GigaDevice Semiconductor Inc.

GD32VW55x
RISC-V 32-bit MCU

适用于 **GD32VW553xx**

用户手册

1.4 版本

(2025 年 8 月)

目录

图索引

表索引

1. 系统及存储器架构

GD32VW55x 系列器件是基于 Nuclei N307 处理器的 32 位通用微控制器，其中 N307 处理器是基于 RSIC-V 架构指令集开发而来，以下简称 RISC-V 处理器。RISC-V 处理器包括两条 AHB 总线分别称为 I-Cache 总线和系统总线。RISC-V 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哈佛结构，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. RISC-V 处理器

RISC-V 处理器适用于低能耗、小面积的嵌入式应用，具有简单的动态分支预测、指令预取缓冲区和 I-cache 等多种高效微架构特点。访问 http://user.nucleisys.com 以获取更多关于 N307 内核相关信息。它支持 64 个通用寄存器（GPRs）：

- 3级管道，采用最先进的处理器微架构，以提供高性能、高效率和低成本；
- 支持机器（M）和用户（U）权限级别；
- 支持不可屏蔽中断（NMI）；
- 支持动态分支预测；
- 可配置指令预取逻辑，进行后续两条指令的预取，隐藏指令内存访问延时；
- 支持WFI（等待中断）和WFE（等待事件）机制进入睡眠模式；
- 中断优先级可配置/可编程；
- 适用于实时性能的增强矢量中断处理；
- 支持中断优先抢占；
- 支持咬尾中断；
- 标准4线JTAG调试端口和2线cJTAG调试端口；
- 支持交互式调试功能；
- 支持8个硬件断点触发器；
- 支持RV32I / M / A / F / D / C / P / B扩展指令；
- 支持两级睡眠模式：浅睡眠模式，和深度睡眠模式；
- 支持64位宽实时计数器（可用作系统滴答计数器）；
- 支持物理存储器保护（PMP），用于保护物理存储，8个区域；
- 支持指令缓存，2路关联，缓存行单元大小为32字节，共32KB；
- 支持单/双精度浮点运算（FPU）；
- 支持2周期浮点MAC；
- 支持packed-SIMD DSP指令。

1.2. 系统架构

GD32VW55x 系列器件采用 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、三个 AHB 总线和两个 APB 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*中，”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

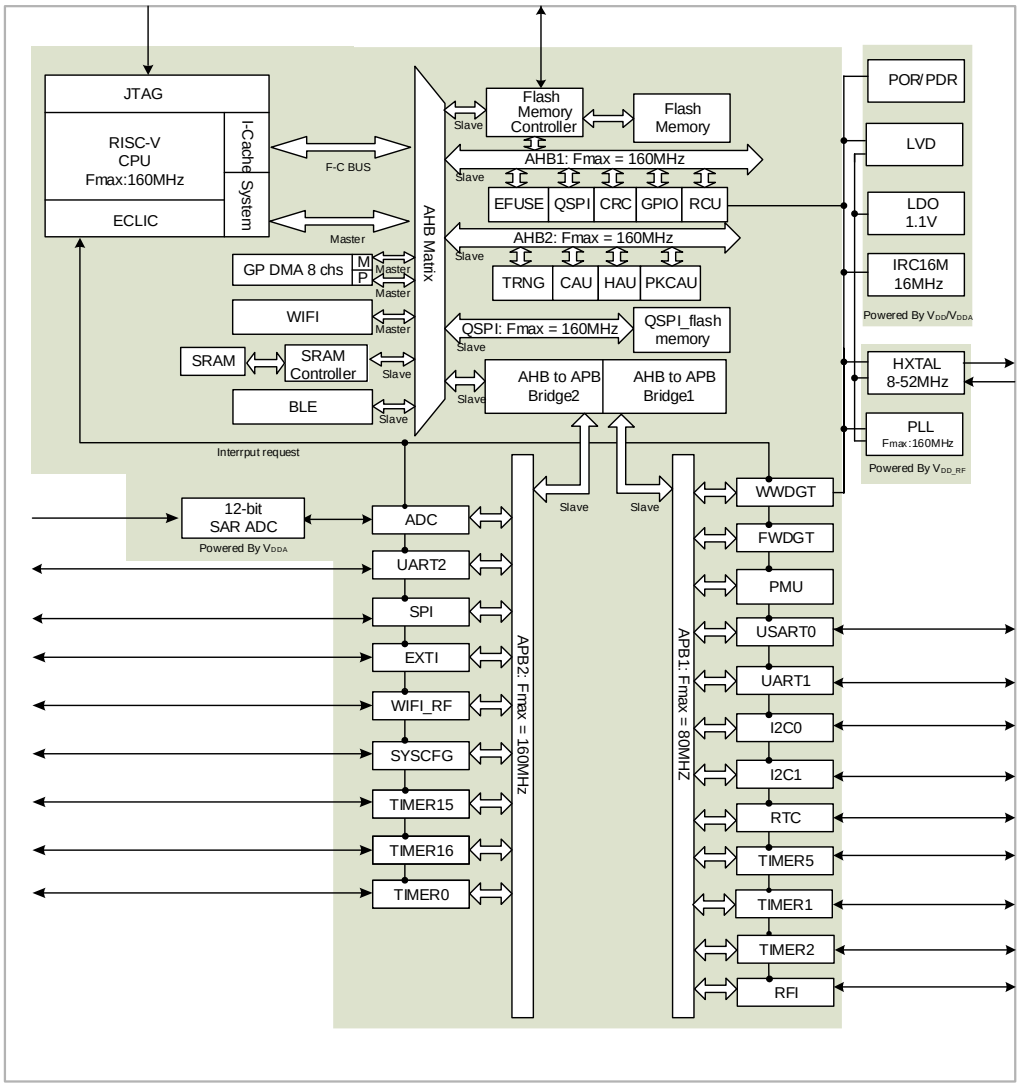
	F-C BUS	S BUS	DMAM	WIFI	DMAP
FMC	1	1	1	0	1
SRAM0	1	1	1	1	1
AHB1	0	1	1	0	1
QSPI	1	1	1	1	1
AHB2	0	1	1	0	1
SRAM1	1	1	1	1	1
SRAM2	1	1	1	1	1
SRAM3	1	1	1	1	1
APB1	0	1	1	0	1
APB2	0	1	1	0	1
BLE	1	1	1	1	1

如*表 1-1. AHB 互联矩阵的互联关系列表*所示，AHB 互联矩阵共连接多个主机，分别为：F-CBUS、SBUS、DMAM、DMAP、WIFI。F-CBUS 是 RISC-V 内核的代码总线。F-CBUS 用于在 cache 打开时对映射到代码区域的内部存储器进行指令获取，F-CBUS 的目标是内部 Flash、外部存储器（QSPI_flash）、BLE 和和内部 SRAMs（SRAM0，SRAM1，SRAM2 和 SRAM3）。类似的，SBUS 是 RISC-V 内核的系统总线，用于指令和向量获取、数据存储和加载以及系统区域的调试访问。系统区域包括内部 SRAM 区域、外部存储器（QSPI_flash）、BLE 和外设区域。DMAM 是 DMA 的存储器总线，DMA 使用 DMAM 来完成对内存的传输，该总线的目标是内部 Flash、内部 SRAMs、外部存储器（QSPI_flash）、BLE 和外设区域。DMAP 是 DMA 的外设总线，DMA 使用 DMAP 访问 AHB 外设或完成内存到内存传输。该总线的目标是 AHB 和 APB 的外设以及数据存储器：内部 Flash、内部 SRAMs（SRAM0, SRAM1, SRAM2 和 SRAM3）、外部存储器（QSPI_flash）和 BLE。WIFI 总线将 WIFI 的 AHB 主机接口连接到总线矩阵，该总线的目标是 SRAMs（SRAM0, SRAM1, SRAM2 和 SRAM3）外部存储器（QSPI_flash）和 BLE。

AHB 互联矩阵也连接了一些从机，分别为：FMC，SRAM0，SRAM1，SRAM2，SRAM3，AHB1，AHB2，APB1，APB2，QSPI 和 BLE。FMC 是闪存控制器的总线接口。SRAM0~SRAM3 是片上静态随机存取存储器。AHB1 是连接所有 AHB1 从机的 AHB 总线。AHB2 是连接 AHB2 从机的的 AHB 总线。QSPI 是 QSPI_flash 内存控制器的总线接口。BLE 是 BLE 的总线接口。APB1 和 APB2 是连接所有 APB 从机的两条 APB 总线。APB1 速度最高为 80MHz，APB2 速度最高为 160MHz。

这些是使用多层 AHB 总线架构互连的，如*图 1-1. GD32VW55x 系列器件的系统架构示意图*所示。

图 1-1. GD32VW55x 系列器件的系统架构示意图



1.3. 存储器映射

RISC-V 处理器采用哈弗架构，可以使用单独的总线来提取指令和加载/存储数据。指令代码和数据都位于相同的存储器地址空间，但在不同的地址范围。程序存储器，数据存储器，寄存器和 I/O 端口组织在同一线性 4GB 地址空间内。这是 RISC-V 的最大地址范围，因为总线地址宽度为 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 RISC-V 处理器提供的规则预先定义的。在存储器映射表中，一部分地址空间由 RISC-V 的系统外设所占用，且不可更改。此外，其余部分地址空间可由芯片供应商定义使用。*表 1-2. GD32VW55x 系列器件的存储器映射表*显示了 GD32VW55x 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. GD32VW55x 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
外部设备	QSPI	0x9800 0000 – 0xD0FF FFFF	保留
		0x9000 0000 - 0x97FF FFFF	QSPI_FLASH（MEM）
		0x7000 0000 - 0x8FFF FFFF	保留
		0x6000 0000 - 0x67FF FFFF	保留
外设	AHB2	0x4C06 3000 - 0x4FFF FFFF	保留
		0x4C06 1000 - 0x4C06 2FFF	PKCAU
		0x4C06 0C00 - 0x4C06 0FFF	保留
		0x4C06 0800 - 0x4C06 0BFF	TRNG
		0x4C06 0400 - 0x4C06 07FF	HAU
		0x4C06 0000 - 0x4C06 03FF	CAU
		0x4C05 0400 - 0x4C05 FFFF	保留
		0x4C05 0000 - 0x4C05 03FF	保留
		0x4C04 0000 - 0x4C04 FFFF	保留
		0x4C00 0000 - 0x4C03 FFFF	保留
	AHB1	0x4904 0000 - 0x4BFF FFFF	保留
		0x4900 0000 - 0x4903 FFFF	保留
		0x400B 1000 - 0x48FF FFFF	保留
		0x400B 0800 - 0x400B 0FFF	保留
		0x400B 0400 - 0x400B 07FF	保留
		0x400B 0000 - 0x400B 03FF	保留
		0x400A 1000 - 0x400A FFFF	保留
		0x400A 0C00 - 0x400A 0FFF	保留
		0x400A 0800 - 0x400A 0BFF	保留
		0x400A 0400 - 0x400A 07FF	保留
		0x400A 0000 - 0x400A 03FF	保留
		0x4008 0400 - 0x4009 FFFF	保留
		0x4008 0000 - 0x4008 03FF	保留
		0x4003 0000 - 0x4007 FFFF	WIFI
		0x4002 BC00 - 0x4002 FFFF	保留
		0x4002 B000 - 0x4002 BBFF	保留
		0x4002 A000 - 0x4002 AFFF	保留
		0x4002 8000 - 0x4002 9FFF	保留
		0x4002 6800 - 0x4002 7FFF	保留
		0x4002 6400 - 0x4002 67FF	保留
		0x4002 6000 - 0x4002 63FF	DMA
		0x4002 5C00 - 0x4002 5FFF	保留
		0x4002 5800 - 0x4002 5BFF	QSPI_FLASH（REG）
		0x4002 5400 - 0x4002 57FF	保留
		0x4002 5000 - 0x4002 53FF	保留

预先定义的地址空间	总线	地址范围	外设
		0x4002 4000 - 0x4002 4FFF	保留
		0x4002 3C00 - 0x4002 3FFF	保留
		0x4002 3800 - 0x4002 3BFF	RCU
		0x4002 3400 - 0x4002 37FF	保留
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2C00 - 0x4002 2FFF	保留
		0x4002 2800 - 0x4002 2BFF	EFUSE
		0x4002 2400 - 0x4002 27FF	保留
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1C00 - 0x4002 1FFF	保留
		0x4002 1800 - 0x4002 1BFF	保留
		0x4002 1400 - 0x4002 17FF	保留
		0x4002 1000 - 0x4002 13FF	保留
		0x4002 0C00 - 0x4002 0FFF	保留
		0x4002 0800 - 0x4002 0BFF	GPIOC
		0x4002 0400 - 0x4002 07FF	GPIOB
		0x4002 0000 - 0x4002 03FF	GPIOA
	APB2	0x4001 8800 - 0x4001 FFFF	保留
		0x4001 8400 - 0x4001 87FF	TIMER16
		0x4001 8000 - 0x4001 83FF	TIMER15
		0x4001 7C00 - 0x4001 7FFF	保留
		0x4001 7800 - 0x4001 7BFF	WIFI_RF
		0x4001 6800 - 0x4001 77FF	保留
		0x4001 6000 - 0x4001 67FF	保留
		0x4001 5800 - 0x4001 5FFF	保留
		0x4001 5400 - 0x4001 57FF	保留
		0x4001 4C00 - 0x4001 53FF	保留
		0x4001 4800 - 0x4001 4BFF	保留
		0x4001 4400 - 0x4001 47FF	保留
		0x4001 4000 - 0x4001 43FF	保留
		0x4001 3C00 - 0x4001 3FFF	EXTI
		0x4001 3800 - 0x4001 3BFF	SYSCFG
		0x4001 3400 - 0x4001 37FF	保留
		0x4001 3000 - 0x4001 33FF	SPI
		0x4001 2C00 - 0x4001 2FFF	保留
		0x4001 2400 - 0x4001 2BFF	保留
		0x4001 2000 - 0x4001 23FF	ADC
		0x4001 1400 - 0x4001 1FFF	保留
		0x4001 1000 - 0x4001 13FF	UART2
		0x4001 0800 - 0x4001 0FFF	保留

预先定义的地址空间	总线	地址范围	外设
		0x4001 0400 - 0x4001 07FF	保留
		0x4001 0000 - 0x4001 03FF	TIMER0
	APB1	0x4000 D000 - 0x4000 FFFF	保留
		0x4000 CC00 - 0x4000 CFFF	RFI
		0x4000 7400 - 0x4000 CBFF	保留
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6C00 - 0x4000 6FFF	保留
		0x4000 5C00 - 0x4000 6BFF	保留
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 4C00 - 0x4000 53FF	保留
		0x4000 4800 - 0x4000 4BFF	USART0
		0x4000 4400 - 0x4000 47FF	UART1
		0x4000 4000 - 0x4000 43FF	保留
		0x4000 3C00 - 0x4000 3FFF	保留
		0x4000 3800 - 0x4000 3BFF	保留
		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	保留
		0x4000 2000 - 0x4000 23FF	保留
		0x4000 1C00 - 0x4000 1FFF	保留
		0x4000 1800 - 0x4000 1BFF	保留
		0x4000 1400 - 0x4000 17FF	保留
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0C00 - 0x4000 0FFF	保留
		0x4000 0800 - 0x4000 0BFF	保留
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM	AHB	0x2101 0000 - 0x3FFF FFFF	保留
		0x2100 0000 - 0x2100 FFFF	BLE
		0x2005 0000 - 0x20FF FFFF	保留
		0x2003 0000 - 0x2004 FFFF	SRAM3（96KB + 共享 32KB）
		0x2002 0000 - 0x2002 FFFF	SRAM2（64KB）
		0x2001 0000 - 0x2001 FFFF	SRAM1（64KB）
		0x2000 0000 - 0x2000 FFFF	SRAM0（64KB）
Code	AHB	0x1000 0000 - 0x1FFF FFFF	外部存储器映射
		0x0FFC 0100 - 0x0FFF FFFF	保留
		0x0FFC 0000 - 0x0FFC 00FF	EFUSE（256 字节）

预先定义的地址空间	总线	地址范围	外设
		0x0BF8 0000 –0x0FFB FFFF	保留
		0x0BF4 0000 - 0x0BF7 FFFF	ROM（256KB）
		0x0A07 0000 - 0x0BF3 FFFF	保留
		0x0A04 0000 - 0x0A06 FFFF	保留
		0x0A02 0000 - 0x0A03 FFFF	保留
		0x0A01 0000 - 0x0A01 FFFF	保留
		0x0A00 0000 - 0x0A00 FFFF	保留
		0x0840 0000 - 0x09FF FFFF	保留
		0x0800 0000 - 0x083F FFFF	主存储
		0x0000 0000 - 0x07FF FFFF	外部存储器映射

1.3.1. 片上 SRAM 存储器

GD32VW55x 系列微控制器含有高达 288KB+共享 32KB 的片上 SRAM（SRAM0 64KB、SRAM1 64KB、SRAM2 64KB、SRAM3 96KB + 共享 32KB），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 4096KB 的片上闪存。闪存包括主存储块，分为 1024 页，每页的容量为 4KB，和 256KB 容量的用于存储引导装载程序（boot loader）的信息块。

详见*闪存控制器（FMC）*。

1.4. 引导配置

启动时，使用 BOOT0 和 BOOT1 引脚选择引导存储器地址。BOOT0 和 BOOT1 值由*表 1-3. BOOT0 模式*和*表 1-4. BOOT1 模式*配置决定。

- BOOT0值可以来自BOOT0引脚，也可以来自EFUSE_CTL0寄存器中SWBOOT0位的值，以便在需要时释放GPIO引脚。
- BOOT1值可以来自PB1引脚，也可以来自EFUSE_CTL0寄存器中SWBOOT1位的值，以便在需要时释放GPIO引脚。

表 1-3. BOOT0 模式

SWBOOT0	EFBOOT0	BOOT0 PC8 引脚	BOOT0
0	-	0	0
0	-	1	1
1	0	-	0
1	1	-	1

表 1-4. BOOT1 模式

SWBOOT1	EFBOOT1	BOOT1 PB1 引脚	BOOT1
0	-	0	0
0	-	1	1
1	0	-	0
1	1	-	1

引导地址参考*表 1-5. 引导模式*。

当 BOOT0 值为 0 时：

- 引导地址由EFUSE_CTL0寄存器的EFSB位来选择。

当 BOOT0 值为 1 时：

- 当EFUSE_CTL0寄存器的EFBOOTLK位为0时，引导地址由BOOT1值来选择。

表 1-5. 引导模式

EFBOOTLK	BOOT0	BOOT1	EFSB	引导地址	引导区域
-	0	-	0	0x08000000	SIP Flash
-	0	-	1	0x0BF46000	secure boot
0	1	0	-	0x0BF40000	Bootloader / ROM
0	1	1	-	0x20000000	SRAM
1	1	-	-	0x0BF40000	Bootloader / ROM

复位释放时，BOOTx（x = 0/1）的值（来自引脚或 EFBOOTx 位）被锁存。用户可以设置 BOOTx 值来选择所需的引导模式。从待机模式退出时，也会对 BOOTx 引脚或 EFBOOTx 位（取决于 EFUSE_CTL0 寄存器中 EFBOOTLK 和 SWBOOTx 位的值）进行重新采样。因此，它们必须在待机模式下保持所需的引导模式配置。启动延迟后，在释放处理器复位之前完成了引导区域的选择。

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新编程。该引导装载程序可通过以下串行接口之一工作：USART0（PB15 和 PA8），UART1（PA4 和 PA5），UART2（PA6 和 PA7）。

1.5. 系统配置寄存器（SYSCFG）

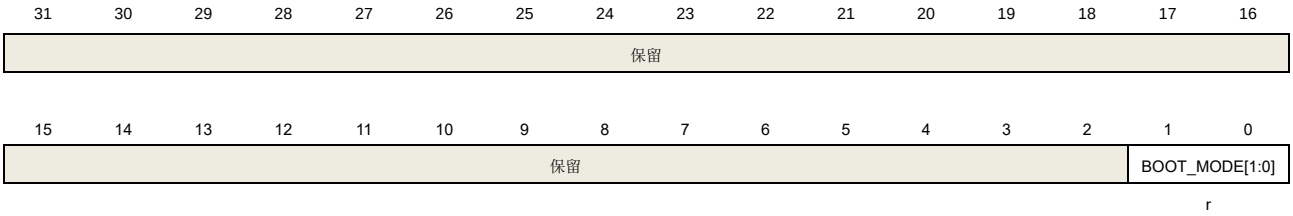
SYSCFG 基地址：0x4001 3800

1.5.1. 配置寄存器 0（SYSCFG_CFG0）

地址偏移：0x00

复位值：0x0000 000X（根据 BOOT0 和 BOOT1 引脚的状态，X 表示 BOOT_MODE[1:0]，可能为任意值）

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。
1:0	BOOT_MODE[1:0]	引导模式（详细请参考*引导配置*） 位0映射到BOOT0引脚，位1的值与BOOT1_n选项位的值相反。 x0：从主存储引导启动 01：从系统存储启动 11：从片上SRAM引导启动

1.5.2. 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。

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 0000：PA3引脚

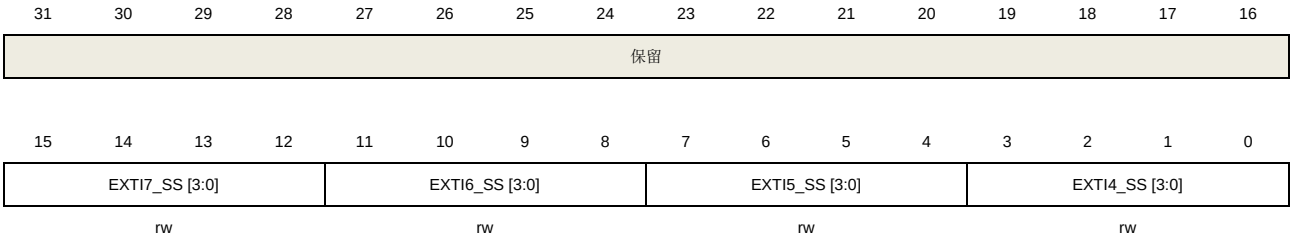
0001：PB3引脚
其他配置保留。

11:8	EXTI2_SS[3:0]	EXTI 2 源选择 0000：PA2引脚 0001：PB2引脚 其他配置保留。
7:4	EXTI1_SS[3:0]	EXTI 1 源选择 0000：PA1引脚 0001：PB1引脚 其他配置保留。
3:0	EXTI0_SS[3:0]	EXTI 0 源选择 0000：PA0引脚 0001：PB0引脚 其他配置保留。

1.5.3. 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。



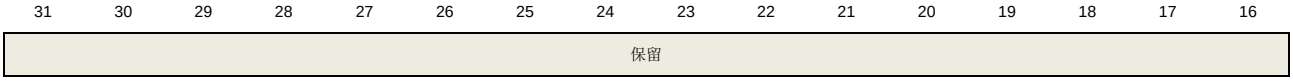
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI7_SS[3:0]	EXTI 7 源选择 0000：PA7引脚 其他配置保留。
11:8	EXTI6_SS[3:0]	EXTI 6 源选择 0000：PA6引脚 其他配置保留。
7:4	EXTI5_SS[3:0]	EXTI 5 源选择 0000：PA5引脚 其他配置保留。
3:0	EXTI4_SS[3:0]	EXTI 4 源选择

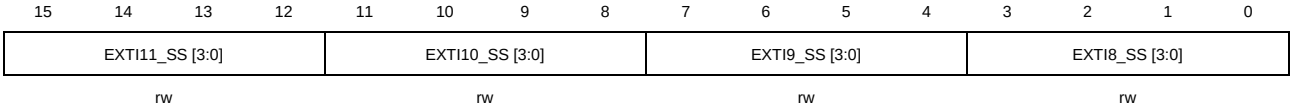
0000：PA4引脚
0001：PB4引脚
其他配置保留。

1.5.4. 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。



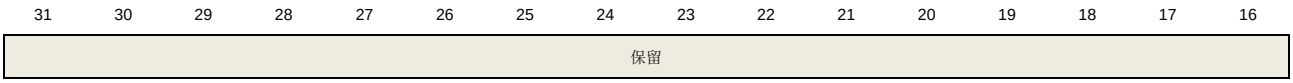


位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI11_SS[3:0]	EXTI 11 源选择 0000：PA11引脚 0001：PB11引脚 其他配置保留。
11:8	EXTI10_SS[3:0]	EXTI 10 源选择 0000：PA10引脚 其他配置保留。
7:4	EXTI9_SS[3:0]	EXTI 9 源选择 0000：PA9引脚 其他配置保留。
3:0	EXTI8_SS[3:0]	EXTI 8 源选择 0000：PA8引脚 0001：PC8引脚 其他配置保留。

1.5.5. 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。



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15_SS [3:0]				EXTI14_SS [3:0]				EXTI13_SS [3:0]				EXTI12_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	EXTI15_SS[3:0]	EXTI 15 源选择 0000：PA15引脚 0001：PB15引脚 0010：PC15引脚 其他配置保留。
11:8	EXTI14_SS[3:0]	EXTI 14 源选择 0000：PA14引脚 0001：PC14引脚 其他配置保留。
7:4	EXTI13_SS[3:0]	EXTI 13 源选择 0000：PA13引脚 0001：PB13引脚 0010：PC13引脚 其他配置保留。
3:0	EXTI12_SS[3:0]	EXTI 12 源选择 0000：PA12引脚 0001：PB12引脚 其他配置保留。

1.5.6. I/O 补偿控制寄存器（SYSCFG_CPSCTL）

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							CPS_RDY	保留							CPS_EN
							r								rw

位/位域	名称	描述
31:9	保留	必须保持复位值。

8	CPS_RDY	I/O补偿单元是否准备好 0：I/O补偿单元未准备就绪 1：I/O补偿单元准备就绪
7:1	保留	必须保持复位值。
0	CPS_EN	I/O补偿单元使能 0：I/O补偿单元掉电 1：I/O补偿单元使能

1.5.7. SYSCFG 配置寄存器 1（SYSCFG_CFG1）

地址偏移：0x54

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

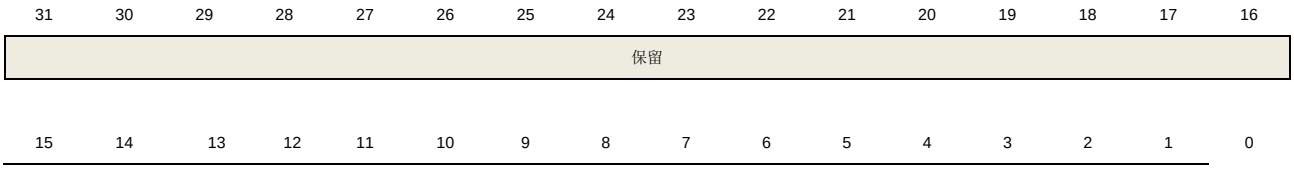
位/位域	名称	描述
31:3	保留	必须保持复位值。
2	LVD_LOCK	LVD 锁定 该位由软件置 1，只能通过系统复位清 0。可用于将 LVD 连接锁定到 TIMER0/15/16 的 break 输入端。 0：LVD 中断从 TIMER0/15/16 的 break 输入端断开。 1：LVD 中断与 TIMER0/TIMER15/TIMER16 的 break 输入端连接。
1:0	保留	必须保持复位值。

1.5.8. SYSCFG 共享 SRAM 配置寄存器（SYSCFG_SCFG）

地址偏移：0x68

复位值：0x0000 0001

该寄存器只能按字（32 位）访问。



保留	SOWNSEL
	rw

位/位域	名称	描述
31:1	保留	必须保持复位值。
0	SOWNSEL	共享 SRAM 所有权选择 该位用于共享的 32KB SRAM 的所有权控制。 0：无线 1：处理器

1.5.9. TIMER 触发选择寄存器（SYSCFG_TIMERxCFG）（x = 0..2）

地址偏移：0x100 + 0x4 * x
复位值：0x0000 0000

TSCFG0[3:0]，TSCFG1[3:0]..TSCFG6[3:0]之间相互互斥，不能同时配置。

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
TSCFG7[3:0]				TSCFG6[3:0]				TSCFG5[3:0]				TSCFG4[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSCFG3[3:0]				TSCFG2[3:0]				TSCFG1[3:0]				TSCFG0[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	TSCFG7[3:0]	ITS 触发源配置 ITS 触发源配置，包括通道输入信号（CIx）和内部触发输入（ITIx）。 0000：保留 0001：内部触发输入 0（ITI0） 0010：内部触发输入 1（ITI1） 0011：内部触发输入 2（ITI2） 0100：内部触发输入 3（ITI3） 0101：CI0 的边沿标志位（CI0F_ED） 其他：保留 在从模式使能时，该位域不能被改写。 注意：使用 TSCFG7[3:0]时，需保证 TSCFGy[3:0]（y=0..6）为零，否则 ITS 触发源需与 TSCFGy[3:0]选择的通道输入源保持一致。
27:24	TSCFG6[3:0]	外部时钟模式 0 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。 0000：外部时钟模式 0 禁能 0001：内部触发输入 0（ITI0）

0010：内部触发输入 1（ITI1）
0011：内部触发输入 2（ITI2）
0100：内部触发输入 3（ITI3）
0101：CI0 的边沿标志位 CI0F_ED）
0110：滤波后的通道 0 输入（CI0FE0）
0111：滤波后的通道 1 输入（CI1FE1）
1000：滤波后的外部触发输入（ETIFP）
其他：保留
在从模式使能时，该位域不能被改写。

23:20	TSCFG5[3:0]	事件模式 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。 0000：事件模式禁能 0001：内部触发输入 0（ITI0） 0010：内部触发输入 1（ITI1） 0011：内部触发输入 2（ITI2） 0100：内部触发输入 3（ITI3） 0101：CI0 的边沿标志位（CI0F_ED） 0110：滤波后的通道 0 输入（CI0FE0） 0111：滤波后的通道 1 输入（CI1FE1） 1000：滤波后的外部触发输入（ETIFP） 其他：保留 在从模式使能时，该位域不能被改写。
19:16	TSCFG4[3:0]	暂停模式 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。 0000：暂停模式禁能 0001：内部触发输入 0（ITI0） 0010：内部触发输入 1（ITI1） 0011：内部触发输入 2（ITI2） 0100：内部触发输入 3（ITI3） 0101：保留 0110：滤波后的通道 0 输入（CI0FE0） 0111：滤波后的通道 1 输入（CI1FE1） 1000：滤波后的外部触发输入（ETIFP） 其他：保留 在从模式使能时，该位域不能被改写。
15:12	TSCFG3[3:0]	复位模式 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。 0000：复位模式禁能 0001：内部触发输入 0（ITI0） 0010：内部触发输入 1（ITI1） 0011：内部触发输入 2（ITI2） 0100：内部触发输入 3（ITI3）

		0101：CI0 的边沿标志位（CI0F_ED） 0110：滤波后的通道 0 输入（CI0FE0） 0111：滤波后的通道 1 输入（CI1FE1） 1000：滤波后的外部触发输入（ETIFP） 其他：保留 在从模式使能时，该位域不能被改写。
11:8	TSCFG2[3:0]	编码器模式 2 0000：编码器模式 2 禁能 其他：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 在从模式使能时，该位域不能被改写。
7:4	TSCFG1[3:0]	编码器模式 1 0000：编码器模式 1 禁能 其他：根据 CI0FE0 的电平，计数器在 CI1FE1 的边沿向上/下计数 在从模式使能时，该位域不能被改写。
3:0	TSCFG0[3:0]	编码器模式 0 0000：编码器模式 0 禁能 其他：根据 CI1FE1 的电平，计数器在 CI0FE0 的边沿向上/下计数 在从模式使能时，该位域不能被改写。

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，包括页擦除，整片擦除，以及编程操作。

2.2. 主要特征

- 两种闪存组织：
  - 主存储块：最大4MB（典型值：2MB）的片上闪存用于指令和数据的存储
  - 信息块：256KB bootloader区域
- 片上闪存页大小为4KB；
- 支持RTDEC（通过AES即时加密算法加密）功能；
- 支持32位整字编程，页擦除和整片擦除操作；
- 闪存安全保护，可防止非法代码/数据访问；
- 页擦除和编程保护，可阻止意外写操作。

2.3. 功能说明

2.3.1. 闪存结构

闪存包括 4MB（最大值）主闪存，分为 1024 页，页大小为 4KB，和 256KB 用于引导加载程序的信息块。每页都可以单独擦除。*表 2-1. 闪存的基地址和构成*显示了闪存组织的详细信息。

表 2-1. 闪存的基地址和构成

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1022页	0x083F E000 - 0x083F EFFF	4KB
	第1023页	0x083F F000 - 0x083F FFFF	4KB
引导装载程序	正常引导区域	0x0BF4 0000 - 0x0BF4 5FFF	24KB
	安全引导区域	0x0BF4 6000 - 0x0BF4 DFFF	32KB
	安全ROM区域	0x0BF4 E000 - 0x0BF7 FFFF	200KB

注意：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。

RTDEC 功能

RTDEC 功能是指从闪存中读取数据时，可以根据 EFUSE 模块中配置的 AES 算法进行实时解密数据（写入闪存的数据已经加密）。当 EFUSE_USERCTL 寄存器中的 AESEN 位置 1 时，开启即时解密功能。这是通过硬件即时实现的，不能通过软件实现。使用 AES 算法，且 AES 密钥使用 EFUSE_AESKEY 寄存器的 AESKEY[127：0]位域来设置。

从 SIP 闪存获取数据，起始地址为 0x08000000，大小为 4MB 的区域，均支持 RTDEC 功能。

NO-RTDEC 功能

即使将 EFUSE_USERCTL 寄存器中 AESEN 位置 1，也可以通过 FMC_NODECx（x=0,1,2,3）寄存器组最多配置四个区域不使用 RTDEC 功能。

读偏移功能

为了满足 Wi-Fi OTA 功能的需求，可配置增加读偏移功能。将总线初始地址增加一个偏移后再从 FMC 总线中读取。通过配置 FMC_OFVR 寄存器和 FMC_OFRG 寄存器设置读取的偏移值和应用的区域后，读取源地址的值等同于读取添加偏移后的地址的值。如果还需配置其他功能，RTDEC 功能配置需要使用源地址。

注意：（1）添加读偏移的区域不支持再配置为 NO-RTDEC 区域。（2）偏移功能仅支持读操作，不支持编程操作和擦除操作。

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 OBWEN 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_NODECx（x=0,1,2,3） / FMC_OFRG / FMC_OFVR 寄存器。

注意：（1）一旦将错误的密钥写入 FMC_CTL，会产生总线错误。（2）将错误的密钥写入 OBKEY 不会产生总线错误。

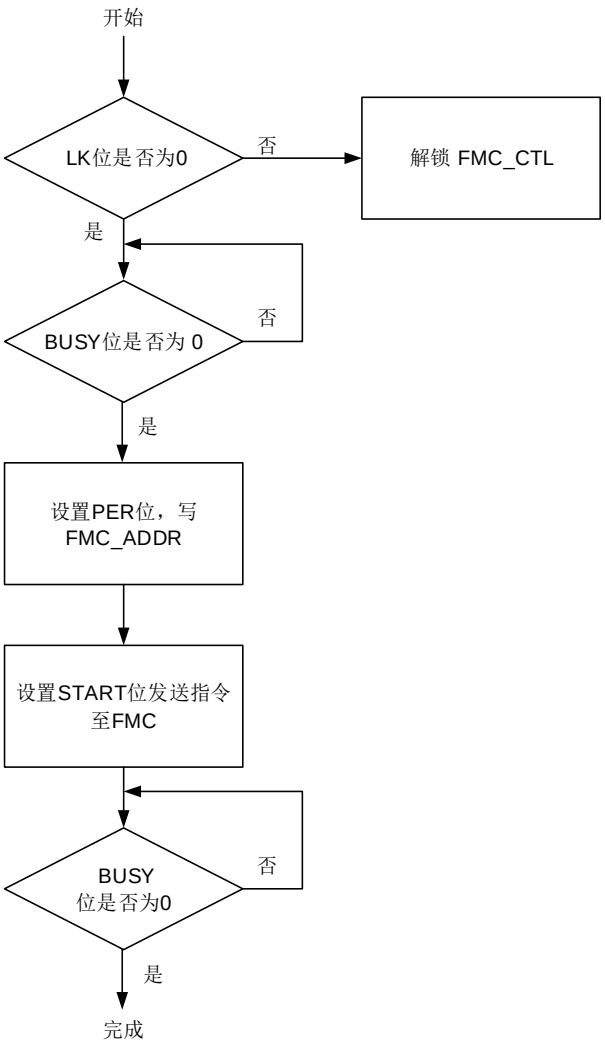
2.3.4. 页擦除

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。*图 2-1. 页擦除操作流程*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：

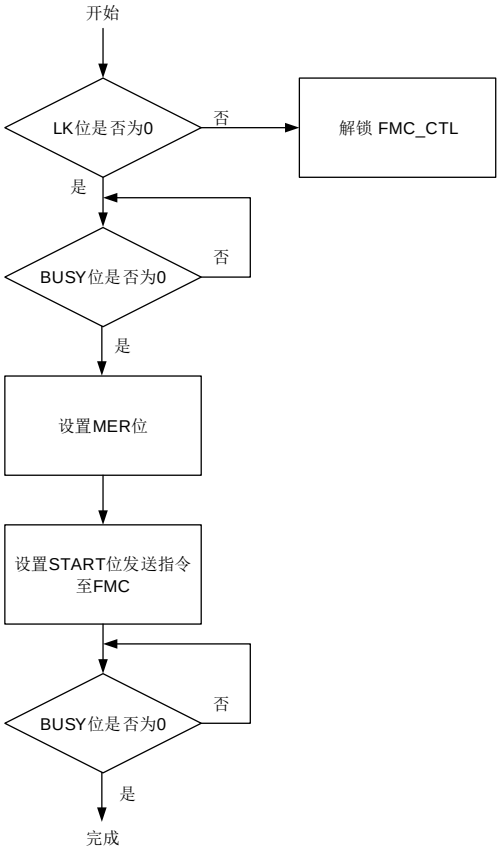
- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 如果整片擦除闪存，置位FMC_CTL寄存器的MER位；
- 通过将FMC_CTL寄存器的START位置1来发送整片擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用SBUS读并验证是否擦除成功。

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。

*图2-2. 整片擦除操作流程*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 位整字编程功能。

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；
- 写一个32位整字到目的绝对地址（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

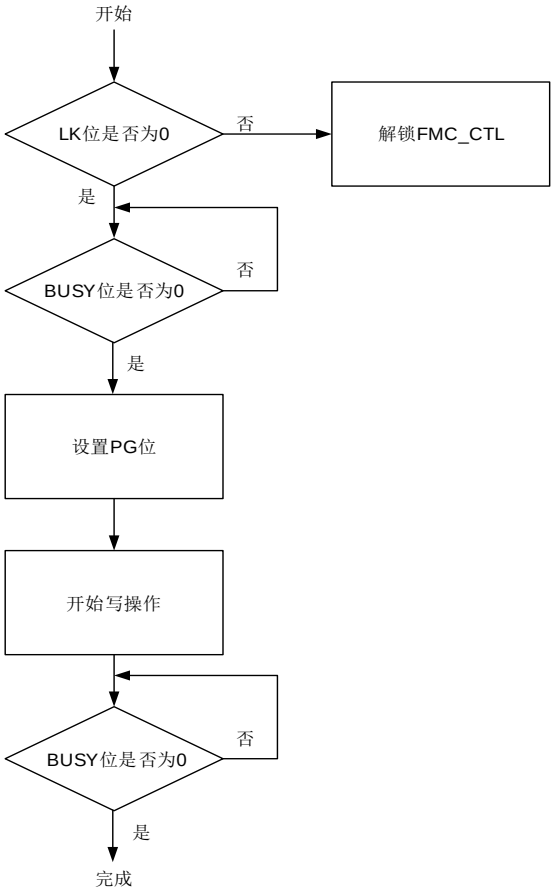
位被置 1，FMC 将触发一个中断。有一些编程错误需要注意：

每个字在擦除后和下次擦除前只能编程一次。注意 PG 位必须在字编程操作前被置位。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。

注意，如果编程数据未能写满 32 位，这些数据不会被编程入 flash 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 WPERR 位来检测发生了哪种错误。*图 2-3. 字编程操作流程*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：1.设置 PG 位后，支持连续编程以满足 WLAN OTA 要求。2.如果用户在编程前不进行软件读操作检查，则编程内容与闪存原有内容线与。3.为了获得最快的编程速度，用户在操作过程中需要注意以下几点：（1）32 位编程并且地址连续。（2）在总线上连续写入，写操作之间必须没有读操作。（3）两次写之间的间隔不能超过 64xT_{hclk}（闪存时钟是 hclk 的 2 分频）。

2.3.7. 选项字节

选项字节说明

选项字节说明见下表*表 2-2. 选项字节*，详细说明请参考寄存器章节。选项字节最终根据应用程序的要求来配置。

表 2-2. 选项字节

名称	寄存器映射
12：SRAM1_RST 11：NRST_DPSLP 10：NRST_STDBY 9：NWDG_HW [7:0]：SPC[7:0]	*选项字节寄存器（FMC_OBR）*
[31:0]：USER[31:0]	*选项字节用户寄存器（FMC_OBUSER）*
[25:16]：WRP0_EPAGE[9:0] [15:0]：WRP0_SPAGE[9:0]	*选项字节写保护/擦除保护寄存器0（FMC_OBWRP0）*
[25:16]：WRP1_EPAGE[9:0] [15:0]：WRP1_SPAGE[9:0]	*选项字节写保护/擦除保护寄存器1（FMC_OBWRP1）*

选项字节编程

修改选项字节，操作步骤如下：

- 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；
- 如有必要，解锁FMC_CTL寄存器中的选项字节操作位；
- 等待，直到FMC_CTL寄存器中的OBWEN位置1；
- 在选项字节寄存器中写入所需的值；
- 将FMC_CTL寄存器中的OBSTART位置1；
- 通过检查FMC_STAT寄存器中的BUSY位的值，等待所有操作完成；
- 设置OBRLD位或启动一次系统复位以加载选项字节，使之生效。

2.3.8. 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。

保护等级 0：无保护

当 FMC_OBR 寄存器中 SPC[7:0]位域被设置成 0xAA 时，系统复位后，闪存将处于无保护状态。闪存、SRAM1 和备份寄存器可以被所有操作模式访问。

保护等级 1：读保护

当 FMC_OBR 寄存器中 SPC[7:0]位域被设置成除 0xAA 外任意值时，系统复位后，闪存将处于保护等级 1 状态。

用户模式：从 flash 启动中执行的代码可以通过所有操作（读取，擦除和编程）访问闪存主存储器，SRAM1 和备份寄存器。

调试或从 RAM 启动或从 bootloader 启动：运行代码时，闪存主存储器，备份寄存器和 SRAM1 完全不可访问。在这些启动模式下，将检测到入侵，并且对闪存或 SRAM1 的读写访问会产生总线错误和内核异常。

表 2-3. Flash 在不同安全保护级别下的访问

访问方式	取指	读	写	页擦除
无保护，保护等级1(1)	OK		无写保护的页： OK 写保护的页：WI和WPERR标志置位	
保护等级1 (2)	总线错误			写无效，WPERR标志置位

注意：（1）从用户 flash 启动，无调试访问。（2）检测到调试访问或从 RAM 启动或从 bootloader 启动。

保护等级修改

通过将 FMC_OBR 寄存器中 SPC[7:0]位域的值更改为除 0xAA 外任意值，可以将安全保护等级可以从 0 级移至 1 级。

通过将 SPC[7:0]编程为 0xAA，可将闪存安全保护等级从 1 级移至 0 级，此时闪存主存储器将执行全片擦除。备用寄存器和所有 SRAM 也被擦除。

2.3.9. 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。在选项字节 FMC_OBR 中定义的写保护区域也被写保护。

2.3.10. Flash 中断

中断：操作结束/操作错误。

表 2-4. Flash 中断请求

中断事件	描述	清除方式	中断使能位
ENDF	操作结束	向 FMC_STAT 寄存器对应位写 1	ENDIE
WPERR	在被保护的页上进行擦除/编程操作		ERRIE

2.4. FMC 寄存器

FMC 基地址：0x4002 2000

2.4.1. 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。

2.4.2. 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。

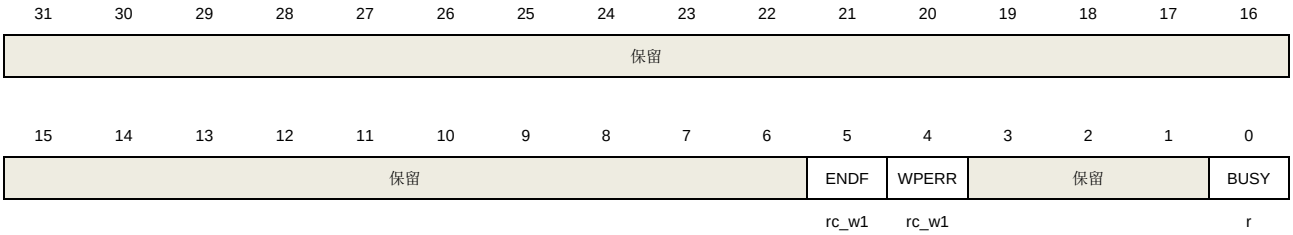
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OBKEY[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OBKEY[15:0]															
w															

位/位域	名称	描述
31:0	OBKEY[31:0]	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OBWEN位。

2.4.3. 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。



位/位域	名称	描述
31:6	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。
3:1	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。

2.4.4. 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。

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	OBRLD	选项字节重加载位 软件置 1 0：没有作用 1：强制选项字节重加载。
14	OBSTART	选项字节修改开始位 软件置 1 0：没有作用

		1：触发一次选项字节操作。仅在 OBWEN 位置 1 时才能写入。该位仅由软件设置，当在 FMC_STAT 中 BUSY 位清除时清除。
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	FMC_OBR/ FMC_OBUSER / FMC_OBWRPx(x=0,1) / FMC_NODECx（x=0,1,2,3）/ FMC_OFRG / FMC_OFVR 写使能位 当正确的序列写入 FMC_OBKEY 寄存器后，此位由硬件置 1。此位可以被软件清 0。
8	保留	必须保持复位值。
7	LK	FMC_CTL 寄存器锁定标志位 当正确的序列写入 FMC_KEY 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。
5:4	保留	必须保持复位值。
3	WTPG	WIFI 微调编程命令位 软件置 1 和清 0 0：无作用 1：WIFI 微调编程命令
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。

2.4.5. 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 该位域通过软件设置。 ADDR 位是闪存擦除/编程命令的地址

2.4.6. 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															

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

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	WP	EFFUSE 配置 32K 写保护状态位. 0：写保护复位 1：写保护置位
1	SPC	安全保护等级 1 是否开启状态位 0：安全保护等级 1 未开启 1：安全保护等级 1 开启

0	保留	必须保持复位值。

2.4.7. 选项字节寄存器（FMC_OBR）

地址偏移：0x40

复位值：0xXXXX XXXX（当 OBRLD 位置位或系统复位时，将从闪存中的值加载到寄存器的 0 至 31 位。其中 SRAM1_RST 位的加载条件必须是上电复位。）

仅当 OBWEN 位置 1 时，才能写入该寄存器。

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	SRAM1_RST	SRAM1 复位使能位 0：没有作用 1：自动清除 SRAM1 数据，系统复位后生效。
11	NRST_DPSLP	进入深度睡眠模式复位状态位 0：当进入 Deepsleep 模式时产生一个复位 1：当进入 Deepsleep 模式时不产生复位
10	NRST_STDBY	进入待机模式复位状态位 0：当进入 Standby 模式时产生一个复位 1：当进入 Standby 模式时不产生复位
9	NWDG_HW	看门狗控制配置位 0：选择硬件看门狗 1：选择软件看门狗
8	保留	必须保持复位值。
7:0	SPC[7:0]	闪存安全保护值，系统复位后生效。 注意：闪存的安全保护以此位域配置为准。

2.4.8. 选项字节用户寄存器（FMC_OBUSER）

地址偏移：0x44

复位值：0xXXXX XXXX（当 OBRLD 位置位或系统复位时，将从闪存中的值加载到寄存器 0

至 31 位。）

仅当 OBWEN 位置 1 时，才能写入该寄存器。

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
USER[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USER[15:0]															
rw															

位/位域	名称	描述
31:0	USER[31:0]	选项字节 USER 值

2.4.9. 选项字节写保护/擦除保护寄存器 0（FMC_OBWRP0）

地址偏移：0x48
复位值：0xXXXX XXXX（当 OBRLD 位置位或系统复位时，将从闪存中的值加载到寄存器 0 至 31 位）

仅当 OBWEN 位置 1 时，才能写入该寄存器。

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						WRP0_EPAGE[9:0]									
						rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						WRP0_SPAGE[9:0]									
						rw									

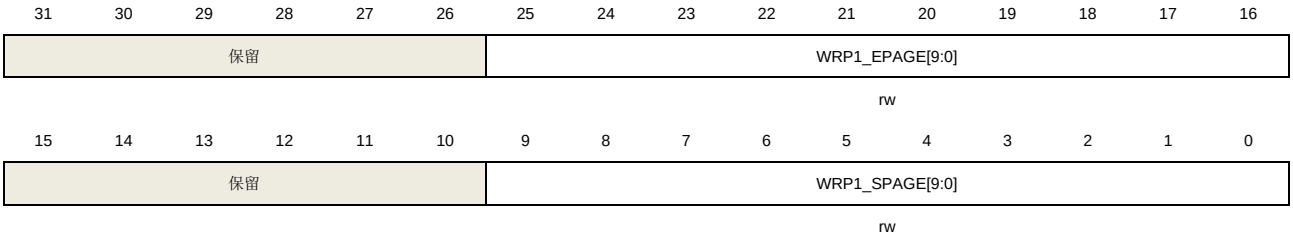
位/位域	名称	描述
31:26	保留	必须保持复位值。
25:16	WRP0_EPAGE[9:0]	写保护/擦除保护区域 0 的末尾页
15:10	保留	必须保持复位值。
9:0	WRP0_SPAGE[9:0]	写保护/擦除保护区域 0 的起始页

2.4.10. 选项字节写保护/擦除保护寄存器 1（FMC_OBWRP1）

地址偏移：0x4C
复位值：0xXXXX XXXX（当 OBRLD 位置位或系统复位时，将从闪存中的值加载到寄存器 0 至 31 位）

仅当 OBWEN 位置 1 时，才能写入该寄存器。

该寄存器只能按字（32 位）访问。



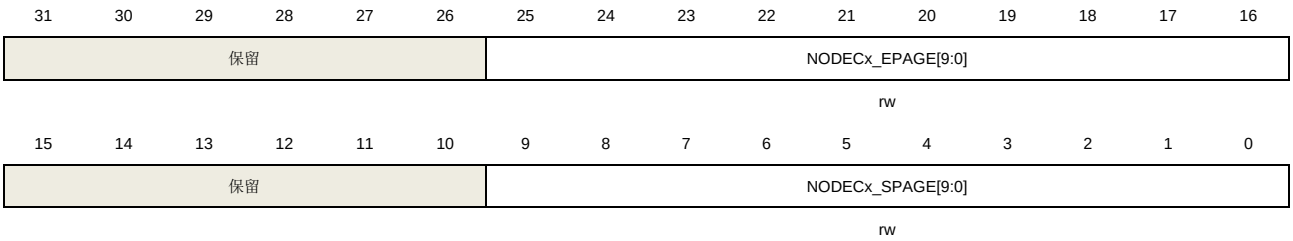
位/位域	名称	描述
31:26	保留	必须保持复位值。
25:16	WRP1_EPAGE[9:0]	写保护/擦除保护区域 1 的末尾页
15:10	保留	必须保持复位值。
9:0	WRP1_SPAGE[9:0]	写保护/擦除保护区域 1 的起始页

2.4.11. NO-RTDEC 区域寄存器 x（FMC_NODECx）（x = 0…3）

地址偏移：0x70 + 0x04 * x
复位值：0x0000 03FF

仅当 OBWEN 位置 1 时，才能写入该寄存器。

该寄存器只能按字（32 位）访问。



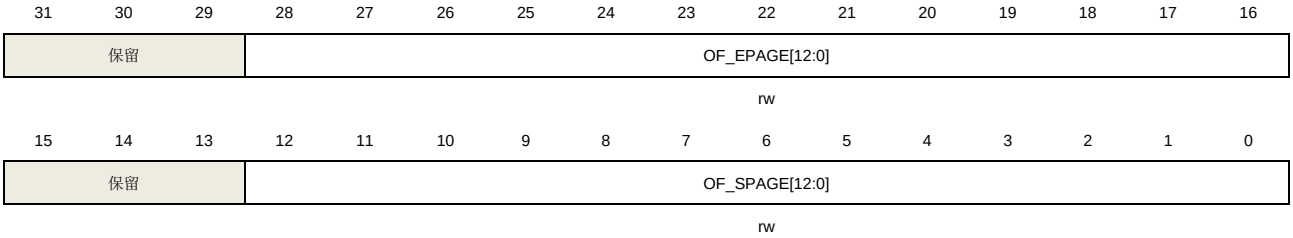
位/位域	名称	描述
31:26	保留	必须保持复位值。
25:16	NODECx_EPAGE[9:0]	NO-RTDEC 区域 x 的末尾页（x=0,1,2,3）
15:10	保留	必须保持复位值。
9:0	NODECx_SPAGE[9:0]	NO-RTDEC 区域 x 的起始页（x=0,1,2,3）

2.4.12. 偏移区域寄存器（FMC_OFRG）

地址偏移：0x80
复位值：0x0000 1FFF

仅当 OBWEN 位置 1 时，才能写入该寄存器。

该寄存器只能按字（32 位）访问。



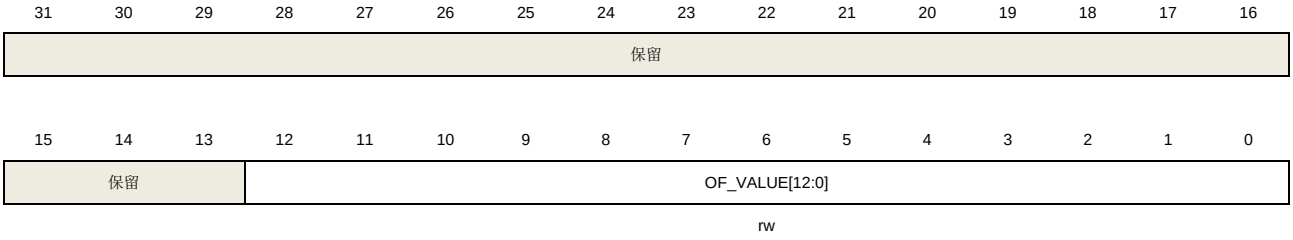
位/位域	名称	描述
31:29	保留	必须保持复位值。
28:16	OF_EPAGE[12:0]	添加偏移区域的末尾页
15:13	保留	必须保持复位值。
12:0	OF_SPAGE[12:0]	添加偏移区域的起始页

2.4.13. 偏移值寄存器（FMC_OFVR）

地址偏移：0x84
复位值：0x0000 0000

仅当 OBWEN 位置 1 时，才能写入该寄存器。

该寄存器只能按字（32 位）访问。

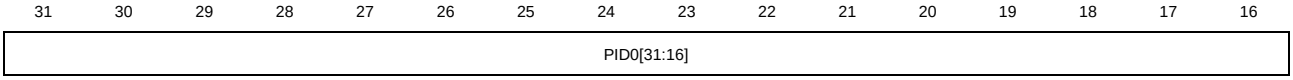


位/位域	名称	描述
31:13	保留	必须保持复位值。
12:0	OF_VALUE[12:0]	偏移值

2.4.14. 产品 ID0 寄存器（FMC_PID0）

地址偏移：0x100
复位值：0xXXXX XXXX

该寄存器只能按字（32 位）访问。



r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID0[15:0]															

r

位/位域	名称	描述
31:0	PID0[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。

2.4.15. 产品 ID1 寄存器（FMC_PID1）

地址偏移：0x104
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
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID1[15:0]															

r

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PID1[15:0]	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。

2.4.16. RF 微调寄存器 0（FMC_RFT0）

地址偏移：0x108
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
BLETXCAL[7:0]								WIFITXCAL[7:0]							

r r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THECAL[7:0]								PABIAST1[3:0]				PABIAST0[3:0]			

r r r

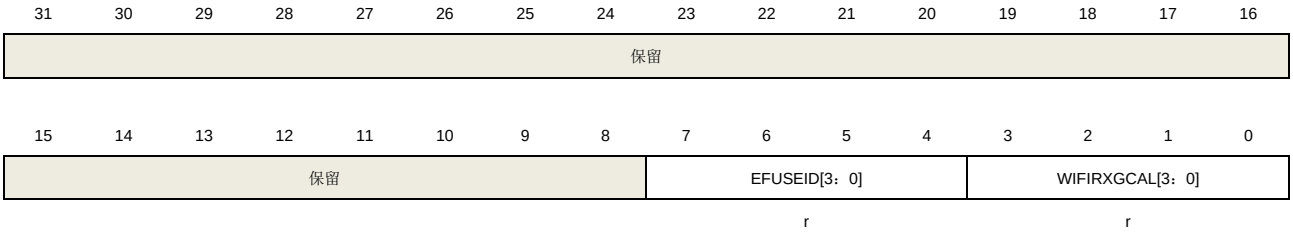
位/位域	名称	描述

31:24	BLETXCAL[7:0]	BLE 发射功率校准值
23:16	WIFITXCAL[7:0]	WIFI 发射功率校准值
15:8	THECAL[7:0]	温度值校准值
7:4	PABIAST1[3:0]	功率放大器微调值
3:0	PABIAST0[3:0]	功率放大器粗调值

2.4.17. RF 微调寄存器 1（FMC_RFT1）

地址偏移：0x10C
复位值：0x0000 00XX

该寄存器只能按字（32 位）访问。

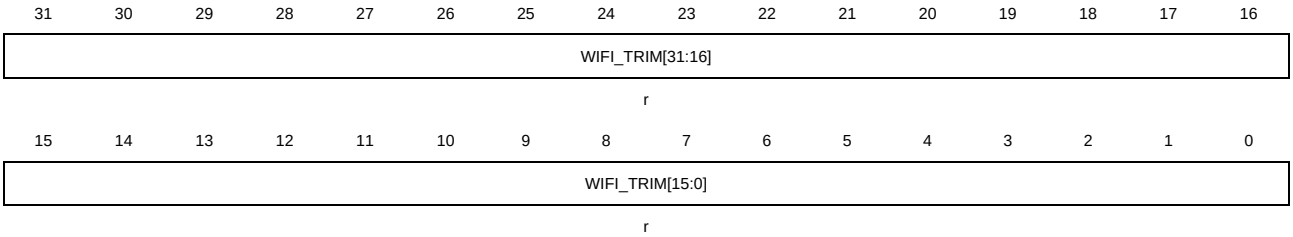


Bits	Field	Descriptions
31:8	保留	必须保持复位值。
7:4	EFUSEID[3:0]	EFUSE 版本 ID
3:0	WIFIRXGCAL[3:0]	WIFI 接收增益校准值

2.4.18. WIFI 调整寄存器 x（FMC_WFTx）（x = 0…15）

地址偏移：0x200 + 0x4 * x
复位值：0xXXXX XXXX

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:0	WIFI_TRIM[31:0]	系统复位后，从闪存加载。 WTPG 设置为 1 后，可以烧写，对应的 flash 不能擦除。在寄存器烧写过程中，WTPG 位被置 1。直到 BUSY 位为 0，表示烧写结束。 这些位在芯片生产时是一次编程，每个编程一个寄存器（32 位）。 系统复位后更新编程

的值。

3. 熔丝（EFUSE）

3.1. 简介

熔丝（EFUSE）作为一种非易失性存储单元存储了一些必需的系统参数。其中的每一个比特位只允许从 0 被改写为 1。根据软件操作，EFUSE 控制器可以对系统参数中的所有位进行编程。

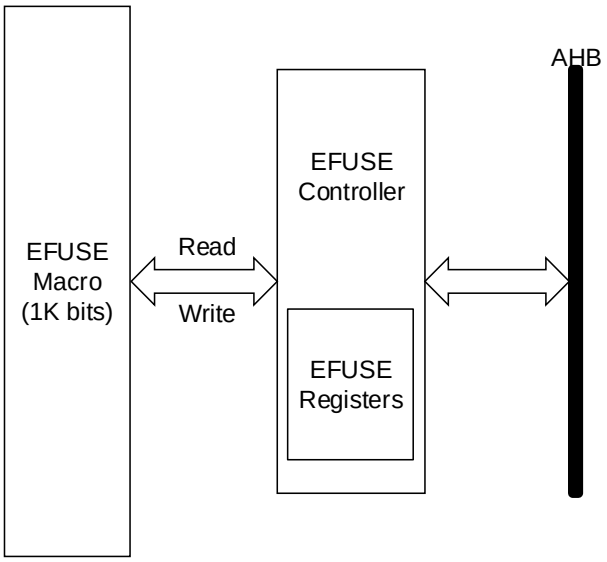
3.2. 主要特性

- 熔丝的存储单元大小为128*8比特
- 熔丝中的所有比特位不支持回退（从1改为0）
- 熔丝内容只能通过相应的寄存器读取

3.3. 模块框图

熔丝控制器实现了熔丝的读写操作逻辑，其中熔丝模块的存储单元共计 1Kb。

图 3-1. 熔丝控制器结构框图



3.4. 功能描述

3.4.1. 熔丝结构

1024 比特的熔丝存储单元划分为 128 个字节，每一个字节有一个 7 位地址，熔丝的详细地址映射见下表*表 3-1. 熔丝地址映射*。

表 3-1. 熔丝地址映射

地址[6:0]	熔丝字节
000_0000	Efuse[0]
000_0001	Efuse[1]
000_0010	Efuse[2]
. . .	. . .
111_1111	Efuse[127]

3.4.2. 熔丝内容简介

熔丝存储单元中存储了 10 个系统参数，不同的系统参数具有不同的位宽。

*表 3-2. 系统参数*显示了熔丝中存储的系统参数详情。

表 3-2. 系统参数

名称	位宽/字节	起始地址	写保护属性	读保护属性	描述	备注
Efuse 控制段 0	1B	7'd0	参数整体可多次写入，但每个比特位不可回退	系统复位后读出并保持不变，总线可读	MCU 启动所需的相关控制参数详细内容请参考*控制寄存器0（EFUSE_CTL0）*	用户自定义
Efuse 控制段 1	1B	7'd1	参数整体可多次写入，但每个比特位不可回退	系统复位后读出并保持不变，总线可读	详细内容请参考*控制寄存器 1（EFUSE_CTL1）*	用户自定义
Flash 存储保护段	1B	7'd2	参数整体可多次写入，但每个比特位不可回退	系统复位后读出并保持不变，总线可读	Flash 安全保护选项 详细内容请参考*安全保护控制寄存器（EFUSE_FPCTL）*	用户自定义
用户控制段	1B	7'd3	参数整体可多次写入，但每个比特位不可回退	系统复位后读出并保持不变，总线可读	用户控制字节选项 详细内容请参考*用户控制寄存器（EFUSE_USERCTL）*	用户自定义
保留段	12B	7'd4	参数整体可多次写入，但每个比特位不可回退	保留待用	详细内容请参考*保留寄存器 x（EFUSE_RESx）*	用户自定义
AES 秘钥	16B	7'd16	仅可写一次	系统复位后读出并保持不变，总	加密固件所需的 AES 秘钥 详细内容请参考*固件 AES 秘钥*	用户自定

名称	位宽/字节	起始地址	写保护属性	读保护属性	描述	备注
				线可读	***寄存器 x（EFUSE_AESKEYx）***	义
RoTPK 秘钥（或其哈希值）	32B	7'd32	仅可写一次	系统复位后仅通过 ROM 可读	可信根秘钥（ECC 的公钥或者 RSA 公钥哈希值）详细内容请参考 ***RoTPK 秘钥寄存器 x（EFUSE_ROTPKKEYx）***	用户自定义
MCU 唯一设备 ID	16B	7'd64	不可改写	系统复位后读出并保持不变，总线可读	MCU UID 详细内容请参考***产品 UID 寄存器 x（EFUSE_PUIDx）***	厂商定义
HUK 秘钥	16B	7'd80	不可改写	系统复位后读出并保持不变，总线可读	硬件唯一密钥，为机密性提供 RoT（信任根）。详细内容请参考 ***HUK 秘钥寄存器 x（EFUSE_HUKKEYx）***	厂商定义
用户自定义数据段	32B	7'd96	系统复位后仅可写一次	系统复位后读出，总线可读	用户自定义数据 详细内容请参考***用户数据寄存器 x（EFUSE_USER_DATAx）***	用户自定义

注意：

(1) 7 位地址编码。
(2) 系统参数必须按其宽度读取，并且也建议根据它的宽度写入。

3.4.3. 读操作

熔丝中的内容只能通过对应的寄存器来访问，系统复位后，熔丝中的值被加载至寄存器中并生效。熔丝的读操作步骤如下：

1. 将EFUSE_CS寄存器中的RDIF位清零，并确保没有出现越界错误；
2. 将EFUSE_CS寄存器中的EFRW位清零；
3. 在EFUSE_ADDR寄存器中填入需要读取的熔丝地址及大小；
4. 将EFUSE_CS寄存器中EFSTR位置1；
5. 等待读操作完成，EFUSE_CS寄存器中的RDIF位置位；
6. 读取对应的寄存器值以获取熔丝值。

当读取操作成功后，EFUSE_CS 寄存器中的 RDIF 位会置位，如果 EFUSE_CS 中的 RDIE 位置位，熔丝控制器会产生一个操作完成中断。

3.4.4. 写操作

熔丝中的内容只能通过对应的寄存器来写入，熔丝的写操作步骤如下：

1. 将EFUSE_CS寄存器中的PGIF位清零，并确保没有出现越界错误；

2. 将EFUSE_CS寄存器中的EFRW位置1；
3. 在EFUSE_ADDR寄存器中填入需要写入的熔丝地址及大小；
4. 在对应的寄存器中写入数据；
5. 将EFUSE_CS寄存器中的EFSTR位置1；
6. 等待写操作完成，EFUSE_CS寄存器中的PGIF位置位。

当写操作完成后，EFUSE_CS 寄存器中的 PGIF 位会置位，如果 EFUSE_CS 中的 PGIE 位置位，熔丝控制器会产生一个操作完成中断。另外需要注意的是，数据写入的寄存器所对应的熔丝地址以及数据大小应与 EFUSE_ADDR 寄存器中的地址和大小相吻合，否则如果 EFUSE_CS 寄存器中的 OVBERIF 置位，会产生一次错误中断。

3.5. EFUSE 寄存器

EFUSE 基地址：0x4002 2800

3.5.1. 控制及状态寄存器（EFUSE_CS）

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					OVBERIC	RDIC	PGIC	保留	OVBERIE	RDIE	PGIE	保留	OVBERIF	RDIF	PGIF
					rc_w1	rc_w1	rc_w1		rw	rw	rw		r	r	r

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

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	OVBERIC	越界错误中断标志清除位 0：无影响 1：清除越界错误标志位
25	RDIC	读操作完成中断标志清除位 0：无影响 1：清除读操作完成中断标志位
24	PGIC	写操作完成中断标志清除位 0：无影响 1：清除写操作完成中断标志位
23	保留	必须保持复位值。
22	OVBERIE	越界错误中断使能位

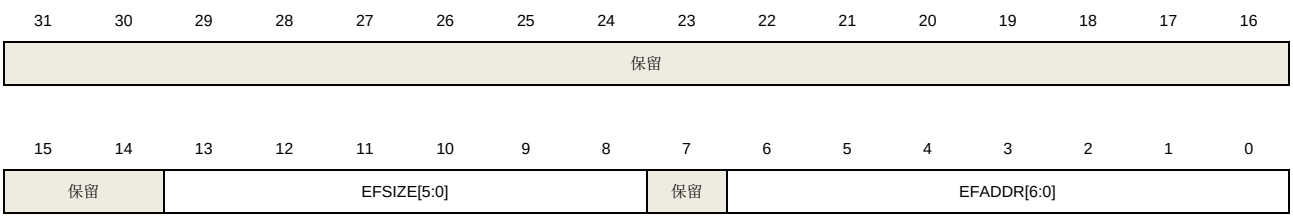
		0：禁能越界错误中断 1：使能越界错误中断
21	RDIE	读操作完成中断使能位 0：禁能读操作完成中断 1：使能读操作完成中断
20	PGIE	写操作完成中断使能位 0：禁能写操作完成中断 1：使能写操作完成中断
19	保留	必须保持复位值。
18	OVBERIF	越界错误标志位 0：未发生越界错误 1：发生越界错误
17	RDIF	读操作完成标志位 0：读操作未完成 1：读操作完成
16	PGIF	写操作完成标志位 0：写操作未完成 1：写操作完成
15:2	保留	必须保持复位值。
1	EFRW	熔丝读写操作选择位 0：读熔丝内容 1：写熔丝内容 当EFSTR为1时该位不可写
0	EFSTR	熔丝读/写操作开始位 该位由软件置1，硬件清0 0：无影响 1：开始读/写操作

3.5.2. 地址寄存器（EFUSE_ADDR）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



rw rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:8	EFSIZE[5:0]	读/写熔丝数据大小
7	保留	必须保持复位值。
6:0	EFADDR[6:0]	读/写熔丝数据起始地址

注意：当 EFUSE_CS 寄存器中的 EFSTR 位为 1 时，该寄存器不可修改。

3.5.3. 控制寄存器 0（EFUSE_CTL0）

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										SWBOOT0	EFBOOT0	SWBOOT1	EFBOOT1	EFBOOTLK	EFSB
										rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	SWBOOT0	BOOT0输出选择位 0：选择BOOT0引脚作为BOOT0输出 1：选择EFBOOT0位作为BOOT0输出
4	EFBOOT0	熔丝BOOT0位 0：Efuse_boot0 = 0 1：Efuse_boot0 = 1
3	SWBOOT1	BOOT1输出选择位 0：选择BOOT1引脚作为BOOT1输出 1：选择EFBOOT1位作为BOOT1输出
2	EFBOOT1	熔丝BOOT1位 0：Efuse_boot1 = 0 1：Efuse_boot1 = 1
1	EFBOOTLK	EFUSE_CTL0寄存器bits[5:2]锁定位 0：可以对EFUSE_CTL0寄存器中的bits[5:2]进行改写

		1：锁定EFUSE_CTL0寄存器bits[5:2]，这些位不可以被改写
0	EFSB	从secure boot启动 该位需要与bits[5:1]配合决定启动方式 0：主Flash启动 1：安全启动

3.5.4. 控制寄存器 1（EFUSE_CTL1）

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								VFCERT	VFIMG	保留			ROTLK	NDBG	保留
								rw	rw				rw	rw	

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	VFCERT	验证固件证书 0：禁能固件证书认证功能 1：使能固件证书认证功能
6	VFIMG	验证固件镜像 0：禁能固件镜像认证功能 1：使能固件镜像认证功能
5:3	保留	必须保持复位值。
2	ROTLK	EFUSE_ROTPKKEY寄存器锁定位 0：EFUSE_ROTPKKEY寄存器，寄存器内容可改写。 1：锁定EFUSE_ROTPKKEY寄存器，寄存器内容不可改写。
1	NDBG	调试设置位 0：不允许调试 1：允许调试
0	保留	必须保持复位值。

3.5.5. 安全保护控制寄存器（EFUSE_FPCTL）

地址偏移：0x10
复位值：0x0000 00X0

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														FP[2:0]	
														rw	

位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	FP[2:0]	熔丝中安全保护值 Bit2：0~32K 写保护 Bit1：保留 Bit0：闪存安全保护等级 1

3.5.6. 用户控制寄存器（EFUSE_USERCTL）

地址偏移：0x14
复位值：0x0000 0006

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									UDLK	AESEN	保留	EFOPLK	保留	LOGUART	SBCLK
									rw	rw		rw		rw	rw

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	UDLK	EFUSE_USER_DATA寄存器锁定位 0：解锁EFUSE_USER_DATA寄存器，寄存器内容可改写。 1：锁定EFUSE_USER_DATA寄存器，寄存器内容不可改写。
5	AESEN	EFUSE_AESKEY寄存器锁定及AES加解密功能使能位 0：禁能AES加解密功能并且解锁EFUSE_AESKEY寄存器，寄存器内容可改写。 1：使能AES加解密功能并锁定EFUSE_AESKEY寄存器，寄存器内容不可改写。
4	保留	必须保持复位值。
3	EFOPLK	EFUSE_FPCTL和EFUSE_USERCTL寄存器锁定位 0：解锁EFUSE_FPCTL和EFUSE_USERCTL寄存器，寄存器内容可改写。

		1：锁定EFUSE_FPCTL和EFUSE_USERCTL寄存器，寄存器内容不可改写。
2	保留	必须保持复位值。
1	LOGUART	安全Boot的Log输出UART选择 0：UART2（PA6/PA7） 1：UART1（PA4/PA5）
0	SBCLK	安全Boot的时钟源选择 0：IRC16M 1：外部HXTAL

3.5.7. 保留寄存器 x（EFUSE_RESx）（x = 0…2）

地址偏移：0x18 + 0x4 * 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
RES[31:16]															
rw															

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

位/位域	名称	描述
31:0	RES[7:0]	熔丝保留内容

3.5.8. 固件 AES 秘钥寄存器 x（EFUSE_AESKEYx）（x = 0…3）

地址偏移：0x24 + 0x4 * 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
AESKEY[31:16]															
rw															

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

位/位域	名称	描述
31:0	AESKEY[31:0]	熔丝中AES秘钥字段值

3.5.9. RoTPK 秘钥寄存器 x（EFUSE_ROTPKKEYx）（x = 0…7）

地址偏移：0x34 + 0x4 * 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
RKEY[31:16]															
rw															

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

位/位域	名称	描述
31:0	RKEY[31:0]	熔丝中RoTP或其HASH字段值

3.5.10. 产品 UID 寄存器 x（EFUSE_PUIDx）（x = 0…3）

地址偏移：0x54 + 0x4 * 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
UID[31:16]															
r															

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

位/位域	名称	描述
31:0	UID[31:0]	熔丝中MCU UID值

3.5.11. HUK 秘钥寄存器 x（EFUSE_HUKKEYx）（x = 0…3）

地址偏移：0x64 + 0x4 * 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
HKEY[31:16]															
r															

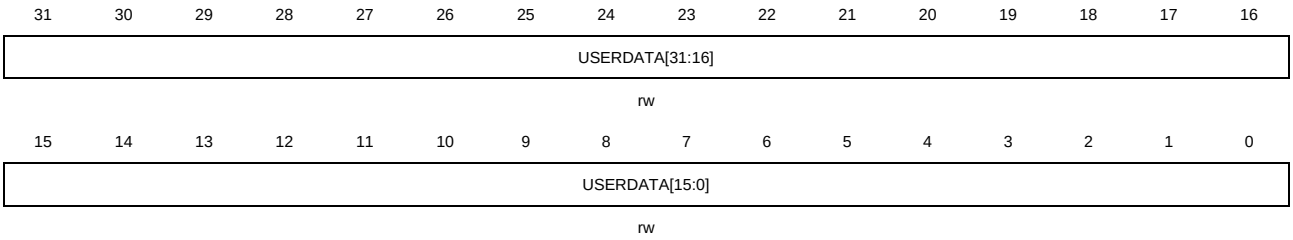
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HKEY[15:0]															
r															

位/位域	名称	描述
31:0	HKEY[31:0]	熔丝中HUK秘钥字段值

3.5.12. 用户数据寄存器 x（EFUSE_USER_DATAx）（x = 0…7）

地址偏移：0x74 + 0x4 * x
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

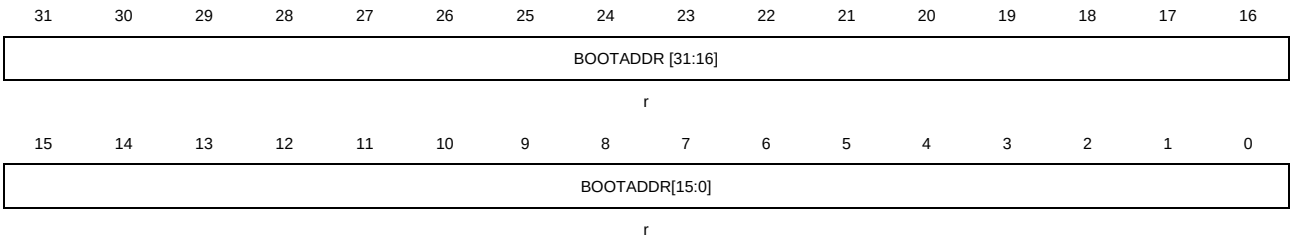


位/位域	名称	描述
31:0	USERDATA[31:0]	熔丝中USER_DATA字段值

3.5.13. 启动地址寄存器（EFUSE_BOOTADDR）

地址偏移：0x124
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:0	BOOTADDR[31:0]	从该寄存器所存地址启动

4. 电源管理单元（PMU）

4.1. 简介

功耗设计是 GD32VW55x 系列产品比较注重的问题之一。电源管理单元提供了六种省电模式，包括睡眠模式，深度睡眠模式，待机模式，SRAM 睡眠模式，Wi-Fi 睡眠模式和 BLE 睡眠模式。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 4-1. 电源域概览*所示，GD32VW55x 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。备份域由 V_{DD} 电源供电。

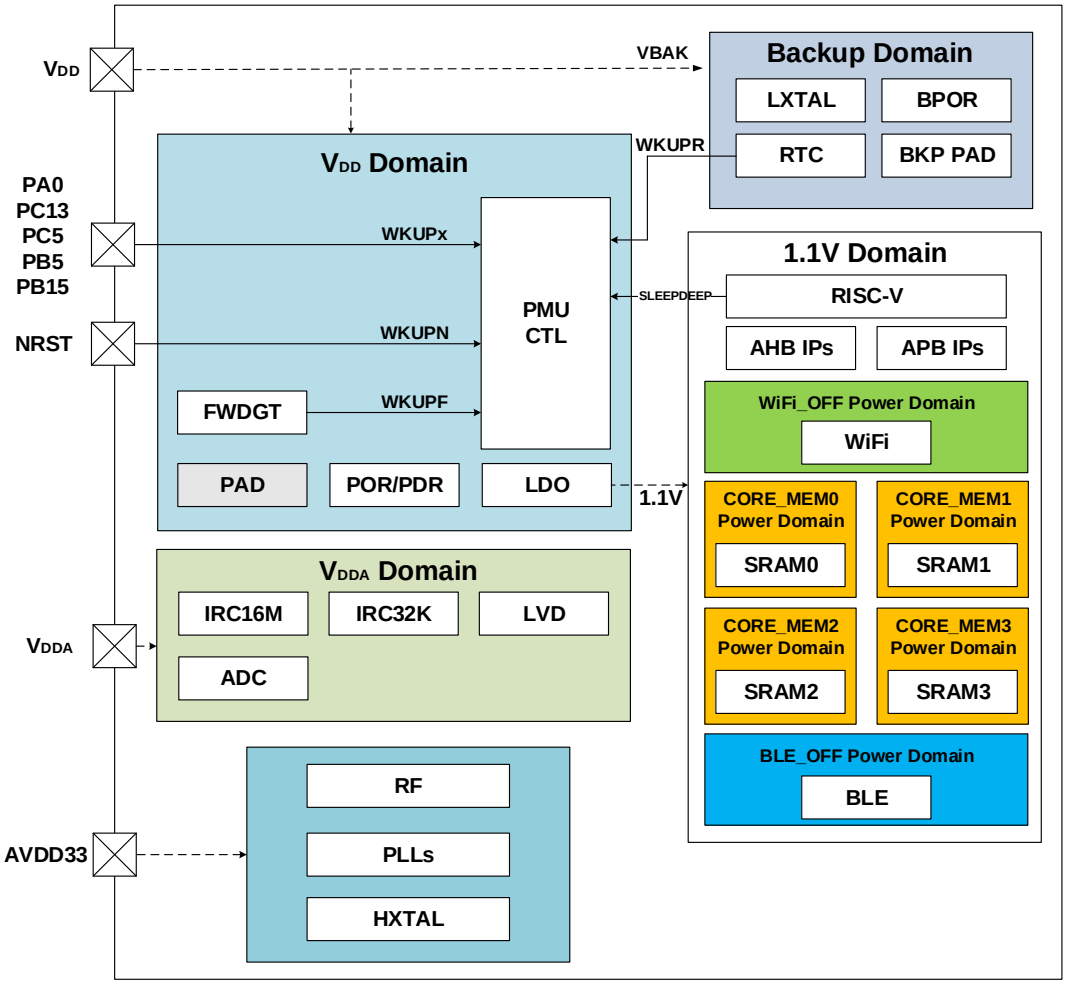
4.2. 主要特征

- 两个电源域：备份域、V_{DD} / V_{DDA}域和1.1V电源域；
- 六种省电模式：睡眠模式、深度睡眠模式，待机模式，SRAM睡眠模式，Wi-Fi睡眠模式和BLE睡眠模式；
- 内部电压调节器（LDO）为1.1V电源域提供1.1V电源；
- 提供低电压检测器（LVD），当V_{DD}电压低于所设定的阈值时能发出中断或事件；
- 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下超低功耗；
- 可以关闭SRAM0/SRAM1/SRAM2/SRAM3的电源以节约能耗；
- 可以关闭WiFi_off域的电源；
- 可以关闭BLE_off域的电源。

4.3. 功能说明

*图 4-1. 电源域概览*提供了 PMU 及相关电源域的内部结构框图。

图 4-1. 电源域概览



LVD：低压检测器　　LDO：电压调节器　　BPOR：V_{BAK}上电复位

POR：上电复位　　PDR：掉电复位

4.3.1. 备份域

备份域由 V_{DD} 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）、BPOR（备份域上电复位），以及 PC13 至 PC15 共 3 个 PAD。

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 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 1-32 分频。在通过 WFI / WFE 指令进入省电模式之前，

RISC-V 需要通过 RTC 寄存器设置预期的闹钟时间并启用闹钟功能，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。

注意：由于 PC13 至 PC15 引脚仅可通过小电流，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)。

4.3.2. 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）和所有 PAD 等等。

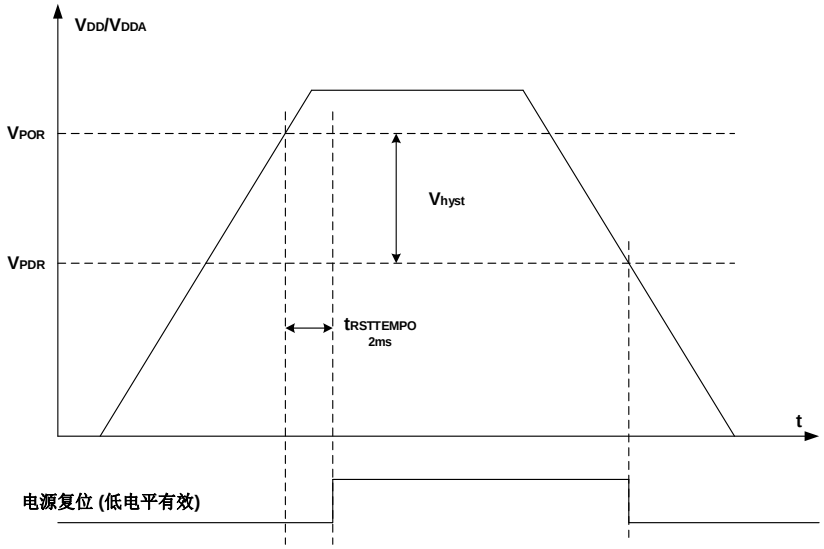
V_{DDA} 域包括 ADC（AD 转换器）、IRC16M（内部 16M RC 振荡器）、IRC32K（内部 32KHz RC 振荡器）、PLLs（锁相环）和 LVD（低电压检测器）等等。

V_{DD} 域

为 1.1V 域供电的 LDO（电压调节器），其复位后保持使能。可以被配置为三种不同的工作状态：包括睡眠模式（全供电状态）、深度睡眠模式（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 4-2. 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，典型值参考 GD32VW55x datasheet，V_{PDR} 表示掉电复位的阈值电压，典型值参考 GD32VW55x datasheet。迟滞电压 V_{hyst} 值参考 GD32VW55x datasheet。

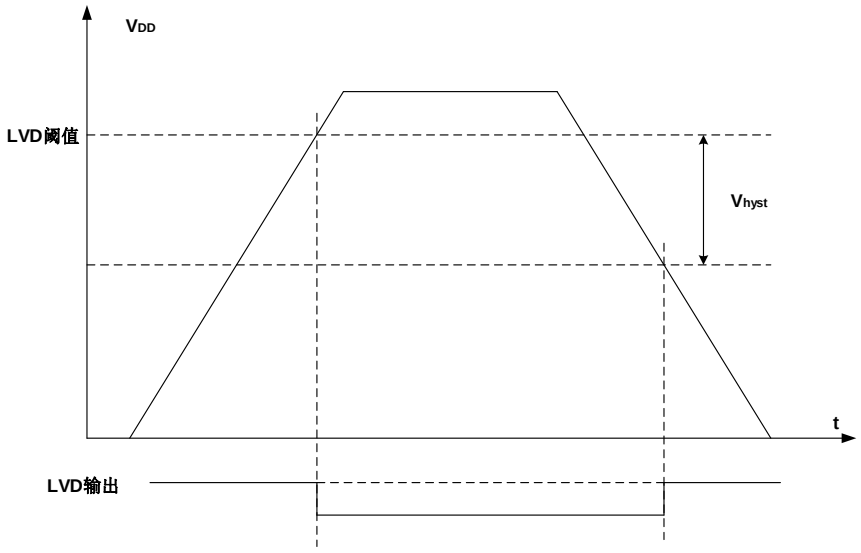
图 4-2. 上电/掉电复位波形图



V_{DDA} 域

LVD 的功能是检测 V_{DD} 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(PMU_CTL0)中的 LVDT[2:0]位进行配置。LVD 通过 LVDEN 置位使能，位于电源状态寄存器(PMU_CS0)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 4-3. 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} 值参考 GD32VW55x datasheet。

图 4-3. 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 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。

4.3.3. 1.1V 电源域

主要功能包括 RISC-V 内核逻辑、AHB / APB 外设、备份域和 V_{DD} / V_{DDA} 域的 APB 接口、SRAM_OFF 域、BLE_OFF 域和 Wi-Fi_OFF 域等。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。关于这方面的详细内容，将在以下章节予以说明。

WiFi_OFF 域

参照 WiFi 规范手册。WiFi_OFF 电源域典型的工作状态为：

1. 待机模式时，WiFi_OFF电源域掉电。
2. 当运行模式/睡眠/深度睡眠模式时，WiFi_OFF电源域可以配置为上电。
3. 当运行模式/睡眠/深度睡眠模式时，WiFi_OFF电源域可以配置为掉电（默认状态）。

CORE_MEM0/1/2/3 域

GD32VW55x 中，CORE_MEM0/1/2/3 电源域是分别用于 48KB SRAM0（SRAM0 的前 16KB 无法配置掉电）/64KB SRAM1/64KB SRAM2/128KB SRAM3 的电源控制。当运行模式/睡眠/深度睡眠模式时，SRAMs 可以配置掉电。典型的工作状态为：

1. 待机模式时，CORE_MEM0/1/2/3电源域掉电。
2. 当运行模式/睡眠/深度睡眠模式时，CORE_MEM0/1/2/3电源域可以配置为上电（默认状态）。
3. 当运行模式/睡眠/深度睡眠模式时，CORE_MEM0/1/2/3电源域可以配置为掉电。

BLE_OFF 域

BLE_OFF 电源域典型的工作状态为：

1. 待机模式时，BLE_OFF电源域掉电。
2. 当运行模式/睡眠/深度睡眠模式时，BLE_OFF电源域可以配置为上电。
3. 当运行模式/睡眠/深度睡眠模式时，BLE_OFF电源域可以配置为掉电（默认状态）。

4.3.4. 省电模式

系统复位或电源复位后，GD32VW55x 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：减慢系统时钟（HCLK，PCLK1，PCLK2），关闭未使用的外设的时钟或通过 PMU_CTL0 寄存器的 LDOVS 来配置 LDO 输出电压。LDOVS 只有在 PLL 关闭情况下才可以配置。此外，六种省电模式可以实现更低的功耗，它们是睡眠模式、深度睡眠模式，待机模式，SRAM 睡眠模式，BLE 睡眠模式和 Wi-Fi 睡眠模式。

睡眠模式

睡眠模式与 RISC-V 的 SLEEPING 模式相对应。在睡眠模式下，仅关闭 RISC-V 的时钟。如需进入睡眠模式，只要清除 RISC-V 系统控制寄存器中的 CSR_SLEEPVALUE 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。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。

深度睡眠模式

深度睡眠模式与 RISC-V 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、HXTAL 及 PLLs 也全部被禁用。SRAM0/1/2/3 和寄存器中的内容被保留。根据 PMU_CTL0 寄存器的 LDOLP 位的配置，可控制 LDO 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 RISC-V 系统控制寄存器的 CSR_SLEEPVALUE 位置 1，再清除 PMU_CTL0 寄存器的 STBMOD 位，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。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 工作在低功耗模式，那么唤醒时需额外的延时时间。

在深度睡眠模式下，通过配置 PMU_CTL0 寄存器的 LDEN[1:0]，LDNP，LDLP，LDOLP 位可以进入低驱动模式。低驱动模式具有低驱动能力，低功耗模式工作能耗很低。

正常驱动&正常功耗：将 PMU_CTL0 寄存器的 LDEN[1:0]位配置为 00，深度睡眠模式就工作在正常驱动模式下。将 PMU_CTL0 寄存器的 LDOLP 清 0 可以退出低功耗模式。

正常驱动&低功耗：将 PMU_CTL0 寄存器的 LDEN[1:0]位配置为 00，深度睡眠模式就工作在正常驱动模式下。将 PMU_CTL0 寄存器的 LDOLP 置 1 可以进入低功耗模式。

低驱动&正常功耗：将 PMU_CTL0 寄存器的 LDEN[1:0]设置为 0b11，LDNP 置 1 可以进入深度睡眠模式的低驱动模式。将 PMU_CTL0 寄存器的 LDOLP 清 0 可以使 LDO 处于正常功耗模式。

低驱动&低功耗：将 PMU_CTL0 寄存器的 LDEN[1:0]设置为 0b11，LDLP 置 1 可以进入深度睡眠模式的低驱动模式。将 PMU_CTL0 寄存器的 LDOLP 置 1 可以使 LDO 处于低功耗模式。

非低驱动：将 PMU_CTL0 寄存器的 LDEN[1:0]设置为 00，深度睡眠模式将不会处在低驱动模式。

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 6-2. 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。

待机模式

待机模式是基于 RISC-V 的 SLEEPDEEP 模式实现的。在待机模式下，整个 1.1V 域全部停止供电，同时 LDO 和包括 IRC16M、HXTAL 和 PLL 也会被关闭。进入待机模式前，先将 RISC-V 系统控制寄存器的 CSR_SLEEPVALUE 位置 1，再将 PMU_CTL0 寄存器的 STBMOD 位置

1，再清除 PMU_CS0 寄存器的 WUF 位，然后执行 WFI 或 WFE 指令，系统进入待机模式，PMU_CS0 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，SRAM0/SRAM1/SRAM2/SRAM3 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 RISC-V 将从 0x00000000 地址开始执行指令代码。

SRAM 睡眠模式

当 SRAM0/SRAM1/SRAM2/SRAM3 中至少有一个掉电时，置位 PMU_CTL1 寄存器中的 SRAMxPSLEEP（x = 0/1/2/3）位，则对应的 SRAMx（x = 0/1/2/3）开始进入掉电状态（需等待几个 PCLK 时钟，SRAM 才可完全掉电，进入 SRAM 睡眠模式）。

置位 PMU_CTL1 寄存器中的 SRAMxPWAKE（x = 0/1/2/3）位，则对应的 SRAMx（x = 0/1/2/3）上电。

当运行模式/睡眠/深度睡眠模式时，SRAMx（x = 0/1/2/3）可以配置为上电或掉电。

当待机模式时，SRAMx（x = 0/1/2/3）掉电。

BLE 睡眠模式

当 BLE 进入 BLE 睡眠模式，BLE_OFF 域掉电。

当退出 BLE 睡眠模式，BLE 处于工作状态，所有 BLE 电源域上电。

Wi-Fi 睡眠模式

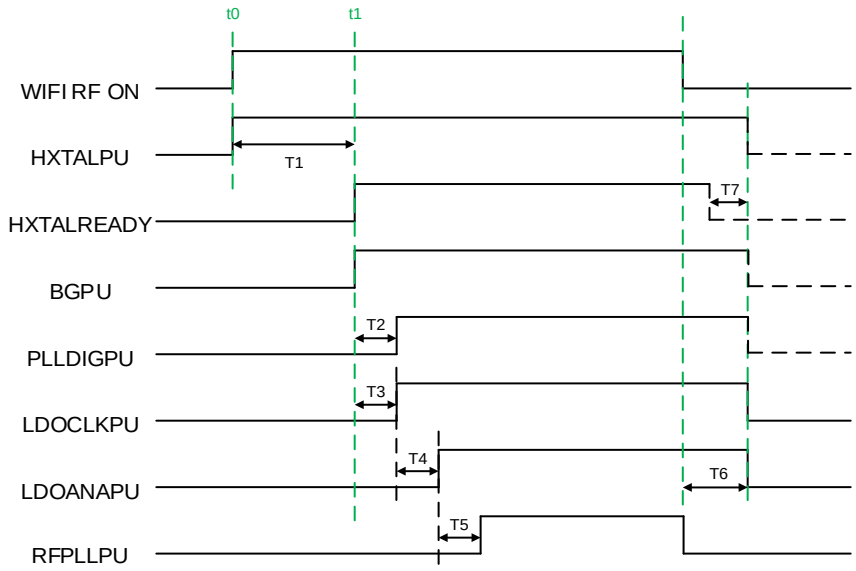
进入 Wi-Fi 睡眠模式可以通过软件置位 WPEN 和 WPSLEEP 位，或者硬件方式（当 WPEN 为 1 时，由 Wi-Fi 硬件 sleep_wl 信号驱动）。退出 Wi-Fi 睡眠模式可以通过软件清零 WPEN 位，或者当 WPEN 位为 1 时置位 WPWAKE，或者硬件方式（当 WPEN 为 1 时，由 Wi-Fi 硬件 wake_wl 信号驱动）。

当 Wi-Fi 进入 Wi-Fi 睡眠模式，Wi-Fi_OFF 电源域掉电。

当退出 Wi-Fi 睡眠模式，Wi-Fi 处于工作状态，所有 Wi-Fi 电源域上电。

RF 模块接口的时序图如下图*图 4-4. RF 时序*所示。

图 4-4. RF 时序



HXTALPU，HXTALREADY 和 PLLDIGPU 信号都是 RCU_CTL 寄存器的位。BGPU，LDOCLKPU，LDOANAPU 和 RFPLLPU 都是 RCU_CFG1 寄存器的位。

当 PLLDIGPU 为 1 时，RCU_PLLCFG 寄存器中的 PLLDIGSEL[1:0]位值不能改变，并且 RCU_CTL 寄存器的 PLLDIGEN 位电平不能有上升沿。

- 若 RFSWEN 位为 0，选择硬件方式配置 RF 时序：

当软件置位 WPSLEEP 位或 Wi-Fi 硬件 sleep_wl 信号有效时，RFPLL / XTAL / BG / RF ANA 时钟将根据*表 4-1. RF 时序中的时间*中的值按时序*图 4-4. RF 时序*自动关断；

当软件置位 WPWAKE 位或 Wi-Fi 硬件 wake_wl 信号有效时，RFPLL / XTAL / BG / RF ANA 时钟将根据*表 4-1. RF 时序中的时间*中的值按时序*图 4-4. RF 时序*自动打开。

- 若 RFSWEN 位为 1，选择软件方式配置 RF 时序：

RFPLL / XTAL / BG / RF ANA 时钟将在软件配置 RCU 相应寄存器位后打开或关断（建议按时序*图 4-4. RF 时序*进行配置，不配置则保持为之前的状态）。

$$t1 = t0 + T1 \quad (5\text{-}1)$$

表 4-1. RF 时序中的时间

名称	时间	描述
T1	HXTAL 模式：1ms。 外部供电模式（HXTAL 旁路模式）：1us。（注意外部供电的时钟已经就绪）	HXTAL 就绪时间
T2	1us - 10us，默认= 1us	BandGap 启动时间
T3	1us	-
T4	1us	上电间隔
T5	1us	上电间隔
T6	1us	-

T7	1us	HXTAL 模拟关闭输出时钟的预留时间

如果 HXTAL 被选择用于输出时钟，当 RCU_CTL 寄存器中的 HXTALREADY 和 HXTALEN 位都为 1 时，输出时钟将在一个 HXTAL 周期内稳定。

表 4-2. 节电模式总结

模式	睡眠	深度睡眠	待机	SRAM 睡眠	BLE 睡眠	Wi-Fi 睡眠
描述	仅关闭 CPU 时钟	1、关闭 1.1V 电源域的所有时钟 2、关闭 IRC16M、HXTAL 和 PLL	1、关闭 1.1V 电源域的供电 2、关闭 IRC16M、HXTAL 和 PLL	SRAM0/SRAM1/SRAM2/SRAM3 中至少一个掉电	BLE_OFF 域掉电	Wi-Fi_OFF 电源域掉电
LDO 状态	开启（正常功耗模式）	开启（正常功耗模式或者低功耗模式，正常驱动或者低驱动模式）	关闭	开启（正常功耗模式或者低功耗模式，正常驱动或者低驱动模式）	开启（正常功耗模式或者低功耗模式，正常驱动或者低驱动模式）	开启（正常功耗模式或者低功耗模式，正常驱动或者低驱动模式）
配置	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST = 1	SRAMxPSLEEP = 1（x = 0/1/2/3）	BLEPSLEEP = 1	1、当 WPEN = 1 时， WPSLEEP = 1 2、或者当 WPEN = 1 时，Wi-Fi 硬件 sleep_wl 信号有效
进入指令	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE	-	-	-
唤醒	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	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	1、NRST 引脚 2、WKUP 引脚 3、FWDGT 复位 4、RTC	SRAMxPWAKE = 1（x = 0/1/2/3）	BLEPWAKE = 1	1、当 WPEN = 1 时，置位 WPWAKE = 1 2、或者清 WPEN = 0 3、或者当 WPEN = 1 时，Wi-Fi 硬件 wake_wl 信号有效
唤醒延迟	无	IRC16M 唤醒时间 如果 LDO 处于低功耗模式，需增加 LDO 唤醒时间	上电序列	100ns	520ns	2100ns

4.4. PMU 寄存器

PMU 基地址：0x4000 7000

4.4.1. 控制寄存器 0（PMU_CTL0）

地址偏移：0x00
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												LDEN[1:0]		保留	
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				LDNP	LDLP	保留	BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	STBMOD	LDOLP
				rw	rw		rw	rw			rw	rc_w1	rc_w1	rw	rw

位/位域	名称	描述
31:20	保留	必须保持复位值。
19:18	LDEN[1:0]	深度睡眠模式下，低驱动模式使能 00：在深度睡眠模式下，禁用低驱动模式 01：保留 10：保留 11：在深度睡眠模式下，使能低驱动模式
17:12	保留	必须保持复位值。
11	LDNP	使用正常功耗 LDO 时，工作在低驱动模式 0：使用正常功耗 LDO 时，工作在正常驱动模式 1：使用正常功耗 LDO 且 LDEN 为 11 时，低驱动模式被使能
10	LDLP	使用低功耗 LDO 时，工作在低驱动模式 0：使用低功耗 LDO 时，工作在正常驱动模式 1：使用低功耗 LDO 且 LDEN 为 11 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：禁止付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
111：3.1V

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 注意：当 SYSCFG_CFG1 寄存器里的 LVD_LOCK 位被置 1 时，LVDEN 和 LVDT[2:0] 仅可读。
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	STBMOD	待机模式 0：当 RISC-V 进入 SLEEPDEEP 模式时，系统进入深度睡眠模式 1：当 RISC-V 进入 SLEEPDEEP 模式时，系统进入待机模式
0	LDOLP	LDO 低功耗模式 0：当系统进入深度睡眠模式时，LDO 仍正常工作 1：当系统进入深度睡眠模式时，LDO 进入低功耗模式

4.4.2. 电源控制和状态寄存器 0（PMU_CS0）

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												LDRF[1:0]		保留	
												rc_w1			

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

位/位域	名称	描述

31:20	保留	必须保持复位值。
19:18	LDRF[1:0]	低驱动模式就绪标志 在深度睡眠模式下，且 LDO 处于低驱动模式，这些位由硬件设置。软件对这些位写 11 可以清 0。 00：深度睡眠模式下，普通驱动模式 01：保留 10：保留 11：深度睡眠模式下，低驱动模式
17:15	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
13:12	保留	必须保持复位值。
11	WUPEN3	WKUP 引脚 3（PA12）唤醒使能 0：关闭 WKUP 引脚 3 唤醒功能 1：开启 WKUP 引脚 3 唤醒功能 如果 WUPEN3 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（PA7）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（PA15）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：收到来自 WKUP 引脚或 RTC 事件包括 RTC 闹钟事件，侵入事件，时间戳事件和自动唤醒事件 该位只能由 POR / PDR 或通过设置 PMU_CTL0 寄存器的 WURST 位来清零。

4.4.3. 控制寄存器 1（PMU_CTL1）

地址偏移：0x08

复位值：0x0024 0002（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						BLE_WAKEUP_REQ	BLEPWAKE	BLEPSLEEP	保留	BLE_SRAM_RET	SRAM0PWAKE	SRAM0PSLEEP	WIFI_SRAM_RET	WIFI_LPDS_ON	RETDIS
						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	SRAM3PWAKE	SRAM3PSLEEP	保留		SRAM2PWAKE	SRAM2PSLEEP	保留		SRAM1PWAKE	SRAM1PSLEEP	保留	WPWAKE	WPSLEEP	WPEN	保留
	rw	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	BLE_WAKEUP_REQ	BLE 唤醒请求 复位值为 0。
24	BLEPWAKE	软件置位该位将唤醒 BLE。硬件清零该位。
23	BLEPSLEEP	软件置位该位 BLE 将进入睡眠模式。硬件清零该位。
22	保留	必须保持复位值。
21	BLE_SRAM_RET	当深度睡眠和 BLE 睡眠模式时，BLE SRAM 进入保持模式。 0：失能保持模式

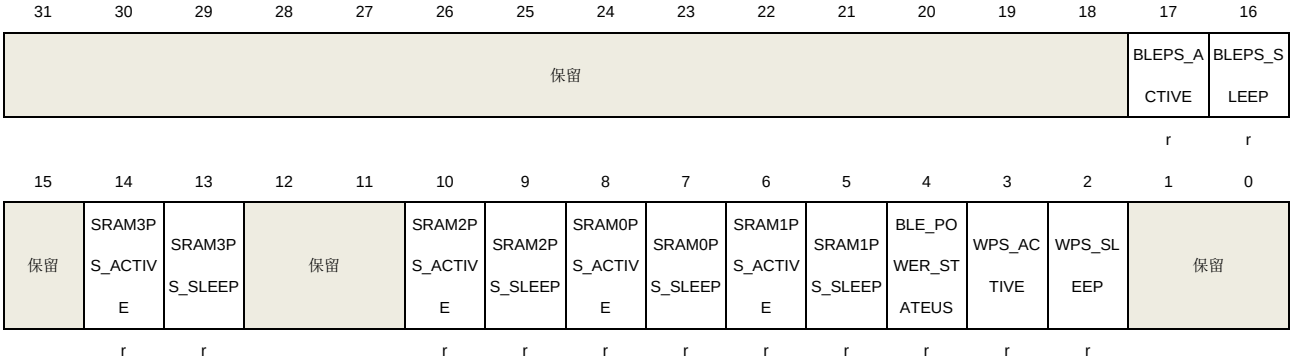
		1：使能保持模式
20	SRAM0PWAKE	软件置位该位将唤醒 SRAM0。硬件清零该位。
19	SRAM0PSLEEP	软件置位该位 SRAM0 将进入睡眠模式。硬件清零该位。
18	WIFI_SRAM_RET	当深度睡眠和 Wi-Fi 睡眠模式时，Wi-Fi SRAM 进入保持模式。 0：失能保持模式 1：使能保持模式
17	WIFI_LPDS_ON	Wi-Fi 自动低功耗深度睡眠 0：退出 Wi-Fi 自动低功耗深度睡眠 1：启动 Wi-Fi 自动低功耗深度睡眠
16	RETDIS	当 Wi-Fi 掉电时无保持寄存器 0：使能保持模式 1：失能保持模式
15	保留	必须保持复位值。
14	SRAM3PWAKE	软件置位该位将唤醒 SRAM3。硬件清零该位。
13	SRAM3PSLEEP	软件置位该位 SRAM3 将进入睡眠模式。硬件清零该位。
12:11	保留	必须保持复位值。
10	SRAM2PWAKE	软件置位该位将唤醒 SRAM2。硬件清零该位。
9	SRAM2PSLEEP	软件置位该位 SRAM2 将进入睡眠模式。硬件清零该位。
8:7	保留	必须保持复位值。
6	SRAM1PWAKE	软件置位该位将唤醒 SRAM1。硬件清零该位。
5	SRAM1PSLEEP	软件置位该位 SRAM1 将进入睡眠模式。硬件清零该位。
4	保留	必须保持复位值。
3	WPWAKE	Wi-Fi 唤醒 仅当 WPEN 位为 1 时，软件置位该位将唤醒 Wi-Fi。也可通过清零 WPEN 位（从 1 变为 0）来唤醒 Wi-Fi。硬件清零该位。
2	WPSLEEP	仅当 WPEN 位为 1 时，软件置位该位将进入 Wi-Fi 睡眠状态。硬件清零该位。
1	WPEN	Wi-Fi电源使能（复位值：1）（当开关该位后需要对Wi-Fi进行全局复位） 0：Wi-Fi_OFF电源域上电。若Wi-Fi_OFF电源域处于掉电状态，清零该位可以唤醒 Wi-Fi（注意IRC16M需要处于运行状态）。 1：在Wi-Fi睡眠状态时，Wi-Fi_OFF电源域掉电；在Wi-Fi唤醒时，Wi-Fi_OFF电源域上电（当该位为1时，Wi-Fi可由软件或硬件方式上下电）。
0	保留	必须保持复位值。

4.4.4. 电源控制和状态寄存器 1（PMU_CS1）

地址偏移：0x0C

复位值：0x8000 0010

该寄存器可以按半字（16 位）或字（32 位）访问。



位/位域	名称	描述
31:18	保留	必须保持复位值。
17	BLEPS_ACTIVE	BLE 处于运行状态。只读。
16	BLEPS_SLEEP	BLE 处于睡眠状态。只读。
15	保留	必须保持复位值。
14	SRAM3PS_ACTIVE	SRAM3 处于运行状态。只读。
13	SRAM3PS_SLEEP	SRAM3 处于睡眠状态。只读。
12:11	保留	必须保持复位值。
10	SRAM2PS_ACTIVE	SRAM2 处于运行状态。只读。
9	SRAM2PS_SLEEP	SRAM2 处于睡眠状态。只读。
8	SRAM0PS_ACTIVE	SRAM0 处于运行状态。只读。
7	SRAM0PS_SLEEP	SRAM0 处于睡眠状态。只读。
6	SRAM1PS_ACTIVE	SRAM1 处于运行状态。只读。
5	SRAM1PS_SLEEP	SRAM1 处于睡眠状态。只读。
4	BLE_POWER_STAT	BLE 上电或掉电 0：BLE 可以进入 SLEEP 1：BLE 可以唤醒
3	WPS_ACTIVE	Wi-Fi 处于运行状态。只读。
2	WPS_SLEEP	Wi-Fi 处于睡眠状态。只读。

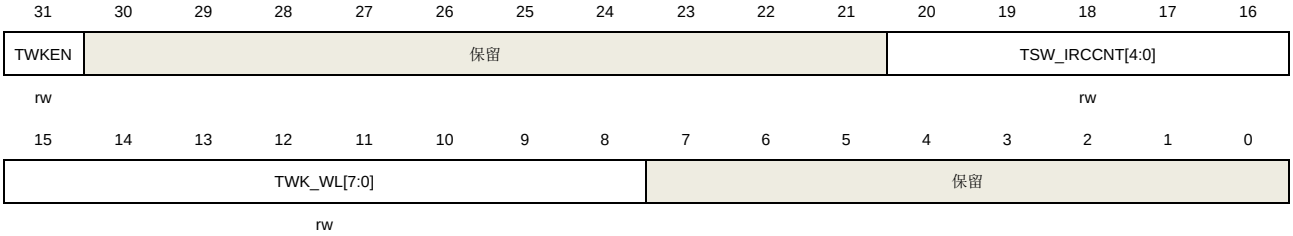
1:0	保留	必须保持复位值。

4.4.5. 参数寄存器 0（PMU_PAR0）

地址偏移：0x10

复位值：0x000A 2000

该寄存器可以按半字（16 位）或字（32 位）访问。



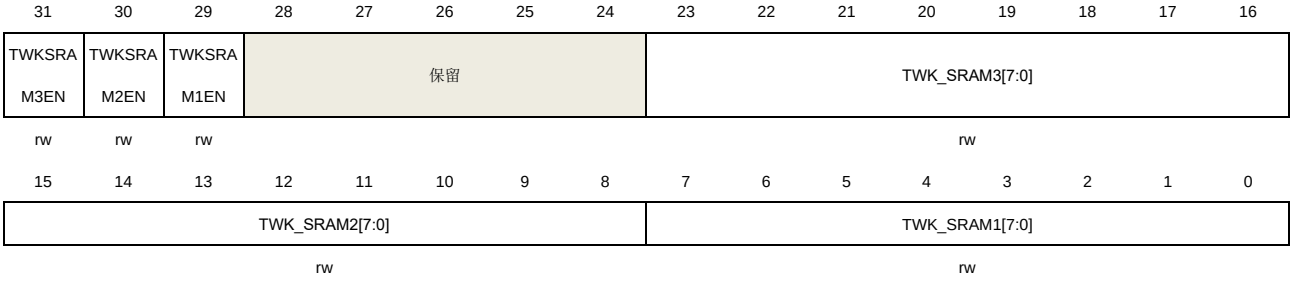
位/位域	名称	描述
31	TWKEN	唤醒 Wi-Fi_OFF 时是否使用用户软件值 0：当唤醒 Wi-Fi_OFF 时使用硬件 ack 信号。 1：当唤醒 Wi-Fi_OFF 时使用软件值，软件的值在 TWK_WL 中设置。
30:21	保留	必须保持复位值。
20:16	TSW_IRCCNT[4:0]	当进入深度睡眠模式，切换到 IRC16M 时钟。默认值为 22 个 IRC16M 时钟。 0x00-0x11：22 个 IRC16M 时钟 0x12：23 个 IRC16M 时钟 0x13：24 个 IRC16M 时钟 0x14：25 个 IRC16M 时钟 0x15：26 个 IRC16M 时钟 0x16：27 个 IRC16M 时钟 0x17：28 个 IRC16M 时钟 0x18：29 个 IRC16M 时钟 0x19：30 个 IRC16M 时钟 0x1A：31 个 IRC16M 时钟 0x1B：32 个 IRC16M 时钟 0x1C：33 个 IRC16M 时钟 0x1D：34 个 IRC16M 时钟 0x1E：35 个 IRC16M 时钟 0x1F：36 个 IRC16M 时钟
15:8	TWK_WL[7:0]	Wi-Fi_OFF 域的唤醒时间。步长为 1 个时钟，最大值为 64us。
7:0	保留	必须保持复位值。

4.4.6. 参数寄存器 1（PMU_PAR1）

地址偏移：0x14

复位值：0x0020 2020

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	TWKSRAM3EN	唤醒 SRAM3 时是否使用用户软件值 0：当唤醒 SRAM3 时使用硬件 ack 信号。 1：当唤醒 SRAM3 时使用软件值，软件的值在 TWK_SRAM3 中设置。
30	TWKSRAM2EN	唤醒 SRAM2 时是否使用用户软件值 0：当唤醒 SRAM2 时使用硬件 ack 信号。 1：当唤醒 SRAM2 时使用软件值，软件的值在 TWK_SRAM2 中设置。
29	TWKSRAM1EN	唤醒 SRAM1 时是否使用用户软件值 0：当唤醒 SRAM1 时使用硬件 ack 信号。 1：当唤醒 SRAM1 时使用软件值，软件的值在 TWK_SRAM1 中设置。
28:24	保留	必须保持复位值。
23:16	TWK_SRAM3[7:0]	SRAM3 域的唤醒时间。步长为 4 个时钟，最大值为 64 us。
15:8	TWK_SRAM2[7:0]	SRAM2 域的唤醒时间。步长为 4 个时钟，最大值为 64 us。
7:0	TWK_SRAM1[7:0]	SRAM1 域的唤醒时间。步长为 4 个时钟，最大值为 64 us。

4.4.7. 参数寄存器 2（PMU_PAR2）

地址偏移：0x18

复位值：0x0000 2020

该寄存器可以按半字（16 位）或字（32 位）访问。

rw rw

位/位域	名称	描述
31	TWKBLEEN	唤醒 BLE 时是否使用用户软件值 0：当唤醒 BLE 时使用硬件 ack 信号。 1：当唤醒 BLE 时使用软件值，软件的值在 TWK_BLE 中设置。
30	TWKSRAM0EN	唤醒 SRAM0 时是否使用用户软件值 0：当唤醒 SRAM0 时使用硬件 ack 信号。 1：当唤醒 SRAM0 时使用软件值，软件的值在 TWK_SRAM0 中设置。
29:16	保留	必须保持复位值。
15:8	TWK_BLE[7:0]	BLE 域的唤醒时间。步长为 4 个时钟，最大值为 64 us。
7:0	TWK_SRAM0[7:0]	SRAM0 域的唤醒时间。步长为 4 个时钟，最大值为 64 us。

4.4.8. RF 控制寄存器（PMU_RFCTL）

地址偏移：0x20

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									MCU_STAE[2:0]			RF_STATE[3:0]			
										r			r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										BLESWEN	MCU_PLLDOWN	MCU_PLLUP	RFFC	RFFS	RFSWEN
										rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值。
22:20	MCU_STATE[2:0]	MCU 状态 000：MCU_IDLE 011：MCU_RUN 其他值：保留
19:16	RF_STATE[3:0]	RF 状态 0000：IDLE 1000：RFRUN 其他值：保留
15:6	保留	必须保持复位值。
5	BLESWEN	0：蓝牙硬件打开 RF

		1：软件控制 RF 打开，RF 不受蓝牙打开关闭影响
4	MCU_PLLDOWN	软件置位或清零。 软件强制关闭 MCU PLL 电源。当无线处于工作状态时，设置此位无效。
3	MCU_PLLUP	软件置位或清零（必须大于 2 个 IRC16M 时钟）。硬件清零。 软件强制打开 MCU PLL 电源。
2	RFFC	软件置位或清零（必须大于 2 个 IRC16M 时钟）。硬件清零。 软件强制关闭 RF 电源，强制进行硬件 RF 时序关断过程。
1	RFFS	软件置位或清零（必须大于 2 个 IRC16M 时钟）。硬件清零。 软件强制打开 RF 电源，强制进行硬件 RF 时序打开过程。
0	RFSWEN	0：硬件自动根据时序图*图 4-4. RF 时序*（默认）来配置 RF 时序 1：软件配置 RF 时序

4.4.9. RF 时序参数寄存器（PMU_RFPAR）

地址偏移：0x24

复位值：0x472A 0164

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		TIM7_PAR[2:0]		TIM89_PAR[3:0]				保留		TIM5_PAR[1:0]		TIM4_PAR[1:0]		TIM3_PAR[1:0]	
		rw		rw						rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				TIM2_PAR[3:0]				保留	TIM1_PAR[6:0]						
				rw					rw						

位/位域	名称	描述
31	保留	必须保持复位值。
30:28	TIM7_PAR[2:0]	步长 250ns，默认值 1us，最大值 2us。
27:24	TIM89_PAR[3:0]	步长 0.1us，默认值 0.5 us，最大值 1us。
23:22	保留	必须保持复位值。
21:20	TIM5_PAR[1:0]	步长 0.5us，默认值 1us，最大值 1.5us。
19:18	TIM4_PAR[1:0]	步长 0.5us，默认值 1us，最大值 1.5us。
17:16	TIM3_PAR[1:0]	步长 0.5us，默认值 1us，最大值 1.5us。
15:12	保留	必须保持复位值。
11:8	TIM2_PAR[3:0]	步长 1us，默认值 1us，最大值 16us。
7	保留	必须保持复位值。

6:0	TIM1_PAR[6:0]	步长 0.1ms，默认值 1ms，最大值 12.7ms。 XTAL 旁路模式：步长 125ns，默认值 1.25us。

4.4.10. PMU 中断标志寄存器（PMU_INTF）

地址偏移：0x2C
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														BLE_PS_RISEF	BLE_PS_FALLF
														r	r

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	BLE_PS_RISEF	0：BLE_POWER_STATUS 被检测到时没有边沿变化。 1：检测到 BLE_POWER_STATUS 的上升边沿，这表示 BLE 请求唤醒。
0	BLE_PS_FALLF	0：BLE_POWER_STATUS 被检测到时没有边沿变化。 1：检测到 BLE_POWER_STATUS 的下降边沿，这表示 BLE 请求唤醒。

4.4.11. PMU 中断使能寄存器（PMU_INTEN）

地址偏移：0x30
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														BLE_PS_RISE_EN	BLE_PS_FALL_EN
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	BLE_PS_RISE_EN	0：失能 BLE_PS_RISE 中断。 1：使能 BLE_PS_RISE 中断。

0	BLE_PS_FALL_EN	0：失能 BLE_PS_FALL 中断。 1：使能 BLE_PS_FALL 中断。

4.4.12. PWR 中断清除寄存器（PMU_INTC）

地址偏移：0x34

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														BLE_PS_RISEC	BLE_PS_FALLC
														w	w

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	BLE_PS_RISEFC	设置该位为 1 清除 BLE_PS_RISEF。 读该位返回 0。
0	BLE_PS_FALLFC	设置该位为 1 清除 BLE_PS_FALLF。 读该位返回 0。

5. 复位和时钟单元（RCU）

5.1. 复位控制单元（RCTL）

5.1.1. 简介

GDGD32VW55x 复位控制包括三种控制方式：电源复位、系统复位和备份域复位。电源复位又称为冷复位，其复位所有系统。系统复位将复位除了 JTAG 控制器和备份域之外的其余部分，包括处理器内核和外设 IP。备份域复位将复位备份区域。这些复位能够被外部信号、内部事件和复位发生器触发。后续章节将介绍关于这些复位的详细信息。

5.1.2. 功能描述

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。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的地址 0x0000_0004。

系统复位

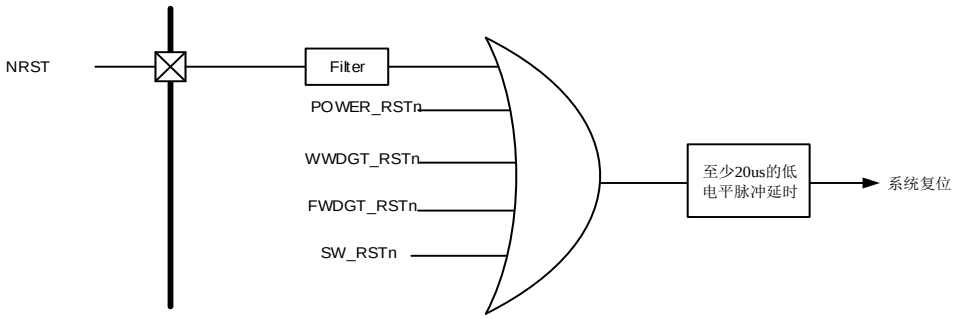
当发生以下任一事件时，产生一个系统复位：

- 上电复位（POWER_RSTn）；
- 外部引脚复位（NRST）；
- 窗口看门狗计数终止（WWDGT_RSTn）；
- 独立看门狗计数终止（FWDGT_RSTn）；
- RISC-V的中断应用和复位控制寄存器中的SYSRESETREQ位置‘1’（SW_RSTn）。

系统复位将复位除了 JTAG 控制器和备份域之外的其余部分，包括处理器内核和外设 IP。

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 20 us 的低电平脉冲延时。

图 5-1. 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（在 V_{DD} 掉电的前提下，V_{DD} 上电）。

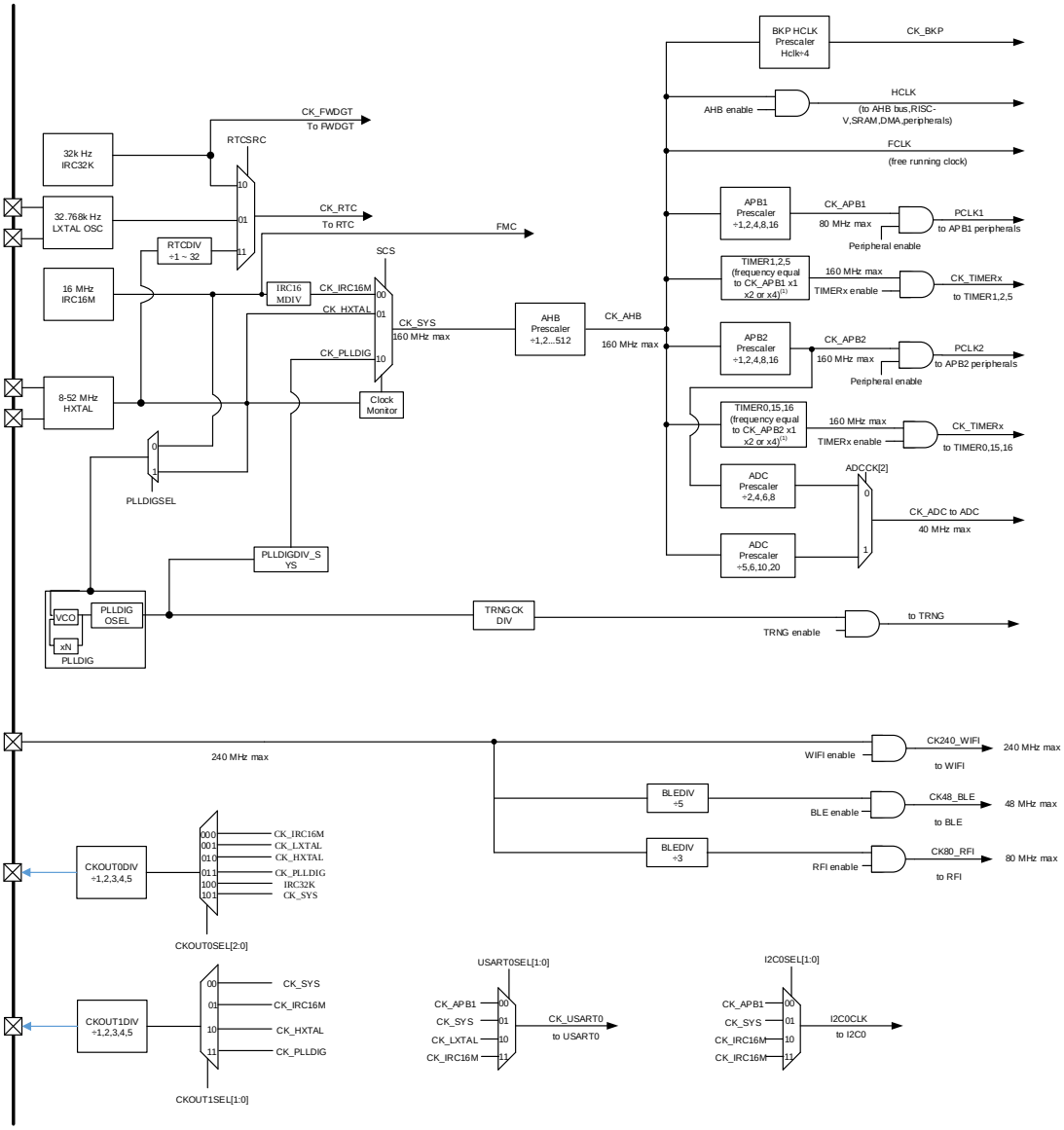
5.2. 时钟控制单元（CCTL）

5.2.1. 简介

时钟控制单元提供了一系列频率的时钟功能，包括一个内部 16 MHz RC 振荡器时钟（IRC16M）、一个外部高速晶体振荡器时钟（HXTAL）、一个内部 32 KHz RC 振荡器时钟（IRC32K）、一个外部低速晶体振荡器时钟（LXTAL）、一个数字锁相环（PLLDIG）、一个 HXTAL 时钟监视器、时钟预分频器、时钟多路复用器和时钟门控电路。

AHB、APB 和 RISC-V 时钟都源自系统时钟（CK_SYS），系统时钟的时钟源可以选择 IRC16M、HXTAL 或 PLLDIG。系统时钟的最大运行时钟频率可以达到 160 MHz。独立看门狗定时器有独立的时钟源（IRC32K），实时时钟（RTC）使用 IRC32K、LXTAL 或 HXTAL 的分频（通过配置 RCU_CFG0 寄存器的 RTCDIV 位）作为时钟源。

图 5-2. 时钟树



（1）详细信息请参考 RCU_CFG1 寄存器的 TIMERSEL 位。

预分频器可以配置 AHB、APB2 和 APB1 域的时钟频率。AHB 和 APB2 / APB1 域的最高时率分别为 160 MHz / 160 MHz / 80 MHz。RCU 通过 AHB 时钟（HCLK）8 分频后作为 RISC-V 系统定时器（SysTick）的外部时钟。通过对 SysTick 控制和状态寄存器的设置，可选择上述时钟或 AHB（HCLK）时钟作为 SysTick 时钟。

ADC 时钟由 APB2 时钟经 2、4、6、8 分频或由 AHB 时钟经 5、6、10、20 分频获得，它们是通过设置 ADC_CCTL 寄存器的 ADCCK 位来选择。

TIMER 时钟由 AHB 时钟分频获得，它的频率可以等于 CK_APBx、CK_APBx 的两倍或 CK_APBx 的四倍。详细信息请参考 RCU_CFG1 寄存器的 TIMERSEL 位。

TRNG 的时钟从 PLLDIG 的时钟中选择。

通过配置RCU_BDCTL寄存器的RTCSRC位，RTC时钟可以选择由LXTAL时钟、IRC32K时钟或HXTAL时钟的1 - 32（由RCU_CFG0寄存器的RTCDIV位域值决定）分频提供。RTC时钟选

择HXTAL时钟的分频做为时钟源后，当1.1V内核电压域掉电时，时钟将停止。RTC时钟选择IRC32K时钟做为时钟源后，当V_{DD}掉电时，时钟将停止。RTC时钟选择LXTAL时钟做为时钟源后，当V_{DD}掉电时，时钟将停止。

当 FWDGT 启动时，FWDGT 时钟被强制选择由 IRC32K 时钟做为时钟源。

BLE 时钟源是 240 MHz RF 时钟经过 5 或 3 分频后输出 48 MHz 或 80 MHz。

通过配置 RCU_CFG1 寄存器的 USART0SEL 位，USART0 的时钟由 IRC16M 时钟或者 LXTAL 时钟或者系统时钟或者 APB1 时钟提供。

通过配置 RCU_CFG1 寄存器的 I2C0SEL 位，I2C0 的时钟由 IRC16M 时钟或者系统时钟或者 APB1 时钟提供。

5.2.2. 主要特性

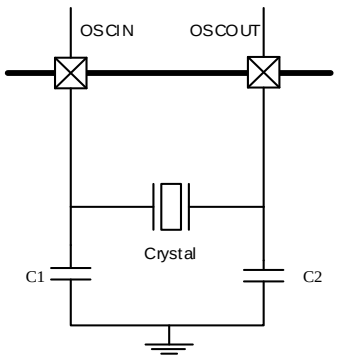
- 8到52 MHz外部高速晶体振荡器（HXTAL）。
- 内部16 MHz RC振荡器（IRC16M）。
- 32,768 Hz外部低速晶体振荡器（LXTAL）。
- 内部32 KHz RC振荡器（IRC32K）。
- PLLDIG时钟源可选HXTAL或IRC16M。
- HXTAL时钟监视器。

5.2.3. 功能描述

外部高速晶体振荡器时钟（HXTAL）

8 到 52 MHz 的外部高速晶体振荡器可为系统时钟提供更为精确时钟源。带有特定频率的晶体必须靠近两个 HXTAL 的引脚连接。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 5-3. HXTAL 时钟源

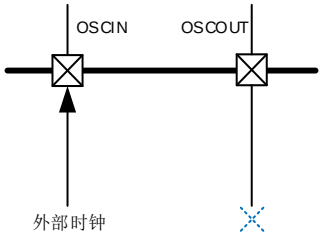


HXTAL 晶体振荡器可以通过设置控制寄存器 RCU_CTL 的 HXTALEN 位来启动或关闭，在控制寄存器 RCU_CTL 中的 HXTALSTB 位用来指示外部高速振荡器是否已稳定。在启动时，直到这一位被硬件置‘1’，时钟才被释放出来。这个特定的延迟时间被称为振荡器的启动时间。当 HXTAL 时钟稳定后，如果在中断寄存器 RCU_INT 中的相应中断使能位 HXTALSTBIE 位被置

'1'，将会产生相应中断。此时，HXTAL 时钟可以被直接用作系统时钟源或者 PLLDIG 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5-4. 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 5-4. 旁路模式下 HXTAL 时钟源



内部 16M RC 振荡器时钟（IRC16M）

内部 16 MHz RC 振荡器时钟，简称 IRC16M 时钟，拥有 16 MHz 的固定频率，设备上电后 CPU 默认选择其做为系统时钟源。IRC16M RC 振荡器能够在不需要任何外部器件的条件下为用户提供更低成本类型的时钟源。IRC16M RC 振荡器可以通过设置控制寄存器（RCU_CTL）中的 IRC16MEN 位被启动和关闭。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 时钟也可用作系统时钟源或 PLLDIG 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。用户可以根据需求、环境条件和成本决定选择哪个时钟作为系统时钟源。

如果 HXTAL 或者 PLLDIG 被选择为系统时钟源，为了最大程度减小系统从深度睡眠模式恢复的时间，当系统从深度睡眠模式初始唤醒时，硬件会强制 IRC16M 时钟作为系统时钟。

如果 USART0 和 I2C0 选择 IRC16M 作为时钟，则 IRC16M 将被强制选择为 USART0 和 I2C0 的时钟。

数字锁相环（PLLDIG）

存在一个内部数字锁相环 PLLDIG。PLLDIG 时钟可做为系统时钟（不超过 160 MHz），PLLDIG 的分频时钟可以做为 TRNG 模块的时钟源。

PLLDIG 可以通过设置 RCU_CTL 寄存器中的 PLLDIGEN / PLLDIGPU 位被启动和关闭。RCU_CTL 寄存器中的 PLLDIGSTB 位用来指示 PLLDIG 时钟是否稳定。如果 RCU_INT 寄存器中的相应中断使能位 PLLDIGSTBIE 被置'1'，在 PLLDIG 稳定以后，将产生一个中断。

当进入 Deepsleep / Standby 模式或者 HXTAL 监视器检测到时钟阻塞时（HXTAL 做为锁相环的输入时钟），PLLDIG 将被关闭。

外部低速晶体振荡器时钟（LXTAL）

LXTAL是一个频率为32.768 KHz的外部低速晶体或陶瓷谐振器。它为实时时钟电路提供一个低功耗且高精准的时钟源。LXTAL振荡器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。

内部 32K RC 振荡器时钟（IRC32K）

IRC32K 内部 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 32 KHz，为独立看门狗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。如果复位源 / 时钟寄存器 RCU_RSTSCK 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 时钟默认做为 CK_SYS 的时钟源，改变配置寄存器 0（RCU_CFG0）中的系统时钟变换位 SCS 可以切换系统时钟源为 HXTAL 或 CK_PLLDIG。当 SCS 的值被改变，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 PLLDIG 间接作为系统时钟时，它将不能被停止。

HXTAL 时钟监视器（CKM）

设置控制寄存器 RCU_CTL 中的 HXTAL 时钟监视使能位 RFCKMEN，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，中断寄存器 RCU_INT 中的 HXTAL 时钟阻塞中断标志位 CKMIF 将被置‘1’，产生 HXTAL 故障事件。这个故障引发的中断和 RISC-V 的不可屏蔽中断 NMI 相连。如果 HXTAL 被选作系统或 PLLDIG 的时钟源，HXTAL 故障将促使选择 IRC16M 为系统时钟源且 PLLDIG 将被自动禁止。

时钟输出功能

时钟输出功能输出从 32 KHz 到 160 MHz 的时钟。通过设置时钟配置寄存器 0（RCU_CFG0）中的 CK_OUT0 时钟源选择位域 CKOUT0SEL 能够选择不同的时钟信号。相应的 GPIO 引脚应该被配置成备用功能 I/O（AFIO）模式来输出选择的时钟信号。CK_OUT1 时钟输出源选择通过设置时钟配置寄存器 RCU_CFG0 中的 CKOUT1SEL 位域实现。

表 5-1. 时钟输出 0 的时钟源选择

时钟输出 0 的时钟源选择位域	时钟源
000	CK_IRC16M

时钟输出 0 的时钟源选择位域	时钟源
001	CK_LXTAL
010	CK_HXTAL
011	CK_PLLDIG
100	CK_IRC32K
101	CK_SYS
110	保留
111	保留

表 5-2. 时钟输出 1 的时钟源选择

时钟输出 1 的时钟源选择位域	时钟源
00	CK_SYS
01	CK_IRC16M
10	CK_HXTAL
11	CK_PLLDIG

通过配置 RCU_CFG0 寄存器的 CKOUT0DIV 位域，可以将 CK_OUT0 输出时钟的频率按比例分频，进而降低 CK_OUT0 的输出频率。

通过配置 RCU_CFG0 寄存器的 CKOUT1DIV 位域，可以将 CK_OUT1 输出时钟的频率按比例分频，进而降低 CK_OUT1 的输出频率。

电压控制

深度睡眠模式电压寄存器（RCU_DSV）中的 DSLPVS[1:0]位域可以控制 1.1V 域在深度睡眠模式下的电压。

表 5-3. 深度睡眠模式下 1.1V 域电压选择

DSLPVS[1:0]	深度睡眠模式电压（V）
00	1.1
01	1.0
10	0.9
11	0.9

RCU_DSV 寄存器被电源解锁寄存器（RCU_VKEY）保护。只有在写 0x1A2B3C4D 到 RCU_VKEY 后，RCU_DSV 寄存器才能被写入。

5.3. RCU 寄存器

RCU 访问基地址：0x4002 3800

5.3.1. 控制寄存器（RCU_CTL）

地址偏移：0x00

复位值：0x0040 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
HXTALREADY	保留		HXTALPU	保留				PLLDIGSTB	RFCKMEN	PLLDIGEN	PLLDIGPU	保留	HXTALBPS	HXTALSTB	HXTALEN
rw			rw					r	rw	rw	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]					IRC16MFON	IRC16MSTB	IRC16MEN
r								rw					rw	r	rw

位/位域	名称	描述
31	HXTALREADY	通过软件设置高速晶体振荡器（HXTAL）稳定，在HXTALEN关闭时写入。 0：未设置HXTAL稳定 1：已设置HXTAL稳定
30:29	保留	必须保持复位值。
28	HXTALPU	通过软件设置高速晶体振荡器（HXTAL）上电，在HXTALEN关闭时写入。 软件置位或复位，当进入深度睡眠或待机模式时由硬件复位。 0：高速晶体振荡器掉电 1：高速晶体振荡器上电
27:24	保留	必须保持复位值。
23	PLLDIGSTB	PLLDIG时钟稳定标志位 硬件置1来表示PLLDIG输出时钟是否稳定待用。 0：PLLDIG未稳定 1：PLLDIG已稳定
22	RFCKMEN	HXTAL时钟监视器使能，检查RF XTAL 0：禁用高速晶体振荡器（HXTAL）时钟监视器 1：使能高速晶体振荡器（HXTAL）时钟监视器
21	PLLDIGEN	软件置位或复位，如果PLLDIGEN时钟作为系统时钟，该位不能被复位。 软件置位或复位，进入深度睡眠或待机模式时硬件自动复位。 0：PLLDIG时钟被打开

		1：PLLDIG时钟被关闭
20	PLLDIGPU	PLLDIG上电，如果PLLDIGEN时钟作为系统时钟，该位不能被复位。 软件置位或复位，进入深度睡眠或待机模式时硬件自动复位。 0：PLLDIG掉电 1：PLLDIG上电
19	保留	必须保持复位值。
18	HXTALBPS	高速晶体振荡器（HXTAL）时钟旁路模式使能 只有在HXTALEN和HXTALPU位都为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	高速晶体振荡器（HXTAL）时钟稳定标志位 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	高速晶体振荡器（HXTAL）使能 软件置位或复位，如果 HXTAL 时钟作为系统时钟，或者当 PLLDIG 时钟作为系统时钟，且 HXTAL 作为 PLLDIG 的输入时钟时，该位不能被关闭。如果 HXTAL 时钟作为系统时钟时，硬件限制不能清除该位。当 PLLDIG 时钟作为系统时钟时，且 HXTAL 作为 PLLDIG 的输入时钟时，如果该位被强制关闭，系统时钟会自动跳转到 IRC16M 的系统时钟。进入深度睡眠或待机模式时硬件自动复位。当 CKM 检测失败时，硬件自动复位。 0：高速晶体振荡器被关闭 1：高速晶体振荡器被打开
15:8	IRC16MCALIB[7:0]	内部16MHz RC振荡器校准值寄存器 上电时自动加载这些位。
7:3	IRC16MADJ[4:0]	内部16 MHz RC振荡器时钟调整值 这些位由软件置位，最终调整值为IRC16MADJ[4:0]位域的当前值加上IRC16MCALIB[7:0]位域的值。最终调整值应该调整IRC16M到16 MHz ± 1%。
2	IRC16MRFON	内部16 MHz RC RF差分时钟信号使能 软件置位或复位。该位只有当IRC16MEN置位时才能置位。 0：内部16 MHz RC RF差分时钟关闭 1：内部16 MHz RC RF差分时钟使能
1	IRC16MSTB	IRC16M内部16 MHz RC振荡器稳定标志位 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	内部16 MHz RC振荡器使能 软件置位或复位，如果IRC16M时钟做为系统时钟时，该位不能被复位。当从深度

睡眠或待机模式返回，或当CKMEN置位同时用作系统时钟的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振荡器被关闭

1：内部16 MHz RC振荡器被打开

5.3.2. PLL 寄存器（RCU_PLL）

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。

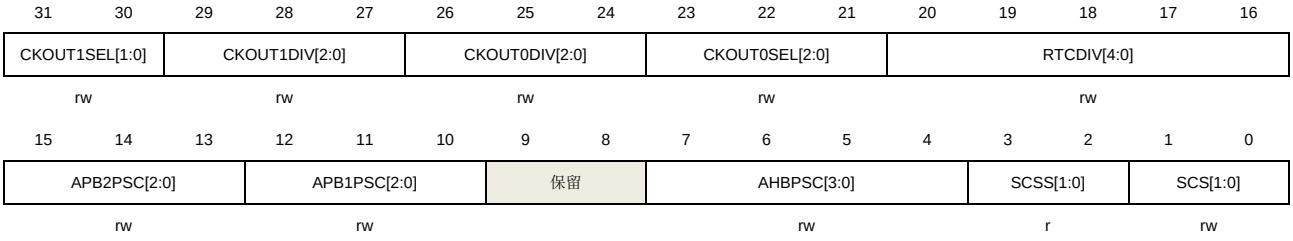
位/位域	名称	描述
31:16	保留	必须保持复位值。
15	PLLDIGSEL	PLLDIG时钟源选择 由软件置位或清零，控制PLLDIG时钟源。 0：选择IRC16M输出时钟作为PLLDIG源时钟 1：选择HXTAL时钟作为PLLDIG源时钟
14:0	保留	必须保持复位值。

5.3.3. 时钟配置寄存器 0（RCU_CFG0）

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。



位/位域	名称	描述

31:30	CKOUT1SEL[1:0]	CKOUT1时钟源选择 由软件置位或清零。 00：选择系统时钟 01：选择内部16 MHz RC振荡器时钟 10：选择高速晶体振荡器时钟（HXTAL） 11：选择CK_PLLDIG时钟
29:27	CKOUT1DIV[2:0]	CK_OUT1分频器，来降低CK_OUT1频率 CK_OUT1时钟源的选择参考RCU_CFG0寄存器的31:30位。 0xx：CK_OUT1不分频 100：CK_OUT1被2分频 101：CK_OUT1被3分频 110：CK_OUT1被4分频 111：CK_OUT1被5分频
26:24	CKOUT0DIV[2:0]	CK_OUT0分频器，来降低CK_OUT0频率 CK_OUT0时钟源的选择参考RCU_CFG0寄存器的23:21位。 0xx：CK_OUT0不分频 100：CK_OUT0被2分频 101：CK_OUT0被3分频 110：CK_OUT0被4分频 111：CK_OUT0被5分频
23:21	CKOUT0SEL[2:0]	CKOUT0时钟源选择 由软件置位或清零。 000：选择内部16 MHz RC振荡器时钟 001：选择低速晶体振荡器时钟（LXTAL） 010：选择高速晶体振荡器时钟（HXTAL） 011：选择数字锁相环时钟（PLLDIG） 100：选择内部32 KHz RC振荡器I时钟 101：选择系统时钟 110 / 111：保留
20:16	RTCDIV[4:0]	RTC时钟分频系数 由软件置位或清零。这些位用作将HXTAL时钟分频生成RTC时钟（不超过1MHz）。 00000：CK_HXTAL / 1 00001：CK_HXTAL / 2 00010：CK_HXTAL / 3 00011：CK_HXTAL / 4 … 1111：CK_HXTAL / 32
15:13	APB2PSC[2:0]	APB2预分频选择 由软件置位或清零，控制APB2时钟分频因子。 0xx：选择CK_AHB时钟不分频

		100：选择CK_AHB时钟2分频 101：选择CK_AHB时钟4分频 110：选择CK_AHB时钟8分频 111：选择CK_AHB时钟16分频
12:10	APB1PSC[2:0]	APB1预分频选择 由软件置位或清零，控制APB1时钟分频因子。 0xx：选择CK_AHB时钟不分频 100：选择CK_AHB时钟2分频 101：选择CK_AHB时钟4分频 110：选择CK_AHB时钟8分频 111：选择CK_AHB时钟16分频
9:8	保留	必须保持复位值。
7:4	AHBPSC[3:0]	AHB预分频选择 由软件置位或清零，控制AHB时钟分频因子。 0xxx：选择CK_SYS时钟不分频 1000：选择CK_SYS时钟2分频 1001：选择CK_SYS时钟4分频 1010：选择CK_SYS时钟8分频 1011：选择CK_SYS时钟16分频 1100：选择CK_SYS时钟64分频 1101：选择CK_SYS时钟128分频 1110：选择CK_SYS时钟256分频 1111：选择CK_SYS时钟512分频
3:2	SCSS[1:0]	系统时钟选择状态 由硬件置位或清零，标识当前系统时钟的时钟源。 00：选择CK_IRC16M时钟作为CK_SYS时钟源 01：选择CK_HXTAL时钟作为CK_SYS时钟源 10：选择CK_PLLDIG时钟作为CK_SYS时钟源 11：无时钟作为CK_SYS时钟源
1:0	SCS[1:0]	系统时钟选择 由软件配置选择系统时钟源。由于CK_SYS的改变存在固有的延迟，因此软件应当读SCSS位来确保时钟源切换是否结束。在从深度睡眠或待机模式中返回时，以及当HXTAL直接或间接作为系统时钟同时HXTAL时钟监视器检测到HXTAL故障时，强制选择IRC16M作为系统时钟。 00：选择CK_IRC16M时钟作为CK_SYS时钟源 01：选择CK_HXTAL时钟作为CK_SYS时钟源 10：选择CK_PLLDIG时钟作为CK_SYS时钟源 11：无时钟作为CK_SYS时钟源

5.3.4. 时钟中断寄存器（RCU_INT）

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								CKMIC	PLLDIGSTBIC	保留		HXTALSTBIC	IRC16MSTBIC	LXTALSTBIC	IRC32KSTBIC
								w	w			w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLLDIGSTBIE	保留		HXTALSTBIE	IRC16MSTBIE	LXTALSTBIE	IRC32KSTBIE	CKMIF	PLLDIGSTBIF	保留		HXTALSTBIF	IRC16MSTBIF	LXTALSTBIF	IRC32KSTBIF
	rw			rw	rw	rw	rw	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时钟阻塞中断清零 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	PLLDIGSTBIC	PLLDIG时钟稳定中断清零 软件写1复位PLLDIGSTBIF标志位。 0：不复位PLLDIGSTBIF标志位 1：复位PLLDIGSTBIF标志位
21:20	保留	必须保持复位值。
19	HXTALSTBIC	HXTAL时钟稳定中断清零 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时钟稳定中断清零 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时钟稳定中断清零 软件写1复位LXTALSTBIF标志位。 0：不复位LXTALSTBIF标志位 1：复位LXTALSTBIF标志位
16	IRC32KSTBIC	IRC32K时钟稳定中断清零 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	PLLDIGSTBIE	PLLDIG时钟稳定中断使能 软件置位和复位来使能 / 禁止PLLDIG时钟稳定中断。 0：禁止PLLDIG时钟稳定中断 1：使能PLLDIG时钟稳定中断
13:12	保留	必须保持复位值。
11	HXTALSTBIE	HXTAL时钟稳定中断使能 软件置位和复位来使能 / 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 软件置位和复位来使能 / 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 软件置位和复位来使能 / 禁止LXTAL时钟稳定中断。 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 软件置位和复位来使能 / 禁止IRC32K时钟稳定中断。 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时钟被阻塞时由硬件置位。 软件置位CKMIC位时清除该位。 0：时钟正常运行 1：HXTAL时钟阻塞
6	PLLDIGSTBIF	PLLDIG时钟稳定中断标志位 当PLLDIG时钟稳定且PLLDIGSTBIE位被置1时由硬件置1。 软件置位PLLDIGSTBIE位时清除该位。 0：无PLLDIG时钟稳定中断产生 1：产生PLLDIG时钟稳定中断
5:4	保留	必须保持复位值。
3	HXTALSTBIF	HXTAL时钟稳定中断标志位 当高速8 ~ 52 MHz晶体振荡器时钟稳定且HXTALSTBIE位被置1时由硬件置1。 软件置位HXTALSTBIC位时清除该位。

		0：无HXTAL时钟稳定中断产生 1：产生HXTAL时钟稳定中断
2	IRC16MSTBIF	IRC16M时钟稳定中断标志位 当内部16 MHz RC振荡器时钟稳定且IRC16MSTBIE位被置1时由硬件置1。 软件置位IRC16MSTBIC位时清除该位。 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 当低速晶体振荡器时钟稳定且LXTALSTBIE位被置1时由硬件置1。 软件置位LXTALSTBIC位时清除该位。 0：无LXTAL时钟稳定中断产生 1：产生LXTAL时钟稳定中断
0	IRC32KSTBIF	IRC32K时钟稳定中断标志位 当内部32 KHz RC振荡器时钟稳定且IRC32KSTBIE位被置1时由硬件置1。 软件置位IRC32KSTBIC位时清除该位。 0：无IRC32K时钟稳定中断产生 1：产生IRC32K时钟稳定中断

5.3.5. AHB1 复位寄存器（RCU_AHB1RST）

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
BLERST	保留									DMARST	保留				
rw										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		WIFIRST	CRCRST	保留									PCRST	PBRST	PARST
		rw	rw										rw	rw	rw

位/位域	名称	描述
31	BLERST	BLE复位 由软件置位或复位。 0：无作用 1：复位BLE
30:22	保留	必须保持复位值。
21	DMARST	DMA复位 由软件置位或复位。 0：无作用

		1：复位DMA
20:14	保留	必须保持复位值。
13	WIFIRST	WIFI复位 由软件置位或复位。 0：无作用 1：复位WIFI
12	CRCRST	CRC复位 由软件置位或复位。 0：无作用 1：复位CRC
11:3	保留	必须保持复位值。
2	PCRST	GPIO端口C复位 由软件置位或复位。 0：无作用 1：复位GPIO端口C
1	PBRST	GPIO端口B复位 由软件置位或复位。 0：无作用 1：复位GPIO端口B
0	PARST	GPIO端口A复位 由软件置位或复位。 0：无作用 1：复位GPIO端口A

5.3.6. AHB2 复位寄存器（RCU_AHB2RST）

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
保留									TRNGRST	HAURST	CAURST	PKCAURST	保留		
									rw	rw	rw	rw			

位/位域	名称	描述

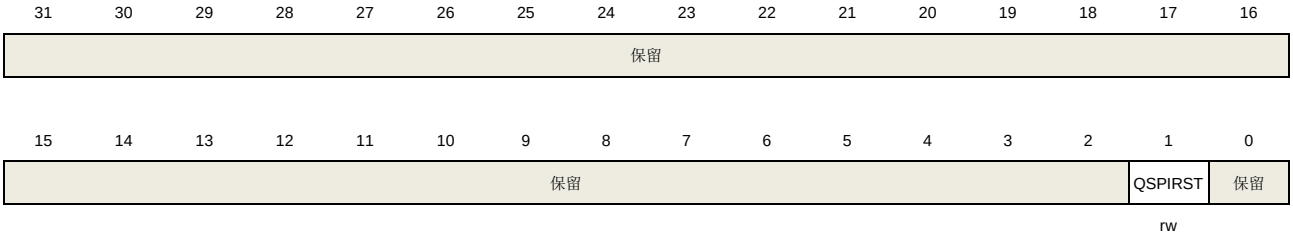
31:7	保留	必须保持复位值。
6	TRNGRST	TRNG复位 由软件置位或复位。 0：无作用 1：复位TRNG
5	HAURST	HAU复位 由软件置位或复位。 0：无作用 1：复位HAU
4	CAURST	CAU复位 由软件置位或复位。 0：无作用 1：复位CAU
3	PKCAURST	PKCAU复位 由软件置位或复位。 0：无作用 1：复位PKCAU
2:0	保留	必须保持复位值。

5.3.7. AHB3 复位寄存器（RCU_AHB3RST）

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。



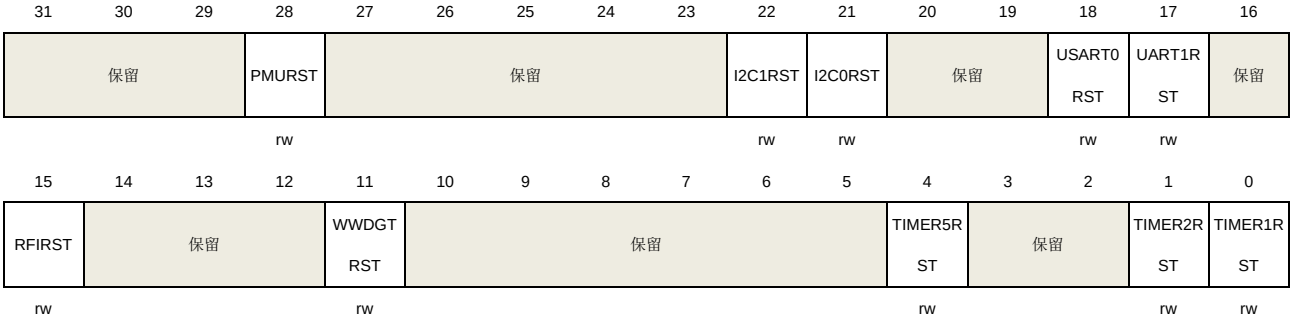
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	QSPIRST	QSPI复位 由软件置位或复位。 0：无作用 1：复位QSPI
0	保留	必须保持复位值。

5.3.8. APB1 复位寄存器（RCU_APB1RST）

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。



位/位域	名称	描述
31:29	保留	必须保持复位值。
28	PMURST	PMU复位 由软件置位或复位。 0：无作用 1：复位PMU
27:23	保留	必须保持复位值。
22	I2C1RST	I2C1复位 由软件置位或复位。 0：无作用 1：复位I2C1
21	I2C0RST	I2C0复位 由软件置位或复位。 0：无作用 1：复位I2C0
20:19	保留	必须保持复位值
18	USART0RST	USART0复位 由软件置位或复位。 0：无作用 1：复位USART0
17	UART1RST	UART1复位 由软件置位或复位。 0：无作用 1：复位UART1

16	保留	必须保持复位值。
15	RFIRST	RFI复位 由软件置位或复位。 0：无作用 1：复位RFI
14:12	保留	必须保持复位值。
11	WWDGTRST	WWDGT复位 由软件置位或复位。 0：无作用 1：复位WWDGT
10:5	保留	必须保持复位值。
4	TIMER5RST	TIMER5复位 由软件置位或复位。 0：无作用 1：复位TIMER5
3:2	保留	必须保持复位值。
1	TIMER2RST	TIMER2复位 由软件置位或复位。 0：无作用 1：复位TIMER2
0	TIMER1RST	TIMER1复位 由软件置位或复位。 0：无作用 1：复位TIMER1

5.3.9. APB2 复位寄存器（RCU_APB2RST）

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
RFRST	保留												TIMER16RST	TIMER15RST	保留
rw													rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SYSCFGRST	保留	SPIRST	保留			ADCRST	保留			UART2RST	保留			TIMER0RST
	rw		rw				rw				rw				rw

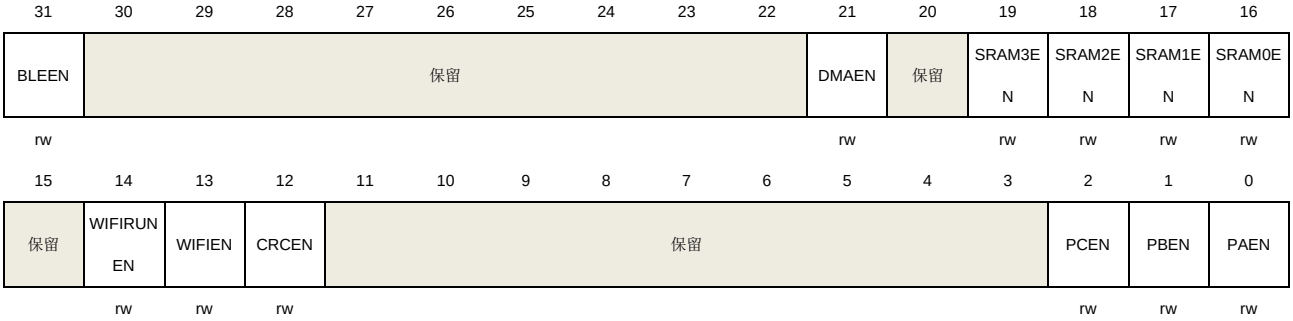
位/位域	名称	描述
31	RFRST	RF复位 由软件置位或复位。 0：无作用 1：复位RF
30:19	保留	必须保持复位值。
18	TIMER16RST	TIMER16复位 由软件置位或复位。 0：无作用 1：复位TIMER16
17	TIMER15RST	TIMER15复位 由软件置位或复位。 0：无作用 1：复位TIMER15
16:15	保留	必须保持复位值。
14	SYSCFGRST	SYSCFG复位 由软件置位或复位。 0：无作用 1：复位SYSCFG
13	保留	必须保持复位值。
12	SPIRST	SPI复位 由软件置位或复位。 0：无作用 1：复位SPI
11:9	保留	必须保持复位值。
8	ADCRST	ADC复位 由软件置位或复位。 0：无作用 1：复位所有ADC
7:5	保留	必须保持复位值。
4	UART2RST	UART2复位 由软件置位或复位。 0：无作用 1：复位UART2
3:1	保留	必须保持复位值。
0	TIMER0RST	TIMER0复位

由软件置位或复位。
0：无作用
1：复位TIMER0

5.3.10. AHB1 使能寄存器（RCU_AHB1EN）

地址偏移：0x30
复位值：0x000F 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	BLEEN	BLE时钟使能 由软件置位或复位。 0：关闭BLE时钟 1：开启BLE时钟
30:22	保留	必须保持复位值。
21	DMAEN	DMA时钟使能 由软件置位或复位。 0：关闭DMA时钟 1：开启DMA时钟
20	保留	必须保持复位值。
19	SRAM3EN	SRAM3时钟使能 由软件置位或复位。 0：关闭SRAM3时钟 1：开启SRAM3时钟
18	SRAM2EN	SRAM2时钟使能 由软件置位或复位。 0：关闭SRAM2时钟 1：开启SRAM2时钟
17	SRAM1EN	SRAM1时钟使能 由软件置位或复位。

		0：关闭SRAM1时钟 1：开启SRAM1时钟
16	SRAM0EN	SRAM0时钟使能 由软件置位或复位。 0：关闭SRAM0时钟 1：开启SRAM0时钟
15	保留	必须保持复位值。
14	WIFIRUNEN	WIFIRUN时钟使能 由软件置位或复位，当WIFIEN为0时，该位失效。 0：关闭WIFIRUN时钟 1：开启WIFIRUN时钟
13	WIFIEN	WIFI时钟使能 由软件置位或复位。 0：关闭WIFI时钟 1：开启WIFI时钟
12	CRCEN	CRC时钟使能 由软件置位或复位。 0：关闭CRC时钟 1：开启CRC时钟
11:3	保留	必须保持复位值。
2	PCEN	GPIO端口C时钟使能 由软件置位或复位。 0：关闭GPIO端口C时钟 1：开启GPIO端口C时钟
1	PBEN	GPIO端口B时钟使能 由软件置位或复位。 0：关闭GPIO端口B时钟 1：开启GPIO端口B时钟
0	PAEN	GPIO端口A时钟使能 由软件置位或复位。 0：关闭GPIO端口A时钟 1：开启GPIO端口A时钟

5.3.11. 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
保留									TRNGEN	HAUEN	CAUEN	PKCAUEN	保留		
									rw	rw	rw	rw			

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	TRNGEN	TRNG时钟使能 由软件置位或复位。 0：关闭TRNG时钟 1：开启TRNG时钟
5	HAUEN	HAU时钟使能 由软件置位或复位。 0：关闭HAU时钟 1：开启HAU时钟
4	CAUEN	CAU时钟使能 由软件置位或复位。 0：关闭CAU时钟 1：开启CAU时钟
3	PKCAUEN	PKCAU时钟使能 由软件置位或复位。 0：关闭PKCAU时钟 1：开启PKCAU时钟
2:0	保留	必须保持复位值。

5.3.12. AHB3 使能寄存器（RCU_AHB3EN）

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。

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

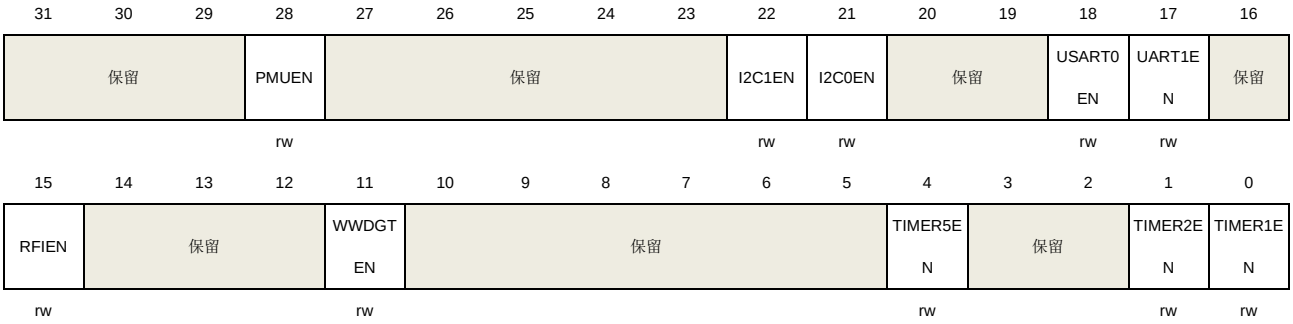
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														QSPIEN	保留
														rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	QSPIEN	QSPI时钟使能 由软件置位或复位。 0：关闭QSPI时钟 1：开启QSPI时钟
0	保留	必须保持复位值。

5.3.13. APB1 使能寄存器（RCU_APB1EN）

地址偏移：0x40
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
31:29	保留	必须保持复位值。
28	PMUEN	PMU时钟使能 由软件置位或复位。 0：关闭PMU时钟 1：开启PMU时钟
27:23	保留	必须保持复位值。
22	I2C1EN	I2C1时钟使能 由软件置位或复位。 0：关闭I2C1时钟 1：开启I2C1时钟
21	I2C0EN	I2C0时钟使能 由软件置位或复位。 0：关闭I2C0时钟 1：开启I2C0时钟

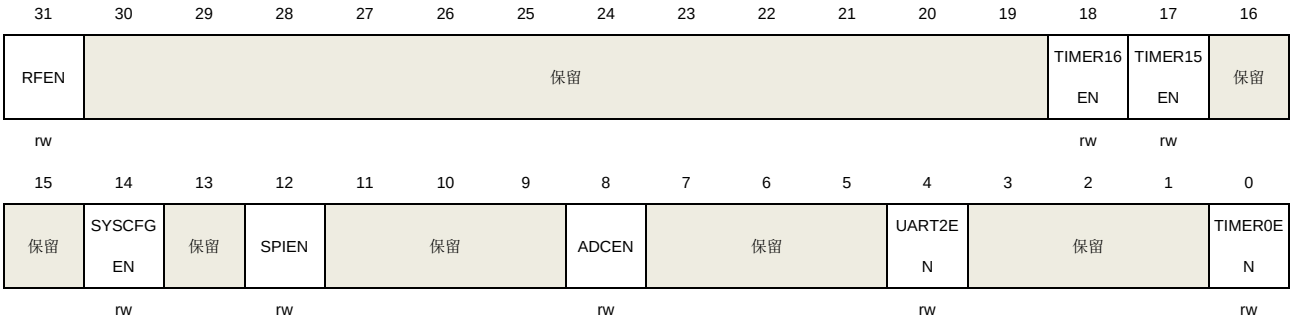
20:19	保留	必须保持复位值。
18	USART0EN	USART0时钟使能 由软件置位或复位。 0：关闭USART0时钟 1：开启USART0时钟
17	UART1EN	UART1时钟使能 由软件置位或复位。 0：关闭UART1时钟 1：开启UART1时钟
16	保留	必须保持复位值。
15	RFIEN	RFI时钟使能 由软件置位或复位。 0：关闭RFI时钟 1：开启RFI时钟
14:12	保留	必须保持复位值。
11	WWDGTEN	WWDGT时钟使能 由软件置位或复位。 0：关闭WWDGT时钟 1：开启WWDGT时钟
10:5	保留	必须保持复位值。
4	TIMER5EN	TIMER5时钟使能 由软件置位或复位。 0：关闭TIMER5时钟 1：开启TIMER5时钟
3:2	保留	必须保持复位值。
1	TIMER2EN	TIMER2时钟使能 由软件置位或复位。 0：关闭TIMER2时钟 1：开启TIMER2时钟
0	TIMER1EN	TIMER1时钟使能 由软件置位或复位。 0：关闭TIMER1时钟 1：开启TIMER1时钟

5.3.14. APB2 使能寄存器（RCU_APB2EN）

地址偏移：0x44

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	RFEN	RF时钟使能 由软件置位或复位。 0：关闭RF时钟 1：开启RF时钟
30:19	保留	必须保持复位值。
18	TIMER16EN	TIMER16时钟使能 由软件置位或复位。 0：关闭TIMER16时钟 1：开启TIMER16时钟
17	TIMER15EN	TIMER15时钟使能 由软件置位或复位。 0：关闭TIMER15时钟 1：开启TIMER15时钟
16:15	保留	必须保持复位值。
14	SYSCFGEN	SYSCFG时钟使能 由软件置位或复位。 0：关闭SYSCFG时钟 1：开启SYSCFG时钟
13	保留	必须保持复位值。
12	SPIEN	SPI时钟使能 由软件置位或复位。 0：关闭SPI时钟 1：开启SPI时钟
11:9	保留	必须保持复位值。
8	ADCEN	ADC时钟使能 由软件置位或复位。 0：关闭ADC时钟

		1：开启ADC时钟
7:5	保留	必须保持复位值。
4	UART2EN	UART2时钟使能 由软件置位或复位。 0：关闭UART2时钟 1：开启UART2时钟
3:1	保留	必须保持复位值。
0	TIMER0EN	TIMER0时钟使能 由软件置位或复位。 0：关闭TIMER0时钟 1：开启TIMER0时钟

5.3.15. AHB1 睡眠模式使能寄存器（RCU_AHB1SPEN）

地址偏移：0x50
复位值：0x0000 8000

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
FMCSPEN	保留														
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	FMCSPEN	在睡眠模式下 FMC 时钟使能 由软件置位或复位 0：在睡眠模式下关闭 FMC 时钟 1：在睡眠模式下开启 FMC 时钟
14:0	保留	必须保持复位值。

5.3.16. 备份域控制寄存器（RCU_BDCTL）

地址偏移：0x70
复位值：0x0000 0018，只能由备份域复位进行复位。

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（RCU_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 由软件置位或复位。 0：无作用 1：复位备份域
15	RTCEN	RTC时钟使能 由软件置位或复位。 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时钟源选择 由软件置位或清零来控制RTC的时钟源。一旦RTC的时钟源选择后，除了将备份域复位否则时钟源不能被改变。 00：没有时钟 01：选择CK_LXTAL时钟作为RTC的时钟源 10：选择CK_IRC32K时钟作为RTC的时钟源 11：选择CK_HXTAL / RTCDIV时钟作为RTC的时钟源，请参考RCU_CFG0寄存器的RTCDIV位域
7:5	保留	必须保持复位值。
4:3	LXTALDRI[1:0]	LXTAL驱动能力 由软件置位或复位。当备份域复位时将复位该值。 00：低驱动能力 01：高驱动能力 10：更高驱动能力 11：最高驱动能力（复位值） 注意：LXTALDRI位在旁路模式下无效。
2	LXTALBPS	LXTAL旁路模式使能

由软件置位或复位。
0：禁止LXTAL旁路模式
1：使能LXTAL旁路模式

1	LXTALSTB	低速晶体振荡器稳定标志位 硬件置‘1’来指示LXTAL振荡器时钟是否稳定待用。 0：LXTAL未稳定 1：LXTAL已稳定
0	LXTALEN	LXTAL时钟使能 由软件置位或复位。 0：关闭LXTAL时钟 1：使能LXTAL时钟

5.3.17. 复位源/时钟寄存器（RCU_RSTSCK）

地址偏移：0x74
复位值：0x0C00 0000，所有复位标志位仅在电源复位时被清零，RSTFC / IRC32KEN 在系统复位时被清零。

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	保留							
r	r	r	r	r	r		rw								

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位。 向RSTFC位写1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。 向RSTFC位写1来清除该位。 0：无窗口看门狗复位发生 1：发生窗口看门狗复位
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。

		向RSTFC位写1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。 向RSTFC位写1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。 向RSTFC位写1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 外部引脚复位发生时由硬件置1。 向RSTFC位写1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：清除所有复位标志位
23:2	保留	必须保持复位值。
1	IRC32KSTB	IRC32K时钟稳定标志位 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已稳定
0	IRC32KEN	IRC32K使能 由软件置位和复位。 0：关闭IRC32K时钟 1：开启IRC32K时钟

5.3.18. PLLDIG 时钟配置寄存器 0（RCU_PLLDIGCFG0）

地址偏移：0x84

复位值：0x0B00 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

PLLDIGDIV_SYS[5:0]	PLLDIGOSEL[1:0]	保留
rw	rw	

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

位/位域	名称	描述
31:26	PLLDIGDIV_SYS[5:0]	PLLDIG时钟的分频因子用于系统时钟 由软件置位和复位，控制PLLDIG分频用于系统时钟。 000000：PLLDIG时钟除以1作为系统时钟 000001：PLLDIG时钟除以2作为系统时钟 000010：PLLDIG时钟除以3作为系统时钟 … 111111：PLLDIG时钟除以64作为系统时钟
25:24	PLLDIGOSEL[1:0]	PLLDIG输出频率选择 00：选择192Mhz作为PLLDIG输出频率 01：选择240Mhz作为PLLDIG输出频率 10：选择320Mhz作为PLLDIG输出频率 11：选择480Mhz作为PLLDIG输出频率
23:0	保留	必须保持复位值。

5.3.19. 时钟配置寄存器 1（RCU_CFG1）

地址偏移：0x8C

复位值：0x0030 0600

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
USART0SEL[1:0]		保留		I2C0SEL[1:0]		保留	TIMERSEL	保留		LDO_ANA_LQB	LDO_CLK_LQB	BGPU	LDOCLKPU	LDOANAPU	RFPLLPU
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
RFPLLLOCK	RFPLLCALEN	保留		BGVBIT[2:0]			IRC16MDIV[8:0]								
ro	rw			rw			rw								

位/位域	名称	描述
31:30	USART0SEL[1:0]	USART0时钟源选择 通过软件置1和复位以控制USART0时钟源。 00：选择CK_APB1作为USART0时钟源 01：选择CK_SYS作为USART0时钟源

		10：选择CK_LXTAL作为USART0时钟源 11：选择CK_IRC16M作为USART0时钟源
29:28	保留	必须保持复位值。
27:26	I2C0SEL[1:0]	I2C0时钟源选择 通过软件置1和复位以控制I2C0时钟源。 00：选择CK_APB1作为I2C0源时钟 01：选择CK_SYS作为I2C0源时钟 1x：选择CK_IRC16M作为I2C0源时钟
25	保留	必须保持复位值。
24	TIMERSEL	TIMER时钟源选择 由软件置位或复位，该位定义了所有定时器的时钟源选择。 0：如果RCU_CFG0寄存器的APB1PSC / APB2PSC位域的值为0b0xx（CK_APBx = CK_AHB）或0b100（CK_APBx = CK_AHB / 2），定时器时钟等于CK_AHB（CK_TIMERx = CK_AHB），否则定时器时钟等于APB时钟的两倍（在APB1域的定时器：CK_TIMERx = 2 x CK_APB1，在APB2域的定时器：CK_TIMERx = 2 x CK_APB2） 1：如果RCU_CFG0寄存器的APB1PSC / APB2PSC位域的值为0b0xx（CK_APBx = CK_AHB），0b100（CK_APBx = CK_AHB / 2），或0b101（CK_APBx = CK_AHB / 4），定时器时钟等于CK_AHB（CK_TIMERx = CK_AHB）。否则定时器时钟等于APB时钟的四倍（在APB1域的定时器：CK_TIMERx = 4 x CK_APB1；在APB2域的定时器：CK_TIMERx = 4 x CK_APB2）
23:22	保留	必须保持复位值。
21	LDO_ANA_LQB	模拟LDO电流偏置模式选择 0：模拟LDO高电流偏置模式 1：模拟LDO低电流偏置模式
20	LDO_CLK_LQB	时钟LDO电流偏置模式选择 0：时钟LDO高电流偏置模式 1：时钟LDO低电流偏置模式
19	BGPU	BandGap上电使能 当PLLDIGEN置1时，该位不能写0。 0：BandGap掉电 1：BandGap上电
18	LDOCLKPU	LDO时钟上电使能RF / ADC 0：LDO时钟掉电 1：LDO时钟上电
17	LDOANAPU	LDO模拟上电启用RF滤波器 0：LDO模拟掉电 1：LDO模拟上电

16	RFPLLPU	RFPLL上电启用 0：RFPLL掉电 1：RFPLL上电
15	RFPLLLOCK	RF PLL锁定 0：RFPLL未锁定 1：RFPLL锁定
14	RFPLLCALEN	RF PLL计算使能 0：RF PLL计算关闭 1：RF PLL计算打开
13:12	保留	必须保持复位值。
11:9	BGVBIT[2:0]	BandGap功率调整，当HXTALEN或PLLDIGEN开启时无法写入。
8:0	IRC16MDIV[8:0]	IRC16M时钟的分频因子用于系统时钟输出频率 当系统时钟选择IRC16M或IRC16MEN时无法写入。 00000000：IRC16M时钟除以1 00000001：IRC16M时钟除以2 00000010：IRC16M时钟除以3 … 11111111：IRC16M时钟除以512

5.3.20. 附加时钟控制寄存器（RCU_ADDCTL）

地址偏移：0x90
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
31:6	保留	必须保持复位值。
5:1	TRNGCKDIV[4:0]	PLLDIG时钟的分频因子用于TRNG时钟 通过软件设置和复位，以控制TRNG时钟的PLLDIG时钟分频因子 00000：PLLDIG时钟除以1作为TRNG时钟 00001：PLLDIG时钟除以2作为TRNG时钟 00010：PLLDIG时钟除以3作为TRNG时钟 …

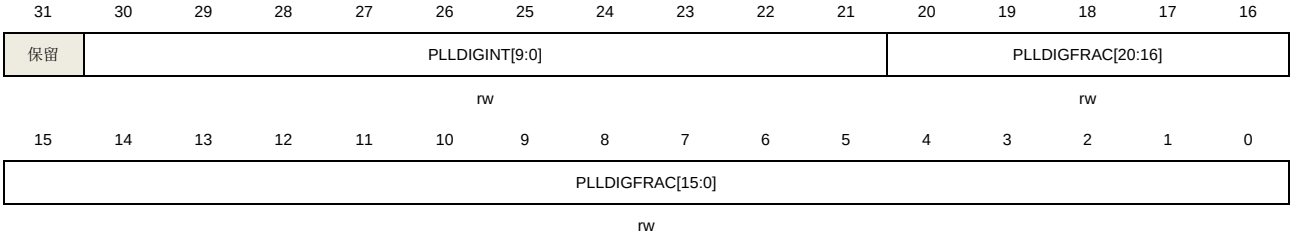
		11111：PLLDIG时钟除以32作为TRNG时钟
0	保留	必须保持复位值。

5.3.21. PLLDIG 时钟配置寄存器 1（RCU_PLLDIGCFG1）

地址偏移：0x94
复位值：0x0780 0000

如果使用 WIFI 功能，则需要配置 RF 时钟源*倍频因子 = 960MHz，才能保证 WIFI 的功能正常。RF 要求时钟频率达 960 MHz，RF 的时钟来源于 HXTAL 或 IRC16M，PLLDIGINT 赋值为“960 MHz / RF 时钟源”的整数部分，PLLDIGFRAC 赋值为“960 MHz / RF 时钟源”的小数部分。

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

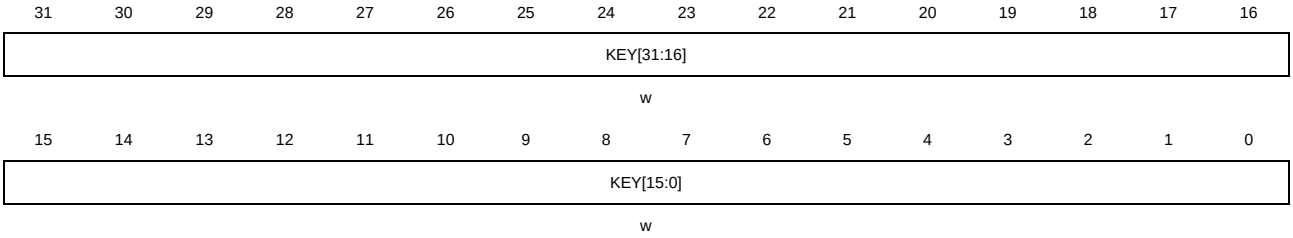


位/位域	名称	描述
31	保留	必须保持复位值。
30:21	PLLDIGINT[9:0]	控制PLLDIG倍频因子的整数部分。 注意：PLLDIGINT[9:0]的赋值应不小于0x08。
20:0	PLLDIGFRAC[20:0]	控制PLLDIG倍频因子的小数部分。

5.3.22. 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

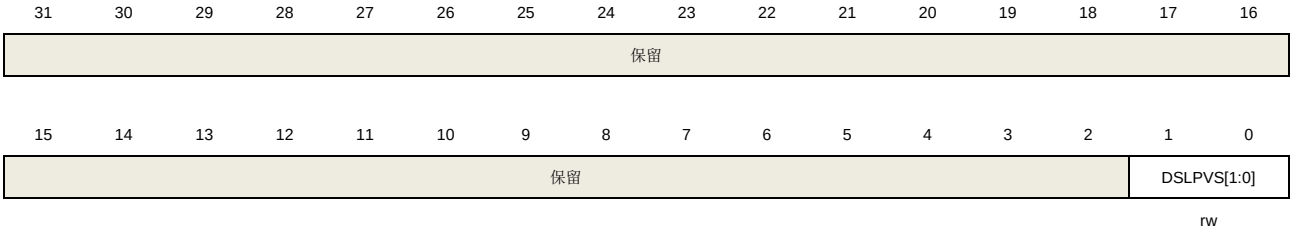
31:0	KEY[31:0]	RCU_DSV寄存器解锁 这些位仅能被软件写，若读这些位，则全为0。只有在向RCC_VKEY寄存器写0x1A2B3C4D后，RCU_DSV寄存器才能被写。

5.3.23. 深度睡眠模式电压寄存器（RCU_DSV）

地址偏移：0x134

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
31:2	保留	必须保持复位值。
1:0	DSLPVS[1:0]	深度睡眠模式电压选择 由软件置位和清零这些位。深度睡眠模式电压选择，LDO需配置为低驱动模式。 注：如果不配置为低驱动模式，LDO电压无法输出。 00：在深度睡眠模式下内核电压为1.1V 01：在深度睡眠模式下内核电压为1.0V 10：在深度睡眠模式下内核电压为0.9V 11：在深度睡眠模式下内核电压为0.9V

6. 中断/事件控制器（EXTI）

6.1. 简介

RISC-V 集成了改进型内核中断控制器（ECLIC）来实现高效的中断处理。ECLIC 旨在为 RISC-V 系统提供低延迟、矢量化、抢占式的中断。当 ECLIC 被激活后，它将包含并替换现有的 RISC-V 处理器局部中断控制器。CLIC 设计有一个基本设计，需要最少的硬件，但它支持额外的扩展来提供硬件加速。ECLIC 设计的目标是为各种软件 ABI 和中断模型提供支持，而不需要复杂的硬件影响高性能处理器的实现。它与处理器核心紧密耦合。可以阅读 RISC-V 的技术参考手册，了解有关 ECLIC 的更多详细信息。

EXTI（中断/事件控制器）包括 25 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

- 多达75种可屏蔽的外设中断；
- 4位中断优先级配置位—提供16个中断优先等级；
- 支持异常抢占和咬尾中断；
- 将系统从省电模式唤醒；
- EXTI中有多达25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. 功能描述

RISC-V 处理器和改进型内核中断控制器（ECLIC）在机器（machine）模式下对所有异常进行优先级区分以及处理。

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。下表列出了所有的中断类型。

表 6-1. 中断向量表

Vector Number	Interrupt Description	Vector Address
3	CLIC_INT_SFT	0x0000_000C
7	CLIC_INT_TMR	0x0000_001C
19	窗口看门狗定时器中断	0x0000_004C
20	连接到 EXTI 线的 LVD 中断	0x0000_0050
21	RTC 侵入和时间戳中断	0x0000_0054

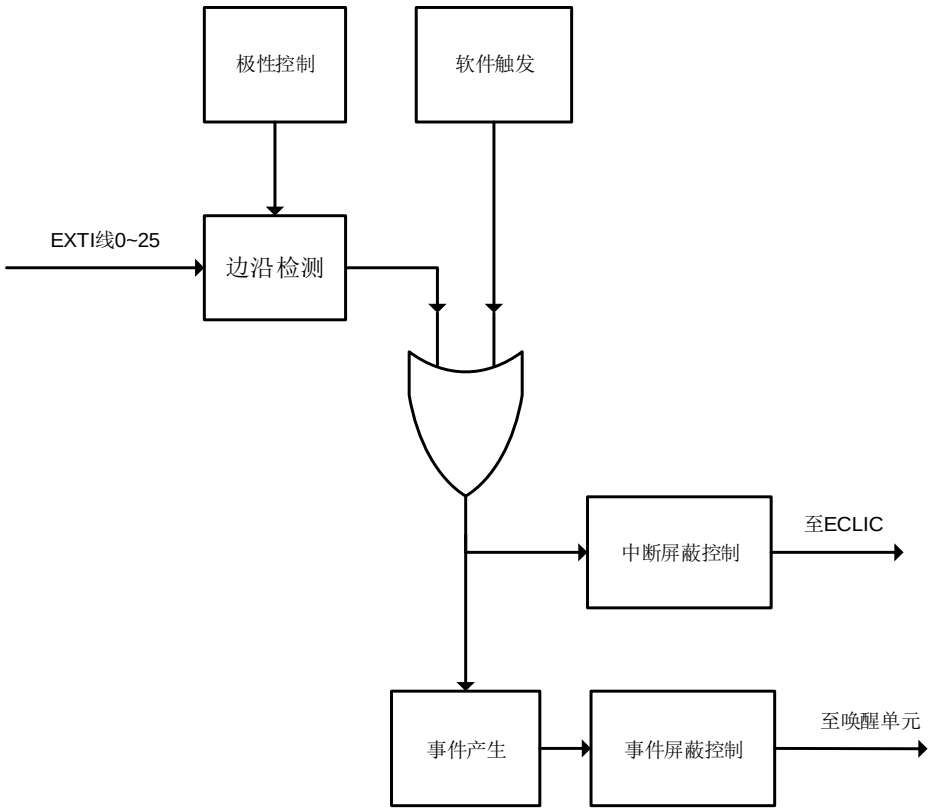
Vector Number	Interrupt Description	Vector Address
22	RTC 唤醒中断	0x0000_0058
23	FMC 全局中断	0x0000_005C
24	RCU 全局中断	0x0000_0060
25	EXTI 线 0 中断	0x0000_0064
26	EXTI 线 1 中断	0x0000_0068
27	EXTI 线 2 中断	0x0000_006C
28	EXTI 线 3 中断	0x0000_0070
29	EXTI 线 4 中断	0x0000_0074
30	DMA 通道 0 全局中断	0x0000_0078
31	DMA 通道 1 全局中断	0x0000_007C
32	DMA 通道 2 全局中断	0x0000_0080
33	DMA 通道 3 全局中断	0x0000_0084
34	DMA 通道 4 全局中断	0x0000_0088
35	DMA 通道 5 全局中断	0x0000_008C
36	DMA 通道 6 全局中断	0x0000_0090
37	DMA 通道 7 全局中断	0x0000_0094
38	ADC 中断	0x0000_0098
39~41	保留	0x0000_009C - 0x0000_00A4
42	EXTI 线 5-9 中断	0x0000_00A8
43	TIMER0 中止中断	0x0000_00AC
44	TIMER0 更新中断	0x0000_00B0
45	TIMER0 触发与通道换相中断	0x0000_00B4
46	TIMER0 通道捕获比较中断	0x0000_00B8
47	TIMER1 全局中断	0x0000_00BC
48	TIMER2 全局中断	0x0000_00C0
49	保留	0x0000_00C4
50	I2C0 事件中断	0x0000_00C8
51	I2C0 错误中断	0x0000_00CC
52	I2C1 事件中断	0x0000_00D0
53	I2C1 错误中断	0x0000_00D4
54	SPI 全局中断	0x0000_00D8
55	保留	0x0000_00DC
56	USART0 全局中断	0x0000_00E0
57	UART1 全局中断	0x0000_00E4
58	UART2 全局中断	0x0000_00E8
59	EXTI 线 10-15 中断	0x0000_00EC
60	RTC 闹钟中断	0x0000_00F0
61~62	保留	0x0000_00F4 - 0x0000_00F8

Vector Number	Interrupt Description	Vector Address
63	TIMER15 全局中断	0x0000_00FC
64	TIMER16 全局中断	0x0000_0100
65~69	保留	0x0000_0104 - 0x0000_0114
70	I2C0 唤醒中断	0x0000_0118
71	USART0 唤醒中断	0x0000_011C
72	Reserved	0x0000_0120
73	TIMER5 全局中断	0x0000_0124
74	WIFI 协议触发中断	0x0000_0128
75	WIFI MAC 中断	0x0000_012C
76	WIFI 发送中断	0x0000_0130
77	WIFI 接收中断	0x0000_0134
78~82	保留	0x0000_0138- 0x0000_0148
83	LA 中断	0x0000_014C
84	WIFI 唤醒中断	0x0000_0150
85	BLE 唤醒中断	0x0000_0154
86	Platform（PLF）唤醒中断	0x0000_0158
87~94	ISO 蓝牙时间戳中断 0~7	0x0000_015C- 0x0000_0178
95	PMU 中断	0x0000_017C
96~97	保留	0x0000_018A- 0x0000_0184
98	CAU 全局中断	0x0000_0188
99	HAU / TRNG 全局中断	0x0000_018C
100	保留	0x0000_0190
101	WIFI 中断	0x0000_0194
102	SW 触发中断	0x0000_0198
103	Fine 定时器目标中断	0x0000_019C
104	时间戳目标 1 中断	0x0000_01A0
105	时间戳目标 2 中断	0x0000_01A4
106	时间戳目标 3 中断	0x0000_01A8
107	加密引擎中断	0x0000_01AC
108	睡眠模式中断	0x0000_01B0
109	Half slot 中断	0x0000_01B4
110	FIFO 活动中断	0x0000_01B8
111	错误中断	0x0000_01BC
112	频率选择中断	0x0000_01C0
113	EFUSE 全局中断	0x0000_01C4
114	QSPI 全局中断	0x0000_01C8

Vector Number	Interrupt Description	Vector Address
115	PKCAU 全局中断	0x0000_01CC

6.4. 外部中断及事件结构框图

图 6-1. EXTI 结构框图



6.5. 外部中断及事件功能概述

EXTI 包含多达 25 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 9 根线，具体细节参考*表 6-2. 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。RISC-V 内核完全支持等待中断（WFI）指令。当某些预期的事件发生时，例如一个特定的 I/O 管脚电平翻转或者 RTC 闹钟动作，EXTI 能唤醒处理器及整个系统。

硬件触发

硬件触发被用来检测外部或内部信号的电压变化。软件需要按如下步骤配置来使用这项功能：

1. 根据应用需要配置SYSCFG模块中的EXTI触发源；
2. 配置EXTI_RTEN寄存器和EXTI_FTEN寄存器以使能相应引脚的上升沿或下降沿检测（软件应当同时配置引脚对应的RTENx和FTENx位以检测该引脚上升沿和下降沿的变化）；
3. 通过配置引脚对应的EXTI_INTEN或EXTI_EVEN位，使能中断或事件；
4. EXTI开始检测被配置的引脚上的电平变化，当这些引脚上期望的变化被检测到时，使能的中断或事件将被触发。如果为中断触发，则对应的PD位将立刻被置1；如果为事件触发，则对应的PD位不被置1。软件需要响应该中断或事件并清除相应PDx位。

软件触发

按照如下步骤软件也可以触发 EXTI 中断或事件：

1. 配置对应的EXTI_INTEN或EXTI_EVEN位使能中断或事件；
2. 配置EXTI_SWIEV寄存器的对应SWIEVx位，使能的中断或事件将被立即触发。如果为中断触发，则对应的PD位将立刻被置1；如果为事件触发，则对应的PD位不被置1。软件需要响应该中断或事件并清除相应PDx位。

表 6-2. EXTI 触发源

EXTI 线编号	触发源
0	PA0 / PB0
1	PA1 / PB1
2	PA2 / PB2
3	PA3 / PB3
4	PA4 / PB4
5	PA5
6	PA6
7	PA7
8	PA8 / PC8
9	PA9
10	PA10
11	PA11 / PB11
12	PA12 / PB12
13	PA13 / PB13 / PC13
14	PA14 / PC14
15	PA15 / PB15 / PC15
16	LVD
17	RTC 闹钟
18	保留
19	WIFI 唤醒
20	RTC 侵入和时间戳
21	RTC 唤醒
22	I2C0 唤醒
23	USART0 唤醒

EXTI 线编号	触发源
24	BLE 唤醒
25	PLF 唤醒

6.6. EXTI 寄存器

EXTI 基地址：0x4001 3C00

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						INTEN25	INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	保留	INTEN17	INTEN16
						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
31:26	保留	必须保持复位值。
25:19	保留	中断使能位 x（x = 19...25） 0：第 x 线中断被禁止 1：第 x 线中断被使能
18	保留	必须保持复位值。
17:0	INTENx	中断使能位 x（x = 0...17） 0：第 x 线中断被禁止 1：第 x 线中断被使能

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						EVEN25	EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	保留	EVEN17	EVEN16
						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

31:26	保留	必须保持复位值。
25:19	EVENx	事件使能位 x（x = 19…25） 0：第 x 线事件被禁止 1：第 x 线事件被使能
18	保留	必须保持复位值。
17:0	EVENx	事件使能位 x（x = 0…17） 0：第 x 线事件被禁止 1：第 x 线事件被使能

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						RTEN25	RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	保留	RTEN17	RTEN16
						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
31:26	保留	必须保持复位值。
25:19	RTENx	上升沿触发使能（x = 19...25） 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）
18	保留	必须保持复位值。
17:0	RTENx	上升沿触发使能（x = 0...17） 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						FTEN25	FTEN24	FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	保留	FTEN17	FTEN16
						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: 26	保留	必须保持复位值。
25:19	FTENx	下降沿触发使能（x = 19...25） 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）
18	保留	必须保持复位值。
17:0	FTENx	下降沿触发使能（x = 0...17） 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						SWIEV25	SWIEV24	SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	保留	SWIEV17	SWIEV16
						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
31:26	保留	必须保持复位值。
25:19	SWIEVx	中断/事件软件触发（x = 19...25） 0：写 0 无效 1：在该位为 0 的情况下，置位该位将触发一个 EXTI 线 x 软件中断/事件请求。通过清除 EXTI_PD 寄存器中相应的 PDx 位可清除该位
18	保留	必须保持复位值。
17:0	SWIEVx	中断/事件软件触发（x = 0...17） 0：写 0 无效 1：在该位为 0 的情况下，置位该位将触发一个 EXTI 线 x 软件中断/事件请求。通过清除 EXTI_PD 寄存器中相应的 PDx 位可清除该位

6.6.6. 挂起寄存器（EXTI_PD）

地址偏移：0x14

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						PD25	PD24	PD23	PD22	PD21	PD20	PD19	保留	PD17	PD16
						rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		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
31:26	保留	必须保持复位值。
25:19	PDx	中断挂起状态（x = 19...25） 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
18	保留	必须保持复位值。
17:0	PDx	中断挂起状态（x = 0...17） 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 29 个通用 I/O 引脚（GPIO），分别为 PA0 ~ PA15，PB0 ~ PB4，PB11 ~ PB13，PB15，PC8 和 PC13 ~ PC15，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。GPIO 引脚上的外部中断在中断/事件控制器（EXTI）中有相关的控制和配置寄存器。

GPIO 端口和其他的备用功能（AFs）共用引脚，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. 主要特性

- 输入/输出方向控制；
- 施密特触发器输入功能使能控制；
- 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；
- 可编程触发沿的外部中断—使用EXTI配置寄存器
- 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. 功能描述

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。当选择 AF 功能时，引脚 AF 输入/输出是通过 AF 功能输出使能来选择。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。

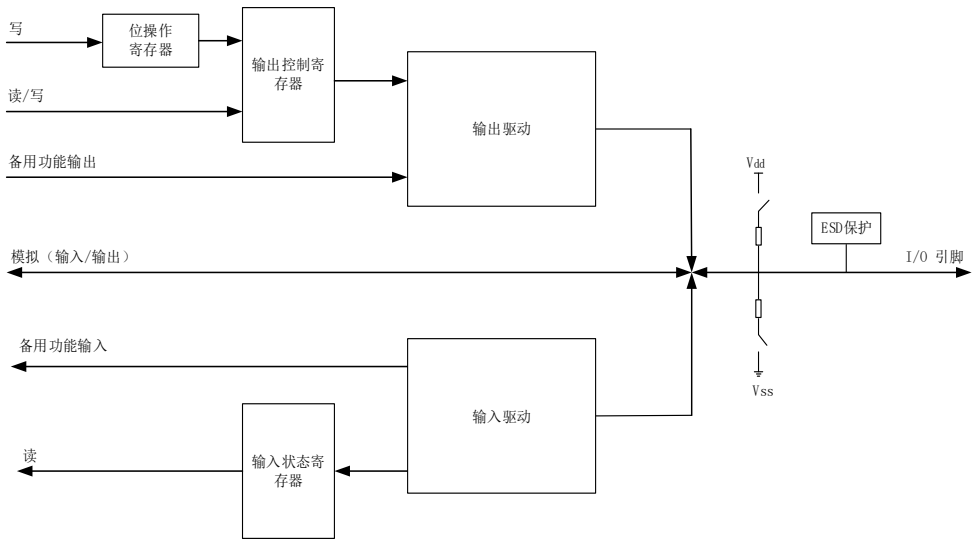
表 7-1. GPIO 配置表

PAD TYPE			CTLy	OMy	PUDy
GPIO 输入	X	浮空	00	X	00
		上拉			01

PAD TYPE			CTLy	OMy	PUDy
		下拉			10
GPIO 输出	推挽	浮空	01	0	00
		上拉			01
		下拉			10
	开漏	浮空		1	00
		上拉			01
		下拉			10
AFIO 输入	X	浮空	10	X	00
		上拉			01
		下拉			10
AFIO 输出	推挽	浮空	10	0	00
		上拉			01
		下拉			10
	开漏	浮空		1	00
		上拉			01
		下拉			10
模拟	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，JTAG 引脚为输入 PU / PD 模式：

PA15：JTDI 为上拉模式；

PA14：JTCK 为下拉模式；

PA13：JTMS 为上拉模式；

PB4：NJTRST 为上拉模式；

PB3：JTDI 为浮空模式。

GPIO 引脚可以配置为输入或输出模式，当 GPIO 引脚可配置为输入引脚时，所有的 GPIO 引脚内部都有一个可选择的弱上拉和弱下拉电阻。外部引脚上的数据在每个 AHB 时钟周期时都会装载到数据输入寄存器（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。输出寄存器（GPIOx_OCTL）的值将会从相应 I/O 引脚上输出。

当对 GPIOx_OCTL 进行位操作时，不需要先读再写，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/事件线

只有在输入模式下配置，端口才能使用外部中断/事件线。

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_AFSELz（z = 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 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。

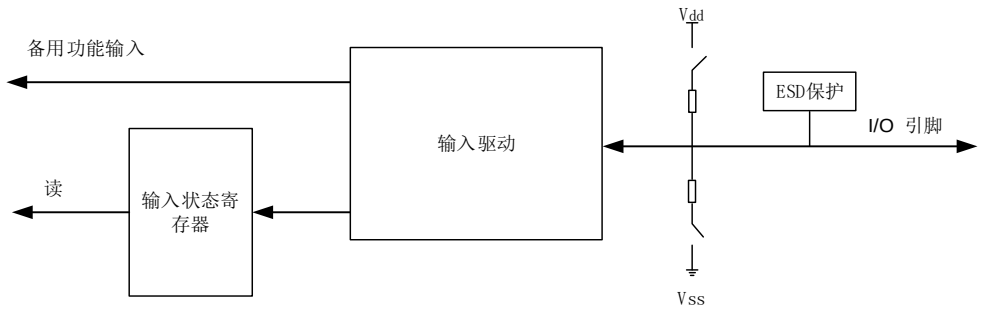
7.3.5. 输入配置

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*显示 I/O 引脚的输入配置。

图 7-2. 输入配置的基本结构



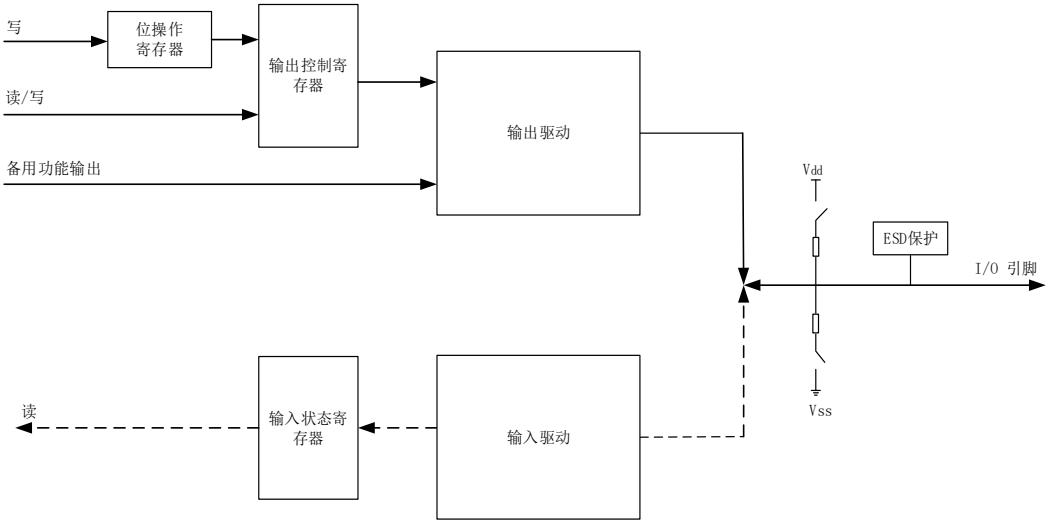
7.3.6. 输出配置

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；
- 输出缓冲器使能；
- 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；
- 对端口输出控制寄存器进行读操作，将返回上次写入的值；
- 对端口输入状态寄存器进行读操作，将获得当前I/O口的状态。

*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；
- 端口输入状态寄存器相应位为“0”。

*图 7-4. 模拟配置的基本结构*是 I/O 端口的模拟模式配置。

图 7-4. 模拟配置的基本结构

7.3.8. 备用功能（AF）配置

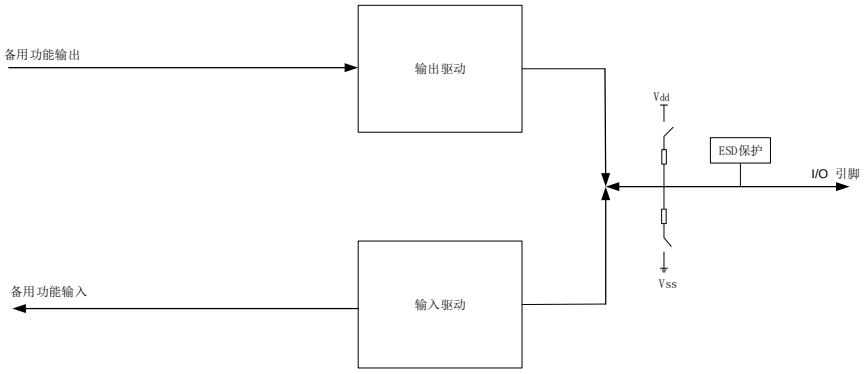
为了适应不同的器件封装，GPIO 端口支持软件配置将一些备用功能应用到其他引脚上。

当引脚配置为备用功能时：

- 使用开漏或推挽功能时，可使能输出缓冲器；
- 输出缓冲器由外设驱动；
- 施密特触发输入使能；
- 在输入配置时，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_AFSELz（z = 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。通过特定的锁定序列配置 GPIOx_LOCK 中的 LKK 位和 LKy 位，相应的端口位被锁定，直到下一个复位前，相应端口位的配置都不能修改。建议在电源驱动模块的配置中使用锁定功能。

7.3.10. GPIO I/O 补偿单元

默认情况下，I/O 补偿单元是不使用的，当 I/O 端口输出速度大于 50MHz 时，建议使用 I/O 补偿单元对 I/O 端口进行斜率控制，从而降低 I/O 端口噪声对工作电源的影响。

在使能 I/O 补偿单元后，将产生一个准备完成标志位 CPS_RDY，用于指示补偿单元已经准备好，可以使用。

7.3.11. 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：0x4002 0000

GPIOB 基地址：0x4002 0400

GPIOC 基地址：0x4002 0800

7.4.1. 端口控制寄存器（GPIOx_CTL，x=A..C）

地址偏移：0x00
复位值：GPIOA_CTL 0xA800 0000

GPIOB_CTL 0x0000 0280

GPIOC_CTL 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。
该寄存器只能按字（32 位）读访问。

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。 参考 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。 参考 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。 参考 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。 参考 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。 参考 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。 参考 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。 参考 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。 参考 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。 参考 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。 参考 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。 参考 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。 参考 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。 参考 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。 参考 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。 参考 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：备用功能描述 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..C）

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）写访问。

该寄存器只能按字（32 位）读访问。

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：端口输出推挽模式（复位值） 1：端口输出开漏模式

7.4.3. 端口输出速度寄存器（GPIOx_OSPD, x=A..C）

地址偏移：0x08

复位值：GPIOA_OSPD 0x0C00 0000

GPIOB_OSPD 0x0000 00C0

GPIOC_OSPD 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。

该寄存器只能按字（32 位）读访问。

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。 00：输出速度等级 0（复位值） 01：输出速度等级 1 10：输出速度等级 2 11：输出速度等级 3

7.4.4. 端口上拉/下拉寄存器（GPIOx_PUD, x=A..C）

地址偏移：0x0C

复位值：GPIOA_PUD 0x6400 0000

GPIOB_PUD 0x0000 0100

GPIOC_PUD 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。

该寄存器只能按字节（32 位）读访问。

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/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
29:28	PUD14[1:0]	Pin 14 上拉/下拉位

		该位由软件置位和清除。 参考 PUD0[1:0]的描述
27:26	PUD13[1:0]	Pin 13 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
25:24	PUD12[1:0]	Pin 12 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
23:22	PUD11[1:0]	Pin 11 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
21:20	PUD10[1:0]	Pin 10 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
19:18	PUD9[1:0]	Pin 9 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
17:16	PUD8[1:0]	Pin 8 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
15:14	PUD7[1:0]	Pin 7 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
13:12	PUD6[1:0]	Pin 6 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
11:10	PUD5[1:0]	Pin 5 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
9:8	PUD4[1:0]	Pin 4 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
7:6	PUD3[1:0]	Pin 3 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
5:4	PUD2[1:0]	Pin 2 上拉/下拉位 该位由软件置位和清除。

参考 PUD0[1:0]的描述

3:2	PUD1[1:0]	Pin 1 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
1:0	PUD0[1:0]	Pin 0 上拉/下拉位 该位由软件置位和清除。 00：浮空模式，无上拉/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..C）

地址偏移：0x10
复位值：0x0000 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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	ISTAT 9	ISTAT 8	ISTAT 7	ISTAT 6	ISTAT 5	ISTAT 4	ISTAT 3	ISTAT 2	ISTAT 1	ISTAT 0
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) 这些位由硬件置位和清除。 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..C）

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）写访问。
该寄存器只能按字（32 位）读访问。

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) 这些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..C）

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) 这些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	端口置位位 y(y=0..15) 这些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..C）

地址偏移：0x1C

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）写访问。

该寄存器只能按字（32 位）读访问。

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

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	锁定序列键 该位只能通过使用 Lock Key 写序列设置，始终可读。 0：GPIO_LOCK 寄存器和端口配置没有锁定 1：直到下一次 MCU 复位前，GPIO_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[15:0]的值必须保持。
15:0	LKy	端口锁定位 y（y = 0..15） 这些位由软件置位和清除。 0：相应的端口位配置没有锁定 1：当 LKK 位置 1 时，相应的端口位配置被锁定

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

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）写访问。
该寄存器只能按字（32 位）读访问。

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 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
27:24	SEL6[3:0]	Pin 6 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
23:20	SEL5[3:0]	Pin 5 备用功能选择

		该位由软件置位和清除。 参考 SEL0 [3:0]的描述
19:16	SEL4[3:0]	Pin 4 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
15:12	SEL3[3:0]	Pin 3 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
11:8	SEL2[3:0]	Pin 2 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
7:4	SEL1[3:0]	Pin 1 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
3:0	SEL0[3:0]	Pin 0 备用功能选择 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 … 1111：选择 AF15 功能

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

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）写访问。

该寄存器只能按字（32 位）读访问。

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 备用功能选择 该位由软件置位和清除。

参考 SEL8[3:0]的描述

27:24	SEL14[3:0]	Pin 14 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
23:20	SEL13[3:0]	Pin 13 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
19:16	SEL12[3:0]	Pin 12 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
15:12	SEL11[3:0]	Pin 11 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
11:8	SEL10[3:0]	Pin 10 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
7:4	SEL9[3:0]	Pin 9 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
3:0	SEL8[3:0]	Pin 8 备用功能选择 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 … 1111：选择 AF15 功能

7.4.11. 位清除寄存器（GPIOx_BC, x=A..C）

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) 这些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..C）

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) 这些位由软件置位和清除。 0：相应 OCTLy 位没有改变 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差错。

CRC 计算单元使用固定多项式计算 32 位 CRC 校验码。

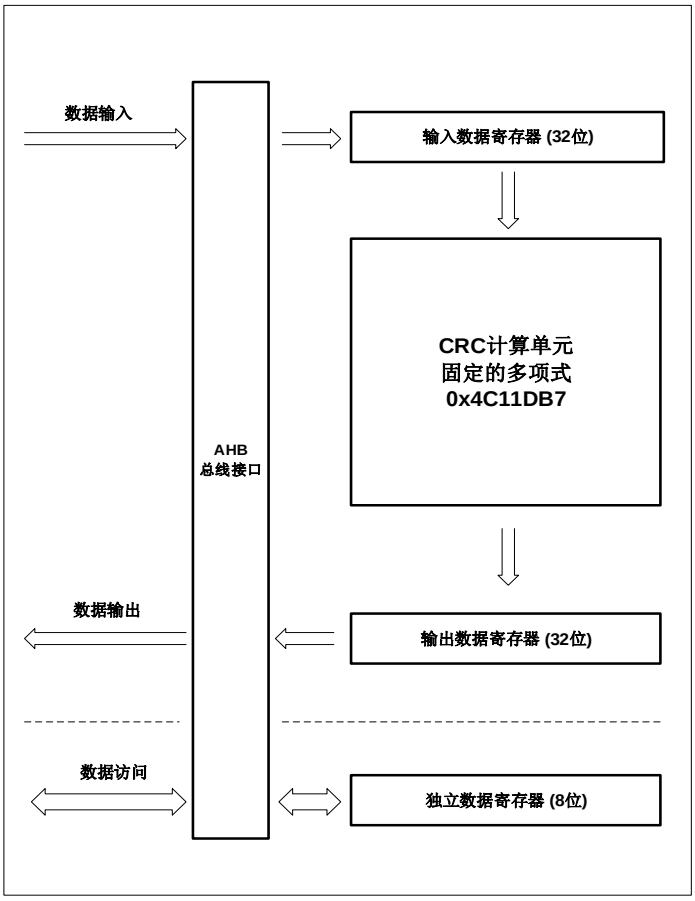
8.2. 主要特征

- 32位数据输入/输出寄存器。对于32位的输入数据，从数据输入到得出计算结果，需要4个AHB的时钟周期；
- 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；
- 固定的计算多项式：0x4C11DB7：

$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$

该 32 位 CRC 多项式与以太网 CRC 计算多项式相同。

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，CRC计算单元将基于新输入的原始数据和前一次CRC_DATA寄存器中的结果进行计算。
  - 对于32位的数据长度的CRC计算，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位的独立数据寄存器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。

8.4. CRC 寄存器

CRC 基地址：0x4002 3000

8.4.1. 数据寄存器（RC_DATA）

地址偏移：0x00
复位值：0xFFFF FFFF

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]	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。

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								FDATA[7:0]							
								rw							

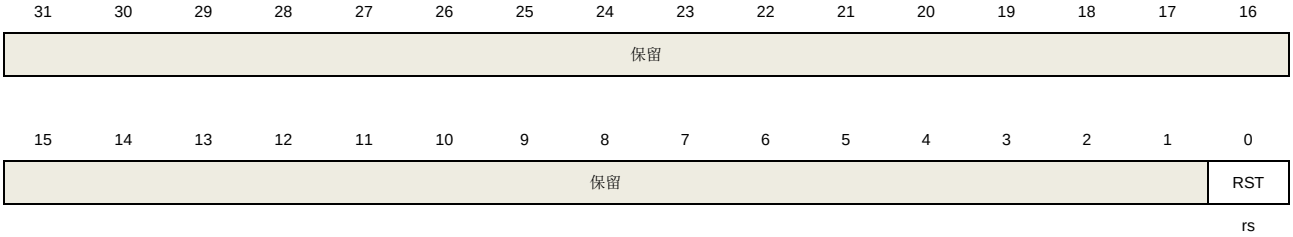
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	FDATA[7:0]	独立数据寄存器位 软件可读可写。 这些位与 CRC 计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受 CRC_CTL 寄存器的影响。

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
31:1	保留	必须保持复位值。
0	RST	将该位置 1 可以复位 CRC_DATA 寄存器，并设置其值为 0xFFFFFFFF，然后该位被硬件自动清零。该位对 CRC_FDATA 寄存器没有影响。 软件可读可写。

9. 真随机数生成器（TRNG）

9.1. 简介

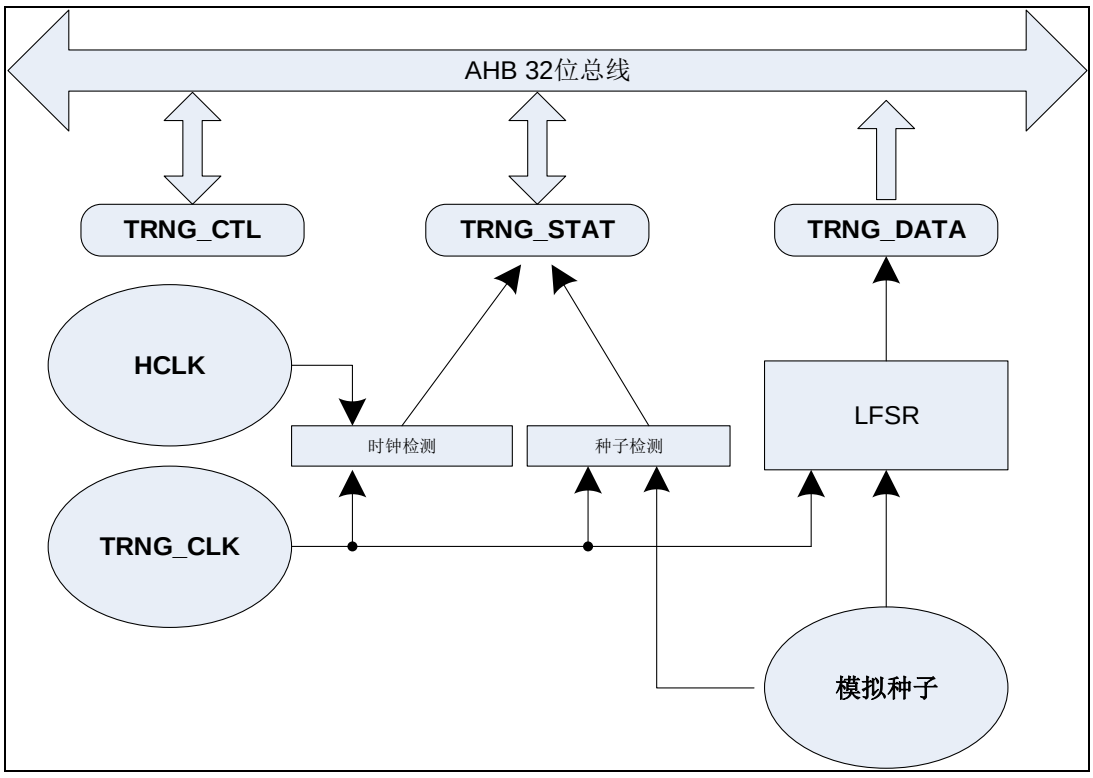
真随机数发生器模块（TRNG）能够通过连续模拟噪声生成一个 32 位的随机数值。

9.2. 主要特性

- 两个连续随机数的间隔大约为40个TRNG_CLK时钟周期；
- 可以关闭TRNG模块以降低芯片功耗；
- 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. 功能描述

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 IE 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 TRNGEN 位；

3）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。

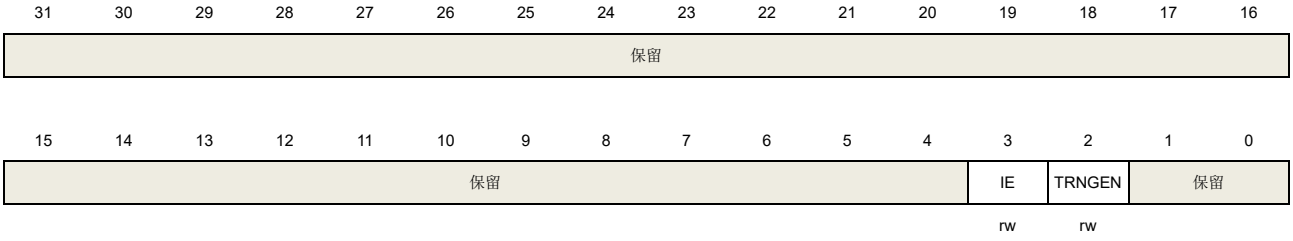
9.4. TRNG 寄存器

TRNG 基地址：0x4C06 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。

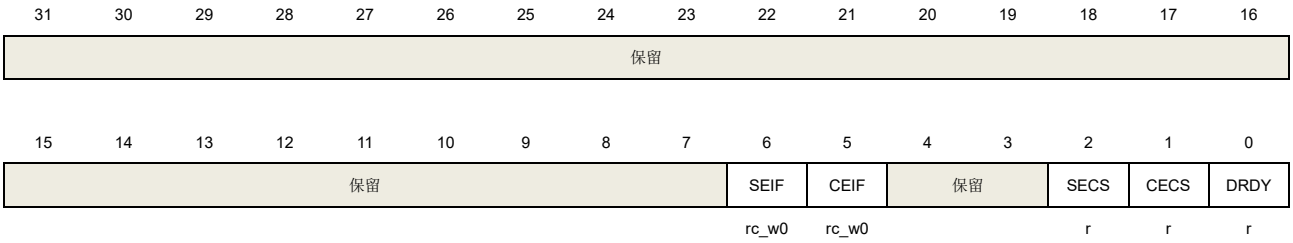


位/位域	名称	描述
31:4	保留	必须保持复位值。
3	IE	中断使能位，当 DRDY，SEIF 或 CEIF 位被置位时该位控制生成一个中断。 0：禁止 TRNG 中断 1：使能 TRNG 中断
2	TRNGEN	TRNG 使能位 0：禁止 TRNG 模块（降低功耗） 1：使能 TRNG 模块
1:0	保留	必须保持复位值。

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	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
TRNDATA[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRNDATA[15:0]															
r															

位/位域	名称	描述

31:0	TRNDATA[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的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 MCU 的介入，避免了 MCU 多次进入中断进行大规模的数据拷贝，最终提高整体的系统性能。

DMA 控制器包含了两个 AHB 总线接口和 8 个 4 字深度的 FIFO，使 DMA 可以高效的传输数据。DMA 控制器有 8 个通道，每个通道可以被分配给一个或多个特定的外设进行数据传输。两个内置的总线仲裁器用来处理 DMA 请求的优先级问题。

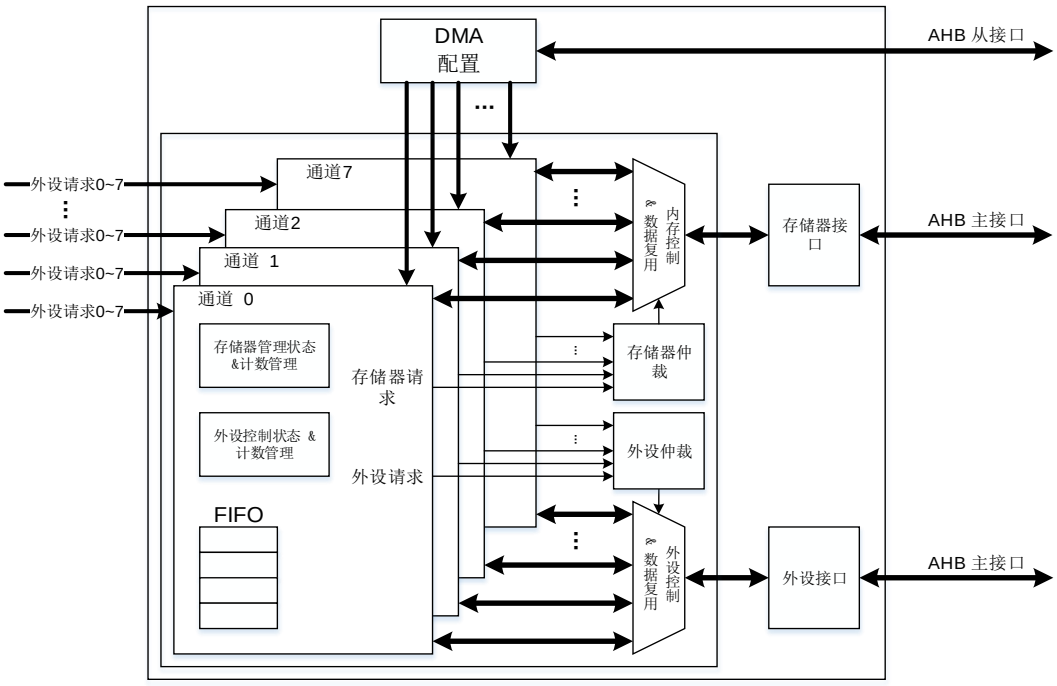
RISC-V 内核与 DMA 控制器都是通过系统总线来处理数据，引入仲裁机制来处理它们之间的竞争关系。当 MCU 和 DMA 指定相同的外设的时候，MCU 将会在特定的总线周期挂起。总线矩阵使用了轮询的算法保证 MCU 至少占用了一半的带宽。

10.2. 主要特征

- 两个 AHB 主机接口传输数据，一个 AHB 从机接口配置 DMA；
- DMA 控制器拥有 8 个通道，每个通道连接 8 个特定的外设请求；
- 存储器和外设支持单一传输，4 拍、8 拍和 16 拍增量突发传输；
- 当外设和存储器传输数据时，支持存储器切换；
- 支持软件优先级（低、中、高、超高）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；
- 支持三种传输方式：
  - 存储器到外设
  - 外设到存储器
  - 存储器到存储器
- DMA 和外设均可配置为传输控制器：
  - DMA 作为传输控制器：可配置数据传输长度，最大为 65535
  - 外设作为传输控制器：数据传输的完成取决于外设的最后一个传输请求
- 支持单数据传输和多数据传输模式，FIFO 深度最大为 4 个字：
  - 多数据传输模式：在存储器数据宽度和外设数据宽度不同的时候，自动打包/解包数据
  - 单数据传输模式：当且仅当 FIFO 空的时候从源地址读取数据，存进 FIFO，然后把 FIFO 的数据写到目标地址
- 每个通道有 5 种类型的事件标志和独立的中断；
- 支持中断的使能和清除。

10.3. 结构框图

图 10-1. 系统架构



如*图 10-1. 系统架构*所示，DMA 控制器由 4 部分组成：

- AHB 从接口配置 DMA；
- 通过两个 AHB 主接口分别为访问内存和访问外设提供数据传输功能；
- 两个仲裁器进行 DMA 请求的优先级管理；
- 数据处理和计数。

10.4. 功能说明

DMA 控制器在没有 MCU 参与的情况下从一个地址向另一个地址传输数据，它支持多种数据宽度，突发类型，地址生成算法，优先级和传输模式，可以灵活的配置以满足应用的需求。所有的寄存器都可以通过 AHB 从机接口进行 32 位的操作。

支持外设到存储器、存储器到外设以及存储器到存储器三种传输模式，具体模式选择通过 DMA_CHxCTL 寄存器中的 TM 位域决定，如*表 10-1. 传输模式*所示。

表 10-1. 传输模式

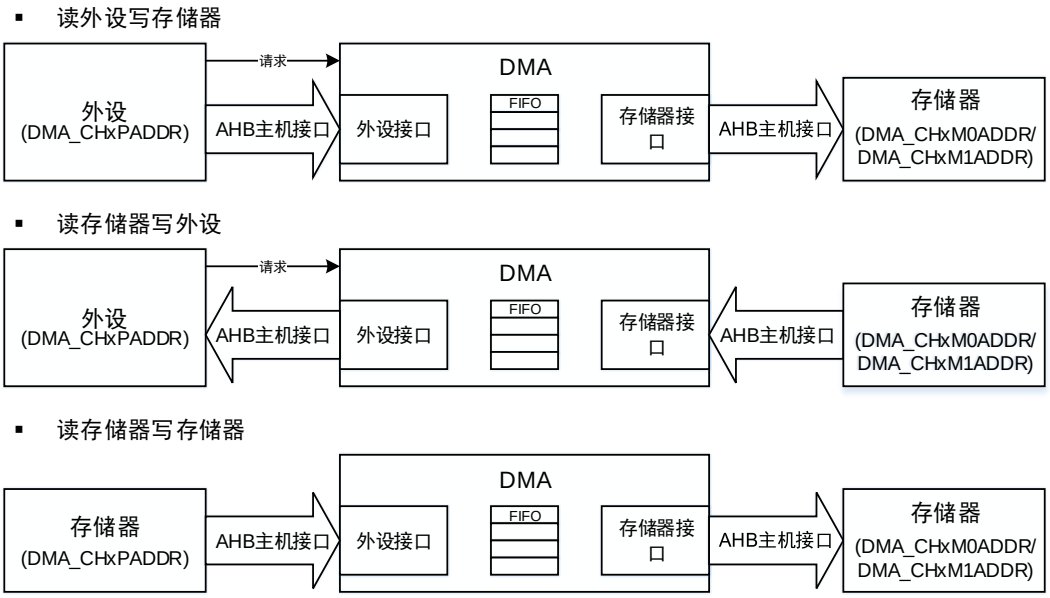
传输模式	TM[1:0]	源地址	目的地址
外设到存储器	00	DMA_CHxPADDR	DMA_CHxM0ADDR/ DMA_CHxM1ADDR
存储器到外设	01	DMA_CHxM0ADDR/ DMA_CHxM1ADDR	DMA_CHxPADDR
存储器到存储器	10	DMA_CHxPADDR	DMA_CHxM0ADDR/ DMA_CHxM1ADDR

注意：

1.寄存器 DMA_CHxCTL 的 MBS 位选择 DMA_CHxM0ADDR 或者 DMA_CHxM1ADDR 作为存储器地址。详细请参考*存储切换模式*;

2.寄存器 DMA_CHxCTL 的 TM 位域禁止配置成‘0b11’，否则通道将会自动关闭。

图 10-2. 三种传输模式的数据流



如*图 10-2. 三种传输模式的数据流*所示，DMA 控制器的两个 AHB 主机接口分别对应存储器和外设的数据访问。

- 外设到存储器：通过 AHB 外设主机接口从外设读取数据，通过 AHB 存储器主机接口向存储器写入数据；
- 存储器到外设：通过 AHB 存储器主机接口从存储器读取数据，通过 AHB 外设主机接口向外设写入数据；
- 存储器到存储器：通过 AHB 外设主机接口从存储器读取数据，通过 AHB 存储器主机接口向存储器写入数据。

10.4.1. 外设握手

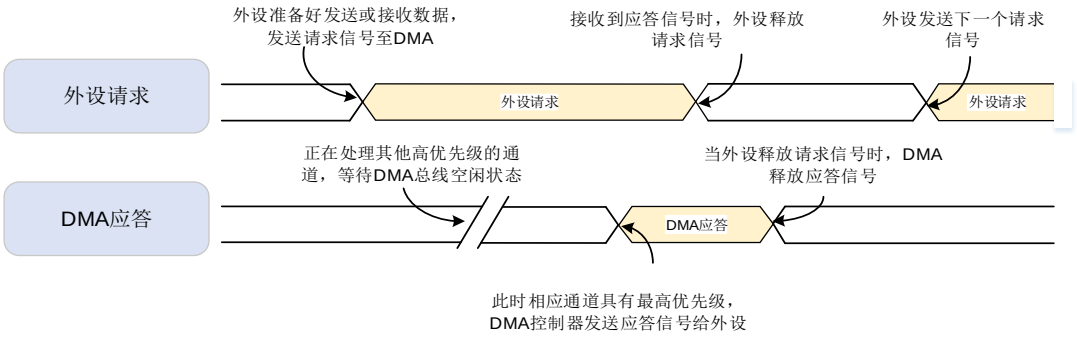
为了保证数据的有效传输，DMA 控制器中引入了外设和存储器的握手机制，包括请求信号和

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。

如*图 10-3. 握手机制*详细描述了 DMA 控制器与外设之间的握手机制。

图 10-3. 握手机制



DMA 控制器有 8 个通道，每个通道有多个外设请求。寄存器 DMA_CHxCTL 的 PERIEN 位域决定了 DMA 通道选中的外设请求。DMA0 与 DMA1 的外设请求映射分别列于*表 10-2. DMA 外设请求*。

如 DMA 的外设请求表*表 10-2. DMA 外设请求*所示，同一个外设请求可以连接到两个 DMA 通道上，这里禁止两个 DMA 通道选择相同的外设请求。例如，I2C0_RX 外设请求连接到通道 0 和通道 5。当寄存器 DMA_CH0CTL，DMA_CH5CTL 的 PERIEN 位域同时配置为‘0b001’时，使能通道 0 和通道 5，当 I2C0 发出 DMA 请求时，会造成通道 0 和通道 5 的响应混乱及数据传输错误。

表 10-2. DMA 外设请求

通道		通道 0	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7
PERIEN[2:0]	000	ADC	TIMER1_TG	TIMER2_TG	TIMER2_CH1	ADC0	•	TIMER0_CH0 TIMER0_CH1 TIMER0_CH2	•
	001	I2C0_RX	•	TIMER2_UP	TIMER2_CH2	•	I2C0_RX	I2C0_TX	I2C0_TX
	010	•	•	TIMER2_CH0	TIMER2_CH3	•	CAU_OUT	CAU_IN	HAU_IN
	011	SPI_RX	TIMER1_CH2 TIMER1_UP	SPI_RX	SPI_TX	SPI_TX	TIMER1_CH0	TIMER1_CH1 TIMER1_CH3	TIMER1_UP TIMER1_CH3
	100	USART1_RX	USART1_TX	USART0_RX	•	USART0_RX	USART2_RX	USART2_TX	USART0_TX
	101	QSPI	QSPI	TIMER15_CH0 TIMER15_UP	TIMER16_CH0 TIMER16_UP	•	•	TIMER15_CH0 TIMER15_UP	TIMER16_CH0 TIMER16_UP
	110	TIMER0_TG	TIMER0_CH0	TIMER0_CH1	TIMER0_CH0	TIMER0_CH3 TIMER0_TG TIMER0_CMT	TIMER0_UP	TIMER0_CH2	•
	111	•	TIMER5_UP	I2C1_RX	I2C1_RX	•	•	•	I2C1_TX

10.4.2. 数据处理

仲裁

每个 DMA 控制器有两个分别对应于外设和存储器的仲裁器。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。优先级规则如下：

- 软件优先级：分为4级，低，中，高和超高。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。

传输宽度，突发传输和计数

传输宽度

寄存器 DMA_CHxCTL 的 PWIDTH 和 MWIDTH 位域决定了外设和存储器的数据传输宽度。DMA 控制器支持 8 位，16 位和 32 位的数据宽度。在多数据传输模式中，如果 PWIDTH 和 MWIDTH 不相等，DMA 会自动的打包/解包数据来进行完整的数据传输。在单数据传输模式中，MWIDTH 在通道使能以后，会被硬件强制设置与 PWIDTH 相等。

突发传输类型

寄存器 DMA_CHxCTL 的 PBURST 和 MBURST 位域决定了外设和存储器的突发传输方式。DMA 控制器的外设和存储器接口均支持单一传输，4 拍，8 拍，16 拍的增量突发传输。对于单数据传输模式，当使能通道后，PBURST 和 MBURST 会被强制设为 0，仅支持单一传输。

在外设到存储器或者存储器到外设传输模式中，如果 PBURST 不为 0，在每次外设请求之后，DMA 控制器会根据 PBURST 的值进行 4 拍，8 拍，16 拍的增量突发传输。如果剩余的数据不够一次突发传输，剩余的数据将会进行单一传输。

AMBA 协议指定突发传输不能超过 1KB 的地址边界，否则会产生传输错误。对于外设和存储器，当突发传输超过 1KB 的地址边界，硬件会自动的把 4 拍，8 拍，16 拍（由 PBURST 和 MBURST 决定）的突发传输拆分为单一传输。

传输计数

当 DMA 进行传输前，寄存器 DMA_CHxCNT 的 CNT 位域决定了需要传输数据的数量，在使能 DMA 通道之前，数据的传输量必须完成配置。当外设作为传输控制器的时候，在使能通道后，CNT 位会被强制设置为‘0xFFFF’。在传输过程中，CNT 表示剩余需要传输的数据数量。

CNT 位的大小与外设的数据传输宽度有关，数据传输总量的字节数等于 CNT 乘以外设数据传输宽度。举例来说，如果 PWIDTH 的值设置为‘0b11’，则传输的数据总量的字节数等于 CNT*4。CNT 的值在外设的每次单一传输或者在突发模式中每个节拍传输完成后都会减 1。

CNT 值的配置需要满足下列要求：

如果关闭循环模式（清除寄存器 DMA_CHxCTL 的 CMEN 位），CNT 值的配置应该满足*表 10-3. CNT 配置*的要求。传输的数据总量的字节数必须是存储器数据传输宽度的整数倍，以保证完整的存储器传输；

注意：如果 PBURST 和 MBURST 都不是‘0b00’，传输的数据总量不用是存储器和外设的突发传输数据的整数倍。对于不满足一次突发传输的剩余数据，硬件会自动的拆分成多个单一传输。

表 10-3. CNT 配置

外设位宽	存储器位宽	CNT 值
8-bit	16-bit	2 的倍数
8-bit	32-bit	4 的倍数
16-bit	32-bit	2 的倍数
其他情况		任意值

如果开启循环模式（置位寄存器DMA_CHxCTL的CMEN位），传输的数据总量必须保证同时是

存储器突发传输数据总量和外设突发传输数据总量的整数倍，否则将不能保证数据的正确性。

$CNT / PBURST\_beats$ 必须是整数；

$(CNT \times PWIDTH\_bytes) / (MBURST\_beats \times MWIDTH\_bytes)$ 必须是整数。

- PWIDTH_bytes 是外设的数据传输宽度的字节数。8 位是 1，16 位是 2，32 位是 4；
- PBURST_beats 是外设突发传输的节拍数，单一传输是 1，4 拍增量突发传输是 4，8 拍增量突发传输是 8，16 拍增量突发传输是 16；
- MWIDTH_bytes 是存储器的数据传输宽度的字节数。8 位是 1，16 位是 2，32 位是 4；
- MBURST_beats 是存储器突发传输的节拍数，单一传输是 1，4 拍增量突发传输是 4，8 拍增量突发传输是 8，16 拍增量突发传输是 16。

举例如下：

1. 如果PWIDTH是16位，PBURST是4拍增量突发传输，MWIDTH是8位，MBURST是16拍增量突发传输，则CNT/4与（CNT*2）/（1*16）必须是整数，所以CNT必须是8的整数倍。
2. 如果PWIDTH是8位，PBURST是16拍增量突发传输，MWIDTH是16位，MBURST是4拍增量突发传输，则CNT/16与（CNT*1）/（2*4）必须是整数，所以CNT必须是16的倍数。

注意：如果使能了存储切换模式（置位寄存器 DMA_CHxCTL 的 SBMEN 位），循环模式会被硬件强制打开，所以也必须满足上述要求。

FIFO

DMA 控制器的每个通道都有一个 4 字深度的 FIFO 用于缓冲数据，从源地址读取的数据会先暂时保存在 FIFO 中，再传输到目的地址。根据 FIFO 的配置，DMA 控制器支持两种数据处理模式：单数据传输模式和多数据传输模式。在存储器到存储器模式下，DMA 控制器仅支持多数据传输模式。

多数据传输模式

通过把寄存器 DMA_CHxFCTL 的 MDMEN 位置 1 来开启多数据传输模式。

在这个模式中，当 FIFO 有足够的空间时，DMA 控制器响应源端的请求，从源地址读取数据存储进 FIFO。如果目的端是外设，当 FIFO 内的数据量满足外设的一次突发传输时，DMA 会响应外设的请求。如果目标端是存储器，寄存器 DMA_CHxFCTL 的 FCCV 位设置的 FIFO 临界值决定 DMA 控制器何时进行将 FIFO 中的数据写入存储器，当 FIFO 计数器达到配置的临界值时，FIFO 中的所有数据会被写入目标存储器地址。

为了保证正确的数据传输，FIFO 的临界值必须配置为存储器一次突发传输数据量的整数倍。这样才能保证 FIFO 中有足够的数据可以完成存储器突发传输。FIFO 计数器的临界值的设置与存储器数据传输宽度和存储器突发传输类型有关，具体见*表 10-4. FIFO 计数器临界值配置*。

表 10-4. FIFO 计数器临界值配置

MWIDTH	MBURST	FIFO 计数临界值			
		1 个字	2 个字	3 个字	4 个字

8-bit	single	4 次单一传输	8 次单一传输	12 次单一传输	16 次单一传输
	INCR4	1 次突发传输	2 次突发传输	3 次突发传输	4 次突发传输
	INCR8	错误	1 次突发传输	错误	2 次突发传输
	INCR16	错误	错误	错误	1 次突发传输
16-bit	single	2 次单一传输	4 次单一传输	6 次单一传输	8 次单一传输
	INCR4	错误	1 次突发传输	错误	2 次突发传输
	INCR8	错误	错误	错误	1 次突发传输
	INCR16	错误	错误	错误	错误
32-bit	single	1 次单一传输	2 次单一传输	3 次单一传输	4 次单一传输
	INCR4	错误	错误	错误	1 次突发传输
	INCR8	错误	错误	错误	错误
	INCR16	错误	错误	错误	错误

注意：当传输模式是外设到存储器时，如果 PBURST_beats × PWIDTH_bytes = 16，FIFO 计数器临界值不能设置成‘0b10’。如果设置成‘0b10’，当接收到外设请求时，DMA 控制器从外设中读取数据并填满 FIFO，然后 DMA 会向存储器中写入 3 个字的数据（这个是由 FIFO 的临界值决定），同时剩余一个字的数据。这时当新的外设请求来临时，FIFO 中没有足够的空间进行下一次的外设突发传输，同时 FIFO 中的数据没有达到 FIFO 临界值也不会进行存储器突发传输，会使通道数据传输冻结。

单数据传输模式

通过把寄存器 DMA_CHxFCTL 的 MDMEN 位清 0 来开启单数据传输模式。

在这个模式中，DMA 控制器一次只能传输一个数据，FIFO 计数器临界值的配置（寄存器 DMA_CHxFCTL 的 FCCV 位域）没有意义。在单数据传输模式中，当且仅当 FIFO 空的时候，DMA 会响应源端的请求，从源地址读取数据进入 FIFO。当 FIFO 非空时，DMA 响应目的端的请求，把 FIFO 中的数据写入目的地址。

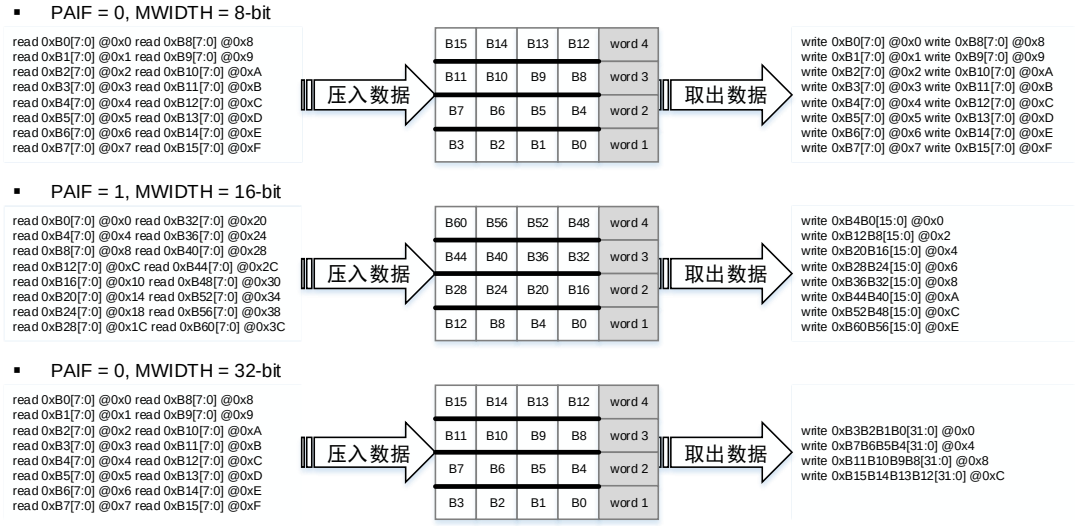
打包/解包

在单数据传输模式中，MWIDTH 会被硬件强制设置与 PWIDTH 相等，无需使用数据的打包/解包功能。

在多数据传输模式中，MWIDTH 与 PWIDTH 相互独立，配置更为灵活。当 MWIDTH 与 PWIDTH 不相等时，DMA 的读写传输宽度不同，DMA 会自动的对数据打包/解包操作。在传输过程中，外设和寄存器都只支持小端操作。

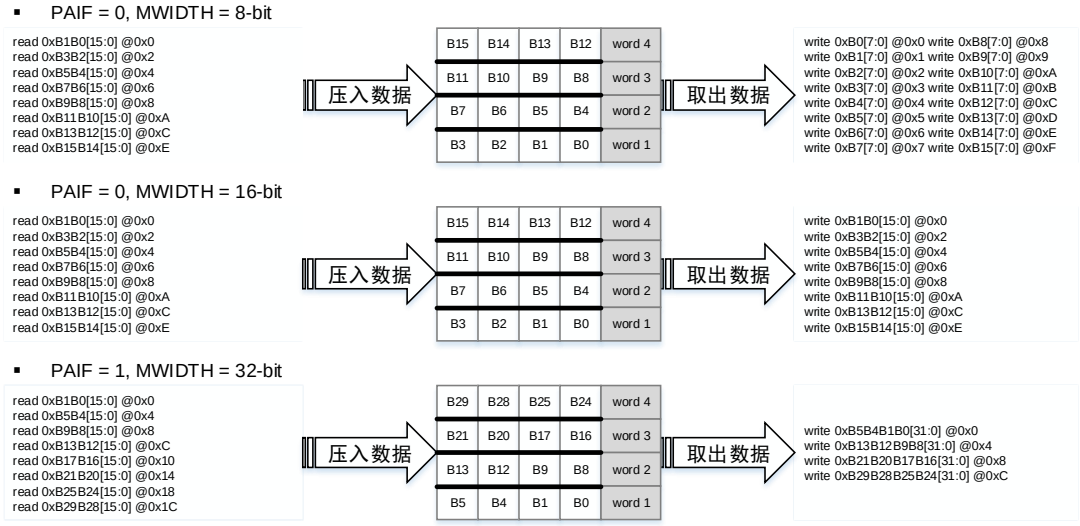
假设当 CNT 被设置为 16，PWIDTH 为 ‘0b00’，PNAGA 和 MNAGA 被置 1。对于不同的 MWIDTH，DMA 的传输操作如*图 10-4. PWIDTH 为‘0b00’时，数据的打包 / 解包*所示。

图 10-4. PWIDTH 为‘0b00’时，数据的打包 / 解包



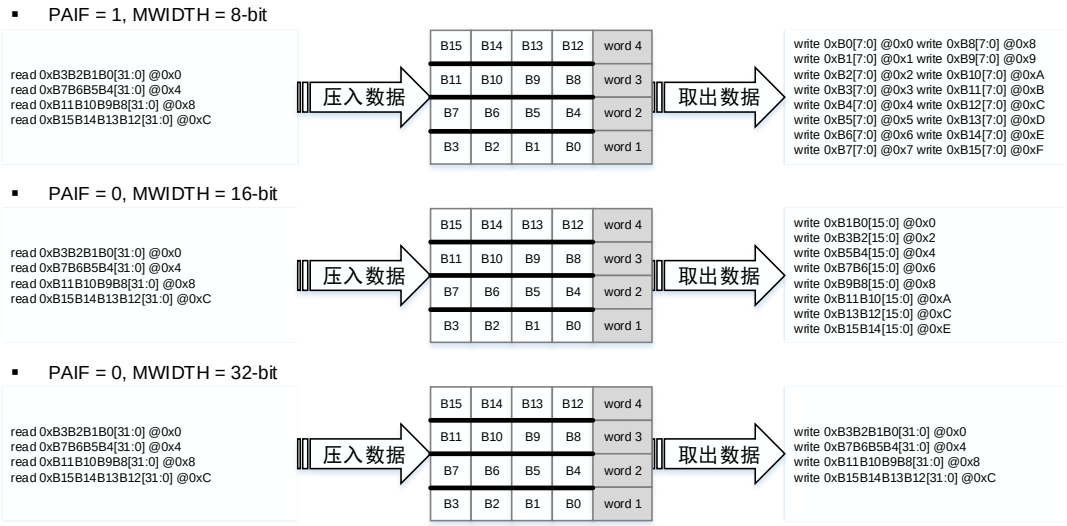
假设当 CNT 被设置为 8，PWIDTH 为‘0b01’，PNAGA 和 MNAGA 被置 1。对于不同的 WIDTH，DMA 的传输操作如*图 10-5. PWIDTH 为‘0b01’时，数据的打包/解包*所示。

图 10-5. PWIDTH 为‘0b01’时，数据的打包/解包



- 当 CNT 被设置为 4，PWIDTH 为 ‘0b10’，PNAGA 和 MNAGA 被置 1。对于不同的 MWIDTH，DMA 的传输操作如*图 10-6. PWIDTH 为‘0b10’时，数据的打包/解包*所示。

图 10-6. PWIDTH 为‘0b10’时，数据的打包/解包



10.4.3. 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_CHxM0ADDR，DMA_CHxM1ADDR）。

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。在多数据传输模式中，若寄存器 DMA_CHxCTL 的 PBURST 配置为‘0b00’，当寄存器 DMA_CHxCTL 的 PAIF 位置 1 使能时，外设的下一次传输的地址增量被固定为 4，与外设的数据传输宽度无关。PAIF 与存储器地址生成无关。

注意：若 DMA_CHxCTL 寄存器中的 PAIF 位配置为’1‘，外设的基地址（寄存器 DMA_CHxPADDR）必须配置为 4 字节对齐。

10.4.4. 循环模式

循环模式用来处理连续的外设请求。可以通过寄存器 DMA_CHxCTL 的 CMEN 位置 1 使能。循环模式只在 DMA 作为传输控制器时有效。当寄存器 DMA_CHxCTL 的 TFCS 位被置 1 时，外设作为传输控制器，在通道使能后，循环模式会被自动关闭。

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，直到出现传输错误或者通道使能位被清 0。

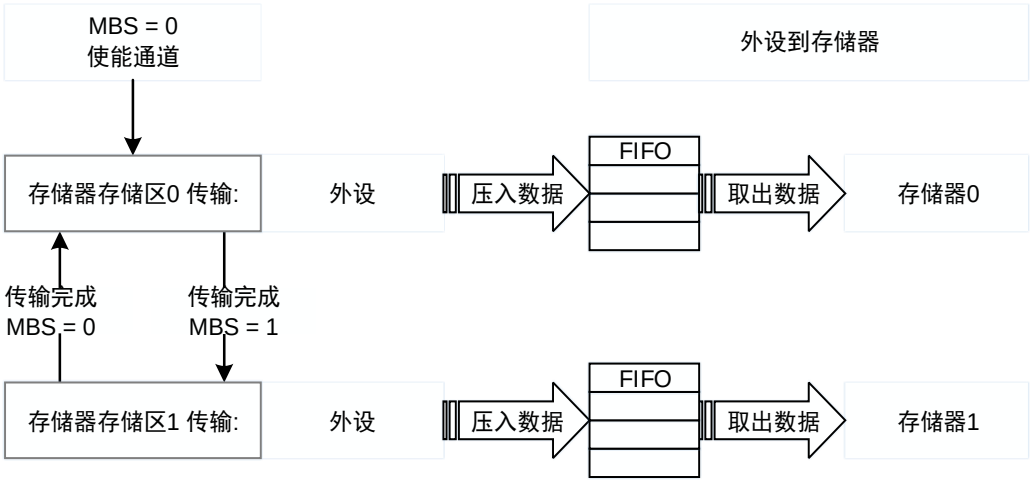
10.4.5. 存储切换模式

与循环模式相同，存储切换模式也是用来处理连续的外设请求。可以通过寄存器 DMA_CHxCTL 的 SBMEN 位置 1 使能。若打开了存储切换模式，在通道使能后，硬件会自动打开循环模式。存储切换模式只能应用于外设与存储器之间的数据传输，在存储器到存储器模式中禁止使用。

存储切换模式支持两个存储器缓冲区，两个存储器基地址可以分别在寄存器 DMA_CHxM0ADDR 和 DMA_CHxM1ADDR 中配置。在每次 DMA 传输完成后，存储器指针指向另一个存储器缓冲区。在 DMA 传输过程中，没有被 DMA 占用的缓冲区可以被其他的 AHB 主机接口操作，且其基地址可以改变。

软件可以通过设定寄存器 DMA_CHxCTL 的 MBS 位来指定第一次数据传输 DMA 使用的缓冲区。DMA 通道使能以后，MBS 可以视为 DMA 存储器缓冲区的标志位，它会在每次传输完成后自动在'0'，'1'之间切换，DMA 存储切换模式操作如*图 10-7. 存储切换模式*所示。

图 10-7. 存储切换模式



10.4.6. 传输控制器

数据传输量的大小由传输控制器决定。寄存器 DMA_CHxCTL 的 TFCS 位决定了传输控制器是外设还是 DMA。

- DMA 为传输控制器：寄存器 DMA_CHxCNT 的 CNT 位域决定传输数据量的大小，必须在通道使能前配置；
- 外设为传输控制器：在通道使能后寄存器 DMA_CHxCNT 的 CNT 位域为会被硬件强制配置为'FFFF'，因此配置 CNT 没有意义。DMA 数据传输完成由外设发送最后一次传输请求决定。

注意：当传输模式是存储器到存储器时，传输控制器只能是 DMA。

10.4.7. 传输操作

数据传输支持三种操作方式：外设到存储器，存储器到外设，和存储器到存储器。存储器和外设都可以配置为源端和目的端。

存储器端数据传输

- 外设到存储器：
  - 单数据传输模式，当 FIFO 非空时，DMA 启动存储器数据传输，写数据到相应的存储器地

址中

- 多数据传输模式，当 FIFO 计数器达到临界值时，DMA 启动单一或突发数据传输，把 FIFO 的数据全部写入存储器中

■ 存储器到外设：

- 单数据传输模式，当通道使能时 DMA 会立刻进行存储器数据传输，读取数据到 FIFO。数据传输过程中，当且仅当 FIFO 为空时，DMA 控制器就会进行存储器读取操作
- 多数据传输模式，当通道使能后，不论是否有外设请求，DMA 都会进行单一或突发数据传输填满 FIFO。在数据传输过程中，当 FIFO 有足够的空间进行一次单一或突发传输时，DMA 控制器就会进行存储器读取操作

■ 存储器到存储器：只支持多数据传输模式。当 FIFO 计数器到达临界值，DMA 进行单一或突发传输把 FIFO 的数据全部写入存储器中。

外设端数据传输

■ 外设到存储器：当 DMA 收到外设请求且 FIFO 有足够的空间进行数据传输，DMA 启动外设数据传输从外设读取数据写入 FIFO；

■ 存储器到外设：当 DMA 收到外设请求且 FIFO 有足够的数据进行数据传输，DMA 启动外设数据传输从 FIFO 读取数据写入外设；

■ 存储器到存储器：只支持多数据传输模式。当通道使能时，DMA 启动单一或突发传输读取数据写满 FIFO。在数据传输过程中，当 FIFO 有足够的空间进行一次单一或突发传输时，DMA 控制器就会进行存储器读取操作。

10.4.8. 传输完成

DMA 传输由硬件自动完成，寄存器 DMA_INTF0 与 DMA_INTF1 位 FTFIFx 在以下情况下会被置 1：

■ 传输完成；

■ 软件清除；

■ 传输错误。

传输完成

当 DMA 使能以后，数据会在外设和存储器之间传输。当寄存器 DMA_CHxCNT 的 CNT 中配置的数据量传输完成或处理完最后一次外设请求以后，DMA 传输结束，寄存器 DMA_CHxCTL 的 CHEN 位自动清零。

■ 外设到存储器：如果 DMA 是传输控制器，CNT 减数到 0 且 FIFO 中的数据完全写入到存储器中，传输完成。如果外设是传输控制器，当外设的最后一个请求完成且 FIFO 中的数据完全写入到存储器中，传输完成；

■ 存储器到外设：如果 DMA 是传输流控制器，当 DMA_CHxCNT 寄存器中的 CNT 位域自减到 0 时传输完成。如果外设作为传输流控制器，当外设的最后一个请求完成时，传输完成；

■ 存储器到存储器：只支持 DMA 为传输控制器，CNT 减数到 0 且 FIFO 中的数据完全写入到存储器中，传输完成。

软件清除

DMA 传输可以通过对寄存器 DMA_CHxCTL 的 CHEN 位清 0 停止。在清 0 操作之后，若 CHEN 仍然为 1，代表存储器或者外设仍然处在传输状态，或者 FIFO 中还有剩余的数据没有传输完成。

- 外设到存储器：软件清 0 操作后，当前的单次或突发传输完成后，DMA 的外设传输将会停止。为了保证从外设读取的数据完全被写入存储器中，存储器在 FIFO 非空的状态下仍然会进行数据传输，直到 FIFO 中的数据完全被写入存储器中。若 FIFO 中剩余的数据量不满足一次存储器突发传输，这些数据将会被拆分成单一传输。如果 FIFO 总剩余的数据量小于存储器传输宽度，这个数据会被高位补 0，写入存储器中。此时读取 CNT 的值可以计算出存储器中的有效数据量。在 FIFO 中的数据传输完毕之后，CHEN 会被硬件自动清 0，寄存器 DMA_INTF0 或 DMA_INTF1 相应通道标志位 FTFIFx 会被置 1；
- 存储器到外设：软件清 0 操作后，当前的存储器和外设传输完成以后，DMA 传输将会停止。CHEN 会自动清 0，寄存器 DMA_INTF0 或 DMA_INTF1 相应通道标志位 FTFIFx 会被置 1；
- 存储器到存储器：与外设到存储器相同，其中源端存储器的传输通过外设端口来实现。

传输错误

三种类型的错误会关闭 DMA 传输：

- FIFO 错误：当检测到 FIFO 错误配置，通道会立即关闭且不会进行任何传输。这种情况下，FTFIFx 不会被置 1。更多 FIFO 错误的信息，请参考*错误*小节；
- 总线错误：当存储器或者外设端口试图访问超出范围的地址时，DMA 控制器会检测到总线错误，通道停止传输且 FTFIFx 不会被置 1。如果错误是由外设端口引起，CNT 仍会进行一次减 1 操作。更多总线错误的信息，请参考*错误*小节；
- 寄存器访问错误：在存储切换模式下，当对 DMA 正在访问的存储器的基地址寄存器进行写操作时，DMA 控制器会检测到寄存器访问错误。发生这个错误后，DMA 控制器的操作与软件清 0 时相同。更多寄存器访问错误的信息，请参考*错误*小节。

10.4.9. 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 0 或等待 DMA 传输完成。当 CHEN 为 0 时，请按照下列步骤配置 DMA；
2. 清除寄存器 DMA_INTF0 或 DMA_INTF1 相应通道标志位 FTFIFx，否则无法使能 DMA；
3. 配置寄存器 DMA_CHxCTL 的 TM 位选择数据传输方式；
4. 配置寄存器 DMA_CHxCTL 的 PERIEN 位域选择外设。当数据传输方式是存储器到存储器时，PERIEN 没有具体意义，这一步可以跳过；
5. 在寄存器 DMA_CHxCTL 中配置存储器和外设突发类型，目标存储器（memory 0 或 memory1），存储切换模式，通道优先级，存储器和外设的传输宽度，存储器和外设的地址生成算法，循环模式，传输控制器；
6. 在寄存器 DMA_CHxFCTL 中配置数据处理方式，如果使用多数据传输方式，需要配置 FCCV 位域以设置 FIFO 计数器临界值；

7. 在寄存器 DMA_CHxCTL 中配置传输完成中断，半传输完成中断，传输错误中断，单数据传输方式异常中断的使能位。在寄存器 DMA_CHxFCTL 中配置 FIFO 错误和异常中断的使能位。中断使能位可根据实际需求配置；
8. 在寄存器 DMA_CHxPADDR 中配置外设基地址；
9. 如果使用存储切换模式，在寄存器 DMA_CHxM0ADDR 和 DMA_CHxM1ADDR 中配置两个存储器基地址。如果只使用一个存储器，寄存器 CHxCTL 的 MBS 位决定配置 DMA_CHxM0ADDR 或者 DMA_CHxM1ADDR；
10. 在寄存器 DMA_CHxCNT 中配置数据传输总量；
11. 寄存器 DMA_CHxCTL 的 CHEN 位置 1，使能 DMA 通道。

如果要继续被暂停的 DMA 传输，建议遵循以下步骤进行操作：

1. 读取 CHEN 位，确定 DMA 的挂起操作已经完成。当 CHEN 为 0 时，DMA 处于空闲状态，可以重新配置 DMA 以继续挂起 DMA 传输；
2. 清除寄存器 DMA_INTF0 或 DMA_INTF1 相应通道标志位 FTFIFx，否则 DMA 通道可能无法再使能；
3. 读取寄存器 DMA_CHxCNT 计算出已经发送的数据量与剩余待发的数据量；
4. 在寄存器 DMA_CHxPADDR 中更新外设基地址；
5. 在寄存器 DMA_CHxM0ADDR 或 DMA_CHxM1ADDR 中更新存储器基地址；
6. 在寄存器 DMA_CHxCNT 中配置剩余待发的数据总量；
7. 寄存器 DMA_CHxCTL 的 CHEN 位置 1，重新启动 DMA 通道。

10.5. 中断

每个 DMA 通道都有专有的中断，包括 5 个中断事件：传输完成中断，半传输完成中断，传输错误中断，单数据传输模式异常中断，FIFO 错误和异常中断。任何一个中断事件都可以引发 DMA 中断。

寄存器 DMA_INTF0 或 DMA_INTF1 包含每个中断事件的标志位，寄存器 DMA_INTC0 或 DMA_INTC1 包含每个中断事件的标志清除位，寄存器 DMA_CHxCTL 和 DMA_CHxFCTL 包含每个中断事件的使能位，具体如*表 10-5. DMA 中断事件*所示。

表 10-5. DMA 中断事件

中断事件	标志位	使能位	清除位
	DMA_INTF0 或 DMA_INTF1	DMA_CHxCTL 或 DMA_CHxFCTL	DMA_INTC0 或 DMA_INTC1
传输完成	FTFIF	FTFIE	FTFIFC
半传输完成	HTFIF	HTFIE	HTFIFC
传输错误	TAEIF	TAEIE	TAEIFC
单数据模式异常	SDEIF	SDEIE	SDEIFC
FIFO 错误与异常	FEEIF	FEPIE	FEEIFC

这 5 个事件可以分为 3 种类型：

- 标志：传输完成和半传输完成；
- 异常：单数据传输模式异常和 FIFO 异常；

- 错误：传输错误和 FIFO 错误。

发生异常事件时，正在进行的 DMA 传输不会被停止，仍将继续传输。发生错误事件时，正在进行的 DMA 传输会被停止。这三种类型的事件在下一节进行详细描述。

10.5.1. 标志

两种标志事件，传输完成事件和半传输完成事件。

发生以下情况时，传输完成标志位将会被置 1：

- 当 DMA 作为传输控制器时，CNT 计数到 0；
- 当外设作为传输控制器时，响应完外设的最后一个数据传输请求后，（如果是读外设写存储器传输方式，还需满足 FIFO 中的数据全部写入存储器中）传输完成；
- 在数据传输完成之前，通过软件清 0 的方式停止数据传输，当前的存储器和外设数据传输完成后，（如果是外设到存储器或存储器到存储器传输模式，还需满足 FIFO 中的数据全部写入存储器中）传输完成；
- 在数据传输完成之前，由于寄存器访问错误导致停止数据传输，当前的存储器和外设数据传输完成后，（如果是外设到存储器或存储器到存储器传输模式，还需满足 FIFO 中的数据全部写入存储器，）传输完成。

当传输完成标志位置 1，且传输完成中断使能时，DMA 控制器产生传输完成中断。

当 DMA 作为传输控制器且 CNT 减数计数达到初始值的一半时，半传输完成标志位会被置 1。当外设作为传输控制器时，DMA 无法得知是否已经传输一半的数据流，此时半传输完成标志仍为 0。

当半传输完成标志位置 1，且半传输完成中断使能时，DMA 控制器产生半传输完成中断。

10.5.2. 异常

两种异常事件，单数据传输模式异常和 FIFO 异常。异常对于 DMA 传输无影响。

单数据传输模式异常

这个异常只有在使能单数据传输模式且传输方式为外设到存储器时才会发生。当 FIFO 非空时，如果外设请求数据传输，但是 DMA 响应后未能获得总线授权，此时单数据传输模式异常标志位置 1。

当单数据传输模式异常标志位置 1，且单数据传输模式异常中断使能，DMA 控制器产生单数据传输模式异常中断。

FIFO 异常

这个异常只有数据在外设和存储器之间传输才会发生，当 FIFO 发生上溢或下溢时，FIFO 异常标志位置 1。

当传输模式为外设到存储器时，如果外设请求有效且得到最高优先级时，FIFO 的剩余空间不

满足单一或突发外设传输，FIFO 发生上溢。直到 FIFO 有足够空间时，DMA 控制器才会响应此次外设请求，该异常不会影响到数据传输的正确性。

当传输模式为存储器到外设时，如果外设请求有效且得到最高优先级时，FIFO 中的数据不够完成单次或突发外设传输，FIFO 发生下溢。直到 FIFO 有足够数据时，DMA 控制器才会响应此次外设请求，该异常不会影响到数据传输的正确性。

当 FIFO 异常标志位置 1，且 FIFO 异常中断使能时，DMA 控制器产生 FIFO 异常中断。

10.5.3. 错误

在数据传输过程中，会发生 FIFO 错误和传输错误（包含寄存器访问错误和总线错误），此时数据传输会被中止。

FIFO 错误

当使用多数据传输模式时，FIFO 计数器临界值设置必须与存储器和外设数据传输宽度匹配，详细见*表 10-4. FIFO 计数器临界值配置*。错误的配置会引发 FIFO 错误，此时，通道会立即关闭，并不启动任何传输。

当 FIFO 错误标志位置 1，且 FIFO 错误中断使能时，DMA 控制器产生 FIFO 错误中断。

寄存器访问错误

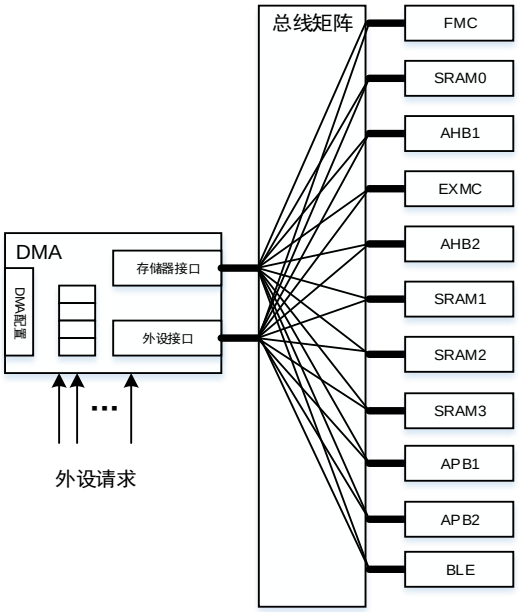
只有在存储切换模式下才会发生寄存器访问错误。如果软件对 DMA 正在使用的存储器的基地址寄存器进行写操作，将会发生寄存器访问错误。举例来说，存储器 0 是 DMA 控制器正在使用的源端或者目的端地址，如果软件对 DMA_CHxM0ADDR 寄存器进行写操作，则会产生寄存器访问错误。寄存器访问错误发生后，当前数据传输完成之后（在读外设写存储器传输模式下，FIFO 中的数据需要全部写入到 memory 中），DMA 会被自动停止。

当寄存器访问错误标志位置 1，且寄存器访问错误中断使能时，DMA 控制器产生寄存器访问错误中断。

总线错误

当 DMA 的存储器端或外设端的主机端口访问的地址超出了允许的范围，会发生总线错误，同时 DMA 通道失能。DMA0 和 DMA1 的存储器和外设端口允许访问的地址空间如*图 10-8. DMA 的系统连接*所示。

图 10-8. DMA 的系统连接



10.6. DMA 寄存器

DMA 基地址：0x4002 6000

10.6.1. 中断标志位寄存器 0（DMA_INTF0）

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				FTFIF3	HTFIF3	TAEIF3	SDEIF3	保留	FEEIF3	FTFIF2	HTFIF2	TAEIF2	SDEIF2	保留	FEEIF2
				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
保留				FTFIF1	HTFIF1	TAEIF1	SDEIF1	保留	FEEIF1	FTFIF0	HTFIF0	TAEIF0	SDEIF0	保留	FEEIF0
				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。
27/21/11/5	FTFIFx	通道x的传输完成标志位（x=0…3） 硬件置位，软件写DMA_INTC0相应位为1清零 0：通道x传输未完成 1：通道x传输完成
26/20/10/4	HTFIFx	通道x的半传输完成标志位（x=0…3） 硬件置位，软件写DMA_INTC0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
25/19/9/3	TAEIFx	通道x的传输错误标志位（x=0…3） 硬件置位，软件写DMA_INTC0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
24/18/8/2	SDEIFx	通道x的单数据传输模式异常标志位（x=0…3） 硬件置位，软件写DMA_INTC0相应位为1清零 0：通道x未发生单数据传输模式异常 1：通道x发生单数据传输模式异常
23/17/7/1	保留	必须保留复位值。
22/16/6/0	FEEIFx	通道x的FIFO错误与FIFO异常标志位（x=0…3） 硬件置位，软件写DMA_INTC0相应位为1清零 0：通道x未发生FIFO错误或FIFO异常 1：通道x发生FIFO错误或FIFO异常

10.6.2. 中断标志位寄存器 1（DMA_INTF1）

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				FTFIF7	HTFIF7	TAEIF7	SDEIF7	保留	FEEIF7	FTFIF6	HTFIF6	TAEIF6	SDEIF6	保留	FEEIF6
				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
保留				FTFIF5	HTFIF5	TAEIF5	SDEIF5	保留	FEEIF5	FTFIF4	HTFIF4	TAEIF4	SDEIF4	保留	FEEIF4
				r	r	r	r		r	r	r	r	r		r

位/位域	名称	描述
31:28	保留	必须保留复位值。
27/21/11/5	FTFIFx	通道x的传输完成标志位（x=4…7） 硬件置位，软件写DMA_INTC1相应位为1清零 0：通道x传输未完成 1：通道x传输完成
26/20/10/4	HTFIFx	通道x的半传输完成标志位（x=4…7） 硬件置位，软件写DMA_INTC1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
25/19/9/3	TAEIFx	通道x的传输错误标志位（x=4…7） 硬件置位，软件写DMA_INTC1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
24/18/8/2	SDEIFx	通道x的单数据传输模式异常标志位（x=4…7） 硬件置位，软件写DMA_INTC1相应位为1清零 0：通道x未发生单数据传输模式异常 1：通道x发生单数据传输模式异常
23/17/7/1	保留	必须保留复位值。
22/16/6/0	FEEIFx	通道x的FIFO错误与FIFO异常标志位（x=4…7） 硬件置位，软件写DMA_INTC1相应位为1清零 0：通道x未发生FIFO错误或FIFO异常 1：通道x发生FIFO错误或FIFO异常

10.6.3. 中断标志位清除寄存器 0（DMA_INTC0）

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				FTFIFC3	HTFIFC3	TAEIFC3	SDEIFC3	保留	FEEIFC3	FTFIFC2	HTFIFC2	TAEIFC2	SDEIFC2	保留	FEEIFC2
				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
保留				FTFIFC1	HTFIFC1	TAEIFC1	SDEIFC1	保留	FEEIFC1	FTFIFC0	HTFIFC0	TAEIFC0	SDEIFC0	保留	FEEIFC0
				w	w	w	w		w	w	w	w	w		w

位/位域	名称	描述
31:28	保留	必须保留复位值。
27/21/11/5	FTFIFCx	通道x的传输完成标志清除位（x=0…3） 0：无影响 1：清除传输完成标志位
26/20/10/4	HTFIFCx	通道x的半传输完成标志清除位（x=0…3） 0：无影响 1：清除半传输完成标志位
25/19/9/3	TAEIFCx	通道x的传输错误标志清除位（x=0…3） 0：无影响 1：清除传输错误标志位
24/18/8/2	SDEIFCx	通道x的单数据传输模式异常标志清除位（x=0…3） 0：无影响 1：清除单数据传输模式异常标志位
23/17/7/1	保留	必须保留复位值。
22/16/6/0	FEEIFCx	通道x的FIFO错误与FIFO异常标志清除位（x=0…3） 0：无影响 1：清除FIFO错误与FIFO异常标志位

10.6.4. 中断标志位清除寄存器 1（DMA_INTC1）

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				FTFIFC7	HTFIFC7	TAEIFC7	SDEIFC7	保留	FEEIFC7	FTFIFC6	HTFIFC6	TAEIFC6	SDEIFC6	保留	FEEIFC6
				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
保留				FTFIFC5	HTFIFC5	TAEIFC5	SDEIFC5	保留	FEEIFC5	FTFIFC4	HTFIFC4	TAEIFC4	SDEIFC4	保留	FEEIFC4
				w	w	w	w		w	w	w	w	w		w

位/位域	名称	描述
31:28	保留	必须保留复位值。
27/21/11/5	FTFIFCx	通道x的传输完成标志清除位（x=4…7） 0：无影响 1：清除传输完成标志位
26/20/10/4	HTFIFCx	通道x的半传输完成标志清除位（x=4…7） 0：无影响 1：清除半传输完成标志位
25/19/9/3	TAEIFCx	通道x的传输错误标志清除位（x=4…7） 0：无影响 1：清除传输错误标志位
24/18/8/2	SDEIFCx	通道x的单数据传输模式异常标志清除位（x=4…7） 0：无影响 1：清除单数据传输模式异常标志位
23/17/7/1	保留	必须保留复位值。
22/16/6/0	FEEIFCx	通道x的FIFO错误与FIFO异常标志清除位（x=4…7） 0：无影响 1：清除FIFO错误与FIFO异常标志位

10.6.5. 通道 x 控制寄存器（DMA_CHxCTL）（x = 0..7）

地址偏移：0x10 + 0x18 * 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				PERIEN[2:0]			MBURST[1:0]		PBURST[1:0]		保留	MBS	SBMEN	PRIO[1:0]	
					rw		rw		rw			rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAIF	MWIDTH[1:0]		PWIDTH[1:0]		MNAGA	PNAGA	CMEN	TM[1:0]		TFCS	FTFIE	HTFIE	TAEIE	SDEIE	CHEN
rw	rw		rw		rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位/位域	名称	名称
31:28	保留	必须保留复位值。
27:25	PERIEN[2:0]	外设使能 软件置1与清0。 000：使能外设0 001：使能外设1 010：使能外设2 011：使能外设3

		100：使能外设4 101：使能外设5 110：使能外设6 111：使能外设7 CHEN为1时不可写入。
24:23	MBURST[1:0]	存储器突发类型 软件置1与清0。 00：单一传输 01：INCR4 (4拍增量突发传输) 10：INCR8 (8拍增量突发传输) 11：INCR16 (16拍增量突发传输) CHEN为1时不可写入。 如果寄存器DMA_CHxFCTL的MDMEN位为0，在使能通道后（CHEN置1），该位域会被硬件强制清零
22:21	PBURST[1:0]	外设突发类型 软件置1与清0。 00：单一传输 01：INCR4 (4拍增量突发传输) 10：INCR8 (8拍增量突发传输) 11：INCR16 (16拍增量突发传输) CHEN为1时不可写入。 如果寄存器DMA_CHxFCTL的MDMEN位为0，在使能通道后（CHEN置1），该位域会被硬件强制清零
20	保留	必须保留复位值。
19	MBS	存储器缓冲选择 硬件置1清0，软件置1清0。 0：存储器0作为存储器传输区域 1：存储器1作为存储器传输区域 CHEN为1时不可写入。 在每次传输完成时，硬件会自动更新该位，以此来表明DMA正在使用哪个存储区
18	SBMEN	存储切换模式使能 软件置1与清0。 0：关闭存储切换模式 1：打开存储切换模式 CHEN为1时不可写入。
17:16	PRIO[1:0]	软件优先级 软件置1与清0。 00：低 01：中 10：高

		11：超高 CHEN为1时不可写入。
15	PAIF	外设地址增量固定 软件置1与清0。 0：外设地址增量由PWIDTH决定 1：外设地址增量固定为4 CHEN为1时不可写入。 如果PNAGA设置为0，该位无影响 如果寄存器DMA_CHxFCTL的MDMEN位为‘0’或者PBURST不为‘00’，在使能通道后（CHEN置1），该位会被硬件强制清零
14:13	MWIDTH[1:0]	存储器传输宽度 软件置1与清0。 00：8位 01：16位 10：32位 11：保留 CHEN为1时不可写入。 如果寄存器DMA_CHxFCTL的MDMEN位为‘0’，在使能通道后（CHEN置1），该位域会被硬件强制与PWIDTH相等
12:11	PWIDTH[1:0]	外设传输宽度 软件置1与清0。 00：8位 01：16位 10：32位 11：保留 CHEN为1时不可写入。
10	MNAGA	存储器地址生成算法 软件置1与清0。 0：固定地址模式 1：增量地址模式 CHEN为1时不可写入。
9	PNAGA	外设地址生成算法 软件置1与清0。 0：固定地址模式 1：增量地址模式 CHEN为1时不可写入。
8	CMEN	循环模式 软件置1与清0。 0：关闭循环模式 1：打开循环模式

		CHEN为1时不可写入 如果TFCS为‘1’，在使能通道后（CHEN置1），该位被自动清0 如果SBMEN为‘1’，在使能通道后（CHEN置1），该位被自动置1
7:6	TM[1:0]	传输方式 软件置1与清0。 00：读外设写存储器 01：读存储器写外设 10：读存储器写存储器 11：保留 CHEN为1时不可写入。
5	TFCS	传输控制器选择 软件置1与清0。 0：DMA作为传输控制器 1：外设作为传输控制器 CHEN为1时不可写入。
4	FTFIE	传输完成中断使能位 软件置1与清0。 0：传输完成中断禁止 1：传输完成中断使能
3	HTFIE	半传输完成中断使能位 软件置1与清0。 0：半传输完成中断禁止 1：半传输完成中断使能
2	TAEIE	传输错误中断使能位 软件置1与清0。 0：传输错误中断禁止 1：传输错误中断使能
1	SDEIE	单数据传输模式异常中断使能位 软件置1与清0。 0：单数据传输模式异常中断禁止 1：单数据传输模式异常中断使能
0	CHEN	通道使能 软件置1，硬件清0。 0：通道禁止 1：通道使能 该位置1，DMA传输开始。发生以下情况该位会被自动清0： – 数据传输完成 – 发生FIFO配置错误或者传输错误

软件清0操作后，读该位仍为1代表还有正在进行的数据传输，软件查询该位可以确定DMA通道是否空闲，可以进行新的数据传输。

10.6.6. 通道 x 计数寄存器（DMA_CHxCNT）（x = 0..7）

地址偏移：0x14 + 0x18 * 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
Reserved															

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	必须保留复位值。
15:0	CNT[15:0]	传输计数 在使能通道后（CHEN置1），该位域不可写。 传输过程中，CNT代表剩余未发的数据量。外设每传输一次数据，CNT减1。如果寄存器DMA_CHxCTL的CMEN位或SBMEN位置1，在每次传输完成时，CNT会由硬件自动重新装载。

10.6.7. 通道 x 外设基地址寄存器（DMA_CHxPADDR）（x = 0..7）

地址偏移：0x18 + 0x18 * 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 在使能通道后（CHEN置1），该位域不可写。 当PWIDTH位‘01’，最低位被忽略，自动半字对齐 当PWIDTH位‘10’，最低位两位被忽略，自动字对齐 注意：若寄存器DMA_CHxCTL的PAIF位置1，该位域必须配置为4字节对齐。

10.6.8. 通道 x 存储器 0 基地址寄存器（DMA_CHxM0ADDR）（x = 0..7）

地址偏移：0x1C + 0x18 * 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
M0ADDR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0ADDR[15:0]															
rw															

位/位域	名称	描述
31:0	M0ADDR[31:0]	存储器0基地址 若寄存器DMA_CHxCTL位MBS为0，该位域定义DMA传输过程中存储器的基地址 如果寄存器DMA_CHxCTL的CHEN位置1且MBS位为0时，该位域不可写 当MWIDTH位‘01’，最低位被忽略，自动半字对齐 当MWIDTH位‘10’，最低位两位被忽略，自动字对齐

10.6.9. 通道 x 存储器 1 基地址寄存器（DMA_CHxM1ADDR）（x = 0..7）

地址偏移：0x20 + 0x18 * 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
M1ADDR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1ADDR[15:0]															
rw															

位/位域	名称	描述
31:0	M1ADDR[31:0]	存储器1基地址 若寄存器DMA_CHxCTL位MBS为1，该位域定义DMA传输过程中存储器的基地址 如果寄存器DMA_CHxCTL的CHEN位置1且MBS为1时，该位域不可写 当MWIDTH位‘01’，最低位被忽略，自动半字对齐 当MWIDTH位‘10’，最低位两位被忽略，自动字对齐

10.6.10. 通道 x FIFO 控制寄存器（DMA_CHxFCTL）（x = 0..7）

地址偏移：0x24 + 0x18 * x
复位值：0x0000 0021

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								FEEIE	保留	FCNT[2:0]			MDMEN	FCCV[1:0]	
								rw		r			rw	rw	

位/位域	名称	描述
31:8	保留	必须保留复位值。
7	FEEIE	FIFO错误和异常中断使能位 软件置1与清0。 0：FIFO错误和异常中断禁止 1：FIFO错误和异常中断使能
6	保留	必须保留复位值。
5:3	FCNT[2:0]	FIFO计数器 硬件置1和清0。 000：FIFO非空并且数据少于1个字 001：FIFO数据等于或多于1个字少于2个字 010：FIFO数据等于或多于2个字少于3个字 011：FIFO数据等于或多余3个字少于4个字 100：FIFO空 101：FIFO满 110~111：保留 该位域表明在数据传输过程FIFO中的数据量。若MDMEN为0，则该位域无意义。
2	MDMEN	多数据传输模式使能 软件置1与清0。 0：关闭多数据传输模式 1：打开多数据传输模式 CHEN为1时不可写入 如果寄存器DMA_CHxCTL的TM位域为‘10’，在通道使能后，该位由硬件强制置1
1:0	FCCV[1:0]	FIFO计数器临界值 软件置1与清0。 00：1个字 01：2个字 10：3个字 11：4个字 CHEN为1时不可写入。 若MDMEN为‘0’，该位域无实际意义。

11. 调试（DBG）

11.1. 简介

GD32VW55x 系列产品提供了调试功能。这些功能通过 RISC-V 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 RISC-V 内核中。调试系统支持 JTAG 调试。调试功能请参考下列文档：

- RISC-V外部调试支持版本0.13。

调试系统帮助调试者在低功耗模式下调试或者一些外设调试。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、WWDGT、FWDGT、I2C 和 RTC。

11.2. JTAG 功能描述

调试工具可以通过 JTAG 调试接口来访问调试功能。

11.2.1. 引脚分配

JTAG 调试提供五个引脚的接口：JTAG 时钟引脚（JTCK），JTAG 模式选择引脚（JTMS），JTAG 数据输入引脚（JTDI），JTAG 数据输出引脚（JTDO），JTAG 复位引脚（NJTRST，低电平有效）。

调试引脚分配：

PA15: JTDI

PA14: JTCK

PA13: JTMS

PB4: NJTRST

PB3: JTDO

默认复位后使用五个引脚的 JTAG 调试，用户可以在不使用 NJTRST 引脚情况下正常使用 JTAG 功能，此时 PB4 可以用作普通 GPIO 功能（NJTRST 硬件拉高）。使用 cJTAG 调试时的引脚分配为：PA14: JTCK、PA13: JTMS，其他引脚（PA15、PB4、PB3）可用作普通 GPIO 功能。如果 JTAG 调试功能都没有使用，这五个引脚都释放作为普通 GPIO 功能。五个引脚具体配置请参考*通用和备用输入/输出接口（GPIO 和 AFIO）*。

11.2.2. JTAG 链状结构

RISC-V 内核的 JTAG TAP 和边界扫描（BSD）TAP 串行连接。边界扫描（BSD）JTAG 的 IR（指令寄存器）是 5 位，RISC-V 内核的 JTAG 的 IR（指令寄存器）也是 5 位。

BSD JTAG ID 代码是 0x790007A3。

11.2.3. 调试复位

系统复位初始化了 RISC-V 的绝大部分组件。NJTRST 仅能复位 JTAG TAP 控制器。

11.3. 调试保持功能描述

11.3.1. 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。

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

当内核停止，并且 DBG 控制寄存器 1/2（DBG_CTL1/2）中的相应位置 1。对于不同外设，有不同动作：

对于 TIMER 外设，TIMER 计数器在调试暂停状态下停止计数；

对于 I2C 外设，SMBUS 超时在调试暂停状态下保持停止；

对于 WWDGT 或者 FWDGT 外设，计数器在调试暂停状态下停止计数；

对于 RTC 外设，计数器在调试暂停状态下停止计数。

11.4. DBG 寄存器

DEBUG 基地址：0xE0044000

11.4.1. ID 寄存器（DBG_ID）

地址偏移：0x00
复位值：0x0000 0000，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

11.4.2. 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															

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

位/位域	名称	描述
31:3	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: 在深度睡眠模式下，系统时钟和 AHB 时钟由 CK_IRC16M 提供

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

11.4.3. 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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved.									I2C1_HOLD	I2C0_HOLD	Reserved.				
									rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved.			FWDGT_HOLD	WWDGT_HOLD	RTC_HOLD	Reserved					TIMER5_HOLD	Reserved		TIMER2_HOLD	TIMER1_HOLD
			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	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
21	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
20:13	保留	必须保持复位值
12	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
11	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
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 定时器计数器不变，用于调试
9:5	保留	必须保持复位值
4	TIMER5_HOLD	TIMER5 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 5 计数器不变，用于调试
3:2	保留	必须保持复位值
1	TIMER2_HOLD	TIMER2 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 2 计数器不变，用于调试
0	TIMER1_HOLD	TIMER1 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 1 计数器不变，用于调试

11.4.4. 控制寄存器 2（DBG_CTL2）

地址偏移：0x0C

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							TIMER16_HOLD	TIMER15_HOLD	保留						
							rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.															TIMER0_HOLD
															rw

位/位域	名称	描述
31:25	保留	必须保持复位值
24	TIMER16_HOLD	TIMER16 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 16 计数器不变，用于调试
23	TIMER15_HOLD	TIMER15 保持位

		该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 15 计数器不变，用于调试
22:1	保留	必须保持复位值
0	TIMER0_HOLD	TIMER0 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 0 计数器不变，用于调试

12. 模数转换器（ADC）

12.1. 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 9 个外部通道和 2 个内部通道上的模拟信号。这 11 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。

12.2. 主要特征

- 高性能：
  - 可配置 12 位、10 位、8 位、或者 6 位分辨率
  - ADC 采样率：12 位分辨率是 3MSPs，10 位分辨率是 3.5MSPs，8 位分辨率是 4.2MSPs，6 位分辨率是 5.25MSPs，分辨率越低，转换越快。
  - 可编程采样时间
  - 数据存储模式：最高有效位对齐（MSB）和最低有效位对齐（LSB）
  - 支持 DMA 请求
- 模拟输入通道：
  - 9 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}）
- 转换开始的发起：
  - 软件
  - 硬件触发
- 运行模式：
  - 转换单个通道，或者扫描一序列通道
  - 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
  - 溢出事件
- 模拟看门狗；
- 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：$0 \le V_{IN} \le V_{DDA}$。

12.3. 引脚和内部信号

*图 12-1. ADC 模块框图*给出了 ADC 模块框图。*表 12-1. ADC 内部输入信号*和*表 12-2. ADC 输入引脚定义*给出了 ADC 内部信号和引脚定义。

表 12-1. ADC 内部输入信号

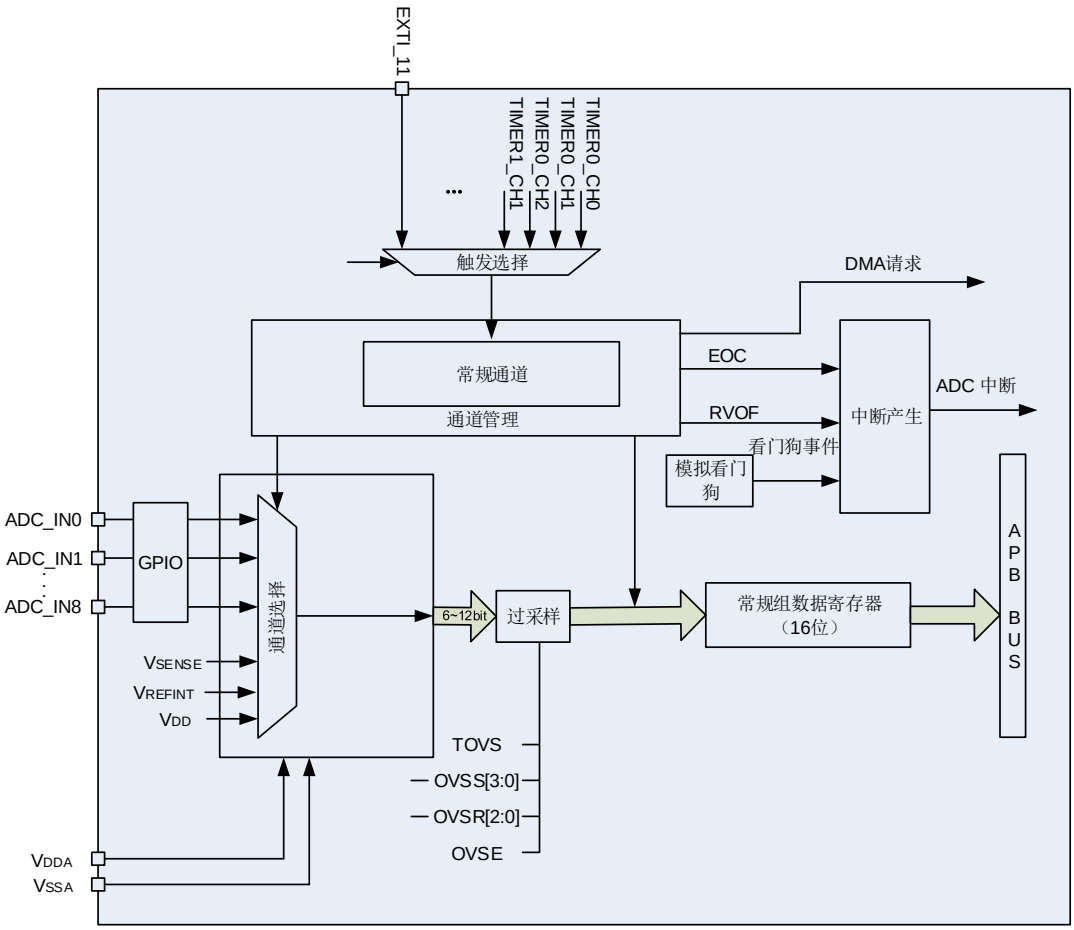
内部信号名称	说明
V_{SENSE}	内部温度传感器电压输出
V_{REFINT}	内部参考电压输出

表 12-2. ADC 输入引脚定义

名称	注释
V_{DDA}	模拟电源输入等于V_{DD}
V_{SSA}	模拟地，等于 V_{SS}
ADCx_IN[8:0]	多达 9 路外部通道

12.4. 功能描述

图 12-1. ADC 模块框图



12.4.1. ADC 时钟

CK_ADC 时钟是由时钟控制器提供的，它和 AHB、APB2 时钟保持同步。ADC 最大的时钟频率为 42MHz。ADC 时钟可以在 RCU 时钟控制器中进行分配和配置。

12.4.2. 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。

12.4.3. 常规序列

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 11 个通道，每个通道称为常规通道。

ADC_RSQ0 寄存器中的 RL[3:0] 位规定了整个常规序列转换序列的长度。

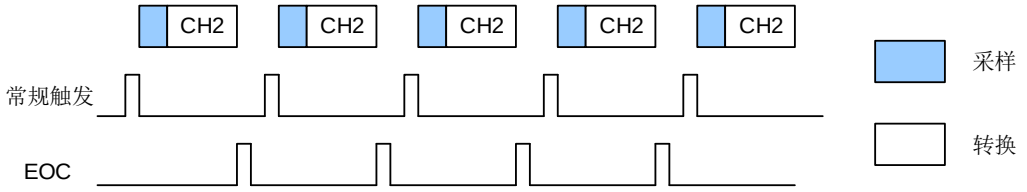
ADC_RSQ0~ADC_RSQ2寄存器规定了常规序列的通道选择。

12.4.4. 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 12-2. 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。

常规序列单次运行模式的软件流程：

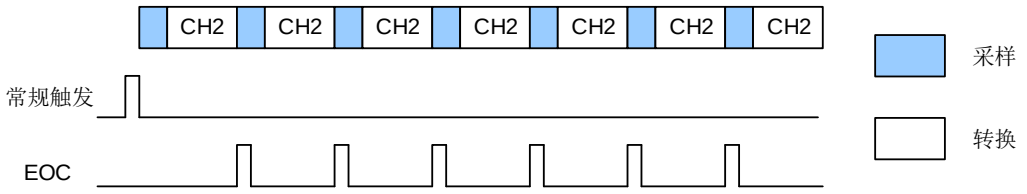
1. 确保ADC_CTL0寄存器的DISRC位和SM位以及ADC_CTL1寄存器中的CTN位为0；
2. 用模拟通道编号来配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，可以配置ADC_CTL1寄存器中的ETMRC位和ETSRC位；
5. 设置SWRCST位，或者为常规序列产生一个外部触发信号；
6. 等到EOC位置1；
7. 延迟一个CK_ADC后，从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位。

注意：当EOC置1后，需延迟一个CK_ADC再读取ADC转换结果。

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 12-3. 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ETMRC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. 延迟一个CK_ADC后，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。

注意：当EOC置1后，需延迟一个CK_ADC再读取ADC转换结果。

可以使用 DMA 来传输转换数据，不需循环查询 EOC 标志位：

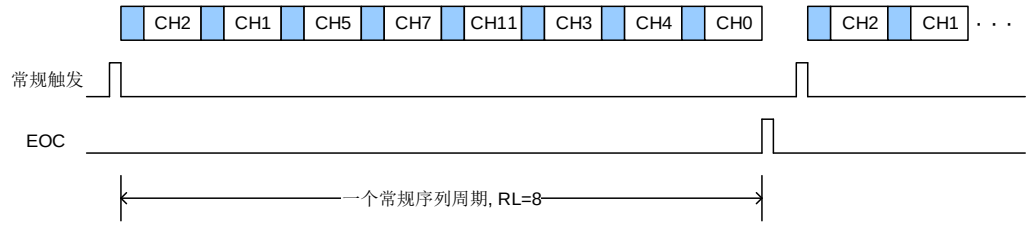
1. 设置ADC_CTL1寄存器的CTN位为1；
2. 根据模拟通道编号配置RSQ0；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETMRC位和ETSRC位；
5. 准备DMA模块，用于传输来自ADC_RDATA寄存器的数据；
6. 设置SWRCST位，或者给常规序列产生一个外部触发。

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。转换数据存储在 ADC_RDATA 或 ADC_IDATAx 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。当常规序列通道工作在扫描运行模式下时，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 12-4. 扫描运行模式，且连续运行模式禁能

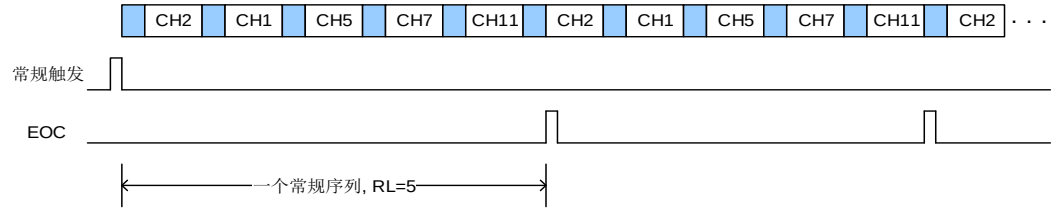


常规序列扫描运行模式的软件流程：

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ETMRC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。

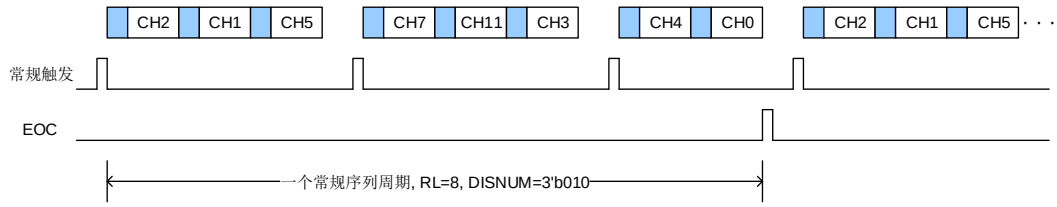
图 12-5. 扫描运行模式，连续运行模式使能



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 不超过 8），该序列是 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 寄存器所配置通道中接下来的 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 12-6. 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ETMRC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。

12.4.5. 转换结果阈值监测功能

ADC_CTL0 寄存器中的 RWDEN 位置 1 时，将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。

12.4.6. 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。

在左对齐中，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 12-7. 12 位数据存储模式*，*图 12-8. 10 位数据存储模式*，*图 12-9. 8 位数据存储模式*和*图 12-10. 6 位数据存储模式*所示。

图 12-7. 12 位数据存储模式

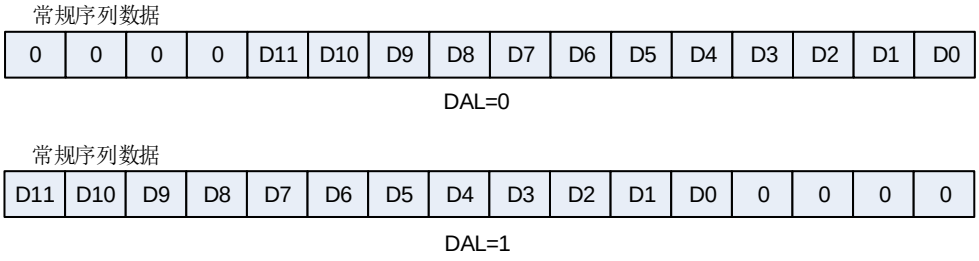


图 12-8. 10 位数据存储模式

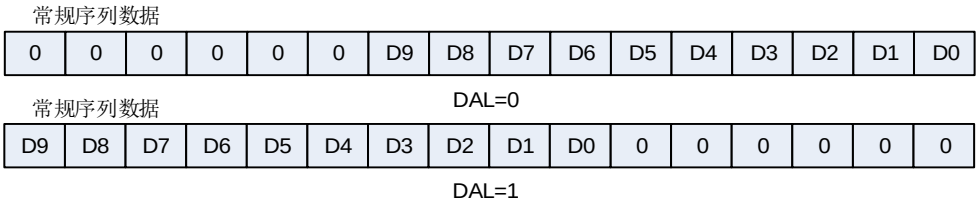


图 12-9. 8 位数据存储模式

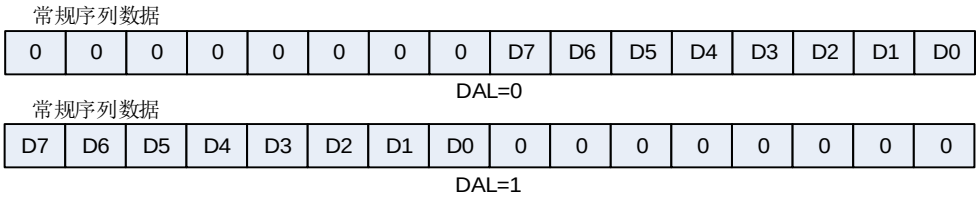
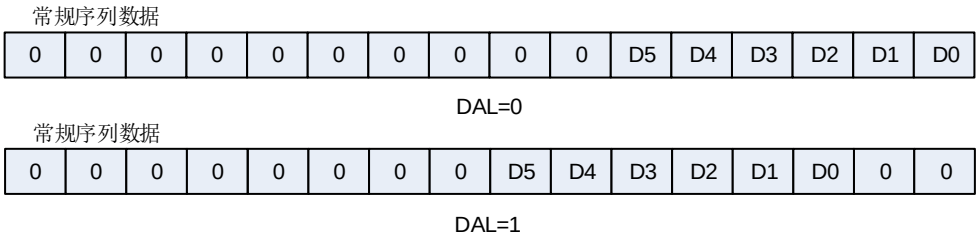


图 12-10. 6 位数据存储模式



12.4.7. 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]位设置配置。每个通道可以使用不同的采样时间。例如，在 12 位分辨率的情况下，总转换时间=采样时间+12 个 CK_ADC 周期。

例如：CK_ADC = 42MHz，采样时间为 2.5 个周期，那么总的转换时间为："2.5+12"个 CK_ADC 周期，即 0.345us。

12.4.8. 外部触发配置

外部触发输入的上升沿、下降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[3:0]位控制。

表 12-3. 外部触发模式

ETMRC[1:0]	触发模式
00	外部触发禁能
01	外部触发信号上升沿触发使能
10	外部触发信号下降沿触发使能
11	外部触发信号双边沿触发使能

表 12-4. ADC 外部触发源

ETSRC[3:0]	触发源	触发类型
0000	TIMER0_CH0	内部信号
0001	TIMER0_CH1	
0010	TIMER0_CH2	
0011	TIMER1_CH1	
0100	TIMER1_CH2	
0101	TIMER1_CH3	
0110	TIMER1_TRGO	
0111	TIMER2_CH0	
1000	TIMER2_TRGO	
1001	TIMER2_CH2	
1010	TIMER15_CH0	
1011	TIMER16_CH0	
1100	保留	
1101	保留	
1110	TIMER5_TRGO	
1111	EXTI_11	外部信号

可以实时修改外部触发的选择，且不会因为修改操作发生触发事件。

12.4.9. DMA 请求

DMA 请求可以用来传输多个通道的转换结果。常规序列的 DMA 请求可以通过设置 ADC_CTL1 寄存器的 DMA 位来使能。当 DMA 位置 1 时，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。

12.4.10. 溢出检测

当 DMA 使能或 ADC_CTL1 寄存器的 EOCM 位置 1 时，可以使能溢出检测。如果一个常规通道转换在上一个常规通道转换数据读出之前已经完成，则会产生一个溢出事件，相应的 ADC_STAT 状态寄存器的 ROVF 标志位会置位。如果 ADC_CTL0 寄存器的 ROVFIE 置位，溢出中断产生。

为了使得 ADC 从 ROVF 溢出状态中恢复过来，建议对 DMA 模块重新进行初始化。内部状态机复位，以保证常规转换数据正确的传输。ADC 转换将会停止，直到 ROVF 位被清零。

ADC 从 ROVF 状态恢复的软件流程如下：

1. 将ADC_CTL1寄存器的DMA位清0；
2. 将ADC_CTL1寄存器的ADCON位清0；
3. 将DMA_CHxCTL寄存器的CHEN位清0，用于重新初始化DMA模块；
4. 将ADC_STAT寄存器的ROVF位清0；
5. 将DMA_CHxCTL寄存器的CHEN位置1；
6. 将ADC_CTL1寄存器的DMA位置1；
7. 将ADC_CTL1的ADCON位置1；
8. 等待T(setup)；
9. 通过软件或触发开始ADC转换。

12.4.11. ADC 内部通道

将 ADC_CCTL 寄存器的 TSVREN 位置 1，可以使能温度传感器通道（ADC_IN9）和 V_{REFINT} 通道（ADC_IN10）。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 t_{s_temp}（具体数值请参考 datasheet 文档）。温度传感器不用时，复位 TSVREN 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，温度变化曲线的偏差在芯片间会有不同(最多相差 45°C)。内部温度传感器更适用于检测温度的变化，而不是用于测量绝对温度。如果需要测量精确的温度，应该使用一个外置的温度传感器来校准这个偏移错误。

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_IN10 输入通道。

使用温度传感器：

1. 配置温度传感器通道（ADC_IN9）的转换序列和采样时间为ts_temp；
2. 置位ADC_CCTL寄存器中的TSVREN位，使能温度传感器；
3. 置位ADC_CTL1寄存器的ADCON位，或者由外部触发触发ADC转换；
4. 从ADC数据寄存器中读取并计算温度传感器数据Vtemperature，并由下面公式计算出实际温度：
5. 温度(°C) = {(V25 – Vtemperature) / Avg_Slope} + 25

V25：温度传感器在 25°C 下的电压，典型值请参考 datasheet 文档。

Avg_Slope：温度与温度传感器电压曲线的均值斜率，典型值请参考 datasheet 文档。

12.4.12. 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。如*表 12-5. 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 12-5. 不同分辨率对应的 tCONV 时间

DRES [1:0] bits	t_{CONV}（ADC 时钟周期）	t_{CONV}（ns）（fADC=42MHz）	t_{SMPL}（ADC 时钟周期）	t_{ADC}（ADC 时钟周期）	t_{ADC}（ns）（f_{ADC}=42MHz）
12	12	310 ns	2	14	333 ns
10	10	262 ns	2	12	286 ns
8	8	214 ns	2	10	238 ns
6	6	167 ns	2	8	190 ns

12.4.13. 片上硬件过采样

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。

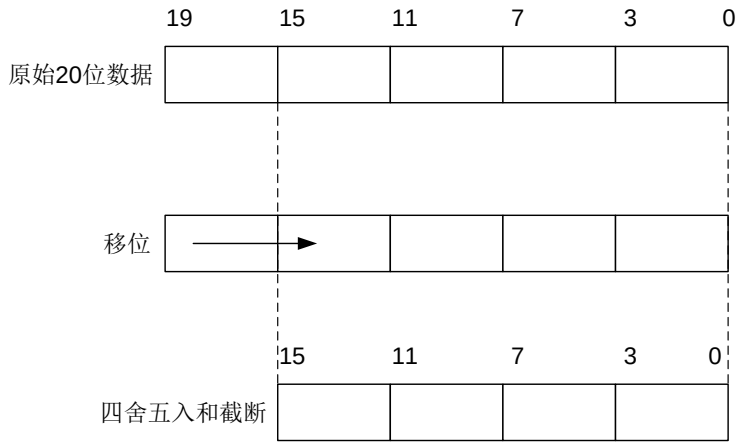
其结果根据如下公式计算得出，其中，N 和 M 的值可以被调整 $D_{out}(n)$是指 ADC 输出的第 n 个数字信号：

$$Result=\frac{1}{M}*\sum_{n=0}^{n=N-1} D_{OUT}(n) \qquad (14\text{-}1)$$

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 12-11. 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 12-12. 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 12-12. 右移 5 位和取整的数例

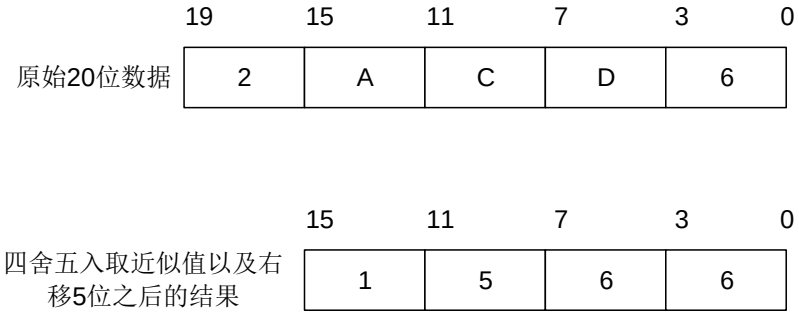


*表 12-6. 不同 N 和 M 组合的最大输出值（灰色值表示截断）*给出了 N 和 M 的各种组合的数据格式，初始转换值为 0xFFF。

表 12-6. 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：

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (14\text{-}2)$$

12.4.14. 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（模拟看门狗状态位置1）；
- 溢出事件。

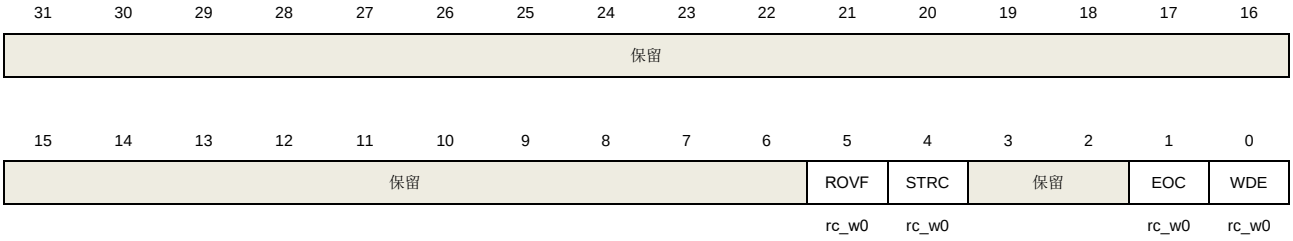
12.5. ADC 寄存器

ADC 基地址：0x4001 2000

12.5.1. 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
31:6	保留	必须保持复位值
5	ROVF	常规序列数据寄存器溢出 0：常规数据寄存器没有溢出 1：常规数据寄存器溢出 在单次或多次模式采样中，当常规数据寄存器溢出时，该位由硬件置位。只有在DMA使能或转换模式结束位置1（EOCM=1）时，这个标志位才会置1。如果出现ROVF置位，则最后的常规数据会被丢失。 软件写0清除。
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：常规序列转换没有结束 1：常规序列转换结束 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。

12.5.2. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留					ROVFIE	DRES		RWDEN	保留						
					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	EOICIE	WDEIE	EOCIE	WDCHSEL [4:0]				
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	ROVFIE	常规序列数据寄存器溢出（ROVF）中断使能 0：ROVF中断禁能 1：ROVF中断使能
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：模拟看门狗禁能 1：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：间断运行模式禁能 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，模拟看门狗在通道的配置 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：扫描运行模式禁能

		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：中断禁能 1：中断使能
5	EOCIE	EOC中断使能 0：中断禁能 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 00011：ADC通道3 00100：ADC通道4 00101：ADC通道5 00110：ADC通道6 00111：ADC通道7 01000：ADC通道8 01001：ADC通道9 01010：ADC通道10 其他值保留。 注意：ADC的模拟输入通道和通道10分别内部连接到温度传感器、V_{REFINT}。

12.5.3. 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	SWRCST	ETMRC[1:0]		ETSRC[3:0]				保留							
	rw	rw		rw											

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

位/位域	名称	描述
31	保留	必须保持复位值
30	SWRCST	常规序列软件启动转换 该位置1启动常规序列转换，该位由软件置位，软件清零或转换开始由硬件清零。

29:28	ETMRC[1:0]	常规序列外部触发模式 00：常规序列外部触发禁能 01：常规序列外部触发上升沿使能 01：常规序列外部触发下降沿使能 11：常规序列外部触发双边沿使能
27:24	ETSRC[3:0]	常规序列外部触发选择 0000：TIMER0 CH0 0001：TIMER0 CH1 0010：TIMER0 CH2 0011：TIMER1 CH1 0100：TIMER1 CH2 0101：TIMER1 CH3 0110：TIMER1 TRGO 0111：TIMER2 CH0 1000：TIMER2 TRGO 1001：TIMER2 CH2 1010：TIMER15 CH0 1011：TIMER16 CH0 1100：保留 1101：保留 1110：TIMER5 TRGO 1111：EXTI外部中断线11
23:12	保留	必须保持复位值
11	DAL	数据对齐 0：最低有效位（LSB）对齐 1：最高有效位（MSB）对齐
10	EOCM	转换模式结束 0：只有在常规转换序列转换结束时，才将EOC置1。如果不设置DMA=1，则溢出检测禁能。 1：在每个常规序列转换结束时，将EOC置1。溢出检测自动使能。
9	DDM	DMA禁能模式 该位用于在单次ADC模式下配置DMA禁能。 0：DMA机制在DMA控制器的传输结束信号之后禁能。 1：DMA=1时，在每个常规序列转换结束时，DMA机制产生一个DMA请求。
8	DMA	常规序列DMA请求使能 0：DMA请求禁能 1：DMA请求使能
7:2	保留	必须保持复位值
1	CTN	连续模式

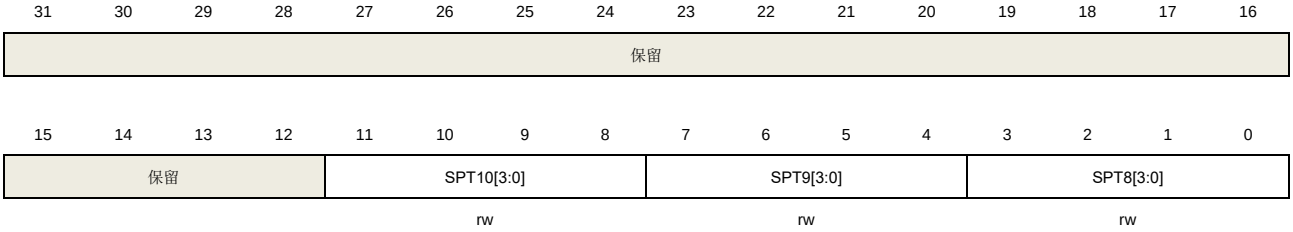
0：连续运行模式禁能
1：连续运行模式使能

0	ADCON	开启ADC。该位从0变成1将在稳定时间结束后唤醒ADC。 0：禁能ADC并掉电 1：使能ADC

12.5.4. 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。

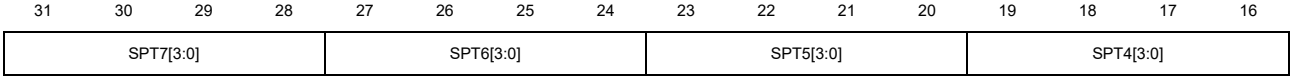


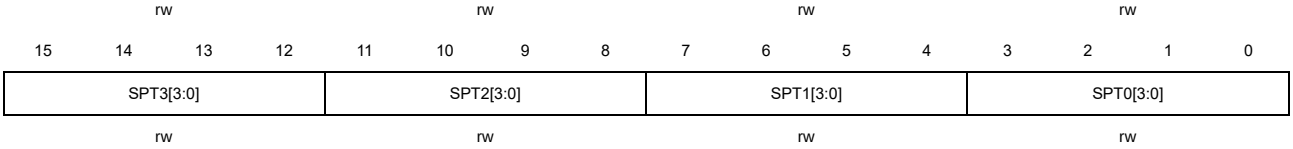
位/位域	名称	描述
31:12	保留	必须保持复位值
11:8	SPT10[2:0]	参考SPT8[2:0]的描述
7:4	SPT9[2:0]	参考SPT8[2:0]的描述
3:0	SPT8[2:0]	通道采样时间 0000：通道采样时间为1.5周期 0001：通道采样时间为2.5周期 0010：通道采样时间为14.5周期 0011：通道采样时间为27.5周期 0100：通道采样时间为55.5周期 0101：通道采样时间为83.5周期 0110：通道采样时间为111.5周期 0111：通道采样时间为143.5周期 1000：通道采样时间为479.5周期

12.5.5. 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。





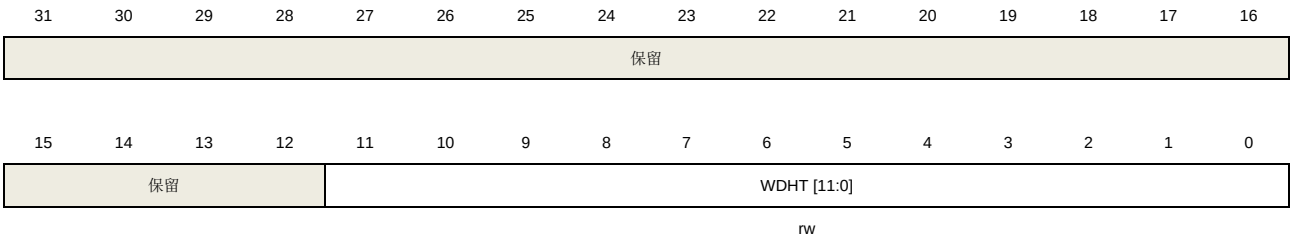
位/位域	名称	描述
31:28	SPT7[3:0]	参考SPT0[3:0]的描述
27:24	SPT6[3:0]	参考SPT0[3:0]的描述
23:20	SPT5[3:0]	参考SPT0[3:0]的描述
19:16	SPT4[3:0]	参考SPT0[3:0]的描述
15:12	SPT3[3:0]	参考SPT0[3:0]的描述
11:8	SPT2[3:0]	参考SPT0[3:0]的描述
7:4	SPT1[3:0]	参考SPT0[3:0]的描述
3:0	SPT0[3:0]	通道采样时间 0000：通道采样时间为1.5周期 0001：通道采样时间为2.5周期 0010：通道采样时间为14.5周期 0011：通道采样时间为27.5周期 0100：通道采样时间为55.5周期 0101：通道采样时间为83.5周期 0110：通道采样时间为111.5周期 0111：通道采样时间为143.5周期 1000：通道采样时间为479.5周期

12.5.6. 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。



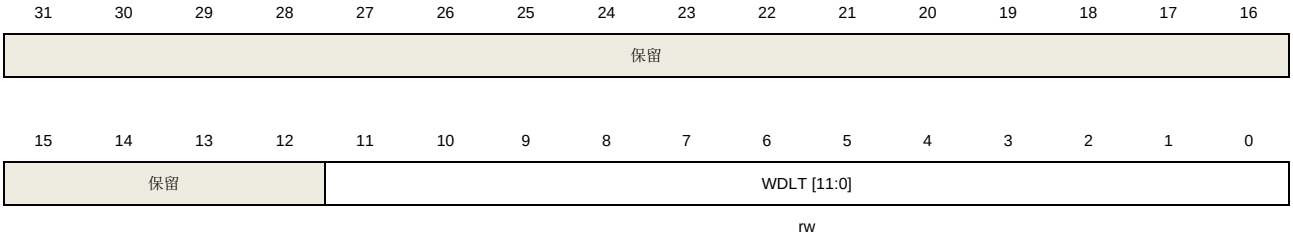
位/位域	名称	描述
31:12	保留	必须保持复位值

11:0	WDHT [11:0]	模拟看门狗高阈值 这些位定义了模拟看门狗的高阈值。

12.5.7. 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。

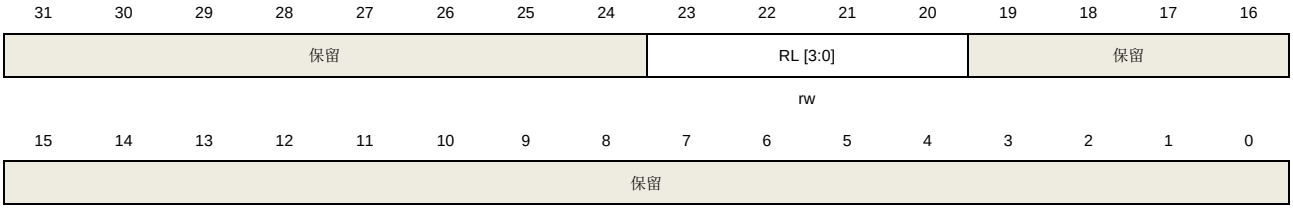


位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	WDLT [11:0]	模拟看门狗低阈值 这些位定义了模拟看门狗的低阈值。

12.5.8. 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。

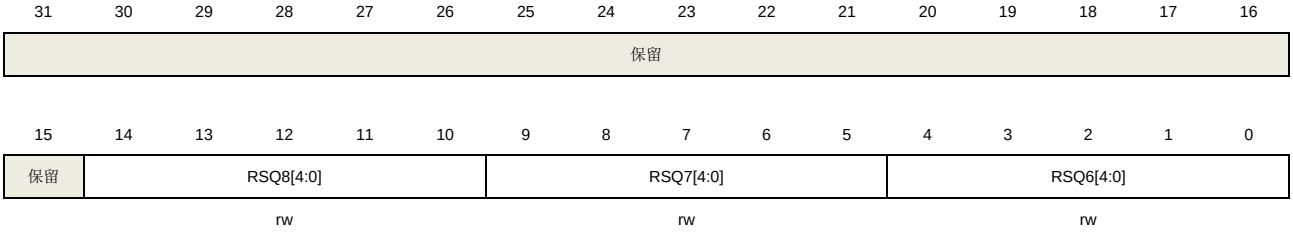


位/位域	名称	描述
31:24	保留	必须保持复位值
23:20	RL [3:0]	常规通道长度 常规通道转换序列中的总通道数目为RL[3:0]+1。
19:0	保留	必须保持复位值

12.5.9. 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。

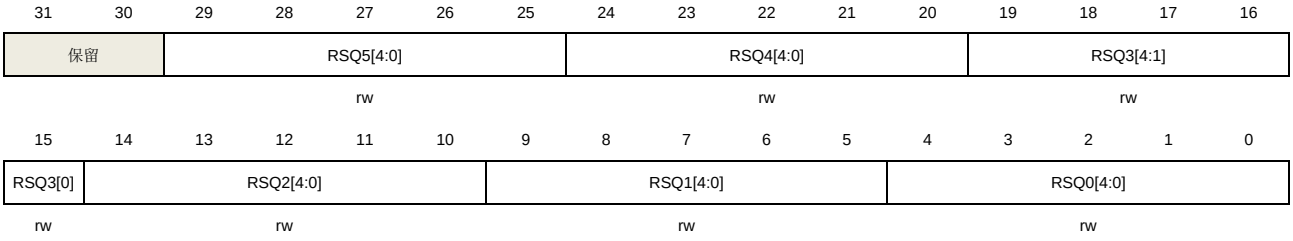


位/位域	名称	描述
31:15	保留	必须保持复位值
14:10	RSQ8[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ7[4:0]	参考RSQ0[4:0]的描述
4:0	RSQ6[4:0]	参考RSQ0[4:0]的描述

12.5.10. 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。



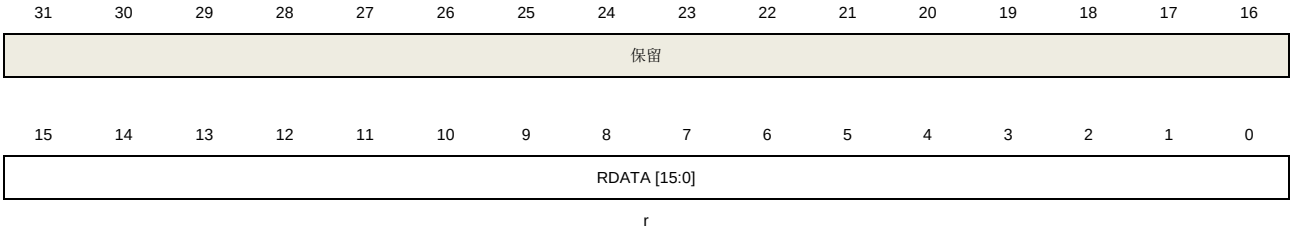
位/位域	名称	描述
31:30	保留	必须保持复位值
29:25	RSQ5[4:0]	参考RSQ0[4:0]的描述
24:20	RSQ4[4:0]	参考RSQ0[4:0]的描述
19:15	RSQ3[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ2[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ1[4:0]	参考RSQ0[4:0]的描述

4:0	RSQ0[4:0]	通道编号（0..11）写入这些位来选择常规通道的第n个转换的通道

12.5.11. 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。

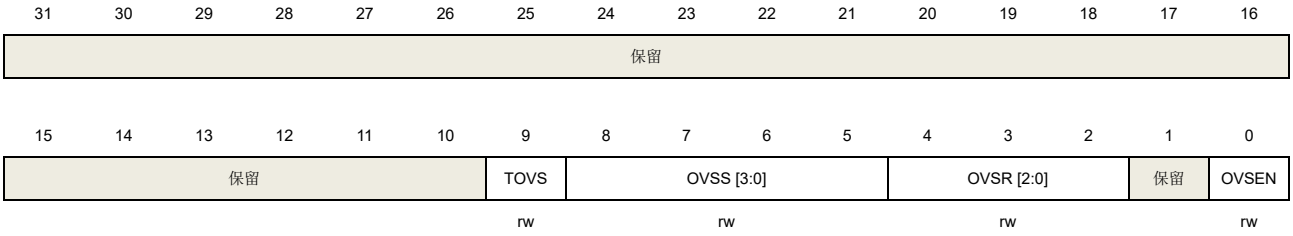


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	RDATA [15:0]	常规通道转换数据 这些位包含了常规通道的转换结果，只读。

12.5.12. 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，对一个通道连续进行过采样转换。 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

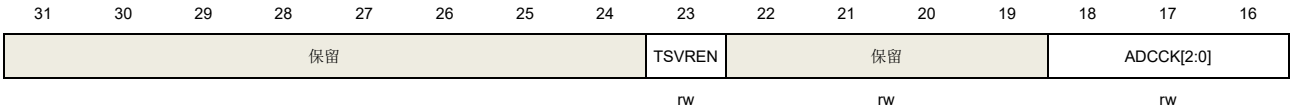
		该位通过软件设置和清除。 0000：不移位 0001：移1位 0010：移2位 0011：移3位 0100：移4位 0101：移5位 0110：移6位 0111：移7位 1000：移8位 其它值保留。 注意：只有在ADCON=0的时候，才允许通过软件对该位进行写操作(确保没有转换正在进行)。
4:2	OVSR [2:0]	过采样率 这些位定义了过采样率的大小。 000：2x 001：4x 010：8x 011：16x 100：32x 101：64x 110：128x 111：256x 注意：只有在ADCON=0的时候，才允许通过软件对该位进行写操作(确保没有转换正在执行)。
1	保留	必须保持复位值
0	OVSEN	过采样使能 该位通过软件置位和清除。 0：过采样禁能 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)。

12.5.13. 通用控制寄存器（ADC_CCTL）

地址偏移：0x304

复位值：0x0000 0000

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															

位/位域	名称	描述
31:24	保留	必须保持复位值
23	TSVREN	ADC通道9（温度传感器）和通道10（内部参考电压）使能 0：通道9和通道10禁能 1：通道9和通道10使能
22:19	保留	必须保持复位值
18:16	ADCCK[2:0]	ADC时钟 这些位用于配置ADC时钟 000：PCLK2 / 2 001：PCLK2 / 4 010：PCLK2 / 6 011：PCLK2 / 8 100：HCLK / 5 101：HCLK / 6 110：HCLK / 10 111：HCLK / 20
15:0	保留	必须保持复位值

13. 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的时候，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。

13.1. 独立看门狗定时器（FWDGT）

13.1.1. 简介

独立看门狗定时器（FWDGT）有独立时钟源（IRC32K）。因此就算是主时钟失效了，它仍然能保持工作状态，这非常适合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。

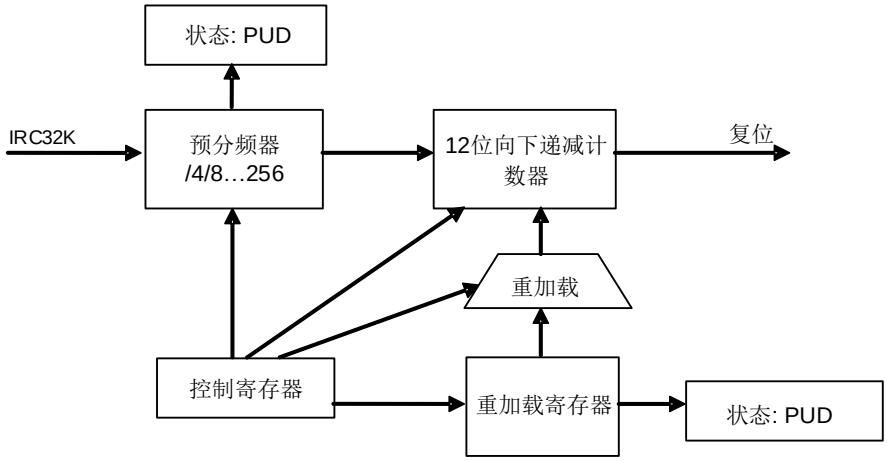
13.1.2. 主要特征

- 自由运行的12位向下计数器；
- 如果看门狗定时器被使能，那么当向下计数器的值达到0时产生系统复位；
- 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，可以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。

13.1.3. 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 13-1. 独立看门狗定时器框图*的独立看门狗定时器的功能模块。

图 13-1. 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，产生一次复位。

在任何时候向控制寄存器（FWDGT_CTL）中写 0xAAAA 都可以重装载计数器，重装载值来源于 FWDGT_RLD 寄存器。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复位。

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。为了避免复位，软件应该在计数器达到 0x000 之前重装载计数器。

FWDGT_PSC 寄存器和 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 到控制寄存器（FWDGT_CTL）中。写其他任何值到控制寄存器中将会再次启动对这些寄存器的写保护。当预分频寄存器（FWDGT_PSC）或者重装载寄存器（FWDGT_RLD）更新时，FWDGT_STAT 寄存器的状态位会被置 1。

如果在 MCU 调试模块中的 FWDGT_HOLD 位被清 0，即使 RISC-V 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 13-1. 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
1/128	101	0.03125	16380.03125
1/256	110 or 111	0.03125	32760.03125

通过 IRC32K 校准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。

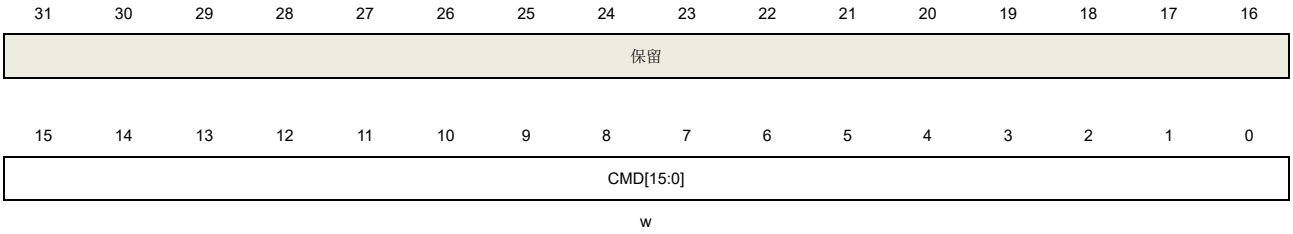
13.1.4. FWDGT 寄存器

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。

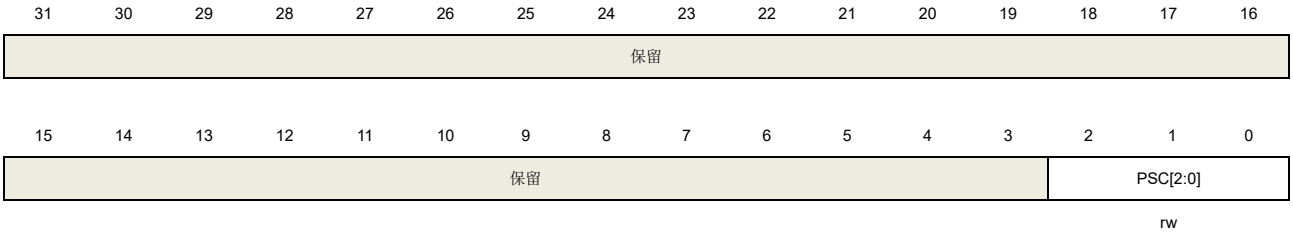


位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	CMD[15:0]	只可写，写入不同的值来产生不同的功能 0x5555：关闭FWDGT_PSC和FWDGT_RLD的写保护 0xCCCC：开启独立看门狗定时器计数器。计数减到0时产生中断 0xAAAA：重装载计数器

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

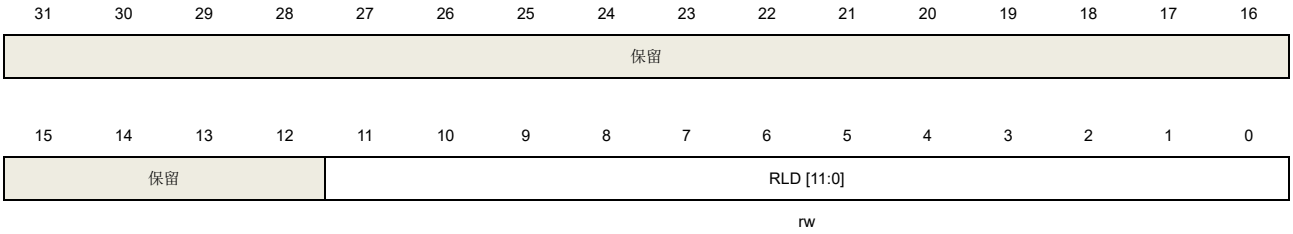
如果应用需要使用几个预分频系数，改变预分频值之前必须等到PUD位被清0。更新了预分频寄存器中的值后，在代码持续执行之前不必等待PUD值被清零。

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。



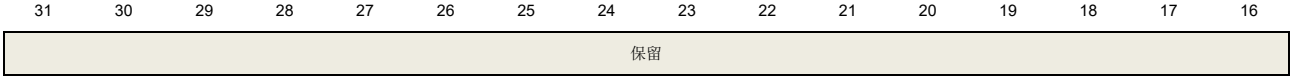
位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	RLD[11:0]	独立看门狗定时器计数器重装载值，向FWDGT_CTL寄存器写入0xAAAA的时候，这个值会被更新到看门狗定时器计数器中。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。在改写这个寄存器的过程中，FWDGT_STAT寄存器的RUD位被置1，从此寄存器中读取的任何值都是无效的。 如果应用需要使用几个重装载值，改变重加载值之前必须等到RUD位被清0。更新了重加载寄存器的值后，在代码持续执行之前不必等待RUD值被清零。

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。



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

位/位域	名称	说明
31:2	保留	必须保持复位值。
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。

13.2. 窗口看门狗定时器（WWDGT）

13.2.1. 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 或者在计数值达到窗口寄存器之前更新计数器，都会产生一个提前唤醒标志，如果使能中断也将会产生中断。

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。

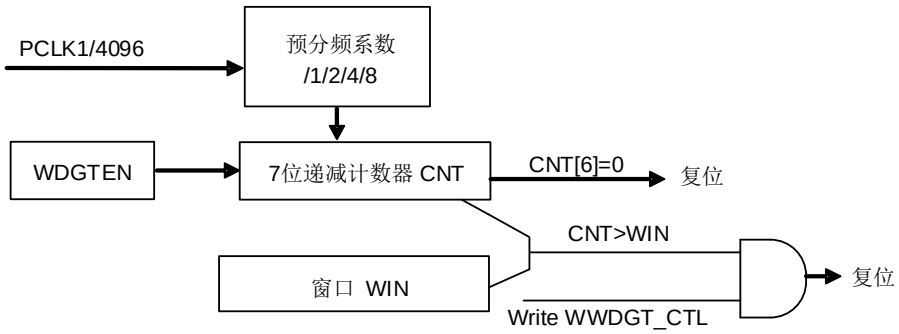
13.2.2. 主要特征

- 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或者在计数值达到窗口寄存器之前更新计数器的时候会产生中断。
- 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。

13.2.3. 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复位。

图 13-2. 窗口看门狗定时器框图



上电复位之后看门狗定时器总是关闭的。软件可以向 WWDGT_CTL 的 WDGTEN 写 1 开启看门狗定时器。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

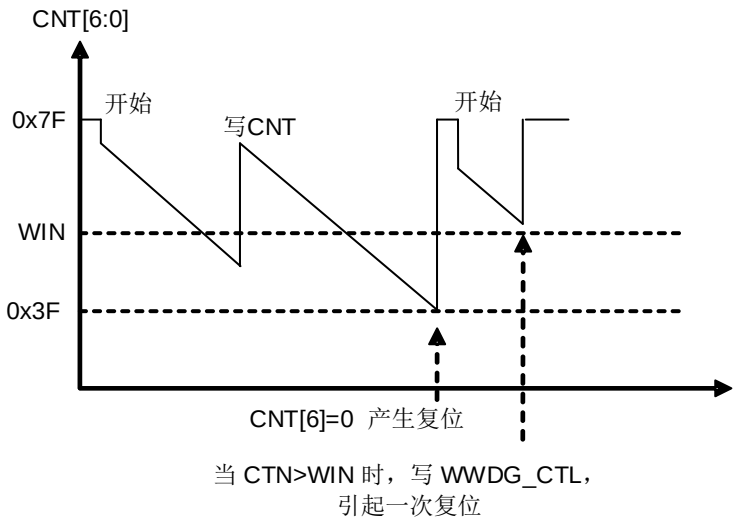
减速度取决于 APB1 时钟和预分频器（WWDGT_CFG 寄存器的 PSC[1:0]位）。

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 或者在计数值达到窗口寄存器之前更新计数器的时候该中断产生。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 13-3. 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

t_{WWDGT} 的最大值和最小值请参考*表 13-2. 在 42MHz（fPCLK1）时的最大/最小超时值*。

表 13-2. 在 42MHz（f_{PCLK1}）时的最大/最小超时值

预分频系数	PSC[1:0]	最小超时 CNT[6:0] =0x40	最大超时 CNT[6:0]=0x7F
1/1	00	97.52 μs	6.24 ms
1/2	01	195.04 μs	12.48 ms
1/4	10	390.08 μs	24.96 ms

1/8	11	780.16 μs	49.92 ms

如果MCU调试模块中的WWDGT_HOLD位被清0，即使RISC-V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。

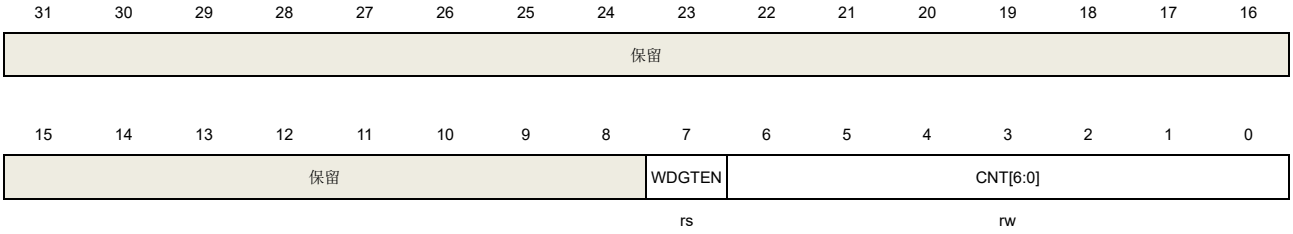
13.2.4. WWDGT 寄存器

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。

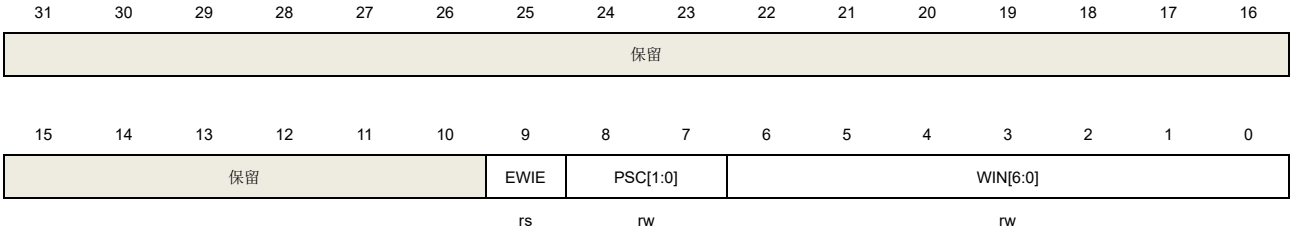


位/位域	名称	说明
31:8	保留	必须保持复位值。
7	WDGTEN	开启窗口看门狗定时器，硬件复位的时候清0，写0无效。 0：关闭窗口看门狗定时器 1：开启窗口看门狗定时器
6:0	CNT[6:0]	看门狗定时器计数器的值。当计数值从0x40降到0x3F时，产生看门狗定时器复位。当计数器值高于窗口值的时候，写计数器可以产生看门狗定时器复位。

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。



位/位域	名称	说明

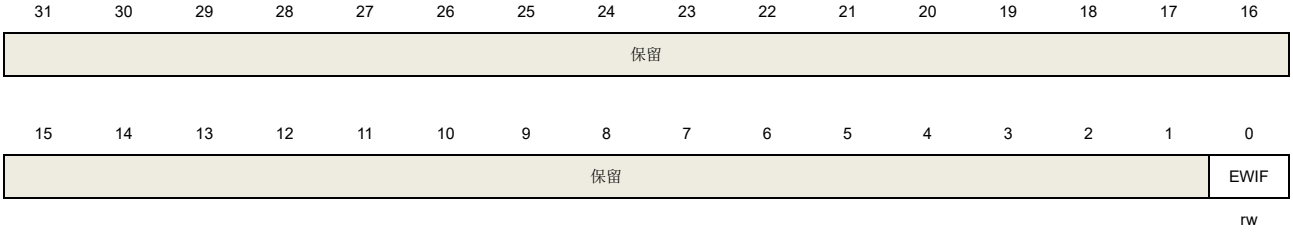
31:10	保留	必须保持复位值。
9	EWIE	提前唤醒中断使能。如果该位被置1，计数值达到0x40时触发中断，或者在计数值达到窗口寄存器之前更新计数器也能触发中断。该位由硬件复位清0，或通过RCU模块的WWDGT软件复位来清0。写0没有任何作用。
8:7	PSC[1:0]	预分频器，看门狗定时器计数器的时间基准 00：PCLK1/4096/1 01：PCLK1/4096/2 10：PCLK1/4096/4 11：PCLK1/4096/8
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器（WWDGT_CTL的CNT位）会产生复位。

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。



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或者在计数值达到窗口寄存器之前更新计数器，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。

14. 实时时钟（RTC）

14.1. 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。

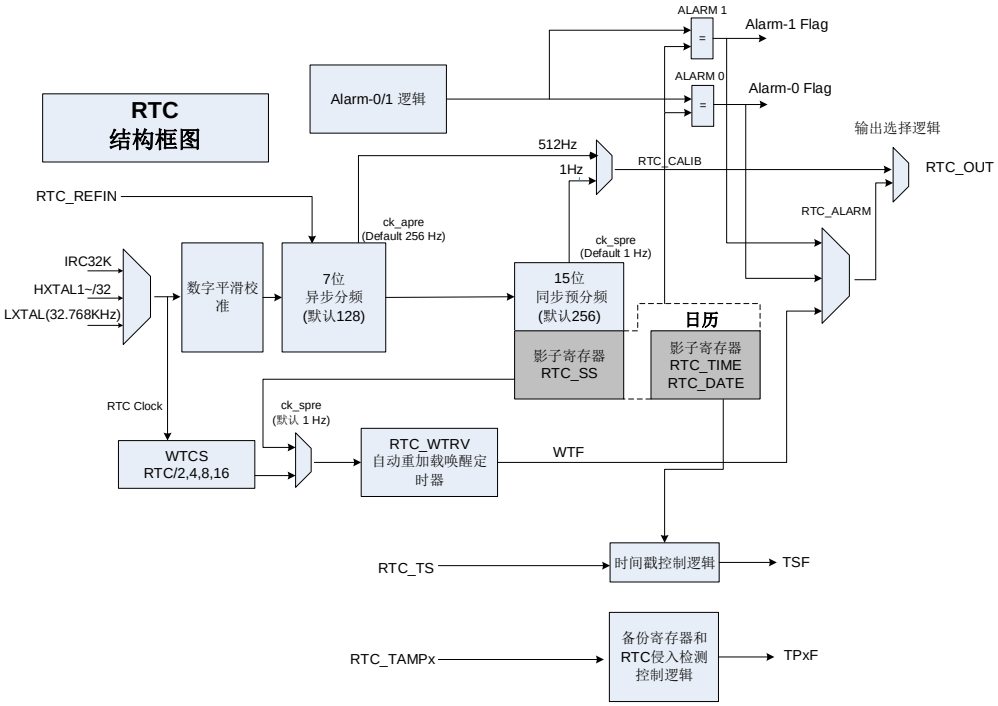
14.2. 主要特性

- 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。
- 两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
- 20个32位（共80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。

14.3. 功能描述

14.3.1. 结构框图

图 14-1. 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）：PC13 / PA3 / PA8；
  - 1Hz（默认预分频值）：PC13 / PA3 / PA8；
  - 闹钟事件（极性可配置）：PC13 / PA3 / PA8；
  - 自动唤醒事件（极性可配置）：PC13 / PA3 / PA8。
- 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；
  - 参考时钟输入 RTC_REFIN（50 或 60Hz）。

RTC_CTL [31]中的 OUT2EN 位，可以在 PA3 或 PA8 引脚上输出 RTC_OUT。此外，TAMPER 1 复用功能对应于 PA0 引脚。

14.3.2. 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 1 ~ 31（由 RCU_CFG0 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：

$$f_{ck\_apre} = \frac{f_{rtcclk}}{FACTOR\_A + 1} \quad (14\text{-}1)$$

$$f_{ck\_spre} = \frac{f_{ck\_apre}}{FACTOR\_S + 1} = \frac{f_{rtcclk}}{(FACTOR\_A + 1)*(FACTOR\_S + 1)} \quad (14\text{-}2)$$

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。

14.3.3. 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。

14.3.4. 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）位控制。当 ALRMxEN=1（x=0）并且闹钟所有位域的值与对应的日历时间值匹配，ALRMxF（x=0）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。

14.3.5. 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 18h
- WTCS[2：1] = 0b11，唤醒中断周期在 18h 到 36h

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。

14.3.6. 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_PSC，RTC_COSC，RTC_HRFC，RTC_SHIFTCTL，RTC_STAT 中的 INITM 位以及 RTC_CTL 中的 CS，S1H，A1H，REFEN 位。

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）位，禁用闹钟；
2. 设置 Alarm 寄存器（RTC_ALRMxTD / RTC_ALRMxSS）；
3. 设置寄存器 RTC_CTL 的 ALRMxEN 位，使能闹钟功能。

14.3.7. 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。

14.3.8. 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。

14.3.9. 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。

14.3.10. 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。

待机模式下和数字粗校准时，参考时钟检测功能不可用。

14.3.11. RTC 数字粗校准

RTC 有两种数字校准方法：数字粗校准和数字平滑校准。两种校准方法不能一起使用。

数字粗校准以异步预分频器输出为源，增加或者减少 ck_apre 时钟周期。

当 COSD = 0，在前 2xCOSS（COSS：0 到 31）分钟内，每分钟增加两个 ck_apre 时钟周期，这样配置会提前更新日历。

当 COSD = 1，在前 2xCOSS（COSS：0 到 31）分钟内，每分钟减少一个 ck_apre 时钟周期，这样配置会推迟更新日历。

仅能在初始化模式下配置数字粗校准，并且在清除 INITM 位后开始校准功能。整个校准过程持续 64 分钟。在 64 分钟内的前 2xCOSS 分钟内调整。

负校准的分辨率约为 2PPM，而正校准的分辨率约为 4PPM。

注意：在 RTC 时钟为 LXTAL 或者 HXTAL 时，可以进行数字粗校准。当 FACTOR_A<6 时，数字粗校准可能无法正确工作。

例子：

FACTOR_A 和 FACTOR_S 为默认值。RTC 时钟为 LXTAL，频率为 32.768 KHz。

在校准窗口内（64 分钟），仅在前 2xCOSS 分钟调整 ck_apre（256Hz）时钟周期。

如果 COSS = 1，表示 64 分钟内前两分钟需要调整。

这种情况下在每个校准窗口内（64min x 60s/min x 32768 周期/s）增加 512（两分钟，每分钟两个 ck_apre 时钟周期）或者减少 256（两分钟，每分钟一个 ck_apre 时钟周期）个 RTC 时钟周期。也就是说每次校准的分辨率为+4.069PPM 或者-2.035PPM。那么每个月的最小校准时间为+10.5 或者-5.27 s，最大校准时间为+5.45 到-2.72 分钟。

14.3.12. 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：

$$f_{cal} = f_{rtcclk} \times (1+\frac{FREQI\times512-CMSK}{2^{N}+CMSK-FREQI\times512}) \quad (14\text{-}3)$$

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：

$$f_{cal} = f_{rtcclk} \times (1+\frac{256-CMSK}{2^{N}+CMSK-256}) \quad (14\text{-}4)$$

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。

14.3.13. 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。

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。

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。

14.3.14. 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。

RTC 备份寄存器（RTC_BKPx）

RTC 备份寄存器处于 VDD 备份域中。从待机模式唤醒或系统复位操作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。

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。当检测到侵入事件，相应的标志位（TPxF）将会置位。如果侵入事件中断使能被启用（TPIE），侵入事件会产生一个中断。

当侵入检测事件发生时，备用寄存器将复位，除非 RTC_TAMP 寄存器中的 TPxNOER 位置 1。在 RTC_TAMP 寄存器中的 BKERASE 位置 1，可以通过软件复位将备份寄存器中的数据擦除。

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：Tamper0 上的侵入检测功能即使 VDD 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。

14.3.15. 校准时钟输出

如果 COEN 位设置为 1，PC13 / PA3 / PA8 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：

$$f_{rtc\_calib} = \frac{f_{rtcclk}}{(FACTOR\_A+1)\times(FACTOR\_S+1)} \quad (14\text{-}5)$$

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。

14.3.16. 闹钟输出

当 OS 控制位被设置为 0x01 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 ALRMxF 值。

RTC_CTL 寄存器中的 OPOL 位可以配置 ALRMxF 位输出时候的极性，因此 RTC_ALARM 的输出电平有可能与相应的位值相反。

14.3.17. RTC 省电模式管理

表 14-1. 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
深度睡眠模式	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
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

14.3.18. 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 14-2. RTC 中断控制

中断	事件标志	控制位	清除中断标志位	退出睡眠模式	退出深度睡眠模式和待机模式
闹钟0	ALRM0F	ALRM0IE	ALRM0FC位写1	Y	Y(*)

中断	事件标志	控制位	清除中断标志位	退出睡眠模式	退出深度睡眠模式和待机模式
闹钟1	ALRM1F	ALRM1IE	ALRM1FC位写1	Y	Y(*)
唤醒	WTF	WTIE	WTFC位写1	Y	Y(*))
时间戳	TSF	TSIE	TSFC位写1	Y	Y(*)
侵入x	TPxF	TPxIE	TPxFC位写1	Y	Y(*)

注意：（*）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。

14.4. RTC 寄存器

RTC 基地址：0x4000 2800

14.4.1. 时间寄存器（RTC_TIME）

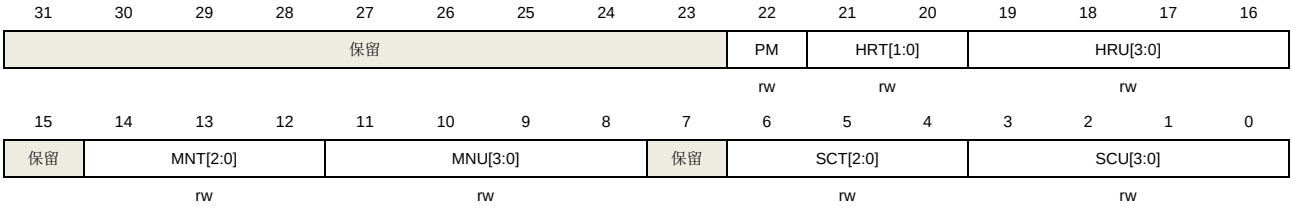
偏移地址：0x00

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

14.4.2. 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[3:0]				保留		DAYT[1:0]		DAYU[3:0]			
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[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

14.4.3. 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							COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
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	CCEN	CS	BPSHAD	REFEN	TSEG	WTCS[2:0]		

rw 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 PA3 或者 PA8
30:24	保留	必须保持复位值。
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。 0x0：禁用 RTC_ALARM 输出 0x1：启用闹钟 0 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	CCEN	粗校准功能使能 0：禁用粗校准功能 1：启用粗校准功能 注意：FACTOR_A 必须大于 6 才能启用，并且只能在初始化状态下写入。
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 2^{16}

14.4.4. 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,除 RTC_STAT[14:8]外。

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															SCPF
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	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	rw	r	rc_w0	r	r	r	r	r

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	SCPF	平滑校准挂起标志 对 RTC_HRFC 进行软件写操作，该位被硬件置 1。当平滑校准周期完成后，该位被硬件清 0。
15	保留	必须保持复位值。
14	TP1F	RTC_TAMP1 检测标志 当在 tamper1 输入引脚上检测到侵入时，硬件将其置 1。可以通过向该位软件写 0 来清除。
13	TP0F	RTC_TAMP0 检测标志 当在 tamper0 输入引脚上检测到侵入时，硬件将其置 1。可以通过向该位软件写 0 来清除。
12	TSOVRF	时间戳溢出标志 如果之前设置了 TSF 位，则在检测到时间戳事件时，该位将由硬件置 1。可以通过向该位软件写 0 来清除。
11	TSF	时间戳标志 当检测到时间戳事件时，由硬件设置 1。可以通过向该位软件写 0 来清除。

10	WTF	唤醒定时器标志 当唤醒定时器减少到 0 时由硬件置 1。在 WTF 再次设置为 1 之前，必须至少在 1.5 个 RTC 时钟周期内清除该标志。
9	ALRM1F	闹钟 1 发生标志 当前时间/日期与闹钟 1 设置值的时间/日期匹配时，由硬件设置为 1。可以通过向该位软件写 0 来清除。
8	ALRM0F	闹钟 0 发生标志 当前时间/日期与闹钟 0 设置值的时间/日期匹配时，由硬件设置为 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	唤醒时间写使能标志位 0：不允许更新唤醒时间 1：允许更新唤醒时间
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 Alarm1 寄存器设置 1：允许修改 Alarm1 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 Alarm0 寄存器设置

1：允许修改 Alarm0 寄存器设置

14.4.5. 预分频寄存器（RTC_PSC）

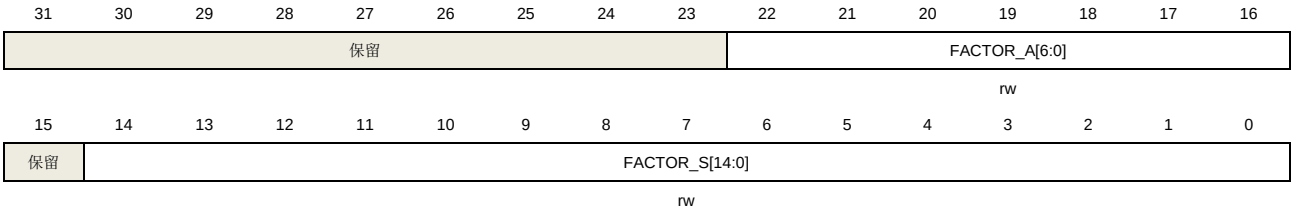
偏移地址：0x10

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）

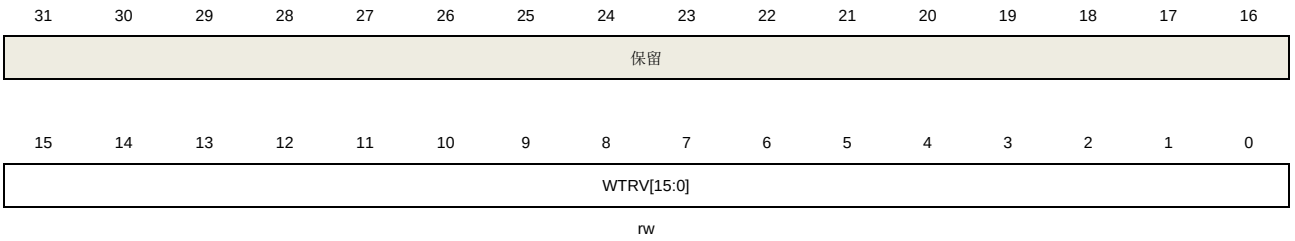
14.4.6. 唤醒定时器寄存器（RTC_WUT）

偏移地址：0x14

系统复位：无影响

备份域复位值：0x0000 FFFF

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

14.4.7. 粗校准寄存器（RTC_COSC）

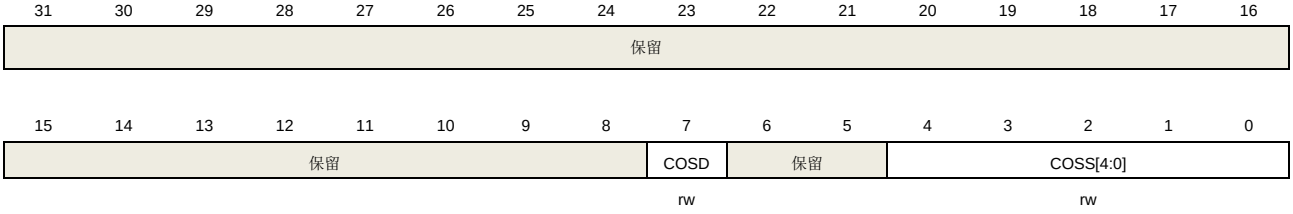
偏移地址：0x18

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。



位/位域	名称	描述
31:8	保留	必须保持复位值。
7	COSD	粗校准方向 0：增加日历更新频率 1：降低日历更新频率
6:5	保留	必须保持复位值。
4:0	COSS[4:0]	粗校准步伐 当 COSD=0： 0x00：+0 PPM 0x01：+4 PPM（近似值） 0x02：+8 PPM（近似值） .... 0x1F：+126 PPM（近似值） 当 COSD=1： 0x00：-0 PPM 0x01：-2 PPM（近似值） 0x02：-4 PPM（近似值）

....

0x1F：-63 PPM（近似值）

14.4.8. 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

14.4.9. 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

14.4.10. 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。

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								WPK[7:0]							
								w							

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

14.4.11. 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。

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
SSC[15:0]															
r															

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）

14.4.12. 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。

14.4.13. 时间戳时间寄存器（RTC_TTS）

偏移地址：0x30

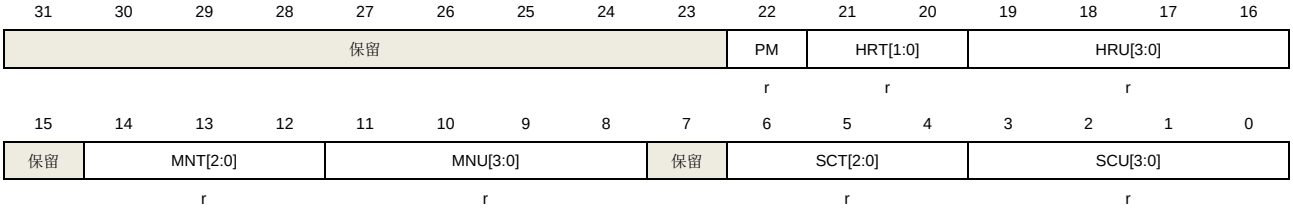
备份域复位值：0x0000 0000

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

14.4.14. 时间戳日期寄存器（RTC_DTS）

偏移地址：0x34

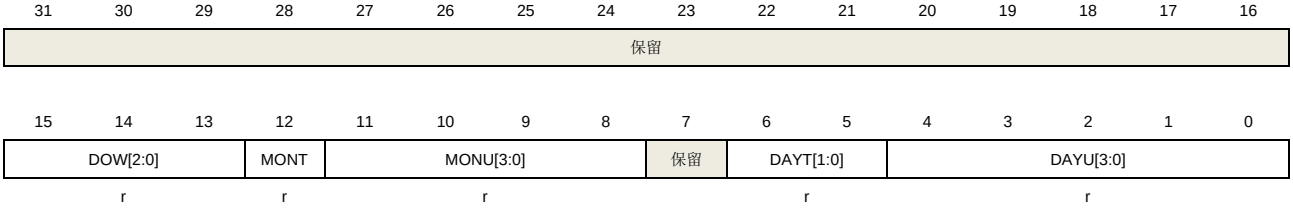
备份域复位值：0x0000 0000

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	保留	必须保持复位值。
6:5	DAYT[1:0]	日期十位值，以 BCD 码形式存储
4:0	DAYU[3:0]	日期个位值，以 BCD 码形式存储

14.4.15. 时间戳亚秒寄存器（RTC_SSTS）

偏移地址：0x38

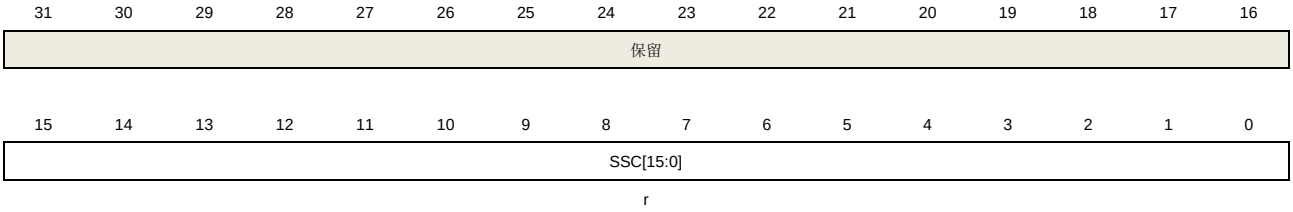
备份域复位：0x0000 0000

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。

14.4.16. 高精度频率补偿寄存器（RTC_HRFC）

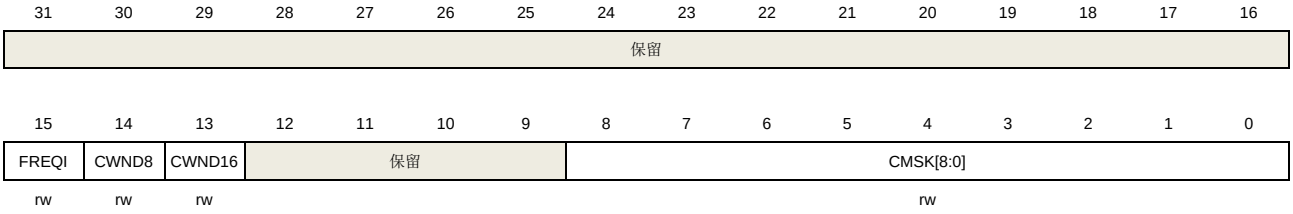
偏移地址：0x3C

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

14.4.17. 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
BKERASE	保留										TP1NOER	TP0NOER	AOT	保留	
											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	保留		TP1EG	TP1EN	TPIE	TP0EG	TP0EN
rw	rw		rw		rw			rw			rw	rw	rw	rw	rw

位/位域	名称	描述
31	BKPERASE	备份寄存器擦除 向该位写“1”将复位备份寄存器。写入 0 无效。该位始终读为 0。
30:21	保留	必须保持复位值。
20	TP1NOER	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	TP0NOER	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	AOT	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
6:5	保留	必须保持复位值。
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。

14.4.18. 闹钟 0 亚秒寄存器（RTC_ALRM0SS）

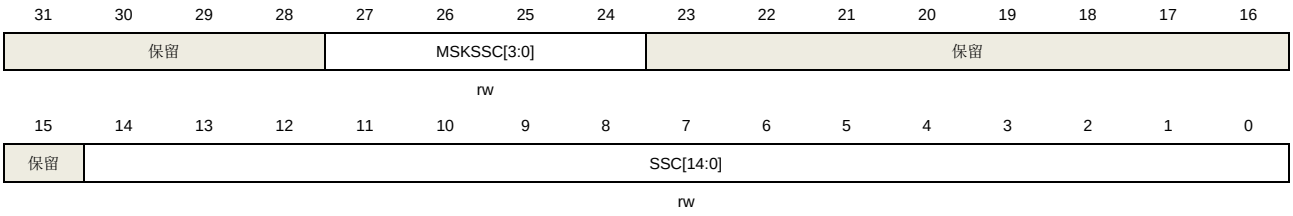
偏移地址：0x44

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。

14.4.19. 闹钟 1 亚秒寄存器（RTC_ALRM1SS）

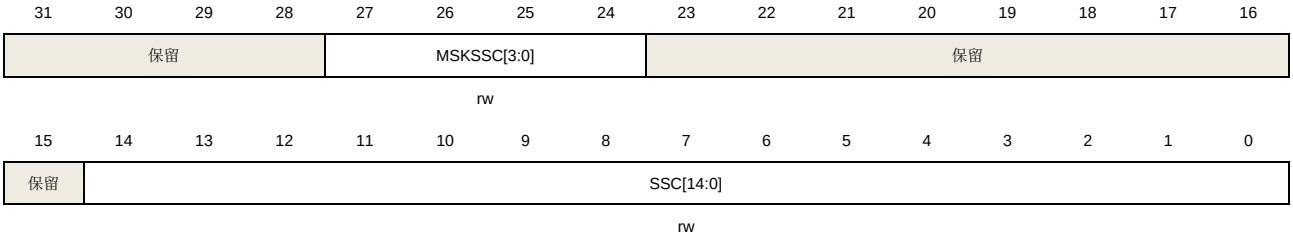
偏移地址：0x48

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。

14.4.20. 备份寄存器（RTC_BKPx）（x = 0…19）

偏移地址： 0x70 + 0x04 * x

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。当侵入检测标志位 TPxF 置 1，这些寄存器会被复位。

15. 定时器（TIMER）

表 15-1. 定时器（TIMERx）分为六种类型

定时器	定时器 0	定时器 1/2	定时器 15/16	定时器 5
类型	高级	通用 L0	通用 L4	基本
预分频器	16 位	16 位	16 位	16 位
计数器	16 位	32 位（TIMER1/2）	16 位	16 位
计数模式	向上，向下，中央对齐	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	4	1	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)	×	×
DMA	●	●	●	●(3)
Debug 模式	●	●	●	●

（1）

TIMER0	**ITI0: 0**	**ITI1:** TIEMR1_TRGO	**ITI2:** TIMER2_TRGO	**ITI3:** 0

（2）

TIMER1	**ITI0:** TIMER0_TRGO	**ITI1:** 0	**ITI2:** TIMER2_TRGO	**ITI3:** 0
TIMER2	**ITI0:** TIMER0_TRGO	**ITI1:** TIEMR1_TRGO	**ITI2:** 0	**ITI3:** 0

（3） 只有更新事件可以产生 DMA 请求，定时器 5 中没有 DMAS 位（DMA 请求源选择位）。

15.1. 高级定时器（TIMERx,x=0）

15.1.1. 简介

高级定时器（TIMER0）是四通道定时器，支持输入捕获和输出比较。可以产生 PWM 信号控制电机和电源管理。高级定时器含有一个 16 位无符号计数器。

高级定时器是可编程的，可以用来计数，其外部事件可以驱动其他定时器。

高级定时器包含了一个死区时间插入模块，非常适合电机控制。

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。

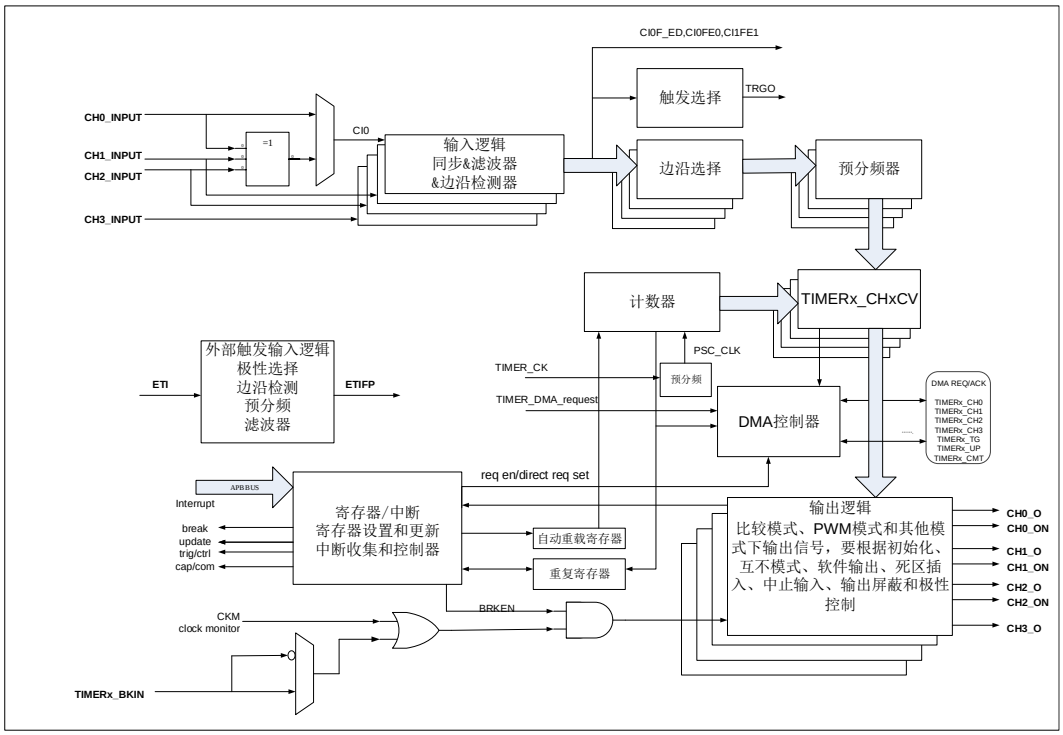
15.1.2. 主要特征

- 总通道数：4；
- 计数器宽度：16位；
- 定时器时钟源可选：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；
- 正交编码器接口：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；
- 可编程的死区时间；
- 自动重装载功能；
- 可编程的计数器重复功能；
- 中止输入功能；
- 中断输出和DMA请求：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。

15.1.3. 结构框图

*图15-1. 高级定时器结构框图*提供了高级定时器的内部配置细节。

图 15-1. 高级定时器结构框图



15.1.4. 功能描述

时钟源配置

高级定时器的时钟源可以是内部时钟源 CK_TIMER，或者是由 TSCFGy[3:0]位确定的时钟源，TSCFGy[3:0]位于 SYSCFG_TIMER0CFG，（y=0,1…6）。

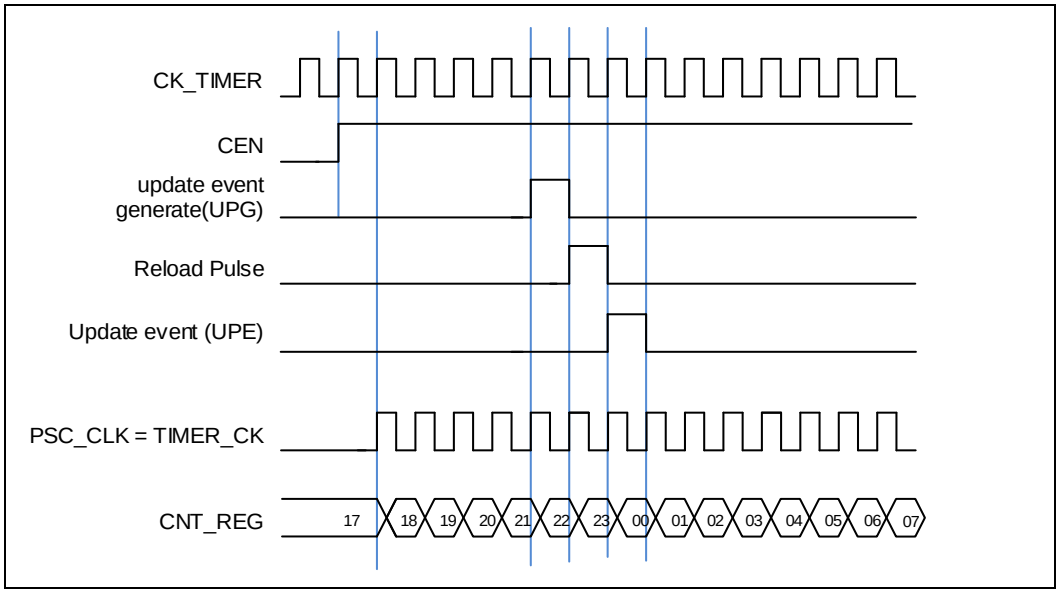
- TSCFGy[3:0] =4’b0000，TSCFGy[3:0]位于SYSCFG_TIMER0CFG，（y=0,1…6），定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

当 TSCFGy[3:0] =4’b0000，TSCFGy[3:0]位于 SYSCFG_TIMER0CFG，（y=0,1…6），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TSCFGy[3:0] !=4’b0000，TSCFGy[3:0]位于SYSCFG_TIMER0CFG，（y=0,1,2,6），预分频器被其他时钟源（由TSCFG6[3:0]区域选择）驱动，更多细节在下文说明，当TSCFGy[3:0]（y=3,4,5）设置为有效值时，计数器预分频器时钟源由内部时钟TIMER_CK驱动。

图 15-2. 内部时钟分频为 1 时，计数器的时序图



- TSCFG6[3:0] !=4'b0000（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 TSCFG6[3:0]为 0x5，0x6 或 0x7 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 TSCFG6[3:0]为 0x1，0x2，0x3 或者 0x4。

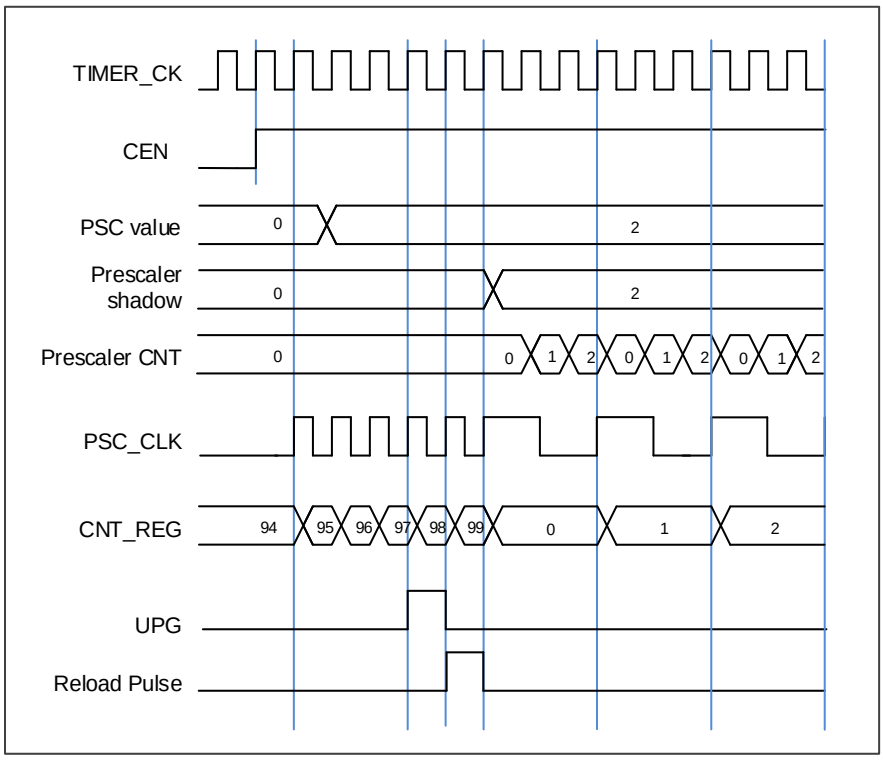
- SMC1=1'b1（外部时钟模式1），定时器选择外部输入引脚ETI作为时钟源。

计数器预分频器可以在外部引脚 ETI 的每个上升沿或下降沿计数。这种模式可以通过设置 TIMERx_SMCFG 寄存器中的 SMC1 位为 1 来选择。另一种选择 ETI 信号作为时钟源方式是，设置 TSCFG6[3:0]为 0x8。注意 ETI 信号是通过数字滤波器采样 ETI 引脚得到的。如果选择 ETI 信号为时钟源，触发控制器包括边沿监测电路将在每个 ETI 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

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 15-3. 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。如果设置了重复计数器，在（TIMERx_CREP+1）次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。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（重复计数寄存器，自动重载寄存器，预分频寄存器）都将被更新。

*图15-4. 向上计数时序图，PSC=0/2*和*图15-5. 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 15-4. 向上计数时序图，PSC=0/2

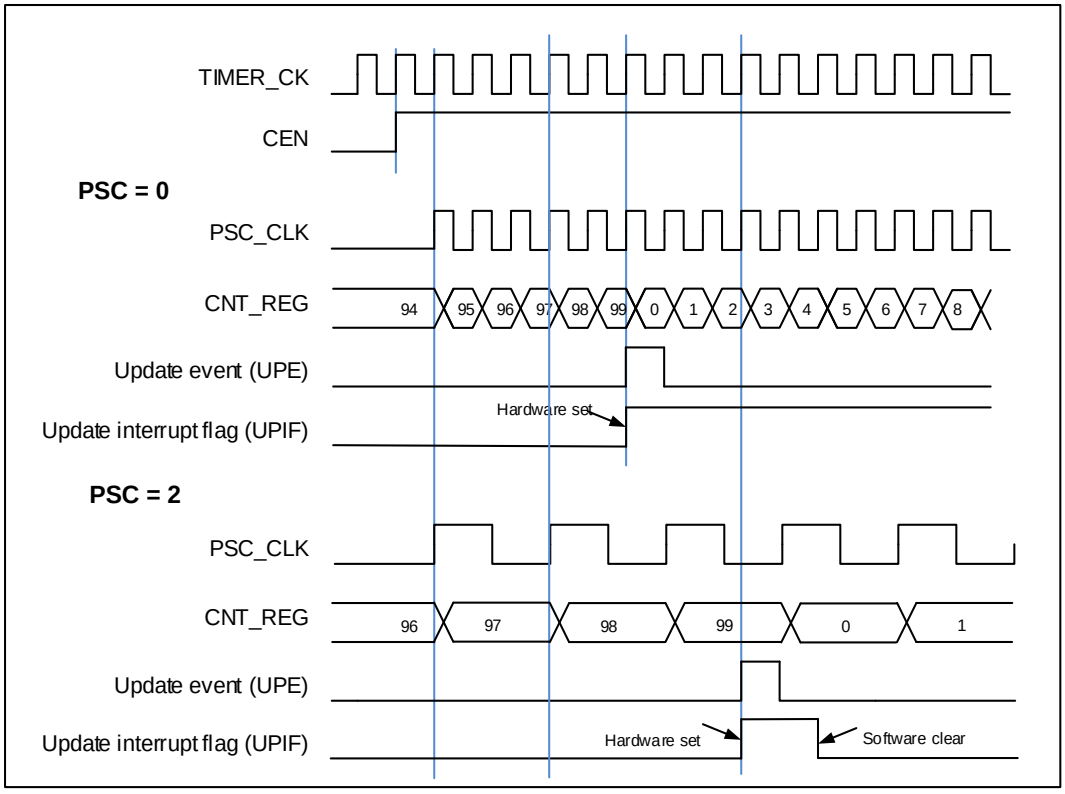
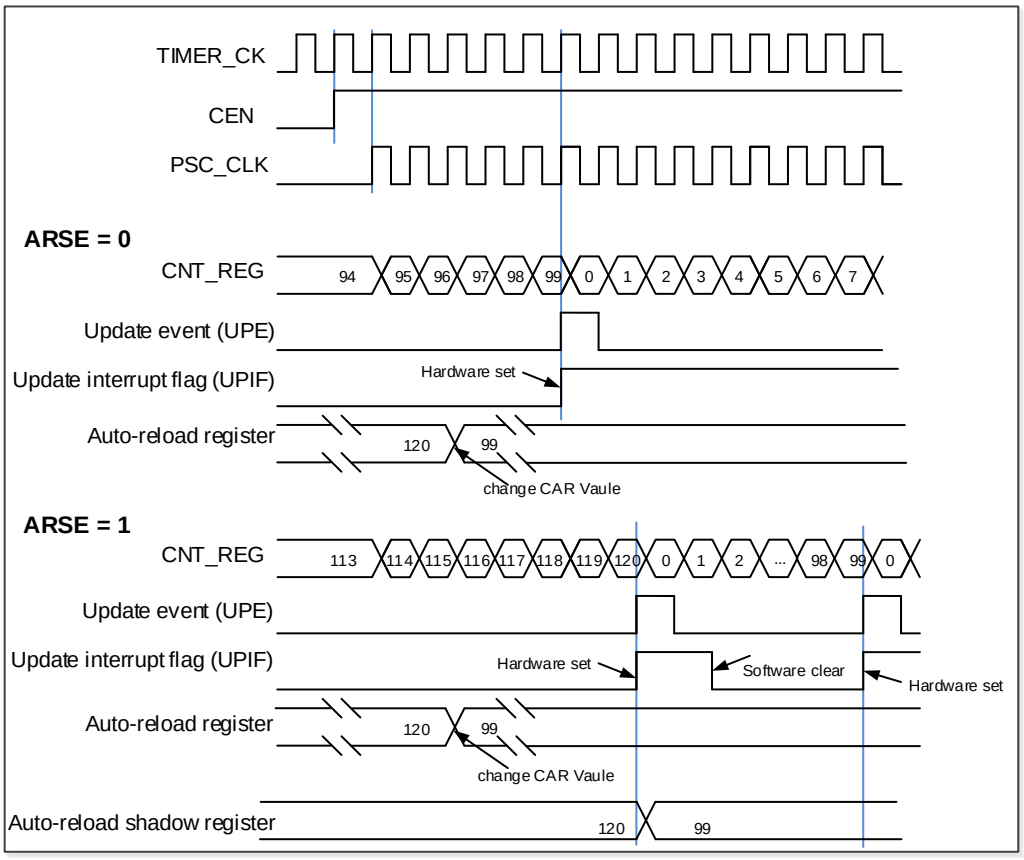


图 15-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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。如果设置了重复计数器，在（TIMERx_CREP+1）次下溢后产生更新事件，否则在每次下溢时都会产生更新事件。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（重复计数器，自动重载寄存器，预分频寄存器）都将被更新。

*图15-6. 向下计数时序图，PSC=0/2*和*图15-7. 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 15-6. 向下计数时序图，PSC=0/2

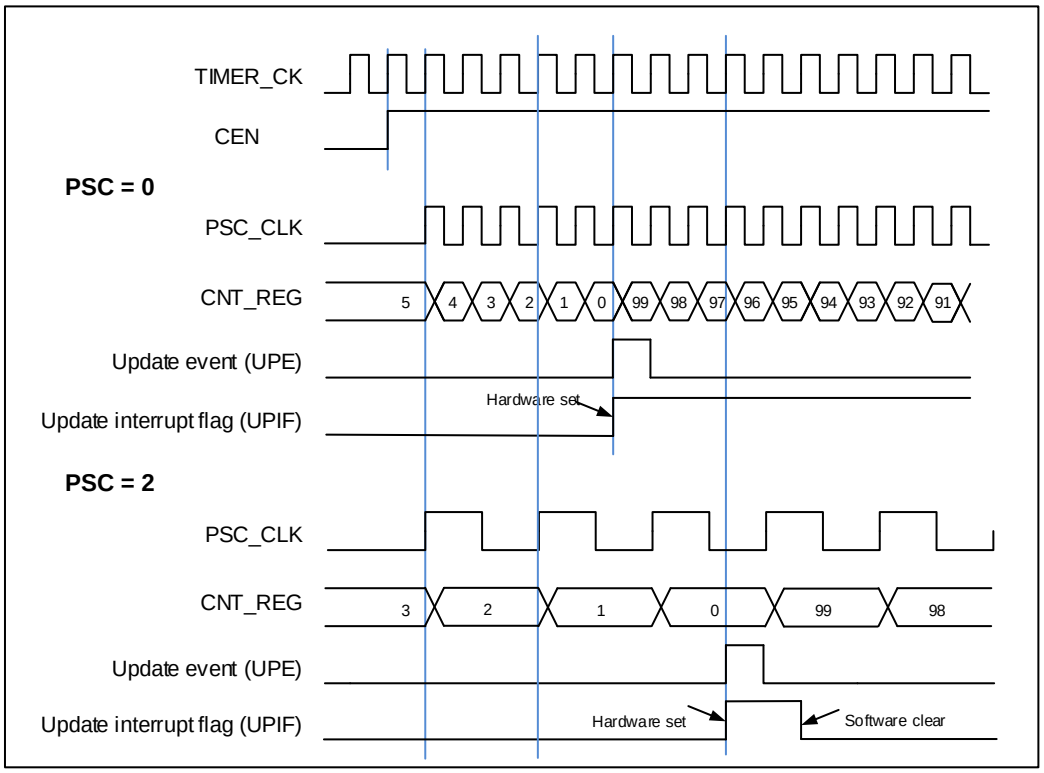
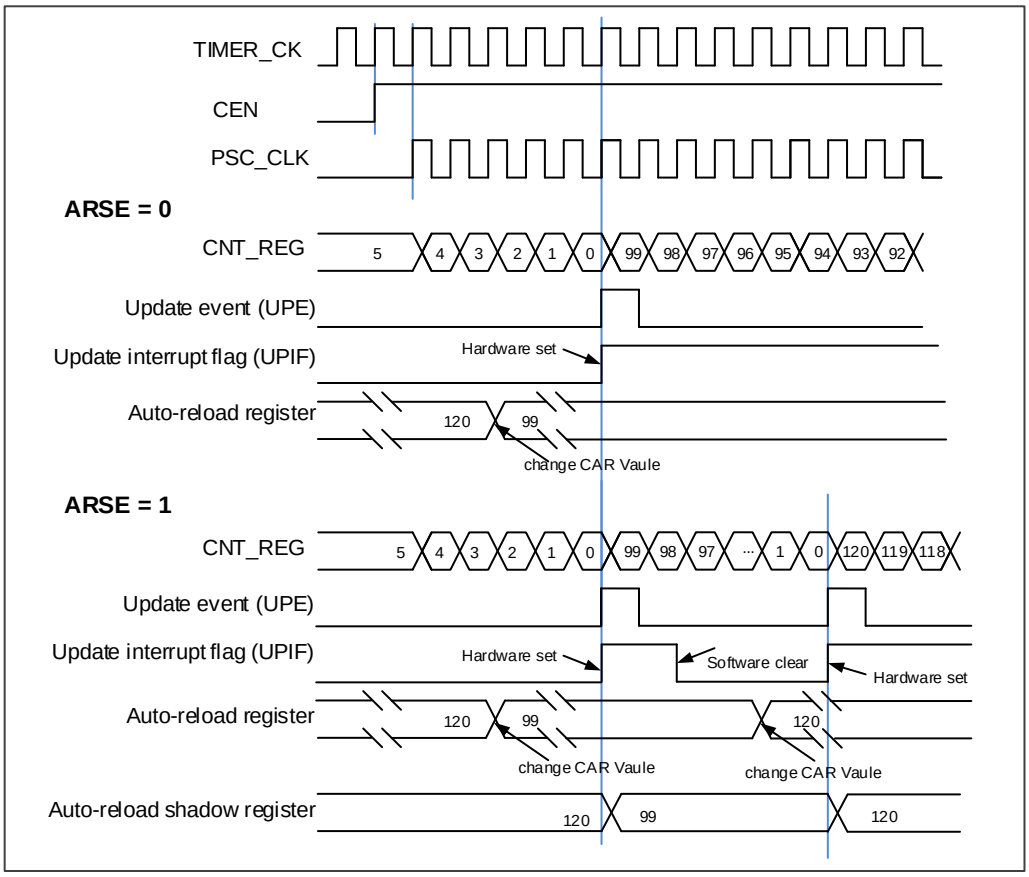


图 15-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



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。

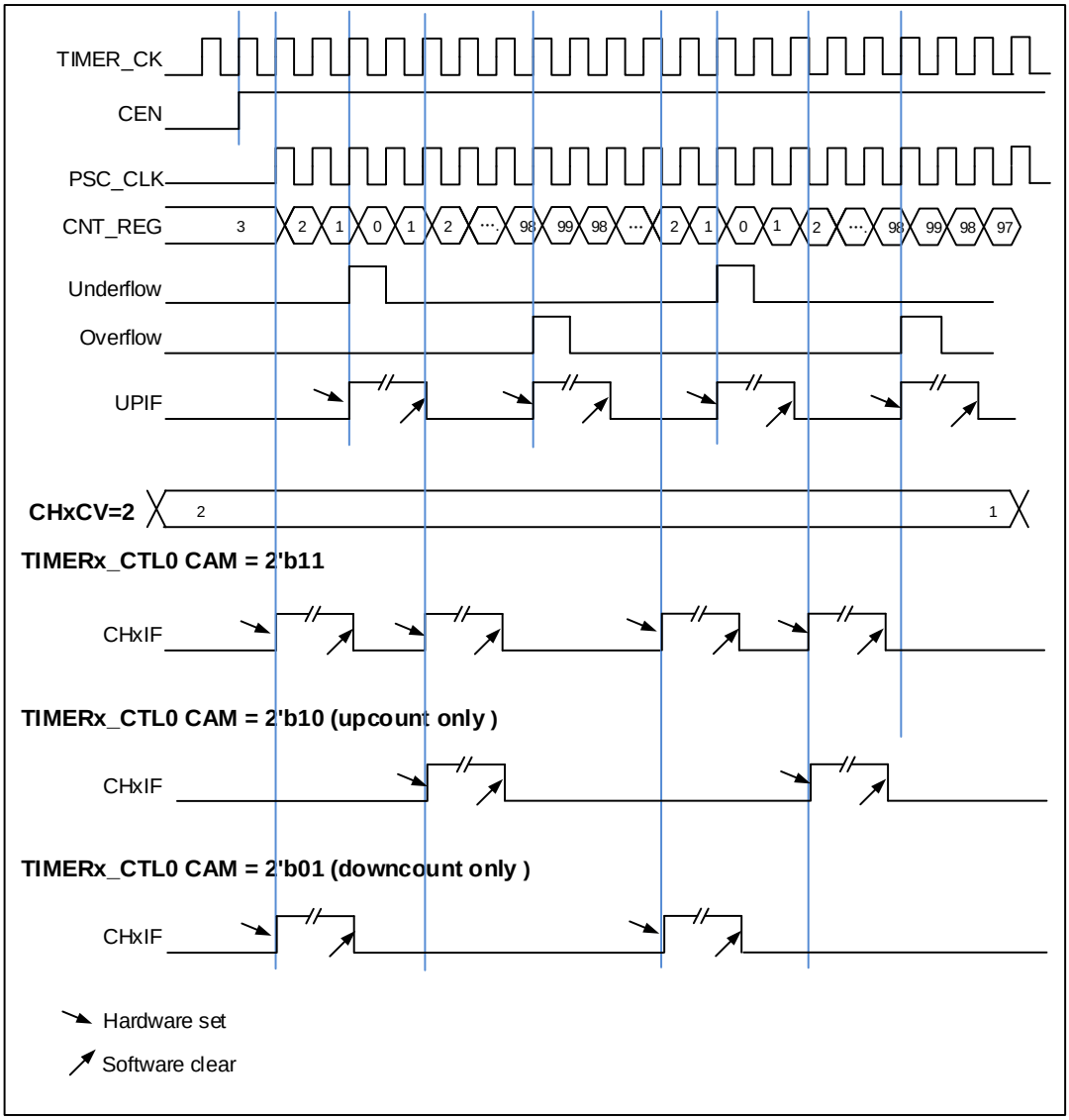
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1。但是 CHxIF 位是否置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图 15-8. 中央计数模式计数器时序图*。

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（重复计数器，自动重载寄存器，预分频寄存器）都将被更新。

*图 15-8. 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，计数器的时序图。

图 15-8. 中央计数模式计数器时序图



更新事件（来自上溢/下溢）频率配置

更新事件的生成频率（来自上溢和下溢事件）可以通过 TIMERx_CREP 寄存器进行配置。重复计数器是用来在 N+1 个计数周期之后产生更新事件，更新定时器的寄存器，N 为 TIMERx_CREP 寄存器的 CREP。重复计数器在每次计数器上溢和下溢时递减（向上计数模式中不存在下溢事件；向下计数模式中不存在上溢事件）。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以重载 TIMERx_CREP 寄存器中 CREP 的值并产生一个更新事件。

新写入的 CREP 值将在下一次更新事件到来时生效。当 CREP 的值为奇数，并且计数器在中央对齐模式下计数时，更新事件发生在上溢或下溢取决于写入的 CREP 值何时生效。如果在写入奇数到 CREP 寄存器后由软件生成更新事件（UPG 位置 1），则在下溢时产生更新事件。如果在写入奇数到 CREP 寄存器后下一个更新事件发生在上溢，此后将在上溢时产生更新事件。

图 15-9. 中央计数模式下计数器重复时序图

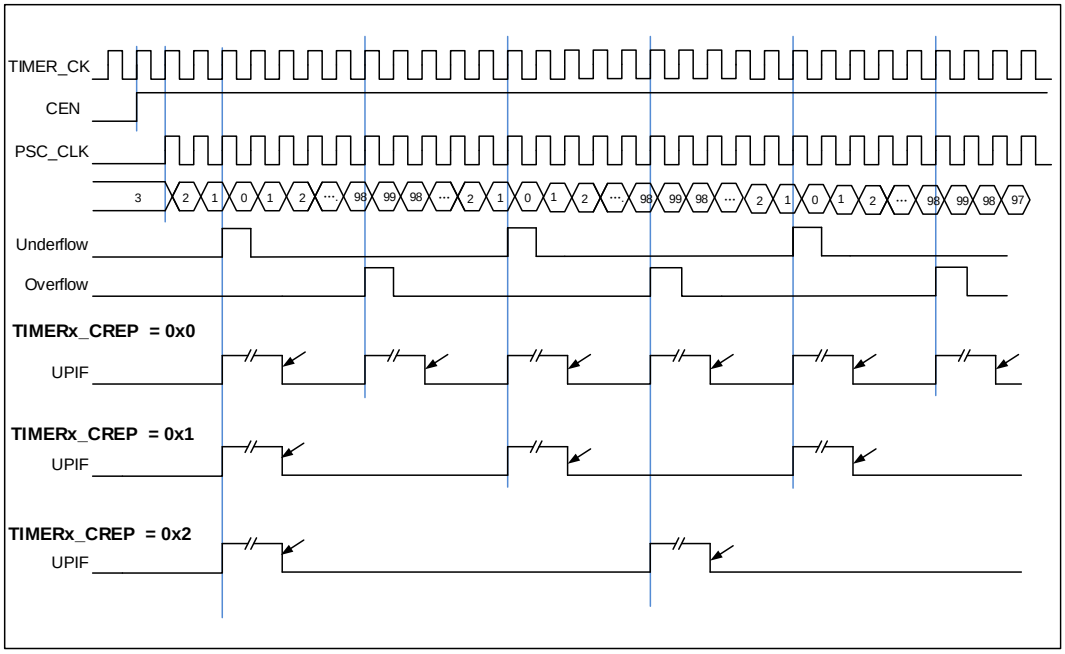


图 15-10. 在向上计数模式下计数器重复时序图

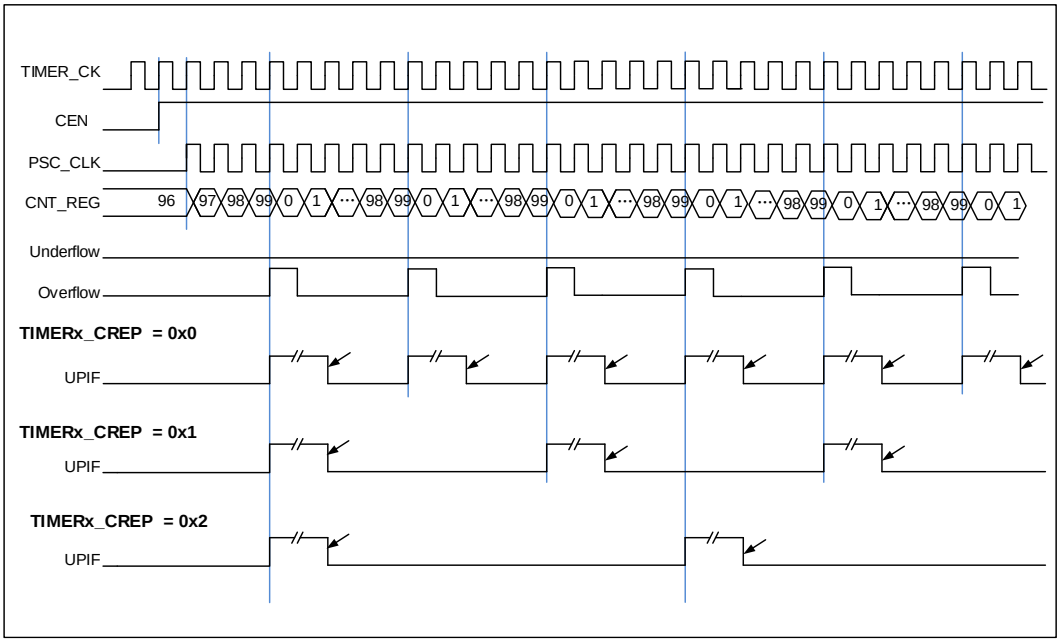
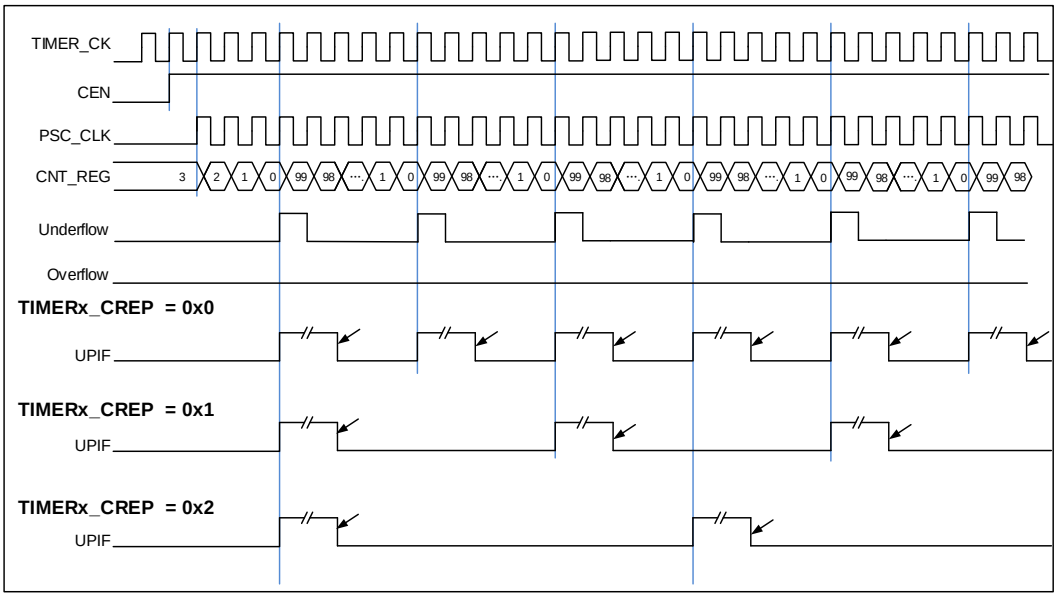


图 15-11. 在向下计数模式下计数器重复时序图



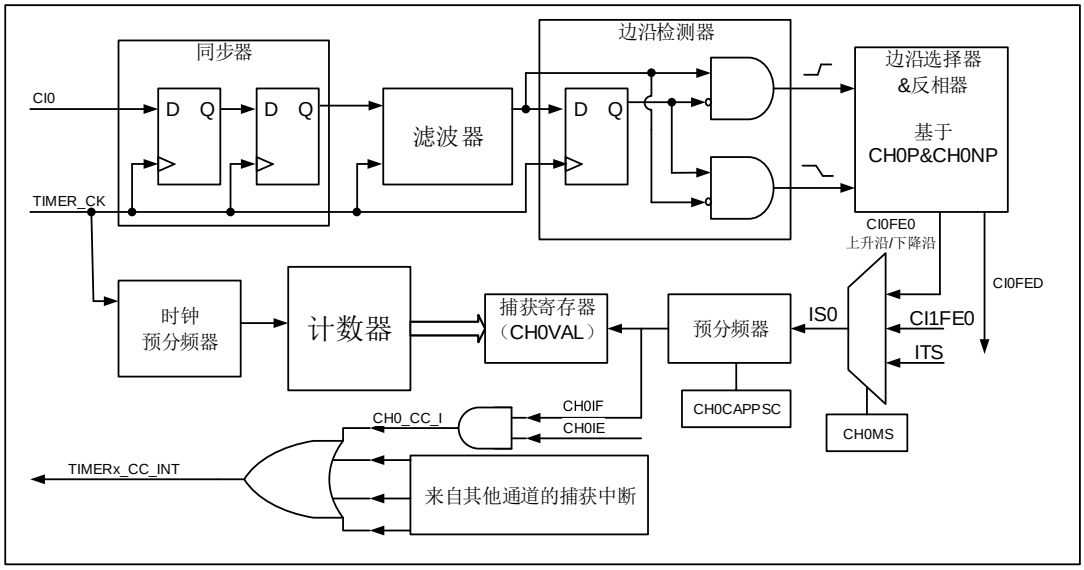
输入捕获和输出比较通道

高级定时器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

输入捕获模式允许通道测量一个波形的时序，频率，周期和占空比等。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，若 CHxIE=1 则产生通道中断。

图 15-12. 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。通过配置CHxMS，还可以选择其他通道的输入信号或内部触发信号作为捕获信号。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，TIMERx_CHxCV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/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_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 位置 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，判断相应的中断和 DMA 请求是否被提出。

直接产生：软件设置 CHxG 位，会直接产生中断和 DMA 请求。

通道输入捕获功能也可用来测量 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，同时设置上升沿捕获。配置 TIMERx_CHCTL0 寄存器中 CH1MS 为 2'b10，选择通道 1 捕获信号为 CI0，同时设置下降沿捕获。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 15-13. 通道输出比较原理（带有互补输出的通道，x=0,1,2）

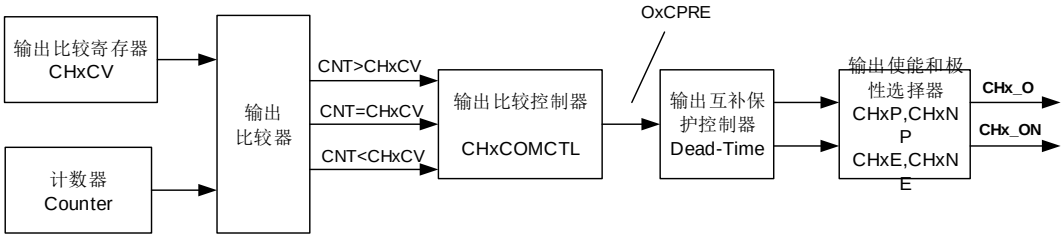


图 15-14. 通道输出比较原理

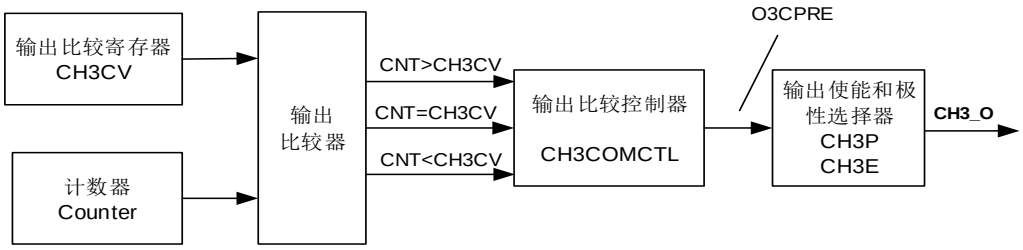


*图15-13. 通道输出比较原理（带有互补输出的通道，x=0,1,2）*和*图15-14. 通道输出比较原理*分别给出了输出比较的原理电路。通道输出信号CHx_O/CHx_ON与OxCPRE信号（详情请见*通道输出准备信号*）的关系描述如下：OxCPRE信号高电平有效，CHx_O/CHx_ON的输出情况与OxCPRE信号，CHxP/CHxNP位和CHxE/CHxNE位有关（具体情况请见TIMERx_CHCTL2寄存器中的描述）。例如：

1）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。

2）当设置 CHxNP=1（CHx_ON 低电平有效，与 OxCPRE 输出极性相反）、CHxNE=1（CHx_ON 输出使能）时：

若 OxCPRE 输出有效（高）电平，则 CHx_ON 输出有效（低）电平；

若 OxCPRE 输出无效（低）电平，则 CHx_ON 输出无效（高）电平。

当 CH0_O 和 CH0_ON 同时输出时，CH0_O 和 CH0_ON 的具体输出情况还与 TIMERx_CCHP 寄存器中的相关位（ROS、IOS、POE 和 DTCFG 等位）有关。详情请见*通道输出互补 PWM*。

在通道输出比较功能，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 TIMERx_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 TIMERx_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/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。

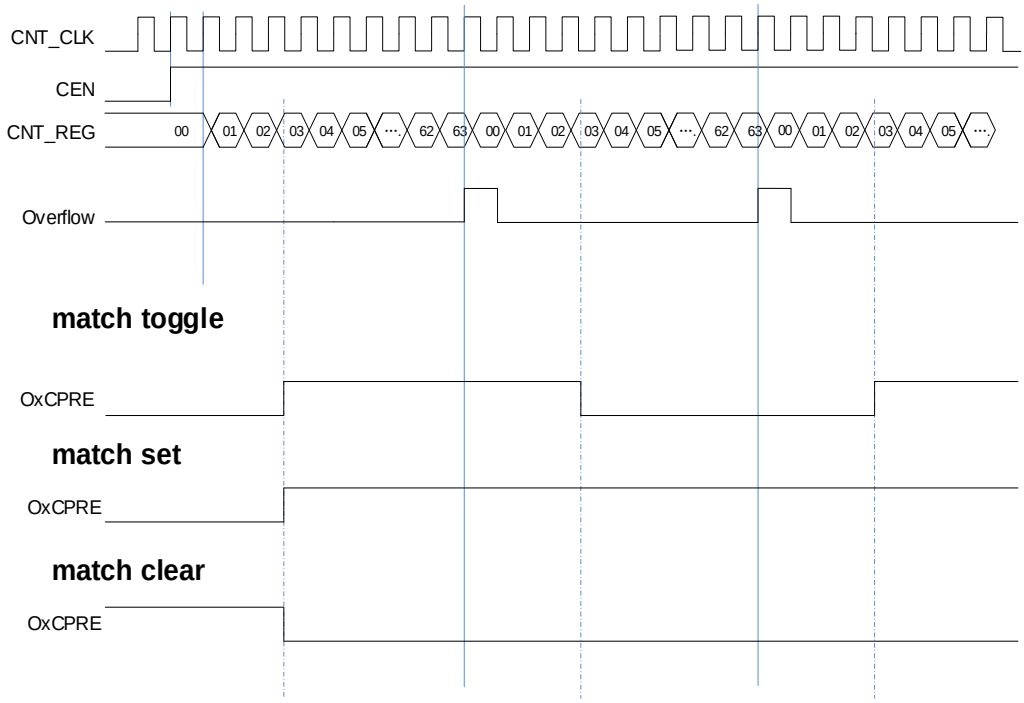
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

TIMERx_CHxCV 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15-15. 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 15-15. 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，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 15-16. EAPWM 时序图*显示了 CAPWM 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 15-17. 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，通道输出一直为有效电平；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，通道输出一直为无效电平。

图 15-16. EAPWM 时序图

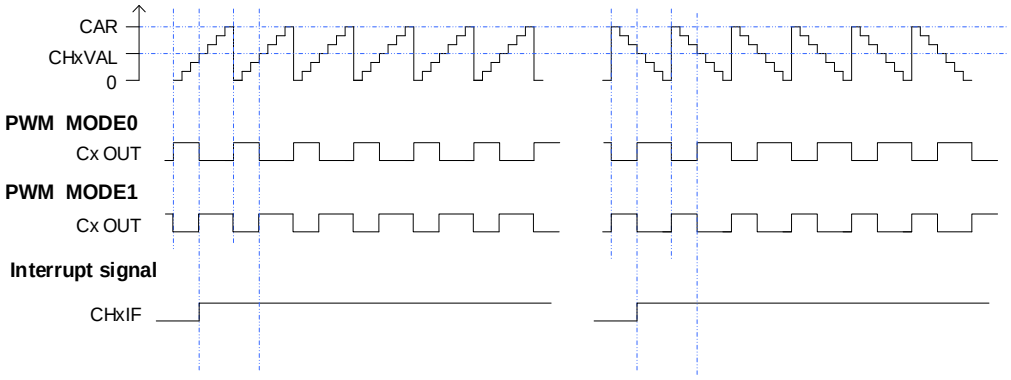
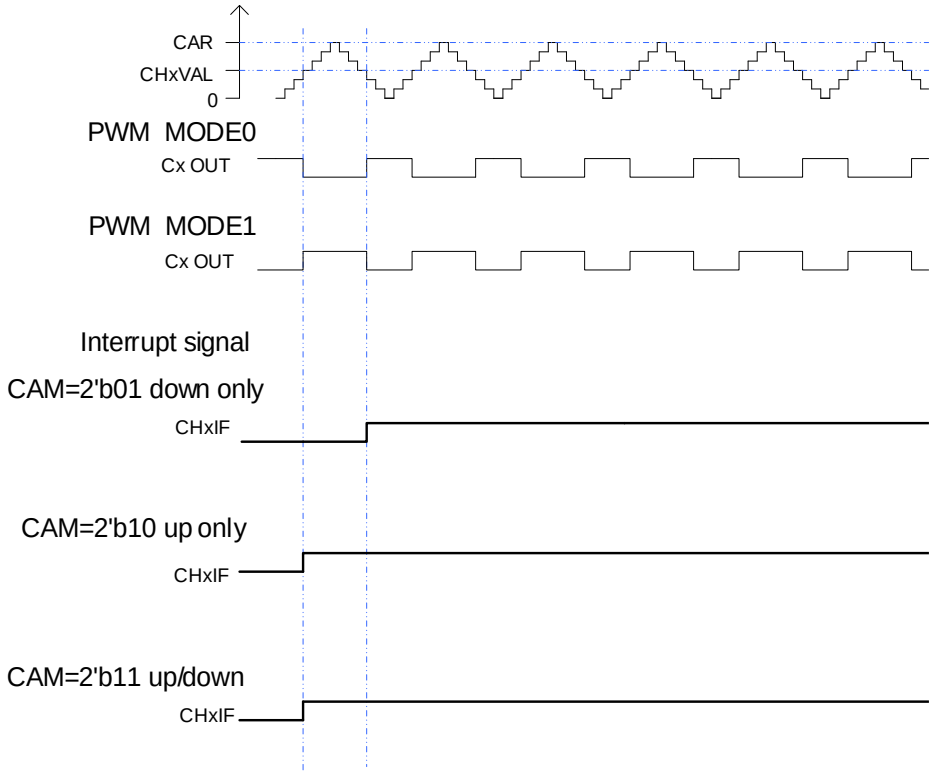


图 15-17. CAPWM 时序图



通道输出准备信号

根据*图15-13. 通道输出比较原理（带有互补输出的通道，x=0,1,2）*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。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。

通道输出互补 PWM

CHx_O 和 CHx_ON 是一对互补输出通道，这两个信号不能同时有效。TIMERx 有四路通道，只有前三路有互补输出通道。互补信号 CHx_O 和 CHx_ON 是由一组参数来决定：TIMERx_CHCTL2 寄存器中的 CHxEN 和 CHxNEN 位，TIMERx_CCHP 寄存器中的 POEN、ROS 和 IOS 位，TIMERx_CTL1 寄存器中的 ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 CHxNP 位来决定。

表 15-2. 由参数控制的互补输出表

<table>
<tr><th colspan="5">互补参数</th><th colspan="2">输出状态</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>CHxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON 输出禁能(1)</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O/CHx_ON输出关闭状态(2)：
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，CHx_ON = ISOxN (3)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">CHx_O/CHx_ON输出关闭状态：
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，CHx_ON = ISOxN</td></tr>
<tr><td rowspan="3">1</td><td rowspan="3">0</td><td rowspan="3">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON输出禁能</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O输出禁能</td><td>CHx_ON=OxCPRE⊕(4)CHxNP
CHx_ON输出使能</td></tr>
<tr><td>1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>CHx_ON = LOW
CHx_ON输出禁能</td></tr>
</table>

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON=(!OxCPRE)(5)⊕ CHxNP CHx_ON输出使能
	1		0	0	CHx_O = CHxP CHx_O输出关闭状态	CHx_ON = CHxNP CHx_ON输出关闭状态
				1	CHx_O = CHxP CHx_O输出关闭状态	CHx_O=OxCPRE⊕CHxNP CHx_ON输出使能
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出关闭状态
				1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON=（!OxCPRE）⊕ CHxNP CHx_ON输出使能

注意：

（1） 输出禁能：CHx_O / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，无上下拉时为悬空高阻态；

（2） 输出关闭状态：CHx_O / CHx_ON 输出无效电平（CHx_O = 0⊕CHxP = CHxP）；

（3） 详情见中止模式章节。

（4） ⊕：异或操作；

（5） (!OxCPRE)：OxCPRE 信号的互补信号。

互补 PWM 插入死区时间

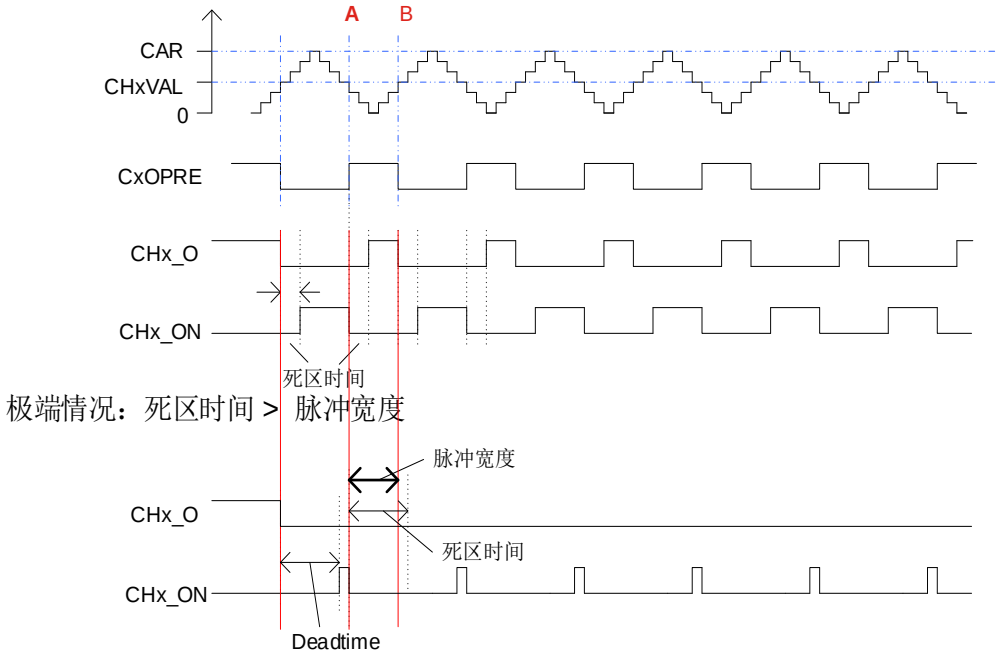
设置 CHxEN 和 CHxNEN 为 1’b1 的同时，设置 POEN 为 1，死区插入就会被使能。DTCFG 位域定义了死区时间，死区时间对除了通道 3 以外的通道有效。死区时间的细节，请参考 TIMERx_CCHP 寄存器。

死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM 模式 0，当通道 x 匹配事件发生时（TIMERx 计数器=CHxVAL），OxCPRE 反转。在 *图 15-18. 带死区时间的互补输出*中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 CHx_ON 信号立刻变为低电平。同样，在 B 点，通道 x 匹配事件再次发生（TIMERx 计数器=CHxVAL），OxCPRE 信号被清 0，CHx_O 信号被立即清零，CHx_ON 信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些极端事件发生，例如：如果死区延时大于或者等于 CHx_ON 信号的占空比，CHx_ON 信号一直为无效值。

图 15-18. 带死区时间的互补输出



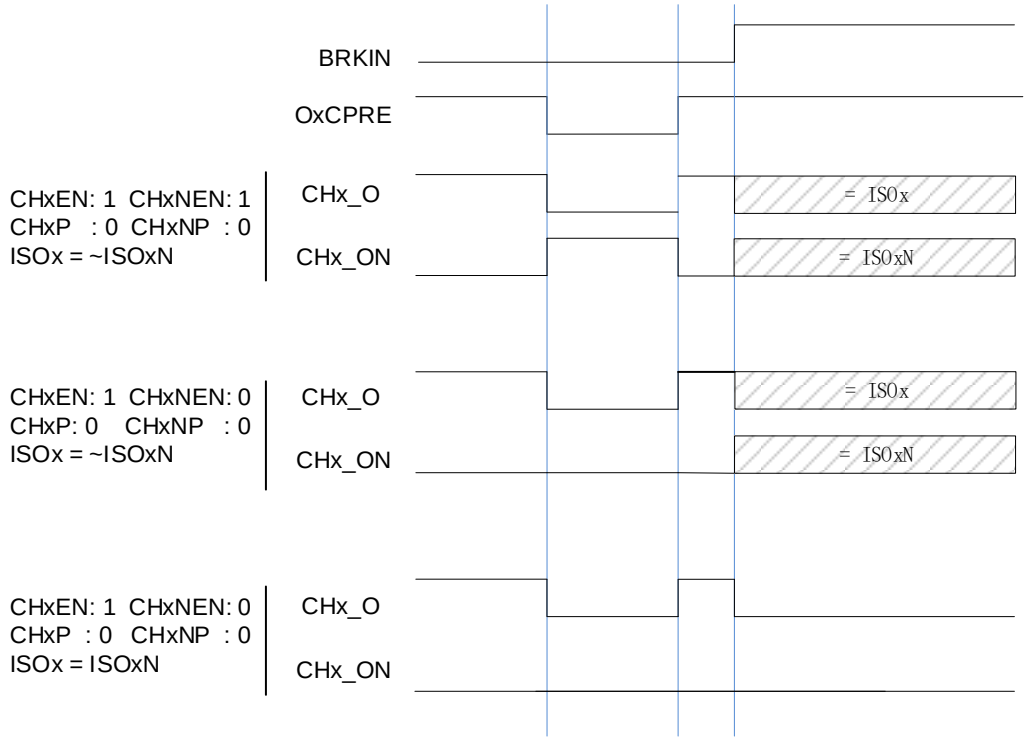
中止模式

使用中止模式时，输出 CHx_O 和 CHx_ON 的信号电平被以下位控制，TIMERx_CCHP 寄存器的 POEN，IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。任何情况下，CHx_O 和 CHx_ON 信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚，也可以选择 HXTAL 时钟失效事件，时钟失效事件由 RCU 中的时钟监视器（CKM）产生。将 TIMERx_CCHP 寄存器的 BRKEN 位置 1 可以使能中止功能。TIMERx_CCHP 寄存器的 BRKP 位决定了中止输入极性。

发生中止时，POEN 位被异步清除，一旦 POEN 位为 0，CHx_O 和 CHx_ON 的输出电平由 TIMERx_CTL1 寄存器中的 ISOx 位和 ISOxN 位决定。如果 IOS=0，定时器释放输出使能，否则输出使能仍然为高。起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由 ISOx 和 ISOxN 位配置。

发生中止时，TIMERx_INTF 寄存器的 BRKIF 位被置 1。如果 BRKIE=1，中断产生。

图 15-19. 通道响应中止输入（高电平有效）时，输出信号的行为



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。通过设置 TSCFGy[3:0] != 4'b0000（y=0,1,2）来选择是仅由 CI0，仅由 CI1，或者由 CI0 和 CI1 来决定定时器的计数方向。在每个方向选择源的电平改变期间，DIR 位会发生改变的。计数器计数方向改变的机制如*表 15-3. 不同编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 15-3. 不同编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
只有CI0	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
只有CI1	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
CI0和CI1	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 15-20.在编码器模式 2 且 CI0FE0 极性不反相时计数器行为

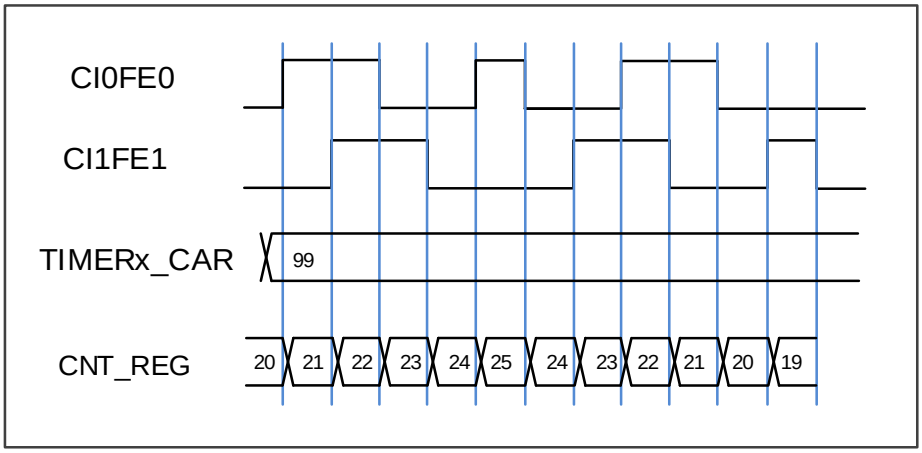
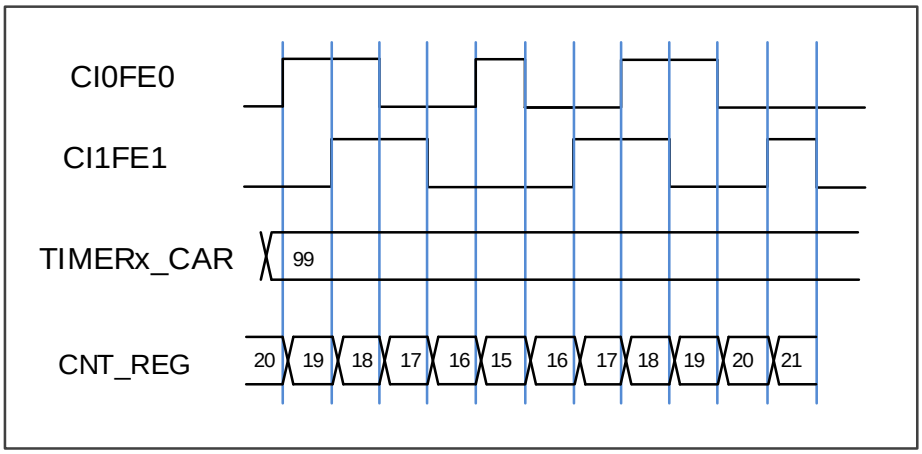


图 15-21.在编码器模式 2 且 CI0FE0 极性反相时计数器行为



霍尔传感器接口功能

高级定时器支持霍尔传感器接口功能，该功能可以用来控制 BLDC 电机。

*图 15-22. 霍尔传感器用在 BLDC 电机控制中*是定时器和电机的连接示意图。众所周知，我们要两个定时器。TIMER_in 定时器（可以是高级定时器或者通用 L0 定时器）接收霍尔传感器的三路信号。

三个霍尔传感器信号与 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。

通过定时器内部连接功能（TRGO-ITIx），TIMER_in 定时器和 TIMER_out 定时器可以连接在一起。TIMER_out 定时器根据 ITIx 触发信号输出 PWM 波，驱动 BLDC 电机，控制 BLDC 电机的速度。这样，TIMER_in 定时器和 TIMER_out 定时器的连接形成了一个反馈电路，可以根据需求改变配置。

高级定时器和通用 L0 定时器具有输入异或功能，可作为 TIMER_in 定时器。同时，高级定时器具备互补输出和死区插入功能，可作为 TIMER_out 定时器。

另外，根据定时器的内部互连关系，可以选择成对的互连定时器，例如：

TIMER_in（TIMER1）-> TIMER_out（TIMER0 ITI1）

选择好合适的互连定时器，线路也已经连接好，就可以配置定时器。有以下关键配置：

- 通过设置TI0S，来使能异或功能。三路输入信号的任何一路发生变化，CI0都会反转，CH0VAL此时会捕获计数器的当前值。
- 通过设置CCUC和CCSE，来选择ITIx触发换相。
- 根据需求配置PWM参数。

图 15-22. 霍尔传感器用在 BLDC 电机控制中

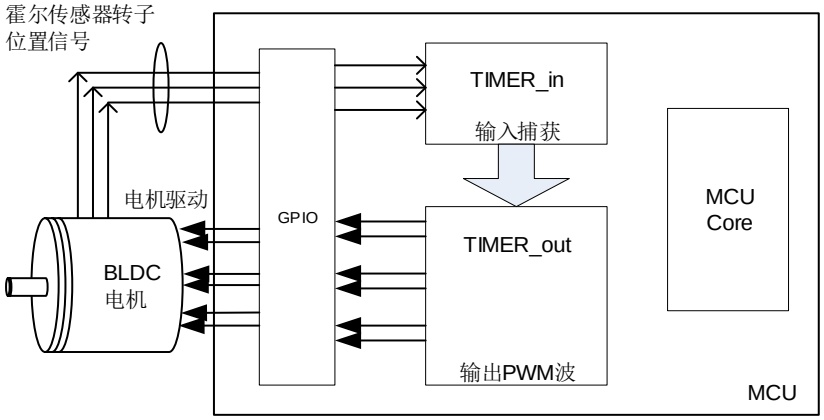
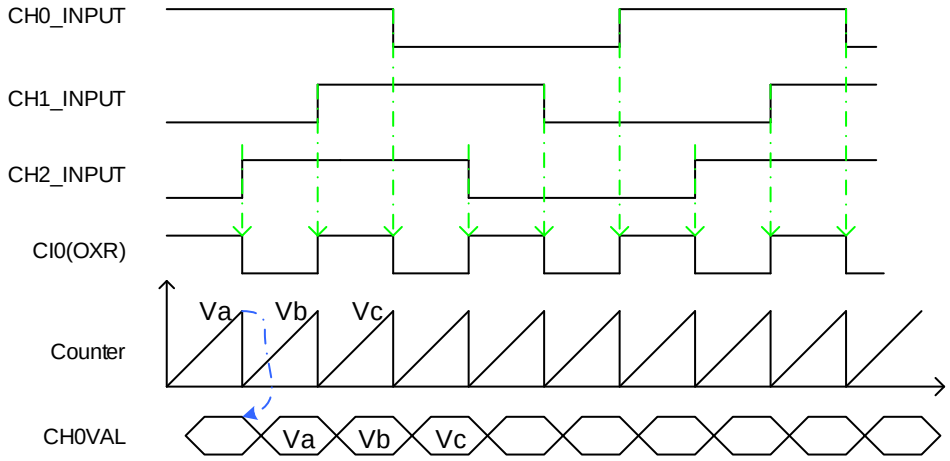
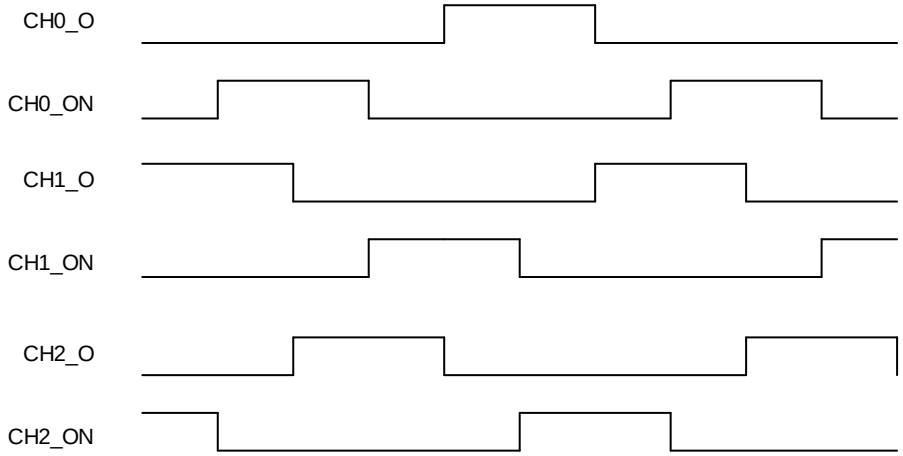


图 15-23. 两个定时器之间的霍尔传感器时序图

高级/通用 L0定时器 TIMER_in 工作在输入捕获模式



高级定时器 TIMER_out 工作在输出比较模式(带有死区的PWM)

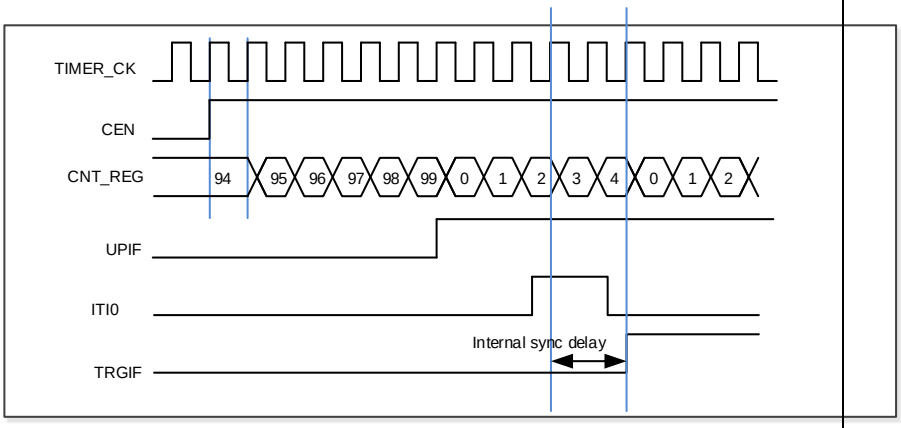
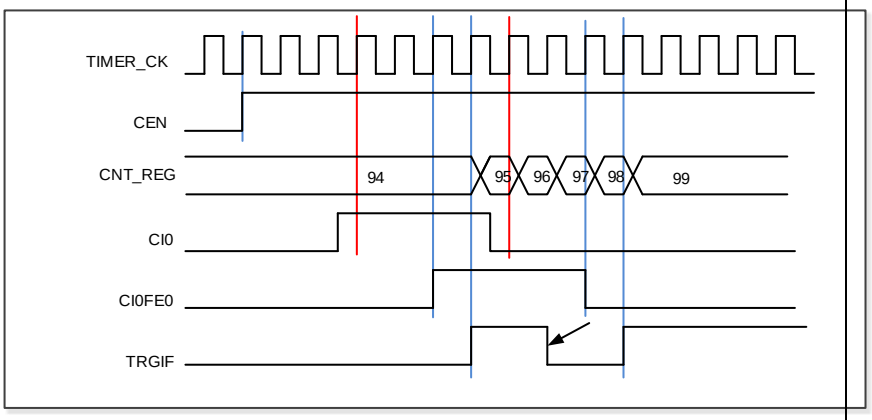


主-从管理

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 SYSCFG_TIMER0CFG（x=3,4,5）寄存器中的 TSCFGy[3:0]配置这些模式。

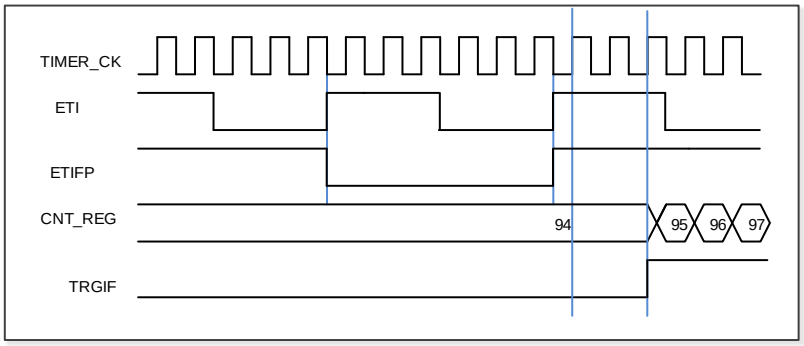
表 15-4. 从模式示例

	模式选择	触发源选择	极性选择	滤波和预分频
列举	TSCFGy[3:0] y=3（复位模式） y=4（暂停模式） y=5（事件模式）	TSCFGy[3:0] 0001: ITI0 0010: ITI1 0011: ITI2 0100: ITI3 0101: CI0F_ED 0110: CI0FE0	如果触发源是CI0FE0或者CI1FE1，配置CHxP和CHxNP来选择极性和反相。 如果触发源是ETIFP，配置ETP选择极性和反相。	若触发源为ITIx，滤波和预分频不可用。 若触发源为CIx，可配置CHxCAPFLT设置滤波，预分频不可用。 若触发源为ETIFP，滤波和预分频均可用。

	模式选择	触发源选择	极性选择	滤波和预分频
		0111: CI1FE1 1000: ETIFP		
例1	**复位模式** 当触发输入上升沿到来时，计数器清零重启。	TSCFG3[3:0] = 4'b 0001选择ITI0为触发源。	若触发源是ITI0，极性选择不可用。	若触发源是ITI0，滤波和预分频不可用。
	图 15-24. 复位模式 			
例2	**暂停模式** 当触发输入为低的时候，计数器暂停计数，当触发输入为高时，计数器计数。	TSCFG4[3:0] = 4'b 0110 选择CI0FE0为触发源。	TI0S=0（非异或）[CH0NP=0，CH0P=0] CI0FE0不反相。捕获发生在上升沿 。	在这个例子中滤波被旁路。
	图 15-25. 暂停模式 			
例3	**事件模式** 触发输入的上升沿计数器开始计数。	TSCFG5[3:0] = 4'b 1000 选择ETIFP为触发源。	ETP = 0，ETI极性不改变。	ETPSC = 1，ETI 2分频。 ETFC = 0，ETI 无滤波。

	模式选择	触发源选择	极性选择	滤波和预分频

图 15-26. 事件模式



单脉冲模式

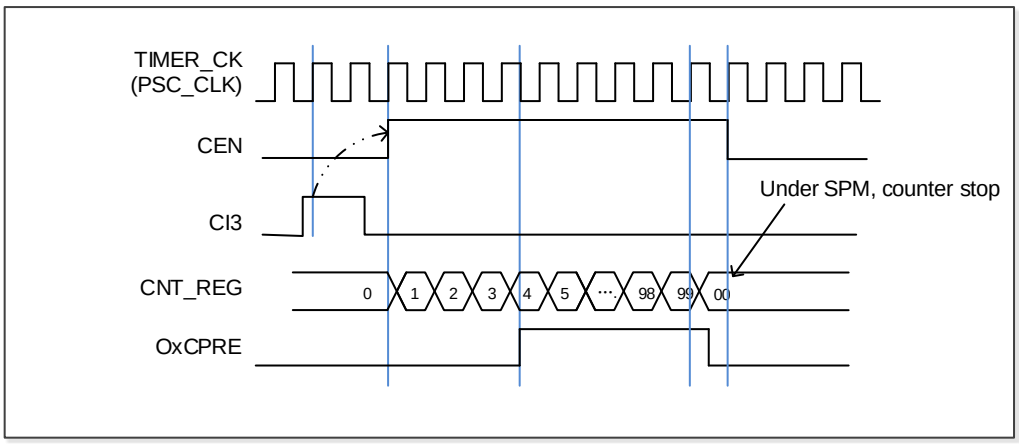
设置 TIMERx_CTL0 寄存器的 SPM 位置 1，使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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 PWM 模式 0 或 PWM 模式 1 时 CHxCOMFEN 位才可用，触发源来源于触发信号。

图 15-27. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 展示了一个例子。

图 15-27. 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。

其他定时器互连的例子：

- 定时器2作为定时器0的预分频器

连接配置定时器 2 为定时器 0 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. 配置定时器 0 在外部时钟模式 0，选择定时器 0 输入触发源为定时器 2，（配置 SYSCFG_TIMERxCFG 寄存器的 TSCFG6[3:0] = 4'b 0001）；

4. 写 1 到 CEN 位启动定时器 0（TIMER0_CTL0 寄存器）；写 1 到 CEN 位启动定时器 2（TIMER2_CTL0 寄存器）。

- 用定时器2的使能/更新信号来启动定时器0

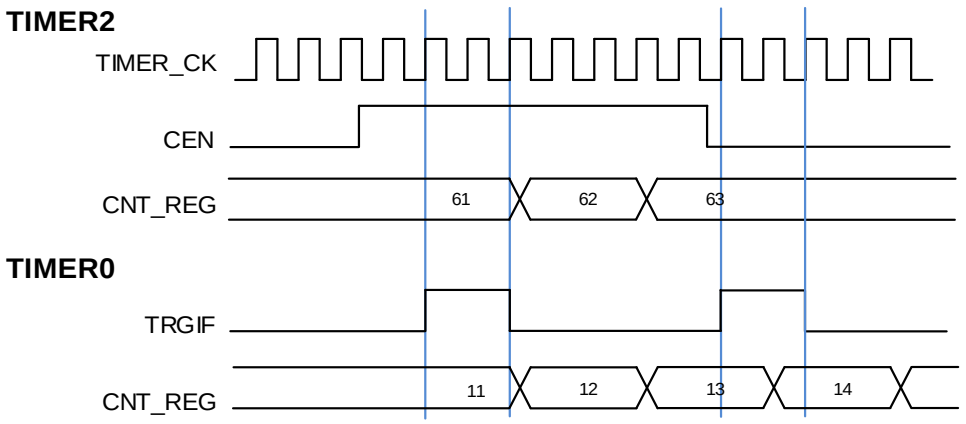
用定时器 2 的使能信号来启动定时器 0，见*图 15-28. 用定时器 2 的使能信号触发定时器 0*。在定时器 2 使能信号输出后，定时器 0 按照分频后的内部时钟从当前值开始计数。

当定时器 0 接收到触发信号，它的 CEN 位置 1，计数器计数直到禁能定时器 0。两个定时器的计数器频率都是 TIMER_CK 经过预分频器 3 分频后的频率（$f_{PSC\_CLK} = f_{TIMER\_CK}/3$）。步骤如下：

1.配置定时器 2 为主模式，发送它的使能信号作为触发输出（配置 TIMER2_CTL1 寄存器的 MMC=3'b001），配置定时器 0 选择输入触发来自定时器 2（配置 SYSCFG_TIMERxCFG 寄存器的 TSCFG5[3:0] = 4'b 0011）；

2. 写 1 到 CEN 来开启定时器 2（TIMER2_CTL0 寄存器）。

图 15-28. 用定时器 2 的使能信号触发定时器 0



在这个例子中，也可以使用更新事件代替使能信号作为触发源。见*图 15-29. 用定时器 2 的更*

新事件来触发定时器 0，按以下步骤进行：

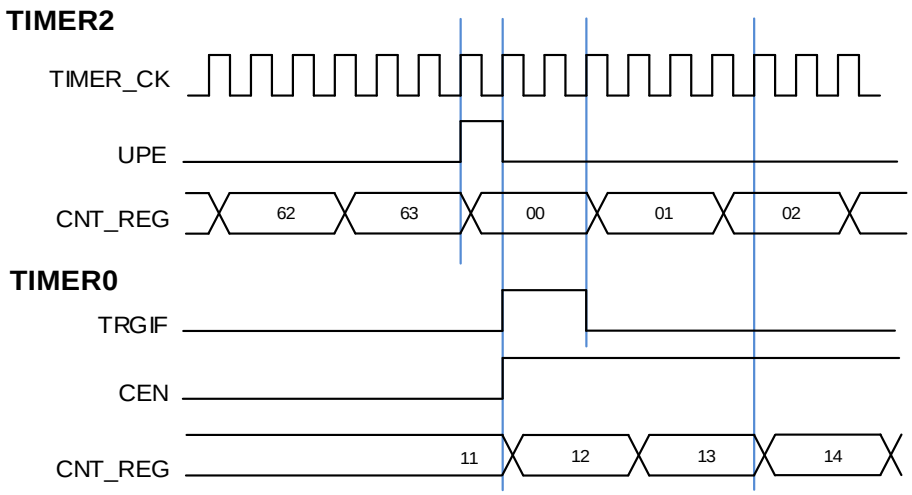
1. 配置定时器 2 为主模式，发送它的更新事件（UPE）作为触发输出（配置 TIMER2_CTL1 寄存器的 MMC=3'b010）；

2. 配置定时器 2 的周期（TIMER2_CARL 寄存器）；

3. 配置定时器 0 选择输入触发来自定时器 2，配置定时器 0 在事件模式（配置 SYSCFG_TIMERxCFG 寄存器的 TSCFG5[3:0] =4'b0011）；

4. 写 1 到 CEN 来开启定时器 2（TIMER2_CTL0 寄存器）。

图 15-29. 用定时器 2 的更新事件来触发定时器 0



■ 使用定时器2的使能/O0CPRE信号来使能定时器0计数。

在这个例子中，使用定时器 2 的使能信号来使能定时器 0。如*图 15-30. 用定时器 2 的使能信号来控制定时器 0*，在定时器 2 被使能后，定时器 0 在内部分频的时钟上开始计数。两个计数器的时钟频率都是由 TIMER_CK 时钟 3 分频得来（$f_{PSC\_CLK} = f_{TIMER\_CK}/3$），步骤如下：

1. 配置定时器 2 在主模式，配置其输出使能信号作为触发输出（配置 TIMER2_CTL1 寄存器的 MMC=3'b001）；

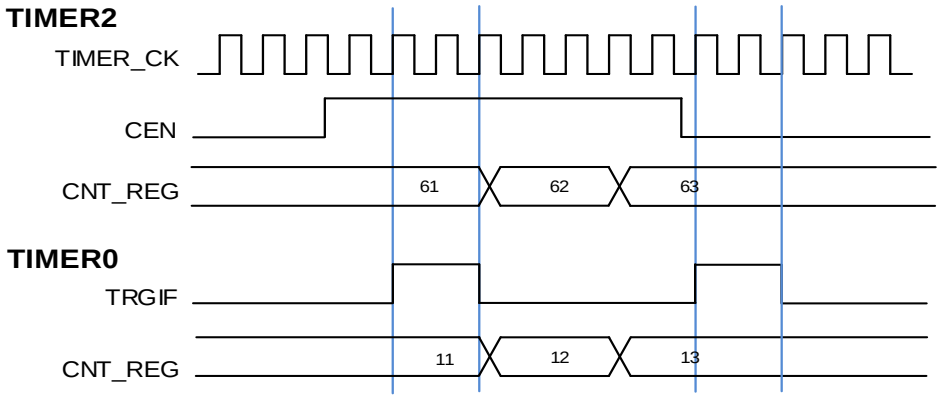
2. 配置定时器 0 从定时器 2 获取输入触发，配置定时器 0 工作在暂停模式（配置 SYSCFG_TIMERxCFG 寄存器的 TSCFG5[3:0] = 4'b 0011）；

3. 写 1 到 CEN 位来使能定时器 0（TIMER0_CTL0 寄存器）；

4. 写 1 到 CEN 位来启动定时器 2（TIMER0_CTL0 寄存器）；

5. 写 0 到 CEN 位来停止定时器 2（TIMER0_CTL0 寄存器）。

图 15-30. 用定时器 2 的使能信号来控制定时器 0 的暂停模式



这个例子中，我们也可以使用定时器 2 的 O0CPRE 信号代替其使能信号输出作为触发源。步骤如下：

1. 配置定时器 2 在主模式下，配置 O0CPRE 信号为触发输出（配置 TIMER2_CTL1 寄存器的 MMS=3'b100）；

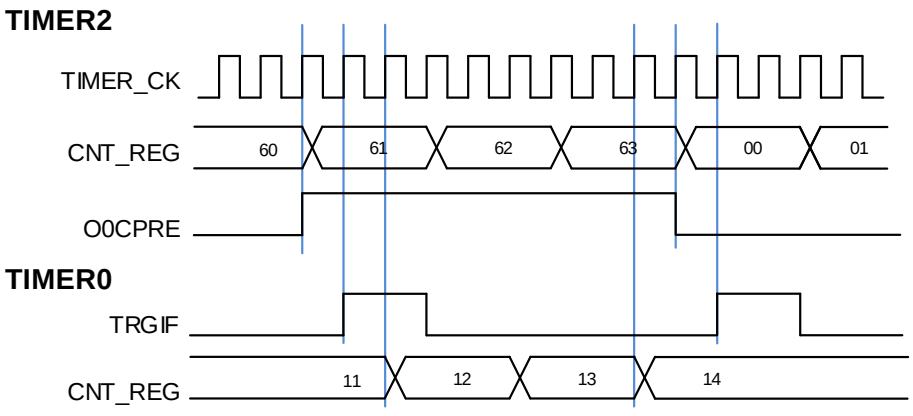
2. 配置定时器 2 的 O0CPRE 波形（TIMER2_CHCTL0 寄存器）；

3. 配置定时器 0 获取来自定时器 2 的输入触发，配置定时器 0 工作在暂停模式（配置 SYSCFG_TIMERxCFG 寄存器 TSCFG5[3:0] = 4'b 0011）；

4. 写 1 到 CEN 位来使能定时器 0（TIMER0_CTL0 寄存器）；

5. 写 1 到 CEN 位来开启定时器 2（TIMER0_CTL0 寄存器）。

图 15-31. 用定时器 2 的 O0CPRE 信号控制定时器 0 的暂停模式



■ 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 0 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 工作在事件模式，并选择 CI0_ED 作为触发输入（TSCFG5[3:0] = 4b'0101 in

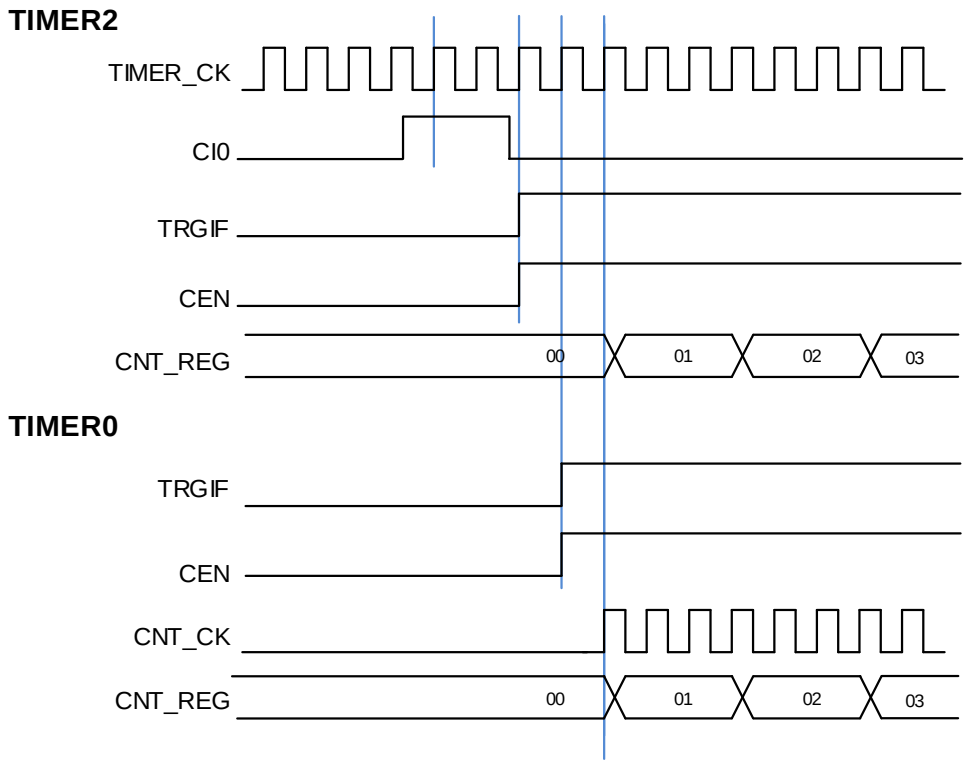
the_SYSCFG_TIMERxCFG register）；

2. 写 MSM=1（TIMER2_SMCFG 寄存器）来配置定时器 2 工作在主/从模式；

3. 配置定时器 0 工作在事件模式，定时器 0 的触发输入为定时器 2（TSCFG5[3:0] = 4b'0011 in the_SYSCFG_TIMERxCFG register）；

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 15-32. 用定时器 2 的 CI0 信号来触发定时器 0 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 RISCV 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。

15.1.5. TIMERx 寄存器（x=0）

TIMER0 基地址：0x4001 0000

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编码器模式时，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、暂停模式和编码器模式才能工作。

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	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISO0N	ISO0	TI0S	MMC[2:0]			DMAS	CCUC	保留	CCSE
	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	ISO3	通道 3 的空闲状态输出 参考 ISO0 位
13	ISO2N	通道 2 的互补通道空闲状态输出 参考 ISO0N 位
12	ISO2	通道 2 的空闲状态输出 参考 ISO0 位
11	ISO1N	通道 1 的互补通道空闲状态输出 参考 ISO0N 位
10	ISO1	通道 1 的空闲状态输出 参考 ISO0 位
9	ISO0N	通道 0 的互补通道空闲状态输出 0：当 POEN 位复位时，CH0_ON 设置低电平 1：当 POEN 位复位时，CH0_ON 设置高电平 此位只有在 TIMERx_CCHP 寄存器的 PROT[1:0]位为 00 的时候可以被更改。
8	ISO0	通道 0 的空闲状态输出 0：当 POEN 位复位时，CH0_O 设置低电平 1：当 POEN 位复位时，CH0_O 设置高电平 如果 CH0_ON 生效，一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP 寄存器的 PROT[1:0]位为 00 的时候可以被更改。
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，TIMERx_CH1 和 TIMERx_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	CCUC	换相控制影子寄存器更新控制 当换相控制影子寄存器（CHxEN，CHxNEN 和 CHxCOMCTL 位）使能（CCSE=1），这些影子寄存器更新控制如下： 0：CMTG 位被置 1 时，更新影子寄存器 1：当 CMTG 位被置 1 或检测到 TRIGI 上升沿时，影子寄存器更新 当通道没有互补输出时，此位无效。
1	保留	必须保持复位值。
0	CCSE	换相控制影子使能 0：影子寄存器（CHxEN，CHxNEN 和 CHxCOMCTL 位）禁能 1：影子寄存器（CHxEN，CHxNEN 和 CHxCOMCTL 位）使能 如果这些位已经被写入了，换相事件到来时这些位才被更新。 当通道没有互补输出时，此位无效。

从模式配置寄存器（TIMERx_SMCFG）

地址偏移：0x08

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
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	保留						
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 TSCFGy[3:0](y=3,4,5)不能为 4’b0111。 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 SYSCFG_TIMERxCFG 寄存器。

13:12 ETPSC[1:0] 外部触发预分频

外部触发信号 ETI 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 ETIP 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_{CK_TIMER}
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。

6:0 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 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	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw	rw	rw	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	CMTDEN	换相 DMA 更新请求使能 0：禁止换相 DMA 更新请求 1：使能换相 DMA 更新请求
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	BRKIE	中止中断使能 0：禁止中止中断 1：使能中止中断
6	TRGIE	触发中断使能 0：禁止触发中断 1：使能触发中断
5	CMTIE	换相更新中断使能 0：禁止换相更新中断

		1：使能换相更新中断
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	BRKIF	TRGIF	CMTIF	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	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 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	保留	必须保持复位值。
7	BRKIF	中止中断标志位 当中止输入有效时，由硬件对该位置‘1’。 当中止输入无效时，则该位可由软件清‘0’。 0：无中止事件产生 1：中止输入上检测到有效电平
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	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时，此标志位被硬件置 1，此位由软件清 0。 0：无通道换相更新中断发生 1：通道换相更新中断发生
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。 当通道 0 在输入模式下时，读 TIMERx_CH0CV 会将此标志清零。 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								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
								w	w	w	w	w	w	w	w

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	BRKG	产生中止事件 该位由软件置 1，用于产生一个中止事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRKIF 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：不产生中止事件 1：产生中止事件
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	CMTG	通道换相更新事件发生 此位由软件置 1，由硬件自动清 0。当此位被置 1，根据 CCSE 位（TIMERx_CTL1 寄存器中）的值，通道捕获/比较控制寄存器（CHxEN，CHxNEN 和 CHxCOMCTL）的互补输出被更新。 0：不产生通道换相更新事件 1：产生通道换相更新事件
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，计数器的当前值被捕获到 TIMERx_CH0CV 寄存器，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 TSCFG7[3:0] 位域（位于 SYSCFG_TIMERxCFG（x=0）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 TSCFG7[3:0] 位域（位于 SYSCFG_TIMERxCFG（x=0）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 TSCFG7[3:0] 位域（位于 SYSCFG_TIMERxCFG（x=0）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 TSCFG7[3:0] 位域（位于 SYSCFG_TIMERxCFG（x=0）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															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	保留	CH3P	CH3EN	CH2NP	CH2NEN	CH2P	CH2EN	CH1NP	CH1NEN	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN
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	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	CH2NEN	通道 2 互补输出使能 参考 CH0NEN 描述
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	CH1NEN	通道 1 互补输出使能 参考 CH0NEN 描述
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互补输出高电平为有效电平 1：通道0互补输出低电平为有效电平 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	CH0NEN	通道 0 互补输出使能 当通道 0 配置为输出模式时，将此位置 1 使能通道 0 的互补输出。 0：禁止通道 0 互补输出 1：使能通道 0 互补输出
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。

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。

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

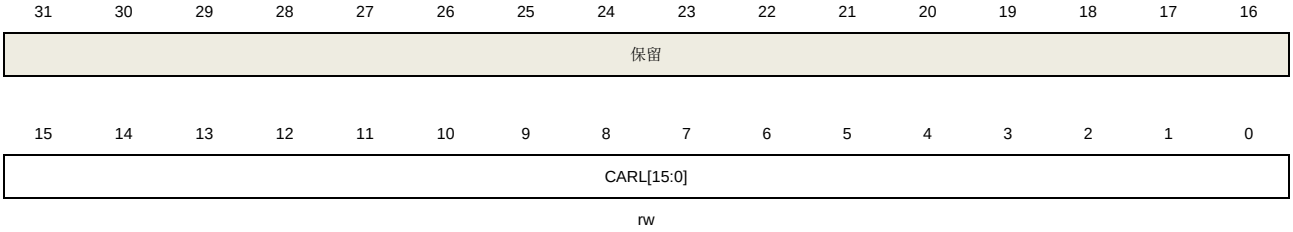
计数器时钟等于 TIMER_CK 时钟除以（PSC+1），每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



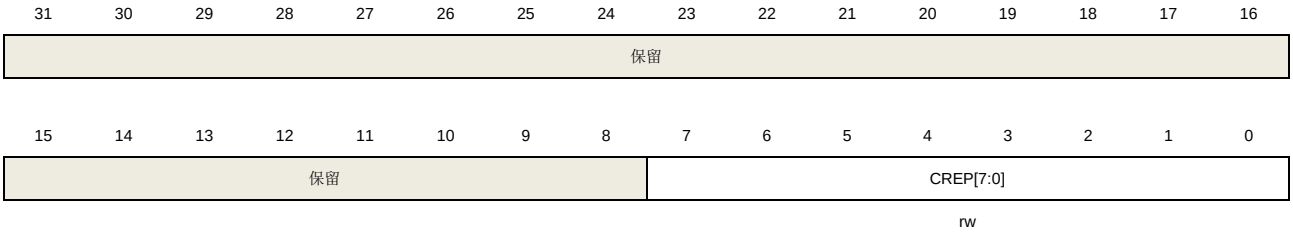
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

重复计数寄存器（TIMERx_CREP）

地址偏移：0x30

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



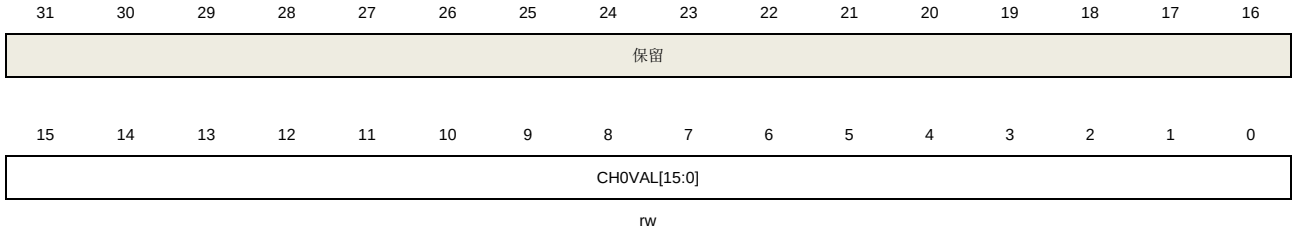
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响（前提是影子寄存器被使能）。

通道 0 捕获/比较寄存器（TIMERx_CH0CV）

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



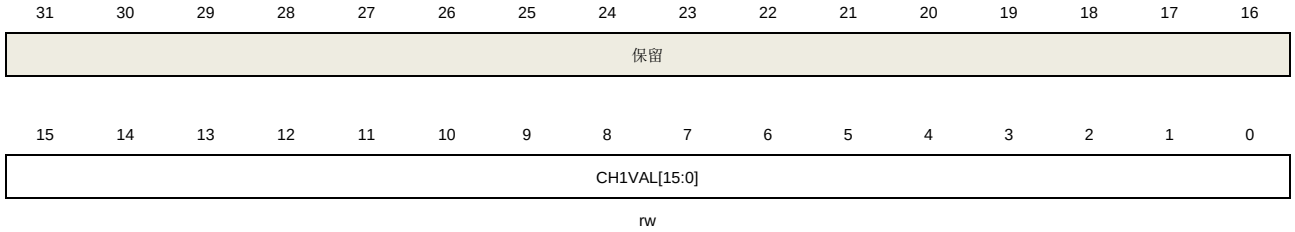
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值，并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



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。

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
CH2VAL[15:0]															
rw															

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。

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
CH3VAL[15:0]															
rw															

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。

互补通道保护寄存器（TIMERx_CCHP）

地址偏移：0x44

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
保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
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	POEN	所有的通道输出使能 该位通过以下方式置 1： –写 1 置位 –如果 OAEN=1，则在下一次更新事件发生时置 1. 该位通过以下方式清 0： –写 0 清 0 –有效的中止输入（异步） 如果一个通道配置为输出模式，如果设置了相应的使能位（TIMERx_CHCTL2 寄存器的 CHxEN, CHxNEN 位），则开启 CHx_O 和 CHx_ON 输出。 0：禁止通道输出 1：使能通道输出 注意：仅当 CHxMS[1:0]=2’b00 时该位有效。
14	OAEN	自动输出使能 0：POEN 位只能使用软件方式置 1。 1：如果中止输入无效，下一次更新事件发生时，POEN 位将会置 1。 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
13	BRKP	中止极性 此位定义了中止输入信号 BRKIN 的极性。 0：中止输入低电平有效。 1：中止输入高电平有效。
12	BRKEN	中止使能 此位置 1 使能中止事件和 CCS 时钟失败事件输入。 0：禁能中止输入。 1：使能中止输入。 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
11	ROS	运行模式下“关闭状态”使能 当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见*表 15-2. 由参数控制的互补输出表*。 0：输出“关闭状态”禁能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“禁能状态”。

1：输出“关闭状态”使能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

10 IOS

空闲模式下“关闭状态”使能
当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见*表 15-2. 由参数控制的互补输出表*。
0：输出“关闭状态”禁能。当 CHxEN 和 CHxNEN 位均被清零，对应通道为输出“禁能状态”。
1：输出“关闭状态”使能。不论 CHxEN 和 CHxNEN 位的值，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

9:8 PROT[1:0]

互补寄存器保护控制
这两位定义了寄存器的写保护特性。
00：禁能保护模式。无写保护。
01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护。
10：PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx_CHCTL2 寄存器中 CHxP/CHxNP 位（如果相应通道配置为输出模式），TIMERx_CCHP 寄存器中 ROS/IOS 位。
11：PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTL0/1 中 CHxCOMCTL/ CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。
系统复位后这两位只能被写一次，一旦 TIMERx_CCHP 寄存器被写入，这两位被写保护。

7:0 DTCFG[7:0]

死区时间控制
DTCFG 值和死区时间的关系如下：

DTCFG[7:5]	The duration of dead-time
3’b0xx	DTCFG[7:0] * t_{DTS_CK}
3’b10x	（64+ DTCFG[5:0]）* t_{DTS_CK} *2
3’b110	（32+ DTCFG[4:0]）* t_{DTS_CK} *8
3’b111	（32+ DTCFG[4:0]）* t_{DTS_CK} *16

注意：
1. t_{DTS_CK} 是 DTS_CK 的周期，由 TIMERx_CTL0 中的 CKDIC[1:0]定义。
2. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

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。

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			DMATC[4:0]					保留			DMATA [4:0]				
			rw								rw				

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。当第一次访问 TIMERx_DMATB 寄存器时，实际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。

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
DMATB[15:0]															
rw															

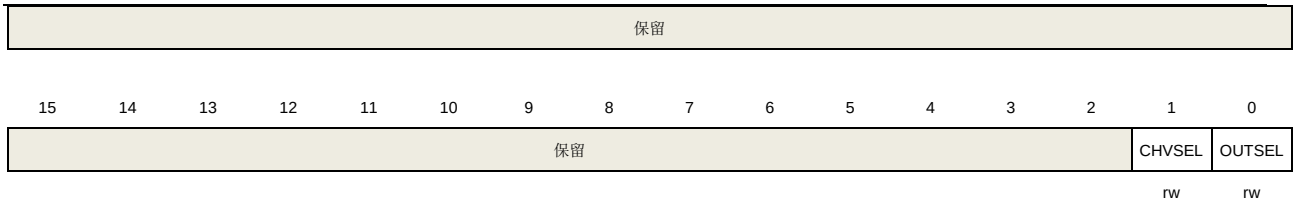
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	DMATB[15:0]	DMA 发送缓冲 对这个寄存器的读或写，从（起始地址）到（起始地址+传输次数*4）地址范围内的寄存器会被访问。传输次数由硬件计算，范围为 0 到 DMATC。

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



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	OUTSEL	输出值选择位 此位由软件写 1 或清 0。 1：如果 POEN 位与 IOS 位均为 0，则输出禁能。 0：无影响。

15.2. 通用定时器 L0（TIMERx, x=1，2）

15.2.1. 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 32 位（TIMER1/2）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。

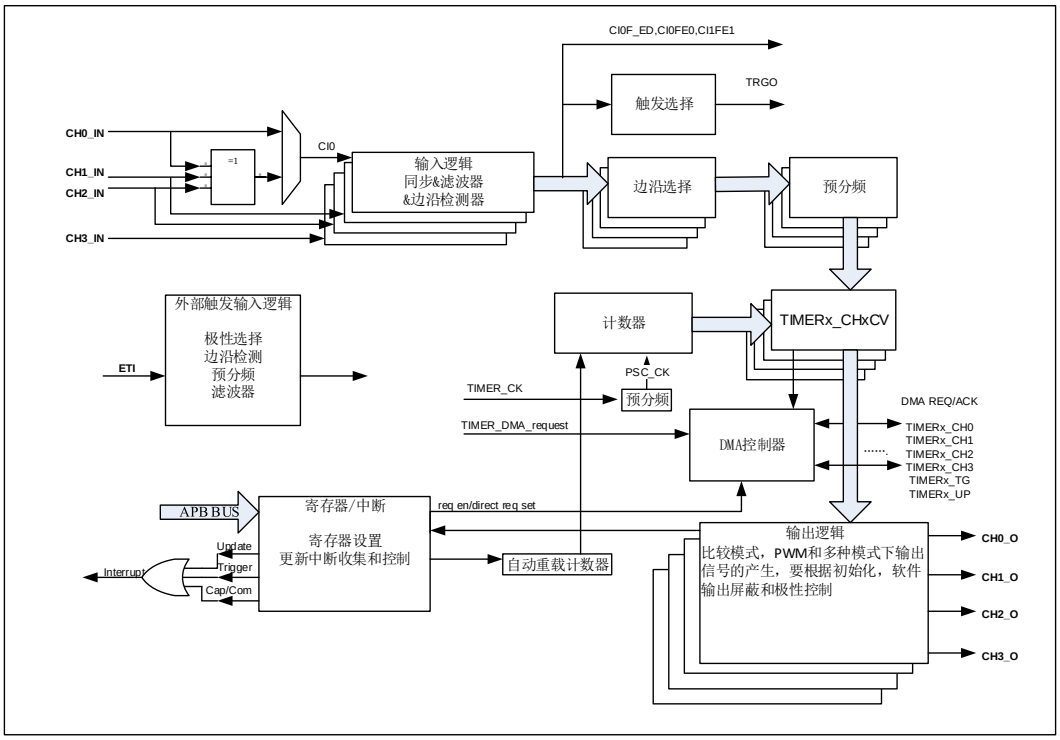
15.2.2. 主要特征

- 总通道数：4；
- 计数器宽度：32位（TIMER1/2）；
- 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；
- 正交编码器接口：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。

15.2.3. 结构框图

*图15-33. 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 15-33. 通用定时器 L0 结构框图



15.2.4. 功能描述

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 TSCFGy[3:0]位确定的时钟源，TSCFGy[3:0]位于 SYSCFG_TIMER0CFG，（y=0,1…6）。

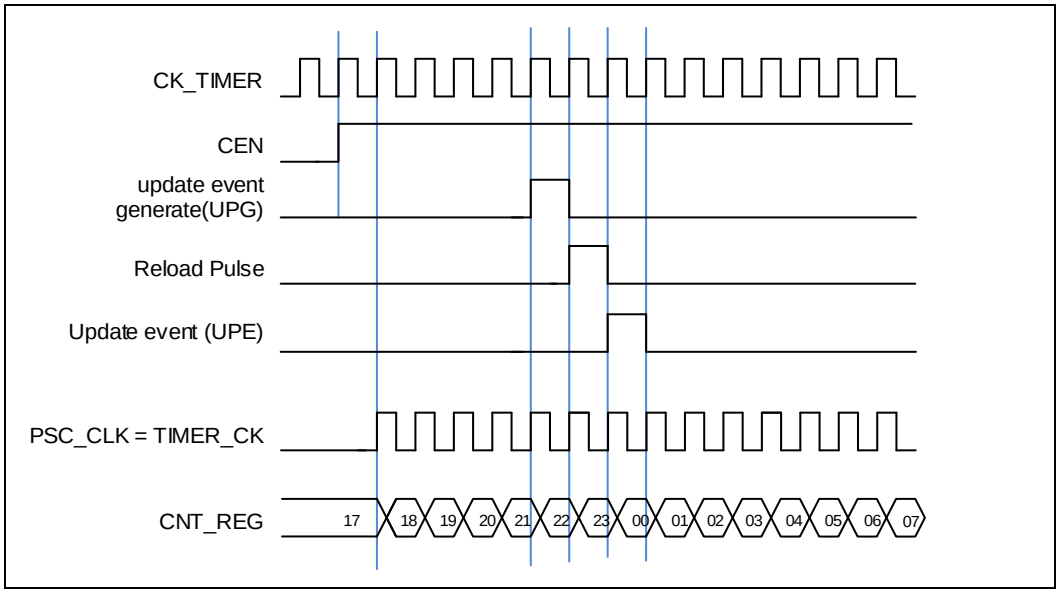
- TSCFGy[3:0] =4’b0000，TSCFGy[3:0]位于SYSCFG_TIMER0CFG，（y=0,1…6），定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

当 TSCFGy[3:0] =4’b0000，TSCFGy[3:0]位于 SYSCFG_TIMER0CFG，（y=0,1…6），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TSCFGy[3:0] !=4’b0000，TSCFGy[3:0]位于SYSCFG_TIMER0CFG，（y=0,1,2,6），预分频器被其他时钟源（由TSCFG6[3:0]区域选择）驱动，更多细节在下文说明，当TSCFGy[3:0]（y=3,4,5）设置为有效值时，计数器预分频器时钟源由内部时钟TIMER_CK驱动。

图 15-34. 内部时钟分频为 1 时，计数器的时序图



- TSCFG6[3:0] !=4'b0000（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 TSCFG6[3:0]为 0x5，0x6 或 0x7 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 TSCFG6[3:0]为 0x1，0x2，0x3 或者 0x4。

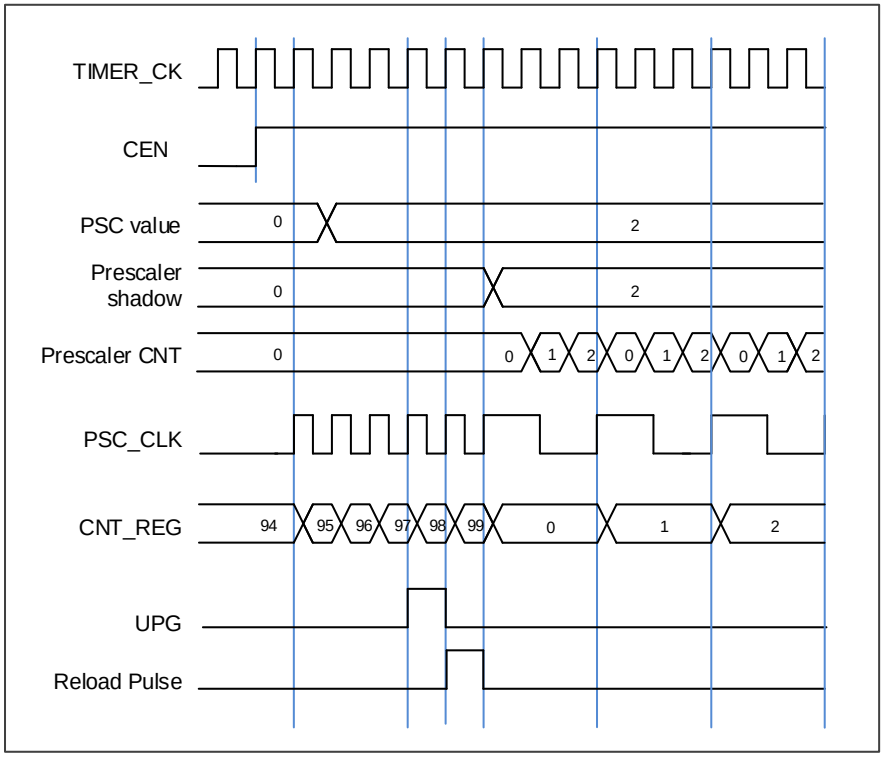
- SMC1=1'b1（外部时钟模式1），定时器选择外部输入引脚ETI作为时钟源。

计数器预分频器可以在外部引脚 ETI 的每个上升沿或下降沿计数。这种模式可以通过设置 TIMERx_SMCFG 寄存器中的 SMC1 位为 1 来选择。另一种选择 ETI 信号作为时钟源方式是，设置 TSCFG6[3:0]为 0x8。注意 ETI 信号是通过数字滤波器采样 ETI 引脚得到的。如果选择 ETI 信号为时钟源，触发控制器包括边沿监测电路将在每个 ETI 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

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 15-35. 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（重复计数寄存器，自动重载寄存器，预分频寄存器）都将被更新。

*图15-36. 向上计数时序图，PSC=0/2*和*图15-37. 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 15-36. 向上计数时序图，PSC=0/2

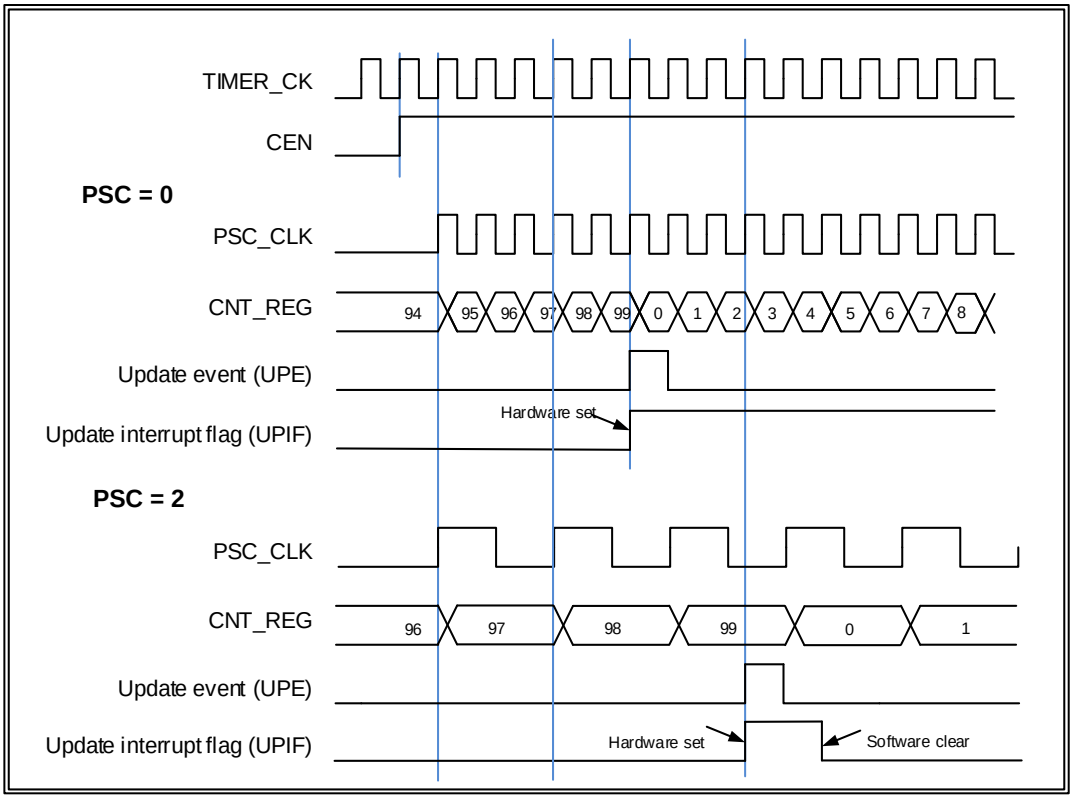
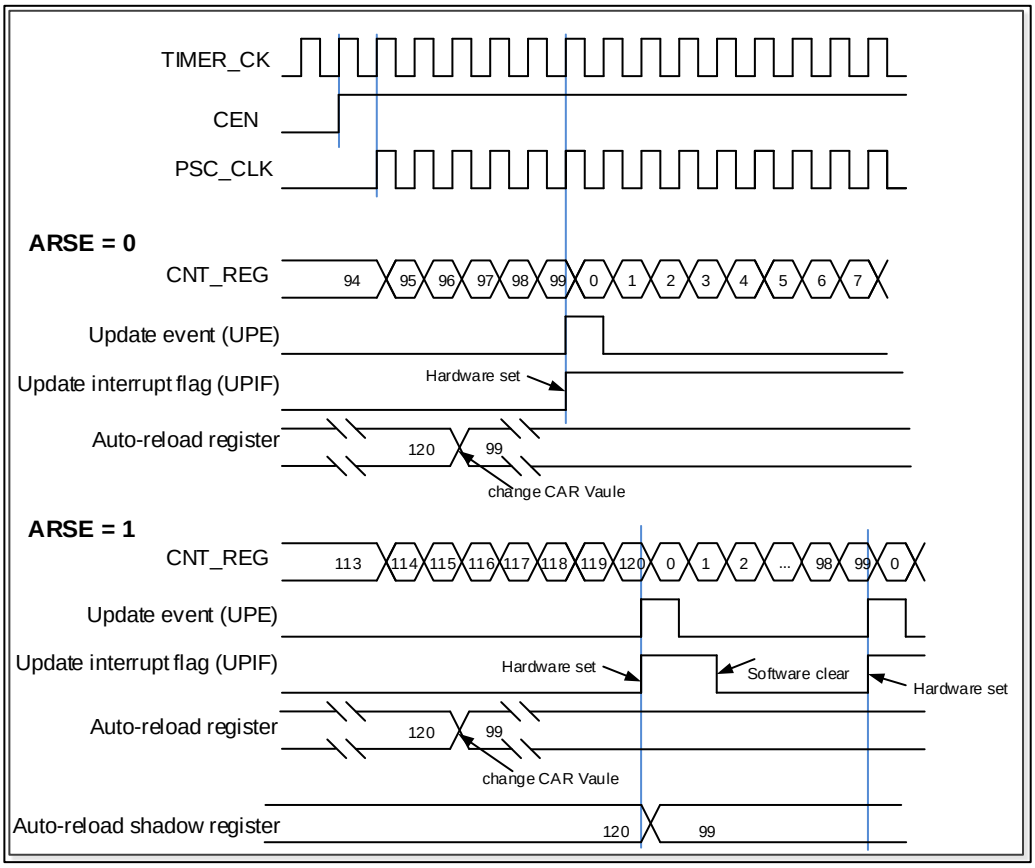


图 15-37. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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（重复计数器，自动重载寄存器，预分频寄存器）都将被更新。

*图15-38. 向下计数时序图，PSC=0/2*和*图15-39. 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 15-38. 向下计数时序图，PSC=0/2

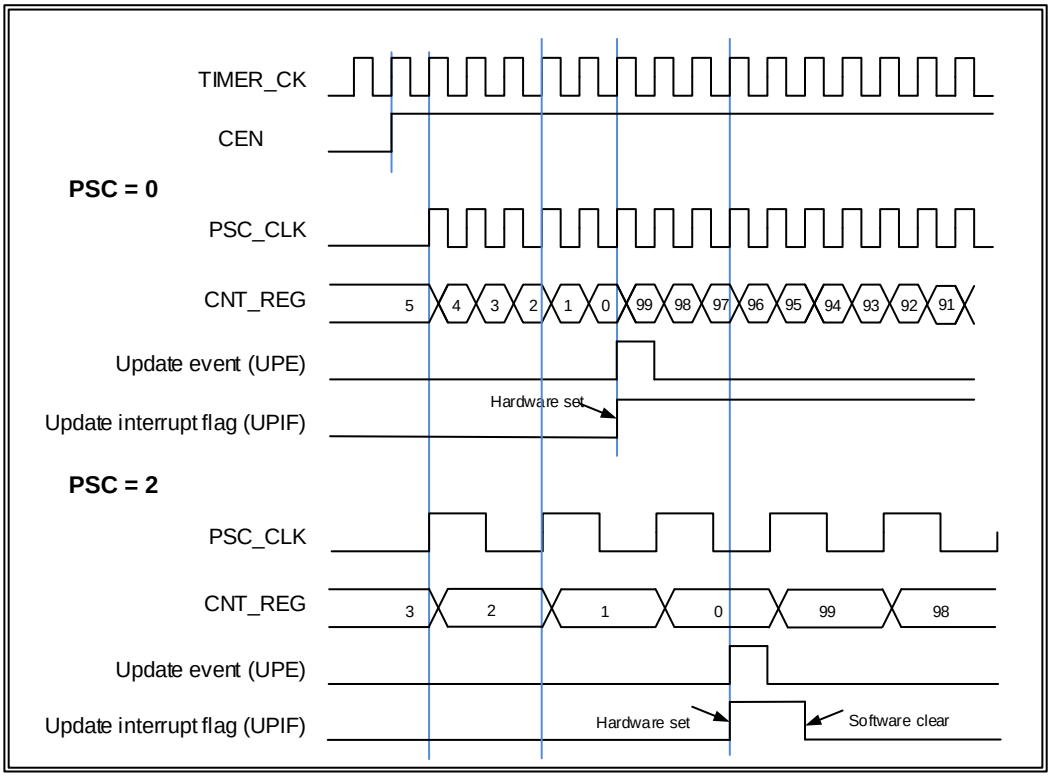
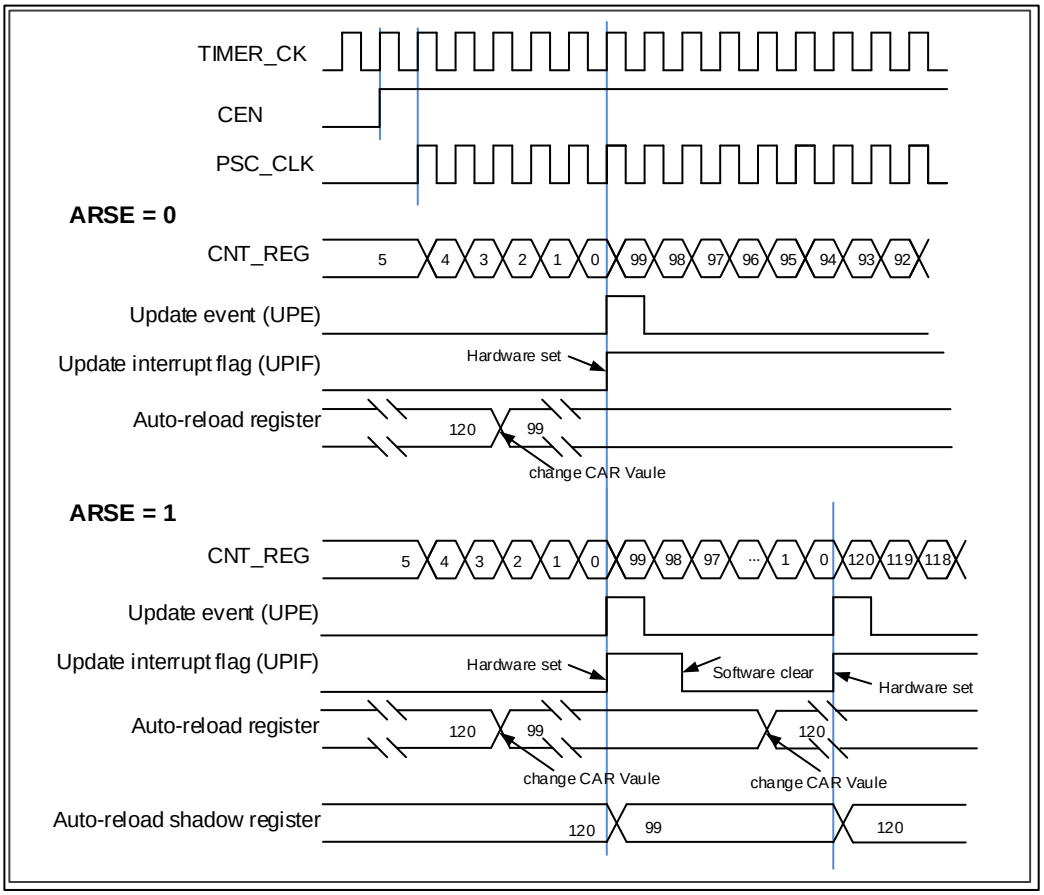


图 15-39. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



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。

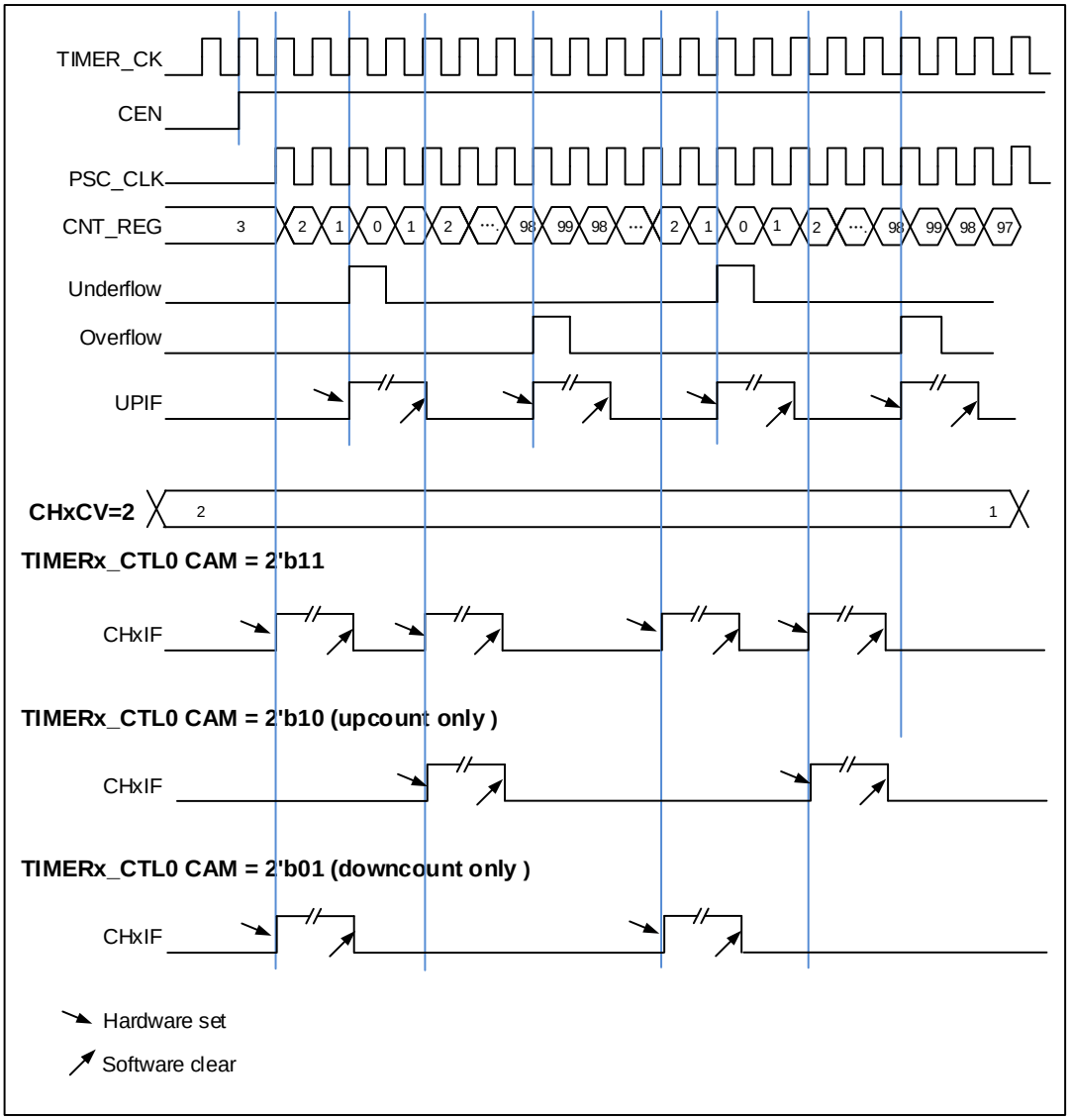
上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1。但是 CHxIF 位是否置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考*图 15-40. 中央计数模式计数器时序图*。

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（自动重载寄存器，预分频寄存器）都将被更新。

*图 15-40. 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 15-40. 中央计数模式计数器时序图



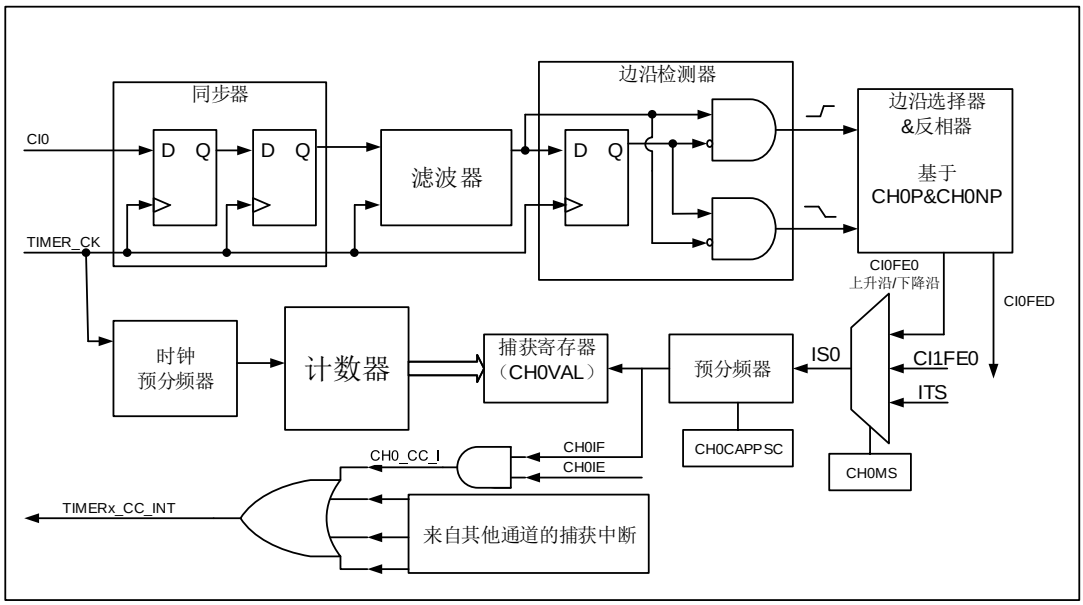
输入捕获和输出比较通道

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 15-41. 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。

通道输入捕获功能也可用来测量 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 15-42. 通道输出比较原理（x=0,1,2,3）

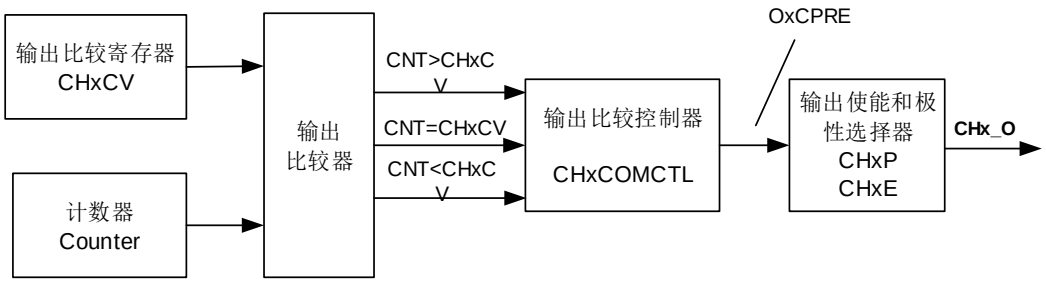


*图15-42. 通道输出比较原理（x=0,1,2,3）*给出了输出比较的原理电路。通道输出信号CHx_O与OxCPRE信号（详情请见*通道输出准备信号*）的关系描述如下：OxCPRE信号高电平有效，CHx_O的输出情况与OxCPRE信号，CHxP位和CHxE位有关（具体情况请见TIMERx_CHCTL2寄存器中的描述）。例如，当设置CHxP=0（CHx_O高电平有效，与OxCPRE输出极性相同）、CHxE=1（CHx_O输出使能）时：

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。

在通道输出比较功能，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。

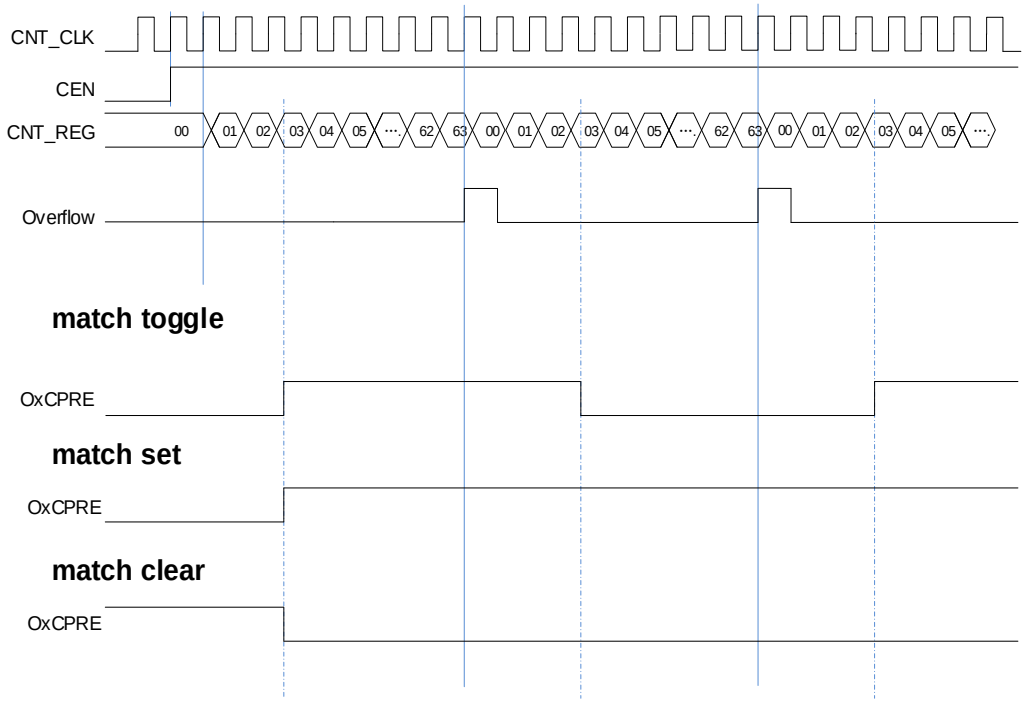
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

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 15-43. 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 15-43. 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 15-44. 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 15-45. 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 15-44. EAPWM 时序图

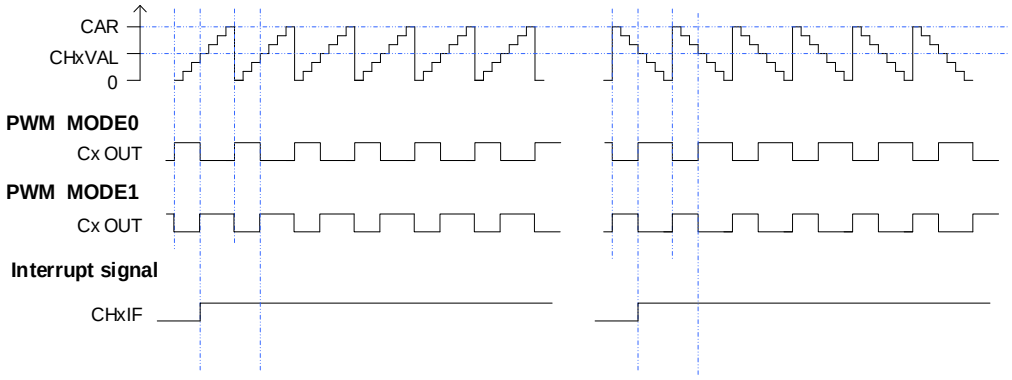
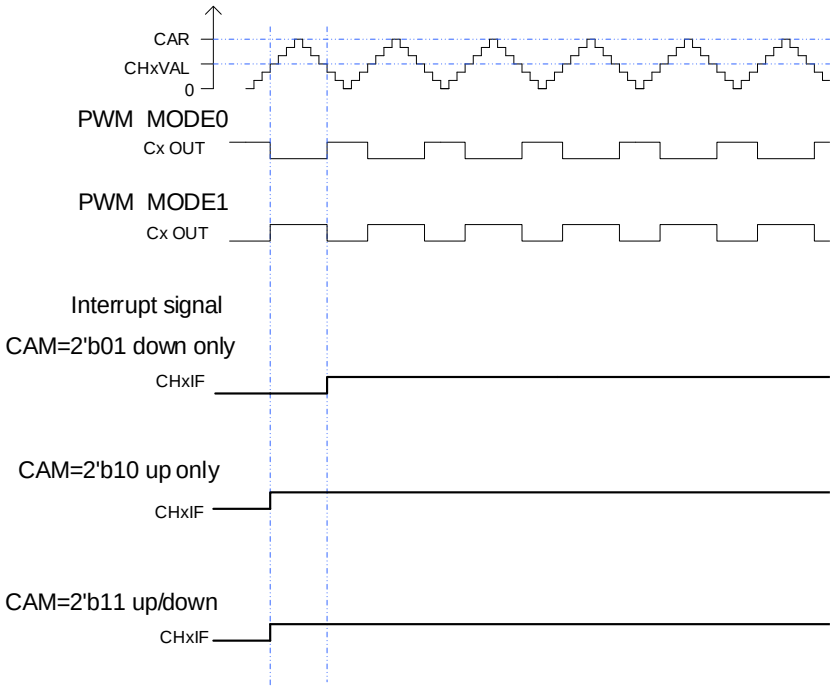


图 15-45. CAPWM 时序图



通道输出准备信号

根据*图15-42. 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

参考*正交译码器*。

霍尔传感器接口功能

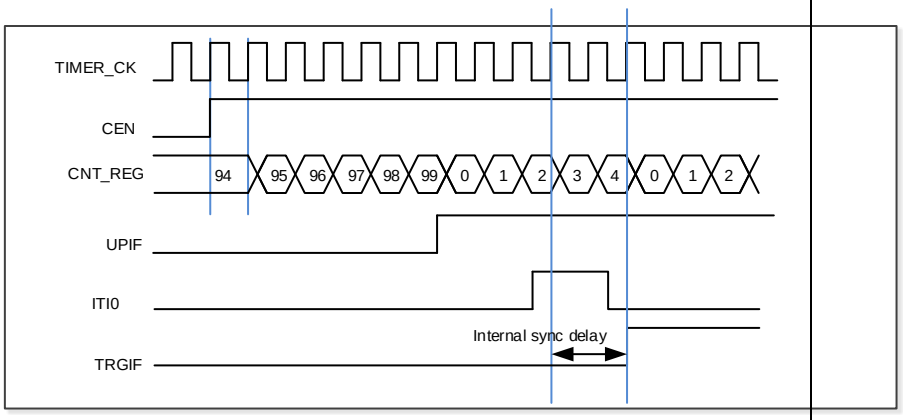
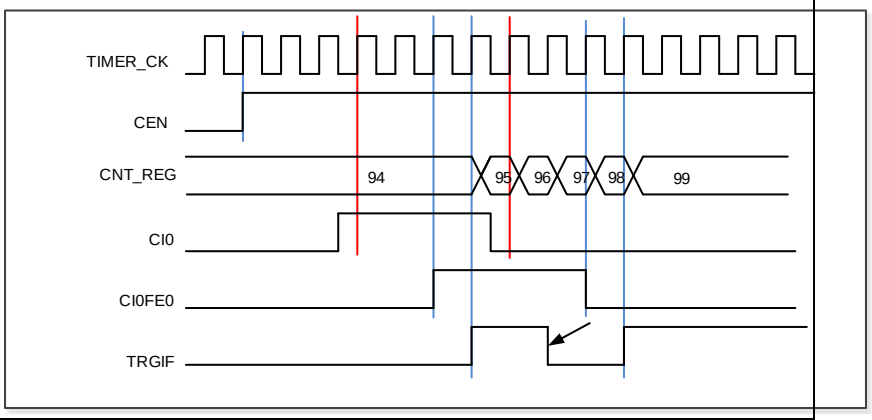
参考*霍尔传感器接口功能*。

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 SYSCFG_TIMERxCFG 寄存器中的 TSCFGy[3:0]（y=3,4,5）配置这些模式。

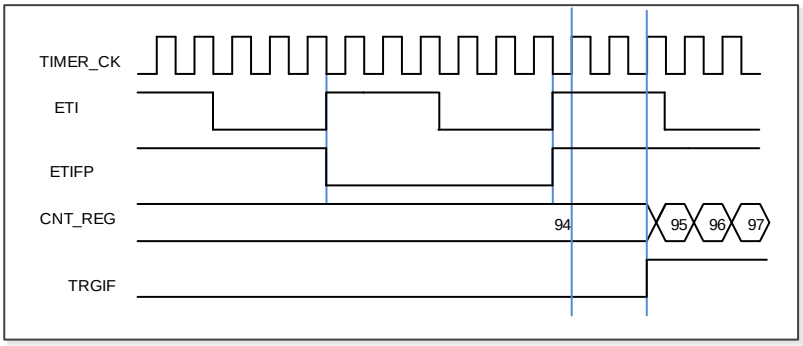
表 15-5. 从模式示例（通用定时器 L0）

	模式选择	触发源选择	极性选择	滤波和预分频
列举	TSCFGy[3:0] y=3（复位模式） y=4（暂停模式） y=5（事件模式）	TSCFGy[3:0] 0001：ITI0 0010：ITI1 0011：ITI2 0100：ITI3 0101：CI0F_ED 0110：CI0FE0 0111：CI1FE1 1000：ETIFP	如果触发源是CI0FE0或者CI1FE1，配置CHxP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	若触发源为ITIx，滤波和预分频不可用。 若触发源为CIx，可配置CHxCAPFLT设置滤波，预分频不可用。 若触发源为ETIFP，滤波和预分频均可用。
例1	复位模式 当触发输入上升沿到来时，计数器清零重启。	TSCFG3[3:0] = 4’b0001选择ITI0为触发源。	若触发源是ITI0，极性选择不可用。	若触发源是ITI0，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
	图 15-46. 复位模式  			
例2	暂停模式 当触发输入为低的时候，计数器暂停计数，当触发输入为高时，计数器计数。	TSCFG4[3:0] = 4'b 0110 选择CI0FE0为触发源。	TI0S=0（非异或）[CH0P=0] CI0FE0不反相。捕获发生在上升沿 。	在这个例子中滤波被旁路。
	图 15-47. 暂停模式  			
例3	事件模式 触发输入的上升沿计数器开始计数。	TSCFG5[3:0] = 4'b 1000 选择ETIFP为触发源。	ETP = 0，ETI极性不改变。	ETPSC = 1，ETI 2分频。 ETFC = 0，ETI 无滤波。

	模式选择	触发源选择	极性选择	滤波和预分频

图 15-48. 事件模式



单脉冲模式

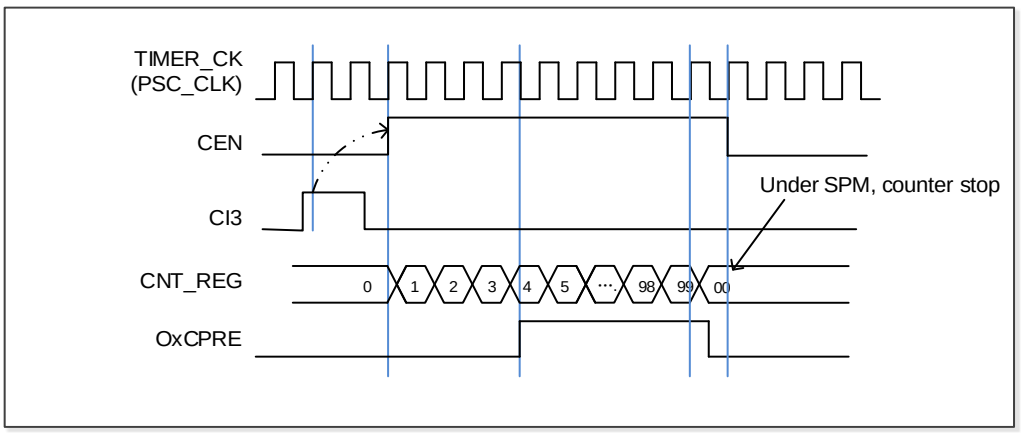
设置 TIMERx_CTL0 寄存器的 SPM 位置 1，使能单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 CHxCOMCTL 配置 TIMERx 为 PWM 模式或者比较模式。

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 PWM 模式 0 或 PWM 模式 1 时 CHxCOMFEN 位才可用，触发源来源于触发信号。

图 15-49. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 展示了一个例子。

图 15-49. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99



定时器互连

参考*高级定时器（TIMERx,x=0）*。

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 RISCV 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。

15.2.5. 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编码器模式时，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、暂停模式和编码器模式才能工作。

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

该寄存器通过字访问（32 位）。

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	保留						
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 TSCFGy[3:0](y=3,4,5)不能为 4’b0111。

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 SYSCFG_TIMERxCFG 寄存器。

13:12 ETPSC[1:0]

外部触发预分频

外部触发信号 ETI 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 ETIP 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_{CK_TIMER}
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。

6:0 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 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。 当通道 0 在输入模式下时，读 TIMERx_CH0CV 会将此标志清零。 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 TSCFG7[3:0] 位域（位于 SYSCFG_TIMERxCFG（x=1，2）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 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：禁能通道 0 输出比较快速功能. 1：使能通道 0 输出比较快速功能。
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 TSCFG7[3:0] 位域（位于 SYSCFG_TIMERxCFG（x=1，2）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 TSCFG7[3:0] 位域（位于 SYSCFG_TIMERxCFG（x=1，2）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 2 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 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：禁能通道 2 输出比较快速功能。 1：使能通道 2 输出比较快速功能。
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 TSCFG7[3:0] 位域（位于

SYSCFG_TIMERxCFG（x=1，2）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）（x=1,2）

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
CNT[31:16]															
rw															
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

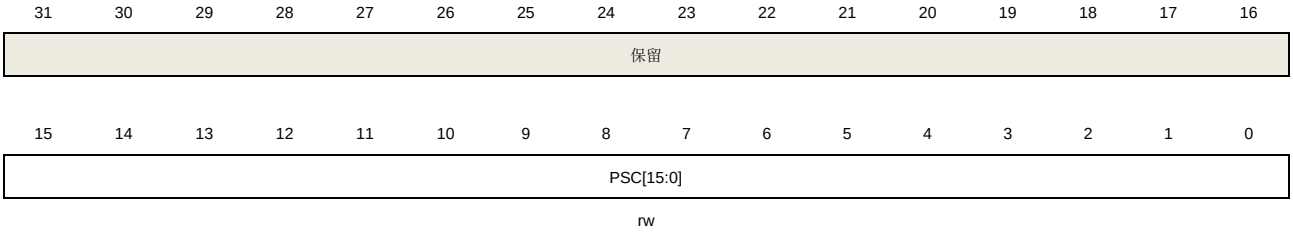
31:0	CNT[31:0]	这些位是当前的计数值。写操作能改变计数器值。

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。



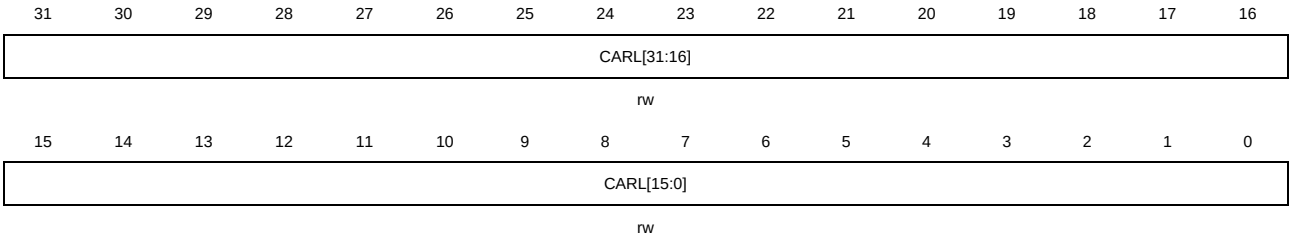
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以（PSC+1），每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

计数器自动重载寄存器（TIMERx_CAR）（x=1,2）

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
31:0	CARL[31:0]	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）（x=1,2）

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
CH0VAL[31:16]															
rw															

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
31:0	CH0VAL[31:0]	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）（x=1,2）

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
CH1VAL[31:16]															
rw															

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
31:0	CH1VAL[31:0]	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）（x=1,2）

地址偏移：0x3C

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
CH2VAL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

位/位域	名称	描述
31:0	CH2VAL[31:0]	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）（x=1,2）

地址偏移：0x40

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
CH3VAL[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

位/位域	名称	描述
31:0	CH3VAL[31:0]	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。

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			DMATC[4:0]					保留			DMATA [4:0]				
			rw								rw				

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）（x=1,2）

地址偏移：0x4C
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
DMATB[31:16]															
rw															

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

位/位域	名称	描述
31:0	DMATB [31:0]	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。

通道输入重映射寄存器（TIMERx_IRMP）（x=2）

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				CI3_RMP[1:0]		保留									
				rw											

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:10	CI3_RMP[1:0]	通道 3 输入重映射 00：通道 3 输入连接到 GPIO（TIMER2_CH3） 01：通道 3 输入连接到 IRC32K 10：通道 3 输入连接到 LXTAL 11：通道 3 输入连接到 CKOUT
9:0	保留	必须保持复位值。

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。

15.3. 通用定时器 L4（TIMERx,x=15,16）

15.3.1. 简介

通用定时器 L4（TIMER15/16）是单通道定时器，支持输入捕获和输出比较。可以产生 PWM 信号控制电机和电源管理。通用定时器 L4 含有一个 16 位无符号计数器。

通用定时器 L4 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

通用定时器 L4 包含了一个死区时间插入模块，非常适合电机控制。

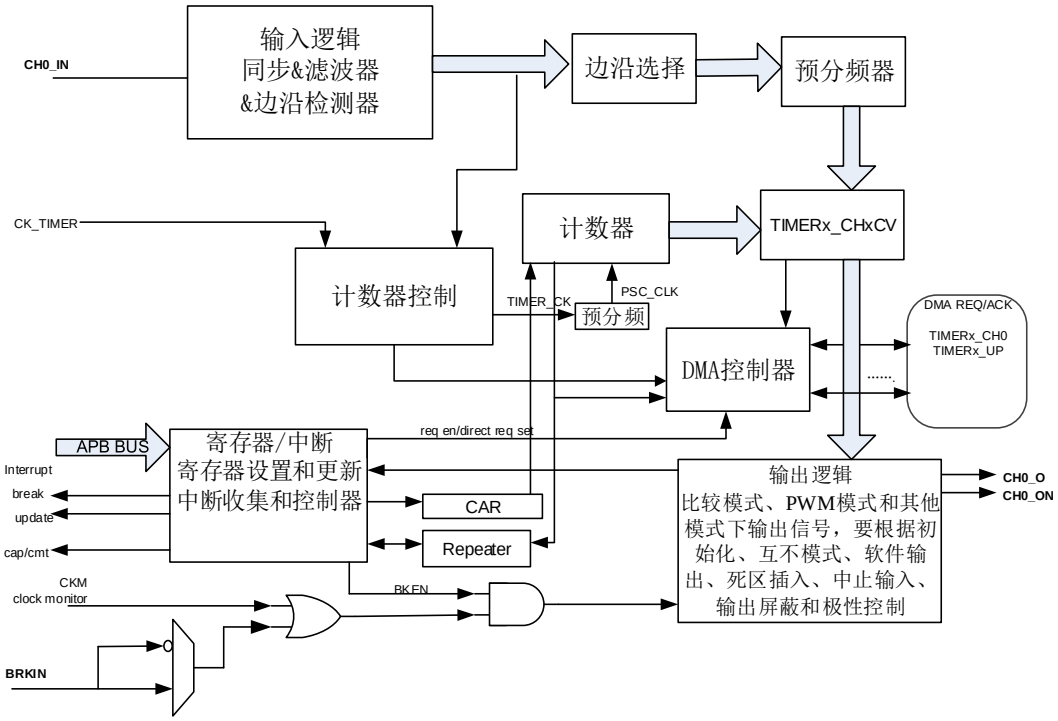
15.3.2. 主要特性

- 总通道数：1；
- 计数器宽度：16位；
- 时钟源可选：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；
- 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；
- 可编程的死区时间；
- 自动重装载功能；
- 可编程的计数器重复功能；
- 中止输入功能；
- 中断输出和DMA请求：更新事件，比较/捕获事件和中止事件。

15.3.3. 结构框图

*图15-50. 通用定时器L4结构框图*提供了通用定时器L4的内部配置细节。

图 15-50. 通用定时器 L4 结构框图



15.3.4. 功能描述

时钟源配置

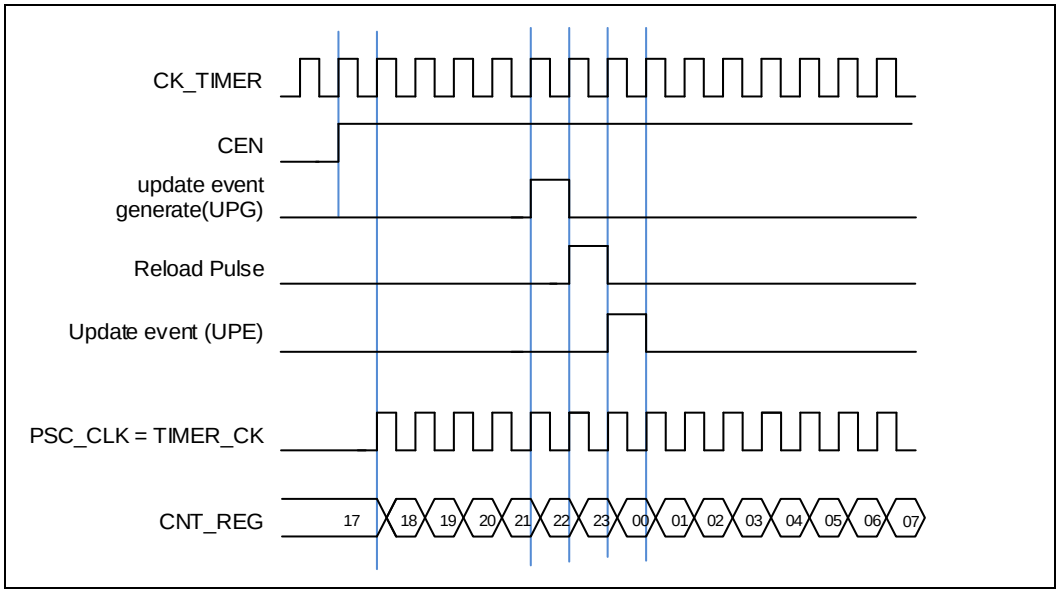
通用定时器 L4 由内部时钟源 CK_TIMER 驱动.

■ 定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

通用定时器 L4 只有一个时钟源：内部时钟源。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。

驱动预分频器计数的 TIMER_CK 等于来自于 RCU 模块的 CK_TIMER

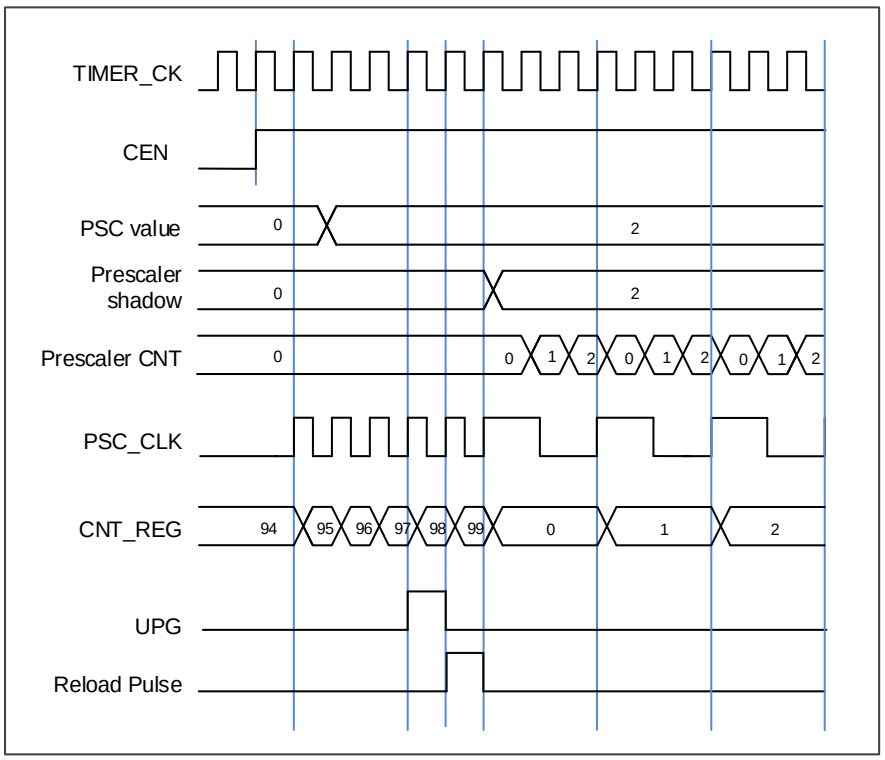
图 15-51. 内部时钟分频为 1 时，计数器的时序图



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 15-52. 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。如果设置了重复计数器，在（TIMERx_CREP+1）次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。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（重复计数器，自动重载寄存器，预分频寄存器）都将被更新。

*图15-53. 向上计数时序图，PSC=0/2*和*图15-54. 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 15-53. 向上计数时序图，PSC=0/2

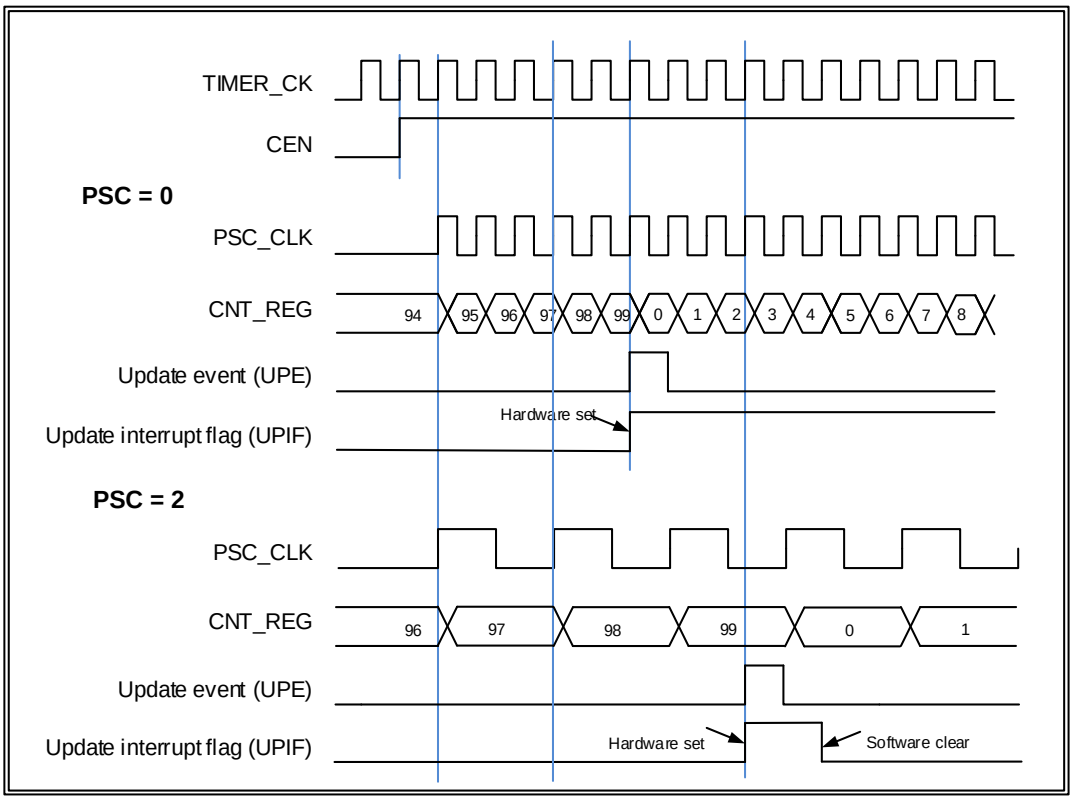
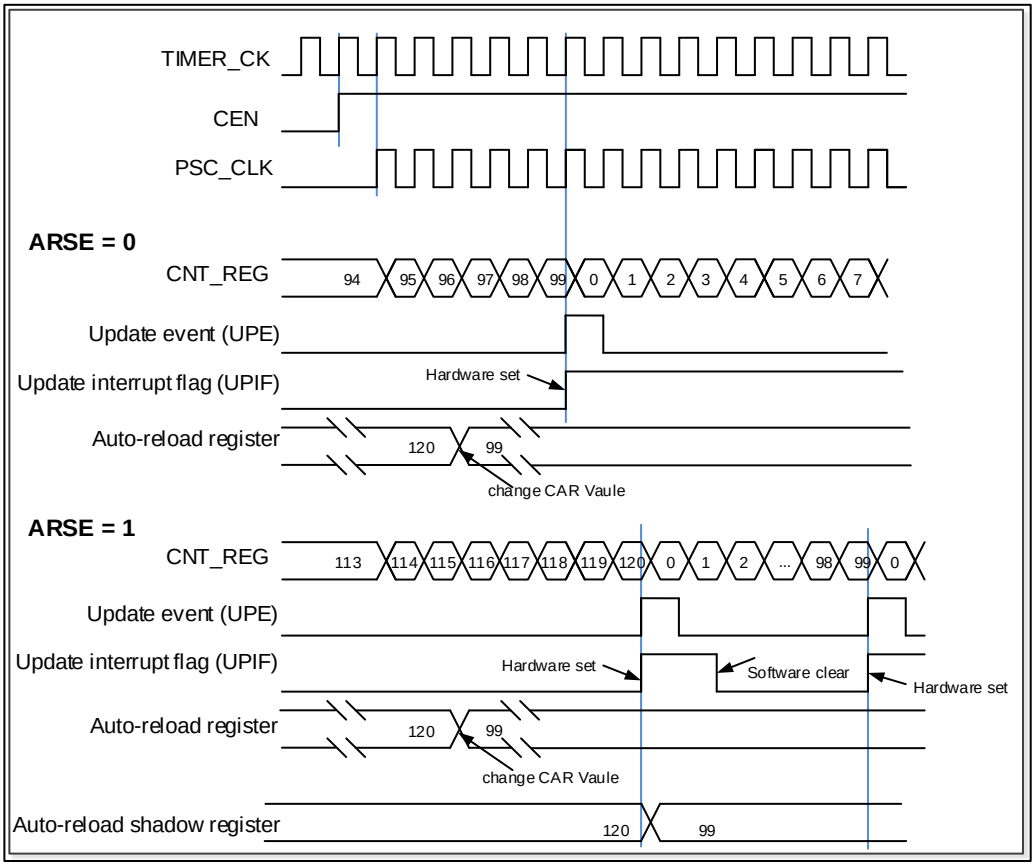


图 15-54. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值

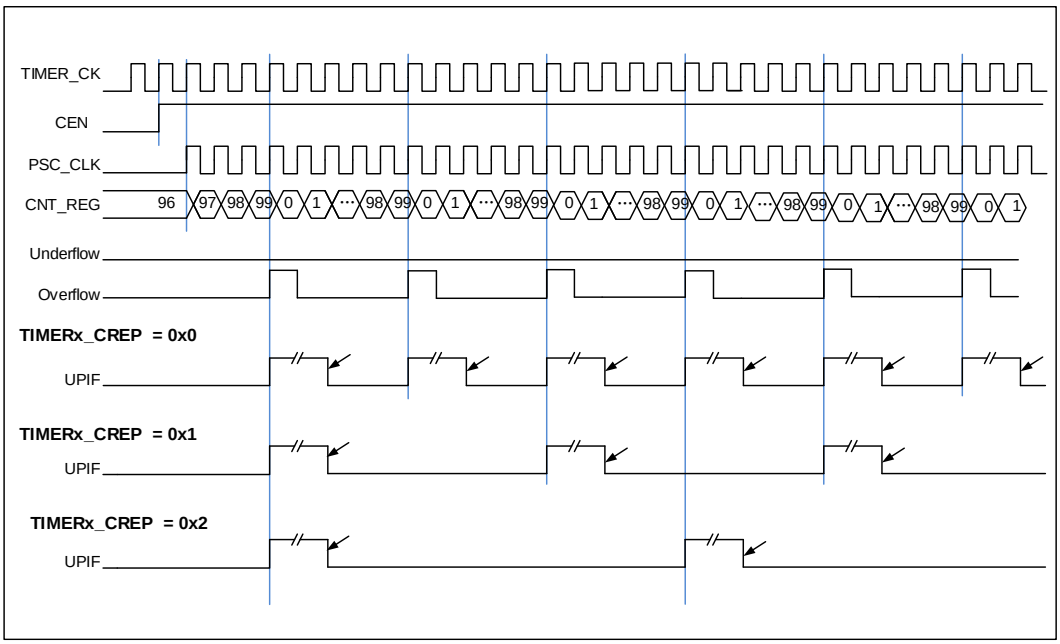


更新事件（来自上溢/下溢）频率配置

重复计数器是用来在 N+1 个计数周期之后产生更新事件，更新定时器的寄存器，N 为 TIMERx_CREP 寄存器的 CREP。向上计数模式下，重复计数器在每次计数器上溢时递减。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以重载 TIMERx_CREP 寄存器中 CREP 的值并产生一个更新事件。

图 15-55. 在向上计数模式下计数器重复时序图



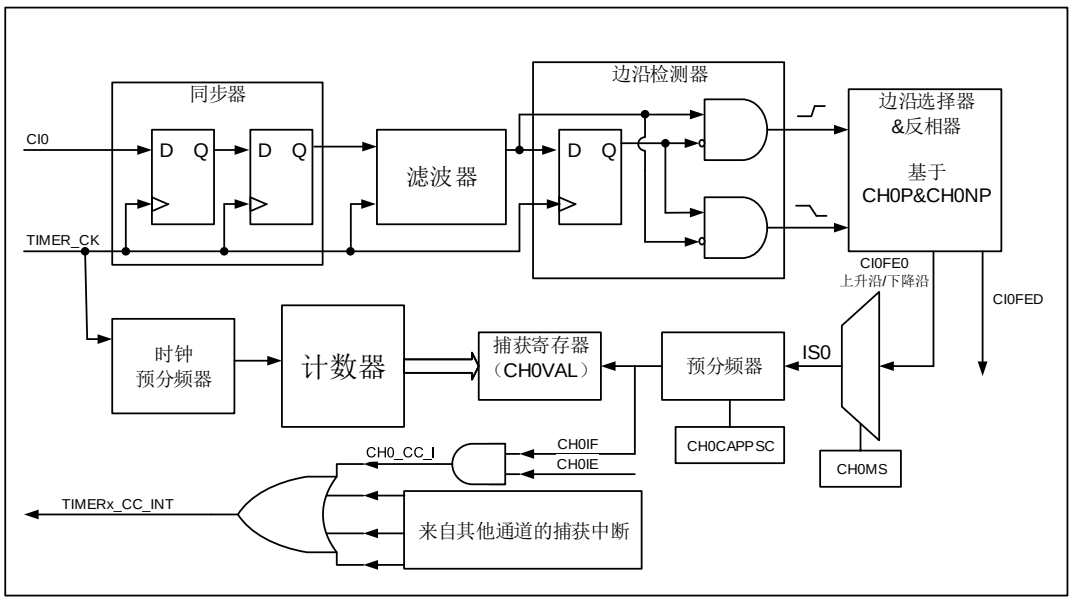
输入捕获和输出比较通道

通用定时器 L4 拥有一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。

- 通道输入捕获模式

通道输入捕获功能允许通道测量一个波形的时序，频率，周期等。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，若 CHxIE=1 则产生通道中断。

图 15-56. 通道输入捕获原理



通道输入信号 CIx 来源于 TIMERx_CHx 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。通过配置 CHxMS，还可以选择其他通道的输入信号或内部触发信号作为捕获信号。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，TIMERx_CHxCV 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/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_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，判断相应的中断和 DMA 请求是否被提出。

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周期。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，同时设置上升沿捕获。。TIMERX_CH0CV 寄存器测量 PWM 的周期值。

- 通道输出比较功能

图 15-57. 通道输出比较原理（带有互补输出的通道，x=0）

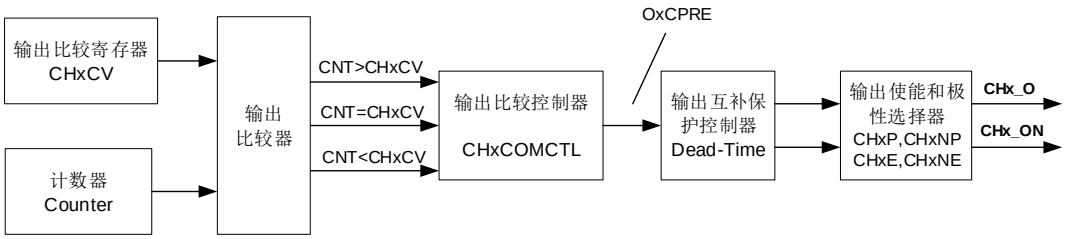


*图15-57. 通道输出比较原理（带有互补输出的通道，x=0）*给出了输出比较的原理电路。通道输出信号CHx_O/CHx_ON与OxCPRE信号（详情请见*通道输出准备信号*）的关系描述如下：OxCPRE信号高电平有效，CHx_O/CHx_ON的输出情况与OxCPRE信号，CHxP/CHxNP位和CHxE/CHxNE位有关（具体情况请见TIMERx_CHCTL2寄存器中的描述）。例如：

1）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。

2）当设置 CHxNP=1（CHx_ON 低电平有效，与 OxCPRE 输出极性相反）、CHxNE=1（CHx_ON 输出使能）时：

若 OxCPRE 输出有效（高）电平，则 CHx_ON 输出有效（低）电平；

若 OxCPRE 输出无效（低）电平，则 CHx_ON 输出无效（高）电平。

当 CH0_O 和 CH0_ON 同时输出时，CH0_O 和 CH0_ON 的具体输出情况还与 TIMERx_CCHP 寄存器中的相关位（ROS、IOS、POE 和 DTCFG 等位）有关。详情请见*通道输出互补 PWM*。

在通道输出比较功能，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 TIMERx_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 TIMERx_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/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。

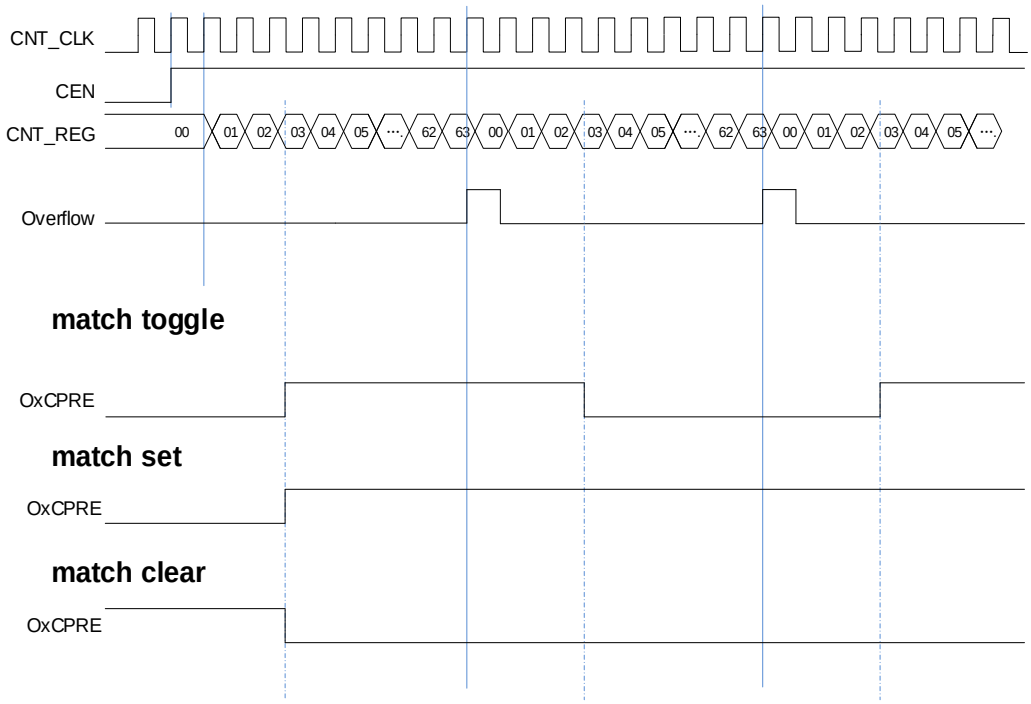
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

TIMERx_CHxCV 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15-58. 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 15-58. 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）。

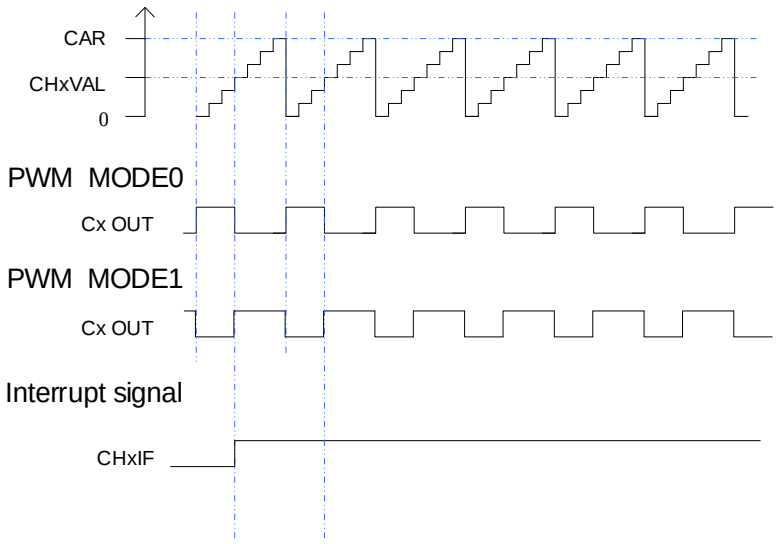
EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。

*图 15-59. PWM 时序图*显示了 PWM 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 15-59. PWM 时序图



通道输出准备信号

根据*图 15-57. 通道输出比较原理（带有互补输出的通道，x=0）*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。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。

通道输出互补 PWM

CHx_O 和 CHx_ON 是一对互补输出通道，这两个信号不能同时有效。TIMERx 只有一路有互补输出通道。互补信号 CHx_O 和 CHx_ON 是由一组参数来决定：TIMERx_CHCTL2 寄存器中的 CHxEN 和 CHxNEN 位，TIMERx_CCHP 寄存器中和 TIMERx_CTL1 寄存器中的 POEN, ROS, IOS, ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 CHxNP 位来决定。

表 15-6. 由参数控制的互补输出表

<table>
<tr><th colspan="5">互补参数</th><th colspan="2">输出状态</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>CHxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="4">0</td><td rowspan="4">0/1</td><td rowspan="3">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON 输出禁能(1)</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">CHx_O/CHx_ON输出关闭状态(2)：
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，CHx_ON = ISOxN (3)</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>x</td><td>x</td><td colspan="2">CHx_O/CHx_ON输出关闭状态：
通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP）；如果死区产生时钟未失效，在死区时间之后：
CHx_O = ISOx，CHx_ON = ISOxN</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">0</td><td rowspan="8">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON输出禁能</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O输出禁能</td><td>CHx_ON=OxCPRE⊕(4)CHxNP
CHx_ON输出使能</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>CHx_ON = LOW
CHx_ON输出禁能</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>CHx_ON=(!OxCPRE)(5)⊕CHxNP
CHx_ON输出使能</td></tr>
<tr><td rowspan="4">1</td><td rowspan="2">0</td><td>0</td><td>CHx_O = CHxP
CHx_O输出关闭状态</td><td>CHx_ON = CHxNP
CHx_ON输出关闭状态</td></tr>
<tr><td>1</td><td>CHx_O = CHxP
CHx_O输出关闭状态</td><td>CHx_O=OxCPRE⊕CHxNP
CHx_ON输出使能</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>CHx_ON = CHxNP
CHx_ON输出关闭状态</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O输出使能</td><td>CHx_ON=（!OxCPRE）⊕CHxNP
CHx_ON输出使能</td></tr>
</table>

注意：

（1） 输出禁能：CHx_O / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，

无上下拉时为悬空高阻态；

（2） 输出关闭状态：CHx_O / CHx_ON 输出无效电平（CHx_O = 0⊕CHxP = CHxP）；

（3） 详情见中止模式章节。

（4） ⊕：异或操作；

（5） (!OxCPRE)：OxCPRE 信号的互补信号。

互补 PWM 插入死区时间

设置CHxEN和CHxNEN为1'b1同时设置POEN，死区插入就会被使能。DTCFG位域定义了死区时间，死区时间对通道0有效。死区时间的细节，请参考TIMERx_CCHP寄存器。

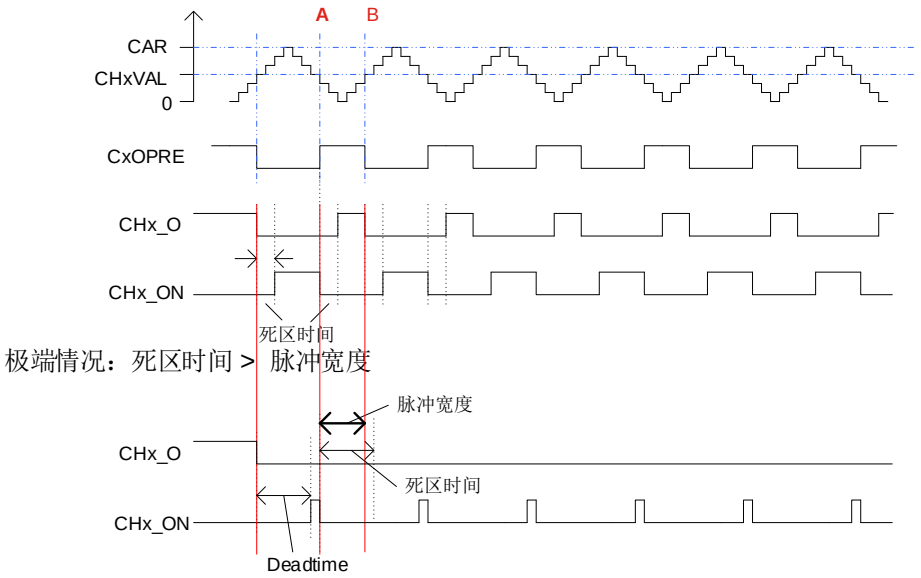
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器= CHxVAL），OxCPRE 反转。在*图 15-60. 带死区时间的互补输出*中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 CHx_ON 信号立刻变为低电平。同样，在 B 点，计数器再次匹配（TIMERx 计数器= CHxVAL），OxCPRE 信号被清 0，CHx_O 信号被立即清零，CHx_ON 信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些死角事件发生，例如：

- 如果死区延时大于或者等于CHx_O信号的占空比，CHx_O信号一直为无效值（如*图15-60. 带死区时间的互补输出*）。
- 如果死区延时大于或者等于CHx_ON信号的占空比，CHx_ON信号一直为无效值。

图 15-60. 带死区时间的互补输出



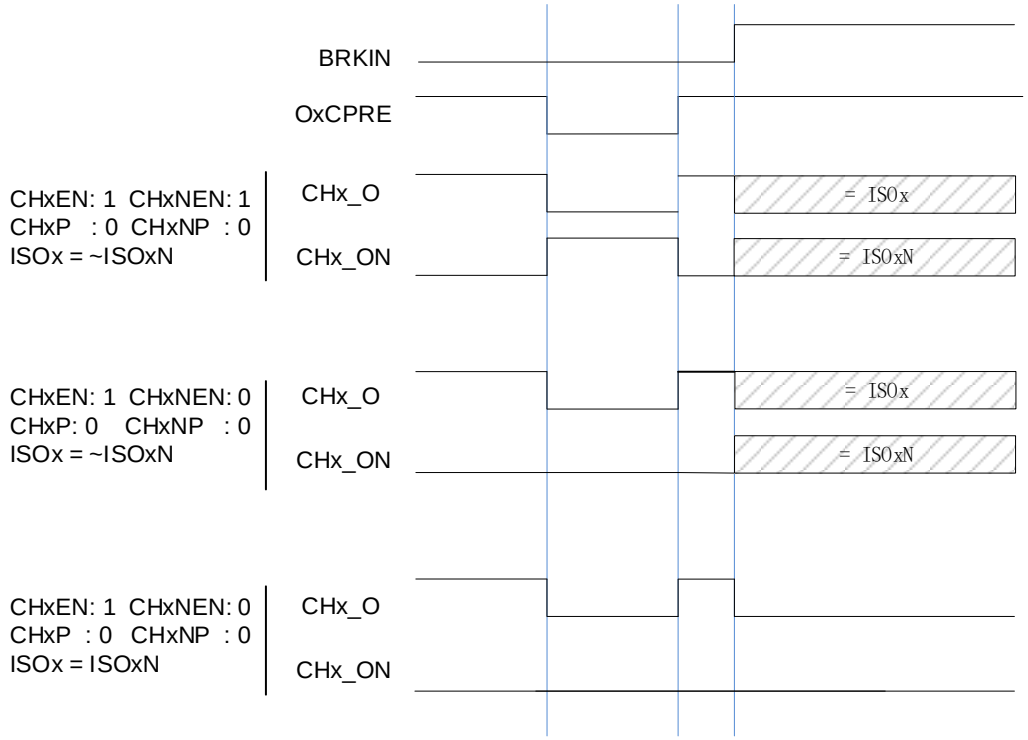
中止模式

使用中止模式时，输出 CHx_O 和 CHx_ON 信号电平被以下位控制，TIMERx_CCHP 寄存器的 POEN, IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。当中止事件发生时，CHx_O 和 CHx_ON 信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚，也可以选择 HXTAL 时钟失效事件。时钟失败事件由 RCU 中的时钟监视器（CKM）产生。将 TIMERx_CCHP 寄存器的 BRKEN 位置 1 可以使能中止功能。TIMERx_CCHP 寄存器的 BRKP 位决定了中止输入极性。

发生中止时，POEN 位被异步清除，一旦 POEN 位为 0，CHx_O 和 CHx_ON 被 TIMERx_CTL1 寄存器中的 ISOx 位和 ISOxN 驱动。如果 IOS=0，定时器释放输出使能，否则输出使能仍然为高。起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由 ISOx 和 ISOxN 位配置。

发生中止时，TIMERx_INTF 寄存器的 BRKIF 位被置 1。如果 BRKIE=1，中断产生。

图 15-61. 通道响应中止输入（高电平有效）时，输出信号的行为



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。

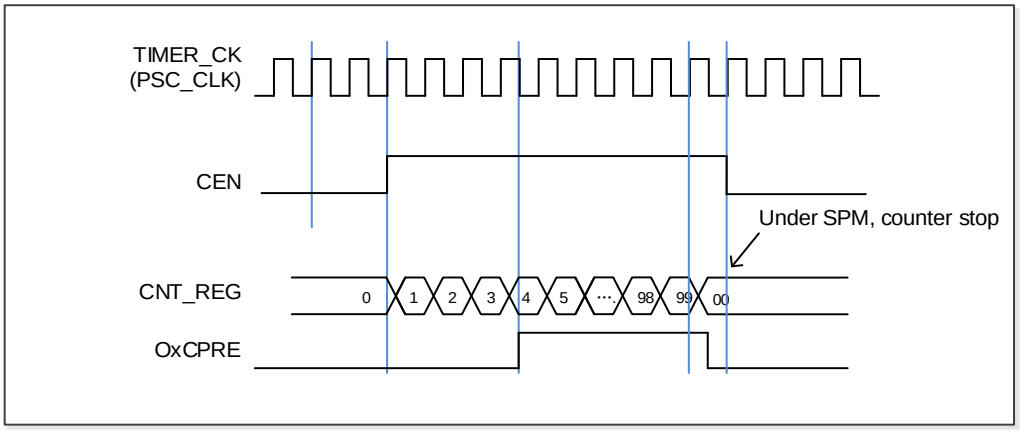
一旦设置定时器运行在单脉冲模式下，没有必要设置 TIMERx_CTL0 寄存器的定时器使能位 CEN=1 来使能计数器。触发信号沿或者软件写 CEN=1 都可以产生一个脉冲，此后 CEN 位一直保持为 1 直到更新事件发生或者 CEN 位被软件写 0。如果 CEN 位被软件清 0，计数器停止

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 15-62. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99 展示了一个例子。

图 15-62. 单脉冲模式，TIMERx_CHxCV = 4 TIMERx_CAR=99



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。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 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，定时器的发送 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 RISCV 内核停止，DBG_CTL1 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

15.3.5. TIMERx 寄存器（x=15,16）

TIMER15 基地址：0x4001 8000

TIMER16 基地址：0x4001 8400

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、暂停模式和编码器模式才能工作。

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						ISO0N	ISO0	保留				DMAS	CCUC	保留	CCSE
						rw	rw					rw	rw		rw

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	ISO0N	通道 0 的互补通道空闲状态输出 0：当 POEN 复位，CH0_ON 设置低电平. 1：当 POEN 复位，CH0_ON 设置高电平 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改.
8	ISO0	通道 0 的空闲状态输出 0：当 POEN 复位，CH0_O 设置低电平 1：当 POEN 复位，CH0_O 设置高电平 如果 CH0_ON 生效，一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改.
7:4	保留	必须保持复位值。

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	CCUC	换相控制影子寄存器更新控制 当换相控制影子寄存器（CHxEN, CHxNEN 和 CHxCOMCTL 位）使能（CCSE=1），这些影子寄存器更新控制如下： 0：CMTG 位被置 1 时更新影子寄存器 1：当 CMTG 位被置 1 或检测到 TRIGI 上升沿时，影子寄存器更新 当通道没有互补输出时，此位无效。
1	保留	必须保持复位值。.
0	CCSE	换相控制影子使能 0：影子寄存器 CHxEN, CHxNEN 和 CHxCOMCTL 位禁能. 1：影子寄存器 CHxEN, CHxNEN 和 CHxCOMCTL 位使能. 如果这些位已经被写入了，换相事件到来时这些位才被更新 当通道没有互补输出时，此位无效

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						CH0DEN	UPDEN	BRKIE	保留	CMTIE	保留			CH0IE	UPIE
						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	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	BRKIE	中止中断使能 0：禁止中止中断 1：使能中止中断

6	保留	必须保持复位值。.
5	CMTIE	换相更新中断使能 0：禁止换相更新中断 1：使能换相更新中断
4:2	保留	必须保持复位值。.
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						CH0OF	保留.	BRKIF	保留	CMTIF	保留.			CH0IF	UPIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

位/位域	名称	描述
31:10	保留	必须保持复位值。.
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	保留	必须保持复位值。.
7	BRKIF	中止中断标志位 当中止输入有效时，由硬件对该位置‘1’。 当中止输入无效时，则该位可由软件清‘0’。 0：无中止事件产生 1：中止输入上检测到有效电平
5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置 1，此位由软件清 0。

0：无通道换相更新中断发生
1：通道换相更新中断发生

4:2	保留	必须保持复位值。.
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。 当通道 0 在输入模式下时，读 TIMERx_CH0CV 会将此标志清零。 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								BRKG	保留	CMTG	保留			CH0G	UPG
								w		w				w	w

位/位域	名称	描述
31:8	保留	必须保持复位值。.
7	BRKG	产生中止事件 该位由软件置 1，用于产生一个中止事件，由硬件自动清 0。当此位被置 1 时，POEN 位被清 0 且 BRKIF 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：不产生中止事件 1：产生中止事件
5	CMTG	通道换相更新事件发生 此位由软件置 1，由硬件自动清 0. 当此位被置 1，通道捕获/比较控制寄存器（CHxEN, CHxNEN 和 CHxCOMCTL）的互补输出被更新（根据 TIMERx_CTL1 中 CCSE 值）。 0：不产生通道控制更新事件 1：产生通道控制更新事件

4:2	保留	必须保持复位值。.
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，在向上计数模式中，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

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
保留								保留	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
								CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
									rw			rw		rw	

输出比较模式：

位/位域	名称	描述
31:7	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	保留	必须保持复位值。
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：保留 11：保留

输入捕获模式：

位/位域	名称	描述
31:8	保留	必须保持复位值。.
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 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.												CH0NP	CH0NEN	CH0P	CH0EN
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
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互补输出高电平为有效电平 1：通道0互补输出低电平为有效电平 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	CH0NEN	通道 0 互补输出使能 当通道 0 配置为输出模式时，将此位置 1 使能通道 0 的互补输出。 0：禁止通道 0 互补输出 1：使能通道 0 互补输出
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。

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。

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 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。

重复计数寄存器（TIMERx_CREP）

地址偏移：0x30

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								CREP[7:0]							
								rw							

位/位域	名称	描述
31:8	保留	必须保持复位值。.
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响（前提是影子寄存器被使能）。

通道 0 捕获/比较寄存器（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。

互补通道保护寄存器（TIMERx_CCHP）

地址偏移：0x44

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
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
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	POEN	所有的通道输出使能 该位通过以下方式置 1： –写 1 置位 –如果 OAEN=1，则在下一次更新事件发生时置 1. 该位通过以下方式清 0： –写 1 清 0 –有效的中止输入（异步） 如果一个通道配置为输出模式，如果设置了相应的使能位（TIMERx_CHCTL2 寄存器的 CHxEN, CHxNEN 位），则开启 CHx_O 和 CHx_ON 输出。 0：禁止通道输出或强制为空闲状态 1：使能通道输出
14	OAEN	自动输出使能 0：POEN 位只能使用软件方式置 1 1：如果中止输入无效，下一次更新事件发生时，POEN 位将会置 1 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
13	BRKP	中止极性 此位定义了中止输入信号 BRKIN 的极性。 0：中止输入低电平有效 1：中止输入高电平有效
12	BRKEN	中止使能 此位置 1 使能中止事件和 CCS 时钟失败事件输入。 0：禁能中止输入 1：使能中止输入 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。.
11	ROS	运行模式下“关闭状态”使能 当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见*表 15-6. 由参数控制的互补输出表*。 0：输出“关闭状态”禁能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“禁能状态”。 1：输出“关闭状态”使能。当 CHxEN 或者 CHxNEN 位被清零，对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下“关闭状态”使能

当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见*表 15-6. 由参数控制的互补输出表*。
0：输出“关闭状态”禁能。当 CHxEN 和 CHxNEN 位均被清零，对应通道为输出“禁能状态”。
1：输出“关闭状态”使能。不论 CHxEN 和 CHxNEN 位的值，对应通道为输出“关闭状态”。
此位在 TIMERx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。

9:8	PROT[1:0]	互补寄存器保护控制 这两位定义了寄存器的写保护特性。 00：禁能保护模式。无写保护. 01：PROT 模式 0。TIMERx_CTL1 寄存器中 ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护 10：PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外，还有 TIMERx_CHCTL2 寄存器中 CHxP/CHxNP 位（如果相应通道配置为输出模式），TIMERx_CCHP 寄存器中 ROS/IOS 位。 11：PROT 模式 2.。除了 PROT 模式 1 下的寄存器写保护外，还有 TIMERx_CHCTLR0/1 中 CHxCOMCTL/ CHxCOMSEN 位（如果相关通道配置为输出模式）写保护。 系统复位后这两位只能被写一次，一旦 TIMERx_CCHP 寄存器被写入，这两位被写保护
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下： DTCFG [7:5] =3’b0xx：DTvalue =DTCFG [7:0]x tDT, tDT=tDTS. DTCFG [7:5] =3’b 10x：DTvalue =（64+DTCFG [5:0]）xtDT, tDT =tDTS*2. DTCFG [7:5] =3’b 110：DTvalue =（32+DTCFG [4:0]）xtDT, tDT=tDTS*8. DTCFG [7:5] =3’b 111：DTvalue =（32+DTCFG [4:0]）xtDT, tDT =tDTS*16. 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]=00 时才可修改。

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。

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			DMATC[4:0]					保留			DMATA [4:0]				
			rw								rw				

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。

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
DMATB[15:0]															
rw															

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	OUTSEL
														rw	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	OUTSEL	输出值选择位 此位由软件写 1 或清 0。 1：如果 POEN 位与 IOS 位均为 0，则输出禁能 0：无影响

15.4. 基本定时器（TIMERx, x=5）

15.4.1. 简介

基本定时器（TIMER5）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。

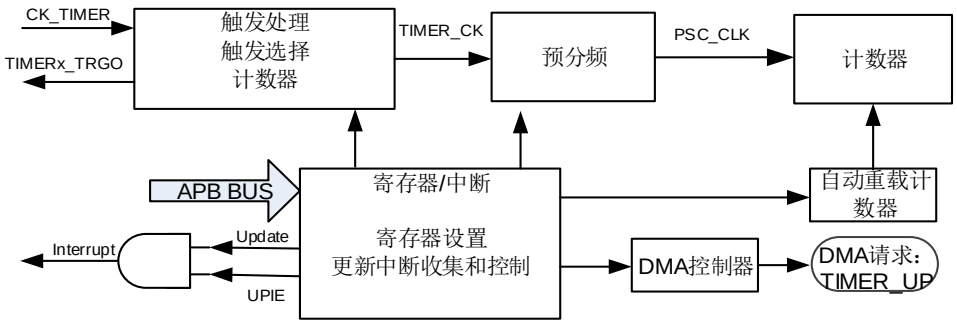
15.4.2. 主要特征

- 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。

15.4.3. 结构框图

*图15-63. 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 15-63. 基本定时器结构框图



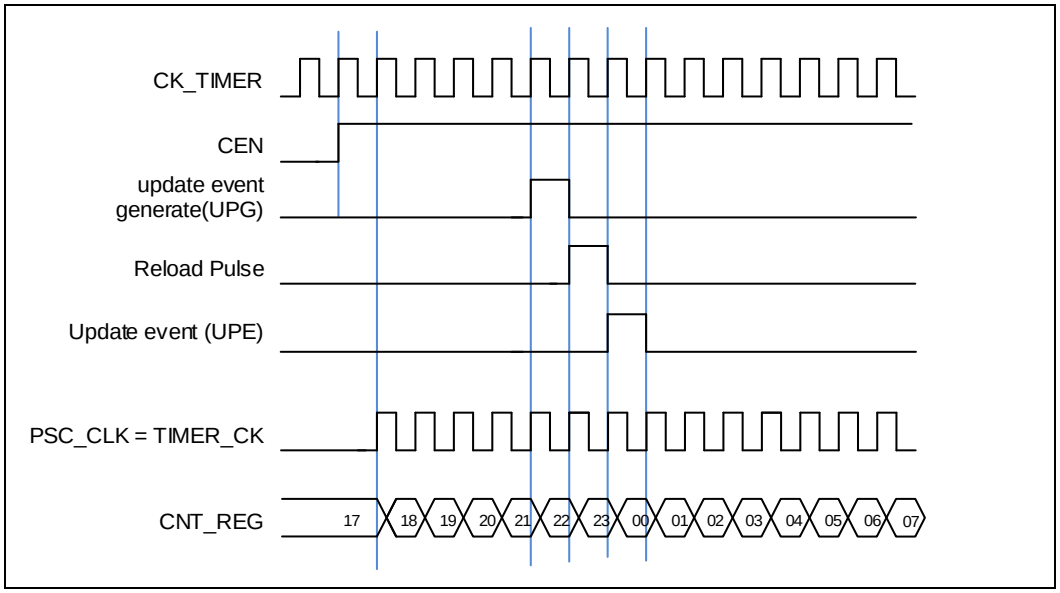
15.4.4. 功能描述

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。

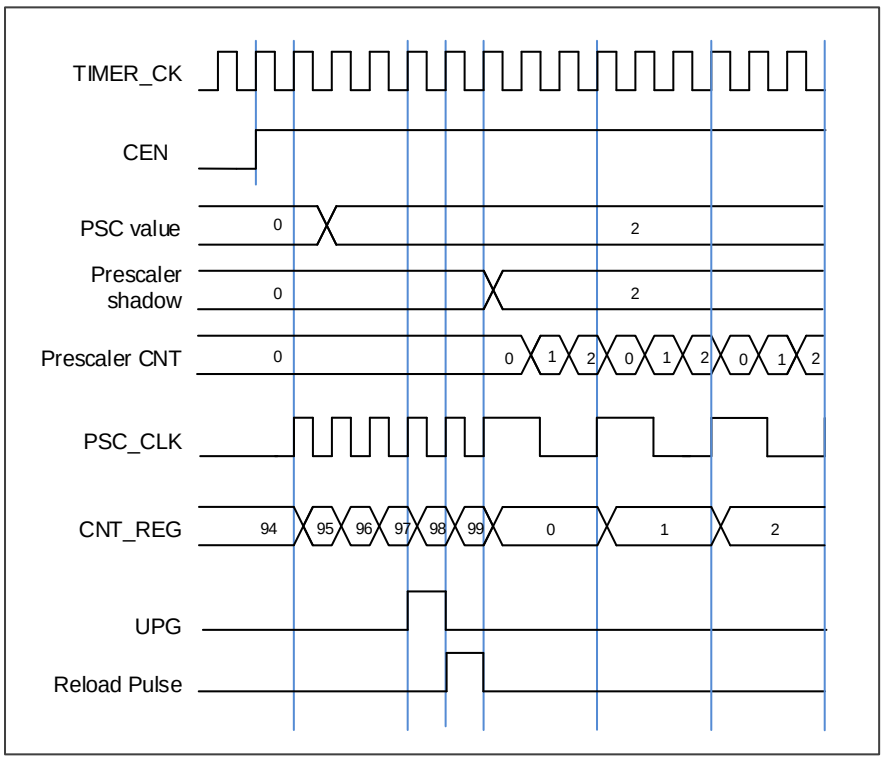
图 15-64. 内部时钟分频为 1 时，计数器的时序图



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 15-65. 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（重复计数寄存器，自动重载寄存器，预分频寄存器）都将被更新。

*图15-36. 向上计数时序图，PSC=0/2*和*图15-37. 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 15-66. 向上计数时序图，PSC=0/2

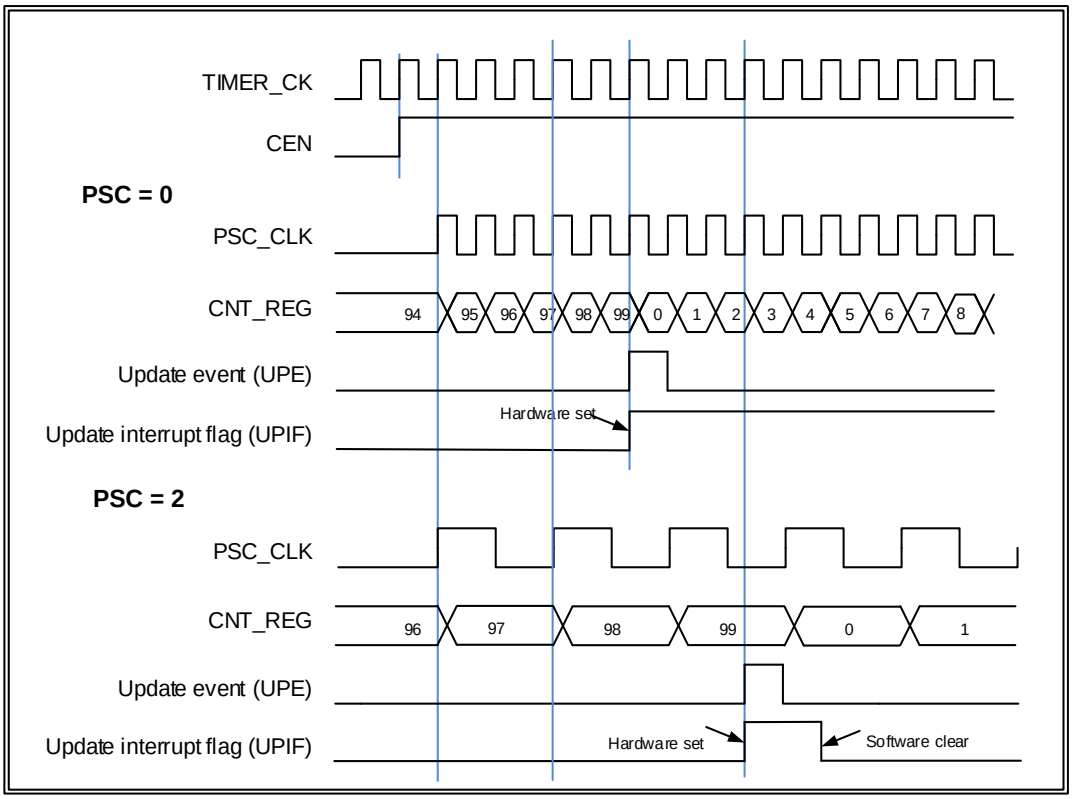
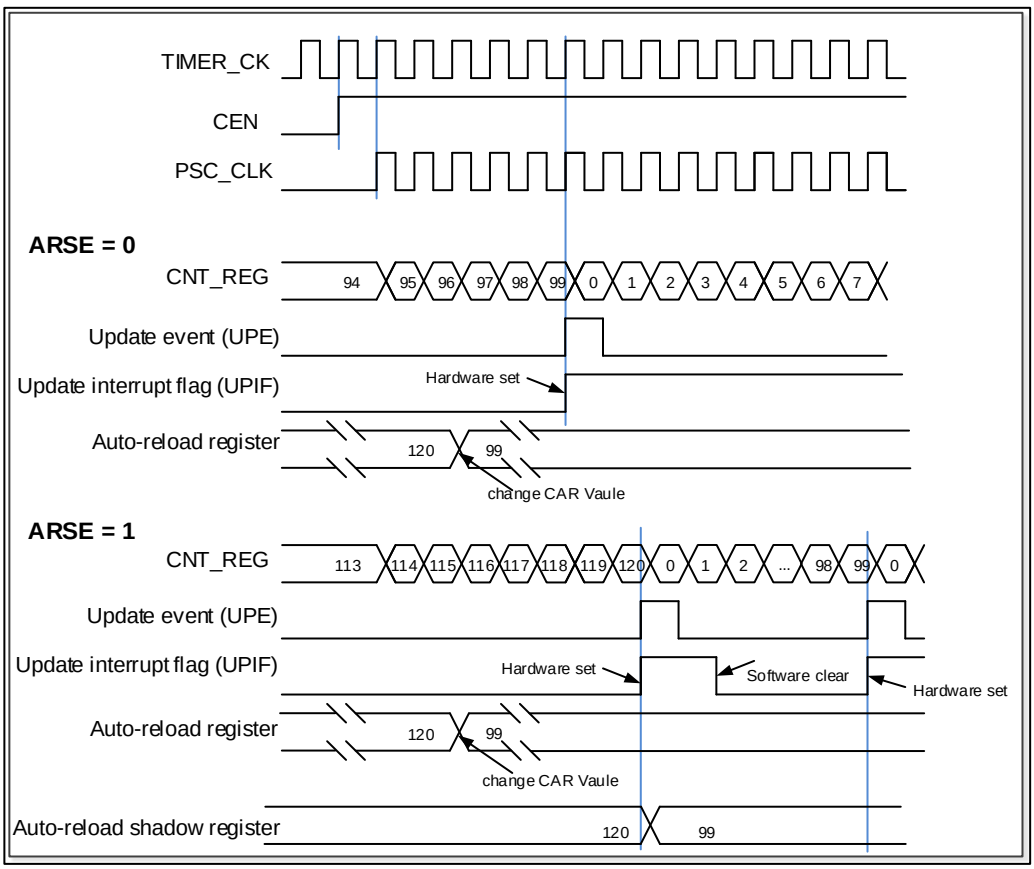


图 15-67. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



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 RISCV 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。

15.4.5. TIMERx 寄存器（x=5）

TIMER5 基地址：0x4000 1000

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								ARSE	保留			SPM	UPS	UPDIS	CEN
								rw				rw	rw	rw	rw

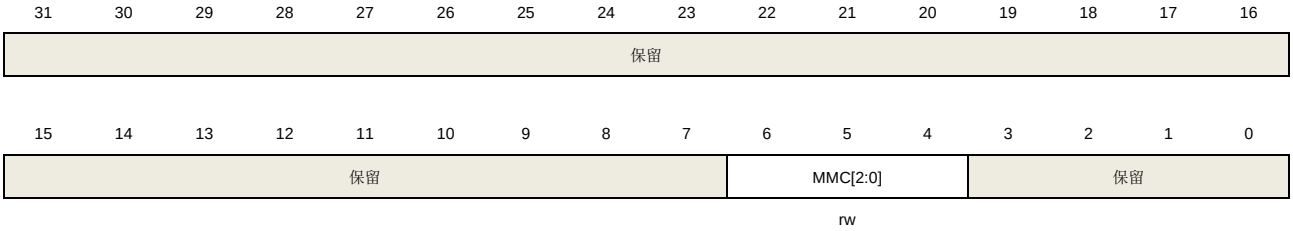
位/位域	名称	描述
31:8	保留	必须保持复位值。
7	ARSE	自动重载影子使能 0：禁能 TIMERx_CAR 寄存器的影子寄存器 1：使能 TIMERx_CAR 寄存器的影子寄存器
6:4	保留	必须保持复位值。
3	SPM	单脉冲模式 0：单脉冲模式禁能。更新事件发生后，计数器继续计数 1：单脉冲模式使能。在下一次更新事件发生时，CEN 位被硬件清零并且计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源. 0：以下事件均会产生更新中断或DMA请求： UPG位被置1 计数器溢出/下溢 复位模式产生的更新 1：下列事件会产生更新中断或DMA请求： 计数器溢出/下溢
1	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 0000

该寄存器只能按字（32 位）访问。

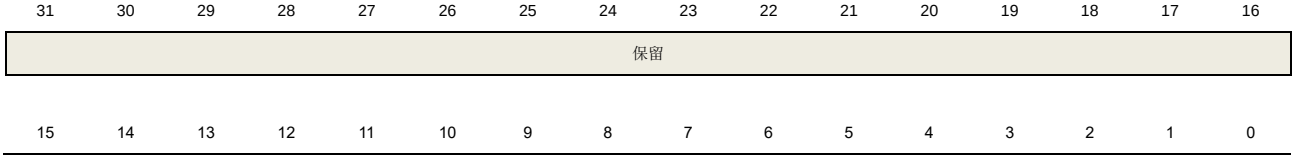


位/位域	名称	描述
31:7	保留	必须保持复位值。
6:4	MMC[2:0]	这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 000：当产生一个定时器复位事件后，输出一个TRGO信号，定时器复位源为： 主定时器产生一个复位事件 TIMERx_SWEVG寄存器中UPG位置1 001：当产生一个定时器使能事件后，输出一个TRGO信号，定时器使能源为： CEN位置1 在暂停模式下，触发输入置1 010：当产生一个定时器更新事件后，输出一个TRGO信号，更新事件源由UPDIS和UPS位决定
3:0	保留	必须保持复位值。

DMA 和中断使能寄存器（TIMERx_DMAINTEN）

地址偏移：0x0C
复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



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 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															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 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	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 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。

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。

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。.

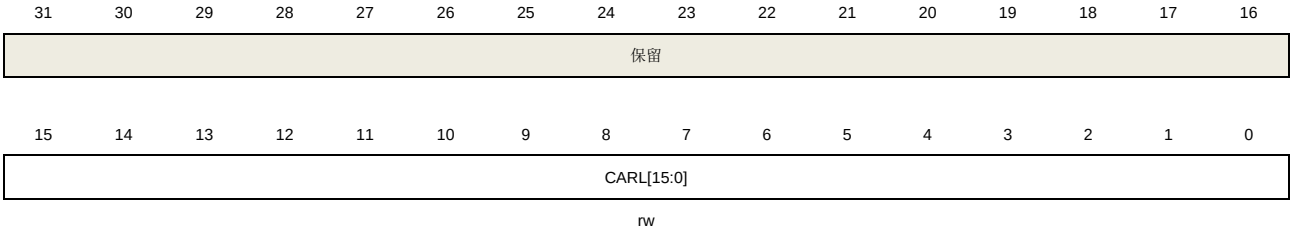
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 TIMER_CK 时钟除以（PSC+1），每次当更新事件产生时，PSC 的值被装入到对应的影子寄存器。

计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C

复位值：0x0000 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	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。

16. 通用同步异步收发器（USART）

16.1. 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（PCLK1，PCLK2 以及仅 USART0 可用的 CK_USART0）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。

16.2. 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 和和独立于 PCLK 时钟的 USART 时钟；
  - 不依赖PCLK设置的波特率设置；
- 可编程的波特率产生器，当时钟频率为160MHz，过采样为8，最高速度可达20MBits/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；
- 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 完全实现上述功能，但是 UART1/UART2 只实现了上面所介绍的部分功能，下面这些功能在 UART1 / UART2 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。

16.3. 功能描述

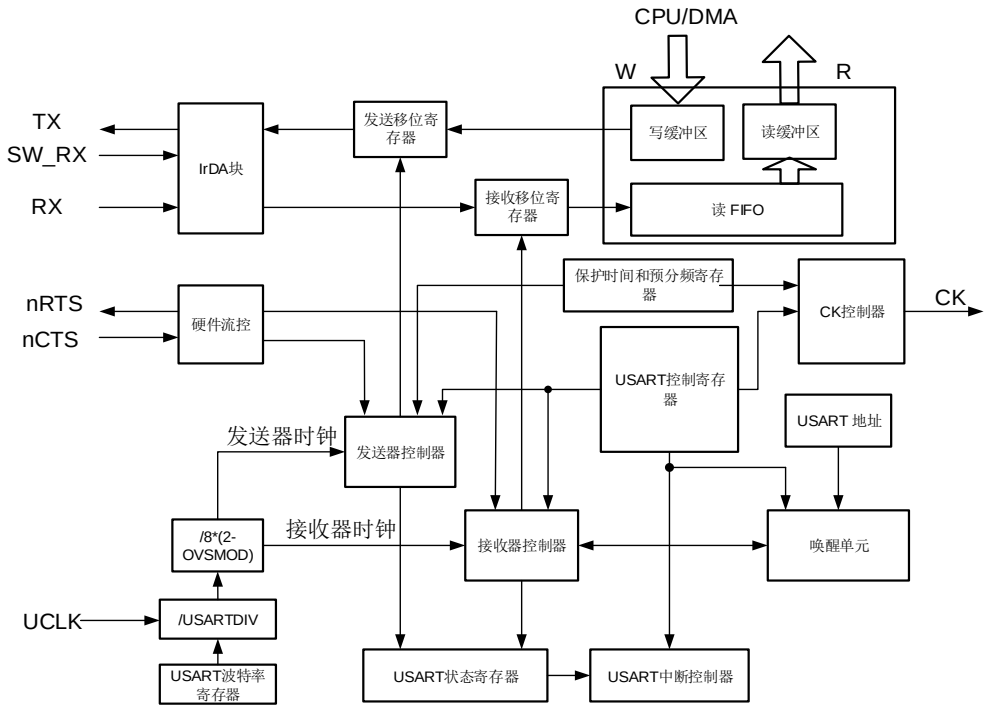
USART 接口通过*表 16-1. USART 重要引脚描述*中主要引脚从外部连接到其他设备。

表 16-1. USART 重要引脚描述

引脚	类型	描述
RX	输入	接收数据
TX	输出	发送数据。当 USART 使能后，若无数据发

引脚	类型	描述
	I/O（单线模式/智能卡模式）	送，默认为高电平
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

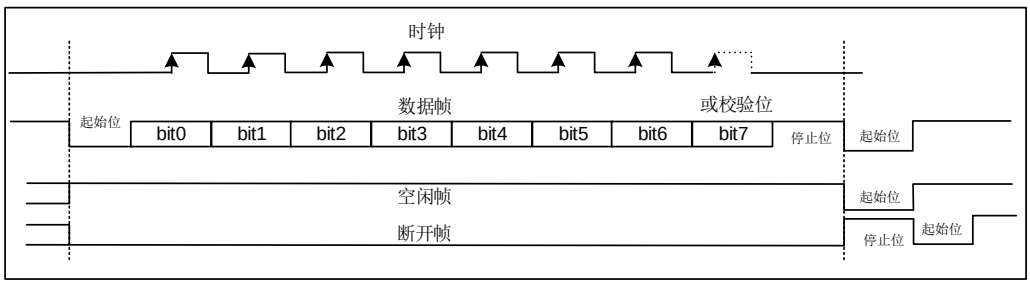
图 16-1. USART 模块内部框图



16.3.1. 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 16-2. 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 16-2. 停止位配置

STB[1:0]	停止位长度（位）	功能描述
00	1	默认值
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。

16.3.2. 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：

$$\text{USARTDIV}=\frac{\text{UCLK}}{16\times\text{Baud Rate}} \quad (16\text{-}1)$$

如果过采样是 8，公式为：

$$\text{USARTDIV}=\frac{\text{UCLK}}{8\times\text{Baud Rate}} \quad (16\text{-}2)$$

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。

16.3.3. 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。

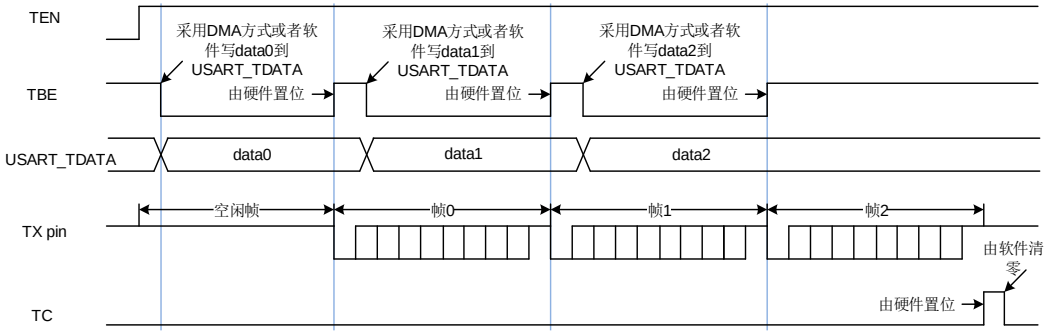
系统上电后，TBE 默认为高电平。在 USART_STAT 寄存器中 TBE 置位时，数据可以在不覆盖前一个数据的情况下写入 USART_TDATA 寄存器。当数据写入 USART_TDATA 寄存器，TBE 位将被清 0。在数据由 USART_TDATA 移入移位寄存器后，该位由硬件置 1。如果数据在一个发送过程正在进行时被写入 USART_TDATA 寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入 USART_TDATA 寄存器时，没有发送过程正在进行，TBE 位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄存器。

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 16-3. 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 16-3. 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。通过向 USART_INTC 寄存器的 TCC 位写 1 可将 TC 位清 0。

当 SBKCMD 置位时，会发送一个断开帧，发送完成后，SBKCMD 将被清 0。

16.3.4. 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。

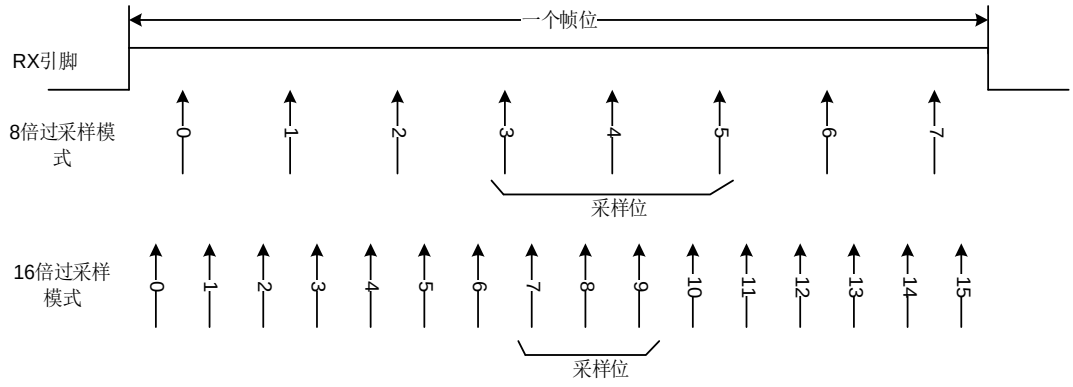
当接收到一个数据帧，USART_STAT 寄存器中的 RBNE 置位，如果设置了 USART_CTL0 寄存器中相应的中断使能位 RBNEIE，将会产生中断。在 USART_STAT 寄存器中可以观察接收状态标志。

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 16-4. 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。

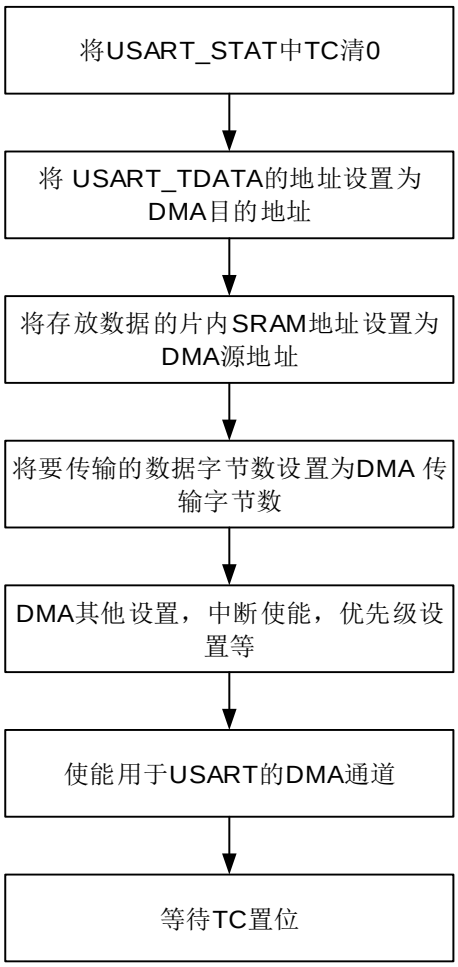
在一个接收过程中，NERR、PERR、FERR、ORERR 总是分别和 RBNE 同时置位。若接收过程中，产生了噪声错误（NERR）、校验错误（PERR）、帧错误（FERR）或溢出错误（ORERR），则 NERR、PERR、FERR 或 ORERR 将和 RBNE 同时置位。

16.3.5. 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 16-5. 采用 DMA 方式实现 USART 数据发送配置步骤*所示。

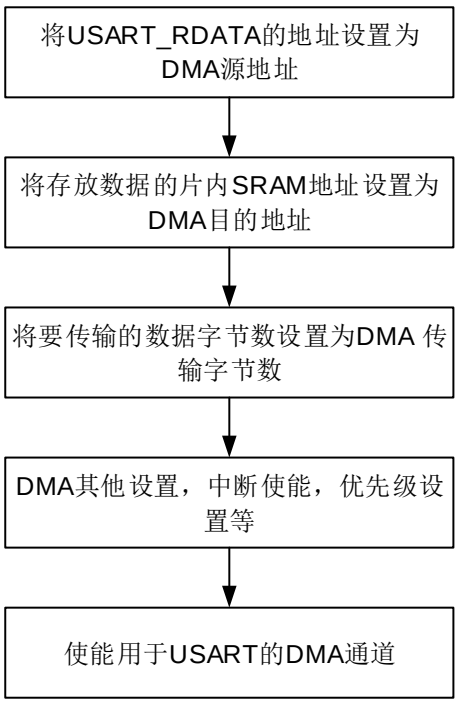
图 16-5. 采用 DMA 方式实现 USART 数据发送配置步骤



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 16-6. 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 16-6. 采用 DMA 方式实现 USART 数据接收配置步骤

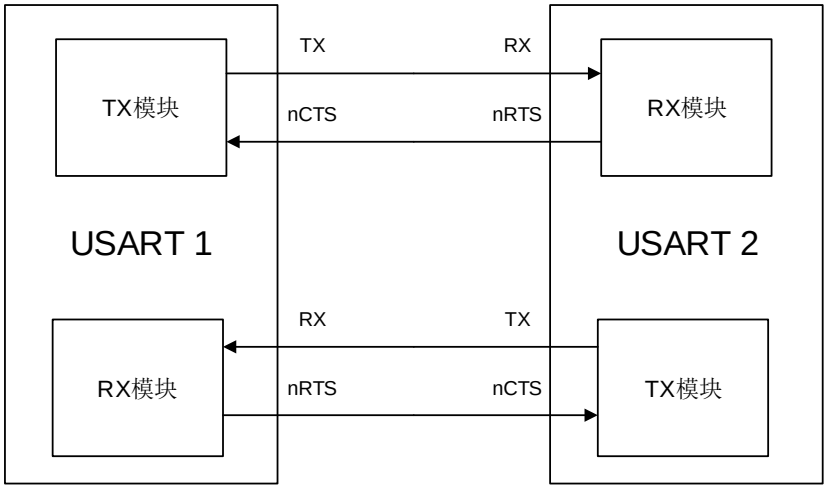


当 USART 接收到的数据数量达到了 DMA 传输数据数量，DMA 模块将产生传输完成中断。

16.3.6. 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 16-7. 两个 USART 之间的硬件流控制



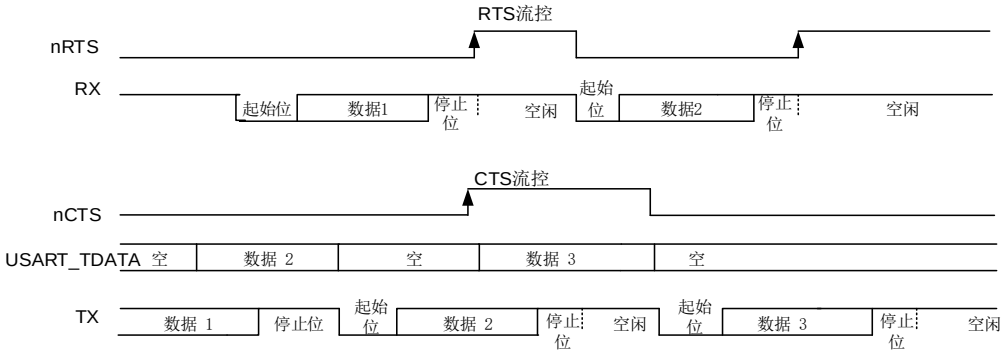
RTS 流控

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 16-8. 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。

16.3.7. 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。

16.3.8. LIN 模式

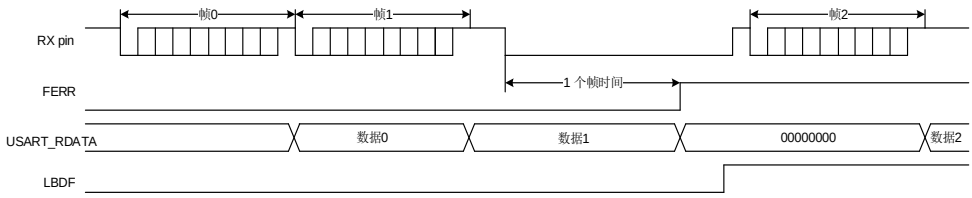
将 USART_CTL1 寄存器的 LMEN 置位即可使能本地互联网络模式。在 LIN 模式下，USART_CTL1 寄存器中 CKEN，STB[1:0]和 USART_CTL2 的 SCEN，HDEN，IREN 位都应该被清 0。

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。

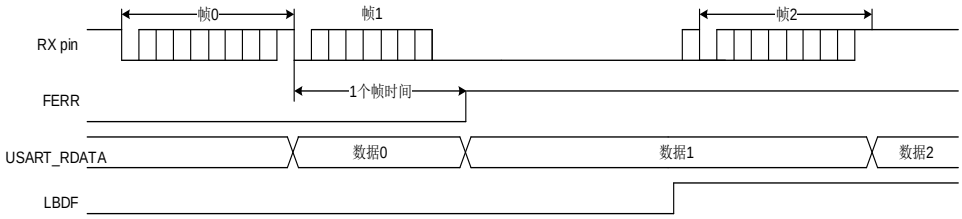
如*图 16-9. 空闲状态下检测断开帧*所示，如果断开帧发生在空闲状态下，USART 接收器会接收到一个全 0 数据帧，同时 FERR 置位。

图 16-9. 空闲状态下检测断开帧



如*图 16-10. 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 16-10. 数据传输过程中检测断开帧



16.3.9. 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 16-11. 同步模式下的 USART 示例

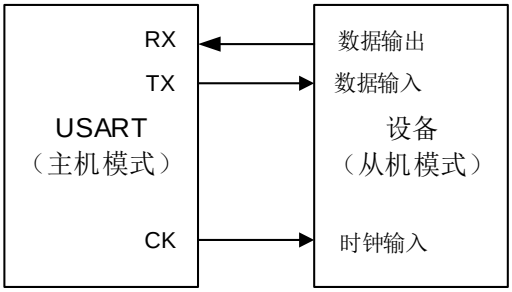
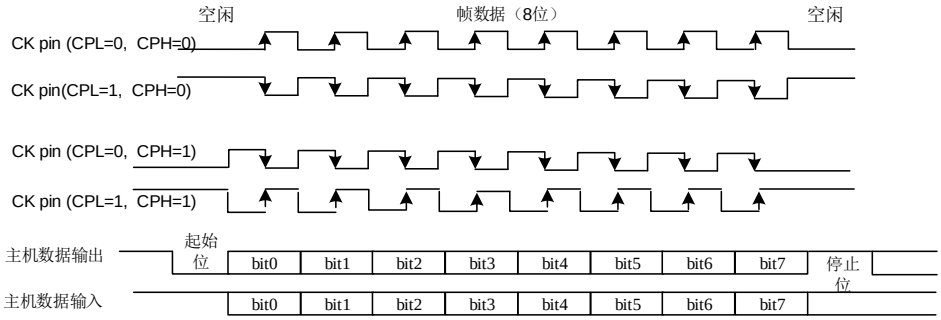


图 16-12. 8-bit 格式的 USART 同步通信波形（CLEN=1）

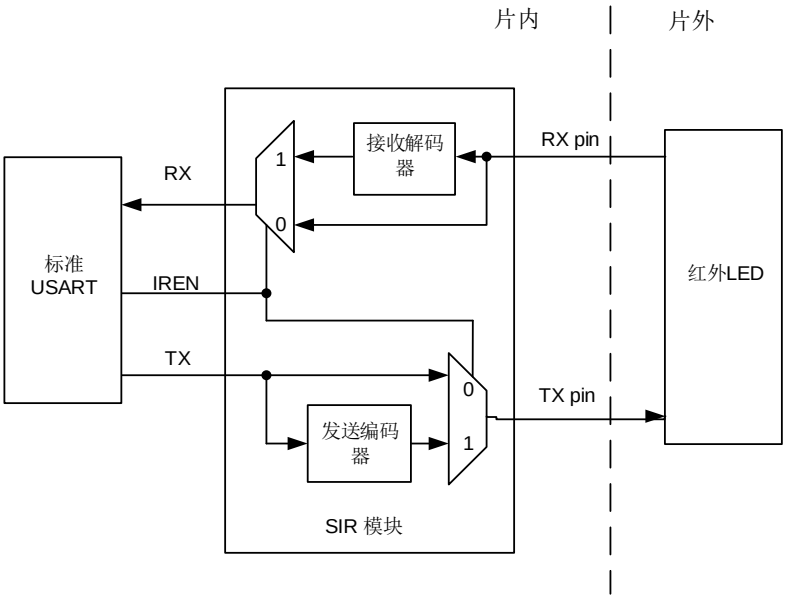


16.3.10. 串行红外（IrDA SIR）编解码功能模块

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。

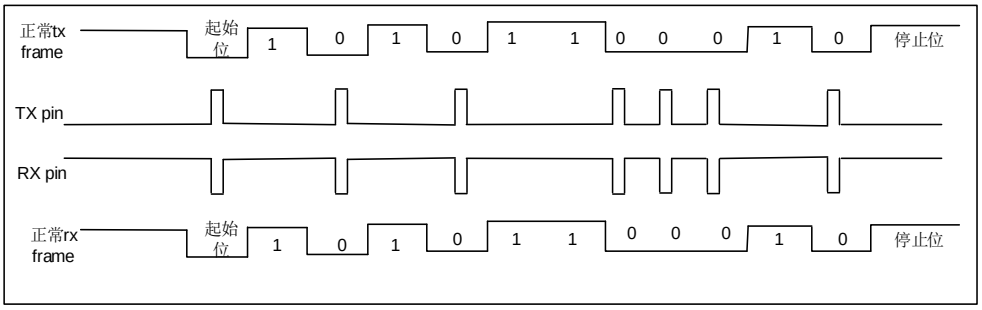
图 16-13. IrDA SIR ENDEC 模块



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 16-14. 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。

16.3.11. 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，RX 引脚不再使用。TX

引脚应被配置为开漏模式，通信冲突由软件处理。当 TEN 被置位时，在数据寄存器中的数据将会被发送。

16.3.12. 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 16-15. 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。

16.3.13. 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。

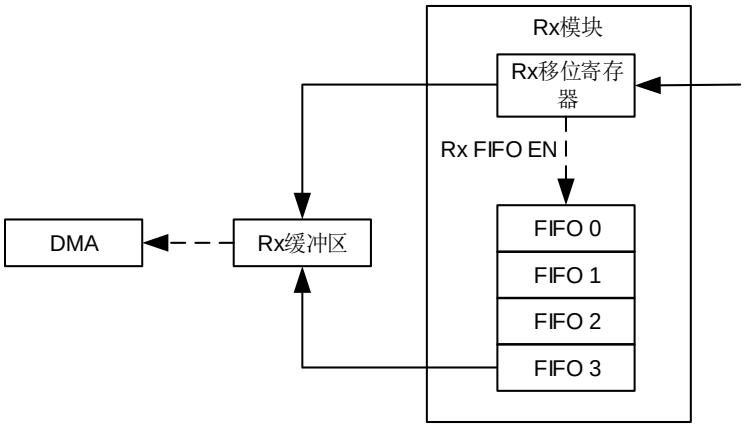
在 ModBus/ASCII 模式下，块尾被认为是一个特定的字符（CR/LF）串。USART 用字符匹配

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。

16.3.14. 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 16-16. 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。

16.3.15. 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 RCU 部分）。

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。

16.3.16. USART 中断

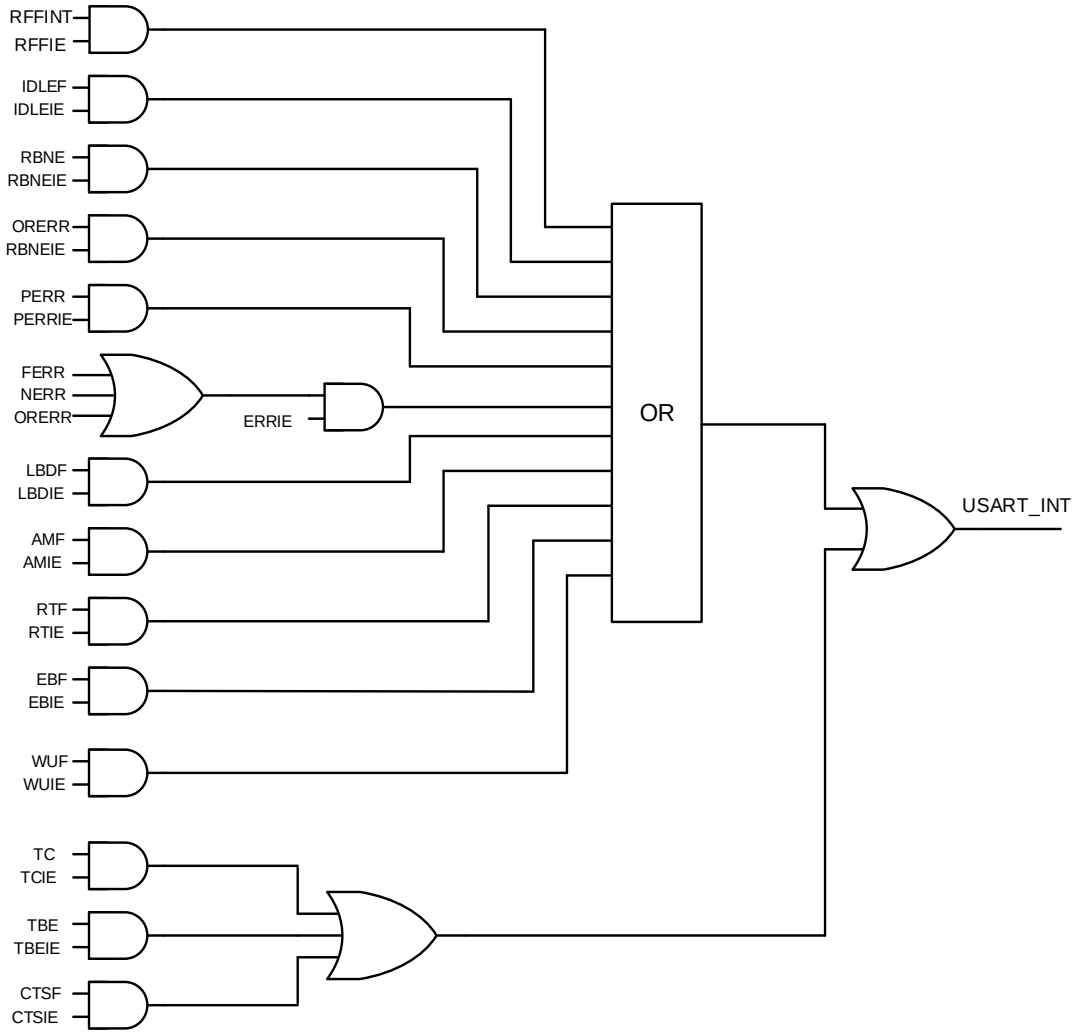
USART 中断事件和标志如*表 16-3. USART 中断请求*所示：

表 16-3. 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
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
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 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 16-17. USART 中断映射框图



16.4. USART 寄存器

USART0 基地址：0x4000 4800

UART1 基地址：0x4000 4400

UART2 基地址：0x4001 1000

16.4.1. 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UART1和UART2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UART1和UART2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UART1和UART2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

16.4.2. 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匹配的检测。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UART1和UART2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UART1和UART2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UART1和UART2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UART1和UART2中，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UART1和UART2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。

16.4.3. 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UART1和UART2中，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UART1和UART2中，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UART1和UART2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	在接收错误时屏蔽DMA请求 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。在发生接收错误时，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，必须首先失能DMA接收（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UART1和UART2中，该位保留。

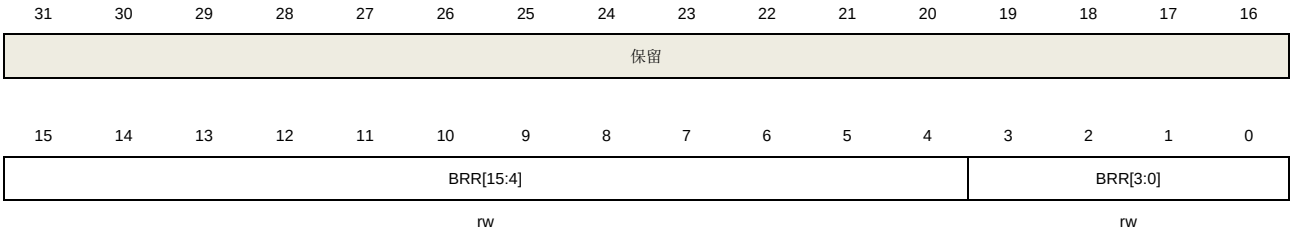
4	NKEN	智能卡模式NACK使能 0：当出现校验错误时不发送NACK 1：当出现校验错误时发送NACK USART被使能（UEN=1）时，该位域不能被改写。 在UART1和UART2中，该位保留。
3	HDEN	半双工使能 0：禁用半双工模式 1：开启半双工模式 USART被使能（UEN=1）时，该位域不能被改写。
2	IRLP	IrDA低功耗模式 0：正常模式 1：低功耗模式 USART被使能（UEN=1）时，该位域不能被改写。
1	IREN	IrDA模式使能 0：IrDA禁用 1：IrDA被使能 USART被使能（UEN=1）时，该位域不能被改写。 在UART1和UART2中，该位保留。
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。

16.4.4. 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[11:0] = 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。

16.4.5. 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 UART1 和 UART2 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 ...

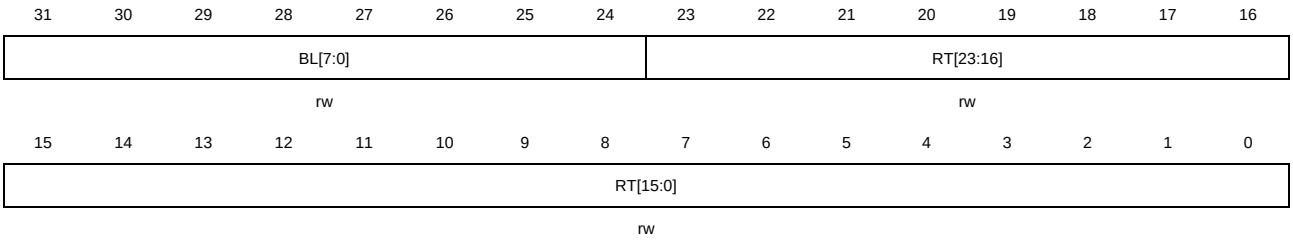
USART被使能（UEN=1）时，该位域不能被改写。

16.4.6. 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 UART1 和 UART2 中，该寄存器保留。

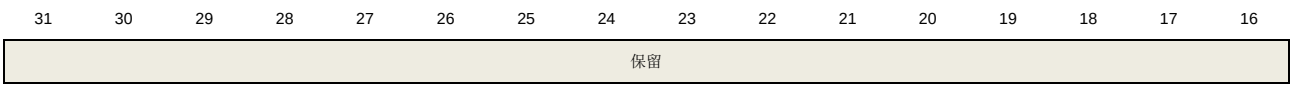


位/位域	名称	描述
31:24	BL[7:0]	块长度 这些位给出了智能卡T=1的接收时块的长度。它的值等于信息字节的长度+结束部分的长度（1-LEC/2-CRC）–1。 这个值可以在块接收开始时设置（用于需要从块的序言提取块的长度的情形），这个只在每一个接收时钟周期只能设置一次。在智能卡模式下，当TBE=0时，块的长度计数器被清0。 在其他模式下，当REN=0（禁用接收器）并且/或者当EBC位被写1时块的长度计数器被清0。
23:0	RT[23:0]	接收器超时门限 该位域指定接收超时值，单位是波特时钟的时长 标准模式下，如果在最后一个字节接收后，在RT规定的时长内，没有检测到新的起始位，RTF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。

16.4.7. 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。



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留											TXFCMD	RXFCMD	MMCMD	SBKCMD	保留
											w	w	w	w	

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UART1和UART2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SBF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。

16.4.8. 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UART1和UART2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UART1和UART2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UART1和UART2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UART1和UART2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。

16.4.9. 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UART1和UART2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UART1和UART2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UART1和UART2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UART1和UART2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。

16.4.10. 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。

16.4.11. 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。

16.4.12. 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

16.4.13. 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UART1和UART2中，该位保留。

17. 内部集成电路总线接口（I2C）

17.1. 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。

17.2. 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）。
- 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。
- I2C0 地址匹配时，由深度睡眠模式唤醒。
- 独立于 PCLK 的时钟。

17.3. 功能说明

I2C 接口的内部结构如*图 17-1. I2C 模块框图*所示。

图 17-1. I2C 模块框图

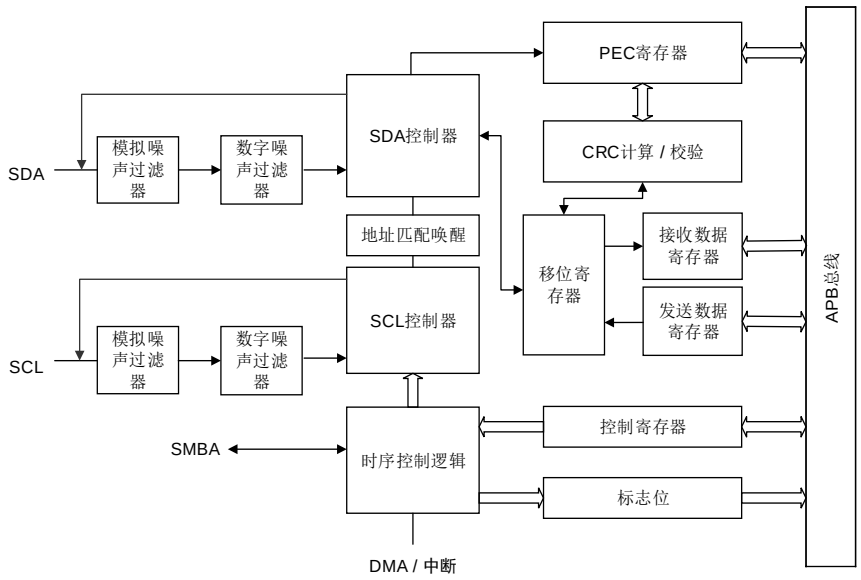


表 17-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

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

17.3.1. 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。

17.3.2. I2C 通讯流程

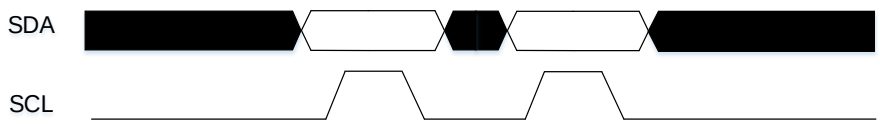
主机和从机都能实现数据收发，因此，I2C 可以实现四种工作模式：

- 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 17-2. 数据有效性*）。每个数据比特传输需要一个时钟脉冲。

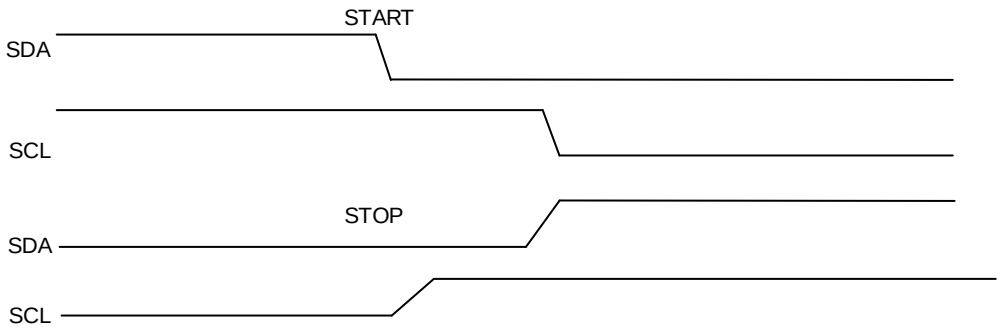
图 17-2. 数据有效性



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 17-3. 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 17-3. 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 17-4. 10 位地址的 I2C 通讯流程（主机发送）

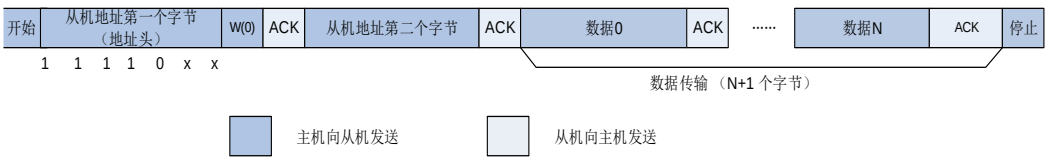


图 17-5. 7 位地址的 I2C 通讯流程（主机发送）

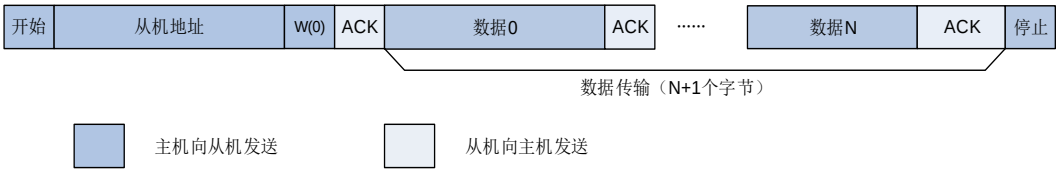
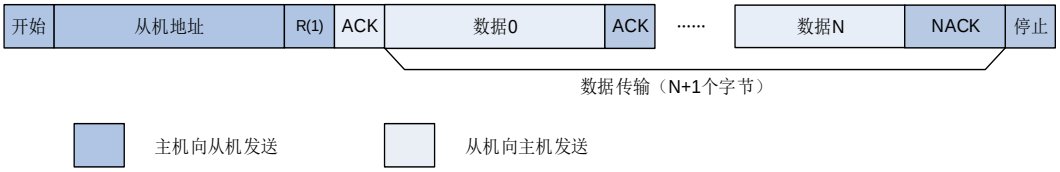


图 17-6. 7 位地址的 I2C 通讯流程（主机接收）



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 17-7. 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 17-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 1）*所示。

图 17-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 0）

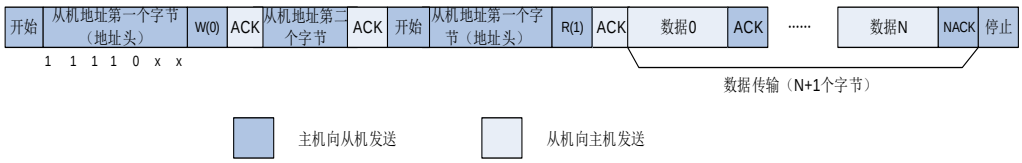
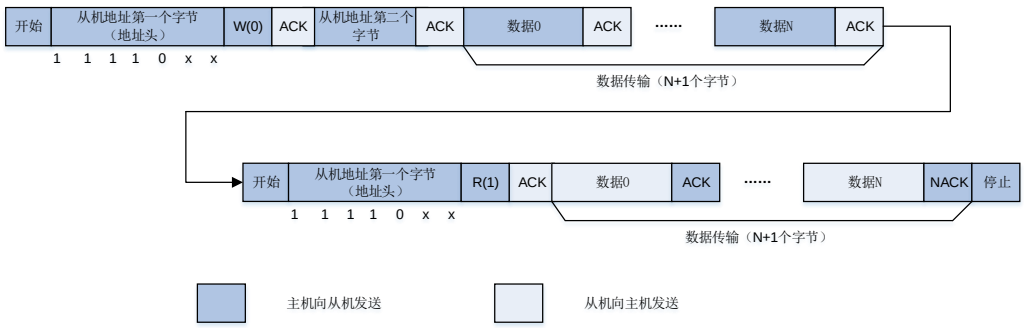


图 17-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R = 1）



17.3.3. 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。

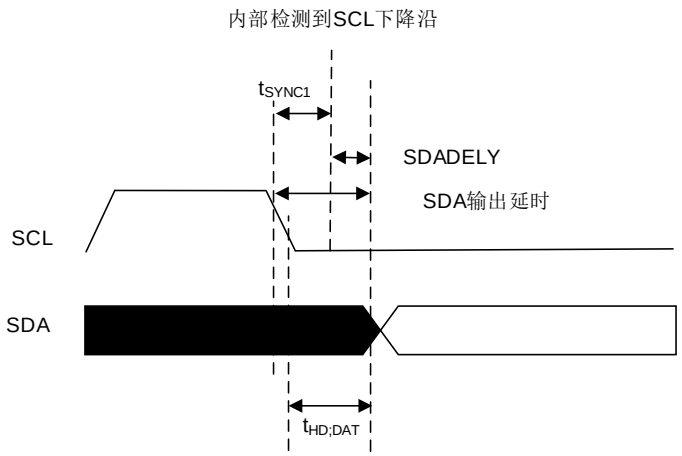
数字滤波器由 I2C_CTL0 寄存器中 DNF[3:0]位来配置。当数字滤波器使能时，SCL 和 SDA 电平保持稳定的时间大于DNF[3:0]×t_{I2CCLK}才会发生内部变化。抑制峰值宽度可由 DNF[3:0]配置。

17.3.4. 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。

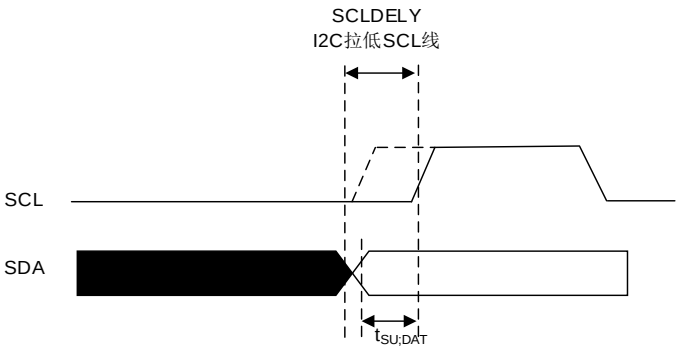
如果数据已经在 I2C_TDATA 寄存器中，在经历 SDADELY 延时后，数据由 SDA 发送，如*图 17-9. 数据保持时间*所示。

图 17-9. 数据保持时间



当数据经过 SDA 发送时，SCLDELY 计数器开启。如图*图 17-10. 数据建立时间*所示。

图 17-10. 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 17-2. 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	

17.3.5. 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：

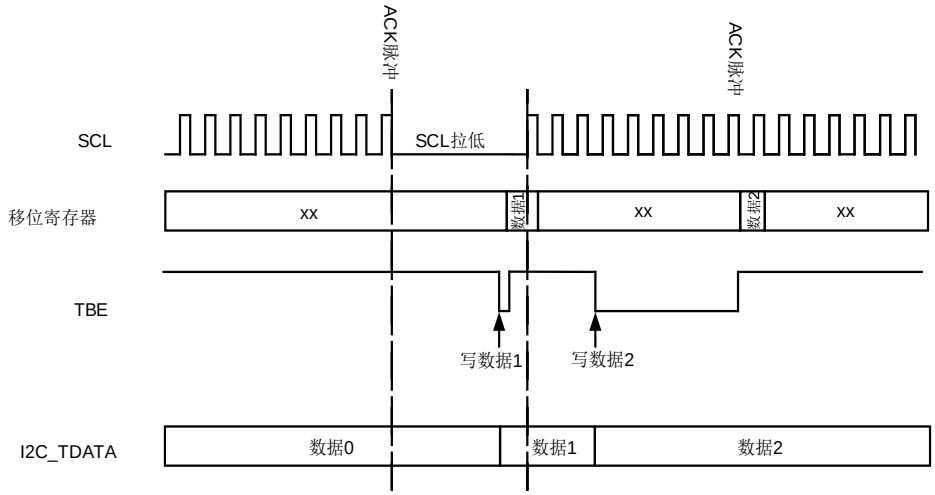
- I2CEN 写 0
- 检查 I2CEN 是否为 0
- I2CEN 写 1

17.3.6. 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。

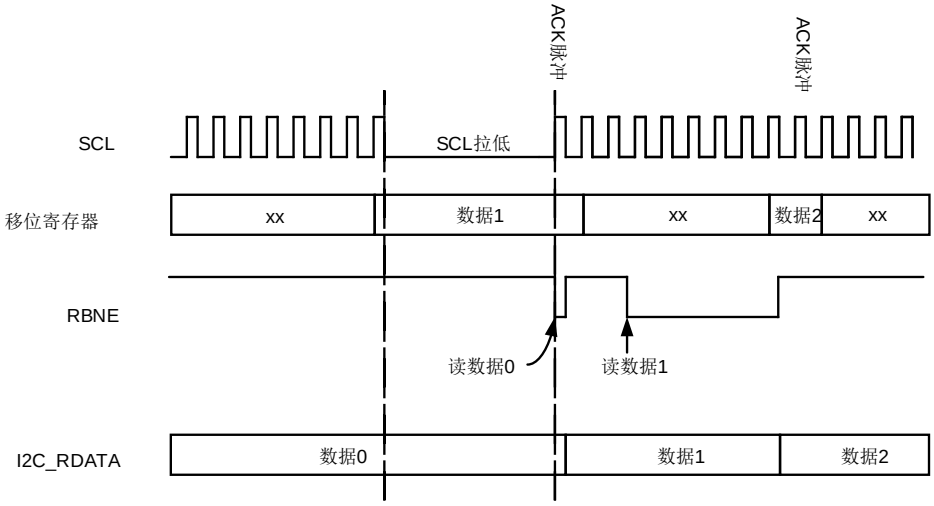
图 17-11. 数据发送



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 17-12. 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 17-3. 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 17-3. 可关闭通信模式

工作模式	行为
主机模式	产生 NACK，STOP 和 RESTART
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。

17.3.7. 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。

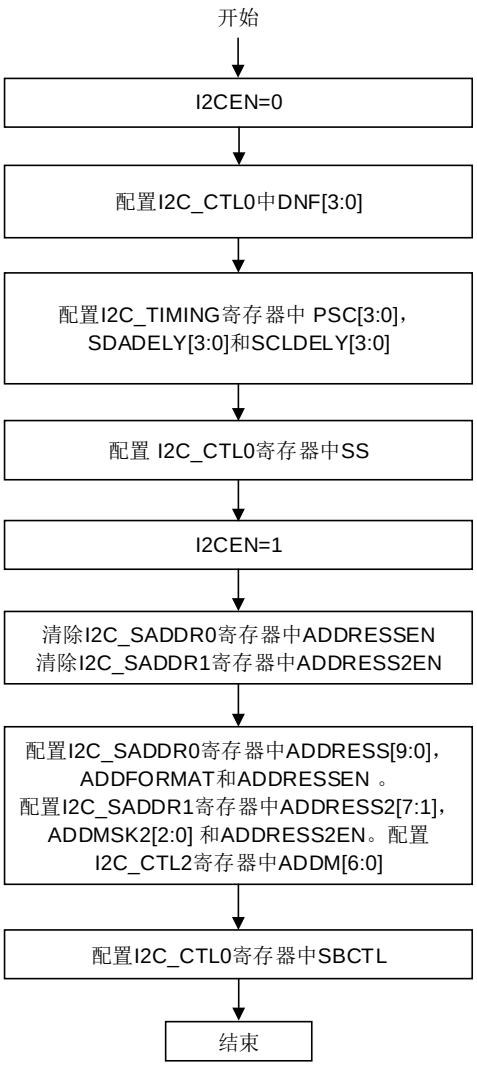
当 BYTENUM[7:0]大于 0x1 时，在 BYTENUM[7:0]数据接收期间，数据流是连续的。

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 17-13. 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 17-14. I2C 从机发送编程模型（SS = 0）

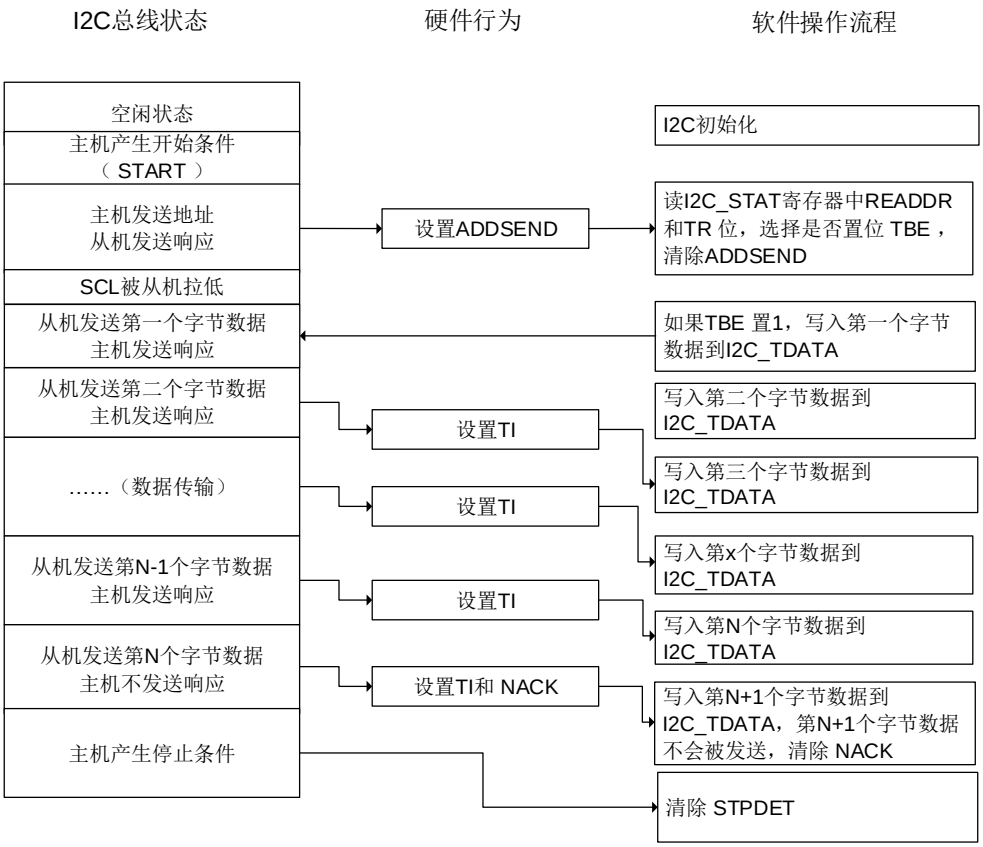
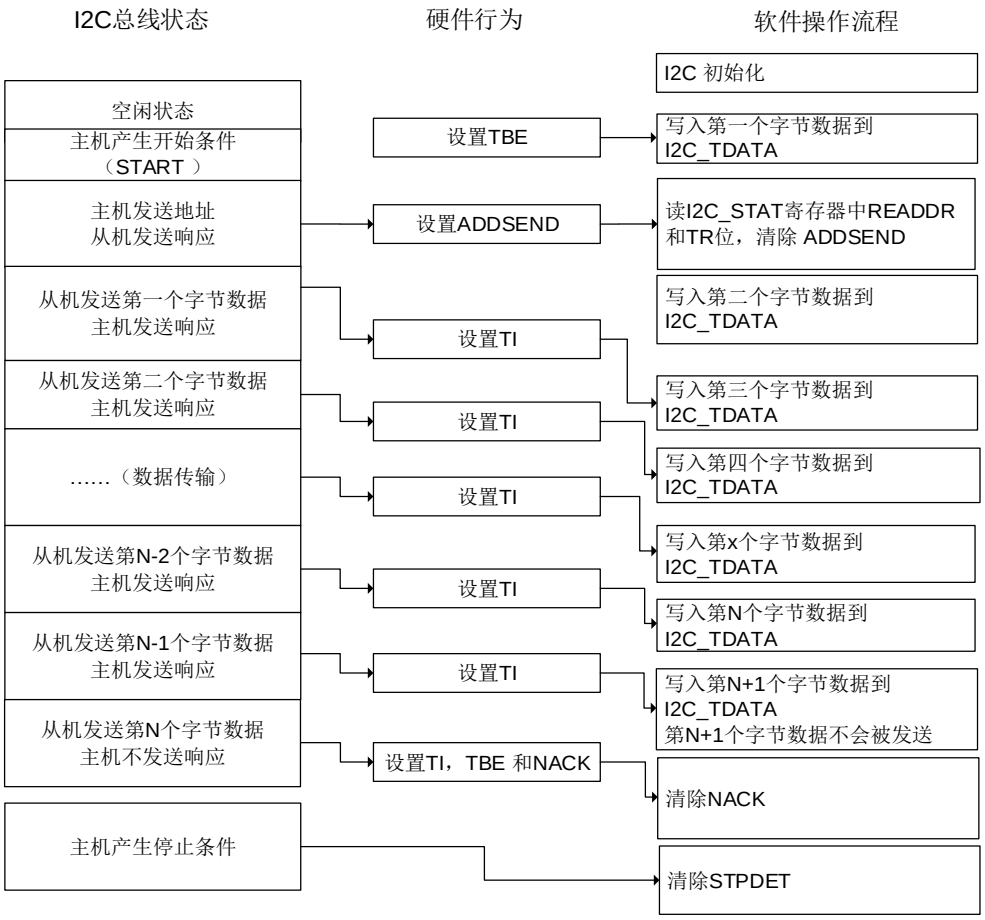


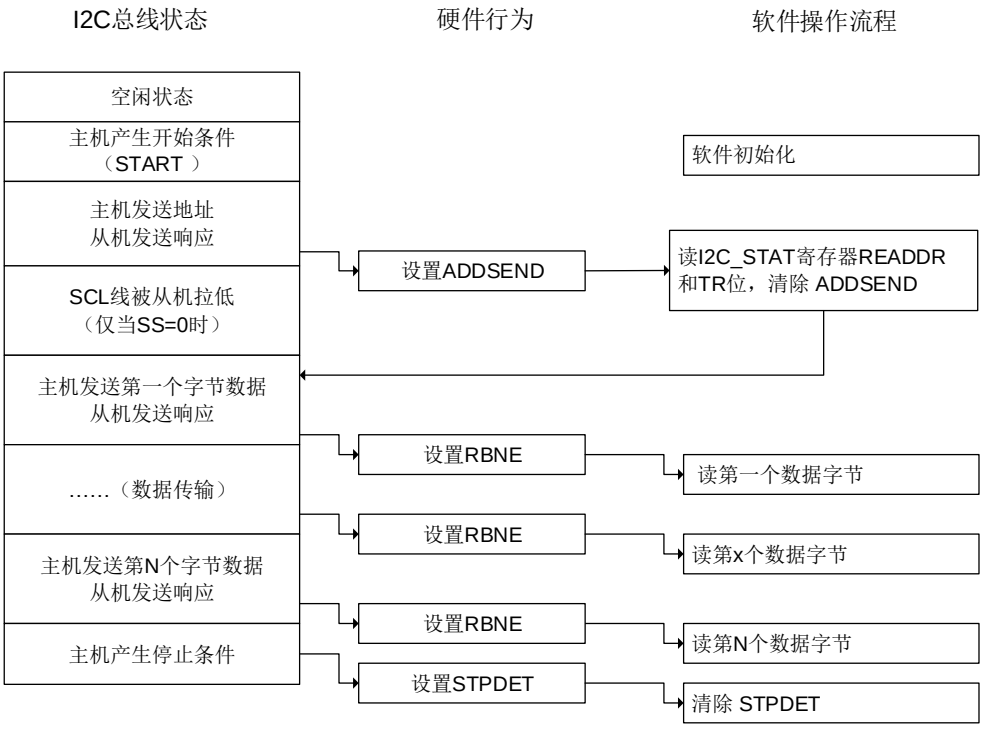
图 17-15. I2C 从机发送编程模型（SS = 1）



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 17-16. I2C 从机接收编程模型



17.3.8. 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$DNF[3:0]*t_{I2CCLK}$。

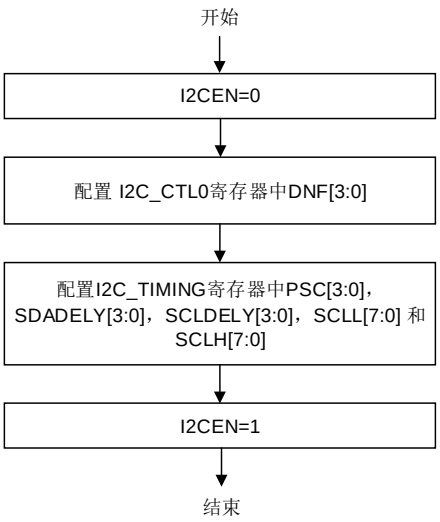
在主机模式下，必须配置 I2C_CTL1 寄存器中 ADD10EN，SADDRESS[9:0]以及 TRDIR 位。当在主机接收模式下使用 10 位寻址时，必须配置 HEAD10R 来选择是执行完整的地址寻址序列，还是只发送地址头。待传输的字节数在 I2C_CTL1 寄存器 BYTENUM[7:0]配置。如果待传输的字节数大于或者等于 255，必须将 BYTENUM[7:0]配置为 0xFF。然后主机发送 START 信号。以上提到的所有位必须在 START 位置 1 之前配置。START 信号发送完成之后，待 I2C_STAT

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 17-17. 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_CTL1 寄存器 BYTENUM[7:0]。如果发送字节数大于 255，必须通过将 I2C_CTL1 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 17-18. I2C 主机发送编程模型（N<=255）

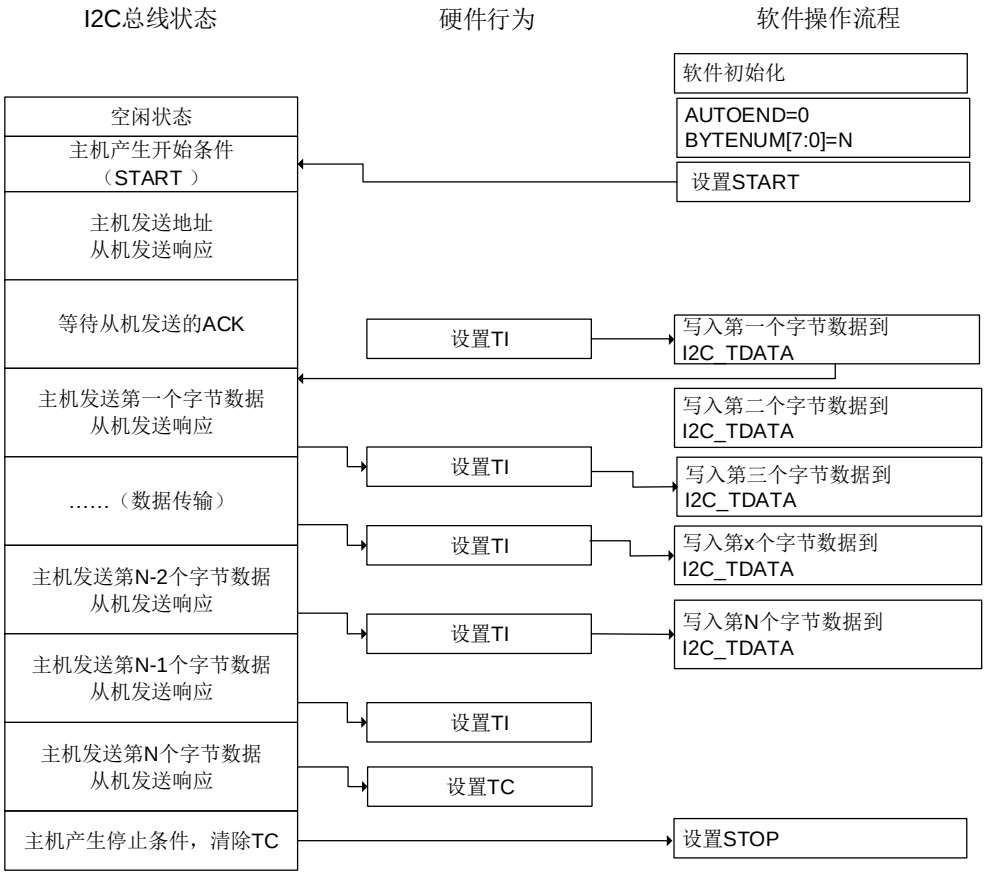
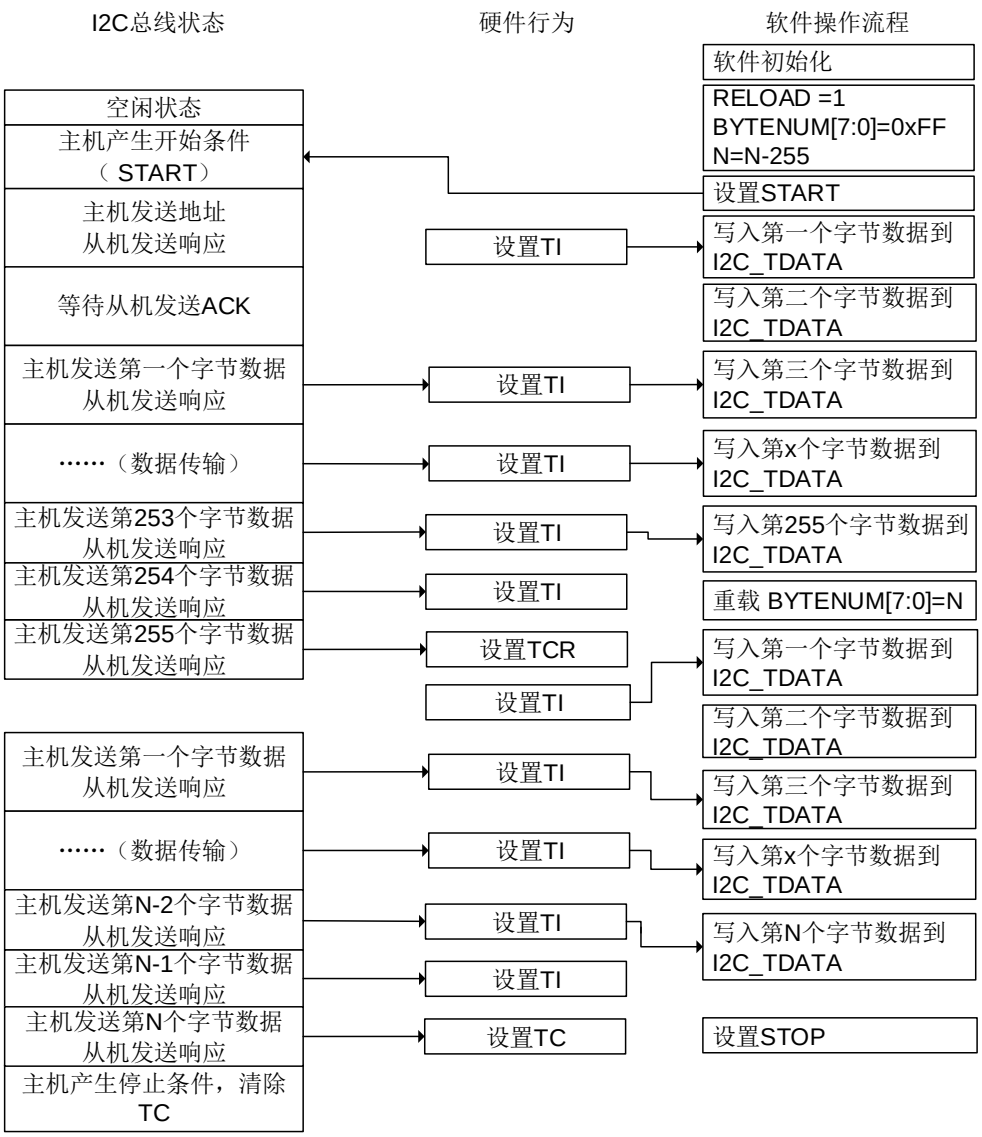


图 17-19. I2C 主机发送编程模型（N>255）



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_CTL1 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 17-20. I2C 主机接收编程模型（N<=255）

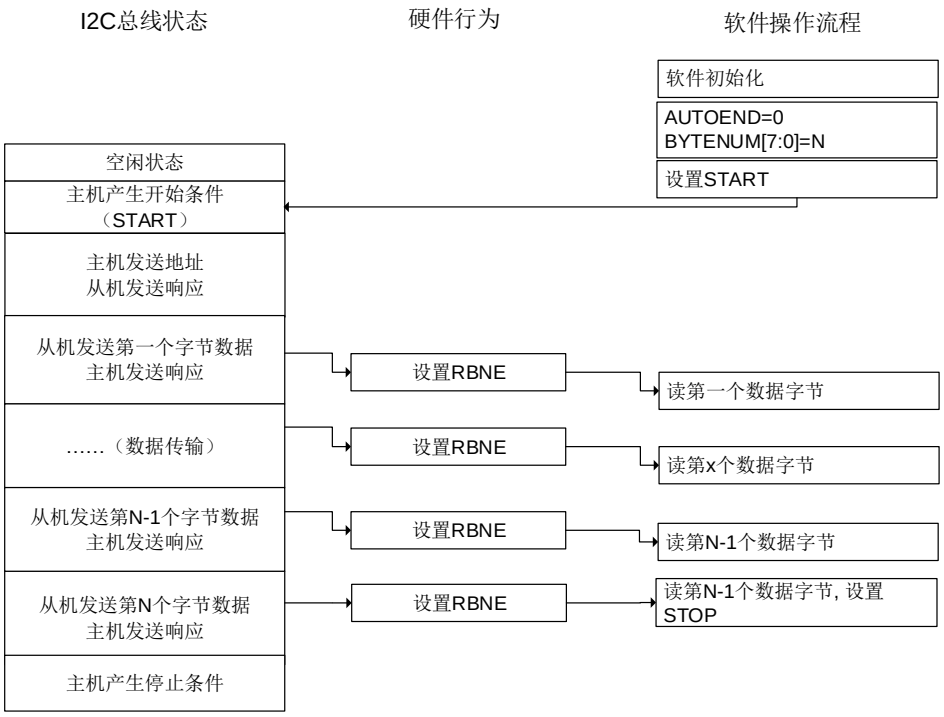


图 17-21. I2C 主机接收编程模型（N>255）

17.3.9. 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）。

17.3.10. SMBus 模式

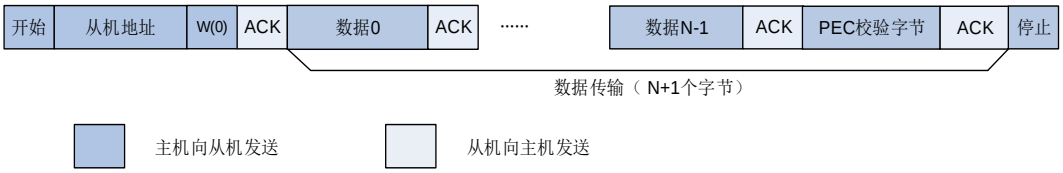
SMBus 主机发送器和从机接收器

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 17-22. 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

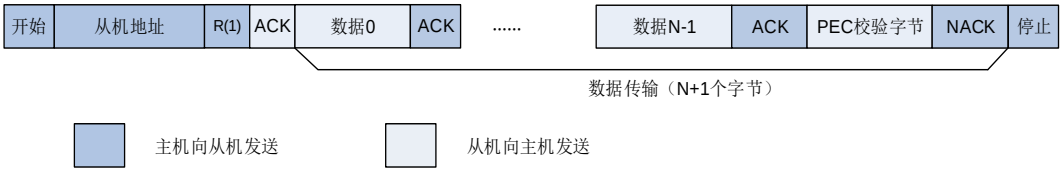
如果 SMBus 主机需要在数据传输完成后接收 PEC 字节，可以使能自动结束模式。在 START 信号发送之前，必须将 PECTRANS 位置 1，且配置好从机地址。在接收 BYTENUM-1 数据之后，接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在停止信号发送之前，接收 PEC 字节之后会给出 NACK 响应。

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 17-23. SMBus 主机接收器和从机发送器通信流程



17.3.11. 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）。

注意：只有 I2C0 的地址匹配能将 MCU 从 Deep-sleep 模式唤醒。

17.3.12. 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。

17.3.13. I2C 错误和中断

I2C 错误标志如*表 17-4. I2C 错误标志*所示。

表 17-4. I2C 错误标志

I2C 错误名称	描述
BERR	总线错误
LOSTARB	仲裁丢失

I2C 错误名称	描述
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 17-5. I2C 中断事件*所示。

表 17-5. 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	

17.3.14. I2C 调试模式

当为控制器进入调试模式（RISC-V 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。

17.4. I2C 寄存器

I2C0 基地址：0x4000 5400

I2C1 基地址：0x4000 5800

17.4.1. 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。 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。在 I2C1 中，该位保留。
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	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。

17.4.2. 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。

17.4.3. 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。

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
ADDRESSEN	保留				ADDFORMAT	ADDRESS[9:8]		ADDRESS[7:1]							ADDRESS0
rw					rw	rw		rw							rw

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。

17.4.4. 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:1]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。

17.4.5. 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。

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

17.4.6. 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。

17.4.7. 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。

17.4.8. 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			

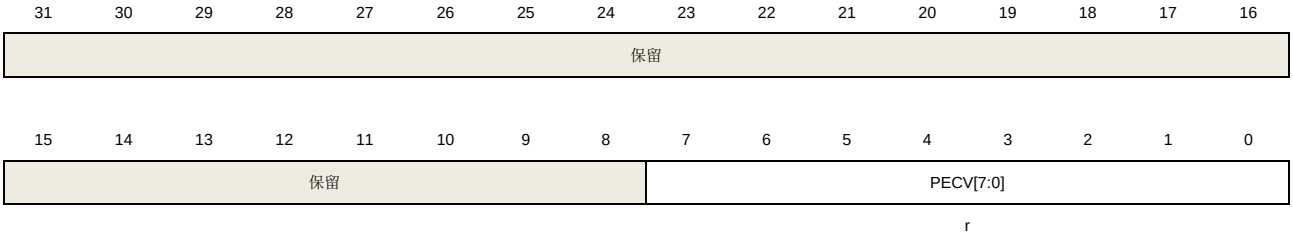
位/位域	名称	描述
31:14	保留	必须保持复位值。
13	SMBALTC	SMBus 报警标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 SMBALT 位。
12	TIMEOUTC	TIMEOUT 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 TIMEOUT 位。
11	PECERRC	PEC 错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 PECERR 位。
10	OUERRC	上溢 / 下溢标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 OUERR 位。

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。

17.4.9. 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。

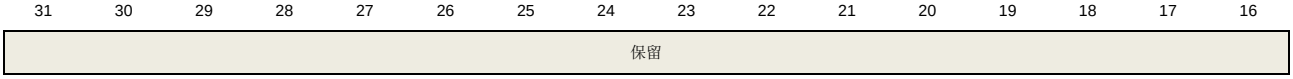


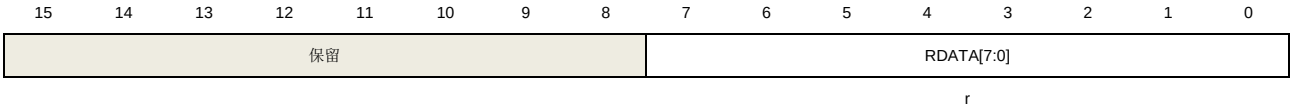
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	PECV[7:0]	在 PEC 使能时，由硬件计算出来的 PEC 值。 当 I2CEN = 0 时，PECV 由硬件清零。

17.4.10. 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。



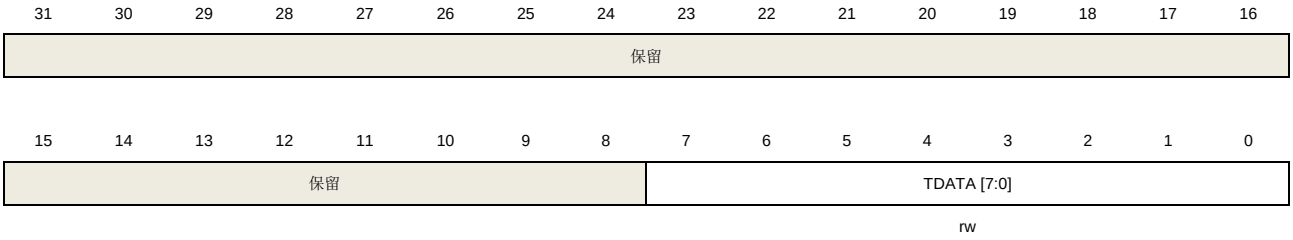


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	RDATA[7:0]	接收到的数据

17.4.11. 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。

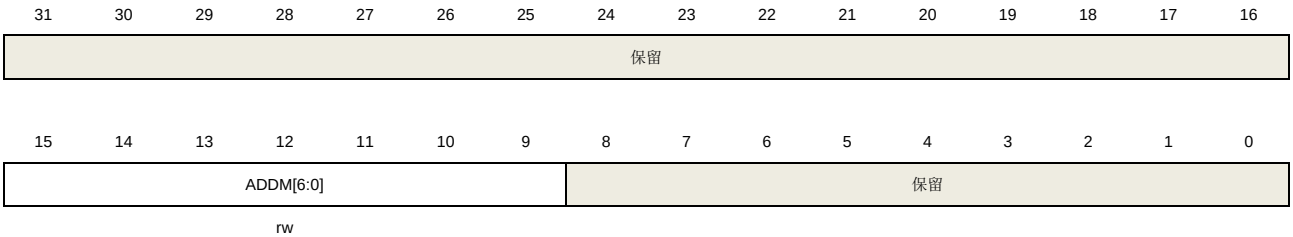


位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TDATA[7:0]	发送的数据

17.4.12. 控制寄存器 2（I2C_CTL2）

地址偏移：0x90
复位值：0x0000 FE00

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。

18. 串行外设接口（SPI）

18.1. 简介

SPI 模块可以通过 SPI 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。

18.2. 主要特征

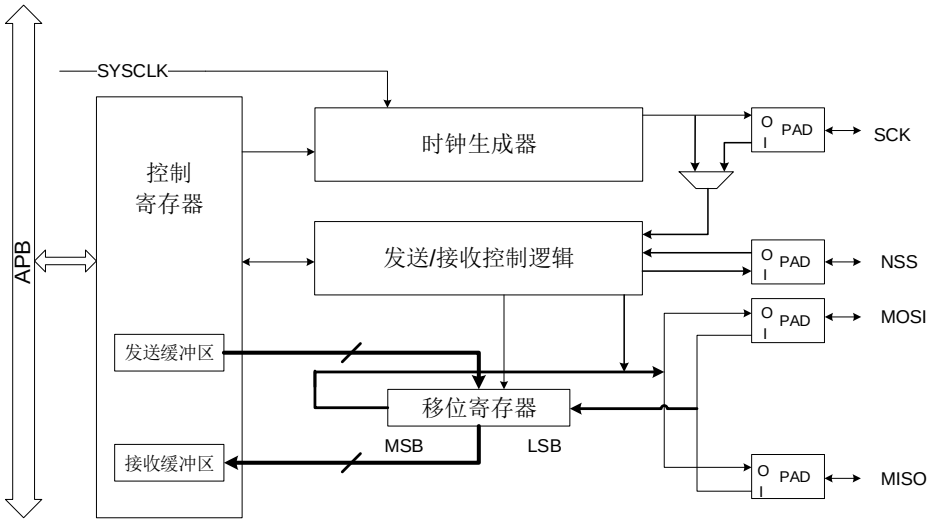
18.2.1. SPI 主要特征

- 具有全双工、半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；
- 8位或16位数据帧格式；
- 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；

18.3. SPI 功能说明

18.3.1. SPI 结构框图

图 18-1. SPI 结构框图



18.3.2. SPI 信号线描述

表 18-1. SPI 信号描述

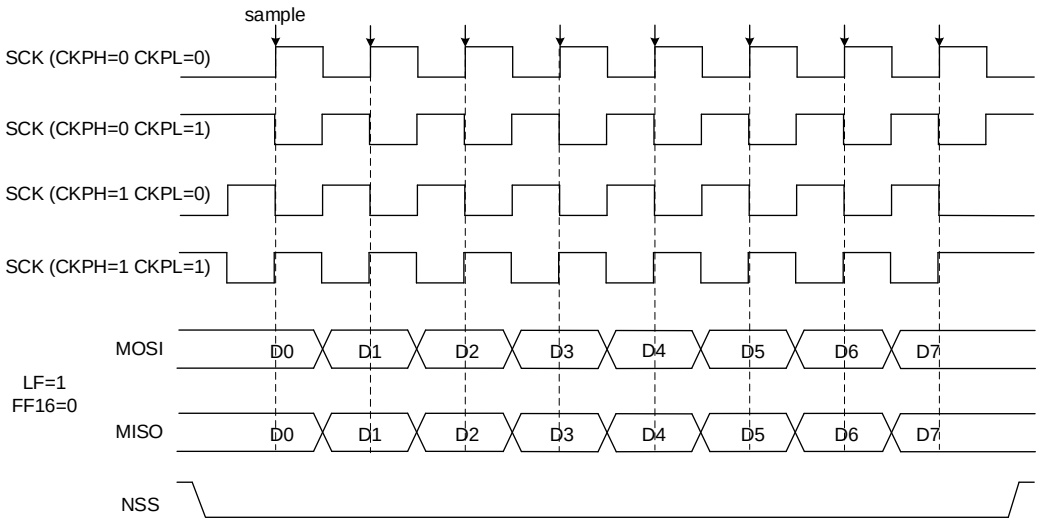
引脚名称	方向	描述
SCK	I/O	主机：SPI 时钟输出 从机：SPI 时钟输入
MISO	I/O	主机：数据接收线 从机：数据发送线 主机双向线模式：不使用 从机双向线模式：数据发送和接收线
MOSI	I/O	主机：数据发送线 从机：数据接收线 主机双向线模式：数据发送和接收线 从机双向线模式：不使用
NSS	I/O	软件 NSS 模式：不使用 主机硬件 NSS 模式：NSSDRV=1 时，为 NSS 输出，适用于单主机模式；NSSDRV=0 时，为 NSS 输入，适用于多主机模式 从机硬件 NSS 模式：为 NSS 输入，作为从机的片选信号。

18.3.3. 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。

图 18-2. 常规模式下的 SPI 时序图



在常规模式中，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，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。

18.3.4. 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 18-2. 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 18-3. 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

18.3.5. SPI 运行模式

表 18-4. 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	MOSI：不使用 MISO：接收

模式	描述	寄存器配置	数据引脚用法
		BDOEN：不要求	
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 RO = 0 BDEN = 0 BDOEN：不要求	MOSI：接收 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 18-3. 典型的全双工模式连接

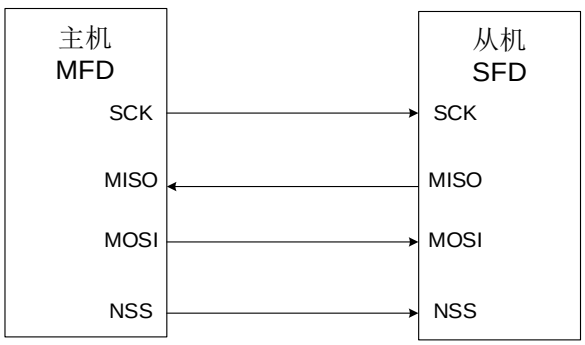


图 18-4. 典型的单工模式连接（主机：接收，从机：发送）

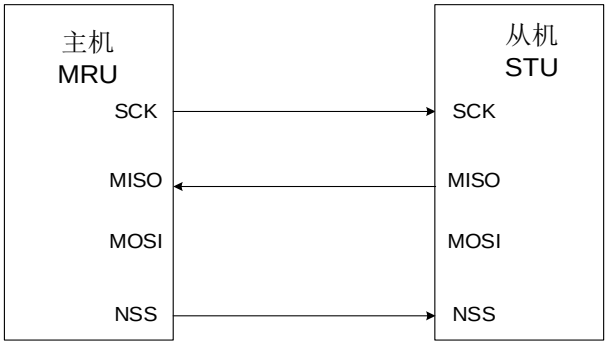


图 18-5. 典型的单工模式连接（主机：只发送，从机：接收）

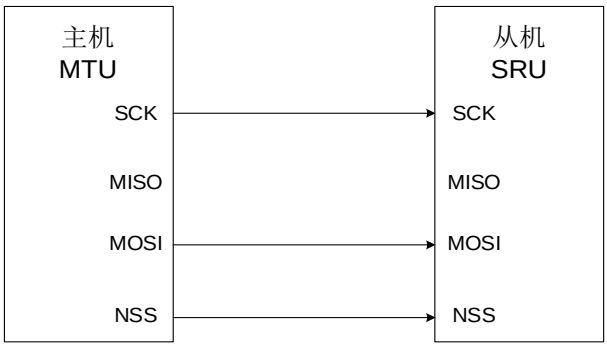
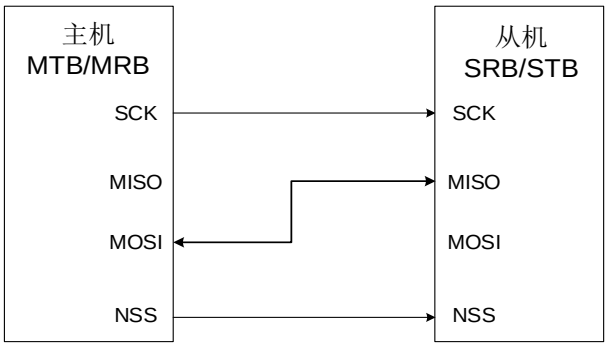


图 18-6. 典型的双向线连接



SPI 初始化流程

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. 根据*表18-4. SPI运行模式*，配置MSTMOD位、RO位、BDEN位和BDOEN位。
8. 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 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（发送缓冲区空）位置 1。TBE 标志位置 1，说明发送缓冲区为空，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，且 RBNE（接收缓冲区非空）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 TI 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 18-7. 主机 TI 模式在不连续发送时的时序图

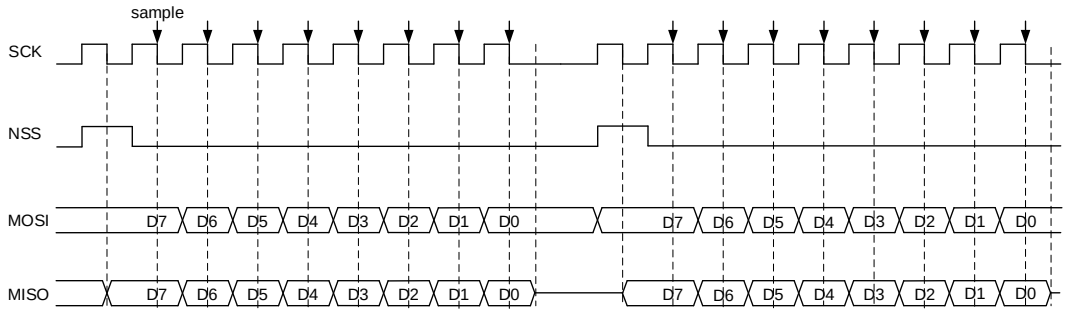
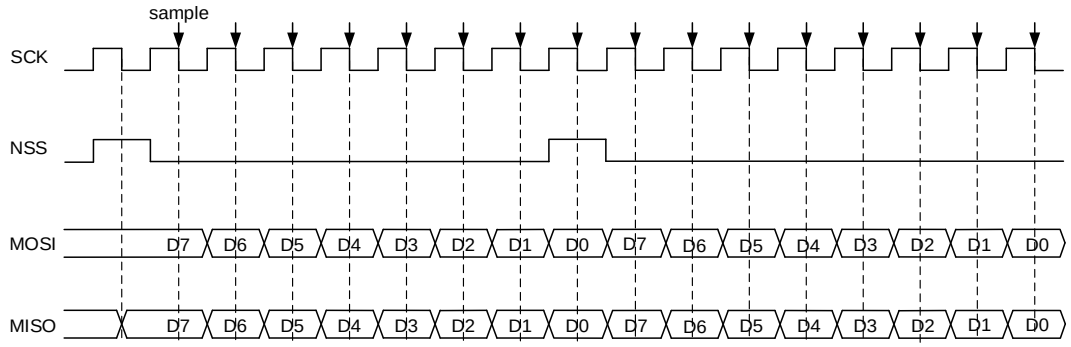
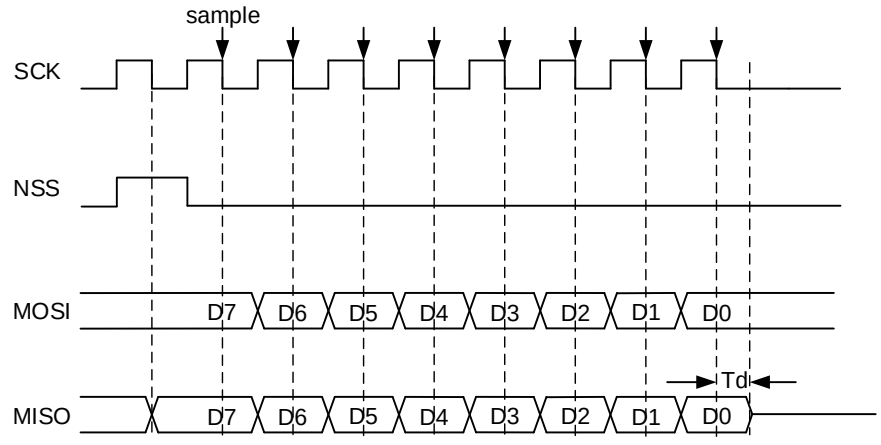


图 18-8. 主机 TI 模式在连续发送时的时序图



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 18-9. 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。

$$T_d=\frac{T_{bit}}{2}+5*T_{pclk} \quad (18\text{-}1)$$

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。

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

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

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

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

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。

18.3.6. 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。

18.3.7. 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。

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。

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。

18.3.8. 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。

- **接收缓冲区非空标志位（RBNE）**

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。这说明，上一帧数据还未被读出而新的数据已经接收了。接收缓冲区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接收到的 CRC 值进行比较，当两者不同时，CRCERR 位将会置 1。

表 18-5. SPI 中断请求

中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区空	写SPI_DATA寄存器	TBEIE
RBNE	接收缓冲区非空	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
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	

18.4. SPI 寄存器

SPI 基地址：0x4001 3000

18.4.1. 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。

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	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 0：8位数据帧格式 1：16位数据帧格式
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/2　100：PCLK/32 001：PCLK/4　101：PCLK/64 010：PCLK/8　110：PCLK/128 011：PCLK/16　111：PCLK/256 PCLK=PCLK2。
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时钟跳变沿采集第一个数据

18.4.2. 控制寄存器 1（SPI_CTL1）

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。

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								TBEIE	RBNEIE	ERRIE	TMOD	保留	NSSDRV	DMATEN	DMAREN
								rw	rw	rw	rw		rw	rw	rw

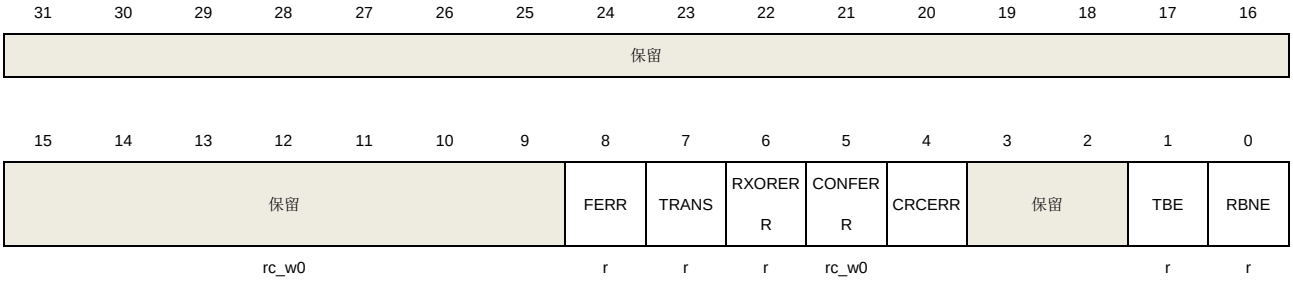
位/位域	名称	描述
31:8	保留	必须保持复位值
7	TBEIE	发送缓冲区空中断使能 0：TBE中断禁止 1：TBE中断使能。当TBE置位时，产生中断。
6	RBNEIE	接收缓冲区非空中断使能 0：RBNE中断禁止. 1：RBNE中断使能。当RBNE置位时，产生中断。
5	ERRIE	错误中断使能 0：错误中断禁止 1：错误中断使能。当CRCERR位，CONFERR位，RXORERR位置1时，产生中断。
4	TMOD	SPI TI模式使能 0：SPI TI模式禁止 1：SPI TI模式使能
3	保留	必须保持复位值
2	NSSDRV	NSS输出使能 0：NSS输出禁止 1：NSS输出使能。 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DMA使能 0：发送缓冲区DMA禁止 1：发送缓冲区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DMA使能 0：接收缓冲区DMA禁止 1：接收缓冲区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。

18.4.3. 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。



位/位域	名称	描述
31:9	保留	必须保持复位值
8	FERR	帧错误 SPI TI模式： 0：没有TI模式帧错误发生 1：TI模式帧错误发生 该位由硬件置位，可以通过写0清除。
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：先读或写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。

3:2	保留	必须保持复位值
1	TBE	发送缓冲区空 0：发送缓冲区非空 1：发送缓冲区空
0	RBNE	接收缓冲区非空 0：接收缓冲区空 1：接收缓冲区非空

18.4.4. 数据寄存器（SPI_DATA）

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。

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 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。

18.4.5. 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]															

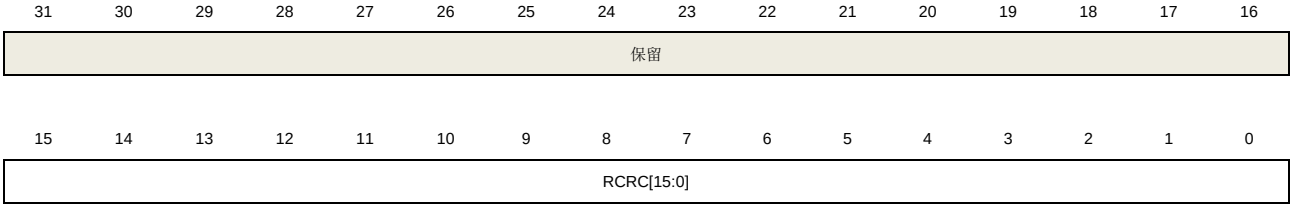
rw

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。

18.4.6. 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。

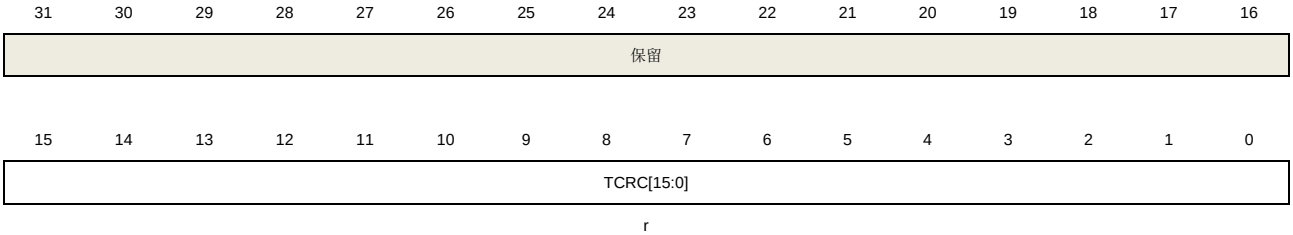


位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	RCRC[15:0]	接收CRC寄存器值 当SPI_CTL0中的CRCEN置位时，硬件计算接收数据的CRC值，并保存到RCRC寄存器中。如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到RCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到RCRC[15:0]。硬件在接收到每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。 当SPI_CTL0寄存器中的CRCEN位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。

18.4.7. 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。



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。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]。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。

19. 四线 SPI 接口（QSPI）

19.1. 简介

QSPI 是一种专用于和 Flash 存储器通信的接口，可以支持单线，双线，四线 SPI FLASH。它可以运行在普通模式、读轮询模式和内存映射模式。

19.2. 主要特征

- 三种模式：普通模式（外部地址），读轮询模式和内存映射模式；
- 可用于普通模式和内存映射模式的完全可编程的命令格式；
- 集成用于接收和发送的FIFO；
- 允许8位、16位或32位数据访问；
- 普通模式支持DMA操作。

19.3. QSPI 功能描述

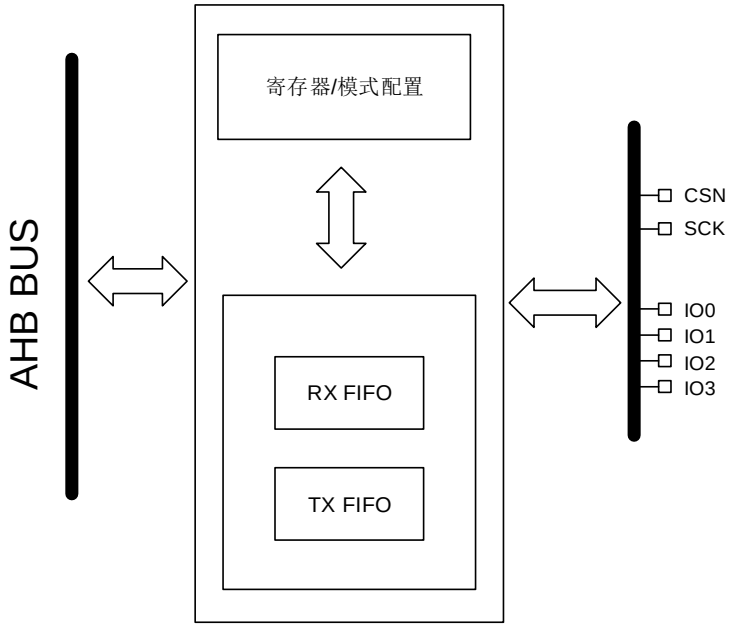
19.3.1. QSPI 结构框图

QSPI 使用 6 根信号线和外部 flash 存储器连接，描述如*表 19-1. QSPI 信号线描述*所示。

表 19-1. QSPI 信号线描述

引脚	方向	描述
CSN	O	片选输出（低电平有效）
SCK	O	时钟输出
IO0 / SO	I / O	单线模式：数据输出 双线模式：数据输入或输出 四线模式：数据输入或输出
IO1 / SI	I / O	单线模式：数据输入 双线模式：数据输入或输出 四线模式：数据输入或输出
IO2	I / O	单线模式：连接 flash 的 WP 引脚，控制“写保护”功能 双线模式：连接 flash 的 WP 引脚，控制“写保护”功能 四线模式：数据输入或输出
IO3	I / O	单线模式：连接 flash 的 HOLD 引脚，控制“保持”功能 双线模式：连接 flash 的 HOLD 引脚，控制“保持”功能 四线模式：数据输入或输出

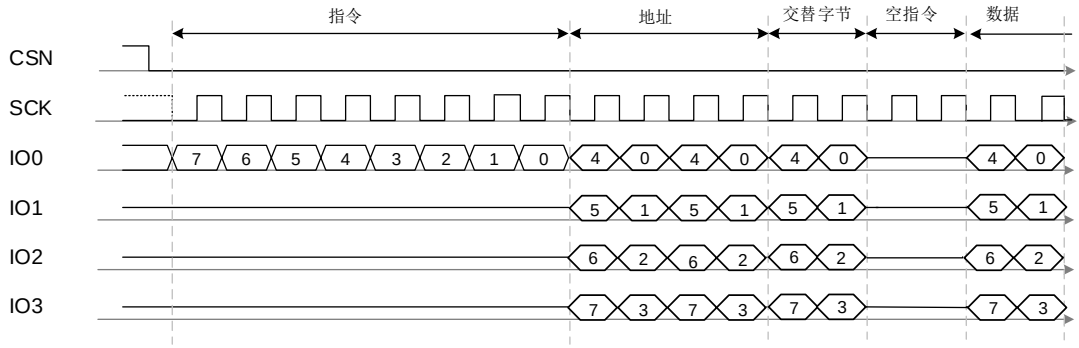
图 19-1. QSPI 结构框图



19.3.2. QSPI 命令格式

QSPI 使用不同格式的命令与 flash 存储器通信。一共最多有五个阶段：指令阶段、地址阶段、交替字节阶段、空指令阶段、数据阶段。任一阶段都可以跳过，但是至少需要包含指令阶段、地址阶段、交替字节、数据阶段的其中一个阶段，这是由软件保证，硬件设计没有任何保护方法。另外，命令的高位始终占用高位信号线。

图 19-2. QSPI 命令格式



命令和相应的配置如*表 19-1. QSPI 信号线描述*所示。

表 19-2. QSPI 命令描述

命令	发送信息	配置	注意
指令	8位指令	QSPI_TCFG寄存器定义指令和信号线模式	-
地址	1-4字节地址	QSPI_ADDR寄存器定义地址信	-

命令	发送信息	配置	注意
		息。QSPI_TCFG寄存器定义发送地址的字节数和信号线模式	
交替字节	1-4交替字节	QSPI_ALTE 寄存器定义交替字节信息，QSPI_TCFG 寄存器定义交替字节的个数和信号线模式	-
空闲	0-31周期	QSPI_TCFG 寄存器定义周期，DATAMOD 位域（QSPI_TCFG 寄存器）定义空闲信号线模式	这期间与外部存储器没有数据交互，等待外部存储器，准备数据。
数据	任意数量的字节	在普通模式下，QSPI_DTLEN 寄存器定义字节数。DATAMOD 位域（QSPI_TCFG 寄存器）定义数据信号线模式，DATAMOD = 00 的配置只能在普通写模式下使用。	在内存映射模式下，传输的字节个数确定 AHB 总线的访问操作，可以 8 位，16 位或者 32 位读写访问，相应传输 1 个，2 个，4 个字节。

注意：信号线模式可以为无指令，单线，双线或者四线。

19.3.3. QSPI 信号线的模式

对于 QSPI 的信号线模式，指令阶段、地址阶段、字节交替阶段、数据阶段都可以通过设置 IMOD / ADDRMOD / ALTEMOD / DATAMOD 位域进行独立配置。

表 19-3. QSPI 信号线模式

<table>
<tr><th colspan="2">信号线模式</th><th>单线模式</th><th>双线模式</th><th>四线模式</th></tr>
<tr><td rowspan="4">配置位域</td><td>IMOD</td><td rowspan="4">01或者00</td><td rowspan="4">10或者00</td><td rowspan="4">11或者00</td></tr>
<tr><td>ADDRMOD</td></tr>
<tr><td>ALTEMOD</td></tr>
<tr><td>DATAMOD</td></tr>
<tr><td rowspan="4">引脚</td><td>IO0（SO）</td><td>输出</td><td rowspan="2">输入：数据阶段读操作（高阻状态）
输出：所有其他阶段</td><td rowspan="3">输入：数据阶段读操作（高阻状态）
输出：所有其他阶段</td></tr>
<tr><td>IO1（SI）</td><td>输入（高阻状态）</td></tr>
<tr><td>IO2</td><td colspan="2">输出0（禁止“写保护”功能）</td></tr>
<tr><td>IO3</td><td colspan="2">输出1（禁止“保持”功能）</td></tr>
<tr><td colspan="2">描述</td><td>DATAMOD = 01时，空闲阶段IO0输出，IO1输入（高阻状态）。</td><td>DATAMOD = 10 时，空闲阶段 IO0 / IO1 一直高阻状态。</td><td>DATAMOD = 11时，空闲阶段IO0 / IO1 / IO2 / IO3 一直高阻状态。</td></tr>
</table>

IO2 / IO3 仅用于四线模式，如果五个阶段都没有配置为四线模式，IO2 / IO3 引脚被释放，即使 QSPI 被使能也可以用于其他功能。

19.3.4. CSN 和 SCK

CSN 默认为高电平，它在命令开始时拉低，结束时拉高。

SCK 是从内部 sck 信号输出的一个门信号，内部 sck 信号是一直存在的。

CSN 在第一个 SCK 有效上升沿的之前一个 SCK 时钟周期拉低，在最后一个 SCK 有效上升沿之后一个 SCK 时钟周期拉高。

当 FIFO 在写操作时为空或者读操作时为满，SCK 会停止并且保持低电平直到 FIFO 可以再次工作。在这时，如果 CSN 为高电平，SCK 会在 CSN 上升沿之后的半个 SCK 时钟周期拉高电平。

19.4. 操作模式

QSPI 可以工作在普通模式、读轮询模式和内存映射模式。在普通模式下，所有的操作都依赖于 QSPI 寄存器。在读轮询模式下，定时读取并检查外部闪存存储器中的状态寄存器值。在内存映射模式下，外部闪存被映射到微控制器地址空间（范围从 0x9000 0000 到 0x97FF FFFF），并作为内部存储器访问。

19.4.1. 普通模式

普通模式写操作是通过将 QSPI_TCFG 寄存器中的 FMOD[1:0]为“00”来选择的。待传输的数据写入 QSPI_DATA。通过将 QSPI_TCFG 寄存器中的 FMOD[1:0]设置为“01”，选择普通模式读操作，接收的数据从 QSPI_DATA 读取。

QSPI_DTLEN 寄存器中的 DTLEN[31:0]定义了待传输的字节数。如果 DTLEN 是 0xFFFF FFFF，则认为传输的字节数是未定义的，在传输字节数达到 QSPI_DCFG 寄存器中 FMSZ[4:0]规定的存储器大小边界时停止传输。如果 DTLEN 为 0xFFFF FFFF 且 FMSZ[4:0]被配置为 0x1F 时，闪存存储器容量为 4GB，传输会一直持续到发生中止请求或 QSPI 被禁用。

当传输数据的字节数达到 DTLEN 寄存器中设定的值时，传输完成标志 TC 会被置 1。如果 DTLEN 为 0xFFFF FFFF，则发送 / 接收字节数等于 FMSZ[4:0]规定的外部存储器大小时，TC 会被置 1。如果 TCIE 和 TC 都被置 1，则会产生中断，通过将 QSPI_STATC 寄存器的 TCC 位置 1 可以清除 TC 位。

初始化一个命令序列

命令序列在根据通信需求配置好最后信息之后立即开始。

当没有地址并且没有数据时，在访问 QSPI_TCFG 寄存器之后立即开始命令序列。

当存在地址但没有数据时，在访问 QSPI_ADDR 寄存器之后立即开始命令序列。

当在普通模式写操作时需要地址并且有数据，在访问 QSPI_DATA 寄存器之后立即开始命令序列。

FIFO

16 字节的 FIFO 用于传输数据。在普通模式写操作时，AHB 写访问方式与 FIFO 增加的字节数的关系如*表 19-4. AHB 写访问方式与 FIFO 增加的字节数的关系*所示。

表 19-4. AHB 写访问方式与 FIFO 增加的字节数的关系

AHB 写访问方式	FIFO 增加的字节数
32 位	4 字节
16 位	2 字节
8 位	1 字节

注意：当 AHB 写访问模式为 8 位或 16 位时，QSPI_DATA 寄存器中最低有效字节是有效的。

FIFO 阈值由 QSPI_CTL 寄存器中的 FTL[3:0]定义，在普通模式读操作时，当 FIFO 中的字节数等于者超过定义的阈值时，QSPI_STAT 寄存器中的 FIFO 阈值标志 FT 将置 1。在数据阶段完成后如果 FIFO 不为空，FT 也会被置 1。在普通模式写操作时，当 FIFO 的空字节数超过阈值时，FT 会被置 1。

如果 FTIE 和 FT 都被置 1，将产生中断。如果 QSPI DMA 使能，DMA 请求由 FT 产生，直到标志清除。

在普通模式读操作时，当 FIFO 变为满时，QSPI 暂时停止 SCK 以避免溢出。读序列不能恢复直到 FIFO 中有大于等于 4 个字节剩余空间。

19.4.2. 读轮询模式

通过将 FMOD[1:0]配置为“10”可以来选择读轮询模式。在读轮询模式下，QSPI 周期性地启动一个读命令，其中最多包含 4 字节的数据。接收到的数据可以按位屏蔽，并与定义的数据内容进行比较，如果匹配发生，且 RPMFIE 位置 1，将生成一个中断。

读轮询访问序列的启动与普通模式读操作相同。在周期性间隔时 BUSY 位保持高电平。

轮询匹配模式位 RPMM 控制比较匹配模式，如果 RPMM = 0，与模式被选择。在该模式下，只在所有的未屏蔽位都有匹配时，状态匹配标志 RPMF 将置 1。如果 RPMM = 1，或模式被选择。在该模式下，任何非屏蔽位只要有一位匹配，RPMF 将置 1。

在读轮询模式下，如果设置了 RPMS 位，当检测到匹配时，读轮询序列将停止，并在数据阶段结束时清除 BUSY 标志。否则，周期序列将继续，直到 ABORT 位置 1 或 QSPI 被关闭。

在读轮询模式下，FIFO 是避开的，读取的状态字节存放在 QSPI_DATA 中，存储的状态字节不会被 MASK 控制域影响。如果有数据阶段，QSPI_DATA 中内容在数据阶段开始时更新。

如果 FT 位在数据阶段结束时被置位，此时认为外部闪存状态字节已经被读取，读取 QSPI_DATA 清除 FT 位。

19.4.3. 内存映射模式

通过将 FMOD[1:0]配置为“11”，可以选择内存映射模式。在内存映射模式下，外部闪存被认为

是内部存储器来访问，最大访问地址为 128MB，即使外部闪存大于 128MB。即使 FMSZ 定义的地址范围在 128MB 范围内，内存映射模式也不允许地址超过 FMSZ 定义的范围。否则，将生成一个错误。如果 AHB 主机是 CPU，会产生硬故障中断。如果 AHB 主机是 DMA，将产生一个传输错误，并且相应 DMA 通道会关闭。

在该模式下，支持字节、半字、字单次访问或突发访问。

内存映射模式支持顺序访问时的预取功能。QSPI 在访问数据之前会首先在下一个地址加载数据，如果下次访问确实是在下一个地址，那么访问速度会更快，因为数据已经被预取了。否则，将重新启动读取序列。在读取序列开始之前拉低 CSN。

当 FIFO 满时，SCK 停止输出，在此期间 CSN 保持低电平。如果 QSPI_CTL 寄存器中 TMOUTEN 位置 1，当低电平的持续时间达到 QSPI_TMOUT 寄存器中指定的 SCK 时钟周期数时，CSN 将被拉高。

在开始传输时，在 CSN 拉低之前 BUSY 位会变为高电平，在发生超时、中止或者 QSPI 被关闭后变为低电平。

19.5. QSPI 配置

19.5.1. Flash 配置

QSPI_DCFG 寄存器的配置可以用来指定外部闪存存储器的特性，从而使 QSPI 可以持续工作。

QSPI_DCFG 寄存器中 FMSZ[4:0]定义了外部存储器大小，FMSZ+1 是外部存储器的地址位数。在普通模式下，flash 容量最大可达 4GB。

CSHC[2:0]定义了片选高电平时间，它规定了在两个命令序列之间保持高电平最少的 SCK 周期数。

19.5.2. IP 配置

QSPI_CTL 寄存器中的配置可以用来指定 QSPI IP 的特征。

QSPI_CTL 寄存器中 PSC[7:0]定义了时钟分频系数。

SSAMPLE 定义哪个 SCK 边沿用于采样数据。默认情况下，QSPI 在外部存储器驱动数据后的半个 SCK 周期采样数据。然而，因为外部信号的延迟，需要推迟采样数据。采样边沿可以使用 SSAMPLE 移位半个 SCK 周期。

通过在 QSPI_STAT 寄存器中设置 DMAEN 位来启用 DMA 请求。在 QSPI_CTL 寄存器中设置 FTL[3:0]位来配置 FIFO 阈值等级。

19.6. 只发送一次指令

将 QSPI_TCFG 寄存器中 SIOO 位置 1，可以使能只发送一次指令模式，该功能对所有模式有

效。如果 SIOO 置 1，在访问 QSPI_TCFG 后该指令只发送一次，后续命令序列会跳过指令阶段，直到 QSPI_TCFG 再次被访问。

19.7. 错误和中断

当*表 19-5. TERR 和 AHB 错误条件*中一个条件发生时，会产生 TERR 和 AHB 错误。

表 19-5. TERR 和 AHB 错误条件

错误名称	条件
TERR	1. 在普通模式或者读轮询模式下，超出FMSZ规定的地址范围，在QSPI_ADDR寄存器中编写了一个错误的地址。 2. 在普通模式下，地址（ADDR）加数据长度（DTLEN）大于外部内存。
AHB错误	1. 在内存映射模式下，AHB主机执行超出范围访问，或QSPI被关闭。 2. AHB主机正在访问内存映射空间，但内存映射模式未使能。

QSPI 中断事件和标志如*表 19-6. QSPI 中断事件*所示。

表 19-6. QSPI 中断事件

中断事件	事件标志	中断使能位	清除方式
FIFO阈值中断	FT	FTIE	硬件清除
传输完成中断	TC	TCIE	QSPI_STATC寄存器中TCC位置1
传输错误中断	TERR	TERRIE	QSPI_STATC寄存器中TERRC位置1
超时中断	TMOUT	TMOUTIE	QSPI_STATC寄存器中TMOUTC位置1
状态匹配中断	RPMF	RPMFIE	QSPI_STATC寄存器中RPMFC位置1

19.8. QSPI 寄存器

QSPI 基地址：0x4002 5800

19.8.1. 控制寄存器（QSPI_CTL）

地址偏移：0x00

复位值：0x0000 0010

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
PSC[7:0]								RPMM	RPMS	保留	TMOUTIE	RPMFIE	FTIE	TCIE	TERRIE
rw								rw	rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCKDVALUE[3:0]				FTL[3:0]				保留	SCKDEN	SSAMPLE[1:0]		TMOUTEN	DMAEN	ABORT	QSPIEN
rw				rw					rw	rw		rw	rw	w	rw

位 / 位域	名称	描述
31:24	PSC[7:0]	该位域定义了从 AHB 时钟分频产生 QSPI 时钟的分频因子，SCK 频率与 AHB 的关系为$f_{SCK}=f_{AHB}/(PSC+1)$。 0：$f_{SCK}=f_{AHB}$ 1：$f_{SCK}=f_{AHB}/2$ 2：$f_{SCK}=f_{AHB}/3$ ... 255：$f_{SCK}=f_{AHB}/256$ 对于奇数时钟分频因子，时钟的占空比没有50%，时钟信号保持低电平时间要比高电平时间少一个周期。 该位域只能在BUSY = 0时修改。
23	RPMM	读轮询匹配模式 该位表明在读轮询时采用什么方式定义产生匹配 0：与模式，如果flash返回的字节所有非屏蔽位都和QSPI_STATMATCH寄存器相应位匹配，状态匹配标志RPMF被置位。 1：或模式，如果flash返回的字节任何一个非屏蔽位都和QSPI_STATMATCH寄存器相应位匹配，状态匹配标志RPMF被置位。 该位只能在BUSY = 0时修改。
22	RPMS	读轮询模式停止 该位表明在产生匹配后停止读轮询模式 0：在ABORT置位或者禁能QSPI模块时读轮询停止。 1：在产生匹配后读轮询停止。 该位只能在BUSY = 0时修改。

21	保留	必须保持复位值。
20	TMOUTIE	超时中断使能 0：中断禁能 1：中断使能
19	RPMFIE	读轮询模式匹配中断使能 0：中断禁能 1：中断使能
18	FTIE	FIFO阈值中断使能 0：中断禁能 1：中断使能
17	TCIE	传输完成中断使能 0：中断禁能 1：中断使能
16	TERRIE	传输错误中断使能 0：中断禁能 1：中断使能
15:12	SCKDVALUE[3:0]	SCK 延时值 该位域仅在SCKDEN使能并且配置了SSAMPLE时有效。
11:8	FTL[3:0]	FIFO阈值等级 该位域在普通模式下使用，FIFO中的字节数会触发FIFO阈值标志被置位。 普通模式写操作时（FMOD = 00）： 0：FT会被置位，如果有1个或者更多字节可以有效写入FIFO 1：FT会被置位，如果有2个或者更多字节可以有效写入FIFO ... 15：FT会被置位，如果有16个字节可以有效写入FIFO 普通模式读操作时（FMOD = 01）： 0：FT会被置位，如果有1个或者更多有效数据能从FIFO中读取 1：FT会被置位，如果有2个或者更多有效数据能从FIFO中读取 ... 15：FT会被置位，如果有16个有效数据能从FIFO中读取 如果DMAEN为1，在改变FTL之前，DMA控制器的相应通道必须禁能。
7	保留	必须保持复位值。
6	SCKDEN	当从flash读数据时SCK延时使能，仅当采样移位SSAMPLE为1时有效。 0：SCK延时禁能 1：SCK延时使能
5:4	SSAMPLE[1:0]	采样延时 默认情况下，QSPI在flash存储器驱动数据后二分之一个SCK时钟周期采样。考虑到外部信号的延迟，该位域可以配置为允许数据稍后对数据进行采样。

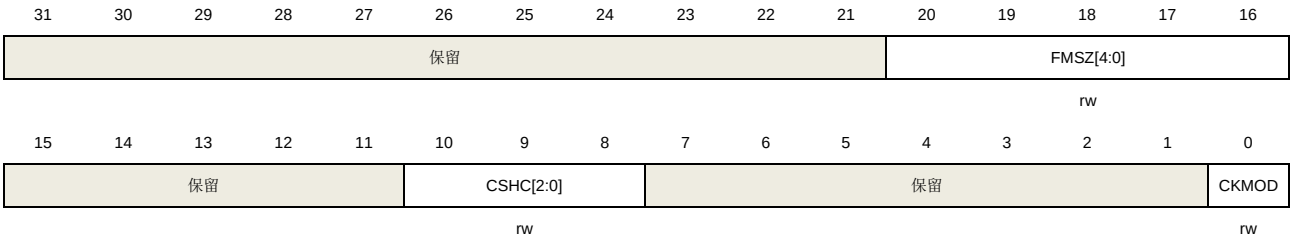
		0：不延时 1：延时半个周期 2：延时一个周期 3：保留 该位只能在BUSY = 0时修改。
3	TMOUTEN	超时计数器使能 在内存映射模式（FMOD=11）下，如果将该位置1，且在TMOUTCYC[15:0]定义的时间之后，没有访问外部flash，片选（CSN）会释放。 0：超时计数器失能，在内存映射模式下访问后片选（CSN）保持低电平 1：超时计数器使能，在内存映射模式下，如果flash在超过TMOUTCYC [15:0]个周期后没有访问外部flash，片选（CSN）会释放。 该位只能在BUSY = 0时修改。
2	DMAEN	DMA使能 普通模式下，可以使用DMA传输数据。当FT位置1时，DMA传输开始。 0：DMA禁能 1：DMA使能
1	ABORT	终止请求 该位停止当前命令，终止请求完成后会自动清除。 读轮询模式或者内存映射模式，如果将该位置1，RPMS位或者DMEN位被清除。 0：无终止请求 1：终止请求
0	QSPIEN	使能QSPI 0：QSPI禁能 1：QSPI使能

19.8.2. 设备配置寄存器（QSPI_DCFG）

地址偏移：0x04

复位值：0x001F 0000

该寄存器只能按字（32 位）访问。



位 / 位域	名称	描述
31:21	保留	必须保持复位值。
20:16	FMSZ[4:0]	flash存储器大小

		该位域定义外部存储器大小： Flash存储器字节数为$2^{[FMSZ+1]}$ FMSZ+1是flash存储器地址位数。普通模式下，flash存储器容量最大到4GB。在内存映射模式下，最大128MB。 该位只能在BUSY = 0时修改。
15:11	保留	必须保持复位值。
10:8	CSHC[2:0]	片选高电平周期数 CSHC+1定义了在两个命令序列之间保持高电平最少的SCK周期数 0：CSN保持高电平至少1个SCK周期 1：CSN保持高电平至少2个SCK周期 ... 7：CSN保持高电平至少8个SCK周期 该位只能在BUSY = 0时修改。
7:1	保留	必须保持复位值。
0	CKMOD	该位表明QSPI空闲时SCK电平 0：当CSN为高时，SCK保持低电平 1：当CSN为高时，SCK保持高电平 该位只能在BUSY = 0时修改。

19.8.3. 状态寄存器（QSPI_STAT）

地址偏移：0x08
复位值：0x0000 0004

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			FL[4:0]					保留		BUSY	TMOUT	RPMF	FT	TC	TERR
			rw							r	r	r	r	r	r

位 / 位域	名称	描述
31:13	保留	必须保持复位值。
12:8	FL[4:0]	FIFO等级 该位域给出FIFO在普通模式下存储的有效字节数。在内存映射模式和读轮询模式下，FL为0。
7:6	保留	必须保持复位值。
5	BUSY	忙状态 flash该位在命令传输时设置。在普通模式下，当一次操作完成后，该位将被清除。如

果在普通读模式下，FIFO也必须为空。

4	TMOUT	超时标志 当TMOUTEN被设置并且在TMOUTCYC[15:0]个周期之后没有访问闪存时，该位置1。
3	RPMF	读轮询匹配标志 在读轮询模式下，当接收到数据匹配QSPI_STATMATCH寄存器中期望值时，该位置1。
2	FT	FIFO阈值标志 在普通模式下，当FIFO阈值到达或者最后一次读操作时FIFO非空，该位置1。在阈值条件不再满足时由硬件清0。 在读轮询模式下，每次从外部flash读取状态寄存器时置位，DATA寄存器被读取时清0。
1	TC	传输完成标志 在普通模式下，当QSPI_DTLEN寄存器中已编程的数据长度以普通模式或终止操作完成时，该位置1。
0	TERR	传输错误标志 在普通模式下，当访问了无效地址时，该位置1。

19.8.4. 状态清除寄存器（QSPI_STATC）

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											TMOUTC	RPMFC	保留	TCC	TERRC
											w	w		w	w

位 / 位域	名称	描述
31:5	保留	必须保持复位值。
4	TMOUTC	清除超时标志 对该位写1清除QSPI_STAT寄存器的TMOUT标志。
3	RPMFC	清除读轮询匹配标志 对该位写1清除QSPI_STAT寄存器的RPMF标志。
2	保留	必须保持复位值。
1	TCC	清除传输完成标志

		对该位写1清除QSPI_STAT寄存器的TC标志。
0	TERRC	清除传输错误标志 对该位写1清除QSPI_STAT寄存器的TERR标志。

19.8.5. 数据长度寄存器（QSPI_DTLEN）

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
DTLEN[31:16]															
rw															

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

位 / 位域	名称	描述
31:0	DTLEN[31:0]	数据长度 该位域指定在普通模式和读轮询模式下要传输的数据的个数n（值+1）。在读轮询模式下，该位域的值应该是一个不大于3（表示4字节）的值。在普通模式下，如果该位域配置为0xFFFF FFFF，表示未定义的长度，QSPI将继续传输数据，直到QSPI_DCFG寄存器中的FMSZ[4:0]定义内存地址结束。 0x0000 0000：待传输字节数为1 0x0000 0001：待传输字节数为2 0x0000 0002：待传输字节数为3 0x0000 0003：待传输字节数为4 ... 0xFFFF FFFD：待传输字节数为4,294,967,294 (4G-2) 0xFFFF FFFE：待传输字节数为4,294,967,295 (4G-1) 0xFFFF FFFF：未定义长度，QSPI_DCFG寄存器中由FMSZ[4:0]定义的所有字节，直到内存结束都将被传输。 如果FMSZ[4:0]为0x1F，将无限地继续读取。 内存映射模式下，该位无影响。 该位只能在BUSY = 0时修改。

19.8.6. 传输配置寄存器（QSPI_TCFG）

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	SIOO	FMOD	DATAMOD[1:0]	保留	DUMYC[4:0]	ALTESZ[1:0]
	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTEMOD[1:0]		ADDRSZ[1:0]		ADDRMOD[1:0]		IMOD[1:0]		INSTRUCTION[7:0]							
rw		rw		rw		rw		rw							

位 / 位域	名称	描述
31:29	保留	必须保持复位值。
28	SIOO	只发送一次指令模式 当IMOD = 00时，该位没有影响。 0：每次命令序列都发送指令 1：命令序列第一次时发送指令 该位只能在BUSY = 0时修改。
27::26	FMOD[1:0]	工作状态 00：普通模式写操作 01：普通模式读操作 10：读轮询模式 11：内存映射模式 如果DMAEN位置1，在改变FMOD位域之前，DMA控制器的相应通道必须关闭。 该位域只能在BUSY = 0时修改。
25:24	DATAMOD[1:0]	数据模式 该位定义数据阶段的操作模式。 00：无数据 01：单线传输数据 10：双线传输数据 11：四线传输数据 该位同时决定空闲阶段操作模式 该位域只能在BUSY = 0时修改。
23	保留	必须保持复位值。
22:18	DUMYC[4:0]	空指令周期数 该位域定义空闲阶段持续时间。 该位域只能在BUSY = 0时修改。
17:16	ALTESZ[1:0]	交替字节大小 00：8位交替字节 01：16位交替字节 10：24位交替字节 11：32位交替字节 该位域只能在BUSY = 0时修改。
15:14	ALTEMOD[1:0]	交替字节模式 00：无交替字节

		01：单线传输交替字节 10：双线传输交替字节 11：四线传输交替字节 该位域只能在BUSY = 0时修改。
13:12	ADDRSZ[1:0]	地址大小 00：8位地址 01：16位地址 10：24位地址 11：32位地址 该位域只能在BUSY = 0时修改。
11:10	ADDRMOD[1:0]	地址模式 00：无地址 01：单线传输地址 10：双线传输地址 11：四线传输地址 该位域只能在BUSY = 0时修改。
9:8	IMOD[1:0]	命令模式 00：无指令 01：单线传输指令 10：双线传输指令 11：四线传输指令 该位域只能在BUSY = 0时修改。
7:0	INSTRUCTION[7:0]	指令 发送到flash存储器的命令信息。 该位域只能在BUSY = 0时修改。

19.8.7. 地址寄存器（QSPI_ADDR）

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
ADDR[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR[15:0]															
rw															

位 / 位域	名称	描述
31:0	ADDR[31:0]	地址 发送到flash存储器的访问地址。

当BUSY=0或在内存映射模式下，对该位域写入值将被忽略。

19.8.8. 交替字节寄存器（QSPI_ALTE）

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
ALTE[31:16]															
rw															

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

位 / 位域	名称	描述
31:0	ALTE[31:0]	交替字节 紧随地址之后，发送给flash存储器的可选数据。 该位只能在BUSY = 0时修改。

19.8.9. 数据寄存器（QSPI_DATA）

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
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]	将要与flash存储器交互的数据。 在普通模式下写操作时，在发送到flash存储器之前，写入到该寄存器数据会被存储到FIFO中。如果FIFO为满，写操作会停止直到FIFO有足够空间。 在普通模式下读操作时，读该寄存器获取从flash存储器接收的数据。如果FIFO没有足够的字节数来满足读命令请求，并且BUSY位为1，那么读操作会被停止直到FIFO中有足够的数据或者传输已经完成。 在读轮询模式下，该寄存器包含从flash读取的最后数据。

19.8.10. 状态屏蔽寄存器（QSPI_STATMK）

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
MASK[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK[15:0]															
rw															

位 / 位域	名称	描述
31:0	MASK[31:0]	读轮询模式下状态屏蔽 读轮询模式下从flash接收的状态字节掩码。 对于MASK[31:0]第n位： 0：接收数据的第n位不参与匹配 1：接收数据的第n位参与匹配 该位域只能在BUSY = 0时修改。

19.8.11. 状态匹配寄存器（QSPI_STATMATCH）

地址偏移：0x28

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
MATCH[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MATCH[15:0]															

位 / 位域	名称	描述
31:0	MATCH[31:0]	读轮询模式下状态匹配 与QSPI_STATMK寄存器中值进行比较匹配的期望值。 该位域只能在BUSY = 0时修改。

19.8.12. 间隔寄存器（QSPI_INTERVAL）

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
INTERVAL[15:0]															
rw															

位 / 位域	名称	描述
31:16	保留	必须保持复位值。
15:0	INTERVAL[15:0]	间隔周期 读轮询模式下两次读命令之间的SCK周期数。 该位域只能在BUSY = 0时修改。

19.8.13. 超时寄存器（QSPI_TMOUT）

地址偏移：0x30

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
TMOUTCYC[15:0]															
rw															

位 / 位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TMOUTCYC[15:0]	超时周期 当内存映射模式，FIFO满时，该位域表明在下次访问到来时片选保持低电平的SCK周期数。 该位域只能在BUSY = 0时修改。

19.8.14. FIFO 刷新寄存器（QSPI_FLUSH）

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
保留															FLUSH
															w

位 / 位域	名称	描述
31:1	保留	必须保持复位值。
0	FLUSH	用于刷新所有内部FIFO

20. 加密处理器（CAU）

20.1. 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。

20.2. 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。

20.3. CAU 数据类型和初始化向量

20.3.1. 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20-1. DATAM不交换 / 半字交换*和*图20-2. 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 20-1. DATAM 不交换 / 半字交换

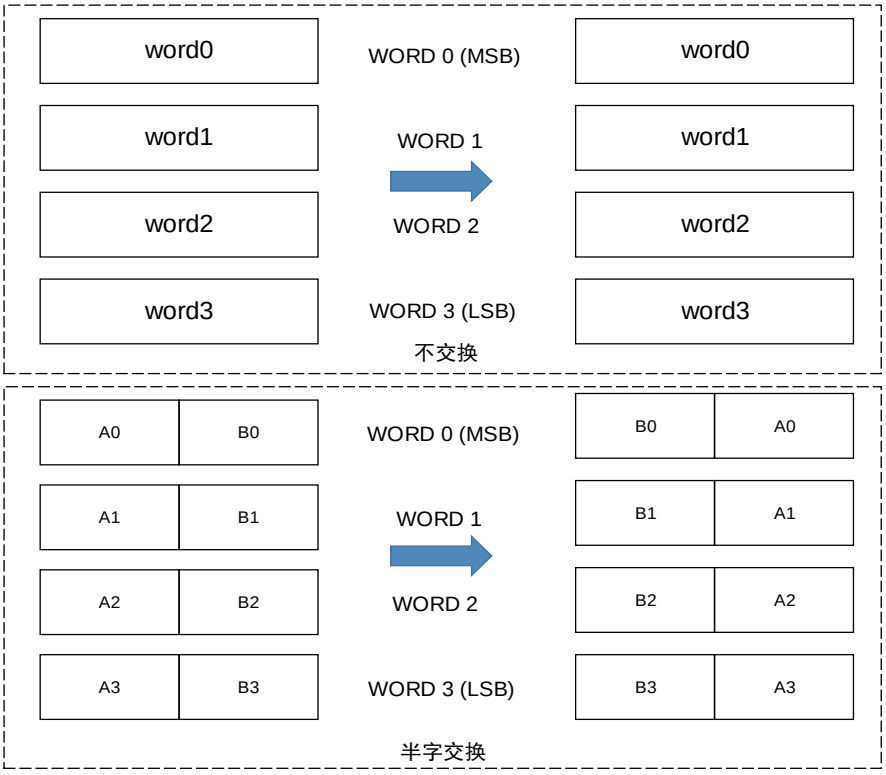
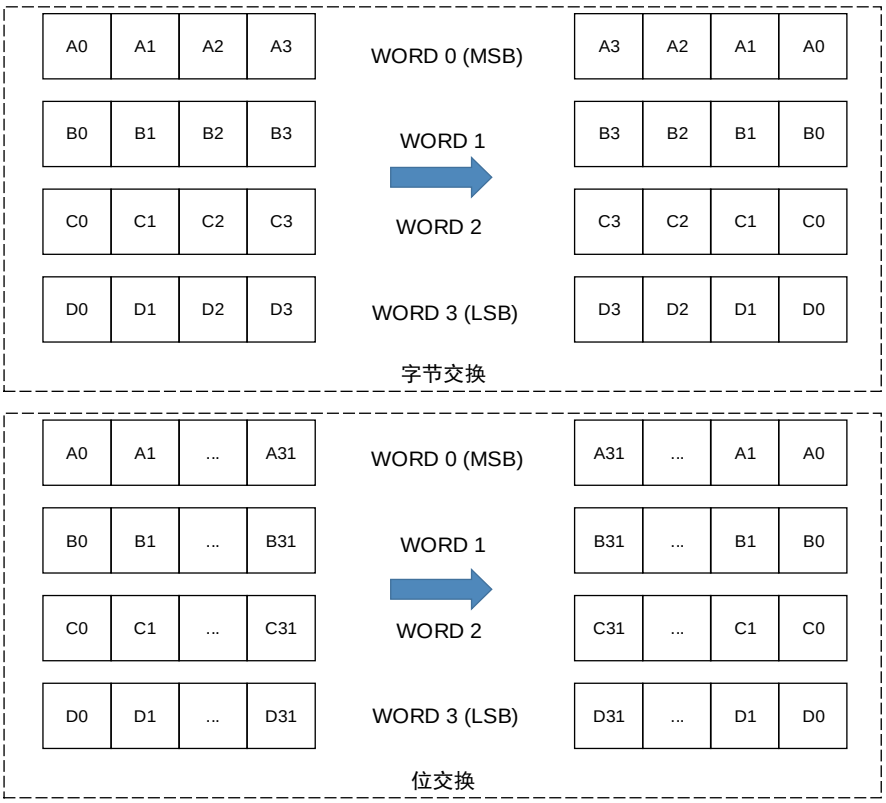


图 20-2. DATATM 字节交换 / 位交换



20.3.2. 初始化向量

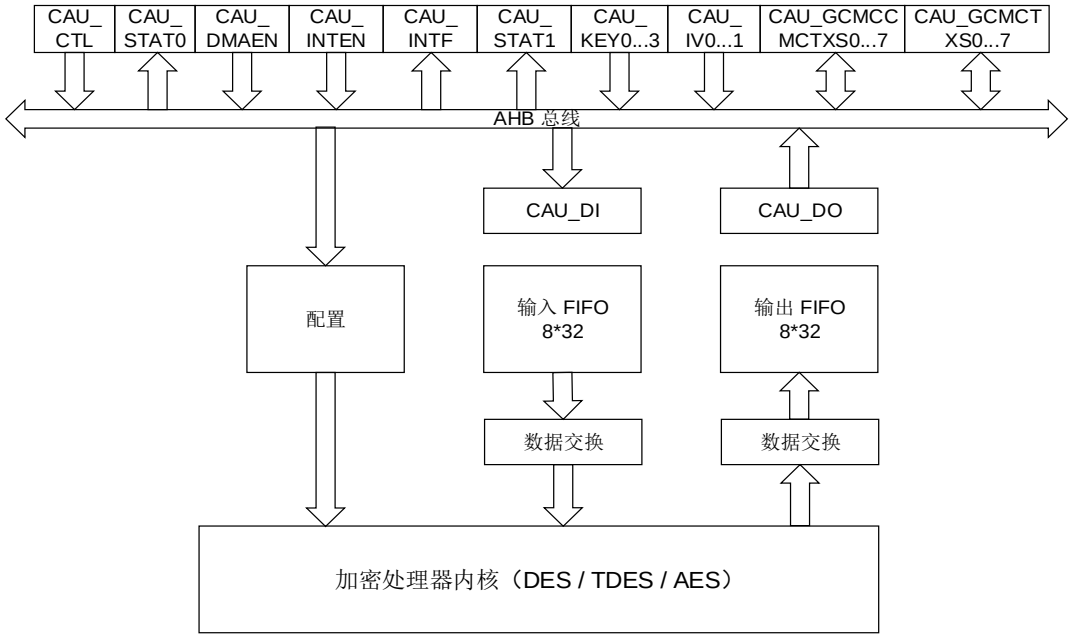
初始化向量用于在 CBC、CTR、GCM、GMAC、CCM、CFB 和 OFB 模式下与数据块进行异或。初始化向量与明文或密码数据无关，而且它们不受 DATAM 值的影响。注意初始化向量寄存器 CAU_IV0..1（H / L）只有在 BUSY 位（CAU_STAT0 寄存器位 4）为 0 时才能被修改，否则写操作都是无效的。

20.4. 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20-3. CAU框图为加密处理器的模块框图。

图 20-3. CAU 框图



20.4.1. 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

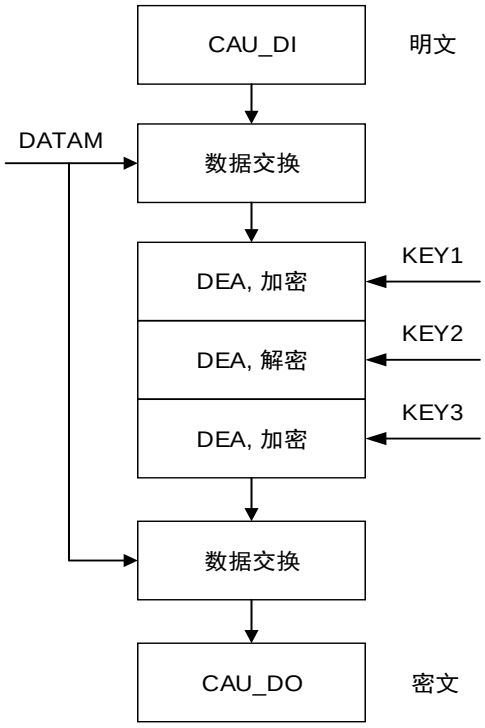
这个选项中，KEY1，KEY2 与 KEY3 都是独立的。详见 FIPS PUB 46-3 -1999（以及 ANSI X9.52-1998）。

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 20-4. DES / TDES ECB 加密*。

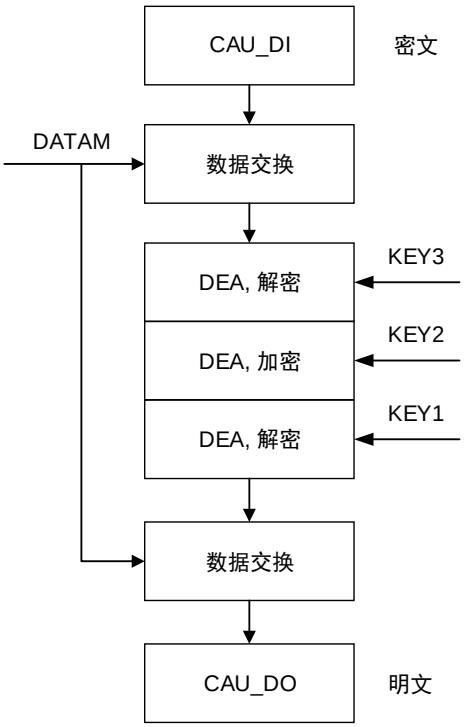
图 20-4. DES / TDES ECB 加密



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 20-5. DES / TDES ECB 解密*。

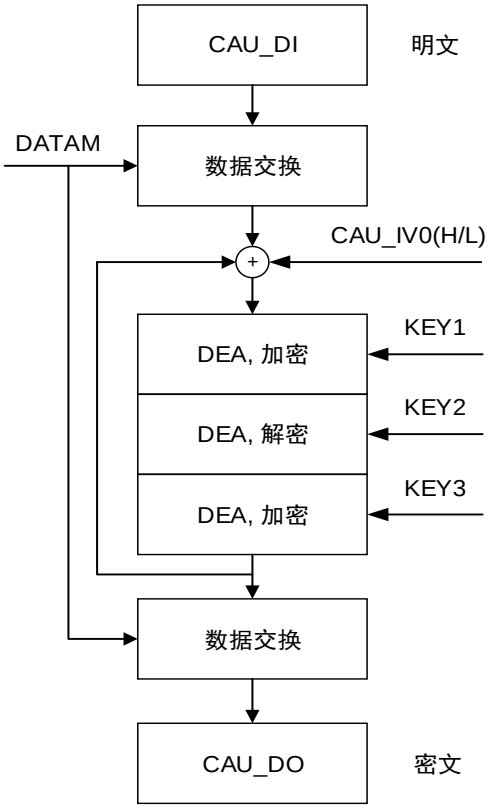
图 20-5. DES / TDES ECB 解密



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 20-6. DES / TDES CBC 加密*。

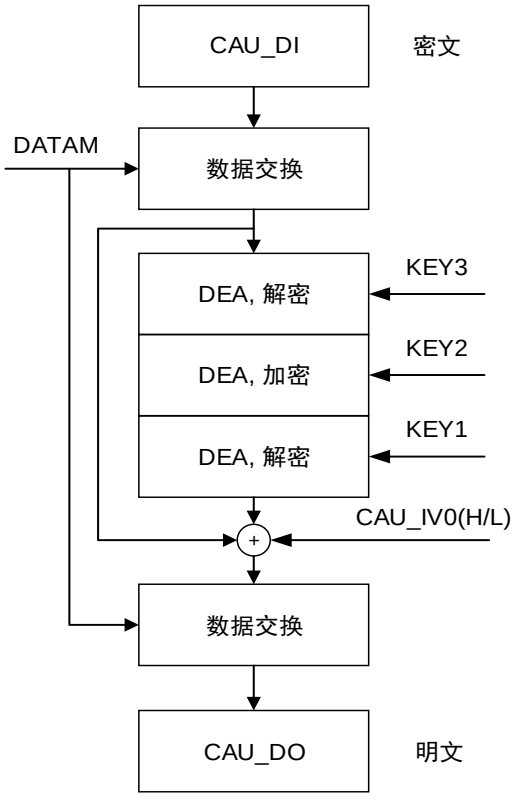
图 20-6. DES / TDES CBC 加密



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 20-7. DES / TDES CBC 解密*。

图 20-7. DES / TDES CBC 解密



20.4.2. 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。

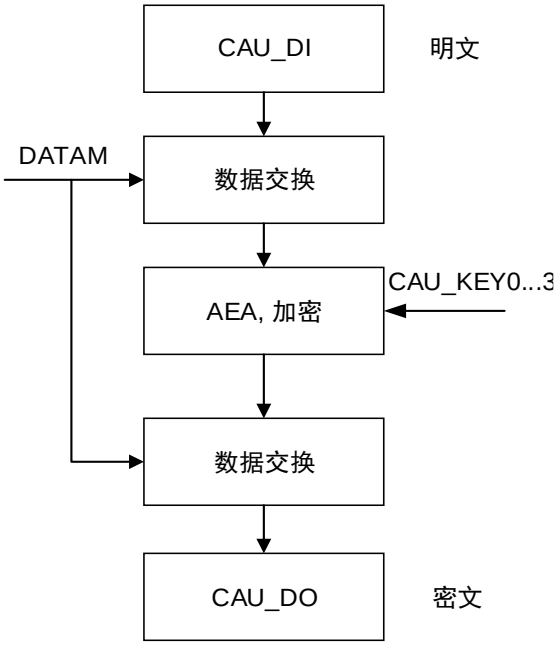
使用 128 位密钥时 AES 密钥为[KEY3 KEY2]，使用 192 位密钥时 AES 密钥为[KEY3 KEY2 KEY1]，使用 256 位密钥时 AES 密钥为[KEY3 KEY2 KEY1 KEY0]。

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 20-8. AES ECB 加密*。

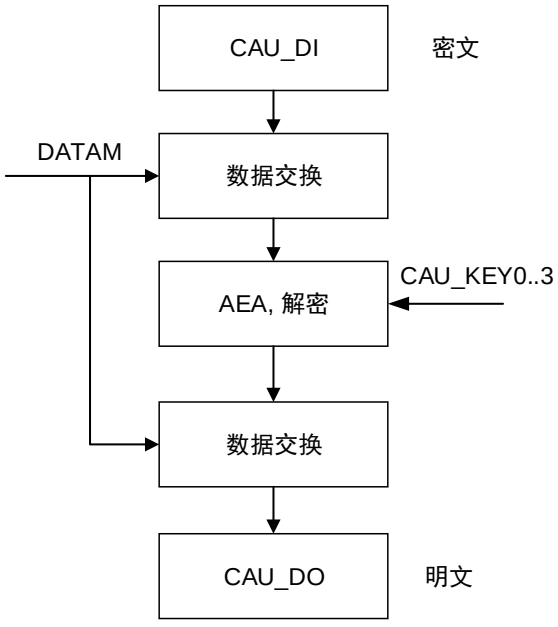
图 20-8. AES ECB 加密



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 20-9. AES ECB 解密*。

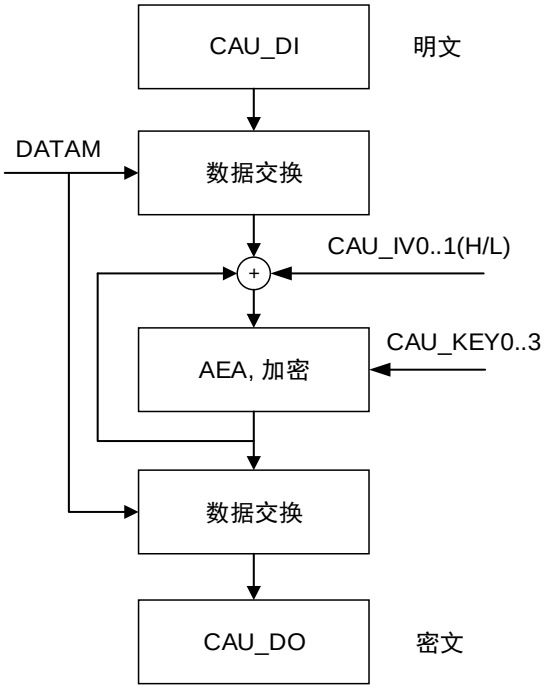
图 20-9. AES ECB 解密



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 20-10. AES CBC 加密*。

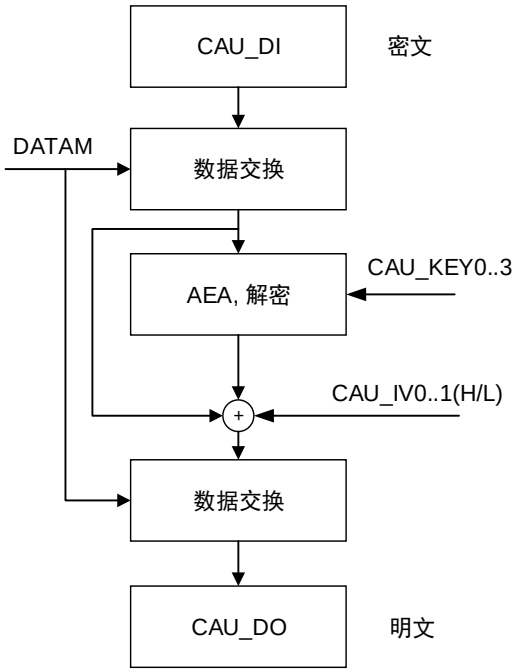
图 20-10. AES CBC 加密



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 20-11. AES CBC 解密*。

图 20-11. AES CBC 解密

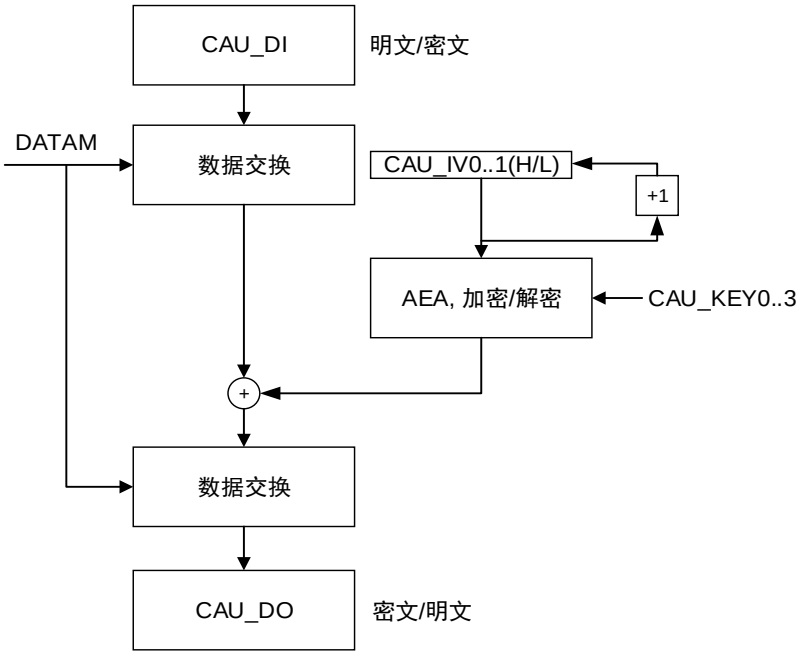


AES 计数器（CTR）模式

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20-12. 计数器块结构*，AES 计数器加密 / 解密流程图见*图20-13. AES CTR 加密 / 解密*。

图 20-12. 计数器块结构

图 20-13. 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。

20.5. 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。

20.6. 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。

20.7. 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。

20.8. 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。

20.9. CAU 寄存器

CAU 基地址：0x4C06 0000

20.9.1. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								NBPILB[3:0]				ALGM[3]	保留	GCM_CCMPH[1:0]	
								rw				rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAUEN	FFLUSH	保留				KEYM[1:0]		DATAM[1:0]		ALGM[2:0]			CAUDIR	保留	
rw	w					rw		rw		rw			rw		

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。

20.9.2. 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

20.9.3. 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。

20.9.4. 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。

20.9.5. 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

20.9.6. 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。

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														OINTEN	IINTEN
														rw	rw

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

20.9.7. 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。

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														OSTA	ISTA
														r	r

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

20.9.8. 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

20.9.9. 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

20.9.10. 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

20.9.11. GCM 或 CCM 模式上下文交换寄存器 x（CAU_GCMCCMCTXSx）（x = 0…7）

偏移地址：0x50 + 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
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。

20.9.12. GCM 模式上下文交换寄存器 x（CAU_GCMCTXSx）（x = 0…7）

偏移地址：0x70 + 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
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。

21. 哈希处理器（HAU）

21.1. 简介

哈希处理器应用于信息安全。支持应用于多种场合的安全哈希算法（SHA-1，SHA-224 和 SHA-256），消息摘要算法（MD5）和哈希运算消息认证码（HMAC）。对长达（2^{64}-1）位的消息，哈希处理器计算消息摘要长度对应于 SHA-1，SHA-224，SHA-256，和 MD5 算法分别为 160 位，224 位，256 位，128 位。而在 HMAC 算法中，SHA-1、SHA-224、SHA-256 或 MD5 将作为哈希函数被调用两次，来产生验证消息。

哈希处理器完全兼容下列标准：

- 联邦信息处理标准出版物180-2（FIPS PUB 180-2）；
- 安全散列标准规范（SHA-1，SHA-224，SHA256）；
- 互联网工程任务组征求意见文档编号1321（IETF RFC 1321）规范（MD5）。

21.2. 主要特性

- 32位AHB从外设；
- 高性能的哈希算法运算；
- 小端数据表示；
- 支持多种数据交换类型，包括32位字不交换，半字交换，字节交换和位交换；
- 可自动填充来适应模数为512位（16×32位）消息摘要的计算；
- 支持DMA模式的数据流传输；
- 哈希/HMAC计算挂起模式。

21.3. 数据类型

哈希处理器每次接收 32 位字，但每次计算处理一个 512 位块。对每个输入字，在送入哈希内核之前都会根据数据类型进行位/字节/半字/不交换。同样在数据输出之前也要进行相同的数据交换。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。SHA-1，SHA-224，SHA-256 的计算均为大端模式。

*图21-1. DATAM 不交换 / 半字交换*和*图21-2. DATAM 字节交换 / 位交换*介绍了在不同数据类型下的数据交换。

图 21-1. DATAM 不交换 / 半字交换

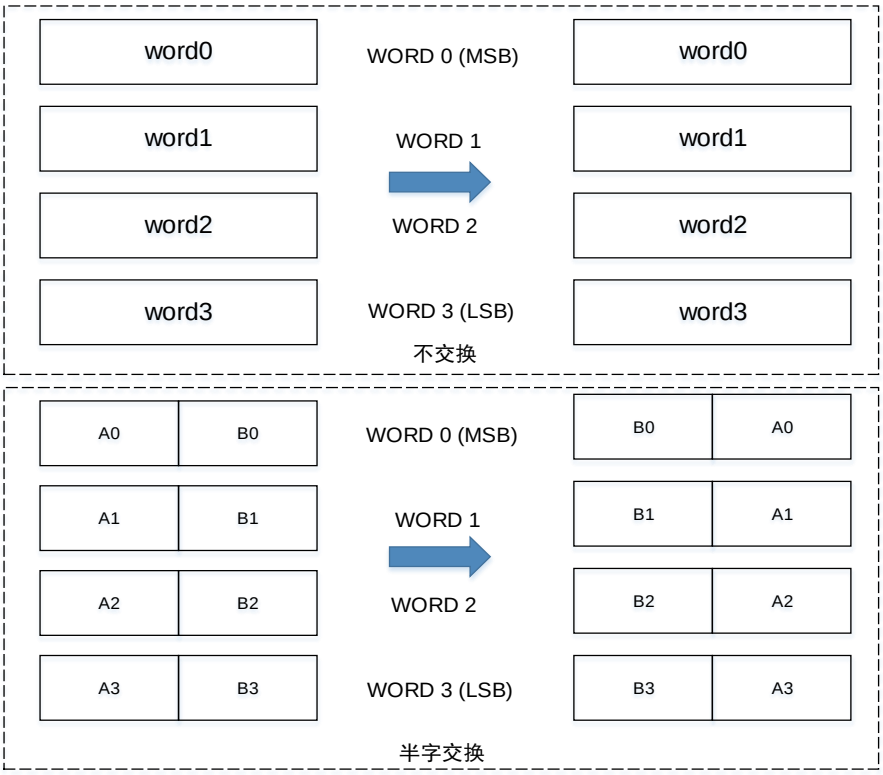


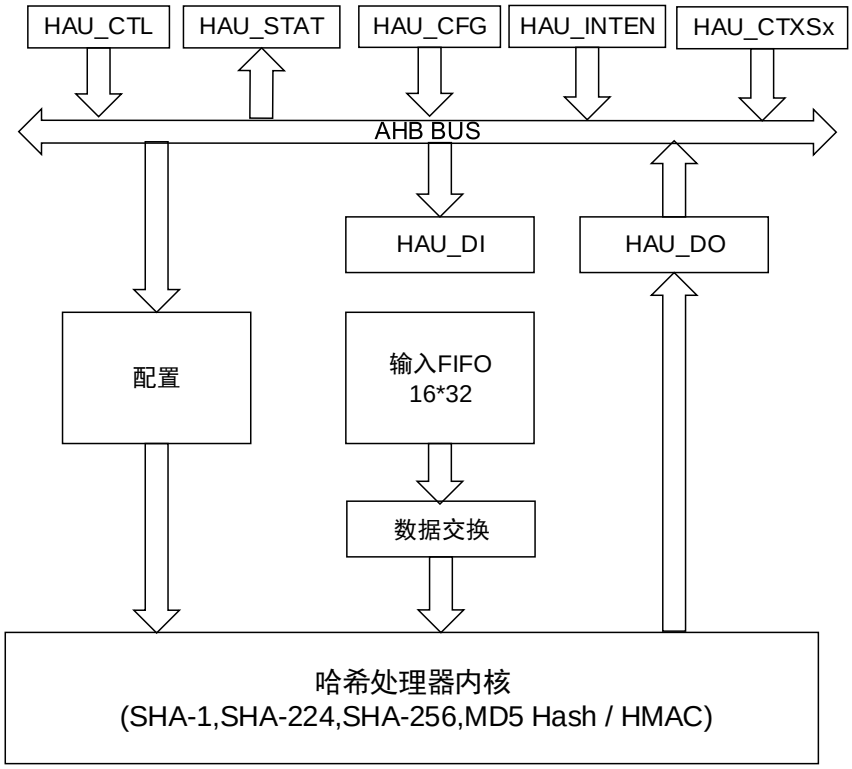
图 21-2. DATAM 字节交换 / 位交换

21.4. HAU 内核

哈希处理器使用安全哈希算法对输入消息进行信息压缩计算。对长度为（2^{64}-1）位的消息摘要计算结果的长度对应于 SHA-1，SHA-224，SHA-256，和 MD5 算法分别为 160 位，224 位，256 位，128 位。哈希处理器可用于生成和验证消息签名，并由于摘要远远小于消息的大小而具有更高的效率。

要由哈希处理器处理的消息应视为位串。消息长度为消息的位数。哈希处理器可以确保信息的安全，因为根据某个给定消息摘要来寻找原对应的消息在计算层面是无法实现的，而在原输入消息上任何的改动都将导致生成完全不同的消息摘要。

图 21-3. HAU 结构框图



21.4.1. 自动数据填充

为确保输入 HAU 内核的数据为 512 位的整数倍，需要对输入消息进行填充。消息填充操作由在原始消息的结尾添加一个 1，后跟几个 0 和一个 64 位整数，填充物（0）将消息填充到整个 512 位的前 448 位，实现生成一个长度为 512 的填充消息块。

消息填充完成后，通过配置 HAU_CFG 寄存器的 VBL 位域来设置上面所述的 64 位整数。设置 HAU_CFG 寄存器的 CALEN 位为 1，开始计算上个数据块的摘要。

数据填充示例：输入消息为“HAU”，对应的 ASCII 码 16 进制表示为

484155

接着根据消息的有效位长度，设置HAU_CFG寄存器的VBL位域为24。接着在位串的第24位处添加一个”1”，随后填充数个“0”使位串模数为448，十六进制结果如下所示：

48415580 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000

之后，添加 64 位整数到已填充的输入消息后，该 64 位整数十六进制值为 18，则最后的结果应该是：

48415580 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000018

21.4.2. 摘要计算

数据填充完成之后，通过 DMA 或 CPU 每次将 512 位的数据块送入 HAU 内核，HAU 对每个数据块进行计算。在 HAU 内核开始计算之前，外设需要知道 HAU_DI 寄存器是否包含消息的最后一位。这可以从输入 FIFO 的状态和 HAU_DI 寄存器来确认。

通过 DMA 传输数据

数据块传输的状态将自动通过 DMA 控制器发送的信息来解释。当 HAU_CFG 寄存器的 CALEN 位置 1 时，将自动开始进行数据填充和摘要计算。

注意：如果消息是个大文件并需要多个 DMA 传输，则应将 MDS 位置 1。另外在传输之前需要设置 VBL 位域。在 DMA 的传输完成之后硬件不会自动将 CALEN 位置 1，以便可以接收新的 DMA 传输。在最后的 DMA 传输期间，需要将 MDS 位清零，从而在最后一个块传输结束时硬件自动将 CALEN 位置 1。

若消息不需要多个 DMA 传输，则将 MDS 置 0 即可，这样在一个 DMA 传输完成之后就会硬件自动置位 CALEN 位。同样的，在 DMA 传输之前也需要先设置 VBL 位域。

通过 CPU 传输数据

当 HAU_DI 寄存器中写入下一个数据块的第一个字时，将开始计算当前数据块的摘要。

将 HAU_CFG 寄存器中 CALEN 位置 1，将开始最后一个数据块的摘要计算。

21.4.3. 哈希模式

将 HAU_CTL 寄存器的 HMS 位设为 0，选择为哈希模式。则当 HAU_CTL 寄存器的 START

位为 1 时，将根据 ALGM 位域的配置选择 SHA-1，SHA-224，SHA-256 或 MD5 算法进行计算。

当从 HAU_DI 寄存器和接收 FIFO 中接收到 512 位的消息块时，将根据 DMA 和 CALEN 位状态开始摘要的计算。

最终的计算结果可以从 HAU_DO0...7 寄存器中读取。

21.4.4. HMAC 模式

HMAC 模式根据用户所选的密钥来进行消息验证。更多关于 HMAC 规范的信息请参阅“HMAC：密钥散列消息认证，H. Krawczyk，M. Bellare，R. Canetti，1997 年 2 月”。

HMAC 算法表示如下：

HMAC（input）= HASH[（（key | opad）XOR 0x5c）| HASH（（（key | ipad）XOR 0x36）| input）]

其中“ipad”和“opad”用于将密钥用数个“0”进行填充扩展到 512 位，“|”为连接符。

HMAC 模式需要四个不同阶段：

1. 将HAU_CTL寄存器的HMS位置1，并根据期望的算法设置ALGM位域。若密钥“key”长度超过64个字节，则还需配置HAU_CTL寄存器的KLM位。之后，将START位置位以启动HAU内核；
2. 密钥“key”作为输入消息来进行哈希模式下的计算；
3. 当输入了最后一个字并开始计算之后，HAU生成新的密钥“key”作为内部哈希密钥；
4. 在第一次的哈希计算后，HAU内核开始接收用于外部哈希函数的密钥，通常外部的哈希函数使用与内部哈希密钥相同的新的密钥“key”。当输入了密钥的最后一个字，则开始进行计算，计算结果可从HAU_DO寄存器中读取。

21.5. HAU 挂起模式

HAU 可以暂时挂起哈希或 HMAC 操作，从而先执行优先级更高的任务，在处理完优先级更高的任务后，再完成被挂起数据块的哈希或 HMAC 操作。

挂起任务前，必须将被挂起任务的上下文从寄存器保存到存储器，恢复任务时，再从存储器恢复到 HAU 寄存器。

以下说明由 CPU 或 DMA 传输数据时，按照下列的步骤来完成被挂起 HAU 任务的处理。

21.5.1. 通过 CPU 加载数据

1. 停止当前数据处理与传输。等待 BUSY 位为 0，若 NWIF[3:0]的值大于 0，则需等待 DIF 位置位（若 NWIF[3:0]的值等于 0，则不等待 DIF 位置位）。只有在当前未处理任何块时才能保存上下文；

2. 保存当前配置。将 HAU_INTEN，HAU_CFG，HAU_CTL，HAU_CTXS0 到 HAU_CTXS37

（如果正在进行 HMAC 操作，则 HAU_CTXS0 到 HAU_CTXS53）寄存器的内容保存到存储器中；

3. 配置并处理新消息；

4. 恢复之前的配置环境。将存储器中保存的值写入 HAU_INTEN，HAU_CFG，HAU_CTL 寄存器中；

5. 恢复消息的计算。置位 HAU_CTL 寄存器的 START 位来初始化重新开始新消息的摘要计算。

6. 恢复之前的内核状态。将存储器中保存的值写入 HAU_CTXS0 到 HAU_CTXS37 寄存器中（如果涉及 HMAC 操作，则 HAU_CTXS0 到 HAU_CTXS53）；

7. 从之前挂起的地方继续处理。

21.5.2. 通过 DMA 加载数据

1. 等待 BUSY 位为 0，此时若 HAU_STAT 寄存器的 CCF 位置位，则不需要后续的上下文交换，否则再等待 BUSY 位为 1；

2. 停止当前数据传输。禁能 DMA1 的通道 7 数据传输，再将 HAU_CTL 寄存器中的 DMAE 位清零以禁能 DMA 请求；

3. 保存当前配置。等待 BUSY 位为 0，此时若 HAU_STAT 寄存器的 CCF 位置位，则不需要后续的上下文交换，否则将 HAU_INTEN，HAU_CFG，HAU_CTL，HAU_CTXS0 到 HAU_CTXS37（如果正在进行 HMAC 操作，则 HAU_CTXS0 到 HAU_CTXS53）寄存器的内容保存到存储器中；

4. 配置并处理新消息；

5. 恢复之前的环境配置。将存储器中保存的值写入 HAU_INTEN，HAU_CFG，HAU_CTL 寄存器中；

6. 恢复 DMA 通道传输。重新配置 DMA 通道以继续数据传输；

7. 恢复消息的计算。置位 HAU_CTL 寄存器的 START 位来初始化重新开始新消息的摘要计算。

8. 恢复之前内核的状态。将存储器中保存的值写入 HAU_CTXS0 到 HAU_CTXS37 寄存器中（如果涉及 HMAC 操作，则 HAU_CTXS0 到 HAU_CTXS53）；

9. 置位 HAU_CTL 寄存器的位 DMAE，从之前挂起的地方继续处理。

注意：如果 HAU_CTL 寄存器的位 NWIF[3:0]为 0，则说明上下文交换发生在在两个块之间，上一个数据块已完全处理，并且下一个块还未推入到输入 FIFO，那么此时不需要保存和恢复 HAU_CTXS22 到 HAU_CTXS37 寄存器。

21.6. HAU 中断

HAU 具有两个独立的中断源，并在 HAU_STAT 有相应的状态位。这两个状态位用于指示输入 FIFO 的状态，以及摘要的计算是否完成。

HAU_INTEN 寄存器为中断使能寄存器。将相应位置 1 可以使能中断。

21.6.1. 输入 FIFO 中断

当输入 FIFO 中的数据处理已完成时，输入 FIFO 标志位 DIF 置位。如果置位 DIIE 位使能了输入 FIFO 中断，则当输入 FIFO 标志位 DIF 置位时会发生输入 FIFO 中断。

21.6.2. 计算完成中断

当摘要计算完成时，计算完成标志位 CCF 将置位。如果置位 CCIE 位使能了计算完成中断，则当计算完成标志位 CCF 置位时会发生计算完成中断。

21.7. HAU 寄存器

HAU 基地址：0x4C06 0400

21.7.1. 控制寄存器（HAU_CTL）

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													ALGM[1]	保留	KLM
													rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		MDS	DINE	NWIF[3:0]				ALGM[0]	HMS	DATAM[1:0]		DMAE	START	保留	
		rw	r	r				rw	rw	rw		rw	w		

位 / 位域	名称	描述
31:19	保留	必须保持复位值。
18	ALGM[1]	算法选择位1
17	保留	必须保持复位值。
16	KLM	密钥长度模式 0：密钥长度 ≤ 64 字节 1：密钥长度 > 64 字节 **注意**：必须在非计算期间修改该位
15:14	保留	必须保持复位值。
13	MDS	多DMA选择 如果哈希消息为大型文件需要多个DMA传输时，将此位置1 0：仅需要单次DMA传输，在DMA传输结束时硬件自动将CALEN位置1 1：需要多次DMA传输，在DMA传输结束时硬件不自动将CALEN位置1
12	DINE	DI寄存器非空 0：DI寄存器空 1：DI寄存器非空 **注意**：当START位或CALEN位为1时此位会清零
11:8	NWIF[3:0]	输入FIFO中的字数 **注意**：当START位置位时，或开始进行摘要计算时（CALEN位置位，或者DMA传输结束），该位域清零
7	ALGM[0]	算法选择位0 该位和CTL寄存器的位18用于选择SHA-1，SHA-224，SHA256或MD5 算法：

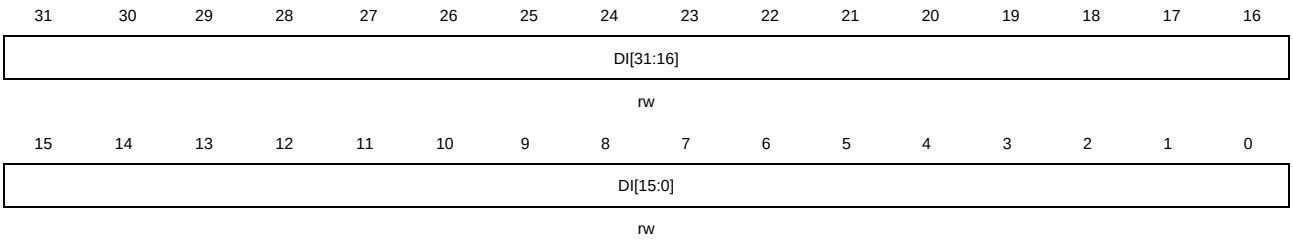
		00：选择SHA-1算法 01：选择MD5算法 10：选择SHA224算法 11：选择SHA256算法
6	HMS	HAU模式选择，必须在非计算期间修改该位 0：选择HASH模式 1：选择HMAC模式。如果密钥长度大于64字节，则还需配置KLM位。
5:4	DATAM[1:0]	数据交换类型 定义输入到HAU_DI寄存器中的数据格式 00：不交换，写入到HAU_DI寄存器的数据将直接送入FIFO，不进行交换 01：半字交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行半字交换 10：字节交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行字节交换 11：位交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行位交换
3	DMAE	DMA使能 0：禁止DMA传输 1：使能DMA传输 **注意：**1.当DMA传输消息的最后一个数据时，将由硬件清零该位。当START置位时，不会清零该位。 2.如果DMA正在传输数据，将该位写入0不会中止当前的传输，而直到当前传输结束或START位置为1之后，才会禁止传输
2	START	开始摘要计算 0：没有影响 1：开始新消息的摘要计算 **注意：**读取该位将始终返回0
1:0	保留	必须保持复位值。

21.7.2. 数据输入寄存器（HAU_DI）

地址偏移：0x04
复位值：0x0000 0000

该数据输入寄存器用于将 512 位的数据块送入输入 FIFO 进行处理。当正在进行摘要计算时，所有对该寄存器的新的写访问将被延迟，直到计算完成。

该寄存器只能按字（32 位）访问。

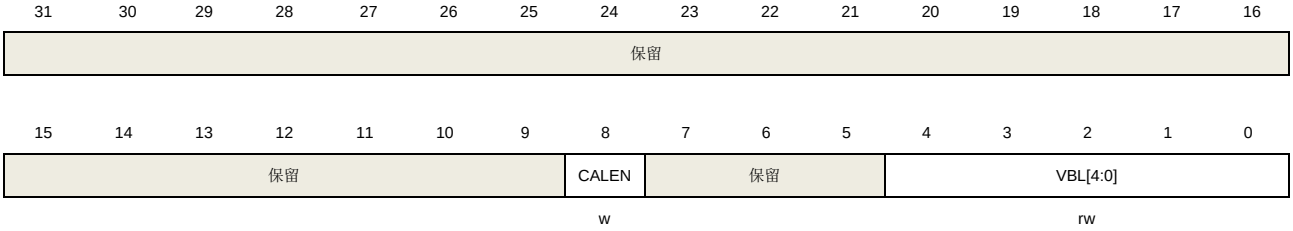


位 / 位域	名称	描述
31:0	DI[31:0]	消息数据输入 当数据写入这些寄存器时，寄存器中当前的内容被推入输入FIFO中同时更新为新的值。当读寄存器时，返回寄存器的当前内容。

21.7.3. 配置寄存器（HAU_CFG）

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
31:9	保留	必须保持复位值。
8	CALEN	使能摘要计算 0：不计算 1：先使用VBL位域对数据进行数据填充，然后开始计算最终消息摘要 **注意：**读该位将返回0
7:5	保留	必须保持复位值。
4:0	VBL[4:0]	消息的最后一个字中的有效位数 0x00：对于写入HAU_DI寄存器的最后一个数据，所有32位（在数据交换后）均有效。 0x01：对于写入HAU_DI寄存器的最后一个数据，仅位[31]（在数据交换后）有效。 0x02：对于写入HAU_DI寄存器的最后一个数据，仅位[31:30]（在数据交换后）有效。 0x03：对于写入HAU_DI寄存器的最后一个数据，仅位[31:29]（在数据交换后）有效。 … 0x1F：对于写入HAU_DI寄存器的最后一个数据，仅位[31:1]（在数据交换后）有效。 **注意：**必须在置位CALEN位之前配置该位。

21.7.4. 数据输出寄存器（HAU_DO0…7）

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

数据输出寄存器为只读寄存器，用于从输出 FIFO 中接收计算结果。置位 START 位将复位该寄存器。当正在进行摘要计算时，所有对该寄存器的新的读访问将被延迟，直到计算完成。

在 SHA-1 模式中，使用 HAU_DO0…4

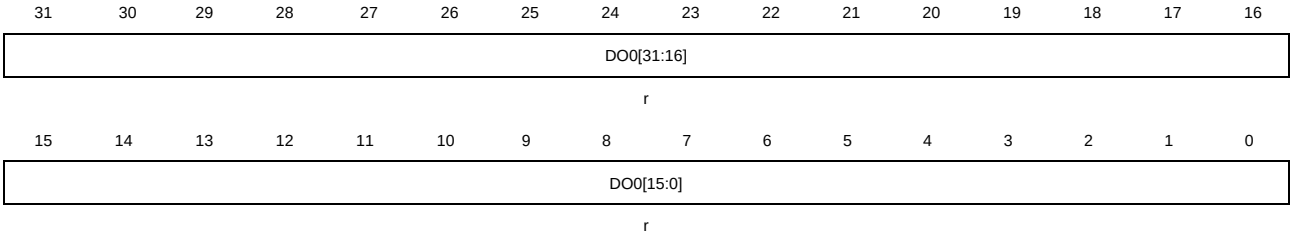
在 MD5 模式中，使用 HAU_DO0…3

在 SHA-224 模式中，使用 HAU_DO0…6

在 SHA-256 模式中，使用 HAU_DO0…7

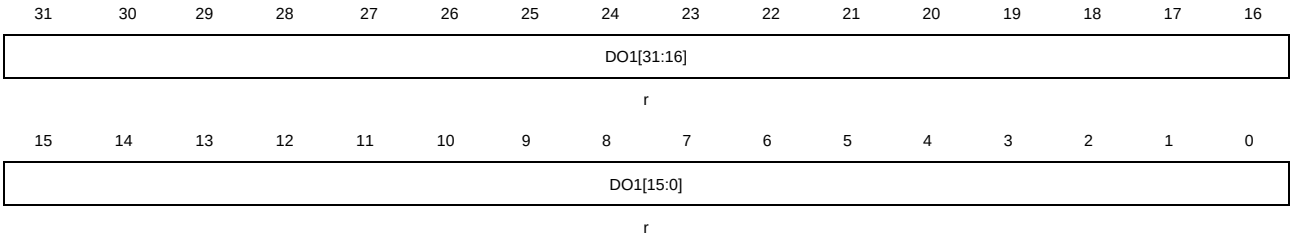
HAU_DO0

地址偏移：0x0C 和 0x310



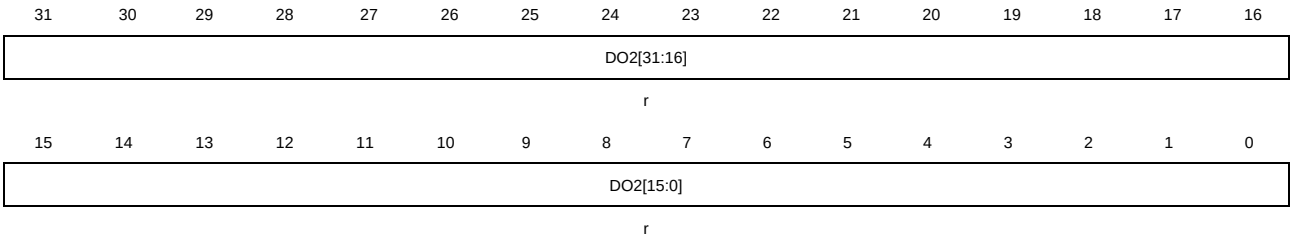
HAU_DO1

地址偏移：0x10 和 0x314



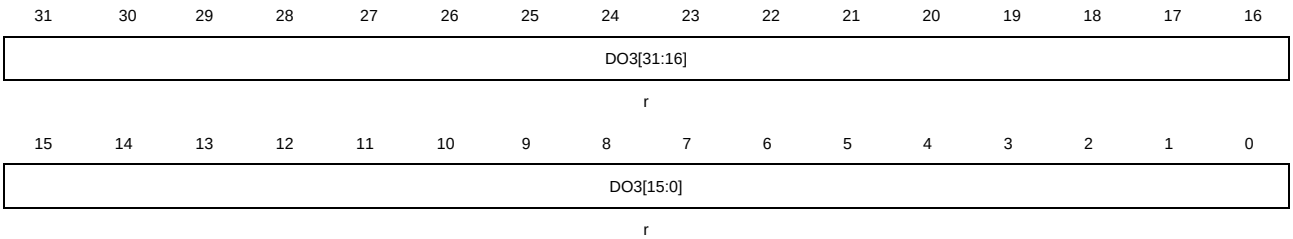
HAU_DO2

地址偏移：0x14 和 0x318



HAU_DO3

地址偏移：0x18 和 0x31C



HAU_DO4

地址偏移：0x1C 和 0x320

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO4[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO4[15:0]															
r															

HAU_DO5

地址偏移：0x324

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO5[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO5[15:0]															
r															

HAU_DO6

地址偏移：0x328

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO6[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO6[15:0]															
r															

HAU_DO7

地址偏移：0x32C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DO7[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DO7[15:0]															
r															

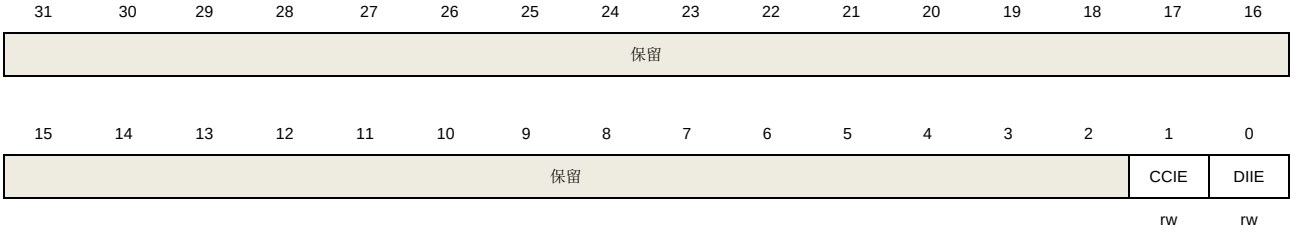
位 / 位域	名称	描述
31:0	DO0..7[31:0]	消息摘要结果

21.7.5. 中断使能寄存器（HAU_INTEN）

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



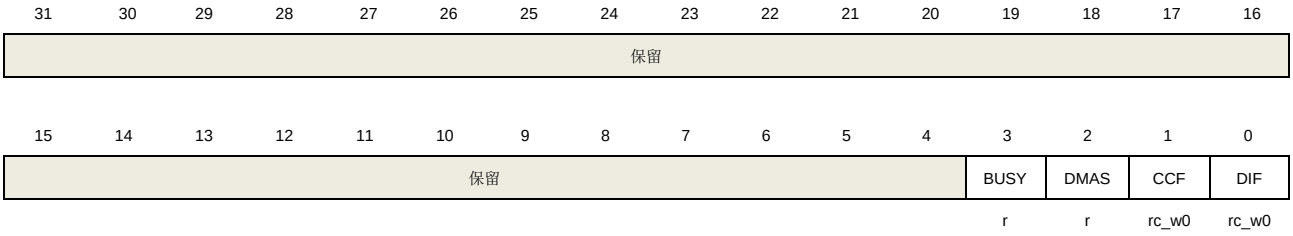
位 / 位域	名称	描述
31:2	保留	必须保持复位值。
1	CCIE	计算完成中断使能 0：禁止计算完成中断 1：使能计算完成中断
0	DIIE	数据输入中断使能 0：禁止数据输入中断 1：使能数据输入中断

21.7.6. 状态与标志寄存器（HAU_STAT）

地址偏移：0x24

复位值：0x0000 0001

该寄存器只能按字（32 位）访问。



位 / 位域	名称	描述
31:4	保留	必须保持复位值。
3	BUSY	忙标志位 0：未处理任何块 1：正在处理某个数据块
2	DMAS	DMA状态标志 0：DMA接口被禁用（DMAE=0）并且未在进行任何传输

		1：DMA接口被使能（DMAE=1）并且未在进行任何传输
1	CCF	计算完成状态标志 0：计算未完成 1：所有消息摘要计算完成
0	DIF	数据输入状态标志 0：有一个字数据写入数据输入寄存器 1：完成一个字数据的初步处理（只有在输入FIFO中的数据才会被处理）

21.7.7. 上下文交换寄存器 x（HAU_CTXSx）（x = 0...53）

地址偏移：0xF8 + 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
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]	HAU处理器完整的内部状态信息。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。

22. 公钥加密处理器（PKCAU）

22.1. 简介

公钥加密又称非对称加密，非对称加密算法加密和解密采用不同的密钥。公钥加密处理器（PKCAU）支持加速 GF(p)（伽罗华域）上的 RSA（Rivest、Shamir 和 Adleman）、Diffie-Hellmann（DH 密钥交换）或 ECC（椭圆曲线加密）加密算法。这些操作在蒙哥马利域内执行能提高运算效率。

22.2. 主要特征

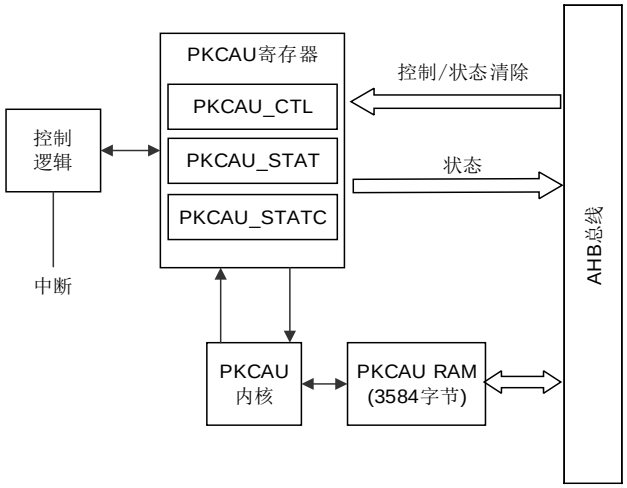
- 支持操作数高达 3136 位的 RSA/DH 算法；
- 支持操作数高达 640 位的 ECC 算法；
- RSA 模幂运算，RSA CRT 求幂；
- ECC 标量乘法，曲线上点的检查；
- ECDSA（椭圆曲线数字签名算法）签名和验证；
- 支持蒙哥马利模法，加速 RSA，DH 和 ECC 运算；
- 内嵌 3584 字节 RAM；
- 蒙哥马利域和自然域之间的相互转换；
- PKCAU 外设为 32 位外设，只支持 32 位访问。

22.3. 功能说明

公钥加速器（PKCAU）用于加速素域 GF(p)上 RSA、DH 以及椭圆曲线加密（ECC）运算。PKCAU 模块包含 PKCAU RAM、PKCAU 内核以及外设寄存器。PKCAU RAM 用于存放运算所需的参数，并在计算完成后，保存计算结果。

PKCAU 的内部结构如*图 22-1. PKCAU 模块框图*所示。

图 22-1. PKCAU 模块框图



22.3.1. 操作数

假设 RSA 操作数长度为 ROS，模长度为 ML，则数据长度ROS = (ML/32+1)个字。假设 ECC 操作数长度为 EOS，模长度为 ML，则数据长度EOS = (ML/32+1)个字。

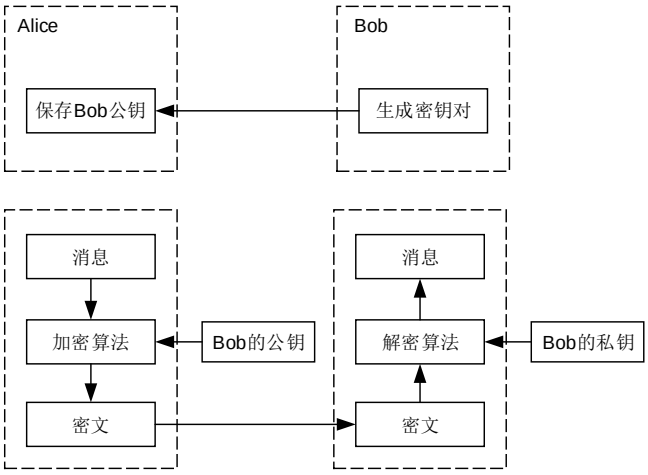
PKCAU 支持操作数高达 3136 位（98 个字）的 RSA/DH 算法和操作数高达 640 位（20 个字）的 ECC 算法。ROS 最大为 99 个字，EOS 最大为 21 个字。

在将输入参数写入 PKCAU RAM 时，必须添加一个 0x00000000。PKCAU RAM 是小端存储，例如，当将用于 ECC 标量乘法的 ECC P256 的输入参数 x_p 写入 PKCAU RAM，模数长度为 8 个字，最低字节存放在偏移为 0x55C 的地址，最高字节存放在偏移为 0x578 的地址，0x00000000 存放在偏移为 0x57C 的地址。

22.3.2. RSA 算法

RSA 算法是一种常用的公钥密码算法，是应用最广泛的非对称密码算法。RSA 算法流程如*图 22-2. RSA 算法流程图*所示。

图 22-2. RSA 算法流程图



一个完整的公钥密码体制包含密钥对（公钥和私钥）、加密算法和解密算法。

RSA 密钥对生成

1、 选择两个大素数 p 和 q（p≠q）；
2、 计算n=p×q，n 为公钥和私钥的模数；
3、 计算L=ø (n)=(p-1)(q-1)，其中ø(n)为欧拉函数；
4、 选择 e，满足1<e<L，同时满足 e 和 L 互质；
5、 计算 d，满足1<d<L，同时满足 e*d mod L = 1。

通过以上计算可以得到*表 22-1. RSA 算法参数*中所示参数：

表 22-1. RSA 算法参数

参数	描述
n	模数
e	公开指数
d	私密指数
(n,e)	公钥
(n,d)	私钥

RSA 加密

Bob 生成符合 RSA 算法标准的密钥对，包含公钥和私钥，并将公钥发送给 Alice，私钥自己保存。Alice 可以通过 Bob 的公钥对消息 m 进行加密，从而得到密文 c。并将密文发给 Bob。密文$c = m^e \bmod n$。

RSA 解密

Bob 收到密文后采用私钥对密文进行解密得到明文。解密过程为$m = c^d \bmod n$。

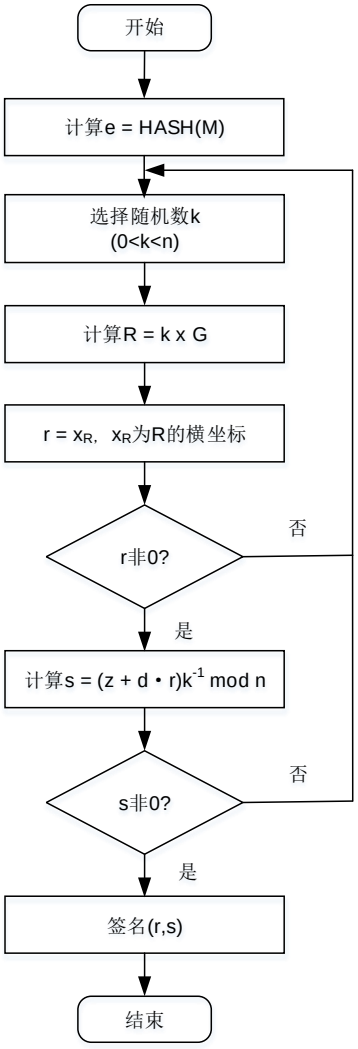
22.3.3. ECC 算法

假设消息为 M，d 为私钥，G 为椭圆曲线上的基点，Q 为椭圆曲线上的某点，椭圆曲线素数阶为 n，散列函数为 HASH()，z 是 HASH(M)最左边的位，L_n 是 n 的位长度，ECDSA 签名和验证详细描述如下：

ECDSA 签名

ECDSA 签名结果由 r 和 s 两部分组成。ECDSA 生成签名流程如*图 22-3. ECDSA 签名流程图*所示。

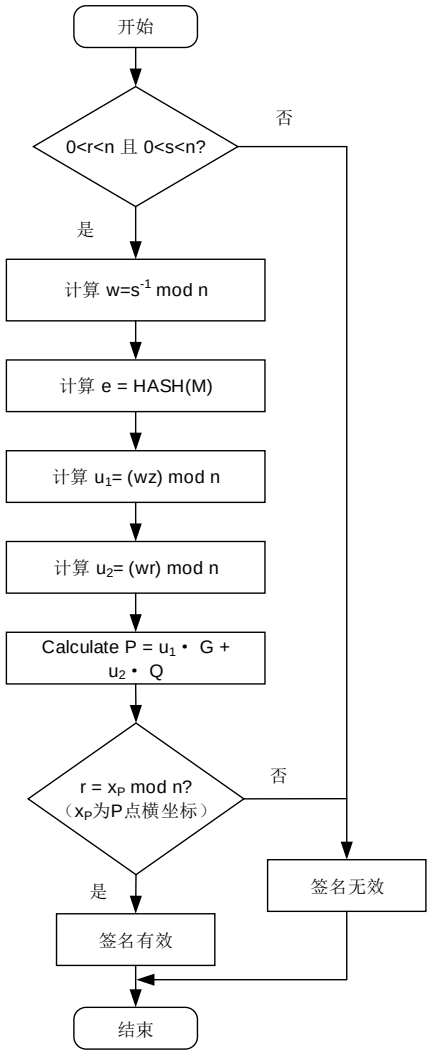
图 22-3. ECDSA 签名流程图



ECDSA 验证签名

在验证签名之前，确保得到签名者的公钥、消息以及签名(r,s)。ECDSA 验证签名的流程如*图 22-4. ECDSA 验证流程图*所示。

图 22-4. ECDSA 验证流程图



注意：上图中的 HSAH 是约定的散列函数。

22.3.4. 整数算术运算模式

通过配置 PKCAU_CTL 寄存器中的 MODSEL[5:0]，可以选择整数算术运算模式。可选运算模式如*表 22-2. 整数算术运算*。

表 22-2. 整数算术运算

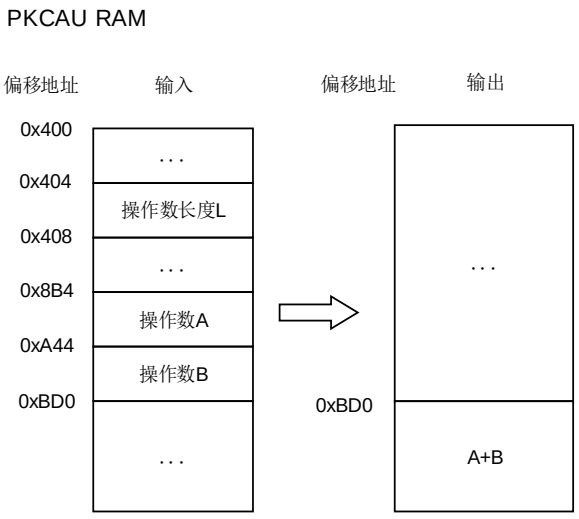
MODSEL[5:0]	运算模式
000000	蒙哥马利参数计算然后模幂
000001	只进行蒙哥马利参数计算
000010	只进行模幂运算（蒙哥马利参数必须预先加载）
000111	RSA CRT 求幂
001000	模逆运算
001001	算术加法
001010	算术减法

MODSEL[5:0]	运算模式
001011	算术乘法
001100	算术比较
001101	取模运算
001110	模加法
001111	模减法
010000	蒙哥马利乘法

算术加法

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“001001”，可以选择运算模式为算术加法运算。运算说明如*图 22-5. 算术加法*所示。运算结果为result = A+B。

图 22-5. 算术加法



其中，$0 \le A < 2^{L}$，$0 \le B < 2^{L}$，$0 \le result < 2^{L+1}$，$0 < L \le 3136$。

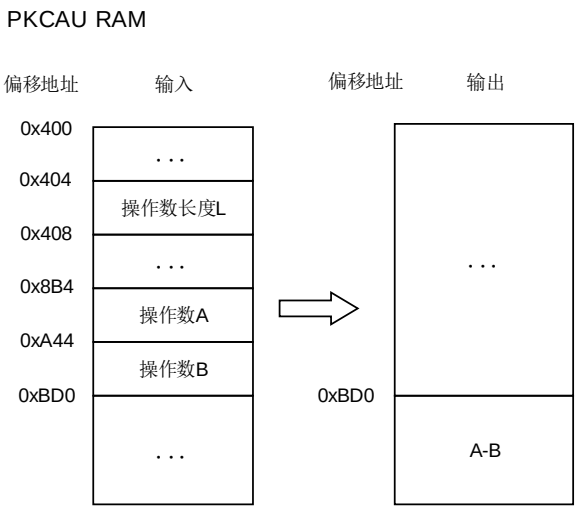
算术减法

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“001010”，可以选择运算模式为算术减法运算。运算说明如*图 22-6. 算术减法*所示。

如果A≥B，运算结果为result = A-B；

如果A<B，运算结果为result = $A-B+2^{L+ceil(L\%32)}$。

图 22-6. 算术减法

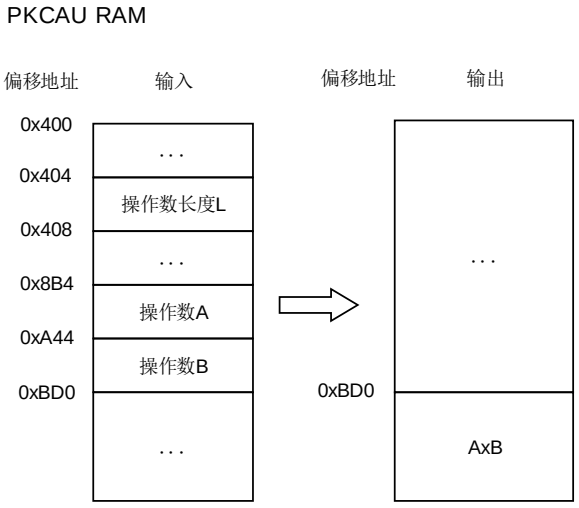


其中，$0 \le A < 2^L$，$0 \le B < 2^L$，$0 \le result < 2^L$，$0 < L \le 3136$。

算术乘法

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“001011”，可以选择运算模式为算术乘法运算。运算说明如*图 22-7. 算术乘法*所示。运算结果为result = A×B。

图 22-7. 算术乘法



其中，$0 \le A < 2^L$，$0 \le B < 2^L$， $0 \le result < 2^{2L}$，$0 < L \le 3136$。

算术比较

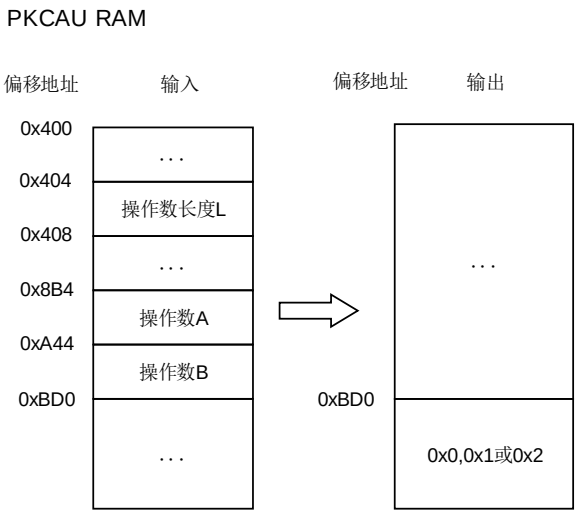
将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“001100”，可以选择运算模式为算术比较运算。运算说明如*图 22-8. 算术比较*所示。

如果A=B，运算结果为result =0x0；

如果A>B，运算结果为result =0x1；

如果A<B，运算结果为result =0x2。

图 22-8. 算术比较

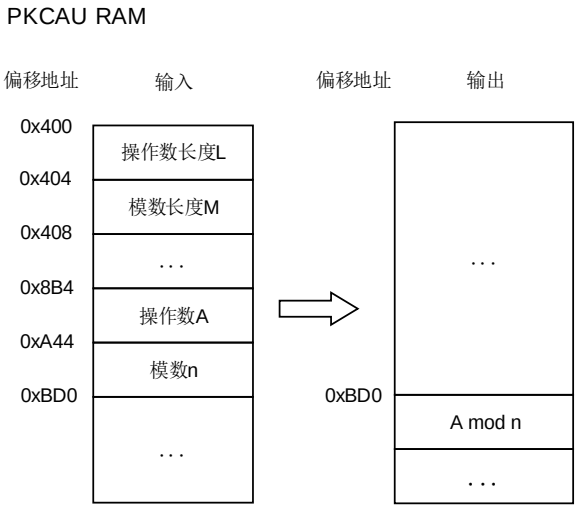


其中，$0 \le A < 2^L$，$0 \le B < 2^L$，result =0x0， 0x01 或 0x2，$0 < L \le 3136$。

取模运算

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“001101”，可以选择运算模式为取模运算。运算说明如*图 22-9. 取模运算*所示。运算结果为result = A mod n。

图 22-9. 取模运算

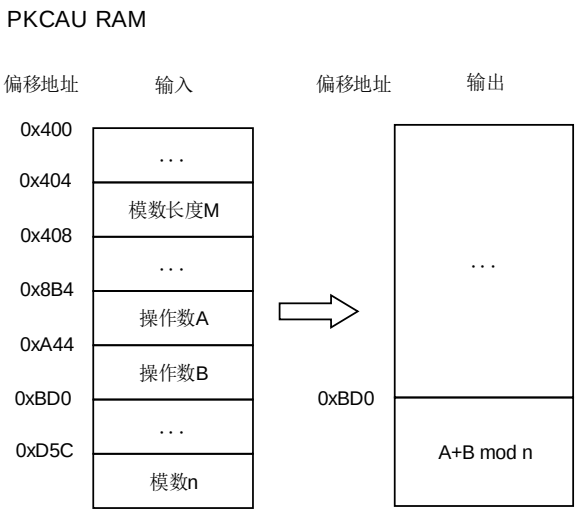


其中，$0 < L \le 3136$，$0 < M \le 3136$， $0 \le A < 2^L$，$0 < n < 2^M$，$0 \le result < n$。

模加法

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“001110”，可以选择运算模式为模加法运算，运算说明如*图 22-10. 模加法*所示。运算结果为result = A+B mod n。

图 22-10. 模加法



其中，0≤A<n，0≤B<n，0≤result<n，0<n<2^M，0<M≤3136。

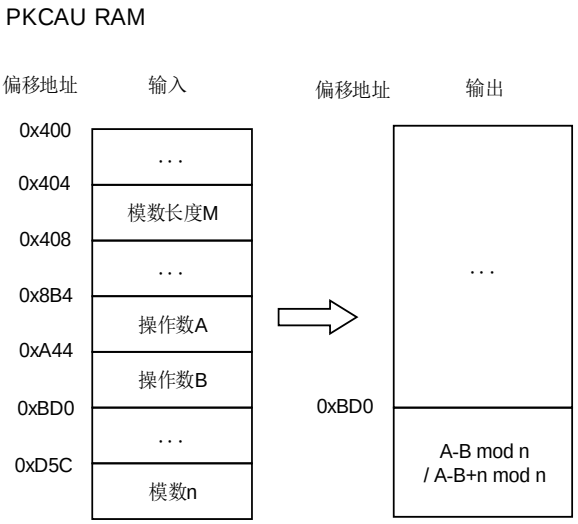
模减法

将PKCAU_CTL 寄存器中的MODSEL[5:0]配置为“001111”，可以选择运算模式为模减法运算。运算说明如*图 22-11. 模减法*所示。

如果A≥B，运算结果为result = A-B mod n。

如果A<B，运算结果为result = A-B+n mod n。

图 22-11. 模减法



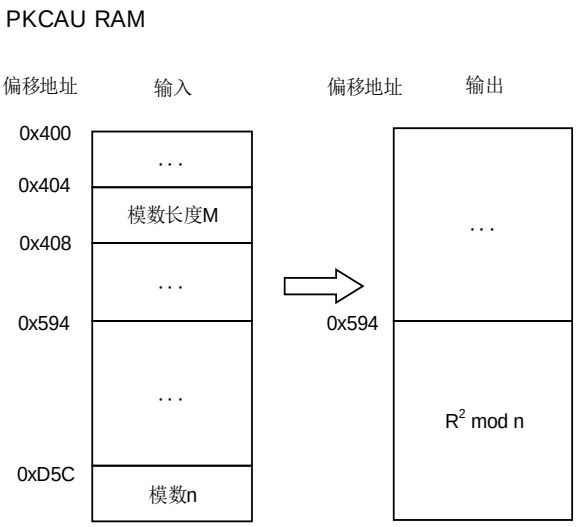
其中，0≤A<n，0≤B<n，0≤result<n， 0<n<2^M，0<M≤3136。

蒙哥马利参数计算

PKCAU 将操作数转换为蒙哥马利剩余系统表示需要使用到蒙马参数(R^2 mod n)。

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“000001”，可以选择运算模式为只进行蒙哥马利参数计算，说明如*图 22-12. 蒙哥马利参数计算*所示。

图 22-12. 蒙哥马利参数计算



其中，0<M≤3136，1<n<2^M（n 为奇数整数）。

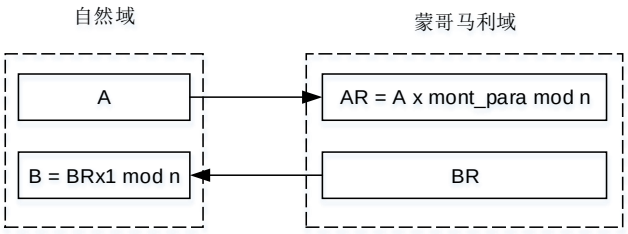
蒙哥马利乘法

假设 A，B，C 均为自然域中的数。“x”指蒙哥马利乘法。蒙哥马利乘法运算的两个主要用途如下：

1、 蒙哥马利域和自然域之间的相互映射。

如*图 22-13. 蒙哥马利域和自然域之间的相互映射*所示。如果 A 是自然域中的整数，蒙哥马利参数 mont_para 为R^2 mod n，AR = A x mont_para mod n为蒙哥马利域 A。相反地，如果 BR 是蒙哥马利域的整数，计算结果B = BRx1 mod n在自然域。

图 22-13. 蒙哥马利域和自然域之间的相互映射



2、 执行模乘运算A x B mod n。

(1)、计算蒙哥马利参数mont_para = R^2 mod n。

(2)、计算AR = A x mont_para mod n，输出在蒙哥马利域。

(3)、计算AB = AR x B mod n，输出在自然域。

多元模乘A x B x C mod n步骤如下：

(1)、计算蒙哥马利参数mont_para = R^2 mod n。

(2)、计算AR = A x mont_para mod n，输出在蒙哥马利域。

(3)、计算BR = B x mont_para mod n，输出在蒙哥马利域。

(4)、计算ABR = AR x BR mod n，输出在蒙哥马利域。

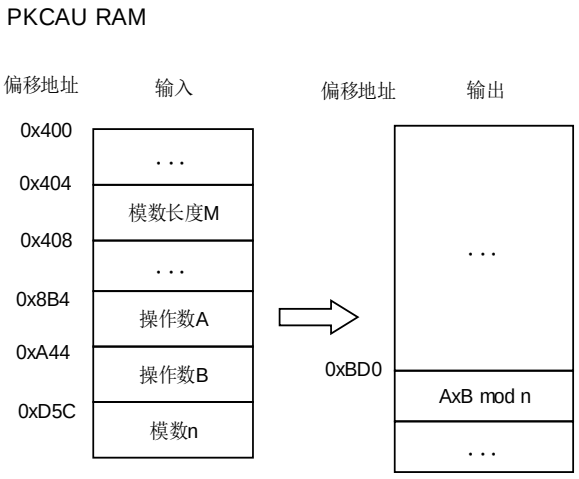
(5)、计算CR = C x mont_para mod n，输出在蒙哥马利域。

(6)、计算ABCR = ABR x CR mod n，输出在蒙哥马利域。

(7)、计算ABC= ABCR x 1 mod n，输出在自然域。

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“010000”，可以选择运算模式为蒙哥马利乘，说明如*图 22-14. 蒙哥马利乘法*所示。

图 22-14. 蒙哥马利乘法



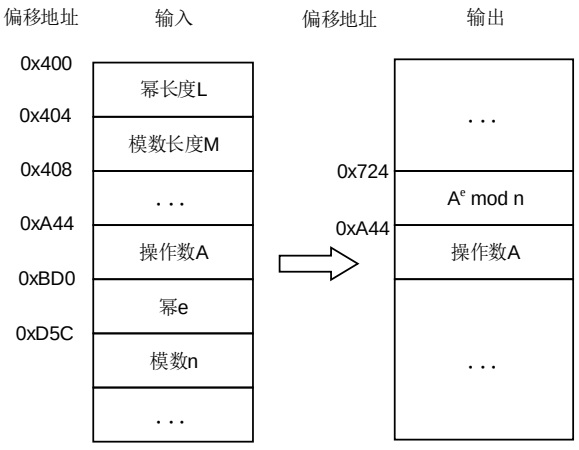
其中，0≤A<n，0≤B<n，0<n<2^M，0<M≤3136（n 为奇数整数）。

模幂运算

普通模式

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“000000”，可以选择运算模式为普通模幂运算，运算说明如*图 22-15. 普通模式模幂运算*所示。运算结果为result = A^e mod n。

图 22-15. 普通模式模幂运算

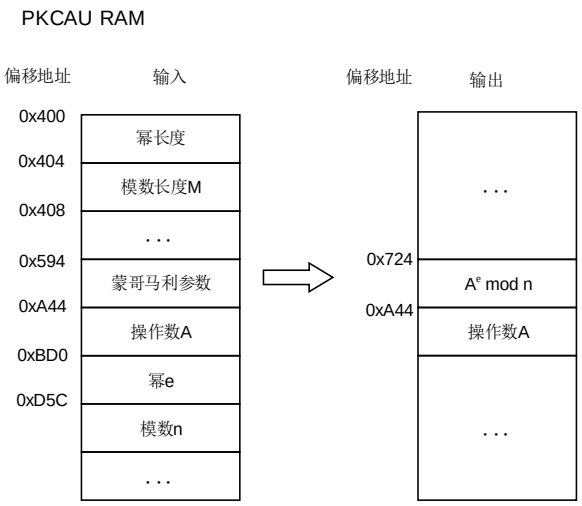


其中，0<L≤3136， 0<M≤3136，0≤A<n，0≤e<2^L，0≤result<n，1<n<2^M（n 为奇数整数）。

快速模式

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“000010”，可以选择运算模式为快速模幂运算，运算说明如*图 22-16. 快速模式模幂运算*所示。运算结果为result = A^e mod n。

图 22-16. 快速模式模幂运算

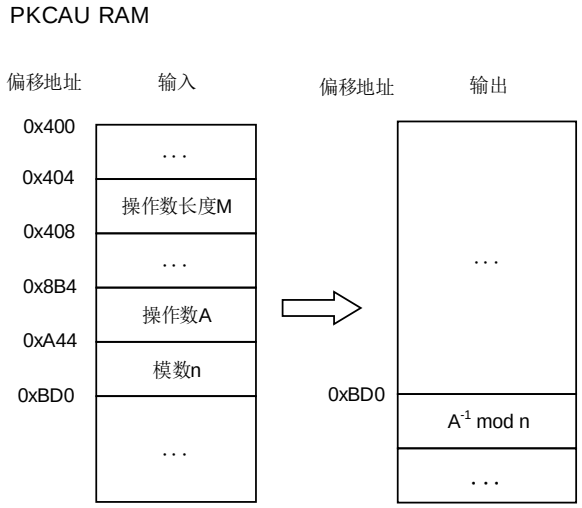


其中，0≤A<n，0≤e<n，0≤result<n， 0<n<2^M， 0<M≤3136，0<蒙哥马利参数(R^2 mod n)<n。

模逆运算

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“001000”，可以选择运算模式为模逆运算，运算说明如*图 22-17. 模逆运算*所示。运算结果为result = A^{-1} mod n。

图 22-17. 模逆运算



其中，0<A<n，0<result<n， 0<n<2^M，0<M≤3136。

注意：

1、如果模数 n 是素数，满足条件1≤A<n的所有 A 的值，都有有效的模逆输出；

2、如果模数 n 不是素数，当 A 和 n 的最大公约数为 1 时，才会有有效的模逆输出。

RSA CRT 求幂

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“000111”，可以选择运算模式为 RSA CRT 求幂。

p 和 q 是私钥的一部分，均为素数

$d_P = d \bmod (p-1)$

$d_Q = d \bmod (q-1)$

$q_{inv} = q^{-1} \bmod p$

以上的参数允许接收方更有效地计算求幂$m = A^d \pmod{pq}$：

$m = A^d \pmod{pq}$

$m_1 = A^{dP} \bmod p$

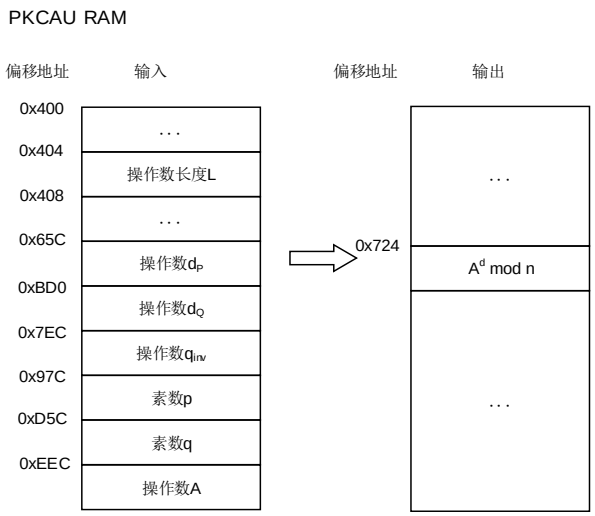
$m_2 = A^{dQ} \bmod p$

$h = q_{inv}(m_1 - m_2) \bmod p,\ m_1 > m_2$

$m = m_2 + hq$

运算说明如*图 22-18. RSA CRT 求幂*所示。运算结果为$result = A^d \bmod pq$。

图 22-18. RSA CRT 求幂



RSA CRT 求幂参数取值范围如*表 22-3. RSA CRT 求幂参数取值范围*所示。

表 22-3. RSA CRT 求幂参数取值范围

参数		取值范围
输入	操作数 d_P	$0 \le d_P < 2^{L/2}$
	操作数 d_Q	$0 \le d_Q < 2^{L/2}$
	操作数 q_{inv}	$0 < q_{inv} < 2^{L/2}$
	素数 p	$0 < p < 2^{L/2}$

参数		取值范围
	素数 q	$0<q<2^{L/2}$
	操作数 A	$0\leq A<2^{L}$
输出	运算结果：A^d mod pq	$0\leq result<pq$

22.3.5. Fp 域椭圆曲线运算模式

通过配置 PKCAU_CTL 寄存器中的 MODSEL[5:0]来选择 Fp 域椭圆曲线相关运算模式。可选运算模式如*表 22-4. 椭圆曲线运算模式选择*。

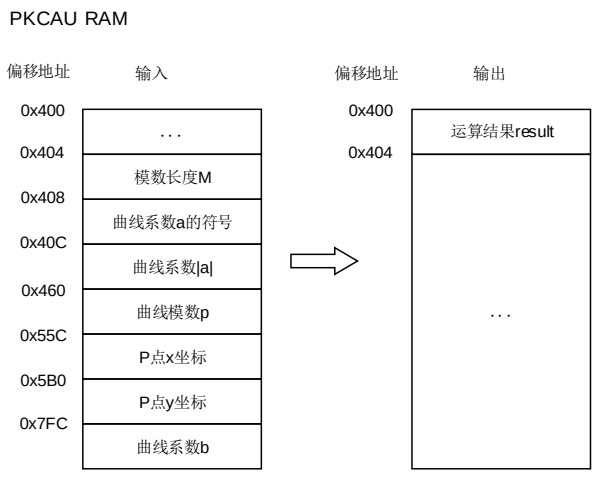
表 22-4. 椭圆曲线运算模式选择

MODSEL[5:0]	运算模式
100000	先进行蒙哥马利参数计算，然后进行 ECC 标量乘法
100010	只进行 ECC 标量乘法（蒙哥马利参数必须预先加载）
100100	ECDSA 签名
100110	ECDSA 验证
101000	椭圆曲线在素域 Fp 上点的检查

椭圆曲线在素域 Fp 上点的检查

该运算用于检查点 P(x,y)是否在素域方程$y^2= x^3+ax+b$ mod p上，其中 a，b 为曲线系数。将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“101000”，可以选择运算模式为检查椭圆曲线在 Fp 域上点，运算说明如*图 22-19. 椭圆曲线在 Fp 域上点的检查*所示。运算结果如果为 0，则表明 P 点在椭圆曲线上；如果不为 0，则表明 P 点不在椭圆曲线上。

图 22-19. 椭圆曲线在 Fp 域上点的检查



椭圆曲线在 Fp 域上点的检查范围如*表 22-5. 椭圆曲线在 Fp 域上点的检查参数取值范围*所示。

表 22-5. 椭圆曲线在 Fp 域上点的检查参数取值范围

输入参数	取值范围
模数长度 M	$0<M\leq 640$

输入参数	取值范围
曲线系数 a 的符号	0x0：正数 0x1：负数
曲线系数|a|	绝对值|a|<p
曲线系数 b	绝对值|b|<p
曲线模数 p	奇素数0<p≤2^M
P 点 x 坐标	x<p
P 点 y 坐标	y<p

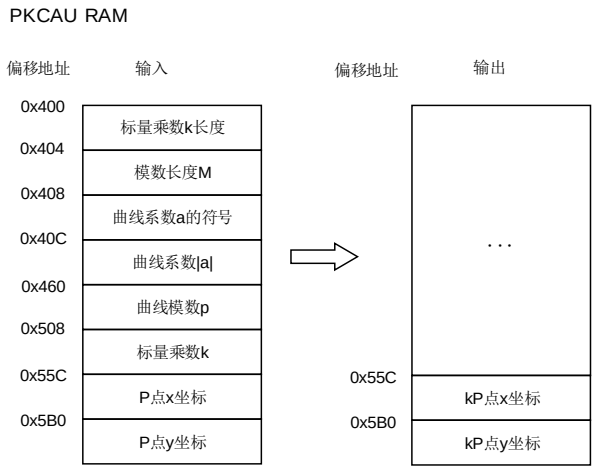
ECC 标量乘法

ECC 标量乘法操作ak×P(x_P,y_P)，其中 P 是椭圆曲线在素域 Fp 上的点，计算结果依然在曲线上，或者是无穷远点。

普通模式

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“100000”，可以选择运算模式为先进行蒙哥马利参数计算，然后进行 ECC 标量乘法，运算说明如*图 22-20. 普通模式 ECC 标量乘法*所示。

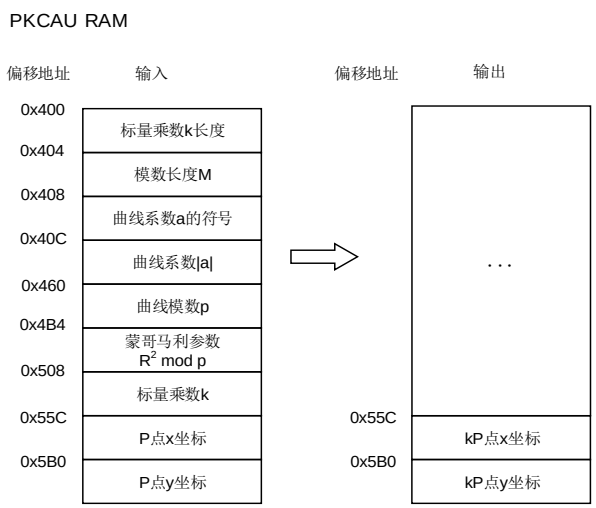
图 22-20. 普通模式 ECC 标量乘法



快速模式

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“100010”，可以选择运算模式为只进行 ECC 标量乘法，运算说明如*图 22-21. 快速模式 ECC 标量乘法*所示。

图 22-21. 快速模式 ECC 标量乘法



ECC 标量参数取值范围如*表 22-6. ECC 标量乘法参数取值范围*所示。

表 22-6. ECC 标量乘法参数取值范围

参数		取值范围
输入	标量乘数 k 的长度 LEN	0<LEN≤640
	模数长度 M	0<M≤640
	曲线系数 a 的符号	0x0：正数 0x1：负数
	曲线系数|a|	绝对值|a|<p
	曲线模数 p	奇素数0<p≤2^M
	标量乘数 k	0≤k<2^{LEN}（k < n，n 是曲线的素数阶）
	P 点 x 坐标x_P	x_P<p
	P 点 y 坐标y_P	y_P<p
输出	kP 点 x 坐标 x	x<p
	kP 点 y 坐标 y	y<p

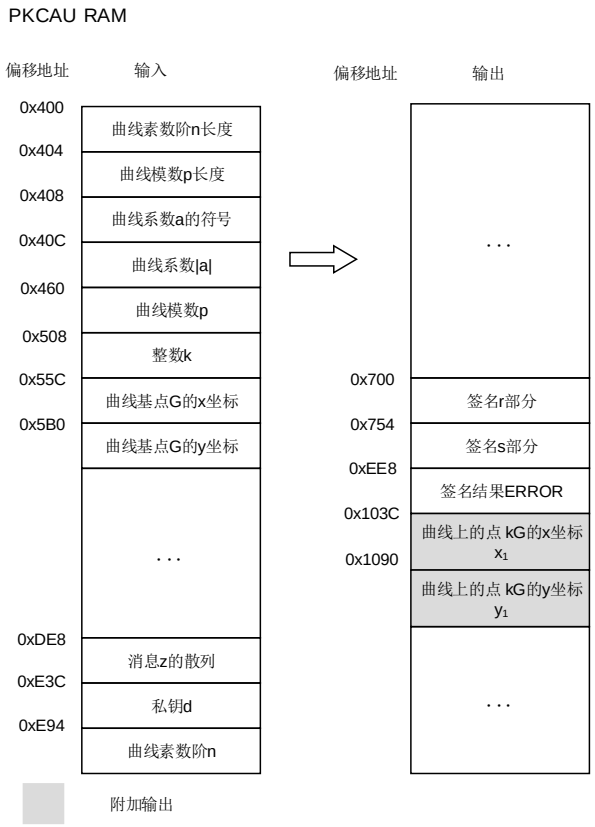
如果 k=0，输出是无穷远处的一点。当 k 是曲线素数阶 n 的倍数时，输出也是无穷远处的一点。在这个模块中，如果结果是无穷远处的一个点，则输出为(0,0)。

如果 k<0，则 k 的绝对值代替 k 作为 ECC 标量乘法的标量乘数。计算完成后，可以用-P = (x, -y)来计算 y 的最终结果。

ECDSA 签名

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“100100”，可以选择运算模式 ECDSA 签名，运算说明如*图 22-22. ECDSA 签名*所示。

图 22-22. ECDSA 签名



ECDSA 签名参数取值范围如*表 22-7. ECDSA 签名参数取值范围*所示。

表 22-7. ECDSA 签名参数取值范围

参数		取值范围
输入	曲线素数阶 n 的长度 LEN	0<LEN≤640
	曲线模数 p 的长度 M	0<M≤640
	曲线系数 a 的符号	0x0：正数 0x1：负数
	曲线系数|a|	绝对值|a|<p
	曲线模数 p	奇素数0<p<2^M
	整数 k	0≤k<2^{LEN}
	曲线基点 G 的 x 坐标	x<p
	曲线基点 G 的 y 坐标	y<p
	消息 z 的散列	z<2^{LEN}
	私钥 d	正整数 d < n
	曲线素数阶 n	素数n<2^{LEN}
输出	签名 r 部分	0<r<n
	签名 s 部分	0<s<n
	签名结果 ERROR	0x0：无错误 0x1：签名 r 部分为 0

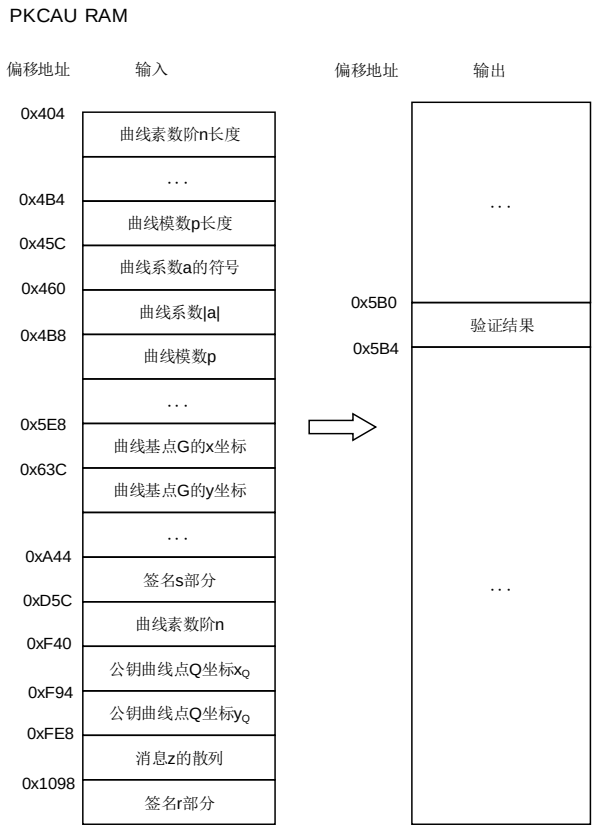
参数		取值范围
		0x2：签名 s 部分为 0
	曲线上的点 kG 的坐标 x_1	$0 \le x_1 < n$
	曲线上的点 kG 的坐标 y_1	$0 \le y_1 < n$

如果签名输出结果不为 0，则应该清除 PKCAU RAM 的内容，以避免泄漏私钥相关信息。

ECDSA 验证

将 PKCAU_CTL 寄存器中的 MODSEL[5:0]配置为“100110”，可以选择运算模式 ECDSA 验证，运算说明如*图 22-23. ECDSA 验证*所示。

图 22-23. ECDSA 验证



ECDSA 验证参数取值范围如*表 22-8. ECDSA 验证参数取值范围*所示。

表 22-8. ECDSA 验证参数取值范围

参数		取值范围
输入	曲线素数阶 n 的长度 LEN	$0 < LEN \le 640$
	曲线模数 p 的长度 M	$0 < M \le 640$
	曲线系数 a 的符号	0x0：正数 0x1：负数
	曲线系数|a|	绝对值$|a| < p$
	曲线模数 p	奇素数$0 < p < 2^M$

参数		取值范围
	曲线基点 G 的 x 坐标	x<p
	曲线基点 G 的 y 坐标	y<p
	公钥曲线点 Q 坐标x_Q	x_Q<p
	公钥曲线点 Q 坐标y_Q	y_Q<p
	签名 r 部分	0<r<n
	签名 s 部分	0<s<n
	消息 z 的散列	z<2^{LEN}
	曲线素数阶 n	素数n<2^{LEN}
输出	签名验证结果	0x0：有效签名 非 0x0：无效签名

22.3.6. PKCAU 运算流程

将 PKCAU_CTL 寄存器中的 PKCAUEN 位置 1 可以使能 PKCAU 外设。当 PKCAU 正在进行计算时，将 PKCAUEN 清 0，这种情况下，将终止正在进行的操作，并且 PKCAU RAM 中的内容将无法得到保证。

当 PKCAUEN = 0 时，应用程序仍然可以通过 AHB 接口访问 PKCAU RAM。

普通模式运算流程

以下流程适用于 PKCAU_CTL 寄存器 MODSEL[5:0]列出来的所有操作。

1、 系统复位后，PKCAU RAM 全片擦除。在这个过程中，PKCAU_STAT 寄存器中 BUSY 置 1。所有对 PKCAU RAM 的操作都应该在 BUSY 位为 0 时才执行；
2、 将初始数据加载到位于偏移地址 0x400 的 PKCAU RAM 中；
3、 在 PKCAU_CTL 寄存器 MODSEL[5:0]中写入要执行的操作，然后将 PKCAU_CTL 寄存器中将 START 位置 1；
4、 等待 PKCAU_STAT 寄存器中的 ENDF 位置 1；
5、 从 PKCAU RAM 中读取结果，然后通过在 PKCAU_STATC 中将 ENDFC 位置 1 来清除 ENDF 位。

快速模式运算流程

快速模式就是在计算很多具有相同模数的操作时，只计算一次蒙哥马利参数。在执行操作时，加载预先计算的蒙哥马利参数来进行计算。

快速模式流程如下：

1、 在位于偏移地址 0x400 的 PKCAU RAM 中加载初始数据；
2、 在 PKCAU_CTL 寄存器中配置 MODSEL[5:0] = 000001，选择蒙马参数计算模式，然后将 START 位置 1；
3、 等待 PKCAU_STAT 寄存器中的 ENDF 位置 1；
4、 从 PKCAU RAM 中读取蒙马参数，然后通过在 PKCAU_STATC 中将 ENDFC 位置 1 来清除 ENDF 位；
5、 在 PKCAU RAM 中加载初始数据以及蒙哥马利参数；

6、 在 PKCAU_CTL 寄存器 MODSEL[5:0]中写入要执行的操作，然后将 PKCAU_CTL 寄存器中将 START 位置 1；
7、 等待 PKCAU_STAT 寄存器中的 ENDF 位置 1；
8、 从 PKCAU 内部 RAM 中读取结果，然后通过在 PKCAU_STATC 中将 ENDFC 位置 1 来清除 ENDF 位。

22.3.7. 计算时间

下表总结了以时钟周期表示的 PKCAU 计算时间。

表 22-9. 模幂计算时间

幂长度（位）	模式	操作数长度（位）		
		1024	2048	3072
1024	标准	6780000	-	-
	快速	6701000	-	-
	CRT	1853000	-	-
2048	标准	-	52196000	-
	快速	-	51910000	-
	CRT	-	13651000	-
3072	标准	-	-	182783000
	快速	-	-	181953000
	CRT	-	-	44905000

表 22-10. ECC 标量乘法计算时间

模式	模数长度（位）					
	160	192	256	320	384	512
标准	626000	951000	1997000	3617000	5762000	13134000
快速	623000	946000	1990000	3607000	5749000	13111000

表 22-11. ECDSA 签名平均计算时间

模数长度（位）					
160	192	256	320	384	512
634000	966000	2029000	3648000	5833000	13177000

表 22-12. ECDSA 验证平均计算时间

模数长度（位）					
160	192	256	320	384	512
1261000	1901000	3997000	7225000	11477000	26287000

表 22-13. 蒙哥马利参数平均计算时间

模数长度（位）								
160	192	256	320	384	512	1024	2048	3072
3873	4658	7109	10330	14526	22301	79116	284359	626909

22.3.8. 状态、错误和中断

PKCAU 有一些状态、错误标志位和中断，通过设置一些寄存器位，便可以通过这些标志触发中断。

- 访问地址错误（ADDRERR）：

当访问的 PKCAU RAM 地址超出预期范围，PKCAU_STAT 寄存器中地址错误标志位 ADDRERR 位将置 1。如果 PKCAU_CTL 寄存器中的 ADDRERRIE 位置 1，将产生一个中断。将 PKCAU_STATC 寄存器中的 ADDRERRC 置 1 可以清除 ADDRERR 位。

- RAM 错误标志（RAMERR）：

当 PKCAU 内核在使用 PKCAU RAM 时，AHB 也在访问 PKCAU RAM，PKCAU_STAT 寄存器中地址错误标志位 RAMERR 位将置 1。如果此时 AHB 读 PKCAU RAM 将返回 0，写将被忽略。如果 PKCAU_CTL 寄存器中的 RAMERRIE 位置 1，将产生一个中断。将 PKCAU_STATC 寄存器中的 RAMERRC 置 1 可以清除 RAMERR 位。

- PKCAU 运算结束标志（ENDF）：

当 PKCAU 完成在 PKCAU_CTL 寄存器 MODSEL[5:0]中指定的操作时，ENDF 将置 1。如果 PKCAU_CTL 寄存器中的 ENDIE 位置 1，将产生一个中断。将 PKCAU_STATC 寄存器中的 ENDFC 置 1 可以清除 ENDF 位。如果通过设置 START 位执行另一个运算，ENDF 位将由硬件自动清除。

PKCAU 中断事件和标志如*表 22-14. PKCAU 中断请求*所示：

表 22-14. PKCAU 中断请求

中断事件	事件标志	标志清除	使能控制位
访问地址错误	ADDRERR	ADDRERRC	ADDRERRIE
RAM 错误	RAMERR	RAMERRC	RAMERRIE
运算结束标志	ENDF	ENDFC	ENDIE

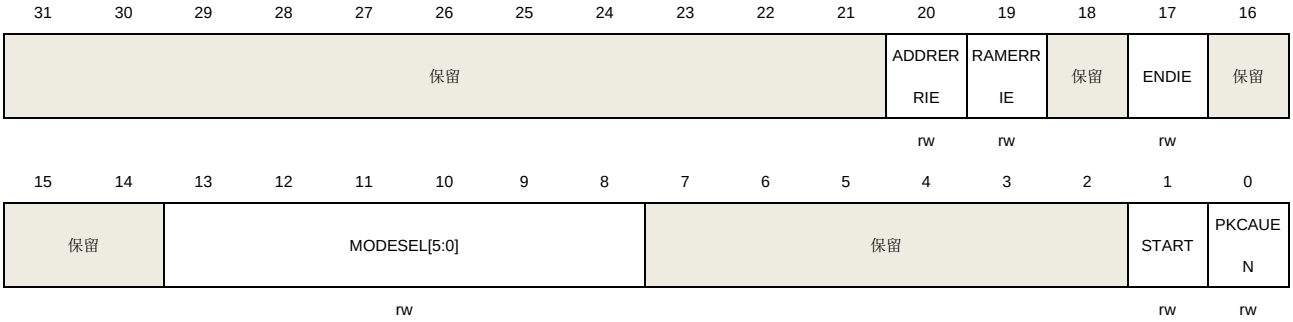
22.4. PKCAU 寄存器

PKCAU 基地址：0x4C06 1000

22.4.1. 控制寄存器（PKCAU_CTL）

地址偏移：0x00
复位值：0x0000 0000

该寄存器可以按字（32 位）访问。



位/位域	名称	描述
31:21	保留	必须保持复位值。
20	ADDRERRIE	地址错误中断使能 0：地址错误中断禁能 1：地址错误中断使能
19	RAMERRIE	RAM 错误中断使能 0：RAM 错误中断禁能 1：RAM 错误中断使能
18	保留	必须保持复位值。
17	ENDIE	运算结束中断使能 0：运算结束中断禁能 1：运算结束中断使能
16:14	保留	必须保持复位值。
13:8	MODESEL	PKCAU 运算模式选择 000000：蒙哥马利参数计算然后模幂 000001：只进行蒙哥马利参数计算 000010：只进行模幂运算（蒙哥马利参数必须预先加载） 000111：RSA CRT 求幂 001000：模逆运算 001001：算术加法 001010：算术减法

		001011：算术乘法 001100：算术比较 001101：取模 001110：模加法 001111：模减法 010000：蒙哥马利乘法 100000：先进行蒙哥马利参数计算，然后进行 ECC 标量乘法 100010：只进行 ECC 标量乘法（蒙哥马利参数必须预先加载） 100100：ECDSA 签名 100110：ECDSA 验证 101000：椭圆曲线 Fp 上点的检查 其他值保留。
7:2	保留	必须保持复位值。
1	START	PKCAU 开始运算 该位由软件置 1 来启动 PKCAU 运算，运算模式在 PKCAU_CTL 寄存器的 MODSEL[5:0]中指定的。 当 PKCAU_STAT 寄存器中 BUSY 位为 1，对该位写 1 无效。
0	PKCAUEN	PKCAU 使能 0：PKCAU 禁能 1：PKCAU 使能

22.4.2. 状态寄存器（PKCAU_STAT）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按字（32 位）访问。

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

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

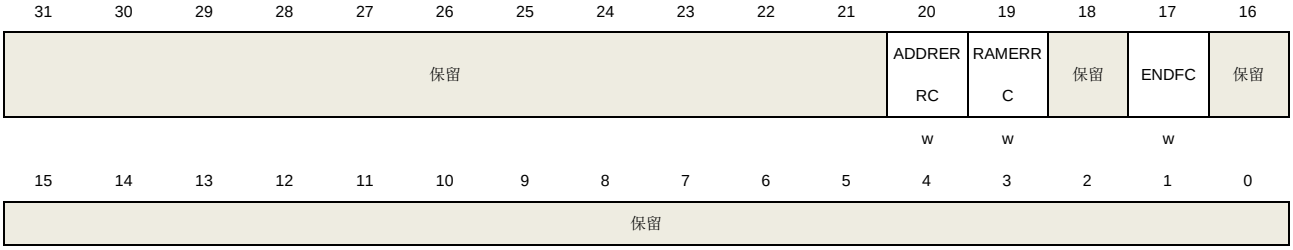
位/位域	名称	描述
31:21	保留	必须保持复位值。
20	ADDRERR	地址错误 0：无地址错误 1：访问的 PKCAU RAM 地址超出预期范围，产生地址错误。
19	RAMERR	PKCAU RAM 错误

		0：未产生 PKCAU RAM 错误 1：当 PKCAU 内核在使用 PKCAU RAM 时，AHB 也在访问 PKCAU RAM，将产生 PKCAU RAM 错误。
18	保留	必须保持复位值。
17	ENDF	PKCAU 运算结束标志 当运算执行完成，该位由硬件置 1。
16	BUSY	忙标志 当 PKCAU_CTL 寄存器中 START 位置 1，该位由硬件置 1。当 PKCAU 运算结束，该位由硬件清 0。
15:0	保留	必须保持复位值。

22.4.3. 状态清除寄存器（PKCAU_STATC）

地址偏移：0x08
复位值：0x0000 0000

该寄存器可以按字（32 位）访问。



位/位域	名称	描述
31:21	保留	必须保持复位值。
20	ADDRERRC	地址错误标志清除 软件对该位写 1 可以清除 PKCAU_STAT 寄存器中 ADDRERR 位
19	RAMERRC	PKCAU RAM 错误标志清除 软件对该位写 1 可以清除 PKCAU_STAT 寄存器中 RAMERR 位
18	保留	必须保持复位值。
17	ENDFC	PKCAU 运算结束标志清除 软件对该位写 1 可以清除 PKCAU_STAT 寄存器中 ENDF 位
16:0	保留	必须保持复位值。

23. 红外接口（IFRP）

23.1. 简介

红外接口（IFRP）用于控制红外光 LED，并发送红外数据实现红外遥控。

该模块没有寄存器，由 TIMER15 和 TIMER16 控制。IFRP_OUT 引脚可以通过 GPIO 备用功能选择寄存器进行配置。

23.2. 主要特性

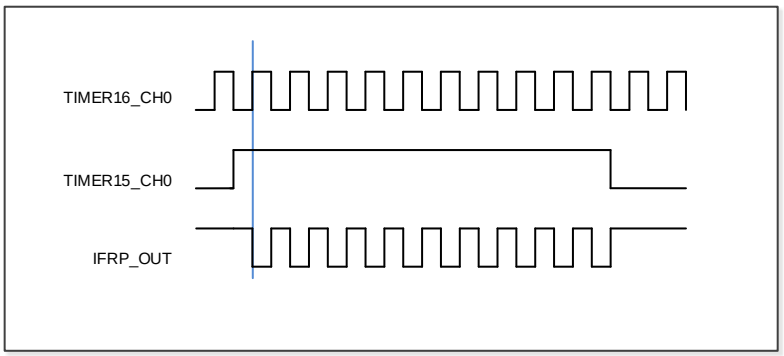
- IFRP 输出信号是由 TIMER15_CH0 和 TIMER16_CH0 决定；
- 为了得到正确的红外线信号，timer15 应产生低频调制包络信号，timer16 应产生高频载波信号。

23.3. 功能描述

IFRP 能够集成 TIMER15 和 TIMER16 的输出，以生成红外线信号。

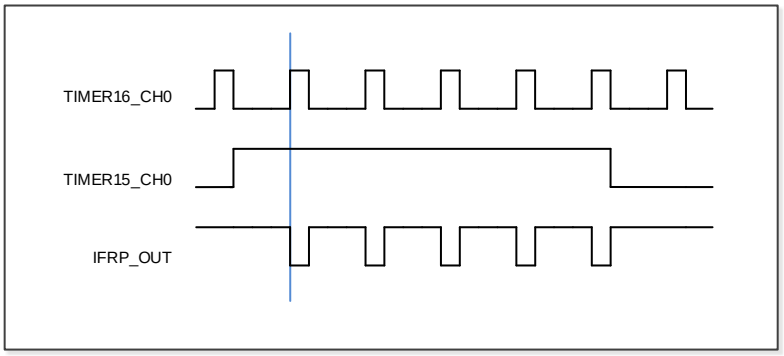
1. 对 TIMER15 的 CH0 进行编程，产生低频 PWM 信号，即调制值信号。对 TIMER16 的 CH0 进行编程，生成高频率 PWM 信号，即载波信号。在产生这些信号之前，信道需要被激活。
2. 通过 GPIO 备用功能选择寄存器配置 IFRP_OUT 引脚。

图 23-1. IFRP 输出时序图 1



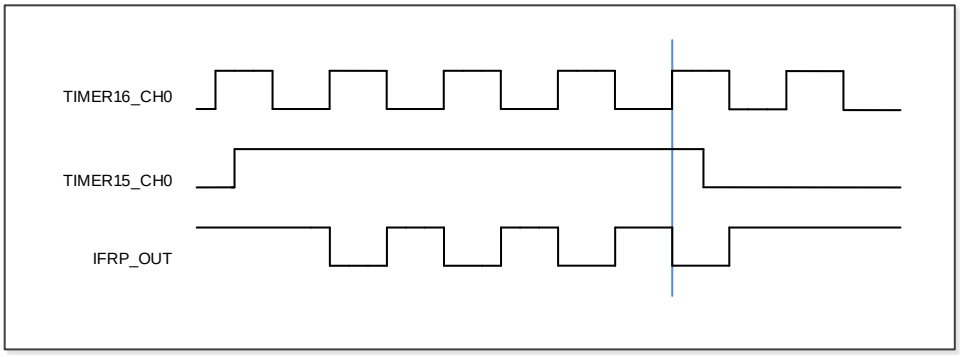
注意: IFRP_OUT 与 TIMER16_CH0 相比有一个 APB 时钟延迟。

图 23-2. IFRP 输出时序图 2



注意: 载波（TIMER15_CH0）的占空比可以改变，当 TIMER15_CH0 占空比较高时，IFRP_OUT 与 TIMER16_CH0 呈反向关系。

图 23-3. IFRP 输出时序图 3



注意: IFRP_OUT 将保持 TIMER16_CH0 的完整性，即使包络信号（TIMER15_CH0）未激活。

24. 无线

24.1. 简介

GD32VW55x 是高度集成的无线片上系统（SoC），包括一个 RISC-V 处理器，单流 Wi-Fi6 MAC 和 PHY，BLE5 链路层和调制解调器，射频收发器，功率放大器（PA）和接收低噪声放大器（LNA）。它是针对运行物联网应用程序的各种智能设备而设计的优化 SoC。

24.2. Wi-Fi

24.2.1. 主要特征

支持标准

- 兼容 802.11b / g / n /ax；
- 802.11e QoS 增强（WMM）；
- 802.11i（WPA，WPA2，WPA3）。开放、共享秘钥和成对秘钥认证服务；
- Wi-Fi WPS；
- Wi-Fi 直连；
- 集成的 TCP / IP 协议。

Wi-Fi MAC

- 目标唤醒时间（TWT）操作；
- 两个 NAV；
- 多 BSSID 操作；
- 基于 OFDMA 的随机访问；
- 空间复用；
- 聚合 MPDU（A-MPDU）的发送和接收，以实现高吞吐量；
- 支持即时 ACK 和 Block-ACK 策略；
- 支持电源管理方案，包括 WMM 节能，多轮询节能（PSMP）和多相 PSMP 操作；
- 帧间间隔定时支持，包括 RIFS；
- 支持 RTS / CTS 和 CTS 到自身的帧序列，以保护帧交换；
- 硬件回退计数器，用于支持 WMM 规范中指定的多个优先级；
- 定时同步功能（TSF），网络分配矢量（NAV）维护和信标预定传输时间（TBTT）的硬件生成；
- 用于 AES-CCMP，旧版 WPA TKIP，旧版 WEP 密码的硬件引擎，以及对密钥管理的支持；
- 可编程的独立基本服务集（IBSS）、基础结构型基本服务集或接入点功能；

Wi-Fi PHY

- 20MHz 信道中的单天线 1x1 流；
- 20M 带宽；
- 非 AP STA 上行和下行支持 MU-OFDMA；
- 非 AP STA 下行支持 MU-MIMO；
- 波束成形作为接收方；
- Rx STBC 方案（一个空间流和两个空时流）；
- Mid-amble；
- DCM；
- 所有防护间隔（0.8 / 1.6 / 3.2us）；
- 支持 802.11ax MCS 至 MCS9，最大 PHY 速率为 114.7Mbps；
- 每包 TX 功率控制；
- 高级信道估计/均衡，自动增益控制，CCA，载波/符号恢复和帧检测；
- 用于处理 CMOS RF 芯片工艺，电压和温度（PVT）变化的数字校准算法；
- 每包信道质量和信号强度测量；
- 符合 FCC 和其他全球法规要求。

24.3. BLE

24.3.1. 主要特征

支持标准

- BLE5.2。

BLE 链路层

- 支持多台硬件同时连接；
- 扩展广播；
- 高负载循环非连接广播；
- 频道选择算法#2；
- 支持多个 BLE 同时连接。

BLE 调制解调器

- 高速 2M PHY；
- 远程编码 PHY；
- 数据速率：125、500、1000、和 2000kbps。

24.4. Radio

Wi-Fi 和 BLE 共享 Radio。

- Fractional-N 支持多个参考时钟；
- 带功耗控制的集成功率放大器；
- 优化的 Tx 增益分布以实现线性和噪声性能；
- 直接转换架构;
- 具有优化噪声系数的片内增益可选 LNA；
- 高动态范围 AGC。

25. 附录

25.1. 寄存器表中使用的缩写列表

表 25-1. 寄存器功能位访问属性

寄存器表中缩写	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）	软件只能读该位，写入 1 触发事件，但对位值没有影响。
可读/可置位（rs）	软件可以读该位，也可以将这个位设置为 1。写入 0 对位值没有影响。
可读/读清零（rc_r）	软件可以读该位，读该位会自动清零。写入 0 对位值没有影响。
可读/读置位（rs_r）	软件可以读该位，读该位会置位该位。写对位值没有影响。
读/写一次（rwo）	软件只可写该位一次，可以读任意次。只有复位可将该位恢复为默认值。
读/写清零（rc_w）	软件可以读该位，对该位写可以清除这个位。写 0 和写 1 效果相同。
只读/写 1 触发（rt_w）	软件只能读该位，写 0 或 1 触发事件，但对位值没有影响。

25.2. 术语表

表 25-2. 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。
选型字节	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

术语	描述
WI	写忽略
RAZ/WI	读为 0/写忽略

25.3. 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。

26. 版本历史

表 26-1. 版本历史

版本号	描述	日期
1.0	1. 初始发布。	2023 年 10 月 13 日
1.1	1. 修改 USART_CTL2 寄存器的 DDRE 位描述，参考 ***USART 控制寄存器 2（USART_CTL2）***。 2. 修改***通道输入重映射寄存器(TIMERx_IRMP)(x=2)***寄存器位域及描述。 3. 修改***图 5-2. 时钟树***并增加时钟树注释。 4. 修改***引导配置***的描述。	2023 年 12 月 30 日
1.2	1. 修改 ***BLE 调制解调器***的描述。 2. 修改 TIMERx_INTF 寄存器中 CHOIF 位描述。 3. 修改关于 I2C ***从省电模式唤醒***的描述。 4. 修改 TIMER 的暂停模式位域描述，将 00101 值的描述改为保留。 5. 修改***模数转换器（ADC）***模块描述。 6. 修改***定时器（TIMER）***模块描述。	2024 年 3 月 25 日
1.3	1. 修改***表 1-2. GD32VW55x 系列器件的存储器映射表***，删除不可访问地址。 2. 修改 TIMER_BKIN 描述为 TIMER_BRKIN。 3. 修改***状态寄存器（RTC_STAT）***寄存器中 SCPF 比特位描述。 4. 修改 ADC 模块***单次运行模式***和***连续运行模式***章节中软件步骤的描述。 5. 修改 PKCAU 模块***控制寄存器（PKCAU_CTL）***中 BIT 17，19，20 位的描述。 6. 修改 TIMER 模块***高级定时器（TIMERx,x=0）***和***通用定时器 L0（TIMERx, x=1，2）***章节中央计数模式下的时序图。 7. 删除***设备电子签名***小节。 8. 修改***表 12-2. ADC 输入引脚定义***。 9. 删除 12.2 小节中的特性“模块供电要求：1.62V 到 3.6V，一般电源电压为 3.3V”。 10. 修改***图 4-1. 电源域概览和 VDD / VDDA 电源域***章节内容。	2025 年 2 月 13 日
1.4	1. 修改 I2C ***主机发送模式下的软件流程***章节中的 I2C_CTL0 为 I2C_CTL1。 2. 修改 USART ***半双工通信模式***章节 TX 脚配置描述为开漏模式。 3. 修改***从机地址寄存器 1（I2C_SADDR1）***中 ADDMSK2[2:0]位域描述。	2025 年 8 月 1 日

	4. 修改 I2C_CTL2 寄存器的复位值。	

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 according to the laws of the People’s Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. 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.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its 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 Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.