说明

CH32F2x 系列产品是基于 ARM®Cortex™-M3 内核设计的通用微控制器，与大部分 ARM 工具和软件兼容。提供了丰富的通讯接口和控制单元，适用于大部分控制、连接、综合等嵌入式领域。

CH32V2x 和 CH32V3x 系列是基于 32 位 RISC-V 指令集及架构设计的工业级通用微控制器。依据产品性能和资源差异搭配了不同的内核型号。CH32V203x 采用青稞 V4B 内核，支持硬件中断堆栈，提升中断响应效率；CH32V208x 采用青稞 V4C 内核，进一步加快硬件除法运算速度，并增加了内存保护功能；CH32V303x/305x/307x 采用青稞 V4F 内核，进一步支持了硬件浮点运算。该系列产品挂载了丰富的外设接口和功能模块。其内部组织架构满足低成本低功耗嵌入式应用场景。

本手册针对用户的应用开发，提供了 CH32F2x 系列、CH32V2x 系列和 CH32V3x 系列产品的详细使用信息，适用于系列中不同存储器容量、功能资源、封装的产品，若有差异将在对应的功能章节做特殊说明。

CH32F2x 系列产品概览

中小容量通用型(F203)		大容量通用型(F203)		连接型(F205)	互联型(F207)	无线型(F208)
32K 闪存	64K 闪存	128K 闪存	256K 闪存	128K 闪存	256K 闪存	128K 闪存
10K SRAM	20K SRAM	32K SRAM	64K SRAM	32K SRAM	64K SRAM	64K SRAM
2*ADC(TKey) ADTM 2*GPTM 2*USART SPI I2C USBD USBHD CAN RTC 2*WDG 2*OPA	2*ADC(TKey) ADTM 3*GPTM 4*U(S)ART 2*SPI 2*I2C USBD USBHD CAN RTC 2*WDG 2*OPA	2*ADC(TKey) 2*DAC ADTM 3*GPTM 3*USART 2*SPI 2*I2C USBD CAN RTC 2*WDG	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 8*U(S)ART 3*SPI(2*I2S) 2*I2C USBD CAN RTC 2*WDG 4*OPA RNG SDIO FSMC	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 5*U(S)ART 3*SPI(2*I2S) 2*I2C USB-OTG USBHS(+PHY) 2*CAN RTC 2*WDG 4*OPA RNG SDIO	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 8*U(S)ART 3*SPI(2*I2S) 2*I2C USB-OTG USBHS(+PHY) 2*CAN RTC 2*WDG 4*OPA RNG SDIO FSMC DVP ETH-1000MAC 10M-PHY	ADC(TKey) ADTM 3*GPTM GPTM(32) 4*U(S)ART 2*SPI 2*I2C USBD USBHD CAN RTC 2*WDG 2*OPA ETH-10M(+PHY) BLE5.3

注：同一类产品的某些外设数量或功能可能受封装限制，选择时请确认产品封装。

CH32V2x 和 CH32V3x 系列产品概览

中小容量通用型(V203)		大容量通用型(V303)		连接型(V305)	互联型(V307)	无线型(V208)
青稞 V4B 内核		青稞 V4F 内核				青稞 V4C 内核
32K 闪存	64K 闪存	128K 闪存	256K 闪存	128K 闪存	256K 闪存	128K 闪存
10K SRAM	20K SRAM	32K SRAM	64K SRAM	32K SRAM	64K SRAM	64K SRAM
2*ADC(TKey) ADTM 2*GPTM 2*USART SPI I2C USBD USBHD CAN RTC 2*WDG 2*OPA	2*ADC(TKey) ADTM 3*GPTM 4*U(S)ART 2*SPI 2*I2C USBD USBHD CAN RTC 2*WDG 2*OPA	2*ADC(TKey) 2*DAC ADTM 3*GPTM 3*USART 2*SPI 2*I2C USBHD CAN RTC 2*WDG 4*OPA	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 8*U(S)ART 3*SPI(2*I2S) 2*I2C USBHD CAN RTC 2*WDG 4*OPA RNG SDIO FSMC	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 5*U(S)ART 3*SPI(2*I2S) 2*I2C USB-OTG USBHS(+PHY) 2*CAN RTC 2*WDG 4*OPA RNG SDIO	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 8*U(S)ART 3*SPI(2*I2S) 2*I2C USB-OTG USBHS(+PHY) 2*CAN RTC 2*WDG 4*OPA RNG SDIO FSMC DVP ETH-1000MAC 10M-PHY	ADC(TKey) ADTM 3*GPTM GPTM(32) 4*U(S)ART 2*SPI 2*I2C USBD USBHD CAN RTC 2*WDG 2*OPA ETH-10M(+PHY) BLE5.3

注：同一类产品的某些外设数量或功能可能受封装限制，选择时请确认产品封装。

RISC-V 内核版本对比概览

特点 / 内核版本	指令集	硬件堆栈级数	中断嵌套级数	快速中断通道数	整数除法周期	向量表模式	扩展指令	内存保护
青稞 V4B	IMAC	2	2	4	9	地址或指令	支持	无
青稞 V4C	IMAC	2	2	4	5	地址或指令	支持	标准
青稞 V4F	IMAFC	3	8	4	5	地址或指令	支持	标准

寄存器中位属性缩写描述：

寄存器位属性	属性描述
RF	只读属性，读出固定值。
RO	只读属性，由硬件改变。
RZ	只读属性，读操作后自动位清 0
WO	只写属性（不可读，读值不确定）
WA	只写属性，安全模式下可写入。
WZ	只写属性，写操作后自动位清 0
RW	可读，可写。
RWA	可读，安全模式下可写入。
RW1	可读，写 1 有效，写 0 无效。
RW0	可读，写 0 有效，写 1 无效。
RW1T	可读，写 0 无效，写 1 翻转。

CH32 系列 MCU 分类描述：

分类缩写	FLASH 大小	产品分类
D6	32KB 或 64KB	中小容量通用型
D8	128KB 或 256KB	大容量通用型
D8C	128KB 或 256KB	连接型或互联型
D8W	128KB 或 256KB	无线型

注：D（Density）、6（2^6）、8（2^8）

具体分类缩写：

CH32F20x_D6 包括：CH32F203K8、CH32F203C6、CH32F203C8；

CH32F20x_D8 包括：CH32F203CB、CH32F203RC、CH32F203VC、CH32F203RB；

CH32F20x_D8C 包括：CH32F205RB、CH32F207VC；

CH32F20x_D8W 包括：CH32F208RB、CH32F208WB；

CH32V20x_D6 包括：CH32V203F6、CH32V203G6、CH32V203K6、CH32V203F8、CH32V203G8、CH32V203K8、CH32V203C6、CH32V203C8；

CH32V20x_D8 包括：CH32V203RB；

CH32V20x_D8W 包括：CH32V208GB、CH32V208CB、CH32V208RB、CH32V208WB；

CH32V30x_D8 包括：CH32V303CB、CH32V303RB、CH32V303RC、CH32V303VC；

CH32V30x_D8C 包括：CH32V305FB 、CH32V305RB、CH32V307RC 、CH32V307WC 、CH32V307VC；

第 1 章 存储器和总线架构

1.1 总线架构

CH32F2x 系列是基于 ARM®Cortex™-M3 内核设计的微控制器，其构架中的内核、仲裁单元、DMA 模块和 SRAM 存储等部分通过多组总线实现交互。其系统框图见图 1-1。

CH32V2x、CH32V3x 系列产品是基于 RISC-V 指令集设计的通用微控制器，其架构中的内核、仲裁单元、DMA 模块、SRAM 存储等部分通过多组总线实现交互。其系统框图见图 1-2。

图 1-1 CH32F2x 系统框图

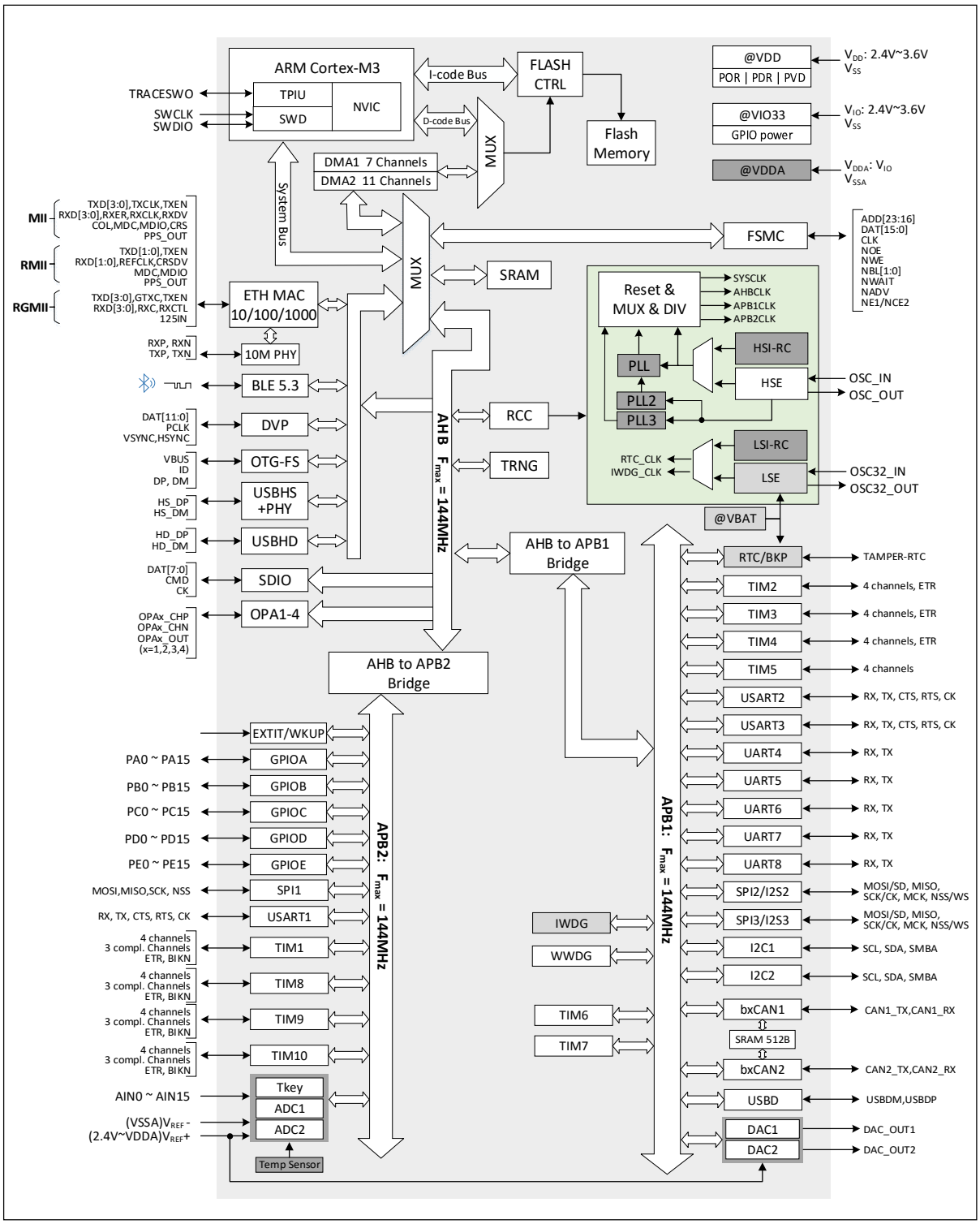
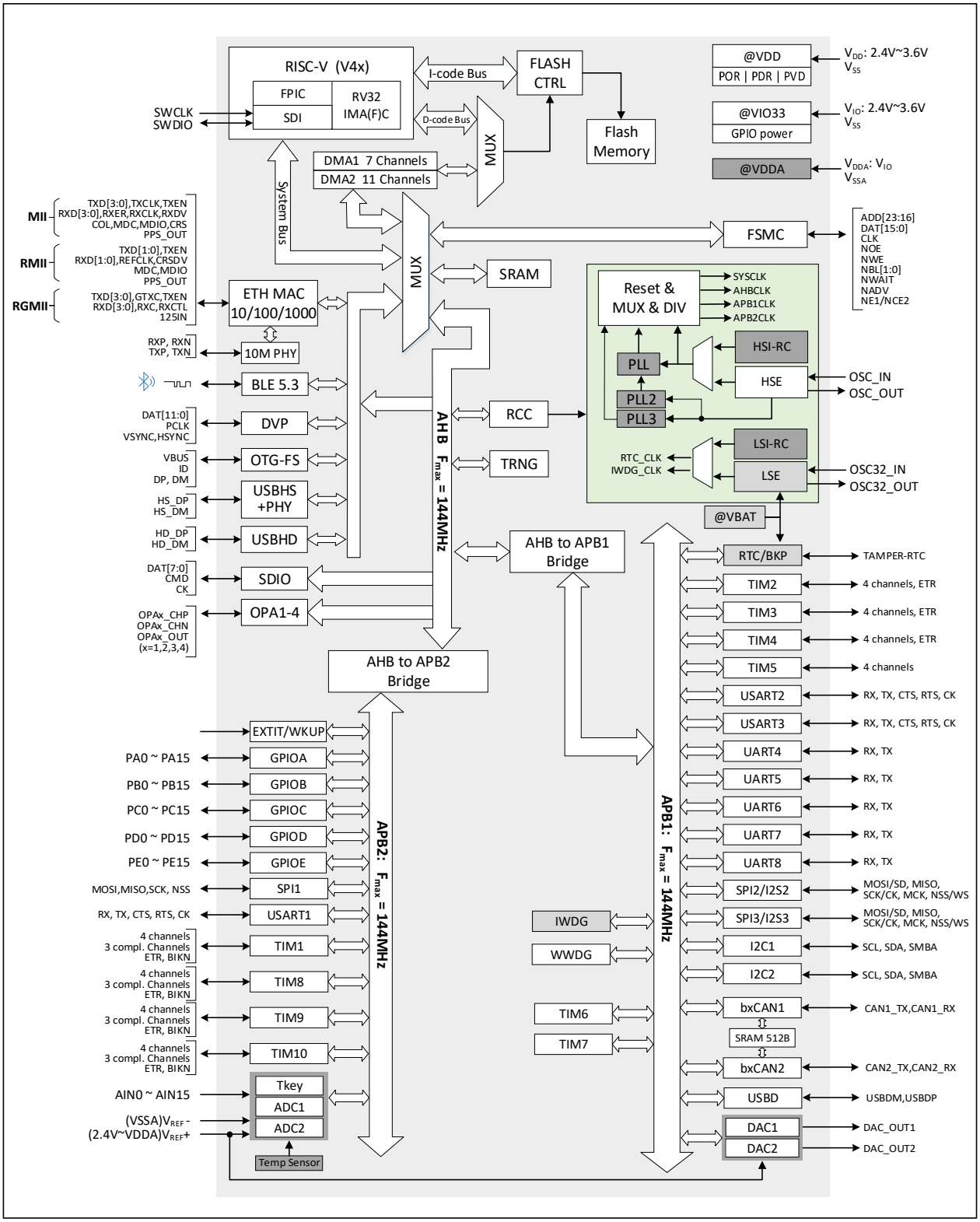


图 1-2 CH32V2x 和 CH32V3x 系统框图

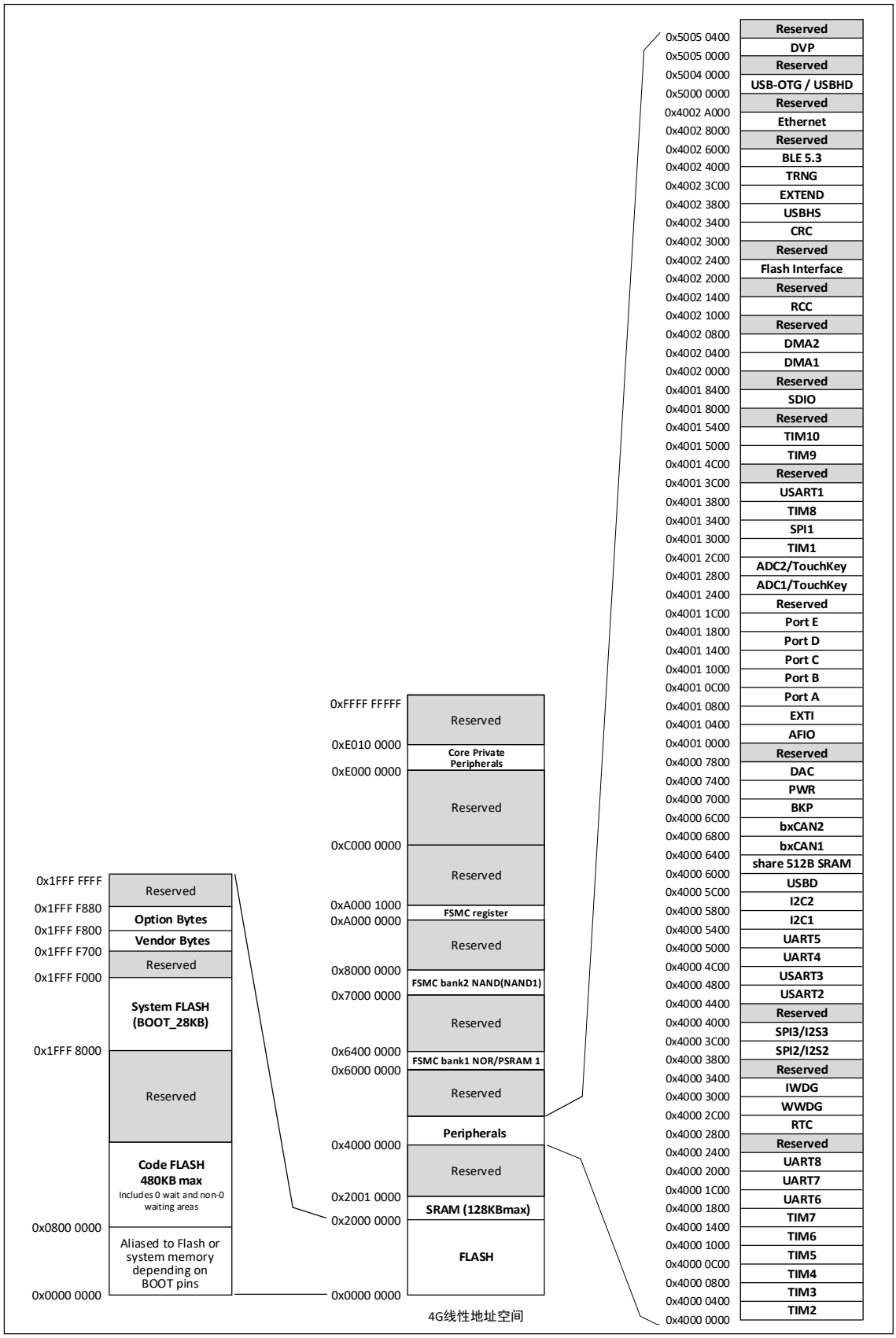


系统中设有：Flash 访问预取机制用以加快代码执行速度；通用 DMA 控制器用以减轻 CPU 负担、提高效率；时钟树分级管理用以降低了外设总的运行功耗，同时还兼有数据保护机制，时钟安全系统保护机制等措施来增加系统稳定性。

- 指令总线(I-Code)将内核和 FLASH 指令接口相连，预取指在此总线上完成。
- 数据总线(D-Code)将内核和 FLASH 数据接口相连，用于常量加载和调试。
- 系统总线将内核和总线矩阵相连，用于协调内核、DMA、SRAM 和外设的访问。
- DMA 总线负责 DMA 的 AHB 主控接口与总线矩阵相连，该总线访问对象是 FLASH 数据、SRAM 和外设。

- 总线矩阵负责的是系统总线、数据总线、DMA 总线、SRAM 和 AHB/APB 桥之间的访问协调。
- AHB/APB 桥，为 AHB 总线和两个 APB 总线提供同步连接。不同的外设挂在不同的 APB 总线下，可以按实际需求配置不同总线时钟，优化性能。

图 1-3 存储映像



1.2 存储器映像

CH32F2x、CH32V2x 和 CH32V3x 系列产品都包含了程序存储器、数据存储器、内核寄存器和外设寄存器等等，它们都在一个 4GB 的线性空间寻址。

系统存储以小端格式存放数据，即低字节存放在低地址，高字节存放在高地址。

1.2.1 位段访问

位操作就是单独读写一个比特位的操作。CH32F2x 产品中通过映射的处理方式提供了对外设寄存器和 SRAM 区内容的位操作读写。具体方法：

1）通过对映射地址区域 32 位的数据进行读操作，读出的值为 0 或非 0，获取目标位域值是 0 或 1；

2）通过对映射地址区域 32 位的数据进行写操作，写入 0 或 1，修改目标位域值为 0 或 1。

地址映射：

目标位域：基地址(BEaddr) +偏移地址(Ofaddr) + 位号(BitN)

映射地址：Mapaddr

Mapaddr = BEaddr +0x2000000 + (Ofaddr×32) + (BitN×4)

举例 1：对 SRAM 区的 0x20000100 地址字节中的 bit3 目标位域进行操作：

Mapaddr= 0x20000000+0x2000000+(0x100*32)+(3*4)= 0x2200200C

则读取 0x2200200C 地址的 4 字节数据内容可知 0x20000100 地址字节中的 bit3 是 0 还是 1；对 0x2200200C 地址执行写 0 或 1 操作，可以修改 0x20000100 地址字节中的 bit3 为 0 还是 1。

举例 2：对外设区域的 0x40021000 地址中的 bit24 进行操作：

Mapaddr = 0x20000000+0x2000000+(0x21000*32)+(24*4)= 0x22420060

则读取 0x22420060 地址的 4 字节数据内容可知 0x40021000 外设地址中的 bit24 是 0 还是 1；对 0x22420060 地址执行写 0 或 1 操作，可以修改 0x40021000 外设地址中的 bit24 为 0 还是 1。

注：CH32V2x 和 CH32V3x 产品不支持位段映射访问方式。

1.2.2 存储器分配

内置最大 128K 字节的 SRAM，起始地址 0x20000000，支持字节、半字(2 字节)、全字(4 字节)访问。

内置最大 480K 字节的程序闪存存储区(CodeFlash)，用于存储用户应用程序。

内置 28K 字节的系统存储器(bootloader)，用于存储系统引导程序（厂家固化自举加载程序）。

内置 128 字节空间用于厂商配置字存储，出厂前固化，用户不可修改。

内置 128 字节空间用于用户选择字存储。

注；储存器分配各型号不同，具体参考 CH32F20xDS0.PDF 和 CH32F20x_30xDS0.PDF。

1.3 启动配置

系统可以通过 BOOT0 和 BOOT1 引脚来选择三种不同的启动模式。

表 1-1 启动模式

BOOT0	BOOT1	启动模式
0	X	从程序闪存存储器启动
1	0	从系统存储器启动
1	1	从内部 SRAM 启动

用户通过设置 BOOT 引脚的状态值来选择复位后的启动模式。系统复位后或者电源复位都会导致 BOOT 引脚的值被重新锁存。

启动模式不同，程序闪存存储器、系统存储器和内部 SRAM 有着不同的访问方式：

- 从程序闪存存储器启动时，程序闪存存储器地址被映射到 0x00000000 地址区域，同时也能够在原地址区域 0x08000000 访问。
- 从系统存储器启动时，系统存储器地址被映射到 0x00000000 地址区域，同时也能够在原地址区域 0x1FFFF000 访问。
- 从内部 SRAM 启动，只能够从 0x20000000 地址区域访问。对于 CH32F2x 系列产品，在此区域启动时，需要通过 NVIC 控制器设置向量表偏移寄存器，重映射向量表到 SRAM 中。对于 CH32V2x 和 CH32V3x 系列产品无需此动作。

第 2 章 电源控制（PWR）

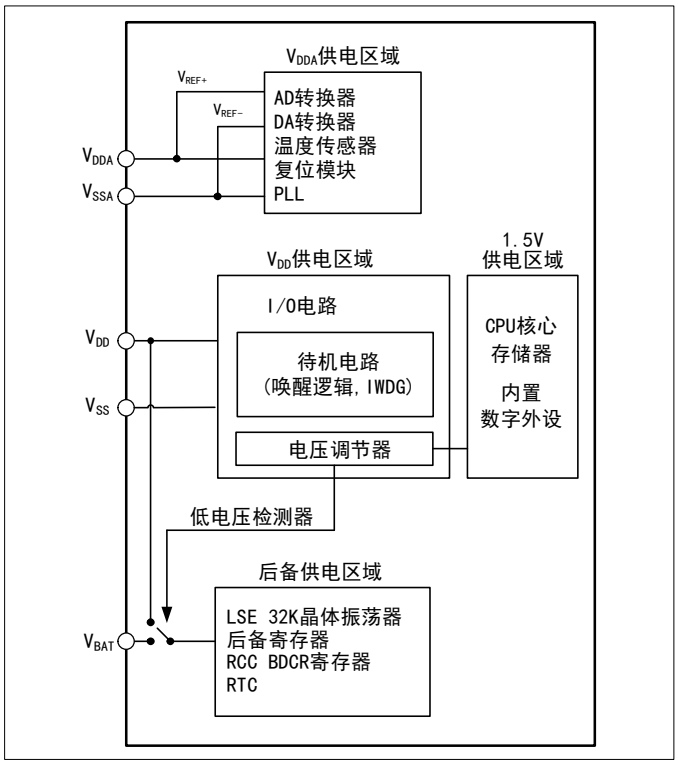
本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

2.1 概述

系统工作电压 V_{DD} 范围为 2.3～3.6V，内置电压调节器提供内核所需的 1.5V 电源。当主电源 V_{DD} 掉电后，电池等后备电源可通过 V_{BAT} 引脚为实时时钟(RTC)和后备寄存器提供电源，如果无需后备电源，建议将 V_{DD} 直接连接到 V_{BAT} 引脚上。

V_{DDA} 和 V_{SSA} 引脚专门为系统中模拟相关电路供电，包括 ADC、DAC、温度传感器等。V_{REF+}和 V_{REF-}作为一些模拟电路的参考点，在芯片内部等于 V_{DDA} 及 V_{SSA}。实际应用中 V_{DDA} 和 V_{SSA} 必须连接到 V_{DD} 和 V_{SS} 端。

图 2-1 电源结构框图



在主电源 V_{DD} 掉电后，模拟开关切换至 V_{BAT}，后备区域由 V_{BAT} 引脚供电，此时 PC13～15 无法作为 GPIO，仅可使用如下功能：

- PC13 可以作为 TAMPER 引脚、RTC 闹钟或秒输出。
- PC14 和 PC15 只能用作 LSE 引脚。

当主电源 V_{DD} 上电稳定后，系统自动切换后备区域由 V_{DD} 供电，PC13～15 可以用作 GPIO 功能。

当 PC13～15 引脚作为 GPIO 输出时，速度必须限制在 2MHz 以下，最大负载电容为 30pF，并且禁止用在持续输出和吸入电流的场合，比如 LED 驱动。

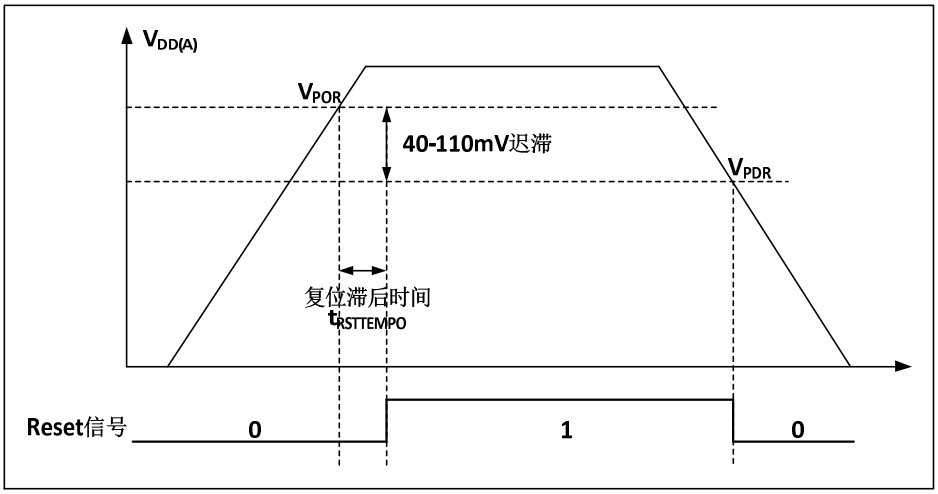
注：在主电源 V_{DD} 恢复供电过程中，内部 V_{BAT} 电源仍然通过对应的 V_{BAT} 引脚连在外部备用电源上，若 V_{DD} 在小于复位滞后时间 $t_{RSTTEMPO}$ 内就达到稳定，并且高于 V_{BAT} 的值 0.6V 以上，则有可能存在较短瞬间，电流通过 V_{DD} 与 V_{BAT} 之间的二极管灌入 V_{BAT}，进而通过 V_{BAT} 引脚注入电池等后备电源，如果后备电源无法承受这样瞬时注入电流，建议在后备电源和 V_{BAT} 引脚之间加一只正向导通低压降二极管。

2.2 电源管理

2.2.1 上电复位和掉电复位

系统内部集成了上电复位 POR 和掉电复位 PDR 电路，当芯片供电电压 V_{DD} 和 V_{DDA} 低于对应门限电压时，系统被相关电路复位，无需外置额外的复位电路。上电门限电压 V_{POR} 和掉电门限电压 V_{PDR} 的参数请参考对应的数据手册。

图 2-2 POR 和 PDR 的工作示意图



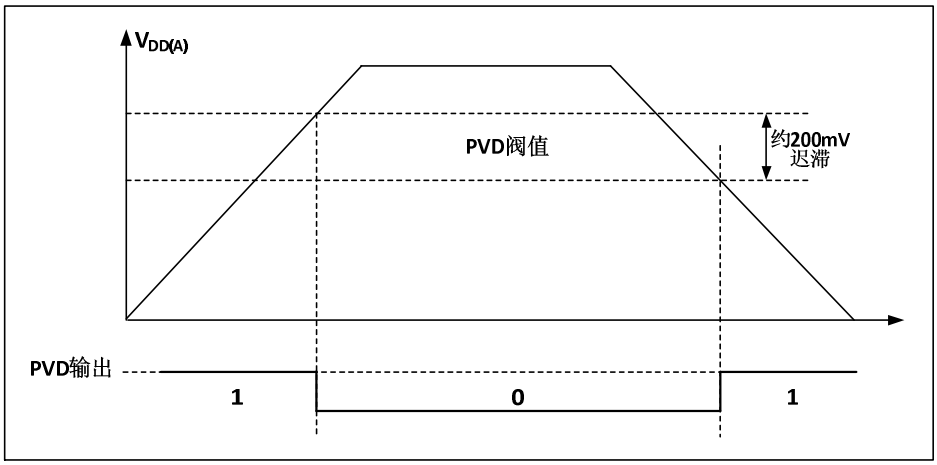
2.2.2 可编程电压监视器

可编程电压监视器 PVD，主要被用于监控系统主电源的变化，与电源控制寄存器 PWR_CTLR 的 PLS[2:0]所设置的门槛电压相比较，配合外部中断寄存器（EXTI）设置，可产生相关中断，以便及时通知系统进行数据保存等掉电前操作。

具体配置如下：

1） 设置 PWR_CTLR 寄存器的 PLS[2:0]域，选择要监控电压阈值。
2） 可选的中断处理。PVD 功能内部连接 EXTI 模块的第 16 线的上升/下降边沿触发设置，开启此中断（配置 EXTI），当 V_{DD} 下降到 PVD 阀值以下或上升到 PVD 阀值之上时就会产生 PVD 中断。
3） 设置 PWR_CTLR 寄存器的 PVDE 位来开启 PVD 功能。
4） 读取 PWR_CSR 状态寄存器的 PVDO 位可获取当前系统主电源与 PLS[2:0]设置阈值关系，执行相应软处理。

图 2-3 PVD 的工作示意图



2.3 低功耗模式

在系统复位后，微控制器处于正常工作状态（运行模式），此时可以通过降低系统主频或者关闭不用外设时钟或者降低工作外设时钟来节省系统功耗。如果系统不需要工作，可设置系统进入低功耗模式，并通过特定事件让系统跳出此状态。

微控制器目前提供了 3 种低功耗模式，从处理器、外设、电压调节器等的工作差异上分为：

- 睡眠模式：内核停止运行，所有外设（包含内核私有外设）仍在运行。
- 停止模式：停止所有时钟，唤醒后系统继续运行。
- 待机模式：停止所有时钟，唤醒后微控制器复位（电源复位）。

表 2-1 低功耗模式一览

模式	进入	唤醒源	对时钟的影响	电压调节器
睡眠	WFI	任意中断唤醒	内核时钟关闭，其他时钟无影响	正常
	WFE	唤醒事件唤醒		
停止	SLEEPDEEP 置 1 PDDS 清 0 WFI 或 WFE	任一外部中断/事件(在外部中断寄存器中设置)、WKUP 引脚上升沿	关闭 HSE、HSI、PLL 和外设时钟	正常：LPDS=0 或 低功耗：LPDS=1
待机	SLEEPDEEP 置 1 PDDS 置 1 WFI 或 WFE	WKUP 引脚上升沿、RTC 闹钟事件、NRST 引脚复位、IWDG 复位。 *注：任意外部中断/事件也可以唤醒系统，但唤醒后系统不复位。*	关闭 HSE、HSI、PLL 和外设时钟	正常：LPDS=0 或 低功耗：LPDS=1

注：SLEEPDEEP 位属于内核私有外设控制位，CH32F2x 产品参考 Cortex-M3 内核手册，CH32V2x 和 CH32V3x 产品参考 PFIC_SCTLR 寄存器。

2.3.1 低功耗配置选项

- WFI 和 WFE 方式

WFI：微控制器被具有中断控制器响应的中断源唤醒，系统唤醒后，将最先执行中断服务函数（微控制器复位除外）。

WFE：唤醒事件触发微控制器将退出低功耗模式。唤醒事件包括：

1） 配置一个外部或内部的 EXTI 线为事件模式，此时无需配置中断控制器；
2） 或者配置某个中断源，等效为 WFI 唤醒，系统优先执行中断服务函数；
3） 或者配置 SLEEPONPEN 位，开启外设中断使能，但不开启中断控制器中的中断使能，系统唤醒后需要清除中断挂起位。

- SLEEPONEXIT

启用：执行 WFI 或 WFE 指令后，微控制器确保所有待处理的中断服务退出后进入低功耗模式。

不启用：执行 WFI 或 WFE 指令后，微控制器立即进入低功耗模式。

- SEVONPEND

启用：所有中断或者唤醒事件都可以唤醒通过执行 WFE 进入的低功耗。

不启用：只有在中断控制器中使能的中断或者唤醒事件可以唤醒通过执行 WFE 进入的低功耗。

2.3.2 睡眠模式

此模式下，所有的 IO 引脚都保持他们运行模式下的状态，所有的外设时钟都正常，所以进入睡眠模式前，尽量关闭无用的外设时钟，以减低功耗。该模式唤醒所需时间最短。

进入：配置内核寄存器控制位 SLEEPDEEP=0，电源控制寄存器 PDDS=0，LPDS 决定内部调压器状态，执行 WFI 或 WFE，可选 SEVONPEND 和 SLEEPONEXIT。

退出：任意中断或者唤醒事件。

2.3.3 停止模式

停止模式是在内核的深睡眠模式（SLEEPDEEP）基础上结合了外设的时钟控制机制，并让电压调节器的运行处于更低功耗的状态。此模式高频时钟（HSE/HSI/PLL）域被关闭，SRAM 和寄存器内容保持，IO 引脚状态保持。该模式唤醒后系统可以继续运行，HSI 称为默认系统时钟。

如果正在进行闪存编程，直到对内存访问完成，系统才进入停止模式；如果正在进行对 APB 的访问，直到对 APB 访问完成，系统才进入停止模式。

停止模式下可工作模块：独立看门狗（IWDG）、实时时钟（RTC）、低频时钟（LSI/LSE）。

进入：配置内核寄存器控制位 SLEEPDEEP=1，电源控制寄存器的 PDDS=0，可选 LPDS 位，执行 WFI 或 WFE，可选 SEVONPEND 和 SLEEPONEXIT。

退出：任一外部中断/事件(在外部中断寄存器中设置)、WKUP 引脚上升沿。

2.3.4 待机模式

待机模式对比停止模式，唯一的差别在于：在某些指定的唤醒条件下退出后，微控制器将被复位，并且执行的是电源复位。

待机模式下可工作模块：独立看门狗（IWDG）、实时时钟（RTC）、低频时钟（LSI/LSE）。

进入：配置内核寄存器控制位 SLEEPDEEP=1，电源控制寄存器的 PDDS=1，执行 WFI 或 WFE，可选 SEVONPEND 和 SLEEPONEXIT。

退出：

1） 任一外部中断/事件(在外部中断寄存器中设置)，此唤醒等效停止模式退出。
2） WKUP 引脚的上升沿、RTC 闹钟事件的上升沿、NRST 引脚上外部复位、IWDG 复位，此唤醒后微控制器执行电源复位。

在待机模式下，当正常供电时，通过配置 PWR_CTLR 寄存器的 R2K_STYEN=1 控制 2K 字节 RAM 不掉电，R30K_STYEN=1 控制 30K 字节 RAM 不掉电；当使用 VBAT 供电时，通过配置 PWR_CTLR 寄存器的 R2K_VBATEN =1 控制 2K 字节 RAM 不掉电，R32K_VBATEN =1 控制 30K 字节 RAM 不掉电。在该基础之上，可以通过配置 PWR_CTLR 寄存器的 RAM_LVEN=1，使能低电压模式，开启该功能，功耗相对更低。

注：调试模式下，使微处理器进入停止或待机模式，将失去调试连接。

2.3.5 RTC 自动唤醒

RTC 可以实现无需外部中断的情况下自动唤醒。通过对时间基数进行编程，可周期性地从停止或待机模式下唤醒。

可选择精准的外部低频 32.768kHz 晶振 LSE 作为 RTC 时钟源，也可以选择内部 LSI 振荡器作为 RTC 时钟源，LSI 的精度和功耗指标要差于 LSE。

RTC 闹钟事件能够把 MCU 从停机模式下唤醒，为了实现此功能，需要把外部中断线 17 配置为上升沿中断，并且把 RTC 设置成可产生闹钟事件。而从待机模式下唤醒，仅需把 RTC 设置成可产生闹钟事件。

2.4 寄存器描述

表 2-2 PWR 相关寄存器列表

名称	访问地址	描述	复位值
R32_PWR_CTLR	0x40007000	电源控制寄存器	0x00000000
R32_PWR_CSR	0x40007004	电源控制/状态寄存器	0x00000000

2.4.1 电源控制寄存器（PWR_CTLR）

偏移地址：0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											RAM_LVEN	R30K_VBATEN	R2K_VBATEN	R30K_STYEN	R2K_STYEN

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DBP	PLS[2:0]			PVDE	CSBF	CWUF	PDDS	LPDS

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留	0
20	RAM_LVEN	RW	RAM 工作在低电压模式使能控制位（功耗相对更低）： 1：开启；　　0：关闭。	0
19	R30K_VBATEN	RW	VBAT 供电时，Standby 模式下 30K RAM 是否带电控制位： 1：带电；　　0：不带电。 *注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C 、 CH32V20x_D8 、 CH32V20x_D8W 、CH32F20x_D8。*	0
18	R2K_VBATEN	RW	VBAT 供电时，Standby 模式下 2K RAM 是否带电控制位： 1：带电；　　0：不带电。	0
17	R30K_STYEN	RW	Standby 模式下 30K RAM 是否带电控制位： 1：带电；　　0：不带电。 *注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C 、 CH32V20x_D8 、 CH32V20x_D8W 、CH32F20x_D8。*	0
16	R2K_STYEN	RW	Standby 模式下 2K RAM 是否带电控制位： 1：带电；　　0：不带电。	0
[15:9]	Reserved	RO	保留	0
8	DBP	RW	后备区域的写使能。当 RTC 时钟为外部时钟的 128 分频时，该位必须设置为 1。 1：允许写 RTC 和后备寄存器； 0：禁止写 RTC 和后备寄存器。	0
[7:5]	PLS[2:0]	RW	PVD 电压监视阈值设置。详细说明见数据手册中电气特性部分。 000：上升沿 2.37V/下降沿 2.29V； 001：上升沿 2.55V/下降沿 2.46V； 010：上升沿 2.63V/下降沿 2.55V； 011：上升沿 2.76V/下降沿 2.67V； 100：上升沿 2.87V/下降沿 2.78V； 101：上升沿 3.03V/下降沿 2.93V； 110：上升沿 3.18V/下降沿 3.06V； 111：上升沿 3.29V/下降沿 3.19V。	0
4	PVDE	RW	电源电压监视功能使能标志位 1：开启电源电压监视功能； 0：禁止电源电压监视功能。	0

3	CSBF	RW1	清除待机状态标志位，读出始终为 0。 1：置 1 清除 SBF 待机状态标志位； 0：清 0 无效。	0
2	CWUF	RW1	清除唤醒状态标志位，读出始终为 0。 1：置 1 后 2 个系统时钟周期后清除 WUF 标志位； 0：清 0 无效。	0
1	PDDS	RW	掉电深睡眠情景下，待机/停机模式选择位。 1：进入待机模式，电压调节器状态由 LPDS 控制； 0：进入停机模式。	0
0	LPDS	RW	停机模式下，电压调节器工作模式选择位。PDDS=0，该位有效。 1：电压调节器工作在低功耗模式； 0：电压调节器工作在正常模式。	0

注：此寄存器从待机模式唤醒时复位。

2.4.2 电源控制/状态寄存器（PWR_CSR）

偏移地址：0x04

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

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

位	名称	访问	描述	复位值
[31:9]	Reserved	RO	保留。	0
8	EWUP	RW	WKUP 引脚使能位 1：WKUP 强制配置为输入下拉状态，用于把 MCU 从待机状态下唤醒； 0：WKUP 引脚可用于通用 IO，无待机唤醒功能。	0
[7:3]	Reserved	RO	保留。	0
2	PVDO	RO	PVD 输出状态标志位。当 PWR_CTLR 寄存器的 PVDE=1 时，该位有效。 1：VDD 和 VDDA 低于 PLS[2:0]设定的 PVD 阈值； 0：VDD 和 VDDA 高于 PLS[2:0]设定的 PVD 阈值。	0
1	SBF	RO	待机状态标志位，可通过 CSBF 位置 1 清除。 1：MCU 进入待机模式； 0：MCU 不在待机模式。	0
0	WUF	RO	唤醒事件状态标志位，可通过 CWUF 位置 1 清除。 1：在 WKUP 引脚检测到唤醒事件或 RTC 闹钟事件； 0：没有唤醒事件发生。	0

注：此寄存器从待机模式唤醒后保持不变。

第 3 章 复位、扩展和时钟控制(RCC)

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

控制器根据电源区域的划分以及应用中的外设功耗管理考虑，提供了不同的复位形式以及可配置的时钟树结构。此章节描述了系统中各个时钟的作用域。

3.1 主要特性

- 多种复位形式
- 多路时钟源，总线时钟管理
- 内置外部晶体振荡监测和时钟安全系统
- 各外设时钟独立管理：复位、开启、关闭
- 支持内部时钟输出

3.2 复位

控制器提供了 3 种复位形式：电源复位、系统复位和后备区域复位。

3.2.1 电源复位

电源复位发生时，将复位除了后备区域外的所有寄存器（后备区域由 V_{BAT} 供电），应用程序中的 PC 指针固定在地址 0x00000004（Reset 向量表）。

其产生条件包括：

- 上电/掉电复位(POR/PDR 复位)
- 从待机模式下唤醒

3.2.2 系统复位

系统复位发生时，将复位除了控制/状态寄存器 RCC_RSTSCKR 中的复位标志和后备区域外的所有寄存器。通过查看 RCC_RSTSCKR 寄存器中的复位状态标志位识别复位事件来源。

其产生条件包括：

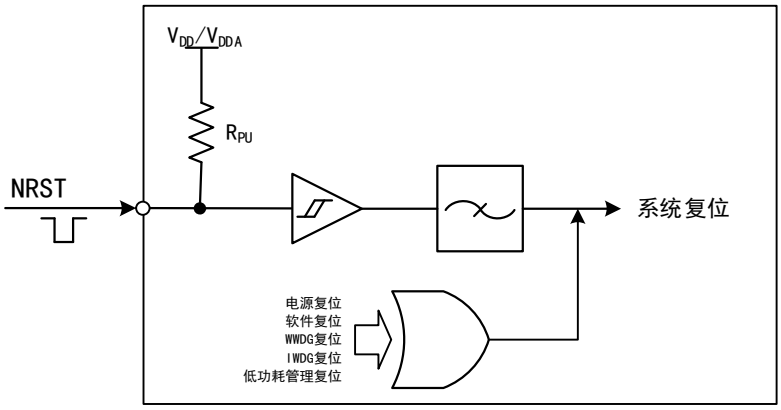
- NRST 引脚上的低电平信号（外部复位）
- 窗口看门狗计数终止(WWDG 复位)
- 独立看门狗计数终止(IWDG 复位)
- 软件复位(SW 复位)
- 低功耗管理复位

窗口/独立看门狗复位：由窗口/独立看门狗外设定时器计数周期溢出触发产生，详细描述看其相应章节。

软件复位：CH32F2x 产品通过内核寄存器 AIRCR 中的 bit2 置 1 复位系统，具体操作请参考 Cortex-M3 内核手册获得更详细信息。CH32V2x 和 CH32V3x 产品通过可编程中断控制器 PFIC 中的中断配置寄存器 PFIC_CFGR 的 SYSRESET 位置 1 复位系统或配置寄存器 PFIC_SCTLR 的 SYSRESET 位置 1 复位系统柜，具体参考对应章节。

低功耗管理复位：通过将用户选择字节中的 STANDY_RST 位置 0，将启用待机模式复位。这时执行了进入待机模式的过程后，将执行系统复位而不是进入待机模式。通过将用户选择字节中的 STOP_RST 位置 0，将启用停机模式复位。这时执行了进入停机模式的过程后，将执行系统复位而不是进入停机模式。

图 3-1 系统复位结构



3.2.3 后备区域复位

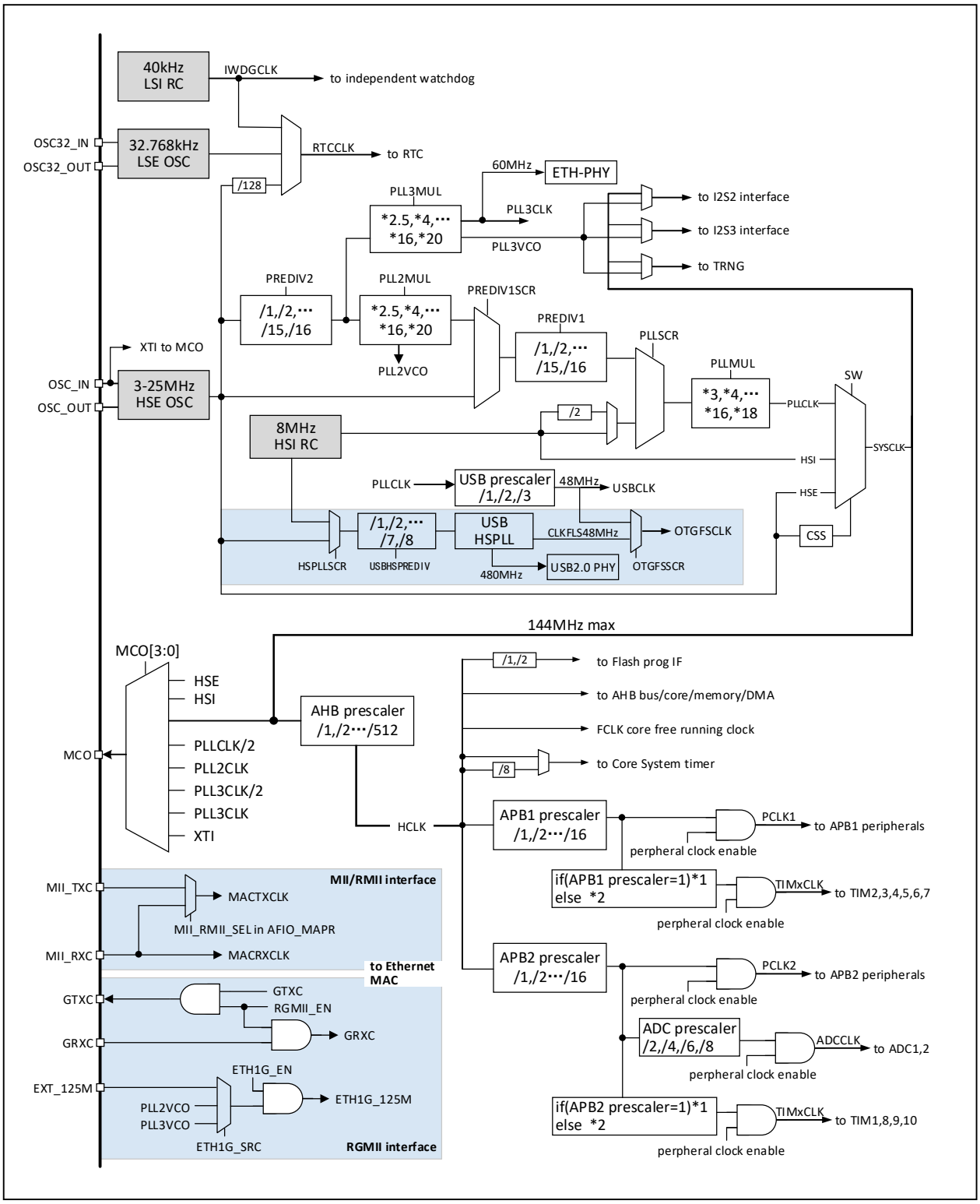
后备区域复位发生时，只会复位后备区域寄存器，包括后备寄存器、RCC_BDCTLR 寄存器（RTC 使能和 LSE 振荡器）。其产生条件包括：

- 在 V_{DD} 和 V_{BAT} 都掉电的前提下，由 V_{DD} 或 V_{BAT} 上电引起
- RCC_BDCTLR 寄存器的 BDRST 位置 1
- RCC_APB1PRSTR 寄存器的 BKPRST 位置 1

3.3 时钟

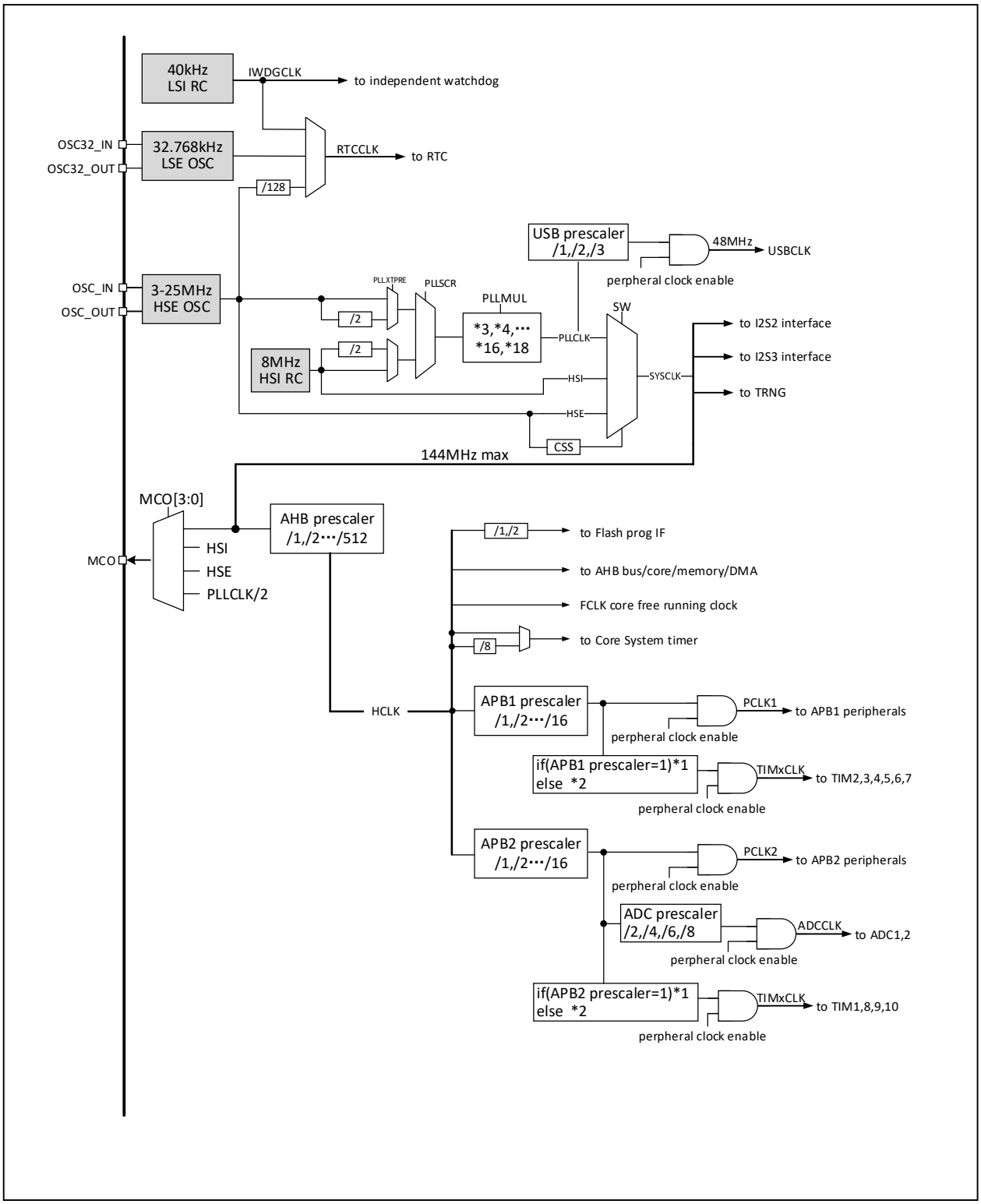
3.3.1 系统时钟结构

图 3-2 CH32V305/307 和 CH32F205/207 时钟树框图



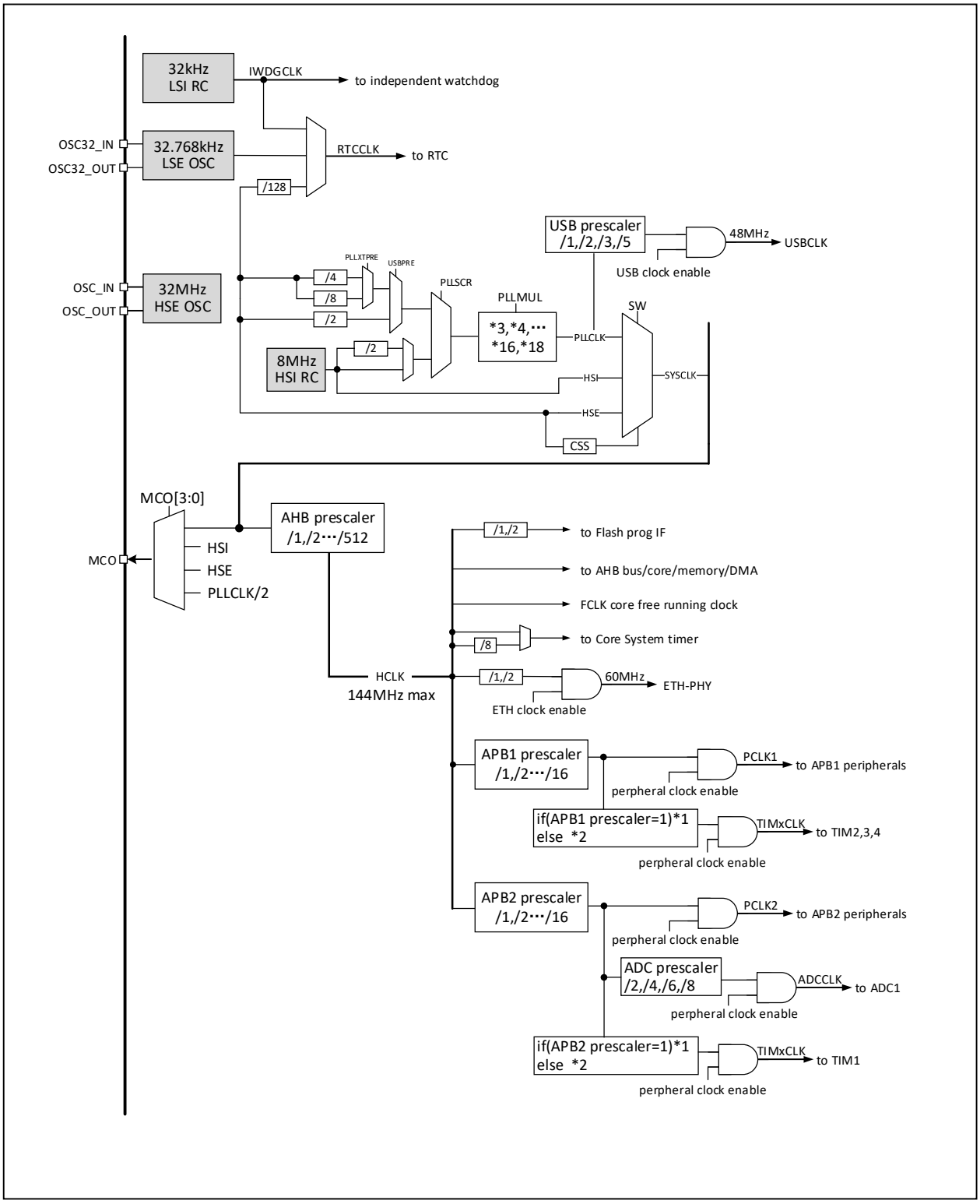
注：本时钟树适用于 CH32F20x_D8C 和 CH32V30x_D8C。当使用 USB 功能时，CPU 的频率必须是 48MHz、96MHz 或 144MHz。使用 USBHD 高速功能时，USBHSPLL 的时钟源只能为 HSE。当系统从睡眠状态唤醒时系统会自动切换为 HSI 做主频。

图 3-3 CH32FV203/V303 时钟树框图



注：本时钟树适用于 CH32F20x_D6、CH32F20x_D8、CH32V20x_D6 和 CH32V30x_D8。

图 3-4 CH32FV208 时钟树结构



注：本时钟树适用于 CH32F20x_D8W、CH32V20x_D8 和 CH32V20x_D8W。若同时使用 USB 和 ETH 功能，需将 USBPRE[1:0]置为 11b。

3.3.2 高速时钟（HSI/HSE）

HSI 是系统内部 8MHz 的 RC 振荡器产生的高速时钟信号。HSI RC 振荡器能够在不需要任何外部器件的条件下提供系统时钟。它的启动时间很短但时钟频率精度较差。HSI 通过设置 RCC_CTLR 寄存器

中的 HSION 位被启动和关闭，HSIRDY 位指示 HSI RC 振荡器是否稳定。系统默认 HSION 和 HSIRDY 置 1（建议不要关闭）。如果设置了 RCC_INTR 寄存器的 HSIRDYIE 位，将产生相应中断。

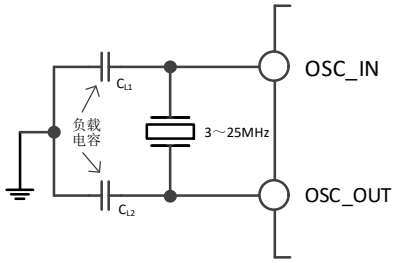
- 出厂校准：制造工艺的差异会导致每个芯片的 RC 振荡频率不同，所以在芯片出厂前，会为每颗芯片进行 HSI 校准。系统复位后，工厂校准值被装载到 RCC_CTLR 寄存器的 HSICAL[7:0]中。
- 用户调整：基于不同的电压或环境温度，应用程序可以通过 RCC_CTLR 寄存器里的 HSITRIM[4:0]位来调整 HSI 频率。

  注：如果 HSE 晶体振荡器失效，HSI 时钟会被作为备用时钟源（时钟安全系统）。

  HSE 是外部的高速时钟信号，包括外部晶体/陶瓷谐振器产生或者外部高速时钟送入。

- 外部晶体/陶瓷谐振器（HSE 晶体）：外接 3-25MHz 外部振荡器为系统提供更为精确的时钟源。进一步信息可参考数据手册的电气特性部分。HSE 晶体可以通过设置 RCC_CTLR 寄存器中的 HSEON 位被启动和关闭，HSERDY 位指示 HSE 晶体振荡是否稳定，硬件在 HSERDY 位置 1 后才将时钟送入系统。如果设置了 RCC_INTR 寄存器的 HSERDYIE 位，将产生相应中断。

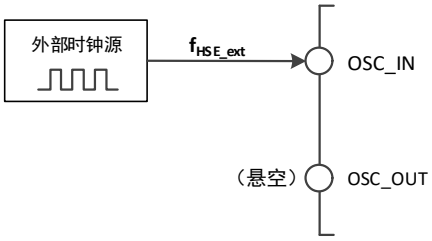
图 3-5 高速外部晶体电路



注：负载电容需要尽可能地靠近振荡器引脚，并根据晶体厂家参数选择容值。

- 外部高速时钟源（HSE 旁路）：此模式从外部直接送入时钟源到 OSC_IN 引脚，OSC_OUT 引脚悬空。最高支持 25MHz 频率。应用程序需在 HSEON 位为 0 情况下，置位 HSEBYP 位，打开 HSE 旁路功能，然后再置位 HSEON 位。

图 3-6 高速时钟源电路



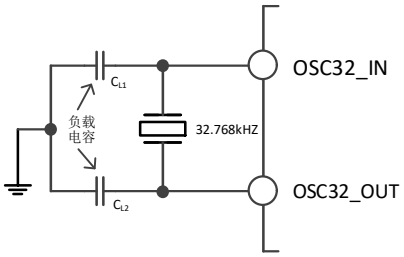
3.3.3 低速时钟（LSI/LSE）

LSI 是系统内部约 40kHz 的 RC 振荡器产生的低速时钟信号。它可以在停机和待机模式下保持运行，为 RTC 时钟、独立看门狗和唤醒单元提供时钟基准。进一步信息可参考数据手册的电气特性部分。LSI 可以通过设置 RCC_RSTSCKR 寄存器中的 LSION 位被启动和关闭，然后通过查询 LSIRDY 位检测 LSI RC 振荡是否稳定，硬件在 LSIRDY 位置 1 后才将时钟送入。如果设置了 RCC_INTR 寄存器的 LSIRDYIE 位，将产生相应中断。

LSE 是外部的低速时钟信号，包括外部晶体/陶瓷谐振器产生或者外部低速时钟送入。它为 RTC 时钟或者其他定时功能提供一个低功耗且精确的时钟源。

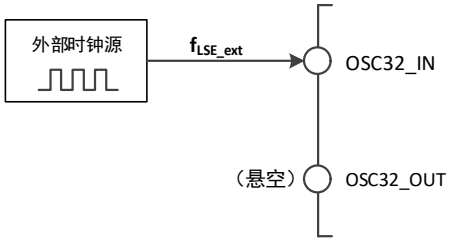
- 外部晶体/陶瓷谐振器（LSE 晶体）：外接 32.768kHz 的外部低速振荡器。LSE 通过设置 RCC_BDCTLR 寄存器中的 LSEON 位被启动和关闭，LSERDY 位指示 LSE 晶体振荡是否稳定，硬件在 LSERDY 位置 1 后才将时钟送入系统。如果设置了 RCC_INTR 寄存器的 LSERDYIE 位，将产生相应中断。

图 3-7 低速外部晶体电路



- 外部低速时钟源（LSE 旁路）：此模式从外部直接送入时钟源到 OSC32_IN 引脚，OSC32_OUT 引脚悬空。应用程序需在 LSEON 位为 0 情况下，置位 LSEBYP 位，打开 LSE 旁路功能，然后再置位 LSEON 位。

图 3-8 低速时钟源电路



3.3.4 PLL 时钟

通过配置 RCC_CFGR0 寄存器和扩展寄存器 EXTEND_CTR，内部 PLL 时钟可以选择 3 种时钟来源和倍频系数，这些设置必须在每个 PLL 被开启前完成，一旦 PLL 被启动，这些参数就不能被改动。设置 RCC_CTLR 寄存器中的 PLLON 位被启动和关闭，PLLRDY 位指示 PLL 时钟是否稳定，硬件在 PLLRDY 位置 1 后才将时钟送入系统。设置 RCC_CTLR 寄存器中的 PLLON2 位被启动和关闭，PLLRDY2 位指示 PLL2 时钟是否稳定，硬件在 PLLRDY2 位置 1 后才将时钟送入系统。设置 RCC_CTLR 寄存器中的 PLLON3 位被启动和关闭，PLLRDY3 位指示 PLL3 时钟是否稳定，硬件在 PLLRDY3 位置 1 后才将时钟送入系统。如果设置了 RCC_INTR 寄存器的 PLLRDYIE 位、PLL2RDYIE 位或 PLL3RDYIE 位，将产生相应中断。

PLL 时钟来源：

- HSI 时钟送入
- HSI 经过 2 分频送入
- HSE 时钟或通过一个可配置的分频器的 PLL2 时钟

PLL2 和 PLL3 由 HSE 通过一个可配置的分频器（PREDIV2）2 提供时钟

3.3.5 总线/外设时钟

3.3.5.1 系统时钟（SYSCLK）

通过配置 RCC_CFGR0 寄存器 SW[1:0]位配置系统时钟来源，SWS[1:0]指示当前的系统时钟源。

- HSI 作为系统时钟
- HSE 作为系统时钟
- PLL 时钟作为系统时钟

控制器复位后，默认 HSI 时钟被选为系统时钟源。时钟源之间的切换必须在目标时钟源准备就绪后才会发生。

3.3.5.2 AHB/APB1/APB2 总线外设时钟（HCLK/PCLK1/PCLK2）

通过配置 RCC_CFGR0 寄存器的 HPRE[3:0]、PPRE1[2:0]、PPRE2[2:0]位，可以分别配置 AHB、APB1、

APB2 总线的时钟。这些总线时钟决定了挂载在其下面的外设接口访问时钟基准。应用程序可以调整不同的数值，来降低部分外设工作时的功耗。

通过 RCC_AHBRSTR、RCC_APB1PRSTR、RCC_APB2PRSTR 寄存器中各个位可以复位不同的外设模块，将其恢复到初始状态。

通过 RCC_AHBPCENR、RCC_APB1PCENR、RCC_APB2PCENR 寄存器中各个位可以单独开启或关闭不同外设模块通讯时钟接口。使用某个外设时，首先需要开启其时钟使能位，才能访问其寄存器。

3.3.5.3 RTC 时钟（RTCCLK）

通过设置 RCC_BDCTLR 寄存器的 RTCSEL[1:0]位，RTCCLK 时钟源可以由 HSE/128、LSE 或 LSI 时钟提供。修改此位前要保证电源控制寄存器(PWR_CR)中的 DBP 位置 1，只有后备区域复位，才能复位此位。

- LSE 作为 RTC 时钟：由于 LSE 处于后备域由 V_{BAT} 供电，只要 V_{BAT} 维持供电，尽管 V_{DD} 供电被切断，RTC 仍继续工作。
- LSI 作为 RTC 时钟：如果 V_{DD} 供电被切断，RTC 自动唤醒不能保证。
- HSE/128 作为 RTC 时钟：如果 VDD 供电被切断或内部电压调压器被关闭(1.8V 域的供电被切断)，则 RTC 状态不确定。

3.3.5.4 独立看门狗时钟

如果独立看门狗已经由硬件配置设置或软件启动，LSI 振荡器将被强制打开，并且不能被关闭。在 LSI 振荡器稳定后，时钟供应给 IWDG。

3.3.5.5 时钟输出（MCO）

微控制器允许输出时钟信号到 MCO 引脚。在相应的 GPIO 端口寄存器配置复用推挽输出模式，通过配置 RCC_CFGR0 寄存器 MCO[3:0]位，可以选择以下 8 个时钟信号作为 MCO 时钟输出：

- 系统时钟(SYSCLK)输出
- HSI 时钟输出
- HSE 时钟输出
- PLL 时钟经过 2 分频输出
- PLL2 时钟输出
- PLL3 时钟输出
- PLL3 时钟经过 2 分频输出
- XT1 外部 3-25MHz 振荡器（用于以太网）

3.3.5.6 USB 时钟

USBD 48MHz 时钟源来自通过一个可配置的分频器的 PLL 时钟，此时 PLL 支持三种时钟配置，包括 48MHz、96MHz 和 144MHz，通过配置寄存器 RCC_CFGR0 的 USBPRE[1:0]位输出 48MHz 时钟到 USBD。

USBOTG 48MHz 时钟源来自通过一个可配置的分频器的 PLL 时钟或者 USBHSPLL 时钟，可通过配置寄存器 RCC_CFGR2 的 USB_CLK_SRC 位来选择。若时钟源选择通过一个可配置的分频器的 PLL 时钟作为时钟源时，则配置步骤可参考 USBD。若时钟源选择 USBHSPLL 时钟作为时钟源时，通过配置寄存器 RCC_CFGR2 的 USBHS_CKREFSEL[1:0]位,选择 USBHS PLL 参考时钟频率(参考时钟频率必须和 USBHS PLL 输入时钟保持一致）。

USBHD 时钟源来自 USBHSPLL 时钟，通过配置寄存器 RCC_CFGR2 的 USBHS_CKREFSEL[1:0]位，选择 USBHS PLL 参考时钟频率（参考时钟频率必须和 USBHS PLL 输入时钟保持一致），通过配置寄存器 RCC_CFGR2 的 USBHS_PLLALIVE 位，使能 USB PHY 内部 PLL。

3.3.5.7 ETH 时钟

ETH 时钟配置参考 27.4.5.1 章节。

3.3.5.8 I2S 和 RNG 时钟

I2S 和 RNG 的时钟源来自 PLL3VCO 或系统时钟（SYSCLK），I2S2、I2S3 和 TRNG 可分别通过配置寄存器 RCC_CFGR2 的 I2S2SRC 位、I2S3SRC 位和 RNG_SRC 位选择时钟源。

3.3.6 时钟安全系统

时钟安全系统是控制器的一种运行保护机制，它可以在 HSE 时钟发送故障的情况下，切换到 HSI 时钟下，并产生中断通知，允许应用程序软件完成营救操作。

通过设置 RCC_CTLR 寄存器的 CSSON 位置 1，激活时钟安全系统。此时，时钟监测器将在 HSE 振荡器启动（HSERDY=1）延迟后被使能，并在 HSE 时钟关闭后关闭。一旦系统运行过程中 HSE 时钟发生故障，HSE 振荡器将被关闭，时钟失效事件将被送到高级定时器(TIM1 和 TIM8)的刹车输入端，并产生时钟安全中断，CSSF 位置 1，并且应用程序进入 NMI 不可屏蔽中断，通过置位 CSSC 位，可以清除 CSSF 位标志，可撤销 NMI 中断挂起位。

如果当前 HSE 作为系统时钟，或者当前 HSE 作为 PLL 输入时钟，PLL 作为系统时钟，时钟安全系统将在 HSE 故障时自动将系统时钟切换到 HSI 振荡器，并关闭 HSE 振荡器和 PLL。

3.4 扩展配置

系统提供了 EXTEND 扩展配置单元（EXTEND_CTR 寄存器）。该单元使用 AHB 时钟，只在系统复位执行复位动作。主要包括以下几个扩展控制位功能：

1) 调节内置电压：LDOTRIM 和 ULLDOTRIM 字段选择默认值，在调节性能和功耗时可以修改其值。
2) PLL 时钟选择：HSIPRE 字段配合原有的时钟配置寄存器，提供了 HSI 时钟进行分频或不分频作为 PLL 的输入时钟的选择。
3) Lock-up 功能监控：LKUPEN 字段启用，将打开系统的 Lock-up 情况监控，一旦发生 Lock-up 情况，系统将进行软件复位，并将 LKUPRESET 字段置 1，读取后可以写 1 清除此标志。
4) USBD 模块的内置电阻及传输速度控制：USB 全速设备控制器（USBD）通过 USBDPU 字段选择是否使用内置的上拉电阻（1.5KΩ），不启用时需要在 USB 的引脚接上拉电阻（低速模式接 UD-引脚，全速模式接 UD+引脚）。USBDLS 字段配置当前 USB 设备速度模式。
5) ETH 模块 10M 以太网和 1000M 以太网 RGMII 接口是否启用控制位：通过 ETH_10M 启用 10M 以太网功能，通过 ETH_RGMII 启用 1000M 以太网 RGMII 接口。
6) 低功耗模式下 HSE 振荡控制位：通过该位可控制在低功耗模式下，HSE 是否振荡。

注：不同型号扩展寄存器位定义不同，具体细节参考配置扩展控制寄存器（EXTEND_CTR）。

3.5 寄存器描述

表 3-1 RCC 相关寄存器列表

名称	访问地址	描述	复位值
R32_RCC_CTLR	0x40021000	时钟控制寄存器	0x0000xx83
R32_RCC_CFGR0	0x40021004	时钟配置寄存器 0	0x00000000
R32_RCC_INTR	0x40021008	时钟中断寄存器	0x00000000
R32_RCC_APB2PRSTR	0x4002100C	APB2 外设复位寄存器	0x00000000
R32_RCC_APB1PRSTR	0x40021010	APB1 外设复位寄存器	0x00000000

R32_RCC_AHBPCENR	0x40021014	AHB 外设时钟使能寄存器	0x00000014
R32_RCC_APB2PCENR	0x40021018	APB2 外设时钟使能寄存器	0x00000000
R32_RCC_APB1PCENR	0x4002101C	APB1 外设时钟使能寄存器	0x00000000
R32_RCC_BDCTLR	0x40021020	后备域控制寄存器	0x00000000
R32_RCC_RSTSCKR	0x40021024	控制/状态寄存器	0x0C000000
R32_RCC_AHBRSTR	0x40021028	AHB 外设复位寄存器	0x00000000
R32_RCC_CFGR2	0x4002102C	AHB 外设复位寄存器	0x00000000

表 3-2 EXTEND 相关寄存器列表

名称	访问地址	描述	复位值
R32_EXTEND_CTR	0x40023800	配置扩展控制寄存器	0x00000A00

3.5.1 时钟控制寄存器（RCC_CTLR）

偏移地址：0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		PLL3 RDY	PLL3 ON	PLL2 RDY	PLL2 ON	PLL RDY	PLL ON	Reserved				CSSON	HSE BYP	HSE RDY	HSEON

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Reserved	HSI RDY	HSION

位	名称	访问	描述	复位值
[31:30]	Reserved	RO	保留。	0
29	PLL3RDY	RO	PLL3 时钟就绪锁定标志位（由硬件置位）： 1：PLL3 时钟锁定； 0：PLL3 时钟未锁定。 *注：适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
28	PLL3ON	RW	PLL3 时钟使能控制位： 1：使能 PLL3 时钟； 0：关闭 PLL3 时钟。 *注：进入停止或待机低功耗模式后，此位由硬件清 0。适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
27	PLL2RDY	RO	PLL2 时钟就绪锁定标志位（由硬件置位）： 1：PLL 时钟锁定； 0：PLL 时钟未锁定。 *注：适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
26	PLL2ON	RW	PLL2 时钟使能控制位： 1：使能 PLL 时钟； 0：关闭 PLL 时钟。 *注：进入停止或待机低功耗模式后，此位由硬件清 0。适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
25	PLLRDY	RO	PLL 时钟就绪锁定标志位（由硬件置位）： 1：PLL 时钟锁定； 0：PLL 时钟未锁定。	0
24	PLLON	RW	PLL 时钟使能控制位：	0

			1：使能 PLL 时钟； 0：关闭 PLL 时钟。 *注：进入停止或待机低功耗模式后，此位由硬件清 0。*	
[23:20]	Reserved	RO	保留。	0
19	CSSON	RW	时钟安全系统使能控制位： 1：使能时钟安全系统。当 HSE 准备好（HSERDY 置 1），硬件开启对 HSE 的时钟监测功能，发现 HSE 异常触发 CSSF 标志及 NMI 中断；当 HSE 没有准备好，硬件关闭对 HSE 的时钟监测功能。 0：关闭时钟安全系统。	0
18	HSEBYP	RW	外部高速晶体旁路控制位： 1：旁路外部高速晶体/陶瓷谐振器（使用外部时钟源）； 0：不旁路高速外部晶体/陶瓷谐振器。 *注：此位需在 HSEON 为 0 下写入。*	0
17	HSERDY	RO	外部高速晶体振荡稳定就绪标志位（由硬件置位）： 1：外部高速晶体振荡稳定； 0：外部高速晶体振荡没有稳定。 *注：在 HSEON 位清 0 后，该位需要 6 个 HSE 周期清 0。*	0
16	HSEON	RW	外部高速晶体振荡使能控制位： 1：使能 HSE 振荡器； 0：关闭 HSE 振荡器。 *注：进入停止或待机低功耗模式后，此位由硬件清 0。*	0
[15:8]	HSICAL	RO	内部高速时钟校准值，在系统启动时被自动初始化。	xxh
[7:3]	HSITRIM	RW	内部高速时钟调整值： 用户可以输入一个调整值叠加到 HSICAL[7:0]数值上，根据电压和温度的变化调整内部 HSI RC 振荡器的频率。 默认值为 16，可以把 HSI 调整到 8MHz±1%；每步 HSICAL 的变化调整约 40kHz。	10000
2	Reserved	RO	保留。	0
1	HSIRDY	RO	内部高速时钟（8MHz）稳定就绪标志位（由硬件置位）： 1：内部高速时钟（8MHz）稳定； 0：内部高速时钟（8MHz）没有稳定。 *注：在 HSION 位清 0 后，该位需要 6 个 HSI 周期清 0。*	1
0	HSION	RW	内部高速时钟（8MHz）使能控制位： 1：使能 HSI 振荡器； 0：关闭 HSI 振荡器。 *注：当从待机和停止模式返回或用作系统时钟的外部振荡器 HSE 发生故障时，该位由硬件置 1 来启动内部 8MHz 的 RC 振荡器。*	1

3.5.2 时钟配置寄存器 0（RCC_CFGR0）

偏移地址：0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC_CLK	Reserved		ETHP	MCO[3:0]				USB		PLLMUL[3:0]				PLL	PLL

_ADJ		RE		PRE		XTPRE	SRC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPRE[1:0]		PPRE2[2:0]			PPRE1[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	

位	名称	访问	描述	复位值
31	ADC_CLK_ADJ	RW	ADC 时钟占空比调整： 1：ADC 时钟低电平时间更长； 0：ADC 时钟占空比为 50%。	0
[30:29]	Reserved	RO	保留。	0
28	ETHPRE	RW	以太网时钟来源预分频控制： 0：不分频； 1：2 分频； *注：适用于 CH32V20x_D8W、CH32V20x_D8、CH32F20x_D8W。*	
[27:24]	MCO	RW	微控制器 MCO 引脚时钟输出控制： 00xx：没有时钟输出； 0100：系统时钟(SYSCLK)输出； 0101：内部 8MHz 的 RC 振荡器时钟(HSI)输出； 0110：外部振荡器时钟(HSE)输出； 0111：PLL 时钟 2 分频后输出； 1000：PLL2 时钟输出； 1001：PLL3 时钟 2 分频后输出； 1010：XT1 外部震荡器时钟输出； 1011：PLL3 时钟输出。 *注：在启动或切换 MCO 时钟时，可能有几个周期的时钟丢失。其中 1000 —— 1011 适用于适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
[23:22]	USBPRE	RW	USBFS/USBOTG 时钟分频配置： 00：1 分频（适用于 PLLCLK=48MHz）； 01：2 分频（适用于 PLLCLK=96MHz）； 10：3 分频（适用于 PLLCLK=144MHz）； 11：不使用。	0
[21:18]	PLLMUL	RW	PLL 时钟倍频系数（在 PLL 关闭才可写入）： 对于 CH32V20x_D8/CH32V20x_D8W/CH32F20x_D8WCH32V30x_D8/CH32F20x_D8/CH32V20x_D8/CH32V20x_D8W/CH32F20x_D8W- 0000：PLL 2 倍频输出；0001：PLL 3 倍频输出； 0010：PLL 4 倍频输出；0011：PLL 5 倍频输出； 0100：PLL 6 倍频输出；0101：PLL 7 倍频输出； 0110：PLL 8 倍频输出；0111：PLL 9 倍频输出； 1000：PLL 10 倍频输出；1001：PLL 11 倍频输出； 1010：PLL 12 倍频输出；1011：PLL 13 倍频输出； 1100：PLL 14 倍频输出；1101：PLL 15 倍频输出； 1110：PLL 16 倍频输出；1111：PLL 18 倍频输出。	0

			对于 CH32F20x_D8C/CH32V30x_D8C- 0000：PLL 18 倍频输出； 0001：PLL 3 倍频输出； 0010：PLL 4 倍频输出； 0011：PLL 5 倍频输出； 0100：PLL 6 倍频输出； 0101：PLL 7 倍频输出； 0110：PLL 8 倍频输出； 0111：PLL 9 倍频输出； 1000：PLL 10 倍频输出； 1001：PLL 11 倍频输出； 1010：PLL 12 倍频输出； 1011：PLL 13 倍频输出； 1100：PLL 14 倍频输出； 1101：PLL 6.5 倍频输出； 1110：PLL 15 倍频输出； 1111：PLL 16 倍频输出。	
17	PLLXTPRE	RW	HSE 分频送入 PLL 控制（在 PLL 关闭才可写入）： 1：HSE 2 分频送入 PLL； 0：HSE 不分频送入 PLL。	0
16	PLLSRC	RW	PLL 的输入时钟源（在 PLL 关闭才可写入）： 1：HSE 不分频或 2 分频送入 PLL； 0：HSI 不分频或 2 分频送入 PLL。	0
[15:14]	ADCPRE	RW	ADC 时钟来源预分频控制： 00：PCLK2 2 分频后作为 ADC 时钟； 01：PCLK2 4 分频后作为 ADC 时钟； 10：PCLK2 6 分频后作为 ADC 时钟； 11：PCLK2 8 分频后作为 ADC 时钟。 *注：ADC 时钟最高不要超过 14MHz。*	0
[13:11]	PPRE2	RW	APB2 时钟来源预分频控制： 0xx：HCLK 不分频； 100：HCLK 2 分频； 101：HCLK 4 分频； 110：HCLK 8 分频； 111：HCLK 16 分频	0
[10:8]	PPRE1	RW	APB1 时钟来源预分频控制： 0xx：HCLK 不分频； 100：HCLK 2 分频； 101：HCLK 4 分频； 110：HCLK 8 分频； 111：HCLK 16 分频	0
[7:4]	HPRE	RW	AHB 时钟来源预分频控制： 0xxx：SYSCLK 不分频； 1000：SYSCLK 2 分频； 1001：SYSCLK 4 分频； 1010：SYSCLK 8 分频； 1011：SYSCLK 16 分频； 1100：SYSCLK 64 分频； 1101：SYSCLK 128 分频； 1110：SYSCLK 256 分频； 1111：SYSCLK 512 分频。 *注：当 AHB 时钟来源的预分频系数大于 1 时，必须开启预取缓冲器。*	0
[3:2]	SWS	RO	系统时钟（SYSCLK）状态（硬件置位）：	0

			00：系统时钟源是 HSI； 01：系统时钟源是 HSE； 10：系统时钟源是 PLL； 11：不可用。	
[1:0]	SW	RW	选择系统时钟来源： 00：HSI 作为系统时钟； 01：HSE 作为系统时钟； 10：PLL 输出作为系统时钟； 11：不可用。 *注：在使能了时钟安全系统下（CSSON=1），当从待机和停止模式返回或用作系统时钟的外部振荡器 HSE 发生故障时，由硬件强制选择 HSI 作为系统时钟。*	0

3.5.3 时钟中断寄存器（RCC_INTR）

偏移地址：0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CSSC	PLL3 RDYC	PLL2 RDYC	PLL RDYC	HSE RDYC	HSI RDYC	LSE RDYC	LSI RDYC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLL3 RDYIE	PLL2 RDYIE	PLL RDYIE	HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE	CSSF	PLL3 RDYF	PLL2 RDYF	PLL RDYF	HSE RDYF	HSI RDYF	LSE RDYF	LSI RDYF

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
23	CSSC	WO	清除时钟安全系统中断标志位（CSSF）： 1：清除 CSSF 中断标志； 0：无动作。	0
22	PLL3RDYC	WO	清除 PLL3 就绪中断标志位： 1：清除 PLL3RDYF 中断标志； 0：无动作。 *注：适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
21	PLL2RDYC	WO	清除 PLL2 就绪中断标志位： 1：清除 PLL2RDYF 中断标志； 0：无动作。 *注：适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
20	PLLRDYC	WO	清除 PLL 就绪中断标志位： 1：清除 PLLRDYF 中断标志； 0：无动作。	0
19	HSERDYC	WO	清除 HSE 振荡器就绪中断标志位： 1：清除 HSERDYF 中断标志； 0：无动作。	0
18	HSIRDYC	WO	清除 HSI 振荡器就绪中断标志位： 1：清除 HSIRDYF 中断标志； 0：无动作。	0
17	LSERDYC	WO	清除 LSE 振荡器就绪中断标志位：	0

			1：清除 LSERDYF 中断标志； 0：无动作。	
16	LSIRDYC	WO	清除 LSI 振荡器就绪中断标志位： 1：清除 LSIRDYF 中断标志； 0：无动作。	0
14	PLL3RDYIE	RW	PLL3 就绪中断使能位： 1：使能 PLL3 就绪中断； 0：关闭 PLL3 就绪中断。 *注：适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
13	PLL2RDYIE	RW	PLL2 就绪中断使能位： 1：使能 PLL2 就绪中断； 0：关闭 PLL2 就绪中断。 *注：适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
12	PLLRDYIE	RW	PLL 就绪中断使能位： 1：使能 PLL 就绪中断； 0：关闭 PLL 就绪中断。	0
11	HSERDYIE	RW	HSE 就绪中断使能位： 1：使能 HSE 就绪中断； 0：关闭 HSE 就绪中断。	0
10	HSIRDYIE	RW	HSI 就绪中断使能位： 1：使能 HSI 就绪中断； 0：关闭 HSI 就绪中断。	0
9	LSERDYIE	RW	LSE 就绪中断使能位： 1：使能 LSE 就绪中断； 0：关闭 LSE 就绪中断。	0
8	LSIRDYIE	RW	LSI 就绪中断使能位： 1：使能 LSI 就绪中断； 0：关闭 LSI 就绪中断。	0
7	CSSF	RO	时钟安全系统中断标志位： 1：HSE 时钟失效，产生了时钟安全中断 CSSI； 0：无时钟安全系统中断。 硬件置位，软件写 CSSC 位 1 清除。	0
6	PLL3RDYF	RO	PLL3 时钟就绪锁定中断标志： 1：PLL3 时钟锁定产生中断； 0：无 PLL3 时钟锁定中断。 硬件置位，软件写 PLL3RDYC 位 1 清除。 *注：适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
5	PLL2RDYF	RO	PLL2 时钟就绪锁定中断标志： 1：PLL2 时钟锁定产生中断； 0：无 PLL2 时钟锁定中断。 硬件置位，软件写 PLL2RDYC 位 1 清除。 *注：适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
4	PLLRDYF	RO	PLL 时钟就绪锁定中断标志： 1：PLL 时钟锁定产生中断； 0：无 PLL 时钟锁定中断。 硬件置位，软件写 PLLRDYC 位 1 清除。	0

3	HSERDYF	RO	HSE 时钟就绪中断标志： 1：HSE 时钟就绪产生中断； 0：无 HSE 时钟就绪中断。 硬件置位，软件写 HSERDYC 位 1 清除。	0
2	HSIRDYF	RO	HSI 时钟就绪中断标志： 1：HSI 时钟就绪产生中断； 0：无 HSI 时钟就绪中断。 硬件置位，软件写 HSIRDYC 位 1 清除。	0
1	LSERDYF	RO	LSE 时钟就绪中断标志： 1：LSE 时钟就绪产生中断； 0：无 LSE 时钟就绪中断。 硬件置位，软件写 LSERDYC 位 1 清除。	0
0	LSIRDYF	RO	LSI 时钟就绪中断标志： 1：LSI 时钟就绪产生中断； 0：无 LSI 时钟就绪中断。 硬件置位，软件写 LSIRDYC 位 1 清除。	0

3.5.4 APB2 外设复位寄存器（RCC_APB2PRSTR）

偏移地址：0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TIM10 RST	TIM9 RST	Reserved		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART1 RST	TIM8 RST	SPI1 RST	TIM1 RST	ADC2 RST	ADC1 RST	Reserved		IOPE RST	IOPD RST	IOPC RST	IOPB RST	IOPA RST	Reserved	AFIO RST

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
20	TIM10RST	RW	TIM10 模块复位控制： 1：复位模块；　　0：无作用。	0
19	TIM9RST	RW	TIM9 模块复位控制： 1：复位模块；　　0：无作用。	0
[18:15]	Reserved	RO	保留。	0
14	USART1RST	RW	USART1 接口复位控制： 1：复位模块；　　0：无作用。	0
13	TIM8RST	RW	TIM8 模块复位控制： 1：复位模块；　　0：无作用。	0
12	SPI1RST	RW	SPI1 接口复位控制： 1：复位模块；　　0：无作用。	0
11	TIM1RST	RW	TIM1 模块复位控制： 1：复位模块；　　0：无作用。	0
10	ADC2RST	RW	ADC2 模块复位控制： 1：复位模块；　　0：无作用。	0
9	ADC1RST	RW	ADC1 模块复位控制： 1：复位模块；　　0：无作用。	0

[8:7]	Reserved	RO	保留。	0
6	IOPERST	RW	IO 的 PE 端口模块复位控制： 1：复位模块； 0：无作用。	0
5	IOPDRST	RW	IO 的 PD 端口模块复位控制： 1：复位模块； 0：无作用。	0
4	IOPCRST	RW	IO 的 PC 端口模块复位控制： 1：复位模块； 0：无作用。	0
3	IOPBRST	RW	IO 的 PB 端口模块复位控制： 1：复位模块； 0：无作用。	0
2	IOPARST	RW	IO 的 PA 端口模块复位控制： 1：复位模块； 0：无作用。	0
1	Reserved	RO	保留。	0
0	AFIORST	RW	IO 辅助功能模块复位控制： 1：复位模块； 0：无作用。	0

3.5.5 APB1 外设复位寄存器（RCC_APB1PRSTR）

偏移地址：0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DAC RST	PWR RST	BKP RST	CAN2 RST	CAN1 RST	Reserved	USBD RST	I2C2 RST	I2C1 RST	USART5 RST	USART4 RST	USART3 RST	USART2 RST	Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	Reserved		WWDG RST	Reserved		UART 8RST	UART 7RST	UART 6RST	TIM7 RST	TIM6 RST	TIM5 RST	TIM4 RST	TIM3 RST	TIM2 RST

位	名称	访问	描述	复位值
[31:30]	Reserved	RO	保留。	0
29	DACRST	RW	DAC 模块复位控制： 1：复位模块； 0：无作用。	0
28	PWRRST	RW	电源接口模块复位控制： 1：复位模块； 0：无作用。	0
27	BKPRST	RW	后备单元复位控制： 1：复位模块； 0：无作用。	0
26	CAN2RST	RW	CAN2 模块复位控制： 1：复位模块； 0：无作用。	0
25	CAN1RST	RW	CAN1 模块复位控制： 1：复位模块； 0：无作用。	0
24	Reserved	RO	保留。	0
23	USBDRST	RW	USBD 模块复位控制： 1：复位模块； 0：无作用。	0
22	I2C2RST	RW	I2C 2 接口复位控制： 1：复位模块； 0：无作用。	0
21	I2C1RST	RW	I2C 1 接口复位控制： 1：复位模块； 0：无作用。	0
20	USART5RST	RW	USART5 接口复位控制：	0

			1：复位模块； 0：无作用。	
19	USART4RST	RW	USART4 接口复位控制： 1：复位模块； 0：无作用。	0
18	USART3RST	RW	USART3 接口复位控制： 1：复位模块； 0：无作用。	0
17	USART2RST	RW	USART2 接口复位控制： 1：复位模块； 0：无作用。	0
16	Reserved	RO	保留。	0
15	SPI3RST	RW	SPI3 接口复位控制： 1：复位模块； 0：无作用。	0
14	SPI2RST	RW	SPI2 接口复位控制： 1：复位模块； 0：无作用。	0
[13:12]	Reserved	RO	保留。	0
11	WWDGRST	RW	窗口看门狗复位控制： 1：复位模块； 0：无作用。	0
[10:9]	Reserved	RO	保留。	0
8	UART8RST	RW	UART8 接口复位控制： 1：复位模块； 0：无作用。	0
7	UART7RST	RW	UART7 接口复位控制： 1：复位模块； 0：无作用。	0
6	UART6RST	RW	UART6 接口复位控制： 1：复位模块； 0：无作用。	0
5	TIM7RST	RW	定时器 7 模块复位控制： 1：复位模块； 0：无作用。	0
4	TIM6RST	RW	定时器 6 模块复位控制： 1：复位模块； 0：无作用。	0
3	TIM5RST	RW	定时器 5 模块复位控制： 1：复位模块； 0：无作用。	0
2	TIM4RST	RW	定时器 4 模块复位控制： 1：复位模块； 0：无作用。	0
1	TIM3RST	RW	定时器 3 模块复位控制： 1：复位模块； 0：无作用。	0
0	TIM2RST	RW	定时器 2 模块复位控制： 1：复位模块； 0：无作用。	0

3.5.6 AHB 外设时钟使能寄存器（RCC_AHBPCENR）

偏移地址：0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETHMACTXEN	ETHMACEN	DVPEN	OTGEN	USBHSEN	SDIOEN	RNGEN	FSMCEN	Reserved	CRCEN	Reserved	Reserved	Reserved	SRAMEN	DMA2EN	DMA1EN

位	名称	访问	描述	复位值

[31:17]	Reserved	RO	保留。	0
16	ETHMACRXEN	RW	以太网 MAC 接收时钟使能： 1：以太网 MAC 接收时钟开启； 0：以太网 MAC 接收时钟关闭。	0
15	ETHMACTXEN	RW	以太网 MAC 发送时钟使能： 1：以太网 MAC 发送时钟开启； 0：以太网 MAC 发送时钟关闭。	0
14	ETHMACEN	RW	以太网 MAC 时钟使能： 1：以太网 MAC 时钟开启； 0：以太网 MAC 时钟关闭。	0
13	DVPEN	RW	DVP 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
12	OTGEN	RW	USBOTG 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
11	USBHSEN	RW	USBHS 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
10	SDIOEN	RW	SDIO 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
9	RNGEN	RW	RNG 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
8	FSMCEN	RW	FSMCEN 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
7	Reserved	RO	保留。	0
6	CRCEN	RW	CRC 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
5	Reserved	RO	保留。	0
4	Reserved	RO	保留。	0
3	Reserved	RO	保留。	0
2	SRAMEN	RW	SRAM 接口模块时钟使能位： 1：睡眠模式时，SRAM 接口模块时钟开启； 0：睡眠模式时，SRAM 接口模块时钟关闭。	1
1	DMA2EN	RW	DMA2 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
0	DMA1EN	RW	DMA1 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0

注：当外设时钟没有启用时，软件不能读出外设寄存器数值，返回的数值始终为 0。

3.5.7 APB2 外设时钟使能寄存器（RCC_APB2PCENR）

偏移地址：0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TIM10 EN	TIM9 EN	Reserved		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART1 EN	TIM8 EN	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	Reserved		IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Reserved	AFIO EN

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
20	TIM10EN	RW	TIM10 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
19	TIM9EN	RW	TIM9 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
[18:15]	Reserved	RO	保留。	0
14	USART1EN	RW	USART1 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
13	TIM8EN	RW	TIM8 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
12	SPI1EN	RW	SPI1 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
11	TIM1EN	RW	TIM1 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
10	ADC2EN	RW	ADC2 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
9	ADC1EN	RW	ADC1 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
[8:7]	Reserved	RO	保留。	0
6	IOPEEN	RW	IO 的 PE 端口模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
5	IOPDEN	RW	IO 的 PD 端口模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
4	IOPCEN	RW	IO 的 PC 端口模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
3	IOPBEN	RW	IO 的 PB 端口模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
2	IOPAEN	RW	IO 的 PA 端口模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
1	Reserved	RO	保留。	0
0	AFIOEN	RW	IO 辅助功能模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0

注：当外设时钟没有启用时，软件不能读出外设寄存器数值，返回的数值始终为 0。

3.5.8 APB1 外设时钟使能寄存器（RCC_APB1PCENR）

偏移地址：0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DAC EN	PWR EN	BKP EN	CAN2 EN	CAN1 EN	Reserved	USBD EN	I2C2 EN	I2C1 EN	UART5 EN	UART4 EN	USART3 EN	USART2 EN	Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Reserved		WWDG EN	Reserved		UART8 EN	UART7 EN	UART6 EN	TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN

位	名称	访问	描述	复位值
[31:30]	Reserved	RO	保留。	0
29	DACEN	RW	DAC 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
28	PWREN	RW	电源接口模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
27	BKPEN	RW	后备单元时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
26	CAN2EN	RW	CAN2 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
25	CAN1EN	RW	CAN1 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
24	Reserved	RO	保留。	0
23	USBDEN	RW	USBD 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
22	I2C2EN	RW	I2C 2 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
21	I2C1EN	RW	I2C 1 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
20	UART5EN	RW	UART5 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
19	UART4EN	RW	UART4 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
18	UART3EN	RW	UART3 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
17	UART2EN	RW	UART2 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
16	Reserved	RO	保留。	0
15	SPI3EN	RW	SPI3 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
14	SPI2EN	RW	SPI2 接口时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
[13:12]	Reserved	RO	保留。	0
11	WWDGEN	RW	窗口看门狗时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
[10:9]	Reserved	RO	保留。	0
8	UART8EN	RW	UART8 使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
7	UART7EN	RW	UART7 使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
6	UART6EN	RW	UART6 使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
5	TIM7EN	RW	定时器 7 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
4	TIM6EN	RW	定时器 6 模块时钟使能位：	0

			1：模块时钟开启；　　0：模块时钟关闭。	
3	TIM5EN	RW	定时器 5 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
2	TIM4EN	RW	定时器 4 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
1	TIM3EN	RW	定时器 3 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0
0	TIM2EN	RW	定时器 2 模块时钟使能位： 1：模块时钟开启；　　0：模块时钟关闭。	0

注：当外设时钟没有启用时，软件不能读出外设寄存器数值，返回的数值始终为 0。

3.5.9 后备域控制寄存器（RCC_BDCTLR）

偏移地址：0x20

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Reserved					RTCSEL[1:0]		Reserved					LSE BYP	LSE RDY	LSEON

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
16	BDRST	RW	后备域软件复位控制： 1：复位整个后备域。 0：撤销复位。	0
15	RTCEN	RW	RTC 时钟使能控制： 1：使能 RTC 时钟； 0：关闭 RTC 时钟。 *注：RTCSEL!=0 的条件下才可以使能 RTC 时钟，否则硬件强制为 0。*	0
[14:10]	Reserved	RO	保留。	0
[9:8]	RTCSEL	RW	RTC 时钟源选择： 00：无时钟； 01：LSE 振荡器作为 RTC 时钟； 10：LSI 振荡器作为 RTC 时钟； 11：HSE 振荡器经 128 分频后作为 RTC 时钟。 *注：一旦 RTC 时钟源被选定(RTCEN=1)，直到下次后备域被复位，它不能再被改变。可通过设置 BDRST 位来恢复默认。*	0
[7:3]	Reserved	RO	保留。	0
2	LSEBYP	RW	外部低速晶体（LSE）旁路控制位： 1：旁路外部低速晶体/陶瓷谐振器（使用外部时钟源）； 0：不旁路低速外部晶体/陶瓷谐振器。 *注：此位需在 LSEON 为 0 下写入。*	0
1	LSERDY	RO	外部低速晶体振荡稳定就绪标志位（由硬件置位）： 1：外部低速晶体振荡稳定；	0

			0：外部低速晶体振荡没有稳定。 *注：在 LSEON 位清 0 后，该位需要 6 个 LSE 周期清 0。*	
0	LSEON	RW	外部低速晶体振荡使能控制位： 1：使能 LSE 振荡器； 0：关闭 LSE 振荡器。	0

注：后备域控制寄存器中(RCC_BDCTLR)的 LSEON、LSEBYP、RTCSEL 和 RTCEN 位处于后备域。因此，这些位在复位后处于写保护状态，只有在电源控制寄存器(PWR_CR)中的 DBP 位置 1 后，才能对这些位进行改动。这些位只能由后备域复位清除。任何内部或外部复位都不会影响这些位。

3.5.10 控制/状态寄存器（RCC_RSTSCKR）

偏移地址：0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	WWDG RSTF	IWDG RSTF	SFT RSTF	POR RSTF	PIN RSTF	Reser ved	RMVF	Reserved							

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

位	名称	访问	描述	复位值
31	LPWRRSTF	RO	低功耗复位标志： 1：发生低功耗复位； 0：无低功耗复位发生。 发生低功耗管理复位时由硬件置 1；软件写 RMVF 位清除。	0
30	WWDGRSTF	RO	窗口看门狗复位标志： 1：发生窗口看门狗复位； 0：无窗口看门狗复位发生。 发生窗口看门狗复位时由硬件置 1；软件写 RMVF 位清除。	0
29	IWDGRSTF	RO	独立看门狗复位标志： 1：发生独立看门狗复位； 0：无独立看门狗复位发生。 发生独立看门狗复位时由硬件置 1；软件写 RMVF 位清除。	0
28	SFTRSTF	RO	软件复位标志： 1：发生软件复位； 0：无软件复位发生。 发生软件复位时由硬件置 1；软件写 RMVF 位清除。	0
27	PORRSTF	RO	上电/掉电复位标志： 1：发生上电/掉电复位； 0：无上电/掉电复位发生。 发生上电/掉电复位时由硬件置 1；软件写 RMVF 位清除。	1
26	PINRSTF	RO	外部手动复位（NRST引脚）标志： 1：发生NRST引脚复位； 0：无NRST引脚复位发生。	1

			在NRST引脚复位发生时由硬件置1；软件写RMVF位清除。	
25	Reserved	RO	保留。	0
24	RMVF	WO	清除复位标志控制： 1：清除复位标志； 0：无作用。	0
[23:2]	Reserved	RO	保留。	0
1	LSIRDY	RO	内部低速时钟（LSI）稳定就绪标志位（由硬件置位）： 1：内部低速时钟（40kHz）稳定； 0：内部低速时钟（40kHz）没有稳定。 *注：在 LSION 位清 0 后，该位需要 3 个 LSI 周期清 0。*	0
0	LSION	RW	内部低速时钟（LSI）使能控制位： 1：使能 LSI（40kHz）振荡器； 0：关闭 LSI（40kHz）振荡器。	0

注：除复位标志只能由上电复位清除，其他由系统复位清除。

3.5.11 AHB 外设复位寄存器（RCC_AHBRSTR）

偏移地址：0x28

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

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

位	名称	访问	描述	复位值
[31:15]	Reserved	RO	保留。	0
14	ETHMACRST	RW	以太网 MAC 复位控制： 1：复位模块；　　0：无作用。	0
13	DVPRST	RO	DVP 复位控制： 1：复位模块；　　0：无作用。	0
12	USBHDRST	RW	USBHD 模块复位控制： 1：复位模块；　　0：无作用。	0
[11:0]	Reserved	RO	保留。	0

3.5.12 时钟控制寄存器（RCC_CFGR2）

偏移地址：0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
USB_CLK_SRC	USBHS_PLLALIVE	USBHS_CKREFSEL		USBHS_PLL_SRC	USBHS_PREDIV			Reserved	ETH1G_125M_EN	ETH1G_SRC		RNG_SRC	I2S3SRC	I2S2SRC	PREDIV1SRC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL3MUL[3:0]				PLL2MUL[3:0]				PREDIV2[3:0]				PREDIV1[3:0]			

位	名称	访问	描述	复位值

31	USB_CLK_SRC	RW	USB 48MHz 时钟源选择： 1：USB PHY 0：PLL CLK	0
30	USBHS_PLLALIVE	RW	USBHS PHY 内部 PLL 控制位： 1：USB PHY 内部 PLL 使能 0：USB PHY 内部 PLL 关闭	0
[29:28]	USBHS_CKREFSEL	RW	USBHS PLL 参考时钟频率选择（USBHS_PLL_SRC/USBHS_PREDIV）： 00：3MHz 01：4MHz 10：8MHz 11：5MHz	00
27	USBHS_PLL_SRC	RW	USBHS PLL 参考源选择： 1：HSI 0:HSE	0
[26:24]	USBHS_PREDIV	RW	USBHS PLL 参考源分频： 000：1 分频 001：2 分频 010：3 分频 011：4 分频 100：5 分频 101：6 分频 110：7 分频 111：8 分频	000
23	Reserved	RO	保留。	0
22	ETH1G_125M_EN	RW	千兆以太网 125M 时钟控制位： 1：开启 0:关闭	
[21:20]	ETH1G_SRC	RW	千兆以太网 125M 时钟选择： 00：PLL2 VCO 01：PLL3 VCO 1x:外部 PB1 引脚输入	
19	RNG_SRC	RW	RNG 时钟源选择： 1：PLL3 VCO 0:系统时钟	
18	I2S3SRC	RW	I2S3 时钟源： 1：PLL3 VCO； 0：系统时钟（SYSCLK）。	0
17	I2S2SRC	RW	I2S2 时钟源： 1：PLL3 VCO； 0：系统时钟（SYSCLK）。	0
16	PREDIV1SRC	RW	PREDIV1 时钟源： 1：PLL2； 0：HSE。	0
[15:12]	PLL3MUL[3:0]	RW	PLL3 倍频因子（在 PLL3 关闭才可写入）。 0000:PLL3 2.5 倍频输出 0001:PLL3 12.5 倍频输出； 0010：PLL3 4 倍频输出； 0011：PLL3 5 倍频输出； 0100：PLL3 6 倍频输出； 0101：PLL3 7 倍频输出； 0110：PLL3 8 倍频输出； 0111：PLL3 9 倍频输出； 1000：PLL3 10 倍频输出；1001：PLL3 11 倍频输出； 1010：PLL3 12 倍频输出；1011：PLL3 13 倍频输出； 1100：PLL3 14 倍频输出；1101：PLL3 15 倍频输出； 1110：PLL3 16 倍频输出；1111：PLL3 20 倍频输出	0000
[11:8]	PLL2MUL[3:0]	RW	PLL2 倍频因子（在 PLL2 关闭才可写入）。	0000

			0000:PLL2 2.5 倍频输出 0001:PLL2 12.5 倍频输出； 0010：PLL2 4 倍频输出； 0011：PLL2 5 倍频输出； 0100：PLL2 6 倍频输出； 0101：PLL2 7 倍频输出； 0110：PLL2 8 倍频输出； 0111：PLL2 9 倍频输出； 1000：PLL2 10 倍频输出；1001：PLL2 11 倍频输出； 1010：PLL2 12 倍频输出；1011：PLL2 13 倍频输出； 1100：PLL2 14 倍频输出；1101：PLL2 15 倍频输出； 1110：PLL2 16 倍频输出；1111：PLL2 20 倍频输出	
[7:4]	PREDIV2[3:0]	RW	PREDIV2 分频因子（在 PLL2 和 PLL3 关闭才可写入） 0000：PREDIV2 不对输入时钟分频； 0001：PREDIV2 对输入时钟 2 分频； 0010：PREDIV2 对输入时钟 3 分频； 0011：PREDIV2 对输入时钟 4 分频； 0100：PREDIV2 对输入时钟 5 分频； 0101：PREDIV2 对输入时钟 6 分频； 0110：PREDIV2 对输入时钟 7 分频； 0111：PREDIV2 对输入时钟 8 分频； 1000：PREDIV2 对输入时钟 9 分频； 1001：PREDIV2 对输入时钟 10 分频； 1010：PREDIV2 对输入时钟 11 分频； 1011：PREDIV2 对输入时钟 12 分频； 1100：PREDIV2 对输入时钟 13 分频； 1101：PREDIV2 对输入时钟 14 分频； 1110：PREDIV2 对输入时钟 15 分频； 1111：PREDIV2 对输入时钟 16 分频；	0000
[3:0]	PREDIV1[3:0]	RW	PREDIV1 分频因子（在 PLL 关闭才可写入） 0000：PREDIV1 不对输入时钟分频； 0001：PREDIV1 对输入时钟 2 分频； 0010：PREDIV1 对输入时钟 3 分频； 0011：PREDIV1 对输入时钟 4 分频； 0100：PREDIV1 对输入时钟 5 分频； 0101：PREDIV1 对输入时钟 6 分频； 0110：PREDIV1 对输入时钟 7 分频； 0111：PREDIV1 对输入时钟 8 分频； 1000：PREDIV1 对输入时钟 9 分频； 1001：PREDIV1 对输入时钟 10 分频； 1010：PREDIV1 对输入时钟 11 分频； 1011：PREDIV1 对输入时钟 12 分频； 1100：PREDIV1 对输入时钟 13 分频； 1101：PREDIV1 对输入时钟 14 分频； 1110：PREDIV1 对输入时钟 15 分频； 1111：PREDIV1 对输入时钟 16 分频； *注：bit0 和* RCC_CFGR0 的 bit17 相同，修改 RCC_CFGR0 的 bit17 会同时改变该寄存器的 bit0。	0000

注：适用于 CH32F20x_D8C、CH32V30x_D8C。

3.5.13 配置扩展控制寄存器（EXTEND_CTR）

偏移地址：0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				LDO TRIM		ULLDO TRIM[1:0]		LKUP RESET	LKUP EN	Reserved	HSI PRE	ETH_RGMII	ETH_10M	USBD PU	USBD LS

位	名称	访问	描述	复位值
[31:13]	Reserved	RO	保留。	0
[12]	HSE_KEEP_LP	RW	低功耗模式下 HSE 振荡控制位： 1：低功耗模式下，HSE 保持振荡； 0：低功耗模式下，HSE 不振荡。 *注：适用于 CH32V20x_D8、CH32V20x_D8W、CH32F20x_D8W。*	0
[11:10]	LDOTRIM[1:0]	RW	调整数字内核电压值，LDO 电压值。	10b
[9:8]	ULLDOTRIM[1:0]	RW	调整低功耗模式下，ULLDO 电压值	10b
7	LKUPRESET	RW1	LOCKUP 复位标志： 1：发生 LOCKUP 导致系统复位，写 1 清除； 0：正常。	0
6	LKUPEN	RW	LOCKUP 监测功能： 1：启用，系统发生 lock-up 时执行复位并将 LOCKUP_RESET 置位； 0：不启用。	0
5	Reserved	RO	保留。	0
4	HSIPRE	RW	HSI 时钟是否分频：（只能在 PLL 关闭下写入） 1：HSI 时钟作为 PLL 输入时钟； 0：HSI 时钟经 2 分频作为 PLL 输入时钟。	0
3	ETH_RGMII	RW	1000M 以太网 RGMII 接口是否启用： 1：启用 1000M 以太网 RGMII 接口； 0：不启用。 *注：适用于 CH32F20x_D8C、CH32V30x_D8C。*	0
2	ETH_10M	RW	10M 以太网是否启用： 1：启用 10M 以太网功能； 0：不启用。 *注：适用于 CH32F20x_D8C、CH32V30x_D8C、CH32V20x_D8、CH32V20x_D8W、CH32F20x_D8W CH32F20x_D8。*	0
1	USBDPU	RW	USBD 内部上拉电阻是否启用： 1：启用（外部不用接上拉电阻）； 0：不启用（外部要接上拉电阻）。	0
0	USBDLS	RW	USBD 工作模式选择： 1：低速模式； 0：全速模式。	0

第 4 章 后备寄存器（BKP）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

后备寄存器（BKP）提供了 10 个 16 位的通用数据寄存器，可以用来存储 20 字节的用户数据。在主电源（V_{DD}）掉电后，这些数据仍可以由 V_{BAT} 供电而保持，不受待机状态、系统复位或电源复位的影响。此外 BKP 单元还提供了侵入检测管理、RTC 时钟校准及脉冲输出功能。

4.1 主要特征

- 提供 20 字节的后备数据寄存器
- 侵入检测（TAMPER）功能
- RTC 时钟校准功能
- 在 PC13 引脚上输出 RTC 时钟 64 分频，闹钟脉冲或者秒脉冲

4.2 功能说明

微控制器复位后对后备寄存器和 RTC 的访问被禁止，需通过以下操作开启对后备寄存器的访问：
1）置寄存器 RCC_APB1PCENR 的 PWREN 位和 BKPEN 位来打开电源和后备接口的操作时钟；
2）置电源控制寄存器 PWR_CTLR 的 DBP 位，使能对后备寄存器和 RTC 寄存器的访问。

4.2.1 后备数据寄存器

后备数据寄存器可以作为通用数据缓存使用，由于其在 V_{DD} 掉电下靠 V_{BAT} 电源保存数据的特性，可以用来存一些重要的或敏感的数据。但这些数据在产生侵入事件后会被全部清除。

4.2.2 侵入检测

侵入检测就是当外界提供了一个信号（上升沿或下降沿）时，表示有“侵入事件”，硬件将自动清除当前系统中保留的重要信息。这种方式可以增加系统信息的安全性。

当侵入检测引脚上出现跳变沿（取决于 TPAL 位）时会产生一个侵入事件，如果使能了侵入检测中断，还会同时产生一个侵入检测中断。只要出现了侵入事件，后备数据寄存器就会被全部清除。此外，硬件检测采用记忆方式，即使侵入检测功能未开启（TPE=0），系统也会采样是否有跳变沿，并在满足 TPAL 位选择情况下，提前锁定侵入事件，并在 TPE 位置 1 下，触发侵入事件。

例如：当 TPAL=0 时，如果 TPE=0 未开启功能，但 TAMPER 引脚已经为高电平，一旦 TPE=1 后，则会产生一个额外的侵入事件（系统提前锁定了上升沿）。当 TPAL=1 时，如果 TPE=0 未开启功能，但 TAMPER 引脚已经为低电平，一旦 TPE=1 后，则会产生一个额外的侵入事件（系统提前锁定了下降沿）。

所以为了防止发生不必要的侵入事件，导致清除了后备寄存器，建议：在希望硬件检测侵入引脚的开始时刻，通过写 BKP_TPCSR 寄存器 CTE 位置 1，先清除硬件可能记忆过的侵入事件，并确保当前侵入检测引脚状态是无效的。

注：当 V_{DD} 电源断开时，侵入检测功能仍然有效。为了避免不必要的复位数据后备寄存器，TAMPER 引脚应该在片外连接到正确的电平。

4.2.3 RTC 校准

此功能必须配置侵入检测引脚作为普通 IO 口使用。配置 BKP_TPCTLR 寄存器 TPE 位清 0。

- 脉冲输出

配置 BKP_OCTLR 寄存器的 ASOE 位，开启 RTC 脉冲输出，设置 ASOS 位，选择秒脉冲输出还是闹钟脉冲输出。

- RTC 校准

配置 BKP_OCTLR 寄存器的 CCO 位后，内部的 RTC 时钟将经过 64 分频后输出到侵入检测引脚（TAMPER）上。通过实际测试，软件配合修改 CAL[6:0]位来调整时钟对 RTC 进行校准。

4.2.4 BKP 接口复位

BKP 区域可以在 V_{DD} 主电源掉电下，由 V_{BAT} 独立供电。应用代码控制 BKP 区域寄存器复位中，后备数据寄存器 BKP_DATAR1-10、ASOS 位、ASOE 位在软件配置 RCC_BDCTLR 寄存器的 BDRST 位下复位，不受 RCC 外设接口控制 BKPRST 位影响。

4.3 寄存器描述

表 4-1 BKP 相关寄存器列表

名称	访问地址	描述	复位值
R16_BKP_DATAR1	0x40006C04	后备数据寄存器 1	0x0000
R16_BKP_DATAR2	0x40006C08	后备数据寄存器 2	0x0000
R16_BKP_DATAR3	0x40006C0C	后备数据寄存器 3	0x0000
R16_BKP_DATAR4	0x40006C10	后备数据寄存器 4	0x0000
R16_BKP_DATAR5	0x40006C14	后备数据寄存器 5	0x0000
R16_BKP_DATAR6	0x40006C18	后备数据寄存器 6	0x0000
R16_BKP_DATAR7	0x40006C1C	后备数据寄存器 7	0x0000
R16_BKP_DATAR8	0x40006C20	后备数据寄存器 8	0x0000
R16_BKP_DATAR9	0x40006C24	后备数据寄存器 9	0x0000
R16_BKP_DATAR10	0x40006C28	后备数据寄存器 10	0x0000
R16_BKP_OCTLR	0x40006C2C	RTC 校准寄存器	0x0000
R16_BKP_TPCTLR	0x40006C30	侵入检测控制寄存器	0x0000
R16_BKP_TPCSR	0x40006C34	侵入检测状态寄存器	0x0000

4.3.1 后备数据寄存器（BKP_DATARx）（x=1-10）

偏移地址：0x04-0x28

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

位	名称	访问	描述	复位值
[15:0]	DATA	RW	后备数据，可以被用户程序调用。 *注：它们仅由后备域复位来复位（BDRST）或（如果侵入检测引脚 TAMPER 功能被开启时）由侵入引脚事件复位。*	0

4.3.2 RTC 校准寄存器（BKP_OCTLR）

偏移地址：0x2C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						ASOS	ASOE	CCO	CAL[6:0]						

位	名称	访问	描述	复位值
[15:10]	Reserved	RO	保留。	0
9	ASOS	RW	TAMPER 引脚闹钟/秒脉冲输出选择。 1：输出秒脉冲； 0：输出闹钟脉冲。 *注：此位只会由后备域复位（BDRST）来复位。*	0
8	ASOE	RW	TAMPER 引脚使能脉冲输出位 0：禁止输出闹钟脉冲或者秒脉冲； 1：使能输出闹钟脉冲或者秒脉冲。 *注：此位只会由后备域复位（BDRST）来复位。*	0
7	CCO	RW	校准时钟输出选择位 1：TEMPER 引脚输出经 64 分频的 RTC 时钟； 0：不输出校准时钟。 *注 1：开启此功能必须关闭侵入检测功能。* *注 2：当 V_{DD} 供电断开时，该位被清除。*	0
[6:0]	CAL	RW	校准值寄存器，这个寄存器的值表示在每 220 个时钟脉冲中有多少个被跳过。这个功能用来校准 RTC 时钟。RTC 时钟可以被减慢 0~121ppm。	0

4.3.3 侵入检测控制寄存器（BKP_TPCTLR）

偏移地址：0x30

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

位	名称	访问	描述	复位值
[15:2]	Reserved	RO	保留。	0
1	TPAL	RW	侵入检测引脚（TEMPER 引脚）有效电平设置 0：侵入检测引脚上的高电平会清除所有后备数据寄存器（硬件锁定上升沿）； 1：侵入检测引脚上的低电平会清除所有后备数据寄存器（硬件锁定下降沿）。	0
0	TPE	RW	侵入检测引脚使能位 0：TEMPER 引脚做普通 IO 口用； 1：TEMPER 引脚做侵入检测用。	0

注：同时将 TPAL 和 TPE 位清除会产生一个假的侵入事件，推荐只在 TPE 为 0 时才改变 TPAL 位的状态。

4.3.4 侵入检测状态寄存器（BKP_TPCSR）

偏移地址：0x34

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TIF	TEF	Reserved					TPIE	CTI	CTE

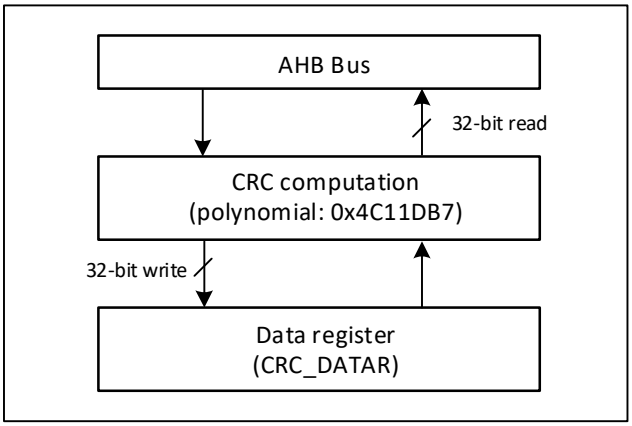
位	名称	访问	描述	复位值
[15:10]	Reserved	RO	保留。	0
9	TIF	RO	侵入中断标志位，当检测到侵入事件且 TPIE 位置 1 时，此位会被置位。通过向 CTI 位写 1 来清除此标志位。如果 TPIE 位被复位，那么此位同时也会被复位。 *注：仅当系统复位或由待机模式唤醒后才复位该位。*	0
8	TEF	RO	侵入事件标志位，当检测到侵入事件时，此位会被置位。通过向 CTE 位写 1 会清除此位。 *注：当此位为 1 时，所有的 BKP_DATARx 寄存器的值会被清除，且在此位不复位前，所有对 BKP_DATARx 寄存器的写入操作都是无效的。*	0
[7:3]	Reserved	RO	保留。	0
2	TPIE	RW	产生侵入中断使能位： 0：禁止侵入检测中断； 1：使能侵入检测中断（TPE 需置 1）。 *注 1：侵入中断无法将内核从低功耗模式唤醒。* *注 2：仅当系统复位或由待机模式唤醒后才复位该位。*	0
1	CTI	WO	侵入检测中断清除位，写 1 清除，读取无效。	0
0	CTE	WO	侵入检测事件清除位，写 1 清除，读取无效。	0

第 5 章 循环冗余校验(CRC)

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

循环冗余校验(CRC)计算单元是根据固定的生成多项式得到任一 32 位数据的 CRC 计算结果。一般用于数据存储和数据通讯领域用来核实数据的正确性。系统提供硬件 CRC 计算单元可以大大节省 CPU 和 RAM 资源提高效率。

图 5-1 CRC 结构框图



5.1 主要特征

- 使用 CRC32 多项式（0x4C11DB7）：$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$；
- 同一个 32 位寄存器作为数据的输入和 CRC32 计算输出
- 单次转换时间：4 个 AHB 时钟周期（HCLK）

5.2 功能描述

- CRC 单元复位

如果要开始一次新数据组的 CRC 计算，需要复位 CRC 计算单元。向控制寄存器 CRC_CTLR 的 RST 位写 1，硬件将复位数据寄存器，恢复初始值 0xFFFFFFFF。

- CRC 计算

CRC 单元的计算是前一次 CRC 计算结果和新参与的数据的 CRC 结果。CRC_DATAR 数据寄存器，对其执行写操作将送入新数据到硬件计算单元；执行读取操作，将得到最新一轮的 CRC 计算值。硬件计算时会中断系统的写操作，因此可以连续写入新的值。

注：CRC 单元是对整个 32 位数据进行计算，而不是逐字节计算。

- 独立数据缓冲区

CRC 单元提供了一个 8 位独立数据寄存器 CRC_IDATAR，用于应用代码临时存放 1 字节的数据，不受 CRC 单元复位影响。

5.3 寄存器描述

表 5-1 CRC 相关寄存器列表

名称	访问地址	描述	复位值
R32_CRC_DATAR	0x40023000	数据寄存器	0xFFFFFFFF
R8_CRC_IDATAR	0x40023004	独立数据缓冲	0x00
R32_CRC_CTLR	0x40023008	控制寄存器	0x00000000

5.3.1 数据寄存器（CRC_DATAR）

偏移地址：0x00

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

位	名称	访问	描述	复位值
[31:0]	DATA	RW	写入原始数据；读出计算结果。	0xFFFFFFFF

5.3.2 独立数据缓冲（CRC_IDATAR）

偏移地址：0x04

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

位	名称	访问	描述	复位值
[7:0]	IDATA	RW	8 位通用寄存器，可以用作数据缓存，这个寄存器不受控制寄存器的 RST 域影响。	0

5.3.3 控制寄存器（CRC_CTLR）

偏移地址：0x08

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

位	名称	访问	描述	复位值
[31:1]	Reserved	RO	保留。	0
0	RST	WO	CRC 计算单元复位控制，写 1 执行，硬件自动清零，执行完后，数据寄存器为 0xFFFFFFFF。	0

第 6 章 实时时钟（RTC）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

实时时钟（RTC）是一个独立的定时器模块，其可编程计数器最大可达到 32 位，配合软件即可以实现实时时钟功能，并且可以修改计数器的值来重新配置系统的当前时间和日期。RTC 模块在后备供电区域，系统复位和待机模式唤醒对其不造成影响。

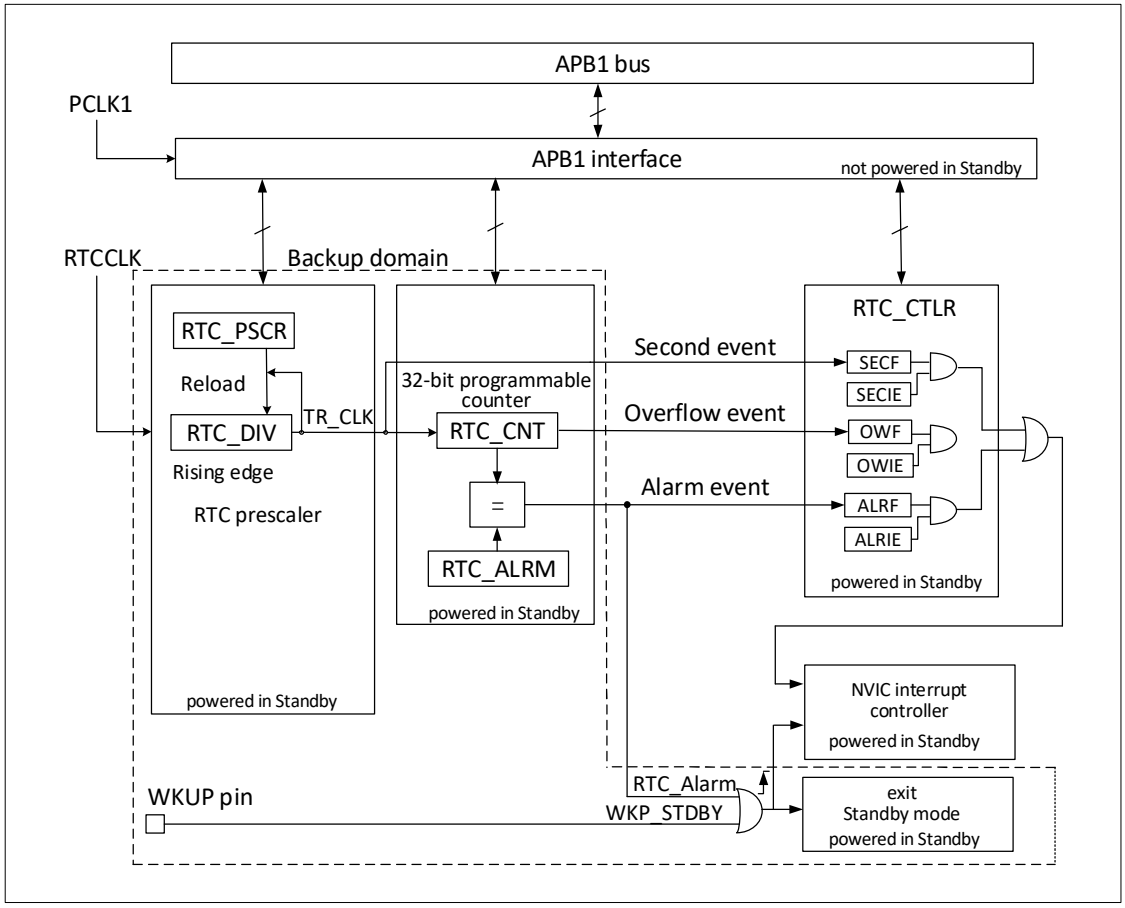
6.1 主要特征

- 最高为 2^{20} 的预分频系数
- 32 位可编程计数器
- 多种时钟源，中断
- 独立复位

6.2 功能描述

6.2.1 概述

图 6-1 RTC 结构框图



由图 6-1 所示，RTC 模块主要是 APB1 总线接口、分频器和计数器、控制和状态寄存器三部分组成，其中分频器和计数器部分在后备区域，可由 V_{BAT} 供电。RTCCK 输入分频器（RTC_DIV）之后，被分频成 TR_CLK。值得注意的是，分频器（RTC_DIV）的内部是一个自减计数器，自减到溢出就会输出一个 TR_CLK，然后从重装值寄存器（RTC_PSCR）里取出预设值重装到分频器里，读分频器实际上是读取它的实时值（read only），写分频系数应该写到重装值寄存器（RTC_PSCR）里。一般 TR_CLK 的周

期被设置为 1 秒，TR_CLK 会触发秒事件，同时会使主计数器（RTC_CNT）自增 1；当主计数器增加到和闹钟寄存器的值一致时，会触发闹钟事件；当主计数器自增到溢出时，会触发溢出事件。以上三种事件都可以触发中断，并对应相应中断使能位控制。

6.2.2 复位

由于实时时钟的特殊用途，其处于后备域的四组寄存器：预分频，预分频重装值，主计数器和闹钟，只能通过后备域的复位信号复位，参照 RCC 的后备域复位章节。实时时钟的控制寄存器受系统复位或电源复位控制。

6.2.3 较特别的读写寄存器操作

由于实时时钟的特殊用处，RTC 和 APB1 总线是独立的，APB1 对 RTC 的读取不一定是实时的，通过 APB1 读取 RTC 的寄存器必须在 APB1 启动后并经过了一个 RTC 上升沿，这种情形可能出现在系统复位和电源复位之后、从待机或者停机模式唤醒后。方便的做法是等待控制寄存器（CTLR）的 RSF 位被置高。对 RTC 的写操作器必须等上一个写操作结束，且必须进入配置模式，具体的步骤为：

1） 查询 RTOFF 位，直到其变为 1；
2） 置 CNF 位，进入配置模式；
3） 对一个或者多个 RTC 寄存器进行写操作；
4） 清 CNF 位，退出配置模式，APB1 接口开始对 RTC 寄存器进行写入；
5） 查询 RTOFF 位，直到其变为 1 即为写完；

6.3 寄存器描述

表 6-1 RTC 相关寄存器列表

名称	访问地址	描述	复位值
R16_RTC_CTLRH	0x40002800	RTC 控制寄存器高位	0x0000
R16_RTC_CTLRL	0x40002804	RTC 控制寄存器低位	0x0000
R16_RTC_PSCRH	0x40002808	预分频器重装值寄存器高位	0x0000
R16_RTC_PSCRL	0x4000280C	预分频器重装值寄存器低位	0x0000
R16_RTC_DIVH	0x40002810	分频器寄存器高位	0x0000
R16_RTC_DIVL	0x40002814	分频器寄存器低位	0x0000
R16_RTC_CNTH	0x40002818	RTC 计数器高位	0x0000
R16_RTC_CNTL	0x4000281C	RTC 计数器低位	0x0000
R16_RTC_ALRMH	0x40002820	闹钟寄存器高位	0xFFFF
R16_RTC_ALRML	0x40002824	闹钟寄存器低位	0xFFFF

6.3.1 RTC 控制寄存器高位（RTC_CTLRH）

偏移地址：0x00

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

位	名称	访问	描述	复位值
[15:3]	Reserved	RO	保留。	0
2	OWIE	RW	溢出中断使能位。	0
1	ALRIE	RW	闹钟中断使能位。	0
0	SECIE	RW	秒中断使能位。	0

6.3.2 RTC 控制寄存器低位（RTC_CTLRL）

偏移地址：0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RTOFF	CNF	RSF	OWF	ALRF	SECF

位	名称	访问	描述	复位值
[15:6]	Reserved	RO	保留。	0
5	RTOFF	RO	RTC 操作状态指示位，表示对 RTC 的最后一次操作的执行状态，对 RTC 的操作必须等待此位为 1。 1：上一次对 RTC 的操作已经完成； 0：上一次对 RTC 的操作还在进行中。	1
4	CNF	RW	配置标志位，将此位写 1 进入配置模式，从而允许向计数器（R16_RTC_CNTx）、闹钟寄存器（R16_RTC_ALRMx）和预分频器重装值寄存器(R16_RTC_PSCRx)写入值.只有将该位写 1 并重新被软件清 0 后才会执行写的操作： 1：进入配置模式； 0：退出配置模式，开始更新 RTC 寄存器。	0
3	RSF	RWO	寄存器同步标志位，在对 RTC 模块的预分频(PSCRx)、闹钟(ALRMx)、计数器(CNTx)这些寄存器进行读写前，都要先保证这个位已经被硬件置位，以确定这些寄存器已经被同步；在进行读写这些寄存器时，或者 APB1 复位或 APB1 时钟停止后，第一步应该将此位复位。 1：寄存器已被同步； 0：寄存器未被同步。	0
2	OWF	RWO	计数器溢出标志，当 32 位计数器溢出时，此位由硬件置位。如果置位了 OWIE 位，还会产生一个溢出中断。此位只能由软件清零，不能被软件置位。	0
1	ALRF	RWO	闹钟标志，当计数器的值达到闹钟寄存器（ALRMx）的值，此位会被硬件置位，如果闹钟中断使能位（ALRIE）置位，还会产生一个闹钟中断。此位只能由软件清零，不能被软件置位。	0
0	SECF	RWO	秒事件标志，当时钟经过预分频器分频后每产生一个下降沿，就会使计数器自增一，同时产生一个秒事件，此位会被置位，如果秒中断被使能（SECIE 被置位），同时还会产生一个秒	0

			中断。此位只能由软件清零，不能被软件置位。	

6.3.3 预分频器重装值寄存器高位（RTC_PSCRH）

偏移地址：0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PSCR[19:16]			

位	名称	访问	描述	复位值
[15:4]	Reserved	RO	保留。	0
[3:0]	PSCRH	RW	重装值高位。	0

6.3.4 预分频器重装值寄存器低位（RTC_PSCRL）

偏移地址：0x0C

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

位	名称	访问	描述	复位值
[15:0]	PSCRL	RW	重装值低位。实际的分频系数就是（PSCR[19:0]+1），比如如果 RTC 输入频率为 32768Hz，那么这个值设为 0x7fff 就可以分频出 1 秒周期的信号。	8000h

6.3.5 分频器寄存器高位（RTC_DIVH）

偏移地址：0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												DIV[19:16]			

位	名称	访问	描述	复位值
[15:4]	Reserved	RO	保留。	0
[3:0]	DIVH	RO	分频器寄存器高位。	0

6.3.6 分频器寄存器低位（RTC_DIVL）

偏移地址：0x14

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

位	名称	访问	描述	复位值
[15:0]	DIVL	RO	分频器寄存器低位。DIV 实际上是一个自减计数器，RTC_CLK 每来一个时钟 DIV 计数器就会减 1，溢出后就会输出一个 TR_CLK，同时从 PSCR 中重装载值。DIV 只能读取，读出的是当前分频器的计数器的剩余值。	8000h

6.3.7 RTC 计数器高位（RTC_CNTH）

偏移地址：0x18

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

位	名称	访问	描述	复位值
[15:0]	CNTH	RW	计数器高位。	0

6.3.8 RTC 计数器低位（RTC_CNTL）

偏移地址：0x1C

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

位	名称	访问	描述	复位值
[15:0]	CNTL	RW	计数器低位，RTC 定时器的核心器件，由 TRCLK（周期一般设为 1 秒）提供时钟。通过读取 CNT[31:0]来计算出当前的时间。写这个值需要进入配置模式。	0

6.3.9 闹钟寄存器高位（RTC_ALRMH）

偏移地址：0x20

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

位	名称	访问	描述	复位值
[15:0]	ALRMH	RW	闹钟寄存器高位。	FFFFh

6.3.10 闹钟寄存器低位（RTC_ALRML）

偏移地址：0x24

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

位	名称	访问	描述	复位值
[15:0]	ALRML	RW	闹钟寄存器低位。当闹钟寄存器 ALRM[31:0]的值和计数器 CNT[31:0]的值一致时会产生一个闹钟事件。更改这个值需要进入配置模式。	FFFFh

第 7 章 独立看门狗（IWDG）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

系统设有独立看门狗（IWDG）用来检测逻辑错误和外部环境干扰引起的软件故障。IWDG 时钟源来自于 LSI，可独立于主程序之外运行，适用于对精度要求低的场合。

7.1 主要特征

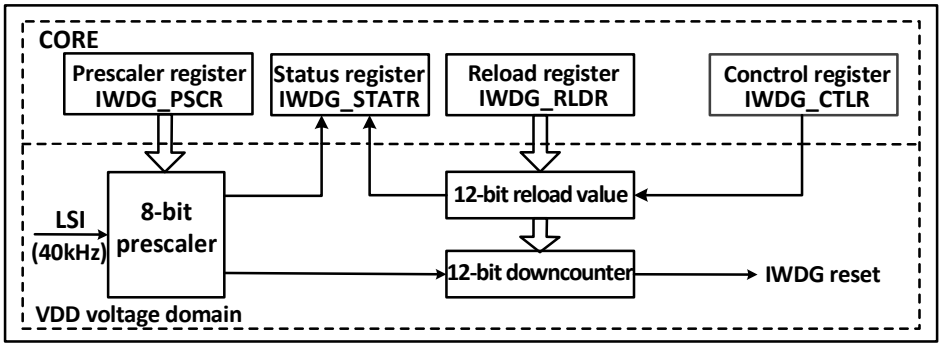
- 12 位自减型计数器
- 时钟来源 LSI 分频，可以在低功耗模式下运行
- 复位条件：计数器值减到 0

7.2 功能说明

7.2.1 原理和用法

独立看门狗的时钟来源 LSI 时钟分频，其功能在停机和待机模式时仍能正常工作。当看门狗计数器自减到 0 时，将会产生系统复位，所以超时时间为（重装载值+1）个时钟。

图 7-1 独立看门狗的结构框图



- 启动独立看门狗

系统复位后，看门狗处于关闭状态，向 IWDG_CTLR 寄存器写 0xCCCC 开启看门狗，随后它不能再被关闭，除非发生复位。

如果在用户选择字开启了硬件独立看门狗使能位(IWDG_SW),在微控制器复位后将固定开启 IWDG。

- 看门狗配置

看门狗内部是一个递减运行的 12 位计数器，当计数器的值减为 0 时，将发生系统复位。开启 IWDG 功能，需要执行下面几点操作：

1) 计数时基：IWDG 时钟来源 LSI，通过 IWDG_PSCR 寄存器设置 LSI 分频值时钟作为 IWDG 的计数时基。操作方法先向 IWDG_CTLR 寄存器写 0x5555,再修改 IWDG_PSCR 寄存器中的分频值。IWDG_STATR 寄存器中的 PVU 位指示了分频值更新状态，在更新完成的情况下才可以进行分频值的修改和读出。
2) 重装载值：用于更新独立看门狗中计数器当前值，并且计数器由此值进行递减。操作方法先向 IWDG_CTLR 寄存器写 0x5555，再修改 IWDG_RLDR 寄存器设置目标重装载值。IWDG_STATR 寄存器中的 RUV 位指示了重装载值更新状态，在更新完成的情况下才可以进行 IWDG_RLDR 寄存器的修改和读出。
3) 看门狗使能：向 IWDG_CTLR 寄存器写 0xCCCC，即可开启看门狗功能。

4) 喂狗：即在看门狗计数器递减到 0 前刷新当前计数器值防止发生系统复位。向 IWDG_CTLR 寄存器写 0xAAAA，让硬件将 IWDG_RLDR 寄存值更新到看门狗计数器中。此动作需要在看门狗功能开启后定时执行，否则会出现看门狗复位动作。

7.2.2 调试模式

系统进入调试模式时，可以由调试模块寄存器配置 IWDG 的计数器继续工作或停止。

7.3 寄存器描述

表 7-1 IWDG 相关寄存器列表

名称	访问地址	描述	复位值
R16_IWDG_CTLR	0x40003000	控制寄存器	0x0000
R16_IWDG_PSCR	0x40003004	分频因子寄存器	0x0000
R16_IWDG_RLDR	0x40003008	重装载值寄存器	0x7FFF
R16_IWDG_STATR	0x4000300C	状态寄存器	0x0000

7.3.1 IWDG 控制寄存器（IWDG_CTLR）

偏移地址：0x00

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

位	名称	访问	描述	复位值
[15:0]	KEY	WO	操作键值锁。 0xAAAA：喂狗。加载 IWDG_RLDR 寄存器值到独立看门狗计数器中； 0x5555：允许修改 R16_IWDG_PSCR 和 R16_IWDG_RLDR 寄存器； 0xCCCC：启动看门狗，如果启用了硬件看门狗（用户选择字配置）则不受这个限制。	0

7.3.2 分频因子寄存器（IWDG_PSCR）

偏移地址：0x04

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

位	名称	访问	描述	复位值
[15:3]	Reserved	RO	保留。	0
[2:0]	PSCR	RW	IWDG 时钟分频系数，修改此域前要向 KEY 中写 0x5555。 000：4 分频；　001：8 分频； 010：16 分频；　011：32 分频； 100：64 分频；　101：128 分频； 110：256 分频；　111：256 分频。	0

			IWDG 计数时基=LSI/分频系数。 *注：读该域值前，要确保 IWDG_STATR 寄存器中的 PVU 位为 0，否则读出值无效。*	

7.3.3 重装载值寄存器（IWDG_RLDR）

偏移地址：0x08

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

位	名称	访问	描述	复位值
[15:12]	Reserved	RO	保留。	0
[11:0]	RLDR	RW	计数器重装载值。修改此域前要向 KEY 中写 0x5555。 当向 KEY 中写 0xAAAA 后，此域的值将会被硬件装载到计数器中，随后计数器从这个值开始递减计数。 *注：读写该域值前，要确保 IWDG_STATR 寄存器中的 RUV 位为 0，否则读写此域无效。*	FFFh

注：此寄存器在待机模式下会被复位。

7.3.4 状态寄存器（IWDG_STATR）

偏移地址：0x0C

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

位	名称	访问	描述	复位值
[15:2]	Reserved	RO	保留。	0
1	RUV	RO	重装值更新标志位。硬件置位或清 0。 1：重装载值更新正在进行中； 0：重装载更新结束（最多 5 个 LSI 周期）。 *注：重装载值寄存器 IWDG_RLDR 只有在 RVU 位被清 0 后才可读写访问。*	0
0	PVU	RO	时钟分频系数更新标志位。硬件置位或清 0。 1：时钟分频值更新正在进行中； 0：时钟分频值更新结束（最多 5 个 LSI 周期）。 *注：分频因子寄存器 IWDG_PSCR 只有在 PVU 位被清 0 后才可读写访问。*	0

注：在预分频或重装值更新后，不必等待 RVU 或 PVU 复位，可继续执行下面的代码。（即使在低功耗模式下，此写操作仍会被继续执行完成。）

第 8 章 窗口看门狗（WWDG）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

窗口看门狗一般用来监测系统运行的软件故障，例如外部干扰、不可预见的逻辑错误等情况。它需要在一个特定的窗口时间（有上下限）内进行计数器刷新（喂狗），否则早于或者晚于这个窗口时间看门狗电路都会产生系统复位。

8.1 主要特征

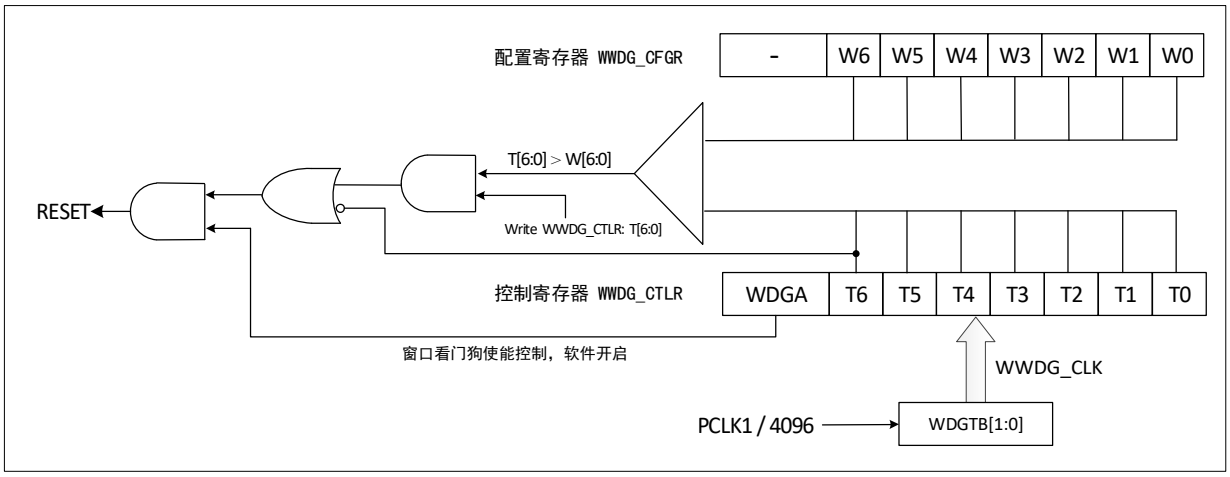
- 可编程的 7 位自减型计数器
- 双条件复位：当前计数器值小于 0x40，或者计数器值在窗口时间外被重装载
- 唤醒提前通知功能（EWI），用于及时喂狗动作防止系统复位

8.2 功能说明

8.2.1 原理和用法

窗口看门狗运行基于一个 7 位的递减计数器，其挂载在 APB1 总线下，计数时基 WWDG_CLK 来源（PCLK1/4096）时钟的分频，分频系数在配置寄存器 WWDG_CFGR 中的 WDGTB[1:0]域设置。递减计数器处于自由运行状态，无论看门狗功能是否开启，计数器一直循环递减计数。如图 8-1 所示，窗口看门狗内部结构框图。

图 8-1 窗口看门狗结构框图



- 启动窗口看门狗

系统复位后，看门狗处于关闭状态，设置 WWDG_CTLR 寄存器的 WDGA 位能够开启看门狗，随后它不能再被关闭，除非发生复位。

注：可以通过设置 RCC_APB1PCENR 寄存器关闭 WWDG 的时钟来源，暂停 WWDG_CLK 计数，间接停止看门狗功能，或者通过设置 RCC_APB1PRSTR 寄存器复位 WWDG 模块，等效为复位的作用。

- 看门狗配置

看门狗内部是一个不断循环递减运行的 7 位计数器，支持读写访问。使用看门狗复位功能，需要执行下面几点操作：

1） 计数时基：通过 WWDG_CFGR 寄存器的 WDGTB[1:0]位域，注意要开启 RCC 单元的 WWDG 模块时钟。

2） 窗口计数器：设置 WWDG_CFGR 寄存器的 W[6:0]位域，此计数器由硬件用作和当前计数器比较使

用，数值由用户软件配置，不会改变。作为窗口时间的上限值。

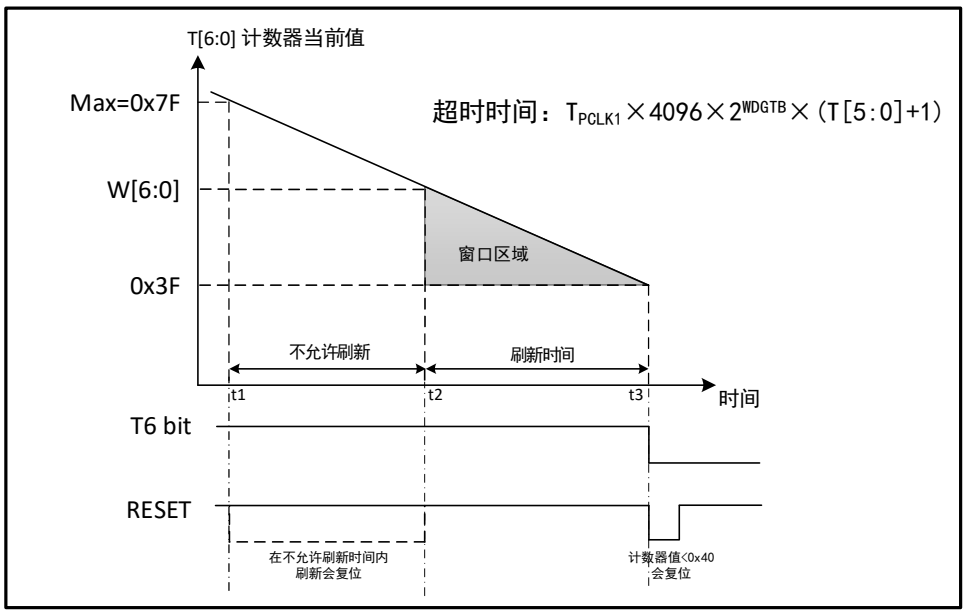
3） 看门狗使能：WWDG_CTLR 寄存器 WDGA 位软件置 1，开启看门狗功能，可以系统复位。

4） 喂狗：即刷新当前计数器值，配置 WWDG_CTLR 寄存器的 T[6:0]位域。此动作需要在看门狗功能开启后，在周期性的窗口时间内执行，否则会出现看门狗复位动作。

- 喂狗窗口时间

如图 8-2 所示，灰色区域为窗口看门狗的监测窗口区域，其上限时间 t2 对应当前计数器值达到窗口值 W[6:0]的时间点;其下限时间 t3 对应当前计数器值达到 0x3F 的时间点。此区域时间内 t2<t<t3 可以进行喂狗操作（写 T[6:0]），刷新当前计数器的数值。

图 8-2 窗口看门狗的计数模式



- 看门狗复位：

1） 当没有及时喂狗操作，导致 T[6:0]计数器的值由 0x40 变成 0x3F，将出现“窗口看门狗复位”，产生系统复位。即 T6-bit 被硬件检测为 0，将出现系统复位。

注：应用程序可以通过软件写 T6-bit 为 0，实现系统复位，等效软件复位功能。

2） 当在不允许喂狗时间内执行计数器刷新动作，即在 t1≤t≤t2 时间内操作写 T[6:0]位域，将出现“窗口看门狗复位”，产生系统复位。

- 提前唤醒

为了防止没有及时刷新计数器导致系统复位，看门狗模块提供了早期唤醒中断（EWI）通知。当计数器自减到 0x40 时，产生提前唤醒信号，WEIF 标志置 1，如果置位了 EWI 位，会同时触发窗口看门狗中断。此时距离硬件复位有 1 个计数器时钟周期（自减为 0x3F），应用程序可在此时间内即时进行喂狗操作。

8.2.2 调试模式

系统进入调试模式时，可以由调试模块寄存器配置 WWDG 的计数器继续工作或停止。

8.3 寄存器描述

表 8-1 WWDG 相关寄存器列表

名称	访问地址	描述	复位值
R16_WWDG_CTLR	0x40002C00	控制寄存器	0x007F
R16_WWDG_CFGR	0x40002C04	配置寄存器	0x0000
R16_WWDG_STATR	0x40002C08	状态寄存器	0x0000

8.3.1 WWDG 控制寄存器（WWDG_CTLR）

偏移地址：0x00

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

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
7	WDGA	RW1	窗口看门狗复位使能位。 1：开启看门狗功能（可产生复位信号）； 0：禁止看门狗功能。 软件写 1 开启，但是只允许复位后硬件清 0。	0
[6:0]	T	RW	7 位自减计数器，每 $4096*2^{WDGTB}$ 个 PCLK1 周期自减 1。当计数器从 0x40 自减到 0x3F 时，即 T6 跳变为 0 时，产生看门狗复位。	7Fh

8.3.2 WWDG 配置寄存器（WWDG_CFGR）

偏移地址：0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						EWI	WDGTB[1:0]		W[6:0]						

位	名称	访问	描述	复位值
[15:10]	Reserved	RO	保留。	0
9	EWI	RW1	提前唤醒中断使能位。 若此位置 1，则在计数器的值达到 0x40 时产生中断。此位只能在复位后由硬件请 0。	0
[8:7]	WDGTB	RW	窗口看门狗时钟分频选择： 00：1 分频，计数时基 = PCLK1/4096； 01：2 分频，计数时基 = PCLK1/4096/2； 10：4 分频，计数时基 = PCLK1/4096/4； 11：8 分频，计数时基 = PCLK1/4096/8。	0
[6:0]	W	RW	窗口看门狗 7 位窗口值。用来与计数器的值做比较。喂狗操作只能在计数器的值小于窗口值且大于 0x3F 时进行。	7Fh

8.3.3 WWDG 状态寄存器（WWDG_STATR）

偏移地址：0x04

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

位	名称	访问	描述	复位值
[15:1]	Reserved	WO	保留。	0
0	WEIF	RWO	提前唤醒中断标志位。 当计数器到达 0x40 时，此位会被硬件置位，必须通过软件清 0，用户置位是无效的。即使 EWI 未被置位，此位在事件发生时仍会照常被置位。	0

第 9 章 中断和事件(NVIC/PFIC)

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

CH32F2x 系列产品采用 Cortex-M3 内核，内置嵌套向量中断控制器(NVIC - Nested Vectored Interrupt Controller)，管理了 88 个可屏蔽外部中断通道和 10 个内核中断通道，其他中断源保留。中断控制器与内核接口紧密相连，以最小的中断延迟提供了灵活的中断管理功能。具体关于 NVIC 控制器的使用说明请参考 Cortex-M3 相关文档说明。

CH32V2x 和 CH32V3x 系列内置可编程快速中断控制器（PFIC - Programmable Fast Interrupt Controller），最多支持 255 个中断向量。当前系统管理了 88 个外设中断通道和 8 个内核中断通道，其他保留。

9.1 主要特征

9.1.1 NVIC 控制器

- 88个可屏蔽的中断通道
- 提供不可屏蔽中断第一时间响应
- 向量化的中断设计实现向量入口地址直接进入内核
- 中断进入和退出时自动压栈和恢复，无需额外指令开销
- 16级嵌套，优先级动态修改

9.1.2 PFIC 控制器

- 88个外设中断，每个中断请求都有独立的触发和屏蔽控制位，有专用的状态位
- 可编程多级中断嵌套，最大嵌套深度8级，硬件压栈深度3级
- 特有快速中断进出机制，硬件自动压栈和恢复，无需指令开销
- 特有免表VTF（Vector Table Free）中断响应机制，4路可编程直达中断向量地址

9.2 系统定时器

- CH32F2x 系列产品

  Cortex-M3 内核自带了一个 24 位自减型计数器（SysTick timer）。支持 HCLK 或 HCLK/8 作为时基，具有较高优先级别（6）。一般可用于操作系统的时基。具体请参考 Cortex-M3 相关文档说明。
- CH32V3x 系列产品

  内核自带了一个 64 位加减计数器（SysTick），支持 HCLK 或者 HCLK/8 作为时基，具有较高优先级，校准后可用于时间基准。

9.3 中断和异常的向量表

表 9-1 CH32F2x 系列产品向量表

位置	优先级	优先级类型	名称	说明	绝对地址
	-	-	-	保留	0x00000000
	-3	固定	Reset	复位	0x00000004
	-2	固定	NMI	不可屏蔽中断	0x00000008
	-1	固定	HardFault	所有类型的失效	0x0000000C

	0	可设置	MemManage	存储器管理	0x00000010
	1	可设置	BusFault	预取指失败，存储器访问失败	0x00000014
	2	可设置	UsageFault	未定义的指令或非法状态	0x00000018
	-	-	-	保留	0x0000001C-0x0000002B
	3	可设置	SVCall	通过 SWI 指令的系统服务调用	0x0000002C
	4	可设置	Debug Monitor	调试监控器	0x00000030
	-	-	-	保留	0x00000034
	5	可设置	PendSV	可挂起的系统服务	0x00000038
	6	可设置	SysTick	系统滴答定时器	0x0000003C
0	7	可编程	WWDG	窗口定时器中断	0x00000040
1	8	可编程	PVD	电源电压检测中断（EXTI）	0x00000044
2	9	可编程	TAMPER	侵入检测中断	0x00000048
3	10	可编程	RTC	实时时钟中断	0x0000004C
4	11	可编程	FLASH	闪存全局中断	0x00000050
5	12	可编程	RCC	复位和时钟控制中断	0x00000054
6	13	可编程	EXTI0	EXTI 线 0 中断	0x00000058
7	14	可编程	EXTI1	EXTI 线 1 中断	0x0000005C
8	15	可编程	EXTI2	EXTI 线 2 中断	0x00000060
9	16	可编程	EXTI3	EXTI 线 3 中断	0x00000064
10	17	可编程	EXTI4	EXTI 线 4 中断	0x00000068
11	18	可编程	DMA1_CH1	DMA1 通道 1 全局中断	0x0000006C
12	19	可编程	DMA1_CH2	DMA1 通道 2 全局中断	0x00000070
13	20	可编程	DMA1_CH3	DMA1 通道 3 全局中断	0x00000074
14	21	可编程	DMA1_CH4	DMA1 通道 4 全局中断	0x00000078
15	22	可编程	DMA1_CH5	DMA1 通道 5 全局中断	0x0000007C
16	23	可编程	DMA1_CH6	DMA1 通道 6 全局中断	0x00000080
17	24	可编程	DMA1_CH7	DMA1 通道 7 全局中断	0x00000084
18	25	可编程	ADC1_2	ADC1 和 ADC2 全局中断	0x00000088
19	26	可编程	USB_HP 或	USB_HP 或 CAN1_TX 全局中断	0x0000008C
20	27	可编程	USB_LP 或	USB_LP 或 CAN1_RX0 全局中断	0x00000090
21	28	可编程	CAN1_RX1	CAN1_RX1 全局中断	0x00000094
22	29	可编程	CAN1_SCE	CAN1_SCE 全局中断	0x00000098
23	30	可编程	EXTI9_5	EXTI 线[9:5]中断	0x0000009C
24	31	可编程	TIM1_BRK	TIM1 刹车中断	0x000000A0
25	32	可编程	TIM1_UP	TIM1 更新中断	0x000000A4
26	33	可编程	TIM1_TRG_COM	TIM1 触发和通信中断	0x000000A8
27	34	可编程	TIM1_CC	TIM1 捕获比较中断	0x000000AC
28	35	可编程	TIM2	TIM2 全局中断	0x000000B0
29	36	可编程	TIM3	TIM3 全局中断	0x000000B4
30	37	可编程	TIM4	TIM4 全局中断	0x000000B8
31	38	可编程	I2C1_EV	I^2C1 事件中断	0x000000BC
32	39	可编程	I2C1_ER	I^2C1 错误中断	0x000000C0
33	40	可编程	I2C2_EV	I^2C2 事件中断	0x000000C4
34	41	可编程	I2C2_ER	I^2C2 错误中断	0x000000C8

35	42	可编程	SPI1	SPI1 全局中断	0x000000CC
36	43	可编程	SPI2	SPI2 全局中断	0x000000D0
37	44	可编程	USART1	USART1 全局中断	0x000000D4
38	45	可编程	USART2	USART2 全局中断	0x000000D8
39	46	可编程	USART3	USART3 全局中断	0x000000DC
40	47	可编程	EXTI15_10	EXTI 线[15:10]中断	0x000000E0
41	48	可编程	RTCAlarm	RTC 闹钟中断（EXTI）	0x000000E4
42	49	可编程	USBWakeUp	USB 唤醒中断（EXTI）	0x000000E8
43	50	可编程	TIM8_BRK	TIM8 刹车中断	0x000000EC
44	51	可编程	TIM8_UP	TIM8 更新中断	0x000000F0
45	52	可编程	TIM8_TRG_COM	TIM8 触发和通信中断	0x000000F4
46	53	可编程	TIM8_CC	TIM8 捕获比较中断	0x000000F8
47	54	可编程	RNG	RNG 全局中断	0x000000FC
48	55	可编程	FSMC	FSMC 全局中断	0x00000100
49	56	可编程	SDIO	SDIO 全局中断	0x00000104
50	57	可编程	TIM5	TIM5 全局中断	0x00000108
51	58	可编程	SPI3	SPI3 全局中断	0x0000010C
52	59	可编程	UART4	UART4 全局中断	0x00000110
53	60	可编程	UART5	UART5 全局中断	0x00000114
54	61	可编程	TIM6	TIM6 全局中断	0x00000118
55	62	可编程	TIM7	TIM7 全局中断	0x0000011C
56	63	可编程	DMA2_CH1	DMA2 通道 1 全局中断	0x00000120
57	64	可编程	DMA2_CH2	DMA2 通道 2 全局中断	0x00000124
58	65	可编程	DMA2_CH3	DMA2 通道 3 全局中断	0x00000128
59	66	可编程	DMA2_CH4	DMA2 通道 4 全局中断	0x0000012C
60	67	可编程	DMA2_CH5	DMA2 通道 5 全局中断	0x00000130
61	68	可编程	ETH	ETH 全局中断	0x00000134
62	69	可编程	ETH_WKUP	ETH 唤醒中断	0x00000138
63	70	可编程	CAN2_TX	CAN2_TX 全局中断	0x0000013C
64	71	可编程	CAN2_RX0	CAN2_RX0 全局中断	0x00000140
65	72	可编程	CAN2_RX1	CAN2_RX1 全局中断	0x00000144
66	73	可编程	CAN2_SCE	CAN2_SCE 全局中断	0x00000148
67	74	可编程	OTG_FS	全速 OTG 中断	0x0000014C
68	75	可编程	USBHSWakeUp	高速 USB 唤醒中断	0x00000150
69	76	可编程	USBHS	高速 USB 全局中断	0x00000154
70	77	可编程	DVP	DVP 全局中断	0x00000158
71	78	可编程	UART6	UART6 全局中断	0x0000015C
72	79	可编程	UART7	UART7 全局中断	0x00000160
73	80	可编程	UART8	UART8 全局中断	0x00000164
74	81	可编程	TIM9_BRK	TIM9 刹车中断	0x00000168
75	82	可编程	TIM9_UP	TIM9 更新中断	0x0000016C
76	83	可编程	TIM9_TRG_COM	TIM9 触发和通信中断	0x00000170
77	84	可编程	TIM9_CC	TIM9 捕获比较中断	0x00000174
78	85	可编程	TIM10_BRK	TIM10 刹车中断	0x00000178

79	86	可编程	TIM10_UP	TIM10 更新中断	0x0000017C
80	87	可编程	TIM10_TRG_COM	TIM10 触发和通信中断	0x00000180
81	88	可编程	TIM10_CC	TIM10 捕获比较中断	0x00000184
82	89	可编程	DMA2_CH6	DMA2 通道 6 全局中断	0x00000188
83	90	可编程	DMA2_CH7	DMA2 通道 7 全局中断	0x0000018C
84	91	可编程	DMA2_CH8	DMA2 通道 8 全局中断	0x00000190
85	92	可编程	DMA2_CH9	DMA2 通道 9 全局中断	0x00000194
86	93	可编程	DMA2_CH10	DMA2 通道 10 全局中断	0x00000198
87	94	可编程	DMA2_CH11	DMA2 通道 11 全局中断	0x0000019C

表 9-2 CH32V2x 和 CH32V3x 系列产品向量表

编号	优先级	类型	名称	描述	入口地址
0	-	-	-	-	0x00000000
1	-	-	-	-	0x00000004
2	-5	固定	NMI	不可屏蔽中断	0x00000008
3	-4	固定	HardFault	异常中断	0x0000000C
4	-	-	-	保留	0x00000010
5	-3	固定	Ecall-M	机器模式回调中断	0x00000014
6-7	-	-	-	保留	0x00000018- 0x0000001C
8	-2	固定	Ecall-U	用户模式回调中断	0x00000020
9	-1	固定	BreakPoint	断点回调中断	0x00000024
10-11	-	-	-	保留	0x00000028- 0x0000002C
12	0	可编程	SysTick	系统定时器中断	0x00000030
13	-	-	-	保留	0x00000034
14	1	可编程	SW	软件中断	0x00000038
15	-	-	-	保留	0x0000003C
16	2	可编程	WWDG	窗口定时器中断	0x00000040
17	3	可编程	PVD	电源电压检测中断（EXTI）	0x00000044
18	4	可编程	TAMPER	侵入检测中断	0x00000048
19	5	可编程	RTC	实时时钟中断	0x0000004C
20	6	可编程	FLASH	闪存全局中断	0x00000050
21	7	可编程	RCC	复位和时钟控制中断	0x00000054
22	8	可编程	EXTI0	EXTI 线 0 中断	0x00000058
23	9	可编程	EXTI1	EXTI 线 1 中断	0x0000005C
24	10	可编程	EXTI2	EXTI 线 2 中断	0x00000060
25	11	可编程	EXTI3	EXTI 线 3 中断	0x00000064
26	12	可编程	EXTI4	EXTI 线 4 中断	0x00000068
27	13	可编程	DMA1_CH1	DMA1 通道 1 全局中断	0x0000006C
28	14	可编程	DMA1_CH2	DMA1 通道 2 全局中断	0x00000070
29	15	可编程	DMA1_CH3	DMA1 通道 3 全局中断	0x00000074
30	16	可编程	DMA1_CH4	DMA1 通道 4 全局中断	0x00000078
31	17	可编程	DMA1_CH5	DMA1 通道 5 全局中断	0x0000007C
32	18	可编程	DMA1_CH6	DMA1 通道 6 全局中断	0x00000080

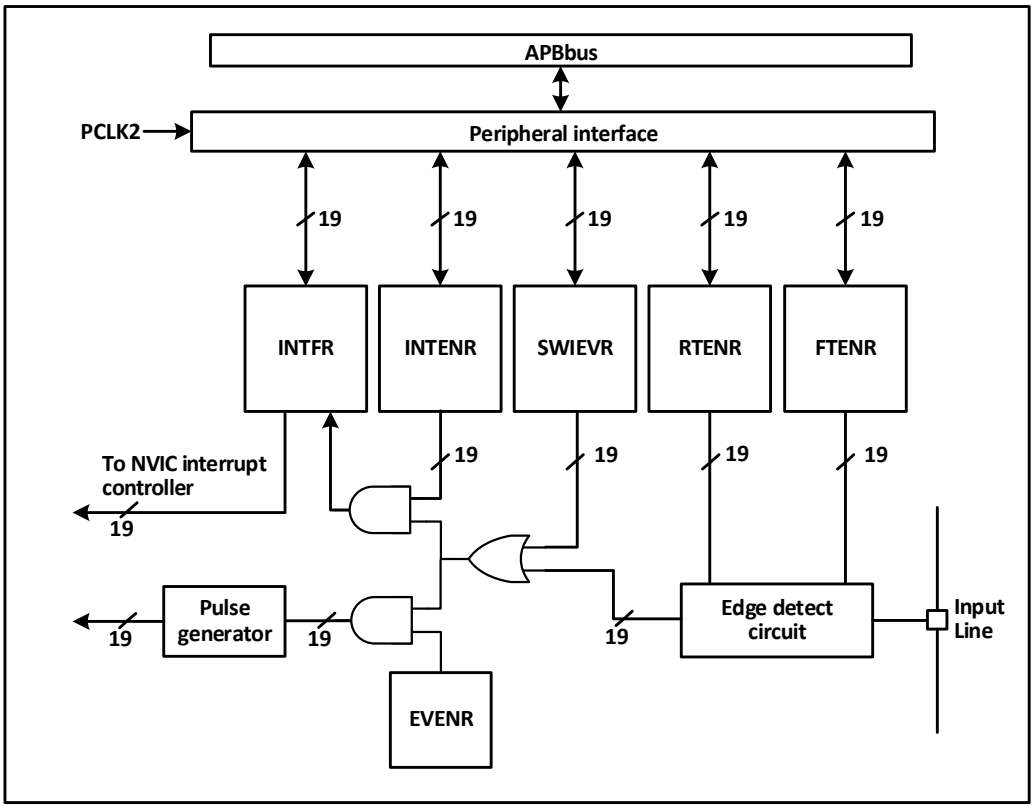
33	19	可编程	DMA1_CH7	DMA1 通道 7 全局中断	0x00000084
34	20	可编程	ADC1_2	ADC1 和 ADC2 全局中断	0x00000088
35	21	可编程	USB_HP 或 CAN1_TX	USB_HP 或 CAN1_TX 全局中断	0x0000008C
36	22	可编程	USB_LP 或 CAN1_RX0	USB_LP 或 CAN1_RX0 全局中断	0x00000090
37	23	可编程	CAN1_RX1	CAN1_RX1 全局中断	0x00000094
38	24	可编程	CAN1_SCE	CAN1_SCE 全局中断	0x00000098
39	25	可编程	EXTI9_5	EXTI 线[9:5]中断	0x0000009C
40	26	可编程	TIM1_BRK	TIM1 刹车中断	0x000000A0
41	27	可编程	TIM1_UP	TIM1 更新中断	0x000000A4
42	28	可编程	TIM1_TRG_COM	TIM1 触发和通信中断	0x000000A8
43	29	可编程	TIM1_CC	TIM1 捕获比较中断	0x000000AC
44	30	可编程	TIM2	TIM2 全局中断	0x000000B0
45	31	可编程	TIM3	TIM3 全局中断	0x000000B4
46	32	可编程	TIM4	TIM4 全局中断	0x000000B8
47	33	可编程	I2C1_EV	I^2C1 事件中断	0x000000BC
48	34	可编程	I2C1_ER	I^2C1 错误中断	0x000000C0
49	35	可编程	I2C2_EV	I^2C2 事件中断	0x000000C4
50	36	可编程	I2C2_ER	I^2C2 错误中断	0x000000C8
51	37	可编程	SPI1	SPI1 全局中断	0x000000CC
52	38	可编程	SPI2	SPI2 全局中断	0x000000D0
53	39	可编程	USART1	USART1 全局中断	0x000000D4
54	40	可编程	USART2	USART2 全局中断	0x000000D8
55	41	可编程	USART3	USART3 全局中断	0x000000DC
56	42	可编程	EXTI15_10	EXTI 线[15:10]中断	0x000000E0
57	43	可编程	RTCAlarm	RTC 闹钟中断（EXTI）	0x000000E4
58	44	可编程	USBWakeUp	USB 唤醒中断（EXTI）	0x000000E8
59	45	可编程	TIM8_BRK	TIM8 刹车中断	0x000000EC
60	46	可编程	TIM8_UP	TIM8 更新中断	0x000000F0
61	47	可编程	TIM8_TRG_COM	TIM8 触发和通信中断	0x000000F4
62	48	可编程	TIM8_CC	TIM8 捕获比较中断	0x000000F8
63	49	可编程	RNG	RNG 全局中断	0x000000FC
64	50	可编程	FSMC	FSMC 全局中断	0x00000100
65	51	可编程	SDIO	SDIO 全局中断	0x00000104
66	52	可编程	TIM5	TIM5 全局中断	0x00000108
67	53	可编程	SPI3	SPI3 全局中断	0x0000010C
68	54	可编程	UART4	UART4 全局中断	0x00000110
69	55	可编程	UART5	UART5 全局中断	0x00000114
70	56	可编程	TIM6	TIM6 全局中断	0x00000118
71	57	可编程	TIM7	TIM7 全局中断	0x0000011C
72	58	可编程	DMA2_CH1	DMA2 通道 1 全局中断	0x00000120
73	59	可编程	DMA2_CH2	DMA2 通道 2 全局中断	0x00000124
74	60	可编程	DMA2_CH3	DMA2 通道 3 全局中断	0x00000128
75	61	可编程	DMA2_CH4	DMA2 通道 4 全局中断	0x0000012C
76	62	可编程	DMA2_CH5	DMA2 通道 5 全局中断	0x00000130
77	63	可编程	ETH	ETH 全局中断	0x00000134

78	64	可编程	ETH_WKUP	ETH 唤醒中断	0x00000138
79	65	可编程	CAN2_TX	CAN2_TX 全局中断	0x0000013C
80	66	可编程	CAN2_RX0	CAN2_RX0 全局中断	0x00000140
81	67	可编程	CAN2_RX1	CAN2_RX1 全局中断	0x00000144
82	68	可编程	CAN2_SCE	CAN2_SCE 全局中断	0x00000148
83	69	可编程	OTG_FS	全速 OTG 中断	0x0000014C
84	70	可编程	USBHSWakeUp	高速 USB 唤醒中断	0x00000150
85	71	可编程	USBHS	高速 USB 全局中断	0x00000154
86	72	可编程	DVP	DVP 全局中断	0x00000158
87	73	可编程	UART6	UART6 全局中断	0x0000015C
88	74	可编程	UART7	UART7 全局中断	0x00000160
89	75	可编程	UART8	UART8 全局中断	0x00000164
90	76	可编程	TIM9_BRK	TIM9 刹车中断	0x00000168
91	77	可编程	TIM9_UP	TIM9 更新中断	0x0000016C
92	78	可编程	TIM9_TRG_COM	TIM9 触发和通信中断	0x00000170
93	79	可编程	TIM9_CC	TIM9 捕获比较中断	0x00000174
94	80	可编程	TIM10_BRK	TIM10 刹车中断	0x00000178
95	81	可编程	TIM10_UP	TIM10 更新中断	0x0000017C
96	82	可编程	TIM10_TRG_COM	TIM10 触发和通信中断	0x00000180
97	83	可编程	TIM10_CC	TIM10 捕获比较中断	0x00000184
98	84	可编程	DMA2_CH6	DMA2 通道 6 全局中断	0x00000188
99	85	可编程	DMA2_CH7	DMA2 通道 7 全局中断	0x0000018C
100	86	可编程	DMA2_CH8	DMA2 通道 8 全局中断	0x00000190
101	87	可编程	DMA2_CH9	DMA2 通道 9 全局中断	0x00000194
102	88	可编程	DMA2_CH10	DMA2 通道 10 全局中断	0x00000198
103	89	可编程	DMA2_CH11	DMA2 通道 11 全局中断	0x0000019C

9.4 外部中断和事件控制器（EXTI）

9.4.1 概述

图 9-1 外部中断(EXTI)接口框图



由图 9-1 可以看出，外部中断的触发源既可以是软件中断（SWIEVR）也可以是实际的外部中断通道，外部中断通道的信号会先经过边沿检测电路（edge detect circuit）的筛选。只要产生软件中断或外部中断信号其一，就会通过图中的或门电路输出给事件使能和中断使能两个与门电路，只要有中断被使能或事件被使能，就会产生中断或事件。EXTI 的六个寄存器由处理器通过 APB2 接口访问。

9.4.2 唤醒事件说明

系统可以通过唤醒事件来唤醒由 WFE 指令引起的睡眠模式。唤醒事件通过以下两种配置产生：

- 在外设的寄存器里使能一个中断，但不在内核的 NVIC 或 PFIC 里使能这个中断，同时在内核里使能 SEVONPEND 位。体现在 EXTI 中，就是使能 EXTI 中断，但不在 NVIC 或 PFIC 中使能 EXTI 中断，同时使能 SEVONPEND 位。当 CPU 从 WFE 中唤醒后，需要清除 EXTI 的中断标志位和 NVIC 或 PFIC 挂起位。
- 使能一个 EXTI 通道为事件通道，CPU 从 WFE 唤醒后无需清除中断标志位和 NVIC 或 PFIC 挂起位的操作。

9.4.3 说明

使用外部中断需要配置相应外部中断通道，即选择相应触发沿，使能相应中断。当外部中断通道上出现了设定的触发沿时，将产生一个中断请求，对应的中断标志位也会被置位。对标志位写 1 可以清除该标志位。

使用外部硬件中断步骤：

1） 配置 GPIO 操作；

2） 配置对应的外部中断通道的中断使能位（EXTI_INTENR）；
3） 配置触发沿（EXTI_RTENR 或 EXTI_FTENR），选择上升沿触发、下降沿触发或双边沿触发；
4） 在内核的 NVIC/PFIC 中配置 EXTI 中断，以保证其可以正确响应。

使用外部硬件事件步骤：
1） 配置 GPIO 操作；
2） 配置对应的外部中断通道的事件使能位（EXTI_EVENR）；
3） 配置触发沿（EXTI_RTENR 或 EXTI_FTENR），选择上升沿触发、下降沿触发或双边沿触发。

使用软件中断/事件步骤：
1） 使能外部中断（EXTI_INTENR）或外部事件（EXTI_EVENR）；
2） 如果使用中断服务函数，需要设置内核的 NVIC 或 PFIC 里 EXTI 中断；
3） 设置软件中断触发（EXTI_SWIEVR），即会产生中断。

9.4.4 外部事件映射

表 9-3 EXTI 中断映射

外部中断/事件线路	映射事件描述
EXTI0～EXTI15	Px0～Px15（x=A/B/C/D/E），任何一个 IO 口都可以启用外部中断/事件功能，由 AFIO_EXTICRx 寄存器配置。
EXTI16	PVD 事件：超出电压监控阀值
EXTI17	RTC 闹钟事件
EXTI18	USBD/USBFSOTG 唤醒事件（*适用 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C*） USBD 唤醒事件（其余芯片型号）
EXTI19	ETH 唤醒事件
EXTI20	USBHS 唤醒事件（*适用 CH32F20x_D8C、CH32V30x_D8C*） USBFS 唤醒事件（*适用其余芯片型号*）
EXTI21	内部 32K 校准唤醒事件（*适用于 CH32V20x_D8、CH32V20x_D8W、CH32F20x_D8W*）

9.5 寄存器描述

9.5.1 EXTI 寄存器描述

表 9-4 EXTI 相关寄存器列表

名称	访问地址	描述	复位值
R32_EXTI_INTENR	0x40010400	中断使能寄存器	0x00000000
R32_EXTI_EVENR	0x40010404	事件使能寄存器	0x00000000
R32_EXTI_RTENR	0x40010408	上升沿触发使能寄存器	0x00000000
R32_EXTI_FTENR	0x4001040C	下降沿触发使能寄存器	0x00000000
R32_EXTI_SWIEVR	0x40010410	软中断事件寄存器	0x00000000
R32_EXTI_INTFR	0x40010414	中断标志位寄存器	0x0000XXXX

9.5.1.1 中断使能寄存器（EXTI_INTENR）

偏移地址：0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											MR20	MR19	MR18	MR17	MR16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0

位	名称	访问	描述	复位值
[31:19]	Reserved	RO	保留。	0
[18:0]	MRx	RW	使能外部中断通道 x 的中断请求信号： 1：使能此通道的中断； 0：屏蔽此通道的中断。	0

9.5.1.2 事件使能寄存器（EXTI_EVENR）

偏移地址：0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											MR20	MR19	MR18	MR17	MR16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:0]	MRx	RW	使能外部中断通道 x 的事件请求信号： 1：使能此通道的事件； 0：屏蔽此通道的事件。	0

9.5.1.3 上升沿触发使能寄存器（EXTI_RTENR）

偏移地址：0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TR20	TR19	TR18	TR17	TR16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0

位	名称	访问	描述	复位值
[31:20]	Reserved	RO	保留。	0
[21:0]	TRx	RW	使能外部中断通道 x 的上升沿触发： 1：使能此通道的上升沿触发； 0：禁止此通道的上升沿触发。	0

9.5.1.4 下降沿触发使能寄存器（EXTI_FTENR）

偏移地址：0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TR20	TR19	TR18	TR17	TR16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留	0
[20:0]	TRx	RW	使能外部中断通道 x 的下降沿触发： 0：禁止此通道的下降沿触发； 1：使能此通道的下降沿触发。	0

9.5.1.5 软中断事件寄存器（EXTI_SWIEVR）

偏移地址：0x10

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

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

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:0]	SWIERx	RW	在相对应的外部触发中断通道上设置一个软件中断。这里置位会使中断标志位（EXTI_INTFR）对应位置位，如果中断使能（EXTI_INTENR）或事件使能（EXTI_EVENR）开启，那么就会产生中断或事件。	0

9.5.1.6 中断标志位寄存器（EXTI_INTFR）

偏移地址：0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											IF20	IF19	IF18	IF17	IF16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF15	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7	IF6	IF5	IF4	IF3	IF2	IF1	IF0

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:0]	IFx	W1	中断标志位，该位置位标志表示发生了对应的外部中断。写 1 可以清除此位。	X

9.5.2 PFIC 寄存器描述

表 9-5 PFIC 相关寄存器列表

名称	访问地址	描述	复位值
R32_PFIC_ISR1	0xE000E000	PFIC 中断使能状态寄存器 1	0x0000000C
R32_PFIC_ISR2	0xE000E004	PFIC 中断使能状态寄存器 2	0x00000000

R32_PFIC_ISR3	0xE000E008	PFIC 中断使能状态寄存器 3	0x00000000
R32_PFIC_ISR4	0xE000E00C	PFIC 中断使能状态寄存器 4	0x00000000
R32_PFIC_IPR1	0xE000E020	PFIC 中断挂起状态寄存器 1	0x00000000
R32_PFIC_IPR2	0xE000E024	PFIC 中断挂起状态寄存器 2	0x00000000
R32_PFIC_IPR3	0xE000E028	PFIC 中断挂起状态寄存器 3	0x00000000
R32_PFIC_IPR4	0xE000E02C	PFIC 中断挂起状态寄存器 4	0x00000000
R32_PFIC_ITHRESDR	0xE000E040	PFIC 中断优先级阈值配置寄存器	0x00000000
R32_PFIC_CFGR	0xE000E048	PFIC 中断配置寄存器	0x00000000
R32_PFIC_GISR	0xE000E04C	PFIC 中断全局状态寄存器	0x00000000
R32_PFIC_VTFIDR	0xE000E050	PFIC VTF 中断 ID 配置寄存器	0x00000000
R32_PFIC_VTFADDRR0	0xE000E060	PFIC VTF 中断 0 偏移地址寄存器	0x00000000
R32_PFIC_VTFADDRR1	0xE000E064	PFIC VTF 中断 1 偏移地址寄存器	0x00000000
R32_PFIC_VTFADDRR2	0xE000E068	PFIC VTF 中断 2 偏移地址寄存器	0x00000000
R32_PFIC_VTFADDRR3	0xE000E06C	PFIC VTF 中断 3 偏移地址寄存器	0x00000000
R32_PFIC_IENR1	0xE000E100	PFIC 中断使能设置寄存器 1	0x00000000
R32_PFIC_IENR2	0xE000E104	PFIC 中断使能设置寄存器 2	0x00000000
R32_PFIC_IENR3	0xE000E108	PFIC 中断使能设置寄存器 3	0x00000000
R32_PFIC_IENR4	0xE000E10C	PFIC 中断使能设置寄存器 4	0x00000000
R32_PFIC_IRER1	0xE000E180	PFIC 中断使能清除寄存器 1	0x00000000
R32_PFIC_IRER2	0xE000E184	PFIC 中断使能清除寄存器 2	0x00000000
R32_PFIC_IRER3	0xE000E188	PFIC 中断使能清除寄存器 3	0x00000000
R32_PFIC_IRER4	0xE000E18C	PFIC 中断使能清除寄存器 4	0x00000000
R32_PFIC_IPSR1	0xE000E200	PFIC 中断挂起设置寄存器 1	0x00000000
R32_PFIC_IPSR2	0xE000E204	PFIC 中断挂起设置寄存器 2	0x00000000
R32_PFIC_IPSR3	0xE000E208	PFIC 中断挂起设置寄存器 3	0x00000000
R32_PFIC_IPSR4	0xE000E20C	PFIC 中断挂起设置寄存器 4	0x00000000
R32_PFIC_IPRR1	0xE000E280	PFIC 中断挂起清除寄存器 1	0x00000000
R32_PFIC_IPRR2	0xE000E284	PFIC 中断挂起清除寄存器 2	0x00000000
R32_PFIC_IPRR3	0xE000E288	PFIC 中断挂起清除寄存器 3	0x00000000
R32_PFIC_IPRR4	0xE000E28C	PFIC 中断挂起清除寄存器 4	0x00000000
R32_PFIC_IACTR1	0xE000E300	PFIC 中断激活状态寄存器 1	0x00000000
R32_PFIC_IACTR2	0xE000E304	PFIC 中断激活状态寄存器 2	0x00000000
R32_PFIC_IACTR3	0xE000E308	PFIC 中断激活状态寄存器 3	0x00000000
R32_PFIC_IACTR4	0xE000E30C	PFIC 中断激活状态寄存器 4	0x00000000
R32_PFIC_IPRIORx	0xE000E400	PFIC 中断优先级配置寄存器	0x00000000
R32_PFIC_SCTLR	0xE000ED10	PFIC 系统控制寄存器	0x00000000

注：1. NMI、EXC、ECALL-M、ECALL-U、BREAKPOINT 中断默认总是使能。

2. ECALL-M、ECALL-U、BREAKPOINT 均为 EXC 的一种情况，状态由 EXC 的状态位 bit3 表示。

3. NMI、EXC 支持中断挂起清除和设置操作，不支持中断使能清除和设置操作。

4. ECALL-M、ECALL-U、BREAKPOINT 不支持中断挂起清除和设置、中断使能清除和设置操作。

9.5.2.1 PFIC 中断使能状态寄存器 1（PFIC_ISR1）

偏移地址：0x00

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

INTS[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–	INTS14	–	INTS12	Reserved								1	1	Reserved	

位	名称	访问	描述	复位值
[31:12]	INTSTA	RO	31#及以下中断当前使能状态。 1：当前编号中断已使能； 0：当前编号中断未启用。	Ch

9.5.2.2 PFIC 中断使能状态寄存器 2（PFIC_ISR2）

偏移地址：0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTS[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTS[47:32]															

位	名称	访问	描述	复位值
[31:0]	INTENSTA	RO	32#-63#中断当前使能状态。 1：当前编号中断已使能； 0：当前编号中断未启用。	0

9.5.2.3 PFIC 中断使能状态寄存器 3（PFIC_ISR4）

偏移地址：0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTS[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTS[79:64]															

位	名称	访问	描述	复位值
[31:0]	INTENSTA	RO	64#-95#中断当前使能状态。 1：当前编号中断已使能； 0：当前编号中断未启用。	0

9.5.2.4 PFIC 中断使能状态寄存器 4（PFIC_ISR4）

偏移地址：0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								INTS[103:96]							

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留	0

[7:0]	INTENSTA	RO	96#-103#中断当前使能状态。 1：当前编号中断已使能； 0：当前编号中断未启用。	0

9.5.2.5 PFIC 中断挂起状态寄存器 1（PFIC_IPR1）

偏移地址：0x20

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	IPS14	-	IPS12	Reserved								IPS3	IPS2	Reserved	

位	名称	访问	描述	复位值
[31:0]	PENDSTA	RO	31#及以下中断当前挂起状态。 1：当前编号中断已挂起； 0：当前编号中断未挂起。	0

9.5.2.6 PFIC 中断挂起状态寄存器 2（PFIC_IPR2）

偏移地址：0x24

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPS[47:32]															

位	名称	访问	描述	复位值
[31:0]	PENDSTA	RO	32#-63#中断当前挂起状态。 1：当前编号中断已挂起； 0：当前编号中断未挂起。	0

9.5.2.7 PFIC 中断挂起状态寄存器 3（PFIC_IPR3）

偏移地址：0x28

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

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

位	名称	访问	描述	复位值
[31:0]	PENDSTA	RO	64#-95#中断当前挂起状态。 1：当前编号中断已挂起； 0：当前编号中断未挂起。	0

9.5.2.8 PFIC 中断挂起状态寄存器 4（PFIC_IPR4）

偏移地址：0x2C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IPS[103:96]							

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:0]	PENDSTA	RO	96#-103#中断当前挂起状态。 1：当前编号中断已挂起； 0：当前编号中断未挂起。	0

9.5.2.9 PFIC 中断优先级阈值配置寄存器（PFIC_ITHRESDR）

偏移地址：0x40

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

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:0]	THRESHOLD	RW	中断优先级阈值设置值。 低于当前设置值的中断优先级值，当挂起时不执行中断服务；此寄存器为 0 时表示阈值寄存器功能无效。 [7:4]：优先级阈值。 [3:0]：保留，固定为 0，写无效。	0

9.5.2.10 PFIC 中断配置寄存器（PFIC_CFGR）

偏移地址：0x48

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

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

位	名称	访问	描述	复位值
[31:16]	KEYCODE	WO	对应不同的目标控制位，需要同步写入相应的安全访问标识数据才能修改，读出数据固定为 0。 KEY1 = 0xFA05； KEY2 = 0xBCAF； KEY3 = 0xBEEF。	0
[15:8]	Reserved	RO	保留。	0
7	SYSRESET	WO	系统复位（同步写入 KEY3）。自动清 0。 写 1 有效，写 0 无效。 *注：与 PFIC_SCTLR 寄存器 SYSRESET 位作用相同。*	0

[6:0]	Reserved	RO	保留。	0

9.5.2.11 PFIC 中断全局状态寄存器（PFIC_GISR）

偏移地址：0x4C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						GPENDSTA	GACTSTA	NESTSTA[7:0]							

位	名称	访问	描述	复位值
[31:10]	Reserved	RO	保留。	0
9	GPENDSTA	RO	当前是否有中断处于挂起： 1：有；　　0：没有。	0
8	GACTSTA	RO	当前是否有中断被执行： 1：有；　　0：没有。	0
[7:0]	NESTSTA	RO	当前中断嵌套状态，目前最大支持 8 级嵌套，硬件压栈深度最大为 3 级，若设置嵌套深度大于 3 级，则应配置低三级中断为硬件压栈，其余高优先级使用软件压栈。 0xFF：第 8 级中断中； 0x7F：第 7 级中断中； 0x3F：第 6 级中断中； 0x1F：第 5 级中断中； 0x0F：第 4 级中断中； 0x07：第 3 级中断中； 0x03：第 2 级中断中； 0x01：第 1 级中断中； 0x00：没有中断发生； 其他：不可能情况。	0

9.5.2.12 PFIC VTF 中断 ID 配置寄存器（PFIC_VTFIDR）

偏移地址：0x50

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VTFID3								VTFID2							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTFID1								VTFID0							

位	名称	访问	描述	复位值
[31:24]	VTFID3	RW	配置 VTF 中断 3 的中断编号。	0
[23:16]	VTFID2	RW	配置 VTF 中断 2 的中断编号。	0
[15:8]	VTFID1	RW	配置 VTF 中断 1 的中断编号。	0
[7:0]	VTFID0	RW	配置 VTF 中断 0 的中断编号。	0

9.5.2.13 PFIC VTF 中断 0 地址寄存器（PFIC_VTFADDRR0）

偏移地址：0x60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR0[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0[15:1]															VTF0EN

位	名称	访问	描述	复位值
[31:1]	ADDR0	RW	VTF 中断 0 服务程序地址 bit[31:1]，bit0 为 0。	0
0	VTF0EN	RW	VTF 中断 0 使能位： 1：启用 VTF 中断 0 通道；0：关闭。	0

9.5.2.14 PFIC VTF 中断 1 地址寄存器（PFIC_VTFADDRR1）

偏移地址：0x64

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1[15:1]															VTF1EN

位	名称	访问	描述	复位值
[31:1]	ADDR1	RW	VTF 中断 1 服务程序地址 bit[31:1]，bit0 为 0。	0
0	VTF1EN	RW	VTF 中断 1 使能位： 1：启用 VTF 中断 1 通道；0：关闭。	0

9.5.2.15 PFIC VTF 中断 2 地址寄存器（PFIC_VTFADDRR2）

偏移地址：0x68

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR2[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2[15:1]															VTF2EN

位	名称	访问	描述	复位值
[31:1]	ADDR2	RW	VTF 中断 2 服务程序地址 bit[31:1]，bit0 为 0。	0
0	VTF2EN	RW	VTF 中断 2 使能位： 1：启用 VTF 中断 2 通道；0：关闭。	0

9.5.2.16 PFIC VTF 中断 3 地址寄存器（PFIC_VTFADDRR3）

偏移地址：0x6C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR3[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3[15:1]															VTF3EN

位	名称	访问	描述	复位值
[31:1]	ADDR3	RW	VTF 中断 3 服务程序地址 bit[31:1]，bit0 为 0。	0
0	VTF3EN	RW	VTF 中断 3 使能位： 1：启用 VTF 中断 3 通道；0：关闭。	0

9.5.2.17 PFIC 中断使能设置寄存器 1（PFIC_IENR1）

偏移地址：0x100

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–	INTEN14	–	INTEN12	Reserved											

位	名称	访问	描述	复位值
[31:12]	INTEN	WO	31#及以下中断使能控制。 1：当前编号中断使能； 0：无影响。	0
[11:0]	Reserved	RO	保留。	0

9.5.2.18 PFIC 中断使能设置寄存器 2（PFIC_IENR2）

偏移地址：0x104

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN[47:32]															

位	名称	访问	描述	复位值
[31:0]	INTEN	WO	32#-63#中断使能控制。 1：当前编号中断使能； 0：无影响。	0

9.5.2.19 PFIC 中断使能设置寄存器 3（PFIC_IENR3）

偏移地址：0x108

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

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

INTEN[79:64]

位	名称	访问	描述	复位值
[31:0]	INTEN	WO	64#-95#中断使能控制。 1：当前编号中断使能； 0：无影响。	0

9.5.2.20 PFIC 中断使能设置寄存器 4（PFIC_IENR4）

偏移地址：0x10C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								INTEN[103:96]							

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:0]	INTEN	WO	96#-103#中断使能控制。 1：当前编号中断使能； 0：无影响。	0

9.5.2.21 PFIC 中断使能清除寄存器 1（PFIC_IRER1）

偏移地址：0x180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTRESET[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	INT RESET14	-	INT RESET12	Reserved											

位	名称	访问	描述	复位值
[31:12]	INTRESET	WO	31#及以下中断关闭控制。 1：当前编号中断关闭； 0：无影响。	0
[11:0]	Reserved	RO	保留。	0

9.5.2.22 PFIC 中断使能清除寄存器 2（PFIC_IRER2）

偏移地址：0x184

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTRESET[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRESET[47:32]															

位	名称	访问	描述	复位值
[31:0]	INTRESET	WO	32#-63#中断关闭控制。 1：当前编号中断关闭； 0：无影响。	0

9.5.2.23 PFIC 中断使能清除寄存器 3（PFIC_IRER3）

偏移地址：0x188

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTRESET[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRESET [79:64]															

位	名称	访问	描述	复位值
[31:0]	INTRESET	WO	64#-95#中断关闭控制。 1：当前编号中断关闭； 0：无影响。	0

9.5.2.24 PFIC 中断使能清除寄存器 4（PFIC_IRER4）

偏移地址：0x18C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								INTRESET [103:96]							

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:0]	INTRESET	WO	96#-103#中断关闭控制。 1：当前编号中断关闭； 0：无影响。	0

9.5.2.25 PFIC 中断挂起设置寄存器 1（PFIC_IPSR1）

偏移地址：0x200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSET[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	PEND SET14	-	PEND SET12	Reserved								PEND SET3	PEND SET2	Reserved	

位	名称	访问	描述	复位值
[31:12]	PENDSET	WO	12#-31#中断挂起设置，13#和 15#保留。 1：当前编号中断挂起； 0：无影响。	0

[11:4]	Reserved	RO	保留。	0
[3:2]	PENDSET	WO	2#-3#中断挂起设置。 1：当前编号中断挂起； 0：无影响。	0
[1:0]	Reserved	RO	保留。	0

9.5.2.26 PFIC 中断挂起设置寄存器 2（PFIC_IPSR2）

偏移地址：0x204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSET[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSET[47:32]															

位	名称	访问	描述	复位值
[31:0]	PENDSET	WO	32#-63#中断挂起设置。 1：当前编号中断挂起； 0：无影响。	0

9.5.2.27 PFIC 中断挂起设置寄存器 3（PFIC_IPSR3）

偏移地址：0x208

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSET[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSET[79:64]															

位	名称	访问	描述	复位值
[31:0]	PENDSET	WO	64#-95#中断挂起设置。 1：当前编号中断挂起； 0：无影响。	0

9.5.2.28 PFIC 中断挂起设置寄存器 4（PFIC_IPSR4）

偏移地址：0x208

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PENDSET [103:96]							

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:0]	PENDSET	WO	96#-103#中断挂起设置。 1：当前编号中断挂起； 0：无影响。	0

9.5.2.29 PFIC 中断挂起清除寄存器 1（PFIC_IPRR1）

偏移地址：0x280

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–	PEND RESET14	–	PEND RESET12	Reserved								PEND RESET3	PEND RESET2	Reserved	

位	名称	访问	描述	复位值
[31:12]	PENDRESET	WO	12#-31#中断挂起清除，13#和 15#保留。 1：当前编号中断清除挂起状态； 0：无影响。	0
[11:4]	Reserved	RO	保留。	0
[3:2]	PENDRESET	WO	2#-3#中断挂起清除。 1：当前编号中断清除挂起状态； 0：无影响。	0
[1:0]	Reserved	RO	保留。	0

9.5.2.30 PFIC 中断挂起清除寄存器 2（PFIC_IPRR2）

偏移地址：0x284

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDRESET[47:32]															

位	名称	访问	描述	复位值
[31:0]	PENDRESET	WO	32#-63#中断挂起清除。 1：当前编号中断清除挂起状态； 0：无影响。	0

9.5.2.31 PFIC 中断挂起清除寄存器 3（PFIC_IPRR3）

偏移地址：0x288

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

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

位	名称	访问	描述	复位值
[31:0]	PENDRESET	WO	64#-95#中断挂起清除。 1：当前编号中断清除挂起状态； 0：无影响。	0

9.5.2.32 PFIC 中断挂起清除寄存器 4（PFIC_IPRR4）

偏移地址：0x28C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PENDRESET [103:96]							

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:0]	PENDSET	WO	96#-103#中断挂起清除。 1：当前编号中断清除挂起状态； 0：无影响	0

9.5.2.33 PFIC 中断激活状态寄存器 1（PFIC_IACTR1）

偏移地址：0x300

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	IACTS14	-	IACTS12	Reserved								IACTS3	IACTS2	Reserved	

位	名称	访问	描述	复位值
[31:12]	IACTS	RO	12#-31#中断执行状态，13#和 15#保留。 1：当前编号中断执行中； 0：当前编号中断没执行。	0
[11:4]	Reserved	RO	保留。	0
[3:2]	IACTS	RO	2#-3#中断执行状态。 1：当前编号中断执行中； 0：当前编号中断没执行。	0
[1:0]	Reserved	RO	保留。	0

9.5.2.34 PFIC 中断激活状态寄存器 2（PFIC_IACTR2）

偏移地址：0x304

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IACTS[47:32]															

位	名称	访问	描述	复位值
[31:0]	IACTS	RO	32#-63#中断执行状态。 1：当前编号中断执行中； 0：当前编号中断没执行。	0

9.5.2.35 PFIC 中断激活状态寄存器 3（PFIC_IACTR3）

偏移地址：0x308

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IACTS[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IACTS[79:64]															

位	名称	访问	描述	复位值
[31:0]	IACTS	RO	64#-95#中断执行状态。 1：当前编号中断执行中； 0：当前编号中断没执行。	0

9.5.2.36 PFIC 中断激活状态寄存器 4（PFIC_IACTR4）

偏移地址：0x308

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IACTS [103:96]							

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:0]	IACTS	WO	96#-103#中断执行状态。 1：当前编号中断执行中； 0：当前编号中断没执行	0

9.5.2.37 PFIC 中断优先级配置寄存器（PFIC_IPRIORx）（x=0-63）

偏移地址：0x400 - 0x4FF

控制器支持 256 个中断（0-255），每个中断使用 8bit 来设置控制优先级。

	31 24	23 16	15 8	7 0
IPRIOR63	PRIO_255	PRIO_254	PRIO_253	PRIO_252
…	…	…	…	…
IPRIORx	PRIO_(4x+3)	PRIO_(4x+2)	PRIO_(4x+1)	PRIO_(4x)
…	…	…	…	…
IPRIOR0	PRIO_3	PRIO_2	PRIO_1	PRIO_0

位	名称	访问	描述	复位值
[2047:2040]	IP_255	RW	同 IP_0 描述。	0
…	…	…	…	…
[31:24]	IP_3	RW	同 IP_0 描述。	0
[23:16]	IP_2	RW	同 IP_0 描述。	0

[15:8]	IP_1	RW	同 IP_0 描述。	0
[7:0]	IP_0	RW	编号 0 中断优先级配置： [7:4]：优先级控制位。 若配置无嵌套，无抢占位； 若配置 2 级嵌套，bit7 为抢占位； 若配置 4 级嵌套，bit7-bit6 为抢占位； 若配置 8 级嵌套，bit7-bit5 为抢占位； 优先级数值越小则优先级越高，同一抢占优先级中断若同时挂起，优先执行优先级高的中断。 [3:0]：保留，固定为 0，写无效。	0

9.5.2.38 PFIC 系统控制寄存器（PFIC_SCTLR）

偏移地址：0xD10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										SET EVENT	SEV ONPEND	WFITO WFE	SLEEP DEEP	SLEEP ONEXIT	Reser ved

位	名称	访问	描述	复位值
31	SYSRESET	WO	系统复位，自动清 0。写 1 有效，写 0 无效，与 PFIC_CFGR 寄存器相同效果	0
[30:6]	Reserved	RO	保留。	0
5	SETEVENT	WO	设置事件，可以唤醒 WFE 的情况。	0
4	SEVONPEND	RW	当发生事件或者中断挂起状态时，可以从 WFE 指令后唤醒系统,如果未执行 WFE 指令，将在下次执行该指令后立即唤醒系统。 1：启用的事件和所有中断（包括未开启中断）都能唤醒系统； 0：只有启用的事件和启用的中断可以唤醒系统。	0
3	WFITOWFE	RW	将 WFI 指令当成是 WFE 执行。 1：将之后的 WFI 指令当做 WFE 指令； 0：无作用。	0
2	SLEEPDEEP	RW	控制系统的低功耗模式： 1：deepsleep　　0：sleep	0
1	SLEEPONEXIT	RW	控制离开中断服务程序后，系统状态： 1：系统进入低功耗模式； 0：系统进入主程序。	0
0	Reserved	RO	保留。	0

9.5.3 专用 CSR 寄存器

RISC-V 架构中定义了一些控制和状态寄存器（Control and Status Register,CSR），用于配置或标识或记录运行状态。CSR 寄存器属于内核内部的寄存器，使用专用的 12 位地址空间。CH32V20x 和 CH32V30x 系列芯片除了 RISC-V 特权架构文档中定义的标准寄存器外，还增加了一些厂商自定义寄存器，需要使用 csr 指令进行访问。

注：此类寄存器标注为“MRW,MRO,MRW1”属性的需要系统在机器模式下才能访问。

9.5.3.1 中断系统控制寄存器（INTSYSCR）

CSR 地址：0x804

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

位	名称	访问	描述	复位值
[31:16]	Reserved	MRO	保留。	0
[15:8]	PMTSTA	MRO	抢占位状态指示： 0x00：优先级配置位中无抢占位，不发生中断嵌套； 0x80：优先级配置位中最高位为抢占位，2 级中断嵌套； 0xC0：优先级配置位中高 2 位为抢占位，4 级中断嵌套； 0xE0：优先级配置位中高 3 位为抢占位，8 级中断嵌套。	0
[7:6]	Reserved	MRO	保留。	0
5	GIHWSTKNEN	MRW1	全局中断和硬件压栈关闭使能。 *注：该位常使用于实时操作系统中，中断切换上下文时，置位该位，可关闭全局中断和硬件压栈出栈，当上下文切换完成，执行完中断返回后，硬件自动清除该位。*	0
4	HWSTKOVEN	MRW	硬件压栈溢出后中断使能： 0：硬件压栈溢出后，关闭全局中断； 1：硬件压栈溢出后，中断仍可执行。 *注：硬件压栈深度为 3 级，当配置嵌套等级大于 3 级，若该位设置 1，需要将低优先级的三级中断配置为硬件压栈，高优先级配置为软件压栈。*	0
[3:2]	PMTCFG	MRW	中断嵌套深度配置： 0b00：无嵌套，抢占位个数为 0； 0b01：2 级嵌套，抢占位个数为 1； 0b10：4 级嵌套，抢占位个数为 2； 0b11：8 级嵌套，抢占位个数为 3。	0

1	INESTEN	MRW	中断嵌套使能： 0：中断嵌套功能关闭； 1：中断嵌套功能使能。	0
0	HWSTKEN	MRW	硬件压栈使能： 0：硬件压栈功能关闭； 1：硬件压栈功能使能。	0

9.5.3.2 异常入口基地址寄存器（MTVEC）

CSR 地址：0x305

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASEADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADDR[15:2]														MODE1	MODE0

位	名称	访问	描述	复位值
[31:2]	BASEADDR[31:2]	MRW	中断向量表基地址。	0
1	MODE1	MRW	中断向量表识别模式： 0：按跳转指令识别，有限范围，支持非跳指令； 1：按绝对地址识别，支持全范围，但必须跳转。	0
0	MODE0	MRW	中断或异常入口地址模式选择： 0：使用统一入口地址； 1：根据中断编号*4 进行地址偏移。	0

9.5.4 物理内存保护单元（PMP）

为了提高系统安全，RISC-V 的架构中定义了一套物理地址访问限制，可以为区域内物理内存设置其读、写、执行属性，区域长度最小 4 字节保护。PMP 单元在用户模式下一直生效，在机器模式下可选生效，如果违背了当前内存限制，将会产生系统异常中断（EXC）。

PMP 单元包含 4 组 8-bit 的配置寄存器（32bit）和 4 组地址寄存器，需要使用 csr 指令进行访问，并且在机器模式下进行。

9.5.4.1 PMP 配置寄存器（PMPCFG0）

CSR 地址：0x3A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
pmp3cfg								pmp2cfg							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
pmp1cfg								pmp0cfg							

位	名称	访问	描述	复位值
[31:24]	pmp3cfg	MRW	见 pmp0cfg。	0
[23:16]	pmp2cfg	MRW	见 pmp0cfg。	0
[15:8]	pmp1cfg	MRW	见 pmp0cfg。	0

[7:0]	pmp0cfg	MRW	位	名称	描述	0
			7	L	锁定使能，机器模式下可解锁 0：不锁定； 1：锁定相关寄存器。	
			[6:5]	-	保留。	
			[4:3]	A	地址对齐及保护区域范围选择。	
			2	X	可执行属性。	
			1	W	可写入属性。	
			0	R	可读出属性。	

其中，地址对齐及保护区域范围选择，对于 A_ADDR≤region<B_ADDR 区域进行内存保护（要求 A_ADDR 和 B_ADDR 均为 4 字节对齐）：

1、如果 B_ADDR - A_ADDR == 2^2，则采用 NA4 方式；

2、如果 B_ADDR - A_ADDR == $2^{(G+2)}$，G≥1，且 A_ADDR 为 $2^{(G+2)}$ 对齐则采用 NAPOT 方式；

3、否则采用 TOR 方式。

A 值	名称	描述
00b	OFF	没有区域要保护
01b	TOR	顶端对齐区域保护： pmp0cfg 下，0≤ region <pmpaddr0； pmp1cfg 下，pmpaddr0≤ region <pmpaddr1； pmp2cfg 下，pmpaddr1≤ region <pmpaddr2； pmp3cfg 下，pmpaddr2≤ region <pmpaddr3。 $pmpaddr_{i-1}$ = A_ADDR >> 2； $pmpaddr_i$ = B_ADDR >> 2。
10b	NA4	固定 4 字节区域保护。 pmp0cfg～pmp3cfg 对应 pmpaddr0～pmpaddr3 作为起始地址。 $pmpaddr_i$ = A_ADDR >> 2。
11b	NAPOT	保护 $2^{(G+2)}$ 区域，G≥1，此时 A_ADDR 为 $2^{(G+2)}$ 对齐。 $pmpaddr_i$ = （(A_ADDR | ($2^{(G+2)}$-1)) & ~(1<<G+1)) >> 2。

9.5.4.2 PMP 地址 0 寄存器（PMPADDR0）

CSR 地址：0x3B0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR0[33:18]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0[17:2]															

位	名称	访问	描述	复位值
[31:0]	ADDR0	MRW	PMP 设置地址 0 的 bit[33:2],实际高 2 位未用。	0

9.5.4.3 PMP 地址 1 寄存器（PMPADDR1）

CSR 地址：0x3B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1[33:18]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1[17:2]															

位	名称	访问	描述	复位值
[31:0]	ADDR1	MRW	PMP 设置地址 1 的 bit[33:2],实际高 2 位未用。	0

9.5.4.4 PMP 地址 2 寄存器（PMPADDR2）

CSR 地址：0x3B2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR2[33:18]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2[17:2]															

位	名称	访问	描述	复位值
[31:0]	ADDR2	MRW	PMP 设置地址 2 的 bit[33:2],实际高 2 位未用。	0

9.5.4.5 PMP 地址 3 寄存器（PMPADDR3）

CSR 地址：0x3B3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR3[33:18]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3[17:2]															

位	名称	访问	描述	复位值
[31:0]	ADDR3	MRW	PMP 设置地址 3 的 bit[33:2],实际高 2 位未用。	0

9.5.5 STK 寄存器描述

表 9-6 STK 相关寄存器列表

名称	访问地址	描述	复位值
R32_STK_CTLR	0xE000F000	系统计数控制寄存器	0x00000000
R32_STK_SR	0xE000F004	系统计数状态寄存器	0x00000000
R32_STK_CNTL	0xE000F008	系统计数器低位寄存器	0x00000000
R32_STK_CNTH	0xE000F00C	系统计数器高位寄存器	0x00000000
R32_STK_CMPLR	0xE000F010	计数比较低位寄存器	0x00000000
R32_STK_CMPHR	0xE000F014	计数比较高位寄存器	0x00000000

9.5.5.1 系统计数控制寄存器（STK_CTLR）

偏移地址：0x00

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

位	名称	访问	描述	复位值
31	SWIE	RW	软件中断触发使能(SWI)： 1：触发软件中断； 0：关闭触发。 进入软件中断后，需软件清 0，否则持续触发。	0
[30:6]	Reserved	RO	保留。	
5	INIT	W1	计数器初始值更新： 1：向上计数时更新为 0，向下计数时更新为比较值； 0：无效。	
4	MODE	RW	计数模式： 1：向下计数； 0：向上计数。	
3	STRE	RW	自动重装载计数使能位： 1：向上计数到比较值后重新从 0 开始计数，向下计数到 0 后，重新从比较值开始计数； 0：向上计数到比较值后继续向上计数，向下计数到 0 后，重新从最大值开始向下计数。	
2	STCLK	RW	计数器时钟源选择位： 1：HCLK 做时基； 0：HCLK/8 做时基；	
1	STIE	RW	计数器中断使能控制位： 1：使能计数器中断； 0：关闭计数器中断。	
0	STE	RW	系统计数器使能控制位： 1：启动系统计数器 STK； 0：关闭系统计数器 STK，计数器停止计数。	0

9.5.5.2 系统计数状态寄存器（STK_SR）

偏移地址：0x04

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

位	名称	访问	描述	复位值
[31:1]	Reserved	RO	保留	0

0	CNTIF	RWO	计数值比较标志，写 0 清除，写 1 无效： 1：向上计数达到比较值，向下计数到 0； 0：未达到比较值。	0

9.5.5.3 系统计数器低位寄存器（STK_CNTL）

偏移地址：0x08

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

位	名称	访问	描述	复位值
[31:0]	CNTL	RW	当前计数器计数值低 32 位。	0

注：寄存器 STK_CNTL 和寄存器 STK_CNTH 共同构成了 64 位系统计数器。

9.5.5.4 系统计数器高位寄存器（STK_CNTH）

偏移地址：0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[47:32]															

位	名称	访问	描述	复位值
[31:0]	CNTH	RW	当前计数器计数值高 32 位。	0

注：寄存器 STK_CNTL 和寄存器 STK_CNTH 共同构成了 64 位系统计数器。

9.5.5.5 计数比较低位寄存器（STK_CMPLR）

偏移地址：0x10

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

位	名称	访问	描述	复位值
[31:0]	CMPL	RW	设置比较计数器值低 32 位。	0

注：寄存器 STK_CMPLR 和寄存器 STK_CMPHR 共同构成了 64 位计数器比较值。

9.5.5.6 计数比较高位寄存器（STK_CMPHR）

偏移地址：0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP[47:32]															

位	名称	访问	描述	复位值
[31:0]	CMPH	RW	设置比较计数器值高 32 位。	0

注：寄存器 STK_CMPLR 和寄存器 STK_CMPHR 共同构成了 64 位计数器比较值。

第 10 章 GPIO 及其复用功能(GPIO/AFIO)

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

GPIO 口可以配置成多种输入或输出模式，内置可关闭的上拉或下拉电阻，可以配置成推挽或开漏功能。GPIO 口还可以复用成其他功能。

10.1 主要特征

端口的每个引脚都可以配置成以下的多种模式之一：

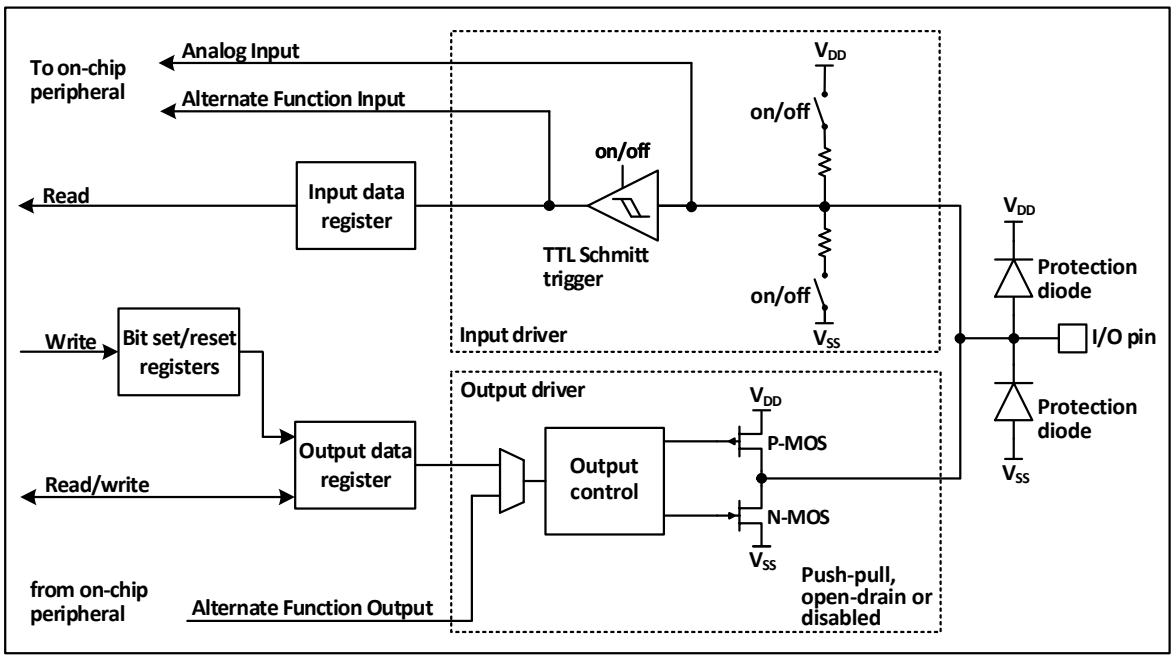
- 浮空输入
- 上拉输入
- 下拉输入
- 模拟输入
- 开漏输出
- 推挽输出
- 复用功能的输入和输出

许多引脚拥有复用功能，很多其他的外设把自己的输出和输入通道映射到这些引脚上，这些复用引脚具体用法需要参照各个外设，而对这些引脚是否复用和是否重映射的内容由本章说明。

10.2 功能描述

10.2.1 概述

图 10-1 GPIO 模块基本结构框图



如图 10-1 所示 IO 口结构，每个引脚在芯片内部都有两只保护二极管，IO 口内部可分为输入和输出驱动模块。其中输入驱动有弱上下拉电阻可选，可连接到 AD 等模拟输入的外设；如果输入到数字外设，就需要经过一个 TTL 施密特触发器，再连接到 GPIO 输入寄存器或其他复用外设。而输出驱动有一对 MOS 管，可通过配置上下的 MOS 管是否使能来将 IO 口配置成开漏或推挽输出；输出驱动内部也可以配置成由 GPIO 控制输出还是由复用的其他外设控制输出。

10.2.2 GPIO 的初始化功能

刚复位后，GPIO 口运行在初始状态，这时大多数 IO 口都是运行在浮空输入状态，但也有 HSE 等外设相关的引脚是运行在外设复用的功能上。具体的初始化功能请参照引脚描述相关的章节。

10.2.3 外部中断

所有的 GPIO 口都可以被配置外部中断输入通道，但一个外部中断输入通道最多只能映射到一个 GPIO 引脚上，且外部中断通道的序号必须和 GPIO 端口的位号一致，比如 PA1（或 PB1、PC1、PD1、PE1 等）只能映射到 EXTI1 上，且 EXTI1 只能接受 PA1、PB1、PC1、PD1 或 PE1 等其中之一的映射，两方都是一对一的关系。

10.2.4 复用功能

使用复用功能必须要注意：

- 使用输入方向的复用功能，端口必须配置成复用输入模式，上下拉设置可根据实际需要来设置
- 使用输出方向的复用功能，端口必须配置成复用输出模式，推挽或开漏可根据实际情况设置
- 对于双向的复用功能，端口必须配置成复用输出模式，这时驱动器被配置成浮空输入模式

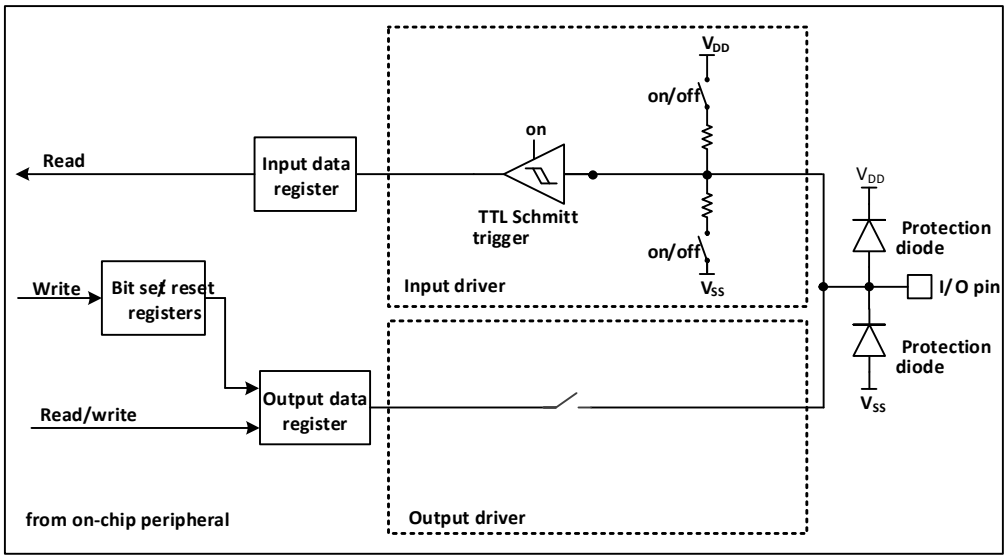
同一个 IO 口可能有多个外设复用到此管脚，因此为了使各个外设都有最大的发挥空间，外设的复用引脚除了默认复用引脚，还可以进行重映射，重映射到其他的引脚，避开被占用的引脚。

10.2.5 锁定机制

锁定机制可以锁定 IO 口的配置。经过特定的一个写序列后，选定的 IO 引脚配置将被锁定，在下一个复位前无法更改。

10.2.6 输入配置

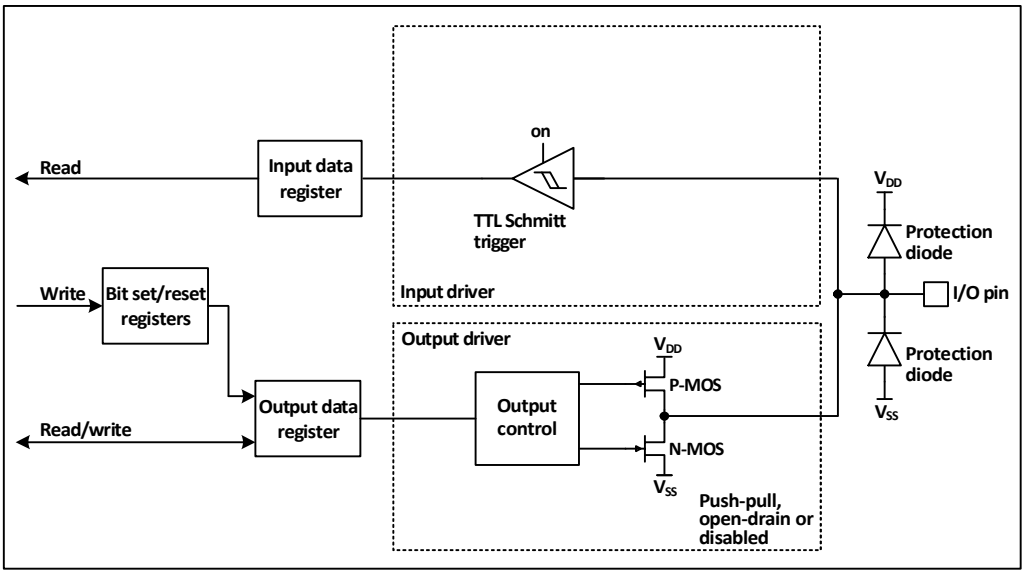
图 10-2 GPIO 模块输入配置结构框图



当 IO 口配置成输入模式时，输出驱动断开，输入上下拉可选，不连接复用功能和模拟输入。在每个 IO 口上的数据在每个 APB2 时钟被采样到输入数据寄存器，读取输入数据寄存器对应位即获取了对应引脚的电平状态。

10.2.7 输出配置

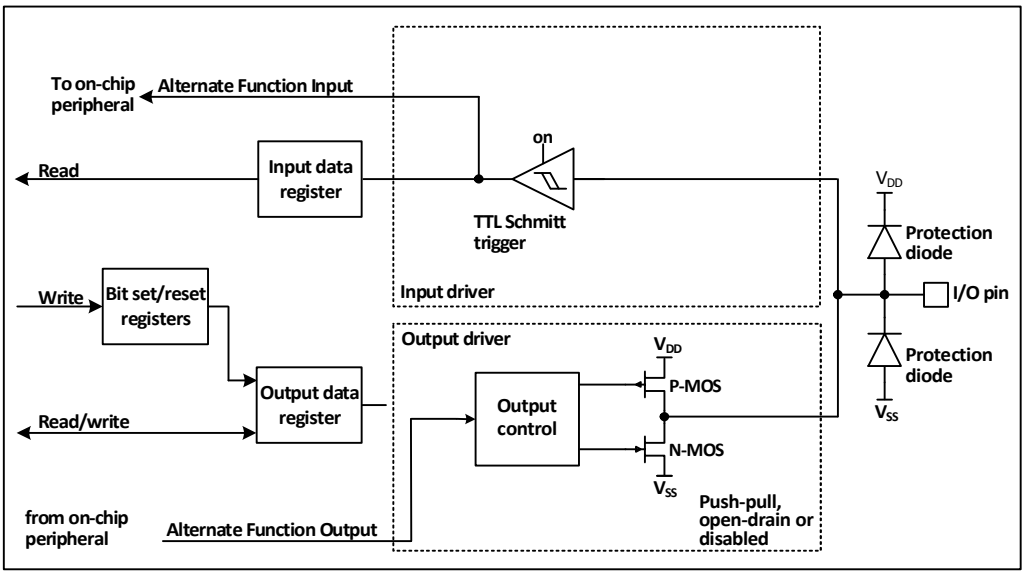
图 10-3 GPIO 模块输出配置结构框图



当 IO 口配置成输出模式时，输出驱动器中的一对 MOS 可根据需要被配置成推挽或开漏模式，不使用复用功能。输入驱动的上下拉电阻被禁用，TTL 施密特触发器被激活，出现在 IO 引脚上的电平将会在每个 APB2 时钟被采样到输入数据寄存器，所以读取输入数据寄存器将会得到 IO 状态，在推挽输出模式时，对输出数据寄存器的访问就会得到最后一次写入的值。

10.2.8 复用功能配置

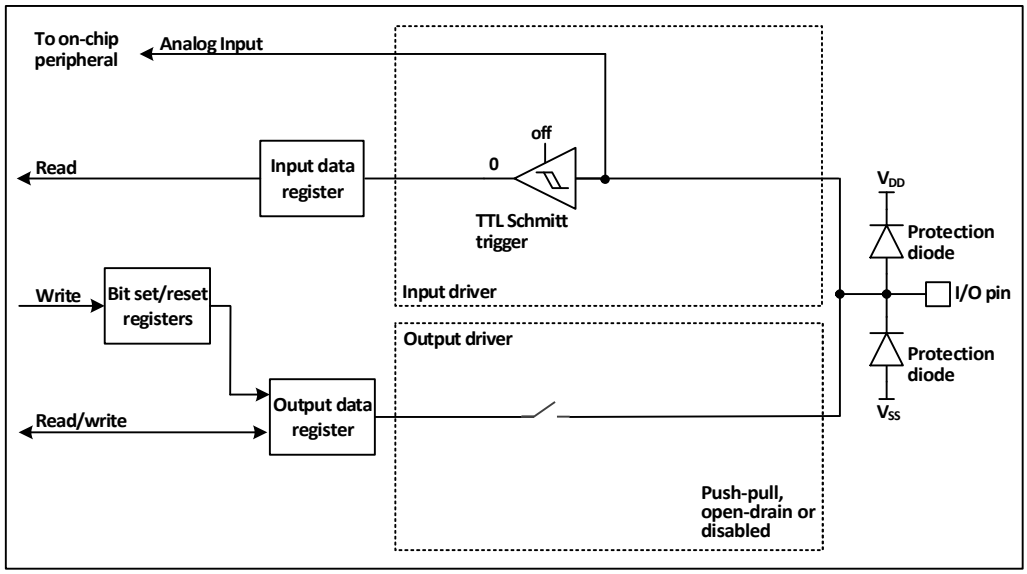
图 10-4 GPIO 模块被其他外设复用时的结构框图



在启用复用功能时，输出驱动器被使能，可以按需要被配置成开漏或推挽模式，施密特触发器也被打开，复用功能的输入和输出线都被连接，但是输出数据寄存器被断开，出现在 IO 引脚上的电平将会在每个 APB2 时钟被采样到输入数据寄存器，在开漏模式下，读取输入数据寄存器将会得到 IO 口当前状态；在推挽模式下，读取输出数据寄存器将会得到最后一次写入的值。

10.2.9 模拟输入配置

图 10-5 GPIO 模块作为模拟输入时的配置结构框图



在启用模拟输入时，输出缓冲器被断开，输入驱动中施密特触发器的输入被禁止以防止产生 IO 口上的消耗，上下拉电阻被禁止，读取输入数据寄存器将一直为 0。

10.2.10 外设的 GPIO 设置

下列表格推荐了各个外设的引脚相应的 GPIO 口配置。

表 10-1 高级定时器（TIM1/8/9/10）

TIM1/8/9/10	配置	GPIO 配置
TIM1/8/9/10_CHx	输入捕获通道 x	浮空输入
	输出比较通道 x	推挽复用输出
TIM1/8/9/10_CHxN	互补输出通道 x	推挽复用输出
TIM1/8/9/10_BKIN	刹车输入	浮空输入
TIM1/8/9/10_ETR	外部触发时钟输入	浮空输入

表 10-2 通用定时器（TIM2/3/4/5）

TIM2/3/4/5 引脚	配置	GPIO 配置
TIM2/3/4/5_CHx	输入捕获通道 x	浮空输入
	输出比较通道 x	推挽复用输出
TIM2/3/4/5_ETR	外部触发时钟输入	浮空输入

表 10-3 通用同步异步串行收发器（USART）

USART 引脚	配置	GPIO 配置
USARTx_TX	全双工模式	推挽复用输出
	半双工同步模式	推挽复用输出
USARTx_RX	全双工模式	浮空输入或带上拉输入
	半双工同步模式	未使用
USARTx_CK	同步模式	推挽复用输出
USARTx_RTS	硬件流量控制	推挽复用输出
USARTx_CTS	硬件流量控制	浮空输入或带上拉输入

表 10-4 串行外设接口（SPI）模块

SPI 引脚	配置	GPIO 配置
SPIx_SCK	主模式	推挽复用输出
	从模式	浮空输入
SPIx_MOSI	全双工主模式	推挽复用输出
	全双工从模式	浮空输入或带上拉输入
	简单的双向数据线/主模式	推挽复用输出
	简单的双向数据线/从模式	未使用
SPIx_MISO	全双工主模式	浮空输入或带上拉输入
	全双工从模式	推挽复用输出
	简单的双向数据线/主模式	未使用
	简单的双向数据线/从模式	推挽复用输出
SPIx_NSS	硬件主或从模式	浮空、上拉或下拉输入
	硬件主模式	推挽复用输出
	软件模式	未使用

表 10-5 内置音频总线（I2S）模块

I2S 引脚	配置	GPIO 配置
I2Sx_WS	主模式	推挽复用输出
	从模式	浮空输入
I2Sx_CK	主模式	推挽复用输出
	从模式	浮空输入
I2Sx_SD	发送器	推挽复用输出
	接收器	浮空、上拉或下拉输入
I2Sx_MCK	主模式	推挽复用输出
	从模式	未使用

表 10-6 内部集成总线（I2C）模块

I^2C 引脚	配置	GPIO 配置
I^2C_SCL	I^2C 时钟	开漏复用输出
I^2C_SDA	I^2C 数据	开漏复用输出

表 10-7 控制器局域网（CAN）模块

CAN 引脚	GPIO 配置
CANx_TX	推挽复用输出
CANx_RX	浮空输入或上拉输入

表 10-8 USB 全速设备（USBD）控制器

USBD 引脚	GPIO 配置
USBD_DM/USBD_DP	使能了 USB 模块之后，复用 IO 口会自动连接到内部 USBD 收发器

表 10-9 USB 主机设备（USBHD）控制器

USBHD 引脚	GPIO 配置
USBHD_DM/USBHD_DP	使能了 USB 模块之后，复用 IO 口会自动连接到内部 USBHD 收发器

表 10-10 USB OTG 控制器

USB OTG 引脚	GPIO 配置
OTG_FS_VBUS	模拟输入
OTG_FS_ID	上拉输入
OTG_FS_DM	由 USB 断电自动控制
OTG_FS_DP	由 USB 断电自动控制

表 10-11 安全数字输入输出（SDIO）模块

SDIO 引脚	配置	GPIO 配置
SDIO_CK	时钟	推挽复用输出
SDIO_CMD	命令	推挽复用输出
SDIO[D7:D0]	数据	推挽复用输出

表 10-12 直接存储器访问控制（FSMC）控制器

FSMC 引脚	GPIO 配置
FSMC_A[23:16] FSMC_D[15:0]	推挽复用输出
FSMC_CK	推挽复用输出
FSMC_NOE FSMC_NWE	推挽复用输出
FSMC_NE1 FSMC_NCE2	推挽复用输出
FSMC_NWAIT	浮空输入或带上拉输入
FSMC_NBL[1:0]	推挽复用输出

表 10-13 模拟转数字转换器（ADC）及数字转模拟转换器（DAC）

ADC/DAC 引脚	GPIO 配置
ADC/DAC	模拟输入

表 10-14 其他的 IO 功能设置

引脚	配置功能	GPIO 配置
TAMPER_RTC	RTC 输出	硬件自动设置
	侵入事件输入	
MCO	时钟输出	推挽复用输出
EXTI	外部中断输入	浮空、上拉或下拉输入

10.3 寄存器描述

10.3.1 GPIO 的寄存器描述

除非特殊说明，GPIO 的寄存器必须以字的方式操作（以 32 位来操作这些寄存器）。

表 10-15 GPIO 相关寄存器列表

名称	访问地址	描述	复位值
R32_GPIOA_CFGLR	0x40010800	PA 端口配置寄存器低位	0x44444444
R32_GPIOB_CFGLR	0x40010C00	PB 端口配置寄存器低位	0x44444444
R32_GPIOC_CFGLR	0x40011000	PC 端口配置寄存器低位	0x44444444

R32_GPIOD_CFGLR	0x40011400	PD 端口配置寄存器低位	0x44444444
R32_GPIOE_CFGLR	0x40011800	PE 端口配置寄存器低位	0x44444444
R32_GPIOA_CFGHR	0x40010804	PA 端口配置寄存器高位	0x44444444
R32_GPIOB_CFGHR	0x40010C04	PB 端口配置寄存器高位	0x44444444
R32_GPIOC_CFGHR	0x40011004	PC 端口配置寄存器高位	0x44444444
R32_GPIOD_CFGHR	0x40011404	PD 端口配置寄存器高位	0x44444444
R32_GPIOE_CFGHR	0x40011804	PE 端口配置寄存器高位	0x44444444
R32_GPIOA_INDR	0x40010808	PA 端口输入数据寄存器	0x0000XXXX
R32_GPIOB_INDR	0x40010C08	PB 端口输入数据寄存器	0x0000XXXX
R32_GPIOC_INDR	0x40011008	PC 端口输入数据寄存器	0x0000XXXX
R32_GPIOD_INDR	0x40011408	PD 端口输入数据寄存器	0x0000XXXX
R32_GPIOE_INDR	0x40011808	PE 端口输入数据寄存器	0x0000XXXX
R32_GPIOA_OUTDR	0x4001080C	PA 端口输出数据寄存器	0x00000000
R32_GPIOB_OUTDR	0x40010C0C	PB 端口输出数据寄存器	0x00000000
R32_GPIOC_OUTDR	0x4001100C	PC 端口输出数据寄存器	0x00000000
R32_GPIOD_OUTDR	0x4001140C	PD 端口输出数据寄存器	0x00000000
R32_GPIOE_OUTDR	0x4001180C	PE 端口输出数据寄存器	0x00000000
R32_GPIOA_BSHR	0x40010810	PA 端口置位/复位寄存器	0x00000000
R32_GPIOB_BSHR	0x40010C10	PB 端口置位/复位寄存器	0x00000000
R32_GPIOC_BSHR	0x40011010	PC 端口置位/复位寄存器	0x00000000
R32_GPIOD_BSHR	0x40011410	PD 端口置位/复位寄存器	0x00000000
R32_GPIOE_BSHR	0x40011810	PE 端口置位/复位寄存器	0x00000000
R32_GPIOA_BCR	0x40010814	PA 端口复位寄存器	0x00000000
R32_GPIOB_BCR	0x40010C14	PB 端口复位寄存器	0x00000000
R32_GPIOC_BCR	0x40011014	PC 端口复位寄存器	0x00000000
R32_GPIOD_BCR	0x40011414	PD 端口复位寄存器	0x00000000
R32_GPIOE_BCR	0x40011814	PE 端口复位寄存器	0x00000000
R32_GPIOA_LCKR	0x40010818	PA 端口锁定配置寄存器	0x00000000
R32_GPIOB_LCKR	0x40010C18	PB 端口锁定配置寄存器	0x00000000
R32_GPIOC_LCKR	0x40011018	PC 端口锁定配置寄存器	0x00000000
R32_GPIOD_LCKR	0x40011418	PD 端口锁定配置寄存器	0x00000000
R32_GPIOE_LCKR	0x40011818	PE 端口锁定配置寄存器	0x00000000

10.3.1.1 GPIO 配置寄存器低位（GPIOx_CFGLR）（x=A/B/C/D/E）

偏移地址：0x00

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CNF7[1:0]	MODE7[1:0]	CNF6[1:0]	MODE6[1:0]	CNF5[1:0]	MODE5[1:0]	CNF4[1:0]	MODE4[1:0]
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
CNF3[1:0]	MODE3[1:0]	CNF2[1:0]	MODE2[1:0]	CNF1[1:0]	MODE1[1:0]	CNF0[1:0]	MODE0[1:0]

位	名称	访问	描述	复位值
[31:30] [27:26]	CNFy	RW	(y=0-7)，端口 x 的配置位，通过这些位配置相应的端口。	01b

[23:22] [19:18] [15:14] [11:10] [7:6] [3:2]			在输入模式时（MODE=00b）： 00:模拟输入模式； 01：浮空输入模式； 10：带有上下拉模式。 11：保留。 在输出模式（MODE>00b）： 00：通用推挽输出模式； 01：通用开漏输出模式； 10：复用功能推挽输出模式； 11：复用功能开漏输出模式。	
[29:28] [25:24] [21:20] [17:16] [13:12] [9:8] [5:4] [1:0]	MODEy	RW	(y=0-7)，端口 x 模式选择，通过这些位配置相应的端口。 00：输入模式； 01：输出模式，最大速度 10MHz； 10：输出模式，最大速度 2MHz； 11：输出模式，最大速度 50MHz。	0

10.3.1.2 GPIO 配置寄存器高位（GPIOx_CFGHR）（x=A/B/C/D/E）

偏移地址：0x04

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
CNF15[1:0]	MODE15[1:0]	CNF14[1:0]	MODE14[1:0]	CNF13[1:0]	MODE13[1:0]	CNF12[1:0]	MODE12[1:0]
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
CNF11[1:0]	MODE11[1:0]	CNF10[1:0]	MODE10[1:0]	CNF9[1:0]	MODE9[1:0]	CNF8[1:0]	MODE8[1:0]

位	名称	访问	描述	复位值
[31:30] [27:26] [23:22] [19:18] [15:14] [11:10] [7:6] [3:2]	CNFy	RW	(y=8-15)，端口 x 的配置位，通过这些位配置相应的端口。 在输入模式时（MODE=00b）： 00：模拟输入模式； 01：浮空输入模式； 10：带有上下拉模式。 11：保留。 在输出模式（MODE>00b）： 00：通用推挽输出模式； 01：通用开漏输出模式； 10：复用功能推挽输出模式； 11：复用功能开漏输出模式。	01b
[29:28] [25:24] [21:20] [17:16] [13:12] [9:8]	MODEy	RW	(y=8-15)，端口 x 的模式位，通过这些位配置相应的端口。 00：输入模式； 01：输出模式，最大速度 10MHz； 10：输出模式，最大速度 2MHz； 11：输出模式，最大速度 50MHz。	0

[5:4] [1:0]				

10.3.1.3 端口输入寄存器（GPIOx_INDR）（x=A/B/C/D/E）

偏移地址：0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	IDRy	RO	(y=0-15)，端口输入数据。这些位只读并只能以 16 位形式读出。读出的值就是对应位的高低状态。	X

10.3.1.4 端口输出寄存器（GPIOx_OUTDR）（x=A/B/C/D/E）

偏移地址：0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	ODRy	RW	(y=0-15)，端口输出的数据。这些数据只能以 16 位的形式操作。IO 口对外输出这些寄存器的值。	0

10.3.1.5 端口复位/置位寄存器（GPIOx_BSHR）（x=A/B/C/D/E）

偏移地址：0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0

位	名称	访问	描述	复位值
[31:16]	BRy	WO	(y=0-15)，对这些位置位会清除对应的 OUTDR 位，写 0 不产生影响。这些位只能以 16 位的形式访问。如果同时设置了 BR 和 BS 位，则 BS 位起作用。	0
[15:0]	BSy	WO	(y=0-15)，对这些位置位会使对应的 OUTDR 位置位，写 0 不产生影响。这些位只能以 16 位	0

			的形式访问。如果同时设置了 BR 和 BS 位，则 BS 位起作用。	

10.3.1.6 端口复位寄存器（GPIOx_BCR）（x=A/B/C/D/E）

偏移地址：0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	BRy	WO	(y=0-15)，对这些位置位会清除对应的 OUTDR 位，写 0 不产生影响。这些位只能以 16 位的形式访问。	0

10.3.1.7 配置锁定寄存器（GPIOx_LCKR）（x=A/B/C/D/E）

偏移地址：0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LCKK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留	0
16	LCKK	RW	锁定键，它可以通过特定的序列写入实现锁定，但它可以随时读出。它读出为 0 时表示未锁定生效，读出 1 时表示锁定生效。 锁定键的写入序列为：写 1-写 0-写 1-读 0-读 1，最后一步非必要，但是可以用以确认锁定键已经激活。 在写入序列时任何错误都不会使激活锁定，且在写入序列时，不能更改 LCK[15:0]的值。锁定生效后，只有在下次复位后才能更改端口的配置。	0
[15:0]	LCKy	RW	(y=0-15)，这些位为 1 时表示锁定对应端口的配置。只能在 LCKK 未锁定前改变这些位。锁定的配置指的是配置寄存器 GPIOx_CFGLR 和 GPIOx_CFGHR。	0

注：当对相应的端口位执行了 LOCK 序列后，在下次系统复位之前将不能再更改端口位的配置。

10.3.2 AFIO 寄存器

除非特殊说明，AFIO 的寄存器必须以字的方式操作（以 32 位来操作这些寄存器）。

表 10-16 AFIO 相关寄存器列表

名称	访问地址	描述	复位值
R32_AFIO_ECR	0x40010000	事件控制寄存器	0x00000000
R32_AFIO_PCFR1	0x40010004	重映射寄存器 1	0x00000000
R32_AFIO_EXTICR1	0x40010008	外部中断配置寄存器 1	0x00000000
R32_AFIO_EXTICR2	0x4001000C	外部中断配置寄存器 2	0x00000000
R32_AFIO_EXTICR3	0x40010010	外部中断配置寄存器 3	0x00000000
R32_AFIO_EXTICR4	0x40010014	外部中断配置寄存器 4	0x00000000
R32_AFIO_PCFR2	0x4001001C	重映射寄存器 2	0x00000000

10.3.2.1 事件控制寄存器（AFIO_ECR）

偏移地址：0x00

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

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
7	EVOE	RW	允许事件输出位，对该位置位会使内核的 EVENTOUT 连接到 PORT 和 PIN 选定的 IO 口。	0
[6:4]	PORT	RW	用于选择内核输出 EVENTOUT 的端口： 000：选择 PA 口；001：选择 PB 口； 010：选择 PC 口；011：选择 PD 口； 其他：保留。	0
[3:0]	PIN	RW	此位的值用来确定选择内核输出 EVENTOUT 到端口的具体引脚号，值 0-15 分别对应 PORT 中选定的 Px 的第 0-15 号引脚。	0

10.3.2.2 重映射寄存器 1（AFIO_PCFR1）

偏移地址：0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	PTP_PPSP_RM	TIM2ITRA_RM	SPI3_RM	Reserved	SWCFG[2:0]			MII_RMII_SEL	CAN2_RM	ETH_RM	ADC2_ETRGREG_RM	ADC2_ETRGINJ_RM	ADC1_ETRGREG_RM	ADC1_ETRGINJ	TIM5CH4_RM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD01RM	CANRM[1:0]		TIM4_RM	TIM3RM[1:0]		TIM2RM[1:0]		TIM1RM[1:0]		USART3RM[1:0]		USART2RM	USART1RM	I2C1RM	SPI1RM

位	名称	访问	描述	复位值
31	Reserved	RO	保留。	0
30	PTP_PPSP_RM	RW	以太网的 PTP PPS 重映射。 0：PTP PPS 不输出到 PB5 引脚； 1：PTP PPS 输出到 PB5 引脚。	0
29	TIM2ITRA_RM	RW	TIM2 内部触发 1 重映射。 0：在内部连接 TIM2_ITR1 至以太网的 PTP 输出； 1：在内部连接 TIM2_ITR1 至全速 USB OTG 的 SOF 输出。	0
28	SPI3_RM	RW	SPI3 重映射。 0：默认映射（NSS/PA15、SCK/PB3、MSIO/PB4、MOSI/PB5）； 1：重映射（NSS/PA4、SCK/PC10、MSIO/PC11、MOSI/PC12）。	0
31	Reserved	RO	保留。	0
[26:24]	SWCFG	WO	这些位用以配置 SW 功能和跟踪功能的 IO 口。SWD（RVSWD）是访问内核的调试接口。系统复位后总是作为 SWD 端口。 0xx：启用 SWD（RVSWD）； 100：关闭 SWD（RVSWD），作为 GPIO 功能； 其他：无效。	0
23	MII_RMII_SEL	RW	MII 或 RMII 选择。配置内部的以太网 MAC 适用外部的 MII 接口还是 RMII 接口的收发器（PHY）。 0：配置以太网的 MAC 使用外部 MII 接口的收发器（PHY）； 1：配置以太网的 MAC 使用外部 RMII 接口的收发器（PHY）；	0
22	CAN2RM	RW	CAN2 重映射位。 0：默认映射（CAN2_RX/PB12，CAN2_TX/PB13）； 1：重映射（CAN2_RX/PB5，CAN2_TX/PB6）。	0
21	ETHRM	RW	以太网的重映射位。 0：默认映射（RX_DV-CRS_DV/PA7，RXD0/PC4，RXD1/PC5，RXD2/PB0，RXD3/PB1）； 1：重映射（RX_DV-CRS_DV/PD8，RXD0/PD9，RXD1/PD10，RXD2/PD11，RXD3/PD12）	0
20	ADC2_ETRGREG_RM	RW	ADC1 外部触发规则转换的重映射位。 0：ADC1 外部触发规则转换与 EXTI11 相连； 1：ADC1 外部触发规则转换与 TIM8_TRGO 相连；	0
19	ADC2_ETRGINJ_RM	RW	ADC2 外部触发注入转换的重映射位。 0：ADC2 外部触发注入转换与 EXTI15 相连； 1：ADC2 外部触发注入转换与 TIM8_CH4 相连；	0
18	ADC1_ETRGREG_RM	RW	ADC1 外部触发规则转换的重映射位。 0：ADC1 外部触发规则转换与 EXTI11 相连； 1：ADC1 外部触发规则转换与 TIM8_TRGO 相连；	0

17	ADC1_ETRGINJ_RM	RW	ADC1 外部触发注入转换的重映射位。 0：ADC1 外部触发注入转换与 EXTI15 相连； 1：ADC1 外部触发注入转换与 TIM8_CH4 相连；	0
16	TIM5CH4RM	RW	定时器 5 通道的重映射。 0：默认映射，定时器 5 通道 4 的重映射； 1：重映射，定时器 5 通道 4 映射至 LSI 内部时钟。	0
15	PD01RM	RW	引脚 PD0&PD1 重映射位，该位可由用户读写。它控制 PD0 和 PD1 的 GPIO 功能是否进行重映射，即 PD0&PD1 映射到 OSC_IN&OSC_OUT。当启用主振荡器 HSE 时(系统运行于内部的 8MHz 阻容振荡器)，PD0 和 PD1 可以映射到 OSC_IN 和 OSC_OUT 引脚。 0：不进行 PD0 和 PD1 的重映射； 1：PD0 映射到 OSC_IN，PD1 映射到 OSC_OUT；	0
[14:13]	CAN1RM	RW	CAN1 复用功能重映射位，这些位可由用户读写。控制 CAN_RX 和 CAN_TX 的重映射： 00：CAN1_RX 映射到 PA11，CAN1_TX 映射到 PA12； 10：CAN1_RX 映射到 PB8，CAN1_TX 映射到 PB9； 01：保留； 11：CAN1_RX 映射到 PD0，CAN1_TX 映射到 PD1；	0
12	TIM4_RM	RW	定时器 4 的重映射位，该位可由用户读写。它控制定时器 4 的通道 1 至 4 在 GPIO 端口的重映射： 0：默认映射（CH1/PB6，CH2/PB7，CH3/PB8，CH4/PB9）； 1：重映射（CH1/PD12，CH2/PD13，CH3/PD14，CH4/PD15）。	0
[11:10]	TIM3RM	RW	定时器 3 的重映射位，这些位可由用户读写。它控制定时器 3 的通道 1 至 4 在 GPIO 端口的重映射： 00：默认映射(CH1/PA6，CH2/PA7，CH3/PB0，CH4/PB1)； 01：保留； 10：部分映射(CH1/PB4，CH2/PB5，CH3/PB0，CH4/PB1)； 11：完全映射(CH1/PC6，CH2/PC7，CH3/PC8，CH4/PC9)； *注：重映射不影响在 PD2 上的 TIM3_ETR。*	0
[9:8]	TIM2RM	RW	定时器 2 的重映射位。这些位可由用户读写。它控制定时器 2 的通道 1 至 4 和外部触发(ETR)在 GPIO 端口的映射： 00：默认映射(CH1/ETR/PA0，CH2/PA1，CH3/PA2，CH4/PA3)； 01：部分映射(CH1/ETR/PA15，CH2/PB3，	0

			CH3/PA2，CH4/PA3）； 10：部分映射（CH1/ETR/PA0，CH2/PA1，CH3/PB10，CH4/PB11）； 11：完全映射（CH1/ETR/PA15，CH2/PB3，CH3/PB10，CH4/PB11）。	
[7:6]	TIM1RM	RW	定时器 1 的重映射位。这些位可由用户读写。它控制定时器 1 的通道 1 至 4、1N 至 3N、外部触发（ETR）和刹车输入（BKIN）在 GPIO 端口的映射： 00：默认映射（ETR/PA12，CH1/PA8，CH2/PA9，CH3/PA10，CH4/PA11，BKIN/PB12，CH1N/PB13，CH2N/PB14，CH3N/PB15）； 01：部分映射（ETR/PA12，CH1/PA8，CH2/PA9，CH3/PA10，CH4/PA11，BKIN/PA6，CH1N/PA7，CH2N/PB0，CH3N/PB1）； 10：保留； 11：完全映射（ETR/PE7，CH1/PE9，CH2/PE11，CH3/PE13，CH4/PE14，BKIN/PE15，CH1N/PE8，CH2N/PE10，CH3N/PE12）。	0
[5:4]	USART3RM	RW	USART3 的重映射位，这些位可由用户读写。它控制 USART3 的 CTS、RTS、CK、TX 和 RX 复用功能在 GPIO 端口的映射： 00：默认映射（TX/PB10，RX/PB11，CK/PB12，CTS/PB13，RTS/PB14）； 01：部分重映射（TX/PC10，RX/PC11，CK/PC12，CTS/PB13，RTS/PB14）； 10：保留； 11：完全重映射（TX/PD8，RX/PD9，CK/PD10，CTS/PD11，RTS/PD12）。 *注：CH32V20x_D6、CH32F20x_D6 只存在默认映射（00b）。*	0
3	USART2RM	RW	USART2 的重映射位。该位可由用户读写。它控制 USART2 的 CTS、RTS、CK、TX 和 RX 复用功能在 GPIO 端口的映射： 0：默认映射（CTS/PA0，RTS/PA1，TX/PA2，RX/PA3，CK/PA4）； 1：重映射（CTS/PD3，RTS/PD4，TX/PD5，RX/PD6，CK/PD7）。 *注：CH32V20x_D6、CH32F20x_D6 只存在默认映射（0b）。*	0
2	USART1RM	RW	UART1 映射配置低位（配合 AFIO_PCFR2 寄存器 bit26 USART1RM1 使用）。 00：默认映射（CK/PA8，TX/PA9，RX/PA10，CTS/PA11，RTS/PA12）； 01：重映射（CK/PA8，TX/PB6，RX/PB7，CTS/PA11，RTS/PA12）；	0

			10：重映射(CK/PA10，TX/PB15，RX/PA8，CTS/PA5，RTS/PA9)； 11：重映射(CK/PA5，TX/PA6，RX/PA7，CTS/PC4，RTS/PC5)。 *注：CH32V20x_D6、CH32F20x_D6 只存在默认映射(00b)、重映射（01b）。*	
1	I2C1RM	RW	I2C1 的重映射。该位可由用户读写。它控制 I2C1 的 SCL 和 SDA 复用功能在 GPIO 端口的映射： 0：默认映射(SCL/PB6，SDA/PB7)； 1：重映射(SCL/PB8，SDA/PB9)。	0
0	SPI1RM	RW	SPI1 的重映射。该位可由用户读写。它控制 SPI1 的 NSS、SCK、MISO 和 MOSI 复用功能在 GPIO 端口的映射： 0：默认映射(NSS/PA4，SCK/PA5，MISO/PA6，MOSI/PA7)； 1：重映射(NSS/PA15，SCK/PB3，MISO/PB4，MOSI/PB5)。	0

10.3.2.3 外部中断配置寄存器 1（AFIO_EXTICR1）

偏移地址：0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:12] [11:8] [7:4] [3:0]	EXTIx	RW	(x=0-3)，外部中断输入引脚配置位。用以决定外部中断引脚映射到哪个端口的引脚上： 0000：PA 引脚的第 x 个引脚； 0001：PB 引脚的第 x 个引脚； 0010：PC 引脚的第 x 个引脚； 0011：PD 引脚的第 x 个引脚； 0100：PE 引脚的第 x 个引脚； 其他：保留。	0

10.3.2.4 外部中断配置寄存器 2（AFIO_EXTICR2）

偏移地址：0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:12] [11:8] [7:4] [3:0]	EXTIx	RW	(x=4-7)，外部中断输入引脚配置位。用以决定外部中断引脚映射到哪个端口的引脚上： 0000：PA 引脚的第 x 个引脚； 0001：PB 引脚的第 x 个引脚； 0010：PC 引脚的第 x 个引脚； 0011：PD 引脚的第 x 个引脚； 0100：PE 引脚的第 x 个引脚； 其他：保留。	0

10.3.2.5 外部中断配置寄存器 3（AFIO_EXTICR3）

偏移地址：0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:12] [11:8] [7:4] [3:0]	EXTIx	RW	(x=8-11)，外部中断输入引脚配置位。用以决定外部中断引脚映射到哪个端口的引脚上： 0000：PA 引脚的第 x 个引脚； 0001：PB 引脚的第 x 个引脚； 0010：PC 引脚的第 x 个引脚； 0011：PD 引脚的第 x 个引脚； 0100：PE 引脚的第 x 个引脚； 其他：保留。	0

10.3.2.6 外部中断配置寄存器 4（AFIO_EXTICR4）

偏移地址：0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:12] [11:8] [7:4] [3:0]	EXTIx	RW	(x=12-15)，外部中断输入引脚配置位。用以决定外部中断引脚映射到哪个端口的引脚上： 0000：PA 引脚的第 x 个引脚； 0001：PB 引脚的第 x 个引脚； 0010：PC 引脚的第 x 个引脚； 0011：PD 引脚的第 x 个引脚；	0

			0100：PE 引脚的第 x 个引脚； 其他：保留。	

10.3.2.7 重映射寄存器 2（AFIO_PCFR2）

偏移地址：0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					USART1_RM2	UART8_RM		UART7_RM		UART6_RM		UART5_RM		UART4_RM	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					FSMC_NADV	Reserved			TIM10_RM		TIM9_RM		TIM8_RM	Reserved	

位	名称	访问	描述	复位值
[31:27]	Reserved	RO	保留。	0
26	USART1_RM2	RW	UART1 映射配置高位（配合 AFIO_PCFR1 寄存器 bit2 USART1RM 使用）。 00：默认映射（CK/PA8，TX/PA9，RX/PA10，CTS/PA11，RTS/PA12）； 01：重映射（CK/PA8，TX/PB6，RX/PB7，CTS/PA11，RTS/PA12）； 10：重映射（CK/PA10，TX/PB15，RX/PA8，CTS/PA5，RTS/PA9）； 11：重映射（CK/PA5，TX/PA6，RX/PA7，CTS/PC4，RTS/PC5）。 *注：CH32V20x_D6、CH32F20x_D6 只存在默认映射（00b）、重映射（01b）。*	0
[25:24]	UART8_RM	RW	UART8 重映射。 00：默认映射（TX/PC4，RX/PC5）； 01：重映射（TX/PA14，RX/PA15）； 1x：重映射（TX/PE14，RX/PE15）。	00b
[23:22]	UART7_RM	RW	UART7 重映射。 00：默认映射（TX/PC2，RX/PC3）； 01：重映射（TX/PA6，RX/PA7）； 1x：重映射（TX/PE12，RX/PE13）。	00b
[21:20]	UART6_RM	RW	UART6 重映射。 00：默认映射（TX/PC0，RX/PC1）； 01：重映射（TX/PB8，RX/PB9）； 1x：重映射（TX/PE10，RX/PE11）。	00b
[19:18]	UART5_RM	RW	UART5 重映射。 00：默认映射（TX/PC12，RX/PD2）； 01：重映射（TX/PB4，RX/PB5）； 1x：重映射（TX/PE8，RX/PE9）。	00b
[17:16]	UART4_RM	RW	UART4 重映射。 00：默认映射（TX/PC10，RX/PC11）； 01：重映射（TX/PB0，RX/PB1）；	00b

			1x：重映射(TX/PE0，RX/PE1)。 *注：适用于 CH32V30x_D8C、CH32V30x_D8、CH32V30x_D8W、CH32V20x_D8 、CH32F20x_D8C、CH32F20x_D8、CH32F20x_D8W。* x0：默认映射(CK/PB2，TX/PB0，RX/PB1，CTS/PB3，RTS/PB4)； x1：重映射(CK/PA6，TX/PA5，RX/PB5，CTS/PA7，RTS/PA15)。 *注：适用于 CH32V20x_D6、CH32F20x_D6。*	
[15:11]	Reserved	RW	保留。	0
10	FSMC_NADV	RW	FSMC_NADV 重映射。 0：FSMC NADV 映射到 PB7； 1：禁止 FSMC NADV 输出。	0
[9:7]	Reserved	RO	保留。	0
[6:5]	TIM10_RM	RW	TIM10 的重映射位。 00：默认映射(ETR/PC10，CH1/PB8，CH2/PB9，CH3/PC3，CH4/PC11，BKIN/PC12，CH1N/PA12，CH2N/PA13，CH3N/PA14)； 01：部分映射(ETR/PB11，CH1/PB3，CH2/PB4，CH3/PB5，CH4/PC15，BKIN/PB10，CH1N/PA5，CH2N/PA6，CH3N/PA7)； 1x：完全映射(ETR/PD0，CH1/PD1，CH2/PD3，CH3/PD5，CH4/PD7，BKIN/PE2，CH1N/PE3，CH2N/PE4，CH3N/PE5)。	00b
[4:3]	TIM9_RM	RW	TIM9 的重映射位。 00：默认映射(ETR/PA2，CH1/PA2，CH2/PA3，CH3/PA4，CH4/PC4，BKIN/PC5，CH1N/PC0，CH2N/PC1，CH3N/PC2)； 01：部分映射(ETR/PA2，CH1/PA2，CH2/PA3，CH3/PA4，CH4/PC14，BKIN/PA1，CH1N/PB0，CH2N/PB1，CH3N/PB2)； 1x：完全映射(ETR/PD9，CH1/PD9，CH2/PD11，CH3/PD13，CH4/PD15，BKIN/PD14，CH1N/PD8，CH2N/PD10，CH3N/PD12)。	00b
2	TIM8_RM	RW	TIM8 的重映射位。 0：默认映射(ETR/PA0，CH1/PC6，CH2/PC7，CH3/PC8，CH4/PC9，BKIN/PA6，CH1N/PA7，CH2N/PB0，CH3N/PB1)； 1：重映射(ETR/PA0，CH1/PB6，CH2/PB7，CH3/PB8，CH4/PC13，BKIN/PB9，CH1N/PA13，CH2N/PA14，CH3N/PA15)；	0
[1:0]	Reserved	RW	保留。	0

第 11 章 直接存储器访问控制（DMA）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

直接存储器访问控制器（DMA）提供在外设和存储器之间或存储器和存储器之间的高速数据传输方式，无须 CPU 干预，数据可以通过 DMA 快速地移动，以节省 CPU 的资源来做其他操作。

DMA 控制器每个通道专门用来管理来自于一个或多个外设对存储器访问的请求。还有一个仲裁器来协调各通道之间的优先级。

11.1 主要特性

- 多个独立可配置通道
- 每个通道都直接连接专用的硬件 DMA 请求，并支持软件触发
- 支持循环的缓冲器管理
- 多个通道之间的请求优先权可以通过软件编程设置(最高、高、中和低)，优先权设置相等时由通道号决定（通道号越低优先级越高）
- 支持外设到存储器、存储器到外设、存储器到存储器之间的传输
- 闪存、SRAM、外设的 SRAM、APB1、APB2 和 AHB 外设均可作为访问的源和目标
- 可编程的数据传输数目：最大为 65535

11.2 功能描述

11.2.1 DMA 通道处理

1）仲裁优先级

多个独立的通道产生的 DMA 请求通过逻辑或结构输入到 DMA 控制器，当前只会有一个通道的请求得到响应。模块内部的仲裁器根据通道请求的优先级来选择要启动的外设/存储器的访问。

软件管理中，应用程序通过对 DMA_CFGRx 寄存器的 PL[1:0]位设置，可以为每个通道独立配置优先等级，包括最高、高、中、低 4 个等级。当通道间的软件设置等级一致时，模块会按固定的硬件优先级选择，通道编号偏低的要比偏高的有较高优先权。

2）DMA 配置

当 DMA 控制器收到一个请求信号时，会访问发出请求的外设或存储器，建立外设或存储器和存储器之间的数据传输。主要包括下面 3 个操作步骤：

1） 从外设数据寄存器或当前外设/存储器地址寄存器指示的存储器地址取数据,第一次传输时的开始地址是 DMA_PADDRx 或 DMA_MADDRx 寄存器指定的外设基地址或存储器地址。
2） 存数据到外设数据寄存器或当前外设/存储器地址寄存器指示的存储器地址,第一次传输时的开始地址是 DMA_PADDRx 或 DMA_MADDRx 寄存器指定的外设基地址或存储器地址。
3） 执行一次 DMA_CNTRx 寄存器中数值的递减操作，该寄存器指示当前未完成转移的操作数目。

每个通道包括 3 种 DMA 数据转移方式：

- 外设到存储器（MEM2MEM=0，DIR=0）
- 存储器到外设（MEM2MEM=0，DIR=1）
- 存储器到存储器（MEM2MEM=1）

注：存储器到存储器方式无需外设请求信号，配置为此模式后（MEM2MEM=1），通道开启（EN=1）即可启动数据传输。此方式不支持循环模式。

配置过程如下：

1） 在 DMA_PADDRx 寄存器中设置外设寄存器的首地址或存储器到存储器方式（MEM2MEM=1）下存储器数据地址。发生 DMA 请求时，这个地址将是数据传输的源或目标地址。
2） 在 DMA_MADDRx 寄存器中设置存储器数据地址。发生 DMA 请求时，传输的数据将从这个地址读出或写入这个地址。
3） 在 DMA_CNTRx 寄存器中设置要传输的数据数量。在每个数据传输后，这个数值递减。
4） 在 DMA_CFGRx 寄存器的 PL[1:0]位中设置通道的优先级。
5） 在 DMA_CFGRx 寄存器中设置数据传输的方向、循环模式、外设和存储器的增量模式、外设和存储器的数据宽度、传输过半、传输完成、传输错误中断使能位，
6） 设置 DMA_CCRx 寄存器的 ENABLE 位，启动通道 x。

注：DMA_PADDRx/DMA_MADDRx/DMA_CNTRx 寄存器以及 DMA_CFGRx 寄存器中的数据传输的方向(DIR)、循环模式（位置）、外设和存储器的增量模式（MINC/PINC）等控制位只有在 DMA 通道被关闭下才可以配置写入。

3）循环模式

设置 DMA_CFGRx 寄存器的 CIRC 位置 1，可以启用通道数据传输的循环模式功能。循环模式下，当数据传输的数目变为 0 时，DMA_CNTRx 寄存器的内容会自动被重新加载为其初始数值，内部的外设和存储器地址寄存器也被重新加载为 DMA_PADDRx 和 DMA_MADDRx 寄存器设定的初始地址值，DMA 操作将继续进行，直到通道被关闭或关闭 DMA 模式。

4）DMA 处理状态

- 传输过半：对应 DMA_INTFR 寄存器中的 HTIFx 位硬件置位。当 DMA 的传输数目减至初始设定值一半以下将会产生 DMA 传输过半标志，如果在 DMA_CCRx 寄存器中置位了 HTIE，则将产生中断。硬件通过此标志提醒应用程序，可以为新一轮数据传输做准备。
- 传输完成：对应 DMA_INTFR 寄存器中的 TCIFx 位硬件置位。当 DMA 的传输数目减至 0 将会产生 DMA 传输完成标志，如果在 DMA_CCRx 寄存器中置位了 TCIE，则将产生中断。
- 传输错误：对应 DMA_INTFR 寄存器中的 TEIFx 位硬件置位。读写一个保留的地址区域，将会产生 DMA 传输错误。同时模块硬件会自动清 0 发生错误的通道所对应的 DMA_CCRx 寄存器的 EN 位，该通道被关闭。如果在 DMA_CCRx 寄存器中置位了 TEIE，则将产生中断。

应用程序在查询 DMA 通道状态时，可以先访问 DMA_INTFR 寄存器的 GIFx 位，判断出当前哪个通道发生了 DMA 事件，进而处理该通道的具体 DAM 事件内容。

11.2.2 可编程的数据传输总大小/数据位宽/对齐方式

DMA 每个通道一轮传输的数据量总大小可编程，最大 65535 次。DMA_CNTRx 寄存器中指示待传输数目。在 EN=0 时，写入设置值，在 EN=1 开启 DMA 传输通道后，此寄存器变为只读属性，在每次传输后数值递减。

外设和存储器的传输数据取值支持地址指针自动递增功能，指针增量可编程。它们访问的第一个传输的数据地址存放在 DMA_PADDRx 和 DMA_MADDRx 寄存器中，通过设置 DMA_CFGRx 寄存器的 PINC 位或 MINC 位置 1，可以分别开启外设地址自增模式或存储器地址自增模式，PSIZE[1:0]设置外设地址取数据大小及地址自增大小，MSIZE[1:0]设置存储器地址取数据大小及地址自增大小，包括 3 种选择：8 位、16 位、32 位。具体数据转移方式如下表：

表 11-1 不同数据位宽下 DMA 转移（PINC=MINC=1）

源端位宽	目标位宽	传输数目	源：地址/数据	目标：地址/数据	传输操作
8	8	4	0x00/B0	0x00/B0	● 源端地址递增量与源端设置的

			0x01/B1 0x02/B2 0x03/B3	0x01/B1 0x02/B2 0x03/B3	数据位宽对齐，取值大小等于源端数据位宽 ● 目标地址递增量与目标设置数据的位宽对齐，取值大小等于目标数据位宽 ● DMA 转移送入目标端的数据依据原则：数据大小不足高位补 0，数据大小溢出高位去掉 ● 存储数据方式：小端模式，低地址存放低字节，高地址存放高字节
8	16	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/00B0 0x02/00B1 0x04/00B2 0x06/00B3	
8	32	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/000000B0 0x04/000000B1 0x08/000000B2 0x0C/000000B3	
16	8	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B0 0x01/B2 0x02/B4 0x03/B6	
16	16	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	
16	32	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/0000B1B0 0x04/0000B3B2 0x08/0000B5B4 0x0C/0000B7B6	
32	8	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B0 0x01/B4 0x02/B8 0x03/BC	
32	16	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B1B0 0x02/B5B4 0x04/B9B8 0x06/BDBC	
32	32	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	

11.2.3 DMA 请求映射

青稞 V4F MCU（包括 CH32V30x_D8C、CH32V30x_D8）以及 ARM®Cortex™-M3 MCU（包括 CH32F20x_D8C、CH32F20x_D8），DMA 控制器提供 18 个通道，其中 DMA1 包含 7 个通道，DMA2 包含 11 个通道，每个通道对应多个外设请求，通过设置相应外设寄存器中对应 DMA 控制位，可以独立的开启或关闭各个外设的 DMA 功能，具体对应关系如下。

图 11-2 DMA1 请求映像

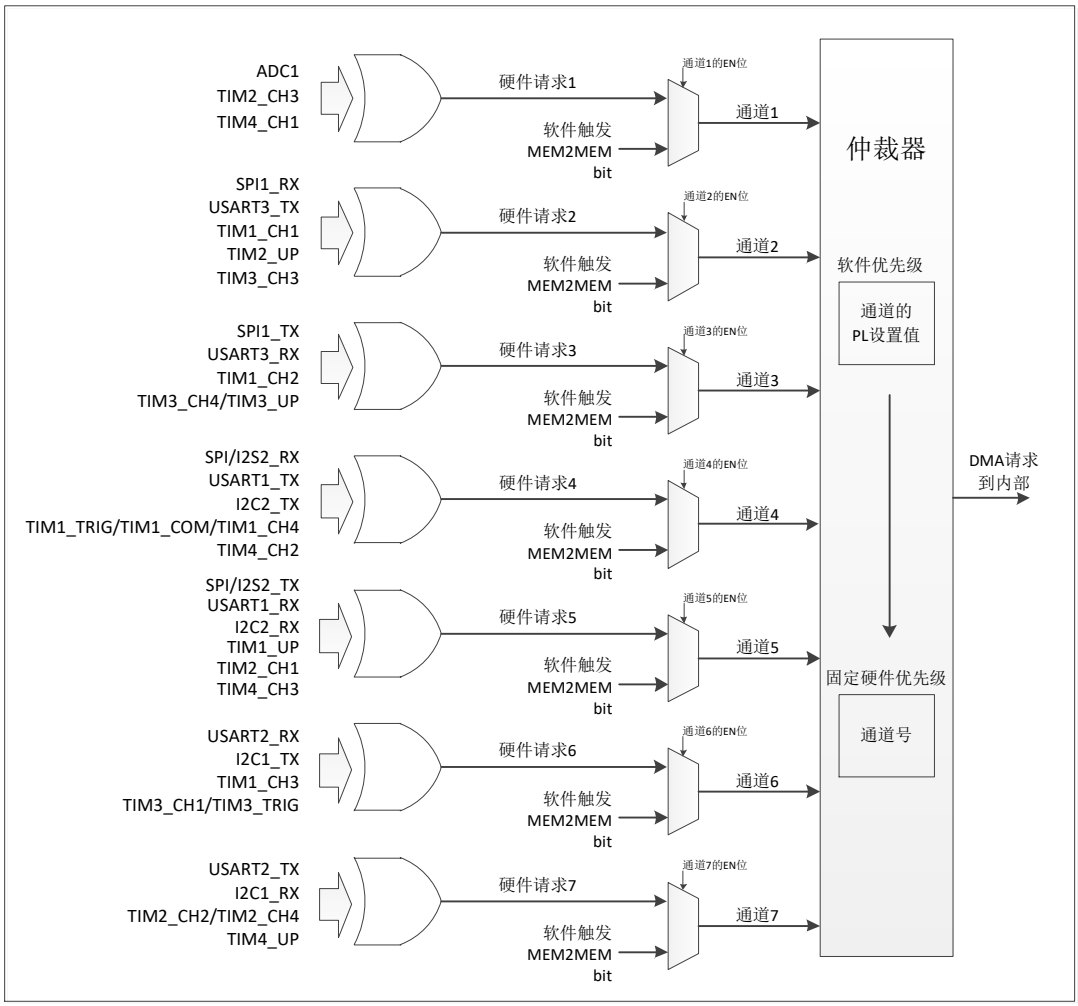


图 11-3 DMA2 请求映像

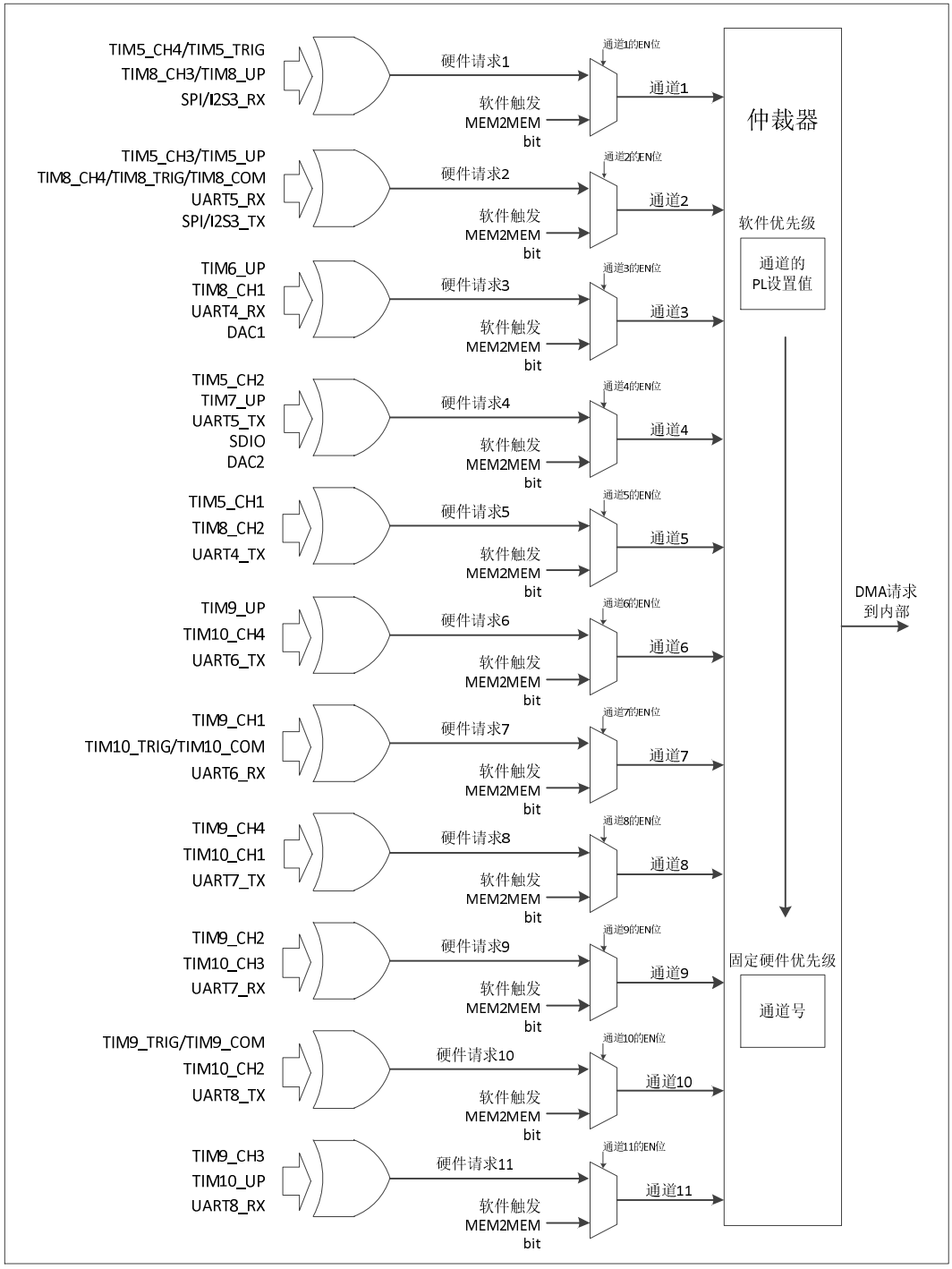


表 11-2 DMA1 各通道外设映射表

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7
ADC1	ADC1						
SPI1		SPI1_RX	SPI1_TX				
SPI/I2S2				SPI/I2S2_RX	SPI/I2S2_TX		
USART1				USART1_TX	USART1_RX		
USART2						USART2_RX	USART2_TX
USART3		USART3_TX	USART3_RX				
I2C1						I2C1_TX	I2C1_RX

I2C2				I2C2_TX	I2C2_RX		
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP			TIM3_CH1 TIM3_TRIG	
TIM4	TIM4_CH1			TIM4_CH2	TIM4_CH3		TIM4_UP

表 11-3 DMA2 各通道外设映射表一

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7
TIM5	TIM5_CH4 TIM5_TRIG	TIM5_CH3 TIM5_UP		TIM5_CH2	TIM5_CH1		
TIM6			TIM6_UP				
TIM7				TIM7_UP			
TIM8	TIM8_CH3 TIM8_UP	TIM8_CH4 TIM8_TRIG TIM8_COM	TIM8_CH1		TIM8_CH2		
TIM9						TIM9_UP	TIM9_CH1
TIM10						TIM10_CH4	TIM10_TRIG TIM10_COM
UART4			USART4_RX		UART4_TX		
UART5		USART5_RX		UART5_TX			
UART6						UART6_TX	UART6_RX
UART7							
UART8							
SPI/I2S3	SPI/I2S3_RX	SPI/I2S3_TX					
SDIO				SDIO			
DAC1			DAC1				
DAC2				DAC2			

表 11-4 DMA2 各通道外设映射表二

外设	通道 8	通道 9	通道 10	通道 11
TIM5				
TIM6				
TIM7				
TIM8				
TIM9	TIM9_CH4	TIM9_CH2	TIM9_TRIG TIM9_COM	TIM9_CH3
TIM10	TIM10_CH1	TIM10_CH3	TIM10_CH2	TIM10_UP
UART4				
UART5				
UART6				
UART7	UART7_TX	UART7_RX		
UART8			UART8_TX	UART8_RX

SPI/I2S3				
SDIO				
DAC1				
DAC2				

青稞 V4B MCU（包括 CH32V20x_D6）以及 ARM®Cortex™-M3 MCU(包括 CH32F20x_D6），DMA 控制器提供 8 个通道，每个通道对应多个外设请求，通过设置相应外设寄存器中对应 DMA 控制位，可以独立的开启或关闭各个外设的 DMA 功能，具体对应关系如下。

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7	通道 8
ADC1	ADC1							
SPI1		SPI1_RX	SPI1_TX					
SPI2				SPI2_RX	SPI2_TX			
USART1				USART1_TX	USART1_RX			
USART2						USART2_RX	USART2_TX	
USART3		USART3_TX	USART3_RX					
USART4	USART4_TX							USART4_RX
I2C1						I2C1_TX	I2C1_RX	
I2C2				I2C2_TX	I2C2_RX			
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3		
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4	
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP			TIM3_CH1 TIM3_TRIG		
TIM4	TIM4_CH1			TIM4_CH2	TIM4_CH3		TIM4_UP	

青稞 V4C MCU(包括 CH32V20x_D8W、CH32V20x_D8)以及 ARM®Cortex™-M3 MCU(包括 CH32F20x_D8W)，DMA 控制器提供 8 个通道，每个通道对应多个外设请求，通过设置相应外设寄存器中对应 DMA 控制位，可以独立的开启或关闭各个外设的 DMA 功能，具体对应关系如下。

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7	通道 8
ADC1	ADC1							
SPI1		SPI1_RX	SPI1_TX					
SPI2				SPI2_RX	SPI2_TX			
USART1				USART1_TX	USART1_RX			
USART2						USART2_RX	USART2_TX	
USART3		USART3_TX	USART3_RX					
USART4	USART4_TX							USART4_RX
I2C1						I2C1_TX	I2C1_RX	
I2C2				I2C2_TX	I2C2_RX			
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3		

TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4	
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP			TIM3_CH1 TIM3_TRIG		
TIM4	TIM4_CH1			TIM4_CH2	TIM4_CH3		TIM4_UP	
TIM5	TIM5_CH2		TIM5_CH3			TIM5_CH4	TIM5_CH1 TIM5_TRIG	TIM5_UP

11.3 寄存器描述

表 11-5 DMA1 相关寄存器列表

名称	访问地址	描述	复位值
R32_DMA1_INTFR	0x40020000	DMA1 中断状态寄存器	0x00000000
R32_DMA1_INTFCR	0x40020004	DMA1 中断标志清除寄存器	0x00000000
R32_DMA1_CFGR1	0x40020008	DMA1 通道 1 配置寄存器	0x00000000
R32_DMA1_CNTR1	0x4002000C	DMA1 通道 1 传输数据数目寄存器	0x00000000
R32_DMA1_PADDR1	0x40020010	DMA1 通道 1 外设地址寄存器	0x00000000
R32_DMA1_MADDR1	0x40020014	DMA1 通道 1 存储器地址寄存器	0x00000000
R32_DMA1_CFGR2	0x4002001C	DMA1 通道 2 配置寄存器	0x00000000
R32_DMA1_CNTR2	0x40020020	DMA1 通道 2 传输数据数目寄存器	0x00000000
R32_DMA1_PADDR2	0x40020024	DMA1 通道 2 外设地址寄存器	0x00000000
R32_DMA1_MADDR2	0x40020028	DMA1 通道 2 存储器地址寄存器	0x00000000
R32_DMA1_CFGR3	0x40020030	DMA1 通道 3 配置寄存器	0x00000000
R32_DMA1_CNTR3	0x40020034	DMA1 通道 3 传输数据数目寄存器	0x00000000
R32_DMA1_PADDR3	0x40020038	DMA1 通道 3 外设地址寄存器	0x00000000
R32_DMA1_MADDR3	0x4002003C	DMA1 通道 3 存储器地址寄存器	0x00000000
R32_DMA1_CFGR4	0x40020044	DMA1 通道 4 配置寄存器	0x00000000
R32_DMA1_CNTR4	0x40020048	DMA1 通道 4 传输数据数目寄存器	0x00000000
R32_DMA1_PADDR4	0x4002004C	DMA1 通道 4 外设地址寄存器	0x00000000
R32_DMA1_MADDR4	0x40020050	DMA1 通道 4 存储器地址寄存器	0x00000000
R32_DMA1_CFGR5	0x40020058	DMA1 通道 5 配置寄存器	0x00000000
R32_DMA1_CNTR5	0x4002005C	DMA1 通道 5 传输数据数目寄存器	0x00000000
R32_DMA1_PADDR5	0x40020060	DMA1 通道 5 外设地址寄存器	0x00000000
R32_DMA1_MADDR5	0x40020064	DMA1 通道 5 存储器地址寄存器	0x00000000
R32_DMA1_CFGR6	0x4002006C	DMA1 通道 6 配置寄存器	0x00000000
R32_DMA1_CNTR6	0x40020070	DMA1 通道 6 传输数据数目寄存器	0x00000000
R32_DMA1_PADDR6	0x40020074	DMA1 通道 6 外设地址寄存器	0x00000000
R32_DMA1_MADDR6	0x40020078	DMA1 通道 6 存储器地址寄存器	0x00000000
R32_DMA1_CFGR7	0x40020080	DMA1 通道 7 配置寄存器	0x00000000
R32_DMA1_CNTR7	0x40020084	DMA1 通道 7 传输数据数目寄存器	0x00000000
R32_DMA1_PADDR7	0x40020088	DMA1 通道 7 外设地址寄存器	0x00000000
R32_DMA1_MADDR7	0x4002008C	DMA1 通道 7 存储器地址寄存器	0x00000000
R32_DMA1_CFGR8	0x40020094	DMA1 通道 8 配置寄存器	0x00000000

R32_DMA1_CNTR8	0x40020098	DMA1 通道 8 传输数据数目寄存器	0x00000000
R32_DMA1_PADDR8	0x4002009C	DMA1 通道 8 外设地址寄存器	0x00000000
R32_DMA1_MADDR8	0x400200A0	DMA1 通道 8 存储器地址寄存器	0x00000000

表 11-6 DMA2 相关寄存器列表

名称	访问地址	描述	复位值
R32_DMA2_INTFR	0x40020400	DMA2 中断状态寄存器	0x00000000
R32_DMA2_INTFCR	0x40020404	DMA2 中断标志清除寄存器	0x00000000
R32_DMA2_CFGR1	0x40020408	DMA2 通道 1 配置寄存器	0x00000000
R32_DMA2_CNTR1	0x4002040C	DMA2 通道 1 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR1	0x40020410	DMA2 通道 1 外设地址寄存器	0x00000000
R32_DMA2_MADDR1	0x40020414	DMA2 通道 1 存储器地址寄存器	0x00000000
R32_DMA2_CFGR2	0x4002041C	DMA2 通道 2 配置寄存器	0x00000000
R32_DMA2_CNTR2	0x40020420	DMA2 通道 2 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR2	0x40020424	DMA2 通道 2 外设地址寄存器	0x00000000
R32_DMA2_MADDR2	0x40020428	DMA2 通道 2 存储器地址寄存器	0x00000000
R32_DMA2_CFGR3	0x40020430	DMA2 通道 3 配置寄存器	0x00000000
R32_DMA2_CNTR3	0x40020434	DMA2 通道 3 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR3	0x40020438	DMA2 通道 3 外设地址寄存器	0x00000000
R32_DMA2_MADDR3	0x4002043C	DMA2 通道 3 存储器地址寄存器	0x00000000
R32_DMA2_CFGR4	0x40020444	DMA2 通道 4 配置寄存器	0x00000000
R32_DMA2_CNTR4	0x40020448	DMA2 通道 4 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR4	0x4002044C	DMA2 通道 4 外设地址寄存器	0x00000000
R32_DMA2_MADDR4	0x40020450	DMA2 通道 4 存储器地址寄存器	0x00000000
R32_DMA2_CFGR5	0x40020458	DMA2 通道 5 配置寄存器	0x00000000
R32_DMA2_CNTR5	0x4002045C	DMA2 通道 5 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR5	0x40020460	DMA2 通道 5 外设地址寄存器	0x00000000
R32_DMA2_MADDR5	0x40020464	DMA2 通道 5 存储器地址寄存器	0x00000000
R32_DMA2_CFGR6	0x4002046C	DMA2 通道 6 配置寄存器	0x00000000
R32_DMA2_CNTR6	0x40020470	DMA2 通道 6 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR6	0x40020474	DMA2 通道 6 外设地址寄存器	0x00000000
R32_DMA2_MADDR6	0x40020478	DMA2 通道 6 存储器地址寄存器	0x00000000
R32_DMA2_CFGR7	0x40020480	DMA2 通道 7 配置寄存器	0x00000000
R32_DMA2_CNTR7	0x40020484	DMA2 通道 7 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR7	0x40020488	DMA2 通道 7 外设地址寄存器	0x00000000
R32_DMA2_MADDR7	0x4002048C	DMA2 通道 7 存储器地址寄存器	0x00000000
R32_DMA2_CFGR8	0x40020490	DMA2 通道 8 配置寄存器	0x00000000
R32_DMA2_CNTR8	0x40020494	DMA2 通道 8 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR8	0x40020498	DMA2 通道 8 外设地址寄存器	0x00000000
R32_DMA2_MADDR8	0x4002049C	DMA2 通道 8 存储器地址寄存器	0x00000000
R32_DMA2_CFGR9	0x400204A0	DMA2 通道 9 配置寄存器	0x00000000
R32_DMA2_CNTR9	0x400204A4	DMA2 通道 9 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR9	0x400204A8	DMA2 通道 9 外设地址寄存器	0x00000000
R32_DMA2_MADDR9	0x400204AC	DMA2 通道 9 存储器地址寄存器	0x00000000
R32_DMA2_CFGR10	0x400204B0	DMA2 通道 10 配置寄存器	0x00000000

R32_DMA2_CNTR10	0x400204B4	DMA2 通道 10 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR10	0x400204B8	DMA2 通道 10 外设地址寄存器	0x00000000
R32_DMA2_MADDR10	0x400204BC	DMA2 通道 10 存储器地址寄存器	0x00000000
R32_DMA2_CFGR11	0x400204C0	DMA2 通道 11 配置寄存器	0x00000000
R32_DMA2_CNTR11	0x400204C4	DMA2 通道 11 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR11	0x400204C8	DMA2 通道 11 外设地址寄存器	0x00000000
R32_DMA2_MADDR11	0x400204CC	DMA2 通道 11 存储器地址寄存器	0x00000000
R32_DMA2_EXTEM_INTFR	0x400204D0	DMA2 扩展中断状态寄存器	0x00000000
R32_DMA2_EXTEM_INTFCR	0x400204D4	DMA2 扩展中断标志清除寄存器	0x00000000

11.3.1 DMAx 中断状态寄存器（DMAx_INTFR）（x=1/2）

偏移地址：0x04 + (x-1)*0x400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1

位	名称	访问	描述	复位值
[31:28]	Reserved	RO	保留。	0
27/23/19/15/11/7/3	TEIFx	RO	通道 x 的传输错误标志（x=1/2/3/4/5/6/7/8）： 1：在通道 x 上发生了传输错误； 0：在通道 x 上没有传输错误。 硬件置位，软件写 CTEIFx 位清除此标志。	0
26/22/18/14/10/6/2	HTIFx	RO	通道 x 的传输过半标志（x=1/2/3/4/5/6/7/8）： 1：在通道 x 上产生了传输过半事件； 0：在通道 x 上没有传输过半。 硬件置位，软件写 CHTIFx 位清除此标志。	0
25/21/17/13/9/5/1	TCIFx	RO	通道 x 的传输完成标志（x=1/2/3/4/5/6/7/8）： 1：在通道 x 上产生了传输完成事件； 0：在通道 x 上没有传输完成事件。 硬件置位，软件写 CTCIFx 位清除此标志。	0
24/20/16/12/8/4/0	GIFx	RO	通道 x 的全局中断标志（x=1/2/3/4/5/6/7/8）： 1：在通道 x 上产生了 TEIFx 或 HTIFx 或 TCIFx； 0：在通道 x 上没有发生 TEIFx 或 HTIFx 或 TCIFx。 硬件置位，软件写 CGIFx 位清除此标志。	0

注：通道 8 适用于 CH32V20x_D8、CH32V20x_D8W、CH32F20x_D8W、CH32F20x_D6、CH32V20x_D6。

11.3.2 DMAx 中断标志清除寄存器（DMAx_INTFCR）（x=1/2）

偏移地址：0x04 + (x-1)*0x400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1

位	名称	访问	描述	复位值
[31:28]	Reserved	RO	保留。	0
27/23/19/15/11/7/3	CTEIFx	WO	清除通道 x 的传输错误标志（x=1/2/3/4/5/6/8）： 1：清除 DMA_INTFR 寄存器中的 TEIFx 标志； 0：无作用。	0
26/22/18/14/10/6/2	CHTIFx	WO	清除通道 x 的传输过半标志（x=1/2/3/4/5/6/8）： 1：清除 DMA_INTFR 寄存器中的 HTIFx 标志； 0：无作用。	0
25/21/17/13/9/5/1	CTCIFx	WO	清除通道 x 的传输完成标志（x=1/2/3/4/5/6/7/8）： 1：清除 DMA_INTFR 寄存器中的 TCIFx 标志； 0：无作用。	0
24/20/16/12/8/4/0	CGIFx	WO	清除通道 x 的全局中断标志（x=1/2/3/4/5/6/7/8）： 1：清除 DMA_INTFR 寄存器中的 TEIFx/HTIFx/TCIFx/GIFx 标志； 0：无作用。	0

注：通道 8 适用于 CH32V20x_D8、CH32V20x_D8W、CH32F20x_D8W、CH32F20x_D6、CH32V20x_D6。

11.3.3 DMAy 通道 x 配置寄存器（DMAy_CFGRx）（x=1/2/3/4/5/6/7/8，y=0/1）

偏移地址：0x08 + (x-1)*20 + (y-1)*0x400

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN

位	名称	访问	描述	复位值
[31:15]	Reserved	RO	保留。	0
14	MEM2MEM	RW	存储器到存储器模式使能： 1：使能存储器到存储器数据传输模式； 0：非存储器到存储器数据传输。	0
[13:12]	PL	RW	通道优先级设置： 00：低；　　01：中； 10：高；　　11：最高。	0
[11:10]	MSIZE	RW	存储器地址数据宽度设置： 00：8 位；　　01：16 位； 10：32 位；　　11：保留。	0
[9:8]	PSIZE	RW	外设地址数据宽度设置： 00：8 位；　　01：16 位； 10：32 位；　　11：保留。	0
7	MINC	RW	存储器地址增量递增模式使能： 1：使能存储器地址增量递增操作； 0：存储器地址保持不变操作。	0
6	PINC	RW	外设地址增量递增模式使能： 1：使能外设地址增量递增操作；	0

			0：外设地址保持不变操作。	
5	CIRC	RW	DMA 通道循环模式使能： 1：使能循环操作； 0：执行单次操作。	0
4	DIR	RW	数据传输方向： 1：从存储器读； 0：从外设读。	0
3	TEIE	RW	传输错误中断使能控制： 1：使能传输错误中断； 0：禁止传输错误中断。	0
2	HTIE	RW	传输过半中断使能控制： 1：使能传输过半中断； 0：禁止传输过半中断。	0
1	TCIE	RW	传输完成中断使能控制： 1：使能传输完成中断； 0：禁止传输完成中断。	0
0	EN	RW	通道使能控制： 1：通道开启；　　0：通道关闭。 发生 DMA 传输错误时，硬件自动将此位清 0，关闭通道。	0

注：通道 8 适用于 CH32V20x_D8、CH32V20x_D8W、CH32F20x_D8W、CH32F20x_D6、CH32V20x_D6。

11.3.4 DMAy 通道 x 传输数据数目寄存器（DMAy_CNTRx）（x=1/2/3/4/5/6/7/8，y=0/1）

偏移地址：0x0C + (x-1)*20 + (y-1)*0x400

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	NDT	RW	数据传输数量，范围 0-65535。 这个寄存器只能在通道不工作（DMA_CFGRx 的 EN=0）时写入。通道开启后该寄存器变为只读，指示剩余的待传输数目（寄存器内容在每次 DMA 传输后递减）。 在通道为循环模式下，寄存器的内容将被自动重新加载为之前配置的数值。	0

注：此寄存器只能在 EN=0 时更改；EN=1 时，为只读寄存器，表示当前待传输数目。当寄存器内容为 0 时，无论通道是否开启，都不会发生任何数据传输。通道 8 适用于 CH32V20x_D8、CH32V20x_D8W、CH32F20x_D8W、CH32F20x_D6、CH32V20x_D6。

11.3.5 DMAy 通道 x 外设地址寄存器（DMAy_PADDRx）（x=1/2/3/4/5/6/7/8）

偏移地址：0x10 + (x-1)*20 + (y-1)*0x400

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

位	名称	访问	描述	复位值
[31:0]	PA	RW	外设基地址，作为外设数据传输的源或目标地址。 当 PSIZE[1:0]=‘01’（16 位），模块自动忽略 bit0，操作地址自动 2 字节对齐；当 PSIZE[1:0]=‘10’（32 位），模块自动忽略 bit[1:0]，操作地址自动 4 字节对齐。	0

注：此寄存器只能在 EN=0 时更改，EN=1 时不可写。通道 8 适用于 CH32V20x_D8、CH32V20x_D8W、CH32F20x_D8W、CH32F20x_D6、CH32V20x_D6。

11.3.6 DMAy 通道 x 存储器地址寄存器（DMAy_MADDRx）（x=1/2/3/4/5/6/7/8）

偏移地址：0x14 + (x-1)*20 + (y-1)*0x400

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

位	名称	访问	描述	复位值
[31:0]	MA	RW	存储器数据地址，作为数据传输的源或目标地址。 当 MSIZE[1:0]=‘01’（16 位），模块自动忽略 bit0，操作地址自动 2 字节对齐；当 MSIZE[1:0]=‘10’（32 位），模块自动忽略 bit[1:0]，操作地址自动 4 字节对齐。	0

注：此寄存器只能在 EN=0 时更改，EN=1 时不可写。通道 8 适用于 CH32V20x_D8、CH32V20x_D8W、CH32F20x_D8W、CH32F20x_D6、CH32V20x_D6。

11.3.7 DMA2 通道 x 配置寄存器（DMA2_CFGRx）（x=8/9/10/11）

偏移地址：0x490 + (x-8)*16

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN

位	名称	访问	描述	复位值
[31:15]	Reserved	RO	保留。	0
14	MEM2MEM	RW	存储器到存储器模式使能： 1：使能存储器到存储器数据传输模式； 0：非存储器到存储器数据传输。	0
[13:12]	PL	RW	通道优先级设置： 00：低；　　01：中； 10：高；　　11：最高。	0
[11:10]	MSIZE	RW	存储器地址数据宽度设置： 00：8 位；　　01：16 位； 10：32 位；　　11：保留。	0
[9:8]	PSIZE	RW	外设地址数据宽度设置：	0

			00：8 位；　　01：16 位； 10：32 位；　　11：保留。	
7	MINC	RW	存储器地址增量递增模式使能： 1：使能存储器地址增量递增操作； 0：存储器地址保持不变操作。	0
6	PINC	RW	外设地址增量递增模式使能： 1：使能外设地址增量递增操作； 0：外设地址保持不变操作。	0
5	CIRC	RW	DMA 通道循环模式使能： 1：使能循环操作； 0：执行单次操作。	0
4	DIR	RW	数据传输方向： 1：从存储器读； 0：从外设读。	0
3	TEIE	RW	传输错误中断使能控制： 1：使能传输错误中断； 0：禁止传输错误中断。	0
2	HTIE	RW	传输过半中断使能控制： 1：使能传输过半中断； 0：禁止传输过半中断。	0
1	TCIE	RW	传输完成中断使能控制： 1：使能传输完成中断； 0：禁止传输完成中断。	0
0	EN	RW	通道使能控制： 1：通道开启；　　0：通道关闭。 发生 DMA 传输错误时，硬件自动将此位清 0，关闭通道。	0

注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。

11.3.8 DMA2 通道 x 传输数据数目寄存器（DMA2_CNTRx）（x=8/9/10/11）

偏移地址：0x494 + (x-8)*16

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	NDT	RW	数据传输数量，范围 0-65535。 这个寄存器只能在通道不工作（DMA_CFGRx 的 EN=0）时写入。通道开启后该寄存器变为只读，指示剩余的待传输数目（寄存器内容在每次 DMA 传输后递减）。在通道为循环模式下，寄存器的内容将被自动重新加载为之前配置的数值。	0

注：此寄存器只能在 EN=0 时更改；EN=1 时，为只读寄存器，表示当前待传输数目。当寄存器内容为 0 时，无论通道是否开启，都不会发生任何数据传输，适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。

11.3.9 DMA2 通道 x 外设地址寄存器（DMA2_PADDRx）（x=8/9/10/11）

偏移地址：0x498 + (x-8)*16

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

位	名称	访问	描述	复位值
[31:0]	PA	RW	外设基地址，作为外设数据传输的源或目标地址。 当 PSIZE[1:0]=‘01’（16 位），模块自动忽略 bit0，操作地址自动 2 字节对齐；当 PSIZE[1:0]=‘10’（32 位），模块自动忽略 bit[1:0]，操作地址自动 4 字节对齐。	0

注：此寄存器只能在 EN=0 时更改，EN=1 时不可写，适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。

11.3.10 DMA2 通道 x 存储器地址寄存器（DMA2_MADDRx）（x=8/9/10/11）

偏移地址：0x49C + (x-8)*16

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

位	名称	访问	描述	复位值
[31:0]	MA	RW	存储器数据地址，作为数据传输的源或目标地址。 当 MSIZE[1:0]=‘01’（16 位），模块自动忽略 bit0，操作地址自动 2 字节对齐；当 MSIZE[1:0]=‘10’（32 位），模块自动忽略 bit[1:0]，操作地址自动 4 字节对齐。	0

注：此寄存器只能在 EN=0 时更改，EN=1 时不可写，适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。

11.3.11 DMA2 扩展中断状态寄存器（DMA2_EXTEM_INTFR）

偏移地址：0x4D0

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

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
15/11/7/3	TEIFx	RO	通道 x 的传输错误标志（x=8/9/10/11）： 1：在通道 x 上发生了传输错误； 0：在通道 x 上没有传输错误。 硬件置位，软件写 CTEIFx 位清除此标志。	0
14/10/6/2	HTIFx	RO	通道 x 的传输过半标志（x=8/9/10/11）：	0

			1：在通道 x 上产生了传输过半事件； 0：在通道 x 上没有传输过半。 硬件置位，软件写 CHTIFx 位清除此标志。	
13/9/5/1	TCIFx	RO	通道 x 的传输完成标志（x=8/9/10/11）： 1：在通道 x 上产生了传输完成事件； 0：在通道 x 上没有传输完成事件。 硬件置位，软件写 CTCIFx 位清除此标志。	0
12/8/4/0	GIFx	RO	通道 x 的全局中断标志（x=8/9/10/11）： 1：在通道 x 上产生了 TEIFx 或 HTIFx 或 TCIFx； 0：在通道 x 上没有发生 TEIFx 或 HTIFx 或 TCIFx。 硬件置位，软件写 CGIFx 位清除此标志。	0

注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。

11.3.12 DMA2 扩展中断标志清除寄存器（DMA2_EXTEM_INTFCR）

偏移地址：0x4D4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF11	CHTIF11	CTCIF11	CGIF11	CTEIF10	CHTIF10	CTCIF10	CGIF10	CTEIF9	CHTIF9	CTCIF9	CGIF9	CTEIF8	CHTIF8	CTCIF8	CGIF8

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
15/11/7/3	CTEIFx	WO	清除通道 x 的传输错误标志（x=8/9/10/11）： 1：清除 DMA_INTFR 寄存器中的 TEIFx 标志； 0：无作用。	0
14/10/6/2	CHTIFx	WO	清除通道 x 的传输过半标志（x=8/9/10/11）： 1：清除 DMA_INTFR 寄存器中的 HTIFx 标志； 0：无作用。	0
13/9/5/1	CTCIFx	WO	清除通道 x 的传输完成标志（x=8/9/10/11）： 1：清除 DMA_INTFR 寄存器中的 TCIFx 标志； 0：无作用。	0
12/8/4/0	CGIFx	WO	清除通道 x 的全局中断标志（x=8/9/10/11）： 1：清除 DMA_INTFR 寄存器中的 TEIFx/HTIFx/TCIFx/GIFx 标志； 0：无作用。	0

注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。

第 12 章 模拟/数字转换（ADC）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

ADC 模块包含 2 个 12 位的逐次逼近型的模拟数字转换器，最高 14MHz 的输入时钟。支持 16 个外部通道和 2 个内部信号源采样源。可完成通道的单次转换、连续转换，通道间自动扫描模式、间断模式、外部触发模式、双重采样等功能。可以通过模拟看门狗功能监测通道电压是否在阈值范围内。

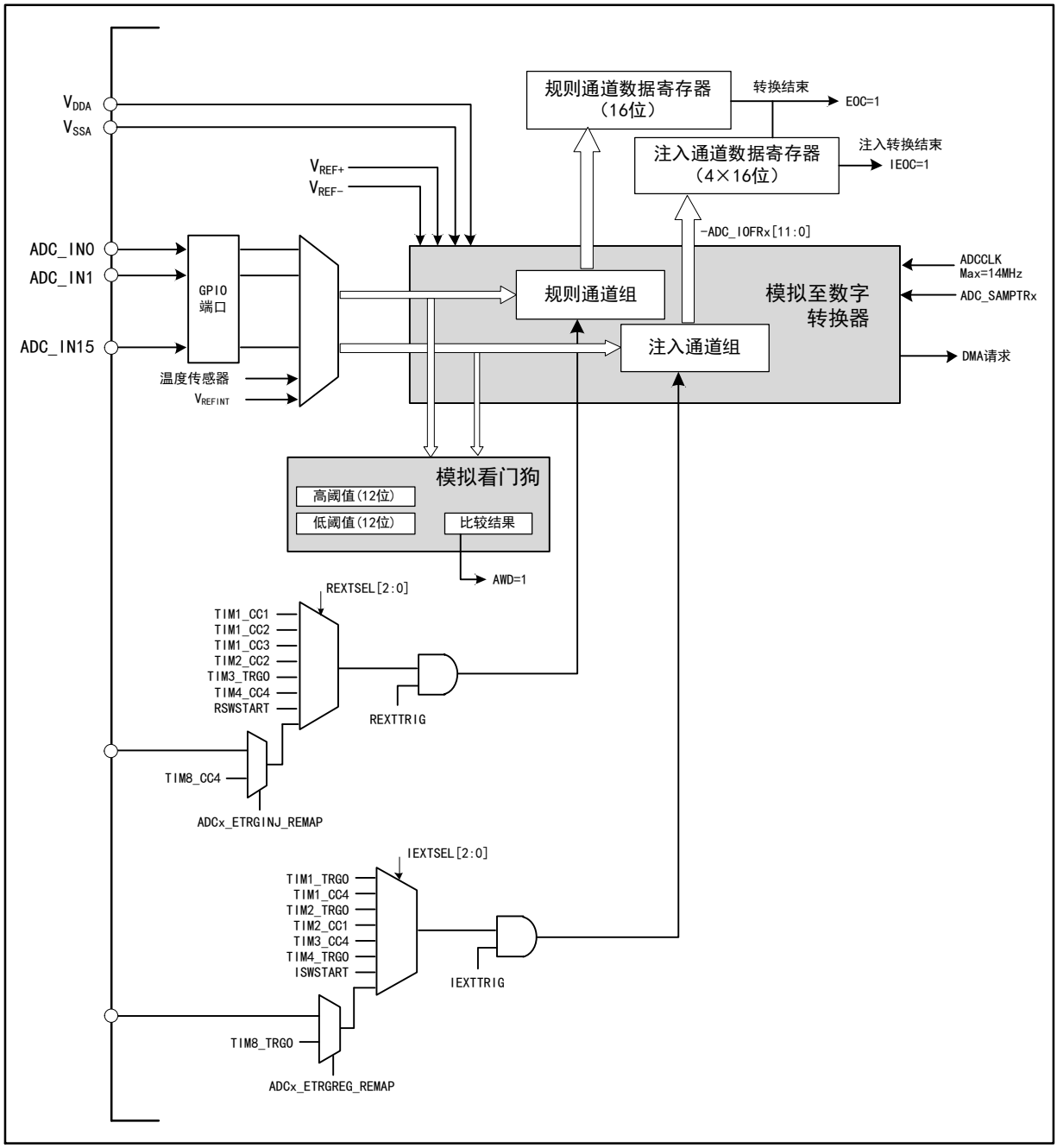
12.1 主要特性

- 12 位分辨率
- 支持 16 个外部通道和 2 个内部信号源采样
- 多通道的多种采样转换方式：单次、连续、扫描、触发、间断等
- 数据对齐模式：左对齐、右对齐
- 采样时间可按通道分别编程
- 规则转换和注入转换均支持外部触发
- 模拟看门狗监测通道电压，自校准功能
- 双重模式
- ADC 通道输入范围：$0 \leqslant V_{IN} \leqslant V_{DDA}$
- 输入增益可调，可实现小信号放大采样

12.2 功能描述

12.2.1 模块结构

图 12-1 ADC 模块框图



12.2.2 ADC 配置

1）模块上电

ADC_CTLR2 寄存器的 ADON 位为 1 表示 ADC 模块上电。当 ADC 模块从断电模式（ADON=0）下进入上电状态（ADON=1）后，需要延迟一段时间 t_{STAB} 用于模块稳定时间。之后再次写入 ADON 位为 1，用于作为软件启动 ADC 转换的启动信号。通过清除 ADON 位为 0，可以终止当前转换并将 ADC 模块置于断电模式，这个状态下，ADC 几乎不耗电。

2）采样时钟

模块的寄存器操作基于 PCLK2（APB2 总线）时钟，其转换单元的时钟基准 ADCCLK 与 PCLK2 同步，由 RCC_CFGR0 寄存器的 ADCPRE[1:0]域配置分频，最大不能超过 14MHz。

3）通道配置

ADC 模块提供了 18 个通道采样源，包括 16 个外部通道和 2 个内部通道。它们可以配置到两种转换组中：规则组和注入组。以实现任意多个通道上以任意顺序进行一系列转换构成的组转换。

转换组：

- 规则组：由多达 16 个转换组成。规则通道和它们的转换顺序在 ADC_RSQRx 寄存器中设置。规则组中转换的总数量应写入 ADC_RSQR1 寄存器的 RLEN[3:0]中。
- 注入组：由多达 4 个转换组成。注入通道和它们的转换顺序在 ADC_ISQR 寄存器中设置。注入组里的转换总数量应写入 ADC_ISQR 寄存器的 ILEN[1:0]中。

注：如果 ADC_RSQRx 或 ADC_ISQR 寄存器在转换期间被更改，当前的转换被终止，一个新的启动信号将发送到 ADC 以转换新选择的组。

2 个内部通道：

- 温度传感器：连接 ADC_IN16 通道，用来测量器件周围的温度(TA)。
- V_{REFINT} 内部参考电压：连接 ADC_IN17 通道。

4）校准

ADC 有一个内置自校准模式。经过校准环节可大幅减小因内部电容器组的变化而造成的精准度误差。在校准期间，在每个电容器上都会计算出一个误差修正码，用于消除在随后的转换中每个电容器上产生的误差。

通过写 ADC_CTLR2 寄存器的 RSTCAL 位置 1 初始化校准寄存器，等待 RSTCAL 硬件清 0 表示初始化完成。置位 CAL 位，启动校准功能，一旦校准结束，硬件会自动清除 CAL 位，将校准码存储到 ADC_RDATAR 中。之后可以开始正常的转换功能。建议在 ADC 模块上电时执行一次 ADC 校准。

注：启动校准前，必须保证 ADC 模块处于上电状态(ADON=1)超过至少两个 ADC 时钟周期。

5）可编程采样时间

ADC 使用若干个 ADCCLK 周期对输入电压采样，通道的采样周期数目可以通过 ADC_SAMPTR1 和 ADC_SAMPTR2 寄存器中的 SMPx[2:0]位更改。每个通道可以分别使用不同的时间采样。

总转换时间如下计算：

T_{CONV} = 采样时间 + $12.5T_{ADCCLK}$

ADC 的规则通道转换支持 DMA 功能。规则通道转换的值储存在一个仅有的数据寄存器 ADC_RDATAR 中，为防止连续转换多个规则通道时，没有及时取走 ADC_RDATAR 寄存器中的数据，可以开启 ADC 的 DMA 功能。硬件会在规则通道的转换结束时（EOC 置位）产生 DMA 请求，并将转换的数据从 ADC_RDATAR 寄存器传输到用户指定的目的地址。

对 DMA 控制器模块的通道配置完成后，写 ADC_CTLR2 寄存器的 DMA 位置 1，开启 ADC 的 DMA 功能。

注：注入组转换不支持 DMA 功能。

6）数据对齐

ADC_CTLR2 寄存器中的 ALIGN 位选择 ADC 转换后的数据存储对齐方式。12 位数据支持左对齐和右对齐模式。

规则组通道的数据寄存器 ADC_RDATAR 保存的是实际转换的 12 位数字值；而注入组通道的数据寄存器 ADC_IDATARx 是实际转换的数据减去 ADC_IOFRx 寄存器的定义的偏移量后写入的值，会存在正负情况，所以有符号位（SIGNB）。

图 12-2 数据左对齐

规则组数据寄存器

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

注入组数据寄存器

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

图 12-3 数据右对齐

规则组数据寄存器

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

注入组数据寄存器

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

12.2.3 外部触发源

ADC 转换的启动事件可以由外部事件触发。如果设置了 ADC_CTLR2 寄存器的 REXTTRI 或 IEXTTRIG 位，则可分别通过外部事件触发规则组或注入组通道的转换。此时，REXTSEL[2:0]和 IEXTSEL[2:0]位的配置决定规则组和注入组的外部事件源。

表 12-1 规则组通道的外部触发源

REXTSEL[2:0]	触发源	类型
000	定时器 1 的 CC1 事件	来自片上定时器的内部信号
001	定时器 1 的 CC2 事件	
010	定时器 1 的 CC3 事件	
011	定时器 2 的 CC2 事件	
100	定时器 3 的 TRGO 事件	
101	定时器 4 的 CC4 事件	
110	EXTI 线 11/TIM8_TRGO	来自外部引脚/内部定时器信号
111	RSWSTART 位置 1 软件触发	软件控制位

表 12-2 注入组通道的外部触发源

IEXTSEL[2:0]	触发源	类型
000	定时器 1 的 TRGO 事件	来自片上定时器的内部信号
001	定时器 1 的 CC4 事件	
010	定时器 2 的 TRGO 事件	
011	定时器 2 的 CC1 事件	
100	定时器 3 的 CC4 事件	
101	定时器 4 的 TRGO 事件	
110	EXTI 线 15/TIM8_CC4	来自外部引脚/内部定时器信号
111	ISWSTART 位置 1 软件触发	软件控制位

12.2.4 转换模式

表 12-3 转换模式组合

ADC_CTLR1 和 ADC_CTLR2 寄存器控制位					ADC 转换模式
CONT	SCAN	RDISCEN/IDISCEN	IAUTO	启动事件	
0	0	0	0	ADON 位置 1	单次单通道模式：某一规则通道执行单次转换。

				外部触发方式	单次单通道模式：规则通道或注入通道的某一通道执行单次转换。
	1	0	0	ADON 位置 1 或外部触发方式	单次扫描模式：按顺序对选中的所有规则组通道（ADC_RSQRx）或所有注入组通道（ADC_ISQR）逐个执行单次转换。 触发注入方式：当规则组通道转换过程中可以插入注入组通道所有转换，之后再继续规则组通道转换；但转换注入组通道时不会插入规则组通道转换。
			1	ADON 位置 1 或外部触发方式	单次扫描模式：按顺序对选中的所有规则组通道（ADC_RSQRx）或所有注入组通道（ADC_ISQR）逐个执行单次转换。 自动注入方式：在规则组通道转换完之后，注入组通道被自动转换。 *注：转换过程中不允许出现注入通道的外部触发信号。*
	0	1（RDISCEN 和 IDISCEN 不能同时为 1）	0	外部触发方式	单次间断模式：每次启动事件，执行一个短序列（DISCNUM[2:0]定义数量）的通道数量转换，直到所有选中通道转换完成才能重头开始。 *注：规则组和注入组选中此模式控制位分别为 IDISCEN 和 RDISCEN，不能同时为规则组和注入组配置间断模式，间断模式只能用于一组转换。*
			1	–	禁止此模式。
	1	1	X	–	无此模式。
1	0	0	0	ADON 位置 1 或外部触发方式	连续单通道/扫描模式：每轮结束后重复新一轮的转换，直到 CONT 清 0 才能终止。
	1	0	0		
			1		

注：规则组和注入组的外部触发事件是不一样的，而且‘ACON’位只能启动规则组通道转换，所以规则组和注入组通道转换的启动事件独立。

1）单次单通道转换模式

此模式下，对当前 1 个通道只执行一次转换。该模式对规则组或注入组中排序第 1 的通道执行转换，其中通过设置 ADC_CTLR2 寄存器的 ADON 位置 1（只适用于规则通道）启动也可通过外部触发启动（适用于规则通道或注入通道）。一旦选择通道的转换完成将：

如果转换的是规则组通道，则转换数据被储存在 16 位 ADC_RDATAR 寄存器中，EOC 标志被置位，如果设置了 EOCIE 位，将触发 ADC 中断。

如果转换的是注入组通道，则转换数据被储存在 16 位 ADC_IDATAR1 寄存器中，EOC 和 IEOC 标志被置位，如果设置了 IEOCIE 或 EOCIE 位，将触发 ADC 中断。

2）单次扫描模式转换

通过设置 ADC_CTLR1 寄存器的 SCAN 位为 1 进入 ADC 扫描模式。此模式用来扫描一组模拟通道，对被 ADC_RSQRx 寄存器（对规则通道）或 ADC_ISQR（对注入通道）选中的所有通道逐个执行单次转换，当前通道转换结束时，同一组的下一个通道被自动转换。

在扫描模式里，根据 IAUTO 位的状态，又分为触发注入方式和自动注入方式。

- 触发注入

IAUTO 位为 0，当在扫描规则组通道过程中，发生了注入组通道转换的触发事件，当前转换被复位，注入通道的序列被以单次扫描方式进行，在所有选中的注入组通道扫描转换结束后，恢复上次被中断的规则组通道转换。

如果当前在扫描注入组通道序列时，发生了规则通道的启动事件，注入组转换不会被中断，而是在注入序列转换完成后再执行规则序列的转换。

注：使用触发的注入转换时，必须保证触发事件的间隔长于注入序列。例如，完成注入序列的转换总体时间需要 28 个 ADCCLK，那么触发注入通道的事件间隔时间最小值为 29 个 ADCCLK。

- 自动注入

IAUTO 位为 1，在扫描完规则组选中的所有通道转换后，自动进行注入组选中通道的转换。这种方式可以用来转换 ADC_RSQRx 和 ADC_ISQR 寄存器中多达 20 个转换序列。

此模式里，必须禁止注入通道的外部触发（IEXTTRIG=0）。

注：对于 ADC 时钟预分频系数（ADCPRE[1:0]）为 4 至 8 时，当从规则转换切换到注入序列或从注入转换切换到规则序列时，会自动插入 1 个 ADCCLK 间隔；当 ADC 时钟预分频系数为 2 时，则有 2 个 ADCCLK 间隔的延迟。

3）单次间断模式转换

通过设置 ADC_CTLR1 寄存器的 RDISCEN 或 IDISCEN 位为 1 进入规则组或注入组的间断模式。此模式区别扫描模式中扫描完整的一组通道，而是将一组通道分为多个短序列，每次外部触发事件将执行一个短序列的扫描转换。

短序列的长度 n（n<=8）定义在 ADC_CTLR1 寄存器的 DISCNUM[2:0]中，当 RDISCEN 为 1，则是规则组的间断模式，待转换总长度定义在 ADC_RSQR1 寄存器的 RLEN[3:0]中；当 IDISCEN 为 1，则是注入组的间断模式，待转换总长度定义在 ADC_ISQR 寄存器的 ILEN[1:0]中。不能同时将规则组和注入组设置为间断模式。

规则组间断模式举例：

RDISCEN=1，DISCNUM[2:0]=3，RLEN[3:0]=8，待转换通道=1，3，2，5，8，4，10，6

第 1 次外部触发：转换序列为：1，3，2

第 2 次外部触发：转换序列为：5，8，4

第 3 次外部触发：转换序列为：10，6，同时产生 EOC 事件

第 4 次外部触发：转换序列为：1，3，2

注入组间断模式举例：

IDISCEN=1，DISCNUM[2:0]=1，ILEN[1:0]=3，待转换通道=1，3，2

第 1 次外部触发：转换序列为：1

第 2 次外部触发：转换序列为：3

第 3 次外部触发：转换序列为：2，同时产生 EOC 和 IEOC 事件

第 4 次外部触发：转换序列为：1

注：1. 当以间断模式转换一个规则组或注入组时，转换序列结束后不自动从头开始。当所有子组被转换完成，下一次触发事件启动第一个子组的转换。

2. 不能同时使用自动注入（IAUTO=1）和间断模式。

3. 不能同时为规则组和注入组设置间断模式，间断模式只能用于一组转换。

4）连续转换

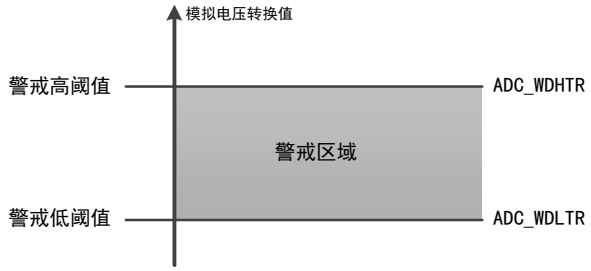
通过设置 ADC_CTLR2 寄存器的 CONT 位为 1，进入 ADC 的连续转换模式。此模式在前面 ADC 转换一结束马上就启动另一次转换，转换不会在选择组的最后一个通道上停止，而是再次从选择组的第一个通道继续转换。

启动事件包括外部触发事件和 ADON 位置 1。结合前面的单次模式中的几种转换方式，也包括连续单通道转换、连续扫描模式（触发注入或自动注入）转换。

12.2.5 模拟看门狗

如果被 ADC 转换的模拟电压低于低阀值或高于高阀值，AWD 模拟看门狗状态位被设置。阀值设置位于 ADC_WDHTR 和 ADC_WDLTR 寄存器的最低 12 个有效位中。通过设置 ADC_CTLR1 寄存器的 AWDIE 位以允许产生相应中断。

图 12-4 模拟看门狗阈值区



配置 ADC_CTLR1 寄存器的 AWDSGL、RAWDEN、IAWDEN 及 AWDCH[4:0]位选择模拟看门狗警戒的通道，具体关系见下表：

表 12-4 模拟看门狗通道选择

模拟看门狗警戒通道	ADC_CTLR1 寄存器控制位			
	AWDSGL	RAWDEN	IAWDEN	AWDCH[4:0]
不警戒	忽略	0	0	忽略
所有注入通道	0	0	1	忽略
所有规则通道	0	1	0	忽略
所有注入和规则通道	0	1	1	忽略
单一注入通道	1	0	1	决定通道编号
单一规则通道	1	1	0	决定通道编号
单一注入和规则通道	1	1	1	决定通道编号

12.2.6 温度传感器

模块内置温度传感器，连接 ADC_INT16 通道，通过 ADC 将传感器输出的电压转换成数字值来反馈器件周围温度，推荐设置采样时间是 17.1us。温度传感器输出的电压随温度线性变化，由于生产差异，其线性变化的曲线斜率和偏移有所不同，所以内部温度传感器更适合于检测温度的变化，而不是测量绝对的温度。如果需要测量精确的温度，应该使用一个外置的温度传感器。

通过设置 ADC_CTLR2 寄存器的 TSVREFE 位置 1，唤醒 ADC 内部采样通道，软件启动或外部触发启动 ADC 的温度传感器通道转换，读取数据结果（mV）。其中，数字值和温度(℃)换算公式如下：

温度(℃) = ((V$_{SENSE}$-V$_{25}$)/Avg_Slope)+25

V25：温度传感器在 25℃下的电压值

Avg_Slope：温度与 V$_{SENSE}$ 曲线的平均斜率（mV/℃）

参考数据手册电气特性章节中 V$_{25}$ 和 Avg_Slope 的实际值。

注：内部温度传感器上电（TSVREFE 位从 0 改为 1）需要一个建立时间，而 ADC 模块上电也需要一个建立时间（ADON 位从 0 改为 1），所以为了缩短等待时间，可以同时设置 ADON 和 TSVREFE 位。

12.2.7 双 ADC 模式

在有 2 个的 ADC 模块产品中，可以使 2 个 ADC 配合使用，实现双 ADC 模式。在双 ADC 模式中，ADC1 为主 ADC，ADC2 为从 ADC。通过配置 ADC1_CTLR1 中 DAULMODE[3:0]以选择模式，实现 ADC1 和 ADC2 交

替触发或同步触发转换。

注：双 ADC 模式中，选择外部触发事件触发时，用户必须使能主从 ADC 的外部触发使能且需要将主 ADC 设为相应的触发，从 ADC 设置为软件触发，防止不必要的触发使从 ADC 进行转换。

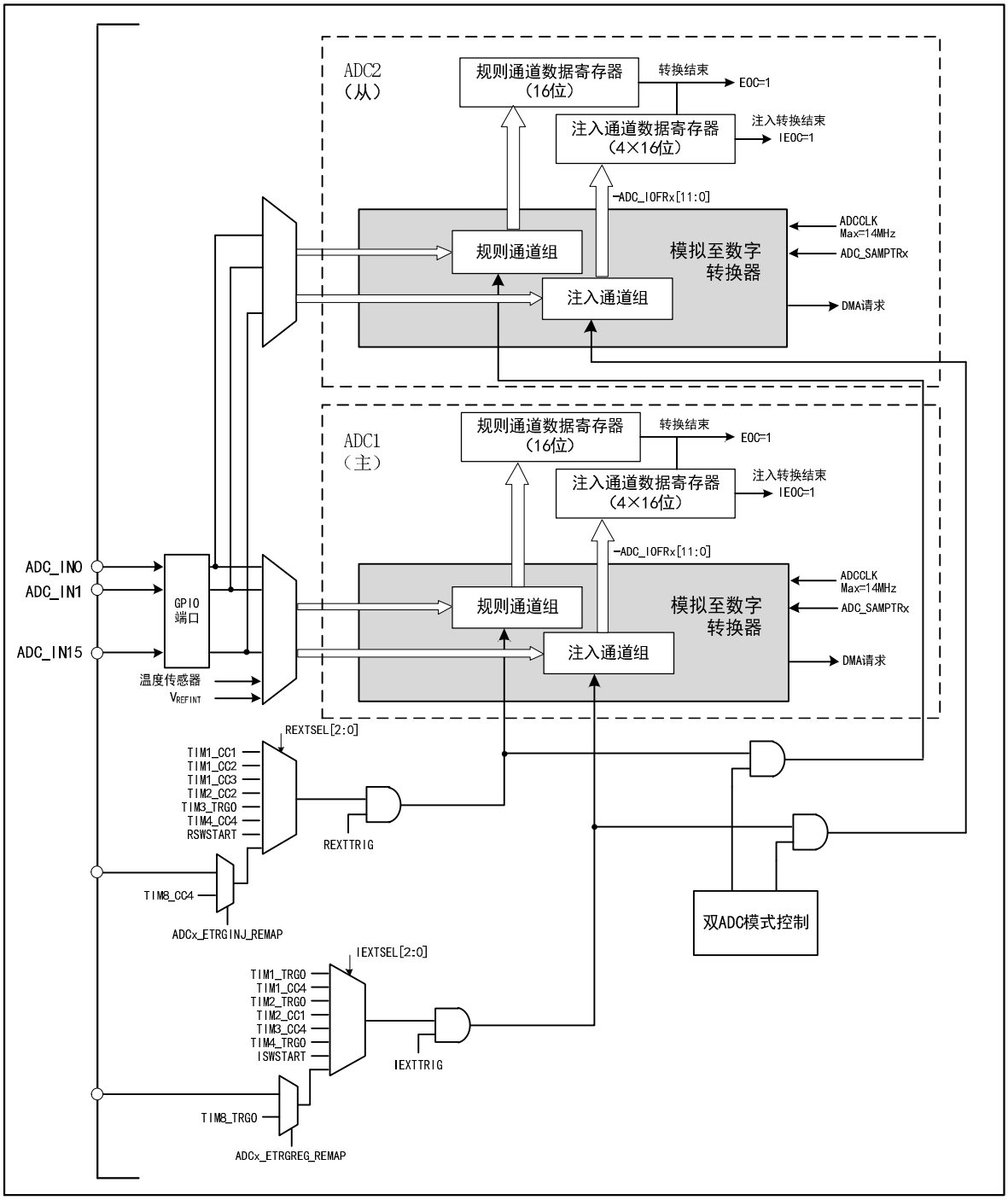
通过配置可以实现以下几种可能的模式：

- 独立模式
- 同步注入模式
- 同步规则模式
- 快速交替模式
- 慢速交替模式
- 交替触发模式
- 同步规则模式+同步注入模式
- 同步规则模式+交替触发模式
- 同步注入模式+快速交替模式
- 同步注入模式+慢速交替模式

注：1. 双 ADC 模式中，为了主数据寄存器能够读取从 ADC 的转换数据，需要使能 DMA 位。

2. 只有 ADC1 拥有 DMA 功能。而 ADC2 转化的数据只可以通过双 ADC 模式，利用 ADC1 的 DMA 功能传输。

图 12-5 双 ADC 框图



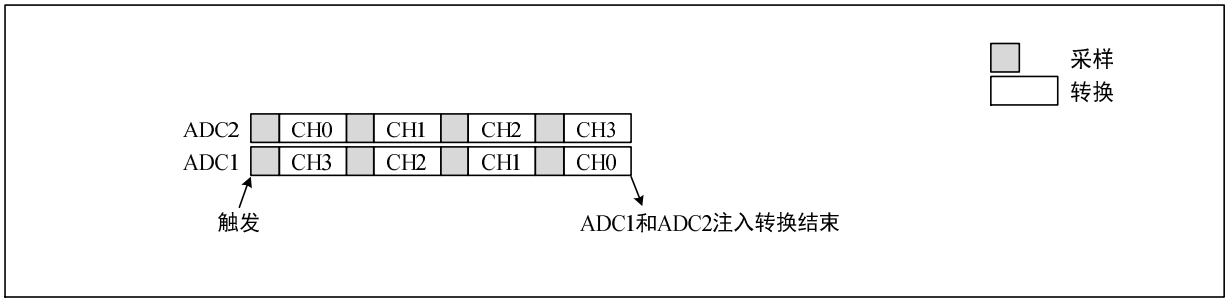
1）独立模式

此模式下，双 ADC 不同步工作，相互之间独立工作。

2）同步注入模式

此模式下用于转换一个注入通道组，设置 ADC1_CTLR1 中 IEXTSEL[2:0]，以选择触发源,同时它也将用于同步触发 ADC2。转换完成时，转换后的数据分别存储在 ADCx 的 ADC_IDATARx 中，且若使能任一 ADC 中断，转换结束后将产生 IEOC 中断。

图 12-6 4 通道同步注入转换



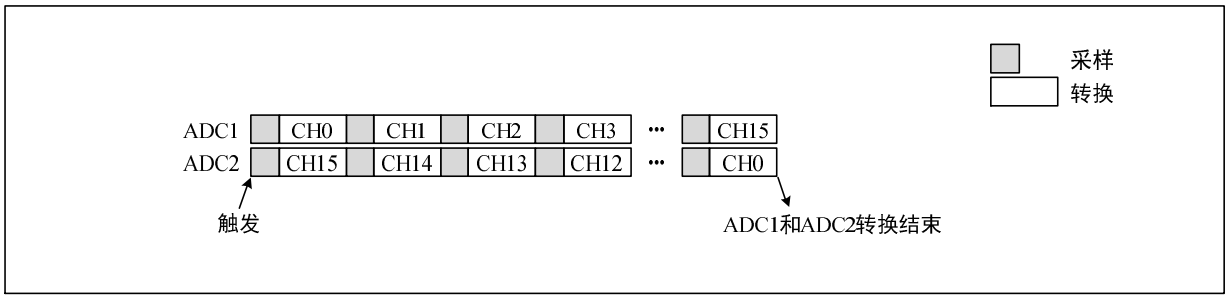
注：1. 同一时刻 ADC1 和 ADC2 的转换通道不应重合。

2. 同步模式下，ADC1 和 ADC2 应有相同时间长度的转换序列或两者之中较长的转换序列的时长小于触发时间间隔，以保证每次触发两个序列均能转换完成。

3）同步规则模式

此模式下用于转换规则通道序列，设置 ADC1_CTLR1 中 REXTSEL[2:0]，以选择触发源，同时它也将用于同步触发 ADC2。转换完成时，将产生一个 32 位 DMA 传输请求，将数据寄存器 ADC1_RDATAR 的内容传输到 SRAM 中，高 16 位包含 ADC2 转换数据，低 16 位包含 ADC1 转换数据。若使能任一 ADC 中断，将产生 EOC 中断。

图 12-7 16 通道同步规则转换



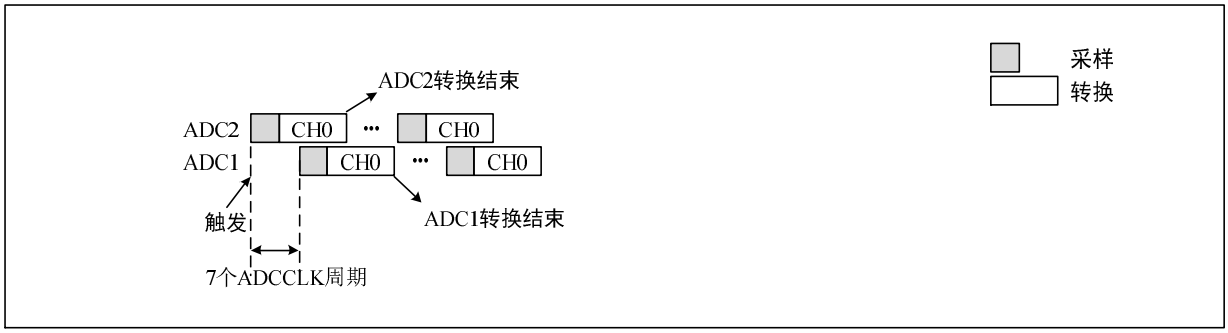
注：1. 同一时刻 ADC1 和 ADC2 的转换通道不应重合；

2. 同步模式下，ADC1 和 ADC2 应有相同时间长度的转换序列或两者之中较长的转换序列的时长小于触发时间间隔，以保证每次触发两个序列均能转换完成。

4）快速交替模式

此模式仅适用于规则通道（往往只有一个通道），设置 ADC1_CTLR1 中 REXTSEL[2:0]，以选择触发源，当触发产生后 ADC2 将立即启动转换，ADC1 在延迟 7 个 ADC 时钟周期后启动转换。如果均开启连续模式（CONT 被置位），两个 ADC 将对规则通道连续交替转换。如果使能中断，ADC1 将产生 EOC 中断，若同时使能 DMA，则将产生一个 32 位的 DMA 传输请求，将数据寄存器 ADC1_RDATAR 的内容传输到 SRAM 中，高 16 位包含 ADC2 转换数据，低 16 位包含 ADC1 转换数据。

图 12-8 单通道快速交替连续转换

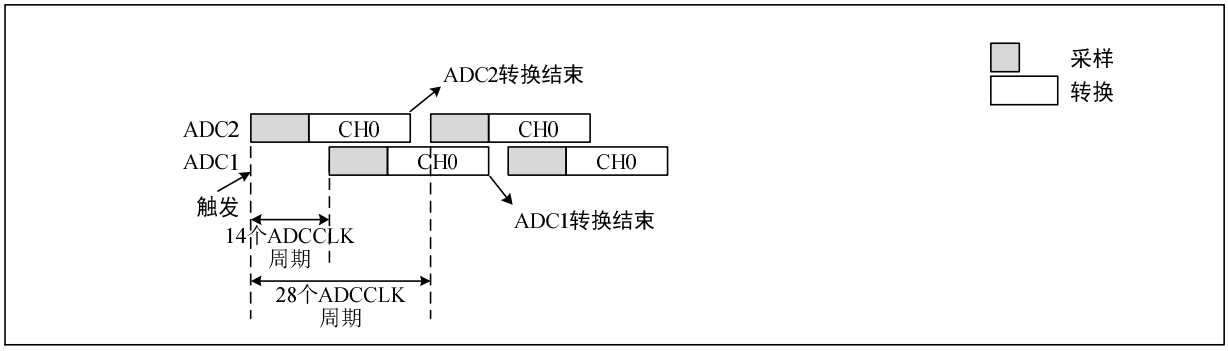


注：采样时间应小于 7 个 ADC 时钟周期，以避免 ADC1 和 ADC2 在采样同一个通道时，出现采样周期重合的问题。

5）慢速交替模式

此模式仅适用于规则通道且只能一个通道。设置 ADC1_CTLR1 中 REXTSEL[2:0]，以选择触发源，当触发产生后 ADC2 将立即启动转换，ADC1 在延迟 14 个 ADC 时钟周期后启动转换，再次延时 14 个 ADC 时钟周期后 ADC2 再次启动，如此往复循环。若使能中断，ADC1 将产生 EOC 中断，若同时使能 DMA，则将产生一个 32 位的 DMA 传输请求，将数据寄存器 ADC1_RDATAR 的内容传输到 SRAM 中，高 16 位包含 ADC2 转换数据，低 16 位包含 ADC1 转换数据。

图 12-9 单通道慢速交替转换

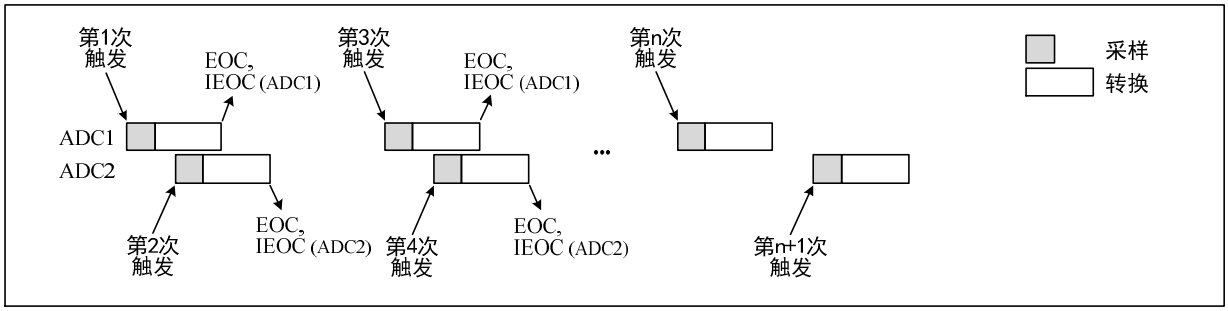


注：1. 采样时间应小于 14 个 ADC 时钟周期，以避免和下一次采样周期重合；
2. 28 个 ADC 时钟周期后自动启动新的 ADC2 转换；
3. 不需要设置 CONT 位。

6）交替触发模式

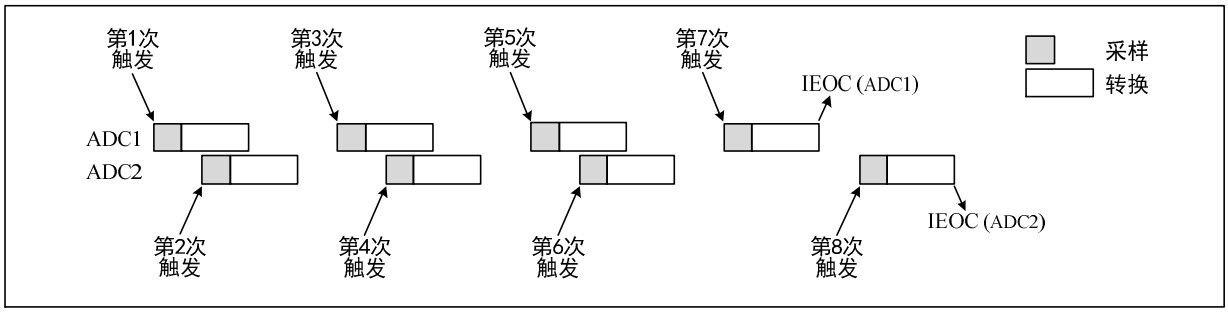
此模式仅适用于注入通道组，设置 ADC1_CTLR1 中 REXTSEL[2:0]，以选择触发源。当第一次触发事件发生时，ADC1 上所有的注入通道被转换，当第二次触发事件发生时，ADC2 上所有的注入通道被转换。依次循环往复，若使能中断，则当 ADC1 所有注入通道转换完成后产生 IEOC 中断，ADC2 所有注入通道转换完成后产生 IEOC 中断。

图 12-10 每个 ADC 注入通道组交替触发转换



若同时使用注入间断模式，则当第一次触发事件发生时，ADC1 上的第一个注入通道被转换，当第二次触发事件发生时，ADC2 上第一个注入通道被转换。以此类推，往复循环。同时若使能中断，则当 ADC1 所有注入通道转换完成后，产生 IEOC 中断，当 ADC2 所有注入通道转换完成后产生 IEOC 中断。

图 12-11 间断模式下每个 ADC 注入通道交替触发转换



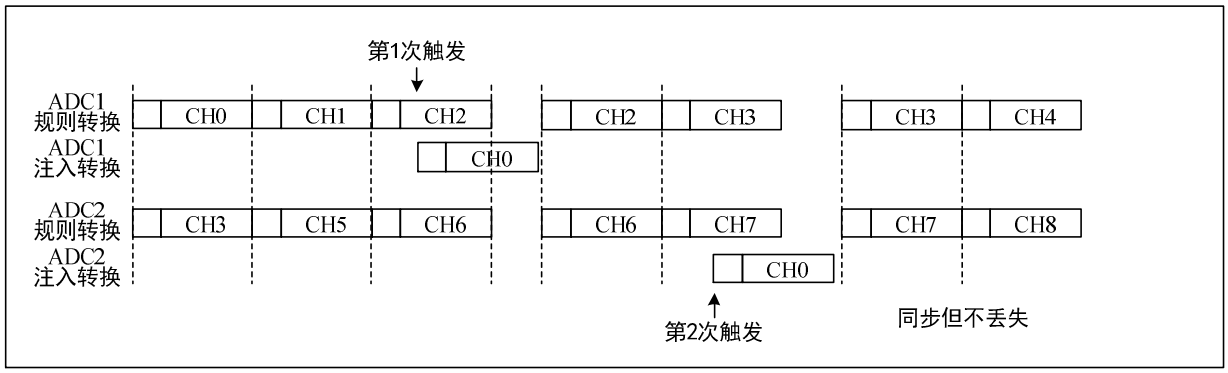
7）同步规则模式+同步注入模式

此模式下规则组的同步转换可以打断，以启动注入组的同步转换。同时此模式下需要保准转换具有相同时间长度的序列，或保证触发时间间隔比两个序列中较长序列的时间长。

8）同步规则模式+交替触发模式

此模式下规则组的同步转换可以被打断，以启动注入组的交替触发转换。当发生注入事件时，交替触发转换立即启动。如果同步规则正在转换，则所有 ADC 的规则转换被停止，在注入转换结束后同步恢复。

图 12-12 同步规则模式下交替触发注入通道转换

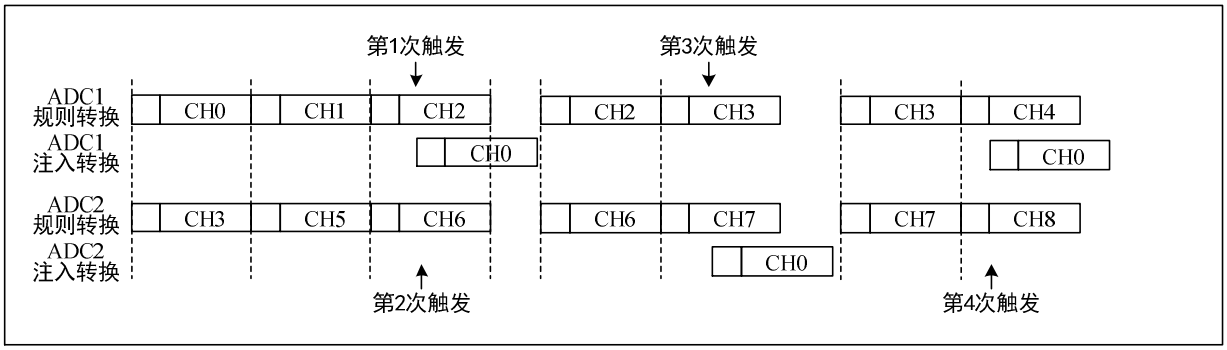


注：此模式下需要保准转换具有相同时间长度的序列，或者保证触发时间间隔比两个序列中较长

序列的时间长。

如果注入触发事件发生在中断了规则转换的注入转换期间，则这个触发事件将会被忽略，例如下图描述的第 2 次触发的情况。

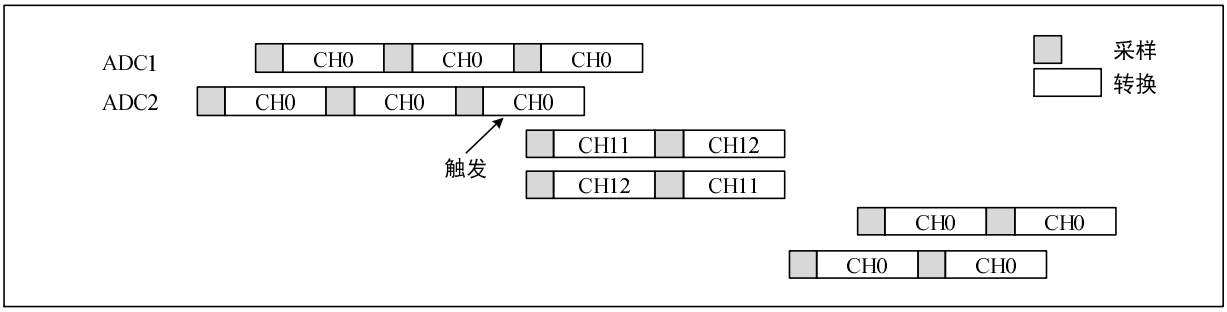
图 12-13 触发事件发生在注入转换期间



9）同步注入模式+交替模式

此模式下，注入转换可以打断交替转换。发生注入事件时，交替转换被打断，注入转换启动，在注入转换结束后，交替转换被恢复。

图 12-14 交替转换下触发注入组转换



注：当 ADC 的预分频系数为 4 时，交替转换恢复后采样间隔不再是均匀的 7 个 ADC 时钟，改为 8 个和 6 个时钟周期交替。

12.3 寄存器描述

表 12-5 ADC1 相关寄存器列表

名称	访问地址	描述	复位值
R32_ADC1_STATR	0x40012400	ADC1 状态寄存器	0x00000000
R32_ADC1_CTLR1	0x40012404	ADC1 控制寄存器 1	0x00000000
R32_ADC1_CTLR2	0x40012408	ADC1 控制寄存器 2	0x00000000
R32_ADC1_SAMPTR1	0x4001240C	ADC1 采样时间配置寄存器 1	0x00000000
R32_ADC1_SAMPTR2	0x40012410	ADC1 采样时间配置寄存器 2	0x00000000
R32_ADC1_IOFR1	0x40012414	ADC1 注入通道数据偏移寄存器 1	0x00000000
R32_ADC1_IOFR2	0x40012418	ADC1 注入通道数据偏移寄存器 2	0x00000000
R32_ADC1_IOFR3	0x4001241C	ADC1 注入通道数据偏移寄存器 3	0x00000000
R32_ADC1_IOFR4	0x40012420	ADC1 注入通道数据偏移寄存器 4	0x00000000
R32_ADC1_WDHTR	0x40012424	ADC1 看门狗高阈值寄存器	0x00000000

R32_ADC1_WDLTR	0x40012428	ADC1 看门狗低阈值寄存器	0x00000000
R32_ADC1_RSQR1	0x4001242C	ADC1 规则通道序列寄存器 1	0x00000000
R32_ADC1_RSQR2	0x40012430	ADC1 规则通道序列寄存器 2	0x00000000
R32_ADC1_RSQR3	0x40012434	ADC1 规则通道序列寄存器 3	0x00000000
R32_ADC1_ISQR	0x40012438	ADC1 注入通道序列寄存器	0x00000000
R32_ADC1_IDATAR1	0x4001243C	ADC1 注入数据寄存器 1	0x00000000
R32_ADC1_IDATAR2	0x40012440	ADC1 注入数据寄存器 2	0x00000000
R32_ADC1_IDATAR3	0x40012444	ADC1 注入数据寄存器 3	0x00000000
R32_ADC1_IDATAR4	0x40012448	ADC1 注入数据寄存器 4	0x00000000
R32_ADC1_RDATAR	0x4001244C	ADC1 规则数据寄存器	0x00000000

表 12-6 ADC2 相关寄存器列表

名称	访问地址	描述	复位值
R32_ADC2_STATR	0x40012800	ADC2 状态寄存器	0x00000000
R32_ADC2_CTLR1	0x40012804	ADC2 控制寄存器 1	0x00000000
R32_ADC2_CTLR2	0x40012808	ADC2 控制寄存器 2	0x00000000
R32_ADC2_SAMPTR1	0x4001280C	ADC2 采样时间配置寄存器 1	0x00000000
R32_ADC2_SAMPTR2	0x40012810	ADC2 采样时间配置寄存器 2	0x00000000
R32_ADC2_IOFR1	0x40012814	ADC2 注入通道数据偏移寄存器 1	0x00000000
R32_ADC2_IOFR2	0x40012818	ADC2 注入通道数据偏移寄存器 2	0x00000000
R32_ADC2_IOFR3	0x4001281C	ADC2 注入通道数据偏移寄存器 3	0x00000000
R32_ADC2_IOFR4	0x40012820	ADC2 注入通道数据偏移寄存器 4	0x00000000
R32_ADC2_WDHTR	0x40012824	ADC2 看门狗高阈值寄存器	0x00000000
R32_ADC2_WDLTR	0x40012828	ADC2 看门狗低阈值寄存器	0x00000000
R32_ADC2_RSQR1	0x4001282C	ADC2 规则通道序列寄存器 1	0x00000000
R32_ADC2_RSQR2	0x40012830	ADC2 规则通道序列寄存器 2	0x00000000
R32_ADC2_RSQR3	0x40012834	ADC2 规则通道序列寄存器 3	0x00000000
R32_ADC2_ISQR	0x40012838	ADC2 注入通道序列寄存器	0x00000000
R32_ADC2_IDATAR1	0x4001283C	ADC2 注入数据寄存器 1	0x00000000
R32_ADC2_IDATAR2	0x40012840	ADC2 注入数据寄存器 2	0x00000000
R32_ADC2_IDATAR3	0x40012844	ADC2 注入数据寄存器 3	0x00000000
R32_ADC2_IDATAR4	0x40012848	ADC2 注入数据寄存器 4	0x00000000
R32_ADC2_RDATAR	0x4001284C	ADC2 规则数据寄存器	0x00000000

12.3.1 ADCx 状态寄存器（ADCx_STATR）（x=1/2）

偏移地址：0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											RSTRT	ISTRT	IEOC	EOC	AWD

位	名称	访问	描述	复位值
[31:5]	Reserved	RO	保留。	0
4	RSTRT	RWO	规则通道转换开始状态： 1：规则通道转换已开始； 0：规则通道转换未开始。 该位由硬件置 1，由软件清 0（写 1 无效）。	0
3	ISTRT	RWO	注入通道转换开始状态： 1：注入通道转换已开始； 0：注入通道转换未开始。 该位由硬件置 1，由软件清 0（写 1 无效）。	0
2	IEOC	RWO	注入通道组转换结束状态： 1：转换完成； 0：转换未完成。 该位由硬件置 1（所有注入通道转换完），由软件清 0（写 1 无效）。	
1	EOC	RWO	转换结束状态： 1：转换完成； 0：转换未完成。 该位由硬件置 1（规则或注入通道组转换结束），由软件清 0（写 1 无效）或读 ADC_RDATAR 时清除。	
0	AWD	RWO	模拟看门狗标志位： 1：发生模拟看门狗事件； 0：没有发生模拟看门狗事件。 该位由硬件置 1（转换值超出 ADC_WDHTR 和 ADC_WDLTR 寄存器范围），由软件清 0（写 1 无效）。	

13.3.2 ADCx 控制寄存器 1（ADCx_CTLR1）（x=1/2）

偏移地址：0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PGA[1:0]		BUF EN	TKI TUNE	TKENABLE	RAWDEN	IAWDEN	Reserved		DUALMODE[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			IDISC EN	RDIS CEN	IAUTO	AWD SGL	SCAN	IEOC IE	AWDIE	EOC IE	AWDCH[4:0]				

位	名称	访问	描述	复位值
[31:29]	Reserved	RO	保留。	0
[28:27]	PGA[1:0]	RW	ADC通道增益配置 00：x1 01：x4 10：x16 11：x64	0
26	BUFEN	RW	ADC BUFFER使能 0：关闭输入Buffer 1：使能输入Buffer	0

25	TKITUNE	RW	TKEY模块充电电流配置 0：充电电流为35uA 1：充电电流减半	0
24	TKENABLE	RW	TKEY模块使能控制，包括TKEY_F和TKEY_V单元： 1：开启TKEY模块； 0：关闭 TKEY 模块。	0
23	RAWDEN	RW	在规则通道上模拟看门狗功能使能位： 1：规则通道上使能模拟看门狗； 0：规则通道上关闭模拟看门狗。	0
22	IAWDEN	RW	在注入通道上模拟看门狗功能使能位： 1：注入通道上使能模拟看门狗； 0：注入通道上关闭模拟看门狗。	0
[21:20]	Reserved	RO	保留。	0
[19:16]	DUALMODE[3:0]	RW	双重模式选择。 0000：独立模式 0001：同步规则+同步注入模式 0010：同步规则+交替触发模式 0011：同步注入+快速交替模式 0100：同步注入+慢速交替模式 0101：同步注入模式 0110：同步规则模式 0111：快速交替模式 1000：慢速交替模式 1001：交替触发模式 *注：ADC2 中这些位为保留位，任何配置位的修改应在双重模式关闭的情况下进行。*	0
[15:13]	DISCNUM	RW	间断模式下，外部触发后要转换的规则通道数目： 000：1 个通道； … 111：8 个通道。	0
12	IDISCEN	RW	注入通道上的间断模式使能位： 1：使能注入通道上的间断模式； 0：关闭注入通道上的间断模式。	0
11	RDISCEN	RW	规则通道上的间断模式使能位： 1：使能规则通道上的间断模式； 0：关闭规则通道上的间断模式。	0
10	IAUTO	RW	开启规则通道完成后，自动转换注入通道组使能位： 1：使能自动的注入通道组转换； 0：关闭自动的注入通道组转换。 *注：此模式需要禁止注入通道的外部触发功能。*	0
9	AWDSGL	RW	扫描模式下，在单一通道上使用模拟看门狗使能位： 1：在单一通道上使用模拟看门狗（AWDCH[4:0]选择）； 0：在所有通道上使用模拟看门狗。	0
8	SCAN	RW	扫描模式使能位： 1：使能扫描模式（连续转换 ADC_IOFRx 和 ADC_RSQRx 选择的所有通道）；	0

			0：关闭扫描模式。	
7	IEOCIE	RW	注入通道组转换结束中断使能位： 1：使能注入通道组转换完成中断（IEOC 标志）； 0：关闭注入通道组转换完成中断。	0
6	AWDIE	RW	模拟看门狗中断使能位： 1：使能模拟看门狗中断； 0：关闭模拟看门狗中断。 *注：在扫描模式下，如果发生此中断将中止扫描。*	0
5	EOCIE	RW	转换结束（规则或注入通道组）中断使能位： 1：使能转换结束中断（EOC 标志）； 0：关闭转换结束中断。	0
[4:0]	AWDCH	RW	模拟看门狗通道选择位： 00000：模拟输入通道 0； 00001：模拟输入通道 1； … 10001：模拟输入通道 17。	0

12.3.3 ADCx 控制寄存器 2（ADCx_CTLR2）（x=1/2）

偏移地址：0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TS VREFE	RSW START	ISW START	REXT TRIG	REXTSEL[2:0]			Reser ved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEXT TRIG	IEXTSEL[2:0]			ALIGN	Reserved		DMA	Reserved				RST CAL	CAL	CONT	ADON

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
23	TSVREFE	RW	温度传感器和内部电压（V_{REFINT}）通道使能位： 1：使能温度传感器和 V_{REFINT} 通道； 0：禁止温度传感器和 V_{REFINT} 通道。	0
22	RSWSTART	RW	启动一个规则通道转换，需要设置软件触发： 1：启动规则通道转换； 0：复位状态。 此位由软件置位，转换开始后硬件清 0。	0
21	ISWSTART	RW	启动一个注入通道转换，需要设置软件触发： 1：启动注入通道转换； 0：复位状态。 此位由软件置位，转换开始后硬件清 0 或者软件清 0。	0
20	REXTTRIG	RW	规则通道的外部触发转换模式使能： 1：使用外部事件启动转换； 0：关闭外部事件启动功能。	0
[19:17]	REXTSEL	RW	启动规则通道转换的外部触发事件选择： 000：定时器 1 的 CC1 事件； 001：定时器 1 的 CC2 事件；	0

			010：定时器 1 的 CC3 事件； 011：定时器 2 的 CC2 事件； 100：定时器 3 的 TRGO 事件； 101：定时器 4 的 CC4 事件； 110：EXTI 线 11/定时器 8 的 TRGO 事件； 111：RSWSTART 软件触发。 *注：仅大容量产品中具有定时器 8 的 TRGO 事件*	
16	Reserved	RO	保留。	0
15	IEXTTRIG	RW	注入通道的外部触发转换模式使能： 1：使用外部事件启动转换； 0：关闭外部事件启动功能。	0
[14:12]	IEXTSEL	RW	启动注入通道转换的外部触发事件选择： 000：定时器 1 的 TRGO 事件； 001：定时器 1 的 CC4 事件； 010：定时器 2 的 TRGO 事件； 011：定时器 2 的 CC1 事件； 100：定时器 3 的 CC4 事件； 101：定时器 4 的 TRGO 事件； 110：EXTI 线 15/定时器 8 的 CC4 事件； 111：ISWSTART 软件触发。 *注：仅大容量产品中具有定时器 8 的 CC4 事件*	0
11	ALIGN	RW	数据对齐方式： 1：左对齐；　　0：右对齐。	0
[10:9]	Reserved	RO	保留。	0
8	DMA	RW	直接存储访问（DMA）模式使能： 1：使能 DMA 模式； 0：关闭 DMA 模式。	0
[7:4]	Reserved	RO	保留。	0
3	RSTCAL	RW	复位校准，此位由软件置位，复位完成后由硬件清 0： 1：初始化校准寄存器； 0：校准寄存器已初始化。 *注：如果正在进行转换时设置 RSTCAL，清除校准寄存器需要额外的周期。*	0
2	CAL	RW	A/D 校准，该位由软件置位，校准结束时由硬件清 0。 1：开始校准； 0：校准完成。	0
1	CONT	RW	连续转换使能： 1：连续转换模式； 0：单次转换模式。 如果设置了此位，则转换将连续进行直到该位被清除。	0
0	ADON	RW	开/关 A/D 转换器 当该位为 0 时，写入 1 将把 ADC 从断电模式下唤醒； 当该位为 1 时，写入 1 将启动转换。 1：开启 ADC 并启动转换； 0：关闭 ADC 转换/校准，并进入断电模式。 *注：当寄存器只有 ADON 改变时，才会启动一次转换，*	0

			如果还有其他任意位发送变化，则不会启动新的转换。	

12.3.4 ADCx 采样时间配置寄存器 1（ADCx_SAMPTR1）（x=1/2）

偏移地址：0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP15[0]	SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:0]	SMPx	RW	SMPx[2:0]：通道 x 的采样时间配置： 000：1.5 周期；001:7.5 周期； 010：13.5 周期；011:28.5 周期； 100：41.5 周期；101:55.5 周期； 110：71.5 周期；111:239.5 周期； 这些位用于独立地选择每个通道的采样时间，在采样周期中通道配置值必须保持不变。	

12.3.5 ADCx 采样时间配置寄存器 2（ADCx_SAMPTR2）（x=1/2）

偏移地址：0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP5[0]	SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]		

位	名称	访问	描述	复位值
[31:30]	Reserved	RO	保留。	0
[29:0]	SMPx	RW	SMPx[2:0]：通道 x 的采样时间配置： 000：1.5 周期；001:7.5 周期； 010：13.5 周期；011:28.5 周期； 100：41.5 周期；101:55.5 周期； 110：71.5 周期；111:239.5 周期； 这些位用于独立地选择每个通道的采样时间，在采样周期中通道配置值必须保持不变。	

12.3.6 ADCy 注入通道数据偏移寄存器 x（ADCy_IOFRx）（y=1/2；x=1/2/3/4）

偏移地址：0x14 + (x-1)*4

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

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

位	名称	访问	描述	复位值
[31:12]	Reserved	RO	保留。	0
[11:0]	IOFFSETx	RW	注入通道 x 的数据偏移值。 转换注入通道时，这个值定义了用于从原始转换数据中减去的数值。转换的结果可以在 ADC_IDATARx 寄存器中读出。	0

12.3.7 ADCx 看门狗高阈值寄存器（ADCx_WDHTR）（x=1/2）

偏移地址：0x24

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

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

位	名称	访问	描述	复位值
[31:12]	Reserved	RO	保留。	0
[11:0]	HT	RW	模拟看门狗高阈值设置值。	0

注：可以在转换过程中更改 WDHTR 和 LTR 的值，但它们将在下次转换时生效。

12.3.8 ADCx 看门狗低阈值寄存器（ADCx_WDLTR）（x=1/2）

偏移地址：0x24

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

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

位	名称	访问	描述	复位值
[31:12]	Reserved	RO	保留。	0
[11:0]	LT	RW	模拟看门狗低阈值设置值。	0

注：可以在转换过程中更改 WDHTR 和 LTR 的值，但它们将在下次转换时生效。

12.3.9 ADCx 规则通道序列寄存器 1（ADCx_RSQR1）（x=1/2）

偏移地址：0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								RLEN[3:0]				RSQ16[4:1]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ16[0]	RSQ15[4:0]					RSQ14[4:0]					RSQ13[4:0]				

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:20]	RLEN	RW	规则通道转换序列中需要转换的通道数目： 0000-1111：1-16 个转换。	0
[19:15]	RSQ16	RW	规则序列中的第 16 个转换通道的编号（0-17）。	0
[14:10]	RSQ15	RW	规则序列中的第 15 个转换通道的编号（0-17）。	0
[9:5]	RSQ14	RW	规则序列中的第 14 个转换通道的编号（0-17）。	0
[4:0]	RSQ13	RW	规则序列中的第 13 个转换通道的编号（0-17）。	0

12.3.10 ADCx 规则通道序列寄存器 2（ADCx_RSQR2）（x=1/2）

偏移地址：0x30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		RSQ12[4:0]					RSQ11[4:0]					RSQ10[4:1]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ10[0]		RSQ9[4:0]					RSQ8[4:0]					RSQ7[4:0]			

位	名称	访问	描述	复位值
[31:30]	Reserved	RO	保留。	0
[29:25]	RSQ12	RW	规则序列中的第 12 个转换通道的编号（0-17）。	0
[24:20]	RSQ11	RW	规则序列中的第 11 个转换通道的编号（0-17）。	0
[19:15]	RSQ10	RW	规则序列中的第 10 个转换通道的编号（0-17）。	0
[14:10]	RSQ9	RW	规则序列中的第 9 个转换通道的编号（0-17）。	0
[9:5]	RSQ8	RW	规则序列中的第 8 个转换通道的编号（0-17）。	0
[4:0]	RSQ7	RW	规则序列中的第 7 个转换通道的编号（0-17）。	0

12.3.11 ADCx 规则通道序列寄存器 3（ADCx_RSQR3）（x=1/2）

偏移地址：0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		RSQ6[4:0]					RSQ5[4:0]					RSQ4[4:1]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ4[0]		RSQ3[4:0]					RSQ2[4:0]					RSQ1[4:0]			

位	名称	访问	描述	复位值
[31:30]	Reserved	RO	保留。	0
[29:25]	RSQ6	RW	规则序列中的第 6 个转换通道的编号（0-17）。	0
[24:20]	RSQ5	RW	规则序列中的第 5 个转换通道的编号（0-17）。	0
[19:15]	RSQ4	RW	规则序列中的第 4 个转换通道的编号（0-17）。	0
[14:10]	RSQ3	RW	规则序列中的第 3 个转换通道的编号（0-17）。	0
[9:5]	RSQ2	RW	规则序列中的第 2 个转换通道的编号（0-17）。	0
[4:0]	RSQ1	RW	规则序列中的第 1 个转换通道的编号（0-17）。	0

12.3.12 ADCx 注入通道序列寄存器（ADCx_ISQR）（x=1/2）

偏移地址：0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										ILEN[1:0]		ISQ4[4:1]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISQ4[0]	ISQ3[4:0]					ISQ2[4:0]					ISQ1[4:0]				

位	名称	访问	描述	复位值
[31:22]	Reserved	RO	保留。	0
[21:20]	ILEN	RW	注入通道转换序列中需要转换的通道数目： 00-11：1-4 个转换。	0
[19:15]	ISQ4	RW	注入序列中的第 4 个转换通道的编号（0-17）。	0
[14:10]	ISQ3	RW	注入序列中的第 3 个转换通道的编号（0-17）。	0
[9:5]	ISQ2	RW	注入序列中的第 2 个转换通道的编号（0-17）。	0
[4:0]	ISQ1	RW	注入序列中的第 1 个转换通道的编号（0-17）。	0

注：不同于规则转换序列，如果 ILEN[1:0]的长度小于 4，则转换的序列顺序是从(4-ILEN)开始。

12.3.13 ADCy 注入数据寄存器 x（ADCy_IDATARx）（y=1/2；x=1/2/3/4）

偏移地址：0x3C + (x-1)*4

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

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	IDATA	RO	注入通道转换数据（数据左对齐或右对齐）。	0

12.3.14 ADCx 规则数据寄存器（ADCx_RDATAR）（x=1/2）

偏移地址：0x4C

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

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	RDATA	RO	规则通道转换数据（数据左对齐或右对齐）。	0

第 13 章 触摸按键检测（TKEY）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

触摸检测控制（TKEY）单元，借助 ADC 模块的电压转换功能，通过将电容量转换为电压量进行采样，实现触摸按键检测功能。检测通道复用 ADC 的 16 个外部通道，通过 ADC 模块的单次转换模式实现触摸按键检测。

13.1 TKEY 功能描述

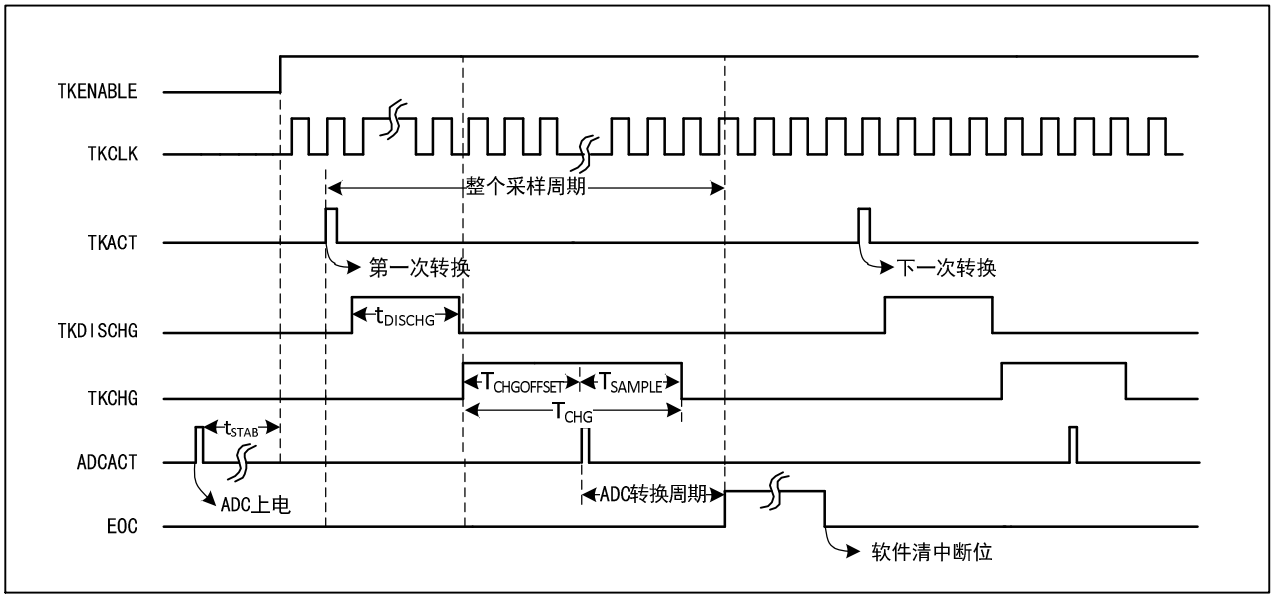
- TKEY 开启

TKEY 检测过程需要 ADC 模块配合进行，所以使用 TKEY 功能时，需要保证 ADC 模块处于上电状态（ADON=1），然后将 ADC_CTLR1 寄存器的 TKENABLE 位置 1，打开 TKEY 单元功能，且可以通过 TKITUNE 位调整 TKEY 模块的充电电流。

TKEY 只支持单次单通道转换模式，将待转换的通道配置到 ADC 模块的规则组序列第一个，软件启动转换（写 TKEY_ACT_DCG 寄存器）。

注：不进行 TKEY 转换时，仍然可以保留 ADC 通道配置转换功能。

图 13-1 TKEY 工作时序图



- 可编程采样时间

TKEY 单元转换需要先使用若干个 ADCCLK 时钟周期（t_{DISCHG}）进行放电，然后再通过若干个 ADCCLK 周期（t_{CHG}）对通道进行充电进行电压采样，充电周期数为 TKEY_CHARGE1 和 TKEY_CHARGE2 寄存器中的 TKCGx[2:0]配置值加上 TKEY_CHGOFFSET 偏移量之和，每个通道可以分别用不同的充电周期来调整采样电压。

13.2 TKEY 操作步骤

TKEY 检测属于 ADC 模块下的扩展功能，其工作原理是通过“触摸”和“非触摸”方式让硬件通道感知的电容量发生变化，进而通过可设置的充放电周期数将电容量的变化转换为电压的变化，最后通过 ADC 模块转换为数字值。

采样时，需要将 ADC 配置为单次单通道工作模式，由 TKEY_ACT 寄存器的“写操作”启动一次转换，具体流程如下：

1） 初始化 ADC 功能，配置 ADC 模块为单次转换模块，置 ACON 位为 1，唤醒 ADC 模块。将 ADC_CTLR1 寄存器的 TKENABLE 位置 1，打开 TKEY 单元。
2） 设置要转换的通道，将通道号写入 ADC 规则组序列中第一个转换位置（ADC_RSQR3[4:0]），设置 RLEN[3:0]为 1。
3） 设置通道的充电采样时间，写 TKEY_CHARGEx 寄存器，可为每个通道配置不同的充电时间。
4） 写 TKEY_CHGOFFSET 寄存器，设置通道的充电时间偏移量（低八位有效），以调整充电时间。
5） 写 TKEY_ACT_DCG 寄存器，设置放电时间（低八位有效），并启动一次 TKEY 的采样和转换。
6） 等待 ADC 状态寄存器的 EOC 转换结束标志位置 1，读取 ADC_DR 寄存器得到此次转换值。
7） 如果需要进行下次转换，重复 2-6 步骤。如果不需修改通道充电采样时间，可省略步骤 3 或 4。

13.3 TKEY 寄存器描述

表 13-1 TKEY1 相关寄存器列表

名称	访问地址	描述	复位值
R32_TKEY1_CHARGE1	0x4001240C	TKEY 充电采样时间寄存器 1	0x00000000
R32_TKEY1_CHARGE2	0x40012410	TKEY 充电采样时间寄存器 2	0x00000000
R32_TKEY1_CHGOFFSET	0x4001243C	TKEY 充电时间偏移量寄存器	0x00000000
R32_TKEY1_ACT_DCG	0x4001244C	TKEY 启动和放电时间寄存器	X
R32_TKEY1_DR	0x4001244C	TKEY 数据寄存器	X

表 13-2 TKEY2 相关寄存器列表

名称	访问地址	描述	复位值
R32_TKEY2_CHARGE1	0x4001280C	TKEY 充电采样时间寄存器 1	0x00000000
R32_TKEY2_CHARGE2	0x40012810	TKEY 充电采样时间寄存器 2	0x00000000
R32_TKEY2_CHGOFFSET	0x4001283C	TKEY 充电时间偏移量寄存器	0x00000000
R32_TKEY2_ACT_DCG	0x4001284C	TKEY 启动和放电时间寄存器	X
R32_TKEY2_DR	0x4001284C	TKEY 数据寄存器	X

13.3.1 TKEYx 充电采样时间寄存器 1（TKEYx_CHARGE1）（x=1/2）

偏移地址：0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TKCG17[2:0]			TKCG16[2:0]			TKCG15[2:1]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKCG15	TKCG14[2:0]			TKCG13[2:0]			TKCG12[2:0]			TKCG11[2:0]			TKCG10[2:0]		

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:0]	TKCGx	RW	TKCGx[2:0] （x=10-17）：选择通道 x 的充电采样时间 这些为用于独立地选择每个通道的充电时间。 000：1.5 周期　100：41.5 周期 001：7.5 周期　101：55.5 周期 010：13.5 周期　110：71.5 周期 011：28.5 周期　111：239.5 周期	0

			时间基准：ADC 时钟。	

注：此寄存器映射 ADC 模块的采样时间寄存器 1（ADC_SAMPTR1）。配置 ADC 功能时，为通道的采用时间；配置 TKEY 功能时，为通道充电时间。

13.3.2 TKEYx 充电采样时间寄存器 2（TKEYx_CHARGE2）（x=1/2）

偏移地址：0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		TKCG9[2:0]			TKCG8[2:0]			TKCG7[2:0]			TKCG6[2:0]			TKCG5[2:1]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKCG5	TKCG4[2:0]			TKCG3[2:0]			TKCG2[2:0]			TKCG1[2:0]			TKCG0[2:0]		

位	名称	访问	描述	复位值
[31:30]	Reserved	RO	保留。	0
[29:0]	TKCGx	RW	TKCGx[2:0]（x=0-9）：选择通道 x 的充电采样时间 这些为用于独立地选择每个通道的充电时间。 000：1.5 周期　100：41.5 周期 001：7.5 周期　101：55.5 周期 010：13.5 周期　110：71.5 周期 011：28.5 周期　111：239.5 周期 时间基准：ADC 时钟。	0

注：此寄存器映射 ADC 模块的采样时间寄存器 1（ADC_SAMPTR2）。配置 ADC 功能时，为通道的采用时间；配置 TKEY 功能时，为通道充电时间。

13.3.3 TKEYx 充电时间偏移量寄存器（TKEYx_CHGOFFSET）（x=1/2）

偏移地址：0x3C

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

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

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:0]	TKCGOFFSET	WO	TKEY 充电时间偏移量配置值。 总充电时间 TCHG=TKCGOFFSET+ TKCGx	0

注：此寄存器映射 ADC 模块的注入数据寄存器 1（ADC_IDATAR1）。因此当该地址寄存器进行“写操作”时，作为 TKEY 充电时间偏移量（TKEY_ CHGOFFSET）执行；进行“读操作”时，作为 ADC 模块的注入数据寄存器 1（ADC_IDATAR1）执行。

13.3.4 TKEYx 启动和放电时间寄存器（TKEYx_ACT_DCG）（x=1/2）

偏移地址：0x4C

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

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

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:0]	TKACT_DCG	WO	写放电时间并启动一次 TKEY 通道检测。	0

注：此寄存器映射 ADC 模块的规则数据寄存器（ADC_RDATAR）。

13.3.5 TKEYx 数据寄存器（TKEYx_DR）（x=1/2）

偏移地址：0x4C

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

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	DATA	RO	转换的数据。	0

注：此寄存器映射 ADC 模块的规则数据寄存器（ADC_RDATAR）。

第 14 章 高级定时器（ADTM）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

高级定时器模块包含一个功能强大的 16 位自动重装定时器（TIM1、TIM8、TIM9 和 TIM10），可用于测量脉冲宽度或产生脉冲、PWM 波等。用于电机控制、电源等领域。

14.1 主要特征

高级定时器（TIM1/8/9/10）的主要特征包括：

- 16 位自动重装计数器，支持增计数模式，减计数模式和增减计数模式；
- 16 位预分频器，分频系数从 1～65536 之间动态可调；
- 支持四路独立的比较捕获通道；
- 每路比较捕获通道支持多种工作模式，比如：输入捕获，输出比较，PWM 生成和单脉冲输出；
- 支持可编程死区时间的互补输出；
- 支持外部信号控制定时器；
- 支持使用重复计数器在确定周期后更新定时器；
- 支持使用刹车信号将定时器复位或置其于确定状态；
- 支持在多种模式下使用 DMA；
- 支持增量式编码器；
- 支持定时器之间的级联和同步

14.2 原理和结构

本节主要论述高级定时器的内部构造。

14.2.1 概述

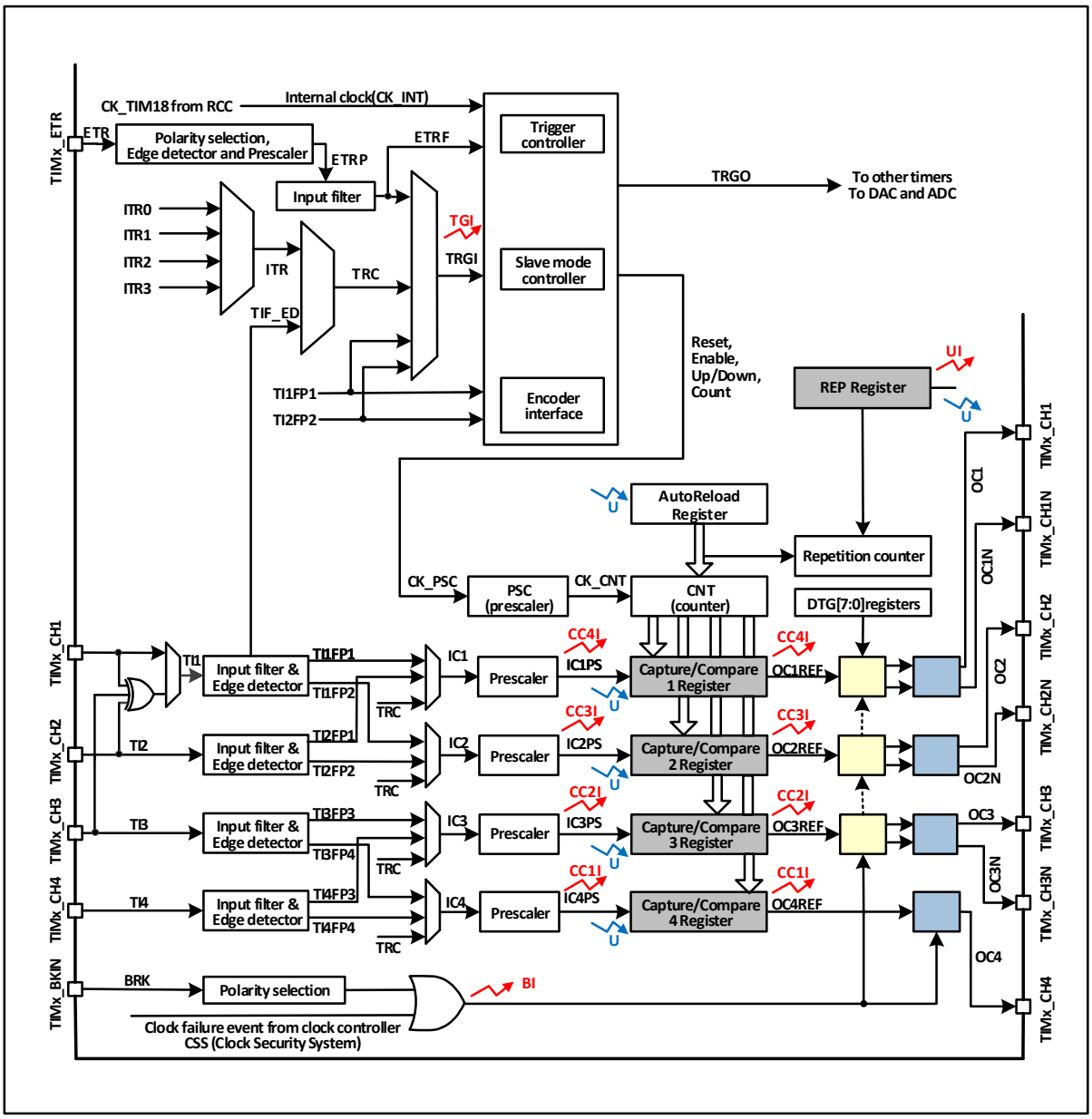
如图 14-1，高级定时器的结构大致可以分为三部分，即输入时钟部分，核心计数器部分和比较捕获通道部分。

高级定时器的时钟可以来自于 APB 总线时钟（CK_INT），可以来自外部时钟输入引脚（TIMx_ETR），亦可以来自于其他具有时钟输出功能的定时器（ITRx），还可以来自于比较捕获通道的输入端（TIMx_CHx）。这些输入的时钟信号经过各种设定的滤波分频等操作后成为 CK_PSC 时钟，输出给核心计数器部分。另外，这些复杂的时钟来源还可以作为 TRGO 输出给其他的定时器、ADC 和 DAC 等外设。

高级定时器的核心是一个 16 位计数器（CNT）。CK_PSC 经过预分频器（PSC）分频后，成为 CK_CNT 并输出给 CNT，CNT 支持增计数模式、减计数模式和增减计数模式，并有一个自动重装值寄存器（ATRLR）在每个计数周期结束后为 CNT 重装载初始值。另外还有个辅助计数器在一旁计数 ATRLR 为 CNT 重装载初值的次数，当次数达到重复计数值寄存器（RPTCR）里设置的次数时，可以产生特定事件。

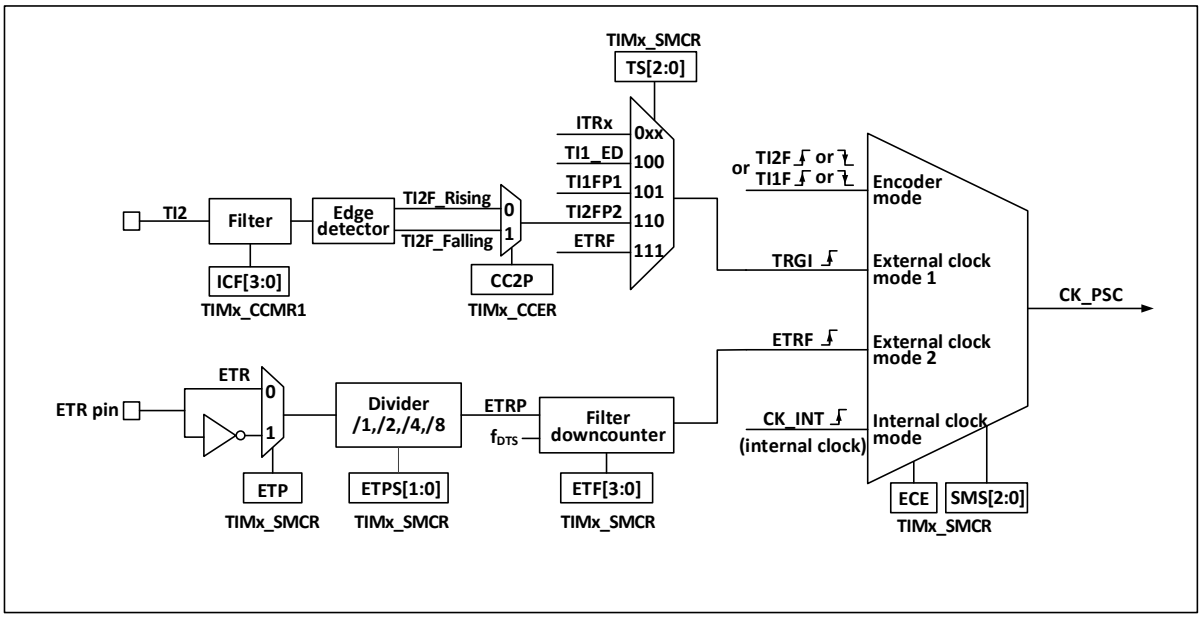
高级定时器拥有四组比较捕获通道，每组比较捕获通道都可以从专属的引脚上输入脉冲，也可以向引脚输出波形，即比较捕获通道支持输入和输出模式。比较捕获寄存器每个通道的输入都支持滤波、分频和边沿检测等操作，并支持通道间的互触发，还能为核心计数器 CNT 提供时钟。每个比较捕获通道都拥有一组比较捕获寄存器（CHxCVR），支持与主计数器（CNT）进行比较而输出脉冲。

图 14-1 高级定时器的结构框图



14.2.2 时钟输入

图 14-2 高级定时器的 CK_PSC 来源框图



高级定时器 CK_PSC 的时钟来源很多，可以分为 4 类：

1） 外部时钟引脚（ETR）输入时钟的路线：ETR→ETRP→ETRF；
2） 内部 APB 时钟输入路线：CK_INT；
3） 来自比较捕获通道引脚（TIMx_CHx）的路线：TIMx_CHx→TIx→TIxFPx，此路线也用于编码器模式；
4） 来自内部其他定时器的输入：ITRx；

通过决定 CK_PSC 来源的 SMS 的输入脉冲选择可以将实际的操作分为 4 类：

1） 选择内部时钟源（CK_INT）；
2） 外部时钟源模式 1；
3） 外部时钟源模式 2；
4） 编码器模式；

上文提到的 4 种时钟源来源都可通过这 4 种操作选定。

14.2.2.1 内部时钟源（CK_INT）

如果将 SMS 域保持 000b 时启动高级定时器，那么就是选定内部时钟源（CK_INT）为时钟。此时 CK_INT 就是 CK_PSC。

14.2.2.2 外部时钟源模式 1

如果将 SMS 域设置为 111b 时，就会启用外部时钟源模式 1。启用外部时钟源 1 时，TRGI 被选定为 CK_PSC 的来源，值得注意的，还需要通过配置 TS 域来选择 TRGI 的来源。TS 域可选择以下几种脉冲作为时钟来源：

1） 内部触发（ITRx，x 为 0,1,2,3）；
2） 比较捕获通道 1 经过边缘检测器后的信号（TI1F_ED）；
3） 比较捕获通道的信号 TI1FP1、TI2FP2；
4） 来自外部时钟引脚输入的信号 ETRF。

14.2.2.3 外部时钟源模式 2

使用外部触发模式 2 能在外部时钟引脚输入的每一个上升沿或下降沿计数。将 ECE 位置位时，将使用外部时钟源模式 2。使用外部时钟源模式 2 时，ETRF 被选定为 CK_PSC。ETR 引脚经过可选的反相器（ETP），分频器（ETPS）后成为 ETRP，再经过滤波器（ETF）后即成为 ETRF。

在 ECE 位置位且将 SMS 设为 111b 时，相当于 TS 选择 ETRF 为输入。

14.2.2.4 编码器模式

将 SMS 置为 001b,010b,011b 将会启用编码器模式。启用编码器模式可以选择在 TI1FP1 和 TI2FP2 中某一个特定的电平下以另一个跳变沿作为信号进行信号输出。此模式用于外接编码器使用的情况下。具体功能参考 14.3.9 节。

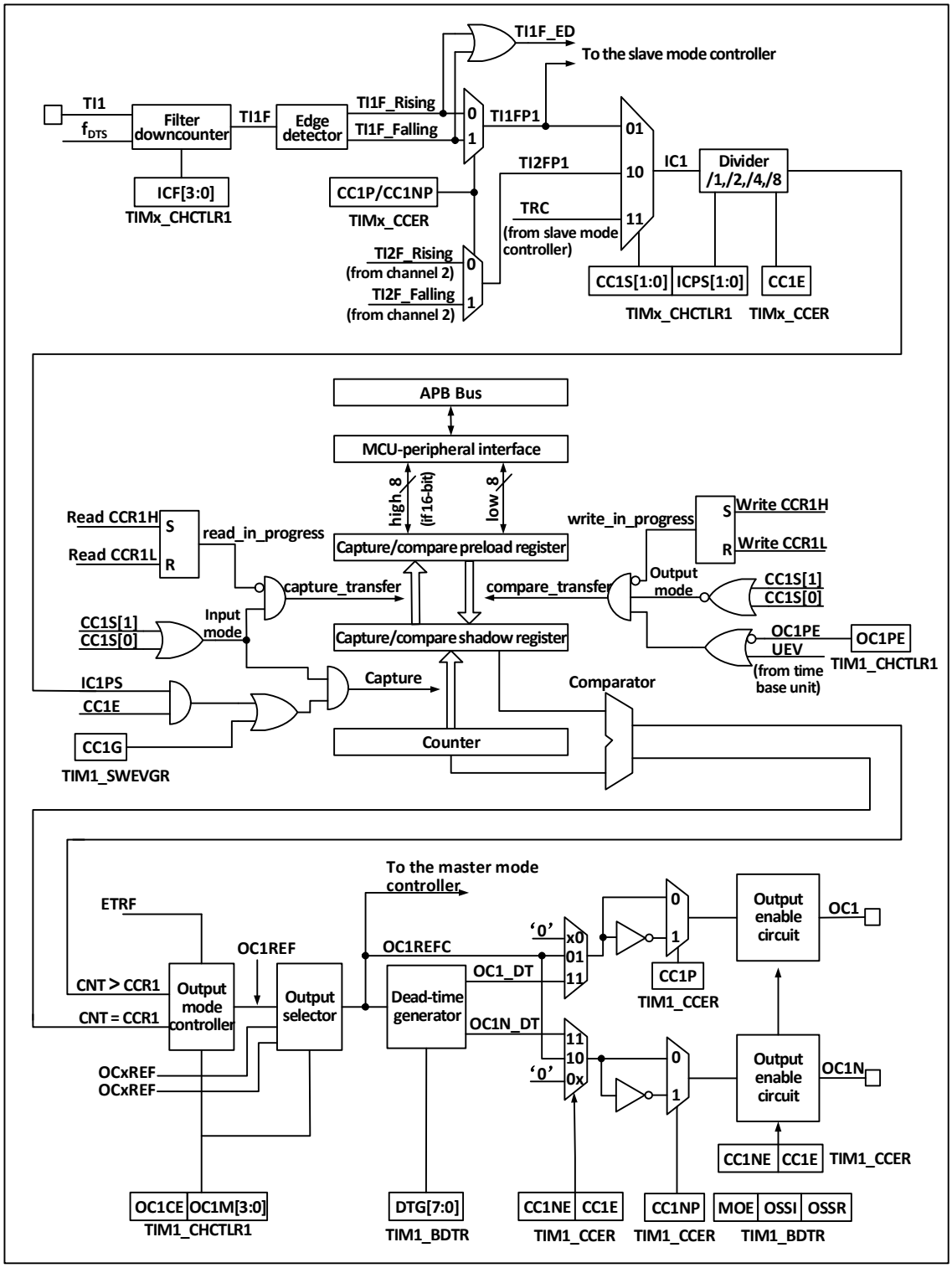
14.2.3 计数器和周边

CK_PSC 输入给预分频器（PSC）进行分频。PSC 是 16 位的，实际的分频系数相当于 R16_TIMx_PSC 的值+1。CK_PSC 经过 PSC 会成为 CK_INT。更改 R16_TIM1_PSC 的值并不会实时生效，而会在更新事件后更新给 PSC。更新事件包括 UG 位清零和复位。定时器的核心是一个 16 位计数器（CNT），CK_CNT 最终会输入给 CNT，CNT 支持增计数模式、减计数模式和增减计数模式，并有一个自动重装值寄存器（ATRLR）在每个计数周期结束后为 CNT 重新装载初始值。另外还有个辅助计数器在一旁记录 ATRLR 为 CNT 重新装载初值的次数，当达到重复计数值寄存器（RPTCR）里设置的次数时，可以产生特定事件。

14.2.4 比较捕获通道和周边

比较捕获通道是定时器实现复杂功能的主要组件，它的核心是比较捕获寄存器，辅以外围输入部分的数字滤波，分频和通道间复用、输出部分的比较器和输出控制组成。

图 14-3 比较捕获通道的结构框图



比较捕获通道的结构框图如图 14-3 所示。信号从通道 x 引脚输入进来后可选做为 TIx（TI1 的来源可以不只是 CH1，见定时器的结构框图 14-1），TI1 经过滤波器（ICF[3:0]）生成 TI1F，再经过边沿检测器分成 TI1F_Rising 和 TI1F_Falling，这两个信号经过选择（CC1P）生成 TI1FP1，TI1FP1 和来自通道 2 的 TI2FP1 一起送给 CC1S 选择成为 IC1，经过 ICPS 分频后送给比较捕获寄存器。

比较捕获寄存器由一个预装载寄存器和一个影子寄存器组成，读写过程仅操作预装载寄存器。在捕获模式下，捕获发生在影子寄存器上，然后复制到预装载寄存器；在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容与核心计数器（CNT）进行比较。

14.3 功能和实现

高级定时器复杂功能的实现都是对定时器的比较捕获通道、时钟输入电路和计数器及周边部分的操作实现的。定时器的时钟输入可以来自于包括比较捕获通道的输入在内的多个时钟源。对比较捕获通道和时钟源选择的操作直接决定其功能。比较捕获通道是双向的，可以工作在输入和输出模式。

14.3.1 输入捕获模式

输入捕获模式是定时器的基本功能之一。输入捕获模式的原理是，当检测到 ICxPS 信号上确定的边沿后，则发生捕获事件，计数器当前的值会被锁存到比较捕获寄存器（R16_TIMx_CHCTLRx）中。发生捕获事件时，CCxIF（在 R16_TIMx_INTFR 中）被置位，如果使能了中断或 DMA，还会产生相应中断或 DMA。如果发生捕获事件时，CCxIF 已经被置位了，那么 CCxOF 位会被置位。CCxIF 可由软件清除，也可以通过读取比较捕获寄存器由硬件清除。CCxOF 由软件清除。

举个通道 1 的例子来说明使用输入捕获模式的步骤，如下：

1） 配置 CCxS 域，选择 ICx 信号的来源。比如设为 10b，选择 TI1FP1 作为 IC1 的来源，而不可以使用默认设置，CCxS 域默认是使比较捕获模块作为输出通道；
2） 配置 ICxF 域，设定 TI 信号的数字滤波器。数字滤波器会以确定的频率，采样确定的次数，再输出一个跳变。这个采样频率和次数是通过 ICxF 来确定的；
3） 配置 CCxP 位，设定 TIxFPx 的极性。比如保持 CC1P 位为低，选择上升沿跳变；
4） 配置 ICxPS 域，设定 ICx 信号成为 ICxPS 之间的分频系数。比如保持 ICxPS 为 00b，不分频；
5） 配置 CCxE 位，允许捕获核心计数器（CNT）的值到比较捕获寄存器中。置 CC1E 位；
6） 根据需要配置 CCxIE 和 CCxDE 位，决定是否允许使能中断或 DMA。

至此已经将比较捕获通道配置完成。

当 TI1 输入了一个被捕获的脉冲时，核心计数器（CNT）的值会被记录到比较捕获寄存器中，CC1IF 被置位，当 CC1IF 在之前就已经被置位时，CCIOF 位也会被置位。如果 CC1IE 位，那么会产生一个中断；如果 CC1DE 被置位，会产生一个 DMA 请求。可以通过写事件产生寄存器（TIMx_SWEVGR）的方式由软件产生一个输入捕获事件。

14.3.2 比较输出模式

比较输出模式是定时器的基本功能之一。比较输出模式的原理是在核心计数器（CNT）的值与比较捕获寄存器的值一致时，输出特定的变化或波形。OCxM 域（在 R16_TIMx_CHCTLRx 中）和 CCxP 位（在 R16_TIMx_CCER 中）决定输出的是确定的高低电平还是电平翻转。产生比较一致事件时还会置 CCxIF 位，如果预先置了 CCxIE 位，则会产生一个中断；如果预先设置了 CCxDE 位，则会产生一个 DMA 请求。

配置为比较输出模式的步骤为下：

1） 配置核心计数器（CNT）的时钟源和自动重装值；
2） 设置需要对比的计数值到比较捕获寄存器（R16_TIMx_CHxCVR）中；
3） 如果需要产生中断，置 CCxIE 位；
4） 保持 OCxPE 为 0，禁用比较寄存器的预装载寄存器；
5） 设定输出模式，设置 OCxM 域和 CCxP 位；
6） 使能输出，置 CCxE 位；
7） 置 CEN 位启动定时器。

14.3.3 强制输出模式

定时器的比较捕获通道的输出模式可以由软件强制输出确定的电平，而不依赖比较捕获寄存器的影子寄存器和核心计数器的比较。

具体的做法是将 OCxM 置为 100b，即为强制将 OCxREF 置为低；或者将 OCxM 置为 101b，即为强制将 OCxREF 置为高。

需要注意的是，将 OCxM 强制置为 100b 或者 101b，内部核心计数器和比较捕获寄存器的比较过程还在进行，相应的标志位还在置位，中断和 DMA 请求还在产生。

14.3.4 PWM 输入模式

PWM 输入模式是用来测量 PWM 的占空比和频率的，是输入捕获模式的一种特殊情况。除下列区别外，操作和输入捕获模式相同：PWM 占用两个比较捕获通道，且两个通道的输入极性设为相反，其中一个信号被设为触发输入，SMS 设为复位模式。

例如，测量从 TI1 输入的 PWM 波的周期和频率，需要进行以下操作：

1） 将 TI1(TI1FP1)设为 IC1 信号的输入。将 CC1S 置为 01b；
2） 将 TI1FP1 置为上升沿有效。将 CC1P 保持为 0；
3） 将 TI1(TI1FP2)置为 IC2 信号的输入。将 CC2S 置为 10b；
4） 选 TI1FP2 置为下降沿有效。将 CC2P 置为 1；
5） 时钟源的来源选择 TI1FP1。将 TS 设为 101b；
6） 将 SMS 设为复位模式，即 100b；
7） 使能输入捕获。CC1E 和 CC2E 置位；

这样比较捕获寄存器 1 的值就是 PWM 的周期，而比较捕获寄存器 2 的值就是其占空比。

14.3.5 PWM 输出模式

PWM 输出模式是定时器的基本功能之一。PWM 输出模式最常见的是使用重装值确定 PWM 频率，使用捕获比较寄存器确定占空比的方法。将 OCxM 域中置 110b 或 111b 使用 PWM 模式 1 或模式 2，置 OCxPE 位使能预装载寄存器，最后置 ARPE 位使能预装载寄存器的自动重装载。由于在发生一个更新事件时，预装载寄存器的值才能被送到影子寄存器，所以在核心计数器开始计数之前，需要置 UG 位来初始化所有寄存器。在 PWM 模式下，核心计数器和比较捕获寄存器一直在进行比较，根据 CMS 位，定时器能够输出边沿对齐或中央对齐的 PWM 信号。

- 边沿对齐

使用边沿对齐时，核心计数器增计数或减计数，在 PWM 模式 1 的情景下，在核心计数器的值大于比较捕获寄存器时，OCxREF 为高；当核心计数器的值小于比较捕获寄存器时（比如核心计数器增长到 R16_TIMx_ATRLR 的值而恢复成全 0 时），OCxREF 为低。

- 中央对齐

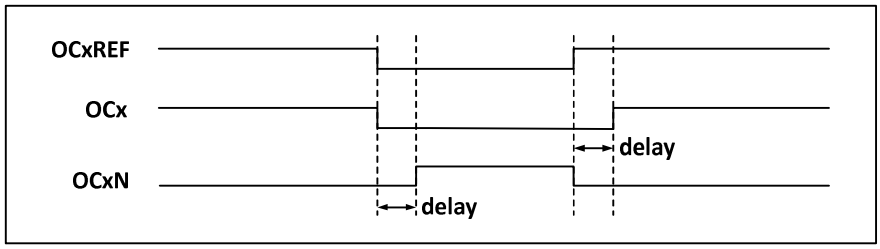
使用中央对齐模式时，核心计数器运行在增计数和减计数交替进行的模式下，OCxREF 在核心计数器和比较捕获寄存器的值一致时进行上升和下降的跳变。但比较标志在三种中央对齐模式下，置位的时机有所不同。在使用中央对齐模式时，最好在启动核心计数器之前产生一个软件更新标志（置 UG 位）。

14.3.6 互补输出和死区

比较捕获通道一般有两个输出引脚（比较捕获通道 4 只有一个输出引脚），能输出两个互补的信号（OCx 和 OCxN），OCx 和 OCxN 可以通过 CCxP 和 CCxNP 位独立地设置极性，通过 CCxE 和 CCxNE 独立地设置输出使能，通过 MOE、OIS、OISN、OSSI、OSSR 位进行死区和其他的控制。同时使能 OCx 和 OCxN 输出将插入死区，每个通道都有一个 10 位的死区发生器。如果存在刹车电路则还要设置 MOE 位。OCx 和 OCxN 由 OCxREF 关联产生，如果 OCx 和 OCxN 都是高有效，那么 OCx 与 OCxREF 相同，只是 OCx 的上升沿相当于 OCxREF 有一个延迟，OCxN 与 OCxREF 相反，它的上升沿相对参考信号的下降沿会有一个延迟，如果延迟大于有效输出宽度，则不会产生相应的脉冲。

如图 14-4 展示了 OCx 和 OCxN 与 OCxREF 的关系，并展示出死区。

图 14-4 互补输出和死区



14.3.7 刹车信号

当产生刹车信号时，输出使能信号和无效电平都会根据 MOE、OIS、OISN、OSSI 和 OSSR 等位进行修改。但 OCx 和 OCxN 不会在任何时间都处在有效电平。刹车事件源可以来自于刹车输入引脚，也可以是一个时钟失败事件，而时钟失败事件由 CSS（时钟安全系统）产生。

在系统复位后，刹车功能被默认禁止（MOE 位为低），置 BKE 位可以使能刹车功能，输入的刹车信号的极性可以通过设置 BKP 设置，BKE 和 BKP 信号可以被同时写入，在真正写入之前会有一个 APB 时钟的延迟，因此需要等一个 APB 周期才能正确读出写入值。

在刹车引脚出现选定的电平系统将产生如下动作：

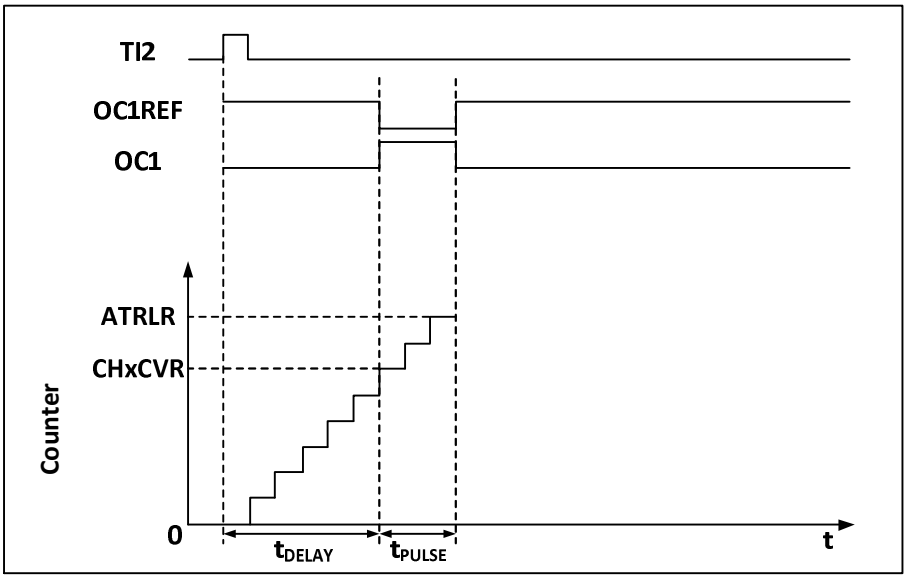
1） MOE 位被异步清零，根据 SOOI 位的设置将输出置为无效状态、空闲状态或复位状态；
2） 在 MOE 被清零后，每一个输出通道输出由 OSIx 确定的电平；
3） 当使用互补输出时：输出被置于无效状态，具体取决于极性；
4） 如果 BIE 被置位，当 BIF 置位，会产生一个中断；如果设置了 BDE 位，则会产生一个 DMA 请求；
5） 如果 AOE 被置位，在下一个更新事件 UEV 时，MOE 位被自动置位。

14.3.8 单脉冲模式

单脉冲模式可以用于让微控制器响应一个特定的事件，使之在一个延迟之后产生一个脉冲，延迟和脉冲的宽度可编程。置 OPM 位可以使核心计数器在产生下一个更新事件 UEV 时（计数器翻转到 0）停止。

如图 14-5，需要在 TI2 输入引脚上检测到一个上升沿开始，延迟 Tdelay 之后，在 OC1 上产生一个长度为 Tpulse 的正脉冲：

图 14-5 单脉冲的产生



1） 设定 TI2 为触发。置 CC2S 域为 01b，把 TI2FP2 映射到 TI2；置 CC2P 位为 0b，TI2FP2 设为上升沿检测；置 TS 域为 110b，TI2FP2 设为触发源；置 SMS 域为 110b，TI2FP2 被用来启动计数器；

2） Tdelay 由比较捕获寄存器的的值确定，Tpulse 由自动重装值寄存器的值和比较捕获寄存器的值确定。

14.3.9 编码器模式

编码器模式是定时器的一个典型应用，可以用来接入编码器的双相输出，核心计数器的计数方向和编码器的转轴方向同步，编码器每输出一个脉冲就会使核心计数器加一或减一。使用编码器的步骤为：将 SMS 域置为 001b（只在 TI2 边沿计数）、010b（只在 TI1 边沿计数）或 011b（在 TI1 和 TI2 双边沿计数），将编码器接到比较捕获通道 1、2 的输入端，给重装值寄存器设一个值，这个值可以设的大一点。在编码器模式时，定时器内部的比较捕获寄存器，预分频器，重复计数寄存器等都正常工作。下表表明了计数方向和编码器信号的关系。

表 14-1 定时器编码器模式的计数方向和编码器信号之间的关系

<table>
<tr><th rowspan="2">计数有效边沿</th><th rowspan="2">相对信号的电平</th><th colspan="2">TI1FP1 信号边沿</th><th colspan="2">TI2FP2 信号</th></tr>
<tr><th>上升沿</th><th>下降沿</th><th>上升沿</th><th>下降沿</th></tr>
<tr><td rowspan="2">仅在 TI1 边沿计数</td><td>高</td><td>向下计数</td><td>向上计数</td><td colspan="2" rowspan="2">不计数</td></tr>
<tr><td>低</td><td>向上计数</td><td>向下计数</td></tr>
<tr><td rowspan="2">仅在 TI2 边沿计数</td><td>高</td><td colspan="2" rowspan="2">不计数</td><td>向上计数</td><td>向下计数</td></tr>
<tr><td>低</td><td>向下计数</td><td>向上计数</td></tr>
<tr><td rowspan="2">在 TI1 和 TI2 双边沿计数</td><td>高</td><td>向下计数</td><td>向上计数</td><td>向上计数</td><td>向下计数</td></tr>
<tr><td>低</td><td>向上计数</td><td>向下计数</td><td>向下计数</td><td>向上计数</td></tr>
</table>

14.3.10 定时器同步模式

定时器能够输出时钟脉冲（TRGO），也能接收其他定时器的输入（ITRx）。不同的定时器的 ITRx 的来源（别的定时器的 TRGO）是不一样的。定时器内部触发连接如表 14-2 所示。

表 14-2 TIMx 内部触发连接

从定时器	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM1	TIM5	TIM2	TIM3	TIM4
TIM8	TIM1	TIM2	TIM4	TIM5
TIM9	TIM10	TIM5	TIM6	TIM7
TIM10	TIM9	TIM2	TIM4	TIM5

14.3.11 调试模式

当系统进入调试模式时，定时器根据 DBG 模块的设置继续运转或停止。

14.4 寄存器描述

表 14-3 TIM1 相关寄存器列表

名称	访问地址	描述	复位值
R16_TIM1_CTLR1	0x40012C00	控制寄存器 1	0x0000
R16_TIM1_CTLR2	0x40012C04	控制寄存器 2	0x0000
R16_TIM1_SMCFGR	0x40012C08	从模式控制寄存器	0x0000
R16_TIM1_DMAINTENR	0x40012C0C	DMA/中断使能寄存器	0x0000
R16_TIM1_INTFR	0x40012C10	中断状态寄存器	0x0000

R16_TIM1_SWEVGR	0x40012C14	事件产生寄存器	0x0000
R16_TIM1_CHCTLR1	0x40012C18	比较/捕获控制寄存器 1	0x0000
R16_TIM1_CHCTLR2	0x40012C1C	比较/捕获控制寄存器 2	0x0000
R16_TIM1_CCER	0x40012C20	比较/捕获使能寄存器	0x0000
R16_TIM1_CNT	0x40012C24	计数器	0x0000
R16_TIM1_PSC	0x40012C28	计数时钟预分频器	0x0000
R16_TIM1_ATRLR	0x40012C2C	自动重装值寄存器	0x0000
R16_TIM1_RPTCR	0x40012C30	重复计数值寄存器	0x0000
R16_TIM1_CH1CVR	0x40012C34	比较/捕获寄存器 1	0x0000
R16_TIM1_CH2CVR	0x40012C38	比较/捕获寄存器 2	0x0000
R16_TIM1_CH3CVR	0x40012C3C	比较/捕获寄存器 3	0x0000
R16_TIM1_CH4CVR	0x40012C40	比较/捕获寄存器 4	0x0000
R16_TIM1_BDTR	0x40012C44	刹车和死区寄存器	0x0000
R16_TIM1_DMACFGR	0x40012C48	DMA 控制寄存器	0x0000
R16_TIM1_DMAADR	0x40012C4C	连续模式的 DMA 地址寄存器	0x0000

表 14-4 TIM8 相关寄存器列表

名称	访问地址	描述	复位值
R16_TIM8_CTLR1	0x40013400	控制寄存器 1	0x0000
R16_TIM8_CTLR2	0x40013404	控制寄存器 2	0x0000
R16_TIM8_SMCFGR	0x40013408	从模式控制寄存器	0x0000
R16_TIM8_DMAINTENR	0x4001340C	DMA/中断使能寄存器	0x0000
R16_TIM8_INTFR	0x40013410	中断状态寄存器	0x0000
R16_TIM8_SWEVGR	0x40013414	事件产生寄存器	0x0000
R16_TIM8_CHCTLR1	0x40013418	比较/捕获控制寄存器 1	0x0000
R16_TIM8_CHCTLR2	0x4001341C	比较/捕获控制寄存器 2	0x0000
R16_TIM8_CCER	0x40013420	比较/捕获使能寄存器	0x0000
R16_TIM8_CNT	0x40013424	计数器	0x0000
R16_TIM8_PSC	0x40013428	计数时钟预分频器	0x0000
R16_TIM8_ATRLR	0x4001342C	自动重装值寄存器	0x0000
R16_TIM8_RPTCR	0x40013430	重复计数值寄存器	0x0000
R16_TIM8_CH1CVR	0x40013434	比较/捕获寄存器 1	0x0000
R16_TIM8_CH2CVR	0x40013438	比较/捕获寄存器 2	0x0000
R16_TIM8_CH3CVR	0x4001343C	比较/捕获寄存器 3	0x0000
R16_TIM8_CH4CVR	0x40013440	比较/捕获寄存器 4	0x0000
R16_TIM8_BDTR	0x40013444	刹车和死区寄存器	0x0000
R16_TIM8_DMACFGR	0x40013448	DMA 控制寄存器	0x0000
R16_TIM8_DMAADR	0x4001344C	连续模式的 DMA 地址寄存器	0x0000

表 14-5 TIM9 相关寄存器列表

名称	访问地址	描述	复位值
R16_TIM9_CTLR1	0x40014C00	控制寄存器 1	0x0000
R16_TIM9_CTLR2	0x40014C04	控制寄存器 2	0x0000
R16_TIM9_SMCFGR	0x40014C08	从模式控制寄存器	0x0000
R16_TIM9_DMAINTENR	0x40014C0C	DMA/中断使能寄存器	0x0000

R16_TIM9_INTFR	0x40014C10	中断状态寄存器	0x0000
R16_TIM9_SWEVGR	0x40014C14	事件产生寄存器	0x0000
R16_TIM9_CHCTLR1	0x40014C18	比较/捕获控制寄存器 1	0x0000
R16_TIM9_CHCTLR2	0x40014C1C	比较/捕获控制寄存器 2	0x0000
R16_TIM9_CCER	0x40014C20	比较/捕获使能寄存器	0x0000
R16_TIM9_CNT	0x40014C24	计数器	0x0000
R16_TIM9_PSC	0x40014C28	计数时钟预分频器	0x0000
R16_TIM9_ATRLR	0x40014C2C	自动重装值寄存器	0x0000
R16_TIM9_RPTCR	0x40014C30	重复计数值寄存器	0x0000
R16_TIM9_CH1CVR	0x40014C34	比较/捕获寄存器 1	0x0000
R16_TIM9_CH2CVR	0x40014C38	比较/捕获寄存器 2	0x0000
R16_TIM9_CH3CVR	0x40014C3C	比较/捕获寄存器 3	0x0000
R16_TIM9_CH4CVR	0x40014C40	比较/捕获寄存器 4	0x0000
R16_TIM9_BDTR	0x40014C44	刹车和死区寄存器	0x0000
R16_TIM9_DMACFGR	0x40014C48	DMA 控制寄存器	0x0000
R16_TIM9_DMAADR	0x40014C4C	连续模式的 DMA 地址寄存器	0x0000

表 14-6 TIM10 相关寄存器列表

名称	访问地址	描述	复位值
R16_TIM10_CTLR1	0x40015000	控制寄存器 1	0x0000
R16_TIM10_CTLR2	0x40015004	控制寄存器 2	0x0000
R16_TIM10_SMCFGR	0x40015008	从模式控制寄存器	0x0000
R16_TIM10_DMAINTENR	0x4001500C	DMA/中断使能寄存器	0x0000
R16_TIM10_INTFR	0x40015010	中断状态寄存器	0x0000
R16_TIM10_SWEVGR	0x40015014	事件产生寄存器	0x0000
R16_TIM10_CHCTLR1	0x40015018	比较/捕获控制寄存器 1	0x0000
R16_TIM10_CHCTLR2	0x4001501C	比较/捕获控制寄存器 2	0x0000
R16_TIM10_CCER	0x40015020	比较/捕获使能寄存器	0x0000
R16_TIM10_CNT	0x40015024	计数器	0x0000
R16_TIM10_PSC	0x40015028	计数时钟预分频器	0x0000
R16_TIM10_ATRLR	0x4001502C	自动重装值寄存器	0x0000
R16_TIM10_RPTCR	0x40015030	重复计数值寄存器	0x0000
R16_TIM10_CH1CVR	0x40015034	比较/捕获寄存器 1	0x0000
R16_TIM10_CH2CVR	0x40015038	比较/捕获寄存器 2	0x0000
R16_TIM10_CH3CVR	0x4001503C	比较/捕获寄存器 3	0x0000
R16_TIM10_CH4CVR	0x40015040	比较/捕获寄存器 4	0x0000
R16_TIM10_BDTR	0x40015044	刹车和死区寄存器	0x0000
R16_TIM10_DMACFGR	0x40015048	DMA 控制寄存器	0x0000
R16_TIM10_DMAADR	0x4001504C	连续模式的 DMA 地址寄存器	0x0000

14.4.1 控制寄存器 1（TIMx_CTLR1）（x=1/8/9/10）

偏移地址：0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD[1:0]		ARPE	CMS[1;0]		DIR	OPM	URS	UDIS	CEN

位	名称	访问	描述	复位值
[15:10]	Reserved	RO	保留。	0
[9:8]	CKD	RW	这 2 位定义在定时器时钟(CK_INT)频率、死区时间和由死区发生器与数字滤波器(ETR,TIx)所用的采样时钟之间的分频比例： 00：Tdts=Tck_int 01：Tdts = 2 x Tck_int 10：Tdts = 4 x Tck_int 11：保留。	0
7	ARPE	RW	自动重装预装使能位： 1：使能自动重装值寄存器（ATRLR）； 0：禁止自动重装值寄存器（ATRLR）。	0
[6:5]	CMS	RW	中央对齐模式选择： 00：边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 01：中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(CHCTLRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向下计数时被设置。 10：中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道(CHCTLRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向上计数时被设置。 11：中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道(CHCTLRx 寄存器中 CCxS=00)的输出比较中断标志位，在计数器向上和向下计数时均被设置。 *注：在计数器使能时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。*	0
4	DIR	RW	计数器方向： 0：计数器的计数模式为增计数； 1：计数器的计数模式为减计数。 *注：当计数器配置为中央对齐模式或编码器模式时，该位无效。*	0
3	OPM	RW	单脉冲模式： 1：在发生下一次更新事件(清除 CEN 位)时，计数器停止。 0：在发生下一次更新事件时，计数器不停止。	0
2	URS	RW	更新请求源，软件通过该位选择 UEV 事件的源。 1：如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求； 0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求。 -计数器溢出/下溢	0

			-设置 UG 位 -从模式控制器产生的更新	
1	UDIS	RW	禁止更新，软件通过该位允许/禁止 UEV 事件的产生。 1：禁止 UEV。不产生更新事件，各寄存器(ARR、PSC、CCRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化； 0：允许 UEV。更新(UEV)事件由下述任一事件产生： -计数器溢出/下溢 -设置 UG 位 -从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。	0
0	CEN	RW	使能计数器。 1：使能计数器； 0：禁止计数器。 *注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。*	0

14.4.2 控制寄存器 2（TIMx_CTLR2）（x=1/8/9/10）

偏移地址：0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	OIS4	OSI3N	OIS3	OSI2N	OIS2	OSI1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	保留	CCPC

位	名称	访问	描述	复位值
15	Reserved	RO	保留。	0
14	OIS4	RW	输出空闲状态 4： 1：当 MOE=0 时，如果实施了 OC4N，则死区后 OC1=1； 0：当 MOE=0 时，如果实施了 OC4N，则死区后 OC1=0。 *注：已经设置了 LOCK(TIMx_BDTR 寄存器)级别 1、2 或 3 后，该位不能被修改。*	0
13	OSI3N	RW	输出空闲状态 3： 1：当 MOE=0 时，死区后 OC1N=1； 0：当 MOE=0 时，死区后 OC1N=0。 *注：已经设置了 LOCK(TIMx_BDTR 寄存器)级别 1、2 或 3 后，该位不能被修改。*	0
12	OIS3	RW	输出空闲状态 3，参见 OIS4。	0
11	OSI2N	RW	输出空闲状态 2，参见 OSI3N。	0
10	OIS2	RW	输出空闲状态 2，参见 OIS4。	0

9	OSI1N	RW	输出空闲状态 1，参见 OSI3N。	0
8	OIS1	RW	输出空闲状态 1，参见 OIS4。	0
7	TI1S	RW	TI1 选择： 1：TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚经异或后连到 TI1 输入； 0：TIMx_CH1 引脚直连到 TI1 输入。	0
[6:4]	MMS	RW	主模式选择:这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。 可能的组合如下： 000：复位－TIMx_EGR 寄存器的 UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟； 001：使能－计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见 TIMx_SMCR 寄存器中 MSM 位的描述)； 010:更新－更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器； 011：比较脉冲－在发生一次捕获或一次比较成功时，当要设置 CC1IF 标志时(即使它已经为高)，触发输出送出一个正脉冲(TRGO)； 100：比较－OC1REF 信号被用于作为触发输出(TRGO)； 101：比较－OC2REF 信号被用于作为触发输出(TRGO)； 110：比较－OC3REF 信号被用于作为触发输出(TRGO)； 111：比较－OC4REF 信号被用于作为触发输出(TRGO)。	0
3	CCDS	RW	捕获比较的 DMA 选择。 1：当发生更新事件时，送出 CHxCVR 的 DMA 请求； 0：当发生 CHxCVR 时，产生 CHxCVR 的 DMA 请求。	0
2	CCUS	RW	比较捕获控制更新选择位。 1：如果 CCPC 置位，可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们； 0：如果 CCPC 置位，只能通过设置 COM 位更新它们。 *注：该位只对具有互补输出的通道起作用。*	0
1	Reserved	RO	保留。	0
0	CCPC	RW	比较捕获预装载控制位。 1：CCxE，CCxNE 和 OCxM 位是预装载的，设置该位后，它们只在设置了 COM 位后被更新；	0

			0：CCxE，CCxNE 和 OCxM 位不是预装载的。 *注：该位只对具有互补输出的通道起作用。*	

14.4.3 从模式控制寄存器（TIMx_SMCFGR）（x=1/8/9/10）

偏移地址：0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		

位	名称	访问	描述	复位值
15	ETP	RO	ETR 触发极性选择，该位选择是直接输入 ETR 还是输入 ETR 的反相。 1：将 ETR 反相，低电平或下降沿有效； 0：ETR，高电平或上升沿有效。	0
14	ECE	RW	外部时钟模式 2 启用选择： 1：使能外部时钟模式 2； 0：禁用外部时钟模式 2。 *注 1：从模式可以与外部时钟模式 2 同时使用：复位模式，门控模式和触发模式；但是，这时 TRGI 不能连到 ETRF（TS 位不能是’111’）。* *注 2：外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETRF。*	0
[13:12]	ETPS	RW	外部触发信号（ETRP）分频，这个信号频率最大不能超过 TIMxCLK 频率的 1/4，可以通过这个域来降频： 00：关闭预分频； 01：ETRP 频率除以 2； 10：ETRP 频率除以 4； 11：ETRP 频率除以 8。	0
[11:8]	ETF	RW	外部触发滤波，实际上，数字滤波器是一个事件计数器，它使用一定的采样的频率，记录到 N 个事件后会产生一个输出的跳变。 0000：无滤波器，以 Fdts 采样； 0001：采样频率 Fsampling=Fck_int，N=2； 0010：采样频率 Fsampling=Fck_int，N=4； 0011：采样频率 Fsampling=Fck_int，N=8； 0100：采样频率 Fsampling=Fdts/2，N=6； 0101：采样频率 Fsampling=Fdts/2，N=8； 0110：采样频率 Fsampling=Fdts/4，N=6； 0111：采样频率 Fsampling=Fdts/4，N=8； 1000：采样频率 Fsampling=Fdts/8，N=6； 1001：采样频率 Fsampling=Fdts/8，N=8； 1010：采样频率 Fsampling=Fdts/16，N=5； 1011：采样频率 Fsampling=Fdts/16，N=6； 1100：采样频率 Fsampling=Fdts/16，N=8； 1101：采样频率 Fsampling=Fdts/32，N=5； 1110：采样频率 Fsampling=Fdts/32，N=6；	0

			1111：采样频率 Fsampling=Fdts/32，N=8。	
7	MSM	RW	主/从模式选择： 1：触发输入(TRGI)上的事件被延迟了，以允许在当前定时器(通过 TRGO)与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的； 0：不发挥作用。	0
[6:4]	TS	RW	触发选择域，这 3 位选择用于同步计数器的触发输入源： 000：内部触发 0(ITR0)； 001：内部触发 1(ITR1)； 010：内部触发 2(ITR2)； 011：内部触发 3(ITR3)； 100：TI1 的边沿检测器(TI1F_ED)； 101：滤波后的定时器输入 1(TI1FP1)； 110：滤波后的定时器输入 2(TI2FP2)； 111：外部触发输入(ETRF)； 以上只有在 SMS 为 0 时改变。 *注：具体见表 14-2。*	0
3	Reserved	RO	保留。	0
[2:0]	SMS	RW	输入模式选择域。选择核心计数器的时钟和触发模式。 000：由内部时钟 CK_INT 驱动； 001：编码器模式 1，根据 TI1FP1 的电平，核心计数器在 TI2FP2 的边沿增减计数； 010：编码器模式 2，根据 TI2FP2 的电平，核心计数器在 TI1FP1 的边沿增减计数； 011：编码器模式 3，根据另一个信号的输入电平，核心计数器在 TI1FP1 和 TI2FP2 的边沿增减计数； 100：复位模式，触发输入(TRGI)的上升沿将初始化计数器，并且产生一个更新寄存器的信号； 101：门控模式，当触发输入(TRGI)为高时，计数器的时钟开启；在触发输入变为低，计数器停止，计数器的启停都是受控的； 110：触发模式，计数器在触发输入 TRGI 的上升沿启动，只有计数器的启动是受控的； 111：外部时钟模式 1，选中的触发输入(TRGI)的上升沿驱动计数器。	0

14.4.4 DMA/中断使能寄存器（TIMx_DMAINTENR）（x=1/8/9/10）

偏移地址：0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE

位	名称	访问	描述	复位值
15	Reserved	RO	保留。	0
14	TDE	RW	触发 DMA 请求使能位。 1：允许触发 DMA 请求； 0：禁止触发 DMA 请求。	0
13	COMDE	RW	COM 的 DMA 请求使能位。 1：允许 COM 的 DMA 请求； 0：禁止 COM 的 DMA 请求。	0
12	CC4DE	RW	比较捕获通道 4 的 DMA 请求使能位。 1：允许比较捕获通道 4 的 DMA 请求； 0：禁止比较捕获通道 4 的 DMA 请求。	0b
11	CC3DE	RW	比较捕获通道 3 的 DMA 请求使能位。 1：允许比较捕获通道 3 的 DMA 请求； 0：禁止比较捕获通道 3 的 DMA 请求。	0
10	CC2DE	RW	比较捕获通道 2 的 DMA 请求使能位。 1：允许比较捕获通道 2 的 DMA 请求； 0：禁止比较捕获通道 2 的 DMA 请求。	0
9	CC1DE	RW	比较捕获通道 1 的 DMA 请求使能位。 1：允许比较捕获通道 1 的 DMA 请求； 0：禁止比较捕获通道 1 的 DMA 请求。	0
8	UDE	RW	更新的 DMA 请求使能位。 1：允许更新的 DMA 请求； 0：禁止更新的 DMA 请求。	0b
7	BIE	RW	刹车中断使能位。 1：允许刹车中断； 0：禁止刹车中断。	0
6	TIE	RW	触发中断使能位。 1：使能触发中断； 0：禁止触发中断。	0
5	COMIE	RW	COM 中断允许位。 1：允许 COM 中断； 0：禁止 COM 中断。	0
4	CC4IE	RW	比较捕获通道 4 中断使能位。 1：允许比较捕获通道 4 中断； 0：禁止比较捕获通道 4 中断。	0
3	CC3IE	RW	比较捕获通道 3 中断使能位。 1：允许比较捕获通道 3 中断； 0：禁止比较捕获通道 3 中断。	0
2	CC2IE	RW	比较捕获通道 2 中断使能位。 1：允许比较捕获通道 2 中断； 0：禁止比较捕获通道 2 中断。	0
1	CC1IE	RW	比较捕获通道 1 中断使能位。 1：允许比较捕获通道 1 中断； 0：禁止比较捕获通道 1 中断。	0
0	UIE	RW	更新中断使能位。	0

			1：允许更新中断； 0：禁止更新中断。	

14.4.5 中断状态寄存器（TIMx_INTFR）（x=1/8/9/10）

偏移地址：0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF

位	名称	访问	描述	复位值
[15:13]	Reserved	RO	保留。	0
12	CC4OF	RW0	比较捕获通道 4 重复捕获标志位。	0
11	CC3OF	RW0	比较捕获通道 3 重复捕获标志位。	0
10	CC2OF	RW0	比较捕获通道 2 重复捕获标志位。	0
9	CC1OF	RW0	比较捕获通道 1 重复捕获标志位，仅用于比较捕获通道被配置为输入捕获模式时。该标记由硬件置位，软件写 0 可清除此位。 1：计数器的值被捕获到捕获比较寄存器时，CC1IF 的状态已经被置位； 0：无重复捕获产生。	0b
8	保留	RO	保留。	0
7	BIF	RW0	刹车中断标志位，一旦刹车输入有效，由硬件对该位置位，可由软件清零。 1：刹车引脚输入上检测到设定的有效电平； 0：无刹车事件产生。	0
6	TIF	RW0	触发器中断标志位，当发生触发事件时由硬件对该位置位，由软件清零。触发事件包括从除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿。 1：触发器事件产生； 0：无触发器事件产生。	0
5	COMIF	RW0	COM 中断标志位，一旦产生 COM 事件，该位由硬件置位，由软件清零。COM 事件包括 CCxE、CCxNE、OCxM 被更新。 1：COM 事件产生； 0：无 COM 事件产生。	0
4	CC4IF	RW0	比较捕获通道 4 中断标志位。	0
3	CC3IF	RW0	比较捕获通道 3 中断标志位。	0
2	CC2IF	RW0	比较捕获通道 2 中断标志位。	0
1	CC1IF	RW0	比较捕获通道 1 中断标志位。 如果比较捕获通道配置为输出模式： 当计数器值与比较值匹配时该位由硬件置位，但在中心对称模式下除外。该位由软件清零。 1：核心计数器的值与比较捕获寄存器 1 的值匹配； 0：无匹配发生。 如果比较捕获通道 1 配置为输入模式：	0

			当捕获事件发生时该位由硬件置位，它由软件清零或通过读比较捕获寄存器清零。 1：计数器值已被捕获比较捕获寄存器 1； 0：无输入捕获产生。	
0	UIF	RW0	更新中断标志位，当产生更新事件时该位由硬件置位，由软件清零。 1：更新中断产生； 0：无更新事件产生。 以下情形会产生更新事件： 若 UDIS=0，当重复计数器数值上溢或下溢时； 若 URS=0、UDIS=0，当置 UG 位时，或当通过软件对计数器核心计数器重新初始化时； 若 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时；	0

14.4.6 事件产生寄存器（TIMx_SWEVGR）（x=1/8/9/10）

偏移地址：0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
7	BG	WO	刹车事件产生位，此位由软件置位和清零，用来产生一个刹车事件。 1：产生一个刹车事件。此时 MOE=0、BIF=1，若使能对应的中断和 DMA，则产生相应的中断和 DMA； 0：无动作。	0
6	TG	WO	触发事件产生位，该位由软件置位，硬件清零，用于产生一个触发事件。 1：产生一个触发事件，TIF 被置位，若使能对应的中断和 DMA，则产生相应的中断和 DMA； 0：无动作。	0
5	COMG	WO	比较捕获控制更新产生位。产生比较捕获控制更新事件。该位由软件置位，由硬件自动清零。 1：当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位； 0：无动作。 *注：该位只对拥有互补输出的通道（通道 1，2，3）有效。*	0
4	CC4G	WO	比较捕获事件产生位 4。产生比较捕获事件 4。	0
3	CC3G	WO	比较捕获事件产生位 3。产生比较捕获事件 3。	0
2	CC2G	WO	比较捕获事件产生位 2。产生比较捕获事件 2。	0
1	CC1G	WO	比较捕获事件产生位 1，产生比较捕获事件 1。 该位由软件置位，由硬件清零。用于产生一个比较捕获事件。 1：在比较捕获通道 1 上产生一个比较捕获事件：	0

			若比较捕获通道 1 配置为输出： 置 CC1IF 位。若使能对应的中断和 DMA，则产生相应的中断和 DMA； 若比较捕获通道 1 配置为输入： 当前核心计数器的值被捕获至比较捕获寄存器 1；置 CC1IF 位，若使能了对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经置位，则置 CC1OF 位。 0：无动作。	
0	UG	WO	更新事件产生位，产生更新事件。该位由软件置位，由硬件自动清零。 1：初始化计数器，并产生一个更新事件； 0：无动作。 *注：预分频器的计数器也被清零，但是预分频系数不变。若在中心对称模式下或增计数模式下则核心计数器被清零；若减计数模式下则核心计数器取重装值寄存器的值。*	0

14.4.7 比较/捕获控制寄存器 1（TIMx_CHCTLR1）（x=1/8/9/10）

偏移地址：0x18

通道可用于输入(捕获模式)或输出(比较模式)，通道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			

比较模式（引脚方向为输出）：

位	名称	访问	描述	复位值
15	OC2CE	RW	比较捕获通道 2 清零使能位。 1:一旦检测到 ETRF 输入高电平,清除 OC2REF 位零； 0：OC2REF 不受 ETRF 输入的影响。	0
[14:12]	OC2M	RW	比较捕获通道 2 模式设置域。 该 3 位定义了输出参考信号 OC2REF 的动作，而 OC2REF 决定了 OC2、OC2N 的值。OC2REF 是高电平有效，而 OC2 和 OC2N 的有效电平取决于 CC2P、CC2NP 位。 000：冻结。比较捕获寄存器的值与核心计数器间的比较值对 OC1REF 不起作用； 001：强制设为有效电平。当核心计数器与比较捕获寄存器 1 的值相同时，强制 OC1REF 为高； 010：强制设为无效电平。当核心计数器的值与比较捕获寄存器 1 相同时，强制 OC1REF 为低； 011：翻转。当核心计数器与比较捕获寄存器 1 的值相同时，翻转 OC1REF 的电平。 100：强制为无效电平。强制 OC1REF 为低。	0

			101：强制为有效电平。强制 OC1REF 为高。 110：PWM 模式 1：在向上计数时，一旦核心计数器大于比较捕获寄存器的值时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦核心计数器大于比较捕获寄存器的值时通道 1 为有效电平，否则为无效电平。 111：PWM 模式 2：在向上计数时，一旦核心计数器大于比较捕获寄存器的值时，通道 1 为有效电平，否则为无效电平；在向下计数时，一旦核心计数器大于比较捕获寄存器的值时，通道 1 为无效电平，否则为有效电平(OC1REF=1)。 *注：一旦 LOCK 级别设为 3 并且 CC1S=00b 则该位不能被修改。在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。*	
11	OC2PE	RW	比较捕获寄存器 2 预装载使能位。 1：开启比较捕获寄存器的预装载功能，读写操作仅对预装载寄存器操作，比较捕获寄存器 1 的预装载值在更新事件到来时被加载至当前影子寄存器中； 0：禁止比较捕获寄存器 2 的预装载功能，可随时写入比较捕获寄存器 2，并且新写入的数值立即起作用。 *注：一旦 LOCK 级别设为 3 并且 CC1S=00，则该位不能被修改；仅仅在单脉冲模式下(OPM=1)可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。*	0
10	OC2FE	RW	比较捕获通道 2 快速使能位，该位用于加快比较捕获通道输出对触发输入事件的响应。 1:输入到触发器的有效沿的作用就像发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和比较捕获通道 2 输出间的延时被缩短为 3 个时钟周期； 0：根据计数器与比较捕获寄存器 2 的值，比较捕获通道 2 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活比较捕获通道 2 输出的最小延时为 5 个时钟周期。 OC2FE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。	0
[9:8]	CC2S	RW	比较捕获通道 2 输入选择域。 00：比较捕获通道 2 被配置为输出； 01：比较捕获通道 2 被配置为输入，IC2 映射在 TI2 上； 10：比较捕获通道 2 被配置为输入，IC2 映射在 TI1 上； 11：比较捕获通道 2 被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由	0

			TS 位选择)。 *注：比较捕获通道 2 仅在通道关闭时(CC2E 为零时)才是可写的。*	
7	OC1CE	RW	比较捕获通道 1 清零使能位。	0
[6:4]	OC1M	RW	比较捕获通道 1 模式设置域。	0
3	OC1PE	RW	比较捕获寄存器 1 预装载使能位。	0
2	OC1FE	RW	比较捕获通道 1 快速使能位。	0
[1:0]	CC1S	RW	比较捕获通道 1 输入选择域。	0

捕获模式（引脚方向为输入）：

位	名称	访问	描述	复位值
[15:12]	IC2F	RW	输入捕获滤波器 2 配置域，这几位设置了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变。 0000：无滤波器，以 fDTS 采样； 1000：采样频率 Fsampling=Fdts/8，N=6； 0001：采样频率 Fsampling=Fck_int，N=2； 1001：采样频率 Fsampling=Fdts/8，N=8； 0010：采样频率 Fsampling=Fck_int，N=4； 1010：采样频率 Fsampling=Fdts/16，N=5； 0011：采样频率 Fsampling=f=Fck_int，N=8； 1011：采样频率 Fsampling=Fdts/16，N=6； 0100：采样频率 Fsampling=Fdts/2，N=6； 1100：采样频率 Fsampling=Fdts/16，N=8； 0101：采样频率 Fsampling=Fdts/2，N=8； 1101：采样频率 Fsampling=Fdts/32，N=5； 0110：采样频率 Fsampling=Fdts/4，N=6； 1110：采样频率 Fsampling=Fdts/32，N=6； 0111：采样频率 Fsampling=Fdts/4，N=8； 1111：采样频率 Fsampling=Fdts/32，N=8。	0
[11:10]	IC2PSC	RW	比较捕获通道 2 预分频配置域，这 2 位定义了比较捕获通道 2 的预分频系数。一旦 CC1E=0，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获； 01：每 2 个事件触发一次捕获； 10：每 4 个事件触发一次捕获； 11：每 8 个事件触发一次捕获。	0
[9:8]	CC2S	RW	比较捕获通道 2 输入选择域，这 2 位定义通道的方向(输入/输出)，及输入脚的选择。 00：比较捕获通道 1 通道被配置为输出； 01：比较捕获通道 1 通道被配置为输入，IC1 映射在 TI1 上； 10：比较捕获通道 1 通道被配置为输入，IC1 映射	0

			在 TI2 上； 11：比较捕获通道 1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TS 位选择)。 *注：CC1S 仅在通道关闭时(CC1E 为 0)才是可写的。*	
[7:4]	IC1F	RW	输入捕获滤波器 1 配置域。	0
[3:2]	IC1PSC	RW	比较捕获通道 1 预分频配置域。	0
[1:0]	CC1S	RW	比较捕获通道 1 输入选择域。	0

14.4.8 比较/捕获控制寄存器 2（TIMx_CHCTLR2）（x=1/8/9/10）

偏移地址：0x1C

通道可用于输入(捕获模式)或输出(比较模式)，通道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			

比较模式（引脚方向为输出）：

位	名称	访问	描述	复位值
15	OC4CE	RW	比较捕获通道 4 清零使能位。	0
[14:12]	OC4M	RW	比较捕获通道 4 模式设置域。	0
11	OC4PE	RW	比较捕获寄存器 4 预装载使能位。	0
10	OC4FE	RW	比较捕获通道 4 快速使能位。	0
[9:8]	CC4S	RW	比较捕获通道 4 输入选择域。	0
7	OC3CE	RW	比较捕获通道 3 清零使能位。	0
[6:4]	OC3M	RW	比较捕获通道 3 模式设置域。	0
3	OC3PE	RW	比较捕获寄存器 3 预装载使能位。	0
2	OC3FE	RW	比较捕获通道 3 快速使能位。	0
[1:0]	CC3S	RW	比较捕获通道 3 输入选择域。	0

捕获模式（引脚方向为输入）：

位	名称	访问	描述	复位值
[15:12]	IC4F	RW	输入捕获滤波器 4 配置域。	0
[11:10]	IC4PSC	RW	比较捕获通道 4 预分频配置域。	0
[9:8]	CC4S	RW	比较捕获通道 4 输入选择域。	0
[7:4]	IC3F	RW	输入捕获滤波器 3 配置域。	0
[3:2]	IC3PSC	RW	比较捕获通道 3 预分频配置域。	0
[1:0]	CC3S	RW	比较捕获通道 3 输入选择域。	0

14.4.9 比较/捕获使能寄存器（TIMx_CCER）（x=1/8/9/10）

偏移地址：0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E

位	名称	访问	描述	复位值
[15:14]	Reserved	RO	保留。	0
13	CC4P	RW	比较捕获通道 4 输出极性设置位。	0
12	CC4E	RW	比较捕获通道 4 输出使能位。	0
11	CC3NP	RW	比较捕获通道 3 互补输出极性设置位。	0
10	CC3NE	RW	比较捕获通道 3 互补输出使能位。	0
9	CC3P	RW	比较捕获通道 3 输出极性设置位。	0
8	CC3E	RW	比较捕获通道 3 输出使能位。	0
7	CC2NP	RW	比较捕获通道 2 互补输出极性设置位。	0
6	CC2NE	RW	比较捕获通道 2 互补输出使能位。	0
5	CC2P	RW	比较捕获通道 2 输出极性设置位。	0
4	CC2E	RW	比较捕获通道 2 输出使能位。	0
3	CC1NP	RW	比较捕获通道 1 互补输出极性设置位。	0
2	CC1NE	RW	比较捕获通道 1 互补输出使能位。	0
1	CC1P	RW	比较捕获通道 1 输出极性设置位。	0
0	CC1E	RW	比较捕获通道 1 输出使能位。	0

14.4.10 高级定时器的计数器（TIMx_CNT）（x=1/8/9/10）

偏移地址：0x24

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

位	名称	访问	描述	复位值
[15:0]	CNT	RW	定时器的计数器的实时值。	0

14.4.11 计数时钟预分频器（TIMx_PSC）（x=1/8/9/10）

偏移地址：0x28

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

位	名称	访问	描述	复位值
[15:0]	PSC	RW	定时器的预分频器的分频系数；计数器的时钟频率等于分频器的输入频率/(PSC+1)。	0

14.4.12 自动重装值寄存器（TIMx_ATRLR）（x=1/8/9/10）

偏移地址：0x2C

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

位	名称	访问	描述	复位值
[15:0]	ATRLR	RW	此域的值将会被装入计数器，ATRLR 何时动作和更新见 14.2.3 章节；ATRLR 为空时，计数器停止。	0

14.4.13 重复计数值寄存器（TIMx_RPTCR）（x=1/8/9/10）

偏移地址：0x30

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

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
[7:0]	RPTCR	RW	重复计数器的值。	0

14.4.14 比较/捕获寄存器 1（TIMx_CH1CVR）（x=1/8/9/10）

偏移地址：0x34

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1CVR[15:0]															

位	名称	访问	描述	复位值
[15:0]	CH1CVR	RW	比较捕获寄存器通道 1 的值。	0

14.4.15 比较/捕获寄存器 2（TIMx_CH2CVR）（x=1/8/9/10）

偏移地址：0x38

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2CVR[15:0]															

位	名称	访问	描述	复位值
[15:0]	CH2CVR	RW	比较捕获寄存器通道 2 的值。	0

14.4.16 比较/捕获寄存器 3（TIMx_CH3CVR）（x=1/8/9/10）

偏移地址：0x3C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3CVR [15:0]															

位	名称	访问	描述	复位值
[15:0]	CH3CVR	RW	比较捕获寄存器通道 3 的值。	0

14.4.17 比较/捕获寄存器 4（TIMx_CH4CVR）（x=1/8/9/10）

偏移地址：0x40

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4CVR [15:0]															

位	名称	访问	描述	复位值
[15:0]	CH4CVR	RW	比较捕获寄存器通道 4 的值。	0

14.4.18 刹车和死区寄存器（TIMx_BDTR）（x=1/8/9/10）

偏移地址：0x44

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							

位	名称	访问	描述	复位值
15	MOE	RW	主输出使能位。一旦刹车信号有效，将被异步清零。 1：允许 OCx 和 OCxN 设为输出； 0：禁止 OCx 和 OCxN 的输出或者强制为空闲状态。	0
14	AOE	RW	自动输出使能。 1：MOE 可以被软件置位或者在下一个更新事件中被置位； 0：MOE 只能被软件置位。	0
13	BKP	RW	刹车输入极性设置位。 1：刹车输入高电平有效； 0：刹车输入低电平有效。 *注：当设置了 LOCK 级别 1 后，该位不能被修改。对该位的写需要一个 APB 时钟以后才能生效。*	0
12	BKE	RW	刹车功能使能位。 1：开启刹车输入； 0：禁止刹车输入。 *注：当设置了 LOCK 级别 1 后，该位不能被修改。对该位的写需要一个 APB 时钟以后才能生效。*	0
11	OSSR	RW	1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，首先开启 OC/OCN 并输出无效电平，然后置 OCx、OCxN 使能输出信号=1； 0：当定时器不工作时，禁止 OC/OCN 输出。 *注：当设置了 LOCK 级别 1 后，该位不能被修改。*	0
10	OSSI	RW	1：当定时器不工作时，一旦 CCxE=1 或 CCxNE=1，OC/OCN 首先输出其空闲电平，然后 OCx、OCxN 使能输出信号=1； 0：当定时器不工作时，禁止 OC/OCN 输出。 *注：当设置了 LOCK 级别 1 后，该位不能被修改。*	0
[9:8]	LOCK	RW	锁定功能设置域。 00：关闭锁定功能； 01：锁定级别 1，不能写 DTG、BKE、BKP、AOE、OISx 和 OISxN 位； 10：锁定级别 2，不能写入锁定级别 1 中的各位，也不能写入 CC 极性位以及 OSSR 和 OSSI 位； 11：锁定级别 3，不能写入锁定级别 2 中的各位，也不能写入 CC 控制位。 *注：在系统复位后，只能写一次 LOCK 位，无法再次修改直到复位。*	0

[7:0]	DTG	RW	死区设置位，这些位定义了互补输出之间的死区持续时间。 假设 DT 表示其持续时间： DTG[7:5]=0xx=>DT=DTG[7:0]*Tdtg，Tdtg =TDTS； DTG[7:5]=10x=>DT=(64+DTG[5:0])*Tdtg ， Tdtg=2*TDTS； DTG[7:5]=110=>DT=(32+DTG[4:0])*Tdtg，Tdtg =8×TDTS； DTG[7:5]=111=>DT=(32+DTG[4:0])*Tdtg，Tdtg =16*TDTS。	0

14.4.19 DMA 控制寄存器（TIMx_DMACFGR）（x=1/8/9/10）

偏移地址：0x48

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

位	名称	访问	描述	复位值
[15:13]	Reserved	RO	保留。	0
[12:8]	DBL	RW	DMA 连续传送的长度，实际值为此域的值+1。	0
[7:5]	Reserved	RO	保留。	0
[4:0]	DBA	RW	这些位定义了 DMA 在连续模式下从控制寄存器 1 所在地址的偏移量。	0

14.4.20 连续模式的 DMA 地址寄存器（TIMx_DMAADR）（x=1/8/9/10）

偏移地址：0x4C

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

位	名称	访问	描述	复位值
[15:0]	DMAADR	RW	连续模式下，DMA 的地址。	0

第 15 章 通用定时器（GPTM）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

通用定时器模块包含一个 16 位可自动重装的定时器（TIM2、TIM3、TIM4 和 TIM5），用于测量脉冲宽度或者产生特定频率的脉冲、PWM 波等。可用于自动化控制、电源等领域。

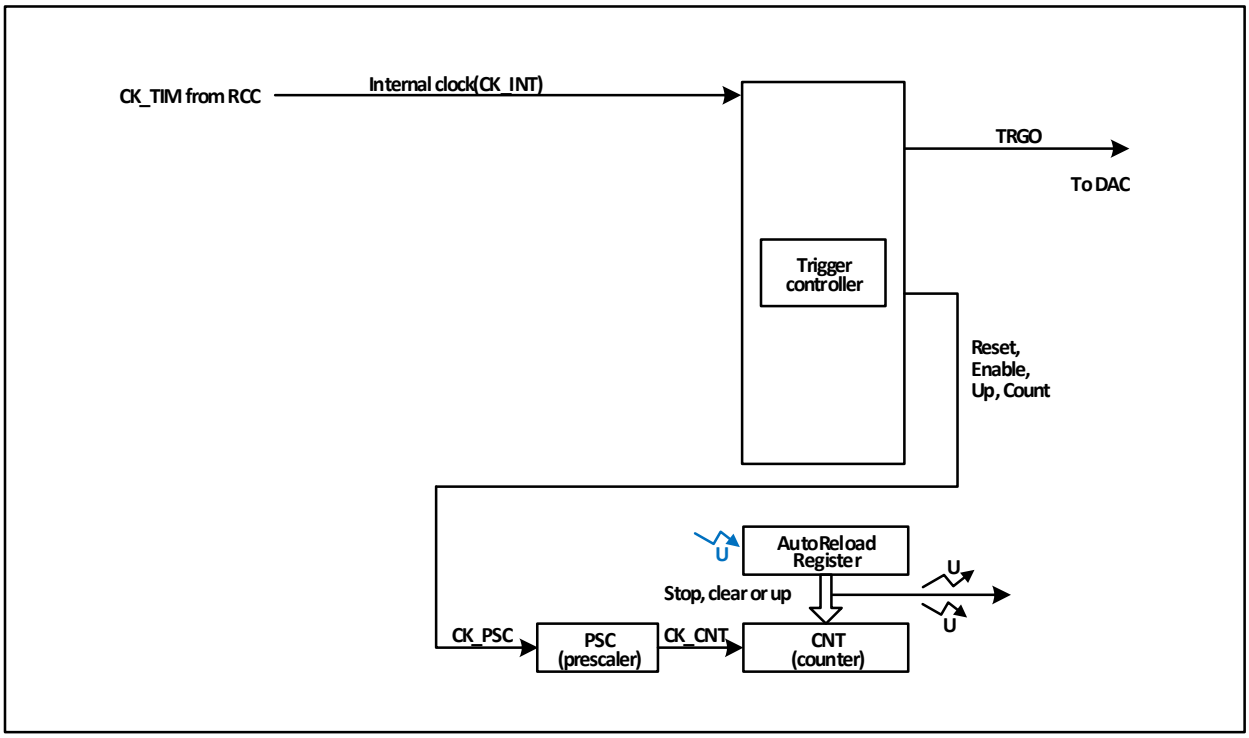
15.1 主要特征

通用定时器的主要特征包括：

- 16 位自动重装计数器，支持增计数模式，减计数模式和增减计数模式
- 16 位预分频器，分频系数从 1～65536 之间动态可调
- 支持四路独立的比较捕获通道
- 每路比较捕获通道支持多种工作模式，比如：输入捕获、输出比较、PWM 生成和单脉冲输出
- 支持外部信号控制定时器
- 支持在多种模式下使用 DMA
- 支持增量式编码，定时器之间的级联和同步

15.2 原理和结构

图 15-1 通用定时器的结构框图



15.2.1 概述

如图 15-1 所示，通用定时器的结构大致可以分为三部分，即输入时钟部分，核心计数器部分和比较捕获通道部分。

通用定时器的时钟可以来自于 AHB 总线时钟（CK_INT），可以来自外部时钟输入引脚（TIMx_ETR），可以来自于其他具有时钟输出功能的定时器（ITRx），还可以来自于比较捕获通道的输入端（TIMx_CHx）。这些输入的时钟信号经过各种设定的滤波分频等操作后成为 CK_PSC 时钟，输出给核

心计数器部分。另外，这些复杂的时钟来源还可以作为 TRGO 输出给其他的定时器、ADC 和 DAC 等外设。

通用定时器的核心是一个 16 位计数器（CNT）。CK_PSC 经过预分频器（PSC）分频后，成为 CK_CNT 再最终输给 CNT，CNT 支持增计数模式、减计数模式和增减计数模式，并有一个自动重装值寄存器（ATRLR）在每个计数周期结束后为 CNT 重装载初始化值。

通用定时器拥有四组比较捕获通道，每组比较捕获通道都可以从专属的引脚上输入脉冲，也可以向引脚输出波形，即比较捕获通道支持输入和输出模式。比较捕获寄存器每个通道的输入都支持滤波、分频、边沿检测等操作，并支持通道间的互触发，还能为核心计数器 CNT 提供时钟。每个比较捕获通道都拥有一组比较捕获寄存器（CHxCVR），支持与主计数器（CNT）进行比较而输出脉冲。

15.2.2 通用定时器和高级定时器的区别

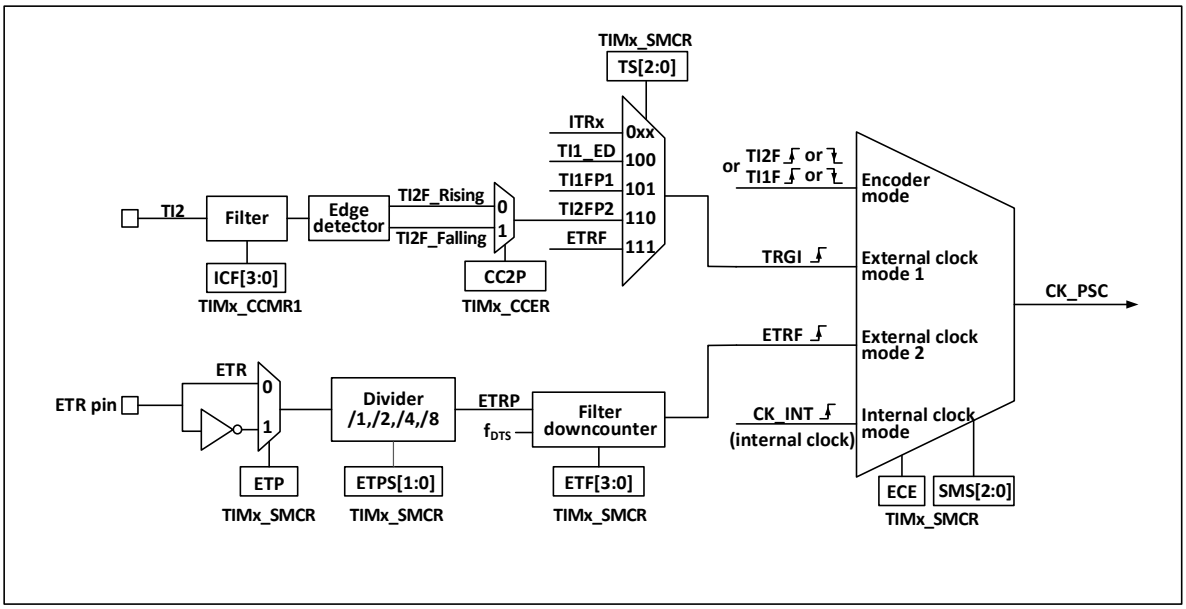
与高级定时器相比，通用定时器缺少以下功能：

1） 通用定时器缺少对核心计数器的计数周期进行计数的重复计数寄存器。
2） 通用定时器的比较捕获通道缺少死区产生，没有互补输出。
3） 通用定时器没有刹车信号机制。
4） 通用定时器的默认时钟 CK_INT 都来自 APB2，而高级定时器的 CK_INT 都来自 APB1。

15.2.3 时钟输入

本节论述 CK_PSC 的来源。此处截取通用定时器的整体结构框图的时钟源部分。

图 15-2 通用定时器 CK_PSC 来源框图



可选的输入时钟可以分为 4 类：

1） 外部时钟引脚（ETR）输入的路线：ETR→ETRP→ETRF；
2） 内部 APB 时钟输入路线：CK_INT；
3） 来自比较捕获通道引脚（TIMx_CHx）的路线：TIMx_CHx→TIx→TIxFPx，此路线也用于编码器模式；
4） 来自内部其他定时器的输入：ITRx。

通过决定 CK_PSC 来源的 SMS 的输入脉冲选择可以将实际的操作分为三类：

1） 选择内部时钟源（CK_INT）；
2） 外部时钟源模式 1；
3） 外部时钟源模式 2；

4） 编码器模式。

上文提到的 4 种时钟源来源都可通过这 4 种操作选定。

15.2.3.1 内部时钟源（CK_INT）

如果将 SMS 域保持为 000b 时启动通用定时器，那么就是选定内部时钟源（CK_INT）为时钟。此时 CK_INT 就是 CK_PSC。

15.2.3.2 外部时钟源模式 1

如果将 SMS 域设置为 111b 时，就会启用外部时钟源模式 1。启用外部时钟源 1 时，TRGI 被选定为 CK_PSC 的来源，值得注意的，用户还需要通过配置 TS 域来选择 TRGI 的来源。TS 域可选择以下几种脉冲作为时钟来源：

1） 内部触发（ITRx，x 为 0,1,2,3）；

2） 比较捕获通道 1 经过边缘检测器后的信号（TI1F_ED）；

3） 比较捕获通道的信号 TI1FP1、TI2FP2；

4） 来自外部时钟引脚输入的信号 ETRF。

15.2.3.3 外部时钟源模式 2

使用外部触发模式 2 能在外部时钟引脚输入的每一个上升沿或下降沿计数。将 ECE 位置位时，将使用外部时钟源模式 2。使用外部时钟源模式 2 时，ETRF 被选定为 CK_PSC。ETR 引脚经过可选的反相器（ETP），分频器（ETPS）后成为 ETRP，再经过滤波器（ETF）后即成为 ETRF。

在 ECE 位置位且将 SMS 设为 111b 时，那么，相当于 TS 选择 ETRF 为输入。

15.2.3.4 编码器模式

将 SMS 置为 001b,010b,011b 将会启用编码器模式。启用编码器模式可以选择在 TI1FP1 和 TI2FP2 中某一个特定的电平下以另一个跳变沿作为信号进行信号输出。此模式用于外接编码器使用的情况下。具体功能参考 14.3.9 节。

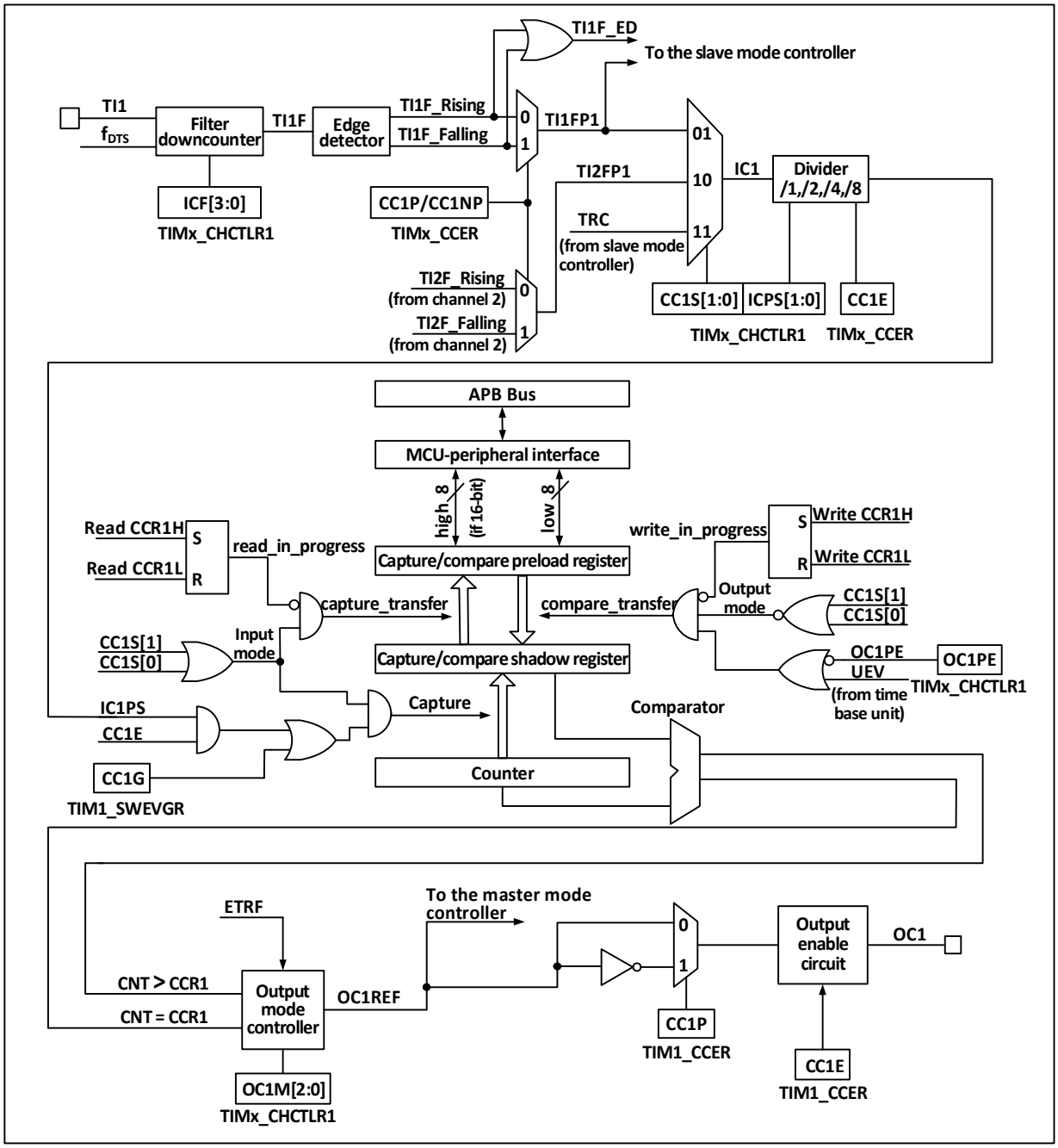
15.2.4 计数器和周边

CK_PSC 输入给预分频器（PSC）进行分频。PSC 是 16 位的，实际的分频系数相当于 R16_TIMx_PSC 的值+1。CK_PSC 经过 PSC 会成为 CK_INT。更改 R16_TIM1_PSC 的值并不会实时生效，而会在更新事件后更新给 PSC。更新事件包括 UG 位清零和复位。

15.2.5 比较捕获通道

比较捕获通道是定时器实现复杂功能的核心，它的核心是比较捕获寄存器，辅以外围输入部分的数字滤波，分频和通道间复用，输出部分的比较器和输出控制组成。比较捕获通道的结构框图如图 15-3 所示。

图 15-3 比较捕获通道的结构框图



信号从通道 x 引脚输入进来后可选做为 TIx(TI1 的来源可以不只是 CH1，见定时器的框图 14-1)，TI1 经过滤波器（ICF[3:0]）生成 TI1F，再经过边沿检测器分成 TI1F_Rising 和 TI1F_Falling，这两个信号经过选择(CC1P)生成 TI1FP1，TI1FP1 和来自通道 2 的 TI2FP1 一起送给 CC1S 选择成为 IC1，经过 ICPS 分频后送给比较捕获寄存器。

比较捕获寄存器由一个预装载寄存器和一个影子寄存器组成，读写过程仅操作预装载寄存器。在捕获模式下，捕获发生在影子寄存器上，然后复制到预装载寄存器；在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容与核心计数器（CNT）进行比较。

15.3 功能和实现

通用定时器复杂功能的实现都是对定时器的比较捕获通道、时钟输入电路和计数器及周边组件进行操作实现的。定时器的时钟输入可以来自于包括比较捕获通道的输入在内的多个时钟源。对比较捕

获寄存通道和时钟源选择的操作直接决定其功能。比较捕获通道是双向的，可以工作在输入和输出模式。

15.3.1 输入捕获模式

输入捕获模式是定时器的基本功能之一。输入捕获模式的原理是，当检测到 ICxPS 信号上确定的边沿后，则产生捕获事件，计数器当前的值会被锁存到比较捕获寄存器（R16_TIMx_CHCTLRx）中。发生捕获事件时，CCxIF（在 R16_TIMx_INTFR 中）被置位，如果使能了中断或者 DMA，还会产生相应中断或者 DMA。如果发生捕获事件时，CCxIF 已经被置位了，那么 CCxOF 位会被置位。CCxIF 可由软件清除，也可以通过读取比较捕获寄存器由硬件清除。CCxOF 由软件清除。

举个通道 1 的例子来说明使用输入捕获模式的步骤，如下：

1） 配置 CCxS 域，选择 ICx 信号的来源。比如设为 10b，选择 TI1FP1 作为 IC1 的来源，不可以使用默认设置，CCxS 域默认是使比较捕获模块作为输出通道；
2） 配置 ICxF 域，设定 TI 信号的数字滤波器。数字滤波器会以确定的频率，采样确定的次数，再输出一个跳变。这个采样频率和次数是通过 ICxF 来确定的；
3） 配置 CCxP 位，设定 TIxFPx 的极性。比如保持 CC1P 位为低，选择上升沿跳变；
4） 配置 ICxPS 域，设定 ICx 信号成为 ICxPS 之间的分频系数。比如保持 ICxPS 为 00b，不分频；
5） 配置 CCxE 位，允许捕获核心计数器（CNT）的值到比较捕获寄存器中。置 CC1E 位；
6） 根据需要配置 CCxIE 和 CCxDE 位，决定是否允许使能中断或者 DMA。

至此已经将比较捕获通道配置完成。

当 TI1 输入了一个被捕获的脉冲时，核心计数器（CNT）的值会被记录到比较捕获寄存器中，CC1IF 被置位，当 CC1IF 在之前就已经被置位时，CC1OF 位也会被置位。如果 CC1IE 位，那么会产生一个中断；如果 CC1DE 被置位，会产生一个 DMA 请求。可以通过写事件产生寄存器的方式（R16_TIMx_SWEVGR）的方式由软件产生一个输入捕获事件。

15.3.2 比较输出模式

比较输出模式是定时器的基本功能之一。比较输出模式的原理是在核心计数器（CNT）的值与比较捕获寄存器的值一致时，输出特定的变化或波形。OCxM 域（在 R16_TIMx_CHCTLRx 中）和 CCxP 位（在 R16_TIMx_CCER 中）决定输出的是确定的高低电平还是电平翻转。产生比较一致事件时还会置 CCxIF 位，如果预先置了 CCxIE 位，则会产生一个中断；如果预先设置了 CCxDE 位，则会产生一个 DMA 请求。

配置为比较输出模式的步骤为下：

1） 配置核心计数器（CNT）的时钟源和自动重装值；
2） 设置好需要对比的计数值到比较捕获寄存器（R16_TIMx_CHxCVR）中；
3） 如果需要产生中断，置 CCxIE 位；
4） 保持 OCxPE 为 0，禁用比较捕获寄存器的预装载寄存器；
5） 设定输出模式，设置 OCxM 域和 CCxP 位；
6） 使能输出，置 CCxE 位；
7） 置 CEN 位启动定时器；

15.3.3 强制输出模式

定时器的比较捕获通道的输出模式可以由软件强制输出确定的电平，而不依赖比较捕获寄存器的影子寄存器和核心计数器的比较。

具体的做法是将 OCxM 置为 100b，即为强制将 OCxREF 置为低；或者将 OCxM 置为 101b，即为强制将 OCxREF 置为高。

需要注意的是，将 OCxM 强制置为 100b 或者 101b，内部主计数器和比较捕获寄存器的比较过程

还在进行，相应的标志位还在置位，中断和 DMA 请求还在产生。

15.3.4 PWM 输入模式

PWM 输入模式是用来测量 PWM 的占空比和频率的，是输入捕获模式的一种特殊情况。除下列区别外，操作和输入捕获模式相同：PWM 占用两个比较捕获通道，且两个通道的输入极性设为相反，其中一个信号被设为触发输入，SMS 设为复位模式。

例如，测量从 TI1 输入的 PWM 波的周期和频率，需要进行以下操作：

1） 将 TI1(TI1FP1)设为 IC1 信号的输入。将 CC1S 置为 01b；
2） 将 TI1FP1 置为上升沿有效。将 CC1P 保持为 0；
3） 将 TI1(TI1FP2)置为 IC2 信号的输入。将 CC2S 置为 10b；
4） 选 TI1FP2 置为下降沿有效。将 CC2P 置为 1；
5） 时钟源的来源选择 TI1FP1。将 TS 设为 101b；
6） 将 SMS 设为复位模式，即 100b；
7） 使能输入捕获。CC1E 和 CC2E 置位。

15.3.5 PWM 输出模式

PWM 输出模式是定时器的基本功能之一。PWM 输出模式最常见的是使用重装值确定 PWM 频率，使用捕获比较寄存器确定占空比的方法。将 OCxM 域中置 110b 或者 111b 使用 PWM 模式 1 或者模式 2，置 OCxPE 位使能预装载寄存器，最后置 ARPE 位使能预装载寄存器的自动重装载。在发生一个更新事件时，预装载寄存器的值才能被送到影子寄存器，所以在核心计数器开始计数之前，需要置 UG 位来初始化所有寄存器。在 PWM 模式下，核心计数器和比较捕获寄存器一直在进行比较，根据 CMS 位，定时器能够输出边沿对齐或者中央对齐的 PWM 信号。

- 边沿对齐

使用边沿对齐时，核心计数器增计数或者减计数，在 PWM 模式 1 的情景下，在核心计数器的值大于比较捕获寄存器时，OCxREF 上升为高；当核心计数器的值小于比较捕获寄存器时（比如核心计数器增长到 R16_TIMx_ATRLR 的值而恢复成全 0 时），OCxREF 下降为低。

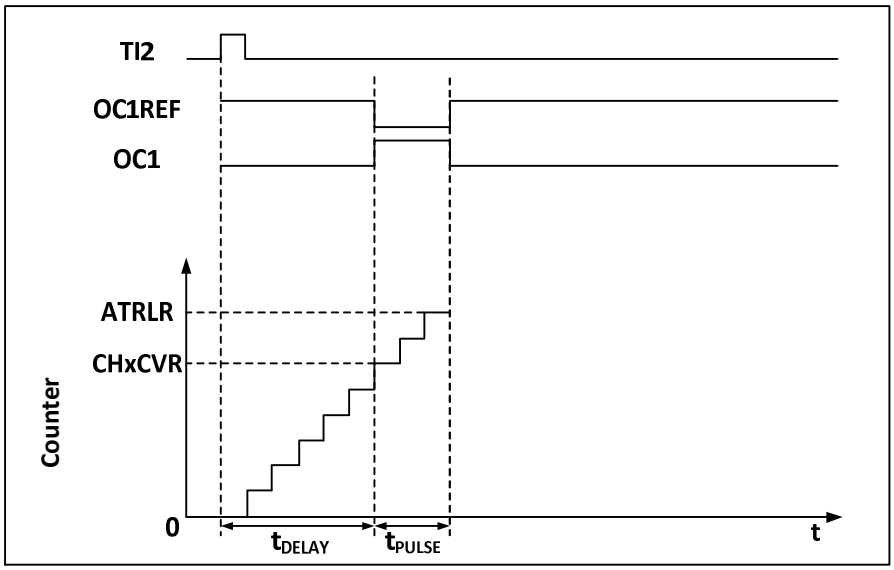
- 中央对齐

使用中央对齐模式时，核心计数器运行在增计数和减计数交替进行的模式下，OCxREF 在核心计数器和比较捕获寄存器的值一致时进行上升和下降的跳变。但比较标志在三种中央对齐模式下，置位的时机有所不同。在使用中央对齐模式时，最好在启动核心计数器之前产生一个软件更新标志（置 UG 位）。

15.3.6 单脉冲模式

单脉冲模式可以响应一个特定的事件，在一个延迟之后产生一个脉冲，延迟和脉冲的宽度可编程。置 OPM 位可以使核心计数器在产生下一个更新事件 UEV 时（计数器翻转到 0）停止。

图 15-4 事件产生和脉冲响应



如图 15-4 所示，需要在 TI2 输入引脚上检测到一个上升沿开始，延迟 Tdelay 之后，在 OC1 上产生一个长度为 Tpulse 的正脉冲：

1） 设定 TI2 为触发。置 CC2S 域为 01b，把 TI2FP2 映射到 TI2；置 CC2P 位为 0b，TI2FP2 设为上升沿检测；置 TS 域为 110b，TI2FP2 设为触发源；置 SMS 域为 110b，TI2FP2 被用来启动计数器；

2） Tdelay 由比较捕获寄存器定义，Tpulse 由自动重装值寄存器的值和比较捕获寄存器的值确定。

15.3.7 编码器模式

编码器模式是定时器的一个典型应用，可以用来接入编码器的双相输出，核心计数器的计数方向和编码器的转轴方向同步，编码器每输出一个脉冲就会使核心计数器加一或减一。使用编码器的步骤为：将 SMS 域置为 001b（只在 TI2 边沿计数）、010b（只在 TI1 边沿计数）或者 011b（在 TI1 和 TI2 双边沿计数），将编码器接到比较捕获通道 1、2 的输入端，设一个重装值计数器的值，这个值可以设的大一点。在编码器模式时，定时器内部的比较捕获寄存器，预分频器，重复计数寄存器等都正常工作。下表表明了计数方向和编码器信号的关系。

表 15-1 定时器编码器模式的计数方向和编码器信号之间的关系

<table>
<tr><th rowspan="2">计数有效边沿</th><th rowspan="2">相对信号的电平</th><th colspan="2">TI1FP1 信号边沿</th><th colspan="2">TI2FP2 信号</th></tr>
<tr><th>上升沿</th><th>下降沿</th><th>上升沿</th><th>下降沿</th></tr>
<tr><td rowspan="2">仅在 TI1 边沿计数</td><td>高</td><td>向下计数</td><td>向上计数</td><td colspan="2" rowspan="2">不计数</td></tr>
<tr><td>低</td><td>向上计数</td><td>向下计数</td></tr>
<tr><td rowspan="2">仅在 TI2 边沿计数</td><td>高</td><td colspan="2" rowspan="2">不计数</td><td>向上计数</td><td>向下计数</td></tr>
<tr><td>低</td><td>向下计数</td><td>向上计数</td></tr>
<tr><td rowspan="2">在 TI1 和 TI2 双边沿计数</td><td>高</td><td>向下计数</td><td>向上计数</td><td>向上计数</td><td>向下计数</td></tr>
<tr><td>低</td><td>向上计数</td><td>向下计数</td><td>向下计数</td><td>向上计数</td></tr>
</table>

15.3.8 定时器同步模式

定时器能够输出时钟脉冲（TRGO），也能接收其他定时器的输入（ITRx）。不同的定时器的 ITRx 的来源（别的定时器的 TRGO）是不一样的。定时器内部触发连接如表 12-2 所示。

表 15-2 GTPM 内部触发连接

从定时器	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM2	TIM1	TIM8/USB/ETH	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM2	TIM3	TIM4	TIM8

15.3.9 调试模式

当系统进入调试模式时，根据 DBG 模块的设置可以控制定时器继续运转或者停止。

15.4 寄存器描述

表 15-3 TIM2 相关寄存器列表

名称	偏移地址	描述	复位值
R16_TIM2_CTLR1	0x40000000	TIM2 控制寄存器 1	0x0000
R16_TIM2_CTLR2	0x40000004	TIM2 控制寄存器 2	0x0000
R16_TIM2_SMCFGR	0x40000008	TIM2 从模式控制寄存器	0x0000
R16_TIM2_DMAINTENR	0x4000000C	TIM2 DMA/中断使能寄存器	0x0000
R16_TIM2_INTFR	0x40000010	TIM2 中断状态寄存器	0x0000
R16_TIM2_SWEVGR	0x40000014	TIM2 事件产生寄存器	0x0000
R16_TIM2_CHCTLR1	0x40000018	TIM2 比较/捕获控制寄存器 1	0x0000
R16_TIM2_CHCTLR2	0x4000001C	TIM2 比较/捕获控制寄存器 2	0x0000
R16_TIM2_CCER	0x40000020	TIM2 比较/捕获使能寄存器	0x0000
R16_TIM2_CNT	0x40000024	TIM2 计数器	0x0000
R16_TIM2_PSC	0x40000028	TIM2 计数时钟预分频器	0x0000
R16_TIM2_ATRLR	0x4000002C	TIM2 自动重装值寄存器	0x0000
R16_TIM2_CH1CVR	0x40000034	TIM2 比较/捕获寄存器 1	0x0000
R16_TIM2_CH2CVR	0x40000038	TIM2 比较/捕获寄存器 2	0x0000
R16_TIM2_CH3CVR	0x4000003C	TIM2 比较/捕获寄存器 3	0x0000
R16_TIM2_CH4CVR	0x40000040	TIM2 比较/捕获寄存器 4	0x0000
R16_TIM2_DMACFGR	0x40000048	TIM2 DMA 控制寄存器	0x0000
R16_TIM2_DMAADR	0x4000004C	TIM2 连续模式的 DMA 地址寄存器	0x0000

表 15-4 TIM3 相关寄存器列表

名称	偏移地址	描述	复位值
R16_TIM3_CTLR1	0x40000400	TIM3 控制寄存器 1	0x0000
R16_TIM3_CTLR2	0x40000404	TIM3 控制寄存器 2	0x0000
R16_TIM3_SMCFGR	0x40000408	TIM3 从模式控制寄存器	0x0000
R16_TIM3_DMAINTENR	0x4000040C	TIM3 DMA/中断使能寄存器	0x0000
R16_TIM3_INTFR	0x40000410	TIM3 中断状态寄存器	0x0000
R16_TIM3_SWEVGR	0x40000414	TIM3 事件产生寄存器	0x0000
R16_TIM3_CHCTLR1	0x40000418	TIM3 比较/捕获控制寄存器 1	0x0000
R16_TIM3_CHCTLR2	0x4000041C	TIM3 比较/捕获控制寄存器 2	0x0000
R16_TIM3_CCER	0x40000420	TIM3 比较/捕获使能寄存器	0x0000

R16_TIM3_CNT	0x40000424	TIM3 计数器	0x0000
R16_TIM3_PSC	0x40000428	TIM3 计数时钟预分频器	0x0000
R16_TIM3_ATRLR	0x4000042C	TIM3 自动重装值寄存器	0x0000
R16_TIM3_CH1CVR	0x40000434	TIM3 比较/捕获寄存器 1	0x0000
R16_TIM3_CH2CVR	0x40000438	TIM3 比较/捕获寄存器 2	0x0000
R16_TIM3_CH3CVR	0x4000043C	TIM3 比较/捕获寄存器 3	0x0000
R16_TIM3_CH4CVR	0x40000440	TIM3 比较/捕获寄存器 4	0x0000
R16_TIM3_DMACFGR	0x40000448	TIM3 DMA 控制寄存器	0x0000
R16_TIM3_DMAADR	0x4000044C	TIM3 连续模式的 DMA 地址寄存器	0x0000

表 15-5 TIM4 相关寄存器列表

名称	偏移地址	描述	复位值
R16_TIM4_CTLR1	0x40000800	TIM4 控制寄存器 1	0x0000
R16_TIM4_CTLR2	0x40000804	TIM4 控制寄存器 2	0x0000
R16_TIM4_SMCFGR	0x40000808	TIM4 从模式控制寄存器	0x0000
R16_TIM4_DMAINTENR	0x4000080C	TIM4 DMA/中断使能寄存器	0x0000
R16_TIM4_INTFR	0x40000810	TIM4 中断状态寄存器	0x0000
R16_TIM4_SWEVGR	0x40000814	TIM4 事件产生寄存器	0x0000
R16_TIM4_CHCTLR1	0x40000818	TIM4 比较/捕获控制寄存器 1	0x0000
R16_TIM4_CHCTLR2	0x4000081C	TIM4 比较/捕获控制寄存器 2	0x0000
R16_TIM4_CCER	0x40000820	TIM4 比较/捕获使能寄存器	0x0000
R16_TIM4_CNT	0x40000824	TIM4 计数器	0x0000
R16_TIM4_PSC	0x40000828	TIM4 计数时钟预分频器	0x0000
R16_TIM4_ATRLR	0x4000082C	TIM4 自动重装值寄存器	0x0000
R16_TIM4_CH1CVR	0x40000834	TIM4 比较/捕获寄存器 1	0x0000
R16_TIM4_CH2CVR	0x40000838	TIM4 比较/捕获寄存器 2	0x0000
R16_TIM4_CH3CVR	0x4000083C	TIM4 比较/捕获寄存器 3	0x0000
R16_TIM4_CH4CVR	0x40000840	TIM4 比较/捕获寄存器 4	0x0000
R16_TIM4_DMACFGR	0x40000848	TIM4 DMA 控制寄存器	0x0000
R16_TIM4_DMAADR	0x4000084C	TIM4 连续模式的 DMA 地址寄存器	0x0000

15.4.1 控制寄存器 1（TIMx_CTLR1）（x=2/3/4）

偏移地址：0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN

位	名称	访问	描述	复位值
[15:10]	Reserved	RO	保留。	0
[9:8]	CKD	RW	这 2 位定义在定时器时钟(CK_INT)频率、数字滤波器所用的采样时钟之间的分频比例： 00：Tdts=Tck_int； 01：Tdts= 2xTck_int； 10：Tdts= 4xTck_int； 11：保留。	0

7	ARPE	RW	自动重装预装使能位： 1：使能自动重装值寄存器（ATRLR）； 0：禁止自动重装值寄存器（ATRLR）。	0
[6:5]	CMS	RW	中央对齐模式选择： 00：边沿对齐模式。计数器依据方向位(DIR)向上或向下计数。 01：中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道(CHCTLRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向下计数时被设置。 10：中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道(CHCTLRx 寄存器中 CCxS=00)的输出比较中断标志位，只在计数器向上计数时被设置。 11：中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道(CHCTLRx 寄存器中 CCxS=00)的输出比较中断标志位，在计数器向上和向下计数时均被设置。 *注：在计数器使能时(CEN=1)，不允许从边沿对齐模式转换到中央对齐模式。*	0
4	DIR	RW	计数器方向： 0：计数器的计数模式为增计数； 1：计数器的计数模式为减计数。 *注：当计数器配置为中央对齐模式或编码器模式时，该位无效。*	0
3	OPM	RW	单脉冲模式。 1：在发生下一次更新事件(清除 CEN 位)时，计数器停止； 0：在发生下一次更新事件时，计数器不停止。	0
2	URS	RW	更新请求源，软件通过该位选择 UEV 事件的源。 1：如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求； 0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： -计数器溢出/下溢 -设置 UG 位 -从模式控制器产生的更新	0
1	UDIS	RW	禁止更新，软件通过该位允许/禁止 UEV 事件的产生。 1：禁止 UEV。不产生更新事件，各寄存器(ATRLR、PSC、CHCTLRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。	0

			0：允许 UEV。更新(UEV)事件由下述任一事件产生： -计数器溢出/下溢 -设置 UG 位 -从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。	
0	CEN	RW	使能计数器 (Counter enable)。 1：使能计数器； 0：禁止计数器。 *注：在软件设置了 CEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。*	0

15.4.2 控制寄存器 2（TIMx_CTLR2）（x=2/3/4）

偏移地址：0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI1S	MMS[2:0]			CCDS	CCUS	Reserved	CCPC

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
7	TI1S	RW	TI1 选择： 1：TIMx_CH1、TIMx_CH2 和 TIMx_CH3 引脚经异或后连到 TI1 输入； 0：TIMx_CH1 引脚直连到 TI1 输入。	0
[6:4]	MMS	RW	主模式选择：这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： 000：复位－UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟； 001:使能－ 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见 TIMx_SMCFGR 寄存器中 MSM 位的描述)； 010：更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器； 011:比较脉冲,在发生一次捕获或一次比较成功时,当要设置 CC1IF 标志时(即使它已经为高)，触发输出送出一个正脉冲(TRGO)；	0

			100：OC1REF 信号被用于作为触发输出(TRGO； 101：OC2REF 信号被用于作为触发输出(TRGO)； 110：OC3REF 信号被用于作为触发输出(TRGO)； 111：OC4REF 信号被用于作为触发输出(TRGO)。	
3	CCDS	RW	1：当发生更新事件时，送出 CHxCVR 的 DMA 请求； 0：当发生 CHxCVR 时，产生 CHxCVR 的 DMA 请求。	0
2	CCUS	RW	比较捕获控制更新选择位。 1：如果 CCPC 置位，可以通过设置 COM 位或 TRGI 上的一个上升沿更新它们； 0：如果 CCPC 置位，只能通过设置 COM 位更新它们。 *注：该位只对具有互补输出的通道起作用。*	0
1	Reserved	RO	保留。	0
0	CCPC	RW	比较捕获预装载控制位。 1：CCxE，CCxNE 和 OCxM 位是预装载的，设置该位后，它们只在设置了 COM 位后被更新； 0：CCxE，CCxNE 和 OCxM 位不是预装载的。 *注：该位只对具有互补输出的通道起作用。*	0

15.4.3 从模式控制寄存器（TIMx_SMCFGR）（x=2/3/4）

偏移地址：0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		

位	名称	访问	描述	复位值
15	ETP	RO	ETR 触发极性选择，该位选择是直接输入 ETR 还是输入 ETR 的反相。 1：将 ETR 反相，低电平或下降沿有效； 0：ETR，高电平或上升沿有效。	0
14	ECE	RW	外部时钟模式 2 启用选择。 1：使能外部时钟模式 2； 0：禁用外部时钟模式 2。 *注 1：从模式可以与外部时钟模式 2 同时使用：复位模式，门控模式和触发模式；但是，这时 TRGI 不能连到 ETRF(TS 位不能是 111b)。* *注 2：外部时钟模式 1 和外部时钟模式 2 同时被使能时，外部时钟的输入是 ETRF。*	0
[13:12]	ETPS	RW	外部触发信号(ETRP)分频，这个信号频率最大不能超过是 TIMxCLK 频率的 1/4，可以通过这个域来降频。 00：关闭预分频； 01：ETRP 频率除以 2； 10：ETRP 频率除以 4； 11：ETRP 频率除以 8。	0
[11:8]	ETF	RW	外部触发滤波，实际上，数字滤波器是一个事件计数器，它使用一定的采样的频率，记录到 N 个事件	0

			后会产生一个输出的跳变。 0000：无滤波器，以 Fdts 采样； 0001：采样频率 Fsampling=Fck_int，N=2； 0010：采样频率 Fsampling=Fck_int，N=4； 0011：采样频率 Fsampling=Fck_int，N=8； 0100：采样频率 Fsampling=Fdts/2，N=6； 0101：采样频率 Fsampling=Fdts/2，N=8； 0110：采样频率 Fsampling=Fdts/4，N=6； 0111：采样频率 Fsampling=Fdts/4，N=8； 1000：采样频率 Fsampling=Fdts/8，N=6； 1001：采样频率 Fsampling=Fdts/8，N=8； 1010：采样频率 Fsampling=Fdts/16，N=5； 1011：采样频率 Fsampling=Fdts/16，N=6； 1100：采样频率 Fsampling=Fdts/16，N=8； 1101：采样频率 Fsampling=Fdts/32，N=5； 1110：采样频率 Fsampling=Fdts/32，N=6； 1111：采样频率 Fsampling=Fdts/32，N=8。	
7	MSM	RW	主/从模式选择： 1：触发输入(TRGI)上的事件被延迟了，以允许在当前定时器(通过 TRGO)与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的； 0：不发挥作用。	0
[6:4]	TS	RW	触发选择域，这 3 位选择用于同步计数器的触发输入源。 000：内部触发 0(ITR0)； 100：TI1 的边沿检测器(TI1F_ED)； 001：内部触发 1(ITR1)； 101：滤波后的定时器输入 1(TI1FP1)； 010：内部触发 2(ITR2)； 110：滤波后的定时器输入 2(TI2FP2)； 011：内部触发 3(ITR3)； 111：外部触发输入(ETRF)； 以上只有在 SMS 为 0 时改变。	0
3	Reserved	RO	保留。	0
[2:0]	SMS	RW	输入模式选择域。选择核心计数器的时钟和触发模式。 000：由内部时钟 CK_INT 驱动； 001：编码器模式 1，根据 TI1FP1 的电平，核心计数器在 TI2FP2 的边沿增减计数； 010：编码器模式 2，根据 TI2FP2 的电平，核心计数器在 TI1FP1 的边沿增减计数； 011：编码器模式 3，根据另一个信号的输入电平，核心计数器在 TI1FP1 和 TI2FP2 的边沿增减计数； 100：复位模式，触发输入(TRGI)的上升沿将初始化计数器，并且产生一个更新寄存器的信号；	0

			101：门控模式，当触发输入(TRGI)为高时，计数器的时钟开启；在触发输入变为低，计数器停止，计数器的启停都是受控的； 110：触发模式，计数器在触发输入 TRGI 的上升沿启动，只有计数器的启动是受控的； 111：外部时钟模式 1，选中的触发输入(TRGI)的上升沿驱动计数器。	

15.4.4 DMA/中断使能寄存器（TIMx_DMAINTENR）（x=2/3/4）

偏移地址：0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Reserved	TIE	保留	CC4IE	CC3IE	CC2IE	CC1IE	UIE

位	名称	访问	描述	复位值
15	Reserved	RO	保留。	0
14	TDE	RW	触发 DMA 请求使能位。 1：允许触发 DMA 请求； 0：禁止触发 DMA 请求。	0
13	COMDE	RW	COM 的 DMA 请求使能位。 1：允许 COM 的 DMA 请求； 0：禁止 COM 的 DMA 请求。	0
12	CC4DE	RW	比较捕获通道 4 的 DMA 请求使能位。 1：允许比较捕获通道 4 的 DMA 请求； 0：禁止比较捕获通道 4 的 DMA 请求。	0
11	CC3DE	RW	比较捕获通道 3 的 DMA 请求使能位。 1：允许比较捕获通道 3 的 DMA 请求； 0：禁止比较捕获通道 3 的 DMA 请求。	0
10	CC2DE	RW	比较捕获通道 2 的 DMA 请求使能位。 1：允许比较捕获通道 2 的 DMA 请求； 0：禁止比较捕获通道 2 的 DMA 请求。	0
9	CC1DE	RW	比较捕获通道 1 的 DMA 请求使能位。 1：允许比较捕获通道 1 的 DMA 请求； 0：禁止比较捕获通道 1 的 DMA 请求。	0
8	UDE	RW	更新的 DMA 请求使能位。 1：允许更新的 DMA 请求； 0：禁止更新的 DMA 请求。	0
7	Reserved	RO	保留。	0
6	TIE	RW	触发中断使能位。 1：使能触发中断； 0：禁止触发中断。	0
5	Reserved	RO	保留。	0
4	CC4IE	RW	比较捕获通道 4 中断使能位。 1：允许比较捕获通道 4 中断； 0：禁止比较捕获通道 4 中断。	0
3	CC3IE	RW	比较捕获通道 3 中断使能位。	0

			1：允许比较捕获通道 3 中断； 0：禁止比较捕获通道 3 中断。	
2	CC2IE	RW	比较捕获通道 2 中断使能位。 1：允许比较捕获通道 2 中断； 0：禁止比较捕获通道 2 中断。	0
1	CC1IE	RW	比较捕获通道 1 中断使能位。 1：允许比较捕获通道 1 中断； 0：禁止比较捕获通道 1 中断。	0
0	UIE	RW	更新中断使能位。 1：允许更新中断； 0：禁止更新中断。	0

15.4.5 中断状态寄存器（R16_TIMx_INTFR）（x=2/3/4）

偏移地址：0x10

15 14 13	12	11	10	9	8 7	6	5	4	3	2	1	0
Reserved	CC4OF	CC3OF	CC2OF	CC1OF	Reserved	TIF	Reserved	CC4IF	CC3IF	CC2IF	CC1IF	UIF

位	名称	访问	描述	复位值
[15:13]	Reserved	RO	保留。	0
12	CC4OF	WO	比较捕获通道 4 重复捕获标志位。	0
11	CC3OF	WO	比较捕获通道 3 重复捕获标志位。	0
10	CC2OF	WO	比较捕获通道 2 重复捕获标志位。	0
9	CC1OF	WO	比较捕获通道 1 重复捕获标志位，仅用于比较捕获通道被配置为输入捕获模式时。该标记由硬件置位，软件写 0 可清除此位。 1：计数器的值被捕获到捕获比较寄存器时，CC1IF 的状态已经被置位； 0：无重复捕获产生。	0
[8:7]	保留	RO	保留。	0
6	TIF	WO	触发器中断标志位，当发生触发事件时由硬件对该位置位，由软件清零。触发事件包括从除门控模式外的其它模式时，在 TRGI 输入端检测到有效边沿，或门控模式下的任一边沿。 1：触发器事件产生； 0：无触发器事件产生。	0
5	保留	RO	保留。	0
4	CC4IF	WO	比较捕获通道 4 中断标志位。	0
3	CC3IF	WO	比较捕获通道 3 中断标志位。	0
2	CC2IF	WO	比较捕获通道 2 中断标志位。	0
1	CC1IF	WO	比较捕获通道 1 中断标志位。 如果比较捕获通道配置为输出模式，当计数器值与比较值匹配时该位由硬件置位，但在中心对称模式下除外。该位由软件清零。 1：核心计数器的值与比较捕获寄存器 1 的值匹配； 0：无匹配发生。	0

			如果比较捕获通道 1 配置为输入模式，当捕获事件发生时该位由硬件置位，它由软件清零或通过读比较捕获寄存器清零。 1：计数器值已被捕获比较捕获寄存器 1； 0：无输入捕获产生。	
0	UIF	WO	更新中断标志位，当产生更新事件时该位由硬件置位，由软件清零。 1：更新中断产生； 0：无更新事件产生。 以下情形会产生更新事件： 若 UDIS=0，当重复计数器数值上溢或下溢时； 若 URS=0、UDIS=0，当置 UG 位时，或当通过软件对计数器核心计数器重新初始化时； 若 URS=0、UDIS=0，当计数器 CNT 被触发事件重新初始化时；	0

15.4.6 事件产生寄存器（TIMx_SWEVGR）（x=2/3/4）

偏移地址：0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
7	BG	WO	刹车事件产生位，此位由软件置位和清零，用来产生一个刹车事件。 1：产生一个刹车事件。此时 MOE=0、BIF=1，若使能对应的中断和 DMA，则产生相应的中断和 DMA； 0：无动作。	0
6	TG	WO	触发事件产生位，该位由软件置位，硬件清零，用于产生一个触发事件。 1：产生一个触发事件，TIF 被置位，若使能对应的中断和 DMA，则产生相应的中断和 DMA； 0：无动作。	0
5	COMG	WO	比较捕获控制更新产生位。产生比较捕获控制更新事件。该位由软件置位，由硬件自动清零。 1：当 CCPC=1，允许更新 CCxE、CCxNE、OCxM 位； 0：无动作。 *注：该位只对拥有互补输出的通道（通道 1，2，3）有效。*	0
4	CC4G	WO	比较捕获事件产生位 4。产生比较捕获事件 4。	0
3	CC3G	WO	比较捕获事件产生位 3。产生比较捕获事件 3。	0
2	CC2G	WO	比较捕获事件产生位 2。产生比较捕获事件 2。	0
1	CC1G	WO	比较捕获事件产生位 1，产生比较捕获事件 1。该位由软件置位，由硬件清零。用于产生一个比较捕获事件。	0

			1：在比较捕获通道 1 上产生一个比较捕获事件： 若比较捕获通道 1 配置为输出：置 CC1IF 位。若使能对应的中断和 DMA，则产生相应的中断和 DMA； 若比较捕获通道 1 配置为输入：当前核心计数器的值被捕获至比较捕获寄存器 1；置 CC1IF 位，若使能了对应的中断和 DMA，则产生相应的中断和 DMA。若 CC1IF 已经置位，则置 CC1OF 位。 0：无动作。	
0	UG	WO	更新事件产生位，产生更新事件。该位由软件置位，由硬件自动清零。 1：初始化计数器，并产生一个更新事件； 0：无动作。 *注：预分频器的计数器也被清零，但是预分频系数不变。若在中心对称模式下或增计数模式下则核心计数器被清零；若减计数模式下则核心计数器取重装值寄存器的值。*	0

15.4.7 比较/捕获控制寄存器 1（TIMx_CHCTLR1）（x=2/3/4）

偏移地址：0x18

通道可用于输入(捕获模式)或输出(比较模式)，通道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			

比较模式（引脚方向为输出）：

位	名称	访问	描述	复位值
15	OC2CE	RW	比较捕获通道 2 清零使能位。 1:一旦检测到 ETRF 输入高电平,清除 OC2REF 位零； 0：OC2REF 不受 ETRF 输入的影响。	0
[14:12]	OC2M	RW	比较捕获通道 2 模式设置域。 该 3 位定义了输出参考信号 OC2REF 的动作，而 OC2REF 决定了 OC2、OC2N 的值。OC2REF 是高电平有效，而 OC2 和 OC2N 的有效电平取决于 CC2P、CC2NP 位。 000：冻结。比较捕获寄存器的值与核心计数器间的比较值对 OC1REF 不起作用； 001：强制设为有效电平。当核心计数器与比较捕获寄存器 1 的值相同时，强制 OC1REF 为高； 010：强制设为无效电平。当核心计数器的值与比较捕获寄存器 1 相同时，强制 OC1REF 为低； 011：翻转。当核心计数器与比较捕获寄存器 1 的值相同时，翻转 OC1REF 的电平。 100：强制为无效电平。强制 OC1REF 为低。 101：强制为有效电平。强制 OC1REF 为高。	0

			110：PWM 模式 1：在向上计数时，一旦核心计数器大于比较捕获寄存器的值时通道 1 为无效电平，否则为有效电平；在向下计数时，一旦核心计数器大于比较捕获寄存器的值时通道 1 为有效电平，否则为无效电平。 111：PWM 模式 2：在向上计数时，一旦核心计数器大于比较捕获寄存器的值时，通道 1 为有效电平，否则为无效电平；在向下计数时，一旦核心计数器大于比较捕获寄存器的值时，通道 1 为无效电平，否则为有效电平(OC1REF=1)。 *注：一旦 LOCK 级别设为 3 并且 CC1S=00b 则该位不能被修改。在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。*	
11	OC2PE	RW	比较捕获寄存器 2 预装载使能位。 1：开启比较捕获寄存器的预装载功能，读写操作仅对预装载寄存器操作，比较捕获寄存器 1 的预装载值在更新事件到来时被加载至当前影子寄存器中； 0：禁止比较捕获寄存器 2 的预装载功能，可随时写入比较捕获寄存器 2，并且新写入的数值立即起作用。 *注：一旦 LOCK 级别设为 3 并且 CC1S=00，则该位不能被修改。仅仅在单脉冲模式下(OPM=1)可以在未确认预装载寄存器情况下使用 PWM 模式，否则其动作不确定。*	0
10	OC2FE	RW	比较捕获通道 2 快速使能位，该位用于加快比较捕获通道输出对触发输入事件的响应。 1:输入到触发器的有效沿的作用就像发生了一次比较匹配。因此，OC 被设置为比较电平而与比较结果无关。采样触发器的有效沿和比较捕获通道 2 输出间的延时被缩短为 3 个时钟周期； 0：根据计数器与比较捕获寄存器 2 的值，比较捕获通道 2 正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活比较捕获通道 2 输出的最小延时为 5 个时钟周期。 OC2FE 只在通道被配置成 PWM1 或 PWM2 模式时起作用；	0
[9:8]	CC2S	RW	比较捕获通道 2 输入选择域。 00：比较捕获通道 2 被配置为输出； 01：比较捕获通道 2 被配置为输入，IC2 映射在 TI2 上； 10：比较捕获通道 2 被配置为输入，IC2 映射在 TI1 上； 11：比较捕获通道 2 被配置为输入，IC2 映射在 TRC 上。此模式仅工作在内部触发器输入被选中时(由 TS 位选择)。	0

			注：比较捕获通道 2 仅在通道关闭时(CC2E 为零时)才是可写的。	
7	OC1CE	RW	比较捕获通道 1 清零使能位。	0
[6:4]	OC1M	RW	比较捕获通道 1 模式设置域。	0
3	OC1PE	RW	比较捕获寄存器 1 预装载使能位。	0
2	OC1FE	RW	比较捕获通道 1 快速使能位。	0
[1:0]	CC1S	RW	比较捕获通道 1 输入选择域。	0

捕获模式（引脚方向为输入）：

位	名称	访问	描述	复位值
[15:12]	IC2F	RW	输入捕获滤波器 2 配置域，这几位设置了 TI1 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变。 0000：无滤波器，以 fDTS 采样； 1000：采样频率 Fsampling=Fdts/8，N=6； 0001：采样频率 Fsampling=Fck_int，N=2； 1001：采样频率 Fsampling=Fdts/8，N=8； 0010：采样频率 Fsampling=Fck_int，N=4； 1010：采样频率 Fsampling=Fdts/16，N=5； 0011：采样频率 Fsampling=f=Fck_int，N=8； 1011：采样频率 Fsampling=Fdts/16，N=6； 0100：采样频率 Fsampling=Fdts/2，N=6； 1100：采样频率 Fsampling=Fdts/16，N=8； 0101：采样频率 Fsampling=Fdts/2，N=8； 1101：采样频率 Fsampling=Fdts/32，N=5； 0110：采样频率 Fsampling=Fdts/4，N=6； 1110：采样频率 Fsampling=Fdts/32，N=6； 0111：采样频率 Fsampling=Fdts/4，N=8； 1111：采样频率 Fsampling=Fdts/32，N=8。	0
[11:10]	IC2PSC	RW	比较捕获通道 2 预分频配置域，这 2 位定义了比较捕获通道 2 的预分频系数。一旦 CC1E=0，则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获； 01：每 2 个事件触发一次捕获； 10：每 4 个事件触发一次捕获； 11：每 8 个事件触发一次捕获。	0
[9:8]	CC2S	RW	比较捕获通道 2 输入选择域，这 2 位定义通道的方向(输入/输出)，及输入脚的选择。 00：比较捕获通道 1 通道被配置为输出； 01：比较捕获通道 1 通道被配置为输入，IC1 映射在 TI1 上； 10：比较捕获通道 1 通道被配置为输入，IC1 映射在 TI2 上； 11：比较捕获通道 1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在内部触发器输入被选中	0

			时(由 TS 位选择)。 *注：CC1S 仅在通道关闭时(CC1E 为 0)才是可写的。*	
[7:4]	IC1F	RW	输入捕获滤波器 1 配置域。	0
[3:2]	IC1PSC	RW	比较捕获通道 1 预分频配置域。	0
[1:0]	CC1S	RW	比较捕获通道 1 输入选择域。	0

15.4.8 比较/捕获控制寄存器 2（TIMx_CHCTLR2）（x=2/3/4）

偏移地址：0x1C

通道可用于输入(捕获模式)或输出(比较模式)，通道的方向由相应的 CCxS 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			

比较模式（引脚方向为输出）：

位	名称	访问	描述	复位值
15	OC4CE	RW	比较捕获通道 4 清零使能位。	0
[14:12]	OC4M	RW	比较捕获通道 4 模式设置域。	0
11	OC4PE	RW	比较捕获寄存器 4 预装载使能位。	0
10	OC4FE	RW	比较捕获通道 4 快速使能位。	0
[9:8]	CC4S	RW	比较捕获通道 4 输入选择域。	0
7	OC3CE	RW	比较捕获通道 3 清零使能位。	0
[6:4]	OC3M	RW	比较捕获通道 3 模式设置域。	0
3	OC3PE	RW	比较捕获寄存器 3 预装载使能位。	0
2	OC3FE	RW	比较捕获通道 3 快速使能位。	0
[1:0]	CC3S	RW	比较捕获通道 3 输入选择域。	0

捕获模式（引脚方向为输入）：

位	名称	访问	描述	复位值
[15:12]	IC4F	RW	输入捕获滤波器 4 配置域。	0
[11:10]	IC4PSC	RW	比较捕获通道 4 预分频配置域。	0
[9:8]	CC4S	RW	比较捕获通道 4 输入选择域。	0
[7:4]	IC3F	RW	输入捕获滤波器 3 配置域。	0
[3:2]	IC3PSC	RW	比较捕获通道 3 预分频配置域。	0
[1:0]	CC3S	RW	比较捕获通道 3 输入选择域。	0

15.4.9 比较/捕获使能寄存器（TIMx_CCER）（x=2/3/4）

偏移地址：0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	Reserved		CC3P	CC3E	Reserved		CC2P	CC2E	Reserved		CC1P	CC1E

位	名称	访问	描述	复位值
[15:14]	Reserved	RO	保留。	0
13	CC4P	RW	比较捕获通道 4 输出极性设置位。	0

12	CC4E	RW	比较捕获通道 4 输出使能位。	0
[11:10]	保留	RO	保留。	0
9	CC3P	RW	比较捕获通道 3 输出极性设置位。	0
8	CC3E	RW	比较捕获通道 3 输出使能位。	0
[7:6]	CC2NP	RO	保留。	0
5	CC2P	RW	比较捕获通道 2 输出极性设置位。	0
4	CC2E	RW	比较捕获通道 2 输出使能位。	0
[3:2]	保留	RO	保留。	0
1	CC1P	RW	比较捕获通道 1 输出极性设置位。	0
0	CC1E	RW	比较捕获通道 1 输出使能位。	0

15.4.10 通用定时器的计数器（TIMx_CNT）（x=2/3/4）

偏移地址：0x24

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

位	名称	访问	描述	复位值
[15:0]	CNT	RW	定时器的计数器的实时值。	0

15.4.11 计数时钟预分频器（TIMx_PSC）（x=2/3/4）

偏移地址：0x28

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

位	名称	访问	描述	复位值
[15:0]	PSC	RW	定时器的预分频器的分频系数；计数器的时钟频率等于分频器的输入频率/(PSC+1)。	0

15.4.12 自动重装值寄存器（TIMx_ATRLR）（x=2/3/4）

偏移地址：0x2C

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

位	名称	访问	描述	复位值
[15:0]	ATRLR	RW	ATRLR[15:0]的值将会被装入计数器，ATRLR 何时动作和更新请阅读 14.2.4 节；ATRLR 为空时，计数器停止。	0

15.4.13 比较/捕获寄存器 1（TIMx_CH1CVR）（x=2/3/4）

偏移地址：0x34

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1CVR[15:0]															

位	名称	访问	描述	复位值
[15:0]	CH1CVR	RW	比较捕获寄存器通道 1 的值。	0

15.4.14 比较/捕获寄存器 2（TIMx_CH2CVR）（x=2/3/4）

偏移地址：0x38

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2CVR[15:0]															

位	名称	访问	描述	复位值
[15:0]	CH2CVR	RW	比较捕获寄存器通道 2 的值。	0

15.4.15 比较/捕获寄存器 3（TIMx_CH3CVR）（x=2/3/4）

偏移地址：0x3C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3CVR[15:0]															

位	名称	访问	描述	复位值
[15:0]	CH3CVR	RW	比较捕获寄存器通道 3 的值。	0

15.4.16 比较/捕获寄存器 4（TIMx_CH4CVR）（x=2/3/4）

偏移地址：0x40

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4CVR[15:0]															

位	名称	访问	描述	复位值
[15:0]	CH4CVR	RW	比较捕获寄存器通道 4 的值。	0

15.4.17 DMA 控制寄存器（TIMx_DMACFGR）（x=2/3/4）

偏移地址：0x48

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

位	名称	访问	描述	复位值
[15:13]	Reserved	RO	保留。	0
[12:8]	DBL	RW	DMA 连续传送的长度，实际值为此域的值+1。	0
[7:5]	Reserved	RO	保留。	0
[4:0]	DBA	RW	这些位定义了 DMA 在连续模式下从控制寄存器 1 所在地址的偏移量。	0

15.4.18 连续模式的 DMA 地址寄存器（TIMx_DMAADR）（x=2/3/4）

偏移地址：0x4C

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

位	名称	访问	描述	复位值
[15:0]	DMAADR	RW	连续模式下，DMA 的地址。	0

第 16 章 基本定时器（BCTM）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

基本定时器模块包含一个 16 位可自动重装的定时器（TIM6 和 TIM7），用于计数和在更新新事件产生中断或 DMA 请求。

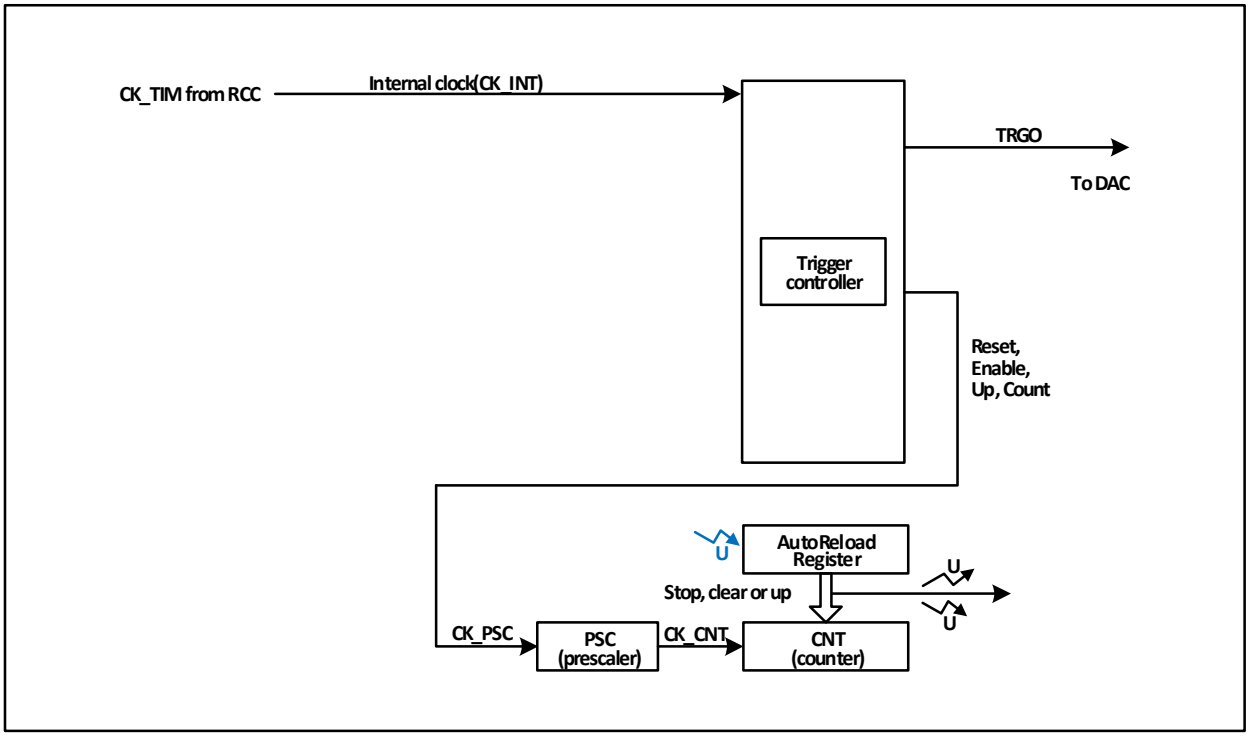
16.1 主要特征

通用定时器的主要特征包括：

- 16 位自动重装计数器，支持增计数模式
- 16 位预分频器，分频系数从 1～65536 之间动态可调
- 触发 DAC 同步电路
- 在更新事件时产生中断或 DMA 请求

16.2 原理和结构

图 16-1 基本定时器的结构框图



16.2.1 概述

如图 15-1 所示，通用定时器的结构大致可以分为两部分，即输入时钟部分和核心计数器部分。

基本定时器的时钟来自于 AHB 总线时钟（CK_INT）。这些输入的时钟信号经过各种设定的滤波分频等操作后成为 CK_PSC 时钟，输出给核心计数器部分。另外，这些复杂的时钟来源还可以作为 TRGO 输出至 DAC 外设。

基本定时器的核心是一个 16 位计数器（CNT）。CK_PSC 经过预分频器（PSC）分频后，成为 CK_CNT 再最终输给 CNT，CNT 支持增计数模式，并有一个自动重装值寄存器（ATRLR）在每个计数周期结束后为 CNT 重装载初始化值。

16.2.2 基本定时器和通用定时器的区别

与高级定时器相比，通用定时器缺少以下功能：

1） 基本定时器缺少减计数模式和增减计数模式。
2） 基本定时器缺少四路独立的比较捕获通道。
3） 基本定时器不支持外部信号控制定时器。
4） 基本定时器不支持增量式编码，定时器之间的级联和同步。

16.2.3 时钟输入

基本定时器的时钟由内部时钟 CK_INT 提供。

16.2.4 计数器和周边

CK_PSC 输入给预分频器（PSC）进行分频。PSC 是 16 位的，实际的分频系数相当于 R16_TIMx_PSC 的值+1。CK_PSC 经过 PSC 会成为 CK_INT。更改 R16_TIM1_PSC 的值并不会实时生效，而会在更新事件后更新给 PSC。更新事件包括 UG 位清零和复位。

16.3 调试模式

当系统进入调试模式时，根据 DBG 模块的设置可以控制定时器继续运转或者停止。

16.4 寄存器描述

表 16-1 TIM6 相关寄存器列表

名称	偏移地址	描述	复位值
R16_TIM6_CTLR1	0x40001000	TIM6 控制寄存器 1	0x0000
R16_TIM6_CTLR2	0x40001004	TIM6 控制寄存器 2	0x0000
R16_TIM6_DMAINTENR	0x4000100C	TIM6 DMA/中断使能寄存器	0x0000
R16_TIM6_INTFR	0x40001010	TIM6 中断状态寄存器	0x0000
R16_TIM6_SWEVGR	0x40001014	TIM6 事件产生寄存器	0x0000
R16_TIM6_CNT	0x40001024	TIM6 计数器	0x0000
R16_TIM6_PSC	0x40001028	TIM6 计数时钟预分频器	0x0000
R16_TIM6_ATRLR	0x4000102C	TIM6 自动重装值寄存器	0x0000

表 16-2 TIM7 相关寄存器列表

名称	偏移地址	描述	复位值
R16_TIM7_CTLR1	0x40001400	TIM7 控制寄存器 1	0x0000
R16_TIM7_CTLR2	0x40001404	TIM7 控制寄存器 2	0x0000
R16_TIM7_DMAINTENR	0x4000140C	TIM7 DMA/中断使能寄存器	0x0000
R16_TIM7_INTFR	0x40001410	TIM7 中断状态寄存器	0x0000
R16_TIM7_CNT	0x40001424	TIM7 计数器	0x0000
R16_TIM7_PSC	0x40001428	TIM7 计数时钟预分频器	0x0000
R16_TIM7_ATRLR	0x4000142C	TIM7 自动重装值寄存器	0x0000

16.4.1 控制寄存器 1（TIMx_CTLR1）（x=6/7）

偏移地址：0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ARPE	Reserved			OPM	URS	UDIS	CEN

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
7	ARPE	RW	自动重装预装使能位： 1：使能自动重装值寄存器（ATRLR）； 0：禁止自动重装值寄存器（ATRLR）。	0
[6:4]	Reserved	RO	保留。	0
3	OPM	RW	单脉冲模式。 1：在发生下一次更新事件(清除 CEN 位)时，计数器停止； 0：在发生下一次更新事件时，计数器不停止。	0
2	URS	RW	更新请求源，软件通过该位选择 UEV 事件的源。 1：如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求； 0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： -计数器溢出/下溢 -设置 UG 位 -从模式控制器产生的更新	0
1	UDIS	RW	禁止更新，软件通过该位允许/禁止 UEV 事件的产生。 1：禁止 UEV。不产生更新事件，各寄存器(ATRLR、PSC、CHCTLRx)保持它们的值。如果设置了 UG 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。 0：允许 UEV。更新(UEV)事件由下述任一事件产生： -计数器溢出/下溢 -设置 UG 位 -从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。	0
0	CEN	RW	使能计数器 (Counter enable)。 1：使能计数器； 0：禁止计数器。	0

16.4.2 控制寄存器 2（TIMx_CTLR2）（x=6/7）

偏移地址：0x04

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

位	名称	访问	描述	复位值
[15:7]	Reserved	RO	保留。	0
[6:4]	MMS	RW	主模式选择：这 3 位用于选择在主模式下送到从定时器的同步信息(TRGO)。可能的组合如下： 000：复位－UG 位被用于作为触发输出(TRGO)。如果是触发输入产生的复位(从模式控制器处于复位模式)，则 TRGO 上的信号相对实际的复位会有一个延迟； 001:使能－ 计数器使能信号 CNT_EN 被用于作为触发输出(TRGO)。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过 CEN 控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟，除非选择了主/从模式(见 TIMx_SMCFGR 寄存器中 MSM 位的描述)； 010：更新事件被选为触发输入(TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。	0
[3:0]	Reserved	RO	保留。	0

16.4.3 DMA/中断使能寄存器（TIMx_DMAINTENR）（x=6/7）

偏移地址：0x0C

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

位	名称	访问	描述	复位值
[15:9]	Reserved	RO	保留。	0
8	UDE	RW	更新的 DMA 请求使能位。 1：允许更新的 DMA 请求； 0：禁止更新的 DMA 请求。	0
[7:1]	Reserved	RO	保留。	0
0	UIE	RW	更新中断使能位。 1：允许更新中断； 0：禁止更新中断。	0

16.4.4 中断状态寄存器（R16_TIMx_INTFR）（x=6/7）

偏移地址：0x10

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

位	名称	访问	描述	复位值
[15:1]	Reserved	RO	保留。	0
0	UIF	RWO	更新中断标志位，当产生更新事件时该位由硬件置位，由软件清零。 1：更新中断产生； 0：无更新事件产生。 以下情形会产生更新事件： 若 UDIS=0，当重复计数器数值上溢或下溢时； 若 URS=0、UDIS=0，当置 UG 位时，或当通过软件对计数器核心计数器重新初始化时；	0

16.4.5 事件产生寄存器（TIMx_SWEVGR）（x=6/7）

偏移地址：0x14

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

位	名称	访问	描述	复位值
[15:1]	Reserved	RO	保留。	0
0	UG	WO	更新事件产生位，产生更新事件。该位由软件置位，由硬件自动清零。 1：初始化计数器，并产生一个更新事件； 0：无动作。 *注：预分频器的计数器也被清零，但是预分频系数不变。若在中心对称模式下或增计数模式下则核心计数器被清零；若减计数模式下则核心计数器取重装值寄存器的值。*	0

16.4.6 通用定时器的计数器（TIMx_CNT）（x=6/7）

偏移地址：0x24

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

位	名称	访问	描述	复位值
[15:0]	CNT	RW	定时器的计数器的实时值。	0

16.4.7 计数时钟预分频器（TIMx_PSC）（x=6/7）

偏移地址：0x28

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

位	名称	访问	描述	复位值
[15:0]	PSC	RW	定时器的预分频器的分频系数；计数器的时钟频率	0

			等于分频器的输入频率/(PSC+1)。	

16.4.8 自动重装值寄存器（TIMx_ATRLR）（x=6/7）

偏移地址：0x2C

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

位	名称	访问	描述	复位值
[15:0]	ATRLR	RW	ATRLR[15:0]的值将会被装入计数器，ATRLR 何时动作和更新请阅读 14.2.4 节；ATRLR 为空时，计数器停止。	0

第 17 章 数字/模拟转换（DAC）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器部分产品。

数字/模拟转换模块（DAC），包含 2 个可配置 8/12 位数字输入转换 2 路模拟电压输出的转换器。内置三角波、噪声波形发生器，支持多种事件触发转换，DMA 功能等。

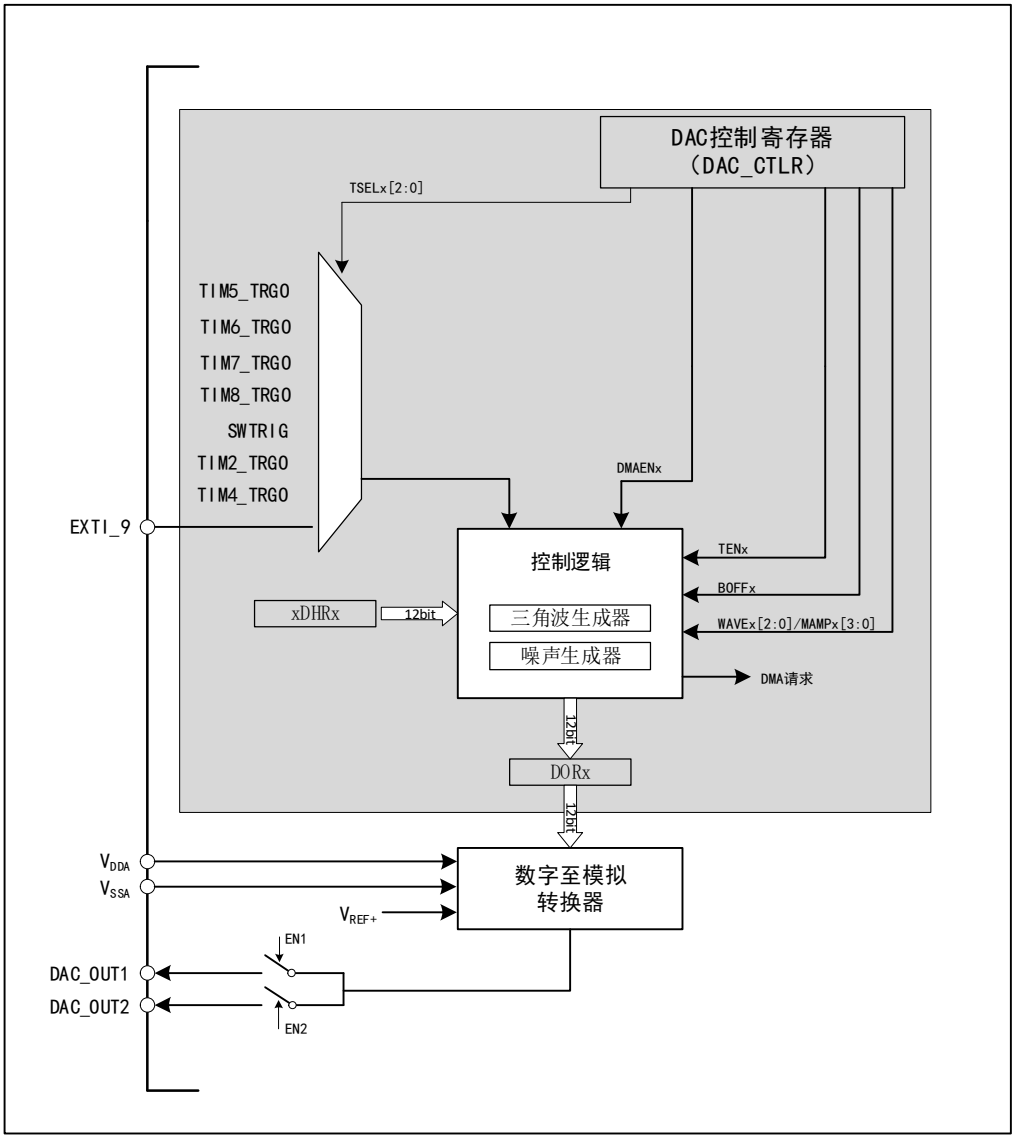
17.1 主要特性

- 2 个 DAC 转换器，每个转换器对应 1 个输出通道
- 三角波、噪声波形发生器
- 可配置 8 位或 12 位输出
- 12 位数据左对齐或右对齐
- 双 DAC 同时或分别转换
- 支持 DMA 功能
- 多种触发事件

17.2 功能描述

17.2.1 DAC 模块结构

图 17-1 DAC 模块框图



17.2.2 DAC 通道配置

17.2.2.1 开启 DAC 功能：

将 DAC_CTLR 寄存器的 ENx 位置 1，即可打开对 DAC 通道 x 的模拟供电。经过一段启动时间，DAC 通道 x 即被使能。DAC 包含 2 个模拟输出通道，可同时或分别独立输出。

注：为了避免寄生的干扰和额外的功耗，DAC 通道对应的引脚需提前设置成模拟输入(AIN)模式。

17.2.2.2 打开输出缓冲：

DAC 集成了输出缓冲，可以用来减少输出阻抗，增加驱动能力直接驱动外部负载。每个 DAC 通道输出缓存可以通过设置 DAC_CTLR 寄存器的 BOFFx 位来使能或关闭。

17.2.2.3 数据格式：

单 DAC 通道模式下，包括 8 位数据右对齐、12 位数据左对齐、12 位数据右对齐。写入数据到

DAC_R8BDHRx[7:0]，模块将加载（1 个 APB1 时钟周期后）其左移数据到数据输出寄存器 DAC_DORx[11:0]，写入数据到 DAC_R12BDHRx[11:0]，模块将加载（1 个 APB1 时钟周期后）右对齐数据到数据输出寄存器 DAC_DORx[11:0]；写入数据到 DAC_L12BDHRx[15:4]，模块经过相应的移位后，将加载（1 个 APB1 时钟周期后）左对齐数据到数据输出寄存器 DAC_DORx[11:0]。双 DAC 通道模式下，同样包括 8 位数据右对齐、12 数据左对齐、12 位数据右对齐三种方式。写入数据到 DAC_RD8BDHR[7:0]，模块将加载（1 个 APB1 时钟周期后）位[7:0]移位后到 DAC_DOR1[11:0]，位[15:8]移位后到 DAC_DOR2[11:0]；写入数据到 DAC_LD12BDHR[31:0]，模块将加载（1 个 APB 时钟周期后）位[15:4]数据移位后到 DAC_DOR1[11:4]，位[31:20]数据移位后到 DAC_DOR2[11:4]；写入数据到 DAC_RD12BDHR[31:0]，模块将加载（1 个 APB 时钟周期后）位[11:0]数据到 DAC_DOR1[11:0]，位[27:16]数据到 DAC_DOR2[11:0]。

图 17-2 单通道数据格式

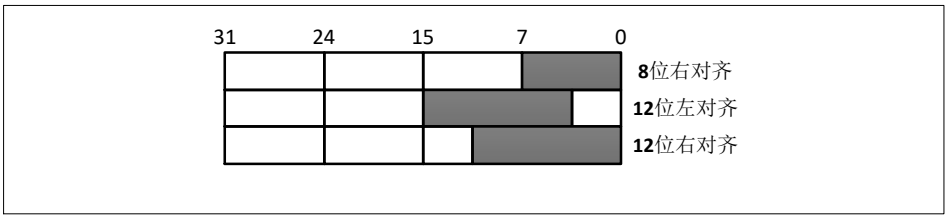
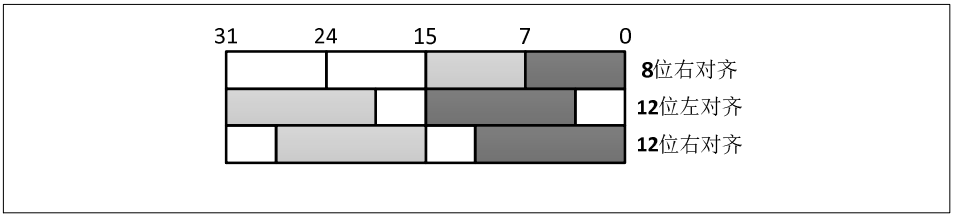


图 17-3 双通道数据格式



17.2.2.4 DMA 功能：

DAC 通道具有 DMA 功能。设置 DAC_CTLR 寄存器的 DMAENx 位为 1，开启对应通道的 DMA 功能。当有触发事件(不包括软件触发)发生，则产生一个 DMA 请求，然后 DAC_DORx 寄存器的数据将被更新。

17.2.2.5 触发事件选择：

DAC 转换可以由以下事件触发进行转换。当配置 DAC_CTLR 寄存器的 TENx 位为 1，配置 TSELx[2:0]控制位选择某个触发事件触发 DAC 转换。

表 17-1 触发事件

触发源	类型	TSELx[2:0]
定时器 6 TRGO 事件	来自片上定时器内部信号	000
定时器 8 TRGO 事件		001
定时器 7 TRGO 事件		010
定时器 5 TRGO 事件		011
定时器 2 TRGO 事件		100
定时器 4 TRGO 事件		101
EXTI 线路 9	外部引脚	110
SWTRIG（软件触发）	软件控制位	111

DAC 接口会监测到来自选中的定时器 TRGO 输出或外部中断线 9 的上升沿，在触发后的 3 个 APB1 时钟周期后，将寄存器 DAC_DORx 更新为新值。

如果配置的是软件触发方式，SWTRIG 位一旦置 1，将会启动一次转换，在触发后的 1 个 APB1 时

钟周期后，将寄存器 DAC_DORx 更新为新值，并且硬件对 SWTRIG 位自动清 0。

注：不能在 ENx 为 1 时改变 TSELx[2:0]位。

17.2.3 DAC 转换

DAC 通道的数据来自 DAC_DORx 寄存器，但不能直接对寄存器 DAC_DORx 写入数据，任何输出到 DAC 通道 x 的数据都必须写入 DAC_R12BDHR1、DAC_L12BDHR1、DAC_R12BDHR2、DAC_L12BDHR2、DAC_RD12BDHR、DAC_LD12BDHR、DAC_RD8BDHR 寄存器中。由系统内部的保持寄存器 DAC_DHRx 会获取上述寄存器值将其经过相应时间送入 DAC_DORx 寄存器。

非触发方式下，写入寄存器 DAC_xDHRx 的数据会在 1 个 APB1 时钟周期后移入 DAC_DORx 寄存器。

软件触发下，事件触发上升沿后 1 个 APB1 时钟周期后自动更新 DAC_DORx 寄存器。

硬件触发（定时器 TRGO 事件或者外部中断线 9 上升沿）下，触发事件后 3 个 APB1 时钟周期后自动更新 DAC_DORx 寄存器。

装入 DAC_DORx 寄存器数据，在经过时间 $t_{SETTLING}$ 之后，输出即有效，这段时间的长短依电源电压和模拟输出负载的不同会有所变化。

数字输入经过 DAC 被线性地转换为模拟电压输出，其范围为 0 到 V_{DDA}。任一 DAC 通道引脚上的输出电压满足下面的关系：

DAC 输出电压 = V_{DDA}* (DAC_DORx/ 4096)。

17.2.4 DAC 三角波生成器

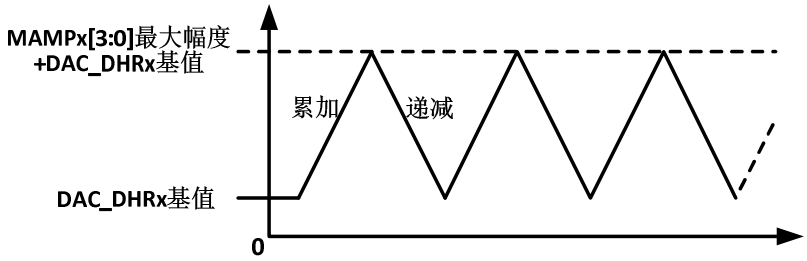
模块内置了一个三角波生成器，可以在基准信号上加上一个小幅度的三角波。设置 WAVEx[1:0]位为‘10’选择 DAC 的三角波生成功能。设置 DAC_CTLR 寄存器的 MAMPx[3:0]位来选择三角波的幅度。

系统内部包含一个从 0 开始的三角波计数器，在每次触发事件后 3 个 APB1 时钟周期后累加 1。计数器的值与 DAC_DHRx 寄存器的数值相加并丢弃溢出位后写入 DAC_DORx 寄存器。在传入 DAC_DORx 寄存器的数值小于 MAMPx[3:0]位定义的最大幅度时，三角波计数器逐步累加，一旦达到设置的最大幅度，则计数器开始递减，达到 0 后再开始累加，周而复始。将 WAVEx[1:0]位置‘00’，可以复位三角波的生成。

注：1. 为了产生三角波，必须使能 DAC 触发，即设 DAC_CTLR 寄存器的 TENx 位为 1。

2. MAMPx[3:0]位必须在使能 DAC 之前设置，否则其值不能修改。

图 17-4 三角波生成



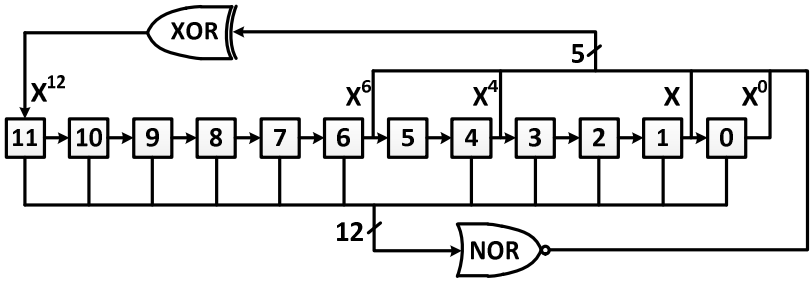
17.2.5 DAC 噪声生成器

模块内置了一个噪声生成器，是利用线性反馈移位寄存器(Linear Feedback Shift Register LFSR)产生幅度变化的伪噪声。设置 WAVE[1:0]位为‘01’选择 DAC 噪声生成功能。设置 DAC_CTLR 寄存器的 MAMPx[3:0]位来选择屏蔽部分 LFSR 的数据。

寄存器 LFSR 的预装入值为 0xAAA。按照特定算法，在每次触发事件后 3 个 APB1 时钟周期之后更新该寄存器的值。设置 DAC_CR 寄存器的 MAMPx[3:0]位可以屏蔽部分或者全部 LFSR 的数据，这样的得到的 LSFR 值与 DAC_DHRx 的数值相加，去掉溢出位之后即被写入 DAC_DORx 寄存器。如果寄存器 LFSR 值为 0x000，则会注入‘1’(防锁定机制)。将 WAVEx[1:0]位置‘00’，可以复位 LFSR 波形的生成算法。

注：为了产生噪声，必须使能 DAC 触发，即设 DAC_CTLR 寄存器的 TENx 位为 1。

图 17-5 LFSR 寄存器算法



17.3 双 DAC 转换

当需要 2 个 DAC 同时转换的情况下，为了更加便捷和高效的操作 DAC 模块，模块集成 3 个双 DAC 模式下的数据寄存器 DAC_RD8BDHR、DAC_LD12BDHR、DAC_RD12BDHR。只需操作其中之一寄存器即可更新 2 个 DAC 的转换值。

针对双 DAC 的转换可配合模块的其他寄存器，可实现 11 种不同组合的转换模式，两个通道待转换值需写入上述 3 个双通道数据寄存器之一。

17.3.1 不同触发下使用相同 LFSR

设置 TENx 置位，TSELx 为不同值，WAVEx 为 0b01，MAMPx 为相同的 LFSR 屏蔽值。当通道 1 触发事件发生时，将通道 1 数据寄存器 DAC_DHR1 值加上带相同屏蔽的 LFSR1 计数值，延迟 3 个 APB1 时钟后送给 DAC_DOR1，用于转换，并更新 LFSR1；当通道 2 触发事件发生时，将通道 2 数据寄存器 DAC_DHR2 值加上带相同屏蔽的 LFSR2 计数值，延迟 3 个 APB1 时钟后送给 DAC_DOR2，用于转换，并更新 LFSR2。

17.3.2 不同触发下使用不同 LFSR

设置 TENx 置位，TSELx 为不同值，WAVEx 为 0b01，MAMPx 为不同的 LFSR 屏蔽值。当通道 1 触发事件发生时，将通道 1 数据寄存器 DAC_DHR1 值加上按 MAMP1[3:0] 所设屏蔽的 LFSR1 计数值，延迟 3 个 APB1 时钟后送给 DAC_DOR1，用于转换，并更新 LFSR1；当通道 2 触发事件发生时，将通道 2 数据寄存器 DAC_DHR2 值加上按 MAMP2[3:0]所设屏蔽的 LFSR2 计数值，延迟 3 个 APB1 时钟后送给 DAC_DOR2，用于转换，并更新 LFSR2。

17.3.3 不同触发下产生相同三角波

设置 TENx 置位，TSELx 为不同值，WAVEx 为 0b1x，MAMPx 设为相同三角波幅值。当通道 1 触发事件发生时，将通道 1 数据寄存器 DAC_DHR1 值加上 MAMPx 所设的相同幅值的三角波计数器值，延迟 3 个 APB1 时钟后送给 DAC_DOR1，用于转换，并更新通道 1 三角波计数器；当通道 2 触发事件发生时，将通道 2 数据寄存器 DAC_DHR2 值加上 MAMPx 所设的相同幅值的三角波计数器值，延迟 3 个 APB1 时钟后送给 DAC_DOR2，用于转换，并更新通道 2 三角波计数器。

17.3.4 不同触发下产生不同三角波

设置 TENx 置位，TSELx 为不同值，WAVEx 为 0b1x，MAMPx 设为不同的三角波幅值。当通道 1 触发事件发生时，将通道 1 数据寄存器 DAC_DHR1 值加上 MAMP1 所设幅值的三角波计数器值，延迟 3 个 APB1 时钟后送给 DAC_DOR1，用于转换，并更新通道 1 三角波计数器；当通道 2 触发事件发生时，将通道 2 数据寄存器 DAC_DHR2 值加上 MAMP2 所设幅值的三角波计数器值，延迟 3 个 APB1 时钟后送给 DAC_DOR2，用于转换，并更新通道 2 三角波计数器。

17.3.5 不同触发下不使用波形发生器

设置 TENx 置位，TSELx 为不同值选择不同触发源。当通道 1 触发事件发生时，将通道 1 数据寄存器 DAC_DHR1 值延迟 3 个 APB1 时钟后送给 DAC_DOR1，用于转换；当通道 2 触发事件发生时，将通道 2 数据寄存器 DAC_DHR2 值延迟 3 个 APB1 时钟后送给 DAC_DOR2，用于转换。

17.3.6 均使用软件触发

在此配置下，双通道数据寄存器写入需要转换值，1 个 APB1 时钟周期后，DAC_DHR1 和 DAC_DHR2 的数据被分别送到 DAC_DOR1 和 DAC_DOR2 用于转换。

17.3.7 相同触发下使用相同 LFSR

设置 TENx 置位，TSELx 为相同值，WAVEx 为 0b01，MAMPx 为相同的 LFSR 屏蔽值。当触发事件发生后，寄存器 DAC_DHR1 的值加上带相同屏蔽的 LFSR1 计数值，延迟 3 个 APB1 时钟后送给 DAC_DOR1，用于转换，并更新 LFSR1，同时寄存器 DAC_DHR2 的值加上带相同屏蔽的 LFSR2 计数值，延迟 3 个 APB1 时钟后送给 DAC_DOR2，用于转换，并更新 LFSR2。

17.3.8 相同触发下使用不同 LFSR

设置 TENx 置位，TSELx 为相同值，WAVEx 为 0b01，MAMPx 为不同的 LFSR 屏蔽值。当触发事件发生后，寄存器 DAC_DHR1 的值加上带不同屏蔽值的 LFSR1 计数值，延迟 3 个 APB1 时钟后送给 DAC_DOR1，用于转换，并更新 LFSR1，同时寄存器 DAC_DHR2 的值加上带不同屏蔽值的 LFSR2 计数值，延迟 3 个 APB1 时钟后送给 DAC_DOR2，用于转换，并更新 LFSR2。

17.3.9 相同触发下产生相同三角波

设置 TENx 置位，TSELx 为相同值，WAVEx 为 0b1x，MAMPx 为相同的三角波幅值。当触发事件发生后，寄存器 DAC_DHR1 的值加上相同三角波幅值的计数器值，延迟 3 个 APB1 时钟后送给 DAC_DOR1，用于转换，并更新通道 1 三角波计数器，同时寄存器 DAC_DHR2 的值加上相同三角波幅值的计数器值，延迟 3 个 APB1 时钟后送给 DAC_DOR2，用于转换，并更新通道 2 三角波计数器值。

17.3.10 相同触发下产生不同三角波

设置 TENx 置位，TSELx 为相同值，WAVEx 为 0b1x，MAMPx 为不同的三角波幅值。当触发事件发生后，寄存器 DAC_DHR1 的值加上 MAMP1[3:0]所设的三角波幅值计数器值，延迟 3 个 APB1 时钟后送给 DAC_DOR1，用于转换，并更新通道 1 三角波计数器，同时寄存器 DAC_DHR2 的值加上 MAMP2[3:0]所设的三角波幅值计数器值，延迟 3 个 APB1 时钟后送给 DAC_DOR2，用于转换，并更新通道 2 三角波计数器值。

17.3.11 相同触发下不使用波形发生器

设置 TENx 置位，TSELx 为相同值。该配置下，当触发事件发生后，寄存器 DAC_DHR1 和 DAC_DHR2 的值在延迟 3 个 APB1 时钟后分别送给 DAC_DOR1 和 DAC_DOR2 用于 DAC 转换。

17.4 寄存器描述

表 17-2 DAC 相关寄存器列表

名称	访问地址	描述	复位值
R32_DAC_CTLR	0x40007400	DAC 配置寄存器	0x00000000

R32_DAC_SWTR	0x40007404	DAC 软件触发寄存器	0x00000000
R32_DAC_R12BDHR1	0x40007408	DAC 通道 1 右对齐 12 位数据保存寄存器	0x00000000
R32_DAC_L12BDHR1	0x4000740C	DAC 通道 1 左对齐 12 位数据保存寄存器	0x00000000
R32_DAC_R8BDHR1	0x40007410	DAC 通道 1 右对齐 8 位数据保存寄存器	0x00000000
R32_DAC_R12BDHR2	0x40007414	DAC 通道 2 右对齐 12 位数据保存寄存器	0x00000000
R32_DAC_L12BDHR2	0x40007418	DAC 通道 2 左对齐 12 位数据保存寄存器	0x00000000
R32_DAC_R8BDHR2	0x4000741C	DAC 通道 2 右对齐 8 位数据保存寄存器	0x00000000
R32_DAC_RD12BDHR	0x40007420	双通道右对齐 12 位数据保存寄存器	0x00000000
R32_DAC_LD12BDHR	0x40007424	双通道左对齐 12 位数据保存寄存器	0x00000000
R32_DAC_RD8BDHR	0x40007428	双通道右对齐 8 位数据保存寄存器	0x00000000
R32_DAC_DOR1	0x4000742C	DAC 通道 1 数据输出寄存器	0x00000000
R32_DAC_DOR2	0x40007430	DAC 通道 2 数据输出寄存器	0x00000000

17.4.1 DAC 配置寄存器（DAC_CTLR）

偏移地址：0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			DMAEN2	MAMP2[3:0]				WAVE2[2:0]		TSEL2[2:0]			TEN2	BOFF2	EN2
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DMAEN1	MAMP1[3:0]				WAVE1[2:0]		TSEL1[2:0]			TEN1	BOFF1	EN1

位	名称	访问	描述	复位值
[31:29]	Reserved	RO	保留。	0
28	DMAEN2	RW	DAC 通道 2 的 DMA 使能： 1：使能 DAC 通道 2 DMA 功能； 0：关闭 DAC 通道 2 DMA 功能。	0
[27:24]	MAMP2	RW	DAC 通道 2 屏蔽/幅值设置，软件设置这个区域用来在噪声生成模式下选择 LFSR 数据屏蔽位，在三角波形生成模式下选择波形的幅值： 0000：不屏蔽 LFSR 位 0/三角波幅值为 1； 0001：不屏蔽 LFSR 位[1:0]/三角波幅值为 3； 0010：不屏蔽 LFSR 位[2:0]/三角波幅值为 7； 0011：不屏蔽 LFSR 位[3:0]/三角波幅值为 15； 0100：不屏蔽 LFSR 位[4:0]/三角波幅值为 31； 0101：不屏蔽 LFSR 位[5:0]/三角波幅值为 63； 0110：不屏蔽 LFSR 位[6:0]/三角波幅值为 127； 0111：不屏蔽 LFSR 位[7:0]/三角波幅值为 255； 1000：不屏蔽 LFSR 位[8:0]/三角波幅值为 511； 1001：不屏蔽 LFSR 位[9:0]/三角波幅值为 1023； 1010：不屏蔽 LFSR 位[10:0]/三角波幅值为 2047； ≥1011：不屏蔽 LFSR 位[11:0]/三角波幅值为 4095。	0
[23:22]	WAVE2	RW	DAC 通道 2 的噪声/三角波生成使能 00：关闭波形发生器； 01：使能噪声波形发生器； 1x：使能三角波波形发生器。	0

[21:19]	TSEL2	RW	DAC 通道 2 触发事件选择设置： 000：TIM6 TRGO 事件； 001：TIM8 TRGO 事件； 010：TIM7 TRGO 事件 011：TIM5 TRGO 事件 100：TIM2 TRGO 事件； 101：TIM4 TRGO 事件； 110：外部中断线 9； 111：软件触发； 其他：保留。	0
18	TEN2	RW	DAC 通道 2 外部触发模式使能： 1：使能 DAC 通道 2 触发功能，写入 DAC_xDHR 寄存器的数据在 3 个 APB1 时钟周期后送入 DAC_DOR2 寄存器。 0：关闭 DAC 通道 2 触发功能，写入 DAC_xDHR 寄存器的数据在 1 个 APB1 时钟周期后送入 DAC_DOR2 寄存器。 *注：如果选择软件触发，DAC_xDHR 中的数据只需 1 个 APB1 时钟周期后送入 DAC_DOR2 寄存器。*	0
17	BOFF2	RW	DAC 通道 2 输出缓冲关闭控制（建议打开）： 1：关闭 DAC 通道 2 输出缓存； 0：打开 DAC 通道 2 输出缓存。	0
16	EN2	RW	DAC 通道 2 使能： 1：使能 DAC 通道 2； 0：关闭 DAC 通道 2。	0
[15:13]	Reserved	RO	保留。	0
12	DMAEN1	RW	DAC 通道 1 的 DMA 使能： 1：使能 DAC 通道 1 DMA 功能； 0：关闭 DAC 通道 1 DMA 功能。	0
[11:8]	MAMP1	RW	DAC 通道 1 屏蔽/幅值设置，软件设置这个区域用来在噪声生成模式下选择 LFSR 数据屏蔽位，在三角波形生成模式下选择波形的幅值： 0000：不屏蔽 LFSR 位 0/三角波幅值为 1； 0001：不屏蔽 LFSR 位[1:0]/三角波幅值为 3； 0010：不屏蔽 LFSR 位[2:0]/三角波幅值为 7； 0011：不屏蔽 LFSR 位[3:0]/三角波幅值为 15； 0100：不屏蔽 LFSR 位[4:0]/三角波幅值为 31； 0101：不屏蔽 LFSR 位[5:0]/三角波幅值为 63； 0110：不屏蔽 LFSR 位[6:0]/三角波幅值为 127； 0111：不屏蔽 LFSR 位[7:0]/三角波幅值为 255； 1000：不屏蔽 LFSR 位[8:0]/三角波幅值为 511； 1001：不屏蔽 LFSR 位[9:0]/三角波幅值为 1023； 1010：不屏蔽 LFSR 位[10:0]/三角波幅值为 2047； ≥1011：不屏蔽 LFSR 位[11:0]/三角波幅值为 4095。	0
[7:6]	WAVE1	RW	DAC 通道 1 的噪声/三角波生成使能。 00：关闭波形发生器； 01：使能噪声波形发生器； 1x：使能三角波波形发生器。	0

[5:3]	TSEL1	RW	DAC 通道 1 触发事件选择设置： 000：TIM6 TRGO 事件； 001：TIM8 TRGO 事件； 010：TIM7 TRGO 事件 011：TIM5 TRGO 事件 100：TIM2 TRGO 事件； 101：TIM4 TRGO 事件； 110：外部中断线 9； 111：软件触发； 其他：保留。	0
2	TEN1	RW	DAC 通道 1 外部触发模式使能： 1：使能 DAC 通道 1 触发功能，写入 DAC_xDHR 寄存器的数据在 3 个 APB1 时钟周期后送入 DAC_DOR1 寄存器。 0：关闭 DAC 通道 1 触发功能，写入 DAC_xDHR 寄存器的数据在 1 个 APB1 时钟周期后送入 DAC_DOR1 寄存器。 *注：如果选择软件触发，DAC_xDHR 中的数据只需 1 个 APB1 时钟周期后送入 DAC_DOR1 寄存器。*	0
1	BOFF1	RW	DAC 通道 1 输出缓冲关闭控制（建议打开）： 1：关闭 DAC 通道 1 输出缓存； 0：打开 DAC 通道 1 输出缓存。	0
0	EN1	RW	DAC 通道 1 使能： 1：使能 DAC 通道 1； 0：关闭 DAC 通道 1。	0

注：配置寄存器包括通道 1 和通道 2 的配置，当同时使能（ENx 位为‘1’）2 个通道的输出，将以通道 1 的配置输出相同波形到 2 个硬件通道上；当只使能 1 号通道的输出，将以 1 号通道配置输出波形到 1 号硬件通道上，2 号通道不输出；当只使能 2 号通道的输出，将以 2 号通道配置输出波形到 2 号硬件通道上，1 号通道不输出。

17.4.2 DAC 软件触发寄存器（DAC_SWTR）

偏移地址：0x04

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														SWTRIG2	SWTRIG1

位	名称	访问	描述	复位值
[31:2]	Reserved	RO	保留。	0
1	SWTRIG2	WO	DAC 通道 2 软件触发控制位： 1：使能 DAC 通道 2 软件触发； 0：关闭 DAC 通道 2 软件触发。 *注：一旦 DAC_xDHR 中的数据（1 个 APB1 时钟周期后）送入 DAC_DOR2 寄存器，该位将硬件清 0。*	0
0	SWTRIG1	WO	DAC 通道 1 软件触发控制位：	0

			1：使能 DAC 通道 1 软件触发； 0：关闭 DAC 通道 1 软件触发。 *注：一旦 DAC_xDHR 中的数据（1 个 APB1 时钟周期后）送入 DAC_DOR1 寄存器，该位将硬件清 0。*	

17.4.3 DAC 通道 1 右对齐 12 位数据保存寄存器（DAC_R12BDHR1）

偏移地址：0x08

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACC1DHR[11:0]											

位	名称	访问	描述	复位值
[31:12]	Reserved	RO	保留。	0
[11:0]	DACC1DHR	RW	DAC 通道 1 的 12 位右对齐数据。	0

17.4.4 DAC 通道 1 左对齐 12 位数据保存寄存器（DAC_L12BDHR1）

偏移地址：0x0C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Reserved			

位	名称	访问	描述	复位值
[31:12]	Reserved	RO	保留。	0
[11:4]	DACC1DHR	RW	DAC 通道 1 的 12 位左对齐数据。	0
[3:0]	Reserved	RO	保留。	0

17.4.5 DAC 通道 1 右对齐 8 位数据保存寄存器（DAC_R8BDHR1）

偏移地址：0x10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DACC1DHR[7:0]							

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:0]	DACC1DHR	RW	DAC 通道 1 的 8 位右对齐数据。	0

17.4.6 DAC 通道 2 右对齐 12 位数据保存寄存器（DAC_R12BDHR2）

偏移地址：0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACC2DHR[11:0]											

位	名称	访问	描述	复位值
[31:12]	Reserved	RO	保留。	0
[11:0]	DACC2DHR	RW	DAC 通道 2 的 12 位右对齐数据。	0

17.4.7 DAC 通道 2 左对齐 12 位数据保存寄存器（DAC_L12BDHR2）

偏移地址：0x18

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[11:0]												Reserved			

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:4]	DACC2DHR	RW	DAC 通道 2 的 12 位左对齐数据。	0
[3:0]	Reserved	RO	保留。	0

17.4.8 DAC 通道 2 右对齐 8 位数据保存寄存器（DAC_R8BDHR2）

偏移地址：0x1C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DACC2DHR[7:0]							

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:0]	DACC2DHR	RW	DAC 通道 2 的 8 位右对齐数据。	0

17.4.9 DAC 双通道右对齐 12 位数据保存寄存器（DAC_RD12BDHR）

偏移地址：0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				DACC2DHR[11:0]											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACC1DHR[11:0]											

位	名称	访问	描述	复位值
[31:28]	Reserved	RO	保留。	0
[27:16]	DACC2DHR	RW	DAC 通道 2 的 12 位右对齐数据。	0
[15:12]	Reserved	RO	保留。	0
[11:0]	DACC1DHR	RW	DAC 通道 1 的 12 位右对齐数据。	0

17.4.10 DAC 双通道左对齐 12 位数据保存寄存器（DAC_LD12BDHR）

偏移地址：0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACC2DHR[11:0]												Reserved			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Reserved			

位	名称	访问	描述	复位值
[31:20]	DACC2DHR	RW	DAC 通道 2 的 12 位左对齐数据。	0
[19:16]	Reserved	RO	保留。	0
[15:4]	DACC1DHR	RW	DAC 通道 1 的 12 位左对齐数据。	0
[3:0]	Reserved	RO	保留。	0

17.4.11 DAC 双通道右对齐 8 位数据保存寄存器（DAC_RD8BDHR）

偏移地址：0x28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[7:0]								DACC1DHR[7:0]							

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:8]	DACC2DHR	RW	DAC 通道 2 的 8 位右对齐数据。	0
[7:0]	DACC1DHR	RW	DAC 通道 1 的 8 位右对齐数据。	0

17.4.12 DAC 通道 1 数据输出寄存器（DAC_DOR1）

偏移地址：0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACC1DOR[11:0]											

位	名称	访问	描述	复位值
[31:12]	Reserved	RO	保留。	0
[11:0]	DACC1DOR	RO	DAC 通道 1 输出数据。	0

17.4.13 DAC 通道 2 数据输出寄存器（DAC_DOR2）

偏移地址：0x30

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACC2DOR[11:0]											

位	名称	访问	描述	复位值
[31:12]	Reserved	RO	保留。	0
[11:0]	DACC2DOR	RO	DAC 通道 2 输出数据。	0

第 18 章 通用同步异步收发器（USART）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

该模块包含 3 个通用同步异步收发器（USART1/2/3）和 5 个通用异步收发器（UART4/5/6/7/8）。

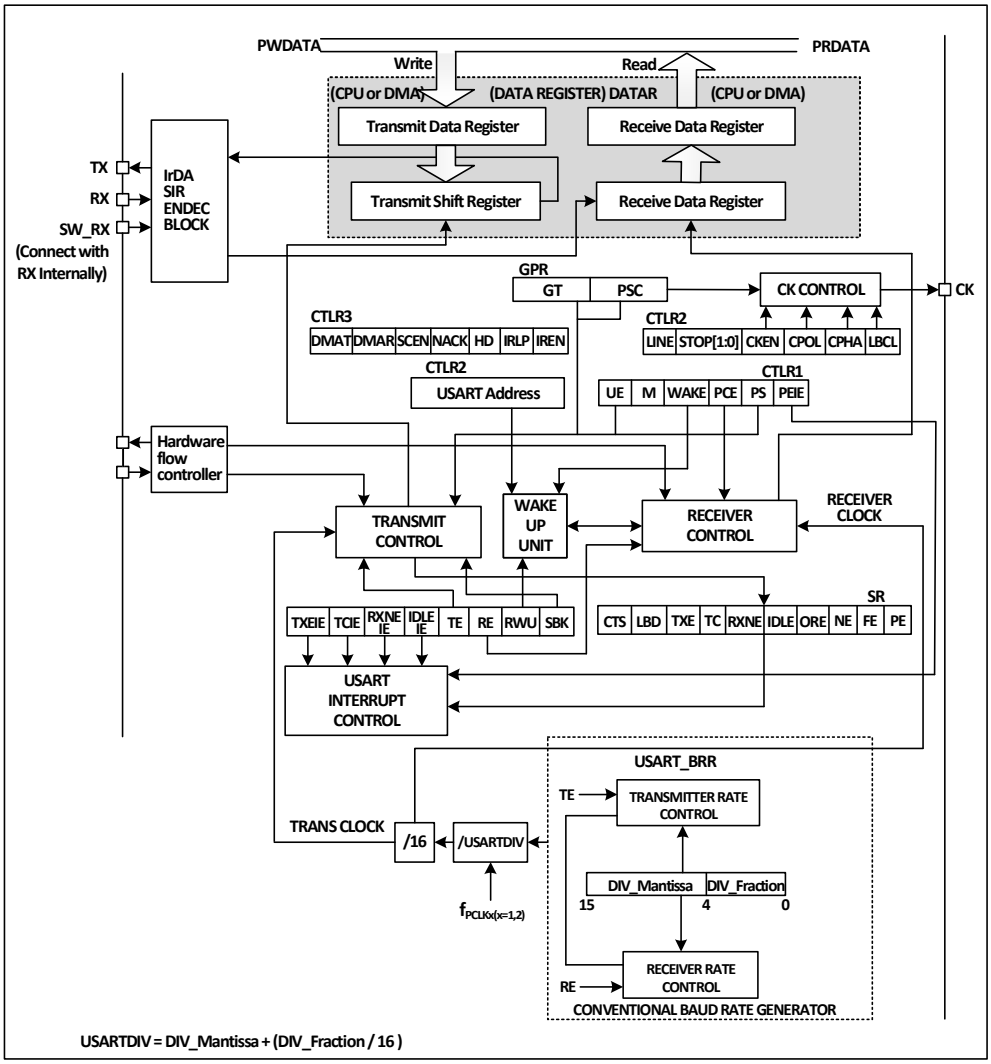
注：对于 CH32V20x_D6、CH32F20x_D6，串口 4 为同步异步收发器（USART4）。

18.1 主要特征

- 全双工或半双工的同步或异步通信
- NRZ 数据格式
- 分数波特率发生器，最高 9Mbps
- 可编程数据长度
- 可配置的停止位
- 支持 LIN，IrDA 编码器，智能卡
- 支持 DMA
- 多种中断源

18.2 概述

图 18-1 通用同步/异步收发器的结构框图



当 TE（发送使能位）置位时，发送移位寄存器里的数据在 TX 引脚上输出，时钟在 CK 引脚上输

出。在发送时，最先移出的是最低有效位，每个数据帧都由一个低电平的起始位开始，然后发送器根据 M（字长）位上的设置发送八位或九位的数据字，最后是数目可配置的停止位。如果配有奇偶检验位，数据字的最后一位为校验位。在 TE 置位后会发送一个空闲帧，空闲帧是 10 位或 11 位高电平，包含停止位。断开帧是 10 位或 11 位低电平，后跟着停止位。

18.3 波特率发生器

收发器的波特率 = FCLK/(16*USARTDIV)，FCLK 是 APBx 的时钟，即 PCLK1 或 PCLK2， USART1 模块使用 PCLK2，其余的使用 PCLK1。USARTDIV 的值是根据 USART_BRR 中的 DIV_M 和 DIV_F 两个域决定的，具体计算的公式为：

USARTDIV = DIV_M+(DIV_F/16)

需要注意的是，波特率产生器产生的比特率不一定能刚好生成用户所需要的波特率，这其中可能是存在偏差。除了尽量取接近的值，减小偏差的方法还可以是增大 APBx 的时钟。比如设定波特率为 115200bps 的时，USARTDIV 的值设为 39.0625，在最高频率时可以得到刚好 115200bps 的波特率，但是如果你需要 921600bps 的波特率时，计算的 USARTDIV 是 4.88，但是实际上在 USART_BRR 里填入的值最接近只能是 4.875，实际产生的波特率是 923076bps，误差达到 0.16%。

发送方发出的串口波形传到接收端时，接收方和发送方的波特率是有一定误差的。误差主要来自三个方面：接收方和发送方实际的波特率不一致；接收方和发送方的时钟有误差；波形在线路中产生的变化。外设模块的接收器是有一定接收容差能力的，当以上三个方面产生的总偏差之和小于模块的容差能力极限时，这个总偏差不影响收发。模块的容差能力极限受是否采用分数波特率和 M 位（数据域字长）影响，采用分数波特率和使用 9 位数据域长度会使容差能力极限降低，但不低于 3%。

18.4 同步模式

同步模式使得系统在使用 USART 模块时可以输出时钟信号。在开启同步模式对外发送数据时，CK 引脚会同时对外输出时钟。

开启同步模式的方式是对控制寄存器 2（R16_USARTx_CTLR2）的 CLKEN 位置位，但同时需要关闭 LIN 模式、智能卡模式、红外模式和半双工模式，即保证 SCEN、HDSEL 和 IREN 位处于复位状态，这三位在控制寄存器 3（R16_USARTx_CTLR3）中。

同步模式使用的要点在于时钟的输出控制。有以下几点需要注意：

USART 模块同步模式只工作在主模式，即 CK 引脚只输出时钟，不接收输入；

只在 TX 引脚输出数据时输出时钟信号；

LBCL 位决定在发送最后一位数据位时是否输出时钟，CPOL 位决定时钟的极性，CPHA 决定时钟的相位，这三个位在控制寄存器 2（R16_USARTx_CTLR2）中，这三个位需要在 TE 和 RE 未被使能的情况下设置，具体区别见图 18-2。

接收器在同步模式下只会在输出时钟时采样，需要从设备保持一定的信号建立时间和保持时间，具体见图 18-3。

图 18-2 USART 时钟时序示例(M=0)

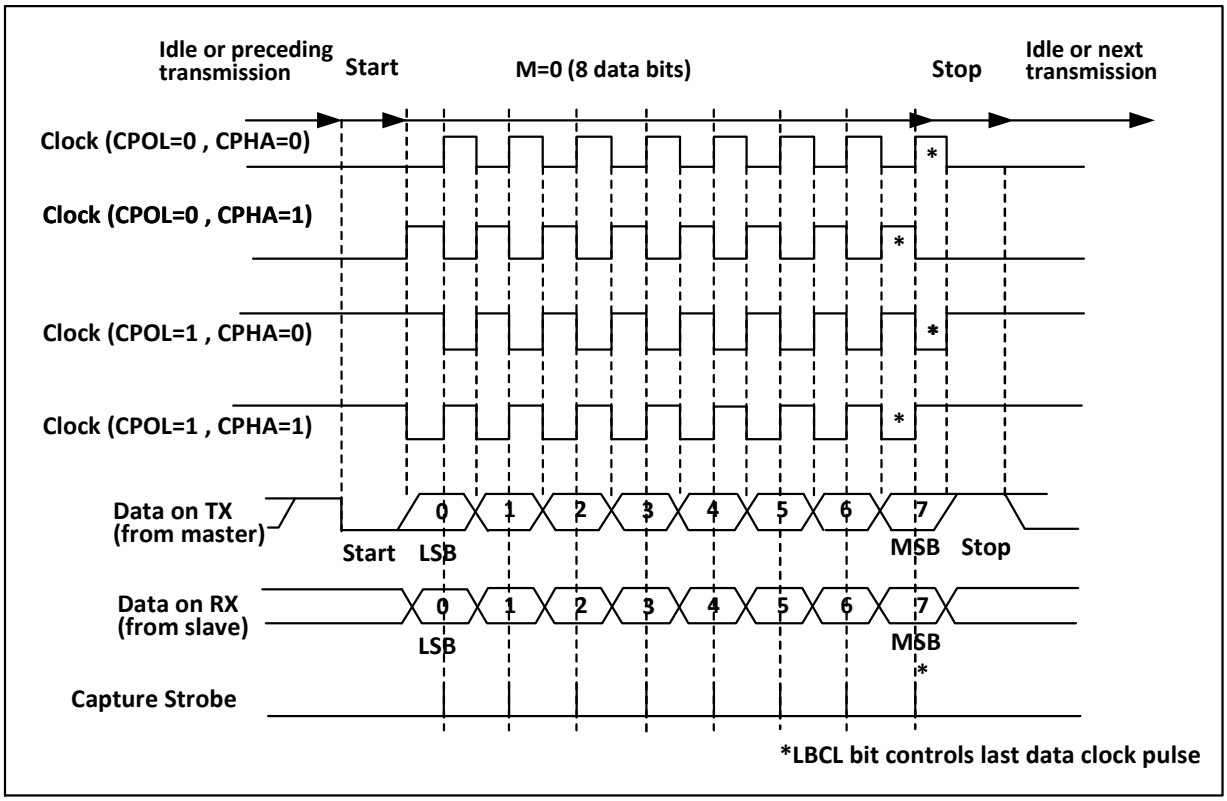
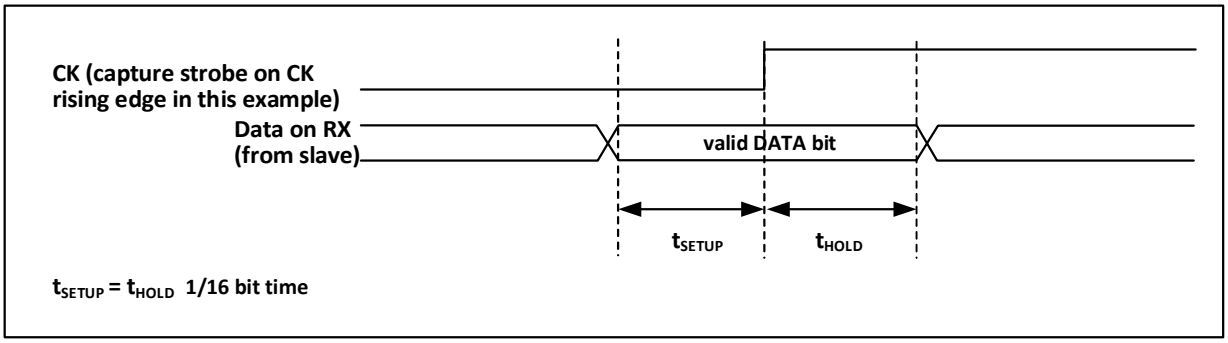


图 18-3 数据采样保持时间



18.5 单线半双工模式

半双工模式支持使用单个引脚（只使用 TX 引脚）来接收和发送，TX 引脚和 RX 引脚在芯片内部连接。

开启半双工模式的方式是对控制寄存器 3（R16_USARTx_CTLR3）的 HDSEL 位置位，但同时需要关闭 LIN 模式、智能卡模式、红外模式和同步模式，即保证 SCEN、CLKEN 和 IREN 位处于复位状态，这三位在控制寄存器 2 和 3（R16_USARTx_CTLR2 和 R16_USARTx_CTLR3）中。

设置成半双工模式之后，需要把 TX 的 IO 口设置成浮空输入或开漏输出高模式。在 TE 置位的情况下，只要将数据写到数据寄存器上，就会发送出去。特别要注意的是，半双工模式可能会出现多设备使用单总线收发时的总线冲突，这需要用户用软件自行避免。

18.6 智能卡

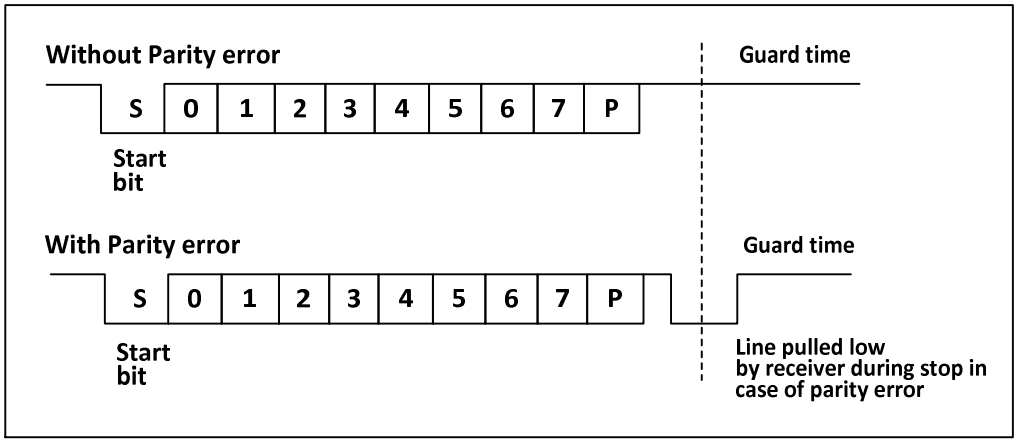
智能卡模式支持 ISO7816-3 协议访问智能卡控制器。

开启智能卡模式的方式是对控制寄存器 3（R16_USARTx_CTLR3）的 SCEN 位置位，但同时需要关

闭 LIN 模式、半双工模式和红外模式，即保证 LINEN、HDSEL 和 IREN 位处于复位状态，但是可以开启 CLKEN 来输出时钟，这些位在控制寄存器 2 和 3（R16_USARTx_CTLR2 和 R16_USARTx_CTLR3）中。

为了支持智能卡模式，USART 应当被置为 8 位数据位外加 1 位校验位，它的停止位建议配置成发送和接收都为 1.5 位，智能卡模式是一种单线半双工的协议，它使用 TX 线作为数据通讯，应当被配置为开漏输出加上拉。当接收方接收一帧数据检测到奇偶校验错误时，会在停止位时，发出一个 NACK 信号，即在停止位期间主动把 TX 拉低一个周期，发送方检测到 NACK 信号后，会产生帧错误，应用程序据此可以重发。图 17-4 展示了正确情况下和发生奇偶校验错误情况下的 TX 引脚上的波形图。USART 的 TC 标志（发送完成标志）可以延迟 GT（保护时间）个时钟产生，接收方也不会将自己置的 NACK 信号认成起始位。

图 18-4 (未)发生奇偶校验错误示意图



在智能卡模式下，CK 引脚使能后输出的波形和通讯无关，它仅仅是给智能卡提供时钟的，它的值是 APB 时钟再经过五位可设置的时钟分频（分频值为 PSC 的两倍，最高 62 分频）。

18.7 IrDA

USART 模块支持控制 IrDA 红外收发器进行物理层通信。使用 IrDA 必须清除 LINEN、STOP、CLKEN、SCEN 和 HDSEL 位。USART 模块和 SIR 物理层（红外收发器）之间使用 NRZ（不归零）编码，最高支持到 115200bps 速率。

IrDA 是一个半双工的协议，如果 UASRT 正在给 SIR 物理层发数据，那么 IrDA 解码器将会忽视新发来的红外信号,如果 USART 正在接受从 SIR 发来的数据,那么 SIR 不会接受来自 USART 的信号。USART 发给 SIR 和 SIR 发给 USART 的电平逻辑是不一样的，SIR 接收逻辑中，高电平为 1，低电平为 0，但是在 SIR 发送逻辑中，高电平为 0，低电平为 1。

18.8 DMA

USART 模块支持 DMA 功能，可以利用 DMA 实现快速连续收发。当启用 DMA 时，TXE 被置位时，DMA 就会从设定的内存空间向发送缓冲区写数据。当使用 DMA 接收时，每次 RXNE 置位后，DMA 就会将接收缓冲区里的数据转移到特定的内存空间。

18.9 中断

USART 模块支持多种中断源，包括发送数据寄存器空（TXE）、CTS、发送完成（TC）、接收数据就绪（TXNE）、数据溢出（ORE）、线路空闲（IDLE）、奇偶校验出错（PE）、断开标志（LBD）、噪

声（NE）、多缓冲通信的溢出（ORT）和帧错误（FE）等等。

表 18-1 中断和对应的使能位的关系

中断源	使能位
数据寄存器空（TXE）	TXEIE
允许发送（CTS）	CTSIE
发送完成（TC）	TCIE
接收数据就绪（TXNE）	TXNEIE
数据溢出（ORE）	
线路空闲（IDLE）	IDLEIE
奇偶校验出错（PE）	PEIE
断开标志（LBD）	LBDIE
噪声（NE）	EIE
多缓冲通信的溢出（ORT）	
多缓冲通信的帧错误（FE）	

18.10 寄存器描述

表 18-2 USART1 相关寄存器列表

名称	访问地址	描述	复位值
R32_USART1_STATR	0x40013800	UASRT1 状态寄存器	0x000000C0
R32_USART1_DATAR	0x40013804	UASRT1 数据寄存器	0x000000XX
R32_USART1_BRR	0x40013808	UASRT1 波特率寄存器	0x00000000
R32_USART1_CTLR1	0x4001380C	UASRT1 控制寄存器 1	0x00000000
R32_USART1_CTLR2	0x40013810	UASRT1 控制寄存器 2	0x00000000
R32_USART1_CTLR3	0x40013814	UASRT1 控制寄存器 3	0x00000000
R32_USART1_GPR	0x40013818	UASRT1 保护时间和预分频寄存器	0x00000000

表 18-3 USART2 相关寄存器列表

名称	访问地址	描述	复位值
R32_USART2_STATR	0x40004400	UASRT2 状态寄存器	0x000000C0
R32_USART2_DATAR	0x40004404	UASRT2 数据寄存器	0x000000XX
R32_USART2_BRR	0x40004408	UASRT2 波特率寄存器	0x00000000
R32_USART2_CTLR1	0x4000440C	UASRT2 控制寄存器 1	0x00000000
R32_USART2_CTLR2	0x40004410	UASRT2 控制寄存器 2	0x00000000
R32_USART2_CTLR3	0x40004414	UASRT2 控制寄存器 3	0x00000000
R32_USART2_GPR	0x40004418	UASRT2 保护时间和预分频寄存器	0x00000000

表 18-4 USART3 相关寄存器列表

名称	访问地址	描述	复位值
R32_USART3_STATR	0x40004800	UASRT3 状态寄存器	0x000000C0
R32_USART3_DATAR	0x40004804	UASRT3 数据寄存器	0x000000XX
R32_USART3_BRR	0x40004808	UASRT3 波特率寄存器	0x00000000
R32_USART3_CTLR1	0x4000480C	UASRT3 控制寄存器 1	0x00000000
R32_USART3_CTLR2	0x40004810	UASRT3 控制寄存器 2	0x00000000

R32_USART3_CTLR3	0x40004814	UASRT3 控制寄存器 3	0x00000000
R32_USART3_GPR	0x40004818	UASRT3 保护时间和预分频寄存器	0x00000000

表 18-5 UART4 相关寄存器列表

名称	访问地址	描述	复位值
R32_UART4_STATR	0x40004C00	UART4 状态寄存器	0x000000C0
R32_UART4_DATAR	0x40004C04	UART4 数据寄存器	0x000000XX
R32_UART4_BRR	0x40004C08	UART4 波特率寄存器	0x00000000
R32_UART4_CTLR1	0x40004C0C	UART4 控制寄存器 1	0x00000000
R32_UART4_CTLR2	0x40004C10	UART4 控制寄存器 2	0x00000000
R32_UART4_CTLR3	0x40004C14	UART4 控制寄存器 3	0x00000000
R32_UART4_GPR	0x40004C18	UART4 保护时间和预分频寄存器	0x00000000

表 18-6 UART5 相关寄存器列表

名称	访问地址	描述	复位值
R32_UART5_STATR	0x40005000	UART5 状态寄存器	0x000000C0
R32_UART5_DATAR	0x40005004	UART5 数据寄存器	0x000000XX
R32_UART5_BRR	0x40005008	UART5 波特率寄存器	0x00000000
R32_UART5_CTLR1	0x4000500C	UART5 控制寄存器 1	0x00000000
R32_UART5_CTLR2	0x40005010	UART5 控制寄存器 2	0x00000000
R32_UART5_CTLR3	0x40005014	UART5 控制寄存器 3	0x00000000
R32_UART5_GPR	0x40005018	UART5 保护时间和预分频寄存器	0x00000000

表 18-7 UART6 相关寄存器列表

名称	访问地址	描述	复位值
R32_UART6_STATR	0x40001800	UART6 状态寄存器	0x000000C0
R32_UART6_DATAR	0x40001804	UART6 数据寄存器	0x000000XX
R32_UART6_BRR	0x40001808	UART6 波特率寄存器	0x00000000
R32_UART6_CTLR1	0x4000180C	UART6 控制寄存器 1	0x00000000
R32_UART6_CTLR2	0x40001810	UART6 控制寄存器 2	0x00000000
R32_UART6_CTLR3	0x40001814	UART6 控制寄存器 3	0x00000000
R32_UART6_GPR	0x40001818	UART6 保护时间和预分频寄存器	0x00000000

表 18-8 UART7 相关寄存器列表

名称	访问地址	描述	复位值
R32_UART7_STATR	0x40001C00	UART7 状态寄存器	0x000000C0
R32_UART7_DATAR	0x40001C04	UART7 数据寄存器	0x000000XX
R32_UART7_BRR	0x40001C08	UART7 波特率寄存器	0x00000000
R32_UART7_CTLR1	0x40001C0C	UART7 控制寄存器 1	0x00000000
R32_UART7_CTLR2	0x40001C10	UART7 控制寄存器 2	0x00000000
R32_UART7_CTLR3	0x40001C14	UART7 控制寄存器 3	0x00000000
R32_UART7_GPR	0x40001C18	UART7 保护时间和预分频寄存器	0x00000000

表 18-9 UART8 相关寄存器列表

名称	访问地址	描述	复位值
R32_UART8_STATR	0x40002000	UART8 状态寄存器	0x000000C0

R32_UART8_DATAR	0x40002004	UART8 数据寄存器	0x000000XX
R32_UART8_BRR	0x40002008	UART8 波特率寄存器	0x00000000
R32_UART8_CTLR1	0x4000200C	UART8 控制寄存器 1	0x00000000
R32_UART8_CTLR2	0x40002010	UART8 控制寄存器 2	0x00000000
R32_UART8_CTLR3	0x40002014	UART8 控制寄存器 3	0x00000000
R32_UART8_GPR	0x40002018	UART8 保护时间和预分频寄存器	0x00000000

18.10.1 USART 状态寄存器（R32_USARTx_STATR）（x=1/2/3/4/5/6/7/8）

偏移地址：0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE

位	名称	访问	描述	复位值
[31:10]	Reserved	RO	保留。	0
9	CTS	RWO	CTS 状态改变标志。如果设置了 CTSE 位，当 nCTS 输出状态改变时，该位将由硬件置高。由软件清零。如果 CTSIE 位已经被置位，则会产生中断。 1：nCTS 状态线上存在变化； 0：nCTS 状态线上没有变化。	0
8	LBD	RWO	LIN 断开检测标志。当检测到 LIN 断开时，该位被硬件置位。由软件清零。如果 LBDIE 已经被置位，则将会产生中断。 1：检测到 LIN 断开； 0：没有检测待 LIN 断开。	0
7	TXE	RO	发送数据寄存器空标志。当 TDR 寄存器中的的数据被硬件转移到移位寄存器的时候，该位被硬件置位。如果 TXEIE 已经被置位时，就会产生中断，对数据寄存器进行写操作，此位将会被复位。 1：数据已经被转移到移位寄存器； 0：数据还没被转移到移位寄存器。	1
6	TC	RWO	发送完成标志。当含有数据的一帧发送完成后，并且 TXE 被置位，则硬件将会此位置位，如果 TCIE 被置位，还会产生对应中断，软件读了此位再写数据寄存器则会清除此位。也可以直接写 0 来清除此位。 1：发送完成； 0：发送还未完成。	1
5	RXNE	RWO	读数据寄存器非空标志，当移位寄存器中的数据被转移到数据寄存器中，该位会被硬件置位。如果 RXNEIE 已经被置位，则还会产生对应的中断。对数据寄存器的读操作可以将该位	0

			清除。也可以直接写 0 来清除该位。 1：数据收到，能够读出； 0：数据还没收到。	
4	IDLE	RO	总线空闲标志。当总线空闲时，该位将会被硬件置位。如果 IDLEIE 已经被置位，则会产生对应的中断。读状态寄存器再读数据寄存器的操作会清除此位。 1：总线正空闲； 0：没有检测到总线空闲。 *注：此位不会被再次置位直到 RXNE 被置位。*	0
3	ORE	RO	过载错误标志。当接收移位寄存器存在数据需要转到数据寄存器时，但是数据寄存器的接收域还有数据未读出时，此位将会被置位。如果 RXNEIE 被置位了，还会产生对应中断。 1：发生过载错误； 0：没有过载错误。 *注：发生过载错误时，数据寄存器的值不会丢失，但是移位寄存器的值会被覆盖。如果设置可 EIE 位，在多缓冲区通讯模式下，ORE 标志位置位会产生中断。*	0
2	NE	RO	噪声错误标志。当检测到噪声错误标志时，由硬件置位。读状态寄存器后，再读数据寄存器的操作会复位此位。 1：检测到噪声； 0：没有检测到噪声。 *注：该位不会产生中断。如果设置了 EIE 位，在多缓冲区通讯模式下，FE 标志位置位会产生中断。*	0
1	FE	RO	帧错误标志。当检测到同步错误，过多的噪声或者断开符，该位将会被硬件置位。读此位再读数据寄存器的操作会复位此位。 1：检测到帧错误； 0：没有检测到帧错误。 *注：该位不会产生中断，如果设置了 EIE 位，在多缓冲区通讯模式下，FE 标志位置位会产生中断。*	0
0	PE	RO	校验错误标志。在接收模式下，如果产生奇偶检验错误，硬件置位此位。读此位再读数据寄存器的操作会复位此位。在清除此位前，软件必须等 RXNE 标志位被置位。如果 PEIE 之前已经被置位，那么此位被置位会产生对应的中断。 1：出现奇偶校验错误； 0：没有检验错误。	0

18.10.2 USART 数据寄存器（USARTx_DATAR）（x=1/2/3/4/5/6/7/8）

偏移地址：0x04

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

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

位	名称	访问	描述	复位值
[31:9]	Reserved	RO	保留。	0
[8:0]	DR	RW	数据寄存器。这个寄存器实际上是接收数据寄存器（RDR）和发送寄存器（TDR）两个寄存器组成，DR 的读写操作起始分别是读接收寄存器（RDR）和写发送寄存器（TDR）。	X

18.10.3 USART 波特率寄存器（USARTx_BRR）（x=1/2/3/4/5/6/7/8）

偏移地址：0x08

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

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:4]	DIV_M	RW	这 12 位定义了分频器除法因子的整数部分。	0
[3:0]	DIV_F	RW	这 4 位定义了分频器除法因子的小数部分。	0

18.10.4 USART 控制寄存器 1（USARTx_CTLR1）（x=1/2/3/4/5/6/7/8）

偏移地址：0x0C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK

位	名称	访问	描述	复位值
[31:14]	Reserved	RO	保留。	0
13	UE	RW	USART 使能位。当此位被置位后，在当前字节传输完成后，USART 的分频器和输出都会停止工作。	0
12	M	RW	字长位。 1：9 个数据位；　　0：8 个数据位。	0
11	WAKE	RW	唤醒位。此位决定了把 USART 唤醒的方法： 1：地址标记；　　0：总线空闲。	0

10	PCE	RW	校验位使能。对于接收方，就是进行对数据的奇偶校验；对于发送方，就是插入校验位。一旦设置了此位，只有当前字节传输完成后，校验位使能才生效。	0
9	PS	RW	奇偶校验选择。0 表示偶校验，1 表示奇校验。设置了该位后，只有当前字节传输完成后，校验位使能才生效。	0
8	PEIE	RW	奇偶检验中断使能位。对此位置位表示允许产生奇偶检验错误中断。	0
7	TXEIE	RW	发送缓冲区空中断使能。对此位置位表示允许产生发送缓冲区空中断。	0
6	TCIE	RW	发送完成中断使能。对此位置位表示允许产生发送完成中断。	0
5	RXNEIE	RW	接收缓冲区非空中断使能。对此位置位表示允许产生接收缓冲区非空中断。	0
4	IDLEIE	RW	总线空闲中断使能。对此位置位表示允许产生总线空闲中断。	0
3	TE	RW	发送使能。置此位会使能发送器。	0
2	RE	RW	接收使能。置此位会使能接收器，接收器开始检测 RX 引脚上的起始位。	0
1	RWU	RW	接收唤醒。该位决定是否把 USART 置于静默模式： 1：接收器处于静默模式； 0：接收器处于正常工作模式。 *注 1：置 RWU 位之前，USART 需要先接收一个数据字节，否则在静默模式下，不能被总线空闲唤醒；* *注 2：当配置成地址标记唤醒时，在 RXNE 被置位时，不能用软件修改 RWU 位。*	0
0	SBK	RW	发送帧断开字符控制位。置此位来发送一个帧断开字符。在断开帧的停止位时，由硬件复位。 1：发送；　　0：不发送。	0

18.10.5 USART 控制寄存器 2（USARTx_CTLR2）（x=1/2/3/4/5/6/7/8）

偏移地址：0x10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LINEN	STOP		CLKEN	CPOL	CPHA	LBCL	Reserved	LBDIE	LBDL	Reserved	ADD[3:0]			

位	名称	访问	描述	复位值
[31:15]	Reserved	R0	保留。	0
14	LINEN	RW	LIN 模式使能位，置位则使能 LIN 模式。在 LIN 模式下，可以使用 SBK 位发送 LIN 同步断开符	0

			号，以及检测 LIN 同步断开符。	
[13:12]	STOP	RW	停止位设置域。这两位来设置停止位。 00：1 个停止位； 01：0.5 个停止位； 10：2 个停止位； 11：1.5 个停止位。	0
11	CLKEN	RW	时钟使能，使能 CK 引脚。 1：使能； 0：禁止。	0
10	CPOL	RW	时钟极性设置位。在同步模式下，可以用该位选择 SLCK 引脚上时钟输出的极性，和 CPHA 一起配合来产生需要的时钟/数据的采样关系。 1：总线空闲时 CK 引脚上保持高电平； 0：总线空闲时 CK 引脚上保持低电平。 *注：使能发送后此位不可被修改。*	0
9	CPHA	RW	时钟相位设置位。在同步模式下，可以用该位选择 SLCK 引脚上的时钟输出的相位，和 CPOL 位一起配合来产生需要的时钟/数据的采样关系。 1：在时钟的第二个边沿进行数据捕获； 0：在时钟的第一个边沿进行数据捕获。 *注：使能发送后此位不可被修改。*	0
8	LBCL	RW	最后一个时钟脉冲控制位。 在同步模式下，使用该位来控制是否在 CK 引脚上输出最后发送的那个数据字节对应的时钟脉冲； 1：最后一位数据的时钟脉冲不从 CK 输出； 0：最后一位数据的时钟脉冲会从 CK 输出。 *注：使能发送后此位不可被修改。*	0
7	Reserved	RW	保留。	0
6	LBDIE	RW	LIN 断开符检测中断使能，该位置位会使能 LBD 引起的中断；	0
5	LBDL	RW	LIN 断开符检测长度，该位用来选择是 11 位还是 10 位的断开符检测。 1：11 位的断开符检测； 0：10 位的断开符检测。	0
4	Reserved	RW	保留。	0
[3:0]	ADD	RW	地址域，用来设置本设备的 USART 节点地址。在多处理器通讯下的静默模式中使用的，使用地址标记来唤醒某个 USART 设备。	0

18.10.6 USART 控制寄存器 3（USARTx_CTLR3）（x=1/2/3/4/5/6/7/8）

偏移地址：0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE

位	名称	访问	描述	复位值
[31:11]	Reserved	RO	保留。	0
10	CTSIE	RW	CTSIE 中断使能位，置此位时在 CTS 被置位时会产生中断。	0
9	CTSE	RW	CTS 使能位，置此位会使能 CTS 流控。	0
8	RTSE	RW	RTS 使能位，置此位会使能 RTS 流控。	0
7	DMAT	RW	DMA 发送使能位。此位置 1 在发送时使用 DMA。	0
6	DMAR	RW	DMA 接收使能位。此位置 1 在接收时使用 DMA。	0
5	SCEN	RW	智能卡模式使能位，置 1 使能智能卡模式。	0
4	NACK	RW	智能卡 NACK 使能位，置此位在校验错误出现时，发送 NACK。	0
3	HDSEL	RW	半双工模式选择位，置此位选择半双工模式。	0
2	IRLP	RW	红外低功耗选择位，置此位在选择红外线时，启用低功耗模式。	0
1	IREN	RW	红外线使能位，置此位使能红外模式。	0
0	EIE	RW	错误使能中断位，置此位后，在 DMAR 被置位的前提下，如果 FE、ORE 或 NE 被置位，就会产生中断。	0

18.10.7 USART 保护时间和预分频寄存器（USARTx_GPR）（x=1/2/3/4/5/6/7/8）

偏移地址：0x18

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

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:8]	GT	RW	保护时间值域。该域规定了以波特率时钟为单位的保护时间。在智能卡模式下，当保护时间过去后，才会设置发送完成标志。	0
[7:0]	PSC	RW	预分频器值域。 在红外低功耗模式下，源时钟被该值（全部 8 位有效）分频，值为 0 时表示保留； 在红外正常模式下，此位只能被设置为 1； 在智能卡模式下，源时钟被该值（低 5 位有效）的两倍分频，来给智能卡提供时钟，值为 0 表示保留。	0

第 19 章 两线通信总线（I2C）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

内部集成电路总线（I2C）广泛用在微控制器和传感器及其他片外模块的通讯上，它本身支持多主多从模式，仅仅使用两根线（SDA 和 SCL）就能以 100KHz（标准）和 400KHz（快速）两种速度通讯。I2C 总线还兼容 SMBus 协议，不仅支持 I2C 的时序，还支持仲裁、定时和 DMA，拥有 CRC 校验功能。

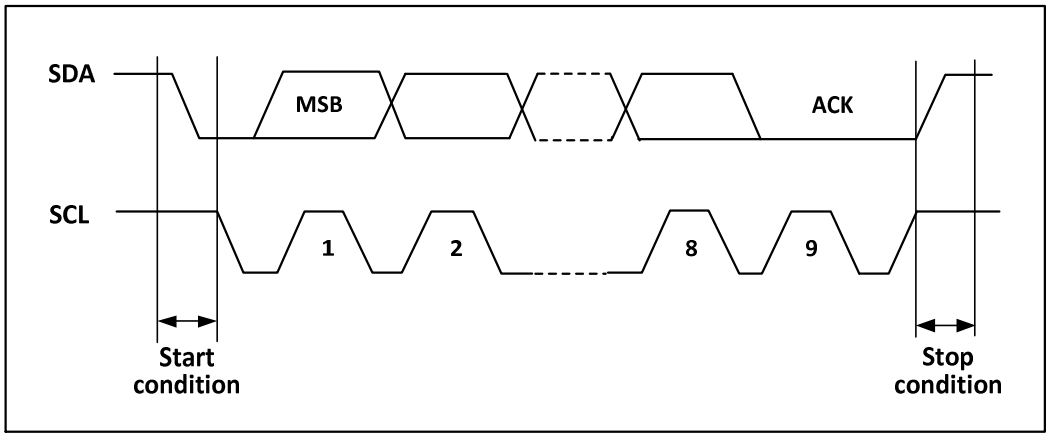
19.1 主要特征

- 支持主模式和从模式
- 支持 7 位或 10 位地址
- 从设备支持双 7 位地址
- 支持两种速度模式：100KHz 和 400KHz
- 多种状态模式，多种错误标志
- 支持加长的时钟功能
- 2 个中断向量
- 支持 DMA
- 支持 PEC
- 兼容 SMBus

19.2 概述

I2C 是半双工的总线，它同时只能运行在下列四种模式中之一：主设备发送模式、主设备接收模式、从设备发送模式和从设备接收模式。I2C 模块默认工作在从模式，在产生起始条件后，会自动地切换到主模式，当仲裁丢失或产生停止信号后，会切换到从模式。I2C 模块支持多主机功能。工作在主模式时，I2C 模块会主动发出数据和地址。数据和地址都以 8 位为单位进行传输，高位在前，低位在后，在起始事件后的是一个字节（7 位地址模式下）或两个字节（10 位地址模式下）地址，主机每发送 8 位数据或地址，从机需要回复一个应答 ACK，即把 SDA 总线拉低，如图 19-1 所示。

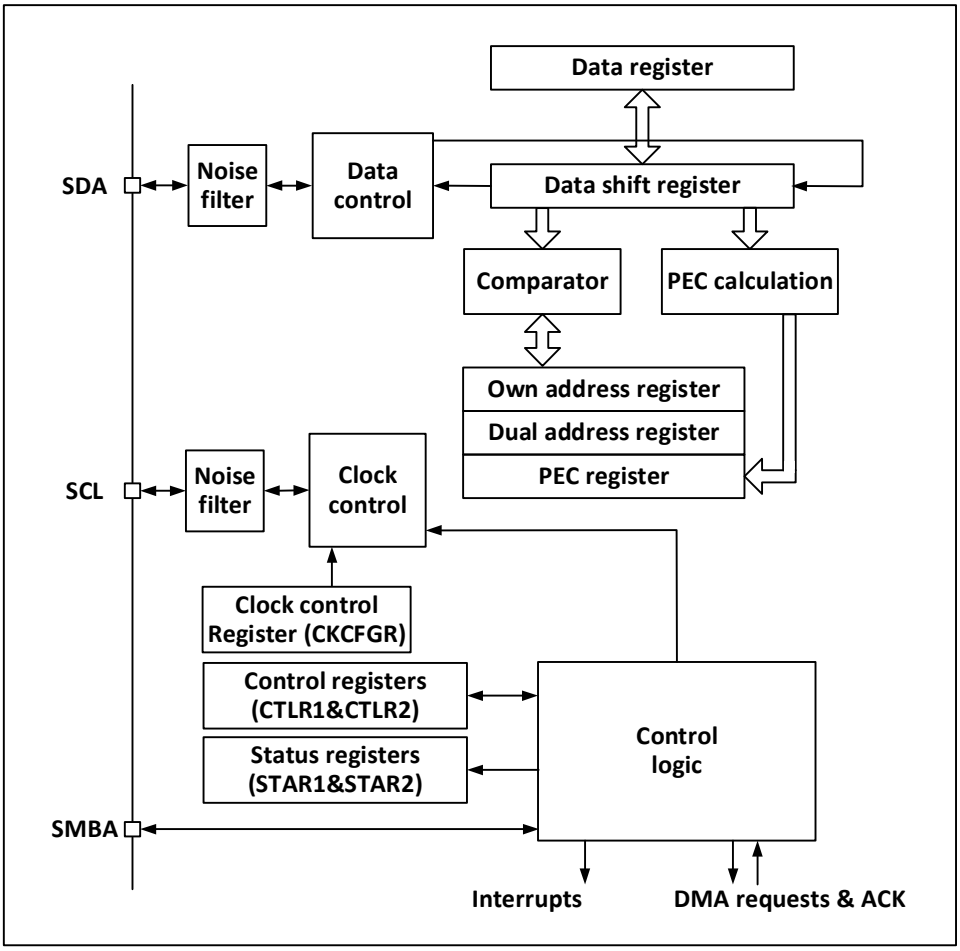
图 19-1 I2C 时序图



为了正常使用必须给 I2C 输入正确的时钟，其中标准模式下，输入时钟最低为 2MHz，在快速模式下，输入时钟最低为 4MHz。

图 19-2 是 I2C 模块功能框图。

图 19-2 I2C 功能框图



19.3 主模式

主模式时，I2C 模块主导数据传输并输出时钟信号，数据传输以开始事件开始，以结束事件结束。使用主模式通讯的步骤为：

在控制寄存器 2（R16_I2Cx_CTLR2）和时钟控制寄存器（R16_I2Cx_CKCFGR）中设置正确的时钟；

在上升沿寄存器（R16_I2Cx_RTR）设置合适的上升沿；

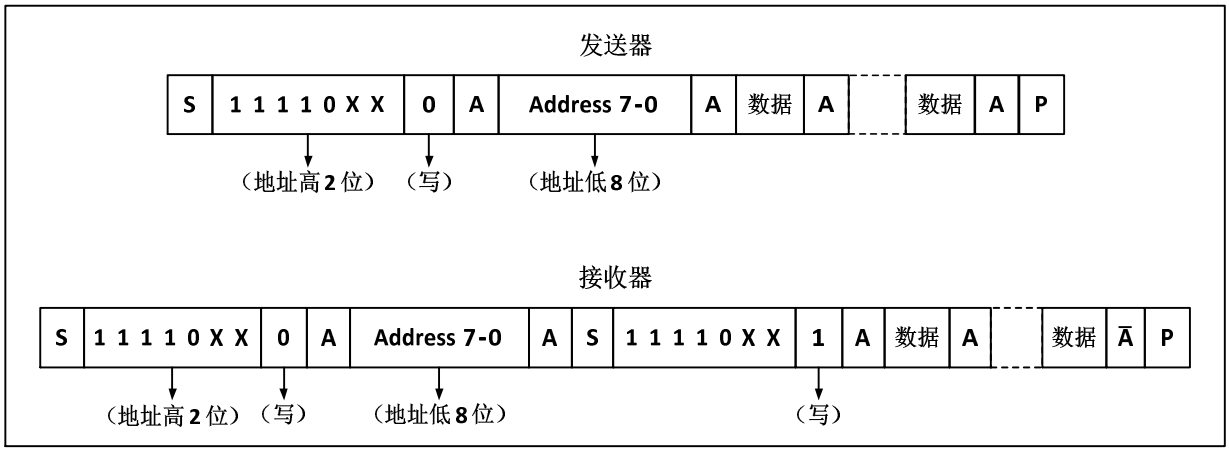
在控制寄存器（R16_I2Cx_CTLR1）中置 PE 位启动外设；

在控制寄存器（R16_I2Cx_CTLR1）中置 START 位，产生起始事件。在置 START 位后，I2C 模块会自动切换到主模式，MSL 位会置位，产生起始事件，在产生起始事件后，SB 位会置位，如果 ITEVTEN 位（在 R16_I2Cx_CTLR2）被置位，则会产生中断。此时应该读取状态寄存器 1（R16_I2Cx_STAR1），写从地址到数据寄存器后，SB 位会自动清除；

如果是使用 10 位地址模式，那么写数据寄存器发送头序列（头序列为 11110xx0b，其中的 xx 位是 10 位地址的最高两位）。在发送完头序列之后，状态寄存器的 ADD10 位会被置位，如果 ITEVTEN 位已经置位，则会产生中断，此时应读取 R16_I2Cx_STAR1 寄存器后，写第二个地址字节到数据寄存器后，清除 ADD10 位。然后写数据寄存器发送第二个地址字节，在发送完第二个地址字节后，状态寄存器的 ADDR 位会被置位，如果 ITEVTEN 位已经置位，则会产生中断，此时应读取 R16_I2Cx_STAR1 寄存器后再读一次 R16_I2Cx_STAR2 寄存器以清除 ADDR 位；如果使用的是 7 位地址模式，那么写数据寄存器发送地址字节，在发送完地址字节后，状态寄存器的 ADDR 位会被置位，如果 ITEVTEN 位已经置位，则会产生中断，此时应读取 R16_I2Cx_STAR1 寄存器后再读一次 R16_I2Cx_STAR2 寄存器以清除 ADDR 位；在 7 位地址模式下，发送的第一个字节为地址字节，头 7 位代表的是目标从设备地址，第 8 位决定了后续报文的方向，0 代表是主设备写入数据到从设备，1 代表是主设备向从设备读取信息。

在 10 位地址模式下，如图 18-3 所示，在发送地址阶段，第一个字节为 11110xx0，xx 为 10 位地址的最高 2 位，第二个字节为 10 位地址的低 8 位。若后续进入主设备发送模式，则继续发送数据；若后续准备进入主设备接收模式，则需要重新发送一个起始条件，跟随发送一个字节为 11110xx1，然后进入主设备接收模式。

图 19-3 10 位地址时主机收发数据示意图



发送模式时，主设备内部的移位寄存器将数据从数据寄存器发送到 SDA 线上，当主设备接收到 ACK 时，状态寄存器 1（R16_I2Cx_STAR1）的 TxE 被置位，如果 ITEVTEN 和 ITBUFEN 被置位，还会产生中断。向数据寄存器写入数据将会清除 TxE 位。如果 TxE 位被置位且上次发送数据之前没有新的数据被写入数据寄存器，那么 BTF 位会被置位，在其被清除之前，SCL 将保持低电平，读 R16_I2Cx_STAR1 后，向数据寄存器写入数据将会清除 BTF 位。而在接收模式时，I2C 模块会从 SDA 线接收数据，通过移位寄存器写进数据寄存器。在每个字节之后，如果 ACK 位被置位，那么 I2C 模块将会发出一个应答低电平，同时 RxNE 位会被置位，如果 ITEVTEN 和 ITBUFEN 被置位，还会产生中断。如果 RxNE 被置位且在新的数据被接收前，原有的数据没有被读出，则 BTF 位将被置位，在清除 BTF 之前，SCL 将保持低电平，读取 R16_I2Cx_STAR1 后，再读取数据寄存器将会清除 BTF 位。

主设备在结束发送数据时，会主动发一个结束事件，即置 STOP 位。在接收模式时，主设备需要在最后一个数据位的应答位置 NAK。注意，产生 NAK 后，I2C 模块将会切换至从模式。

19.4 从模式

从模式时，I2C 模块能识别它自己的地址和广播呼叫地址。软件能控制开启或禁止广播呼叫地址的识别。一旦检测到起始事件，I2C 模块将 SDA 的数据通过移位寄存器与自己的地址（位数取决于 ENDUAL 和 ADDMODE）或广播地址（ENGC 置位时）相比较，如果不匹配将会忽略，直到产生新的起始事件。如果与头序列相匹配，则会产生一个 ACK 信号并等待第二个字节的地址；如果第二字节的地址也匹配或 7 位地址情况下全段地址匹配，那么：首先产生一个 ACK 应答；ADDR 位被置位，如果 ITEVTEN 位已经置位，那么还会产生相应的中断；如果使用的是双地址模式（ENDUAL 位被置位），还需要读取 DUALF 位来判断主机唤起的是哪一个地址。

从模式默认是接收模式，在接收的头序列的最后一位为 1，或 7 位地址最后一位为 1 时（取决于第一次接收到头序列还是普通的 7 位地址），I2C 模块将进入到发送器模式，TRA 位将指示当前是接收器还是发送器模式。

发送模式时，在清除 ADDR 位后，I2C 模块将字节从数据寄存器通过移位寄存器发送到 SDA 线上。在收到一个应答 ACK 后，TxE 位将被置位，如果设置了 ITEVTEN 和 ITBUFEN，还会产生一个中断。如果 TxE 被置位但在下一个数据发送结束前没有新的数据被写入数据寄存器时，BTF 位将被置位。在清除 BTF 前，SCL 将保持低电平，读取状态寄存器 1（R16_I2Cx_STAR1）后，再向数据寄存器写入数据

将会清除 BTF 位。

接收模式时，在 ADDR 被清除后，I2C 模块将 SDA 上的数据通过移位寄存器存进数据寄存器，在每接收到一个字节后，I2C 模块都会置一个 ACK 位，并置 RxNE 位，如果设置了 ITEVTEN 和 ITBUFEN，还会产生一个中断。如果 RxNE 被置位，且在接收到新的数据前旧的数据没有被读出，那么 BTF 会被置位。在清除 BTF 位之前 SCL 会保持低电平。读取状态寄存器 1（R16_I2Cx_STAR1）并读取数据寄存器里的数据会清除 BTF 位。

当 I2C 模块检测到停止事件时，将置 STOPF 位，如果设置了 ITEVFEN 位，还会产生一个中断。用户需要读取状态寄存器（R16_I2Cx_STAR1）再写控制寄存器（比如复位控制字 SWRST）来清除。

19.5 错误

19.5.1 总线错误 BERR

在传输地址或数据期间，I2C 模块检测到外部的起始或停止事件时，将产生一个总线错误。产生总线错误时，BERR 位被置位，如果设置了 ITERREN 还会产生一个中断。在从模式下，数据被丢弃，硬件释放总线。如果是起始信号，硬件会认为是重启信号，开始等待地址或停止信号；如果是停止信号，则提前按正常的停止条件操作。在主模式下，硬件不会释放总线，同时不影响当前传输，由用户代码决定是否中止传输。

19.5.2 应答错误 AF

当 I2C 模块检测到一个字节后没有应答时，会产生应答错误。产生应答错误时：AF 会被置位，如果设置了 ITERREN 还会产生一个中断；遇到 AF 错误，如果 I2C 模块工作在从模式，硬件必须释放总线，如果处于主模式，软件必须生成一个停止事件。

19.5.3 仲裁丢失 ARLO

当 I2C 模块检测到仲裁丢失时，产生仲裁丢失错误。产生仲裁丢失错误时：ARLO 位被置位，如果设置了 ITERREN 还会产生一个中断；I2C 模块切换到从模式，并不再响应针对它的从地址发起的传输，除非有主机发起新的起始事件；硬件会释放总线。

19.5.4 过载/欠载错误 OVR

- 过载错误：

在从机模式下，如果禁止时钟延长，I2C 模块正在接收数据，如果已经接受到一个字节的数据，但是上一次接收到数据还没有被读出，则会产生过载错误。发生过载错误时，最后收到的字节将被丢弃，发送方应当重发最后一次发送的字节。

- 欠载错误：

在从模式下，如果禁止时钟延长，I2C 模块正在发送数据，如果在下一个字节的时钟到来之前新的数据还没有被写入到数据寄存器，那么将产生欠载错误。在发生欠载错误时，前一次数据寄存器里的数据将被发送两次，如果发生欠载错误，那么接收方应该丢弃重复收到的数据。为了不产生欠载错误，I2C 模块应当在下一个字节的第一个上升沿之前将数据写入数据寄存器。

19.6 时钟延长

如果禁止时钟延长，那么就存在发生过载/欠载错误的可能。但如果使能了时钟延长：

- 在发送模式下，如果 TxE 置位且 BTF 置位，SCL 将一直为低，一直等待用户读取状态寄存器，并向数据寄存器写入待发送的数据；

- 在接收模式下，如果 RxNE 置位且 BTF 置位，那么 SCL 在接收到数据后将保持低，直到用户读取状态寄存器，并读取数据寄存器；

由此可见，使能时钟延长可以避免出现过载/欠载错误。

19.7 SMBus

SMBus 也是一种双线接口，它一般应用在系统和电源管理之间。SMBus 和 I2C 有很多相似的地方，例如 SMBus 使用和 I2C 一样的 7 位地址模式，以下是 SMBus 和 I2C 的共同点：

1） 主从通信模式，主机提供时钟，支持多主多从；
2） 两线通讯结构，其中 SMBus 可选一个警示线；
3） 都支持 7 位地址格式。

同时 SMBus 和 I2C 也存在区别：

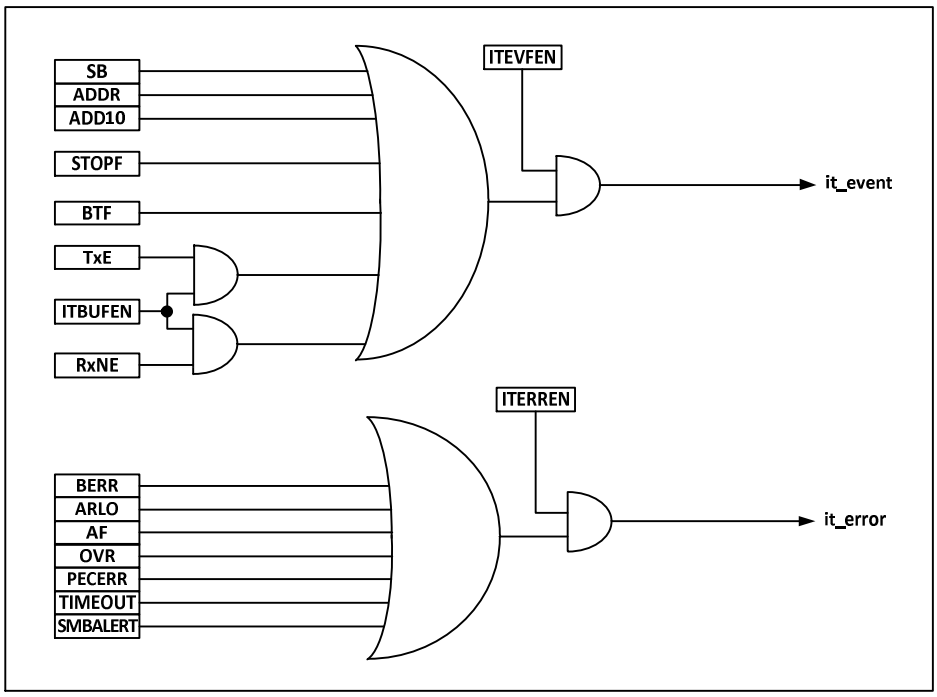
1） I2C 支持的速度最高 400KHz，而 SMBus 支持的最高是 100KHz，且 SMBus 有最小 10KHz 的速度限制；
2） SMBus 的时钟为低超过 35mS 时，会报超时，但 I2C 无此限制；
3） SMBus 有固定的逻辑电平，而 I2C 没有，取决于 VDD；
4） SMBus 有总线协议，而 I2C 没有。

SMBus 还包括设备识别、地址解析协议、唯一的设备标识符、SMBus 提醒和各种总线协议，具体请参考 SMBus 规范 2.0 版本。当使用 SMBus 时，只需要置控制寄存器的 SMBus 位，按需配置 SMBTYPE 位和 ENAARP 位。

19.8 中断

每个 I2C 模块都有两种中断向量，分别是事件中断和错误中断。两种中断支持图 19-4 的中断源。

图 19-4 I2C 中断请求



19.9 DMA

可以使用 DMA 来进行批量数据的收发。使用 DMA 时不能对控制寄存器的 ITBUFEN 位进行置位。

- 利用 DMA 发送

通过将 CTLR2 寄存器的 DMAEN 位置位可以激活 DMA 模式。只要 TxE 位被置位，数据将由 DMA 从设定的内存装载进 I2C 的数据寄存器。需要进行以下设定来为 I2C 分配通道。

1） 向 DMA_PADDRx 寄存器设置 I2Cx_DATAR 寄存器地址，DMA_MADDRx 寄存器中设置存储器地址，这样在每个 TxE 事件后，数据将从存储器送至 I2Cx_DATAR 寄存器。
2） 在 DMA_CNTRx 寄存器中设置所需的传输字节数。在每个 TxE 事件后，此值将被递减。
3） 利用 DMA_CFGRx 寄存器中的 PL[0:1]位配置通道优先级。
4） 设置 DMA_CFGRx 寄存器中的 DIR 位，并根据应用要求可以配置在整个传输完成一半或全部完成时发出中断请求。
5） 通过设置 DMA_CFGRx 寄存器上的 EN 位激活通道。

当 DMA 控制器中设置的数据传输数目已经完成时，DMA 控制器给 I2C 接口发送一个传输结束的 EOT/ EOT_1 信号。在中断允许的情况下，将产生一个 DMA 中断。

- 利用 DMA 接收

置位 CTLR2 寄存器的 DMAEN 后即可进行 DMA 接收模式。使用 DMA 接收时，DMA 将数据寄存器里的数据传送到预设的内存区域。需要以下步骤来为 I2C 分配通道。

1） 向 DMA_PADDRx 寄存器设置 I2Cx_DATAR 寄存器地址，DMA_MADDRx 寄存器中设置存储器地址，这样在每个 RxNE 事件后，数据将从 I2Cx_DATAR 寄存器写入存储器。
2） 在 DMA_CNTRx 寄存器中设置所需的传输字节数。在每个 RxNE 事件后，此值将被递减。
3） 用 DMA_CFGRx 寄存器中的 PL[0:1]配置通道优先级。
4） 清除 DMA_CFGRx 寄存器中的 DIR 位，根据应用要求可以设置在数据传输完成一半或全部完成时发出中断请求。
5） 设置 DMA_CFGRx 寄存器中的 EN 位激活该通道。

当 DMA 控制器中设置的数据传输数目已经完成时，DMA 控制器给 I2C 接口发送一个传输结束的 EOT/EOT_1 信号。在中断允许的情况下，将产生一个 DMA 中断。

19.10 包校验错误

包错误校验(PEC)是为了提供传输的可靠性而增加一项 CRC8 校验的步骤，使用以下多项式对每一位串行数据进行计算：

$$C=X^8+X^2+X+1$$

PEC 计算是由控制寄存器的 ENPEC 位激活，对所有信息字节进行计算，包括地址和读写位在内。在发送时，启用 PEC 会在最后一字节数据之后加上一个字节的 CRC8 计算结果；而在接收模式，在最后一字节被认为是 CRC8 校验结果，如果和内部的计算结果不符合，就会回复一个 NAK，如果是主接收器，无论校验结果正确与否，都会回复一个 NAK。

19.11 调试模式

当系统进入调试模式之后，可以通过 DEBUG 模块的 DBG_I2Cx_SMBUS_TIMEOUT 位来决定 I2CSMBus 的超时控制是继续工作还是停止。

19.12 寄存器描述

表 19-1 I2C1 相关寄存器列表

名称	访问地址	描述	复位值
R16_I2C1_CTLR1	0x40005400	I2C1 控制寄存器 1	0x0000
R16_I2C1_CTLR2	0x40005404	I2C1 控制寄存器 2	0x0000
R16_I2C1_OADDR1	0x40005408	I2C1 地址寄存器 1	0x0000
R16_I2C1_OADDR2	0x4000540C	I2C1 地址寄存器 2	0x0000
R16_I2C1_DATAR	0x40005410	I2C1 数据寄存器	0x0000
R16_I2C1_STAR1	0x40005414	I2C1 状态寄存器 1	0x0000
R16_I2C1_STAR2	0x40005418	I2C1 状态寄存器 2	0x0000
R16_I2C1_CKCFGR	0x4000541C	I2C1 时钟寄存器	0x0000
R16_I2C1_RTR	0x40005400	I2C1 上升时间寄存器	0x0002

表 19-2 I2C2 相关寄存器列表

名称	访问地址	描述	复位值
R16_I2C2_CTLR1	0x40005800	I2C2 控制寄存器 1	0x0000
R16_I2C2_CTLR2	0x40005804	I2C2 控制寄存器 2	0x0000
R16_I2C2_OADDR1	0x40005808	I2C2 地址寄存器 1	0x0000
R16_I2C2_OADDR2	0x4000580C	I2C2 地址寄存器 2	0x0000
R16_I2C2_DATAR	0x40005810	I2C2 数据寄存器	0x0000
R16_I2C2_STAR1	0x40005814	I2C2 状态寄存器 1	0x0000
R16_I2C2_STAR2	0x40005818	I2C2 状态寄存器 2	0x0000
R16_I2C2_CKCFGR	0x4000581C	I2C2 时钟寄存器	0x0000
R16_I2C2_RTR	0x40005800	I2C2 上升时间寄存器	0x0002

19.12.1 I2C 控制寄存器（I2Cx_CTLR1）（x=1/2）

偏移地址：0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	Reserved	ALERT	PEC	POS	ACK	STOP	START	NOSTRETCH	ENGC	ENPEC	ENARP	SMBTYPE	Reserved	SMBUS	PE

位	名称	访问	描述	复位值
15	SWRST	RW	软件重置,用户代码置此位会使 I2C 外设重置。在复位前确定 I2C 总线的引脚被释放，总线处于空闲状态。 *注：该位可以在总线上没有检测到停止条件但是 busy 位为 1 时，重置 I2C 模块。*	0
14	Reserved	RO	保留。	0
13	ALERT	RW	SMBus 提醒位，用户代码可以设置此位或清除此位；当 PE 置位后，此位可以被硬件清除。 1：驱动 SMBusALERT 引脚使其变低，响应地址头应紧跟在 ACK 信号后面； 0：释放 SMBusALERT 引脚使其变高，响应地址	0

			头应紧跟在 NACK 信号后面。	
12	PEC	RW	数据包出错检测使能位，置此位启用数据包出错检测。用户代码可以对此位置位或清零；当 PEC 被传输后，产生开始或结束信号，或 PE 位清 0 时，硬件清零该位； 1：带 PEC； 0：不带 PEC。 *注：仲裁丢失时，PEC 失效。*	0
11	POS	RW	ACK 和 PEC 位置设置位，该位可以被用户代码置位或清零，在 PE 被清零后，可以被硬件清除； 1：ACK 位控制在移位寄存器里接收的下一个字节的 ACK 或 NAK。PEC 移位寄存器里接收的下一字节是 PEC； 0：ACK 位控制当前移位寄存器内正在接受的字节的 ACK 或 NAK。PEC 位表明当位前移位寄存器的字节是 PEC。 *注：POS 位在 2 字节数据接收中的用法如下：必须在接收之前配置好。为了 NACK 第 2 个字节，必须在清除 ADDR 位后立刻清除 ACK 位；为了检测第二个字节的 PEC，必须在 ADDR 事件发生后，配置 POS 位后设置 PEC 位。*	0
10	ACK	RW	应答使能位，该位可以被用户代码置位或清零，当 PE 位被置位时，该位可以被硬件清除； 1：在接收到一个字节后返回一个应答； 0：不设应答。	0
9	STOP	RW	停止事件产生位，该位可以被用户代码置位或清零，或当检测到停止事件时，由硬件清除，或检测到超时错误时，由硬件将其置位。 主模式下： 1：在当前字节传输或当前起始条件发出后产生停止事件； 0：无停止事件产生。 从模式下： 1：在当前字节传输后释放 SCL 和 SDA 线； 0：无停止事件产生。	0
8	START	RW	起始事件产生位，该位可以被用户代码置位或清零，当起始条件发出后或 PE 被清零时，由硬件清零。 主模式下： 1：重复产生起始事件； 0：无起始事件产生。 从模式下： 1：当总线空闲时，产生起始事件； 0：无起始事件产生。	0
7	NOSTRETCH	RW	禁止时钟延长位，此位用于在 ADDB 或 BTF 标	0

			志被置位的情况下，禁止从模式下的时钟延长，直至被软件清零。 1：禁止时钟延长； 0：允许时钟延长。	
6	ENGC	RW	广播呼叫使能位，置此位使能广播呼叫，应答广播地址 00h。	0
5	ENPEC	RW	PEC 使能位，置此位开启 PEC 计算。	0
4	ENARP	RW	ARP 使能位，置此位使能 ARP。 如果 SMBTYPE=0，则使用 SMBus 设备的默认地址；如果 SMBTYPE=1，则使用 SMBus 的主地址。	0
3	SMBTYPE	RW	SMBus 设备类型，置 1 为 SMBus 主设备，置 0 为 SMBus 从设备。	0
2	Reserved	RO	保留。	0
1	SMBUS	RW	SMBus 模式选择位，置 1 为使用 SMBus 模式，置 0 为使用 I2C 模式。	0
0	PE	RW	I2C 外设使能位。 1：启用 I2C 模块； 0：禁用 I2C 模块。	0

19.12.2 I2C 控制寄存器 2（I2Cx_CTLR2）（x=1/2）

偏移地址：0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			LAST	DMAEN	ITBUFEN	ITEVTEN	ITERREN	Reserved		FREQ[5:0]					

位	名称	访问	描述	复位值
[15:13]	Reserved	RO	保留。	0
12	LAST	RW	DMA 最后一次传输设置位。 1：下一次 DMA 的 EOT 是最后的传输； 0：下一次 DMA 的 EOT 不是最后的传输。 *注：该位在主接收模式使用，可以在最后一次接收数据时产生一个 NAK。*	0
11	DMAEN	RW	DMA 请求使能位，置此位在 TxE 或 RxEN 被置位时允许 DMA 请求。	0
10	ITBUFEN	RW	缓冲器中断使能位。 1：当 TxE 或 RxEN 被置位时，产生事件中断； 0：当 TxE 或 RxEN 被置位时，不产生中断。	0
9	ITEVTEN	RW	时间中断使能位，置此位使能事件中断。 在下列条件下，将产生此中断： SB=1（主模式）； ADDR=1（主从模式）； ADDR10=1（主模式）； STOPF=1（从模式）； BTF=1，但是没有 TxE 或 RxEN 事件； 如果 ITBUFEN=1，TxE 事件为 1；	0

			如果 ITBUFEN=1，RxNE 事件为 1。	
8	ITERREN	RW	出错中断使能位，置位表示允许出错中断。在下列条件下，将产生该中断；BERR=1；ARLO=1；AF=1；OVR=1；PECERR=1；TIMEOUT=1；SMBAlert=1。	0
[7:6]	Reserved	RO	保留。	0
[5:0]	FREQ	RW	I2C 模块时钟频率域，必须输入正确的时钟频率以产生正确的时序，允许的范围在 2-36MHz 之间。必须设置在 000010b 到 100100b 之间，单位为 MHz。	0

19.12.3 I2C 地址寄存器 1（I2Cx_OADDR1）（x=1/2）

偏移地址：0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD MODE	MUST1	Reserved				ADD[9:8]		ADD[7:1]							ADD0

位	名称	访问	描述	复位值
15	ADDMODE	RW	地址模式。 1：10 位从机地址（不响应 7 位地址）； 0：7 位从机地址（不响应 10 位地址）。	0
14	MUST1	RW1	必须始终由软件写 1。	0
[13:10]	Reserved	RO	保留。	0
[9:8]	ADD9_8	RW	接口地址，在使用 10 位地址时为第 9-8 位，在使用 7 位地址时忽略。	0
[7:1]	ADD7_1	RW	接口地址，第 7-1 位。	0
0	ADD0	RW	接口地址，使用 10 位地址时为第 0 位，在使用 7 位地址时忽略。	0

19.12.4 I2C 地址寄存器 2（I2Cx_OADDR2）（x=1/2）

偏移地址：0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADD2[7:1]							ENDUAL

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
[7:1]	ADD2	RW	接口地址，双地址模式下地址的 7-1 位。	0
0	ENDUAL	RW	双地址模式使能位，置此位可以让 ADD2 也能被识别。	0

19.12.5 I2C 数据寄存器（I2Cx_DATAR）（x=1/2）

偏移地址：0x10

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

位	名称	访问	描述	复位值
15:8	Reserved	RO	保留。	0
7:0	DATAR	RW	数据寄存器，该域用来存放接收到的数据或存放用于发送到总线的数据。	0

19.12.6 I2C 状态寄存器 1（I2Cx_STAR1）（x=1/2）

偏移地址：0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMBALERT	TIMEOUT	Reserved	PECERR	OVR	AF	ARLO	BERR	TxE	RxNE	Reserved	STOPF	ADD10	BTF	ADDR	SB

位	名称	访问	描述	复位值
15	SMBALERT	RWO	SMBus 警示位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。 在 SMBus 主机模式下： 1：在引脚上产生了 SMBus 警示； 0：无 SMBus 警示。 在 SMBus 从机模式下： 1：收到 SMBAlert 响应地址头序列直到 SMBAlert 变低； 0：没有收到 SMBAlert 响应地址头序列。	0
14	TIMEOUT	RWO	超时或 Tlow 错误标志位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。 1：SCL 处于低已达到 25mS，或主机低电平累计时钟扩招时间超过 10mS，或从设备低电平累计时间超过 25mS； 0：无超时错误。 *注：在从模式下此位被置位，从设备会复位通讯，硬件会释放总线；在主模式下此位被置位，硬件会发出停止条件。*	0
13	Reserved	RO	保留。	0
12	PECERR	RWO	在接收时发生 PEC 错误标志位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。 1：有 PEC 错误，接收到 PEC 后，返回 NAK； 0：无 PEC 错误。	0
11	OVR	RWO	过载、欠载标志位。 1：有过载、欠载事件发生：当 NOSTRETCH=1 时，在接收模式中收到一个新的字节时，数据寄存器里的内容还未被读出，则新接收的字节将丢失；在发送模式时，没有新的数据写入数据寄存器，同样的字节将被发送两次； 0：无过载、欠载事件。	0
10	AF	RWO	应答失败标志位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。	0

			1：应答错误； 0：应答正常。	
9	ARLO	RW0	仲裁丢失标志位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。 1：检测到仲裁丢失，模块失去对总线的控制； 0：仲裁正常。	0
8	BERR	RW0	总线出错标志位，该位可以由用户写 0 复位，或在 PE 变低时由硬件复位。 1：起始或停止条件出错； 0：正常。	0
7	TxE	R0	数据寄存器为空标志位，向数据寄存器写数据可以清除，或产生一个起始或停止位后，或当 PE 为 0 后，由硬件自动清除。 1：发送数据时，发送数据寄存器为空； 0：数据寄存器非空。	0
6	RxNE	R0	数据寄存器非空标志位，对数据寄存器的读写操作将清除此位，或当 PE 为 0 后，由硬件清除此位。 1：接收数据时，数据寄存器不为空； 0：正常。	0
5	Reserved	R0	保留。	0
4	STOPF	R0	停止事件标志位,用户读取状态寄存器 1 之后，对控制寄存器 1 的写操作将会清除该位，或当 PE 为 0 后，由硬件清除此位。 1：在应答之后，从设备在总线上检测到停止事件； 0：没有检测到停止事件。	0
3	ADD10	R0	10 位地址头序列发送标志位，用户读取状态寄存器 1 之后，对控制寄存器 1 的写操作将会清除该位，或当 PE 为 0 后，由硬件清除此位。 1：在 10 位地址模式下，主设备已经将第一个地址字节发送出去； 0：无。	0
2	BTF	R0	字节发送结束标志位，用户读取状态寄存器 1 后，对数据寄存器的读写将清除此位；在传输中，发起一个起始或者停止事件后，或当 PE 为 0 后，由硬件清除此位。 1：字节发送结束。当 NOSTRETCH=0 时：发送时，当一个新数据被发送且数据寄存器还未被写入新数据；接收时，当接收一个新的字节但是数据寄存器还未被读取； 0：无。	0
1	ADDR	RW0	地址被发送/地址匹配标志位，用户读取状态寄存器 1 后，对状态寄存器 2 的读操作将会清除此位，或当 PE 为 0 时，由硬件清除此位。 主模式：	0

			1：地址发送结束：在 10 位地址模式下，当收到地址的第二个字节的 ACK 后改为被置位；在 7 位地址模式下，当收到地址的 ACK 后被置位； 0：地址发送没有结束。 从模式： 1：收到的地址匹配； 0：地址不匹配或没有收到地址。	
0	SB	RO	起始位发送标志位，读取状态寄存器 1 后写数据寄存器的操作将清除该位，或当 PE 为 0 时，硬件将会清除此位。 1：已发送起始位； 0：未发送起始位。	0

19.12.7 I2C 状态寄存器 2（I2Cx_STAR2）（x=1/2）

偏移地址：0x18

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEC[7:0]								DUALF	SMBHOST	SMBDEFAULT	GENCALL	Reserved	TRA	BUSY	MSL

位	名称	访问	描述	复位值
[15:8]	PEC	RO	包错误检查域，当 PEC 使能时（ENPEC 置位），此域存放 PEC 的值。	0
7	DUALF	RO	匹配检测标志位，在产生停止位或起始位时，或在 PE=0 时，硬件会将该位清零。 1：接收到的地址与 OAR2 中的内容相符； 0：接收到的地址与 OAR1 中的内容相符。	0
6	SMBHOST	RO	SMBus 主机头标志位，在产生停止位或起始位时，或在 PE=0 时，硬件会将该位清零。 1：当 SMBTYPE=1 且 ENARP=1 时，收到了 SMBus 主机地址； 0：未接收到 SMBus 主机地址。	0
5	SMBDEFAULT	RO	SMBus 设备默认地址标志位，在产生停止位或起始位时，或在 PE=0 时，硬件会将该位清零。 1：当 ENARP=1，收到 SMBus 设备的默认地址； 0：未收到地址。	0
4	GENCALL	RO	广播呼叫地址标志位，在产生停止位或起始位时，或者在 PE=0 时，硬件会将该位清零。 1：当 ENGC=1 时，收到广播呼叫的地址； 0：未收到广播呼叫地址。	0
3	Reserved	RO	保留。	0
2	TRA	RO	发送/接收标志位，在检测到停止事件（STOPF=1），重复的起始条件、总线仲裁丢失（ARLO=1）或 PE=0 时，硬件会将其清零。 1：数据已发送； 0：接收到数据。	0

			该位根据地址字节的 R/W 位来决定。	
1	BUSY	RO	总线忙标志位，该位在检测到一个停止位时会被清零。在接口被禁用时（PE=0），该信息仍被更新。 1：总线忙：SDA 或 SCL 存在低电平； 0：总线空闲无通讯。	0
0	MSL	RO	主从模式指示位，当接口处于主模式时（SB=1），硬件将该位置位；当总线检测到一个停止位，仲裁丢失时，或 PE=0 时，硬件会清除该位。	0

19.12.8 I2C 时钟寄存器（I2Cx_CKCFGR）（x=1/2）

偏移地址：0x1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F/S	DUTY	Reserved		CCR[11:0]											

位	名称	访问	描述	复位值
15	F/S	RW	主模式选择位。 1：快速模式； 0：标准模式。	0
14	DUTY	RW	快速模式时的高电平时间比上低电平时间的占空比。 1：36%；　　0：33.3%。	0
[13:12]	Reserved	RO	保留。	0
[11:0]	CCR	RW	时钟分频系数域，决定 SCL 时钟的频率波形。	0

19.12.9 I2C 上升时间寄存器（I2Cx_RTR）（x=1/2）

偏移地址：0x20

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

位	名称	访问	描述	复位值
[15:6]	Reserved	RO	保留。	0
[5:0]	TRISE	RW	最大上升时间域。这个位设置主模式的 SCL 的上升时间。最大的上升沿时间等于 TRISE-1 个时钟周期。此位只能在 PE 清零下设置。比如如果 I2C 模块的输入时钟周期为 125nS，而 TRISE 的值为 9，那么最大上升沿时间为（9-1）*125nS，即 1000nS。	000010b

第 20 章 串行外设接口（SPI/I2S）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

SPI 支持以三线同步串行模式进行数据交互，加上片选线支持硬件切换主从模式，支持以单根数据线通讯。

I2S 也是三线的同步船型接口通信协议，它支持四种音频标准，包括飞利浦 I2S 标准、MSB 对齐标准、LSB 对齐标准和 PCM 标准。

20.1 主要特征

20.1.1 SPI 特征

- 支持全双工同步串行模式
- 支持单线半双工模式
- 支持主模式和从模式，多从模式
- 支持 8 位或 16 位数据结构
- 最高时钟频率支持到 Fpclk 的一半
- 数据顺序支持 MSB 或 LSB 在前
- 支持硬件或软件控制 NSS 引脚
- 收发支持硬件 CRC 校验
- 收发缓冲器支持 DMA 传输
- 支持修改时钟相位和极性

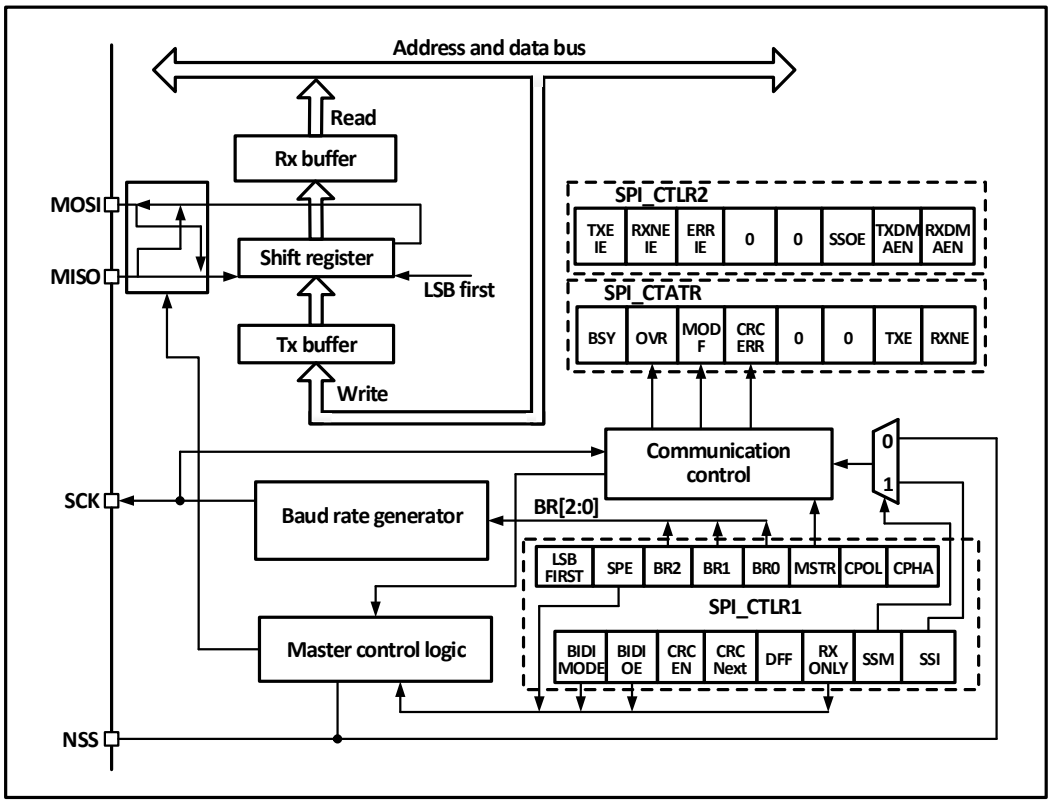
20.1.2 I2S 特征

- 支持单工通信
- 支持主模式和从模式
- 支持 16 位、24 位和 32 位数据格式
- 音频采样频率支持范围 8KHz-96KHz
- 支持时钟极性可编程
- 支持常用 I2S 协议：飞利浦标准、MSB 对齐标准、MSL 对齐标准和 PCM 标准
- 收发缓冲器支持 DMA 传输
- 支持主时钟向外部音频设备输出

20.2 SPI 功能描述

20.2.1 概述

图 20-1 SPI 结构框图



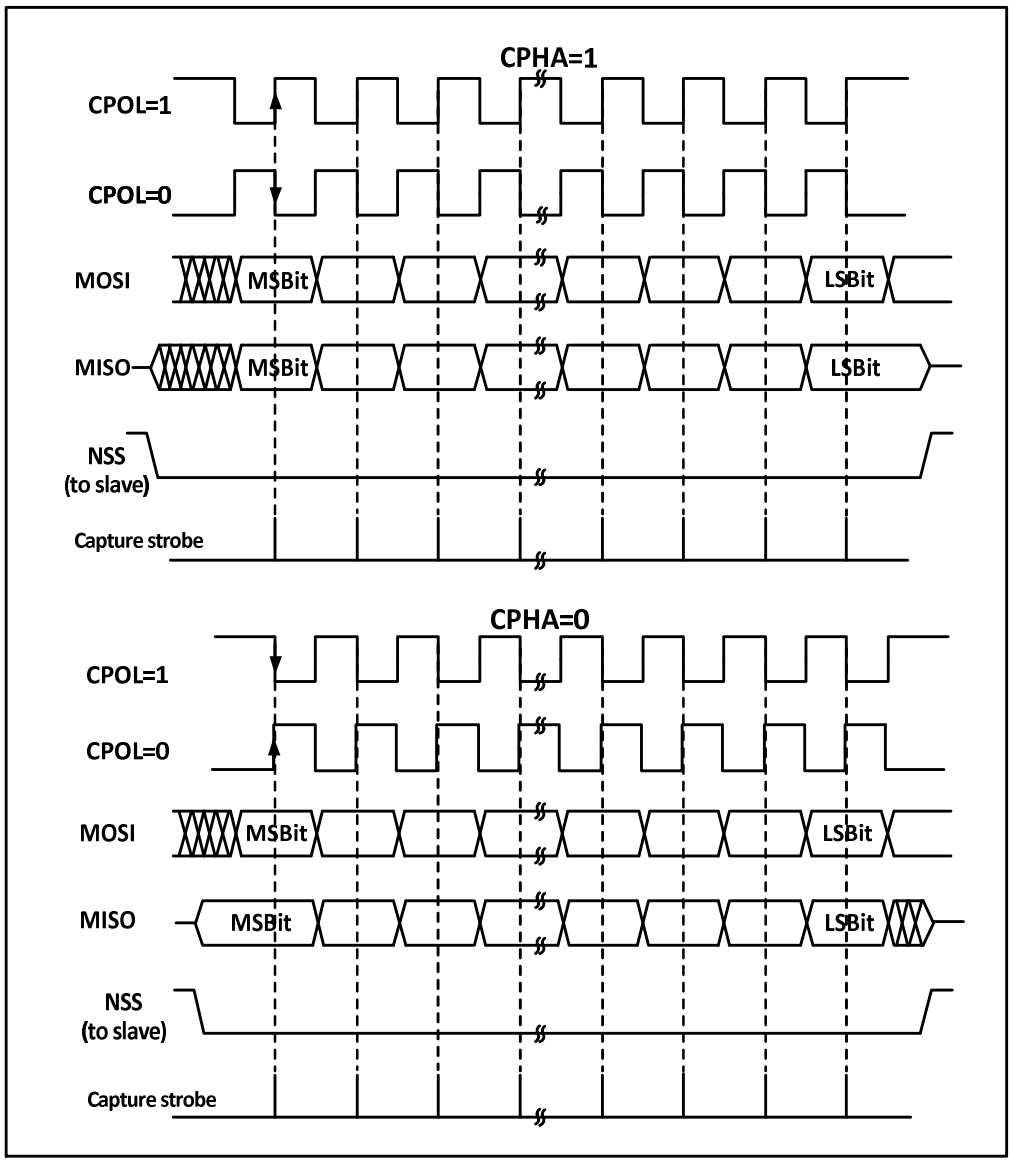
由图 20-1 可以看出，与 SPI 相关的主要是 MISO、MOSI、SCK 和 NSS 四个引脚。其中 MISO 引脚在 SPI 模块工作在主模式下时，是数据输入引脚；工作在从模式下时，是数据输出引脚。MOSI 引脚工作在主模式下时，是数据输出引脚；工作在从模式时，是数据输入引脚。SCK 是时钟引脚，时钟信号一直由主机输出，从机接收时钟信号并同步数据收发。NSS 引脚是片选引脚，有以下用法：

1） NSS 由软件控制：此时 SSM 被置位，内部 NSS 信号由 SSI 决定输出高还是低，这种情况一般用于 SPI 主模式；

2） NSS 由硬件控制：在 NSS 输出使能时，即 SSOE 置位时，在 SPI 主机向外发送输出时会主动拉低 NSS 引脚，如果拉低 NSS 脚，则会产生一个硬件错误；SSOE 不置位，则可以用于多主机模式，如果它被拉低则会强行进入从机模式，MSTR 位会被自动清除。

可以通过 CPHA 和 CPOL 配置 SPI 的工作模式。CPHA 置位表示模块在时钟的第二个边沿进行数据采样，数据被锁存，CPHA 不置位表示 SPI 模块在时钟的第一个边沿进行采样，数据被锁存。CPOL 则表示无数据时时钟保持高电平还是低电平。具体见下图 20-2。

图 20-2 SPI 模式



主机和设备需要设置为相同的 SPI 模式，在配置 SPI 模式前，需要清除 SPE 位。DEF 位可以决定 SP 的单个数据长度是 8 位还是 16 位。LSBFIRST 可以控制单个数据字是高位在前还是低位在前。

20.2.2 主模式

在 SPI 模块工作在主模式时，由 SCK 产生串行时钟。配置成主模式进行以下步骤：

配置控制寄存器的 BR[2:0]域来确定时钟；

配置 CPOL 和 CPHA 位来确定 SPI 模式；

配置 DEF 确定数据字长；

配置 LSBFIRST 确定帧格式；

配置 NSS 引脚，比如置 SSOE 位让硬件去置 NSS。也可以置 SSM 位并把 SSI 位置高；

置 MSTR 位和 SPE 位，需要保证 NSS 此时已经是高。

需要发送数据时只需要向数据寄存器写要发送的数据就行了。SPI 会从发送缓冲区并行地把数据送到移位寄存器，然后按照 LSBFIRST 的设置将数据从移位寄存器发出去，当数据已经到了移位寄存器时，TXE 标志会被置位，如果已经置位了 TXEIE，那么会产生中断。如果 TXE 标志位置位需要向数据寄存器里填数据，维持完整的数据流。

当接收器接收数据时，当数据字的最后一个采样时钟沿到来时，数据从移位寄存器并行地转移到接收缓冲区，RXNE 位被置位，如果之前置位了 RXNEIE 位，还会产生中断。此时应该尽快读取数据寄存器取走数据。

20.2.3 从模式

当 SPI 模块工作在从模式时，SCK 用于接收主机发来的时钟，自身的波特率设置无效。配置成从模式的步骤如下：

配置 DEF 位设置数据位长度；

配置 CPOL 和 CPHA 位匹配主机模式；

配置 LSBFIRST 匹配主机数据帧格式；

硬件管理模式下，NSS 管脚需要保持为低电平，如果设置 NSS 为软件管理（SSM 置位），那么请保持 SSI 不被置位；

清除 MSTR 位，置 SPE 位，开启 SPI 模式。

在发送时，当 SCK 出现第一个从机接收采样沿时，从机开始发送。发送的过程就是发送缓冲区的数据移到发送移位寄存器，当发送缓冲区的数据移到了移位寄存器之后，会置位 TXE 标志，如果之前置位了 TXEIE 位，那么会产生中断。

在接收时，最后一个时钟采样沿之后，RXNE 位被置位，移位寄存器接收到的字节被转移到接收缓冲区，读数据寄存器的读操作可以获得接收缓冲区里的数据。如果在 RXNE 置位之前 RXNEIE 已经被置位，那么会产生中断。

20.2.4 单工模式

SPI 接口可以工作在半双工模式，即主设备使用 MOSI 引脚，从设备使用 MISO 引脚进行通讯。使用半双工通讯时需要把 BIDIMODE 置位，使用 BIDIOE 控制传输方向。

在正常全双工模式下将 RXONLY 位置位可以将 SPI 模块设置为仅仅接收的单工模式，在 RXONLY 置位之后会释放一个数据脚，主模式和从模式释放的引脚并不相同。也可以不理会接收的数据将 SPI 置成只发送的模式。

20.2.5 CRC

SPI 模块使用 CRC 校验保证全双工通信的可靠性，数据收发分别使用单独的 CRC 计算器。CRC 计算的多项式由多项式寄存器决定，对于 8 位数据宽度和 16 位数据宽度，会分别使用不同的计算方法。

设置 CRCEN 位会启用 CRC 校验，同时会使 CRC 计算器复位。在发送完最后一个数据字节后，置 CRCNEXT 位会在当前字节发送结束后发送 TXCRCR 计算器的计算结果，同时最后接收到的接收移位寄存器的值如果与本地算出来的 RXCRCR 的计算值不相符，那么 CRCERR 位会被置位。使用 CRC 校验需要在配置 SPI 工作模式时设置多项式计算器并置 CRCEN 位，并在最后一个字或半字置 CRCNEXT 位发送 CRC 并进行接收 CRC 的校验。注意，收发双方的 CRC 计算多项式应该统一。

20.2.6 DMA

SPI 模块支持使用 DMA 加快数据通讯速度，可以使用 DMA 向发送缓冲区填写数据，或使用 DMA 从接收缓冲区及时取走数据。DMA 会以 RXNE 和 TXE 为信号及时取走或发来数据。DMA 也可以工作在单工或加 CRC 校验的模式。

20.2.7 错误

- 主模式失效错误

当 SPI 工作在 NSS 引脚硬件管理模式下，发生了外部拉低 NSS 引脚的操作；或在 NSS 引脚软件管理模式下，SSI 位被清零；或 SPE 位被清零，导致 SPI 被关闭；或 MSTR 位被清零，SPI 进入从模式。

如果 ERRIE 位已经被置位，还会产生中断。清除 MODF 位步骤：首先执行一次对 R16_SPI1_STATR 的读或写操作，然后写 R16_SPI1_CTLR1。

- 溢出错误

  如果主机发送了数据，而从设备的接收缓冲区中还有未读取的数据，就会发生溢出错误，OVR 位被置位，如果 ERRIE 被置位还会产生中断。发送溢出错误应该重新开始当前传输。读取数据寄存器再读取状态寄存器会消除此位。

- CRC 错误

  当接收到的 CRC 校验字和 RXCRCR 的值不匹配时，会产生 CRC 校验错误，CRCERR 位会被置位。

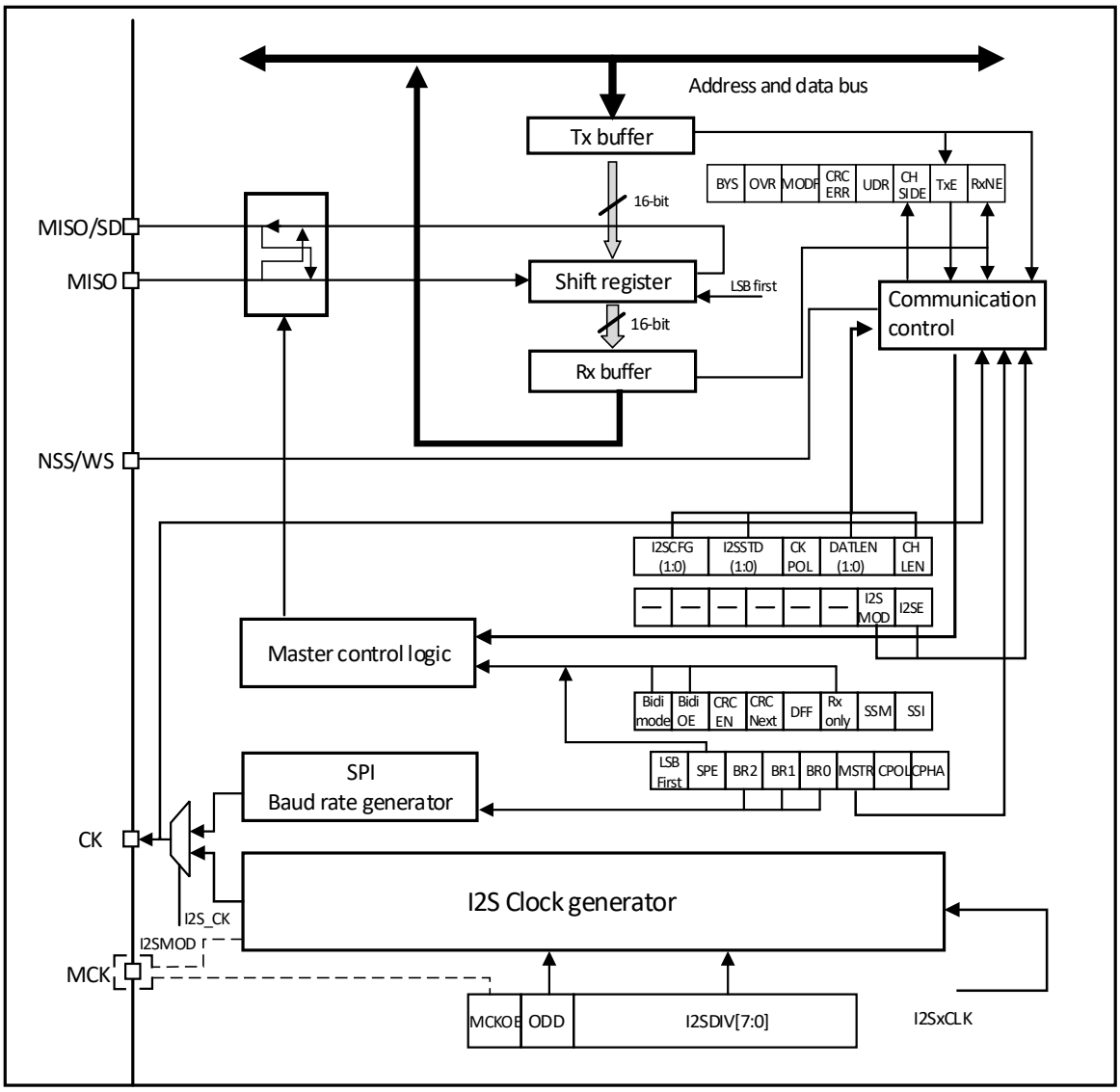
20.2.8 中断

SPI 模块的中断支持五个中断源，其中发送缓冲区空、接收缓冲区非空这两个事件分别会置位 TXE 和 RXNE，在分别置位了 TXEIE 和 RXNEIE 位的情况下会产生中断。除此之外上面提到的三种错误也会产生中断，分别是 MODF、OVR 和 CRCERR，在使能了 ERRIE 位之后，这三种错误也会产生错误中断。

20.3 I2S 功能描述

20.3.1 I2S 概述

图 20-3 I2S 结构框图



通过将寄存器 I2SCFGR 的 I2SMOD 位置位，使能 I2S 功能。此时，可以把 SPI 模块用作 I2S 音频接口。I2S 与 SPI 共用 3 个引脚：

- SD：串行数据(映射至 MOSI 引脚)，用来发送和接收 2 路时分复用通道的数据；
- WS：字选(映射至 NSS 引脚)，主模式下作为数据控制信号输出，从模式下作为输入；
- CK：串行时钟(映射至 SCK 引脚)，主模式下作为时钟信号输出，从模式下作为输入。

在某些外部音频设备需要主时钟时，可以另有一个附加引脚输出时钟：

- MCK：主时钟(独立映射)，在 I2S 配置为主模式，寄存器 I2SPR 的 MCKOE 位为 1 时，作为输出额外的时钟信号引脚使用。输出时钟信号的频率预先设置为 256×Fs，其中 Fs 是音频信号的采样频率。

设置成主模式时，I2S 使用自身的时钟发生器来产生通信用的时钟信号。这个时钟发生器也是主时钟输出的时钟源。I2S 模式下有 2 个额外的寄存器，一个是与时钟发生器配置相关的寄存器 I2SPR，另一个是 I2S 通用配置寄存器 I2SCFGR(可设置音频标准、从/主模式、数据格式、数据包帧、时钟极性等参数)。在 I2S 模式下不使用寄存器 CTLR1 和所有的 CRCR 寄存器。同样，I2S 模式下也不使用寄

存器 CTLR2 的 SSOE 位，和寄存器 STATR 的 MODF 位和 CRCERR 位。I2S 使用与 SPI 相同的寄存器 DATAR 用作 16 位宽模式数据传输。

20.3.2 支持的音频协议

三线总线支持 2 个声道上音频数据的时分复用：左声道和右声道，但是只有一个 16 位寄存器用作发送或接收。因此，软件必须在对数据寄存器写入数据时，根据当前传输中的声道写入相应的数据；同样，在读取寄存器数据时，通过检查寄存器 STATR 的 CHSIDE 位来判断接收到的数据属于哪个声道。左声道总是先于右声道发送数据(CHSIDE 位在 PCM 协议下无意义)。有四种可用的数据和包帧组合。可以通过以下四种数据格式发送数据：

- 16 位数据打包进 16 位帧
- 16 位数据打包进 32 位帧
- 24 位数据打包进 32 位帧
- 32 位数据打包进 32 位帧

在使用 16 位数据扩展到 32 位帧时，前 16 位(MSB)是有意义的数据，后 16 位(LSB)被强制为 0，该操作不需要软件干预，也不需要有 DMA 请求(仅需要一次读或写操作)。24 位和 32 位数据帧需要 CPU 对寄存器 DATAR 进行 2 次读或写操作，在使用 DMA 时，需要 2 次 DMA 传输。对于 24 位数据，扩展到 32 位后，最低 8 位由硬件置 0。对于所有的数据格式和通讯标准，总是先发送最高位(MSB)。I2S 接口支持四种音频标准，可以通过设置寄存器 I2SCFGR 的 I2SSTD[1:0]位和 PCMSYNC 位来选择。

20.3.2.1 I2S 飞利浦标准

在此标准下，引脚 WS 用来指示正在发送的数据属于哪个声道。在发送第一位数据(MSB)前 1 个时钟周期，该引脚即为有效。发送方在时钟信号(CK)的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在时钟信号的下降沿变化。

图 20-4 飞利浦协议波形（16/32 全精度，CPOL=0）

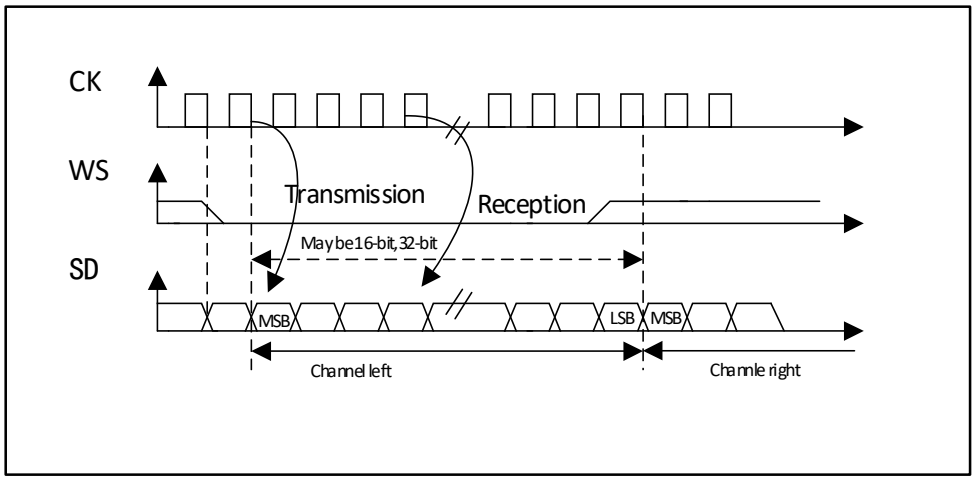
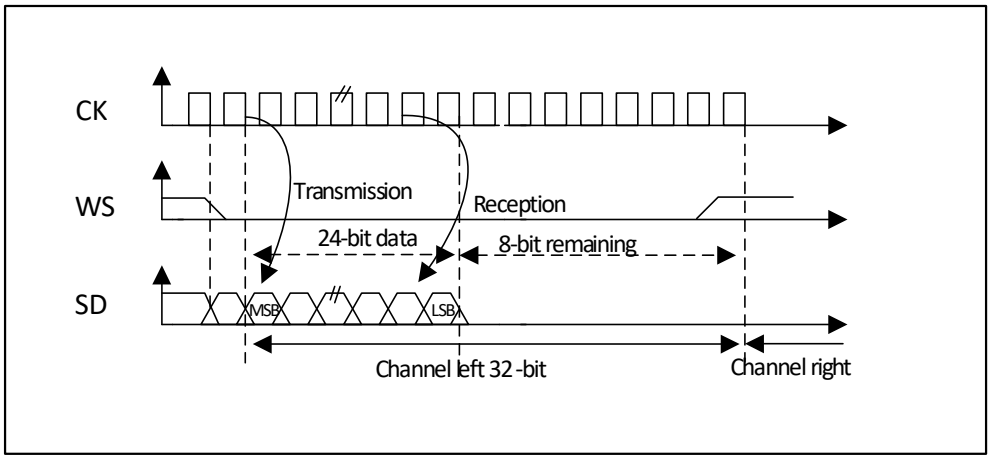
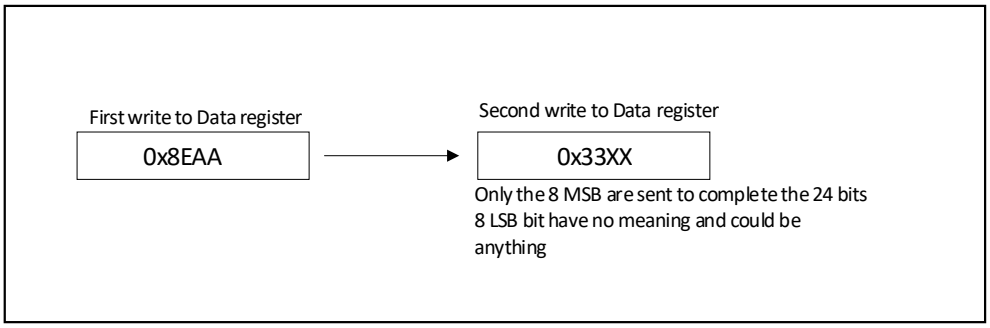


图 20-5 飞利浦协议波形（24 位帧，CPOL=0）



此模式需要对寄存器 SPI_DATAR 进行 2 次读或写操作。在发送模式下：如果需要发送 0x8EAA33（24 位）：



在接收模式下：如果接收 0x8EAA33：

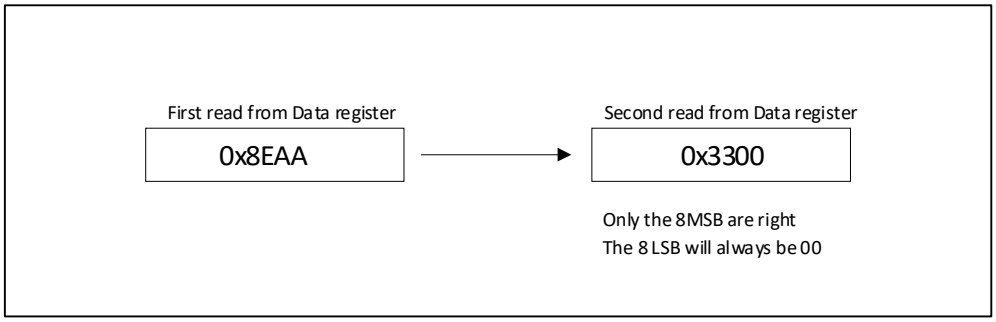
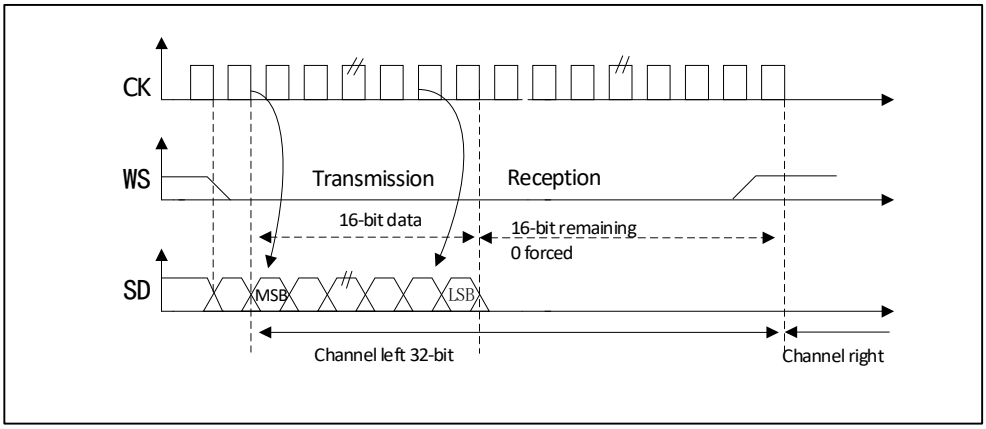


图 20-6 飞利浦协议标准波形（16 位扩展至 32 位包帧，CPOL=0）



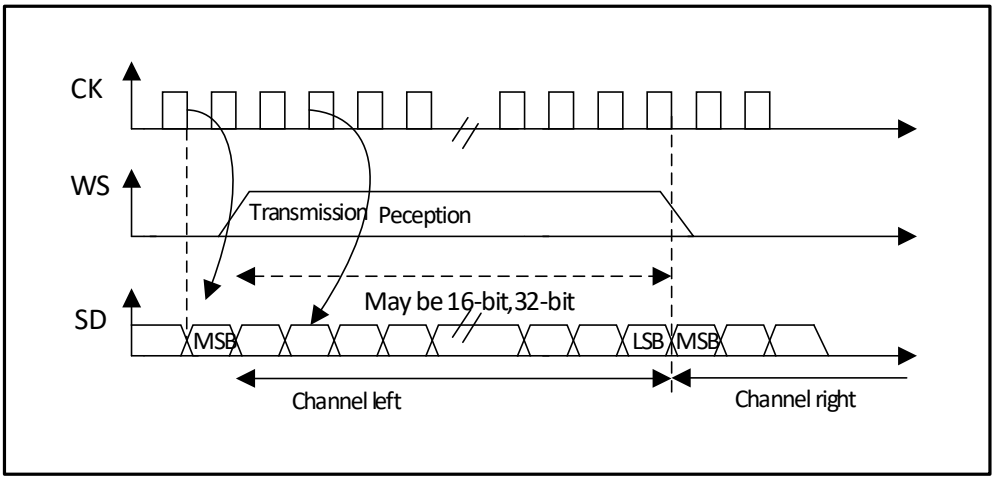
在 I2S 配置阶段，如果选择将 16 位数据扩展到 32 位声道帧，只需要访问一次寄存器 DATAR 用来扩展到 32 位的低 16 位被硬件置为 0x0000。如果待传输或接收的数据是 0x76A3（扩展到 32 位是 0x76A30000），只需要操作一次 DATAR。在发送时需要将 MSB 写入寄存器 DATAR；标志位 TXE 为 1 表示可以写入新的数据，如果允许了相应的中断，则可以产生中断。发送是由硬件完成的，即使还未发送

出后 16 位的 0x0000，也会设置 TXE 并产生相应的中断。接收时，每次收到高 16 位半字(MSB)后，标志位 RXNE 置 1，如果允许了相应的中断，则可以产生中断。这样，在 2 次读和写之间有更多的时间，可以防止下溢或上溢的情况发生。

20.3.2.2 MSB 对齐标准

在此标准下，WS 信号和第一个数据位，即最高位(MSB)同时产生。

图 20-7 MSB 对齐 16 位或 32 位全精度（CPOL = 0）



发送方在时钟信号的下降沿改变数据；接收方是在上升沿读取数据。

图 20-8 MSB 对齐 24 位数据，CPOL = 0

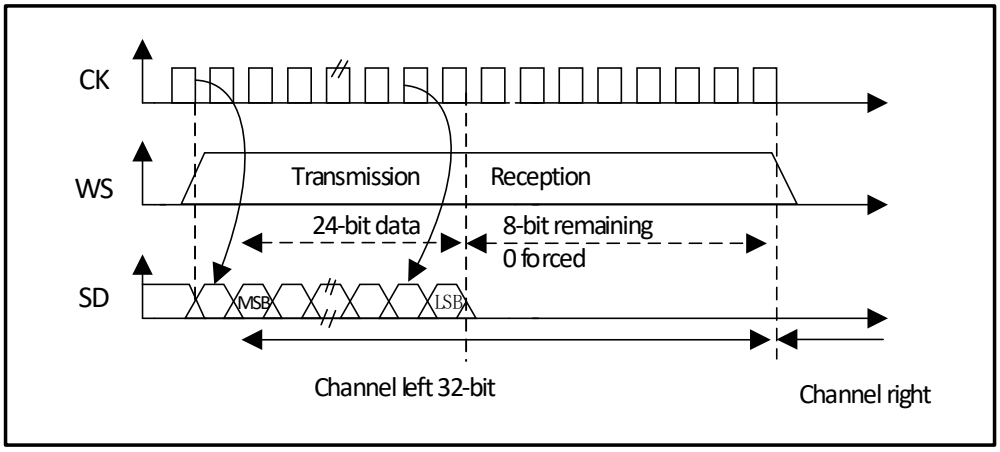
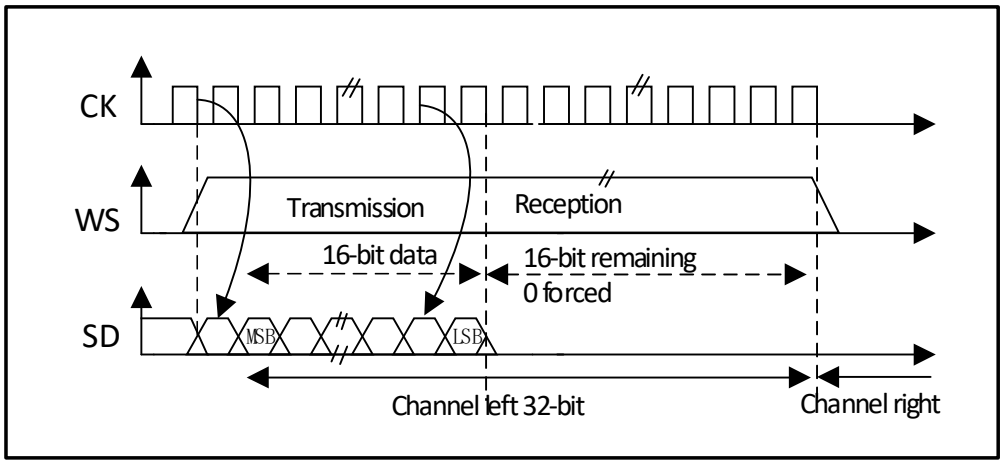


图 20-9 MSB 对齐 16 位数据扩展到 32 位包帧，CPOL = 0



20.3.2.3 LSB 对齐标准

此标准与 MSB 对齐标准类似(在 16 位或 32 位全精度帧格式下无区别)。

图 20-10 LSB 对齐 16 或 32 位全精度，CPOL = 0

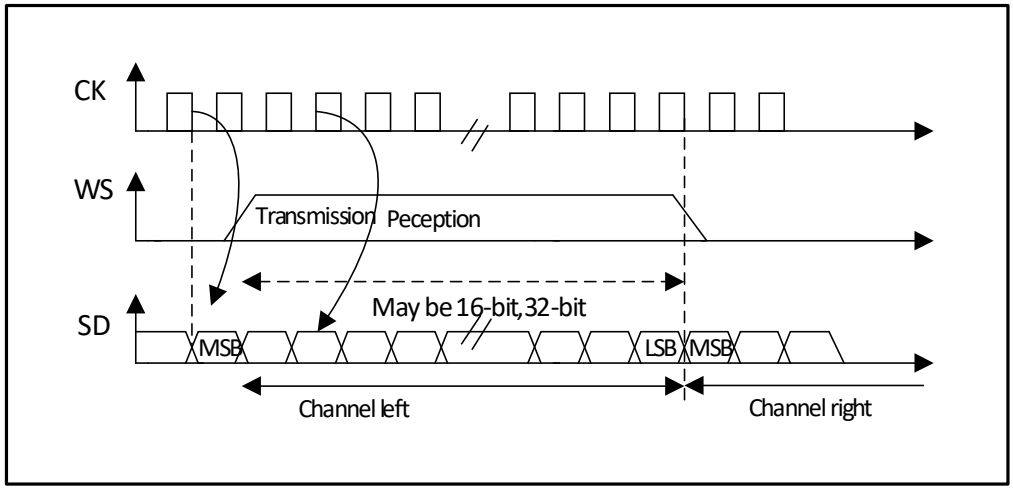
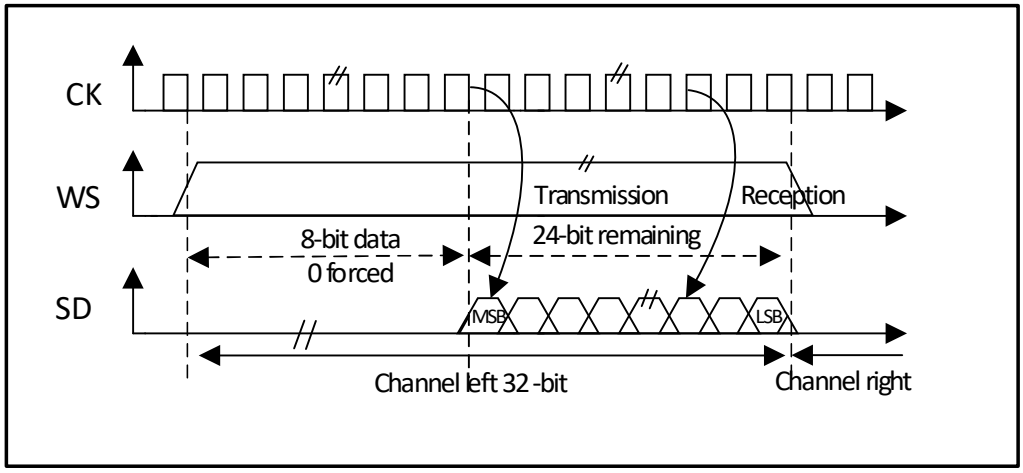
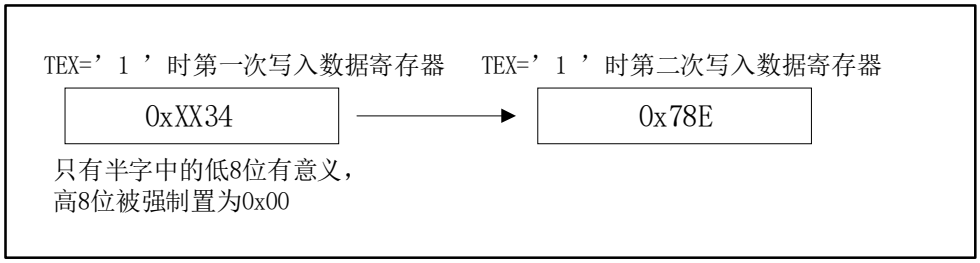


图 20-11 LSB 对齐 24 位数据，CPOL = 0



在发送模式下如果要发送数据 0x3478AE，需要通过软件或 DMA 对寄存器 DATAR 进行 2 次写操作。



在接收模式下如果要接收数据 0x3478AE，需要在 2 个连续的 RXNE 事件发生时，分别对寄存器 DATAR 进行 1 次读操作。

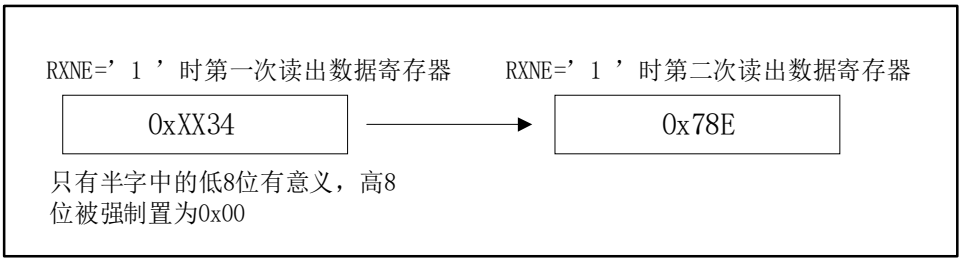
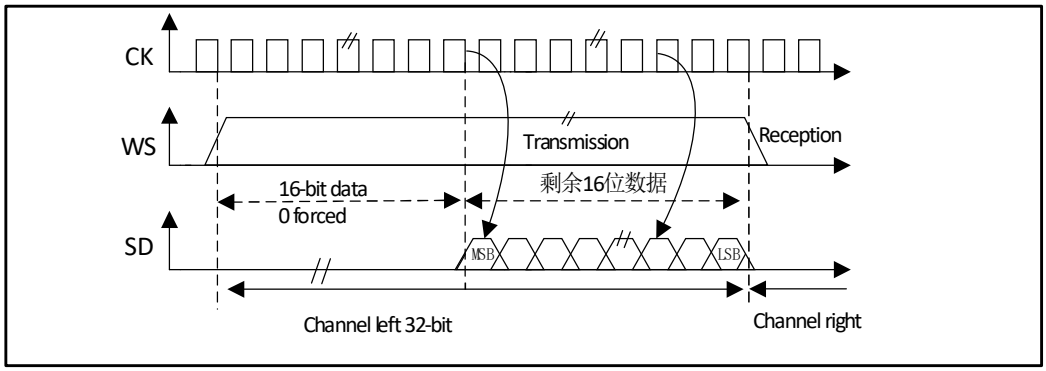


图 20-12 LSB 对齐 16 位数据扩展到 32 位包帧， CPOL = 0



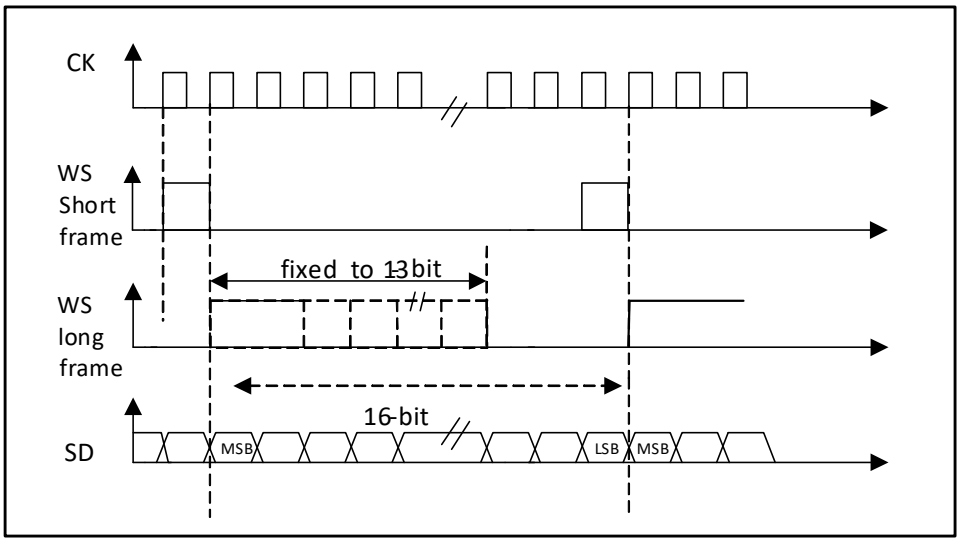
在 I2S 配置阶段，如果选择将 16 位数据扩展到 32 声道帧，只需要访问一次寄存器 DATAR。此时，扩展到 32 位后的高半字(16 位 MSB)被硬件置为 0x0000。

如果待传输或接收的数据是 0x76A3(扩展到 32 位是 0x000076A3)，只需要操作一次 DATAR，在发送时，如果 TXE 为 1，用户需要写入待发送的数据(即 0x76A3)。用来扩展到 32 位的 0x0000 部分由硬件首先发送出去，一旦有效数据开始从 SD 引脚送出，即发生下一次 TXE 事件。在接收时，一旦接收到有效数据(而不是 0x0000 部分)，即发生 RXNE 事件。这样，在 2 次读和写之间有更多的时间，可以防止下溢或上溢的情况发生。

20.3.2.4 PCM 标准

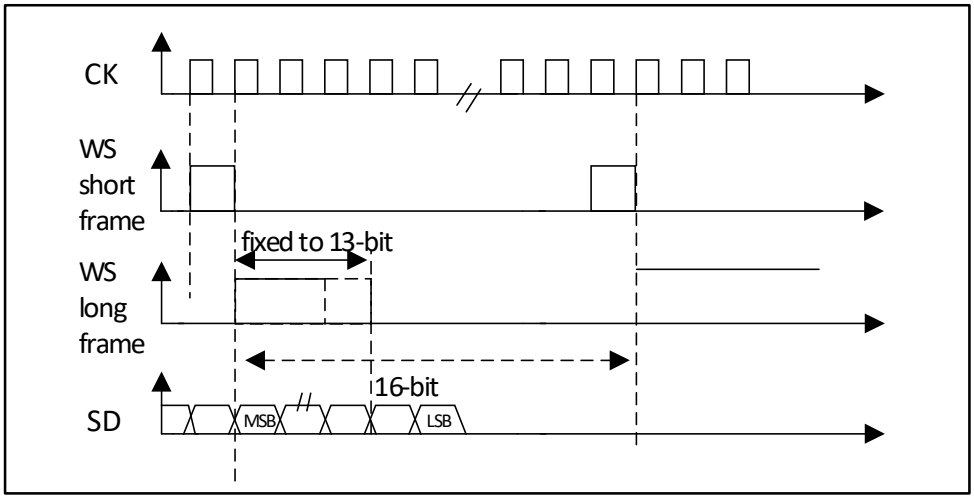
在 PCM 标准下，不存在声道选择的信息。PCM 标准有 2 种可用的帧结构，短帧或长帧，可以通过设置寄存器 I2SCFGR 的 PCMSYNC 位来选择。

图 20-13 PCM 标准波形（16 位）



对于长帧，主模式下，用来同步的 WS 信号有效的时间固定为 13 位。对于短帧，用来同步的 WS 信号长度只有 1 位。

图 20-14 PCM 标准波形（16 位）



无论哪种模式(主或从)、哪种同步方式(短帧或长帧)，连续的 2 帧数据之间和 2 个同步信号之间的时间差，(即使是从模式)需要通过设置 I2SCFGR 寄存器的 DATLEN 位和 CHLEN 位来确定。

20.3.3 时钟发生器

I2S 的比特率即确定了在 I2S 数据线上的数据流和 I2S 的时钟信号频率。I2S 比特率 = 每个声道的比特数×声道数目×音频采样频率。

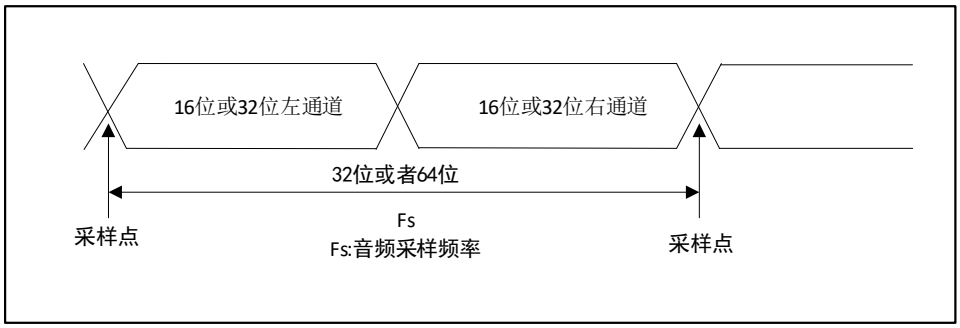
对于一个具有左右声道和 16 位音频信号，I2S 比特率计算如下：

I2S 比特率 = 16 × 2 × Fs

如果包长为 32 位，I2S 比特率计算如下：

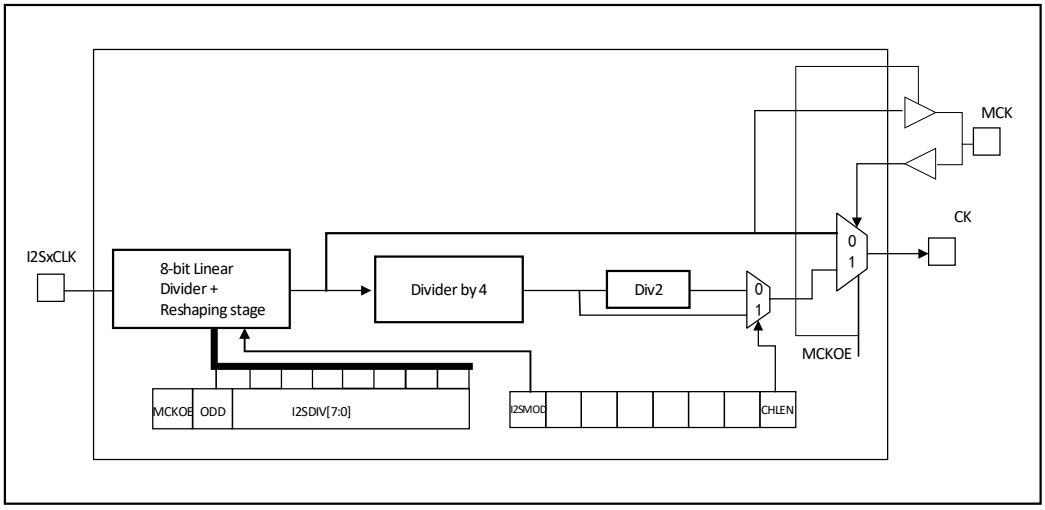
I2S 比特率 = 32 × 2 × Fs

图 20-15 音频采样频率定义



在主模式下，为了获得需要的音频频率，需要正确地对线性分频器进行设置。

图 20-16 I2S 时钟发生器结构



图中 I2SxCLK 的时钟源是系统时钟(即驱动 AHB 时钟的 HSI、HSE 或 PLL)。I2SxCLK 可以来自 SYSCLK，或 PLL3 VCO(2xPLL3CLK)时钟，可以通过 RCC_CFGR2 寄存器的 I2S2SRC 和 I2S3SRC 位选择。音频的采样频率可以是 96kHz、48kHz、44.1kHz、32kHz、22.05kHz、16kHz、11.025kHz 或 8kHz(或任何此范围内的数值)。为了获得需要的频率，需按照以下公式设置线性分频器：

当需要生成主时钟时(寄存器 SPI_I2SPR 的 MCKOE 位为 1)：

声道的帧长为 16 位时， Fs = I2SxCLK / [(16*2) * ((2*I2SDIV) + ODD)*8]

声道的帧长为 32 位时， Fs = I2SxCLK / [(32*2) * ((2*I2SDIV) + ODD)*4]

当关闭主时钟时(MCKOE 位为’0’)：

声道的帧长为 16 位时， Fs = I2SxCLK / [(16*2) * ((2*I2SDIV) + ODD)]

声道的帧长为 32 位时， Fs = I2SxCLK / [(32*2) * ((2*I2SDIV) + ODD)]

20.3.4 I2S 主模式

设置 I2S 工作在主模式，串行时钟由引脚 CK 输出，字选信号由引脚 WS 产生。可以通过设置寄存器 I2SPR 的 MCKOE 位来选择输出或不输出主时钟(MCK)。

20.3.4.1 配置流程

- 设置寄存器 I2SPR 的 I2SDIV[7:0]定义与音频采样频率相符的串行时钟波特率。同时也要定义寄存器 I2SPR 的 ODD 位。
- 设置 CKPOL 位定义通信时钟在空闲时的电平状态。如果需要向外部的 DAC/ADC 音频器件提供主时

钟 MCK，需寄存器 I2SPR 的 MCKOE 位置为。

- 设置寄存器 I2SCFGR 的 I2SMOD 位为 1 激活 I2S 功能，设置 I2SSTD[1:0]和 PCMSYNC 位选择所用的 I2S 标准，设置 CHLEN 选择每个声道的数据位数。还要设置寄存器 SPI_I2SCFGR 的 I2SCFG[1:0]选择 I2S 主模式和方向(发送端还是接收端)。
- 如果需要，可以通过设置寄存器 CR2 来打开所需的中断功能和 DMA 功能。
- 必须将寄存器 I2SCFGR 的 I2SE 位置为 1。
- 引脚 WS 和 CK 需要配置为输出模式。如果寄存器 SPI_I2SPR 的 MCKOE 位为 1，引脚 MCK 也要配置成输出模式。

20.3.4.2 发送流程

当写入 1 个半字(16 位)的数据至发送缓存，发送流程开始。假设第一个写入发送缓存的数据对应的是左声道数据。当数据从发送缓存移到移位寄存器时，标志位 TXE 置 1，这时，要把对应右声道的数据写入发送缓存。标志位 CHSIDE 提示了目前待传输的数据对应哪个声道。标志位 CHSIDE 的值在 TXE 为 1 时更新，因此它在 TXE 为 1 时有意义。在先左声道后右声道的数据都传输完成后，才能被认为是一个完整的数据帧。不可以只传输部分数据帧，如仅有左声道的数据。

当发出第一位数据的同时，半字数据被并行地传送至 16 位移位寄存器，然后后面的位依次按高位在先的顺序从引脚 MOSI/SD 发出。每次数据从发送缓存移至移位寄存器时，标志位 TXE 置为 1，如果寄存器 CR2 的 TXEIE 位为 1，则产生中断。

为了保证连续的音频数据传输，建议在当前传输完成之前，对寄存器 DATAR 写入下一个要传输的数据。建议在要关闭 I2S 功能时，等待标志位 TXE=1 及 BSY=0，再将 I2SE 位清’0’。

20.3.4.3 接收流程

接收流程的配置步骤除了第 3 点外，与发送流程的一致(参见前述的”发送流程”)，需要通过配置 I2SCFG[1:0]来选择主接收模式。无论何种数据和声道长度，音频数据总是以 16 位包的形式接收。即每次填满接收缓存后，标志位 RXNE 置 1，如果寄存器 CR2 的 RXNEIE 位为 1，则产生中断。根据配置的数据和声道长度，收到左声道或右声道的数据会需要 1 次或 2 次把数据传送到接收缓存的过程。对寄存器 DATAR 进行读操作即可清除 RXNE 标志位。每次接收以后即更新 CHSIDE。它的值取决于 I2S 单元产生的 WS 信号。如果前一个接收到的数据还没有被读取，又接收到新数据，即发生上溢，标志位 OVR 被置为 1，如果寄存器 CR2 的 ERRIE 位为 1，则产生中断，表示发生了错误。若要关闭 I2S 功能，需要执行特别的操作，以保证 I2S 模块可以正常地完成传输周期而不会开始新的数据传输。操作过程与数据配置和通道长度、以及音频协议的模式相关：

- 16 位数据扩展到 32 位通道长度(DATLEN=00 并且 CHLEN=1)，使用 LSB(低位)对齐模式(I2SSTD=10)
  a) 等待倒数第二个(n-1)RXNE=1；
  b) 等待 17 个 I2S 时钟周期(使用软件延迟)；
  c) 关闭 I2S(I2SE=0)。
- 16 位数据扩展到 32 位通道长度(DATLEN=00 并且 CHLEN=1)，使用 MSB(高位)对齐、I2S 或 PCM 模式(分别为 I2SSTD=00、I2SSTD=01 或 I2SSTD=11)
  a) 等待最后一个 RXNE=1；
  b) 等待 1 个 I2S 时钟周期(使用软件延迟)；
  c) 关闭 I2S(I2SE=0)。
- 所有其它 DATLEN 和 CHLEN 的组合，I2SSTD 选择的任意音频模式，使用下述方式关闭 I2S：
  a) 等待倒数第二个(n-1)RXNE=1；
  b) 等待一个 I2S 时钟周期(使用软件延迟)；
  c) 关闭 I2S(I2SE=0)。

注： 在传输期间 BSY 标志始终为低。

20.3.5 I2S 从模式

在从模式下，I2S 可以设置成发送和接收模式。从模式的配置方式基本遵循和配置主模式一样的流程。在从模式下，不需要 I2S 接口提供时钟。时钟信号和 WS 信号都由外部主 I2S 设备提供，连接到相应的引脚上。因此用户无需配置时钟。

配置步骤如下：

- 设置寄存器 I2SCFGR 的 I2SMOD 位激活 I2S 功能；设置 I2SSTD[1:0]来选择所用的 I2S 标准；设置 DATLEN[1:0]选择数据的比特数；设置 CHLEN 选择每个声道的数据位数。设置寄存器 I2SCFGR 的 I2SCFG[1:0]选择 I2S 从模式的数据方向(发送端还是接收端)。
- 根据需要，设置寄存器 CR2 打开所需的中断功能和 DMA 功能。
- 必须设置寄存器 I2SCFGR 的 I2SE 位为 1。

20.3.5.1 发送流程

当外部主设备发送时钟信号，并且当 NSS_WS 信号请求传输数据时，发送流程开始。必须先使能从设备，并且写入 I2S 数据寄存器之后，外部主设备才能开始通信。对于 I2S 的 MSB 对齐和 LSB 对齐模式，第一个写入数据寄存器的数据项对应左声道的数据。当开始通信时，数据从发送缓冲器传送到移位寄存器，然后标志位 TXE 置为 1；这时，要把对应右声道的数据项写入 I2S 数据寄存器。标志位 CHSIDE 提示了目前待传输的数据对应哪个声道。与主模式的发送流程相比，在从模式中， CHSIDE 取决于来自外部主 I2S 的 WS 信号。这意味着从 I2S 在接收到主机生成的时钟信号之前，就要准备好第一个要发送的数据。 WS 信号为 1 表示先发送左声道。

20.3.5.2 接收流程

配置步骤除了第 1 点外，与发送流程一致。需要通过配置 I2SCFG[1:0]来选择主接收模式。无论何种数据和声道长度，音频数据总是以 16 位包的形式接收，即每次填满接收缓存，标志位 RXNE 置 1，如果寄存器 I2S_CTLR2 的 RXNEIE 位为 1，则产生中断。按照不同的数据和声道长度设置，收到左声道或右声道数据会需要 1 次或 2 次传输数据至接收缓冲器的过程。每次接收到数据(将要从 DATAR 读出)以后即更新 CHSIDE，它对应 I2S 单元产生的 WS 信号。读取 SPI_DATAR 寄存器，将清除 RXNE 位。在还没有读出前一个接收到的数据，又接收到新数据时，即产生上溢，并设置标志位 OVR 为 1；如果寄存器 I2S_CTLR2 的 ERRIE 位为 1，则产生中断，指示发生了错误。要关闭 I2S 功能时，需要在接收到最后一次 RXNE=1 时将 I2SE 位清 0。

20.3.6 状态标志位

有 3 个状态标志位供用户监控 I2S 总线的状态。

20.3.6.1 状态标志位（BSY）

BSY 标志由硬件设置与清除(写入此位无效果)，该标志位指示 I2S 通信层的状态。该位为 1 时表明 I2S 通讯正在进行中，但有一个例外：主接收模式(I2SCFG=11)下，在接收期间 BSY 标志始终为低。在软件要关闭 SPI 模块之前，可以使用 BSY 标志检测传输是否结束，这样可以避免破坏最后一次传输，因此需要严格按照下述过程执行。当传输开始时， BSY 标志被置为 1，除非 I2S 模块处于主接收模式。当传输结束时或者当关闭 I2S 模块时，该标志位被清除。当通信是连续的时候在主发送模式时，整个传输期间， BSY 标志始终为高；在从模式时，每个数据项传输之间， BSY 标志在 1 个 I2S 时钟周期内变低。

20.3.6.2 发送缓冲标志位（TXE）

该标志位为 1 表示发送缓冲器为空，可以对发送缓冲器写入新的待发送数据。在发送缓冲器中已有数据时，标志位清 0。在 I2S 被关闭时(I2SE 位为 0)，该标志位也为 0。

20.3.6.3 接收缓存非空标志位（RXNE）

该标志位置 1 表示在接收缓存里有接收到的有效数据。在读取寄存器 DATAR 时，该位清 0。

20.3.6.4 声道标志位（CHSIDE）

在发送模式下，该标志位在 TXE 为高时刷新，指示从 SD 引脚上发送的数据所在的声道。如果在从发送模式下发生了下溢错误，该标志位的值无效，在重新开始通讯前需要把 I2S 关闭再打开。在接收模式下，该标志位在寄存器 DATAR 接收到数据时刷新，指示接收到的数据所在的声道。注意，如果发生错误，该标志位无意义，需要将 I2S 关闭再打开。在 PCM 标准下，无论短帧格式还是长帧格式，这个标志位都没有意义。如果寄存器 STATR 的标志位 OVR 或 UDR 为 1，且寄存器 CR2 的 ERRIE 位为 1，则会产生中断。(中断源已经被清除后)可以通过读寄存器 STATR 来清除中断标志。

20.3.7 错误标志位

20.3.7.1 下溢标志位（UDR）

在从发送模式下，如果数据传输的第一个时钟边沿到达时，新的数据仍然没有写入 DATAR 寄存器，该标志位会被置 1。在寄存器 I2SCFGR 的 I2SMOD 位置 1 后，该标志位才有效。如果寄存器 CR2 的 ERRIE 位为 1，就会产生中断。通过对寄存器 STATR 进行读操作来清除该标志位。

20.3.7.2 上溢标志位（OVR）

如果还没有读出前一个接收到的数据时，又接收到新的数据，即产生上溢，该标志位置 1，如果寄存器 CTLR2 的 ERRIE 位为 1，则产生中断指示发生了错误。这时，接收缓存的内容，不会刷新为从发送设备送来的新数据。对寄存器 DATAR 的读操作返回最后一个正确接收到的数据。其他所有在上溢发生后由发送设备发出的 16 位数据都会丢失。通过先读寄存器 DATAR 再读寄存器 STATR，来清除该标志位。

20.3.8 I2S 中断

I2S 有 4 个中断源，其中发送缓冲区空，接收缓冲区非空这两个事件分别会置位 TXE 和 RXNE，在分别置位了 TXEIE 和 RXNEIE 位的情况下会产生中断。如果还没有读出前一个接收到的数据时，又接收到新的数据，即产生上溢，如果置位了 ERRIE 会产生上溢中断；在从发送模式下，如果数据传输的第一个时钟边沿到达时，新的数据仍然没有写入 DATAR 寄存器如果置位了 ERRIE 会产生下溢中断。

20.3.9 DMA 功能

DMA 的工作方式在 I2S 模式除了 CRC 功能不可用以外，与在 SPI 模式完全相同。因为在 I2S 模式下没有数据传输保护系统。

20.4 寄存器描述

表 20-1 SPI1 相关寄存器列表

名称	访问地址	描述	复位值
R16_SPI1_CTLR1	0x40013000	SPI1 控制寄存器 1	0x0000
R16_SPI1_CTLR2	0x40013004	SPI1 控制寄存器 2	0x0000
R16_SPI1_STATR	0x40013008	SPI1 状态寄存器	0x0002
R16_SPI1_DATAR	0x4001300C	SPI1 数据寄存器	0x0000

R16_SPI1_CRCR	0x40013010	SPI1 多项式寄存器	0x0007
R16_SPI1_RCRCR	0x40013014	SPI1 接收 CRC 寄存器	0x0000
R16_SPI1_TCRCR	0x40013018	SPI1 发送 CRC 寄存器	0x0000
R16_SPI1_HSCR	0x4001301C	SPI1 高速控制寄存器	0x0000

表 20-2 SPI2 相关寄存器列表

名称	访问地址	描述	复位值
R16_SPI2_CTLR1	0x40003800	SPI2 控制寄存器 1	0x0000
R16_SPI2_CTLR2	0x40003804	SPI2 控制寄存器 2	0x0000
R16_SPI2_STATR	0x40003808	SPI2 状态寄存器	0x0002
R16_SPI2_DATAR	0x4000380C	SPI2 数据寄存器	0x0000
R16_SPI2_CRCR	0x40003810	SPI2 多项式寄存器	0x0007
R16_SPI2_RCRCR	0x40003814	SPI2 接收 CRC 寄存器	0x0000
R16_SPI2_TCRCR	0x40003818	SPI2 发送 CRC 寄存器	0x0000
R16_SPI2_HSCR	0x4000381C	SPI2 高速控制寄存器	0x0000

表 20-3 SPI3 相关寄存器列表

名称	访问地址	描述	复位值
R16_SPI3_CTLR1	0x40003C00	SPI3 控制寄存器 1	0x0000
R16_SPI3_CTLR2	0x40003C04	SPI3 控制寄存器 2	0x0000
R16_SPI3_STATR	0x40003C08	SPI3 状态寄存器	0x0002
R16_SPI3_DATAR	0x40003C0C	SPI3 数据寄存器	0x0000
R16_SPI3_CRCR	0x40003C10	SPI3 多项式寄存器	0x0007
R16_SPI3_RCRCR	0x40003C14	SPI3 接收 CRC 寄存器	0x0000
R16_SPI3_TCRCR	0x40003C18	SPI3 发送 CRC 寄存器	0x0000
R16_SPI3_HSCR	0x40003C1C	SPI3 高速控制寄存器	0x0000

20.4.1 SPI 控制寄存器 1（SPIx_CTLR1）（x=1/2/3）

偏移地址：0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRCEN	CRC NEXT	DFF	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA

位	名称	访问	描述	复位值
15	BIDIMODE	RW	单向数据模式使能位。 1：选择单线双向模式； 0：选择双线双向模式。	0
14	BIDIOE	RW	单线输出使能位，和 BIDIMODE 配合使用。 1：使能输出，仅发送； 0：禁止输出，仅接收。	0
13	CRCEN	RW	硬件 CRC 校验使能位，该位只能在 SPE 为 0 时写入，该位只能在全双工模式下使用。 1：启动 CRC 计算； 0：禁止 CRC 计算。	0
12	CRCNEXT	RW	在接下来的一次数据传输后，发送 CRC 寄存器	0

			的值。这位应该在向数据寄存器写入最后一个数据后立刻置位。 1：发送 CRC 校验结果； 0：继续发送数据寄存器的数据。	
11	DFF	RW	数据帧长度位，此位只能在 SPE 为 0 时写入。 1：使用 16 位数据长度进行收发； 0：使用 8 位数据长度进行收发。	0
10	RXONLY	RW	双线模式下只接收位，该位和 BIDIMODE 配合使用。置此位可以让设备只接收不发送。 1：只接收，单工模式； 0：全双工模式。	0
9	SSM	RW	片选引脚管理位，此位决定 NSS 引脚的电平由硬件还是软件控制。 1：软件控制 NSS 引脚； 0：硬件控制 NSS 引脚。	0
8	SSI	RW	片选引脚控制位，在 SSM 置位的情况下，此位决定 NSS 引脚的电平。 1：NSS 为高电平； 0：NSS 为低电平。	0
7	LSBFIRST	RW	帧格式控制位。不可以在通讯时修改此位。 1：先发送 LSB； 0：先发送 MSB。	0
6	SPE	RW	SPI 使能位。 1：启用 SPI； 0：禁用 SPI。	0
[5:3]	BR	RW	波特率设置域，在通讯时不可以修改此域。 000：$F_{PCLK}/2$；　001：$F_{PCLK}/4$； 010：$F_{PCLK}/8$；　011：$F_{PCLK}/16$； 100：$F_{PCLK}/32$；　101：$F_{PCLK}/64$； 110：$F_{PCLK}/128$；　111：$F_{PCLK}/256$。	0
2	MSTR	RW	主从设置位，在通讯时不可以修改此位。 1：配置为主设备； 0：配置为从设备。	0b
1	CPOL	RW	时钟极性选择位，在通讯时不可以修改此位。 1：空闲状态时，SCK 保持高电平； 0：空闲状态时，SCK 保持低电平。	0
0	CPHA	RW	时钟相位设置位，在通讯时不可以修改此位。 1：数据采样从第二个时钟沿开始； 0：数据采样从第一个时钟沿开始。	0

20.4.2 SPI 控制寄存器 2（SPIx_CTLR2）（x=1/2/3）

偏移地址：0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXEIE	RXNEIE	ERRIE	Reserved		SSOE	TXDMAEN	RXDMAEN

控制寄存器 2

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
7	TXEIE	RW	发送缓冲区空中断使能位。置此位允许 TXE 被置位时产生中断。	0
6	RXNEIE	RW	接收缓冲区非空中断使能位。置此位允许 RXNE 被置位时产生中断。	0
5	ERRIE	RW	错误中断使能位。置此位允许在产生错误（CRCERR，OVR，MODF）时产生中断。	0
[4:3]	Reserved	RO	保留。	0
2	SSOE	RW	SS 输出使能。禁止 SS 输出可以工作在多主模式下。 1：使能 SS 输出； 0：禁止主模式下的 SS 输出。	0
1	TXDMAEN	RW	发送缓冲区 DMA 使能位。 1：启用发送缓冲区 DMA； 0：禁用发送缓冲区 DMA。	0
0	RXDMAEN	RW	接收缓冲区 DMA 使能位。 1：启用接收缓冲区 DMA； 0：禁用接收缓冲区 DMA。	0

20.4.3 SPI 状态寄存器（SPIx_STATR）（x=1/2/3）

偏移地址：0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BSY	OVR	MODF	CRC ERR	Reserved		TXE	RXNE

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
7	BSY	RO	忙标志位，该位由硬件置位或复位。 1：SPI 正在通讯，或发送缓冲区非空； 0：SPI 不在通讯。	0
6	OVR	RWO	溢出标志位，该位由硬件置位，软件复位。 1：出现溢出错误； 0：没有出现溢出错误。	0
5	MODF	RO	模式错误标志位，该位由硬件置位，软件复位。 1：出现了模式错误； 0：没有出现模式错误。	0
4	CRCERR	RWO	CRC 错误标志位，该位由硬件置位，软件复位。 1：收到的 CRC 值与 RCRCR 的值不一致； 0：收到的 CRC 值与 RCRCR 的值一致。	0
3	UDR	RO	下溢标志位，该位由硬件置位，软件复位。 1：发生下溢； 0：未发生下溢。	0

2	CRCERR	RO	声道，该位由硬件置位，软件复位。 1：需要传输或接收左声道； 0：需要传输或接收右声道。	0
1	TXE	RO	发送缓冲区为空标志位。 1：发送缓冲区为空； 0：发送缓冲区非空。	1
0	RXNE	RO	接收缓冲区非空标志位。 1：接收缓冲区非空； 0：接收缓冲区为空。	0

20.4.4 SPI 数据寄存器（SPIx_DATAR）（x=1/2/3）

偏移地址：0x0C

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

位	名称	访问	描述	复位值
[15:0]	DATAR	RW	数据寄存器。数据寄存器用于存放接收到的数据或预存将要发送出去的数据，因此数据寄存器的读写实际上是对应操作不同的区域，其中读对用接 收缓冲区，写对应发送缓冲区。数据的接收和发送可以是 8 位或者 16 位的，需要在传输之前就确定使用多少位的数据。使用 8 位进行数据传输时，只有数据寄存器的低 8 位被使用，接收时高 8 位强制为 0。使用 16 位数据结构则会使全部 16 位数据寄存器被使用。	0

20.4.5 SPI 多项式寄存器（SPIx_CRCR）（x=1/2/3）

偏移地址：0x10

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

位	名称	访问	描述	复位值
[15:0]	CRCPOLY	RW	CRC 多项式。此域定义 CRC 计算用到的多项式。	7

20.4.6 SPI 接收 CRC 寄存器（SPIx_RCRCR）（x=1/2/3）

偏移地址：0x14

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

位	名称	访问	描述	复位值
[15:0]	RXCRC	RO	接收 CRC 值。存储着计算出来的接收到的字节的 CRC 校验的结果。对 CRCEN 置位会复位该寄	0

			存器。计算方法使用 CRCPOLY 用到的多项式。8 位模式下只有低 8 位参与计算，16 位模式下全部 16 位都会参与计算。需要在 BSY 为 0 时去读取这个寄存器。	

20.4.7 发送 CRC 寄存器（SPIx_TCRCR）（x=1/2/3）

偏移地址：0x18

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

位	名称	访问	描述	复位值
[15:0]	TXCRC	RO	发送 CRC 值。存储着计算出来的已经发送出去的字节的 CRC 校验的结果。对 CRCEN 置位会复位该寄存器。计算方法使用 CRCPOLY 用到的多项式。8 位模式下只有低 8 位参与计算，16 位模式下全部 16 位都会参与计算。需要在 BSY 为 0 时去读取这个寄存器。	0

20.4.8 SPI_I2S 配置寄存器（SPI_I2S_CFGR）（x=1/2/3）

偏移地址：0x1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				I2S MOD	I2SE	I2SCFG		PCM SYNC	保留	I2SSTD		CKPOL	DATLEN		CHLEN

位	名称	访问	描述
[15:13]	Reserved	RO	保留
11	I2SMOD	RW	I2S 模式选择，该位只有在关闭了 SPI 或者 I2S 时才能设置。 1：选择 I2S 模式； 0：选择 SPI 模式。
10	I2SE	RW	I2S 使能，在 SPI 模式下不使用。 1：I2S 使能； 0：关闭 I2S。
[9:8]	I2SCFG	RW	I2S 模式选择，此该位只有在关闭了 I2S 时才能设置： 00：从设备发送； 01：从设备接收； 10：主设备发送； 11：主设备接受。
7	PCMSYNC	RW	PCM 帧同步。该位只在 I2SSTD = 11（使用 PCM 标准）时有意义。 1：长帧同步； 0：短帧同步。
6	Reserved	RO	保留
[5:4]	I2SSTD	RW	I2S 标准选择，只有在关闭了 I2S 时才能设置该位。

			00：I2S 飞利浦标准； 01：高字节对齐标准（左对齐）； 10：低字节对齐标准(右对齐)； 11：PCM 标准。
3	CKPOL	RW	静止态时钟极性，为了正确操作，该位只有在关闭了 I2S 时才能设置。 1：I2S 时钟静止态为高电平； 0：I2S 时钟静止态为低电平。
[2:1]	DATLEN	RW	待传输数据长度，为了正确操作，该位只有在关闭了 I2S 时才能设置。 00：16 位数据长度； 01：24 位数据长度； 10：32 位数据长度； 11：不允许。
0	CHLEN	RW	声道长度，只有在 DATLEN = 00 时该位的写操作才有意义，否则声道长度都由硬件固定为 32 位。。 1：32 位宽； 0：16 位宽。

20.4.9 SPI_I2S 预分频寄存器（SPIx_I2SPR）（x=2/3）

偏移地址：0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						MCKOE	ODD	I2SDIV							

位	名称	访问	描述
[15:10]	Reserved	RO	保留
9	MCKOE	RW	主设备时钟输出使能，为了正确操作，该位只有在关闭了 I2S 时才能设置。仅在 I2S 主设备模式下使用该位。 1：主设备时钟输出使能； 0：关闭主设备时钟输出。
8	I2SCFG	RW	奇系数预分频，为了正确操作，该位只有在关闭了 I2S 时才能设置。仅在 I2S 主设备模式下使用该位。 1：实际分频系数 = (I2SDIV * 2)+1； 0：实际分频系数 = I2SDIV *2。
[7:0]	I2SDIV	RW	I2S 线性预分频。为了正确操作，该位只有在关闭了 I2S 时才能设置。仅在 I2S 主设备模式下使用该位。禁止设置 I2SDIV [7:0] = 0 或者 I2SDIV [7:0] = 1 参见 19.3.3 节。

20.4.10 SPI 高速控制寄存器（SPIx_HSCR）（x=1/2/3）

偏移地址：0x20

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

位	名称	访问	描述
[15:1]	Reserved	RO	保留
0	HSRXEN	WO	SPI 高速模式下（CLK 大于等于 36MHz）读使能。该模式仅在时钟 2 分频（即 CTLR1 寄存器的 BR = 000）时有效。该位不可读。 1：使能高速读模式； 0：关闭高速读模式。

第 21 章 USB 全速设备控制器（USBD）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器系列部分产品。

USBD 模块是基于 USB2.0 全速设备技术规范，设计的 USB 全速、低速协议通讯控制器。内置硬件自动处理物理信号的反向不归零（NRZI）编码/解码、位填充。控制可驱动出 USB 总线多种状态、协议包收发，并提供自动应答进行流控保证应用程序处理时间等功能。

21.1 主要特性

- 符合 USB2.0 全速设备技术规范
- 支持 USB 全速 12Mbps、低速 1.5Mbps 模式
- 支持配置 16 个传输通道
- 支持端点地址范围 0-15
- 支持控制、中断、批量、同步传输
- 支持批量/同步端点的双缓冲机制
- USB 挂起、唤醒、恢复操作
- 硬件自动进行数据 PID 翻转、传输流控
- 帧锁定时钟脉冲生成

注：USBD 和 CAN 控制器在设计中共享了一个专用的 512 字节 SRAM 区域用于数据的发送和接收，因此同时使用 USBD 和 CAN 功能时，需要合理分配此共享区域，防止出现数据冲突。

21.2 功能描述

21.2.1 功能介绍

USBD 模块为 USB 主机（一般是 PC）和微控制器之间的数据通讯提供了一条符合 USB 规范的通信连接，使用时由应用程序和模块硬件配合完成。模块中包含一块共享的 512 字节专用 SRAM 区域作为 USB 收发数据缓冲区，由配置的端点数目和每个端点最大数据包长度决定实际使用范围。每个端点最大可使用 1023 字节缓冲区，最多可用于 16 个单向或 8 个双向端点。

USBD 模块功能包括：

- 物理信号编码/解码：根据 USB 规范实现令牌包、数据包、握手包的 PID 检测，包括位填充、CRC 的生成和校验、帧头同步识别等。
- 事务处理：判断正确传输和错误状态，提供各自标志状态及中断通知。
- 总线挂起/复位/唤醒状态识别通知。
- 自动数据包 PID：根据协议，对非同步端点、同步端点的收发数据包 PID 进行硬件翻转或锁定，减少应用程序工作。
- 自动应答包 PID：根据协议，完成一次 USB 事务后，对非同步端点会自动修改应答包状态来为应用程序提供足够的处理和准备时间，但不影响 USB 总线上的物理收发。
- 管理数据收发：定位端点配置及缓冲区描述区域，检测缓冲区边界防溢出。单缓冲/双缓冲区域管理、按端点类型中断上报优先级管理等。
- 提供通用类、端点类、缓冲区描述类寄存器配置。

应用程序可以：

- 获取基于 USB 协议的帧间隔时间点，总线状态：挂起、复位。
- 自定义端点数目、端点类型、端点大小。自定义传输数据缓冲区域。
- 获取当前或已挂起端点的服务进行处理。
- 获取如位填充、格式、CRC、协议、缺失 ACK、缓冲区溢出/缓冲区未满等错误状态。

- 驱动模块进入低功耗模式。

USBD 模块提 USB 事件映射到 3 个不同的 NVIC 或 PFIC 请求线上（使用了 3 个中断号）：

1） USB 高优先级中断：仅能由同步和双缓冲批量传输的正确传输事件触发，目的是保证最大的传输速率。
2） USB 低优先级中断：可由所有 USB 事件触发(正确传输，USB 复位等)。固件在处理中断前应当首先确定中断源。
3） USB 唤醒中断：由 USB 挂起模式的唤醒事件触发。

21.2.2 功能配置

21.2.2.1 GPIO 端口

一旦使能了 USBD 模块，作为 UDP 和 UPM 的 GPIO 口会自动连接到内部 USB 收发器，而断开其 GPIO 外设的端口设置。所以推荐 GPIO 口配置为推挽方式输出低电平，防止在未开启 USBD 功能前，出现端口不确定状态或连接 PC 主机时，提前通知有 USB 设备接入。

USBD 模块内置 USB 设备模式的 1.5K 上拉电阻，无需外接上拉电阻。具体配置请参考配置扩展控制寄存器（EXTEND_CTR）说明。

21.2.2.2 模块初始化

首先，USB 收发器相关的模拟部分需要标准的 48MHz 时钟作为基准时钟，此时钟来源于 AHB 总线。应用程序需要先通过配置时钟管理逻辑的相应控制位(RCC_CFGR0 寄存器)保证当前 USB 时钟是 48MHz，再使能 USB 接口时钟，使程序可以访问 USBD 模块的寄存器。

其次，在模块强制复位时（USBD_CNTR 寄存器上的 FRES 位默认为 1），应用程序应该初始化所需要的寄存器和分组缓冲区描述表。包括：分组缓冲区描述表地址寄存器（USBD_BTABLE）、端点配置寄存器（USBD_EPRx）和分组缓冲区描述表寄存器。配置 USBD_DADDR 寄存器 ADD[6:0]域为 0（USB 协议默认地址），置位 EF 位使能端点传输功能。

最后，启用内部 1.5K 上拉电阻和设置速度模式（EXTEND_CTR 寄存器），然后，清除 USBD_CNTR 寄存器上的 FRES 位，撤销 USBD 模块强制复位状态来使能 USBD 模块，清除 USBD_ISTR 寄存器的各种状态标志，以便在使能其他任何单元的操作之前清除未处理的假中断标志。开启 USBD_CNTR 寄存器中需要的中断控制位。

21.2.2.3 USB 复位

USB 复位包括：USBD 模块强制复位和 USB 总线复位（协议复位）。两者皆会产生 USBD_ISTR 寄存器的 RESET 标志。发生 USB 复位时，所有端点的通信都被禁止(USBD 模块不会响应任何包传输)。在 USB 复位后，USBD 模块被使能，同时 USB 端点也需要被使能以便可以响应 USB 主机（USB_DADDR 寄存器的 EF 位为 1）。在 USB 设备的枚举阶段，主机将分配给设备一个唯一的地址，这个地址必须写入 USB_DADDR 寄存器的 ADD[6:0]位中。

注：RESET 标志来源 USBD 模块强制复位控制位（FRES）的状态和 USB 总线复位信号起始。

21.2.2.4 端点配置及缓冲区描述表

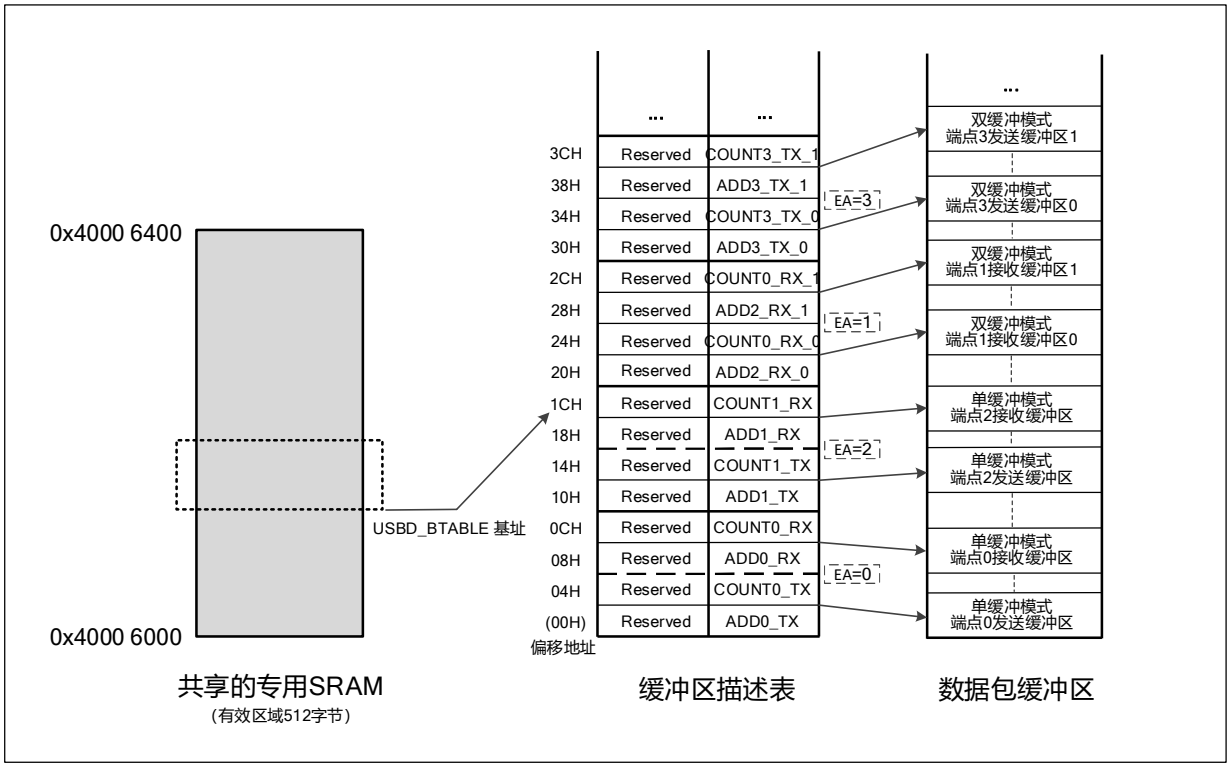
每个端点配置寄存器可以配置一个双向端点单缓冲属性，也可以配置一个单向端点双缓冲属性。例如：配置双向端点单缓冲属性，端点配置寄存器 3（USBD_EPRx），EA[3:0]为 2，那么可以在 USB 传输上存在端点 2 上传通道和端点 2 下传通道（具体由描述符信息决定）；配置单向端点双缓冲属性（只针对批量端点和同步端点），端点配置寄存器 3（USBD_EPRx），EA[3:0]为 2，端点类型（EPTYPE）为同步或批量端点，EP_KIND 位置 1，那么可以在 USB 传输上存在端点 2 上传通道或端点 2 下传通道，2 选 1，收发速度上与单缓冲相比更快，微控制器处理和 USBD 模块物理收发可以同步进行，降低等待

时间。

注：USBD 模块内置冲突仲裁机制，使得微控制器和 USBD 模块对分组缓冲区的访问如同对一个双端口 SRAM 的访问，即使微控制器连续访问缓冲区，也不会产生访问冲突。

每个端点配置寄存器对应一组缓冲区描述类寄存器（描述表）及相应数据收发缓冲区域，他们都位于共享的 512 字节专用 SRAM 区域内（基地址 0x40006000）。其中，USBD_BTABLE 寄存器定义缓冲区描述表在 SRAM 区域内的起始地址，而数据收发缓冲区域可以位于整个专用 SRAM 区域内的任意位置，因为它们的地址和长度都定义在对应的缓冲区描述表中，注意分配冲突问题。

图 21-1 缓冲区描述表结构



不论接收或发送，分组缓冲区都是从底部开始使用的。USBD 模块不会改变超出当前分配到的缓冲区区域以外的其他缓冲区的内容。如果缓冲区收到一个比自己大的数据分组，它只会接收最大为自身大小的数据，其他的丢掉，即发生了所谓的缓冲区溢出异常。

1）端点初始化

初始化端点的第一步是把适当的值写到 USBD_ADDRx_TX 或 USBD_ADDRx_RX 寄存器中，以便 USBD 模块能找到要传输的数据或准备好接收数据的缓冲区。USBD_EPRx 寄存器的 EPTYPE[1:0]位确定端点的基本类型，EP_KIND 位确定端点的特殊特性。作为发送方，需要设置 USBD_EPRx 寄存器的 STAT_TX 位来使能端点，并配置 COUNTx_TX 位决定发送长度。作为接收方，需要设置 STAT_RX 位来使能端点，并且设置 BL_SIZE 和 NUM_BLOCK 位，确定接收缓冲区的大小，以检测缓冲区溢出的异常。对于非同步非双缓冲批量传输的单向端点，只需要设置一个传输方向上的寄存器。一旦端点被使能，应用程序就不能再修改 USBD_EPRx 寄存器的值和 USBD_ADDRx_TX/USBD_ADDRx_RX、USBD_COUNTx_TX/USBD_COUNTx_RX 寄存器所在的位置，因为这些值会被硬件实时修改。当数据传输完成时，CTR 中断会产生，此时上述寄存器可以被访问，并重新使能新的传输。

2）IN 事务（进行数据发送）

当接收到 IN 令牌包时，如果接收到的地址和一个配置好的端点地址相符合，并且此时寄存器 USBD_EPRx 上的 STAT_TX 位表示可发送的话，USBD 模块将会根据缓冲区描述表的内容及 DTOG_TX 位进行组包编码发出数据包。如果收到的令牌包所对应的端点是无效的，将根据 USBD_EPRx 寄存器上的

STAT_TX 位发送 NAK 或 STALL 握手包而不发送数据包。

在接收到主机响应的 ACK 握手包后，USBD_EPRx 寄存器的值有以下更新：DTOG_TX 位被翻转，STAT_TX 位为‘10’（NAK 状态），使端点无效，CTR_TX 位被置位。应用程序需要通过 USBD_ISTR 寄存器的 EP_ID 和 DIR 位识别产生中断的 USB 端点。CTR_TX 事件的中断服务程序需要首先清除中断标志位，如果要继续发送数据（可以在任何需要发送数据时执行），需要准备好需要发送的数据缓冲区，更新 COUNTx_TX 为下次需要传输的字节数，最后再设置 STAT_TX 位为‘11’（ACK，端点有效），再次使能数据传输。当 STAT_TX 位为‘10’时(NAK 状态)，任何发送到该端点的 IN 请求都会被 NAK，USB 主机会重发 IN 请求直到该端点确认请求有效。

3）OUT 事务和 SETUP 事务（进行数据接收）

USBD 模块对这两种事务的处理方式基本相同；当接收到一个 OUT 或 SETUP 包时，如果接收到的地址和一个配置好的端点地址相符合，并且此时寄存器 USBD_EPRx 上的 STAT_RX 位表示可接收的话，USBD 模块根据 DTOG_RX 位判断接收数据是否 PID 匹配，如果匹配将访问缓冲区描述表，找到与该端点相关的 ADDRx_RX 和 COUNTx_RX 寄存器，将接收的数据包(先收到的为低字节)保存到 ADDRx_RX 定义的地址空间内并根据 BL_SIZE 和 NUM_BLOCK 的值检测接收是否溢出缓冲区。如果传输中没有任何错误发生，则发送 ACK 握手包到主机。即使发生 CRC 错误或其他类型的错误(位填充，帧错误等)，数据还是会被保存到分组缓冲区中，至少会保存到发生错误的数据点，只是不会发送 ACK 握手包，并且 USBD_ISTR 寄存器的 ERR 位将会置位。在这种情况下，应用程序通常不需要干涉处理，USBD 模块将从传输错误中自动恢复，并为下一次传输做好准备。如果收到的包所对应的端点没有准备好，USBD 模块将根据 USBD_EPRx 寄存器的 STAT_RX 位发送 NAK 或 STALL 握手包，数据将不会被写入接收缓冲区。

ADDRx_RX 的值决定接收缓冲区的起始地址，COUNTx_RX 决定接收缓冲区大小（期望有效数据长度+2 字节 CRC）。如果接收到的数据包长度超出了缓冲区的范围，超过范围的数据不会被写入缓冲区，USBD 模块将报告缓冲区发生溢出，并向主机发送 STALL 握手包，并置位分组缓冲区溢出标志 PMAOVR。

如果传输正确完成，USBD 模块将发送 ACK 握手包，并将实际接收数据包中的有效数据字节数写入 COUNTx_RX 寄存器中。USBD_EPRx 寄存器的值有以下更新：DTOG_RX 位翻转，STAT_RX 位为‘10’（NAK 状态）使端点无效，CTR_RX 位被置位。应用程序需要通过 USBD_ISTR 寄存器的 EP_ID 和 DIR 位识别产生中断的 USB 端点。CTR_RX 事件的中断服务程序首先要根据 SETUP 位确定传输的类型，同时清除中断标志位，然后读相关的缓冲区描述表表项指向的 COUNTx_RX 寄存器，获得此次传输的总字节数，处理接收数据。处理完后，应用程序需要将 USBD_EPRx 中的 STAT_RX 位置成‘11’（ACK 状态），使能下一次的传输。当 STAT_RX 位为‘10’时(NAK 状态)，任何一个发送到端点上的 OUT 请求都会被 NAK，SETUP 请求除外（协议规定 SETUP 请求必须以 ACK 握手包接收）。PC 主机将不断重发被 NAK 的 OUT 事务包，直到收到端点的 ACK 握手包。

4）控制传输

控制（SETUP）传输一定发生在端点 0 上，所以也称端点 0 位控制端点。控制传输由 3 个阶段组成，首先是主机发送 SETUP 事务的 SETUP 阶段，然后是主机发送零个或多个数据（IN/OUT 事务）的数据阶段，最后是状态阶段，由与数据阶段方向相反的数据事务构成。

SETUP 事务非常类似于 OUT 事务的传输过程，所以控制端点在每次发生 CTR_RX 中断时，都必须检查 USBD_EPRx 寄存器的 SETUP 位，以识别是普通的 OUT 事务还是 SETUP 事务。当主机发送 SETUP 事务下来，USBD 模块会固定回复 ACK 握手包接收下来，而忽略判断 STAT_RX 和 DTOG_RX 的内容。然后强制将 DTOG_RX 和 DTOG_TX 设置为 DATA1 状态，并设置 STAT_RX 和 STAT_TX 为‘10’（NAK），保证应用程序可以根据 SETUP 事务中的相应数据决定后面的传输是 IN 还是 OUT。如果拒绝后续数据传输或出现错误，应用程序可以设置 STAT_RX 或 STAT_TX 为‘01’，应答 STALL 握手包。如果应用程序收到一个 SETUP 事务并处理时，此时 CTR_RX 仍然保持置位，又收到一个 SETUP 包，USBD 模块会丢掉此 SETUP 包，并不给于任何握手包应答，以此来模拟一个接收错误，迫使主机再次发送 SETUP 包，这样做是为了避免丢失紧随一次 CTR_RX 中断之后的又一个 SETUP 事务传输。

在控制传输的状态阶段，如果执行的是由主机发送给设备的 OUT 事务，那么 STATUS_OUT 位(USBD_EPRx 寄存器中的 EP_KIND)应该被置位，只有这样，在状态阶段传输过程中收到了非零长度的数据分组，才会产生传输错误。在完成状态阶段传输后，应用程序应该清除 STATUS_OUT 位，并且将 STAT_RX 设为 ACK 表示已准备好接收一个新的命令请求，将 STAT_TX 则设为 NAK，不接受任何数据上传的请求。

21.2.3 双缓冲机制

在 USB 协议标准里，对不同的数据传输方式进行了应用描述。其中，批量传输适用于 USB 主机和设备间进行大批量的数据传输，主机在帧时间内利用尽可能多的带宽执行批量传输。但这种传输需要保证数据的正确性和完整性，所以传输中包含令牌包、数据包、握手包的顺序进行。同步传输适用于对数据要求恒定速率传送，但对错误有一定容忍性，认为传输一般都可以成功，主机在每个帧时间内有固定的带宽来执行同步传输以此保证传输速率，所以传输中包含令牌包、数据包的顺序进行，没有握手包来确认传输状态和终止传输。

21.2.3.1 单向双缓冲批量端点

批量传输，在单缓冲方式下，当应用程序处理批量端点的前一次的数据传输时，又收到新的数据包，USBD 模块将回应 NAK 握手包，使 PC 主机不断重发同样的数据包，直到应用程序重新设置 ACK 握手包。这样的重传占用了很多带宽，影响了批量传输的速率。因此对批量端点引入双缓冲机制来提高数据传输率。在双缓冲方式下，单向批量端点有 2 个数据缓冲区，即该端点的接收和发送两块数据缓冲区。数据翻转位（DTOG_RX 或 DTOG_TX）用来选择当前使用到两块缓冲区中的哪一块，使应用程序可以在 USBD 模块访问其中一块缓冲区的同时，对另一块缓冲区进行操作。例如，对一个双缓冲批量端点进行 OUT 事务传输时，USBD 模块将来自 PC 主机的数据保存到一个缓冲区，同时应用程序可以对另一个缓冲区中的数据进行处理(对于 IN 事务来说，情况是一样的)。这样利用 USBD 模块的接收或发送数据的时间完成应用程序的数据处理，提高了 USB 收发效率。因为对于一个传输方向需要 2 个缓冲区，所以配置双向缓冲区的批量端点必须配置为单向端点，其 USBD_EPRx 寄存器只需设定 STAT_RX 位(作为双缓冲批量接收端点)或 STAT_TX 位(作为双缓冲批量发送端点)。为尽可能利用双缓冲的优势，达到较高的传输速率，USBD 模块处理双缓冲批量端点的流控与其他端点的稍有不同。它只在缓冲区发生访问冲突时才会设置端点为 NAK 状态，而不是在每次传输成功后都将端点设为 NAK 状态。

USBD_EPRx 寄存器中的 DTOG_xx 位用来标识 USBD 模块和应用程序当前分别使用的存储缓存区，以避免发生访问冲突。当配置为单向发送双缓冲区端点时，DTOG_TX 标识 USBD 模块当前使用缓冲区，而 DTOG_RX 标识应用程序当前使用缓冲区；当配置为单向接收双缓冲区端点时，DTOG_RX 标识 USBD 模块当前使用缓冲区，而 DTOG_TX 标识应用程序当前使用缓冲区。我们命名 USBD 模块使用缓冲区标识为 DTOG，应用程序使用缓冲区标识为 SW_BUF。所以双缓冲单向批量端点标识定义如下：

表 21-1 缓冲区标识

缓冲区标识位	发送端点	接收端点
DTOG	DTOG_TX（USBD_EPRx 寄存器 bit6）	DTOG_RX(USBD_EPRx 寄存器 bit14)
SW_BUF	DTOG_RX(USBD_EPRx 寄存器 bit14)	DTOG_TX（USBD_EPRx 寄存器 bit6）

表 21-2 双缓冲批量端点缓冲区

端点类型	DTOG	SW_BUF	USBD 模块使用缓冲区	应用程序使用缓冲区
IN 端点	0	1	ADDRx_TX_0/COUNTx_TX_0	ADDRx_TX_1/COUNTx_TX_1
	1	0	ADDRx_TX_1/COUNTx_TX_1	ADDRx_TX_0/COUNTx_TX_0
	0	0	设置端点处于 NAK 状态	ADDRx_TX_0/COUNTx_TX_0
	1	1	设置端点处于 NAK 状态	ADDRx_TX_1/COUNTx_TX_1
OUT 端点	0	1	ADDRx_RX_0/COUNTx_RX_0	ADDRx_RX_1/COUNTx_RX_1

	1	0	ADDRx_RX_1/COUNTx_RX_1	ADDRx_RX_0/COUNTx_RX_0
	0	0	设置端点处于 NAK 状态	ADDRx_RX_0/COUNTx_RX_0
	1	1	设置端点处于 NAK 状态	ADDRx_RX_1/COUNTx_RX_1

应用程序配置一个双缓冲批量端点，需要设置 USBD_EPRx 寄存器的 EPTYPE[1:0]为‘00’，EP_KIND 位为‘1’。根据传输开始时用到的缓冲区来初始化 DTOG 和 SW_BUF 位。每成功完成一次传输后，USBD 模块将根据双缓冲批量端点的流量控制操作，并且持续到 EP_KIND 变为无效为止。每次传输结束，根据端点的传输方向，CTR_RX 位或 CTR_TX 位将会置位。与此同时，硬件将设置相应的 DTOG_xx 位（翻转），并实现缓冲区交换，如果没有发生 USBD 模块和应用程序的缓冲区访问冲突(即 DTOG 和 SW_BUF 为相同的值，见表 154)，则保持 STAT_xx 位的状态值，否则将会被置为‘10’（NAK 状态）。所以应用程序访问缓冲区之后，需要即使翻转 SW_BUF 位，以通知 USB 模块该块缓冲区已变为可用状态。

21.2.3.2 同步端点

同步传输一般用于传输音频流、压缩的视频流等对数据传输率有严格要求的数据。执行同步传输的端点即为同步端点。USB 主机会在每个帧时间内分配固定的带宽给同步端点进行 IN 事务或 OUT 事务传输，并且没有重传机制，无握手协议，同时传输的数据包 PID 固定为 DATA0，不会出现 DATA0 和 DATA1 数据翻转机制（控制/批量/中断传输中出现）。

因为同步传输中没有握手机制，USBD_EPRx 寄存器的 STAT_RX 位和 STAT_TX 位分别只能设成‘00’(禁止传输)和‘11’(运行传输)两种状态。同步传输使用双缓冲机制来简化软件流程，它同样使用两个缓冲区，以确保在 USB 模块使用其中一块缓冲区时，应用程序可以访问另外一块缓冲区。不同于单向批量端点的双缓冲机制，同步端点由于在 USB 标准中传输有固定的时间间隔，及容错能力，所以 USBD 模块不判断与应用程序缓冲区冲突情况，只使用 DTOG 位来标识自身当前使用的缓冲区（USBD_EPRx 寄存器中的 DTOG_RX 位用来标识接收同步端点，DTOG_TX 位用来标识发送同步端点）。

表 21-3 同步端点缓冲区标识

端点类型	DTOG	USBD 模块使用缓冲区	应用程序使用缓冲区
IN 端点	0	ADDRx_TX_0/COUNTx_TX_0	ADDRx_TX_1/COUNTx_TX_1
	1	ADDRx_TX_1/COUNTx_TX_1	ADDRx_TX_0/COUNTx_TX_0
OUT 端点	0	ADDRx_RX_0/COUNTx_RX_0	ADDRx_RX_1/COUNTx_RX_1
	1	ADDRx_RX_1/COUNTx_RX_1	ADDRx_RX_0/COUNTx_RX_0

应用程序配置一个同步端点，需要设置 USBD_EPRx 寄存器的 EPTYPE[1:0]为‘10’。根据传输开始时用到的缓冲区来初始化 DTOG 位。每成功完成一次传输后，根据端点的传输方向，CTR_RX 位或 CTR_TX 位将会置位。与此同时，硬件将设置相应的 DTOG_xx 位（翻转）实现缓冲区交换，但不会改变期望或发送的数据包 PID（固定为 DATA0）。STAT_RX 或 STAT_TX 位不会发生变化。同步传输中，即使 OUT 事务发生 CRC 错误或者缓冲区溢出，本次传输仍被看作是正确的，并且可以触发 CTR_RX 中断事件，但是，发生 CRC 错误时硬件会设置 USB_ISTR 寄存器的 ERR 位，提醒应用程序数据可能损坏。

21.2.4 挂起/唤醒流程

USB 标准中定义了一种总线状态——总线挂起，如果 USB 总线在 3ms 内没有任何活动就进入挂起状态。在这种状态下 USB 总线上提供电流会降低（低速设备一般不超过 500uA，高速或支持远程唤醒功能设备一般不超过 2.5mA）。这种电流限制对于由总线供电的 USB 设备至关重要，而自供电的设备则不需要严格遵守这样的电流消耗限制。

正常工作状态下，USB 主机会以 1ms 间隔时间发送 SOF 包，所以如果 USBD 模块检测到 3 个连续的 SOF 包丢失事件即可判断主机发出了挂起请求，此时，它会置位 USBD_ISTR 寄存器的 SUSP 位，如

果使能了中断还会触发挂起中断。USBD 模块会不断检测总线的挂起状态，并更新 SUSP 位（一直处于总线挂起状态下清除 SUSP 位标志仍然会由硬件再次置位）。所以当应用程序收到 USB 总线挂起事件后，需要执行以下流程：

将 USBD_CNTR 寄存器的 FSUSP 位置 1，屏蔽硬件的挂起状态检测，防止不断触发挂起事件。

消除或减少 USBD 模块以外其他模块的静态电流消耗。

将 USBD_CNTR 寄存器的 LPMODE 位置 1，让 USBD 模块处于低功耗运行状态，但仍可检测总线唤醒信号。

可选择关闭外部振荡器和 PLL，以停止设备的任何活动。

处于挂起状态的 USB 设备或主机，将由“唤醒”序列唤醒。所谓的“唤醒”序列可以由 USB 主机发起唤醒挂起的 USB 设备，也可以由 USB 设备触发唤醒挂起的 USB 主机，但最终由 USB 主机结束“唤醒”序列。此外，作为挂起的 USB 设备，还需能够检测 RESET 信号（总线复位）的功能，并将其当作一次正常的复位操作来执行。

挂起的 USBD 模块收到唤醒信号后会触发一个 WKUP 中断事件（通道 42），并将 USBD_ISTR 寄存器的 WKUP 位置 1，自动清除 LPMODE 位。当应用程序收到 USB 唤醒事件后，需要执行以下流程：

清除 USBD_CNTR 寄存器的 FSUSP 位，重新开启 USB 总线的挂起状态检测功能；

可选择启动外部振荡器和 PLL。

查询 USBD_FNR 寄存器的 RXDP 和 RXDM 位判断是什么触发了唤醒事件，并执行相应软件操作。

USBD 模块可以发出唤醒序列唤醒被挂起的 USB 主机。在这种情况下，先将 USBD_CNTR 寄存器的 RESUME 位置 1，然后在 1ms-15ms 之间再把它清为 0 可以启动唤醒序列。RESUME 位被清零后，唤醒过程将由主机 PC 完成（USB 主机唤醒后会继续执行此序列以唤醒其他挂载的 USB 设备）。应用程序可以查询 USBD_FNR 寄存器的 RXDP 和 RXDM 位来判断唤醒是否完成。

注：只有在 USBD 模块被设置为挂起状态时(设置 USB_CNTR 寄存器的 FSUSP 位为‘1’)，才可以设置 RESUME 位。

表 21-4 USB 总线状态

RXDP	RXDM	条件	USB 总线状态
0	0	>10ms	总线复位
0	1	>1ms（全速设备）	唤醒序列开始
		>3ms（低速设备）	挂起状态
1	0	>3ms（全速设备）	挂起状态
		>1ms（低速设备）	唤醒序列开始
1	1	-	总线错误（或干扰）

21.3 寄存器描述

USBD 模块有以下 3 类寄存器：

- 通用类寄存器：USBD 模块控制、中断相关，基地址 0x40005C00。
- 端点类寄存器：端点配置、收发状态相关，基地址 0x40005C00。
- 缓冲区描述类寄存器：数据收发缓冲区相关，基地址 0x40006000。

表 21-5 USBD 通用类寄存器列表

名称	访问地址	描述	复位值
R16_USBD_CNTR	0x40005C40	USB 控制寄存器	0x0003
R16_USBD_ISTR	0x40005C44	USB 中断状态寄存器	0x0000
R16_USBD_FNR	0x40005C48	USB 帧编号寄存器	X

R16_USBD_DADDR	0x40005C4C	USB 设备地址寄存器	0x0000
R16_USBD_BTABLE	0x40005C50	USB 分组缓冲区描述表地址寄存器	0x0000

表 21-6 USBD 端点类寄存器列表

名称	访问地址	描述	复位值
R16_USBD_EPR0	0x40005C00	USB 端点配置寄存器 0	0x0000
R16_USBD_EPR1	0x40005C04	USB 端点配置寄存器 1	0x0000
R16_USBD_EPR2	0x40005C08	USB 端点配置寄存器 2	0x0000
R16_USBD_EPR3	0x40005C0C	USB 端点配置寄存器 3	0x0000
R16_USBD_EPR4	0x40005C10	USB 端点配置寄存器 4	0x0000
R16_USBD_EPR5	0x40005C14	USB 端点配置寄存器 5	0x0000
R16_USBD_EPR6	0x40005C18	USB 端点配置寄存器 6	0x0000
R16_USBD_EPR7	0x40005C1C	USB 端点配置寄存器 7	0x0000

表 21-7 USBD 缓冲区描述类寄存器列表

名称	访问地址	描述	复位值
R16_USBD_ADDR0_TX	0x40006000+[USBD_BTABLE]	端点发送缓存区地址寄存器 0	0x0000
R16_USBD_COUNT0_TX	0x40006004+[USBD_BTABLE]	端点发送数据字节数寄存器 0	0x0000
R16_USBD_ADDR0_RX	0x40006008+[USBD_BTABLE]	端点接收缓存区地址寄存器 0	0x0000
R16_USBD_COUNT0_RX	0x4000600C+[USBD_BTABLE]	端点接收数据字节数寄存器 0	0x0000
R16_USBD_ADDR1_TX	0x40006010+[USBD_BTABLE]	端点发送缓存区地址寄存器 1	0x0000
R16_USBD_COUNT1_TX	0x40006014+[USBD_BTABLE]	端点发送数据字节数寄存器 1	0x0000
R16_USBD_ADDR1_RX	0x40006018+[USBD_BTABLE]	端点接收缓存区地址寄存器 1	0x0000
R16_USBD_COUNT1_RX	0x4000601C+[USBD_BTABLE]	端点接收数据字节数寄存器 1	0x0000
R16_USBD_ADDR2_TX	0x40006020+[USBD_BTABLE]	端点发送缓存区地址寄存器 2	0x0000
R16_USBD_COUNT2_TX	0x40006024+[USBD_BTABLE]	端点发送数据字节数寄存器 2	0x0000
R16_USBD_ADDR2_RX	0x40006028+[USBD_BTABLE]	端点接收缓存区地址寄存器 2	0x0000
R16_USBD_COUNT2_RX	0x4000602C+[USBD_BTABLE]	端点接收数据字节数寄存器 2	0x0000
R16_USBD_ADDR3_TX	0x40006030+[USBD_BTABLE]	端点发送缓存区地址寄存器 3	0x0000
R16_USBD_COUNT3_TX	0x40006034+[USBD_BTABLE]	端点发送数据字节数寄存器 3	0x0000
R16_USBD_ADDR3_RX	0x40006038+[USBD_BTABLE]	端点接收缓存区地址寄存器 3	0x0000
R16_USBD_COUNT3_RX	0x4000603C+[USBD_BTABLE]	端点接收数据字节数寄存器 3	0x0000
R16_USBD_ADDR4_TX	0x40006040+[USBD_BTABLE]	端点发送缓存区地址寄存器 4	0x0000
R16_USBD_COUNT4_TX	0x40006044+[USBD_BTABLE]	端点发送数据字节数寄存器 4	0x0000
R16_USBD_ADDR4_RX	0x40006048+[USBD_BTABLE]	端点接收缓存区地址寄存器 4	0x0000
R16_USBD_COUNT4_RX	0x4000604C+[USBD_BTABLE]	端点接收数据字节数寄存器 4	0x0000
R16_USBD_ADDR5_TX	0x40006050+[USBD_BTABLE]	端点发送缓存区地址寄存器 5	0x0000
R16_USBD_COUNT5_TX	0x40006054+[USBD_BTABLE]	端点发送数据字节数寄存器 5	0x0000
R16_USBD_ADDR5_RX	0x40006058+[USBD_BTABLE]	端点接收缓存区地址寄存器 5	0x0000
R16_USBD_COUNT5_RX	0x4000605C+[USBD_BTABLE]	端点接收数据字节数寄存器 5	0x0000
R16_USBD_ADDR6_TX	0x40006060+[USBD_BTABLE]	端点发送缓存区地址寄存器 6	0x0000
R16_USBD_COUNT6_TX	0x40006064+[USBD_BTABLE]	端点发送数据字节数寄存器 6	0x0000
R16_USBD_ADDR6_RX	0x40006068+[USBD_BTABLE]	端点接收缓存区地址寄存器 6	0x0000

R16_USBD_COUNT6_RX	0x4000606C+[USBD_BTABLE]	端点接收数据字节数寄存器 6	0x0000
R16_USBD_ADDR7_TX	0x40006070+[USBD_BTABLE]	端点发送缓存区地址寄存器 7	0x0000
R16_USBD_COUNT7_TX	0x40006074+[USBD_BTABLE]	端点发送数据字节数寄存器 7	0x0000
R16_USBD_ADDR7_RX	0x40006078+[USBD_BTABLE]	端点接收缓存区地址寄存器 7	0x0000
R16_USBD_COUNT7_RX	0x4000607C+[USBD_BTABLE]	端点接收数据字节数寄存器 7	0x0000

注：以上缓冲区描述类寄存器和端点配置寄存器使用相对应。例如：USB 端点配置寄存器 0 对应端点发送缓存区地址寄存器 0、端点发送数据字节数寄存器 0、端点接收缓存区地址寄存器 0、端点接收数据字节数寄存器 0。

21.3.1 USB 控制寄存器（USBD_CNTR）

偏移地址：0x40

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRM	PMA OVRM	ERRM	WKUPM	SUSPM	RESETM	SOFM	ESOFM	Reserved			RESUME	FSUSP	LP MODE	Reserved	FRES

位	名称	访问	描述	复位值
15	CTRM	RW	正确传输中断使能位： 1：使能正确传输(CTR)中断，在中断寄存器的相应位被置 1 时产生中断； 0：禁止正确传输(CTR)中断。	0
14	PMAOVRM	RW	分组缓冲区溢出中断使能位： 1：使能 PMAOVR 中断，在中断寄存器的相应位被置 1 时产生中断； 0：禁止 PMAOVR 中断。	0
13	ERRM	RW	出错中断使能位： 1：使能出错中断，在中断寄存器的相应位被置 1 时产生中断； 0：禁止出错中断。	0
12	WKUPM	RW	唤醒中断使能位： 1：使能唤醒中断，在中断寄存器的相应位被置 1 时产生中断； 0：禁止唤醒中断。	0
11	SUSPM	RW	挂起中断使能位： 1：使能挂起(SUSP)中断，在中断寄存器的相应位被置 1 时产生中断； 0：禁止挂起(SUSP)中断。	0
10	RESETM	RW	USB 复位（总线复位或强制复位）中断使能位： 1：使能 USB 复位中断，在中断寄存器的相应位被置 1 时产生中断； 0：禁止 USB 复位中断。	0
9	SOFM	RW	帧首（SOF）中断使能位： 1：使能 SOF 中断，在中断寄存器的相应位被置 1 时产生中断； 0：禁止 SOF 中断。	0
8	ESOFM	RW	定时帧首丢失中断使能位：	0

			1：使能 ESOF 中断，在中断寄存器的相应位被置 1 时产生中断； 0：禁止 ESOF 中断。	
[7:5]	Reserved	RO	保留。	0
4	RESUME	RW	唤醒请求控制位： 1：输出唤醒信号； 0：空闲状态。 根据 USB 协议，如果此位在 1ms 到 15ms 内保持有效，主机将对 USBD 模块实行唤醒操作。 *注：只有在 FSUSP 位为 1 时，才可以设置此位。*	0
3	FSUSP	RW	屏蔽挂起检测控制位： 1：屏蔽总线挂起状态检测。此时 USB 模拟收发器的时钟和静态功耗仍然保持。如果需要进入低功耗状态(总线供电类的设备)，需要先置位 FSUSP 再置位 LPMODE。 0：打开总线挂起状态检测。 *注：当 USB 总线上保持 3ms 没有数据通信（包括 SOF）时，SUSP 中断会被触发，此时软件必需设置此位，否则会一直触发 SUSP 中断。*	0
2	LPMODE	RW	低功耗模式控制位： 此模式用于在 USB 挂起状态下降低功耗。在此模式下，除了外接上拉电阻的供电，其他的静态功耗都被关闭，系统时钟将会停止或者降低到一定的频率来减少耗电。USB 总线上的活动(唤醒事件)将会清除此位(软件也可以清 0)。 1：低功耗模式； 0：非低功耗模式。	0
1	Reserved	RO	保留。	1
0	FRES	RW	强制 USB 复位控制位： 1：对 USBD 模块强制复位。USBD 模块将一直保持在复位状态下直到软件清除此位。如果 USB 复位中断被使能，将产生复位中断； 0：清除 USB 复位。	1

21.3.2 USB 中断状态寄存器（USBD_ISTR）

偏移地址：0x44

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR	PMAOVR	ERR	WKUP	SUSP	RESET	SOF	ESOF	Reserved			DIR	EP_ID[3:0]			

位	名称	访问	描述	复位值
15	CTR	RO	正确的传输状态指示。此位在端点正确完成一次数据传输后由硬件置位。应用程序可以通过 DIR 和 EP_ID 位来识别是哪个端点完成了正确的数据传输。	0
14	PMAOVR	RW0	分组缓冲区溢出标志。 此位在微控制器长时间没有响应一个访问 USB 分组缓冲区请求时由硬件置位。USBD 模块通常在以下情况时	0

			置位该位：在接收过程中一个 ACK 握手分组没有被发送，或在发送过程中发生了比特填充错误，在以上两种情况下主机都会要求数据重传。在正常的数据传输中不会产生 PMAOVR 中断。由于失败的传输都将由主机发起重传，应用程序就可以在这个中断的服务程序中加速设备的其他操作，并准备重传。但这个中断不会在同步传输中产生(同步传输不支持重传)因此数据可能会丢失。 此位可读，写 0 清除，写 1 无效。	
13	ERR	RWO	出错标志，下列错误发生时硬件会置位此位： NANS：无应答。主机的应答超时。 CRC：校验错误。USB 的包中的 CRC 校验出错。 BST：位填充错误。USB 数据位中检测出位填充错误。 FVIO：帧格式错误。收到非标准帧(如 EOP 出现在错误的时刻，错误的令牌等)。 USB 应用程序通常可以忽略这些错误，因为 USBD 模块和主机在发生错误时都会启动重传机制。此位产生的中断可以用于应用程序的开发阶段，可以用来监测 USB 总线的传输质量，标识用户可能发生的错误(连接线松，环境干扰严重，USB 线损坏等)。 此位可读，写 0 清除，写 1 无效。	0
12	WKUP	RWO	唤醒信号标志： 当 USBD 模块处于挂起状态时，如果检测到唤醒信号，此位将由硬件置 1。此时 CTLR 寄存器的 LP_MODE 位将被清 0，FSUSP 位需要软件清 0 开启挂起检测。同时 USB_WAKEUP 被激活，通知设备的其他部分(如唤醒单元)将开始唤醒过程。 此位可读，写 0 清除，写 1 无效。	0
11	SUSP	RWO	总线挂起标志： 此位在 USB 线上超过 3ms 没有信号传输时由硬件置位。USB 复位（总线复位或强制复位）撤销后，硬件立即使能对挂起信号的检测，但在挂起模式下(FSUSP＝1)硬件不会再检测挂起信号直到唤醒过程结束。 此位可读，写 0 清除，写 1 无效。	0
10	RESET	RWO	USB 复位（总线复位或者强制复位）标志： 此位在 USBD 模块检测到 USB 总线复位信号边沿或强制复位状态时由硬件置位。此时 USBD 模块将复位内部协议状态机，并在中断使能的情况下触发复位中断来响应。USBD 模块的发送和接收部分将被禁止，直到此位被清除。所有的配置寄存器不会被复位，除非应用程序对他们清零。这用来保证在复位撤销后 USB 传输还可以立即正确执行。但设备的地址和端点寄存器会被 USB 复位所复位。 此位可读，写 0 清除，写 1 无效。	0
9	SOF	RWO	帧首（SOF）标志： 此位在 USBD 模块检测到总线上的 SOF 包时由硬件置	0

			位。中断服务程序可以通过检测 SOF 事件来完成与主机的 1ms 同步，并正确读出寄存器在收到 SOF 时的更新内容(此功能在同步传输时非常有意义)。 此位可读，写 0 清除，写 1 无效。	
8	ESOF	RWO	定时帧首（ESOF）丢失标志： 此位在 USBD 模块未按时收到 SOF 包时由硬件置位。主机应该每毫秒都发送 SOF 包，但如果 USBD 模块没有收到，挂起定时器将触发此中断。如果连续发生 3 次 ESOF 中断，也就是连续 3 次未收到 SOF 包，将产生 SUSP 中断。 此位可读，写 0 清除，写 1 无效。	0
[7:5]	Reserved	RO	保留。	0
4	DIR	RO	事务数据传输方向。此位在完成数据传输产生中断后由硬件根据传输方向写入。 如果 DIR＝0，相应端点的 CTR_TX 位被置位，标志一个 IN 事务(数据从 USBD 模块传输到 PC 主机)的传输完成。 如果 DIR＝1，相应端点的 CTR_RX 位被置位，标志一个 OUT 事务(数据从 PC 主机传输到 USBD 模块)的传输完成。如果 CTR_TX 位同时也被置位，就标志同时存在挂起的 OUT 事务和 IN 事务。 应用程序可以利用该信息访问 USBD_EPnR 位对应的操作，它表示挂起中断传输方向的信息。	0
[3:0]	EP_ID	RO	端点号。 此位在 USBD 模块完成数据传输产生中断后由硬件根据请求中断的端点号写入。如果同时有多个端点的请求中断，硬件写入优先级最高的端点号。端点的优先级按以下方法定义：同步端点和双缓冲批量端点具有高优先级，其他的端点为低优先级。如果多个同优先级的端点请求中断，则根据端点号来确定优先级，即端点 0 具有最高优先级，端点号越小，优先级越高。 应用程序可以通过上述的优先级策略顺序处理端点的中断请求。	0

21.3.3 USB 帧编号寄存器（USBD_FNR）

偏移地址：0x48

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDP	RXDM	LCK	LSOF[1:0]		FN[10:0]										

位	名称	访问	描述	复位值
15	RXDP	RO	D+数据线电平状态。	0
14	RXDM	RO	D-数据线电平状态。	0
13	LCK	RO	SOF 包计数停止锁定位。 USBD 模块在复位或唤醒序列结束后会检测 SOF 包，如果连续检测到至少 2 个 SOF 包，则硬件会置位此位。此位一旦锁定，帧计数器将停止计数，一直等到 USBD	0

			模块复位或总线挂起时再恢复计数。	
[12:11]	LSOF	RO	帧首丢失标志位。 当 ESOF 事件发生时，硬件会将丢失的 SOF 包的数目写入此域。如果再次收到 SOF 包，此域被清除。	X
[10:0]	FN	RO	帧编号。 此域为最新收到的 SOF 包中的 11 位帧编号。主机每发送一个帧，帧编号都会自加，这对于同步传输非常有意义。此域在发生 SOF 中断时更新。	X

21.3.4 USB 设备地址寄存器（USBD_DADDR）

偏移地址：0x4C

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

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
7	EF	RW	USB 功能使能位。此位在需要使能 USB 设备功能时由应用程序置位。如果此位为 0，USBD 模块将停止工作，忽略所有寄存器的设置，不响应任何 USB 通信。 1：使能 USB 设备功能； 0：停止 USB 设备功能。	0
[6:0]	ADD	RW	USB 设备地址。 此域是 USB 主机在枚举过程中为 USB 设备分配的地址值。该地址值和 EA 位必需和 USB 令牌包中的地址信息匹配，才能在指定的端点进行正确的 USB 传输。	0

21.3.5 USB 分组缓冲区描述表地址寄存器（USBD_BTABLE）

偏移地址：0x4C

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

位	名称	访问	描述	复位值
[15:3]	BLABLE	RW	缓冲表。 此域是分组缓冲区描述表的基地址。分组缓冲区描述表用来指示每个端点的分组缓冲区地址和大小，按 8 字节对齐(即最低 3 位为 000)。每次传输开始时，USBD 模块读取相应端点所对应的分组缓冲区描述表获得缓冲区地址和大小信息。	
[2:0]	Reserved	RO	保留。	0

21.3.6 USB 端点配置寄存器 x（USBD_EPRx）（x=0/1/2/3/4/5/6/7）

偏移地址：0x00-0x1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR_RX	DTOG_RX	STAT_RX[1:0]		STEUP	EPTYPE[1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX[1:0]		EA[3:0]			

位	名称	访问	描述	复位值
15	CTR_RX	RW0	正确接收标志位（OUT/SETUP）。 此位在正确接收到 OUT 或 SETUP 事务（发送 ACK 应答）时由硬件置位。如果 CTRM 位已置位，相应的中断会产生，应用程序需要在处理完该事件后清除此位。收到的是 OUT 事务还是 SETUP 事务可以通过下面的 SETUP 位确定。 此位可读，写 0 清除，写 1 无效。 *注：以 NAK 或 STALL 应答的事务或出错的传输此位不会置位。*	0
14	DTOG_RX	RW1T	期望下次接收的数据包 PID(OUT/SETUP)，硬件设置： 1：期望 DATA1； 0：期望 DATA0。 对于非同步端点，在接收正确的 PID 数据包后，USBD 模块发送 ACK 握手包，硬件自动翻转此位。 对于控制端点，硬件在收到正确的 SETUP 包后置位（DATA1）。 对于有双缓冲属性的端点，硬件除了自动翻转此位表示期望数据包 PID 外，还根据此位标识来支持双缓冲区的交换(请参考双缓冲机制中描述)。 对于同步端点，硬件不判断数据包 PID，仅通过此位标识支持双缓冲区的交换。 此位可读，写 0 无效，写 1 翻转。 *注：应用程序可以对此位进行初值设定，或者翻转此位用于特殊用途。*	0
[13:12]	STAT_RX	RW1T	表示数据接收的状态位(OUT/SETUP 事务中)： 00：DISABLED，端点忽略所有的接收请求，不应答； 01：STALL，端点以 STALL 包响应接收请求； 10：NAK，端点以 NAK 包响应接收请求； 11：ACK，端点以 ACK 包响应接收请求。 当一次正确的 OUT 或 SETUP 数据传输完成后(CTR_RX=1)，硬件会自动设置此位为 NAK 状态，使应用程序有足够的时间处理并响应下一个事务。 对于双缓冲批量端点，由于使用特殊的传输流量控制策略，会根据使用的缓冲区状态控制传输状态(请参考双缓冲端点)。 对于同步端点，由于端点状态只能是有效或禁用，因此硬件不会在正确的传输之后设置此位。 如果将此域设为 STALL 或 NAK，USBD 模块响应的操作是未定义的。	0

<table>
<tr><td></td><td></td><td></td><td>此域可读，位写 0 无效，写 1 翻转。
注：应用程序可以对域位进行初值设定。</td><td></td></tr>
<tr><td>11</td><td>STEUP</td><td>RO</td><td>SETUP 事务传输完成标志位：
1：是 SETUP 事务，并正确接收（发送 ACK 应答）；
0：非 SETUP 事务。
注意：硬件会在 CTR_RX=0 条件时，才可能修改此位。</td><td>0</td></tr>
<tr><td>[10:9]</td><td>EPTYPE</td><td>RW</td><td>传输端点类型：
00：BULK，批量端点；
01：CONTROL，控制端点；
10：ISO，同步端点；
11：INTERRUPT，中断端点。
只有控制端点才会有 SETUP 传输，其他类型的端点无视此类传输。SETUP 传输不能以 NAK 或 STALL 包响应，如果控制端点在收到 SETUP 包时处于 NAK 状态，USBD 模块将不响应请求，就会出现接收错误。如果控制端点处于 STALL 状态，SETUP 包会被正确接收，数据会被正确传输，并产生一个正确传输完成的中断。控制端点的 OUT 包按普通端点的方式处理。
批量端点和中断端点的处理方式非常类似，仅在对 EP_KIND 位的处理上有差别。</td><td>0</td></tr>
<tr><td>8</td><td>EP_KIND</td><td>RW</td><td>端点特殊类型控制位（配合 EP_TYPE 使用）：
<table>
<tr><th>EPTYPE[1:0]</th><th>EP_KIND</th></tr>
<tr><td>BULK</td><td>DBL_BUF：开启双缓冲。</td></tr>
<tr><td>CONTROL</td><td>STATUS_OUT：控制传输状态阶段数据包长度判断。</td></tr>
<tr><td>ISO</td><td>未使用。</td></tr>
<tr><td>INTERRUTP</td><td>未使用。</td></tr>
</table>
DBL_BUF：设置此位使能批量端点的双缓冲功能。
STATUS_OUT：设置此位表示 USB 设备期望主机发送控制传输中的状态阶段事务，此时，设备对于任何长度不为 0 的数据包都响应 STALL 握手包。（此功能仅用于控制端点，有利于提供对于协议层错误的检测。）
如果 STATUS_OUT 位被清除，处于状态阶段的 OUT 事务可以包含任意长度的数据。</td><td>0</td></tr>
<tr><td>7</td><td>CTR_TX</td><td>RWO</td><td>正确发送标志位（IN）：
此位在正确的 IN 事务（收到 ACK 应答）完成时由硬件置位。如果 CTRM 位已被置位，会产生相应的中断，应用程序需要在处理完该事件后清除此位。在 IN 分组结束时，如果主机响应 NAK 或 STALL 则此位不会被置位，因为数据传输没有成功。
此位可读，写 0 清除，写 1 无效。
注：如果主机以 NAK 或 STALL 响应则此位不会置位。</td><td>0</td></tr>
<tr><td>6</td><td>DTOG_TX</td><td>RW1T</td><td>要发送的数据包 PID(IN)，硬件设置：
1：发送 DATA1；
0：发送 DATA0。
对于非同步端点，在发送正确的 PID 数据包后，如果</td><td>0</td></tr>
</table>

			USBD 模块收到主机的 ACK 握手包，硬件自动翻转此位。 对于控制端点，硬件在收到正确的 SETUP 包后置位（DATA1）。 对于有双缓冲属性的端点，硬件除了自动翻转此位表示发送数据包 PID 外，还根据此位标识来支持双缓冲区的交换(请参考双缓冲机制中描述)。 对于同步端点，硬件强制发送数据包 DATA0，并且通过此位标识支持双缓冲区的交换。 此位可读，写 0 无效，写 1 翻转。 *注：应用程序可以对此位进行初值设定，或翻转此位用于特殊用途。*	
[5:4]	STAT_TX	RW1T	表示发送数据的状态位(IN 事务中)： 00：DISABLED，端点忽略所有的发送请求，不应答； 01：STALL，端点以 STALL 包响应主机 IN 请求； 10：NAK，端点以 NAK 包响应主机 IN 请求； 11：ACK，端点可以发送数据。 当正确完成一次 IN 事务数据传输完成后(CTR_TX=1)，硬件会自动设置此位为 NAK 状态，以保证应用程序有足够的时间处理并响应下一个事务传输。 对于双缓冲批量端点，由于使用特殊的传输流量控制策略，会根据使用的缓冲区状态控制传输状态(请参考双缓冲端点)。 对于同步端点，由于端点状态只能是有效或禁用，因此硬件不会在正确的传输之后设置此位。 如果将此域设为 STALL 或者 NAK，USBD 模块响应的操作是未定义的。 此域可读，位写 0 无效，写 1 翻转。 *注：应用程序可以对域位进行初值设定。*	0
[3:0]	EA	RW	端点地址域（设置端点号）： 应用程序要为此端点配置寄存器设置一个端点地址。	0

21.3.7 端点发送缓存区地址寄存器 x（USBD_ADDRx_TX）（x=0/1/2/3/4/5/6/7）

偏移地址：[USBD_BTABLE] + x*16

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

位	名称	访问	描述	复位值
[15:1]	ADDRx_TX	RW	待发送数据缓冲区起始地址（IN 事务中）。	0
0	-	RZ	缓冲区的地址必须按 2 字节对齐，所以此位必须为 0。	0

21.3.8 端点发送数据字节数寄存器 x（USBD_COUNTx_TX）（x=0/1/2/3/4/5/6/7）

偏移地址：[USBD_BTABLE] + x*16 + 4

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

位	名称	访问	描述	复位值
[15:10]	Reserved	RO	保留。	0
[9:0]	COUNTx_TX	RW	待发送数据长度字节数（IN 事务中）。	0

注：双缓冲区和同步 IN 端点有 2 个 USBD_ADDRx_TX 寄存器和 2 个 USB_COUNTx_TX 寄存器：分别为 USBD_ADDRx_TX_1 和 USBD_ADDRx_TX_0，USB_COUNTx_TX_1 和 USB_COUNTx_TX_0，内容如下：

USBD_ADDRx_TX 映射为 USBD_ADDRx_TX_0

USBD_ADDRx_RX 映射为 USBD_ADDRx_TX_1

USBD_COUNTx_TX 映射为 USB_COUNTx_TX_0

USBD_COUNTx_RX 映射为 USB_COUNTx_TX_1

21.3.9 端点接收缓存区地址寄存器 x（USBD_ADDRx_RX）（x=0/1/2/3/4/5/6/7）

偏移地址：[USBD_BTABLE] + x*16 + 8

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

位	名称	访问	描述	复位值
[15:1]	ADDRx_RX	RW	待接收数据缓冲区起始地址（OUT 或 SETUP 事务中）。	0
0	-	RZ	缓冲区的地址必须按 2 字节对齐，所以此位必须为 0。	0

21.3.10 端点接收数据字节数寄存器 x（USBD_COUNTx_RX）（x=0/1/2/3/4/5/6/7）

偏移地址：[USBD_BTABLE] + x*16 + 4

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

位	名称	访问	描述	复位值
15	BLSIZE	RW	存储区块大小： 1：块大小 2 字节，配合 NUM_BLOCK 使用，可分配接收缓冲区范围 2-62 字节； 0：块大小 32 字节，配合 NUM_BLOCK 使用，可分配接收缓冲区范围 32-512 字节。	0
[14:10]	NUM_BLOCK	RW	存储区块数目。	0
[9:0]	COUNTx_RX	RO	端点实际接收数据长度字节数(OUT 或 SETUP 事务中)。	X

注：双缓冲区和同步 IN 端点有 2 个 USBD_ADDRx_RX 寄存器和 2 个 USB_COUNTx_RX 寄存器：分别为 USBD_ADDRx_RX_1 和 USBD_ADDRx_RX_0，USB_COUNTx_RX_1 和 USB_COUNTx_RX_0，内容如下：

USBD_ADDRx_TX 映射为 USBD_ADDRx_RX_0

USBD_ADDRx_RX 映射为 USBD_ADDRx_RX_1

USBD_COUNTx_TX 映射为 USB_COUNTx_RX_0

USBD_COUNTx_RX 映射为 USB_COUNTx_RX_1

USBD_COUNTx_RX 寄存器的高 6 位定义了接收分组缓冲区的大小，以便 USBD 模块可以检测缓冲区的溢出边界。缓冲区的大小可以依据设备枚举过程中的端点描述符中参数 maxPacketSize 表述。

表 20-8 缓冲区大小定义

NUM_BLOCK[4:0]	接收缓冲区限制大小	
	BLSIZE = 0	BLSIZE = 1
00000	不允许使用	32 字节
00001	2 字节	64 字节
00010	4 字节	96 字节
00011	6 字节	128 字节
…	…	…
01111	30 字节	512 字节
10000	32 字节	保留
…	…	…
11110	60 字节	保留
11111	62 字节	保留

第 22 章 USB 主机/设备控制器（USBHD）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器系列部分产品。

22.1 USB 控制器简介

内嵌 USB2.0 控制器和 USB-PHY，具有主机控制器和 USB 设备控制器双重角色。当作为主机控制器时，它可支持低速、全速和高速的 USB 设备/HUB。当作为设备控制器时，可以灵活设置为低速、全速或高速模式以适应各种应用。

USB 控制器特性如下：

- 支持 USB Host 主机功能和 USB Device 设备功能。
- 主机模式下支持下行端口连接高速/全速 HUB。
- 设备模式下支持 USB2.0 高速 480Mbps、全速 12Mbps 或低速 1.5Mbps。
- 支持 USB 控制传输、批量传输、中断传输和同步/实时传输。
- 支持 DMA 直接访问各端点缓冲区的数据。
- 支持挂起，唤醒/远程唤醒。
- 端点 0 支持最大 64 字节的数据包，除设备端点 0 外，其他端点均支持最大 1024 字节的数据包，且均支持双缓冲。

22.2 寄存器描述

USB 相关寄存器分为 3 个部分，部分寄存器是在主机和设备模式下进行复用的。

- USB 全局寄存器
- USB 设备控制寄存器
- USB 主机控制寄存器

22.2.1 全局寄存器描述

表 22-1 USBHD 相关寄存器列表

名称	访问地址	描述	复位值
R8_USB_CTRL	0x40023400	USB 控制寄存器	0x06
R8_USB_INT_EN	0x40023402	USB 中断使能寄存器	0
R8_USB_DEV_AD	0x40023403	USB 设备地址寄存器	0
R16_USB_FRAME_NO	0x40023404	USB 帧号寄存器	0
R8_USB_SUSPEND	0x40023406	USB 挂起控制寄存器	0
R8_USB_SPPED_TYPE	0x40023408	USB 当前速度类型寄存器	0
R8_USB_MIS_ST	0x40023409	USB 杂项状态寄存器	xx10_1000b
R8_USB_INT_FG	0x4002340A	USB 中断标志寄存器	0
R8_USB_INT_ST	0x4002340B	USB 中断状态寄存器	00xx_xxxxb
R16_USB_RX_LEN	0x4002340C	USB 接收长度寄存器	xx

22.2.1.1 USB 控制寄存器(R8_USB_CTRL)

位	名称	访问	描述	复位值
7	RB_UC_HOST_MODE	RW	USB 工作模式选择位： 1：主机模式（HOST）； 0：设备模式（DEVICE）。	0

[6:5]	RB_UC_SPEED_TYPE	RW	USB 总线信号传输速率选择位： 00：全速； 01：高速； 10：低速。	0
4	RB_UC_DEV_PU_EN	RW	设备模式下,USB 设备使能和内部上拉电阻控制位： 1:使能USB设备传输并且启用内部上拉电阻； 0：不启用。	0
3	RB_UC_INT_BUSY	RW	USB 传输完成中断标志未清零前自动暂停使能位： 1：在中断标志 UIF_TRANSFER 未清零前自动暂停，设备模式下自动应答忙 NAK，主机模式下自动暂停后续传输； 0：不暂停。	0
2	RB_UC_RESET_SIE	RW	USB 协议处理器软件复位控制位： 1：强制复位 USB 协议处理器（SIE），需要软件清零； 0：不复位。 该位清除后,PB6/PB7 自动切换为 USBIO 模式	1
1	RB_UC_CLR_ALL	RW	USB 的 FIFO 和中断标志清零： 1:清空 USB 中断标志和 FIFO,需要软件清零； 0：不清空。	1
0	RB_UC_DMA_EN	RW	使能 USB 的 DMA,正常传输模式下该位必须设置为 1： 1：使能 DMA 功能和 DMA 中断； 0：关闭 DMA。	0

22.2.1.2 USB 中断使能寄存器(R8_USB_INT_EN)

位	名称	访问	描述	复位值
7	RB_UIE_DEV_NAK	RW	USB 设备模式，接收到 NAK 中断： 1：使能相应中断；0：禁止相应中断。	0
6	RB_UIE_ISO_ACT	RW	同步传输开始发送/接收数据中断： 1：使能相应中断；0：禁止相应中断。	0
5	RB_UIE_SETUP_ACT	RW	SETUP 事务完成中断： 1：使能相应中断；0：禁止相应中断。	0
4	RB_UIE_FIFO_OV	RW	FIFO 溢出中断： 1：使能中断；　　0：禁止中断。	0
3	RB_UIE_SOF_ACT	RW	USB 主机模式，SOF 定时中断： 1：使能中断；　　0：禁止中断。 USB 设备模式,使能后接收 SOF 包产生传输完成中断	0
2	RB_UIE_SUSPEND	RW	USB 总线挂起或唤醒事件中断： 1：使能中断；　　0：禁止中断。	0
1	RB_UIE_TRANSFER	RW	USB 传输（不包括 SETUP 事务）完成中断： 1：使能中断；　　0：禁止中断。	0

0	RB_UIE_DETECT	RW	USB 主机模式，USB 设备连接或断开事件中断： 1：使能中断；　　0：禁止中断。	0
	RB_UIE_BUS_RST	RW	USB 设备模式，USB 总线复位事件中断： 1：使能中断；　　0：禁止中断。	0

22.2.1.3 USB 设备地址寄存器(R16_USB_DEV_AD)

位	名称	访问	描述	复位值
7	保留	RO	保留。	0
[6:0]	RB_MASK_USB_ADDR	RW	主机模式下是当前操作的 USB 设备的地址或 HUB 地址； 设备模式：该 USB 自身地址。	0

22.2.1.4 USB 帧号寄存器（R16_USB_FRAME_NO）

位	名称	访问	描述	复位值
[15:0]	USB_FRAME_NO	RO	帧号，主机模式下表示即将发送的 SOF 包的帧号，设备模式下表示当前接收到的 SOF 包的帧号。其中低 11 位为有效帧号，高 3 位为高速模式的微帧号。	0

注：USB_FRAME_NO 是 16 为寄存器，其中低 11 位表示 SOF 包帧号，高 3 为表示当前属于第几个微帧，可在操作高速 HUB 下进行中断、同步/实时传输时使用。

22.2.1.5 USB 挂起寄存器（R8_USB_SUSPEND）

位	名称	访问	描述	复位值
[7:6]	保留	RO	保留。	0
[5:4]	RB_USB_LINESTATE	RO	PHY 的 Linestate 信号	X
3	保留	RO	保留。	0
2	RB_USB_WAKEUP_ST	RO	在挂起状态下，如果接收到主机的唤醒信号，该位置 1，直到退出挂起状态。	X
[1:0]	RB_USB_SYS_MOD	RW	主机模式下的测试模式	00b

注：需要远程唤醒时，将 RB_UH_REMOTE_WKUP 位拉高再拉低即可。

22.2.1.6 USB 速度类型寄存器（R8_USB_SPEED_TYPE）

位	名称	访问	描述	复位值
[7:2]	保留	RO	保留。	0
[1:0]	RB_USB_SPEED_TYPE	RO	在主机模式下，表示当前连接的设备速度类型，在设备模式下，表示当前设备的速度类型； 00：全速； 01：高速； 10：低速。	00b

注：区别于 R8_USB_CTRL 寄存器中的 RB_UC_SPEED_TYPE，RB_UC_SPEED_TYPE 表示期望处于的最高速度，假设在设备模式下，设置 RB_UC_SPEED_TYPE 为高速，当该设备连接在一个全速主机下，则实际的速度类型就是全速，通过查询 R8_USB_SPEED_TYPE 寄存器可以获知。在主机模式下，设置

RB_UC_SPEED_TYPE 为高速，当连接一个全速设备时，则实际通讯速度就是全速，通过查询 R8_USB_SPEED_TYPE 寄存器可以获知。

22.2.1.7 USB 杂项状态寄存器(R8_USB_MIS_ST)

位	名称	访问	描述	复位值
7	RB_UMS_SOF_PRES	RO	USB 主机模式下 SOF 包预示状态位： 1：将要发送 SOF 包，此时如有其它 USB 数据包将被自动延后； 0：无 SOF 包发送。	X
6	RB_UMS_SOF_ACT	RO	USB 主机模式下 SOF 包传输状态位： 1：正在发出 SOF 包； 0：发送完成或空闲。	X
5	RB_UMS_SIE_FREE	RO	USB 协议处理器的空闲状态位： 1：协议器空闲； 0：忙，正在进行 USB 传输。	1
4	RB_UMS_R_FIFO_RDY	RO	USB 接收 FIFO 数据就绪状态位： 1：接收 FIFO 非空； 0：接收 FIFO 为空。	0
3	RB_UMS_BUS_RESET	RO	USB 总线复位状态位： 1：当前 USB 总线处于复位态； 0：当前 USB 总线处于非复位态。	X
2	RB_UMS_SUSPEND	RO	USB 挂起状态位： 1：USB 总线处于挂起态，有一段时间没有 USB 活动； 0：USB 总线处于非挂起态。	0
1	RB_UMS_DEV_ATTACH	RO	USB 主机模式下端口的 USB 设备连接状态位： 1：端口已经连接 USB 设备； 0：端口没有 USB 设备连接。	0
0	RB_UMS_SPLIT_CAN	RO	USB 主机模式下，SPLIT 包发送允许位： 1：允许发送 SPLIT 包； 0：禁止发送	0

22.2.1.8 USB 中断标志寄存器(R8_USB_INT_FG)

位	名称	访问	描述	复位值
7	保留	RO	保留	0
6	RB_UIF_ISO_ACT	RO	同步传输开始发送/接收数据中断标志位，写 1 清零： 1：开始发送/接收数据触发； 0：无事件。 *注：对于接收如果接收 CRC16 错误，将不会产生 UIF_TRANSFER，否则在事务完成后依然会产生 UIF_TRANSFER*	0
5	RB_ UIF_SETUP_ACT	RO	SETUP 事务完成中断标志位，写 1 清零： 1：SETUP 事务完成；	1

			0：无事件。	
4	RB_UIF_FIFO_OV	RW	USB FIFO 溢出中断标志位，写 1 清零： 1：FIFO 溢出触发； 0：无事件。	0
3	RB_UIF_HST_SOF	RW	USB 主机模式下 SOF 定时中断标志位，写 1 清零： 1：SOF 包传输完成触发；0：无事件。	0
2	RB_UIF_SUSPEND	RW	USB 总线挂起或唤醒事件中断标志位，写 1 清零： 1：USB 挂起事件或唤醒事件触发； 0：无事件。	0
1	RB_UIF_TRANSFER	RW	USB 传输完成中断标志位，写 1 清零： 1：一个 USB 传输完成触发； 0：无事件。	0
0	RB_UIF_DETECT	RW	USB 主机模式下 USB 设备连接或断开事件中断标志位，写 1 清零： 1：检测到 USB 设备连接或断开触发； 0：无事件。	0
	RB_UIF_BUS_RST	RW	USB 设备模式下 USB 总线复位事件中断标志位，写 1 清零： 1：USB 总线复位事件触发； 0：无事件。	0

22.2.1.9 USB 中断状态寄存器(R8_USB_INT_ST)

位	名称	访问	描述	复位值
7	RB_UIS_IS_NAK	RO	USB 设备模式下，NAK 响应状态位，同 RB_U_IS_NAK： 1：当前 USB 传输过程中回应 NAK； 0：无 NAK 响应。	0
6	RB_UIS_TOG_OK	RO	USB 事务接收完成后，接收到的数据包的 Toggle 与设置的期望值 匹配状态位： 1：toggle 匹配； 0：toggle 不匹配。	0
[5:4]	MASK_UIS_TOKEN	RO	设备模式下，当前 USB 传输事务的令牌 PID 标识。	XXb
[3:0]	MASK_UIS_ENDP	RO	设备模式下，当前 USB 传输事务的端点号。	XXXXb
	MASK_UIS_H_RES	RO	主机模式下，当前 USB 传输事务的应答 PID 标识，0000 表示设备无应答或超时；其它值表示应答 PID。	XXXXb

注：MASK_UIS_TOKEN 用于 USB 设备模式下标识当前 USB 传输事务的令牌 PID：00 表示 OUT 包；01 表示 SOF 包；10 表示 IN 包；11 表示 SETUP 包。

MASK_UIS_H_RES 仅在主机模式下有效。在主机模式下，若主机发送 OUT/SETUP 令牌包时，则该 PID 是握手包 ACK/NAK/STALL，或是设备无应答/超时。若主机发送 IN 令牌包，则该 PID 是数据包的 PID（DATA0/DATA1）或握手包 PID。

22.2.1.10 USB 接收长度寄存器(R16_USB_RX_LEN)

位	名称	访问	描述	复位值
[15:0]	R16_USB_RX_LEN	RO	当前 USB 端点接收的数据字节数	X

22.2.2 设备寄存器描述

USBHD 模块在 USB 设备模式下，提供了端点 0-15 共 30 组双向端点，除端点 0 之外的所有端点的最大数据包长度都是 1024 字节，端点 0 的最大数据包长度为 64 字节。

- 端点 0 是默认端点，支持控制传输，发送和接收共用一个 64 字节数据缓冲区。
- 端点 1-15 各自包括一个发送端点 IN 和一个接收端点 OUT，发送和接收各有一个独立的数据缓冲区，支持批量传输、中断传输和实时/同步传输。
- 端点 0 具有独立的 DMA 地址，收发共用，端点 1~15 的发送和接收各有一个 DMA 地址。通过入 R32_UEPn_BUF_MOD 寄存器可以设置数据缓冲区的模式为双缓冲或单缓冲。若使用双缓冲区模式，该端点只能使用单方向传输。
- 每组端点都具有收发控制寄存器 R8_UEPn_TX_CTRL、R8_UEPn_RX_CTRL 和发送长度寄存器 R16_UEPn_T_LEN 和 R32_UEPn_*_DMA（n=0~15），用于配置该端点的同步触发位、对 OUT 事务和 IN 事务的响应以及发送数据的长度等。

作为 USB 设备所必要的 USB 总线上拉电阻可以由软件随时设置是否启用，当 USB 控制寄存器 R8_USB_CTRL 中的 RB_UC_DEV_PU_EN 置 1 时，控制器根据 RB_UC_SPEED_TYPE 的速度设置，在内部为 USB 总线的 DP/DM 引脚连接上拉电阻，并启用 USB 设备功能。

当检测到 USB 总线复位、USB 总线挂起或唤醒事件，或当 USB 成功处理完数据发送或数据接收后，USB 协议处理器都将设置相应的中断标志，如果中断使能打开，还会产生相应的中断请求。应用程序可以直接查询或在 USB 中断服务程序中查询并分析中断标志寄存器 R8_USB_INT_FG，根据 RB_UIF_BUS_RST 和 RB_UIF_SUSPEND 进行相应的处理；并且，如果 RB_UIF_TRANSFER 有效，那么还需要继续分析 USB 中断状态寄存器 R8_USB_INT_ST，根据当前端点号 MASK_UIS_ENDP 和当前事务令牌 PID 标识 MASK_UIS_TOKEN 进行相应的处理。如果事先设定了各个端点的 OUT 事务的同步触发位 RB_UEP_R_TOG，那么可以通过 RB_U_TOG_OK 或 RB_UIS_TOG_OK 判断当前所接收到的数据包的同步触发位是否与该端点的同步触发位匹配，如果数据同步，则数据有效；如果数据不同步，则数据应该被丢弃。每次处理完 USB 发送或接收中断后，都应该正确修改相应端点的同步触发位，用于下次所发送的数据包或下次所接收的数据包是否同步检测；另外，设置 RB_UEP_T_TOG_AUTO 或 RB_UEP_R_TOG_AUTO 可以实现在发送成功或接收成功后自动修改相应的同步触发位（翻转或自减）。

各个端点准备发送的数据在各自的缓冲区中，准备发送的数据长度是独立设定在 R16_UEPn_T_LEN 中；各个端点接收到的数据在各自的缓冲区中，但是接收到的数据长度都在 USB 接收长度寄存器 R16_USB_RX_LEN 中，可以在 USB 接收中断时根据当前端点号区分。

表 22-2 设备相关寄存器列表

名称	访问地址	描述	复位值
R32_UEP_CONFIG	0x40023410	端点使能配置寄存器	00000000h
R32_UEP_TYPE	0x40023414	端点类型配置寄存器	00000000h
R32_UEP_BUF_MOD	0x40023418	端点缓冲区模式寄存器	00000000h
R32_UEP0_DMA	0x4002341C	端点 0 缓冲区的起始地址	xxxxh
R32_UEP1_RX_DMA	0x40023420	端点 1 接收缓冲区的起始地址	xxxxh
R32_UEP2_RX_DMA	0x40023424	端点 2 接收缓冲区的起始地址	xxxxh
R32_UEP3_RX_DMA	0x40023428	端点 3 接收缓冲区的起始地址	xxxxh
R32_UEP4_RX_DMA	0x4002342C	端点 4 接收缓冲区的起始地址	xxxxh
R32_UEP5_RX_DMA	0x40023430	端点 5 接收缓冲区的起始地址	xxxxh

R32_UEP6_RX_DMA	0x40023434	端点 6 接收缓冲区的起始地址	xxxxh
R32_UEP7_RX_DMA	0x40023438	端点 7 接收缓冲区的起始地址	xxxxh
R32_UEP8_RX_DMA	0x4002343C	端点 8 接收缓冲区的起始地址	xxxxh
R32_UEP9_RX_DMA	0x40023440	端点 9 接收缓冲区的起始地址	xxxxh
R32_UEP10_RX_DMA	0x40023444	端点 10 接收缓冲区的起始地址	xxxxh
R32_UEP11_RX_DMA	0x40023448	端点 11 接收缓冲区的起始地址	xxxxh
R32_UEP12_RX_DMA	0x4002344C	端点 12 接收缓冲区的起始地址	xxxxh
R32_UEP13_RX_DMA	0x40023450	端点 13 接收缓冲区的起始地址	xxxxh
R32_UEP14_RX_DMA	0x40023454	端点 14 接收缓冲区的起始地址	xxxxh
R32_UEP15_RX_DMA	0x40023458	端点 15 接收缓冲区的起始地址	xxxxh
R32_UEP1_TX_DMA	0x4002345C	端点 1 发送缓冲区的起始地址	xxxxh
R32_UEP2_TX_DMA	0x40023460	端点 2 发送缓冲区的起始地址	xxxxh
R32_UEP3_TX_DMA	0x40023464	端点 3 发送缓冲区的起始地址	xxxxh
R32_UEP4_TX_DMA	0x40023468	端点 4 发送缓冲区的起始地址	xxxxh
R32_UEP5_TX_DMA	0x4002346C	端点 5 发送缓冲区的起始地址	xxxxh
R32_UEP6_TX_DMA	0x40023470	端点 6 发送缓冲区的起始地址	xxxxh
R32_UEP7_TX_DMA	0x40023474	端点 7 发送缓冲区的起始地址	xxxxh
R32_UEP8_TX_DMA	0x40023478	端点 8 发送缓冲区的起始地址	xxxxh
R32_UEP9_TX_DMA	0x4002347C	端点 9 发送缓冲区的起始地址	xxxxh
R32_UEP10_TX_DMA	0x40023480	端点 10 发送缓冲区的起始地址	xxxxh
R32_UEP11_TX_DMA	0x40023484	端点 11 发送缓冲区的起始地址	xxxxh
R32_UEP12_TX_DMA	0x40023488	端点 12 发送缓冲区的起始地址	xxxxh
R32_UEP13_TX_DMA	0x4002348C	端点 13 发送缓冲区的起始地址	xxxxh
R32_UEP14_TX_DMA	0x40023490	端点 14 发送缓冲区的起始地址	xxxxh
R32_UEP15_TX_DMA	0x40023494	端点 15 发送缓冲区的起始地址	xxxxh
R16_UEP0_MAX_LEN	0x40023498	端点 0 最大长度包寄存器	xxxxh
R16_UEP1_MAX_LEN	0x4002349C	端点 1 最大长度包寄存器	xxxxh
R16_UEP2_MAX_LEN	0x400234A0	端点 2 最大长度包寄存器	xxxxh
R16_UEP3_MAX_LEN	0x400234A4	端点 3 最大长度包寄存器	xxxxh
R16_UEP4_MAX_LEN	0x400234A8	端点 4 最大长度包寄存器	xxxxh
R16_UEP5_MAX_LEN	0x400234AC	端点 5 最大长度包寄存器	xxxxh
R16_UEP6_MAX_LEN	0x400234B0	端点 6 最大长度包寄存器	xxxxh
R16_UEP7_MAX_LEN	0x400234B4	端点 7 最大长度包寄存器	xxxxh
R16_UEP8_MAX_LEN	0x400234B8	端点 8 最大长度包寄存器	xxxxh
R16_UEP9_MAX_LEN	0x400234BC	端点 9 最大长度包寄存器	xxxxh
R16_UEP10_MAX_LEN	0x400234C0	端点 10 最大长度包寄存器	xxxxh
R16_UEP11_MAX_LEN	0x400234C4	端点 11 最大长度包寄存器	xxxxh
R16_UEP12_MAX_LEN	0x400234C8	端点 12 最大长度包寄存器	xxxxh
R16_UEP13_MAX_LEN	0x400234CC	端点 13 最大长度包寄存器	xxxxh
R16_UEP14_MAX_LEN	0x400234D0	端点 14 最大长度包寄存器	xxxxh
R16_UEP15_MAX_LEN	0x400234D4	端点 15 最大长度包寄存器	xxxxh
R16_UEP0_T_LEN	0x400234D8	端点 0 发送长度寄存器	xxxxh
R8_UEP0_TX_CTRL	0x400234DA	端点 0 发送控制寄存器	00h
R8_UEP0_RX_CTRL	0x400234DB	端点 0 接收控制寄存器	00h

R16_UEP1_T_LEN	0x400234DC	端点 1 发送长度寄存器	xxxxh
R8_UEP1_TX_CTRL	0x400234DE	端点 1 发送控制寄存器	00h
R8_UEP1_RX_CTRL	0x400234DF	端点 1 接收控制寄存器	00h
R16_UEP2_T_LEN	0x400234E0	端点 2 发送长度寄存器	xxxxh
R8_UEP2_TX_CTRL	0x400234E2	端点 2 发送控制寄存器	00h
R8_UEP2_RX_CTRL	0x400234E3	端点 2 接收控制寄存器	00h
R16_UEP3_T_LEN	0x400234E4	端点 3 发送长度寄存器	xxxxh
R8_UEP3_TX_CTRL	0x400234E6	端点 3 发送控制寄存器	00h
R8_UEP3_RX_CTRL	0x400234E7	端点 3 接收控制寄存器	00h
R16_UEP4_T_LEN	0x400234E8	端点 4 发送长度寄存器	xxxxh
R8_UEP4_TX_CTRL	0x400234EA	端点 4 发送控制寄存器	00h
R8_UEP4_RX_CTRL	0x400234EB	端点 4 接收控制寄存器	00h
R16_UEP5_T_LEN	0x400234EC	端点 5 发送长度寄存器	xxxxh
R8_UEP5_TX_CTRL	0x400234EE	端点 5 发送控制寄存器	00h
R8_UEP5_RX_CTRL	0x400234EF	端点 5 接收控制寄存器	00h
R16_UEP6_T_LEN	0x400234F0	端点 6 发送长度寄存器	xxxxh
R8_UEP6_TX_CTRL	0x400234F2	端点 6 发送控制寄存器	00h
R8_UEP6_RX_CTRL	0x400234F3	端点 6 接收控制寄存器	00h
R16_UEP7_T_LEN	0x400234F4	端点 7 发送长度寄存器	xxxxh
R8_UEP7_TX_CTRL	0x400234F6	端点 7 发送控制寄存器	00h
R8_UEP7_RX_CTRL	0x400234F7	端点 7 接收控制寄存器	00h
R16_UEP8_T_LEN	0x400234F8	端点 8 发送长度寄存器	xxxxh
R8_UEP8_TX_CTRL	0x400234FA	端点 8 发送控制寄存器	00h
R8_UEP8_RX_CTRL	0x400234FB	端点 8 接收控制寄存器	00h
R16_UEP9_T_LEN	0x400234FC	端点 9 发送长度寄存器	xxxxh
R8_UEP9_TX_CTRL	0x400234FE	端点 9 发送控制寄存器	00h
R8_UEP9_RX_CTRL	0x400234FF	端点 9 接收控制寄存器	00h
R16_UEP10_T_LEN	0x40023500	端点 10 发送长度寄存器	xxxxh
R8_UEP10_TX_CTRL	0x40023502	端点 10 发送控制寄存器	00h
R8_UEP10_RX_CTRL	0x40023503	端点 10 接收控制寄存器	00h
R16_UEP11_T_LEN	0x40023504	端点 11 发送长度寄存器	xxxxh
R8_UEP11_TX_CTRL	0x40023506	端点 11 发送控制寄存器	00h
R8_UEP11_RX_CTRL	0x40023507	端点 11 接收控制寄存器	00h
R16_UEP12_T_LEN	0x40023508	端点 12 发送长度寄存器	xxxxh
R8_UEP12_TX_CTRL	0x4002350A	端点 12 发送控制寄存器	00h
R8_UEP12_RX_CTRL	0x4002350B	端点 12 接收控制寄存器	00h
R16_UEP13_T_LEN	0x4002350C	端点 13 发送长度寄存器	xxxxh
R8_UEP13_TX_CTRL	0x4002350E	端点 13 发送控制寄存器	00h
R8_UEP13_RX_CTRL	0x4002350F	端点 13 接收控制寄存器	00h
R16_UEP14_T_LEN	0x40023510	端点 14 发送长度寄存器	xxxxh
R8_UEP14_TX_CTRL	0x40023512	端点 14 发送控制寄存器	00h
R8_UEP14_RX_CTRL	0x40023513	端点 14 接收控制寄存器	00h
R16_UEP15_T_LEN	0x40023514	端点 15 发送长度寄存器	xxxxh

R8_UEP15_TX_CTRL	0x40023516	端点 15 发送控制寄存器	00h
R8_UEP15_RX_CTRL	0x40023517	端点 15 接收控制寄存器	00h

22.2.2.1 USB 端点配置寄存器（R32_UEP_CONFIG）

位	名称	访问	描述	复位值
[31:17]	RB_UEP_R_EN	RW	端点 0-15 接收使能	0
16	保留	RO	保留	0
[15:1]	RB_UEP_T_EN	RW	端点 0-15 发送使能	0
0	保留	RO	保留	0

注：端点 0 的收发使能信号始终有效。

22.2.2.2 USB 端点类型控制寄存器（R32_UEP_TYPE）

位	名称	访问	描述	复位值
[31:15]	RB_UEP_R_TYPE	RW	端点 0-15,OUT 方向传输类型，1 表示同步传输,	0
0	保留	RO	保留	0
[15:1]	RB_UEP_T_TYPE	RW	端点 0-15,IN 方向传输类型，1 表示同步传输	0
0	保留	RO	保留	0

22.2.2.3 USB 端点缓冲区模式控制寄存器（R32_UEP_BUF_MOD）

位	名称	访问	描述	复位值
[31:16]	RB_UEP_ISO_BUF_MOD	RW	同步端点缓冲区模式控制位，1 有效。	0
[15:0]	RB_UEP_BUF_MOD	RW	端点缓冲区模式控制位	0

注：当 RB_UEP_ISO_BUF_MOD 为 1 时，对于同步 IN 端点，在接收到 SOF 包后，硬件会有以下操作：将 EPx_R_TOG 内容加载到 EPx_T_TOG 中；将 EPx_MAX_LEN 值加载到 EPx_T_LEN 中；将 UEPn_RX_DMA 值加载到 UEPn_TX_DMA 中。

当 RB_UEP_ISO_BUF_MOD 为 1 时，对于同步 OUT 端点，在接收到 SOF 包后，硬件会有以下操作：将 EPx_T_TOG 内容加载到 EPx_R_TOG 中；将 UEPn_TX_DMA 的值加载到 UEPn_RX_DMA 中。

表 21-3 端点 n 缓冲区模式（n=1-15）

UEPn_RX_EN	UEPn_TX_EN	UEPn_BUF_MOD	描述：以 UEPn_DMA 为起始地址由低向高排列
0	0	x	端点被禁用，未用到 UEPn_*_DMA 缓冲区。
1	0	0	接收（OUT）缓冲区首地址为 UEPn_RX_DMA
1	0	1	RB_UEPn_RX_TOG[0]=0,使用缓冲区 UEPn_RX_DMA RB_UEPn_RX_TOG[0]=1,使用缓冲区 UEPn_TX_DMA
0	1	0	发送(IN)缓冲区首地址为 UEPn_TX_DMA。
0	1	1	RB_UEPn_TX_TOG[0]=0,使用缓冲区 UEPn_TX_DMA RB_UEPn_TX_TOG[0]=1,使用缓冲区 UEPn_RX_DMA

端点 n 缓冲区起始地址(R32_UEP0_DMA)

位	名称	访问	描述	复位值
[31:17]	保留	RO	保留	0

[16:0]	R32_UEPn_DMA	RW	端点 0 缓冲区起始地址。 低 16 位有效，地址必须 4 字节对齐。	X

USB 端点 n 发送缓冲区起始地址（R32_UEPn_TX_DMA）（n=1-15）

位	名称	访问	描述	复位值
[31:17]	保留	RO	保留	0
[16:0]	R32_UEPn_TX_DMA	RW	端点 n 发送缓冲区起始地址。 低 16 位有效，地址必须 4 字节对齐。	X

USB 端点 n 接收缓冲区起始地址（R32_UEPn_RX_DMA）（n=1-15）

位	名称	访问	描述	复位值
[16:0]	R32_UEPn_RX_DMA	RW	端点 n 接收缓冲区起始地址。 低 16 位有效，地址必须 4 字节对齐。	X

22.2.2.4 端点 n 最大长度包寄存器（R16_UEPn_MAX_LEN）（n=0-15）

位	名称	访问	描述	复位值
[15:11]	保留	RO	保留	0
[10:0]	UEPn_MAX_LEN	RW	端点 n 接收数据的最大包长度。	xxxxh

注：这个最大包长度决定了端点可接收数据最大长度，超出此长度的数据会被丢弃，不会写入缓冲区。

22.2.2.5 端点 n 发送长度寄存器(R16_UEPn_T_LEN)（n=0-15）

位	名称	访问	描述	复位值
[10:0]	UEPn_T_LEN	RW	设置 USB 端点 n 准备发送的数据字节数，对于控制端点（0），低 7 位有效。	X

22.2.2.6 端点 n 发送控制寄存器（R8_UEPn_TX_CTRL）（n=0-15）

位	名称	访问	描述	复位值
[7:6]	保留	RO	保留	0
5	RB_UEP_T_TOG_AUTO	RW	同步触发位自动翻转使能控制位，软件可修改： 1：对于非同步端点，数据发送成功后自动翻转 MASK_UEP_T_TOG [0]，对于同步端点，数据发送成功后 MASK_UEP_T_TOG 自动减 1 0：不自动翻转，可以手动切换。 *注：端点 0 此位为保留位。*	0
[4:3]	MASK_UEP_T_TOG	RW	USB 端点 n 的发送器（处理 IN 事务）准备的同步触发位： 00：发送 DATA0； 01：发送 DATA1； 10：发送 DATA2； 11：发送 MDATA。	0

2	保留	RO	保留	00b
[1:0]	MASK_UEP_T_RES	RW	端点 n 的发送器对 IN 事务的响应控制： 00：数据就绪并期望 ACK； 10：应答 NAK 或忙； 11：应答 STALL 或错误。	00b

22.2.2.7 端点 n 接收控制寄存器（R8_UEPn_RX_CTRL）（n=0-15）

位	名称	访问	描述	复位值
[7:6]	保留	RO	保留	0
5	RB_UEP_R_TOG_AUTO	RO	同步触发位自动翻转使能控制位： 1：数据接收成功后自动翻转 MASK_UEP_R_TOG [0] 0：不自动翻转，可以手动切换。 *注：端点 0 此位为保留位。*	0
[4:3]	MASK_UEP_R_TOG	RW	USB 端点 n 的接收器（处理 OUT 事务）准备的同步触发位： 00：期望 DATA0； 01：期望 DATA1； 10：期望 DATA2； 11：期望 MDATA。 对于实时/同步传输无效。	0
2	保留	RO	保留	00b
[1:0]	MASK_UEP_R_RES	RW	端点 n 的接收器对 OUT 事务的响应控制： 00：数据就绪并期望 ACK； 10：应答 NAK 或忙； 11：应答 STALL 或错误； 01：应答 NYET。 对于实时/同步传输无效。	00b

22.2.3 USB 主机寄存器

在 USB 主机模式下，芯片提供了一组双向主机端点，包括一个发送端点 OUT 和一个接收端点 IN，一个数据包的最大长度是 1024 字节（同步传输），支持控制传输、中断传输、批量传输和实时/同步传输。

主机端点发起的每一个 USB 事务，在处理结束后总是自动设置 RB_UIF_TRANSFER 中断标志。应用程序可以直接查询或在 USB 中断服务程序中查询并分析中断标志寄存器 R8_USB_INT_FG，根据各中断标志分别进行相应的处理；并且，如果 RB_UIF_TRANSFER 有效，那么还需要继续分析 USB 中断状态寄存器 R8_USB_INT_ST，根据当前 USB 传输事务的应答 PID 标识 MASK_UIS_H_RES 进行相应的处理。

如果事先设定了主机接收端点的 IN 事务的同步触发位（RB_UH_R_TOG），那么可以通过 RB_U_TOG_OK 或 RB_UIS_TOG_OK 判断当前所接收到的数据包的同步触发位是否与主机接收端点的同步触发位匹配，如果数据同步，则数据有效；如果数据不同步，则数据应该被丢弃。每次处理完 USB 发送或接收中断后，都应该正确修改相应主机端点的同步触发位，用于同步下次所发送的数据包和检测下次所接收的数据包是否同步；另外，通过设置 RB_UH_T_AUTO_TOG 和 RB_UH_R_AUTO_TOG 可以实现在发送成功或接收成功后自动翻转相应的同步触发位。

USB 主机令牌设置寄存器 R8_UH_EP_PID 用于设置被操作的目标设备的端点号和本次 USB 传输事务的令牌 PID 包标识。SETUP 令牌和 OUT 令牌所对应的数据由主机发送端点提供，准备发送的数据在 R16_UH_TX_DMA 缓冲区中，准备发送的数据长度设置在 R16_UH_TX_LEN 中；IN 令牌所对应数据由目标设备返回给主机接收端点，接收到数据存放 R16_UH_RX_DMA 缓冲区中，接收到的数据长度存放在

R16_USB_RX_LEN 中。

表 22-3 主机相关寄存器列表

名称	访问地址	描述	复位值
R8_UHOST_CTRL	0x40023401	USB 主机控制寄存器	00h
R32_UH_CONFIG	0x40023410	USB 主机端点配置寄存器	00000000h
R32_UH_EP_TYPE	0x40023414	USB 主机端点类型寄存器	00000000h
R32_UH_RX_DMA	0x40023424	USB 主机接收缓冲区起始地址	16hxxxx
R32_UH_TX_DMA	0x40023464	USB 主机发送缓冲区起始地址	16hxxxx
R16_UH_RX_MAX_LEN	0x400234A0	USB 主机接收最大长度包寄存器	16hxxxx
R8_UH_EP_PID	0x400234E0	USB 主机令牌设置寄存器	8h00
R8_UH_RX_CTRL	0x400234E3	USB 主机接收端点控制寄存器	8h00
R16_UH_TX_LEN	0x400234E4	USB 主机发送长度寄存器	16hxxxx
R8_UH_TX_CTRL	0x400234E6	USB 主机发送端点控制寄存器	8h00
R16_UH_SPLIT_DATA	0x400234E8	USB 主机发送 SPLIT 包的数据	16hxxxx

22.2.3.1 USB 主机控制寄存器(R8_UHOST_CTRL)

位	名称	访问	描述	复位值
7	RB_UH_SOF_EN	RW	自动产生 SOF 包使能控制位： 1：主机自动发生 SOF 包； 0：不产生 SOF 包。 该位在从连接状态变为断开状态时由硬件自动清零	0
6	RB_UH_SOF_FREE	RO	总线空闲	0
5	保留	RO	保留。	0
4	RB_UH_PHY_SUSPENDM	RW	USB-PHY 的处于挂起状态，内部 USB-PLL 将被关闭，低有效	0
3	RB_UH_REMOTE_WKUP	RW	远程唤醒。	0
2	RB_UH_TX_BUS_RESUME	RW	主机模式下，表示主机唤醒设备，软件拉高 50ns 后，硬件自动发送 30ms 的唤醒信号。	0
1	RB_UH_TX_BUS_SUSPEND	RW	USB 主机发送挂起信号，需由软件拉高 10ms。	0
0	RB_UH_TX_BUS_RESET	RW	USB 主机发送总线复位信号，需由软件拉高 10ms。	0

注：复位的时间由 RB_UH_TX_BUS_RESET 的高电平持续时间决定（建议至少 10ms，10ms 后直接查询速度类型）。如果主机唤醒设备， bUH_TX_BUS_RESUME 拉高之后，硬件自动发送 30ms 的唤醒信号（K），bUH_TX_BUS_RESUME 需要手动清除，以免影响下一次主机挂起（bUH_TX_BUS_RESUME 高电平至少维持 50ns）。

22.2.3.2 USB 主机端点配置控制寄存器(R32_UH_CONFIG)

位	名称	访问	描述	复位值
[31：19]	保留	RO	保留	0
18	RB_UH_EP_RX_EN	RW	主机接收使能	0
[17:4]	保留	RO	保留。	0

3	RB_UH_EP_TX_EN	RW	主机发送使能	0
[2:0]	保留	RO	保留。	0

22.2.3.3 USB 主机端点类型寄存器(R32_UH_EP_TYPE)

位	名称	访问	描述	复位值
[31:19]	保留	RO	保留	0
[18]	RB_UH_EP_RX_TYPE	RW	主机接收端点类型，1 表示同步传输	0
[17:4]	保留	RO	保留。	0
[3]	RB_UH_EP_TX_TYPE	RW	主机发送端点类型，1 表示同步传输	0
[2:0]	保留	RO	保留。	0

USB 主机接收缓冲区起始地址(R32_UH_RX_DMA)

位	名称	访问	描述	复位值
[31:17]	保留	RO	保留	0
[16:0]	R16_UH_RX_DMA	RW	主机端点数据接收缓冲区起始地址，最低 2 位固定为 0（4 字节对齐）	X

USB 主机发送缓冲区起始地址(R32_UH_TX_DMA)

位	名称	访问	描述	复位值
[31:17]	保留	RO	保留	0
[16:0]	R16_UH_TX_DMA	RW	主机端点数据发送缓冲区起始地址（不需要 4 字节对齐）	X

22.2.3.4 USB 主机接收最大长度包寄存器（R16_UH_RX_MAX_LEN）

位	名称	访问	描述	复位值
[15:11]	保留	RO	保留	0
[10:0]	UH_RX_MAX_LEN	RW	主机端点接收数据的最大包长度。	xxxxh

注：这个最大包大小决定了端点可接收数据最大长度，超出此长度的数据会被丢弃。

22.2.3.5 USB 主机令牌设置寄存器(R8_UH_EP_PID)

位	名称	访问	描述	复位值
[7:4]	MASK_UH_TOKEN	RW	设置本次 USB 传输事务的令牌 PID 标识。	0
[3:0]	MASK_UH_ENDP	RW	设置本次被操作的目标设备的端点号。	0

22.2.3.6 USB 主机接收端点控制寄存器(R8_UH_RX_CTRL)

位	名称	访问	描述	复位值
7	保留	RO	保留。	0
6	RB_UH_R_DATA_NO	RW	1：不期待数据包，用于主机模式下的操作高速 HUB； 0：期望数据包（IN）。	0
5	RB_UH_R_AUTO_TOG	RW	同步触发位自动翻转使能控制位： 1：对于非同步传输，数据接收成功后自	0

			动翻转相应 MASK_UH_R_TOG[0]；对于同步传输，数据接收成功后 MASK_UH_R_TOG 会自动减 1。 0：不自动翻转，可以手动切换。	
[4:3]	MASK_UH_R_TOG	RW	主机接收器（处理 IN 事务）期望的同步触发位， 00：期望 DATA0； 01：期望 DATA1； 10：期望 DATA2； 11：期望 MDATA。	0
2	RB_UH_R_RES_NO	RW	1：无应答，用于实现非端点 0 的实时/同步传输。此时忽略 MASK_UEP_R_RES； 0：接收数据成功后发送应答。	0
[1:0]	MASK_UH_R_RES	RW	主机接收器对 IN 事务的响应控制位： 00：应答 ACK； 对于实时/同步传输无效。	0

22.2.3.7 USB 主机发送长度寄存器(R16_UH_TX_LEN)

位	名称	访问	描述	复位值
[15:11]	保留	RO	保留	0
[10:0]	R16_UH_TX_LEN	RW	设置 USB 主机发送端点准备发送的数据字节。	X

22.2.3.8 USB 主机发送端点控制寄存器(R8_UH_TX_CTRL)

位	名称	访问	描述	复位值
7	保留	RO	保留。	0
6	RB_UH_T_DATA_NO	RW	1：不发送数据包(PING/SPLIT)； 0：发送数据包（OUT/SETUP）。	0
5	RB_UH_T_AUTO_TOG	RW	同步触发位自动翻转使能控制位，可软件修改： 1：对于非同步传输，数据发送成功后自动翻转 MASK_UH_T_TOG[0]。 0：不自动翻转，可以手动切换。	0
[4:3]	MASK_UH_T_TOG	RW	USB 主机发送器（处理 SETUP/OUT 事务）准备的同步触发位 00 表示发送 DATA0； 01 表示发送 DATA1； 10 表示发送 DATA2； 11 表示发送 MDATA。	0
2	RB_UH_T_RES_NO	RW	1：无应答，用于实现非端点 0 的实时/同步传输。此时忽略 MASK_UEP_T_RES； 0：发送数据成功后期待应答。	0
[1:0]	MASK_UH_T_RES	RW	USB 主机发送器对 SETUP/OUT 事务的响应控制位 00：期望应答 ACK；	0

			10：期望应答 NAK 或忙； 11：期望应答 STALL 或错误； 01：期望应答 NYET 。 对于实时/同步传输无效。	

USB 主机发送 SPLIT 包的数据（R16_UH_SPLIT_DATA）

位	名称	访问	描述	复位值
[15:12]	保留	RO	保留	0
[11:0]	UH_SPLIT_DATA	RW	主机端点发送 SPLIT 包的数据内容，低 12 位有效，高 4 位固定为 0	0xxxh

第 23 章 USB OTG 全速控制器(OTG_FS)

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器系列部分产品。

23.1 USB 控制器简介

芯片内嵌 USB 控制器及收发器，特性如下：

- 双重角色设备控制器，支持 USB Host 主机功能和 USB Device 设备功能。
- 遵循 On-The-GoSupplement to the USB2.0 规范，主机和设备模式均支持 USB2.0 全速 12Mbps 或低速 1.5Mbps。
- 支持软件 HNP 和 SRP 协议。
- 支持 USB 控制传输、批量传输、中断传输、同步/实时传输。
- 支持最大 64 字节的数据包，内置 FIFO，支持中断和 DMA。

23.2 寄存器描述

USB 相关寄存器分为 3 个部分，部分寄存器是在主机和设备模式下进行复用的。

- USB 全局寄存器
- USB 设备控制寄存器
- USB 主机控制寄存器

23.2.1 全局寄存器描述

表 23-1 USB OTG 相关寄存器列表

名称	访问地址	描述	复位值
R8_USB_CTRL	0x50000000	USB 控制寄存器	0x06
R8_USB_INT_EN	0x50000002	USB 中断使能寄存器	0x00
R8_USB_DEV_AD	0x50000003	USB 设备地址寄存器	0x00
R32_USB_STATUS	0x50000004	USB 状态寄存器	0xXX20XXXX
R8_USB_MIS_ST	0x50000005	USB 杂项状态寄存器	0xXX
R8_USB_INT_FG	0x50000006	USB 中断标志寄存器	0x20
R8_USB_INT_ST	0x50000007	USB 中断状态寄存器	0xXX
R16_USB_RX_LEN	0x50000008	USB 接收长度寄存器	0xXX
R32_USB_OTG_CR	0x50000054	USB OTG 控制寄存器	0x00
R32_USB_OTG_SR	0x50000058	USB OTG 状态寄存器	0x00

23.2.1.1 USB 控制寄存器(R8_USB_CTRL)

位	名称	访问	描述	复位值
7	RB_UC_HOST_MODE	RW	USB 工作模式选择位： 1：主机模式（HOST）； 0：设备模式（DEVICE）。	0
6	RB_UC_LOW_SPEED	RW	USB 总线信号传输速率选择位： 1：1.5Mbps；　　0：12Mbps。	0
5	RB_UC_DEV_PU_EN	RW	USB 设备模式下,USB 设备使能和内部上拉电阻控制位，为 1 则使能 USB 设备传输并且启用内部上拉电阻。	0

[5:4]	MASK_UC_SYS_CTRL	RW	见下表配置 USB 系统。	0
3	RB_UC_INT_BUSY	RW	USB 传输完成中断标志未清零前自动暂停使能位： 1：在中断标志 UIF_TRANSFER 未清零前自动暂停，设备模式下自动应答忙 NAK，主机模式下自动暂停后续传输； 0：不暂停。	0
2	RB_UC_RESET_SIE	RW	USB 协议处理器软件复位控制位： 1：强制复位 USB 协议处理器（SIE），需要软件清零； 0：不复位。	1
1	RB_UC_CLR_ALL	RW	USB 的 FIFO 和中断标志清零： 1：强制清空和清零；　0：不清。	1
0	RB_UC_DMA_EN	RW	USB 的 DMA 和 DMA 中断控制位： 1：使能 DMA 功能和 DMA 中断； 0：关闭 DMA。	0

由 RB_UC_HOST_MODE 和 MASK_UC_SYS_CTRL 组成 USB 系统控制组合：

表 23-2 USB 系统控制组合

RB_UC_HOST_MODE	MASK_UC_SYS_CTRL	USB 系统控制描述
0	00	禁止 USB 设备功能，关闭内部上拉电阻。
0	01	使能 USB 设备功能，关闭内部上拉电阻，需加外部上拉。
0	1x	使能 USB 设备功能，启用内部 1.5K 上拉电阻。该上拉电阻优先于下拉电阻，也可用于 GPIO 模式。
1	00	USB 主机模式，正常工作状态。
1	01	USB 主机模式，强制 DP/DM 输出 SE0 状态。
1	10	USB 主机模式，强制 DP/DM 输出 J 状态。
1	11	USB 主机模式，强制 DP/DM 输出 K 状态/唤醒。

23.2.1.2 USB 中断使能寄存器(R8_USB_INT_EN)

位	名称	访问	描述	复位值
7	RB_UIE_DEV_SOF	RW	USB 设备模式，接收 SOF 包中断： 1：使能中断；　0：禁止中断。	0
6	RB_UIE_DEV_NAK	RW	USB 设备模式，接收到 NAK 中断： 1：使能中断；　0：禁止中断。	0
5	Reserved	RO	保留。	0
4	RB_UIE_FIFO_OV	RW	FIFO 溢出中断： 1：使能中断；　0：禁止中断。	0
3	RB_UIE_HST_SOF	RW	USB 主机模式，SOF 定时中断： 1：使能中断；　0：禁止中断。	0
2	RB_UIE_SUSPEND	RW	USB 总线挂起或唤醒事件中断： 1：使能中断；　0：禁止中断。	0
1	RB_UIE_TRANSFER	RW	USB 传输完成中断： 1：使能中断；　0：禁止中断。	0

0	RB_UIE_DETECT	RW	USB 主机模式，USB 设备连接或断开事件中断： 1：使能中断；　　0：禁止中断。	0
	RB_UIE_BUS_RST	RW	USB 设备模式，USB 总线复位事件中断： 1：使能中断；　　0：禁止中断。	0

23.2.1.3 USB 设备地址寄存器(R8_USB_DEV_AD)

位	名称	访问	描述	复位值
7	RB_UDA_GP_BIT	RW	USB 通用标志位，用户自定义。	0
[6:0]	MASK_USB_ADDR	RW	主机模式：当前操作的 USB 设备地址； 设备模式：该 USB 自身地址。	0

23.2.1.4 USB 杂项状态寄存器(R8_USB_MIS_ST)

位	名称	访问	描述	复位值
7	RB_UMS_SOF_PRES	RO	USB 主机模式下 SOF 包预示状态位： 1：将要发送 SOF 包，此时如有其它 USB 数据包将被自动延后； 0：无 SOF 包发送。	X
6	RB_UMS_SOF_ACT	RO	USB 主机模式下 SOF 包传输状态位： 1：正在发出 SOF 包； 0：发送完成或者空闲。	X
5	RB_UMS_SIE_FREE	RO	USB 协议处理器的空闲状态位： 1：协议器空闲； 0：忙，正在进行 USB 传输。	1
4	RB_UMS_R_FIFO_RDY	RO	USB 接收 FIFO 数据就绪状态位： 1：接收 FIFO 非空； 0：接收 FIFO 为空。	0
3	RB_UMS_BUS_RESET	RO	USB 总线复位状态位： 1：当前 USB 总线处于复位态； 0：当前 USB 总线处于非复位态。	X
2	RB_UMS_SUSPEND	RO	USB 挂起状态位： 1：USB 总线处于挂起态，有一段时间没有 USB 活动； 0：USB 总线处于非挂起态。	0
1	RB_UMS_DM_LEVEL	RO	USB 主机模式下，设备刚连入 USB 端口时 DM 引脚的电平状态，用于判断速度： 1：高电平/低速； 0：低电平/全速。	0
0	RB_UMS_DEV_ATTACH	RO	USB 主机模式下端口的 USB 设备连接状态位： 1：端口已经连接 USB 设备； 0：端口没有 USB 设备连接。	0

23.2.1.5 USB 中断标志寄存器(R8_USB_INT_FG)

位	名称	访问	描述	复位值
7	RB_U_IS_NAK	RO	USB 设备模式下，NAK 响应状态位： 1：当前 USB 传输过程中回应 NAK； 0：无 NAK 响应。	0
6	RB_U_TOG_OK	RO	当前 USB 传输 DATA0/1 同步标志匹配状态位： 1：同步；0：不同步。	0
5	RB_U_SIE_FREE	RO	USB 协议处理器空闲状态位： 1：USB 空闲； 0：忙，正在进行 USB 传输。	1
4	RB_UIF_FIFO_OV	RW	USB FIFO 溢出中断标志位，写 1 清零： 1：FIFO 溢出触发；　0：无事件。	0
3	RB_UIF_HST_SOF	RW	USB 主机模式下 SOF 定时中断标志位，写 1 清零： 1：SOF 包传输完成触发；0：无事件。	0
2	RB_UIF_SUSPEND	RW	USB 总线挂起或唤醒事件中断标志位，写 1 清零： 1：USB 挂起事件或唤醒事件触发； 0：无事件。	0
1	RB_UIF_TRANSFER	RW	USB 传输完成中断标志位，写 1 清零： 1：一个 USB 传输完成触发；0：无事件。	0
0	RB_UIF_DETECT	RW	USB 主机模式下 USB 设备连接或断开事件中断标志位，写 1 清零： 1：检测到 USB 设备连接或断开触发； 0：无事件。	0
	RB_UIF_BUS_RST	RW	USB 设备模式下 USB 总线复位事件中断标志位，写 1 清零： 1：USB 总线复位事件触发； 0：无事件。	0

23.2.1.6 USB 中断状态寄存器(R8_USB_INT_ST)

位	名称	访问	描述	复位值
7	RB_UIS_IS_NAK	RO	USB 设备模式下，NAK 响应状态位，同 RB_U_IS_NAK： 1：当前 USB 传输过程中回应 NAK； 0：无 NAK 响应。	0
6	RB_UIS_TOG_OK	RO	当前 USB 传输 DATA0/1 同步标志匹配状态位，同 RB_U_TOG_OK： 1：同步；0：不同步。	0
[5:4]	MASK_UIS_TOKEN	RO	设备模式下，当前 USB 传输事务的令牌 PID 标识。	XXb
[3:0]	MASK_UIS_ENDP	RO	设备模式下，当前 USB 传输事务的端点号。	XXXXb
	MASK_UIS_H_RES	RO	主机模式下，当前 USB 传输事务的应答	XXXXb

			PID 标识，0000 表示设备无应答或超时；其它值表示应答 PID。	

MASK_UIS_TOKEN 用于 USB 设备模式下标识当前 USB 传输事务的令牌 PID：00 表示 OUT 包；01 表示 SOF 包；10 表示 IN 包；11 表示 SETUP 包。

MASK_UIS_H_RES 仅在主机模式下有效。在主机模式下，若主机发送 OUT/SETUP 令牌包时，则该 PID 是握手包 ACK/NAK/STALL，或是设备无应答/超时。若主机发送 IN 令牌包，则该 PID 是数据包的 PID（DATA0/DATA1）或握手包 PID。

23.2.1.7 USB 接收长度寄存器(R16_USB_RX_LEN)

位	名称	访问	描述	复位值
[9:0]	R16_USB_RX_LEN	RO	当前 USB 端点接收的数据字节数	X

23.2.1.8 USB OTG 控制寄存器(R32_USB_OTG_CR)

位	名称	访问	描述	复位值
[31:6]	保留	RO	保留	0
5	RB_CR_SESS_VTH	RW	OTG 会话有效阈值电压设置 1：SESS_VLD 电平为 1.4V； 0：SESS_VLD 电平为 0.8V。	0
4	RB_CR_VBUS_VTH	RW	OTG VBUS 阈值电压设置 1：VBUS_VLD 电平为 4.4V； 0：VBUS_VLD 电平为 4.8V。	0
3	RB_CR_OTG_EN	RW	OTG 功能使能 1：使能；0：禁止	0
2	RB_CR_IDPU	RW	USB_OTG_ID 引脚上拉使能 1：使能；0：禁止	0
1	RB_CR_CHARGE_VBUS	RW	OTG VBUS 充电使能 1：使能；0：禁止	0
0	RB_CR_DISCHAR_VBUS	RW	OTG VBUS 放电使能 1：使能；0：禁止	0

注：此寄存器仅适用于 CH32V305、CH32V307、CH32F205 和 CH32F207

23.2.1.9 USB OTG 控制寄存器(R32_USB_OTG_SR)

位	名称	访问	描述	复位值
[31:4]	保留	RO	保留	0
3	RB_SR_ID_DIG	RO	OTG ID 标志 1：B 设备；0：A 设备。	0
2	RB_SR_SESS_END	RO	OTG 会话结束有效标志 1：有效；0：无效	0
1	RB_SR_SESS_VLD	RO	OTG 会话有效标志 1：有效，会话有效电平大于阈值电压 0：无效，会话有效电平小于阈值电压	0
0	RB_SR_VBUS_VLD	RO	OTG VBUS 输入电平 1：VBUS 电压大于阈值电压； 0：VBUS 电压小于阈值电压。	X

注：此寄存器仅适用于 CH32V305、CH32V307、CH32F205 和 CH32F207

23.2.2 设备寄存器描述

USB OTG 模块在 USB 设备模式下，提供了端点号 0-7 共 8 组双向端点配置寄存器，可映射端点号 8-15 的配置，除 3 号端点外所有端点最大包长度为 64 字节，3 号端点的最大数据包长度都是 1023 字节（同步传输）。

- 端点 0 是默认端点，支持控制传输，发送和接收共用一个 64 字节数据缓冲区
- 端点 1-15，可配置独立的 64 字节发送和接收缓冲区或者双 64 字节数据缓冲区，支持批量传输、中断传输和实时/同步传输。

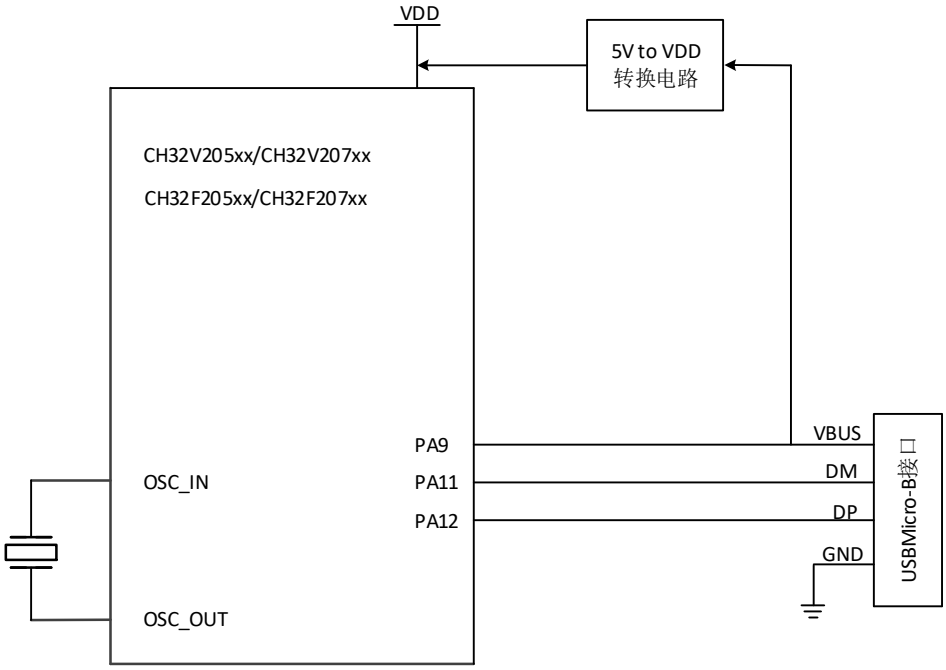
每组端点都具有一个控制寄存器 R8_UEPn_CTRL 和发送长度寄存器 R16_UEPn_T_LEN，用于设定该端点的同步触发位、对 OUT 事务和 IN 事务的响应以及发送数据的长度等。

USB OTG 模块主机和设备的角色由 OTG_FS_ID 引脚状态决定，当 OTG_FS_ID 引脚悬空，其内置的上拉电阻会使得 USB OTG 状态寄存器 R32_USB_OTG_SR 的 RB_SR_ID_DIG 位置 1，此时控制器应初始化为 B 设备。当 OTG_FS_ID 引脚接地，此时 USB OTG 状态寄存器 R32_USB_OTG_SR 的 RB_SR_ID_DIG 位为 0，控制器应初始化为 A 设备。

B 设备在开始会话根据 OTG 状态寄存器 R32_USB_OTG_SR 的 RB_SR_SESS_VLD 来确保 VBUS 的电平低于 $V_{SESS_VLD,min}$，若该位为 0，则可开始新的会话。若该位为 1，则可将 OTG 控制寄存器 R32_USB_OTG_CR 的 RB_CR_DISCHAR_VBUS 位置 1 进行放电，使得 VBUS 小于会话阈值电平。

作为 B 类设备时，需从 VBUS 取电，则外部需一个转换电路，如图 22-1 所示。

图 23-1 OTG B 类设备连接图



作为 USB 设备所必要的 USB 总线上拉电阻可以由软件随时设置是否启用，当 USB 控制寄存器 R8_USB_CTRL 中的 RB_UC_DEV_PU_EN 置 1 时，控制器根据 RB_UD_LOW_SPEED 的速度设置，在内部为 USB 总线的 DP/DM 引脚连接上拉电阻，并启用 USB 设备功能。

当检测到 USB 总线复位、USB 总线挂起或唤醒事件，或当 USB 成功处理完数据发送或数据接收后，USB 协议处理器都将设置相应的中断标志，如果中断使能打开，还会产生相应的中断请求。应用程序可以直接查询或在 USB 中断服务程序中查询并分析中断标志寄存器 R8_USB_INT_FG，根据 RB_UIF_BUS_RST 和 RB_UIF_SUSPEND 进行相应的处理；并且，如果 RB_UIF_TRANSFER 有效，那么还需要继续分析 USB 中

断状态寄存器R8_USB_INT_ST,根据当前端点号MASK_UIS_ENDP和当前事务令牌PID标识MASK_UIS_TOKEN进行相应的处理。如果事先设定了各个端点的 OUT 事务的同步触发位 RB_UEP_R_TOG，那么可以通过RB_U_TOG_OK 或 RB_UIS_TOG_OK 判断当前所接收到的数据包的同步触发位是否与该端点的同步触发位匹配，如果数据同步，则数据有效；如果数据不同步，则数据应该被丢弃。每次处理完 USB 发送或接收中断后，都应该正确修改相应端点的同步触发位，用于下次所发送的数据包或下次所接收的数据包是否同步检测；另外，设置 RB_UEP_AUTO_TOG 可以实现在发送成功或接收成功后自动翻转相应的同步触发位。

各个端点准备发送的数据在各自的缓冲区中,准备发送的数据长度是独立设定在R8_UEPn_T_LEN中；各个端点接收到的数据在各自的缓冲区中，但是接收到的数据长度都在 USB 接收长度寄存器R8_USB_RX_LEN 中，可以在 USB 接收中断时根据当前端点号区分。

表 23-3 设备相关寄存器列表

名称	访问地址	描述	复位值
R8_UDEV_CTRL	0x50000001	USB 设备物理端口控制寄存器	0xX0
R8_UEP4_1_MOD	0x5000000C	端点 1(9)/4(8/12)模式控制寄存器	0x00
R8_UEP2_3_MOD	0x5000000D	端点 2(10)/3(11)模式控制寄存器	0x00
R8_UEP5_6_MOD	0x5000000E	端点 5(13)/6(14)模式控制寄存器	0x00
R8_UEP7_MOD	0x5000000F	端点 7(15)模式控制寄存器	0x00
R16_UEP0_DMA	0x50000010	端点 0 缓冲区起始地址	0xXXXX
R16_UEP1_DMA	0x50000014	端点 1(9)缓冲区起始地址	0xXXXX
R16_UEP2_DMA	0x50000018	端点 2(10)缓冲区起始地址	0xXXXX
R16_UEP3_DMA	0x5000001C	端点 3(11)缓冲区起始地址	0xXXXX
R16_UEP4_DMA	0x50000020	端点 4(8/12)缓冲区起始地址	0xXXXX
R16_UEP5_DMA	0x50000024	端点 5(13)缓冲区起始地址	0xXXXX
R16_UEP6_DMA	0x50000028	端点 6(14)缓冲区起始地址	0xXXXX
R16_UEP7_DMA	0x5000002C	端点 7(15)缓冲区起始地址	0xXXXX
R8_UEP0_T_LEN	0x50000030	端点 0 发送长度寄存器	0xXX
R8_UEP0_TX_CTRL	0x50000032	端点 0 发送控制寄存器	0x00
R8_UEP0_RX_CTRL	0x50000033	端点 0 接收控制寄存器	0x00
R8_UEP1_T_LEN	0x50000034	端点 1(9)发送长度寄存器	0xXX
R8_UEP1_TX_CTRL	0x50000036	端点 1(9)发送控制寄存器	0x00
R8_UEP1_RX_CTRL	0x50000037	端点 1(9)接收控制寄存器	0x00
R8_UEP2_T_LEN	0x50000038	端点 2(10)发送长度寄存器	0xXX
R8_UEP2_TX_CTRL	0x5000003A	端点 2(10)发送控制寄存器	0x00
R8_UEP2_RX_CTRL	0x5000003B	端点 2(10)接收控制寄存器	0x00
R8_UEP3_T_LEN	0x5000003C	端点 3(11)发送长度寄存器	0xXX
R8_UEP3_TX_CTRL	0x500003E	端点 3(11)发送控制寄存器	0x00
R8_UEP3_RX_CTRL	0x500003F	端点 3(11)接收控制寄存器	0x00
R8_UEP4_T_LEN	0x50000040	端点 4(8/12)发送长度寄存器	0xXX
R8_UEP4_TX_CTRL	0x50000042	端点 4(8/12)发送控制寄存器	0x00
R8_UEP4_RX_CTRL	0x50000043	端点 4(8/12)接收控制寄存器	0x00
R8_UEP5_T_LEN	0x50000044	端点 5(13)发送长度寄存器	0xXX
R8_UEP5_TX_CTRL	0x50000046	端点 5(13)发送控制寄存器	0x00
R8_UEP5_RX_CTRL	0x50000047	端点 5(13)接收控制寄存器	0x00
R8_UEP6_T_LEN	0x50000048	端点 6(14)发送长度寄存器	0xXX
R8_UEP6_TX_CTRL	0x5000004A	端点 6(14)发送控制寄存器	0x00

R8_UEP6_RX_CTRL	0x5000004B	端点 6(14)接收控制寄存器	0x00
R8_UEP7_T_LEN	0x5000004C	端点 7(15)发送长度寄存器	0xXX
R8_UEP7_TX_CTRL	0x5000004E	端点 7(15)发送控制寄存器	0x00
R8_UEP7_RX_CTRL	0x5000004F	端点 7(15)接收控制寄存器	0x00

23.2.2.1 USB 设备物理端口控制寄存器(R8_UDEV_CTRL)

位	名称	访问	描述	复位值
7	RB_UD_PD_DIS	RW	USB 设备端口 UD+/UD-引脚内部下拉电阻控制位： 1：禁用内部下拉；0：使能内部下拉。 可用于 GPIO 模式提供下拉电阻。	1
6	Reserved	RO	保留。	0
5	RB_UD_DP_PIN	RO	当前 UD+引脚状态： 1：高电平；　　0：低电平。	X
4	RB_UD_DM_PIN	RO	当前 UD-引脚状态： 1：高电平；　　0：低电平。	X
3	Reserved	RO	保留。	0
2	RB_UD_LOW_SPEED	RW	USB 设备物理端口低速模式使能位： 1：选择 1.5Mbps 低速模式； 0：选择 12Mbps 全速模式。	0
1	RB_UD_GP_BIT	RW	USB 设备模式通用标志位，用户自定义。	0
0	RB_UD_PORT_EN	RW	USB 设备物理端口使能位： 1：使能物理端口；0：禁用物理端口。	0

23.2.2.2 端点 1(9)/4(8/12)模式控制寄存器(R8_UEP4_1_MOD)

位	名称	访问	描述	复位值
7	RB_UEP1_RX_EN	RW	1：使能端点 1(9)接收（OUT）； 0：禁止端点 1(9)接收。	0
6	RB_UEP1_TX_EN	RW	1：使能端点 1(9)发送（IN）； 0：禁止端点 1(9)发送。	0
5	Reserved	RO	保留。	0
4	RB_UEP1_BUF_MOD	RW	端点 1(9)数据缓冲区模式控制位。 *注：该位为 1 时，UEP1_RX_EN 和 UEP1_TX_EN 不能同时为 1。*	0
3	RB_UEP4_RX_EN	RW	1：使能端点 4(8/12)接收（OUT）； 0：禁止端点 4(8/12)接收。	0
2	RB_UEP4_TX_EN	RW	1：使能端点 4(8/12)发送（IN）； 0：禁止端点 4(8/12)发送。	0
1	Reserved	RO	保留。	0
0	RB_UEP4_BUF_MOD	RW	端点 4(8/12)数据缓冲区模式控制位。 *注：该位为 1 时，UEP4_RX_EN 和 UEP4_TX_EN 不能同时为 1。* *注：此位控制只支持 CH32V103x 系列。*	0

注：端点 1 配置选项映射端点 9，端点 4 配置选项映射端点 8 和 12。

23.2.2.3 端点 2(10)/3(11)模式控制寄存器(R8_UEP2_3_MOD)

位	名称	访问	描述	复位值
7	RB_UEP3_RX_EN	RW	1：使能端点 3(11)接收（OUT）； 0：禁止端点 3(11)接收。	0
6	RB_UEP3_TX_EN	RW	1：使能端点 3(11)发送（IN）； 0：禁止端点 3(11)发送。	0
5	Reserved	RO	保留。	0
4	RB_UEP3_BUF_MOD	RW	端点 3(11)数据缓冲区模式控制位。 *注：该位为 1 时，UEP3_RX_EN 和 UEP3_TX_EN 不能同时为 1。*	0
3	RB_UEP2_RX_EN	RW	1：使能端点 2(10)接收（OUT）； 0：禁止端点 2(10)接收。	0
2	RB_UEP2_TX_EN	RW	1：使能端点 2(10)发送（IN）； 0：禁止端点 2(10)发送。	0
1	Reserved	RO	保留。	0
0	RB_UEP2_BUF_MOD	RW	端点 2(10)数据缓冲区模式控制位。 *注：该位为 1 时，UEP2_RX_EN 和 UEP2_TX_EN 不能同时为 1。*	0

注：端点 2 配置选项映射端点 10，端点 3 配置选项映射端点 11。

23.2.2.4 端点 5(13)/6(14)模式控制寄存器(R8_UEP5_6_MOD)

位	名称	访问	描述	复位值
7	RB_UEP6_RX_EN	RW	1：使能端点 6(14)接收（OUT）； 0：禁止端点 6(14)接收。	0
6	RB_UEP6_TX_EN	RW	1：使能端点 6(14)发送（IN）； 0：禁止端点 6(14)发送。	0
5	Reserved	RO	保留。	0
4	RB_UEP6_BUF_MOD	RW	端点 6(14)数据缓冲区模式控制位。 *注：该位为 1 时，UEP6_RX_EN 和 UEP6_TX_EN 不能同时为 1。*	0
3	RB_UEP5_RX_EN	RW	1：使能端点 5(13)接收（OUT）； 0：禁止端点 5(13)接收。	0
2	RB_UEP5_TX_EN	RW	1：使能端点 5(13)发送（IN）； 0：禁止端点 5(13)发送。	0
1	Reserved	RO	保留。	0
0	RB_UEP5_BUF_MOD	RW	端点 5(13)数据缓冲区模式控制位。 *注：该位为 1 时，UEP5_RX_EN 和 UEP5_TX_EN 不能同时为 1*	0

注：端点 5 配置选项映射端点 13，端点 6 配置选项映射端点 14。

23.2.2.5 端点 7(15)模式控制寄存器(R8_UEP7_MOD)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0

3	RB_UEP7_RX_EN	RW	1：使能端点 7(15)接收（OUT）； 0：禁止端点 7(15)接收。	0
2	RB_UEP7_TX_EN	RW	1：使能端点 7(15)发送（IN）； 0：禁止端点 7(15)发送。	0
1	Reserved	RO	保留。	0
0	RB_UEP7_BUF_MOD	RW	端点 7(15)数据缓冲区模式控制位。	0

注：端点 7 配置选项映射端点 15。

由 RB_UEPn_RX_EN 和 RB_UEPn_TX_EN 以及 RB_UEPn_BUF_MOD 组合分别配置 USB 端点 1-15 的数据缓冲区模式，具体参考表 23-4。其中，在双 64 字节缓冲区模式下，USB 数据传输时将根据 RB_UEP_*_TOG=0 选择前 64 字节缓冲区，根据 RB_UEP_*_TOG=1 选择后 64 字节缓冲区，设置 RB_UEP_AUTO_TOG=1 可实现自动切换。

表 23-4 端点 n 缓冲区模式（n=1-7）

RB_UEPn_RX_EN	RB_UEPn_TX_EN	RB_UEPn_BUF_MOD	描述：以 R16_UEPn_DMA 为起始地址由低向高排列
0	0	X	端点被禁用，未用到 R16_UEPn_DMA 缓冲区。
1	0	0	单 64 字节接收缓冲区(OUT)。
1	0	1	双 64 字节接收缓冲区(OUT)，由 RB_UEP_R_TOG 选择。
0	1	0	单 64 字节发送缓冲区(IN)。
0	1	1	双 64 字节发送缓冲区(IN)，由 RB_UEP_T_TOG 选择。
1	1	0	单 64 字节接收缓冲区(OUT)，单 64 字节发送缓冲区(IN)。
1	1	1	双 64 字节接收缓冲区(OUT)，通过 RB_UEP_R_TOG 选择， 双 64 字节发送缓冲区(IN)，通过 RB_UEP_T_TOG 选择。 全部 256 字节排列如下： UEPn_DMA+0 地址：RB_UEP_R_TOG=0 时端点接收地址； UEPn_DMA+64 地址：RB_UEP_R_TOG=1 时端点接收地址； UEPn_DMA+128 地址：RB_UEP_T_TOG=0 时端点发送地址； UEPn_DMA+192 地址：RB_UEP_T_TOG=1 时端点发送地址。

注：表 21-4 的配置选择支持 n=1-7，端点 8-15 配置映射端点 1-7 配置。

端点 n 缓冲区起始地址(R8_UEPn_DMA) (n=0-7)

位	名称	访问	描述	复位值
[17:0]	R8_UEPn_DMA	RW	端点 n 缓冲区起始地址。 低 15 位有效，地址必须 4 字节对齐。	X

注：1. 接收数据的缓冲区的长度 >= min（可能收到的最大数据包长度 + 2 字节，64 字节）。
2. 端点 DMA 配置支持 0-7 端点，可映射配置端点 8-15 端点。

23.2.2.6 端点 n 发送长度寄存器(R16_UEPn_T_LEN) (n=0-7)

位	名称	访问	描述	复位值
[9:0]	R8_UEP3_T_LEN	RW	设置 USB 端点 3 准备发送的数据字节数。	X
[7:0]	R8_UEPn_T_LEN	RW	设置 USB 端点 n 准备发送的数据字节数 n=0, 1, 2, 4, 5, 6, 7。	

注 1. 端点发送长度配置支持 0-7 端点，可映射配置 8-15 端点的发送。
2. 主机发送支持最大 1023 字节（针对同步端点）

23.2.2.7 端点 n 控制寄存器(R8_UEPn_TX_CTRL) (n=0-7)

位	名称	访问	描述	复位值
[7:4]	保留	RO	保留。	0
3	RB_UEP_T_AUTO_TOG	RW	同步触发位自动翻转使能控制位： 1：数据发送成功后自动翻转相应的同步触发位； 0：不自动翻转，可以手动切换。 *注：端点 0 此位保留。*	0
2	RB_UEP_T_TOG	RW	USB 端点 n 的发送器（处理 IN 事务）准备的同步触发位： 1：发送 DATA1；　　0：发送 DATA0。	00b
[1:0]	MASK_UEP_T_RES	RW	端点 n 的发送器对 IN 事务的响应控制： 00：DATA0/DATA1 数据就绪并期望 ACK； 01：应答 DATA0/DATA1 并期望无响应，用于非端点 0 的实时/同步传输； 10：应答 NAK 或忙； 11：应答 STALL 或错误。	00b

注：端点配置支持 0-7 端点，可映射配置端点 8-15 端点。

23.2.2.8 端点 n 控制寄存器(R8_UEPn_ RX_CTRL) (n=0-7)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0
3	RB_UEP_R_AUTO_TOG	RW	同步触发位自动翻转使能控制位： 1：数据接收成功后自动翻转相应的同步触发位； 0：不自动翻转，可以手动切换。 *注：端点 0 此位保留。*	0
2	MASK_UEP_R_TOG	RW	USB 端点 n 的接收器（处理 OUT 事务）期望的同步触发位： 1：期望 DATA1；　　0：期望 DATA0。	00b
[1:0]	MASK_UEP_R_RES	RW	端点 n 的接收器对 OUT 事务的响应控制： 00：应答 ACK； 01：超时/无响应，用于非端点 0 的实时/同步传输； 10：应答 NAK 或忙； 11：应答 STALL 或错误。	00b

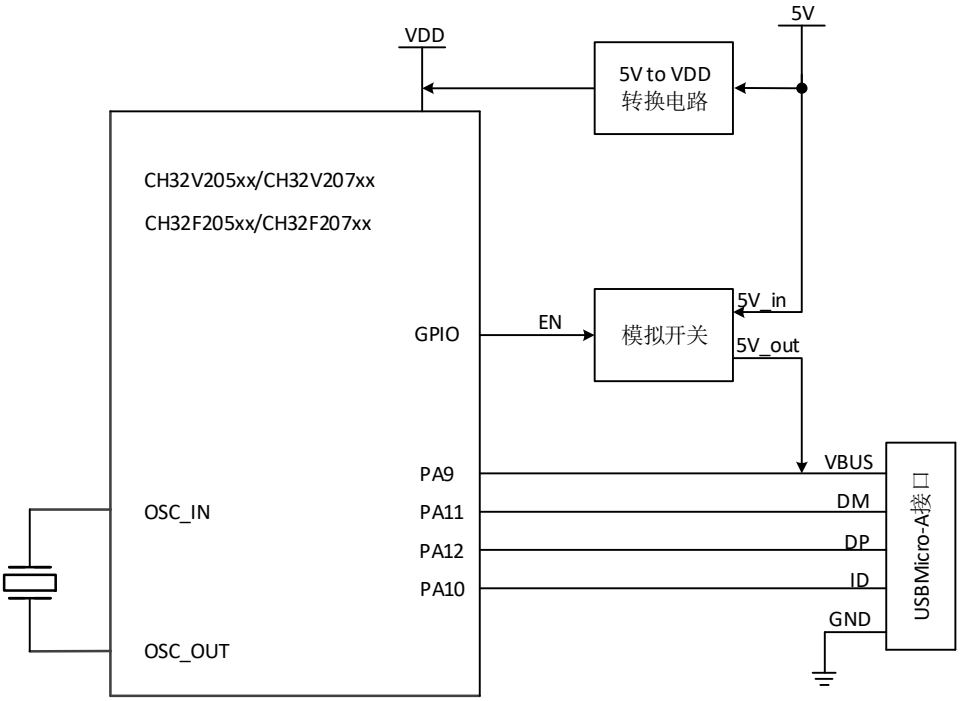
注：端点配置支持 0-7 端点，可映射配置端点 8-15 端点。

22.2.3 USB 主机寄存器

在 USB OTG 主机模式下，芯片提供了一组双向主机端点，包括一个发送端点 OUT 和一个接收端点 IN，一个数据包的最大长度是 1023 字节，支持控制传输、中断传输、批量传输和实时/同步传输。

在 USB OTG 主机模式下，若控制器不能为 VUBS 提供 5V 电源，则需外接电荷泵，若应用板能提供 5V 电源则可采用模拟开关控制 VBUS 的打开和关断，如图 23-2 所示。

图 23-2 OTG 的 A 类设备连接



主机端点发起的每一个 USB 事务，在处理结束后总是自动设置 RB_UIF_TRANSFER 中断标志。应用程序可以直接查询或在 USB 中断服务程序中查询并分析中断标志寄存器 R8_USB_INT_FG，根据各中断标志分别进行相应的处理；并且，如果 RB_UIF_TRANSFER 有效，那么还需要继续分析 USB 中断状态寄存器 R8_USB_INT_ST，根据当前 USB 传输事务的应答 PID 标识 MASK_UIS_H_RES 进行相应的处理。

如果事先设定了主机接收端点的 IN 事务的同步触发位（RB_UH_R_TOG），那么可以通过 RB_U_TOG_OK 或 RB_UIS_TOG_OK 判断当前所接收到的数据包的同步触发位是否与主机接收端点的同步触发位匹配，如果数据同步，则数据有效；如果数据不同步，则数据应该被丢弃。每次处理完 USB 发送或接收中断后，都应该正确修改相应主机端点的同步触发位，用于同步下次所发送的数据包和检测下次所接收的数据包是否同步；另外，通过设置 RB_UH_T_AUTO_TOG 和 RB_UH_R_AUTO_TOG 可以实现在发送成功或接收成功后自动翻转相应的同步触发位。

USB 主机令牌设置寄存器 R8_UH_EP_PID 用于设置被操作的目标设备的端点号和本次 USB 传输事务的令牌 PID 包标识。SETUP 令牌和 OUT 令牌所对应的数据由主机发送端点提供，准备发送的数据在 R16_UH_TX_DMA 缓冲区中，准备发送的数据长度设置在 R16_UH_TX_LEN 中；IN 令牌所对应数据由目标设备返回给主机接收端点，接收到数据存放 R16_UH_RX_DMA 缓冲区中，接收到的数据长度存放在 R16_USB_RX_LEN 中，主机端点可接收的最大包长度需要提前写入到 R16_UH_RX_MAX_LEN 寄存器中。

表 23-5 主机相关寄存器列表

名称	访问地址	描述	复位值
R8_UHOST_CTRL	0x50000001	USB 主机物理端口控制寄存器	0xX0
R32_UH_EP_MOD	0x5000000D	USB 主机端点模式控制寄存器	0x00
R16_UH_RX_DMA	0x50000018	USB 主机接收缓冲区起始地址	X
R16_UH_TX_DMA	0x5000001C	USB 主机发送缓冲区起始地址	X
R16_UH_SETUP	0x50000036	USB 主机辅助设置寄存器	0x00
R8_UH_EP_PID	0x50000038	USB 主机令牌设置寄存器	0x00
R8_UH_RX_CTRL	0x5000003A	USB 主机接收端点控制寄存器	0x00
R16_UH_TX_LEN	0x5000003C	USB 主机发送长度寄存器	X

R8_UH_TX_CTRL	0x5000003E	USB 主机发送端点控制寄存器	0x00

22.2.3.1 USB 主机物理端口控制寄存器(R8_UHOST_CTRL)

位	名称	访问	描述	复位值
7	RB_UH_PD_DIS	RW	USB 主机端口 UD+/UD-引脚内部下拉电阻控制位： 1：禁用内部下拉； 0：使能内部下拉。 可用于 GPIO 模式提供下拉电阻。	1
6	Reserved	RO	保留。	0
5	RB_UH_DP_PIN	RO	当前 UD+引脚状态： 1：高电平；　　0：低电平。	X
4	RB_UH_DM_PIN	RO	当前 UD-引脚状态： 1：高电平；　　0：低电平。	X
3	Reserved	RO	保留。	0
2	RB_UH_LOW_SPEED	RW	USB 主机端口低速模式使能位： 1：选择 1.5Mbps 低速模式； 0：选择 12Mbps 全速模式。	0
1	RB_UH_BUS_RESET	RW	USB 主机模式总线复位控制位： 1：强制输出 USB 总线复位； 0：结束输出。	0
0	RB_UH_PORT_EN	RW	USB 主机端口使能位： 1：使能主机端口； 0：禁用主机端口。 当 USB 设备断开连接时，该位自动清 0。	0

22.2.3.2 USB 主机端点模式控制寄存器(R32_UH_EP_MOD)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
6	RB_UH_EP_TX_EN	RW	主机发送端点发送(SETUP/OUT)使能位： 1：使能端点发送； 0：禁止端点发送。	0
5	Reserved	RO	保留。	0
4	RB_UH_EP_TBUF_MOD	RW	主机发送端点发送数据缓冲区模式控制位。	0
3	RB_UH_EP_RX_EN	RW	主机接收端点接收（IN）使能位： 1：使能端点接收； 0：禁止端点接收。	0
[2:1]	Reserved	RO	保留。	0
0	RB_UH_EP_RBUF_MOD	RW	USB 主机接收端点接收数据缓冲区模式控制位。	0

由 RB_UH_EP_TX_EN 和 RB_UH_EP_TBUF_MOD 组合控制主机发送端点数据缓冲区模式，参考下表。

表 23-6 主机发送缓冲区模式

RB_UH_EP_TX_EN	RB_UH_EP_TBUF_MOD	描述：以 R16_UH_TX_DMA 为起始地址
0	X	端点被禁用，未用到 R16_UH_TX_DMA 缓冲区。
1	0	单 64 字节发送缓冲区（SETUP/OUT）。
1	1	双 64 字节发送缓冲区，通过 RB_UH_T_TOG 选择： 当 RB_UH_T_TOG=0 时选择前 64 字节缓冲区； 当 RB_UH_T_TOG=1 时选择后 64 字节缓冲区。

由 RB_UH_EP_RX_EN 和 RB_UH_EP_RBUF_MOD 组合控制主机接收端点数据缓冲区模式，参考下表。

表 23-7 主机接收缓冲区模式

RB_UH_EP_RX_EN	RB_UH_EP_RBUF_MOD	结构描述：以 R16_UH_TX_DMA 为起始地址
0	X	端点被禁用，未用到 R16_UH_RX_DMA 缓冲区。
1	0	单 64 字节接收缓冲区（IN）。
1	1	双 64 字节接收缓冲区，通过 RB_UH_R_TOG 选择： 当 RB_UH_R_TOG=0 时选择前 64 字节缓冲区； 当 RB_UH_R_TOG=1 时选择后 64 字节缓冲区。

USB 主机接收缓冲区起始地址(R16_UH_RX_DMA)

位	名称	访问	描述	复位值
[15:0]	R16_UH_RX_DMA	RW	主机端点数据接收缓冲区起始地址。 低 15 位有效，地址必须 4 字节对齐。	X

USB 主机发送缓冲区起始地址(R16_UH_TX_DMA)

位	名称	访问	描述	复位值
[15:0]	R16_UH_TX_DMA	RW	主机端点数据发送缓冲区起始地址。 低 15 位有效，地址必须 4 字节对齐。	X

22.2.3.3 USB 主机辅助设置寄存器(R16_UH_SETUP)

位	名称	访问	描述	复位值
[15:11]	保留	RO	保留。	0
10	RB_UH_PRE_PID_EN	RW	低速前导包 PRE PID 使能位： 1：使能，用于通过外部 HUB 与低速 USB 设备通讯。 0：禁用低速前导包。	0
[9:3]	保留	RO	保留。	0
2	RB_UH_SOF_EN	RW	自动产生 SOF 包使能位： 1：主机自动产生 SOF 包； 0：关闭自动 SOF 功能。	0
[1:0]	保留	RO	保留。	0

22.2.3.4 USB 主机令牌设置寄存器(R8_UH_EP_PID)

位	名称	访问	描述	复位值
[7:4]	MASK_UH_TOKEN	RW	设置本次 USB 传输事务的令牌 PID 标识。	0
[3:0]	MASK_UH_ENDP	RW	设置本次被操作的目标设备的端点号。	0

22.2.3.5 USB 主机接收端点控制寄存器(R8_UH_RX_CTRL)

位	名称	访问	描述	复位值
[7:4]	保留	RO	保留。	0
3	RB_UH_R_AUTO_TOG	RW	同步触发位自动翻转使能控制位： 1：数据接收成功后自动翻转相应的期待同步触发位(RB_UH_R_TOG)； 0：手动控制同步触发位(RB_UH_R_TOG)。	0
2	RB_UH_R_TOG	RW	主机接收器（处理 IN 事务）准备的同步触发位： 1：无响应，用于非 0 端点的实时/同步传输； 0：应答 ACK。	0
1	保留	RO	保留。	0
0	RB_UH_R_RES	RO	主机接收器对 IN 事务的响应控制位： 1：无响应，用于非 0 端点的实时/同步传输； 0：应答 ACK。	0

22.2.3.6 USB 主机发送长度寄存器(R16_UH_TX_LEN)

位	名称	访问	描述	复位值
[16:0]	R16_UH_TX_LEN	RW	设置 USB 主机发送端点准备发送的数据字节数。	X

22.2.3.7 USB 主机发送端点控制寄存器(R8_UH_TX_CTRL)

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0
3	RB_UH_T_AUTO_TOG	RO	同步触发位自动翻转使能控制位： 1：数据发送成功后自动翻转相应的同步触发位(RB_UH_T_TOG)； 0：手动控制同步触发位(RB_UH_T_TOG)。	0
2	RB_UH_T_TOG	RW	USB 主机发送器（处理 SETUP/OUT 事务）准备的同步触发位： 1：表示发送 DATA1； 0：表示发送 DATA0。	0
1	Reserved	RO	保留。	0
0	RB_UH_T_RES	RW	USB 主机发送器对 SETUP/OUT 事务的响应控制位： 1：期望无响应，用于非 0 端点的实时/同步传输；	0

			0：期望应答 ACK。	

第 24 章 控制器局域网（CAN）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器系列部分产品。

控制器局域网是一种用于串行数据通信的高性能通信协议。CAN 控制器提供了一个完整的 CAN 协议实现方案，支持 CAN 协议 2.0A 和 2.0B。CAN 控制器可以用来构建强大的局域网来实现安全的分布式实时控制，以较小的 CPU 负荷来处理大量的数据报文，在工业和汽车领域有着广泛的应用。

24.1 主要特性

- 兼容 CAN 规范 2.0A 和 2.0B
- 可编程的传输速率，最高可达 1Mbit/s
- 支持时间触发通信功能，避免低优先级消息阻塞
- 支持三个发送邮箱，发送报文优先级可由报文标识符或发送请求的次序决定，并可记录发送报文 SOF 时刻的时间戳
- 支持三级邮箱深度的 2 个接收 FIFO，28 个报文过滤器组可供配置，大容量产品 CAN1 和 CAN2 共用 28 个过滤器，每个过滤器组可配置成 32 或 16 位模式，屏蔽位或标识符列表模式，能够尽量减少软件对报文筛选的干预，FIFO 溢出处理方式灵活，并可记录接收报文 SOF 时刻的时间戳
- 占用 4 个中断向量，每个中断源可以独立配置

24.2 CAN 控制器工作模式

CAN 控制器可以对寄存器 CAN_CTLR 中的 SLEEP 或 INRQ 位进行操作，实现在初始化模式、睡眠模式和正常模式 3 个工作模式下切换。

24.2.1 初始化模式

在复位后，CAN 默认工作在睡眠模式以减低功耗，此时禁止报文收发，TX 引脚的内部上拉电阻使能，TX 引脚输出隐性位。对寄存器 CAN_CTLR 中的 INRQ 位置 1，请求 CAN 控制器进入初始化模式，当寄存器 CAN_STATR 的 INAK 位自动置 1 则成功进入初始化状态。同样对寄存器 CAN_CTLR 中的 INRQ 位清零，请求 CAN 控制器退出初始化模式，当寄存器 CAN_STATR 的 INAK 位自动清 0 则成功退出初始化状态。

对过滤器组进行初始化，可以在非初始化模式下进行，不过必须对寄存器 CAN_FCTLR 的 FINIT 位进行置 1，此时禁止接收报文。

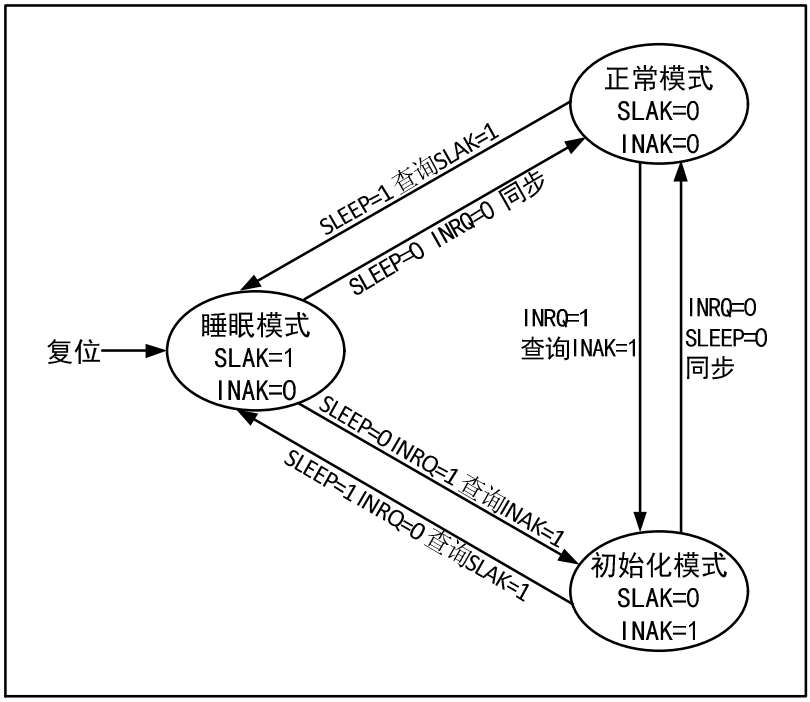
24.2.2 睡眠模式

对寄存器 CAN_CTLR 中的 SLEEP 位置 1，请求 CAN 控制器进入睡眠模式，当寄存器 CAN_STATR 的 SNAK 位自动置 1 则 CAN 成功进入睡眠模式，此时 CAN 控制器的时钟停止，但邮箱寄存器仍可访问。

由睡眠模式进入初始化模式，必须对 CAN_CTLR 的 SLEEP 位清 0，INRQ 位置 1，当寄存器 CAN_STATR 的 INAK 位自动置 1 则切换为初始化状态完成。

由睡眠模式进入正常模式，必须对 CAN_CTLR 的 SLEEP 位清 0，当寄存器 CAN_STATR 的 SNAK 位自动清 0 则进入正常模式。

图 24-1 CAN 工作模式切换



24.3 CAN 控制器测试模式

在初始化模式下，对寄存器 CAN_BTIMR 的 SILM 和 LBKM 位进行操作，可以选择一种测试模式，然后通过对寄存器 CAN_CTLR 的 INRQ 位清零，退出初始化模式，进入测试模式。测试模式分为静默模式、环回模式和静默环回模式三种。

24.3.1 静默模式

对寄存器 CAN_BTIMR 的 SILM 位置 1，可选择进入静默模式。该模式下，CAN 控制器可以接收，不能对外发送报文，对外总是处于隐性位，可以避免对总线产生影响，但是报文能够被所在节点的控制器所接收。通常静默模式被用于 CAN 总线的状态分析。

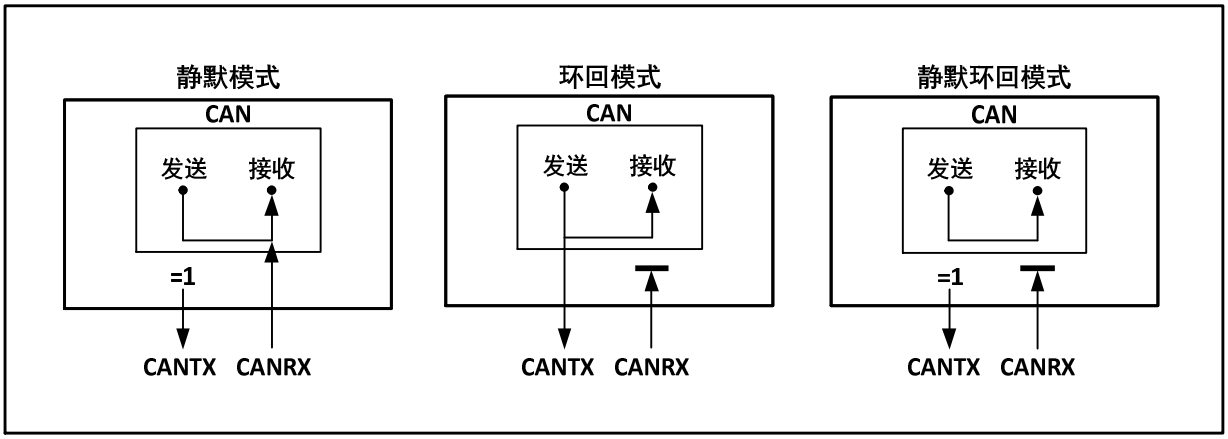
24.3.2 环回模式

对寄存器 CAN_BTIMR 的 LBKM 位置 1，可选择进入环回模式。该模式下，CAN 控制器可以对外发送报文，不能接收外部报文，但是发送报文能够被所在节点的控制器所接收，接收过滤机制有效。通常环回模式被用于 CAN 控制器的收发测试。

24.3.3 静默环回模式

对寄存器 CAN_BTIMR 的 SILM 和 LBKM 位置 1，可选择进入静默环回模式。该模式通常用于 CAN 控制器封闭自测试，在该模式下，对 CAN 总线无影响，RX 引脚与总线断开，TX 引脚置隐性位。

图 24-2 CAN 总线的三种测试模式



24.4 MCU 处于调试模式下 CAN 控制器的工作状态

当 MCU 进入调试模式后，内核处于暂停状态，但可以通过调试模块中配置位来决定 CAN 控制器是处于正常运行或停止状态。

24.5 CAN 控制器功能描述

24.5.1 发送处理流程

发送处理流程如下：如果三个发送邮箱中有空置的邮箱，应用层软件仅对空置邮箱的寄存器具有写入权限，对寄存器 CAN_TXMIRx、CAN_TXMDTRx、CAN_TXMDLRx 和 CAN_TXMDHRx 进行操作，可以设置报文标识符、报文长度、时间戳和报文数据等。在数据准备好之后，对寄存器 CAN_TXMIRx 的 TXRQ 位置 1 请求发送，邮箱进入挂号状态，并进行优先级排队；一旦成为最高优先级邮箱，则变为预定发送状态，等待 CAN 总线空闲；当 CAN 总线空闲时，预定发送邮箱的报文立刻进入发送状态；报文发送完毕后，邮箱重新成为空置邮箱，并且寄存器 CAN_TSTATR 的 RQCP 和 TXOK 位置 1，来指示发送成功；若发送时仲裁失败，寄存器 CAN_TSTATR 的 ALST 位置 1，若发送错误，则 TERR 位置 1。

24.5.2 发送优先级

发送优先级可以由标识符或发送请求先后次序决定，寄存器 CAN_CTLR 的 TXFP 位置 1 按发送请求先后次序发送，按发送请求先后次序主要应用于分段发送；TXFP 位清 0 按标识符优先级决定发送次序，标识符越小则优先级越高，同标识符的情况下，则低编号的邮箱有更高优先级。

24.5.3 发送中止处理

若对寄存器 CAN_TSTATR 的 ABRQ 位置 1，则可以中止发送请求。当邮箱状态为挂号或预定发送状态时，发送请求直接中止；当邮箱处于发送状态时，中止请求可能会成功（停止发送），也有可能会失败（发送完成），结果可由寄存器 CAN_TSTATR 的 TXOK 位来查询。

24.5.4 基于时间触发模式

传统的 CAN 通信总线繁忙时，容易造成低优先级的消息长时间阻塞，甚至无法满足其时限的要求。为了解决该瓶颈，推出了基于时间触发模式的相关协议，此类协议在工业上有一定规模的应用，基于时间触发模式的功能即为配合此类协议的应用。

在时间触发模式下有两种模式可供选择，使用该模式需关闭自动重传功能，通过配置 CAN_TTCTLR 寄存器的 MODE 位来选择默认模式和增强模式。对寄存器 CAN_CTLR 的 TTCM 和 NART 位置 1，使能时间触

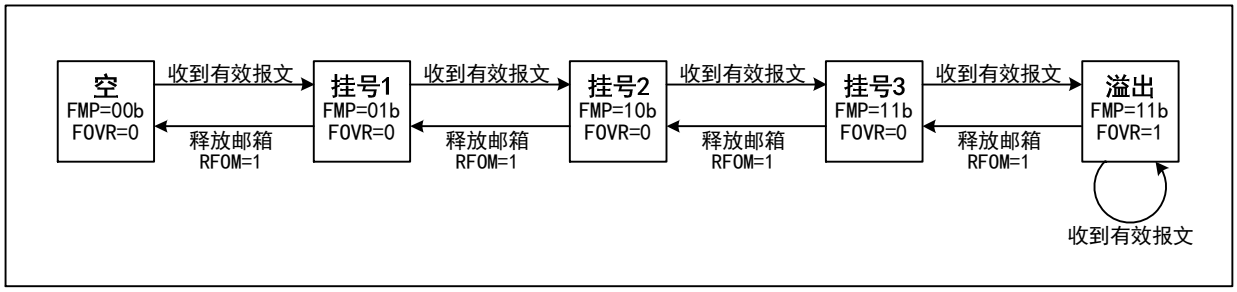
发模式并禁止自动重传，CAN_TTCTLR 寄存器的 MODE 位默认为 0，此时工作在默认模式，内部定时器被激活用来产生发送和接收邮箱的时间戳，定时器在 CAN 位时间累加，内部定时器在接收和发送的帧起始位的采样点位置被采样并产生时间戳。若使用增强模式，需要将配置 CAN_TTCTLR 寄存器的 MODE 位为 1，来开启增强模式，使用该模式，在整个 CAN 网络中，必须存在三个或三个以上的节点，其中一个节点发送时间基准，其他节点收到该基准节点的时间戳后，通过向 CAN_TTCTLR 寄存器的 TIMRST 位写 1 复位内部计数器，将内部计数器进行同步，这样除了发送时间基准的节点之外，其余 CAN 节点实现了时间同步，之后将待发送的数据写入发送邮箱，依次配置各节点的时间触发计数值（CAN_TTCNT 寄存器的 TIMCNT）和内部计数器计数终值（CAN_TTCTLR 寄存器 TIMCMV），时间触发计数值和内部计数器计数终值由 CAN 节点的个数、CAN 通信速率和一帧数据位数决定，配置完成后，各节点等待内部计数器计数到时间触发计数值后，触发发送动作。

24.5.5 接收处理流程

CAN 总线报文的接收，由控制器硬件来完成，无需 MCU 的干涉，减轻了 MCU 的处理负荷。所接收到的报文，根据寄存器 CAN_FAFIFOR 的设置，分别被存储到两个具有 3 级邮箱深度的 FIFO 中，应用层如需获取报文，只能通过接收 FIFO 邮箱来读取有效接收报文。

初始时，接收 FIFO 为空，接收 FIFO 寄存器 CAN_RFIFOx 的 FMR[1:0]值为二进制 00b，接到一个有效接收报文后，变为挂号 1 状态，控制器自动把接收 FIFO 寄存器 CAN_RFIFOx 的 FMR[1:0]设置二进制 01b；若此时读取邮箱数据寄存器 CAN_RXMDLRx 和 CAN_RXMDHRx，通过对接收 FIFO 寄存器 CAN_RFIFOx 的 RFOM 位置 1 来释放邮箱，接收 FIFO 状态又变为空；如果在挂号 1 状态时不释放邮箱，下一个有效接收报文被接到后，接收 FIFO 状态切换为挂号 2 状态，此时接收 FIFO 寄存器 CAN_RFIFOx 的 FMR[1:0]自动置二进制 10b；若读取邮箱数据寄存器并释放邮箱，则状态回到挂号 1；如果在挂号 2 状态不释放邮箱，则接收 FIFO 进入挂号 3 状态；同样在挂号 3 状态下读取报文并释放邮箱，则返回挂号 2 状态；若在挂号 3 状态不释放邮箱，则在接收到下一个有效报文时，必然导致报文丢失情况出现。

图 24-3 接收 FIFO 状态切换图



上文中的报文丢失情况，即接收 FIFO 为满，报文溢出导致报文丢失，接收 FIFO 寄存器 CAN_RFIFOx 的 FOVR 位会硬件自动置 1，以供溢出查询。寄存器 CAN_CTLR 的 RFLM 位置 1，则接收 FIFO 锁定功能启用，丢弃的报文为新接收报文；寄存器 CAN_CTLR 的 RFLM 位清 0，则接收 FIFO 锁定功能停用，接收 FIFO 的三个原报文中，最后接收的报文会被新报文覆盖。

当寄存器 CAN_INTENR 相关位置位，可以使接收 FIFO 状态切换时产生中断，以便更高效的处理接收报文，详见 24.6 节 CAN 中断。

24.5.6 接收报文标识符过滤

模块中有着多达 28 个过滤器组，通过设置过滤器组，每个 CAN 节点都可以接收到符合过滤规则的报文，不符合过滤规则的报文被硬件丢弃，无需软件干涉。

每个过滤器组由 2 个 32 位寄存器 CAN_FxR0 和 CAN_FxR1 组成。过滤器组的位宽都可以通过设置寄存器 CAN_FSCFGR 的各个位独立配置成 1 个 32 位过滤器或两个 16 位过滤器。每个过滤器组可通过设置

寄存器 CAN_FMCFGR 的各个位配置为屏蔽位或标识符列表模式，各个过滤器组可以通过设置寄存器 CAN_FWR 的各个位选择启用或禁用。设置寄存器 CAN_FAFIFOR 的各个位可以把选择通过过滤器的报文存放到哪个接收 FIFO。

如下表 24-1 所示，屏蔽位模式下，两个寄存器分别为标识符寄存器和屏蔽寄存器，两者需要配合使用，标识符寄存器每一位指示相应的位期望值为显性或隐性，屏蔽寄存器每一位指示相应位是否需要对应标识符寄存器位期望值一致。

表 24-1 32 位屏蔽位模式

标识符寄存器	CAN_FxR1[31:24]	CAN_FxR1[23:16]		CAN_FxR1[15:8]	CAN_FxR1[7:0]			
屏蔽位寄存器	CAN_FxR2[31:24]	CAN_FxR2[23:16]		CAN_FxR2[15:8]	CAN_FxR2[7:0]			
映射	STID[10:3]	STID[2:0]	EXID[17:13]	EXID[12:5]	EXID[4:0]	IDE	RTR	0

标识符列表模式下，两个寄存器都被用作标识符寄存器，接收报文标识符的每一位都必须与其中一个寄存器保持一致才能通过筛选。

表 24-2 32 位标识符列表模式

标识符寄存器	CAN_FxR1[31:24]	CAN_FxR1[23:16]		CAN_FxR1[15:8]	CAN_FxR1[7:0]			
屏蔽位寄存器	CAN_FxR2[31:24]	CAN_FxR2[23:16]		CAN_FxR2[15:8]	CAN_FxR2[7:0]			
映射	STID[10:3]	STID[2:0]	EXID[17:13]	EXID[12:5]	EXID[4:0]	IDE	RTR	0

在 16 位模式下，寄存器组被拆分成四个寄存器，屏蔽位模式每组过滤器的屏蔽位模式可以有 2 个过滤器，每个过滤器里各包含一个 16 位标识符寄存器和 16 位屏蔽寄存器；标识符列表模式下四个寄存器都用作标识符寄存器。

表 24-3 16 位屏蔽位模式

标识符寄存器 n	CAN_FxR1[15:8]	CAN_FxR1[7:0]			
屏蔽位寄存器 n	CAN_FxR1[31:24]	CAN_FxR1[23:16]			
标识符寄存器 n+1	CAN_FxR2[15:8]	CAN_FxR2[7:0]			
屏蔽位寄存器 n+1	CAN_FxR2[31:24]	CAN_FxR2[23:16]			
映射	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]

表 24-4 16 位标识符列表模式

标识符寄存器 n	CAN_FxR1[15:8]	CAN_FxR1[7:0]			
屏蔽位寄存器 n	CAN_FxR1[31:24]	CAN_FxR1[23:16]			
标识符寄存器 n+1	CAN_FxR2[15:8]	CAN_FxR2[7:0]			
屏蔽位寄存器 n+1	CAN_FxR2[31:24]	CAN_FxR2[23:16]			
映射	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]

报文进入 FIFO 邮箱中，会被应用程序读取并存放，通常应用程序根据报文标识符来区分报文数据。CAN 控制器对接收 FIFO 中通过不同过滤器筛选的报文，提供了过滤器编号，编号被存放在寄存器 CAN_RXMDTRx 的 FMI[7:0]中，编号时不考虑过滤器组是否启用。编号规则详见图 22-4 的示例。

当出现某个报文能通过多个过滤器的过滤，则接收邮箱中存放的过滤器编号根据过滤器优先级规则来决定存放哪个过滤器的编号，过滤器优先级规则如下：

- 所有 32 位的过滤器优先级均高于 16 位的过滤器
- 对于同样宽度的过滤器，标识符列表的过滤器优先级高于屏蔽位模式的过滤器
- 宽度和模式都一致的过滤器，编号小的过滤器优先级更高

如图 22-5 所示：在接收报文时，先把标识符与 32 位标识符列表模式过滤器进行匹配筛选，没有匹

配再与 32 位屏蔽位模式过滤器进行匹配筛选，没有匹配则继续与 16 位标识符列表模式过滤器进行匹配筛选，没有匹配最后与 16 位屏蔽位模式过滤器进行匹配筛选，最后如果都没有匹配则丢弃报文，出现匹配则报文存入接收 FIFO 的邮箱，标识符编号存入寄存器 CAN_RXMDTRx 的 FMI 中。

图 24-4 过滤器编号的示例

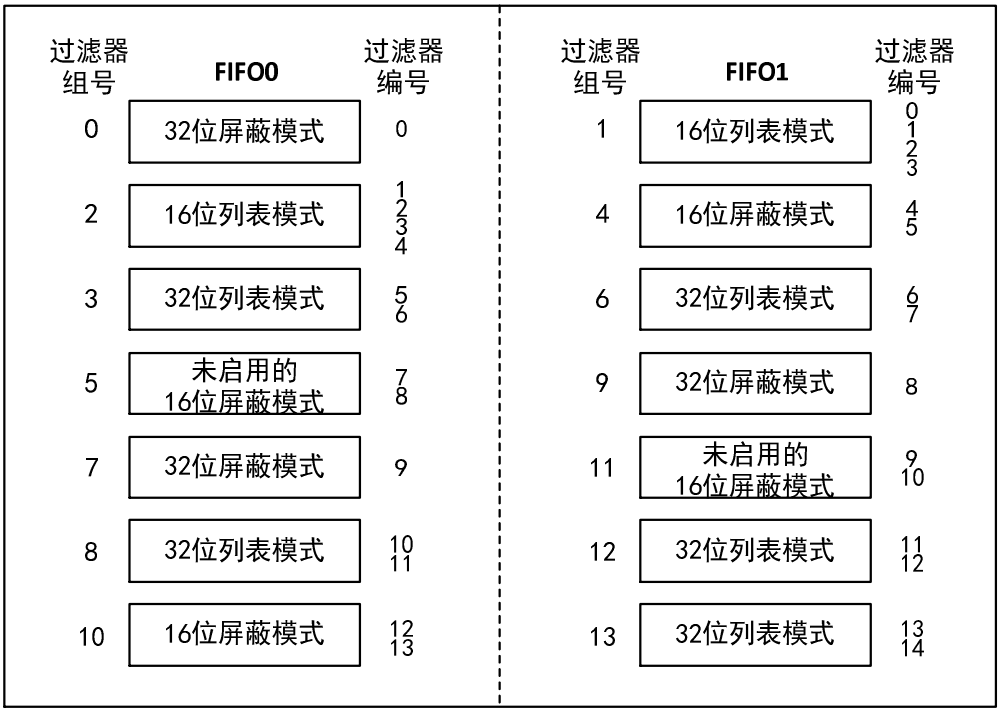
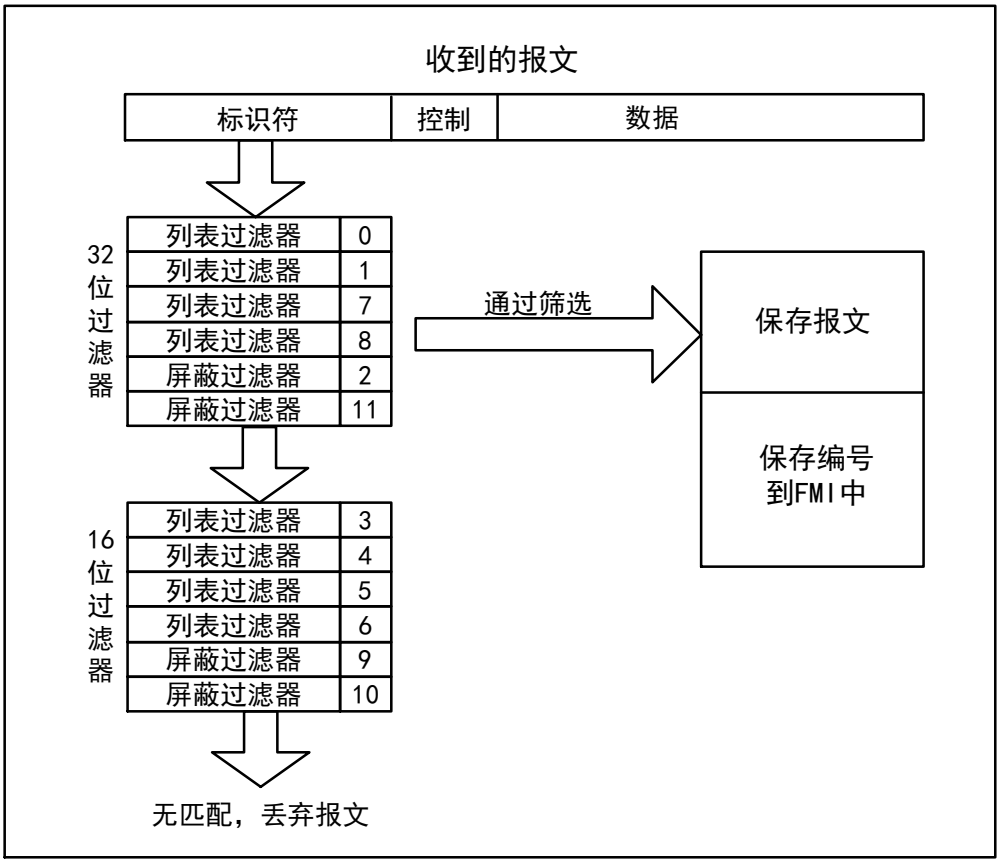


图 24-5 过滤器过滤示例



24.5.7 出错处理

CAN 控制器依靠状态错误寄存器 CAN_ERRSR，对于总线上的出错管理。状态错误寄存器 CAN_ERRSR 里的 TEC 和 REC，分别代表发送和接收错误计数值，根据随着收发错误的增加而增加，收发成功而减小，可以根据它们的值来判断 CAN 总线的稳定性。

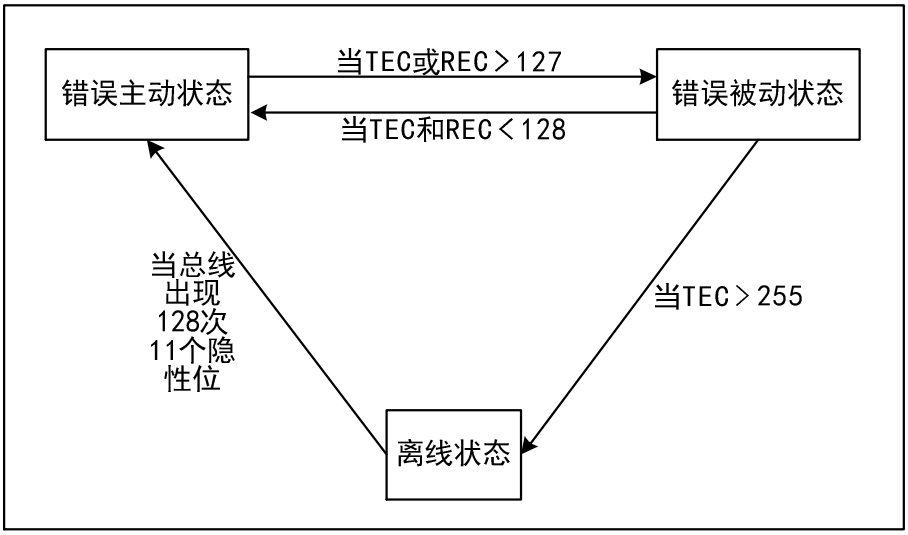
当状态错误寄存器 CAN_ERRSR 里的 TEC 和 REC 小于 128 时，当前 CAN 节点处于错误主动状态，可以正常参与总线通信，并且在侦测到错误的时候发出主动错误标志。

当状态错误寄存器 CAN_ERRSR 里的 TEC 和 REC 大于 127 时，当前 CAN 节点处于错误被动状态，并且在侦测到错误的时候不允许发出主动错误标志，只能发出被动错误标志。

当状态错误寄存器 CAN_ERRSR 里的 TEC 大于 255 时，当前 CAN 节点进入离线状态。

当总线监视到 128 次出现 11 个连续的隐性位时，恢复到错误主动状态，该恢复方式受主控制寄存器 CAN_CTLR 里的 ABOM 位影响。若 ABOM 置 1，则硬件自动退出离线状态。若 ABOM 为 0，则需要软件操作 INRQ 位进入初识化模式，随后退出初始化，才能退出离线状态。

图 24-6 CAN 错误状态切换图



24.5.8 位时序

按照 CAN 总线的标准，将每一位时间分为四段：分别为同步段、传播时间段、相位缓冲段 1 和相位缓冲段 2。这些段由最小时间单元 Tq 组成。CAN 控制器通过采样来监测 CAN 总线变化，通过帧起始位的边沿进行同步

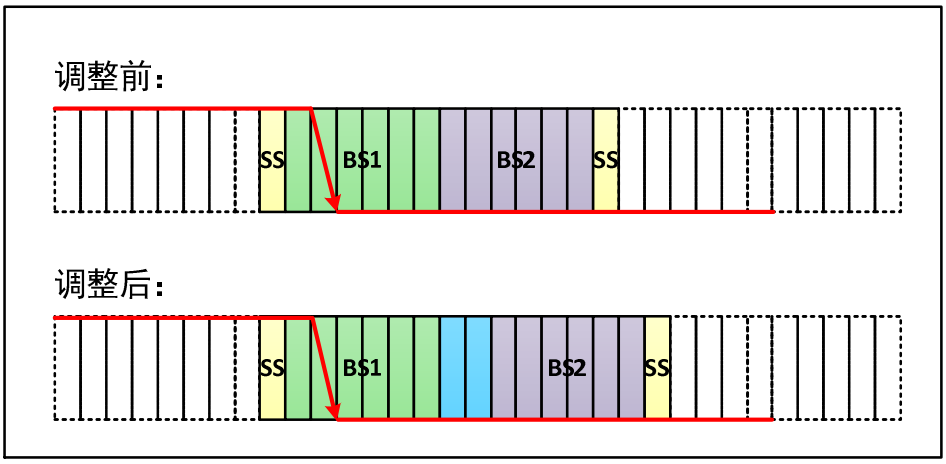
CAN 控制器把上述四段重新划分为三段，分别为：

- 同步段(SS)：也就是 CAN 标准里的同步段，固定为 1 个最小时间单元，正常情况下所期望的位跳变发生在本时间段内。
- 时间段 1(BS1)：包含 CAN 标准里的传播时间段和相位缓冲段 1，可以被设置为包含 1 到 16 最小时间单元，可以被自动延长，用于补偿 CAN 总线上不同节点频率精度误差带来的相位正向漂移。该时间段结束为采样点位置。
- 时间段 2(BS2)：也就是 CAN 标准里的相位缓冲段 2，可以被设置为 1 到 8 个最小时间单元，可以被自动缩短，以补偿 CAN 总线上不同节点频率精度误差带来的相位负向漂移。

重新同步跳转宽度(SJW)，是每位中可以延长和缩小的最小时间单元数量上限，范围可设置为 1 到 4 个最小时间单元。

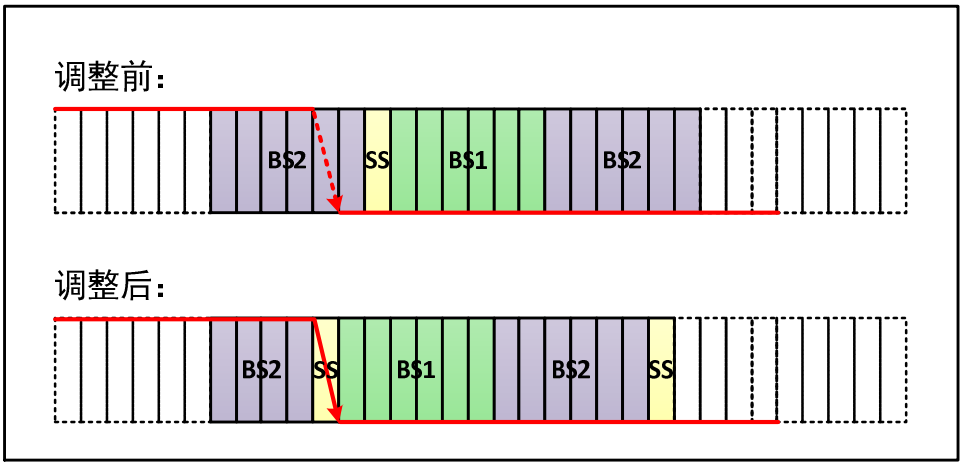
上述参数都可以在 CAN 总线时序寄存器 CAN_BTIMR 里配置。

图 24-7 跳变出现在 BS1 中



如图 24-7，SJW 为 2，总线电平跳变在时间段 1 被检测到，则需要延长时间段 1 的长度，最大延长 SJW，从而延迟采样点的位置。

图 24-8 跳变出现在 BS2 中



如图 24-8，SJW 为 2，总线电平跳变在时间段 2 被检测到，则需要缩小时间段 2 的长度，最大缩小 SJW，从而提前采样点的位置。

24.6 CAN 中断

CAN 控制器有四个中断向量，分别为发送中断、FIFO_0 中断、FIFO_1 中断、错误及状态变化中断。

设置 CAN 中断允许寄存器 CAN_INTENR，可以允许或禁用各个中断源。

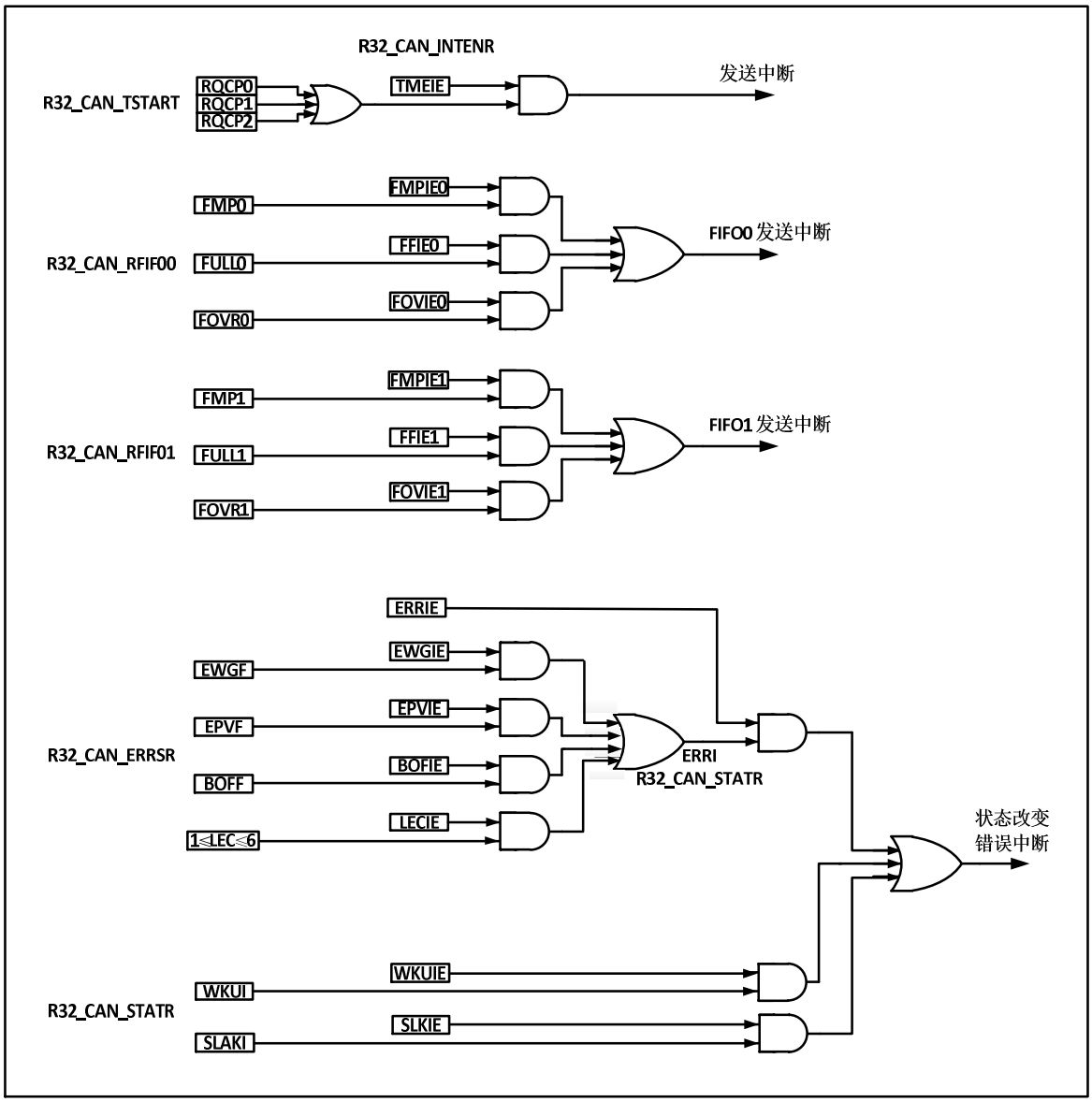
发送中断由发送邮箱变空事件产生，中断产生后，查询寄存器 CAN_TSTATR 的 RQCP0、RQCP1 和 RQCP2 位来判断是哪个邮箱变空事件产生。

FIFO0 中断由接收新报文、接收邮箱变满和溢出事件产生，中断产生后，查询寄存器 CAN_RFIFO0 的 FMP0、FULL0 和 FOVER0 位来判断是哪个邮箱变空事件产生。

FIFO1 中断由接收新报文、接收邮箱变满和溢出事件产生，中断产生后，查询寄存器 CAN_RFIFO1 的 FMP1、FULL1 和 FOVER1 位来判断是哪个邮箱变空事件产生。

错误及状态变化中断由出错、唤醒和睡眠事件产生。

图 24-9 CAN 中断逻辑图



24.7 寄存器描述

CAN 控制器相关的寄存器必须用 32 位字的方式来操作。为了避免当前节点对整个 CAN 总线的影响，所以应用软件只能在初始化模式下修改位时序寄存器 CAN_BTIMR。

表 24-5 CAN1 相关寄存器列表

名称	访问地址	描述	复位值
R32_CAN1_CTLR	0x40006400	CAN1 主控制寄存器	0x00010002
R32_CAN1_STATR	0x40006404	CAN1 主状态寄存器	0x00000C02
R32_CAN1_TSTATR	0x40006408	CAN1 发送状态寄存器	0x1C000000
R32_CAN1_RFIFO0	0x4000640C	CAN1 接收 FIFO0 控制和状态寄存器	0x00000000
R32_CAN1_RFIFO1	0x40006410	CAN1 接收 FIFO1 控制和状态寄存器	0x00000000
R32_CAN1_INTENR	0x40006414	CAN1 中断使能寄存器	0x00000000
R32_CAN1_ERRSR	0x40006418	CAN1 错误状态寄存器	0x00000000
R32_CAN1_BTIMR	0x4000641C	CAN1 位时序寄存器	0x01230000

R32_CAN1_TTCTLR	0x40006420	CAN1 时间触发控制寄存器	0x0000FFFF
R32_CAN1_TTCNT	0x40006424	CAN1 时间触发计数值寄存器	0x00000000

表 24-6 CAN2 相关寄存器列表

名称	访问地址	描述	复位值
R32_CAN2_CTLR	0x40006800	CAN2 主控制寄存器	0x00010002
R32_CAN2_STATR	0x40006804	CAN2 主状态寄存器	0x00000C02
R32_CAN2_TSTATR	0x40006808	CAN2 发送状态寄存器	0x1C000000
R32_CAN2_RFIFO0	0x4000680C	CAN2 接收 FIFO0 控制和状态寄存器	0x00000000
R32_CAN2_RFIFO1	0x40006810	CAN2 接收 FIFO1 控制和状态寄存器	0x00000000
R32_CAN2_INTENR	0x40006814	CAN2 中断使能寄存器	0x00000000
R32_CAN2_ERRSR	0x40006818	CAN2 错误状态寄存器	0x00000000
R32_CAN2_BTIMR	0x4000681C	CAN2 位时序寄存器	0x01230000
R32_CAN2_TTCTLR	0x40006820	CAN2 时间触发控制寄存器	0x0000FFFF
R32_CAN2_TTCNT	0x40006824	CAN2 时间触发计数值寄存器	0x00000000

表 24-7 CAN1 邮箱相关寄存器列表

名称	访问地址	描述	复位值
R32_CAN1_TXMIR0	0x40006580	CAN1 发送邮箱 0 标识符寄存器	X
R32_CAN1_TXMDTR0	0x40006584	CAN1 发送邮箱 0 数据长度和时间戳寄存器	X
R32_CAN1_TXMDLR0	0x40006588	CAN1 发送邮箱 0 低字节数据寄存器	X
R32_CAN1_TXMDHR0	0x4000658C	CAN1 发送邮箱 0 高字节数据寄存器	X
R32_CAN1_TXMIR1	0x40006590	CAN1 发送邮箱 1 标识符寄存器	X
R32_CAN1_TXMDTR1	0x40006594	CAN1 发送邮箱 1 数据长度和时间戳寄存器	X
R32_CAN1_TXMDLR1	0x40006598	CAN1 发送邮箱 1 低字节数据寄存器	X
R32_CAN1_TXMDHR1	0x4000659C	CAN1 发送邮箱 1 高字节数据寄存器	X
R32_CAN1_TXMIR2	0x400065A0	CAN1 发送邮箱 2 标识符寄存器	X
R32_CAN1_TXMDTR2	0x400065A4	CAN1 发送邮箱 2 数据长度和时间戳寄存器	X
R32_CAN1_TXMDLR2	0x400065A8	CAN1 发送邮箱 2 低字节数据寄存器	X
R32_CAN1_TXMDHR2	0x400065AC	CAN1 发送邮箱 2 高字节数据寄存器	X
R32_CAN1_RXMIR0	0x400065B0	CAN1 接收 FIFO0 邮箱标识符寄存器	X
R32_CAN1_RXMDTR0	0x400065B4	CAN1 接收 FIFO0 邮箱数据长度和时间戳寄存器	X
R32_CAN1_RXMDLR0	0x400065B8	CAN1 接收 FIFO0 邮箱低字节数据寄存器	X
R32_CAN1_RXMDHR0	0x400065BC	CAN1 接收 FIFO0 邮箱高字节数据寄存器	X
R32_CAN1_RXMIR1	0x400065C0	CAN1 接收 FIFO1 邮箱标识符寄存器	X
R32_CAN1_RXMDTR1	0x400065C4	CAN1 接收 FIFO1 邮箱数据长度和时间戳寄存器	X
R32_CAN1_RXMDLR1	0x400065C8	CAN1 接收 FIFO1 邮箱低字节数据寄存器	X
R32_CAN1_RXMDHR1	0x400065CC	CAN1 接收 FIFO1 邮箱高字节数据寄存器	X

表 24-8 CAN2 邮箱相关寄存器列表

名称	访问地址	描述	复位值
R32_CAN2_TXMIR0	0x40006980	CAN2 发送邮箱 0 标识符寄存器	X
R32_CAN2_TXMDTR0	0x40006984	CAN2 发送邮箱 0 数据长度和时间戳寄存器	X
R32_CAN2_TXMDLR0	0x40006988	CAN2 发送邮箱 0 低字节数据寄存器	X
R32_CAN2_TXMDHR0	0x4000698C	CAN2 发送邮箱 0 高字节数据寄存器	X
R32_CAN2_TXMIR1	0x40006990	CAN2 发送邮箱 1 标识符寄存器	X

R32_CAN2_TXMDTR1	0x40006994	CAN2 发送邮箱 1 数据长度和时间戳寄存器	X
R32_CAN2_TXMDLR1	0x40006998	CAN2 发送邮箱 1 低字节数据寄存器	X
R32_CAN2_TXMDHR1	0x4000699C	CAN2 发送邮箱 1 高字节数据寄存器	X
R32_CAN2_TXMIR2	0x400069A0	CAN2 发送邮箱 2 标识符寄存器	X
R32_CAN2_TXMDTR2	0x400069A4	CAN2 发送邮箱 2 数据长度和时间戳寄存器	X
R32_CAN2_TXMDLR2	0x400069A8	CAN2 发送邮箱 2 低字节数据寄存器	X
R32_CAN2_TXMDHR2	0x400069AC	CAN2 发送邮箱 2 高字节数据寄存器	X
R32_CAN2_RXMIR0	0x400069B0	CAN2 接收 FIFO0 邮箱标识符寄存器	X
R32_CAN2_RXMDTR0	0x400069B4	CAN2 接收 FIFO0 邮箱数据长度和时间戳寄存器	X
R32_CAN2_RXMDLR0	0x400069B8	CAN2 接收 FIFO0 邮箱低字节数据寄存器	X
R32_CAN2_RXMDHR0	0x400069BC	CAN2 接收 FIFO0 邮箱高字节数据寄存器	X
R32_CAN2_RXMIR1	0x400069C0	CAN2 接收 FIFO1 邮箱标识符寄存器	X
R32_CAN2_RXMDTR1	0x400069C4	CAN2 接收 FIFO1 邮箱数据长度和时间戳寄存器	X
R32_CAN2_RXMDLR1	0x400069C8	CAN2 接收 FIFO1 邮箱低字节数据寄存器	X
R32_CAN2_RXMDHR1	0x400069CC	CAN2 接收 FIFO1 邮箱高字节数据寄存器	X

表 24-9 CAN1 过滤器相关寄存器列表

名称	访问地址	描述	复位值
R32_CAN1_FCTLR	0x40006600	CAN1 过滤器主控制寄存器	0x2A1C0E01
R32_CAN1_FMCFGR	0x40006604	CAN1 过滤器模式寄存器	0x00000000
R32_CAN1_FSCFGR	0x4000660C	CAN1 过滤器位宽寄存器	0x00000000
R32_CAN1_FAFIFOR	0x40006614	CAN1 过滤器 FIFO 关联寄存器	0x00000000
R32_CAN1_FWR	0x4000661C	CAN1 过滤器激活寄存器	0x00000000
R32_CAN1_F0R1	0x40006640	CAN1 过滤器组 0 寄存器 1	X
R32_CAN1_F0R2	0x40006644	CAN1 过滤器组 0 寄存器 2	X
R32_CAN1_F1R1	0x40006648	CAN1 过滤器组 1 寄存器 1	X
R32_CAN1_F1R2	0x4000664C	CAN1 过滤器组 1 寄存器 2	X
R32_CAN1_F2R1	0x40006650	CAN1 过滤器组 2 寄存器 1	X
R32_CAN1_F2R2	0x40006654	CAN1 过滤器组 2 寄存器 2	X
R32_CAN1_F3R1	0x40006658	CAN1 过滤器组 3 寄存器 1	X
R32_CAN1_F3R2	0x4000665C	CAN1 过滤器组 3 寄存器 2	X
R32_CAN1_F4R1	0x40006660	CAN1 过滤器组 4 寄存器 1	X
R32_CAN1_F4R2	0x40006664	CAN1 过滤器组 4 寄存器 2	X
R32_CAN1_F5R1	0x40006668	CAN1 过滤器组 5 寄存器 1	X
R32_CAN1_F5R2	0x4000666C	CAN1 过滤器组 5 寄存器 2	X
R32_CAN1_F6R1	0x40006670	CAN1 过滤器组 6 寄存器 1	X
R32_CAN1_F6R2	0x40006674	CAN1 过滤器组 6 寄存器 2	X
R32_CAN1_F7R1	0x40006678	CAN1 过滤器组 7 寄存器 1	X
R32_CAN1_F7R2	0x4000667C	CAN1 过滤器组 7 寄存器 2	X
R32_CAN1_F8R1	0x40006680	CAN1 过滤器组 8 寄存器 1	X
R32_CAN1_F8R2	0x40006684	CAN1 过滤器组 8 寄存器 2	X
R32_CAN1_F9R1	0x40006688	CAN1 过滤器组 9 寄存器 1	X
R32_CAN1_F9R2	0x4000668C	CAN1 过滤器组 9 寄存器 2	X
R32_CAN1_F10R1	0x40006690	CAN1 过滤器组 10 寄存器 1	X
R32_CAN1_F10R2	0x40006694	CAN1 过滤器组 10 寄存器 2	X

R32_CAN1_F11R1	0x40006698	CAN1 过滤器组 11 寄存器 1	X
R32_CAN1_F11R2	0x4000669C	CAN1 过滤器组 11 寄存器 2	X
R32_CAN1_F12R1	0x400066A0	CAN1 过滤器组 12 寄存器 1	X
R32_CAN1_F12R2	0x400066A4	CAN1 过滤器组 12 寄存器 2	X
R32_CAN1_F13R1	0x400066A8	CAN1 过滤器组 13 寄存器 1	X
R32_CAN1_F13R2	0x400066AC	CAN1 过滤器组 13 寄存器 2	X
R32_CAN1_F14R1	0x400066B0	CAN1 过滤器组 14 寄存器 1	X
R32_CAN1_F14R2	0x400066B4	CAN1 过滤器组 14 寄存器 2	X
R32_CAN1_F15R1	0x400066B8	CAN1 过滤器组 15 寄存器 1	X
R32_CAN1_F15R2	0x400066BC	CAN1 过滤器组 15 寄存器 2	X
R32_CAN1_F16R1	0x400066C0	CAN1 过滤器组 16 寄存器 1	X
R32_CAN1_F16R2	0x400066C4	CAN1 过滤器组 16 寄存器 2	X
R32_CAN1_F17R1	0x400066C8	CAN1 过滤器组 17 寄存器 1	X
R32_CAN1_F17R2	0x400066CC	CAN1 过滤器组 17 寄存器 2	X
R32_CAN1_F18R1	0x400066D0	CAN1 过滤器组 18 寄存器 1	X
R32_CAN1_F18R2	0x400066D4	CAN1 过滤器组 18 寄存器 2	X
R32_CAN1_F19R1	0x400066D8	CAN1 过滤器组 19 寄存器 1	X
R32_CAN1_F19R2	0x400066DC	CAN1 过滤器组 19 寄存器 2	X
R32_CAN1_F20R1	0x400066E0	CAN1 过滤器组 20 寄存器 1	X
R32_CAN1_F20R2	0x400066E4	CAN1 过滤器组 20 寄存器 2	X
R32_CAN1_F21R1	0x400066E8	CAN1 过滤器组 21 寄存器 1	X
R32_CAN1_F21R2	0x400066EC	CAN1 过滤器组 21 寄存器 2	X
R32_CAN1_F22R1	0x400066F0	CAN1 过滤器组 22 寄存器 1	X
R32_CAN1_F22R2	0x400066F4	CAN1 过滤器组 22 寄存器 2	X
R32_CAN1_F23R1	0x400066F8	CAN1 过滤器组 23 寄存器 1	X
R32_CAN1_F23R2	0x400066FC	CAN1 过滤器组 23 寄存器 2	X
R32_CAN1_F24R1	0x40006700	CAN1 过滤器组 24 寄存器 1	X
R32_CAN1_F24R2	0x40006704	CAN1 过滤器组 24 寄存器 2	X
R32_CAN1_F25R1	0x40006708	CAN1 过滤器组 25 寄存器 1	X
R32_CAN1_F25R2	0x4000670C	CAN1 过滤器组 25 寄存器 2	X
R32_CAN1_F26R1	0x40006710	CAN1 过滤器组 26 寄存器 1	X
R32_CAN1_F26R2	0x40006714	CAN1 过滤器组 26 寄存器 2	X
R32_CAN1_F27R1	0x40006718	CAN1 过滤器组 27 寄存器 1	X
R32_CAN1_F27R2	0x4000671C	CAN1 过滤器组 27 寄存器 2	X

表 24-10 CAN2 过滤器相关寄存器列表

名称	访问地址	描述	复位值
R32_CAN2_FCTLR	0x40006A00	CAN2 过滤器主控制寄存器	0x2A1C0E01
R32_CAN2_FMCFGR	0x40006A04	CAN2 过滤器模式寄存器	0x00000000
R32_CAN2_FSCFGR	0x40006A0C	CAN2 过滤器位宽寄存器	0x00000000
R32_CAN2_FAFIFOR	0x40006A14	CAN2 过滤器 FIFO 关联寄存器	0x00000000
R32_CAN2_FWR	0x40006A1C	CAN2 过滤器激活寄存器	0x00000000
R32_CAN2_F0R1	0x40006A40	CAN2 过滤器组 0 寄存器 1	X
R32_CAN2_F0R2	0x40006A44	CAN2 过滤器组 0 寄存器 2	X
R32_CAN2_F1R1	0x40006A48	CAN2 过滤器组 1 寄存器 1	X

R32_CAN2_F1R2	0x40006A4C	CAN2 过滤器组 1 寄存器 2	X
R32_CAN2_F2R1	0x40006A50	CAN2 过滤器组 2 寄存器 1	X
R32_CAN2_F2R2	0x40006A54	CAN2 过滤器组 2 寄存器 2	X
R32_CAN2_F3R1	0x40006A58	CAN2 过滤器组 3 寄存器 1	X
R32_CAN2_F3R2	0x40006A5C	CAN2 过滤器组 3 寄存器 2	X
R32_CAN2_F4R1	0x40006A60	CAN2 过滤器组 4 寄存器 1	X
R32_CAN2_F4R2	0x40006A64	CAN2 过滤器组 4 寄存器 2	X
R32_CAN2_F5R1	0x40006A68	CAN2 过滤器组 5 寄存器 1	X
R32_CAN2_F5R2	0x40006A6C	CAN2 过滤器组 5 寄存器 2	X
R32_CAN2_F6R1	0x40006A70	CAN2 过滤器组 6 寄存器 1	X
R32_CAN2_F6R2	0x40006A74	CAN2 过滤器组 6 寄存器 2	X
R32_CAN2_F7R1	0x40006A78	CAN2 过滤器组 7 寄存器 1	X
R32_CAN2_F7R2	0x40006A7C	CAN2 过滤器组 7 寄存器 2	X
R32_CAN2_F8R1	0x40006A80	CAN2 过滤器组 8 寄存器 1	X
R32_CAN2_F8R2	0x40006A84	CAN2 过滤器组 8 寄存器 2	X
R32_CAN2_F9R1	0x40006A88	CAN2 过滤器组 9 寄存器 1	X
R32_CAN2_F9R2	0x40006A8C	CAN2 过滤器组 9 寄存器 2	X
R32_CAN2_F10R1	0x40006A90	CAN2 过滤器组 10 寄存器 1	X
R32_CAN2_F10R2	0x40006A94	CAN2 过滤器组 10 寄存器 2	X
R32_CAN2_F11R1	0x40006A98	CAN2 过滤器组 11 寄存器 1	X
R32_CAN2_F11R2	0x40006A9C	CAN2 过滤器组 11 寄存器 2	X
R32_CAN2_F12R1	0x40006AA0	CAN2 过滤器组 12 寄存器 1	X
R32_CAN2_F12R2	0x40006AA4	CAN2 过滤器组 12 寄存器 2	X
R32_CAN2_F13R1	0x40006AA8	CAN2 过滤器组 13 寄存器 1	X
R32_CAN2_F13R2	0x40006AAC	CAN2 过滤器组 13 寄存器 2	X
R32_CAN2_F14R1	0x40006AB0	CAN2 过滤器组 14 寄存器 1	X
R32_CAN2_F14R2	0x40006AB4	CAN2 过滤器组 14 寄存器 2	X
R32_CAN2_F15R1	0x40006AB8	CAN2 过滤器组 15 寄存器 1	X
R32_CAN2_F15R2	0x40006ABC	CAN2 过滤器组 15 寄存器 2	X
R32_CAN2_F16R1	0x40006AC0	CAN2 过滤器组 16 寄存器 1	X
R32_CAN2_F16R2	0x40006AC4	CAN2 过滤器组 16 寄存器 2	X
R32_CAN2_F17R1	0x40006AC8	CAN2 过滤器组 17 寄存器 1	X
R32_CAN2_F17R2	0x40006ACC	CAN2 过滤器组 17 寄存器 2	X
R32_CAN2_F18R1	0x40006AD0	CAN2 过滤器组 18 寄存器 1	X
R32_CAN2_F18R2	0x40006AD4	CAN2 过滤器组 18 寄存器 2	X
R32_CAN2_F19R1	0x40006AD8	CAN2 过滤器组 19 寄存器 1	X
R32_CAN2_F19R2	0x40006ADC	CAN2 过滤器组 19 寄存器 2	X
R32_CAN2_F20R1	0x40006AE0	CAN2 过滤器组 20 寄存器 1	X
R32_CAN2_F20R2	0x40006AE4	CAN2 过滤器组 20 寄存器 2	X
R32_CAN2_F21R1	0x40006AE8	CAN2 过滤器组 21 寄存器 1	X
R32_CAN2_F21R2	0x40006AEC	CAN2 过滤器组 21 寄存器 2	X
R32_CAN2_F22R1	0x40006AF0	CAN2 过滤器组 22 寄存器 1	X
R32_CAN2_F22R2	0x40006AF4	CAN2 过滤器组 22 寄存器 2	X
R32_CAN2_F23R1	0x40006AF8	CAN2 过滤器组 23 寄存器 1	X
R32_CAN2_F23R2	0x40006AFC	CAN2 过滤器组 23 寄存器 2	X

R32_CAN2_F24R1	0x40006B00	CAN2 过滤器组 24 寄存器 1	X
R32_CAN2_F24R2	0x40006B04	CAN2 过滤器组 24 寄存器 2	X
R32_CAN2_F25R1	0x40006B08	CAN2 过滤器组 25 寄存器 1	X
R32_CAN2_F25R2	0x40006B0C	CAN2 过滤器组 25 寄存器 2	X
R32_CAN2_F26R1	0x40006B10	CAN2 过滤器组 26 寄存器 1	X
R32_CAN2_F26R2	0x40006B14	CAN2 过滤器组 26 寄存器 2	X
R32_CAN2_F27R1	0x40006B18	CAN2 过滤器组 27 寄存器 1	X
R32_CAN2_F27R2	0x40006B1C	CAN2 过滤器组 27 寄存器 2	X

24.7.1 CANx 主控制寄存器（CANx_CTLR）（x=1/2）

偏移地址：0x04

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET	Reserved							TTCM	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
16	DBF	RW	调试是否禁止 CAN 总线工作 1：调试时，CAN 的收发被禁止，但是接收 FIFO 的控制和读写操作一切正常； 0：调试时，CAN 控制器正常工作。	1
15	RESET	RW1	CAN 控制器软件复位请求，该位写 0 无效 1：对 CAN 控制器进行复位，复位后控制器进入睡眠模式，然后硬件自动清 0； 0：CAN 控制器正常状态。	0
[14:8]	Reserved	RO	保留	0
7	TTCM	RW	是否允许时间触发模式 1：使能时间触发模式； 0：禁止时间触发模式。 时间触发模式主要是配合 TTCAN 协议使用。	0
6	ABOM	RW	离线自动退出控制 1：硬件检测到 128 次连续 11 个隐性位，自动退出离线状态； 0：需要软件操作寄存器 CAN_CTLR 的 INRQ 位置 1 然后清 0，当检测到 128 次连续 11 个隐性位后，退出离线状态。	0
5	AWUM	RW	CAN 控制器自动唤醒使能 1：当检测到报文时，硬件自动唤醒，寄存器 CAN_STATR 的 SLEEP 和 SLAK 位自动清 0； 0：需要软件操作寄存器 CAN_CTLR 的 SLEEP 位清 0，唤醒 CAN 控制器。	0
4	NART	RW	报文自动重传功能禁止 1：无论发送成功与否，报文只能被发送一次；	0

			0：CAN 控制器一直重传至发送成功为止。	
3	RFLM	RW	接收 FIFO 报文锁定模式使能 1：当接收 FIFO 溢出时，已接收邮箱报文未读出，邮箱未释放时，新接收到的报文被丢弃； 0：当接收 FIFO 溢出时，已接收邮箱报文未读出，邮箱未释放时，新接收到的报文会覆盖原有报文。	0
2	TXFP	RW	发送邮箱优先级方式选择 1：优先级由发送请求的先后顺序决定； 0：优先级由报文标识符来决定。	0
1	SLEEP	RW	睡眠模式请求位 1：置 1 请求 CAN 控制器进入睡眠模式，当前活动完成后，控制器进入睡眠模式，若 AWUM 位置 1，则在接收到报文时，控制器把 SLEEP 位清 0； 0：软件清 0 后，控制器退出睡眠模式。	1
0	INRQ	RW	初始化模式请求位 1：置 1 请求 CAN 控制器进入初始化模式，当前活动完成后，控制器进入初始化模式，硬件对寄存器 CAN_STATR 的 INAK 位置 1； 0：置 0 请求 CAN 控制器退出初始化模式，进入正常模式，硬件对寄存器 CAN_STATR 的 INAK 位清 0。	0

24.7.2 CANx 主状态寄存器（CANx_STATR）（x=1/2）

偏移地址：0x04

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				RX	SAMP	RXM	TXM	Reserved			SLAKI	WKUI	ERRI	SLAK	INAK

位	名称	访问	描述	复位值
[31:12]	Reserved	RO	保留。	0
11	RX	RO	CAN 控制器接收引脚 RX 当前实际电平。	1
10	SAMP	RO	CAN 控制器接收引脚 RX 上一个接收位的电平	1
9	RXM	RO	接收模式查询位 1：当前 CAN 控制器为接收模式； 0：当前 CAN 控制器非接收模式。	0
8	TXM	RO	发送模式查询位 1：当前 CAN 控制器为发送模式； 0：当前 CAN 控制器非发送模式。	0
[7:5]	Reserved	RO	保留	0
4	SLAKI	RW1	睡眠中断使能时，即寄存器 CAN_INTENR 的 SLKIE 位置 1 时，中断产生标志位，写 1 清 0，写 0 无效。	0

			1：进入睡眠模式时，中断产生，硬件置 1； 0：退出睡眠模式时，硬件清 0 也可软件清 0。	
3	WKUI	RW1	唤醒中断标志位。当寄存器 CAN_INTENR 的 WKUI 位置 1 时,若 CAN 控制器处于睡眠模式时，检测到 SOF 位，则硬件置 1。软件置 1 清 0，置 0 无效。	0
2	ERRI	RW1	出错中断状态标志位。当寄存器 CAN_INTENR 的 ERRIE 位置 1 时，产生错误及状态变化中断。该位软件置 1 清 0，置 0 无效。	0
1	SLAK	RO	睡眠模式指示位。 1：CAN 控制器正处于睡眠模式； 0：CAN 控制器不在睡眠模式。	1
0	INAK	RO	初始化模式指示位。 1：CAN 控制器正在初始化模式； 0：CAN 控制器工作在非初始化模式。	0

24.7.3 CANx 发送状态寄存器（CANx_TSTATR）（x=1/2）

偏移地址：0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOW2	LOW1	LOW0	TME2	TME1	TME0	CODE[1:0]		ABRQ2	Reserved			TERR2	ALST2	TXOK2	RQCP2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRQ1	Reserved			TERR1	ALST1	TXOK1	RQCP1	ABRQ0	Reserved			TERR0	ALST0	TXOK0	RQCP0

位	名称	访问	描述	复位值
31	LOW2	RO	发送邮箱 2 的最低优先级标志位 1：表示发送邮箱 2 的优先级最低； 0：表示发送邮箱 2 的优先级非最低。	0
30	LOW1	RO	发送邮箱 1 的最低优先级标志位 1：表示发送邮箱 1 的优先级最低； 0：表示发送邮箱 1 的优先级非最低。	0
29	LOW0	RO	发送邮箱 0 的最低优先级标志位 1：表示发送邮箱 0 的优先级最低； 0：表示发送邮箱 0 的优先级非最低。	0
28	TME2	RO	发送邮箱 2 的空标志位 1：表示发送邮箱 2 无等待发送报文； 0：表示发送邮箱 2 有等待发送报文。	1
27	TME1	RO	发送邮箱 1 的空标志位 1：表示发送邮箱 1 无等待发送报文； 0：表示发送邮箱 1 有等待发送报文。	1
26	TME0	RO	发送邮箱 0 的空标志位 1：表示发送邮箱 0 无等待发送报文； 0：表示发送邮箱 0 有等待发送报文。	1
[25:24]	CODE	RO	邮箱编号 当有 1 个以上邮箱为空时，表示下一个为空的邮箱号；当邮箱全空时，表示优先级最低	0

			的邮箱号。	
23	ABRQ2	RW1	发送邮箱 2 的发送中止请求。软件置 1，可以中止邮箱 2 的发送请求，发送报文被清除时硬件清 0，若邮箱 2 清空，软件置 1 无效。	0
[22:20]	Reserved	RO	保留	0
19	TERR2	RW1	发送邮箱 2 发送失败标志位，当发送邮箱 2 发送失败，该位自动置 1。软件置 1 清 0，软件写 0 无效。	0
18	ALST2	RW1	发送邮箱 2 仲裁失败标志位，当发送邮箱 2 仲裁优先级低导致发送失败，该位自动置 1。软件置 1 清 0，软件写 0 无效。	0
17	TXOK2	RW1	发送邮箱 2 发送成功标志位 1：上次发送成功； 0：上次发送失败。 软件置 1 清 0，软件写 0 无效。	0
16	RQCP2	RW	发送邮箱 2 请求完成标志位，当发送邮箱 2 的发送或中止请求完成时，该位自动置 1。软件置 1 清 0，软件写 0 无效。	0
15	ABRQ1	RWO	发送邮箱 1 的发送中止请求。软件置 1，可以中止邮箱 1 的发送请求，发送报文被清除时硬件清 0。软件写 0 无效。	0
[14:12]	Reserved	RO	保留	0
11	TERR1	RW1	发送邮箱 1 发送失败标志位，当发送邮箱 1 发送失败，该位自动置 1。软件置 1 清 0，软件写 0 无效。	0
10	ALST1	RW1	发送邮箱 1 仲裁失败标志位，当发送邮箱 1 仲裁优先级低导致发送失败，该位自动置 1。	0
9	TXOK1	RW1	发送邮箱 1 发送成功标志位 1：上次发送成功； 0：上次发送失败。 软件置 1 清 0，软件写 0 无效。	0
8	RQCP1	RW	发送邮箱 1 请求完成标志位，当发送邮箱 1 的发送或中止请求完成时，该位自动置 1。软件置 1 清 0，软件写 0 无效。	0
7	ABRQ0	RWO	发送邮箱 0 的发送中止请求。软件置 1，可以中止邮箱 0 的发送请求，发送报文被清除时硬件清 0。软件写 0 无效。	0
[6:4]	Reserved	RO	保留	0
3	TERR0	RW1	发送邮箱 0 发送失败标志位，当发送邮箱 0 发送失败，该位自动置 1。软件置 1 清 0，软件写 0 无效。	0
2	ALST0	RW1	发送邮箱 0 仲裁失败标志位，当发送邮箱 0 仲裁优先级低导致发送失败，该位自动置 1。软件置 1 清 0，软件写 0 无效。	0
1	TXOK0	RW1	发送邮箱 0 发送成功标志位 1：上次发送成功；	0

			0：上次发送失败。 软件置 1 清 0，软件写 0 无效。	
0	RQCP0	RW	发送邮箱 0 请求完成标志位，当发送邮箱 0 的发送或中止请求完成时，该位自动置 1。 软件置 1 清 0，软件写 0 无效。	0

24.7.4 CANx 接收 FIFO 0 状态寄存器（CANx_RFIFO0）（x=1/2）

偏移地址：0x0C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RFOM0	FOVR0	FULL0	Reserved	FMP0[1:0]	

位	名称	访问	描述	复位值
[31:6]	Reserved	RO	保留。	0
5	RFOM0	RW1	软件对该位置 1，则释放接收 FIFO_0 的当前邮箱报文，释放完后自动清 0，软件写 0 无效。	0
4	FOVR0	RW1	接收 FIFO_0 溢出标志位。当 FIFO_0 中有三个报文时，又接到新报文，硬件置 1。该位需要软件置 1 清 0，软件写 0 无效。	0
3	FULL0	RW1	接收 FIFO_0 满标志位。当 FIFO_0 中有三个报文时，硬件置 1。该位需要软件置 1 清 0，软件写 0 无效。	0
2	Reserved	RO	保留。	0
[1:0]	FMP0	RO	接收 FIFO_0 报文数目。	0

24.7.5 CANx 接收 FIFO 1 状态寄存器（CANx_RFIFO1）（x=1/2）

偏移地址：0x10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RFOM1	FOVR1	FULL1	Reserved	FMP1[1:0]	

位	名称	访问	描述	复位值
[31:6]	Reserved	RO	保留。	0
5	RFOM1	RW1	软件对该位置 1，则释放接收 FIFO_1 的当前邮箱报文，释放完后自动清 0，软件写 0 无效。	0
4	FOVR1	RW1	接收 FIFO_1 溢出标志位。当 FIFO_1 中有三个报文时，又接到新报文，硬件置 1。该位需要软件置 1 清 0，软件写 0 无效。	0

3	FULL1	RW1	接收 FIFO_1 满标志位。当 FIFO_1 中有三个报文时，硬件置 1。该位需要软件置 1 清 0，软件写 0 无效。	0
2	Reserved	RF	保留	0
[1:0]	FMP1	RO	接收 FIFO_1 报文数目。	0

24.7.6 CANx 中断使能寄存器（CANx_INTENR）（x=1/2）

偏移地址：0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Reserved			LECIE	BOFIE	EPVIE	EWGIE	Reserved	FOVIE1	FFIE1	FMPIE1	FOVIE0	FFIE0	FMPIE0	TMEIE

位	名称	访问	描述	复位值
[31:18]	Reserved	RO	保留。	0
17	SLKIE	RW	睡眠中断使能位。 1：进入睡眠状态时，产生中断； 0：进入睡眠状态时，不产生中断。	0
16	WKUIE	RW	唤醒中断使能位。 1：当 CAN 控制器被唤醒时，产生中断； 0：当 CAN 控制器被唤醒时，不产生中断。	0
15	ERRIE	RW	错误中断使能位，CAN 错误中断总使能位。 1：当 CAN 控制器产生错误时，产生中断； 0：当 CAN 控制器产生错误时，不产生中断。	0
[14:12]	Reserved	RF	保留。	0
11	LECIE	RW	上次错误号中断使能位。 1：检测到错误时，硬件更新 LEC[2:0]，更新 ERRI 位为 1，触发错误中断； 0：检测到错误时，硬件更新 LEC[2:0]，不更新 ERRI 位,不触发错误中断。	0
10	BOFIE	RW	离线中断使能位。 1：进入离线状态时，更新 ERRI 位为 1，触发错误中断； 0：进入离线状态时，不更新 ERRI 位，不触发错误中断。	0
9	EPVIE	RW	错误被动中断使能位。 1：进入错误被动状态时，更新 ERRI 位为 1，触发错误中断； 0：进入错误被动状态时，不更新 ERRI 位，不触发错误中断。	0
8	EWGIE	RW	错误警告中断使能位。 1：出错次数达到警告阈值时，更新 ERRI 位为 1，触发错误中断； 0：出错次数达到警告阈值时，不更新 ERRI	0

			位，不触发错误中断。	
7	Reserved	RF	保留。	0
6	FOVIE1	RW	接收 FIFO_1 溢出中断使能位。 1：当 FIFO_1 溢出，触发 FIFO_1 中断； 0：当 FIFO_1 溢出，不触发 FIFO_1 中断。	0
5	FFIE1	RW	接收 FIFO_1 满中断使能位。 1：当 FIFO_1 为满，触发 FIFO_1 中断； 0：当 FIFO_1 为满，不触发 FIFO_1 中断。	0
4	FMPIE1	RW	接收 FIFO_1 消息挂号中断使能位。 1：当 FIFO_1 更新 FMP 位，且不为 0，触发 FIFO_1 中断； 0：当 FIFO_1 更新 FMP 位，且不为 0，不触发 FIFO_1 中断。	0
3	FOVIE0	RW	接收 FIFO_0 溢出中断使能位。 1：当 FIFO_0 溢出，触发 FIFO_0 中断； 0：当 FIFO_0 溢出，不触发 FIFO_0 中断。	0
2	FFIE0	RW	接收 FIFO_0 满中断使能位。 1：当 FIFO_0 为满，触发 FIFO_0 中断； 0：当 FIFO_0 为满，不触发 FIFO_0 中断。	0
1	FMPIE0	RW	接收 FIFO_0 消息挂号中断使能位。 1：当 FIFO_0 更新 FMP 位，且不为 0，触发 FIFO_0 中断； 0：当 FIFO_0 更新 FMP 位，且不为 0，不触发 FIFO_0 中断。	0
0	TMEIE	RW	发送邮箱空中断。 1：当发送邮箱为空时，产生中断； 0：当发送邮箱为空时，不产生中断。	0

24.7.7 CANx 错误状态寄存器（CANx_ERRSR）（x=1/2）

偏移地址：0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REC[7:0]								TEC[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									LEC[2:0]			Reserved	BOFF	EPVF	EWGF

位	名称	访问	描述	复位值
[31:24]	REC	RO	接收错误计数器。 当 CAN 接收出错时，根据出错条件，该计数器加 1 或 8；接收成功后，该计数器减 1 或设为 120(错误计数值大于 127)。计数器值超过 127 时，CAN 进入错误被动状态。	0
[23:16]	TEC	RO	发送错误计数器。 当 CAN 发送出错时，根据出错条件，该计数器加 1 或 8；发送成功后，该计数器减 1 或	0

			设为 120(错误计数值大于 127)。计数器值超过 127 时，CAN 进入错误被动状态。	
[15:7]	Reserved	RO	保留。	0
[6:4]	LEC{2:0}	RW	上次错误代号。 检测到 CAN 总线上发送错误时，控制器根据出错情况设置，当正确收发报文时，置 000b。 000：无错误； 001：位填充错误； 010：FORM 格式错误； 011：ACK 确认错误； 100：隐性位错误； 101：显性位错误； 110：CRC 错误； 111：软件设置。 通常应用软件读取到错误时，把代号设置为 111b，可以检测到代号更新。	0
3	Reserved	RO	保留。	0
2	BOFF	RO	离线状态标志位。 当 CAN 控制器进入离线状态时，硬件自动置 1；退出离线状态时，硬件自动清 0。	0
1	EPVF	RO	错误被动标志位。 当收发错误计数器达到错误被动阈值时，即大于 127 时，硬件置 1。	0
0	EWGF	RO	错误警告标志位。 当收发错误计数器达到警告阈值时，即大于等于 96 时，硬件置 1。	0

24.7.8 CANx 位时序寄存器（CANx_BTIMR）(x=1/2)

偏移地址：0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SILM	LBKM	Reserved				SJW[1:0]		Reserved	TS2[2:0]			TS1[3:0]			

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

位	名称	访问	描述	复位值
31	SILM	RW	静默模式设置位。 1：进入静默模式； 0：退出静默模式。	0
30	LBKM	RW	环回模式设置位。 1：进入环回模式； 0：退出环回模式。	0
[29:26]	Reserved	RO	保留。	0
[25:24]	SJW	RW	定义了重新同步跳转宽度设置值。 实现重新同步时，位中可以延长和缩小的最	01b

			小时间单元数量上限，实际值为（SJW[1:0]+1），范围可设置为 1 到 4 个最小时间单元。	
23	Reserved	RO	保留。	0
[22:20]	TS2	RW	时间段 2 设置值。 定义了时间段 2 占用了多少个最小时间单元，实际值为（TS2[1:0]+1）。	010b
[19:16]	TS1	RW	时间段 1 设置值。 定义了时间段 1 占用了多少个最小时间单元，实际值为（TS1[1:0]+1）。	0011b
[15:10]	Reserved	RO	保留。	0
[9:0]	BRP	RW	最小时间单元长度设置值 Tq = (BRP[9:0]+1) × t_{pclk}	0

24.7.9 CANx 时间触发控制寄存器（CANx_TTCTLR）(x=1/2)

偏移地址：0x20

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

位	名称	访问	描述	复位值
[31:18]	Reserved	RO	保留。	0
17	MODE	RW	时间触发模式选择位。 1：增强模式； 0：默认模式。	0
16	TIMRST	WZ	内部计数器复位控制位。 写 1 复位内部计数器，硬件自动清 0	0
[15:0]	TIMCMV	RW	内部计数器计数终值	0XFFFF

24.7.10 CANx 时间触发计数值寄存器（CANx_TTCNT）(x=1/2)

偏移地址：0x24

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	TIMCNT	RW	时间触发计数值	0

24.7.11 CANx 发送邮箱标识符寄存器（CANx_TXMIRy）（x=0/1,y=0/1/2）

偏移地址：0x180,0x190,0x1A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STID[10:0]/EXID[28:18]											EXID[17:13]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXID[12:0]													IDE	RTR	TXRQ

位	名称	访问	描述	复位值
[31:21]	STID_EXIDH	RW	标准标识符或扩展标识符的高 11 位。	X
[20:3]	EXIDL	RW	扩展标识符的低 18 位。	X
2	IDE	RW	标识符选择标志位。 1：选用扩展标识符； 0：选用标准标识符。	X
1	RTR	RW	远程帧(也称遥控帧)选择标志位。 1：当前为远程帧； 0：当前为数据帧。	X
0	TXRQ	RW	数据发送请求标志位。 软件置 1 时，请求发送邮箱里的数据，发送完毕邮箱为空时，硬件清 0。	0

24.7.12 CANx 发送邮箱数据长度和时间戳寄存器（CANx_TXMDTRy）（x=0/1,y=0/1/2）

偏移地址：0x184,0x194,0x1A4

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

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

位	名称	访问	描述	复位值
[31:16]	TIME	RW	用于发送报文 SOF 时刻的 16 位定时器值。	X
[15:9]	Reserved	RO	保留。	X
8	TGT	RW	报文时间戳发送选择标志位。该位在 TTCM 置 1，并报文长度为 8 时有效。 1：发送时间戳，值为 TIME[15:0]的即时值，替换了 8 字节报文的最后两个字节； 0：不发送时间戳。	X
[7:4]	Reserved	RO	保留。	X
[3:0]	DLC	RW	数据帧的数据长度或远程帧请求数据长度 数据长度可设置范围为 0 到 8。	X

24.7.13 ANx 发送邮箱低字节数据寄存器（CANx_TXMDLRy）（x=0/1,y=0/1/2）

偏移地址：0x188,0x198,0x1A8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							

位	名称	访问	描述	复位值
[31:24]	DATA3	RW	发送数据字节 3 的内容。	X
[23:16]	DATA2	RW	发送数据字节 2 的内容。	X
[15:8]	DATA1	RW	发送数据字节 1 的内容。	X
[7:0]	DATA0	RW	发送数据字节 0 的内容。	X

24.7.14 CANx 发送邮箱高字节数据寄存器（CANx_TXMDHRy）（x=0/1,y=0/1/2）

偏移地址：0x18C,0x19C,0x1AC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							

位	名称	访问	描述	复位值
[31:24]	DATA7	RW	发送数据字节 7 的内容。	X
[23:16]	DATA6	RW	发送数据字节 6 的内容。	X
[15:8]	DATA5	RW	发送数据字节 5 的内容。	X
[7:0]	DATA4	RW	发送数据字节 4 的内容。	X

24.7.15 CANx 接收邮箱标识符寄存器（CANx_RXMIRy）（x=0/1,y=0/1）

偏移地址：0x1B0,0x1C0

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

位	名称	访问	描述	复位值
[31:21]	STID_EXIDH	RO	标准标识符或扩展标识符的高 11 位。	X
[20:3]	EXIDL	RO	扩展标识符的低 18 位。	X
2	IDE	RO	标识符选择标志位。 1：选用扩展标识符； 0：选用标准标识符。	X
1	RTR	RO	远程帧(也称遥控帧)选择标志位。 1：当前为远程帧； 0：当前为数据帧。	X

0	Reserved	RO	保留。	X

24.7.16 CANx 接收邮箱数据长度和时间戳寄存器（CANx_RXMDTRy）（x=0/1,y=0/1）

偏移地址：0x1B4,0x1C4

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

位	名称	访问	描述	复位值
[31:16]	TIME	RO	用于接收报文 SOF 时刻的 16 位定时器值。	X
[15:8]	FMI	RO	报文所匹配的过滤器编号。	X
[7:4]	Reserved	RO	保留。	X
[3:0]	DLC	RO	接收报文数据长度。 数据帧长度 0 到 8，远程帧为 0。	X

24.7.17 CANx 接收邮箱低字节数据寄存器（CANx_RXMDLRy）（x=0/1,y=0/1）

偏移地址：0x1B8,0x1C8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							

位	名称	访问	描述	复位值
[31:24]	DATA3[7:0]	RO	接收报文的数据字节 3。	X
[23:16]	DATA2[7:0]	RO	接收报文的数据字节 2。	X
[15:8]	DATA1[7:0]	RO	接收报文的数据字节 1。	X
[7:0]	DATA0[7:0]	RO	接收报文的数据字节 0。	X

24.7.18 CANx 接收邮箱高字节数据寄存器（CANx_RXMDHRy）（x=0/1,y=0/1）

偏移地址：0x1BC,0x1CC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							

位	名称	访问	描述	复位值
[31:24]	DATA7[7:0]	RO	接收报文的数据字节 7。	X
[23:16]	DATA6[7:0]	RO	接收报文的数据字节 6。	X
[15:8]	DATA5[7:0]	RO	接收报文的数据字节 5。	X
[7:0]	DATA4[7:0]	RO	接收报文的数据字节 4。	X

24.7.19 CANx 过滤器主控制寄存器（CANx_FCTLR）（x=0/1）

偏移地址：0x200

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CAN2SB[5:0]						Reserved							FINT

位	名称	访问	描述	复位值
[31:14]	Reserved	RO	保留。	复位值
[13:8]	CAN2SB[5:0]	RW	CAN2 过滤器开始组（取值范围 1-27）。	01110b
[7:1]	Reserved	RO	保留。	复位值
0	FINT	RW	过滤器初始化模式使能标志位。 1：过滤器组为初始化模式； 0：过滤器组为正常模式。	1

24.7.20 CANx 过滤器模式寄存器（CANx_FMCFGR）（x=0/1）

偏移地址：0x204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FBM27	FBM26	FBM25	FBM24	FBM23	FBM22	FBM21	FBM20	FBM19	FBM18	FBM17	FBM16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBM15	FBM14	FBM13	FBM12	FBM11	FBM10	FBM9	FBM8	FBM7	FBM6	FBM5	FBM4	FBM3	FBM2	FBM1	FBM0

位	名称	访问	描述	复位值
[31:28]	Reserved	RO	保留。	0
[27:0]	FBMx	RW	过滤器组 x 的工作模式控制位，FINT 为 1 才能写入。 1：过滤器组 x 的寄存器为屏蔽位模式； 0：过滤器组 x 的寄存器为标识符列表模式。	0

24.7.21 CANx 过滤器位宽寄存器（CANx_FSCFGR）（x=0/1）

偏移地址：0x20C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FSC27	FSC26	FSC25	FSC24	FSC23	FSC22	FSC21	FSC20	FSC19	FSC18	FSC17	FSC16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSC15	FSC14	FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0

位	名称	访问	描述	复位值
[31:28]	Reserved	RO	保留。	0
[27:0]	FSCx	RW	过滤器组 x 的位宽控制位,FINT 为 1 才能写入。 1：过滤器组 x 的寄存器为单个 32 位； 0：过滤器组 x 的寄存器为 2 个 16 位。	0

24.7.22 CANx 过滤器 FIFO 关联寄存器（CANx_FAFIFOR）（x=0/1）

偏移地址：0x214

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FFA27	FFA26	FFA25	FFA24	FFA23	FFA22	FFA21	FFA20	FFA19	FFA18	FFA17	FFA16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFA15	FFA14	FFA13	FFA12	FFA11	FFA10	FFA9	FFA8	FFA7	FFA6	FFA5	FFA4	FFA3	FFA2	FFA1	FFA0

位	名称	访问	描述	复位值
[31:28]	Reserved	RO	保留。	0
[27:0]	FFAx	RW	过滤器组 x 的关联 FIFO 控制位，FINT 为 1 才能写入。 1：过滤器组 x 被关联到 FIFO_1； 0：过滤器组 x 被关联到 FIFO_0。	0

24.7.23 CANx 过滤器激活寄存器（CANx_FWR）（x=0/1）

偏移地址：0x21C

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

位	名称	访问	描述	复位值
[31:28]	Reserved	RO	保留。	0
[27:0]	FACTx	RW	过滤器组 x 的激活控制位。 1：过滤器组 x 激活； 0：过滤器组 x 禁用。	0

24.7.24 CANx 过滤器组的过滤寄存器（CANx_FiRy）（x=1/2, i=0-27, y=1/2）

偏移地址：0x240-0x31C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FB31	FB30	FB29	FB28	FB27	FB26	FB25	FB24	FB23	FB22	FB21	FB20	FB19	FB18	FB17	FB16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB15	FB14	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0

位	名称	访问	描述	复位值
[31:0]	FB	RW	过滤器组中寄存器的标志位，FINT 为 1 才能写入。 标识符模式下： 1：相应位期望电平为隐性位； 0：相应位期望电平为显性位。 屏蔽位模式下： 1：必须和对应的标识符寄存器位一致；	0

			0：不需要和对应的标识符寄存器位一致。	

第 25 章 数字视频接口（DVP）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器系列部分产品。

数字视频接口（DVP），它支持使用 DVP 接口时序来获取图像数据流，支持按原始的行、帧格式组织的图像数据，如 YUV、RGB 等，也支持如 JPEG 格式的压缩图像数据，能够接收外部 8 位、10 位、12 位的摄像头模块输出的高速并行数据流。

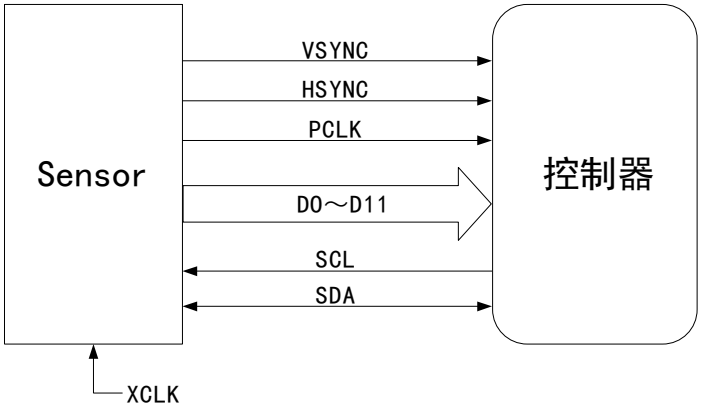
25.1 主要特性

- 可配置8/10/12位数据宽度模式
- 支持YUV、RGB格式数据
- 支持JPEG压缩格式数据
- 内置FIFO，支持DMA传输
- 支持双缓冲接收
- 支持裁剪功能
- 支持连续模式和快照模式

25.2 功能描述

25.2.1 与传感器相连

图 25-1 DVP 接口连接



- PLCK（Pixel clk）：像素时钟，每个时钟对应一个像素数据（非压缩数据）。外部 DVP 接口传感器输出的 PCLK 时钟最大支持 96MHz。
- HSYNC（horizonal synchronization）：行同步信号。
- VSYNC（vertical synchronization）：帧同步信号。
- DATA：像素数据或压缩数据，位宽支持 8/10/12 位。
- XCLK：Sensor 的参考时钟，可以由微控制器提供或外部提供，一般使用晶振。
- I2C 接口：用来配置 sensor 的寄存器，可通过控制的普通 GPIO 口模拟 I2C 时序通讯，亦可使用硬件 I2C 接口操作。

表 25-1 DVP 引脚

名称	信号类型描述
PLCK	像素时钟输入

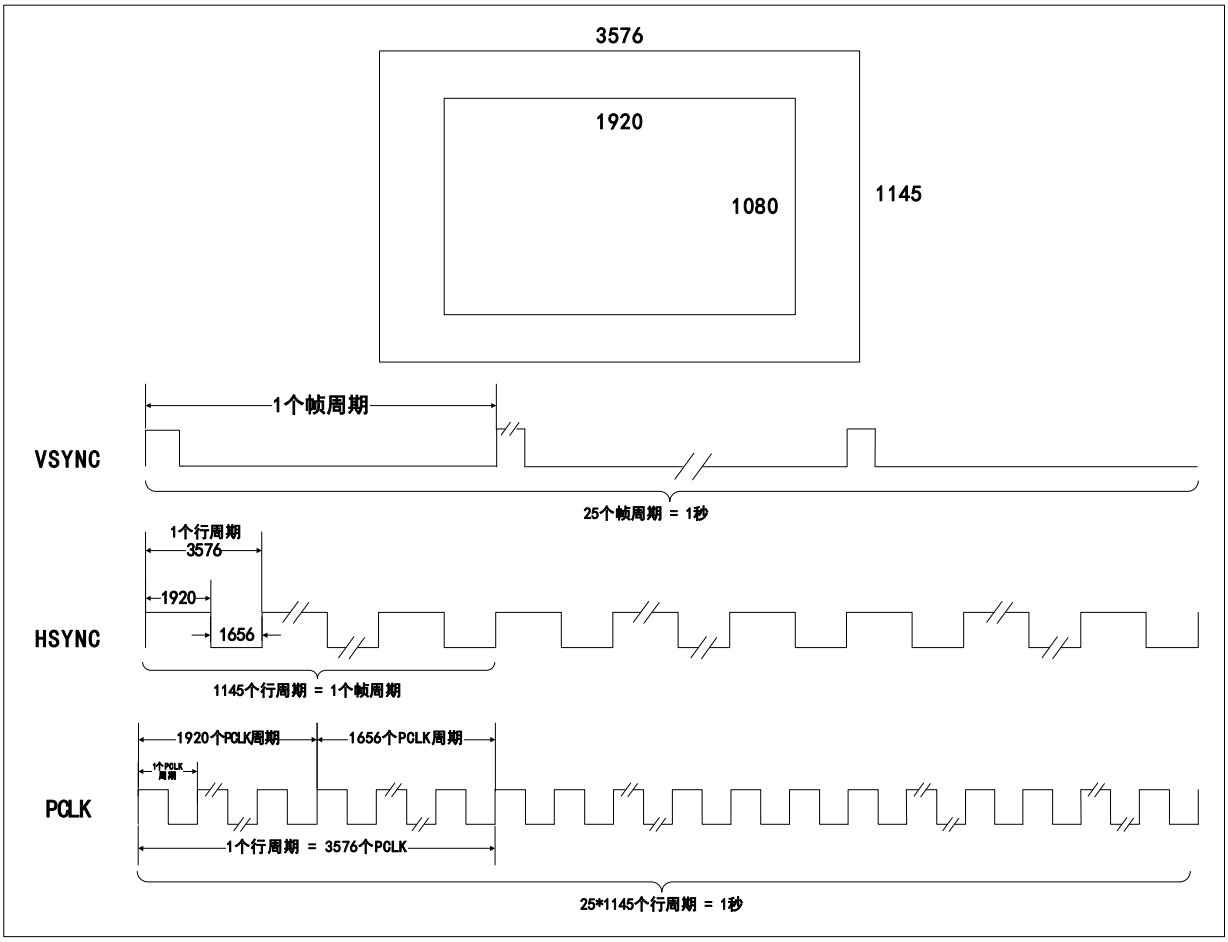
DATA[11:0]	像素数据输入
VSYNC	帧同步信号输入
HSYNC	行同步信号输入

25.2.2 工作时序

一般使用的 DVP 接口传感器模块输出的数字信号数据流和图像大小存在一定的关系，下面以一个图像数据举例。

如图 25-2 所示，传感器内部是一个 3576*1145 大小完整的图像尺寸，经过内部缩放处理，最终从接口输出的图像数据大小为 1920*1080，图像刷新率为 25。

图 25-2 时序描述



PCLK 是一个像素传输的时间，所以 HSYNC 是 PCLK 的 3576 倍，在 3576 个像素中，只有 1920 个像素是有效的，剩下的 1656 个像素点时间内 Sensor 不传输数据。VSYNC 是帧同步信号，所以 VSYNC 时间是 PCLK 的 3576*1145 倍，同样，只有在 1920*1080 个有效像素时间内，Sensor 在传输数据。如果传感器传输的是 JPEG 压缩后的数据，可能无需使用 HSYNC 信号。

DVP 接口信号和图像数据的关系具体以选用的传感器数据手册说明为主。

25.2.3 RGB/YUV/JPEG 压缩数据格式说明

- RGB

  三原色：红色、绿色、蓝色

- YUV

亮度信号 Y，色度信号 U 和 V、Pb 和 Pr、Cb 和 Cr。

- JPEG（Joint Photographic Experts Group）

有损失压缩，但损失的部分是人视觉不易察觉的部分，利用人眼对计算机色彩中高频信息部分不敏感的特点。去除视觉上多余信息（空间冗余度），去除数据本身的多余信息（结构冗余度）。

25.3 数字视频接口的应用

25.3.1 数字图像接口配置说明

1） DVP 的数据接收中，每一帧数据由 BUF0 和 BUF1 交替存储，从 BUF0 开始。对于 RGB 和 YUV 数据流，硬件在每次帧信号从无效电平变为有效电平时会复位选择 BUF0 开始，存满一行数据将会切换的 BUF1，实现交替存储；对于 JPEG 压缩数据，硬件会根据设置的 DMA 接收长度设置 BUF0 和 BUF1 的切换阈值。

2） 当数据总线宽度为 10 位或者 12 位时，每接收一个数据时，系统将自动对数据进行无符号扩展到 16 位再进行存储。

3） R16_DVP_ROW_NUM 和 R16_DVP_COL_NUM 寄存器必须与传感器实际输出的图像大小匹配。

4） 在视频流 RGB 模式下，R16_DVP_COL_NUM 表示一行数据的有效 PCLK 周期数，R16_DVP_ROW_NUM 表示一帧图像数据内包含的行数；在图像 JPEG 模式下，R16_DVP_COL_NUM 用于配置 DMA 长度，此模式下，R16_DVP_ROW_NUM 寄存器不起作用。

25.3.2 数字图像接口应用说明

在使用数字图像接口接收图像数据时，须正确配置 DVP 相关的控制寄存器，使其与图像传感器的模式相匹配，具体操作步骤如下：

1） 通过 R8_DVP_CR1 寄存器清除 RB_DVP_ALL_CLR 和 RB_DVP_RCV_CLR 字段。

2） 通过 R8_DVP_CR0 寄存器配置图像模式、数据位宽、PCLK 极性、HSYNC 极性和 VSYNC 极性，使其与 SENSOR 的输出相匹配。

3） 根据配置后的图像传感器输出的有效图像像素，配置 R16_DVP_ROW_NUM 和 R16_DVP_COL_NUM 寄存器，使其与 SENSOR 的输出相匹配，在图像 JPEG 模式下，只需配置 R16_DVP_COL_NUM 寄存器。

4） 通过 R32_DVP_DMA_BUF0/1 寄存器配置 DMA 接收地址。

5） 若使用快照模式，需通过 R8_DVP_CR1 寄存器，配置 RB_DVP_CM 字段，开启快照模式。

6） 若使用裁剪模式，需通过 R8_DVP_CR1 寄存器，配置 RB_DVP_CROP 和 RB_DVP_FCRC 字段，开启裁剪功能和控制帧捕获率，同时通过配置 R16_DVP_HOFFCNT、R16_DVP_VST、R16_DVP_CAPCNT 和 R16_DVP_VLINE 设置裁剪图像的尺寸。

7） 根据需求，通过 R8_DVP_IER 寄存器使能相应中断，通过中断控制器 NVIC 或 PFIC 配置中断优先级以及使能 DVP 中断。

8） 通过 R8_DVP_CR1 寄存器使能 DMA，通过 R8_DVP_CR0 寄存器使能 DVP 接口。

9） 等待相关接收中断的产生，及时处理接收数据。

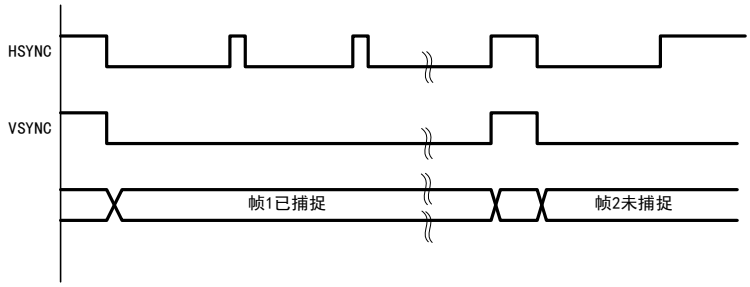
25.3.3 捕获模式说明

DVP 接口支持两种捕获模式：快照（单帧）模式和连续模式。

25.3.3.1 快照模式

在该模式下，只捕获单帧（R8_DVP_CR1 寄存器中 RB_DVP_CM 置 1），当使能 DVP 接口后，等待系统检测帧起始，之后开始进行图像数据采样，在接收完整一帧数据后，将会关闭 DVP 接口（R8_DVP_CR0 寄存器的 RB_DVP_ENABLE 字段清 0）。快照模式下的单帧捕获波形如图 25-3 所示。

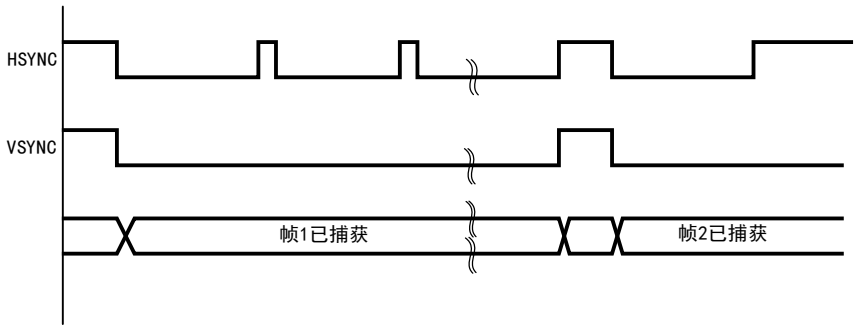
图 25-3 快照模式下的单帧捕获波形



25.3.3.2 连续模式

在该模式下（R8_DVP_CR1 寄存器中 RB_DVP_CM 清 0），当使能 DVP 接口后，在 R8_DVP_CR0 寄存器的 RB_DVP_ENABLE 字段清 0 前，会持续采样每帧数据。连续模式下的帧捕获波形如图 25-4 所示。

图 25-4 连续模式下帧捕获波形



25.3.4 裁剪功能说明

DVP 可以使用裁剪功能从接收到的图像中截取一个矩形窗口，该矩形窗口起始坐标（矩形左上角 X 坐标 R16_DVP_HOFFCNT,Y 坐标 R16_DVP_VST)和窗口大小(R16_DVP_CAPCNT 表示水平尺寸、R16_DVP_VLINE 表示垂直尺寸）可配置。裁剪窗口的坐标和大小如图 25-5 所示。裁剪窗口数据捕获波形如图 25-6 所示。

图 25-5 裁剪窗口的坐标和大小

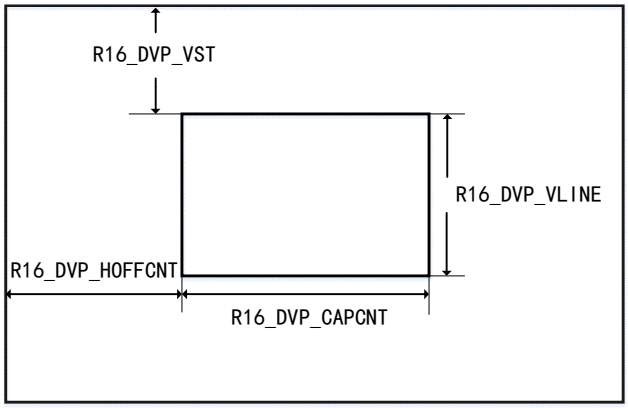
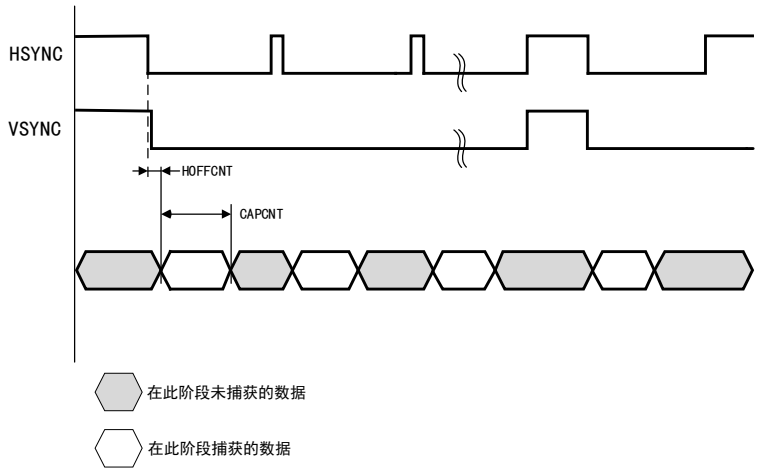


图 25-6 裁剪窗口数据捕获波形



25.4 寄存器描述

表 25-2 DVP 相关寄存器列表

名称	访问地址	描述	复位值
R8_DVP_CR0	0x50050000	DVP 控制寄存器 0	0x00
R8_DVP_CR1	0x50050001	DVP 控制寄存器 1	0x06
R8_DVP_IER	0x50050002	DVP 中断使能寄存器	0x00
R16_DVP_ROW_NUM	0x50050004	图像行数配置寄存器	0x0000
R16_DVP_COL_NUM	0x50050006	图像列数配置寄存器	0x0000
R32_DVP_DMA_BUF0	0x50050008	DVP DMA 地址 0 寄存器	0x00000000
R32_DVP_DMA_BUF1	0x5005000C	DVP DMA 地址 1 寄存器	0x00000000
R8_DVP_IFR	0x50050010	DVP 中断标志寄存器	0x00
R8_DVP_STATUS	0x50050011	DVP 接收 FIFO 状态寄存器	0x00
R16_DVP_ROW_CNT	0x50050014	DVP 行计数器寄存器	0x0000
R16_DVP_HOFFCNT	0x50050018	窗口开始的水平方向位移寄存器	0x0000
R16_DVP_VST	0x5005001A	窗口开始的行数寄存器	0x0000
R16_DVP_CAPCNT	0x5005001C	捕获计数寄存器	0x0000
R16_DVP_VLINE	0x5005001E	垂直行计数寄存器	0x0000
R32_DVP_DR	0x50050020	数据寄存器	0x00000000

25.4.1 DVP 配置寄存器（R8_DVP_CR0）

偏移地址：0x00

位	名称	访问	描述	复位值
6	RB_DVP_JPEG	RW	JPEG 模式使能。 1：JPEG 压缩格式； 0：原始数据格式。	0
[5:4]	RB_DVP_MSK_DAT_MOD	RW	DVP 数据位宽配置。 00：8 位模式； 01：10 位模式； 1x：12 位模式。	0

3	RB_DVP_P_POLAR	RW	PCLK 极性配置。 1：在 PCLK 下降沿采样数据； 0：在 PCLK 上升沿采样数据。	0
2	RB_DVP_H_POLAR	RW	HSYNC 极性配置。 1：HSYNC 低电平数据有效； 0：HSYNC 高电平数据有效。	0
1	RB_DVP_V_POLAR	RW	VSYNC 极性配置。 1：VSYNC 高电平数据有效； 0：VSYNC 低电平数据有效。	0
0	RB_DVP_ENABLE	RW	DVP 功能使能。 1：使能 DVP；　　0：禁用 DVP。	0

25.4.2 DVP 配置寄存器（R8_DVP_CR1）

偏移地址：0x01

位	名称	访问	描述	复位值
[7:6]	RB_DVP_FCRC	RW	DVP 帧捕获率控制。 00：捕获所有帧； 01：每隔一帧捕获一次； 10：每隔三帧捕获一次。 11：保留	0
5	RB_DVP_CROP	RW	裁剪功能控制。 0：捕获完整图像； 1：仅捕获剪裁寄存器所指定的窗口中的数据。	0
4	RB_DVP_CM	RW	捕获模式。 0：连续模式； 1：快照模式。	0
3	RB_DVP_BUF_TOG	RWT	缓冲地址标志位。硬件控制翻转，软件置 1 翻转该位，写 0 无效。 1：数据存储在接收地址 1； 0：数据存储在接收地址 0。	0
2	RB_DVP_RCV_CLR	RW	接收逻辑复位控制。 1：复位接收逻辑电路； 0：取消复位操作。	1
1	RB_DVP_ALL_CLR	RW	标志与 FIFO 清除控制，由软件写 1 或写 0： 1：复位标志与 FIFO； 0：取消复位操作。	1
0	RB_DVP_DMA_ENABLE	RW	DMA 使能控制位： 1：使能 DMA；　　0：禁用 DMA。	0

25.4.3 DVP 中断使能寄存器（R8_DVP_IER）

偏移地址：0x02

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	-
4	RB_DVP_IE_STP_FRM	RW	帧结束中断使能。（在 VSYNC 从有效电平变为无效电平时产生中断。） 1：使能帧结束中断； 0：禁止帧结束中断。	0
3	RB_DVP_IE_FIFO_OV	RW	接收 FIFO 溢出中断使能。 1：使能 FIFO 溢出中断； 0：禁止 FIFO 溢出中断。	0
2	RB_DVP_IE_FRM_DONE	RW	帧接收完成中断使能。（计数器达到 RAW/COL_NUM 配置值时产生中断，表示最后一个数据已写入 RAM） 1：使能帧接收完成中断； 0：禁止帧接收完成中断。	0
1	RB_DVP_IE_ROW_DONE	RW	行结束中断使能。（计数器达到 COL_NUM 配置值时产生中断） 1：使能行结束中断； 0：禁止行结束中断。	0
0	RB_DVP_IE_STR_FRM	RW	新一帧开始中断使能。（在 VSYNC 从无效电平变为有效电平时产生中断，表示新的一帧开始，数据即将到来。） 1：使能新一帧开始中断； 0：禁止新一帧开始中断。	0

25.4.4 DVP 图像有效行数配置寄存器（R16_DVP_ROW_NUM）

偏移地址：0x04

位	名称	访问	描述	复位值
[15:0]	RB_DVP_ROW_NUM	RW	在 RGB 模式下，表示一帧图像数据内包含的行数。在 JPEG 模式下，该寄存器无实际意义。	0

25.4.5 DVP 图像有效列配置寄存器（16_DVP_COL_NUM）

偏移地址：0x06

位	名称	访问	描述	复位值
[15:0]	RB_DVP_COL_NUM	RW	在 RGB 模式下，表示一行数据内包含的 PCLK 周期数。在 JPEG 模式下，用于配置 DMA 接收长度。	0

25.4.6 DVP DMA 接收地址 0 寄存器(R32_DVP_DMA_BUF0)

偏移地址：0x08

位	名称	访问	描述	复位值
[16:0]	RB_DVP_DMA_BUF0	RW	DMA 接收地址 0。	0

25.4.7 DVP DMA 接收地址 1 寄存器(R32_DVP_DMA_BUF1)

偏移地址：0x0C

位	名称	访问	描述	复位值
[16:0]	RB_DVP_DMA_BUF1	RW	DMA 接收地址 1。	0

25.4.8 DVP 中断标志寄存器(R8_DVP_IFR)

偏移地址：0x10

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	-
4	RB_DVP_IF_STP_FRM	RW	帧结束中断标志，高有效，写 0 清除。	0
3	RB_DVP_IF_FIFO_OV	RW	接收 FIFO 溢出中断标志，高有效，写 0 清除。	0
2	RB_DVP_IF_FRM_DONE	RW	帧结束中断标志，高有效，写 0 清除。	0
1	RB_DVP_IF_ROW_DONE	RW	行结束中断标志，高有效，写 0 清除。	0
0	RB_DVP_IF_STR_FRM	RW	帧开始中断标志，高有效，写 0 清除。	0

25.4.9 DVP 接收 FIFO 状态寄存器(R8_DVP_STATUS)

偏移地址：0x11

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
[6:4]	RB_DVP_FIFO_CNT	RO	FIFO 计数器。	0
3	Reserved	RO	保留。	0
2	RB_DVP_FIFO_OV	RO	FIFO 溢出状态。 1：FIFO 溢出；　　0：FIFO 未溢出。	0
1	RB_DVP_FIFO_FULL	RO	FIFO 满状态。 1：缓存已满；　　0：FIFO 未满。	0
0	RB_DVP_FIFO_RDY	RO	FIFO 就绪状态。 1：FIFO 中有数据；　　0：FIFO 中无数据。	0

25.4.10 DVP 接收图像行数寄存器(R16_DVP_ROW_CNT)

偏移地址：0x14

位	名称	访问	描述	复位值
[15:0]	RB_DVP_ROW_CNT	RO	实际接收中，一帧图像数据包含的行数，此寄存器在帧结束时更新。 在 JPEG 格式下，该寄存器的值没有意义。	0

25.4.11 DVP 窗口开始的水平方向位移寄存器(R16_DVP_HOFFCNT)

偏移地址：0x18

位	名称	访问	描述	复位值
[15:0]	RB_DVP_HOFFCNT	RW	窗口行内，每行在捕获数据前需要空出 PCLK 周期数。低 14 位有效。	0

25.4.12 DVP 窗口开始的行数寄存器(R16_DVP_VST)

偏移地址：0x1A

位	名称	访问	描述	复位值
[15:0]	RB_DVP_VST	RW	图像开始捕获的行数，此行之前的数据不捕获。低 13 位有效。	0

25.4.13 DVP 捕获计数寄存器(R16_DVP_CAPCNT)

偏移地址：0x1C

位	名称	访问	描述	复位值
[15:0]	RB_DVP_CAPCNT	RW	裁剪窗口内需要捕获的 PCLK 周期数。低 14 位有效。	0

25.4.14 DVP 垂直行计数寄存器(R16_DVP_VLINE)

偏移地址：0x1E

位	名称	访问	描述	复位值
[15:0]	RB_DVP_VLINE	RW	裁剪窗口内需要捕获的行数。低 14 位有效。	0

25.4.15 DVP 数据寄存器(R32_DVP_DR)

偏移地址：0x20

位	名称	访问	描述	复位值
[32:0]	RB_DVP_DR	RO	DVP 接口每次接收 4 字节数据，才会触发一次 DMA 请求，4 字节深度的 FIFO 可为 DMA 传输留有充足的时间。可有效防止出现 DMA 溢出情况。	0

第 26 章 可变静态存储控制器（FSMC）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器系列部分产品。

可变静态存储控制器（FSMC），支持多种静态存储器类型以及丰富的存储操作方法，可根据系统应用需要，对不同类型大容量静态存储器进行扩展。

26.1 主要特性

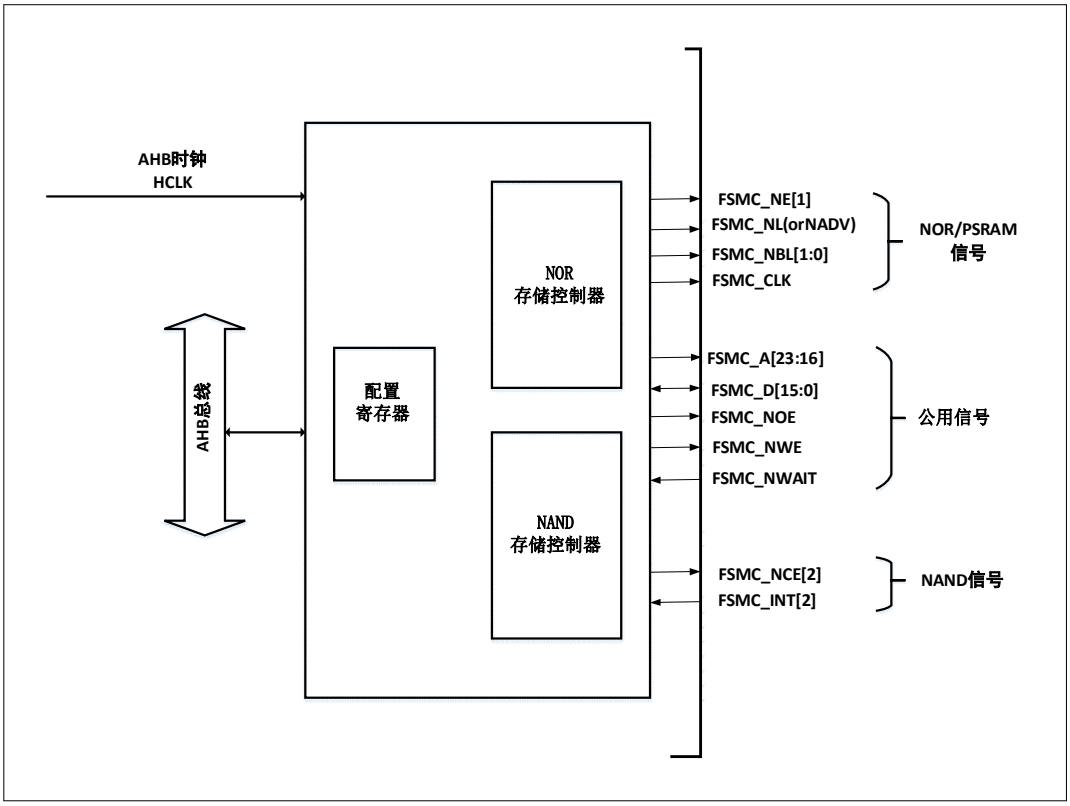
- 支持操作SRAM、ROM、NOR FLASH和PSRAM
- 支持操作NAND FLASH，并且内置硬件ECC，最多可检测8KByte数据
- 支持对同步器件操作，如PSRAM
- 支持8bit或16bit数据总线宽度
- 时序信号可软件编程

26.2 功能描述

26.2.1 模块结构

FSMC 主要包括 AHB 总线接口、NOR 存储控制器、NAND 存储控制器和外部设备接口。

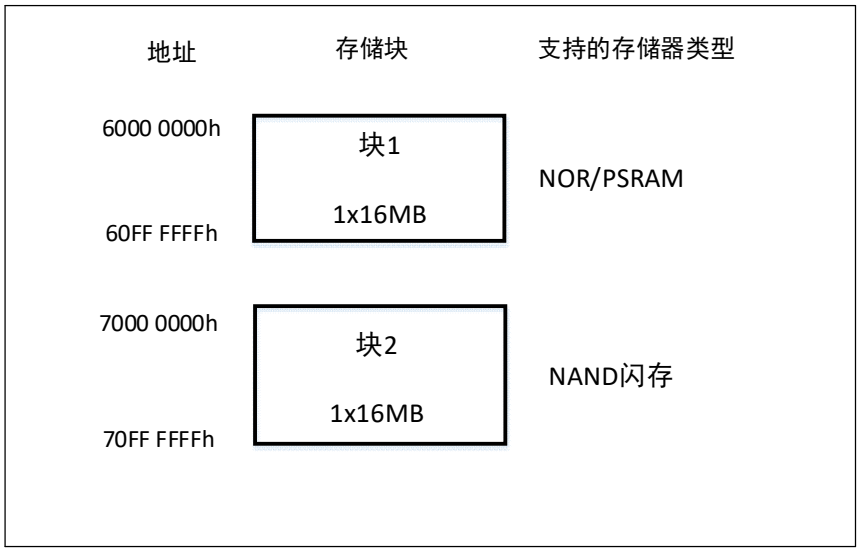
图 26-1 FSMC 模块框图



26.2.2 外部设备地址映射

FSMC 根据地址线将存储块分为固定大小 16MByte 的两个存储块，见图 26-2。

图 26-2 FSMC 存储块



26.2.2.1 NOR/PSRAM 地址映射

表 26-1 外部存储器地址

数据宽度	配连接到存储器的地址线	最大访问存储器空间(bit)
8bit	HADDR[23:0]与 FSMC_A[23:0]对应相连	16MBx8=128Mbit
16bit	HADDR[23:1]与 FSMC_A[22:0]对应相连，HADDR[0]未接	16MB/2x16=128Mbit

注：100 脚芯片，支持 FSMC，并且仅支持地址和数据线复用模式。

NOR FLASH 和 PSRAM 支持非对齐访问。对于异步模式，每次操作需要准确的地址；对于同步模式，只需发出一次地址信号，之后批量的数据通过 CLK 顺序进行。对于支持非对齐批量访问的 NOR FLASH，可以将存储器的非对齐访问模式设置为和 AHB 相同的模式，若不能设置，则禁用非对齐访问模式，并把费对齐的访问请求分开成两个连续的访问操作。

26.2.2.2 NAND 地址映射

NAND FLASH 映像和时序寄存器，见表 26-2。

表 26-2 存储器映像和时序寄存器

起始地址	结束地址	FSMC 存储块	存储空间	时序寄存器
0x78000000	0x78FFFFFF	块 2-NAND FLASH	属性	FSMC_PATT2(0x6C)
0x70000000	0x70FFFFFF		通用	FSMC_PMEM2(0x68)

通用和属性空间可以在低 256KB 划分为地址区（第二个 128KB 区域）、命令区（第二个 64KB 区域）和数据区（前 64KB 区域），见表 26-3。

软件通过操作这 3 个区域访问 NAND FLASH 的具体流程：

- 向 NAND FLASH 发送读写等操作命令：软件可以操作命令区任何地址发送命令；
- 向 NAND FLASH 发送将要操作的地址：软件可以操作地址区任何地址发送命令；

- 从 NAND FLASH 读写数据：软件可以操作数据区任何地址写入或读出数据；

表 26-3 NAND 存储块选择

区域名称	HADDR[17:16]	地址范围
地址区	1X	0x020000-0x03FFFF
命令区	01	0x010000-0x01FFFF
数据区	00	0x000000-0x00FFFF

26.2.3 NOR/PSRAM 控制器

FSMC 支持 8bit、16bit 和 32bit 异步操作 SRAM 和 ROM，支持异步模式和突发模式操作 PSRAM，支持异步模式和突发模式操作 NOR FLASH。所有控制器的输出信号在内部时钟 HCLK 的上升沿改变，对于同步写模式（PSRAM）下，输出的数据在存储器时钟（CLK）的下降沿改变，具体参照同步传输和异步传输时序图。存储器的读写参数可软件配置，见表 26-4。

表 26-4 软件可控的 NOR/PSRAM 读写参数

参数	读写方式	参数取值范围
地址建立时间	异步	1<< T <<16 (AHB HCLK)
地址保持时间	异步	1<< T <<16 (AHB HCLK)
数据建立时间	异步	2<< T <<256 (AHB HCLK)
总线恢复时间	异步或同步读	1<< T <<16 (AHB HCLK)
时钟分频因子	同步	2<< T <<16 (AHB HCLK)
数据产生时间	同步	2<< T <<17 (Memory CLK)

26.2.3.1 外部存储器复用接口信号

NOR FLASH 和 PSRAM 接口，见表 26-5、表 26-6。

表 26-5 复用 NOR FLASH 接口

FSMC 引脚	方向	描述
CLK	输出	时钟线（仅用于同步突发模式）
A[23:16]	输出	地址线
AD[15:0]	输入/输出	16bit 地址/数据线（复用）
NE[1]	输出	片选线
NOE	输出	输出使能
NWE	输出	写使能
NL(NADV)	输出	锁存使能
NWAIT	输入	等待信号线（仅用于 NOR FLASH）
NBL[1]	输出	高字节使能（NUB）
NBL[0]	输出	低字节使能（NLB）

注：前缀“N”的信号，表示低电平有效。

表 26-6 复用 PSRAM 接口

FSMC 引脚	方向	描述
CLK	输出	时钟线（仅用于同步突发模式）
A[23:16]	输出	地址线
AD[15:0]	输入/输出	16bit 地址/数据线（复用）
NE[1]	输出	片选线

NOE	输出	输出使能
NWE	输出	写使能
NL(NADV)	输出	锁存使能
NWAIT	输入	等待信号线（仅用于 NOR FLASH）

26.2.3.2 支持的存储器以及操作方式

支持的存储器以及操作方式见表 26-7。

表 26-7 支持存储器和操作方式

存储器	模式	AHB 数据宽度	存储器宽度	描述
NOR FLASH	异步读	8	16	
	异步读	16	16	
	异步写	16	16	
	异步读	32	16	分成 2 次 FSMC 访问
	异步写	32	16	分成 2 次 FSMC 访问
	同步读	16	16	
	同步读	32	16	分成 2 次 FSMC 访问
PSRAM	异步读	8	16	
	异步写	8	16	使用字节信 NBL[1:0]
	异步读	16	16	
	异步写	16	16	
	异步读	32	16	分成 2 次 FSMC 访问
	异步写	32	16	分成 2 次 FSMC 访问
	同步读	16	16	
	同步读	32	16	分成 2 次 FSMC 访问
	同步写	8	16	使用字节信 NBL[1:0]
	同步写	16/32	16	
SRAM 和 ROM	异步读	8/16/32	8/16	使用字节信 NBL[1:0]
	异步写	8/16/32	8/16	使用字节信 NBL[1:0]

26.2.3.3 NOR/PSRAM 异步传输地址/数据复用的时序图

对于异步静态存储器（NOR FLASH 和 PSRAM）操作需注意如下：

- 当启用扩展模式时，可混合使用模式 A、B、C 和 D 对存储器的读写操作。

图 26-3 模式 1 读操作（复用）

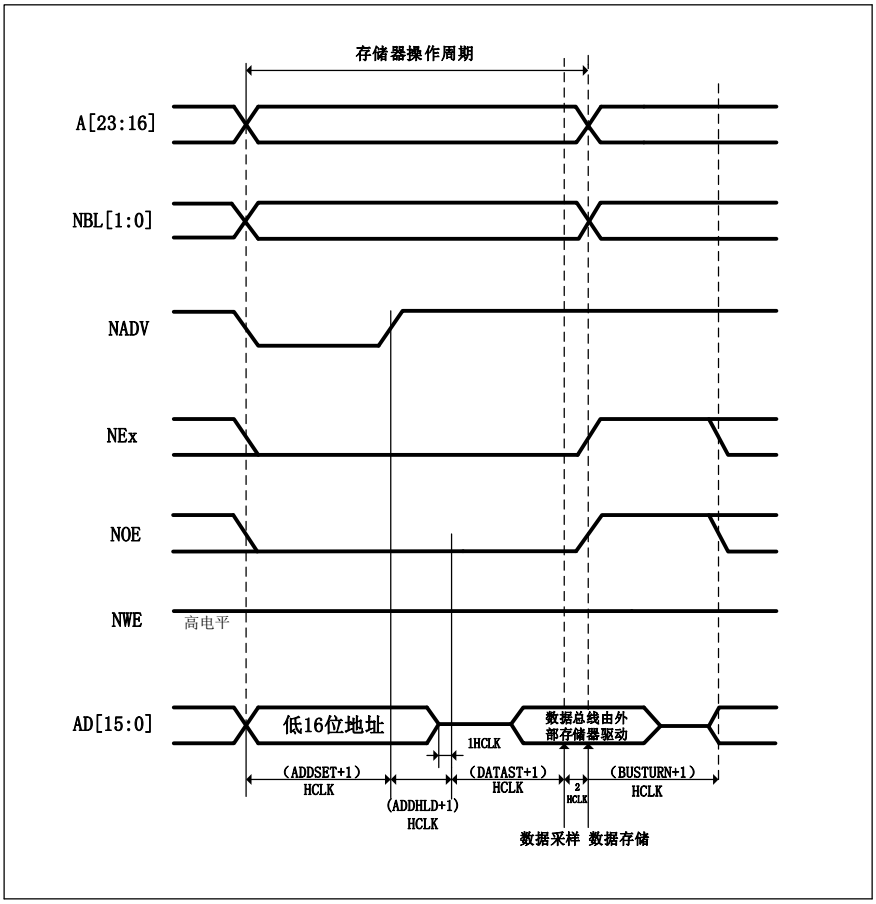


图 26-4 模式 1 写操作（复用）

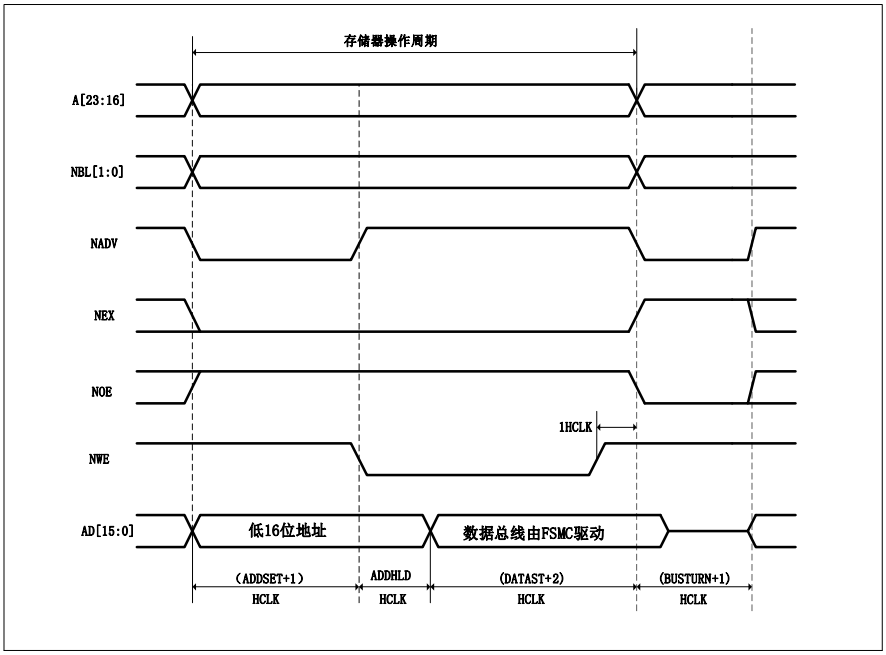


表 26-8 模式 1 FSMC_BCR1 位域

Bitx	名称	配置值
[31:20]	Reserved	0x000
19	CBURSTRW	0x0
[18:16]	Reserved	0x0
15	ASYNCWAIT	如果存储器支持该功能则为 1，否则为 0
14	EXTMOD	0x0
13	WAITEN	0x0
12	WREN	根据需要设置
11	WAITCFG	该位不起作用
10	WRAPMOD	0x0
9	WAITPOL	当 bit15 为 1 时，有意义
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	该位不起作用
[5:4]	MWID	根据需要设置
[3:2]	MTYP	根据需要设置，不包含 0x2(NOR FLASH)
1	MUXEN	0x1
0	MBKEN	0x1

表 26-9 模式 1 FSMC_BTR1 位域

Bitx	名称	配置值
[31:30]	Reserved	0x0
[29:28]	ACCMOD	该位不起作用
[27:24]	DATLAT	该位不起作用
[23:20]	CLKDIV	该位不起作用
[19:16]	BUSTURN	0x0
[15:8]	DATAST	数据建立时间。写操作为（DATAST+1 HCLK），读操作为（DATAST+3 HCLK）。该位最小为 1。
[7:4]	ADDHLD	地址保持时间。（ADDHLD+1 HCLK）
[3:0]	ADDSET	地址建立时间。（ADDSET+1 HCLK）。

图 26-5 模式 A 读操作（复用）

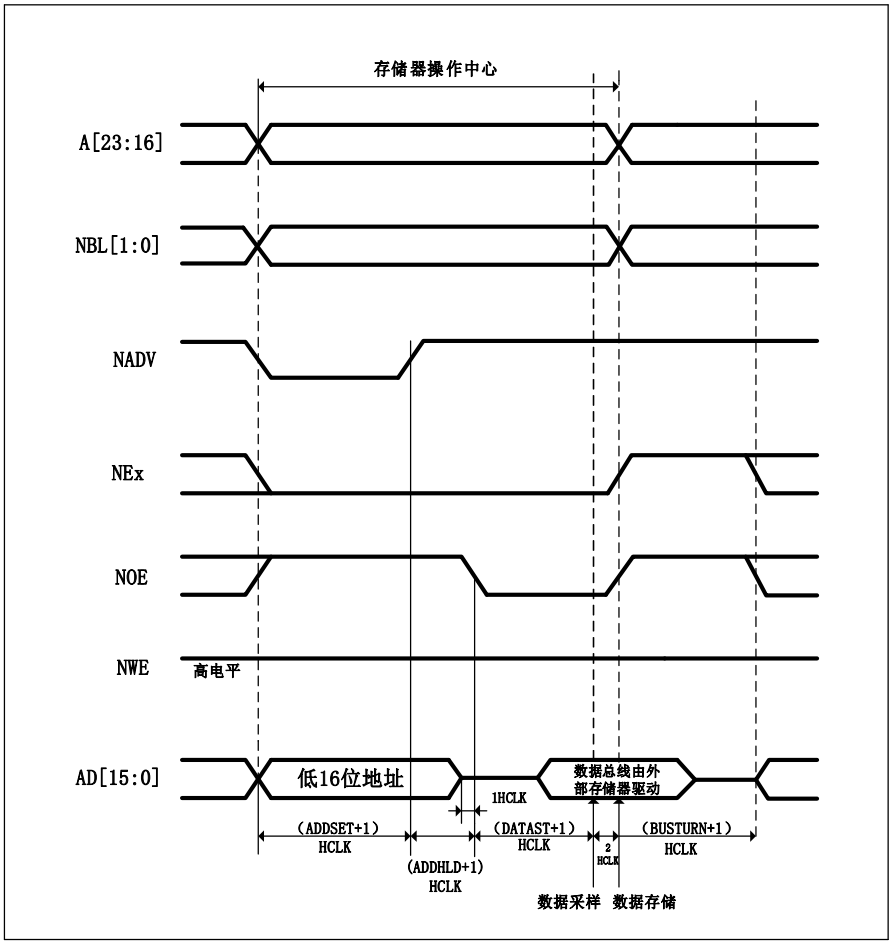


图 26-6 模式 A 写操作（复用）

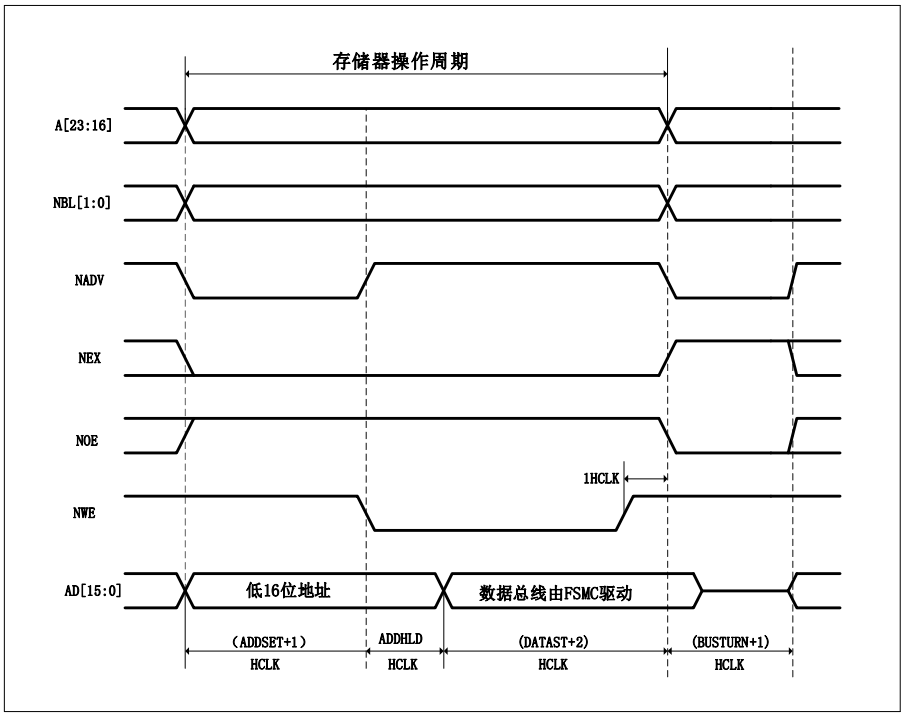


表 26-10 模式 A FSMC_BCR1 位域

Bitx	名称	配置值
[31:20]	Reserved	0x000
19	CBURSTRW	0x0
[18:16]	Reserved	0x0
15	ASYNCWAIT	如果存储器支持该功能则为 1，否则为 0
14	EXTMOD	0x1
13	WAITEN	0x0
12	WREN	根据需要设置
11	WAITCFG	该位不起作用
10	WRAPMOD	0x0
9	WAITPOL	当 bit15 为 1 时，有意义
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	该位不起作用
[5:4]	MWID	根据需要设置
[3:2]	MTYP	根据需要设置，不包含 0x2(NOR FLASH)
1	MUXEN	0x1
0	MBKEN	0x1

表 26-11 模式 A FSMC_BTR1 位域

Bitx	名称	配置值
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	该位不起作用
[23:20]	CLKDIV	该位不起作用
[19:16]	BUSTYRN	0x0
[15:8]	DATAST	数据建立时间。写操作为（DATAST+1 HCLK），读操作为（DATAST+3 HCLK）。该位最小为 1。
[7:4]	ADDHLD	该位不起作用
[3:0]	ADDSET	地址建立时间。读操作为（ADDSET+1 HCLK）。

表 26-12 模式 A FSMC_BWTR1 位域

Bitx	名称	配置值
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	该位不起作用
[23:20]	CLKDIV	该位不起作用
[19:16]	BUSTURN	0x0
[15:8]	DATAST	数据建立时间。写操作为（DATAST+1 HCLK），读操作为（DATAST+3 HCLK）。该位最小为 1。
[7:4]	ADDHLD	地址保持时间。（ADDHLD+1 HCLK）

[3:0]	ADDSET	地址建立时间。写操作为（ADDSET+1 HCLK）。

图 26-7 模式 B 读操作（复用）

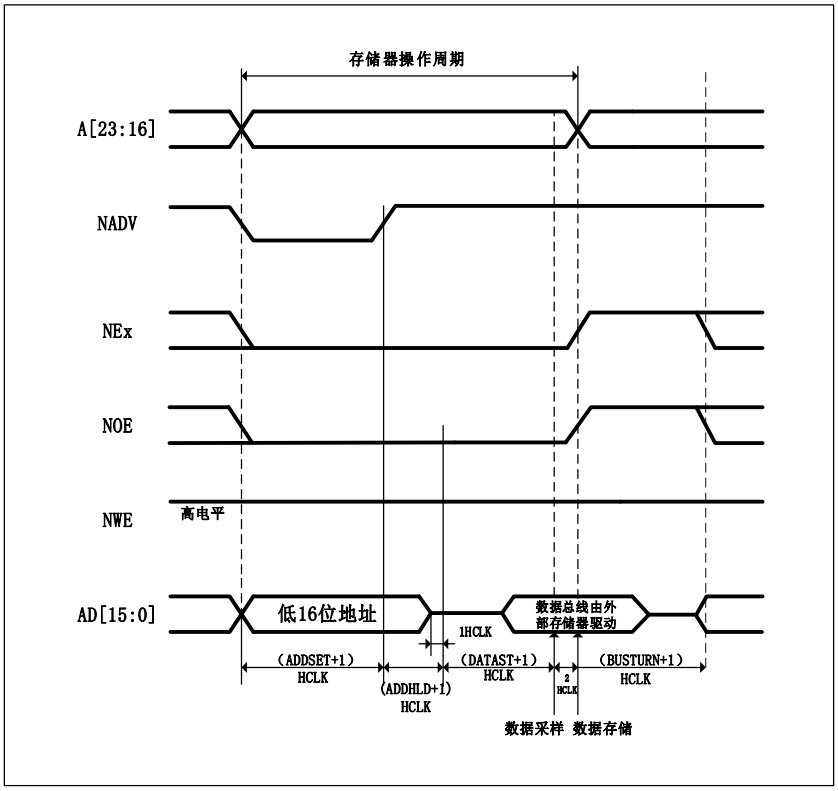


图 26-8 模式 B 写操作（复用）

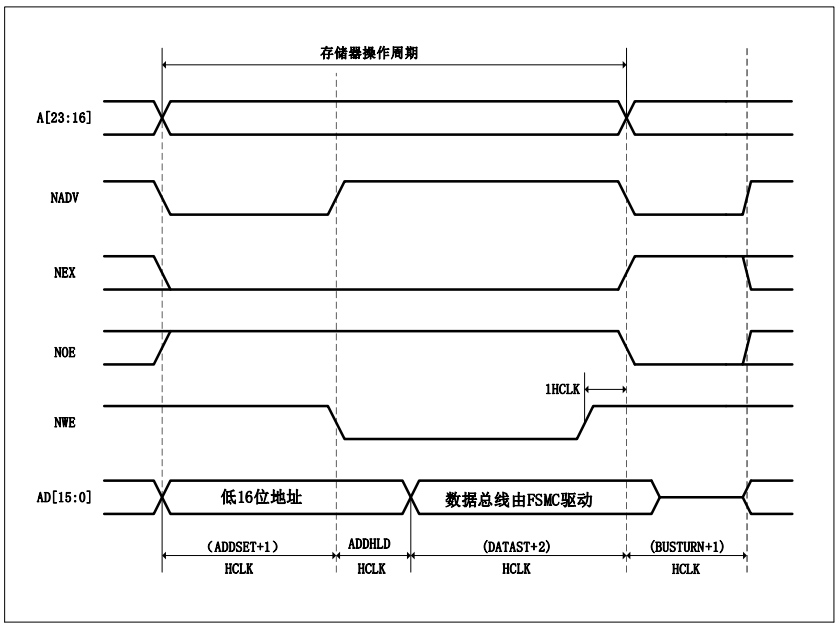


表 26-13 模式 B FSMC_BCR1 位域

Bitx	名称	配置值
[31:20]	Reserved	0x000
19	CBURSTRW	0x0
[18:16]	Reserved	0x0

15	ASYNCWAIT	如果存储器支持该功能则位 1，否则为 0
14	EXTMOD	0x1
13	WAITEN	0x0
12	WREN	根据需要设置
11	WAITCFG	该位不起作用
10	WRAPMOD	0x0
9	WAITPOL	当 bit15 为 1 时，有意义。
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	0x1
[5:4]	MWID	根据需要设置
[3:2]	MTYP	0x2(NOR FLASH)
1	MUXEN	0x1
0	MBKEN	0x1

表 26-14 模式 B FSMC_BTR1 位域

Bitx	名称	配置值
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x1
[27:24]	DATLAT	该位不起作用
[23:20]	CLKDIV	该位不起作用
[19:16]	BUSTURN	0x0
[15:8]	DATAST	数据建立时间。写操作为（DATAST+1 HCLK），读操作为（DATAST+3 HCLK）。该位最小为 1。
[7:4]	ADDHLD	地址保持时间。（ADDHLD+1 HCLK）
[3:0]	ADDSET	地址建立时间。读操作为（ADDSET+1 HCLK）。

表 26-15 模式 B FSMC_BWTR1 位域

Bitx	名称	配置值
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x1
[27:24]	DATLAT	该位不起作用
[23:20]	CLKDIV	该位不起作用
[19:16]	BUSTURN	0x0
[15:8]	DATAST	数据建立时间。写操作为（DATAST+1 HCLK），读操作为（DATAST+3 HCLK）。该位最小为 1。
[7:4]	ADDHLD	地址保持时间。（ADDHLD+1 HCLK）
[3:0]	ADDSET	地址建立时间。写操作为（ADDSET+1 HCLK）。

图 26-9 模式 C 读操作（复用）

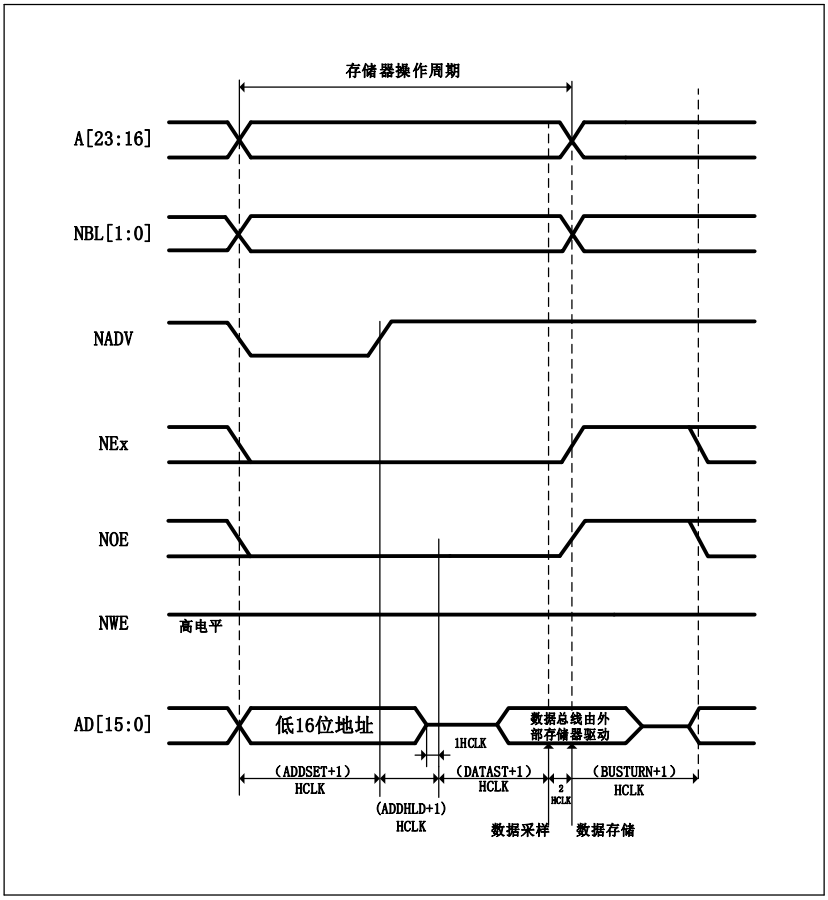


图 26-10 模式 C 写操作（复用）

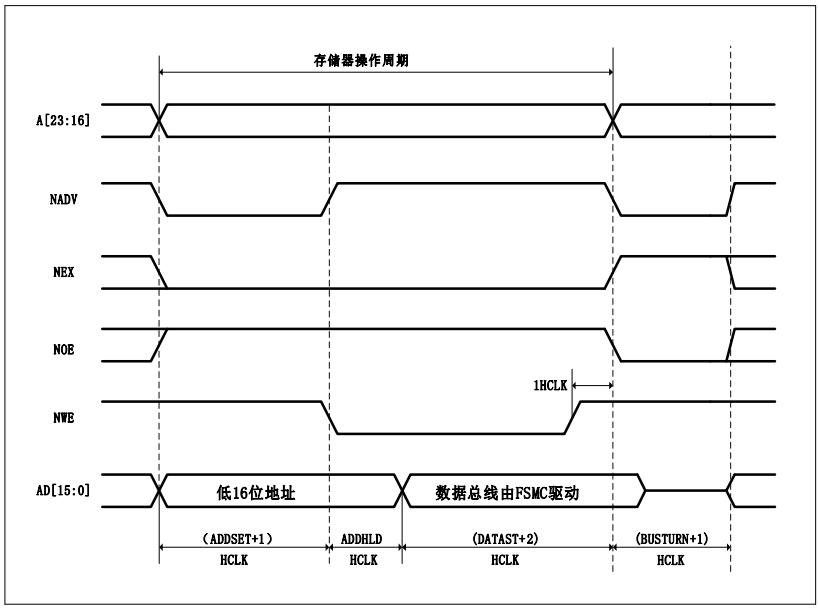


表 26-16 模式 C FSMC_BCR1 位域

Bitx	名称	配置值
[31:20]	Reserved	0x000
19	CBURSTRW	0x0
[18:16]	Reserved	0x0
15	ASYNCWAIT	如果存储器支持该功能则位 1，否则为 0
14	EXTMOD	0x1

13	WAITEN	0x0
12	WREN	根据需要设置
11	WAITCFG	该位不起作用
10	WRAPMOD	0x0
9	WAITPOL	当 bit15 为 1 时，有意义。
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	0x1
[5:4]	MWID	根据需要设置
[3:2]	MTYP	0x2(NOR FLASH)
1	MUXEN	0x1
0	MBKEN	0x1

表 26-17 模式 C FSMC_BTR1 位域

Bitx	名称	配置值
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x2
[27:24]	DATLAT	该位不起作用
[23:20]	CLKDIV	该位不起作用
[19:16]	BUSTURN	0x0
[15:8]	DATAST	数据建立时间。写操作为（DATAST+1 HCLK），读操作为（DATAST+3 HCLK）。该位最小为 1。
[7:4]	ADDHLD	地址保持时间。（ADDHLD+1 HCLK）
[3:0]	ADDSET	地址建立时间。读操作为（ADDSET+1 HCLK）。

表 26-18 模式 C FSMC_BWTR1 位域

Bitx	名称	配置值
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x2
[27:24]	DATLAT	该位不起作用
[23:20]	CLKDIV	该位不起作用
[19:16]	BUSTURN	0x0
[15:8]	DATAST	数据建立时间。写操作为（DATAST+1 HCLK），读操作为（DATAST+3 HCLK）。该位最小为 1。
[7:4]	ADDHLD	地址保持时间。（ADDHLD+1 HCLK）
[3:0]	ADDSET	地址建立时间。写操作为（ADDSET+1 HCLK）。

图 26-11 模式 D 读操作（复用）

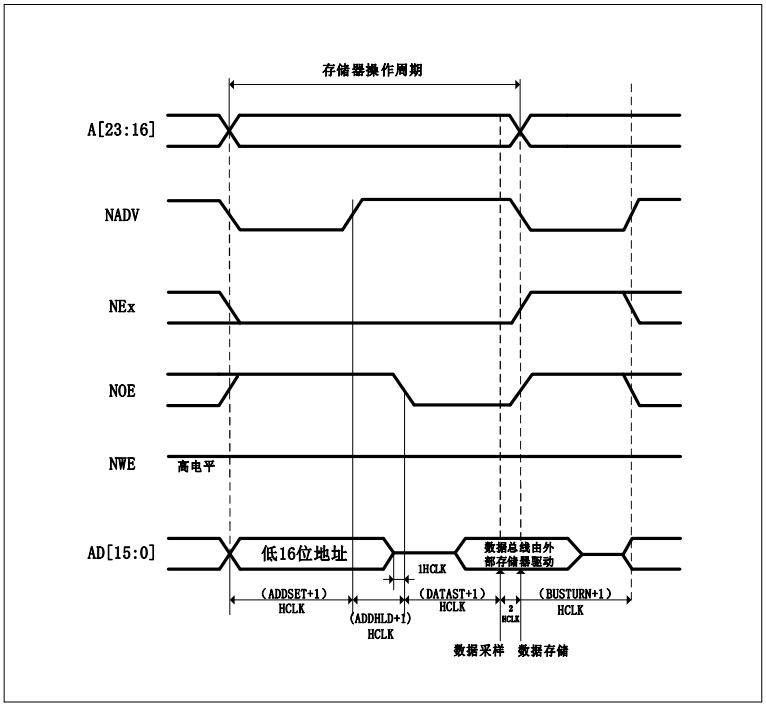


图 26-12 模式 D 写操作（复用）

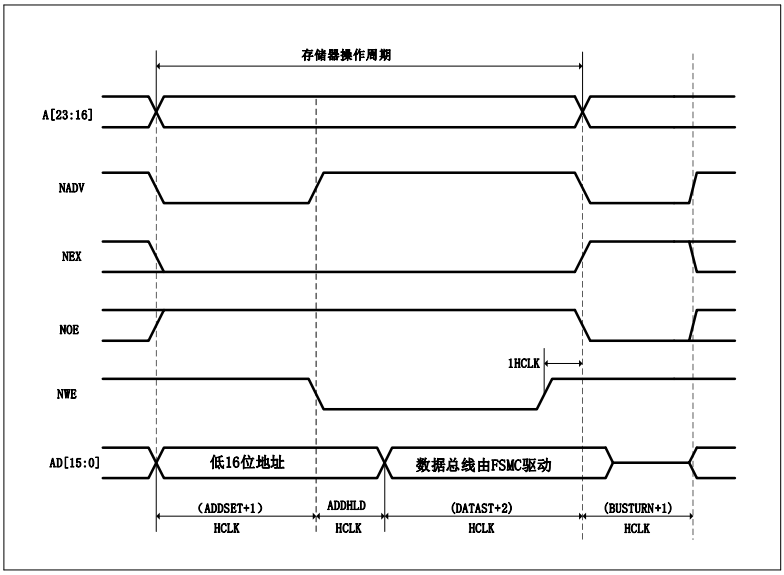


表 26-19 模式 D FSMC_BCR1 位域

Bitx	名称	配置值
[31:20]	Reserved	0x000
19	CBURSTRW	0x0
[18:16]	Reserved	0x0
15	ASYNCWAIT	如果存储器支持该功能则位 1，否则为 0
14	EXTMOD	0x1
13	WAITEN	0x0
12	WREN	根据需要设置
11	WAITCFG	该位不起作用

10	WRAPMOD	0x0
9	WAITPOL	当 bit15 为 1 时，有意义。
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	根据需要设置
[5:4]	MWID	根据需要设置
[3:2]	MTYP	根据需要设置
1	MUXEN	0x1
0	MBKEN	0x1

表 26-20 模式 D FSMC_BTR1 位域

Bitx	名称	配置值
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x3
[27:24]	DATLAT	该位不起作用
[23:20]	CLKDIV	该位不起作用
[19:16]	BUSTURN	0x0
[15:8]	DATAST	数据建立时间。写操作为（DATAST+1 HCLK），读操作为（DATAST+3 HCLK）。该位最小为 1。
[7:4]	ADDHLD	地址保持时间。(ADDHLD+1 HCLK)。
[3:0]	ADDSET	地址建立时间。读操作为（ADDSET+1 HCLK）。

表 26-21 模式 D FSMC_BWTR1 位域

Bitx	名称	配置值
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x3
[27:24]	DATLAT	0x0
[23:20]	CLKDIV	0x0
[19:16]	BUSTYRN	0x0
[15:8]	DATAST	数据建立时间。写操作为（DATAST+1 HCLK），读操作为（DATAST+3 HCLK）。该位最小为 1。
[7:4]	ADDHLD	地址保持时间。(ADDHLD+1 HCLK)。
[3:0]	ADDSET	地址建立时间。写操作为（ADDSET+1 HCLK）。

26.2.3.4 NOR/PSRAM 同步传输地址/数据复用的时序图

数据延迟与 NOR FLASH 的延时需注意，DATALAT 数值必须与 NOR FLASH 配置寄存器中的定义一致。NADV 信号为低时的时钟周期不计算在延迟参数中。FSMC 的 DATLAT 参数可以为 DATALAT+2 或为 DATALAT+3，对于特别的存储器会在数据保持时间阶段产生 NWAIT 信号，DATLAT 需要设置为最小值，对于另外一些存储器不会在数据保持时间阶段产生 NWAIT 信号,FSMC 和存储器的数据保持时间必须设置一致。

对于单次批量传输需注意，存储器配置位同步批量模式，若传输 16bit 数据，FSMC 会执行 1 次长度

为 1 的批量传输，若传输 32bit 数据，FSMC 分为 2 次 16bit 传输，执行 1 次长度为 2 的批量传输。

NOR FLASH 同步批量模式访问时，在保持时间（DATLAT+1 CLK）之后，若检测 NWAIT 信号为低电平时，在 NWAIT 变成高电平之前 FSMC 需要插入等待周期，当 NWAIT 变为高电平时，FSMC 认为数据有效。在 NWAIT 信号控制的等待状态插入期间，控制器会持续向存储器发送时钟脉冲、保持片选信号和输出有效信号，同时忽略无效的数据信号。在批量传输模式下，NOR FLASH 的 NWAIT 信号通过配置 WAITCFG 位选择两种时序配置。

图 26-13 同步总线复用读操作（复用）

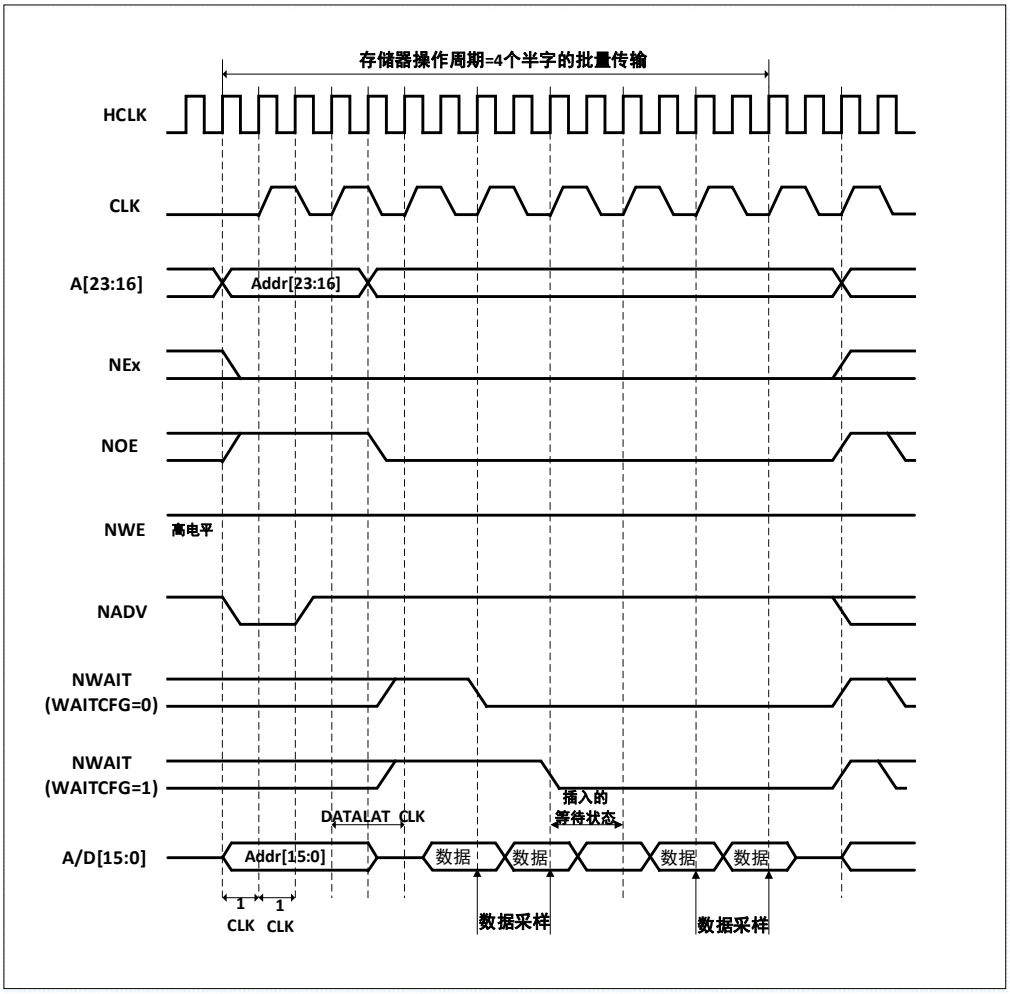


表 26-22 模式 D FSMC_BCR1 位域

Bitx	名称	配置值
[31:20]	Reserved	0x000
19	CBURSTRW	该位不起作用
[18:16]	Reserved	0x0
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	0x0
12	WREN	该位不起作用
11	WAITCFG	根据需要设置
10	WRAPMOD	0x0

9	WAITPOL	根据需要设置
8	BURSTEN	0x1
7	Reserved	0x1
6	FACCEN	根据需要设置
[5:4]	MWID	根据需要设置
[3:2]	MTYP	0x1 或 0x2
1	MUXEN	0x1
0	MBKEN	0x1

表 26-23 模式 D FSMC_BTR1 位域

Bitx	名称	配置值
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	数据保持时间
[23:20]	CLKDIV	0x0 – CLK=1 HCLK 0x1 – CLK=2 HCLK
[19:16]	BUSTURN	该位不起作用
[15:8]	DATAST	该位不起作用
[7:4]	ADDHLD	该位不起作用
[3:0]	ADDSET	该位不起作用

图 26-14 同步总线复用写操作（复用）

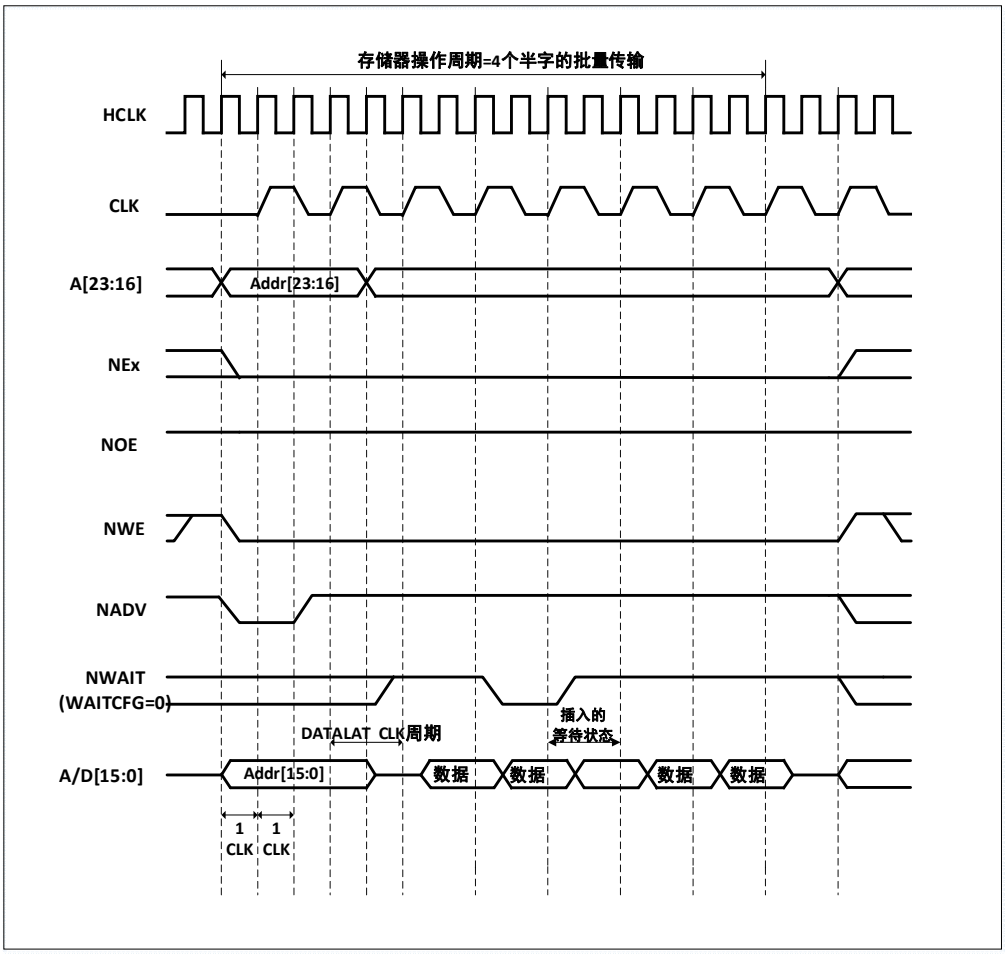


表 26-24 模式 D FSMC_BCR1 位域

Bitx	名称	配置值
[31:20]	Reserved	0x000
19	CBURSTRW	0x1
[18:16]	Reserved	0x0
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	如果存储器支持该功能则位 1，否则为 0
12	WREN	0x1
11	WAITCFG	0x0
10	WRAPMOD	0x0
9	WAITPOL	根据需要设置
8	BURSTEN	该位不起作用
7	Reserved	0x1
6	FACCEN	根据需要设置
[5:4]	MWID	根据需要设置
[3:2]	MTYP	0x1
1	MUXEN	0x1
0	MBKEN	0x1

表 26-25 模式 D FSMC_BTR1 位域

Bitx	名称	配置值
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	数据保持时间
[23:20]	CLKDIV	0x0 - CLK=1 HCLK 0x1 - CLK=2 HCLK
[19:16]	BUSTURN	该位不起作用
[15:8]	DATAST	该位不起作用
[7:4]	ADDHLD	该位不起作用
[3:0]	ADDSET	该位不起作用

26.2.4 NAND FLASH 控制器

FSMC 支持 8bit、16bit 操作 NAND FLASH，FSMC 可以根据需要产生多个地址周期，理论上 FSMC 对 NAND FLASH 的容量没有限制。NAND FLASH 的读写参数可软件配置，见表 26-26。

表 26-26 软件可控的 NAND 读写参数

参数	功能	读写方式	参数取值范围
存储器建立时间	发出命令之前建立地址的时间	读/写	1<< T <<256 (AHB HCLK)
存储器等待时间	发出命令的最短持续时间	读/写	1<< T <<256 (AHB HCLK)
存储器保持时间	再发送命令结束后保持地址的时间，写操作时也是数据的保持时间	读/写	1<< T <<255 (AHB HCLK)
存储器数据总线	启动写操作之后数据总线	写	0<< T <<255 (AHB HCLK)

高阻时间	持续为高阻态时间		

26.2.4.1 外部存储器接口信号

8bit 和 16bit NAND FLASH 接口，见表 26-27、表 26-28。

表 26-27 8bit NAND FLASH

FSMC 引脚	方向	描述
A[17]	输出	地址锁存允许信号（ALE）
A[16]	输出	命令锁存允许信号（CLE）
D[7:0]	输入/输出	8bit 双向地址/数据复用总线
NCE[2]	输出	片选线
NOE	输出	输出使能
NWE	输出	写使能
NWAIT	输入	等待信号线

表 26-28 16bit NAND FLASH

FSMC 引脚	方向	描述
A[17]	输出	地址锁存允许信号（ALE）
A[16]	输出	命令锁存允许信号（CLE）
D[15:0]	输入/输出	16bit 双向地址/数据复用总线
NCE[2]	输出	片选线
NOE	输出	输出使能
NWE	输出	写使能
NWAIT	输入	等待信号线

26.2.4.2 支持的存储器以及操作方式

支持的存储器以及操作方式见表 26-29。

表 26-29 支持存储器和操作方式

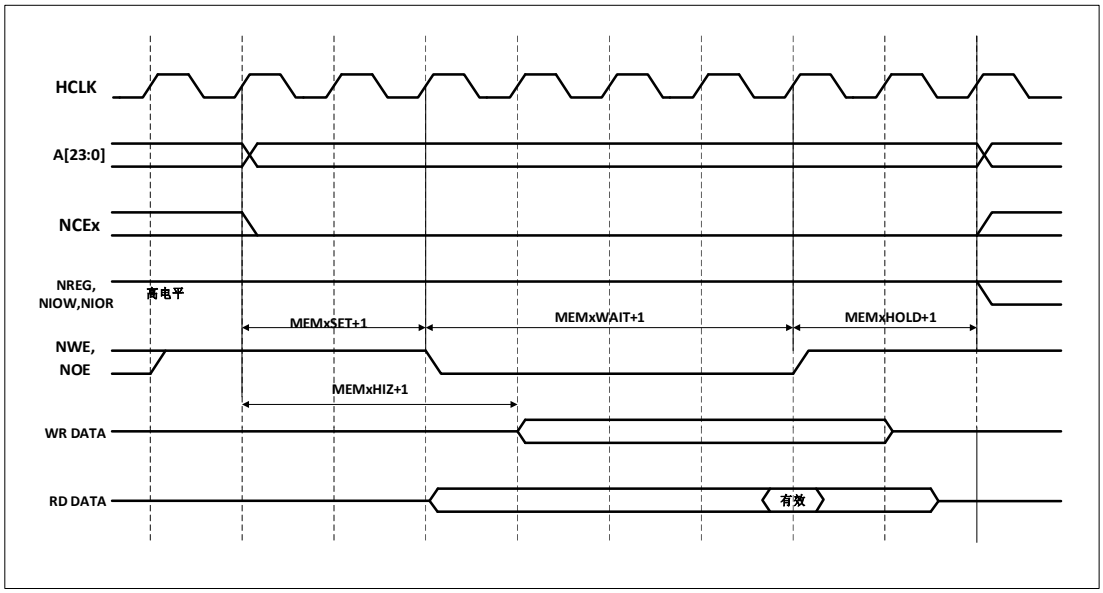
存储器	模式	AHB 数据宽度	存储器宽度	描述
8bit NAND	异步读	8	8	
	异步写	8	8	
	异步读	16	8	分成 2 次 FSMC 访问
	异步写	16	8	分成 2 次 FSMC 访问
	异步读	32	8	分成 4 次 FSMC 访问
	异步写	32	8	分成 4 次 FSMC 访问
16bit NAND	异步读	8	16	
	异步读	16	16	
	异步写	16	16	
	异步读	32	16	分成 2 次 FSMC 访问
	异步写	32	16	分成 2 次 FSMC 访问

26.2.4.3 NAND FLASH 时序图 （包括预等待功能）

NAND FLASH 操作时序控制，涉及 FSMC_PMEM2 和 FSMC_PATT2 时序寄存器，每一个寄存器包含 4 个参数。MEMHOLD、MEMWAIT 和 ATTSET 三个参数对应操作 NAND FLASH 的三个阶段的 HCLK 周期数，MEMHIZ 参

数对应 FSMC 开始驱动数据总线的时机。NAND FLASH 控制器通用存储空间访问时序，见图 26-15。

图 26-15 NAND FLASH 控制器通用存储空间的访问时序



26.2.4.4 NAND FLASH 操作流程

NAND FLASH 的命令锁存使能信号（CLE）和地址锁存使能信号（ALE）分别由 FSMC 地址线 A16 和 A17 驱动，故在对 NAND FLASH 操作发送命令或地址时，需要对存储空间的特定地址执行写操作。对 NAND FLASH 的读操作过程如下：

1） 根据即将操作的 NAND FLASH 特性，配置 FSMC_PCR2、FSMC_PMEM2 和 FSMC_PATT2 寄存器的 PWID、PTYP、PWAITEN 和 PBKEN。

2） 在通用存储空间写入命令字节，在 NWE 为低电平期间，CLE 输出高电平，此时该命令字节被 NAND FLASH 识别为一个命令，并锁存该命令，随后的页读操作不需要重复发送该命令。

3） 之后通过向存储器空间写入 4 个字节，作为读操作的起始地址。在 NWE 为低电平期间，ALE 输出高电平，此时 4 个字节被 NAND FLASH 识别为读操作的起始地址。在操作属性存储空间时，可以使 FSMC 产生不同的时序，实现某些 NAND FLASH 预等待功能。

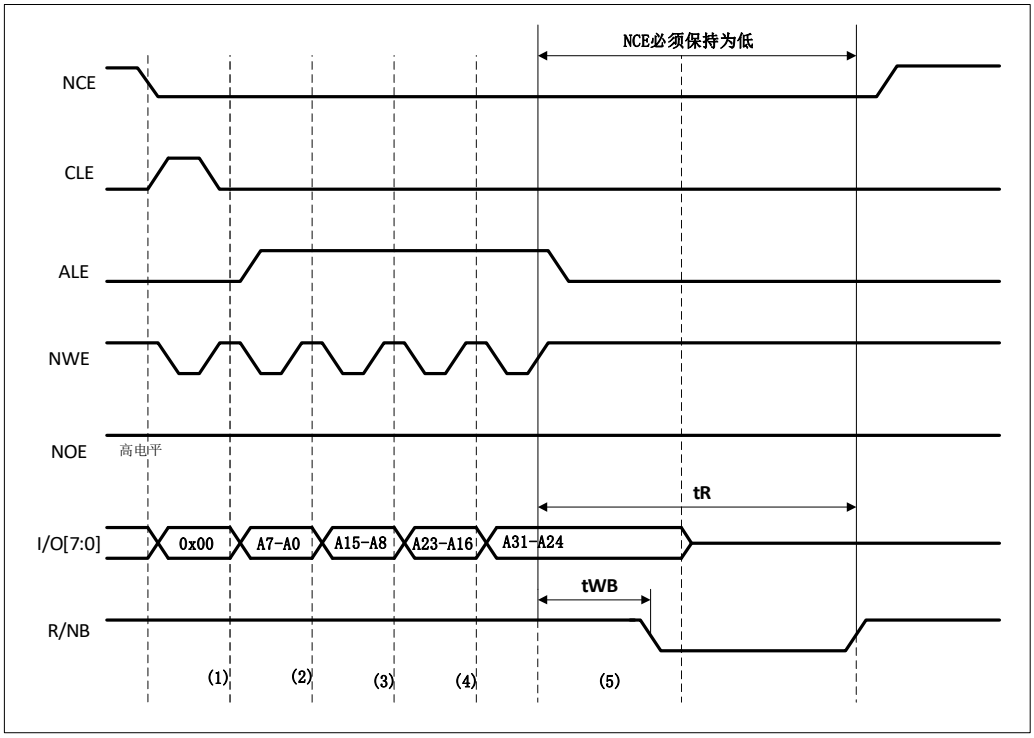
4） FSMC 控制器开始新的操作前需要等待 NAND FLASH 的 R/NB 信号变为高电平，在等待期间 FSMC 控制器需要保持 NCE 信号为低电平。

5） FSMC 控制器可以通过操作通用存储空间，可以从 NAND FLASH 逐字节的读出存储页。

26.2.4.5 NAND FLASH 预等待功能

某些特殊的 NAND FLASH 需要在输入最后一个地址字节后，R/NB 信号变为低电平，见图 26-16。图中(1)为 CPU 写字节命令 0x00 到 0x70010000；图中(2)为 CPU 写 NAND FLASH 的地址 A7-A0 至 0x70020000；图中（3）为 CPU 写 NAND FLASH 的地址 A15-A8 至 0x70020000；图中（4）为 CPU 写 NAND FLASH 的地址 A23-A16 至 0x70020000；图中（5）为 CPU 写 NAND FLASH 的地址 A31-A24 至 0x70020000；

图 26-16 操作特殊型 NAND FLASH



使用该功能时，通过配置 FSMC_PMEM2 寄存器的 MEMHOLD 位来保证 tWB 的时序，之后对 NAND FLASH 的读写操作，FSMC 控制器都会在 NEW 信号的上升沿至下一次操作之间插入（MEMHOLD+1）HCLK 保持延时。为了解决该问题，这里使用属性空间配置 ATTHOLD 数值使之符合 tWB 的时序，同时需要保持 MEMHOLD 为最小值。此时，只有在写入 NAND FLASH 地址的最后一个字节时，FSMC 控制器需要写入属性存储空间，其余 NAND FLASH 的读写操作使用通用存储空间。

26.2.4.6 NAND FLASH ECC 功能

FSMC 的 NAND FLASH 控制器包含 1 个纠错码计算硬件模块，可有效减少 CPU 在处理纠错码时的软件工作量。ECC 模块在读写 NAND FLASH 时，支持每 256、512、1024、2048、4096 或 8192 个字节中，矫正 1 个 bit 错误并且检测出 2 个 bit 错误。页大小对应 ECC 结果有效位，见表 26-30。在 ECC 计算电路使能后，ECC 模块监测 NAND FLASH 的数据总线和读/写信号（NCE 和 NWE）。ECC 使用时需注意如下：

- 在访问 NAND FLASH 时，出现在 D[15:0]总线上的数据被锁存并用于 ECC 计算。
- 当规定数目的字节已经被写入 NAND FLASH 或从 NAND FLASH 读出后，软件从 FSMC_ECCR2 寄存器读出 ECC 值。若需再次计算 ECC，先将 ECCEN 清 0，再写 1 重新使能 ECC 计算。

表 26-30 页大小对应 ECC 结果有效位

ECCPS[2:0]	页大小（字节）	ECC 有效位
000	256	ECC[21:0]
001	512	ECC[23:0]
010	1024	ECC[25:0]
011	2048	ECC[27:0]
100	4096	ECC[29:0]
101	8192	ECC[31:0]

26.3 寄存器描述

表 26-31 FSMC 相关寄存器列表

名称	访问地址	描述	复位值
R32_FSMC_BCR1	0xA0000000	SRAM/NOR-Flash 片选控制寄存器 1	0x000030DX
R32_FSMC_BTR1	0xA0000004	SRAM/NOR-Flash 片选时序寄存器 1	0xFFFFFFFF
R32_FSMC_PCR2	0xA0000060	NAND-Flash 控制寄存器 2	0x00000018
R32_FSMC_SR2	0xA0000064	FIFO 状态和中断寄存器 2	0x00000040
R32_FSMC_PMEM2	0xA0000068	通用存储空间时序寄存器 2	0xFCFCFCFC
R32_FSMC_PATT2	0xA000006C	属性存储空间时序寄存器 2	0xFCFCFCFC
R32_FSMC_ECCR2	0xA0000074	ECC 结果寄存器 2	0x00000000
R32_FSMC_BWTR1	0xA0000104	SRAM/NOR-Flash 写时序寄存器 1	0xFFFFFFFF

26.3.1 SRAM/NOR-Flash 片选控制寄存器 1（FSMC_BCR1）

偏移地址：0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Reserved	FACCEN	MWID[1:0]		MTYP[1:0]		MUXEN	MBKEN

位	名称	访问	描述	复位值
[31:20]	Reserved	RO	保留。	0
19	CBURSTRW	RW	批量写使能。 1：写操作处于同步模式； 0：写操作处于异步模式。	0
[18:16]	Reserved	RO	保留。	0
15	ASYNCWAIT	RW	异步传输期间的等待信号。 1：启用异步传输期间等待信号； 0：禁用异步传输期间等待信号。	0
14	EXTMOD	RW	扩展模式使能。 1：允许写使用 FSMC_BWTR 寄存器； 0：不允许写使用 FSMC_BWTR 寄存器。	0
13	WAITEN	RW	等待使能。当闪存存储器处于批量传输模式时，该位控制是否根据 NWAIT 信号插入等待信号。 1：使用 NWAIT 信号； 0：禁用 NWAIT 信号。	1
12	WREN	RW	写操作使能。 1：允许 FSMC 对存储器进行写操作； 0：禁止 FSMC 对存储器进行写操作。	1
11	WAITCFG	RW	等待时序配置。 1：NWAIT 信号在等待状态期间有效；	0

			0：NWAIT 信号在等待状态前的一个数据周期有效。	
10	WRAPMOD	RW	批量模式支持非对齐使能。 1：支持直接的非对齐批量操作； 0：不支持直接的非对齐批量操作。	0
9	WAITPOL	RW	等待信号极性。 1：NWAIT 等待信号为高有效； 0：NWAIT 等待信号为低有效。	0
8	BURSTEN	RW	批量模式使能。 1：使能批量模式； 0：不使能批量模式。	0
7	Reserved	RO	保留。	1
6	FACCEN	RW	NOR FLASH 访问使能。 1：允许对 NOR FLASH 访问； 0：禁止对 NOR FLASH 访问。	1
[5:4]	MWID	RW	数据总线宽度。 11：保留； 10：保留； 01：16 位； 00：8 位；	01b
[3:2]	MTYP	RW	储存器类型。 11：保留； 10：NOR FLASH； 01：PSRAM； 00：SRAM、ROM；	复位值
1	MUXEN	RW	地址/数据复用使能。 1：地址/数据复用； 0：地址/数据不复用。	复位值
0	MBKEN	RW	存储器块使能。 1：启用对应的存储器块； 0：禁用对应的存储器块。	复位值

26.3.2 SRAM/NOR-Flash 片选时序寄存器 1（FSMC_BTR1）

偏移地址：0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			

位	名称	访问	描述	复位值
[31:30]	Reserved	RO	保留。	0
[29:28]	ACCMOD	RW	异步访问模式。 11：模式 D； 10：模式 C；	00b

			01：模式 B； 00：模式 A。	
[27:24]	DATLAT	RW	数据保持时间。用于同步批量模式(仅适用于 NOR FLASH) 1111:第一个数据保持时间为 17 个 CLK 时钟； 1110:第一个数据保持时间为 16 个 CLK 时钟； …… 0000：第一个数据保持时间为 2 个 CLK 时钟；	1111b
[23:20]	CLKDIV	RW	时钟分频比(CLK 信号)。访问异步 NOR FLASH、SRAM 和 ROM 时，该参数不起作用。 1111：1 个 CLK 周期=16 个 HCLK 周期； 1110：1 个 CLK 周期=15 个 HCLK 周期； …… 0001：1 个 CLK 周期=2 个 HCLK 周期； 0000：保留。	1111b
[19:16]	BUSTURN	RW	总线恢复时间。(仅适用于 NOR FLASH) 1111：总线恢复时间=16 个 HCLK 周期； 1110：总线恢复时间=15 个 HCLK 周期； …… 0000：总线恢复时间=1 个 HCLK 周期；	1111b
[15:8]	DATAST	RW	数据保持时间。 11111111：数据保持时间=256 个 HCLK 周期； 11111110：数据保持时间=255 个 HCLK 周期； …… 00000001：数据保持时间=2 个 HCLK 周期； 00000000：保留。	0xFF
[7:4]	ADDHLD	RW	地址保持时间。(仅适用于异步操作) 1111：地址保持时间=16 个 HCLK 周期； 1110：地址保持时间=15 个 HCLK 周期； …… 0000：地址保持时间=1 个 HCLK 周期；	1111b
[3:0]	ADDSET	RW	地址建立时间。（仅适用于异步操作） 1111：地址建立时间=16 个 HCLK 周期； 1110：地址建立时间=15 个 HCLK 周期； …… 0000：地址建立时间=1 个 HCLK 周期；	1111b

26.3.3 NAND-Flash 控制寄存器 2（FSMC_PCR2）

偏移地址：0x60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												ECCPS[2:0]			TAR[3]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAR[2:0]			TCLR[3:0]				Reserved		ECCEN	PWID[1:0]		PTYP	PBKEN	PWAITEN	Reserved

位	名称	访问	描述	复位值
[31:20]	Reserved	RO	保留。	0
[19:17]	ECCPS	RW	ECC 页面大小 111：保留。 101：8192Byte 100：4096Byte 011：2048Byte 100：1024Byte 100：512Byte 100：256Byte	0
[16:13]	TAR	RW	ALE 至 RE 的延迟。 1111：16 个 HCLK 周期； 1110：15 个 HCLK 周期； …… 0000：1 个 HCLK 周期。	0
[12:9]	TCLR	RW	CLE 至 RE 的延迟。 1111：16 个 HCLK 周期； 1110：15 个 HCLK 周期； …… 0000：1 个 HCLK 周期。	0
[8:7]	Reserved	RO	保留。	0
6	ECCEN	RW	ECC 使能。 1：使能 ECC； 0：禁用 ECC。	0
[5:4]	PWID	RW	数据总线宽度。 11：保留； 10：保留； 01：16 位； 00：8 位。	01b
3	PTYP	RW	存储器类型。 1：NAND 闪存； 0：保留。	1
2	PBKEN	RW	NAND 存储器使能。 1：使能对应的存储器块； 0：禁用对应的存储器块。	0
1	PWAITEN	RW	等待功能使能。 1：使能； 0：禁用。	0
0	Reserved	RO	保留。	0

26.3.4 FIFO 状态和中断寄存器 2（FSMC_SR2）

偏移地址：0x64

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS

位	名称	访问	描述	复位值
[31:7]	Reserved	RO	保留。	0
6	FEMPT	RO	FIFO 空标志。 1：FIFO 空； 0：FIFO 不空。	1
5	IFEN	RW	中断下降沿检测使能。 1：使能； 0：禁用。	0
4	ILEN	RW	中断高电平检测使能。 1：使能； 0：禁用。	0
3	IREN	RW	上升沿中断检测使能。 1：使能； 0：禁用。	0
2	IFS	RW	中断下降沿状态。 1：产生中断下降沿； 0：没有产生下降沿。	0
1	ILS	RW	中断高电平状态。 1：产生中断高电平； 0：没有产生中断高电平。	0
0	IRS	RW	中断上升沿状态。 1：没有产生中断上升沿； 0：产生了中断上升沿。	0

26.3.5 通用存储空间时序寄存器 2（FSMC_PMEM2）

偏移地址：0x68

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

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

位	名称	访问	描述	复位值
[31:24]	MEMHIZx	RW	通用空间数据总线的高阻时间。 11111111：NAND FLASH 为 255 个 HCLK 周期； 11111110：NAND FLASH 为 254 个 HCLK 周期； ……	0xFC

			00000000：NAND FLASH 为 0 个 HCLK 周期。	
[23:16]	MEMHOLDx	RW	通用空间保持时间。 11111111：255 个 HCLK 周期； 11111110：254 个 HCLK 周期； …… 00000001：1 个 HCLK 周期； 00000000：保留。	0xFC
[15:8]	MEMWAITx	RW	通用空间等待时间。（需要加上由 NWAIT 信号变低引入的等待周期） 11111111：256 个 HCLK 周期； 11111110：255 个 HCLK 周期； …… 00000001：2 个 HCLK 周期； 00000000：保留。	0xFC
[7:0]	MEMSETx	RW	通用空间建立时间。 11111111：NAND FLASH 257 个 HCLK 周期； 11111110：NAND FLASH 256 个 HCLK 周期； …… 00000000：NAND FLASH 2 个 HCLK 周期。	0xFC

26.3.6 属性存储空间时序寄存器 2（FSMC_PATT2）

偏移地址：0x6C

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

位	名称	访问	描述	复位值
[31:24]	ATTHIZx	RW	属性空间数据总线的高阻时间。 11111111：255 个 HCLK 周期； 11111110：254 个 HCLK 周期； …… 00000000：0 个 HCLK 周期。	0xFC
[23:16]	ATTHOLDx	RW	属性空间保持时间。 11111111：255 个 HCLK 周期； 11111110：254 个 HCLK 周期； …… 00000001：1 个 HCLK 周期； 00000000：保留。	0xFC
[15:8]	ATTWAITx	RW	属性空间等待时间。（需要加上由 NWAIT 信号变低引入的等待周期） 11111111：256 个 HCLK 周期； 11111110：255 个 HCLK 周期； ……	0xFC

			00000001：2 个 HCLK 周期； 00000000：1 个 HCLK 周期。	
[7:0]	ATTSETx	RW	属性空间建立时间。 11111111：256 个 HCLK 周期； 11111110：255 个 HCLK 周期； …… 00000000：1 个 HCLK 周期。	0xFC

26.3.7 ECC 结果寄存器 2（FSMC_ ECCR2）

偏移地址：0x74

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

位	名称	访问	描述	复位值
[31:0]	ECC	RO	ECC 计算结果	0

26.3.8 SRAM/NOR-Flash 写时序寄存器 1（FSMC_BWTR1）

偏移地址：0x104

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

位	名称	访问	描述	复位值
[31:30]	Reserved	RO	保留。	0
[29:28]	ACCMOD	RW	访问模式。（该位在 EXTMOD 为 1 时有效） 00：访问模式 A； 01：访问模式 B； 10：访问模式 C； 11：访问模式 D。	00b
[27:24]	DATLAT	RW	数据保持时间。（NOR FLASH 同步批量模式） CLK 为闪存时钟。 1111：17 个 CLK 周期； 1110：16 个 CLK 周期； …… 0000：1 个 HCLK 周期。	1111b
[23:20]	CLKDIV	RW	时钟分频比（CLK）。 1111：1 个 CLK 周期=16 个 HCLK 周期； 1110：1 个 CLK 周期=15 个 HCLK 周期； …… 0001：1 个 CLK 周期=2 个 HCLK 周期；	1111b

			0000：保留。	
[19:16]	Reserved	RO		1111b
[15:8]	DATAST	RW	数据保持时间。 11111111：256 个 HCLK 周期； 11111110：255 个 HCLK 周期； …… 00000001：2 个 HCLK 周期； 00000000：保留。	0xFF
[7:4]	ADDHLD	RW	地址保持时间。 1111：16 个 HCLK 周期； 1110：15 个 HCLK 周期； …… 0001：2 个 HCLK 周期； 0000：保留。	1111b
[3:0]	ADDSET	RW	地址建立时间。 1111：16 个 HCLK 周期； 1110：15 个 HCLK 周期； …… 0001：2 个 HCLK 周期； 0000：1 个 HCLK 周期。	1111b

第 27 章 以太网收发器（ETH）

本章模块描述适用于 CH32V3x 和 CH32F20x 微控制器系列部分产品。

本章提到的“以太网收发器”是一个专有名词，意为微控制器内部的以太网（Ethernet）数据链路层收发器，通讯速率最高为千兆位每秒（1Gbps），是一个通讯外设。本章提到的“MAC”指的是以太网收发器在数据链路层的角色名称，是以太网收发器的组成部分。

以太网收发器（Ethernet Transceiver MAC）是微控制器一个重要的高速通讯部件，它允许微控制器以千兆位（Gigabit）的连接速度接入以太网，实现极快的数据沟通。

27.1 CH32F20x_D8C、CH32V30x_D8C 系列

27.1.1 主要特征

微控制器的以太网收发器是微控制器的一个重要高速通讯外设，它集成千兆 MAC（媒体访问控制器）、32 位宽的 DMA 控制器、管理计数器（MMC）、精确时间协议控制器（PTP）和一个十兆速度的以太网物理层（10BASE-T PHY）。在外接千兆以太网物理层（PHY）后，它能以千兆位（1Gbps）的速度接入以太网进行数据收发。以太网收发器工作在数据链路层，需要通过软件实现 TCP/IP 协议栈和接口。以太网收发器支持标准 MII、RMII 和 RGMII 三种 MII 接口连接 PHY，如果用户要达到千兆的接入速度，必须使用 RGMII 接口；以太网收发器通过 SMI 接口控制 PHY，接口的时序由 MAC 自动实现，无需用户通过软件生成。RGMII 接口支持发送时钟相位翻转和相对数据延迟，最大延迟 4 纳秒。以太网收发器的 MAC 支持标准 IEEE802.3 协议的以太网，支持魔法帧和特定唤醒帧。以太网收发器搭配的 DMA 控制器以描述符的形式进行数据的收发管理和内存搬运，描述符的数量由用户根据沟通密集程度自行确定，DMA 控制器能以 32 位宽的速度向描述符指定的内存空间写入接收到的数据或从取出将发送的数据。此外，以太网收发器的 MAC 还支持 IEEE1588 精确时间协议。

27.1.1.1 MAC 特征

- MAC 挂载在 AHB 总线上
- 支持 10M/100M/1000M 以太网
- 支持 MII/RMII/RGMII 接口
- RGMII 支持发送时钟延迟和翻转
- 支持全双工与半双工
- 支持自动插入帧头序列和 SFD
- 支持自动插入和校验 CRC
- 支持自动计算和校验 IP/ICMP/TCP/UDP 协议数据包的检验和
- 支持发送帧长控制
- 支持发送间隙调节
- 支持 VLAN 帧
- 支持帧接收地址的完美地址过滤、HSAH 过滤
- 支持帧发送地址过滤
- 支持多播广播帧接收控制
- 支持混杂模式
- 支持 SMI 管理接口（RGMII 与 MII 各一套）
- 支持魔法帧和自定义的唤醒帧唤醒微控制器
- 支持专门的以太网唤醒中断入口

- 支持链路层的数据回环

27.1.1.2 DMA 特征

- 32 位宽的 MAC 专享 DMA
- 最大程度减小 CPU 的操作
- DMA 挂载在 AHB 总线上，也同时做 AHB 总线的主设备直接访问 RAM
- 支持以字节对齐方式访问 RAM
- 以描述符的方式管理收发缓冲区
- 接收发送的一部分状态反馈在描述符中
- 可以动态修改非正在使用的描述符和缓冲区
- 支持手动停止或启动
- 支持链式或环式的形式连接描述符
- 支持收发完成中断等多种中断源

27.1.1.3 MMC 模块特征

- 支持手动复位、停止或冻结
- 多个支持发送和接收的多种计数模式的计数器
- 支持多种中断

27.1.1.4 PTP 模块特征

- 支持 IEEE 1588 协议
- 支持自动在收发时保存当前的时刻
- 自带一个 32 位的秒计时器和一个 32 位有符号的亚秒计时器
- 支持以精调和粗调两种方式调节时间
- 支持一个中断源
- 支持 PPS 输出

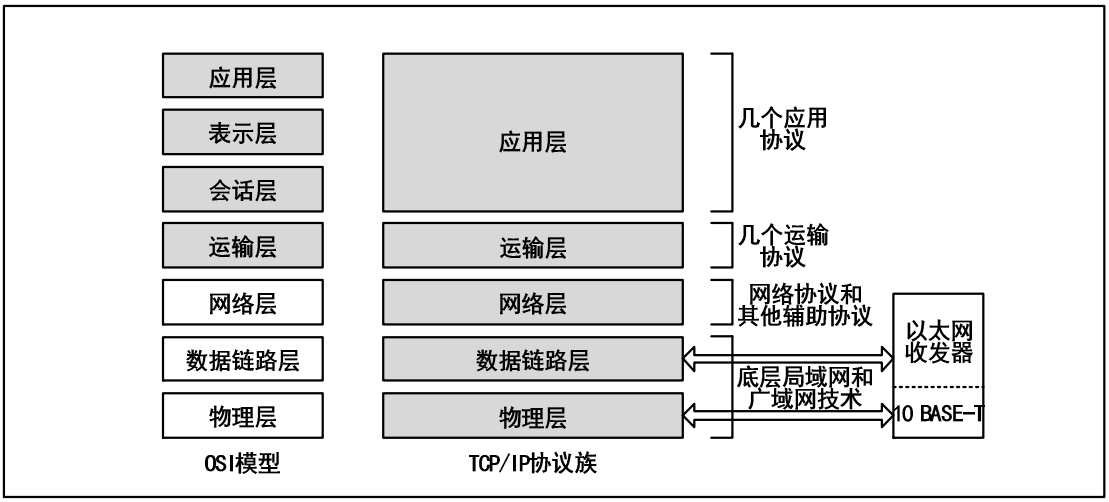
27.1.1.5 内部 10BASE-T 物理层特征

- 支持 10BASE-T
- 通过 SMI 接口管理
- 支持 MDIX 自动翻转
- 支持 MDI 接口的差分对 p/n 极性翻转
- 自持手动复位
- 支持物理层的数据回环
- 支持半双工

27.1.2 概述

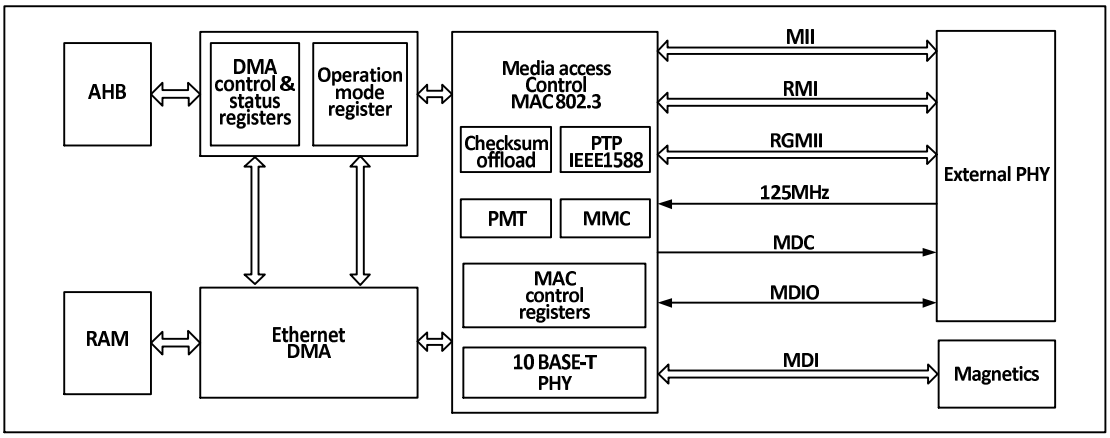
以太网收发器工作在 OSI 七层模型中的数据链路层和物理层。如果需要实现大于 10Mbps 的数据传输速率，需要外接百兆或千兆物理层芯片（PHY）实现物理连接。为了能在广泛使用的以太网中建立 IP、TCP 和 UDP 等协议的通讯，用户还需要用软件实现 TCP/IP 协议栈。以太网收发器由媒体访问控制层(MAC)、搭配的 DMA 及二者的控制寄存器及十兆物理层组成。

图 27-1 以太网收发器在 OSI 模型和 TCP/IP 模型中的位置



以太网收发器的 MAC 按照 IEEE802.3 协议的规范设计，搭配 32 位宽的 DMA，保证数据能快速地从网线上转发到微控制器的内存中。以太网收发器拥有强大完整的 DMA 控制寄存器、MAC 控制寄存器和模式控制寄存器，微控制器的 CPU 通过 AHB 总线操作以太网收发器的寄存器。以太网收发器通过 RGMII 接口和千兆以太网物理层连接，如果只需要百兆以太网的速度，则可以通过 MII 或 RMII 接口和百兆以太网物理层连接。以太网收发器的 MII、RMII 和 RGMII 接口的管脚是复用的，具体参看 27.3 节。以太网收发器通过 SMI 接口管理以太网物理层。使用以太网收发器时，AHB 总线的时钟不能低于 50MHz。

图 27-2 以太网收发器的结构框图



此外，以太网收发器还支持 IEEE1588 精确时间协议（PTP），为微控制器系统或片外提供精确的时间数据。

27.1.3 以太网收发器使用引脚的分布和配置

微控制器支持三种 MII 接口，即标准 MII、RMII 和 RGMII。MII/RMII 接口专用于十兆和百兆以太网物理层，RGMII 接口可用于十兆、百兆和千兆以太网。下表显示了标准 MII，RMII 和 RGMII 接口及内部物理层在封装引脚上的分布。

表 27-1 微控制器的媒体独立接口、内置物理层的介质相关接口
及其他相关的引脚分布和需要进行的配置

引脚	RGMII	RGMII 的引脚配置	MII	RMII	MII/RMII 的引脚配置
PA2	TXCLK		MDC		推挽复用输出
PC1	RXCTL	浮空输入	MDIO		推挽复用输出
PA1	RXD3	浮空输入	RX_CLK	REF_CLK	浮空输入
PA7	TXD0	推挽复用输出	RX_DV	CRS_DV	浮空输入
PC4	TXD1	推挽复用输出	RXD0	RXD0	浮空输入
PC5	TXD2	推挽复用输出	RXD1	RXD1	浮空输入
PB0	TXD3	推挽复用输出	RXD2		浮空输入
PB1	125MHz_IN	浮空输入	RXD3		浮空输入
PB10			RXER		浮空输入
PC3	RXD1	浮空输入	TX_CLK		浮空输入
PB11			TX_EN	TX_EN	推挽复用输出
PB12	MDC	推挽复用输出	TXD0	TXD0	推挽复用输出
PB13	MDIO	推挽复用输出	TXD1	TXD1	推挽复用输出
PC2	RXD0	浮空输入	TXD2		推挽复用输出
PB8			TXD3		推挽复用输出
PA0	RXD2	浮空输入	CRS		浮空输入
PA3	TXCTL	推挽复用输出	COL		浮空输入
PB5	PPS_OUT（推挽复用输出）				
PC6	10BASE-T_TX_p(无需 IO 配置)				
PC7	10BASE-T_TX_n(无需 IO 配置)				
PC8	10BASE-T_RX_p(无需 IO 配置)				
PC9	10BASE-T_RX_n(无需 IO 配置)				

27.1.4 物理层（PHY）管理和数据交互

以太网收发器的 MAC 通过站点管理接口（SMI 接口）对 PHY 进行管理，使用媒体独立接口（MII 接口）与 PHY 进行数据交互。微控制器支持的 MII 接口包括标准 MII（一般就写为 MII）、精简 MII（RMII）和精简的千兆 MII（RGMII）。微控制器在 MII/RMII 模式下和 RGMII 模式下使用不同管脚引出 SMI 接口。

27.1.4.1 SMI 接口

SMI 接口是一种串行通讯接口，使用 MDC（时钟线）和 MDIO（数据线）两线来访问 PHY 的寄存器实现对 PHY 的管理，最多可以管理 32 个 PHY 芯片。其中 MDC 是时钟线，空闲时保持低，MDIO 是数据线。SMI 的读写操作和帧的组成都是由 MAC 主导的，用户只需要写入地址和数据。相关的寄存器分别为 MII 地址寄存器（R32_ETH_MACMIIAR）和 MII 数据寄存器（R32_ETH_MACMIIDR）。

27.1.4.1.1 帧格式

SMI 接口管理帧的格式如下表 27-2 所示。

表 27-2 SMI 帧格式

	Preamble	STR	OP	PHY ADR	REG ADR	T	DATA	P
读	32 个“1”	01	10	PPPPP	RRRRR	Z0	DDDDDDDDDDDDDDDD	Z
写	32 个“1”	01	01	PPPPP	RRRRR	10	DDDDDDDDDDDDDDDD	Z

管理帧各域的定义如下：

1） Preamble：先导符，由 32 个“1”组成，用于 MAC 和 PHY 同步；
2） STR：起始符，固定为“01”；
3） OP：操作符，读为“10”，写为“01”；
4） PHY ADR：物理层地址，5 位；
5） REG ADR：寄存器地址，5 位；
6） T：转换符，两位，用来切换 MAC 和 PHY 对 MDIO 线的控制权。在读操作时，MAC 保持对 MDIO 线的高阻，PHY 对第一位保持高阻，对第二位下拉，并获取 MDIO 的控制权；在写操作时，MAC 对 MDIO 线先置高再下拉，PHY 对 MDIO 保持高阻状态，MAC 保持对 MDIO 的控制权；
7） DATA：数据域，16 位，MAC 对 PHY 读写的数据，MSB 在前；
8） P：MAC 和 PHY 都对 MDIO 保持高阻状态，但 PHY 的上拉电阻会把 MDIO 拉高。

27.1.4.1.2 读写时序

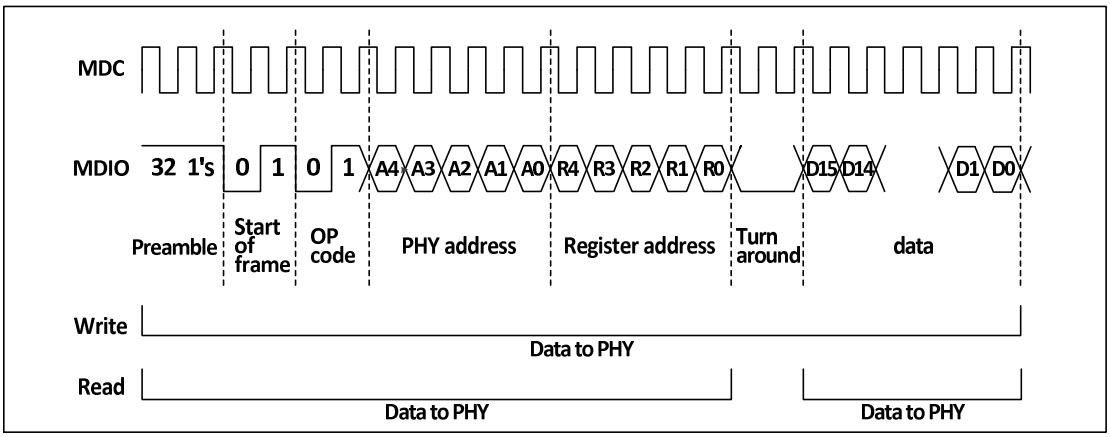
写 PHY 寄存器操作如下：

当用户设置了 MII 写位（ETH_MACMIIAR:MW）和忙位(ETH_MACMIIAR:MB)，SMI 接口会向 PHY 发送 PHY 地址和寄存器地址，然后发送数据(ETH_MACMIIDR)。在 SMI 接口发送数据的过程中，不能修改 MII 地址寄存器和 MII 数据寄存器的值；在此过程中,忙位保持为高，对 MII 地址寄存器或 MII 数据寄存器的写操作将被忽视，并且不对整个传输造成影响。当完成写操作时，SMI 接口将清除忙位，用户可以根据忙位判断写操作结束。如图 27-3 的写部分。

读 PHY 寄存器操作如下：

当用户把以太网 MAC 的 MII 地址寄存器(ETH_MACMIIAR)的 MII 忙位置位，而保持 MII 写位复位时，SMI 接口则发送 PHY 地址和寄存器地址，执行读 PHY 寄存器的操作。在整个传输过程中，用户不能修改 MII 地址寄存器和 MII 数据寄存器的内容。在传输过程中，忙位保持为高，对 MII 地址寄存器或 MII 数据寄存器的写操作将被忽视，并且不影响整个传输的正确完成。在读操作完成后，SMI 接口将清除忙位，并把从 PHY 读回的数据写入 MII 数据寄存器。如图 27-3 的读部分。

图 27-3 SMI 接口读写时速图



27.1.4.1.3 SMI 时钟

一般来说 SMI 时钟需要保持在一个固定的范围，以保证实际上 SMI 的时钟需要被 PHY 接收。具体参考用户选用的 PHY 芯片的手册。

SMI 时钟频率通过 MII 地址寄存器（ETH_MACMIIAR）的 CR 域从 AHB 时钟分频得到。下表显示和 SMI 时钟和 AHB 时钟的分频关系。默认选择 42 分频。

表 27-3 SMI 时钟和 AHB 时钟的分频关系

	适合搭配的 AHB 时钟的范围	SMI 时钟
0000b	60MHz 以上	AHB 时钟/42
0010b	20-35MHz	AHB 时钟/16
0011b	35-60MHz	AHB 时钟/26
其它值	无意义	

27.1.4.2 MII/RMII 接口

27.1.4.2.1 概述

媒体独立接口（MII）是 MAC 和 PHY 进行数据沟通的接口，根据速度和引线数量区分，有标准 MII、RMII、GMII、RGMII 和 SGMII 等。媒体独立接口一般是有接收、发送各一组，每组由时钟线、若干数据线和辅助线组成。支持的 MII 接口为：标准 MII/RMII 支持十兆和百兆物理层，RGMII 支持十兆、百兆和千兆物理层。由于市场上常用的物理层一般是 10M/100M 自动协商物理层、10M/100M/1000M 自动协商物理层，所以用户在使用 10M/100M 自动协商物理层时应该使用标准 MII/RMII 接口，在使用 10M/100M/1000M 物理层时应该使用 RGMII 接口。

一般情况下，媒体独立接口发送方向（TX 开头）的时钟和数据是由 MAC 发出的，接收方向（RX 开头）的时钟和数据是由 PHY 发出的，但是 RMII 的时钟是公用的。同方向的引脚走线应注意等长布线，具体布线要点参看用户选用的物理层芯片的 Layout 手册或本手册 27.9 章节。

27.1.4.2.2 引脚

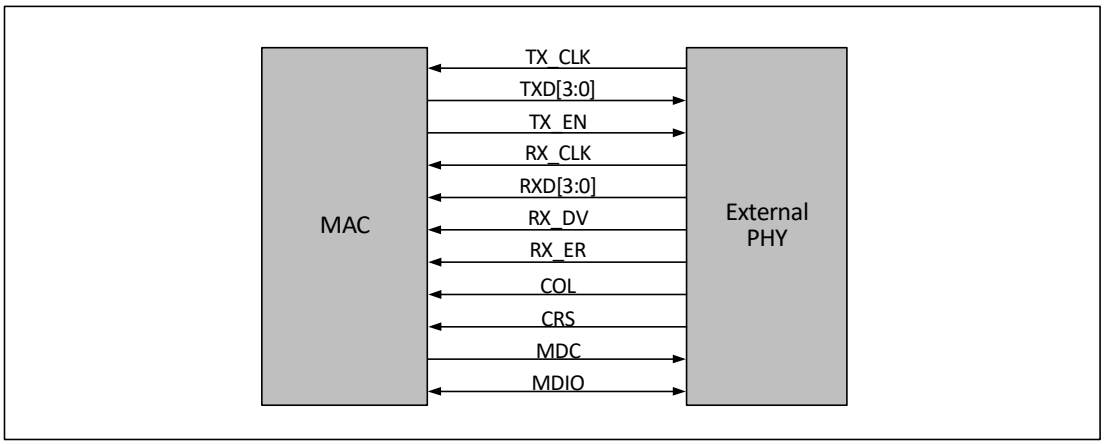
MII 接口的引脚和功能如下：

表 27-4 MII 接口的引脚和功能

引脚	功能
TXC	发送时钟（2.5MHz 或 25MHz）
TXEN	发送数据使能
TXD0	发送数据线[0:3]
TXD1	
TXD2	
TXD3	
RXC	接收时钟（2.5MHz 或 25MHz）
RXDV	指示接收数据有效
RXER	接收数据错误
RXD0	接收数据线[0:3]
RXD1	
RXD2	
RXD3	
COL	冲突检测
CRS	载波检测

图 27-4 显示了使用 MII 接口的定义时，MAC 和 PHY 如何连接。

图 27-4 以太网收发器在使用 MII 规范时，MAC 和 PHY 如何接线



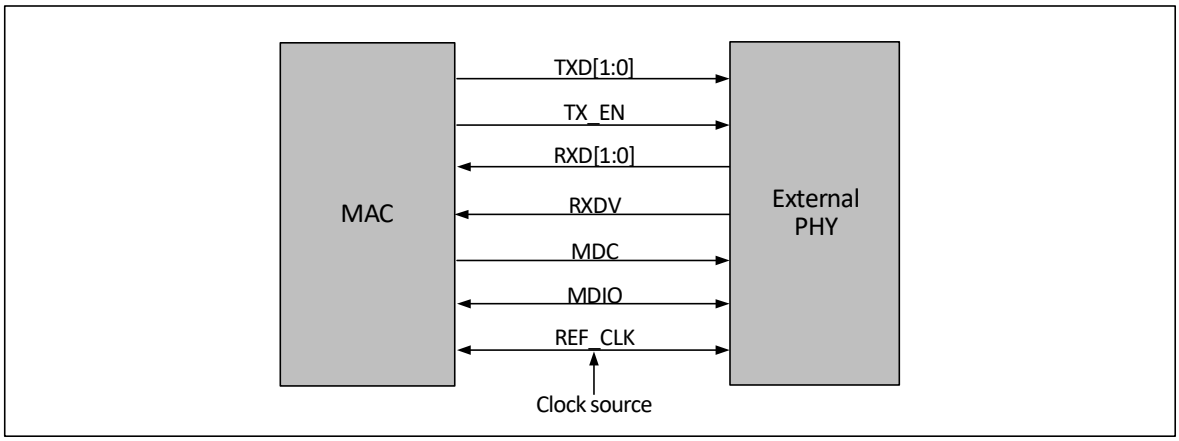
RMII 接口的引脚和功能如下：

表 27-5 RMII 接口的引脚和功能

引脚	功能
CLK_REF	收发时钟（50MHz）
TXEN	发送数据使能
TXD0	发送数据线[0:1]
TXD1	
CRSDV	接收数据有效
RXD0	接收数据线[0:1]
RXD1	

使用 RMII 接口时，收发时钟共用同一根线传输，时钟频率固定为 50MHz。当 RMII 使用在 10Mbps 的速率时，RX 和 TX 每隔 10 个周期采样一个数据，数据线需要将数据保留 10 个周期；MAC 和 PHY 需要使用同一个时钟源。可以将 MAC 的 MCO 输出接到 PHY 的外部时钟源输入引脚上。图 27-5 显示了使用 RMII 接口时，MAC 和 PHY 如何接线。

图 27-5 使用 RMII 接口时，MAC 和 PHY 的接线示意图



注：以太网收发器需要从 RXC 引脚输入外界提供的 REF_CLK 50MHz 时钟频率。

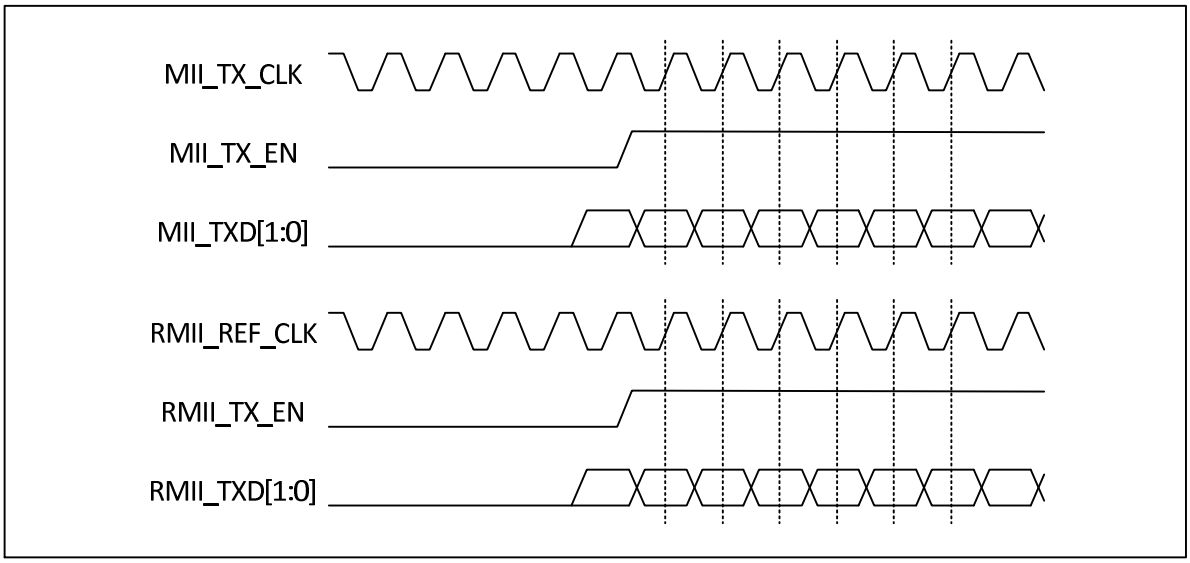
27.1.4.2.3 时序

RMII 和 MII 的时序的区别在于：

- RMII 的时钟固定为 50MHz，而 MII 的收发时钟随着实际连接速度的不同而工作在 2.5MHz 或 25MHz；
- RMII 收发共用同一个时钟线；
- RMII 是两位数据线，而 MII 是四位数据线。

如图 27-6 显示了 RMII 和 MII 的时序。

图 27-6 MII 和 RMII 的时序图



27.1.4.3 RGMII 接口

27.1.4.3.1 引脚

RGMII 的引脚和功能如下表

表 27-6 RGMII 的引脚和功能

引脚	功能
TXC	发生时钟
TXCTL	发生数据控制
TXD0	发送数据线[0:3]
TXD1	
TXD2	
TXD3	
RXC	接收时钟
RXCTL	接收数控制
RXD0	接收数据线[0:3]
RXD1	
RXD2	
RXD3	

注：1. TXCTL/RXCTL 在本微控制器的以太网收发器中用作收发数据有效信号来使用；
2. RGMII 有独立的 SMI 接口。

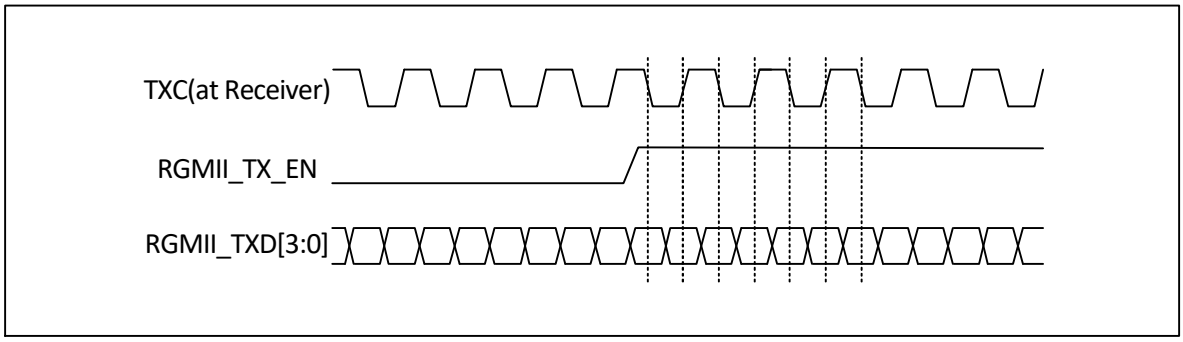
27.1.4.3.2 时序

RGMII 工作在十兆和百兆速率的模式下，时序和 MII 类似；工作在千兆时，RGMII 的时钟为 125MHz，采用双边沿采样的方式。RGMII 不支持半双工。

由于 RGMII 使用双边沿采样且工作在 125MHz 的时钟频率下，因此电路设计人员应注意信号完整性问题。

RGMII 的接收方接收到的时钟应相对数据滞后 90° 以保证正确采样。以太网收发器设计了发送时钟延迟输出和相位翻转的功能。

图 27-7 RGMII 的时序示意图



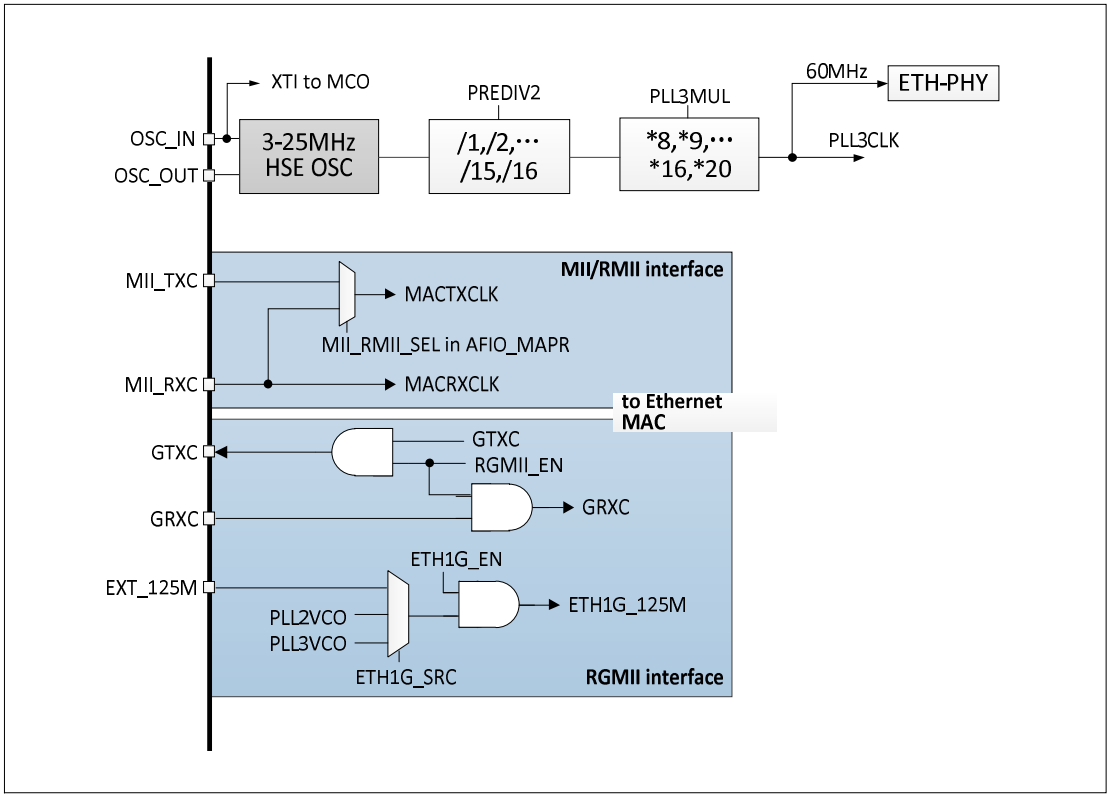
27.1.4.4 内部 10M 物理层使用时的注意事项

在置位扩展寄存器的内部物理层使能位时，所有的 MII 相关的配置都将被无视，对 SMI 接口的读写将直接映射到内部物理层，且忽略 SMI 接口写入的物理层地址。用户可以对内部物理层的寄存器进行访问以获取物理层连接状态，详见 27.8.5 章节。

27.1.4.5 时钟产生和输出

27.1.4.5.1 外设时钟源配置

图 27-8 以太网外设时钟树



由上图可以看出，内部 10M 以太网物理层的时钟由 PLL3 提供，且必须为 60MHz。使用内部物理层时，需要把扩展寄存器的第 2 位置位，置位后，MII/RMII/RGMII 相关的设置均无效。

在使用 MII 时，收发时钟都由外部的物理层提供，为 2.5MHz 或 25MHz。在使用 RMII 时，REF_CLK

是唯一的时钟，从 RXC 引脚输入微控制器，固定为 50MHz，使用 RMII 时需要将 AFIO 重映射寄存器中的第 23 位置位。

在使用 RGMII 时，MAC 需要频率为 125MHz 的时钟，由内部 PLL2/PLL3 的 VCO 产生(VCO 输出是 PLL 输出频率的两倍)或外部输入，通过 EXT_125M 引脚输入 125MHz 时钟。RGMII 的发送时钟由 MAC 产生，接收时钟由 PHY 产生。开启 RGMII 需要在扩展寄存器中将第 3 位置位，并在 RCC 的第二个配置寄存器中将第 22 位置位，同时在[21:20]中选择合适的 125MHz 时钟源，详见官网 EVT 例程。

27.1.4.5.2 MCO 输出

已知 MCO 支持 HSE、HSI、系统主频、PLLCLK/2、PLL2CLK、PLL3CLK、PLL3CLK/2 和 XTI 这八种时钟的输出，用户可以利用 MCO 输出应用中所需要的频率，比如常见物理层芯片所需要的 25MHz 时钟，并以此节省一颗晶体。MCO 的输出频率不应大于 100MHz。

27.1.5 IEEE802.3 和 IEEE1588

IEEE802.3 协议及其补充协议组成目前以太网的官方标准，它详细定义了目前使用的以太网的方方面面。我们可以认为以太网处于 OSI 模型中的物理层和数据链路层的一部分。本节在应用的角度上讨论 IEEE802.3 协议中用户需要用到的部分，即帧格式和 MAC 地址相关的内容。同时，微控制器的以太网收发器还支持 IEEE1588 精确时间协议，本文还将讨论 IEEE1588 的部分内容。

IEEE802.3 协议构建的以太网模型中，数据传输单元是以太网帧。以太网帧在不同的介质上以不同的速率传输时有不同的编码形式，接收后经过物理层解码，通过 MII 接口发给 MAC，MAC 校验和过滤通过后，由 TCP/IP 协议栈提取出应用信息发给不同的应用进程。

27.1.5.1 帧格式

以太网帧的帧格式如表 27-7 所示。

表 27-7 普通以太网帧格式

前同步码	SFD	目标地址	源地址	长度或类型	数据域	CRC
7 Bytes	1Byte	6 Bytes	6 Bytes	2 Bytes	46-1500 Bytes	4 Bytes
总长度：64 至 1518 字节加上 8 字节的物理层首部（前同步码和 SFD）						

前同步码：56 位（7 字节）交替的低电平和高电平跳跃，值固定，用十六进制表示即为 0xAA-0xAA-0xAA-0xAA-0xAA-0xAA-0xAA。此域用来进行时钟同步。该域由硬件自动添加/去除，用户不需要理会。

SFD：帧首定界符，8 位（1 字节），值为 10101011b，SFD 用来提醒接收方这是最后一次进行时钟同步的机会，其后为目标地址。该字节由硬件自动添加/去除，用户不需要理会。

目标地址：发送这个帧的设备希望接收这个帧的设备地址，这里指的是硬件地址，又称 MAC 地址，由 IEEE 为生产商分配，全球唯一，6 字节 48 位。本公司（WCH）的微控制器的 MAC 地址出厂时已烧录在芯片内部。地址的发送遵循低有效位在前的原则。。

源地址：发送这个帧的设备的硬件地址。源地址必须为单播地址。

长度或类型：2 字节，以太网一般用作类型，IEEE802.3 标准中也用作长度，以 1536（即 0x0600）分界，1536 以上表示协议，表示数据部分是根据何种上层协议组织起来的，例如 0x0806 表示 ARP，0x0800 表示 IPv4，0x86dd 表示 IPv6；1536 以下表示数据长度。

数据域：最小 46 字节，最大 1500 字节。当数据域不足 46 字节需要加填充增至 46 字节，超过 1500 字节请另组一帧。数据域中装载着需要实际发送的数据。

CRC：循环冗余校验，此处用的是 CRC32 校验。

27.1.5.2 帧发送

以太网收发器发送数据帧的时候，其专有的 DMA 控制器从发送描述符（trans descript 在 14.1.1 节）指定的 RAM 中取出要发送的数据通过专用数据总线压进 MAC。FIFO 的填充程度会返回到 DMA 控制器中，当要发送的数据全部发送完，DMA 控制器会向 MAC 发送一个数据开始信号，MAC 会启动发送；从 FIFO 中读取数据，向 MII 接口（RMII/RGMII）发送先导、SFD，发送实际数据；在 DMA 控制器向 MAC 发送数据结束信号后，MAC 加上自动计算的 CRC。

MAC 会按照配置自动设备发送帧的类型，并为其中的检验和域计算校验和，并取代原来校验和域的值。此功能可以关闭。

MAC 会自动向填充后面加上 CRC32 校验，也可以设置成不加上 CRC 校验位。按照 IEEE802.3 协议的规定，数据域长度不低于 46 字节，当 MAC 检测到将要发送的数据域短于 46 字节时，会自动在数字域后加上填充，如果选择自动填充，则必然会加上 CRC32 校验，无视寄存器配置。MAC 采用的 CRC32 检验公式如下：

G(x)=x32 +x26 +x23 +x22 +x16 +x12 +x11 +x10 +x8 +x7 +x5 +x4 +x2 +x +1

在使能了 IEEE 1588(PTP)功能的情况下，MAC 在发送以太网帧时，将保存当前时间戳在描述符中，但同时会覆盖掉部分信息，用户可以在接收中断中及时读取时间戳并补全描述符。

27.1.5.3 帧接收

以太网收发器在接收以太网帧的时候，以太网帧从 MII、RMII 或 RGMII 进入到 MAC，在 MAC 中首先进入到 FIFO，然后被 DMA 转发到 RAM 中的缓冲区。MAC 会对以太网帧进行过滤和检查，过滤包括完美地址过滤和 HASH 过滤，检查则一般是对帧进行帧长度，IP/ICMP/TCP/UDP 的检验和 CRC 检查，未通过过滤或检查的帧会被描述符标记出来或被丢弃，超过长度的包可能会被掐断。

在使能了 IEEE 1588(PTP)功能的情况下，接收帧时，MAC 会保存当前时间戳在描述符中，但同时会覆盖掉部分信息，用户可以在接收中断中及时读取时间戳并补全描述符。

27.1.5.4 帧过滤

使用 MAC 帧过滤器可以对接收的帧的目标 MAC 地址和源 MAC 地址进行完美过滤或 HASH 过滤，并可以设置使不通过过滤的帧被丢弃、被接收、或是被在接收描述符中标记出来。此功能请详细阅读 MAC 帧过滤寄存器（R32_ETH_MACFFR）的描述。MAC 内置了四个 MAC 地址寄存器，其中 MAC 地址寄存器 0 被默认用作存储自身的 MAC 地址，其余三个 MAC 地址寄存器可以被用作完美过滤，与接收到帧的源地址或目标地址对比。在置位 R32_ETH_MACFFR：RA 后，所有接收到的帧都将会被收到，但相应的状态会在之后描述符的第一个字的状态域中被标记出来，如果置位 R32_ETH_MACFFR：PM 会有类似的效果，但是状态不会被标记。在置位 R32_ETH_MACFFR：DAIF/SAIF 后，会有结果翻转的效果，原来通过过滤的将会被丢弃或标记，未通过过滤的反而会被转给 RAM 中的缓冲区。

27.1.5.4.1 单播过滤

MAC 通过 HPF 位和 HU 位来确定单播帧进行 HASH 过滤还是完美地址过滤。

27.1.5.4.2 多播过滤

MAC 通过 HPF 位和 HM 位来确定单播帧进行 HASH 过滤还是完美地址过滤。当 PAM 位置位后，所有的多播包都能通过过滤器。

27.1.5.4.3 广播过滤

通过置位 BFD 位，MAC 可以阻断所有的广播包。

27.1.5.4.4 源地址过滤选择

通过设置 MAC 地址寄存器中的 AE 位,可以启用该 MAC 地址寄存器，而设置 SA 位可以决定将该 MAC 地址寄存器作为源地址样本还是目标地址样本来进行对比。

27.1.5.4.5 小结

对目标地址和源地址的过滤设置分别如表 27-8 和表 27-9。

表 27-8 R32_ETH_MACFFR 各个位的设置对接收到帧的目的 MAC 地址的接受程度

帧类型	M	PF	U	DAIF	M	PAM	效果
广播帧	1	-	-	-	-	-	通过
	0	-	-	-	-	-	不通过（BFD 置位）
单播帧	1	-	-	-	-	-	所有帧通过
	0	-	0	0	-	-	完美滤波匹配时通过
	0	-	0	1	-	-	完美滤波匹配时不通过
	0	0	1	0	-	-	HSAH 滤波匹配时通过
	0	0	1	1	-	-	HSAH 滤波匹配时不通过
	0	1	1	0	-	-	完美滤波或 HSAH 滤波匹配时通过
	0	1	1	1	-	-	完美滤波或 HSAH 滤波匹配时不通过
多播帧	1	-	-	-	-	-	通过
	-	-	-	-	-	1	通过
	0	-	-	0	0	0	完美滤波匹配时通过
	0	0	-	0	1	0	HSAH 滤波匹配时通过
	0	1	-	0	1	0	完美滤波或 HSAH 滤波匹配时通过
	0	-	-	1	0	0	完美滤波匹配时不通过
	0	0	-	1	1	0	HSAH 滤波匹配时不通过
	0	1	-	1	1	0	完美滤波或 HSAH 滤波匹配时不通过

表 27-9 R32_ETH_MACFFR 各个位的设置对接收到帧的源 MAC 地址的接受程度

帧类型	RA	SAIF	SAF	效果
单播帧	1	-	-	所有帧通过
	0	0	0	完美过滤器匹配时对此帧进行标记，但不丢弃未通过的帧
	0	1	0	完美过滤器不匹配时对此帧进行标记，但不丢弃未通过的帧
	0	0	1	完美过滤器匹配时通过，丢弃未通过的帧
	0	1	1	完美过滤器匹配时不通过，丢弃未通过的帧

注：“-”表示不关心该位的设置。

27.1.5.5 MMC

管理计数器（MMC）的作用主要是对各种指示帧接收发送状态和 MAC 运转状态进行计数。它可以产生设置并产生中断。一般情况下，我们可以通过接收“好”帧计数器（MMCRGUFCR）获取当前接收到的完好的帧的数量，通过发送“好”帧计数器（MMCTGUFCR）来获取成功发送出去的帧的数量，通过接收 CRC 校验错误的帧计数器（MMCRFCECR）查看有无接收 CRC 错误的帧，一般情况下，如果数据时正确的但是 CRC 错误，可以认为 RGMII 线 layout 方面有问题。

用户需要明晰一个概念：什么样的帧是“好”的帧。正常情况下，只要一个帧启动发送，它的帧长符合要求（开启了自动填充），并且开启了自动计算 CRC，那么它就会是一个“好”帧。而一个帧被接收到，只要它 CRC 正确，帧长在以太网帧长度范围内，帧长和长度域的值一致（如果长度类型域表示的

是长度），或没有发生不对齐，那么就会被认为是一个“好”的接收帧。

27.1.5.6 PMT

27.1.5.6.1 概述

PMT（power management）部分的功能主要是通过以太网使微控制器从低功耗模式唤醒。千兆以太网控制器支持两种帧使系统唤醒，即魔法帧和（远程）唤醒帧。当以太网收到这两种帧时，由于微控制器处于低功耗模式，因此并不一定会产生接收中断，即使其合法，但是如果其通过 MAC 的魔法帧和唤醒帧识别，就会产生 PMT 中断，PMT 中断是和以太网中断独立的中断，通过查询 PMT 控制和状态寄存器就能查出是哪种以太网帧产生了中断。

27.1.5.6.2 魔法帧

AMD 公司定义的魔法帧（magic package，也有称幻数据包）是常用的一种唤醒微控制器的以太网帧，它有固定而特殊的帧格式，即通过帧过滤能让目标网卡接收到，可以封装成广播帧，帧中有连续 6 个字节的全高（0XFF），之后紧跟 16 次重复的目标网卡的 MAC 地址。这个组合可以存在于帧的任何位置，因此魔法帧可以封装成 Mac 帧或 IP 包甚至 UDP 包。以下是魔法帧的格式。

xx xx xx xx xx xx xx xx(之前的数据无限制,甚至可以没有之前的数据)—ff ff ff ff ff ff 84 c2 e4 01 02 03 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 xx xx xx xx xx xx（有的网卡要求魔法帧最后附带上密码）。

27.1.5.6.3 唤醒帧

由于魔法帧的格式存在限制，（远程）唤醒帧的格式可以由用户自行定义。只要是通过远程唤醒帧过滤寄存器（ETH_MACRWUFFR）组的以太网帧都被认为是唤醒帧，用户通过设置远程唤醒帧过滤寄存器组来定义满足自己需要的唤醒帧。需要注意的是，远程唤醒帧过滤寄存器组只有一个地址入口，用户可以通过连续的八次写操作向其中写数据，再通过连续的八次读数据即可读出原来写进远程唤醒帧过滤寄存器组的数据。

远程唤醒帧过滤寄存器组的结构如下表：

表 27-10 远程唤醒帧过滤寄存器组的结构

过滤寄存器 0	字节屏蔽寄存器 0							
过滤寄存器 1	字节屏蔽寄存器 1							
过滤寄存器 2	字节屏蔽寄存器 2							
过滤寄存器 3	字节屏蔽寄存器 3							
过滤寄存器 4	保留	命令 3	保留	命令 2	保留	命令 1	保留	命令 0
过滤寄存器 5	偏移 3		偏移 2		偏移 1		偏移 0	
过滤寄存器 6	过滤器 1				过滤器 0			
过滤寄存器 7	过滤器 3				过滤器 2			

由上表可以看出，字节屏蔽寄存器、命令、偏移好过滤器这四个域共同作用能确定一个帧是否是远程唤醒帧，实际上可以设置四种不同的满足要求的帧。4 位命令域的最高位表示对什么样的帧起作用，1 为只对多播地址有效，0 为只对单播地址有效；命令域的第 2 位和第 1 位保留，第 0 位为使能位，置高表示启用这组过滤器；偏移域表示从帧头开始偏移多少个字节开始计算 CRC16 的值，最小填 12，实际生效的值为此域的值加一，如若偏移域为 1，则为从开头第 13 个字节开始计算 CRC16 的值，即刚好为网络层的第一个字节开始。32 位字节屏蔽表示从偏移域定义的开头开始的 31 个字节，哪些需要参与 CRC16 的计算，字节屏蔽寄存器的最高位必须为 0，最多有 31 个字节全部参与计算。而过滤器域则存放了用户

期望计算出的 CRC 结果的值，MAC 会将自己计算出的 CRC16 的值与这个域之中的值进行比对，若果一致，则当前帧通过远程唤醒帧过滤器的过滤，被 MAC 认定为（远程）唤醒帧，如果使能了唤醒帧中断和 PMT 中断，则还会产生 PMT 中断。

另外，根据根据 PMT 控制状态寄存器的位定义，如果置位了 GU 位，那么通过帧过滤的单播帧也会被认为是唤醒帧。

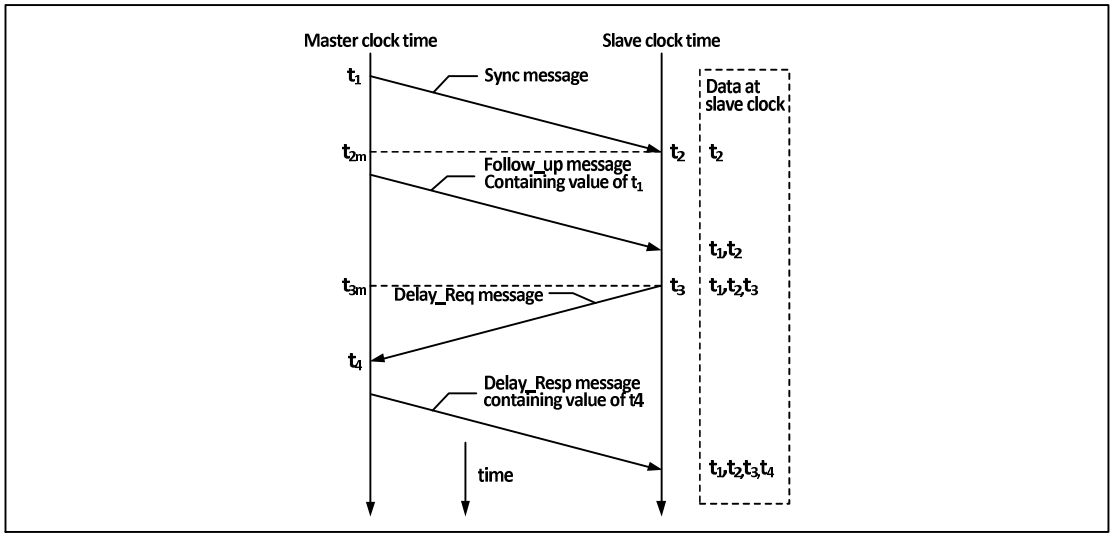
27.1.5.7 IEEE1588 PTP

27.1.5.7.1 PTP 原理和实现

IEEE1588 标准定义了一套获取时间的精确协议，它的目标是实现 10 微秒误差之内的时间同步。原本 NTP 协议已经可以实现 200 微秒的级别的时间同步，但实际上这个级别还无法满足工业自动化领域的时间同步需求，于是网络精密时钟同步委员会起草了 PTP 协议，并在 2002 年底被 IEEE 标准委员会通过，作为 IEEE1588 标准。

PTP 协议的实现需要主机和从机都能精确记录 MAC 接收和发送以太网帧的时间，这需要主机和从机都有一套自己的高精度时间计数器。随后，支持 PTP 的主从机通过一套流程进行对时，从机可以以此得到自己和主机的时间差并进行修正。下图显示了主从机对时的流程：

图 27-9 IEEE1588 PTP 协议同步报文时序图



分步描述：

- 主机向从机发送 syn 消息，从机接收到此消息，记录下接收到 syn 消息时的本地时间 t2；
- 主机向从机发送 follow_up 消息，包含主机发送 syn 消息时的主机时间 t1；
- 从机向主机发送 delay_req 消息，从机记录下发送时间 t3；
- 主机向从机发送 delay_resq 消息，包含 delay_req 消息的接收时间 t4；

实际使用中，主机是每隔两秒对外发一次 syn 消息的，而每隔一个 syn 消息都可以认为是上个 syn 消息的 follow_up 消息，他们都会附带上一次 syn 消息的发送时间。从机经过这个过程可以得知主机网络到从机网络延迟时间 Tdelay，然后算出主机时间和从机时间的时间偏移。

$$Tdelay = \frac{(t2-t1) + (t4-t3)}{2}$$

任一主机发出的时间减去偏移即是主机的时间。

PTP 的同步流程一般通过 UDP 协议实现，当然用户也可以自定协议实现。PTP 高度依赖内网的延迟稳定性。

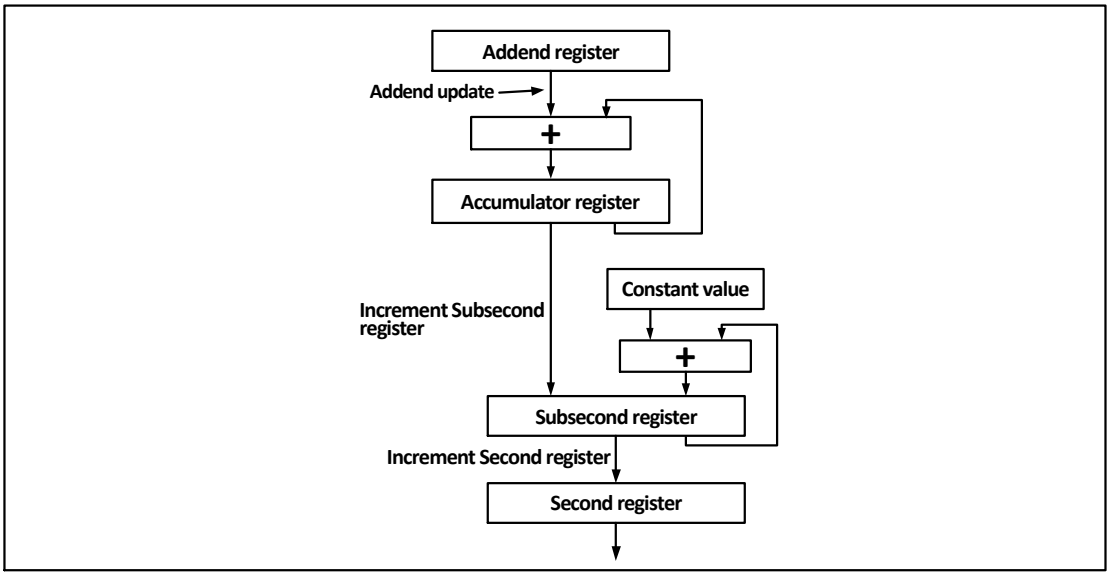
27.1.5.7.2 本地时间更新与校正

为了实现 PTP 的主机，设备本地是需要有一个高精度的时间计数器的，起码要精确到纳秒级。微控制器的千兆以太网计数器的 PTP 模块拥有一个 32 位以秒为单位的计数器，和一个 31 位的亚秒计数器，当亚秒计数器溢出时会引起秒计数器自增，因此本地时间的分辨率可以做到 0.46 纳秒左右。

本地时间的更新方式分为两种，即粗调和精调。粗调方式的更新时机是由外部决定的，在需要进行时间更新时，将时间戳控制寄存器的时间更新位（PTPTSCR:TSSTU）置位，微控制器的 PTP 模块会将秒计数器和亚秒计数器减去或加上时间戳更新寄存器（TSHUR、TSLUR）的值。粗调是一种较为简单且方便的时间更新机制，但是精确较差。

精调的时间更新方式是更常用的。其更新流程如下：

图 27-10 使用精调方式更新时间的流程



使用精调模式的方式需要对系统主频和精调的流程有较清晰的了解。与粗调方式不同，精调方式更新事件的时机是累加器（32 位，寄存器列表中未列出。图中的 Accumlator register）溢出，累加器在每个系统主频时钟周期都会自增加数寄存器（PTPTSAR,图中的 Addend register）的值，一旦溢出就会产生时间更新事件，即亚秒自增寄存器（PTPSSIR，图中的 Constant Value）的值会加到亚秒计数器（Subsecond register），完成时间更新。在亚秒计数器溢出时，秒计数器会自增。而实际上，亚秒计数器自增一位的时间为 1/(2^31)=0.46566128730…纳秒，用户需要自行保证累加器溢出花费的时间刚好等于亚秒自增寄存器的值乘以亚秒计数器自增一位的时间。

本地时间的校正较为简单，将时间戳控制寄存器的时间校正位（PTPTSCR:TSSTI）置位，则秒计数器和亚秒计数器的值会被时间戳更新计数器（TSHUR、TSLUR）的值替代。

27.1.6 DMA 操作

27.1.6.1 概述

在以太网收发器中，数据从 MII 接口进入 FIFO，然后被 DMA 转入 RAM 中。即使是最大的普通以太网帧，数据部分达到最大的 1500 字节，也只需要几十微秒左右就能接收发送完毕，即使算上 MAC 的接收检测，DMA 转移时间和帧间隔时间，CPU 也需要在一百微秒内就要处理一个帧。以太网收发器的优势是速度快和吞吐量大，为了维持这个优势，必须把以太网帧接受发送过程中，需要 CPU 进行的干预尽可能得减少，这里就要使用以太网收发器专用 DMA。

以太网使用的 DMA 是 32 位的，它通过两种数据结构接受 CPU 的管理：传统的控制和状态寄存器，接收和发送描述符。由于 DMA 是 32 位宽的，所以要求收发描述符队列在 RAM 中的基地址是 4 字节对齐的。收发缓冲区则没有对齐要求。

27.1.6.2 DMA 描述符（DMA Desciptor）

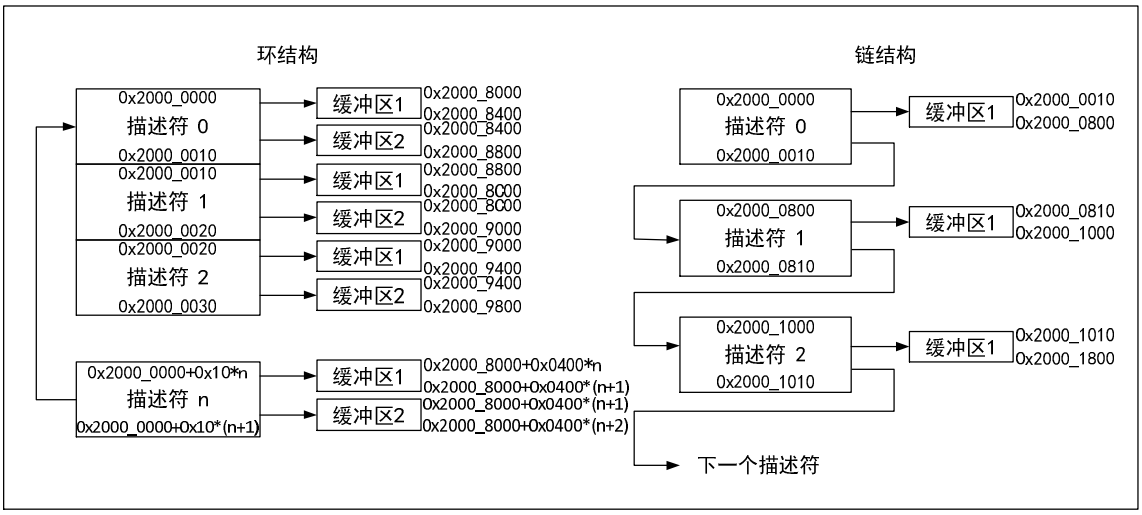
用户使用传统外设主要是通过写寄存器的控制位实现，并且以读取状态寄存器的方式来获取外设的状态和返回信息。这些寄存器是独立于内核，SRAM 和非易失性存储器之外的空间，是实际存在的，可以称之为“硬件寄存器”。传统的通讯外设，比如 UART 或 SPI，都有一个数据寄存器来暂存收发的数据，有一组 DMA 将收到的所有数据统一保存到特定的地址空间。

而以太网收发器以其极高的数据传输速度、极大的数据吞吐量和独特的数据帧组织形式，使得它的操作不同于传统的通讯收发器。以太网收发器尽可能密集地接收大量的数据，它必须将收到的数据流以帧为单位存到单独的内存空间，自行完成接收和发送操作，使 CPU 能以最少的操作就能读取并处理掉这些数据，释放出对应的内存空间，并保证 DMA 控制器和 CPU 不会就内存的使用权限起冲突。内存中被开辟用来暂存以太网帧的缓冲区单个大小一般设置为以太网帧的最大包长，IEEE802.3 规定为 1518 个字节（包括源止硬件地址、长度或类型域和 CRC32 校验域共 18 个字节）；以太网帧收发缓冲区的数量根据实际交互的频度和微控制器内存资源由用户自行确定。

以太网帧的缓冲区以类似队列的形式组织起来，在接收方向，以太网帧通过 MAC 被 DMA 写入内存而入队，被 CPU 读取而出队；在发送方向，以太网帧被 CPU 写入内存而入队，被 DMA 读取压入发送 FIFO 而出队。队列的深度即为缓冲区的个数。和普通的通讯外设不同，以太网帧的缓冲区的起始地址和数量并不是固化在某个寄存器中，而是由一类特殊的数据结构管理，这类数据结构存储在内存中，单个此类数据结构单元管理一个缓冲区，里面保存着缓冲区的起始地址、长度、DMA 控制器调用此缓冲区时需要进行的设置、DMA 发送完成回写的发送状态以及下一个此类数据结构的地址。这种数据结构起到传统通讯外设控制寄存器、状态寄存器的作用，但是实际位置又在内存中，所以可以称之为“软件寄存器”，正式名称为“DMA 描述符”。

DMA 描述符分发送和接收两种，每种的格式固定。DMA 描述符的存储结构分为两种，一种是链表形式，即每个描述符的第四个字为下一个描述符的地址，DMA 控制器会直接从 TDes3/RDes3 读取下一个描述符；另一种是环式结构，所有的描述符必须紧密排列，DMA 控制从当前描述符结束的位置取下一个描述符，当 DMA 控制器检测到 TDes4/RDes4 的 TER/RER 位置位时，即从描述符列表开头的位置取下一个描述符。描述符数组的地址必须 4 字节对齐，单个描述符的大小为 16 字节。

图 27-11 分别描述了环式结构和链式结构的一种缓冲区分配方案，供用户分配空间时参考。



27.1.6.2.1 发送 DMA 描述符

表 27-11 表明了发送描述符的结构。

表 27-11 发送描述符的结构

	31							0
TDes0	OWM (31)	CTRL (30:26)	TTSE (25)	保留	控制 (23:20)	保留 (19:18)	TTSS (17)	状态 (16:0)
TDes1	保留 (31:29)	缓冲区 2 字节计数 (28:16)			保留 (15:13)	缓冲区 1 字节计数 (12:0)		
TDes2	缓冲区 1 地址、时间戳低位							
TDes3	缓冲区 2 地址、下一个描述符的地址、时间戳高位							

由上表可以看出发送描述符由四个 32 位字组成，分别是 TDes0、TDes1、TDes2 和 TDes3，其中 TDes9 用作控制和返回发送状，TDes1 用以指示发送长度，TDes2 用以表示发送缓冲区的位置或返回发送时间戳的低位，TDes3 用以表示用于第二个发送缓冲区的地址（TCH 未置位时）或下一描述符的地址（TCH 置位时），时间戳使能时，发送后返回 IEEE1588 时间戳高位。各 32 位字的描述如下。

表 27-12 TDes0 的各位定义

31	30	29	28	27	26	25	24	23:22	21	19	19:18	17	16
OWN	IC	LS	FS	DC	DP	TTE	Res	CIC	TER	TCH	Res	TSS	IHE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	JT	FF	IPE	LCA	NC	LCO	EC	VF	CC				ED	UF	DB

位	名称	描述
31	OWN	描述符归属位，此位指示此描述符被谁占用。 0：此描述符归 CPU 所有，CPU 可以修改此描述符的值； 1：此描述符归 DMA 所有，CPU 无权限修改此描述符。 此位为 0 时，在 CPU 完成对描述符和缓冲区的操作后，由 CPU 置 1；此位为 1 时，在 DMA 完成对描述符和缓冲区的操作后，由 DMA 自动写 0。以此完成用户软件和硬件对描述符和收发缓冲区的操作交接。
30	IC	发送完成中断使能位，置此位后，在发送完当前帧之后，发送中断位标志位（ETH_DMASR：TS）将被置位。
29	LS	末段指示位，此位被置位表示此描述符指示的缓冲区里包含帧的结束部分。
28	FS	首段指示位，此位被置位表示此描述符指示的缓冲区里包含帧的开头部分。
27	DC	禁止自动 CRC 计算位，置此位后，DMA 控制器不会计算以太网帧的 CRC32 检验值，也不会有值附加到帧末尾。此位只有在 FS 位被置位时有效。另外，DP 位的设置优先级高于此位。
26	DP	禁止自动填充位，置此位后，DMA 控制器不会为不足 64 字节的以太网帧添加自动填充。当此位为 0 时，DMA 会自动为不足 64 字节的以太网帧添加填充和 CRC 校验值，忽

		视 DC 是否被置位。
25	TTE	时间戳发送使能位，在 ETH_PTPTSCR：TSE 置位的前提下，置此位后 DMA 控制器将在当面描述符指示的以太网帧打开 IEEE1588 时间戳功能。该位仅在 FS 被置位时有效。
24	Reserved	未使用。
23：22	CIC	校验和和插入控制域。 00：禁止插入校验和； 01：仅使能 IP 报头校验和的计算和插入； 10：保留； 11：使能 IP 报头校验和和有效负载检验和的计算和插入，使能计算伪报头检验和；
21	TER	发送描述符结束标志设置位，用户对此位置位指示 DMA 控制器，当前发送描述符已经是发送描述符数组的最后一个描述符。DMA 控制器下次会读取发送描述符数组最前面的一个描述符。
20	TCH	下一描述符地址有效指示位。该位置位表示第二个地址是下一个描述符的地址而不是下一个缓冲区的地址。此位置位时，TBS2 域的值不起作用。该位只在 FS 位置位时有效。TER 位的优先级高于此位。
19：18	Reserved	未使用。
17	TSS	发送时间戳捕获状态位，DMA 控制器置此位表示，发送时间戳已经捕获到，并存放到 TDes2 和 TDes3 中。
16	IHE	IP 报头错误状态位。DMA 控制器会根据接收到的数据检查 IP 报头：对于 IPv4，DMA 控制器会检车报头长度域是否正确；对于 IPv6，DMA 控制器会检查报头是否为 40 个字节。另外，IP 协议类型必须和以太网帧中类型/长度域一致。
15	ES	错误汇总位。如果发送帧时遇到错误，DMA 控制器会置此位。在下列位有一位被置位时，ES 位就会被置位： UF[TDES0:1]数据下溢错误位、 IPE[TDES0:12]IP 数据错误位、 FF[TDES0:13]帧清空位、 JT[TDES0:14]啰嗦超时位、 IHE[TDES0:16]IP 报头错误位。
14	JT	啰嗦超时位，此位被置位时表示 MAC 发送端发生了啰嗦超时错误。该位只有在 JD 位(ETH_MACCR:22)没有被置位时才会被置位。
13	FF	帧清空位，此位被置位表示由于 CPU 发出命令，DMA 控制器把帧从 FIFO 清空；
12	IPE	IP 包头错误指示位。MAC 会把接收到 TCP/UDP/ICMP 包的 IPv4 或 IPv6 包头中的包总长度和实际包长度做对比，若不一致就会置此位。
11	LCA	载波丢失指示位。此位置位表示帧在发送时发生了载波丢失，即 CSR 信号存在无效状态。该位只在半工模式下起作用。

10	NC	无载波指示位。该位表示帧在发送时物理层的载波侦听信号无效。该位只在半工模式下起作用。
9	LCO	迟到冲突指示位。该位表示帧在发送完前导符之后出现冲突。该位只在半工模式下起作用。
8	EC	冲突过多指示位。该位表示帧发送时出现了 16 位以上的冲突。如果 MACCR 的 RD 位置位，该位表示只发送了一次冲突。该位只在半工模式下起作用。
7	VF	VLAN 位，在发送 VLAN 帧时，此位会被置位。
6：3	CC	冲突计数器域。此域表示帧在发送时发生了多少次冲突。EC 置位时无效。该域只在半工模式下起作用。
2	EC	顺延过多指示位。此位置位表示在 MACCR 的 DC 置位时，发送帧因为顺延超过 24288 位而导致发送失败。该位只在半工模式下起作用。
1	UF	UF 数据下溢错误位。当 DMA 控制器发送时从指定的 RAM 取数据发现缓冲区为空时，发送进入暂停状态，并置该位和 DMASR 寄存器的相关位。
0	DB	顺延指示位。此位置位表示发送过程由于载波占用而失败。该位只在半工模式下起作用。

表 27-13 TDes1 的各位定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			TBS2												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TBS1												

位	名称	描述
31:29	Reserved	未使用。
28:16	TBS2	发送缓冲区 2 的大小。
15:13	Reserved	未使用。
12:0	TBS1	发送缓冲区 1 的大小。

表 27-14 TDes2 的各位定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TBAD1/TTSL															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBAD1/TTSL															

位	名称	描述
31:0	TBAD1/TTSL	TDes2 全部 32 位作为一整个单元用来存放发送缓冲区 1 的地址。在启用 IEEE1588 模式中，在完成发送之后，TDes2 也用来存放 MAC 返回的时间戳，同时 MAC 会把 OWN 位

		(TDes0;31)清零。此域存放时间戳的低 32 位。

表 27-15 TDes3 的各位定义

31	30	29	28	27	26	25	24	23	22	21	20	19	20	17	16
TDAD2/TTSH															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDAD2/TTSH															

位	名称	描述
31:0	TDAD2/TTSH	此域用来存放下一个描述符的地址，在 TCH 未置位时，此域存放发送缓冲区 2 的地址。在启用 IEEE1588 模式中，在完成发送之后，TDes32 用来存放 MAC 返回的时间戳，同时 MAC 会把 OWN 位(TDes0;31)清零。此域存放时间戳的高 32 位。

27.1.6.2.2 接收 DMA 描述符

表 27-16 接收描述符的结构

31 0

RDes0	OWM (31)	状态 (30:0)			
RDes1	控制 (31)	保留 (30:29)	缓冲区 2 字节计数 (28:16)	RER (15:14)	保留 13	缓冲区 1 字节计数 (12:0)
RDes2	缓冲区 1 地址、时间戳低位				
RDes3	缓冲区 2 地址、下一个描述符的地址、时间戳高位				

由上图可以看出接收描述符也是由四个 32 位字组成的，其中第一个 32 位字主要是返回接收时的状态，第二个 32 位字包含了接收的数据长度，第三个 32 位字定义了接收缓冲区的地址或者返回时间戳的低位，第四个 32 位字做第二个缓冲区的地址（RCH 未置位），下一缓冲区的地址（RCH 置位）或返回时间戳的高位。

接收描述符各字的各位意义如下：

表 27-17 RDes0 的各位定义

31	30	29：16
OWN	AFM	FL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	DE	SAF	LE	OE	VLAN	FS	LS	IPHCE	LCO	PT	RWT	RE	DE	CE	PCE

位	名称	描述
31	OWN	描述符归属位，此位指示此描述符被谁占用。 0：此描述符归 CPU 所有，CPU 可以修改此描述符的值；

		1：此描述符归 DMA 所有，CPU 无权限修改此描述符。 此位为 0 时，在 CPU 完成对描述符和缓冲区的操作后，由 CPU 置 1；此位为 1 时，在 DMA 完成对描述符和缓冲区的操作后，由 DMA 自动写 0。以此完成用户软件和硬件对描述符和收发缓冲区的操作交接。
30	AFM	目标地址未通过标志位。如果接收的帧未通过 MAC 的目标地址过滤器，那么此标志位会被置位。
29:16	FL	帧长域。此域在 ES 位（RDS0;14）为 0 时有效。在 LS 位（RDS;8）为 1 时，此域指示了 DMA 控制器接收到的帧的长度，包括 CRC；在 LS 位为 0 时，此域表示目前为止 DMA 控制器发往内存的累计长度，单位都是字节。
15	ES	错误汇总位。当 MAC 检测到以下任一错误时，会置位此位： CE[RDES:1]:CRC 错误； RE[RDES;3]:接收错误； RWT[RDES;4]:看门狗超时； IPHCE[RDES;7]:巨型帧（注意，报 IPHCE 时需要分辨到底是巨形帧还是由 IP 头错误）； OE[RDES;11]:溢出错误； DE[RDES;14]:描述符错误。
14	DE	描述符错误位。该位被置位表示由于描述符指示的缓冲区装不上当前帧而被切断，DMA 又不占用下一描述符。该位只在 LS 位被置位时(RDES0[8])才有效。
13	SAF	源地址过滤未通过标志位，此位被置位表示帧没有通过 MAC 的源地址过滤器。
12	LE	长度错误位，此位被置位表示实际收到的帧长度和以太网类型/长度域中指示的长度不符合。此位只在 FT（RDES0[5]）被置位时有效。
11	OE	溢出错误位，此位被置位表示由于接收 FIFO 溢出，接收到的帧被破坏。
10	VLAN	VLAN 标签位，该位被置位表示接收到的是 VLAN 帧。
9	FS	首描述符指示位，该位置位表示这个描述符包含帧的开头。
8	LS	尾描述符指示位，该位置位表示这个描述符包含帧的结尾。
7	IPHCE	IP 报头校验和错误标志位。该位置位表示 IPv4 或者 IPv6 报头存在错误，具体原因可能是： 1，以太网帧类型/长度域指示的协议与实际的 IP 版本不一致； 2，IP 报头校验和不对； 3，IP 报头指示的长度不对。
6	LCO	迟到冲突指示位。该位置位表示产生了迟到冲突。该位只在半双工模式下起作用。
5	FT	帧类型指示位，此位为 1 时表示接收到的帧长度/类型域的值大于 0x0600，表示上层协议；该位为 1 时表示接收到的帧长度/类型域的值小于 0x0600，表示长度。当长度

		/类型域小于 14 字节时，该位无效。
4	RWT	接收看门狗超时标志位。该位置位表示在接收当前帧时，看门狗超时，当前帧被截断。
3	RE	接收错误标志位。该位置位表示在接收帧的过程中，RX_DV 有效时，RX_ERR 信号有效。
2	DE	Dribble 比特错误位。该位置位表示 MAC 接收到的帧的长度不是 8bit 的帧数倍，可能有漏掉周期的现象。
1	CE	CRC 错误。该位置位表示接收到的帧存在 CRC 校验错误。该位只在 LS 位（RDES0[8]）置位是时有效。
0	PCE	负载校验和错误。该位置位表示 MAC 接收到的 TCP/UDP/ICMP 包与其校验和域标示的值不符。

可以看出 MAC 的接收流程中做了校验检验机制。实际上在以太网帧层（数据链路层）、网络层（IPv4/IPv6）和运输层（TCP/UDP/SCTP）都有对本层数据长度进行说明，对数据内容正确性采取某种手段的校验。RDES0 的第 0、5 和 7 位都提到了对数据的校验检验，下表进行归纳。

表 27-18 RDES:7/5/0 的值和接收到的帧状态的关系

RDES0;7	RDES0;5	RDES0;0	帧状态
IPHCE	FT	PCE	
1	1	1	为 IP 帧，IP 层和运输层存在校验错误。
1	1	0	为 IP 帧，IP 层存在校验错误。
1	0	1	为 IP 帧，运输层存在校验错误。
1	0	0	为 IP 帧，未检测到校验和错误。
0	1	1	非 IP 帧，未执行检验检测。
0	1	0	保留。
0	0	1	为 IP 帧，不支持的运输层协议，未检测到 IP 层校验和错误。
0	0	0	长度/类型域小于 0x0600 的帧。

表 27-19 RDes1 的各位定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIC	Reserved		RBS2												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RER	RCH	Res	RBS1												

位	名称	描述
31	DIC	关闭接收完成中断设置位。
30:29	Reserved	未使用。
28:16	RBS2	接收缓冲区 2 大小。
15	RER	末端接收描述符标志，指示当前描述符是最后一个描述符。DMA 控制器会回到描述符对列基地址寄存器（ETH_DMARDLAR）去取下一个描述符。
14	RCH	下一接收描述符地址有效位。该位置位表示最后一个 32 位字中是下一个接收描述符的地址，否则则是第 2 个缓冲

		区的地址。
13	Reserved	未使用。
12:0	RBS1	接收缓冲区 1 大小。

表 27-20 RDes2 的各位定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RBAD1/RTSL															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBAD1/RTSL															

位	名称	描述
31:0	RBAD/RTSL	RDes2 全部 32 位作为一整个单元用来存放接收缓冲区的地址。在启用 IEEE1588 模式中，在完成接收之后，RDes2 也用来存放 MAC 返回的时间戳，同时 MAC 会把 OWN 位(RDes0;31)清零。

表 27-21 RDes3 的各位定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDAD2/RTSH															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDAD2/RTSH															

位	名称	描述
31:0	RDAD/RTSH	RCR 置位时用来存放下一个缓冲区的地址，否则用来存在第二个缓冲区的地址。在启用 IEEE1588 模式中，在完成发送之后，RDes3 用来存放 MAC 返回的时间戳，同时 MAC 会把 OWN 位(RDes0:31)清零。此域存放时间戳的高 32 位。

27.1.6.3 数据缓存对齐

由于收发缓冲区和收发描述符都是由 DMA 控制器调用，并实质存在于 RAM 空间的，而 DMA 是 32 位的，因此设置收发描述符队列都需要保证它们的起始地址在 4 字节对齐的位置，但是设置缓冲区没有这个强制要求。

为了提高效率，一个以太网帧由一个缓冲区接收完毕是最恰当的，如将缓冲区设为 1518 个字节可以容纳下最大的以太网普通帧，包括源止地址域、长度类型域、数据填充域和 CRC 校验域。需要注意的是，带标签的 VLAN 帧的最大帧长比一般的以太网帧长 4 个字节，即 1522 个字节，如果用户想做到 4 字节对齐，一个缓冲区可以为 1524 字节。

27.1.6.4 DMA 收发配置

接收和发送的数据由 DMA 负责自动转存和推送，但是如果遇到致命错误，DMA 会停止运转，并更新 DMA 状态寄存器。用户用需要初始化 DMA 后手动启动 DMA 才能继续运转。

27.1.6.4.1 发送 DMA 配置

DMA 控制器建立运转机制的步骤如下：

1） 设置发送缓冲区队列，将要发送的内容填入发送缓冲区。设置发送描述符队列，填好发送描述符的各域各位，并将 OWN 置位，将发送描述符的管理权交付给 DMA 控制器；将描述符初始地址注册到 DMATDLAR 寄存器中；

2） 置位 ST 位（DMAOMR:13），开启 DMA；

3） 在运行模式下，DMA 描述符会自动读取发送描述符的内容，根据其指示的地址和长度把数据推送到发送 FIFO。结束后 DMA 会按照链式结构读取紧接着的下一个发送描述符进行下一次发送。当 DMA 控制器检测到 OWM 位未置位导致无权访问发送描述符或者其他正常的错误，就会终止传输，并将 TBUS 位（DMASR:2）或其他位（非 OWN 为 0 引起的错误）和 NIS 位（DMASR:16）置位。

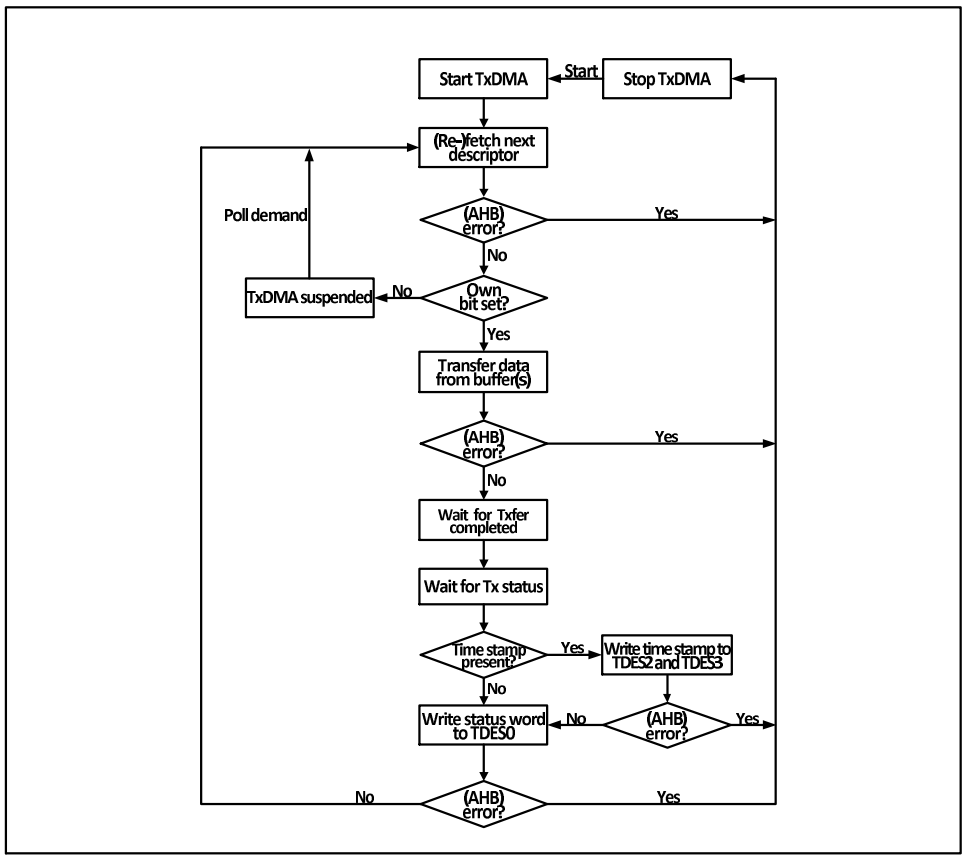
4） 不允许单个帧跨越多个描述符，一帧必须由一个描述符和一个缓冲区关联清楚。

5） 如果 MAC 开启了 IEEE1588 PTP 模式，那么在 DMA 控制器把数据推送到 FIFO 之后，MAC 会把发送时间戳写到 TDes2 和 TDes3 中，并复位 OWN 位。

6） 在发送完一个帧之后，如果发送描述符使能了发送完成中断（置位 TDes1:31），DMA 控制器就会置位发送完成中断标志位（DMASR:0），然后继续取下一个发送描述符。

下图展示了默认发送流程。

图 27-11 发送流程



27.1.6.4.2 接收 DMA 配置

建立 DMA 接收流转机制的步骤如下：

1） 设置发送缓冲区队列和描述符队列，设置好接收描述符的各个域各个位；将描述符初始地址注册到 DMARDLAR 寄存器中；置位 OWN 位，将描述符使用权限交给 DMA 控制器；

2） 置位 SR 位，开始接收流程；

3） 在接收流转机制运行时，DMA 控制器获取下一个描述符，检查接收描述符配置，在 FIFO 接收到下一

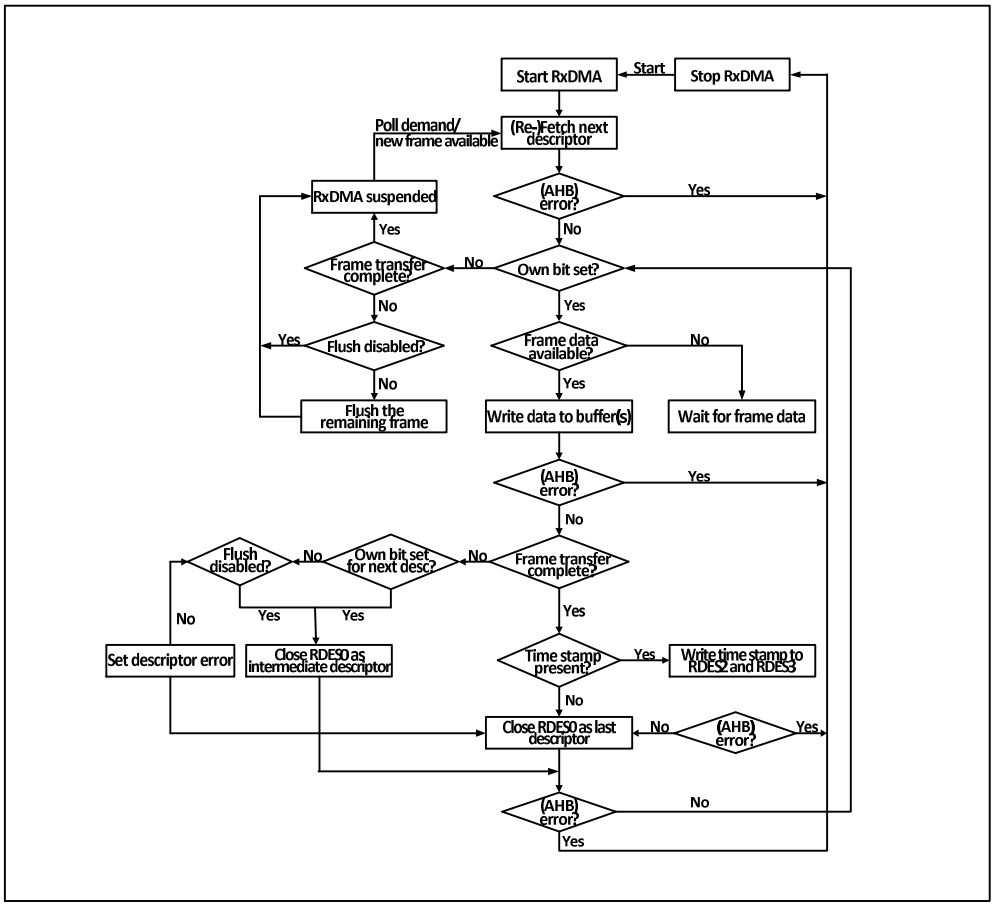
帧时，对帧内容进行过滤和识别等多项检查，将帧数据转发描述符指定的缓冲区中，写描述符状态域并获取下一个接收描述符，报接收完成中断。如果帧未通过过滤会被标记或丢弃，如果帧存在检验和错误、CRC 错误或帧过短将会被标记，如果帧过长可能会被掐断或报接收看门狗超时错误。DMA 控制器遇到接收描述符不可用等致命错误将会停止接收流程，用户需要特别留意；

4） 用户至少需要使能一个接收完成中断，在以太网中断函数中将使用过的接收描述符恢复到待命状态，以保证接收流程可以不间断地运行下去。用户可以在中断函数中将待处理的数据缓冲区地址传递出来，或处理一些打断接收流程的异常事件。

5） 如果使能了 PTP 时间戳，再 DMA 控制器转存数据，写描述符状态域时，同时也会将当前的时间戳写进描述符的后两个字。用户应及时将时间戳读出并补全原写在后两个字的缓冲区地址和下一描述符地址。

下图展示了默认接收流转机制：

图 27-12 接收流程



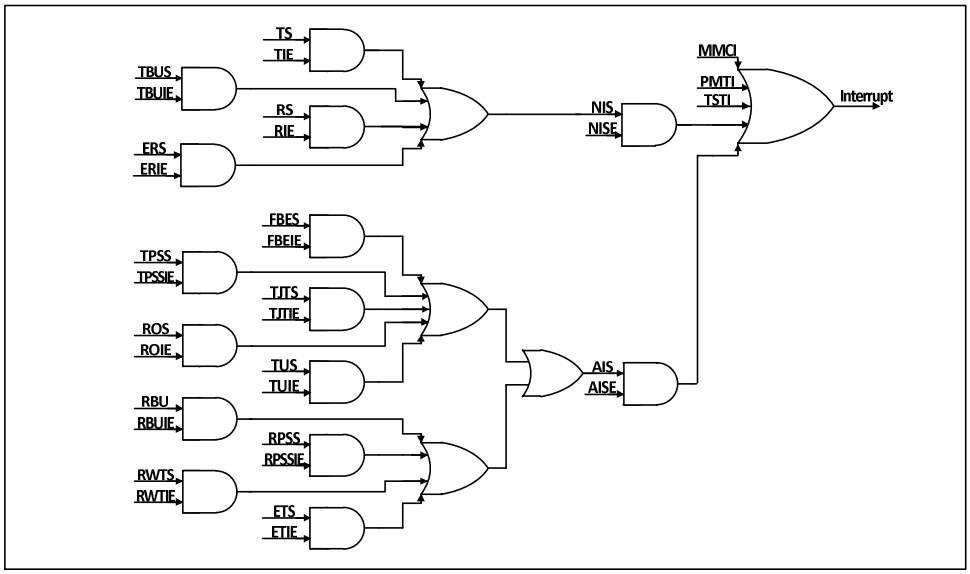
27.1.7 中断

以太网收发器拥有两个中断向量，一个是以太网唤醒事件，另一个是普通的收发事件。当检测到唤醒帧或魔法帧的时候，会触发以太网唤醒事件。普通的以太网收发中断事件包括 DMA 中断和 ETH 中断。

27.1.7.1 DMA 中断

DMA 中断大致可以分为两组，即正常的中断（NIS）和异常的中断（AIS），异常的中断一般意味着数据收发的异常，需要特别注意。用户在处理中断时需要检索所有的中断标志位，并将已经产生的中断源全部处理掉。下图是以太网收发器的中断示意图。

图 27-13 中断示意图



DMA 中断是以太网收发器中最重要的中断，一般的用户逻辑中都需要依靠中断接收帧，确认帧发送出去，或是及时处理被打断的收发逻辑。

27.1.7.2 ETH 中断

ETH 中断主要包括 PTP 的时间闹钟触发，收发计数到了 MMC 寄存器的某种设定值，以及 PMT。ETH 的用处主要是功能性的，相对 DMA 中断而言并不十分复杂和重要。用户可以使用 PTP 中断实现闹钟，使用 MMC 中断实现测速，或使用 PMT 中断实现远程唤醒。另外，ETH 还支持当内置物理层连接状态发生改变时产生中断。

27.1.7.3 PMT 中断

将 PMT 中断单独提出来重述一遍的原因是这个中断拥有独立的中断向量，使用时需注意。

27.1.8 寄存器描述

表 27-22 以太网收发器相关寄存器列表

MAC 控制相关寄存器地址映射

名称	偏移地址	描述	复位值
R32_ETH_MACCR	0x40028000	MAC 控制寄存器	0x00000000
R32_ETH_MACFFR	0x40028004	帧过滤寄存器	0x00000000
R32_ETH_MACHTHR	0x40028008	哈希值列表寄存器高位	0x00000000
R32_ETH_MACHTLR	0x4002800C	哈希值列表寄存器低位	0x00000000
R32_ETH_MACMIIAR	0x40028010	MII 地址寄存器	0x00000000
R32_ETH_MACMIIDR	0x40028014	MII 数据寄存器	0x00000000
R32_ETH_MACFCR	0x40028018	MAC 流控寄存器	0x00000000
R32_ETH_MACVLAN	0x4002801C	VLAN 标签寄存器	0x00000000
R32_ETH_MACRWUFFR	0x40028028	唤醒帧过滤器寄存器	0x00000000
R32_ETH_MACPMTCSR	0x4002802C	PMT 控制和状态寄存器	0x00000000
R32_ETH_MACSR	0x40028038	MAC 中断状态寄存器	0x00000000
R32_ETH_MACIMR	0x4002803C	MAC 中断屏蔽寄存器	0x00000000
R32_ETH_MACAOHR	0x40028040	MAC 地址 0 寄存器高位	0x8000FFFF

R32_ETH_MACA0LR	0x40028044	MAC 地址 0 寄存器低位	0xFFFFFFFF
R32_ETH_MACA1HR	0x40028048	MAC 地址 1 寄存器高位	0x0000FFFF
R32_ETH_MACA1LR	0x4002804C	MAC 地址 1 寄存器低位	0xFFFFFFFF
R32_ETH_MACA2HR	0x40028050	MAC 地址 2 寄存器高位	0x0000FFFF
R32_ETH_MACA2LR	0x40028054	MAC 地址 2 寄存器低位	0xFFFFFFFF
R32_ETH_MACA3HR	0x40028058	MAC 地址 3 寄存器高位	0x0000FFFF
R32_ETH_MACA3LR	0x4002805C	MAC 地址 3 寄存器低位	0xFFFFFFFF

MMC 控制相关寄存器地址映射，注意：地址非连续

名称	偏移地址	描述	复位值
R32_ETH_MMCCR	0x40028100	MMC 控制寄存器	0x00000000
R32_ETH_MMCRIR	0x40028104	MMC 接收寄存器	0x00000000
R32_ETH_MMCTIR	0x40028108	MMC 发送中断寄存器	0x00000000
R32_ETH_MMCRIMR	0x4002810C	MMC 接收中断屏蔽寄存器	0x00000000
R32_ETH_MMCTIMR	0x40028110	MMC 发送中断屏蔽寄存器	0x00000000
R32_ETH_MMCTGFSCCR	0x4002814C	MMC 一次冲突发送好帧计数器	0x00000000
R32_ETH_MMCTGFMSCCR	0x40028150	MMC 多次冲突发送好帧计数器	0x00000000
R32_ETH_MMCTGFCR	0x40028168	MMC 发送好帧计数寄存器	0x00000000
R32_ETH_MMCRFCECR	0x40028194	MMC 存在 CRC 检验错误的帧接收计数寄存器	0x00000000
R32_ETH_MMCRFAECR	0x40028198	接收对齐错误帧计数器	0x00000000
R32_ETH_MMCRGUFCR	0x400281C4	MMC 接收好单播帧计数寄存器	0x00000000

IEEE1588（PTP）相关寄存器地址映射

名称	偏移地址	描述	复位值
R32_ETH_PTPTSCR	0x40028700	PTP 时间戳控制寄存器	0x00000000
R32_ETH_PTPSSIR	0x40028704	PTP 亚秒递增寄存器	0x00000000
R32_ETH_PTPTSHR	0x40028708	PTP 时间戳寄存器高位	0x00000000
R32_ETH_PTPTSLR	0x4002870C	PTP 时间戳寄存器低位	0x00000000
R32_ETH_PTPTSHUR	0x40028710	PTP 时间戳更新寄存器高位	0x00000000
R32_ETH_PTPTSLUR	0x40028714	PTP 时间戳更新寄存器低位	0x00000000
R32_ETH_PTPTSAR	0x40028718	PTP 时间戳加数寄存器	0x00000000
R32_ETH_PTPTTHR	0x4002871C	PTP 目标寄存器高位	0x00000000
R32_ETH_PTPTTLR	0x40028720	PTP 目标寄存器低位	0x00000000

DMA 相关寄存器地址映射注意：地址非连续

名称	偏移地址	描述	复位值
R32_ETH_DMABMR	0x40029000	DMA 总线模式寄存器	0x00002101
R32_ETH_DMATPDR	0x40029004	DMA 发送查询寄存器	0x00000000
R32_ETH_DMARPTR	0x40029008	DMA 接收查询寄存器	0x00000000
R32_ETH_DMARDLAR	0x4002900C	DMA 接收描述符地址寄存器	0x00000000
R32_ETH_DMATDLAR	0x40029010	DMA 发送描述符地址寄存器	0x00000000
R32_ETH_DMASR	0x40029014	DMA 状态寄存器	0x00000000

R32_ETH_DMAOMR	0x40029018	DMA 操作模式寄存器	0x00000000
R32_ETH_DMAIER	0x4002901C	DMA 中断使能寄存器	0x00000000
R32_ETH_DMAMFBOCR	0x40029020	DMA 丢失帧寄存器	0x00000000
R32_ETH_DMACHTDR	0x40029048	DMA 当前发送描述符寄存器	0x00000000
R32_ETH_DMACHRDR	0x4002904C	DMA 当前接收描述符寄存器	0x00000000
R32_ETH_DMACHTBAR	0x40029050	DMA 当前发送缓存寄存器	0x00000000
R32_ETH_DMACHRBAR	0x40029054	DMA 当前接收缓存寄存器	0x00000000

内部 10M 物理层相关寄存器地址

名称	偏移地址	描述	复位值
BMCR	0x00	基本控制寄存器	0x2100
BMSR	0x01	基本状态寄存器	0x1809
PHY_SR	0x10	物理层状态寄存器	0x0000
PHY_MDIX	0x1E	自动翻转寄存器	0x0000

注；内部物理层寄存器的偏移地址在 SMI 接口中使用

27.1.8.1 MAC 控制相关寄存器各位域

27.1.8.1.1 MAC 控制寄存器（R32_ETH_MACCR）

偏移地址：0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCD			Reserved					WD	JD	PI	PR	IFG			CSD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FES		ROD	LM	DM	IPCO	RD	Res	APCS	BL		DC	TE	RE	TCF	Res

位	名称	访问	描述	复位值
[31:29]	TCD	RW	发送时钟延迟域，此域用来延迟发送时钟。MAC 控制器把经过 CES 位（ETH_MACCR[1]）选择的发送时钟进行延时输出。延时时间计算公式： Tdalay=TCD（ETH_MACCR[31:29]）*0.5ns；	000b
[28:24]	Reserved	RO	未使用。	
23	WD	RW	看门狗设置位。 0：MAC 打开看门狗，只能接受最大 2048 字节的以太网帧，超长部分会被切断； 1：MAC 关闭开门狗，能接收最大 16384 个字节的以太网帧；	0b
22	JD	RW	Jabber 设置位。 0：如果用户试图发送超过 2048 字节以上长度的以太网帧，MAC 会关闭发送器； 1：MAC 关闭 Jabber 定时器，最多能发送 16384 字节的以太网帧；	0b
21	PI	RW	内置 10MPHY 发送驱动偏置电流设置位。 0：额定驱动；	0b

			1：节能发送；	
20	PR	RW	内置 10MPHY 片内 50 欧电阻上拉开启设置位。 0：片内 50 欧电阻断开； 1：片内 50 欧电阻连接；	0b
[19:17]	IFG	RW	帧间隙设置域，这里设置了发送两个帧之间的最短时间间隙。 000：96 位时间； 001：88 位时间； 010：80 位时间； 011：72 位时间； 100：64 位时间； 101：56 位时间； 110：48 位时间； 111：40 位时间。	000b
16	Reserved	RO	未使用。	0b
[15:14]	FES	RW	以太网速度设置域。 00：10Mbit/s； 01：100Mbit/s； 10：1Gbit/s； 11：保留，未使用。	00b
13	Reserved	RO	未使用。	0b
12	LM	RW	自循环模式使能位。置该位使能自循环模式。	0b
11	DM	RW	双工模式使能位。置该位使能全双工模式。	0b
10	IPCO	RW	IPv4 校验检验使能位。 0:关闭接收端 IPv4 的校验和检验功能,相应的 PCE、PHCE 标志位总是为 0。见接收描述符各位定义； 1:使能 IPv4 的校验检验,MAC 控制器会对 TCP、UDP 和 ICMP 报头进行校验和检验。	0b
[9:8]	Reserved	RO	未使用。	
7	APCS	RW	填充&CRC 自动剥离使能位。 0：MAC 不改变帧的内容； 1：在接收长度小于等于 1500 字节的以太网帧时，MAC 自动去除帧的填充字节和 CRC 域；在大于 1500 字节的以太网帧中，MAC 不做更改。	0b
[6:4]	Reserved	RO	未使用。	
3	TE	RW	发送使能位。 0：MAC 在发送完当前帧之后，关闭发送器，不再发送任何帧； 1：使能 MAC 发送器。	0b
2	RE	RW	接收使能位。 0：MAC 在接收完当前帧之后，关闭接收器，不再接收任何帧； 1：使能 MAC 接收器。	0b
1	TCF	RW	发送时钟翻转设置位。 0:将 TCES 选择的 TXC 直接作为芯片输出的 GTX_CLK；	0b

			1：将 TCES 选择的 TXC 的反相作为芯片输出的 GTX_CLK。	
0	Res	RO	未使用。	0b

27.1.8.1.2 MAC 帧过滤寄存器（R32_ETH_MACFFR）

偏移地址：0x04

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					HPF	SAF	SAIF	PCF		BFD	PAM	DAIF	HM	HU	PM

位	名称	访问	描述	复位值
31	RA	RW	接收全部位。 0：MAC 只把通过了过滤器的帧转发到接收队列中； 1：MAC 把所有收到的帧转发到接收队列中，不管其是否通过了过滤器。	0b
[30:11]	Reserved	RO	未使用。	
10	HPF	RW	HASH 过滤或者完美过滤选择位。 0：在 HM 或者 HU 置位的前提下，只要符合 HASH 过滤器，就能通过地址过滤； 1：根据 HM 或者 HU 的值来确定单播/多播模式下使用什么过滤方式。	0b
9	SAF	RW	源地址过滤选择位。 0：MAC 对未通过源 MAC 地址过滤的帧进行标记； 1：MAC 会直接丢弃未通过源 MAC 地址过滤的帧；	0b
8	SAIF	RW	源地址过滤结果颠倒位。 0:接收的帧的源地址如果和 MAC 地址寄存器中启用的源地址不一致则认为是未通过源地址过滤器； 1:接收的帧的源地址如果和 MAC 地址寄存器中启用的源地址一致则认为是未通过源地址过滤器；	0b
[7:6]	PCF	RW	流控帧通过控制域。 00/01：MAC 不转发任何流控帧给应用程序； 10：MAC 转发所有流控帧到应用程序中，包括未通过地址过滤器的流控帧； 11：MAC 只转发通过地址过滤器的流控帧；	0
5	BFD	RW	广播帧接收控制位。 0：接收所有广播帧； 1：丢弃所有广播帧；	0b
4	PAM	RW	通过全部多播帧控制位。 0：多播帧能否通过过滤取决于 HM 的值； 1：全部的多播帧都能通过地址过滤；	0b
3	DAIF	RW	目的 MAC 地址过滤器过滤结果颠倒控制位。 0：过滤器结果正常生效；	0b

			1：对多播帧和单播帧，是否过滤器通过的结果进行颠倒再生效；	
2	HM	RW	多播帧过滤模式选择位。 0：进行完美地址过滤； 1：进行 HSAH 地址过滤；	0b
1	HU	RW	单播帧过滤模式选择位。 0：进行完美地址过滤； 1：进行 HSAH 地址过滤；	0b
0	PM	RW	混杂模式使能位。所有帧都能通过地址过滤器，且不标记过滤结果。	0b

27.1.8.1.3 哈希值列表寄存器（R32_ETH_MACHTHR、R32_ETH_MACHTLR）

偏移地址：0x08

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

偏移地址：0x0C

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

位	名称	访问	描述	复位值
[31:0]	HTH	RW	哈希列表高 32 位	00000000h
[31:0]	HTL	RW	哈希列表低 32 位	00000000h

27.1.8.1.4 MII 地址寄存器（R32_ETH_MACMIIAR）

偏移地址：0x10

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	未使用。	
[15:11]	PA	RW	物理层地址域。用户将需要操作的物理层地址写进此域。	0
[10:6]	MR	RW	物理层寄存器地址域。用户将需要操作的寄存器地址写进此域。	0

5	Reserved	RO	未使用。	
[4:2]	CR	RW	时钟范围设定域。用户请保持为 000b，使 MII 频率为主频的 42 分频。	0
1	MW	RW	读写设定位。 0：对物理层进行读操作； 1：对物理层进行写操作；	0b
0	MB	W1	MII 忙标志位。 该位由用户置位，表示命令硬件开始进行读或者写的操作，读写期间应保持物理地址，寄存器地址和数据域不变。在硬件将此位清零后，表示完成操作。	0b

27.1.8.1.5 MII 数据寄存器（R32_ETH_MACMIIDR）

偏移地址:0x14

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

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	未使用。	
[15:0]	MD	RW	MII 操作数据域。此域用来存放将要通过 MII 接口从物理层读取的数据，或者存放向物理层写入的数据。	0

27.1.8.1.6 MAC 流控寄存器（R32_ETH_MACFCR）

偏移地址:0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PT															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ZQPD	Res	PLT		UPFD	RFCE	TFCE	FCB

位	名称	访问	描述	复位值
[31:16]	PT	RW	Pause 间隔域。此域用来作为控制 Pause 时间域的值，单位为当前的 MII 接口发送 64 字节的耗时。	0
[15:8]	Reserved	RO	未使用。	
7	ZQPD	RW	关闭零值 Pause 功能控制位。此位被置位时，将关闭自动零值 Pause 控制帧的生成。	0b
6	Reserved	RO	未使用。	
[5:4]	PLT	RW	自动重发 Pause 帧阈值。这个域的值应该小于 PT 的值。重发时间等于 PT 减去 PLT 所代表的时间。 00：4 个时间间隙； 01：28 个时间间隙；	0

			10：144 个时间间隙； 11：256 个时间间隙；	
3	UPFD	RW	单播 Pause 帧检测位。 0:MAC 只接收带协议规范定义的唯一地址的 Pause 帧； 1：MAC 同时检测 Pause 帧是否 MAC 地址寄存器 0 中定义的单播地址；	0b
2	RFCE	RW	接收流控使能位。 0：MAC 不解析 Pause 帧。 1：MAC 解析 Pause 帧并关闭发送器一段时间。	0b
1	TFCE	RW	发送流控使能位。 0：MAC 关闭发送流控，不发送 Pause 帧； 1：MAC 使能发送流控，可以发送 Pause 帧；	0b
0	FCB	W1	流控忙标志位。对此位置位可以发送一个 Pause 帧，发送完成后由硬件清零。在对整个 MACFCR 寄存器进行操作时，需要保证 FCB 位为 0。	0b

27.1.8.1.7 VLAN 标签寄存器（MACVLAN）

偏移地址：0x1C

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

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	未使用。	
16	VLANT	WR	标签比较控制位。 0：将 VLAN 帧的第 15，16 字节的全部 16 位数据都与 VLANTI 域进行比较； 1：只使用 VLAN 帧的第 15，16 字节的[11:0]位数据都与 VLANTI 域的相应位进行比较；	0
[15:0]	VLANTI	WR	标签对比样本域。根据 IEEE 802.1 协议，VLAN 帧的第[15:13]位是用户优先级，[12]是规范格式指示符，位[11:0]VLAN 标识符域。 如果 VLANTI 全为 0，那么 MAC 将不再关心 VLAN 帧的第 15，16 字节，而当其第 13，14 字节为 0x8100（注意大小端）时即判定为 VLAN 帧。	0

27.1.8.1.8 唤醒帧过滤寄存器（R32_MACRWUFFR）

偏移地址：0x28

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

RWUFFR

位	名称	访问	描述	复位值
[31：0]	RWUFFR	RW	唤醒帧过滤寄存器实际上时八个不同的寄存器，对其进行连续八次的读操作可以读出全部寄存器，同样对其进行连续八次的写操作可以写进全部八个寄存器。这八个寄存器各个位的说明详见 14.5.6.3 节的描述。	0

27.1.8.1.9 PMT 控制和状态寄存器（R32_ETH_MACPMTCSR）

偏移地址：0x2C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						GU	Reserved		WFR	MPR	Reserved		WFE	MPE	PD

位	名称	访问	描述	复位值
31	WFFRP	W	寄存器复位控制位。置此位会将 PMTCSR 寄存器全部清零。此位会在一个系统周期自动清零。	0b
[30:10]	Reserved	RO	未使用。	
9	GU	RW	全局单播位。置此位会使 MAC 将所有通过过滤器的单播帧都认为是唤醒帧。	0b
[8:7]	Reserved	RO	未使用。	
6	WFR	RC	唤醒帧接收标志位。当收到唤醒帧时，此位会被置位。读取自动清零。	0b
5	MPR	RC	魔法帧接收标志位。当收到魔法帧时，此位会被置位。读取自动清零。	0b
[4:3]	Reserved	RO	未使用。	
2	WFE	RW	唤醒帧使能位。置此位允许在接收到唤醒帧时产生 PMT 事件。	0b
1	MPE	RW	魔法帧使能位。置此位允许在接收到魔法帧时产生 PMT 事件。	0b
0	PD	W	掉电控制位。置此位会使 MAC 进入掉电模式：丢弃其他所有帧，直到其收到了唤醒帧或魔法帧，唤醒后 MAC 自动清零。在置此位前需要将 WFE 或者 MPE 置位。	0b

27.1.8.1.10 MAC 中断状态寄存器（R32_ETH_MACSR）

偏移地址：0x38

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TSTS	Reserved		MMCTS	MMCRS	MMCS	PMTS	Reserved		

位	名称	访问	描述	复位值
[31:10]	Reserved	RO	未使用。	
9	TSTS	RC	时间戳触发中断标志位。当到达 PTP 系统时钟设定的时间时，此标志位会被置位。	
[8:7]	Reserved	RO	未使用。	
6	MMCTS	R	MMC 发送中断标志位。当 MMC 寄存器组中的 MMCTIR 寄存器产生任一中断时，该位置位。当 MMC 寄存器组中的 MMCTIR 寄存器全部清零时，此位也即清零。	
5	MMCRS	R	MMC 接收中断标志位。当 MMC 寄存器组中的 MMCRIR 寄存器产生任一中断时，该位置位。当 MMC 寄存器组中的 MMCRIR 寄存器全部清零时，此位也即清零。	
4	MMCS	R	MMC 状态标志位。当 MMCTS 或 MMCRS 置位时会触发此位置位;当 MMCTS、MMCRS 全部清零时此位即清零。	
3	PMTS	R	PMT 状态标志位。在掉电模式下，如果 MAC 收到魔法帧或者唤醒帧唤醒了 MAC，那么此位会被置位；清除 WFR 和 MPR 两位后此位被清零。	
[2:0]	Reserved	RO	未使用。	

27.1.8.1.11 MAC 中断屏蔽寄存器（R32_ETH_MACIMR）

偏移地址：0x3C

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

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

位	名称	访问	描述	复位值
[31:10]	Reserved	RO	未使用。	
9	TSTIM	RW	时间戳中断屏蔽位。置此位将禁止产生时间戳中断。	0
[8:4]	Reserved	RO	未使用。	
3	PMTIM	RW	PMT 中断屏蔽位。置此位将屏蔽 PMT 中断。	0
[2:0]	Reserved	RO	未使用。	

27.1.8.1.12 MAC 地址寄存器 0 高 32 位(R32_ETH_MACA0HR)

偏移地址：0x40

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

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

MACAOH

位	名称	访问	描述	复位值
31	MO	RO	总是为 1。	1b
[30:16]	Reserved	RO	未使用。	
[15:0]	MACAOH	RW	MAC 地址的高 16 位，即 47:32 位。	FFFFh

27.1.8.1.13 MAC 地址寄存器 0 低 32 位(R32_ETH_MACAOLR)

偏移地址：0x44

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

位	名称	访问	描述	复位值
[31:0]	MACAOL	RW	MAC 地址的低 32 位，即 31:0 位。一般情况下，MAC 地址寄存器 0 的地址是 MAC 本身的地址。	FFFFFFFFh

27.1.8.1.14 MAC 地址寄存器 1 高 32 位(R32_ETH_MACA1HR)

偏移地址：0x48

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	SA	MBC						Reserved							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA1H															

位	名称	访问	描述	复位值
31	AE	RW	地址过滤使能位。 0：地址过滤器使用 MAC 地址 1 来进行完美过滤； 1：地址过滤器忽略 MAC 地址 1；	0b
30	SA	RW	地址角色选择位。 0:MAC 地址 1 被用来和接收到的帧的目标地址对比； 1：MAC 地址 1 被用来和接收到的帧的源地址对比；	0b
[29:24]	MBC	RW	屏蔽字控制域。MBC 的各个位用来对应屏蔽 MAC 地址 1 的某个字节，用来禁止将 MAC 地址 1 的某个字节参与到接收帧的源地址或目标地址的比较中。 R32_ETH_MACA1HR[29]置位则屏蔽 MACA1H[15:8]； R32_ETH_MACA1HR[28]置位则屏蔽 MACA1H[7:0]； R32_ETH_MACA1HR[27]置位则屏蔽 MACA1L[31:24]； R32_ETH_MACA1HR[26]置位则屏蔽 MACA1L[23:16]； R32_ETH_MACA1HR[25]置位则屏蔽 MACA1L[15:8]； R32_ETH_MACA1HR[24]置位则屏蔽 MACA1L[7:0]；	000000b

[23:16]	Reserved	RO	未使用。	
[15:0]	MACA1H	RW	MAC 地址的高 16 位，即 47:32 位。	FFFFb

27.1.8.1.15 MAC 地址寄存器 1 低 32 位(R32_ETH_MACA1LR)

偏移地址：0x4C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACA1L															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA1L															

位	名称	访问	描述	复位值
[31:0]	MACA1L	RW	MAC 地址的低 32 位，即 31:0 位。	FFFFFFFFh

27.1.8.1.16 MAC 地址寄存器 2 高 32 位(R32_ETH_MACA2HR)

偏移地址：0x50

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	SA	MBC						Reserved							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA2H															

位	名称	访问	描述	复位值
31	AE	RW	地址过滤使能位。 0：地址过滤器使用 MAC 地址 2 来进行完美过滤； 1：地址过滤器忽略 MAC 地址 2；	0b
30	SA	RW	地址角色选择位。 0:MAC 地址 2 被用来和接收到的帧的目标地址对比； 1：MAC 地址 2 被用来和接收到的帧的源地址对比；	0b
[29:24]	MBC	RW	屏蔽字控制域。MBC 的各个位用来对应屏蔽 MAC 地址 2 的某个字节，用来禁止将 MAC 地址 2 的某个字节参与到接收帧的源地址或目标地址的比较中。 R32_ETH_MACA2HR[29]置位则屏蔽 MACA2H[15:8]； R32_ETH_MACA2HR[28]置位则屏蔽 MACA2H[7:0]； R32_ETH_MACA2HR[27]置位则屏蔽 MACA2L[31:24]； R32_ETH_MACA2HR[26]置位则屏蔽 MACA2L[23:16]； R32_ETH_MACA2HR[25]置位则屏蔽 MACA2L[15:8]； R32_ETH_MACA2HR[24]置位则屏蔽 MACA2L[7:0]；	000000b
[23:16]	Reserved	RO	未使用。	
[15:0]	MACA2H	RW	MAC 地址的高 16 位，即 47:32 位。	FFFFh

27.1.8.1.17 MAC 地址寄存器 2 低 32 位(R32_ETH_MACA2LR)

偏移地址：0x54

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACA2L															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA2L															

位	名称	访问	描述	复位值
31:0	MACA2L	RW	MAC 地址的低 32 位，即 31:0 位。	FFFFFFFFh

27.1.8.1.18 MAC 地址寄存器 3 高 32 位(R32_ETH_MACA3HR)

偏移地址：0x58

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA2H															

位	名称	访问	描述	复位值
31	AE	RW	地址过滤使能位。 0：地址过滤器使用 MAC 地址 3 来进行完美过滤； 1：地址过滤器忽略 MAC 地址 3；	0b
30	SA	RW	地址角色选择位。 0:MAC 地址 3 被用来和接收到的帧的目标地址对比； 1：MAC 地址 3 被用来和接收到的帧的源地址对比；	0b
[29:24]	MBC	RW	屏蔽字控制域。MBC 的各个位用来对应屏蔽 MAC 地址 3 的某个字节，用来禁止将 MAC 地址 3 的某个字节参与到接收帧的源地址或目标地址的比较中。 R32_ETH_MACA3HR[29]置位则屏蔽 MACA3H[15:8]； R32_ETH_MACA3HR[28]置位则屏蔽 MACA3H[7:0]； R32_ETH_MACA3HR[27]置位则屏蔽 MACA2L[31:24]； R32_ETH_MACA3HR[26]置位则屏蔽 MACA2L[23:16]； R32_ETH_MACA3HR[25]置位则屏蔽 MACA3L[15:8]； R32_ETH_MACA3HR[24]置位则屏蔽 MACA3L[7:0]；	000000b
[23:16]	Reserved	RO	未使用。	
15:0	MACA2H	RW	MAC 地址的高 16 位，即 47:32 位。	FFFFFFFFh

27.1.8.1.19 MAC 地址寄存器 3 低 32 位(R32_ETH_MACA3LR)

偏移地址：0x5C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACA3L															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA3L															

位	名称	访问	描述	复位值
[31:0]	MACA3L	RW	MAC 地址的低 32 位，即 31:0 位。	FFFFFFFFh

27.1.8.2 MMC 相关寄存器各位域

27.1.8.2.1 MMC 控制寄存器（R32_ETH_MMCCR）

偏移地址：0x0100

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												MCF	ROR	CSR	CR

位	名称	访问	描述	复位值
[31:4]	Reserved	RO	未使用。	
3	MCF	RW	计数器冻结控制位。置此位会将 MMC 所有的计数器值全部冻结。复此位即恢复各计数器计数。在冻结时置位 ROR 再读取任意计数器，会导致该计数器清零。	0b
2	ROR	RW	读时复位控制位。置此位会使读取任一计数器之后清零计数器的值。	0b
1	CSR	RW	计数器回转停止位。置此位会使计数器增正到最大值后停止而不自动归零。	0b
0	CR	W	计数器复位控制位。置此位将复位全部计数器。此位在一个系统周期后将自动清零。	0b

27.1.8.2.2 MMC 接收中断寄存器（R32_ETH_MMRIR）

偏移地址：0x0104

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

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

位	名称	访问	描述	复位值
[31:18]	Reserved	RO	未使用。	
17	RGUFS	RC	接收好帧数量过半时会置此位。	0b
[16:6]	Reserved	RO	未使用。	
5	RFCES	RC	接收 CRC 校验错误的帧数量过半时会置此位。	0b

4:0	Reserved	RO	未使用。	

27.1.8.2.3 MMC 发送中断寄存器（R32_ETH_MMCTIR）

偏移地址：0x0108

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

位	名称	访问	描述	复位值
[31:22]	Reserved	RO	未使用。	
21	TGFS	RC	发送帧数过半时此位会置位。	0b
[20:0]	Reserved	RO	未使用。	

27.1.8.2.4 MMC 接收中断屏蔽寄存器（R32_ETH_MMRIMR）

偏移地址：0x010C

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

位	名称	访问	描述	复位值
[31:18]	Reserved	RO	未使用。	
17	FGUFM	RW	接收好帧过半中断屏蔽位。置此位将屏蔽接收好帧计数器值达到一半时产生的中断。	0b
[16:6]	Reserved	RO	未使用。	
5	FRCRM	RW	接收 CRC 错误帧过半中断屏蔽位。置此位将屏蔽接收到 CRC 错误的帧计数器值达到一半时产生的中断。	0b
[4:0]	Reserved	RO	未使用。	

27.1.8.2.5 MMC 发送中断屏蔽寄存器（R32_ETH_MMCTIMR）

偏移地址：0x0110

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

位	名称	访问	描述	复位值

[31:22]	Reserved	RO	未使用。	
21	TGFM	RW	发送好帧过半中断屏蔽位。置此位将屏蔽发送好帧计数器值达到一半时产生的中断。	0b
[20:0]	Reserved	RO	未使用。	

27.1.8.2.6 MMC 一次冲突后发生好帧计数器（R32_ETH_MMCTGFSCCR）

偏移地址：0x014C

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

位	名称	访问	描述	复位值
[31:0]	TGFSCCR	RO	这个域用来统计半双工模式下，发生帧时只遇到一次冲突的计数器。	0

27.1.8.2.7 MMC 多次冲突后发生好帧计数器（R32_ETH_MMCTGFMSCCR）

偏移地址：0x0150

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

位	名称	访问	描述	复位值
[31:0]	TGFMSCCR	RO	这个域用来统计半双工模式下，发生帧时只遇到一次以上冲突的计数器。	0

27.1.8.2.8 MMC 发送好帧计数寄存器（R32_ETH_MMCTGFCR）

偏移地址：0x0168

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

位	名称	访问	描述	复位值
[31:0]	TGFC	RO	MAC 发送出去的正确帧的计数量。	0

27.1.8.2.9 MMC 接收 CRC 有误帧计数寄存器（R32_ETH_MMCRFCECR）

偏移地址：0x0194

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

位	名称	访问	描述	复位值
[31:0]	RFCECR	RO	MAC 接收到的存在 CRC 校验错误的帧的计数器。	0

27.1.8.2.10 MMC 接收对齐错误计数器寄存器（R32_ETH_MMCRFAECR）

偏移地址：0x0198

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

位	名称	访问	描述	复位值
[31:0]	RFAECR	RO	MAC 接收到的存在对齐错误的帧的计数器。	0

27.1.8.2.11 MMC 接收好帧计数寄存器（R32_ETH_MMCRGUFCR）

偏移地址：0x01C4

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

位	名称	访问	描述	复位值
[31:0]	RGUFCR	RO	MAC 接收到的正常的帧的数量。	0

27.1.8.3 IEEE 1588（PTP）相关寄存器各位域

27.1.8.3.1 PTP 时间戳控制寄存器（R32_ETH_PTPTSCR）

偏移地址：0x0700

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

位	名称	访问	描述	复位值

[31:6]	Reserved	RO	未使用。	
5	TSARU	RW	加数寄存器更新控制位。置此位后，加数寄存器的值将会加到累加器中。精调模式下使用此位。累加器自增完毕后此位自动清零。此位只能在为 0 时置位。	0b
4	TSITE	RW	时间戳中断触发使能位。置此位后，当 PTP 系统时间达到设定目标时间寄存器的值时，将产生中断。	0b
3	TSSTU	RW	系统时间更新控制位。置此位后，PTP 系统时间将加上更新寄存器中的值。更新完毕后此位自动清零。	0b
2	TSSTI	RW	时间戳初始化控制位。置此位后，PTP 的系统时间将会被替换成更新寄存器中的值。更新完毕后此位自动清零。	0b
1	TSFCU	RW	更新模式选择控制位。 0：表示使用粗调模式； 1：表示使用精调模式；	0b
0	TSE	RW	附加时间戳使能位。 0：不向描述符中添加时间戳； 1：发送或接收完成后，向描述符中添加时间戳；	0b

27.1.8.3.2 PTP 亚秒递增寄存器（R32_ETH_PTPSSIR）

偏移地址：0x0704

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

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

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	未使用。	
[7:0]	STSSI	RW	亚秒步进值。在粗调模式下，每个主频周期，PTP 系统时间会自增这个值；在精调模式下，在累加器溢出时，PTP 系统时间会自增这个值。	0

27.1.8.3.3 PTP 时间戳寄存器高位(R32_ETH_PTPTSHR)

偏移地址：0x0708

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

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

位	名称	访问	描述	复位值
[31:0]	TSHR	RO	PTP 系统时间值，实时值，单位为秒。	0

27.1.8.3.4 PTP 时间戳寄存器低位(R32_ETH_PTPTSLR)

偏移地址：0x070C

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

位	名称	访问	描述	复位值
31	STPNS	RO	系统时间正负标志位。1 表示当前时间为负，0 表示当前时间为正。	0
[30:0]	STSS	RO	PTP 系统时间值，实时值，单位为亚秒，即约 0.46 纳秒。STSS 溢出时，STS 自增 1 秒。	0

27.1.8.3.5 PTP 时间戳更新寄存器高 32 位 (R32_ETH_PTPTSHUR)

偏移地址：0x0710

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

位	名称	访问	描述	复位值
[31:0]	TSUS	RW	时间戳更新秒值，用来替换到 PTP 系统时间高位或者表示其在系统时间上加减的秒值。	0

27.1.8.3.6 PTP 时间戳更新寄存器高 32 位 (R32_ETH_PTPTSHUR)

偏移地址：0x0714

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

位	名称	访问	描述	复位值
31	TSUPNS	RW	时间戳更新正负标志位。当更新时间方式为使用 TSUS 和 TSUSS 直接替换 PTP 系统时间时，TSUPNS 应为 0,；当更新时间方式为使用 TSUS 和 TSUSS 作在原 PTP 系统时间值上的加减时，TSUPNS 为 1 表示在 PTP 系统时间的基础上减去 TSUS 和 TSUSS，为 0 表示在 PTP 系统时间的基础上加上 TSUS 和 TSUSS。	0

[30:0]	TSUSS	RW	时间戳更新亚秒值，用来替换到 PTP 系统时间高位或者表示其在系统时间上加减的亚秒值。	0

27.1.8.3.7 PTP 时间戳加数寄存器（R32_ETH_PTPTSAR）

偏移地址：0x0718

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

位	名称	访问	描述	复位值
[31:0]	TSA	RW	时间戳加数值。这个寄存器只在精调模式下才会被使用。TSA 的值会在每个系统周期被加到累加器中，如果累加器溢出，则会触发精调模式下的系统时间更新。	0

27.1.8.3.8 PTP 目标时间寄存器高 32 位（R32_ETH_PTPTTHR）

偏移地址：0x071C

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

位	名称	访问	描述	复位值
[31:0]	TTSH	RW	目标时间秒值。如果 PTP 系统时间达到或者超过这个值，且使能了相关中断，那么会产生一个中断。	0

27.1.8.3.9 PTP 目标时间寄存器高 32 位(R32_ETH_PTPTTHR)

偏移地址：0x0720

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

位	名称	访问	描述	复位值
[31:0]	TTSL	RW	目标时间亚秒值。如果 PTP 系统时间达到或超过这个值，且使能了相关中断，那么会产生一个中断。	0

27.1.8.4 DMA 控制相关寄存器各位域

27.1.8.4.1 DMA 发送查询寄存器（R32_ETH_DMATPDR）

偏移地址：0x1004

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

位	名称	访问	描述	复位值
[31:0]	TPDR	RW	发送查询命令。用户通过向这个寄存器写任意值来启动被暂停的发送流程。重启发送流程后，这个寄存器会被自动清零。	0

27.1.8.4.2 DMA 接收查询寄存器（R32_ETH_DMARPDR）

偏移地址：0x1008

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

位	名称	访问	描述	复位值
[31:0]	RPDR	RW	接收查询命令。接收流程可能会被各种意外而打断，需要用户向这个寄存器写任意值来重启接收流程。重启接收流程后，这个寄存器会被自动清零。	0

27.1.8.4.3 DMA 接收描述符地址寄存器（R32_ETH_DMARDLAR）

偏移地址：0x100C

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

位	名称	访问	描述	复位值
[31:0]	RDLAR	RW	这个寄存器用来存储第一个接收 DMA 描述符的地址。注意描述符需要 16 字节对齐，所以这个寄存器的后 4 位应该为 0。	0

27.1.8.4.4 DMA 发送描述符地址寄存器（R32_ETH_DMATDLAR）

偏移地址：0x1010

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

位	名称	访问	描述	复位值
[31:0]	TDLAR	RW	这个寄存器用来存储第一个发送 DMA 描述符的地址。注意描述符需要 16 字节对齐，所以这个寄存器的后 4 位应该为 0。	0

27.1.8.4.5 DMA 状态寄存器（R32_ETH_DMASR）

偏移地址：0x1014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLS	Res	TSTS	PMTS	MMCS	Res	EBS			TPS			RPS			NIS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIS	ERS	FBES	Reserved		ETS	RWTS	RPSS	RBUS	RS	TUS	ROS	TJTS	TBUS	TPSS	TS

位	名称	访问	描述	复位值
31	PLS	RC	内部 10M 物理层连接状态改变标志位。此位置位表示物理层连接上或者已断开。或写 1 清零。	0b
30	Reserved	RO	未使用。	
29	TSTS	RO	时间戳触发状态标志位。在 PTP 部分产生中断事件时，此位会被置位，如果使能了 PTP 中断则会产生中断。清除 PTP 部分的全部标志位后，此位会自动清除。	0b
28	PMTS	RO	PMT 触发状态标志位。在 PMT 部分产生中断事件时，此位会被置位，如果使能了 PMT 中断则会产生中断。清除 PMT 部分的全部标志位后，此位会自动清除。	0b
27	MMCS	RO	MMC 触发状态标志位。在 MMC 部分产生中断事件时，此位会被置位，如果使能了 MMC 中断则会产生中断。清除 MMC 部分的全部标志位后，此位会自动清除。	0b
26	Reserved	RO	未使用。	0b
[25:23]	EBS	RO	错误状态域。该域表示造成总线错误的类型。此域只在 DMASR[13]位被置位的情况下才有效。 DMASR[23]:0:发送 DMA 转发数据时出错； 1：接收 DMA 转发数据时出错； DMASR[24]:0:读数据转发时出错； RO 1:写数据转发时出错； DMASR[25]:0:访问描述符时出错； 1:访问数据缓存时出错；	0

[22:20]	TPS	RO	发送流程状态域。这个域用来表示当前发送 DMA 的状态。 000：停止，接收到复位或者停止发送命令； 001：运行，正在取发送描述符； 010：运行，正在等待状态信息； 011：运行，正在读取发送缓冲区数据并压进 FIFO； 100,101：保留； 110：暂停，发送描述符不可用或者发送缓存数据下溢； 111：运行，正在关闭发送描述符；	0
[19:17]	RPS	RO	接收流程状态域。这个域用来表示当前接收 DMA 的状态。 000：停止，接收到复位或者停止发送命令； 001：运行，正在取接收描述符； 010：保留； 011：运行，正在等待接收数据包； 100：暂停，接收描述符不可用； 101：运行，正在关闭接收描述符； 110：保留； 111：运行，正在把接收数据从 FIFO 压入内存中；	0
16	NIS	RO	正常中断汇总位。在 DMAIER 寄存器中使能的中断下，如果下列位任一被置位，NIS 位同样会被置位。 -DMASR[0]:发送中断； -DMASR[2]:发送缓存不可用； -DMASR[6]:接收中断； -DMASR[14]:早接收中断； 将以上所有的位清零，则 NIS 位将自动清零；	0b
15	AIS	RO	异常中断汇总位。在 DMAIER 寄存器中使能的中断下，如果下列位任一被置位，AIS 位同样会被置位。 -DMASR[1]:发送流程停止； -DMASR[3]:发送啰嗦超时； -DMASR[4]:接收 FIFO 溢出； -DMASR[5]:发送数据下溢； -DMASR[7]:发送缓存不可用； -DMASR[8]:接收流程停止； -DMASR[9]:接收看门狗超时； -DMASR[10]:早发送； -DMASR[13]:总线错误； 将以上所有的位清零，则 AIS 位将自动清零；	0b
14	ERS	RO	早接收状态位。该位被置位时，表示收到数据帧时 DMA 已经填满了第一个缓冲区，但是完整的帧还没接收完毕。RS 置位后，ERS 位自动清零。	0b
16	FBES	RO	总线错误位。该位被置位时，表示发送了总线错误。具体原因见[25:23]位。该位被置位后，相应的 DMA 控制器将关闭总线访问。	0b
[12:11]	Reserved	RO	未使用。	

10	ETS	RC	早发送状态位。置位时表示发送帧已经全部压入 FIFO。	0b
9	RWTS	RC	看门狗超时标志位。置位时表示帧长已经超过了 2048 字节。	0b
8	RPSS	RC	接收流程停止状态位。该位置位表示接收流程已经停止。	0b
7	RBUS	RC	接收缓存不可用状态位。该位置位表示接收描述符权限归属 CPU，DMA 无法获得，接收流程已经暂停。用户需要释放描述符并向 RPDR 寄存器中填入一个值来恢复接收流程。DMA 会在接收下一帧时重试获取描述符。	0b
6	RS	RC	接收完成状态位。该位置位表示已经完成一帧的接收，帧信息也更新到描述符中。	0b
5	TUS	RC	发送数据下溢位。该位置位表示发送缓存在发送帧时发生了发送数据下溢。此时发送进程暂停并把数据下溢错误位置位。	0b
4	ROS	RC	接收状态溢出位。该位被置位表示接收缓冲发生了数据溢出。	0b
3	TJTS	RC	发送啰嗦定时器超时位。该位被置位表示发送器过于繁忙，发送已经停止，描述符的啰嗦超时位已经置位。	0b
2	TBUS	RC	发送缓存不可用状态位。该位被置位表示发送描述符被 CPU 占用，DMA 无法获取，发送流程已经暂停。第[22:20]位显示了当前的发送状态。应用程序需要释放发送描述符。	0b
1	TPSS	RC	发送流程停止状态位。该位被置位表示发送流程已经停止。	0b
0	TS	RC	发送完成标志位。该位被置位表示已经完成一帧的发送，描述符归属权已经交还 CPU。	0b

27.1.8.4.6 DMA 操作模式寄存器（R32_ETH_DMAOMR）

偏移地址：0x1018

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ST	Reserved					FEF	FUGF	Reserved				SR	Res

位	名称	访问	描述	复位值
[31:27]	Reserved	RO	未使用。	
26	DTCEFD	RW	不丢弃 TCP/IP 校验和错误帧控制位。 0：如果 FEF 位为 0，MAC 会丢弃所有有错误的帧； 1:发现 TCP/IP/ICMP/UDP 之类协议的检验和存在错	0b

			误时，不丢弃该帧；	
[25:22]	Reserved	RO	未使用。	
21	TSF	RW	发送存储转发控制位。 0：发送流程写入 FIFO 的数据达到确定值之后就会启动发送； 1：发送流程将完整的帧全部写入 FIFO 后才会启动发送；	0b
20	FTF	RW	发送 FIFO 清空控制位。置此位将复位发送 FIFO。	0b
[19:14]	Reserved	RO	未使用。	
13	ST	RW	开始或停止发送控制位。 0：发送完当前帧之后，发送进程进入停止状态； 1：把发送进程置为运行状态。	0b
[12:8]	Reserved	RO	未使用。	
7	FEF	RW	转发错误帧控制位。 0：除了过短的帧之外，所有的帧都会转发给 DMA。 1：接收 FIFO 会丢弃有错误的帧。	0b
6	FUGF	RW	转发过短的帧控制位。 0：接收 FIFO 丢弃所有长度小于 64 字节的帧； 1：接收 FIFO 转发长度过短的帧。	0b
[5:2]	Reserved	RO	未使用。	
1	SR	RW	开始或停止接收控制位。 0:在转发完当前接收到的帧后，接收 DMA 进入停止模式，下一次传输从当前接收描述符位置开始； 1：开启接收流程，DMA 从当前位置取接收描述符或者从标识符表头位置取接收描述符。	0b
0	Reserved	RO	未使用。	

27.1.8.4.7 DMA 中断使能寄存器（R32_ETH_DMAIER）

偏移地址：0x101C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AISE	ERS	FBES	Reserved		ETIE	RWTIE	RPSIE	RBUIE	RIE	TUIE	ROIE	TJTIE	TBUIE	TPSIE	TIE

位	名称	访问	描述	复位值
31	PLE	RW	内部 10M 物理层连接状态改变中断使能位。	0b
[30:17]	Reserved	RO		
16	NISE	RW	正常中断使能位。使能该位将使能以下中断。 -DMASR[0]:发送中断； -DMASR[2]:发送缓存不可用； -DMASR[6]:接收中断 -DMASR[14]:早接收中断	0b
15	AISE	RW	异常中断。使能该位将使能以下中断。	0b

			-DMASR[1]:发送流程停止; -DMASR[3]:发送啰嗦超时; -DMASR[4]:接收 FIFO 溢出; -DMASR[5]:发送数据下溢; -DMASR[7]:发送缓存不可用; -DMASR[8]:接收流程停止; -DMASR[9]:接收看门狗超时; -DMASR[10]:早发送; -DMASR[13]:总线错误;	
14	ERS	RW	早接收中断使能位。使能该位即可以产生早接收中断。NISE 须置位。AISE 须置位。	0b
13	FBES	RW	总线致命错误中断使能位。使能该位即可以产生总线错误中断。AISE 须置位。	0b
[12:11]	Reserved	RO	未使用。	
10	ETIE	RW	早发送中断使能位。使能该位即可以产生早发送中断。AISE 须置位。	0b
9	RWTIE	RW	接收看门狗中断使能位。使能该位即可以产生接收看门狗超时中断。AISE 须置位。	0b
8	RPSIE	RW	接受流程停止中断使能位。使能该位即可以产生接受流程停止中断。	0b
7	RBUIE	RW	接受缓存不可用中断使能位。使能该位即可以产生接受缓存不可用中断。AISE 须置位。	0b
6	RIE	RW	接收完成中断使能位。使能该位即可以产生接收完成中断。NISE 须置位。	0b
5	TUIE	RW	发送下溢中断使能。使能该位即可以产生发送下溢中断。AISE 须置位。	0b
4	ROIE	RW	接收溢出中断。使能该位即可以产生接收溢出中断。AISE 须置位。	0b
3	TJTIE	RW	发送啰嗦超时中断使能。使能该位即可以产生发送啰嗦超时中断。AISE 须置位。	0b
2	TBUIE	RW	发送缓存不可用中断使能。使能该位即可以产生发送缓存不可用中断。NISE 须置位	0b
1	TPSIE	RW	发送流程停止中断使能。使能该位即可以产生发送流程停止中断。AISE 须置位。	0b
0	TIE	RW	发送完成中断使能。使能该位即可以产生发送完成中断。NISE 须置位。	0b

27.1.8.4.8 DMA 丢失帧寄存器（R32_ETH_DMAMFBOCR）

偏移地址：0x1020

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

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	未使用。	
16	OMFC	RW	丢失帧计数器溢出位。	0b
[15:0]	MFC	RW	丢失帧计数器。这个域表示由于接收缓冲区不可用导致的帧丢失的数量。	0

27.1.8.4.9 DMA 当前发送描述符寄存器（R32_ETH_DMACHTDR）

偏移地址：0x1048

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

位	名称	访问	描述	复位值
[31:0]	CHTDR	RO	这个寄存器值指向目前正在使用的发送描述符。此寄存器由 DMA 负责实时更新。	0

27.1.8.4.10 DMA 当前接收描述符寄存器（R32_ETH_DMACHRDR）

偏移地址：0x104C

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

位	名称	访问	描述	复位值
[31:0]	CHRDR	RO	这个寄存器值指向目前正在使用的接收描述符。此寄存器由 DMA 负责实时更新。	0

27.1.8.4.11 DMA 当前发送缓冲区寄存器（R32_ETH_DMACHTBAR）

偏移地址：0x1050

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

位	名称	访问	描述	复位值
[31:0]	CHTBAR	RO	这个寄存器值指向目前正在使用的发送缓冲区。此寄存器由 DMA 负责实时更新。	0

27.1.8.4.12 DMA 当前接收缓冲区寄存器（R32_ETH_DMACHRBAR）

偏移地址：0x1054

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

位	名称	访问	描述	复位值
[31:0]	CHRBAR	RO	这个寄存器值指向目前正在使用的接收缓冲区。此寄存器由 DMA 负责实时更新。	0

27.1.8.5 内部 10M 物理层相关寄存器地址

27.1.8.5.1 基础控制寄存器（BMCR）

偏移地址：0

位	名称	访问	描述	复位值
15	Reset	RW/SC	1：PHY 复位； 0：正常操作；	0b
14	Loopback	RW	1：使能回环； 0：正常操作；	0b
13	Speed Selection	RO	0：10Mb/s；	0b
12	Auto-Negotiation	RW	1：使能自动协商； 0：禁止自动协商；	1b
[11:10]	Reserved	RO	保留；	0b
9	Restart Auto-Negotiation	RW/SC	1：重新自动协商； 0：正常操作；	0b
8	Duplex Mode	RW	1：全双工； 0：半双工；	1b
7	Collision Test	RW	1：冲突检测使能； 0：正常操作；	0b
[6:0]	Reserved	RO	保留；	0b

27.1.8.5.2 基础状态寄存器（BMSR）

地址偏移：1

位	名称	访问	描述	复位
[15:6]	Reserved	RO	保留；	0
5	Auto-Negotiation Complete	RO	1：自动协商完成； 0：自动协商未完成；	0b
[4:3]	Reserved	RO	保留；	01b
2	Link	RO	1：物理层建立链接； 0：物理层未建立链接；	0b
[1:0]	Reserved	RO	保留；	01b

27.1.8.5.3 物理层状态寄存器（PHYSR）

偏移地址：10

位	名称	访问	描述	复位值
[15:4]	Reserved	RO	保留；	0
3	Loopback_10M	RO	1：PHY 工作于 10M 自循环 0：正常模式	0b
2	Full_10M	RO	1：PHY 工作于 10M 全双工 0：半双工模式	0b
[1:0]	Reserved	RO	保留；	0

27.1.8.5.4 自动翻转寄存器（MDIX）

偏移地址寄存器：0x1E

位	名称	访问	描述	复位值
[15:4]	Reserved	RO	保留；	0
[4:3]	P/N 极性	RW	1x：P/N 极性自动选择 00：P/N 极性正常 01：P/N 极性颠倒	0
[1:0]	T/R 选择	RW	00：自动 01：MDIX 1x：MDI	0

27.2 CH32F20x_D8W、CH32V20x_D8、CH32V20x_D8W 系列

27.2.1 以太网控制器简介

芯片集成了以太网控制器 MAC 和 PHY 以及 DMA，兼容 IEEE802.3 协议。内部 DMA 传输数据和接收数据到系统 RAM 中。PHY 物理层是一个 10Mbit/S 的以太网络收发器，提供部分网络 PHY 寄存器，可以设置发送和接收性能。

网络底层操作主要以子程序库提供应用支持，对寄存器不作详细介绍。主要特性：

- 支持全双工和半双工。
- 支持短包填充设置。
- 支持 CRC 设置和填充。
- 支持巨型帧接收。
- 支持不同的过滤模式组合。
- 支持 pause 帧发送和设置。
- 支持自动协商机制。
- 支持 DMA，用于发送和接收数据。
- PHY 收发器兼容 10BASE-T，发送模块支持节能模式。
- 内置 50 欧姆传输阻抗匹配电阻，也可选择外接。
- 提供由 IEEE 分配的全球唯一 MAC 地址。

27.2.2 寄存器描述

表 18-1 以太网控制器相关寄存器列表

名称	偏移地址	描述	复位值
R8_ETH_EIE	0x40009003	中断使能寄存器	0x00
R8_ETH_EIR	0x40009004	中断标志寄存器	0x00
R8_ETH_ESTAT	0x40009005	状态寄存器	0x00

R8_ETH_ECON2	0x40009006	PHY 模拟参数设置寄存器	0x06
R8_ETH_ECON1	0x40009007	收发控制寄存器	0x00
R32_ETH_TX	0x40009008	发送 DMA 控制寄存器	0xXXXXXXXX
R16_ETH_ETXST	0x40009008	发送 DMA 缓存区起始地址寄存器	0xXXXX
R16_ETH_ETXLN	0x4000900A	发送长度寄存器	0xXXXX
R32_ETH_RX	0x4000900C	接收 DMA 控制寄存器	0x00000000
R16_ETH_ERXST	0x4000900C	接收 DMA 缓冲区起始地址寄存器	0x0000
R16_ETH_ERXLN	0x4000900E	接收长度寄存器	0x0000
R32_ETH_HTL	0x40009010	Hash 表低字节寄存器	0x484EA033
R32_ETH_HTH	0x40009014	Hash 表高字节寄存器	0x5000EF97
R32_ETH_MACON	0x40009018	接收过滤设置寄存器	0x10000000
R8_ETH_ERXFCON	0x40009018	接收包过滤控制寄存器	0x00
R8_ETH_MACON1	0x40009019	Mac 层流控制寄存器	0x00
R8_ETH_MACON2	0x4000901A	Mac 层封包控制寄存器	0x00
R8_ETH_MABBIPG	0x4000901B	最小包间间隔寄存器	0x10
R32_ETH_TIM	0x4000901C	流控制暂停帧时间寄存器	0xXXXXXXXX
R16_ETH_EPAUS	0x4000901C	流控制暂停帧时间寄存器	0xXXXX
R16_ETH_MAMXFL	0x4000901E	最大接收包长度寄存器	0x0000
R16_ETH_MIRD	0x40009020	MII 读数据寄存器	0x1100
R32_ETH_MIWR	0x40009024	MII 写寄存器	0x00000000
R8_ETH_MIREGADR	0x40009024	MII 地址寄存器	0x00
R8_ETH_MISTAT	0x40009025	MII 状态寄存器	0x00
R16_ETH_MIWR	0x40009026	MII 写数据寄存器	0x0000
R32_ETH_MAADRL	0x40009028	MAC 地址低字节寄存器	0xXXXXXXXX
R16_ETH_MAADRH	0x4000902C	MAC 地址高字节寄存器	0xXXXX

27.2.2.1 中断使能寄存器(R8_ETH_EIE)

位	名称	访问	描述	复位值
7	RB_ETH_EIE_INTIE	RW	以太网中断使能： 0：关闭中断； 1：开启中断。	0
6	RB_ETH_EIE_RXIE	RW	接收完成中断使能： 0：关闭中断； 1：开启中断。	0
5	Reserved	RO	保留。	0
4	RB_ETH_EIE_LINKIE	RW	Link 变化中断使能： 0：关闭中断； 1：开启中断。	0
3	RB_ETH_EIE_TXIE	RW	发送完成中断使能： 0：关闭中断； 1：开启中断。	0
2	RB_ETH_EIE_R_EN50	RW	内置的 50 欧姆阻抗匹配电阻使能： 0：片内电阻断开；1：片内电阻连接。	0
1	RB_ETH_EIE_TXERIE	RW	发送错误中断使能： 0：关闭中断； 1：开启中断。	0
0	RB_ETH_EIE_RXERIE	RW	接收错误中断使能： 0：关闭中断； 1：开启中断。	0

27.2.2.2 中断标志寄存器(R8_ETH_EIR)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
6	RB_ETH_EIR_RXIF	RW1	接收完成标志。	0
5	Reserved	RO	保留。	0
4	RB_ETH_EIR_LINKIF	RW1	Link 变化标志。	0
3	RB_ETH_EIR_TXIF	RW1	发送完成标志。	0
2	Reserved	RO	保留。	0
1	RB_ETH_EIR_TXERIF	RW1	发送错误标志。	0
0	RB_ETH_EIR_RXERIF	RW1	接收错误标志。	0

27.2.2.3 状态寄存器(R8_ETH_ESTAT)

位	名称	访问	描述	复位值
7	RB_ETH_ESTAT_INT	RW1	中断。	0
6	RB_ETH_ESTAT_BUFER	RW1	Buffer 错误。	0
5	RB_ETH_ESTAT_RXCRCER	RO	接收 CRC 出错。	0
4	RB_ETH_ESTAT_RXNIBBLE	RO	接收 nibble 错误。	0
3	RB_ETH_ESTAT_RXMORE	RO	接收超过设定最大数据包。	0
2	RB_ETH_ESTAT_RXBUSY	RO	接收进行中。	0
1	RB_ETH_ESTAT_TXABRT	RO	发送被 MCU 打断。	0
0	Reserved	RO	保留。	0

27.2.2.4 PHY 模拟参数设置寄存器（R8_ETH_ECON2）

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0000b
[3:1]	RB_ETH_ECON2_RX RB_ETH_ECON2_MUST	RW	保留，必须写入 011。	011b
0	RB_ETH_ECON2_TX	RW	发送端节能驱动控制： 0：额定驱动；　　1：节能驱动。	0

27.2.2.5 收发控制寄存器(R8_ETH_ECON1)

位	名称	访问	描述	复位值
7	RB_ETH_ECON1_TXRST	RW	发送模块复位： 0：不复位；　　1：复位发送模块。	0
6	RB_ETH_ECON1_RXRST	RW	接收模块复位： 0：不复位；　　1：复位接收模块。	0
[5:4]	Reserved	RO	保留。	00b
3	RB_ETH_ECON1_TXRTS	RW	发送开始，发送完成后自动清零： 1：启动发送；　　0：无动作。	0
2	RB_ETH_ECON1_RXEN	RW	接收使能控制： 0：关闭接收；　　1：开启接收。	0
[1:0]	Reserved	RO	保留。	00b

27.2.2.6 发送 DMA 缓存区地址寄存器(R16_ETH_ETXST)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_ETXST	RW	发送 DMA 缓冲区起始地址。 低 15 位有效，地址必须 4 字节对齐。	XXXXh

27.2.2.7 发送长度(R16_ETH_ETXLN)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_ETXLN	RW	发送长度。	XXXXh

27.2.2.8 接收 DMA 缓冲区地址寄存器(R16_ETH_ERXST)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_ERXST	RW	接收 DMA 缓冲区起始地址。 低 15 位有效，地址必须 4 字节对齐。	XXXXh

27.2.2.9 接收长度寄存器(R16_ETH_ERXLN)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_ERXLN	RO	接收长度。	0000h

27.2.2.10 Hash 表低字节寄存器(R32_ETH_HTL)

位	名称	访问	描述	复位值
[31:24]	R8_ETH_EHT3	RW	Hash Table 字节 3。	48h
[23:16]	R8_ETH_EHT2	RW	Hash Table 字节 2。	4Eh
[15:8]	R8_ETH_EHT1	RW	Hash Table 字节 1。	A0h
[7:0]	R8_ETH_EHT0	RW	Hash Table 字节 0。	33h

27.2.2.11 Hash 表高字节寄存器(R32_ETH_HTH)

位	名称	访问	描述	复位值
[31:24]	R8_ETH_EHT7	RW	Hash Table 字节 7。	50h
[23:16]	R8_ETH_EHT6	RW	Hash Table 字节 6。	00h
[15:8]	R8_ETH_EHT5	RW	Hash Table 字节 5。	EFh
[7:0]	R8_ETH_EHT4	RW	Hash Table 字节 4。	97h

27.2.2.12 接收过滤设置寄存器(R8_ETH_ERXFCON)

位	名称	访问	描述	复位值
7	RB_ETH_ERXFCON_UCEN	RW	目标地址匹配过滤设置： 0：不使用该过滤条件； 1：RB_ETH_ERXFCON_ANDOR=0 目标地址匹配将被接收，RB_ETH_ERXFCON_ANDOR =1 目标地址不匹配将被过滤。	0
6	RB_ETH_ERXFCON_ANDOR	RW	过滤条件 AND 和 OR 设置： 0：任一过滤条件满足包均被接收； 1：所有过滤条件都满足包才被接收。	0
5	RB_ETH_ERXFCON_CRCEN	RW	CRC 校验过滤设置： 0：不使用该过滤条件；	0

			1：RB_ETH_ERXFCON_ANDOR=0 CRC 校验正确将被接收，RB_ETH_ERXFCON_ANDOR =1 CRC 校验错将被过滤。	
4	Reserved	RO	保留。	0
3	RB_ETH_ERXFCON_MPEN	RW	魔法包过滤设置： 0：不使用该过滤条件； 1：RB_ETH_ERXFCON_ANDOR=0 魔法包将被接收，RB_ETH_ERXFCON_ANDOR=1 非魔法包将被过滤。	0
2	RB_ETH_ERXFCON_HTEN	RW	hash 表匹配过滤设置： 0：不使用该过滤条件； 1：RB_ETH_ERXFCON_ANDOR=0 hash 表匹配将被接收，RB_ETH_ERXFCON_ANDOR =1 hash 表不匹配将被过滤。	0
1	RB_ETH_ERXFCON_MCEN	RW	组播包匹配过滤设置： 0：不使用该过滤条件； 1：RB_ETH_ERXFCON_ANDOR=0 组播包匹配将被接收，RB_ETH_ERXFCON_ANDOR=1 组播包不匹配将被过滤。	0
0	RB_ETH_ERXFCON_BCEN	RW	广播包匹配过滤设置： 0：不使用该过滤条件； 1：RB_ETH_ERXFCON_ANDOR=0 广播包将被接收，RB_ETH_ERXFCON_ANDOR=1 非广播包将被过滤。	0

27.2.2.13 Mac 层流控制寄存器(R8_ETH_MACON1)

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留	00b
[5:4]	RB_ETH_MACON1_FCEN	RW	pause 帧设置，全双工下有效 00：停止发送暂停帧； 01：发送一次暂停帧，然后停止发送； 10：周期性发送暂停帧； 11：发送 0 timer 暂停帧，然后停止发送。	00b
3	RB_ETH_MACON1_TXPAUS	RW	发送 pause 帧使能控制： 0：不发送 pause 帧；1：使能发送。	0
2	RB_ETH_MACON1_RXPAUS	RW	接收 pause 帧使能： 0：不接收 pause 帧；1：使能接收。	0
1	RB_ETH_MACON1_PASSALL	RW	控制帧设置： 0：控制帧将被过滤； 1：没被过滤的控制帧将写入缓存。	0
0	RB_ETH_MACON1_MARXEN	RW	MAC 层接收使能： 0：MAC 不接收数据；1：MAC 接收使能。	0

27.2.2.14 Mac 层封包控制寄存器(R8_ETH_MACON2)

位	名称	访问	描述	复位值
[7:5]	RB_ETH_MACON2_PADCFG	RW	短包填充设置： 7：所有短包填充 0 至 64 字节，再 4 字节 CRC； 6：不填充短包； 5:检测到字段为 8100h 的 VLAN 网络包自动填充 0 至 64 字节，否则短包填充 60 字节 0，填充后再 4 字节 CRC； 4：不填充短包； 3：所有短包填充 0 至 64 字节，再 4 字节 CRC； 2：不填充短包； 1：所有短包填充 0 至 60 字节，再 4 字节 CRC； 0：不填充短包。	000b
4	RB_ETH_MACON2_TXCRCEN	RW	发送添加 CRC 控制： 0：硬件不填充 CRC；1：硬件填充 CRC。	0
3	RB_ETH_MACON2_PHDREN	RW	特殊 4 字节不参与 CRC 校验。	0
2	RB_ETH_MACON2_HFRMEN	RW	允许接收巨型帧： 0：不允许接收巨型帧；1：允许接收。	0
1	Reserved	RO	保留。	0
0	RB_ETH_MACON2_FULDPX	RW	以太网络通讯模式： 0：半双工；　　1：全双工。	0

27.2.2.15 最小包间间隔寄存器(R8_ETH_MABBIPG)

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
[6:0]	R8_ETH_MABBIPG	RW	最小包间间隔字节数。	0010000b

27.2.2.16 流控制暂停帧时间寄存器(R16_ETH_EPAUS)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_EPAUS	RW	流控制暂停帧时间。	XXXXh

27.2.2.17 最大接收包长度寄存器(R16_ETH_MAMXFL)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_MAMXFL	RW	最大接收包长度。	0000h

27.2.2.18 MII 读数据寄存器(R16_ETH_MIRD)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_MIRD	RW	MII 读数据寄存器。	1100h

27.2.2.19 MII 地址寄存器(R8_ETH_MIREGADR)

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	000b
[4:0]	RB_ETH_MIREGADR_MIRDL	RW	PHY 寄存器地址。	00000b

27.2.2.20 MII 状态寄存器(R8_ETH_MISTAT)

位	名称	访问	描述	复位值
[7:1]	Reserved	RO	保留。	0000000b
0	R8_ETH_MII_STA	RO	MII 寄存器操作状态： 1：写 MII 寄存器；0：读 MII 寄存器	0

27.2.2.21 MII 写数据寄存器(R16_ETH_MIWR)

位	名称	访问	描述	复位值
[15:0]	R16_ETH_MIWR	WO	MII 写数据寄存器。	0000h

27.2.2.22 MAC 地址寄存器（R32_ETH_MAADRL、R16_ETH_MAADRH)

位	名称	访问	描述	复位值
[31:0]	R32_ETH_MAADRL	RW	MAC Address 字节 1～4。	XXXXXXXXh
[15:0]	R16_ETH_MAADRH	RW	MAC Address 字节 5～6。	XXXXh

27.2.3 操作指南

1. 初始化

（1）、配置安全寄存器进入安全模式，打开以太网络的时钟和电源；
（2）、开启相应的中断，可选的，启动阻抗匹配电阻；
（3）、配置接收过滤模式、CRC 功能、MAC 地址；
（4）、设置缓存；
（5）、启动接收，开启中断。

2. 发送数据

（1）、写入 R16_ETH_ETXLN 数据长度；
（2）、写入 R16_ETH_ETXST 数据地址；
（3）、使能 RB_ETH_ECON1_TXRTS 标志，启动发送。

3. 接收数据

（1）、预先设置好接收地址，使能接收；
（2）、使用中断或查询到接收完成状态；
（3）、读取 R16_ETH_ERXLN 接收长度；
（4）、更新 R16_ETH_ERXST 接收地址

具体的应用请基于以太网协议栈库使用，并参考提供的网络应用示例。

第 28 章 SDIO 接口（SDIO）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

本章单独所述“SDIO”，是指微控制器上一个为操作 SD 卡等外部存储卡或其他设备而设计的通信接口，是微控制器的一个外设。微控制器的 SDIO 直接挂载在 AHB 总线上，由 HCLK 直接提供时钟，能实现较高的通讯速度，微控制器的 SDIO 用作 SDIO 主机，被控制的设备也被统称 SDIO 设备。应用中一般使用 SDIO 来读写 SD 卡、TF 卡或 eMMC 颗粒，或控制其他使用 SDIO 作为通讯接口的设备，比如 WiFi/4G 模块。

28.1 主要特征

28.1.1 特征

- 支持 SD 卡、SDIO 卡和 MMC 卡
- 支持 1 位、4 位和 8 位总线
- SDIO 的时钟最快可达到 HCLK 的一半
- 兼容 MMC 规范 4.5
- 兼容 SD 卡规范 2.0，SDIO 卡规范 2.0
- 不兼容 SPI 或 QSPI

28.1.2 概述

微控制器的 SDIO 支持与 SD 卡或 MMC 卡等存储器通讯，需要明确的是，SDIO 仅仅是提供一组实现 SD 卡、MMC 卡规范单次命令传输所需要的时钟，数据和命令控制时序，各命令间的先后组合需用用户通过程序自行确定。此外，对于各种存储卡而言，SDIO 仅仅只能实现读写功能，文件系统所提供的对文件的功能需要用户自行通过程序构建文件系统而实现。

SDIO 不同于 QSPI 接口，其没有片选引脚，并多出一个 CMD 引脚，CMD 可以认为是一个特殊数据线，专门用来传输命令和响应；SDIO 有 1 位、4 位和 8 位三种数据线宽度可选；SDIO 的时钟在配置时一般工作在 400KHz 以下的频率，当正式进行数据传输时，则可以配成 SDIO 设备所支持的最大时钟，微控制器所支持的最大 SDIO 时钟输出为 HCLK 的一半，按照协议当 SDIO 设备接收的时钟大于某个阈值时，需要降低时钟线、数据线和命令线的波形峰值，以节省波形上升和下降的耗时。

与 SD 卡不同，SDIO 卡经常指代使用 SDIO 接口的 WIFI/蓝牙模块和 4G 模块等非存储设备。如果没有特殊说明，本章叙述的内容一定适用于 SD 卡，只适用于 SDIO 卡或 MMC 卡的内容将会特别指出。本章在叙述时，优先将 SD 卡视为潜在操作对象，其次为 SDIO 卡，最后为 MMC 卡。

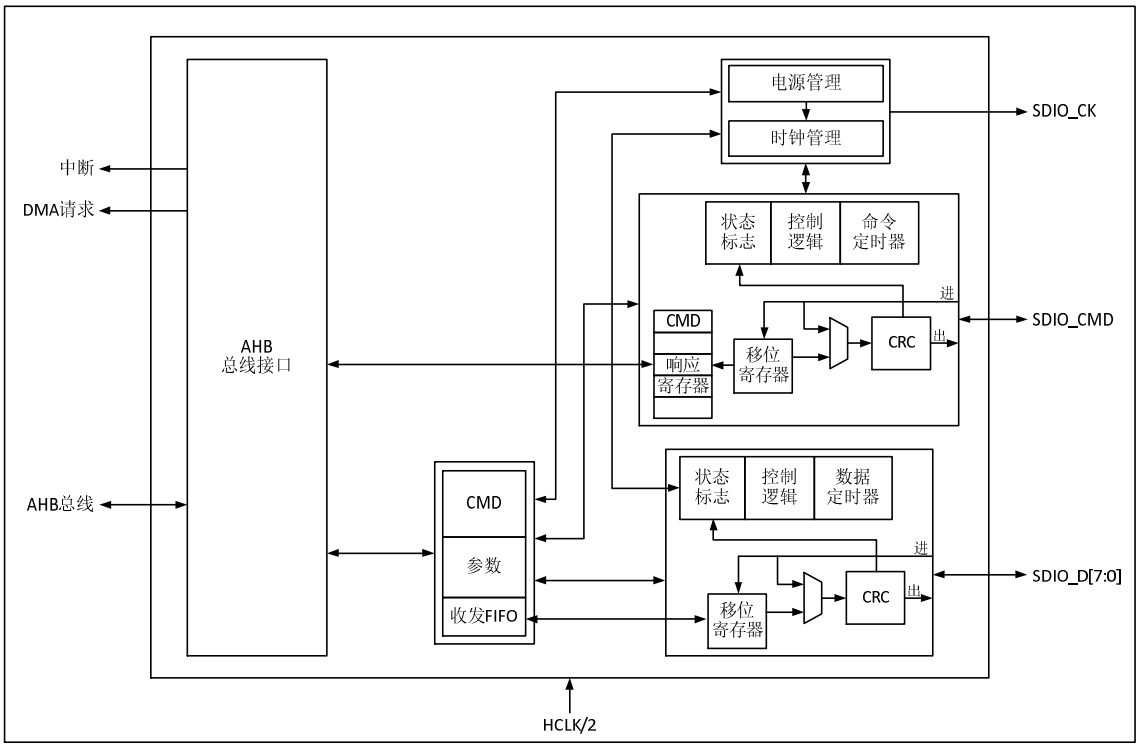
28.2 接口和时钟

SDIO 通过 AHB 总线接收 CPU 对其的控制，SDIO 的寄存器中有一个 FIFO 接口，CPU 或 DMA 通过读写 FIFO 获取或发生数据。SDIO 由 HCLK 直接供给时钟，并拥有一个中断入口，支持多种中断源。由 SDIO 直接控制的引脚有 SDIO_CK、SDIO_CMD、SDIO_D[7:0]这十个，通过几个引脚连接到 SDIO 设备上。SDIO 是一个主机主导的通讯接口，所有的传输必须由微控制器发起。

28.2.1 外设结构

SDIO 的结构如图 28-1 所示。

图 28-1 SDIO 的结构图



SDIO 是微控制器作为主机直接操作 SDIO 设备的外设，其大概由 AHB 接口、时钟控制部分、CMD 线控制部分，数据线控制部分和控制寄存器部分这五个模块组成，SDIO 是一个半双工的外设，CPU 向控制寄存器写入需要发送的命令和数据，命令线和数据线控制模块负责将命令或数据推到 IO 上，并加上 CRC。SDIO 的数据流由通用 DMA 负责完成从 RAM 到 SDIO 的 FIFO 的搬移，SDIO 的 FIFO 有 32 个 32 位的大小。

28.2.2 引脚及其配置

SDIO 需要配置的引脚及其模式见表 28-1。

表 28-1 SDIO 的引脚配置

GPIO	SDIO 复用功能	需要配置成的引脚模式	需要配置成的引脚速度
PC[8:11]	SDIO_D[0:3]	推挽复用输出	50MHz
PC12	SDIO_CK	推挽复用输出	50MHz
PD2	SDIO_CMD	推挽复用输出	50MHz
PB8	SDIO_D4	推挽复用输出	50MHz
PB9	SDIO_D5	推挽复用输出	50MHz
PC6	SDIO_D6	推挽复用输出	50MHz
PC7	SDIO_D7	推挽复用输出	50MHz

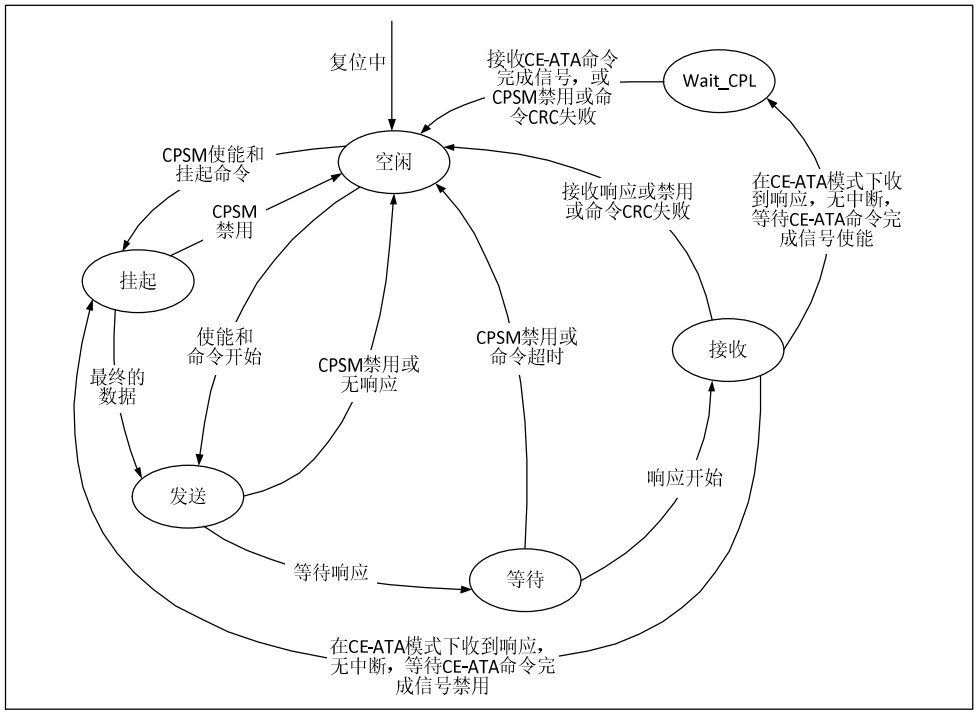
28.2.3 时钟

SDIO 的时钟引脚为 SDIO_CK，其输出时钟由 HCLK 分频得到，分频系数可配置为 2-261 之间的任意整数值。SDIO 设备在初始化的时候一般只支持最高 400KHz 时钟的单总线模式，在初始化后，主机一般会发起切换至低电压的操作，同时将时钟提升至微控制器和 SDIO 设备双方能接受的最大时钟。不同版本和速度等级的 SD 卡所支持时钟速度和切换流程有所区别，用户需要自行了解。

28.2.4 命令状态机

SDIO 的命令工作流程遵循如下图的状态机。

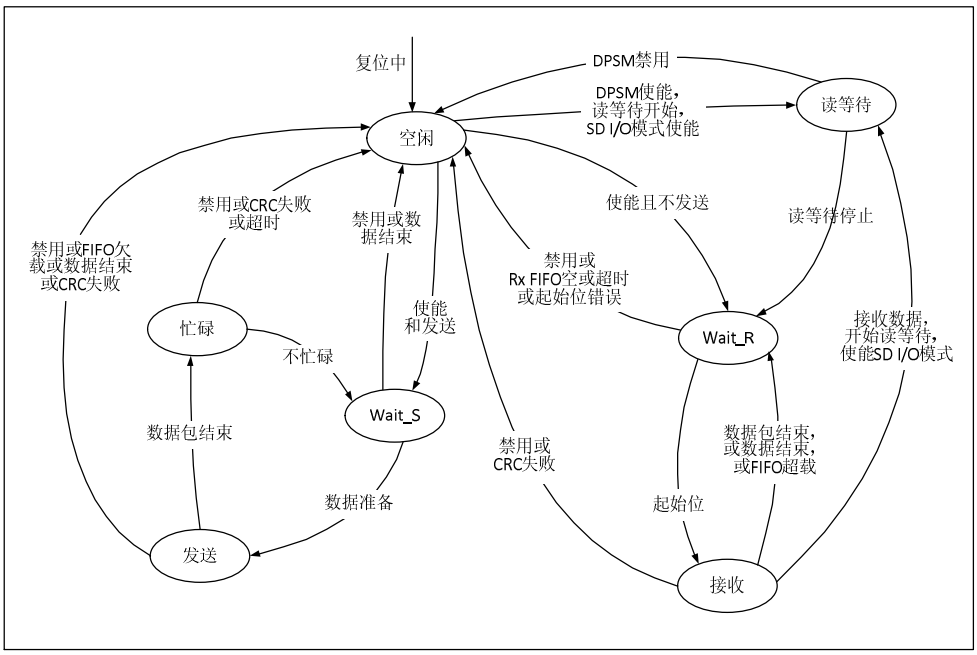
图 28-2 命令状态机



28.2.5 数据状态机

SDIO 的数据工作流程遵循如下图的状态机。

图 28-3 数据状态机



28.3 SDIO 协议简述

SDIO 上的通讯以传输为最小单位，每个传输总是以主机在 CMD 线上发送命令为起始，有的命令发送之后，SDIO 设备同样会在 CMD 线上发送一段数据回复主机，称为“响应”，有时还会伴随着数据的传输，数据的传输是在 D 线上。命令和响应的格式是固定的，各域各位的定义根据不同的命令或响应而确定。

响应和数据传输需要在命令或响应结束后规定的时间内发出或停止，否则将产生超时错误。

本小节的目的是以较小的篇幅让用户对使用 SDIO 所必要的一些规范的细节有初步的了解，并不保证详尽和更新及时。微控制器的 SDIO 也仅仅保证其实现了 SD 2.0，SDIO 2.0 及 MMC 4.5 规范的硬件操作基础，对于更高版本规范定义的功能，例如双边沿采样等，并不一定支持。用户在做具体开发时应以 SD 规范，SDIO 规范和 MMC 规范为依据进行 SDIO 交互编程。

28.3.1 总线时序

SD 卡的传输都是以主机发起 CMD 发起的，SD 卡可能不回复响应，可能回复短响应和长响应，有的响应后面还会伴随数据传输。而在 SDIO 卡或 MMC 卡的通讯中，还可能会有 SDIO 设备主动报中断的情况。时序如下图组所示。

图 28-4 SDIO 的无响应时序和无数据时序（以 SD 卡为例）

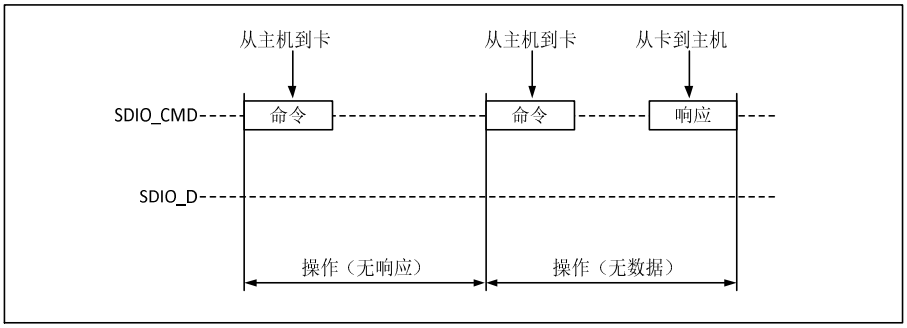


图 28-5 SDIO 的多块读时序（以 SD 卡为例）

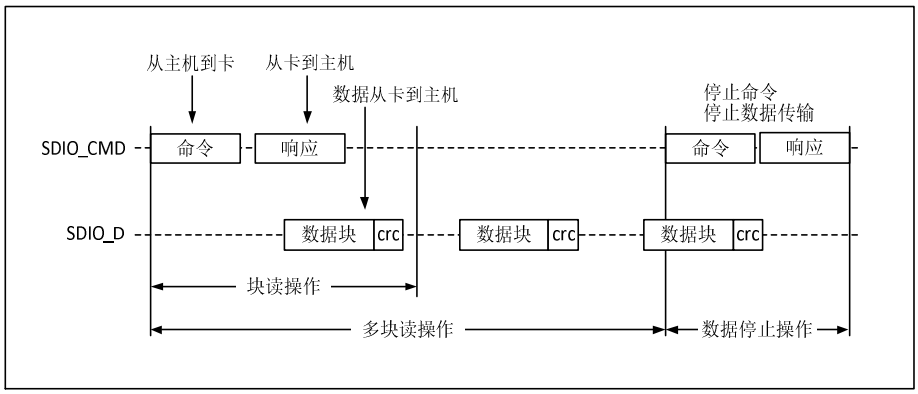


图 28-6 SDIO 的多块写时序（以 SD 卡为例）

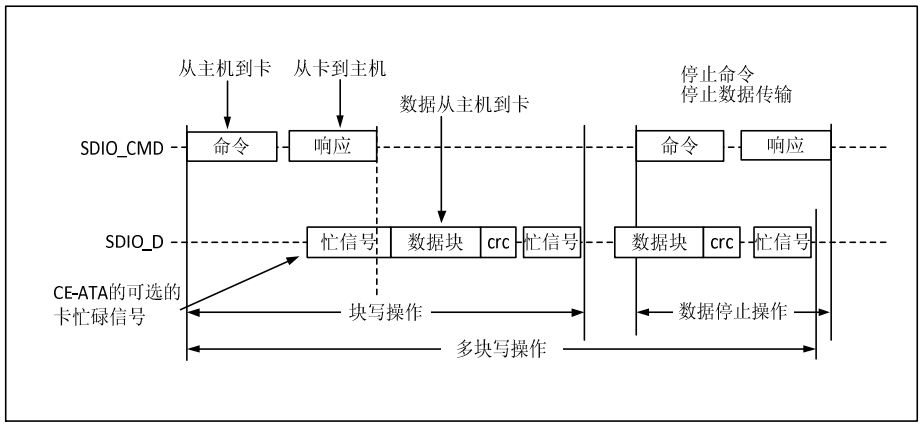


图 28-7 SDIO 的数据流读时序（以 SDIO 卡为例）

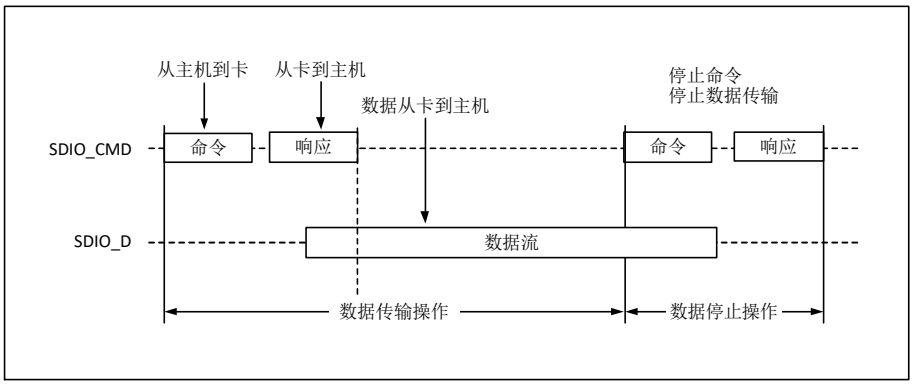
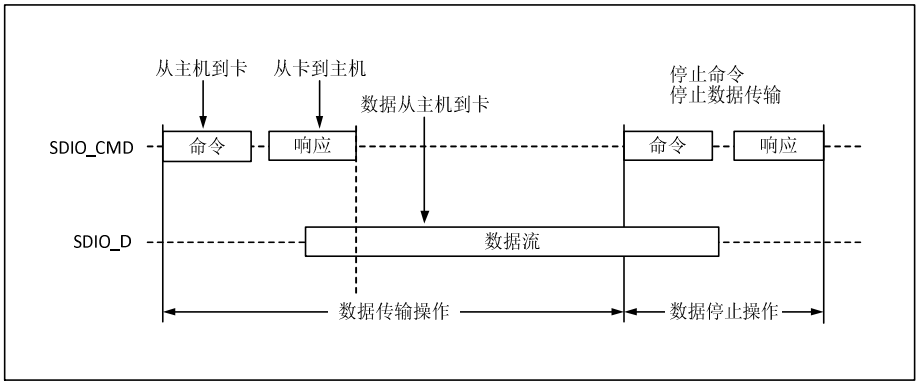


图 28-8 SDIO 的数据流写时序（以 SDIO 卡为例）



28.3.2 命令

SDIO 传输大部分以命令（CMD）开始，SDIO 主机通过命令向设备告知自己的意图。命令的格式如下表。

表 28-2 命令格式

位/域的名称	起始位	传输位	命令索引	命令参数	CRC7	结束位
位次/宽度	47	46	[45:40] / 6 bits	[39:8] / 32 bits	[7:1] / 7 bits	0
值	0b	1b	X	X	X	1b

命令大致有四种：

1） 广播命令（bc）：发给总线上的所有的卡，没有响应返回；
2） 带响应的广播命令（bcr）：发给总线上的所有的卡，有响应返回；
3） 点对点命令（ac）：发给特定的卡，但没有数据传输；
4） 带数据传输的点对点命令（adtc）：发给特定的卡，并附带数据传输；

下面是一些常用的命令。

28.3.2.1 基础命令

基础命令是 SD 卡所支持的一些较基本的功能。

表 28-3 SD 卡的基础命令

命令索引	类型	参数	回复类型	简写	描述
CMD0	bc	[31:0]填充位无意义；	无	GO_IDLE_STATE	复位所有的卡到 IDLE 状态

CMD2	bcr	[31:0]填充位无意义；	R2	ALL_SEND_CID	所有卡回复 CID
CMD3	bcr	[31:0]填充位无意义；	R6	SEND_RELATIVE_ADDR	回复新的 RCA
CMD7	ac	[31:16]RCA； [15:0]填充位无意义；	R1b 选中卡	SELECT/DESELECT_CARD	选中或者取消选选中某个卡
CMD8	bcr	[31:12]保留； [11:8]供电； [7:0]校验模式；	R7	SEND_IF_COND	发送 SD 卡接口条件。
CMD9	ac	[31:16]RCA； [15:0]填充位无意义；	R2	SEND_CSD	要求 CSD。
CMD10	ac	[31:16]RCA； [15:0]填充位无意义；	R2	SEND_CID	要求 CID。
CMD11	ac	[31:0]填充位无意义；	R1	VOLATGE_SWITCH	切换到 1.8V 电平。
CMD12	ac	[31:0]填充位无意义；	R1b	STOP_TRANSMISSION	强制卡停止传输；
CMD13	ac	[31:16]RCA； [15]发送任务状态寄存器； [14:0] 填充位；	R1	SEND_STATUS/SEND_TASK_STATUS	发送状态或任务状态寄存器
CMD15	ac	[31:16]RCA； [15:0]填充位无意义；	无	GO_INACTIVE_STATE	要求卡进到 INACTIVE 模式；

28.3.2.2 擦除命令

SD 卡也是 FLASH 的结构，写入前也需要擦除 FLASH，但是 SD 卡内部集成了擦除逻辑，在执行写命令前如果发现没有擦除会自动补上擦除操作。在很多情况下特别是大批量写之前，如果主动执行擦除有助于提高效率。

表 28-4 SD 卡的擦除命令[1]

命令索引	类型	参数	回复类型	简写	描述
CMD32	ac	[31:0]开始擦除的地址[2]	R1	ERASE_WR_BLK_START	设定擦除的首地址
CMD33	ac	[31:0]结束擦除的地址[2]	R1	ERASE_WR_BLK_END	设定擦除的尾地址
CMD38	ac	[31:0]擦除模式	R1b	ERASE	参数为 0 即普通擦

注 1：这里的擦除命令是 SD 协议规范定义的 SD 卡的擦除命令，SDIO 卡和 MMC 卡的擦除命令与此有区别；2：目前常用的 SDHC/SDXC（2GB 到 2TB）级别的卡写入的擦除地址必须是块地址，即 512 字节对齐；

28.3.2.3 块传输的读命令

表 28-5 SD 卡的块读命令[1]

命令索引	类型	参数	回复类型	简写	描述
CMD16	ac	[31:0]块长度	R1	SET_BLOCKLEN	写入块长度，512
CMD17	adtc	[31:0]单个块的地址[1]	R1	READ_SINGLE_BLOCK	设定单个读的起始地址
CMD18	adtc	[31:0]多个块的地址[1]	R1	READ_MULTIPLE_BLOCK	设定单个读的起始地址
CMD19	adtc	[31:0]保留位	R1	SEND_TUNING_BLOCK	发送表示模式变化的 64 字节序列

CMD20	ac	[31:28]保留位 [27:0]速度控制位	R1b	SPEED_CLASS_CONTROL	速度控制控制
CMD22	ac	[31:6]保留位 [5:0]扩展地址	R1	ADDRESS_EXTENSION	SDUC 才会用用到。
CMD23	ac	[31:0]块计数器	R1	SET_BLOCK_COUNT	块计数器

注：1：目前常用的 SDHC/SDXC（2GB 到 2TB）级别的卡写入的擦除地址必须是块地址，即 512 字节对齐。

28.3.2.4 块传输的写命令

表 28-6 SD 卡的块读命令[1]

命令索引	类型	参数	回复类型	简写	描述
CMD16	ac	[31:0]块长度	R1	SET_BLOCKLEN	写入块长度，512
CMD24	adtc	[31:0]块的地址[1]	R1	WRITE_BLOCK	设定单个块写的起始地址
CMD25	adtc	[31:0]多个块的地址[1]	R1	WRITE_MULTIPLE_BLOCK	设定多个块写的起始地址
CMD27	adtc	[31:0]填充位	R1	PROGRAM_CSD	对 CSD 可编程的字编程

注：1：目前常用的 SDHC/SDXC（2GB 到 2TB）级别的卡写入的擦除地址必须是块地址，即 512 字节对齐。

28.3.3 响应

响应作为 SDIO 设备对主机的必要回复，同样也是在 CMD 线上传输的，且必须在规定的时间内回复。响应传输为高位在前低位在后，响应长度和各比特各域的定义由响应的类型具体决定，但所有的响应都是由一个固定为 0 的起始位开始，跟着一个为固定 0 的传输方向位[1]。所有的响应最后都有一个停止位，固定为 1[1]。大致有 7 种响应，SD 卡支持 R1 / R1b / R2 / R3 / R6 / R7，SDIO 卡还支持 R4 / R5，格式见下文。

28.3.3.1 R1 响应

普通响应，总长 48 位，有 CRC7 校验，卡状态域为 32 位。格式如下表。

表 28-7 R1 的格式

位/域的名称	起始位	传输位	响应索引	卡状态	CRC7	结束位
位次/宽度	47	46	[45:40] / 6 bits	[39:8] / 32 bits	[7:1] / 7 bits	0
值	0b	0b	跟随 CMD 索引	X	X	1b

28.3.3.2 R1b 响应

R1b 的格式和 R1 一致，但是可以在响应后添加繁忙信号，即钳住数据线 D2。主机收到繁忙信号（检测到 SDIO_D2 为低）后需要进行相应处理。

28.3.3.3 R2 响应

应对特定几个命令的响应，总长 136 位，CRC7 校验包含在卡状态域中，卡状态域为 128 位。卡状态域存放 CID 寄存器或 CSD 寄存器的值,CID 寄存器一般作为 CMD2/CMD10 的回复,CSD 寄存器一般作为 CMD9 的回复，CID/CSD 寄存器的具体含义见 28.4.2 设备寄存器部分。需要注意的是 R2 只会回复 CID/CSD 寄存器的[127:1]段，CID/CSD[0]固定为 1 的保留位[2]被结束位占据，结束位固定也为 1。格式如下表。

表 28-8 R2 的格式

位/域的名称	起始位	传输位	命令索引	卡状态	结束位
位次/宽度	135	134	[133:128] / 6 bits	[127:1] / 127 bits	0
值	0b	0b	111111b	CID/CSD	1b

28.3.3.4 R3 响应

回复 OCR 寄存器的专用响应，总长 48 位，没有 CRC7 校验。OCR 寄存器一般作为 ACMD41 的回复。格式如下表。

表 28-9 R3 的格式

位/域的名称	起始位	传输位	命令索引	卡状态	保留	结束位
位次/宽度	47	46	[45:40] / 6 bits	[39:8] / 32 bits	[7:1] / 7 bits	0
值	0b	0b	111111b	OCR	1111111b	1b

28.3.3.5 R4 响应

应对 CMD5 的响应回复 OCR 及相关寄存器的专用响应，总长 48 位，有 CRC7 校验。R4 应用在 SDIO 卡中，格式如下表。

表 28-10 R4 的格式

位/域的名称	位次	宽度（单位：bit）	值
起始位	47	1	0b
传输位	46	1	0b
保留	[45:40]	6	111111b
卡就绪	39	1	X
IO 功能数目	[38:36]	3	X
当前寄存器	[35]	1	X
填充位	[34:33]	2	00b
S18A	32	1	X
IO OCR	[31:8]	24	OCR
CRC 校验域	[7:1]	7	1111111b
结束位	0	1	1b

注：MMC 卡 R4 的格式和 SDIO 卡不同。

28.3.3.6 R5 响应

应对 CMD5 的专用响应，总长 48 位，有 CRC7 校验。R5 应用在 SDIO 卡中，格式如下表。

表 28-11 R5 的格式

位/域的名称	起始位	传输位	命令索引	填充位	响应格式	读写数据	CRC7	结束位
位次	47	46	[45:40]	[39:24]	[23:16]	[15:8]	[7:1]	0
宽度	1	1	6	16	8	8	7	1
值	0b	0b	110100b	0000h	X	X	X	1b

注：MMC 卡 R5 的格式和 SDIO 卡不同。

28.3.3.7 R6 响应

回复 RCA 的专用响应，总长 48 位，有 CRC7 校验，格式如下表。

表 28-12 R6 的格式

位/域的名称	起始位	传输位	命令索引	卡的 RCA	卡状态位	CRC7	结束位
位次	47	46	[45:40]	[39:24]	[23:8]	[7:1]	0
宽度	1	1	6	16	16	7	1
值	0b	0b	000011b	X	X	X	1b

28.3.3.8 R7 响应

应对 CMD8 的专用响应，表明支持的电压的信息，总长 48 位，有 CRC7 校验，格式如下表。

表 28-13 R7 的格式

位/域的名称	起始位	传输位	命令索引	保留位	PCIe 1V2 支持	PCIe 响应	接受的电压	检查回馈	CRC7	结束位
位次	47	46	[45:40]	[39:22]	21	20	[19:16]	[15:8]	[7:1]	0
宽度	1	1	6	18	1	1	4	8	7	1
值	0b	0b	001000b	00000h	X	X	X	X	X	1b

28.3.4 数据传输

数据传输在数据线 SDIO_D 上进行，有 1/4/ 8 位三种宽度，数据传输时一般是一个时钟的起始位在最前面，每个字节的高位在前，低位在后。对于 SD 卡仅仅支持的块传输模式，每个块的数据传输结束后还跟着 CRC 校验。MMC 卡还支持数据流传输模式，此时不附带 CRC。下图为数据传输的格式。

图 28-9 SD 卡单总线数据传输字节的格式

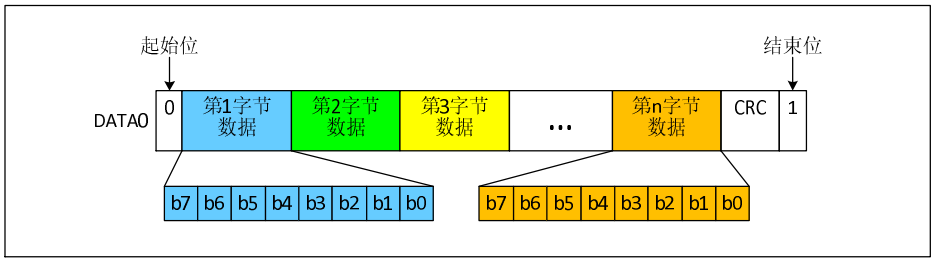


图 28-10 SD 卡 4 总线数据传输字节的格式

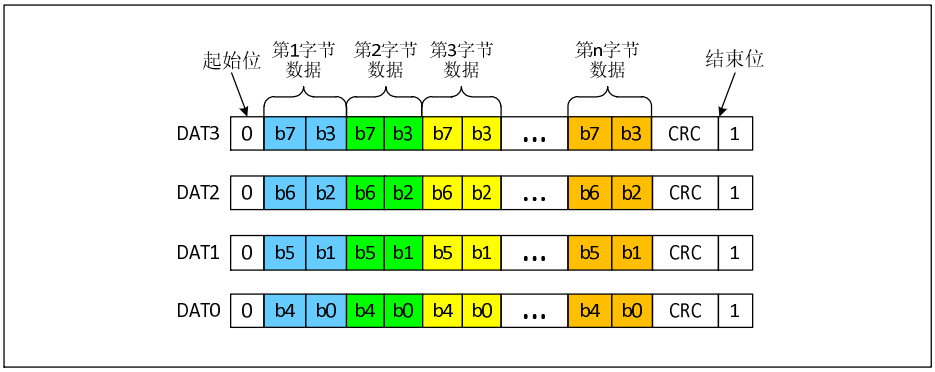


图 28-11 SD 卡单总线数据传输一个 512 位字的格式

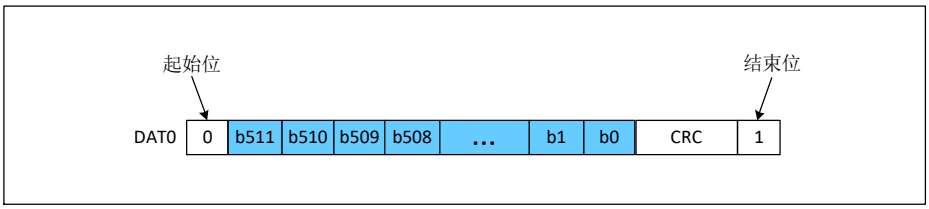
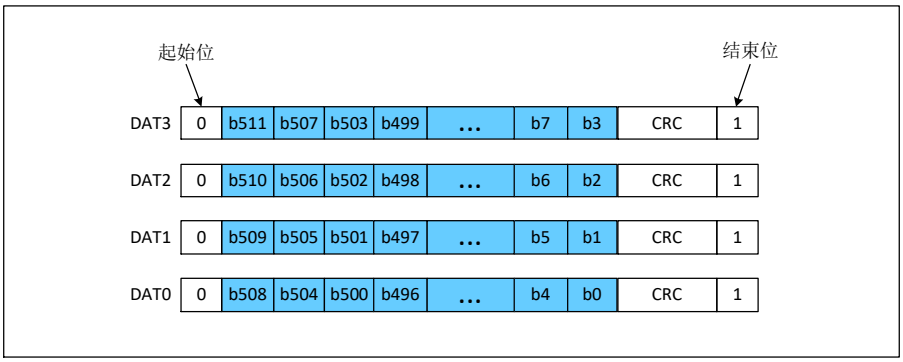


图 28-12 SD 卡 4 总线数据传输一个 512 位字的格式



28.4 应用

28.4.1 设备初始化和设备寄存器

28.4.1.1 OCR 寄存器

操作条件寄存器（Operation Conditions Register）存储了 SD 卡的一些其接收的供电电压的信息和相关的状态位。相关的位定义如下表。

表 28-14 OCR 寄存器位定义

位次	位定义	描述
0-3	保留。	支持的 VDD 电压范围，单位为伏特。
4	保留。	
5	保留。	
6	保留。	
7	为低电压范围保留	
8	保留	
9	保留	
10	保留	
11	保留	
12	保留	
13	保留	
14	保留	
15	2.7-2.8	
16	2.8-2.9	
17	2.9-3.0	
18	3.0-3.1	
19	3.1-3.2	
20	3.2-3.3	
21	3.3-3.4	

22	3.4-3.5	
23	3.5-3.6	
24	接受切换到 1.8V	
25-26	保留	
27	超过 2TB 支持状态位	
28	保留	
29	UHS-II 卡状态位	此位被置位表示此卡支持 UHS-II 接口。
30	卡容量状态（CCS）	此位被置位表示卡容量大于 2GB。
31	卡上电状态位（busy）	此位在卡上电完成后被置位。上电完成后，其他位才有意义。

28.4.1.2 CID 寄存器

CID 存储了一些身份识别信息。

表 28-15 CID 寄存器各位各域的定义

位/域名称	简写	宽度	位次
厂商 ID	MID	8	[127:120]
应用 ID	OID	16	[119:104]
产品名称	PNM	40	[103:64]
产品版本	PRV	8	[63:56]
产品序列号	PSN	32	[55:24]
保留	无	4	[23:20]
生产日期	MDT	12	[19:8]
CRC7	CRC	7	[7:1]
固定位，固定为 1	无	1	[0]

28.4.1.3 CSD 寄存器

CSD 寄存器存储了 SD 卡的特征数据。以目前最常用的 SDHC 和 SDXC 卡最常用的第二版的 CSD 为例，各位域的定义如下表。

表 28-16 CSD 寄存器各位域的含义

名称	简写	宽度	值	读写	位次
CSD 版本	CSD_STRUCTURE	2	01b	RO	[127:126]
保留	无	6	00_0000b	RO	[125120
读数访问时间	TAAC	8	0Eh	RO	[119:112]
用时钟周期表示的读数访问时间	NSAC	8	00h	RO	[111:104]
最大数据发送速度	TRAN_SPEED	8	32h5Ah0Bh2Bh	RO	[103:96]
卡命令类	CCC	12	X1X1101101X1b	RO	[95:84]
读数据块最大长度	READ_BL_LEN	4	9	RO	[83:80]
允许块部分读	READ_BL_PARTIAL	1	0	RO	[79]

块写不对齐	WRITE_BLK_MISALIGN	1	0	RO	[78]
块读不对齐	READ_BLK_MISALIGN	1	0	RO	[77]
执行的 DSR	DSR_IMP	1	X	RO	[76]
保留	无	6	00_0000b	RO	[75:70]
设备大小	C_SIZE	22	XXXXXXh	RO	[69:48]
保留	无	1	0	RO	[47]
单块擦使能	ERASE_BLK_EN	1	1	RO	[46]
擦扇区尺寸	SECTOR_SIZE	7	7Fh	RO	[45:39]
写保护组大小	WP_GRP_SIZE	7	0000000b	RO	[38:32]
写保护组使能	WP_GRP_ENABLE	1	0	RO	[31]
保留	无	2	00b	RO	[30:29]
写速度因素	R2W_FACTOR	3	010b	RO	[28:26]
最大写数据块长度	WRITE_BL_LEN	4	9	RO	[25:22]
允许块部分写	WRITE_BL_PARTIAL	1	0	RO	[21]
保留	无	5	00000b	RO	[20:16]
文件格式组	FILE_FORMAT_GRP	1	0	RO	[15]
复制标志	TMP_WRITE_PROTECT	1	X	RW OTP	[14]
永久写保护	PERM_WRITE_PROTECT	1	X	RW OTP	[13]
临时写保护	TMP_WRITE_PROTECT	1	X	RW	[12]
文件格式	FILE_FOMAT	2	00b	RO	[11:10]
保留	无	2	00b	RO	[9:8]
CRC	CRC	7	0000000b	RW	[7:1]
未使用，必须使用 1	无	1	1b	RO	[0]

28.4.1.4 RCA 寄存器

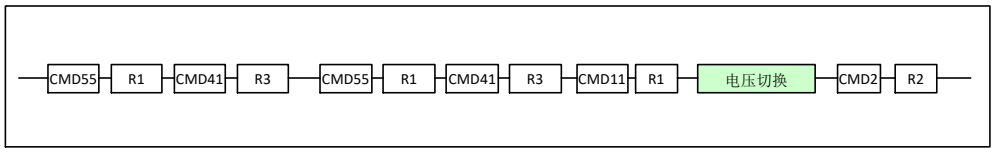
相对卡地址寄存器存储了卡的地址，为 16 位，默认值为 0.

28.4.2 电压切换

在 SD 卡初始化后期，需要进行接口电平切换，将 SD 卡的时钟线数据线和命令线的 IO 电平切换到 1.8V 水平。对于压摆率不是足够优秀的器件，使用更低的电平标准有助于提升频率。但是 SD 的供电电压并不一定变化，只在较新版本的协议才出现了低电压供电的 SD 卡。

切换电压的步骤如下图。

图 28-13 电压切换序列



28.4.3 时钟切换

初始化时 SD 卡的时钟只有 400KHz，在电压完成切换之后可以将时钟提升至较高的水平，例如 SDHC 卡 UHS-I 模式第一档速度，总线时钟可以达到 80MHz，鉴于微控制器的 IO 输出能力，应将时钟限制在 50MHz 之内。

28.5 中断

28.5.1 SDIO 中断

SDIO 支持多种中断源，如中断使能寄存器（R32_SDIO_IER）所示的有 24 种情况都可以触发中断，用户可酌情自行开启。

28.5.2 SDIO 设备中断

需要注意的是不光 SDIO 外设可以向 CPU 报中断，外接 SDIO 卡和 MMC 卡也能向 SDIO 外设报中断。在 4 位总线模式下，中断线是 D1，在 8 位总线模式下，中断线是 D7，低电平有效。如果在空闲状态下 SDIO 检测到 D1 或 D7 为低电平，应读取 SDIO 设备的状态寄存器或中断标志寄存器及时响应中断。CPU 通过 R32_SDIO_SR:22 位可以得到 SDIO 主机是否收到中断。

28.6 寄存器描述

表 28-17 SDIO 相关寄存器列表

名称	访问地址	描述	复位值
R32_SDIO_PWR	0x40018C00	电源寄存器	0x00000000
R32_SDIO_CLK	0x40018C04	时钟寄存器	0x00000000
R32_SDIO_ARG	0x40018C08	命令参数寄存器	0x00000000
R32_SDIO_CMD	0x40018C0C	命令寄存器	0x00000000
R32_SDIO_RESI	0x40018C10	响应寄存器	0x00000000
R128_SDIO_RESX	0x40018C14	响应参数寄存器	0x00000000
R32_SDIO_DTIM	0x40018C24	数据定时寄存器	0x00000000
R32_SDIO_TLEN	0x40018C28	传输长度寄存器	0x00000000
R32_SDIO_DCTLR	0x40018C2C	数据控制寄存器	0x00000000
R32_SDIO_DCNT	0x40018C30	传输计数寄存器	0x00000000
R32_SDIO_SR	0x40018C34	状态寄存器	0x00000000
R32_SDIO_ICR	0x40018C38	中断清除寄存器	0x00000000
R32_SDIO_IER	0x40018C3C	中断使能寄存器	0x0000_0000
R32_SDIO_FIFOCNT	0x40018C48	FIFO 计数器	0x0000_0000
R32_SDIO_FIFO	0x40018C80	FIFO 寄存器	0x0000_0000

28.6.1 电源寄存器（R32_SDIO_PWR）

偏移地址：0x00

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

位	名称	访问	描述	复位值
[31:2]	Reserved	RO	保留。	0
[1:0]	PWRCTL	RW	电源检测位： 00：电源关闭，时钟停止； 01：保留； 10：保留的上电状态； 11：上电状态，卡时钟开启；	0

28.6.2 时钟寄存器(R32_SDIO_CLK)

偏移地址：0x04

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	HWCE	EDGE	WDBUS		BYPAS	PSPB	CLKEN	CLKDIV							

位	名称	访问	描述	复位值
[31:15]	Reserved	RO	保留。	0
[14]	HWCE	RW	硬件流控使能位。此位置位后，TXFIFOE 和 RXFIFOF 信号才起作用。 0：关闭硬件流控； 1：开启硬件流控；	0
[13]	EDGE	RW	SDIO_CK 相位选择位。 0：在 HCLK 的上升沿产生 SDIO_CK； 1：在 HLCK 的下降沿生产 SDIO_CK；	0
[12:11]	WDBUS	RW	总选宽度配置域。 00：1 位总线模式，使用 SDIO_D0; 01：4 位总线模式，使用 SDIO_D[3:0]; 10：8 位总线模式，使用 SDIO_D[7:0]; 11：未使用；	0
[10]	BYPAS	RW	时钟旁路使能位。 0：SDIO_CK 通过分频器分频得出； 1：SDIO_CK 直接接到 HCLK/2 上；	0
[9]	FCC	RW	空闲时时钟状态配置位。 0：SDIO_CK 始终输出； 1：SDIO_CK 只在需要时输出；	0
[8]	CLKEN	RW	时钟使能位。 0：SDIO_CK 被禁止输出； 1：SDIO_CK 被允许输出；	0
[7:0]	CLKDIV	RW	时钟分频系数域，此域表示 SDIO_CK 与 HCLK 的关系。SDIO_CK=HCLK/(CLKDIV+2) 。注意在初始化阶段，SDIO_CK 应低于 400KHz。	0

注：时钟配置寄存器用来控制 SDIO_CK 相关的参数，需要注意的是本寄存器在读写数据期间到 7 个 HCLK 周期内不能更改。

28.6.3 命令参数寄存器(R32_SDIO_ARG)

偏移地址：0x08

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

位	名称	访问	描述	复位值
[31:0]	CMDARG	RW	命令的参数域。此域存放的是命令中的参数，会作为命令的一部分一起发生到 CMD 线上。	0

28.6.4 命令寄存器(R32_SDIO_CMD)

偏移地址：0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	ATA	NIE	NEC	SDS	CPSE	WAIT	WAITI	WAITP		CMDIDX					

位	名称	访问	描述	复位值
[31:15]	Reserved	RO	保留。	0
[14]	ATA	RW	进行 CE-ATA 命令。如果置此位，CPSM 将转到 CMD61。	0
[13]	NIE	RW	不使能 CE-ATA 中断设置位。如果置此位，CE-ATA 将不产生中断。	0
[12]	NEC	RW	使能 CMD 完成信号使能位。如果置此位，命令完成将产生信号。	0
[11]	SDS	RW	暂停命令发生位。如果置此位，将发送一个暂停信号。仅适用于 SDIO 卡。	0
[10]	CPSE	RW	CPSM(命令通道状态机)使能位。如果置此位，将使能 CPSM。	0
[9]	WAIT	RW	命令等待控制位。如果设置此位，发送命令前，CPSM 会等待数据传输完成。	0
[8]	WAITI	RW	命令等待中断控制位。如果置此位，CPSM 会关闭超时控制并等待中断产生。	0
[7:6]	WAITP	RW	响应类型等位域。此域指示 CPSM 期望收到的响应类型。 00：无响应，等待 CMDSENT 标志； 01：短响应，等待 CMDREND 或 CCRCFAIL 标志； 10：无响应，等待 CMDSENT 标志； 11：长响应，等待 CMDREND 或 CCRCFAIL 标志；	0
[5:0]	CMDIDX	RW	命令索引域。此域表明具体的命令值。	0

28.6.5 响应寄存器(R32_SDIO_RESI)

偏移地址：0x10

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

位	名称	访问	描述	复位值
[31:6]	Reserved	RO	保留。	0
[5:0]	RESI	RO	这个域记录了收到的响应的索引值。	000000b

28.6.6 响应参数寄存器（R128_SDIO_RESX）

28.6.6.1 响应参数寄存器高 32 位（R128_SDIO_RESX[127:96]）

偏移地址：0x10

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

28.6.6.2 响应参数寄存器次高 32 位（R128_SDIO_RESX[95:64]）

偏移地址：0x10

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

28.6.6.3 响应参数寄存器次低 32 位（R128_SDIO_RESX[63:32]）

偏移地址：0x10

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

28.6.6.4 响应参数寄存器低 32 位（R128_SDIO_RESX[31:0]）

偏移地址：0x10

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

位	名称	访问	描述	复位值
[128:0]	RESP	RO	当响应为长响应时，整 128 位均表示卡状态；当响应为短响应时，低 32 位表示卡状态。SDIO 外设先收到卡状态的最高位，并从 R128_SDIO_RESX 最低位开始存储。	0

28.6.7 数据定时寄存器(R32_SDIO_DTIM)

偏移地址：0x24

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

位	名称	访问	描述	复位值
[31:0]	DTIM	RW	数据超时时长。以 SDIO_CK 的周期数为单位。	0

28.6.8 传输长度寄存器（R32_SDIO_TLEN）

偏移地址：0x28

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

位	名称	访问	描述	复位值
[31:25]	Reserved	RO	保留。	0
[24:0]	TLEN	RW	传输数据长度域。此域的值在开启传输时会被加载到传输计数器。对于块传输，此域的值必须是块大小的整数倍，块大小由 SDIO 设备定义，存储在 R32_SDIO_DCTLR[7:4]中，常见值为 512B 等。	0

28.6.9 数据控制寄存器（R32_SDIO_DCTLR）

偏移地址：0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SDEN	RWMD	RWSTP	RWST	BLKLEN				DMAEN	TMOD	TDIR	TEN

位	名称	访问	描述	复位值
[31:12]	Reserved	RO	保留。	0
[11]	SDEN	RW	SDIO 使能位。此位被置位后，DPSM 才能进行运转。	0
[10]	RWMD	RW	读等待模式。 0：停止 SDIO_CK 控制读等待； 1：使用 SDIO_D2 控制读等待；	0
[9]	RWSTP	RW	读等待停止位。如果 RWST 位被置位，那么读等待将被停止。	0
[8]	RWST	RW	读等待开始位。置此位将执行读等待操作。	0

[7:4]	BLKLEN	RW	数据块长度域。此域存储了数据块的长度，采用块传输前必须定义块传输长度。此域可以写入的值为 0 到 1110b 之间，表示的块传输长度为 2^{BLKLEN}，即 0 到 16384 字节之间。	0
[3]	DMAEN	RW	DMA 使能位。置此位使能 DMA。	0
[2]	TMOD	RW	传输模式设置位。置此位设置传输模式。 0：块传输； 1：流传输；	0
[1]	TDIR	RW	传输方向设置位。置此位设置传输方向。 0：控制器到卡； 1：卡到控制器；	0
[0]	TEN	RW	传输使能位。置此位开始数据传输。 具体流程为，置此位后，DPSM 进入 Wait_S 或 Wait_R 的流程（取决于传输方向），	0

28.6.10 传输计数寄存器（R32_SDIO_DCNT）

偏移地址：0x30

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

位	名称	访问	描述	复位值
[31:25]	Reserved	RO	保留。	0
[24:0]	DCNT	RO	传输数据计数器域。在启动传输时，发送长度寄存器的值将会被加载到此计数器中，并随着传输过程递减。	0

28.6.11 状态寄存器（R32_SDIO_SR）

偏移地址：0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CEED	SDIT	RXVA	TXVA	RXEM	TXEM	RXFL	TXFL
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXHF	TXHE	RXACT	TXACT	CMDACT	BCCE	NSDW	DD	CS	CR	RXO	TUO	DTO	CTO	BCEE	CREE

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23]	CEED	RO	此位被置位时,CMD61 接收到 CE-ATA 完成信号。	0
[22]	SDIT	RO	此位被置位时，SDIO 收到设备中断。	0
[21]	RXVA	RO	此位被置位时，接收 FIFO 数据可用。	0
[20]	TXVA	RO	此位被置位时，发送 FIFO 中的数据可用。	0
[19]	RXEM	RO	此位被置位时，接收 FIFO 空。	0
[18]	TXEM	RO	此位被置位时，发送 FIFO 空。	0

[17]	RXFL	RO	此位被置位时，接收 FIFO 满。	0
[16]	TXFL	RO	此位被置位时，发送 FIFO 满。	0
[15]	RXHF	RO	此位被置位时，接收 FIFO 半满。	0
[14]	TXHE	RO	此位被置位时，发送 FIFO 半空。	0
[13]	RXACT	RO	此位被置位时，正在接收数据。	0
[12]	TXACT	RO	此位被置位时，正在发送数据。	0
[11]	CMDACT	RO	此位被置位时，正在传输命令。	0
[10]	BCCE	RO	此位被置位时，已经发送或者接受了数据块且 CRC 校验通过。	0
[9]	NSDW	RO	此位被置位时，在宽总线模式下，所有数据线都没有检测到起始信号。	0
[8]	DD	RO	此位被置位时，数据传输结束（传输计数器为零）。	0
[7]	CS	RO	此位被置位时，命令已经发送出去。	0
[6]	CR	RO	此位被置位时，已经收到响应，CRC 检验成功。	0
[5]	RXO	RO	此位被置位时，接收 FIFO 上溢。	0
[4]	TUO	RO	此位被置位时，发送 FIFO 下溢。	0
[3]	CTO	RO	此位被置位时，数据超时。	0
[2]	CTO	RO	此位被置位时，命令超时超过了 64 个 SDIO_CK 周期。	0
[1]	BCEE	RO	此位被置位时，已经发送或者接受了数据块但 CRC 校验失败。	0
[0]	CREE	RO	此位被置位时，已经收到响应，但 CRC 检验失败。	0

28.6.12 中断清除寄存器(R32_SDIO_ICR)

偏移地址：0x38

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BCCEC	NSDWC	DDC	CSC	CRC	RXOC	TUOC	CTOC	CTOC	BCEEC	CREEC

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23]	CEEDC	RW	置此位清除状态寄存器中的 CEED 位。	0
[22]	SDITC	RW	置此位清除状态寄存器中的 SDIT 位。	0
[21:11]	Reserved	RW	保留。	0
[10]	BCCEC	RW	置此位清除状态寄存器中的 BCCE 位。	0
[9]	NSDWC	RW	置此位清除状态寄存器中的 NSDW 位。	0
[8]	DDC	RW	置此位清除状态寄存器中的 DD 位。	0
[7]	CSC	RW	置此位清除状态寄存器中的 CS 位。	0
[6]	CRC	RW	置此位清除状态寄存器中的 CR 位。	0
[5]	RXOC	RW	置此位清除状态寄存器中的 RXO 位。	0
[4]	TUOC	RW	置此位清除状态寄存器中的 TUO 位。	0

[3]	CTOC	RW	置此位清除状态寄存器中的 CTO 位。	0
[2]	CTOC	RW	置此位清除状态寄存器中的 CTO 位。	0
[1]	BCEEC	RW	置此位清除状态寄存器中的 BCEE 位。	0
[0]	CREEC	RW	置此位清除状态寄存器中的 CREE 位。	0

28.6.13 中断使能寄存器(R32_SDIO_IER)

偏移地址：0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CEEDE	SDITE	RXVAE	TXVAE	RXEME	TXEME	RXFLE	TXFLE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXHFE	TXHEE	RXACTE	TXACTE	CMDACTE	BCCEE	NSDWE	DDE	CSE	CRE	RXOE	TUOE	DTOE	CTOE	BCEEE	CREEE

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23]	CEEDE	RW	置此位后，状态寄存器在置 CEED 位时将产生中断。	0
[22]	SDITE	RW	置此位后，状态寄存器在置 SDIT 位时将产生中断。	0
[21]	RXVAE	RW	置此位后，状态寄存器在置 RXVA 位时将产生中断。	0
[20]	TXVAE	RW	置此位后，状态寄存器在置 TXVA 位时将产生中断。	0
[19]	RXEME	RW	置此位后，状态寄存器在置 RXEM 位时将产生中断。	0
[18]	TXEME	RW	置此位后，状态寄存器在置 TXEM 位时将产生中断。	0
[17]	RXFLE	RW	置此位后，状态寄存器在置 RXFL 位时将产生中断。	0
[16]	TXFLE	RW	置此位后，状态寄存器在置 TXFL 位时将产生中断。	0
[15]	RXHFE	RW	置此位后，状态寄存器在置 RXHF 位时将产生中断。	0
[14]	TXHEE	RW	置此位后，状态寄存器在置 TXHE 位时将产生中断。	0
[13]	RXACTE	RW	置此位后,状态寄存器在置 RXACT 位时将产生中断。	0
[12]	TXACTE	RW	置此位后,状态寄存器在置 TXACT 位时将产生中断。	0
[11]	CMDACTE	RW	置此位后,状态寄存器在置 CMDACT 位时将产生中断。	0
[10]	BCCEE	RW	置此位后，状态寄存器在置 BCCE 位时将产生中断。	0
[9]	NSDWE	RW	置此位后，状态寄存器在置 NSDW 位时将产生中断。	0
[8]	DDE	RW	置此位后，状态寄存器在置 DD 位时将产生中断。	0
[7]	CSE	RW	置此位后，状态寄存器在置 CS 位时将产生中断。	0
[6]	CRE	RW	置此位后，状态寄存器在置 CR 位时将产生中断。	0
[5]	RXOE	RW	置此位后，状态寄存器在置 RXO 位时将产生中断。	0
[4]	TUOE	RW	置此位后，状态寄存器在置 TUO 位时将产生中断。	0
[3]	CTOE	RW	置此位后，状态寄存器在置 CTO 位时将产生中断。	0
[2]	CTOE	RW	置此位后，状态寄存器在置 CTO 位时将产生中断。	0
[1]	BCEEE	RW	置此位后，状态寄存器在置 BCEE 位时将产生中断。	0
[0]	CREEE	RW	置此位后，状态寄存器在置 CREE 位时将产生中断。	0

28.6.14 FIFO 计数寄存器(R32_SDIO_FIFOCNT)

偏移地址：0x48

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

位	名称	访问	描述	复位值
[31:0]	FIFOCNT	RO	FIFO 包含还未写入 FIFO 或者还未从 FIFO 读出的字（32bit）数。在设置 R32_SDIO_DCTLR：R32_SDIO_DCTLR 时，如果 DPSM 空闲，FIFO 计数器将从 R32_SDIO_TLEN 中加载传输长度值，如果此值不能被 4 整除，则最后的 1 至 3 个字节会被当做一个字处理。	0

28.6.15 FIFO(R32_SDIO_FIFO)

偏移地址：0x80

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

位	名称	访问	描述	复位值
[31:0]	FIFO	RW	FIFO 数据域。此域即为 FIFO 的数据。读写此域将读出接收到的数据或者发送待发的数据。SDIO 的 FIFO 共 32 个字（一个字为 32 位）。	0

第 29 章 随机数发生器（RNG）

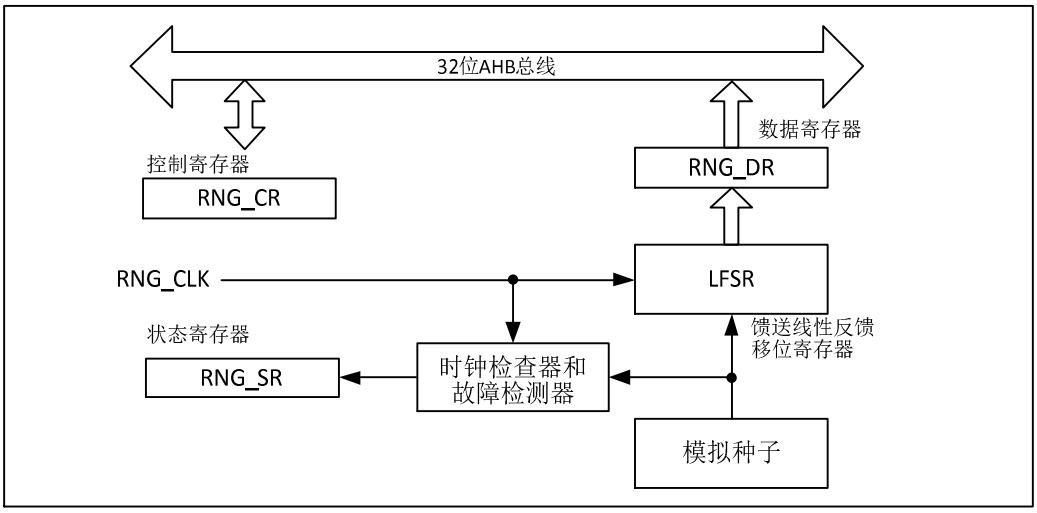
本章模块描述适用于 CH32F2x 和 CH32V3x 微控制器系列部分产品。

随机数发生器（RNG），以连续模拟噪声为基础，在主机读取数据时提供 32bit 的随机数。

29.1 主要特性

- 可产生 32bit 随机数
- 可实现错误管理
- 可被单独禁止，降低功耗

图 29-1 RNG 模块框图



29.2 功能描述

随机发送器采用模拟电路实现，该电路产生线性反馈移位寄存器（RNG_LFSR）的种子，用于生成 32 位随机数。RNG_LFSR 由专用时钟（PLL48CLK）按照恒定频率提供时钟信息，故随机数质量与 HCLK 时钟有关。当有大量种子引入 RNG_LFSR 后，RNG_LFSR 的内容会传入数据寄存器（RNG_DR）。

29.2.1 RNG 操作

RNG 具体操作步骤如下：

1） 若使能中断，需通过将 RNG_CR 寄存器中的 IE 位置 1（当准备好随机数或出现错误时产生该中断）。
2） 通过配置 RNG_CR 寄存器的 RNGEN 位使能随机数产生，同时激活模拟部分、RNG_LFSR 和错误检测器。
3） 若使能中断，每次产生中断时，通过查询 RNG_SR 寄存器中的 SEIS 和 CEIS 位为 0 确定未出现错误且 DRDY 为为 1 确定随机数已准备就绪。之后即可读取 RNG_DR 寄存器中的内容。

29.2.2 错误管理

RNG 出错包括时钟错误和种子错误。当出现时钟错误时，RNG 无法再产生随机数，此时需检查时钟控制器是否正确配置，是否可提供 RNG 时钟，然后将 CEIS 位清零。当 CECS 位为 0 时，RNG 可正常工作。当产生时钟错误时，对上一个随机数没有影响，可正常使用。当出现种子错误时，此时 RNG_DR 寄存器中的值不可使用该随机数，若用重新使用 RNG，需要先将 SEIS 位清零，然后将 RNGEN 位清零并置 1，重新初始化和重新启动 RNG。

29.3 寄存器描述

表 29-1 OPA 相关寄存器列表

名称	访问地址	描述	复位值
R32_RNG_CR	0x40023C00	RNG 控制寄存器	0x00000000
R32_RNG_SR	0x40023C04	RNG 状态寄存器	0x00000000
R32_RNG_DR	0x40023C08	RNG 数据寄存器	0x00000000

29.3.1 RNG 控制寄存器（RNG_CR）

偏移地址：0x00

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

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

位	名称	访问	描述	复位值
[31:4]	Reserved	RO	保留。	0
3	IE	RW	中断使能控制 0：禁止 RNG 中断 1：使能 RNG 中断	0
2	RNGEN	RW	随机数发生器使能 0：禁止随机数发生器 1：使能随机数发生器	0
[1:0]	Reserved	RW	保留。	0

29.3.2 RNG 状态寄存器（RNG_SR）

偏移地址：0x04

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									SEIS	CEIS			SECS	CECS	DRDY

位	名称	访问	描述	复位值
[31:7]	Reserved	RO	保留。	0
6	SEIS	RW	种子错误中断状态(此位与 SECS 同时设置) 0：未检测倒错误序列 1：检测到以下错误序列之一： -超过 64 个相同连续位； -超过 32 个连续交替的 0 和 1；	0
5	CEIS	RW	时钟错误中断状态(此位与 CECS 同时设置) 0：正确检测到 PLL48CLK 时钟	0

			1：未正确检测到 PLL48CLK 时钟	
[4:3]	MODE4	RW	保留。	0
2	SECS	RO	种子错误当前状态 0：未检测出错误序列 1：检测到以下错误序列之一： -超过 64 个相同连续位； -超过 32 个连续交替的 0 和 1；	0
1	CECS	RO	时钟错误当前状态 0：正确检测到 PLL48CLK 时钟 1：未正确检测到 PLL48CLK 时钟	0
0	DRDY	RO	数据就绪（读取 RNG_DR 寄存器后，该位清 0） 0：RNG_DR 寄存器无效，此随机数不可用 1：RNG_DR 寄存器有效，此随机数可用	0

29.3.3 RNG 状态寄存器（RNG_DR）

偏移地址：0x08

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

位	名称	访问	描述	复位值
[31:0]	RNDATA	RO	32 位随机数	0

第 30 章 运算放大器（OPA）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器系列部分产品。

运算放大器模块（OPA），包含 4 个可独立配置的运算放大器。每个运算放大器的输入和输出均连接至 I/O 口，且输入引脚可选择，输出引脚可选择配置到通用 I/O 口或复用为 ADC 采样通道的 I/O。

30.1 主要特性

- 输入引脚可选择
- 输出引脚可选择通用 I/O 口或 ADC 采样通道

30.2 功能描述

置位 OPAx_EN，即可使能对应的 OPAx，配置 OPAx_MODE 可选择 OPAx 的输出通道为 ADC 采样通道或者普通 I/O 口，配置 OPAx_PSEL，可选择 OPAx 的正向输入引脚，配置 OPAx_NSEL，可选择 OPAx 的负向输入引脚。

注：各 OPA 详细的输入输出引脚，参考数据手册中引脚说明。

30.3 寄存器描述

表 30-1 OPA 相关寄存器列表

名称	访问地址	描述	复位值
R32_OPA_CTLR	0x40023804	OPA 配置寄存器	0x00000000

30.3.1 OPA 配置寄存器（OPA_CTLR）

偏移地址：0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSEL4	NSEL4	MODE4	EN4	PSEL3	NSEL3	MODE3	EN3	PSEL2	NSEL2	MODE2	EN2	PSEL1	NSEL1	MODE1	EN1

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
15	PSEL4	RW	OPA4 正向输入端选择 0：CHP0 1：CHP1 *注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。*	0
14	NSEL4	RW	OPA4 负向输入端选择 0：CHN0 1：CHN1 *注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。*	0

13	MODE4	RW	OPA4 输出通道选择 0：输出通道为 ADC 采样通道 1：输出通道为 I/O 口 *注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。*	0
12	EN4	RW	OPA4 使能 0：禁止 OPA4 1：使能 OPA4 *注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。*	0
11	PSEL3	RW	OPA3 正向输入端选择 0：CHP0 1：CHP1 *注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。*	0
10	NSEL3	RW	OPA3 负向输入端选择 0：CHN0 1：CHN1 *注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。*	0
9	MODE3	RW	OPA3 输出通道选择 0：输出通道为 ADC 采样通道 1：输出通道为 I/O 口 *注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。*	0
8	EN3	RW	OPA3 使能 0：禁止 OPA3 1：使能 OPA3 *注：适用于 CH32F20x_D8、CH32F20x_D8C、CH32V30x_D8、CH32V30x_D8C。*	0
7	PSEL2	RW	OPA2 正向输入端选择 0：CHP0 1：CHP1	0
6	NSEL2	RW	OPA2 负向输入端选择 0：CHN0 1：CHN1	0
5	MODE2	RW	OPA2 输出通道选择 0：输出通道为 ADC 采样通道 1：输出通道为 I/O 口	0
4	EN2	RW	OPA2 使能 0：禁止 OPA2 1：使能 OPA2	0
3	PSEL1	RW	OPA1 正向输入端选择 0：CHP0 1：CHP1	0
2	NSEL1	RW	OPA1 负向输入端选择	0

			0：CHN0 1：CHN1	
1	MODE1	RW	OPA1 输出通道选择 0：输出通道为 ADC 采样通道 1：输出通道为 I/O 口	0
0	EN1	RW	OPA1 使能 0：禁止 OPA1 1：使能 OPA1	0

第 31 章 电子签名（ESIG）

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

电子签名包含了芯片识别信息：闪存区容量和唯一身份标识。它由厂家在出厂时烧录到存储器模块的系统存储区域，可以通过 SWD（SDI）或者应用代码读取。

31.1 功能描述

闪存区容量：指示当前芯片用户应用程序可以使用大小。

唯一身份标识：96 位二进制码，对任意一个微控制器都是唯一的，用户只能读访问不能修改。此唯一标识信息可以用作微控制器（产品）的安全密码、加解密钥、产品序列号等，用来提高系统安全机制或表明身份信息。

以上内容用户都可以按 8/16/32 位进行读访问。

31.2 寄存器描述

表 31-1 ESIG 相关寄存器列表

名称	访问地址	描述	复位值
R16_ESIG_FLACAP	0x1FFFF7E0	闪存容量寄存器	0xXXXX
R32_ESIG_UNIID1	0x1FFFF7E8	UID 寄存器 1	0xXXXXXXXX
R32_ESIG_UNIID2	0x1FFFF7EC	UID 寄存器 2	0xXXXXXXXX
R32_ESIG_UNIID3	0x1FFFF7F0	UID 寄存器 3	0xXXXXXXXX

31.2.1 闪存容量寄存器（ESIG_FLACAP）

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

位	名称	访问	描述	复位值
[15:0]	FLASHSIZE	RO	以 Kbyte 为单位的闪存容量。 例：0x0080 = 128 K 字节	X

31.2.2 UID 寄存器（ESIG_UNIID1）

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

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

位	名称	访问	描述	复位值
[31:0]	U_ID	RO	UID 的 0-31 位。	X

31.2.3 UID 寄存器（ESIG_UNIID2）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[47:32]															

位	名称	访问	描述	复位值
[31:0]	U_ID	RO	UID 的 32-63 位。	X

31.2.4 UID 寄存器（ESIG_UNIID3）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[79:64]															

位	名称	访问	描述	复位值
[31:0]	U_ID	RO	UID 的 64-95 位。	X

第 32 章 闪存及用户选择字(FLASH)

本章模块描述适用于 CH32F2x、CH32V2x 和 CH32V3x 微控制器全系列产品。

32.1 闪存组织

芯片内部闪存组织结构如下（以 xVCT6 为例）：

表 32-1 闪存组织结构

块	名称	地址范围	大小（字节）
主存储器	页 0	0x08000000 - 0x080000FF	256
	页 1	0x08000100 - 0x080001FF	256
	页 2	0x08000200 - 0x080002FF	256
	页 3	0x08000300 - 0x080003FF	256
	页 4	0x08000400 - 0x080004FF	256
	页 5	0x08000500 - 0x080005FF	256
	页 6	0x08000600 - 0x080006FF	256
	页 7	0x08000700 - 0x080007FF	256
	…	…	…
	页 1919	0x08077EFF - 0x08077FFF	256
信息块	系统引导代码存储	0x1FFF8000 - 0x1FFFEFFF	28K
	用户选择字	0x1FFFF800 - 0x1FFFF87F	128

注：

1） 上述主存储器区域用于用户的应用程序存储，以 4K 字节（16 页）单位进行写保护划分；除了“厂商配置字”区域出厂锁定，用户不可访问，其他区域在一定条件下用户可操作。

2） 在主频超过 100MHz 时，操作 FLASH 时需注意：
在进行非零等待区域 FLASH 和零等待区域 FLASH、用户字读写以及厂商配置字和 Boot 区域读时，需做以下操作，首先将 HCLK 进行 2 分频，FLASH 操作完成后再恢复，保证 FLASH 操作是频率低于 100Mhz，具体操作可参考相关 FLASH 操作例程。

32.2 闪存编程及安全性

32.2.1 两种编程/擦除方式

- 标准编程：此方式是默认编程方式（兼容方式）。这种模式下 CPU 以单次 2 字节方式执行编程，单次 4K 字节执行擦除及整片擦除操作。
- 快速编程：此方式采用页操作方式（推荐）。经过特定序列解锁后，执行单次 256 字节的编程及 256 字节擦除、32K 字节擦除、64K 字节擦除及整片擦除。

32.2.2 安全性-防止非法访问（读、写、擦）

- 页写入保护
- 读保护

芯片处于读保护状态下时：

1） 主存储器 0-15 页（4K 字节）自动写保护状态，不受 FLASH_WPR 寄存器控制；解除读保护状态，所有主存储页都由 FLASH_WPR 寄存器控制。

2） 系统引导代码区、SWD 或 SDI 模式、RAM 区域都不可对主存储器进行擦除或编程，整片擦除除外。

可擦除或编程用户选择字区域。如果试图解除读保护（编程用户字），芯片将自动擦除整片用户区。

注：进行闪存的编程/擦除操作时，必须打开内部 RC 振荡器（HSI）。

32.3 FLASH 增强读模式

FLASH 增强读模式适用于用户程序运行在 FLASH 中（用户代码空间超过用户选择字 SRAM_CODE_MODE[1:0]位配置的 CODE 大小空间），开启该模式，可提高 FLASH 访问效率。开启该模式需将 FLASH_CTLR 寄存器的 ENHANCE_MOD 位置 1,关闭该模式需先将 ENHANCE_MOD 位清 0,再将 RSEN_ACT 置 1。同时可通过配置 FLASH_CTLR 寄存器的 SCK_MOD 位选择访问时钟频率。

注：在使用 FLASH 增强读模式，需注意以下几点：

1） 在对 FLASH 进行任何模式的擦除或编程（包括解除读保护等用户字编程）等操作之前，须先退出增强读模式，否者会导致擦除和编程操作失败；

2） 在进入停止模式之前，须先退出增强读模式，否者可能导致停止模式异常；

3） 在电源复位和系统复位结束后，芯片由硬件控制自动退出增强读模式。

32.4 寄存器描述

表 32-2 FLASH 相关寄存器列表

名称	访问地址	描述	复位值
R32_FLASH_KEYR	0x40022004	FPEC 键寄存器	X
R32_FLASH_OBKEYR	0x40022008	OBKEY 寄存器	X
R32_FLASH_STATR	0x4002200C	状态寄存器	0x00000000
R32_FLASH_CTLR	0x40022010	配置寄存器	0x00000080
R32_FLASH_ADDR	0x40022014	地址寄存器	0x00000000
R32_FLASH_OBR	0x4002201C	选择字寄存器	0x03FFFFFC
R32_FLASH_WPR	0x40022020	写保护寄存器	0xFFFFFFF
R32_FLASH_MODEKEYR	0x40022024	扩展键寄存器	X

32.4.1 FPEC 键寄存器（FLASH_KEYR）

偏移地址：0x04

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

位	名称	访问	描述	复位值
[31:0]	KEYR	WO	FPEC 键，用于输入 FPEC 的解锁键包括： RDPRT 键 = 0x000000A5； KEY1 = 0x45670123； KEY2 = 0xCDEF89AB。	X

32.4.2 OBKEY 寄存器（FLASH_OBKEYR）

偏移地址：0x08

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

位	名称	访问	描述	复位值
[31:0]	OBKEYR	WO	选择字键，用于输入选择字键解除 OPTWRE。	X

32.4.3 状态寄存器（FLASH_STATR）

偏移地址：0x0C

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

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
7	ENHANCE_MOD_STA	RO	FLASH 增强读模式是否开启位： 1：FLASH 增强读模式开启 0：FLASH 增强读模式关闭	0
6	Reserved	RO	保留。	0
5	EOP	RW1	指示操作结束，写 1 清零。 每次成功擦除或编程时，硬件会置位。	0
4	WRPRTERR	RW1	指示写保护错误，写 1 清零。 如果对写保护的地址编程时，硬件会置位。	0
3	Reserved	RO	保留。	0
2	Reserved	RO	保留。	0
1	WR_BSY	RO	该位在快速页编程时使用，指示编程数据正在写入。在页编程时，当写入数据，该位被设置‘1’，硬件自动清‘0’；如果该位为‘0’，表示允许写入下个数据。	0
0	BSY	RO	指示忙状态： 1：表示闪存操作正在进行； 0：操作结束。	0

注：进行编程操作时，需要确定 FLASH_CTLR 寄存器的 STRT 位为 0。

32.4.4 配置寄存器（FLASH_CTLR）

偏移地址：0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						SCK_MOD	ENHANCE_MOD	Reserved	RSEN_ACT	PG_START	Reserved	BER64	BER32	PAGE_ER	PAGE_PG
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLOCK	Reserved		EOPIE	Reser	ERRIE	OBWRE	Reser	LOCK	STRT	OBER	OBPG	Rese	MER	SER	PG

			ved			ved					rved			

位	名称	访问	描述	复位值
[31:26]	Reserved	RO	保留。	0
25	SCK_MOD	RW	FLASH 访问时钟配置： 1:FLASH 访问时钟频率=系统时钟(SYSCLK)； 0：FLASH 访问时钟频率=系统时钟一半（SYSCLK/2）； *注：FLASH 访问时钟频率不超过 72MHZ。*	0
24	ENHANCE_MOD	RW	FLASH 增强读模式： 该模式，在程序运行在 FLASH 时，可提高访问效率。 1：使能 FLASH 增强读模式； 0：关闭 FLASH 增强读模式，需配合 RSEN_ACT 位一起操作，退出步骤先将 ENHANCE_MOD 位清 0，再将 RSEN_ACT 置 1。	0
23	Reserved	RO	保留。	0
22	RSEN_ACT	WO	退出增强读模式，硬件自动清除，需配合 ENHANCE_MOD 位一起操作，退出步骤先将 ENHANCE_MOD 位清 0，再将 RSEN_ACT 置 1。	0
21	PG_START	RWO	开始。置 1 启动一次页编程，硬件自动清除。	0
20	Reserved	RO	保留。	0
19	BER64	RW	执行 64KB 擦除。	0
18	BER32	RW	执行 32KB 擦除。	0
17	PAGE_ER	RW	执行快速页（256Byte）擦除操作。	0
16	PAGE_PG	RW	执行快速页编程操作。	0
15	FLOCK	RW1	快速编程锁。只能写‘1’。当该位为‘1’时表示快速编程/擦除模式不可用。在检测到正确的解锁序列后，硬件清除此位为‘0’。软件置 1，重新加锁。	1
[14:13]	Reserved	RO	保留。	0
12	EOPIE	RW	操作完成中断控制（FLASH_STATR 寄存器中 EOP 置位）： 1：允许产生中断； 0：禁止产生中断。	0
11	Reserved	RO	保留。	0
10	ERRIE	RW	错误状态中断控制（FLASH_STATR 寄存器中 PGERR/WRPRTERR 置位）： 1：允许产生中断； 0：禁止产生中断。	0
9	OBWRE	RWO	用户选择字锁，软件清 0： 1：表示可以对用户选择字进行编程操作。需要在 FLASH_OBKEYR 寄存器中写入正确序列后由硬件置位。 0：软件清零后重新加锁用户选择字。	0

8	Reserved	RO	保留。	0
7	LOCK	RW1	锁。只能写‘1’。当该位为‘1’时表示 FPEC 和 FLASH_CTLR 被锁住不可写。在检测到正确的解锁序列后，硬件清除此位为‘0’。 在一次不成功的解锁操作后，直到下次系统复位前，该位不会再改变。	1
6	STRT	RW1	开始。置 1 启动一次擦除动作，硬件自动清 0（BSY 变‘0’）。	0
5	OBER	RW	执行用户选择字擦除	0
4	OBPG	RW	执行用户选择字编程	0
3	Reserved	RO	保留。	0
2	MER	RW	执行全擦除操作（擦除整个用户区）。	0
1	PER	RW	执行标准页（4KB）擦除操作。	0
0	PG	RW	执行标准编程操作。	0

32.4.5 地址寄存器（FLASH_ADDR）

偏移地址：0x14

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

位	名称	访问	描述	复位值
[31:0]	ADDR	WO	闪存地址，进行编程时为编程的地址，进行擦除时为擦除的起始地址。 当 FLASH_SR 寄存器中的 BSY 位为‘1’时，不能写此寄存器。	0

32.4.6 选择字寄存器（FLASH_OBR）

偏移地址：0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						USER Reserved		POR CTR	USBD PU	USBD MODE	STANDY RST	STOP RST	IWDG SW	RDPRT	OBERR

位	名称		访问	描述	复位值
[31:10]	Reserved		RO	保留。	0
[9:8]	USER	SRAM_CODE_MODE	RO	00：CODE-192KB + RAM-128KB 01：CODE-224KB + RAM-96KB 10：CODE-256KB + RAM-64KB 11：CODE-288KB + RAM-32KB	X

				注：适用于 CH32V303RC、CH32V303VC、CH32V307RC、CH32V307WC、CH32V307VC、CH32F203RC、CH32F203VC、CH32F207VC。 00：CODE-128KB + RAM-64KB 01：CODE-144KB + RAM-48KB 1x：CODE-160KB + RAM-32KB *注：适用于 CH32V20x_D8W、CH32V20x_D8、CH32F20x_D8W。*	
[7:5]		Reserved	RO	未使用。	X
4		STANDY_RST	RO	待机模式下系统复位控制。	X
3		STOP_RST	RO	停止模式下系统复位控制。	X
2		IWDG_SW	RO	独立看门狗（IWDG）硬件使能位。	1
1	RDPRT		RO	读保护状态。 1：表示闪存当前读保护有效。	1
0	OBERR		RO	选择字错误。 1：表示选择字和它的反码不匹配。	0

注：USER 和 RDPRT 在系统复位后从用户选择字区域加载。

32.4.7 写保护寄存器（FLASH_WPR）

偏移地址：0x20

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

位	名称	访问	描述	复位值
[31:0]	WPR	RO	闪存写保护状态。 1：写保护失效； 0：写保护有效。 每个比特位代表 4K 字节（16 页）存储写保护状态。	X

注：WPR 在系统复位后从用户选择字区域加载。

32.4.8 扩展键寄存器（FLASH_MODEKEYR）

偏移地址：0x24

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

位	名称	访问	描述	复位值
[31:0]	MODEKEYR	WO	输入下面序列解锁快速编程/擦除模式： KEY1 = 0x45670123；	X

			KEY2 = 0xCDEF89AB。	

32.5 闪存操作流程

32.5.1 读操作

在通用地址空间内进行直接寻址，任何 8/16/32 位数据的读操作都能访问闪存模块的内容并得到相应的数据。

32.5.2 解除闪存锁

系统复位后，闪存控制器（FPEC）和 FLASH_CTLR 寄存器是被锁定的，不可访问。通过写入序列到 FLASH_KEYR 寄存器可解锁闪存控制器模块。

解锁序列：

1） 向 FLASH_KEYR 寄存器写入 KEY1 = 0x45670123（第 1 步必须是 KEY1）；

2） 向 FLASH_KEYR 寄存器写入 KEY2 = 0xCDEF89AB（第 2 步必须是 KEY2）。

上述操作必须按序并连续执行，否则属于错误操作，会锁死 FPEC 模块和 FLASH_CTLR 寄存器并产生总线错误，直到下次系统复位。

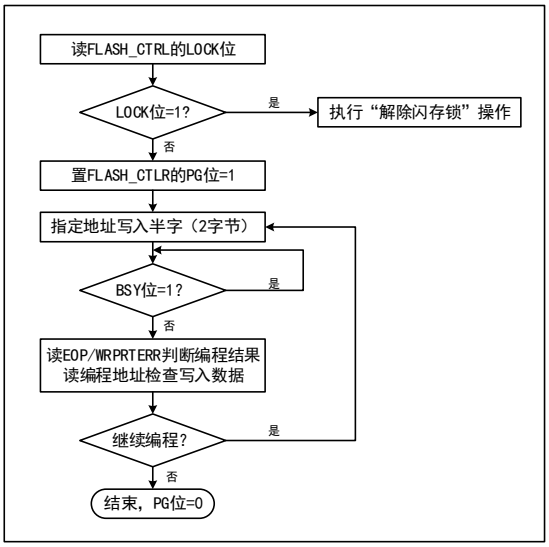
闪存控制器（FPEC）和 FLASH_CTLR 寄存器可以通过将 FLASH_CTLR 寄存器的“LOCK”位，置 1 来再次锁定。

32.5.3 主存储器标准编程

标准编程每次可以写入 2 字节。当 FLASH_CTLR 寄存器的 PG 位为‘1’时，每次向闪存地址写入半字（2 字节）将启动一次编程，写入任何非半字数据，FPEC 都会产生总线错误。编程过程中，BSY 位为‘1’，编程结束，BSY 位为‘0’，EOP 位为‘1’。

注：当 BSY 位为‘1’时，将禁止对任何寄存器执行写操作。

图 32-1 FLASH 编程



1）检查 FLASH_CTLR 寄存器 LOCK，如果为 1，需要执行“解除闪存锁”操作。

2）设置 FLASH_CTLR 寄存器的 PG 位为‘1’，开启标准编程模式。

3）向指定闪存地址（偶地址）写入要编程的半字。

4）等待 BSY 位变为‘0’或 FLASH_STATR 寄存器的 EOP 位为‘1’表示编程结束，将 EOP 位清 0。

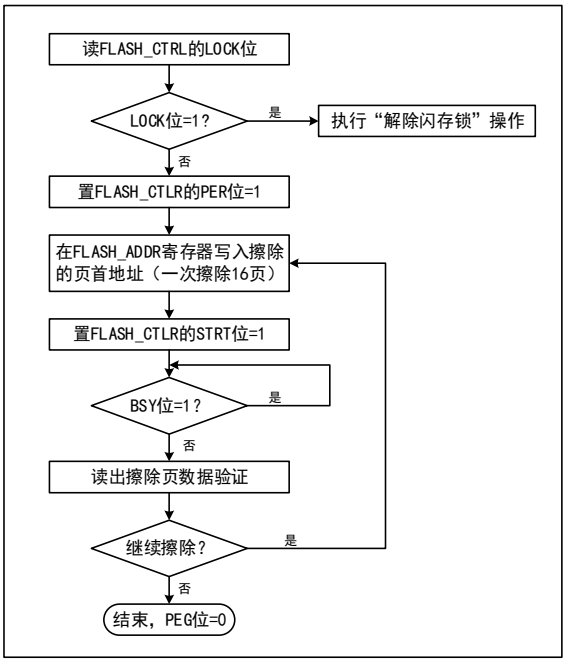
5）查询 FLASH_STATR 寄存器看是否有错误，或者读编程地址数据校验。

6）继续编程可以重复 3-5 步骤，结束编程将 PG 位清 0。

32.5.4 主存储器标准擦除

闪存可以按标准页（4K 字节）擦除，也可以整片擦除。

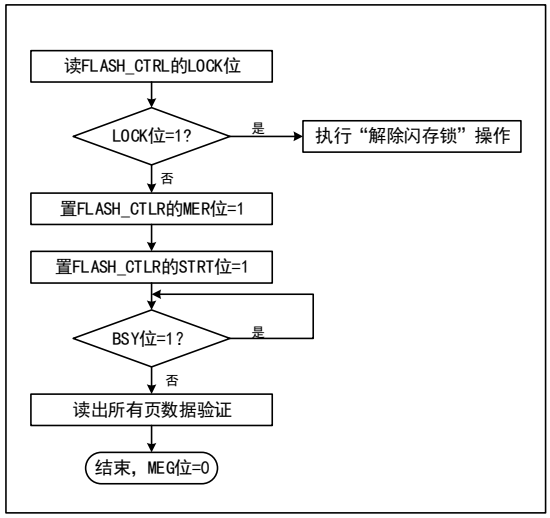
图 32-2 FLASH 页擦除



1）检查 FLASH_CTLR 寄存器 LOCK 位，如果为 1，需要执行“解除闪存锁”操作。
2）设置 FLASH_CTLR 寄存器的 PER 位为‘1’，开启标准页擦除模式。
3）向 FLASH_ADDR 寄存器写入选择擦除的页首地址。
4）设置 FLASH_CTLR 寄存器的 STAT 位为‘1’，启动一次擦除动作。
5）等待 BYS 位变为‘0’或 FLASH_STATR 寄存器的 EOP 位为‘1’表示擦除结束，将 EOP 位清 0。
6）读擦除页的数据进行校验。
7）继续标准页擦除可以重复 3-5 步骤，结束擦除将 PEG 位清 0。

注：擦除成功后，字读- 0xe339e339，半字读- 0xe339，偶地址字节读- 0x39，奇地址读 0xe3。

图 32-3 FLASH 整片擦除



1）检查 FLASH_CTLR 寄存器 LOCK 位，如果为 1，需要执行“解除闪存锁”操作。
2）设置 FLASH_CTLR 寄存器的 MER 位为‘1’，开启整片擦除模式。
3）设置 FLASH_CTLR 寄存器的 STAT 位为‘1’，启动擦除动作。
4）等待 BYS 位变为‘0’或 FLASH_STATR 寄存器的 EOP 位为‘1’表示擦除结束，将 EOP 位清 0。
5）读擦除页的数据进行校验。
6）将 MER 位清 0。

32.5.5 快速编程模式解锁

通过写入序列到 FLASH_MODEKEYR 寄存器可解锁快速编程模式操作。解锁后，FLASH_CTLR 寄存器的 FLOCK 位将清 0，表示可以进行快速擦除和编程操作。通过将 FLASH_CTLR 寄存器的“FLOCK”位软件置 1 来再次锁定。

解锁序列：

1）向 FLASH_MODEKEYR 寄存器写入 KEY1 = 0x45670123；
2）向 FLASH_MODEKEYR 寄存器写入 KEY2 = 0xCDEF89AB。

上述操作必须按序并连续执行，否则属于错误操作会锁定，直到下次系统复位才能重新解锁。

注：快速编程操作需要解除“LOCK”和“FLOCK”两层锁定。

32.5.6 主存储器快速编程

快速编程按页（256 字节）进行编程。

1）检查 FLASH_CTLR 寄存器 LOCK 位，如果为‘1’，需要执行“解除闪存锁”操作。
2）检查 FLASH_CTLR 寄存器 FLOCK 位，如果为‘1’，需要执行“快速编程模式解锁”操作。
3）检查 FLASH_STATR 寄存器的 BSY 位，以确认没有其他正在进行的编程操作。
4）设置 FLASH_CTLR 寄存器的 PAGE_PG 位为‘1’，使能快速页编程模式。
5）使用 32 位方式向 FLASH 地址写入数据，例如
   （uint32_t）0x80000000 = 0x12345678;
6）等待 FLASH_STATR 寄存器的 WR_BSY 为‘0’，写入下个数据。
7）重复步骤 5-6 共 64 次。
8）设置 FLASH_CTLR 寄存器的 PG_START 位为‘1’，启动快速页编程。
9）等待 BSY 位变为‘0’或 FLASH_STATR 寄存器的 EOP 位为‘1’表示一次快速页编程完成，将 EOP 位清 0。
10）查询 FLASH_STATR 寄存器看是否有错误，或者读编程地址数据校验。
11）继续快速页编程可以重复 5-10 步骤，结束编程将 PAGE_PG 位清 0。

32.5.7 主存储器快速擦除

快速擦除按页（256 字节）进行擦除。

1）检查 FLASH_CTLR 寄存器 LOCK 位，如果为 1，需要执行“解除闪存锁”操作。
2）检查 FLASH_CTLR 寄存器 FLOCK 位，如果为 1，需要执行“快速编程模式解锁”操作。
3）检查 FLASH_STATR 寄存器的 BSY 位，以确认没有其他正在进行的编程操作。
4）设置 FLASH_CTLR 寄存器的 PAGE_ER 位为‘1’，开启快速页擦除（256 字节）模式功能。
5）向 FLASH_ADDR 寄存器写入快速擦除页的首地址。
6）设置 FLASH_CTLR 寄存器的 STAT 位为‘1’，启动一次快速页擦除（256 字节）动作。
7）等待 BSY 位变为‘0’或 FLASH_STATR 寄存器的 EOP 位为‘1’表示擦除结束，将 EOP 位清 0。
8）查询 FLASH_STATR 寄存器看是否有错误，或者读擦除页地址数据校验。
9）继续快速页擦除可以重复 5-8 步骤，结束擦除将 PAGE_ER 位清 0。

注：擦除成功后，字读-0xe339e339，半字读-0xe339，偶地址字节读-0x39，奇地址读 0xe3。

快速擦除按块（32K 字节）进行擦除。

1）检查 FLASH_CTLR 寄存器 LOCK 位，如果为 1，需要执行“解除闪存锁”操作。
2）检查 FLASH_CTLR 寄存器 FLOCK 位，如果为 1，需要执行“快速编程模式解锁”操作。
3）检查 FLASH_STATR 寄存器的 BSY 位，以确认没有其他正在进行的编程操作。
4）设置 FLASH_CTLR 寄存器的 BER32 位为‘1’，开启快速块擦除（32K 字节）模式功能。
5）向 FLASH_ADDR 寄存器写入快速擦除块的首地址。
6）设置 FLASH_CTLR 寄存器的 STAT 位为‘1’，启动一次快速块擦除（32K 字节）动作。
7）等待 BYS 位变为‘0’或 FLASH_STATR 寄存器的 EOP 位为‘1’表示擦除结束，将 EOP 位清 0。
8）查询 FLASH_STATR 寄存器看是否有错误，或者读擦除页地址数据校验。
9）继续快速页擦除可以重复 5-8 步骤，结束擦除将 BER32 位清 0。

注：擦除成功后，字读-0xe339e339，半字读-0xe339，偶地址字节读-0x39，奇地址读 0xe3。

快速擦除按块（64K 字节）进行擦除。

1）检查 FLASH_CTLR 寄存器 LOCK 位，如果为 1，需要执行“解除闪存锁”操作。
2）检查 FLASH_CTLR 寄存器 FLOCK 位，如果为 1，需要执行“快速编程模式解锁”操作。
3）检查 FLASH_STATR 寄存器的 BSY 位，以确认没有其他正在进行的编程操作。
4）设置 FLASH_CTLR 寄存器的 BER64 位为‘1’，开启快速块擦除（64K 字节）模式功能。
5）向 FLASH_ADDR 寄存器写入快速擦除块的首地址。
6）设置 FLASH_CTLR 寄存器的 STAT 位为‘1’，启动一次快速块擦除（64K 字节）动作。
7）等待 BYS 位变为‘0’或 FLASH_STATR 寄存器的 EOP 位为‘1’表示擦除结束，将 EOP 位清 0。
8）查询 FLASH_STATR 寄存器看是否有错误，或者读擦除页地址数据校验。
9）继续快速页擦除可以重复 5-8 步骤，结束擦除将 BER64 位清 0。

注：擦除成功后，字读-0xe339e339，半字读-0xe339，偶地址字节读-0x39，奇地址读 0xe3。

32.6 用户选择字

用户选择字固化在 FLASH 中，在系统复位后会被重新装载到相应寄存器，用户可以任意的进行擦除和编程。用户选择字信息块总共有 8 个字节（4 个字节为写保护，1 个字节为读保护，1 个字节为配置选项，2 个字节存储用户数据），每个位都有其反码位用于装载过程中的校验。下面描述了选择字信息结构和意义。

表 32-3 32 位选择字格式划分

[31:24]	[23:16]	[15:8]	[7:0]
选择字字节 1 反码	选择字字节 1	选择字字节 0 反码	选择字字节 0

表 32-4 用户选择字信息结构

地址 / 位	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFFF800	nUSER	USER	nRDPR	RDPR
0x1FFFF804	nData1	Data1	nData0	Data0
0x1FFFF808	nWRPR1	WRPR1	nWRPR0	WRPR0
0x1FFFF80C	nWRPR3	WRPR3	nWRPR2	WRPR2

名称/字节	描述	复位值

RDPR			读保护控制位，配置是否可以读出闪存中的代码。 0xA5：若此字节为 0xA5（nRDP 必须为 0x5A），表示当前代码处于非读保护状态，可以读出； 其他值：表示代码读保护状态，不可读，0-31 页（4K）将自动写保护，不受 WRPR0 控制。	0x01
USER	[7:6]	SRAM_CODE_MODE	00：CODE-192KB + RAM-128KB 01：CODE-224KB + RAM-96KB 10：CODE-256KB + RAM-64KB 11：CODE-288KB + RAM-32KB *注：适用于 CH32V30x_D8C、CH32V30x_D8、CH32F20x_D8C、CH32F20x_D8。* 00：CODE-128KB + RAM-64KB 01：CODE-144KB + RAM-48KB 1x：CODE-160KB + RAM-32KB *注：适用于 CH32V20x_D8W、CH32V20x_D8、CH32F20x_D8W。*	xxb
	[5:3]	Reserved	保留。	1
	2	STANDY_RST	待机模式下系统复位控制： 1：不启用，进入待机模式系统不复位； 0：启用，进入待机模式产生系统复位。	1
	1	STOP_RST	停止模式下系统复位控制： 1：不启用，进入停止模式不复位系统； 0：启用，进入停止模式产生系统复位。	1
	0	IWDG_SW	独立看门狗（IWDG）硬件使能位： 1：IWDG 功能由软件开启，禁止硬件开启； 0：IWDG 功能由硬件开启（随 LSI 时钟决定）。	1
Data0 - Data1			存储用户数据 2 字节。	FFFFh
WRPR0 - WRPR3			写保护控制位。每个比特位用于控制主存储器中 1 个扇区（4K 字节/扇区）的写保护状态： 1：关闭写保护； 0：启用写保护。 4 个字节用于保护总共 512K 字节的主存储器。 WRP0：第 0-7 扇区存储写保护控制； WRP1：第 8-15 扇区存储写保护控制； WRP2：第 16-23 扇区存储写保护控制； WRP3：位 0-6 提供第 24-30 扇区的写保护；位 7 提供第 31-127 扇区的写保护。	FFFFFFFFh

32.6.1 用户选择字解锁

通过写入序列到 FLASH_OBKEYR 寄存器可解锁用户选择字操作。解锁后，FLASH_CTLR 寄存器的 OBWRE 位将置 1，表示可以进行用户选择字的擦除和编程。通过将 FLASH_CTLR 寄存器的“OBWRE”位，软件清 0 来再次锁定。

解锁序列：

1）向 FLASH_OBKEYR 寄存器写入 KEY1 = 0x45670123；

2）向 FLASH_OBKEYR 寄存器写入 KEY2 = 0xCDEF89AB。

注：用户选择字操作需要解除“LOCK”和“OBWRE”两层锁定。

32.6.2 用户选择字编程

只支持标准编程方式，一次写入半字（2 字节）。实际过程中，对用户选择字进行编程时，FPEC 只使用半字中的低字节，并自动计算出高字节（高字节为低字节的反码），然后开始编程操作，这将保证用户选择字中的字节和它的反码始终是正确的。

1）检查 FLASH_CTLR 寄存器 LOCK 位，如果为 1，需要执行“解除闪存锁”操作。
2）检查 FLASH_STATR 寄存器的 BSY 位，以确认没有其他正在进行的编程操作。
3）检查 FLASH_CTLR 寄存器 OBWRE 位，如果为 0，需要执行“用户选择字解锁”操作。
4）设置 FLASH_CTLR 寄存器的 OBPG 位为‘1’，之后设置 FLASH_CTLR 寄存器的 STAT 位为‘1’，开启用户选择字编程。
5）写入要编程的半字（2 字节）到指定地址。
6）等待 BYS 位变为‘0’或 FLASH_STATR 寄存器的 EOP 位为‘1’表示编程结束，将 EOP 位清 0。
7）读编程地址数据校验。
8）继续编程可以重复 5-7 步骤，结束编程将 OBPG 位清 0。

注：当修改选择字中的“读保护”变成“非保护”状态时，会自动执行一次整片擦除主存储区操作。如果修改“读保护”之外的选型，则不会出现整片擦除的操作。

32.6.3 用户选择字擦除

直接擦除整个 128 字节用户选择字区域。

1）检查 FLASH_CTLR 寄存器 LOCK 位，如果为 1，需要执行“解除闪存锁”操作。
2）检查 FLASH_STATR 寄存器的 BSY 位，以确认没有正在进行的编程操作。
3）检查 FLASH_CTLR 寄存器 OBWRE 位，如果为 0，需要执行“用户选择字解锁”操作。
4）设置 FLASH_CTLR 寄存器的 OBER 位为‘1’，之后设置 FLASH_CTLR 寄存器的 STAT 位为‘1’，开启用户选择字擦除。
5）等待 BYS 位变为‘0’或 FLASH_STATR 寄存器的 EOP 位为‘1’表示擦除结束，将 EOP 位清 0
6）读擦除地址数据校验。
7）结束将 OBER 位清 0。

注：擦除成功后，字读-0xe339e339，半字读-0xe339，字节读-0x39。

32.6.4 解除读保护

闪存是否读保护，由用户选择字决定。读取 FLASH_OBR 寄存器，当 RDPRT 位为‘1’表示当前闪存处于读保护状态，闪存操作上受到读保护状态的一系列安全防护。解除读保护过程如下：

1）擦除整个用户选择字区域，此时读保护字段 RDPR，此时读保护仍然有效。
2）用户选择字编程，写入正确的 RDPR 代码 0xA5 以解除闪存的读保护。（此步骤首先将导致系统自动对闪存执行整片擦除操作）
3）进行上电复位以重新加载选择字节（包括新的 RDPR 码），此时读保护被解除。

32.6.5 解除写保护

闪存是否写保护，由用户选择字决定。读取 FLASH_WPR 寄存器，每个比特位代表 4K 字节闪存空间，当比特位为‘1’表示非写保护状态，为‘0’表示写保护。解除写保护过程如下：

1）擦除整个用户选择字区域。
2）写入正确的 RDPR 码 0xA5，允许读访问；
3）进行系统复位，重新加载选择字节（包括新的 WRPR[3:0]字节），写保护被解除。