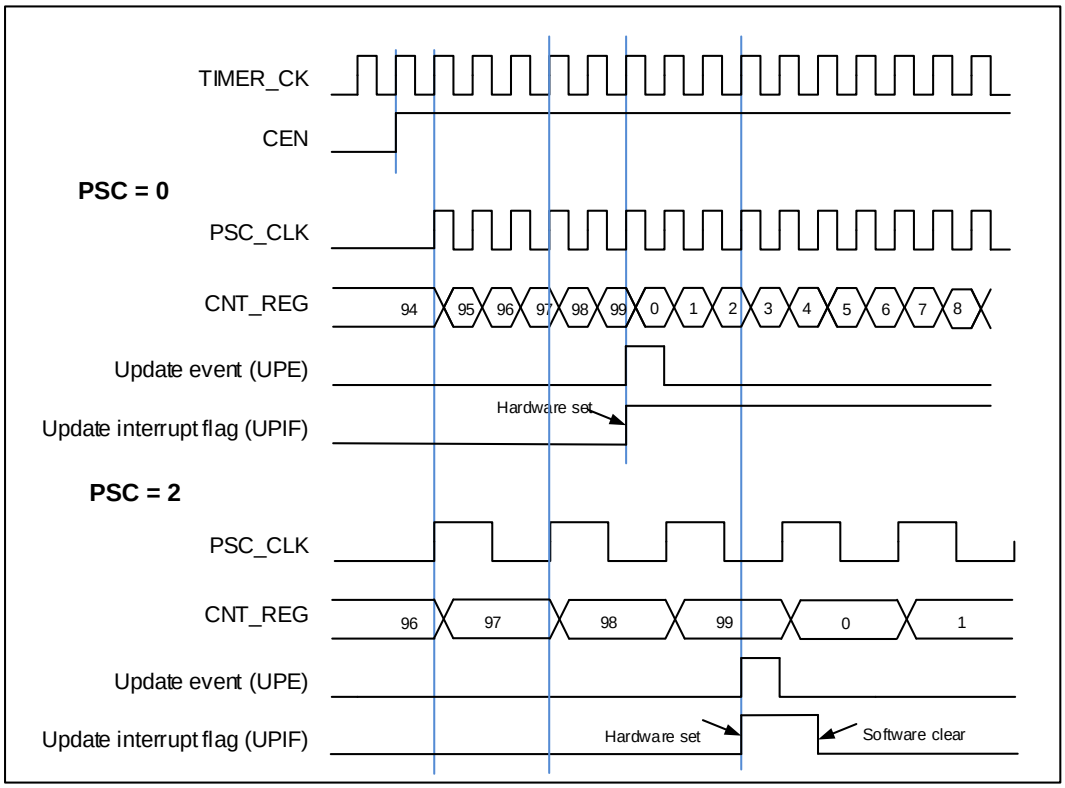
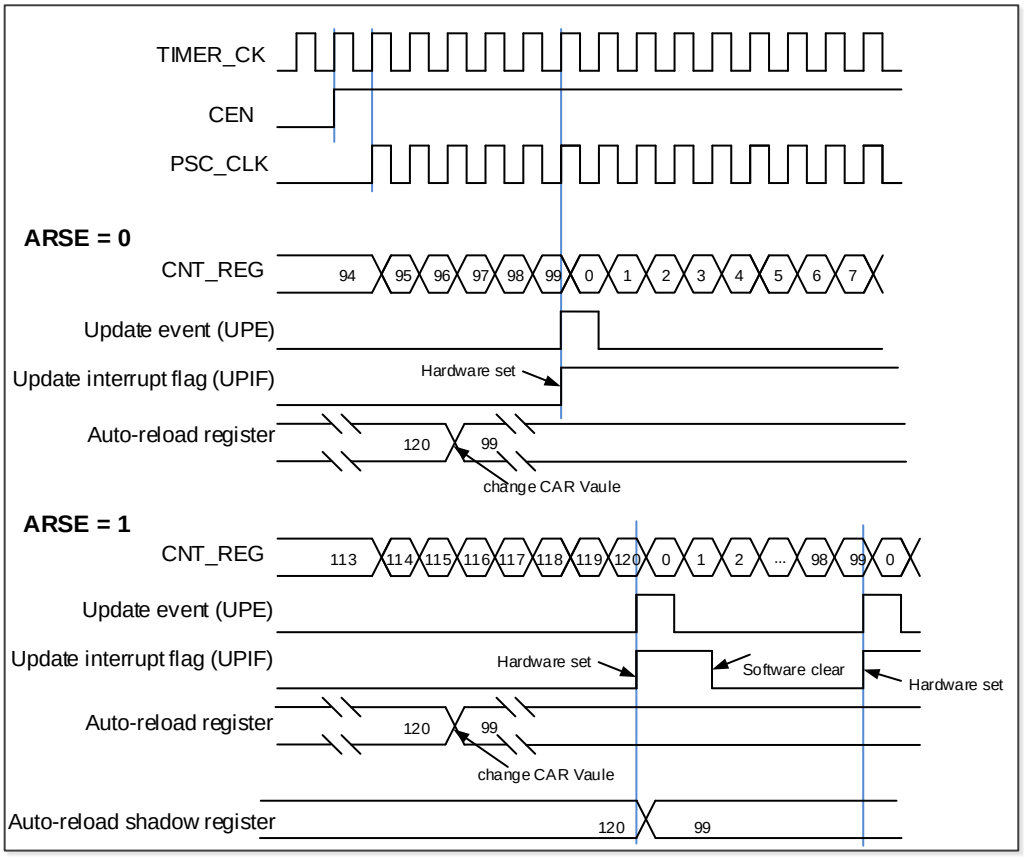


图 17-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



计数器向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在 TIMERx_CAR 寄存器中）向下连续计数到 0。一旦计数器计数到 0，计数器会重新从自动加载值开始计数并产生下溢。在向下计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 1。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被初始化为自动加载值，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图17-6. 向下计数时序图，PSC=0/2*和*图17-7. 向下计数时序图，在运行时改变TIMERx_CAR寄存器值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同时钟频率下的行为。

图 17-6. 向下计数时序图，PSC=0/2

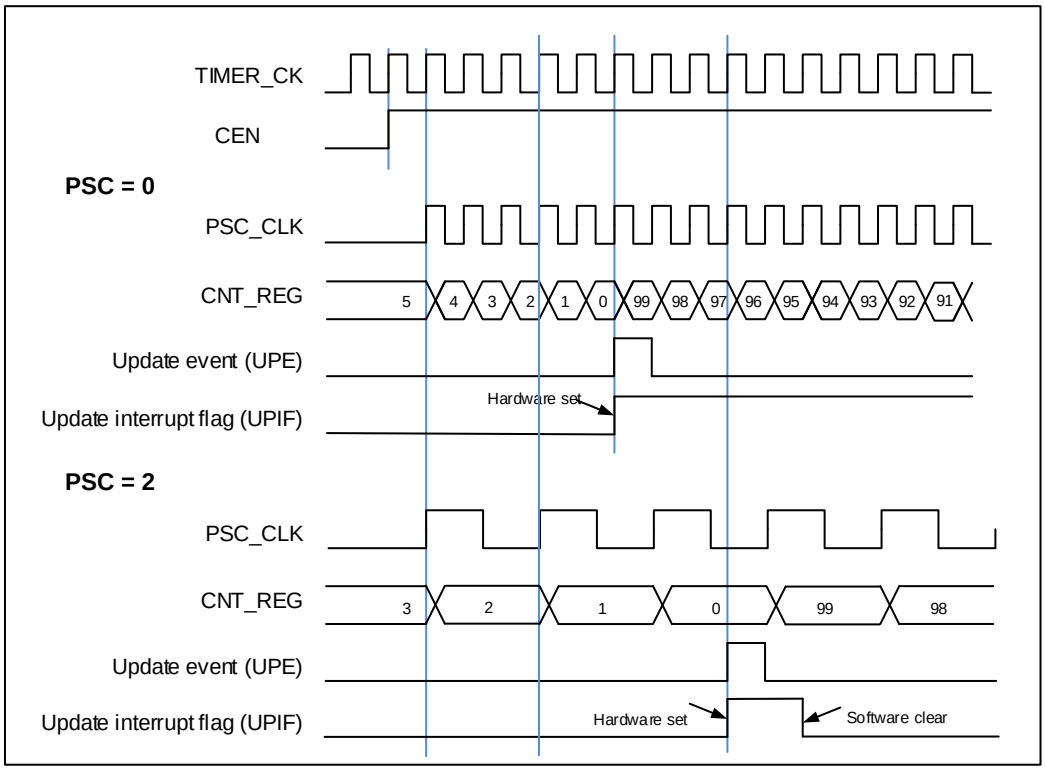
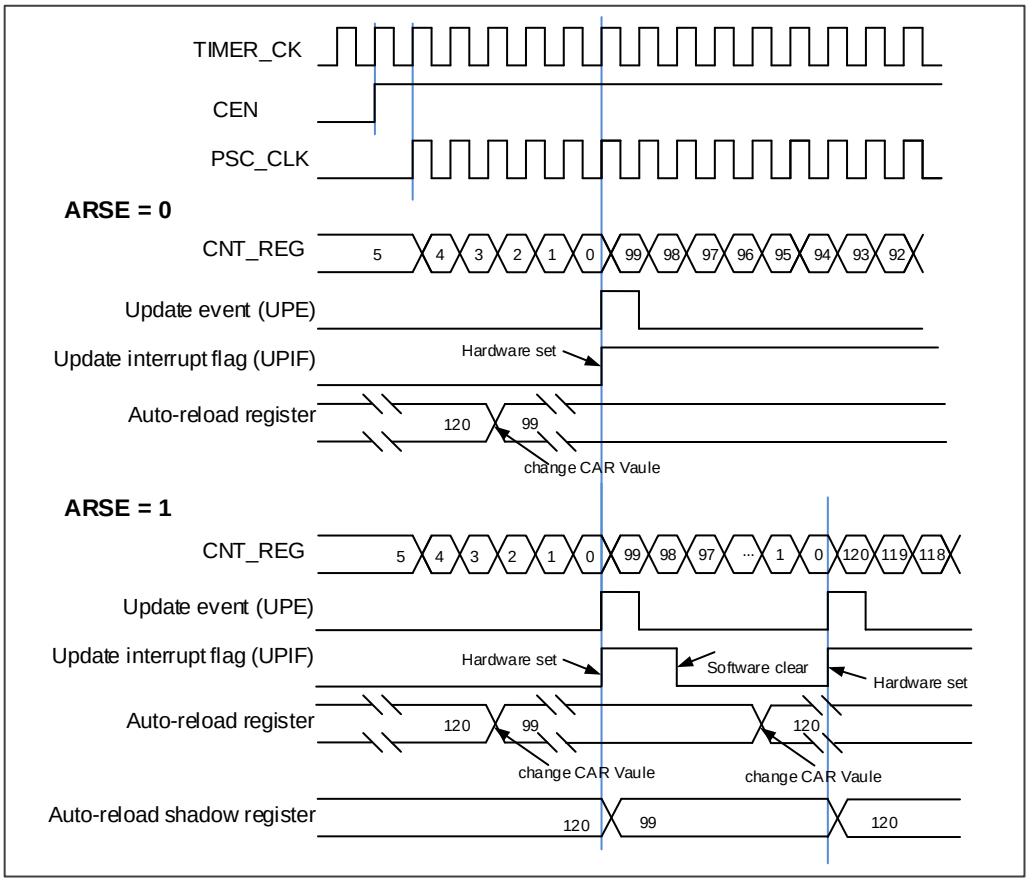


图 17-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



计数器中央对齐模式

在中央对齐模式下，计数器交替的从 0 开始向上计数到自动加载值，然后再向下计数到 0。向上计数模式中，定时器模块在计数器计数到（自动加载值-1）产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 只读，表明了的计数方向。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

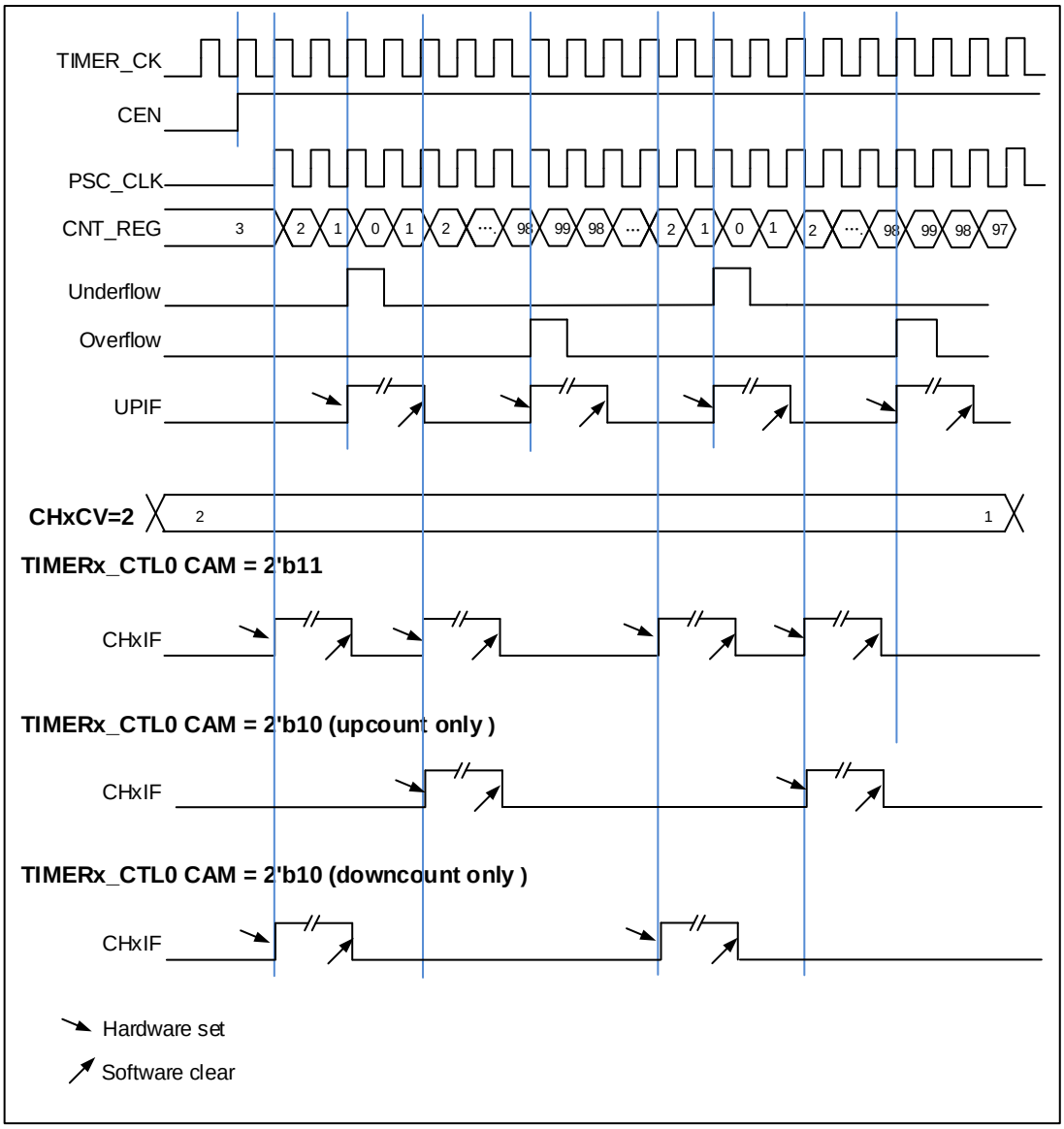
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1。但是 CHxIF 位是否置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图17-8. 中央计数模式计数器时序图*。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

*图 17-8. 中央计数模式计数器时序图*给出了一些例子，当 TIMERx_CAR=0x99，TIMERx_PSC=0x0 时，计数器的行为。

图 17-8. 中央计数模式计数器时序图



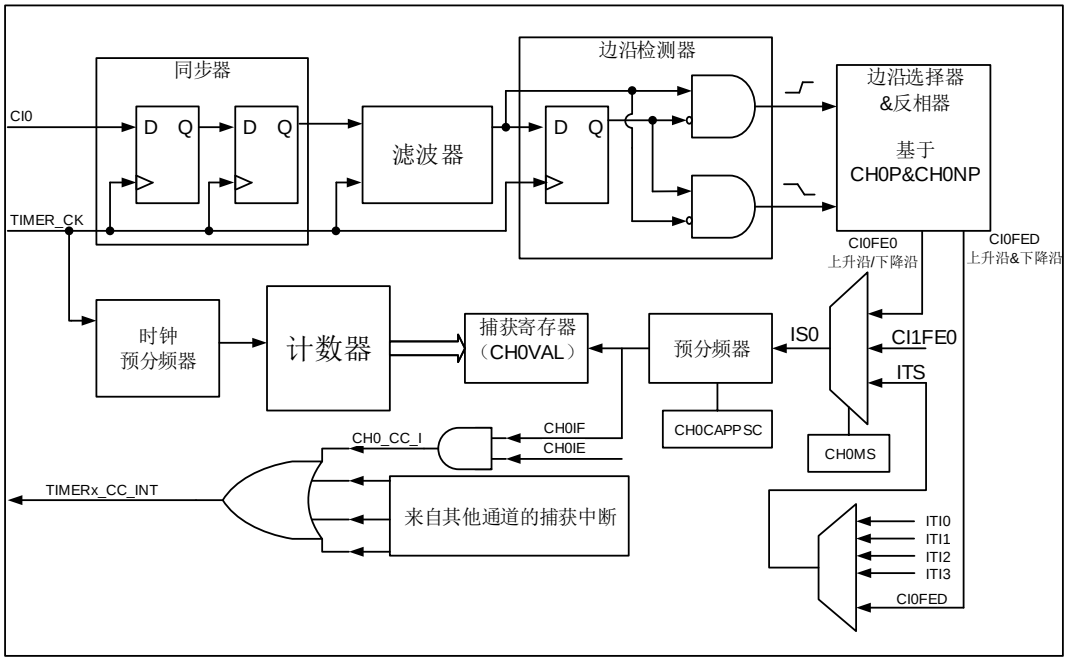
输入捕获和输出比较通道

通用定时器 L0 拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

图 17-9. 通道输入捕获原理



通道输入信号 CIx 有两种选择，一种是 TIMERx_CHx 信号，另一种是 TIMERx_CH0，TIMERx_CH1 和 TIMERx_CH2 异或之后的信号。通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，CHxVAL 存储计数器的值。

配置步骤如下：

第一步：滤波器配置（TIMERx_CHCTL0寄存器中CHxCAPFLT）：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。

第二步：边沿选择（TIMERx_CHCTL2寄存器中CHxP）：
配置CHxP选择上升沿或者下降沿。

第三步：捕获源选择（TIMERx_CHCTL0寄存器中CHxMS）：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式（CHxMS!=0x0），而且TIMERx_CHxCV寄存器不能再被写。

第四步：中断使能（TIMERx_DMAINTEN寄存器中CHxIE 和 CHxDEN）：
使能相应中断，可以获得中断和DMA请求。

第五步：捕获使能（TIMERx_CHCTL2寄存器中CHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN的配置，相应的中断和DMA请求会被提出。

直接产生：软件设置CHxG位，会直接产生中断和DMA请求。

输入捕获模式也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 2'b10，选择通道 1 捕获信号为 CI0 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

通道输出比较功能

图 17-10. 通道输出比较原理（x=0，1，2，3）

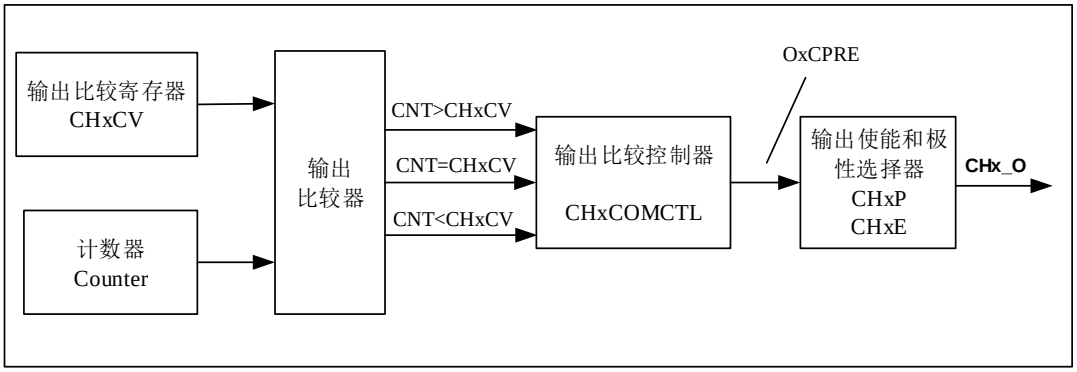


*图 17-10. 通道输出比较原理(x=0，1，2，3)*给出了输出比较的原理电路。通道输出信号 CHx_O 与 OxCPRE 信号（详情请见*通道输出准备信号*）的关系描述如下：OxCPRE 信号高电平有效，CHx_O 的输出情况与 OxCPRE 信号，CHxP 位和 CHxE 位有关（具体情况请见 TIMERx_CHCTL2 寄存器中的描述）。例如，当设置 CHxP=0（CHx_O 高电平有效，与 OxCPRE 输出极性相同）、CHxE=1（CHx_O 输出使能）时：

若OxCPRE输出有效（高）电平，则CHx_O输出有效（高）电平；

若OxCPRE输出无效（低）电平，则CHx_O输出无效（低）电平。

在输出比较模式，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 CHxCV 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 CHxCV 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CxCDE=1 则会产生 DMA 请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CxCDE位配置中断/DMA请求使能。

第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

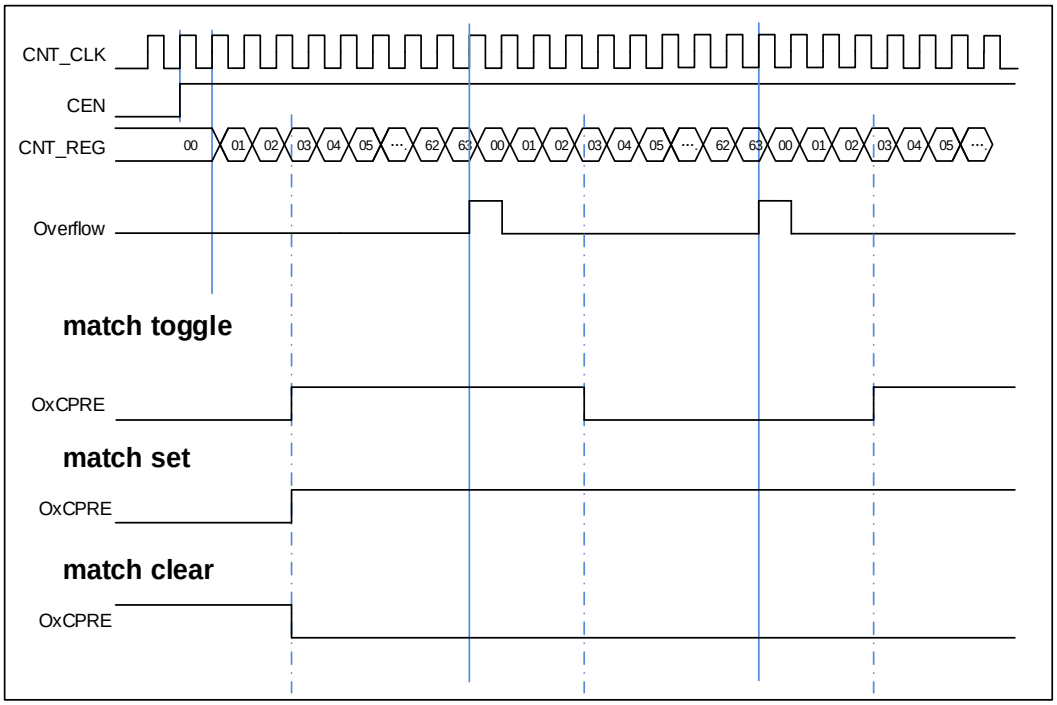
CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图 17-11. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，

CHxVAL=0x3。

图 17-11. 三种输出比较模式



输出 PWM 功能

在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL 为 3'b110，PWM 模式 1 是配置 CHxCOMCTL 为 3'b111），通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

根据计数模式，我们可以分为两种 PWM 波：EAPWM(边沿对齐 PWM)和 CAPWM(中央对齐 PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 17-12. EAPWM 时序图*显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由（2*TIMERx_CAR 寄存器值）决定，占空比由（2*TIMERx_CHxCV 寄存器值）决定。*图 17-13. CAPWM 时序图*显示了 CAPWM 的输出波形和中断。

在向上计数模式中， PWM 模式 0 下(CHxCOMCTL=3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为无效电平；PWM 模式 1 下(CHxCOMCTL=3'b111)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

图 17-12. EAPWM 时序图

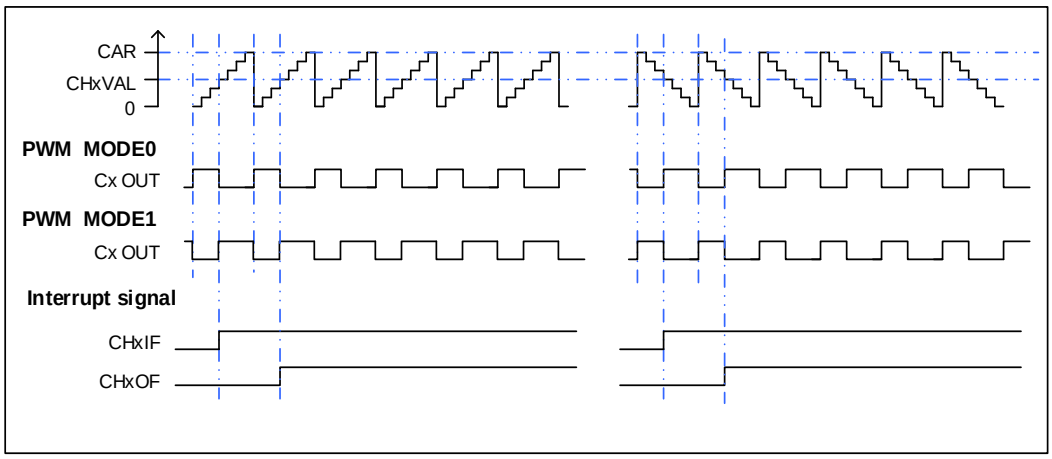
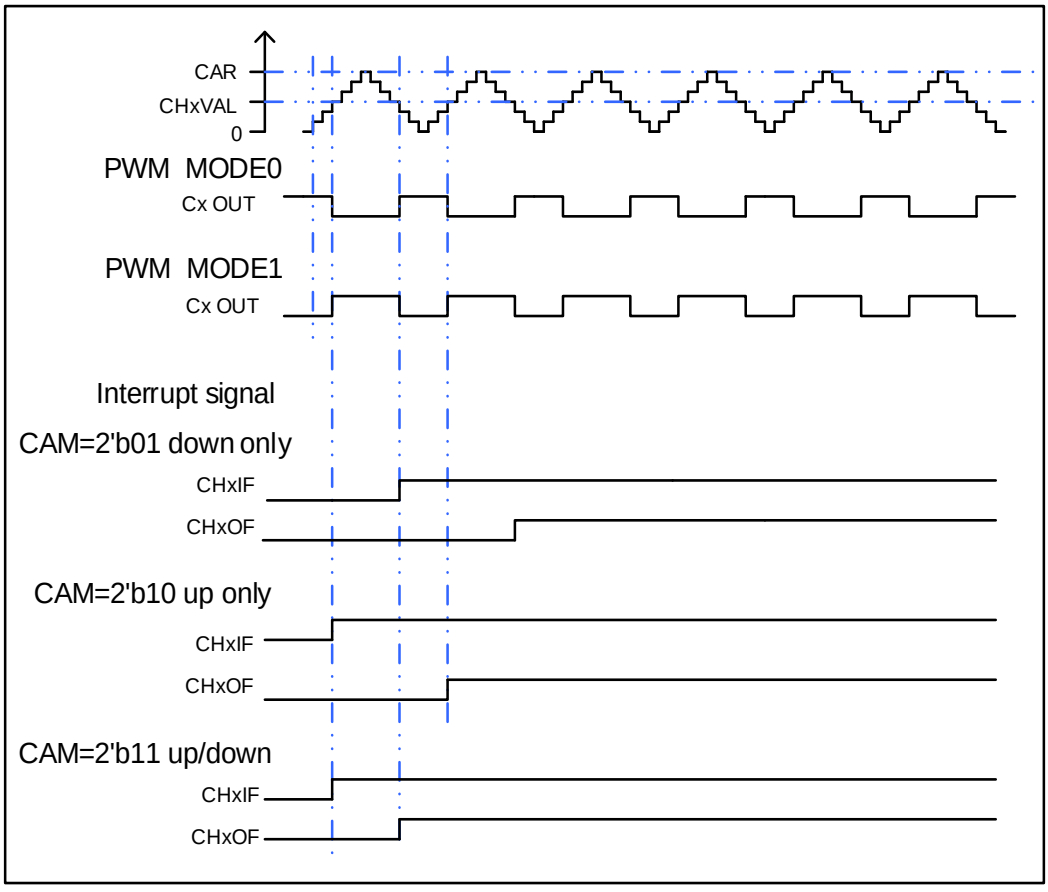


图 17-13. CAPWM 时序图



通道输出准备信号

根据*图 17-10. 通道输出比较原理（x=0，1，2，3）*所示，当 TIMERx 用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号 OxCPRE 信号（通道 x 输出准备信号）。设置 CHxCOMCTL 位可以定义 OxCPRE 信号类型。当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 输出准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为

低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域为 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL=0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间的比较结果。

设置 CHxCOMCEN=1，当由外部 ETI 引脚信号产生的 ETIFP 信号为高电平时，OxCPRE 被强制为低电平。在下一次更新事件到来时，OxCPRE 信号才会回到有效电平状态。

正交译码器

正交译码器功能使用由TIMERx_CH0和TIMERx_CH1引脚生成的CI0FE0和CI1FE1正交信号各自相互作用产生计数值。在每个输入源改变期间，DIR位会发生改变。输入源可以是只有CI0FE0，可以只有CI1FE1，或着可以同时有CI0FE0和CI1FE1，通过设置SMC=0x01, 0x02或0x03来选择使用哪种模式。计数器计数方向改变的机制如*表17-2. 不同正交译码器模式下的计数方向*所示。正交译码器可以当作一个带有方向选择的外部时钟，这意味着计数器会在0和自动加载值之间连续的计数。因此，用户必须在计数器开始计数前配置TIMERx_CAR寄存器。

表 17-2. 不同正交译码器模式下的计数方向

计数模式	电平	CI0FE0		CI1FE1	
		上升	下降	上升	下降
正交译码器模式0 SMC[2:0]=3'b001	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
正交译码器模式1 SMC [2:0]=3'b010	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
正交译码器模式2 SMC [2:0]=3'b011	CI1FE1=1	向下	向上	X	X
	CI1FE1=0	向上	向下	X	X
	CI0FE0=1	X	X	向上	向下
	CI0FE0=0	X	X	向下	向上

注意: "-" 意思是"无计数"; "X" 意思是不可能。"0" 意思是低电平, "1" 意思是高电平。

图 17-14. 在正交译码器模式 2 且 CI0FE0 极性不反相时计数器行为

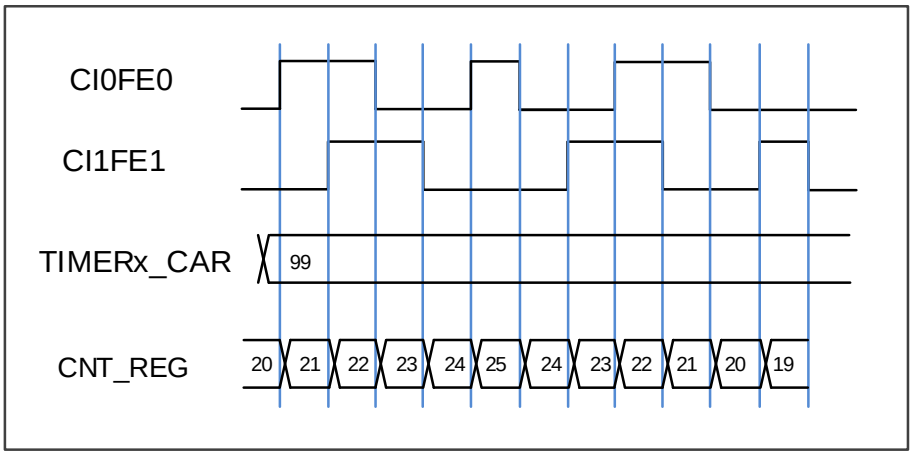
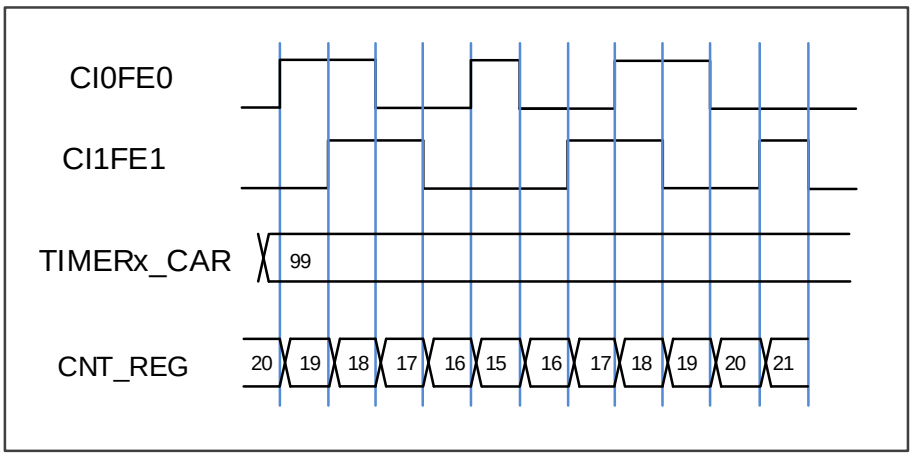


图 17-15. 在正交译码器模式 2 且 CI0FE0 极性反相时计数器行为



霍尔传感器接口功能

通用定时器 L0 支持霍尔传感器接口功能，该功能可以用来控制 BLDC 电机。

三个霍尔传感器与 TIMER_in 定时器的三路输入捕获引脚一一对应连接，每个霍尔传感器输入一路波形到输入引脚，分析三路霍尔信号可以计算出转子的位置和速度。

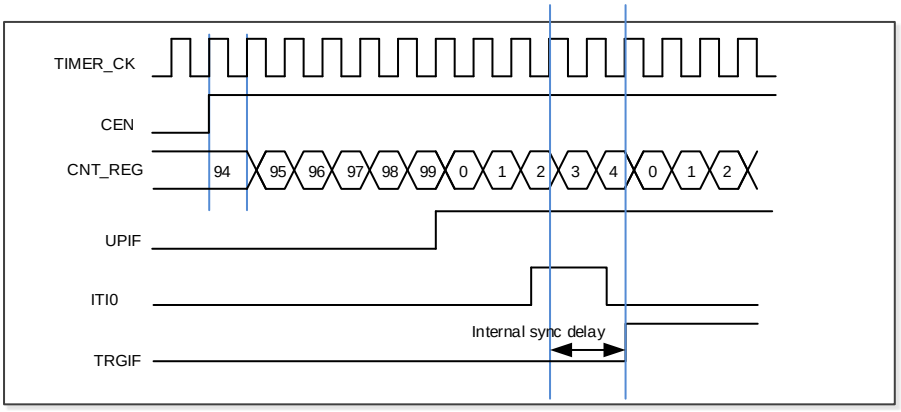
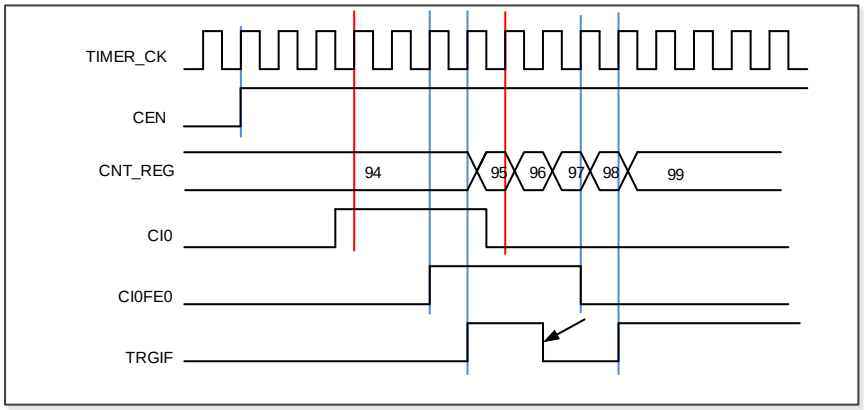
置位 TI0S 可以使能异或功能，则每个输入信号电平发生变化时 CI0 都会翻转。CH0VAL 将会记录 CI0 翻转时的计数器数值。

主-从管理

TIMERx 能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置 TIMERx_SMCFG 寄存器中的 SMC [2:0]配置这些模式。这些模式的输入触发源可以通过设置 TIMERx_SMCFG 寄存器中的 TRGS [2:0]来选择。

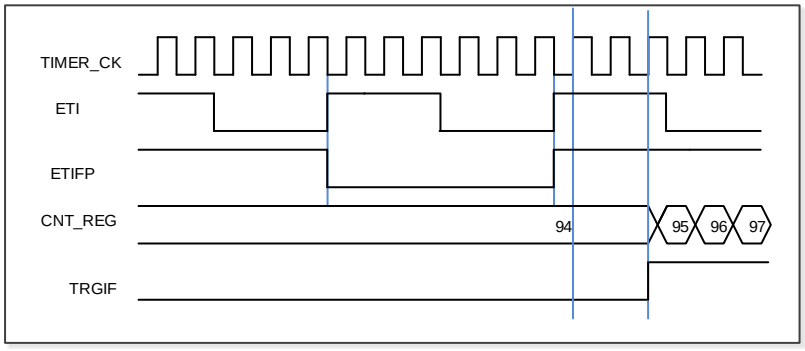
表 17-3. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100 (复位模式)	TRGS[2:0] 000: ITI0 001: ITI1	如果触发源是CI0FE0或者CI1FE1，配置CHxP和	触发源ITIx，滤波和预分频不可用

	模式选择	触发源选择	极性选择	滤波和预分频
	3'b101 (暂停模式) 3'b110 (事件模式)	010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP	CHxNP来选择极性和反相 如果触发源是ETIF，配置ETP选择极性和反相	触发源CIx，配置CHxCAPFLT设置滤波，分频不可用 触发源是ETIF，滤波和预分频不可用
例1	复位模式 当触发输入上升沿，计数器清零重启	TRGIS[2:0]=3'b000 选择ITI0为触发源	触发源是ITI0，极性选择不可用	触发源是ITI0，滤波和预分频不可用
	图 17-16. 复位模式下的控制电路 			
例2	暂停模式 当触发输入为低的时候，计数器暂停计数	TRGIS[2:0]=3'b101 选择CI0FE0为触发源	TI0S=0.（非异或）[CH0NP==0, CH0P==0]不反相.在上升沿捕获	在这个例子中滤波被旁路
	图 17-17. 暂停模式下的控制电路 			
例3	事件模式 触发输入的上升沿计数器开始计数	TRGIS[2:0]=3'b111 选择ETIF为触发源.	ETP = 0 没有极性改变	ETPSC = 1, 2分频. ETFC = 0，无滤波

	模式选择	触发源选择	极性选择	滤波和预分频

图 17-18. 事件模式下的控制电路



单脉冲模式

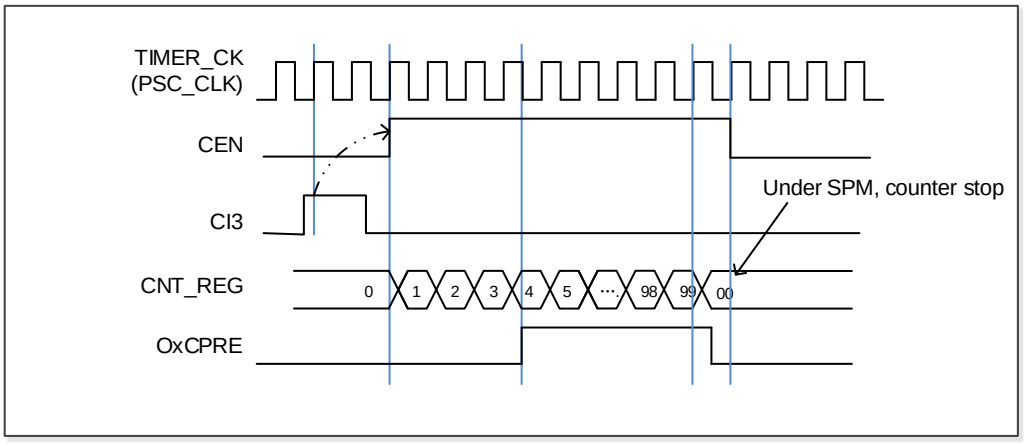
单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置 TIMERx_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。触发信号沿或者软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止工作，计数值被保持。

在单脉冲模式下，有效的外部触发边沿会将 CEN 位置 1，使能计数器。然而，执行计数值和 TIMERx_CHxCV 寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将 TIMERx_CHCTL0/1 寄存器的 CHxCOMFEN 位置 1。单脉冲模式下，触发上升沿产生之后， OxCPRE 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为 PWM1 或 PWM2 输出运行模式下时 CHxCOMFEN 位才可用，触发源来源于触发信号。

图 17-19. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 展示了一个例子

图 17-19. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99

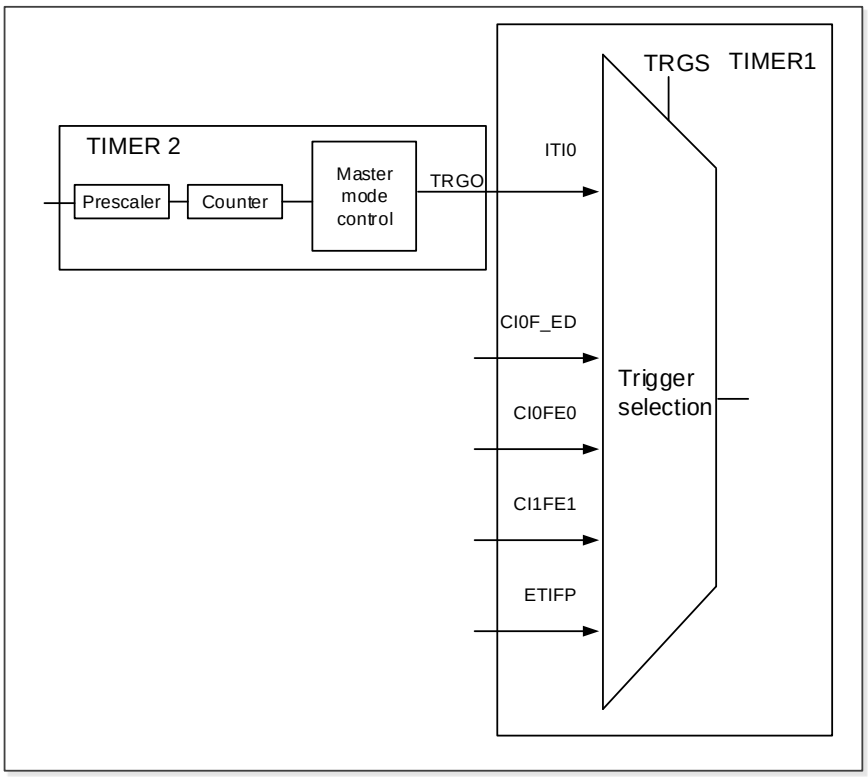


定时器互连

定时器之间的相互连接可以实现定时器的级联或者同步。可以通过配置一个定时器工作在主模式，另一个定时器工作在从模式来实现。下面的几张图显示了一些主从模式触发选择的例子。

图 17-20. 定时器 1 主/从模式的例子 显示了当定时器 1 配置为从模式时的触发选择

图 17-20. 定时器 1 主/从模式的例子



其他互联的例子：

- **定时器 2 作为定时器 1 的预分频器**

参考*图 17-20. 定时器 1 主/从模式的例子*连接配置定时器 2 为定时器 1 的预分频器，步骤如下：

1. 配置定时器2为主模式，选择其更新事件(UPE)为触发输出(配置TIMER2_CTL1寄存器的MMC=3’b010)。定时器2在每次计数器溢出产生更新事件时，输出一个周期信号；

2. 配置定时器2周期(TIMER2_CAR寄存器)；

3. 选择定时器1输入触发源为定时器2 (配置TIMER1_SMCFG寄存器的TRGS=3’b000)；

4. 配置定时器1在外部时钟模式0(配置TIMER1_SMCFG寄存器的SMC=3’b111)；

5. 写1到CEN位启动定时器1 (TIMER1_CTL0寄存器)；

6. 写1到CEN位启动定时器2 (TIMER2_CTL0寄存器)。

- **使用一个外部触发来同步两个定时器。**

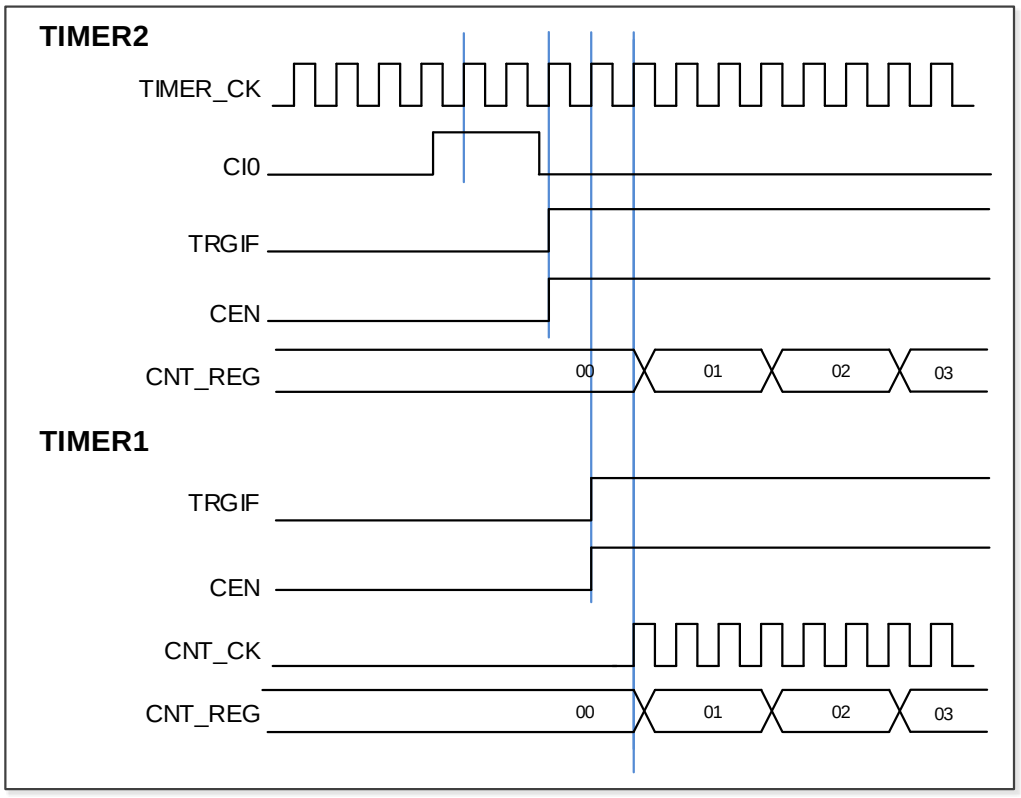
配置定时器2的使能信号触发定时器1的开启，配置定时器2的CI0输入信号上升沿来触发定时

器2。为了确保两个定时器同步开启，定时器2必须配置在主/从模式。步骤如下:

1. 配置定时器2工作在从模式，并选择CI0作为触发输入(配置TIMER2_SMCFG寄存器的TRGS=3'b101)；
2. 配置定时器2工作在事件模式(配置TIMER2_SMCFG寄存器的SMC=3'b110)；
3. 写MSM=1(TIMER2_SMCFG寄存器)来配置定时器2工作在主/从模式；
4. 配置定时器1的触发输入为定时器2 (配置TIMERx_SMCFG 寄存器的TRGS=3'b000)；
5. 配置定时器1工作在事件模式(配置TIMER1_SMCFG寄存器的SMC=3'b110)。

当定时器 2 的 CI0 信号产生上升沿时，两个定时器的计数器在内部时钟下开始同步计数，二者的 TRGIF 标志位都被置 1。

图 17-21. 用定时器 2 的 CI0 信号来触发定时器 1 和定时器 2



定时器 DMA 模式

定时器 DMA 模式是指通过 DMA 模块配置定时器的寄存器。有两个跟定时器 DMA 模式相关的寄存器：TIMERx_DMACFG 和 TIMERx_DMATB。必须使能相应的 DMA 请求位，一些内部中断事件才可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P（传输方向为从内存到外设）模式，PADDR（外设基地址）为 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG

寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4，DMATA+0x8，DMATA+0xC 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送（DMATC+1）次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

定时器调试模式

当 Cortex®-M23 内核停止，DBG_CTL0 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

17.1.5. TIMERx 寄存器（x=1，2）

TIMER1基地址：0x4000 0000
TIMER2基地址：0x4000 0400

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值。
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK\_TIMER}$ 01：$f_{DTS}= f_{CK\_TIMER} /2$ 10：$f_{DTS}= f_{CK\_TIMER} /4$ 11：保留
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器 1：使能 TIMERx_CAR 寄存器的影子寄存器
6:5	CAM[1:0]	计数器对齐模式选择 00：无中央对齐计数模式(边沿对齐模式)。 DIR位指定了计数方向 01：中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向下计数时，CHxF位置1 10：中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向上计数时，CHxF位置1 11：中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），在向上和向下计数时，CHxF位都会置1 当计数器使能以后，该位不能从 0x00 切换到非 0x00。
4	DIR	方向 0：向上计数 1：向下计数

当计数器配置为中央对齐计数模式或正交译码器模式时，该位只读。

3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：更新事件禁能. 注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能 0：计数器禁能 1：计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和正交译码器模式才能工作。

控制寄存器 1（TIMERx_CTL1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TI0S	MMC[2:0]			DMAS	保留.		
								rw	rw			rw			

位/位域	名称	描述
31:8	保留	必须保持复位值。

7	TI0S	通道0触发输入选择 0：选择 TIMERx_CH0 引脚作为通道 0 的触发输入 1：选择 TIMERx_CH0，CH1 和 CH2 引脚异或的结果作为通道 0 的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能 000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定 011：当通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲 100：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O0CPRE 101：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O1CPRE 110：当产生一次比较事件时，输出一个TRGO信号，比较事件源来自O2CPRE 111：当产生一次比较事件时，输出一个 TRGO 信号，比较事件源来自 O3CPRE
3	DMAS	DMA 请求源选择 0：当通道捕获/比较事件发生时，发送通道 x 的 DMA 请求 . 1：当更新事件发生，发送通道 x 的 DMA 请求
2:0	保留	必须保持复位值。

从模式配置寄存器（TIMERx_SMCFG）

地址偏移：0x08

复位值： 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	TRGS[2:0]			保留	SMC[2:0]		
rw	rw	rw		rw				rw	rw				rw		

位/位域	名称	描述
31:16	保留	必须保持复位值
15	ETP	外部触发极性 该位指定 ETI 信号的极性 0：ETI 高电平或上升沿有效 .

1：ETI 低电平或下降沿有效 .

14 SMC1

SMC 的一部分为了使能外部时钟模式 1

在外部时钟模式 1，计数器由 ETIF 信号上的任意有效边沿驱动

0：外部时钟模式 1 禁能

1：外部时钟模式 1 使能

当从模式配置为复位模式，暂停模式和事件模式时，定时器仍然可以工作在外部时钟模式 1。但是 TRGS 必须不能为 3'b111。

如果外部时钟模式 0 和外部时钟模式 1 同时被被配置，外部时钟的输入是 ETIF

注意：外部时钟模式 0 使能在寄存器的 SMC[2:0]位域。

13:12 ETPSC[1:0]

外部触发预分频

外部触发信号 ETIFP 的频率不能超过 TIMER_CK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETIFP 的频率。

00：预分频禁能

01：2 分频

10：4 分频

11：8 分频

11:8 ETFC[3:0]

外部触发滤波控制

外部触发信号可以通过数字滤波器进行滤波，该位域定义了数字滤波器的滤波能力。数字滤波器的基本原理是：以 f_{SAMP} 频率连续采样外部触发信号，同时记录采样相同电平的次数。当该次数达到配置的滤波能力时，则认为是一个有效的电平信号。

EXTFC[3:0]	次数	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{TIMER_CK}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS\_CK}/2$
4'b0101	8	
4'b0110	6	$f_{DTS\_CK}/4$
4'b0111	8	
4'b1000	6	$f_{DTS\_CK}/8$
4'b1001	8	
4'b1010	5	$f_{DTS\_CK}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS\_CK}/32$
4'b1110	6	
4'b1111	8	

7 MSM

主-从模式

该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。

0：主从模式禁能

1：主从模式使能

6:4	TRGS[2:0]	触发选择 该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP 从模式被使能后这些位不能改
3	保留	必须保持复位值
2:0	SMC[2:0]	从模式控制 000：关闭从模式. 如果 CEN=1，则预分频器直接由内部时钟驱动 001：正交译码器模式 0. 根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数 010：正交译码器模式 1. 根据 CI0FE0 的电平，计数器在 CI1FE1 的边沿向上/下计数 011：正交译码器模式 2. 根据另一个信号的输入电平，计数器在 CI0FE0 和 CI1FE1 的 边沿向上/ 下计数 100：复位模式. 选中的触发输入的上升沿重新初始化计数器，并且产生更新事件. 101：暂停模式. 当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器时钟停止 110：事件模式.计数器在触发输入的上升沿启动。 111：外部时钟模式 0. 选中的触发输入的上升沿驱动计数器

DMA 和中断使能寄存器（TIMERx_DMAINTEN）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	保留	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	保留	TRGIE	保留	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值。.
14	TRGDEN	触发 DMA 请求使能

0：禁止触发 DMA 请求
1：使能触发 DMA 请求

13	保留	必须保持复位值。.
12	CH3DEN	通道 3 比较/捕获 DMA 请求使能 0：禁止通道 3 比较/捕获 DMA 请求 1：使能通道 3 比较/捕获 DMA 请求
11	CH2DEN	通道 2 比较/捕获 DMA 请求使能 0：禁止通道 2 比较/捕获 DMA 请求 1：使能通道 2 比较/捕获 DMA 请求
10	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0：禁止通道 1 比较/捕获 DMA 请求 1：使能通道 1 比较/捕获 DMA 请求
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0：禁止通道 0 比较/捕获 DMA 请求 1：使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7	保留	必须保持复位值。.
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5	保留	必须保持复位值。.
4	CH3IE	通道 3 比较/捕获中断使能 0：禁止通道 3 中断 1：使能通道 3 中断
3	CH2IE	通道 2 比较/捕获中断使能 0：禁止通道 2 中断 1：使能通道 2 中断
2	CH1IE	通道 1 比较/捕获中断使能 0：禁止通道 1 中断 1：使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 1：使能通道 0 中断
0	UPIE	更新中断使能 0：禁止更新中断

1：使能更新中断

中断标志寄存器（TIMERx_INTF）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			CH3OF	CH2OF	CH1OF	CH0OF	保留		TRGIF	保留	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
			rc_w0	rc_w0	rc_w0	rc_w0		.	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:13	保留	必须保持复位值。.
12	CH3OF	通道 3 捕获溢出标志 参见 CH0OF 描述
11	CH2OF	通道 2 捕获溢出标志 参见 CH0OF 描述
10	CH1OF	通道 1 捕获溢出标志 参见 CH0OF 描述
9	CH0OF	通道 1 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0. 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8:7	保留	必须保持复位值。.
6	TRGIF	触发中断标志 当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。 0：无触发事件产生 1：触发中断产生
5	保留	必须保持复位值。.
4	CH3IF	通道 3 比较/捕获中断标志 参见 CH0IF 描述
3	CH2IF	通道 2 比较/捕获中断标志 参见 CH0IF 描述

2	CH1IF	通道 1 比较/捕获中断标志 参见 CH0IF 描述
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 0：无通道 0 中断发生 1：通道 0 中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器（TIMERx_SWEVG）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TRGG	保留.	CH3G	CH2G	CH1G	CH0G	UPG
									w		w	w	w	w	w

位/位域	名称	描述
31:7	保留	必须保持复位值。.
6	TRGG	触发事件产生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，TIMERx_INTF 寄存器的 TRGIF 标志位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：无触发事件产生 1：产生触发事件
5	保留	必须保持复位值。.
4	CH3G	通道 3 捕获或比较事件发生 参见 CH0G 描述
3	CH2G	通道 2 捕获或比较事件发生 参见 CH0G 描述
2	CH1G	通道 1 捕获或比较事件发生 参见 CH0G 描述
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被

置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。
0：不产生通道 0 捕获或比较事件
1：发生通道 0 捕获或比较事件

0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，如果选择了中央对齐或向上计数模式，计数器被清 0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0（TIMERx_CHCTL0）

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式:

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道 1 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 00：通道 1 配置为输出

		01：通道 1 配置为输入，IS1 映射在 CI1FE1 上 10：通道 1 配置为输入，IS1 映射在 CI0FE1 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 注意：当 CH1MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 信号输入高电平时，O0CPRE 参考信号被清 0 0：禁止通道 0 输出比较清零 1：使能通道 0 输出比较清零
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 100：强制为低。强制 O0CPRE 为低电平 101：强制为高。强制 O0CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00（比较模式）时此位不能被改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器的情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00 时此位不能被改变。
2	CH0COMFEN	通道 0 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，

CH0_O 被设置为比较电平而与比较结果无关。
0：禁止通道 0 输出比较快速.
1：使能通道 0 输出比较快速。

1:0 CH0MS[1:0] 通道 0 I/O 模式选择
这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。
00：通道 0 配置为输出
01：通道 0 配置为输入，IS0 映射在 CI0FE0 上
10：通道 0 配置为输入，IS0 映射在 CI1FE0 上
11：通道 0 配置为输入，IS0 映射在 ITS 上
注意：当 CH0MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入

输入捕获模式:

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	CH1CAPFLT[3:0]	通道 1 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH1CAPPSC[1:0]	通道 1 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH1MS[1:0]	通道 1 模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	

		4’b1101	5	f_{DTS}/32
		4’b1110	6	
		4’b1111	8	

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器

这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。

00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获

01：每 2 个事件触发一次捕获

10：每 4 个事件触发一次捕获

11：每 8 个事件触发一次捕获

1:0 CH0MS[1:0] 通道 0 模式选择

与输出比较模式相同

通道控制寄存器 1（TIMERx_CHCTL1）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMCEN	CH3COMCTL[2:0]			CH3COMSEN	CH3COMFEN	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	CH2COMFEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式:

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 CH0COMCEN 描述
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式 参见 CH0COMCTL 描述
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH3COMFEN	通道 3 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH3MS[1:0]	通道 3 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH3EN 位被清 0)时这些位才可以写。

00：通道 3 配置为输出
01：通道 3 配置为输入，IS3 映射在 CI3FE3 上
10：通道 3 配置为输入，IS3 映射在 CI2FE3 上
11：通道 3 配置为输入，IS3 映射在 ITS 上
注意：当 CH3MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入

7	CH2COMCEN	通道 2 输出比较清 0 使能 当此位被置 1，当检测到 ETIFP 输入高电平时，O2CPRE 参考信号被清 0 0：使能通道 2 输出比较清零 1：禁止通道 2 输出比较清零
6:4	CH2COMCTL[2:0]	通道 2 输出比较模式 此位定义了输出准备信号 O2CPRE 的输出比较模式，而 O2CPRE 决定了 CH2_O、CH2_ON 的值。另外，O2CPRE 高电平有效，而 CH2_O、CH2_ON 通道的极性取决于 CH2P、CH2NP 位。 000：时基。输出比较寄存器 TIMERx_CH2CV 与计数器 TIMERx_CNT 间的比较对 O2CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH2CV 相同时，强制 O2CPRE 翻转。 100：强制为低。强制 O2CPRE 为低电平 101：强制为高。强制 O2CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH2CV 时，O2CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH2CV 时，O2CPRE 为高电平，否则为低电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O2CPRE 电平才改变。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =00（比较模式）时此位不能被改变。
3	CH2COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 输出/比较影子寄存器 1：使能通道 2 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认影子寄存器情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =00 时此位不能被改变。
2	CH2COMFEN	通道 2 输出比较快速使能

当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O 被设置为比较电平而与比较结果无关。
0：禁止通道 2 输出比较快速.
1：使能通道 2 输出比较快速。

1:0 CH2MS[1:0] 通道 2 I/O 模式选择
这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH2EN 位被清 0)时这些位才可写。
00：通道 2 配置为输出
01：通道 2 配置为输入，IS2 映射在 CI2FE2 上
10：通道 2 配置为输入，IS2 映射在 CI3FE2 上
11：通道 2 配置为输入，IS2 映射在 ITS 上.
注意：当 CH2MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入

输入捕获模式:

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	CH3CAPFLT[3:0]	通道 3 输入捕获滤波控制 参见 CH0CAPFLT 描述
11:10	CH3CAPPSC[1:0]	通道 3 输入捕获预分频器 参见 CH0CAPPSC 描述
9:8	CH3MS[1:0]	通道 3 模式选择 与输出模式相同
7:4	CH2CAPFLT[3:0]	通道 2 输入捕获滤波控制 CI2 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。 数字滤波器的基本原理：根据 f_{SAMP} 对 CI2 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。 滤波器参数配置如下：

CH2CAPFLT [3:0]	采样次数	f_{SAMP}
4’b0000	无滤波器	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$

		4’b1011	6	
		4’b1100	8	
		4’b1101	5	f_{DTS}/32
		4’b1110	6	
		4’b1111	8	

3:2	CH2CAPPSC[1:0]	通道 2 输入捕获预分频器 这 2 位定义了通道 2 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH2EN =0 时，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01：每 2 个事件触发一次捕获 10：每 4 个事件触发一次捕获 11：每 8 个事件触发一次捕获
1:0	CH2MS[1:0]	通道 2 模式选择 与输出比较模式相同

通道控制寄存器 2（TIMERx_CHCTL2）

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	保留	CH3P	CH3EN	CH2NP	保留	CH2P	CH2EN	CH1NP	保留	CH1P	CH1EN	CH0NP	保留	CH0P	CH0EN
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH3NP	通道 3 互补输出极性 参考 CH0NP 描述
14	保留	必须保持复位值。
13	CH3P	通道 3 极性 参考 CH0P 描述
12	CH3EN	通道 3 使能 参考 CH0EN 描述
11	CH2NP	通道 2 互补输出极性 参考 CH0NP 描述
10	保留	必须保持复位值。

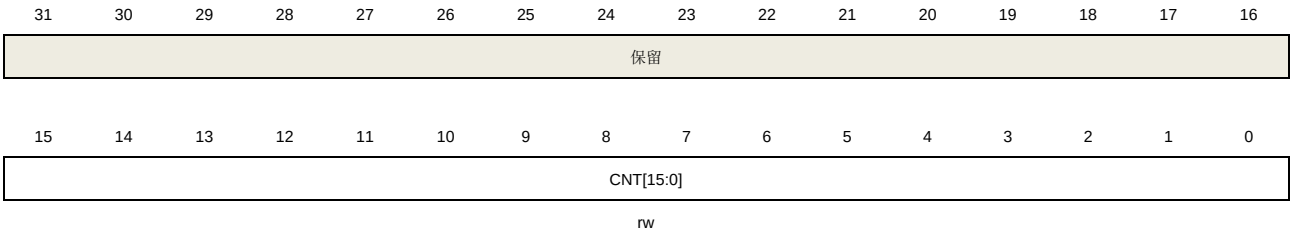
9	CH2P	通道 2 极性 参考 CH0P 描述
8	CH2EN	通道 2 使能 参考 CH0EN 描述
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述
6	保留	必须保持复位值。
5	CH1P	通道 1 极性 参考 CH0P 描述
4	CH1EN	通道 1 使能 参考 CH0EN 描述
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，该位保持 0。 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1：0]=11 或 10 时此位不能被更改。
2	保留	必须保持复位值。
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP， CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [CH0NP==0， CH0P==0]：把 CIxFE0 的上升沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 [CH0NP==0， CH0P==1]：把 CIxFE0 的下降沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 会被翻转。 [CH0NP==1， CH0P==0]：保留。 [CH0NP==1， CH0P==1]：把 CIxFE0 的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入模式时，将此位置 1 使能通道 0 上的捕获事件。 0：禁止通道 0 1：使能通道 0

计数器寄存器（TIMERx_CNT）

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

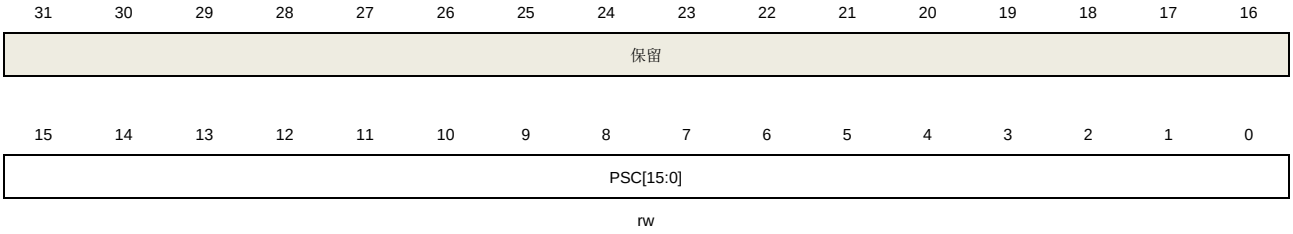


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器（TIMERx_PSC）

地址偏移：0x28
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

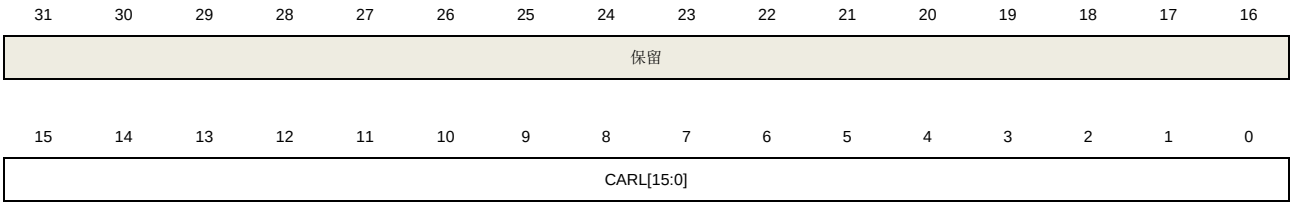


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

通道 0 捕获/比较值寄存器（TIMERx_CH0CV）

地址偏移：0x34
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较值寄存器（TIMERx_CH1CV）

地址偏移：0x38
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															

rw

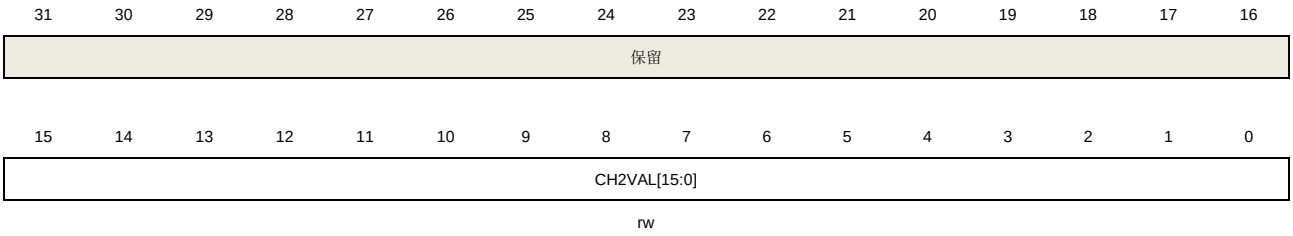
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值

当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。
当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 2 捕获/比较值寄存器（TIMERx_CH2CV）

地址偏移：0x3C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

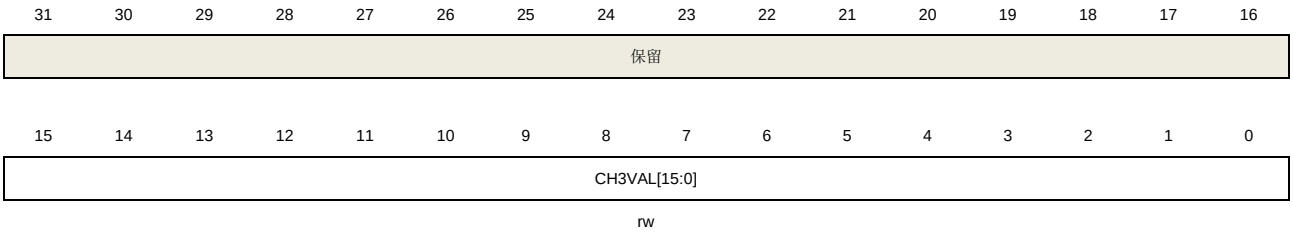


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 3 捕获/比较值寄存器（TIMERx_CH3CV）

地址偏移：0x40
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



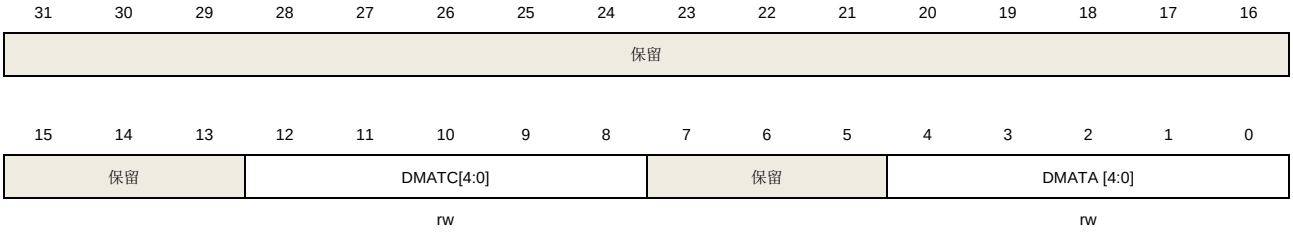
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值 当通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。

当通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

DMA 配置寄存器（TIMERx_DMACFG）

地址偏移：0x48
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

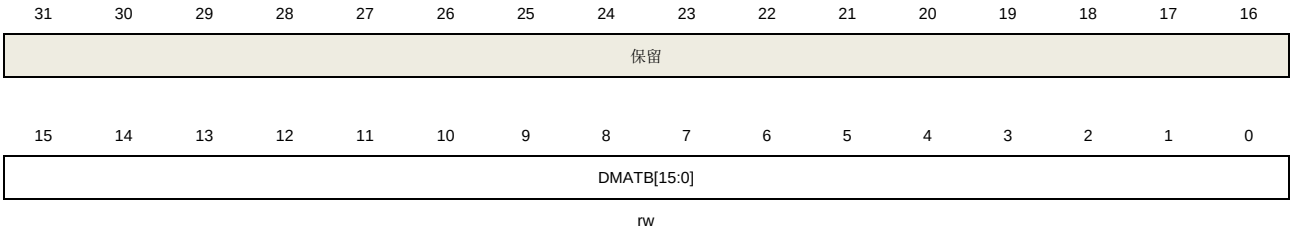


位/位域	名称	描述
31:13	保留	必须保持复位值。.
12:8	DMATC [4:0]	DMA 传输计数 该位域定义了 DMA 访问（读写）TIMERx_DMATB 寄存器的数量 n，n = (DMATC [4:0] +1). DMATC [4:0] 从 5'b0_0000 到 5'b1_0001.
7:5	保留	必须保持复位值。
4:0	DMATA [4:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMATB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。

DMA 发送缓冲区寄存器（TIMERx_DMATB）

地址偏移：0x4C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。.
15:0	DMATB [15:0]	DMA 发送缓冲

对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问
传输次数由硬件计算，范围为 0 到 DMATC。

配置寄存器（TIMERx_CFG）

地址偏移：0xFC
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CHVSEL	保留
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0：无影响
0	保留	必须保持复位值。

17.2. 通用定时器 L1（TIMERx, x=8,11）

17.2.1. 简介

通用定时器 L1(Timer8, 11)是两通道定时器，支持输入捕获和输出比较，可以产生 PEM 信号控制电机和电源管理。通用定时器 L1 含有一个 16 位无符号计数器。

通用定时器 L1 是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

定时器和定时器之间是相互独立，但是他们可以被同步在一起形成一个更大的定时器，这些定时器的计数器一致地增加。

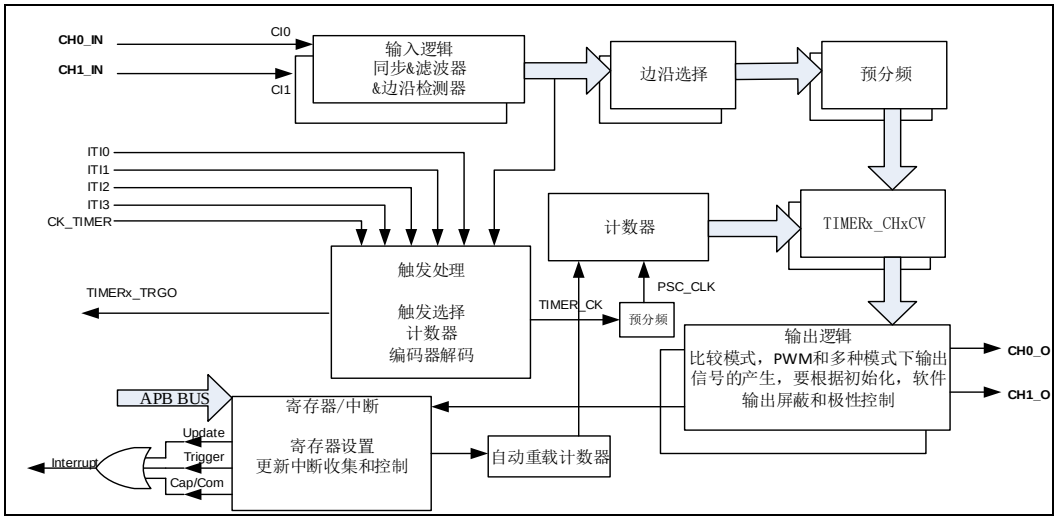
17.2.2. 主要特征

- 总通道数：2；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 计数模式：向上计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 自动重装载功能；
- 中断输出：更新事件，触发事件，比较/捕获事件和中止事件；
- 多个定时器的菊链使得一个定时器可以同时启动多个定时器；
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数；
- 定时器主-从管理。

17.2.3. 结构框图

*图 17-22. 通用定时器 L1 结构框图*提供了通用定时器 L1 的内部配置细节。

图 17-22. 通用定时器 L1 结构框图



17.2.4. 功能描述

时钟源配置

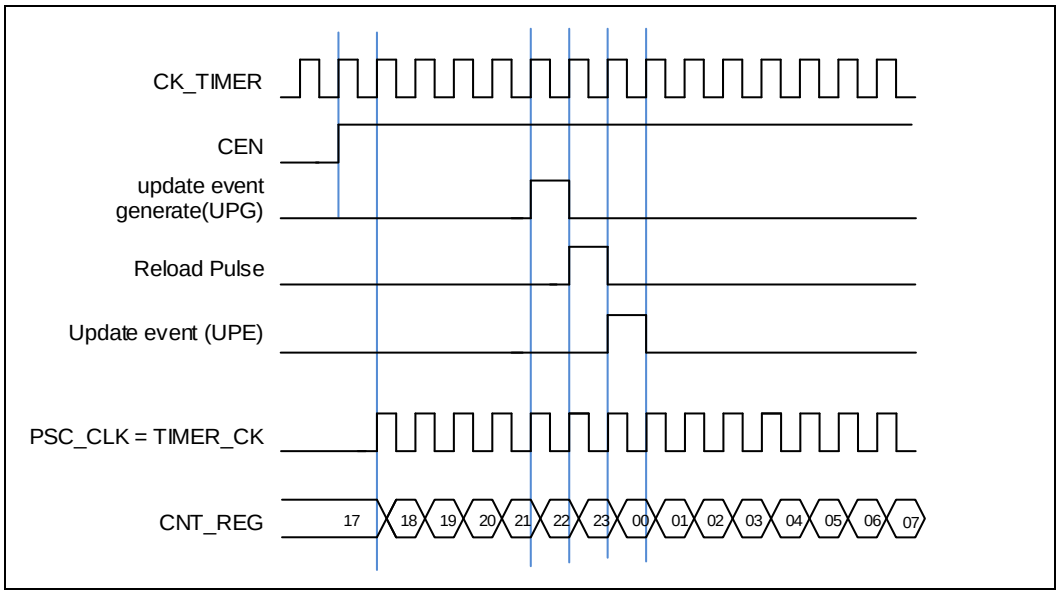
通用定时器 L1 可以由内部时钟源 CK_TIMER 或者由 SMC（TIMERx_SMCFG 寄存器位[2:0]）控制的复用时钟源驱动。

- SMC[2:0]==3'b000，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

如果 SMC[2:0]==3'b000，默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当 CEN 置位，CK_TIMER 经过预分频器（预分频值由 TIMERx_PSC 寄存器确定）产生 PSC_CLK。

如果将 TIMERx_SMCFG 寄存器的 SMC[2:0]设置为 0x1、0x2、0x3 和 0x7，预分频器被其他时钟源(由 TIMERx_SMCFG 寄存器的 TRGS [2:0]区域选择)驱动，在下文说明。当 SMC 位被设置为 0x4、0x5 和 0x6，计数器预分频器时钟源由内部时钟 CK_TIMER 驱动。

图 17-23. 内部时钟分频为 1 时，计数器的时序图



- SMC[2:0]==3'b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

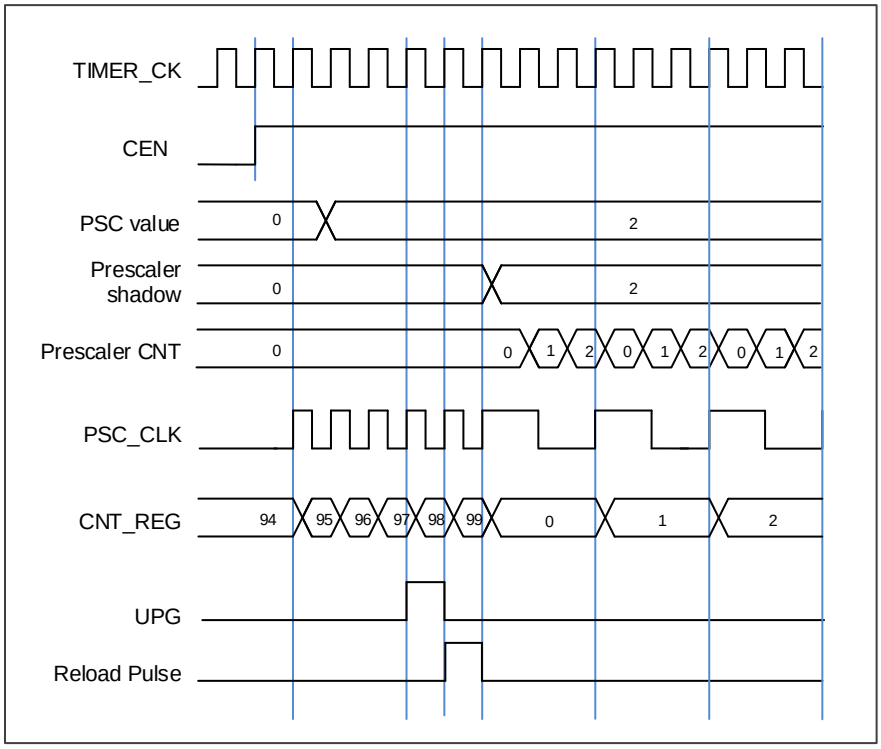
计数器预分频器可以在 TIMERx_CI0/ TIMERx_CI1 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x4，0x5 或 0x6 来选择。

计数器预分频器也可以在内部触发信号 ITI0/1/2/3 的上升沿计数。这种模式可以通过设置 SMC [2:0]为 0x7 同时设置 TRGS [2:0]为 0x0, 0x1, 0x2 或者 0x3。

时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 17-24. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值（定义在 TIMERx_CAR 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

图 17-25. 向上计数时序图，PSC=0/2 和 *图 17-26. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 17-25. 向上计数时序图，PSC=0/2

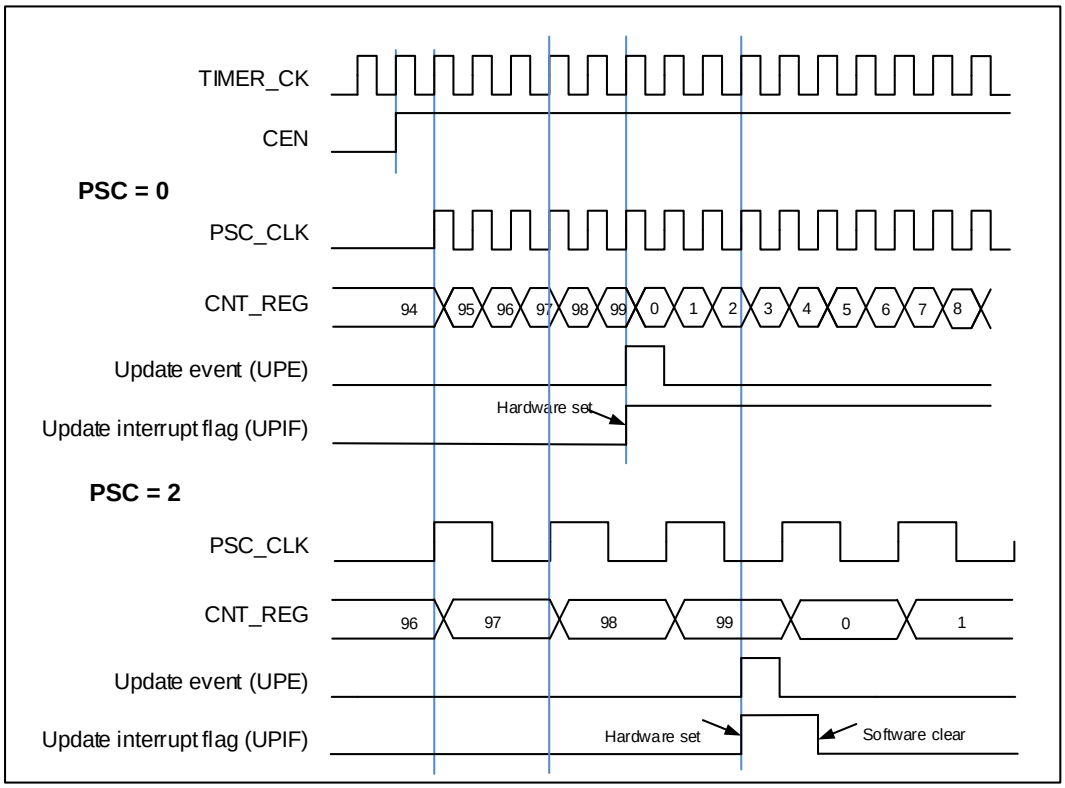
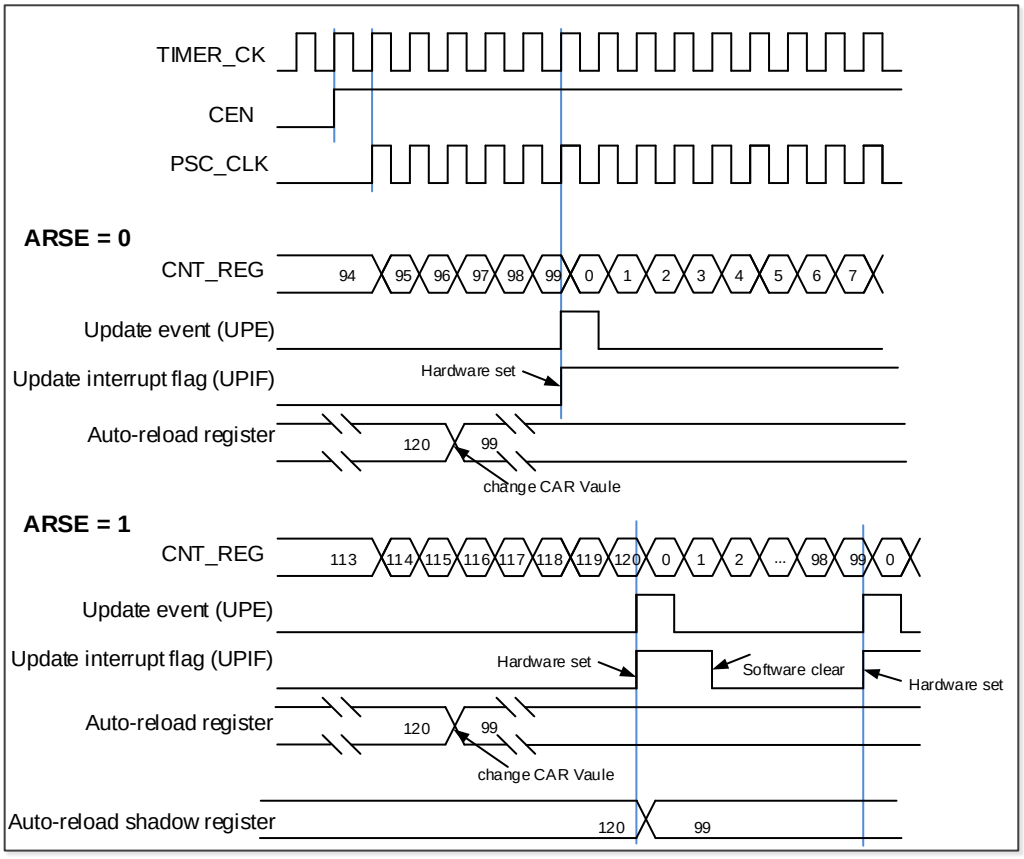


图 17-26. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



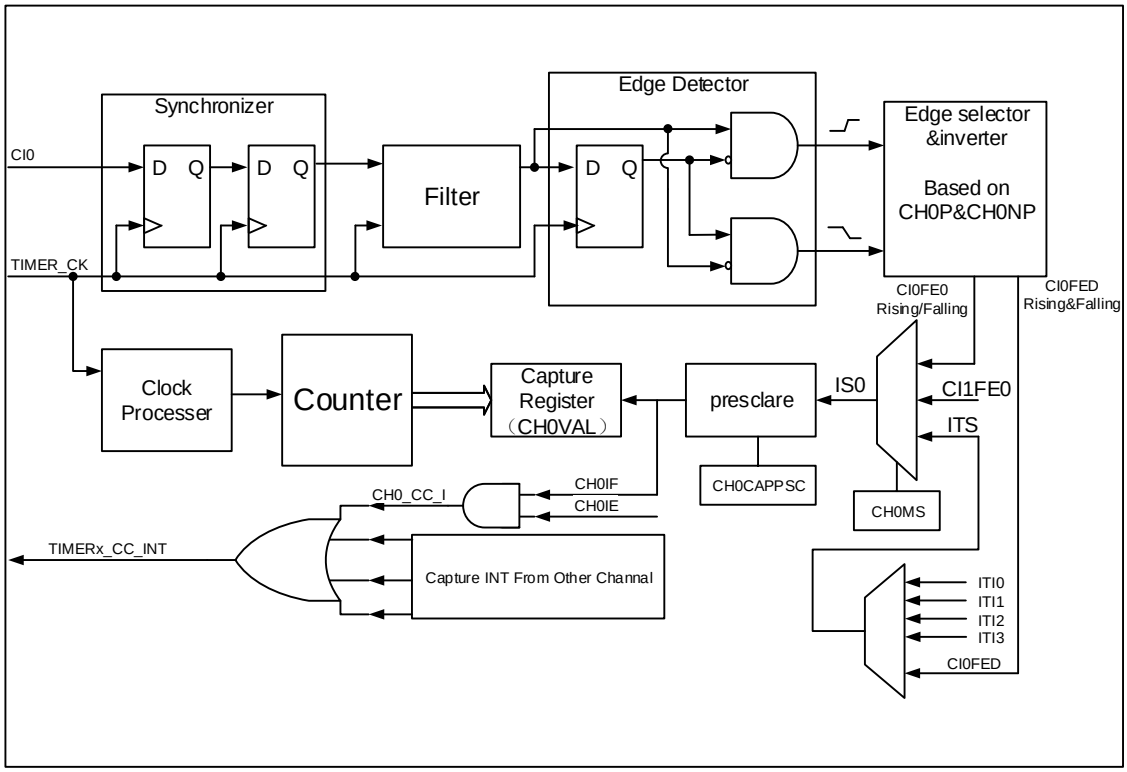
输入捕获与输出比较通道

通用定时器 L1 拥有两个个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 通道输入捕获功能

通道输入捕获功能允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV 寄存器会捕获计数器当前的值，同时 CHxIF 位被置 1，如果 CHxIE = 1 则产生通道中断。

图 17-27. 通道输入捕获原理



通道输入信号 CIx 先被 TIMER_CK 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CHxP 选择使用上升沿或者下降沿。配置 CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，CxCV 存储计数器的值。

第一步：滤波器配置（TIMERx_CHCTL0寄存器中CHxCAPFLT）：
根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。

第二步：边沿选择（TIMERx_CHCTL2寄存器中CHxP/CHxNP）：
配置CHxP/CHxNP选择上升沿或者下降沿。

第三步：捕获源选择（TIMERx_CHCTL0寄存器中CHxMS）：
一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式（CHxMS!=0x0），而且TIMERx_CxCV寄存器不能再被写。

第四步：中断使能（TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN）：

使能相应中断，可以获得中断和DMA请求。

第五步：捕获使能（TIMERx_CHCTL2寄存器中CHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE和CHxDEN的配置，相应的中断和DMA请求会被提出。

直接产生：软件设置CHxG位，会直接产生中断和DMA请求。

输入捕获模式也可用来测量 TIMERx_CHx 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 CI0。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CI0 并设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 2'b10，选择通道 1 捕获信号为 CI0 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。TIMERX_CH0CV 寄存器测量 PWM 的周期值，TIMERx_CH1CV 寄存器测量 PWM 占空比值。

- 通道输出比较功能

在输出比较模式，TIMERx 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 CxCV 寄存器与计数器的值匹配时，根据 CHxCOMCTL 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 CxCV 寄存器的值匹配时，CHxIF 位被置 1，如果 CHxIE = 1 则会产生中断，如果 CxCDE=1 则会产生 DMA 请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CxCDE位配置中断/DMA请求使能。

第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

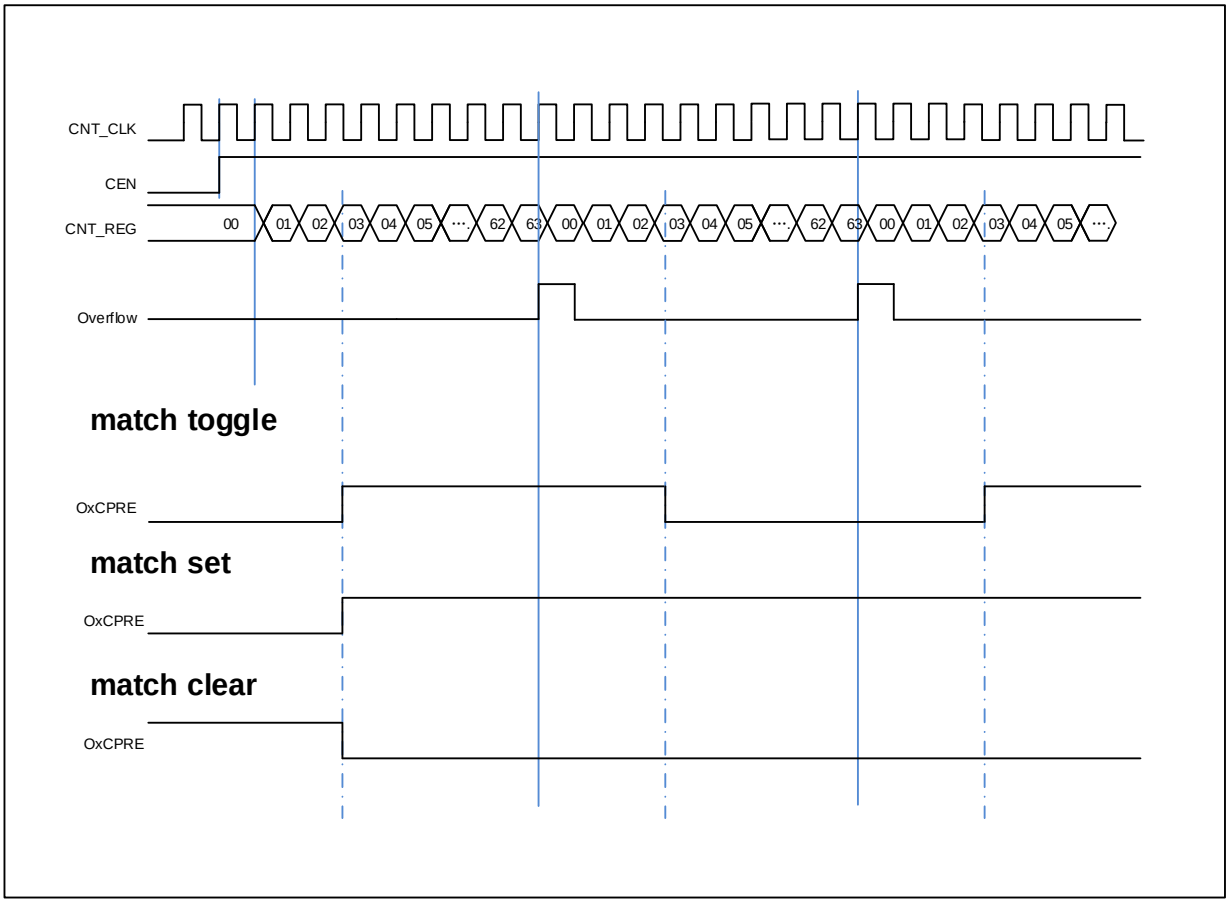
CxCV可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

*图 17-28. 三种输出比较模式*显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63,

CxCV=0x3。

图 17-28. 三种输出比较模式



输出 PWM 功能

在 PWM 输出模式下（PWM 模式 0 是配置 CHxCOMCTL 为 3'b110，PWM 模式 1 是配置 CHxCOMCTL 为 3'b111），通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

根据计数模式，我们可以分为两种 PWM 波：EAPWM(边沿对齐 PWM)和 CAPWM(中央对齐 PWM)。

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。*图 17-29. EAPWM 时序图*显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由（2*TIMERx_CAR 寄存器值）决定，占空比由（2*TIMERx_CHxCV 寄存器值）决定。*图 17-30. CAPWM 时序图*显示了 CAPWM 的输出波形和中断。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下(CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 17-29. EAPWM 时序图

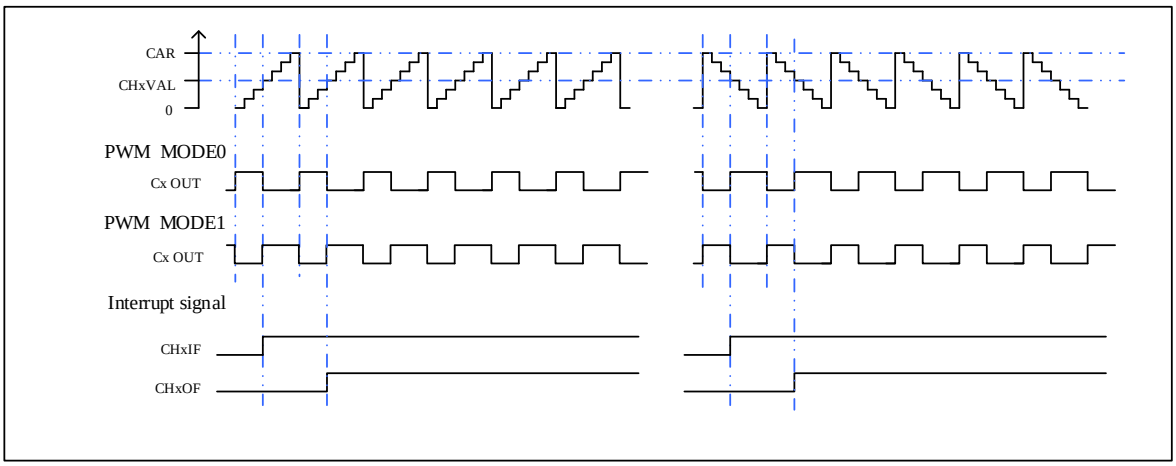
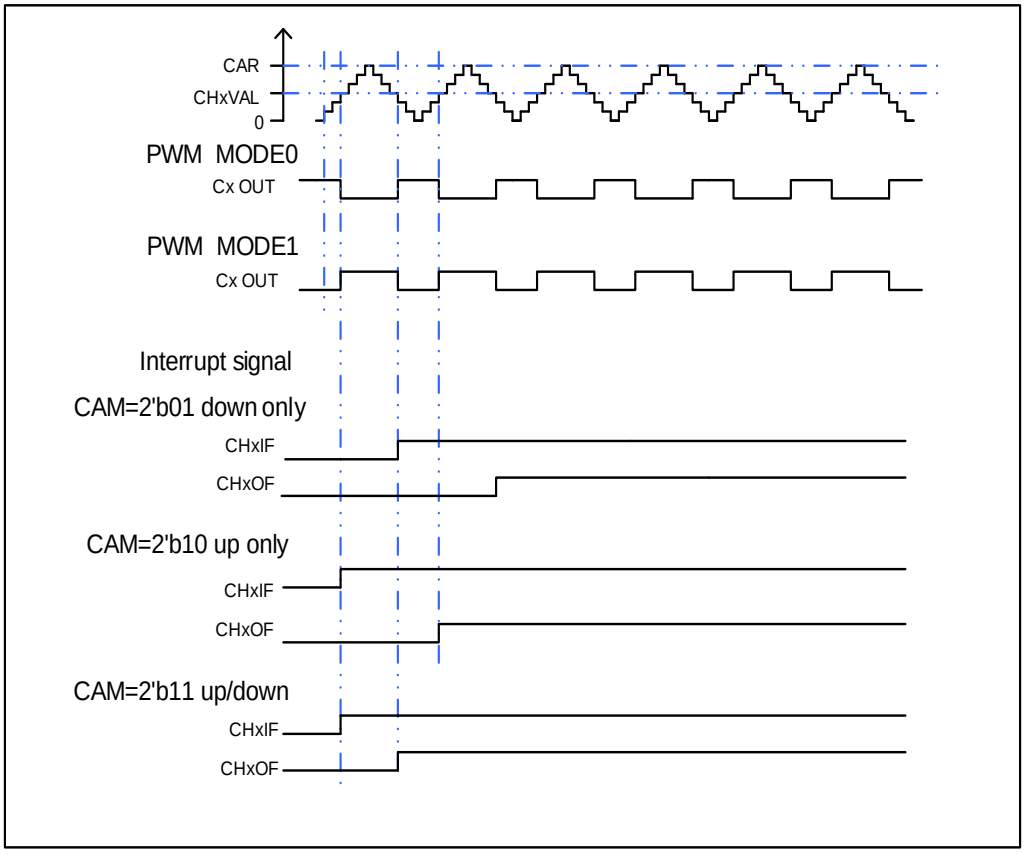


图 17-30. CAPWM 时序图



通道输出准备信号

当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域位 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL =0x04 或 0x05 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间间的比较结果。

设置 CHxCOMCEN=1，当由外部 ETI 引脚信号产生的 ETIFE 信号为高电平时，OxCPRE 被强制为低电平。在下一次更新事件到来时，OxCPRE 信号才会回到有效电平状态。

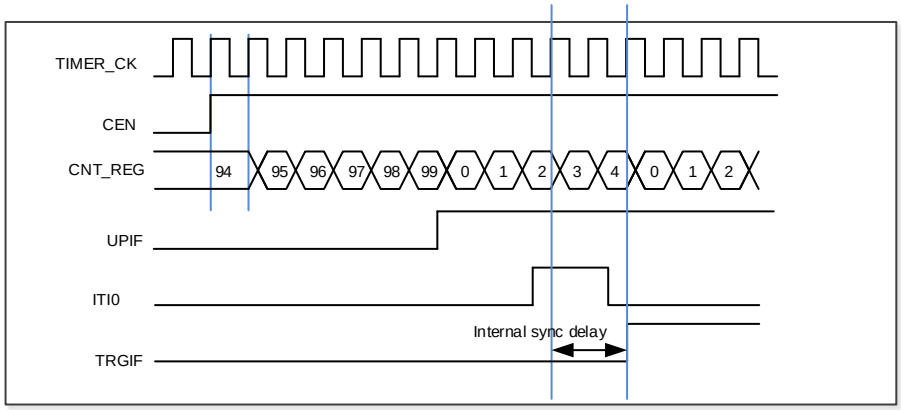
主-从管理

TIMERx 能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置 TIMERx_SMCFG 寄存器中的 SMC[2:0]配置这些模式。这些模式的输入触发源可以通过设置 TIMERx_SMCFG 寄存器中的 TRGS[2:0]来选择。

表 17-4. 从机模式列表和举例

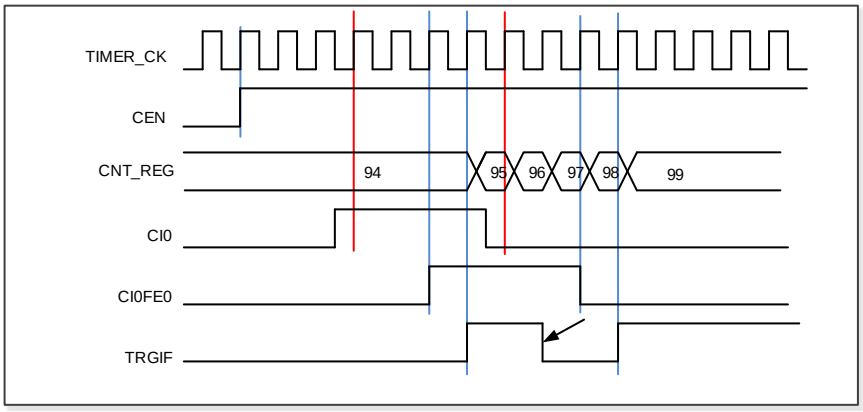
	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100 (复位模式) 3'b101 (暂停模式) 3'b110 (事件模式)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: 保留	如果触发源是CI0FE0或者CI1FE1，配置CHxP和CHxNP来选择极性和反相	触发源ITIx，滤波和预分频不可用 触发源CIx，配置CHxCAPFLT设置滤波，分频不可用
例1	复位模式 当触发输入上升沿，计数器清零重启	TRGIS[2:0]=3'b000 选择ITI0为触发源	触发源是ITI0，极性选择不可用	触发源是ITI0，滤波和预分频不可用

图 17-31. 复位模式下的控制电路



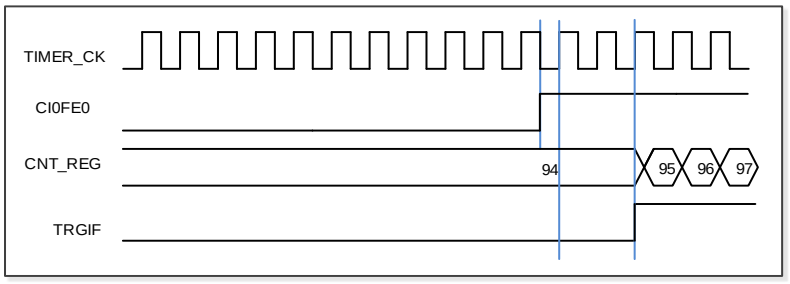
	模式选择	触发源选择	极性选择	滤波和预分频
例2	暂停模式 当触发输入为低的时候，计数器暂停计数	TRGIS[2:0]=3'b101 选择CI0FE0为触发源	TI0S=0.（非异或） CH0P==0，不反相. 在上升沿捕获	在这个例子中滤波被旁路

图 17-32. 暂停模式下的控制电路



例3	事件模式 触发输入的上升沿计数器开始计数	TRGIS[2:0]=3'b101 选择CI0FE0为触发源	TI0S=0.（非异或） CH0P==0，不反相.	在这个例子中滤波被旁路

图 17-33. 事件模式下的控制电路



单脉冲模式

单脉冲模式与重复模式是相反的，设置 TIMERx_CTL0 寄存器的 SPM 位置 1，则使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

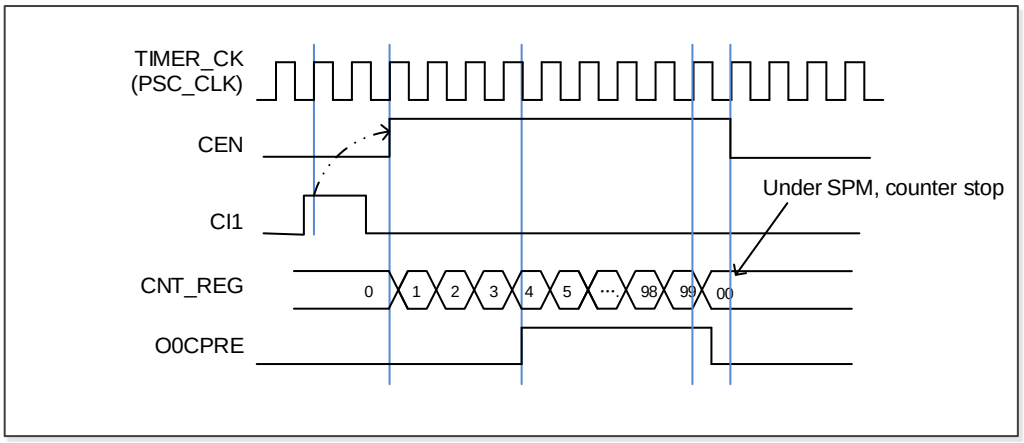
一旦设置定时器运行在单脉冲模式下，没有必要设置 TIMERx_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。触发信号沿或者软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止工作，计数值被保持。

在单脉冲模式下，有效的外部触发边沿会将 CEN 位置 1，使能计数器。然而，执行计数值和 TIMERx_CHxCV 寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户

可以将 TIMERx_CHCTL0/1 寄存器的 CHxCOMFEN 位置 1。单脉冲模式下，触发上升沿产生之后， OxCPRE 信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为 PWM0 或 PWM1 输出运行模式下时 CHxCOMFEN 位才可用，触发源来源于触发信号。

图 17-34. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 展示了一个例子。

图 17-34. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99



定时器互连

参考*通用定时器L0（TIMERx，x=1，2）定时器互连*。

定时器调试模式

当Cortex®-M23内核停止，DBG_CTL2寄存器中的TIMERx_HOLD配置位被置1，定时器计数器停止

17.2.5. TIMERx 寄存器（x=8,11）

TIMER8 基地址:0x4001 4C00

TIMER11 基地址:0x4000 1800

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CKDIV[1:0]		ARSE	保留			SPM	UPS	UPDIS	CEN
						rw		rw				rw	rw	rw	rw

位/位域	名称	描述
31:10	保留	必须保持复位值。
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(CK_TIMER) 与死区时间和数字滤波器采样时钟(DTS)之间的分频系数。 00：$f_{DTS}=f_{CK\_TIMER}$ 01：$f_{DTS}= f_{CK\_TIMER} /2$ 10：$f_{DTS}= f_{CK\_TIMER} /4$ 11：保留
7	ARSE	自动重载影子使能 0: 禁能 TIMERx_CAR 寄存器的影子寄存器 1: 使能 TIMERx_CAR 寄存器的影子寄存器
6:4	保留	必须保持复位值。
3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢

1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：更新事件禁能. 注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化
0	CEN	计数器使能 0: 计数器禁能 1: 计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和正交译码器模式才能工作。

从模式配置寄存器（TIMERx_SMCFG)

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								MSM	TRGS[2:0]			Res.	SMC[2:0]		
								rw	rw			rw	rw		

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	MSM	主-从模式 该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。 0: 主从模式禁能 1: 主从模式使能
6:4	TRGS[2:0]	触发选择 该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0

		110: CI1FE1 111: 保留 从模式被使能后这些位不能改
3	保留	必须保持复位值。
2:0	SMC[2:0]	从模式控制 000: 关闭从模式. 如果 CEN=1，则预分频器直接由内部时钟驱动 001: 保留 010: 保留 011: 保留 100: 复位模式. 选中的触发输入的上升沿重新初始化计数器，并且更新影子寄存器. 101:暂停模式. 当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止 110: 事件模式.计数器在触发输入的上升沿启动。 111: 外部时钟模式 0. 选中的触发输入的上升沿驱动计数器

DMA 和中断使能寄存器（TIMERx_DMAINTEN）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TRGIE	保留			CH1IE	CH0IE	UPIE
									rw		rw		rw	rw	rw

位/位域	名称	描述
31:7	保留	必须保持复位值。.
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5:3	保留	必须保持复位值。.
2	CH1IE	通道 1 比较/捕获中断使能 0：禁止通道 1 中断 1：使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0：禁止通道 0 中断 1：使能通道 0 中断

0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器（TIMERx_INTF）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CH1OF	CH0OF	保留		TRGIF	保留			CH1IF	CH0IF	UPIF
					rc_w0	rc_w0			rc_w0				rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:11	保留	必须保持复位值。.
10	CH1OF	通道 1 捕获溢出标志 参见 CH0OF 描述
9	CH0OF	通道 0 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0. 0: 无捕获溢出中断发生 1: 发生了捕获溢出中断
8:7	保留	必须保持复位值。.
6	TRGIF	触发中断标志 当发生触发事件时，此标志会置 1，此位由软件清 0。当暂停模式使能时，触发输入的任意边沿都可以产生触发事件。否则，其它模式时，仅在触发输入端检测到有效边沿，产生触发事件。 0：无触发事件产生 1：触发中断产生
5:3	保留	必须保持复位值。.
2	CH1IF	通道 1 比较/捕获中断标志 参见 CH0IF 描述
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 0：无通道 0 中断发生 1：通道 0 中断发生

0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器（TIMERx_SWEVG）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									TRGG	保留			CH1G	CH0G	UPG
									w				w	w	w

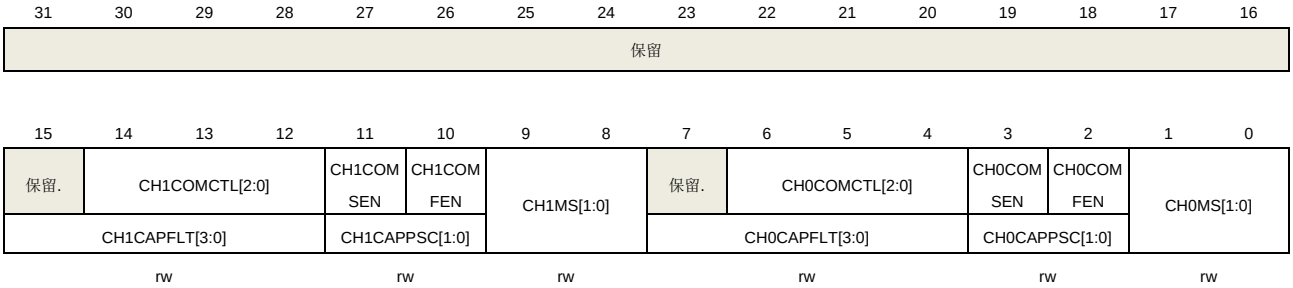
位/位域	名称	描述
31:7	保留	必须保持复位值。.
6	TRGG	触发事件产生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，TIMERx_INTF 寄存器的 TRGIF 标志位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：无触发事件产生 1：产生触发事件
5:3	保留	必须保持复位值。.
2	CH1G	通道 1 捕获或比较事件发生 参见 CH0G 描述
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。 0：不产生通道 0 捕获或比较事件 1：发生通道 0 捕获或比较事件
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，如果选择了中央对齐或向上计数模式，计数器被清 0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0（TIMERx_CHCTL0）

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能按字(32位)访问



输出比较模式：

位/位域	名称	描述
31:15	保留	必须保持复位值。.
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道 1 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH1EN 位被清 0)时这些位才可以写。 00：通道 1 配置为输出 01：通道 1 配置为输入，IS1 映射在 CI1FE1 上 10：通道 1 配置为输入，IS1 映射在 CI0FE1 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 注意：当 CH1MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入
7	保留	必须保持复位值。.
6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 此位定义了输出准备信号 O0CPRE 的输出比较模式，而 O0CPRE 决定了 CH0_O、CH0_ON 的值。另外，O0CPRE 高电平有效，而 CH0_O、CH0_ON 通道的极性取决于 CH0P、CH0NP 位。 000：时基。输出比较寄存器 TIMERx_CH0CV 与计数器 TIMERx_CNT 间的比较对 O0CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。

010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。
011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。
100：强制为低。强制 O0CPRE 为低电平
101：强制为高。强制 O0CPRE 为高电平
110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。
111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为低电平，否则为高电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为高电平，否则为低电平。
如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，O0CPRE 电平才改变。
当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00（比较模式）时此位不能被改变。

3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下(SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =00 时此位不能被改变。
2	CH0COMFEN	通道 0 输出比较快速使能 当该位为 1 时，如果通道配置为 PWM0 模式或者 PWM1 模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O 被设置为比较电平而与比较结果无关。 0：禁止通道 0 输出比较快速. 1：使能通道 0 输出比较快速。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2 寄存器的 CH0EN 位被清 0)时这些位才可写。 00：通道 0 配置为输出 01：通道 0 配置为输入，IS0 映射在 CI0FE0 上 10：通道 0 配置为输入，IS0 映射在 CI1FE0 上 11：通道 1 配置为输入，IS1 映射在 ITS 上 注意：当 CH0MS[1:0]=11 时，需要通过 TRGS 位（位于 TIMERx_SMCFG 寄存器）选择内部触发输入

输入捕获模式：

位/位域	名称	描述
31:16	保留	必须保持复位值。

15:12 CH1CAPFLT[3:0] 通道 1 输入捕获滤波控制
参见 CH0CAPFLT 描述

11:10 CH1CAPPSC[1:0] 通道 1 输入捕获预分频器
参见 CH0CAPPSC 描述

9:8 CH1MS[1:0] 通道 1 模式选择
与输出模式相同

7:4 CH0CAPFLT[3:0] 通道 0 输入捕获滤波控制
CI0 输入信号可以通过数字滤波器进行滤波，该位域配置滤波参数。
数字滤波器的基本原理：根据 f_{SAMP} 对 CI0 输入信号进行连续采样，并记录信号相同电平的次数。达到该位配置的滤波参数后，认为是有效电平。
滤波器参数配置如下：

CH0CAPFLT [3:0]	采样次数	f_{SAMP}
4'b0000	无滤波器	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$
4'b1011	6	
4'b1100	8	
4'b1101	5	$f_{DTS}/32$
4'b1110	6	
4'b1111	8	

3:2 CH0CAPPSC[1:0] 通道 0 输入捕获预分频器
这 2 位定义了通道 0 输入的预分频系数。当 TIMERx_CHCTL2 寄存器中的 CH0EN =0 时，则预分频器复位。
00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获
01：每 2 个事件触发一次捕获
10：每 4 个事件触发一次捕获
11：每 8 个事件触发一次捕获

1:0 CH0MS[1:0] 通道 0 模式选择
与输出比较模式相同

预分频寄存器（TIMERx_PSC）

地址偏移： 0x28
复位值： 0x0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

通道控制寄存器 2（TIMERx_CHCTL2）

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.								CH1NP	保留	CH1P	CH1EN	CH0NP	保留	CH0P	CH0EN
								rw		rw	rw	rw		rw	rw

位/位域	名称	名称
31:8	保留	必须保持复位值。
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述
6	保留	必须保持复位值。
5	CH1P	通道 1 极性 参考 CH0P 描述
4	CH1EN	通道 1 使能 参考 CH0EN 描述
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，此位定义了互补输出信号的极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。

		当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	保留	必须保持复位值。
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平为有效电平 1：通道0低电平为有效电平 当通道 0 配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]：把 CIxFE0 的上升沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 [CH0NP==0, CH0P==1]：把 CIxFE0 的下降沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 会被翻转。 [CH0NP==1, CH0P==0]：保留。 [CH0NP==1, CH0P==1]：把 CIxFE0 的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
0	CH0EN	通道 0 捕获/比较使能 当通道 0 配置为输出模式时，将此位置 1 使能 CH0_O 信号有效。当通道 0 配置为输入模式时，将此位置 1 使能通道 0 上的捕获事件。 0：禁止通道 0 1：使能通道 0

计数器寄存器（TIMERx_CNT）

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

通道 0 捕获/比较值寄存器（TIMERx_CH0CV）

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较值寄存器（TIMERx_CH1CV）

地址偏移：0x38

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

输入重映射寄存器（TIMERx_IRMP）（x=8）

地址偏移：0x50

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CI0_RMP[1:0]	
														Rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。.
1:0	CI0_RMP[1:0]	通道 0 输入重映射 00：通道 0 输入连接到 GPIO(TIMER8_CH0) 01：通道 0 输入连接到 LXTAL 10：通道 0 输入连接到 HXTAL/32 时钟 11: 通道 0 输入连接到 CKOUTSEL

输入重映射寄存器（TIMERx_IRMP）（x=11）

地址偏移：0x50

复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CI0_RMP[1:0]	
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。.
1:0	CI0_RMP[1:0]	通道 0 输入重映射 00：通道 0 输入连接到 GPIO(TIMER11_CH0) 01：通道 0 输入连接到 IRC32K 10：通道 0 输入连接到 LXTAL 11: 通道 0 输入连接到 RTC_OUT

配置寄存器（TIMERx_CFG）

地址偏移：0xFC
复位值：0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CHVSEL	保留
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 1：当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0：无影响
0	保留	必须保持复位值。

17.3. 基本定时器（TIMERx，x=5，6）

17.3.1. 简介

基本定时器(TIMER5，6)包含一个无符号 16 位计数器。基本定时器可以配置产生 DMA 请求，TRGO 触发连接到 DAC。

17.3.2. 主要特征

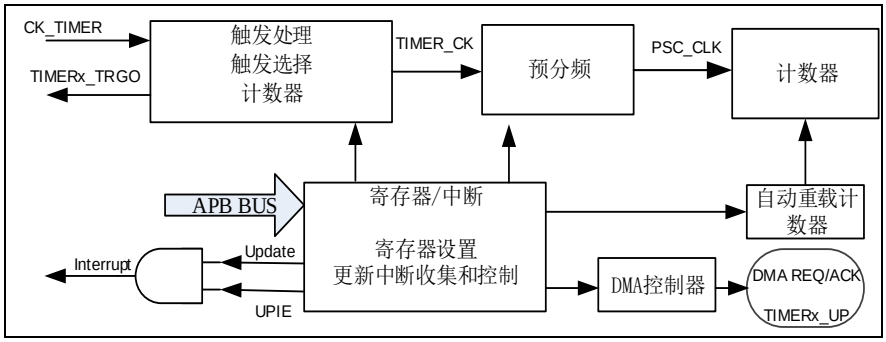
- 计数器宽度：16位；
- 定时器时钟源只有内部时钟；

- 计数模式：向上计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 自动重装载功能；
- 中断输出和DMA请求：更新事件。

17.3.3. 结构框图

*图 17-35. 基本定时器结构框图*提供了基本定时器内部配置的细节。

图 17-35. 基本定时器结构框图



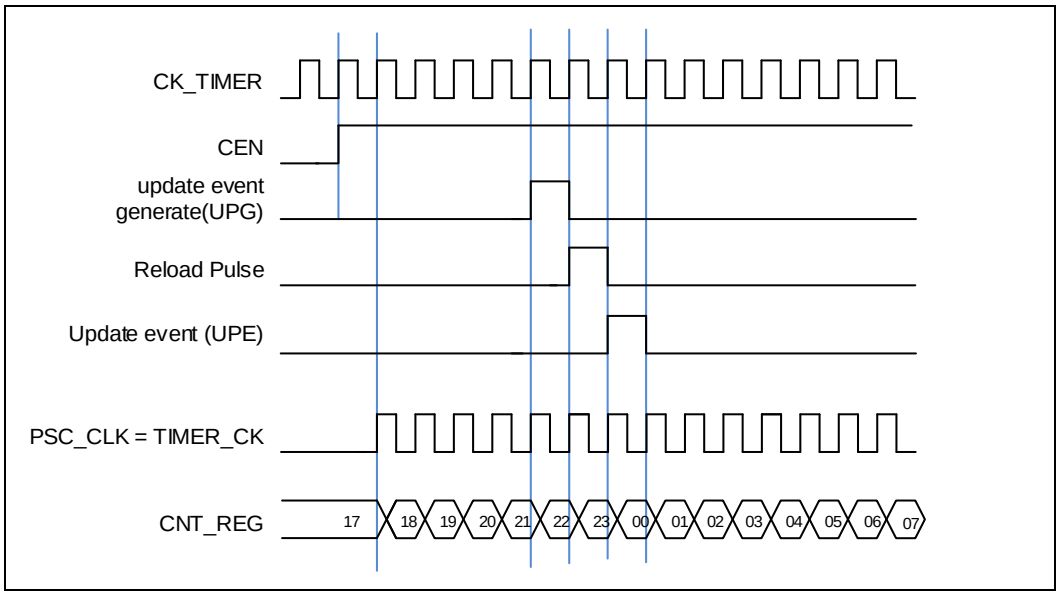
17.3.4. 功能说明

时钟源配置

基本定时器可以是内部时钟源 CK_TIMER 驱动。

基本定时器仅有一个时钟源 CK_TIMER，用来驱动计数器预分频器。当 CEN 置位，CK_TIMER 经过预分频器（预分频值由 TIMERx_PSC 寄存器确定）产生 PSC_CLK。

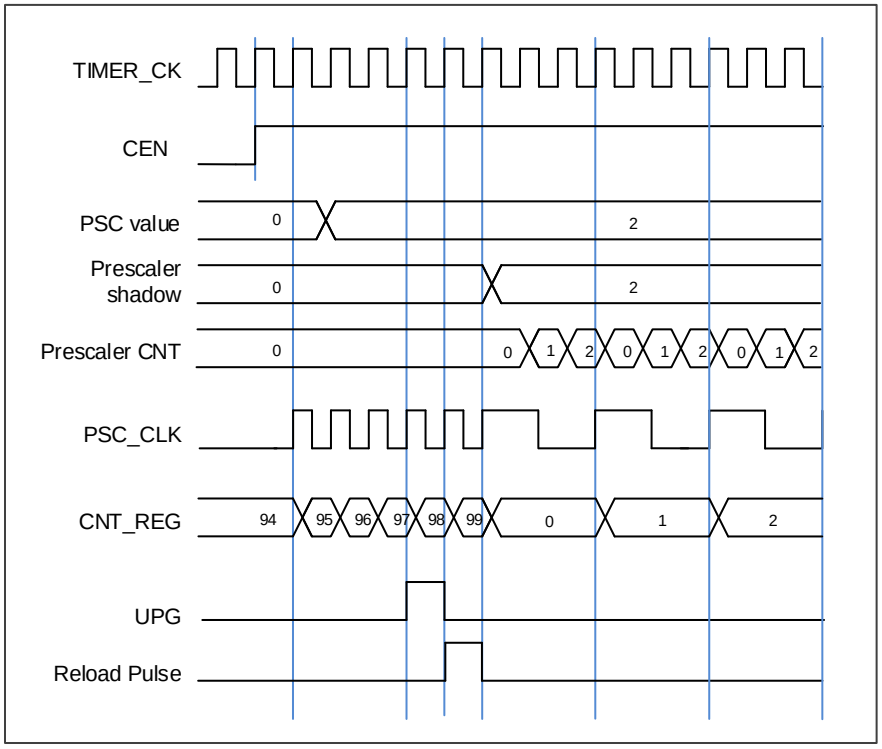
图 17-36. 内部时钟分频为 1 时，计数器的时序图



时钟预分频器

预分频器可以将定时器的时钟（TIMER_CK)频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 17-37. 当 PSC 数值从 0 变到 2 时，计数器的时序图



计数器向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值（定义在 TIMERx_CAR 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当通过 TIMERx_SWEVG 寄存器的 UPG 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有影子寄存器(计数器自动重载寄存器，预分频寄存器)都将被更新。

图 17-38. 向上计数时序图，PSC=0/2 和 *图 17-39. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值*给出了一些例子，当 TIMERx_CAR=0x99 时，计数器在不同预分频因子下的行为。

图 17-38. 向上计数时序图，PSC=0/2

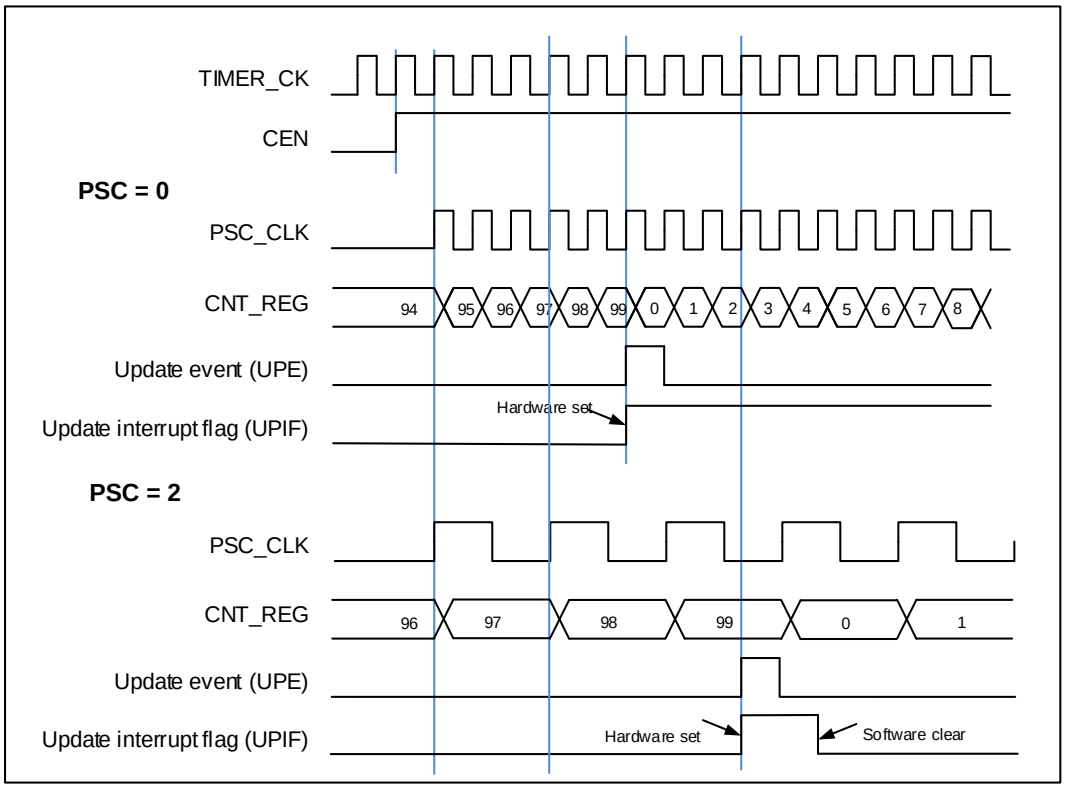
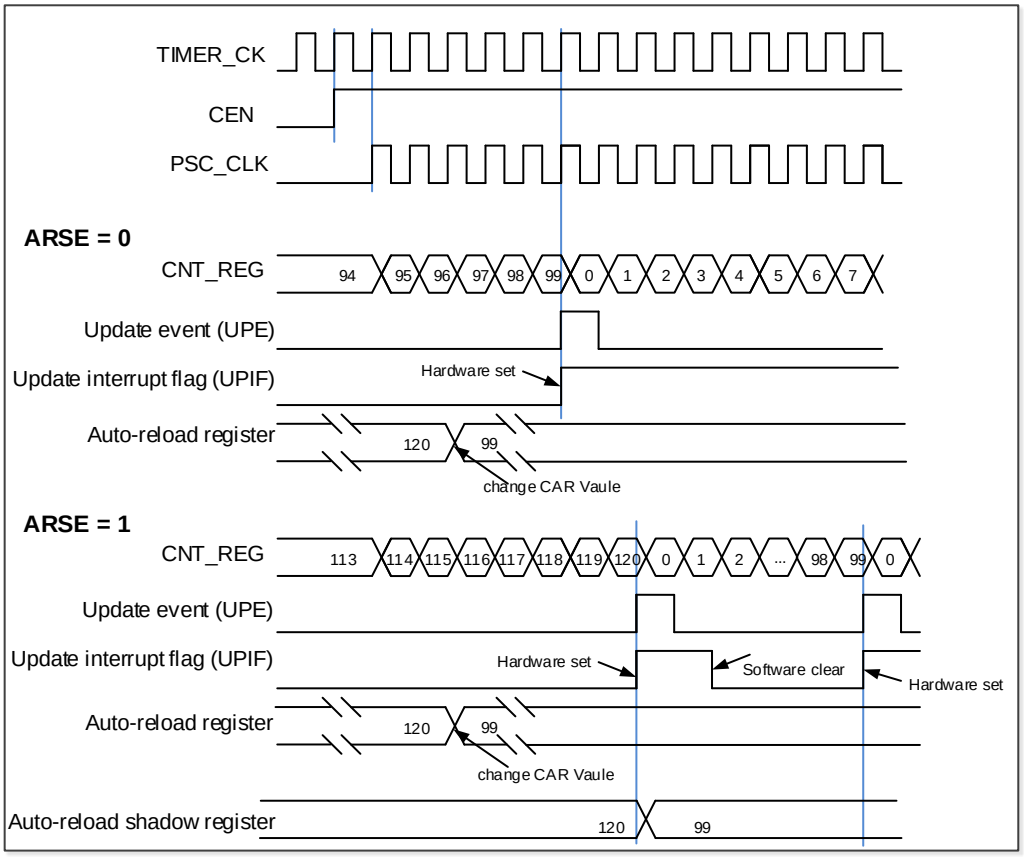


图 17-39. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



单脉冲模式

单脉冲模式与重复模式是相反的，设置TIMERx_CTL0寄存器的SPM位置1，则使能单脉冲模式。当SPM置1，计数器在下次更新事件到来后清零并停止计数。

一旦设置定时器运行在单脉冲模式下，需要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。

定时器调试模式

当 Cortex®-M23 内核停止，DBG_CTL0 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

17.3.5. TIMERx 寄存器（x=5，6）

TIMER5基地址：0x4000 1000

TIMER6基地址：0x4000 1400

控制寄存器 0（TIMERx_CTL0）

地址偏移：0x00
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								ARSE	保留			SPM	UPS	UPDIS	CEN
								rw				rw	rw	rw	rw

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器 1：使能 TIMERx_CAR 寄存器的影子寄存器
6:4	保留	必须保持复位值。
3	SPM	单脉冲模式 0：单脉冲模式禁能，更新事件发生后，计数器继续计数 1：单脉冲模式使能，在下一次更新事件发生时，计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	UPDIS	禁止更新. 该位用来使能或禁能更新事件的产生 0：更新事件使能. 更新事件发生时，相应的影子寄存器被装入预装载值，以下事件均会产生更新事件： UPG位被置1 计数器溢出/下溢 复位模式产生的更新

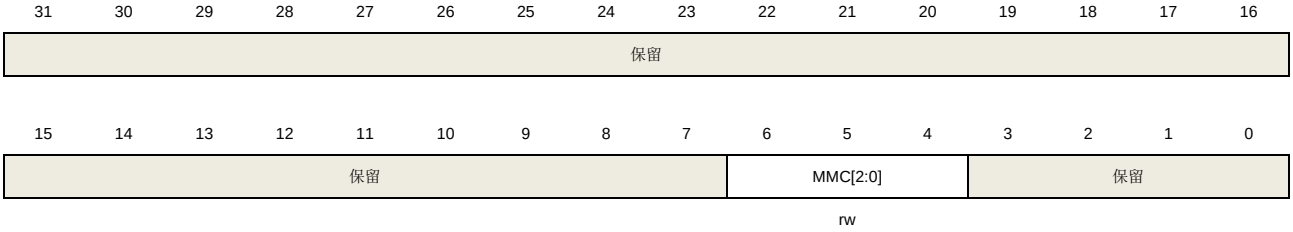
1：更新事件禁能.
注意：当该位被置 1 时，UPG 位被置 1 或者复位模式不会产生更新事件，但是计数器和预分频器被重新初始化

0	CEN	计数器使能 0：计数器禁能 1：计数器使能

控制寄存器 1（TIMERx_CTL1）

地址偏移：0x04
复位值：0x0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	MMC[2:0]	主模式控制 这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定 011~111：保留。
3:0	保留	必须保持复位值。

DMA 和中断使能寄存器（TIMERx_DMAINTEN）

地址偏移：0x0C
复位值：0x0000

该寄存器只能按字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							UPDEN	保留							UPIE
							rw								rw

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	UPDEN	更新 DMA 请求使能 0：禁止更新 DMA 请求 1：使能更新 DMA 请求
7:1	保留	必须保持复位值。
0	UPIE	更新中断使能 0：禁止更新中断 1：使能更新中断

中断标志寄存器（TIMERx_INTF）

地址偏移：0x10
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															UPIF
															rc_w0

位/位域	名称	描述
31:1	保留	必须保持复位值。
0	UPIF	更新中断标志 此位在更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器（TIMERx_SWEVG）

地址偏移：0x14
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															UPG
															w

位/位域	名称	描述
31:1	保留	必须保持复位值。
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1 并且向上计数模式，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

计数器寄存器（TIMERx_CNT）

地址偏移：0x24
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器（TIMERx_PSC）

地址偏移：0x28
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

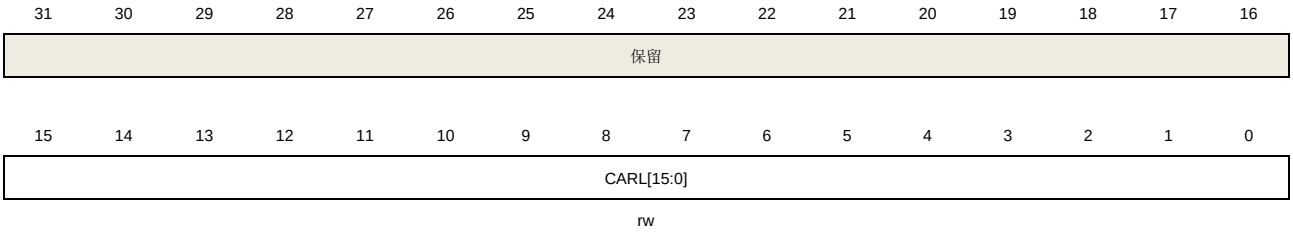
位/位域	名称	描述

31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以(PSC+1)，每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C
复位值：0x0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

18. 低功耗定时器（LPTIMER）

18.1. 简介

LPTIMER 是一个 32 位的定时器，它能够在除待机模式（Standby mode）以外的所有功耗模式下运行。LPTIMER 提供了灵活的时钟机制，在将功耗降至最低的同时，还可以实现所需的功能和性能。

LPTIMER 可以用作没有内部时钟源的脉冲计数器。LPTIMER 可以将系统从低功耗模式唤醒，非常适合于以极低的功耗实现超时模式的场合。

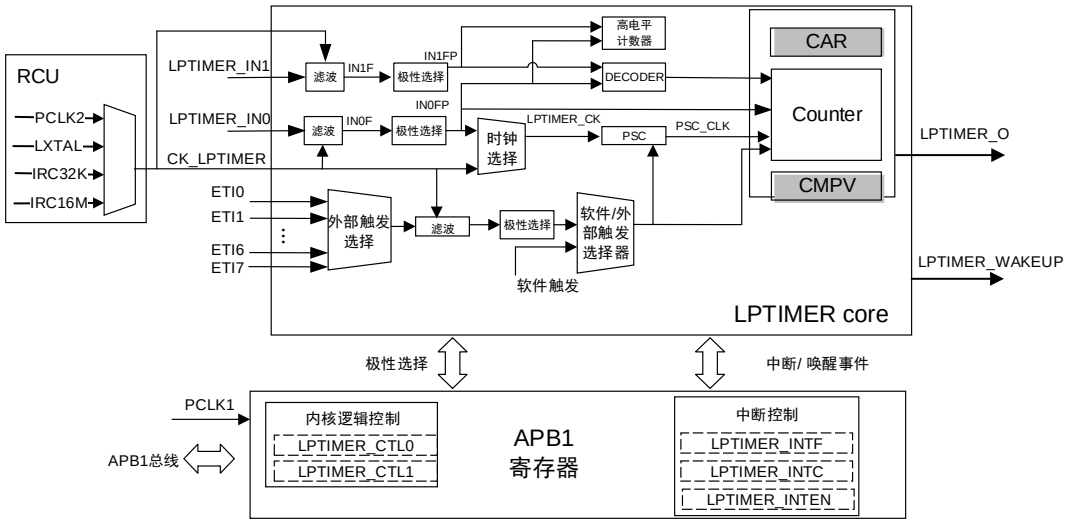
18.2. 主要特征

- 计数器宽度：32位
- 时钟源可选：
  - 内部时钟源：内部16MHz RC晶振（IRC16M），内部32KHz RC晶振（IRC32K），32.768 KHz低速晶振（LXTAL）和APB2时钟（PCLK2）
  - 外部时钟源：来自于LPTIMER_IN0引脚上的时钟源（作为脉冲计数器）
- 计数模式：向上计数
- 运行模式：连续计数模式或单次计数模式
- 可编程的预分频器：3位
- 通道输出可配置：可编程的PWM模式，单脉冲模式，置位模式
- 自动重装载功能
- 中断输出
- 可选择的触发：软件触发或硬件输入触发
- 正交译码器模式：正交译码器模式0和正交译码器模式1

18.3. 结构框图

*图 18-1. LPTIMER 结构框图*提供了低功耗定时器的内部配置细节。

图 18-1. LPTIMER 结构框图



18.4. 功能描述

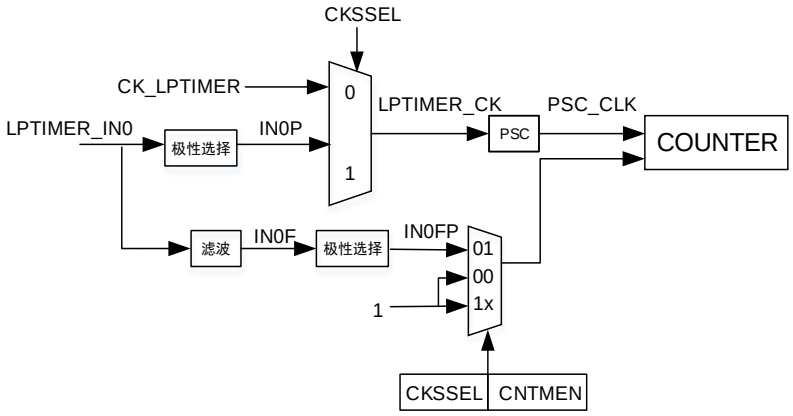
18.4.1. 时钟源配置

LPTIMER 可以由多个时钟源提供时钟，如内部时钟源有：内部 16MHz RC 晶振（IRC16M），内部 32KHz RC 晶振（IRC32K），32.768 KHz 低速晶振（LXTAL）和 APB2 时钟（PCLK2），这些时钟源来自于复位和时钟单元 RCU。

LPTIMER 还可以使用外部引脚 LPTIMER_IN0 上的外部时钟信号作为时钟，当使用外部时钟作为时钟源时，LPTIMER 有以下两种配置方式：

- **Case 0**：当LPTIMER由外部信号提供时钟时，还需要APB2或其他晶振（如IRC16M、IRC32K 和LXTAL）同时提供内部时钟信号；
- **Case 1**：LPTIMER的时钟仅由LPTIMER_IN0引脚上外部时钟信号提供。在进入低功耗模式后，所有晶振关闭，此配置可用于实现超时模式或脉冲计数器功能。

图 18-2. LPTIMER 时钟源选择



LPTIMER 可以由内部时钟信号或外部时钟信号（由 LPTIMER_CTL0 寄存器中的 CNTMEN 位和 CKSSEL 位控制）驱动。CKSSEL 位用于选择哪个时钟驱动计数器预分频器，默认时钟源为 PCLK2。CNTMEN 位用于选择哪个时钟信号驱动 LPTIMER 计数器。

当 LPTIMER 使用外部时钟信号时，CKPSEL 用于配置计数器的有效边沿。计数器可以由外部时钟的上升沿、下降沿或双边沿更新，具体由 CKPSEL [1:0]位域配置。

需要注意的是，当由外部引脚 LPTIMER_IN0 提供外部时钟信号时，如果有效边沿选择双边沿（CKPSEL=2'b10）或者 LPTIMER_IN0 引脚由数字滤波器采样（ECKFLT≠2'b00），则还需要提供内部时钟信号（Case 0）。在这种情况下，内部时钟信号频率至少是外部时钟频率的 4 倍。

可以根据 CKSSEL 位和 CNTMEN 位的配置，选择以下的时钟模式：

- CKSSEL = 0：LPTIMER时钟由内部时钟信号提供

  - 内部时钟模式0（CNTMEN = 0）

  LPTIMER由内部时钟信号提供时钟，计数器在内部时钟的每个脉冲进行计数。

  - 内部时钟模式1（CNTMEN = 1）

  外部时钟信号（LPTIMER_IN0）由内部时钟进行采样，因此，为了保证不丢失任何事件，外部时钟的变化频率不能超过内部时钟的频率。并且，LPTIMER的内部时钟信号不能预分频（PSC [2:0] = 000）。

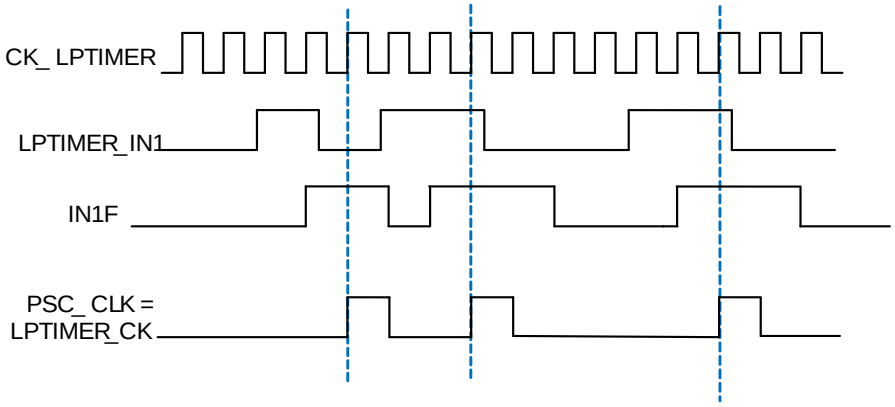
- CKSSEL = 1：LPTIMER时钟由外部时钟信号提供

  这种情况下，可以将CNTMEN位置1或清零，LPTIMER不需要使用内部时钟源（除非使能了输入滤波或是选择了双边沿作为外部时钟的有效边沿）。LPTIMER_IN0引脚上的外部信号作为LPTIMER的系统时钟，该情况适用于没有嵌入式晶振的工作情况。

  这种情况下，LPTIMER的计数器可以在外部时钟信号的上升沿或下降沿计数，不能在双边沿计数。

  由于LPTIMER_IN0引脚上的外部时钟信号也用于驱动LPTIMER的内核逻辑部分，因此，在计数器计数之前会有一些初始延迟（LPTIMER使能之后）。因此，在使能LPTIMER之后，LPTIMER_IN0引脚上的外部时钟信号前5个有效边沿将会丢失。

图 18-3. 内部时钟模式 1（CKSSEL = 0，CNTMEN = 1，PSC[2:0] = 000）



18.4.2. LPTIMER 使能

LPTIMER_CTL1 寄存器的 LPTEN 位用于使能 LPTIMER 的内核逻辑模块。在 LPTEN 位置 1 后，实际使能 LPTIMER 之前需要延迟两个 LPTIMER_CK 时钟周期。

只有在 LPTIEMR 禁能时，才能修改 LPTIMER_CTL0 和 LPTIMER_INTEN 寄存器（INHLCOIE 位和 HLCMVUPIF 位除外）。

18.4.3. 预分频器

预分频器可以将 LPTIMER 的时钟 LPTIMER_CK 除以 2 的乘幂分频为计数器时钟 PSC_CLK。只有在 LPTIEMR 禁能（LPTEN=0）时，才能修改 PSC[2:0]位域。下表列出了所有的分频系数：

表 18-1. 预分频器的分频系数

预分频系数	PSC[2:0]位域
1/1	000
1/2	001
1/4	010
1/8	011
1/16	100
1/32	101
1/64	110
1/128	111

18.4.4. 输入滤波

LPTIMER_INx 引脚上的外部（映射到 GPIO）或内部（映射到片上外设，如比较器）信号需要通过数字滤波器进行滤波，以防止毛刺和噪声干扰在 LPTIMER 中扩散，这可以有效防止误计数和误触发。

在使用数字滤波器之前，需要先给 LPTIMER 提供内部时钟源，这样可以确保滤波器的正确运行。

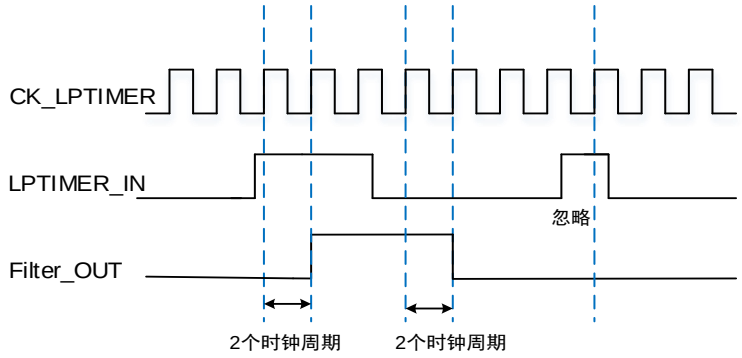
数字滤波器有两种类型：

- 第1种：用于保护LPTIMER的外部输入（LPTIMER_IN0/ LPTIMER_IN1），数字滤波器由ECKFLT [1:0]位进行配置；
- 第2种：用于保护LPTIMER的触发输入（ETIx），数字滤波器由TFLT [1:0]位进行配置；

注意：相同类型的数字滤波器应该保持相同的配置。

数字滤波器的灵敏度取决于 LPTIMER 输入引脚上连续相同采样的数量，并将信号电平变化视为有效。*图 18-4. 输入滤波时序图(ECKFLT=2'b01)*显示了输入滤波器 2 次连续采样的示例。

图 18-4. 输入滤波时序图（ECKFLT=2'b01）



注意：如果没有内部时钟信号，则必须禁能数字滤波器（设置 ECKFLT=0，TFLT=0）。这种情况下，可以使用外部模拟滤波器来保证 LPTIMER 的外部输入不受干扰。

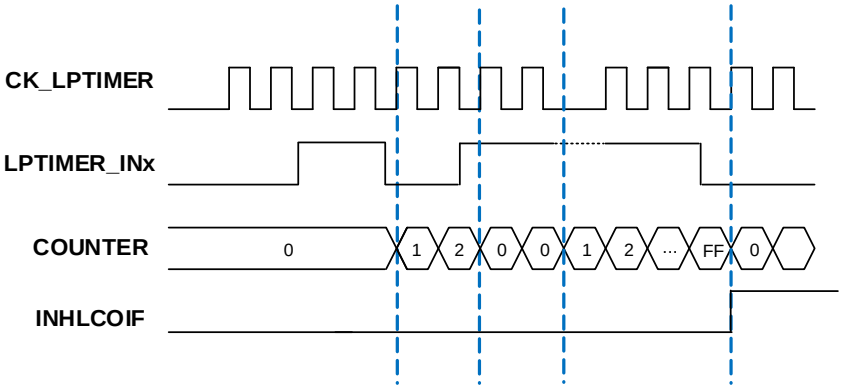
18.4.5. 外部输入高电平计数器

将 INHLCEN 位置 1 可以使能外部输入 LPTIMER_INx 的高电平计数器功能，高电平计数器的时钟由内部时钟 CK_LPTIMER 提供。当 LPTIMER_INx 引脚上出现高电平时，计数器开始计数，一旦出现低电平，计数器清零。

当高电平计数器的计数值等于 INHLCMVAL 位域（在 LPTIMER_INHLCMV 寄存器中）定义的数值时，LPTIMER_INTF 寄存器中的 INHLCOIF 位由硬件置位。若使能了 LPTIMER_INTEN 寄存器中的 INHLCOIE 位，则会产生中断。可以通过向 INTC 寄存器中的 INHLCOIC 位写 1 来清除 INHLCOIF 中断标志位。

*图 18-5. 外部输入高电平计数器*给出了外部输入高电平计数器的示例。

图 18-5. 外部输入高电平计数器



APB 总线和 CK_LPTIMER 使用不同的时钟，因此，APB 写操作和 LPTIMER_INHLCMV 寄存器实际使用这些值的时间存在一些延迟。在此延迟时间内，应当避免对该寄存器的任何其他写操作。

LPTIMER_INTF 寄存器中的 HLCMVUPIF 位用于说明对 LPTIMER_INHLCMV 寄存器的写操作何时完成。

18.4.6. 计数器启动

LPTIMER 的计数器可以通过软件触发或通过检测 8 个触发输入上的有效边沿来启动。ETMEN[1:0]位域用于配置 LPTIMER 的触发模式：

- ETMEN[1:0] = 2'b 00：一旦软件置位CTNMST位或SMST位，LPTIMER的计数器就启动了；
- ETMEN[1:0] ≠ 2'b00：ETSEL [2:0]位用于选择8个触发输入中的一个来启动LPTIMER。ETMEN[1:0]位域的其余3个非零值用于配置触发输入所使用的有效边沿。一旦检测到有效边沿，LPTIMER计数器就会启动。

外部触发可视为 LPTIMER 的异步信号，因此，一旦检测到触发信号，为了实现同步，在 LPTIMER 开始运行之前需要延迟 2 个计数器时钟周期。如果在 LPTIMER 启动后发生新的触发事件，该触发事件将会被忽略（除非使能了超时模式）。

注意：在置位 SMST 位和 CTNMST 位之前，必须将 LPTEN 位置位。当 LPTEN=0 时，对这些位的任何写操作都将被硬件丢弃。

18.4.7. 外部触发映射

LPTIMER 外部触发连接的情况如下*表 18-2. 外部触发映射*所示：

表 18-2. 外部触发映射

ETSEL[2:0]	外部触发映射
ETI0	GPIO
ETI1	RTC闹钟0
ETI2	RTC闹钟1
ETI3	RTC_TAMP0
ETI4	RTC_TAMP1
ETI5	RTC_TAMP2
ETI6	CMP0_OUT
ETI7	CMP1_OUT

18.4.8. 计数器运行模式

LPTIMER计数器运行在两种模式下：

- 连续计数模式：LPTIMER计数器由触发事件启动（软件触发或外部触发）后连续运行，直到LPTIMER禁能后才会停止；
- 单次计数模式：LPTIMER计数器由触发事件启动（软件触发或外部触发），在计数到

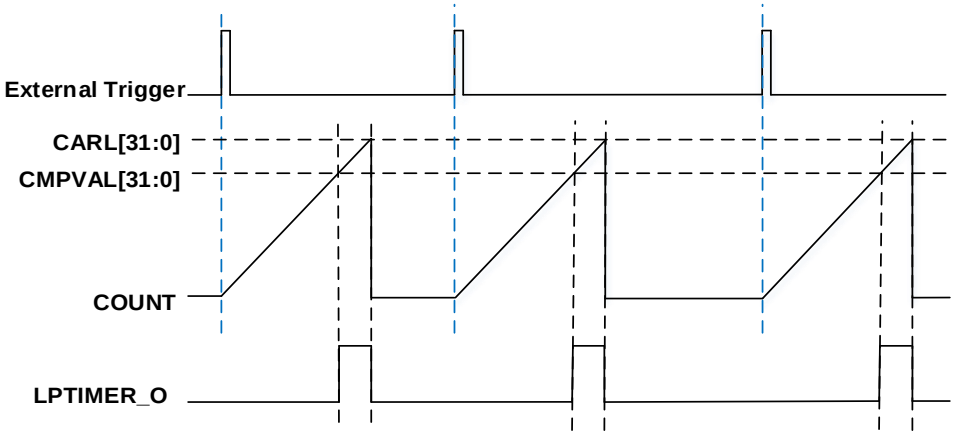
CARL[31:0]位域（在LPTIMER_CAR寄存器中）定义的值后停止；

单次计数模式

将 LPTIMER_CTL0 寄存器中的 SMST 位置 1，可以使能 LPTIMER 计数器的单次计数模式。该模式下，一次新的触发事件将重新启动 LPTIMER 计数器。在计数器启动之后且计数器达到 CARL[31:0]位域定义的值之前发生的任何触发事件都将被忽略。

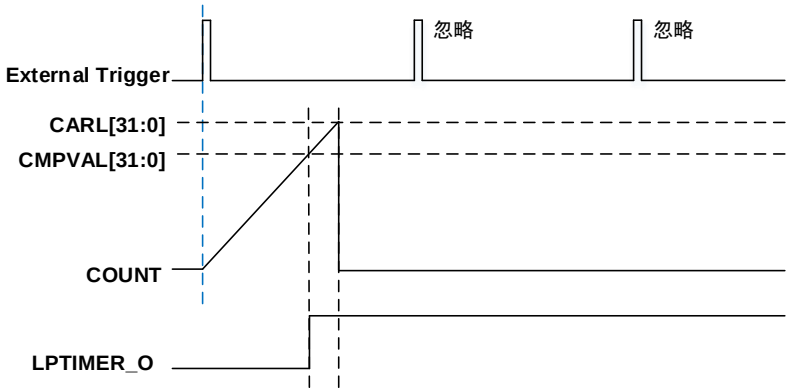
如果选择了外部触发来启动 LPTIMER 计数器，则当 SMST 位置 1 时，在计数器停止计数后（CNT[31:0]位域的值为 0）到达的每一个外部触发事件都将启动计数器进行新的计数周期计数，具体如*图 18-6. LPTIMER 输出（SMST = 1）*所示。

图 18-6. LPTIMER 输出（SMST = 1）



将 LPTIMER_CTL0 寄存器中的 OMSEL 位置 1，可以使能 LPTIMER 的置位模式。该模式下，LPTIMER 的计数器仅在第一次触发后启动，之后的所有触发事件都将被忽略，具体如*图 18-7. LPTIMER 输出（OMSEL = 1）*所示。

图 18-7. LPTIMER 输出（OMSEL = 1）



当 ETMEN [1:0] = 2'b 00 时，软件触发使能，将 SMST 位置 1，LPTIMER 以单次计数模式启动。

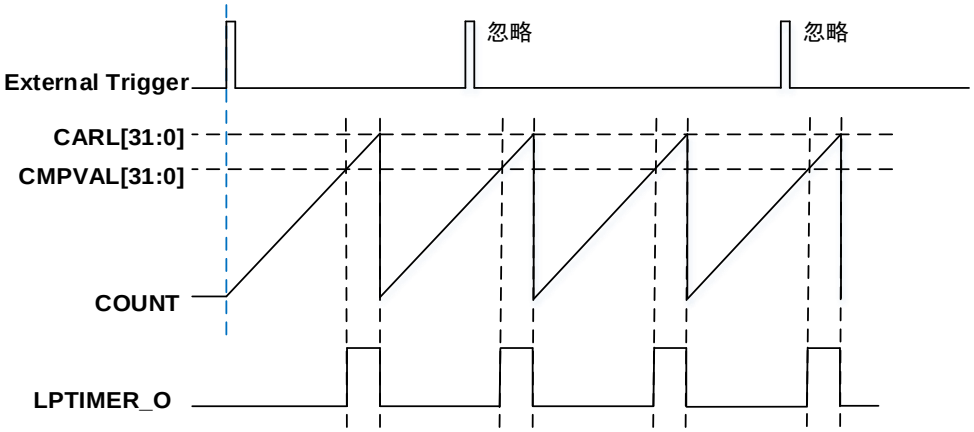
连续计数模式

将 LPTIMER_CTL0 寄存器中的 CTNMST 位置 1，可以使能 LPTIMER 计数器的连续计数模式。

如果选择了外部触发来启动 LPTIMER 计数器，则在 CTNMST 位置 1 之后到达的外部触发事件将启动计数器的连续计数模式，在 LPTIMER 启动后到达的任何触发事件都将被忽略，具体如*图 18-8. LPTIMER 输出（CTNMST = 1）*所示。

当 ETMEN [1:0] = 2'b 00 时，软件触发使能，将 CTNMST 位置 1，LPTIMER 以连续计数模式启动。

图 18-8. LPTIMER 输出（CTNMST = 1）



只有当 LPTIMER 使能（LPTEN=1）时，才能修改 SMST 位和 CTNMST 位，单次计数模式和连续计数模式可以实时进行修改。

若 LPTIMER 工作在连续计数模式，将 SMST 位置 1 后将切换到单次计数模式。计数器计数到 CARL[31:0]位域定义的值之后将停止计数。

若 LPTIMER 工作在单次计数模式，将 CTNMST 位置 1 后将切换到连续计数模式。计数器计数到 CARL[31:0]位域定义的值之后将重新开始计数。

18.4.9. 输出模式

通过配置 LPTIMER_CARL 寄存器和 LPTIMER_CMPV 寄存器，LPTIMER 可以输出几种不同的波形。

LPTIMER 可以输出以下 3 种波形：

- PWM模式：当LPTIMER_CNT的值和LPTIMER_CMPV寄存器的值匹配时，LPTIMER输出置位。当LPTIMER_CNT的值和LPTIMER_CAR寄存器的值匹配时，LPTIMER输出复位；
- 单脉冲模式：输出波形与PWM模式的第一个脉冲相同，之后始终输出复位。
- 置位模式：输出波形与单脉冲模式类似，输出保持为信号的最后电平（具体由LPTIMER_CTL0寄存器中的OPSEL位确定）。

这三种输出模式都要求 LPTIMER_CAR 寄存器的值大于 LPTIMx_CMPV 寄存器的值。

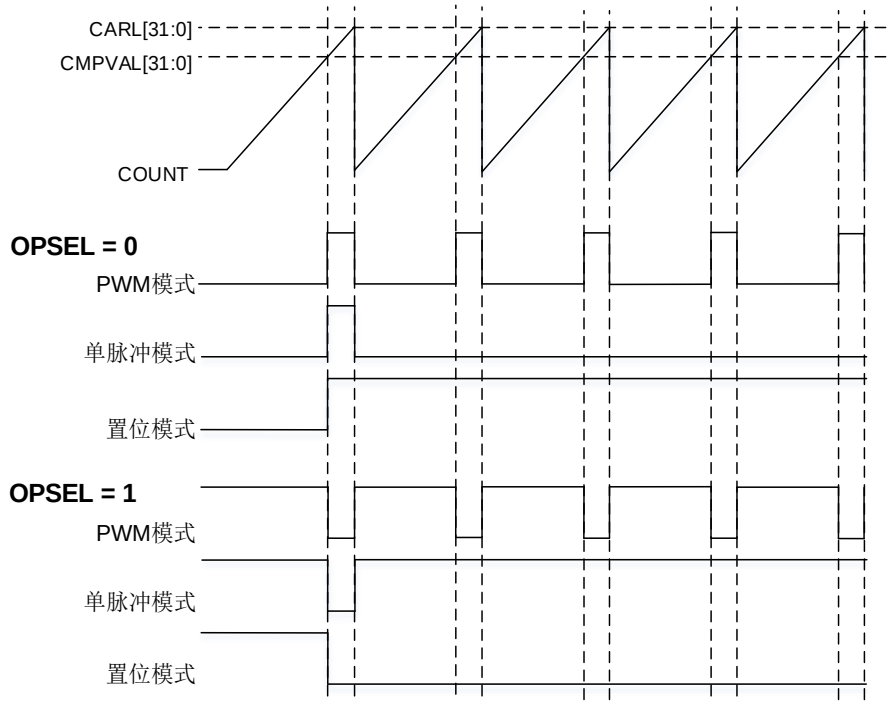
LPTIMER_CTL0 寄存器中的 OMSEL 位用于选择这三种输出模式。

- OMSEL = 0：LPTIMER输出为PWM模式或单脉冲模式（具体由CTNMST位或SPMST位配置）；
- OMSEL = 1：LPTIMER输出置位模式。

OPSEL 位用于配置 LPTIMER 的输出极性，修改该位立即生效。因此，在使能 LPTIMER 之前，只要修改极性配置位，输出默认值就会立即改变。

LPTIMER 输出波形的最大频率为 LPTIMER 时钟频率的二分之一，*图 18-9. LPTIMER_O 输出（OPSEL=0/1）*所示为 LPTIMER 输出的三种波形，和 OPSEL 位的值对输出波形极性的影响。

图 18-9. LPTIMER_O 输出（OPSEL=0/1）



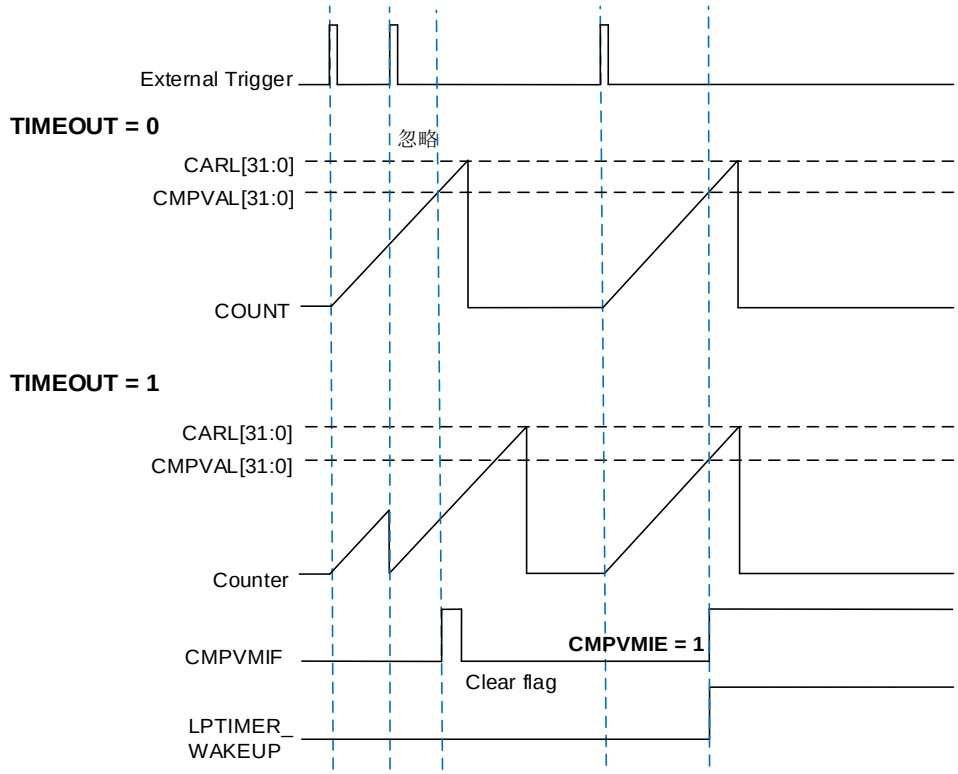
18.4.10. 超时模式

将 TIMEOUT 位置 1，可以通过在选定触发输入上检测到的有效边沿来复位 LPTIMER。第 1 个触发事件将用于启动 LPTIMER，之后的触发事件将复位和重新启动 LPTIMER。

LPTIMER 可以实现低功耗的超时模式，超时值可由 LPTIMx_CMPV 寄存器的值进行配置。

在配置好的比较寄存器值范围内若没有触发发生，当计数器计数到比较寄存器值时，将产生比较匹配中断唤醒 MCU。

图 18-10. LPTIMER 超时模式



18.4.11. 编码器模式

LPTIMER由两种编码器模式：

- 编码器模式0：LPTIMER_IN0和LPTIMER_IN1输入正交信号，当DECMEN=1且DECMSEL=0使能该模式；
- 编码器模式1：LPTIMER_IN0和LPTIMER_IN1输入非交信号，当DECMEN=1且DECMSEL=1使能该模式。

编码器模式 0

编码器模式 0 用于 LPTIMER_IN0 和 LPTIMER_IN1 输入为正交信号的情况，两个正交信号相互作用产生计数。

首先，将 CTNMST 位置 1 使能连续计数模式，同时 DECMEN 位置 1 使能编码器模式。设置 DECMSEL = 0 选择编码器模式 0，然后设置 CKPSEL [1:0] = 2b’00、2b’01 或 2b’10 来选择计数器上升沿、下降沿或双边沿计数。

在 IN0F（LPTIMER_IN0 滤波信号）信号和 IN1F（LPTIMER_IN1 滤波信号）信号电平变化期间，硬件会自动改变计数方向。计数方向变化机制如*表 18-3. 计数方向与编码器信号之间的关系*所示。编码器可以看作是一个带有方向选择的外部时钟，这意味着计数器会在 0 和自动重载值之间连续的计数。

因此，必须在计数器计数之前配置 TIMERx_CAR 寄存器。

当计数器计数方向改变时，相应的标志位会置 1。当计数器从向上计数改为向下计数时，DOWNIF 位置 1；当计数器从向下计数改为向上计数时，UPIF 位置 1。当寄存器中的 DOWNIE

= 1 或 UPIE = 1 时，相应的中断产生。

表 18-3. 计数方向与编码器信号之间的关系

计数模式（CKPSEL [1:0]）	电平	IN0F		IN1F	
		上升沿	下降沿	上升沿	下降沿
上升沿计数	IN0F = 1	x	x	向上	-
	IN0F = 0	x	x	向下	-
	IN1F = 1	向下	-	x	x
	IN1F = 0	向上	-	x	x
下降沿计数	IN0F = 1	x	x	-	向下
	IN0F= 0	x	x	-	向上
	IN1F = 1	-	向上	x	x
	IN1F = 0	-	向下	x	x
双边沿计数	IN0F = 1	x	x	向上	向下
	IN0F = 0	x	x	向下	向上
	IN1F = 1	向下	向上	x	x
	IN1F = 0	向上	向下	x	x

注意： “-”是不计数；“x”是不可能。

*图18-11. 计数器运行在编码器模式0（上升沿计数）*和*图18-12. 计数器运行在编码器模式0（下降沿计数）*分别给出了上升沿计数和下降沿计数的示例。

图 18-11. 计数器运行在编码器模式 0（上升沿计数）

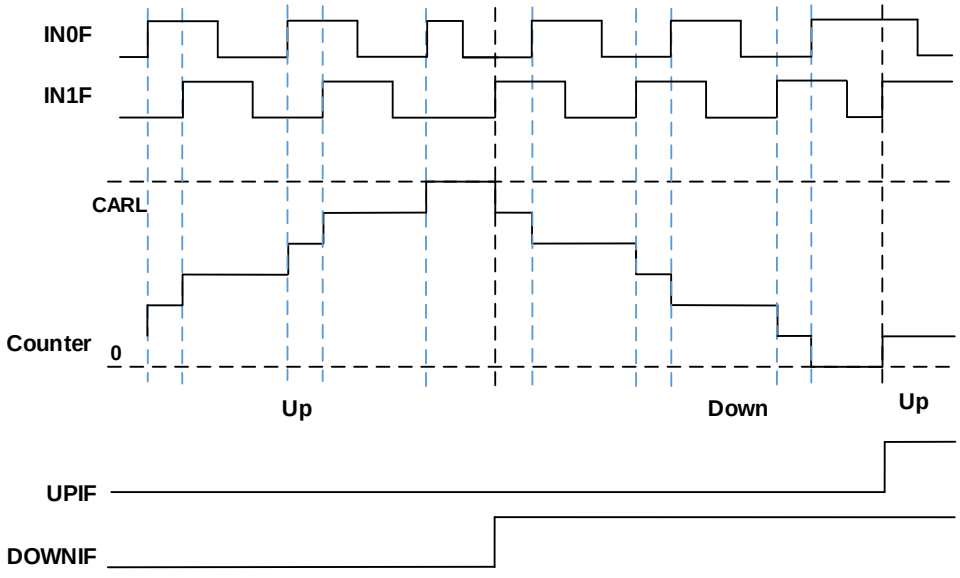
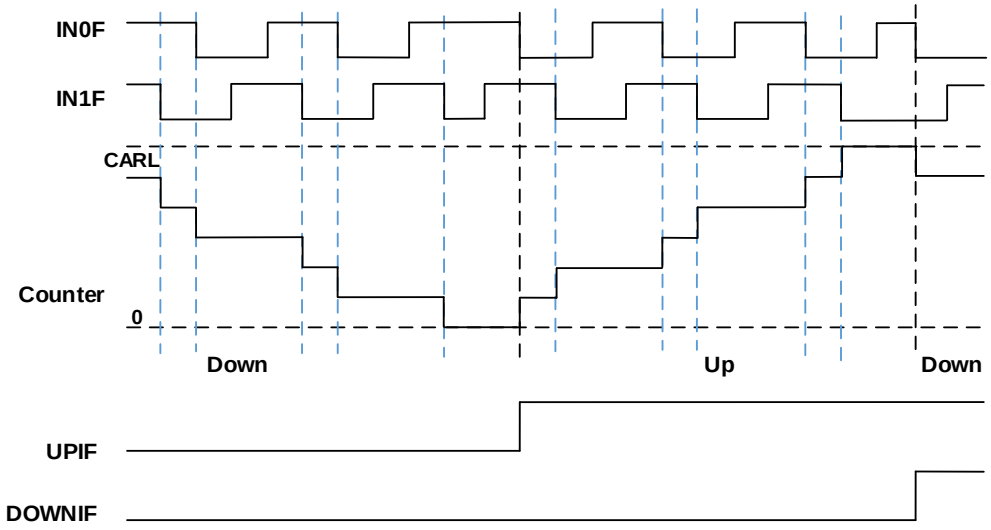


图 18-12. 计数器运行在编码器模式 0（下降沿计数）



编码器模式 1

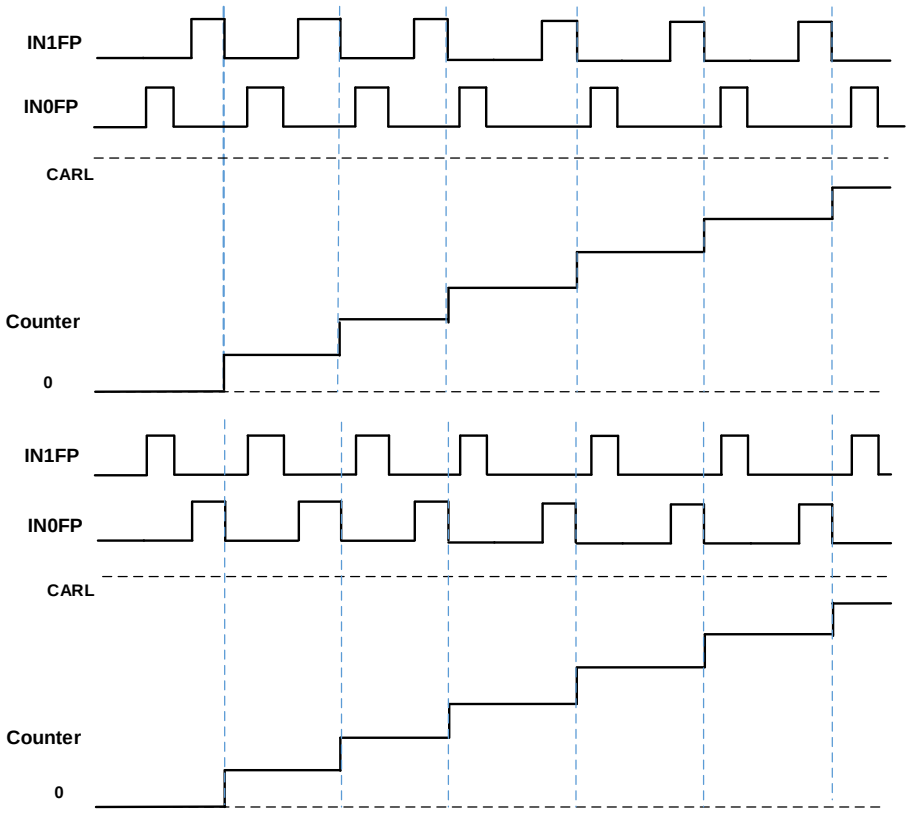
编码器模式 0 用于 LPTIMER_IN0 和 LPTIMER_IN1 输入为非交信号的情况，两个非交信号相互作用产生计数。

首先，将 CTNMST 位置 1 使能连续计数模式，同时 DECMEN 位置 1 使能编码器模式。设置 DECMSEL = 1 选择编码器模式 1，然后设置 CKPSEL [1:0] = 2b'00、2b'01 来选择 LPTIMER_IN0 和 LPTIMER_IN1 输入同相或反相。

当 IN0F 信号和 IN1F 信号依次出现两个不重叠脉冲时，计数器增计数一次。*图 18-13. 计数器运行在编码器模式 1（同相）*所示在编码器模式 1 下正确计数的信号波形时序图。IN0F 信号和 IN1F 信号的高电平不重叠。

编码器可以看作是一个带有方向选择的外部时钟，这意味着计数器会在 0 和自动重载值之间连续的计数。因此，必须在计数器计数之前配置 TIMERx_CAR 寄存器。

图 18-13. 计数器运行在编码器模式 1（同相）



当 IN0F 信号和 IN1F 信号波形不满足*图 18-13. 计数器运行在编码器模式 1（同相）*所示的时序关系时，计数器不能计数。根据两个信号输入波形的情况，相应的标志位（IN1EIF，IN0EIF，INRFOEIF，INHLOEIF）将置位，若 LPTIMER_INTEN 寄存器中的 IN1EIE 位、IN0EIE 位、INRFOEIE 位和 INHLOEIE 位置 1，则将产生相应的中断。

图 18-14. 计数器运行在编码器模式 1（同相，IN1EIF）

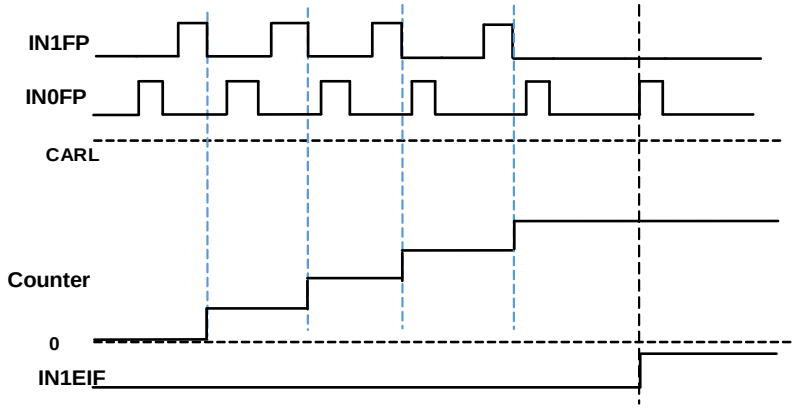


图 18-15. 计数器运行在编码器模式 1（同相，IN0EIF）

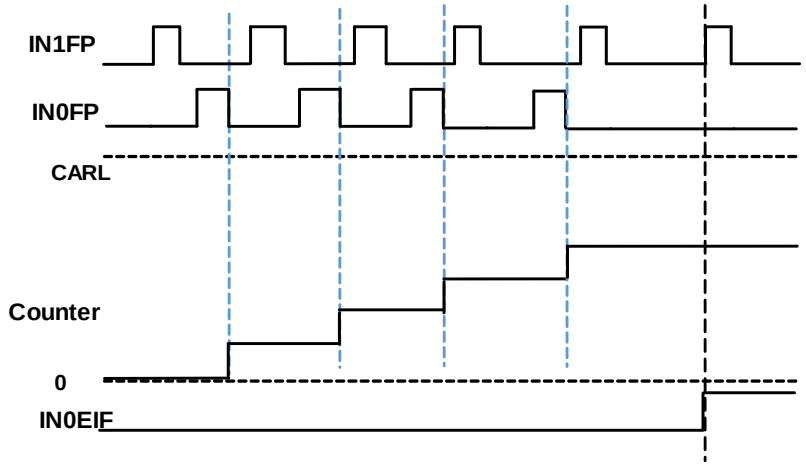


图 18-16. 计数器运行在编码器模式 1（同相，INRFOEIF）

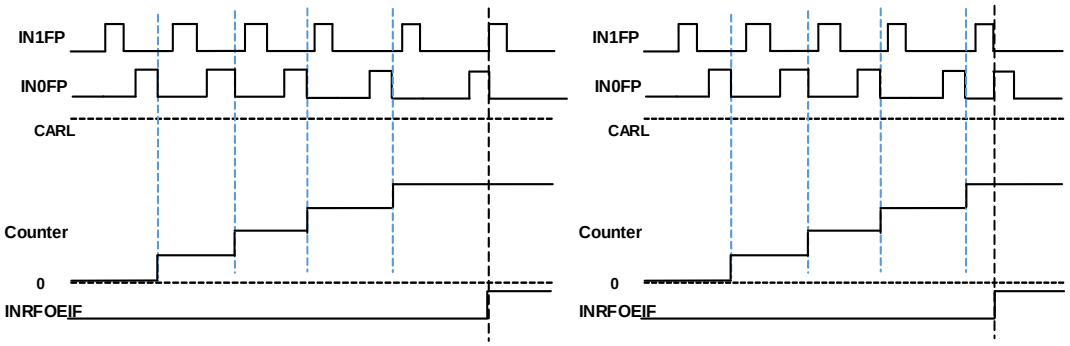
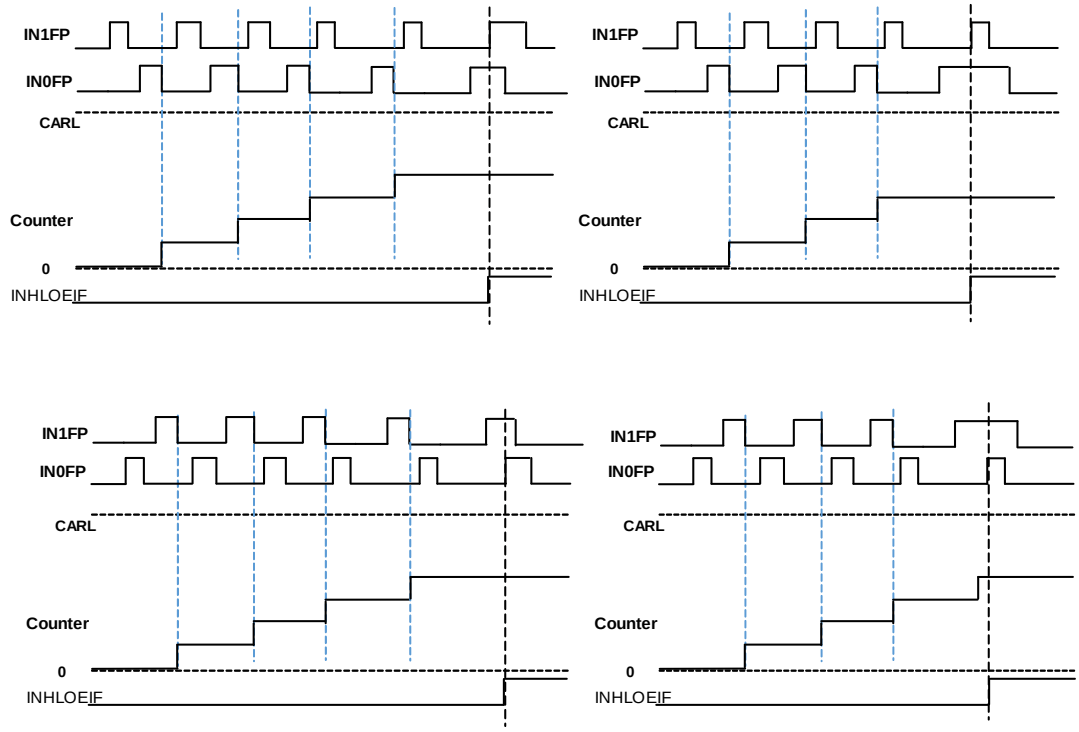


图 18-17. 计数器运行在编码器模式 1（同相，INHLOEIF）



注意，LPTIMER 使用编码器模式时，还应提供一个内部时钟信号（CKSSEL = 0），且该时钟

信号不能进行预分频（PSC [2:0] = 000）。在这种情况下，内部时钟信号的频率至少是外部时钟信号频率的四倍。

18.4.12. 寄存器更新操作

LPTIMER_CAR 寄存器和 LPTIMER_CMPV 寄存器在 APB 总线完成写操作之后立即更新，或当 LPTIMER 启动后在当前周期完成后更新。SHWEN 位用于配置 LPTIMER_CAR 寄存器和 LPTIMER_CMPV 寄存器的更新情况：

- SHWEN = 0：在APB总线写操作后，LPTIMER_CAR寄存器和LPTIMER_CMPV寄存器立即更新；
- SHWEN = 1：当LPTIMER启动之后，LPTIMER_CAR寄存器和LPTIMER_CMPV寄存器在当前周期完成后更新。

APB 总线和 CK_LPTIMER 使用不同的时钟，因此，APB 写操作和 LPTIMER_CAR 寄存器和 LPTIMER_CMPV 寄存器实际使用这些值的时间存在一些延迟。在此延迟时间内，应当避免对这些寄存器的任何其他写操作。

LPTIMER_INTF 寄存器中的 CARUPIF 位和 CMPVUPIF 位分别用于说明 LPTIMER_CAR 寄存器和 LPTIMER_CMPV 寄存器的写操作何时完成。

在写 LPTIMER_CAR 寄存器和 LPTIMER_CMPV 寄存器之后，只有完成了当前的写操作，才能对同一寄存器进行新的写操作。在 CMPVUPIF 标志位和 CARUPIF 标志位分别置 1 前，任何连续的写操作将造成不可预测的结果。

18.4.13. 低功耗模式

LPTIMER 具有多种时钟源，可以在除待机模式（Standby mode）以外的所有功耗模式下保持运行。LPTIMER 可以将系统从低功耗模式唤醒，非常适合于以极低的功耗实现超时模式的场合。

表 18-4. LPTIMER 工作在低功耗模式

模式	描述
睡眠模式	正常运行，LPTIMER的中断会使设备退出睡眠模式
运行2模式	正常运行
睡眠2模式	正常运行，LPTIMER的中断会使设备退出低功耗睡眠模式
深度睡眠0/ 1模式	当LPTIMER由LXTAL或内部低速晶振提供时钟时，LPTIMER的中断会使设备退出深度睡眠0/ 1模式。
深度睡眠2模式	LPTIMER的中断会使设备退出深度睡眠模式2

18.4.14. 中断

若在 LPTIMER_INTEN 寄存器中使能了相应的位，下面的事件可以产生中断或唤醒事件：

- LPTIMER_IN1错误
- LPTIMER_IN0错误
- LPTIMER_IN0和LPTIMER_IN1下降沿和上升沿重叠错误

- LPTIMER_IN0和LPTIMER_IN1高电平重叠错误
- LPTIMER_INx（x=0,1）高电平计数器溢出
- 输入高电平计数最大值寄存器更新中断标志位
- LPTIMER计数器由向上计数改为向下计数
- LPTIMER计数器由向下计数改为向上计数
- 计数器自动重载寄存器更新
- 比较寄存器更新
- 外部触发边沿事件
- 计数器自动重载寄存器匹配
- 比较寄存器匹配

如果 LPTIMER_INTF 寄存器中的中断标志位在相应的中断使能位置 1（LPTIMER_INTEN 寄存器中）前置位了，则该中断无效。

表 18-5. LPTIMER 中断事件

中断事件	描述
LPTIMER_IN1错误	当LPTIMER_IN1信号不在LPTIMER_IN0信号的两个连续上升沿之间发生跳变时，该标志位置1（仅用于编码器模式1）。
LPTIMER_IN0错误	当LPTIMER_IN0信号不在LPTIMER_IN1信号的两个连续上升沿之间发生跳变时，该标志位置1（仅用于编码器模式1）。
LPTIMER_IN0和LPTIMER_IN1下降沿和上升沿重叠	当LPTIMER_IN0下降沿和LPTIMER_IN1上升沿同时发生或者LPTIMER_IN0上升沿和LPTIMER_IN1下降沿同时发生时，该标志位置1（仅用于编码器模式1）。
LPTIMER_IN0和LPTIMER_IN1高电平重叠	当LPTIMER_IN0和LPTIMER_IN1的高电平重叠时，该标志位置1（仅用于编码器模式1）。
LPTIMER_INx（x=0,1）高电平计数器溢出	当LPTIMER_INx的高电平计数器与外部输入高电平计数最大值寄存器（LPTIMER_INHLCMV）值相等时，该标志位置1。
输入高电平计数最大值寄存器更新	当APB总线完成对LPTIMER_INHLCMV寄存器的写操作时，该标志位置1。
LPTIMER计数器由向上计数改为向下计数	在编码器模式中，当计数器由向上计数改为向下计数时，该标志位置1。
LPTIMER计数器由向下计数改为向上计数	在编码器模式中，当计数器由向下计数改为向上计数时，该标志位置1。
计数器自动重载寄存器更新	当APB总线完成对LPTIMER_CAR寄存器的写操作时，该标志位置1。
比较寄存器更新	当APB总线完成对LPTIMER_CMPV寄存器的写操作时，该标志位置1。
外部触发边沿事件	当外部触发的有效边沿发生时，该标志位置1。
计数器自动重载寄存器匹配	当LPTIMER_CNT的值与LPTIMER_CAR寄存器的值相等时，该标志位置1。
比较寄存器匹配	当LPTIMER_CNT的值与LPTIMER_CMPV寄存器的值相等时，该标志位置1。

18.4.15. LPTIMER 调试模式

当Cortex®-M23内核停止时，DBG_CTL1寄存器中的LPTIMER_HOLD配置位被置1，LPTIMER的计数器停止。

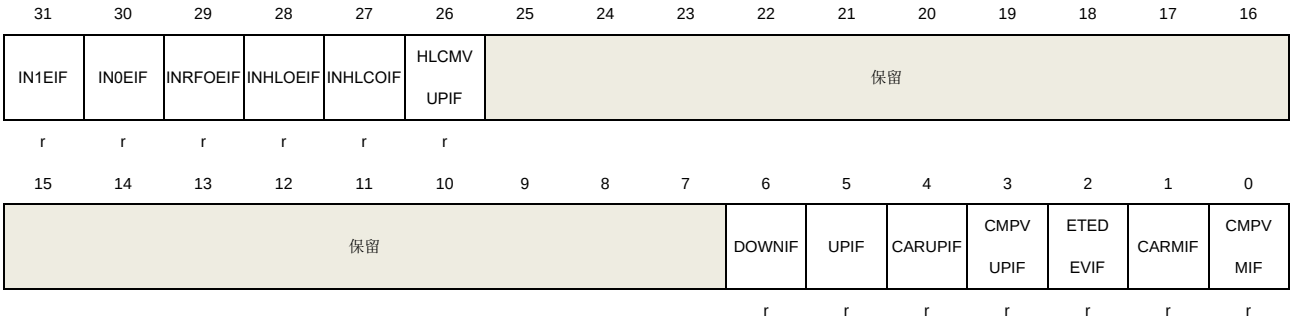
18.5. LPTIMER 寄存器

LPTIMER 基地址：0x4000 9400

18.5.1. 中断标志寄存器（LPTIMER_INTF）

地址偏移：0x00
复位值：0x0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31	IN1EIF	LPTIMER_IN1 错误中断标志位 当 LPTIMER_IN1 信号不在 LPTIMER_IN0 信号的两个连续上升沿之间发生跳变时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 IN1EIC 位写入 1 来清除 IN1EIF 标志。 **注意**：该标志位仅用于编码器模式 1。
30	IN0EIF	LPTIMER_IN0 错误中断标志位 当 LPTIMER_IN0 信号不在 LPTIMER_IN1 信号的两个连续上升沿之间发生跳变时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 IN0EIC 位写入 1 来清除 IN0EIF 标志。 **注意**：该标志位仅用于编码器模式 1。
29	INRFOEIF	LPTIMER_IN0 和 LPTIMER_IN1 下降沿和上升沿重叠错误中断标志位 当 LPTIMER_IN0 下降沿和 LPTIMER_IN1 上升沿同时发生或者 LPTIMER_IN0 上升沿和 LPTIMER_IN1 下降沿同时发生时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 INRFOEIC 位写入 1 来清除 INRFOEIF 标志。 **注意**：该标志位仅用于编码器模式 1。
28	INHLOEIF	LPTIMER_IN0 和 LPTIMER_IN1 高电平重叠错误中断标志位 当 LPTIMER_IN0 和 LPTIMER_IN1 的高电平重叠时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 INHLOEIC 位写入 1 来清除 INHLOEIF 标志。 **注意**：该标志位仅用于编码器模式 1。
27	INHLCOIF	LPTIMER_INx（x=0,1）高电平计数器溢出中断标志位 当 LPTIMER_INx 的高电平计数器与外部输入高电平计数最大值寄存器（LPTIMER_INHLCMV）值相等时，该标志位由硬件置 1。可以通过向 INTC 寄存器

		的 INHLCOIC 位写入 1 来清除 INHLCOIF 标志。
26	HLCMVUPIF	输入高电平计数最大值寄存器更新中断标志位 当 APB 总线完成对 LPTIMER_INHLCMV 寄存器的写操作时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 HLCMVUPIC 位写入 1 来清除 HLCMVUPIF 标志。
25:7	保留	必须保持复位值。
6	DOWNIF	LPTIMER计数器由向上计数改为向下计数中断标志位 在编码器 0 模式中，当计数器由向上计数改为向下计数时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 DOWNIC 位写入 1 来清除 DOWNIF 标志。
5	UPIF	LPTIMER计数器由向下计数改为向上计数中断标志位 在编码器0模式中，当计数器由向下计数改为向上计数时，该标志位由硬件置1。可以通过向INTC寄存器的UPIC位写入1来清除UPIF标志。
4	CARUPIF	计数器自动重载寄存器更新中断标志位 当 APB 总线完成对 LPTIMER_CAR 寄存器的写操作时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 CARUPIC 位写入 1 来清除 CARUPIF 标志。
3	CMPVUPIF	比较寄存器更新中断标志位 当 APB 总线完成对 LPTIMER_CMPV 寄存器的写操作时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 CMPVUPIC 位写入 1 来清除 CMPVUPIF 标志。
2	ETEDEVIF	外部触发边沿事件中断标志位 当外部触发的有效边沿发生时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 ETEDEVIC 位写入 1 来清除 ETEDEVIF 标志。 **注意**：当外部触发的有效边沿发生在 LPTIMER 启动之后，该标志位不会置位。
1	CARMIF	计数器自动重载寄存器匹配中断标志位 当 LPTIMER_CNT 的值与 LPTIMER_CAR 寄存器的值相等时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 CARMIC 位写入 1 来清除 CARMIF 标志。
0	CMPVMIF	比较寄存器匹配中断标志位 当 LPTIMER_CNT 的值与 LPTIMER_CMPV 寄存器的值相等时，该标志位由硬件置 1。可以通过向 INTC 寄存器的 CMPVMIC 位写入 1 来清除 CMPVMIF 标志。

18.5.2. 中断标志清除寄存器（LPTIMER_INTC）

地址偏移：0x04
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IN1EIC	IN0EIC	INRFOEIC	INHLOEIC	INHLCOIC	HLCMV UPIC	保留									
w	w	w	w	w	w										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									DOWNIC	UPIC	CARUPIC	CMPV	ETED	CARMIC	CMPV

				UPIC	EVIC		MIC
	w	w	w	w	w	w	w

位/位域	名称	描述
31	IN1EIC	LPTIMER_IN1 错误中断标志清除位 向该位写 1 来清除 IN1EIF 标志，写 0 无影响。
30	IN0EIC	LPTIMER_IN0 错误中断标志清除位 向该位写 1 来清除 IN0EIF 标志，写 0 无影响。
29	INRFOEIC	LPTIMER_IN0 和 LPTIMER_IN1 下降沿和上升沿重叠错误中断标志清除位 向该位写 1 来清除 INRFOEIF 标志，写 0 无影响。
28	INHLOEIC	LPTIMER_IN0 和 LPTIMER_IN1 高电平重叠错误中断标志清除位 向该位写 1 来清除 INHLOEIF 标志，写 0 无影响。
27	INHLCOIC	LPTIMER_INx（x=0,1）高电平计数器溢出中断标志清除位 向该位写 1 来清除 INHLCOIF 标志，写 0 无影响。
26	HLCMVUPIC	输入高电平计数最大值寄存器更新中断标志清除位 向该位写 1 来清除 HLCMVUPIF 标志，写 0 无影响。
25:7	保留	必须保持复位值。
6	DOWNIC	LPTIMER计数器由向上计数改为向下计数中断标志清除位 向该位写 1 来清除 DOWNIF 标志，写 0 无影响。
5	UPIC	LPTIMER计数器由向下计数改为向上计数中断标志清除位 向该位写 1 来清除 UPIF 标志，写 0 无影响。
4	CARUPIC	计数器自动重载寄存器更新中断标志清除位 向该位写 1 来清除 CARUPIF 标志，写 0 无影响。
3	CMPVUPIC	比较寄存器更新中断标志清除位 向该位写 1 来清除 CMPVUPIF 标志，写 0 无影响。
2	ETEDEVIC	外部触发边沿事件中断标志清除位 向该位写 1 来清除 ETEDEVIF 标志，写 0 无影响。
1	CARMIC	计数器自动重载寄存器匹配中断标志清除位 向该位写 1 来清除 CARMIF 标志，写 0 无影响。
0	CMPVMIC	比较寄存器匹配中断标志清除位 向该位写 1 来清除 CMPVMIF 标志，写 0 无影响。

18.5.3. 中断使能寄存器（LPTIMER_INTEN）

地址偏移：0x08
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IN1EIE	IN0EIE	INRFOEIE	INHLOEIE	INHLCOIE	HLCMVUPIE	保留									
rw	rw	rw	rw	rw	rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									DOWNIE	UPIE	CARUPIE	CMPVUPIE	ETEDEVIE	CARMIE	CMPVMIE
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31	IN1EIE	LPTIMER_IN1 错误中断使能位 0：禁止 LPTIMER_IN1 错误中断 1：使能 LPTIMER_IN1 错误中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
30	IN0EIE	LPTIMER_IN0 错误中断使能位 0：禁止 LPTIMER_IN0 错误中断 1：使能 LPTIMER_IN0 错误中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
29	INRFOEIE	LPTIMER_IN0 和 LPTIMER_IN1 下降沿和上升沿重叠错误中断使能位 0：禁止 LPTIMER_IN0 和 LPTIMER_IN1 下降沿和上升沿重叠错误中断 1：使能 LPTIMER_IN0 和 LPTIMER_IN1 下降沿和上升沿重叠错误中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
28	INHLOEIE	LPTIMER_IN0 和 LPTIMER_IN1 高电平重叠错误中断使能位 0：禁止 LPTIMER_IN0 和 LPTIMER_IN1 高电平重叠错误中断 1：使能 LPTIMER_IN0 和 LPTIMER_IN1 高电平重叠错误中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
27	INHLCOIE	LPTIMER_INx（x=0,1）高电平计数器溢出中断使能位 0：禁止 LPTIMER_INx（x=0,1）高电平计数器溢出中断 1：使能 LPTIMER_INx（x=0,1）高电平计数器溢出中断 只有在 LPTIMER 的外部输入高电平计数器禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 INHLCEN 位为 0）。
26	HLCMVUPIE	输入高电平计数最大值寄存器更新中断使能位 0：禁止输入高电平计数最大值寄存器更新中断 1：使能输入高电平计数最大值寄存器更新中断 只有在 LPTIMER 的外部输入高电平计数器禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 INHLCEN 位为 0）。
25:7	保留	必须保持复位值。

6	DOWNIE	LPTIMER计数器由向上计数改为向下计数中断使能位 0：禁止计数器由向上计数改为向下计数中断 1：使能计数器由向上计数改为向下计数中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
5	UPIE	LPTIMER计数器由向下计数改为向上计数中断使能位 0：禁止计数器由向下计数改为向上计数中断 1：使能计数器由向下计数改为向上计数中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
4	CARUPIE	计数器自动重载寄存器更新中断使能位 0：禁止计数器自动重载寄存器更新中断 1：使能计数器自动重载寄存器更新中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
3	CMPVUPIE	比较寄存器更新中断使能位 0：禁止比较寄存器更新中断 1：使能比较寄存器更新中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
2	ETEDEVIE	外部触发边沿事件中断使能位 0：禁止外部触发边沿事件中断 1：使能外部触发边沿事件中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
1	CARMIE	计数器自动重载寄存器匹配中断使能位 0：禁止计数器自动重载寄存器匹配中断 1：使能计数器自动重载寄存器匹配中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
0	CMPVMIE	比较寄存器匹配中断使能位 0：禁止比较寄存器匹配中断 1：使能比较寄存器匹配中断 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。

18.5.4. 控制寄存器 0（LPTIMER_CTL0）

地址偏移：0x0C
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						DECMSEL	DECMEN	CNTMEN	SHWEN	OPSEL	OMSEL	TIMEOUT	ETMEN[1:0]		保留
						rw	rw	rw	rw	rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETSEL[2:0]			保留	PSC[2:0]			保留	TFLT [1:0]		保留	ECKFLT[1:0]		CKPSEL[1:0]		CKSSEL
rw				rw				rw			rw		rw		rw

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	DECMSEL	编码器模式选择 0：编码器模式 0 1：编码器模式 1 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
24	DECMEN	编码器模式使能 0：编码器模式禁能 1：编码器模式使能 只有在LPTIMER禁能时，才能修改该位（LPTIMER_CTL1寄存器中的LPTEN位为0）。
23	CNTMEN	计数器模式选择 该位用于选择 LPTIMER 计数器的时钟源。 0：计数器在内部时钟每一个脉冲都计数 1：计数器在 LPTIMER_IN0 引脚上的每一个有效脉冲计数 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
22	SHWEN	LPTIMER_CAR 和 LPTIMER_CMPV 影子寄存器使能 0：影子寄存器禁能。在每一次 APB 写操作之后，这两个寄存器立即更新 1：影子寄存器使能。这两个寄存器在 LPTIME 周期结束之后更新 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
21	OPSEL	输出极性选择 该位用于控制 LPTIMER 输出的极性。 0：输出同相。向上计数时，当计数器值与 LPTIMER_CMPV 的值匹配，输出高电平；当计数器值与 LPTIMER_CAR 的值匹配，输出低电平。 1：输出反相。向上计数时，当计数器值与 LPTIMER_CMPV 的值匹配，输出低电平；当计数器值与 LPTIMER_CAR 的值匹配，输出高电平。 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
20	OMSEL	输出模式选择 该位用于控制 LPTIMER 的输出模式。 0：PWM 模式或单脉冲模式（CTNMST 位选择 PWM 模式，SMST 位选择单脉冲

		模式） 1：置位模式 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
19	TIMEOUT	超时模式使能 该位用于控制 LPTIMER 的超时模式。 0：LPTIMER 启动后，新的触发事件会被忽略 1：LPTIMER 启动后，新的触发事件会复位和重新启动 LPTIMER 只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
18:17	ETMEN[1:0]	外部触发模式使能 该位域用于配置 LPTIMER 的外部触发模式。 00：外部触发禁能（软件触发） 01：外部触发上升沿有效 10：外部触发下降沿有效 11：外部触发上升沿和下降沿都有效 只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
16	保留	必须保持复位值。
15:13	ETSEL[2:0]	外部触发选择 该位域用于选择 LPTIMER 的外部触发源 000：ETI0（GPIO） 001：ETI1（RTC 闹钟 0） 010：ETI2（RTC 闹钟 1） 011：ETI3（RTC_TAMP0） 100：ETI4（RTC_TAMP1） 101：ETI5（RTC_TAMP2） 110：ETI6（CMP0_OUT） 111：ETI7（CMP1_OUT） 只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
12	保留	必须保持复位值。
11:9	PSC[2:0]	时钟预分频器选择 该位域用于配置预分频器，将 LPTIMER 的时钟 LPTIMER_CK 分频到计数器时钟 PSC_CLK。 000：$f_{PSC\_CLK}=f_{LPTIMER\_CK}$ 001：$f_{PSC\_CLK}=f_{LPTIMER\_CK}/2$ 010：$f_{PSC\_CLK}=f_{LPTIMER\_CK}/4$ 011：$f_{PSC\_CLK}=f_{LPTIMER\_CK}/8$ 100：$f_{PSC\_CLK}=f_{LPTIMER\_CK}/16$ 101：$f_{PSC\_CLK}=f_{LPTIMER\_CK}/32$

110：$f_{PSC\_CLK}=f_{LPTIMER\_CK}$ / 64
111：$f_{PSC\_CLK}=f_{LPTIMER\_CK}$ / 128
只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。

8	保留	必须保持复位值。
7:6	TFLT[1:0]	触发滤波 该位域用于配置触发的数字滤波器，使用该功能时，必须使用内部时钟源。 00：无滤波器，触发信号的每个有效电平都有效 01：触发信号的有效电平变化必须保持 2 个时钟周期 10：触发信号的有效电平变化必须保持 4 个时钟周期 11：触发信号的有效电平变化必须保持 8 个时钟周期 只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
5	保留	必须保持复位值。
4:3	ECKFLT[1:0]	外部时钟滤波 该位域用于配置外部时钟的数字滤波器，使用该功能时，必须使用内部时钟源。 00：无滤波器，外部时钟的每个有效电平变化都有效 01：外部时钟的有效电平变化必须保持 2 个时钟周期 10：外部时钟的有效电平变化必须保持 4 个时钟周期 11：外部时钟的有效电平变化必须保持 8 个时钟周期 只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
2:1	CKPSEL[1:0]	时钟极性选择 当 LPTIMER 使用外部时钟源时，该位域用于配置计数器计数的有效边沿。 00：上升沿计数 若 LPTIMER 配置为编码器模式 0（DECMEN=1,DECMSEL=0），编码器的上升沿计数模式有效； 若 LPTIMER 配置为编码器模式 1（DECMEN=1,DECMSEL=1），LPTIMER_IN0 和 LPTIMER_IN1 的输入同相； 若 LPTIMER 外部输入高电平计数器使能（INHLCEN=1），LPTIMER_IN0 和 LPTIMER_IN1 的输入同相。 01：下降沿计数 若 LPTIMER 配置为编码器模式 0（DECMEN=1,DECMSEL=0），编码器的下降沿计数模式有效； 若 LPTIMER 配置为编码器模式 1（DECMEN=1,DECMSEL=1），LPTIMER_IN0 和 LPTIMER_IN1 的输入反相； 若 LPTIMER 外部输入高电平计数器使能（INHLCEN=1），LPTIMER_IN0 和 LPTIMER_IN1 的输入反相。 10：双边沿计数 当外部时钟的双边沿都能有效计数时，LPTIMER 必须使用内部时钟源，并且内部时钟源的频率最少是外部时钟源频率的 4 倍。 若 LPTIMER 配置为编码器模式 0（DECMEN=1，DECMSEL=0），编码器的双边

沿计数模式有效；
LPTIMER 不能配置为编码器模式 1；
若 LPTIMER 外部输入高电平计数器使能（INHLCEN=1），LPTIMER_IN0 和 LPTIMER_IN1 的输入同相。
11：保留
只有在 LPTIMER 禁能时，才能修改该位域（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。

0 CKSSEL 时钟源选择
该位用于选择 LPTIMER 的时钟源。
0：LPTIMER 使用内部时钟源
1：LPTIMER 使用外部时钟源（LPTIMER_IN0）
只有在 LPTIMER 禁能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 0）。
注意：当 DECMEN 位置 1 使能编码器模式时，CKSSEL 会自动清零。

18.5.5. 控制寄存器 1（LPTIMER_CTL1）

地址偏移：0x10
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INHLCEN	LPTENF	保留													
rw	r														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													CTNMST	SMST	LPTEN
													rw	rw	rw

位/位域	名称	描述
31	INHLCEN	LPTIMER 外部输入高电平计数器使能 0：禁能 1：使能
30	LPTENF	LPTIMER 从 LPTIMER 内核使能标志位 该位由硬件置位和清零。 0：LPTIMER 禁能 1：LPTIMER 使能
29:3	保留	必须保持复位值。
2	CTNMST	LPTIMER 以连续计数模式启动 该位由软件置位和硬件清零。 只有在 LPTIMER 使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 1）。
1	SMST	LPTIMER 以单次计数模式启动

该位由软件置位和硬件清零。
只有在 LPTIMER 使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 1）。

0	LPTEN	LPTIMER 使能 该位由软件置位和清零。 0：LPTIMER 禁能 1：LPTIMER 使能

18.5.6. 比较寄存器（LPTIMER_CMPV）

地址偏移：0x14
复位值：0x0000

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31:0	CMPVAL[31:0]	比较值 这些位定义了计数器的比较值。 只有在 LPTIMER 使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 1）。

18.5.7. 计数器自动重载寄存器（LPTIMER_CAR）

地址偏移：0x18
复位值：0x0001

该寄存器只能按字（32位）访问。

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

位/位域	名称	描述
31:0	CARL[31:0]	计数器自动重载寄存器值 这些位定义了计数器的自动重载值。

只有在 LPTIMER 使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 LPTEN 位为 1）。

18.5.8. 计数器寄存器（LPTIMER_CNT）

地址偏移：0x1C
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
r															

位/位域	名称	描述
31:0	CNT[31:0]	计数器值 **注意**：当 LPTIMER 使用异步时钟时，对 LPTIMER_CNT 寄存器的读操作可能会返回不可靠的值。因此，需要执行两个连续的读操作，并且确认两次的读取值是否相同。

18.5.9. 外部输入映射寄存器（LPTIMER_EIRMP）

地址偏移：0x20
复位值：0x0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														IN1_RMP	IN0_RMP
														rw	rw

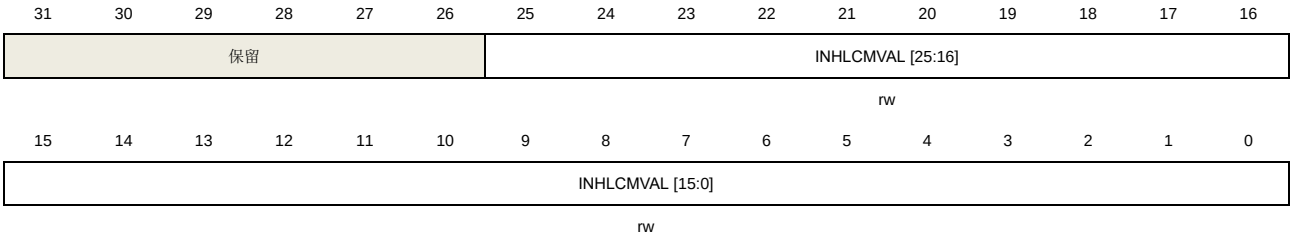
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	IN1_RMP	外部输入 LPTIMER_IN1 映射 0：外部输入引脚 1 映射到 GPIO 1：外部输入引脚 1 映射到 CMP1_OUT
0	IN0_RMP	外部输入 LPTIMER_IN0 映射 0：外部输入引脚 0 映射到 GPIO 1：外部输入引脚 0 映射到 CMP0_OUT

18.5.10. 输入高电平计数最大值寄存器（LPTIMER_INHLCMV）

地址偏移：0X24

复位值：0x0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:26	保留	必须保持复位值。
25:0	INHLCMVAL	输入高电平计数最大值 只有在 LPTIMER 的外部输入高电平计数器使能时，才能修改该位（LPTIMER_CTL1 寄存器中的 INHLCEN 位为 1）。

19. 通用同步异步收发器（USART）

19.1. 简介

通用同步/异步收发器（USART）提供了一个灵活方便的串行数据交换接口。数据帧可以通过全双工或半双工，同步或异步的方式进行传输。USART提供了可编程的波特率发生器，能对UCLK（PCLK，CK_SYS，LXTAL或IRC16M）时钟进行分频产生USART发送和接收所需的特定频率。

USART不仅支持标准的异步收发模式，还实现了一些其他类型的串行数据交换模式，如红外编码规范，SIR，智能卡协议，LIN，半双工以及同步模式。它还支持多处理器通信和硬件流控操作（CTS/RTS）。数据帧支持从LSB或者MSB开始传输。数据位的极性和TX/RX引脚都可以灵活配置。

所有USART都支持DMA功能，以实现高速率的数据通信。

19.2. 主要特征

- NRZ标准格式。
- 全双工异步通信。
- 半双工单线通信。
- 接收FIFO功能。
- 双时钟域：
  - 互为异步关系的PCLK和独立于PCLK时钟的USART时钟。
  - 不依赖UCLK设置的波特率设置。
- 可编程的波特率产生器，当时钟频率为64MHz，过采样为8，最高速度可达8MBits/s。
- 完全可编程的串口特性：
  - 数据位（8或9位）低位或高位在前。
  - 偶校验位，奇校验位，无校验位的生成或检测。
  - 产生0.5，1，1.5或者2个停止位。
- 可互换的Tx/Rx引脚。
- 可配置的数据极性。
- 支持硬件Modem流控操作（CTS/RTS）和RS485驱动使能。
- 可配置的多级缓存通信DMA访问数据缓冲区。
- 发送器和接收器可分别使能。
- 奇偶校验位控制：
  - 发送奇偶校验位。
  - 检测接收的数据字节的奇偶校验位。
- LIN断开帧的产生和检测。
- 支持红外数据协议（IrDA）。
- 同步传输模式以及为同步传输输出发送时钟。
- 支持兼容ISO7816-3的智能卡接口：
  - 字节模式（T=0）。

  - 块模式（T=1）。
  - 直接和反向转换。
- 多处理器通信：
  - 如果地址不匹配，则进入静默模式。
  - 通过线路空闲检测或者地址标记检测从静默模式唤醒。
- 支持ModBus通信：
  - 超时功能。
  - CR/LF字符识别。
- 从深度睡眠模式唤醒：
  - 通过标准的RBNE中断。
  - 通过WUF中断。
- 多种状态标志：
  - 传输检测标志：接收缓冲区不为空（RBNE），接收FIFO满（RFF），发送缓冲区为空（TBE），传输完成（TC）。
  - 错误检测标志：过载错误（ORERR），噪声错误（NERR），帧格式错误（FERR），奇偶校验错误（PERR）。
  - 硬件流控操作标志：CTS变化（CTSF）。
  - LIN模式标志：LIN断开检测（LBDF）。
  - 多处理器通信模式标志：IDLE帧检测（IDLEF）。
  - ModBus通信标志：地址/字符匹配（AMF），接收超时（RTF）。
  - 智能卡模式标志：块结束（EBF）和接收超时（RTF）。
  - 从深度睡眠模式唤醒标志。
  - 若相应的中断使能，这些事件发生将会触发中断。

USART0和USART1完全实现上述功能，但是UART3/UART4只实现了上面所介绍的部分功能，下面这些功能在UART3/UART4中没有实现：

- 智能卡模式
- IrDA SIR ENDEC模块
- LIN模式
- 双时钟域和从深度睡眠模式唤醒
- 接收超时中断
- ModBus通信
- 同步模式
- 硬件流控

19.3. 功能描述

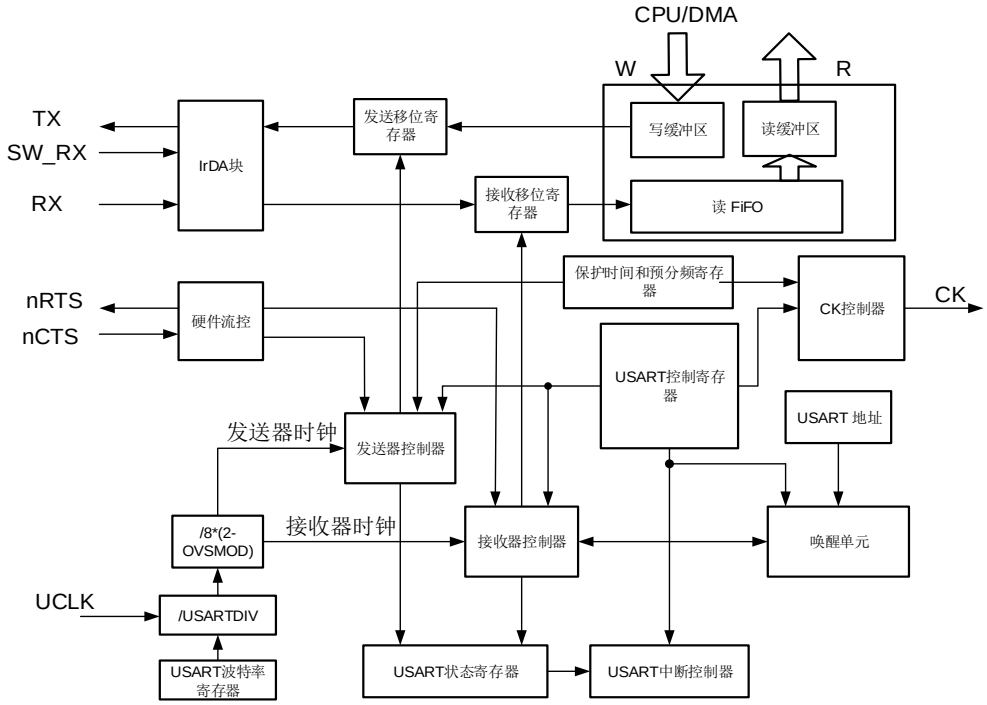
USART 接口通过*表 19-1. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

表 19-1. USART 重要引脚描述

引脚	类型	描述
RX	输入	接收数据
TX	输出 I/O（单线模式/智能卡模式）	发送数据。当 USART 使能后，若无数据发送，默认为高电平

引脚	类型	描述
CK	输出	用于同步通信的串行时钟信号
nCTS	输入	硬件流控模式发送使能信号
nRTS	输出	硬件流控模式发送请求信号

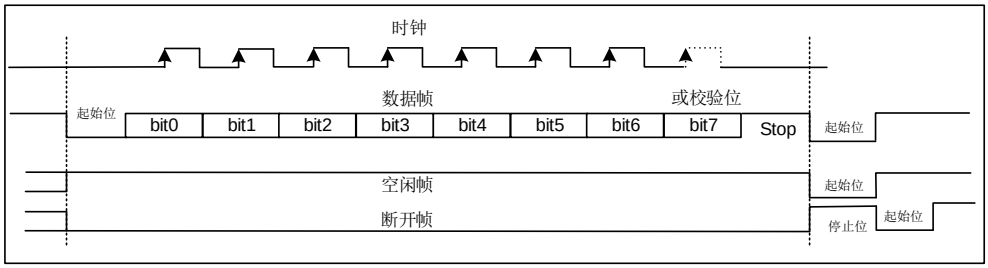
图 19-1. USART 模块内部框图



19.3.1. USART 帧格式

USART数据帧开始于起始位，结束于停止位。USART_CTL0寄存器中WL位可以设置数据长度。将USART_CTL0寄存器中PCEN置位，最后一个数据位可以用作校验位。若WL位为0，第七位为校验位。若WL位置1，第八位为校验位。USART_CTL0寄存器中PM位用于选择校验位的计算方法。

图19-2. USART字符帧（8数据位和1停止位）



在发送和接收中，停止位可以在USART_CTL1寄存器中STB[1:0]位域中配置。

表 19-2. 停止位配置

STB[1:0]	停止位长度（位）	功能描述
00	1	默认值

STB[1:0]	停止位长度（位）	功能描述
01	0.5	智能卡模式接收
10	2	标准 USART 和单线模式
11	1.5	智能卡模式发送和接收

在一个空闲帧中，所有位都为1。数据帧长度与正常USART数据帧长度相同。

紧随停止位后多个低电平为中断帧。USART数据帧的传输速度由UCLK时钟频率，波特率发生器的配置，以及过采样模式共同决定。

19.3.2. 波特率发生

波特率分频系数是一个16位的数字，包含12位整数部分和4位小数部分。波特率发生器使用这两部分组合所得的数值来确定波特率。由于具有小数部分的波特率分频系数，将使USART能够产生所有标准波特率。

波特率分频系数（USARTDIV）与UCLK具有如下关系：

如果过采样率是16，公式为：

$$\text{USARTDIV}=\frac{\text{UCLK}}{16\times\text{Baud Rate}} \quad (19\text{-}1)$$

如果过采样是8，公式为：

$$\text{USARTDIV}=\frac{\text{UCLK}}{8\times\text{Baud Rate}} \quad (19\text{-}2)$$

例如，当过采样是16：

1. 由USART_BAUD寄存器的值得到USARTDIV：
   假设USART_BAUD=0x21D，则INTDIV=33（0x21），FRADIV=13（0xD）。
   UASRTDIV=33+13/16=33.81。
2. 由USARTDIV得到USART_BAUD寄存器的值：
   假设要求UASRTDIV=30.37，INTDIV=30（0x1E）。
   16*0.37=5.92，接近整数6，所以FRADIV=6 (0x6)。
   USART_BAUD=0x1E6。

注意：若取整后FRADIV=16（溢出），则进位必须加到整数部分。

19.3.3. USART 发送器

如果USART_CTL0寄存器的发送使能位（TEN）被置位，当发送数据缓冲区不为空时，发送器将会通过TX引脚发送数据帧。TX引脚的极性可以通过USART_CTL1寄存器中TINV位来配置。时钟脉冲通过CK引脚输出。

TEN置位后发送器会发出一个空闲帧。TEN位在数据发送过程中是不可以被复位的。

系统上电后，TBE默认为高电平。在USART_STAT寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入USART_TDATA寄存器。当数据写入USART_TDATA寄存器，TBE位将

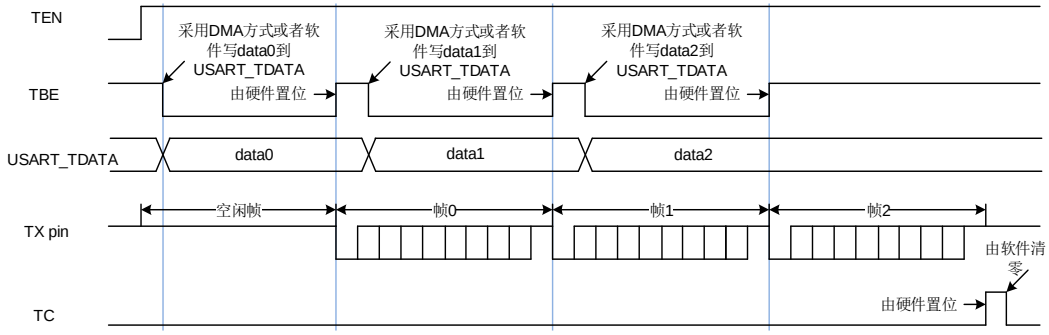
被清0。在数据由USART_TDATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入USART_TDATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入USART_TDATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄存器。

假如一帧数据已经被发送出去，并且TBE位已被置位，那么USART_STAT寄存器中TC位将被置1。如果USART_CTL0寄存器中的中断使能位（TCIE）为1，将会产生中断。

*图 19-3. USART 发送步骤*给出了 USART 发送步骤。软件操作按以下流程进行：

1. 通过USART_CTL0寄存器的WL设置字长；
2. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在USART_CTL2寄存器中使能DMA（DENT位）；
4. 在USART_BAUD寄存器中设置波特率；
5. 在USART_CTL0寄存器中置位UEN位，使能USART；
6. 在USART_CTL0寄存器中设置TEN位；
7. 等待TBE置位；
8. 向USART_TDATA寄存器写数据；
9. 若DMA未使能，每发送一个字节都需重复步骤7-8；
10. 等待TC=1，发送完成。

图 19-3. USART 发送步骤



在禁用USART或进入低功耗状态之前，必须等待TC置位。通过将USART_INTC寄存器的TCC位置1可以将TC位清零。

当SBKCMD置位时，会发送一个断开帧，发送完成后，SBKCMD将被清0。

19.3.4. USART 接收器

上电后，按以下步骤使能USART接收器：

1. 写USART_CTL0寄存器的WL位去设置字长；
2. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在USART_CTL2寄存器中使能DMA（DENR位）；
4. 在USART_BAUD寄存器中设置波特率；
5. 在USART_CTL0寄存器中置位UEN位，使能USART；
6. 在USART_CTL0中设置REN位。

接收器在使能后若检测到一个有效的起始脉冲便开始接收码流。在接收一个数据帧的过程中会

检测噪声错误，奇偶校验错误，帧错误和过载错误。

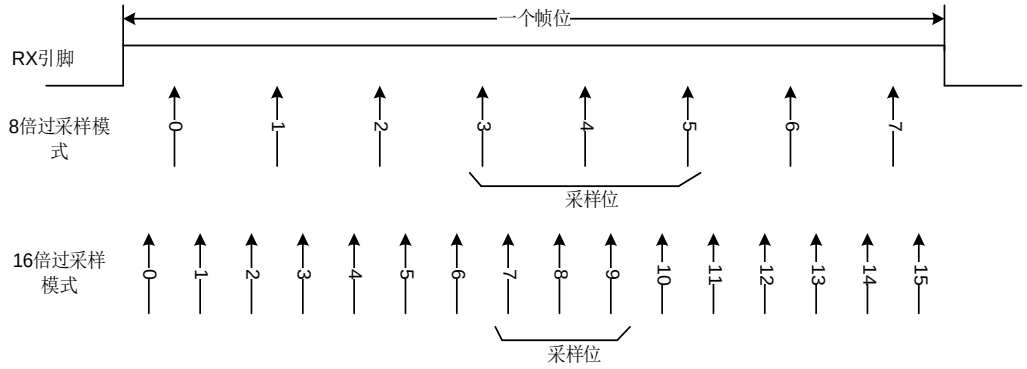
当接收到一个数据帧，USART_STAT寄存器中的RBNE置位，如果设置了USART_CTL0寄存器中相应的中断使能位RBNEIE，将会产生中断。在USART_STAT寄存器中可以观察接收状态标志。

软件可以通过读USART_RDATA寄存器或者DMA方式获取接收到的数据。不管是直接读寄存器还是通过DMA，只要是对USART_RDATA寄存器的一个读操作都可以清除RBNE位。

在接收过程中，需使能REN位，不然当前的数据帧将会丢失。

在默认情况下，接收器通过获取三个采样点的值来估计该位的值。如果是8倍过采样模式，选择第3、4、5个采样点；如果是16倍过采样模式，选择第7、8、9个采样点。如果在3个采样点中有2个或3个为0，该数据位被视为0，否则为1。如果3个采样点中有一个采样点的值与其他两个不同，不管是起始位，数据位，奇偶校验位或者停止位，都将产生噪声错误（NERR）。如果使能DMA，并置位USART_CTL2寄存器中ERRIE，将会产生中断。如果在USART_CTL2中置位OSB，接收器将仅获取一个采样点来估计一个数据位的值。在这种情况下将不会检测到噪声错误。

图 19-4. 过采样方式接收一个数据位（OSB=0）



通过置位USART_CTL0寄存器中的PCEN位使能奇偶校验功能，接收器在接收一个数据帧时计算预期奇偶校验值，并将其与接收到的奇偶校验位进行比较。如果不相等，USART_STAT寄存器中PERR被置位。如果置位了USART_CTL0寄存器中的PERRIE位，将产生中断。

如果在停止位传输过程中RX引脚为0，将产生帧错误，USART_STAT寄存器中FERR置位。如果使能DMA并置位USART_CTL2寄存器中ERRIE位，将产生中断。根据停止位的配置，有以下几种情形：

- 0.5个停止位：0.5个停止位时，停止位不采样
- 1个停止位：1个停止位时，在停止位的中间进行采样
- 1.5个停止位：1.5个停止位时，1.5个停止位可以分为两个部分：0.5个停止位的部分不采样和1个停止位的中间进行采样
- 2个停止位：2个停止位时，如果在第一个停止位期间检测到帧错误，帧错误标志置位，则第二个停止位不检测帧错误。如果第一个停止位期间没有检测到帧错误，则在第二个停止位继续检测帧错误。

当接收到一帧数据，而RBNE位还没有被清零，随后的数据帧将不会存储在数据接收缓冲区中。USART_STAT寄存器中的溢出错误标志位ORERR将置位。如果使能DMA并置位USART_CTL2寄存器中ERRIE位或者置位RBNEIE，将产生中断。

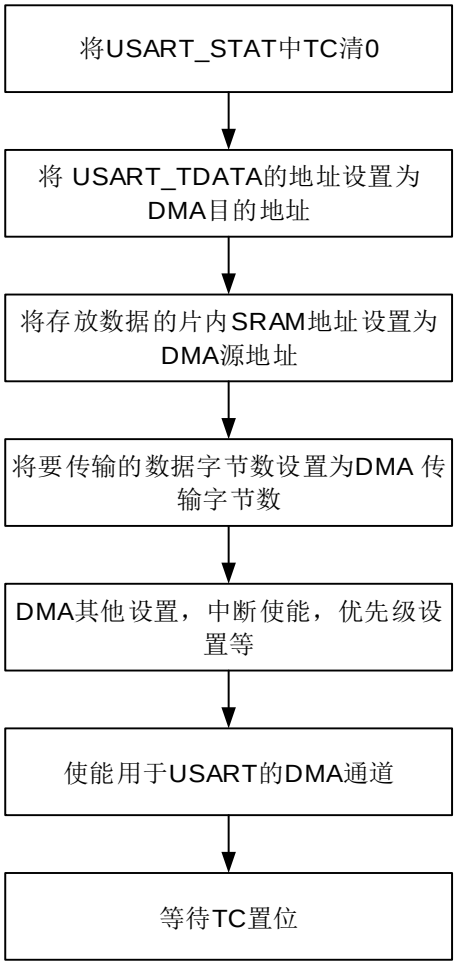
若接收过程中，产生了噪声错误（NERR）、校验错误（PERR）、帧错误（FERR）或溢出错误（ORERR），则NERR、PERR、FERR或ORERR将和RBNE同时置位。如果没有使能DMA，RBNE中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

19.3.5. DMA 方式访问数据缓冲区

为减轻处理器的负担，可以采用DMA访问发送缓冲区或者接收缓冲区。置位USART_CTL2寄存器中DENT位可以使能DMA发送，置位USART_CTL2寄存器中DENR位可以使能DMA接收。

当 DMA 用于 USART 发送时，DMA 将数据从片内 SRAM 传送到 USART 的数据缓冲区。配置步骤如*图 19-5. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

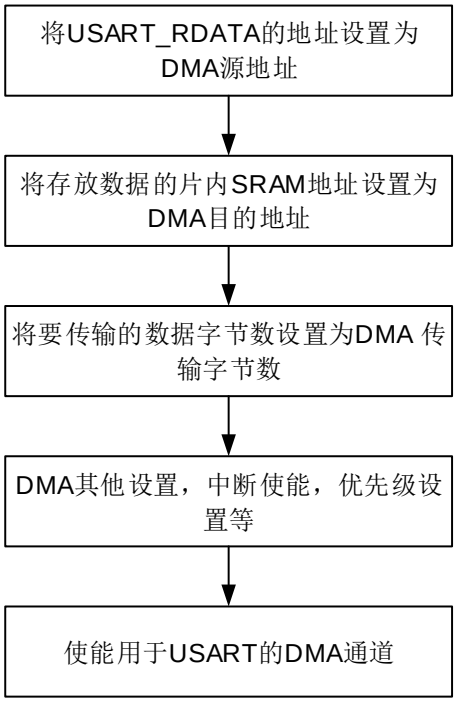
图 19-5. 采用 DMA 方式实现 USART 数据发送配置步骤



所有数据帧都传输完成后，USART_STAT寄存器中TC位置1。如果USART_CTL0寄存器中TCIE置位，将产生中断。

当 DMA 用于 USART 接收时，DMA 将数据从接收缓冲区传送到片内 SRAM。配置步骤如*图 19-6. 采用 DMA 方式实现 USART 数据接收配置步骤*所示。如果将 USART_CTL2 寄存器中 ERRIE 位置 1，USART_STAT 寄存器中的错误标志位（FERR、ORERR 和 NERR）置位时将产生中断。

图 19-6. 采用 DMA 方式实现 USART 数据接收配置步骤

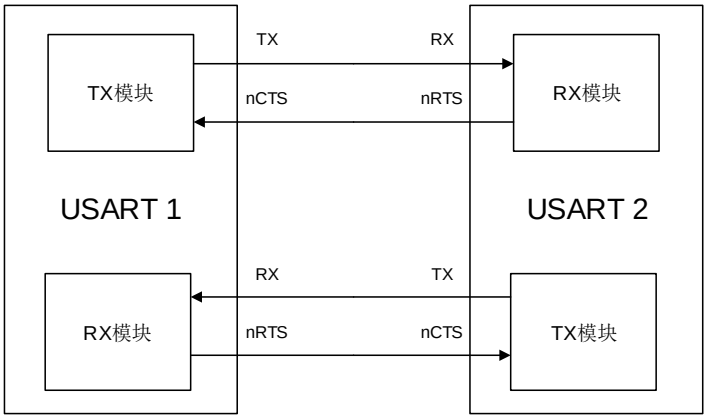


当USART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

19.3.6. 硬件流控制

硬件流控制功能通过nCTS和nRTS引脚来实现。通过将USART_CTL2寄存器中RTSEN位置1来使能RTS流控，将USART_CTL2寄存器中CTSEN位置1来使能CTS流控。

图 19-7. 两个 USART 之间的硬件流控制



RTS 流控

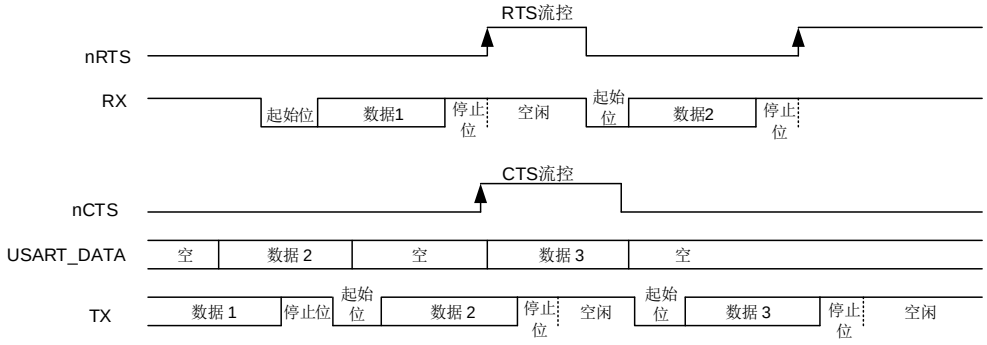
USART接收器输出nRTS，它用于反映接收缓冲区状态。当一帧数据接收完成，nRTS变成高电平，这样是为了阻止发送器继续发送下一帧数据。当接收缓冲区满时，nRTS保持高电平。

CTS 流控

USART发送器监视nCTS输入引脚来决定数据帧是否可以发送。如果USART_STAT寄存器中

TBE位是0且nCTS为低电平，发送器发送数据帧。在发送期间，若nCTS信号变为高电平，发送器将会在当前数据帧发送完成后停止发送。

图 19-8. 硬件流控制



RS485 驱动使能

驱动使能功能通过设置USART_CTL2控制寄存器的DEM位来打开。它允许用户通过DE（Driver Enable）信号激活外部收发器控制。提前时间是驱动使能信号和第一个字节的起始位之间的时间间隔。这个时间可以在USART_CTL0控制器的DEA[4:0]位域中进行设置。滞后时间是一个发送信息最后一个字节的停止位与释放DE信号之间的时间间隔。这个时间可以在USART_CTL0控制寄存器的DED[4:0]位域中进行设置。DE信号的极性可以通过USART_CTL2控制寄存器的DEP位进行设置。

19.3.7. 多处理器通信

在多处理器通信中，多个USART被连接成一个网络。对于一个设备来说，监视所有来自RX引脚的消息，是一种巨大的负担。为减轻设备负担，软件可以通过将USART_CMD寄存器中MMCMD位置1使USART进入静默模式。

如果USART处于静默模式，所有的接收状态标志位将不会被置位。此外，USART可以由硬件用以下两种方式中的一种来唤醒：空闲总线检测和地址匹配检测。

设备默认使用空闲总线检测方法唤醒USART。如果RWU位为0，RX引脚检测到空闲帧，USART_STAT寄存器中的IDLEF位会置位。如果RWU位置位，RX引脚检测到空闲帧时，硬件会将RWU清零，从而退出静默模式，当它是被空闲帧唤醒时，USART_STAT寄存器中IDLEF位不会被置1。

当USART_CTL0寄存器中WM被置位，数据最高位会被认为是地址标志位。如果地址标志位为1，该字节被认为是地址字节。如果地址标志位是0，该字节被认为是数据字节。如果地址字节的低4位或低7位与USART_CTL1寄存器中的ADDR位相同，硬件会将RWU清零，并退出静默模式。接收到将USART唤醒的数据帧，RBNE将置位。状态标志可以从USART_STAT寄存器中获取。如果地址字节的低4位或低7位与USART_CTL1寄存器中的ADDR位不相同，硬件会置位RWU并自动进入静默模式。在这种情况下，RBNE不会被置位。

如果USART_CTL0寄存器中PCEN位被置位，地址字节最高位被视为校验位，其余位被视为地址位。如果ADDM位被置位，且接收帧为7位的数据，其中最低的6位将与ADDR[5:0]比较。如果ADDM位被置位，且接收帧为9位的数据，其中低8位将与ADDR[7:0]进行比较。

19.3.8. LIN 模式

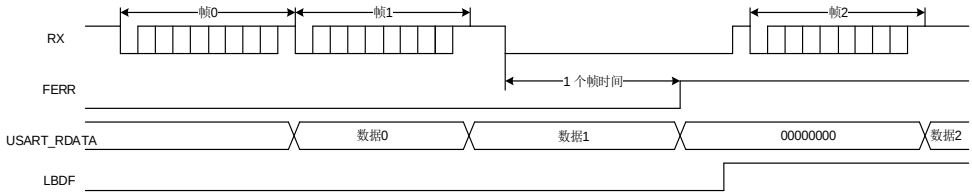
将USART_CTL1寄存器的LMEN置位即可使能本地互联网络模式。在LIN模式下，USART_CTL1寄存器中CKEN，STB[1:0]和USART_CTL2的SCEN，HDEN，IREN位都应该被清0。

在发送一个普通数据帧时，LIN发送过程与普通发送过程相同。数据位的长度只能为8。一个停止位后连续13个0为断开帧。

断开检测功能完全独立于普通USART接收器。因此，断开检测可以是在空闲状态下，也可以在数据传输过程中。USART_CTL1寄存器中LBLEN位可以选择断开帧的长度。如果在RX引脚检测到大于或等于与预期的断开帧长度的0（LBLEN=0时，10个0；LBLEN=1时，11个0），USART_STAT寄存器中LBDF置位。如果USART_CTL1寄存器中LBDIE被置位，将产生中断。

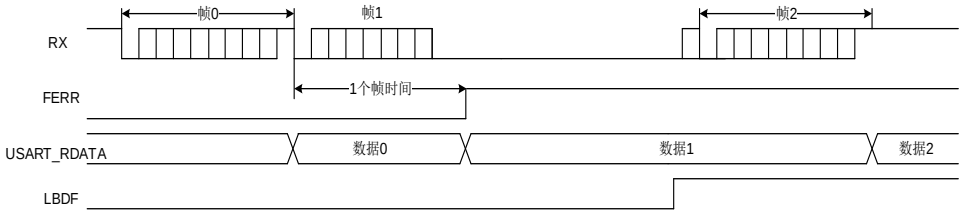
如*图 19-9. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART 接收器会接收到一个全 0 数据帧，同时 FERR 置位。

图 19-9. 空闲状态下检测断开帧



如*图 19-10. 数据传输过程中检测断开帧*所示，如果断开帧发生在数据传输过程中，当前传输帧发生错误，FERR 置位。

图 19-10. 数据传输过程中检测断开帧



19.3.9. 同步通信模式

USART支持主机模式下的全双工同步串行通信，可以通过置位USART_CTL1的CKEN位来使能。在同步模式下，USART_CTL1的LMEN和USART_CTL2的SCEN，HDEN，IREN位应被清0。CK引脚作为USART同步发送器的时钟输出，仅当TEN位被使能时，它才被激活。在起始位和停止位传送期间，不会从CK引脚输出时钟脉冲。USART_CTL1的CLEN位用来决定在最低位（地址索引位）发送期间是否有时钟信号输出。在空闲状态和断开帧的发送过程中，也不会有时钟信号产生。USART_CTL1的CPH位用来决定数据在第一个时钟沿被采样还是在第二个时钟沿被采样。USART_CTL1的CPL位用来决定在USART同步模式空闲状态下，时钟引脚的电平。

CK引脚输出波形由USART_CTL1寄存器中CPL，CPH，CLEN位决定。软件仅在USART禁用（UEN=0）时才可以改变它们的值。

时钟与已发送的数据同步。同步模式下的接收器按照发送器的时钟进行采样，并无任何过采样。

图 19-11. 同步模式下的 USART 示例

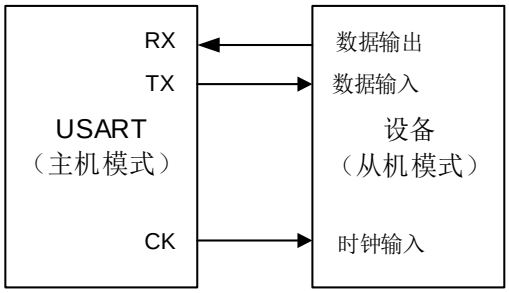
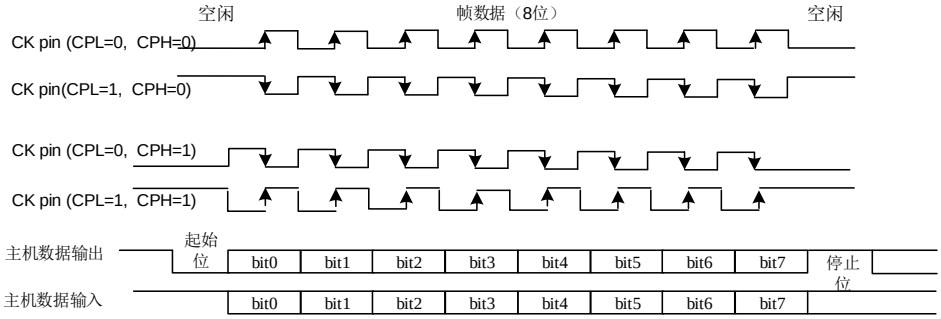


图 19-12. 8-bit 格式的 USART 同步通信波形（CLEN=1）

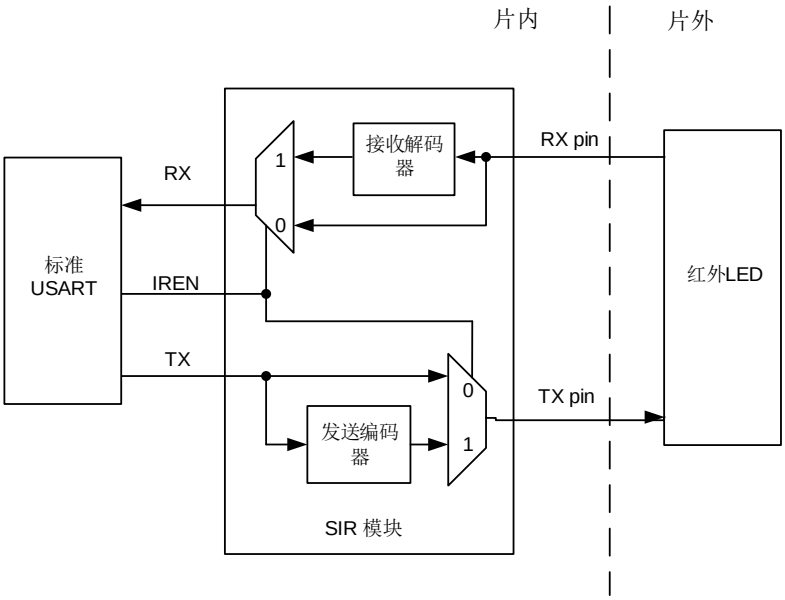


19.3.10. 串行红外（IrDA SIR）编解码功能模块

串行红外编解码功能通过置位USART_CTL2寄存器中IREN使能。在IrDA模式下，USART_CTL1寄存器的LMEN，STB[1:0]，CKEN位和USART_CTL2寄存器的HDEN，SCEN位应被清0。

在IrDA模式下，USART数据帧由SIR发送编码器进行调制，调制后的信号经由红外LED进行发送，经解调后将数据发送至USART接收器。对于编码器而言，波特率应小于115200。

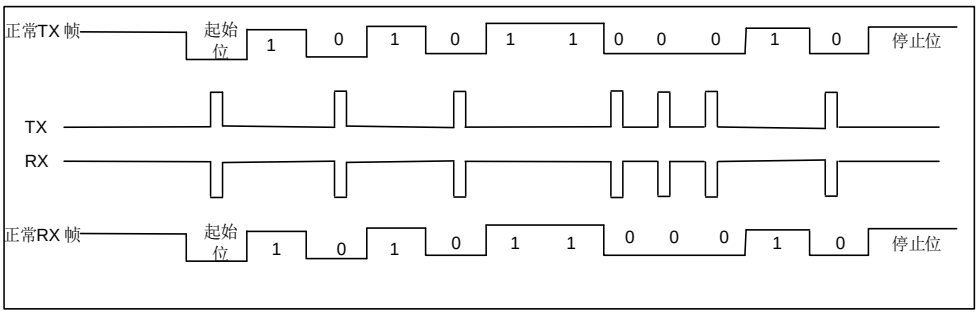
图 19-13. IrDA SIR ENDEC 模块



在IrDA模式下，TX引脚与RX引脚电平不同。TX引脚通常为低电平，RX引脚通常为高电平。IrDA引脚电平保持稳定代表逻辑‘1’，红外光源脉冲（RTZ信号）代表逻辑‘0’。其脉冲宽度通常占一个位时间的3/16。IrDA无法检测到宽度小于1个PSC时钟的脉冲。如果脉冲宽度大于1但是小于2倍PSC时钟，IrDA则无法可靠地检测到。

由于IrDA是一种半双工协议，因此在IrDA SIR ENDEC模块中，发送和接收不得同时进行。

图 19-14. IrDA 数据调制



将USART_CTL2寄存器中IRLP置位可以使SIR子模块工作在低功耗模式下。发送编码器由PCLK分频得到的低速时钟来驱动。分频系数在USART_GP寄存器中PSC[7:0]位配置。TX引脚脉冲宽度可以为低功耗波特率的3倍。接收解码器工作模式与正常IrDA模式相同。

19.3.11. 半双工通信模式

通过设置USART_CTL2寄存器的HDEN位，可以使能半双工模式。在半双工通信模式下，USART_CTL1寄存器的LMEN，CKEN位和USART_CTL2寄存器的SCEN，IREN位应被清零。

半双工模式下仅用单线通信。TX引脚和RX引脚从内部连接到一起，TX引脚应被配置为IO管脚。通信冲突应由软件处理。当TEN被置位时，在数据寄存器中的数据将会被发送。

19.3.12. 智能卡（ISO7816-3）模式

智能卡模式是一种异步通信模式，支持ISO7816-3协议。支持字节模式（T=0）和块模式（T=1）。将USART_CTL2寄存器的SCEN位置1，即可使能智能卡模式。在智能卡模式下，USART_CTL1寄存器的LMEN位和USART_CTL2的HDEN，IREN位应该清0。

如果CKEN位被置位，USART将向智能卡提供一个时钟。该时钟可以分频用于其他用途。

智能卡模式下的帧格式为：1起始位+9数据位（包括1个奇偶校验位）+1.5停止位。

智能卡模式是一种半双工通信协议模式。当与智能卡连接时，TX引脚须被设置成开漏模式，这个引脚将会与智能卡驱动同一条双向连线。

图 19-15. ISO7816-3 数据帧格式

字节模式（T=0）

相较于正常操作模式下的时序，从发送移位寄存器到TX引脚的传递时间延迟了半个波特率时钟，并且TC标志的置位将根据USART_GP寄存器的GUAT[7:0]设置延迟某一特定时间。在智能卡模式下，在最后一帧数据的停止位之后，内部保护时间计数器将开始计数，GUAT[7:0]的值配置为ISO7816-3协议的CGT减12。在保护时间寄存器向上计数这段时间TC将被强制拉低，当计数达到设定值时，TC被置位。

在USART发送期间，如果检测到有奇偶校验错误，TX引脚在停止位最后一个位时间内被拉低，智能卡发送一个NACK信号。根据协议，USART会自动重发SCRTNUM次。在重发数据帧前面会插入2.5位的帧间隔。最后一次重发字节后，TC会立即被置位。如果在最大重发次数后仍然收到NACK信号，USART将会停止发送，帧错误标志被置位。USART不会将NACK信号作为起始位。

在USART接收期间，如果在当前数据帧检测到校验错误，TX引脚在停止位的最后一个位时间内会被拉低。智能卡会接收到NACK信号。然后在智能卡端会产生一个帧错误。如果接收到的字节是错误的，RBNE中断和接收DMA请求都不会被激活。根据协议，智能卡将重新发送数据。如果在最大的重新发送次数后（这个次数的具体值在SCRTNUM位域），接收到的字符仍然是错误的，USART停止发送NACK信号和标注这个错误为奇偶校验错误。将USART_CTL2寄存器中的NKEN置位可以使能NACK信号。

空闲帧和断开帧在智能卡模式下不适用。

块模式（T=1）

在T=1（块模式）下，USART_CTL2寄存器的NKEN位应该清零来关闭校验错误发送。

当要从智能卡读取数据时，软件必须将USART_RT寄存器的RT[23:0]位域设置成BWT（块等待时间）-11的值，并将RBNEIE置位。如果到了这个时间，还没有从智能卡收到应答，将引起超时中断。如果在超时之前收到了第一个字节，则会引起RBNE中断。块模式下，如果用DMA从智能卡读取数据，也只能在第一个字节接收完后再去使能DMA。

在接收到第一个字节之后（RBNE中断）必须将USART_RT寄存器设置为CWT（字节等待时间）-11之间的某个值（这个时间以波特时间作为单位），这是为了自动检测两个连续字符之间的最大等待时间。如果智能卡在前一个字符发送结束后到设定的CWT周期之间没有发送字符，USART会通过RTF标志提醒软件，当RTIE被置位时，会引起中断。

USART用一个块长度计数器统计收到的字节数，这个计数器在USART开始发送的时候自动清0（TBE=0）。这个块长度信息位于智能卡发出数据的第三个字节（序言部分）。这个值必须写入USART_RT寄存器的BL[7:0]。当使用DMA模式时，在块开始之前，这个寄存器必须被设定为最小值（0x0）。为了得到这个值，在收到第四个字节后，会引起一个中断。软件可以从接收缓冲区读取第三个字节作为块长度。

在中断驱动接收模式，块的长度可以由软件提取出来并做检测或者通过设置BL的值得到。但是在块开始之前，BL（0xFF）可以被设置为最大值。实际值则要在接收到第三个字节后写到寄存器中。

整个块的长度（包括序言区，收尾区和信息区）等于BL+4。块尾通过EBF标志和相应中断提醒给软件（当EBIE位置1时）。如果块长度出错，将会引起一个RT中断。

直接和反向转换

智能卡协议定义了两种转换方式：直接转换和反向转换。

如果选择直接转换，从数据帧的最低位开始传输，TX引脚高电平代表逻辑‘1’，偶校验。在这种情况下，MSBF位和DINV位都应设置为0（默认值）。

如果选择反向转换，从数据帧的最高位开始传输，TX引脚低电平代表逻辑‘1’，偶校验。在这种情况下，MSBF位和DINV位都应设置为1。

19.3.13. ModBus 通信

通过实现块尾检测功能，USART提供实现ModBus/RTU和ModBus/ASCII协议的基本支持。

在ModBus/RTU模式下，通过一个超过2个字符长度的空闲状态来识别块尾。这个功能是通过一个可编程的超时检测功能来实现的。

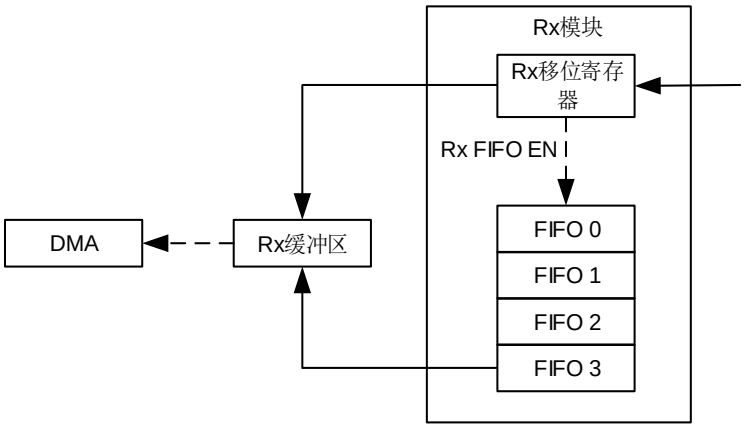
为了检测空闲状态，必须置位USART_CTL1寄存器的RTEN位和USART_CTL0寄存器的RTIE位。USART_RT寄存器必须被设置成与2个字节超时所对应的值。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件当前块接收已经完成。

在ModBus/ASCII模式下，块尾被认为是一个特定的字符（CR/LF）串。USART用字符匹配机制实现这个功能。具体是通过将LF的ASCII码配置到ADDR区域并激活地址匹配中断（AMIE=1）来实现。软件将在收到LF或可以在DMA缓存中查找到CR/LF时得到提示。

19.3.14. 接收 FIFO

通过将USART_RFCS寄存器的RFEN置位使能接收FIFO，可以避免当CPU无法迅速响应RBNE中断时，发生过载错误。接收FIFO和接收缓存区可储存多至5帧的数据。若接收FIFO满，RFFINT位将被置位。如果RFFIE被置位，将产生中断。

图 19-16. USART 接收 FIFO 结构



如果软件在响应RBNE中断时读数据接收缓冲区，在响应开始时，RBNEIE位应清0。当所有接收的数据被读出后，RBNEIE位应置位。在读出接收的数据前，PERR，NERR，FERR，EBF都应被清0。

19.3.15. 从 Deepsleep 模式唤醒

通过标准RBNE中断或WUM中断USART能从深度睡眠模式唤醒MCU。

UESM位必须置1并且USART时钟必须设置为IRC16M或LXTAL（请参考*配置寄存器 2（RCU_CFG2）*）。

当使用RBNE标准中断时，必须在进入深度睡眠模式前将RBNEIE位置位。

当使用WUIE中断时，WUIE中断源可以通过WUM位来选择。

在进入深度睡眠模式前，必须禁用DMA。在进入深度睡眠模式前，软件必须检测USART是否正在传送数据。这可以通过USART_STAT寄存器中的BSY标志来判断。REA位必须被检测以确保USART是使能的。

当检测到唤醒事件时，无论MCU工作在深度睡眠模式还是正常模式，WUF标志位通过硬件被置1，并且在WUIE被置位的情况下，触发一个唤醒中断。

19.3.16. USART 中断

USART 中断事件和标志如*表 19-3. USART 中断请求*所示：

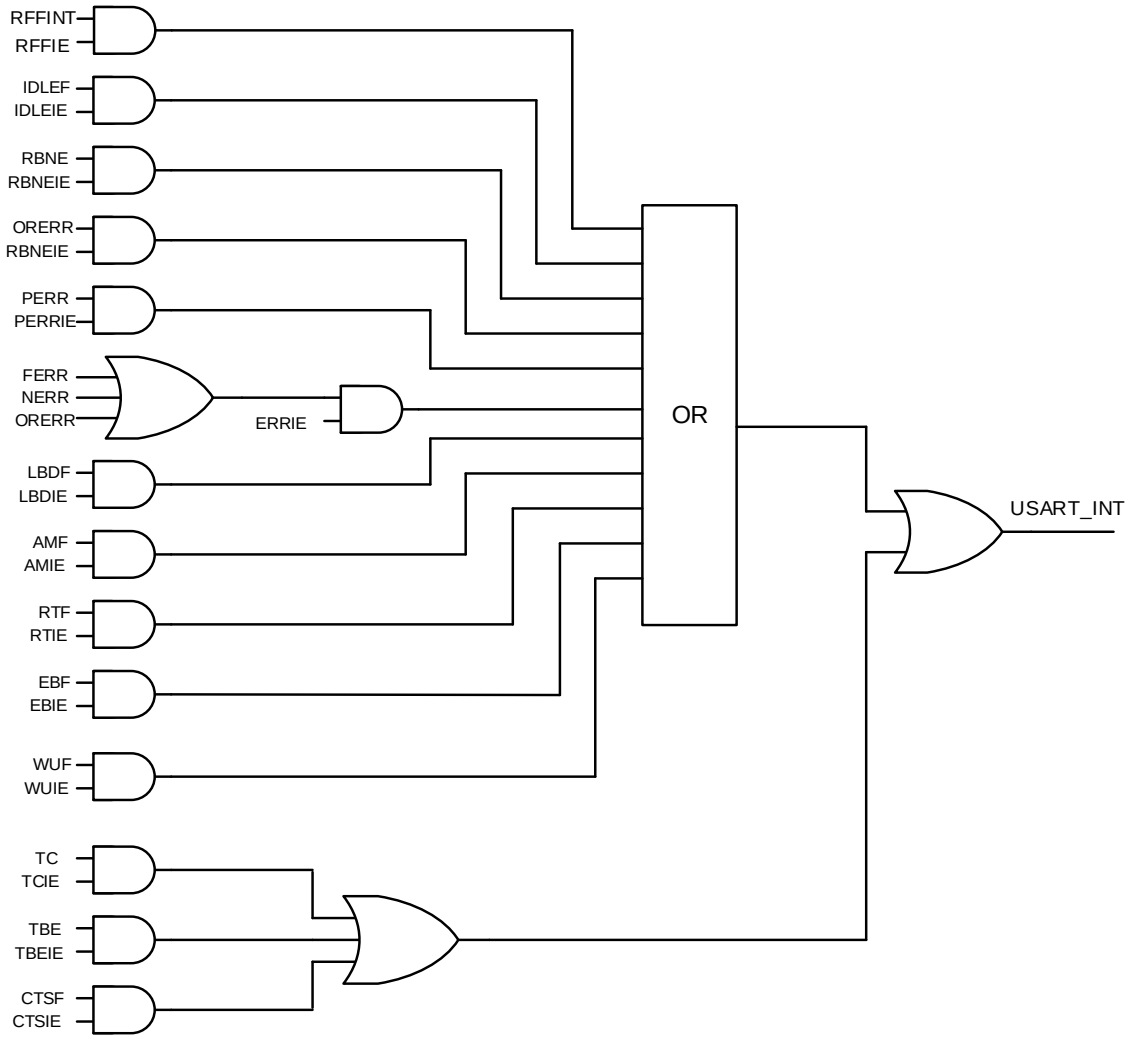
表 19-3. USART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器空	TBE	TBEIE

中断事件	事件标志	使能控制位
CTS标志	CTSF	CTSIE
发送结束	TC	TCIE
接收到的数据可以读取	RBNE	RBNEIE
检测到过载错误	ORERR	
接收FIFO满	RFFINT	RFFIE
检测到线路空闲	IDLEF	IDLEIE
奇偶校验错误	PERR	PERRIE
LIN模式下，检测到断开标志	LBDF	LBDIE
当DMA接收使能时，接收错误（噪声错误、溢出错误、帧错误）	NERR或ORERR或FERR	ERRIE
字符匹配	AMF	AMIE
接收超时错误	RTF	RTIE
发现块尾	EBF	EBIE
从Deepsleep模式唤醒	WUF	WUIE

在发送给中断控制器之前，所有的中断事件是逻辑或的关系。因此在任何时候 USART 只能向控制器产生一个中断请求。不过软件可以在一个中断服务程序里处理多个中断事件。

图 19-17. USART 中断映射框图



19.4. USART 寄存器

USART0基地址：0x4001 3800
USART1基地址：0x4000 4400
UART3基地址：0x4000 4C00
UART4基地址：0x4000 5000

19.4.1. USART 控制寄存器 0（USART_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				EBIE	RTIE	DEA[4:0]					DED[4:0]				
				rw	rw	rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	AMIE	MEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	UESM	UEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27	EBIE	块尾中断使能 0：中断禁止 1：中断使能 在UART3和UART4中，该位保留。
26	RTIE	接收超时中断使能 0：中断禁止 1：中断使能 在UART3和UART4中，该位保留。
25:21	DEA[4:0]	驱动使能置位时间 这些数字用来定义DE（驱动使能）信号的置位与第一个字节的起始位之间的时间间隔。它以采样时间为单位（1/8或1/16位时间），可以通过OVSMOD位来配置。 当USART被使能（UEN=1）时，该位域不能被改写。
20:16	DED[4:0]	驱动使能置低时间 这些位用来定义一个发送信息最后一个字节的停止位与置低DE（驱动使能）信号之间的时间间隔。它以采样时间为单位（1/8或1/16位时间），可以通过OVSMOD位来配置。 当USART被使能（UEN=1）时，该位域不能被改写。
15	OVSMOD	过采样模式 0：16倍过采样 1：8倍过采样

		在LIN, IrDA 和智能卡模式，该位保持清0。 当USART被使能（UEN=1）时，该位域不能被改写。
14	AMIE	ADDR字符匹配中断使能 0：ADDR字符匹配中断禁用 1：ADDR字符匹配中断使能
13	MEN	静默模式使能 0：静默模式禁用 1：静默模式被使能
12	WL	字长 0：8数据位 1：9数据位 当USART被使能（UEN=1）时，该位域不能被改写。
11	WM	从静默模式唤醒方法 0：空闲线 1：地址标记 当USART被使能（UEN=1）时，该位域不能被改写。
10	PCEN	校验控制使能 0：校验控制禁用 1：校验控制被使能 当USART被使能（UEN=1）时，该位域不能被改写。
9	PM	校验模式 0：偶校验 1：奇校验 当USART被使能（UEN=1）时，该位域不能被改写。
8	PERRIE	校验错误中断使能 0：校验错误中断禁用 1：当USART_STAT寄存器的PERR位置位时，将触发中断。
7	TBEIE	发送寄存器空中断使能 0：中断禁止 1：当USART_STAT寄存器的TBE位置位时，将触发中断。
6	TCIE	发送完成中断使能 如果该位置1，USART_STAT寄存器中TC被置位时产生中断。 0：发送完成中断禁用 1：发送完成中断使能
5	RBNEIE	读数据缓冲区非空中断和过载错误中断使能 0：读数据缓冲区非空中断和过载错误中断禁用 1：当USART_STAT寄存器的ORERR或RBNE位置位时，将触发中断。
4	IDLEIE	IDLE线检测中断使能 0：IDLE线检测中断禁用

		1：当USART_STAT寄存器的IDLEF位置位时，将触发中断。
3	TEN	发送器使能 0：发送器关闭 1：发送器打开
2	REN	接收器使能 0：接收器关闭 1：接收器打开并且开始搜索起始位。
1	UESM	USART在深度睡眠模式下使能 0：USART不能从深度睡眠模式唤醒MCU 1：USART能从深度睡眠模式唤醒MCU。条件是USART的时钟源必须是IRC16M或LXTAL。 在UART3和UART4中，该位保留。
0	UEN	USART使能 0：USART预分频器和输出禁用 1：USART预分频器和输出被使能

19.4.2. USART 控制寄存器 1（USART_CTL1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[7:0]								RTEN	保留			MSBF	DINV	TINV	RINV
rw								rw				rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRP	LMEN	STB[1:0]		CKEN	CPL	CPH	CLEN	保留	LBDIE	LBLEN	ADDM	保留			
rw	rw	rw		rw	rw	rw	rw		rw	rw	rw				

位/位域	名称	描述
31:24	ADDR[7:0]	USART的节点地址 这些位给出USART的节点地址。 在多处理器通信并且静默模式或者深度睡眠模式期间，这些位用来唤醒进行地址标记的检测。接收到的最高位为1的数据帧将和这些位进行比较。当ADDM位被清零时，仅仅ADDR[3:0]被用来比较。 在正常的接收期间，这些位也用来进行字符检测。所有接收到的字符（8位）与ADDR[7:0]的值进行比较，如果匹配，AMF标志将被置位。 当接收器（REN=1）和USART（UEN=1）被使能时，该位域不能被改写。
23	RTEN	接收器超时使能 0：接收器超时功能禁用 1：接收器超时功能被使能

		在UART3和UART4中，该位保留。
22:20	保留	必须保持复位值。
19	MSBF	高位在前 0：数据发送/接收，采用低位在前 1：数据发送/接收，采用高位在前 USART被使能（UEN=1）时，该位域不能被改写。
18	DINV	数据位反转 0：数据位信号值没有反转 1：数据位信号值被反转 USART被使能（UEN=1）时，该位域不能被改写。
17	TINV	TX管脚电平反转 0：TX管脚信号值没有反转 1：TX管脚信号值被反转. USART被使能（UEN=1）时，该位域不能被改写。
16	RINV	RX管脚电平反转 0：RX管脚信号值没有反转. 1：RX管脚信号值被反转 USART被使能（UEN=1）时，该位域不能被改写。
15	STRP	交换TX/RX管脚 0：TX和RX管脚功能不被交换 1：TX和RX管脚功能被交换 当USART被使能（UEN=1）时，该位域不能改写。
14	LMEN	LIN模式使能 0：LIN模式关闭 1：LIN模式开启 USART被使能（UEN=1）时，该位域不能被改写。 在UART3和UART4中，该位保留。
13:12	STB[1:0]	STOP位长 00：1停止位 01：0.5停止位 10：2停止位 11：1.5停止位 USART被使能（UEN=1）时，该位域不能被改写。
11	CKEN	CK管脚使能 0：CK管脚禁用 1：CK管脚被使能 USART被使能（UEN=1）时，该位域不能被改写。 在UART3和UART4中，该位保留。
10	CPL	时钟极性

		0：在同步模式下，CK管脚不对外发送时保持为低电平 1：在同步模式下，CK管脚不对外发送时保持为高电平 USART被使能（UEN=1）时，该位域不能被改写。
9	CPH	时钟相位 0：在同步模式下，在首个时钟边沿采样第一个数据 1：在同步模式下，在第二个时钟边沿采样第一个数据 USART被使能（UEN=1）时，该位域不能被改写。
8	CLEN	CK长度 0：在同步模式下，最后一位（MSB）的时钟脉冲不输出到CK管脚 1：在同步模式下，最后一位（MSB）的时钟脉冲输出到CK管脚 USART被使能（UEN=1）时，该位域不能被改写。
7	保留	必须保持复位值。
6	LBDIE	LIN断开信号检测中断使能 0：断开信号检测中断禁用 1：当USART_STAT的LBDF位置位，将产生中断。 在USART1，该位保留。
5	LBLEN	LIN断开帧长度 0：检测10位断开帧 1：检测11位断开帧 USART被使能（UEN=1）时，该位域不能被改写。 在UART3和UART4中，该位保留
4	ADDM	地址检测模式 该位用来选择4位地址检测或全位地址检测。 0：4位地址检测 1：全位地址检测。在7位，8位和9位数据模式下，地址检测分别按6位，7位和8位地址（ADDR[5:0]，ADDR[6:0]和ADDR[7:0]）执行。 USART被使能（UEN=1）时，该位域不能被改写。
3:0	保留	必须保持复位值。

19.4.3. USART 控制寄存器 2（USART_CTL2）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									WUIE	WUM[1:0]		SCRTNUM[2:0]			保留
									rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRD	OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	WUIE	从深度睡眠模式唤醒中断使能 0：从深度睡眠模式唤醒中断禁用 1：从深度睡眠模式唤醒中断被使能 在USART1，该位保留。
21:20	WUM[1:0]	从深度睡眠模式唤醒模式 这个位域指定什么事件可以置位USART_STAT寄存器中的WUF（从深度睡眠唤醒标志）标志。 00：WUF在地址匹配的时候置位。如何实现地址匹配在ADDR和ADDM中定义。 01:保留 10：WUF在检测到起始位时置位 11：WUF在检测到RBNE时置位 USART被使能（UEN=1）时，该位域不能被改写。 在USART1，该位保留。
19:17	SCRTNUM[2:0]	智能卡自动重试数目 在智能卡模式下，这些位用来指定在发送和接收时重试的次数。在发送模式下，它指的是在产生发送错误（FERR位置位）之前自动重试的发送次数。 在接收模式下，它指的是在产生接收错误（RBNE位和PERR位置位）之前自动重试的接收次数。 当这些位被设置为0x0时，在发送模式下这些位将不会自动发送。 USART被使能（UEN=1）时，该位域被清零，并停止重发。 在UART3和UART4中，该位保留。
16	保留	必须保持复位值。
15	DEP	驱动使能的极性选择模式 0：DE信号高有效 1：DE信号低有效 USART被使能（UEN=1）时，该位域不能被改写。
14	DEM	驱动使能模式 用户使能该位以后，可以通过DE信号对外部收发器进行控制。DE信号是从RTS 管脚输出的。 0：DE功能禁用 1：DE功能开启 USART被使能（UEN=1）时，该位域不能被改写。
13	DDRE	在接收错误时禁止DMA 0：在发生接收错误的情况下，不禁用DMA。所有的错误数据不会产生DMA请求，以确保错误的数据不会被传输，但是下一个接收到的正确的数据会被传输。RBNE位保持0以阻止过载错误，但是相应错误标志位会被置位。这种模式可用于智能卡模式。 1：在接收错误的情况下，DMA被关闭。 DMA请求会被屏蔽，直到相应的标志位被

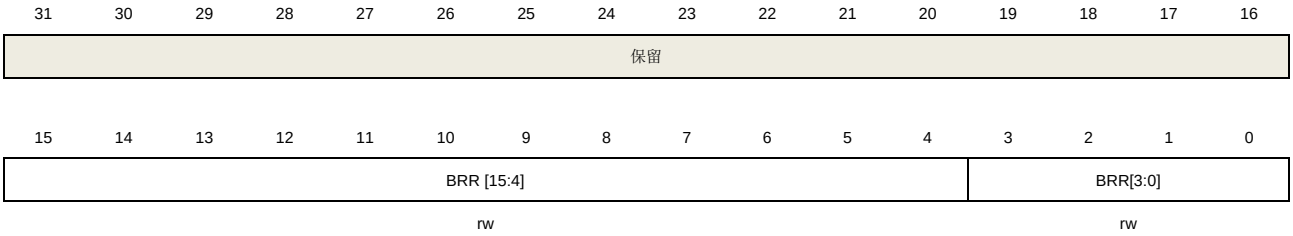
		清0。RBNE标志和相应的错误标志位会被置位。软件在清除错误标志前，必须首先关DMA请求（DMAR = 0）或清RBNE。 USART被使能（UEN=1）时，该位域不能被改写。
12	OVRD	溢出禁止 0：溢出功能被使能。当接收到的数据在新数据到达前没有被读走，ORERR错误标志位将被置位，并且新数据将会丢失。 1：溢出功能禁止。当接收到的数据在新数据到达前没有被读走，ORERR错误标志位将不会被置位，新数据会将USART_RDATA寄存器以前的内容覆盖。 USART被使能（UEN=1）时，该位域不能被改写。
11	OSB	单次采样方式 0：三次采样方法 1：一次采样方法 USART被使能（UEN=1）时，该位域不能被改写。
10	CTSIE	CTS中断使能 0：CTS中断屏蔽 1：当USART_STAT的CTS位置位时，会产生中断。
9	CTSEN	CTS使能 0：CTS硬件流控禁用 1：CTS硬件流控被使能 USART被使能（UEN=1）时，该位域不能被改写。
8	RTSEN	RTS使能 0：RTS硬件流控禁用 1：RTS硬件流控被使能，只有当接收缓冲区有空间的时候，才会请求下一个数据。 USART被使能（UEN=1）时，该位域不能被改写。
7	DENT	DMA发送使能 0：关闭DMA发送模式 1：开启DMA发送模式
6	DENR	DMA接收使能 0：关闭DMA接收模式 1：开启DMA接收模式
5	SCEN	智能卡模式使能 0：智能卡模式禁用 1：智能卡模式使能 USART被使能（UEN=1）时，该位域不能被改写。 在UART3和UART4中，该位保留。
4	NKEN	智能卡模式NACK使能 0：当出现校验错误时不发送NACK 1：当出现校验错误时发送NACK USART被使能（UEN=1）时，该位域不能被改写。

		在UART3和UART4中，该位保留。
3	HDEN	半双工使能 0：禁用半双工模式 1：开启半双工模式 USART被使能（UEN=1）时，该位域不能被改写。
2	IRLP	IrDA低功耗模式 0：正常模式 1：低功耗模式 USART被使能（UEN=1）时，该位域不能被改写。
1	IREN	IrDA模式使能 0：IrDA禁用 1：IrDA被使能 USART被使能（UEN=1）时，该位域不能被改写。 在UART3和UART4中，该位保留。
0	ERRIE	多级缓存通信模式的错误中断使能 0：禁用错误中断 1：在多级缓存通信时，当USART_STAT寄存器的FERR位，ORERR位或NERR位被置位时，会产生中断。

19.4.4. USART 波特率寄存器（USART_BAUD）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当USART（UEN=1）被使能时，该寄存器不能被改写。

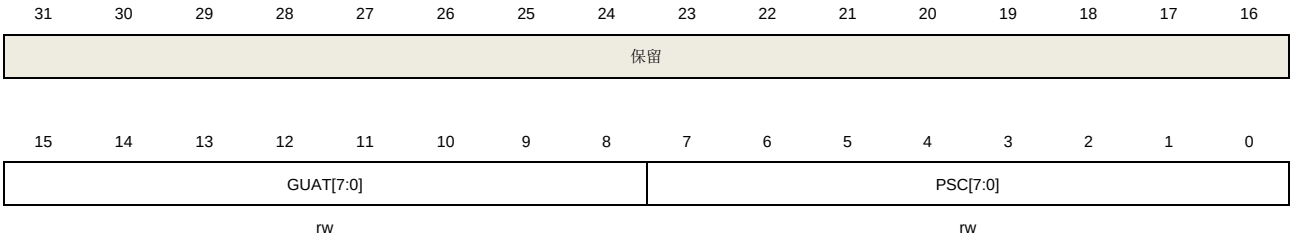


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:4	BRR[15:4]	波特率分频系数的整数部分 INTDIV = BRR[15:4]
3:0	BRR[3:0]	波特率分频系数的小数部分 如果OVSMOD = 0，FRADIV = BRR [3:0]； 如果OVSMOD = 1，FRADIV = BRR [2:0]，BRR [3]必须被置0。

19.4.5. USART 保护时间和预分频器寄存器（USART_GP）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
USART被使能（UEN=1）时，该寄存器不能被改写。
在USART1中，该寄存器保留。



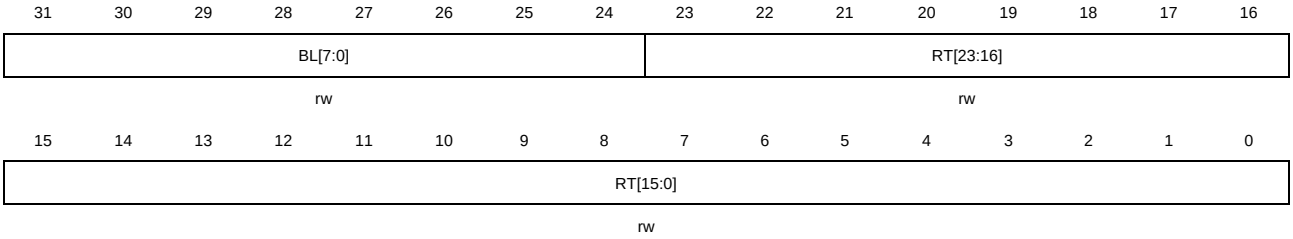
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	GUAT[7:0]	在智能卡模式下的保护时间值 USART被使能（UEN=1）时，该位域不能被改写。
7:0	PSC[7:0]	预分频器值 在红外低功耗模式下，对系统时钟进行分频已获得低功耗模式下的频率。寄存器的值是分频系数 00000000：保留 – 不设置这个值 00000001：1分频 00000010：2分频 ... 在IrDA正常模式下的分频值 00000001：仅能设为这个值 在智能卡模式下，对系统时钟进行分频的值存于PSC[4:0]位域中。PSC[7:5]位保持为复位值。分频系数是寄存器中值的两倍。 00000：保留 -不设置这个值 00001：2分频 00010：4分频 00011：6分频 ... USART被使能（UEN=1）时，该位域不能被改写。

19.4.6. USART 接收超时寄存器（USART_RT）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

在UART3和UART4中，该寄存器保留。



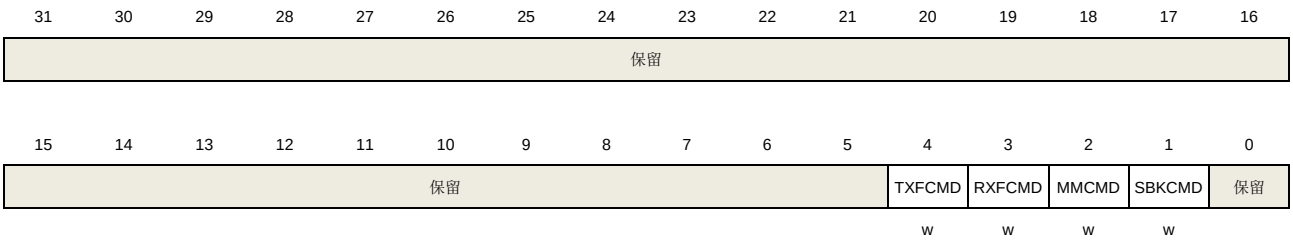
位/位域	名称	描述
31:24	BL[7:0]	块长度 这些位给出了智能卡T=1的接收时块的长度。它的值等于信息字节的长度+结束部分的长度（1-LEC/2-CRC）–1。 这个值可以在块接收开始时设置（用于需要从块的序言提取块的长度的情形），这个只在每一个接收时钟周期只能设置一次。在智能卡模式下，当TBE=0时，块的长度计数器被清0。 在其他模式下，当REN=0（禁用接收器）并且/或者当EBC位被写1时块的长度计数器被清0。
23:0	RT[23:0]	接收器超时门限 该位域指定接收超时值，单位是波特时钟的时长 标准模式下，如果在最后一个字节接收后，在RT规定的时长内，没有检测到新的起始位，RTF标志被置位。 在智能卡模式，这个值被用来实现CWT和BWT。在这种情况下，超时检测是从最后一个接收字节的起始位开始。 这些位可以在工作时改写。假如一个新数据到来的时间比RT规定的晚，RTF标志会被置位。对于每个接收字符，这个值只能改写一次。

19.4.7. USART 请求寄存器（USART_CMD）

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:5	保留	必须保持复位值。
4	TXFCMD	发送数据清空请求 向该位写1去置位TBE标志位，以取消发送数据。

		在UART3和UART4中，该位保留。
3	RXFCMD	接收数据清空请求 向该位写1来清除RBNE标志位，以丢弃未读的接收数据。
2	MMCMD	静默模式请求 向该位写1使USART进入静默模式并且置位RWU标志位。
1	SBKCMD	发送断开帧请求 向该位写1置位SBKF标志并使USART在空闲时发送一个断开帧。
0	保留	必须保持复位值。

19.4.8. USART 状态寄存器（USART_STAT）

地址偏移：0x1C
复位值：0x0000 00C0

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留									REA	TEA	WUF	RWU	SBF	AMF	BSY
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			EBF	RTF	CTS	CTSF	LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
			r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	REA	接收使能通知标志 这位反映了USART核心逻辑的接收使能状态，该位可以通过硬件设置。 0：USART核心接收逻辑禁用 1：USART核心接收逻辑被使能
21	TEA	发送使能通知标志 该位反映了USART核心逻辑的发送使能状态，该位可以通过硬件设置。 0：USART核心发送逻辑禁用 1：USART核心发送逻辑被使能
20	WUF	从深度睡眠模式唤醒标志 0：没有从深度睡眠模式唤醒 1：已从深度睡眠模式唤醒， 如果在USART_CTL2寄存器的WUFIE=1并且MCU处于深度睡眠模式，将引发一个中断。 当检测到一个唤醒事件时，该位通过硬件置位，这个事件在WUM位域被定义。 向USART_INTC寄存器中的WUC写1，该位被清0。 当UESM被清0时，该位清0。 在UART3和UART4中，该位保留。

19	RWU	接收器从静默模式唤醒 该位表示USART处于静默模式。 0：接收器在工作状态 1：接收器在静默状态 当在唤醒和静默模式切换时，它通过硬件清0或者置1。静默模式控制（地址帧还是空闲帧）是用通过USART_CTL0寄存器的WM位选择。 如果选择空闲信号唤醒，只能通过向USART_CMD寄存器的MMCMD位写1来将该位置位。
18	SBF	断开信号发送标识 0：没发送断开字符 1：将要发送断开字符 该位表示一个断开发送信号被请求。 通过向USART_CMD寄存器的SBKCMD写1来置位。 在断开帧的停止位发送期间，硬件清0。
17	AMF	ADDR匹配标志 0：ADDR和接收到的字符不匹配 1：ADDR和接收到的字符匹配，如果USART_CTL0寄存器的AMIE=1，将引发一个中断。 当接收到ADDR [7:0]中定义的字符时，硬件置位。 通过向USART_INTC寄存器的AMC写1清0。
16	BSY	忙标志 0：USART处于空闲 1：USART正在接收
15:13	保留	必须保持复位值。
12	EBF	块结束标志 0：块没有结束 1：块结束已到（足够的字节数），如果USART_CTL1寄存器的EBIE=1，将引发一个中断。 当接收到的字节数（从块开始，包括序言部分）等于或大于BLEN + 4，硬件置位。 通过向USART_INTC寄存器的EBC写1清0。 在UART3和UART4中，该位保留。
11	RTF	接收超时标志 0：尚未超时 1：已经超时，如果USART_CTL1寄存器的RTIE被置位，将会引发中断。 如果空闲的时间已经超过了在USART_RT寄存器中设定的RT值，通过硬件置1。 通过向USART_INTC寄存器的RTC位写1清0。 在智能卡模式，这个超时相当于CWT或BWT计时。 在UART3和UART4中，该位保留。
10	CTS	CTS电平 这个值等于nCTS输入引脚电平的反向拷贝。 0：nCTS输入引脚高电平

		1：nCTS输入引脚低电平
9	CTSF	CTS变化标志 0：nCTS状态线没有变化 1：nCTS状态线发生变化 如果USART_CTL2寄存器的CTSIE位置位，将引发中断。 当nCTS输入变化时，由硬件置位。 通过向USART_INTC寄存器的CTSC位写1，清零该位。
8	LBDF	LIN断开检测标志 0：没有检测到LIN断开字符 1：检测到LIN断开字符。当USART_CTL1寄存器的LBDIE位被置位时，将会有中断产生。 当LIN断开帧被检测到的时候，硬件置位。 通过向USART_INTC寄存器的LBDC位写1，清零该位。 在USART1中，该位保留。
7	TBE	发送数据寄存器空 0：数据没有发送到移位寄存器 1：数据发送到移位寄存器。如果USART_CTL0寄存器的TBEIE位置位，将会有中断产生。 当USART_ TDATA寄存器的内容已经被转移到移位寄存器或者向USART_CMD寄存器的TXFCMD位写1时，由硬件置位。 通过向USART_TDATA寄存器中写数据来清0。
6	TC	发送完成 0：发送没有完成 1：发送完成。如果USART_CTL0寄存器的TCIE被置位，将会有中断产生。 如果一个包含数据的帧的发送完成且TBE被置位，该位由硬件置位。 通过向USART_INTC寄存器的TCC位写1清0。
5	RBNE	读数据缓冲区非空 0：没有接收到数据 1：已接收到数据并且可以读取。当寄存器USART_CTL0的RBNEIE位被置位，将会有中断产生。 当接收移位寄存器的内容已经被转移到寄存器USART_ RDATA，由硬件置位。 通过读USART_RDATA寄存器或向USART_CMD寄存器的RXFCMD位写1清0。
4	IDLEF	空闲线检测标志 0：没检测到空闲线 1：检测到空闲线。如果USART_CTL0寄存器的IDLEIE位置1，将会有中断产生。 当检测到空闲线时，通过硬件置位。直到RBNE位置位，否则它不会被再次置位。 向USART_INTC寄存器的IDLEC位写1清0。
3	ORERR	溢出错误 0：未检测到溢出错误 1：检测到溢出错误。在多级缓存通信中，如果寄存器USART_CTL0的RBNEIE位置位，将会引发中断。如果寄存器USART_CTL2的ERRIE位置位也会引发中断。

在RBNE置位的情况下，如果接收移位寄存器的数据传递给USART_RDATA寄存器，将会由硬件置位。
向USART_INTC寄存器的OREC位写1清0。

2	NERR	噪声错误标志 0：未检测到噪声错误 1：检测到噪声错误。在多级缓存通信中，如果寄存器USART_CTL2的ERRIE位置位，将会有中断产生。 在接收帧的时候检测到噪声错误，将会由硬件置位。 向寄存器USART_INTC的NEC位写1清0。
1	FERR	帧错误 0：未检测到帧错误 1：检测到帧错误或者断开字符。在多级缓存通信中，如果寄存器USART_CTL2的ERRIE位置位，将会有中断产生。 当一个不同步，强噪声或者断开字符被检测到时，硬件置位。在智能卡模式下，当发送次数达到上限，仍然没有收到发送成功应答（卡一直响应NACKs），该位也将被置位。 向USART_INTC寄存器的FEC位写1清0。
0	PERR	校验错误 0：未检测到校验错误 1：检测到校验错误，在多级缓存通信中，如果寄存器USART_CTL0的PERRIE位置位，将会有中断产生。 当在接收模式的时候检测到校验错误，将会由硬件置位。 向USART_INTC寄存器的PEC位写1清0。

19.4.9. USART 中断标志清除寄存器（USART_INTC）

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											WUC	保留		AMC	保留
											w			w	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			EBC	RTC	保留	CTSC	LBDC	保留	TCC	保留	IDLEC	OREC	NEC	FEC	PEC
			w	w		w	w		w		w	w	w	w	w

位/位域	名称	描述
31:21	保留	必须保持复位值。
20	WUC	从深度睡眠模式唤醒标志的清除 向该位写1清除USART_STAT寄存器的WUF位。 在UART3和UART4中，该位保留。

19:18	保留	必须保持复位值。
17	AMC	ADDR匹配标志清除 向该位写1清除USART_STAT寄存器的AMF位。
16:13	保留	必须保持复位值。
12	EBC	块结束标志清除 向该位写1清除USART_STAT寄存器的EBF位。 在UART3和UART4中，该位保留。
11	RTC	接收超时标志清除 向该位写1清除USART_STAT寄存器的RTF标志。 在UART3和UART4中，该位保留。
10	保留	必须保持复位值。
9	CTSC	CTS变化标志清除 向该位写1清除USART_STAT寄存器的CTSF位。
8	LBDC	LIN断开字符检测标志清除 向该位写1清除USART_STAT寄存器的LBDF标志位。 在UART3和UART4中，该位保留。
7	保留	必须保持复位值。
6	TCC	发送完成标志清除 向该位写1清除USART_STAT寄存器的TC位。
5	保留	必须保持复位值。
4	IDLEC	空闲线检测标志清除 向该位写1清除USART_STAT寄存器的IDLEF位。
3	OREC	溢出标志清除 向该位写1清除USART_STAT寄存器的ORERR位。
2	NEC	噪声检测清除 向该位写1清除USART_STAT寄存器的NERR位。
1	FEC	帧格式错误标志清除 向该位写1清除USART_STAT寄存器的FERR位。
0	PEC	校验错误标志清除 向该位写1清除USART_STAT寄存器的PERR位。

19.4.10. USART 数据接收寄存器（USART_RDATA）

地址偏移：0x24
复位值：未定义

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							RDATA[8:0]								
							r								

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	RDATA[8:0]	接收数据的值 包含接收到的数据字节 如果接收到的数据打开了奇偶校验位（USART_CTL0寄存器的PCEN置1），那么接收到的数据的最高位（第7位或8位，取决于数据的长度）是奇偶校验位。

19.4.11. USART 数据发送寄存器（USART_TDATA）

地址偏移：0x28
复位值：未定义

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							TDATA[8:0]								
							rw								

位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	TDATA[8:0]	发送数据的值 包含发送的数据字节 如果发送到的数据打开了奇偶校验位（USART_CTL0寄存器的PCEN置1），那么发送的数据的最高位（第7位或8位取决于数据的长度）将会被奇偶校验位替代。 只有当USART_STAT寄存器的TBE位被置位时，这个寄存器才可以改写。

19.4.12. USART 兼容性控制寄存器（USART_CHC）

地址偏移：0xC0
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							EPERR	保留							HCM
							rc_w0								rw

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	EPERR	校验错误超前检测标志。 在RBNE置位前，校验位被检测到时该标志置位。 软件写0可以清除该位。 0：没有检测到校验错误 1：检测到校验错误
7:1	保留	必须保持复位值。
0	HCM	硬件流控制兼容性模式 0：nRTS信号等于RBNE状态寄存器 1：当最后一个数据位（PCE置位时的奇偶位）被采样时，nRTS信号置位

19.4.13. USART 接收 FIFO 控制和状态寄存器（USART_RFCS）

地址偏移：0xD0

复位值：0x0000 0400

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFFINT	RFCNT[2:0]			RFF	RFE	RFFIE	RFEN	保留							ELNACK
r_w0	r			r	r	rw	rw								rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	RFFINT	接收FIFO满中断标志
14:12	RFCNT[2:0]	接收FIFO计数值
11	RFF	接收FIFO满标志 0：接收FIFO不为满 1：接收FIFO满
10	RFE	接收FIFO空标志 0：接收FIFO不为空 1：接收FIFO空
9	RFFIE	接收FIFO满中断使能

		0：禁止接收FIFO满中断 1：使能接收FIFO满中断
8	RFEN	接收FIFO使能 当UESM=1，该位置位。 0：禁止使用接收FIFO 1：使能接收FIFO
7:1	保留	必须保持复位值。
0	ELNACK	若选择了智能卡模式，提前NACK 如果检测到校验位错误，NACK脉冲提前1/16位的时间。 0：若选择了智能卡模式，禁止提前NACK 1：若选择了智能卡模式，使能提前NACK 在UART3和UART4中，该位保留。

20. 低功耗通用异步收发器（LPUART）

20.1. 简介

低功耗通用异步收发器（LPUART）提供了一个低功耗的灵活方便的串行数据交换接口。即使功耗很低，LPUART也可以执行异步串行通信。数据帧可以通过全双工或半双工，异步的方式进行传输。提供了可编程的波特率发生器，能对LPUCLK（PCLK1，CK_SYS，LTXAL或IRC16M）时钟进行分频产生LPUART发送和接收所需的特定宽范围波特率时钟。

LPUART不仅支持标准的异步收发模式，还实现了半双工串行数据交换模式。它还支持多处理器通信和硬件流控操作（CTS/RTS）。数据帧支持从LSB或者MSB开始传输。数据位的极性和TX/RX引脚都可以灵活配置。

LPUART支持DMA功能，以实现高速率的数据通信。

20.2. 主要特征

- NRZ标准格式；
- 全双工异步通信；
- 半双工单线通信；
- 双时钟域：
  - 互为异步关系的PCLK和独立于PCLK时钟的LPUART时钟；
  - 不依赖LPUCLK设置的波特率设置；
- 使用LXTAL时钟，可编程产生300-9600波特率；
- 可编程的波特率产生器，当时钟频率为32MHz，最高速度可达10MBits/s；
- 完全可编程的串口特性：
  - 数据位（7或8或9位）低位或高位在前；
  - 偶校验位，奇校验位，无校验位的生成或检测；
  - 产生1或2个停止位；
- 可互换的Tx/Rx引脚；
- 可配置的数据极性；
- 支持硬件Modem流控操作（CTS/RTS）和RS485驱动使能；
- 借助集中式DMA，实现可配置的多级缓存通信；
- 发送器和接收器可分别使能；
- 奇偶校验位控制：
  - 发送奇偶校验位；
  - 检测接收的数据字节的奇偶校验位；
- 多处理器通信：
  - 如果地址不匹配，则进入静默模式；
  - 通过线路空闲检测或者地址匹配检测从静默模式唤醒；
- 从深度睡眠模式唤醒：
  - 通过标准的RBNE中断；
  - 通过WUF中断。

- 多种状态标志：
  - 传输检测标志：接收缓冲区不为空（RBNE），发送缓冲区为空（TBE），传输完成（TC）；
  - 错误检测标志：过载错误（ORERR），噪声错误（NERR），帧格式错误（FERR），奇偶校验错误（PERR）；
  - 硬件流控操作标志：CTS变化（CTSF）；
  - 多处理器通信模式标志：IDLE帧检测（IDLEF）；
  - 从深度睡眠模式唤醒标志（WUF）；
  - 若相应的中断使能，这些事件发生将会触发中断；

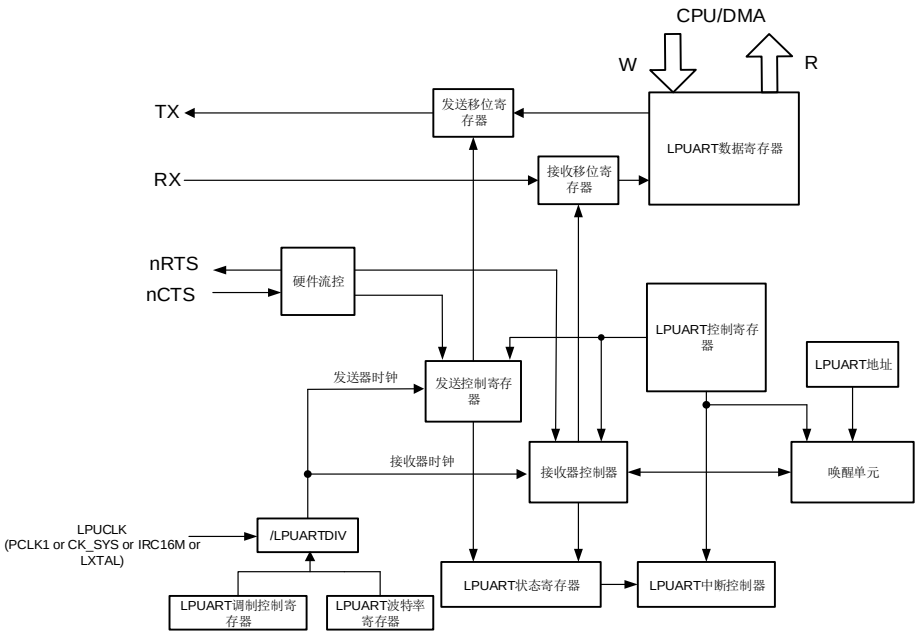
20.3. 功能说明

LPUART 接口通过*表 19-1. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

表 20-1. LPUART 重要引脚描述

引脚	类型	描述
RX	输入	接收数据
TX	输出 I/O（单线模式）	发送数据。当 LPUART 使能后，若无数据发送，默认为高电平
nCTS	输入	硬件流控模式发送使能信号
nRTS	输出	硬件流控模式发送请求信号

图 20-1. LPUART 模块内部框图

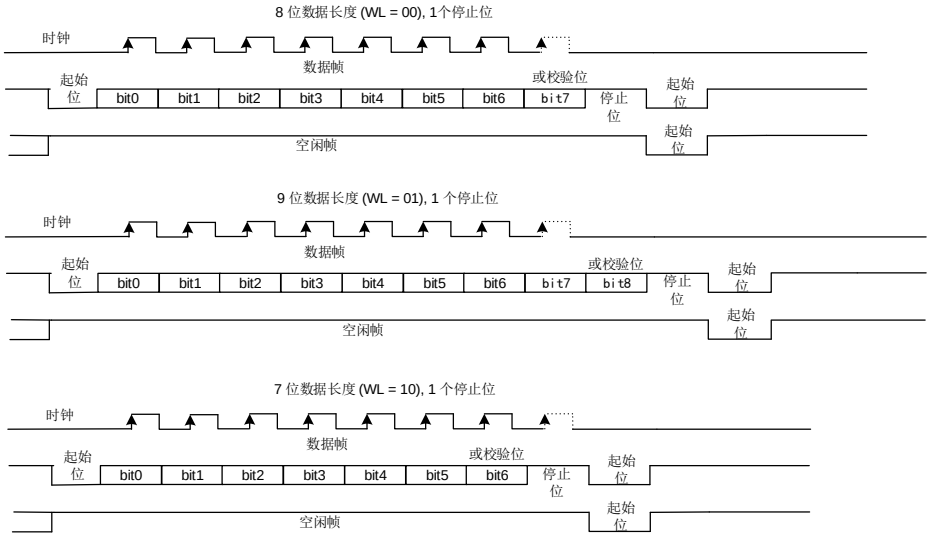


20.3.1. LPUART 帧格式

LPUART数据帧开始于起始位，结束于停止位。LPUART_CTL0寄存器中WL[1:0]位可以设置数

据长度，参考*图20-2. LPUART字符帧*。将LPUART_CTL0寄存器中PCEN置位，最后一个数据位可以用作校验位。LPUART_CTL0寄存器中PM位用于选择校验位的计算方法。

图20-2. LPUART字符帧



在发送和接收中，停止位可以在LPUART_CTL1寄存器中STB[1:0]位域中配置：

- STB[1:0] = 00：1 个停止位
- STB[1:0] = 10：2 个停止位

在一个空闲帧中，所有位都为1。数据帧长度与正常LPUART数据帧长度相同。

20.3.2. 波特率发生

波特率分频系数是一个20位的数值，波特率发生器使用该数值来确定波特率。波特率分频系数（LPUARTDIV）与LPUCLK具有如下关系：

$$\text{LPUARTDIV} = \frac{256 \times \text{LPUCLK}}{\text{Baud Rate}} \quad (20\text{-}1)$$

其中：

- LPUARTDIV：波特率分频系数，在 LPUART_BAUD 寄存器中定义

注意：

1. LPUART_BAUD[19:0]中的值必须大于 0x300。
2. （3x 波特率） ≤ LPUCLK ≤ （4096x 波特率）。
3. 在通信期间 LPUART_BAUD 寄存器的值不能被改动。

20.3.3. LPUART 发送器

如果LPUART_CTL0寄存器的发送使能位（TEN）被置位，当发送数据缓冲区不为空时，发送器将会通过TX引脚发送数据帧。TX引脚的极性可以通过LPUART_CTL1寄存器中TINV位来配

置。

TEN置位后发送器会发出一个空闲帧。TEN位在数据发送过程中是不可以被复位的。

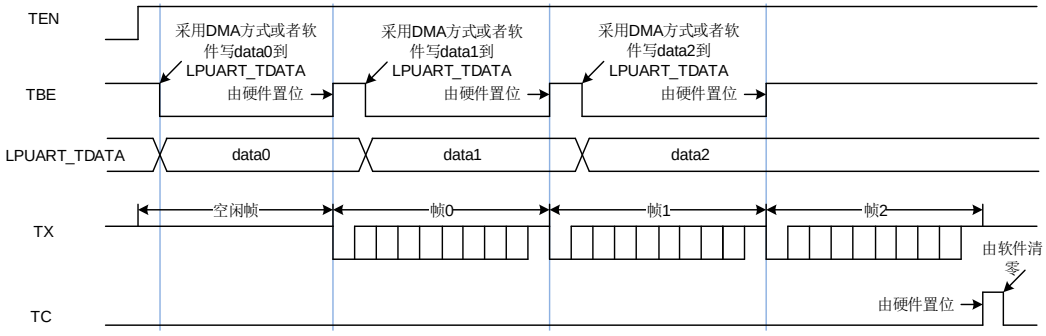
系统上电后，TBE默认为高电平。在LPUART_STAT寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入LPUART_TDATA寄存器。当数据写入LPUART_TDATA寄存器，TBE位将被清0。在数据由LPUART_TDATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入LPUART_TDATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入LPUART_TDATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄存器。

假如一帧数据已经被发送出去，并且TBE位已被置位，那么LPUART_STAT寄存器中TC位将被置1。如果LPUART_CTL0寄存器中的中断使能位（TCIE）为1，将会产生中断。

*图 19-3. USART 发送步骤*给出了 LPUART 发送步骤。软件操作按以下流程进行：

1. 通过LPUART_CTL0寄存器的WL[1:0]设置字长；
2. 在LPUART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在LPUART_CTL2寄存器中使能DMA（DENT位）；
4. 在LPUART_BAUD寄存器中设置波特率；
5. 在LPUART_CTL0寄存器中置位UEN位，使能LPUART；
6. 在LPUART_CTL0寄存器中设置TEN位；
7. 等待TBE置位；
8. 向LPUART_TDATA寄存器写数据；
9. 若DMA未使能，每发送一个字节都需重复步骤7-8；
10. 等待TC=1，发送完成。

图 20-3. LPUART 发送步骤



在禁用LPUART或进入低功耗状态之前，必须等待TC置位。通过将LPUART_INTC寄存器的TCC位置1可以将TC位清零。

20.3.4. LPUART 接收器

上电后，按以下步骤使能LPUART接收器：

1. 写LPUART_CTL0寄存器的WL[1:0]位去设置字长；
2. 在LPUART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在LPUART_CTL2寄存器中使能DMA（DENR位）；
4. 在LPUART_BAUD寄存器中设置波特率；

5. 在LPUART_CTL0寄存器中置位UEN位，使能LPUART；
6. 在LPUART_CTL0中设置REN位。

接收器在使能后，RX线上有一个下降沿发生则检测到起始位，然后在起始位的中间进行采样以确认电平是否仍为0。若起始位采样为1，则噪声错误标志（NERR）置位，该起始位被忽略，如果设置了LPUART_CTL2寄存器中的ERRIE位则会产生一个中断。接收器在检测到一个有效的起始脉冲便开始接收码流。接收器在数据位的中间进行一次采样来评估该数据位的值，数据位的采样没有噪声检测。

当接收到一个数据帧，LPUART_STAT寄存器中的RBNE置位，如果设置了LPUART_CTL0寄存器中相应的中断使能位RBNEIE，将会产生中断。在LPUART_STAT寄存器中可以观察接收状态标志。

软件可以通过读LPUART_RDATA寄存器或者DMA方式获取接收到的数据。不管是直接读寄存器还是通过DMA，只要是对LPUART_RDATA寄存器的一个读操作都可以清除RBNE位。

在接收过程中，需使能REN位，不然当前的数据帧将会丢失。

通过置位LPUART_CTL0寄存器中的PCEN位使能奇偶校验功能，接收器在接收一个数据帧时计算预期奇偶校验值，并将其与接收到的奇偶校验位进行比较。如果不相等，LPUART_STAT寄存器中PERR被置位。如果设置了LPUART_CTL0寄存器中的PERRIE位，将产生中断。

如果在停止位传输过程中RX引脚为0，将产生帧错误，LPUART_STAT寄存器中FERR将置位。如果设置了LPUART_CTL2寄存器中的ERRIE位，将产生一个中断。当被配置为1个停止位时，在停止位的中间进行采样。当被配置为2个停止位时，在第二个停止位的中间进行采样，第一个停止位不检测帧错误。

当接收到一帧数据，而RBNE位还没有被清零，随后的数据帧将不会存储在数据接收缓冲区中。LPUART_STAT寄存器中的溢出错误标志位ORERR将置位。如果设置了LPUART_CTL2寄存器中ERRIE位或者RBNEIE位，将产生中断。

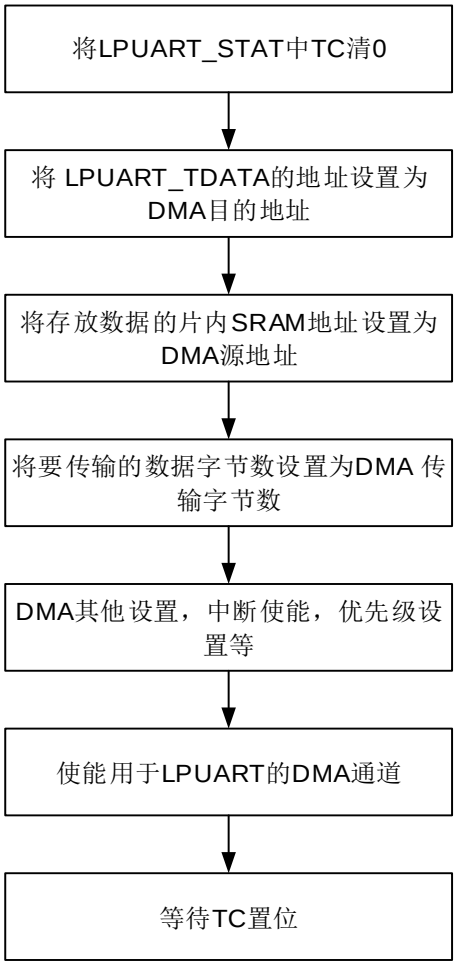
在一个接收过程中，NERR、PERR、FERR、ORERR总是分别和RBNE同时置位。如果没有使能DMA，软件需检查RBNE中断是否由NERR、PERR、FERR或者ORERR置位产生。

20.3.5. DMA 方式访问数据缓冲区

为减轻处理器的负担，可以采用DMA访问发送缓冲区或者接收缓冲区。置位LPUART_CTL2寄存器中DENT位可以使能DMA发送，置位LPUART_CTL2寄存器中DENR位可以使能DMA接收。

当 DMA 用于 LPUART 发送时，DMA 将数据从片内 SRAM 传送到 LPUART 的数据缓冲区。配置步骤如*图 19-5. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

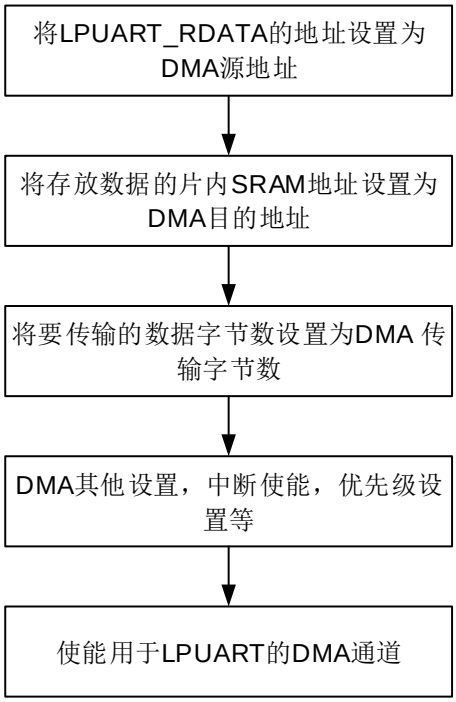
图 20-4. 采用 DMA 方式实现 LPUART 数据发送配置步骤



所有数据帧都传输完成后，LPUART_STAT寄存器中TC位置1。如果LPUART_CTL0寄存器中TCIE置位，将产生中断。

当 DMA 用于 LPUART 接收时，DMA 将数据从接收缓冲区传送到片内 SRAM。配置步骤如*图 19-6. 采用 DMA 方式实现 USART 数据接收配置步骤*所示。如果将 LPUART_CTL2 寄存器中 ERRIE 位置 1，LPUART_STAT 寄存器中的错误标志位（FERR、ORERR 和 NERR）置位时将产生中断。

图 20-5. 采用 DMA 方式实现 LPUART 数据接收配置步骤

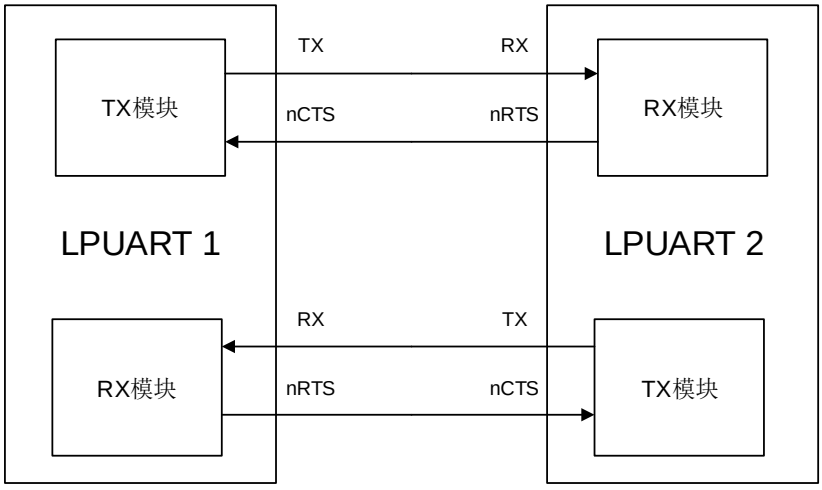


当LPUART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

20.3.6. 硬件流控制

硬件流控制功能通过nCTS和nRTS引脚来实现。通过将LPUART_CTL2寄存器中RTSEN位置1来使能RTS流控，将LPUART_CTL2寄存器中CTSEN位置1来使能CTS流控。

图 20-6. 两个 LPUART 之间的硬件流控制



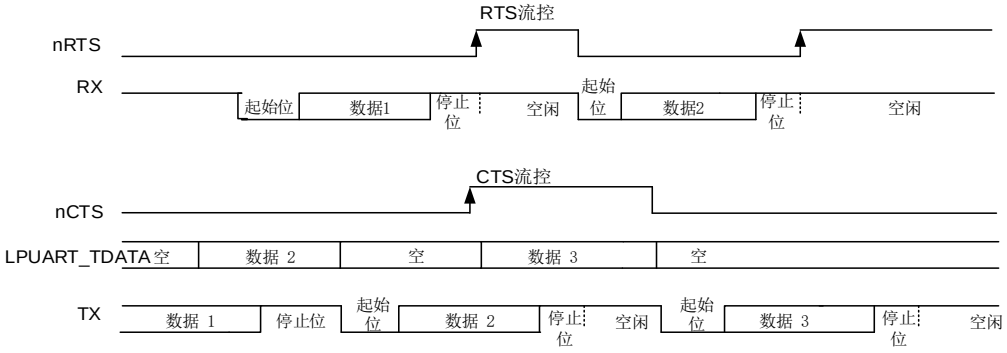
RTS 流控

LPUART接收器输出nRTS，它用于反映接收缓冲区状态。当一帧数据接收完成，nRTS变成高电平，这样是为了阻止发送器继续发送下一帧数据。当接收缓冲区满时，nRTS保持高电平。

CTS 流控

LPUART发送器监视nCTS输入引脚来决定数据帧是否可以发送。如果LPUART_STAT寄存器中TBE位是0且nCTS为低电平，发送器发送数据帧。在发送期间，若nCTS信号变为高电平，发送器将会在当前数据帧发送完成后停止发送。

图 20-7. 硬件流控制



RS485 驱动使能

驱动使能功能通过设置LPUART_CTL2控制寄存器的DEM位来打开。它允许用户通过DE（Driver Enable）信号激活外部收发器控制。提前时间是驱动使能信号和第一个字节的起始位之间的时间间隔。这个时间可以在LPUART_CTL0控制器的DEA[4:0]位域中进行设置。滞后时间是一个发送信息最后一个字节的停止位与释放DE信号之间的时间间隔。这个时间可以在LPUART_CTL0控制寄存器的DED[4:0]位域中进行设置。DE信号的极性可以通过LPUART_CTL2控制寄存器的DEP位进行设置。

LPUART的DEA和DED通过LPUCLK（f_{ck}）表示，如*表20-2 驱动使能提前时间和滞后时间*所示：

表 20-2 驱动使能提前时间和滞后时间

BRR[14:11]	驱动使能提前时间	驱动使能滞后时间
BRR[14:11] = 0	（1+DEA）×f_{ck}	（1+DED）×f_{ck}
BRR[14:11] ≠ 0	（1+（DEA×BRR[14:11]））×f_{ck}	（1+（DED×BRR[14:11]））×f_{ck}

20.3.7. 多处理器通信

在多处理器通信中，多个LPUART被连接成一个网络。对于一个设备来说，监视所有来自RX引脚的消息，是一种巨大的负担。为减轻设备负担，软件可以通过将LPUART_CMD寄存器中MMCMD位置1使LPUART进入静默模式。

如果LPUART处于静默模式，所有的接收状态标志位将不会被置位。此外，LPUART可以由硬件用以下两种方式中的一种来唤醒：空闲总线检测和地址匹配检测。

设备默认使用空闲总线检测方法唤醒LPUART。如果RWU位为0，RX引脚检测到空闲帧，LPUART_STAT寄存器中的IDLEF位会置位。如果RWU位置位，RX引脚检测到空闲帧时，硬件会将RWU清零，从而退出静默模式，当它是被空闲帧唤醒时，LPUART_STAT寄存器中IDLEF位不会被置1。

当LPUART_CTL0寄存器中WM被置位，数据最高位会被认为是地址标志位。如果地址标志位为1，该字节被认为是地址字节。如果地址标志位是0，该字节被认为是数据字节。如果地址字节的低4位或低7位（通过LPUART_CTL1寄存器中ADDM位配置）与LPUART_CTL1寄存器中的ADDR位相同，硬件会将RWU清零，并退出静默模式。接收到将LPUART唤醒的数据帧，RBNE将置位。状态标志可以从LPUART_STAT寄存器中获取。如果地址字节的低4位或低7位与LPUART_CTL1寄存器中的ADDR位不相同，硬件会置位RWU并自动进入静默模式。在这种情况下，RBNE不会被置位。

如果LPUART_CTL0寄存器中PCEN位被置位，地址字节最高位被视为校验位，其余位被视为地址位。如果ADDM位被置位，且接收帧为7位的数据，其中最低的6位将与ADDR[5:0]比较。如果ADDM位被置位，且接收帧为9位的数据，其中低8位将与ADDR[7:0]进行比较。

注意：如果设置了MEN，WM位和RWU位被复位，RX引脚检测到空闲帧，IDLEF位会置位。如果RWU位置位，则IDLEF位不会置位。

20.3.8. 半双工通信模式

通过设置LPUART_CTL2寄存器的HDEN位，可以使能半双工模式。

半双工模式下仅用单线通信。TX引脚和RX引脚从内部连接到一起，TX引脚应被配置为IO管脚。通信冲突应由软件处理。当TEN被置位时，在数据寄存器中的数据将会被发送。

20.3.9. 从深度睡眠模式唤醒

通过标准RBNE中断或WUM中断LPUART能从深度睡眠模式唤醒MCU。

UESM位必须置1并且LPUART时钟必须设置为IRC16M或LXTAL（请参考*配置寄存器 2（RCU_CFG2）*）。当LPUART的时钟源被配置为IRC16M或LXTAL时，通过将LPUART_CTL2寄存器中的UCESM位置1，可以在深度睡眠模式下保持启用该时钟。

当使用RBNE标准中断时，必须在进入深度睡眠模式前将RBNEIE位置位。

当使用WUIE中断时，WUIE中断源可以通过WUM位来选择。

在进入深度睡眠模式前，必须禁用DMA。在进入深度睡眠模式前，软件必须检测LPUART是否正在传送数据。这可以通过LPUART_STAT寄存器中的BSY标志来判断。REA位必须被检测以确保LPUART是使能的。

当检测到唤醒事件时，无论MCU工作在深度睡眠模式还是正常模式，WUF标志位通过硬件被置1，并且在WUIE被置位的情况下，触发一个唤醒中断。

20.3.10. LPUART 中断

LPUART 中断事件和标志如*表 19-3. USART 中断请求*所示：

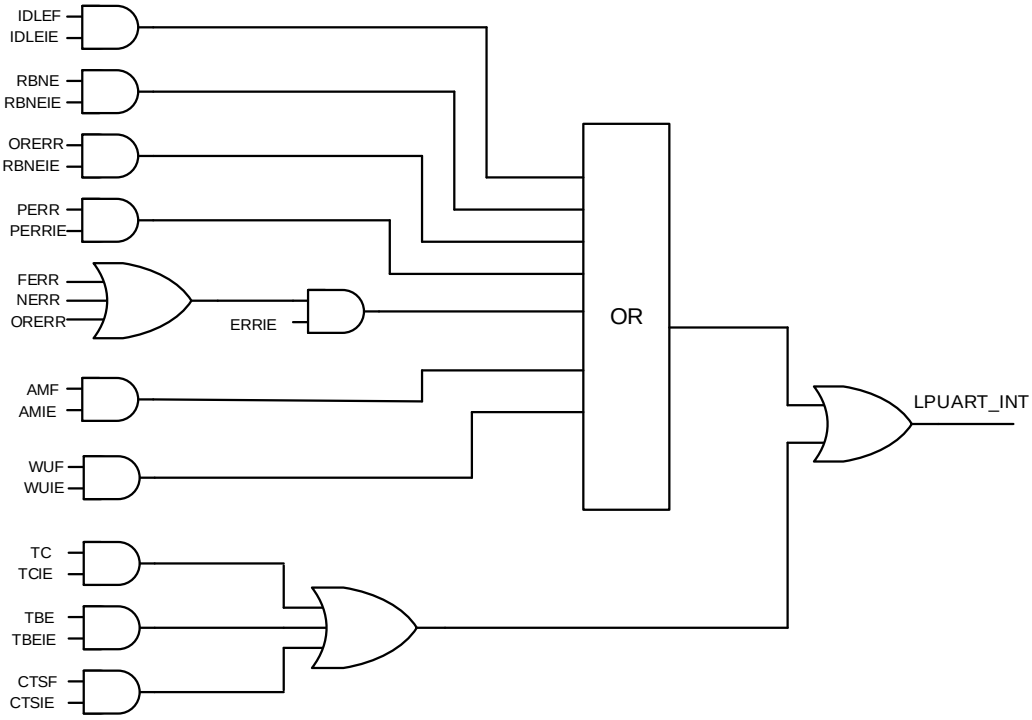
表 20-3. LPUART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器空	TBE	TBEIE
CTS标志	CTSF	CTSIE

中断事件	事件标志	使能控制位
发送结束	TC	TCIE
接收到的数据可以读取	RBNE	RBNEIE
检测到过载错误	ORERR	
检测到线路空闲	IDLEF	IDLEIE
奇偶校验错误	PERR	PERRIE
接收错误(噪声错误、溢出错误、帧错误)	NERR或ORERR或FERR	ERRIE
字符匹配	AMF	AMIE
从深度睡眠模式唤醒	WUF	WUIE

在发送给中断控制器之前，所有的中断事件是逻辑或的关系。因此在任何时候 LPUART 只能向控制器产生一个中断请求。不过软件可以在一个中断服务程序里处理多个中断事件。

图 20-8. LPUART 中断映射框图



20.4. LPUART 寄存器

LPUART基地址：0x4000 8000

20.4.1. LPUART 控制寄存器 0（LPUART_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			WL1	保留		DEA[4:0]					DED[4:0]				
			rw			rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	AMIE	MEN	WL0	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	UESM	UEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	WL1	字长 该位与WL0位决定字长 WL[1:0] = 00，8 数据位 WL[1:0] = 01，9 数据位 WL[1:0] = 10，7 数据位 WL[1:0] = 11，7 数据位 当LPUART被使能（UEN=1）时，该位域不能被改写。
27:26	保留	必须保持复位值。
25:21	DEA[4:0]	驱动使能置位时间 这些数字用来定义DE（驱动使能）信号的置位与第一个字节的起始位之间的时间间隔。它通过LPUART CLK来表示。 当LPUART被使能（UEN=1）时，该位域不能被改写。
20:16	DED[4:0]	驱动使能置低时间 这些位用来定义一个发送信息最后一个字节的停止位与置低DE（驱动使能）信号之间的时间间隔。它通过LPUART CLK来表示。 当LPUART被使能（UEN=1）时，该位域不能被改写。
15	保留	必须保持复位值。
14	AMIE	ADDR字符匹配中断使能 0：ADDR字符匹配中断禁用 1：ADDR字符匹配中断使能
13	MEN	静默模式使能 0：静默模式禁用

		1：静默模式被使能
12	WL0	字长 该位与WL1位决定字长 WL[1:0] = 00，8 数据位 WL[1:0] = 01，9 数据位 WL[1:0] = 10，7 数据位 WL[1:0] = 11，7 数据位 当LPUART被使能（UEN=1）时，该位域不能被改写。
11	WM	从静默模式唤醒方法 0：空闲线 1：地址标记 当LPUART被使能（UEN=1）时，该位域不能被改写。
10	PCEN	校验控制使能 0：校验控制禁用 1：校验控制被使能 当LPUART被使能（UEN=1）时，该位域不能被改写。
9	PM	校验模式 0：偶校验 1：奇校验 当LPUART被使能（UEN=1）时，该位域不能被改写。
8	PERRIE	校验错误中断使能 0：校验错误中断禁用 1：当LPUART_STAT寄存器的PERR位置位时，将触发中断。
7	TBEIE	发送寄存器空中断使能 0：中断禁止 1：当LPUART_STAT寄存器的TBE位置位时，将触发中断。
6	TCIE	发送完成中断使能 如果该位置1，LPUART_STAT寄存器中TC被置位时产生中断。 0：发送完成中断禁用 1：发送完成中断使能
5	RBNEIE	读数据缓冲区非空中断和过载错误中断使能 0：读数据缓冲区非空中断和过载错误中断禁用 1：当LPUART_STAT寄存器的ORERR或RBNE位置位时，将触发中断。
4	IDLEIE	IDLE线检测中断使能 0：IDLE线检测中断禁用 1：当LPUART_STAT寄存器的IDLEF位置位时，将触发中断。
3	TEN	发送器使能 0：发送器关闭 1：发送器打开

2	REN	接收器使能 0：接收器关闭 1：接收器打开并且开始搜索起始位。
1	UESM	LPUART在深度睡眠模式下使能 0：LPUART不能从深度睡眠模式唤醒MCU 1：LPUART能从深度睡眠模式唤醒MCU。条件是LPUART的时钟源必须是IRC16M或LXTAL。
0	UEN	LPUART使能 0：LPUART预分频器和输出禁用 1：LPUART预分频器和输出被使能

20.4.2. LPUART 控制寄存器 1（LPUART_CTL1）

地址偏移：0x04
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[7:0]								保留				MSBF	DINV	TINV	RINV
rw												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRP	保留	STB[1:0]		保留							ADDM	保留			
rw		rw									rw				

位/位域	名称	描述
31:24	ADDR[7:0]	LPUART的节点地址 这些位给出LPUART的节点地址。 在多处理器通信并且静默模式或者深度睡眠模式期间，这些位用来唤醒进行地址标记的检测。接收到的最高位为1的数据帧将和这些位进行比较。当ADDM位被清零时，仅仅ADDR[3:0]被用来比较。 在正常的接收期间，这些位也用来进行字符检测。所有接收到的字符（8位）与ADDR[7:0]的值进行比较，如果匹配，AMF标志将被置位。 当接收器（REN=1）和LPUART（UEN=1）被使能时，该位域不能被改写。
23:20	保留	必须保持复位值
19	MSBF	高位在前 0：数据发送/接收，采用低位在前 1：数据发送/接收，采用高位在前 LPUART被使能（UEN=1）时，该位域不能被改写。
18	DINV	数据位反转 0：数据位信号值没有反转 1：数据位信号值被反转

		LPUART被使能（UEN=1）时，该位域不能被改写。
17	TINV	TX管脚电平反转 0：TX管脚信号值没有反转 1：TX管脚信号值被反转. LPUART被使能（UEN=1）时，该位域不能被改写。
16	RINV	RX管脚电平反转 0：RX管脚信号值没有反转. 1：RX管脚信号值被反转 LPUART被使能（UEN=1）时，该位域不能被改写。
15	STRP	交换TX/RX管脚 0：TX和RX管脚功能不被交换 1：TX和RX管脚功能被交换 当LPUART被使能（UEN=1）时，该位域不能改写。
14	保留	必须保持复位值
13:12	STB[1:0]	STOP位长 00：1停止位 01：1停止位 10：2停止位 11：2停止位 LPUART被使能（UEN=1）时，该位域不能被改写。
11:5	保留	必须保持复位值
4	ADDM	地址检测模式 该位用来选择4位地址检测或全位地址检测。 0：4位地址检测 1：全位地址检测。在7位，8位和9位数据模式下，地址检测分别按6位，7位和8位地址（ADDR[5:0]，ADDR[6:0]和ADDR[7:0]）执行。 LPUART被使能（UEN=1）时，该位域不能被改写。
3:0	保留	必须保持复位值

20.4.3. LPUART 控制寄存器 2（LPUART_CTL2）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								UCESM	WUIE	WUM[1:0]		保留			
								rw	rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRD	保留	CTSIE	CTSEN	RTSEN	DENT	DENR	保留		HDEN	保留		ERRIE

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

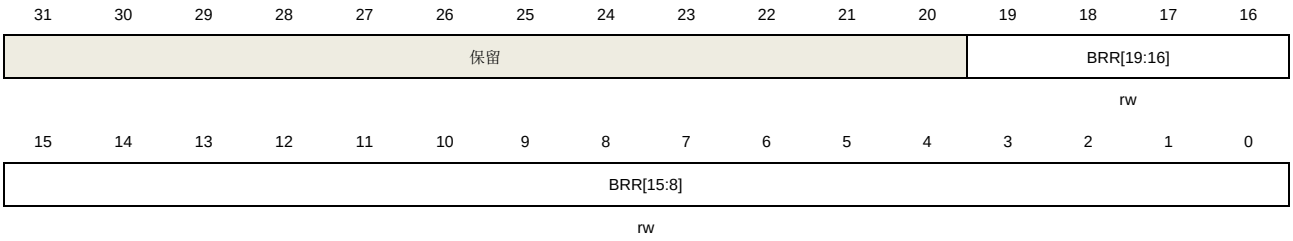
位/位域	名称	描述
31:24	保留	必须保持复位值。
23	UCESM	Deep-sleep模式下LPUART时钟使能 0：Deep-sleep模式下LPUART时钟失能 1：Deep-sleep模式下LPUART时钟使能
22	WUIE	从深度睡眠模式唤醒中断使能 0：从深度睡眠模式唤醒中断禁用 1：从深度睡眠模式唤醒中断被使能
21:20	WUM[1:0]	从深度睡眠模式唤醒模式 这个位域指定什么事件可以置位LPUART_STAT寄存器中的WUF（从深度睡眠唤醒标志）标志。 00：WUF在地址匹配的时候置位。如何实现地址匹配在ADDR和ADDM中定义。 01：保留 10：WUF在检测到起始位时置位 11：WUF在检测到RBNE时置位 LPUART被使能（UEN=1）时，该位域不能被改写。
19:16	保留	必须保持复位值。
15	DEP	驱动使能的极性选择模式 0：DE信号高有效 1：DE信号低有效 LPUART被使能（UEN=1）时，该位域不能被改写。
14	DEM	驱动使能模式 用户使能该位以后，可以通过DE信号对外部收发器进行控制。DE信号是从RTS管脚输出的。 0：DE功能禁用 1：DE功能开启 LPUART被使能（UEN=1）时，该位域不能被改写。
13	DDRE	在接收错误时禁止DMA 0：在发生接收错误的情况下，不禁用DMA。所有的错误数据不会产生DMA请求，以确保错误的数据不会被传输，但是下一个接收到的正确的数据会被传输。RBNE位保持0以阻止过载错误，但是相应错误标志位会被置位。 1：在接收错误的情况下，DMA被关闭。DMA请求会被屏蔽，直到相应的标志位被清0。RBNE标志和相应的错误标志位会被置位。软件在清除错误标志前，必须首先关DMA请求（DMAR = 0）或清RBNE。 LPUART被使能（UEN=1）时，该位域不能被改写。
12	OVRD	溢出禁止 0：溢出功能被使能。当接收到的数据在新数据到达前没有被读走，ORERR错误标志位将被置位，并且新数据将会丢失。

		1：溢出功能禁止。当接收到的数据在新数据到达前没有被读走，ORERR错误标志位将不会被置位，新数据会将LPUART_RDATA寄存器以前的内容覆盖。 LPUART被使能（UEN=1）时，该位域不能被改写。
11	保留	必须保持复位值。
10	CTSIE	CTS中断使能 0：CTS中断屏蔽 1：当LPUART_STAT的CTS位置位时，会产生中断。
9	CTSEN	CTS使能 0：CTS硬件流控禁用 1：CTS硬件流控被使能 LPUART被使能（UEN=1）时，该位域不能被改写。
8	RTSEN	RTS使能 0：RTS硬件流控禁用 1：RTS硬件流控被使能，只有当接收缓冲区有空间的时候，才会请求下一个数据。 LPUART被使能（UEN=1）时，该位域不能被改写。
7	DENT	DMA发送使能 0：关闭DMA发送模式 1：开启DMA发送模式
6	DENR	DMA接收使能 0：关闭DMA接收模式 1：开启DMA接收模式
5:4	保留	必须保持复位值。
3	HDEN	半双工使能 0：禁用半双工模式 1：开启半双工模式 LPUART被使能（UEN=1）时，该位域不能被改写。
2:1	保留	必须保持复位值。
0	ERRIE	多级缓存通信模式的错误中断使能 0：禁用错误中断 1：在多级缓存通信时，当LPUART_STAT寄存器的FERR位，ORERR位或NERR位被置位时，会产生中断。

20.4.4. LPUART 波特率寄存器（LPUART_BAUD）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。
当LPUART（UEN=1）被使能时，该寄存器不能被改写。

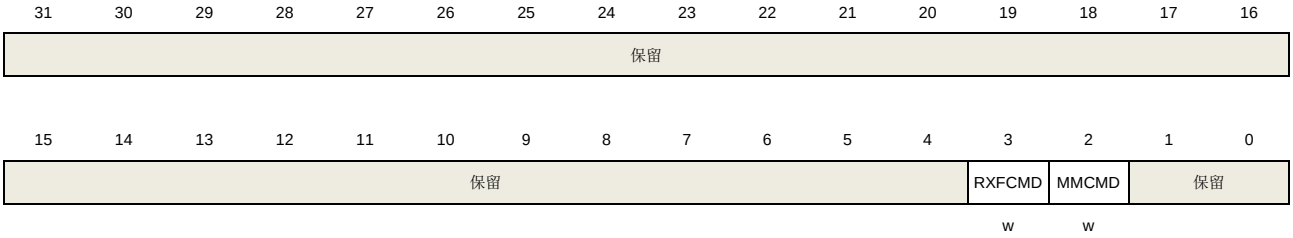


位/位域	名称	描述
31:20	保留	必须保持复位值。
19:0	BRR[19:0]	LPUARTDIV的值 **注意**：BRR[19:0] ≥ 0x300并且(3x波特率) ≤ LPUCLK ≤ (4096x波特率)。

20.4.5. LPUART 请求寄存器（LPUART_CMD）

地址偏移：0x18
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

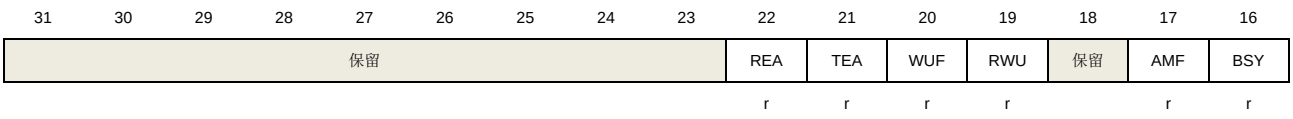


位/位域	名称	描述
31:4	保留	必须保持复位值。
3	RXFCMD	接收数据清空请求 向该位写1来清除RBNE标志位，以丢弃未读的接收数据。
2	MMCMD	静默模式请求 向该位写1使LPUART进入静默模式并且置位RWU标志位。
1:0	保留	必须保持复位值。

20.4.6. LPUART 状态寄存器（LPUART_STAT）

地址偏移：0x1C
复位值：0x0000 00C0

该寄存器只能按字（32位）访问。



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CTS	CTSF	保留	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
					r	r		r	r	r	r	r	r	r	r

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	REA	接收使能通知标志 这位反映了LPUART核心逻辑的接收使能状态，该位可以通过硬件设置。 0：LPUART核心接收逻辑禁用 1：LPUART核心接收逻辑被使能
21	TEA	发送使能通知标志 该位反映了LPUART核心逻辑的发送使能状态，该位可以通过硬件设置。 0：LPUART核心发送逻辑禁用 1：LPUART核心发送逻辑被使能
20	WUF	从深度睡眠模式唤醒标志 0：没有从深度睡眠模式唤醒 1：已从深度睡眠模式唤醒，如果在LPUART_CTL2寄存器的WUFIE=1并且MCU处于深度睡眠模式，将引发一个中断。 当检测到一个唤醒事件时，该位通过硬件置位，这个事件在WUM位域被定义。 向LPUART_INTC寄存器中的WUC写1，该位被清0。 当UESM被清0时，该位清0。
19	RWU	接收器从静默模式唤醒 该位表示LPUART处于静默模式。 0：接收器在工作状态 1：接收器在静默状态 当在唤醒和静默模式切换时，它通过硬件清0或者置1。静默模式控制（地址帧还是空闲帧）是用通过LPUART_CTL0寄存器的WM位选择。 如果选择空闲信号唤醒，只能通过向LPUART_CMD寄存器的MMCMD位写1来将该位置位。
18	保留	必须保持复位值。
17	AMF	ADDR匹配标志 0：ADDR和接收到的字符不匹配 1：ADDR和接收到的字符匹配，如果LPUART_CTL0寄存器的AMIE=1，将引发一个中断。 当接收到ADDR[7:0]中定义的字符时，硬件置位。 通过向LPUART_INTC寄存器的AMC写1清0。
16	BSY	忙标志 0：LPUART处于空闲 1：LPUART正在接收
15:11	保留	必须保持复位值。

10	CTS	CTS电平 这个值等于nCTS输入引脚电平的反向拷贝。 0：nCTS输入引脚高电平 1：nCTS输入引脚低电平
9	CTSF	CTS变化标志 0：nCTS状态线没有变化 1：nCTS状态线发生变化，如果LPUART_CTL2寄存器的CTSIE位置位，将引发中断。 当nCTS输入变化时，由硬件置位。 通过向LPUART_INTC寄存器的CTSC位写1，清零该位。
8	保留	必须保持复位值
7	TBE	发送数据寄存器空 0：数据没有发送到移位寄存器 1：数据发送到移位寄存器。如果LPUART_CTL0寄存器的TBEIE位置位，将会有中断产生。 当LPUART_TDATA寄存器的内容已经被转移到移位寄存器时，由硬件置位。 通过向LPUART_TDATA寄存器中写数据来清0。
6	TC	发送完成 0：发送没有完成 1：发送完成。如果LPUART_CTL0寄存器的TCIE被置位，将会有中断产生。 如果一个包含数据的帧的发送完成且TBE被置位，该位由硬件置位。 通过向LPUART_INTC寄存器的TCC位写1清0。
5	RBNE	读数据缓冲区非空 0：没有接收到数据 1：已接收到数据并且可以读取。当寄存器LPUART_CTL0的RBNEIE位被置位，将会有中断产生。 当接收移位寄存器的内容已经被转移到寄存器LPUART_RDATA，由硬件置位。 通过读LPUART_RDATA寄存器或向LPUART_CMD寄存器的RXFCMD位写1清0。
4	IDLEF	空闲线检测标志 0：没检测到空闲线 1：检测到空闲线。如果LPUART_CTL0寄存器的IDLEIE位置1，将会有中断产生。 当检测到空闲线时，通过硬件置位。直到RBNE位置位，否则它不会被再次置位。 向LPUART_INTC寄存器的IDLEC位写1清0。
3	ORERR	溢出错误 0：未检测到溢出错误 1：检测到溢出错误。在多级缓存通信中，如果寄存器LPUART_CTL0的RBNEIE位置位，将会引发中断。如果寄存器LPUART_CTL2的ERRIE位置位也会引发中断。 在RBNE置位的情况下，如果接收移位寄存器的数据传递给LPUART_RDATA寄存器，将会由硬件置位。 向LPUART_INTC寄存器的OREC位写1清0。

2	NERR	噪声错误标志 0：未检测到噪声错误 1：检测到噪声错误。在多级缓存通信中，如果寄存器LPUART_CTL2的ERRIE位置位，将会有中断产生。 在接收帧的时候检测到噪声错误，将会由硬件置位。 向寄存器LPUART_INTC的NEC位写1清0。
1	FERR	帧错误 0：未检测到帧错误 1：检测到帧错误或者断开字符。在多级缓存通信中，如果寄存器LPUART_CTL2的ERRIE位置位，将会有中断产生。 当一个不同步，强噪声或者断开字符被检测到时，硬件置位。在智能卡模式下，当发送次数达到上限，仍然没有收到发送成功应答（卡一直响应NACKs），该位也将被置位。 向LPUART_INTC寄存器的FEC位写1清0。
0	PERR	校验错误 0：未检测到校验错误 1：检测到校验错误，在多级缓存通信中，如果寄存器LPUART_CTL0的PERRIE位置位，将会有中断产生。 当在接收模式的时候检测到校验错误，将会由硬件置位。 向LPUART_INTC寄存器的PEC位写1清0。

20.4.7. LPUART 中断标志清除寄存器（LPUART_INTC）

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											WUC	保留		AMC	保留
											w			w	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CTSC	保留		TCC	保留	IDLEC	OREC	NEC	FEC	PEC
						w			w		w	w	w	w	w

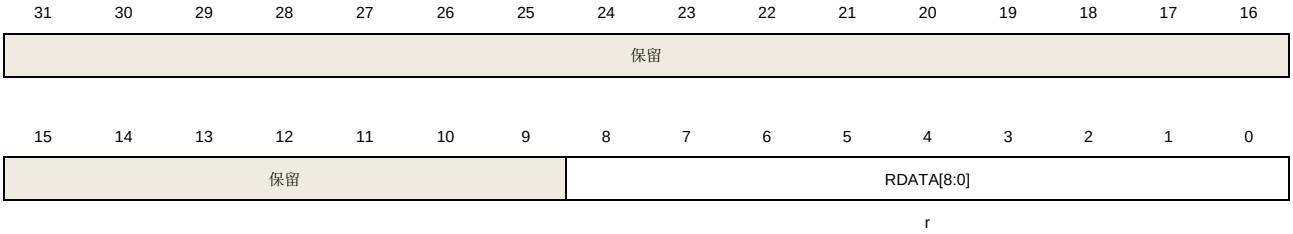
位/位域	名称	描述
31:21	保留	必须保持复位值。
20	WUC	从深度睡眠模式唤醒标志的清除 向该位写1清除LPUART_STAT寄存器的WUF位。
19:18	保留	必须保持复位值。
17	AMC	ADDR匹配标志清除 向该位写1清除LPUART_STAT寄存器的AMF位。

16:110	保留	必须保持复位值。
9	CTSC	CTS变化标志清除 向该位写1清除LPUART_STAT寄存器的CTSF位。
8:7	保留	必须保持复位值。
6	TCC	发送完成标志清除 向该位写1清除LPUART_STAT寄存器的TC位。
5	保留	必须保持复位值。
4	IDLEC	空闲线检测标志清除 向该位写1清除LPUART_STAT寄存器的IDLEF位。
3	OREC	溢出标志清除 向该位写1清除LPUART_STAT寄存器的ORERR位。
2	NEC	噪声检测清除 向该位写1清除LPUART_STAT寄存器的NERR位。
1	FEC	帧格式错误标志清除 向该位写1清除LPUART_STAT寄存器的FERR位。
0	PEC	校验错误标志清除 向该位写1清除LPUART_STAT寄存器的PERR位。

20.4.8. LPUART 数据接收寄存器（LPUART_RDATA）

地址偏移：0x24
复位值：未定义

该寄存器只能按字（32位）访问。

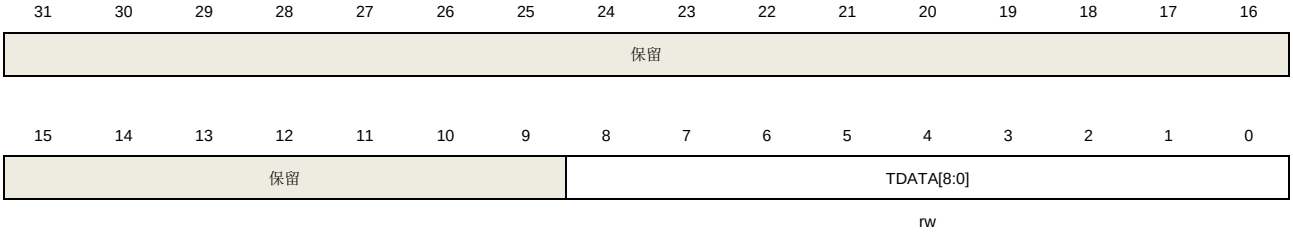


位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	RDATA[8:0]	接收数据的值 包含接收到的数据字节 如果接收到的数据打开了奇偶校验位（LPUART_CTL0寄存器的PCEN置1），那么接收到的数据的最高位（第7位或8位，取决于数据的长度）是奇偶校验位。

20.4.9. LPUART 数据发送寄存器（LPUART_TDATA）

地址偏移：0x28
复位值：未定义

该寄存器只能按字（32 位）访问。

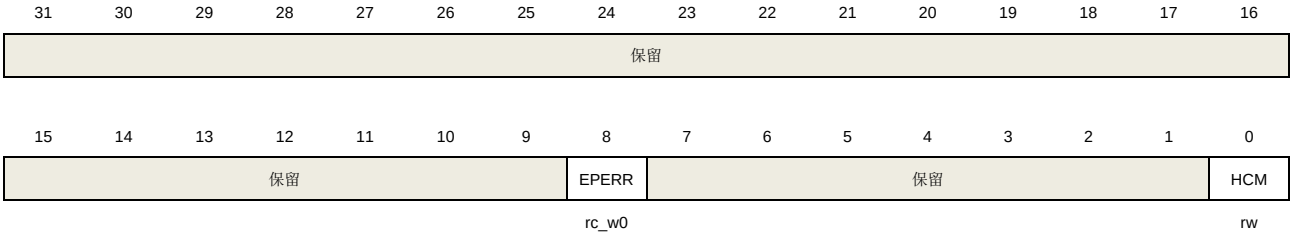


位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	TDATA[8:0]	发送数据的值 包含发送的数据字节 如果发送到的数据打开了奇偶校验位（LPUART_CTL0寄存器的PCEN置1），那么发送的数据的最高位（第7位或8位取决于数据的长度）将会被奇偶校验位替代。 只有当LPUART_STAT寄存器的TBE位被置位时，这个寄存器才可以改写。

20.4.10. LPUART 兼容性控制寄存器（LPUART_CHC）

地址偏移：0xC0
复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:9	保留	必须保持复位值。
8	EPERR	校验错误超前检测标志。 在RBNE置位前，校验位被检测到时该标志置位。 软件写0可以清除该位。 0：没有检测到校验错误 1：检测到校验错误
7:1	保留	必须保持复位值。

0	HCM	硬件流控制兼容性模式 0：nRTS信号等于RBNE状态寄存器 1：当最后一个数据位（PCE置位时的奇偶位）被采样时，nRTS信号置位

21. 内部集成电路总线接口（I2C）

21.1. 简介

I2C（内部集成电路总线）模块提供了符合工业标准的两线串行制接口，可用于 MCU 和外部 I2C 设备的通讯。I2C 总线使用两条串行线：串行数据线 SDA 和串行时钟线 SCL。

I2C 接口模块实现了 I2C 协议的标速模式，快速模式以及快速+ 模式，具备 CRC 计算和校验功能、支持 SMBus（系统管理总线）和 PMBus（电源管理总线）。此外，I2C 接口模块还支持多主机 I2C 总线架构。I2C 接口模块也支持 DMA 模式，可有效减轻 CPU 的负担。

21.2. 主要特征

- 并行总线至 I2C 总线协议的转换及接口。
- 同一接口既可实现主机功能又可实现从机功能。
- 主从机之间的双向数据传输。
- 支持 7 位和 10 位的地址模式和广播寻址。
- 多个 7 位从机地址（两个地址可配置地址位屏蔽）。
- 可编程的建立时间和保持时间。
- 支持 I2C 多主机模式。
- 支持标速（最高 100 kHz），快速（最高 400 kHz）和快速+ 模式（最高 1MHz，必须在 SYSCFG_CFG0 中使能大电流能力 IO）。
- 从机模式下可配置的 SCL 主动拉低。
- 支持 DMA 模式。
- 兼容 SMBus 3.0 和 PMBus 1.3。
- 可选择的 PEC（报文错误校验）生成和校验。
- 可编程模拟过滤器和数字过滤器。
- I2C 地址匹配时，由深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2 唤醒。
- 独立于 PCLK 的时钟。

21.3. 功能说明

I2C 接口的内部结构如*图 21-1. I2C 模块框图*所示。

图 21-1. I2C 模块框图

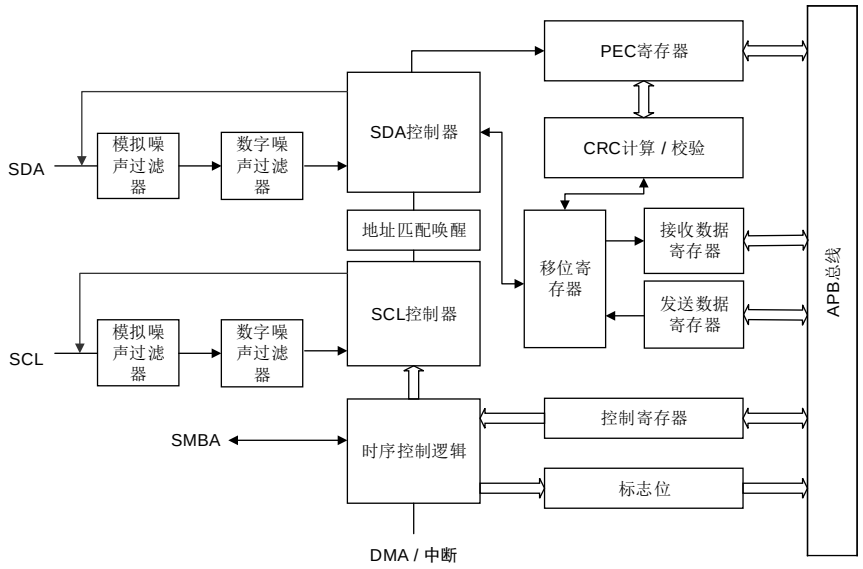


表 21-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

术语	说明
发送器	发送数据到总线的设备
接收器	从总线接收数据的设备
主机	初始化数据传输，产生时钟信号和结束数据传输的设备
从机	由主机寻址的设备
多主	不破坏信息的前提下同时控制总线的多个主机
仲裁	如果超过一个主机同时试图控制总线，只有一个主机被允许，且获胜主机的信息不被破坏，保证上述的过程叫仲裁

21.3.1. 时钟要求

I2C 时钟独立于 PCLK 时钟，因此可以独立操作 I2C。

I2C 时钟（I2CCLK）可以从以下三个时钟源中选择：

- APB1 时钟 PCLK1（默认值）
- 内部 16M RC 时钟 IRC16M
- 系统时钟 SYSCLK

I2C 时钟周期t_{I2CCLK}必须满足以下条件：

- $t_{I2CCLK} < (t_{LOW} - t_{filters})/4$
- $t_{I2CCLK} < t_{HIGH}$

其中：

t_{LOW}：SCL 低电平时间

t_{HIGH}：SCL 高电平时间

$t_{filters}$：在使能滤波器时，表示模拟滤波器和数字滤波器产生的延时总和。模拟滤波器产生

的延时最大值为 260ns，数字滤波器产生的延时为DNF[3:0]×t_{I2CCLK}。

PCLK 时钟周期t_{PCLK}必须满足以下条件：

- t_{PCLK}<4/3*t_{SCL}

其中：

t_{SCL}：SCL 周期

注意：当 I2C 内核时钟由 PCLK 提供时，PCLK 必须符合t_{I2CCLK}的条件。

21.3.2. I2C 通讯流程

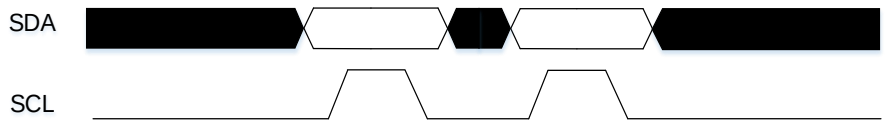
主机和从机都能实现数据收发，因此，I2C 可以实现四种工作模式：

- 从机发送
- 从机接收
- 主机发送
- 主机接收

数据有效性

时钟信号的高电平期间 SDA 线上的数据必须稳定。只有在时钟信号 SCL 变低的时候数据线 SDA 的电平状态才能跳变（如*图 21-2. 数据有效性*）。每个数据比特传输需要一个时钟脉冲。

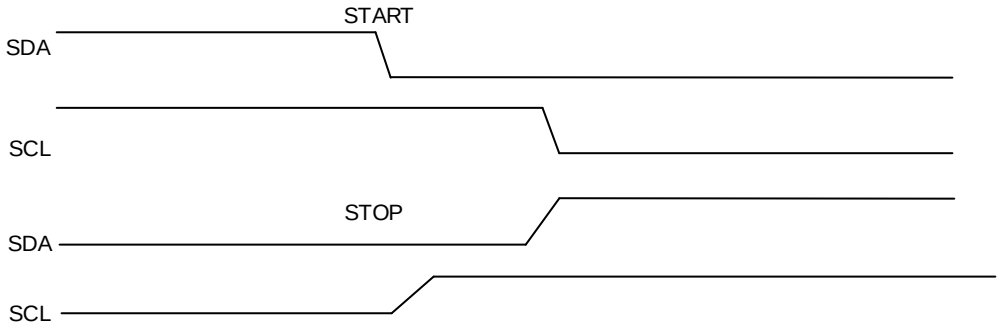
图 21-2. 数据有效性



开始和停止信号

所有的数据传输起始于一个 START 结束于一个 STOP（参见*图 21-3. 开始和停止信号*）。START 信号定义为，在 SCL 为高时，SDA 线上出现一个从高到低的电平转换。STOP 结束位定义为，在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。

图 21-3. 开始和停止信号



每个 I2C 设备（不管是微控制器，LCD 驱动，存储器或者键盘接口）都通过唯一的地址进行识别，根据设备功能，他们既可以是发送器也可作为接收器。在默认情况下，I2C 设备工作在从机模式下。当 START 信号产生时，I2C 设备由从机模式切换成主机模式。如果仲裁丢失或者 STOP 信号产生时，I2C 由主机模式切换成从机模式。支持 I2C 多主机模式。

I2C 从机检测到 I2C 总线上的 START 信号之后，就开始从总线上接收地址，之后会把从总线接收到的地址和自身的地址（通过软件编程）进行比较，当两个地址相同时，I2C 从机将发送一个确认应答（ACK），并响应总线的后续命令：发送或接收所需数据。此外，如果软件开启了广播呼叫，则 I2C 从机始终对一个广播地址（0x00）发送确认应答。I2C 模块支持 7 位和 10 位的地址模式。

数据和地址都是 8 位传输，高位在前。START 信号之后的字节（在 7 位地址模式下是一个字节，10 位地址模式下是两个字节）是主机发送的从机地址。

8 个时钟周期字节发送后，第 9 个时钟脉冲期间接收器会发送应答信号至发送器。是否产生 ACK 信号可以软件配置。

I2C 主机负责产生 START 信号和 STOP 信号来开始和结束一次传输，并且负责产生 SCL 时钟。

在主机模式下，如果 AUTOEND=1，STOP 信号由硬件产生。如果 AUTOEND=0，STOP 信号由软件产生，或者主机可以产生 RESTART 信号来启动新的数据传输。

图 21-4. 10 位地址的 I2C 通讯流程（主机发送）

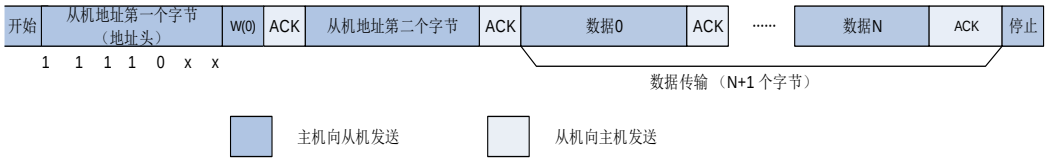


图 21-5. 7 位地址的 I2C 通讯流程（主机发送）

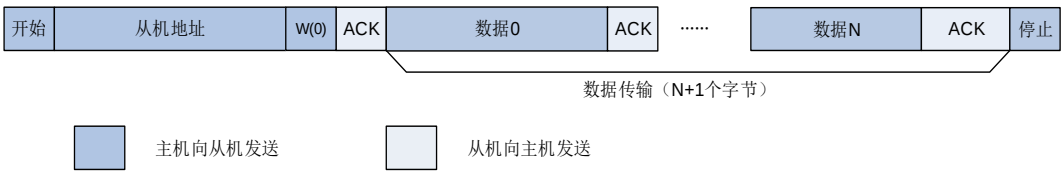
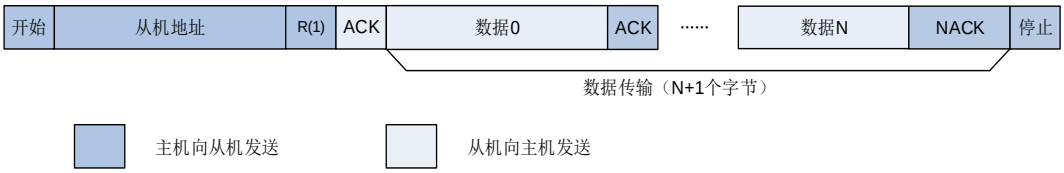


图 21-6. 7 位地址的 I2C 通讯流程（主机接收）



在 10 位寻址模式中，配置 HEAD10R 位可以选择执行完整的寻址序列或只发送地址头。当 HEAD10R=0，执行完整的 10 位地址寻址读序列 START+10 位地址头（写）+第二个地址字节 +RESTART+10 位地址头（读），如*图 21-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=0）*所示。

在10位寻址模式中，如果主机接收是在主机发送结束后执行，读寻址序列可以是RESTART+10

位地址头（读），如*图 21-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）*所示。

图 21-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=0）

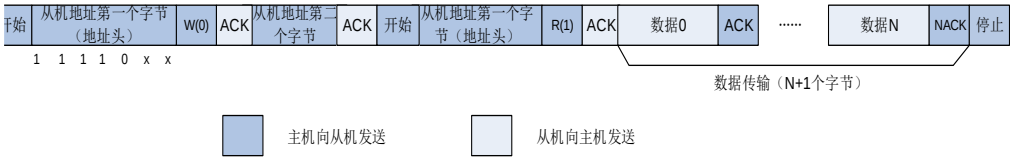
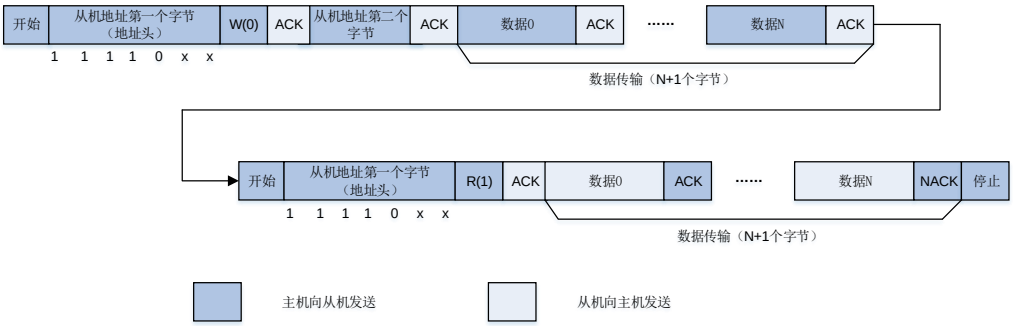


图 21-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）



21.3.3. 噪声滤波器

I2C 外设集成了模拟噪声滤波器和数字噪声滤波器，噪声滤波器可根据实际需要在 I2C 外设启用前进行配置。

将 I2C_CTL0 寄存器中 ANOFF 位置 1 可以禁用模拟噪声滤波器，将 ANOFF 位清 0 时使能模拟噪声滤波器。在快速模式和快速+ 模式下，模拟滤波器需要抑制脉冲宽度高达 50ns 的峰值。

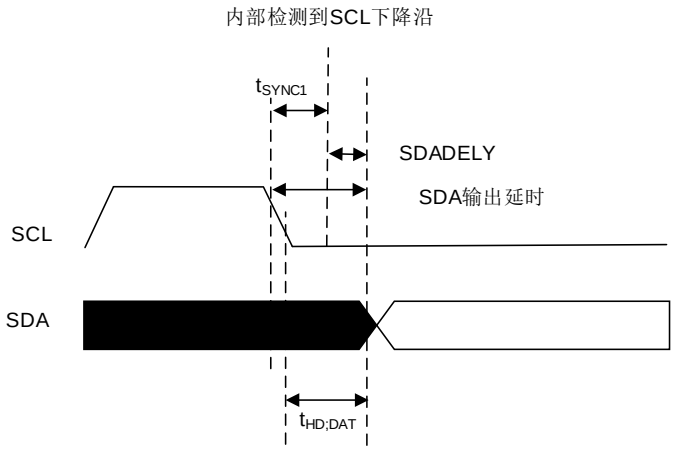
数字滤波器由 I2C_CTL0 寄存器中 DNF[3:0]位来配置。当数字滤波器使能时，SCL 和 SDA 电平保持稳定的时间大于DNF[3:0]×t_{I2CCLK} 才会发生内部变化。抑制峰值宽度可由 DNF[3:0]配置。

21.3.4. I2C 时序配置

在 I2C 通信中，I2C_TIMING 寄存器中 PSC[3:0]，SCLDELY[3:0]和 SDADELY[3:0]用于保证正确的数据保持时间和数据建立时间。

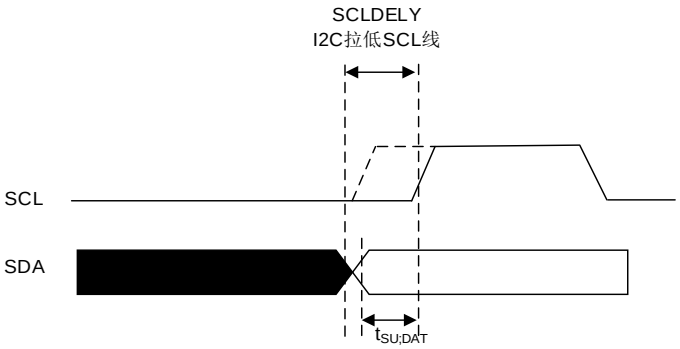
如果数据已经在 I2C_TDATA 寄存器中，在经历 SDADELY 延时后，数据由 SDA 发送，如*图 21-9. 数据保持时间*所示。

图 21-9. 数据保持时间



当数据经过 SDA 发送时，SCLDELY 计数器开启。如图*图 21-10. 数据建立时间*所示。

图 21-10. 数据建立时间



当内部检测到 SCL 下降沿时，在 SDA 发送之前会插入一个延时。该延时为 $t_{SDADELY}$=SDADELY*t_{PSC}+t_{I2CCLK}，其中t_{PSC}=(PSC+1)*t_{I2CCLK}。$t_{SDADELY}$会影响$t_{HD;DAT}$。SDA 输出总延时为t_{SYNC1}+{[SDADELY*(PSC+1)+1]*t_{I2CCLK}}。t_{SYNC1}由 SCL 下降斜率，模拟滤波器延时，数字滤波器延时和 SCL 与 I2CCLK 时钟的同步延时共同决定。SCL 与 I2CCLK 时钟的同步延时为 2 至 3 个t_{I2CCLK}。

SDADELY 必须符合以下条件：

- SDADELY≥{t_f(max)+$t_{HD;DAT}$(min)-t_{AF}(min)-[(DNF+3)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]
- SDADELY≤{$t_{HD;DAT}$(max)-t_{AF}(max)-[(DNF+4)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]

注意：t_{AF}为模拟滤波器延时，$t_{HD;DAT}$必须小于$t_{VD;DAT}$的最大值。

当 SS=0 时，经过延时$t_{SDADELY}$，在数据写入 I2C_TDATA 寄存器之前，从机会拉低时钟线。在数据建立时间期间 SCL 保持低电平。数据建立时间$t_{SCLDELY}$=(SCLDELY+1)*t_{PSC}。$t_{SCLDELY}$影响$t_{SU;DAT}$。

SCLDELY 必须符合以下条件：

- SCLDELY≥{[t_r(max)+$t_{SU;DAT}$(min)]/[(PSC+1)*t_{I2CCLK}]}-1

在主机模式下，SCL 时钟高低电平由 I2C_TIMING 寄存器中 PSC[3:0]，SCLH[7:0]和 SCLL[7:0]控制。

当内部检测到 SCL 下降沿，在释放 SCL 输出之前会插入一个延时，该延时为 $t_{SCLL}=(SCLL+1)*t_{PSC}$，其中$t_{PSC}=(PSC+1)*t_{I2CCLK}$。$t_{SCLL}$影响 SCL 低电平持续时间$t_{LOW}$。

当内部检测到 SCL 上升沿，在将 SCL 拉低之前会插入一个延时，该延时为 $t_{SCLH}=(SCLH+1)*t_{PSC}$，其中$t_{PSC}=(PSC+1)*t_{I2CCLK}$。$t_{SCLH}$影响 SCL 高电平持续时间$t_{HIGH}$。

注意：时序配置和 SS 位在 I2C 外设使能时是不能改变的。

表 21-2. 数据建立时间和数据保持时间

符号	参数	标准模式		快速模式		快速 +模式		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{HD;DAT}$	数据保持时间	0	-	0	-	0	-	0.3	-	us
$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	数据建立时间	250	-	100	-	50	-	250	-	ns
t_r	SCL 和 SDA 信号上升时间	-	1000	-	300	-	120	-	1000	
t_f	SCL 和 SDA 信号下降时间	-	300	-	300	-	120	-	300	

21.3.5. I2C 复位

清除 I2C_CTL0 寄存器中 I2CEN 位可以实现软件复位。当软件复位产生时，SCL 和 SDA 均被释放。通信控制位和状态位也还原成复位值。软件复位对配置寄存器无影响。受到影响的位为 I2C_CTL1 寄存器中 START，STOP 和 NACKEN，I2C_STAT 寄存器中 I2CBSY，TBE，TI，RBNE，ADDSEND，NACK，TCR，TC，STPDET，BERR，LOSTARB 和 OUERR。另外，如果支持 SMBus 模式，I2C_CTL1 寄存器中 PECTRANS 位，I2C_STAT 寄存器中 PECERR，TIMEOUT 和 SMBALT 位也会受到影响。

为了实现软件复位，I2CEN 必须在至少 3 个 APB 时钟周期内保持低电平。可以通过以下写软件序列来保证软件复位：

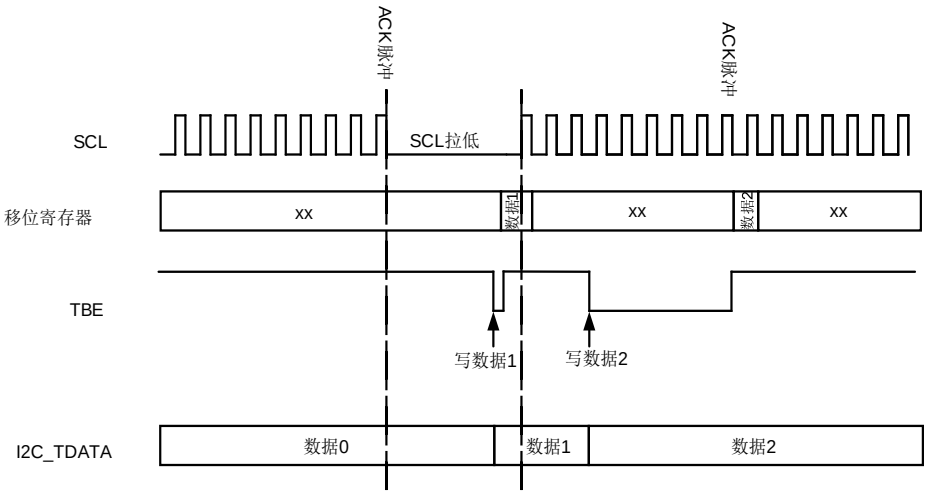
- I2CEN 写 0
- 检查 I2CEN 是否为 0
- I2CEN 写 1

21.3.6. 数据传输

数据发送

在发送数据时，如果 TBE 为 0，表明 I2C_TDATA 寄存器非空，在第九个 SCL 脉冲（应答脉冲）后，I2C_TDATA 寄存器中的数据移入到移位寄存器。移位寄存器中的数据通过 SDA 线移出。如果 TBE 为 1，则表明 I2C_TDATA 寄存器为空，在 I2C_TDATA 不为空之前 SCL 将被拉低。SCL 拉低是在第九个 SCL 脉冲之后。

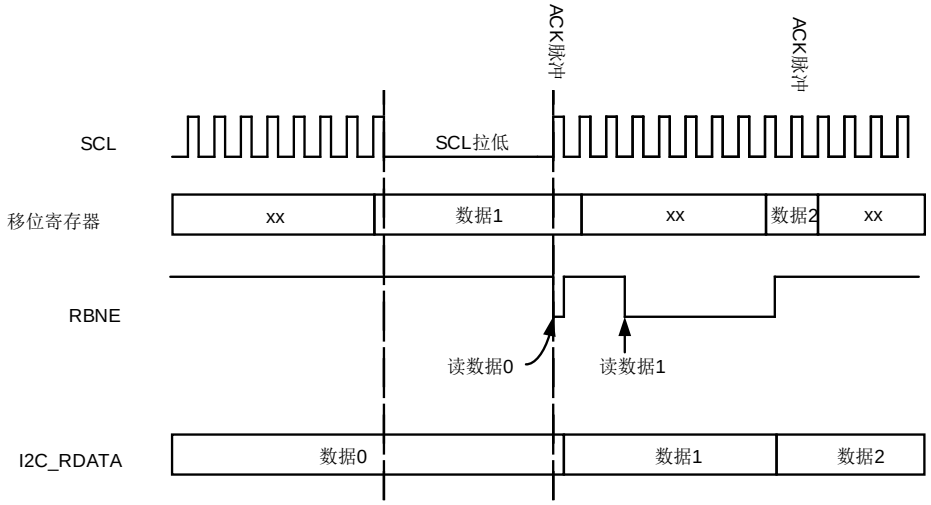
图 21-11. 数据发送



数据接收

在接收数据时，数据首先被接收到移位寄存器。如果 RBNE 为 0，移位寄存器中的数据将被移入 I2C_RDATA 寄存器。如果 RBNE 为 1，SCL 时钟将被拉低，直到之前接收到的数据字节被读取。这个时钟拉低被插入应答脉冲之前。

图 21-12. 数据接收



重载和自动结束模式

为了管理字节传输和中断如*表 21-3. 可关闭通信模式*所示几种通信模式，I2C 硬件嵌入了字节计数器。

表 21-3. 可关闭通信模式

工作模式	行为
主机模式	产生 NACK，STOP 和 RESTART

工作模式	行为
从机接收模式	ACK 控制
SMBus 模式	PEC 生成/校验

传输的字节数由 BYTENUM[7:0]在 I2C_CTL1 寄存器中配置。如果 BYTENUM 大于 255，或者处于从机字节控制模式，则必须通过将 I2C_CTL1 寄存器中 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM 计数到 0 时，TCR 位将置 1，如果 TCIE 位置 1 将产生中断。当 TCR 位置 1 时，SCL 将被拉低。在 BYTENUM 写一个非零值将清除 TCR 位。

注意：重载模式必须在 BYTENUM[7:0]最后一次重载后禁用。

当使能自动结束模式时，必须禁用重载模式。在自动结束模式下，当 BYTENUM[7:0]计数到 0 时，主机将自动发送一个 STOP 信号。

当重载模式和自动结束模式都被禁用时，I2C 通信进程需要由软件终止。如果 BYTENUM[7:0]中的字节数已经传输完成，软件应将 STOP 位置 1 来产生一个 STOP 信号，然后清除 TC。

21.3.7. I2C 从机模式

初始化

从机模式下，至少使能一个从机地址。第一个从机地址写在 I2C_SADDR0 寄存器中，第二个从机地址写在 I2C_SADDR1 寄存器中。在使用从机地址时，必须相应地将 I2C_SADDR0 寄存器中 ADDRESSEN 位和 I2C_SADDR1 寄存器中 ADDRESS2EN 置 1。通过设置 I2C_SADDR0 寄存器中 ADDFORMAT 位可以选择 7 位地址或 10 位地址，该地址被写在 ADDRESS[9:0]。

I2C_CTL2 寄存器中 ADDM[6:0]定义 ADDRESS[7:1]的哪些位和接收到的地址进行比较，哪些位不比较。

ADDMSK2[2:0]用于屏蔽 I2C_SADDR1 寄存器中 ADDRESS2[7:1]，相关详细信息参考 I2C_SADDR1 寄存器 ADDMSK2[2:0]位域描述。

当 I2C 接收到的地址与使能的地址其中一个匹配成功时，ADDSEND 将被置 1，如果 ADDMIE 置位，将产生中断。I2C_STAT 寄存器 READDR[6:0]将会存储接收到的地址。在 ADDSEND 置位时，I2C_STAT 寄存器中 TR 位状态更新。TR 的状态指示从机是作为发送器还是接收器。

SCL 线控制

当 SS=0 时，时钟拉低功能默认用在从机模式下，在需要的时候 SCL 会被拉低。在下列情况下，SCL 会被拉低。

- 当 ADDSEND 置位时 SCL 线拉低，并在 ADDSEND 位清零之后释放。
- 在从机发送模式下，ADDSEND 清零之后，SCL 在第一个字节写入 I2C_TDATA 寄存器之前都是被拉低的。在前一个字节发送完成之后，新的字节写入 I2C_TDATA 寄存器之前，SCL 也是被拉低的。
- 在从机接收模式下，接收过程已完成但是 I2C_RDATA 寄存器中的数据还未被读取，SCL 将被拉低。
- 当 SBCTL=1 且 RELOAD=1 时，在最后一个字节传输结束后，TCR 置位。在 TCR 清除

之前 SCL 将被拉低。

- SCL 下降沿被检测到之后，在[(SDADELY+SCLDELY+1)*(PSC+1)+1]*t_{I2CCLK}期间 SCL 被拉低。

SCL 线控制可以通过将 I2C_CTL0 寄存器中 SS 位置 1 来禁能。在下列情况下，SCL 不会被拉低。

- 在 ADDSEND 置位时 SCL 将不会被拉低。
- 在从机发送模式下，数据必须在它传输过程产生的第一个 SCL 脉冲之前写入 I2C_TDATA 寄存器。否则 I2C_STAT 寄存器中 OUERR 位将会置 1，如果 ERRIE 位也被置 1，将产生一个中断。当 STPDET 位置 1 并且第一个数据开始发送，I2C_STAT 寄存器中 OUERR 位也将置 1。
- 在从机接收模式下，数据必须在下一个字节接收产生的第九个 SCL 脉冲（ACK 脉冲）之前读取。否则 I2C_STAT 寄存器中 OUERR 位也将置 1。如果 ERRIE 位也被置 1，将产生一个中断。

从机字节控制模式

在从机接收模式下要实现字节 ACK 控制，可以通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 来使能从机字节控制模式。当 SS=1 时，从机字节控制模式无效。

在使用从机字节控制模式时，必须通过置位 I2C_CTL1 寄存器中 RELOAD 位来使能重载模式。从机字节控制模式中，在 ADDSEND 中断服务程序中 I2C_CTL1 寄存器中 BYTENUM[7:0]必须配置为 1，并且在每个字节接收完成时重载为 1。当接收到一个字节时，I2C_STAT 寄存器中 TCR 位置 1，在第八个和第九个 SCL 时钟脉冲之间从机将 SCL 时钟拉低。然后数据可以从 I2C_RDATA 寄存器中读取出来，通过配置 I2C_CTL1 寄存器中 NACKEN 位，从机可以决定发送 ACK 或者是 NACK。当在 BYTENUM[7:0]写入非零值时，从机释放 SCL 时钟线。

当 BYTENUM[7:0]大于 0x1 时，在 BYTENUM[7:0]数据接收期间，数据流是连续的。

注意：在下列情况下，可以配置 SBCTL 位：

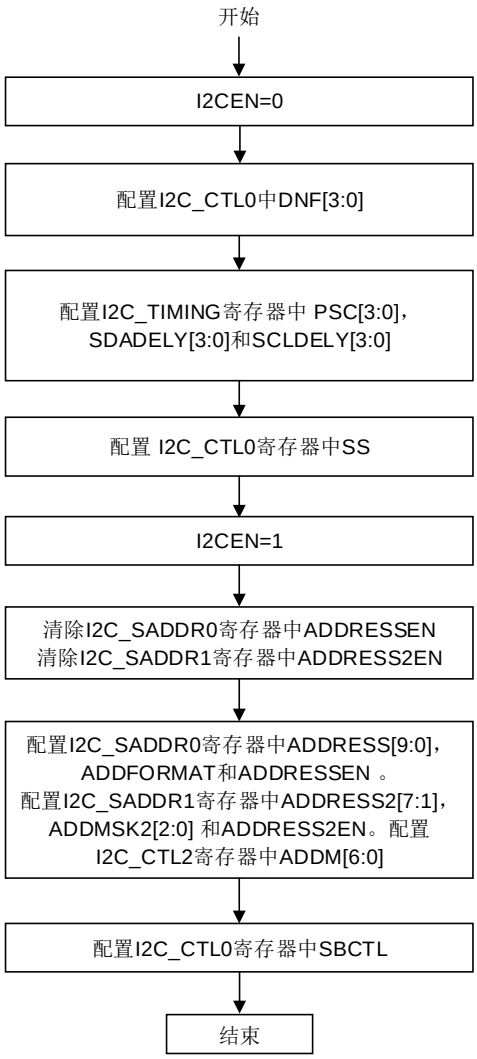
1、 I2CEN=0。

2、 从机还未被寻址。

3、 ADDSEND=1。

当 ADDSEND=1，或者 TCR=1 时，RELOAD 才可以被修改。

图 21-13. I2C 从机初始化



从机发送模式下的软件流程

当 I2C_TDATA 寄存器为空，I2C_STAT 寄存器中 TI 位将会置位。如果 I2C_CTL0 寄存器中 TIE 位置 1，将产生中断。当接收到 NACK 时，I2C_STAT 寄存器中 NACK 位会置位。如果 I2C_CTL0 寄存器中 NACKIE 位置 1，将产生中断。当接收到 NACK 信号时，I2C_STAT 寄存器中 TI 位将不会置位。

当接收到 STOP 信号时，I2C_STAT 寄存器中 STPDET 位将置 1。如果 I2C_CTL0 寄存器中 STPDETIE 位置 1，将产生中断。

当 SBCTL=0 时，如果 ADDSEND=1，且 I2C_STAT 寄存器中 TBE 位为 0，可以选择发送 I2C_TDATA 寄存器中的数据或者是将 TBE 置 1 来清空 I2C_TDATA 寄存器。

当 SBCTL=1 时，从机工作在字节控制模式，BYTENUM[7:0]必须在 ADDSEND 中断服务程序中配置。TI 事件的数量与 BYTENUM[7:0]的值相等。

当 SS=1 时，I2C_STAT 寄存器中 ADDSEND 位置位时 SCL 时钟线不会被拉低。在这种情况下，I2C_TDATA 寄存器中数据不能在 ADDSEND 中断服务程序中清空。因此待发送的第一个字节应该在 ADDSEND 置位之前就被编程到 I2C_TDATA 寄存器。

- 该数据可以是上一次数据传输最后一次 TI 事件写入的数据。
- 如果该数据不是待发送数据，可通过将 TBE 位置 1 来刷新 I2C_TDATA 寄存器，从而编程新的数据。在数据发送开始时 STPDET 位必须为 0。否则 I2C_STAT 寄存器中 OUERR 位将置 1 并产生下溢错误。
- 从机发送模式下使用中断或者 DMA 时，如果需要一个 TI 事件，TI 位和 TBE 位都必须置 1。

图 21-14. I2C 从机发送编程模型（SS=0）

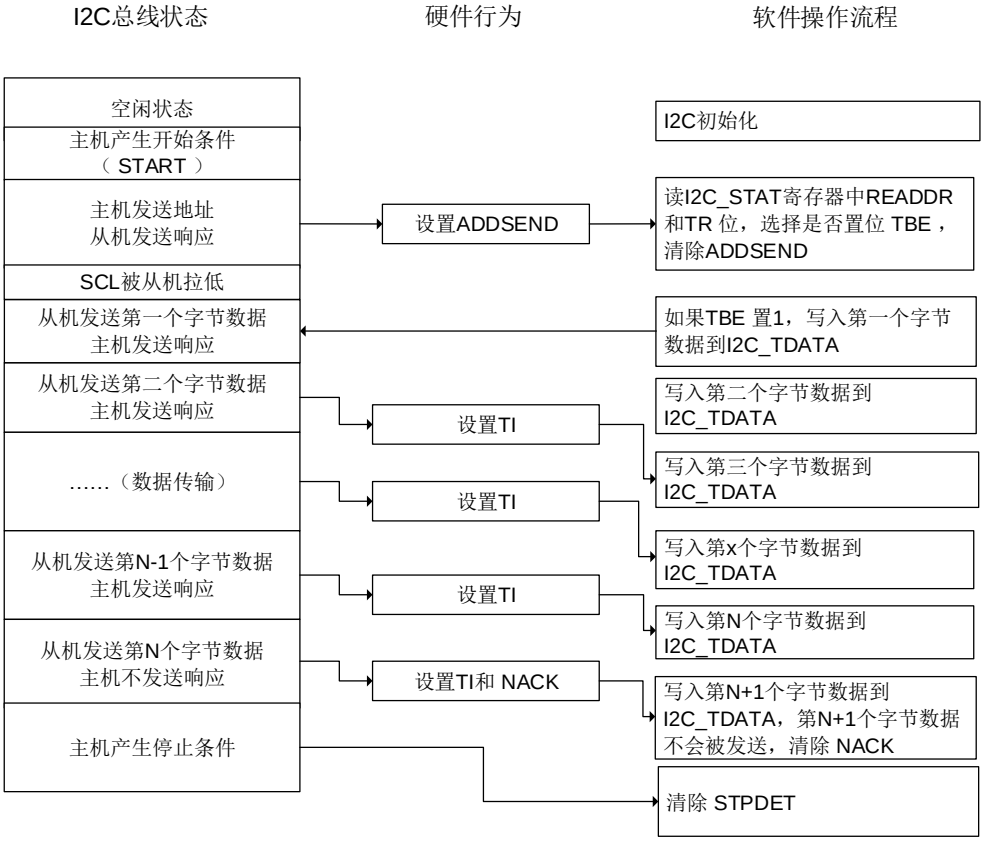
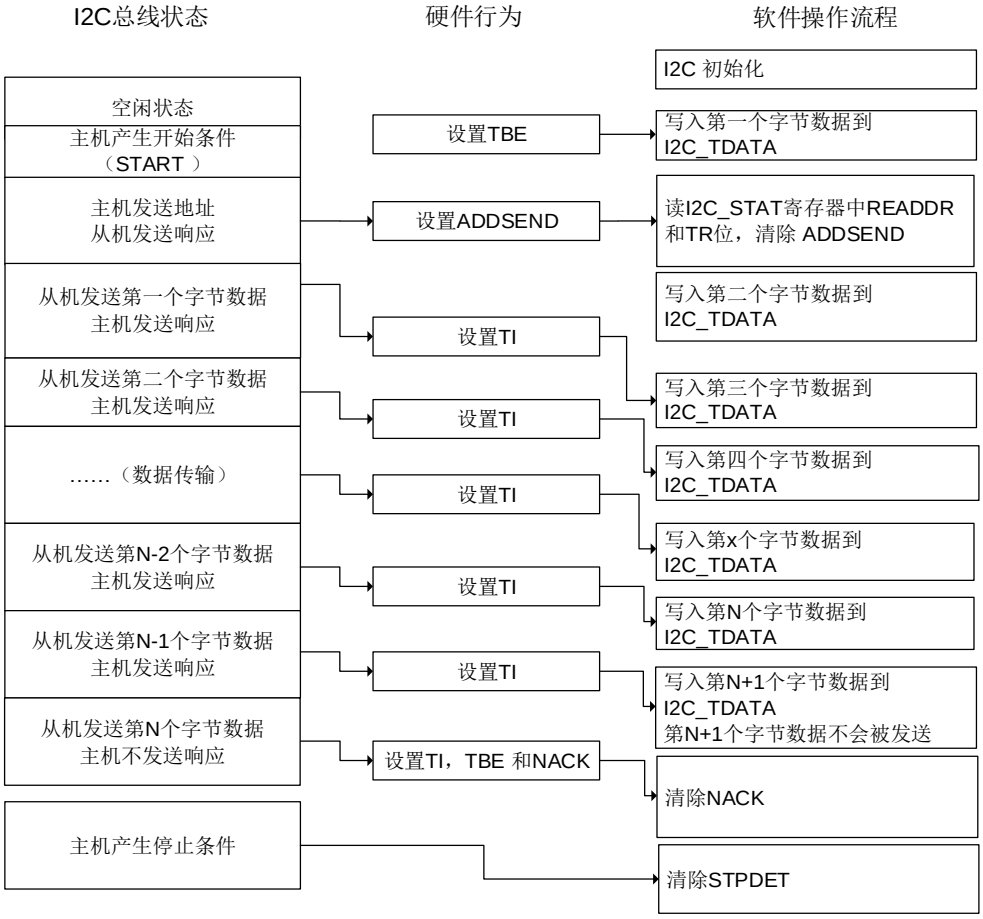


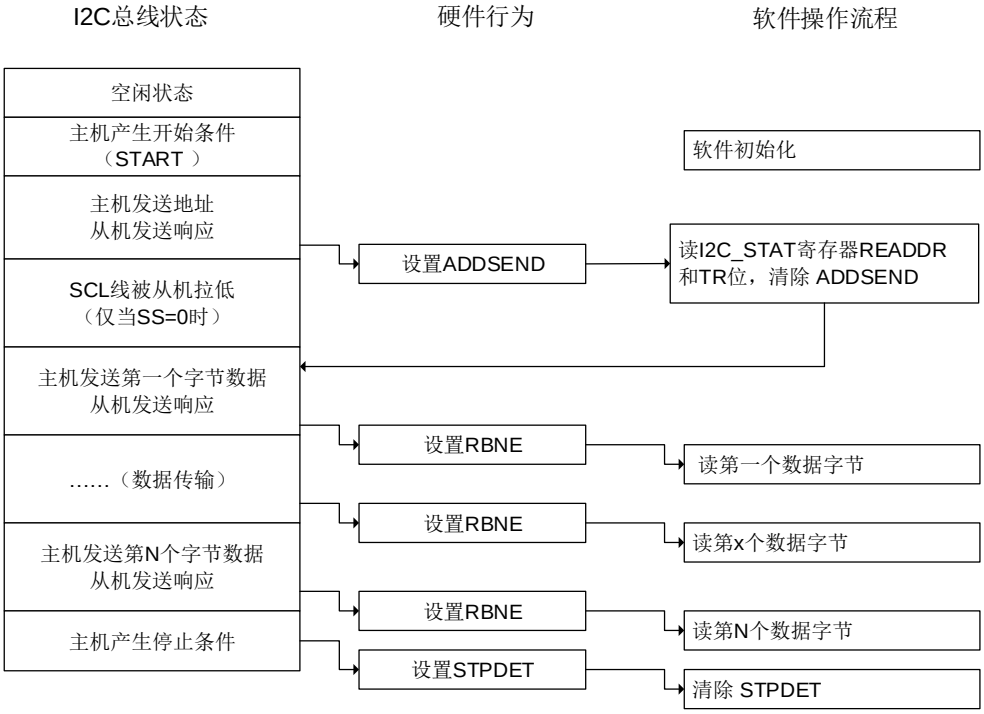
图 21-15. I2C 从机发送编程模型（SS=1）



从机接收模式下的软件流程

当 I2C_RDATA 寄存器非空，I2C_STAT 寄存器中 RBNE 位置 1，如果 I2C_CTL0 寄存器中 RBNEIE 位置 1，将产生中断。当接收到 STOP 信号时，I2C_STAT 寄存器中 STPDET 位将置 1。如果 I2C_CTL0 寄存器中 STPDETIE 置 1，将产生中断。

图 21-16. I2C 从机接收编程模型



21.3.8. I2C 主机模式

初始化

I2C_TIMING 寄存器中 SCLH[7:0]和 SCLL[7:0]必须在 I2CEN=0 时配置。为了支持多主机通信和从机时钟拉低，I2C 实现了时钟同步机制。

SCLL[7:0]和 SCLH[7:0]分别用于低电平计数和高电平计数。经过t_{SYNC1}延时后，当检测到 SCL 低电平时，SCLL[7:0]开始计数，如果 SCLL[7:0]计数器的值达到 I2C_TIMING 寄存器中 SCLL[7:0]时，I2C 将释放 SCL 时钟。经过t_{SYNC2}延时后，当检测到 SCL 高电平时，SCLH[7:0]开始计数，如果 SCLH[7:0]计数器的值达到 I2C_TIMING 寄存器中 SCLH[7:0]时，I2C 将拉低 SCL 时钟。

因此主机时钟周期为：$t_{SCL}=t_{SYNC1}+t_{SYNC2}+\{[(SCLH[7:0]+1)+(SLLL[7:0]+1)]*(PSC+1)*t_{I2CCLK}\}$。

t_{SYNC1}取决于 SCL 下降沿斜率，SCL 输入模拟和数字噪声滤波器延时以及 SCL 与 I2CCLK 时钟的同步产生的延时，一般为 2 到 3 个 I2CCLK 时钟周期。t_{SYNC2}取决于 SCL 上升沿斜率，SCL 输入模拟和数字噪声滤波器延时以及 SCL 与 I2CCLK 时钟的同步产生的延时，一般为 2 到 3 个 I2CCLK 时钟周期。数字噪声滤波器产生的延时为$DNF[3:0]*t_{I2CCLK}$。

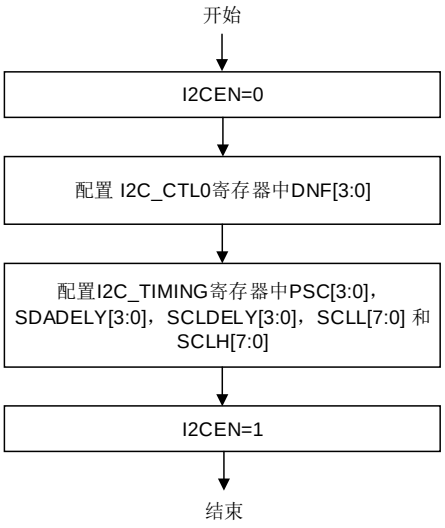
在主机模式下，必须配置 I2C_CTL1 寄存器中 ADD10EN，SADDRESS[9:0]以及 TRDIR 位。当在主机接收模式下使用 10 位寻址时，必须配置 HEAD10R 来选择是执行完整的地址寻址序列，还是只发送地址头。待传输的字节数在 I2C_CTL1 寄存器 BYTENUM[7:0]配置。如果待传输的字节数大于或者等于 255，必须将 BYTENUM[7:0]配置为 0xFF。然后主机发送 START 信号。以上提到的所有位必须在 START 位置 1 之前配置。START 信号发送完成之后，待 I2C_STAT 寄存器 I2CBSY 位为 0 时，发送从机地址。当仲裁丢失时，主机切换成从机模式，START 位

由硬件清零。当从机地址发送完成时，START 位由硬件清零。

在 10 位寻址模式下，在发送 10 位地址头之后，如果主机接收到 NACK，主机将重发 10 位地址头直到收到 ACK。将 ADDSENDC 置 1 可以停止重发从机地址。

如果 START 位置 1 时，I2C 作为从机被寻址成功，ADDSEND 置 1，主机将切换为从机模式。START 位将在 ADDSENDC 置 1 时清零。

图 21-17. I2C 主机初始化



主机发送模式下的软件流程

在主机发送模式下，每一个字节发送完成并接收到 ACK 信号之后，TI 位将置 1。如果 I2C_CTL0 寄存器中 TIE 位置 1，将产生中断。待发送的字节数编程在 I2C_CTL0 寄存器 BYTENUM[7:0]。如果发送字节数大于 255，必须通过将 I2C_CTL0 寄存器 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器 TCR 位将置 1，并且在 BYTENUM[7:0]更新一个非零值之前，SCL 被拉低。

如果接收到 NACK，TI 位将不会置 1。

- 如果 BYTENUM[7:0]个字节传输完成且 RELOAD=0，将 I2C_CTL1 寄存器中 AUTOEND 置 1 可以自动产生 STOP 信号。当 AUTOEND=0 时，I2C_STAT 寄存器 TC 位将置 1 且 SCL 被拉低。在这种情况下，主机可以通过将 I2C_CTL1 寄存器中 STOP 位置 1 来产生 STOP 信号。或者产生 RESTART 信号来开始一个新的数据传输过程。将 START / STOP 置 1 可以清除 TC 位。
- 如果接收到 NACK 信号，I2C 将自动产生 STOP 信号。I2C_CTL0 寄存器中 NACK 将置 1，如果 NACKIE 位置 1，将产生中断。

注意：当 RELOAD=1 时，AUTOEND 位无效。

图 21-18. I2C 主机发送编程模型（N<=255）

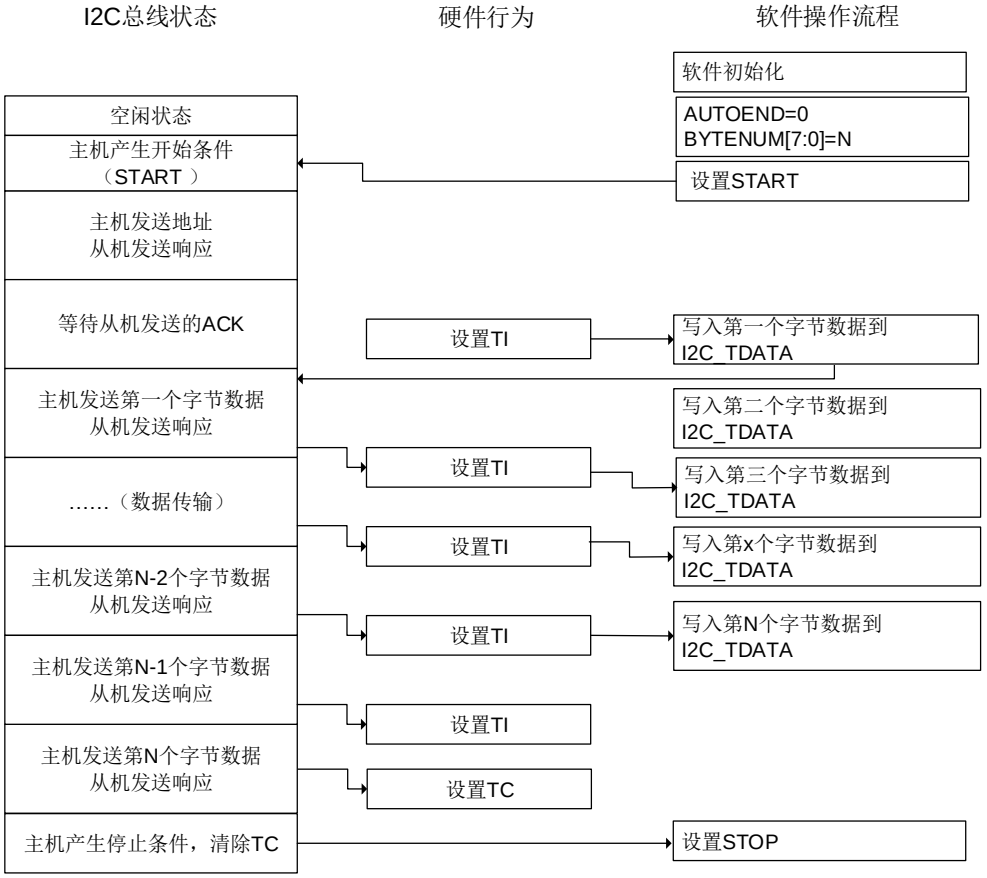
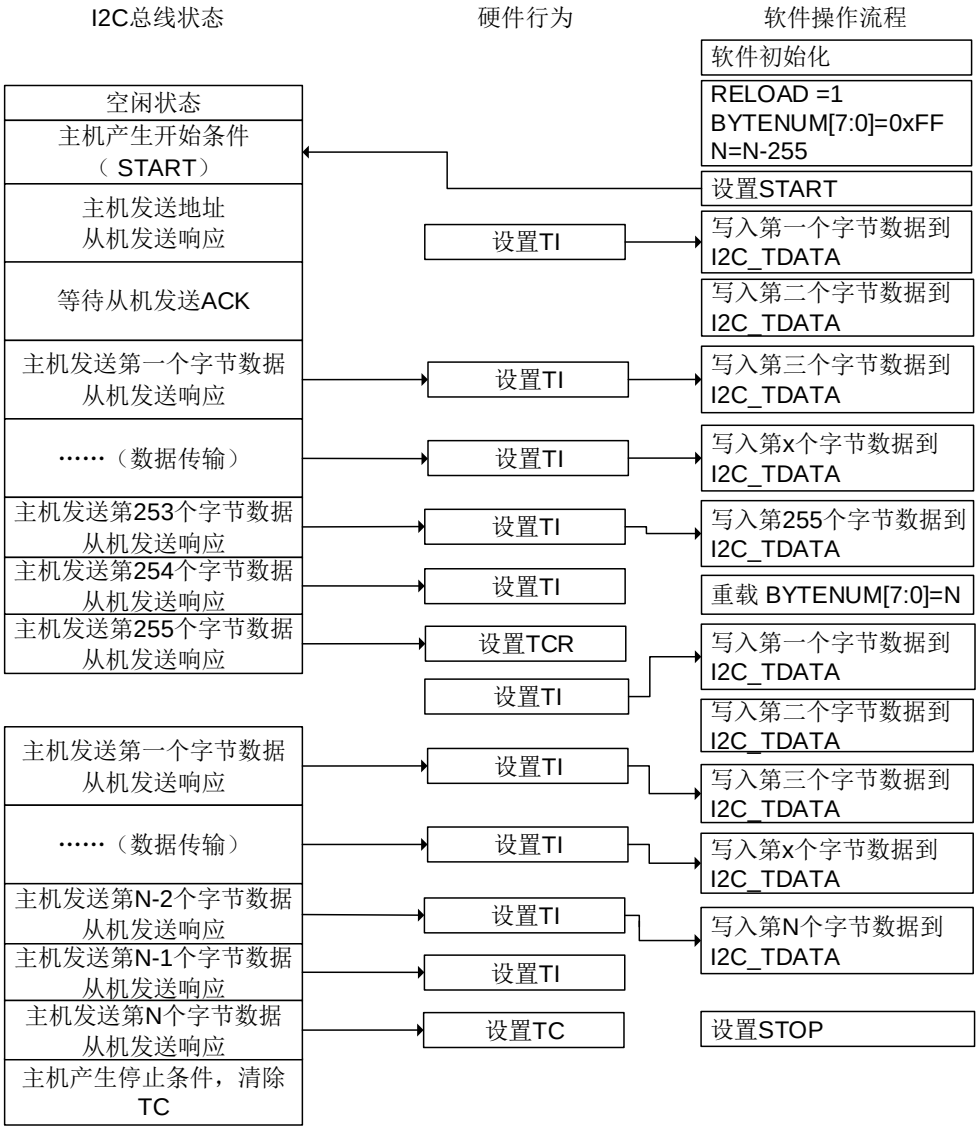


图 21-19. I2C 主机发送编程模型（N>255）



主机接收模式下的软件流程

在主机接收模式下，当接收到一个字节时，I2C_STAT 寄存器中 RBNE 位置 1。如果 I2C_CTL0 寄存器中 RBNEIE 置 1，将产生一个中断。如果待接收字节数大于 255，必须将 I2C_CTL0 寄存器中 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器中 TCR 位将置 1，在 BYTENUM[7:0]中写入一个非零值之前，SCL 被拉低。

如果 BYTENUM[7:0]个字节传输完成且 RELOAD=0，将 I2C_CTL1 寄存器中 AUTOEND 置 1 可以自动产生 STOP 信号。当 AUTOEND=0 时，I2C_STAT 寄存器 TC 位将置 1 且 SCL 被拉低。在这种情况下，主机可以通过将 I2C_CTL1 寄存器中 STOP 位置 1 来产生 STOP 信号。或者产生 RESTART 信号来开始一个新的数据传输过程。将 START / STOP 置 1 可以清除 TC 位。

图 21-20. I2C 主机接收编程模型（N<=255）

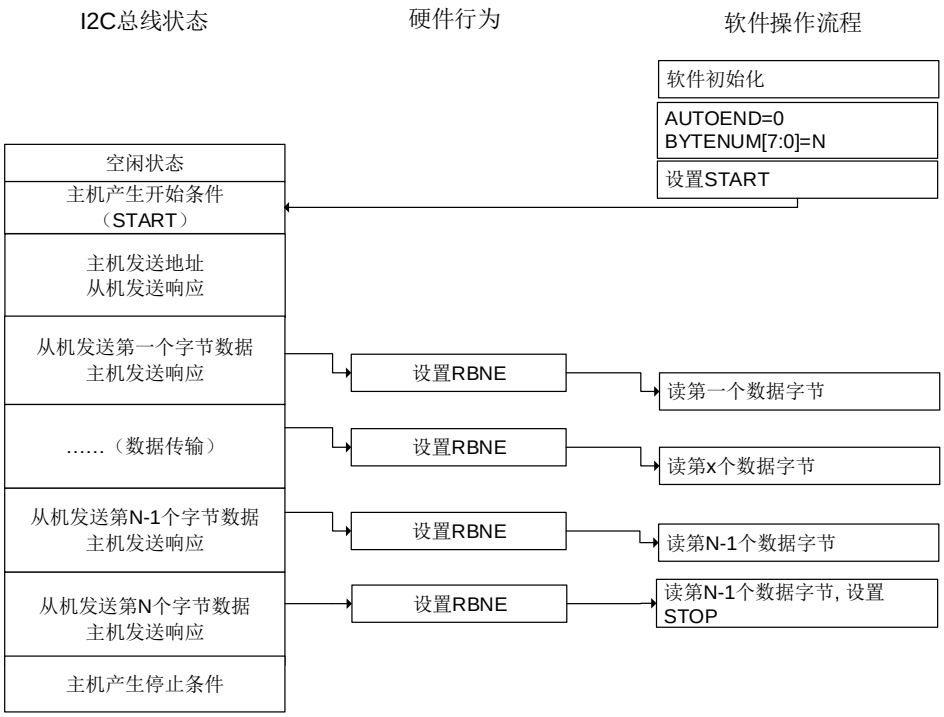


图 21-21. I2C 主机接收编程模型（N>255）

21.3.9. SMBus 支持

系统管理总线（System Management Bus，简写为 SMBus 或 SMB）是一种结构简单的单端双线制总线，可实现轻量级的通信需求。一般来说，SMBus 最常见于计算机主板，主要用于电源传输 ON/OFF 指令的通信。SMBus 是 I2C 的一种衍生总线形式，主要用于计算机主板上的低带宽设备间通信，尤其是与电源相关的芯片，例如笔记本电脑的可充电电池子系统（参见 Smart Battery Data）。

SMBus 协议

SMBus 上每个报文交互都遵从 SMBus 协议中预定义的格式。SMBus 是 I2C 规范中数据传输格式的子集。只要 I2C 设备可通过 SMBus 协议之一进行访问，便视为兼容 SMBus 规范。不符合这些协议的 I2C 设备，将无法被 SMBus 和 ACPI 规范所定义的标准方法访问。

地址解析协议

SMBus 采用了 I2C 硬件以及 I2C 的硬件寻址方式，但在 I2C 的基础上增加了二级软件处理，建立自己独特的系统。比较特别的是 SMBus 规范包含一个地址解析协议，可用于实现动态地址分配。动态识别硬件和软件使得总线设备能够支持热插拔，无需重启系统便能即插即用。总线中的设备将被自动识别并分配唯一地址。这个优点非常有利于实现即插即用的用户界面。在此协议中，系统中的 host 与设备之间有一个重要的区别，即 host 具有分配地址的功能。

SMBus 从机字节控制

SMBus 接收器从机字节控制与 I2C 一样。它允许 ACK 控制每个字节。必须能对接收到的命令或者数据进行 NACK 应答。通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 来使能从机字节控制模式。

主机通知协议

通过将 I2C_CTL0 寄存器 SMBHAEN 位置 1，SMBus 可以支持主机通知协议。在该协议中，从设备作为主机，主设备作为从机，主机将应答 SMBus 主机地址。

超时特性

SMBus 有一种超时特性：假如某个通信耗时太久，便会自动复位设备。这就解释了为什么最小时钟周期为 10kHz——为了防止长时间锁死总线。I2C 在本质上可以视为一个“直流”总线，也就是说当主机正在访问从机的时候，假如从机正在执行一些子程序无法及时响应，从机可以拉住主机的时钟。这样便可以提醒主机：从机正忙，但并不想放弃当前的通信。从机的当前任务结束之后，将可以继续 I2C 通信。I2C 总线协议中并没有限制这个延时的上限，但在 SMBus 系统中，这个时间被限定为 25~35ms。按照 SMBus 协议的假定，如果某个会话耗时太久，就意味着总线出了问题，此时所有设备都应当复位以消除这种（问题）状态。这样就并不允许从设备将时钟拉低太长时间。

将 I2C_TIMEOUT 寄存器中 TOEN 位和 EXTOEN 位置 1 可以使能超时检测。配置定时器必须保证在 SMBus 规范规定的时间最大值之前检测出超时情况。

在 BUSTOA[11:0]中编程的值被用来检查$t_{TIMEOUT}$参数。必须将 TOIDLE 位配置为 0，以检测 SCL 低电平超时。将 I2C_TIMEOUT 寄存器中 TOEN 位置 1 来使能定时器，在 TOEN 置 1 之后，BUSTOA[11:0] 和 TOIDLE 位不能被修改。如果 SCL 低电平时间大于(BUSTOA+1)*2048*t_{I2CCLK}，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

BUSTOA[11:0]为从机校验$t_{LOW:SEXT}$，为主机校验$t_{LOW:MEXT}$。通过将 I2C_TIMEOUT 寄存器中 EXTOEN 位置 1 来使能定时器。在 EXTOEN 置 1 之后，BUSTOB[11:0]不能被修改。如果 SMBus 外设 SCL 拉低时间大于(BUSTOB+1)*2048*t_{I2CCLK}，并且达到了总线空闲检测章节中描述的超时时间间隔，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

报文错误校验

I2C 模块中有一个 PEC 模块，它使用 CRC-8 计算器来执行 I2C 数据的报文校验。一个 PEC 字节（PEC 错误码）附加在每次传输结束。PEC 的计算方式是对所有消息字节（包含地址和读/写位）使用 CRC-8 计算校验和。CRC-8 多项式位x^8+x^2+x+1（CRC-8-ATM HEC 算法，初

始化为 0）。

当 I2C 被禁用时，通过 I2C_CTL0 寄存器中的 PECEN 位置 1 可以使能 PEC。由于 PEC 传输是由 I2C_CTL1 寄存器中 BYTENUM[7:0]管理的，因此在从机模式下必须将 SBCTL 位置 1。当 PECTRANS 置 1，RELOAD 为 0 时，在 BYTENUM[7:0]-1 数据字节后发送 PEC。PEC 在 BYTENUM[7:0]-1 传输完成后发送。当 RELOAD 置 1 时 PECTRANS 无效。

SMBus 警报

SMBus 还有一个额外的共享的中断信号，称为 SMBALERT#。从机上发生事件后，可通过这个信号通知主机来访问从机。主机会处理该中断，并通过报警响应地址，同时访问所有 SMBALERT#设备。如果 SMBALERT#电平被设备拉低，这些设备会应答报警响应地址。当配置为从设备（SMBHAEN=0）时，通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以将 SMBA 引脚电平拉低。同时也使能了报警响应地址。当配置为主设备（SMBHAEN=1），且 SMBALTEN 置 1 时，当在 SMBA 引脚检测到下降沿时，I2C_STAT 寄存器中 SMBALT 位将置 1。如果 I2C_CTL0 寄存器中 ERRIE 位置 1，将产生中断。当 SMBALTEN=0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。当 SMBALTEN=0 时，SMBA 引脚可用作标准 GPIO。

总线空闲检测

如果主机检测到时钟信号和数据信号的高电平持续时间大于$t_{HIGH,MAX}$，总线被视为空闲。

该时序参数已考虑到主机已动态添加至总线，但可能还未检测到 SMBCLK 或 SMBDAT 线上的状态转换的情况。在这种情况下，为了保证当前没有数据传输正在进行，主机必须等待足够长的时间。

要使能t_{IDLE}检查，必须将 BUSTOA[11:0]编程为定时器重载值，以获取t_{IDLE}参数。必须将 TIDLE 位置 1，以检测 SCL 和 SDA 高电平超时。然后通过将 I2C_TIMEOUT 寄存器中的 TOEN 位置 1 来使能定时器。TOEN 置 1 后，BUSTOA[11:0]和 TIDLE 不能被修改。如果 SCL 和 SDA 的高电平持续时间都大于(BUSTOA+1)*4*t_{I2CCLK}，I2C_STAT 寄存器中 TIMEOUT 位将置位。

SMBus 从机模式

SMBus 接收器必须能够对接收到的命令和数据进行 NACK 应答。对于从机模式下的 ACK 控制，通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 可以使能从机字节控制模式。

必要时应使能特定的 SMBus 地址。通过将 I2C_CTL0 寄存器中 SMBDAEN 置 1 可以使能 SMBus 设备默认地址（0b1100 001）。通过将 I2C_CTL0 寄存器中 SMBHAEN 置 1 可以使能 SMBus 主机地址（0b0001 000）。通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以使能报警响应地址（0b0001 100）。

21.3.10. SMBus 模式

SMBus 主机发送器和从机接收器

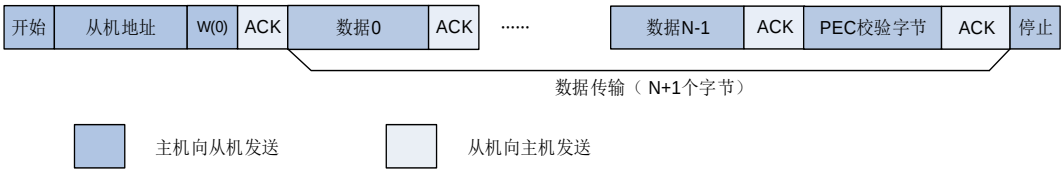
当 SMBus 主机发送 PEC 时，必须在 START 位置 1 前，将 PECTRANS 位置 1 并在

BYTENUM[7:0]位域中配置字节数。在这种情况下，总 TI 中断数为 BYTENUM-1。因此，如果 BYTENUM=0x1 且 PECTRANS 位置 1，则 I2C_PEC 寄存器的数据将自动发送。如果 AUTOEND 为 1，SMBus 主机在 PEC 字节发送完成之后将自动发送 STOP 信号。如果 AUTOEND 为 0，SMBus 主机可以在 PEC 字节发送完成之后发送 RESTART 信号。I2C_PEC 寄存器中的数据将在 BYTENUM -1 个字节发送完成后发送，PEC 字节发送完成后 TC 位将置 1。SCL 线被拉低。RESTART 位必须在 TC 中断服务程序中置 1。

SMBus 作为从机接收器时，为了在数据发送完成时进行 PEC 校验，SBCTL 位必须置 1。要对每个字节进行 ACK 控制，必须通过将 RELOAD 位置 1 来使能 RELOAD 模式。如果要校验 PEC 字节，必须将 RELOAD 位清零同时将 PECTRANS 置 1。在 BYTENUM-1 个字节接收完成后，接收的下一个字节将与 I2C_PEC 寄存器中的数据进行比较。如果校验值不匹配，将自动产生 NACK 信号；如果校验值匹配将自动产生 ACK 信号，将忽略 NACKEN 位的值。当接收到 PEC 字节时，PEC 字节会存到 I2C_RDATA 寄存器中，RBNE 位将置 1。如果 I2C_CTL0 寄存器中 ERRIE 位置 1，且 PEC 值不匹配，PECERR 将会置 1 并产生中断。如果无须使用 ACK 控制，PECTRANS 可以设置为 1，BYTENUM 可以根据待接收字节数来配置。

注意：在 RELAOD 位置 1 之后，PECTRANS 不可以被修改。

图 21-22. SMBus 主机发送器和从机接收器通信流程



SMBus 主机接收器和从机发送器

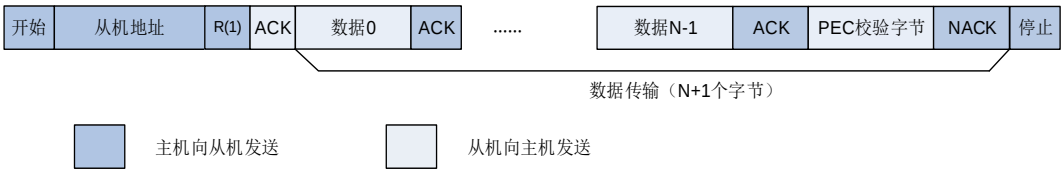
如果 SMBus 主机需要在数据传输完成后接收 PEC 字节，可以使能自动结束模式。在 START 信号发送之前，必须将 PECTRANS 位置 1，且配置好从机地址。在接收 BYTENUM-1 数据之后，接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在停止信号发送之前，接收 PEC 字节之后会给出 NACK 响应。

如果 SMBus 主机需要在接收到 PEC 字节之后产生 RESTART 信号，需要禁能自动结束模式。在 START 信号发送之前，PECTRANS 位必须置 1，且配置好从机地址。在接收 BYTENUM-1 数据之后，接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在 PEC 字节发送完成之后 TC 位将置 1，SCL 线被拉低。在 TC 中断服务程序中可将 RESTART 位置 1。

当 SMBus 作为从机发送器时，为了在 BYTENUM[7:0]个字节发送完成之后发送 PEC 字节，SBCTL 位必须置 1。如果 PECTRANS 置 1，字节数 BYTENUM[7:0]包含 PEC 字节。在这种情况下，如果主机请求接收的字节数大于 BYTENUM-1，总 TI 中断数为 BYTENUM-1，I2C_PEC 寄存器中的数据将自动发送。

注意：PECTRANS 位在 RELOAD 置 1 之后不能被修改。

图 21-23. SMBus 主机接收器和从机发送器通信流程



21.3.11. 从省电模式唤醒

当 I2C 地址匹配成功时，MCU 从深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2 被唤醒。为了将 MCU 从这些省电模式唤醒，I2C_CTL0 寄存器中 WUEN 位必须置 1，同时 I2CCLK 时钟源选择 IRC16M。在深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2 下，IRC16M 关闭。当 I2C 检测到 START 信号时，IRC16M 打开，I2C 会将 SCL 拉低直到 IRC16M 被唤醒。在接收地址期间，IRC16M 为 I2C 提供时钟。当地址匹配时，在 MCU 唤醒期间，I2C 的 SCL 线被拉低。当 ADDSEND 清除时，SCL 线被释放，数据传输过程恢复正常。如果检测到的地址不匹配，IRC16M 会再次关闭，MCU 将不会被唤醒。

只有地址匹配中断（ADDMIE=1）能唤醒 MCU。如果 I2C 的时钟源是系统时钟，或者 WUEN=0，IRC16M 在接收到 START 信号之后将不会打开。当从省电模式唤醒使能时，数字滤波器必须禁能，I2C_CTL0 寄存器中 SS 位也必须清 0。如果禁止从省电模式唤醒（WUEN=0），则在进入省电模式之前必须禁能 I2C 外设（I2CEN=0）。

21.3.12. DMA 模式下数据传输

如 I2C 从机模式和主机模式中描述，每当 TI 位和 RBNE 位被置 1 之后，软件都应该写或读一个字节，这样将导致 CPU 的负荷较重。I2C 的 DMA 功能可以在 TI 或 RBNE 位置 1 时，自动进行一次写或读操作。

将 I2C_CTL0 寄存器中 DENT 置 1 可以使能 DMA 发送请求。将 I2C_CTL0 寄存器中 DENR 置 1 可以使能 DMA 接收请求。在主机模式下，由软件写入从机地址，传输方向，待发送字节数和 START 位。DMA 必须在 START 位置 1 之前初始化。在 I2C_CTL1 寄存器 BYTENUM[7:0] 位配置待传输字节数。在从机模式下，DMA 必须在地址匹配事件发生之前或 ADDSEND 中断服务程序中清除 ADDSEND 标志之前完成初始化。

21.3.13. I2C 错误和中断

I2C 错误标志如*表 21-4. I2C 错误标志*所示。

表 21-4. I2C 错误标志

I2C 错误名称	描述
BERR	总线错误
LOSTARB	仲裁丢失
OUERR	上溢 / 下溢标志
PECERR	CRC 值不匹配
TIMEOUT	SMBus 模式下总线超时
SMBALT	SMBus 报警

I2C 中断和事件标志如*表 21-5. I2C 中断事件*所示。

表 21-5. I2C 中断事件

中断事件	事件标志	使能控制位
在接收期间 I2C_RDATA 非空	RBNE	RBNEIE
发送中断	TI	TIE
从机模式下检测到 STOP 信号	STPDET	STPDETIE
传输完成重载	TCR	TCIE
传输完成	TC	
地址匹配	ADDSEND	ADDMIE
接收到 NACK	NACK	NACKIE
总线错误	BERR	ERRIE
仲裁丢失	LOSTARB	
上溢/下溢错误	OUERR	
PEC 错误	PECERR	
超时错误	TIMEOUT	
SMBus 报警	SMBALT	

21.3.14. I2C 调试模式

当为控制器进入调试模式（Cortex®-M23 内核停止），SMBus 超时定时器会根据 DBG 模块中的 I2Cx_HOLD 配置位选择继续正常工作还是停止工作。

21.4. I2C 寄存器

I2C0 基地址：0x4000 5400

I2C1 基地址：0x4000 5800

I2C2 基地址：0x4000 C000

21.4.1. 控制寄存器 0（I2C_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								PECEN	SMBALTEN	SMBDAEN	SMBHAEN	GCEN	WUEN	SS	SBCTL
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DENR	DENT	保留	ANOFF	DNF[3:0]				ERRIE	TCIE	STPDETIE	NACKIE	ADDMIE	RBNEIE	TIE	I2CEN
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	PECEN	PEC 计算开关。 0：PEC 计算关闭。 1：PEC 计算打开。
22	SMBALTEN	SMBus 报警使能。 0：从机模式下 SMBA 引脚高电平或主机模式下 SMBus 报警引脚 SMBA 禁能。 1：从机模式下 SMBA 引脚低电平或主机模式下 SMBus 报警引脚 SMBA 使能。
21	SMBDAEN	SMBus 设备默认地址使能。 0：设备默认地址禁能，对默认地址 0b1100001x 进行 NACK 应答。 1：设备默认地址使能，对默认地址 0b1100001x 进行 ACK 应答。
20	SMBHAEN	SMBus 主机地址使能。 0：主机地址禁能，对地址 0b0001000x 进行 NACK 应答。 1：主机地址使能，对地址 0b0001000x 进行 ACK 应答。
19	GCEN	是否响应对地址（0x00）的广播呼叫。 0：从机不响应广播呼叫。 1：从机将响应广播呼叫。
18	WUEN	使能从省电模式中唤醒，包含深度睡眠模式，深度睡眠模式 1 和深度睡眠模式 2。 当 MCU 从省电模式唤醒时该位清零。

		0：禁止从省电模式中唤醒。 1：使能从省电模式中唤醒。 **注意：**当 DNF[3:0] = 0 时，WUEN 才能被置 1。I2C2 支持唤醒 Deep-sleep / Deep-sleep1 / Deep-sleep2 模式，I2C0 / I2C1 支持唤醒 Deep-sleep / Deep-sleep1 模式。
17	SS	在从机模式下数据未就绪时是否将 SCL 拉低。 软件置 1 和清 0。 0：拉低 SCL 1：不拉低 SCL **注意：**在主机模式下，该位必须为 0。该位只能在 I2CEN=0 时被修改。
16	SBCTL	从机模式下字节控制。 该位用于在从机模式下使能硬件字节控制。 0：从机模式下字节控制禁能。 1：从机模式下字节控制使能。
15	DENR	DMA 接收使能 0：DMA 接收禁能 1：DMA 接收使能
14	DENT	DMA 发送使能 0：DMA 发送禁能 1：DMA 发送使能
13:12	保留	必须保持复位值。
12	ANOFF	模拟噪声滤波器禁能 0：模拟噪声滤波器使能。 1：模拟噪声滤波器禁能。 **注意：**该位只有在 I2C 禁能（I2CEN=0）时被编程。
11:8	DNF[3:0]	数字噪声滤波器 0000：数字噪声滤波器禁能。 0001：数字噪声滤波使能并且可以滤除脉宽宽度不大于 1 t_{I2CCLK}的尖峰。 ... 1111：数字噪声滤波使能并且可以滤除脉宽宽度不大于 15 t_{I2CCLK}的尖峰。 这些位只能在 I2C 禁能（I2CEN = 0）时修改。
7	ERRIE	错误中断使能 0：错误中断禁能 1：错误中断使能，当 BERR，LOSTARB，OUERR，PECERR，TIMEOUT 或 SMBALT 位置 1 时，将产生中断。
6	TCIE	传输完成中断使能 0：传输完成中断禁能。 1：传输完成中断使能。
5	STPDETIE	停止信号检测中断使能 0：停止信号（STPDET）检测中断禁能。

1：停止信号（STPDET）检测中断使能。

4	NACKIE	接收到 NACK 应答中断使能 0：接收到 NACK 应答中断禁能。 1：接收到 NACK 应答中断使能。
3	ADDMIE	从机模式下地址匹配中断使能 0：地址匹配中断禁能。 1：地址匹配中断使能。
2	RBNEIE	接收中断使能 0：接收（RBNE）中断禁能。 1：接收（RBNE）中断使能。
1	TIE	发送中断使能 0：发送中断（TI）禁能。 1：发送中断（TI）使能。
0	I2CEN	I2C 外设使能 0：I2C 禁能。 1：I2C 使能。

21.4.2. 控制寄存器 1（I2C_CTL1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					PECTRANS	AUTOEND	RELOAD	BYTENUM[7:0]							
					rw	rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACKEN	STOP	START	HEAD10R	ADD10EN	TRDIR	SADDRESS[9:0]									
rw	rw	rw	rw	rw	rw	rw									

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	PECTRANS	PEC 传输 软件置 1 和清 0，硬件在以下条件下清除此位：PEC 传输完成或者 ADDSEND 置 1 或者检测到 STOP 信号或者 I2CEN=0。 0：不传输 PEC 值。 1：传输的 PEC 值。 **注意：**当 RELOAD=1 或者从机模式下 SBCTL=0 时，该位无效。
25	AUTOEND	主机模式下自动结束模式

0：当 BYTENUM[7:0]个字节传输完成后时，TC 位置 1。
1：当 BYTENUM[7:0]个字节传输完成后时，自动发送 STOP 信号。
注意：该位仅在 RELOAD=0 时有效。该位由软件置 1 和清 0。

24	RELOAD	重载模式使能 0：当 BYTENUM[7:0]个字节传输完成后时，传输结束。 1：当 BYTENUM[7:0]个字节传输完成后时，传输未结束，重载新的 BYTENUM[7:0]。每次 BYTENUM[7:0]个字节传输完成，I2C_STAT 寄存器中 TCR 位将置 1。 该位由软件置 1 和清 0。
23:16	BYTENUM[7:0]	待传输的字节数 这些用来编程待传输的字节数。当 SBCTL=0 时，这些位无效。 **注意：**当 START 位置 1 时，这些位不能被修改。
15	NACKEN	从机模式下产生 NACK 0：在接收到新的字节时，发送 ACK。 1：在接收到新的字节时，发送 NACK。 **注意：**该位可由软件置 1，并在以下情况下由硬件清零：NACK 发送完成或检测到 STOP 信号或 ADDSEND 置 1，或 I2CEN=0。当 PEC 使能时，发送 ACK 还是 NACK 与 NACKEN 值无关。当 SS=1 时，且 OUERR 位置 1，NACKEN 的值会被忽略，并且发送 NACK。
14	STOP	I2C 总线上产生一个 STOP 结束信号。 该位由软件置 1，并在 I2CEN=0 或检测到 STOP 信号时由硬件清零。 0：不发送 STOP。 1：发送 STOP。
13	START	I2C 总线上产生一个 START 信号 该位由软件置 1，并在从机地址发送后由硬件清零。当仲裁丢失时，或发生超时错误，或 I2CEN=0 时，该位也可以由硬件清零。将 I2C_STATC 寄存器中 ADDSENDC 位置 1 可以软件清除该位。 0：不发送 START。 1：发送 START。
12	HEAD10R	在主机接收模式下仅执行 10 位地址头读操作。 0：主机发送 10 位从机地址读序列为 START + 10 位地址头（写） + 第二个地址字节 + RESTART + 10 位地址头（读）。 1：主机寻址读序列为 RESTART + 10 位地址头（读）。 **注意：**当 START 位置 1 时，该位不能被修改。
11	ADD10EN	主机模式下使能 10 位寻址模式 0：主机工作在 7 位寻址模式下。 1：主机工作在 10 位寻址模式下。 **注意：**当 START 位置 1 时，该位不能被修改。
10	TRDIR	主机模式下传输方向 0：主机发送 1：主机接收

注意：当 START 位置 1 时，该位不能被修改。

9:0	SADDRESS[9:0]	待发送的从机地址 SADDRESS[9:8]：从机地址 9:8 位。 如果 ADD10EN = 0，该位域无效。 如果 ADD10EN = 1，将该位域写入待发送从机地址的 9:8 位。 SADDRESS[7:1]：从机地址 7:1 位。 如果 ADD10EN = 0，在这些位写入待发送 7 位从机地址。 如果 ADD10EN = 1，在这些位写入待发送从机地址的 7:1 位。 SADDRESS0：从机地址 0 位。 如果 ADD10EN = 0，这些位无效。 如果 ADD10EN = 1，在这些位写入待发送从机地址的 0 位。 **注意：**当 START 位置 1 时，该位不能被修改。

21.4.3. 从机地址寄存器 0（I2C_SADDR0）

地址偏移：0x08
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ADDRESSEN	I2C 地址使能 0：I2C 地址禁能。 1：I2C 地址使能。
14:11	保留	必须保持复位值。
10	ADDFORMAT	I2C 从机地址模式 0：7 位地址。 1：10 位地址。 **注意：**当 ADDRESSEN =1 时，该位不能被改写。
9:8	ADDRESS[9:8]	10 位地址的最高两位 **注意：**当 ADDRESSEN =1 时，该位不能被改写。
7:1	ADDRESS[7:1]	7 位地址或者 10 位地址的第 7-1 位 **注意：**当 ADDRESSEN =1 时，该位不能被改写。

0	ADDRESS0	10 位地址的第 0 位 **注意：**当 ADDRESSEN =1 时，该位不能被改写。

21.4.4. 从机地址寄存器 1（I2C_SADDR1）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32位）访问。

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ADDRESS2EN	I2C 第二个地址使能 0：I2C 第二个地址禁能 1：I2C 第二个地址使能
14:11	保留	必须保持复位值。
10:8	ADDMSK2[2:0]	ADDRESS2[7:1]掩码 定义接收到的地址哪些位需要与 ADDRESS2[7:1]进行比较，哪些位屏蔽（不比较）。 000：不屏蔽，所有的位都进行比较。 n（001~110）：ADDRESS2[n:0]屏蔽。ADDRESS2[7:n+1]需要进行比较。 111：ADDRESS2[7:1]屏蔽。对于接收到的所有 7 位地址都会进行 ACK 应答，保留地址（0b0000xxx 和 0b1111xxx）除外。 **注意：**当 ADDRESS2EN =1 时，该位不能被改写。如果 ADDMSK2 不等于 0，即使所有位都匹配，I2C 保留地址（0b0000xxx 和 0b1111xxx）也不会进行 ACK 应答。
7:1	ADDRESS2[7:1]	I2C 从机的第二个地址 **注意：**当 ADDRESS2EN =1 时，该位不能被改写。
0	保留	必须保持复位值。

21.4.5. 时序寄存器（I2C_TIMING）

地址偏移：0x10
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

PSC[3:0]	保留	SCLDELY[3:0]	SDADELY[3:0]
rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw								rw							

位/位域	名称	描述
31:28	PSC[3:0]	时序预分频 为了生成用于数据建立和数据保持的计数器的时钟周期t_{PSC}，这些位用于配置 I2CCLK 时钟预分频。t_{PSC}也用于 SCL 高电平和低电平计数器。 $t_{PSC}=(PSC+1)*t_{I2CCLK}$。
27:24	保留	必须保持复位值。
23:20	SCLDELY[3:0]	数据建立时间 这些位用于在 SDA 边沿和 SCL 上升沿之间生成延时$t_{SCLDELY}$。在主机模式下和在从机模式下 SS=0 时，在$t_{SCLDELY}$期间 SCL 线被拉低。 $t_{SCLDELY}=(SCLDELY+1)*t_{PSC}$。
19:16	SDADELY[3:0]	数据保持时间 这些位用于在 SCL 下降沿和 SDA 边沿之间生成延时$t_{SDADELY}$。在主机模式下和在从机模式下 SS=0 时，在$t_{SDADELY}$期间 SCL 线被拉低。 $t_{SDADELY}=SDADELY*t_{PSC}$。
15:8	SCLH[7:0]	SCL 高电平周期 SCL 高电平周期可以通过配置这些位来产生。 $t_{SCLH}=(SCLH+1)*t_{PSC}$。 **注意：**这些位只能用于主机模式。
7:0	SCLL[7:0]	SCL 低电平周期 SCL 低电平周期可以通过配置这些位来产生。 $t_{SCLL}=(SCLL+1)*t_{PSC}$。 **注意：**这些位只能用于主机模式。

21.4.6. 超时寄存器（I2C_TIMEOUT）

地址偏移：0x14
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTOEN	保留			BUSTOB[11:0]											
rw				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEN	保留		TOIDLE	BUSTOA[11:0]											
rw			rw	rw											

位/位域	名称	描述
31	EXTOEN	时钟信号延展超时使能 当 SCL 累计拉低时间大于 $t_{LOW:EXT}$ 时，将会产生超时错误，$t_{LOW:EXT}$=(BUSTOB+1)*2048*t_{I2CCLK}。 0：时钟信号延展超时检测禁能。 1：时钟信号延展超时检测使能。
30:28	保留	必须保持复位值。
27:16	BUSTOB[11:0]	总线超时 B 配置累积时钟延展超时。在主机模式下，检测主机累计时钟低电平延展时间 $t_{LOW:MEXT}$。从机模式下，检测从机累计时钟低电平延展时间$t_{LOW:SEXT}$。 $t_{LOW:EXT}$=(BUSTOB+1)*2048*t_{I2CCLK}。 **注意：**该位域仅在 EXTOEN = 0 时可以被修改。
15	TOEN	时钟超时使能 当 TOIDLE = 0，SCL 拉低时间大于$t_{TIMEOUT}$或当 TOIDLE = 1，SCL 拉低时间大于 t_{IDLE}，将检测到超时错误。 0：SCL 超时检测禁能 1：SCL 超时检测使能
14:13	保留	必须保持复位值。
12	TOIDLE	空闲时钟超时检测 0：BUSTOA 用于检测 SCL 低电平超时。 1：BUSTOA 用于检测 SCL 和 SDA 高电平超时（总线空闲条件）。 **注意：**该位域仅在 TOEN = 0 时可以被改写。
11:0	BUSTOA[11:0]	总线超时 A 当 TOIDLE=0 时，$t_{TIMEOUT}$=(BUSTOA+1)*2048*t_{I2CCLK} 当 TOIDLE=1 时，t_{IDLE}=(BUSTOA+1)*4*t_{I2CCLK} **注意：**该位域仅在 TOEN = 0 时可以被改写。

21.4.7. 状态寄存器（I2C_STAT）

地址偏移：0x18

复位值：0x0000 0001

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								READDR[6:0]							TR
											r				r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2CBSY	保留	SMBALT	TIMEOUT	PECERR	OUERR	LOSTARB	BERR	TCR	TC	STPDET	NACK	ADDSEND	RBNE	TI	TBE
r		r	r	r	r	r	r	r	r	r	r	r	r	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:17	READDR[6:0]	从机模式下接收到的匹配地址 当 ADDSEND 置 1 时，这些位用于存储接收到的地址。在 10 位地址情况下，READDR[6:0]存储 10 位地址头和地址的最高两位。
16	TR	I2C 在从机模式下作为发送端还是接收端 该位在 ADDSEND 位置 1 时更新。 0：接收端 1：发送端
15	I2CBSY	忙标志 该位在硬件检测到 START 信号时置 1。在 STOP 信号后硬件清 0。当 I2CEN=0 时，由硬件清零。 0：无 I2C 通讯 1：I2C 正在通讯
14	保留	必须保持复位值。
13	SMBALT	SMBus 报警 当 SMBHAEN=1，SMBALTEN=1 且在 SMBA 引脚检测到 SMBALERT 事件（下降沿）时，该位由硬件置 1。SMBALTC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：在 SMBA 引脚上检测到 SMBALERT 事件。 1：在 SMBA 引脚上未检测到 SMBALERT 事件。
12	TIMEOUT	超时标志 当发生超时或延展时钟超时，该位将置 1。TIMEOUTC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：无超时或延展时钟超时发生。 1：发生超时或延展时钟超时。
11	PECERR	PEC 错误 当接收到的 PEC 字节与 I2C_PEC 寄存器中的内容不匹配时，该位置 1。然后将自动发生 NACK。PECERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：接收到 PEC 与 I2C_PEC 的内容匹配。 1：接收到 PEC 与 I2C_PEC 的内容不匹配，此时 I2C 将忽略 NACKEN 位的值，并直接发送 NACK。
10	OUERR	从模式下上溢 / 下溢错误 在从机模式下且 SS=1，当发生上溢 / 下溢错误时，该位置 1。OUERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0：未发生上溢 / 下溢错误。 1：发生上溢 / 下溢错误。
9	LOSTARB	仲裁丢失 LOSTARBC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。

0：无仲裁丢失。
1：发生仲裁丢失，I2C 模块返回从机模式。

8 BERR 总线错误
当 I2C 总线上发生了预料之外的 START 信号或 STOP 信号时，将产生总线错误，该位将置 1。BERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。
0：无总线错误。
1：发生了总线错误。

7 TCR 传输完成重载
当 RELOAD=1 且 BYTENUM[7:0]个字节传输完成时，该位置 1。在 BYTENUM[7:0]写入一个非零值可以软件清零该位。
0：当 RELOAD=1 时，BYTENUM[7:0]个字节传输未完成。
1：当 RELOAD=1 时，BYTENUM[7:0]个字节传输完成。

6 TC 主机模式下传输完成
当 RELOAD=0，AUTOEND=0 且 BYTENUM[7:0]个字节传输完成时，该位置 1。当 START 位或 STOP 位置 1 时该位清零。
0：BYTENUM[7:0]个字节传输未完成。
1：BYTENUM[7:0]个字节传输完成。

5 STPDET 总线上检测到 STOP 信号
当在总线上检测到 STOP 信号时，主机和从机的该位由硬件置 1。STPDETC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。
0：未监测到 STOP 结束位。
1：监测到 STOP 结束位。

4 NACK 接收到 NACK 应答
当接收到 NACK 时，该位置 1。NACKC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。
0：接收到 ACK。
1：接收到 NACK。

3 ADDSEND 从机模式下接收到的地址与自身地址匹配
当接收到的地址与使能的从机地址之一匹配时，该位由硬件置 1。ADDSENDC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。
0：接收到的地址不匹配。
1：接收到的地址匹配。

2 RBNE 接收期间 I2C_RDATA 非空
当接收到的数据移入 I2C_RDATA 寄存器时，该位置 1。读 I2C_RDATA 可清除该位。
0：I2C_RDATA 空。
1：I2C_RDATA 非空，软件可以读。

1 TI 发送中断
当 I2C_TDATA 为空且 I2C 已经做好发送数据准备时，该位置 1。在下一个待发送字节写入 I2C_TDATA 寄存器时该位清零。当 SS=1 时，可由软件将该位置 1 来产生 TI 事件（TIE=1 时为中断，DENT=1 时为 DMA 请求）。

0：I2C_TDATA 非空或者 I2C 还未做好发送数据准备。
1：I2C_TDATA 空且 I2C 已经做好发送数据准备。

0	TBE	发送期间 I2C_TDATA 空 当 I2C_TDATA 寄存器为空，该位置 1。当下一个待发送数据写入 I2C_TDATA 寄存器时，该位清零。可以软件将该位置 1 来清空 I2C_TDATA 寄存器。 0：I2C_TDATA 非空。 1：I2C_TDATA 空。

21.4.8. 状态清除寄存器（I2C_STATC）

地址偏移：0x1C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		SMBALTC	TIMEOUTC	PECERRC	OUERRC	LOSTARBC	BERRC	保留		STPDETC	NACKC	ADDSENDC	保留		
		w	w	w	w	w	w			w	w	w			

位/位域	名称	描述
31:14	保留	必须保持复位值。
13	SMBALTC	SMBus 报警标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 SMBALT 位。
12	TIMEOUTC	TIMEOUT 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 TIMEOUT 位。
11	PECERRC	PEC 错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 PECERR 位。
10	OUERRC	上溢 / 下溢标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 OUERR 位。
9	LOSTARBC	仲裁丢失标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 LOSTARB 位。
8	BERRC	总线错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 BERR 位。
7:6	保留	必须保持复位值。
5	STPDETC	停止位检测标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 STPDET 位。

4	NACKC	NACK 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 NACK 位。
3	ADDSENDC	地址匹配标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 ADDSEND 位。
2:0	保留	必须保持复位值。

21.4.9. PEC 寄存器（I2C_PEC）

地址偏移：0x20
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	PECV[7:0]	在 PEC 使能时，由硬件计算出来的 PEC 值。 当 I2CEN = 0 时，PECV 由硬件清零。

21.4.10. 接收数据寄存器（I2C_RDATA）

地址偏移：0x24
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	RDATA[7:0]	接收到的数据

21.4.11. 发送数据寄存器（I2C_TDATA）

地址偏移：0x28
复位值：0x0000 0000

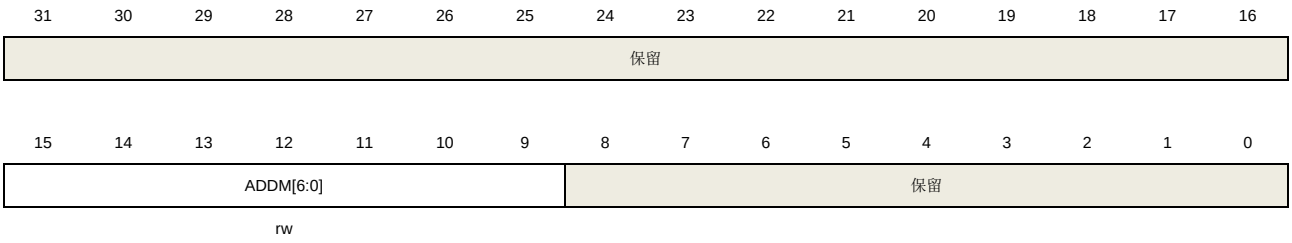
该寄存器只能按字（32 位）访问。

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TDATA[7:0]	发送的数据

21.4.12. 控制寄存器 2（I2C_CTL2）

地址偏移：0x90
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:9	ADDM[6:0]	定义 ADDRESS[7:1]的哪些位和接收到的地址进行比较，哪些位不比较。ADDM[6:0]中设置为 1 的位使能 ADDRESS[7:1]中的相应位与接收到的地址进行比较，设置为 0 的位则忽略（此时接收到的地址在该位可以为 0 或 1）。
8:0	保留	必须保持复位值。

22. 串行外设接口/片上音频接口（SPI/I2S）

22.1. 简介

SPI/I2S模块可以通过SPI协议或I2S音频协议与外部设备进行通信。

串行外设接口（Serial Peripheral Interface，缩写为SPI）提供了基于SPI协议的数据发送和接收功能，可以工作于主机或从机模式。SPI接口支持具有硬件CRC计算和校验的全双工和单工模式。只有SPI0支持SPI四线主机模式。

片上音频接口（Inter-IC Sound，缩写为I2S）支持四种音频标准，分别是I2S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。它可以在四种模式下运行，包括主机发送模式，主机接收模式，从机发送模式和从机接收模式。

22.2. 主要特性

22.2.1. SPI 主要特性

- 具有全双工和单工模式的主从操作。
- 16位宽度，独立的发送和接收缓冲区（只有SPI1）。
- 32位宽度，独立的发送和接收FIFO（只有SPI0）。
- 8位或16位数据帧格式（只有SPI1）。
- 4位到16位的数据帧格式（只有SPI0）。
- 低位在前或高位在前的数据位顺序。
- 软件和硬件NSS管理。
- 硬件CRC计算、发送和校验。
- 发送和接收支持DMA模式。
- 支持SPI TI模式。
- 支持SPI NSS脉冲模式。
- 支持SPI四线功能的主机模式（只有SPI0）。

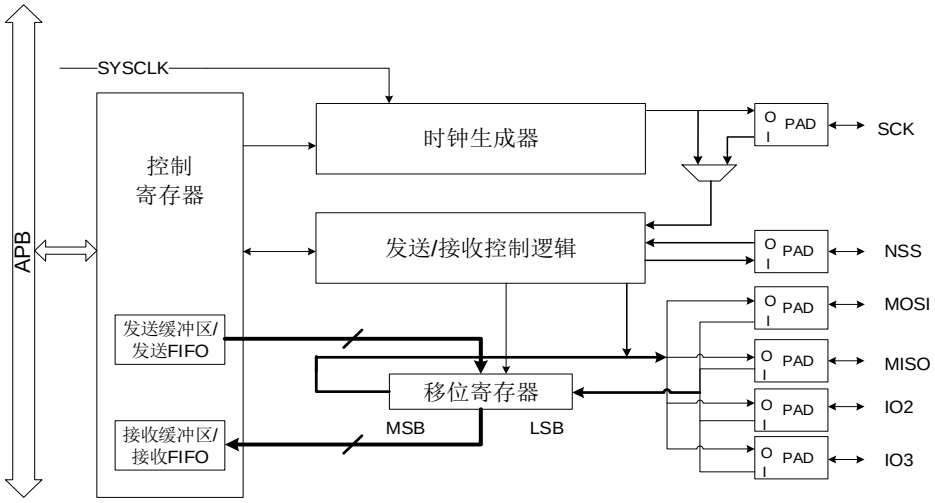
22.2.2. I2S 主要特性

- 具有发送和接收功能的主从操作。
- 支持四种I2S音频标准：飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。
- 数据长度可以为16位，24位和32位。
- 通道长度为16位或32位。
- 16位缓冲区用于发送和接收。
- 通过I2S时钟分频器，可以得到8 kHz到192 kHz的音频采样频率。
- 可编程空闲状态时钟极性。
- 可以输出主时钟（MCK）。
- 发送和接收支持DMA功能。

22.3. SPI 功能说明

22.3.1. SPI 结构框图

图 22-1. SPI 结构框图



22.3.2. SPI 信号线描述

常规配置（非 SPI 四线模式）

表 22-1. SPI 信号描述

引脚名称	方向	描述
SCK	I/O	主机：SPI 时钟输出 从机：SPI 时钟输入
MISO	I/O	主机：数据接收线 从机：数据发送线 主机双向线模式：不使用 从机双向线模式：数据发送和接收线
MOSI	I/O	主机：数据发送线 从机：数据接收线 主机双向线模式：数据发送和接收线 从机双向线模式：不使用
NSS	I/O	软件 NSS 模式：不使用 主机硬件 NSS 模式：NSSDRV=1 时，为 NSS 输出，适用于单主机模式；NSSDRV=0 时，为 NSS 输入，适用于多主机模式。 从机硬件 NSS 模式：为 NSS 输入，作为从机的片选信号。

SPI 四线配置

SPI默认配置为单线模式，当SPI_QCTL中的QMOD位置1时，配置为SPI四线模式（只适用于

SPI0）。SPI四线模式只能工作在主机模式。

通过配置SPI_QCTL中的IO23_DRV位，在常规非四线SPI模式下，软件可以驱动IO2引脚和IO3引脚为高电平。

在SPI四线模式下，SPI通过以下6个引脚与外部设备连接：

表 22-2. SPI 四线信号描述

引脚名称	方向	描述
SCK	O	SPI 时钟输出
MOSI	I/O	发送或接收数据 0
MISO	I/O	发送或接收数据 1
IO2	I/O	发送或接收数据 2
IO3	I/O	发送或接收数据 3
NSS	O	NSS 输出

22.3.3. SPI 时序和数据帧格式

SPI_CTL0寄存器中的CKPL位和CKPH位决定了SPI时钟和数据信号的时序。CKPL位决定了空闲状态时SCK的电平，CKPH位决定了第一个或第二个时钟跳变沿为有效采样边沿。在TI模式下，这两位没有意义。

在SPI0常规模式中，通过SPI_CTL1中的DZ[3:0]位域配置数据长度，可以设置为4位至16位。该设置不仅适用于数据的发送也适用于数据的接收。不论设置的数据长度是多少，对FIFO的读访问必须与SPI_CTL1寄存器中的BYTEN位设置的对齐。在SPI四线模式下，数据长度固定为8位。

同样，通过设置SPI_CTL0中的LF位可以配置数据顺序，当LF=1时，SPI先发送LSB位，当LF=0时，则先发送MSB位。在TI模式中，数据顺序固定为先发MSB位。

当访问SPI_DATA寄存器时，数据帧总是右对齐成一个字节（如果数据长度小于或等于一个字节）或一个半字。通讯时，只有数据长度内的位会随时钟输出。

图 22-2. SPI0 常规模式下的时序图

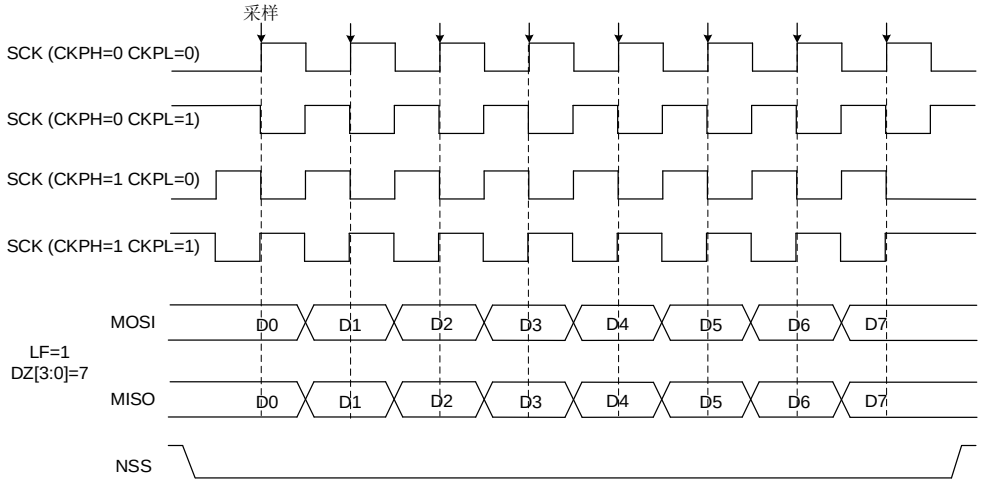
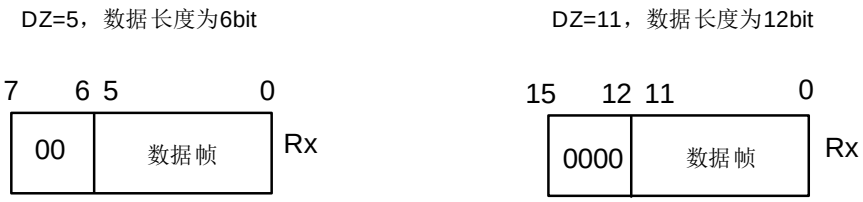


图 22-3. SPI0 数据帧右对齐示意图



在SPI1常规模式中，通过SPI_CTL0中的FF16位配置数据长度，当FF16=1时，数据长度为16位，否则为8位。

通过设置SPI_CTL0中的LF位可以配置数据顺序，当LF=1时，SPI1先发送LSB位，当LF=0时，则先发送MSB位。在TI模式中，数据顺序固定为先发MSB位。

图 22-4. SPI1 常规模式下的时序图

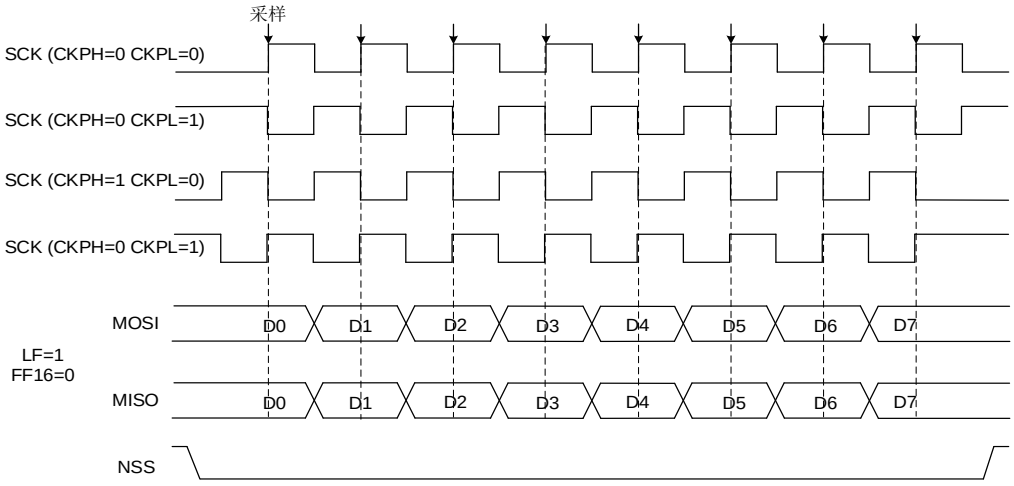
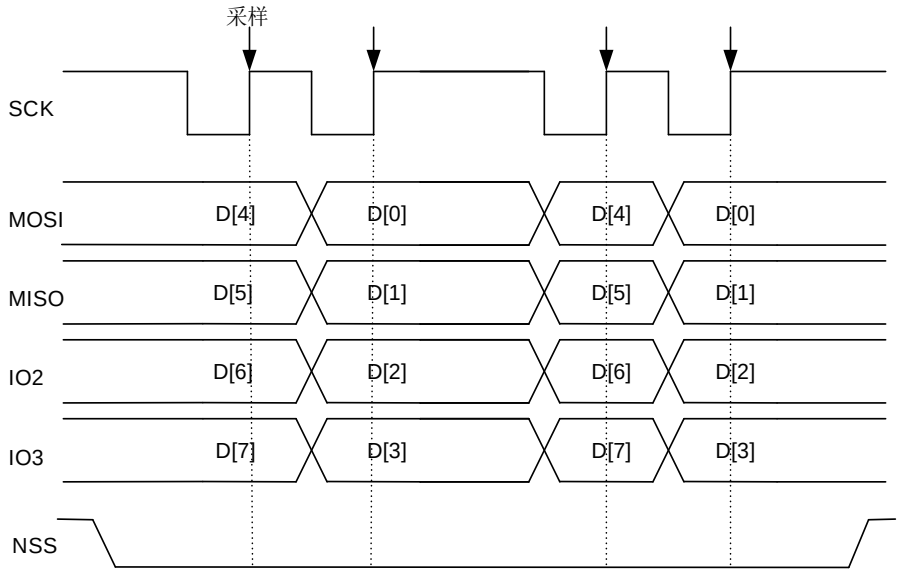


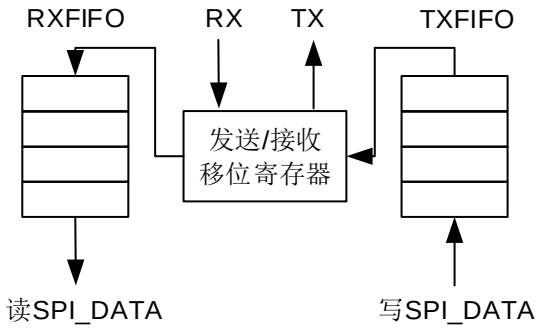
图 22-5. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0)



22.3.4. 独立发送和接收缓冲区

独立的32位的接收缓冲区（RXFIFO）和发送缓冲区（TXFIFO）分别用于SPI数据传输的不同方向，它们使得SPI可以连续工作（只适用于SPI0）。

图 22-6. 发送/接收缓冲区



当当前TXFIFO的存储量小于或等于整体存储能力的一半时，TXFIFO被视为空(1)并且此时TBE被硬件置1。当TBE位置位时，向SPI_DATA寄存器写数据，会把数据存入发送FIFO的末尾。当当RXFIFO被视为非空(2)时硬件将RBNE位置1。当RBNE位置位时，从SPI_DATA寄存器读数据，将从接收FIFO获得最早数据。

注意：

（1）对于SPI0，TXFIFO空意味着TXFIFO当前的存储量小于或等于TXFIFO整体存储能力的一半。TXFIFO满的意义与之相反。所以，当数据长度不大于8位时，TXFIFO最多能存储3个数据帧。如果下文出现TXFIFO空或者满，如无特殊说明，意义与这里说明的相同。

（2）对于SPI0，RXFIFO空的意义分为以下两种情况：如果SPI_CTL1中BYTEN位为1时，RXFIFO空意味着当前RXFIFO的存储量小于RXFIFO整体存储能力的四分之一。此时，当数据长度不大于8位时，RXFIFO最多可以存储4个数据帧。如果SPI_CTL1中BYTEN位为0时，RXFIFO空意味着当前RXFIFO的存储量小于RXFIFO整体存储能力的一半。RXFIFO满的意义与之相反。如果下文出现RXFIFO空或者满，如无特殊说明，意义与这里说明的相同。

数据合并（仅适用于 SPI0）

在SPI_CTL1寄存器中DZ[3:0]配置传输数据位宽为8位或者小于8位的情况下，通过配置SPI_CTL1寄存器中BYTEN位为0，开启数据合并传输模式功能。在配置SPI_CTL1寄存器中DZ[3:0]配置传输数据位宽为小于等于8位时，该功能可以实现当对SPI_DATA寄存器进行16位写访问时，两个数据帧的发送是并行方式而不是串行方式。同样的，在接收端接收器通过对SPI_DATA的一次16位读访问，获取这两个数据帧，并且这两帧数据在接收时，仅会产生一个RBNE事件。

注意：当被传输的数据为奇数个字节时，在发送端，需要用8位访问SPI_DATA，发出最后一个数据帧。在接收端，为了产生最后一个字节的RBNE事件，接收器必须在接收最后一个数据帧时，改变BYTEN位。

22.3.5. NSS 功能

从机模式

当配置为从机模式（MSTMOD=0）时，在硬件NSS模式（SWNSSEN = 0）下，SPI从NSS引脚获取NSS电平，在软件NSS（SWNSSEN = 1）下，SPI根据SWNSS位得到NSS电平。只有当NSS为低电平时，发送或接收数据。在软件NSS模式下，不使用NSS引脚。

表 22-3. 从机模式 NSS 功能

模式	寄存器配置	描述
从机硬件 NSS 模式	MSTMOD = 0 SWNSSEN = 0	SPI 从机 NSS 电平从 NSS 引脚获取。
从机软件 NSS 模式	MSTMOD = 0 SWNSSEN = 1	SPI 从机 NSS 电平由 SWNSS 位决定。 SWNSS = 0：NSS 电平为低 SWNSS = 1：NSS 电平为高

主机模式

在主机模式（MSTMOD=1）下，如果应用程序使用多主机连接方式，NSS可以配置为硬件输入模式（SWNSSEN=0，NSSDRV=0）或者软件模式（SWNSSEN=1）。一旦NSS引脚（在硬件NSS模式下）或SWNSS位（在软件NSS模式下）被拉低，SPI将自动进入从机模式，并且产生主机配置错误，CONFERR位置1。

如果应用程序希望使用NSS引脚控制SPI从设备，NSS应该配置为硬件输出模式（SWNSSEN=0，NSSDRV=1）。使能SPI之后，NSS保持高电平，当发送或接收过程开始时，NSS变为低电平。当禁用SPI时，NSS变为高电平。

应用程序可以使用一个通用I/O口作为NSS引脚，以实现更加灵活的NSS应用。

表 22-4. 主机模式 NSS 功能

模式	寄存器配置	描述
主机硬件 NSS 输出模式	MSTMOD = 1 SWNSSEN = 0 NSSDRV=1	适用于单主机模式，主机使用 NSS 引脚控制 SPI 从设备，此时 NSS 配置为硬件输出模式。使能 SPI 后 NSS 为低电平。
主机硬件 NSS 输入模式	MSTMOD = 1 SWNSSEN = 0 NSSDRV=0	适用于多主机模式，此时 NSS 配置为硬件输入模式，一旦 NSS 引脚被拉低，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。
主机软件 NSS 模式	MSTMOD = 1 SWNSSEN = 1 SWNSS = 0 NSSDRV：不要求	适用于多主机模式，一旦 SWNSS = 0，SPI 将自动进入从机模式，并且产生主机配置错误，CONFERR 位置 1。
	MSTMOD = 1 SWNSSEN = 1 SWNSS = 1 NSSDRV：不要求	从机可以使用硬件或软件 NSS 模式

22.3.6. SPI 运行模式

表 22-5. SPI 运行模式

模式	描述	寄存器配置	数据引脚用法
MFD	全双工主机模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：发送 MISO：接收
MTU	单向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：发送 MISO：不使用
MRU	单向线连接主机接收模式	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN：不要求	MOSI：不使用 MISO：接收
MTB	双向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI：发送 MISO：不使用
MRB	双向线连接主机接收模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 0	MOSI：接收 MISO：不使用
SFD	全双工从机模式	MSTMOD = 0	MOSI：接收

模式	描述	寄存器配置	数据引脚用法
		RO = 0 BDEN = 0 BDOEN：不要求	MISO：发送
STU	单向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：不使用 MISO：发送
SRU	单向线连接从机接收模式	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN：不要求	MOSI：接收 MISO：不使用
STB	双向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 1	MOSI：不使用 MISO：发送
SRB	双向线连接从机接收模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 0	MOSI：不使用 MISO：接收

图 22-7. 典型的全双工模式连接

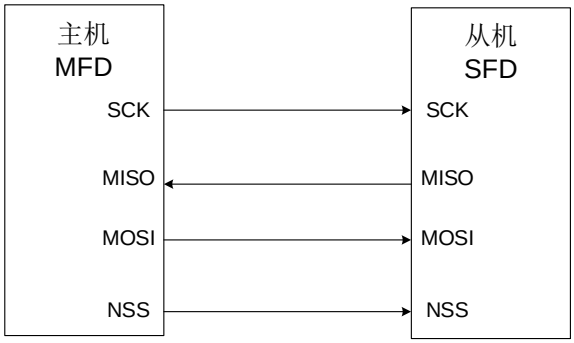


图 22-8. 典型的单工模式连接（主机：接收，从机：发送）

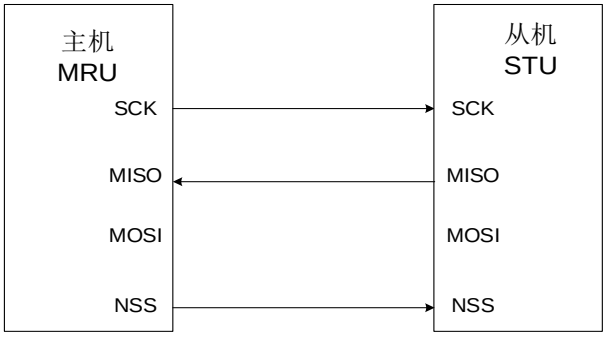


图 22-9. 典型的单工模式连接（主机：只发送，从机：接收）

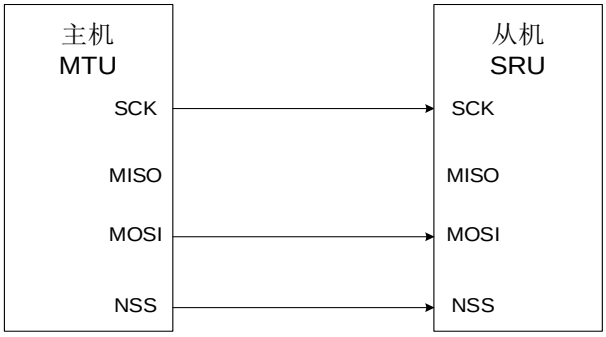
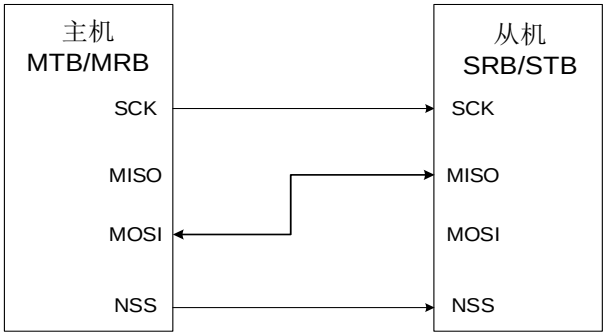


图 22-10. 典型的双向线连接



SPI 初始化流程

SPI0：

1. 如果工作在主机模式或从机TI模式，配置SPI_CTL0中的PSC[2:0]位来生成预期波特率的SCK信号，或配置TI模式下的Td时间。否则，忽略此步骤。
2. 配置时钟时序（SPI_CTL0中的CKPL位和CKPH位）。
3. 配置帧格式（SPI_CTL0中的LF位）。
4. 配置数据格式（SPI_CTL1中的DZ[3:0]位域）和SPI_DATA的访问方式（SPI_CTL1中的BYTEN）。
5. 按照上文***NSS 功能***的描述，根据应用程序的需求，配置NSS模式（SPI_CTL0中的SWNSSEN位和NSSDRV位）。
6. 如果工作在TI模式，需要将SPI_CTL1中的TMOD位置1，否则，忽略此步骤。
7. 如果工作在 NSSP 模式，需要将 SPI_CTL1 中的 NSSP 位置 1，否则，忽略此步骤。
8. 根据***表22-5. SPI运行模式***，配置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 根据应用程序的需求，配置TXDMA_ODD和RXDMA_ODD位。
10. 如果工作在SPI四线模式，需要将SPI_QCTL中的QMOD位置1，如果不是，则忽略此步骤。
11. 使能SPI（将SPIEN位置1）。

注意：在通信过程中，不应更改CKPH、CKPL、MSTMOD、PSC[2:0]、LF、DZ[3:0]位。

SPI1：

在发送或接收数据之前，应用程序应遵循如下的SPI初始化流程：

1. 如果工作在主机模式或从机TI模式，配置SPI_CTL0中的PSC[2:0]位来生成预期波特率的

SCK信号，或配置TI模式下的Td时间。否则，忽略此步骤。

2. 配置数据格式（SPI_CTL0中的FF16位）。
3. 配置时钟时序（SPI_CTL0中的CKPL位和CKPH位）。
4. 配置帧格式（SPI_CTL0中的LF位）。
5. 按照上文*NSS功能*的描述，根据应用程序的需求，配置NSS模式（SPI_CTL0中的SWNSSEN位和NSSDRV位）。
6. 如果工作在TI模式，需要将SPI_CTL1中的TMOD位置1，否则，忽略此步骤。
7. 如果工作在NSSP模式，需要将SPI_CTL1中的NSSP位置1，否则，忽略此步骤。
8. 根据*表22-5. SPI运行模式*，配置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 使能SPI（将SPIEN位置1）。

注意：在通信过程中，不应更改CKPH、CKPL、MSTMOD、PSC[2:0]、LF、LF位。

SPI 基本发送和接收流程

发送流程

在完成初始化过程之后，SPI模块使能并保持在空闲状态。在主机模式下，当软件写一个数据到发送缓冲区/发送FIFO时，发送过程开始。在从机模式下，当SCK引脚上的SCK信号开始翻转，且NSS引脚电平为低，发送过程开始。所以，在从机模式下，应用程序必须确保在数据发送开始前，数据已经写入发送缓冲区/发送FIFO中。

当SPI开始发送一个数据帧时，首先将这个数据帧从数据缓冲区/发送FIFO加载到移位寄存器中，然后开始发送加载的数据。在数据帧的第一位发送之后，TBE（发送缓冲区/发送FIFO空）位置1。TBE标志位置1，说明发送缓冲区/发送FIFO为空，此时如果需要发送更多数据，软件应该继续写SPI_DATA寄存器。

在主机模式下，若想要实现连续发送功能，那么在当前数据帧发送完成前，软件应该将下一个数据写入SPI_DATA寄存器中。

接收流程

在最后一个采样时钟边沿之后，接收到的数据将从移位寄存器存入到接收缓冲区/接收FIFO，且RBNE（接收缓冲区/接收FIFO非空）位置1。软件通过读SPI_DATA寄存器获得接收的数据，此操作会自动清除RBNE标志位。在MRU和MRB模式中，为了接收下一个数据帧，硬件需要连续发送时钟信号，而在全双工主机模式（MFD）中，仅当发送缓冲区/发送FIFO非空时，硬件才接收下一个数据帧。

SPI 不同模式下的操作流程（非 SPI 四线模式，TI 模式或 NSSP 模式）

在全双工模式下，无论是MFD模式或者SFD模式，应用程序都应该监视RBNE标志位和TBE标志位，并且遵循上文描述的操作流程。

发送模式（MTU，MTB，STU或STB）与全双工模式中的发送流程类似，不同的是需要忽略RBNE位和RXORERR位。

相比于发送模式的情况，主机接收模式（MRU或MRB）与全双工的接收流程大不相同。在MRU模式或MRB模式下，在SPI使能后，SPI产生连续的SCK信号，直到SPI停止。所以，软件应该忽略TBE标志位，并且在RBNE位置1后，读出接收缓冲区/接收FIFO内的数据，否则，将会产

生接收过载错误。

除了忽略TBE标志位，且只执行上述的接收流程之外，从机接收模式（SRU或SRB）与全双工模式类似。

SPI TI 模式

SPI TI模式将NSS作为一种特殊的帧头标志信号，它的操作流程与上文描述的常规模式类似。上文描述的模式（MFD，MTU，MRU，MTB，MRB，SFD，STU，SRU，STB和SRB）都支持TI模式。但是，在TI模式中，SPI_CTL0中的CKPL位和CKPH位是没有意义的，SCK信号的采样边沿为下降沿。

图 22-11. 主机 TI 模式在不连续发送时的时序图

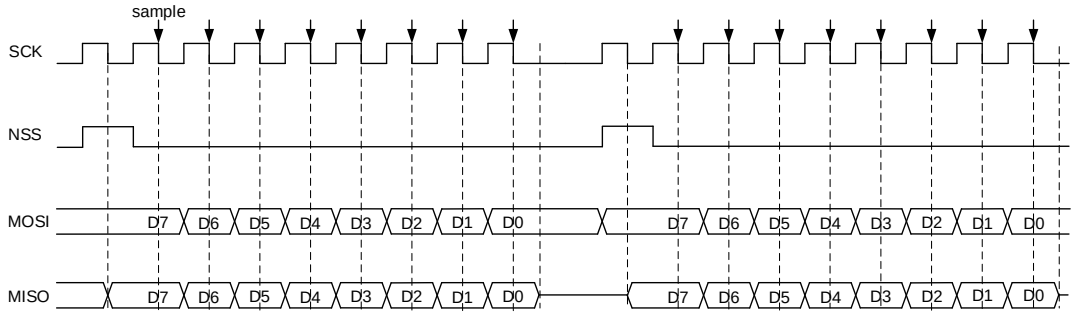
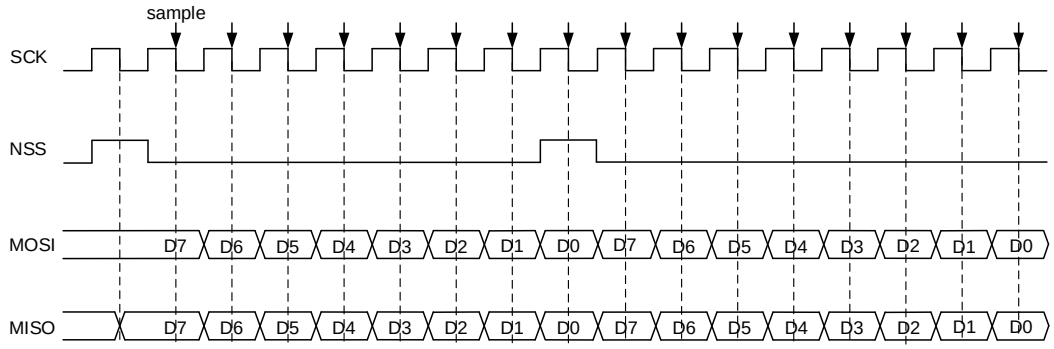
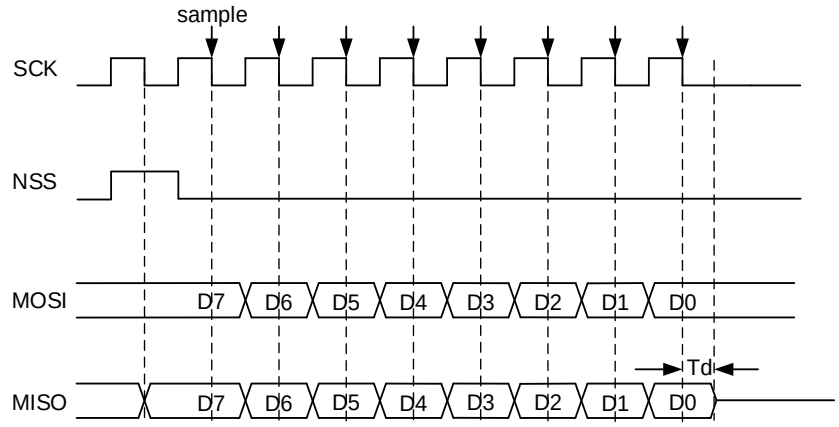


图 22-12. 主机 TI 模式在连续发送时的时序图



在主机TI模式下，SPI模块可实现连续传输或者不连续传输。如果主机写SPI_DATA的速度很快，那么就是连续传输，否则，为不连续传输。在不连续传输中，在每个字节传输前需要一个额外的时钟周期。在连续传输中，额外的时钟周期只存在于第一个字节之前，随后字节的起始时钟周期被前一个字节的最后一位的时钟周期覆盖。

图 22-13. 从机 TI 模式时序图



在从机TI模式中，在SCK信号的最后一个上升沿，从机开始发送最后一个字节的LSB位，在半位的时间之后，主机开始采集数据。为了确保主机采集到正确的数据，在释放该引脚之前，从机需要在SCK信号的下降沿之后继续驱动该位一段时间，这段时间称为T_d，T_d通过SPI_CTL0寄存器中的PSC[2:0]位来设置。

$$T_d=\frac{T_{bit}}{2}+5*T_{pclk} \quad (22\text{-}1)$$

例如，如果PSC[2:0] = 010，那么T_d数值为$9*T_{pclk}$。

在从机模式下，从机需要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

NSS 脉冲模式操作流程

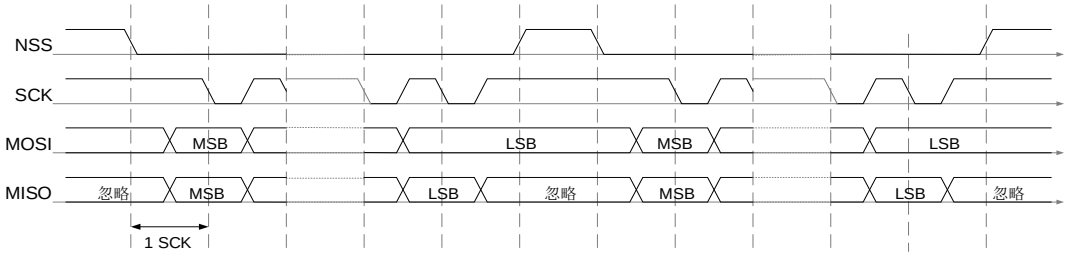
配置SPI_CTL1寄存器中的NSSP位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通SPI协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

总之：MSTMOD = 1，NSSP = 1，CKPH = 0。

当使用NSS脉冲模式时，根据内部数据发送缓冲区/发送FIFO的状态，NSS脉冲会在两个连续的数据帧之间产生，且持续时间至少为1个SCK时钟周期。如果数据发送缓冲区/发送FIFO保持为空，可能会持续多个SCK时钟周期。NSS脉冲功能专为单一的主从应用设计，支持从机锁存数据。

下图描述了NSS脉冲模式在主机连续发送时的时序图。

图 22-14. NSS 脉冲模式时序图（主机连续发送）



SPI 四线模式操作流程

SPI四线模式用于控制四线SPI flash外设。

要配置成SPI四线模式，首先要确认TBE位置1，且TRANS位清零，然后将SPI_QCTL寄存器中的QMOD位置1。在SPI四线模式，SPI_CTL0寄存器中BDEN位、BDOEN位、CRCEN位、CRCNT位、CRCNT位、RO位和LF位保持清零，DZ[3:0]位域配置数据长度为8位，且MSTMOD位置1，以保证SPI工作于主机模式。SPIEN位、PSC位、CKPL位和CKPH位根据需要进行配置。

SPI四线模式有两种运行模式：四线写模式和四线读模式，通过SPI_QCTL寄存器中的QRD位进行配置。

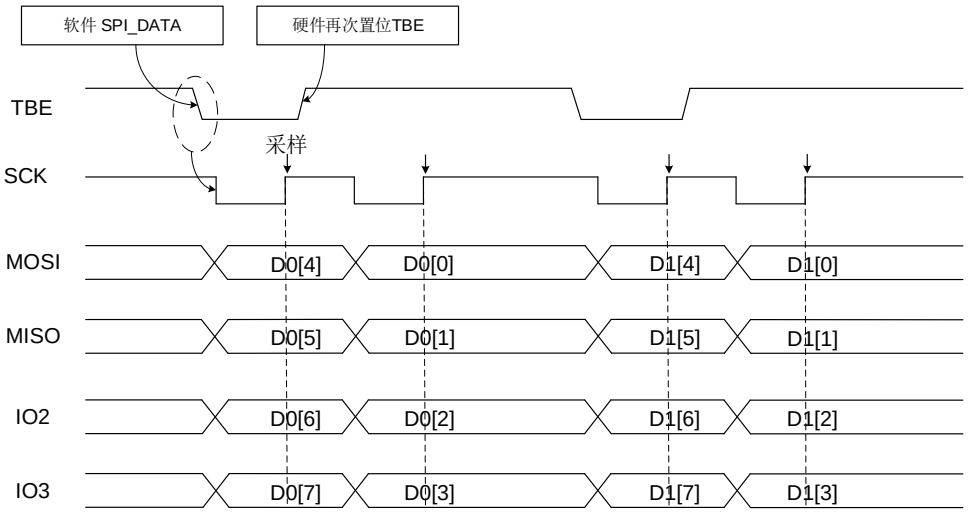
四线写模式

当SPI_QCTL寄存器中的QMOD位置1且QRD位清零时，SPI工作在四线写模式。在四线写模式中，MOSI、MISO、IO2和IO3都用作输出引脚，在SCK产生时钟信号后，一旦数据写入SPI_DATA寄存器（TBE位清零）且SPIEN位置1时，将会通过这四个引脚发送写入的数据。SPI开始数据传输之后，每发送一个数据帧都要检测TBE标志位，若不能满足条件则停止传输。

四线模式下发送操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中的时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 向SPI_DATA寄存器中写入一个字节的数据，TBE标志位将会清零；
4. 等待硬件将TBE位重新置位，然后写入下一个字节数据。

图 22-15. SPI 四线模式四线写操作时序图



四线读模式

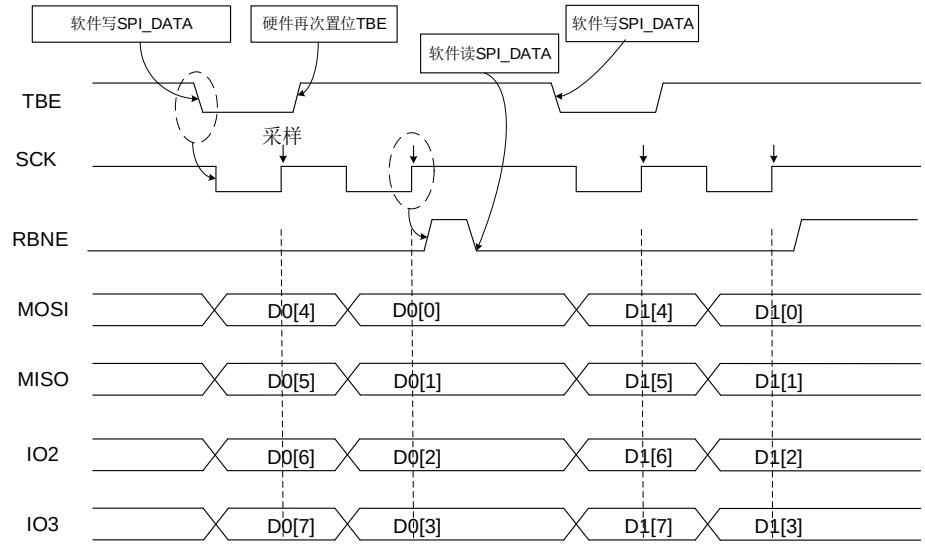
当SPI_QCTL寄存器中的QMOD位和QRD位都置1时，SPI工作在四线读模式。在四线读模式中，MOSI、MISO、IO2和IO3都用作输入引脚，一旦数据写入SPI_DATA寄存器（TBE位清零）且SPIEN位置1时，在SCK信号线产生时钟信号。写数据到SPI_DATA寄存器只是为了产生SCK时钟信号，所以可以写入任何数据。SPI开始数据传输之后，每发送一个数据帧都要检测SPIEN

位和TBE位，若条件不满足则停止传输。所以软件需要一直向SPI_DATA写空闲数据，以产生SCK时钟信号。

四线模式下接收操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位和QRD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 写任意数据（例如0xFF）到SPI_DATA寄存器；
4. 等待RBNE位置1，然后读SPI_DATA寄存器来获取接收的数据；
5. 写任意数据（例如0xFF）到SPI_DATA寄存器，以接收下一个字节数据。

图 22-16. SPI 四线模式四线读操作时序图



SPI 停止流程

不同运行模式下采用不同的流程来停止SPI功能。

MFD SFD

SPI0：

等待TXLVL[1:0]=00和TRANS=0，接着通过清零SPIEN位关闭SPI。最后，读取数据直到RXLVL[1:0]=00。

SPI1：

等待最后一个RBNE位并接收最后一个数据，等待TBE=1和TRANS=0，最后，通过清零SPIEN位关闭SPI。

MTU MTB STU STB

SPI0：

等待TXLVL[1:0]=00和TRANS=0，接着通过清零SPIEN位关闭SPI。

SPI1：

将最后一个数据写入SPI_DATA寄存器，等待TBE位置1，等待TRANS位清零，通过清零SPIEN位关闭SPI。

MRU MRB

SPI0：

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0，读取数据直到RXLVL[1:0]=00。

SPI1：

等待倒数第二个RBNE位置1，从SPI_DATA寄存器读数据，等待一个SCK时钟周期，然后通过清零SPIEN位关闭SPI。等待最后一个RBNE位置1，并从SPI_DATA读数据。

SRU SRB

SPI0：

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0，读取数据直到RXLVL[1:0]=00。

SPI1：

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0以确保当前通信过程结束。

TI模式

TI模式的停止流程与上面描述过程相同。

NSS脉冲模式

NSS脉冲模式的停止流程与上面描述过程相同。

SPI四线模式

在禁用SPI四线模式和关闭SPI功能之前，软件应该先检查：TBE位置1，TRANS位清零，SPI_QCTL中的QMOD位和SPI_CTL0中的SPIEN位清零。

22.3.7. DMA 功能

DMA功能在传输过程中将应用程序从数据读写过程中释放出来，从而提高了系统效率。

通过置位SPI_CTL1寄存器中的DMATEN位和DMAREN位，使能SPI模式的DMA功能。为了使用DMA功能，软件首先应当正确配置DMA模块，然后通过初始化流程配置SPI模块，最后使能SPI。

SPI使能后，如果DMATEN位置1，每当TBE=1时，SPI将会发出一个DMA请求，然后DMA应答该请求，并自动写数据到SPI_DATA寄存器。如果DMAREN位置1，每当RBNE=1时，发出一个DMA请求，然后DMA应答该请求，并自动从SPI_DATA寄存器读取数据。

DMA 数据合并传输

采用DMA进行数据传输，当BYTEN设置为0且DZ[3:0]配置的数据长度小于或等于8位且数据合

并模式使能时，DMA将会以16位方式访问SPI_DATA寄存器，自动完成数据的发送。

在数据合并模式使能且传输数据帧的帧数不是偶数倍的情况下，为了避免最后一次DMA传输多一帧数据的问题，需要将SPI_CTL1寄存器中TXDMA_ODD/RXDMA_ODD位为设置为1。

22.3.8. CRC 功能

SPI模块包含两个CRC计算单元：分别用于发送数据和接收数据。CRC计算单元使用SPI_CRCPOLY寄存器中定义的多项式。

通过配置SPI_CTL0中的CRCEN位使能CRC功能。对于数据线上每个发送和接收的数据，CRC单元逐位计算CRC值，计算得到的CRC值可以从SPI_TCRC寄存器和SPI_RCRC寄存器中读取。

为了传输计算得到的CRC值，应用程序需要在最后一个数据写入发送缓冲区之后，设置SPI_CTL0中的CRCNT位。在全双工模式（MFD或SFD），当SPI发送一个CRC值并且准备校验接收到的CRC值时，会将最新接收到的数据当作CRC值。在接收模式（MRB，MRU，SRU和SRB）下，在倒数第二个数据帧被接收后，软件将CRCNT位置1。在CRC校验失败时，CRCERR错误标志位将会置1。

对于SPI1，如果是8位数据长度，CRC计算基于CRC8标准进行。如果是16位数据长度，CRC计算基于CRC16标准进行。如果使能了DMA功能，软件不需要设置CRCNT位，硬件将会自动处理CRC传输和校验。

对于SPI0，只有数据长度为8位或者16位时，SPI提供CRC计算，且独立于数据长度，可以固定设置为8位或16位CRC计算。对于其他所有的数据长度，CRC无效。CRC数据交换，通常需要在数据序列结束后，再占用一个或多个数据通信的时间。例如，当设置为8位的数据长度并做16位CRC检查时，发送完整的CRC数据就要两帧。如果使能了DMA功能，硬件将会自动处理CRC传输和校验，但SPI需设置DMA发送通道和接收通道的计数器值。发送DMA计数器值为不包括CRC帧的数据帧的数量。接收DMA计数器值的配置如下：

1.全双工模式：假设SPI接收的数据量为L，当CRCL = 0且DZ = 8时，，则DMA接收通道的计数值等于L + 1，否则DMA接收通道的计数值等于L + 2。

2.只接收模式：DMA接收通道计数值只等于接收的数据量。接收数据完成后，通过软件读取SPI_RCRC寄存器的方式获取CRC值。

注意：当SPI处于从机模式且CRC功能使能时，无论SPI是否使能，CRC计算器都对输入SCK时钟敏感。只有当时钟稳定时，软件才能启用CRC，以避免错误的CRC计算。当SPI作为从机工作时，在数据阶段和CRC阶段之间，内部NSS信号需要保持低电平。

22.3.9. SPI 中断

状态标志位

- **发送缓冲区空标志位（TBE）**

当发送缓冲区为空或当前发送FIFO的存储量小于或等于总存储量的一半时，TBE置位。软件可以通过写SPI_DATA寄存器将下一个待发送数据写入发送缓冲区/发送FIFO。

■ **接收缓冲区非空标志位（RBNE）**

对于SPI0，该位根据SPI_CTL1中的BYTEN位设置：如果BYTEN=0，则当前接收FIFO的存储量大于或等于总存储量的1/2时，RBNE置位。如果BYTEN=1，则当前接收FIFO的存储量大于或等于总存储量的1/4时，RBNE置位。表示此时接收到数据，并已存入接收FIFO中，软件可以通过读SPI_DATA寄存器来读取此数据。

对于SPI1，当接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入到接收缓冲区中，软件可以通过读SPI_DATA寄存器来读取此数据。

■ **SPI通信进行中标志位（TRANS）**

TRANS位是用来指示当前传输是否正在进行或结束的状态标志位，它由内部硬件置位和清除，无法通过软件控制。该标志位不会产生任何中断。

错误标志

■ **配置错误标志（CONFERR）**

在主机模式中，CONFERR位是一个错误标志位。在硬件NSS模式中，如果NSSDRV没有使能，当NSS被拉低时，CONFERR位被置1。在软件NSS模式中，当SWNSS位为0时，CONFERR位置1。当CONFERR位置1时，SPIEN位和MSTMOD位由硬件清除，SPI关闭，设备强制进入从机模式。

在CONFERR位清零之前，SPIEN位和MSTMOD位保持写保护，从机的CONFERR位不能置1。在多主机配置中，设备可以在CONFERR位置1时进入从机模式，这意味着发生了系统控制的多主冲突。

■ **接收过载错误（RXORERR）**

在RBNE位为1时，如果再有数据被接收，RXORERR位将会置1。对于SPI1，这说明，上一帧数据还未被读出而新的数据已经接收了。对于SPI0，这说明，接收FIFO没有足够的空间来存储接收到的数据了。接收缓冲区/接收FIFO的内容不会被新接收的数据覆盖，所以新接收的数据丢失。

■ **帧错误（FERR）**

在TI从机模式下，从机也要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

■ **CRC错误（CRCERR）**

当CRCEN位置1时，SPI_RCRC寄存器中接收到的数据的CRC计算值将会和紧随着最后一帧数据后接收到的CRC值进行比较，当两者不同时，CRCERR位将会置1。

表 22-6. SPI 中断请求

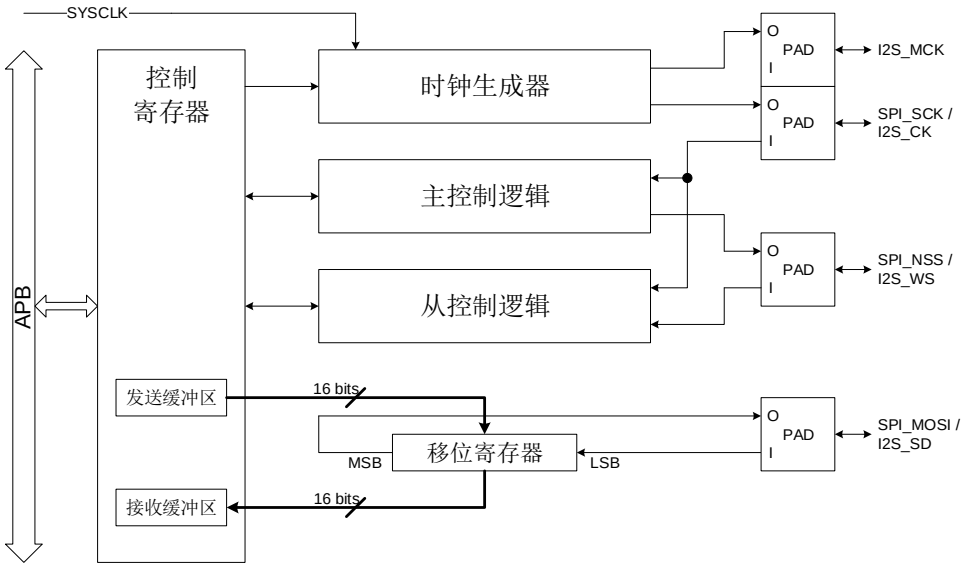
中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区/发送FIFO空	写SPI_DATA寄存器	TBEIE
RBNE	接收缓冲区/接收FIFO非空	读SPI_DATA寄存器	RBNEIE
CONFERR	配置错误	读或写 SPI_STAT 寄存器，然后写 SPI_CTL0 寄存器	ERRIE

中断事件	描述	清除方式	中断使能位
RXORERR	接收过载错误	读SPI_DATA寄存器，然后读SPI_STAT寄存器	
CRCERR	CRC错误	写0到CRCERR位	
FERR	TI模式帧错误	写0到FERR位	

22.4. I2S 功能说明

22.4.1. I2S 结构框图

图 22-17. I2S 结构框图



I2S功能有5个子模块，分别是控制寄存器、时钟生成器、主机控制逻辑、从机控制逻辑和移位寄存器。所有的用户可配置寄存器都在控制寄存器模块实现，其中包括发送缓冲区和接收缓冲区。时钟生成器用来在主机模式下生成I2S通信时钟。主机控制逻辑用来在主机模式下生成I2S_WS信号并控制通信。从机控制逻辑根据接收到的I2S_CK和I2S_WS信号来控制从机模式的通信。移位寄存器控制I2S_SD上的串行数据发送和接收。

22.4.2. I2S 信号线描述

I2S接口有4个引脚，分别是I2S_CK、I2S_WS、I2S_SD和I2S_MCK。I2S_CK是串行时钟信号，与SPI_SCK共享引脚。I2S_WS是数据帧控制信号，与SPI_NSS共享引脚。I2S_SD是串行数据信号，与SPI_MOSI共享引脚。I2S_MCK是主时钟信号，它最大可提供一个256倍于Fs的时钟频率，其中Fs是音频采样率。

22.4.3. I2S 音频标准

I2S音频标准是通过设置SPI_I2SCTL寄存器中的I2SSTD位来选择的，可以选择四种音频标准：I2S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。除PCM之外的所有标准都是两个通道（左通道和右通道）的音频数据分时复用I2S接口的，并通过I2S_WS信号来区分当前数据

属于哪个通道。对于PCM标准，I2S_WS信号表示帧同步信息。

数据长度和通道长度可以通过SPI_I2SCTL寄存器中的DTLEN位和CHLEN位来设置。由于通道长度必须大于或等于数据长度，所以有四种数据包类型可供选择。它们分别是：16位数据打包成16位数据帧格式，16位数据打包成32位数据帧格式，24位数据打包成32位数据帧格式，32位数据打包成32位数据帧格式。用于发送和接收的数据缓冲区都是16位宽度。所以，要完成数据长度为24位或32位的数据帧传输，SPI_DATA寄存器需要被访问2次；而要完成数据长度为16位的数据帧传输，SPI_DATA寄存器只需被访问1次。如需将16位数据打包成32位数据帧，硬件会自动插入16位0将16位数据扩展为32位格式。

对于所有标准和数据包类型来说，数据的最高有效位总是最先被发送的。对于所有基于两通道分时复用的标准来说，总是先发送左通道，然后是右通道。

I2S 飞利浦标准

对于I2S飞利浦标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化，I2S_WS在数据的前一个时钟开始有效。各种配置情况的时序图如下所示。

图 22-18. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

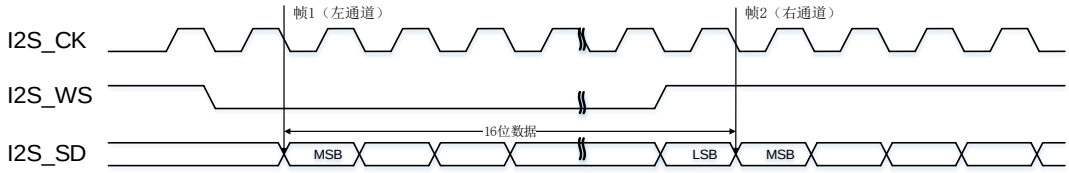
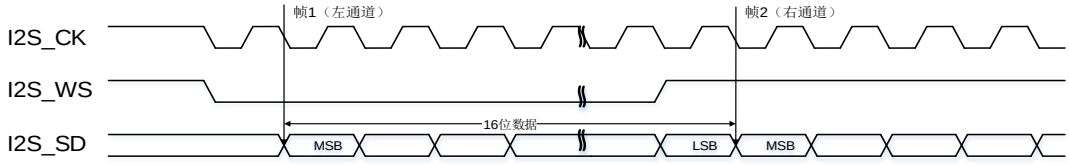


图 22-19. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1）



当 16 位数据打包成 16 位数据帧时，每完成一帧数据的传输只需要访问 SPI_DATA 寄存器一次。

图 22-20. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

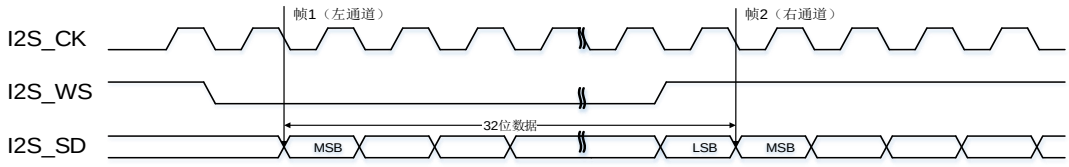
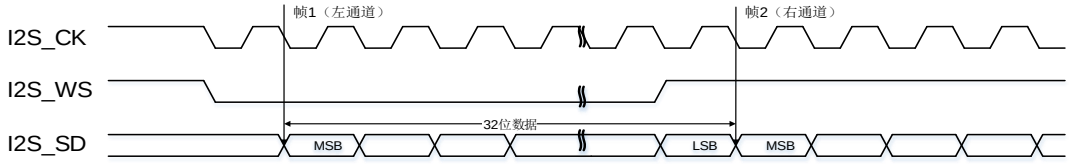


图 22-21. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1）



当32位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个32位数据，第一个写入SPI_DATA寄存器的数据应该是高16位数据，第二个数据应该是低16位数据。在接收模式下，如果要接收一个32位数据，第一个从SPI_DATA寄存器读到的数据应该是高16位数据，第二个数据应该是低16位数据。

图 22-22. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

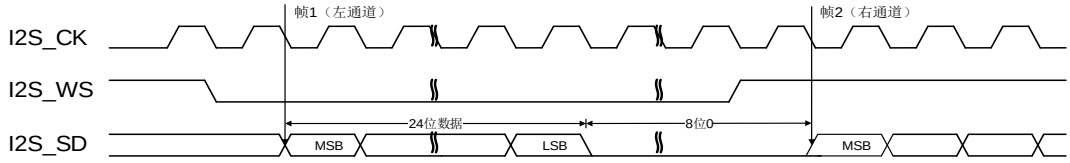
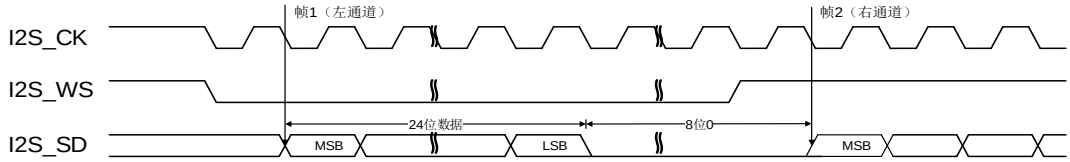


图 22-23. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=1）



当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI_DATA寄存器的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据可以是任意值。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI_DATA寄存器读到的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据全是0。

图 22-24. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

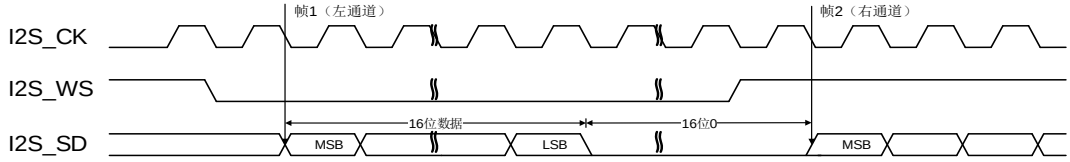
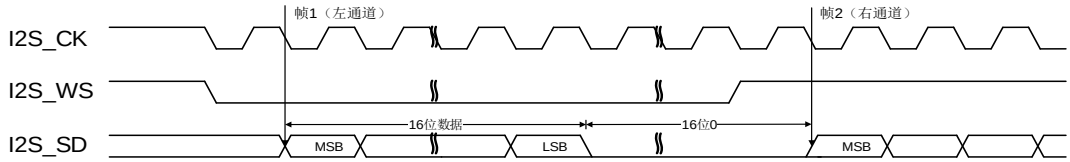


图 22-25. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问SPI_DATA寄存器一次。为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为0x0000。

MSB 对齐标准

对于MSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。SPI_DATA 寄存器的处理方式与I2S飞利浦标准完全相同。各个配置情况的时序图如下所示。

图 22-26. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

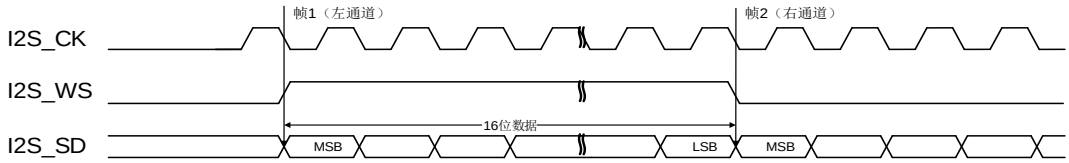


图 22-27. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=1）

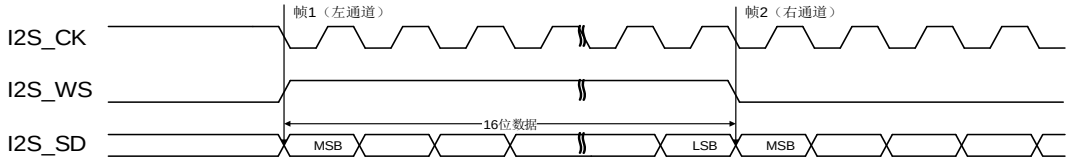


图 22-28. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

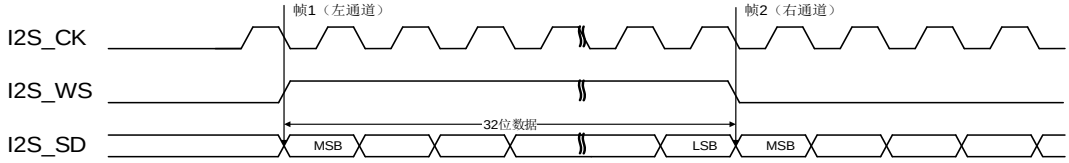


图 22-29. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=1）

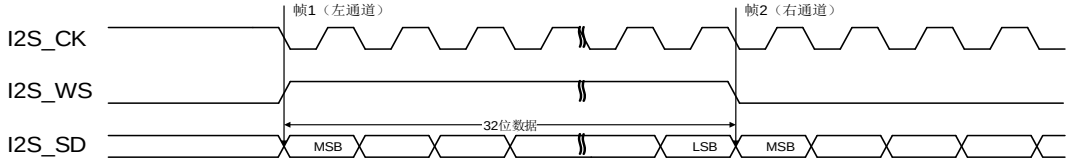


图 22-30. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

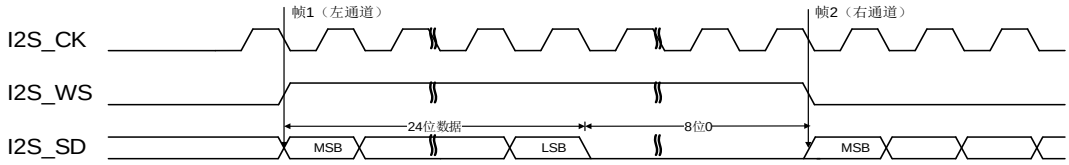


图 22-31. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）

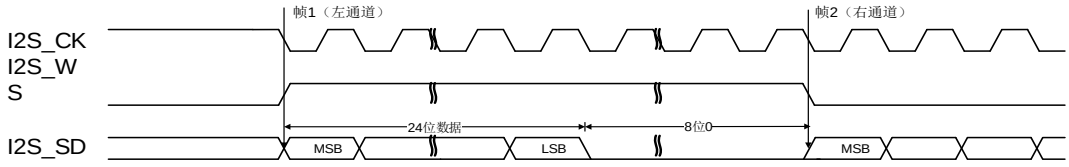


图 22-32. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

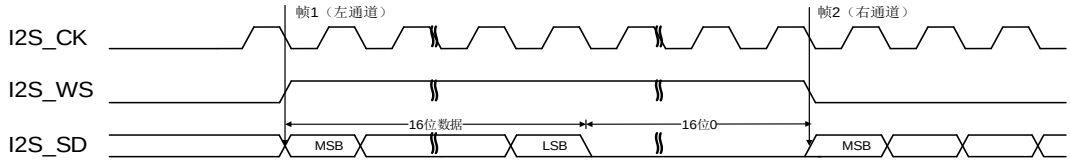
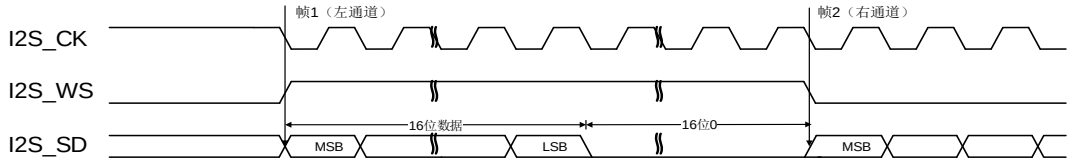


图 22-33. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



LSB 对齐标准

对于LSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。在通道长度与数据长度相同的情况下，LSB对齐标准和MSB对齐标准是完全相同的。对于通道长度大于数据长度的情况，LSB对齐标准的有效数据与最低位对齐，而MSB对齐标准的有效数据与最高位对齐。通道长度大于数据长度的各种配置情况时序图如下所示。

图 22-34. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）

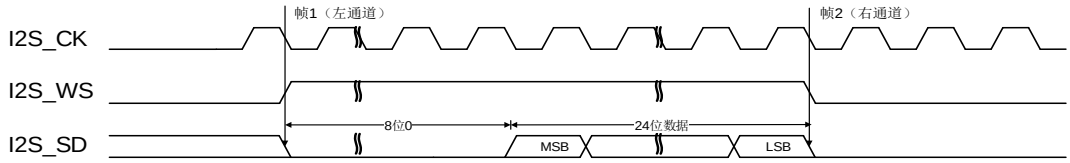
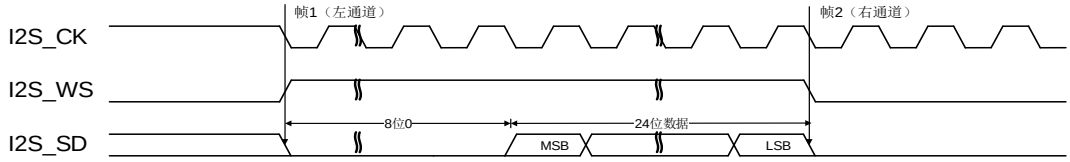


图 22-35. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）



当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI_DATA寄存器的数据应该是一个16位数据，该16位数据的高8位可以是任意值，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI_DATA寄存器读到的数据应该是一个16位数据，该16位数据的高8位是0，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。

图 22-36. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）

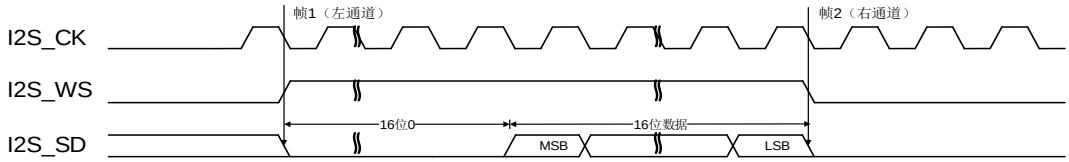
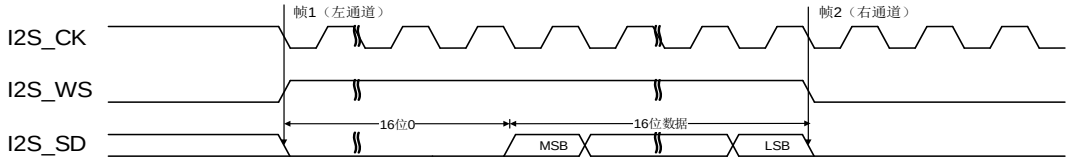


图 22-37. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）



当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问SPI_DATA寄存器一次。为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为0x0000。

PCM 标准

对于PCM标准，I2S_WS和I2S_SD在I2S_CK的上升沿变化，I2S_WS信号表示帧同步信息。可以通过SPI_I2SCTL寄存器的PCMSMOD位来选择短帧同步模式和长帧同步模式。SPI_DATA寄存器的处理方式与I2S飞利浦标准完全相同。短帧同步模式的各种配置情况时序图如下所示。

图 22-38. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）

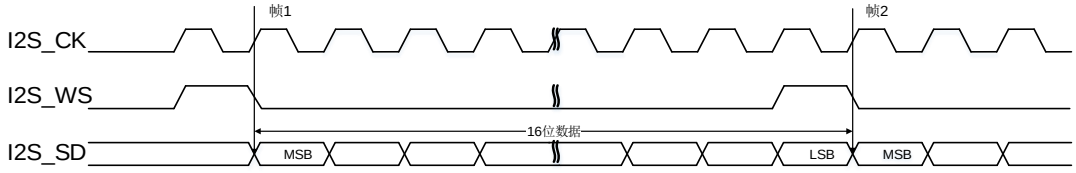


图 22-39. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）

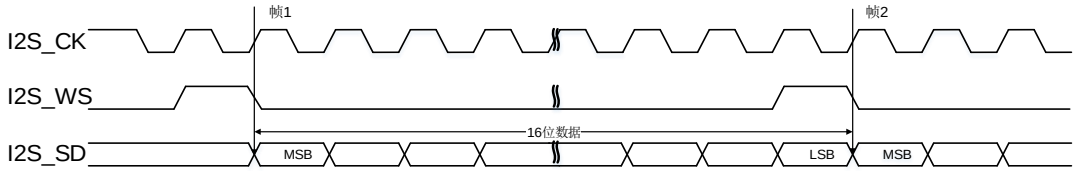


图 22-40. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）

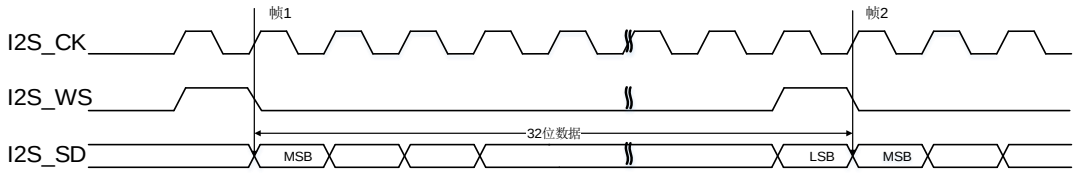


图 22-41. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）

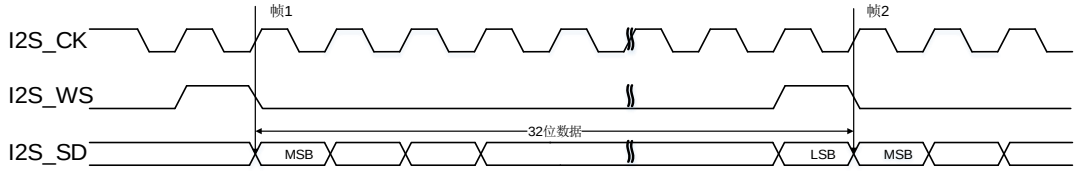


图 22-42. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）

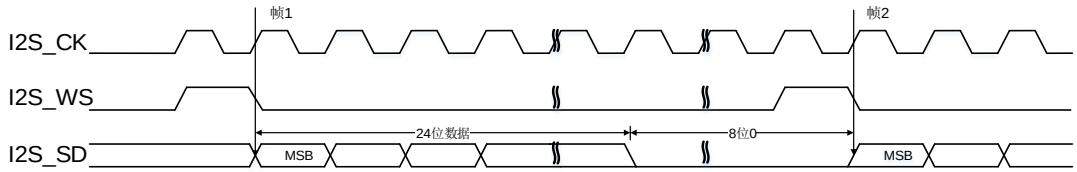


图 22-43. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）

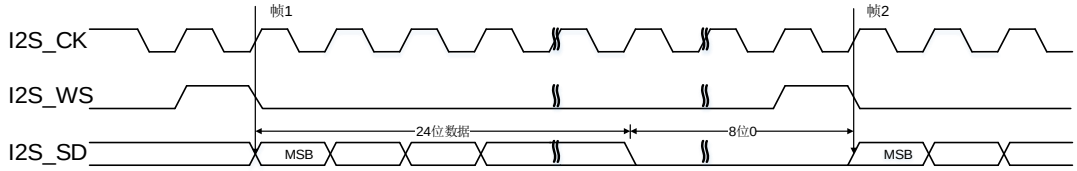


图 22-44. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）

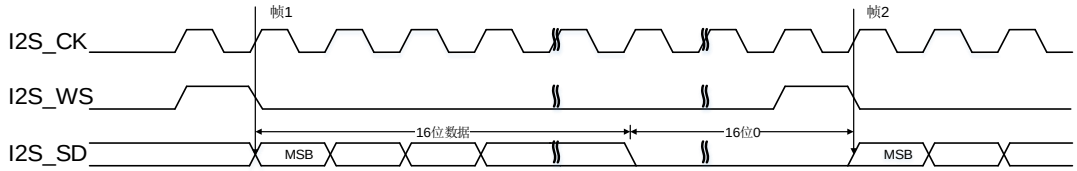
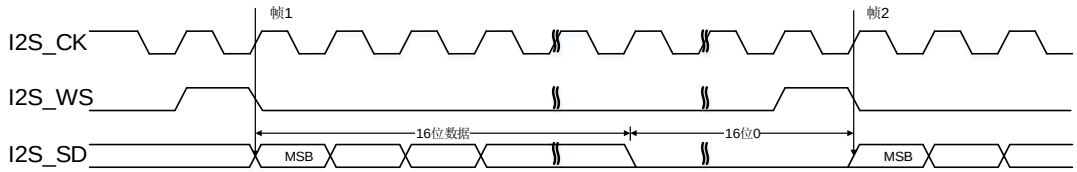


图 22-45. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）



长帧同步模式的各种配置情况时序图如下所示。

图 22-46. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）

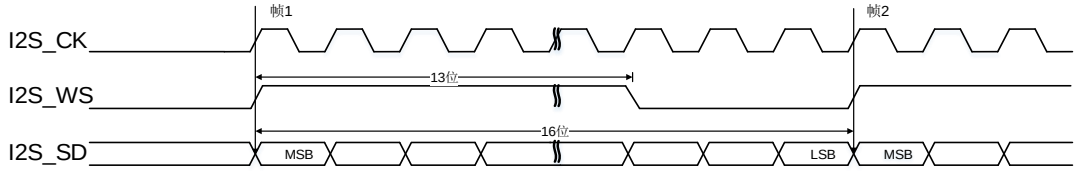


图 22-47. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）

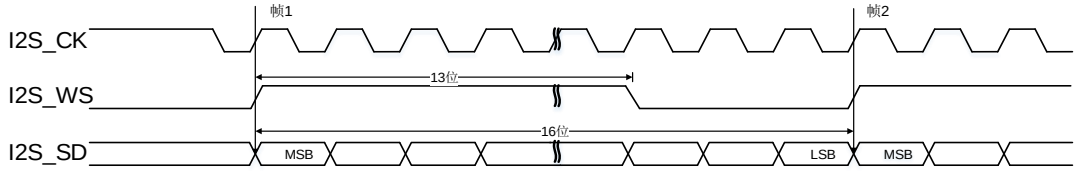


图 22-48. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）

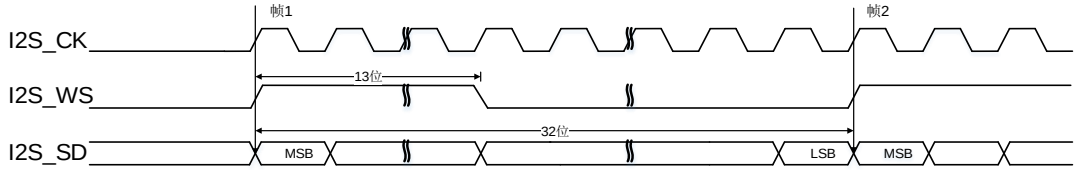


图 22-49. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）

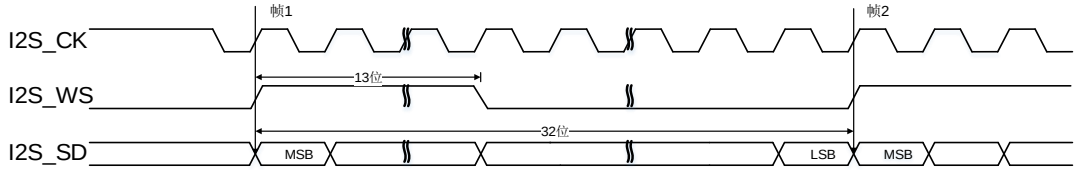


图 22-50. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）

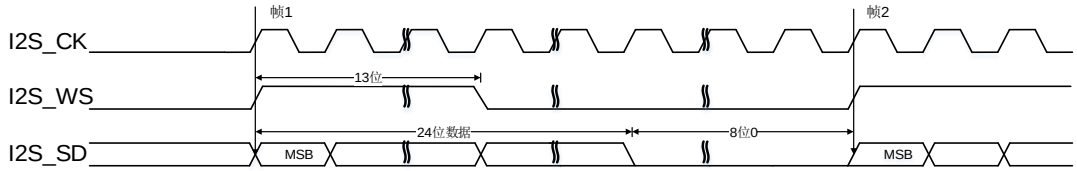


图 22-51. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）

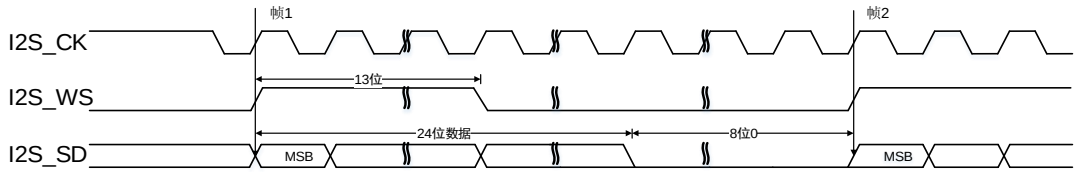


图 22-52. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）

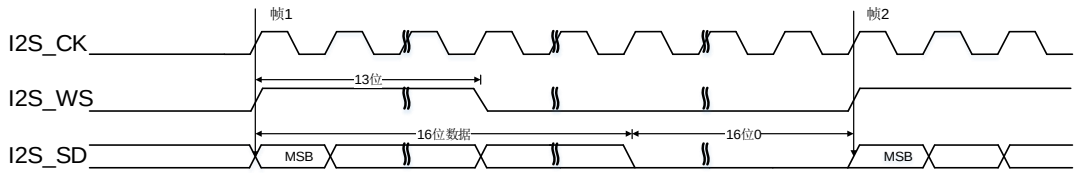
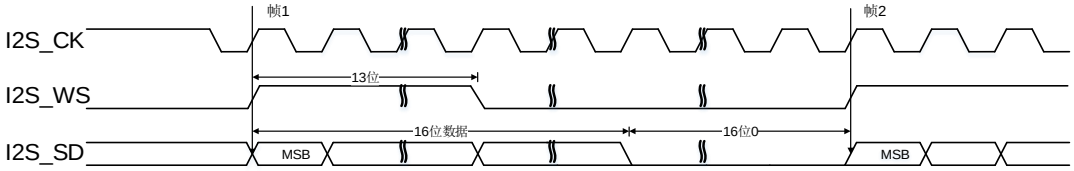
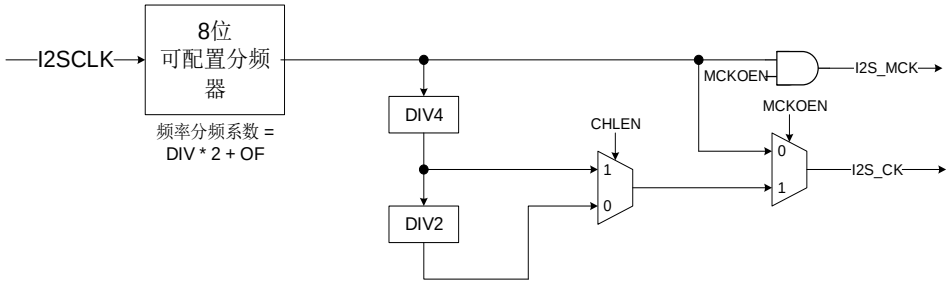


图 22-53. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）



22.4.4. I2S 时钟

图 22-54. I2S 时钟生成结构框图



I2S 时钟生成器框图如*图22-54. I2S 时钟生成结构框图*所示。I2S 接口时钟是通过 SPI_I2SPSC 寄存器的 DIV 位，OF 位和 MCKOEN 位以及 SPI_I2SCTL 寄存器的 CHLEN 位来配置的。时钟源是系统时钟（CK_SYS）。I2S 比特率可以通过*表 22-7. I2S 比特率计算公式*所示的公式计算。

表 22-7. I2S 比特率计算公式

MCKOEN	CHLEN	公式
0	0	I2SCLK / (DIV * 2 + OF)
0	1	I2SCLK / (DIV * 2 + OF)
1	0	I2SCLK / (8 * (DIV * 2 + OF))
1	1	I2SCLK / (4 * (DIV * 2 + OF))

音频采样率（Fs）和I2S比特率的关系由如下公式定义：

Fs = I2S比特率 / (通道长度 * 通道数)

所以，为了得到期望的音频采样率，时钟生成器需要按*表22-8. 音频采样频率计算公式*所列的公式进行配置。

表 22-8. 音频采样频率计算公式

MCKOEN	CHLEN	公式
0	0	I2SCLK / (32 * (DIV * 2 + OF))
0	1	I2SCLK / (64 * (DIV * 2 + OF))
1	0	I2SCLK / (256 * (DIV * 2 + OF))
1	1	I2SCLK / (256 * (DIV * 2 + OF))

22.4.5. 运行

运行模式

运行模式是通过 SPI_I2SCTL 寄存器的 I2SOPMOD 位来选择的。共有四种运行模式可供选择：主机发送模式，主机接收模式，从机发送模式和从机接收模式。各种运行模式下 I2S 接口信号的方向如*表 22-9. 各种运行模式下 I2S 接口信号的方向*所示。

表 22-9. 各种运行模式下 I2S 接口信号的方向

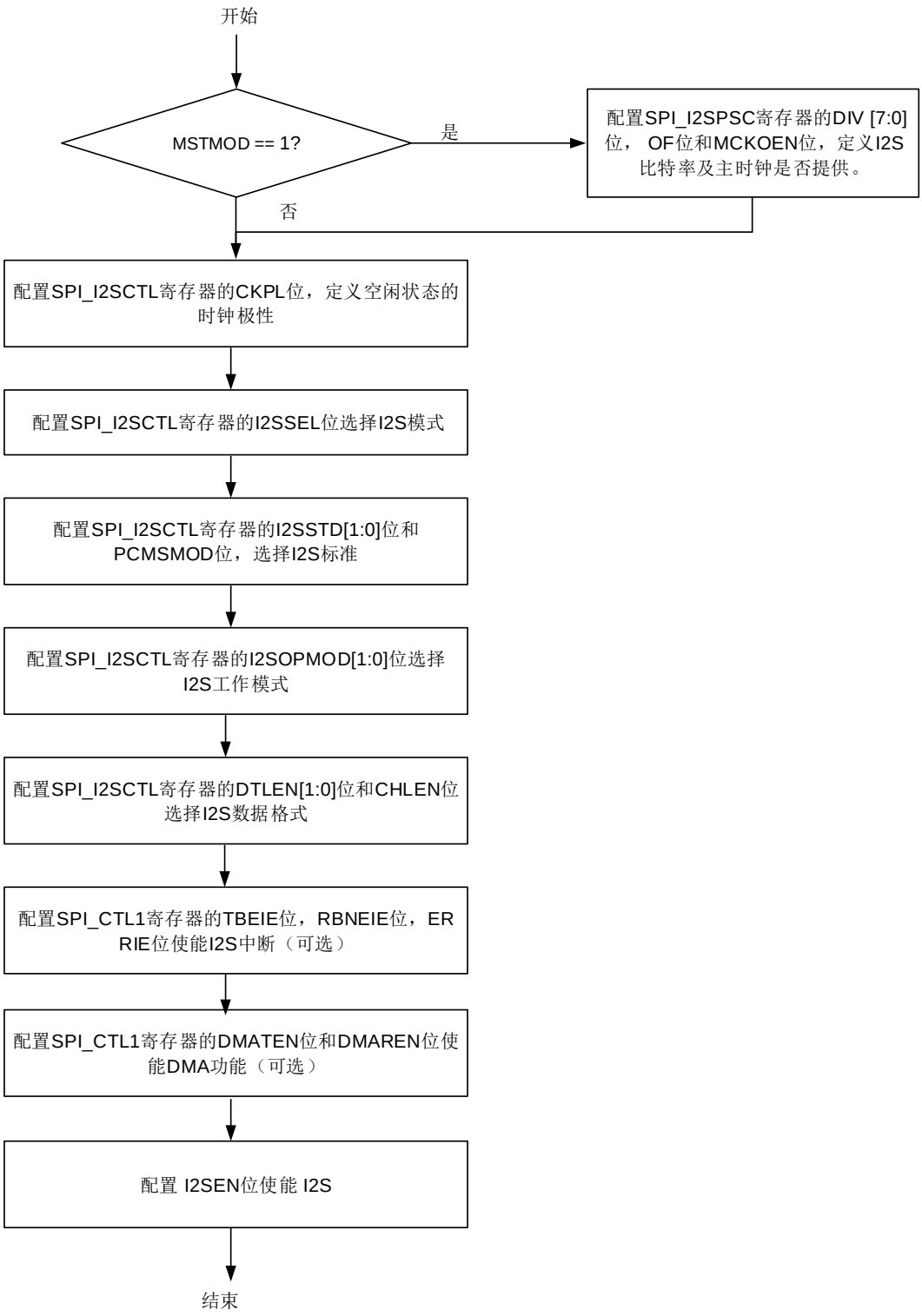
运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
主机发送	输出或 NU(1)	输出	输出	输出
主机接收	输出或 NU(1)	输出	输出	输入
从机发送	输出或 NU(1)	输入	输入	输出
从机接收	输出或 NU(1)	输入	输入	输入

1. NU表示该引脚没有被I2S使用，可以用于其他功能。

I2S 初始化流程

I2S初始化过程如*图22-55. I2S初始化流程*所示。

图 22-55. I2S 初始化流程



I2S 主机发送流程

TBE标志位被用来控制发送流程。如前文所述，TBE标志位表示发送缓冲区空，此时，如果SPI_CTL1寄存器的TBEIE位为1，将产生中断。首先，发送缓冲区为空（TBE为1），且移位寄

存器中没有发送序列。当16位数据被写入SPI_DATA寄存器时（TBE变为0），数据立即从发送缓冲区装载到移位寄存器中（TBE变为1）。此时，发送序列开始。

数据是并行地装载到16位移位寄存器中的，然后串行地从I2S_SD引脚发出（高位先发）。下一个数据应该在TBE为1时写入SPI_DATA寄存器。数据写入SPI_DATA寄存器之后，TBE变为0。当前发送序列结束时，发送缓冲区的数据会自动装载到移位寄存器中，然后TBE标志变回1。为了保证连续的音频数据发送，下一个将要发送的数据必须在当前发送序列结束之前写入SPI_DATA寄存器。

对于除PCM标准外的所有标准，I2SCH标志用来区别当前传输数据所属的通道。I2SCH标志在每次TBE标志由0变1的时候更新。刚开始I2SCH标志为0，表示左通道的数据应该被写入SPI_DATA寄存器。

为了关闭I2S，I2SEN位必须在TBE标志为1且TRANS标志为0之后清零。

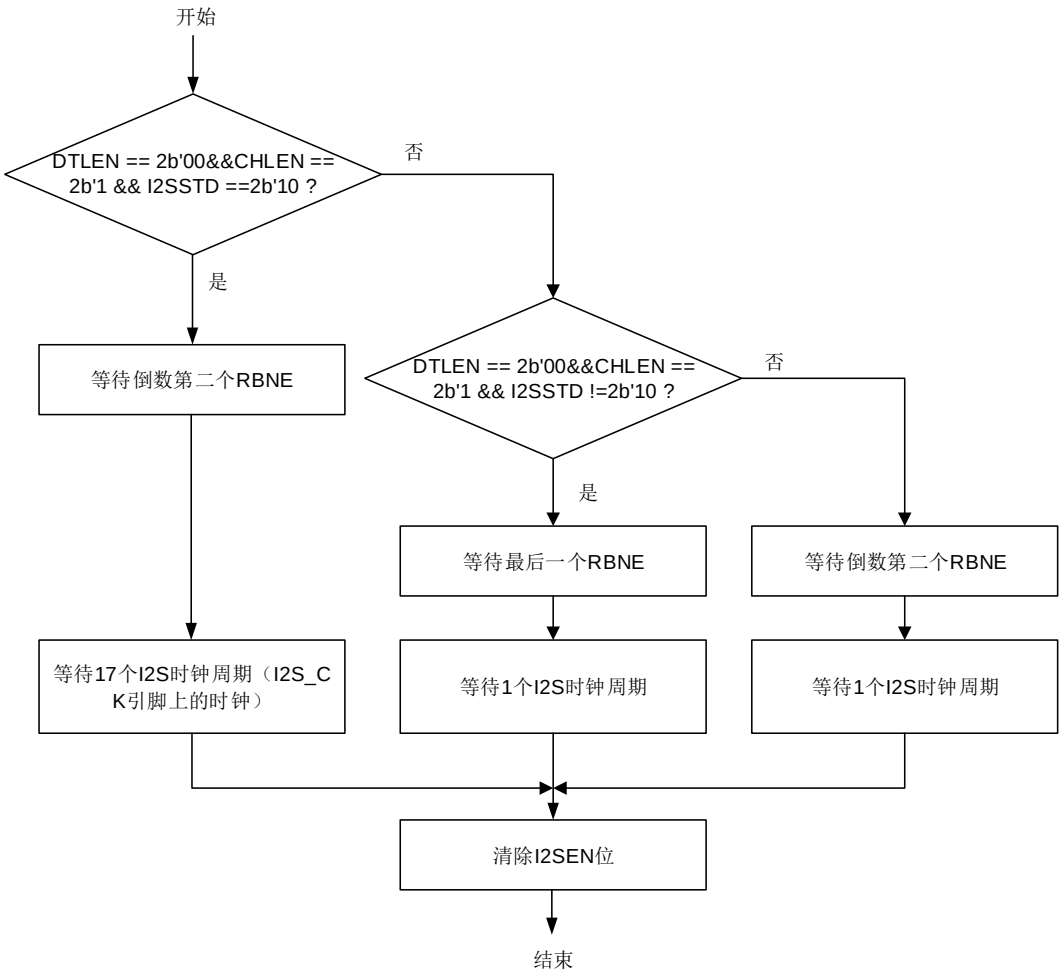
I2S 主机接收流程

RBNE标志被用来控制接收序列。如前文所述，RBNE标志表示接收缓冲区非空，如果SPI_CTL1寄存器的RBNEIE位为1，将产生中断。当SPI_I2SCTL寄存器的I2SEN位被置1时，接收流程立即开始。首先，接收缓冲区为空（RBNE为0）。当一个接收流程结束时，接收到的数据将从移位寄存器装载到接收缓冲区（RBNE变为1）。当RBNE为1时，用户应该将数据从SPI_DATA寄存器中读走。读操作完成后，RBNE变为0。必须在下一次接收结束之前读走SPI_DATA寄存器中的数据，否则将发生接收过载错误。此时RXORERR标志位会被置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S，然后再恢复通讯。

对于除PCM之外的所有标准来说，I2SCH标志用来区分当前传输数据所属的通道。I2SCH标志在每次RBNE标志由0变1时更新。

为了关闭I2S，不同的音频标准，数据长度和通道长度采用不同的操作步骤。每种情况的操作如*图22-56. I2S主机接收禁能流程*所示。

图 22-56. I2S 主机接收禁能流程



I2S 从机发送流程

从机发送流程和主机发送流程相似，不同之处如下：

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且I2S_WS信号请求传输数据时，发送流程开始。数据需要在外部主机发起通讯之前写入SPI_DATA寄存器。为了确保音频数据的连续传输，必须在当前发送序列结束之前将下一个待发送的数据写入SPI_DATA寄存器，否则会产生发送欠载错误。此时TXURERR标志会置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S来恢复通讯。从机模式下，I2SCH标志是根据外部主机发送的I2S_WS信号而变化的。

为关闭I2S，必须在TBE标志变为1且TRANS标志变为0之后，才能清除I2SEN位。

I2S 从机接收流程

从机接收流程与主机接收流程类似。不同之处如下。

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且I2S_WS信号指示数据开始时，接收流程开始。从机模式下，I2SCH标志是根据外部主机发送

的I2S_WS信号而变化的。

为了关闭I2S，必须在收到最后一个RBNE之后立即清除I2SEN位。

22.4.6. DMA 功能

DMA功能与SPI模式完全一样，唯一不同的地方就是I2S模式不支持CRC功能。

22.4.7. I2S 中断

状态标志位

SPI_STAT寄存器中有4个可用的标志位，分别是TBE、RBNE、TRANS和I2SCH，用户通过这些标志位可以全面监视I2S总线的状态。

- 发生缓冲区空标志（TBE）：
  当发送缓冲区为空时，TBE置位。软件可以通过写SPI_DATA寄存器将下一个数据写入发送缓冲区。
- 接收缓冲区非空标志（RBNE）：
  接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入接收缓冲区中，软件可以通过读SPI_DATA寄存器来读取此数据。
- I2S通信进行中标志（TRANS）：
  TRANS是用来指示当前传输是否正在进行或结束的状态标志，它由内部硬件置位和清除，无法进行软件操作。该标志位不会产生任何中断。
- I2S通道标志（I2SCH）：
  I2SCH用来表明当前传输数据的通道信息，对PCM音频标准来说没有意义。在发送模式下，I2SCH标志在每次TBE由0变1时更新，在接收模式下，I2SCH标志在每次RBNE由0变1时更新。该标志位不会产生任何中断。

错误标志

有三个错误标志：

- 发送欠载错误标志（TXURERR）：
  在从发送模式下，有效的SCK信号开始发送，当发送缓冲区为空时，发送欠载错误标志TXURERR置位。
- 接收过载错误标志（RXORERR）：
  当接收缓冲区已满且又接收到一个新的数据时，接收过载错误标志RXORERR置位。当接收过载发生时，接收缓冲区中的数据没有更新，新接收的数据丢失。
- 帧格式错误（FERR）：
  在从I2S模式下，I2S模块监视I2S_WS信号，如果I2S_WS信号在一个错误的位置发生翻转，将会置位FERR帧错误标志位。

*表22-10. I2S中断*总结了I2S中断事件和相应的使能位。

表 22-10. I2S 中断

中断标志	描述	清除方式	中断使能位
TBE	发送缓冲区空	写 SPI_DATA 寄存器	TBEIE
RBNE	接收缓冲区非空	读 SPI_DATA 寄存器	RBNEIE
TXURERR	发送欠载错误	读 SPI_STAT 寄存器	ERRIE
RXORERR	接收过载错误	读 SPI_DATA 寄存器，然后再读 SPI_STAT 寄存器	
FERR	I2S 帧错误	读 SPI_STAT 寄存器	

22.5. SPI/I2S 寄存器

SPI0基地址：0x4001 3000
SPI1/I2S1基地址：0x4000 3800

22.5.1. 控制寄存器 0（SPI_CTL0）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

该寄存器在I2S模式下没有意义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16 CRCL	RO	SWNSS EN	SWNSS	LF	SPIEN	PSC[2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	BDEN	双向数据模式使能 0：2线单向传输模式 1：1线双向传输模式。数据在主机的MOSI引脚和从机的MISO引脚之间传输。
14	BDOEN	双向传输输出使能 当BDEN置位时，该位决定了数据的传输方向。 0：工作在只接收模式 1：工作在只发送模式
13	CRCEN	CRC计算使能 0：CRC计算禁止 1：CRC计算使能
12	CRCNT	下一次传输CRC 0：下一次传输值为数据 1：下一次传输值为CRC值（TCRC） 当数据传输由DMA管理时，CRC值由硬件传输，该位应该被清零。 在全双工和只发送模式下，当最后一个数据写入SPI_DATA寄存器后应将该位置1。 在只接收模式下，在接收完倒数第二个数据后应将该位置1。
11	FF16	数据帧格式（只有SPI1） 0：8位数据帧格式

		1：16位数据帧格式
	CRCL	CRC长度（只有SPI0） 0：8位CRC长度 1：16位CRC长度
10	RO	只接收模式 当BDEN清零时，该位决定了数据的传输方向。 0：全双工模式 1：只接收模式
9	SWNSSEN	NSS软件模式使能 0：NSS硬件模式，NSS电平取决于NSS引脚 1：NSS软件模式，NSS电平取决于SWNSS位 该位在SPI TI模式下没有意义。
8	SWNSS	NSS软件模式下NSS引脚选择 0：NSS引脚拉低 1：NSS引脚拉高 只有在SWNSSEN置位时，该位有效。 该位在SPI TI模式下没有意义。
7	LF	最低有效位先发模式 0：先发送最高有效位 1：先发送最低有效位 该位在SPI TI模式下没有意义。
6	SPIEN	SPI使能 0：SPI设备禁止 1：SPI设备使能
5:3	PSC[2:0]	主时钟预分频选择 000：PCLK/2 001：PCLK/4 010：PCLK/8 011：PCLK/16 100：PCLK/32 101：PCLK/64 110：PCLK/128 111：PCLK/256 当使用SPI0时，PCLK=PCLK2。当使用SPI1时，PCLK=PCLK1。
2	MSTMOD	主从模式使能 0：从机模式 1：主机模式
1	CKPL	时钟极性选择 0：SPI为空闲状态时，CLK引脚拉低

		1：SPI为空闲状态时，CLK引脚拉高
0	CKPH	时钟相位选择 0：在第一个时钟跳变沿采集第一个数据 1：在第二个时钟跳变沿采集第一个数据

22.5.2. 控制寄存器 1（SPI_CTL1）

地址偏移：0x04

复位值：SPI0：0x0000 0700

SPI1：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TXDMA_ODD	RXDMA_ODD	BYTEN	DZ[3:0]				TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN
	rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	TXDMA_ODD	DMA发送通道奇数字节（只有SPI0） 在数据合并传输模式中，当通过DMA发送的数据总数为奇数时置位。仅在DMA功能开启且合并模式开启时（数据长度小于等于8位且对SPI_DATA写入访问是16位宽）有效。 必须在SPI禁止时写入。 0：通过DMA发送的数据总量为偶数个。 1：通过DMA发送的数据总量为奇数个。
13	RXDMA_ODD	DMA接收通道奇数字节（只有SPI0） 在数据合并传输模式中，当通过DMA接收的数据总数为奇数时置位。仅在DMA功能开启且合并模式开启时（数据长度小于等于8位且对SPI_DATA写入访问是16位宽）有效。 必须在SPI禁止时写入。 0：通过DMA接收的数据总量为偶数个。 1：通过DMA接收的数据总量为奇数个。
12	BYTEN	字节访问使能（只有SPI0） 该位用于指示对FIFO的访问宽度，并设置产生RBNE的RXFIFO的阈值。 0：半字访问，且当RXLVL>=2时，RBNE置位。 1：字节访问，且当RXLVL>=1时，RBNE置位。
11:8	DZ[3:0]	数据位宽（只有SPI0） 这些位配置SPI传输数据的位宽：

		0000：强制为“0111” 0001：强制为“0111” 0010：强制为“0111” 0011：4位 0100：5位 …… 1111：16位
7	TBEIE	发送缓冲区/发送FIFO空中断使能 0：TBE中断禁止 1：TBE中断使能。当TBE置位时，产生中断。
6	RBNEIE	接收缓冲区/接收FIFO非空中断使能 0：RBNE中断禁止 1：RBNE中断使能。当RBNE置位时，产生中断。
5	ERRIE	错误中断使能 0：错误中断禁止 1：错误中断使能。当CRCERR位，CONFERR位，RXORERR位或者TXURERR位置1时，产生中断。
4	TMOD	SPI TI模式使能 0：SPI TI模式禁止 1：SPI TI模式使能
3	NSSP	SPI NSS脉冲模式使能 0：SPI NSS脉冲模式禁止 1：SPI NSS脉冲模式使能
2	NSSDRV	NSS输出使能 0：NSS输出禁止 1：NSS输出使能。 当SPI使能时，如果NSS引脚配置为输出模式，NSS引脚在主模式时被拉低。如果NSS引脚配置为输入模式，NSS引脚在主模式时被拉高，此时该位无效。
1	DMATEN	发送缓冲区/发送FIFO DMA使能 0：发送缓冲区/发送FIFO DMA禁止 1：发送缓冲区/发送FIFO DMA使能。当SPI_STAT中的TBE置位时，将会在相应的DMA通道上产生一个DMA请求。
0	DMAREN	接收缓冲区/接收FIFO DMA使能 0：接收缓冲区/接收FIFO DMA禁止 1：接收缓冲区/接收FIFO DMA使能。当SPI_STAT中的RBNE置位时，将会在相应的DMA通道上产生一个DMA请求。

22.5.3. 状态寄存器（SPI_STAT）

地址偏移：0x08

复位值：0x0000 0002

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			TXLVL[1:0]		RXLVL[1:0]		FERR	TRANS	RXORERR	CONFERR	CRCERR	TXURERR	I2SCH	TBE	RBNE
			r		r		rc_w0	r	r	r	rc_w0	r	r	r	r

位/位域	名称	描述
31:13	保留	必须保持复位值。
12:11	TXLVL[1:0]	发送FIFO状态（只有SPI0） 00：空 01：1/4满 10：1/2满 11：满 **注意：**这里的FIFO状态是指FIFO当前实际的存储量。在这里，当FIFO存储量大于总存储量的1/2时认为FIFO已满。
10:9	RXLVL[1:0]	接收FIFO状态（只有SPI0） 00：空 01：1/4满 10：1/2满 11：满 这些位在打开了CRC计算功能时的SPI只接收模式下，不使用。 **注意：**这里的FIFO状态是指FIFO当前实际的存储量。在这里，当FIFO存储量大于总存储量的1/2时认为FIFO已满。
8	FERR	帧错误 SPI TI模式： SPI TI模式： 0：没有TI模式帧错误发生 1：TI模式帧错误发生 I2S模式： 0：没有I2S帧错误发生 1：I2S帧错误发生
7	TRANS	通信进行中标志 0：SPI空闲 1：SPI当前正在发送且/或接收数据 该位由硬件置位和清除。
6	RXORERR	接收过载错误标志 0：没有接收过载错误发生

		1：接收过载错误发生 该位由硬件置位，软件序列清零。软件序列为：先读SPI_DATA寄存器，然后读SPI_STAT寄存器。
5	CONFERR	SPI配置错误 0：无配置错误发生 1：配置错误发生（主机模式下，在硬件NSS模式时NSS引脚被拉低，或者软件NSS模式时SWNSS位为0，都会产生CONFERR错误） 该位由硬件置位，软件序列清零。软件序列为：读或写SPI_STAT寄存器，然后写SPI_CTL0寄存器。
4	CRCERR	SPI CRC错误标志 0：SPI_RCRC值等于最后接收到的CRC值 1：SPI_RCRC值不等于最后接收到的CRC值该位由硬件置位，可以通过写0清除。
3	TXURERR	发送欠载错误标志 0：无发送欠载错误发生 1：发送欠载错误发生 该位由硬件置位，通过读SPI_STAT寄存器清除。 SPI模式下不使用该位。
2	I2SCH	I2S通道标志 0：下一个将要发送或刚刚接收到的数据属于左通道 1：下一个将要发送或刚刚接收到的数据属于右通道 该位由硬件置位和清除。 SPI模式下该位无用，I2S PCM模式下该位没有意义。
1	TBE	发送缓冲区/发送FIFO空 0：发送缓冲区/发送FIFO非空 1：发送缓冲区/发送FIFO空
0	RBNE	接收缓冲区/接收FIFO非空 0：接收缓冲区/接收FIFO空 1：接收缓冲区/接收FIFO非空

22.5.4. 数据寄存器（SPI_DATA）

地址偏移：0x0C
复位值：0x0000 0000

对于SPI0，该寄存器可以按字节（8位）或半字（16位）访问。对于SPI1，该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI_DATA[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SPI_DATA[15:0]	数据传输寄存器值 对于SPI0，硬件有两个FIFO：TXFIFO和RXFIFO。向SPI_DATA写数据将会把数据存入发送FIFO，从SPI_DATA读数据，将从接收FIFO获得数据。 对于SPI1，硬件有两个缓冲区：发送缓冲区和接收缓冲区。向SPI_DATA写数据将会把数据存入发送缓冲区，从SPI_DATA读数据，将从接收缓冲区获得数据。当数据帧格式为8位时，SPI_DATA[15:8]强制为0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是8位。如果数据帧格式为16位，SPI_DATA[15:0]用于发送和接收数据，发送和接收缓冲区也是16位。 **注意：**对于SPI0，实际上硬件只根据配置好的BYTEN这一位来判断每一次访问SPI_DATA的位宽，与软件当前操作所使用的位宽无关。

22.5.5. CRC 多项式寄存器（SPI_CRCPOLY）

地址偏移：0x10
复位值：0x0000 0007

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CRCPOLY[15:0]	CRC多项式寄存器值 该值包含了CRC多项式，用于CRC计算，默认值为0007h。

22.5.6. 接收 CRC 寄存器（SPI_RCRC）

地址偏移：0x14
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RCRC[15:0]
r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RCRC[15:0]	接收CRC寄存器值 当SPI_CTL0中的CRCEN置位时，硬件计算接收数据的CRC值，并保存到RCRC寄存器中。对于SPI1，如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到RCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到RCRC[15:0]。对于SPI0，只有当数据长度为8位或16位时，CRC有效。当CRC长度设置为8位并且数据长度等于8位时，CRC计算基于CRC8标准进行，并将值保存在RCRC[7：0]中，否则CRC计算基于CRC16标准进行，并将值保存在RCRC[15：0]中。硬件在接收到每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。 当SPI_CTL0寄存器中的CRCEN位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。

22.5.7. 发送 CRC 寄存器（SPI_TCRC）

地址偏移：0x18
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRC[15:0]															
r															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TCRC[15:0]	发送CRC寄存器值 当SPI_CTL0中的CRCEN置位时，硬件计算发送数据的CRC值，并保存到TCRC寄存器中。对于SPI1，如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到TCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到TCRC[15:0]。对于SPI0，只有当数据长度为8位或16位时，CRC有效。当CRC长度设置为8位并且数据长度等于8位时，CRC计算基于CRC8标准进行，并将值保存在TCRC[7：0]中，否则CRC计算基于CRC16标准进行，并将值保存在TCRC[15：0]中。硬件在发送出每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。不同的数据帧格式（SPI_CTL0中的LF位决定）将会得到不同的CRC值。 当SPI_CTL0寄存器中的CRCEN位或RCU复位寄存器中的SPIxRST位置位时，该寄

存器复位。

22.5.8. I2S 控制寄存器（SPI_I2SCTL）

地址偏移：0x1C
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				I2SSEL	I2SEN	I2SOPMOD[1:0]		PCMSMOD	保留	I2SSTD[1:0]		CKPL	DTLEN[1:0]		CHLEN
				rw	rw	rw		rw		rw		rw	rw		rw

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	I2SSEL	I2S模式选择 0：SPI模式 1：I2S模式 当SPI或I2S关闭时配置该位。
10	I2SEN	I2S使能 0：I2S禁止 1：I2S使能 SPI模式不使用该位。
9:8	I2SOPMOD[1:0]	I2S运行模式 00：从机发送模式 01：从机接收模式 10：主机发送模式 11：主机接收模式 当I2S关闭时配置该位。SPI模式不使用该位。
7	PCMSMOD	PCM帧同步模式 0：短帧同步 1：长帧同步 只有在PCM标准下，该位才有意义。 当I2S关闭时配置该位。SPI模式不使用该位。
6	保留	必须保持复位值。
5:4	I2SSTD[1:0]	I2S标准选择 00：I2S飞利浦标准 01：MSB对齐标准

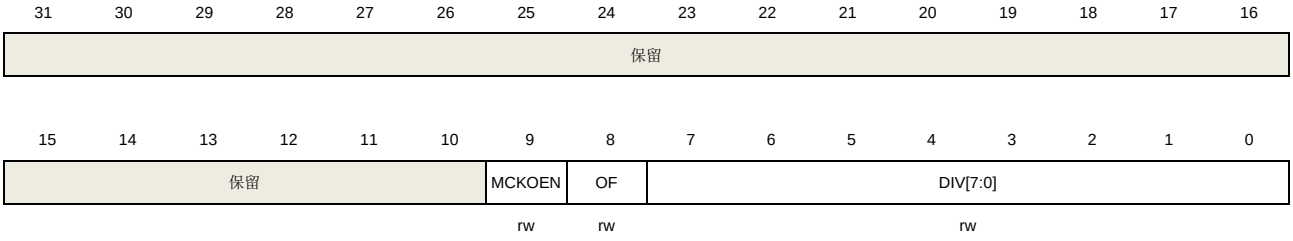
		10：LSB对齐标准 11：PCM标准 当I2S关闭时配置该位。SPI模式不使用该位。
3	CKPL	空闲状态时钟极性 0：I2S_CK空闲状态为低电平 1：I2S_CK空闲状态为高电平 当I2S关闭时配置该位。SPI模式不使用该位。
2:1	DTLEN[1:0]	数据长度 00：16位 01：24位 10：32位 11：保留 当I2S关闭时配置该位。SPI模式不使用该位。
0	CHLEN	通道长度 0：16位 1：32位 通道长度必须大于或等于数据长度。 当I2S关闭时配置该位。SPI模式不使用该位。

22.5.9. I2S 时钟预分频寄存器（SPI_I2SPSC）

地址偏移：0x20

复位值：0x0000 0002

该寄存器可以按半字（16位）或字（32位）访问。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9	MCKOEN	I2S_MCK输出使能 0：I2S_MCK输出禁止 1：I2S_MCK输出使能 当I2S关闭时配置该位。 SPI模式不使用该位。
8	OF	预分频器的奇系数 0：实际分频系数为DIV * 2 1：实际分频系数为DIV * 2 + 1

		当I2S关闭时配置该位。SPI模式下不使用该位。
7:0	DIV[7:0]	预分频器的分频系数 实际分频系数是DIV * 2 + OF。 DIV不能为0。 当I2S关闭时配置该位。SPI模式下不使用该位。

22.5.10. SPI0 四线 SPI 控制寄存器（SPI_QCTL）

地址偏移：0x80

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	IO23_DRV	IO2和IO3输出使能 0：单线模式下IO2和IO3输出关闭 1：单线模式下IO2和IO3输出高电平 该位仅适用于SPI0。
1	QRD	四线SPI模式读选择 0：SPI四线模式写操作 1：SPI四线模式读操作 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI0。
0	QMOD	四线SPI模式使能 0：SPI工作在单线模式 1：SPI工作在四线模式 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI0。

23. 加密处理器（CAU）

23.1. 简介

加密处理单元支持处理DES，三重DES或AES（128，192或256）算法，对数据进行加密或解密。加密处理器完全兼容下列标准：

- 联邦信息处理标准出版物“FIPS PUB 46-3，1999年10月25日”规定的数据加密标准（DES）和三重DES（TDES）。它遵循美国国家标准协会（ANSI）X9.52标准；
- 联邦信息处理标准出版物（FIPS PUB 197，2001年11月26日）规定的高级加密标准（AES）。

CAU处理器可在多种模式下使用DES/三重DES/多种长度密钥的AES算法执行数据加密和解密。

CAU外设为32位AHB外设，它支持对输入FIFO和输出FIFO的DMA传输。

23.2. 主要特征

- 支持DES，三重DES和AES加密解密算法
- 支持DES，三重DES和AES下的多种模式，包括电子密码本（ECB）、加密分组链接（CBC）模式、计数器模式（CTR）、伽罗瓦/计数器模式（GCM）、伽罗瓦消息验证码模式（GMAC）、加密分组链接-消息验证码模式（CCM）、密码反馈模式（CFB）和输出反馈模式（OFB）
- 输入与输出FIFO支持DMA传输

DES/三重DES

- 支持电子密码本（ECB）或加密分组链接（CBC）模式
- 支持在CBC模式下使用2×32位初始化向量（IV）
- 输入FIFO和输出FIFO可存储8x32位数据
- 对于输入/输出FIFO的数据支持半字、字节、位交换或不交换
- 数据可通过DMA或CPU中断进行传输，也可以不通过两者进行传输

AES

- 支持支持电子密码本（ECB）、加密分组链接（CBC）模式、计数器模式（CTR）、伽罗瓦/计数器模式（GCM）、伽罗瓦消息验证码模式（GMAC）、加密分组链接-消息验证码模式（CCM）、密码反馈模式（CFB）和输出反馈模式（OFB）
- 支持128位、192位或256位密钥
- 支持在CBC、CTR、GCM、GMAC、CCM、CFB和OFB模式下使用4×32位初始化向量(IV)
- 输入和输出FIFO各8字深
- 对于输入/输出FIFO的数据支持半字、字节、位交换或不交换
- 数据可通过DMA或CPU中断进行传输，也可以不通过两者进行传输

23.3. CAU 数据类型和初始化向量

23.3.1. 数据类型

CAU处理器一次输入32位（字）数据，DES每64位对数据流进行处理，AES每128位对数据流进行处理。对于每个数据块，在其进入CAU处理器之前，可对这些数据执行位、字节、半字交换或不交换操作（取决于要加密的数据类型）。在CAU数据写入OUT FIFO之前，需要对其执行同样的交换操作。注意由于系统存储器结构采用小端模式，无论使用何种数据类型，最低有效数据均占用最低地址位置。

*图23-1. DATAM不交换/半字交换*和*图23-2. DATATM字节交换/位交换*介绍了128位AES块在不同数据类型下的数据交换。（对于DES，数据块大小为2个32位字，请参考图中前两个字的数据交换）

图 23-1. DATAM 不交换/半字交换

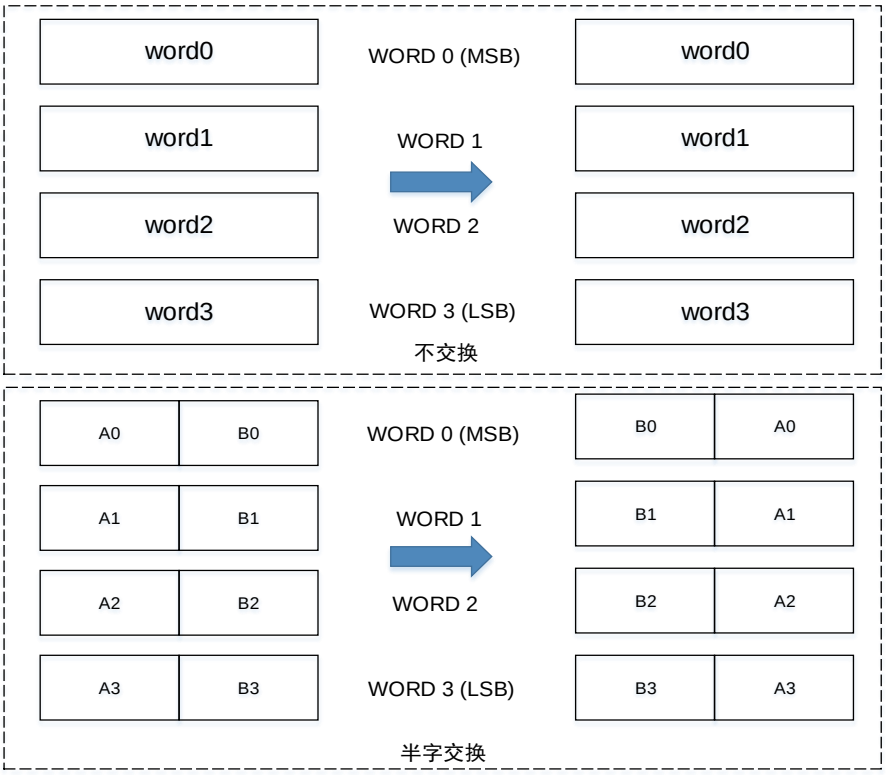
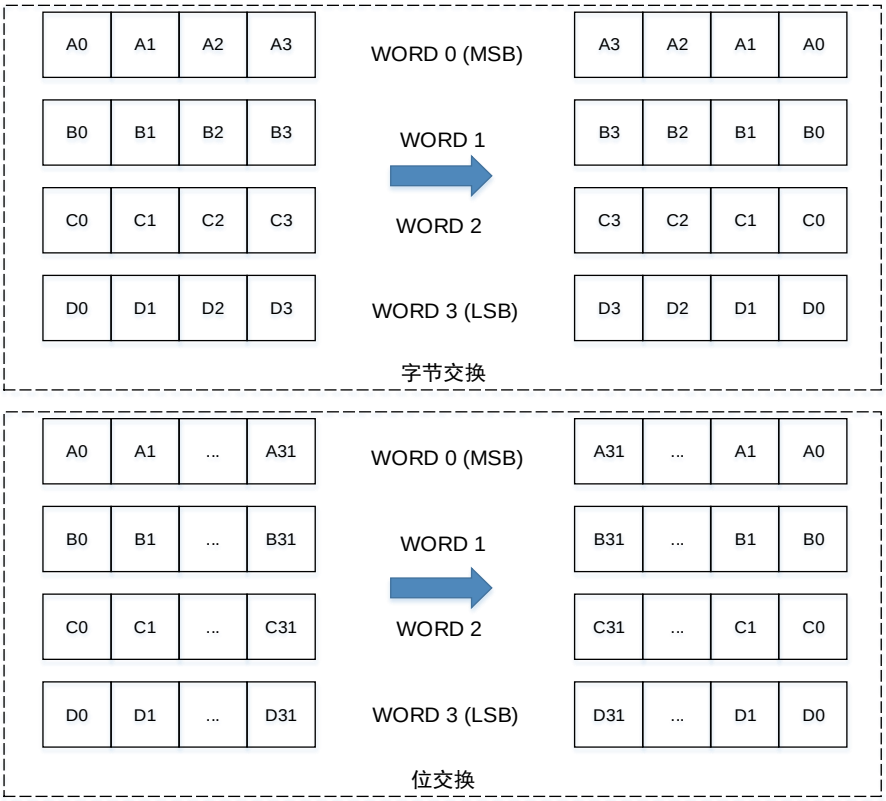


图 23-2. DATATM 字节交换/位交换



23.3.2. 初始化向量

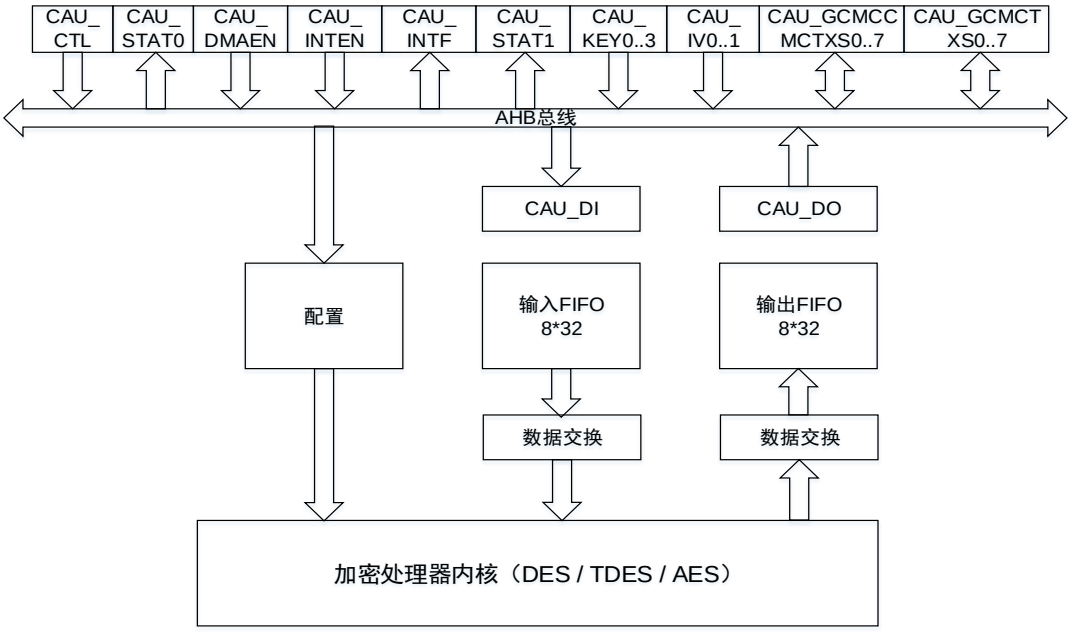
初始化向量用于在CBC、CTR、GCM、GMAC、CCM、CFB和OFB模式下与数据块进行异或。初始化向量与明文或密码数据无关，而且它们不受DATAM值的影响。注意初始化向量寄存器CAU_IV0..1（H/L）只有在BUSY位（CAU_STAT0寄存器位4）为0时才能被修改，否则写操作都是无效的。

23.4. 加密处理器流程

加密处理器关于DES和AES加密处理的实现具体请参考章节***DES / TDES加密处理流程***和***AES加密处理流程***。

图23-3. CAU框图为加密处理器的模块框图。

图 23-3. CAU 框图



23.4.1. DES / TDES 加密处理流程

DES/三重DES加密处理器由DES算法（DEA），密钥（DES算法使用1个密钥，TDES算法使用3个密钥），以及在CBC模式下使用的初始化向量组成。

DES / TDES 密钥

DES模式密钥为[KEY1]，TDES模式密钥为[KEY3 KEY2 KEY1]。当配置使用TDES算法，支持以下三种密钥选项：

1. 三个相同密钥

三个密钥KEY3、KEY2和KEY1是相同的，即KEY3=KEY2=KEY1。该选项详见FIPS PUB 46-3-1999（以及ANSI X9.52-1998）。这种模式下实际上与DES是等同的。

2. 两个独立密钥

这个选项中，KEY2与KEY1不同，KEY3与KEY1相同，即KEY1与KEY2独立，而KEY3=KEY1。该选项详见FIPS PUB 46-3-1999（以及ANSI X9.52-1998）。

3. 三个独立密钥

这个选项中，KEY1，KEY2与KEY3都是独立的。详见FIPS PUB 46-3 -1999（以及ANSI X9.52-1998）。

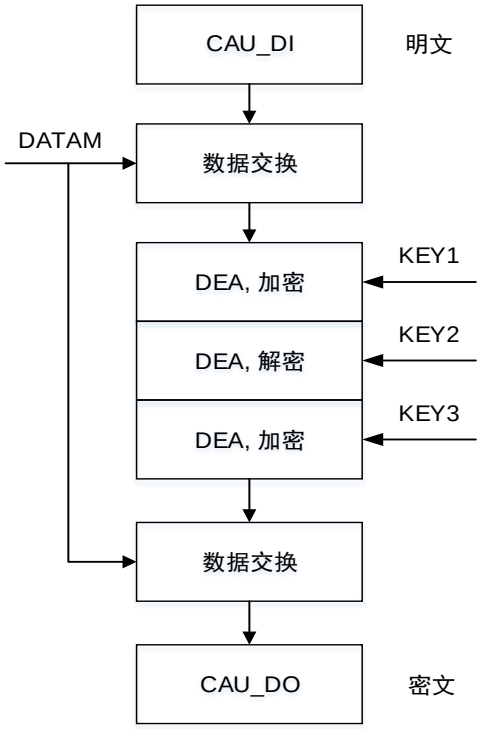
FIPS PUB 46-3（以及ANSI X9.52-1998）对DES/TDES中密钥的使用进行了详尽的解释，在本手册中不进行赘述。

DES/TDES 电子密码本（ECB）加密

64位输入明文数据首先经过根据数据类型值进行数据交换后作为输入数据块。若配置使用的是TDES算法，则输入数据块通过DEA使用KEY1进行加密处理。处理结果输出直接反馈到DEA，使用KEY2进行解密处理。之后处理结果输出直接反馈到到最后的DEA，使用KEY3进行加密处

理。上述的处理过程的输出需要再次根据数据类型值进行数据交换，生成一个64位密文输出数据块。若配置使用的是DES算法，在通过DEA使用KEY1进行加密处理后的结果直接根据数据类型值进行数据交换，生成一个64位密文输出数据块。DES/TDES电子密码本加密流程图见*图23-4. DES/TDES ECB加密*。

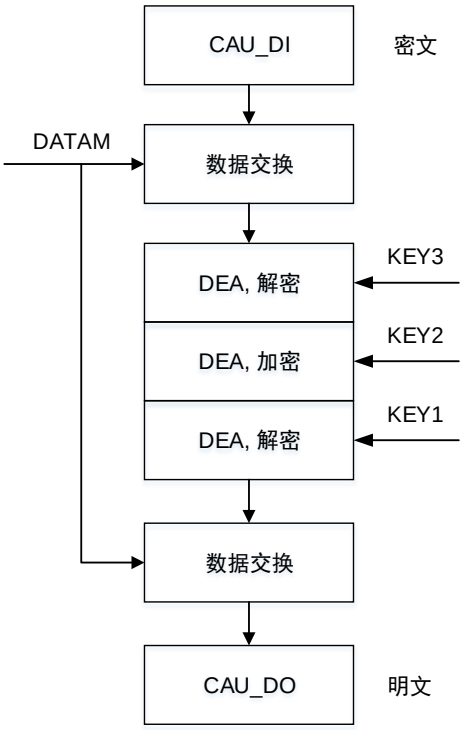
图 23-4. DES/TDES ECB 加密



DES/TDES 电子密码本（ECB）解密

根据数据类型进行数据交换后，首先得到64位的输入密文。若配置使用的是TDES算法，将在DEA中读取输入数据块并使用KEY3进行解密处理。处理结果输出直接反馈到下一个DEA，使用KEY2进行加密处理。之后处理结果输出直接反馈到到最后的DEA，使用KEY1进行解密处理。上述的处理过程的输出需要再次根据数据类型进行数据交换，生成一个64位明文输出数据块。若配置使用的是DES算法，在通过DEA使用KEY1进行解密处理后的结果直接根据数据类型值进行数据交换，生成一个64位明文输出数据块。DES/TDES电子密码本解密流程图见*图23-5. DES/TDES ECB解密*。

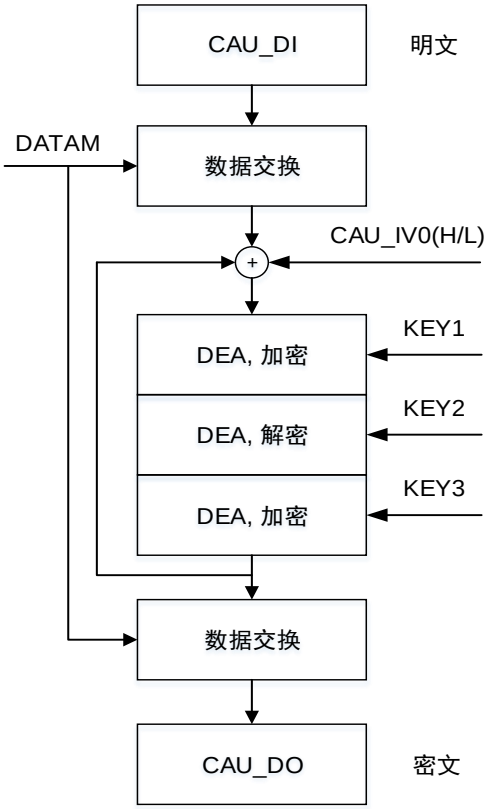
图 23-5. DES/TDES ECB 解密



DES/TDES 加密分组链接（CBC）加密

CBC模式下DEA块的输入包括两部分：根据数据类型进行数据交换后的输入明文数据块，以及初始化向量。若配置使用的是TDES算法，第一个数据交换后的输入明文数据块与64位初始化向量CAU_IV0..1进行异或运算，结果在DEA中读取并使用KEY1进行加密处理。处理结果输出直接反馈到下一个DEA，使用KEY2进行解密处理。之后处理结果输出直接反馈到到最后的DEA，使用KEY3进行加密处理。上述的处理过程的输出作为下一个初始化向量，并与下一个明文数据块进行异或运算，进行下一轮的加密处理。重复上述的操作，直到完成最后一个明文数据块的加密处理。注意如果明文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行加密处理。最后，对上述处理结果的输出需要再次根据数据类型进行数据交换，生成密文输出数据块。若配置使用的是DES算法，则在上述的步骤操作中忽略第二次和第三次DEA的运算处理。DES/TDES加密分组链接加密流程图见*图23-6. DES/TDES CBC加密*。

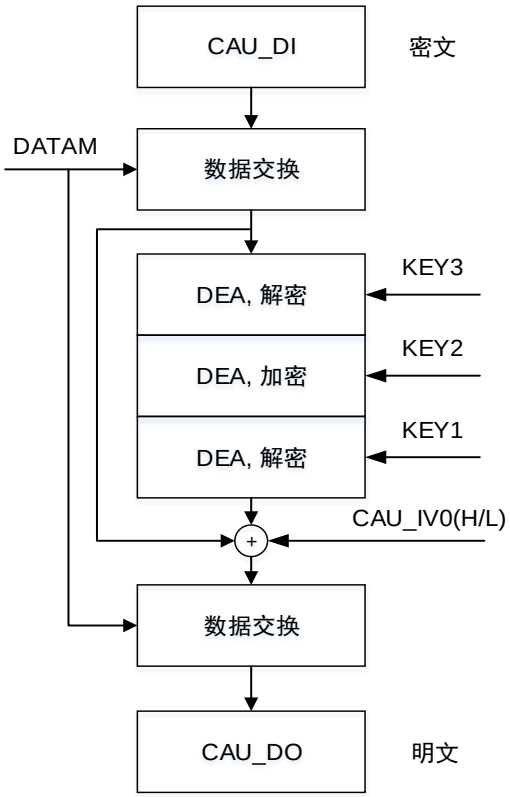
图 23-6. DES/TDES CBC 加密



DES/TDES 密码块链接（CBC）解密

使用DES/TDES CBC模式解密，若配置使用的是TDES算法，第一个数据交换后的输入密文数据块，通过DEA读取并使用KEY3进行解密处理。处理结果输出直接反馈到下一个DEA，使用KEY2进行加密处理。之后处理结果输出直接反馈到到最后的DEA，使用KEY1进行解密处理。上述的处理过程的输出再与64位初始化向量CAU_IV0..1进行异或运算。之后，第一个输入密文数据块作为下一个初始化向量，并与后续的DEA解密处理后的输出结果进行异或运算。重复上述的操作，直到完成最后一个密文数据块的解密处理。注意如果密文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行解密处理。最后，对上述处理结果的输出需要再次根据数据类型进行数据交换，生成明文输出数据块。若配置使用的是DES算法，则在上述的步骤操作中忽略第二次和第三次DEA的运算处理。DES/TDES加密分组链接解密流程图见 *图23-7. DES/TDES CBC解密*。

图 23-7. DES/TDES CBC 解密



23.4.2. AES 加密处理流程

AES加密处理器由AES算法（AEA），多个密钥，以及初始化向量或随机数三部分组成。

AES支持三种长度的密钥：128、192和256位密钥，根据操作模式的不同使用不同数目的初始化向量或随机数。

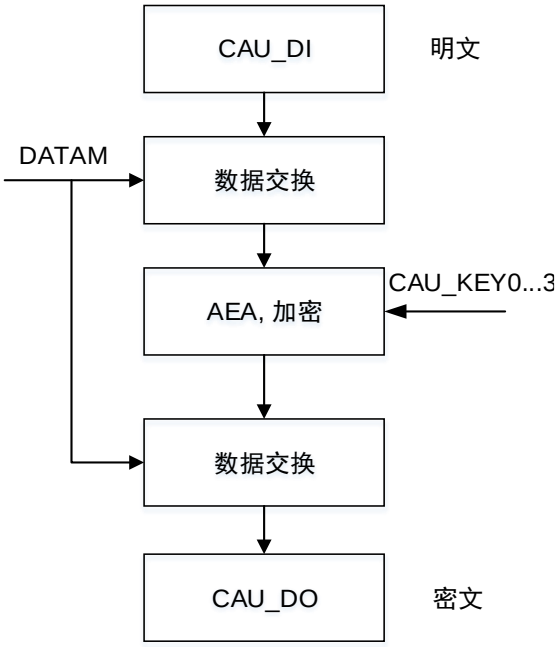
使用128位密钥时AES密钥为[KEY3 KEY2]，使用192位密钥时AES密钥为[KEY3 KEY2 KEY1]，使用256位密钥时AES密钥为[KEY3 KEY2 KEY1 KEY0]。

FIPS PUB 197（2001年11月26日）中对AES中使用的密钥进行了详细的解释，本手册不再进行赘述。

AES 电子密码本（ECB）加密

根据数据类型进行数据交换后，首先得到128位输入明文数据块。输入数据块通过AEA使用128位，或192位，或256位密钥进行加密处理。处理结果再根据数据类型进行数据交换，生成一个128位密文输出数据块，并存储在输出FIFO中。AES电子密码本加密流程图见*图23-8. AES ECB加密*。

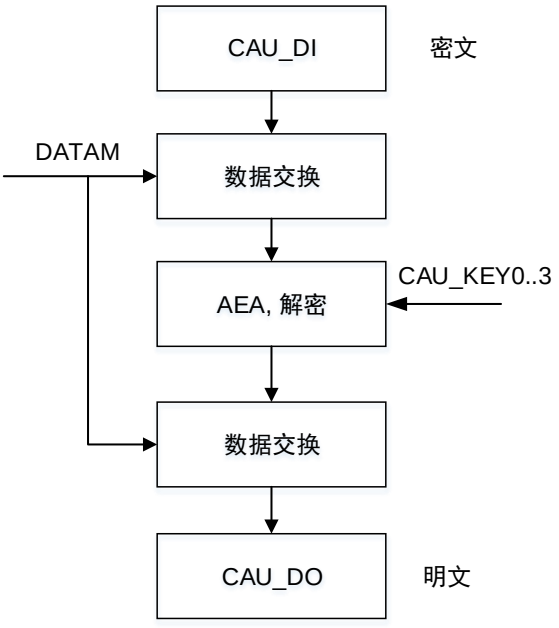
图 23-8. AES ECB 加密



AES 电子密码本（ECB）解密

首先需要准备密钥，以用于解密，密钥准备过程的输入密钥与加密处理中的密钥相同。从上述操作中获得的最后一个密钥将作为解密处理用的第一个密钥。密钥准备完成后，首先根据数据类型进行数据交换得到128位输入密文数据块。输入数据块在AEA中读取并使用上面准备的密钥进行解密处理。处理结果输出再根据数据类型值进行数据交换，生成一个128位明文输出数据块。AES电子密码本解密流程图见*图23-9. AES ECB解密*。

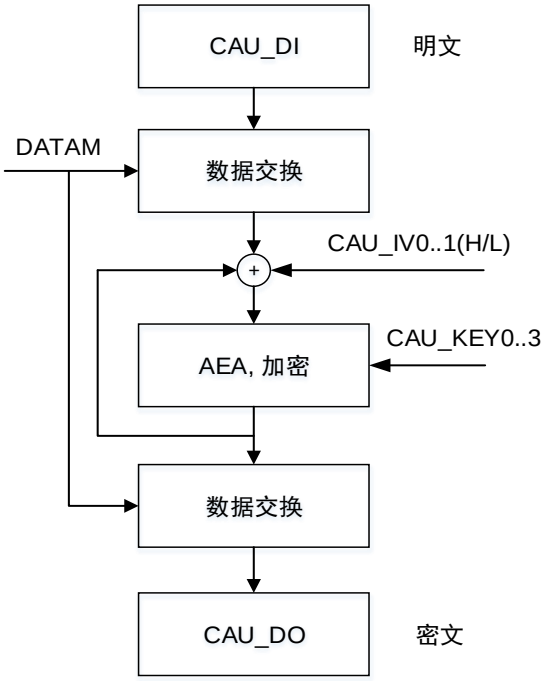
图 23-9. AES ECB 解密



AES 加密分组链接（CBC）加密

CBC模式下AEA块的输入包括两部分：根据数据类型进行数据交换后的输入明文数据块，以及初始化向量。数据交换后的输入明文数据块与128位初始化向量CAU_IV0..1进行异或运算，结果再通过AEA使用128位，或192位，或256位密钥进行加密处理。处理结果作为下一个初始化向量，并与下一个输入明文数据块进行异或运算，进行下一轮加密处理。重复上述的操作，直到完成最后一个明文数据块的加密处理。注意如果明文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行加密处理。最后，对上述处理结果的输出需要再次根据数据类型值进行数据交换，生成密文输出数据块。AES加密分组链接加密流程图见*图23-10. AES CBC加密*。

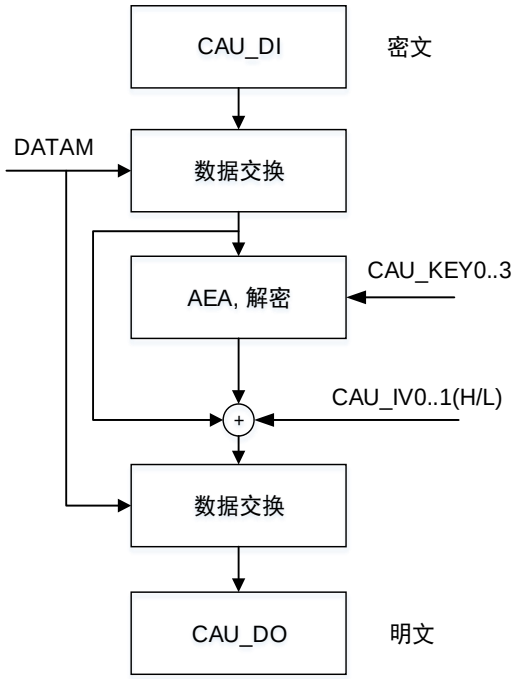
图 23-10. AES CBC 加密



AES 加密分组链接（CBC）解密

与AES电子密码本（ECB）模式解密类似，首先需要准备密钥以用于解密，密钥准备过程的输入密钥与加密处理中的密钥相同。从上述操作中获得的最后一个密钥将作为解密处理用的第一个密钥。密钥准备完成后，首先根据数据类型进行数据交换，得到128位输入密文数据块，输入数据块在AEA中读取并使用准备的密钥进行解密处理。之后，第一个输入密文数据块作为下一个初始化向量，并与下一个AEA解密处理结果进行异或运算（第一次的初始化向量为输入CAU_IV0..1的初值）。重复上述的操作，直到完成最后一个密文数据块的解密处理。注意如果密文消息中的数据块数不是整数，则应按指定的方式对最后的不完整数据块进行解密处理。最后，对上述处理结果的输出需要再次根据数据类型进行数据交换，生成明文输出数据块。AES加密分组链接解密流程图见*图23-11. AES CBC解密*。

图 23-11. AES CBC 解密

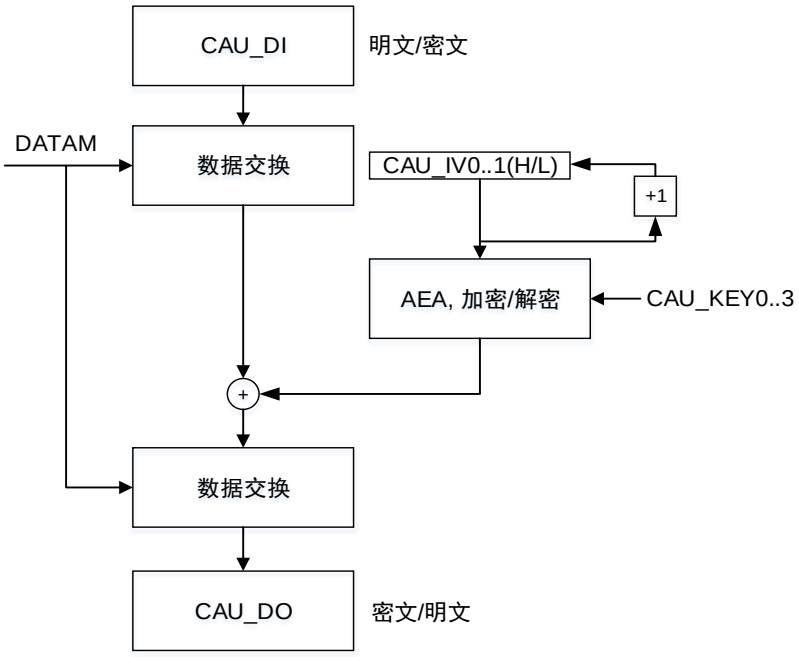


AES 计数器（CTR）模式

在计数器模式下，随机数与计数器的组合会作为AEA计算单元的输入来进行运算，运算结果会与输入的明文或密文进行异或，来求得最终加密或者解密的结果。由于加密和解密处理的计数器值是由相同的初始值进行递增的，因此加密和解密处理用的密钥序列是相同的。解密处理的操作与加密操作的流程完全相同。128位初始化向量的低32位表示为计数器值，这意味着其余96位在操作过程中保持不变，并且计数器的初始值应当设置为1。随机数是一个32位一次性值，应当更新到每个通信块。64位的初始化向量应确保每个给定值只用于一个给定密钥。计数器块框图结构见*图23-12. 计数器块结构*，AES计数器加密/解密流程图见*图23-13. AES CTR加密/解密*。

图 23-12. 计数器块结构

图 23-13. AES CTR 加密/解密



AES-GCM 模式

AES伽罗瓦/计数器模式(GCM)可用于加密或验证消息，来获得密文和标签。该算法基于AES计数器模式，保证了机密性。利用固定的有限域乘法运算来生成标签。

在该模式中，执行加密/解密需要四个步骤：

1. GCM准备阶段

   内部计算和保存哈希密钥以在后续使用。

   (a) 将CAUEN清零，禁能CAU；
   (b) 配置ALGM[3:0]位域为‘1000’；
   (c) 配置GCM_CCMPH[1:0]位域为‘00’；
   (d) 配置密钥寄存器CAU_KEY0..3(H/L)和初始化向量寄存器CAU_IV0..1(H/L)；
   (e) 置位CAUEN位，使能CAU；
   (f) 等待CAUEN位被硬件清零，然后再置位CAUEN，使能CAU，进行下个步骤。

2. GCM AAD（附加身份验证数据）阶段

   AAD阶段必须在GCM初始化阶段之后进行，并在加密解密阶段之前。在这个阶段，数据仅进行了验证，而没有被保密。

   (g) 配置GCM_CCMPH[1:0]位域为‘01’；
   (h) 将AAD数据写入CAU_DI寄存器，并使用CAU_STAT0寄存器的INF和IEM标志来判断输入FIFO是否能接收数据。AAD大小必须为128位的倍数。也可使用DMA来写入AAD数据。
   (i) 重复步骤(h)直到所有AAD数据都写入，并等待CAU_STAT0寄存器的BUSY位清零。

3. GCM加密解密阶段

加密解密阶段必须在GCM AAD阶段之后进行。在这个阶段，对消息进行了验证，并加密或解密。

(j) 配置GCM_CCMPH[1:0]位为‘10’；
(k) 配置CAUDIR位来选择算法方向；
(l) 将有效负载消息写入CAU_DI寄存器，并使用CAU_STAT0寄存器的INF和IEM标志来判断输入FIFO是否能接收数据。使用CAU_STAT0寄存器的ONE和OFU标志判断输出FIFO是否为空，如果不为空，就读取CAU_DO寄存器。也可使用DMA来写入有效负载消息。
(m) 重复步骤(l)直到所有的有效负载块都完成计算。

4. GCM标签阶段

在这个阶段，将生成最后的验证标签。

(n) 配置GCM_CCMPH[1:0]位为‘11’；
(o) 将最后的数据块（由64位AAD大小和64位有效负载消息大小组成）写入CAU_DI寄存器；
(p) 在完成写4次CAU_DI寄存器之后，等待CAU_STAT0寄存器的ONE标志置位，然后读取CAU_DO寄存器4次，这个输出数据就是最后生成的验证标签；
(q) 禁能CAU。

注意： 解密时，必须在开始阶段时准备好密钥。

AES-GMAC 模式

AES伽罗瓦消息验证码（GMAC）模式支持提供对消息的完整性验证。这个模式处理流程可视为AES-GCM模式流程除去加密解密阶段。

AES-CCM 模式

AES结合了类似于AES-GCM的密码机模式，支持消息的保密，以及完整性验证。AES-CCM模式基于AES-CTR模式来确保了消息的保密性，使用AES-CBC模式来生成128位标签。

CCM标准（RFC 3610 Counter with CBC-MAC (CCM) 标准，2003年9月发布）为首个验证块（在该标准中称为B0）定义了特定的编码规则，具体来说，首个块包括标志、随机数以及以字节计的有效负载大小。CCM标准为加密/解密指定了另外的格式，称为A或者计数器。计数器在有效负载阶段递增计数，在生成标签阶段计数器低32位有效位初始化为’1’（在CCM标准中称为A0数据包）。

注意：B0数据包的格式化操作需要在软件中处理完成。

在该模式中，执行加密/解密需要四个步骤：

1. CCM准备阶段

准备阶段，将B0数据包（首个块）写入CAU_DI寄存器。在这个阶段，CAU_DO寄存器不包含任何输出数据。

(a) 清零CAUEN位，禁能CAU；
(b) 配置ALGM[3:0]位域为‘1001’；

(c) 配置GCM_CCMPH[1:0]位域为‘00’；
(d) 配置密钥寄存器CAU_KEY0..3(H/L)和初始化向量寄存器CAU_IV0..1(H/L)；
(e) 置位CAUEN位，使能CAU；
(f) 将B0数据包写入CAU_DI寄存器；
(g) 等待CAUEN位被硬件清零，然后再置位CAUEN，使能CAU，进行下个步骤。

2. CCM AAD（附件身份验证数据）阶段

AAD阶段必须在CCM准备阶段之后进行，并在加密解密阶段之前。在这个阶段，CAU_DO寄存器不包含任何输出数据。

如果没有附加的验证数据，可以跳过这个阶段。

(h) 配置GCM_CCMPH[1:0]位域为‘01’；
(i) 将AAD数据写入到CAU_DI寄存器，并使用CAU_STAT0寄存器的INF和IEM标志来判断输入FIFO是否能接收数据。AAD大小必须为128位的倍数。也可以使用DMA来写入AAD数据。
(j) 重复步骤(i)直到所有的AAD数据都写入，并等待CAU_STAT0寄存器的BUSY位清零。

3. CCM加密解密阶段

加密解密阶段必须在CCM AAD阶段之后进行。在这个阶段，对消息进行了验证，并加密或解密。

与GCM类似，CCM链接模式可用于仅由经过验证的原文数据(即只有AAD，没有有效负载)组成的消息。需要注意的是，这种使用CCM的方式不称为CMAC(它与GCM/GMAC不同)。

(k) 配置GCM_CCMPH[1:0]位为‘10’；
(l) 配置CAUDIR位来选择算法方向；
(m) 将有效负载消息写入CAU_DI寄存器，并使用CAU_STAT0寄存器的INF和IEM标志来判断输入FIFO是否能接收数据。使用CAU_STAT0寄存器的ONE和OFU标志判断输出FIFO是否为空，如果不为空，就读取CAU_DO寄存器。也可使用DMA来写入有效负载消息。
(n) 重复步骤(m)直到所有的有效负载块都完成计算。

4. CCM标签阶段

在这个阶段，将生成最后的验证标签。

(o) 配置GCM_CCMPH[1:0]位为‘11’；
(p) 将128位A0数据包写入到CAU_DI寄存器，分为4次的写操作；
(q) 等待CAU_STAT0寄存器的ONE标志置位，然后读取CAU_DO寄存器4次，这个输出数据就是最后生成的验证标签；
(r) 禁能CAU。

AES-CFB 模式

密码反馈(CFB)模式是保密模式，其特征在于将连续密文段反馈到前向密码的输入块中，以生成与明文异或的输出块，从而产生密文，反之解密过程与加密的过程类似。

AES-OFB 模式

输出反馈(OFB)模式是保密模式，其特征在于在IV上对前向密码进行迭代，以生成与明文异或以产生密文的输出块序列，反之解密过程与加密的过程类似。

23.5. 操作模式

加密

1. 将CAU_CTL寄存器的CAUEN位清零，以禁用CAU；
2. 将PMU_CTL1寄存器中的CORE1WAKE置位以使能CAU的电源域，再打开CAU的外设时钟；
3. 若选择了AES算法，则对CAU_CTL寄存器的KEYM位进行选择设置，配置密钥的长度；
4. 根据算法配置CAU_KEY0..3(H/L)寄存器；
5. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型；
6. 设置CAU_CTL寄存器的ALGM[3:0]位，配置算法（DES/TDES/AES）和模式（ECB/CBC/CTR/GCM/GMAC/CCM/CFB/OFB）；
7. 设置CAU_CTL寄存器的CAUDIR位为0，配置为加密操作；
8. 设置CAU_IV0..1(H/L)寄存器，配置初始化向量；
9. 在CAUEN位为0时，设置CAU_CTL寄存器的FFLUSH位，配置刷新输入FIFO和输出FIFO；
10. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
11. 当CAU_STAT0寄存器的INF位为1时，向CAU_DI寄存器写数据块。数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
12. 等待CAU_STAT0寄存器的ONE位为1时，读CAU_DO寄存器。输出数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
13. 重复步骤10和步骤11，直到所有的数据块都完成加密；

解密

1. 将CAU_CTL寄存器的CAUEN位清零，以禁用CAU；
2. 将PMU_CTL1寄存器中的CORE1WAKE置位以使能CAU的电源域，再打开CAU的外设时钟；
3. 若选择了AES算法，则对CAU_CTL寄存器的KEYM位进行选择设置，配置密钥的长度；
4. 根据算法配置CAU_KEY0..3(H/L)寄存器；
5. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型；
6. 设置CAU_CTL寄存器的ALGM[3:0]位为“0111”，配置准备密钥用于解密；
7. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
8. 等待BUSY位和CAUEN位为0，确保解密用的密钥已准备好；
9. 设置CAU_CTL寄存器的ALGM[3:0]位，配置算法（DES/TDES/AES）和模式（ECB/CBC/CTR/GCM/GMAC/CCM/CFB/OFB）；
10. 设置CAU_CTL寄存器的CAUDIR位为1，配置为解密操作；
11. 设置CAU_IV0..1(H/L)寄存器，配置初始化向量；
12. 在CAUEN位为0时，设置CAU_CTL寄存器的FFLUSH位，配置刷新输入FIFO和输出FIFO；

13. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
14. 当CAU_STAT0寄存器的INF位为1时，向CAU_DI寄存器写数据块。数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
15. 等待CAU_STAT0寄存器的ONE位为1时，读CAU_DO寄存器。输出数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
16. 重复步骤13和步骤14，直到所有的数据块都完成解密；

数据填充

对于GCM加密和CCM解密，CAU模块支持非128比特整数倍的数据块处理。当最后一个数据块不满128比特时，使用‘0’对其剩余位进行填充，然后在CAU_CTL寄存器的NBPILB位域中配置用于填充的字节数，AES会自动去除相应填充数量的填充块后进行加密。需要注意的是，只有在倒数第二个数据块加密完成后，才可以对NBPILB位域进行配置。

23.6. CAU DMA 接口

DMA可用于CAU模块的数据块传输。DMA的传输操作由CAU_DMAEN寄存器来控制。DMAIEN位用于输入数据的DMA请求传输使能，由DMA将一个字数据写入CAU_DI寄存器。DMAOEN位用于输出数据的DMA请求传输使能，获得CAU输出的一个字。

DMA输出数据的传输请求优先级高于输入数据的传输请求，因此输出FIFO空的事件可能会早于输入FIFO满的事件。

23.7. CAU 中断

CAU有两个中断状态寄存器，CAU_STAT1和CAU_INTF寄存器。CAU中的中断用于指示输入和输出FIFO的状态。

可以通过配置CAU_INTEN寄存器来使能或禁用输入或输出FIFO中断。将寄存器中相应位置1可以使能相应中断。

输入 FIFO 中断

当输入FIFO中的数据少于4个字时产生输入FIFO中断，ISTA位置位。此时如果IINTEN位为1，使能了输入FIFO中断，则IINTF位将置位。注意当CAUEN位为0时，ISTA位和IINTF位将保持为0。

输出 FIFO 中断

当输出FIFO中存在一个或多个字数据时产生输出FIFO中断，OSTA位置位。此时如果OINTEN位为1从而使能了输出FIFO中断，则OINTF位将置位。注意与输入FIFO中断不同的是，当CAUEN位为0时，不会影响到OSTA位与OINTF位的状态。

23.8. CAU 挂起模式

当CAU中待处理的新的数据块优先级高于正在处理的数据块，则正在处理的数据块可能被挂起。按照下列的步骤来完成被挂起数据块的加密/解密处理。

当使用 DMA 进行数据传输：

1. 停止当前输入数据传输。将CAU_DMAEN寄存器的DMAIEN位清零。
2. 若为DES或AES算法，则需等待直到输入和输出FIFO均为空，如果检查到输入FIFO不为空即IEM位为0，则写入一个字的数据，再检查IEM位，直到IEM位为1，则停止写入数据，再等待BUSY位为0，以确保下一个数据块不会被上一个数据块影响。若为TDES算法，则与AES算法相似，但不需要等待输入FIFO为空。
3. 将CAU_DMAEN寄存器中的DMAOEN位清零，停止输出数据传输。并将CAU_CTL寄存器中的CAUEN位清零，禁用CAU。
4. 保存当前配置，包括密钥长度，数据类型，算法模式，算法方向，GCM CCM阶段，以及密钥值。若为CBC/CTR/GCM/GMAC/CCM/CFB/OFB模式，则还需要保存初始化向量。若为GCM/GMAC/CCM模式，则还需要保存上下文交换寄存器CAU_GCMCCMCTXSx (x=0..7)和CAU_GCMCTXSx (x=0..7)。
5. 配置并处理新数据块。
6. 恢复之前的处理环境。将CAU重新用存储的参数进行配置，并准备好密钥和初始化向量，还需恢复CAU_GCMCCMCTXSx (x=0..7)和CAU_GCMCTXSx (x=0..7)寄存器。再将CAU_CTL寄存器的CAUEN位置位以使能CAU。

当使用 CPU 来传输数据到 CAU_DI 和 CAU_DO：

1. 当使用CPU来进行数据传输，则需要等待第四次读CAU_DO寄存器，并在写CAU_DI之前，以确保一个数据块处理结束的时候再挂起消息处理。
2. 将CAU_CTL寄存器的CAUEN位清零，禁用CAU。
3. 保存当前配置，包括密钥长度，数据类型，算法模式，算法方向，GCM CCM阶段，以及密钥值。若为CBC/CTR/GCM/GMAC/CCM/CFB/OFB模式，则还需要保存初始化向量。若为GCM/GMAC/CCM模式，则还需要保存上下文交换寄存器CAU_GCMCCMCTXSx (x=0..7)和CAU_GCMCTXSx (x=0..7)。
4. 配置并处理新数据块。
5. 恢复之前的处理环境。将CAU重新用存储的参数进行配置，并准备好密钥和初始化向量，还需恢复CAU_GCMCCMCTXSx (x=0..7)和CAU_GCMCTXSx (x=0..7)寄存器。再将CAU_CTL寄存器的CAUEN位置位以使能CAU。

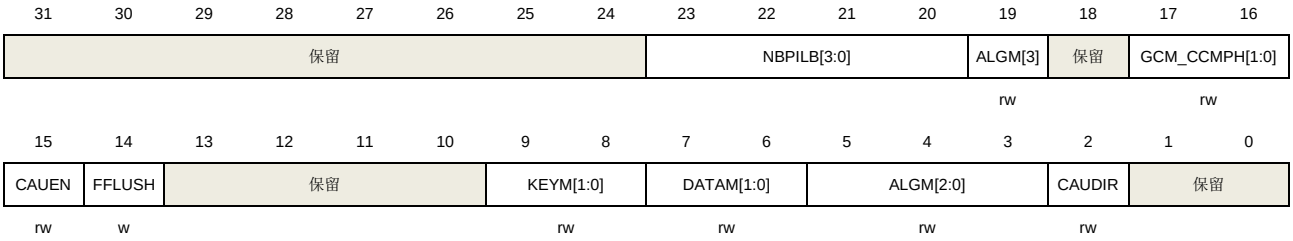
23.9. CAU 寄存器

CAU基地址：0x5006 0000

23.9.1. 控制寄存器（CAU_CTL）

偏移地址：0x00
复位值：0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:24	保留	必须保持复位值。
23:20	NBPILB[3:0]	最后一个非 128 比特整数倍数据块的填充字节数 0000：所有数据有效（无填充） 0001：一个填充字节 … 1111：15 个填充字节
19	ALGM[3]	加密/解密算法模式位 3
18	保留	必须保持复位值。
17:16	GCM_CCMPH[1:0]	GCM CCM 阶段 00：准备阶段 01：AAD 阶段 10：加密解密阶段 11：标签阶段
15	CAUEN	加密处理器使能 0：加密处理器禁用 1：加密处理器使能 **注意**：当准备密钥（ALGM=0111b）完成后，CAUEN 位将硬件自动清零。
14	FFLUSH	FIFO 刷新 0：不产生影响 1：当 CAUEN=1 时，刷新输入和输出 FIFO 读取该位时，始终返回 0
13:10	保留	必须保持复位值

9:8	KEYM[1:0]	AES 密钥长度配置，必须在 BUSY=0 时才可配置 00：128 位密钥长度 01：192 位密钥长度 10：256 位密钥长度 11：保留
7:6	DATAM[1:0]	数据交换模式配置，必须在 BUSY=0 时才可配置 00：不交换 01：半字交换 10：字节交换 11：位交换
5:3	ALGM[2:0]	加密/解密算法模式位 0 到位 2 该位域和位 19 必须在 BUSY=0 时才可配置。 0000：TDES-ECB（三重 DES 电子密码本），使用 CAU_KEY1, 2, 3. 不使用初始化向量（CAU_IV0..1） 0001：TDES-CBC（三重 DES 加密分组链接），使用 CAU_KEY1, 2, 3. 使用初始化向量（CAU_IV0）与数据块进行异或 0010：DES-ECB（DES 电子密码本），仅使用 CAU_KEY1 不使用初始化向量（CAU_IV0..1） 0011：DES-CBC（DES 加密分组链接），仅使用 CAU_KEY1 使用初始化向量（CAU_IV0）与数据块进行异或 0100：AES-ECB（AES 电子密码本），使用 CAU_KEY0, 1, 2, 3. 不使用初始化向量（CAU_IV0..1） 0101：AES-CBC（AES 加密分组链接），使用 CAU_KEY0, 1, 2, 3. 使用初始化向量（CAU_IV0..1）与数据块进行异或 0110：AES-CTR（AES 计数器模式），使用 CAU_KEY0, 1, 2, 3. 使用初始化向量（CAU_IV0..1）与数据块进行异或 该模式下，加密与解密处理相同，忽略 CAUDIR 位 0111：AES 解密密钥准备模式。输入密钥必须与加密处理中用的密钥相同。BUSY 位将保持置位直到完成密钥的准备，随后 CAUEN 位会清零。 1000：AES-GCM（伽罗瓦/计数器模式），该模式算法同样适用于GMAC算法。 1001：AES-CCM（加密分组链接-消息验证码模式）。 1010：AES-CFB（密码反馈模式） 1011：AES-OFB（输出反馈模式）
2	CAUDIR	CAU 算法方向，必须在 BUSY=0 时才可配置 0：加密 1：解密
1:0	保留	必须保持复位值

23.9.2. 状态寄存器 0 (CAU_STAT0)

偏移地址：0x04

复位值：0x0000 0003

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											BUSY	OFU	ONE	INF	IEM
											r	r	r	r	r

位/位域	名称	描述
31:5	保留	必须保持复位值。
4	BUSY	忙碌标志位 0：CAU 内核空闲，这是由于 - CAUEN=0 从而 CAU 内核被禁用，或这处理已完成 - 正在等待输入数据或输出 FIFO 有足够的自由空间来处理数据块 1：CAU 内核忙碌，正在处理数据块或准备密钥
3	OFU	输出 FIFO 满 0：输出 FIFO 未满 1：输出 FIFO 满
2	ONE	输出 FIFO 非空 0：输出 FIFO 为空 1：输出 FIFO 非空
1	INF	输入 FIFO 未满 0：输入 FIFO 满 1：输入 FIFO 未满
0	IEM	输入 FIFO 空 0：输入 FIFO 非空 1：输入 FIFO 空

23.9.3. 数据输入寄存器 (CAU_DI)

偏移地址：0x08
复位值：0x0000 0000

数据输入寄存器用于传输明文或密文数据块到输入FIFO中进行处理。首先写入FIFO的是数据块的MSB，最后才是LSB。当CAUEN位为0，并且输入FIFO非空时，读取该寄存器时返回FIFO中的首个字。当CAUEN位为1时，读取该寄存器返回一个不确定的值。一旦执行了读操作，则必须要刷新FIFO以处理新数据块。

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DI[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DI[15:0]															
rw															

位/位域	名称	描述
31:0	DI[31:0]	数据输入 写这些位，数据会写入输入 FIFO。当 CAUEN 位为 0 时，读这些位将返回输入 FIFO 中的值，否则将返回不确定的值。

23.9.4. 数据输出寄存器 (CAU_DO)

偏移地址：0x0C
复位值：0x0000 0000

数据输出寄存器是只读寄存器，用于接收来自输出FIFO的明文或密文处理结果。与CAU_DI类似，读取时首先读取的是数据块的MSB，最后才是LSB。

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO[15:0]															
r															

位/位域	名称	描述
31:0	DO[31:0]	数据输出 这些位为只读，读这些位将返回输出 FIFO 中的值

23.9.5. DMA 使能寄存器 (CAU_DMAEN)

偏移地址：0x10
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														DMAOEN	DMAIEN
														rw	rw

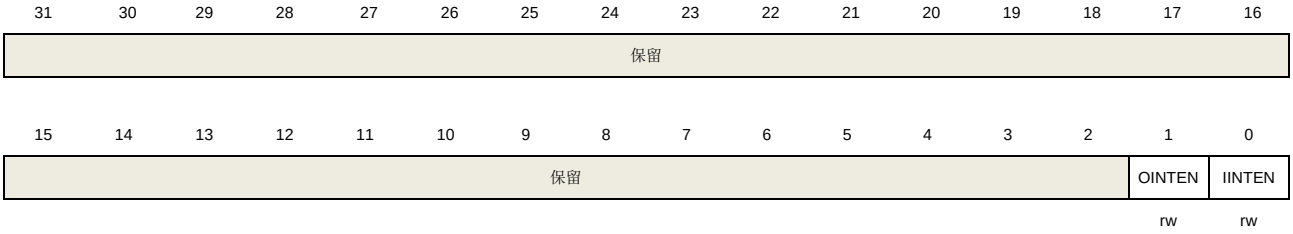
位/位域	名称	描述
31:2	保留	必须保持复位值。

1	DMAOEN	DMA 输出使能 0：禁用用于输出 FIFO 数据传输的 DMA 1：使能用于输出 FIFO 数据传输的 DMA
0	DMAIEN	DMA 输入使能 0：禁用用于输入 FIFO 数据传输的 DMA 1：使能用于输入 FIFO 数据传输的 DMA

23.9.6. 中断使能寄存器 (CAU_INTEN)

偏移地址：0x14
复位值：0x0000 0000

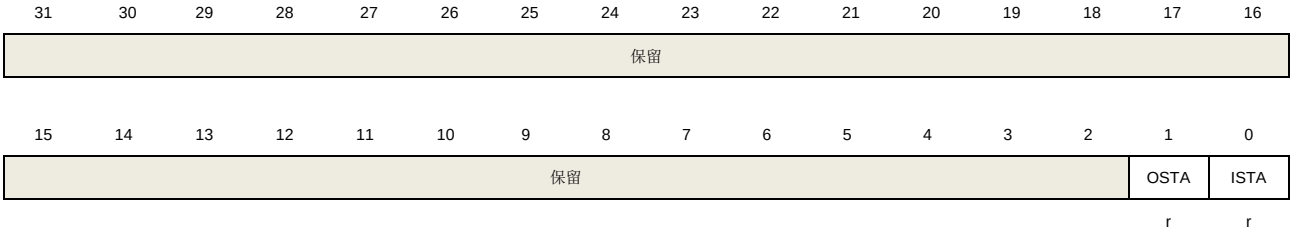
该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:2	保留	必须保持复位值。
1	OINTEN	输出 FIFO 中断使能 0：禁用输出 FIFO 中断 1：使能输出 FIFO 中断
0	IINTEN	输入 FIFO 中断使能 0：禁用输入 FIFO 中断 1：使能输入 FIFO 中断

23.9.7. 状态寄存器 1 (CAU_STAT1)

偏移地址：0x18
复位值：0x0000 0001
该寄存器只能按字(32位)访问。



位/位域	名称	描述

31:2	保留	必须保持复位值。
1	OSTA	输出 FIFO 状态 0：输出 FIFO 状态未挂起 1：输出 FIFO 状态挂起
0	ISTA	输入 FIFO 状态 0：输入 FIFO 状态未挂起 1：输入 FIFO 状态挂起

23.9.8. 中断标志寄存器 (CAU_INTF)

偏移地址：0x1C
复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														OINTF	IINTF
														r	r

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	OINTF	输出 FIFO 中断标志 0：输出 FIFO 中断状态未挂起 1：输出 FIFO 中断状态挂起
0	IINTF	输入 FIFO 中断标志 0：输入 FIFO 中断状态未挂起 1：当 CAUEN 位为 1 时输入 FIFO 中断状态挂起

23.9.9. 密钥寄存器 (CAU_KEY0..3(H/L))

偏移地址：0x20~0x3C
复位值：0x0000 0000

该寄存器只能按字(32位)访问，必须在BUSY位为0时写这些寄存器。

在DES模式下，仅使用CAU_KEY1。

在TDES模式下，使用CAU_KEY1，CAU_KEY2和CAU_KEY3。

在AES-128模式下，KEY2H[31:0]和KEY2L[31:0]分别对应于AES_KEY[0:63]的高32位与低32位，而KEY3H[31:0]和KEY3L[31:0]分别对应于AES_KEY[64:127]的高32位与低32位。

在AES-192模式下，KEY1H[31:0]和KEY1L[31:0]分别对应于AES_KEY[0:63]的高32位与低32

位， KEY2H[31:0] 和 KEY2L[31:0] 分别对应于 AES_KEY[64:127] 的高 32 位与低 32 位，KEY3H[31:0]和KEY3L[31:0]分别对应于AES_KEY[128:191]的高32位与低32位。

在AES-256模式下，KEY0H[31:0]和KEY0L[31:0]分别对应于AES_KEY[0:63]的高32位与低32位， KEY1H[31:0] 和 KEY1L[31:0] 分别对应于 AES_KEY[64:127] 的高 32 位与低 32 位，KEY2H[31:0]和KEY2L[31:0]分别对应于AES_KEY[128:191]的高32位与低32位，KEY3H[31:0]和KEY3L[31:0]分别对应于AES_KEY[192:255]的高32位与低32位。

CAU_KEY0H

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY0H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0H[15:0]															
w															

CAU_KEY0L

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY0L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0L[15:0]															
w															

CAU_KEY1H

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY1H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1H[15:0]															
w															

CAU_KEY1L

偏移地址：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY1L[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1L[15:0]															
w															

CAU_KEY2H

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY2H[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2H[15:0]															
w															

CAU_KEY2L

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY2L[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2L[15:0]															
w															

CAU_KEY3H

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY3H[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY3H[15:0]															
w															

CAU_KEY3L

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

KEY3L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY3L[15:0]															
w															

位/位域	名称	描述
31:0	KEY0...3(H/L)	用于 DES 或 TDES 或 AES 的密钥

23.9.10. 初始化向量寄存器 (CAU_IV0..1(H/L))

偏移地址：0x40~0x4C
复位值：0x0000 0000

该寄存器只能按字(32位)访问，必须在BUSY位为0时写这些寄存器。

在DES/TDES模式下，IV0H和IV0L分别对应于初始化向量的高32位和低32位。

在AES模式下，IV0H和IV1H分别对应于128位初始化向量的最高32位和最低32位。

CAU_IV0H

偏移地址：0x40
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0H[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0H[15:0]															
rw															

CAU_IV0L

偏移地址：0x44
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0L[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0L[15:0]															
rw															

CAU_IV1H

偏移地址：0x48
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV1H[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1H[15:0]															
rw															

CAU_IV1L

偏移地址：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV1L[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1L[15:0]															
rw															

位/位域	名称	描述
31:0	IV0...1(H/L)	用于 DES 或 TDES 或 AES 的初始化向量

23.9.11. GCM 或 CCM 模式上下文交换寄存器 x (CAU_GCMCCMCTXSx) (x=0..7)

偏移地址：0x50 to 0x6C

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTXx[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTXx[15:0]															
rw															

位/位域	名称	描述
31:0	CTXx[31:0]	CAU处理器的内部状态信息。当有一个更高优先级的任务需要处理时，读取并保存这些寄存器的数据，恢复的时候将保存的数据写回到这些寄存器从而恢复前面被挂起的任务。 **注意：**这些寄存器只能在GCM，GMAC，或CCM模式下使用。

23.9.12. GCM 模式上下文交换寄存器 x (CAU_GCMCTXSx) (x=0..7)

偏移地址：0x70 to 0x8C

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTXx[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTXx[15:0]															
rw															

位/位域	名称	描述
31:0	CTXx[31:0]	CAU处理器的内部状态信息。当有一个更高优先级的任务需要处理时，读取并保存这些寄存器的数据，恢复的时候将保存的数据写回到这些寄存器从而恢复前面被挂起的任务。 **注意：**这些寄存器只能在GCM或GMAC模式下使用。

24. VREF

24.1. 简介

MCU 有一个精准的内部参考电路，用于为 ADC/DAC 提供基准电压，或由连接到 VREF 引脚的片外电路使用。

24.2. 主要特征

精准的内部参考特性描述如下：

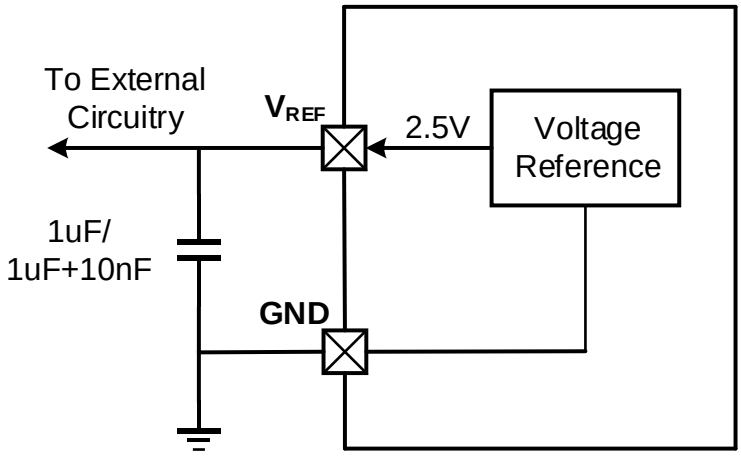
- 电压稳定，产品修整；
- 连接 V_{REF} 引脚至片外电路；
- 提供 2.5 V 参考电压。

24.3. 功能描述

通过将 VREF_CS 寄存器中的 VREFEN 位置 1 使能该精准的内部参考（在这之前需要将 RCU_APB2EN 寄存器中的 SYSCFGEN 位置 1），产生 2.5V 参考电压并连接到 V_{REF} 引脚。当 VREFEN 被禁用时，可将片外参考电压注入到 V_{REF} 引脚作为 ADC/DAC 的参考源。如果没有 V_{REF} 引脚(请参阅数据手册)，则 V_{REF} 连接到 V_{DDA}，VREFEN 位必须保持 0。

当使用精准的内部参考电压时，建议连接一个 1uF（或 1uF 和 10nF 并联）的旁路电容，并接地。

图 24-1. VREF 连接



根据 VREFEN 和 HIPM 位的配置，内部参考电压可以被配置成四种不同的模式。这些模式如下表所示：

表 24-1 VREF 模式

VREFEN	HIPM	模式
0	0	VREF 失能

VREFEN	HIPM	模式
		– V_{REF} 引脚下拉到 V_{SSA}
0	1	外部参考电压模式： – VREF 失能 – V_{REF} 引脚浮空
1	0	内部参考电压模式： – VREF 使能 – V_{REF} 引脚连接到 VREF 输出
1	1	保持模式： – VREF 失能 – V_{REF} 引脚浮空。通过外部电容保持电压 – 失能 VREFRDY 位检测，VREFRDY 位保持最后一个状态

24.4. VREF 寄存器

VREF 基地址：0x4001 0030

24.4.1. 控制和状态寄存器（VREF_CS）

地址偏移：0x00
复位值：0x0000 0002

该寄存器可以按半字（16 位）或字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												VREFRDY	保留	HIPM	VREFEN
												r		rw	rw

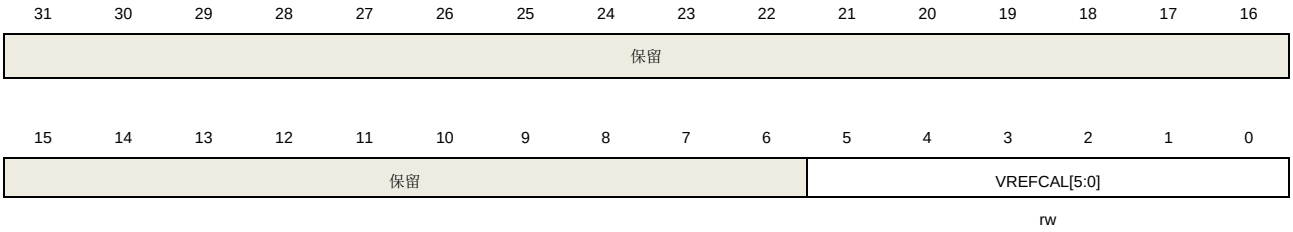
位/位域	名称	描述
31:4	保留	必须保持复位值。
3	VREFRDY	VREF就绪 0：VREF输出未就绪 1：VREF输出就绪
2	保留	必须保持复位值
1	HIPM	高阻抗模式 0：V_{REF+}内部连接到 VREF 输出 1：V_{REF+}引脚为高阻抗
0	VREFEN	VREF使能 0：VREF失能 1：VREF使能

24.4.2. 校准寄存器（VREF_CALIB）

地址偏移：0x04

复位值：0x0000 00xx

该寄存器可以按半字（16 位）或字（32 位）访问



位/位域	名称	描述
31:6	保留	必须保持复位值。
5:0	VREFCAL	VREF校准 复位后，这些位将在生产测试期间使用存储在FLASH中的修整值自动初始化。写入这些位可调节内部V_{REF}电压。

25. 段码 LCD 控制器（SLCD）

25.1. 简介

SLCD 驱动器通过自动产生 SEG 和 COM 交流电压信号来直接驱动 LCD 显示。该驱动器可以驱动单色液晶显示器（LCD），这是一种由若干段（像素或完整的符号）构成的，有可见和不可见两种状态的显示屏。SLCD 驱动器支持最大 32 个 SEG 和 8 个 COM。

25.2. 主要特征

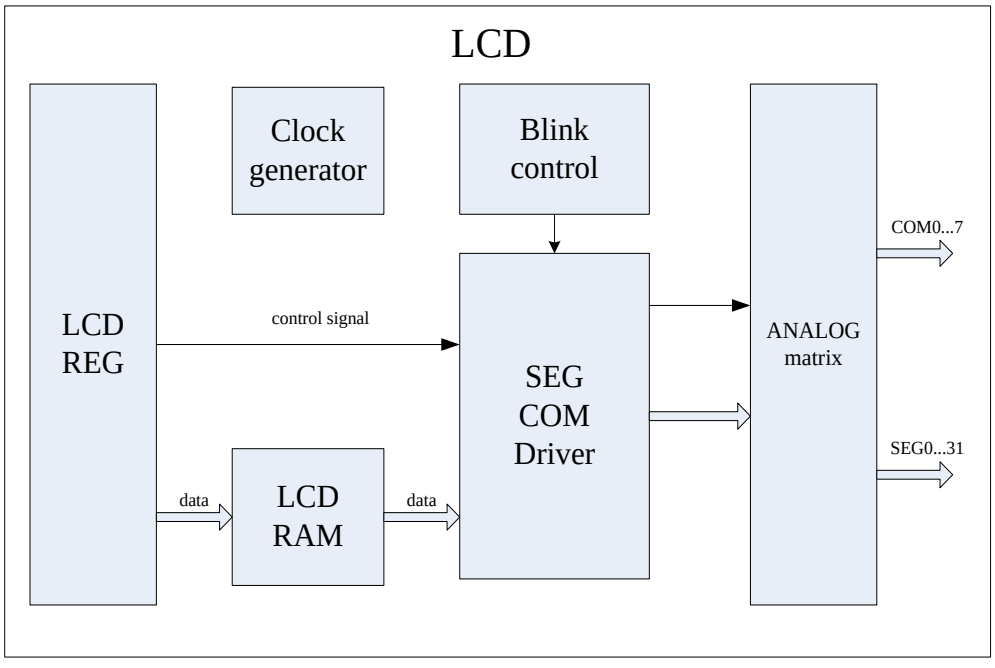
- 可配置帧率；
- 单个SEG或所有SEG的闪烁；
- 支持静态、1/2、1/3、1/4、1/6和1/8占空比；
- 支持1/2、1/3和1/4偏置；
- 双路缓冲器可多达8x32位寄存器来存储SLCD_DATAx；
- 对比度也可通过配置死区时间来调整；
- 可配置电压输出驱动用于增强SLCD驱动能力。

25.3. 功能描述

25.3.1. SLCD 架构

SLCD控制器框图如下所示：

图 25-1. SLCD 模块框图



SLCD REG 是 SLCD 控制器的寄存器，包括 SLCD_CTL、SLCD_CFG、SLCD_STAT、SLCD_STATC 和 SLCD_DATAx 五个寄存器，它们可通过 APB 总线配置，且可使 CPU 产生中断。

时钟发生器可以从输入时钟产生 SLCD 时钟，SLCD 时钟可以驱动闪烁控制和 SEG/COM 驱动器。闪烁控制可以产生闪烁频率和闪烁像素，SEG/COM 驱动器可产生 SEG 和 COM 信号输送到 ANALOG 矩阵，且 ANALOG 矩阵可实现 SEG 和 COM 电压。

25.3.2. 时钟发生器

SLCD 输入时钟与 RTCCLK 共用，允许三种不同的时钟源：通过配置 RCU_BDCTL 寄存器的 RTCSRC 位域，可以选择 LXTAL、IRC32K 或 HXTAL 的 32 分频。输入时钟频率变化范围为 32KHz 到 1MHz。

SLCD 控制器可使用从集成时钟分频器输出的 SLCD 时钟信号来产生 SEG 和 COM 线的时序。SLCD 时钟信号来自于 RCU 输入时钟。SLCD 时钟频率可在 SLCD_CFG 寄存器中的 PSC 位和 DIV 位来选择，由此产生的时钟频率计算结果如下：

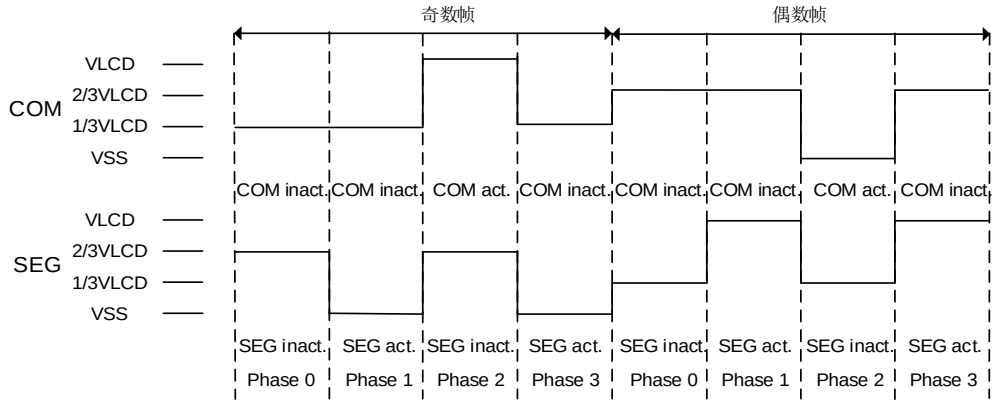
$$f_{SLCD} = \frac{f_{in\_clk}}{2^{PSC} \times (DIV+16)} \quad (25\text{-}1)$$

SLCD 时钟作为 SLCD 控制器的时钟基准。SLCD 的时钟频率相当于相频率。一个 SLCD 帧是一个奇数帧或一个偶数帧，它们拥有与有效的 COM 一样多的相。占空比定义为 1/（SLCD 屏显示需要的 COM 数）。因此帧频率计算结果如下：

$$f_{frame} = f_{SLCD} \times Duty \quad (25\text{-}2)$$

在帧起始，SLCD_STAT 寄存器的 SOF 位由硬件置位，如果 SLCD_CFG 寄存器的 SOFIE 位被置位，SLCD 中断将被执行。向 SLCD_STATC 寄存器的 SOFC 位写 1 将清除 SOF 位。

图 25-2. 1/3 偏置，1/4 占空比



25.3.3. 闪烁控制

SLCD 控制器也支持闪烁功能。闪烁模式可通过 SLCD_CFG 寄存器中的 BLKMOD 位来控制，BLKMOD = 01 表示允许在 SEG0 和 COM0 上闪烁单个段，BLKMOD = 10 表示允许闪烁所有 COM 和 SEG0，BLKMOD = 11 表示允许闪烁所有 COM 和所有 SEG，BLKMOD = 00 表示禁用闪烁。

闪烁频率源于 SLCD 时钟，可通过 SLCD_CFG 中 BLKDIV 位来选择，由此产生的 BLINK 频率计算结果如下：

$$f_{BLINK} = \frac{f_{SLCD}}{2^{(BLKDIV+3)}} \quad (25\text{-}3)$$

在选择 BLKMOD = 01，10 或 11 中的一种闪烁模式后，使能的 SEG 或所有 SEG 都会在下一帧分界变成空白，且会停留半个 BLKCLK 周期，然后它们会在一个帧分界再次变白之前完成下一帧分界激活并继续停留另半个 BLKCLK 周期时间。

25.3.4. SEG/COM 驱动器

SEG/COM 驱动器可以产生 SEG 和 COM 信号。

偏置发生器：

偏置可通过 SLCD_CTL 寄存器中 BIAS 位来选择，仅在对应的一个帧周期的段内可以达到最大幅值 VSLCD 或 VSS。奇数帧电压和偶数帧电压如下表所示：

表25-1. 奇数帧电压

偏置	静态	1/2 偏置	1/3 偏置	1/4 偏置
COM 有效	VSLCD	VSLCD	VSLCD	VSLCD
COM 无效	/	1/2 VSLCD	1/3 VSLCD	1/4 VSLCD
SEG 有效	VSS	VSS	VSS	VSS
SEG 无效	VSLCD	VSLCD	2/3 VSLCD	1/2 VSLCD

表25-2. 偶数帧电压

偏置	静态	1/2 偏置	1/3 偏置	1/4 偏置
COM 有效	VSS	VSS	VSS	VSS
COM 无效	/	1/2 VSLCD	2/3 VSLCD	3/4 VSLCD
SEG 有效	VSLCD	VSLCD	VSLCD	VSLCD
SEG 无效	VSS	VSS	1/3 VSLCD	1/2 VSLCD

COM信号：

COM 信号可通过 SLCD_CTL 寄存器中的 DUTY 位来选择。当 DUTY 是 000 时，静态占空比被选择，仅 COM[0]被使用，且仅一段在奇数帧或偶数帧内，COM[0]驱动器信号一直有效。当 DUTY 是 001 时，仅 COM[1:0]和 2 段被使用。当 DUTY 是 010 时，仅 COM[2:0]和 3 段被使用。当 DUTY 是 011 时，仅 COM[3:0]和 4 段被使用。当 DUTY 是 100 时，COM[7:0]和 8 段被使用。当 DUTY 是 101 时，COM[5:0]和 6 段被使用。所有的 COM 信号驱动器如下表所示：

表25-3. 所有COM信号驱动器

段	1	2	3	4	5	6	7	8
COM0	有效	无效	无效	无效	无效	无效	无效	无效
COM1	无效	有效	无效	无效	无效	无效	无效	无效
COM2	无效	无效	有效	无效	无效	无效	无效	无效
COM3	无效	无效	无效	有效	无效	无效	无效	无效
COM4	无效	无效	无效	无效	有效	无效	无效	无效
COM5	无效	无效	无效	无效	无效	有效	无效	无效

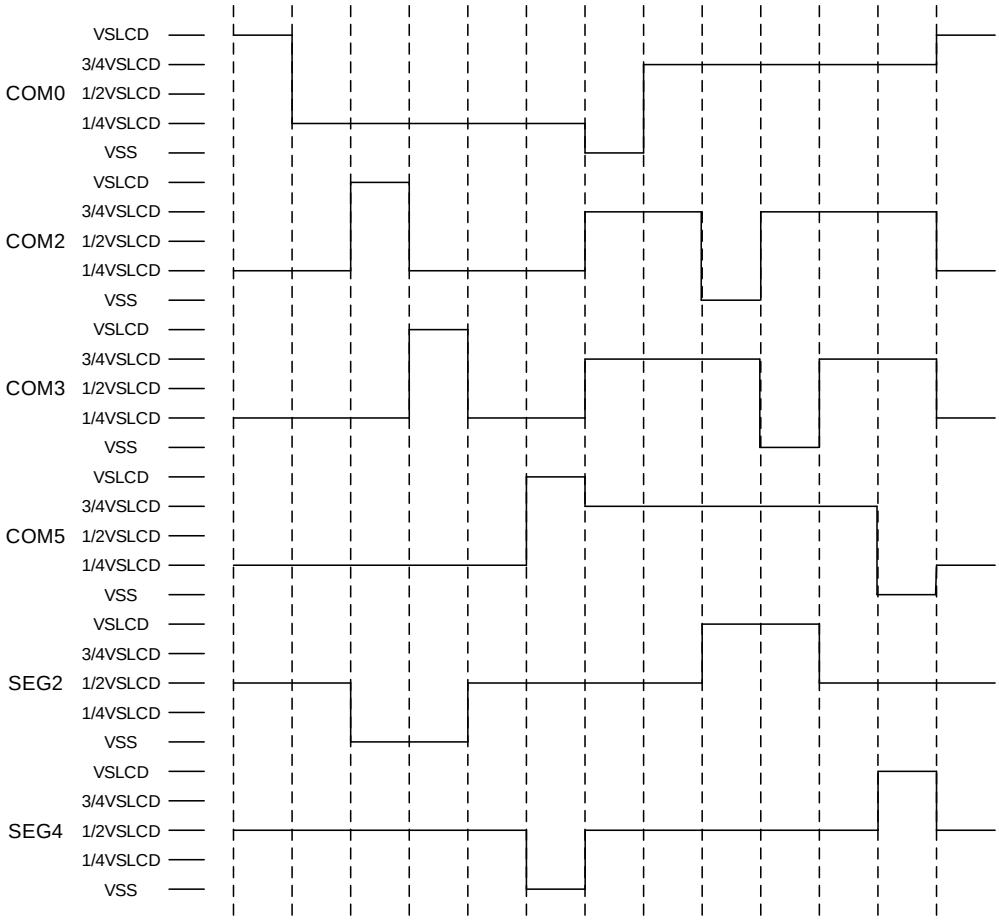
COM6	无效	无效	无效	无效	无效	无效	有效	无效
COM7	无效	无效	无效	无效	无效	无效	无效	有效

SEG信号：

SEG 信号从 SLCD_DATAx 寄存器中读取。SLCD_DATAx 表示相 x 的 SEG 信号数据。当该位为 1 时，对应的 SEG 驱动有效信号，当该位为 0 时，对应的 SEG 驱动无效信号。

例如，应用程序需要激活像素 COM2-SEG2，COM3-SEG2 和 COM5-SEG4，则 SLCD_DATA2 寄存器的 bit2、SLCD_DATA3 寄存器的 bit2、SLCD_DATA5 寄存器的 bit4 应被置位。这样 SEG2 信号将在每个奇数帧与偶数帧的第三、四相变为有效，SEG4 信号将在每个奇数帧与偶数帧的第六相变为有效。有效与无效电压请参考*表 25-1. 奇数帧电压*和*表 25-2. 偶数帧电压*。*图 25-3. 1/4 偏置 1/6 占空比*展现了当配置为 1/4 偏置，1/6 占空比时的 SEG 信号。

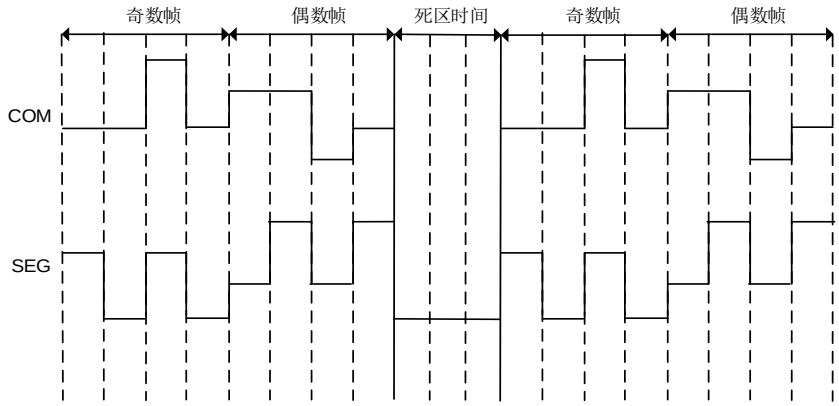
图 25-3. 1/4 偏置 1/6 占空比



死区时间：

死区时间通过 SLCD_CFG 寄存器中 DTD 位来配置，它在每个偶数帧后插入 VSS，插入的段数时间由 DTD 位定义。应用程序可以通过配置死区时间调节对比度。

图 25-4. SLCD 死区时间（1/3 偏置，1/4 占空比）



25.3.5. 双缓冲存储

双缓冲存储用于确保显示信息的连续性。

应用程序通过修改 SLCD_DATAx 寄存器组访问第一缓冲区。在将显示信息写入 SLCD_DATAx 寄存器组后，应用程序需要将 SLCD_STAT 寄存器的 UPRF 位置位，之后硬件将数据从第一缓冲区传入第二缓冲区，在传输的这段时间内，UPRF 位将保持置位，同时 SLCD_DATAx 寄存器组将写保护。当传输结束后，UPRF 位被清除同时 UPDF 位将被硬件置位，如果 UPDIE 位被置位，将会产生一个中断。SEG 信号由第二缓冲区内的数据驱动，因此写 SLCD_DATAx 寄存器组将不会对显示信息造成影响。

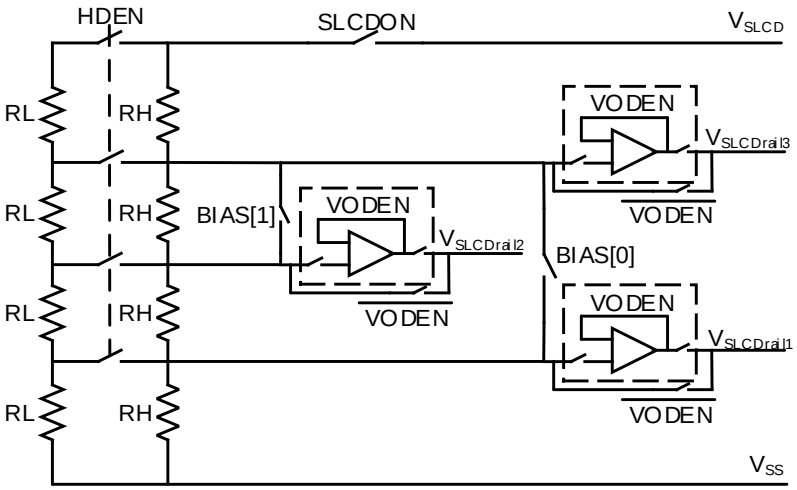
如果 UPRF 位在显示失能（SLCDON=0）时被置位，传输将不会发生，直到 SLCDON 置位。

25.3.6. 模拟矩阵

模拟矩阵提供 SLCD 电压。SLCD 电压电平可由 VSLCD 引脚或内部电压升压转换器产生（由 SLCD_CTL 寄存器中 VSRC 位选择）。当使用内部电压源时，VSLCD 的值可以通过配置 SLCD_CFG 寄存器的 CONR[2:0]位域从 VSLCD0 到 VSLCD7 中选择（VSLCDx 的值请参考产品数据手册）。应用程序可以通过改变 VLCD 的值调节对比度。另外一种调节对比度的方法是使用死区时间。

模拟矩阵通过如*图 25-5. 电阻分压网络*所示的内部电阻分压网络提供 VSS 和 VSLCD 中间的电压电平（1/3 VSLCD、2/3 VSLCD 或 1/4 VSLCD、2/4 VSLCD 和 3/4 VSLCD）。

图 25-5. 电阻分压网络



在转换期间，为了快速到达静态状态，低值电阻（R_L）被使用以增加电流。之后低值电阻被关闭，高值电阻（R_H）被使用以减小功耗。R_L 被使用的时间长度依据 SLCD_CFG 寄存器的 PULSE[2:0]位域的值。通过配置 SLCD_CFG 寄存器的 HDEN 位，R_L 可以一直被使用。

增强模式：

SLCD 模块集成了可配置的电压输出驱动器，可通过配置 SLCD_CTL 寄存器的 VODEN 位进入增强模式。启用电压输出驱动器，可以减少由电桥上的 LCD 电容负载引起的电压干扰，从而获得稳定的电压，达到增强 SLCD 驱动能力的目的。

在增强模式下，高值电阻器桥（RHN）将产生一个中间电压，HDEN 位或 PULSE 位配置将被忽略，低值电阻器桥（RLN）将被自动禁用，从而降低了功耗。

VLCD 电源不用于电压输出驱动器。仅当未激活 SLCD 控制器时才能配置电压输出驱动器。

25.3.7. V_{SLCD} 电压源

V_{SLCD} 电压监测

ADC_CTL1 寄存器中的 VSLCDEN 位用于使能 VSLCD 电压测量。由于 VSLCD 电压可能高于 VDDA，因此为了确保 ADC 的正常工作，内部 VSLCDrail1 模拟电压连接到 ADC_IN19 输入通道。

在不同 BIAS[1:0]偏置下，VSLCDrail1 的值是不同的，这是由内部模拟电路决定的。

1. BIAS [1:0] = 00，$V_{SLCDrail1} = 1 / 4V_{SLCD}$，可以使用 V_{SLCD} 电压监视功能，通过 ADC 转换获得的值是 V_{SLCD} 电压的四分之一。
2. BIAS [1:0] = 01，$V_{SLCDrail1}$ 是无效值，并且 VSLCD 电压监视功能无法正常使用。
3. BIAS [1:0] = 10，$V_{SLCDrail1} = 1 / 3V_{SLCD}$，可以使用 V_{SLCD} 电压监视功能，通过 ADC 转换获得的值是 V_{SLCD} 电压的三分之一。

为防止意外消耗电池电量，建议仅在执行 ADC 转换时才启用 VSLCDEN。

V_{SLCD} 电压源配置

SLCD 的电压源可通过 SLCD_CTL 寄存器的 VSRC 位配置为内部电压源或外部电压源。SLCD 使用内/外电压源的注意事项如下：

内部电压源：

当 SLCD 选择内部电压源时，PD6 引脚需要配置为模拟模式，且在与 GND 之间需外接一个电容，其电容值请参考 Datasheet。当 SLCD 选取内部电压源，其配置流程需遵循以下方式。

1. 配置 PD6 引脚为模拟模式。
2. 配置 SLCD 寄存器，并选择内部电压源。
3. 等待外接电容完成充电（典型情况，当外接电容为 2uF 时，充电时间约为 1.5ms）。
4. 使能 SLCD 模块。

外部电压源：

当 SLCD 选择外部电压源时，PD6 引脚需要配置为模拟模式，并连接外部供电电压源。其配置流程需遵循以下方式。

1. 配置 PD6 引脚为模拟模式。
2. 配置 SLCD 寄存器，并选择外部电压源。
3. 使能 SLCD 模块。

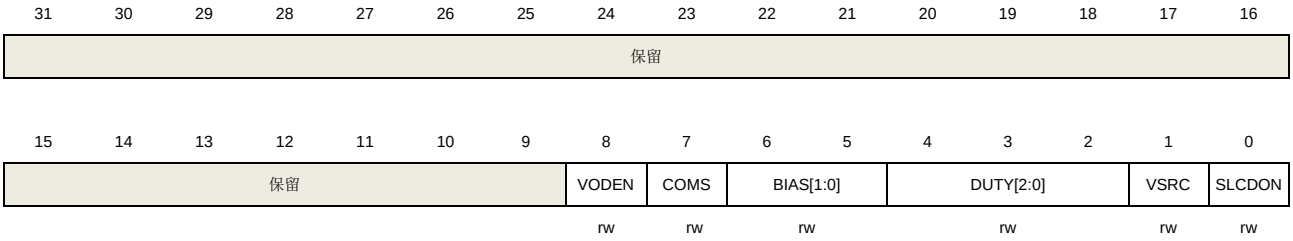
25.4. SLCD 寄存器

SLCD 基地址：0x4000 2400

25.4.1. 控制寄存器（SLCD_CTL）

偏移地址：0x00
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。



位/位域	名称	描述
31:9	保留	必须保持复位值。
8	VODEN	电压输出驱动使能 0：电压输出驱动禁止 1：电压输出驱动使能 当 VODEN=1 时，可以提高 SLCD 的电压驱动能力。
7	COMS	COM/SEG 引脚选择 该位用于 COM/SEG 引脚的选择。当占空比选择 1/8 或 1/6 时，SLCD_COM[7:4]总是作为 SLCD_COM[7:4]功能，无论该位有无被置位。 0：SLCD_COM[7:4]引脚选择 SLCD_COM[7:4] 1：SLCD_COM[7:4]引脚选择 SLCD_SEG[31:28]
6:5	BIAS[1:0]	偏置选择 偏置为驱动 SLCD 时的电压水平参数。它被定义为 1/（驱动 SLCD 显示屏能够使用的电压水平总数-1）。 00：1/4 偏置（5 个电压水平：VSS，1/4VSLCD，1/2VSLCD，3/4VSLCD，VSLCD） 01：1/2 偏置（3 个电压水平：VSS，1/2VSLCD，VSLCD） 10：1/3 偏置（4 个电压水平：VSS，1/3VSLCD，2/3VSLCD，VSLCD） 11：保留
4:2	DUTY[2:0]	占空比选择 这些位决定占空比。占空比定义为 1/（SLCD 显示屏需要的 COM 数） 000：静态占空比 001：1/2 占空比 010：1/3 占空比 011：1/4 占空比 100：1/8 占空比 101：1/6 占空比

		110：保留 111：保留
1	VSRC	SLCD 电压源 配置该位决定 SLCD 电压源。 0：内部电压源 1：外部电压源（VSLCD 引脚）
0	SLCDON	SLCD 控制器开始 通过软件置 1 该位开始 SLCD 控制器。通过软件清零该位停止 SLCD 控制器，且 SLCD 控制器在下一帧开始时停止。 0：SLCD 控制器停止 1：SLCD 控制器开始

25.4.2. 配置寄存器（SLCD_CFG）

偏移地址：0x04
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						PSC[3:0]				DIV[3:0]				BLKMOD[1:0]	
						rw				rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLKDIV[2:0]			CONR[2:0]			DTD[2:0]			PULSE[2:0]			UPDIE	保留	SOFIE	HDEN
rw			rw			rw			rw			rw		rw	rw

位/位域	名称	描述
31:26	保留	必须保持复位值。
25:22	PSC[3:0]	SLCD 时钟预分频器 配置这些位定义 SLCD 时钟预分频器。 0000：$f_{PSC} = f_{in\_clk}$ 0001：$f_{PSC} = f_{in\_clk}/2$ 0010：$f_{PSC} = f_{in\_clk}/4$ ... 1111：$f_{PSC} = f_{in\_clk}/32768$
21:18	DIV[3:0]	SLCD 时钟分频器 配置这些位定义 DIV 分频器的分频因子。 0000：$f_{SLCD} = f_{PSC}/16$ 0001：$f_{SLCD} = f_{PSC}/17$ 0010：$f_{SLCD} = f_{PSC}/18$ ... 1111：$f_{SLCD} = f_{PSC}/31$
17:16	BLKMOD[1:0]	闪烁模式

00：不闪烁
01：闪烁 SEG[0]、COM[0]（1 像素）
10：闪烁 SEG[0]和所有 COM（由可编程占空比可实现最大 8 像素）
11：闪烁所有 SEG 和所有 COM（所有像素）

15:13 BLKDIV[2:0] 闪烁分频器
000：f_{BLINK} =f_{SLCD}/8
001：f_{BLINK} =f_{SLCD}/16
010：f_{BLINK} =f_{SLCD}/32
011：f_{BLINK} =f_{SLCD}/64
100：f_{BLINK} =f_{SLCD}/128
101：f_{BLINK} =f_{SLCD}/256
110：f_{BLINK} =f_{SLCD}/512
111：f_{BLINK} =f_{SLCD}/1024

12:10 CONR[2:0] 对比度
当选择内部电压源（VSRC=0）时，这些位表示 VSLCD 电压，其范围从 VSLCD0 到 VSLCD7（典型值为 2.65V 到 3.67V），VSLCDx 值请参考数据手册。当使用外部电压源时（VSRC=1）这些位无效。
000：VSLCD0
001：VSLCD1
010：VSLCD2
011：VSLCD3
100：VSLCD4
101：VSLCD5
110：VSLCD6
111：VSLCD7

9:7 DTD[2:0] 死区时间
配置这些位定义帧间死区时间的长度。
000：无死区时间
001：1 相周期死区时间
010：2 相周期死区时间
...
111：7 相周期死区时间

6:4 PULSE[2:0] 脉冲持续时间
配置这些位根据 PSC 脉冲来定义脉冲持续时间。
000：0
001：1/f_{PSC}
010：2/f_{PSC}
011：3/f_{PSC}
100：4/f_{PSC}
101：5/f_{PSC}
110：6/f_{PSC}
111：7/f_{PSC}

3	UPDIE	SLCD 更新完成中断使能 该位可被软件置 1 和清零。 0：禁用 SLCD 更新完成中断 1：使能 SLCD 更新完成中断
2	保留	必须保持复位值。
1	SOFIE	帧开始中断使能 该位可被软件置 1 和清零。 0：禁用 SLCD 帧开始中断 1：使能 SLCD 帧开始中断
0	HDEN	高驱动使能 该位可被软件置 1 和清零。 0：禁用持久的高驱动。R_L 使能的持续时间通过 PULSE[2:0]配置 1：使能持久的高驱动。R_L 总是被使能，PULSE[2:0]位无效

25.4.3. 状态标志寄存器（SLCD_STAT）

偏移地址：0x08
复位值：0x0000 0020

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										SYNF	VRDYF	UPDF	UPRF	SOF	ONF
										r	r	r	rs	r	r

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	SYNF	SLCD_CFG 寄存器同步标志 当 SLCD_CFG 寄存器更新到 SLCD 时钟域时，该位置 1。当写 SLCD_CFG 寄存器时，通过硬件清零该位。 0：SLCD_CFG 寄存器尚未同步 1：SLCD_CFG 寄存器已与 SLCD 时钟域同步
4	VRDYF	SLCD 电压就绪标志 该位根据 SLCD 电压由硬件置位或清零。 0：SLCD 电压未就绪 1：升压转换器已使能并准备提供准确电压
3	UPDF	更新 SLCD 数据完成标志 当更新完成 SLCD 数据时，该位通过硬件置位。通过向 SLCD_STATC 寄存器的 UPDC 位写 1 来清零该位。

		0：无影响 1：SLCD 数据更新完成
2	UPRF	更新 SLCD 数据请求标志 通过 SLCD_DATAx 寄存器组修改第一缓冲区后，应用程序应当置位该位将数据传输到第二缓冲区中。该位将保持置位直到传输完成，在这段时间 SLCD_DATAx 寄存器组为写保护状态。 0：无影响 1：请求 SLCD 数据更新
1	SOF	帧开始标志 在一个新帧开始时，该位通过硬件置 1。通过往 SLCD_STATC 寄存器中 SOFC 位写 1 来清零该位。 0：无影响 1：帧开始标志
0	ONF	SLCD 控制器开启标志 当 SLCDON 为 1 时，该位通过硬件置 1。在 SLCDON 位被清零并且最后的帧被显示后，该位通过硬件清零。 0：SLCD 控制器关闭 1：SLCD 控制器开启

25.4.4. 状态标志清除寄存器（SLCD_STATC）

偏移地址：0x0C
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												UPDC	保留	SOFC	保留
												rc_w1		rc_w1	

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	UPDC	SLCD 数据更新完成清除位 置 1 该位清除 SLCD_STAT 寄存器中的 UPDF 标志。 0：无影响 1：清除 UPDF 标志
2	保留	必须保持复位值。
1	SOFC	帧开始标志清除 置 1 该位清除 SLCD_STAT 寄存器中的 SOF 标志。

		0：无影响 1：清除 UPDF 标志
0	保留	必须保持复位值。

25.4.5. 显示数据寄存器（SLCD_DATAx, x=0~7）

偏移地址：0x14+0x08*x
复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

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

位/位域	名称	描述
31:0	SEG_DATAx[31:0]	每一位对应一个像素来显示。 0：该像素无效 1：该像素有效

26. 比较器 (CMP)

26.1. 简介

通用比较器(CMP0 和 CMP1)可独立工作，其输出端可用于 I/O 口，也可和定时器结合使用。

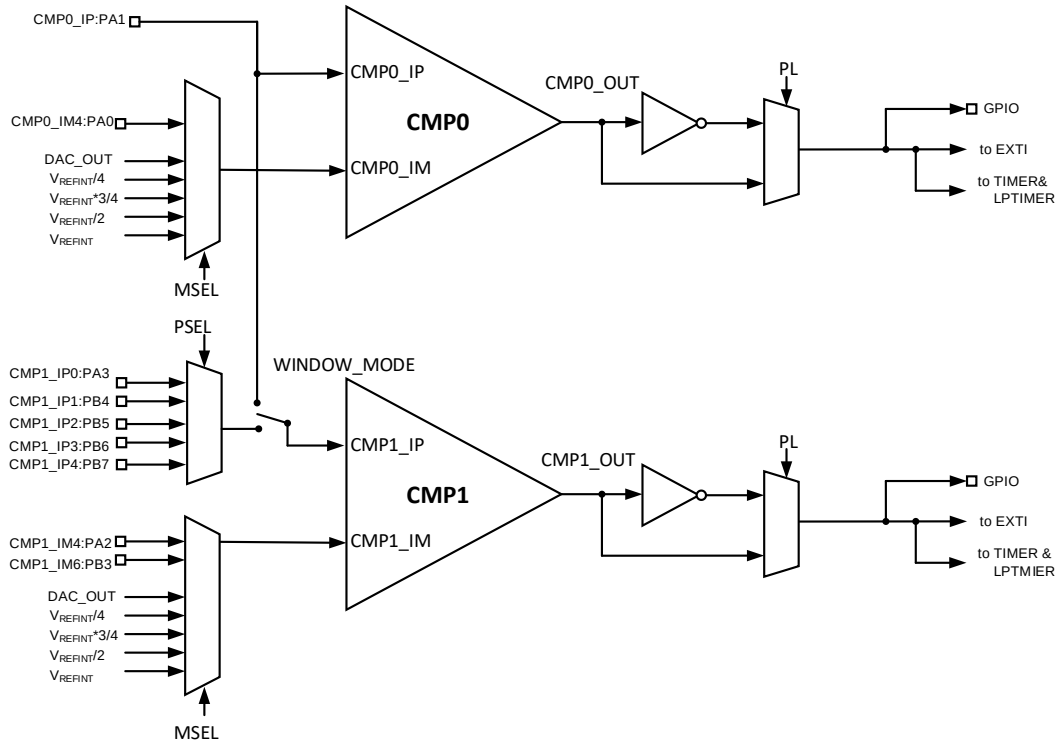
26.2. 主要特征

- 轨对轨比较器
- 迟滞可配置
- 速度、功耗可配置
- 比较器可配置以下模拟信号作为输入源
  - DAC 输出
  - 若干个 I/O 引脚
  - 0.25、0.5、0.75、1 倍的内部参考电压
- 窗口看门狗
- 作为消隐源
- 作为触发源输出到 IO 口或定时器
- 输出到 EXTI

26.3. 功能描述

比较器的框图展示如下：

图 26-1 比较器框图



注意：VREFINT 是 1.2V。

26.3.1. 比较器时钟和复位

比较器的时钟与 APB2 时钟同步。

26.3.2. 比较器的 I/O 配置

在被选为比较器输入端之前，相应管脚必须配置为模拟模式。

参考 Datasheet 的引脚定义，比较器输出必须连接到对应的复用 I/O 口。

比较器输出内部连接到定时器，他们的连接关系如下：

- CMP 输出连接到定时器输入通道。
- CMP 输出连接到定时器中止功能。
- CMP 输出连接到定时器 OCPRE_CLR。

为了在深度睡眠模式下工作，比较器端口的极性选择和输出重定向可独立于 APB2 工作。

比较器的输出可同时实现内部和外部输出。

比较器的输出端内连到外部中断和事件控制器，每个比较器都有其 EXTI 线，可用于产生中断或事件，同样的机制可运用于从省电模式中退出。

表 26-1 比较器输入与输出

	CMP0	CMP1
CMP 正向输入 IO 口	PA1	PA3 / PB4 / PB5

	CMP0	CMP1
		/ PB6 / PB7
CMP 反向输入 IO 口	PA0	PA2 / PB3
CMP 反向输入端所连接内部信号	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC_OUT	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT}, DAC_OUT
CMP 输出 I/O 口	PA0 / PA6 / PB0 / PB10 / PA11 / PB8	PA2 / PA7 /PB11 / PA12 / PB15 / PB9
CMP 输出端所连接内部信号	TIMER1_CH3, TIMER2_CH0, LPTIMER_CH0	TIMER1_CH3, TIMER2_CH0, LPTIMER_CH1

26.3.3. 比较器供电模式

对于给定的程序，在比较器功耗和传输迟滞之间存在着协定，可通过寄存器 CMPx_CS 的位 PM 的配置进行调整。当 PM 为 2'b 00 时，比较器以运行速度最快和功耗最大模式工作，但当 PM 位 2'b 11 时，比较器以运行速度最慢和功耗最小的模式工作。

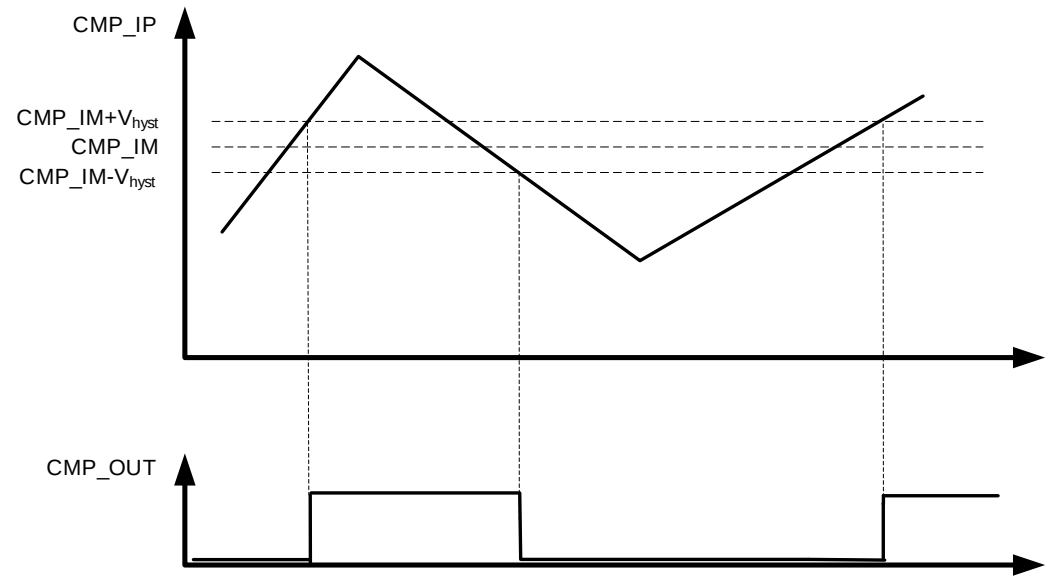
26.3.4. 比较器窗口模式

如果寄存器 CMP1_CS 的 WEN 位被置位，比较器的窗口模式被使能，比较器 1 的正向输入端即与比较器 0 的正向输入端相连。如果 CMP0 和 CMP1 的反向输入端连接不同的内部电压，可以通过分析 CMP0 和 CMP1 的输出结果监测输入电压的范围，该范围的上辨别阈和下辨别阈由反向输入端所连接的内部电压值决定。

26.3.5. 比较器迟滞

为了避免噪声信号所引起的假输出，比较器的迟滞可配置，该功能可以在用户不需要时关闭。

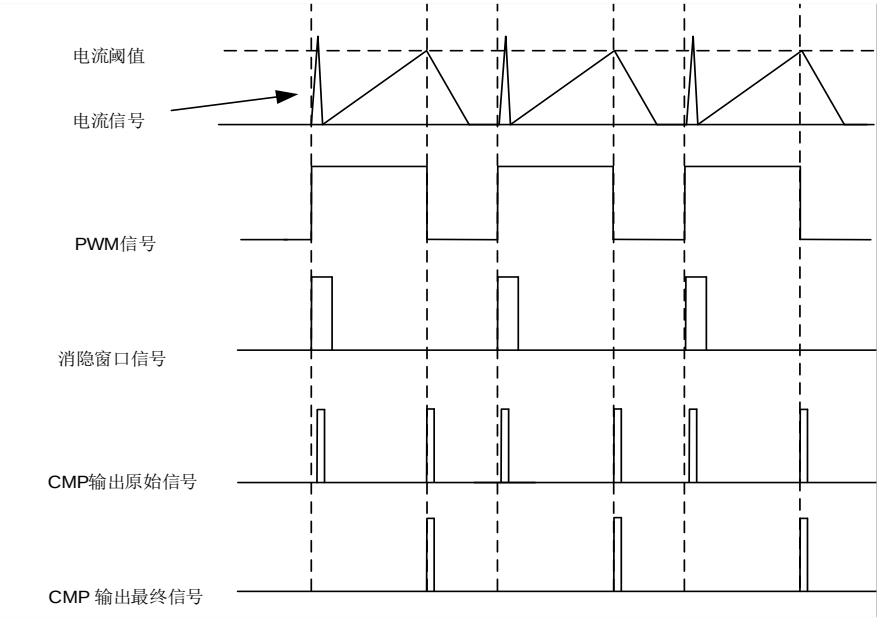
图 26-2.比较器迟滞



26.3.6. 比较器输出消隐

为了规避 PWM 周期启动时刻的电流脉冲的影响，计时器输出比较信号被选择为消隐源，消隐信号和比较器输出结果进行与运算，从而输出比较器预期结果。

图 26-3 比较器输出信号消隐功能



26.3.7. 比较器寄存器写保护

比较器的控制状态寄存器（CMPx_CS）可通过设置 LK 位为 1 来进行保护，CMPx_CS 寄存器，包含 LK 位，就会变为只读位，只有在 MCU 复位时才可以复位。

26.4. 比较器寄存器

CMP 基地址：0x4001 7C00

26.4.1. CMP0 控制状态寄存器(CMP0_CS)

地址偏移: 0x00
复位值: 0x0000 0000

该寄存器通过字访问(32 位)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	OUT	保留						SEN	BEN	保留	BLK[2:0]			HST1:0]	
rs	r							rw	rw		rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PL	OSEL[1:0]		保留						MSEL[2:0]			PM[1:0]		保留	EN
rw	rw								rw			rw			rw

位/位域	名称	描述
31	LK	CMP0 锁定 该位可将 CMP0 的各控制位设为只读，该位可写一次，通过系统复位清除，可通过软件置位 0：CMP0_CS[31:16]是可读可写位 1：CMP0_CS[31:16]是只读位
30	OUT	CMP0 输出 该位即 CMP0 输出状态，是只读位 0：正相输入端低于反相输入端，输出为低电平 1：正相输入端高于反相输入端，输出为高电平
29:24	保留	必须为保留值
23	SEN	电压标量使能位 该位可通过软件置位和清除，可使能 VREFINT 分频器的输出，被视为 CMP0 反向输入端 0：除能带隙标量 1：使能带隙标量
22	BEN	标量桥接使能位 0：在 CMP1_CS 寄存器 BEN 位为 0 的情景下，除能标量电阻桥接功能 1：使能标量电阻桥接功能
21	保留	必须为保留值
20:18	BLK[2:0]	CMP0消隐源选择位，选择合适的定时器输出来控制CMP0的输出消隐 000：没有消隐源 001：TIMER1 OC1作为消隐源 010：TIMER2 OC1作为消隐源

		100：TIMER8 OC1作为消隐源 101：TIMER11 OC1作为消隐源 其余值：保留
17:16	HST[1:0]	CMP0 迟滞 该域用于控制迟滞水平 00：无迟滞 01：低迟滞 10：中迟滞 11：高迟滞
15	PL	CMP0 输出极性 该位是用来控制输出极性 0：输出是正相的 1：输出是反相的
14:13	OSEL[1:0]	CMP0 输出选择 该位是用来选择 CMP0 输出目标 00：无配置 01：TIMER1 通道 3 输入捕获 10：TIMER2 通道 0 输入捕获 11：保留
12:7	保留	必须为保留值
6:4	MSEL[2:0]	CMP0_IM 输入选择 该位用来选择 CMP0 的输入端 CMP0_IM 的输入源 000：$V_{REFINT}/4$ 001：$V_{REFINT}/2$ 010：$V_{REFINT}*3/4$ 011：V_{REFINT} 100：PA0 101：DAC_OUT/PA4 其余值：保留
3:2	PM[1:0]	CMP0 功耗模式 00：高速/全功耗 01/10：中速/中功耗 11：低速/低功耗
1	保留	必须为保留值
0	EN	CMP0 使能 0：CMP0 除能 1：CMP0 使能

26.4.2. CMP1 控制状态寄存器(CMP1_CS)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器通过字访问(32 位)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	OUT	保留						SEN	BEN	保留	BLK[2:0]			HST1:0]	
rs	r							rw	rw		rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PL	OSEL[1:0]		保留		PSEL[2:0]			保留	MSEL[2:0]			PM[1:0]		WEN	EN
rw	rw				rw				rw			rw		rw	rw

位/位域	名称	描述
31	LK	CMP1 锁定 该位可将 CMP1 的各控制位设为只读，该位可写一次，通过系统复位清除，可通过软件置位 0：CMP1_CS[31:16]是可读可写位 1：CMP1_CS[31:16]是只读位
30	OUT	CMP1 输出 该位即 CMP1 输出状态，是只读位 0：正相输入端低于反相输入端，输出为低电平 1：正相输入端高于反相输入端，输出为高电平
29:24	保留	必须为保留值
23	SEN	电压标量使能位 该位可通过软件置位和清除，可使能 VREFINT 分频器的输出，被视为 CMP1 反向输入端 0：除能带隙标量 1：使能带隙标量
22	BEN	标量桥接使能位 0：在 CMP0_CS 寄存器 BEN 位为 0 的情景下，除能标量电阻桥接功能 1：使能标量电阻桥接功能
21	保留	必须为保留值
20:18	BLK[2:0]	CMP1消隐源选择位，选择合适的定时器输出来控制CMP1的输出消隐 000：没有消隐源 001：TIMER1 OC1作为消隐源 010：TIMER2 OC1作为消隐源 100：TIMER8 OC1作为消隐源 101：TIMER11 OC1作为消隐源 其余值：保留
17:16	HST[1:0]	CMP1 迟滞

		该域用于控制迟滞水平 00：无迟滞 01：低迟滞 10：中迟滞 11：高迟滞
15	PL	CMP1 输出极性 该位是用来控制输出极性 0：输出是正相的 1：输出是反相的
14:13	OSEL[1:0]	CMP1 输出选择 该位是用来选择 CMP1 输出目标 00：无配置 01：TIMER1 通道 3 输入捕获 10：TIMER2 通道 0 输入捕获 11：保留
12:11	保留	必须为保留值
10:8	PSEL[2:0]	CMP1_IP 选择位 000：PA3 001：PB4 010：PB5 011：PB6 100：PB7 其余值：保留
7	保留	必须为保留值
6:4	MSEL[2:0]	CMP1_IM 输入选择 该位用来选择 CMP1 的输入端 CMP1_IM 的输入源 000：$V_{REFINT}/4$ 001：$V_{REFINT}/2$ 010：$V_{REFINT}*3/4$ 011：V_{REFINT} 100：PA2 101：DAC_OUT/PA4 110：PB3 其余值：保留
3:2	PM[1:0]	CMP1 功耗模式 00：高速/全功耗 01/10：中速/中功耗 11：低速/低功耗
1	WEN	窗口模式使能位 0：CMP1 的正相输入端未与 CMP0 相连接

1：CMP1的正相输入端与CMP0的正相输入端连接

0	EN	CMP1 使能 0：CMP1 除能 1：CMP1 使能

27. 通用串行总线全速设备接口（USBD）

27.1. 概述

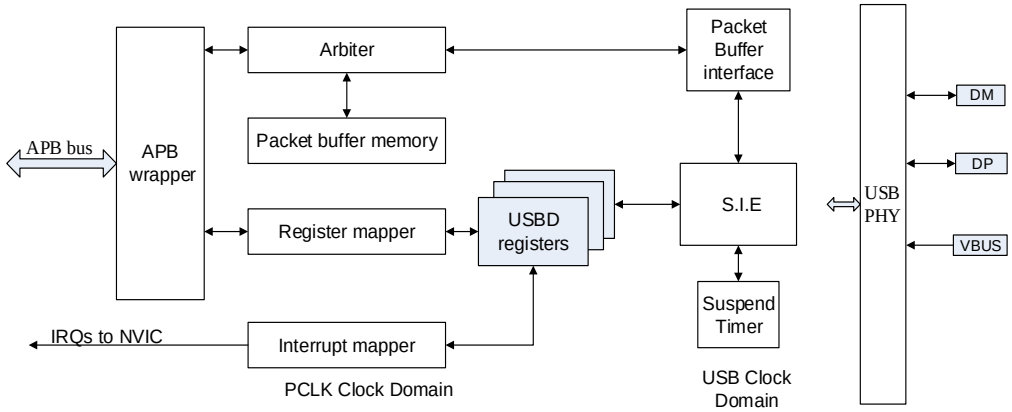
通用串行总线全速设备接口（USBD）模块提供了一个实现符合USB 2.0全速协议外设的方案。它内部包含了一个USB物理层而不需要额外的外部物理层芯片。USBD支持USB 2.0协议所定义的四种传输类型（控制、批量、中断和同步传输）。

27.2. 主要特征

- USB 2.0全速设备控制器；
- 最多支持8个可配置的端点；
- 支持双缓冲的批量传输端点/同步传输端点；
- 支持USB 2.0链接电源管理；
- 每个端点都支持控制，批量，同步或中断传输（端点0除外，端点0只支持控制传输）；
- 支持USB挂起/恢复操作；
- 拥有用于数据缓冲的512字节专用SRAM；
- 集成的USB物理层；
- USBD连接/断开功能。

27.3. 模块图

图 27-1. USBD 模块图



27.4. 信号描述

表 27-1. USBD 信号描述

输入/输出端口	类型	描述
VBUS	输入	总线电源端口
DM	输入/输出	差分数据线 D-端口
DP	输入/输出	差分数据线 D+端口

注意：一旦USBD被使能，这些引脚会自动连到USBD内部收发器上。

27.5. 时钟配置

根据USB标准定义，USB全速模块采用了固定的48MHz时钟。要使用USBD，需要打开两个时钟，一个是USB控制器时钟，它的频率必须配到48MHz，另一个是APB1到USB接口时钟，它也是APB1的总线时钟，其频率可以高于也可以低于48MHz。

注意：为了满足USB数据传输率和分组缓冲区接口的系统需求，APB1总线时钟的频率必须大于24MHz，以避免数据缓冲区的上下溢出。

USB控制器的48MHz时钟可以通过MCU的内部晶振或者外部晶振分频后再经过PLL倍频得到：

- 16MHz的内部晶振2分频后作为PLL的输入，再进行6倍频得到
- 8MHz的外部晶振直接作为PLL的输入，然后进行6倍的倍频，得到所需的USB时钟

注意：无论使用外部晶振还是内部晶振产生的USB时钟，其时钟准确度都必须要达到±500 ppm。如果USB时钟的准确度下降，传输数据可能无法满足USB规范要求，甚至直接导致USB无法运行。

27.6. 功能说明

27.6.1. USB 端点

USBD支持8个可以独立配置的USB端点。

每个端点支持：

- 单或双缓冲（端点0不能使用双缓冲）；
- 一个端点缓冲区描述符；
- 可编程的缓冲区起始地址与长度；
- 对于数据包的响应可配置；
- 控制传输（仅端点0）。

端点缓冲区

设备操作的功能就是将存储映像中的请求发到USB总线上或是从USB总线上接收请求存储到存储映像中。为了有效地管理USB端点通信，USBD实现了一个可被USB外设直接访问的512

字节专用SRAM数据包缓冲区。它被映射到APB1外设存储区，从地址0x4000 6000到0x4000 6400。总容量为1KB，但是由于总线宽度原因USBD实际只使用了512字节。

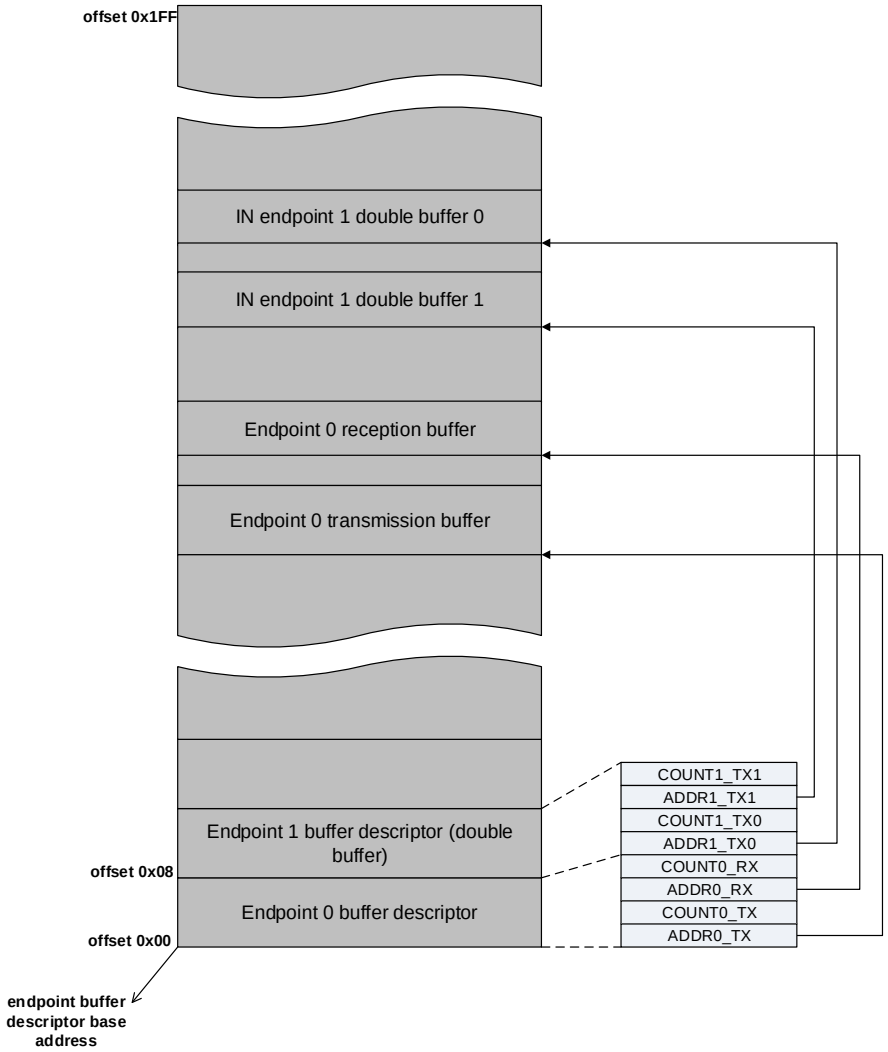
每个端点都有一个或者两个用于存储当前数据负载的数据包缓冲区。双向端点通常有两个缓冲区，一个用于数据发送，另一个用于数据接收。单向端点只用一个缓冲区用于数据操作。

端点缓冲器描述符表

USBD实现了一个定义了端点的缓冲区地址与长度的端点缓冲区描述符表，其也位于端点数据包缓冲区中。相应的端点缓冲区描述符作为了一个可以在应用程序固件与系统存储器中SIE之间的通讯端口。由于每个端点方向要求两个16位字的缓冲区描述。因此，每个表条目包含了4个16位字（发送与接收两个方向）并按照8字节对齐。当一个端点为双缓冲端点，SIE将以ping-pong的方式使用这两个缓冲区。USBD端点缓冲地址寄存器指向端点缓冲区描述表。

*图27-2*描述了端点缓冲区描述表与数据包缓冲区之间的关系

图 27-2. 缓冲描述符表的用法示例（USBD_BADDR = 0）



注意：此图是未按照实际大小来绘的，而以USB总线16位模式来定址。

双缓冲端点

双缓冲特性是为了提高批量传输的性能。为了实现新的流控方案，USBD应该要知道哪一个数据包缓冲区正在被应用软件使用，从而避免发生冲突。既然在USBD_EPxCS寄存器中有两个数据翻转位，而USBD只使用一位来进行硬件数据处理（由于双缓冲功能所需的单向约束），那么，应用程序可以使用另外一位来表明当前正在使用哪个缓冲区。这个新的缓冲区标志位称作SW_BUF。*表27-2.* **双缓冲标志定义**解释了USBD_EPxCS寄存器位与DTG/SW_BUF定义之间的对应关系。

表 27-2. 双缓冲标志定义

缓冲标志	发送端点	接收端点
DTG	TX_DTG（USBD_EPxCS 第 6 位）	RX_DTG（USBD_EPxCS 第 14 位）
SW_BUF	RX_DTG（USBD_EPxCS 第 14 位）	TX_DTG（USBD_EPxCS 第 6 位）

DTG位和SW_BUF位负责流控。当一个传输完成的时候，USB外设翻转DTG位；当数据被复制后，应用软件需要翻转SW_BUF位。除了首次传输，如果DTG位的值等于SW_BUF位的值，传输将会暂停，并且向主机发送NAK数据包。当这两位不相等的时候，传输会继续。

表 27-3. 双缓冲的用法

端点类型	DTG	SW_BUF	USBD 使用的数据包缓冲	应用程序使用的数据包缓冲
OUT	0	1	EPxRBADDR / EPxRBCNT 缓冲区描述符列表地址	EPxTBADDR / EPxTBCNT 缓冲区描述符列表地址.
	1	0	EPxTBADDR / EPxTBCNT 缓冲区描述符列表地址	EPxRBADDR / EPxRBCNT 缓冲区描述符列表地址
IN	0	1	EPxTBADDR / EPxTBCNT 缓冲区描述符列表地址	EPxRBADDR / EPxRBCNT 缓冲区描述符列表地址
	1	0	EPxRBADDR / EPxRBCNT 缓冲区描述符列表地址	EPxTBADDR / EPxTBCNT 缓冲区描述符列表地址

端点存储请求仲裁

由于USB外设通过APB1接口连到APB1总线上，所以APB1接口接收来自于APB1总线的存储请求与来自于USB接口的端点存储请求。它通过给APB1总线更高的优先权来解决冲突，并且总是保留一半的存储器带宽供USBD完成传输。它采用时分复用的策略实现了虚拟的双端口SRAM，即在USB传输的同时，允许应用程序访问存储器。此策略也允许任意长度的多字节APB1传输。

27.6.2. USB 传输

USB 事务处理

在端点配置后并且事务被请求，硬件将会检测令牌包。当USBD收到令牌包后，将执行数据传输。数据传输完成后，根据传输方向USBD将产生相应的握手包发送出去或期望主机发送相应的握手包。

事务处理完成后，一个端点相关的中断将被触发。在中断处理例程中，应用程序将作相应的处

理。

事务格式化是硬件完成的，包括CRC的产生与校验。

一旦端点被使能。端点控制和状态寄存器、端点缓冲区地址和传输长度都不能被应用程序修改。当正确传输中断通知数据传输操作完成时，它们就可以被再次访问，从而使能一次新的操作。

IN 事务

当一个有效的已配置端点收到一个IN令牌包，它将会发送数据包给主机。如果端点无效，根据当前的端点状态，将发送NAK或STALL握手包。

在数据包传输过程中：首先将发送一个配置好的数据PID，然后端点缓冲区中的实际数据会加载到输出移位寄存器中发送出去，在数据发送后硬件会发送计算好的CRC。

当收到来自主机发送的ACK包，USB外设将翻转数据PID，设置当前的端点状态为NAK。同时，正确传输中断将被触发。在中断服务例程中，应用程序将数据包存储器填满数据后通过设置端点状态为VALID再次使能端点从而开始下一次传输。

OUT 和 SETUP 事务

USBD在处理这两类令牌包时基本使用相同的方式，处理SETUP包的不同点会在下面关于控制传输的部分详细叙述。

在接收端点配置好并且使能后，主机将发送OUT/SETUP令牌给设备。当USBD接收到令牌包后，将会访问端点缓冲区描述符来初始化端点缓冲区地址与长度。然后接收的数据将被陆续打包成字（LSB模式）传输到端点缓冲区。当检测到DATA包传输结束时，计算的CRC的值将和接收到的CRC值进行比较。如果没有错误发生，将向主机发送一个ACK握手包。

当事务正确完成时，USBD将翻转数据PID并设置端点状态为NAK。然后硬件将触发端点正确传输中断。在中断服务例程中，应用程序可以判断事务类型并从端点缓冲区中读出接收数据。在接收数据被处理后，应用程序通过设置端点状态为VALID来发起下一次事务。

如果接收期间出现了任何错误，USBD设置错误中断位并且继续将数据复制到报文缓冲区，但不发送ACK包。USBD自己能够从接收错误中恢复并且继续处理下一个数据传输。USBD的访问从不超出数据包缓冲区，这是通过内部寄存器的配置来控制的。接收到的2字节的CRC也会复制到数据包缓冲区，紧跟在数据字节之后。如果数据的长度比实际分配的长度大，超出的数据不被复制。这种情况称为缓冲过载。此时将发送STALL握手包，当前会话失败。

如果一个被寻址的端点是无效的，将按照当前端点状态发送NAK或STALL握手包而不是ACK，并且不向接收缓冲区写入数据。

控制传输

控制传输要求主机从一个到设备的SETUP事务开始，此事务描述设备应当执行的访问控制类型。SETUP事务后面跟着零个或者若干个携带被请求访问特定信息的数据会话。最后，一个状态事务完成控制传输并且允许端点返回控制传输的状态到客户端软件。在状态会话之后，控制传输完成，主机可以操作该端点的下一个控制传输。

USBD总是使用双向的端点0作为默认的控制端点来处理控制传输。USBD通过解析SETUP事

务的内容获取其关心的数据量和传输方向，并且要求将未使用的方向置为STALL，除了最后一个数据阶段。

在最后一个数据阶段，应用软件将控制端点相反方向的端点0状态设置为NAK。这将使主机等待控制操作的完成。如果操作成功完成，软件将会把NAK改成VALID，否则改成STALL。如果状态阶段是一个OUT事务，STATUS_OUT位将会被置位，这样携带非零数据的状态会话将会被应答STALL以表明已发生了错误。

根据USB协议，设备不需要废弃当前命令，再开启新命令，因此，设备必须用ACK握手包回复SETUP包，而不是NAK或STALL握手包。

当已配置的控制端点接受到SETUP令牌包，USBD接收这个数据，执行被要求的数据传输并且发送回一个ACK握手包。如果关于此前的数据传输请求未被成功处理，USB丢弃SETUP命令包，视当前情况为错误情况，然后促使主机重新发送请求令牌包。

同步传输

同步传输可以保证固定的传输速率以及固定的延迟，但当总线发生错误时并不支持数据重发。因此，同步协议没有一个握手阶段，在数据包发送之后没有ACK包。它也不支持数据翻转，DATA0 PID仅仅被用来开始一个数据包的发送。

同步端点的状态只能被设定为DISABLED和VALID，其他的任何值是非法的。应用软件可以实现双缓冲以提高性能。通过在每个会话时交换发送和接收包缓冲，应用软件可以将数据复制进缓冲或复制出缓冲，同时USB外设能在另一个缓冲区中处理数据的发送或接收。通过查询DTG位即可知道USB外设现在正在使用哪一个缓冲区。

应用软件按照要用的首个缓冲区去初始化DTG。在每一个事务的结尾，RX_ST或TX_ST位被置位，这取决于使能方向，而忽略CRC错误或缓冲过载情况（如果错误发生，ERRIF位将被置位）。同时，USB外设将会翻转DTG位，但是不影响STAT位。

27.6.3. USB 事件与中断

每一个USB行为都通过应用程序初始化，由USB中断或事件来驱动。在系统复位后，应用程序需要等待一系列的USB中断和事件。

复位事件

系统和上电复位

一旦系统或上电复位，应用程序首先要提供USB模块与接口所需的所有时钟，然后清除复位信号以访问该模块的寄存器，最后打开和USB收发器相连的模拟部分。

USB固件需要做以下工作：

- 复位USBD_CTL寄存器中的CLOSE位；
- 等待内部参考电压稳定；
- 清除USBD_CTL寄存器的SETRST位；
- 清除USBD_INTF寄存器以移除冗余的挂起中断，然后使能其他单元。

USB复位（复位中断）

当这个事件发生时，USB外设的状态同系统复位后状态是一样的。

USB固件需要做以下工作：

- 在10ms内设定USBD_DADDR寄存器的USBEN位来使能USB模块；
- 初始化USBD_EP0CS寄存器和它相关的数据包缓冲。

挂起和恢复事件

在任何需要的情况下，通过写控制寄存器（USBD_CTL）总可以强制使USB模块置于低功耗模式（SUSPEND模式）。此时，不产生任何静态电流消耗，同时USB时钟也会减慢或停止。通过对USB线上数据传输的检测，可以在低功耗模式下唤醒USB模块。

USB协议一直强调由USB从设备进行电源管理。如果设备从总线获得电源（总线供电设备）的话，这一点变得尤其重要。总线供电设备必须满足下述限制：

- 处于非配置状态的从设备，从USB总线最多获取100mA的电流；
- 已配置的设备只能按照配置描述符的MaxPower位域的设置获得电流且最大值不超过500 mA；
- 已挂起设备最多获取500uA电流。

假如在USB总线上没有活动超过3ms，设备将进入挂起状态。如果有来自主机的唤醒信号，将唤醒一个挂起的设备。

USB外设还支持软件初始化的远程唤醒。为了启动远程唤醒功能，应用软件在MCU唤醒后必须使能所有的时钟，并清除挂起位。这将导致硬件产生一个远程唤醒信号的上行数据流。

将SETSPS位设为1，即可使能挂起模式并且禁用对于SOF接收的检查。将LOWM位设为1将关闭USB模拟收发器的静态功耗，但是此时仍能检测到恢复信号。

链接电源管理（LPM）等级 L1

为了优化在挂起/恢复状态下的电源消耗，USB 2.0实现了链接电源管理（LPM）。LPM包括从L0到L3共4种状态。LPM L1状态（睡眠状态）是新的电源管理状态。

如果主机给设备发送了一个成功的LPM事务，设备将进入L1状态。L1不会对于连接的设备强加任何具体的电源需求（来自VBUS）。

具体内容请参考文档USB2_LinkPowerManagement_ECN。

USB 中断

USB控制器有三个中断线：低优先级中断，高优先级中断和唤醒中断。软件可以配置这些中断以将特定的中断条件连接到位于NVIC表中的这些中断信号上。如果中断状态位和对应的中断使能位都被置位，硬件将会产生一个中断。如果中断条件产生，中断状态位都将被硬件置位（不管中断使能位是否设置）。

- USB低优先级中断（通道20）：可被所有USB事件触发；
- USB高优先级中断（通道19）：只能被同步和双缓冲批量传输的正确传输事件触发；
- USB唤醒中断（通道42）：可被所有的唤醒事件触发。

27.6.4. 操作指南

此部分主要描述USBD的操作指南。

USBD 寄存器初始化过程

1. 清除USBD_CTL寄存器的CLOSE位，然后清除SETRST位；
2. 清除USBD_INTF寄存器来移除冗余的挂起中断；
3. 编程USBD_BADDR寄存器来设定端点缓冲区基地址；
4. 设定USBD_CTL来使能中断；
5. 等待复位中断（RSTIF）；
6. 在复位中断中初始化控制端点0来发起枚举过程，然后编程USBD_BADDR来设定设备地址为0并使能USB模块功能；
7. 配置端点0来接收SETUP包。

USBD 端点初始化过程

1. 编程USBD_EPxTBADDR/USBD_EPxRBADDR寄存器来设定发送或者接收数据缓冲区地址；
2. 根据端点的用处编程USBD_EPxCS寄存器的EP_CTL和EP_KCTL位来设定端点类型和缓冲区类型；
3. 如果端点是单缓冲端点：

   1） 编程USBD_EPxCS寄存器的TX_DTG或者RX_DTG位来初始化端点的数据翻转位，但是端点0需要将这两位分别设置为1和0来进行控制传输；

   2） 编程USBD_EPxCS寄存器的TX_STA或者RX_STA位来配置寄存器的状态，但是如果使用端点0来发起控制传输，则这两位都要配置为NAK。

   如果端点是双缓冲端点：

   1） 发送与接收数据翻转位都需要编程。如果端点是发送端点，清除USBD_EPxCS寄存器中的TX_DTG和RX_DTG位，否则如果是接收端点，需要翻转TX_DTG位；

   2） 编程USBD_EPxTBCNT和USBD_EPxRBCNT寄存器来设定传输数据字节数；

   3） 端点发送与接收状态都需要配置。如果端点是发送端点，设定TX_STA位为NAK和RX_STA位为DISABLED，否则如果是接收端点，RX_STA位设定为VALID，TX_STA位设定为DISABLED。

SETUP 和 OUT 数据传输

1. 编程USBD_EPxRBCNT寄存器来设定BLKSIZ和EPRCNT域，这些域定义了端点缓冲区长度；

2. 通过编程USBD_EPxCS寄存器配置端点寄存器状态为VALID来使能端点；
3. 等待正确传输中断（STIF）；
4. 在中断处理例程中，应用程序通过读取USBD_EPxCS寄存器的SETUP位可以决定事务的类型。然后应用程序从端点数据缓冲区中USBD_EPxRBAR寄存器定义的起始地址处读取数据负载。最后应用程序会解析这些数据并进行相应的处理。

IN 数据传输

1. 编程USBD_EPxTBCNT寄存器来设定EPTCNT域，此域定义了端点缓冲区长度；
2. 通过编程USBD_EPxCS寄存器配置端点寄存器状态为VALID来使能端点去发送数据；
3. 等待正确传输中断（STIF）；
4. 在中断处理例程中，应用程序需要更新用户缓冲区长度与位置指针。然后应用程序使用用户缓冲区的数据填充端点缓冲区。最后应用程序将配置端点状态为VALID来进行下一次传输。

27.7. USBD 寄存器

USBD 基地址：0x4000 5C00

27.7.1. USBD 控制寄存器（USBD_CTL）

地址偏移：0x40
复位值：0x0003

该寄存器可半字(16 位)或全字(32 位)访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIE	PMOUIE	ERRIE	WKUPIE	SPSIE	RSTIE	SOFIE	ESOFIE	L1REQIE	保留	L1RSREQ	RSREQ	SETSPS	LOWM	CLOSE	SETRST
rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位/位域	名称	描述
15	STIE	成功传输中断使能 0：禁用成功传输中断 1：当USBD_INTF寄存器的STIF位被置位，产生中断
14	PMOUIE	包缓冲上溢/下溢中断使能 0：当包缓冲上溢/下溢不产生中断 1：当USBD_INTF寄存器的PMOUIF位被置位，产生中断.
13	ERRIE	错误中断使能 0：禁用错误中断 1：当USBD_INTF寄存器的ERRIF位被置位，产生中断
12	WKUPIE	唤醒中断使能 0：禁用唤醒中断 1：当USB_IFR寄存器的WKUPIF位被置位，产生中断
11	SPSIE	挂起状态中断使能 0：禁用挂起状态中断 1：当USBD_INTF寄存器的SPSIF位被置位，产生中断
10	RSTIE	USB复位中断使能 0：禁用USB复位中断 1：当USBD_INTF寄存器的RSTIF位被置位，产生中断
9	SOFIE	帧起始中断使能 0：禁用帧起始中断 1：当USBD_INTF寄存器的SOFIF位被置位，产生中断

8	ESOFIE	预期的帧起始中断使能 0：禁用预期的帧起始中断 1：当USBD_INTF寄存器的ESOFIF位被置位，产生中断
7	L1REQIE	LPM L1状态请求中断使能 0：禁用LPM L1状态请求中断 1：当USBD_INTF寄存器的L1REQ位被置位，产生中断
6	保留	必须保持复位值。
5	L1RSREQ	LPM L1恢复请求 MCU可以设置此位来发送一个LPM L1恢复信号给主机。在信号发送过程结束后，硬件会清除此位。
4	RSREQ	恢复请求 软件向USB主机设置一个中断请求，USB主机应该按USB规范驱动这个恢复序列 0：没有恢复请求 1：发送恢复请求
3	SETSPS	设置挂起 当USBD_INTF寄存器的SPSIF位被置位时，软件应该设置挂起状态 0：没有设置挂起状态 1：设置挂起状态
2	LOWM	低功耗状态 当置位这一位时，USB在挂起状态进入低功耗模式。如果从挂起状态恢复，硬件会复位这一位 0：无影响 1：在挂起模式进入低功耗模式
1	CLOSE	关闭状态 当这一位被置位的时候，USBD进入关闭状态，并且完全关闭USBD，同主机断开 0：不在关断状态 1：在关断状态
0	SETRST	设定复位 当这位置位，USBD外设应该被复位 0：无影响 1：发生复位

27.7.2. USBD 中断标志寄存器（USBD_INTF）

地址偏移：0x44

复位值：0x0000

该寄存器可半字（16 位）或全字（32 位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIF	PMOUIF	ERRIF	WKUPIF	SPSIF	RSTIF	SOFIF	ESOFIF	L1REQ	保留		DIR	EPNUM[3:0]			
r	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0			r	r			

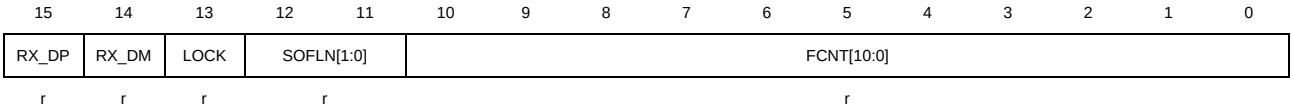
位/位域	名称	描述
15	STIF	成功传输中断标志 当一个会话成功完成时，硬件置位该位
14	PMOUIF	包缓冲溢出/下溢中断标志 硬件置位该位表示包缓冲区存储不下所有所传输的数据。软件写0清该位
13	ERRIF	错误中断标志 当在会话期间有错误发生时，硬件置位该位。软件写0清该位
12	WKUPIF	唤醒中断标志 在SUSPEND状态下，当总线上有活动被检测到时，硬件置位该位。 软件写0清该位
11	SPSIF	挂起状态中断标志 当USB总线无任何活动超过3ms时，硬件置位该位，表明有SUSPEND请求。软件写0清该位
10	RSTIF	USB复位中断标志 当检测到USB RESET信号时硬件置位该位。软件写0清该位
9	SOFIF	帧起始中断标志 一个新的SOF包到达时硬件置位该位。软件写0清该位
8	ESOFIF	预期的帧起始中断标志 硬件置位表示一个SOF被预期但是还没有到达。软件写0清该位
7	L1REQ	当LPM L1事务被正确地接受和响应后，硬件会置位此位。软件写0清该位。
7:5	保留	必须保持复位值。
4	DIR	会话传输方向 硬件置位表示会话的传输方向 0：IN 类型 1：OUT 类型
3:0	EPNUM[3:0]	端点号 硬件置位确认当前会话所关联的端点

27.7.3. USBD 状态寄存器（USBD_STAT）

地址偏移：0x48

复位值：0x0XXX 这里X是未定义的

该寄存器可半字（16 位）或全字（32 位）访问



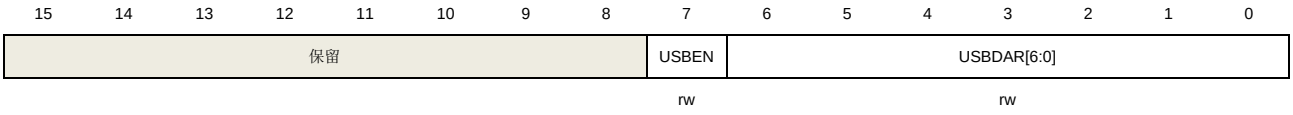
位/位域	名称	描述
15	RX_DP	接收数据 + 线状态 代表DP线的状态
14	RX_DM	接收数据 - 线状态 代表DM线的状态
13	LOCK	锁定USB 硬件置位表明接收到了至少两个连续SOF包
12:11	SOFLN[1:0]	丢失SOF 当每次发生ESOFIF事件时，硬件递增此位，一旦再次接收到SOF则清除该位
10:0	FCNT[10:0]	帧编号计数器 每次收到SOF，帧编号计数器将会增加

27.7.4. USBD 设备地址寄存器（USBD_ADDR）

地址偏移：0x4C

复位值：0x0000

该寄存器可半字（16 位）或全字（32 位）访问



位/位域	名称	描述
15:8	保留	必须保持复位值。
7	USBEN	USB设备使能 通过软件设置该位使能USB设备 0：USB设备禁用。没有会话要处理 1：USB设备使能
6:0	USBDAR[6:0]	USBD设备地址 总线复位之后，地址被复位为0x00。若USB使能位被置位，则从设备会响应功能地址DEV_ADDR的报文。

27.7.5. USBD 缓冲器地址寄存器（USBD_BADDR）

地址偏移：0x50

复位值：0x0000

该寄存器可半字（16 位）或全字（32 位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BAR[12:0]													保留		
rw															

位/位域	名称	描述
15:3	BAR[12:0]	缓冲器地址 所分配缓冲器(512byte on-chip SRAM)的起始地址，用来保存缓冲描述符表以及包缓冲
2:0	保留	必须保持复位值。

27.7.6. USBD 端点 x 控制/状态寄存器（USB_EPxCS），x=[0..7]

地址偏移：0x00 to 0x1C
复位值：0x0000

该寄存器可半字（16 位）或全字（32 位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_ST	RX_DTG	RX_STA[1:0]		SETUP	EP_CTL[1:0]		EP_KCTL	TX_ST	TX_DTG	TX_STA[1:0]		EP_ADDR[3:0]			
rc_w0	t	t		r	rw		rw	rc_w0	t	t		rw			

位/位域	名称	描述
15	RX_ST	正确接收 当一个成功的OUT/SETUP会话完成时，硬件置位此位 通过软件写0清该位
14	RX_DTG	接收数据PID翻转位 本标志位代表非同步端点的翻转数据位（0=DATA0，1=DATA1） 用来实现双缓冲端点的流控功能 用于同步端点的缓冲区交换
13:12	RX_STA[1:0]	接收状态位 通过软件写1翻转 写0保持不变 参考下表
11	SETUP	Setup会话完成 当一个SETUP会话完成时，硬件置位此位
10:9	EP_CTL[1:0]	端点类型控制l 参考下表
8	EP_KCTL	端点类别控制 其具体含义取决于端点类型的设置 参考下表
7	TX_ST	正确发送

		当一个IN会话成功完成时，硬件置位此位 软件清0
6	TX_DTG	发送数据PID翻转位 本标志位代表非同步端点的翻转数据位（0=DATA0，1=DATA1） 用来实现双缓冲端点的流控功能 用于同步端点的缓冲区交换
5:4	TX_STA[1:0]	发送状态位 参考下表
3:0	EP_ADDR	端点地址 用来指示会话的目标端点

表 27-1 接收状态编码

RX_STA[1:0]	含义
00	**DISABLED:**忽略此端点的所有接收请求
01	**STALL:**握手状态为 STALL
10	**NAK:**握手状态为 NAK
11	**VALID:**使能端点的接收

表 27-2. 端点类型编码

EP_CTL[1:0]	含义
00	**BULK:**批量端点
01	**CONTROL:**控制端点
10	**ISO:**同步端点
11	**INTERRUPT:**中断端点

表 27-3. 端点类别编码

EP_KCTL[1:0]		含义
00	BULK	DBL_BUF
01	CONTROL	STATUS_OUT

表 27-4. 发送状态编码

TX_STA[1:0]	含义
00	**DISABLED:**忽略端点的所有发送请求
01	**STALL:**握手包状态为 STALL
10	**NAK:**握手包状态为 NAK
11	**VALID:**使能端点的发送

27.7.7. USBD 端点 x 发送缓冲地址寄存器（USBD_EPxTBADDR），x=[0…7]

地址偏移：[USBD_BADDR] + x * 16
USB本地地址：[USBD_BADDR] + x * 8

该寄存器可半字（16 位）或全字（32 位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EPTXBAR[15:1]	EPTXBAR[0]
rw	rw

位/位域	名称	描述
15:1	EPTXBAR[15:1]	发送缓冲地址 在收到下一个IN分组时，需要发送的数据所在的缓冲区起始地址
0	EPTXBAR[0]	必须设为0

27.7.8. USBD 端点 x 发送缓冲区字节数目寄存器（USBD_EPxTBCNT），x=[0…7]

地址偏移：[USBD_BADDR] + x * 16 + 4
USB本地地址：[USBD_BADDR] + x * 8 + 2

该寄存器可半字（16 位）或全字（32 位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						EPTXCNT[9:0]									
						rw									

位/位域	名称	描述
15:10	保留	必须保持复位值。
9:0	EPTXCNT[9:0]	发送字节数 在收到下一个IN令牌后，将发送的字节数

27.7.9. USBD 端点 x 接收缓冲器地址寄存器（USBD_EPxRBADDR），x=[0…7]

地址偏移：[USBD_BADDR] + x * 16 + 8
USB本地地址：[USBD_BADDR] + x * 8 + 4

该寄存器可半字（16 位）或全字（32 位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPRBAR[15:1]															EPRBAR[0]
rw															rw

位/位域	名称	描述
15:1	EPRBAR[15:1]	接收缓冲器地址 收到下一个OUT或者SETUP分组时，用于保存数据的缓冲区起始地址。
0	EPRBAR[0]	必须设为0

27.7.10. USBD 端点 x 接收缓冲区字节数目寄存器 n（USBD_EPxRBCNT），x=[0…7]

地址偏移：[USBD_BADDR] + x * 16 + 12
USB本地地址：[USBD_BADDR] + x * 8 + 6

该寄存器可半字（16 位）或全字（32 位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLKSIZ	BLKNUM[4:0]					EPRCNT[9:0]									
rw	rw					r									

位/位域	名称	描述
15	BLKSIZ	块的大小 0：块大小是2字节 1：块大小是32字节
14:10	BLKNUM[4:0]	块数目 包缓冲区所分配的块的数目
9:0	EPRCNT[9:0]	接收字节数 在收到下一个OUT/SETUP令牌后，接收到数据的字节数

27.7.11. USBD LPM 控制和状态寄存器（USBD_LPMCS）

地址偏移：0x54
复位值：0x0000

该寄存器可半字（16 位）或全字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												CHIRD			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RHIRD				BLSTAT[3:0]				REMWK	保留	LPMACK	LPMEN
				r				r				r		rw	rw

位/位域	名称	描述
31:20	保留	必须保持复位值。
19:16	CHIRD	配置的HIRD值
15:12	保留	必须保持复位值。
11:8	RHIRD	接收的HIRD值
7:4	BLSTAT[3:0]	bLinkState值 此位域包含最后一个LPM令牌包被确认后产生的bLinkState值

3	REMWK	bRemoteWake值 此位域包含最后一个LPM令牌包被确认后产生的bRemoteWake值
2	保留	必须保持复位值。
1	LPMACK	LPM令牌包响应使能 0：有效的LPM令牌包将响应NYET 1：有效的LPM令牌包将响应ACK NYET/ACK仅仅在LPM事务成功后才被返回： EXT令牌与LPM令牌都没有错误（否则错误） 等于0001B（L1）的有效bLinkState被接收（否则STALL）
0	LPMEN	LPM支持使能 软件设置此位来使能USB设备的LPM支持。如果此位为0，将不会有LPM事务被处理

27.7.12. USBD DP 上拉控制寄存器（USBD_DPC）

地址偏移：0x58
复位值：0x0000

该寄存器可半字（16位）或全字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPUEN	保留														
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	DPUEN	DP 上拉控制 0：禁用 DP 线上的内部上拉，断开与主机的连接 1：在 DP 线上启用内部上拉，连接到主机
14:0	保留	必须保持复位值。

28. 附录

28.1. 寄存器表中使用的缩写列表

表 28-1. 寄存器功能位访问属性

功能位访问属性	描述
读/写(rw)	软件可以对这个位进行读写。
只读(r)	软件只能对这个位进行读。
只写(w)	软件只能对这个位进行写。读取该位将返回复位值。
读/写 1 清零(rc_w1)	软件可以读该位，对该位写入 1 可以清除这个位。写入 0 对位值没有影响。
读/写 0 清零(rc_w0)	软件可以读该位，对该位写入 0 可以清除这个位。写入 1 对位值没有影响。
翻转(t)	软件可以通过写 1 来翻转该位。写入 0 对位值没有效果。
只读/写 1 触发(rt_w1)	软件可以读该位，写入 1 触发事件，但对位值没有影响。

28.2. 术语表

表 28-2. 术语

术语	描述
字	32 位长度数据
半字	16 位长度数据.
字节	8 位长度数据
IAP(应用内编程)	IAP 是在用户程序运行时对微控制器的闪存重新编程的能力。
ICP(在线编程)	ICP 是当设备安装在用户应用板上时，一个使用 JTAG 协议，SWD 协议或引导加载程序的微控制器的闪存编程能力。
选项字节	存储在闪存中的产品配置位
AHB	高级高性能总线
APB	高级外设总线
RAZ	读为 0
WI	写忽略
RAZ/WI	读为 0/写忽略

28.3. 可用外设

对于各个 MCU 系列的外设及其数量，请参考相应型号的数据手册。

29. 版本历史

表 29-1. 版本历史

版本号	描述	日期
1.0	初稿发布	2021 年 6 月 22 日
1.1	1. 修改章节*片上闪存* 2. 删除*主存储闪存块快速编程*章节注意 2 中的“使用 eflash 宏中的 BUS，不是 CBUFPBUF/cache” 3. 在*段码 LCD 控制器*中添加 *VSLCD 电压源配置*小节	2021 年 12 月 9 日
1.2	1. 修改 DAC 章节*控制寄存器 DDUDRIE 欠载中断使能从 0 修改为 1*。 2. *电源管理（PMU）*章节部分功能优化表述。 3. *内部集成电路总线接口（I2C）*章节部分功能优化表述。 4. *比较器（CMP）*章节部分功能语言优化表述。 5. 调试（DBG）中删除 *JTAG 接口*描述。 6. 复位和时钟单元（RCU）中 RCU_CFG1 寄存器 bit[3:0] 描述修改为 *PLL 输入源分频因子*，RCU_RSTSCK 寄存器 bit23 修改为 *V11RSTF* 7. 加密处理器（CAU）中修改*图 23 3. CAU 框图*。	2022 年 6 月 29 日
1.3	1. 在*复位和时钟单元（RCU）*章节中更改 HXTAL 范围为 4-48MHz。 2. 修改 *25.3.7. VSLCD 电压源*中关于 SLCD 电压源配置的描述	2022 年 12 月 14 日
1.4	1. 在复位和时钟单元 RCU 章节，添加关于 SLCD 部分描述。	2023 年 6 月 21 日

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.