DPM32M08x

ARM® Cortex®-M0 32 位 MCU

用户手册

(版本：V1.3)

目录

1 文档约定

1.1 寄存器相关缩写词列表

寄存器说明中使用以下缩写词：

- rw：可读可写，读取时返回真实值。
- wo：只写，读取时返回默认值。
- ro：只读，读取时返回真实值。写入无效。
- w1c：可读且写 1 清 0，读取时返回真实值，写 1 会清 0 该 bit 值。
- rc：只读，读取时返回真实值后清 0 该 bit 值。
- Res.：保留字段/域。

1.2 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

- 字： 32 位数据。
- 半字：16 位数据。
- 字节：8 位数据。
- AHB：AMBA(Advanced Microcontroller Bus Architecture)高级高性能总线。
- APB：AMBA 高级外设总线。
- ICP： 在电路编程，指通过 JTAG、SWD 等调试接口对电路板上 MCU 的 Flash 进行编程。
- ISP：在系统编程，一般指用户通过 UART 等通信接口向 MCU 发起 ISP 指令，再由 MCU 的启动引导程序调用相应的 Flash 擦写函数完成对 Flash 的擦除和编程操作。
- IAP：一般指在用户程序运行期间对 MCU 的 Flash 进行重新编程的操作。

1.3 外设的可用性

有关各型号产品的外设可用性以及数量信息，请参见相关器件数据手册。

2 存储器和总线架构

2.1 系统架构

主系统包括：
主器件：
– Cortex®-M0 内核
– 直接存储器访问控制器（DMA）
从器件：
– 内部 SRAM
– 内部 Flash
– AHB-APB 桥，用于连接多数 APB 外设

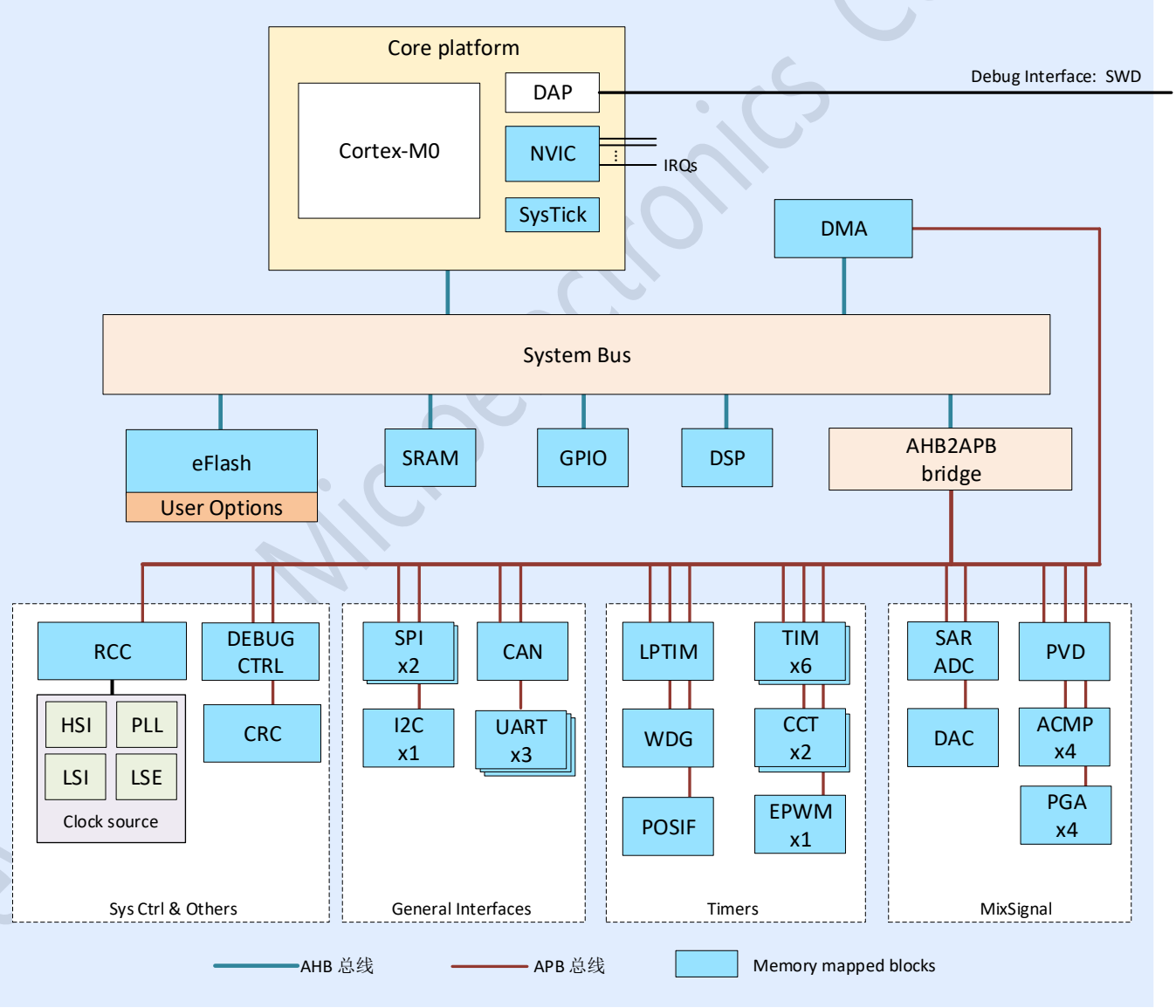

图 2-1 系统架构

GPIO，DSP 连接在系统总线上，访问速度较快。其余多数外设通过 AHB-APB 桥与系统总线互连。

Cortex®-M0 内核与 DMA 作为主设备，连接到系统总线矩阵，总线矩阵管理着内核访问和 DMA 访问之间的仲裁，默认内核的优先级高于 DMA。
总线矩阵支持 PCGE（peripheral clock gated error），当经总线访问外设的时钟关闭时，产生系统总线错误，CPU 会进入 Hard fault 中断。访问外设之前，必须先在 RCC_CLKEN_CR 寄存器中使能其时钟。
AHB-APB 桥：
AHB-APB 桥可在系统总线 AHB 与外设 APB 总线之间实现协议转换和同步连接，系统总线为 HCLK，APB 总线外设为 PCLK，在复位和时钟控制(RCC)中可配置分频比。
总线支持字节（8 位），半字（16 位）或字（32 位）访问。

2.2 存储器构成

2.2.1 系统存储空间映射

程序存储器，数据存储器，寄存器和 I/O 接口排列在地址连续的 4 GB 地址空间内。
系统的存储空间映射如表 2-1 所示。
内置最大 16K Bytes SRAM。支持字节(8 位)、半字(16 位)或字(32 位)访问。
内置最大 128K Bytes 主闪存存储器，Flash 控制器支持指令预取加速 CPU 代码执行，可在 Flash 控制器中配置使能。
用户选项字节（User Option Bytes）可用于存储保护用户配置选项，通过 Flash 控制器进行烧写和擦除。
APB 和 AHB 外设占用 512KB 字节空间。

表 2-1 系统存储空间映射

开始地址	结束地址	大小	存储区分配
0x0000_0000	0x0FFF_FFFF	128MB	重映射空间
0x1000_0000	0x1000_17FF	6KB	Reserved
0x1000_1800	0x1000_1BFF	1KB	User Option Bytes
0x1000_1C00	0x17FF_FFFF		Reserved
0x1800_0000	0x1801_FFFF	128KB	嵌入式 eFlash
0x1802_0000	0x1FFF_FFFF		Reserved
0x2000_0000	0x2000_3FFF	16KB	SRAM
0x2000_4000	0x3FFF_FFFF		Reserved
0x4000_0000	0x4007_FFFF	512KB	128 x 4KB APB 外设
0x4008_0000	0x400F_FFFF	512KB	128 x 4KB AHB 外设
0x4010_0000	0xDFFF_FFFF		Reserved
0xE000_0000	0xE00F_FFFF	1MB	Cortex-M0 peripherals
0xE010_0000	0xFFFF_FFFF		Reserved

2.2.2 AHB 外设存储器映射

表 2-2 AHB 外设存储器映射

分类	开始地址	大小	存储区分配
GPIOs	0x40080000	4KB	GPIO A
	0x40081000	4KB	GPIO B
	0x40082000	4KB	GPIO C
	0x40083000	4KB	GPIO D
	0x40084000	4KB	Reserved
	0x40085000	4KB	Reserved
	0x40086000	4KB	Reserved
	0x40087000	4KB	Reserved
DSP	0x40088000	4KB	电机算法加速 DSP（Motor Turbo DSP）
eFlash 控制器	0x40089000	4KB	嵌入式 Flash 控制器
保留	0x4008A000	4KB	Reserved
	0x4008B000	4KB	Reserved
	0x4008C000	4KB	Reserved
	0x4008D000	4KB	Reserved
	0x4008E000	4KB	Reserved
	0x4008F000	4KB	Reserved

2.2.3 APB 外设存储器映射

表 2-3 APB 外设存储器映射

分类	开始地址	大小	存储区分配
系统控制	0x40001000	4KB	复位和时钟控制(RCC)，DEBUG 控制，PGA 控制
	0x40002000	4KB	Reserved
	0x40003000	4KB	System ROM table
	0x40004000	4KB	Reserved
	0x40005000	4KB	Reserved
	0x40006000	4KB	Reserved
	0x40007000	4KB	Reserved
低速外设接口	0x40008000	4KB	UART0
	0x40009000	4KB	UART1
	0x4000A000	4KB	UART2
	0x4000B000	4KB	Reserved

	0x4000C000	4KB	Reserved
	0x4000D000	4KB	Reserved
	0x4000E000	4KB	SPI0
	0x4000F000	4KB	SPI1
	0x40010000	4KB	Reserved
	0x40011000	4KB	Reserved
	0x40012000	4KB	I2C
	0x40013000	4KB	Reserved
	0x40014000	4KB	CAN
	0x40015000	4KB	Reserved
	0x40016000	4KB	Reserved
	0x40017000	4KB	Reserved
	0x40018000	4KB	Reserved
定时器	0x40019000	4KB	TIM0
	0x4001A000	4KB	TIM1
	0x4001B000	4KB	TIM2
	0x4001C000	4KB	TIM3
	0x4001D000	4KB	TIM4
	0x4001E000	4KB	TIM5
	0x4001F000	4KB	CCT0
	0x40020000	4KB	CCT1
	0x40021000	4KB	Reserved
	0x40022000	4KB	Reserved
	0x40023000	4KB	EPWM
	0x40024000	4KB	Reserved
	0x40025000	4KB	Reserved
	0x40026000	4KB	Reserved
	0x40027000	4KB	WDG
	0x40028000	4KB	Reserved
	0x40029000	4KB	POSIF (HALL/编码器)
	0x4002A000	4KB	Reserved
	0x4002B000	4KB	Reserved
	0x4002C000	4KB	Reserved
	0x4002D000	4KB	LPTIM
模拟外设	0x40033000	4KB	SAR ADC
	0x40034000	4KB	Reserved
	0x40035000	4KB	DAC
	0x40036000	4KB	Reserved
	0x40037000	4KB	Reserved
	0x40038000	4KB	Reserved
	0x40039000	4KB	PVD
	0x4003A000	4KB	ACMP
	0x4003B000	4KB	Reserved
其他	0x4003C000	4KB	CRC

0x4003D000	4KB	DMA

2.2.4 物理地址重映射 REMAP

芯片支持应用软件修改 CPU 执行代码访问的存储器位置，通过配置实现对 CPU 访问地址的重映射功能。即可将 CPU 发出的 0x00000000 地址访问映射到不同的存储器区域。可在 RCC_SYSTEM_CR 寄存器中配置 REMAP 来实现将 CPU 访问地址的硬件映射。

（注：在配置 RCC 寄存器时，必须对 RCC_LOCK_CR 寄存器写 0x900D0000 解除锁定，才能配置其他 RCC 寄存器。）

芯片在电源复位或按键复位之后，开始执行用户程序。

芯片在系统复位（非电源复位或按键复位）之后，不会复位物理重映射寄存器 REMAP。

表 2-4 物理地址重映射

REMAP 配置	地址映射	映射区域
00	CPU 发出的 0x00xx_xxxx 地址被映射到 0x18xx_xxxx（从 Flash 启动）	Flash
01	CPU 发出的 0x00xx_xxxx 地址被映射到 0x20xx_xxxx（从 SRAM 启动）	SRAM

2.2.5 ISP 编程

在上电复位或按键复位后，用户可通过串口 UART0 向 MCU 发送特定指令进行 ISP 编程（复位释放之后 10ms 内），详见 ISP 文档。

3 嵌入式 Flash (eFlash)

3.1 简介

嵌入式闪存，支持最大 128K Bytes 主存储区，1K Bytes 选项字节（Option Bytes）。支持读，页擦除，整片擦除，可通过 8/16/32 bits 方式编程写入。支持 Flash 读保护、擦写保护和配置寄存器写保护。支持指令预取和缓存，加速取指执行。

3.2 主要特性

- 支持最大 128K Bytes 片上闪存
- 1K Bytes 用户选项字节
- 页大小：512 Bytes
- Flash 读操作
- 页擦除和整片擦除
- Flash 编程（8/16/32 bits）
- 支持 Flash 读保护
- 主存储区和用户选项字节区擦写保护
- 配置寄存器写保护
- 支持动态打开或关闭指令预取和缓存功能
- 支持动态调整 Flash 读等待周期和系统工作时钟频率
- 支持 ICP、ISP 和 IAP

3.3 功能说明

3.3.1 闪存结构

Flash 每个存储单元存储一个 Word (32 bits)，每个页 512 Bytes（128 Words）。
存储区域主要由主存储块，系统存储块和选项字节块组成。
- 主存储块支持最大 128K Bytes 空间。
- 选项字节块共 1K Bytes 空间（2 个页），分别用作用户选项字节块 0 和用户选项字节块 1。

主存储块可以按页为单位配置擦写保护，各个页的擦写保护使能存储在用户选项字节块 0 区域中。用户选项字节块 0 和 1 区域也可以单独配置擦写保护，擦写保护使能分别存储在用户选项字节块 0 和 1 中。

表 3-1 Flash 结构

区域	名称	地址范围	大小(Bytes)
主存储块	Page0	0x1800_0000 - 0x1800_01FF	512B
	Page1	0x1800_0200 - 0x1800_03FF	512B

	Page2	0x1800_0400 - 0x1800_05FF	512B
	.....	.....	.....
	Page255	0x1801_FE00 - 0x1801_FFFF	512B
选项字节块	用户选项字节块 0	0x1000_1800 - 0x1000_19FF	512B
	用户选项字节块 1	0x1000_1A00 - 0x1000_1BFF	512B

3.3.2 用户选项字节说明

Flash 中有两个页用作了用户选项字节块，分别为用户选项字节块 0 和用户选项字节块 1。用户选项字节块在 Flash 中的存储位置见
表 3-1。
用户选项字节块的定义见表 3-2 和表 3-3。

用户选项字节块中各字段的内容由用户根据具体应用需求写到 Flash 内各选项字节对应的存储位置。除了 USR_DATAx (x=0,1,...,125) 选项字节外，其他所有用户选项字节都在每次系统上电或按键复位后由引导装载程序（Boot ROM）自动从 Flash 中读取并加载生效。

以下是对用户选项字节块 0 各个字段定义的详细说明。
BOOTLOADER_ENTRY_ADDR：用户 Bootloader 入口地址
为支持 IAP，用户需配置该字段为用户 Bootloader 入口地址。
当该值范围在 FLASH 的空间内且用户选项字节块 0 生效时，用户程序会从该地址开始运行。

BL_VLD: BOOTLOADER_ENTRY_ADDR 字段有效标志。
BL_VLD 值为 1 且 BL_VLD_RED 值为 0 时，BOOTLOADER_ENTRY_ADDR 字段有效。
BL_VLD 值为 0 且 BL_VLD_RED 值为 1 时，BOOTLOADER_ENTRY_ADDR 字段无效。
BL_VLD 和 BL_VLD_RED 值相同时（正常情况下二者值应相反，值相同时认为异常），BOOTLOADER_ENTRY_ADDR 字段无效。

DIS_DBG：调试接口（SWD）访问禁止位
DIS_DBG 值为 0 且 DIS_DBG_RED 值为 1 时，通过调试接口（SWD）对系统进行访问被允许。
DIS_DBG 值为 1 且 DIS_DBG_RED 值为 0 时，通过调试接口（SWD）对系统的访问被禁止。
DIS_DBG 和 DIS_DBG_RED 值相同时（正常情况下二者值应相反，值相同时认为异常），通过调试接口（SWD）对系统的访问被禁止。
如果用户没有对用户选项字节块 0 编程，则系统上电复位后默认允许通过调试接口（SWD）对系统进行访问。为了确保 Flash 中用户存储的程序和数据安全，建议用户产品量产时将 DIS_DBG 配置为 1 且 DIS_DBG_RED 配置为 0 来禁止通过调试接口（SWD）对系统的任何访问。

FLASH_RP: Flash 读保护使能位
FLASH_RP 值为 1 且 FLASH_RP_RED 值为 0 时，如果调试接口（SWD）未被禁止，则：
- 通过调试接口（SWD）无法读取 Flash 主存储区和用户选项字节区的内容
- 通过调试接口（SWD）对 Flash 控制器所有寄存器的写访问无效，读访问不受影响
- 通过调试接口（SWD）对系统 SRAM 的读和写访问都无效
- 通过调试接口（SWD）对 DMA 配置/控制寄存器写访问无效，读访问不受影响

- 通过调试接口（SWD）对电机算法加速 DSP 内的指令存储器读和写访问都无效

FLASH_RP 和 FLASH_RP_RED 值相同时（正常情况下二者值应相反，值相同时认为异常），则使能 Flash 读保护，保护效果同上。

FLASH_RP 值为 0 且 FLASH_RP 值为 1 时，则 Flash 读保护关闭，无上述访问保护。

如果用户没有对用户选项字节块 0 编程，则默认关闭 Flash 读保护。

DIS_ISP：在系统编程（ISP）禁止位

DIS_ISP 值为 0 且 DIS_ISP_RED 值为 1 时，在系统编程（ISP）被允许。

DIS_ISP 值为 1 且 DIS_ISP_RED 值为 0 时，在系统编程（ISP）被禁止。

DIS_ISP 和 DIS_ISP_RED 值相同时（正常情况下二者值应相反，值相同时认为异常），在系统编程（ISP）被禁止。

如果用户没有对用户选项字节块 0 编程，则系统上电复位后默认允许在系统编程（ISP）。

DIS_RECOV：恢复出厂设置禁止位

DIS_RECOV 值为 0 且 DIS_RECOV_RED 值为 1 时，恢复出厂设置被允许。

DIS_RECOV 值为 1 且 DIS_RECOV_RED 值为 0 时，恢复出厂设置被禁止。

DIS_RECOV 和 DIS_RECOV_RED 值相同时（正常情况下二者值应相反，值相同时认为异常），恢复出厂设置被禁止。

如果用户没有对用户选项字节块 0 编程，则系统上电复位后默认允许恢复出厂设置操作。

OPB0_WP：选项字节块 0 擦写保护位

OPB0_WP 值为 0 且 OPB0_WP_RED 值为 1 时，关闭对选项字节块 0 的擦写保护。

OPB0_WP 值为 1 且 OPB0_WP_RED 值为 0 时，打开对选项字节块 0 的擦写保护。

OPB0_WP 和 OPB0_WP_RED 值相同时（正常情况下二者值应相反，值相同时认为异常），打开对选项字节块 0 的擦写保护。

当选项字节块 0 的擦写保护打开时，对选项字节块 0 的页擦除或对选项字节块 0 内任何存储单元的写操作都被禁止。如果发起对选项字节块 0 的页擦除或对选项字节块 0 内任何存储单元的写操作，则该擦除或写操作不生效，且 FLASH_SR 寄存器的 OPB_ERA_PRG_ERR 位被硬件置 1。

选项字节块 0 的擦写保护打开或关闭，不影响对 Flash 的整片擦除和恢复出厂设置操作。

MAIN_SEC_WPx[31:0] (x=0-7): 主存储区页（Page）擦写保护使能位

每个 bit 用作主存储区一个页的擦写使能，具体对应关系详见表 3-2。

当对应的 bit 值为 1 时，该页的擦写保护使能，任何对该页的擦除或对该页内任何存储单元的写操作都被禁止。如果发起对该页的擦除或对该页内任何存储单元的写操作，则该擦除或写操作不生效，且 FLASH_SR 寄存器的 MN_ERA_PRG_ERR 位被硬件置 1。

当对应的 bit 值为 0 时，该页的擦写保护关闭，按规定流程可实现对该页的擦除或对该页内任意存储单元的写操作。

当有任意一个或多个页的擦写保护使能时，如果发起对 flash 的整片擦除，则整片擦除不生效，且 FLASH_SR 寄存器的 CHP_ERA_ERR 位被硬件置 1。

OPB0_PROGRAMMED[31:0]：选项字节块 0 有效标志

任意非 0xFFFFFFFF 值表示用户选项字节块 0 生效。如用户需使用选项字节块 0，则需向该字段写入任意非 0xFFFFFFFF 值。

0xFFFFFFFF 表示用户选项字节块 0 无效。

注意：如果需要使用用户选项字节块 0，必须确保其内所有定义的字段都编程/写为用户期

待值。

以下是对用户选项字节块 1 各个字段定义的详细说明。

USR_DATAx[31:0] (x=0,1,2,...125): 用户自定义选项字节

用户自定义选项字节共包含 126 个字（存储单元）。用户可根据实际应用需求配置这些字段，并在需要时由软件读取。

OPB1_LOCKED[31:0]：选项字节块 1 擦写保护使能位

任意非 0xFFFFFFFF 值表示用户选项字节块 1 擦写保护使能。

当选项字节块 1 的擦写保护打开时，对选项字节块 1 的页擦除或对选项字节块 1 内任何存储单元的写操作都被禁止。如果发起对选项字节块 1 的页擦除或对选项字节块 1 内任何存储单元的写操作，则该擦除或写操作不生效，且 FLASH_SR 寄存器的 OPB_ERA_PRG_ERR 位被硬件置 1。

选项字节块 1 的擦写保护打开或关闭，不影响对 Flash 的整片片擦除操作和恢复出厂设置操作。

更改选项字节块 0 的方法

- 如果用户选项字节 0 写保护（OPB0_WP）关闭且调试接口（SWD）未禁止，则可以通过调试接口（SWD）来直接擦除或改写 Flash 中用户选项字节块 0 的值，然后向选项字节块 0 的 OPB0_PROGRAMMED[31:0]字段写入非 0xFFFFFFFF 的值。系统重新上电复位或按键复位后，新配置的选项字节块 0 的值会生效。
- 如果用户选项字节 0 写保护（OPB0_WP）关闭且 ISP 和调试接口（SWD）都未禁止，则可以通过 ISP 加载的程序来擦除或改写 Flash 中用户选项字节块 0 的值，并向选项字节块 0 的 OPB0_PROGRAMMED[31:0]字段写入非 0xFFFFFFFF 的值。系统重新上电复位或按键复位后，新配置的选项字节块 0 的值会生效。
- 如果用户选项字节 0 写保护（OPB0_WP）打开，则无法更改用户选项字节块 0 的配置。

更改选项字节块 1 的方法

- 如果用户选项字节 1 写保护（OPB1_LOCKED）关闭且调试接口（SWD）未禁止，则可以通过调试接口（SWD）来直接擦除或改写 Flash 中用户选项字节块 1 的值。
- 如果用户选项字节 1 写保护（OPB1_LOCKED）关闭且 ISP 和调试接口（SWD）都未禁止，则可以通过 ISP 加载的程序来擦除或改写 Flash 中用户选项字节块 1 的值。
- 如果用户选项字节 1 写保护（OPB1_LOCKED）打开，则无法更改用户选项字节块 1 的配置。

表 3-2 用户选项字节块 0 定义

地址	定义			默认值
0x10001800	BOOTLOADER_ENTRY_ADDR[31:0] 用户 Bootloader 入口地址			0xFFFFFFFF
0x10001804	Bit31 :	BL_VLD_RED	BL_VLD 冗余保护位，存 BL_VLD 的值取反	0xFFFFFFFF
	Bit 30-25 :	Reserved		
	Bit 24 :	OPB0_WP_RED	OPB0_WP 冗余保护位，存 OPB0_WP 的值取反	
	Bit 23-20 :	Reserved		
	Bit 19:	DIS_RECOV_RED	DIS_RECOV 冗余保护位，存 DIS_RECOV 的值取反	
	Bit 18:	DIS_ISP_RED	DIS_ISP 冗余保护位，存 DIS_ISP 位的值取反	
	Bit 17:	FLASH_RP_RED	FLASH_RP 冗余保护位，存 FLASH_RP 位的值取反	
	Bit 16:	DIS_DBG_RED	DIS_DBG 冗余保护位，存 DIS_DBG 位的值取反	
	Bit 15:	BL_VLD	BOOTLOADER_ENTRY_ADDR 有效标志。0：无效 1：有效	
	Bit 14-9 :	Reserved		

	Bit 8 : OPB0_WP 0：关选项字节块 0 擦写保护 1：开选项字节块 0 擦写保护 Bit 7-4 : Reserved Bit 3: DIS_RECOV 0:允许恢复出场设置 1:禁止恢复出场设置 Bit 2: DIS_ISP 0: 使能 ISP 1：禁止 ISP Bit 1: FLASH_RP 0: 关闭 Flash 读保护 1：使能 Flash 读保护 Bit 0: DIS_DBG 0：使能 Debug 1：禁止 Debug	
0x10001808	MAIN_SEC_WP0[31:0] 主存储区页 31-0 擦写保护使能， 0：关闭，1：使能擦写保护	0xFFFFFFFF
0x1000180C	MAIN_SEC_WP1[31:0] 主存储区页 63-32 擦写保护使能， 0：关闭，1：使能擦写保护	0xFFFFFFFF
0x10001810	MAIN_SEC_WP2[31:0] 主存储区页 95-64 擦写保护使能， 0：关闭，1：使能擦写保护	0xFFFFFFFF
0x10001814	MAIN_SEC_WP3[31:0] 主存储区页 127-96 擦写保护使能， 0：关闭，1：使能擦写保护	0xFFFFFFFF
0x10001818	MAIN_SEC_WP4[31:0] 主存储区页 159-128 擦写保护使能，0：关闭，1：使能擦写保护	0xFFFFFFFF
0x1000181C	MAIN_SEC_WP5[31:0] 主存储区页 191-160 擦写保护使能，0：关闭，1：使能擦写保护	0xFFFFFFFF
0x10001820	MAIN_SEC_WP6[31:0] 主存储区页 223-192 擦写保护使能，0：关闭，1：使能擦写保护	0xFFFFFFFF
0x10001824	MAIN_SEC_WP7[31:0] 主存储区页 255-224 擦写保护使能，0：关闭，1：使能擦写保护	0xFFFFFFFF
0x10001828	保留：0x10001828 - 0x100019F8	0xFFFFFFFF
0x100019FC	OPB0_PROGRAMMED[31:0] 非 0xFFFFFFFF 表示用户选项字节块 0 生效	0xFFFFFFFF

表 3-3 用户选项字节块 1 定义

地址	定义	默认值
0x10001A00	USR_DATA0[31:0]	0xFFFFFFFF
0x10001A04	USR_DATA1[31:0]	0xFFFFFFFF
0x10001A08	USR_DATA2[31:0]	0xFFFFFFFF
……	……	0xFFFFFFFF
0x10001BF4	USR_DATA125[31:0]	0xFFFFFFFF
0x10001BF8	保留	0xFFFFFFFF
0x10001BFC	OPB1_LOCKED[31:0] 非 0xFFFFFFFF 表示用户选项字节块 1 擦写保护使能	0xFFFFFFFF

3.3.3 Flash 配置寄存器解锁

为了防止软件运行异常、外部干扰或恶意攻击造成的对 Flash 的误配置和误操作，在对 FLASH_CR、FLASH_ADDR、FLASH_DATA 和 FLASH_FAC_REC 寄存器的写操作之前，需要进行解锁操作，否则对这些寄存器的写操作无效。

解锁操作如下：

对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D，LK 字段写入 0x0。注意：上述解锁操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无法解锁）。

锁定操作如下：

对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D，LK 字段写入 0x1。注意：上述解锁操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。

每次系统复位后，FLASH_CR、FLASH_ADDR、FLASH_DATA 和 FLASH_FAC_REC 寄存器的写访问都处于锁定状态（无法写入），按上述解锁操作一次解锁成功后，如果没有再次通过上述锁定操作锁定，则解锁状态一直有效，直到下次系统复位后又会被锁定。

为了系统安全可靠，建议每次解锁并对 Flash 配置寄存器完成所需配置后，通过上述锁定操作再次锁定。
如未解锁时，对 FLASH_CR、FLASH_ADDR、FLASH_DATA 或 FLASH_FAC_REC 寄存器进行写访问，则写访问无效且 FLASH_SR 寄存器的 WR_CR_LK_ERR 位被硬件置 1。
对 Flash 所有寄存器的读访问，不受上述锁定影响。

3.3.4 读操作

CPU 模块和外部调试器（通过 SWD 接口）都可以直接发起对 Flash 存储单元的读访问（包括主存储块和选项字节块）。这些读访问都需要经过系统 AHB 总线并最终到达 Flash 存储单元。因此，读访问的速度受 AHB 总线时钟(HCLK)频率的影响。因 Flash 存储器自身有最大读取速率的限制，当 HCLK 的频率高于一定值且在没打开指令预取功能的情况下，对 Flash 的读访问需要插入等待周期（1-3 个 HCLK 周期），详见表 1-4。

表 3-4 不同总线时钟频率对应的 Flash 读等待周期

HCLK 频率	Flash 读访问等待周期（WS）
0Mhz < HCLK ≤48Mhz	1
48Mhz < HCLK ≤72Mhz	2
72Mhz < HCLK ≤96Mhz	3

系统上电复位后，默认的 HCLK 频率是 48Mhz，默认的 Flash 读等待周期配置为 3（HCLK 48Mhz 时对应的 Flash 读等待周期本来是 1，但上电复位后默认配置等待周期为 3 是为了稳定可靠）。用户需要根据实际应需求用配置 HCLK 的分频器并根据分频后 HCLK 的频率值查阅表 3-4 来配置对应的读等待周期值到 FLASH_CR 寄存器的 WS 字段。
如果当 FLASH_SR 寄存器的 BUSY 位为 1 时（表示正在进行擦除或编程操作）发起对 Flash 的读访问，则该次读操作会阻塞系统 AHB 总线，直到 BUSY 位为 0 后才能完成对 Flash 的读取操作并释放总线。

支持动态更改 CPU/系统工作频率和相对应的 Flash 的读等待周期。无需停止任何时钟和软件运行的情况下按一定配置流程即可完成更改。具体流程请参考如下：

1. 读 FLASH_SR 寄存器并判断 BUSY 位是否为 1，若为 1，则需轮询，直到 BUSY 位为 0 时才可进行后续步骤。
2. 检查并清除 FLASH_SR 寄存器历史遗留的错误标志（如有，向对应 bit 写 1 即可清除该 bit 之前置 1 的错误标志）。
3. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x0，解锁 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无法解锁）。
4. 对 FLASH_CR 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0xACCE，WS 字段写入 0x3。此操作必须是 32 bits 写访问且对 KEY 和 WS 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。
5. 对 RCC_LOCK 寄存器写 0x900D 0000，解锁 RCC 各配置寄存器的写访问权限。
6. 配置 RCC_COMMON_CR 寄存器选择系统时钟源和 HCLK 分频系数。
7. 对 FLASH_CR 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0xACCE，WS 字段写入 CPU/系统新的目标工作频率（HCLK 频率）对应的读等待周期值（根据表 3-4）此操作必须是 32 bits 写访问且对 KEY 和 WS 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。

8. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x1，重新锁定 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。

所有 CPU 模块和外部调试器（通过 SWD 接口）直接对 Flash 存储单元的写访问都无效，但不会产生总线错误。对 Flash 存储单元的写访问，需要配置 FLASH 寄存器并按照特定流程才能完成。

如果在 Flash 擦除或者编程过程中（FLASH_SR 寄存器的 BUSY 位为 1 时）发起了读操作，该笔读操作会被挂起，直到当前进行的擦除或者编程操作完成后，才能返回正确的读数据。

3.3.5 指令预取

当配置的 Flash 的读等待周期大于 0 时，每次 CPU 从 Flash 读取指令时都会插入等待周期，这样 CPU 的取指执行的效率会降低。为了提高这种情况下的 CPU 取指执行的效率，可以通过配置 FLASH_CR 寄存器的 PF_EN 位为 1 来打开指令预取功能。
当指令预取功能打开时，硬件会自动从 Flash 中预取指令到内部缓存，能够减少 CPU 取指令的等待周期，大大提高 CPU 从 Flash 取指执行的效率。
指令预取功能支持动态打开或关闭，即无需停止任何时钟和软件运行的情况下直接配置 FLASH_CR 寄存器的 PF_EN 位即可。每次系统复位后，指令预取功能默认关闭。
每当对 Flash 有任何页擦除、整片擦除、编程操作或每次关闭预取功能时，指令缓存器中的内容都会被硬件自动清除。

打开/关闭预取功能的操作流程：

1. 读 FLASH_SR 寄存器并判断 BUSY 位是否为 1，若为 1，则需轮询，直到 BUSY 位为 0 时才可进行后续步骤。
2. 检查并清除 FLASH_SR 寄存器历史遗留的错误标志（如有，向对应 bit 写 1 即可清除该 bit 之前置 1 的错误标志）。
3. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x0，解锁 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无法解锁）。
4. 对 FLASH_CR 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0xACCE, PF_EN 字段写入 0x1（打开预取）或 0x0（关闭预取）。此操作必须是 32 bits 写访问且对 KEY 和 PF_EN 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。
5. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x1，重新锁定 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。

3.3.6 页擦除

支持对主存储区和用户选项字节区按页擦除且每次操作只擦除一个页。每个页大小为 512 Bytes（128 个存储单元）。页擦除完成后，被擦除页的所有存储单元值为 0xFFFFFFFF。
页擦除具体操作流程如下：

1. 读 FLASH_SR 寄存器并判断 BUSY 位是否为 1，若为 1，则需轮询，直到 BUSY 位为 0 时才可进行后续步骤。

2. 检查并清除 FLASH_SR 寄存器历史遗留的错误标志（如有，向对应 bit 写 1 即可清除该 bit 之前置 1 的错误标志）。
3. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D，LK 字段写入 0x0，解锁 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无法解锁）。
4. 配置 FLASH_ADDR 寄存器的值为待擦除页内的任意存储单元地址（如：该页起始地址）。
   注意，这里配置的是系统存储空间上给 Flash 分配的地址空间内的地址。合法的配置值范围如下：
   0x1800_0000 - 0x1800_7FFF （主存储区）
   0x1000_1800 - 0x1000_1BFF （用户选项字节区）
5. 对 FLASH_CR 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0xACCE，OP_MODE 字段写入 0x2（页擦除）。此操作必须是 32 bits 写访问且对 KEY 和 OP_MODE 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。
6. 对 FLASH_CR 寄存器进行 32 bits 写访问,其中 KEY 字段写入 0xACCE, OP_START 字段写入 0x1，启动页擦除。此操作必须是 32 bits 写访问且对 KEY 和 OP_START 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。
7. 读 FLASH_SR 寄存器并判断 BUSY 位，若为 1，则该次页擦除在进行中，若为 0，则该次页擦除完成。
8. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D，LK 字段写入 0x1，重新锁定 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。

如果对被设置了擦写保护的页进行上述页擦除操作，则该操作无效且 FLASH_SR 寄存器中 MN_ERA_PRG_ERR(主存储区擦写错误标志)位或 OP_ERA_PRG_ERR(用户选项字节区擦写错误标志)位会被硬件置 1。

如果配置 FLASH_ADDR 寄存器[31:28]位的值不等于 0x1（超出了 Flash 地址空间范围）且执行了后续的启动页擦除操作，则该擦除无效，FLASH_SR 寄存器中 ERA_PRG_ADDR_ERR 位置 1。

3.3.7 整片擦除

支持整片擦除，一次整片擦除操作即可完成对主存储区所有存储单元（最大包含 64 页，32K Bytes）的擦除。整片擦除完成后，主存储区的所有存储单元值为 0xFFFFFFFF。整片擦除操作不会擦除系统存储区(Boot ROM)和用户选项字节区。
整片擦除具体操作流程如下：

1. 读 FLASH_SR 寄存器并判断 BUSY 位是否为 1，若为 1，则需轮询，直到 BUSY 位为 0 时才可进行后续步骤。
2. 检查并清除 FLASH_SR 寄存器历史遗留的错误标志（如有，向对应 bit 写 1 即可清除该 bit 之前置 1 的错误标志）。
3. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x0，解锁 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无法解锁）。
4. 对 FLASH_CR 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0xACCE, OP_MODE 字段写入 0x3（整片擦除）。此操作必须是 32 bits 写访问且对 KEY 和 OP_MODE 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。
5. 对 FLASH_CR 寄存器进行 32 bits 写访问,其中 KEY 字段写入 0xACCE, OP_START 字段写入 0x1，

启动整片擦除。此操作必须是 32 bits 写访问且对 KEY 和 OP_START 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。

6. 读 FLASH_SR 寄存器并判断 BUSY 位，若为 1，则该次整片擦除在进行中，若为 0，则该次整片擦除完成。
7. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x1，重新锁定 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。

如果在主存储区有任何页擦写保护使能的情况下启动整片擦除，则该操作无效且 FLASH_SR 寄存器中 CHP_ERA_LK_ERR 位会被硬件置 1。

3.3.8 编程

主存储区和用户选项字节区支持按 8 bits, 16 bits 或 32 bits 编程。
支持 ICP, ISP 和 IAP。

编程操作具体操作流程如下：

1. 擦除待编程的 Flash 存储单元地址所在的页（根据实际使用需求，使用页擦除或整片擦除，具体流程参考页擦除或整片擦除章节）。
2. 读 FLASH_SR 寄存器并判断 BUSY 位是否为 1，若为 1，则需轮询，直到 BUSY 位为 0 时才可进行后续步骤。
3. 检查并清除 FLASH_SR 寄存器历史遗留的错误标志（如有，向对应 bit 写 1 即可清除该 bit 之前置 1 的错误标志）。
4. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x0，解锁 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无法解锁）。
5. 配置 FLASH_ADDR 寄存器的值为待编程的存储单元的地址。
   注意，这里配置的是系统存储空间上给 Flash 分配的地址空间内的地址。合法的配置值范围如下：
   0x1800_0000 - 0x1800_7FFF （主存储区）
   0x1000_1800 - 0x1000_1BFF （用户选项字节区）
6. 配置 FLASH_DATA 寄存器的值为需要编程写入的 Word 值。若该 Word 包含无需编程改写的 Byte，将对应的位置写入 0xFF 即可。对值为 0xFF 的 Byte，编程时不会更改 Flash 存储单元相对应 Byte 的内容。示例如下：
   8 bits 编程：
   待编程值 0x000000AB，向 FLASH_DATA 写入 0xFFFFFFAB
   待编程值 0x0000AB00，向 FLASH_DATA 写入 0xFFFFABFF
   待编程值 0x00AB0000，向 FLASH_DATA 写入 0xFFABFFFF
   待编程值 0xAB000000，向 FLASH_DATA 写入 0xABFFFFFF
   16 bits 编程：
   待编程值 0x0000ABCD，向 FLASH_DATA 写入 0xFFFFABCD
   待编程值 0xABCD0000，向 FLASH_DATA 写入 0xABCDFFFF
   32 bits 编程：
   待编程值 0xABCD55AA，向 FLASH_DATA 写入 0xABCD55AA
7. 对 FLASH_CR 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0xACCE, OP_MODE 字段写入 0x1

（编程）。此操作必须是 32 bits 写访问且对 KEY 和 OP_MODE 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。

8. 对 FLASH_CR 寄存器进行 32 bits 写访问,其中 KEY 字段写入 0xACCE, OP_START 字段写入 0x1，启动编程操作。此操作必须是 32 bits 写访问且对 KEY 和 OP_START 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。
9. 读 FLASH_SR 寄存器并判断 BUSY 位，若为 1，则该次编程正在进行中，若为 0，则该次编程完成。
10. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x1，重新锁定 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）

如果在待编程的 Flash 存储单元所在页擦写保护使能的情况下启动对该对该存储单元的编程操作，则该操作无效且 FLASH_SR 寄存器中 MN_ERA_PRG_ERR（主存储区擦写错误标志）位或 OP_ERA_PRG_ERR(用户选项字节区擦写错误标志)位会置 1。

如果配置 FLASH_ADDR 寄存器[31:28]位的值不等于 0x1（超出了 Flash 地址空间范围）且执行了后续的启动编程的操作，则该次编程操作无效，FLASH_SR 寄存器中 ERA_PRG_ADDR_ERR 位置 1。

如果在对某个 Flash 存储单元进行编程操作前没有擦除该存储单元所在的页，则该次编程无效。

3.3.9 恢复出厂设置

恢复出厂设置会把整个 Flash 主存储区和用户选项字节区所有存储单元的数据全部擦除，无论这些区域是否配置了擦写保护。

恢复出厂设置的具体操作流程如下：

1. 读 FLASH_SR 寄存器并判断 BUSY 位是否为 1，若为 1，则需轮询，直到 BUSY 位为 0 时才可进行后续步骤。
2. 检查并清除 FLASH_SR 寄存器历史遗留的错误标志（如有，向对应 bit 写 1 即可清除该 bit 之前置 1 的错误标志）。
3. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x0，解锁 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无法解锁）。
4. 对 FLASH_FAC_REC 寄存器进行连续两次 32 bits 写访问，第一次 KEY 字段写入 0x900D, UNLK_DATA 字段写入任意数 M，第二次 KEY 字段写 0xBEEF，UNLK_DATA 字段写入~M（将 M 按位取反）。每次操作必须是 32 bits 写访问且对 KEY 和 UNLK_DATA 字段在一次写操作中完成配置（16 bits，8 bits 写访问无法解锁）。若上述连续两次访问写入的 KEY 值不正确或两次访问 UNLK_DATA 字段写入的数值不正确，则后续操作无效。
5. 对 FLASH_FAC_REC 的 UNLK_DATA 字段写入 0x1，启动恢复出厂设置操作。
6. 读 FLASH_SR 寄存器并判断 BUSY 位，若为 1，则该次恢复出厂设置操作正在进行中，若为 0，则该次恢复出厂设置完成。
7. 对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x1，重新锁定 FLASH 配置寄存器写访问权限。该操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。

如果用户选项字节里配置禁止恢复出厂设置，则上述恢复出厂设置的过程无效。

3.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 3-5 FLASH 寄存器概览

偏移地址	寄存器名	寄存器描述	复位值
0x000	FLASH_CR	Flash控制寄存器	0x00000030
0x004	FLASH_ADDR	Flash编程和擦除地址寄存器	0x00000000
0x008	FLASH_DATA	Flash编程数据寄存器	0x00000000
0x00c	FLASH_CR_LOCK	Flash配置锁定寄存器	0x00000001
0x010	FLASH_SR	Flash状态寄存器	0x00000000
0x014	FLASH_FAC_REC	Flash恢复出厂模式寄存器	0x00000000

3.4.1 Flash 控制寄存器(FLASH_CR)

偏移地址：0x000
复位值：0x00000030

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

Bit	Field	Description
31:16	KEY	FLASH_CR 寄存器写访问许可密钥。当需要更改 FLASH_CR 寄存器的其他字段的值时，必须对其进行 32 bits 写访问，KEY[15:0] 必须写入 0xACCE（其他值无效）且同时向需要更改的字段写入配置值（8 bits 和 16 bits 写访问无效）。
15:7	Res.	保留，必须保持复位值。
6	PF_EN	指令预取使能。 0：关闭指令预取 1：打开指令预取
5:4	WS	读等待周期数。需根据当前配置的系统 AHB 总线时钟（HCLK）频率来选择配置相对应的等待周期数，详见 3.3.4 读操作。 00：Reserved. 01：1 个等待周期（当 0Mhz < HCLK ≤48Mhz 时,配置此值） 10：2 个等待周期（当 48Mhz < HCLK ≤72Mhz 时,配置此值） 11：3 个等待周期（当 72Mhz < HCLK ≤96Mhz 时,配置此值）
3:1	OP_MODE	操作模式配置。 001：编程 010：页擦除

		011：整片擦除 其他值：保留
0	OP_START	操作启动控制，0：未启动 1：启动 页擦除、整片擦除和编程操作的启动控制位。当配置为 1 后启动 OP_MODE[2:0]配置值对应的操作。操作启动后，硬件自动清 0 该位。

3.4.2 Flash 编程和擦除地址寄存器(FLASH_ADDR)

偏移地址：0x004
复位值：0x00000000

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

Bit	Field	Description
31:0	ADDR	配置待擦除页的起始地址或待编程存储单元地址。这里配置的是系统存储空间上给 Flash 分配的地址空间内的地址（主存储区或用户选项字节区地址）。

3.4.3 Flash 编程数据寄存器(FLASH_DATA)

偏移地址：0x008
复位值：0x00000000

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

Bit	Field	Description
31:0	DATA	编程数据。配置待编程存储单元的待编程数据。

3.4.4 Flash 配置锁定寄存器(FLASH_CR_LOCK)

偏移地址：0x00C
复位值：0x00000001

KEY[15:0]															
wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LK
															rw

Bit	Field	Description
31:16	KEY	Flash 配置寄存器写访问解锁密钥。当需要更改 LK 位的值时，必须对 FLASH_CR_LOCK 寄存器进行 32 bits 写访问，KEY[15:0] 必须写入 0x900D（其他值无效）且同时向 LK 字段写入配置值（8 bits 和 16 bits 写访问无效）。
15:1	Res.	保留。
0	LK	Flash 配置寄存器写访问权限锁定控制。 0：关闭写权限 1：打开写权限 当此位为 0 时，对 FLASH_CR, FLASH_ADDR 和 FLASH_DATA 寄存器的写访问无效，读访问不受此位影响。

3.4.5 Flash 状态寄存器(FLASH_SR)

偏移地址：0x010
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDR_ERR	OP_WP_ERR	MN_WP_ERR	CPE_WP_ERR	CR_LK_ERR	CR_BSY_ERR	BUSY
									w1c	w1c	w1c	w1c	w1c	w1c	ro

Bit	Field	Description
31:7	Res.	保留。
6	ADDR_ERR	如果配置 FLASH_ADDR[31:28]的值不等于 0x1 并启动编程或页擦除操作，则产生此错误。
5	OP_WP_ERR	用户选项字节区页擦写保护打开时擦写错误。

		当用户选项字节区某个页的擦写保护打开时，如果对该页进行页擦除或者对该页内的存储单元编程，则操作无效且会产生此错误。
4	MN_WP_ERR	主存储区页擦写保护打开时擦写错误。 当主存储区某个页的擦写保护打开时，如果对该页进行页擦除或者对该页内的存储单元编程，则操作无效且会产生此错误。
3	CPE_WP_ERR	主存储区页擦写保护打开时整片擦除错误。 当有任何主存储区页擦写保护打开时启动整片擦除，则擦除无效且会产生此错误。
2	CR_LK_ERR	配置寄存器锁定时写入错误。 对 FLASH_CR、FLASH_ADDR、FLASH_DATA 寄存器的写操作只能在解锁后（FLASH_CR_LOCK 寄存器 LK 位为 0）进行，否则会产生此错误。
1	CR_BSY_ERR	Flash 忙时写配置寄存器错误。 对 FLASH_CR、FLASH_ADDR、FLASH_DATA 和 FLASH_CR_LOCK 寄存器的写操作只能在 Flash 不忙时（BUSY 为 0）进行，否则会产生此错误。
0	BUSY	Flash 工作忙指示位。 0：Flash 空闲 1： Flash 工作忙 当 Flash 进行页擦除、整片擦除或编程操作时，硬件自动将此位置 1。 当上述操作完成后或没有进行上述任何操作时，硬件自动将此为清 0。

3.4.6 Flash 恢复出厂模式寄存器(FLASH_FAC_REC)

偏移地址：0x014
复位值：0x00000000

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

Bit	Field	Description
31:16	KEY	Flash 恢复出厂模式解锁密钥。详见下述说明。
15:0	UNLK_DATA	Flash 恢复出厂模式解锁数据。详见下述说明。

当需要 FLASH 恢复出厂模式时，对 FLASH_FAC_REC 寄存器进行连续两次 32 bits 写访问，第一次 KEY 字段写入 0x900D，UNLK_DATA 字段写入任意数 M，第二次 KEY 字段写 0xBEEF，UNLK_DATA 字段写入~M（将 M 按位取反）。每次操作必须是 32 bits 写访问且对 KEY 和 UNLK_DATA 字段在一次写操作中完成配置（16 bits，8 bits 写访问无法解锁）。若上述连续两次访问写入的 KEY 值不正确或两次访问 UNLK_DATA 字段写入的数值不对，则后续操作无效。
上述操作后，对 FLASH_FAC_REC 的 UNLK_DATA 字段写入 0x1，启动恢复出厂模式操作。

4 电源控制

4.1 电源

MCU 采用单电源供电，工作电压(VDD)要求介于 2.5V 到 5.5V 之间，并且针对不同的外设提供有不同的电压。

- 芯片 I/O 的电压与 VDD 相同。
- 内部有嵌入式低压差线性稳压器(LDO)输出 1.5V 电压 Vcore，用于给 Cortex®-M0 内核、SRAM、Flash 和数字外设等供电。

该系列中所有集成预驱的型号都内置了 5V LDO，其输出可直接给 MCU 供电。
详细供电方案请见对应型号的数据手册。

4.2 ADC、ACMP 和 PGA 参考电压

对于模拟外设 ADC，ACMP 和 PGA，用户可配置来自以下两部分的参考电压：

- 芯片供电电压 VDD。
- 芯片内部带隙基准(Bandgap)输出电压 2.4V。

4.3 电源监控器

4.3.1 上电复位(POR)和掉电复位(PDR)

该芯片集成有上电复位(POR)和掉电复位(PDR)电路，可在对应状态下自动复位，只有外部供电电压 VDD 达到 2.0V 时芯片释放复位并开始正常工作。
当供电电压 VDD/VDDA 低于芯片的上电和掉电阈值电压 VPOR/VPDR 时，芯片自动进入复位状态，不需要额外的外部复位电路。该芯片上电复位和掉电复位的时序关系如图 4-1，关于 POR 和 PDR 的参数请参考数据手册的电气特性部分。

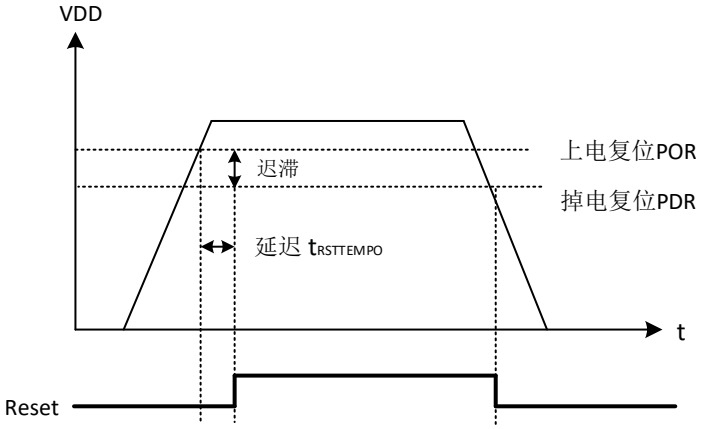

图 4-1 POR 与 PDR 复位的时序关系

4.3.2 可编程电压检测器(PVD)

该芯片支持 PVD 监视芯片 VDD 电压，当 VDD 降至 PVD 设置的电压阈值以下，或者当 VDD 升至 PVD 设置的电压阈值以上时，可以产生 PVD 中断并在中断服务程序中执行紧急关闭系统的任务，具体见 PVD 的使用参考章节。

5 复位和时钟控制(RCC)

5.1 复位

芯片有两种类型的复位，分别为系统复位和电源复位。

5.1.1 电源复位

该芯片发生如下任意事件将会产生电源复位，电源复位将所有的寄存器配置进行复位，等同于上电复位。
- 上电复位(POR)，掉电复位(PDR)。
- 外部按键复位，由 GPIO PD9 引入复位源，内部滤波器将会滤除小于 22us 的毛刺干扰信号，按键复位必须保证按键 PD9 置低电平时间大于 22us。
- MCU 受到外部强干扰引起 Flash 访问异常，触发硬件自动复位。

5.1.2 系统复位

该芯片发生如下任意事件将会产生系统复位：
- 软件复位（程序将 CPU 的应用中断控制寄存器（AIRCR）的 SYSRESETREQ 位置 1 触发)。
- CPU Lockup 复位（CPU 发生死锁，默认发生 Lockup 时芯片不复位，需要配置寄存器 RCC_SYSTEM_CR 中的位 LKUPRST_EN 为 1 来使能在 Lockup 时芯片复位)。
- DSP 看门狗复位（通过 DSP 寄存器配置使能)。
- WDG 系统看门狗复位（通过 WDG 寄存器配置使能)。
- PVD 监测电压异常复位（通过 PVD 寄存器配置使能)。

在发生复位后，软件读取 RCC_RESET_SR 寄存器中的状态位来检查复位来源，复位源标记由硬件置 1，软件写 RCC_RESET_SR 寄存器中的 CLR 位置 1 清除状态标志。
系统复位不会使已经连接的 Debugger 断开，并且 RCC_SYSTEM_CR 寄存器的值保持。

5.2 时钟

5.2.1 时钟树

芯片提供以下时钟源：
- 高速内部时钟(HSI RC 时钟)
- 锁相环(PLL 时钟)
- 低速内部时钟(LSI RC 时钟)
- 低速外部时钟(LSE OSC 时钟)

时钟树框图如下图 5-1 RCC 时钟树框图所示：
通过配置寄存器（RCC_COMMON_CR）来选择不同的时钟源作为系统时钟，也可配置分频器来分别配置 AHB 和 APB 的频率。

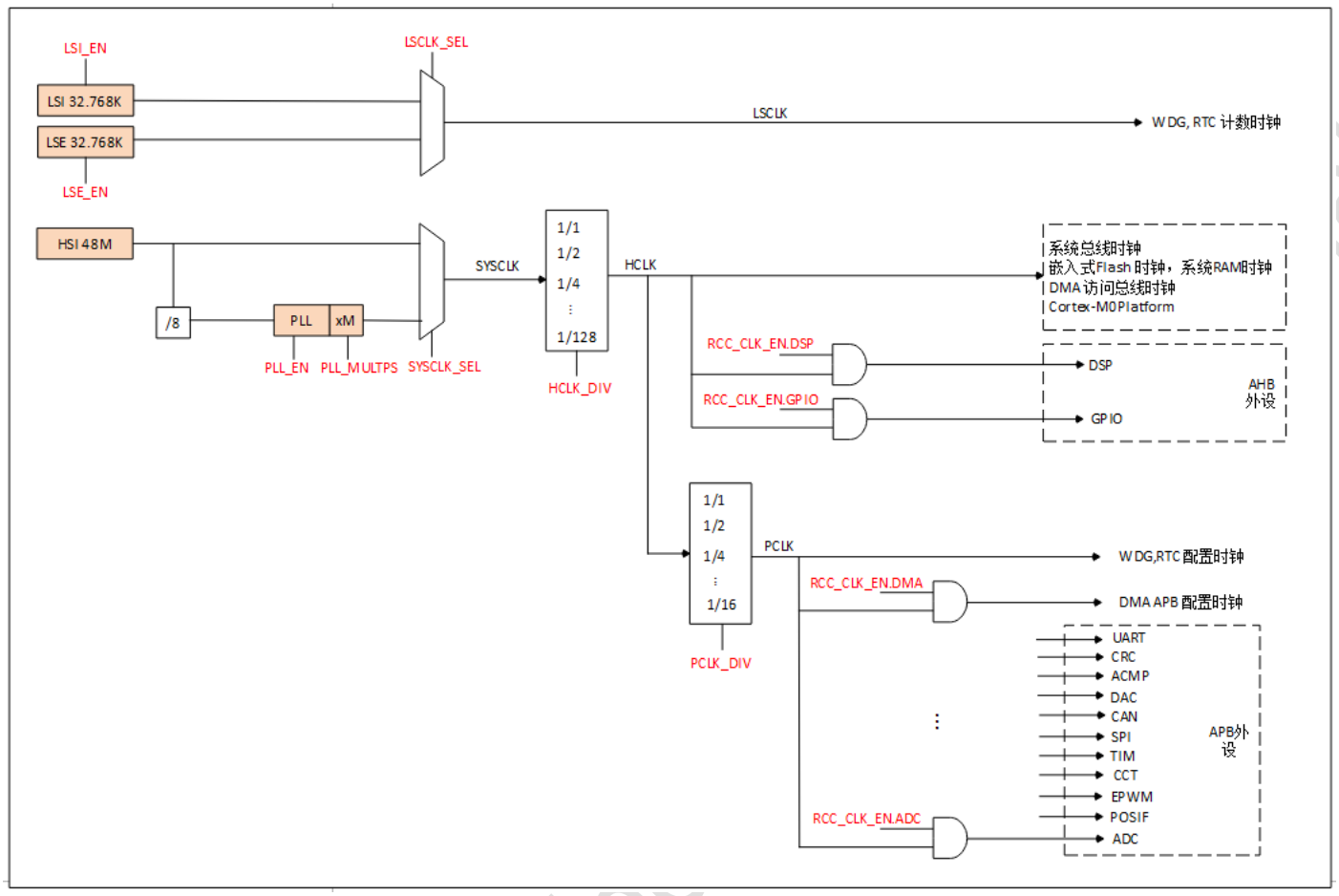

图 5-1 RCC 时钟树框图

5.2.2 HSI 时钟

HSI 时钟由内部 48MHz 振荡器产生，不支持通过软件主动关闭，只有在芯片进入低功耗 STOP 模式下时才会被硬件自动关闭降低功耗。

软件可以配置 HSI 从 STOP 模式下唤醒时的启动稳定时间（RCC_COMMON_CR 中的位 HSI_ST），硬件会等待设置的时钟周期数后从 STOP 模式恢复至正常工作。

5.2.3 PLL

PLL 的输入参考时钟是 HSI 时钟 8 分频（也即 6MHz）。

使能 PLL 时钟流程如下：

（1）软件对 RCC_LOCK_CR 写 0x900D 0000，解锁 RCC 寄存器。

（2）配置 RCC_PLL_CR 寄存器中的位 MULTPS[3:0]来设置 PLL 的倍频系数，最后配置使能 PLL。

（3）软件读取 RCC_PLL_CR 寄存器中的位 RDY，检查 PLL 时钟是否准备完成。

5.2.4 LSI 时钟

LSI 时钟是由 RC 振荡器产生的 32.768 kHz 时钟，可作为 LPTIM 和 WDG 的时钟源，具有功耗低且精度高的优点。

使能 LSI 时钟流程如下：

（1）软件对 RCC_LOCK_CR 寄存器写 0x900D 0000，解锁 RCC 寄存器。

（2）配置启动等待稳定时间 RCC_COMMON_CR 寄存器中的位 LSI_ST，也可使用默认值。

（3）配置 RCC_COMMON_CR 寄存器中的位 LSI_EN 为 1，使能 LSI。

（4）软件读取 RCC_COM_CR 寄存器中的位 LSI_RDY，等待该位置 1 时钟准备就绪。

软件可配置 RCC_COMMON_CR 寄存器中的位 LSI_EN 为 0，关闭 LSI。

5.2.5 LSE 时钟

LSE 时钟是由外部输入的由晶振或陶瓷谐振器产生的 32.768 kHz 时钟，可作为 RTC 和 WDG 的工作时钟源，具有功耗低且精度高的优点。

软件启用 LSE 时钟流程如下：

（1）软件对 RCC_LOCK 写 0x900D 0000，解锁 RCC 寄存器。

（2）配置 RCC_COMMON_CR 寄存器中的位 LSE_ST，设置外部时钟启动时间。

（3）配置时钟输入 GPIO 引脚为模拟模式。

（4）配置 RCC_COMMON_CR 寄存器中的位 LSE_EN 为 1，使能 LSE 时钟。

（5）软件读取 RCC_COMMON_CR 寄存器中的位 LSE_RDY，等待 LSE_RDY 置高。

软件可配置 RCC_COMMON_CR 寄存器中的位 LSE_EN 为 0，关闭 LSE。

5.2.6 系统时钟(SYSCLK)选择

软件通过配置 SYSCK_SEL 来选择以 HSI 或者 PLL 时钟作为系统时钟(SYSCLK)，系统时钟最大频率为 96MHz，复位后默认选择 HSI 时钟。

软件在切换系统时钟时，必须保证将要切换的时钟已经稳定，例如切换到 PLL，则 PLL_RDY 必须已经为 1。

5.2.7 低速时钟(LSCLK)

LSCLK 低速时钟(32.768 kHz)可以选择来自 LSI 或者 LSE，在时钟切换前需要确保时钟已经稳定（也即 LSE_RDY 或 LSI_RDY 为 1）。

LSCLK 时钟选择由 RCC_COMMON_CR 寄存器中的位 LSCLK_SEL 控制，为 0 表示选择 LSI，为 1 表示选择 LSE。

5.2.8 时钟分频与外设时钟使能

芯片 AHB 和 APB 外设的工作时钟 HCLK/PCLK 由 SYSCLK 时钟分频产生，软件通过配置寄存器 RCC_COMMON_CR 中的位 HCKD[2:0]和 PCKD[2:0]来设置分频，其中 APB 外设的时钟 PCLK 由 HCLK

时钟分频产生。
芯片的 AHB 和 APB 外设工作时钟默认为关闭状态，软件通过配置寄存器 RCC_CLKEN_CR 中的使能位来开启对应外设的工作时钟。
注意，软件发起对未开启工作时钟的外设寄存器访问时将会产生总线错误，CPU 将会产生 Hard Fault 中断。

5.2.9 ADC 时钟

ADC 工作时钟由 PCLK 提供，软件通过配置 ADC 工作寄存器设置采样时钟频率，具体参考 ADC 描述章节。

5.2.10 监测时钟输出(MCO)

支持将芯片内工作时钟输出到 MCO 引脚上，输出时钟可选择如下来源：
- HSI 时钟 6 分频
- LSI 时钟
- LSE 时钟
- PLL 时钟 6 分频

软件通过配置 RCC_COMMON_CR 寄存器中的位 MCO_SEL 来选择输出时钟源。
通过 GPIO 输出 MCO 时钟时，软件必须配置对应的 GPIO 为复用功能模式并将复用功能选择为 MCO。详请参考 GPIO 章节描述以及数据手册中引脚定义及复用功能章节。

5.3 低功耗模式

芯片支持睡眠(SLEEP)，深度睡眠(DEEP_SLEEP)，停止(STOP)三种低功耗模式。
软件通过配置 RCC_LPM_CR、RCC_SLEEP_CR 和 RCC_STOP_CR 寄存器设置芯片的低功耗状态，软件执行 WFI/WFE 后芯片将进入低功耗模式。低功耗特性如下表 5-1 RCC 低功耗特性所示。

表 5-1 RCC 低功耗特性

低功耗模式	睡眠(SLEEP)	深度睡眠(DEEP SLEEP)	停止(STOP)
寄存器 RCC_LPM_CR	-	MODE=00	MODE=01
CM0 寄存器 SCB_SCR.SLEEPDEEP	0	1	1
进入指令	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE
唤醒源	任意中断可唤醒	任意中断可唤醒	可被 GPIO、LPTIM、WDG 中断唤醒
IO 输出状态	保持	保持	保持
SRAM、寄存器	保持	保持	保持

定义	只有CPU的工作时钟被关闭，所有外设（包括CPU内部外设，如NVIC、SysTick）正常工作。	1.CPU 工作时钟和 CPU 内部外设时钟被关闭。 2. 系统外设时钟可在 RCC_SLEEP_CR 寄存器中配置为开启或关闭状态。	1. CPU 工作时钟和 CPU 内部外设时钟被关闭。 2.除低速时钟 LSI 可配置外，其他所有系统外设时钟关闭。软件可通过配置寄存器 RCC_STOP_CR 选择低速时钟 LSI 在 STOP 模式下打开或者关闭。 3.高速时钟源 HSI 关闭，ADC、ACMP、PVD 等模块关闭。 4. Flash 进入深度睡眠模式（Deep Standby）。
唤醒后状态	MCU 恢复到正常运行模式，程序继续执行。	MCU 恢复到正常运行模式，程序继续执行	MCU 恢复到正常运行模式，程序继续执行
注意事项	-	若配置 UART/SPI/I2C 时钟在深度睡眠模式下被关闭，则进入该模式前软件需先关闭这些外设的使能。	若配置停止模式时 LSI 时钟关闭，则 LPTIM 计数值会被复位。 进入停止模式前，软件需先关闭 UART/SPI/I2C 的使能。

5.4 寄存器概述

有关寄存器说明中使用的缩写，请参见表 5-2 通用寄存器概述。
如无特殊说明，外设寄存器可支持字节（8 位），半字（16 位）或字（32 位）访问。
在配置 RCC 寄存器时，必须对 RCC_LOCK_CR 寄存器写 0x900D0000 解除锁定，才能配置其它 RCC 寄存器。

表 5-2 通用寄存器概述

偏移地址	寄存器名	寄存器描述	复位值
0x000	RCC_COMMON_CR	RCC通用配置寄存器	0x5C000000
0x004	RCC_CLKEN_CR	RCC时钟使能控制寄存器	0x00000000
0x010	RCC_LPM_CR	RCC低功耗模式配置寄存器	0x00000000
0x014	RCC_SLEEP_CR	RCC深度睡眠模式时钟控制寄存器	0x00000000
0x01C	RCC_STOP_CR	RCC停止模式配置寄存器	0x00000006
0x024	RCC_RESET_SR	RCC复位状态寄存器	0x00000001
0x02C	RCC_SYSTEM_CR	RCC系统配置寄存器	0x00000000
0x030	RCC_LOCK_CR	RCC锁定配置寄存器	0x00000001
0x060	RCC_PLL_CR	RCC锁相环控制器	0x00000060

5.4.1 RCC 通用配置寄存器(RCC_COMMON_CR)

偏移地址：0x000
复位值：0x5C000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LSE_ST[1:0]		Res.	Res.	LSI_ST[1:0]		HSI_ST[1:0]		LSE_RDY	Res.	LSI_RDY	Res.	LSE_EN	Res.	LSI_EN	Res.
rw	rw			rw	rw	rw	rw	ro		ro		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCO_SEL[2:0]			MCO_EN	Res.	Res.	LSCK_SEL	SYSCK_SEL	Res.	HCKD[2:0]			Res.	PCKD[2:0]		
rw	rw	rw	rw			rw	rw		rw	rw	rw		rw	rw	rw

Bit	Field	Description
30:31	LSE_ST	LSE 时钟源启动稳定时间设置，硬件在等待设定等待时钟周期后，视为时钟准备就绪。 00：等待 1024 个时钟周期 01：等待 4096 个时钟周期 10：等待 16384 个时钟周期 11：等待 0 个时钟周期
29:28	Res.	保留，必须保持复位值。
27:26	LSI_ST	LSI 时钟源启动稳定时间设置，硬件在等待设定等待时钟周期后，视为时钟准备就绪。 00：等待 4 个时钟周期 01：等待 16 个时钟周期 10：等待 64 个时钟周期 11：等待 256 个时钟周期
25:24	HSI_ST	HSI 时钟源启动稳定时间设置，硬件在等待设定等待时钟周期后，视为时钟准备就绪。 00：等待 4 个时钟周期 01：等待 16 个时钟周期 10：等待 64 个时钟周期 11：等待 256 个时钟周期
23	LSE_RDY	LSE 时钟准备就绪。 0：LSE 时钟未准备就绪 1：LSE 时钟准备就绪
22	Res.	保留，必须保持复位值。
21	LSI_RDY	LSI 时钟准备就绪。 0：LSI 时钟未准备就绪 1：LSI 时钟准备就绪
20	Res.	保留，必须保持复位值。

19	LSE_EN	LSE 时钟使能。 0：关闭 LSE 时钟 1：使能 LSE 时钟
18	Res.	保留，必须保持复位值。
17	LSI_EN	LSI 时钟使能。 0：关闭 LSI 时钟 1：使能 LSI 时钟
16	Res.	保留，必须保持复位值。
15:13	MCO_SEL	时钟输出(MCO)源选择。 000：HSI 时钟 6 分频 001：保留 010：LSI 时钟 011：LSE 时钟 100：PLL 时钟 6 分频 101：PLL 时钟 12 分频 其它：HSI 时钟 6 分频
12	MCO_EN	时钟输出(MCO)使能。 0：关闭时钟输出 MCO 1：使能时钟输出 MCO
11	Res.	保留，必须保持复位值。
10	Res.	保留，必须保持复位值。
9	LSCK_SEL	RTC 低速时钟(LSCLK)选择。 0：选择 LSI 时钟 1：选择 LSE 时钟
8	SYSCK_SEL	系统时钟(SYSCLK)选择。 0：选择 HSI 时钟 1：选择 PLL 时钟
7	Res.	保留，必须保持复位值。
6:4	HCKD	AHB CLK 分频系数。 000：1 分频，即不分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110：64 分频 111：128 分频
3	Res.	保留，必须保持复位值。
2:0	PCKD	APB CLK 分频系数。 000：1 分频，即不分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 其它：16 分频

5.4.2 RCC 时钟使能控制寄存器(RCC_CLKEN_CR)

偏移地址：0x004
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	DSP	GPIO	Res.	CRC	DMA	TIM5	TIM4	ACMP	CAN	POSIF
						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
I2C	DAC	ADC	PVD	EPWM	CCT1	CCT0	TIM3	TIM2	TIM1	TIM0	SPI1	SPI0	UART2	UART1	UART0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:26	Res.	保留，必须保持复位值。
25	DSP	DSP 时钟使能。 0：关闭时钟 1：使能时钟
24	GPIO	GPIO 时钟使能。 0：关闭时钟 1：使能时钟
23	Res.	保留，必须保持复位值。
22	CRC	CRC 时钟使能。 0：关闭时钟 1：使能时钟
21	DMA	DMA 时钟使能。 0：关闭时钟 1：使能时钟
20	TIM5	TIM5 时钟使能。 0：关闭时钟 1：使能时钟
19	TIM4	TIM4 时钟使能。 0：关闭时钟 1：使能时钟
18	ACMP	ACMP 时钟使能。 0：关闭时钟 1：使能时钟
17	CAN	CAN 时钟使能。 0：关闭时钟 1：使能时钟 注意：DPM32M087RBT7、DPM32M087CBT7、DPM32M087KBT7、型号器件的寄存器该位才有效，其余的器件型号保留，且保持为复位

		值。
16	POSIF	POSIF 时钟使能。 0：关闭时钟 1：使能时钟
15	I2C	I2C 时钟使能。 0：关闭时钟 1：使能时钟
14	DAC	DAC 时钟使能。 0：关闭时钟 1：使能时钟
13	ADC	ADC 时钟使能。 0：关闭时钟 1：使能时钟
12	PVD	PVD 时钟使能。 0：关闭时钟 1：使能时钟
11	EPWM	EPWM 时钟使能。 0：关闭时钟 1：使能时钟
10	CCT1	CCT1 时钟使能。 0：关闭时钟 1：使能时钟
9	CCT0	CCT0 时钟使能。 0：关闭时钟 1：使能时钟
8	TIM3	TIM3 时钟使能。 0：关闭时钟 1：使能时钟
7	TIM2	TIM2 时钟使能。 0：关闭时钟 1：使能时钟
6	TIM1	TIM1 时钟使能。 0：关闭时钟 1：使能时钟
5	TIM0	TIM0 时钟使能。 0：关闭时钟 1：使能时钟
4	SPI1	SPI0 时钟使能。 0：关闭时钟 1：使能时钟 注意：DPM32M087KBT7、DPM32M080KBT7、DPM32M080GBT7、DPM32M080HBQ7 型号的器件该位保留，且保持为复位值。
3	SPI0	SPI0 时钟使能。 0：关闭时钟 1：使能时钟

2	UART2	UART2 时钟使能。 0：关闭时钟 1：使能时钟 注意：DPM32M087KBT7、DPM32M080KBT7、DPM32M080GBT7、DPM32M080HBQ7 型号的器件该位保留，且保持为复位值。
1	UART1	UART1 时钟使能。 0：关闭时钟 1：使能时钟
0	UART0	UART0 时钟使能。 0：关闭时钟 1：使能时钟

5.4.3 RCC 低功耗模式寄存器(RCC_LPM_CR)

偏移地址：0x010
复位值：0x00000000

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

Bit	Field	Description
31:2	Res.	保留，必须保持复位值。
1:0	MODE	低功耗模式配置。 00：深度睡眠模式 01：停止模式 10：保留 11：保留

5.4.4 深度睡眠模式配置寄存器(RCC_SLEEP_CR)

偏移地址：0x014
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	DSP	GPIO	Res.	CRC	DMA	TIM5	TIM4	ACMP	CAN	POSIF
						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
I2C	DAC	ADC	PVD	EPWM	CCT1	CCT0	TIM3	TIM2	TIM1	TIM0	SPI1	SPI0	UART2	UART1	UART0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:26	Res.	保留，必须保持复位值。
25	DSP	在深度睡眠模式时，DSP 时钟使能。 0：关闭时钟 1：使能时钟
24	GPIO	在深度睡眠模式时，GPIO 时钟使能。 0：关闭时钟 1：使能时钟
23	Res.	保留，必须保持复位值。
22	CRC	在深度睡眠模式时，CRC 时钟使能。 0：关闭时钟 1：使能时钟
21	DMA	在深度睡眠模式时，DMA 时钟使能。 0：关闭时钟 1：使能时钟
20	TIM5	在深度睡眠模式时，TIM5 时钟使能。 0：关闭时钟 1：使能时钟
19	TIM4	在深度睡眠模式时，TIM4 时钟使能。 0：关闭时钟 1：使能时钟
18	ACMP	在深度睡眠模式时，ACMP 时钟使能。 0：关闭时钟 1：使能时钟
17	CAN	在深度睡眠模式时，CAN 时钟使能。 0：关闭时钟 1：使能时钟 注意：DPM32M087RBT7、DPM32M087CBT7、DPM32M087KBT7 型号器件的寄存器该位才有效，其余的器件型号保留，且保持为复位值。
16	POSIF	在深度睡眠模式时，POSIF 时钟使能。 0：关闭时钟 1：使能时钟
15	I2C	在深度睡眠模式时，I2C 时钟使能。 0：关闭时钟 1：使能时钟

14	DAC	在深度睡眠模式时，DAC 时钟使能。 0：关闭时钟 1：使能时钟
13	ADC	在深度睡眠模式时，ADC 时钟使能。 0：关闭时钟 1：使能时钟
12	PVD	在深度睡眠模式时，PVD 时钟使能。 0：关闭时钟 1：使能时钟
11	EPWM	在深度睡眠模式时，EPWM 时钟使能。 0：关闭时钟 1：使能时钟
10	CCT1	在深度睡眠模式时，CCT1 时钟使能。 0：关闭时钟 1：使能时钟
9	CCT0	在深度睡眠模式时，CCT0 时钟使能。 0：关闭时钟 1：使能时钟
8	TIM3	在深度睡眠模式时，TIM3 时钟使能。 0：关闭时钟 1：使能时钟
7	TIM2	在深度睡眠模式时，TIM2 时钟使能。 0：关闭时钟 1：使能时钟
6	TIM1	在深度睡眠模式时，TIM1 时钟使能。 0：关闭时钟 1：使能时钟
5	TIM0	在深度睡眠模式时，TIM0 时钟使能。 0：关闭时钟 1：使能时钟
4	SPI1	在深度睡眠模式时，SPI1 时钟使能 0：关闭时钟 1：使能时钟 注意：DPM32M087KBT7、DPM32M080KBT7、DPM32M080GBT7、DPM32M080HBQ7 型号的器件该位保留，且保持为复位值。
3	SPI0	在深度睡眠模式时，SPI0 时钟使能 0：关闭时钟 1：使能时钟
2	UART2	在深度睡眠模式时，UART2 时钟使能。 0：关闭时钟 1：使能时钟 注意：DPM32M087KBT7、DPM32M080KBT7、DPM32M080GBT7、DPM32M080HBQ7 型号的器件该位保留，且保持为复位值。
1	UART1	在深度睡眠模式时，UART1 时钟使能。 0：关闭时钟

		1：使能时钟
0	UART0	在深度睡眠模式时，UART0 时钟使能。 0：关闭时钟 1：使能时钟

5.4.5 停止模式配置寄存器(RCC_STOP_CR)

偏移地址：0x01C
复位值：0x00000006

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

Bit	Field	Description
31:1	Res.	保留，必须保持复位值。
0	LSCLK_EN	在停止模式下，系统低速时钟 LSCLK 是否使能。 0：关闭系统 LSCLK 时钟 1：使能系统 LSCLK 时钟

5.4.6 复位源标记寄存器(RCC_RESET_SR)

偏移地址：0x024
复位值：0x00000001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR
															w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSTB_DET	PVD	WDG	DSP	LOCKUP	SW	BUTTON	POR
								ro	ro	ro	ro	ro	ro	ro	ro

Bit	Field	Description
31:17	Res.	保留，必须保持复位值。
16	CLR	系统复位状态清除，软件写 1 清除，硬件自动置 0。 0：未清除复位状态标志，或复位状态标志清除完成 1：发起清除复位状态标志，或复位状态标志正在清除

15:8	Res.	保留，必须保持复位值。
7	DSTB_DET	MCU 受到外部强干扰引起 Flash 访问异常，触发硬件自动复位。硬件置 1，软件写 CLR 位清除。 0：未发生复位，或者复位状态已经清除 1：发生干扰检测异常复位（Disturb detection）
6	PVD	PVD 监测电压异常复位标记。硬件置 1，软件写 CLR 位清除。 0：未发生电压异常 1：发生电压异常复位标记
5	WDG	WDG 看门狗监事件复位标记。硬件置 1，软件写 CLR 位清除。 0：未发生 WDG 复位事件 1：发生 WDG 复位事件标记
4	DSP	DSP 看门狗监事件复位标记。硬件置 1，软件写 CLR 位清除。 0：未发生 DSP 复位事件 1：发生 DSP 复位事件标记
3	LOCKUP	CPU lockup 复位事件标记。硬件置 1，软件写 CLR 位清除。 0：未发生 lockup 复位事件 1：发生 lockup 复位事件标记
2	SW	CPU 软件复位事件标记。硬件置 1，软件写 CLR 位清除。 0：未发生软件复位事件 1：发生软件复位事件标记
1	BUTTON	按键输入复位标记。硬件置 1，软件写 CLR 位清除。 0：未发生按键输入复位 1：发生按键输入复位标记
0	POR	上电 POR 和掉电 PDR 复位标记。硬件置 1，软件写 CLR 位清除。 0：未有上电 POR 和掉电 PDR 复位 1：上电 POR 和掉电 PDR 复位标记

5.4.7 RCC 系统控制寄存器(RCC_SYSTEM_CR)

偏移地址：0x02C
复位值：0x00000000

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

Bit	Field	Description
31:3	Res.	保留，必须保持复位值。
2	LKUPRST_EN	在 CPU lockup 时，是否触发系统复位。

		0：关闭 CPU lockup 触发系统复位 1：使能 CPU lockup 触发系统复位
1:0	REMAP	系统存储映射配置，将不同的存储区域映射到启动空间。 00：将 Flash 映射到启动空间 01：将 SRAM 映射到启动空间 10：将 BOOT ROM 映射到启动空间 11：保留

5.4.8 RCC 锁定寄存器(RCC_LOCK_CR)

偏移地址：0x030
复位值：0x000000001

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

Bit	Field	Description
31:16	KEY	当写入 KEY==0x900d 时，对第 0 位 LOCK_EN 的配置才会生效。
15:1	Res.	保留，必须保持复位值。
0	LOCK_EN	LOCK_EN 为 1 时，锁定所有 RCC 寄存器，无法写入。当 KEY 正确时，对于寄存器的配置才会生效。 0：关闭 RCC 寄存器写锁定 1：锁定所有 RCC 寄存器，无法写入只可读

5.4.9 RCC PLL 寄存器(RCC_PLL_CR)

偏移地址：0x060
复位值：0x000000060

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

Bit	Field	Description

31:10	Res.	保留，必须保持复位值。
9	RDY	PLL 准备就绪。 0：PLL 未准备就绪 1：PLL 准备就绪
8	EN	PLL 使能控制。 0：关闭 PLL 1：使能 PLL
7:4	MULTPS	PLL 倍频系数。 PLL 倍频系数为 MULTPS +1，MULTPS 取值为 0~15，倍频范围为 1~16
3:0	Res.	保留，必须保持复位值。

6 嵌套向量中断控制器（NVIC）

6.1 简介

嵌套的向量中断控制器（NVIC）主要特性：

- 32 个可屏蔽中断通道
- 4 个可编程优先级
- 低延迟异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

NVIC 和处理器内核接口紧密配合，可以实现低延迟的中断处理和晚到中断的高效处理。

6.2 主要特性

Cortex®-M0 处理器与内嵌中断向量控制（NVIC）对所有的异常进行优先级区分处理。当异常发生时，系统会将当前处理的工作压栈，执行完中断服务程序后出栈。更多详细信息，请参考“ARM®Cortex®-M0 技术参考手册”和“ARM®v6-M 架构参考手册”。

6.3 中断说明和映射表

IRQ ID	Priority	Allocation	Description	Address
-	-	Reserved	保留	0x0000_0000
-	-3	Reset	复位	0x0000_0004
-	-2	NMI	不可屏蔽中断	0x0000_0008
-	-1	Hard Fault	硬故障中断	0x0000_000C
-	可设置	SVCall	通过 SWI 指令的系统服务调用	0x0000_002C
-	-	Reserved	保留	0x0000_0030
-	-	Reserved	保留	0x0000_0034
-	可设置	PendSV	可挂起的系统服务	0x0000_0038
-	可设置	Systick	系统嘀嗒定时器	0x0000_003C
0	可设置	WDG	看门狗中断	0x0000_0040
1	可设置	LVD	电源电压检测中断	0x0000_0044
2	可设置	TIM0	TIM0 中断	0x0000_0048
3	可设置	TIM1	TIM1 中断	0x0000_004C
4	可设置	TIM2/TIM3	TIM2/TIM3 中断	0x0000_0050
5	可设置	TIM4/TIM5	TIM4/TIM5 中断	0x0000_0054
6	可设置	CCT0	捕获比较定时器 0 中断	0x0000_0058
7	可设置	CCT1	捕获比较定时器 1 中断	0x0000_005C
8	可设置	EPWM0	增强型 PWM0 定时器中断	0x0000_0060
9	-	Reserved	保留	0x0000_0064
10	可设置	POSIF0	霍尔和编码器接口控制器 0 中断	0x0000_0068

11	-	Reserved	保留	0x0000_006C
12	可设置	UART0	通用异步收发器 0 中断	0x0000_0070
13	可设置	UART1	通用异步收发器 1 中断	0x0000_0074
14	可设置	UART2	通用异步收发器 2 中断	0x0000_0078
15	可设置	SPI0	SPI0 中断	0x0000_007C
16	可设置	SPI1	SPI1 中断	0x0000_0080
17	可设置	I2C0	I2C0 中断	0x0000_0084
18	可设置	DMA	DMA 中断	0x0000_0088
19	可设置	CAN0	CAN0 中断	0x0000_008C
20	可设置	LPTIM	LPTIM 中断	0x0000_0090
21	可设置	ADC0	ADC0 中断	0x0000_0094
22	可设置	ADC1	ADC1 中断	0x0000_0098
23	可设置	GPIOA	GPIOA 中断	0x0000_009C
24	可设置	GPIOB	GPIOB 中断	0x0000_00A0
25	可设置	GPIOC/GPIOD	GPIOC/GPIOD 中断	0x0000_00A4
26	-	Reserved	保留	0x0000_00A8
27	可设置	DSP	DSP 中断	0x0000_00AC
28	可设置	ACMP0-3	ACMP0-3 中断	0x0000_00B0
29	-	Reserved	保留	0x0000_00B4
30	-	Reserved	保留	0x0000_00B8
31	-	Reserved	保留	0x0000_00BC

上表中 GPIO 和其他外设的中断使能、中断触发类型的配置以及中断状态的查询详见 GPIO 和各外设的寄存器章节。

7 外设硬件触发互连矩阵

7.1 简介

MCU 多个外设间有直接连接。支持硬件相互触发，硬件连接触发能消除软件延迟，且在配置完后，不需要软件介入，可以节省 CPU 资源。
硬件触发互连可以在正常模式和低功耗模式下工作，具体取决于互连的外设。

7.2 连接汇总

表 6-1 连接汇总

源外设事件	目的外设					
	ADC	DSP	CCT0	CCT1	POSIF	EPWM
IO 触发（ADC_TRIG）	Y					
EPWM ADC 比较事件 1	Y					
EPWM ADC 比较事件 2	Y					
DMA 通道 0 轮传输完成事件		Y				
DMA 通道 1 轮传输完成事件		Y				
DMA 通道 2 轮传输完成事件		Y				
DMA 通道 3 轮传输完成事件		Y				
DMA 通道 4 轮传输完成事件		Y				
TIM0 计数完成事件	Y					
TIM1 计数完成事件	Y					
TIM2 计数完成事件	Y					
TIM3 计数完成事件	Y					
TIM4 计数完成事件	Y					
TIM5 计数完成事件	Y					
ACMP 0 输出			Y	Y	Y	Y
ACMP 1 输出			Y	Y	Y	Y
ACMP 2 输出			Y	Y	Y	Y
ACMP 3 输出			Y	Y	Y	Y

7.3 互连详细信息

7.3.1 从 TIM0、TIM1、TIM2、TIM3、TIM4、TIM5、EPWM、GPIO 到 ADC

来自 TIM，EPWM 或者 GPIO 的外部信号均可触发 ADC 采样。具体选择哪个触发源可以通过 ADC 触发控制寄存器 ADC0_TRIG_CR 进行配置。
- TIMx 计数完成事件可以触发 ADC 采样。实现这一功能并不需要额外对 TIMx 进行配置。

- 在 EPWM 计数过程中，ADC 比较值事件可触发 ADC 采样，需要在 EPWM 模块内配置 ADC 比较值并使能。EPWM 支持配置两个 ADC 比较值，触发两次 ADC 采样。
- 可配置特定的 GPIO（ADC_TRIG）输入触发信号，使 ADC 可在这个信号触发下采样（可配置极性）。

7.3.2 从 ACMP 到 CCT、POSIF、EPWM

CCT 支持以 ACMP 比较器的输出作为捕获信号输入，可通过 CCT 模式选择配置寄存器(CCTx_CR)进行设置，详情可参考 CCT 章节详细说明。
POSIF 支持以 ACMP 比较器的输出作为捕获信号输入，可通过 POSIF 控制寄存器(POSIF_CR)进行设置，详情可参考 POSIF 章节详细说明。
EPWM 支持以 ACMP 比较器的输出作为急停信号输入，可通过控制寄存器（EPWM_CR）进行设置，详情可参考 EPWM 章节详细说明。
以上三个功能都无需在 ACMP 中进行额外配置，只需要在各个外设中配置选择 ACMP 输出作为信号源即可。

7.3.3 从 DMA 到 DSP

DMA 的某个通道完成一轮数据搬运事件可以触发 DSP 启动。
实现此功能无需在 DMA 中进行额外配置，只需要在 DSP 中使能 DMA 触发并选择 DMA 通道。

8 直接存储器访问控制器（DMA）

8.1 简介

直接存储器访问控制器（DMA）可以在无须 CPU 介入的情况下，主动通过系统总线实现任意存储器/外设之间的数据传输。共支持 5 个传输通道，可灵活配置每个通道的源地址和目的地址。

8.2 主要特性

- 5 个独立传输通道
- 固定优先级，从通道 0 到通道 4，优先级依次下降
- 每个传输通道均可单独配置如下参数：
  - 3 种数据传输位宽： 8-bit、16-bit、32-bit，源和目的需保持一致
  - 支持软件触发和硬件触发，硬件触发源可以从多个外设中进行选择
  - 源地址和目标地址可灵活配置为任意外设或者存储器
  - 源地址和目的地址变化模式可配（自增，循环）
  - 传输数据块大小可配置: 1~65536
  - 传输轮数可配置: 1~4096 功能说明

8.2.1 DMA 框图

DMA 拥有 5 个独立的通道，每个通道均可单独配置 DMA 请求源、是否循环搬运、传输轮数、每轮传输地址变化、传输数据块大小、数据传输位宽、源地址和目的地址等参数。

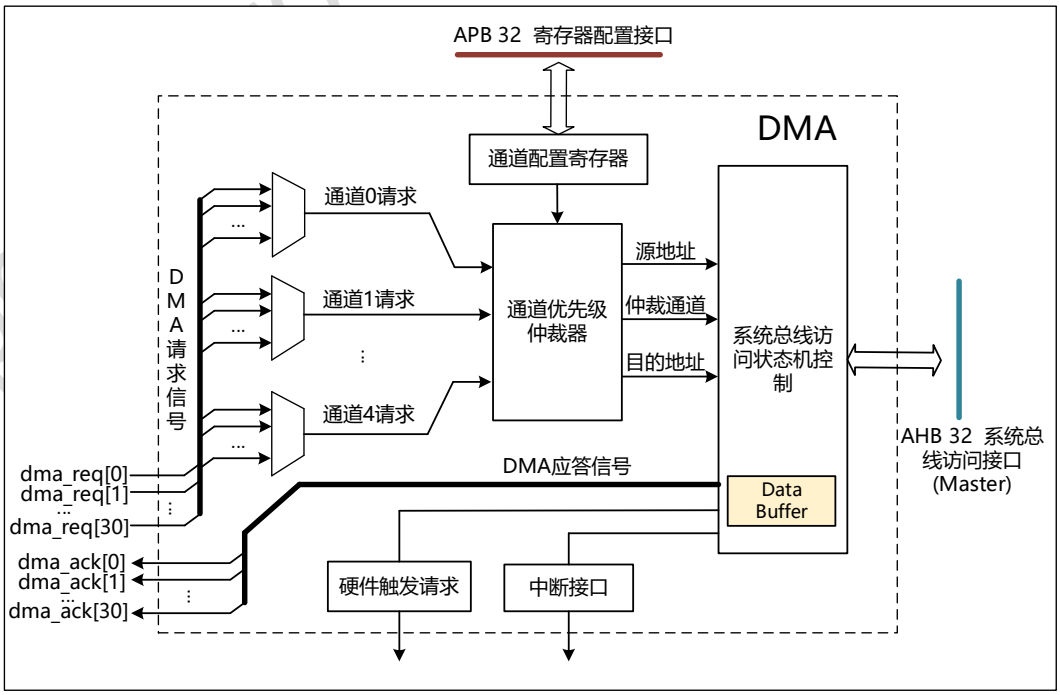

图 8-1 DMA 框图

DMA 请求源可以来自于 DMA 本身软件触发，也可以来自外设。不同通道的 DMA 请求信号首先将在通道优先级仲裁器中进行仲裁（从通道 0 到通道 4 优先级依次下降），然后根据仲裁结果，依次响应各通道的请求。

基本的数据搬运流程是先通过系统总线从源地址读取数据，然后存储到内部的数据缓存模块(Data Buffer)中，最后再通过系统总线将数据写入到目的地址。

每完成一次 DMA 请求，仲裁器会重新对其他的 DMA 请求进行仲裁，直到所有通道的 DMA 请求被处理完毕。

DMA 与系统其他主设备(CPU)共用 AHB 系统总线来进行数据传输，系统总线矩阵会对不同主设备的访问执行循环调度。当 CPU 和 DMA 同时发出请求时，会先执行 CPU 访问，再执行 DMA 访问。

8.2.2 DMA 请求映射

DMA 请求源共 31 种，包括 DMA 自身软件触发和由 30 个外设发出的 DMA 请求，如表 8-1 所示。DMA 的每个通道都可以通过 DMA_CHx_CFG0.REQ_SEL 配置该通道的 DMA 请求源。

表 8-2 DMA 请求映射

外设	DMA 请求事件	DMA 请求源 ID
DMA	SW_TRIG (软件触发)	0
UART0	UART0 的发送缓冲为空	1
	UART0 的接收缓冲非空	2
UART1	UART1 的发送缓冲为空	3
	UART1 的接收缓冲非空	4
UART2	UART2 的发送缓冲为空	5
	UART2 的接收缓冲非空	6
SPI0	SPI0 的发送缓冲为空	7
	SPI0 的接收缓冲非空	8
SPI1	SPI1 的发送缓冲为空	9
	SPI1 的接收缓冲非空	10
I2C	I2C0 的发送缓冲为空	11
	I2C0 的接收缓冲非空	12
Motor Turbo DSP	DSP 指令执行完成	13
ADC0	ADC0 软件触发单通道数据采样结束	14
	ADC0 序列扫描数据采样结束	15
	ADC0 硬件触发 0 单通道数据采样结束	16
	ADC0 硬件触发 1 单通道数据采样结束	17
ADC1	ADC1 软件触发单通道数据采样结束	18
	ADC1 序列扫描数据采样结束	19
	ADC1 硬件触发 0 单通道数据采样结束	20

	ADC1 硬件触发 1 单通道数据采样结束	21
TIM0	TIM0 计数事件	22
TIM1	TIM1 计数事件	23
TIM2	TIM2 计数事件	24
TIM3	TIM3 计数事件	25
TIM4	TIM4 计数事件	26
TIM5	TIM5 计数事件	27
CCT0	CCT0 计数/比较/捕获事件	28
CCT1	CCT1 计数/比较/捕获事件	29
EPWM	EPWM 计数事件	30

8.2.3 DMA 仲裁

DMA 5 个通道为固定优先级：通道 0 > 通道 1 > 通道 2 > 通道 3 > 通道 4。
当有多个请求在 DMA 空闲状态同时发生时，会根据优先级对不同通道的请求进行仲裁，当响应完本次仲裁出的通道请求后，DMA 恢复到空闲状态，此时可以重新进行仲裁。
仲裁只能在空闲状态发生，即使是高优先级的请求也无法打断正在进行数据传输的低优先级的请求。
在实际使用中，由于 DMA 的每个通道都支持所有的 DMA 请求源，因此可以根据实际场景的优先级将不同的外设请求映射到对应优先级的通道上。

8.2.4 DMA 通道配置

DMA 的全局状态寄存器 DMA_STATUS 包含了所有通道的传输状态和中断标志。
每个 DMA 通道拥有 DMA_CHx_CFG0，DMA_CHx_CFG1，DMA_CHx_SRC_ADDR，DMA_CHx_DST_ADDR 四个独立的寄存器（x 为通道编号，x=0,1,2,3,4）。
每个通道有单独的使能寄存器 DMA_CHx_CFG0.EN，只有开启通道使能后，对应的通道请求才可以被响应，另外此后不能再配置 DMA_CHx_CFG0.REQ_SEL、DMA_CHx_CFG0.CIRC、DMA_CHx_CFG1、DMAC_CHx_SRC_ADDR 和 DMA_CHx_DST_ADDR 寄存器，因此建议先将其他参数配置完成后再开启使能。
通过配置寄存器 DMA_CHx_CFG0.REQ_SEL 选择通道的 DMA 请求源，可以来自于软件触发或者外设，只能选择一个源头，默认为软件触发模式。
通过配置寄存器 DMA_CHx_CFG1.DATA_SIZE 设置传输数据的位宽，支持 8,16 或 32 位(byte，half-word or word)三种选择，从源地址读数据和向目的地址写数据的位宽必须一致。
DMA_CHx_SRC_ADDR 为通道传输初始的源地址，DMA_CHx_DST_ADDR 为通道传输初始的目的地址，二者可以被配置为系统中任意的有效地址。
对于每个 DMA 请求，DMA 需要搬运(DMA_CHx_CFG1.DATA_NUM+1)个位宽为 DATA_SIZE 的数据，此处将 DMA 响应一次请求的过程称为一轮。每个通道总共可响应(DMA_CHx_CFG1.ROUND_NUM+1)轮 DMA 请求。
如果 DMA 请求由外设发出，在完成一轮数据搬运后，DMA 会向对应外设发送应答信号。外设收到应答信号后，会释放 DMA 请求，随后 DMA 会撤销应答信号，如此就完成了一次外设 DMA 请求；

对于软件触发，首先需要先配置 DMA_CHx_CFG0.REQ_SEL 为 0，然后对寄存器 DMA_CHx_CFG0.S

W_TRIG 写 1。当 DMA 完成一轮数据搬运后，硬件会将该位清 0，软件可以通过读取 SW_TRIG 来判断本轮传输是否完成。

多轮传输地址变化情况如图 8-2 多轮传输过程中地址变化所示，其中 INIT_BASE 可以是 DMA_CHx_SRC_ADDR 或 DMA_CHx_DST_ADDR。

- DMA 在响应每轮请求期间，读地址和写地址都会根据位宽 DMA_CHx_CFG1.DATA_SIZE 增加：当数据位宽为 8 时，地址自增 1；当数据位宽为 16 时，地址自增 2；当数据位宽为 32 时，地址自增 4。DMA 会按照此规律进行(DMA_CHx_CFG1.DATA_NUM+1)个数据传输。
- 在收到新一轮 DMA 请求时，地址变化由寄存器 DMA_CHx_CFG1.SRC_INC_MODE/DST_INC_MODE 决定：配置为 1 时，新地址应在上一轮最后传输地址的基础上增加；配置为 0 时则应复位为初始地址。DMA 会按照此规律进行(DMA_CHx_CFG1.ROUND_NUM+1)轮数据传输。
- 完成所有轮的数据搬运后，若为单次模式(DMA_CHx_CFG0.CIRC=0)，则停止响应 DMA 请求，硬件会自动清除使能；若为循环模式(DMA_CHx_CFG0.CIRC=1)，则继续响应 DMA 请求，DMA 内部的源地址和目的地址会更新为初始配置的地址，然后重新从第一轮开始搬运数据。

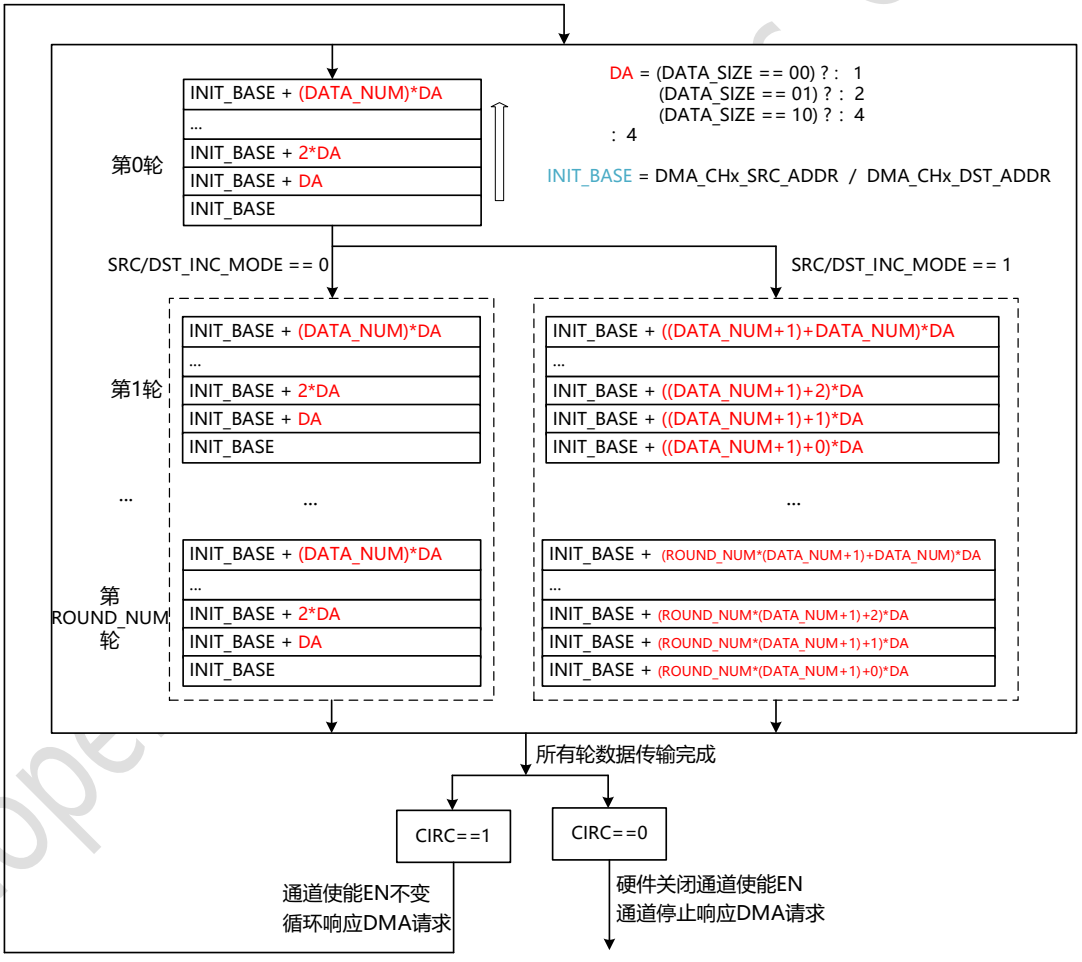

图 8-2 多轮传输过程中地址变化

8.2.5 DMA 通道配置示例

- 当处理 UART0 的接收数据时，可选定 DMA 通道 1，然后配置 DMA 请求源为 UART0 的接收缓冲

非空(DMA_CH1_CFG0.REQ_SEL=2)，源地址变化模式为复位成初始源地址(DMA_CH1_CFG1.SRC_INC_MODE=0)，目的地址变化模式为在上一轮最后的目的地址基础上增加(DMA_CH1_CFG1.DST_INC_MODE=1)，传输数据位宽为 32 比特(DMA_CH1_CFG1.DATA_SIZE=0b10)，源地址(DMA_CH1_SRC_ADDR)为 UART0 接收数据地址，目的地址(DMA_CH1_DST_ADDR)为内存，每次传输数据块大小为 1(DMA_CH1_CFG1.DATA_NUM=0)，轮数(DMA_CH1_CFG1.ROUND_NUM)为接收的数据量，循环模式关闭(DMA_CH1_CFG0.CIRC=0)。UART0 每次接收到数据后都会发出 DMA 请求，然后 DMA 负责将接收到的数据搬运到内存中。

- 循环模式一般适用于需要循环搬运数据的场景，例如在 ADC0 的扫描模式下处理数据。可选定 DMA 通道 2，然后配置 DMA 请求源为 ADC0 序列扫描数据采样结束事件(DMA_CH2_CFG0.REQ_SEL=15)，源地址变化模式为复位成初始源地址(DMA_CH2_CFG1.SRC_INC_MODE=0)，目的地址变化模式为在上一轮最后的目的地址基础上增加(DMA_CH2_CFG1.DST_INC_MODE=1)，传输数据位宽为 32 比特(DMA_CH2_CFG1.DATA_SIZE=0b10)，源地址(DMA_CH2_SRC_ADDR)为 ADC0 首个扫描转换结果寄存器地址，目的地址(DMA_CH2_DST_ADDR)为内存，每次传输数据块大小(DMA_CH2_CFG1.DATA_NUM)为扫描通道个数，轮数为 1(DMA_CH2_CFG1.ROUND_NUM=0)，循环模式开启(DMA_CH2_CFG0.CIRC=1)。ADC0 每完成一次扫描采样后，触发 DMA 请求，然后 DMA 会依次将采样数据搬运到内存中。

8.2.6 DMA 中断及错误管理

DMA 每个通道的传输状态由 DMA_SR.CHx_STATUS 标记：00 表示空闲状态；01 表示 DMA 在访问源地址时产生系统总线错误；10 表示 DMA 在访问目的地址时产生系统总线错误（产生总线访问错误的原因可能是访问地址不存在或者访问受限）；11 表示 Busy 状态，通道正在进行数据传输（当需要传输的数据较少时，软件可能无法即时读到 Busy 状态）。
对于传输错误标记，可以通过软件向 DMA_SR.CHx_STATUS 写 01/10 清除；也可以通过重新使能通道(DMA_CHx_CFG0.EN)进行清除。

中断类型有两种：
- 所有轮传输完成中断。当 DMA 完成(DMA_CHx_CFG1.ROUND_NUM+1)轮数据传输后，会触发完成中断 (对应的中断标志为 DMA_STATUS.CHx_FIS_IF，对应的中断使能为 DMA_CHx_CFG0.FIS_IE)。该位由硬件置 1，软件写 1 清 0，写 0 无效。
- 传输错误中断。当 DMA 接收到总线返回错误时，将清除通道使能(DMA_CHx_CFG0.EN==0)，停止响应 DMA 请求，并触发总线错误中断（对应的中断标志为 DMA_SR.CHx_ERR_IF，对应的中断使能为 DMA_CHx_CFG0.ERR_IE)。该位由硬件置 1，软件写 1 清 0，写 0 无效。软件可以读取 DMA_SR.CHx_STATUS 来进一步判断错误类型。

若在数据传输过程中，软件主动清除通道使能(DMA_CHx_CFG0.EN=0)，DMA 会终止正在进行的读/写操作并退出 Busy 状态。
重新使能通道也可以清除中断。

8.3 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 8-3 DMA 寄存器概述

偏移地址	寄存器名	寄存器描述	复位值
0x000	DMA_SR	DMA状态寄存器	0x00000000
0x004	DMA_CH0_CFG0	DMA通道0配置寄存器0	0x00000000
0x008	DMA_CH0_CFG1	DMA通道0配置寄存器1	0x00000000
0x00C	DMA_CH0_SRC_ADDR	DMA通道0源地址寄存器	0x00000000
0x010	DMA_CH0_DST_ADDR	DMA通道0目的地址寄存器	0x00000000
0x014	DMA_CH1_CFG0	DMA通道1配置寄存器0	0x00000000
0x018	DMA_CH1_CFG1	DMA通道1配置寄存器1	0x00000000
0x01C	DMA_CH1_SRC_ADDR	DMA通道1源地址寄存器	0x00000000
0x020	DMA_CH1_DST_ADDR	DMA通道1目的地址寄存器	0x00000000
0x024	DMA_CH2_CFG0	DMA通道2配置寄存器0	0x00000000
0x028	DMA_CH2_CFG1	DMA通道2配置寄存器1	0x00000000
0x02C	DMA_CH2_SRC_ADDR	DMA通道2源地址寄存器	0x00000000
0x030	DMA_CH2_DST_ADDR	DMA通道2目的地址寄存器	0x00000000
0x034	DMA_CH3_CFG0	DMA通道3配置寄存器0	0x00000000
0x038	DMA_CH3_CFG1	DMA通道3配置寄存器1	0x00000000
0x03C	DMA_CH3_SRC_ADDR	DMA通道3源地址寄存器	0x00000000
0x040	DMA_CH3_DST_ADDR	DMA通道3目的地址寄存器	0x00000000
0x044	DMA_CH4_CFG0	DMA通道4配置寄存器0	0x00000000
0x048	DMA_CH4_CFG1	DMA通道4配置寄存器1	0x00000000
0x04C	DMA_CH4_SRC_ADDR	DMA通道4源地址寄存器	0x00000000
0x050	DMA_CH4_DST_ADDR	DMA通道4目的地址寄存器	0x00000000

8.3.1 DMA 状态寄存器(DMA_SR)

偏移地址：0x000
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH4_STATUS[1:0]		CH4_ERR_IF	CH4_FIS_IF
												ro	ro	w1c	w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3_STATUS[1:0]		CH3_ERR_IF	CH3_FIS_IF	CH2_STATUS[1:0]		CH2_ERR_IF	CH2_FIS_IF	CH1_STATUS[1:0]		CH1_ERR_IF	CH1_FIS_IF	CH0_STATUS[1:0]		CH0_ERR_IF	CH0_FIS_IF
ro	ro	w1c	w1c	ro	ro	w1c	w1c	ro	ro	w1c	w1c	ro	ro	w1c	w1c

Bit	Field	Description
[19:18], [15:14], [11:10], [7:6],[3:2]	CHx_STATUS	通道 x 传输状态(x=0,1,2,3,4)。 00：空闲 01：访问源地址出错 10：访问目的地址出错 11：Busy，正在传输数据

17,13,9,5,1	CHx_ERR_IF	通道 x 传输错误中断标志(x=0,1,2,3,4)。由硬件置 1，软件写 1 清零，写 0 无效。 0：未发生总线错误 1：发生总线错误
16,12,8,4,0	CHx_FIS_IF	通道 x 所有轮传输完成中断标志(x=0,1,2,3,4)。由硬件置 1，软件写 1 清零，写 0 无效。 0：所有轮未传输完成 1：所有轮传输完成

8.3.2 DMA 通道 x 配置寄存器 0(DMA_CHx_CFG0)(x=0,1,2,3,4)

偏移地址：0x004 + x*16
复位值：0x0000 0000

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

Bit	Field	Description
31:13	Res.	保留，必须保持复位值。
12:8	REQ_SEL	DMA 请求源选择。 0：软件触发(SW_TRIG) 其余为外设请求； 具体请参考表 8-1，未定义请求 ID 为保留源，配置后无效；
7:5	Res.	保留，必须保持复位值。
4	ERR_IE	传输错误中断使能 0：不使能 1：使能
3	FIS_IE	所有轮传输完成中断使能 0：不使能 1：使能
2	SW_TRIG	软件写 1 触发 DMA 请求，传输完成后由硬件清 0 通道未使能时，对 SW_TRIG 写 1 无效，读回值为 0。
1	CIRC	循环模式使能 0：不使能 1：使能
0	EN	通道使能位。 0：通道禁止 1：通道使能 在非循环模式下，当通道所有轮传输完成时，硬件将自动清除使能。在

		循环模式下，硬件不会自动清除使能，需要由软件清除。

8.3.3 DMA 通道 x 配置寄存器 1(DMA_CHx_CFG1) (x=0,1,2,3,4)

偏移地址：0x008 + x*16
复位值：0x0000 0000

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

Bit	Field	Description
31:20	ROUND_NUM	通道响应 DMA 请求轮数，实际生效值为 ROUND_NUM+1
19	DST_INC_MODE	每次触发 DMA 请求，目的地址变化模式 0：复位为初始目的地址 1：在上一轮最后的目的地址基础上增加
18	SRC_INC_MODE	每次触发 DMA 请求，源地址变化模式 0：复位为初始源地址 1：在上一轮最后的源地址基础上增加
17:16	DATA_SIZE	传输数据位宽 00：8bit 01：16bit 10：32bit 11：保留
15:0	DATA_NUM	通道每轮传输数据数量，实际生效值为 DATA_NUM+1

8.3.4 DMA 通道 x 源地址寄存器(DMA_CHx_SRC_ADDR) (x=0,1,2,3,4)

偏移地址：0x00C + x*16
复位值：0x0000 0000

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

Bit	Field	Description
31:0	SRC_ADDR	通道初始源地址

8.3.5 DMA 通道 x 目的地址寄存器(DMA_CHx_DST_ADDR) (x=0,1,2,3,4)

偏移地址：0x010 + x*16
复位值：0x0000 0000

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

Bit	Field	Description
31:0	DST_ADDR	通道初始目的地址

9 通用输入输出接口（GPIO）

9.1 简介

最大支持 62 个通用输入输出引脚(GPIO)，包含 PA，PB，PC 和 PD 四个端口。
每个端口都有对应的一组配置、状态寄存器，用户可以根据应用需求独立灵活的将各个 IO 配置成通用输入/输出、外设复用功能(AF)或模拟功能使用。
支持上升沿/下降沿/上下沿和高/低电平触发的 GPIO 中断。
每个 GPIO 引脚都可以灵活的配置上拉、下拉或无上下拉电阻。数字输出支持推挽和开漏的选择，最大 4 档输出速率可选。

9.2 主要特性

- 各 IO 的功能可独立灵活配置
- 支持通用输入\输出模式，复用功能(AF)和模拟功能模式
- 从数据输出寄存器（GPIO_DATA_OUT）或内部外设输出数据到 IO
- 从 IO 接收数据到数据输入寄存器（GPIO_DATA_IN）或内部外设
- 支持上升沿/下降沿/上下沿和高/低电平触发的 GPIO 中断，可用作异步唤醒源
- 各 IO 支持上拉、下拉或无上下拉灵活配置
- 数字输出支持推挽和开漏选择
- 数字输出支持最大 4 档速率选择
- 支持软件置位/复位各 IO 输出值（通用输出模式时）
- 配置锁定机制，软件可灵活锁定/解锁 GPIO 配置

9.3 功能说明

IO 端口位的基本结构如图 9-1 IO 端口位基本结构所示。
IO 端口位各种不同的配置请见表 9-1 端口位配置表。

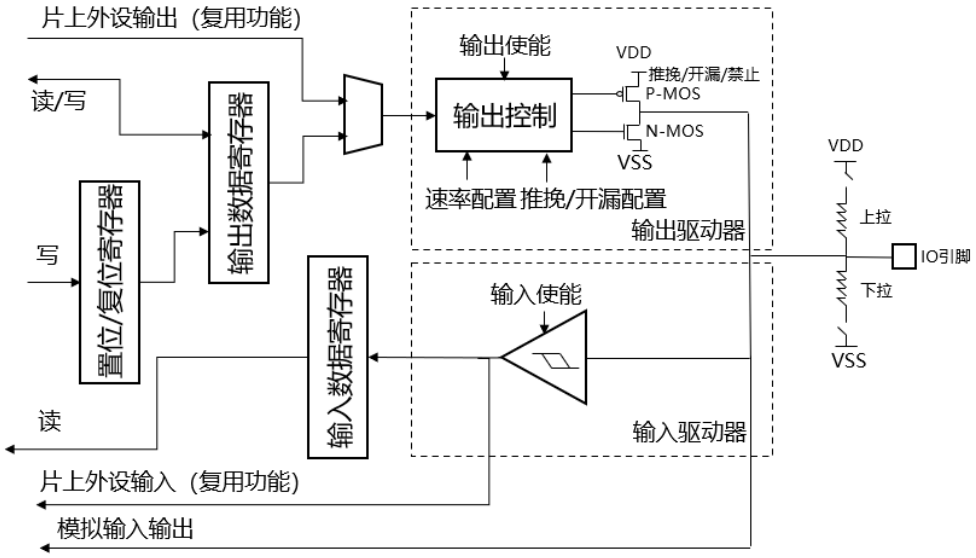

图 9-1 IO 端口位基本结构

表 9-1 端口位配置表

MODEi[1:0]	OTYPEi[1:0]	OSPEEDi[1:0]	PUPDi[1:0]	IO 配置描述
00	X	X	00	通用输入，无上下拉
	X	X	01	通用输入，上拉
	X	X	10	通用输入，下拉
	X	X	11	保留
01	0	OSPEEDi[1:0]	00	通用输出，推挽，无上下拉
	0	OSPEEDi[1:0]	01	通用输出，推挽，上拉
	0	OSPEEDi[1:0]	10	通用输出，推挽，下拉
	0	OSPEEDi[1:0]	11	保留
	1	OSPEEDi[1:0]	00	通用输出，开漏，无上下拉
	1	OSPEEDi[1:0]	01	通用输出，开漏，上拉
	1	OSPEEDi[1:0]	10	通用输出，开漏，下拉
	1	OSPEEDi[1:0]	11	保留
10	0	OSPEEDi[1:0]	00	复用功能（AF）输出，推挽，无上下拉
	0	OSPEEDi[1:0]	01	复用功能（AF）输出，推挽，上拉
	0	OSPEEDi[1:0]	10	复用功能（AF）输出，推挽，下拉
	0	OSPEEDi[1:0]	11	保留
	1	OSPEEDi[1:0]	00	复用功能（AF）输出，开漏，无上下拉
	1	OSPEEDi[1:0]	01	复用功能（AF）输出，开漏，上拉
	1	OSPEEDi[1:0]	10	复用功能（AF）输出，开漏，下拉
	1	OSPEEDi[1:0]	11	保留
11	X	X	X	模拟输出输出
	X	X	X	

	X	X	X	
	X	X	X	

注： 上表中各个配置字段的定义请见寄存器说明。上表中 X 代表任意值 i=0~15。

9.3.1 通用输入/输出模式

通过配置 GPIOx_MODE（x=A,B,C,D）寄存中的 MODEi[1:0] (i=0~15)字段为 00 或 01 可以将对应的 IO 配置为通用输入或输出模式。
当配置为通用输出模式后，写入到输出数据寄存器（GPIOx_DATA_OUT）的值将在对应的 I/O 引脚上输出。另外，可根据实际使用需求配置以下寄存器：
- GPIOx_OUT_TYPE 寄存器选择推挽模式或开漏模式输出
- GPIOx_OUT_SPEED 寄存器选择输出速率（共 4 档）

无论配置为通用输入或输出模式，硬件在每个系统 AHB 总线时钟周期捕获一次 IO 引脚的数据到输入数据寄存器（GPIOx_DATA_IN），用户可以通过软件读取。
无论配置为通用输入或输出模式，都可以根据需要配置 GPIOx_PUPD 寄存器来打开或者关闭 IO 内部上拉和下拉电阻。

系统复位后，除了 PA13，PA14，PD8 和 PD9 外，其他所有 IO 都默认配置为通用输入模式且无内部上下拉。
PA13 和 PA14 在系统复位后默认被用作系统调试接口，系统复位后默认配置如下：
PA13：复用功能模式 0（AF0：SWDIO），内部上拉打开
PA14：复用功能模式 0（AF0：SWCLK），内部下拉打开
PD8 在系统复位后默认配置为通用输入模式且打开了内部下拉
PD9 在系统复位后默认被用作按键复位（NRST）输入引脚，复位后默认配置为复用功能模式 0（AF0：NRST），内部上拉打开。

注意：
- 因 PD9 默认用作按键复位（NRST）功能，当将 PD9 配置成通用输入模式使用时，需要特别小心。需要确保使用过程种有任何系统复位发生前先将 PD9 的输入信号变为恒定的高电平，否则某些条件下可能导致系统一直被复位。
  因为任何系统复位发生后都会将 PD9 的模式复位到默认值（AF0： NRST）作为按键复位（低有效）的输入，如果系统复位后外部信号驱动 PD9 一直维持在低电平或持续低电平较长时间，系统会检测并识别成按键复位有效并产生系统复位。
- PD9 一但被配置成其他模式（模拟模式、通用输入/输出或非 ALT0 的其他复用模式）后，不建议再切回 ALT0（按键复位 NRST），否则可能导致不期待的系统复位。

9.3.2 复用功能（AF）模式

当配置 GPIOx_MODE（x=A,B,C,D）寄存中的 MODEi[1:0] (i=0~15)字段为 10 时，对应的 IO 被配置为复用功能（AF）模式，用于内部外设数据的输出和输出。即从内部外设输出数据到 IO 或从 IO 接收数据到内部外设。
每个 IO 可能被多个不同内部外设复用，同一个内部外设也可能复用多个不同的 IO，用户可以通过配置 GPIOx_AF_LOW 和 GPIOx_AF_HIGH 寄存器来选择各个 IO 被哪个外设使用。关于各端口引脚数字复

用功能的定义，请参考器件数据手册。
另外，可根据实际使用需求配置以下寄存器：
- GPIOx_OUT_TYPE 寄存器选择推挽模式或开漏模式输出
- GPIOx_OUT_SPEED 寄存器选择输出速率（共 4 档）
- GPIOx_PUPD 寄存器来打开或者关闭各个 IO 内部上拉和下拉电阻

无论内部外设通过 IO 输出或者输入，硬件在每个系统 AHB 总线时钟周期捕获一次 I/O 引脚的数据到输入数据寄存器（GPIOx_DATA_IN），用户可以通过软件读取。

9.3.3 模拟模式

当配置 GPIOx_MODE（x=A,B,C,D）寄存中 MODEi（i=0~15)字段为 11 时，对应的 IO 被配置为模拟模式，用于内部模拟外设的输入输出。
芯片的一部分 IO 会被内部模拟外设复用，用户可以通过配置 GPIOx_ANA_AF 寄存器来选择这些 IO 被哪个模拟外设复用。关于各端口引脚模拟复用功能的定义，请参考器件数据手册。

9.3.4 输出数据寄存器按位操作

可通过配置 GPIO 端口 bit 置位复位寄存器 GPIOx_BIT_SET_RST（x=A,B,C,D）对输出数据寄存器 GPIOx_DATA_OUT（x=A,B,C,D）中各数据位执行置位和复位操作。
GPIOx_DATA_OUT 寄存器中的每个数据位对应于 GPIOx_BIT_SET_RST 中的两个控制位：SETi(i=0~15)和 RSTi(i=0~15)。当 SETi 被写入 1 后，会置位对应的 DATA_OUTi 位。当 RSTi 被写入 1 后，会复位对应的 DATA_OUTi 位。
如果 在 GPIOx_BIT_SET_RST 寄存器中同时对某个数据 bit 执行置位和复位操作，则置位操作优先。
类似的，可通过配置 GPIO 端口 bit 复位寄存器 GPIOx_BIT_RST（x=A,B,C,D）对输出数据寄存器 GPIOx_DATA_OUT（x=A,B,C,D）中各数据位执行复位操作。
GPIOx_DATA_OUT 寄存器中的每个数据位对应于 GPIOx_BIT_RST 中的一个控制位：RSTi(i=0~15)。当 RSTi 被写入 1 后，会复位对应的 DATA_OUTi 位。
向 GPIOx_BIT_SET_RST 或 GPIOx_BIT_RST 寄存器中任何位写入 0 都不会对 GPIOx_DATA_OUT 寄存器的值产生任何影响。
使用 GPIOx_BIT_SET_RST 或 GPIOx_BIT_RST 寄存器更改 GPIOx_DATA_OUT 中各个位的值是一个“单次”操作，不会锁定 GPIOx_DATA_OUT 的值。软件随时都可以直接改写 GPIOx_DATA_OUT 的值。

9.3.5 配置寄存器锁定

为了防止软件运行异常、外部干扰或恶意攻击造成的 GPIO 的误配置和误操作，支持对 GPIOx_MODE、GPIOx_OUT_TYPE、GPIOx_OUT_SPEED、GPIOx_PUPD、
GPIOx_AF_LOW 、GPIOx_AF_HIGH 和 GPIOx_ANA_AF 寄存器的写操作权限的锁定功能。

锁定操作流程如下：
对 GPIO_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x1。注意：上述解锁操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。

解锁操作流程如下：
对 GPIO_LOCK 寄存器进行 32 bits 写访问，其中 KEY 字段写入 0x900D, LK 字段写入 0x0。注意：上述解锁操作必须是 32 bits 写访问且对 KEY 和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。

每次系统复位后，所有配置寄存器都处于解锁状态，无需解锁即可配置。
按上述解锁操作解锁成功后，如果没有再次通过上述锁定操作锁定，则解锁状态一直有效。
为了系统安全可靠，建议每次解锁并对 GPIO 配置寄存器完成所需配置后，通过上述锁定操作再次锁定。
如未解锁时，对 GPIOx_MODE、GPIOx_OUT_TYPE、GPIOx_OUT_SPEED、GPIOx_PUPD、GPIOx_AF_LOW 和 GPIOx_AF_HIGH 寄存器进行写访问，则写访问无效。
对 GPIO 所有寄存器的读访问，不受上述锁定影响。

9.3.6 外部中断/唤醒功能

当没有配置成模拟功能模式时，所有 IO 都支持外部输入信号触发中断。系统可用此中断作为异步唤醒源来从睡眠或停止模式将系统唤醒。
通过配置 GPIOx_INT_EN（x=A,B,C,D）寄存器可以独立控制每个 IO 的中断使能和关闭。
通过配置 GPIOx_INT_TYPE_LOW（x=A,B,C,D）和 GPIOx_INT_TYPE_HIGH（x=A,B,C,D）可以选择每个 IO 输入信号触发中断类型，支持上升沿、下降沿、高电平、低电平触发和上升下降沿都触发五种类型。

9.3.7 通用输入配置

将 I/O 端口配置为通用输入模式时：
- 输出驱动器被关闭
- 施密特触发器输入被打开
- 根据 GPIOx_PUPD 寄存器中的值决定是否打开上拉和下拉电阻
- 每个系统 AHB 总线时钟周期捕获一次 IO 引脚的数据到输入数据寄存器
- 对输入数据寄存器的读访问可获取 I/O 状态

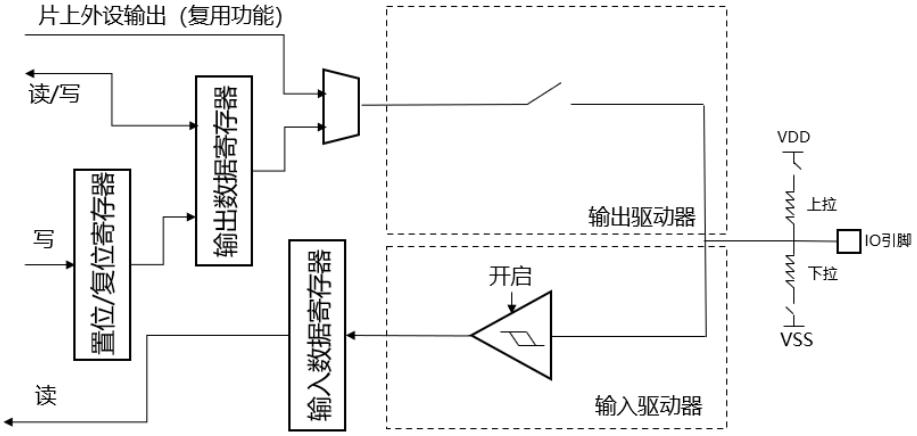

图 9-2 端口位通用输入配置

9.3.8 通用输出配置

将 I/O 端口配置为通用输入模式时：

- 输出驱动器被打开
- 开漏模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”会使端口保持高阻态（Hi-Z）（P-MOS 始终不激活）
- 推挽模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”可激 活 P-MOS
- 施密特触发器输入被打开
- 根据 GPIOx_PUPD 寄存器中的值决定是否打开上拉和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态
- 对输出数据寄存器的读访问可获取最后的写入值

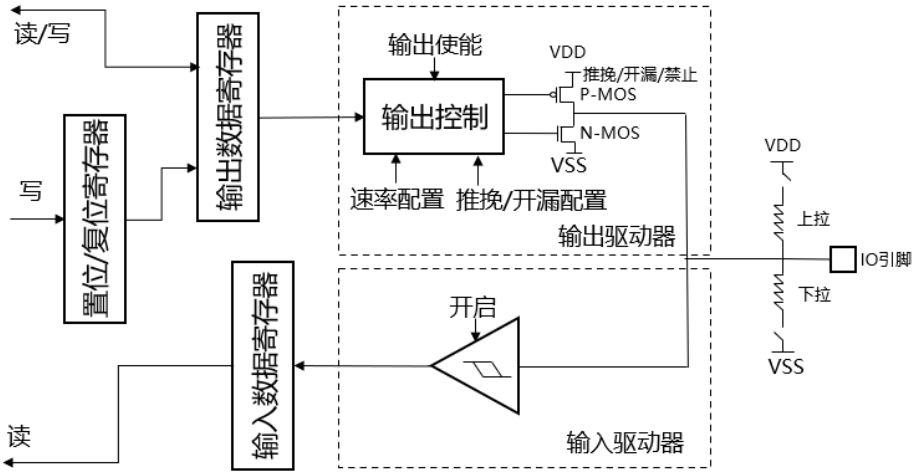

图 9-3 端口位通用输出配置

9.3.9 复用功能配置

将 I/O 端口配置为复用功能时：

- 可将输出驱动器配置为开漏或推挽模式
- 输出驱动器由来自内部外设的信号驱动
- 施密特触发器输入被打开
- 根据 GPIOx_PUPD 寄存器中的值决定是否打开上拉电阻和下拉电阻
- 输入数据寄存器每隔 1 个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态

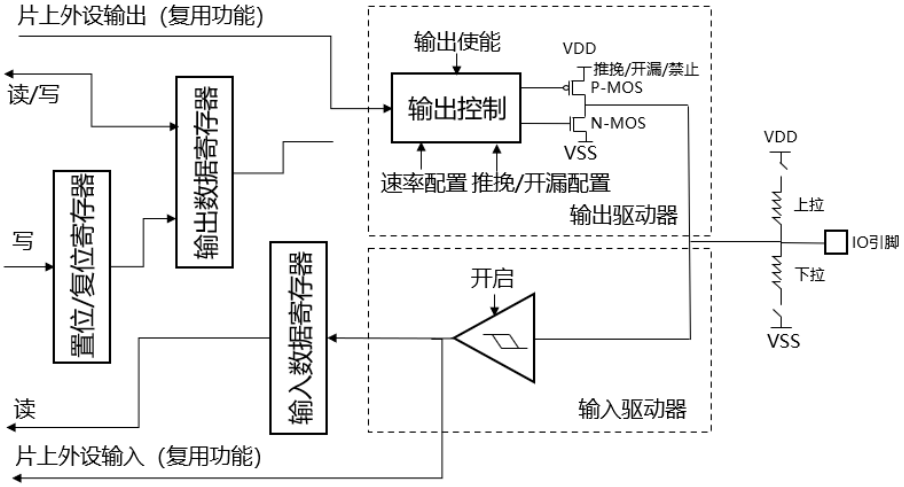

图 9-4 端口位复用功能配置

9.3.10 模拟模式配置

将 I/O 端口编程为模拟配置时：

- 输出驱动器被禁止
- 施密特触发器输入关闭
- 上拉和下拉电阻被硬件关闭
- 对输入数据寄存器的读访问不能获取 I/O 状态

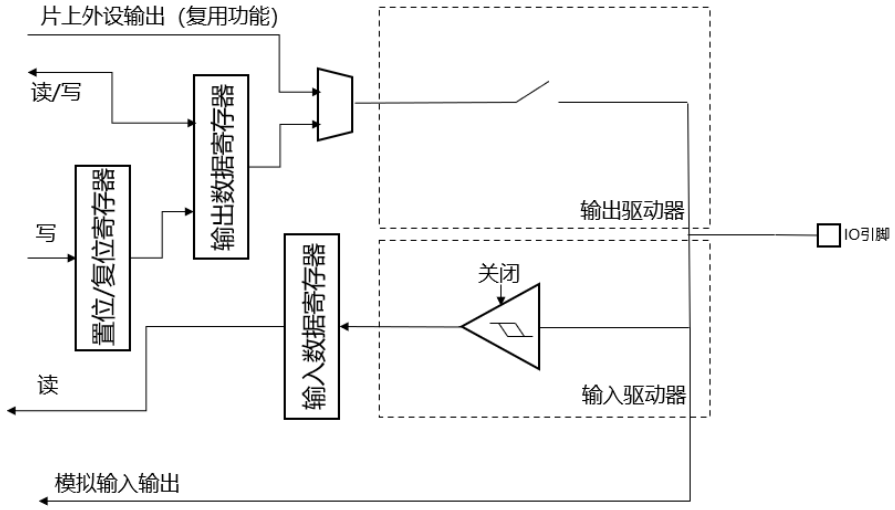

图 9-5 端口模拟功能配置

9.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 9-2 GPIO 寄存器概览

偏移地址	寄存器名	寄存器描述	复位值
0x000	GPIOx_MODE	GPIOx 端口模式配置寄存器	见章节 9.4.1
0x004	GPIOx_OUT_TYPE	GPIOx 端口输出类型配置寄存器	0x00000000
0x008	GPIOx_OUT_SPEED	GPIOx 端口输出速率配置寄存器	见章节 9.4.3
0x00c	GPIOx_PUPD	GPIOx 端口上下拉配置寄存器	见章节 9.4.4
0x010	GPIOx_DATA_IN	GPIOx 端口输入数据寄存器	0x00000000
0x014	GPIOx_DATA_OUT	GPIOx 端口输出数据寄存器	0x00000000
0x018	GPIOx_BIT_SET_RST	GPIOx 端口数据 bit 置位复位寄存器	0x00000000
0x01c	GPIOx_BIT_RST	GPIOx 端口数据 bit 复位寄存器	0x00000000
0x020	GPIOx_LOCK	GPIOx 端口配置锁定寄存器	0x00000000
0x024	GPIOx_AF_LOW	GPIOx 低位复用功能选择寄存器	0x00000000
0x028	GPIOx_AF_HIGH	GPIOx 高位复用功能选择寄存器	0x00000000
0x02c	GPIOx_ANA_AF	GPIOx 模拟复用功能选择寄存器	0x00000000
0x030	GPIOx_INT_EN	GPIOx 中断使能寄存器	0x00000000
0x034	GPIOx_INT_TYPE_LOW	GPIOx 低位中断类型配置寄存器	0x00000000
0x038	GPIOx_INT_TYPE_HIGH	GPIOx 高位中断类型配置寄存器	0x00000000
0x03c	GPIOx_INT_SR	GPIOx 中断状态寄存器	0x00000000

9.4.1 GPIOx 端口模式配置寄存器(GPIOx_MODE) (x=A,B,C,D)

偏移地址：0x000
复位值：
GPIOA: 0x28000000
GPIOB: 0x00000000
GPIOC: 0x00000000
GPIOD: 0x00080000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
2i+1:2i (i=0~15)	MODEi[1:0]	GPIO 端口 x 引脚 i 模式寄存器 (x=A ~D, i=0~15) 00: 通用输入模式 01: 通用输出模式 10: 复用功能模式 11: 模拟模式

注：
- PA13 和 PA14 在系统复位后默认被用作系统调试接口(SWCLK，SWDIO)，需使用复用功能 0(AF0)，所以 GPIOA_MODE 复位值为 0x2800_0000
- PD9 在系统复位后默认被用作按键复位（NRST）输入引脚，需使用复用功能 0(AF0)，所以 GPIOD_MODE 复位值为 0x0008_0000

9.4.2 GPIOx 端口输出类型配置寄存器(GPIOx_OUT_TYPE) (x=A,B,C,D)

偏移地址：0x004
复位值：0x00000000

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

Bit	Field	Description

31:16	Res.	保留。
i (i=0~15)	OTYPEi	GPIO 端口 x 引脚 i 输出类型寄存器 (x=A ~D, i=0~15) 0：推挽输出 1：开漏输出

9.4.3 GPIOx 端口输出速率配置寄存器 (GPIOx_OUT_SPEED) (x=A,B,C,D)

偏移地址：0x008
复位值：
GPIOA: 0x0C000000
GPIOB: 0x00000000
GPIOC: 0x00000000
GPIOD: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
2i+1:2i (i=0~15)	OSPEEDi[1:0]	GPIO 端口 x 引脚 i 输出速率寄存器 (x=A ~D, i=0~15) 00: 超低速 01: 低速 10: 高速 11: 超高速

注：
- 关于每种速率的频率范围以及电源和负载条件，请参考器件数据手册。
- PA13 在系统复位后默认被用作系统调试数据接口(SWDIO)，PA13 输出需为最大速率，所以 GPIOA_OUT_SPEED 复位值为 0x0C00_0000。

9.4.4 GPIOx 端口上下拉配置寄存器(GPIOx_PUPD) (x=A,B,C,D)

偏移地址：0x00C
复位值：
GPIOA:0x24000000
GPIOB:0x00000000
GPIOC:0x00000000
GPIOD:0x00060000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
2i+1:2i (i=0~15)	PUPDi[1:0]	GPIO 端口 x 引脚 i 上拉/下拉寄存器 (x=A ~D, i=0~15) 00: 无上拉或下拉 01: 上拉 10: 下拉 11: 保留

注：

- PA13 和 PA14 在系统复位后默认被用作系统调试接口(SWDIO,SWCLK)，PA13 需被上拉，PA14 需被下拉，所以 GPIOA_PUPD 复位值为 0x2400_0000
- PD8 在系统复位后默认被用作 BOOT 模式输入引脚，需被下拉；PD9 在系统复位后默认被用作按键复位（NRST）输入引脚，需被上拉，所以 GPIOD_MODE 复位值为 0x0006_0000

9.4.5 GPIOx 端口输入数据寄存器(GPIOx_DATA_IN) (x=A,B,C,D)

偏移地址：0x010
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN15	DIN14	DIN13	DIN12	DIN11	DIN10	DIN9	DIN8	DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro

Bit	Field	Description
31:16	Res.	保留。
i (i=0~15)	DINi	GPIO 端口 x 引脚 i 数据输入 (x=A ~D, i=0~15)

9.4.6 GPIOx 端口输出数据寄存器(GPIOx_DATA_OUT) (x=A,B,C,D)

偏移地址：0x014
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOUT15	DOUT14	DOUT13	DOUT12	DOUT11	DOUT10	DOUT9	DOUT8	DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:16	Res.	保留。
i (i=0~15)	DOUTi	GPIO 端口 x 引脚 i 输出数据 (x=A ~D, i=0~15)

注：通过配置 GPIOx_BIT_SET_RST 或 GPIOx_BIT_RST 寄存器可以实现对 GPIOx_DATA_OUT 寄存器按位置位或复位操作。

9.4.7 GPIOx 端口数据 bit 置位复位寄存器(GPIOx_BIT_SET_RST) (x=A,B,C,D)

偏移地址：0x018
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RST15	RST14	RST13	RST12	RST11	RST10	RST9	RST8	RST7	RST6	RST5	RST4	RST3	RST2	RST1	RST0
wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SET15	SET14	SET13	SET12	SET11	SET10	SET9	SET8	SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo

Bit	Field	Description
i+16 (i=0~15)	RSTi	GPIO 端口 x 引脚 i 输出数据复位 (x=A ~D, i=0~15) 0: 不改变 GPIOx_DATA_OUT 的值。 1: 将 GPIOx_DATA_OUT 中 DOUTi 清 0。完成后，RSTi 的值硬件自动清 0。
i (i=0~15)	SETi	GPIO 端口 x 引脚 i 输出数据置位 (x=A ~D, i=0~15) 0: 不改变 GPIOx_DATA_OUT 的值。 1: 将 GPIOx_DATA_OUT 中 DOUTi 置 1。完成后，SETi 的值硬件自动清 0。

注：当同时将 SETi 和 RSTi 置 1 时，SETi 的优先级更高。

9.4.8 GPIOx 端口数据 bit 复位寄存器(GPIOx_BIT_RST) (x=A,B,C,D)

偏移地址：0x01C
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RST15	RST14	RST13	RST12	RST11	RST10	RST9	RST8	RST7	RST6	RST5	RST4	RST3	RST2	RST1	RST0
wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo

Bit	Field	Description
31:16	Res.	保留。
i (i=0~15)	RSTi	GPIO 端口 x 引脚 i 输出数据复位 (x=A ~D, i=0~15) 0: 不改变 GPIOx_DATA_OUT 的值。 1: 将 GPIOx_DATA_OUT 中 DOUTi 清 0。完成后，RSTi 的值硬件自动清 0。

9.4.9 GPIOx 端口配置锁定寄存器(GPIOx_LOCK) (x=A,B,C,D)

偏移地址：0x020
复位值：0x00000000

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

Bit	Field	Description
31:16	KEY	GPIO 端口 x 配置寄存器锁定/解锁密钥。有效值 0x900D, 其他值无效。
15:1	Res.	保留。
0	LK	GPIO 端口 x 配置寄存器锁定。(x=A ~D) 0：不锁定任何寄存器。 1：锁定以下配置寄存器，锁定后，对其写访问无效，读访问正常。 GPIOx_MODE GPIOx_OUT_TYPE

		GPIOx_OUT_SPEED GPIOx_PUPD GPIOx_AF_LOW GPIOx_AF_HIGH GPIOx_ANA_AF

注：更改 LK 位的值时必须对 GPIOx_LOCK 寄存器进行 32 bits 写访问且对 KEY（0x900D）和 LK 字段在一次写操作中完成配置（16 bits，8 bits 写访问无效）。
解锁成功后一直处于解锁状态，直到软件更改 LK 位为 1 再次锁定。

9.4.10 GPIOx 低位复用功能选择寄存器(GPIOx_AF_LOW) (x=A,B,C,D)

偏移地址：0x024
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
4i+3:4i (i=0~7)	AFSELi[3:0]	GPIO 端口 x 引脚 i 数字复用功能选择 (x=A ~D, i=0~7) 0000: 选择复用功能 0（AF0） 0001: 选择复用功能 1（AF1） 0010: 选择复用功能 2（AF2） 0011: 选择复用功能 3（AF3） 0100: 选择复用功能 4（AF4） 0101: 选择复用功能 5（AF5） 0110: 选择复用功能 6（AF6） 0111: 选择复用功能 7（AF7） 1xxx: 保留。

注：关于各端口引脚数字复用功能的定义，请参考器件数据手册。

9.4.11 GPIOx 高位复用功能选择寄存器(GPIOx_AF_HIGH) (x=A,B,C,D)

偏移地址：0x028
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL15[3:0]				AFSEL14[3:0]				AFSEL13[3:0]				AFSEL12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL11[3:0]				AFSEL10[3:0]				AFSEL9[3:0]				AFSEL8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
4*(i-8)+3 : 4*(i-8) (i=8~15)	AFSELi[3:0]	GPIO 端口 x 引脚 i 数字复用功能选择 (x=A ~D, i=8~15) 0000: 选择复用功能 0（AF0） 0001: 选择复用功能 1（AF1） 0010: 选择复用功能 2（AF2） 0011: 选择复用功能 3（AF3） 0100: 选择复用功能 4（AF4） 0101: 选择复用功能 5（AF5） 0110: 选择复用功能 6（AF6） 0111: 选择复用功能 7（AF7） 1xxx: 保留。

注：关于各端口引脚数字复用功能的定义，请参考器件数据手册。

9.4.12 GPIOx 模拟复用功能选择寄存器(GPIOx_ANA_AF) (x=A,B,C,D)

偏移地址：0x02C
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ANA_AFSEL15[1:0]		ANA_AFSEL14[1:0]		ANA_AFSEL13[1:0]		ANA_AFSEL12[1:0]		ANA_AFSEL11[1:0]		ANA_AFSEL10[1:0]		ANA_AFSEL9[1:0]		ANA_AFSEL8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANA_AFSEL7[1:0]		ANA_AFSEL6[1:0]		ANA_AFSEL5[1:0]		ANA_AFSEL4[1:0]		ANA_AFSEL3[1:0]		ANA_AFSEL2[1:0]		ANA_AFSEL1[1:0]		ANA_AFSEL0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
2i+1:2i (i=0~15)	ANA_AFSELi[1:0]	GPIO 端口 x 引脚 i 模拟复用功能选择 (x=A ~D, i=0~15) 00: 选择模拟复用功能 0 01: 选择模拟复用功能 1 1x: 保留。

注：关于各端口引脚模拟复用功能的定义，请参考器件数据手册。

9.4.13 GPIOx 中断使能寄存器(GPIOx_INT_EN) (x=A,B,C,D)

偏移地址：0x030
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE15	IE14	IE13	IE12	IE11	IE10	IE9	IE8	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:16	Res.	保留。
i (i=0~15)	IEi	GPIO 端口 x 引脚 i 中断使能 (x=A ~D, i=0~15) 0: 中断关闭 1: 中断使能

9.4.14 GPIOx 低位中断类型配置寄存器(GPIOx_INT_TYPE_LOW) (x=A,B,C,D)

偏移地址：0x034
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	INT_TYPE7[2:0]			Res.	INT_TYPE6[2:0]			Res.	INT_TYPE5[2:0]			Res.	INT_TYPE4[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	INT_TYPE3[2:0]			Res.	INT_TYPE2[2:0]			Res.	INT_TYPE1[2:0]			Res.	INT_TYPE0[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

Bit	Field	Description
4i+3 (i=0~7)	Res.	保留。
4i+2 : 4i (i=0~7)	INT_TYPEi[2:0]	GPIO 端口 x 引脚 i 中断类型 (x=A ~D, i=0~7) 000: 上升沿触发中断 001: 下降沿触发中断 010: 高电平触发中断 011: 低电平触发中断 100: 上升和下降沿都触发中断 101~111: 保留

9.4.15 GPIOx 高位中断类型配置寄存器(GPIOx_INT_TYPE_HIGH) (x=A,B,C,D)

偏移地址：0x038
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	INT_TYPE15[2:0]			Res.	INT_TYPE14[2:0]			Res.	INT_TYPE13[2:0]			Res.	INT_TYPE12[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	INT_TYPE11[2:0]			Res.	INT_TYPE10[2:0]			Res.	INT_TYPE9[2:0]			Res.	INT_TYPE8[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

Bit	Field	Description
4*(i-8)+3 (i=8~15)	Res.	保留。
4*(i-8)+2 : 4*(i-8) (i=8~15)	INT_TYPEi[2:0]	GPIO 端口 x 引脚 i 中断类型 (x=A ~D, i=8~15) 000: 上升沿触发中断 001: 下降沿触发中断 010: 高电平触发中断 011: 低电平触发中断 100: 上升和下降沿都触发中断 101~111: 保留

9.4.16 GPIOx 中断状态寄存器(GPIOx_INT_SR) (x=A,B,C,D)

偏移地址：0x03C
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT15	INT14	INT13	INT12	INT11	INT10	INT9	INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c

Bit	Field	Description
31:16	Res.	保留。
i (i=0~15)	INTi	GPIO 端口 x 引脚 i 中断标志(x=A ~D, i=0~15) 0: 无中断 1: 中断发生 当中断类型配置为上升沿触发、下降沿触发或上升下降沿都触发时，向

		INTi 写 1 可以清除该中断标志位。 当中断类型配置为高电平或低电平触发时，向 INTi 写 1 无法清除该中断标志位，只有对应 IO 的输入电平无效时，该中断标志位自动清除。

10 电机算法加速 DSP（Motor Turbo DSP）

10.1 简介

针对电机算法设计了专用加速 DSP，其支持加法、乘累加、饱和、移位等单周期指令和除法、正余弦和反正切与取模等多周期指令。DSP 内嵌数据和指令存储器，可独立执行电机控制算法，以此减少 CPU 资源占用。

10.2 主要特性

- 包含 256 x 32 指令存储器（可存储 512 条指令，指令长度为 16 比特），16 个 16 位通用数据寄存器，32 个 16 位静态数据寄存器
- 支持加法、乘累加、饱和、移位、立即数加载等单周期指令
- 内嵌 Cordic 计算单元，支持正余弦和反正切与取模运算
- 内嵌除法单元
- ALU 支持并行计算
- 内嵌硬件断点，支持硬件调试
- 内嵌专用看门狗，硬件自动喂狗，可监测程序的执行
- 支持软硬件触发程序运行
- 支持灵活配置程序开始地址和程序结束地址

10.3 功能说明

10.3.1 DSP 框图

DSP 内部主要包括如下资源：

- ALU 计算单元，可执行加法、减法、乘法、除法、移位、正余弦、反正切与取模等运算。
- 控制与状态寄存器，负责控制 DSP 运行，监测运行状态，硬件断点调试等。
- 16 个通用数据寄存器 DSP_GPR0-DSP_GPR15（简记为 R0~R15）。每个寄存器的有效位宽为 16 比特，可存储 16 位有符号数，其中最高位为符号位，低 15 位为小数部分（即 Q15 格式）。
- 32 个静态数据寄存器 DSP_SDR0-DSP_SDR31（简记为 S0~S31）。其数据格式与通用数据寄存器相同。
- 256 个指令存储器 DSP_INS0-DSP_INS255. 每个存储器的有效位宽为 32 比特，可以存储 2 条指令，分别位于存储器的高 16 位低 16 位。

用户需要基于本 DSP 指令集对电机算法进行实现，对于常见算法，可参考例程。

DSP 的基本配置流程如下：

- 将算法中用到的数据和参数写入到通用数据寄存器(DSP_GPRx)和静态数据寄存器(DSP_SDRx)中. 通用数据寄存器可以在算术指令中被索引，然后直接参与 ALU 的运算，一般用于存放动态数据。静态数据寄存器不能被运算指令索引，一般用于存放电机参数等静态数据。通用数据寄存器和静态

数据寄存器可通过数据转移指令进行数据交互。

- 将设计好的指令写入到指令存储器，然后设置相应的程序开始地址(DSP_CTRL.PC_START)和程序结束地址(DSP_CTRL.PC_END)。
- 配置模块使能(DSP_CTRL.EN=1)。

在完成配置后，可通过两种方式触发执行程序：

- 软件触发。通过软件对软件触发寄存器 DSP_START.START 写 1。
- 硬件触发。支持由 DMA 一轮数据传输结束事件来触发运算，此时需要使能 DMA 触发(DSP_CTRL.DMA_TRIG_EN)和选择 DMA 通道(DSP_CTRL.DMA_TRIG_SEL)。

DSP 执行流程：

- 触发运算后，DSP 首先从 DSP_CTRL.PC_START 对应的指令存储器依次取指令，对各指令进行译码后执行计算或者跳转操作，直到指令计数器 PC 等于 DSP_CTRL.PC_END 时停止（PC_END 对应的最后一条指令也会被执行）。指令执行完成后，可以触发完成中断和 DMA 请求。
- 软件可以通过读取 DSP_STATUS.PC 查看正在执行哪条指令。
- 在 DSP 执行程序过程中，无法修改 DSP_CTRL.PC_START 和 DSP_CTRL.PC_END。

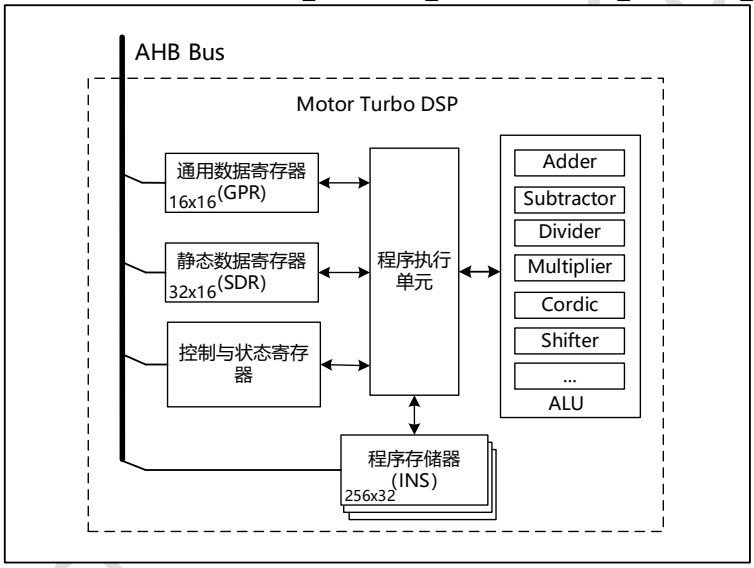

图 10-1 DSP 框图

10.3.2 指令集

指令集中定义的指令都为 16 位等长指令，其主要分为以下 3 类：

（1） 算术逻辑指令

16 位算术逻辑指令指令结构如图 10-2 算术逻辑指令结构所示。算术逻辑指令的操作数和结果只能来自或者存放在通用寄存器 R0~R15。

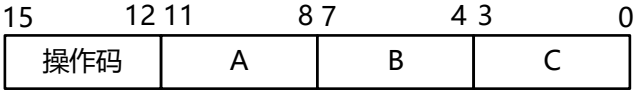

图 10-2 算术逻辑指令结构

其中 15-12 位为操作码，A，B，C 分别占用 4 位，可用于存放操作数或者计算结果的寄存器地址，地址范围为 0-15，与通用寄存器的 R0~R15 相对应；B，C 也可以是 4 位的立即数（Imm）。

各个指令的具体定义如表 10-1 所示。汇编语法中的各字符与指令结构中各部分依次对应。其中 Rs1，Rs2，Rd1，Rd2 为直接索引通用数据寄存器的地址，对指令进行译码时会从对应编号的通用寄存器中

取操作数，或者将结果写入到对应的通用数据寄存器。

表 10-2 算术逻辑指令定义

指令名称	操作码(二进制)	汇编语法	功能
加法指令	0001	ADD Rd1 Rs1 Rs2	Rd1 = Rs1 + Rs2
减法指令	0010	SUB Rd1 Rs1 Rs2	Rd1 = Rs1 – Rs2
乘法移位指令	0100	MULH Rd1 Rs1 Rs2	Rd1 = (Rs1 × Rs2) >>15 （Q15 有符号数乘 Q15 有符号数，结果硬件自动移位，仍为 Q15）
乘法指令	0101	MULL Rd1 Rs1 Rs2	Rd1 = (Rs1 × Rs2) （Q15 有符号数乘 Q0 有符号数，结果为 Q15）
乘累加指令	0011	MAC Rs1 Rs2 Rs3	R15 = (Rs2 × Rs3) >> 15 + Rs1 操作数均为 16 位有符号数。计算结果默认存放在通用数据寄存器 R15。 乘法结果硬件自动右移 15 位，与乘法移位指令相同，保证中间结果也是 Q15。
算术右移指令	0110	ASR Rd1 Rs1 Imm	Rd1 = Rs1 >>> Imm（有符号算术右移）
算术左移指令	0111	ASL Rd1 Rs1 Imm	Rd1 = Rs1 <<< Imm（有符号算术左移）
除法指令	1010	DIV Rd1 Rs1 Rs2	Rd1 = (Rs1 <<< 15) / Rs2 R15 = (Rs1 <<<15) % Rs2 除法计算的余数结果默认存放在通用数据寄存器 R15。 被除数 Rs1 会由硬件自动左移 15 位（算术移位），移位结果为 31 位有符号数，最低 15 位为 0，最高位为符号位。
正余弦指令	1000	SIN_COS Rd1 Rd2 Rs1	Rd1 = cos (Rs1) Rd2 = sin (Rs1) Rs1 为角度，用弧度制表示，取值范围为（-32768~32767），对应表示(–π~π)的角度。

			Rd1,Rd2 分别存放 cos 和 sin 计算结果，取值范围为（-32768~32767），对应表示(–1~1)的正余弦值。
反正切与取模指令	1001	ARCTAN_SQRT Rd1 Rs1 Rs2	Rd1 = arctan(Rs2/Rs1) R14 = sqrt(Rs1^2+Rs2^2) sqrt 取模结果默认存放在通用数据寄存器 R14。 Rs2,Rs1 可理解为角度对应的 cos 值和 sin 值，Rd1 为反正切求得的对应角度。取值范围为（-32768~32767），对应表示(–π~π)的角度。 sqrt($Rs1^2+Rs2^2$)为计算模值，需要注意如果结果超过取值范围(-32768~32767)，会被溢出保护。
饱和指令	1011	SAT Rd1 Rs1 Rs2	对 Rd1 操作数进行限制。Rd1 与 Rs1，Rs2 为有符号数比较。 if (Rd1<Rs1) Rd1=Rs1 else if (Rd1>Rs2) Rd1=Rs2 else Rd1= Rd1
低位立即数加载指令	1100	LDIL Rd1 Imm	加载 8 位立即数 Imm 到 Rd1 低 8 位。 Rd1[7:0]= Imm
高位立即数加载指令	1101	LDIH Rd1 Imm	加载 8 位立即数 Imm 到 Rd1 高 8 位。 Rd1[15:8]= Imm

指令计算周期与优先级说明：

1. DIV 计算周期为 10，在第 11 个周期将结果写入指定的通用数据寄存器，但如果有一个并行的运算在第 11 个周期需要使用结果寄存器的数据，此时实际使用的是寄存器中旧的数据，即除法的结果直到第 12 个周期才能被使用。
   启动当前除法运算后，在之后的第 0~11 个周期内无法再启动新的除法。
2. SIN_COS 与 ARCTAN_SQRT 指令均调用 Cordic 计算单元实现，当 cordic 单元处于工作状态时，触发新的需要调用 cordic 单元的指令将不会生效。例如当一条 SIN_COS 指令正在执行期间，触发新的 SIN_COS 指令和 ARCTAN_SQRT 指令都将无效。
   SIN_COS 计算周期为 8，在第 9 个周期将结果写入指定的通用数据寄存器，在第 10 个周期可以使用计算结果，在第 9 个周期可以启动新的 SIN_COS 运算。
   ARCTAN_SQRT 计算周期为 9，在第 10 个周期将结果写入到指定的通用数据寄存器，在第 11 个周期可以使用计算结果，在第 10 个周期可以启动新的 ARCTAN_SQRT 运算。
3. DIV、SIN_COS 和 ARCTAN_SQRT 为多周期指令，其余指令均为单周期指令。单周期指令与多周期指令可以并行执行，调用不同计算单元的多周期指令也可以并行执行。

例如，指令 x 为多周期指令，指令 x 执行后，需要多个周期才能得到计算结果，期间可以执行多条单周期指令，也可以执行非同一计算单元的多周期指令。

当在同一周期有两个以上计算结果需要写入到相同的通用数据寄存器时，其优先级如下：SIN_COS/ARCTAN_SQRT > DIV > 单周期运算指令 > 立即数加载指令。只有优先级高的结果将被写入通用寄存器，优先级低的结果将被放弃。

4. 对于 ARCTAN_SQRT 指令，如果反正切结果寄存器被设置为 R14（与默认的取模结果寄存器相同），那么当计算完成后，只会将反正切结果写入到 R14 中。同理，对于 DIV 指令，只会将商的结果写入到 R15；对于 SIN_COS 指令，只会将余弦结果写入到通用寄存器（此时正弦和余弦的结果寄存器相同）。

（2） 数据转移指令

16 位数据转移指令指令结构如图 10-3 数据转移指令结构所示。数据转移指令的作用是在通用数据寄存器和静态数据寄存器之间转移数据。

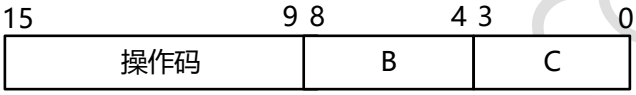

图 10-3 数据转移指令结构

其中 15-9 位为操作码， B，C 分别占用 5 位和 4 位，B 为索引静态数据寄存器的地址，C 为索引通用数据寄存器的地址。关于各个指令的具体定义如表 10-3 所示。

表 10-4 数据转移指令定义

指令名称	操作码（二进制）	汇编语法	功能
加载指令	0000010	LD Rs1 Rs2	将数据从静态数据寄存器 Rs1 加载到通用数据寄存器 Rs2。
存储指令	0000011	STR Rs1 Rs2	将数据从通用数据寄存器 Rs2 存储到静态数据寄存器 Rs1。

（3） 程序跳转指令

包含 JLEI 和 JUMPI 两条跳转指令。JLEI 为条件跳转，指令结构如图 10-2 算术逻辑指令结构所示。JUMPI 为立即数跳转，指令结构如图 10-4 程序跳转指令结构所示，其中 C 为 9 位立即数。用户可以利用程序跳转指令实现复杂的分支处理及跳转逻辑。

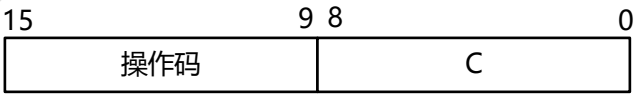

图 10-4 程序跳转指令结构

表 10-5 程序跳转指令定义

指令名称	操作码（二进制）	汇编语法	功能

条件跳转指令	1111	JLEI Rs1 Rs2 Imm	if (Rs1 <= Rs2)
			PC = PC + Imm;
			else
			PC = PC + 1;
			其中 Rs1 和 Rs2 在比较大小时被视为是有符号数。

立即数跳转指令	0000 001	JUMPI Imm	PC = Imm

10.3.3 硬件自清除看门狗

为了监测 DSP 的运行状态，防止因受到外部干扰导致程序跑飞，进而导致电机失控，DSP 内嵌了看门狗。开始执行程序后，若使能看门狗(DSP_EWDG.EWDG_EN=1)，则看门狗计数器会同步开始计数，在每个时钟周期加 1。当看门狗计数器达到设定的溢出值(DSP_EWDG.TIM_OUT_VAL)时，如果程序还未执行完成，此时会产生看门狗溢出中断标志 DSP_STATUS.EWDG_IF(对应的中断使能为 DSP_EWDG.EWDG_IE)和复位请求(对应的使能为 DSP_EWDG.SYS_RST_EN)。
若程序执行结束后，看门狗计数器未达到 TIM_OUT_VAL，则会自动清除计数值，等到下次执行程序再重新开始计数。
在 DSP 暂停状态下（硬件断点调试），看门狗计数器不计数。
当 DSP_STATUS.EWDG_IF 为 1 时，软件和硬件触发都无法触发新的运算。

10.3.4 硬件断点

DSP 支持硬件断点调试，可以通过 DSP_DEBUG.BREAK_POINT_EN 进行使能，通过 DSP_DEBUG.BREAK_POINT 设置断点 PC 值。当 PC==BREAK_POINT 时，DSP 暂停执行，所有寄存器保持不变，同时触发断点中断（对应的中断标志为 DSP_STATUS.BREAK_POINT_IF，对应的中断使能为 DSP_DEBUG.BREAK_POINT_IE)。
通过软件将 DSP_STATUS.BREAK_POINT_IF 写 1 清 0 后，DSP 可以继续运行。
当 DSP 处于暂停状态时，断点值可以被修改，指令存储器不能被读写，通用数据寄存器和静态数据寄存器可以被读取，但不能被写入。

10.3.5 状态与中断

当 DSP 执行程序时，DSP_STATUS.BUSY 为 1，软件可以读取 DSP_STATUS.PC 来确定 DSP 指令计数器值，判断 DSP 正在执行哪条指令。
触发中断场景如下：

- 当 PC==PC_END 时，表示程序执行结束，将触发完成中断（对应的中断标志为 DSP_STATUS.DONE_IF，对应的中断使能为 DSP_CTRL.DONE_IE）和 DMA 请求（对应的使能为 DSP_CTRL.DONE_DE)。
- 若执行除法指令时除数为 0，将触发除数为 0 中断（对应的中断标志为 DSP_STATUS.DIV_ZERO_IF，对应的中断使能为 DSP_CTRL.DIV_ZERO_IE)。
- 若执行到未定义的错误指令,将触发错误指令中断(对应的中断标志为 DSP_STATUS.INS_ERR_IF，对应的中断使能为 DSP_CTRL.INS_ERR_IE)。
- 若在执行过程中再次被触发（可来自软件或硬件)，将触发重复触发中断（对应的中断标志为 DSP_STATUS.UDR_IF，对应的中断使能为 DSP_CTRL. UDR_IE)。

10.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 10-6 DSP 寄存器概述

偏移地址	寄存器名	寄存器描述	复位值
0x000	DSP_CR	DSP控制寄存器	0x00000000
0x004	DSP_START	DSP软件触发寄存器	0x00000000
0x008	DSP_SR	DSP状态寄存器	0x00000000
0x00C	DSP_EWDG	DSP看门狗配置寄存器	0x00000000
0x010	DSP_DEBUG	DSP硬件断点寄存器	0x00000000
0x040+x*4	DSP_GPRx	DSP通用数据寄存器x	0xXXXXXXXX
0x080+x*4	DSP_SDRx	DSP静态数据寄存器x	0xXXXXXXXX
0x400+x*4	DSP_INSx	DSP指令存储器x	0x00000000

其中，通用数据寄存器 DSP_GPRx，x 取值为 0~15。静态数据寄存器 DSP_SDRx，x 取值为 0~31。指令存储器 DSP_INSx，x 取值为 0~255。地址均为 32 位对齐。

10.4.1 DSP 控制寄存器(DSP_CTRL)

偏移地址：0x000
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DONE_DE	DONE_IE	DMA_TRIG_SEL[2:0]			DMA_TRIG_EN	EN	PC_END[8:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INS_ERR_IE	UDR_IE	DIV_ZERO_IE	Res.	Res.	Res.	Res.	PC_START[8:0]								
rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31	DONE_DE	发送 DMA 请求使能。 0：不使能 1：使能
30	DONE_IE	完成中断使能。 0：不使能 1：使能
29:27	DMA_TRIG _SEL	选择 DMA 某个通道作为 DSP 硬件触发源。 000：选择 DMA 通道 0

		001：选择 DMA 通道 1 010：选择 DMA 通道 2 011：选择 DMA 通道 3 100：选择 DMA 通道 4 其它：保留，不选择 DMA 通道触发。
26	DMA_TRIG_EN	DMA 一轮数据传输完成触发 DSP 运算使能。 0：不使能 1：使能
25	EN	模块使能，DSP 只有在使能后才可以被触发执行程序。 0：不使能 1：使能
24:16	PC_END	程序结束地址（Busy 状态下不可更改）。
15	INS_ERR_IE	错误指令中断使能。 0：不使能 1：使能
14	UDR_IE	重复触发中断使能。 0：不使能 1：使能
13	DIV_ZERO_IE	除数为 0 中断使能。 0：不使能 1：使能
12:9	Res.	保留，必须保持复位值。
8:0	PC_START	程序开始地址（Busy 状态下不可更改）。

10.4.2 DSP 软件触发寄存器(DSP_START)

偏移地址：0x004
复位值：0x00000000

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

Bit	Field	Description
31:1	Res.	保留，必须保持复位值。
0	START	软件写 1 启动程序，硬件自动清零； Busy 状态下，写 1 不能生效；

		配置 START 之前必须先配置使能 DSP_CTRL.EN.

10.4.3 DSP 状态寄存器(DSP_STATUS)

偏移地址：0x008
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INS_ERR_IF	UDR_IF	BREAK_POINT_IF	EWDG_IF	DONE_IF	DIV_ZERO_IF	BUSY
									w1c	w1c	w1c	w1c	w1c	w1c	ro
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC[8:0]								
							ro	ro	ro	ro	ro	ro	ro	ro	ro

Bit	Field	Description
31:23	Res.	保留，必须保持复位值。
22	INS_ERR_IF	错误指令中断标志。 0：未发生 1：监测到错误指令
21	UDR_IF	重复触发中断标志。 0：未发生 1：重复触发
20	BREAK_POINT_IF	硬件断点中断标志。 0：未发生 1：遇到硬件断点
19	EWDG_IF	看门狗计数器溢出中断标志。 0：未溢出 1：看门狗计数器溢出
18	DONE_IF	完成中断标志。 0：未完成 1：程序执行完毕
17	DIV_ZERO_IF	除数为 0 中断标志。 0：除数不为 0 1：除数为 0
16	BUSY	DSP 运行状态 0：空闲 1：正在运行
15:9	Res.	保留，必须保持复位值。
8:0	PC	指令计数值

10.4.4 看门狗配置寄存器(DSP_EWDG)

偏移地址：0x00C
复位值：0x00000000

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

Bit	Field	Description
31:19	Res.	保留，必须保持复位值。
18	SYS_RST_EN	看门狗计数器超过溢出值后请求系统复位使能。 0：不使能 1：使能
17	EWDG_IE	看门狗计数器溢出中断使能。 0：不使能 1：使能
16	EWDG_EN	看门狗使能。 0：不使能 1：使能
15:10	Res.	保留，必须保持复位值。
9:0	TIM_OUT_VAL	看门狗计数器溢出值

10.4.5 硬件断点寄存器(DSP_DEBUG)

偏移地址：0x010
复位值：0x00000000

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

Bit	Field	Description
31:17	Res.	保留，必须保持复位值。
17	BREAK_POINT_IE	硬件断点中断使能。 0：不使能 1：使能
16	BREAK_POINT_EN	硬件断点使能。 0：不使能 1：使能
15:9	Res.	保留，必须保持复位值。
8:0	BREAK_POINT	硬件断点 PC 值，当 PC==BREAK_POINT 时，暂停执行程序。

10.4.6 通用数据寄存器 x(DSP_GPRx)(x=0~15)

偏移地址：0x040+x*4
复位值：0xXXXXXXXX

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	DATA	数据

10.4.7 静态数据寄存器 x(DSP_SDRx)(x=0~31)

偏移地址：0x080+x*4
复位值：0xXXXXXXXX

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	DATA	数据

10.4.8 指令存储器 x(DSP_INSx)(x=0~255)

偏移地址：0x400+x*4
复位值：0x000000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INS1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INS0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:16	INS1	指令 1
15:0	INS0	指令 0

11 32 位通用定时器（TIM）

11.1 简介

32 位通用定时器 (TIM)，定时器可以对输入的时钟进行计数，并在计数值从设定值变为零时触发中断。通用定时器具备单次计数和周期计数两种模式。

11.2 主要特性

- 32 位向下计数器，32 位自动装载值寄存器
- 单次计数/周期计数两种模式
- 计数完成触发中断和 DMA 请求

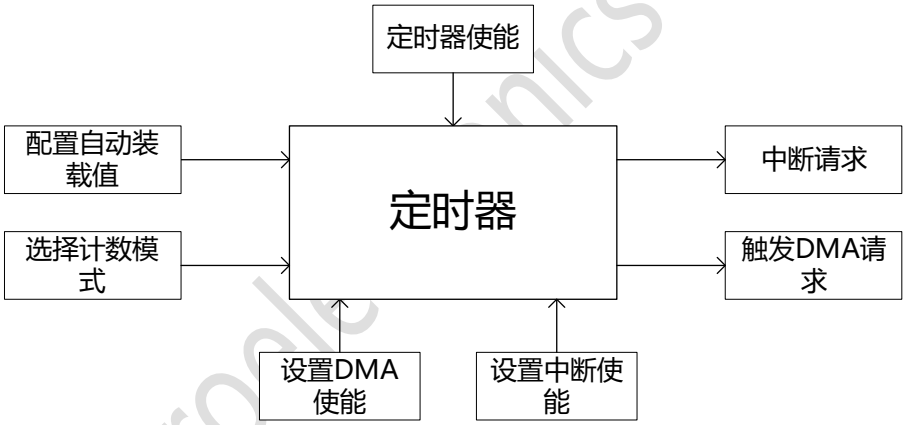

图 11-1 定时器工作原理

11.3 功能说明

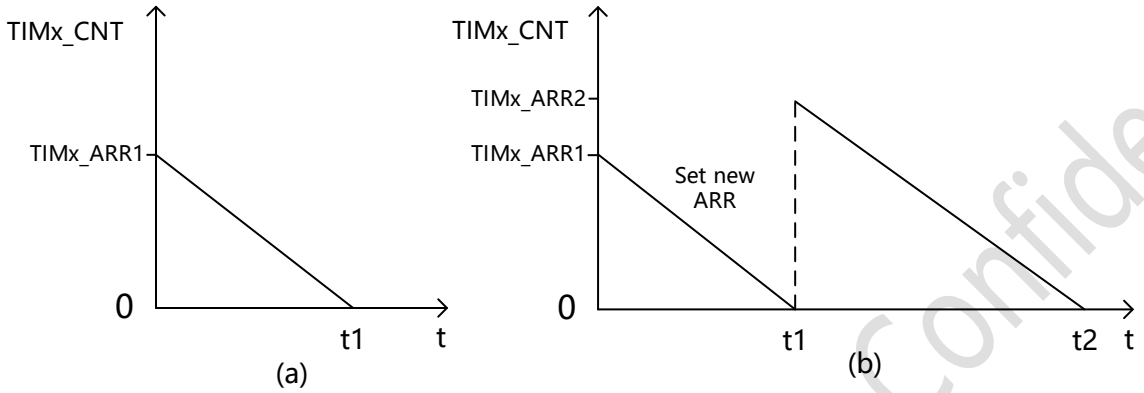

图 11-2 (a)单次计数模式，(b)周期计数模式

- 支持单次计数和周期计数两种模式。
- 直接使用 PCLK 作为计数时钟。
- 在使能定时器之前,需要将计数初值写入自动装载值寄存器（TIMx_ARR）。
- 若设置为单次计数模式（TIMx_CR. SINGLE=1），则在向下计数为 0 之后停止计数并触发计数完成中断(对应的中断标志为 TIMx_SR.IF，对应的中断使能为 TIMx_CR.IE）和 DMA 请求(对应的 DMA 请求使能为 TIMx_CR.DE)，同时硬件会自动清除定时器使能 TIMx_CR. EN.
- 若设置为周期计数模式（TIMx_CR. SINGLE=0），则每当向下计数到 0 后，硬件会自动将计数值更新为装载值，并触发计数完成中断和 DMA 请求，随后重新开始向下计数，直到通过软件关闭定时器(TIMx_CR.EN=0).
- 定时器计数过程中可以通过读取 TIMx_CNT 寄存器来获取当前计数值。
- 自动装载值寄存器(TIMx_ARR)在计数过程中可以被修改，但直到下一个计数周期才会正式生效。如图 11-2(b)所示，如果在 0-t1 之间的某个时刻设置新的装载值，那么直到 t1 时刻才会将其赋值给计数器。
- 如果在计数过程中关闭定时器，计数值将保持不变。直到重新启动定时器后，计数值会被更新为装载值。
- 在计数过程中可以通过软件主动修改计数值。

11.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 11-1 TIMx 寄存器概述

偏移地址	寄存器名	寄存器描述	复位值
0x000	TIMx_CNT	TIMx 计数值寄存器	0x00000000
0x004	TIMx_ARR	TIMx 自动装载值寄存器	0x00000000
0x008	TIMx_CR	TIMx 控制配置寄存器	0x00000000
0x00c	TIMx_SR	TIMx 状态配置寄存器	0x00000000

11.4.1 TIMx 计数值寄存器（TIMx_CNT）

偏移地址：0x000
复位值：0x00000000

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

Bit	Field	Description
31:0	CNT	当前计数器值（Counter value）.

11.4.2 TIMx 自动装载值寄存器（TIMx_ARR）

偏移地址：0x004
复位值：0x00000000

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

Bit	Field	Description
31:0	ARR	自动重装载值 (Auto-reload value).

11.4.3 TIMx 控制配置寄存器（TIMx_CR）

偏移地址：0x008
复位值：0x00000000

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

Bit	Field	Description
31:4	Res.	保留，必须保持复位值。
3	DE	触发 DMA 请求使能（DMA Enable）： 0：不使能； 1：使能。
2	IE	计数完成中断使能（Interrupt Enable）： 0: 不使能； 1：使能。
1	SINGLE	计数模式选择(Cnt mode select)： 0：周期计数模式； 1：单次计数模式。
0	EN	定时器使能(Timer enable)： 0：禁止计数器； 1：使能计数器。

11.4.4 TIMx 状态配置寄存器（TIMx_SR）

偏移地址：0x00C
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IF
															w1c

Bit	Field	Description
31:1	Res.	保留，必须保持复位值。
0	IF	计数完成中断标志（Count completion interrupt flag）： 硬件置 1，软件写 1 清 0。

12 捕获比较定时器（CCT）

12.1 简介

捕获比较定时器具备 16 位向上计数器，支持单次计数和周期计数两种计数模式，计数时钟是经过预分频器的分频时钟。模块内部的定时器支持两种工作模式：捕获输入模式和比较输出模式。前者可以进行输入滤波和边沿检测，后者可以输出边沿模式的 PWM 信号。

12.2 主要特性

- 16 位向上计数器，16 位自动装载值寄存器
- 两个可以独立运行的通道
- 单次计数/周期计数两种计数模式
- 定时器使用预分频器产生的分频时钟进行计数，分频会立即生效
- 两种工作模式：输入捕获模式，输出比较模式
- PWM 生成（边沿模式）和单脉冲模式输出
- 每个通道可单独设置工作模式

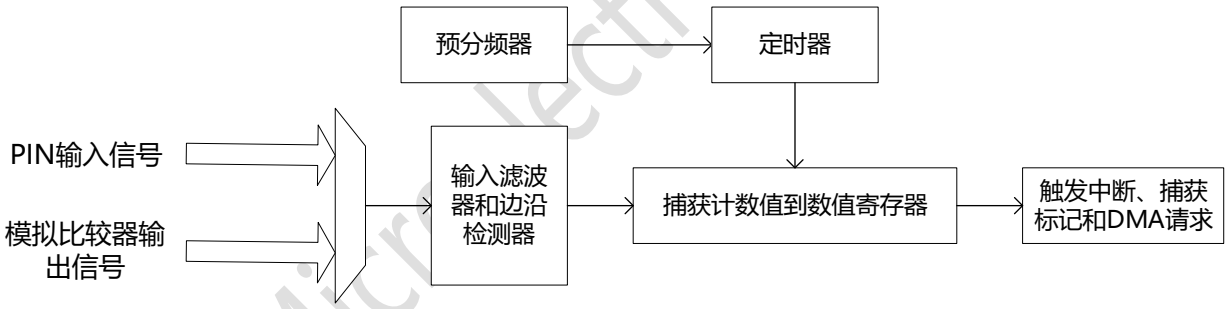

图 12-1 捕获模式

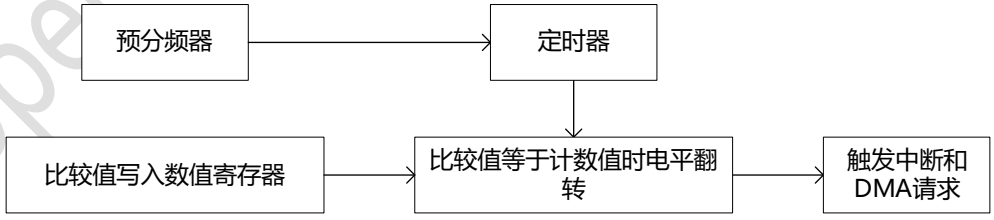

图 12-2 比较模式

12.3 功能说明

12.3.1 定时器

- 支持单次计数和周期计数两种模式。
- 定时器的计数时钟是经过预分频后的分频时钟，分频系数可通过 CCTx_CR. CLK_DIV 进行配置，取值为 1/2/4/8/16/32/64/128。
- 在使能定时器之前，需要先将计数溢出值写入自动装载值寄存器（CCTx_ARR)。
- 若设置为单次计数模式（CCTx_CR. SIGNLE=1)，则在向上计数到自动装载值后停止计数并触发装载值中断(对应的中断标志为 CCTx_SR. LVAL_IF，对应的中断使能为 CCTx_CR. LVAL_IE）和 DMA 请求（需要配置 CCTx_CR. DMA_SRC_SEL)，同时硬件会自动清除定时器使能 CCTx_CR.EN（计数值不清零)。
- 若设置为周期计数模式（CCTx_CR. SIGNLE=0)，则每当向上计数到自动装载值后，硬件会自动清零计数器并触发装载值中断和 DMA 请求，随后重新开始向上计数，直到通过软件关闭定时器(CCTx_CR. EN = 0)。
- 定时器计数过程中可以通过读取 CCTx_CNT 寄存器来获取当前计数值。
- 自动装载值寄存器（CCTx_ARR）在计数过程中可以被修改，但直到下一个计数周期才会正式生效。

12.3.2 捕获工作模式

- 两路通道(CCTx_CH0，CCTx_CH1)均可配置为捕获模式(CCTx_CR. CHx_MODE=0).每个通道有单独的使能信号(CCTx_CR. CHx_EN)，并可以单独配置工作参数。
- 捕获输入信号的源头可以选择为 PIN 输入或 ACMP 模拟比较器输出(CCTx_CR. INx_SEL).当选择模拟比较器输出作为信号源时，可以指定具体选择哪个模拟比较器(CCTx_CR. INx_ACMP_SEL)。
- 数字滤波器有两个参数：滤波时钟分频系数(CCTx_CAP_CFG. CHx_FLT_SAMPLE)和滤波长度(CCTx_CAP_CFG. CHx_FLT_LEN)。分频系数取值为 1/4/16/32，滤波长度取值为 8/16/32。在滤波分频时钟下，长度小于滤波长度的脉冲将会被过滤掉。
- 捕获的边沿类型可以被设置为只有上升沿、只有下降沿和同时捕获上升沿和下降沿(CCTx_CAP_CFG.CHx_EDGE)。
- 捕获事件发生后，计数器当前值会被锁存到数值寄存器 CCTx_CHx_VAL 中，相应的捕获中断(对应的中断标志为 CCTx_SR.CHx_CAP_IF，对应的中断使能为 CCTx_CR. CHx_CAP_IE)、边沿捕获标记 CCTx_SR.CHx_CAP_FE/CCTx_SR.CHx_CAP_RE 和 DMA 请求(需要配置 CCTx_CR. DMA_SRC_SEL)会被置 1。
- 通过读取 CCTx_SR 寄存器可以查看捕获到的具体边沿类型。边沿捕获标记和中断一样可以被软件写 1 清零。
- 可以通过 CCTx_CAP_CFG. CHx_CNT_CLR 设置捕获事件发生后是否重置计数器。

12.3.3 比较工作模式

- 在比较模式下，首先需要配置通道的初始输出状态 CCTx_CMP_CFG. CHx_INIT_O 和比较值 CCTx_CHx_VALUE.CHx_VAL。
- 模块启动时，将初始输出状态赋值给输出电平；每完成一次计数周期，将初始输出状态赋值给输出电平。
- 当计数值等于比较值时，输出电平翻转并触发比较事件中断(对应的中断标志为 CCTx_SR.CHx_CMP_IF, 对应的中断使能为 CCTx_CR. CHx_CMP_IE) 和 DMA 请求（需要配置 CCTx_CR. DMA_SRC_SEL)。
- 比较值 CCTx_CHx_VALUE.CHx_VAL 在计数过程中可以被修改，但直到下一个计数周期才会正式生效。

12.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 12-1 CCT 寄存器概述

偏移地址	寄存器名	寄存器描述	复位值
0x000	CCTx_CNT	CCTx计数值寄存器	0x00000000
0x004	CCTx_ARR	CCTx自动装载值寄存器	0x00000000
0x008	CCTx_CR	CCTx控制寄存器	0x00000000
0x00C	CCT_SR	CCTx状态寄存器	0x00000000
0x010	CCTx_CAP_CFG	CCTx捕获配置寄存器	0x00000000
0x014	CCTx_CMP_CFG	CCTx比较配置寄存器	0x00000000
0x018	CCTx_CH0_VAL	CCTx通道0数值寄存器	0x00000000
0x01C	CCTx_CH1_VAL	CCTx通道1数值寄存器	0x00000000

12.4.1 CCTx 计数值寄存器（CCTx_CNT）

偏移地址：0x000
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CNT	当前计数值(Counter value)。

12.4.2 CCTx 自动装载值寄存器（CCTx_ARR）

偏移地址：0x004
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	ARR	自动重装载值（Auto-reload value）。

12.4.3 CCTx 控制寄存器（CCTx_CR）

偏移地址：0x008
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	IN1_ACMP_SEL[1:0]		IN1_SEL	IN0_ACMP_SEL[1:0]		IN0_SEL	Res.	Res.	Res.	Res.	Res.	DMA_SRC_SEL[2:0]		
		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
CH1_CMP_IE	CH0_CMP_IE	CH1_CAP_IE	CH0_CAP_IE	LVAL_IE	CLK_DIV[2:0]			Res.	Res.	CH1_MODE	CH0_MODE	CH1_EN	CH0_EN	SINGLE	EN
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:30	Res.	保留，必须保持复位值。
29:28	IN1_ACMP_SEL	输入 1 选择模拟比较器源头（Input 1 analog compare source select）： 00：ACMP0；01：ACMP1； 10：ACMP2；11：ACMP3。
27	IN1_SEL	输入 1 选择捕获信号输入（Input 1 capture source select）： 0：选择从引脚输入； 1：选择从模拟比较器输入。
26:25	IN0_ACMP_SEL	输入 0 选择模拟比较器源头（Input 0 analog compare source select）： 00：ACMP0；01：ACMP1； 10：ACMP2；11：ACMP3。
24	IN0_SEL	输入 1 选择捕获信号输入（Input 1 capture source select）：

		0：选择从引脚输入； 1：选择从模拟比较器输入。
23:19	Res.	保留，必须保持复位值。
18:16	DMA_SRC_SEL	触发 DMA 请求事件选择（DAM request source select）： 000：复位值，CCT 不触发 DMA 请求； 001：计数器溢出触发 DMA 请求； 010：通道 0 捕获事件触发 DMA 请求； 011：通道 1 捕获事件触发 DMA 请求； 100：通道 0 比较事件触发 DMA 请求； 101：通道 1 比较事件触发 DMA 请求。
15	CH1_CMP_IE	通道 1 比较事件中断使能（Channel 1 compare event interrupt enable）： 0：不使能； 1：使能。
14	CH0_CMP_IE	通道 0 比较事件中断使能（Channel 0 compare event interrupt enable）： 0：不使能； 1：使能。
13	CH1_CAP_IE	通道 1 捕获事件中断使能（Channel 1 capture event interrupt enable）： 0：不使能； 1：使能。
12	CH0_CAP_IE	通道 0 捕获事件中断使能（Channel 0 capture event interrupt enable）： 0：不使能； 1：使能。
11	LVAL_IE	装载值中断使能（Load value interrupt enable）： 0：不使能； 1：使能。
10:8	CLK_DIV	时钟分频系数（Clock division）： 000：1 分频； 001：2 分频； 010：4 分频； 011：8 分频； 100：16 分频；101：32 分频；110：64 分频；111：128 分频。
7:6	Res.	保留，必须保持复位值。
5	CH1_MODE	通道 1 模式选择（Channel 1 mode select）： 0：捕获模式； 1：比较模式。
4	CH0_MODE	通道 0 模式选择（Channel 0 mode select）： 0：捕获模式； 1：比较模式。
3	CH1_EN	通道 1 使能（Channel 1 enable）： 0：不使能； 1：使能。 具体工作状态由模式选择决定。

2	CH0_EN	通道 0 使能（Channel 0 enable）： 0：不使能； 1：使能。 具体工作状态由模式选择决定。
1	SINGLE	计数模式选择(Cnt mode select)： 0：周期计数模式； 1：单次计数模式。
0	EN	定时器使能(Timer enable)： 0：禁止计数器； 1：使能计数器。

12.4.4 CCTx 状态寄存器（CCTx_SR）

偏移地址：0x00C
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CH1_CMP_IF	CH0_CMP_IF	Res.	CH1_CAP_FE	CH1_CAP_RE	CH0_CAP_FE	CH0_CAP_RE	CH1_CAP_IF	CH0_CAP_IF	LVAL_IF
						w1c	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c

Bit	Field	Description
31:10	Res.	保留，必须保持复位值。
9	CH1_CMP_IF	通道 1 比较事件中断标志（Channel 1 compare event interrupt flag）： 硬件置 1，软件写 1 清零。
8	CH0_CMP_IF	通道 0 比较事件中断标志（Channel 0 compare event interrupt flag）： 硬件置 1，软件写 1 清零。
7	Res.	保留，必须保持复位值。
6	CH1_CAP_FE	通道 1 下降沿捕获标志（Channel 1 capture falling edge flag）： 硬件置 1，软件写 1 清零。
5	CH1_CAP_RE	通道 1 上升沿捕获标志（Channel 1 capture rising edge flag）： 硬件置 1，软件写 1 清零。
4	CH0_CAP_FE	通道 1 下降沿捕获标志（Channel 0 capture falling edge flag）： 硬件置 1，软件写 1 清零。
3	CH0_CAP_RE	通道 0 上升沿捕获标志（Channel 0 capture rising edge flag）： 硬件置 1，软件写 1 清零。

2	CH1_CAP_IF	通道 1 捕获中断标志（Channel 1 capture interrupt flag）： 硬件置 1，软件写 1 清零。
1	CH0_CAP_IF	通道 0 捕获中断标志（Channel 0 capture interrupt flag）： 硬件置 1，软件写 1 清零。
0	LVAL_IF	装载值中断标志（ARR interrupt flag）： 硬件置 1，软件写 1 清零。

12.4.5 CCTx 捕获配置寄存器（CCTx_CAP_CFG）

偏移地址：0x010
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CH1_EDGE[1:0]		CH1_CNTR_CLR	CH1_FLT_LEN[1:0]		CH1_FLT_SAMPLE[1:0]		Res.	CH0_EDGE[1:0]		CH0_CNTR_CLR	CH0_FLT_LEN[1:0]		CH0_FLT_SAMPLE[1:0]	
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:15	Res.	保留，必须保持复位值。
14:13	CH1_EDGE	通道 1 捕获边沿选择（Channel 1 edge select）： 00：无动作（No Action）； 01：捕获上升沿（Rising Edge）； 10：捕获下升沿（Falling Edge）； 11：捕获上升沿和下升沿（Both Edge）。
12	CH1_CNT_CLR	通道 1 清零控制（Channel 1 control clear）： 0：捕获边沿之后计数器不清零； 1：捕获边沿之后计数器清零。
11:10	CH1_FLT_LEN	通道 1 滤波长度（Channel 1 filter length）： 00：保留值，禁止配置；　01：通道 1 滤波长度为 8； 10：通道 1 滤波长度为 16；　11：通道 1 滤波长度为 32；
9:8	CH1_FLT_SAMPLE	通道 1 滤波时钟分频系数选择（Channel 1 filter clock sample select）： 00：通道 1 滤波时钟分频系数为 1； 01：通道 1 滤波时钟分频系数为 4； 10：通道 1 滤波时钟分频系数为 16； 11：通道 1 滤波时钟分频系数为 32；

7	Res.	保留，必须保持复位值。
6:5	CH0_EDGE	通道 0 捕获边沿选择（Channel 0 edge select）： 00：无动作（No Action）； 01：捕获上升沿（Rising Edge）； 10：捕获下升沿（Falling Edge）； 11：捕获上升沿和下升沿（Both Edge）。
4	CH0_CNT_CLR	通道 0 控制清零（Channel 0 control clear）： 0：捕获边沿之后计数器不清零； 1：捕获边沿之后计数器清零。
3:2	CH0_FLT_LEN	通道 0 滤波长度（Channel 0 filter length）： 00：保留值，禁止配置；　01：通道 0 滤波长度为 8； 10：通道 0 滤波长度为 16；　11：通道 0 滤波长度为 32；
1:0	CH0_FLT_SAMPLE	通道 0 滤波时钟分频系数选择（Channel 0 filter clock sample select）： 00：通道 0 滤波时钟分频系数为 1； 01：通道 0 滤波时钟分频系数为 4； 10：通道 0 滤波时钟分频系数为 16； 11：通道 0 滤波时钟分频系数为 32；

12.4.6 CCTx 比较配置寄存器（CCTx_CMP_CFG）

偏移地址：0x014
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH1_INIT_O	CH0_INIT_O
														rw	rw

Bit	Field	Description
31:2	Res.	保留，必须保持复位值。
1	CH1_INIT_O	通道 1 初始输出状态（Channel 1 initial output）：
0	CH0_INIT_O	通道 0 初始输出状态（Channel 0 initial output）：

12.4.7 CCTx 通道 0 数值寄存器（CCTx_CH0_VAL）

偏移地址：0x018
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CH0_VAL	通道 0 数值（Channel 0 value）： 在捕获模式下为通道 0 捕获到的计数值； 在比较模式下为通道 0 输出比较值。

12.4.8 CCTx 通道 1 数值寄存器（CCTx_CH1_VAL）

偏移地址：0x01C
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CH1_VAL	通道 1 数值（Channel 1 value）： 在捕获模式下为通道 1 捕获到的计数值； 在比较模式下为通道 1 输出比较值。

13 增强型 PWM 定时器（EPWM）

13.1 简介

EPWM (Enhanced Pulse Width Modulation)，即增强型的脉冲宽度调制，支持输出4通道8路PWM。

13.2 主要特性

- 16 位计数器，支持增减计数（中央对齐模式），单增计数两种模式
- 支持单次计数/周期计数两种模式
- 计数时钟分频系数 1/2/4/8/16/32/64/128
- 可配置完成周期计数产生 DMA 请求
- 支持 4 通道 PWM 生成模块，可产生 4 对互补（8 路）PWM 信号，支持 PWM 移相
- 支持死区插入
- 支持强制改变 I/O 输出状态
- 支持产生 2 路同步触发 ADC 采样信号，与 PWM 时间基准相同。
- 支持多路硬件急停信号，支持急停信号的硬件滤波和极性选择，急停之后的 I/O 输出状态可设置。

13.3 功能模块说明

13.3.1 EPWM 模块结构框图

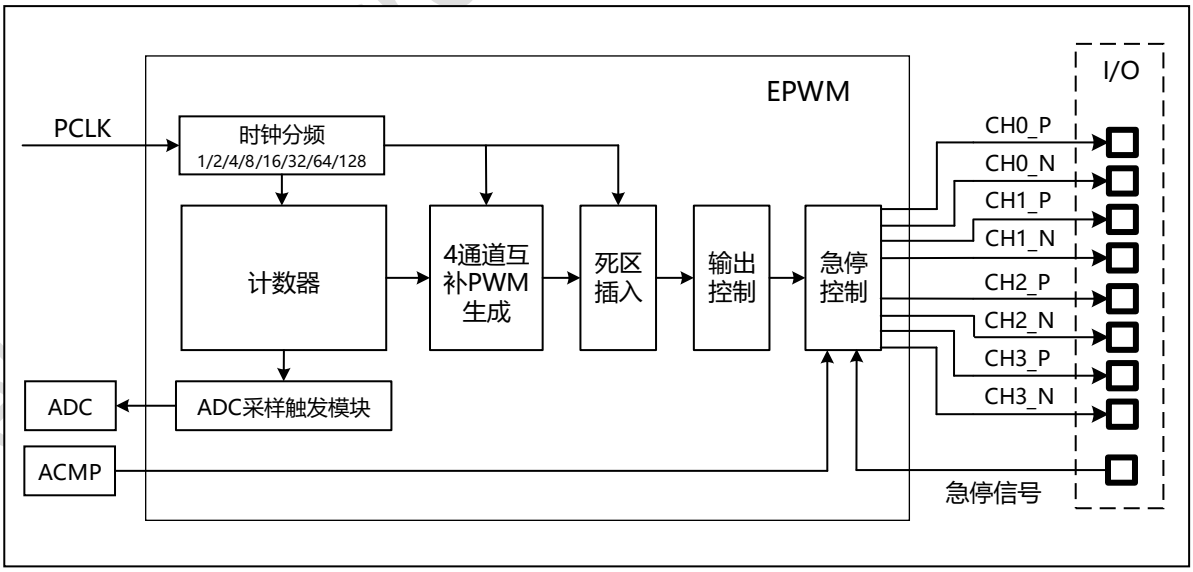

图 13-1 EPWM 模块结构框图

13.3.2 计数模块

EPWM 核心为一个 16 位计数器，其计数自动装载值为 EPWM_ARR。可通过 EPWM_CR.MODE 选择计数模式：单增计数或增减计数，配置 EPWM_CR.EN 为 1 后，计数器从 0 时刻开始从计数，如图 13-2 所示。

计数模块的计数时钟可配置为 PCLK 时钟的分频时钟（EPWM_CR.CLK_DIV），支持分频系数为 1/2/4/8/16/32/64/128，计数器在每个分频时钟有效时更新计数值。

单次计数/周期计数两种模式可通过配置 EPWM_CR.SINGLE 实现，单次计数仅执行一次周期计数，完成一次单次计数后，硬件自动清除 EPWM_CR.EN，软件重新配置 EPWM_CR.EN 为 1 时触发新一次周期计数。

当计数一个完整周期结束时，可配置产生中断，中断使能位为 EPWM_CR.PERIOD_IE，中断标志位为 EPWM_SR.PERIOD_IF，中断标记由硬件置 1，软件写 1 清除。

当计数值等于 ARR 时，可配置产生中断，中断使能位为 EPWM_CR.LOAD_VAL_IE，中断标志位为 EPWM_SR.LOAD_VAL_IF，中断标记由硬件置 1，软件写 1 清除。

当计数一个完整周期结束时，可配置产生 DMA 请求，DMA 请求使能位 EPWM_CR.PERIOD_DE。

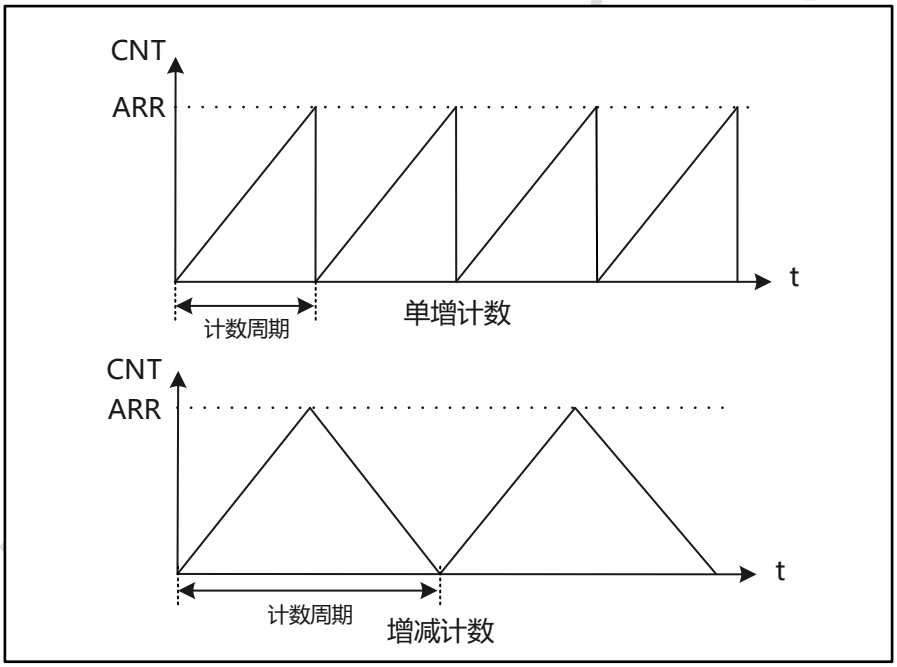

图 13-2 EPWM 计数模式

13.3.3 4 通道 PWM 互补输出

EPWM 包含 4 通道互补输出，每个通道可输出互补的一组 PWM（CHx_P 和 CHx_N，CHx 可为 CH0，CH1，CH2，CH3），每个通道有单独的通道使能(EPWM_CR.CHx_EN)，配置使能后该通道才能输出 PWM。

可配置 CHx_P 输出状态的初始输出状态 EPWM_CH_CR.CHx_INIT_O，CHx_N 与此相反，在完成每一个计数周期后，硬件会将通道的输出复位为初始输出。

每个通道有两个比较值：EPWM_CHx_CMP1，EPWM_CHx_CMP2。可单独配置每个比较值是否使能(EPWM_CMP_CFG.CHx_CMPx_EN)。可设置比较值在计数器增计数时生效，或减计时生效

(EPWM_CMP_CFG.CHx_CMPx_DIR)，两个比较值可设置为都在增计数时生效，或减计数时生效，或分别在增计数和减计数生效。

对于单增计数模式，可配置比较值都在增计数生效。

当 EPWM_CNT == EPWM_CHx_CMPx，且生效方向匹配时，PWM 生成模块会依据通道配置情况改变或保持 PWM 输出电平，具体行为由 EPWM_CH_CR.CHx_CMPx_ACT 来决定，可设置为输出 0，输出 1，输出极性翻转，或输出保持不变。

图 13-3 给出了不同 CHx_CMPx_ACT 配置情况下的互补输出波形。

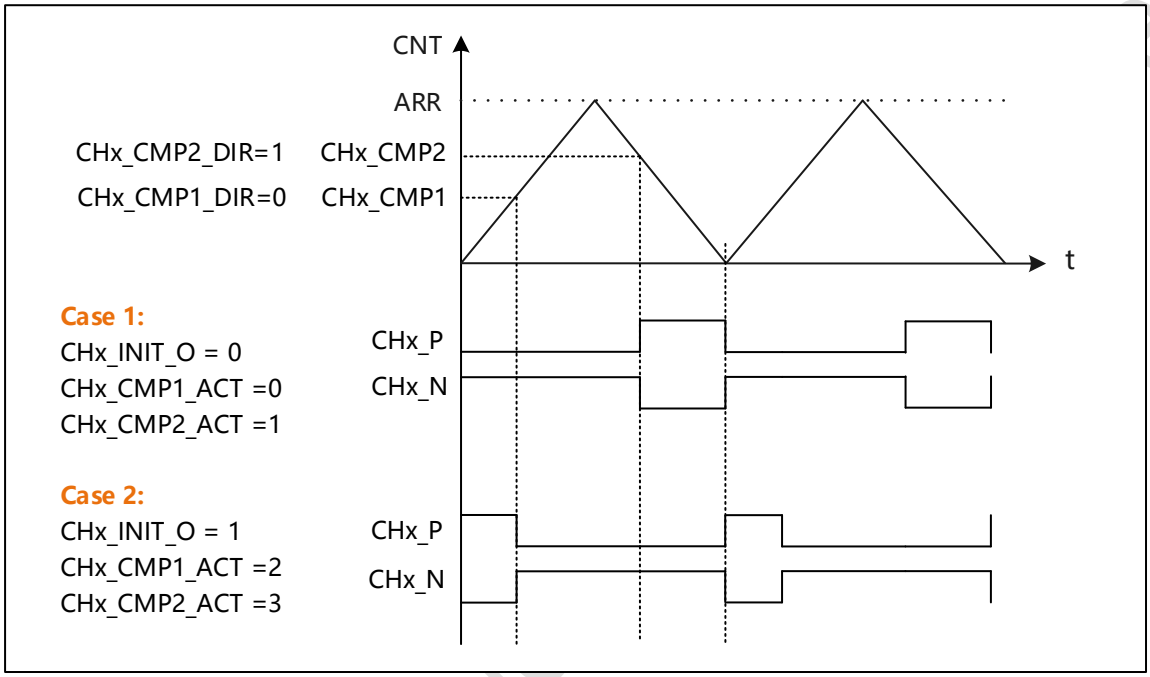

图 13-3 输出互补 PWM 示意

13.3.4 触发 ADC 采样

支持以计数器作为时间基准产生 ADC 采样触发信号，可设置两个 ADC 采样比较值 EPWMx_ADC_CMP1 和 EPWMx_ADC_CMP2。与通道比较值类似，使用时需要配置 EPWM_CMP_CFG.ADC_CMPx_EN 使能，可配置 ADC 采样比较值在计数器增计数时生效或减计时生效（EPWM_CMP_CFG.ADC_CMPx_DIR）。

当 EPWM_CNT == EPWM_ADC_CMPx，且生效方向匹配时，触发 ADC 进行采样。

可配置触发 ADC 采样时产生中断，中断使能位为 EPWM_SR.ADC_CMPx_IE，中断标记位为 EPWM_SR.ADC_CMPx_IF。

13.3.5 死区插入

4 对互补输出的 PWM 可通过设置 EPWM_DT_CR.DT_LEN 插入死区，所有通道的死区宽度相同。死区插入的时钟和计数时钟相同，死区插入原理如图 13-4 所示。

当 DT_LEN == 0 时，不插入死区。

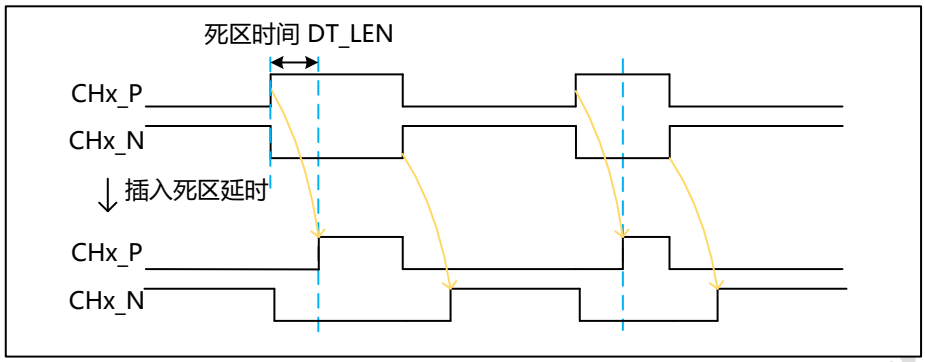

图 13-4 死区插入原理

13.3.6 急停控制

急停控制模块主要用于在出现异常时迅速关断或改变 PWM 输出，以免对硬件造成不可逆的损坏。
EPWM 急停信号触发来源：模拟比较器输出信号（可选触发源）；外部急停信号（来自 I/O 引脚输入）；软件急停。急停模块结构如图 13-5 所示。
当急停信号有效时，8 路 PWM 输出切换为 EPWM_STOP_CR.CHx_P/N_STOP_O 状态，可配置为急停时输出为高电平或者低电平。

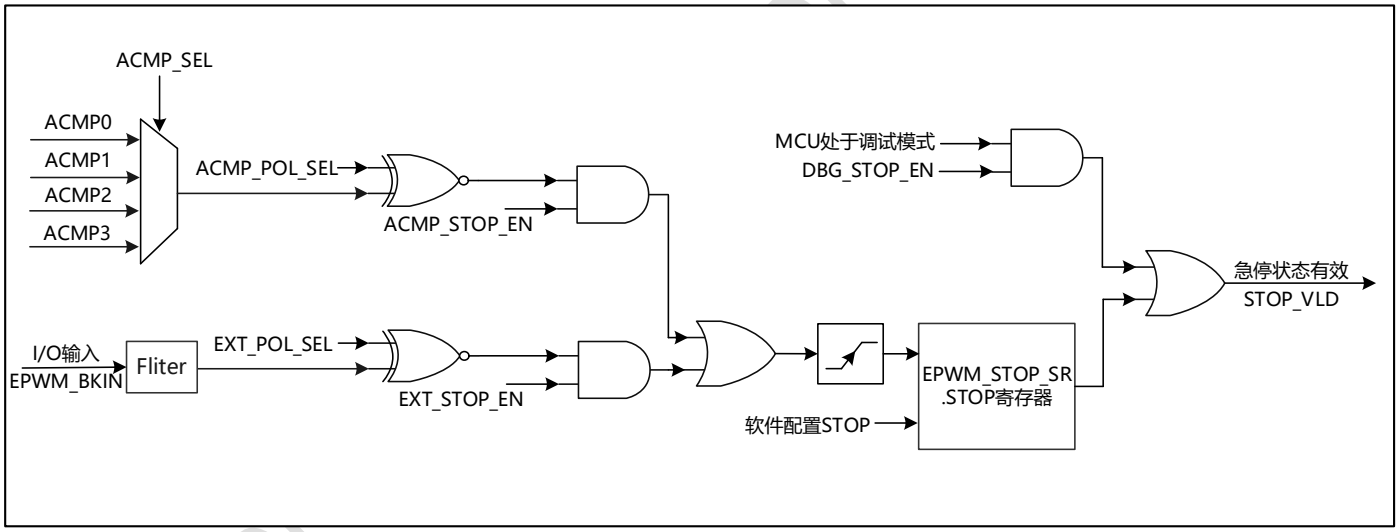

图 13-5 急停控制模块

（1）模拟比较器输出信号

配置 EPWM_STOP_CR.ACMP_SEL 选择 ACMP 输出，EPWM_STOP_CR.ACMP_POL_SEL 配置输入有效极性。
配置 EPWM_STOP_CR.ACMP_STOP_EN 使能模拟比较器输出信号触发急停。

（2）外部急停信号

配置 EXT_STOP_EN 使能外部急停信号输入(数字复用功能 EPWM_BKIN)，可配置外部急停信号有效极性 EPWM_STOP_CR.EXT_POL_SEL。
内置滤波器可对输入信号进行滤波，滤波器时钟为 PCLK，可配置滤波长度寄存器(EPWM_STOP_CR.FILT_LEN)和滤波采样频率寄存器(EPWM_STOP_CR.FILT_SAMPLE)，选择滤波采样点数目和滤波采样频率。滤波器以固定采样频率采样输入电平，连续采样到多个采样点的相同电平时，

输出该电平。

（3）软件急停

软件对 EXT_STOP_SR.STOP 写 1 设置软件急停，写 0 恢复运行状态，解除急停状态。软件急停不会触发急停中断(也即 EPWM_SR.STOP_IF 不会被硬件置 1)。

软件可通过读取 EXT_STOP_SR.ACMP_TRIG，EXT_STOP_SR.EXT_TRIG，查询硬件产生急停事件。

硬件急停（EXT 或 ACMP）事件发生时可配置产生中断（EPWM_CR.STOP_IE），中断标记位为 EPWM_SR.STOP_IF。

当 EXT 或 ACMP 刹车信号有效时，硬件会将 EXT_STOP_SR.STOP 置 1，软件对 STOP 写 0 时，硬件会自动清除 ACMP_TRIG，EXT_TRIG，STOP 标记。

在连接调试器进行调试时，可配置使能 EPWM_STOP_CR.DBG_STOP_EN 为 1，在断点或单步调试时使得 EPWM 输出为急停状态。

13.3.7 输出控制（输出取反/强制输出电平）

硬件支持 EPWM 通道将 PWM 取反输出，配置 EPWM_CH_CR.CHx_PP 或者 CHx_NP 生效。

8 路互补输出支持对单路强制设置输出电平。

当配置 EPWM_OUT_CR.CHx_P/N_FORCE_EN 为 1 时，可以强制切换对应通道的输出为 EPWM_OUT_CR.CHx_P/N_FORCE_O，配置为 0 时，解除强制输出状态，默认输出生成的 PWM 波形，EPWM 输出结构如图 13-6 所示。

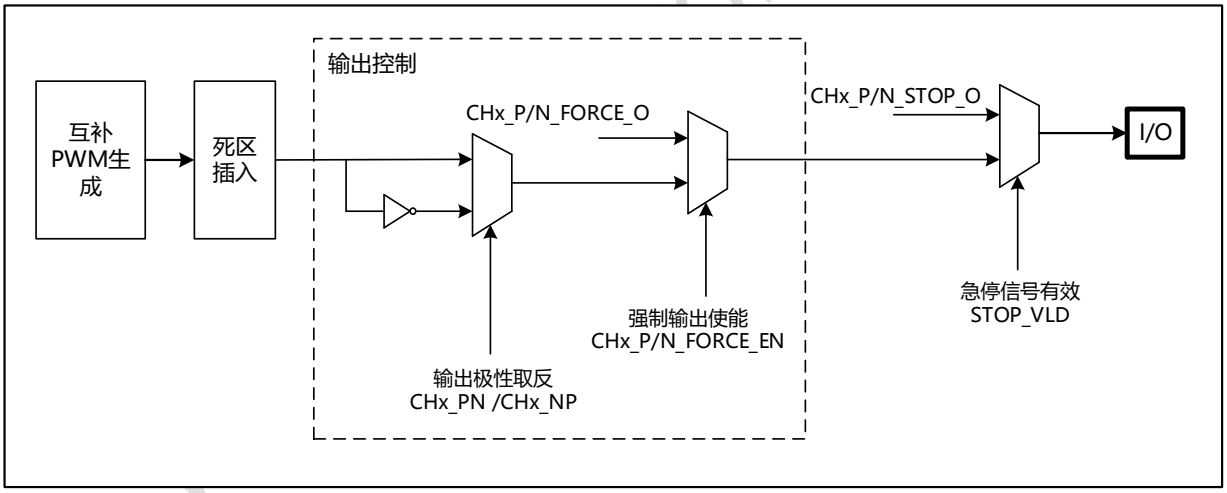

图 13-6 EPWM 输出结构

13.3.8 寄存器更新

EPWM 部分寄存器具有影子寄存器（Shadow Register），当软件对这部分寄存器进行配置时，并不会立即生效，在下述情况下才会将影子寄存器更新到工作寄存器。

（1）**硬件更新**：当完成一次计数周期时，硬件自动更新所有具有影子寄存器的寄存器。

（2）**软件更新**：对 EPWMx_UPDATE.UPDATE 写 1 可以触发更新所有具有影子寄存器的寄存器。

具有影子寄存器的寄存器：

EPWM_ARR

EPWM_ADC_CMP1，EPWM_ADC_CMP2

EPWM_CH0_CMP1，EPWM_CH0_CMP2
EPWM_CH1_CMP1，EPWM_CH1_CMP2
EPWM_CH2_CMP1，EPWM_CH2_CMP2
EPWM_CH3_CMP1，EPWM_CH3_CMP2

13.3.9 寄存器 LOCK 锁定保护

为了防止寄存器被错误更改，对部分寄存器进行了 LOCK 锁定保护。
被 LOCK 锁定保护的寄存器：
EPWM_ARR
EPWM_CR
EPWM_CMP_CFG
EPWM_UPDATE
EPWM_CH_CR
EPWM_OUT_CR
EPWM_DT_CR
EPWM_STOP_CR，EPWM_STOP_SR
对 EPWM_LOCK 寄存器写 0x900D0001 时 (必须按 Word 整体写入，按 Half word，Byte 写入无效)，受保护寄存器会被锁定，无法写入，可以读取。在锁定状态下写受保护寄存器会触发 Hard Fault 异常。
对 EPWM_LOCK 寄存器写 0x900D0000 时 (必须按 Word 整体写入，按 Half word，Byte 写入无效)，解除锁定状态，所有寄存器值都可以被写入。

13.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 13-1 EPWM 寄存器概览

偏移地址	寄存器名	寄存器描述	复位值
0x000	EPWM_CNT	EPWM计数器计数值寄存器	0x00000000
0x004	EPWM_ARR	EPWM自动重装载值寄存器	0x00000000
0x008	EPWM_CR	EPWM控制寄存器	0x00000000
0x00C	EPWM_SR	EPWM状态寄存器	0x00000000
0x010	EPWM_CMP_CFG	EPWM比较值配置寄存器	0x00000000
0x014	EPWM_ADC_CMP1	EPWM ADC触发比较值1配置寄存器	0x00000000
0x018	EPWM_ADC_CMP2	EPWM ADC触发比较值2配置寄存器	0x00000000
0x01C	EPWM_CH0_CMP1	EPWM通道0比较值1寄存器	0x00000000
0x020	EPWM_CH0_CMP2	EPWM通道0比较值2寄存器	0x00000000
0x024	EPWM_CH1_CMP1	EPWM通道1比较值1寄存器	0x00000000
0x028	EPWM_CH1_CMP2	EPWM通道1比较值2寄存器	0x00000000
0x02C	EPWM_CH2_CMP1	EPWM通道2比较值1寄存器	0x00000000
0x030	EPWM_CH2_CMP2	EPWM通道2比较值2寄存器	0x00000000
0x034	EPWM_CH3_CMP1	EPWM通道3比较值1寄存器	0x00000000
0x038	EPWM_CH3_CMP2	EPWM通道3比较值2寄存器	0x00000000
0x03C	EPWM_UPDATE	EPWM更新事件控制寄存器	0x00000000
0x040	EPWM_CH_CR	EPWM通道控制寄存器	0x00000000
0x044	EPWM_OUT_CR	EPWM输出控制寄存器	0x00000000
0x048	EPWM_DT_CR	EPWM死区控制寄存器	0x00000000
0x04C	EPWM_STOP_CR	EPWM急停控制寄存器	0x00000000
0x050	EPWM_STOP_SR	EPWM急停状态寄存器	0x00000000
0x054	EPWM_LOCK	EPWM锁定保护寄存器	0x00000000

13.4.1 计数器计数值寄存器（EPWM_CNT）

偏移地址：0x000
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CNT	当前计数器值 (Counter value)

13.4.2 自动重装载值寄存器（EPWM_ARR）

偏移地址：0x004
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	ARR	自动重装载值（Auto-reload value）。

13.4.3 控制寄存器（EPWM_CR）

偏移地址：0x008
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIOD_DE
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP_IE	ADC_CMP2_IE	ADC_CMP1_IE	PERIOD_IE	LOAD_VAL_IE	CLK_DIV[2:0]			CH3_EN	CH2_EN	CH1_EN	CH0_EN	Res.	MODE	SINGLE	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

Bit	Field	Description
31:17	Res.	保留，必须保持复位值。
16	PERIOD_DE	计数一个完整周期产生 DMA 请求（Period DMA enable） 0：禁止　　1：使能
15	STOP_IE	硬件触发急停进入 STOP 状态中断使能位（Stop interrupt enable） 0：禁止　　1：使能
14	ADC_CMP2_IE	计数器值等于 ADC 比较值 2 中断使能（ADC compare value 2 interrupt enable） 0：禁止　　1：使能

Bit	Field	Description
13	ADC_CMP1_IE	计数器值等于 ADC 比较值 1 中断使能（ADC compare value 2 interrupt enable) 0：禁止　　1：使能
12	PERIOD_IE	计数一个完整周期产生中断使能（Period interrupt enable) 0：禁止　　1：使能
11	LOAD_VAL_IE	计数器等于自动装载值时产生中断使能（Load value interrupt enable）： 0：禁止　　1：使能
10:8	CLK_DIV	时钟分频系数（Clock division factor) 000：1 分频　　001：2 分频 010：4 分频　　011：8 分频 100：16 分频　　101：32 分频 110：64 分频　　111：128 分频
7	CH3_EN	通道 3 输出 PWM 使能（Channel 3 enable）： 0：禁止　　1：使能
6	CH2_EN	通道 2 输出 PWM 使能（Channel 2 enable）： 0：禁止　　1：使能
5	CH1_EN	通道 1 输出 PWM 使能（Channel 1 enable）： 0：禁止　　1：使能
4	CH0_EN	通道 0 输出 PWM 使能（Channel 0 enable）： 0：禁止　　1：使能
3	Res.	保留，必须保持复位值。
2	MODE	模式选择（Mode selection）： 0：增减计数　　1：单增计数。
1	SINGLE	计数模式（Count mode) 1：单次计数模式 0：周期计数模式。
0	EN	定时器使能位（Counter enable) 0：禁止　　1：使能

13.4.4 状态寄存器（EPWM_SR)

偏移地址：0x00C
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOP_IF	ADC_CMP2_IF	ADC_CMP1_IF	PERIOD_IF	LOAD_VAL_IF
											w1c	w1c	w1c	w1c	w1c

Bit	Field	Description
31:5	Res.	保留，必须保持复位值。
4	STOP_IF	硬件触发急停进入 STOP 状态中断标记(Stop interrupt flag) 由硬件置 1，软件写 1 清 0。
3	ADC_CMP2_IF	计数器值等于 ADC 比较值 2 中断标记(ADC compare value 2 interrupt flag) 由硬件置 1，软件写 1 清 0。
2	ADC_CMP1_IF	计数器值等于 ADC 比较值 1 中断标记(ADC compare value 1 interrupt flag) 由硬件置 1，软件写 1 清 0。
1	PERIOD_IF	计数一个完整周期中断标记(Period interrupt flag) 由硬件置 1，软件写 1 清 0。
0	LOAD_VAL_IF	计数器值等于自动装载值中断标记(Load value interrupt flag) 由硬件置 1，软件写 1 清 0。

13.4.5 比较值配置寄存器（EPWM_CMP_CFG)

偏移地址：0x010
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC_CMP2_DIR	ADC_CMP2_EN	ADC_CMP1_DIR	ADC_CMP1_EN
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3_CMP2_DIR	CH3_CMP2_EN	CH3_CMP1_DIR	CH3_CMP1_EN	CH2_CMP2_DIR	CH2_CMP2_EN	CH2_CMP1_DIR	CH2_CMP1_EN	CH1_CMP2_DIR	CH1_CMP2_EN	CH1_CMP1_DIR	CH1_CMP1_EN	CH0_CMP2_DIR	CH0_CMP2_EN	CH0_CMP1_DIR	CH0_CMP1_EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description

31:20	Res.	保留，必须保持复位值。
19	ADC_CMP2_DIR	ADC 比较值 2 生效方向 (ADC compare value 2 direction) 0：在计数器增计数时生效 1：在计数器减计数时生效
18	ADC_CMP2_EN	ADC 比较值 2 使能 (ADC compare value 2 enable) 0：禁止 1：使能
17	ADC_CMP1_DIR	ADC 比较值 1 生效方向 (ADC compare value 2 direction) 0：在计数器增计数时生效 1：在计数器减计数时生效
16	ADC_CMP1_EN	ADC 比较值 1 使能 (ADC compare value 1 enable) 0：禁止 1：使能
15	CH3_CMP2_DIR	通道 3 比较值 2 生效方向 (Channel 3 compare value 2 direction) 0：在计数器增计数时生效 1：在计数器减计数时生效
14	CH3_CMP2_EN	通道 3 比较值 2 使能 (Channel 3 compare value 2 enable) 0：禁止 1：使能
13	CH3_CMP1_DIR	通道 3 比较值 1 生效方向 (Channel 3 compare value 1 direction) 0：在计数器增计数时生效 1：在计数器减计数时生效
12	CH3_CMP1_EN	通道 3 比较值 1 使能 (Channel 3 compare value 1 enable) 0：禁止 1：使能
11	CH2_CMP2_DIR	通道 2 比较值 2 生效方向 (Channel 2 compare value 2 direction) 0：在计数器增计数时生效 1：在计数器减计数时生效
10	CH2_CMP2_EN	通道 2 比较值 2 使能 (Channel 2 compare value 2 enable) 0：禁止 1：使能
9	CH2_CMP1_DIR	通道 2 比较值 1 生效方向 (Channel 2 compare value 1 direction) 0：在计数器增计数时生效 1：在计数器减计数时生效
8	CH2_CMP1_EN	通道 2 比较值 1 使能 (Channel 2 compare value 1 enable) 0：禁止 1：使能
7	CH1_CMP2_DIR	通道 1 比较值 2 生效方向 (Channel 1 compare value 2 direction) 0：在计数器增计数时生效 1：在计数器减计数时生效
6	CH1_CMP2_EN	通道 1 比较值 2 使能 (Channel 1 compare value 2 enable) 0：禁止 1：使能
5	CH1_CMP1_DIR	通道 1 比较值 1 生效方向 (Channel 1 compare value 1 direction) 0：在计数器增计数时生效 1：在计数器减计数时生效
4	CH1_CMP1_EN	通道 1 比较值 1 使能 (Channel 1 compare value 1 enable) 0：禁止 1：使能

3	CH0_CMP2_DIR	通道 0 比较值 2 生效方向 (Channel 0 compare value 2 direction) 0：在计数器增计数时生效 1：在计数器减计数时生效
2	CH0_CMP2_EN	通道 0 比较值 2 使能 (Channel 0 compare value 2 enable) 0：禁止 1：使能
1	CH0_CMP1_DIR	通道 0 比较值 1 生效方向 (Channel 0 compare value 1 direction) 0：在计数器增计数时生效 1：在计数器减计数时生效
0	CH0_CMP1_EN	通道 0 比较值 1 使能 (Channel 0 compare value 1 enable) 0：禁止 1：使能

13.4.6 ADC 触发比较值 1 配置寄存器（EPWM_ADC_CMP1）

偏移地址：0x014
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CMP_VAL	ADC 触发比较值 1 (ADC compare value 1)

13.4.7 ADC 触发比较值 2 配置寄存器（EPWM_ADC_CMP2）

偏移地址：0x018
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CMP_VAL	ADC 触发比较器数值 2 (ADC compare value 2)

13.4.8 通道 0 比较值 1 寄存器（EPWM_CH0_CMP1）

偏移地址：0x01C
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CMP_VAL	通道 0 比较值 1 (Channel 0 compare value 1)

13.4.9 通道 0 比较值 2 寄存器（EPWM_CH0_CMP2）

偏移地址：0x020
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CMP_VAL	通道 0 比较值 2 (Channel 0 compare value 2)

13.4.10 通道 1 比较值 1 寄存器（EPWM_CH1_CMP1）

偏移地址：0x024
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CMP_VAL	通道 1 比较值 1 (Channel 1 compare value 1)

13.4.11 通道 1 比较值 2 寄存器（EPWM_CH1_CMP2）

偏移地址：0x028
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CMP_VAL	通道 1 比较值 2 (Channel 1 compare value 2)

13.4.12 通道 2 比较值 1 寄存器（EPWM_CH2_CMP1）

偏移地址：0x02C
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CMP_VAL	通道 2 比较值 1 (Channel 2 compare value 1)

13.4.13 通道 2 比较值 2 寄存器（EPWM_CH2_CMP2）

偏移地址：0x030
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CMP_VAL	通道 2 比较值 2 (Channel 2 compare value 2)

13.4.14 通道 3 比较值 1 寄存器（EPWM_CH3_CMP1）

偏移地址：0x034
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CMP_VAL	通道 3 比较值 1 (Channel 3 compare value 1)

13.4.15 通道 3 比较值 2 寄存器（EPWM_CH3_CMP2）

偏移地址：0x038
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CMP_VAL	通道 3 比较值 2 (Channel 3 compare value 2)

13.4.16 更新事件控制寄存器（EPWM_UPDATE)

偏移地址：0x03C
复位值：0x00000000

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

Bit	Field	Description
31:1	Res.	保留，必须保持复位值。
0	UPDATE	软件写 1 更新所有具有影子寄存器的寄存器，硬件自动清 0；

13.4.17 通道控制寄存器（EPWM_CH_CR)

偏移地址：0x040
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3_PP	CH3_NP	CH2_PP	CH2_NP	CH1_PP	CH1_NP	CH0_PP	CH0_NP	Res.	Res.	Res.	Res.	CH3_INIT_O	CH2_INIT_O	CH1_INIT_O	CH0_INIT_O
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3_CMP2_ACT[1:0]		CH3_CMP1_ACT[1:0]		CH2_CMP2_ACT[1:0]		CH2_CMP1_ACT[1:0]		CH1_CMP2_ACT[1:0]		CH1_CMP1_ACT[1:0]		CH0_CMP2_ACT[1:0]		CH0_CMP1_ACT[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31	CH3_PP	CH3 P 通道输出极性 (Channel 3 p channel output polarity) 0：正常输出 1：取反输出
30	CH3_NP	CH3 N 通道输出极性 (Channel 3 n channel output polarity) 0：正常输出 1：取反输出
28	CH2_NP	CH2 N 通道输出极性 (Channel 2 n channel output polarity) 0：正常输出 1：取反输出
27	CH1_PP	CH1 P 通道输出极性 (Channel 1 p channel output polarity) 0：正常输出 1：取反输出
26	CH1_NP	CH1 N 通道输出极性 (Channel 1 n channel output polarity) 0：正常输出 1：取反输出
25	CH0_PP	CH0 P 通道输出极性 (Channel 0 p channel output polarity) 0：正常输出 1：取反输出
24	CH0_NP	CH0 N 通道输出极性 (Channel 0 n channel output polarity) 0：正常输出 1：取反输出
23:20	Res.	保留，必须保持复位值。
19	CH3_INIT_O	通道 3 初始输出状态 (Channel 3 initial output) 0：初始状态为 0 1：初始状态为 1
18	CH2_INIT_O	通道 2 初始输出状态 (Channel 2 initial output) ： 0：初始状态为 0 1：初始状态为 1
17	CH1_INIT_O	通道 1 初始输出状态 (Channel 1 initial output) ： 0：初始状态为 0 1：初始状态为 1
16	CH0_INIT_O	通道 0 初始输出状态 (Channel 0 initial output) ： 0：初始状态为 0 1：初始状态为 1
15:14	CH3_CMP2_ACT	通道 3 比较值 2 输出类型 (Channel 3 compare value 2 action) 00：计数值等于通道 3 比较值 2 输出 0 01：计数值等于通道 3 比较值 2 输出 1 10：计数值等于通道 3 比较值 2 输出翻转 11：计数值等于通道 3 比较值 2 输出不变

13:12	CH3_CMP1_ACT	通道 3 比较值 1 输出类型 (Channel 3 compare value 1 action) 00：计数值等于通道 3 比较值 1 输出 0 01：计数值等于通道 3 比较值 1 输出 1 10：计数值等于通道 3 比较值 1 输出翻转 11：计数值等于通道 3 比较值 1 输出不变
11:10	CH2_CMP2_ACT	通道 2 比较值 2 输出类型 (Channel 2 compare value 2 action) 00：计数值等于通道 2 比较值 2 输出 0 01：计数值等于通道 2 比较值 2 输出 1 10：计数值等于通道 2 比较值 2 输出翻转 11：计数值等于通道 2 比较值 2 输出不变
9:8	CH2_CMP1_ACT	通道 2 比较值 1 输出类型 (Channel 2 compare value 1 action) 00：计数值等于通道 2 比较值 1 输出 0 01：计数值等于通道 2 比较值 1 输出 1 10：计数值等于通道 2 比较值 1 输出翻转 11：计数值等于通道 2 比较值 1 输出不变
7:6	CH1_CMP2_ACT	通道 1 比较值 2 输出类型 (Channel 1 compare value 2 action) 00：计数值等于通道 1 比较值 2 输出 0 01：计数值等于通道 1 比较值 2 输出 1 10：计数值等于通道 1 比较值 2 输出翻转 11：计数值等于通道 1 比较值 2 输出不变
5:4	CH1_CMP1_ACT	通道 1 比较值 1 输出类型 (Channel 1 compare value 1 action) 00：计数值等于通道 1 比较值 1 输出 0 01：计数值等于通道 1 比较值 1 输出 1 10：计数值等于通道 1 比较值 1 输出翻转 11：计数值等于通道 1 比较值 1 输出不变
3:2	CH0_CMP2_ACT	通道 0 比较值 2 输出类型 (Channel 0 compare value 2 action) 00：计数值等于通道 0 比较值 2 输出 0 01：计数值等于通道 0 比较值 2 输出 1 10：计数值等于通道 0 比较值 2 输出翻转 11：计数值等于通道 0 比较值 2 输出不变
1:0	CH0_CMP1_ACT	通道 0 比较值 1 输出类型 (Channel 0 compare value 1 action) 00：计数值等于通道 0 比较值 1 输出 0 01：计数值等于通道 0 比较值 1 输出 1 10：计数值等于通道 0 比较值 1 输出翻转 11：计数值等于通道 0 比较值 1 输出不变

13.4.18 输出控制寄存器（EPWM_OUT_CR)

偏移地址：0x044
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3_P_FORCE_O	CH3_N_FORCE_O	CH3_P_FORCE_EN	CH3_N_FORCE_EN	CH2_P_FORCE_O	CH2_N_FORCE_O	CH2_P_FORCE_EN	CH2_N_FORCE_EN	CH1_P_FORCE_O	CH1_N_FORCE_O	CH1_P_FORCE_EN	CH1_N_FORCE_EN	CH0_P_FORCE_O	CH0_N_FORCE_O	CH0_P_FORCE_EN	CH0_N_FORCE_EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15	CH3_P_FORCE_O	通道 3 的 p 通道强制输出 (Channel 0 p force output) 0：强制输出低电平 1：强制输出高电平
14	CH3_N_FORCE_O	通道 3 的 n 通道强制输出 (Channel 0 n force output) 0：强制输出低电平 1：强制输出高电平
13	CH3_P_FORCE_EN	通道 3 的 p 通道强制输出使能 (Channel 0 p force output enable) 0：禁止 1：使能
12	CH3_N_FORCE_EN	通道 3 的 n 通道强制输出使能 (Channel 0 n force output enable) 0：禁止 1：使能
11	CH2_P_FORCE_O	通道 2 的 p 通道强制输出 (Channel 0 p force output) 0：强制输出低电平 1：强制输出高电平
10	CH2_N_FORCE_O	通道 2 的 n 通道强制输出 (Channel 0 n force output) 0：强制输出低电平 1：强制输出高电平
9	CH2_P_FORCE_EN	通道 2 的 p 通道强制输出使能 (Channel 0 p force output enable) 0：禁止 1：使能
8	CH2_N_FORCE_EN	通道 2 的 n 通道强制输出使能 (Channel 0 n force output enable) 0：禁止 1：使能
7	CH1_P_FORCE_O	通道 1 的 p 通道强制输出 (Channel 0 p force output) 0：强制输出低电平 1：强制输出高电平
6	CH1_N_FORCE_O	通道 1 的 n 通道强制输出 (Channel 0 n force output) 0：强制输出低电平 1：强制输出高电平
5	CH1_P_FORCE_EN	通道 1 的 p 通道强制输出使能 (Channel 0 p force output enable) 0：禁止 1：使能
4	CH1_N_FORCE_EN	通道 1 的 n 通道强制输出使能 (Channel 0 n force output enable) 0：禁止 1：使能
3	CH0_P_FORCE_O	通道 0 的 p 通道强制输出 (Channel 0 p force output) 0：强制输出低电平 1：强制输出高电平
2	CH0_N_FORCE_O	通道 0 的 n 通道强制输出 (Channel 0 n force output) 0：强制输出低电平 1：强制输出高电平
1	CH0_P_FORCE_EN	通道 0 的 p 通道强制输出使能 (Channel 0 p force output enable) 0：禁止 1：使能

0	CH0_N_FORCE_EN	通道 0 的 n 通道强制输出使能 (Channel 0 n force output enable) 0：禁止 1：使能

13.4.19 死区控制寄存器（EPWM_DT_CR）

偏移地址：0x048
复位值：0x00000000

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

Bit	Field	Description
31:10	Res.	保留，必须保持复位值。
9:0	DT_LEN	死区插入长度 (Dead zone time length) 当 DT_LEN==0 时相当于不插入死区

13.4.20 急停控制寄存器（EPWM_STOP_CR）

偏移地址：0x04C
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH3_P_STOP_O	CH3_N_STOP_O	CH2_P_STOP_O	CH2_N_STOP_O	CH1_P_STOP_O	CH1_N_STOP_O	CH0_P_STOP_O	CH0_N_STOP_O
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DBG_STOP_EN	ACMP_SEL[1:0]		ACMP_POL_SEL	ACMP_STOP_EN	EXT_POL_SEL	EXT_STOP_EN	Res.	Res.	Res.	Res.	FLT_LEN[1:0]		FLT_SAMPLE[1:0]	
	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
23	CH3_P_STOP_O	通道 3 的 p 通道急停状态输出 (Channel 3 p channel stop output) 0：急停状态输出为 0 1：急停状态输出为 1
22	CH3_N_STOP_O	通道 3 的 n 通道急停状态输出 (Channel 3 n channel stop output) 0：急停状态输出为 0 1：急停状态输出为 1
21	CH2_P_STOP_O	通道 2 的 p 通道急停状态输出 (Channel 2 p channel stop output) ： 0：急停状态输出为 0 1：急停状态输出为 1
20	CH2_N_STOP_O	通道 2 的 n 通道急停状态输出 (Channel 2 n channel stop output) 0：急停状态输出为 0 1：急停状态输出为 1
19	CH1_P_STOP_O	通道 1 的 p 通道急停状态输出 (Channel 1 p channel stop output) 0：急停状态输出为 0 1：急停状态输出为 1
18	CH1_N_STOP_O	通道 1 的 n 通道急停状态输出 (Channel 1 n channel stop output) 0：急停状态输出为 0 1：急停状态输出为 1
17	CH0_P_STOP_O	通道 0 的 p 通道急停状态输出 (Channel 0 p channel stop output) 0：急停状态输出为 0 1：急停状态输出为 1
16	CH0_N_STOP_O	通道 0 的 n 通道急停状态输出 (Channel 0 n channel stop output) 0：急停状态输出为 0 1：急停状态输出为 1
15	Res.	保留，必须保持复位值。
14	DBG_STOP_EN	在断点或单步调试时，I/O 会切换到急停模式输出电平 0：禁止 1：使能
13:12	ACMP_SEL	模拟比较器选择 (ACMP input source selection) 00：选择模拟比较器 ACMP0 输出 01：选择模拟比较器 ACMP1 输出 10：选择模拟比较器 ACMP2 输出 11：选择模拟比较器 ACMP3 输出
11	ACMP_POL_SEL	模拟比较值极性选择 (ACMP input polarity selection) 0：输入低电平有效 1：输入高电平有效
10	ACMP_STOP_EN	模拟比较器输出触发急停使能 (ACMP input stop enable) 0：禁止 1：使能

9	EXT_POL_SEL	外部信号输入极性选择（External input polarity selection） 0：输入低电平有效 1：输入高电平有效
8	EXT_STOP_EN	外部信号输入控制急停使能（External input signal control stop enable） 0：禁止　　1：使能
7:4	Res.	保留，必须保持复位值。
3:2	FLT_LEN	数字滤波宽度（Filter length selection） 00：保留值，禁止配置　　01：滤波长度为 8 10：滤波长度为 16　　11：滤波长度为 32
1:0	FLT_SAMPLE	数字滤波采样分频系数（Filter sample selection） 00：滤波时钟分频系数为 1 01：滤波时钟分频系数为 4 10：滤波时钟分频系数为 16 11：滤波时钟分频系数为 32

13.4.21　急停状态寄存器（EPWM_STOP_SR）

偏移地址：0x050
复位值：0x00000000

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

Bit	Field	Description
31:3	Res.	保留，必须保持复位值。
2	ACMP_TRIG	模拟比较器触发急停标记（ACMP stop control trigger） 0：模拟比较器未触发急停 1：模拟比较器触发急停，清除 STOP 时同时清除，无法软件写 1 清零。
1	EXT_TRIG	外部信号输入触发急停标记（External signal stop control trigger） 0：外部信号输入未触发急停 1：外部信号输入触发急停，清除 STOP 时同时清除，无法软件写 1 清零。

0	STOP	急停状态标记（Sop status flag） 0：EPWM 模块处于正常输出状态 1：EPWM 处于急停输出状态，可由硬件或软件置 1，硬件接收到硬件急停信号置 1。软件写 1 设置急停，写 0 恢复正常运行状态，解除急停状态。

13.4.22 锁定保护寄存器（EPWM_LOCK）

偏移地址：0x054
复位值：0x00000000

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

Bit	Field	Description
31:16	KEY	当写入 KEY == 0x900D 时，VAL 可以被更改，其余值无效，VAL 必须和 KEY 必须同时按 Word 整体写入，按 Half word，Byte 写入无效，KEY 读出值为 0；
15:1	Res.	保留，必须保持复位值。
0	VAL	锁定状态（Lock value） 0：解锁状态　　1：锁定保护状态 对 EPWM_LOCK 寄存器写 0x900D0001 时（必须，受保护寄存器会被锁定，无法写入，可以读取。在锁定状态下写受保护寄存器会触发 Hard Fault 异常。 对 EPWM_LOCK 寄存器写 0x900D0000 时（必须按 Word 整体写入，按 Half word，Byte 写入无效），解除锁定状态，所有寄存器值都可以被写入。 受锁定保护寄存器： EPWM_ARR；EPWM_CR；EPWM_CMP_CFG； EPWM_UPDATE；EPWM_CH_CR；EPWM_OUT_CR； EPWM_DT_CR；EPWM_STOP_CR；EPWM_STOP_SR。

14 霍尔和编码器接口控制器（POSIF）

14.1 简介

POSIF 可捕获编码器或霍尔传感器输入，实现硬件解析输入信号并自动计数，可用于电机控制中获取电机的工作状态。

14.2 主要特性

- 可配置为两种模式：编码器，霍尔传感器
- 可设置计数预分频器(1/2/4/8/16/32/64/128)
- 支持 3 路输入信号滤波，可选择模拟比较器输出作为输入
- 16 位编码器计数器，支持正交计数，方向计数
- 正交编码器支持 Z 相输入自动复位计数值
- 正交编码器支持配置最大计数值，减计数到 0，硬件自动调整为最大计数值
- 编码器支持 23 位输入脉冲计数器，可计数输入脉冲个数
- 23 位霍尔宽度计数器，硬件自动捕获输入边沿变化，可配置边沿变化产生中断
- 可配置霍尔宽度最大值，超过最大值可配置产生中断

14.3 功能说明

14.3.1 POSIF 结构框图

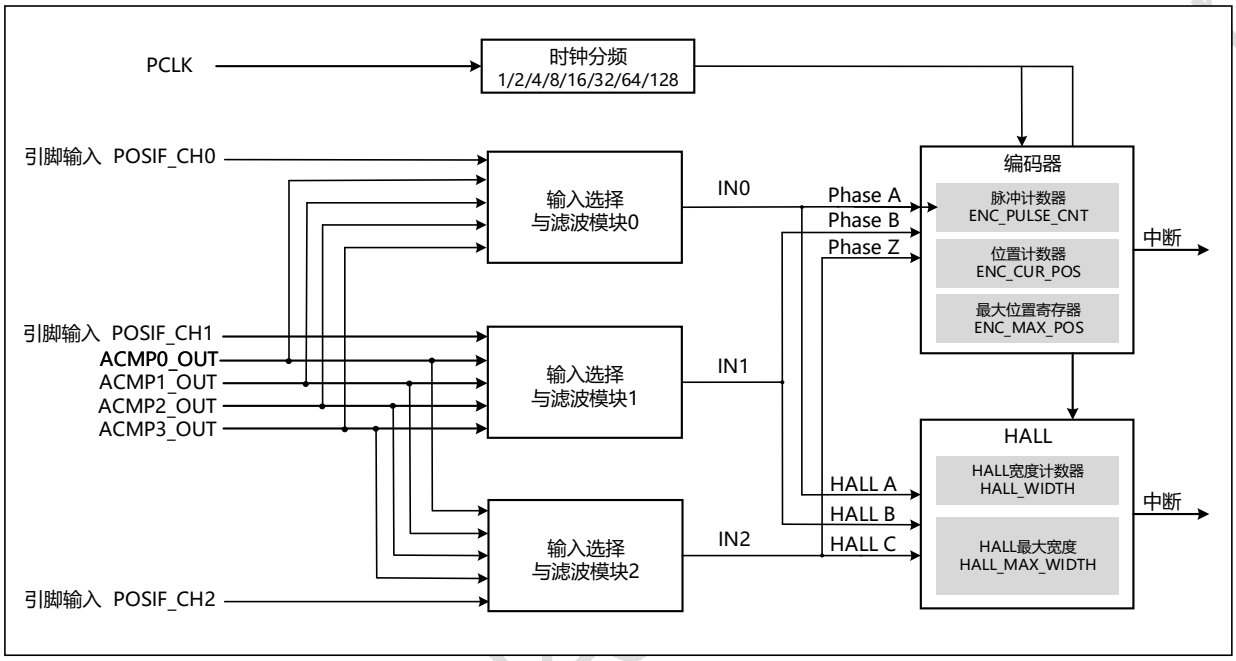

图 14-1 POSIF 结构框图

14.3.2 工作模式与输入选择

可配置 POSIF_CR.MODE 选择编码器模式或者霍尔模式，使用模块时需配置模块使能 POSIF_CR.EN 为 1。

可配置 POSIF_CR.CLK_DIV 确定模块的计数时钟频率，基于 PCLK 进行分频，分频系数为 1/2/4/8/16/32/64/128，编码器模式或者霍尔模式共用计数分频模块，所有计数器基于分频时钟进行计数。

模块支持三个输入信号：IN0，IN1，IN2，每个输入具有相同的逻辑结构，如图 14-2 所示。

输入信号支持选择从引脚输入，或模拟比较器的输出作为输入，配置 POSIF_CR.INx_SEL 选择输入源。选择为模拟比较器时，可通过 POSIF_CR.INx_ACMP_SEL 配置具体选择模拟比较器的源头。

模块内置滤波器可对输入信号进行滤波，可配置滤波长度寄存器(POSIF_CR.FLT_LEN)和滤波采样频率寄存器(POSIF_CR.FLT_SAMPLE)，选择滤波采样点数目和滤波采样频率。滤波器以固定采样频率采样输入电平，连续采样到多个采样点的相同电平时，输出该电平。滤波器的工作时钟为 PCLK。

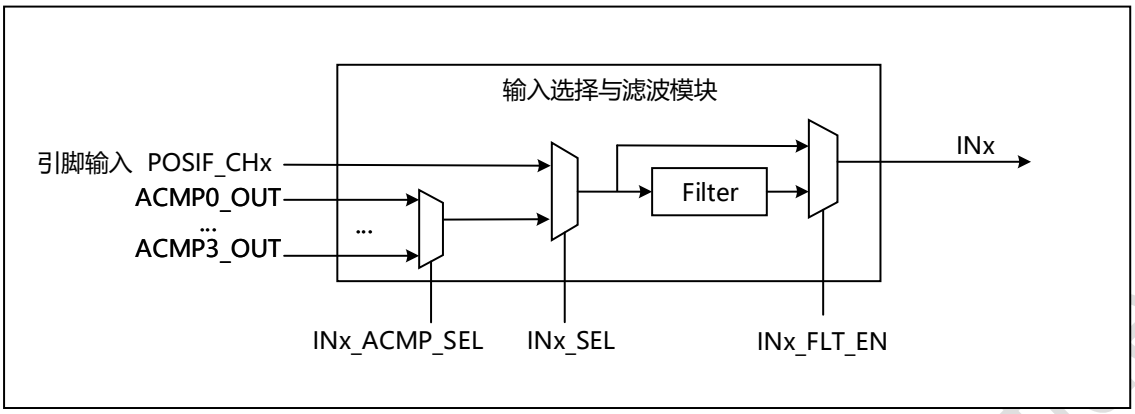

图 14-2 输入选择与滤波

14.3.3 编码器输入模式

编码器支持正交模式或者方向计数模式，可配置 ENC_CR.ENC_MODE 选择模式，可读取 ENC_CUR_POS 获取当前计数值，当前计数值表示当前位置。

当配置为正交编码器模式时，三个输入信号：IN0，IN1，IN2 分别对应编码器的 A 相，B 相，Z 相信号。正交编码器可配置在 IN0 输入变化，IN1 输入变化或两个输入任一变化时计数，不同模式下正交编码器计数规律如表 14-1 所示。

表 14-1 正交编码器计数模式

计数模式	相反信号的电平状态（IN0 变化计数对应 IN1 IN1 变化计数对应 IN0）	IN0 信号边沿变化		IN1 信号边沿变化	
		上升	下降	上升	下降
仅 IN0 变化计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅 IN1 变化计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 IN0 和 IN1 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

在正交编码器计数时，可配置为仅在 IN0 输入变化，仅在 IN1 输入变化，或两者变化时都计数。当 IN0 或 IN1 变化时，计数值的增加或减少需要参考相反信号，即 IN0 变化时，参考 IN1 电平状态增加或减少计数值，IN1 变化时，参考 IN0 电平状态增加或减少计数值。

正交编码器可配置 ENC_MAX_POS 设置最大计数值，当超过该值时，硬件将计数值清 0，当减计数到 0 时，下一个计数值变为最大计数值。可配置 ENC_CR.MAX_POS_IE 使能在计数值达到最大计数值时产生中断，中断标记位为 ENC_SR.MAX_POS_IF。

正交编码器每旋转一周会发一个脉冲，称之为零位脉冲或标识脉冲（即 Z 相信号），零位脉冲可用于决定零位置或标识位置。

可配置使能 Z 相输入复位计数器（ENC_CR.ZIN_EN），支持两种模式：单次触发，或每次收到 Z 输入脉冲都触发复位计数器，由 ENC_CR.ZIN_ACTION 配置决定。

图 14-3 示意了在使能 Z 相输入，且在 IN0，IN1 变化都计数时的场景，计数值 ENC_CUR_POS 可以标识当前编码器的状态。可以读取 ENC_SR.ENC_DIR 获取编码器的计数方向。

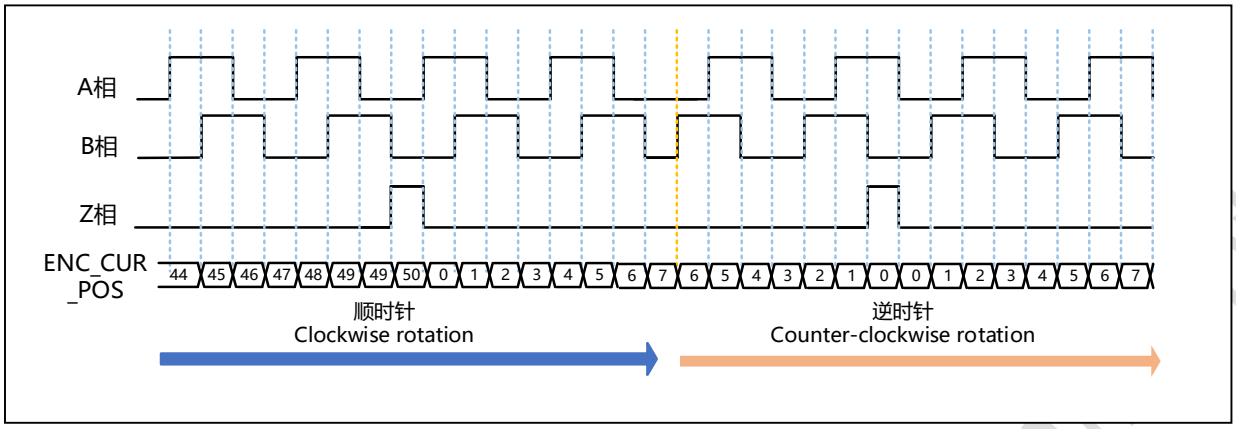

图 14-3 正交编码器计数示意

编码器配置为方向计数模式时，当 IN1 输入为方向控制，IN0 输入每来一个脉冲时，进行一次计数，由方向决定是增计数或减计数，如图 14-4 所示。

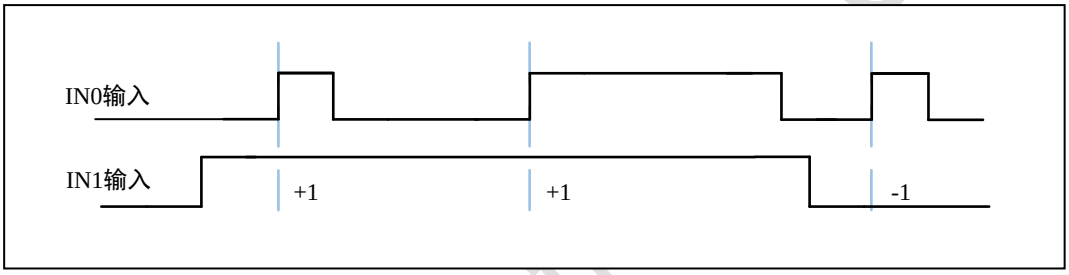

图 14-4 方向编码器计数示意

内嵌脉冲计数模块，**可计数 IN0 输入的脉冲个数**，用于 M 法测速。配置 ENC_CR.PULSE_CNT_EN 使能脉冲计数模块，读取 ENC_PULSE_CNT 寄存器获取脉冲计数值，使能时硬件会自动清除该计数值。配置 ENC_CR.PULSE_CNT_OF_IE 使能脉冲计数器溢出中断，中断标记位为 ENC_SR.PULSE_CNT_OF_IF。

利用 CCT0 模块可进行 T 法测速，两路输入 POSIF_CH0，POSIF_CH1 可作为 CCT0 模块的捕获输入。

14.3.4 霍尔输入模式

模块输入 IN0，IN1，IN2 对应霍尔信号的 ABC 三相，也可以设置选择 ACMP 模拟比较器作为输入，用于无感电机控制。

当三个输入霍尔信号任意一个发生变化时，HALL 状态（POSIF_HALL_SR.DATA）将会经过设定延时(POSIF_HALL_DELAY)之后被更新，如图 14-5 所示，也可以获取没有经过滤波的原始 HALL 值(POSIF_HALL_SR.RAW)。

当 POSIF_HALL_DELAY 配置为 0 时，若 HALL 状态变化，状态寄存器将被立即更新。

与此同时，硬件也将更新 POSIF_HALL_WIDTH 寄存器，POSIF_HALL_WIDTH 用于计数相邻两个输入边沿之间的值，由硬件自动更新，可用于检测电机转速。硬件支持设置最大宽度计数值(POSIF_HALL_MAX_WIDTH)，当超过设定的最大宽度之后，若使能 POSIF_HALL_CR.MAX_WIDTH_IE 则会产生溢出中断，中断标记位为 POSIF_HALL_SR. MAX_WIDTH_IF，最大宽度计数值可用于监测电机转速是否过低。

可设置在 HALL 状态被更新时产生中断（POSIF_HALL_CR.UPDATE_IE），中断标记位为 POSIF_HALL_SR.UPDATE_IF。

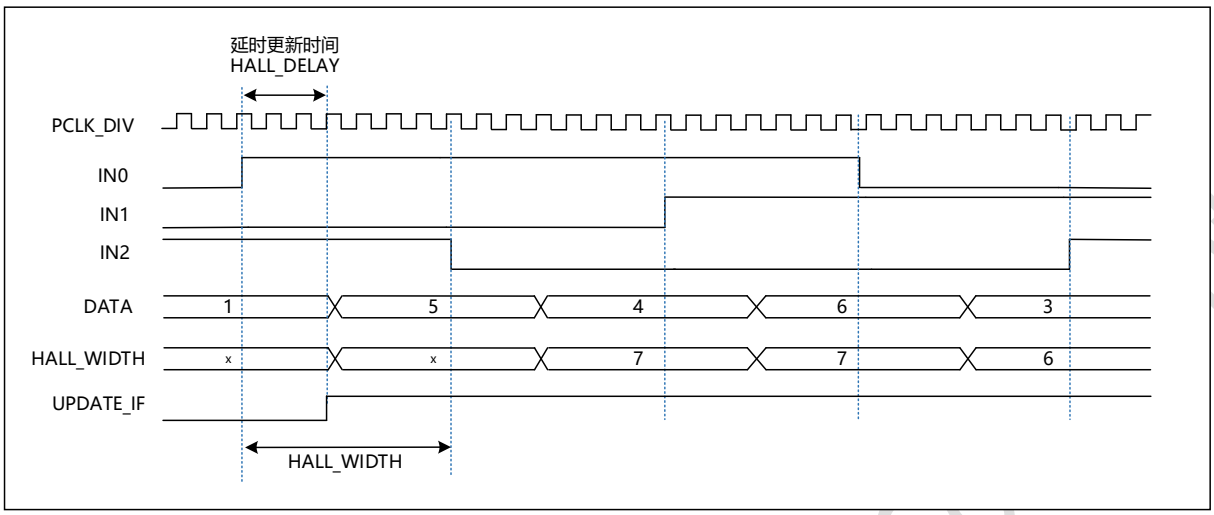

图 14-5 霍尔输入模式示意

14.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 14-2 POSIF 寄存器概览

偏移地址	寄存器名	寄存器描述	复位值
0x000	POSIF_CR	POSIF控制寄存器	0x00000000
0x004	POSIF_ENC_CR	POSIF编码器控制寄存器	0x00000000
0x008	POSIF_ENC_SR	POSIF编码器状态寄存器	0x00000000
0x00C	POSIF_MAX_POS	POSIF编码器最大计数值寄存器	0x00000000
0x010	POSIF_CUR_POS	POSIF编码器当前位置寄存器	0x00000000
0x014	POSIF_PULSE_CNT	POSIF编码器脉冲计数寄存器	0x00000000
0x018	POSIF_HALL_CR	POSIF霍尔控制寄存器	0x00000000
0x01C	POSIF_HALL_SR	POSIF霍尔状态寄存器	0x00000000
0x020	POSIF_HALL_WIDTH	POSIF霍尔宽度计数器寄存器	0x00000000
0x024	POSIF_HALL_MAX_WIDTH	POSIF霍尔最大宽度计数值寄存器	0x00000000
0x028	POSIF_HALL_DELAY	POSIF霍尔状态延时更新时间寄存器	0x00000000

14.4.1 控制寄存器（POSIF_CR）

偏移地址：0x000
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	IN2_FLT_LEN[1:0]		IN2_FLT_SAMPLE[1:0]		IN2_ACMP_SEL[1:0]		IN2_SEL	Res.	IN1_FLT_LEN[1:0]		IN1_FLT_SAMPLE[1:0]		IN1_ACMP_SEL[1:0]		IN1_SEL
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	IN0_FLT_LEN[1:0]		IN0_FLT_SAMPLE[1:0]		IN0_ACMP_SEL[1:0]		IN0_SEL	Res.	Res.	Res.	CLK_DIV[2:0]			MODE	EN
	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

Bit	Field	Description
31	Res.	保留，必须保持复位值。
30:29	IN2_FLT_LEN	输入 2 滤波长度 (Input 2 filter length selection) 00：保留值，禁止配置； 01：输入 2 滤波长度为 8； 10：输入 2 滤波长度为 16； 11：输入 2 滤波长度为 32；
28:27	IN2_FLT_SAMPLE	输入 2 滤波时钟分频系数选择 (Channel 2 filter clock sample selection) 00：通道 2 滤波时钟分频系数为 1； 01：通道 2 滤波时钟分频系数为 4； 10：通道 2 滤波时钟分频系数为 16； 11：通道 2 滤波时钟分频系数为 32；
26:25	IN2_ACMP_SEL	输入 2 选择模拟比较器输入源头 (Input 2 ACMP source selection) 00：ACMP0； 01：ACMP1； 10：ACMP2； 11：ACMP3。
24	IN2_SEL	输入 2 选择信号输入 (Input 2 source selection) 0：选择从引脚输入； 1：选择从模拟比较器输入。
23	Res.	保留，必须保持复位值。
22:21	IN1_FLT_LEN	输入 1 滤波长度 (Input 1 filter length selection) 00：保留值，禁止配置； 01：输入 1 滤波长度为 8； 10：输入 1 滤波长度为 16； 11：输入 1 滤波长度为 32；
20:19	IN1_FLT_SAMPLE	输入 1 滤波时钟分频系数选择 (Channel 1 filter clock sample selection) 00：通道 1 滤波时钟分频系数为 1； 01：通道 1 滤波时钟分频系数为 4； 10：通道 1 滤波时钟分频系数为 16； 11：通道 1 滤波时钟分频系数为 32；
18:17	IN1_ACMP_SEL	输入 1 选择模拟比较器源头 (Input 1 ACMP source selection) 00：ACMP0； 01：ACMP1； 10：ACMP2； 11：ACMP3。
16	IN1_SEL	输入 1 选择信号输入 (Input 1 source selection) 0：选择从引脚输入； 1：选择从模拟比较器输入。
15	Res.	保留，必须保持复位值。

14:13	IN0_FLT_LEN	输入 0 滤波长度（Input 1 filter length selection） 00：保留值，禁止配置； 01：输入 0 滤波长度为 8； 10：输入 0 滤波长度为 16； 11：输入 0 滤波长度为 32；
12:11	IN0_FLT_SAMPLE	输入 0 滤波时钟分频系数选择（Channel 1 filter clock sample selection） 00：通道 0 滤波时钟分频系数为 1； 01：通道 0 滤波时钟分频系数为 4； 10：通道 0 滤波时钟分频系数为 16； 11：通道 0 滤波时钟分频系数为 32；
10:9	IN0_ACMP_SEL	输入 0 选择模拟比较器源头（Input 0 ACMP source selection） 00：ACMP0； 01：ACMP1； 10：ACMP2； 11：ACMP3。
8	IN0_SEL	输入 0 选择信号输入（Input 0 source selection） 0：选择从引脚输入； 1：选择从模拟比较器输入。
7:5	Res.	保留，必须保持复位值。
4:2	CLK_DIV	时钟分频系数（Clock division） 000：1 分频； 001：2 分频； 010：4 分频； 011：8 分频； 100：16 分频；101：32 分频； 110：64 分频；111：128 分频。
1	MODE	POSIF 模式选择： 0：霍尔（HALL，Hall mode）； 1：编码器（ECD，Encoder Mode）。
0	EN	模块器使能位： 0：禁止计数器； 1：使能计数器。

14.4.2 编码器控制寄存器（POSIF_ENC_CR）

偏移地址：0x004
复位值：0x00000000

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

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7	PULSE_CNT_OF_IE	脉冲控制溢出中断使能（Pulse control overflow interrupt enable) 0：禁止； 1：使能。
6	MAX_POS_IE	最大位置中断使能（Max position interrupt enable) 0：禁止中断使能； 1：计数器 ENC_CUR_POS 达到 MAX_POS 时，产生中断使能。
5	ZIN_ACTION	Z 相输入动作选择（Z-phase input action selection) 0：仅复位位置计数器一次； 1：多次复位 ENC_CUR_POS。
4	ZIN_EN	Z 相输入使能（Z-phase input function enable) 0：不复位计数器 ENC_CUR_POS； 1： ZIN 输入可以对 ENC_CUR_POS 清 0。
3	PULSE_CNT_EN	脉冲计数使能（Pulse counter enable) 0：停止计数； 1：使能脉冲计数，软件写 1 时硬件将 PULSE_CNT 清 0，并启动计数。
2:0	ENC_MODE	编码器模式选择（Encoder mode selection) 000：正交编码器仅在 IN0 边沿计数 001：正交编码器仅在 IN1 边沿计数； 010：正交编码器在 IN0 和 IN1 都边沿计数； 011：保留； 100：方向计数模式 IN0 为方向控制； 101：方向计数模式 IN1 为方向控制； 110：保留； 111：保留。

14.4.3 编码器状态寄存器（POSIF_ENC_SR)

偏移地址：0x008
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC_DIR	PULSE_CNT_OF_IF	MAX_POS_IF
													ro	w1c	w1c

Bit	Field	Description
31:3	Res.	保留，必须保持复位值。
2	ENC_DIR	编码器方向标志位（Encoder direction） 0：反转； 1：正转。
1	PULSE_CNT_OF_IF	脉冲控制溢出中断标志位（Pulse control overflow interrupt flag） 0：未溢出； 1：溢出。
0	MAX_POS_IF	最大位置计数值中断标志（Max position interrupt flag） 0：未到达最大位置计数值； 1：到达最大位置计数值。

14.4.4 编码器最大位置计数值寄存器（POSIF_ENC_MAX_POS）

偏移地址：0x00C
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	MAX_POS	编码器最大计数值(Encoder max counter value) 编码器计数到最大计数值后，下次增计数为 0。当减计数到 0 时，下次减计数值为 MAX_POS。

14.4.5 编码器当前位置计数值寄存器（POSIF_ENC_CUR_POS）

偏移地址：0x010
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	CUR_POS	编码器当前计数值。

14.4.6 编码器脉冲计数寄存器（POSIF_ENC_PULSE_CNT）

偏移地址：0x014
复位值：0x00000000

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

Bit	Field	Description
31:24	Res.	保留，必须保持复位值。
23:0	CNT	捕获脉冲的计数器计数值，溢出之后硬件自动归零，启动时从 0 开始计数。

14.4.7 霍尔控制寄存器（POSIF_HALL_CR）

偏移地址：0x018
复位值：0x00000000

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

Bit	Field	Description
31:2	Res.	保留，必须保持复位值。
1	MAX_WIDTH_IE	最大宽度中断使能（Max width interrupt enable) 0：禁止； 1：HALL_WIDTH 达到 MAX_WIDTH 时产生中断使能。
0	UPDATE_IE	更新 HALL 状态中断使能（Update interrupt enable）： 0：禁止； 1；使能。

14.4.8 霍尔状态寄存器（POSIF_HALL_SR)

偏移地址：0x01C
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UPDATE_IF	MAX_WIDTH_IF	RAW[2:0]			DATA[2:0]		
								w1c	w1c	ro	ro	ro	ro	ro	ro

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7	UPDATE_IF	更新中断标志（Update interrupt flag) 0：未更新； 1：HALL 状态更新标志。

6	MAX_WIDTH_IF	最大宽度中断标志 (Max width interrupt flag) 0：未更新； 1：HALL_WIDTH 达到最大值 MAX_WIDTH 标志。
5:3	RAW	未滤 HALL 输入状态波值(Raw)。
2:0	DATA	经滤波之后 HALL 输入状态值(Data)。

14.4.9 霍尔宽度计数器寄存器（POSIF_HALL_WIDTH）

偏移地址：0x020
复位值：0x00000000

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

Bit	Field	Description
31:24	Res.	保留，必须保持复位值。
23:0	HALL_WIDTH	HALL 三相输入相邻两个输入边沿之间的计数值，会在每个时钟周期更新计数值。当超过设定的最大宽度 HALL_MAX_WIDTH 之后，仍会继续计数，更新计数值。

14.4.10 霍尔最大宽度计数值寄存器（POSIF_HALL_MAX_WIDTH）

偏移地址：0x024
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
23:0	MAX_WIDTH	HALL 计数最大宽度值。

14.4.11 霍尔状态延时更新时间寄存器（POSIF_HALL_DELAY）

偏移地址：0x028
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，必须保持复位值。
15:0	VAL	当 HALL 输入边沿变化后，延时 HALL_DELAY 计数时间后，更新 HALL 状态。

15 看门狗（WDG）

15.1 简介

看门狗外设，具有安全性高、定时准确及使用灵活的优点。
看门狗外设可检测并解决由软件错误导致的故障，并在计数器达到给定的超时值时触发系统复位。

15.2 主要特性

- 22 位内部计数器，当计数达到设定时间后产生复位信号。
- 在未达到计数周期之前，可以通过喂狗操作来复位计数器，对 WDG_CLR 寄存器写 0x900DBEEF 实现喂狗操作。

15.3 功能说明

在系统复位后，看门狗总是处于关闭状态，MCU 运行过程中可以通过停止或者开启看门狗。
共有 16 种计数器溢出周期可选(WDG_CSR.MODE)，计数最长时间 128s，精度为 1/32.768k=0.03ms。
只有在看门狗禁止（WDG_CSR.EN=0）时，才可以切换溢出周期，在看门狗使能时无法切换。
看门狗超时后复位信号会送到 Reset 模块中，实现系统复位功能。
可选择设置使能中断（WDG_CSR.IE），在剩余溢出时间还有 1.9ms 时会产生中断。将 WDG_CSR. EN 写 0，会清除中断标志位 WDG_CSR. IF；喂狗操作也会清除 WDG_CSR. IF。
MCU 处于低功耗，睡眠模式时 WDG 不会停止，处于 Debug 模式会暂停计数。

15.4 寄存器概述

如无特殊说明，下列寄存器均可支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 15-1 WDG 寄存器概述

偏移地址	寄存器名	寄存器描述	复位值
0x000	WDG_CSR	WDG 控制状态寄存器	0x00000000
0x004	WDG_CLR	WDG 复位寄存器	0x00000000

15.4.1 WDG 控制状态寄存器（WDG_CSR）

偏移地址：0x000
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M_VAL[15:0]															
wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[3:0]				Res.	IF	IE	EN
								rw	rw	rw	rw		w1c	rw	rw

Bit	Field	Description
31:16	M_VAL	密码值（Magic value）： 高 16 比特为密码位，只写，读出为 0。 当写入高 16 位为 900d 时，配置为有效，其余情况下配置无效。
15:8	Res.	保留，必须保持复位值。
7:4	MODE	选择看门狗计数器溢出的时间间隔 16 种。 0000：计数器实际使用位数 7，最大计数值 128，计数时间 0.00390625； 0001：计数器实际使用位数 8，最大计数值 256 ，计数时间 0.0078125 0010：计数器实际使用位数 9，最大计数值 512 ，计数时间 0.015625 0011：计数器实际使用位数 10，最大计数值 1024，计数时间 0.03125 0100：计数器实际使用位数 11，最大计数值 2048，计数时间 0.0625 0101：计数器实际使用位数 12，最大计数值 4096，计数时间 0.125 0110：计数器实际使用位数 13，最大计数值 8192，计数时间 0.25 0111：计数器实际使用位数 14，最大计数值 16384，计数时间 0.5 1000：计数器实际使用位数 15，最大计数值 32768，计数时间 1

		1001：计数器实际使用位数 16，最大计数值 65536，计数时间 2 1010：计数器实际使用位数 17，最大计数值 131072，计数时间 4 1011：计数器实际使用位数 18，最大计数值 262144，计数时间 8 1100：计数器实际使用位数 19，最大计数值 524288，计数时间 16 1101：计数器实际使用位数 20，最大计数值 1048576，计数时间 32 1110：计数器实际使用位数 21，最大计数值 2097152，计数时间 64 1111：计数器实际使用位数 22，最大计数值 4194304，计数时间 128
3	Res.	保留，必须保持复位值。
2	IF	中断标志位（Interrupt flag）： 写 1 清 0。
1	IE	中断使能（Interrupt enable）： 在看门狗剩余溢出时间还有 1.9ms 时，若使能中断，则会在对应的时间触发中断请求。
0	EN	WDT 使能位： 0：禁止看门狗； 1：使能看门狗。

15.4.2 WDG 复位寄存器（WDG_CLR）

偏移地址：0x004
复位值：0x00000000

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

Bit	Field	Description
31:0	CLR_WDG	复位看门狗，软件写 900DBEEF 时，配置为有效，进行复位。 其余配置无法复位看门狗。只写寄存器，读回为 0。

16 低功耗定时器（LPTIM）

16.1 简介

LPTIM 包含 32 位向上计数器，使用 32K 低频时钟作为计数时钟，可以周期性触发中断。低功耗模式下，可作为 MCU 唤醒源。

16.2 主要特性

LPTIM 包含以下主要功能

- 周期性自动唤醒

16.3 功能模块说明

16.3.1 周期性自动唤醒

LPTIM 包含周期性自动唤醒功能，可以周期性触发中断唤醒 MCU。
32 位内部计数器计数时钟：LSCLK 时钟。
使能 LPTIM 前需要配置唤醒周期(LPTIM_WAKEUP_PERIOD)，以及配置中断使能(LPTIM_CR.WK_IE)。
在使能 LPTIM 后，无法重新配置唤醒周期和中断使能。
在配置使能（LPTIM_CR.WK_EN=1）后，计数器开始从 0 计数，当计数值等于唤醒周期值（LPTIM_WAKEUP_CNTR == LPTIM_WAKEUP_PERIOD）时，计数器将自动清零，重新开始周期性重复计数。
若配置中断使能（LPTIM_CR.WK_IE=1），则可在计数值等于唤醒周期值时产生中断，中断标记为 LPTIM_SR.WK_IF。软件可写 1 清除中断标记。
在计数过程中，软件可通过 LPTIM_WAKEUP_CNTR 来实时获取计数值。
注：
1. 在软件配置使能计数器时，由于内部时钟同步的原因，实际需要约 2 个 LSCLK 时钟周期后，LPTIM_CR.WK_EN 才能被硬件置 1，计数器开始计数。在时钟同步期间，软件无法中止计数器开启过程。
2. 唤醒周期（LPTIM_WAKEUP_PERIOD）最小值为 3。若配置值小于 3，那最小唤醒周期仍为 3。

16.3.2 低功耗特性

使能 LPTIM 计数后，MCU 进入不同低功耗模式场景如下：
1.在进入睡眠(SLEEP) 或 深度睡眠(DEEP SLEEP) 模式时，LSCLK 时钟不会被关闭，LPTIM 计数器会持续计数。
2.在进入停止(STOP)模式时，若 RCC_STOP_CR. LSCLK_EN 配置为 1，则 LSCLK 时钟不会被关闭，

LPTIM 计数器会持续计数，可产生中断。

若 RCC_STOP_CR. LSCLK_EN 配置为 0，则 LSCLK 时钟会被关闭，并且 LPTIM 计数器会被清 0。在退出停止模式后，LSCLK 时钟会被自动恢复，LPTIM 计数器从 0 开始自动计数。

16.4 低功耗定时器寄存器概述

寄存器可支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 16-1 低功耗定时器寄存器描述

偏移地址	寄存器名	寄存器描述	复位值
0x000	LPTIM_CR	LPTIM控制配置寄存器	0x00000000
0x004	LPTIM_SR	LPTIM状态配置寄存器	0x00000000
0x008	LPTIM_WAKEUP_CNTR	LPTIM计数值寄存器	0x00000000
0x00c	LPTIM_WAKEUP_PERIOD	LPTIM唤醒计数值配置寄存器	0x00000000

16.4.1 LPTIM 控制配置寄存器（LPTIM_CR）

偏移地址：0x000
复位值：0x00000000

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

Bit	Field	Description
31:9	Res.	保留，必须保持复位值。
8	WK_IE	唤醒中断使能（Wakeup interrupt flag enable）。必须在模块使能前配置。 0：不使能唤醒中断标志； 1：使能唤醒中断标志。
0	WK_EN	定时器使能位。 0：禁止计数器； 1：使能计数器。

16.4.2 LPTIM 状态配置寄存器（LPTIM_SR）

偏移地址：0x004
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WK_IF
															w1c

Bit	Field	Description
31:1	Res.	保留，必须保持复位值。
0	WK_IF	唤醒中断标记（Wakeup interrupt flag）： 由硬件置 1；软件写 1 清 0。

16.4.3 LPTIM 计数值寄存器（LPTIM_WAKEUP_CNTR）

偏移地址：0x008
复位值：0x00000000

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

Bit	Field	Description
31:0	WK_CNT	当前唤醒计数器值（Wakeup counter value）

16.4.4 LPTIM 唤醒计数值配置寄存器（LPTIM_WAKEUP_PERIOD）

偏移地址：0x00C
复位值：0x00000000

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

Bit	Field	Description
31:0	WK_P	周期性自动唤醒的周期。 必须在模块使能前配置。最小值为 3，若配置值小于 3，那最小唤醒周期仍为 3。

17 循环冗余校验计算单元（CRC）

17.1 简介

循环冗余校验(CRC)计算单元，根据特定多项式和 8、16、32 位数据计算 CRC 校验码，被广泛运用于校验数据传输或数据存储的完整性。

17.2 主要特性

- 可选 CRC-32 多项式：0x4C11DB7
  $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$
- 可选 CRC-16 多项式：0x1021 $X^{16}+X^{12}+X^{5}+1$
- 可选 CRC-16 多项式：0x8005 $X^{16}+X^{15}+X^{2}+1$
- 支持 8、16、32 位输入数据位宽
- 使用 4 个 AHB 时钟周期完成 32 位数据的 CRC 计算
- 输入数据、输出结果可反转
- 可配置的 CRC 初值

17.3 功能说明

17.3.1 CRC 计算单元框图

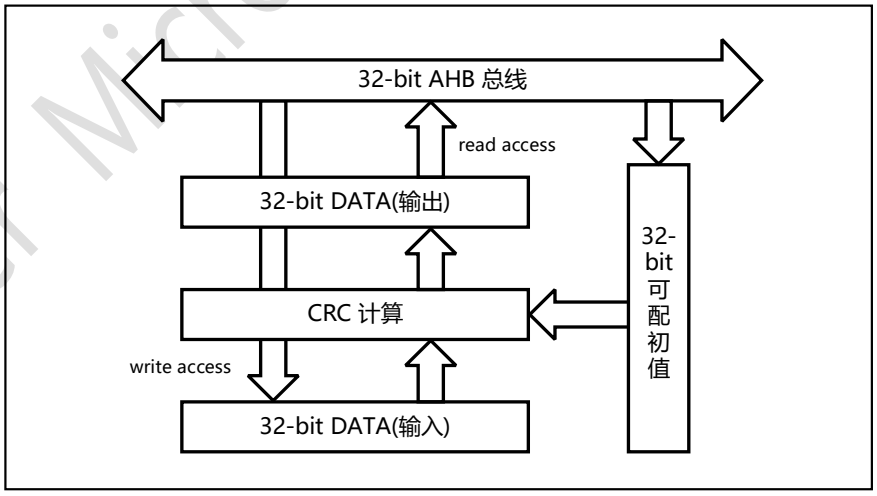

图 17-1

17.3.2 CRC 操作

CRC 计算单元支持三种多项式，配置模式(CRC_CR.MODE)以选定 CRC 多项式。
该单元含有数据寄存器(CRC_DATA)，对该寄存器的每次写操作，会触发一次初值和数据的 CRC 计算，计算结果从结果寄存器(CRC_RESULT)返回。
该单元含有初值寄存器(CRC_INIT)，该值在计算前可被配置为任意值，并在 CRC 计算完成时更新，用于下次的 CRC 计算。
数据寄存器(CRC_DATA)支持写入右对齐的字、半字、字节，因此特定字长的数据可动态调整写入方式，以最小化写访问的次数。例如，对 5 个字节的 CRC 计算，可调整为 1 个字和 1 个字节的两次写入。
支持输入数据按位反转，根据反转输入寄存器(CRC_CR.REV_IN)的值对数据按 8、16 和 32 的位宽按位反转。
例如：输入数据 0x1A2B3C4D 进行 CRC 计算，反转结果：
字节反转 0x58D43CB2；半字反转 0xD458B23C；字反转 0xB23CD458;
支持输出结果按位反转，通过配置反转输出(CRC_CR.REV_OUT)寄存器实现。例如：输出结果 0x11223344，反转结果 0x22CC4488。
输出结果支持按位取反，配置异或输出寄存器 CRC_CR.XOR 为 1，则 CRC 运算结果按位异或全 1 后再输出结果。
注：结果寄存器的值关联初值寄存器的值，在计算完成后，应先读取 CRC_RESULT 的值。若先改变 CRC_INIT 值，则 CRC_RESIULT 值也会改变。

17.4 寄存器概述

有关寄存器说明中使用的缩写，请参见通用寄存器概述。
CRC 寄存器可支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 17-1 CRC 寄存器概述

偏移地址	寄存器名	寄存器描述	复位值
0x000	CRC_CR	CRC控制寄存器	0x00000000
0x004	CRC_DATA	CRC数据寄存器	0x00000000
0x008	CRC_INIT	CRC初始值寄存器	0xFFFFFFFF
0x00C	CRC_RESULT	CRC结果值寄存器	0xFFFFFFFF

17.4.1 CRC 控制寄存器（CRC_CR）

偏移地址：0x000
复位值：0x00000000

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

Bit	Field	Description
31:6	Res.	保留，必须保持复位值。
5	XOR_OUT	异或输出（XOR Output） 0：输出值异或 0x00000000； 1：输出值异或 0xFFFFFFFF；
4	REV_OUT	反转输出（Reverse Output） 0：输出不反转； 1：输出反转。
3:2	REV_IN	反转输入（Reverse Input） 00：输入不反转； 01：输入按字节反转； 10：输入按半字反转； 11：输入按字反转。
1:0	MODE	模式选择（MODE） 00：CRC-32 0x4C11DB7； 01：CRC-16 0x1021； 10：CRC-16 0x8005。

17.4.2 CRC 数据寄存器（CRC_DATA）

偏移地址：0x004
复位值：0x00000000

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

Bit	Field	Description
31:0	DATA	CRC 输入数据(DATA)

17.4.3 CRC 初值值寄存器（CRC_INIT）

偏移地址：0x008
复位值：0xFFFFFFFF

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

Bit	Field	Description
31:0	INIT	CRC 初始值(Initial value)

17.4.4 CRC 结果值寄存器（CRC_RESULT）

偏移地址：0x00C
复位值：：0xFFFFFFFF

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

Bit	Field	Description
31:0	RESULT	CRC 计算结果（Result）

18 I2C

18.1 简介

I2C 接口硬件实现了 I2C 的时序序列、传输协议、仲裁判定和事件中断等功能，提供了多主机通信能力。该 IP 可在多主机 I2C 总线中用作主机或从机，能够实现标准传输模式、快速传输模式和超快速传输模式。I2C 使能期间，SCL、SDA 对应的 GPIO 应配置成开漏模式，并通过外部电阻上拉。

18.2 主要特性

- 主或从模式选择
- 支持标准模式（高达 100 Kbit/s）
- 支持快速模式（高达 400 Kbit/s）
- 支持超快速模式（高达 1 Mbit/s）
- 支持 7 位和 10 位寻址模式
- 提供广播呼叫功能
- 可编程的 SDA 数据建立保持时间
- 支持总线事件管理
- 支持多主机通信
- 支持 SCL 下拉调控通信速率
- 支持对输入信号的数字滤波
- 支持通信超时监测
- 收发 1 字节缓冲
- 支持通过 DMA 收发数据

18.3 I2C 协议

18.3.1 起始和停止时序

I2C 通信以 START 事件发起，并且以 STOP 事件结束。时序如下图所示。

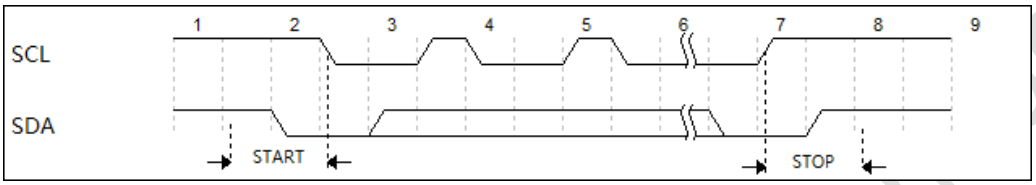

图 18-1 总线启动和停止事件时序

I2C 总线空闲时，SCL 和 SDA 都保持为高电平，当 SCL 为高电平且 SDA 由高电平到低电平转换时为 START 事件，当 SCL 为高电平且 SDA 由低电平到高电平转换时为 STOP 事件，START 和 STOP 事件由主机产生。

18.3.2 地址协议

I2C 具有两种寻址方式，7 位寻址和 10 位寻址，两种方式的寻址方式有所区别。
7 位地址寻址格式如下图所示，7 位地址在 START 事件后开始发起（从地址高位开始连续传输至地址最低位），主机发起写/读位，以及由从机回应的 ACK/NACK 响应位。

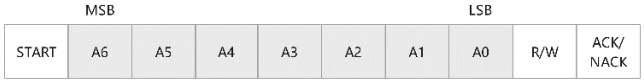

图 18-2 7 位地址读写寻址格式

主机向从机发起 10 位地址写操作寻址，如下图所示，主机发起 START 之后的第一个字节中，最后一位为 0，表示写方向，若从设备回应 ACK，则之后发送 10 位地址的低 8 位地址值。

图 18-3 10 位地址写寻址格式

主机向从机发起 10 位地址读操作寻址如下图所示，主机发起 START 并完成两个字节地址（写方向）发送后，再发起 RESTART 事件，并在第一个字节的最后一位转换为读方向，方向位值为 1，若从设备回应 ACK 则完成了 10 位地址读操作寻址。

图 18-4 10 位地址读寻址格式

18.3.3 ACK 和 NACK 标志

I2C 通信过程中，每个字节数据传输后都有 ACK/NACK 响应位，主机必须在完成字节数据传输后生成第 9 个 SCL 钟来确保响应产生。
总线上的 I2C 设备发出 NACK 响应时，SDA 则会在第 9 个 SCL 高电平期间保持高电平，主机则可以生成 STOP 事件终止通信，或者生成 RESTART 事件来发起新的通信。
以下有三种情况会导致 NACK 的产生：

- 主机发起的地址在 I2C 总线上没有对应的设备。
- 设备正忙不响应。
- 在正常的数据通信过程中，接收设备满足特定状态下在第 9 个 SCL 回应 NACK。

18.3.4 数据传输时序

主机设备在 START 事件后开始传输地址和数据，其中地址数据位由主机向从设备发起，而数据位则根据 R/W 读写的方向，也由主机发送，通讯完成后，主机发起 STOP 结束数据的通信，如下图所示。

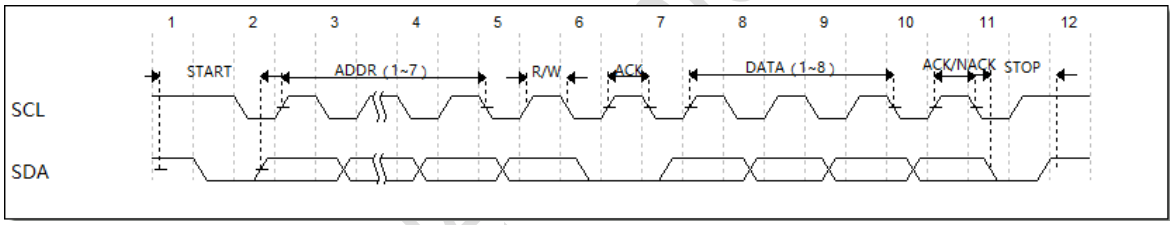

图 18-5 I2C 总线数据传输时序

主机向从机发起读数据
主机发送完成 7 位地址数据后，第 8 个 SCL 时钟发送高电平，表示读方向，从机在第 9 个 SCL 时钟回应 ACK，并在后续的 8 个 SCL 时钟通过 SDA 传输数据字节。
数据字节传输完成后，主机设备在第 9 个 SCL 时钟可以回应 ACK，从机设备将会在之后的 8 个 SCL 时钟继续传输数据字节。若主机设备回应 NACK，则从机设备将会释放 SDA 数据线，主机设备可以发起 STOP 结束传输或者发起 RESTART 事件发起新的传输。
主机向从机发起写数据
主机发送完成 7 位地址数据后，第 8 个 SCL 时钟发送低电平，表示写方向，从机在第 9 个 SCL 时钟回应 ACK，之后的 8 个 SCL 通过 SDA 接收数据字节。
数据字节传输完成后，若主机设备在第 9 个 SCL 时钟监测到 ACK，主机设备将会在之后的 8 个 SCL 时钟继续传输数据字节。若从机设备回应 NACK，则主机设备发起 STOP 结束传输，或发起 RESTART 事件发起新的传输。

18.4 功能说明

18.4.1 I2C 模块框图

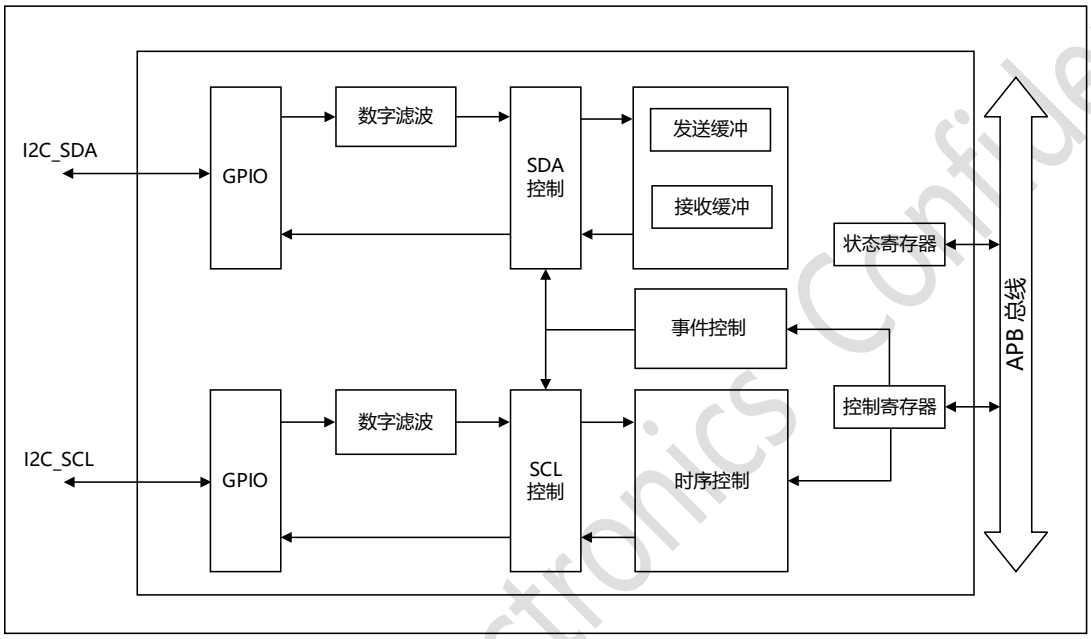

图 18-6 I2C 模块结构框图

18.4.2 I2C 初始化配置

I2C 模块支持标准速度模式、快速模式和超快速模式，模块内部时钟先基于 APB 时钟 PCLK 预分频，再被用于调控 SCL 时钟的高低电平设置。

时钟预分频

I2C 模块内的工作时钟 I2CCLK 是由 APB 时钟 PCLK 根据 I2C_CR1.PRESC 的值分频得到，I2CCLK 的时钟周期用 T_{I2CCLK} 表示，如下式所示。

$$T_{I2CCLK} = (1 + PRESC) \times T_{PCLK}$$

注：预分频寄存器配置值建议大于 2。

SCL、SDA 输出延时调节

SDL 和 SDA 在总线上的建立保持时间关系，可通过配置 SDA 延时输出寄存器 I2C_TIMING.SDADEL_O 和 SCL 延时输出寄存器 I2C_TIMING.SCLDEL_O 进行调节。

寄存器 SDADEL_O 用于调节保持时间，即在 SCL 下降沿后，经过$\mathrm{SDADEL_O} \times T_{I2CCLK}$时长，SDA 变化。

寄存器 SCLDEL_O 用于调节建立时间，即在 SDA 变化沿后，增加$\mathrm{SCLDEL_O} \times T_{I2CCLK}$时长的低电平时间。

SCL 高低电平配置

配置 I2C_TIMING.SCLH 控制 SCL 的高电平时长，I2C 模块内部对总线上 SCL 进行同步和内部处理共计需要 4 个周期，高电平时长计算如下式所示。

$$t_{SCLH} = (1 + SCLH) \times T_{I2CCLK} + 4 \times T_{PCLK}$$

配置 I2C_TIMING.SCLL 控制 SCL 的低电平时长，该时长同时也受到寄存器 SDADEL_O、SCLDEL_O 的影响。

$$t_{SCLL} = (SDADEL\_O + SCLDEL\_O + (1 + SCLL)) \times T_{I2CCLK}$$

注：由于走线等物理延时，总线上高低电平的实际延时会稍大于t_{SCLH}和t_{SCLL}。

SCL、SDA 输入延时调节

从模式下，SDA 和 SCL 信号输入 I2C 模块内部的延时，可由寄存器 I2C_TIMING.SDADEL_I 和 I2C_TIMING.SCLDEL_I 分别调节，用于平衡外部的走线延时。

寄存器 SDADEL_I 使得总线上 SDA 电平，延时$\text{SDADEL_I} \times T_{I2CCLK}$时长，到达 I2C 模块内部。寄存器 SCLDEL_I 使得总线上 SCL 电平，在延时$\text{SCLDEL_I} \times T_{I2CCLK}$时长后，再到达 I2C 模块内部。

默认情况下，SDADEL_I=1，SCLDEL_I=0。

18.4.3 I2C 主模式

I2C 工作在主模式时，可在总线上产生 START 事件，访问特定地址并指定读写方向，若地址匹配成功则收到 ACK 响应，匹配失败则收到 NACK。地址匹配成功后，根据指定方向，主机向从机发送数据或读取从机的数据；发送/读取数据结束后，若仍需继续通信，则需在总线上产生 RESTART 事件，重新向该从机发起访问或向其他从机发起访问；若要停止通信，则需在总线上产生 STOP 信号停止通信，释放总线。

访问地址管理

主机的读写方向通过 I2C_CR2.WR 寄存器控制，WR 为 0 时表示对从机发起写访问，为 1 时表示对从机发起读访问。

主机访问的地址由 I2C_CR2.ADDR10 和 I2C_CR2.SADDR 控制。主机发送地址时，若 ADDR10=0，则 I2C 主机以 7 位格式发送地址，SADDR[7:1]中的地址有效；若 ADDR10=1，则 I2C 主机以 10 位格式发送地址，SADDR[9:0]中的地址有效。

NACK 处理

若地址匹配失败或发送数据阶段，从机发送 NACK 响应，则中断状态寄存器 I2C_ISR.NACKF 硬件置 1，软件写 1 清零。若 NACKF_IE=1，则会产生中断。

检测到总线上的 NACK 后，主机可配置 START_STOP=11，在总线上发送 RESTART 再次发起通信；或配置 START_STOP=10，在总线上发送 STOP，终止本次通信。

主模式发送数据流程

1. 配置时序寄存器 I2C_TIMING，设置 SCL 高低电平时长等时序参数。
2. 配置模式选择寄存器 I2C_CR1.MS=0，I2C 处于主模式工作状态。
3. 配置 I2C_CR2.WR=0 选择向从机写数据。
4. 配置访问地址 I2C_CR2.SADDR，并配置 I2C_CR2.ADDR10 选择 7/10 位访问格式。
5. 检查状态寄存器 I2C_ISR.BUS_BUSY=0;
6. 配置 I2C_CR2.START_STOP=01，I2C 主机在总线上发送 START，并自动发送方向和地址。
7. 查询发送缓冲状态 I2C_ISR.TXE，若 TXE=1，软件向 I2C_TDR 写入数据，若地址匹配成功，则硬件自动发送 I2C_TDR 中的数据。若 TXE_IE 为 1，则会产生中断。
8. 重复上述步骤 7 连续发送数据。
9. 所有数据都填入 I2C_TDR 后，查询寄存器 I2C_ISR.TXC，若所有数据都发送完成，则 TXC 由硬件置 1。
10. TXC=1 时，若 I2C_CR1.TXC_IE=1，则会产生中断。
11. 所有数据发送完成后，若需要继续对其他从机发起通信，则重新配置地址方向等参数，配置

START_STOP=11，重新发起访问。若停止访问，则配置 START_STOP=10，终止本次通信。

注：主模式下，I2C 发送完成全部地址、数据和事件后，无待发送字节或事件时，TXC 状态寄存器会置 1，可用于判断主机的发送过程是否完成。

接收数据管理

I2C 主机接收数据时，通过接收数据数目寄存器 I2C_CR2.BYTES_NUM 和接收重载寄存器 I2C_CR1.RELOAD 来管理。地址匹配成功后，若 BYTES_NUM 为非零值，则 I2C 将连续读取从机 BYTES_NUM 个数据。读取所有数据完成时，RELOAD 用于控制在本次 BYTES_NUM 个数据接收完成后，是否还继续接收数据。

若 RELOAD=0，则连续接收完成 BYTES_NUM 个数据后，BYTES_NUM 被硬件清零，I2C 将在最后一个响应位处发送 NACK 响应，接收完成状态寄存器 I2C_ISR.RXC 由硬件置 1，软件写 1 清零。后续只能发送 RESTART 或 STOP。

若 RELOAD=1，则连续接收完成 BYTES_NUM 个数据后，BYTES_NUM 被硬件清零，I2C 将在最后一个响应位处发送 ACK 响应，重载接收完成状态寄存器 I2C_ISR.RXC_RE 由硬件置 1，软件写 1 清零。此时，软件应再配置 BYTES_NUM 寄存器和 RELOAD 寄存器继续接收数据。

主模式接收流程

1. 配置时序寄存器 I2C_TIMING，设置 SCL 高低电平时长等时序参数。
2. 配置模式选择寄存器 I2C_CR1.MS=0，I2C 处于主模式工作状态。
3. 配置 I2C_CR2.WR=1 选择向从机读取数据。
4. 配置访问地址 I2C_CR2.SADDR，并配置 I2C_CR2.ADDR10 选择 7/10 位访问格式。
5. 检查状态寄存器 I2C_ISR.BUS_BUSY=0;
6. 配置 I2C_CR2.START_STOP=01，I2C 主机在总线上发送 START 事件，并自动发送方向和地址。
7. 根据期望读取的数据数量，配置 I2C_CR2.RELOAD 和 I2C_CR2.BYTES_NUM，硬件自动读取数据。
8. 读取的数据存放到接收数据寄存器 I2C_RDR 中，该寄存器非空时 I2C_ISR.RXNE 由硬件置 1，软件读取 I2C_RDR 清零或软件写 1 清零，若 I2C_CR1.RXNE_IE=1，则会产生中断。
9. 接收 BYTES_NUM 个数据完成后，若 RELOAD=0，则 I2C_ISR.RXC 硬件置 1，写 1 清零；若 RELOAD=1，则 I2C_ISR.RXC_RE 硬件置 1，写 1 清零。
10. 读取数据完成后，若 RELOAD 为 1，则重复上述步骤 7-9，直到最后配置 RELOAD 为 0，并接收完剩下的数据。
11. 所有数据接收完成后，若需要继续对其他从机发起通信，则重新配置地址方向等参数，配置 START_STOP=11，重新发起访问。若停止访问，则配置 START_STOP=10，终止本次通信。

18.4.4 从模式

I2C 工作在从模式下，收到 START 或 RESTART 事件，则 I2C 准备接收地址并进行匹配，地址匹配成功则发送 ACK 响应，不匹配则发送 NACK 响应。若地址匹配成功，则地址匹配寄存器 I2C_ISR.ADDRM 由硬件置 1，软件写 1 清零，若 I2C_CR1. ADDRM_IE=1，则会产生中断。

本机地址管理

I2C 从模式下的本机地址和本机地址访问格式，分别由 I2C_OAR.OWN_ADDR 和 I2C_OAR.OWN_ADDR10 控制。

当 OWN_ADDR10 配置为 0 时，I2C 被 7 位地址访问，OWN_ADDR[7:1]有效。

当 OWN_ADDR10 配置为 1 时，I2C 被 10 位地址访问，OWN_ADDR[9:0]有效。

方向管理

从机的读写方向，在地址匹配后，记录到 I2C_ISR.DIR 寄存器中。若 DIR=0，则从机接收数据，软件从

I2C_RDR 寄存器读取，直到主机发起新的访问或停止通信。从机接收数据时，通常会发送 ACK 响应，则通信继续进行。

若 DIR=1，从机发送数据，软件向 I2C_TDR 寄存器写入数据，发送给主机，直到检测到主机发送 NACK 响应，发送数据结束，I2C 从机等待主机发起新的访问或停止通信。

NACK 处理

从模式下，接收到 NACK 后，I2C 作为从机等待总线事件，若收到 START 或 RESTART 事件，则重新匹配地址。若收到 STOP 事件，则表示这次通信被中止。

从模式接收数据时，I2C 也能主动发送 NACK，接收数据过程中，若配置控制寄存器 I2C_CR2.NACK=1，则会在接收完成当前字节后，发送 NACK 响应，该寄存器会在 NACK 发送完成后硬件清零，或检测到总线上的 START、RESTART 和 STOP 事件后由硬件清零。

从模式发送数据流程

1. 配置时序寄存器 I2C_TIMING，设置 SCL 高低电平时长等时序参数。
2. 配置模式选择寄存器 I2C_CR1.MS=1，I2C 处于从模式工作状态。
3. 配置本机地址 I2C_OAR.OWN_ADDR10，并配置 I2C_OAR.OWN_ADDR10 选择 7/10 位本机地址访问格式。
4. 查询地址匹配标志位 I2C_ISR.ADDRM，若 ADDRM=1 则表示本机地址匹配成功，查询 I2C_DIR 寄存器判断收发数据方向。
5. 查询 DIR=1，则为从机发送数据，查询发送缓冲状态 I2C_ISR.TXE，若 TXE=1，软件向 I2C_TDR 写入数据，若地址匹配成功，则硬件自动发送 I2C_TDR 中的数据。TXE=1 时，若 TXE_IE 为 1，则会产生中断。
6. 重复上述步骤 5 连续发送数据。
7. 若从机需要提前中止数据发送，则配置 I2C_CR2.NACK=1，则会在接收完成当前字节后，发送 NACK 响应。
8. 等待总线上的 STOP 事件结束通信或总线上的 RESTART 事件重建通信。

从模式接收数据流程

1. 配置时序寄存器 I2C_TIMING，设置 SCL 高低电平时长等时序参数。
2. 配置模式选择寄存器 I2C_CR1.MS=1，I2C 处于从模式工作状态。
3. 配置本机地址 I2C_OAR.OWN_ADDR10，并配置 I2C_OAR.OWN_ADDR10 选择 7/10 位本机地址访问格式。
4. 查询地址匹配标志位 I2C_ISR.ADDRM，若 ADDRM=1 则表示本机地址匹配成功，查询 I2C_DIR 寄存器判断收发数据方向。
5. 若 DIR=0，则为数据接收方向，接收数据后，RXNE 置 1，软件读取 I2C_RDR 寄存器清零或软件写 1 清零，若 I2C_CR1.RXNE_IE=1，则会产生中断。
6. 重复上述步骤 5 连续接收数据。
7. 等待总线上的 STOP 结束通信或总线上的 RESTART 重建通信。

18.4.5 时钟扩展

I2C 作为从机使用时，能够输出 SCL 低电平扩展时钟，包括如下情况：

- 从机配置的 SCL 低电平时长长于主机，则主机的 SCL 的低电平时长会被从机延长。
- 从机地址匹配且为数据发送方向，若 I2C_ISR.TXE=1，发送缓冲中缺少数据，则从机会输出 SCL 低电平，直到软件向 I2C_TDR 中填入数据，从而防止发送无效数据。

- 从机地址匹配且为数据接收方向，若 I2C_ISR.RXNE=1，接收缓冲中数据未被软件处理，同时下一笔数据已接收完成，则从机会输出 SCL 低电平，等待软件处理数据，从而防止数据丢失。

配置 I2C_CR1.NO_STRETCH=1 会禁用时钟扩展功能，禁用该功能后，若 I2C_TDR 无待发送数据，而主机在读取数据，发生下溢；若 I2C_RDR 中有待处理数据，而新的数据被接收，则发生上溢。
上溢或下溢发生时候，状态寄存器 I2C_ISR.UDRR_OVRR 硬件置 1，软件写 1 清零，若 I2C_CR1.ERR_IE=1，则会产生中断。

18.4.6 总线状态监测

I2C 使能后，持续监测总线上的事件。
START/RESTART 事件
若总线上出现 START 或 RESTART 事件，则 I2C_ISR.STARTF 由硬件置 1，软件写 1 清零，若 I2C_ISR.STARTF_IE=1，则会产生中断。
STOP 事件
若总线上出现 STOP 事件，则 I2C_ISR.STOPF 由硬件置 1，软件写 1 清零，若 I2C_CR1.STOPF_IE=1，则会产生中断。
总线状态
I2C 检测到总线上的 START 事件后，I2C_ISR.BUSY 状态位由硬件置 1，检测到 STOP 事件后，BUSY 由硬件清零，软件通过检测 BUSY 能判断总线是否空闲。新的寻址行为应当在总线空闲时进行。

18.4.7 广播功能

I2C 支持广播地址 0b0000000 的检测，通过配置寄存器 I2C_CR1.GC_EN=1 使能该功能。
从模式下且 GC_EN=1，检测到广播地址时，I2C 匹配该地址，状态位 I2C_ISR.GC 由硬件置 1，软件写 1 清零，若 I2C_CR1.GC_IE=1，则会产生中断。

18.4.8 无地址操作模式

I2C 通常基于地址进行通信，在无地址操作模式下，总线上的所有事件和传输的数据交由软件处理。
无地址操作主模式
主模式下，软件可发送 START、RESTART 和 STOP 事件，但硬件不会自动发送地址和方向，因此软件需要在 START 发送后，将发送的地址和方向，按照对应位的次序，写入 I2C_TDR 寄存器，硬件自动发送地址字节。
地址发送完成且 I2C_ISR.NACKF 未置位，表明地址匹配成功且收到响应位，后续读写数据、事件发送等操作遵循一般主模式操作。
无地址操作从模式
从模式下，软件通过查询 I2C_ISR.STARTF 状态或对应中断，判断总线上有无设备发起访问，检测到 STARTF 置 1 后，总线上的数据会按次序接收并存入接收缓冲寄存器 I2C_RDR 中，软件依次读取其中的数据，并判断总线上传输的地址是否为本机地址，并用软件解析收到的方向位，将该方向位写入 I2C_ISR.DIR 寄存器中。
若写入值为 1，发送方向，后续操作遵循从模式发送数据流程。
若写入值为 0，接收方向，后续操作遵循从模式接收数据流程。

18.4.9 多主机通信

总线上存在多个主机时，每个主机应配置特定的本机地址。

仲裁

总线上多个主机同时发起对从机的访问时，总线上可能发生仲裁，SCL 高电平期间，若有主机发送的 SDA 地址或数据位为低电平，而其他主机发送的对应位为高电平，则 SDA 为低电平的 I2C 设备仲裁成功，其他主机仲裁失败。仲裁成功的主机继续完成其通信过程。

地址/数据仲裁失败时，寄存器 I2C_ISR.ARBI_LOST 置 1，若 I2C_CR1.ERR_IE=1，则会产生中断。

主从转换

若地址匹配阶段发送仲裁失败，主机可转换为从模式并继续地址匹配，仲裁发生前的地址不会丢失。

若寄存器 I2C_CR1.DIS_SLAVE=1，则仲裁失败后，主机不会因仲裁转换为从模式，也不进行后续的地址匹配流程。

18.4.10 数字滤波

I2C 内置数字滤波器，对 SDA 和 SCL 输入信号滤波处理，通过寄存器 I2C_CR1.DNF 配置，DNF=0000 时，滤波功能被禁用；配置 DNF 为非零值时，滤波器能滤除最大脉冲宽度不超过 DNF * Tpclk 的噪声。

注：数字滤波器工作时，会增加总线上 SCL、SDA 同步到 I2C 模块的延时，因此 I2C 通信速率较高时，DNF 配置值不应太大。

18.4.11 清除发送缓冲

I2C 作主机或从机发送数据时，通信过程可能会被总线上的 NACK 或其他意外事件打断传输，导致 I2C_TDR 中有未发送的数据存在，发送缓冲空状态位 I2C_ISR.TXE=0。若软件需要释放发送缓冲，可置位 I2C_CR2.SET_TXE=1，硬件将置位 TXE，旧数据不再被发送。

18.4.12 超时检测

I2C 具有超时检测功能，对 SCL 为高电平且 SDA 为低电平的时长进行统计，若时长达到阈值，则超时状态寄存器 I2C_ISR.TIMEOUT 由硬件置 1，软件写 1 清零，若 ERR_IE=1，则会产生中断，该功能通过 I2C_CR1.TIMEOUT_SEL 寄存器配置，如下图 18-7 所示。TIMEOUT_SEL=000、110、111 时，禁用该功能。

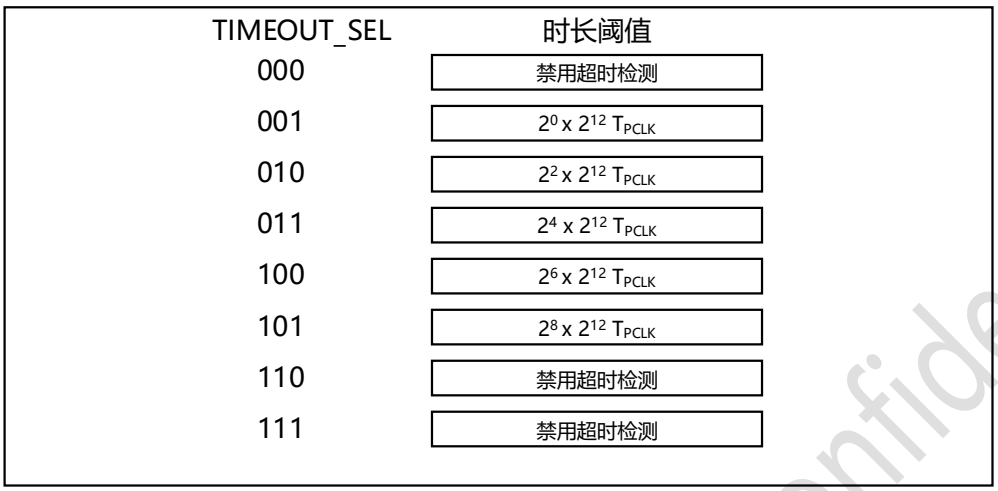

图 18-7 超时检测配置阈值示意图

18.4.13 寄存器 I2C_TIMING 配置示例

表 18-1 f_{PCLK}=48MHz 时序设置示例

参数	标准模式		快速模式	超快速模式
总线 SCL 频率	10kHz	100kHz	400kHz	1000kHz
PRESC [4:0]	0xB	0xB	0x2	0x2
SCLH [7:0]	0xC3	0x10	0x9	0x3
SCLL [7:0]	0xC5	0x12	0x16	0x8
SCLDEL_O [3:0]	0x2	0x2	0x3	0x0
SDADEL_O [3:0]	0x4	0x4	0x4	0x1
tSCLH	约 49us	约 4us	约 0.7us	约 350ns
tSCLL	约 51us	约 6us	约 1.8us	约 650ns

18.4.14 I2C 通信错误检测

状态寄存器 I2C_ISR 中的标志位 TIMEOUT、OVRR_UNRR、BUS_ERR 或 ARBI_LOST 为 1 时，表示 I2C 传输过程发生对应错误。

若有错误标志位为 1，错误中断使能寄存器 I2C_CR1.ERR_IE=1 时，会产生中断。

通信超时错误

从模式下，配置 TIMEOUT_SEL 使能超时检查并选择合适的阈值，当 SCL 电平为高、SDA 电平为低的时间超过阈值时，TIMEOUT 置 1。

溢出错误

当满足 I2C_CR1.NO_STRETCH=1 和以下条件时发生上溢或下溢错误：

- 主模式或从模式下，I2C_RDR 寄存器的值还未读出，I2C_ISR.RXNE=1，此时接收一个新的字节，上溢发生，新接收的字节丢失，并硬件自动发送 NACK，此时发生上溢出错误，OVRR_UNRR 由硬件置 1。

- 从模式下，从机发送完一个数据且收到 ACK 回应后，若软件未向 I2C_TDR 中填入新数据，TXE=1 时，同时主机再次发起读数据，则发生下溢出，且 OVRR_UNRR 置 1。

总线错误

I2C 模块工作时，当检测到 START 或 STOP 事件发生，且发生的位置不在每帧的第 9 个 SCL 时钟脉冲之后，则发生总线错误，BUS_ERR 置 1。

从模式下，当检测到错位的 START 或 RESTART 时，I2C 会重新进入地址识别状态；检测到 STOP 事件时，则会停止通讯。

仲裁失败错误

主模式下，总线上多个 I2C 同时发起通讯时，会发生仲裁，若输出 SDA 为高电平，而总线上 SDA 为低电平，则表明仲裁失败时，ARBI_LOST 由硬件置 1。

主模式下，发送地址和数据的阶段都能够仲裁。

若在地址匹配阶段仲裁失败，I2C_CR1.DIS_SLAVE=0 时，I2C 从主模式切换到从模式，并对总线上的地址进行匹配；I2C_CR1.DIS_SLAVE=1 时，I2C 保持为主模式，且停止地址发送。

若在数据匹配阶段仲裁失败，则由软件进行处理。

18.4.15 I2C 基于 DMA 操作

I2C 的主从模式均支持 DMA 操作，有关 DMA 的配置请参见第 8 章：直接存储器访问控制器（DMA）。

通过 DMA 发送数据

配置 DMA 发送使能 I2C_CR1.TX_DMAEN=1，I2C 使能有效后，若 I2C_ISR.TXE=1，则数据将由 DMA 配置的 SRAM 区写入 I2C_TDR 寄存器。

通过 DMA 接收数据

配置 DMA 接收使能 I2C_CR1.RX_DMAEN=1，I2C 使能有效后，若 I2C_ISR.RXNE=1，则 I2C_RDR 的数据将由 DMA 读取并存入 SRAM 中。

18.4.16 I2C 中断说明

表 18-2 I2C 中断请求说明

中断事件	状态标志	使能控制	中断清除方式
起始位检测	STARTF	STARTF_IE	中断标志写 1 清零
上溢/下溢	OVRR_UDRR	ERR_IE	对应中断标志写 1 清零 注：ARBI_LOST 主模式下有效 TIMEOUT 从模式下有效
总线错误	BUS_ERR		
仲裁丢失	ARBI_LOST		
超时检测	TIMEOUT		
接收完成	RXC	RXC_IE	对应中断标志写 1 清零 注：主模式下有效
重载接收完成	RXC_RE		
广播地址检测	GC	GC_IE	中断标志写 1 清零 注：广播功能使能时有效
无响应检测	NACKF	NACKF_IE	中断标志写 1 清零
停止位检测	STOPF	STOPF_IE	中断标志写 1 清零
地址匹配	ADDRM	ADDRM_IE	中断标志写 1 清零 注：从模式下有效

接收缓冲非空	RXNE	RXNE_IE	中断标志写 1 清零，或读 I2C_RDR 清零
发送缓冲空	TXE	TXE_IE	中断标志写 1 清零，或写 I2C_TDR 清零，或配置 I2C_CR2.TXE_SET 寄存器
发送完成	TXC	TXC_IE	中断标志写 1 清零，或写 I2C_TDR 清零 注：主模式下有效

18.4.17 低功耗特性

表 18-3 I2C 低功耗模式特性

模式	功能描述
睡眠（SLEEP）	该模式下 CPU 工作时钟将会关闭，I2C 能够正常使用，并且 I2C 中断能够退出睡眠模式。
深度睡眠 （DEEP_SLEEP）	该模式下 CPU 工作时钟将会关闭，I2C 工作时钟将会根据时钟 RCC 模块中的 RCC_SLEEP_CR 寄存器的值选择开关。 若深度睡眠模式下 I2C 配置成开启时钟，则 I2C 将正常工作，I2C 中断能够退出深度睡眠模式。 若深度睡眠模式下 I2C 配置成关闭时钟，则应先停止通讯并关闭 I2C 使能，再进入深度睡眠模式。
停止（STOP）	该模式下 CPU 和 I2C 的工作时钟都会关闭，I2C 模块内控制寄存器的内容会保持。 进入该模式前，应停止通讯并关闭 I2C 使能，再进入该模式。

18.5 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字(16 位)、字(32 位）访问。
注：I2C 中断状态寄存器不支持字符访问

表 18-4 I2C 寄存器概览

偏移地址	寄存器名	寄存器描述	复位值
0x00	I2C_CR1	I2C 控制寄存器1	0xB0000000
0x04	I2C_CR2	I2C 控制寄存器2	0x00000000
0x08	I2C_ISR	I2C 中断状态寄存器	0x00000002
0x0C	I2C_TIMING	I2C 时序配置寄存器	0x01420F13
0x10	I2C_OAR	I2C 本机地址配置寄存器	0x00000000
0x014	I2C_TDR	I2C 发送数据寄存器	0x00000000
0x018	I2C_RDR	I2C 接收数据寄存器	0x00000000

18.5.1 I2C 控制寄存器 1（I2C_CR1）

偏移地址：0x000
复位值：0xB0000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC[4:0]					Res.	STARTF_IE	ERR_IE	RXC_IE	GC_IE	NACKF_IE	STOPF_IE	ADDRM_IE	RXNE_IE	TXE_IE	TXC_IE
rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DNF[3:0]				Res.	TIMEOUT_SEL[2:0]			NOADDR	DIS_SLAVE	TX_DMAEN	RX_DMAEN	GC_EN	NO_STRETCH	MS	EN
rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:27	PRESC[4:0]	I2C 工作时钟预分频，由软件配置，用于 SCL 信号的高电平和低电平计数，以及 SDA 数据建立时间和保持时间计数。分频后的 I2C 工作时钟周期为 T_{I2CCLK}=(1+PRESC) x TPCLK。 注：该寄存器值应大于等于 2。
26	Res.	保留，保持为复位值。
25	STARTF_IE	起始位检测中断使能（Start detection flag interrupt enable） 0：禁止起始位检测 STARTF=1 产生中断。 1：使能起始位检测 STARTF=1 产生中断。
24	ERR_IE	错误检测中断使能（Error interrupt enable） 0：禁止错误产生中断。 1：使能错误产生中断。 注：错误标志位包括 ARBI_LOST=1、BUS_ERR=1、OVRR_UDRR=1、TIMEOUT=1。
23	RXC_IE	读数据完成中断使能（Master read complete interrupt enable） 0：禁止主模式读数据完成 RXC=1 或 RXC_RE=1 产生中断。 1：使能主模式读数据完成 RXC=1 或 RXC_RE=1 产生中断。 注：读数据完成中断只有在主模式下有效。
22	GC_IE	I2C 广播呼叫中断使能（General call interrupt enable） 0：禁止广播呼叫 GC=1 产生中断。 1：使能广播呼叫 GC=1 产生中断。
21	NACKF_IE	无应答标志中断使能（Not acknowledge flag interrupt enable） 0：禁止否定应答接收 NACKF=1 产生中断。 1：使能否定应答接收 NACKF=1 产生中断。
20	STOPF_IE	停止位标志中断使能（Stop detection flag interrupt enable） 0：禁止停止位检测 STOPF=1 产生中断。 1：使能停止位检测 STOPF=1 产生中断。
19	ADDRM_IE	地址匹配中断使能（Address match interrupt enable） 0：禁止地址匹配 ADDRM=1 产生中断。 1：使能地址匹配 ADDRM=1 产生中断。 注：地址匹配中断只有在从模式下有效。
18	RXNE_IE	接收缓冲非空中断使能（Receive buffer not empty interrupt enable）

		0：禁止接收数据缓冲非空 RXNE=1 产生中断。 1：使能接收数据缓冲非空 RXNE=1 产生中断。
17	TXE_IE	发送缓冲空中断使能（Transmit buffer empty interrupt enable） 0：禁止发送数据缓冲非空 TXE=1 产生中断。 1：使能发送数据缓冲非空 TXE=1 产生中断。
16	TXC_IE	主模式写完成中断使能（Transmit complete interrupt enable） 0：禁止主模式写数据结束 TXC=1 产生中断。 1：使能主模式写数据结束 TXC=1 产生中断。 注：写完成中断只有在主模式下有效。
15:12	DNF[3:0]	数字噪声滤波器（Digital noise filter），用于输入数字信号 SDA 和 SCL 的噪声滤波器，其可滤除脉宽不超过 DNF[3:0] x T_{PCLK} 的噪声。 0000：禁止数字滤波器。 0001：使能数字滤波器，可滤除脉宽不超过 T_{PCLK} 的噪声。 … … 1111：使能数字滤波器，可滤除脉宽不超过 15 x T_{PCLK} 的噪声。
11	Res.	保留，保持为复位值。
10:8	TIMEOUT_SEL[2:0]	I2C 从模式下超时中断阈值，用于检测 SCL 在高电平时 SDA 为低电平的持续时间。 000：禁用超时检测。 001：使能超时检测，超时阈值为 2^0x2^{12} T_{PCLK}。 010：使能超时检测，超时阈值为 2^2x2^{12} T_{PCLK}。 011：使能超时检测，超时阈值为 2^4x2^{12} T_{PCLK}。 100：使能超时检测，超时阈值为 2^6x2^{12} T_{PCLK}。 101：使能超时检测，超时阈值为 2^8x2^{12} T_{PCLK}。 110：禁用超时检测。 111：禁用超时检测。 注：超时检测只在从模式下有效。
7	NOADDR	无地址操作（No address operation），该位有效时，硬件不再发送或匹配地址，发送数据、接收数据和总线的事件控制都交由软件处理。 0：正常操作。 1：无地址操作。
6	DIS_SLAVE	主从自动切换禁用（Slave jump disable） 0：启用主从自动切换，I2C 在主模式下仲裁失败时切换为从机。 1：禁止主从自动切换，I2C 在主模式下仲裁失败时切换为从机。
5	TX_DMAEN	DMA 发送使能（Transmit DMA enable） 0：禁止 DMA 发送请求。 1：使能 DMA 发送请求。I2C 使能有效后，若 I2C_ISR.TXE=1，则产生 DMA 发送请求。
4	RX_DMAEN	DMA 接收使能（Receive DMA enable） 0：禁止 DMA 接收请求。 1：使能 DMA 接收请求。I2C 使能有效后，若 I2C_ISR.RXNE=1，则产生 DMA 接收请求。
3	GC_EN	广播呼叫功能使能（General call enable） 0：禁止广播呼叫功能，不对地址 0b00000000 应答。

		1：使能广播呼叫功能，对地址 0b00000000 应答。
2	NO_STRETCH	时钟周期延长禁止（Clock stretching disable），该位用于禁止 I2C 通过拉低 SCL 的方式，来降低通讯的时钟频率。 0：使能时钟延长。 1：禁止时钟延长。
1	MS	主从模式选择（Master or slave selection） 0：I2C 工作在主机模式。 1：I2C 工作在从机模式。 注：在多主机 I2C 总线上，MS=0 和 DIS_SLAVE=0 时，若发生地址仲裁失败，该位将由硬件置 1。
0	EN	I2C 工作开启或关闭。 0：关闭 I2C。 1：开启 I2C。

18.5.2 I2C 控制寄存器 2（I2C_CR2）

偏移地址：0x004
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RELOAD	BYTES_NUM[7:0]							
							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
TXE_SET	NACK	START_STOP[1:0]		WR	ADDR10	SADDR[9:0]									
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:25	Res.	保留，保持为复位值。
24	RELOAD	重载入（Reload）。Reload=1 时，主模式在读取了 BYTES_NUM 个数据后，应再配置 BYTES_NUM，继续读取数据。若 Reload=0，则置位 RXC；若 Reload=1，则置位 RXC_RE。 0：禁止重载入。主模式在读取了 BYTES_NUM 个数据，RXC 置 1，后续应结束本次通讯，发送 Stop 或 Restart。 1：使能重载入。主模式在读取了 BYTES_NUM 个数据后，RXC_RE 置 1，后续应再配置 BYTES_NUM，继续读取数据。 注：主模式读取数据时有效果。
23:16	BYTES_NUM[7:0]	读数据计数值（Bytes number），用于主模式读取数据字节计数，达到计数值后置位 M_RXC 或 RXC_RE，接收数据完成时硬件自动清零。 00000000：不接收数据。 00000001：接收数据为 1 字节。

		… … 11111111：接收数据为 255 字节。 注：只有在 EN=0 时软件写操作设置。
15	TXE_SET	发送缓冲释放（Set transmit buffer empty flag），用于无效发送缓冲 TX_TDR 中的待发送数据，软件写 1，硬件自动清零。 0：无操作。 1：置位发送缓冲空标志。
14	NACK	无应答标志生成（Nack generation），用于在从模式下，接收完当前字节后，产生无应答 NACK 标志，由软件设置为 1，发送完成后硬件清零。 0：无操作。 1：生成无应答标志。 注：从模式下接收地址/数据时有效。
13:12	START_STOP[1:0]	起始位、重启位和停止位生成（Strat or restart or stop generation），I2C 主模式下，在总线发起开始、重启和停止信号，由软件设置，发送完成后硬件清零。 00：无操作。 01：发起开始信号。 10：发起停止信号。 11：发起重启信号。 注：主模式下有效。
11	WR	读写操作（Read or Write） 0：I2C 主机发起写操作。 1：I2C 主机发起读操作。
10	ADDR10	10 位地址选择（10bit address） 0：I2C 主机发起地址为 7 位。 1：I2C 主机发起地址为 10 位。
9:0	SADDR[9:0]	访问的从机地址（Slave address） SADDR10=0 时，SADDR[7:1]为访问的从机地址；SADDR10=1 时，则 SADDR[9:0]为访问目标从机地址。

18.5.3 I2C 中断状态寄存器（I2C_ISR）

偏移地址：0x008
复位值：0x00000002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT	STARTF	BUSY	DIR	OVRR_UDRR	BUS_ERR	ARBI_LOST	GC	RXC	RXC_RE	NACKF	STOPF	ADDRM	RXNE	TXE	TXC
w1c	w1c	ro	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c

Bit	Field	Description
31:16	Res.	保留，保持为复位值。
15	TIMEOUT	超时检测标志（Timeout detection flag），当 SCL 为高电平状态且 SDA 为低电平状态的时长达到阈值，该标志位硬件置 1，软件写 1 清零。 注：该标志位仅当 TIMEOUT_SEL[2:0]配置使能，且 MS=1 从模式下有效。
14	STARTF	起始位检测标志（Start detection flag），总线上检测到起始位或重启位，该标志位硬件置 1，软件写 1 清零。
13	BUSY	总线忙（Bus busy），该标志用于指示总线上正在进行通信。检测到起始位时，该位硬件置 1；检测到停止位时，该位硬件清零。
12	DIR	传输方向标志（Transfer direction），从模式下只读，该标志位在地址匹配时硬件更新，软件可读。 0：读传输，I2C 从模式接收数据。 1：写传输，I2C 从模式发送数据。 注：从模式下且 NOADDR=1 时，需要软件处理总线数据，识别传输方向并将方向值写入该寄存器。
11	OVRR_UDRR	上溢/下溢标志位（Overrun/Underrun），数据通信期间若 I2C_RDR 或 I2C_TDR 发生上溢/下溢错误，该标志由硬件置 1，软件写 1 清零。
10	BUS_ERR	总线错误（Bus error），当检测到 I2C 总线上出现错位的起始位或停止位，该标志位硬件置 1，软件写 1 清零。
9	ARBI_LOST	仲裁丢失（Arbitration lost），总线上多主机同时发起通信，发生仲裁且仲裁丢失时，该标志位硬件置 1，软件写 1 清零。
8	GC	广播检测标志（General call），GC_EN=1 时，I2C 检测总线上的广播地址，收到广播地址时，该标志位硬件置 1，软件写 1 清零。
7	RXC	读数据完成（Receive complete），Reload=0 时，接收数据字节数达到 BYTES_NUM，该位硬件置 1，软件写 1 清零。
6	RXC_RE	重载读数据完成（Receive complete reload），Reload=1 时，接收数据字节数达到 BYTES_NUM，该标志位硬件置 1，软件写 1 清零或软件重写 BYTES_NUM 清零该标志位。
5	NACKF	无应答标志检测（NACK detection flag），I2C 发送字节后检测到 NACK 时，该标志位硬件置 1，软件写 1 清零。
4	STOPF	停止位检测标志（Stop detection flag），总线上检测到停止位时，该

		标志位硬件置 1，软件写 1 清零。
3	ADDRM	地址匹配（Address match），接收地址与本机地址一致时，该标志位硬件置 1，软件写 1 清零。
2	RXNE	接收缓冲非空标志（Receive buffer not empty），I2C_RDR 接收到新数据时，该标志位硬件置 1，软件写 1 清零或通过读 I2C_RDR 寄存器清零。
1	TXE	发送缓冲空标志（Transmit buffer not empty），I2C_TDR 中无需发送数据时，硬件置 1，软件写 1 清零或者写 I2C_TDR 寄存器清零。
0	TXC	发送完成标志（Transmit complete），I2C 主模式下至少发送一个字节地址/数据或事件，且所有地址/数据/事件发送完成时，该标志位硬件置 1，软件写 1 清零。若存在数据或事件待发或正在发送，该状态位硬件清零。 注：该标志位主模式下有效。

18.5.4 I2C 时序配置寄存器（I2C_TIMING）

偏移地址：0x00C
复位值：0x 01420F13

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

Bit	Field	Description
31:30	Res.	保留，保持为复位值。
29:27	SCLDEL_I[2:0]	SCL 输入延时（SCL_DELAY_INPUT）。从模式下 SCL 信号采样延时，用于延迟 SCL 信号被内部逻辑采样。 000：无延时。 001：延时 1 x T_{I2CCLK}。 010：延时 2 x T_{I2CCLK}。 …… 111：延时 7 x T_{I2CCLK}。
26:24	SDADEL_I[2:0]	SDA 输入延时（SDA_DELAY_INPUT）。从模式下 SDA 信号采样延时，用于延迟 SDA 信号被内部逻辑采样。 000：无延时。 001：延时 1 x T_{I2CCLK}。 010：延时 2 x T_{I2CCLK}。 ……

		111：延时 7 x T_{I2CCLK}。
23:20	SCLDEL_O[3:0]	SCL 延时输出（SCL_DELAY_OUTPUT）。I2C 主从模式下，输出 SCL 建立时间延时，用于在检测到 SDA 边沿信号后输出 SCL 信号上升沿延时，由软件配置，硬件将延迟 SCL 信号上升沿输出。 0000：无延时。 0001：延时 1 x T_{I2CCLK}。 0010：延时 2 x T_{I2CCLK}。 …… 1111：延时 15 x T_{I2CCLK}。
19:16	SDADEL_O[3:0]	SDA 延时输出（SDA_DELAY_OUTPUT）。I2C 主从模式下，输出 SDA 保持时间延时，用于在检测到 SCL 下降沿信号后延时 SDA 信号输出。 0000：无延时。 0001：延时 1 x T_{I2CCLK}。 0010：延时 2 x T_{I2CCLK}。 …… 1111：延时 15 x T_{I2CCLK}。
15:8	SCLH [7:0]	SCL 高电平时长（SCL high period），用于主模式下生成输出的 SCL 高电平周期。 TSCLH= (1+ SCLH) x T_{I2CCLK}。
7:0	SCLL[7:0]	SCL 低电平时长（SCL low period） TSCLL= (1+ SCLL) x T_{I2CCLK}。

18.5.5 I2C 本机地址配置寄存器（I2C_OAR）

偏移地址：0x010
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	OWN_ADDR10	OWN_ADDR[9:0]									
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:11	Res.	保留，保持为复位值。
10	OWN_ADDR10	本机地址 10 位模式（Own Address 10-bit mode） 0：本机地址为 7 位地址，OWN_ADDR[7:1]有效。 1：本机地址为 10 位地址，OWN_ADDR[9:0]有效。
9:0	OWN_ADDR[9:0]	本机地址（Own Address）。

18.5.6 I2C 发送数据寄存器（I2C_TDR）

偏移地址：0x014
复位值：0x00000000

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

Bit	Field	Description
31:8	Res.	保留，保持为复位值。
7:0	TDR[7:0]	发送数据寄存器（Transmit data register）。

18.5.7 I2C 接收数据寄存器（I2C_RDR）

偏移地址：0x018
复位值：0x00000000

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

Bit	Field	Description
31:8	Res.	保留，保持为复位值。
7:0	RDR[7:0]	接收数据寄存器（Receive data register）。

19 通用异步收发器（UART）

19.1 简介

UART 能够灵活地与外部设备进行全双工数据交换，内部可编程波特率发生器实现了多种波特率，满足了外部设备对工业标准非归零码（NRZ）异步串行数据格式的要求。

19.2 主要特性

UART 主要的功能特性如下：

- 全双工异步通信
- NRZ 标准格式
- 可编程波特率
- 可编程数据字长（6 位、7 位、8 位或 9 位）
- 可编程的数据传输顺序，支持 MSB 在前或 LSB 在前
- 停止位数目可配（支持 1 个或 2 个停止位）
- 单线半双工通信
- 数据收发有独立使能
- 支持数据收发中断和错误检测中断
- 支持奇偶校验
- 支持通过 DMA 收发数据

19.3 功能说明

19.3.1 UART 模块框图

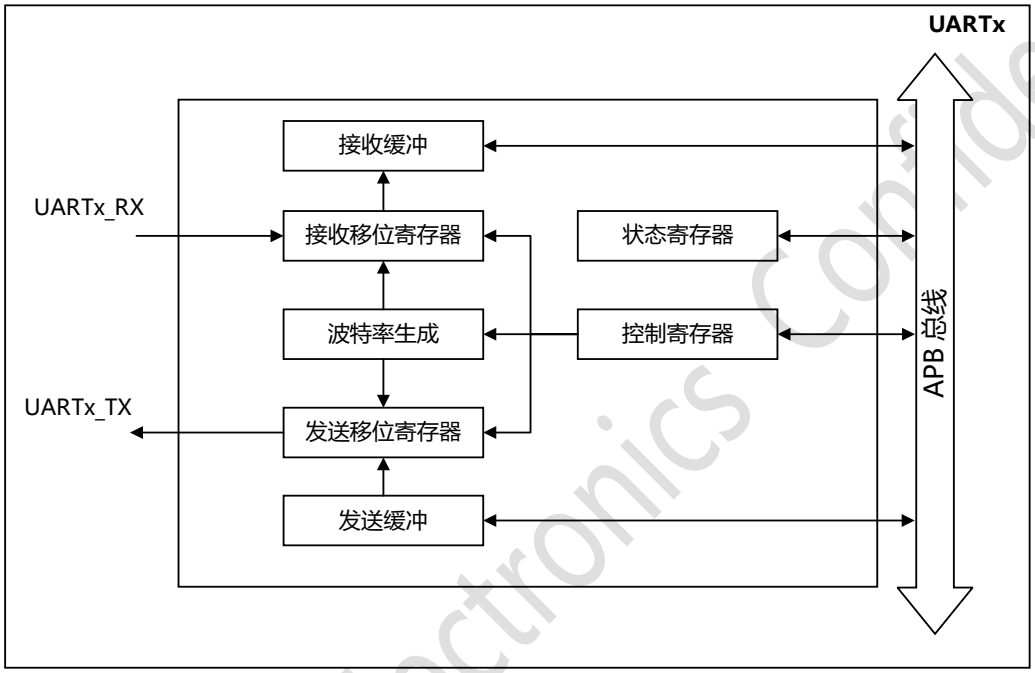

图 19-1 UART 模块结构框图

19.3.2 UART 功能引脚

表 19-1 UART 输入/输出引脚

引脚	类型	描述
UARTx_RX	输入引脚	串行数据输入引脚，UART 模块对其进行 16 倍过采样。
UARTx_TX	输出引脚	串行数据输出引脚，UART 模块以特定波特率输出数据。

19.3.3 UART 数据帧

UART 控制寄存器用于配置数据的格式，数据结构包括起始位、数据位、奇偶校验位（若使能）和停止位。空闲帧是指一帧时间内，管脚电平为默认电平，默认电平为高电平。

不同配置下的通信数据格式如下图所示。

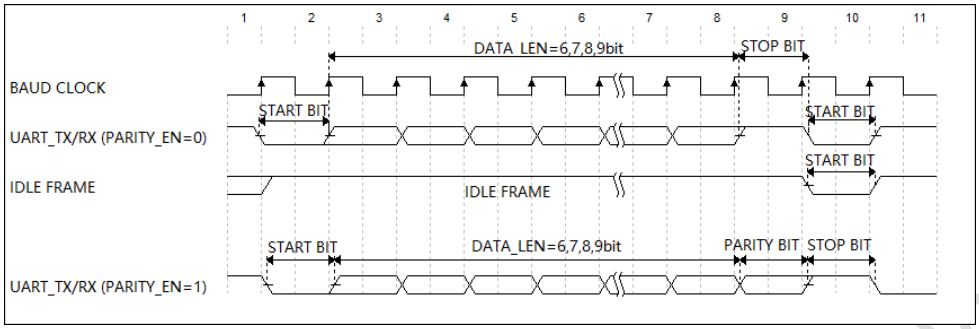

图 19-2 UART 数据帧时序图

UART_CR.DATA_LEN 寄存器控制收发数据长度，UART_CR.STOP_LEN 寄存器控制收发停止位的长度。

DATA_LEN	数据长度	STOP_LEN	停止位长度
2'b00	6位数据	1'b0	1位停止位
2'b01	7位数据	1'b1	2位停止位
2'b10	8位数据		
2'b11	9位数据		

图 19-3 数据长度和停止位长度控制示意图

奇偶校验

UART_CR.PARITY_EN 控制 UART 是否发送/接收奇偶校验位，若使能，则 UART 通信时，会在数据位和停止位之间，再发送/接收一位奇偶校验位。

UART_CR.PARITY_SEL 用于选择是奇校验还是偶校验：

- 0：偶校验计算。例：数据=00110101，则偶校验位值为 0，使‘1’的个数位为偶数。
- 1：奇校验计算。例：数据=00110101，则奇校验位值为 1，使‘1’的个数位为奇数。

数据位发送顺序

UART 收发数据各位的默认顺序为先收发最低有效位 LSB。UART 支持先发最高有效位 MSB 的功能，若 UART_CR.MSB_FIRST=1，则 UART 先收发最高有效位，该寄存器的值须在使能 UART 前配置，且通讯过程中不应改变。

19.3.4 UART 数据发送

UART 发送数据前，必须打开 UART_CR.TX_EN 来使能发送功能，且 UART 通信的双方应保持一致的波特率。

UART 数据发送流程

1. 配置 UART_CR 寄存器的 DATA_LEN、PARITY_EN、PARITY_SEL 和 STOP_LEN，设定发送数据长度、是否奇偶校验以及校验类型、发送数据位顺序和停止位长度。
2. 配置 UART_BRR 寄存器的波特率。
3. 使能 UART_CR 寄存器的发送使能寄存器 TX_EN 和 UART 使能寄存器 EN。
4. 检查寄存器 UART_ISR 中的 TXE 寄存器值是否为 1，若为 1 则表明寄存器 UART_TDR 为空，无待发送数据，软件向 UART_TDR 写入需发送的数据。重复该步骤，UART 将连续发送数据。
   当 TXE=1 时，若 UART_CR 寄存器中的位 TXE_IE=1，则会产生中断。
5. 在最后一个数据写入 UART_TDR 后，等待发送完成状态标志位 TXC 被硬件置 1，表明最后一个数据发送完成，此时可以关闭 UART_CR.EN。

当 TXC=1 时，若 UART_CR 寄存器中的 TXC_IE=1，则会产生中断。

注：URAT 在发送数据期间，不应改变 UART_CR 和 UART_BRR 寄存器，否则可能导致发送数据不完整。

19.3.5 UART 数据接收

UART 接收数据前，必须打开 UART_CR.RX_EN 来使能接收功能，且 UART 通信的双方应保持一致的波特率。

UART 数据接收流程

1. 配置 UART_CR 寄存器中的 DATA_LEN[1:0]、PARITY_EN、PARITY_SEL 和 STOP_LEN，设定接收数据长度、是否奇偶校验以及校验类型、停止位长度。
2. 配置 UART_BRR 寄存器设定波特率。
3. 使能 UART_CR 寄存器的接收使能寄存器 RX_EN 和总使能寄存器 EN。
4. 检查寄存器 UART_ISR 中的 RXNE 寄存器值为 1，则表明寄存器 UART_RDR 中有未读取的接收数据，软件从 UART_RDR 读取需发送的数据。重复该步骤，读取接收的数据。
   RXNE=1 时，若配置 UART_CR.RXNE_IE=1，则会产生中断。

起始位检测

UART 起始位检测同样使用 16 倍采样率的方式，当 UARTx_RX 数据线的信号由高电平跳变至低电平，根据 UART_BRR 的值，UART 在起始位的中间位置采样三次，至少 2 次采样为 0 时，认为起始位有效，开始接收后续数据；否则，认为起始位无效并回到初始状态。

空闲帧检测

UART 接收到空闲帧时：

- UART_ISR.IDLE 被硬件置位。
- 若配置 UART_CR.IDLE_IE=1，则产生中断。

溢出错误

若 URAT_RDR 接收的数据被读出前，又接收了一个数据，则发生溢出错误。

溢出错误发生时：

- UART_ISR.OVRR 被硬件置位。
- 若配置 UART_CR.ERR_IE=1，则产生中断。
- UART_RDR 中数据保留，新接收的数据丢失。

帧错误

若停止位未在预期时间被接收到，则发生帧错误。

帧错误发生时：

- UART_IST.FRAME_ERR 被硬件置位。
- 若配置 UART_CR.ERR_IE=1，则产生中断。
- 接收的数据存到 UART_RDR 寄存器中，UART_ISR.RXNE 硬件置位。

奇偶校验错误

若奇偶校验位使能，硬件内部对接收的数据计算奇偶校验位，若该值与接收的校验位不一致，则发生奇偶校验错误。

奇偶校验错误发生时：

- UART_IST.PARITY_ERR 被硬件置位。
- 若配置 UART_CR.ERR_IE=1，则产生中断。
- 接收的数据存到 UART_RDR 寄存器中，UART_ISR.RXNE 硬件置位。

19.3.6 UART 波特率发生器

UART 模块数据收发数据时，波特率发生器通过 UART_BRR 寄存器中 DIV_FRAC[3:0]和整数分频值 DIV_INT[15:0]配置。其中，DIV_FRAC[3:0]为小数分频值，DIV_INT[15:0]为整数分频值。
波特率计算方式为：

$$波特率 = \frac{f_{PCLK}}{(DIV\_INT + \frac{DIV\_FRAC}{16})}$$

例如，在 f_{PCLK}=48MHz 时，UART 传输波特率为 115200bps，则有：

$$\frac{48000000}{115200} \approx 416 + 0.667 = DIV\_INT + \frac{DIV\_FRAC}{16}$$

根据上述计算方式，DIV_INT[15:0] = 416 = 0x1A0，DIV_FRAC[3:0] = 0xA。

19.3.7 UART 单线半双工模式

UART 支持单线半双工通讯，UART_CR.HALF_DUP=1 时，切换到单线半双工模式，特性如下：
- 使用 UART_TX 引脚收发数据。
- 硬件禁用 UART_RX 引脚。
- 无数据发送时，UART_TX 引脚始终处于高阻状态，准备接收数据。
- UART_TDR 中有待发送数据时，立即发送。

单线半双工模式下，UART_TX 管脚应配置成开漏模式且外部上拉。当 UART_TDR 中有数据需要发送时，不论是否正在接受数据，都将开始数据发送。因此软件需要避免线上的数据冲突。

19.3.8 UART 基于 DMA 操作

UART 的数据接发都支持 DMA 功能，有关 DMA 的配置请参见章节：直接存储器访问控制器(DMA) 。
通过 DMA 发送数据
配置 DMA 发送使能 UART_CR.TX_DMAEN=1，UART 使能有效后，若 UART_ISR.TXE=1，则 SRAM 的数据将由 DMA 写入 UART_TDR 寄存器，UART 将自动发送该数据。
通过 DMA 接收数据
配置 DMA 接收使能 UART_CR.RX_DMAEN=1，UART 使能有效后，若 UART_ISR.RXNE=1，则 UART_RDR 寄存器的数据将由 DMA 读取并存入 SRAM 中。

19.3.9 UART 中断说明

UART 支持发送缓冲空中断、发送完成中断、接收缓冲非空中断、错误中断、空闲帧中断。

表 19-2 数据长度和停止位长度控制示意图

中断事件	中断标志	中断使能	清零方式
发送缓冲空中断	TXE	TXE_IE	中断标志写 1 清零 向 UART_TDR 写数据清零

发送完成中断	TXC	TXC_IE	中断标志写 1 清零 向 UART_TDR 写数据清零
接收缓冲非空中断	RXNE	RXNE_IE	中断标志写 1 清零 读 UART_RDR 硬件自动清零
错误中断	FRAME_ERR PARITY_ERR OVRR_ERR	ERR_IE	对应错误中断标志写 1 清零
空闲帧中断	IDLE	IDLE_IE	中断标志写 1 清零

注：发送完成指 UART 内所有数据发送完成，即 UART_TDR 无待发送数据，且没有正在发送的数据。

19.3.10 UART 低功耗特性

表 19-3 UART 低功耗模式特性

模式	功能描述
睡眠（SLEEP）	该模式下 CPU 工作时钟将会关闭，UART 能够正常使用，UART 中断能够退出睡眠模式。
深度睡眠 （DEEP_SLEEP）	该模式下 CPU 工作时钟将会关闭，UART 工作时钟的开关将会由 RCC 模块中的 RCC_SLEEP_CR 寄存器的配置决定。 若深度睡眠模式下 UART 配置成开启时钟，则 UART 将正常工作，UART 中断能够退出深度睡眠模式。 若深度睡眠模式下 UART 配置成关闭时钟，则应先关闭 UART 使能，再进入深度睡眠模式。
停止（STOP）	该模式下 CPU 和 UART 的工作时钟都会关闭，UART 模块内控制寄存器的状态会保持。 进入该模式前，应先关闭 URAT 使能，再进入该模式。

19.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字(16 位)、字(32 位）访问。

表 19-4 UART 寄存器概览

偏移地址	寄存器名	寄存器描述	复位值
0x000	UART_CR	UART控制寄存器	0x00000000
0x004	UART_ISR	UART中断状态寄存器	0x00000000
0x008	UART_BRR	UART波特率配置寄存器	0x00000341
0x00c	UART_TDR	UART发送数据寄存器	0x00000000
0x010	UART_RDR	UART接收数据寄存器	0x00000000

19.4.1 UART 控制寄存器（UART_CR）

偏移地址：0x000
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HALF_DUP	MSB_FIRST	TX_DMAEN	RX_DMAEN
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	IDLE_IE	ERR_IE	TXE_IE	TXC_IE	RXNE_IE	PARITY_SEL	PARITY_EN	STOP_LEN	DATA_LEN[1:0]		TX_EN	RX_EN	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:20	Res.	保留，保持为复位值。
19	HALF_DUP	单线半双工模式（Half-duplex single-wire mode） 0：正常工作模式。 1：选择半双工模式。
18	MSB_FIRST	最高有效位在前（Most significant bit first） 0：在起始位之后，发送/接收数据从 LSB 开始。 1：在起始位之后，发送/接收数据从 MSB 开始。
17	TX_DMAEN	发送数据 DMA 使能（Transmit DMA enable） 0：禁止 DMA 发送请求。 1：使能 DMA 发送请求。UART 使能有效后，若 UART_ISR.TXE=1，则产生 DMA 发送请求。
16	RX_DMAEN	接收数据 DMA 使能（Receive DMA enable） 0：禁止 DMA 接收请求。 1：使能 DMA 接收请求。UART 使能有效后，若 UART_ISR.RXNE=1，则产生 DMA 接收请求。
15:13	Res.	保留，保持为复位值。
12	IDLE_IE	空闲帧检测中断使能（Idle interrupt enable） 0：禁止 IDLE=1 产生中断。 1：使能 IDLE=1 产生中断。
11	ERR_IE	错误中断使能（Error interrupt enable） 0：禁止错误产生中断。 1：使能错误产生中断。 注：错误包括 OVRR_ERR=1 或 FRAME_ERR=1 或 PARITY_ERR=1。
10	TXE_IE	发送缓冲空中断使能（Transmit buffer empty interrupt enable） 0：禁止 TXE=1 产生中断。 1：使能 TXE=1 产生中断。
9	TXC_IE	发送完成中断使能（Transmit complete interrupt enable） 0：禁止 TXC=1 产生中断。 1：使能 TXC=1 产生中断。
8	RXNE_IE	接收缓冲非空中断使能（Recept buffer not empty interrupt enable） 0：禁止 RXNE=1 产生中断。

		1：使能 RXNE=1 产生中断。
7	PARITY_SEL	奇偶校验选择（Parity selection） 0：选用偶校验。 1：选用奇校验。
6	PARITY_EN	奇偶校验控制使能（Parity control enable） 0：禁止奇偶校验控制。 1：使能奇偶校验控制。
5	STOP_LEN	停止位（Stop bit length） 0：1 个停止位。 1：2 个停止位。
4:3	DATA_LEN[1:0]	数据位长（Data bit length） 00：数据的长度为 6 位。 01：数据的长度为 7 位。 10：数据的长度为 8 位。 11：数据的长度为 9 位。
2	TX_EN	数据发送使能（Transmit enable） 0：禁止 UART 数据发送。 1：使能 UART 数据发送。
1	RX_EN	数据接收使能（Receive enable） 0：禁止 UART 数据接收。 1：使能 UART 数据接收。
0	EN	UART 使能（Enable） 0：关闭 UART。 1：开启 UART。

19.4.2 UART 中断状态寄存器（UART_ISR）

偏移地址：0x004
复位值：0x00000006

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDLE	OVRR_ERR	FRAME_ERR	PARITY_ERR	TXE	TXC	RXNE
									w1c	w1c	w1c	w1c	w1c	w1c	w1c

Bit	Field	Description
31:7	Res.	保留，保持为复位值。
6	IDLE	空闲帧检测标志（Idle flag） 0：未检测到空闲帧。

		1：检测到空闲帧。 此标志由硬件置 1 和清零。
5	OVRR_ERR	溢出错误（Overrun error） 0：未发生上溢。 1：发生溢出。 此标志由硬件置 1，由软件写 1 清零。
4	FRAME_ERR	传输帧格式错误（Frame error） 0：未发生错误。 1：发生错误。 此标志由硬件置 1，由软件写 1 清零。
3	PARITY_ERR	奇偶检验错误（Parity error） 0：未发生错误。 1：发生错误。 此标志由硬件置 1，由软件写 1 清零。
2	TXE	发送缓冲区为空（Transmit buffer empty） 0：发送缓冲区非空。 1：发送缓冲区为空。 此标志由硬件置 1，由软件写 1 清零；或软件写 UART_TDR 寄存器时硬件自动清零。
1	TXC	发送完成（Transfer complete） 0：发送数据正在进行。 1：发送数据完成，或正在等待数据发送。 此标志由硬件置 1，由软件写 1 清零。或软件写 UART_TDR 寄存器时硬件自动清零。
0	RXNE	接收缓冲非空（Receive buffer not empty） 0：接收缓冲为空。 1：接收缓冲非空。 此标志由硬件置 1，由软件写 1 清零；或软件读 UART_RDR 寄存器时硬件自动清零。

19.4.3 UART 波特率寄存器（UART_BRR）

偏移地址：0x008
复位值：0x00000341

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

Bit	Field	Description
31:20	Res.	保留，保持为复位值。
19:16	DIV_FRAC [3:0]	分频值小数部分（Frequency division fraction）。
15:0	DIV_INT [15:0]	分频值整数部分（Frequency division integer）。

19.4.4 UART 发送数据寄存器（UART_TDR）

偏移地址：0x00C
复位值：0x00000000

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

Bit	Field	Description
31:9	Res.	保留，保持为复位值。
8:0	TDR [8:0]	发送数据寄存器（Transmit data register）。

19.4.5 UART 接收寄存器寄存器（UART_RDR）

偏移地址：0x010
复位值：0x00000000

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

Bit	Field	Description
31:9	Res.	保留，保持为复位值。
8:0	RDR [8:0]	接收数据寄存器（Receiver data register）。

20 串行外设接口(SPI)

20.1 简介

SPI 接口广泛用于 MCU 与外部设备间的通信，支持全双工、单工同步串行的通信方式，软件通过控制片选信号能实现一主多从的通讯网络结构。

20.2 主要特性

SPI 主要特性如下：

- 主或从工作模式
- 全双工同步传输
- 单工同步传输
- 4 至 16 位可配传输数据长度
- 支持软件或硬件的 NSS 片选管理
- 可编程的时钟极性和相位
- 可编程的数据顺序，支持 MSB 在前或 LSB 在前
- 主模式只发送数据支持最高时钟频率为 24MHz
- 主模式接收数据支持最高时钟频率 12MHz
- 从模式支持最高时钟频率为 8MHz
- 支持数据接收发送中断
- 支持传输错误中断
- 支持通过 DMA 收发数据

20.3 SPI 功能说明

20.3.1 SPI 模块框图

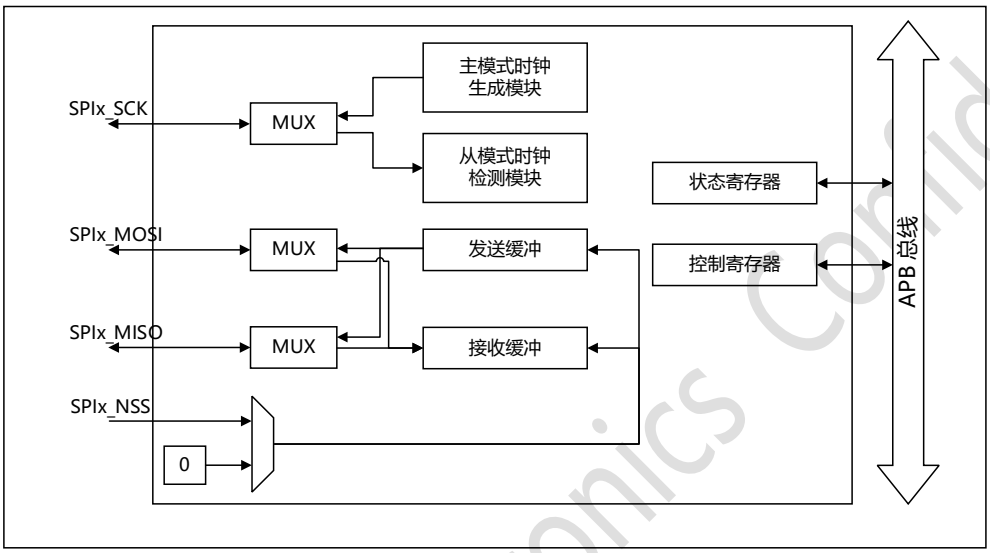

图 20-1 SPI 模块结构框图

20.3.2 SPI 模块功能引脚

表 20-1 SPI 功能引脚说明

引脚	类型	描述
SPIx_MOSI	数字输入输出	主模式下发送数据输出，从模式下接收数据输入。
SPIx_MISO	数字输入输出	主模式下接收数据输入，从模式下发送数据输出。
SPIx_SCK	数字输入输出	主模式下生成时钟输出，从模式下接收时钟输入。
SPIx_NSS	数字输入	从模式下外部输入的 SPI 片选信号，该信号可由软件设置为有效。

20.3.3 SPI 单主单从通信

SPI 在主模式或从模式下，根据全双工、单工通信场景的不同可以使用 2、3 或 4 个引脚进行通信，其中 SPI_SCK 始终由主机发出。

全双工通信

在全双工通信时，主机和从机的两组 SPIx_MOSI 和 SPIx_MISO 数据线连接，通信过程中，数据随主机提供的 SPIx_SCK 时钟边沿同步发送或接收。主机通过 SPIx_MOSI 发送数据，并通过 SPIx_MISO 接收从机数据。当数据帧传输完成时，主机和从机即完成信息交换。

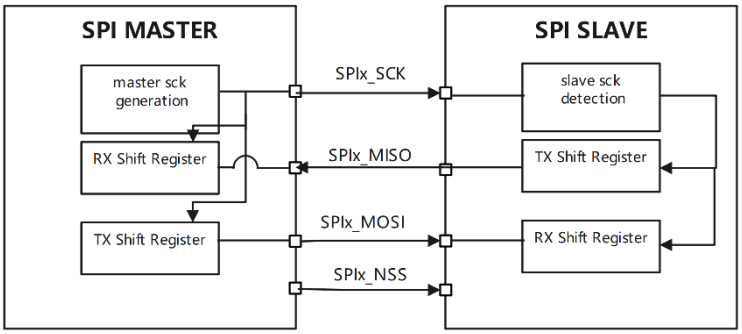

图 20-2 SPI 全双工引脚连接图

注：SPI 模块在主模式下，引脚 SPIx_NSS 的输出通过 GPIO 控制。

单工通信

在单工通信时，通过寄存器 SPI_CR.TX_EN 和 SPI_CR.RX_EN 将 SPI 模块配置为只发送模式或只接收模式，SPI 以单工模式通信。在该配置下，仅使用一条线在主机和从机传输数据。另一个引脚不用于通信，可作标准 GPIO 使用。

只发送模式配置 TX_EN=1，RX_EN=0；只接收模式配置 TX_EN=0，RX_EN=1。

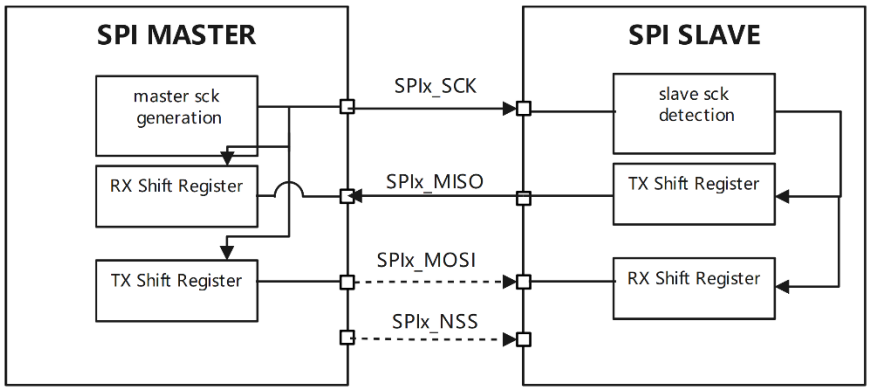

图 20-3 SPI 单工引脚连接图

注：SPI 模块在主模式下，引脚 SPIx_NSS 的输出通过 GPIO 代替输出。

注：单工通信场景，SPIx_MISO 和 SPIx_MOSI 只有一路引脚连接。

20.3.4 SPI 一主多从通信

主模式下，SPI 主机与多个从机连接时，主机使用 GPIO 引脚管理从机的片选信号，主机选中特定从机并输出低电平给该从机的片选输入，只有选中的从机与主机通信。通信完成后，主机输出高电平给该从机的片选输入，再选中其他从机以同样方式通信。

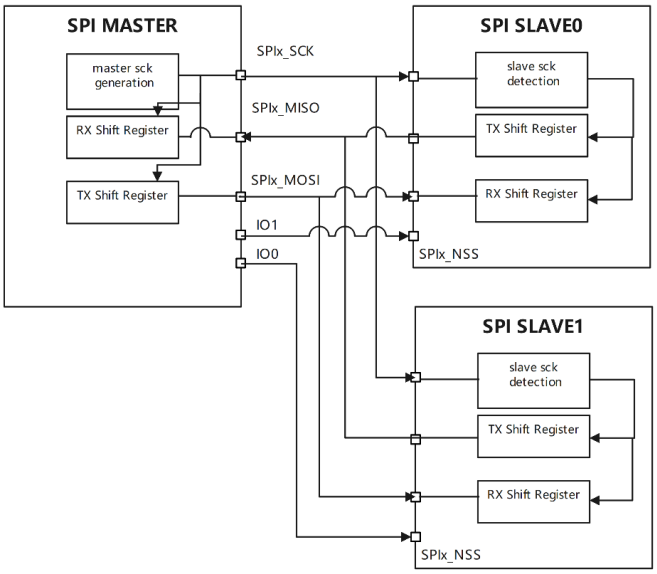

图 20-4 SPI 主模式下与多个从设备连接图

20.3.5 SPI 传输数据格式

SPI 通信过程中，主机和从机都基于串行时钟 SCK 收发数据，为保证数据传输正确，主机和从机必须遵循相同的通信格式。

时钟极性和相位

时钟极性 CPOL 是指未传输数据时，时钟 SCK 的空闲状态。若 CPOL 为 0，SCK 引脚在空闲状态处于低电平。若将 CPOL 为 1，SCK 引脚在空闲状态处于高电平。

若时钟的相位 CPHA 为 0，则第一个数据会在 SCK 的第一个边沿被采样（若将 CPOL 为 1，则为下降沿；若将 CPOL 为 0，则为上升沿）。若 CPHA 为 1，则第一个数据会在 SCK 的第二个边沿被采样（若 CPOL 为 0，则为下降沿；若将 CPOL 为 1，则为上升沿）。

SPI 通信的时钟极性和相位，通过 SPI_CR.CPOL 和 SPI_CR.CPHA 寄存器配置。

注意：软件在配置 SPI_CR.CPOL 或 SPI_CR.CPHA 位之前，应先关闭 SPI 模块（即配置寄存器 SPI_CR.SPI_EN=0）。

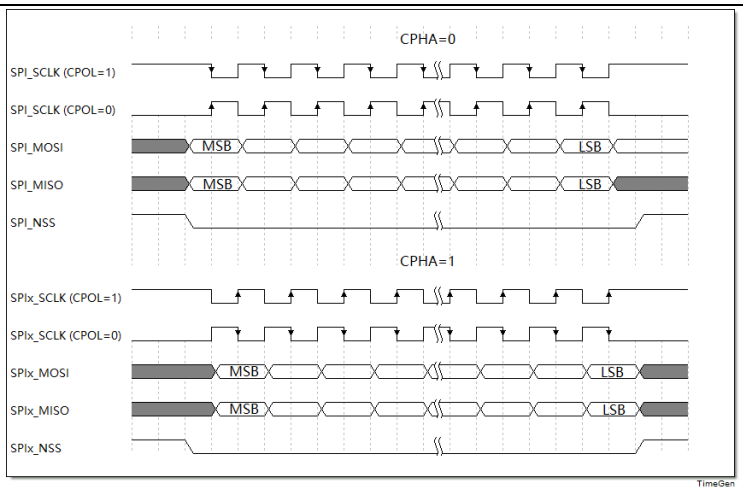

图 20-5 SPI 传输数据时序图

数据帧格式

SPI 收发数据可选择以 MSB 在前或 LSB 在前的方式，通过 SPI_CR.LSB_FIRST 配置。数据帧的长度通过 SPI_CR.DATA_SIZE 位进行选择，数据帧的长度可配置为 4 位到 16 位。在访问 SPI_TDR 和 SPI_RDR 寄存器时，数据帧始终按字节（数据不超过一个字节时）或半字进行右对齐，如下图所示。

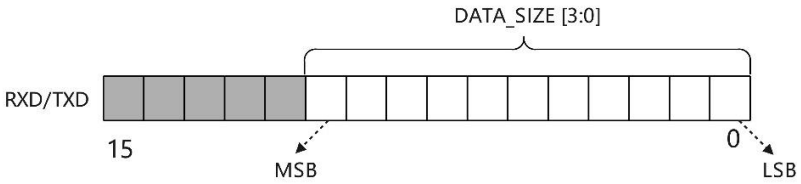

图 20-6 SPI 数据帧格式图

20.3.6 波特率配置

SPI 作主机时，给从机提供时钟 SCK，波特率通过波特率配置寄存器 SPI_BRR 配置，计算公式如下：

$$\text{波特率} = \frac{f_{pclk}}{2 \times (BRR + 1)}$$

式中，f_{pclk}表示 MCU 的 APB 时钟，BRR 是 SPI_BRR 寄存器的配置值。

20.3.7 SPI 初始化

初始化配置流程：

1. 软件配置 SPI_CR.MASTER 选择 SPI 的主从模式。
2. 配置 SPI_CR.CPOL 和 SPI_CR.CPHA 选择 SPI 的极性和相位。
3. 配置 SPI_CR.DATA_SIZE 选择数据长度。
4. 配置 SPI_CR.TX_EN 和 SPI_CR.RX_EN 选择收发数据的使能。
5. 若为主模式，配置 SPI_BRR，设定生成 SCK 的波特率；
   若为从模式，配置 SPI_CR.NSS_SEL 为 0 指定片选信号 SPI_NSS 由外部输入，或配置 NSS_SEL 为 1，内部置低使片选有效。
6. 配置所需中断的使能。

7. 配置 SPI_EN=1，SPI 使能有效。

20.3.8 主模式数据收发

主模式下，若配置 TX_EN 和 RX_EN 都为 1，则为全双工数据传输；若只配置 TX_EN=1 或 RX_EN=1 则为单工数据传输。

主模式数据发送：

1. 初始化配置并使能 SPI 后，配置 GPIO 输出低电平给选中的从机的片选管脚 NSS。
2. 查询 SPI_ISR.TXE 为 1 时，向 SPI_TDR 填入数据，SPI 硬件自动发送 SCK 和数据。
   TXE=1 时，若配置 SPI_CR.TXE_IE=1，则会产生中断。该状态位可写 1 清零或向 SPI_TDR 写数据硬件自动清零。
3. 重复上述过程连续发送。

主模式数据接收：

1. 初始化配置并使能 SPI 后，配置 GPIO 输出低电平给选中的从机的片选管脚 NSS。
2. SPI 只接收数据时，也需要向 SPI_TDR 填入数据以触发输出 SCK 时钟信号。因此，查询 SPI_ISR.TXE=1 时，软件需要向 SPI_TDR 填入无效数据，例如全 0，SPI 硬件自动发送 SCK.TXE=1 时，若配置 SPI_CR.TXE_IE=1，则会产生中断。
   SPI 发送 SCK 时钟的同时，根据配置的时钟极性和相位采集数据，接收数据完成时，SPI_ISR.RXNE=1，该状态位可写 1 清零或从 SPI_RDR 读数据时硬件自动清零。
   RXNE=1 时，若配置 SPI_CR.RXNE_IE=1，则会产生中断。
3. 重复上述过程连续触发数据接收。

20.3.9 从模式数据收发

从模式下，SCK 由外部设备输入，可选择全双工数据传输或单工数据传输。

从模式数据发送：

1. 初始化配置并使能 SPI，且 SPI_CR.TX_EN=1。
2. 查询 SPI_ISR.TXE=1 时，向 SPI_TDR 写入数据，当输入片选为低电平或 SPI_CR.NSS_SEL 为 1，且收到 SCK 时，SPI 硬件自动发送数据。
   TXE=1 时，若配置 SPI_CR.TXE_IE=1，则会产生中断。该状态位可写 1 清零或向 SPI_TDR 写数据时硬件自动清零。
3. 重复上述过程可连续发送数据。
   注：应在主机发送 SCK 前向 SPI_TDR 写入数据。

从模式数据接收：

1. 初始化配置并使能 SPI，且 SPI_CR.RX_EN=1。
   输入片选有效或 SPI_CR.NSS_SEL=1 时，收到 SCK 时，SPI 接收数据，接收完成后 SPI_ISR.RXNE=1，该状态位可写 1 清零或从 SPI_RDR 读数据清零。
   RXNE=1 时，若 SPI_CR.RXNE_IE=1，则会产生中断。
2. 重复该过程连续接收数据。

20.3.10 SPI 状态标志

忙标志（BUSY）
BUSY 标志由硬件置位和清零，BUSY=1 表示 SPI 正在传输数据，BUSY=0 表示当前 SPI 空闲。
下溢标志（UDRR）
从模式下，SPI 发送首位数据，但此时 SPI_RDR 寄存器中没有写入的待发送数据时，下溢标志 SPI_ISR.UDRR 硬件置位。软件可写 1 清零。
下溢发生时，若 SPI_CR.UDRR_IE=1 则会产生中断。
上溢错误标志（OVRR_ERR）
主或从模式下，SPI 在 SPI_ISR.RXNE=1 时，若再次接收到新数据，则 SPI_ISR.OVRR_ERR 被硬件置位，发生上溢错误时。此时 SPI_RDR 中数据不变，新接收的数据丢失。上溢发生时，若配置 ERR_IE=1，则会产生中断。
故障错误标志（FAULT_ERR）
从模式下，SPI 正在收发数据的过程中，NSS 信号发生 0 到 1 的跳变，会导致 SPI 数据传输故障，SPI_ISR.FAULT_ERR 被硬件置位。
发生故障错误时，若 ERR_IE=1，则会产生中断。

20.3.11 SPI 基于 DMA 操作

SPI 的主从模式均支持 DMA 操作，有关 DMA 的配置请参见第 8 章：直接存储器访问控制器（DMA）。
通过 DMA 发送数据
配置 DMA 发送使能 SPI_CR.TX_DMAEN=1，SPI 使能有效后，若 SPI_ISR.TXE=1，则数据将由 DMA 配置的 SRAM 区写入 SPI_TDR 寄存器。
通过 DMA 接收数据
配置 DMA 接收使能 SPI_CR.RX_DMAEN=1，SPI 使能有效后，若 SPI_ISR.RXNE=1，则 SPI_RDR 的数据将由 DMA 读取并存入 SRAM 中。

20.3.12 SPI 中断说明

SPI 支持发送缓冲空中断、接收缓冲非空中断、错误中断、下溢中断。

表 20-2 SPI 中断请求说明

中断事件	中断标志	中断使能	清零方式
发送缓冲空中断	TXE	TXE_IE	中断标志写 1 清零 向 SPI_TDR 写数据时硬件自动清零
接收缓冲非空中断	RXNE	RXNE_IE	中断标志写 1 清零 读 SPI_RDR 时硬件自动清零
错误中断	FAULT_ERR OVRR_ERR	ERR_IE	对应错误中断标志写 1 清零
下溢状态标志	UDRR	UDRR_IE	中断标志写 1 清零

20.3.13 SPI 低功耗特性

表 20-3 SPI 低功耗模式特性

模式	功能描述
睡眠（SLEEP）	该模式下 CPU 工作时钟将会关闭，SPI 能够正常使用，并且 SPI 中断能够退出睡眠模式。
深度睡眠（DEEP_SLEEP）	该模式下 CPU 工作时钟将会关闭，SPI 工作时钟的开关将会由 RCC 模块中的 RCC_SLEEP_CR 寄存器的配置决定。 若深度睡眠模式下 SPI 配置成开启时钟，则 SPI 将正常工作，SPI 中断能够退出深度睡眠模式。 若深度睡眠模式下 SPI 配置成关闭时钟，则应先关闭 SPI 使能，再进入深度睡眠模式。
停止（STOP）	该模式下 CPU 和 SPI 的工作时钟都会关闭，SPI 模块内控制寄存器的内容会保持。 进入该模式前，应先关闭 SPI 使能，再进入该模式。

20.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字(16 位)、字(32 位）访问。

表 20-4 SPI 寄存器概览

偏移地址	寄存器名	寄存器描述	复位值
0x00	SPI_CR	SPI控制寄存器	0x00000700
0x04	SPI_ISR	SPI中断状态寄存器	0x00000002
0x08	SPI_BRR	SPI波特率配置寄存器	0x00000000
0x0C	SPI_TDR	SPI 发送数据寄存器	0x00000000
0x10	SPI_RDR	SPI接收数据寄存器	0x00000000

20.4.1 SPI 控制寄存器（SPI_CR）

偏移地址：0x000
复位值：0x00000700

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NSS_SEL	LSB_FIRST	UDRR_IE	ERR_IE	TXE_IE	RXNE_IE
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DATA_SIZE[3:0]				TX_DMAEN	RX_DMAEN	TX_EN	RX_EN	CPOL	CPHA	MASTER	EN
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:22	Res.	保留，保持为复位值。
21	NSS_SEL	内部从机选择（Internal slave select） SPIx_NSS 的输入由片内接低电平，并忽略 SPIx_NSS 引脚的 I/O 值，置 1 有效。
20	LSB_FIRST	LSB 在前选择（LSB first） 0：发送/接收数据时 MSB 在前。 1：发送/接收数据时 LSB 在前。
19	UDRR_IE	下溢中断使能（Underrun interrupt enable） 0：禁止 UDRR=1 产生中断。 1：使能 UDRR=1 产生中断。
18	ERR_IE	SPI 传输错误中断使能（Error interrupt enable） 0：禁止 OVRR_ERR=1 或 FAULT_ERR=1 产生中断。 1：使能 OVRR_ERR=1 或 FAULT_ERR=1 产生中断。
17	TXE_IE	发送缓冲空中断使能（Transmit buffer empty interrupt enable） 0：禁止 TXE=1 产生中断。 1：使能 TXE=1 产生中断。
16	RXNE_IE	接收缓存非空中断使能（Receive buffer not empty interrupt enable） 0：禁止 RXNE=1 产生中断。 1：使能 RXNE=1 产生中断。
15:12	Res.	保留，保持为复位值。
11:8	DATA_SIZE [3:0]	数据长度（Data size） 0000：保留。 0001：保留。 0010：保留。 0011：4 位。 0100：5 位。 0101：6 位。 0110：7 位。 0111：8 位。 1000：9 位。 1001：10 位。 1010：11 位。 1011：12 位。 1100：13 位。 1101：14 位。 1110：15 位。 1111：16 位。
7	TX_DMAEN	发送 DMA 使能（Transmit DMA enable） 0：禁止 DMA 发送请求。 1：使能 DMA 发送请求。SPI 使能有效后，若 SPI_ISR.TXE=1，则产生 DMA 发送请求。
6	RX_DMAEN	接收 DMA 使能（Receive DMA enable）

		0：禁止 DMA 接收请求。 1：使能 DMA 接收请求。SPI 使能有效后，若 SPI_ISR.RXNE=1，则产生 DMA 接收请求。
5	TX_EN	数据发送使能（Transmit enable） 0：禁止 SPI 数据发送。 1：使能 SPI 数据发送。
4	RX_EN	数据接收使能（Receive enable） 0：禁止 SPI 数据接收。 1：使能 SPI 数据接收。
3	CPOL	时钟极性（Clock polarity） 0：空闲状态时，SCK 默认电平为低电平。 1：空闲状态时，SCK 默认店铺为高电平。
2	CPHA	时钟相位（Clock phase） 0：从第一个时钟边沿开始采样数据。 1：从第二个时钟边沿开始采样数据。
1	MASTER	主从模式选择（Master selection） 0：SPI 工作在从模式。 1：SPI 工作在主模式。
0	EN	SPI 工作使能（SPI enable） 0：禁止 SPI。 1：使能 SPI。

20.4.2 SPI 中断状态寄存器（SPI_ISR）

偏移地址：0x004
复位值：0x00000002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	UDRR	FAULT_ERR	OVRR_ERR	TXE	RXNE
										ro	w1c	w1c	w1c	w1c	w1c

Bit	Field	Description
31:6	Res.	保留，保持为复位值。
5	BUSY	忙标志（Busy flag） 0：SPI 空闲。 1：SPI 正在通信。
4	UDRR	下溢标志（Underrun flag）

		0：未发生下溢。 1：发生下溢。 此标志由硬件置 1，由软件写 1 清零。
3	FAULT_ERR	传输错误（Fault error） 0：未发生模式故障。 1：发生模式故障。 此标志由硬件置 1，由软件写 1 清零。
2	OVRR_ERR	上溢标志（Overrun error） 0：未发生上溢。 1：发生上溢。 此标志由硬件置 1，由软件写 1 清零。
1	TXE	发送缓冲为空（Transmit buffer empty） 0：发送缓冲非空。 1：发送缓冲为空。 此标志由硬件置 1，由软件写 1 清零；或软件写 SPI_TDR 寄存器时硬件自动清零。
0	RXNE	接收缓冲非空（Receive buffer not empty） 0：接收缓冲为空。 1：接收缓冲非空。 此标志由硬件置 1，由软件写 1 清零；或软件读 SPI_RDR 寄存器时硬件自动清零。

20.4.3 SPI 波特率配置寄存器（SPI_BRR）

偏移地址：0x008
复位值：0x00000000

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

Bit	Field	Description
31:8	Res.	保留，保持为复位值。
7:0	BRR [7:0]	波特率配置寄存器(Baud rate register)， 00000001：$f_{PCLK}/4$ 00000010：$f_{PCLK}/6$ … … 11111110：$f_{PCLK}/510$ 11111111：$f_{PCLK}/512$ 注：主模式只发送数据时，BRR 最小配置为 1； 主模式接收数据时，BRR 最小配置为 3； 从模式下，BRR 最小配置为 5。

20.4.4 SPI 发送数据寄存器（SPI_TDR）

偏移地址：0x00C
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，保持为复位值。
15:0	TDR [15:0]	发送数据寄存器（Transmit data register）。

20.4.5 SPI 接收数据寄存器（SPI_RDR）

偏移地址：0x010
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，保持为复位值。
15:0	RDR [15:0]	接收数据寄存器（Received data register）。

21 控制器局域网（CAN）

21.1 简介

控制器局域网（CAN, Controller Area Network）协议是基于非归零码的异步串行通信协议，能安全有效地支持分布式实时控制，用于汽车、工业应用等复杂环境中的快速稳健通信。

21.2 主要特性

- 支持 CAN 协议的 2.0A 和 2.0B
- 扩展的接收缓冲器
- 同时支持 11 位和 29 位的 ID 识别码
- 速率可达 1Mbits/s
- 支持报文接收的硬件筛选
- 支持单次发送报文（非自动重发）
- 支持 BasicCAN、PeliCAN 两种工作模式
- PeliCAN 模式提供扩展功能：
  - 可读写的错误计数器
  - 可编程的错误警告阈值
  - 记录最近一次通讯错误码
  - 仲裁丢失中断
  - 只听模式（被动接收报文且无响应标志）
  - 自接收模式
  - 接收报文的硬件筛选

21.3 功能说明

CAN 协议用于实现多节点间的可靠数据传输，CAN 节点是指 CAN 总线上的控制器设备，各节点在总线上传输报文进行通信，如下图所示。

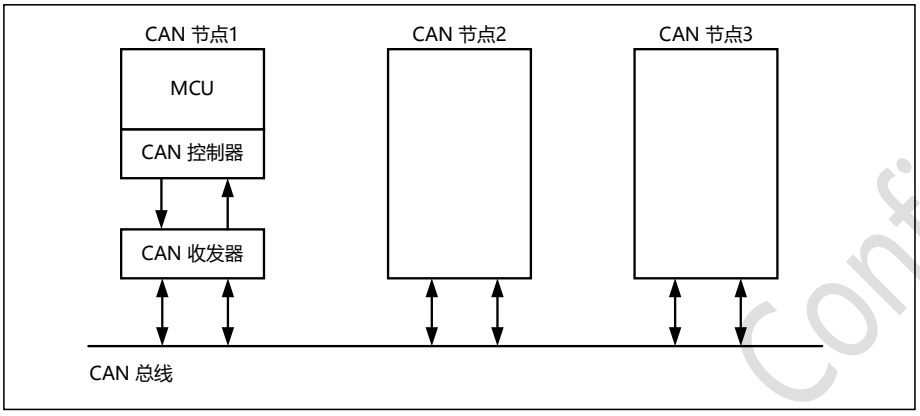

图 21-1 CAN 总线节点拓扑图

21.3.1 CAN 控制器结构框图

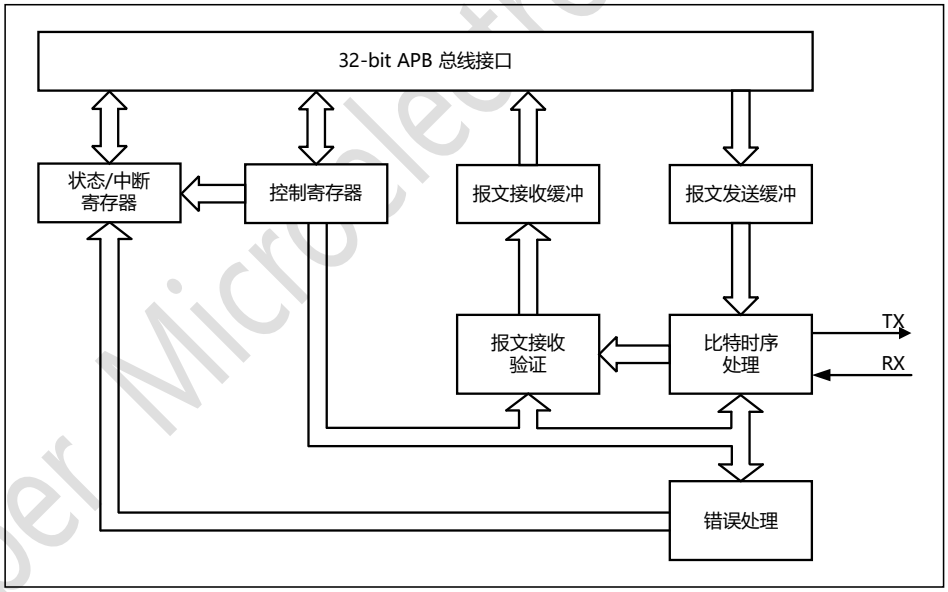

图 21-2 CAN 控制器结构框图

21.3.2 CAN 控制器工作模式选择

CAN 控制器的工作模式分为 BasicCAN 模式和扩展的 PeliCAN 模式。
系统复位时，默认选择的 BasicCAN 模式。能处理 CAN 2.0A 协议的各类型帧。
复位模式下，可更改配置选择为 PeliCAN 模式，能处理 CAN 2.0B 协议的各类型帧，并且提供了一些增强的扩展功能，包括：

- 标准帧和扩展帧的收发
- 可读写的错误计数器
- 可编程的错误警告阈值
- 记录最近一次通讯错误码
- 仲裁丢失中断
- 只听模式（被动接收报文且无响应标志）
- 自接收模式

21.3.3 BasicCAN 寄存器功能表

BasicCAN 模式下，CAN 可配置为复位模式和工作模式。各偏移地址的寄存器功能及读写属性如下表所示。

表 21-1 BasicCAN 模式下寄存器表

偏移地址	功能	操作模式		复位模式	
		读	写	读	写
0x000	控制	控制	控制	控制	控制
0x004		0x000000FF	命令	0x000000FF	命令
0x008		状态	-	状态	-
0x00C		中断	-	中断	-
0x010		0x000000FF	-	验证代码	验证代码
0x014		0x000000FF	-	验证掩码	验证掩码
0x018		0x000000FF	-	总线时序0	总线时序0
0x01C		0x000000FF	-	总线时序1	总线时序1
0x020		0x00000000	-	0x00000000	-
0x024		0x00000000	-	0x00000000	-
0x028	发送缓冲	识别码（10 ~ 3）	识别码（10 ~ 3）	0x000000FF	-
0x02C		识别码（2 ~ 0） RTR位 和 DLC码	识别码（2 ~ 0） RTR位 和 DLC码	0x000000FF	-
0x030		数据1	数据1	0x000000FF	-
0x034		数据2	数据2	0x000000FF	-
0x038		数据3	数据3	0x000000FF	-
0x03C		数据4	数据4	0x000000FF	-
0x040		数据5	数据5	0x000000FF	-
0x044		数据6	数据6	0x000000FF	-
0x048		数据7	数据7	0x000000FF	-
0x04C		数据8	数据8	0x000000FF	-
0x050	接收缓冲	识别码（10 ~ 3）	识别码（10 ~ 3）	识别码（10 ~ 3）	识别码（10 ~ 3）
0x054		识别码（2 ~ 0） RTR位 和 DLC码	识别码（2 ~ 0） RTR位 和 DLC码	识别码（2 ~ 0） RTR位 和 DLC码	识别码（2 ~ 0） RTR位 和 DLC码
0x058		数据1	数据1	数据1	数据1
0x05C		数据2	数据2	数据2	数据2
0x060		数据3	数据3	数据3	数据3
0x064		数据4	数据4	数据4	数据4
0x068		数据5	数据5	数据5	数据5
0x06C		数据6	数据6	数据6	数据6
0x070		数据7	数据7	数据7	数据7
0x074		数据8	数据8	数据8	数据8
0x078		0x00000000	-	0x00000000	-
0x07C	模式控制	工作模式		工作模式	工作模式

21.3.4 PeliCAN 模式寄存器功能表

工作模式寄存器中的 CAN_MODE 寄存器用于选择 BasicCAN 或 PeliCAN，CAN_MODE=0 时为 BasicCAN，CAN_MODE=1 时为 PeliCAN。
PeliCAN 模式下，CAN 可配置为复位模式和工作模式。各偏移地址的寄存器功能及读写属性如下表所示。

表 21-2 PeliCAN 模式下寄存器表

偏移地址	操作模式				复位模式	
	读		写		读	写
0x000	模式		模式		模式	模式
0x004	0x00000000		命令		0x00000000	命令
0x008	状态		-		状态	-
0x00C	中断		-		中断	-
0x010	中断使能		-		中断使能	中断使能
0x014	0x00000000		-		0x00000000	-
0x018	总线时序0		-		总线时序0	总线时序0
0x01C	总线时序1		-		总线时序1	总线时序1
0x020	0x00000000		-		0x00000000	-
0x024	0x00000000		-		0x00000000	-
0x028	0x00000000		-		0x00000000	-
0x02C	仲裁丢失记录		-		仲裁丢失记录	-
0x030	错误码记录		-		错误码记录	-
0x034	错误警告阈值		-		错误警告阈值	错误警告阈值
0x038	RX错误计数器		-		RX错误计数器	RX错误计数器
0x03C	TX错误计数器		-		TX错误计数器	TX错误计数器
0x040	RX 帧信息 SFF	RX 帧信息 EFF	TX 帧信息 SFF	TX 帧信息 EFF	验收代码0	验收代码0
0x044	RX 识别码1	RX 识别码1	TX 识别码1	TX 识别码1	验收代码1	验收代码1
0x048	RX 识别码2	RX 识别码2	TX 识别码2	TX 识别码2	验收代码2	验收代码2
0x04C	RX 数据1	RX 识别码3	TX 数据1	TX 识别码3	验收代码3	验收代码3
0x050	RX 数据2	RX 识别码4	TX 数据2	TX 识别码4	验收掩码0	验收掩码0
0x054	RX 数据3	RX 数据1	TX 数据3	TX 数据1	验收掩码1	验收掩码1
0x058	RX 数据4	RX 数据2	TX 数据4	TX 数据2	验收掩码2	验收掩码2
0x05C	RX 数据5	RX 数据3	TX 数据5	TX 数据3	验收掩码3	验收掩码3
0x060	RX 数据6	RX 数据4	TX 数据6	TX 数据4	0x00000000	-
0x064	RX 数据7	RX 数据5	TX 数据7	TX 数据5	0x00000000	-
0x068	RX 数据8	RX 数据6	TX 数据8	TX 数据6	0x00000000	-
0x06C	-	RX 数据7	-	TX 数据7	0x00000000	-
0x070	-	RX 数据8	-	TX 数据8	0x00000000	-
0x074	接收信息计数器		-		接收信息计数器	-
0x078	0x00000000		-		0x00000000	-
0x07C	工作模式		-		工作模式	工作模式

21.3.5 CAN 控制器使能

CAN 控制器使能关闭时，会复位控制器内部的状态机和控制流、错误计数器等寄存器，使硬件处于初始

化状态，不能向总线收发报文，也不产生任何中断；同时，软件能在复位模式下配置各项控制参数，如时序寄存器、中断使能等寄存器。
CAN 控制器使能有效后，软件能在复位模式下配置各项控制参数，包括时序寄存器、中断使能、错误计数器等寄存器，并且，在操作模式下向总线发送报文或从总线接收报文。

21.3.6 复位模式

复位模式为初始模式，硬件复位或控制器进入离线状态（buf-off），控制器自动进入复位模式，BasicCAN 的复位请求寄存器值为 1，或 PeliCAN 的复位模式寄存器值为 1。
不同工作模式下，MCU 对 CAN 控制器中各组寄存器的访问权限不同，需参考 BasicCAN 和 PeliCAN 的模式寄存器功能表。

21.3.7 操作模式

BasicCAN 将复位请求寄存器配置为 0，或 PeliCAN 将复位模式寄存器配置为 0，CAN 控制器处于操作模式。
1、若上次复位是由软件复位或硬件上电复位，则 CAN 控制器会等待总线空闲标志（11 个隐性位）。
2、若上次复位是由于进入离线状态导致，则需做如下处理将设备重新上线。
BasicCAN：
- 软件复位 CAN 使能，再置位 CAN 使能，将复位请求寄存器配置为 0，则 CAN 会等待总线空闲标志后重新上线，并处于初始状态。

PeliCAN：
- 软件复位 CAN 使能，再置位 CAN 使能，将复位模式寄存器配置为 0，则 CAN 会等待总线空闲标志后重新上线，并处于初始状态。
- 软件写错误计数器为 128，再将复位模式寄存器配置为 0，则 CAN 会等待总线空闲标志后重新上线，并处于错误消极状态。

注：软件可选择在 CAN 控制器离线后，等待一段时间再按上述流程重新将 CAN 上线。
注：CAN 控制器使能的清零和置位，不影响配置寄存器中的信息。

21.3.8 报文发送

报文的发送由 CAN 控制器硬件完成。MCU 配置报文 ID、数据长度和待发送数据；将命令寄存器组的发送请求寄存器置 1，CAN 控制器自动将报文发送至总线。
在 CAN 控制器发送报文的过程中，发送缓冲不可写入。所以在新报文写入发送缓冲之前，软件必须检查状态寄存器的发送缓冲状态，若值为 1，则表示发送缓冲已释放，可写入新值。
仅置位发送请求寄存器，硬件会在报文发送失败时自动重发报文。
单次发送报文
同时配置命令寄存器组的发送请求和中止发送寄存器，会发送一次报文，且不会重发。
发送中止
发送中止寄存器置 1，能中止还未进入发送过程的报文发送请求，正发送报文的进程不会被该命令中止。

21.3.9 报文接收

有效报文

CAN 控制器在总线状态为 0 时，接收总线上的报文，若报文完整且 ID 符合硬件筛选条件，则该报文存入到接收缓冲中，并由接收缓冲状态位和接收中断标志位标明。

查询控制接收

接收有效报文并存入接收缓冲后，状态寄存器组的接收缓冲状态位硬件置 1。软件通过该状态寄存器查看接收缓冲中是否有未读报文。

若接收缓冲状态位为 1，则表示接收缓冲中有未读报文，软件读取报文后，软件将命令寄存器组的释放接收缓冲寄存器置 1，使接收缓冲释放当前报文占据的缓冲空间。

缓冲器中所有报文占的空间都被释放后，接收缓冲状态位硬件清 0。

中断控制接收

软件将接收中断使能配置为 1，CAN 控制器会在接收报文后，产生一个接收中断。在对应的中断服务程序中，应在把数据读取完成后，使用释放接收缓冲命令，来释放当前报文占据的接收缓冲空间。

溢出

溢出情况包括新报文部分存入接收缓冲和新报文无法存入接收缓冲两种情况。

接收缓冲的空闲空间不足时，新报文部分存入。缓冲中的该非完整报文可读时，溢出状态位硬件置 1，溢出中断使能时，则会产生溢出中断。软件需使用清除数据溢出命令，来清除数据溢出状态寄存器。

两种溢出情况之一发生时，硬件将溢出发生状态位立即置 1，该状态位读后清 0；若溢出发生中断使能，则产生溢出发生中断。

21.3.10 ID 过滤

为减少软件对接收报文的筛选工作，CAN 控制器硬件带有 ID 过滤机制。

BasicCAN 的 ID 过滤

BasicCan 通过配置验收代码寄存器和验收掩码寄存器，筛选 ID.10~ID.3 符合要求的报文存放至接收缓冲。

若报文未通过筛选，则硬件将报文舍弃，无需软件参与。

AM.X=1 表示对报文的对应 ID 位的值不关心；AM.X=0 表示，报文的对应 ID 位的值须与 AC.X 的值一致，即接收报文的 ID 满足如下表达式（ID[i+3] == AC[i]）| AM[i]=1，i=7~0。

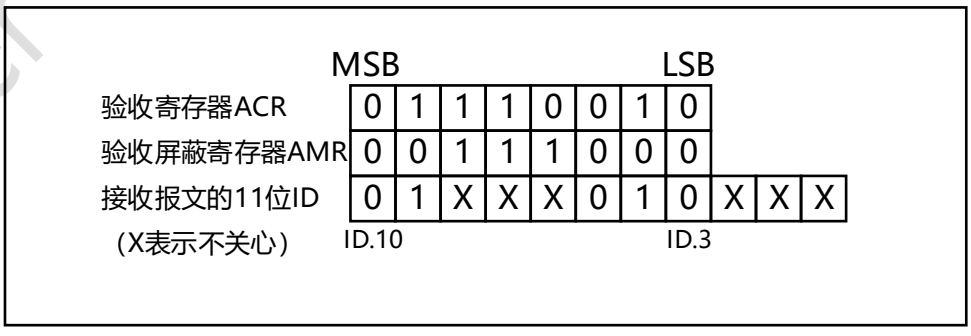

图 21-3 CAN 控制器接收报文 ID 过滤示例

PeliCAN 的 ID 过滤

验收滤波器由验收代码寄存器（ACRn）和验收屏蔽寄存器（AMRn）定义，该组寄存器复位模式下可配。

PeliCAN 具备两种不同的过滤模式，由模式寄存器组的位 3 选择：

- 单滤波器模式（1）
- 双滤波器模式（0）

单滤波器模式

该配置下可定义一个长滤波器（4 字节），滤波器字节和报文字节的对应关系取决于当前接收帧的格式。
标准帧：若接收标准帧格式的报文，验收滤波器比较包括 RTR 位的完整 ID 和前两个数据字节。若由于 RTR 为 1，而导致没有数据字节，或由于设置相应数据长度代码导致没有或只有一个数据字节，信息也会被接收。
注：若 ACR1 和 AMR1 低 4 位不使用，可通过设置 AMR1 寄存器低 4 位为 1 来忽略。

该配置下可定义一个长滤波器（4 字节），滤波器字节和报文字节的对应关系取决于当前接收帧的格式。
标准帧：若接收标准帧格式的报文，验收滤波器比较包括 RTR 位的完整 ID 和前两个数据字节。若由于 RTR 为 1，而导致没有数据字节，或由于设置相应数据长度代码导致没有或只有一个数据字节，信息也会被接收。
注：若 ACR1 和 AMR1 低 4 位不使用，可通过设置 AMR1 寄存器低 4 位为 1 来忽略。

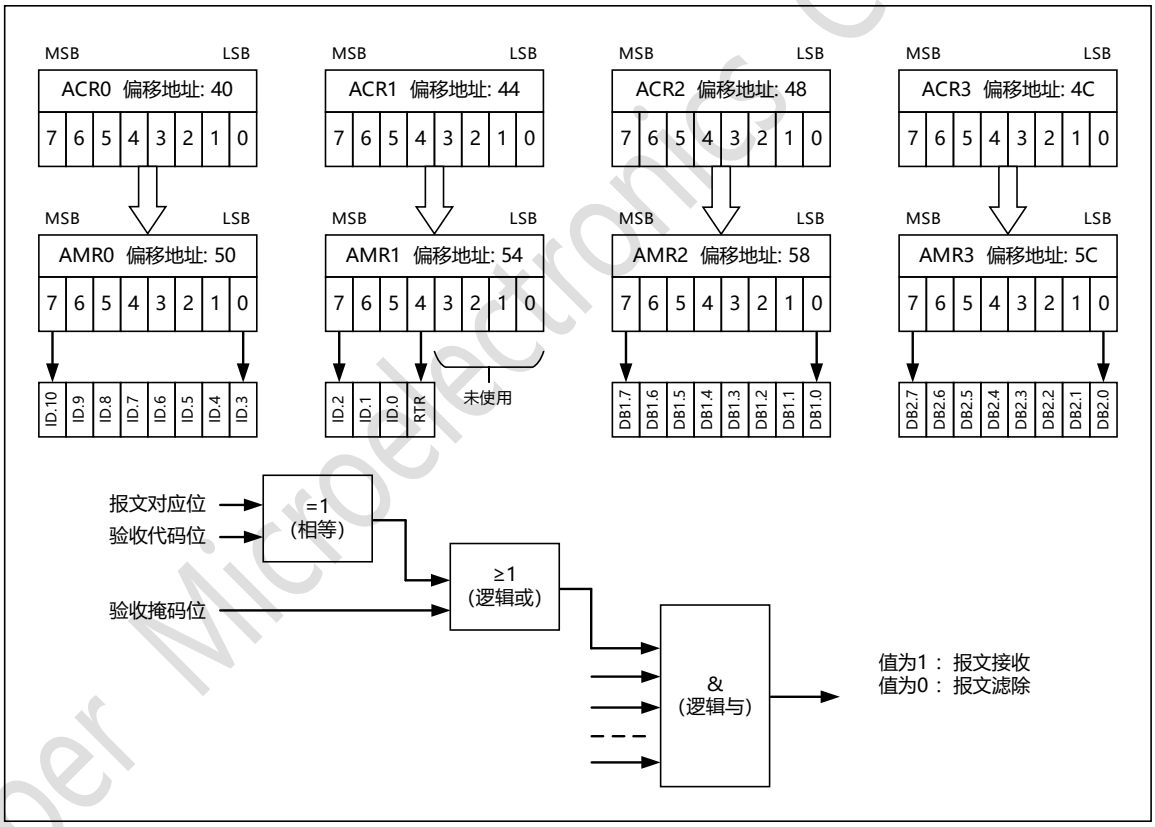

图 21-4 单滤波器模式接收标准帧示意图

扩展帧：若接收扩展帧格式的报文，验收滤波器比较包括 RTR 位的完整 ID。
注：ACR3 和 AMR3 低 2 位不使用，可通过设置 AMR3 寄存器低 4 位为 1 来忽略。

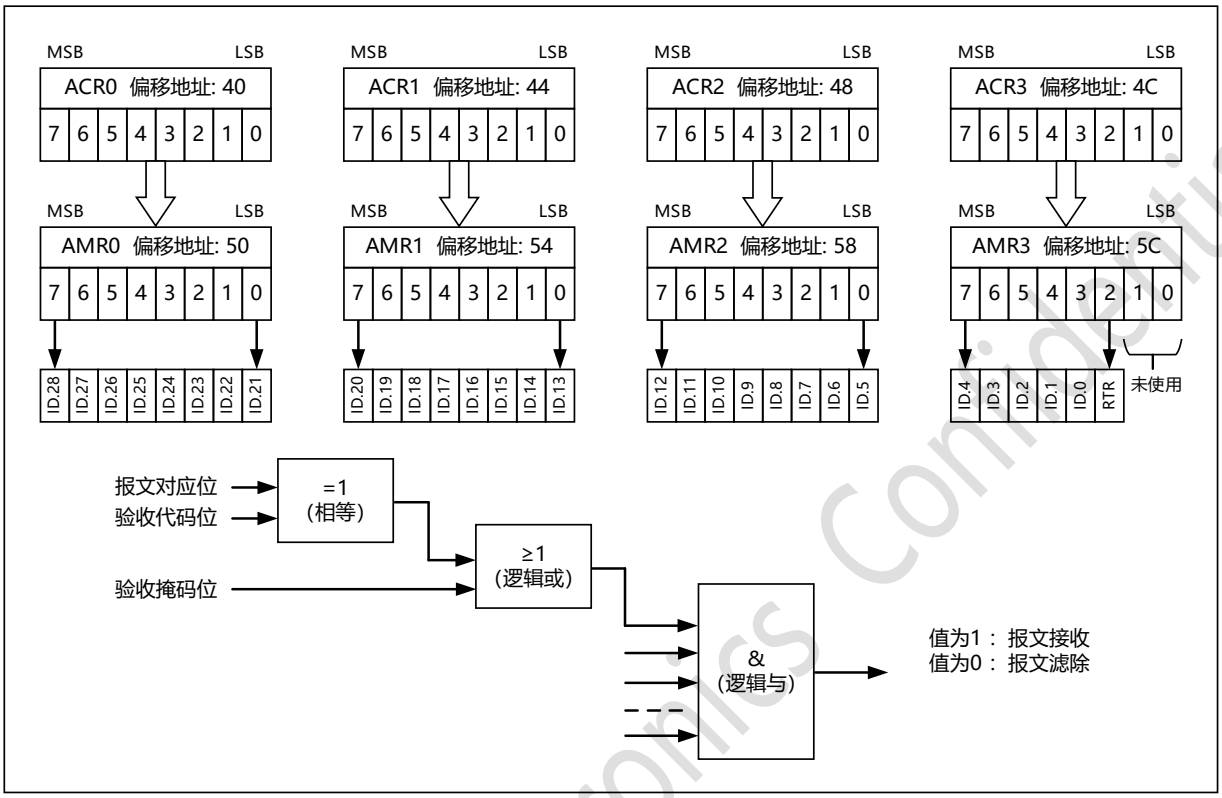

图 21-5 单滤波器模式接收扩展帧示意图

双滤波器模式

该配置可定义两个短滤波器，接收的报文和两个滤波器比较，报文通过任一滤波器，则报文有效被存入接收缓冲。滤波器字节和报文字节的对应关系取决于当前接收帧的格式。

标准帧：如果接收的是标准帧信息，被定义的两个滤波器是不一样的。第一个滤波器比较包括 RTR 位的整个 ID 和第一个数据字节。第二个滤波器只比较包括 RTR 位的整个 ID。

所有位在比较时应至少有一个滤波器表示接收报文，则接收报文。若 RTR 位置 1 或数据长度代码是 0 时，则表示没有数据字节存在，若包括 RTR 位的整个 ID 都被表示接收，信息就可以通过滤波器 1。

如果没有向滤波器请求数据字节过滤，AMR1 和 AMR3 的低四位必须被置 1，即忽略。

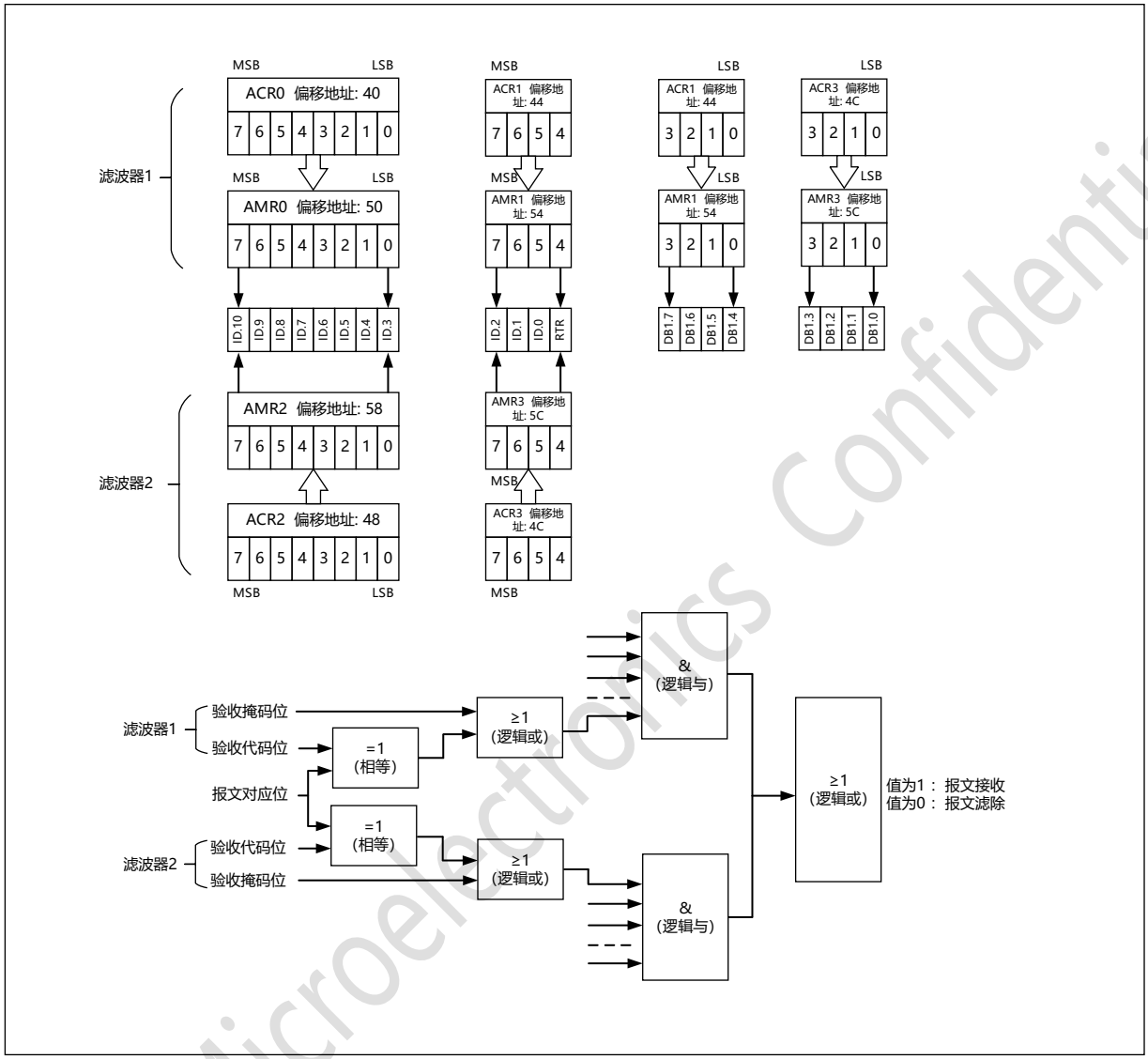

图 21-6 双滤波器模式接收标准帧示意图

扩展帧：如果接收到扩展帧信息，定义的两个滤波器是相同的。两个滤波器都只比较扩展识别码的前两个字节。

所有位在比较时应至少有一个滤波器表示接收报文，则接收报文。

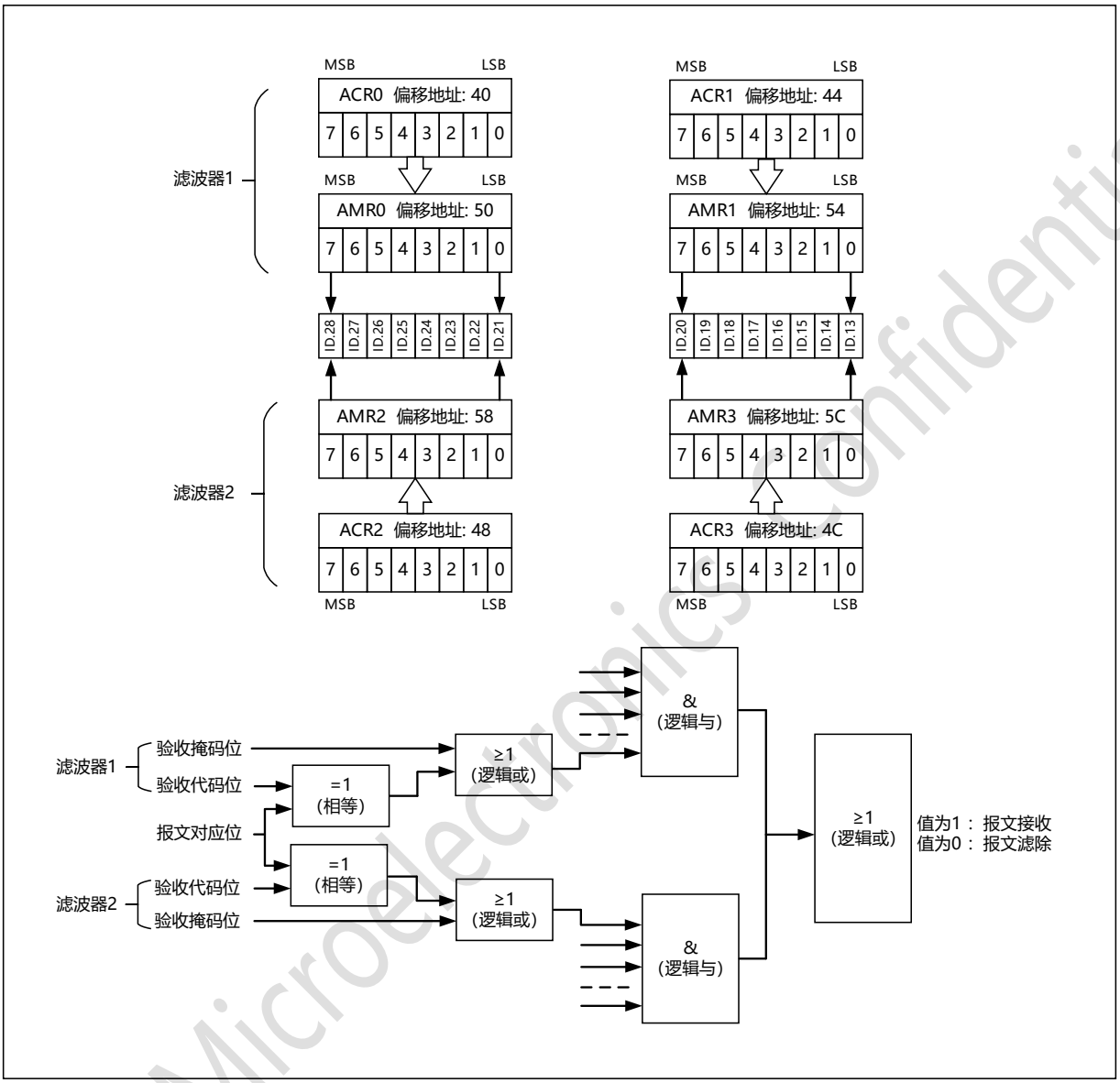

图 21-7 双滤波器模式接收扩展帧示意图

21.3.11 报文存储

BasicCAN 和 PeliCAN 都使用 64 字节的 RXFIFO 接收数据，两种工作模式下分别使用各自的偏移地址访问接收缓冲，接收缓冲是指 RXFIFO 的可访问部分。

BasicCAN 报文存储

BasicCAN 的发送缓冲和接收缓冲结构如下图所示，发送缓冲的偏移地址为 0x28 ~ 0x4C，接收缓冲的偏移地址为 0x50 ~ 0x74。发送缓冲最多可存放 10 个字节，2 个 ID 字节和 8 个数据字节。

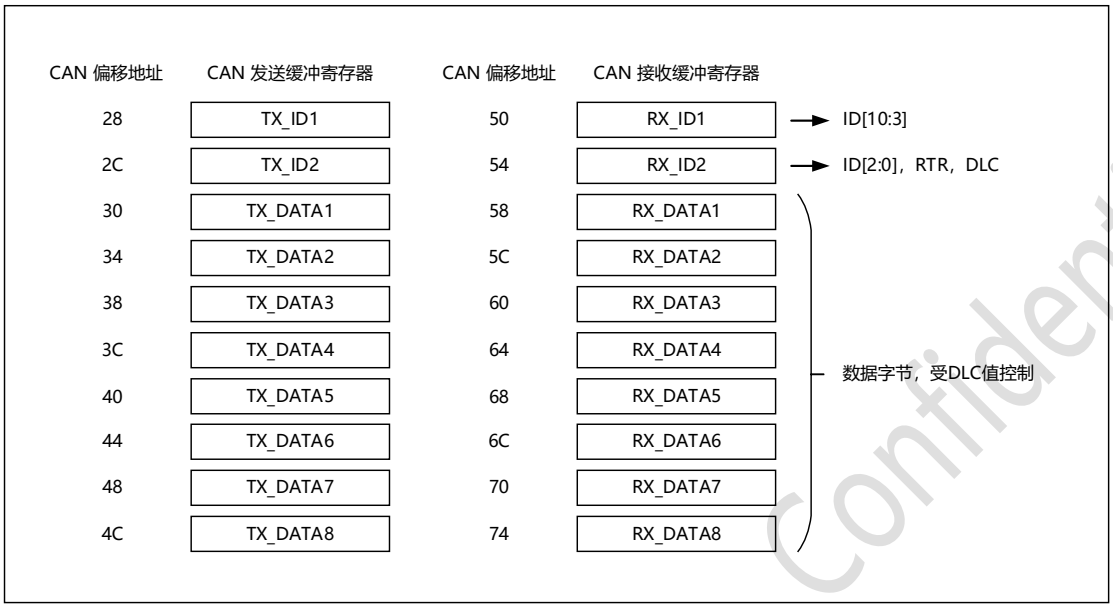

图 21-8 接收和发送缓冲示意图

PeliCAN 报文存储

发送缓冲复用了 CAN 偏移地址的 0x40 ~ 0x70。

发送缓冲最大为 13 个字节，1 字节帧信息，2 或 4 个 ID 字节（标准帧或扩展帧），最多 8 个数据字节，如下图所示。

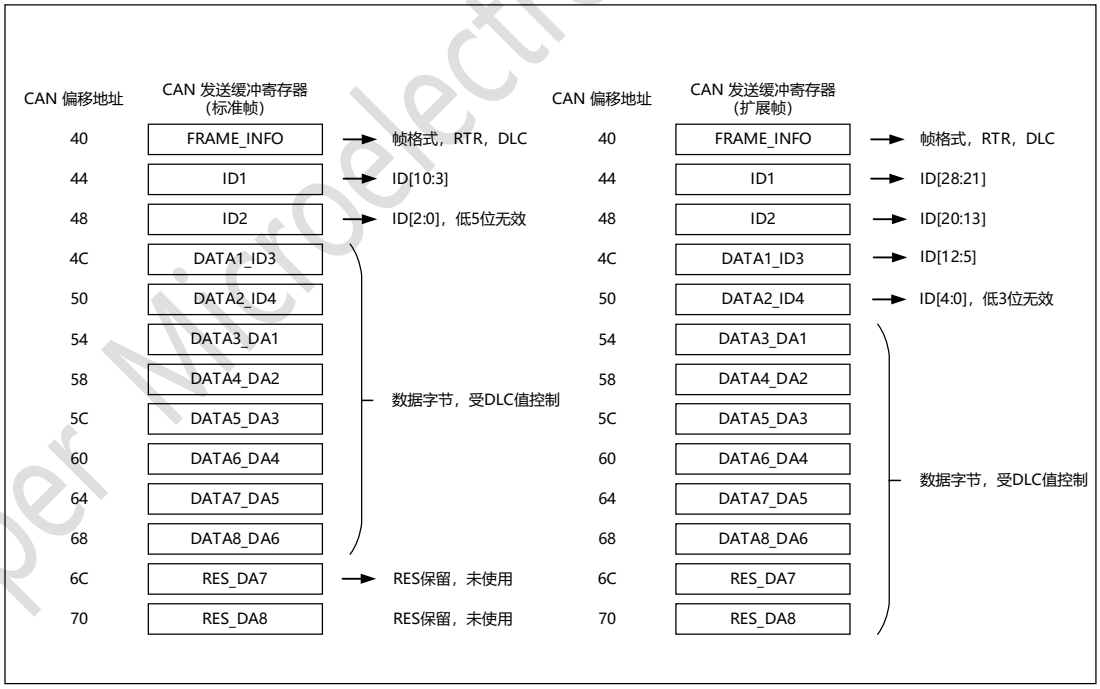

图 21-9 标准帧和扩展帧在发送缓冲器中的偏移地址

接收的报文存储在接收 FIFO 中，接收缓冲器是指接收 FIFO 的可访问部分，如下图所示。每条报文都分为描述区和数据区。

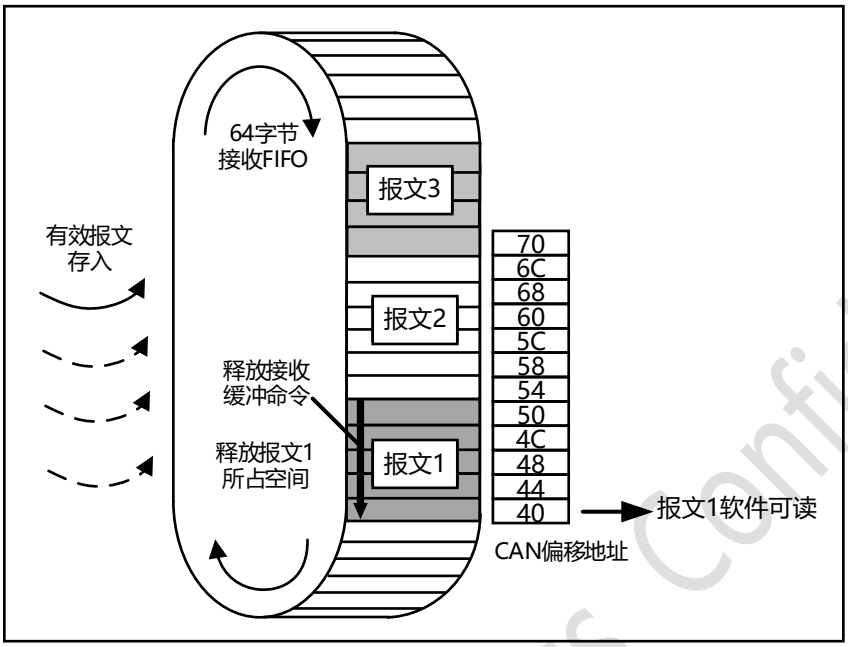

图 21-10 接收报文在接收缓冲的存储方式示意图

接收 FIFO 共有 64 个字节的空间。可累积存储的报文条数取决于数据的长度。如果接收 FIFO 中没有足够的空间来存储新的信息，会导致溢出。

溢出发生状态及中断

溢出发生时，包括接收缓冲存入部分报文，以及未存入报文的溢出情况，溢出发生状态位置 1，软件读清零。溢出发生中断使能有效时，则会产生中断。

溢出状态位及中断

由于溢出导致报文不完整的存入接收 FIFO 中，该不完整报文可读时，溢出状态位置 1，软件通过清除数据溢出命令清除该状态位，溢出中断使能有效时，则会产生中断。

21.3.12 出错管理

CAN 控制器中有发送错误计数器和接收错误计数器，基于错误计数器的值，CAN 控制器可能工作在三种状态：错误主动、错误消极和离线状态。若任一错误计数器小于 128，则 CAN 控制器处于错误激活状态，检测到总线错误会产生错误主动标志位（连续 6 个显性位）。若仍一计数器错误值在 128~255 之间，则 CAN 控制器处于错误消极状态，检测到总线错误会产生错误消极标志位（连续 6 个隐性位）。若错误计数值超过 255，则处于离线状态，并且硬件置位复位请求/复位模式寄存器，使控制器处于复位状态，CAN 控制器对总线无影响。

错误计数器

PeliCAN 模式下，错误计数器可读写。

CAN 使能禁止时，错误计数器值清零。

CAN 使能有效时，软件在复位模式下，可对错误计数器寄存器读写访问，在操作模式，可对错误计数器寄存器读访问。

错误状态

错误计数值超出错误警告阈值时，错误状态硬件置 1。错误警告阈值默认值为 96，PeliCAN 可在复位模式下读写错误阈值寄存器。

错误中断

BasicCAN 模式下，错误中断使能 EIE 有效时，总线错误状态改变或总线状态改变，产生错误中断。
PeliCAN 模式下，错误中断如下：

- 总线错误中断，总线错误中断使能 BEIE 有效时，总线出现错误，产生总线错误中断。
- 错误消极中断，错误消极中断使能 EPIE 有效时，控制器从错误主动变为错误消极状态，或错误消极变为错误主动状态，都产生错误消极中断。
- 错误中断，错误中断使能 EIE 有效时，总线错误状态改变或总线状态改变时，产生错误中断。

错误码记录

PeliCAN 模式下，总线出现错误时，错误码记录寄存器 ECC 会记录错误的相关信息，包括错误类型，错误方向和错误发生的字段，具体细节见 PeliCAN 寄存器概述的 CAN 错误码记录寄存器。

离线恢复

BasicCAN：

- 软件复位 CAN 使能，再置位 CAN 使能，将复位请求寄存器配置为 0，则 CAN 会等待总线空闲标志后重新上线，并处于初始状态。

PeliCAN：

- 软件复位 CAN 使能，再置位 CAN 使能，将复位请求寄存器配置为 0，则 CAN 会等待总线空闲标志后重新上线，并处于初始状态。
- 软件写发送错误计数器为 128，再将复位请求寄存器配置为 0，则 CAN 会等待总线空闲标志后重新上线，并处于错误消极状态。

注：软件可在 CAN 控制器离线后，等待一段时间再按上述流程重新将 CAN 上线。

21.3.13 时序配置

CAN 控制器的时序控制由时序配置寄存器 0 中的 SJW、BRP 寄存器，和时序配置寄存器 1 中的 TSEG2、TSEG1 共同决定。
CAN 的系统时钟由 APB 总线时钟分频生成，CAN 控制器内部的时钟周期为$T_{CAN} = 2 \times T_{APB} \times (BRP + 1)$。
控制器根据总线信号进行同步，以保证采样位置的准确。同步跳转步长为$t_{SJW} = T_{CAN} \times (SJW + 1)$。
总线上 1 位数据的时长为$t_{bit} = T_{CAN} \times (1 + t_{SEG1} + t_{SEG2})$。
采样位置通过 TSEG1 和 TSEG2 配置，$t_{SEG1} = T_{CAN} \times (1 + TSEG1)$，$t_{SEG2} = T_{CAN} \times (1 + TSEG2)$。
对一位数据的采样次数，通过时序配置寄存器 1 的 SAM 寄存器选择，SAM=0 时为单次采样，SAM=1 时为三次采样。

注 1：建议配置值范围，TSEG1>=3，TSEG1>TSEG2，TSEG2>=SJW；
例如：APB 时钟为 96M，CAN 以 1M 速率通讯时，可参考配置 BRP=3, TSEG1=7, TSEG2=4, SJW=4。
注 2：CAN 控制器波特率与总线一致，但通信不成功时，有如下调节方式：1、改变同步跳转步长 SJW 的值，提高同步效率；2、调节 TSEG2 和 TSEG1 的值，选取合适的采样位置；3、降低预分频 BRP 的值，以提升 CAN 控制器内部采样电路的主频，使数据处理更加细粒度，同时增大 TSEG2 和 TSEG1 的值，并保持 CAN 控制器的波特率不变。

21.3.14 命令类型

发送请求：置位有效，发送缓冲中的数据将被发送。

中止发送：置位有效，能取消正准备发送出的报文发送，若报文正在发送则不会被该命令打断。
释放接收缓冲：置位有效，释放最早收到的报文占据的缓冲空间，释放的空间将用于存放新的报文数据。
清除数据溢出：置位有效，清零数据溢出状态位。
PeliCAN 支持额外的自接收请求命令：在自测试模式下，软件发送并接收改报文，即使总线没有其他设备响应也不会报错。
注：中止发送和发送请求同时配置，将只发送一条数据，且只发送一次。

21.3.15 仲裁丢失

总线上多个 CAN 控制器同时发起通讯时，CAN 控制器根据总线信号在报文的仲裁域字段进行仲裁，若仲裁丢失，则 CAN 控制器转为接收状态。若仲裁丢失中断使能有效，则产生仲裁丢失中断。同时，仲裁发生位置被记录于仲裁丢失捕获寄存器。软件读取该值后，控制器才能产生新的仲裁丢失中断，并重新记录仲裁位置。

21.3.16 状态类型

CAN 控制器的状态寄存器包括如下类型：
接收缓冲状态：接收缓冲内可读的报文非空，该位硬件置 1；接收缓冲无可读的报文，该位硬件清 0。
数据溢出状态：溢出导致报文接收不完整，该报文可读时，该位硬件置 1；由清除数据溢出命令清 0。
发送缓冲状态：正在发送数据时，该位为 0，表明发送缓冲内数据不可写；硬件未发送报文时，该位硬件置 1，表明发送缓冲已释放。
发送完成状态位：发送报文成功或准备发送的报文被中止发送命令停止后，该状态位硬件置位为 1；发送请求未处理完成时，该位硬件清 0。
接收状态：正在接收数据时，该位硬件置 1；未接收数据时，硬件清 0。
错误状态：任一错误计数器值值超过阈值 (默认为 96)，硬件置 1；两计数器错误累计值都小于阈值时，硬件置 0。
总线状态：0 表示 CAN 控制器未离线；1 表示 CAN 控制器处于离线状态。
溢出发生状态：CAN 控制器的接收缓冲由于容量限制导致最后一条报文接收不完整，或由于接收缓冲满导致新的报文完全未存入缓冲中，溢出发生时该位立即硬件置 1；该状态位软件读清零。

21.3.17 中断类型

CAN 在两种工作模式下有不同的中断标志，如下所示。所有的中断标志位，软件读该组寄存器时清零。
BasicCAN
BasicCAN 支持的中断包括：接收中断、发送中断、错误中断、溢出中断和溢出发生中断。
接收中断：接收 FIFO 中报文数量非空，且接收中断使能有效，产生接收中断。
发送中断：发送缓冲状态从 0 变为 1，且发送中断使能有效，产生发送中断。
错误中断：错误状态位或总线状态位的变化，且错误中断使能有效，产生错误中断。
溢出中断：接收缓冲中由于溢出导致的不完整报文可读时，数据溢出状态为从 0 变为 1，且溢出中断使能有效，产生溢出中断。
溢出发生中断：接收 FIFO 接收不完整报文，或由于接收缓冲满导致新的报文完全未存入缓冲，且溢出发生中断有效，产生溢出发生中断。

PeliCAN

PeliCAN 支持的中断包括：接收中断、发送中断、错误中断、溢出中断、错误消极中断、仲裁丢失中断、总线错误中断、溢出发生中断。

接收中断：接收缓冲中报文数量非空，若接收中断使能有效，产生接收中断。

发送中断：发送缓冲状态从 0 变为 1，若发送中断使能有效，产生发送中断。

错误警告中断：错误状态位和总线状态位改变，若错误警告中断使能有效，产生错误警告中断。

错误消极中断：控制器从错误主动状态进入错误消极状态，或从错误消极状态转换为错误主动状态时，若错误消极中断使能有效，产生错误消极中断。

仲裁丢失中断：发生仲裁丢失时，CAN 控制器转为接收状态，若仲裁丢失中断使能有效，产生仲裁丢失中断。

总线错误中断：CAN 控制器检测到总线错误，若总线错误中断使能有效，产生总线错误中断。

溢出发生中断：接收缓冲接收不完整报文，或由于接收缓冲满导致新的报文完全未存入缓冲，若溢出发生中断有效，产生溢出发生中断。

21.4 BasicCAN 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字(16 位)、字(32 位）访问。

表 21-3 BasicCAN 寄存器概览

偏移地址	寄存器名	寄存器描述	复位值
0x000	CAN_CR	CAN控制寄存器	0x00000021
0x004	CAN_CMR	CAN命令寄存器	0x00000000
0x008	CAN_SR	CAN状态寄存器	0x0000000C
0x00c	CAN_IR	CAN中断寄存器	0x000000E0
0x010	CAN_ACR	CAN验收代码寄存器	0x00000000
0x014	CAN_AMR	CAN验收掩码寄存器	0x0000000FF
0x018	CAN_BTR0	CAN总线时序寄存器0	0x00000000
0x01C	CAN_BTR1	CAN总线时序寄存器1	0x00000000
0x028	CAN_TX_ID1	CAN发送ID字节1	0x000000XX
0x02C	CAN_TX_ID2	CAN发送ID字节2	0x000000XX
0x030	CAN_TX_DATA1	CAN发送数据字节1	0x000000XX
0x034	CAN_TX_DATA2	CAN发送数据字节2	0x000000XX
0x038	CAN_TX_DATA3	CAN发送数据字节3	0x000000XX
0x03C	CAN_TX_DATA4	CAN发送数据字节4	0x000000XX
0x040	CAN_TX_DATA5	CAN发送数据字节5	0x000000XX
0x044	CAN_TX_DATA6	CAN发送数据字节6	0x000000XX
0x048	CAN_TX_DATA7	CAN发送数据字节7	0x000000XX
0x04C	CAN_TX_DATA8	CAN发送数据字节8	0x000000XX
0x050	CAN_RX_ID1	CAN接收数据ID1	0x000000XX
0x054	CAN_RX_ID2	CAN接收数据ID2	0x000000XX
0x058	CAN_RX_DATA1	CAN接收数据字节1	0x000000XX
0x05C	CAN_RX_DATA2	CAN接收数据字节2	0x000000XX
0x060	CAN_RX_DATA3	CAN接收数据字节3	0x000000XX
0x064	CAN_RX_DATA4	CAN接收数据字节4	0x000000XX
0x068	CAN_RX_DATA5	CAN接收数据字节5	0x000000XX
0x06C	CAN_RX_DATA6	CAN接收数据字节6	0x000000XX
0x070	CAN_RX_DATA7	CAN接收数据字节7	0x000000XX
0x074	CAN_RX_DATA8	CAN接收数据字节8	0x000000XX
0x07C	CAN_EXTEND	CAN工作模式选择寄存器	0x00000000

21.4.1 CAN 控制寄存器(CAN_CR)

偏移地址：0x00
复位值：0x00000021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVRROIE	Res.	EN	Res.	OIE	EIE	TIE	RIE	RR
							rw		rw		rw	rw	rw	rw	rw

Bit	Field	Description
31:9	Res.	保留，必须保持复位值。
8	OVRROIE	溢出发生中断使能（Overrun occur interrupt enable）： 1：使能溢出发生中断。溢出发生时，OVRROI 置位，产生中断。 0：禁止溢出发生中断。 注：溢出包括接收缓冲存入部分报文，和未存入报文的两种溢出情况。
7	Res.	保留，必须保持复位值。
6	EN	CAN 控制器使能（Enable）： 1：使能 CAN 控制器。 0：禁止 CAN 控制器。
5	Res.	保留，必须保持复位值。
4	OIE	溢出中断使能（Overrun interrupt enable）： 1：使能溢出中断。接收缓冲的数据是由溢出导致的不完整数据时，DOI 置位，产生中断。 0：禁止溢出中断。
3	EIE	错误中断使能（Error interrupt enable）： 1：使能错误中断。错误状态或总线状态改变时，EI 置位，产生中断。 0：禁止错误中断。 注：CAN_EN=0 是，不会产生错误中断。
2	TIE	发送中断使能（Transmission interrupt enable）： 1：使能发送中断。报文被成功发送，或发送缓冲可再次访问（中断传输命令有效），TI 中断标志位置位，控制器产生中断。 0：禁止发送中断。
1	RIE	接收中断使能（Receive interrupt enable）： 1：使能接收中断。接收缓冲中有报文数据时，RI 中断标志位置位，产生中断。 0：禁止接收中断。
0	RR	复位模式（Reset request）： 1：复位模式。 0：操作模式。

21.4.2 CAN 命令寄存器(CAN_CMR)

偏移地址：0x04
复位值：0x00000000

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

Bit	Field	Description
31:4	Res.	保留，必须保持复位值。
3	CDO	清除数据溢出（Clear data overrun）： 1：清除接收缓冲中不完整报文所引起的数据溢出状态位。 0：无行为。
2	RRB	释放缓冲空间（Release receive buffer）： 1：释放一条报文所占据的接收缓冲空间。 0：无行为。
1	AT	中断发送（Abort transmission）： 1：计划发送但还未发送的报文，取消发送请求。 0：无行为。
0	TR	发送请求（Transmission request）： 1：一条报文将会被发送。 0：无行为。

21.4.3 CAN 状态寄存器(CAN_SR)

偏移地址：0x08
复位值：0x0000000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVRROS	BS	ES	TS	RS	TCS	TBS	DOS	RBS
							ro	ro	ro	ro	ro	ro	ro	ro	ro

Bit	Field	Description
31:9	Res.	保留，必须保持复位值。
8	OVRROS	溢出发生状态（Overrun occur status）： 1：溢出发生时，该状态位硬件置 1，软件读清零。 0：未发生任何形式的溢出。 注：溢出包括接收缓冲存入部分报文，和未存入报文的两种溢出情况。
7	BS	总线状态（Bus status）： 1：离线，CAN 控制器未处于 CAN 总线中。 0：在线，CAN 控制器处于 CAN 总线中。
6	ES	错误状态（Error status）： 1：报错，任一错误计数器值达到或超过警告阈值。 0：无错误，两个错误计数器值都低于警告阈值。
5	TS	发送状态（Transmit status）： 1：发送，正在发送报文。 0：空闲，未发送报文。
4	RS	接收状态（Receive status）： 1：接收，正在接收报文。 0：空闲，未接收报文。
3	TCS	发送完成状态（Transmitssion complete status）： 1：完成，最近的发送请求成功执行完。 0：未完成，最近的发送请求未执行完。
2	TBS	发送缓冲状态（Transmit buffer status）： 1：释放，软件可以向发送缓冲中写入报文。 0：锁定有报文正在等待发送，软件不该改变发送缓冲中的报文。
1	DOS	数据溢出状态（Data overrun status）： 1：接收缓冲中的报文由于发生溢出导致保存不完整，当该报文可读时，DOS 状态位硬件置 1。 0：表示接收缓冲的可读报文内容完整。 注：需要用清除数据溢出命令将该状态位清零。
0	RBS	接收缓冲状态（Receive buffer status）： 1：非空，一条或多条报文在接收缓冲中可读。 0：空，接收缓冲中没有接收的报文。 注：需要用释放缓冲空间命令清零。

21.4.4 CAN 中断寄存器(CAN_IR)

偏移地址：0x00C
复位值：0x000000E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVRROI	Res.	Res.	Res.	Res.	DOI	EI	TI	RI
							ro					ro	ro	ro	ro

Bit	Field	Description
31:9	Res.	保留，必须保持复位值。
8	OVRROI	溢出发生中断（Overrun occur interrupt）： 1：溢出发生时，若对应中断使能开启，该中断标志置 1。 0：软件读清零。 注：溢出包括接收缓冲存入部分报文，和未存入报文的两种溢出情况。
7:4	Res.	保留，必须保持复位值。
3	DOI	数据溢出中断（Data overrun interrupt）： 1：接收缓冲中存在溢出导致的不完整报文，若对应中断使能开启，该中断标志置 1。 0：软件读清零。
2	EI	错误中断（Error interrupt）： 1：错误状态或总线状态发生改变时，若对应中断使能开启，该中断标志置 1。 0：软件读清零。
1	TI	发送中断（Transmit interrupt）： 1：发送状态位从 0 到 1 时，发送缓冲数据可软件修改，若对应中断使能开启，该中断标志置 1。 0：软件读清零。
0	RI	接收中断（Receive interrupt）： 1：接收缓冲非空时，若对应中断使能开启，该中断标志置 1。 0：软件读清零。

21.4.5 CAN 验收代码寄存器(CAN_ACR)

偏移地址：0x010
复位值：0x00000000

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

Bit	Field	Description
31:9	Res.	保留，必须保持复位值。
7:0	AC	验收代码（Acceptance code）： 接收报文的高 8 位(ID.10 to ID.3)中，被验收掩码标记为与验收代码关联的位，与验收代码(AC.7 to AC.0)的对应位完全一致，则该报文通过滤波器，存入接收缓冲；否则，不存入接收缓冲。

21.4.6 CAN 验收掩码寄存器(CAN_AMR)

偏移地址：0x014
复位值：0x000000FF

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

Bit	Field	Description
31:9	Res.	保留，必须保持复位值。
7:0	AM	验收掩码（Acceptance mask）： 接收报文的高 8 位(ID.10 to ID.3)中，对应 AM 寄存器(AM.7 to AM.0)，AM.X 为 0 表示对应 ID 位关联，AM.X 为 1 表示对应 ID 位不关联（不与 AC.X 进行比较）。

21.4.7 CAN 时序配置寄存器 0(CAN_BTR0)

偏移地址：0x018
复位值：0x00000000

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

Bit	Field	Description
31:9	Res.	保留，必须保持复位值。
9:8	SJW	同步跳转位宽（Synchronization jump width）： CAN 控制器在同步时的跳转位宽。
7:0	BRP	波特率预分频（Baud rate prescaler）： CAN 控制器基于系统时钟先进行预分频，在使用预分频后的时钟计算 Tseg2 和 Tseg1。

21.4.8 CAN 时序配置寄存器 1(CAN_BTR1)

偏移地址：0x01C
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.			SAM	TSEG2[2:0]			TSEG1[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7	SAM	采样（Sample）： 1：总线上信号被采样 3 次。 0：总线上信号被采样 1 次。
6	Res.	保留，必须保持复位值。
5:4	TSEG2	时间字段 2 (Time segment2)： 基于预分频后的时钟，指定时间字段 2 的时长。
3:0	TSEG1	时间字段 1 (Time segment1)： 基于预分频后的时钟，指定时间字段 1 的时长。

21.4.9 CAN 发送缓冲 ID 字节 1(CAN_TX_ID1)

偏移地址：0x028
复位值：0x000000XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TX_ID1[7:0]							
								wo	wo	wo	wo	wo	wo	wo	wo

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	TX_ID1	发送报文的标识符 ID.10 ~ ID.3。

21.4.10 CAN 发送缓冲 ID 字节 2(CAN_TX_ID2)

偏移地址：0x02C
复位值：0x000000XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TX_ID2[2:0]			TX_RTR	TX_DLC[3:0]			
								wo	wo	wo	wo	wo	wo	wo	wo

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:5	TX_ID2	发送报文的标识符 ID.2 ~ ID.0。
4	TX_RTR	发送报文的 RTR (Remote transmission request 远程发送请求) 。
3:0	TX_DLC	发送报文的 DLC (Data length code 数据长度码) ，最大值为 8 数据字节数=8 × DLC.3 + 4 × DLC.2 + 2 × DLC.1 + DLC.0。

21.4.11 CAN 发送数据字节 x 寄存器(CAN_TX_DATAx)(x = 1，2...8)

偏移地址：0x030 ~ 0x04C

复位值：0x000000XX

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

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	TX_DATAx	发送报文的数据，发送数据的数量由发送报文的 DLC 值决定。

21.4.12 CAN 接收缓冲 ID 字节 1（CAN_RX_ID1）

偏移地址：0x050
复位值：0x000000XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RX_ID1[7:0]							
								ro	ro	ro	ro	ro	ro	ro	ro

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	RX_ID1	接收报文的标识符 ID.10 ~ ID.3。

21.4.13 CAN 接收缓冲 ID 字节 2(CAN_RX_ID2)

偏移地址：0x054
复位值：0x000000XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RX_ID2[2:0]			RX_RTR	RX_DLC[3:0]			

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:5	RX_ID2	接收报文的标识符 ID.2 ~ ID.0。
4	RX_RTR	接收报文的 RTR (Remote transmission request 远程发送请求) 。
3:0	RX_DLC	接收报文的 DLC (Data length code 数据长度码)，最大值为 8 数据字节数=8 × DLC.3 + 4 × DLC.2 + 2 × DLC.1 + DLC.0。

21.4.14 CAN 接收数据字节 x 寄存器(CAN_RX_DATAx) (x = 1，2...8)

偏移地址：0x058 ~ 0x074
复位值：0x000000XX

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

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	RX_DATAx	接收报文的数据，接收数据的数量由接收报文的 DLC 值决定。

21.4.15 CAN 工作模式选择寄存器（CAN_EXTEND）

偏移地址：0x07C
复位值：0x00000000

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

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7	CAN_MODE	CAN 工作模式选择： 1：CAN 控制器处于 PeliCan 模式。 0：CAN 控制器处于 BasicCan 模式。
6:0	Res.	保留，必须保持复位值。

21.5 CAN 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字(16 位)、字(32 位）访问。

表 21-4 PeliCAN 寄存器概览

偏移地址	寄存器名	寄存器描述	复位值
0x000	CAN_MOD	CAN模式寄存器	0x00000001
0x004	CAN_CMR	CAN命令寄存器	0x00000000
0x008	CAN_SR	CAN状态寄存器	0x0000000C
0x00C	CAN_IR	CAN中断寄存器	0x00000000
0x010	CAN_IER	CAN中断使能寄存器	0x00000000
0x018	CAN_BTR0	CAN时序配置寄存器0	0x00000000
0x01C	CAN_BTR1	CAN时序配置寄存器1	0x00000000
0x02C	CAN_ALC	CAN仲裁丢失记录寄存器	0x00000000
0x030	CAN_ECC	CAN错误码记录寄存器	0x00000000
0x034	CAN_EWLR	CAN错误警告阈值寄存器	0x00000060
0x038	CAN_RXERR	CAN接收错误计数器	0x00000000
0x03C	CAN_TXERR	CAN发送错误计数器	0x00000000
0x040	CAN_FRAME_INFO	CAN帧信息寄存器	0x000000XX
0x044	CAN_ID1	CAN报文ID字节1	0x000000XX
0x048	CAN_ID2	CAN报文ID字节2	0x000000XX
0x04C	CAN_DATA1_ID3	CAN SFF数据字节1或EFF报文ID字节3	0x000000XX
0x050	CAN_DATA2_ID4	CAN SFF数据字节2或EFF报文ID字节4	0x000000XX
0x054	CAN_DATA3_DA1	CAN SFF数据字节3或EFF数据字节1	0x000000XX
0x058	CAN_DATA4_DA2	CAN SFF数据字节4或EFF数据字节2	0x000000XX
0x05C	CAN_DATA5_DA3	CAN SFF数据字节5或EFF数据字节3	0x000000XX
0x060	CAN_DATA6_DA4	CAN SFF数据字节6或EFF数据字节4	0x000000XX
0x064	CAN_DATA7_DA5	CAN SFF数据字节7或EFF数据字节5	0x000000XX
0x068	CAN_DATA8_DA6	CAN SFF数据字节8或EFF数据字节6	0x000000XX
0x06C	CAN_RES_DA7	CAN EFF数据字节7	0x000000XX
0x070	CAN_RES_DA8	CAN EFF数据字节8	0x000000XX
0x074	CAN_RMC	CAN 接收报文计数寄存器	0x00000000
0x07C	CAN_EXTEND	CAN工作模式选择寄存器	0x00000000

21.5.1 CAN 模式寄存器(CAN_MOD)

偏移地址：0x00
复位值：0x00000001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN	Res.	Res.	AFM	STM	LOM	RM
									rw			rw	rw	rw	rw

Bit	Field	Description
31:7	Res.	保留，必须保持复位值。
6	EN	CAN 控制器使能（Enable）： 0：使能 CAN 控制器。 1：禁止 CAN 控制器。
5:4	Res.	保留，必须保持复位值。
3	AFM	接收滤波模式（Acceptance filter mode）： 1：单接收滤除模式，单个 32 位长度的报文滤波器被激活。 0：双接收滤除模式，两个 16 位长度的报文滤波器被激活。 注：该模式只在复位模式下可配。
2	STM	自测试模式（Self test mode）： 1：自测试模式，该模式下，可以进行完整的 CAN 节点测试，不需要其他激活的 CAN 节点。CAN 控制器执行报文发送，无响应也视为发送成功。 0：正常，发送成功需要收到响应。 注：该模式只在复位模式下可配。
1	LOM	只听模式（Listen only mode）： 1：只听模式，该模式下，即使报文成功接收，CAN 控制器也不响应 CAN 总线，错误计数器不进行计数。 0：正常。 注：该模式只在复位模式下可配。
0	RM	复位模式（Reset mode）： 1：操作模式。 0：复位模式。

21.5.2 CAN 命令寄存器(CAN_CMR)

偏移地址 0x004
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRR	CDO	RRB	AT	TR
											wo	wo	wo	wo	wo

Bit	Field	Description
31:5	Res.	保留，必须保持复位值。
4	SRR	自接收请求（Self reception request）： 1：自测试模式下有效，发送缓冲报文，同时接收总线报文。 0：无行为。
S	CDO	清除数据溢出（Clear data overrun）： 1：清除不完整报文引起的数据溢出状态位。 0：无行为。
2	RRB	释放缓冲空间（Release receive buffer）： 1：释放一条报文所占据的接收缓冲空间。 0：无行为。
1	AT	中断发送（Abort transmission）： 1：计划发送但还未发送的报文，取消发送请求。 0：无行为。
0	TR	发送请求（Transmission request）： 1：一条报文将会备发送； 0：无行为。

21.5.3 CAN 状态寄存器(CAN_SR)

偏移地址 0x008
复位值：0x0000000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVRROS	BS	ES	TS	RS	TCS	TBS	DOS	RBS
							ro	ro	ro	ro	ro	ro	ro	ro	ro

Bit	Field	Description
31:9	Res.	保留，必须保持复位值。

8	OVRROS	溢出发生状态（Overrun occur status）： 1：溢出发生时，该状态位硬件置 1，软件读清零。 0：未发生任何形式的溢出。 注：溢出包括接收缓冲存入部分报文，和未存入报文的两种溢出情况。
7	BS	总线状态（Bus status）： 1：离线，CAN 控制器未处于 CAN 总线中。 0：在线，CAN 控制器处于 CAN 总线中。
6	ES	错误状态（Error status）： 1：报错，任一错误计数器值达到或超过警告阈值。 0：无错误，两个错误计数器值都低于警告阈值。
5	TS	发送状态（Transmit status）： 1：发送，正在发送报文。 0：空闲，未发送报文。
4	RS	接收状态（Receive status）： 1：接收，正在接收报文。 0：空闲，未接收报文。
3	TCS	发送完成状态（Transmitssion complete status）： 1：完成，最近的发送请求成功执行完。 0：未完成，最近的发送请求未执行完。
2	TBS	发送缓冲状态（Transmit buffer status）： 1：释放，软件可以向发送缓冲中写入报文。 0：锁定有报文正在等待发送，软件不该改变发送缓冲中的报文。
1	DOS	数据溢出状态（Data overrun status）： 1：接收缓冲中的报文由于发生溢出导致保存不完整，当该报文可读时，DOS 状态位硬件置 1。 0：表示接收缓冲的可读报文内容完整。 注：需要用清除数据溢出命令将该状态位清零。
0	RBS	接收缓冲状态（Receive buffer status）： 1：非空，一条或多条报文在接收缓冲中可读。 0：空，接收缓冲中没有接收的报文。 注：需要用释放缓冲空间命令清零。

21.5.4 CAN 中断寄存器(CAN_IR)

偏移地址：0x00C
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVRROI	BEI	ALI	EPI	Res.	DOI	EI	TI	RI
							ro	ro	ro	ro		ro	ro	ro	ro

Bit	Field	Description
31:9	Res.	保留，必须保持复位值。
8	OVRROI	溢出发生中断（Overrun occur interrupt）： 1：溢出发生时，若对应中断使能开启，该中断标志置 1。 0：软件读清零。 注：溢出包括接收缓冲存入部分报文，和未存入报文的两种溢出情况。
7	BEI	总线错误中断（Bus error interrupt）： 1：CAN 控制器上检测到总线错误时，若对应中断使能开启，则该中断标志位置 1。 0：软件读清零。
6	ALI	仲裁丢失中断 （Arbitration lost interrupt）： 1：CAN 控制器发生仲裁丢失时，从发送报文转换为接收总线报文，若对应中断使能开启，则该中断标志位置 1。 0：软件读清零。
5	EPI	错误消极中断（Error passive interrupt）： 1：CAN 控制器进入错误消极状态（任一错误计数器值超过 127），或控制器从错误消极状态进入错误积极状态，若对应中断使能开启，该中断标志位置 1。 0：软件读清零。
4	Res.	保留，必须保持复位值。
3	DOI	数据溢出中断（Data overrun interrupt）： 1：溢出导致的不完整报文可读时，若对应中断使能开启，该中断标志置 1。 0：软件读清零。
2	EI	错误中断（Error interrupt）： 1：错误状态或总线状态发生改变时，若对应中断使能开启，该中断标志置 1。 0：软件读清零。
1	TI	发送中断（Transmit interrupt）： 1：发送状态位从 0 到 1 时，发送缓冲数据可软件修改，若对应中断使能开启，该中断标志置 1。 0：软件读清零。
0	RI	接收中断（Receive interrupt）： 1：接收缓冲非空时，若对应中断使能开启，该中断标志置 1。

		0：软件读清零。

21.5.5 CAN 中断使能寄存器(CAN_IER)

偏移地址：0x010
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVRRHIE	BEIE	ALIE	EPIE		DOIE	EIE	TIE	RIE
							rw	rw	rw	rw		rw	rw	rw	rw

Bit	Field	Description
31:9	Res.	保留，必须保持复位值。
8	OVRROIE	溢出发生中断使能（Overrun occur interrupt enable）： 1：使能溢出发生中断。溢出发生时，OVRROI 置位，产生中断。 0：禁止溢出发生中断。 注：溢出包括接收缓冲存入部分报文，和未存入报文的两种溢出情况。
7	BEIE	总线错误中断使能（Bus error interrupt enable）： 1：总线错误中断使能有效， 总线上错误发生时，控制器产生中断； 0：禁止。
6	ALIE	仲裁丢失中断使能（Arbitration lost interrupt enable）： 1：仲裁丢失中断使能有效，CAN 控制器发生仲裁丢失时，控制器产生中断； 0：禁止。
5	EPIE	错误消极中断使能（Error passive interrupt enable）： 1：错误消极中断使能有效，CAN 控制器进入错误消极状态（任一错误计数器值超过 127），或控制器从错误消极状态进入错误积极状态，控制器产生中断； 0：禁止。
4	Res.	保留，必须保持复位值。
3	DOIE	溢出中断使能（Data overrun interrupt enable）： 1：数据溢出中断使能有效，数据溢出状态置位时，控制器产生中断； 0：禁止。
2	EIE	错误中断使能（Error interrupt enable）： 1：错误中断使能有效，错误状态或总线状态改变时，控制器产生中断； 0：禁止。
1	TIE	发送中断使能(Transmission interrupt enable)：

		1：发送中断使能有效，报文被成功发送，或发送缓冲可再次访问（在中断传输命令后），控制器产生中断； 0：禁止。
0	RIE	接收中断使能(Receive interrupt enable)： 1：接收中断使能有效，接收缓冲中有报文数据时，控制器产生中断； 0：禁止。

21.5.6 CAN 时序配置寄存器 0(CAN_BTR0)

偏移地址：0x018
复位值：0x00000000

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

Bit	Field	Description
31:9	Res.	保留，必须保持复位值。
9:8	SJW	同步跳转位宽（Synchronization jump width）： CAN 控制器在同步时的跳转位宽。
7:0	BRP	波特率预分频（Baud rate prescaler）： CAN 控制器基于系统时钟先进行预分频，在使用预分频后的时钟计算 Tseg2 和 Tseg1。

21.5.7 CAN 时序配置寄存器 1(CAN_BTR1)

偏移地址：0x01C
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAM	TSEG2[2:0]			TSEG1[3:0]			
									rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7	SAM	采样（Sample）： 1：总线上信号被采样 3 次； 0：总线上信号被采样 1 次。
6	Res.	保留，必须保持复位值。
5:4	TSEG2	时间字段 2 (Time segment2)： 基于预分频后的时钟，指定时间字段 2 的时长。
3:0	TSEG1	时间字段 1 (Time segment1)： 基于预分频后的时钟，指定时间字段 1 的时长。

21.5.8 CAN 仲裁丢失记录寄存器(CAN_ALC)

偏移地址：0x02C
复位值：0x00000000

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

Bit	Field	Description
31:5	Res.	保留，必须保持复位值。
4:0	ALC	仲裁丢失捕获（Arbitration lost capture）： 记录仲裁失败的位置，ALC 用于表示仲裁失败的位置，其值 0~31 对应于{ID.28~ID.18, SRTR, IDE, ID.17~ID.0, RTR} 的相应位置，十进制值对应如下。 0：仲裁丢失发生在 ID.28。 1：仲裁丢失发生在 ID.27。 2：仲裁丢失发生在 ID.26。 3：仲裁丢失发生在 ID.25。 4：仲裁丢失发生在 ID.24。 5：仲裁丢失发生在 ID.23。 6：仲裁丢失发生在 ID.22。 7：仲裁丢失发生在 ID.21。 8：仲裁丢失发生在 ID.20。 9：仲裁丢失发生在 ID.19。 10：仲裁丢失发生在 ID.18。 11：仲裁丢失发生在 SRTR。

		12：仲裁丢失发生在 IDE。 13：仲裁丢失发生在 ID.17。 14：仲裁丢失发生在 ID.16。 15：仲裁丢失发生在 ID.15。 16：仲裁丢失发生在 ID.14。 17：仲裁丢失发生在 ID.13。 18：仲裁丢失发生在 ID.12。 19：仲裁丢失发生在 ID.11。 20：仲裁丢失发生在 ID.10。 21：仲裁丢失发生在 ID.9。 22：仲裁丢失发生在 ID.8。 23：仲裁丢失发生在 ID.7。 24：仲裁丢失发生在 ID.6。 25：仲裁丢失发生在 ID.5。 26：仲裁丢失发生在 ID.4。 27：仲裁丢失发生在 ID.3。 28：仲裁丢失发生在 ID.2。 29：仲裁丢失发生在 ID.1。 30：仲裁丢失发生在 ID.0。 31：仲裁丢失发生在 RTR。

21.5.9 CAN 错误码记录寄存器(CAN_ECC)

偏移地址：0x030
复位值：0x00000000

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

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:6	ERRC	错误码 (Error code) ： 00：位错误 bit error。 01：格式错误 form error。 10：填充错误 stuff error。 11：其他错误。
5	DIR	方向 (Direction) ：

		1：TX，发送时发生错误。 0：RX，接收时发生错误。
4:0	SEG	段（Segment）： 00010： ID.28-ID.21。 00011： 帧开始。 00100： SRTR 位。 00101： IDE 位。 00110： ID.20-ID.18。 00111： ID.17-ID.13。 01001： 保留位 0。 01010： 数据区。 01011： 数据长度代码。 01100： RTR 位。 01101： 保留位 1。 01110： ID.4-ID.0。 01111： ID.12-ID.5。 10001： 活动错误标志。 10010： 中止。 10011： 支配（控制）位误差。 10110： 消极错误标志。 10111： 错误定义符。 11000： CRC 定义符。 11001： 应答通道。 11010： 帧结束。 11011： 应答定义符/CRC 序列； ERRC=01 时，为应答定义符错误； ERRC=11 时，为 CRC 序列错误； 11100： 溢出标志。 其他：保留

21.5.10 CAN 错误警告阈值寄存器(CAN_EWLR)

偏移地址：0x034
复位值：0x00000060

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

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	EWL	错误警告阈值（Error warning limit）： 默认值为 96，复位模式下，软件可改写该组寄存器值。任一错误计数器的值超过错误警告阈值时，错误状态位被置位。

21.5.11 CAN 接收错误计数寄存器(CAN_RXERR)

偏移地址：0x038
复位值：0x00000000

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

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	RX_ERR	接收错误计数器（RX error counter）： 初值为 0，复位模式下可改写该组寄存器值。

21.5.12 CAN 发送错误计数寄存器(CAN_TXERR)

偏移地址：0x03C
复位值：0x00000000

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

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	TX_ERR	发送错误计数器（TX error counter）： 初值为 0，复位模式下可改写该组寄存器值。

21.5.13 CAN 帧信息寄存器(CAN_FRAME_INFO)

偏移地址：0x040
复位值：0x000000XX

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

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	TX_ERR	发送错误计数器（TX error counter）： 初值为 0，复位模式下可改写该组寄存器值。

21.5.14 CAN 报文 ID 字节 1(CAN_ID1)

偏移地址：0x044
复位值：0x000000XX
工作模式下读写时数据结构相同，写时为发送寄存器，读时为接收寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ID1[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	ID1	帧格式为标准帧 SFF 时，对应发送报文的标识符 ID.10 ~ ID.3。 帧格式为扩展帧 EFF 时，对应发送报文的标识符 ID.28 ~ ID.21。

复位模式下，该地址可读写访问验收代码寄存器 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACR0[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	ACR0	验收代码寄存器 0 (Acceptance code register 0) 。

21.5.15 CAN 报文 ID 字节 2(CAN_ID2)

偏移地址：0x048
复位值：0x000000XX
工作模式下读写时数据结构相同，写时为发送寄存器，读时为接收寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ID2[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	ID2	帧格式为标准帧 SFF 时，高 3 位对应发送报文的标识符 ID.10 ~ ID.3，低 5 位不关心。 帧格式为扩展帧 EFF 时，对应发送报文的标识符 ID.20 ~ ID.13。

复位模式下，该地址可读写访问验收代码寄存器 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACR1[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	ACR1	验收代码寄存器 1 (Acceptance code register 1) 。

21.5.16 CAN SFF 数据字节 1 或 EFF 报文 ID 字节 3(CAN_DATA1_ID3)

偏移地址：0x04C
复位值：0x000000XX
工作模式下读写时数据结构相同，写时为发送寄存器，读时为接收寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA1_ID3[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	DATA1_ID3	帧格式为标准帧 SFF 时，对应发送报文的数据 DATA1。 帧格式为扩展帧 EFF 时，对应发送报文的标识符 ID.12 ~ ID.5。

复位模式下，该地址可读写访问验收代码寄存器 2。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACR2[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	ACR2	验收代码寄存器 2 (Acceptance code register 2) 。

21.5.17 CAN SFF 数据字节 2 或 EFF 报文 ID 字节 4(CAN_DATA2_ID4)

偏移地址：0x050

复位值：0x000000XX
工作模式下读写时数据结构相同，写时为发送寄存器，读时为接收寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA2_ID4[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	DATA2_ID4	帧格式为标准帧 SFF 时，对应发送报文的数据 DATA2。 帧格式为扩展帧 EFF 时，高 5 位对应发送报文的标识符 ID.4 ~ ID.0，低 3 位不关心。

复位模式下，该地址可读写访问验收代码寄存器 3。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACR3[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	ACR3	验收代码寄存器 3 (Acceptance code register 3) 。

21.5.18 CAN SFF 数据字节 3 或 EFF 数据字节 1(CAN_DATA3_DA1)

偏移地址：0x054
复位值：0x000000XX
工作模式下读写时数据结构相同，写时为发送寄存器，读时为接收寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA3_DA1[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	DATA3_DA1	帧格式为标准帧 SFF 时，对应发送报文的数据 DATA3。 帧格式为扩展帧 EFF 时，对应发送报文的数据 DATA1。

复位模式下，该地址可读写访问验收掩码寄存器 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AMR0[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	AMR0	验收掩码寄存器 0 (Acceptance mask register 0) 。

21.5.19 CAN SFF 数据字节 4 或 EFF 数据字节 2(CAN_DATA4_DA2)

偏移地址：0x058
复位值：0x000000XX
工作模式下读写时数据结构相同，写时为发送寄存器，读时为接收寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA4_DA2[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	DATA4_DA2	帧格式为标准帧 SFF 时，对应发送报文的数据 DATA4。 帧格式为扩展帧 EFF 时，对应发送报文的数据 DATA2。

复位模式下，该地址可读写访问验收掩码寄存器 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AMR1[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	AMR1	验收掩码寄存器 1 (Acceptance mask register 1) 。

21.5.20 CAN SFF 数据字节 5 或 EFF 数据字节 3(CAN_DATA5_DA3)

偏移地址：0x05C
复位值：0x000000XX
工作模式下读写时数据结构相同，写时为发送寄存器，读时为接收寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA5_DA3[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	DATA5_DA3	帧格式为标准帧 SFF 时，对应发送报文的数据 DATA5。 帧格式为扩展帧 EFF 时，对应发送报文的数据 DATA3。

复位模式下，该地址可读写访问验收掩码寄存器 2。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AMR2[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	AMR2	验收掩码寄存器 2 (Acceptance mask register 2) 。

21.5.21 CAN SFF 数据字节 6 或 EFF 数据字节 4(CAN_DATA6_DA4)

偏移地址：0x060
复位值：0x000000XX
工作模式下读写时数据结构相同，写时为发送寄存器，读时为接收寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA6_DA4[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	DATA6_DA4	帧格式为标准帧 SFF 时，对应发送报文的数据 DATA6。 帧格式为扩展帧 EFF 时，对应发送报文的数据 DATA4。

复位模式下，该地址可读写访问验收掩码寄存器 3。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AMR3[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	AMR3	验收掩码寄存器 3 (Acceptance mask register 3) 。

21.5.22 CAN SFF 数据字节 7 或 EFF 数据字节 5(CAN_DATA7_DA5)

偏移地址：0x064
复位值：0x000000XX
工作模式下读写时数据结构相同，写时为发送寄存器，读时为接收寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA7_DA5[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	DATA7_DA5	帧格式为标准帧 SFF 时，对应发送报文的数据 DATA7。 帧格式为扩展帧 EFF 时，对应发送报文的数据 DATA5。

21.5.23 CAN SFF 数据字节 8 或 EFF 数据字节 6(CAN_DATA8_DA6)

偏移地址：0x068
复位值：0x000000XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA8_DA6[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	DATA8_DA6	帧格式为标准帧 SFF 时，对应发送报文的数据 DATA8。 帧格式为扩展帧 EFF 时，对应发送报文的数据 DATA6。

21.5.24 CAN EFF 数据字节 7(CAN_RES_DA7)

偏移地址：0x06C
复位值：0x000000XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RES_DA7[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	RES_DA7	帧格式为扩展帧 EFF 时，对应发送报文的数据 DATA7。

21.5.25 CAN EFF 数据字节 8(CAN_RES_DA8)

偏移地址：0x070
复位值：0x000000XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RES_DA8[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7:0	RES_DA8	帧格式为扩展帧 EFF 时，对应发送报文的数据 DATA8。

21.5.26 CAN 接收报文计数寄存器(CAN_RMC)

偏移地址：0x074
复位值：0x00000000

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

Bit	Field	Description
31:7	Res.	保留，必须保持复位值。
6:0	RMC	接收报文计数器（RX message counter）。

21.5.27 CAN 工作模式选择寄存器(CAN_EXTEND)

偏移地址：0x07C
复位值：0x00000000

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

Bit	Field	Description
31:8	Res.	保留，必须保持复位值。
7	CAN_MODE	CAN 工作模式选择： 1：CAN 控制器处于 PeliCan 模式。 0：CAN 控制器处于 BasicCan 模式。
6:0	Res.	保留，必须保持复位值。

22 电源电压监测器（PVD）

22.1 简介

MCU 内嵌电源电压监测器(PVD)，可用于对供电电压 AVDD 进行监控。通过 PVD 可设置比较参考电压，当供电电压高于或低于比较参考电压时，产生中断或复位。

22.2 主要特性

- 可选参考电压：2.2V/2.4V/2.6V/2.8V/3.0V/3.2V/3.4V/3.6V/3.8V/4.0V
- 支持数字滤波消抖
- 可选择上升、下降沿和高低电平四种触发事件
- 可设置事件触发响应为中断或复位

22.3 功能说明

22.3.1 PVD 结构框图

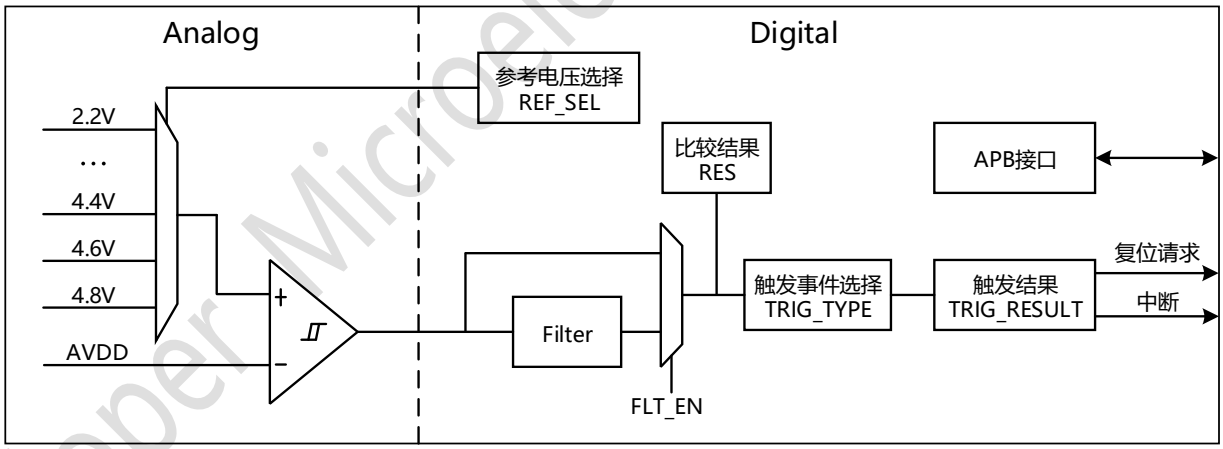

图 22-1 PVD 结构框图

22.3.2 PVD 功能

PVD 支持可编程参考电压，通过参考电压选择寄存器(PVD_CR.REF_SEL)配置，参考电压梯度为 0.2V，范围为 2.2V - 4.8V。电源电压与参考电压通过模拟比较器进行比较，当电源电压超过参考电压，输出 0，低于参考电压时，输出 1，比较结果可从 PVD_SR.RES 中读取。

PVD 内嵌数字滤波器，数字滤波器通过滤波使能 PVD_CR.FILT_EN 开启，用于对模拟比较器输出消抖，

禁用时，将比较结果直接输出。
在配置滤波器使能前，需先配置滤波长度寄存器(PVD_CR.FLT_LEN)和滤波采样频率寄存器(PVD_CR.FLT_SAMPLE)，选择滤波采样点数目和滤波采样频率。滤波器以固定采样频率采样输入电平，连续采样到多个采样点的相同电平时，输出该电平。滤波器的工作时钟为 PCLK。
在 PVD 使能后，软件需等待 PVD 准备状态就绪，当 PVD_SR.RDY 被硬件置 1 时，表明 PVD 准备就绪，所有 PVD 功能有效，准备时长为 128 个 HSI 时钟周期（约 2.67us）。
PVD 在使能后，无法更改滤波器和触发事件配置。

22.3.3 PVD 触发事件

可配置 PVD 产生中断或复位，触发事件可通过寄存器 PVD_CR.TRIG_TYPE 选择，可配置比较器输出结果为高电平、低电平、上沿或下沿作为触发事件。
当 PVD_CR.TRIG_RESULT 配置为 0，中断 PVD_CR.IE 为 1 时，触发事件发生时会产生中断，中断标志位 PVD_SR.IF 被硬件置 1，中断标记位软件写 1 清 0。
当 PVD_CR.TRIG_RESULT 配置为 1 时，触发事件发生时会触发系统复位。

22.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字(16 位)、字(32 位)访问。

表 22-1 PVD 寄存器概览

偏移地址	寄存器名	寄存器描述	复位值
0x000	PVD_CR	PVD控制寄存器	0x00000000
0x004	PVD_SR	PVD状态寄存器	0x00000000

22.4.1 控制寄存器（PVD_CR）

偏移地址：0x000
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	IE	FLT_EN	FLT_LEN[1:0]		FLT_SAMPLE[1:0]		REF_SEL[3:0]				TRIG_RESULT	TRIG_TYPE[1:0]		EN
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:14	Res.	保留，必须保持复位值。
13	IE	中断使能（Interrupt enable）

		0：中断禁止 1：中断使能
12	FLT_EN	滤波器使能（Filter enable） 0：关闭滤波器 1：开启滤波器
11:10	FLT_LEN	滤波器滤波长度（Filter length selection） 00：滤波长度为 1； 01：滤波长度为 8； 10：滤波长度为 16； 11：滤波长度为 32；
9:8	FLT_SAMPLE	滤波器滤波时钟分频系数选择（Filter clock sample selection） 00：滤波时钟分频系数为 1； 01：滤波时钟分频系数为 4； 10：滤波时钟分频系数为 16； 11：滤波时钟分频系数为 32；
7:4	REF_SEL	参考电压选择（Reference voltage selection） 0000：保留 0001：保留 0010：参考电压 2.2V 0011：参考电压 2.4V 0100：参考电压 2.6V 0101：参考电压 2.8V 0110：参考电压 3.0V 0111：参考电压 3.2V 1000：参考电压 3.4V 1001：参考电压 3.6V 1010：参考电压 3.8V 1011：参考电压 4.0V 1100：保留 1101：保留 1110：保留 1111：保留
3	TRIG_RESULT	触发结果（Trigger Result） 0：事件触发中断 1：事件触发复位
2:1	TRIG_TYPE	触发事件类型（Trigger Type） 00：比较结果下降沿 01：比较结果上升沿 10：比较结果为低电平 11：比较结果为高电平
0	EN	PVD 使能位 0：禁止 PVD 1：使能 PVD

22.4.2 状态寄存器（PVD_SR）

偏移地址：0x004
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDY	RES	IF
													ro	ro	w1c

Bit	Field	Description
31:3	Res.	保留，必须保持复位值。
2	RDY	准备状态（Ready） 0：PVD 未准备就绪 1：PVD 准备就绪
1	RES	输出结果（Result） 输出 PVD 比较电源电压和参考电压的结果 1：电源电压低于参考电压 0：电源电压高于参考电压
0	IF	触发事件中断标记（Interrupt flag） 0：中断无效 1：中断有效

23 模数转换器(ADC)

23.1 简介

ADC 是 12 位的逐次逼近型模数转换器，其具有两个模数转换单元，能够对最多 15 个模拟输入通道采样，其中有 1 个模拟输入通道为内部信号源（温度传感器），有 14 个模拟输入通道为外部信号源。ADC 模块内的两个模数转换单元能够同步触发和独立触发，每个转换单元支持单通道单次采集，单通道连续采集和序列扫描采集工作模式，每种工作模式都有独立的 16 位数据寄存器。支持硬件过采样处理的特性，该功能有效降低了 CPU 对信号监测和过采样数据处理的计算负担。支持模拟看门狗功能。另外，ADC 模块还支持通道插队数据采集，以满足在连续或扫描模式下及时获得通道转换数据。

23.2 主要特性

- 两个 12 位的 SAR ADC，支持最高 256 倍过采样，输出最高分辨率可达 16 位；
- 可配置的采样时间；
- 高达 3Msps 的 ADC 转换速率；
- 最多 14 个外部 IO 通道；
- 1 个内部通道可以测量内部温度传感器
- 最多可配置 4 个可编程增益放大器（PGA0/1/2/3）输出分别占用外部通道 0/1/2/3 作为 ADC 的输入
- 所有通道可配置使能 Buffer；
- ADC 转换可通过软件触发，也可通过配置由硬件触发，硬件触发源包括 EPWM，外部 PAD 和定时器 TIM0~5；
- 支持多种工作模式
  - 单通道单次采集；
  - 单通道连续采集；
  - 序列单次扫描采集；
  - 序列连续扫描采集；
  - 单通道单次插队采集；
- 在单通道采集结束、序列扫描结束、单通道单次插队采集结束以及发生模拟看门狗或溢出事件时都支持中断产生；
- 支持模拟看门功能，包含信号窗口电压监测和阈值检测功能；
- 支持 DMA 传输；

23.3 功能说明

23.3.1 ADC 模块框图

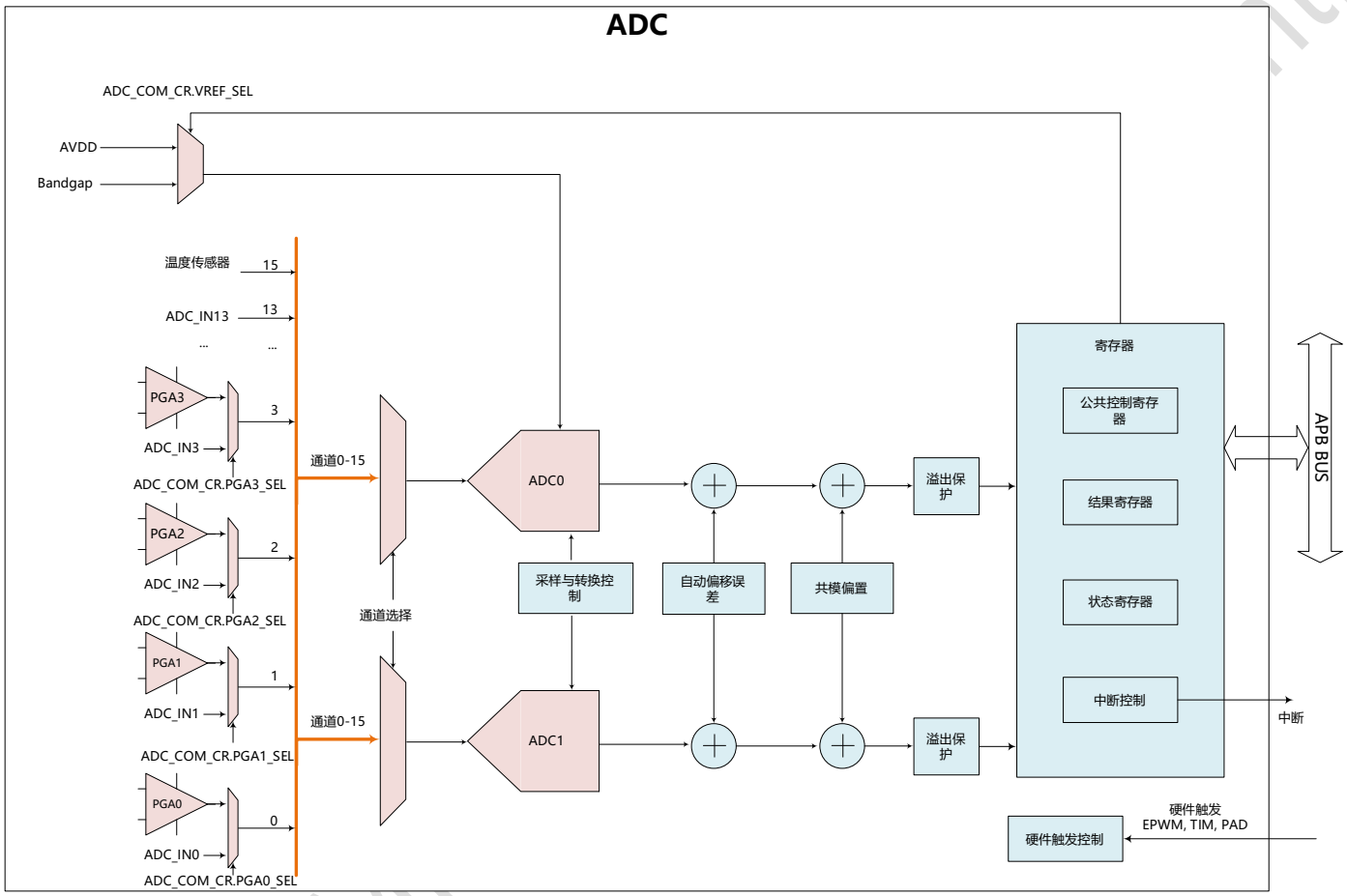

图 23-1 ADC 模块结构框图

23.3.2 ADC 参考电压源

ADC 具有两个模拟参考电压源，AVDD 和 Bandgap，其通过配置寄存器 ADC_CCR 中的 VREF_SEL 值选择。

23.3.3 ADC 工作时钟

ADC 采样的时钟由 APB 时钟(PCLK)经过分频后产生。寄存器 ADC_COM_CR 中的位 PRESC[2:0]用于设置预分频值，默认值为 2 分频。

23.3.4 ADC 通道校准

因为在生产制造过程中由于工艺等原因，芯片内的 ADC 各个通道存在不同的偏移误差，这些误差会在出厂过程中预先校准完成。

23.3.5 ADC 工作使能

在上电复位后，ADC 默认处于关闭状态，通过配置寄存器 ADCx_CFG、ADCx_TRIG_CR 和 ADCx_IE 可以启动 ADC 数据采集和中断。

其中，寄存器 ADCx_CFG 中的位 ADC_EN 用于使能 ADC，该位置 1 后对应的 ADC 模块内部输出信号 ADC_EN 将为高。寄存器 ADCx_TRIG_CR 中的位 SW_START、HW0_EN 和 HW1_EN 用于使能软件触发和硬件触发，当触发产生后对应的 ADC 模块内部时序如下图 23-2 所示。

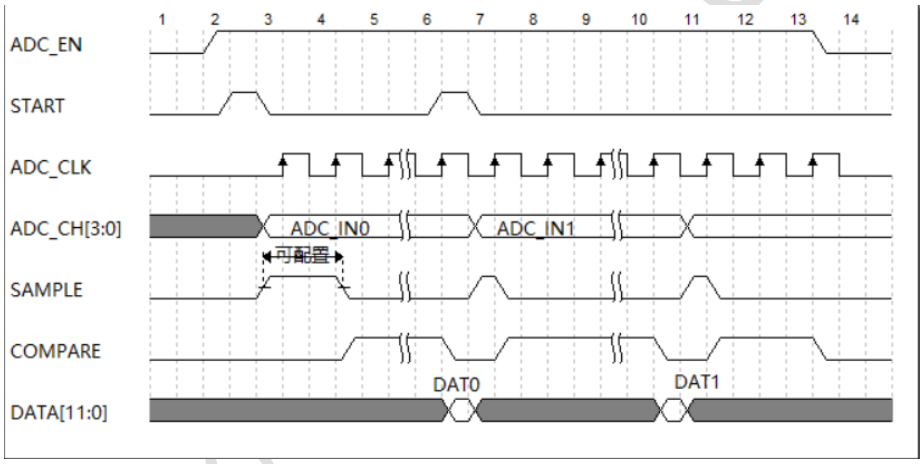

图 23-2 ADC 内部信号图

请参照以下流程使能 ADC：

1. 检查确认 SYS_CFG 模块中配置使能 ADC 模块工作时钟；
2. 将 ADCx_CFG 寄存器中的 ADC_EN 位置 1，置 1 后将使能 ADCx；
3. 将 ADCx_IE 寄存器中的中断使能；
4. 将 ADCx_TRIG_CR 寄存器中的 SW_START、HW0_EN、HW1_EN 使能，ADCx 在软件触发下转换数据或者等待硬件触发转换数据。

请参照以下流程关闭 ADC：

1. 将 ADCx_TRIG_CR 寄存器中的 STOP 软件置 1 或者 HW0_EN 和 HW1_EN 软件置 0，ADC 将停止数据转换或者禁止硬件触发；
2. 将 ADCx_CFG 寄存器中的 ADC_EN 位置 0，ADC 关闭；

23.3.6 ADC 通道选择

ADC 共有最多 14 个外部输入通道和 1 个内部通道，ADC 在不同的工作模式下，会通过不同的寄存器来选择采样通道。
寄存器 ADCx_TRIG_CR 中的 MODE[2:0]设置 ADC 的工作模式。
工作在软件触发模式时，寄存器 ADCx_TRIG_CR 中的 SW_CH[3:0]用于设置选择 ADC 通道；
工作在硬件触发模式时，寄存器 ADCx_TRIG_CR 中的 HW1_CH[3:0]、HW0_CH[3:0] 用于设置选择 ADC 通道；
工作在插队模式下，寄存器 ADC_INJ_CR 中的 CH[3:0] 用于设置选择 ADC 插队通道；
在序列扫描模式下，寄存器 ADCx_SCAN_SQR0 和 ADCx_SCAN_SQR1 中的 SQ0_CH[3:0]至 SQ13_CH[3:0]决定 ADC 每次序列转换采集时的通道。
14 个外部通道中，通道 3~0 可以通过 PGA3..0_SEL 用于选择时外部输入通道还是程控增益放大器输入。

23.3.7 ADC 通道采样时间设置

ADC 在逐次比较转换电压之前，其测量通道输入的电压必须先与采样电容建立连接，即采样电容需要经过时间 t_{sample} 的充电。
寄存器 ADC_COM_SMPT 中的位 SMPT0[2:0]和 SMPT1[2:0]代表采样时间，可以通过软件进行设置，ADC 的模拟输入通道 0~15 对应的采样时间 t_{sample}，通过 SMPT_SELx(x=0~13,15)中各自对应的位选择是 SMPT0 还是 SMPT1。
总转换时间的计算公式如下：
$t_{conv} = t_{sample} + 14 \times$ ADC 工作时钟周期
示例：
如果 ADC 输入的 PCLK=48MHz，预分频 PRESC[2:0]=1，则 ADC 工作时钟为 24MHz。
当 SMPT0/1=0x1 时，t_{sample}=4 ×ADC 工作时钟周期，t_{conv}=18 × ADC 工作时钟周期=0.75us。

23.3.8 ADC 触发源

ADC 的触发源分为软件触发和硬件触发，软件触发通过写寄存器 ADCx_TRIG_CR 中的位 SW_ START 为 1 产生，硬件触发源则来自外部 EPWM，PAD 或 TIM0~5 模块，硬件触发源设置 HW0_SRC[3:0]或 HW1_SRC[3:0] 来选择。
下表 23-1 显示了外部触发源和 HW_TRIGx_SRC[3:0]对照。

表 23-1 外部硬件触发源

HWx_SRC[3:0]	触发源	描述
0000	TIM0	TIM0 完成一次周期计数时，输出的触发信号
0001	TIM1	TIM1 完成一次周期计数时，输出的触发信号
0010	TIM2	TIM2 完成一次周期计数时，输出的触发信号
0011	TIM3	TIM3 完成一次周期计数时，输出的触发信号

0100	TIM4	TIM4 完成一次周期计数时，输出的触发信号
0101	TIM5	TIM5 完成一次周期计数时，输出的触发信号
0110	ADC_TRIG	引脚 ADC_TRIG 输入的上升沿触发信号
0111	ADC_TRIG	引脚 ADC_TRIG 输入的下降沿触发信号
1000	EPWM0_ADC_CMP1	EPWM0 中计数值等于 EPWM_ADC_CMP1 比较值时，输出的触发信号
1001	EPWM0_ADC_CMP2	EPWM0 中计数值等于 EPWM_ADC_CMP2 比较值时，输出的触发信号

23.3.9 ADC 采样精度设置

寄存器 ADC_COM_SMPC 中的位 NORM_RES[1:0]和 OVS_RES[2:0]分别用于设置非过采样与过采样时的数据精度，非过采样数据精度可配置为：6、8、10、12bit，过采样数据精度可配置为：12、13、14、15、16bit。

寄存器 ADC_CH_SMPC 中的位 OVSEN0~13,15、OVS_RATE [1:0]和 OVS_RES [2:0]用于配置 ADC 的过采样功能，其中 OVSEN0~13,15 分别用于使能模拟输入通道 0~15 过采样，OVS_RATE 用于设置所有通道的过采样率，OVS_RES 用于设置所有通道得到过采样数据的精度。ADC 模块内部的过采样处理单元对同一通道数据进行累加，并根据配置的过采样率计算平均值，平均值结果为 16 位的单个数据，该过程计算方式如下：

$$data\_avg = \frac{\sum_{n=0}^{n=N-1} adc\_data[n]}{2^M}$$

其中，过采样率 N 通过 OVS_RATE [1:0]位进行定义，它的范围是 4x 到 256x。M 表示数据移位，在进行过采样数据出来时最大可以右移 8 位，移位值由 OVS_RATE [1:0]和 OVS_RES [2:0]配置决定，对应如下表

表 23-2 移位值与 OVS_RATE [1:0]和 OVS_RES [2:0]关系

过采样率(OVS_RATE [1:0])	数据精度(OVS_RES [2:0])	移位值(M)
4x	12 位	2
	13 位	1
	14 位	0
	15 位	-1
	16 位	-2
16x	12 位	4
	13 位	3
	14 位	2
	15 位	1
	16 位	0
64x	12 位	6
	13 位	5
	14 位	4
	15 位	3
	16 位	2

128x	12 位	8
	13 位	7
	14 位	6
	15 位	5
	16 位	4

23.3.10 ADC 采样数据处理

ADC 采样得到的数据，将会根据出厂设置的通道偏移误差进行修正。偏移误差修正后的值再减去用户配置的通道偏置寄存器 ADC_COM_CH_OFFSET0~13,15 中的偏置 VALUEx[11:0](x=0~13,15)值，最后根据寄存器 ADCx_FMT 的配置对数据格式进行转换。

23.3.11 ADC 数据格式设置

寄存器 ADCx_FMT_CR 用于配置数据的对齐方式和符号类型，其中的位 ALIGN0~13,15 用于设置转换后的数据在寄存器（ADCx_DR_SINGLE[15:0]、ADCx_DR_INJ[15:0]、ADCx_DR_TRIG0/1[15:0]、ADCx_DR _SCAN0~13[15:0]）中的左右对齐，而位 SIGN0~10 用于设置数据符号类型。
数据精度、对齐方式和符号类型构成的数据格式如下图 23-3 所示。

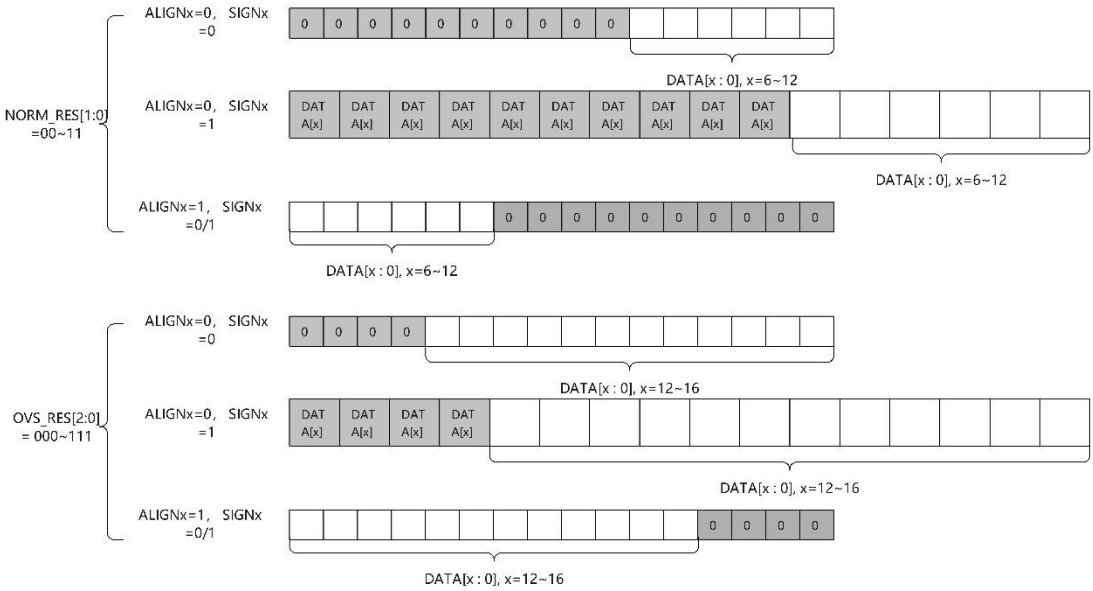

图 23-3 ADC 模拟输入选通结构

23.3.12 中断

ADC 通道采样完成后，会置位对应的标志位，并在中断使能时输出中断给 CPU。
当 CPU 未及时读取数据寄存器中的数据，取而又有新的数据被硬件更新时，会产生数据溢出标志。

在溢出状态下 OVR_EOI、OVR_EOC、OVR_EOT0、OVR_EOT1 或 OVR_EOS 置 1，如果 ADCx_IE 寄存器中的 OVR 为 1 则 ADC 会产生中断事件。在溢出状态下 ADC 能够继续触发并转换数据，新的数据会覆盖对应的数据寄存器，软件可以通过向 OVR_EOI、OVR_EOC、OVR_EOT0、OVR_EOT1、OVR_EOS 或者 OVR_SUM 写 1 来清零。

中断事件	事件标志	中断使能	数据溢出标志
软件单通道转换完成	EOC	EOC_IE	OVR_EOC
硬件 TRIG0 单通道转换完成	EOT0	EOT0_IE	OVR_EOT0
硬件 TRIG1 单通道转换完成	EOT1	EOT1_IE	OVR_EOT1
序列扫描转换完成	EOS	EOS_IE	OVR_EOS
插队转换完成	EOI	EOI_IE	OVR_EOI

23.3.13 单通道单次数据采集模式

单通道单次数据采集模式可以设置为软件触发或者硬件触发两种，当设置寄存器 ADCx_TRIG_CR 中的位 MODE[2:0]=000 时为软件触发，当设置 ADCx_TRIG_CR 中的位 MODE[2:0]=100 时为硬件触发。
在 MODE[2:0]=000 单通道单次软件触发模式下时，ADC 采集数据的流程如下：

1. 软件设置寄存器 ADCx_IE 中的位 EOC 为 1，启用单通道单次软件触发模式中断。
2. 软件设置寄存器 ADCx_TRIG_CR 中的位 MODE[2:0]=000，并且可以同时配置 ADC 转换输入的模拟通道 SW_CH[3:0]。
3. 软件写寄存器 ADCx_TRIG_CR 中的位 SW _START 为 1，等待寄存器 ADCx_ISR 中的位 EOC 中断事件产生。
4. 软件读寄存器 ADCx_DR_SINGLE 中的位 DATA [15:0]得到 ADC 转换后的数据。

在 MODE[2:0]=100 单通道单次硬件触发模式下时，ADC 采集数据的流程如下：

1. 软件设置寄存器 ADCx_IE 中的位 EOT0 或 EOT1 为 1，启用单通道单次硬件触发模式中断。
2. 软件设置寄存器 ADCx_TRIG_CR 中的位 MODE[2:0]=100，并且可以同时配置 ADC 转换输入的模拟通道 HW0_CH[3:0]或 HW1_CH[3:0]，以及对应的硬件触发源 HW0_SRC[3:0]或 HW1_SRC[3:0]。
3. 软件写寄存器 ADCx_TRIG_CR 中的位 HW0_EN 或 HW1_EN 为 1，等待硬件触发。
4. 当硬件触发源输入触发信号后，寄存器 ADCx_ISR 中的位 EOT0 或 EOT1 中断事件产生，软件读寄存器 ADCx_DR_TRIG0 和 ADC_DR_TRIG1 中的 DATA[15:0]得到硬件触发采集的单通道数据。

23.3.14 序列单次扫描数据采集模式

序列单次扫描数据采集模式可以设置为软件触发或者硬件触发两种，当设置寄存器 ADCx_TRIG_CR 中的位 MODE[2:0]=001 时为软件触发，当设置 ADCx_TRIG_CR 中的位 MODE[2:0]=101 时为硬件触发。
在 MODE[2:0]=001 序列单次扫描软件触发模式下时，ADC 采集数据的流程如下：

1. 软件设置寄存器 ADCx_IE 中的位 EOS 为 1，使能序列扫描模式中断。
2. 软件设置寄存器 ADCx_TRIG_CR 中的位 MODE[2:0]=001，并且配置寄存器 ADCx_SCAN_SQR0 和 ADCx_SCAN_SQR1，其中的序列长度 LEN[3:0]=0~14，序列 0~13 对应的采样通道由 SQ0_CH[3:0] ~ SQ13_CH[3:0]设置。
3. 软件写寄存器 ADCx_TRIG_CR 中的位 SW_START 为 1，等待寄存器 ADCx_ISR 中的位 EOS 中断事件产生。
4. 软件读寄存器 ADCx_DR_SCAN0~ADCx_DR_SCAN13 中的位 DATA [15:0]得到每个序列的采样通道转换数据。

在 MODE[2:0]=101 通道序列单次扫描硬件触发模式下时，ADC 采集数据的流程如下：

1. 软件设置寄存器 ADCx_IE 中的位 EOS 为 1，使能序列扫描模式中断。
2. 软件设置寄存器 ADCx_TRIG_CR 中的位 MODE[2:0]=001，并且配置寄存器 ADCx_SCAN_SQR0 和 ADCx_SCAN_SQR1，其中的序列长度 LEN[3:0]=1~14，序列 0~13 对应的采样通道由 SQ0_CH[3:0] ~ SQ13_CH[3:0]设置。
3. 写寄存器 ADCx_TRIG_CR 中的位 HW0_EN 或 HW1_EN 为 1，等待硬件触发。
4. 当硬件触发源输入触发信号后，ADCx_ISR 中的位 EOS 中断事件产生。
5. 软件读寄存器 ADCx_DR_SCAN0~ADCx_DR_SCAN13 中的位 DATA[15:0]得到每个序列的采样通道转换数据。

23.3.15 软件触发连续采集模式

软件触发连续数据采集模式分为单通道连续采集和序列连续采集，设置寄存器 ADCx_TRIG_CR 中的位 MODE[2:0]=010 时为单通道连续采集模式，当设置 ADCx_TRIG_CR 中的位 MODE[2:0]=011 时为序列连续采集模式。

单通道连续采集和序列连续采集流程如下：

1. 软件设置寄存器 ADCx_IE 中的位 EOS 或为 EOC1，使能序列扫描模式或单通道模式中断。
2. a) 单通道连续采集模式，软件设置寄存器 ADCx_TRIG_CR 中的位 MODE[2:0]=010，并且可以同时配置 ADC 转换输入的模拟通道 SW_CH[3:0]。

   b) 序列连续采集模式，软件设置寄存器 ADCx_TRIG_CR 中的位 MODE[2:0]=011，并且配置寄存器 ADCx_SCAN_SQR0 和 ADCx_SCAN_SQR1，其中的序列长度 LEN[3:0]=0~14，序列 0~13 对应的采样通道由 SQ0_CH[3:0] ~ SQ13_CH[3:0]设置。
3. 寄存器 ADCx_TRIG_CR 中的位 CONT_WAIT[2:0]用于设置连续采时间间隔，软件触发前通过软件进行配置。
4. 软件写寄存器 ADCx_TRIG_CR 中的位 SW_ START 为 1，等待寄存器 ADCx_ISR 中的位 EOC 或 EOS 中断事件产生。
5. 软件读寄存器 ADCx_DR_SCAN0~ADCx_DR_SCAN13 中的位 DATA[15:0]得到每个序列的采样通道转换数据，或者读取寄存器 ADCx_DR_SINGLE 中的位 DATA [15:0]得到单通道每次采集的数据。
6. 软件写寄存器 ADCx_TRIG_CR 中的位 STOP 为 1，等待 STOP 和 SW_START 由硬件设置为 0 时

ADC 连续采集模式停止。

23.3.16 通道插队采样

在单通道连续采样或序列扫描采样时，为了满足应用不中止 ADC 采样模式而及时获得模拟输入通道 0~15 的数据，可以通过写寄存器 ADCx_INJ_CR 中的位 CH[3:0]和位 START 来快速触发 ADC 转换通道数据。

寄存器 ADCx_INJ_CR 中的位 CH[3:0]为插入的模拟通道，位 START 通过软件写 1 启动通道插队，在完成插队通道数据转换后位 START 由硬件清零，转换的数据由硬件存储在寄存器 ADCx_DR_INJ。

数据寄存器 ADCx_DR_INJ 由硬件更新，如果 ADCx_IE 寄存器中的位 EOI 软件设置为 1，当 ADCx_DR_INJ 数据更新时中断标志 EOI 将会为 1，必须通过软件向位 EOI 写 1 清除中断。如果位 OVR 软件设置为 1，在中断标志 EOI 将会为 1 时 ADCx_DR_INJ 再次更新了数据，则会产生中断标志 OVR_EOI，必须通过软件向位 OVR_EOI 写 1 清除中断。

23.3.17 模拟窗口看门狗

模拟窗口看门狗可以用于监测 ADC 采样通道转换的数据是否保持在用户设置的电压范围内，此功能可以极大的减少 CPU 的负担，软件只需配置寄存器 AWD_CR 和 AWD_TH。

寄存器 AWD_CR 中的位 EN0~10 对应 ADC 输入的模拟通道 0~10 监测功能的使能，位 WH、位 WM 和 WL 用于使能所有通道都开启大于阈值、阈值范围内和小于阈值的监测功能。

寄存器 AWD_TH 通过软件设置用户定义的阈值，其中的位 HT[15:0]用于设置窗口阈值的上限，而 LT[15:0]用于设置窗口阈值的下限。用户配置 HT[15:0]和 LT[15:0]应该与监测的通道转换数据保持一致的精度、对齐和符号类型，相应配置情况如下表所示。

表 23-3 AWD_TH 寄存器配置

数据精度	符号类型	数据对齐	阈值
6 位	无符号	右对齐	HT[5:0]和 LT[5:0]为有效的阈值，HT[15:6]和 LT[15:6]必须为 0。
		左对齐	HT[15:10]和 LT[15:10]为有效的阈值，HT[9:0]和 LT[9:0]必须为 0。
	有符号	右对齐	HT[5:0]和 LT[5:0]为有效的阈值，HT[15:6]和 LT[15:6]必须为符号位的值。
		左对齐	HT[15:10]和 LT[15:10]为有效的阈值，HT[9:0]和 LT[9:0]必须为 0。
8 位	无符号	右对齐	HT[7:0]和 LT[7:0]为有效的阈值，HT[15:8]和 LT[15:8]必须为 0。
		左对齐	HT[15:8]和 LT[15:8]为有效的阈值，HT[7:0]和 LT[7:0]必须为 0。
	有符号	右对齐	HT[7:0]和 LT[7:0]为有效的阈值，HT[15:8]和 LT[15:8]必须为符号位的值。
		左对齐	HT[15:8]和 LT[15:8]为有效的阈值，HT[7:0]和 LT[7:0]必须为

			0。
10 位	无符号	右对齐	HT[9:0]和 LT[9:0]为有效的阈值，HT[15:10]和 LT[15:10]必须为 0。
		左对齐	HT[15:6]和 LT[15:6]为有效的阈值，HT[5:0]和 LT[5:0]必须为 0。
	有符号	右对齐	HT[9:0]和 LT[9:0]为有效的阈值，HT[15:10]和 LT[15:10]必须为符号位的值。
		左对齐	HT[15:6]和 LT[15:6]为有效的阈值，HT[5:0]和 LT[5:0]必须为 0。
12 位	无符号	右对齐	HT[11:0]和 LT[11:0]为有效的阈值，HT[15:12]和 LT[15:12]必须为 0。
		左对齐	HT[15:4]和 LT[15:4]为有效的阈值，HT[3:0]和 LT[3:0]必须为 0。
	有符号	右对齐	HT[11:0]和 LT[11:0]为有效的阈值，HT[15:12]和 LT[15:12]必须为符号位的值。
		左对齐	HT[15:4]和 LT[15:4]为有效的阈值，HT[3:0]和 LT[3:0]必须为 0。
13 位	无符号	右对齐	HT[12:0]和 LT[12:0]为有效的阈值，HT[15:13]和 LT[15:13]必须为 0。
		左对齐	HT[15:3]和 LT[15:3]为有效的阈值，HT[2:0]和 LT[2:0]必须为 0。
	有符号	右对齐	HT[12:0]和 LT[12:0]为有效的阈值，HT[15:13]和 LT[15:13]必须为符号位的值。
		左对齐	HT[15:3]和 LT[15:3]为有效的阈值，HT[2:0]和 LT[2:0]必须为 0。
… …	… …	… …	… …
16 位	无符号	右对齐	HT[15:0]和 LT[15:0]为有效的阈值。
		左对齐	
	有符号	右对齐	
		左对齐	

根据用户配置的阈值 HT[15:0]和 LT[15:0]，ADC 模拟看门狗能够实现三种电压范围监测，分别如下：

如果软件设置 AWD_CR 寄存器位 WH=1，当位 ENx=1 且转换后的数据 ADC_DR[15:0]大于 HT[15:0]时，AWD_SUM 将会被硬件置 1，若 IE 寄存器中位 AWD=1，则中断标志 AWD_CHx 会被硬件置 1 并产生中断直到软件向位 AWD_CHx 写 1 清除中断标志。

如果软件设置 AWD_CR 寄存器位 WM=1，当位 ENx=1 且转换后的数据 ADC_DR[15:0]小于等于 HT[15:0]且大于等于 LT[15:0]时，AWD_SUM 将会被硬件置 1，若 IE 寄存器中位 AWD=1，则中断标志 AWD_CHx 会被硬件置 1 并产生中断直到软件向位 AWD_CHx 写 1 清除中断标志。

如果软件设置 AWD_CR 寄存器位 WL=1，当位 ENx=1 且转换后的数据 ADC_DR[15:0]小于 LH[15:0]时，AWD_SUM 将会被硬件置 1，若 IE 寄存器中位 AWD=1，则中断标志 AWD_CHx 会被硬件置 1 并产生中断直到软件向位 AWD_CHx 写 1 清除中断标志。

23.3.18 采集数据 DMA 传输

数据寄存器 ADCx_DR_SINGLE、ADCx_DR_TRIG0、ADCx_DR_TRIG1 和 ADCx_DR_SCAN0~13 具有独立地址，在低采样率时可以由 ADC 中断通知 CPU 来获取转换的数据，在高采样率时则需要通过 DMA 来高效的获取数据，以防止数据寄存器发生溢出。

ADC 模块具有 4 路 DMA 请求，分别为硬件触发 1 单通道数据 DMA 请求、硬件触发 0 单通道数据 DMA 请求、序列扫描数据 DMA 请求和软件触发单通道数据 DMA 请求。

寄存器 ADCx_CFG 中的位 EOT1_DMA_EN、EOT0_DMA_EN、EOS_DMA_EN 和 EOC_DMA_EN 通过软件配置，用于使能 DMA 请求。

23.4 寄存器概述

如无特殊说明，下列寄存器均可支持字符（8 位）、半字（16 位）、字（32 位）访问。

表 23-4 ADC 寄存器概述

偏移地址	寄存器名	寄存器描述	复位值
0x00	ADC_COM_CR	ADC 公共控制寄存器	0210FFFF
0x04	ADC_COM_SMPT	ADC 采样时间寄存器	0x0000
0x08	ADC_COM_SMPC	ADC 采样控制寄存器	0x0000
0x0C	Reserved	保留寄存器	0x0000
0x10	Reserved	保留寄存器	0x0000
0x14	Reserved	保留寄存器	0x0000
0x18	Reserved	保留寄存器	0x0000
0x1C	Reserved	保留寄存器	0x0000
0x20	ADC_COM_CH_OFFSET0	ADC 通道偏移寄存器 0	0x0000
0x24	ADC_COM_CH_OFFSET1	ADC 通道偏移寄存器 1	0x0000
0x28	ADC_COM_CH_OFFSET2	ADC 通道偏移寄存器 2	0x0000
0x2C	ADC_COM_CH_OFFSET3	ADC 通道偏移寄存器 3	0x0000
0x30	ADC_COM_CH_OFFSET4	ADC 通道偏移寄存器 4	0x0000
0x34	ADC_COM_CH_OFFSET5	ADC 通道偏移寄存器 5	0x0000
0x38	ADC_COM_CH_OFFSET6	ADC 通道偏移寄存器 6	0x0000
0x3C	ADC_COM_CH_OFFSET7	ADC 通道偏移寄存器 7	0x0000
0x40	ADC_COM_CH_OFFSET8	ADC 通道偏移寄存器 8	0x0000
0x44	ADC_COM_CH_OFFSET9	ADC 通道偏移寄存器 9	0x0000
0x48	ADC_COM_CH_OFFSET10	ADC 通道偏移寄存器 10	0x0000
0x4C	ADC_COM_CH_OFFSET11	ADC 通道偏移寄存器 11	0x0000
0x50	ADC_COM_CH_OFFSET12	ADC 通道偏移寄存器 12	0x0000
0x54	ADC_COM_CH_OFFSET13	ADC 通道偏移寄存器 13	0x0000
0x58	ADC_COM_CH_OFFSET14	ADC 通道偏移寄存器 14	0x0000
0x5C	ADC_COM_CH_OFFSET15	ADC 通道偏移寄存器 15	0x0000
0x100	ADC0_ISR	ADC0 中断状态寄存器	0x0000

0x104	ADC0_IE	ADC0 中断使能寄存器	0x0000
0x108	ADC0_CFG	ADC0 配置寄存器	0x0000
0x10C	ADC0_FMT_CR	ADC0 数据格式寄存器	0x0000
0x110	ADC0_TRIG_CR	ADC0 触发控制寄存器	0x0000
0x114	ADC0_INJ_CR	ADC0 插队控制寄存器	0x0000
0x118	ADC0_AWD_CR	ADC0 模拟看门狗控制寄存器	0x0000
0x11C	ADC0_AWD_TH	ADC0 模拟看门狗触发寄存器	0x0000
0x120	ADC0_SCAN_SQR0	ADC0 扫描序列寄存器 0	0x0000
0x124	ADC0_SCAN_SQR1	ADC0 扫描序列寄存器 1	0x0000
0x128	ADC0_DR_SINGLE	ADC0 单次数据寄存器	0x0000
0x12C	ADC0_DR_INJ	ADC0 插队数据寄存器	0x0000
0x130	ADC0_DR_TRIG0	ADC0 硬件触发数据寄存器 0	0x0000
0x134	ADC0_DR_TRIG1	ADC0 硬件触发数据寄存器 1	0x0000
0x138	ADC0_DR_SCAN0	ADC0 扫描数据寄存器 0	0x0000
0x13C	ADC0_DR_SCAN1	ADC0 扫描数据寄存器 1	0x0000
0x140	ADC0_DR_SCAN2	ADC0 扫描数据寄存器 2	0x0000
0x144	ADC0_DR_SCAN3	ADC0 扫描数据寄存器 3	0x0000
0x148	ADC0_DR_SCAN4	ADC0 扫描数据寄存器 4	0x0000
0x14C	ADC0_DR_SCAN5	ADC0 扫描数据寄存器 5	0x0000
0x150	ADC0_DR_SCAN6	ADC0 扫描数据寄存器 6	0x0000
0x154	ADC0_DR_SCAN7	ADC0 扫描数据寄存器 7	0x0000
0x158	ADC0_DR_SCAN8	ADC0 扫描数据寄存器 8	0x0000
0x15C	ADC0_DR_SCAN9	ADC0 扫描数据寄存器 9	0x0000
0x160	ADC0_DR_SCAN10	ADC0 扫描数据寄存器 10	0x0000
0x164	ADC0_DR_SCAN11	ADC0 扫描数据寄存器 11	0x0000
0x168	ADC0_DR_SCAN12	ADC0 扫描数据寄存器 12	0x0000
0x16C	ADC0_DR_SCAN13	ADC0 扫描数据寄存器 13	0x0000
0x200	ADC1_ISR	ADC1 中断状态寄存器	0x0000
0x204	ADC1_IE	ADC1 中断使能寄存器	0x0000
0x208	ADC1_CFG	ADC1 配置寄存器	0x0000
0x20C	ADC1_FMT_CR	ADC1 数据格式寄存器	0x0000
0x210	ADC1_TRIG_CR	ADC1 触发控制寄存器	0x0000
0x214	ADC1_INJ_CR	ADC1 插队控制寄存器	0x0000
0x218	ADC1_AWD_CR	ADC1 模拟看门狗控制寄存器	0x0000
0x21C	ADC1_AWD_TH	ADC1 模拟看门狗触发寄存器	0x0000
0x220	ADC1_SCAN_SQR0	ADC1 扫描序列寄存器 0	0x0000
0x224	ADC1_SCAN_SQR1	ADC1 扫描序列寄存器 1	0x0000
0x228	ADC1_DR_SINGLE	ADC1 单次数据寄存器	0x0000
0x22C	ADC1_DR_INJ	ADC1 插队数据寄存器	0x0000
0x230	ADC1_DR_TRIG0	ADC1 硬件触发数据寄存器 0	0x0000
0x234	ADC1_DR_TRIG1	ADC1 硬件触发数据寄存器 1	0x0000
0x238	ADC1_DR_SCAN0	ADC1 扫描数据寄存器 0	0x0000
0x23C	ADC1_DR_SCAN1	ADC1 扫描数据寄存器 1	0x0000

0x240	ADC1_DR_SCAN2	ADC1 扫描数据寄存器 2	0x0000
0x244	ADC1_DR_SCAN3	ADC1 扫描数据寄存器 3	0x0000
0x248	ADC1_DR_SCAN4	ADC1 扫描数据寄存器 4	0x0000
0x24C	ADC1_DR_SCAN5	ADC1 扫描数据寄存器 5	0x0000
0x250	ADC1_DR_SCAN6	ADC1 扫描数据寄存器 6	0x0000
0x254	ADC1_DR_SCAN7	ADC1 扫描数据寄存器 7	0x0000
0x258	ADC1_DR_SCAN8	ADC1 扫描数据寄存器 8	0x0000
0x25C	ADC1_DR_SCAN9	ADC1 扫描数据寄存器 9	0x0000
0x260	ADC1_DR_SCAN10	ADC1 扫描数据寄存器 10	0x0000
0x264	ADC1_DR_SCAN11	ADC1 扫描数据寄存器 11	0x0000
0x268	ADC1_DR_SCAN12	ADC1 扫描数据寄存器 12	0x0000
0x26C	ADC1_DR_SCAN13	ADC1 扫描数据寄存器 13	0x0000

23.4.1 ADC 公共控制寄存器(ADC_COM_CR)

偏移地址：0x000
复位值：0x0210FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SYNC_TRIG	Res.	VREF_SEL	Res.	PRESC[2:0]			PGA_SEL3	PGA_SEL2	PGA_SEL1	PGA_SEL0
					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
BUFEN15	Res.	BUFEN13	BUFEN12	BUFEN11	BUFEN10	BUFEN9	BUFEN8	BUFEN7	BUFEN6	BUFEN5	BUFEN4	BUFEN3	BUFEN2	BUFEN1	BUFEN0
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:27	Res.	保留，保持为复位值。
26	SYNC_TRIG	ADC0 和 ADC1 软件同步触发使能（Software synchronization triggers），由软件置 1 和清零，以选择使能同步触发功能： 1：使能；0：禁止。
25	Res.	保留，保持为复位值 1。
24	VREF_SEL	ADC 参考电压源选择： 1：Bandgap；0：AVDD。 注：仅当 ADC 已禁止时（寄存器 ADC0_CFG[0]值为 0），才允许通过软件对这些位写操作。
23	Res.	保留，保持为复位值。
22：20	PRESC[2:0]	ADC 工作时钟预分频（ADC prescaler ），由软件置 1 和清零，以选择工作时钟频率。 000：输入 PCLK 时钟未分频；

		001：输入 PCLK 时钟 2 分频； 010：输入 PCLK 时钟 3 分频； 011：输入 PCLK 时钟 4 分频； 100：输入 PCLK 时钟 5 分频； 101：输入 PCLK 时钟 6 分频； 110：输入 PCLK 时钟 7 分频； 111：输入 PCLK 时钟 8 分频； 注：仅当 ADC 已禁止时（寄存器 ADC0_CFG[0]值为 0），才允许通过软件对这些位写操作。
19	PGA_SEL3	可编程增益放大器（PGA）通道使能，由软件置 1 和清零。 1：使能；0：禁止。
18	PGA_SEL2	
17	PGA_SEL1	
16	PGA_SEL0	
15	BUFEN15	模拟通道 15 上的 buffer 使能，由软件置 1 和清零。 1：使能；0：禁止。
14	Res.	保留，保持为复位值。
13	BUFEN13	模拟通道 0~13 上的 buffer 使能，由软件置 1 和清零。 1：使能；0：禁止。
12	BUFEN12	
11	BUFEN11	
10	BUFEN10	
9	BUFEN9	
8	BUFEN8	
7	BUFEN7	
6	BUFEN6	
5	BUFEN5	
4	BUFEN4	
3	BUFEN3	
2	BUFEN2	
1	BUFEN1	
0	BUFEN0	

23.4.2 ADC 采样时间寄存器(ADC_COM_SMPT)

偏移地址：0x04
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SMPT1[2:0]			Res.	SMPT0[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw	rw		rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMPT_SEL15	Res.	SMPT_SEL13	SMPT_SEL12	SMPT_SEL11	SMPT_SEL10	SMPT_SEL9	SMPT_SEL8	SMPT_SEL7	SMPT_SEL6	SMPT_SEL5	SMPT_SEL4	SMPT_SEL3	SMPT_SEL2	SMPT_SEL1	SMPT_SEL0
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31	Res.	保留，保持为复位值。
30:28	SMPT1[2:0]	通道采样时间值选择 1(Sampling time selection 1)，这些位由软件写入，用于选择应用于所有通道的采样时间。 000：2 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 001：4 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 010：8 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 011：16 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 100：32 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 101：64 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 110：128 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 111：256 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 注：仅当 ADC 已禁止时（寄存器 ADC0_CFG[0]值为 0），才允许通过软件对这些位写操作。
27	Res.	保留，保持为复位值。
26:24	SMPT0[2:0]	通道采样时间值选择 0 (Sampling time selection 0)，这些位由软件写入，用于选择应用于所有通道的采样时间。 000：2 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 001：4 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 010：8 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 011：16 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 100：32 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 101：64 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 110：128 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 111：256 个 ADC 工作时钟周期（输入 PCLK 预分频的时钟周期）； 注：仅当 ADC 已禁止时（寄存器 ADC0_CFG[0]值为 0），才允许通过软件对这些位写操作。
23:16	Res.	保留，保持为复位值。
15	SMPT_SEL15	通道 15 采样时间选择
14	Res.	保留，保持为复位值。
13	SMPT_SEL13	通道 0~13 采样时间选择（Channel 0~13 sampling time selection），由软件写入这些位配置使用的采样时间。 0：采样时间使用 SMP0[2:0]寄存器的设置。 1：采样时间使用 SMP1[2:0]寄存器的设置。
12	SMPT_SEL12	
11	SMPT_SEL11	
10	SMPT_SEL10	

9	SMPT_SEL9	注：仅当 ADC 已禁止时（寄存器 ADC0_CFG[0]值为 0），才允许通过软件对这些位写操作。
8	SMPT_SEL8	
7	SMPT_SEL7	
6	SMPT_SEL6	
5	SMPT_SEL5	
4	SMPT_SEL4	
3	SMPT_SEL3	
2	SMPT_SEL2	
1	SMPT_SEL1	
0	SMPT_SEL0	

23.4.3 ADC 采样控制寄存器(ADC_COM_SMPC)

偏移地址：0x08
复位值：0x00000000

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

Bit	Field	Description
31	Res.	保留，保持为复位值。
30:29	OVS_RATE [1:0]	过采样率（Oversampling ratio），该域定义过采样率的数值。 00：4x 01：16x 10：64x 11：256x 注：仅当 ADC 已禁止时（寄存器 ADC0_CFG[0]值为 0），才允许通过软件对这些位写操作。
28:26	OVS_RES [2:0]	过采样精度（Oversampling accuracy），该域定义过采样转换值精度。 000~011：12bit 100：13bit 101：14bit 110：15bit 111：16bit 注：仅当 ADC 已禁止时（寄存器 ADC0_CFG[0]值为 0），才允许通过软件对这些位写操作。

25:24	NORM_RES [1:0]	非过采样时精度（Sample accuracy），该域定义采样转换值精度。 00：6bit 01：8bit 10：10bit 11：12bit 注：仅当 ADC 已禁止时（寄存器 ADC0_CFG[0]值为 0），才允许通过软件对这些位写操作。
23:16	Res.	保留，保持为复位值。
15	OVSEN15	通道 15 过采样器使能（Oversampler Enable），此位由软件置 1 和清零。 0：禁止过采样器； 1：使能过采样器； 注：仅当 ADC 已禁止时（寄存器 ADC0_CFG[0]值为 0），才允许通过软件对这些位写操作。
14	Res.	保留，保持为复位值。
13	OVSEN13	通道 0~13 过采样器使能（Oversampler Enable），此位由软件置 1 和清零。 0：禁止过采样器； 1：使能过采样器； 注：仅当 ADC 已禁止时（寄存器 ADC0_CFG[0]值为 0），才允许通过软件对这些位写操作。
12	OVSEN12	
11	OVSEN11	
10	OVSEN10	
9	OVSEN9	
8	OVSEN8	
7	OVSEN7	
6	OVSEN6	
5	OVSEN5	
4	OVSEN4	
3	OVSEN3	
2	OVSEN2	
1	OVSEN1	
0	OVSEN0	

23.4.4 ADC 通道偏移寄存器 0~13,15(ADC_COM_CH_OFFSET0~13,15)

偏移地址：0x20~0x54,0x5C
复位值：0x00000000

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

Bit	Field	Description
31:12	Res.	保留，保持为复位值。
11:0	VALUEx[11:0]	通道 x=0~13,15 偏移值(Offset)，无符号数，由软件写入。 -写入后的值会立即生效，并且包含在采集后的数据中。

23.4.5 ADCx 中断状态寄存器(ADCx_ISR) (x=0,1)

偏移地址：0x100+(0x100*x)
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AWD_CH15	Res.	AWD_CH13	AWD_CH12	AWD_CH11	AWD_CH10	AWD_CH9	AWD_CH8	AWD_CH7	AWD_CH6	AWD_CH5	AWD_CH4	AWD_CH3	AWD_CH2	AWD_CH1	AWD_CH0
w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OVR_EOI	OVR_EOC	OVR_EOT1	OVR_EOT0	OVR_EOS	Res.	EOI	EOC	EOT1	EOT0	EOS	Res.	AWD_SUM	OVR_SUM	EO_SUM
	w1c	w1c	w1c	w1c	w1c		w1c	w1c	w1c	w1c	w1c		w1c	w1c	w1c

Bit	Field	Description
31	AWD_CH15	模拟看门狗 15 标志（Analog watchdog 15 flag)
30	Res.	保留，保持为复位值。
29	AWD_CH13	模拟看门狗 0~13 标志（Analog watchdog 0~13 flag)，当转换电压满足 AWDEN 和 AWDT 寄存器中编程的值时，硬件将该位置 1。通过软件编程为 1 可将该位清零。 0：未发生模拟看门狗事件（或标志事件已通过软件确认并清零)。 1：发生模拟看门狗事件。
28	AWD_CH12	
27	AWD_CH11	
26	AWD_CH10	
25	AWD_CH9	
24	AWD_CH8	
23	AWD_CH7	
22	AWD_CH6	
21	AWD_CH5	
20	AWD_CH4	
19	AWD_CH3	
18	AWD_CH2	
17	AWD_CH1	
16	AWD_CH0	
15	Res.	保留，保持为复位值。
14	OVR_EOI	ADC 插队模式采集数据溢出（ADC Inject data overrun)，该位在发生溢出事件时由硬件置 1，这意味着在 EOI 标志已置 1 时，又发生一次新的通道插队数据转换，通过软件写入 1 可将该位清零。

		0：未发生溢出事件（或标志事件已通过软件确认并清零）。 1：发生溢出。
13	OVR_EOC	ADC 单通道模式采集数据溢出（ADC Single data overrun），该位在发生溢出事件时由硬件置 1，这意味着在 EOC 标志已置 1 时，又发生一次新的单通道数据转换，通过软件写入 1 可将该位清零。 0：未发生溢出事件（或标志事件已通过软件确认并清零）。 1：发生溢出。
12	OVR_EOT1	ADC 硬件触发 1 模式采集数据溢出（ADC Trigger 1 data overrun），该位在发生溢出事件时由硬件置 1，这意味着在 EOT1 标志已置 1 时，又发生一次新的硬件触发源 1 数据转换，通过软件写入 1 可将该位清零。 0：未发生溢出事件（或标志事件已通过软件确认并清零）。 1：发生溢出。
11	OVR_EOT0	ADC 硬件触发 0 模式采集数据溢出（ADC Trigger 0 data overrun），该位在发生溢出事件时由硬件置 1，这意味着在 EOT0 标志已置 1 时，又发生一次新的硬件触发源 0 数据转换，通过软件写入 1 可将该位清零。 0：未发生溢出事件（或标志事件已通过软件确认并清零）。 1：发生溢出。
10	OVR_EOS	ADC 序列扫描模式采集数据溢出（ADC SCAN data overrun），该位在发生溢出事件时由硬件置 1，这意味着在 EOS 标志已置 1 时，又开始新的序列扫描数据转换，通过软件写入 1 可将该位清零。 0：未发生溢出事件（或标志事件已通过软件确认并清零）。 1：发生溢出。
9	Res.	保留，保持为复位值。
8	EOI	插队采集通道数据转换结束标志（End of inject conversion flag）。 当插队通道的每次转换结束，新数据结果出现在 ADC_DATA_INJ 寄存器，硬件将该位置 1，通过软件向该位写入 1 可将该位清零。 0：插队通道转换未完成（或标志事件已通过软件确认并清零）。 1：插队通道转换已完成。
7	EOC	软件单通道模式数据转换结束标志（End of conversion flag）。 当通道的每次转换结束，新数据结果出现在 ADC_DATA_SINGLE 寄存器，硬件将该位置 1，通过软件向该位写入 1 可将该位清零。 0：通道转换未完成（或标志事件已通过软件确认并清零）。 1：通道转换已完成。
6	EOT1	硬件触发 1 单通道模式数据采集结束标志（End of trigger 1 flag），在 HW_TRIG1_CH[3:0]和 HW_TRIG1_SRC[3:0]寄存器配置的触发通道转换结束时，新数据结果出现在 ADC_DATA_TRIG1 寄存器，硬件将该位置 1，通过软件写入 1 可将该位清零。 0：通道转换未完成（或标志事件已通过软件确认并清零）。 1：通道转换已完成。
5	EOT0	硬件触发 0 单通道模式数据采集结束标志（End of trigger 0 flag），在 HW_TRIG0_CH[3:0]和 HW_TRIG0_SRC[3:0]寄存器配置的通道转换结束时，新数据结果出现在 ADC_DATA_TRIG0 寄存器，硬件将该位置

		1，通过软件写入 1 可将该位清零。 0：通道转换未完成（或标志事件已通过软件确认并清零）。 1：通道转换已完成。
4	EOS	序列扫描模式数据采集结束标志（End of sequence flag），在由 ADC_CH_SCAN0 和 ADC_CH_SCAN1 寄存器配置的一系列通道转换结束时，数据结果出现在 ADC_DATA_SCAN0~13 寄存器，硬件将该位置 1，通过软件写入 1 可将该位清零。 0：转换序列未完成（或标志事件已通过软件确认并清零）。 1：转换序列已完成。
3	Res.	保留，保持为复位值。
2	AWD_SUM	模拟看门狗标志（Analog watchdog flag），当 AWD_CH 0~10 状态标志存在 1 时，硬件将该位置 1，通过软件编程为 1 或者通过软件清除 AWD_CH 0~10 所有标志，则可将该位清零。 0：AWD_CH 0~10 未发生模拟看门狗事件（或标志事件已通过软件确认并清零）。 1： AWD_CH 0~10 发生模拟看门狗事件。
1	OVR_SUM	采集数据溢出标志（Data overrun flag），当 OVR_EOI、OVR_EOC、OVR_EOT1、OVR_EOT0 和 OVR_EOS 状态标志存在 1 时，硬件将该位置 1，通过软件编程为 1 或者通过软件清除 OVR_EOI、OVR_EOC、OVR_EOT1、OVR_EOT0 和 OVR_EOS 所有标志，则可将该位清零。 0：未发采集数据溢出事件（或标志事件已通过软件确认并清零）。 1：发采集数据溢出事件。
0	EO_SUM	采集数据完成标志（End of Data flag），当 EOI、EOC、EOT1、EOT0 和 EOS 状态标志存在 1 时，硬件将该位置 1，通过软件编程为 1 或者通过软件清除 EOI、EOC、EOT1、EOT0 和 EOS 所有标志，则可将该位清零。 0：未发采集数据事件（或标志事件已通过软件确认并清零）。 1：发采集数据事件。

23.4.6 ADCx 中断使能寄存器(ADCx_IE)(x=0,1)

偏移地址：0x104+(0x100*x)
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	AWD	OVR	Res.	Res.	Res.	EOI	EOC	EOT1	EOT0	EOS	Res.	Res.	Res.	Res.
		rw	rw				rw	rw	rw	rw	rw				

Bit	Field	Description
31:14	Res.	保留，保持为复位值。
13	AWD	模拟看门狗通道中断使能（Analog watchdog interrupt enable），通过软件将该位置 1 和清零可使能/禁止模拟看门狗中断。 0：禁止模拟看门狗中断； 1：使能模拟看门狗中断；
12	OVR	采集数据溢出中断使能（Data overrun interrupt enable），通过软件将该位置 1 和清零可使能/禁止中断。 0：禁止中断； 1：使能中断；
11:9	Res.	保留，保持为复位值。
8	EOI	通道插队数据采集完成中断使能（Inject interrupt enable），通过软件将该位置 1 和清零可使能/禁止中断。 0：禁止中断； 1：使能中断；
7	EOC	软件单通道数据采集完成中断使能（Single data interrupt enable），通过软件将该位置 1 和清零可使能/禁止中断。 0：禁止中断； 1：使能中断；
6	EOT1	硬件触发 1 单通道数据采集完成中断使能（Trigger 1 data interrupt enable），通过软件将该位置 1 和清零可使能/禁止中断。 0：禁止中断； 1：使能中断；
5	EOT0	硬件触发 0 单通道数据采集完成中断使能（Trigger 0 data interrupt enable），通过软件将该位置 1 和清零可使能/禁止中断。 0：禁止中断； 1：使能中断；
4	EOS	序列通道扫描采集数据完成中断使能（Scan data interrupt enable），通过软件将该位置 1 和清零可使能/禁止中断。 0：禁止中断； 1：使能中断；
3:0	Res.	保留，保持为复位值。

23.4.7 ADCx 配置寄存器(ADCx_CFG)(x=0,1)

偏移地址：0x108+(0x100*x)
复位值：0x00000000

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

Bit	Field	Description
31:1	Res.	保留，保持为复位值。
0	ADC_EN	ADC 使能 (ADC enable)用于使能 ADC，该位通过软件置 1。 0：写入 0 来关闭 ADC。 1：写入 1 来使能 ADC。

23.4.8 ADCx 数据格式寄存器(ADCx_FMT_CR)(x=0,1)

偏移地址：0x10C+(0x100*x)
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALIGN15	Res.	ALIGN13	ALIGN12	ALIGN11	ALIGN10	ALIGN9	ALIGN8	ALIGN7	ALIGN6	ALIGN5	ALIGN4	ALIGN3	ALIGN2	ALIGN1	ALIGN0
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGN15	Res.	SIGN13	SIGN12	SIGN11	SIGN10	SIGN9	SIGN8	SIGN7	SIGN6	SIGN5	SIGN4	SIGN3	SIGN2	SIGN1	SIGN0
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31	ALIGN15	通道 15 数据对齐 (Data alignment)
30	Res.	保留，保持为复位值。
29	ALIGN13	通道 0~13 数据对齐 (Data alignment)，此位由软件置 1 和清零，用于选择右对齐或左对齐。 0：数据右对齐； 1：数据左对齐； 注：仅当 ADC_EN 位清零时（这可确保当前未进行任何转换），才允许通过软件进行配置。
28	ALIGN12	
27	ALIGN11	
26	ALIGN10	
25	ALIGN9	
24	ALIGN8	
23	ALIGN7	
22	ALIGN6	

21	ALIGN5	
20	ALIGN4	
19	ALIGN3	
18	ALIGN2	
17	ALIGN1	
16	ALIGN0	
15	SIGN15	通道 15 数据符号（Data sign）
14	Res.	保留，保持为复位值。
13	SIGN13	
12	SIGN12	
11	SIGN11	
10	SIGN10	
9	SIGN9	通道 0~13 数据符号（Data sign），此位由软件置 1 和清零，用于选择有符号格式或无符号格式。 0：数据为无符号数； 1：数据为有符号数； 注：仅当 ADC_EN 位清零时（这可确保当前未进行任何转换），才允许通过软件进行配置。
8	SIGN8	
7	SIGN7	
6	SIGN6	
5	SIGN5	
4	SIGN4	
3	SIGN3	
2	SIGN2	
1	SIGN1	
0	SIGN0	

23.4.9 ADCx 触发控制寄存器(ADCx_TRIG_CR)(x=0,1)

偏移地址：0x110+(0x100*x)
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HW1_CH[3:0]				HW1_SRC[3:0]				HW0_CH[3:0]				HW0_SRC[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SW_CH[3:0]				Res.	CONT_WAIT[2:0]			STOP	HW1_EN	HW0_EN	SW_START	Res.	MODE[2:0]		
rw	rw	rw	rw		rw	rw	rw	w1c	rw	rw	w1c		rw	rw	rw

Bit	Field	Description
31:28	HW1_CH [3:0]	单通道硬件触发 1 通道选择（Hardware trigger 1 single channel selection），这些位将由软件设置，用于 ADC_DATA_TRIG1 转换数据的输入通道选择。 0000： ADC 模拟输入通道 0；

		0001： ADC 模拟输入通道 1； ..... 1101： ADC 模拟输入通道 13； 1111： ADC 模拟输入通道 15； 注： 仅当 HW_TRIG1_EN 位清零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。
27:24	HW1_SRC [3:0]	硬件触发 1 源选择（Hardware trigger 1 source selection），这些位将由软件设置，用于 HW_TRIG1_CH 通道采样触发信号源选择。 0000：触发信号来源为 TIM0； 0001：触发信号来源为 TIM1； 0010：触发信号来源为 TIM2； 0011：触发信号来源为 TIM3； 0100：触发信号来源为 TIM4； 0101：触发信号来源为 TIM5； 0110：来源为引脚 ADC_TRIG 输入的上升沿触发信号； 0111：来源为引脚 ADC_TRIG 输入的下降沿触发信号； 1000：来源为 EPWM0_ADC_CMP1 触发信号； 1001：来源为 EPWM0_ADC_CMP2 触发信号； 其他值：保留； 注： 仅当 HW_TRIG1_EN 位清零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。
23:20	HW0_CH [3:0]	单通道硬件触发 0 通道选择（Hardware trigger 0 single channel selection），这些位将由软件设置，用于 ADC_DATA_TRIG0 转换数据的输入通道选择。 0000： ADC 模拟输入通道 0； 0001： ADC 模拟输入通道 1； ..... 1101： ADC 模拟输入通道 13； 1111： ADC 模拟输入通道 15； 注： 仅当 HW_TRIG0_EN 位清零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。
19:16	HW0_SRC [3:0]	硬件触发 0 源选择（Hardware trigger 0 source selection），这些位将由软件设置，用于 HW_TRIG0_CH 通道采样触发信号源选择。 0000：触发信号来源为 TIM0； 0001：触发信号来源为 TIM1； 0010：触发信号来源为 TIM2； 0011：触发信号来源为 TIM3； 0100：触发信号来源为 TIM4； 0101：触发信号来源为 TIM5； 0110：来源为引脚 ADC_TRIG 输入的上升沿触发信号； 0111：来源为引脚 ADC_TRIG 输入的下降沿触发信号； 1000：来源为 EPWM0_ADC_CMP1 触发信号； 1001：来源为 EPWM0_ADC_CMP2 触发信号； 其他值：保留；

		注： 仅当 HW_TRIG0_EN 位清零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。
15:12	SW _CH [3:0]	单通道软件触发通道选择（Software trigger single channel selection），这些位将由软件设置，用于 ADC_DATA_SINGLE 转换数据的输入通道选择。 0000： ADC 模拟输入通道 0； 0001： ADC 模拟输入通道 1； ..... 1101： ADC 模拟输入通道 13； 1111： ADC 模拟输入通道 15； 注： 仅当 SW_TRIG_START 位清零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。
11	Res.	保留，保持为复位值。
10:8	CONT_WAIT [2:0]	软件触发通道连续数据采集等待时间，这些位将由软件设置，用于在软件触发连续采集模式下 ADC_DATA_SINGLE 和 ADC_DATA_SCAN 数据再次转换时间间隔设置。 000：间隔时间为 0； 001：间隔时间为 4 个 ADC 工作时钟周期（PCLK 预分频的时钟周期）； 010：间隔时间为 8 个 ADC 工作时钟周期（PCLK 预分频的时钟周期）； 011：间隔时间为 16 个 ADC 工作时钟周期（PCLK 预分频的时钟周期）； 100：间隔时间为 32 个 ADC 工作时钟周期（PCLK 预分频的时钟周期）； 101：间隔时间为 64 个 ADC 工作时钟周期（PCLK 预分频的时钟周期）； 110：间隔时间为 128 个 ADC 工作时钟周期（PCLK 预分频的时钟周期）； 111：间隔时间为 256 个 ADC 工作时钟周期（PCLK 预分频的时钟周期）； 注： 仅当 SW_TRIG_START 位清零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。
7	STOP	ADC 数据转换停止，这些位将由软件设置，用于停止当前正在进行的数据转换。 0：未发生采集数据事件（或标志事件已通过软件配置清零）。 1：停止采集数据事件。
6	HW1_EN	ADC 硬件触发源 1 使能，这些位将由软件设置，用于使能 HW_TRIG1_SRC[3:0]配置的触发源。 0：禁用触发源。 1：使能触发源。
5	HW0_EN	ADC 硬件触发源 0 使能，这些位将由软件设置，用于使能 HW_TRIG0_SRC[3:0]配置的触发源。 0：禁用触发源。 1：使能触发源。
4	SW_START	ADC 数据软件触发，这些位将由软件设置，用于软件触发模式下触发通

		道数据转换。 0：软件未触发采集数据事件。 1：软件触发采集数据事件。 注： 仅当 STOP 位清零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。
3	Res.	保留，保持为复位值。
2:0	MODE[2:0]	ADC 数据采集工作模式配置，这些位将由软件设置，用于选择 ADC 工作的采样模式。 000: 软件触发单次单通道转换； 010: 软件触发单通道连续转换； 011: 软件触发序列连续扫描； 100: 硬件触发单次单通道转换； 101: 硬件触发单次序列扫描； 其它：保留； 注： 仅当 SW_TRIG_START、HW_TRIG0_EN、HW_TRIG1_EN、STOP 位清零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。

23.4.10 ADCx 插队控制寄存器(ADCx_INJ_CR)(x=0,1)

偏移地址：0x114+(0x100*x)
复位值：0x00000000

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

Bit	Field	Description
31:5	Res.	保留，保持为复位值。
4	START	启动通道插队数据采集，这些位将由软件设置，插队通道采集的数据将会由硬件写入 ADC_DATA_INJ 寄存器。 0：未启动通道插队数据采集（或者插队通道数据采集已经完成）； 1：启动通道插队数据采集（或者插队通道数据采集正在进行）； 注： 仅当 INJ_START 位为零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。
3:0	CH [3:0]	启插队通道选择，这些位将由软件设置，用于配置插队数据采集的 ADC 模拟通道。 0000： ADC 模拟输入通道 0；

		0001： ADC 模拟输入通道 1； ..... 1101： ADC 模拟输入通道 13； 1111： ADC 模拟输入通道 15； 注： 仅当 INJ_START 位为零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。

23.4.11 ADCx 模拟看门狗控制寄存器(ADCx_AWD_CR)(x=0,1)

偏移地址：0x118+(0x100*x)
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	WH	WM	WL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
					rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN15	Res.	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:27	Res.	保留，保持为复位值。
26	WH	模拟看门狗上限阈值监测使能，这些位将由软件设置，用于使能监测通道采集数据大于 AWDTH[11:0]设置的值时 AWD_CHx 产生中断标志。 0：禁止启用上限阈值监测； 1：启用上限阈值监测；
25	WM	模拟看门狗窗口阈值监测使能，这些位将由软件设置，用于使能监测通道采集数据小于等于 AWDTH[11:0]且大于等于 AWDTL[11:0]设置的值时 AWD_CHx 产生中断标志。 0：禁止启用窗口阈值监测； 1：启用窗口阈值监测；
24	WL	模拟看门狗下限阈值监测使能，这些位将由软件设置，用于使能监测通道采集数据小于 AWDTL[11:0]设置的值时 AWD_CHx 产生中断标志。 0：禁止启用下限阈值监测； 1：启用下限阈值监测；
23:16	Res.	保留，保持为复位值。
15	EN15	通道 15 模拟看门狗使能（Analog watchdog enable）
14	Res.	保留，保持为复位值。
13	EN13	模拟看门狗使能（Analog watchdog enable），这些位将由软件设置，用于使能 ADC 模拟通道 0~13 对应的采集数据监测。 0：禁止 ADC 通道模拟看门狗
12	EN12	
11	EN11	

10	EN10	1：使能 ADC 通道模拟看门狗
9	EN9	
8	EN8	
7	EN7	
6	EN6	
5	EN5	
4	EN4	
3	EN3	
2	EN2	
1	EN1	
0	EN0	

23.4.12 ADCx 模拟看门狗触发寄存器(ADCx_AWD_TH)(x=0,1)

偏移地址：0x11C+(0x100*x)
复位值：0x00000000

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

Bit	Field	Description
31:16	TH [15:0]	模拟看门狗阈值上限（Analog watchdog higher threshold），通过软件写入这些位可为模拟看门狗定义阈值上限。 注：仅当 AWDEN 0~13,15 位为零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。
15:0	TL [15:0]	模拟看门狗阈值下限（Analog watchdog lower threshold），通过软件写入这些位可为模拟看门狗定义阈值下限。 注：仅当 AWDEN 0~13,15 位为零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。

23.4.13 ADCx 扫描序列寄存器 0(ADCx_SCAN_SQR0)(x=0,1)

偏移地址：0x120+(0x100*x)
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SQ5_CH[3:0]				SQ4_CH[3:0]				SQ3_CH[3:0]				SQ2_CH[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ1_CH[3:0]				SQ0_CH[3:0]				Res.	Res.	Res.	Res.	LEN[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw

Bit	Field	Description
31:28	SQ5_CH [3:0]	序列 0~5 转换通道(Scan sequence 0~5)，，通过软件编程这些位，并将通道编号（0..10）分配给当前次序列转换。 0000：CH0 0001：CH1 ..... 1101：CH13 1111：CH15 注：仅当 SW_TRIG_START、HW_TRIG0_EN、HW_TRIG1_EN、STOP 位清零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。
27:24	SQ4_CH [3:0]	
23:20	SQ3_CH [3:0]	
19:16	SQ2_CH [3:0]	
15:12	SQ1_CH [3:0]	
11:8	SQ0_CH [3:0]	
7:4	Res.	保留，保持为复位值。
3:0	LEN [3:0]	扫描序列长度(Scan sequence length)，通过软件进行配置，用于指定序列扫描采集模式下通道序列的长度。 0000：序列长度为 1； 0001：序列长度为 2； ..... 1101：序列长度为 14； 其他：保留； 注：仅当 SW_TRIG_START、HW_TRIG0_EN、HW_TRIG1_EN、STOP 位清零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。

23.4.14 ADCx 扫描序列寄存器 1(ADCx_SCAN_SQR1)(x=0,1)

偏移地址：0x124+(0x100*x)
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SQ13_CH[3:0]				SQ12_CH[3:0]				SQ11_CH[3:0]				SQ10_CH[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ9_CH[3:0]				SQ8_CH[3:0]				SQ7_CH[3:0]				SQ6_CH[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:28	SQ13_CH [3:0]	序列 6~13 转换通道(Scan sequence 6~13)，通过软件编程这些位，并将通道编号（0..10）分配给当前次序列转换。 0000：CH0 0001：CH1 ..... 1101：CH13 1111：CH15 注：仅当 SW_TRIG_START、HW_TRIG0_EN、HW_TRIG1_EN、STOP 位清零时（这可确保当前未进行任何转换），才允许通过软件执行写操作。
27:24	SQ12_CH [3:0]	
23:20	SQ11_CH [3:0]	
19:16	SQ10_CH [3:0]	
15:12	SQ9_CH [3:0]	
11:8	SQ8_CH [3:0]	
7:4	SQ7_CH [3:0]	
3:0	SQ6_CH [3:0]	

23.4.15 ADCx 单次数据寄存器(ADCx_DR_SINGLE)(x=0,1)

偏移地址：0x128+(0x100*x)
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，保持为复位值。
15:0	DATA [15:0]	软件触发单通道模式转换的数据 (Software trigger converted data)。

23.4.16 ADCx 插队数据寄存器(ADCx_DR_INJ)(x=0,1)

偏移地址：0x12C+(0x100*x)

复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，保持为复位值。
15:0	DATA [15:0]	通道插队模式转换的数据（Channel inject converted data)。

23.4.17 ADCx 硬件触发数据寄存器 0(ADCx_DR_TRIG0)(x=0,1)

偏移地址：0x130+(0x100*x)
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，保持为复位值。
15:0	DATA [15:0]	硬件触发 0 单通道模式转换的数据（Hardware trigger 0 converted data)。

23.4.18 ADCx 硬件触发数据寄存器 1(ADCx_DR_TRIG1)(x=0,1)

偏移地址：0x134+(0x100*x)
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，保持为复位值。
15:0	DATA [15:0]	硬件触发 1 单通道模式转换的数据（Hardware trigger 1 converted data)。

23.4.19 ADCx 扫描数据寄存器 0~13(ADC0_DR_SCAN0~13)(x=0,1)

偏移地址：0x138+(0x100*x) ~ 0x16C+(0x100*x)
复位值：0x00000000

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

Bit	Field	Description
31:16	Res.	保留，保持为复位值。
15:0	DATAn [15:0]	n (n=0~13)序列扫描模式转换的数据（0~13 Sequence scan converted data)。

24 数模转换器(DAC)

24.1 简介

DAC 模块具有固定 12 位电压输出数模转换的功能，可以通过 CPU 更新 DAC 转换数据。DAC 模块支持单通道转换，两路可配置参考电压源（AVDD 和 Bandgap）和 Buffer 输出（驱动能力大于 1mA），并且DAC输出可以作为模拟比较器(ACMP)的输入基准电压。DAC输出如果未配置到引脚DAC_OUT，则 DAC_OUT 引脚可作为通用输入/输出（GPIO）。

24.2 主要特性

- 一个 DAC 模块，单通道输出
- 12 位转换数据，单通道最高达 1M 转换速率
- AVDD 和 Bandgap 两路可选参考电压源
- Buffer 输出，驱动能力大于 1mA
- DAC 输出可作为内部 ACMP 输入基准
- 输出引脚 DAC_OUT 支持 GPIO 复用
- 支持 DAC 输出与片上外设连接
- 低功耗模式下可以关闭 DAC 模拟供电

24.3 DAC 功能

表 24-1 DAC 功能引脚

功能(Function)	描述(Description)
I/O 连接	模拟输出信号功能引脚为 DAC_OUT
ACMP 输入基准	模拟比较器 ACMP0~3 输入 DAC_OUT

24.4 功能说明

24.4.1 DAC 模块框图

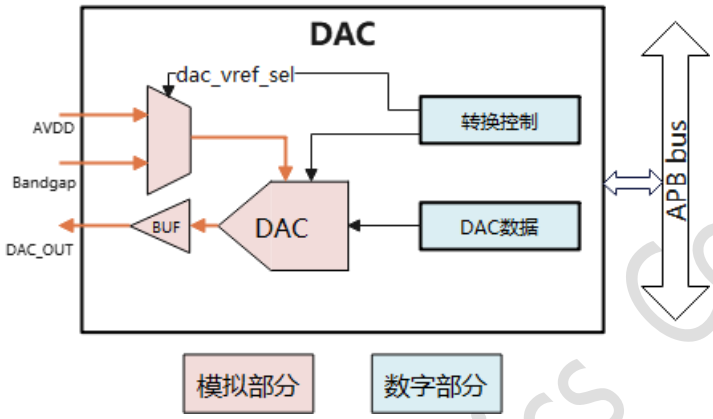

图 24-1 DAC 模块结构框图

24.4.2 DAC 模块功能引脚

表 24-2 DAC 输入/输出引脚

引脚	类型	描述
AVDD	模拟电源基准电压输入	模拟电源，电压范围 2.5~5.5V，基准电压输入为 5V
Bandgap	模拟基准电压输入	工作电压范围 2.5~5.5V，基准电压输入为 2V
DAC_OUT	模拟信号输出	DAC_DATA 转换后的电压信号输出

24.4.3 DAC 数据格式

DAC 正常工作时，支持 CPU 通过字（32 位）和半字（16 位）的方式写 DAC_DR 寄存器跟新转换数据。访问 DAC_DR 寄存器时，只有 DATA[11:0]为有效数据位。

24.4.4 DAC 通道工作方式

DAC 工作时钟开启时，配置 DAC 参考电压源 DAC_CR 寄存器中的位 VREF_SEL 并使能位 EN，DAC 将会根据 DAC_DR 寄存器中的 DATA 值输出 DAC_OUT。
DAC 工作时，可以通过 CPU 写 DAC_DR 寄存器的 DATA 控制 DAC 转换数据输出，在 CPU 完成写 DATA 寄存器操作后，DAC 将会在一段时间 t-settling 后在引脚 DAC_OUT 上输出。

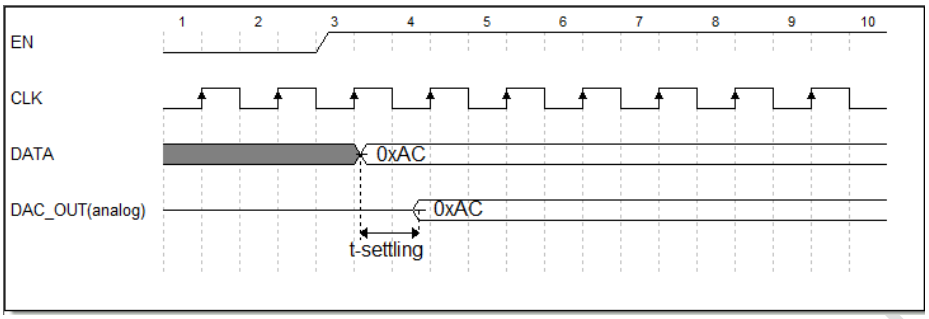

图 24-2 DAC_OUT 输出转换时序图

24.4.5 DAC 输出电压

通过 DAC 模块线性转换后，DAC_DR 寄存器中的 DATA[11:0]值将转换为电压 0.0V 到参考电压（AVDD 或 Bandgap）之间的值输出到 DAC_OUT。
DAC_OUT 引脚的模拟输出电压通过以下公式确定：

$$V_{DAC\_OUT} = V_{REF} \times \frac{DATA}{4096}$$

24.5 DAC 低功耗特性

表 24-3 DAC 低功耗模式特性

模式	功能描述
睡眠（SLEEP）	该模式下 CPU 工作时钟将会关闭，DAC 能够正常使用，并且其他的外设都将保持正常状态
深度睡眠（DEEP_SLEEP）	该模式下 CPU 工作时钟将会关闭，DAC 工作时钟将会根据系统配置控制器（SYSCFG）模块中的 SLEEP_CFG 寄存器的值进行关闭
停止（STOP）	该模式下 DAC 将会关闭，其参考电压源 Bandgap 将会掉电，DAC 输出 I/O DAC_OUT 将会保持电压输出

24.6 寄存器概述

如无特殊说明，下列寄存器均可支持字符（8 位）、半字（16 位）、字（32 位）访问。
DAC 寄存器概述如下表。

表 24-4 DAC 寄存器概述

偏移地址	寄存器名	寄存器描述	复位值
0x00	DAC_CR	DAC 控制寄存器	0x0000
0x04	DAC_DR	DAC 数据寄存器	0x0000

24.6.1 DAC 控制寄存器（DAC_CR）

偏移地址：0x00
复位值：0x00000000

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

Bit	Field	Description
31:2	Reserved	保留，保持为复位值
1	VREF_SEL	参考电压源选择（Reference voltage source selection） 此位由软件设置，选择 AVDD 或 Bandgap 作为 DAC 参考电压源 0：选择 AVDD 为 DAC 参考电压源 1：选择 Bandgap 为 DAC 参考电压源
0	EN	DAC 通道使能（DAC channel enable） 此位由软件置 1 和清零，以使能/禁止 DAC 通道。 0：禁止 DAC 通道 1：使能 DAC 通道

24.6.2 DAC 数据寄存器（DAC_DR）

偏移地址：0x04
复位值：0x0000 0000

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

Bit	Field	Description
31:12	Reserved	保留，保持为复位值。
11:0	DATA	DAC 通道 12 位数据（DAC channel 12-bit data），这些位通过软件写入，用于指定 DAC 转换数据。

25 温度传感器(Temp Sensor)

25.1 简介

温度传感器内置于器件中，可以用于测量器件的结温(T_J)，温度传感器的模拟输出连接到模数转换器(ADC)的输入通道 14，软件可以通过 ADC 采样得到温度数据。温度传感器的输出电压随温度线性变化，由于工艺不同，该线性的偏移量取决于各个芯片。为了提高测量温度传感器的准确性，ADC 在温度传感器的采样通道会对偏移误差进行校准。

25.2 主要特性

支持的温度范围：–40 到 125°C
线性度：最高±5°C，精度取决于校准情况

25.3 功能说明

表 25-1 功能

功能(Function)	描述(Description)
器件内部温度测量	通过 ADC 采集模拟输入通道 15 的电压信号，采集数据经过换算即得测量传感器温度

25.3.1 工作电压源选择

为了方便计算，对温度传感器进行采样时建议 ADC 参考电压源选择 Bandgap，其通过配置寄存器 ADC_COM_CR 中的 VREF_SEL 值选择。

25.4 温度计算

Flash 器件信息区中的 T_REF[15:0]和 V_REF[15:0]存储了标准温度 (单位°C)和标准温度下对应的电压值，Avg_Slope[31:0]存储了温度与电压曲线的斜率的倒数（详见下表）。ADC 采集得到的温度传感器数据 ADC_DATA[15:0]按照如下方式计算得到修正的温度值。

$$T_{temp}\left(°C\right) = (\text{ADC\_DATA} - \text{V\_REF}) \times \frac{\text{Avg}_{\text{Slope}}}{4096} + \text{T\_REF}$$

Flash 地址	存储的参数
0x10001D00	V_REF[15:0]
0x10001D04	Avg_Slope[31:0]
0x10001D08	T_REF[15:0]

更多关于 T_REF、V_REF 和 Avg_Slope 的信息，请参见相应的器件数据手册。

25.5 低功耗特性

表 25-2 低功耗模式特性

模式	功能描述
睡眠（SLEEP）	无影响。
深度睡眠（DEEP_SLEEP）	无影响。
停止（STOP）	该模式下温度传感器将会关闭并掉电，其参考电压源 Bandgap 和 AVDD 将会掉电。

26 模拟比较器（ACMP）

26.1 简介

MCU 内置四个模拟比较器（ACMP），模拟比较器会对正负端输入电压进行比较，输出逻辑电平。

26.2 主要特性

- ACMP 正端输入选择：ACMP_IN0-5，PGA0 和 PGA1 输出
- ACMP 负端输入选择：ACMP_IN0-5，DAC 输出，可编程参考分压 Vref_div
- 支持数字滤波消抖
- 支持比较器输出极性选择
- 支持跳变沿触发中断
- 支持迟滞功能
- 支持比较结果输出到互连矩阵
- 支持使能内嵌反电动势(BEMF)电阻

26.3 功能说明

26.3.1 ACMPx 结构框图

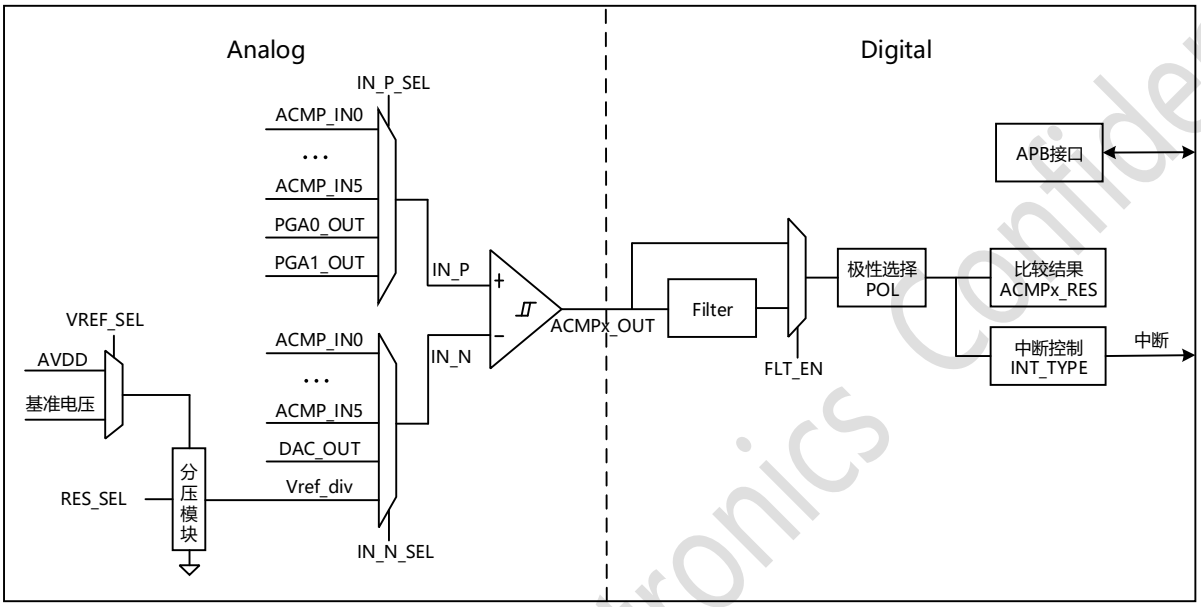

图 26-1 ACMPx 结构框图

26.3.2 ACMPx 输入选择

用作比较器输入的 I/O 口必须在 GPIO 寄存器中配置为模拟模式。

ACMPx 的正负端输入可从多个端口中选择，通过比较器正极选择寄存器(ACMPx_CFG. IN_P_SEL)和比较器负极选择寄存器(ACMPx_CFG.IN_N_SEL)配置。

- **正端选择：ACMPx_CFG.IN_P_SEL**

可选择比较器正端为 ACMP_IN0 ~ ACMP_IN5、PGA0_OUT（PGA0 输出）或 PGA1_OUT（PGA1 输出）端口。

- **负端选择：ACMPx_CFG.IN_N_SEL**

可选择比较器负端为 ACMP_IN0 ~ ACMP_IN5、DAC_OUT 或可编程参考分压 Vref_div。其中，可编程 Vref_div 参考分压源头由参考电压选择寄存器(ACMP_CR.VREF_SEL)配置为 VDD 或基准电压，分压值由电阻选择寄存器(ACMP_CR.RES_SEL)配置，分压范围为输入电压的 2/20~17/20。

ACMPx 的输出结果可通过 ACMP_SR.ACMPx_RES 读取。可通过配置极性控制寄存器(ACMPx_CFG.POL)取反，实现输出高低电平的转换。

每个 ACMP 有独立使能控制 ACMPx_CFG.EN，使能后，需要等待 ACMP_SR.ACMPx_RDY 由硬件置 1，表明 ACMP 准备就绪，所有 ACMP 功能有效。准备时长为 128 个 HSI 时钟周期（约 2.67us）。

26.3.3 ACMPx 滤波功能

ACMPx 内嵌数字滤波器，数字滤波器通过滤波使能(ACMPx_CFG.FLT_EN)开启，用于对模拟比较器输

出进行消抖，禁用时，ACMPx 的比较结果直接输出。
在配置滤波器使能前，需先配置滤波长度寄存器(ACMPx_CFG.FLT_LEN)和滤波采样频率寄存器(ACMPx_CFG.FLT_SAMPLE)，选择滤波采样点数目和滤波采样频率。滤波器以固定采样频率采样输入电平，连续采样到多个采样点的相同电平时，输出该电平。滤波器的工作时钟为 PCLK。

26.3.4 ACMPx 中断

ACMPx 支持异步中断，由中断使能寄存器(ACMPx_CFG.IE)控制，中断触发类型可通过 ACMPx_CFG.INT_TYPE 选择，ACMPx 输出的上升沿、下降沿、上升沿及下降沿事件来触发异步中断。

26.3.5 ACMPx 迟滞功能

ACMPx 具有迟滞功能，开启后在有噪声信号时避免发生意外输出转换。迟滞可在不需要时禁止，配置 ACMP_CR.HYS_EN 为 1 时使能迟滞功能，迟滞电压为 50mV，如图 26-2 所示。

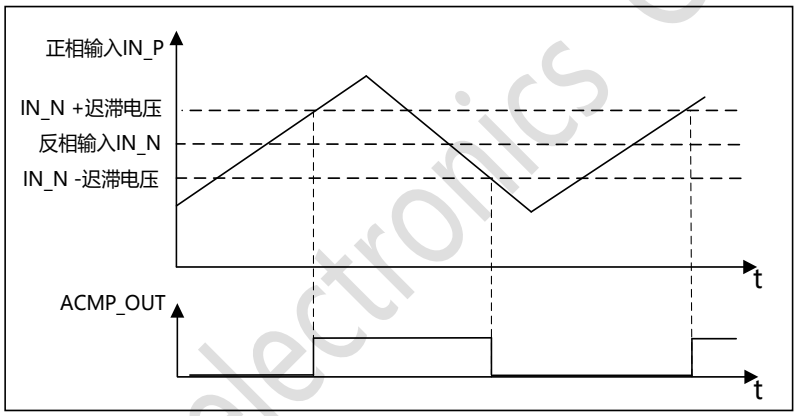

图 26-2 比较器迟滞

26.3.6 反电动势电阻使能

ACMP0~2 的输入端内嵌连接星形电阻，可通过将 ACMP_CR.BEMFR_EN 置 1 启用，给模拟模块选取三相中心电平，如图 26-3 BEMFR 功能示意图图 26-3 所示。

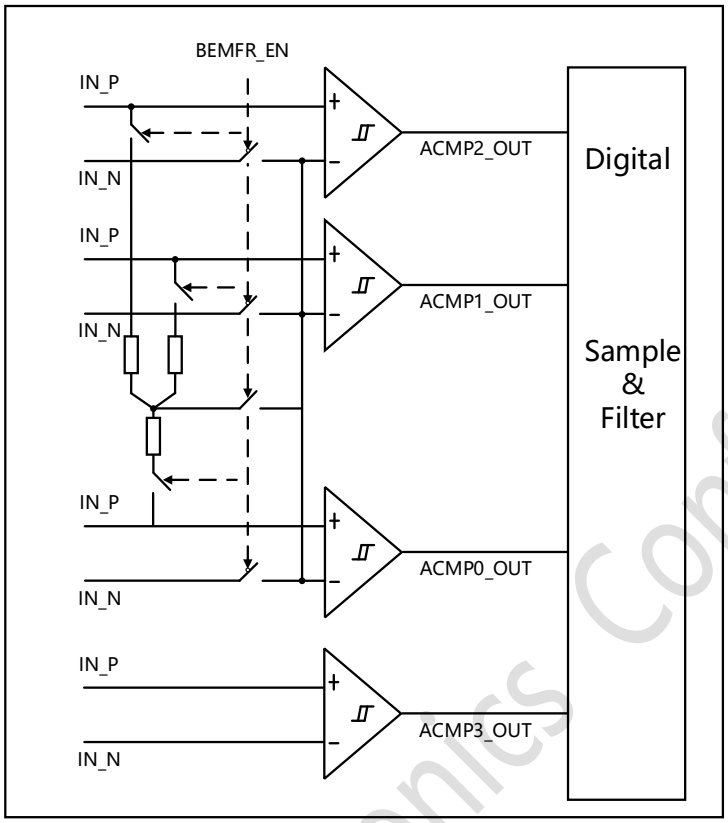

图 26-3 BEMFR 功能示意图

26.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字(16 位)、字(32 位)访问。

表 26-1 ACMP 寄存器概览

偏移地址	寄存器名	寄存器描述	复位值
0x000	ACMP_CR	ACMP控制寄存器	0x00000000
0x004	ACMP_SR	ACMP状态寄存器	0x00000000
0x008	ACMP0_CFG	ACMP0配置寄存器	0x00000000
0x00C	ACMP1_CFG	ACMP1配置寄存器	0x00000000
0x010	ACMP2_CFG	ACMP2配置寄存器	0x00000000
0x014	ACMP3_CFG	ACMP3配置寄存器	0x00000000

26.4.1 控制寄存器（ACMP_CR）

偏移地址：0x000
复位值：0x00000000

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

Bit	Field	Description
31:6	Res.	保留，必须保持复位值。
5	BEMFR_EN	反电动势电阻使能（Back electromotive force resistance enable） 0：禁止 1：使能
4	VREF_SEL	参考电压选择（Reference voltage selection） 0：选择 VDD 作为参考电压源头 1：选择基准电压作为参考电压源头
3:0	RES_SEL	分压选择（Resistance selection） 选择参考电压 Vref 的分压值作为比较器的负端输入 0000：2/20 Vref 0001：3/20 Vref 0010：4/20 Vref 0011：5/20 Vref 0100：6/20 Vref 0101：7/20 Vref 0110：8/20 Vref 0111：9/20 Vref 1000：10/20 Vref 1001：11/20 Vref 1010：12/20 Vref 1011：13/20 Vref 1100：14/20 Vref 1101：15/20 Vref 1110：16/20 Vref 1111：17/20 Vref

26.4.2 状态寄存器（ACMP_SR）

偏移地址：0x004
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACMP3_RDY	ACMP2_RDY	ACMP1_RDY	ACMP0_RDY
												ro	ro	ro	ro
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ACMP3_RES	ACMP2_RES	ACMP1_RES	ACMP0_RES	Res.	Res.	Res.	Res.	ACMP3_IF	ACMP2_IF	ACMP1_IF	ACMP0_IF
				ro	ro	ro	ro					w1c	w1c	w1c	w1c

Bit	Field	Description
31:20	Res.	保留，必须保持复位值。
19	ACMP3_RDY	ACMP3 就绪标记（ACMP3 ready） 0：ACMP3 未准备就绪 1：ACMP3 准备就绪
18	ACMP2_RDY	ACMP2 就绪标记（ACMP2 ready） 0：ACMP2 未准备就绪 1：ACMP2 准备就绪
17	ACMP1_RDY	ACMP1 就绪标记（ACMP1 ready） 0：ACMP1 未准备就绪 1：ACMP1 准备就绪
16	ACMP0_RDY	ACMP0 就绪标记（ACMP0 ready） 0：ACMP0 未准备就绪 1：ACMP0 准备就绪
15:12	Res.	保留，必须保持复位值。
11	ACMP3_RES	ACMP3 比较结果（Result） 0：IN_P 电压低于 IN_N 电压 1：IN_P 电压高于 IN_N 电压
10	ACMP2_RES	ACMP2 比较结果（Result） 0：IN_P 电压低于 IN_N 电压 1：IN_P 电压高于 IN_N 电压
9	ACMP1_RES	ACMP1 比较结果（Result） 0：IN_P 电压低于 IN_N 电压 1：IN_P 电压高于 IN_N 电压
8	ACMP0_RES	ACMP0 比较结果（Result） 0：IN_P 电压低于 IN_N 电压 1：IN_P 电压高于 IN_N 电压
7：4	Res.	保留，必须保持复位值。
3	ACMP3_IF	ACMP3 中断标记（Interrupt Flag） 0：中断无效 1：中断有效，写 1 清 0
2	ACMP2_IF	ACMP2 中断标记（Interrupt Flag） 0：中断无效 1：中断有效，写 1 清 0
1	ACMP1_IF	ACMP1 中断标记（Interrupt Flag） 0：中断无效

		1：中断有效，写 1 清 0
0	ACMP0_IF	ACMP0 中断标记（Interrupt Flag） 0：中断无效 1：中断有效，写 1 清 0

26.4.3 ACMP0 配置寄存器（ACMP0_CFG）

偏移地址：0x008
复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN_N_SEL[2:0]				IN_P_SEL[2:0]		
									rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			FLT_EN	FLT_LEN[1:0]		FLT_SAMPLE[1:0]		Res.	HYS_EN	INT_TYPE[1:0]		IE	Res.	POL	EN
			rw	rw	rw	rw	rw		rw	rw	rw	rw		rw	rw

Bit	Field	Description
31:23	Res.	保留，必须保持复位值。
22：20	IN_N_SEL	负端输入选择： 000：ACMP_IN0 001：ACMP_IN1 010：ACMP_IN2 011：ACMP_IN3 100：ACMP_IN4 101：ACMP_IN5 110：DAC_OUT 111：参考电压 Vref_div
19	Res.	保留，必须保持复位值。
18：16	IN_P_SEL	正端输入选择： 000：ACMP_IN0 001：ACMP_IN1 010：ACMP_IN2 011：ACMP_IN3 100：ACMP_IN4 101：ACMP_IN5 110：PGA0 输出 111：PGA1 输出
15：13	Res.	保留，必须保持复位值。
12	FLT_EN	滤波器使能（Filter enable） 0：关闭滤波器 1：开启滤波器
11：10	FLT_LEN	滤波器滤波长度（Filter length） 00：滤波长度为 1　　01：滤波长度为 8

		10：滤波长度为 16　11：滤波长度为 32
9：8	FILT_SAMPLE	滤波器滤波时钟分频系数选择（Filter clock sample selection） 00：滤波时钟分频系数为 1 01：滤波时钟分频系数为 4 10：滤波时钟分频系数为 16 11：滤波时钟分频系数为 32
7	Res.	保留，必须保持复位值。
6	HYS_EN	迟滞使能 0：禁用迟滞功能 1：使能迟滞功能
5：4	INT_TYPE	中断触发类型（Trigger type） 00：比较结果输出的上升沿触发 01：比较结果输出的下降沿触发 10：比较结果输出的上升沿或下降沿触发 11：保留
3	IE	中断使能（Interrupt enable） 0：中断禁止 1：中断使能
2	Res.	保留，必须保持复位值。
1	POL	极性选择（Polarity） 0：比较器结果输出 1：比较器结果取反输出
0	EN	比较器使能（ACMP enable） 0：禁用比较器 1：使能比较器

26.4.4 ACMP1 配置寄存器（ACMP1_CFG）

偏移地址：0x00C
复位值：0x00000000
寄存器列表同 ACMP0_CFG

26.4.5 ACMP2 配置寄存器（ACMP2_CFG）

偏移地址：0x010
复位值：0x00000000
寄存器列表同 ACMP0_CFG

26.4.6 ACMP3 配置寄存器（ACMP3_CFG）

偏移地址：0x014
复位值：0x00000000
寄存器列表同 ACMP0_CFG

27 可编程增益放大器(PGA)

27.1 简介

MCU 内置有最多 4 路可编程增益放大器，可通过模拟引脚以差分方式输入，PGA 输出可作为 ADC 和模拟比较器 ACMP 输入。

27.2 主要特性

- 支持差分输入模式
- 增益可配：1，2，4，8，16，32
- PGA 的输出在芯片内部可以直接连接至 ADC 模拟输入
- PGA0 和 PGA1 的输出在芯片内部可以直接连接至 ACMP

27.3 功能说明

27.3.1 PGA 模块框图

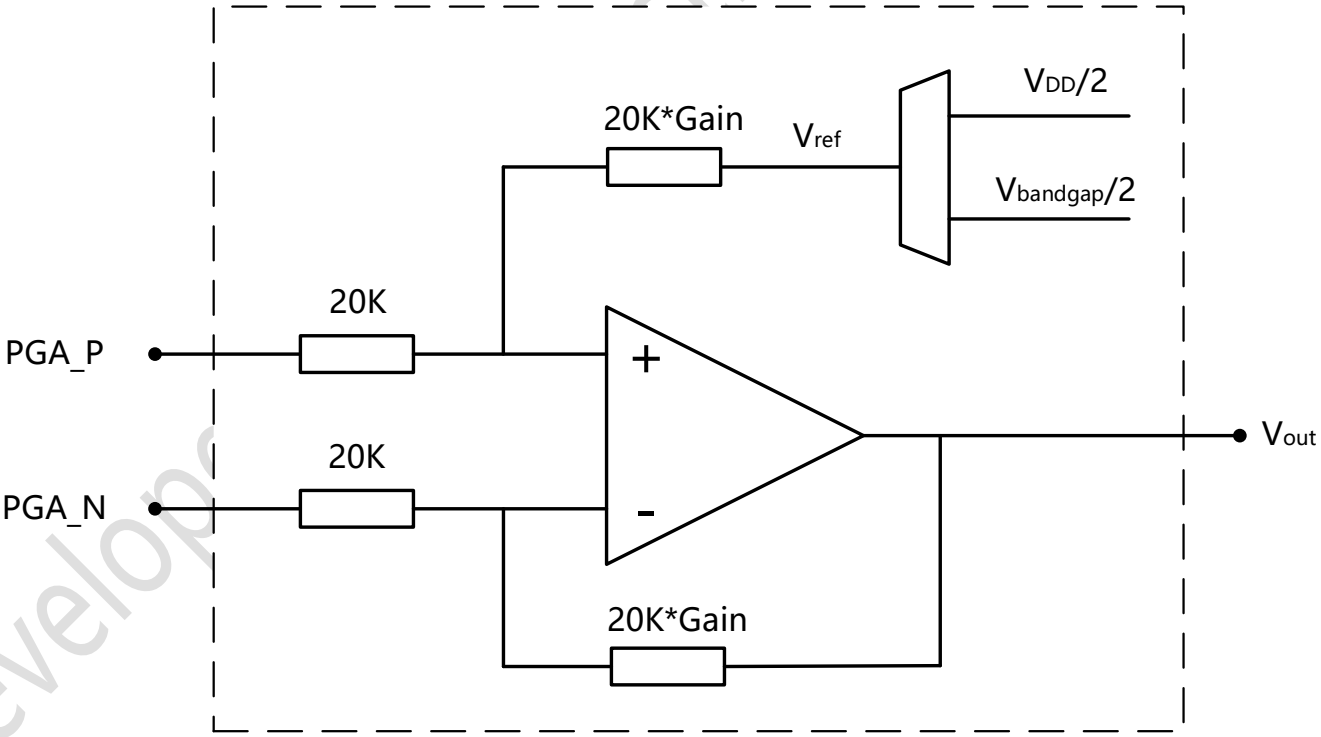

图 27-1 PGA 连接结构图

27.3.2 PGA 差分输入模式

当 PGA_CR.PGAx_MODE 为 1 时，程控增益放大器工作在差分输入模式。
差分输入模式可配置 PGA_CR.OFFSET_EN 为 1，使输出叠加偏置电压，可配置 PGA_CR.OFFSET_SEL 选择偏置电压为 1/2 AVDD 或 1/2 基准电压，所有 PGA 的偏置选择一致。可配置 PGA_CR.PGAx_GAIN 选择放大器增益(1/2/4/8/16/32)。
可配置 PGA_CR.OUT_BUF_EN 为 1 使能输出 Buffer，提高放大器带宽。
差分输入模式输出电压可表示为：

PGAx_OUT = (PGAx_IN_P - PGAx_IN_N) * PGAx_GAIN + OFFSET

27.3.3 低功耗特性

在空闲(IDLE)和睡眠(SLEEP)模式，PGA 无影响。
在停止(STOP)模式，PGA 将会关闭并掉电，无法工作。

27.3.4 寄存器概述

如无特殊说明，下列寄存器均支持字符（8 位）、半字（16 位）、字（32 位）访问。
（注：在配置 PGA_CR 寄存器时，必须对 RCC_LOCK_CR 寄存器写 0x900D0000 解除锁定，才能配置 PGA_CR 寄存器。）

27.3.5 PGA 控制寄存器（PGA_CR）

基地址：0x40001000
偏移地址：0x64
复位值：0x00006000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	PGA3_GAIN[2:0]			Res.	PGA2_GAIN[2:0]			Res.	PGA1_GAIN[2:0]			Res.	PGA0_GAIN[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PGA_SPEED[1:0]		OFFSET_SEL	PGA3_OFFSET_EN	PGA2_OFFSET_EN	PGA1_OFFSET_EN	PGA0_OFFSET_EN	PGA3_MODE	PGA2_MODE	PGA1_MODE	PGA0_MODE	PGA3_EN	PGA2_EN	PGA1_EN	PGA0_EN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31	Res.	保留，保持为复位值。
30:28	PGA3_GAIN[2:0]	PGA3 程控增益配置 000：增益为 1　　001：增益为 2 010：增益为 4　　011：增益为 8

		100：增益为 16　　101：增益为 32 其它值为保留值。
27	Res.	保留，保持为复位值。
26:24	PGA2_GAIN[2:0]	PGA2 程控增益配置 000：增益为 1　　001：增益为 2 010：增益为 4　　011：增益为 8 100：增益为 16　　101：增益为 32 其它值为保留值。
23	Res.	保留，保持为复位值。
22:20	PGA1_GAIN[2:0]	PGA1 程控增益配置 000：增益为 1　　001：增益为 2 010：增益为 4　　011：增益为 8 100：增益为 16　　101：增益为 32 其它值为保留值。;
19	Res.	保留，保持为复位值。
18:16	PGA0_GAIN[2:0]	PGA0 程控增益配置 000：增益为 1　　001：增益为 2 010：增益为 4　　011：增益为 8 100：增益为 16　　101：增益为 32 其它值为保留值。
15	OUT_BUF_EN	PGA 输出 Buffer 使能 1：使能输出 Buffer　　0：禁止输出 Buffer
14:13	PGA_SPEED	PGA 带宽配置 11：PGA 快速模式，必须保持为此模式 其它值为保留值。
12	OFFSET_SEL	PGA 输出偏置选择 1：1/2 基准电压 Bandgap　　0：1/2 AVDD
11	PGA3_OFFSET_EN	PGA3 叠加输出偏置使能 1：叠加偏置输出　　0：输出无偏置
10	PGA2_OFFSET_EN	PGA2 叠加输出偏置使能 1：叠加偏置输出　　0：输出无偏置
9	PGA1_OFFSET_EN	PGA1 叠加输出偏置使能 1：叠加偏置输出　　0：输出无偏置
8	PGA0_OFFSET_EN	PGA0 叠加输出偏置使能 1：叠加偏置输出　　0：输出无偏置
7	PGA3_MODE	PGA3 模式配置 1：差分输入　　0：保留值
6	PGA2_MODE	PGA2 模式配置 1：差分输入　　0：保留值
5	PGA1_MODE	PGA1 模式配置 1：差分输入　　0：保留值
4	PGA0_MODE	PGA0 模式配置 1：差分输入　　0：保留值
3	PGA3_EN	PGA3 使能 1：使能 PGA　　0：关闭 PGA
2	PGA2_EN	PGA2 使能

		1：使能 PGA	0：关闭 PGA
1	PGA1_EN	PGA1 使能 1：使能 PGA	0：关闭 PGA
0	PGA0_EN	PGA0 使能 1：使能 PGA	0：关闭 PGA

28 器件电子签名

器件电子签名为只读存储单元，可以使用调试接口或 CPU 对其进行读取，包含 96 位唯一身份标识码，封装类型，Flash 及 RAM 大小等信息，支持字节（8 位），半字（16 位）和全字（32 位）访问，所有信息为出厂前设置的用户不可更改信息。

28.1 唯一器件 ID 寄存器（UID，96 位）

UID 对于不同器件具有唯一性，可用于作为 Flash 加密密码或者序列号使用。

28.1.1 唯一标识码字段 0(UID_WORD0)

基地址：0x10001C10
偏移地址：0x000
复位值：0xXXXX XXXX

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

Bit	Field	Description
31:0	UID	唯一标识码

28.1.2 唯一标识码字段 1(UID_WORD1)

基地址：0x10001C10
偏移地址：0x004
复位值：0xXXXX XXXX

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

Bit	Field	Description
31:0	UID	唯一标识码

28.1.3 唯一标识码字段 2(UID_WORD2)

基地址：0x10001C10
偏移地址：0x008
复位值：0xXXXX XXXX

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

Bit	Field	Description
31:0	UID	唯一标识码

28.2 器件信息

28.2.1 器件信息字段 0(DEV_INFO0)

基地址：0x10001C10
偏移地址：0x00C
复位值：0xXXXX XXXX

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

Bit	Field	Description
31:28	PROD_TYPE	产品类型 0000: F系列，通用 0001: L系列，低功耗 0010: H系列，高性能 0011: M 系列，电机驱动 其它值为保留值
27:12	SUB_FAM	子系列编码
11:8	PIN_NUM	PIN 数量 0000: 8 pins 0001: 16 pins 0010: 20 pins

		0011: 24 pins 0100: 28 pins 0101: 32 pins 0110: 40 pins 0111: 48 pins 1000: 52 pins 1001: 64 pins 1010: 100 pins 1011: 144 pins 其它值为保留值
7:4	FLASH_SIZE	Flash 空间大小 000：4KB 001：8KB 010：16KB 011：32KB 100：64KB 101：128KB 110：256KB 111：512KB
3:0	PKG_TYPE	封装类型 0000: TSSOP 0001: SSOP 0010: LQPF 0011: QFN 0100: SOP 其它值为保留值

28.2.2 器件信息字段 1(DEV_INFO1)

基地址：0x10001C10
偏移地址：0x010
复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GATE_DRIVER[3:0]				INT_LDO[3:0]				INT_VERSION[3:0]				TEMP[3:0]			
ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PKG_VERSION[3:0]				MCU_VERSION[3:0]				RAM_SIZE[7:0]							
ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro

Bit	Field	Description
31:28	GATE_DRIVER	集成预驱类型 0000：无驱动 0001：6N 预驱 0010：3P3N 预驱

		0011：3P3N 预驱+MOS 0100：6N 预驱+MOS 其它值为保留值
27:24	INT_LDO	集成电源类型 0000：无 LDO 0001：5V 0010：3.3V 0011：12V + 5V 0100：12V + 3.3V 0101：15V + 5V 0111：15V + 3.3V 其它值为保留值
23:20	INT_VERSION	集成预驱/电源版本信息
19:16	TEMP	工作温度范围 3：- 40 to 125℃ 6：- 40 to 85℃ 7：- 40 to 105℃ 其它值为保留值
15:12	PKG_VERSION	封装引脚分布版本信息
11:8	MCU_VERSION	MCU 版本信息
7:0	RAM_SIZE	RAM 空间大小。 000：1KB 001：2KB 010：4KB 011：8KB 100：16KB 101：32KB 110：64KB 111：128KB 其它值为保留值

29 调试支持(Debug support)

29.1 概述

MCU 的内核是 Cortex®-M0，该内核内置硬件扩展以支持高级调试功能。这些调试扩展允许内核在取指（指令断点）或取访问数据（数据断点）时停顿。在内核停顿期间，用户可以查询其内部状态及系统的外部状态。使用完成后，内核和系统会自动恢复并继续正常执行程序。

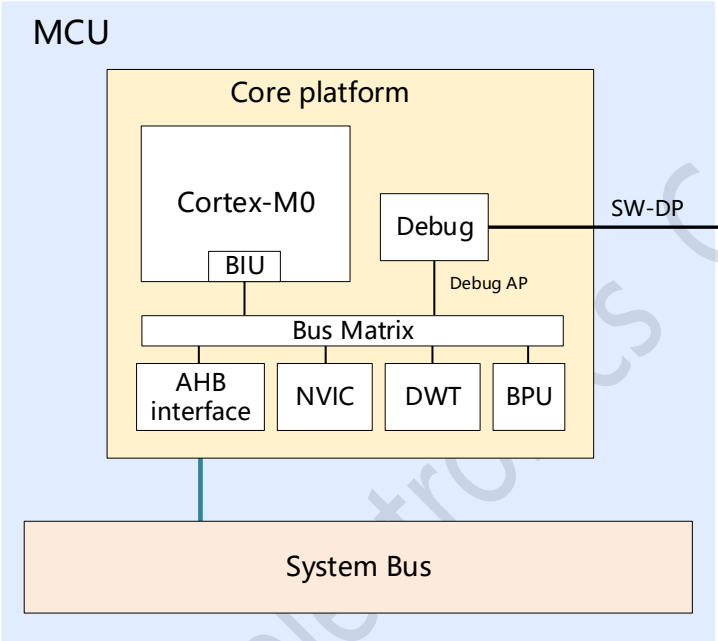

图 29-1 DPM32 MCU 和 Cortex®-M0 调试支持框图

Arm Cortex®-M0 内核集成调试功能。它包括：

- SW-DP：串行数据线
- BPU：断点单元
- DWT：数据观察点与跟踪单元

注:有关 ARM® Cortex®-M0 内核支持的调试功能的详细信息,请参见 Cortex®-M0 技术参考手册。

29.2 SWD 调试端口引脚分配

GPIO 的 PA13 和 PA14 两个引脚默认被用作 SW-DP 调试端口，所有封装都支持这些引脚。复位后，SWD 引脚默认为输入上拉模式。

软件也可在 GPIO 中将 SWD 调试引脚配置为普通 I/O。

表 29-1 SWD 调试端口引脚分配

SW-DP 引脚名称	SW 调试端口		引脚分配	复位后状态
	类型	功能		
SWDIO	输入/输出	串行线数据输入/输出	PA13	输入上拉
SWCLK	输入	串行线时钟	PA14	输入下拉

29.3 ID 代码和锁定机制

29.3.1 器件信息

微控制器内部包含器件信息 DEV_INFO，器件信息定义了 MCU 的产品型号（产品系列、子系列号、版本号、硬件资源）等信息，详见器件电子签名章节，通过用户代码与 SWD 调试接口均能够获取此信息。

29.3.2 JEDEC-106 ID

MCU 内部集成了 JEDEC-106 ID，位于基地址为 0x40003000 的系统 4 KB ROM Table 中，映射地址为 0xE00FF000-0xE00FFFFF 的内部 PPB 总线，通过用户代码与 SWD 调试接口均可访问。

29.4 MCU 调试组件(DBG)

MCU 调试组件帮助调试器为以下各项提供支持：
- 低功耗模式
- 断点或单步调试期间的 EPWM，CCT，TIM，WDG 和 LPTIM 的时钟控制

29.4.1 对低功耗模式的调试支持

在 MCU 处于低功耗模式时，调试器通过 SWD 端口发起连接将会唤醒 MCU，可通过 SWD 访问内核和外设寄存器，对标准调试功能没有任何限制。
当 MCU 已经处于调试状态时，MCU 无法进入低功耗模式。

29.4.2 对 EPWM，CCT，TIM，WDG 和 LPTIM 的调试支持

断点或单步调试期间，可配置 DBG_CFG 寄存器来配置计数器外设（EPWM，CCT，TIM，WDG 和 LPTIM）是否暂停计数。
例如，当 EPWM 控制电机时，可配置计数器暂停或者继续计数。
EPWM 支持在断点或单步调试时，切换输出电平为急停模式输出，详见 EPWM 章节。

29.4.3 禁止 DBG 连接

用户选项中可配置禁止 SWD 连接。请参考嵌入式 Flash(eFlash)章节的用户选项字节说明。

29.5 寄存器概述

如无特殊说明，以下寄存器均可支持字节（8 位），半字（16 位）或字（32 位）访问。
（注：在配置 DBG_CFG 寄存器时，必须对 RCC_LOCK_CR 寄存器写 0x900D0000 解除锁定，才能配置 DBG_CFG 寄存器。）

29.5.1 DBG 配置寄存器(DBG_CFG)

基地址：0x40001028
偏移地址：0x000
复位值：0x0000067F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	EPWM_HALT	LPTIM_HALT	CCT1_HALT	CCT0_HALT	TIM5_HALT	TIM4_HALT	TIM3_HALT	TIM2_HALT	TIM1_HALT	TIM0_HALT	WDG_HALT
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:11	Res.	保留，必须保持复位值。
10	EPWM_HALT	断点或单步调试时，EPWM 计数器是否暂停计数。 0：断点或单步调试时，继续计数。 1：断点或单步调试时，暂停计数。
9	LPTIM_HALT	断点或单步调试时，LPTIM 计数器是否暂停计数。 0：断点或单步调试时，继续计数。 1：断点或单步调试时，暂停计数。
8	CCT1_HALT	断点或单步调试时，CCT1 计数器是否暂停计数。 0：断点或单步调试时，继续计数。 1：断点或单步调试时，暂停计数。
7	CCT0_HALT	断点调试时，CCT0 计数器是否暂停计数。 0：断点调试时，继续计数。 1：断点调试时，暂停计数。
6	TIM5_HALT	断点或单步调试时，TIM5 计数器是否暂停计数。 0：断点或单步调试时，继续计数。 1：断点或单步调试时，暂停计数。
5	TIM4_HALT	断点或单步调试时，TIM4 计数器是否暂停计数。 0：断点或单步调试时，继续计数。 1：断点或单步调试时，暂停计数。
4	TIM3_HALT	断点或单步调试时，TIM3 计数器是否暂停计数。 0：断点或单步调试时，继续计数。 1：断点或单步调试时，暂停计数。
3	TIM2_HALT	断点或单步调试时，TIM2 计数器是否暂停计数。 0：断点或单步调试时，继续计数。 1：断点或单步调试时，暂停计数。
2	TIM1_HALT	断点或单步调试时，TIM1 计数器是否暂停计数。 0：断点或单步调试时，继续计数。 1：断点或单步调试时，暂停计数。
1	TIM0_HALT	断点或单步调试时，TIM0 计数器是否暂停计数。 0：断点或单步调试时，继续计数。

		1：断点或单步调试时，暂停计数。
0	WDG_HALT	断点或单步调试时，WDG 计数器是否暂停计数。 0：断点或单步调试时，继续计数。 1：断点或单步调试时，暂停计数。

30 版本修订说明

版本	修订日期	修订内容
V0.8	2023.11.14	初始版本
V0.9	2024.11.30	更新 ADC 通道描述，增加温度传感器章节
V1.0	2024.07.17	更正 ADCx_TRIG_CR 寄存器中 HW0_EN 和 HW1_EN 的定义
V1.1	2024.09.12	更新 PGA 差分模式输出电压公式中变量名称 更正 DMA、DSP 和 TIM 章节部分图表引用序号
V1.2	2024.10.11	更正 ADC 转换时间计算示例的计算结果
V1.3	2025.03.18	3.4.1 章节 FLASH_CR 寄存器说明中 PE_EN 更正为 PF_EN

31 声明

德普微尽力确保本产品规格书内容的准确和可靠，但是保留在没有通知的情况下，修改规格书内容的权利。客户在下订单前应联系德普微获取最新的相关信息，并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的本公司销售条款与条件。

德普微会不定期更新本文档内容，产品实际参数可能因型号或者其他事项不同有所差异，本文档不作为任何明示或暗示的担保或授权。

本产品规格书未包含任何针对德普微或第三方所有的知识产权的授权。针对本产品规格书所记载的信息，德普微不做任何明示或暗示的保证，包括但不限于对规格书内容的准确性、商业上的适销性，特定目的的适用性或者不侵犯德普微或任何第三人知识产权做任何明示或暗示保证，德普微也不就因本规格书本身及其使用有关的偶然或必然损失承担任何责任。

德普微对应用帮助或客户产品设计不承担任何义务。客户应对其使用本公司的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险，客户应提供充分的设计与操作安全验证。

针对本规格书所披露的内容，在未获得德普微的授权下，任何第三方不得使用、复制、转换，一经发现本公司必依法追究其法律责任，并赔偿由此对本公司造成的一切损失。

请注意在本资料记载的条件范围内使用产品，特别请注意绝对最大额定值、工作电压范围和电气特性等。因在本资料记载的条件范围外使用产品而造成的故障和（或）事故等的损害，本公司对此概不承担任何责任。

本公司一直致力于提高产品的质量和可靠度，但所有的半导体产品都有一定的失效概率，这些失效概率可能会导致一些人身事故、火灾事故等。当设计产品时，请充分留意冗余设计并采用安全指标， 这样可以避免事故的发生。

使用本公司的 IC 生产产品时，如因其产品中对该 IC 的使用方法或产品的规格，或因进口国等原因，包含本 IC 产品在内的制品发生专利纠纷时，本公司概不承担相应责任。