

说明

CH32H417 是基于青稞 RISC-V5F 和 RISC-V3F 双内核设计的互联型通用微控制器。CH32H417 集成了 USB 3.2 Gen1 控制器和收发器、百兆以太网 MAC 及 PHY、SerDes 高速隔离收发器、Type-C/PD 控制器及 PHY，提供 SD/EMMC 控制器、500MBytes 通用高速接口 UHSIF、DVP 数字图像接口、单线协议主接口 SWPMI、可编程协议 I/O 控制器 PI0C、灵活存储控制器 FMC、DFSMD、LTDC、GPHA、DMA 控制器、多组定时器、8 组串口、I3C、4 组 I2C、2 组 QSPI、4 组 SPI，2 组 I2S、3 组 CAN 等外设资源，内置了 5M 采样率双 12 位 ADC 单元、20M 采样率 10 位高速 HSADC 单元、16 路 Touchkey、双 DAC 单元、3 组运放 OPA、电压比较器 CMP 等模拟资源，支持 10M/100M 以太网通讯，支持 USB 2.0 和 USB 3.0，支持 USB Host 主机和 USB Device 设备功能、Type-C 和 PDUSB 快充功能，支持 SerDes 高速隔离及远距离传输，支持双内核分工提升网络协议处理效率和通讯响应速度。

本手册针对用户的应用开发，提供了 CH32H417、H416、H415 芯片的详细使用信息。

有关 CH32H417 系列芯片的器件特性请参考 CH32H417DS0 手册。

RISC-V 内核版本概览

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	流水线	向量表模式	扩展指令	指令缓存	内存保护	接口调试
青稞 V3F	IMABCF-X	2	2	4	3	地址或指令	支持	-	标准 ^[1]	单/双线
青稞 V5F	IMABCF-X	8	8	4	7-9	地址或指令	支持	支持	标准	单/双线

注[1]：仅批号第五位不为 0 的芯片支持内存保护。

有关 RISC-V 内核的相关信息，可参考第 4 章中央处理器（CPU）。

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。
RW1Z	可读，写 1 清除此位，写 0 无效。
SC	自动清除。

第 1 章 存储器和总线架构

1.1 总线架构

CH32H417系列芯片是内部集成双内核结构，采用青稞V3F和V5F设计，其架构中将内核、仲裁单元、DMA模块、SRAM存储等部分通过多组总线实现交互。设计中集成通用DMA控制器以减轻CPU负担、提高访问效率，同时兼有数据保护机制，时钟自动切换保护等措施增加了系统稳定性。其系统框图见下图。

图 1-1-1 CH32H417 系统框图

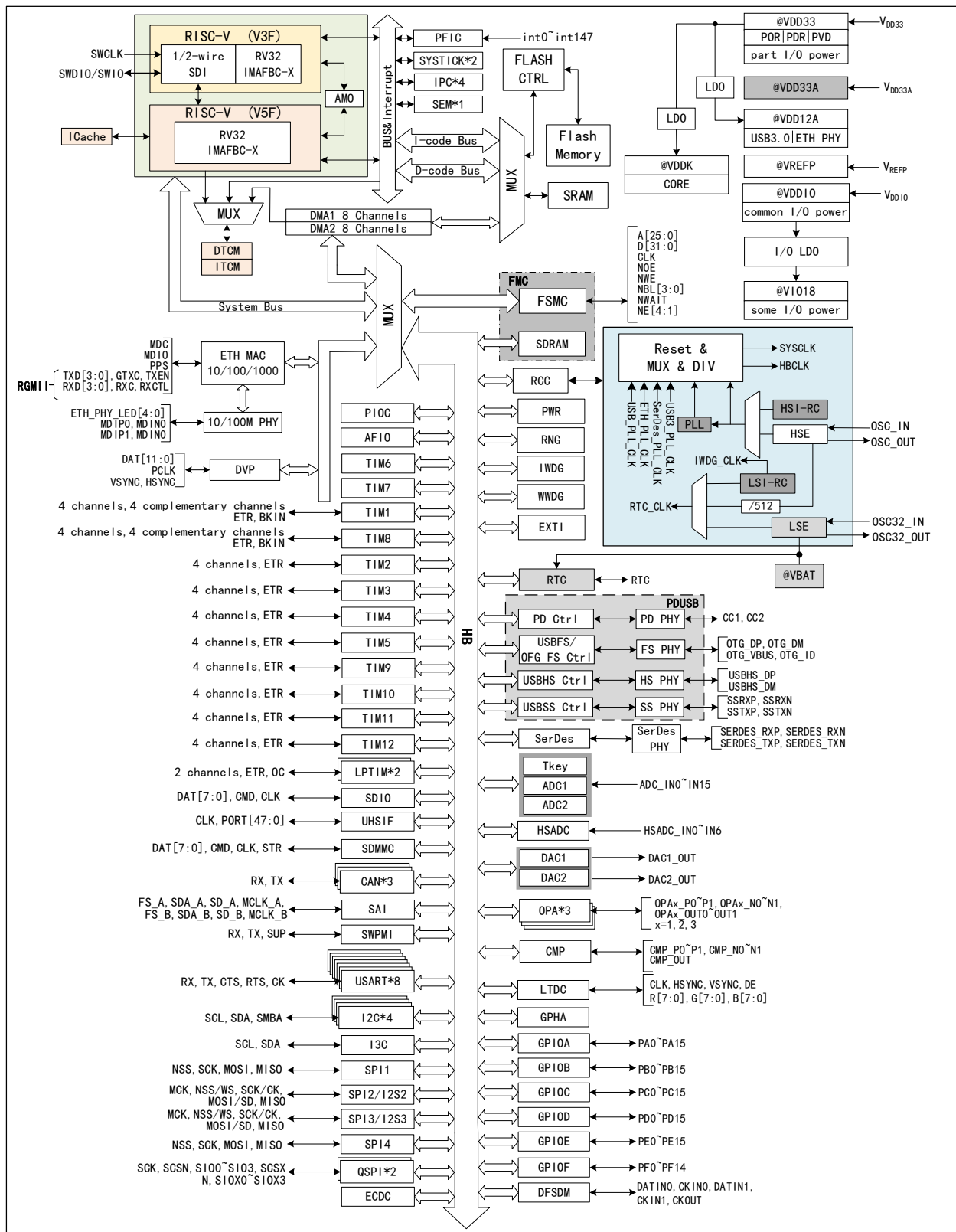


图 1-1-2 CH32H416 系统框图

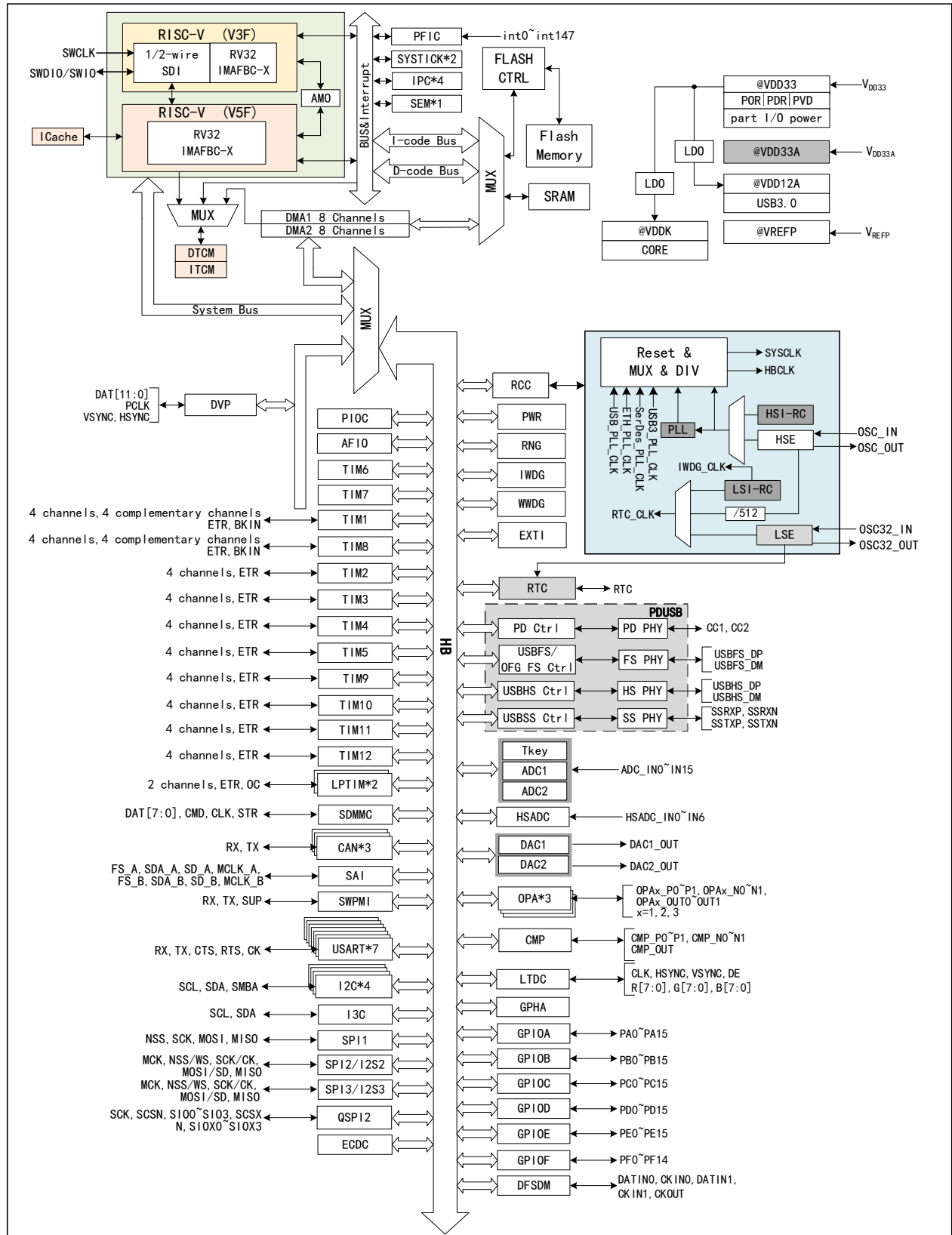
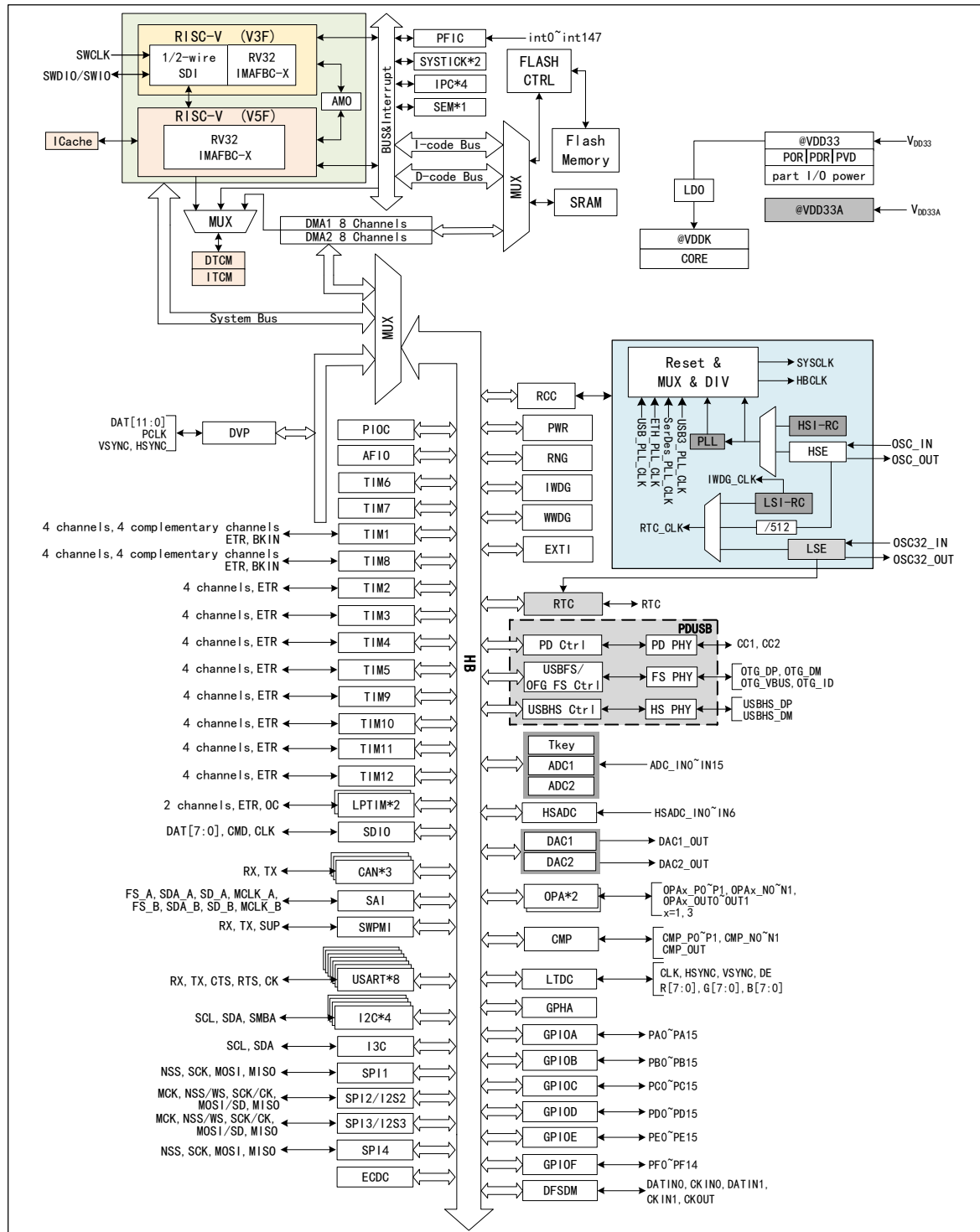


图 1-1-3 CH32H415 系统框图



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。

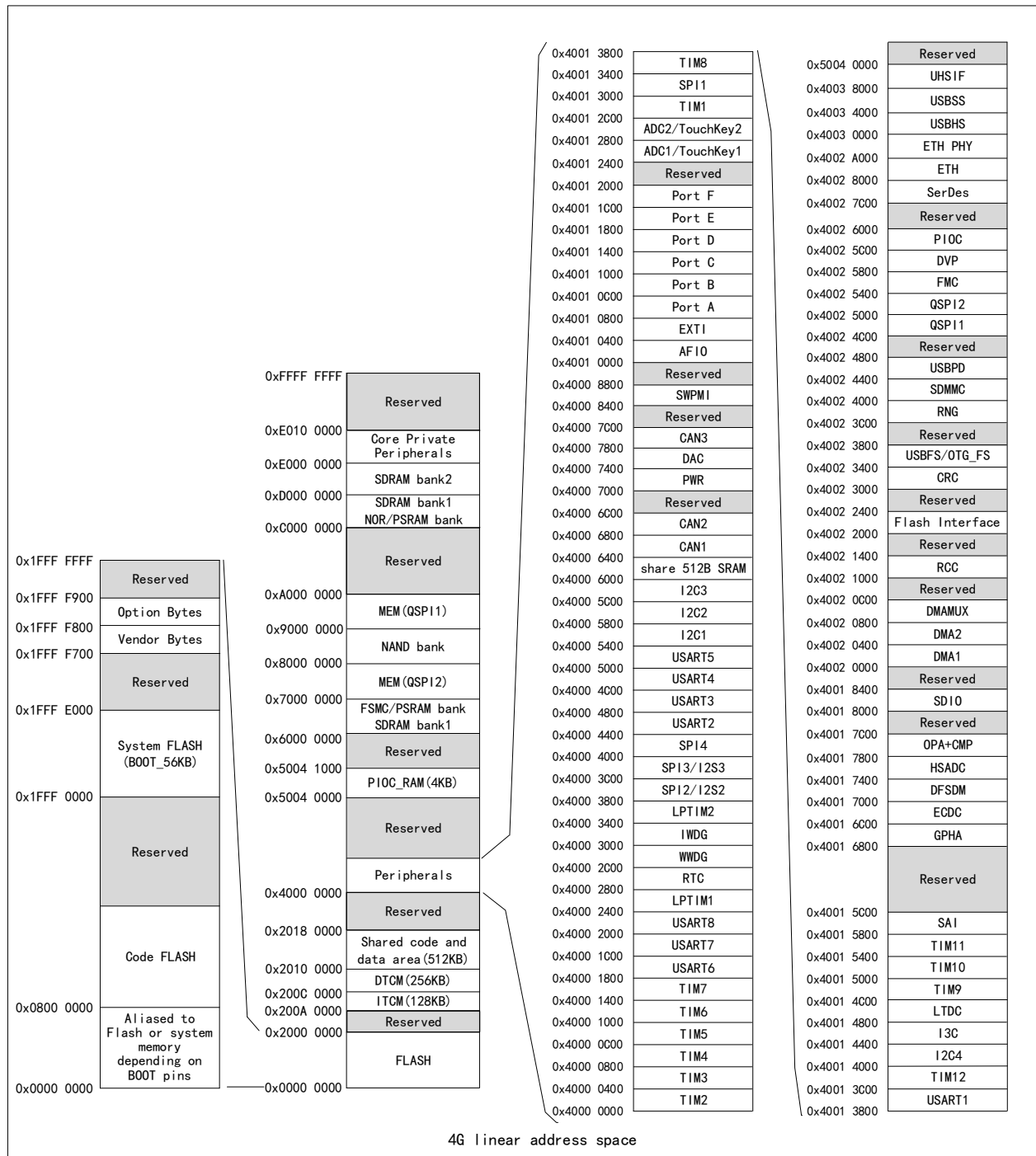
- 指令总线（I-Code）将内核和 FLASH 指令接口相连，预取指在此总线上完成
- 数据总线（D-Code）将内核和 FLASH 数据接口相连，用于常量加载和调试
- 系统总线将内核和总线矩阵相连，用于协调内核、DMA、SRAM 和外设的访问
- DMA 总线负责 DMA 的 HB 主控接口与总线矩阵相连，该总线访问对象是 FLASH 数据、SRAM 和外设
- 总线矩阵负责的是系统总线、数据总线、DMA 总线、SRAM 之间的访问协调

1.2 存储器映像

CH32H417 系列芯片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。

图 1-2 存储映像



1.2.1 存储器分配

内置总容量 896K 字节的 SRAM，部分用于数据区，部分用于零等待代码区，掉电后数据丢失。上述 SRAM 分为 3 块：128KB 的 ITCM 内核 1 紧耦合零等待代码区、256KB 的 DTCM 内核 1 紧耦合零等待数据区、剩余的 512KB 共享代码和数据区。另外，内核 1 还有 32KB 的高速指令缓存区。

512KB 共享区可配置为 RISC-V3F 的零等待代码区和数据区，建议以 128KB 为单位根据需要灵活

分配。

作为选项，128KB 的 ITCM 和 256KB 的 DTCM 合计 384KB 均可以配置为 RISC-V5F 的代码区，其中 DTCM 作为代码区在发生跳转时会增加 1 个时钟等待。

RISC-V3F 可以按 HCLK 时钟 2 个等待访问 ITCM 或 DTCM。

RISC-V5F 和 RISC-V3F 可以按 HCLK 时钟零等待访问 512KB 共享区。

内置 256 字节系统非易失配置信息存储区，用于厂商配置字存储，出厂前固化，用户不可修改。

内置 256 字节用户自定义信息存储区，用于用户选择字存储。

用户区和 BOOT 区大小通过 R32_FLASH_CFGRO 寄存器 DBMODE 位读取：

当 DBMODE=1 时，芯片内置 960K 字节非零等待的程序闪存存储区（Code FLASH），即用户区，用于用户的应用程序和常量数据存储，等效频率约 25MHz。内置最大 56K 字节系统存储区（System FLASH），即 BOOT 区，用于系统引导程序存储，内置自举加载程序。

当 DBMODE=0 时，芯片内置 480K 字节非零等待的程序闪存存储区（Code FLASH），即用户区，用于用户的应用程序和常量数据存储，等效频率约 12.5MHz。内置最大 28K 字节系统存储区（System FLASH），即 BOOT 区，用于系统引导程序存储，内置自举加载程序。

芯片的 BOOT 功能使用和注意事项请参考《CH32H417 评估板说明书》。

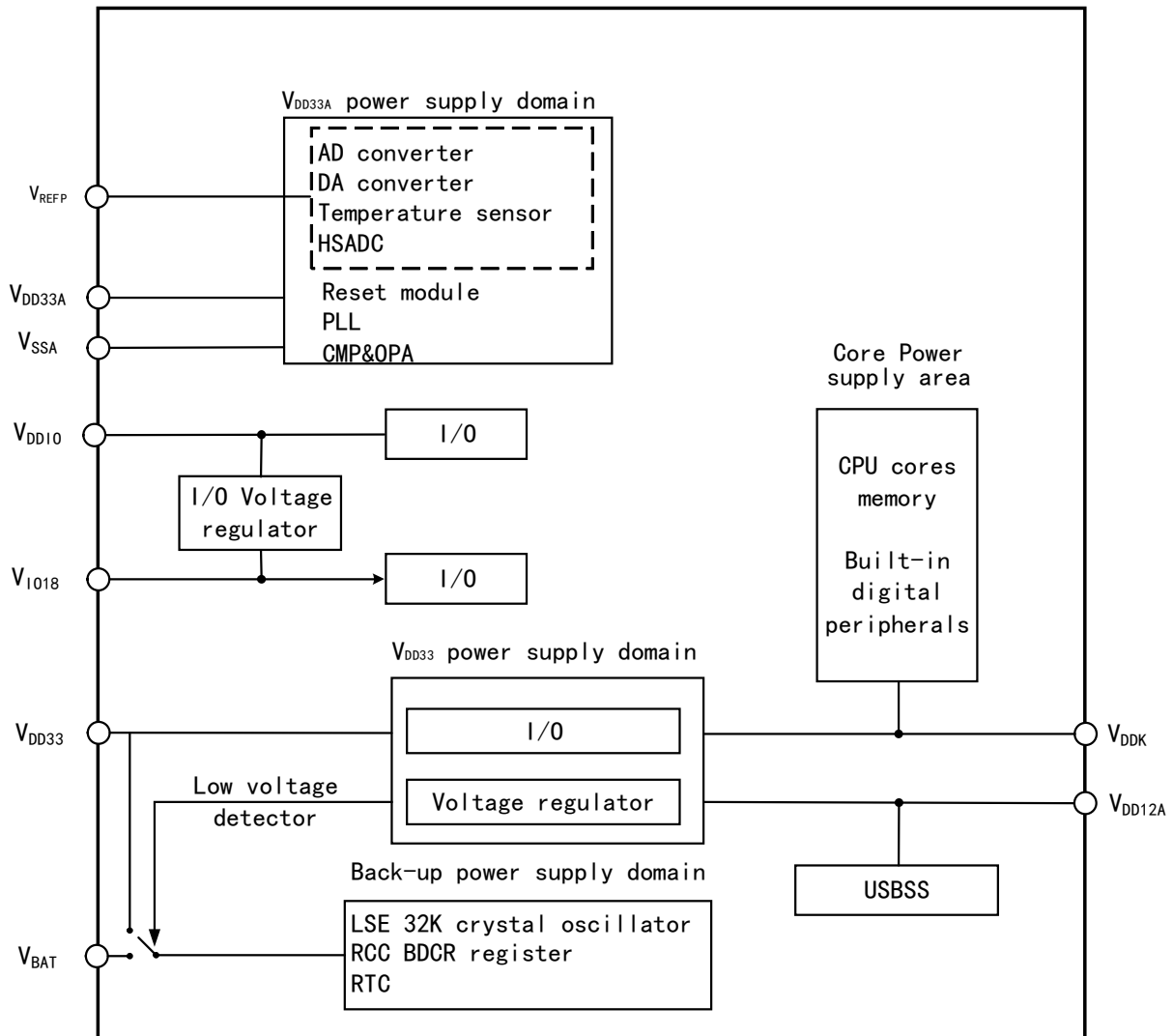
第 2 章 电源控制（PWR）

2.1 概述

CH32H417 芯片系统工作电压 V_{DD33} 范围为 2.4~3.6V，内置电压调节器提供内核所需的工作电源。当主电源 V_{DD33} 掉电后，电池等后备电源可通过 V_{BAT} 引脚为实时时钟（RTC）提供电源，如果无需后备电源，建议将 V_{DD33} 直接连接到 V_{BAT} 引脚上。

V_{DD33A} 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_{REFP} 作为一些模拟电路例如 ADC 电压上限的参考点，部分封装为独立引脚，未封装出来则在芯片内部等于 V_{DD33A} 。

图 2-1 电源结构框图

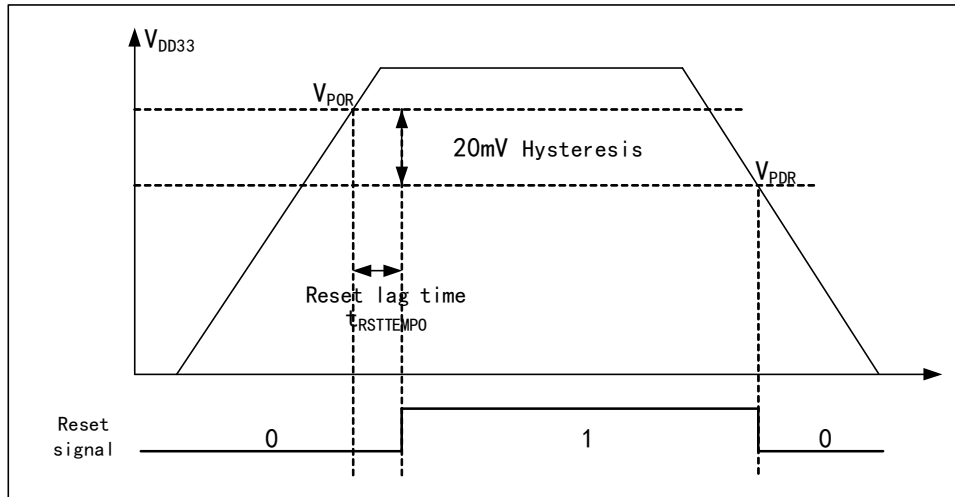


2.2 电源管理

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_{DD33} 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



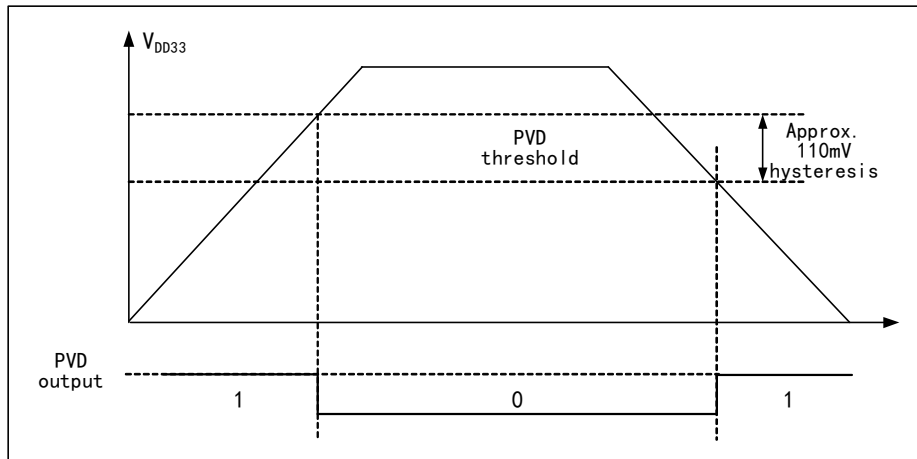
2.2.2 可编程电压监测器

可编程电压监测器 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，可产生状态标志，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) 设置 PWR_CTLR 寄存器的 PVDE 位来开启 PVD 功能。
- 3) 读取 PWR_CSR 状态寄存器的 PVD0 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。当 V_{DD33} 电压高于 PLS[2:0] 设置阈值，PVD0 位置 0；当 V_{DD33} 电压低于 PLS[2:0] 设置阈值，PVD0 位置 1。

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 2 种低功耗模式，从处理器、外设、电压调节器等工作差异上分为：

- 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：V3F 和 V5F 均进入停止模式时，停止模式才生效。停止所有时钟，唤醒后系统继续运行。

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 WFI 或 WFE	任意外部中断/事件、RST 上的外部复位信号、IWDG 复位。	V3F 和 V5F 均进入停止模式：关闭 HSE、HSI、PLL 和外设时钟。	正常：LPDS=0 或低功耗：LPDS=1

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) 中断响应和中断退出标记作为事件，保持在中断控制器中；
- 3) 或者配置 SEVONPEND 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。
- 4) 调试器的调试请求。

● 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（SLEEP）

此模式下，所有的 I/O 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，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（STOP）

停止模式是在 V3F 和 V5F 内核均进入深睡眠模式（SLEEPDEEP）的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。V3F 和 V5F 均进入此模式：高频时钟（HSE/HSI/PLL）域被关闭，SRAM 和寄存器内容保持，I/O 引脚状态保持。该模式唤醒后系统可以继续运行，HSI 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 HB 的访问，直到对 HB 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，可选 LPDS 位，执行 WFI 或 WFE，可选 SEVONPEND 和 SLEEPONEXIT。

退出：任意外部中断/事件、RST 上的外部复位信号、IWDG 复位。

在停止模式下，可选 LPDS 位，LPDS=0，电压调节器工作在正常模式；LPDS=1，电压调节器工作在低功耗模式。

2.3.4 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 从停止模式下唤醒，为了实现此功能，需要配置外部中断线 17，并且把 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	0x00000X00
R32_SYS_CFGR0	0x5003C000	系统配置寄存器 0	0xFFFFFFFF

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			VSEL_VI018[2:0]			VIO_SW_CR	DBP	PLS[2:0]			PVDE	Reserved			LPDS

位	名称	访问	描述	复位值
[31:13]	Reserved	R0	保留。	0
[12:10]	VSEL_VI018[2:0]	RW	V _{IO18} 电源调节位： 000: 1.2V; 001: 1.8V; 010: 2.5V; 011: 3.3V; 100: V _{IO18} 断电，且有 10mA 放电; 101: V _{IO18} 断电，输出浮空; 110: V _{IO18} 断电，输出浮空; 111: V _{IO18} 断电，输出浮空。	0
9	VIO_SW_CR	RW	V _{IO18} 电源调节方式选择位： 1: 通过 VSEL_VI018 软件配置; 0: X0 外部下拉电阻硬件配置。 注：V _{IO18} 上电初始值是由硬件配置的，即通过 X0 外部下拉电阻决定： Float: 1.8V; 下拉 330K: 2.5V; 下拉 100K: 1.2V。 注：1. 对于外封 V _{IO18} 的芯片，用户使用 3.3V 时片外需	0

			短接 V_{DD10} 和 V_{I018} ，而且上电完成后需要在软件中通过 $VSEL\_VI018=111b$ 关闭 $LD0$ ，减小功耗。 2. 对于不外封 V_{I018} 的芯片，片内短接了 V_{DD10} 和 V_{I018} ，而且上电完成后需要在软件中通过 $VSEL\_VI018=111b$ 关闭 $LD0$ ，减小功耗。	
8	DBP	RW	后备区域的写使能： 在复位状态下，后备区域的寄存器均受到写访问保护。 必须将此位置 1 才能对这些寄存器进行写访问。 1：使能对后备区域的寄存器的访问； 0：禁止对后备区域的寄存器的访问。	0
[7:5]	PLS[2:0]	RW	PVD 电压监测阈值设置。详细说明见数据手册中电气特性部分。 000：上升沿 2.54V/下降沿 2.44V； 001：上升沿 2.6V/下降沿 2.49V； 010：上升沿 2.7V/下降沿 2.59V； 011：上升沿 2.8V/下降沿 2.69V； 100：上升沿 2.9V/下降沿 2.79V； 101：上升沿 3.0V/下降沿 2.89V； 110：上升沿 3.1V/下降沿 2.99V； 111：上升沿 3.2V/下降沿 3.09V。	0
4	PVDE	RW	电源电压监测功能使能位： 1：开启电源电压监测功能； 0：禁止电源电压监测功能。	0
[3:1]	Reserved	RO	保留。	0
0	LPDS	RW	停止模式下，电压调节器工作模式选择位。 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

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						VI018_SR [1:0]		Reserved						PVD0	

位	名称	访问	描述	复位值
[31:10]	Reserved	RO	保留。	0
[9:8]	VI018_SR[1:0]	RO	V_{I018} 电源初始状态指示位： 00：1.2V； 01：2.5V； 1x：1.8V。 注： V_{I018} 上电初始值由 VIO_SW_CR 和 $VSEL\_VI018[2:0]$ 决定。	x

[7:1]	Reserved	R0	保留。	0
0	PVDO	R0	PVD 输出状态标志位。当 PWR_CTLR 寄存器的 PVDE=1 时，该位有效。 1: V_{DD33} 低于 PLS[2:0] 设定的 PVD 下降沿阈值； 0: V_{DD33} 高于 PLS[2:0] 设定的 PVD 上升沿阈值。	0

2.4.3 系统配置寄存器 0 (SYS_CFGR0)

访问地址：0x5003C000

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_VDDK[2:0]			

位	名称	访问	描述	复位值
[31:20]	Reserved	RW	保留，写入时必须保持原值。	x
[6:4]	LDO_VDDK[2:0]	RW	调整数字内核电压值，LDO_VDDK 电压值调节位： 011：额定电压； 101：提升电压优化性能，仅适用于 $T_A \leq 70^\circ\text{C}$ 且散热良好的商业级应用； 其他：保留。 当应用中采用外部供电时，建议设置 001b 或 000b。	011b
[3:0]	Reserved	RW	保留，写入时必须保持原值。	x

第 3 章 复位和时钟控制（RCC）

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）。

其产生条件包括：

- 上电/掉电复位（POR/PDR 复位）

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：

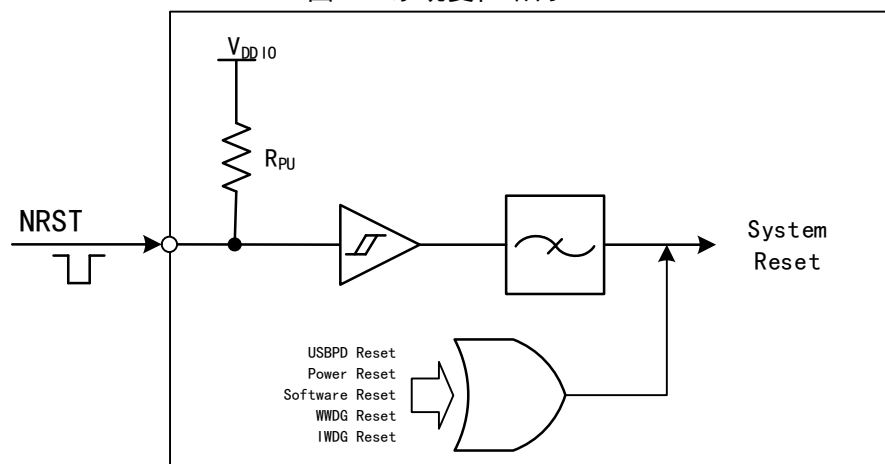
- RST 引脚上的低电平信号（外部复位）
- 窗口看门狗计数终止（WWDG 复位）
- 独立看门狗计数终止（IWDG 复位）
- 软件复位（SW 复位）
- USBPD 复位

窗口/独立看门狗复位：由窗口/独立看门狗外设定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：通过可编程中断控制器 PFIC 中的中断配置寄存器 PFIC_CFGR 的 SYSRST 位置 1 复位系统或配置寄存器 PFIC_SCTLR 的 SYSRST 位置 1 复位系统，具体参考对应章节。

USBPD 复位：当 PD_RST_EN 为 1 时，CH32H417 支持 USB PD 信号帧 Hard Reset 产生的复位；如果 IE_RX_RESET 也为 1，则还支持信号帧 Cable Reset 产生的复位。USB PD 没有复位标志，但产生的复位效果同软件复位。

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 RCC_BDCTLR 寄存器（RTC 使能和 LSE 振荡器）。其产生条件包括：

- 在 V_{DD33} 和 V_{BAT} 都掉电的前提下，由 V_{DD33} 或 V_{BAT} 上电引起
- RCC_BDCTLR 寄存器的 BDRST 位置 1

3.3 时钟

3.3.1 系统时钟结构

图 3-2 时钟树框图 (仅适用于批号第五位为 0 的芯片)

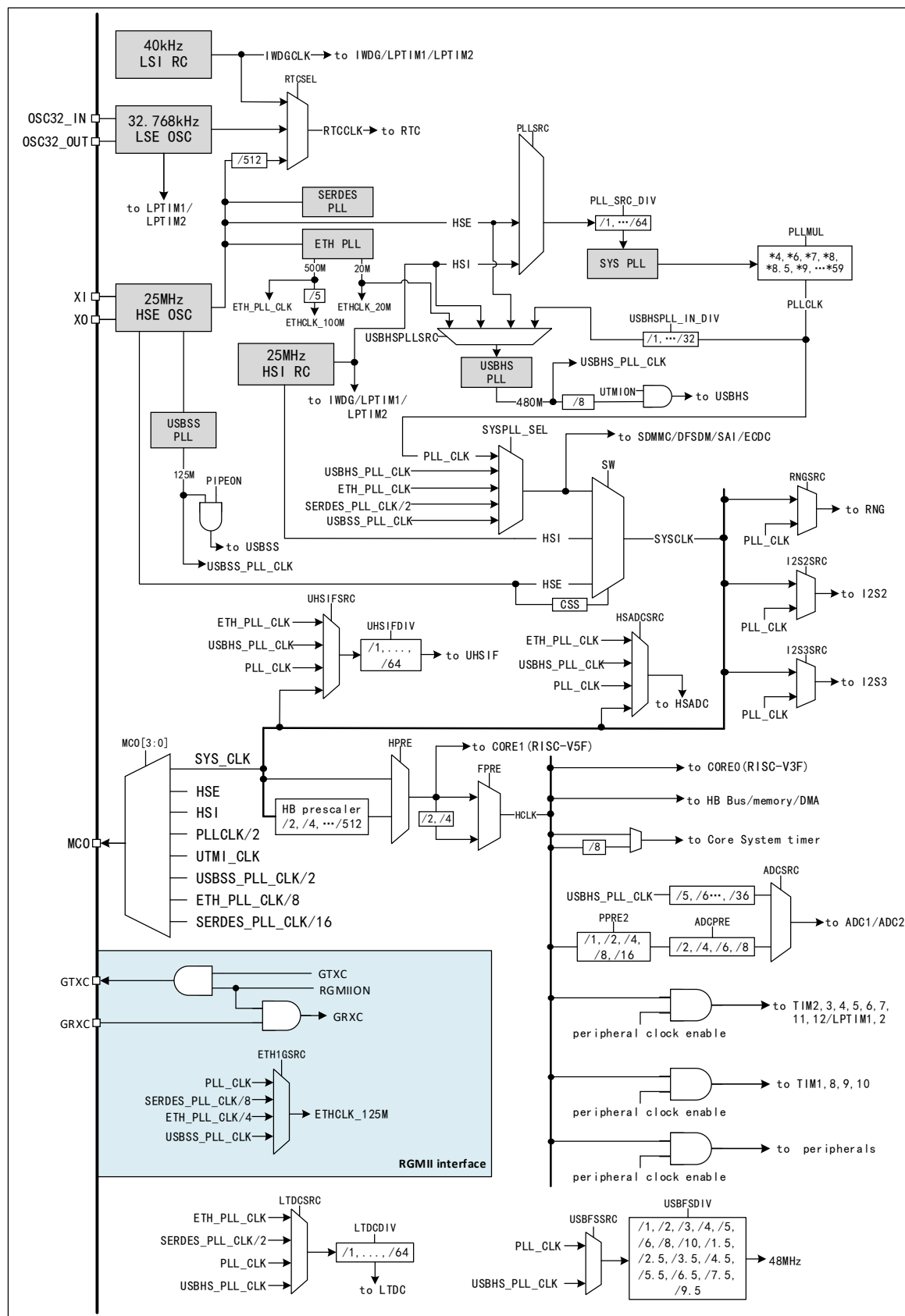
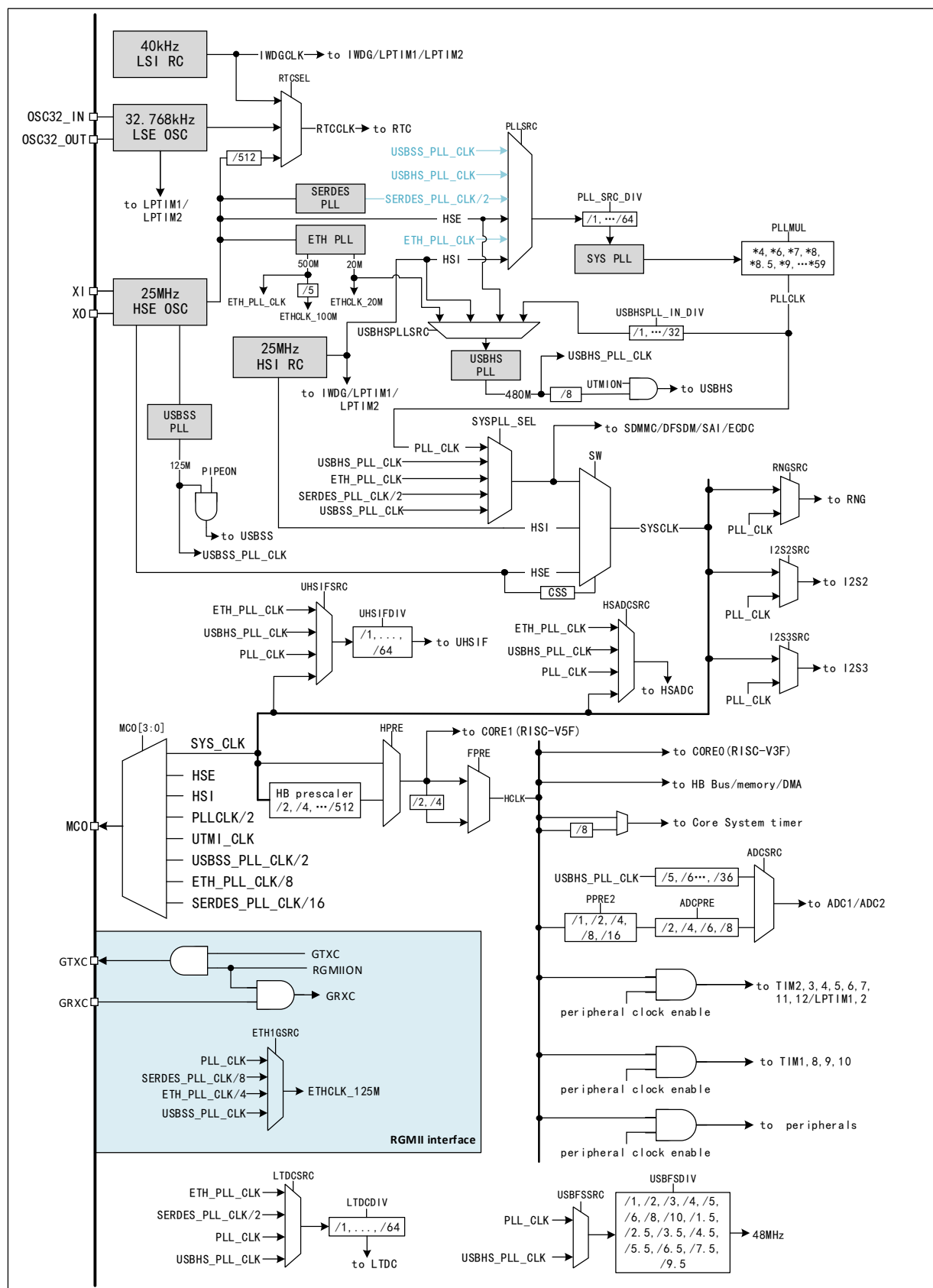


图 3-3 时钟树框图 (适用于批号第五位大于 0 的芯片)



3.3.2 高速时钟 (HSI/HSE)

HSI 是系统内部 25MHz 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。

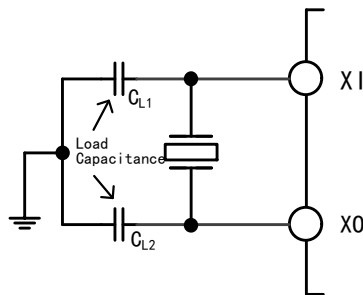
- 出厂校准: 制造工艺的差异会导致每个芯片的 RC 振荡频率不同, 所以在芯片出厂前, 会为每颗芯片进行 HSI 校准。系统复位后, 工厂校准值被装载到 RCC_CTLR 寄存器的 HSICAL[7:0] 中。
- 用户调整: 基于不同的电压或环境温度, 应用程序可以通过 RCC_CTLR 寄存器里的 HSITRIM[4:0] 位来调整 HSI 频率。

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。

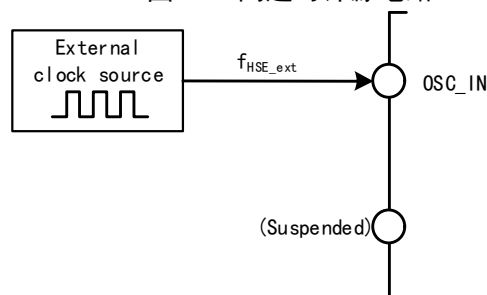
图 3-4 高速外部晶体电路



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 X1 引脚, X0 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 3-5 高速时钟源电路



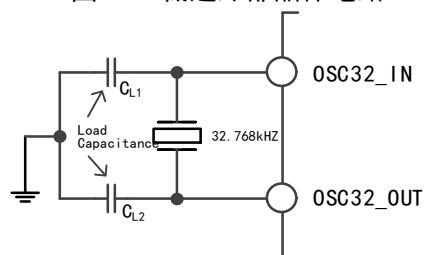
3.3.3 低速时钟 (LSI/LSE)

LSI 是系统内部的 RC 振荡器产生的低速时钟信号。它可以在停止模式下保持运行, 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_RSTSKR 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。

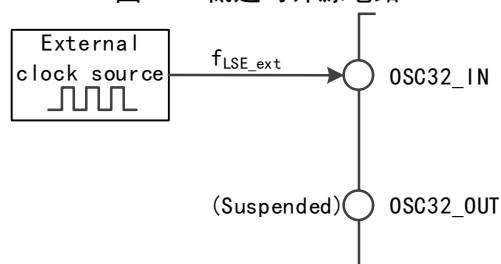
- 外部晶体/陶瓷谐振器 (LSE 晶体): 外接 32.768kHz 的外部低速振荡器。LSE 通过设置 RCC_BDCTL 寄存器中的 LSEON 位被启动和关闭, LSERDY 位指示 LSE 晶体振荡是否稳定, 硬件在 LSERDY 位置 1 后才将时钟送入系统。如果设置了 RCC_INTR 寄存器的 LSERDYIE 位, 将产生相应中断。

图 3-6 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 3-7 低速时钟源电路



3.3.4 PLL 时钟

通过配置 RCC_CFGFR0 寄存器和 RCC_PLLCFGR 寄存器，内部 PLL 时钟可以选择 6 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 USBHS_PLLON 位被启动和关闭，USBHS_PLLRDY 位指示 USBHS_PLL 时钟是否稳定，硬件在 USBHS_PLLRDY 位置 1 后才将时钟送入系统。

设置 RCC_CTLR 寄存器中的 USBSS_PLLON 位被启动和关闭，USBSS_PLLRDY 位指示 USBSS_PLL 时钟是否稳定，硬件在 USBSS_PLLRDY 位置 1 后才将时钟送入系统。

设置 RCC_CTLR 寄存器中的 ETH_PLLON 位被启动和关闭，ETH_PLLRDY 位指示 ETH_PLL 时钟是否稳定，硬件在 ETH_PLLRDY 位置 1 后才将时钟送入系统。

设置 RCC_CTLR 寄存器中的 SERDES_PLLON 位被启动和关闭，SERDES_PLLRDY 位指示 SERDES_PLL 时钟是否稳定，硬件在 SERDES_PLLRDY 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、ETH_PLLRDYIE 位或 SERDES_PLLRDYIE 位，将产生相应中断。

PLL 时钟来源：

- HSI 时钟送入
- HSE 时钟送入
- USBHS_PLL (480MHz) 时钟送入
- ETH_PLL (500MHz) 时钟送入
- USBSS_PLL (125MHz) 时钟送入
- SERDES_PLL 时钟 2 分频送入

3.3.5 总线/外设时钟

3.3.5.1 系统时钟 (SYSCLK)

通过配置 RCC_CFGFR0 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 HB 总线外设时钟 (HCLK)

通过配置 RCC_CFGR0 寄存器的 HPRE[3:0]、PPRE2[2:0] 位，可以配置 HB 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_HBRSTR、RCC_HB1PRSTR、RCC_HB2PRSTR 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_HBPCENR、RCC_HB1PCENR、RCC_HB2PCENR 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 分频、LSE 或 LSI 时钟提供。修改此位前要保证电源控制寄存器 (PWR_CTLR) 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_{DD33} 供电被切断，RTC 仍继续工作。
- LSI 作为 RTC 时钟：如果 V_{DD33} 供电被切断，RTC 自动唤醒不能保证。
- HSE 经 512 分频后作为 RTC 时钟：如果 V_{DD33} 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 2 分频后输出
- UTMI 时钟输出
- USBSS_PLL 时钟 2 分频后输出
- ETH_PLL 时钟 8 分频后输出
- SERDES_PLL 时钟 16 分频后输出

3.3.5.6 USB 时钟

USBHS 时钟来自 USBHSPLL 时钟，通过配置 RCC_PLLCFGR2 寄存器的 USBHSPLLSRC 位，选择 USBHS_PLL 时钟源；若选择输入时钟为分频后的 SYS_PLL_CLK，分频系数可通过 USBHSPLL_IN_DIV[4:0] 位选择；通过配置 USBHSPLL_REFSEL[1:0] 位对时钟频率进行选择，时钟频率必须和 USBHS_PLL 输入时钟保持一致；通过配置 RCC_CTLR 寄存器的 USBHS_PLLON 位，使能 USBHS_PLL 时钟。

USBFS/OTG_FS 48MHz 时钟源来自通过一个可配置的分频器的 PLL 时钟或者 USBFS_PLL 时钟，可

通过配置 RCC_CFGR2 寄存器的 USBFSSRC 位来选择。通过配置 RCC_HBPCENR 寄存器的 OTGFSEN 位，使能 USBOTG_FS 时钟。

USBSS 时钟来自一个 USBSS_PLL 时钟，其输出 125MHz 时钟到 USBSS。通过配置 RCC_CTLR 寄存器的 USBSS_PLLON 位，使能 USBSS_PLL 时钟。

3.3.5.7 ETH 时钟

ETH 时钟配置参考 31.4.3 章节。

3.3.5.8 I2S 和 RNG 时钟

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

CSSON 在开启的情况下发生 HSE 失效事件，是不影响 HSE 和 PLL 使能的，软件亦可在 NMI 中断里进行处理。若是在激活时钟安全系统 (CSSON=1) 前，通过设置 RCC_CTLR 寄存器 CSS_HSE_DIS 位置 1，在 CSSON 开启的情况下发生 HSE 失效事件，硬件将关闭 HSE 及所有使用 HSE 的 PLL。

3.3.7 RTC 校准

此功能必须配置 PC13 引脚作为普通 IO 口使用。

- 脉冲输出

配置 RCC_BDCTL 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 RCC_BDCTL 寄存器的 CC0 位后，内部的 RTC 时钟将经过 64 分频后输出到 PC13 引脚上。通过实际测试，软件配合修改 RTCCAL[6:0] 位来调整时钟对 RTC 进行校准。

3.3.8 BKP 接口复位

BKP 区域可以在 V_{DD33} 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，ASOS 位、ASOE 位在软件配置 RCC_BDCTL 寄存器的 BDRST 位下复位，不受 RCC 外设接口控制 BKPRST 位影响。

3.4 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_CFGRO	0x40021004	时钟配置寄存器 0	0x00000000

R32_RCC_PLLCFGR	0x40021008	PLL 配置寄存器	0x00000004
R32_RCC_INTR	0x4002100C	时钟中断寄存器	0x00000000
R32_RCC_HB2PRSTR	0x40021010	HB2 外设复位寄存器	0x00000000
R32_RCC_HB1PRSTR	0x40021014	HB1 外设复位寄存器	0x00000000
R32_RCC_HBPCENR	0x40021018	HB 外设时钟使能寄存器	0x00000000
R32_RCC_HB2PCENR	0x4002101C	HB2 外设时钟使能寄存器	0x00000000
R32_RCC_HB1PCENR	0x40021020	HB1 外设时钟使能寄存器	0x00000000
R32_RCC_BDCTL	0x40021024	后备域控制寄存器	0x00000000
R32_RCC_RSTSCKR	0x40021028	控制/状态寄存器	0x08000000
R32_RCC_HBRSTR	0x4002102C	HB 外设复位寄存器	0x00000000
R32_RCC_CFGR2	0x40021030	时钟配置寄存器 2	0x00000000
R32_RCC_PLLCFGR2	0x40021034	PLL 配置寄存器 2	0x00080020

3.4.1 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
CSS_HSE_DIS	Reserved	SERDES_PLLRDY	SERDES_PLLON	ETH_PLLRDY	ETH_PLLON	PLLRDY	PLLON	USBSS_PLLRDY	USBSS_PLLON	USBHS_PLLRDY	USBHS_PLLON	CSSON	HSEBYP	HSERDY	HSEON
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]				Reserved	HSIRDY	HSION	

位	名称	访问	描述	复位值
31	CSS_HSE_DIS	RW	时钟安全系统开启情况下发生 HSE 失效事件的配置： 1: CSSON 开启的情况下发生 HSE 失效事件，硬件关闭 HSE 及所有使用 HSE 的 PLL； 0: CSSON 开启的情况下发生 HSE 失效事件，不影响 HSE 和 PLL 的使能，软件可在 NMI 中断里进行处理。 注：该位仅在 CSSON 为 0 时才可写入。	0
30	Reserved	RO	保留。	0
29	SERDES_PLLRDY	RO	SERDES_PLL 锁定标志位： 1: SERDES_PLL 时钟锁定； 0: SERDES_PLL 时钟未锁定。	0
28	SERDES_PLLON	RW	SERDES_PLL 时钟使能控制位： 1: 使能 SERDES_PLL 时钟； 0: 关闭 SERDES_PLL 时钟。 进入深睡眠后硬件自动清零。	0
27	ETH_PLLRDY	RO	ETH_PLL 锁定标志位： 1: ETH_PLL 时钟锁定； 0: ETH_PLL 时钟未锁定。	0
26	ETH_PLLON	RW	ETH_PLL 使能控制位： 1: 使能 ETH_PLL 时钟； 0: 关闭 ETH_PLL 时钟。	0

			进入深睡眠后硬件自动清零。	
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： 1：使能 PLL 时钟； 0：关闭 PLL 时钟。 注：进入停止模式后，此位由硬件清 0。	0
23	USBSS_PLLRDY	RO	USBSS_PLL 时钟就绪锁定标志位： 1：USBSS_PLL 时钟锁定； 0：USBSS_PLL 时钟未锁定。	0
22	USBSS_PLLON	RW	USBSS_PLL 时钟使能控制位： 1：使能 USBSS_PLL 时钟； 0：关闭 USBSS_PLL 时钟。 进入深睡眠后硬件自动清零；	0
21	USBHS_PLLRDY	RO	USBHS_PLL 时钟就绪锁定标志位： 1：USBHS_PLL 时钟锁定； 0：USBHS_PLL 时钟未锁定。	0
20	USBHS_PLLON	RW	USBHS_PLL 时钟使能控制位： 1：使能 USBHS_PLL 时钟； 0：关闭 USBHS_PLL 时钟。 进入深睡眠后硬件自动清零；	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：进入停止低功耗模式后，此位由硬件清 0。	0
[15:8]	HSICAL[7:0]	RO	内部高速时钟校准值，在系统启动时被自动初始化。	x
[7:3]	HSITRIM[4:0]	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 25MHz±0.25%；每步	10000b

			HSICAL 的变化调整约 50kHz。	
2	Reserved	RO	保留。	0
1	HSIRDY	RO	内部高速时钟(25MHz)稳定就绪标志位(由硬件置位): 0: 内部高速时钟(25MHz)没有稳定; 1: 内部高速时钟(25MHz)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(25MHz)使能控制位: 1: 使能 HSI 振荡器; 0: 关闭 HSI 振荡器。 注: 当从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 2MHz 的 RC 振荡器。	1

3.4.2 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
ADCSRC	ADC_DUTY_SEL	Reserved	MCO[3:0]				UTMION	PIPEON	RGMION	Reserved				FPRE[1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPRE[1:0]		PPRE2[2:0]		Reserved				HPRE[3:0]				SWS[1:0]		SW[1:0]	

位	名称	访问	描述	复位值
31	ADCSRC	RW	ADC 输入时钟源选择: 1: USBHS_PLL (480MHZ); 0: HCLK。	0
30	ADC_DUTY_SEL	RW	ADC 时钟占空比选择: 1: ADC 时钟占空比为 75%; 0: ADC 时钟占空比为 50%。 注: 该位仅在 ADC 输入时钟源选择 HCLK 时有效。	0
[29:28]	Reserved	RO	保留。	0
[27:24]	MCO[3:0]	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 25MHz 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: UTMI 时钟输出; 1001: USBSS_PLL 时钟 2 分频后输出; 1010: ETH_PLL 时钟 8 分频后输出; 1011: SERDES_PLL 时钟 16 分频后输出; 其他: 保留。	0
23	UTMION	RW	UTMI 时钟门控使能: 1: 使能;	0

			0: 关闭。	
22	PIPEON	RW	PIPE 时钟门控使能: 1: 使能; 0: 关闭。	0
21	RGMIIION	RW	1000M 以太网 RGMII 接口是否启用和时钟使能: 1: 启用 1000M 以太网 RGMII 接口并使能时钟; 0: 不启用并关闭时钟。	0
[20:18]	Reserved	RO	保留。	0
[17:16]	FPRE[1:0]	RW	HCLK 输入时钟源预分频控制: 00: 不分频; 01: 2 分频; 1x: 4 分频。	0
[15:14]	ADCPRE[1:0]	RW	ADC 时钟来源选择 USBHS_PLL 预分频控制高 2 位(配合 RCC_CFGR0 寄存器 bit[13:11] PPRE2[2:0]使用): 00000: 5 分频; 00001: 6 分频; 00010: 7 分频; 00011: 8 分频; 11110: 35 分频; 11111: 36 分频。 注: 当使用 TIM1/8/9/10 时, 禁止 PPRE2[2:0]配置成 101b/110b/111b。 ADC 时钟来源选择 HCLK 预分频控制: 00: 2 分频; 01: 4 分频; 10: 6 分频; 11: 8 分频。 注: ADC 时钟最高不要超过 80MHz。	0
[13:11]	PPRE2[2:0]	RW	作用于 ADC 时, 作为时钟来源选择 USBHS_PLL 预分频控制低 3 位(配合 RCC_CFGR0 寄存器 bit[15:14] ADCPRE[1:0]使用)。 作用于 ADC 时, 时钟来源选择 HCLK 预分频控制: 0xx: 不分频; 100: 2 分频; 101: 4 分频; 110: 8 分频; 111: 16 分频。 注: 当使用 TIM1/8/9/10 时, 禁止 PPRE2[2:0]配置成 101b/110b/111b。	0
[10:8]	Reserved	RO	保留。必须保持为 0。	0

[7:4]	HPRE[3:0]	RW	HB 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。	0
[3:2]	SWS[1:0]	RO	系统时钟（SYSCLK）状态（硬件置位）： 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。	0
[1:0]	SW[1:0]	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

3.4.3 PLL 配置寄存器（RCC_PLLCFGR）

偏移地址：0x08

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

位	名称	访问	描述	复位值
31	SYSPLL_GATE	RW	系统时钟切换至 PLL 前，该位需置位。	0
[30:28]	SYSPLL_SEL[2:0]	RW	系统时钟切换至 PLL 选择，只有 SYSPLL_GATE 为 0 才允许写： 0xx: PLL_CLK； 100: USBHS_PLL（480MHz）； 101: ETH_PLL（500MHz）； 110: SERDES_PLL/2； 111: USBSS_PLL（125MHz）。	0
[27:14]	Reserved	RO	保留。	0
[13:8]	PLL_SRC_DIV[5:0]	RW	PLL 输入时钟源分频，只有在 PLLON 为 0 时才能写：	0

	0]		000000: 1 分频; 000001: 2 分频; 000010: 3 分频; ... 111111: 64 分频。	
[7:5]	PLLSRC[2:0]	RW	PLL 输入时钟来源, 只有在 PLLON 为 0 时才能写: 000: HSI; 001: HSE; 100: USBHS_PLL (480MHz) ; 101: ETH_PLL (500MHz) ; 110: USBSS_PLL (125MHz) ; 111: SERDES_PLL/2。 注: 主 PLL 的输入参考时钟, 由上述时钟源经 PLL_SRC_DIV 分频, PLLMUL 倍频得到。	0
[4:0]	PLLMUL[4:0]	RW	PLL 时钟倍频系数, 只有在 PLLON 为 0 时才能写: 00000: PLL 4 倍频输出; 00001: PLL 6 倍频输出; 00010: PLL 7 倍频输出; 00011: PLL 8 倍频输出; 00100: PLL 8.5 倍频输出; 00101: PLL 9 倍频输出; 00110: PLL 9.5 倍频输出; 00111: PLL 10 倍频输出; 01000: PLL 10.5 倍频输出; 01001: PLL 11 倍频输出; 01010: PLL 11.5 倍频输出; 01011: PLL 12 倍频输出; 01100: PLL 12.5 倍频输出; 01101: PLL 13 倍频输出; 01110: PLL 14 倍频输出; 01111: PLL 15 倍频输出; 10000: PLL 16 倍频输出; 10001: PLL 17 倍频输出; 10010: PLL 18 倍频输出; 10011: PLL 19 倍频输出; 10100: PLL 20 倍频输出; 10101: PLL 22 倍频输出; 10110: PLL 24 倍频输出; 10111: PLL 26 倍频输出; 11000: PLL 28 倍频输出; 11001: PLL 30 倍频输出; 11010: PLL 32 倍频输出; 11011: PLL 34 倍频输出; 11100: PLL 36 倍频输出; 11101: PLL 38 倍频输出; 11110: PLL 40 倍频输出;	00100b

			11111: PLL 59 倍频输出。	

3.4.4 时钟中断寄存器 (RCC_INTR)

偏移地址: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CSSC	SERDESPLL RDYC	ETHPLL 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	SERDESPLL RDYIE	ETHPLL RDYIE	PLL RDYIE	HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE	CSSF	SERDESPLL RDYF	ETHPLL 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	SERDESPLL RDYC	WO	清除 SERDES_PLL 就绪中断标志位: 1: 清除 SERDES RDYF 中断标志; 0: 无动作。	0
21	ETHPLL RDYC	WO	清除 ETH_PLL 就绪中断标志位: 1: 清除 ETH RDYF 中断标志; 0: 无动作。	0
20	PLL RDYC	WO	清除 PLL 就绪中断标志位: 1: 清除 PLL RDYF 中断标志; 0: 无动作。	0
19	HSE RDYC	WO	清除 HSE 振荡器就绪中断标志位: 1: 清除 HSE RDYF 中断标志; 0: 无动作。	0
18	HSI RDYC	WO	清除 HSI 振荡器就绪中断标志位: 1: 清除 HSI RDYF 中断标志; 0: 无动作。	0
17	LSE RDYC	WO	清除 LSE 振荡器就绪中断标志位: 1: 清除 LSE RDYF 中断标志; 0: 无动作。	0
16	LSI RDYC	WO	清除 LSI 振荡器就绪中断标志位: 1: 清除 LSI RDYF 中断标志; 0: 无动作。	0
15	Reserved	RO	保留。	0
14	SERDESPLL RDYIE	RW	SERDES_PLL 就绪中断使能位: 1: 使能 SERDES_PLL 就绪中断; 0: 关闭 SERDES_PLL 就绪中断。	0
13	ETHPLL RDYIE	RW	ETH_PLL 就绪中断使能位: 1: 使能 ETH_PLL 就绪中断;	0

			0: 关闭 ETH_PLL 就绪中断。	
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	SERDESPLLRDYF	RO	SERDES_PLL 时钟就绪锁定中断标志: 1: SERDES_PLL 时钟锁定产生中断; 0: 无 SERDES_PLL 时钟锁定中断。 硬件置位, 软件写 SERDESRDYC 位 1 清除。	0
5	ETHPLLRDYF	RO	ETH_PLL 时钟就绪锁定中断标志: 1: ETH_PLL 时钟锁定产生中断; 0: 无 ETH_PLL 时钟锁定中断。 硬件置位, 软件写 ETHPLLRDYC 位 1 清除。	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

		0: 无 LSI 时钟就绪中断。 硬件置位, 软件写 LSIRDYC 位 1 清除。	

3.4.5 HB2 外设复位寄存器 (RCC_HB2PRSTR)

偏移地址: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I3C RST	LTDCR ST	Reserved		GPHA RST	ECDCR ST	DFSDM RST	Reser ved	OPCMR ST	TIM12 RST	TIM11 RST	TIM10 RST	TIM9 RST	SDIO RST	Reser ved	SAI RST
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C4 RST	USART1 RST	TIM8 RST	SPI1 RST	TIM1 RST	ADC2 RST	ADC1 RST	Reser ved	IOPF RST	IOPB RST	IOPD RST	IOPC RST	IOPA RST	HSADC RST	AFIO RST	

位	名称	访问	描述	复位值
31	I3CRST	RW	I3C 模块复位控制: 1: 复位模块; 0: 无作用。	0
30	LTDCRST	RW	LTDC 模块复位控制: 1: 复位模块; 0: 无作用。	0
[29:28]	Reserved	RO	保留。	0
27	GPHARST	RW	GPHA 模块复位控制: 1: 复位模块; 0: 无作用。	0
26	ECDCRST	RW	ECDC 模块复位控制: 1: 复位模块; 0: 无作用。	0
25	DFSDMRST	RW	DFSDM 模块复位控制: 1: 复位模块; 0: 无作用。	0
24	Reserved	RO	保留。	0
23	OPCMRST	RW	OPCM 模块复位控制: 1: 复位模块; 0: 无作用。	0
22	TIM12RST	RW	TIM12 模块复位控制: 1: 复位模块; 0: 无作用。	0
21	TIM11RST	RW	TIM11 模块复位控制: 1: 复位模块; 0: 无作用。	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	SDIORST	RW	SDIO 模块复位控制: 1: 复位模块; 0: 无作用。	0
17	Reserved	RO	保留。	0
16	SAIRST	RW	SAI 模块复位控制: 1: 复位模块; 0: 无作用。	0
15	I2C4RST	RW	I2C4 模块复位控制: 1: 复位模块; 0: 无作用。	0
14	USART1RST	RW	USART1 模块复位控制: 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 模块复位控制： 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	Reserved	RO	保留。	0
7	IOPFRST	RW	IO 的 PF 端口模块复位控制： 1：复位模块； 0：无作用。	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	HSADCRST	RW	HSADC 模块复位控制： 1：复位模块； 0：无作用。	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

3.4.6 HB1 外设复位寄存器 (RCC_HB1PRSTR)

偏移地址：0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWPMI RST	I2C3 RST	DAC RST	PWR RST	BKP RST	CAN2 RST	CAN1 RST	CAN3 RST	Reser ved	I2C2 RST	I2C1 RST	USART5 RST	USART4 RST	USART3 RST	USART2 RST	SPI4 RST
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	QSPI 2 RST	QSPI1 RST	WWDG RST	LPTI M2 RST	LPTI M1 RST	USAR T8RS T	USAR T7RS T	USAR T6RS T	TIM7 RST	TIM6 RST	TIM5 RST	TIM4 RST	TIM3 RST	TIM2 RST

位	名称	访问	描述	复位值
31	SWPMIRST	RW	SWPMI 模块复位控制： 1：复位模块； 0：无作用。	0
30	I2C3RST	RW	I2C3 模块复位控制： 1：复位模块； 0：无作用。	0
29	DACRST	RW	DAC 模块复位控制：	0

			1: 复位模块; 0: 无作用。	
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	CAN3RST	RW	CAN3 模块复位控制: 1: 复位模块; 0: 无作用。	0
23	Reserved	RO	保留。	0
22	I2C2RST	RW	I2C2 模块复位控制: 1: 复位模块; 0: 无作用。	0
21	I2C1RST	RW	I2C1 模块复位控制: 1: 复位模块; 0: 无作用。	0
20	USART5RST	RW	USART5 模块复位控制: 1: 复位模块; 0: 无作用。	0
19	USART4RST	RW	USART4 模块复位控制: 1: 复位模块; 0: 无作用。	0
18	USART3RST	RW	USART3 模块复位控制: 1: 复位模块; 0: 无作用。	0
17	USART2RST	RW	USART2 模块复位控制: 1: 复位模块; 0: 无作用。	0
16	SPI4RST	RW	SPI4 模块复位控制: 1: 复位模块; 0: 无作用。	0
15	SPI3RST	RW	SPI3 模块复位控制: 1: 复位模块; 0: 无作用。	0
14	SPI2RST	RW	SPI2 模块复位控制: 1: 复位模块; 0: 无作用。	0
13	QSPI2RST	RW	QSPI2 模块复位控制: 1: 复位模块; 0: 无作用。	0
12	QSPI1RST	RW	QSPI1 模块复位控制: 1: 复位模块; 0: 无作用。	0
11	WWDGRST	RW	窗口看门狗复位控制: 1: 复位模块; 0: 无作用。	0
10	LPTIM2RST	RW	LPTIM2 模块复位控制: 1: 复位模块; 0: 无作用。	0
9	LPTIM1RST	RW	LPTIM1 模块复位控制: 1: 复位模块; 0: 无作用。	0
8	USART8RST	RW	USART8 模块复位控制: 1: 复位模块; 0: 无作用。	0
7	USART7RST	RW	USART7 模块复位控制: 1: 复位模块; 0: 无作用。	0
6	USART6RST	RW	USART6 模块复位控制:	0

			1: 复位模块; 0: 无作用。	
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

3.4.7 HB 外设时钟使能寄存器 (RCC_HBPCENR)

偏移地址: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PIOC EN	Reser ved	SERDE SEN	USBPD EN	UHSIF EN	USBOTG FS	Reser ved
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reser ved	ETHMA CEN	DVPEN	USBSS EN	USBHS EN	SDMMC EN	RNGEN	FMCEN	Reser ved	CRC EN	Reserved				DMA2 EN	DMA1 EN

位	名称	访问	描述	复位值
[31:23]	Reserved	RO	保留。	0
22	PIOCEN	RW	PIOC 模块时钟使能位: 1: 模块时钟开启; 0: 模块时钟关闭。	0
21	Reserved	RO	保留。	0
20	SERDESEN	RW	SERDES 模块时钟使能位: 1: 模块时钟开启; 0: 模块时钟关闭。	0
19	USBPDEN	RW	USBPD 模块时钟使能位: 1: 模块时钟开启; 0: 模块时钟关闭。	0
18	UHSIFEN	RW	HSPP 模块时钟使能位: 1: 模块时钟开启; 0: 模块时钟关闭。	0
17	OTGFSEN	RW	USBOTG_FS 模块时钟使能位: 1: 模块时钟开启; 0: 模块时钟关闭。	0
[16:15]	Reserved	RO	保留。	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	USBSSSEN	RW	USBSS 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	SDMMCEN	RW	SDMMC 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	FMCEN	RW	FMC 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
[5:2]	Reserved	RO	保留。	0
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

3.4.8 HB2 外设时钟使能寄存器 (RCC_HB2PCENR)

偏移地址：0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I3C EN	LTDC EN	Reserved		GPHA EN	ECDC EN	DFSDM EN	Reser ved	OPCM EN	TIM12 EN	TIM11 EN	TIM10 EN	TIM9 EN	SDIO EN	Reser ved	SAI EN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C4 EN	USART1 EN	TIM8 EN	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	Reser ved	IOPF EN	IOPB EN	IOPD EN	IOPC EN	IOPA EN	HSADC EN	AFIO EN	

位	名称	访问	描述	复位值
31	I3CEN	RW	I3C 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
30	LTDCEN	RW	LTDC 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
[29:28]	Reserved	RO	保留。	0
27	GPHAEN	RW	GPHA 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
26	ECDCEN	RW	ECDC 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
25	DFSDMEN	RW	DFSDM 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	OPCMEN	RW	OPCM 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
22	TIM12EN	RW	TIM12 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
21	TIM11EN	RW	TIM11 模块时钟使能位：	0

			1: 模块时钟开启; 0: 模块时钟关闭。	
20	TIM10EN	RW	TIM10 模块时钟使能位: 1: 模块时钟开启; 0: 模块时钟关闭。	0
19	TIM9EN	RW	TIM9 模块时钟使能位: 1: 模块时钟开启; 0: 模块时钟关闭。	0
18	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
17	Reserved	RO	保留。	0
16	SAIEN	RW	SAI 模块时钟使能位: 1: 模块时钟开启; 0: 模块时钟关闭。	0
15	I2C4EN	RW	I2C4 模块时钟使能位: 1: 模块时钟开启; 0: 模块时钟关闭。	0
14	USART1EN	RW	USART1 模块时钟使能位: 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 模块时钟使能位: 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	Reserved	RO	保留。	0
7	IOPFEN	RW	IO 的 PF 端口模块时钟使能位: 1: 模块时钟开启; 0: 模块时钟关闭。	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	HSADCEN	RW	HSADC 模块时钟使能位: 1: 模块时钟开启; 0: 模块时钟关闭。	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

3.4.9 HB1 外设时钟使能寄存器 (RCC_HB1PCENR)

偏移地址: 0x20

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

SWPMI EN	I2C3 EN	DAC EN	PWR EN	BKP EN	CAN2 EN	CAN1 EN	CAN3 EN	Reserved	I2C2 EN	I2C1 EN	USART5 EN	USART4 EN	USART3 EN	USART2 EN	SPI4 EN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	QSPI2 EN	QSPI1 EN	WWDG EN	LPTIM 2EN	LPTIM 1EN	USART 8EN	USART 7EN	USART 6EN	TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN

位	名称	访问	描述	复位值
31	SWPMIEN	RW	SWPMI 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
30	I2C3EN	RW	I2C3 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	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	CAN3EN	RW	CAN3 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
23	Reserved	RO	保留。	0
22	I2C2EN	RW	I2C2 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
21	I2C1EN	RW	I2C1 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
20	USART5EN	RW	USART5 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
19	USART4EN	RW	USART4 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
18	USART3EN	RW	USART3 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
17	USART2EN	RW	USART2 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
16	SPI4EN	RW	SPI4 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
15	SPI3EN	RW	SPI3 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
14	SPI2EN	RW	SPI2 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
13	QSPI2EN	RW	QSPI2 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0

12	QSPI1EN	RW	QSPI1 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	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	LPTIM2EN	RW	LPTIM2 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
9	LPTIM1EN	RW	LPTIM1 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
8	USART8EN	RW	USART8 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
7	USART7EN	RW	USART7 模块时钟使能位： 1：模块时钟开启； 0：模块时钟关闭。	0
6	USART6EN	RW	USART6 模块时钟使能位： 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： 1：模块时钟开启； 0：模块时钟关闭。	0
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

3.4.10 后备域控制寄存器 (RCC_BDCTLR)

偏移地址：0x24

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
RTCCAL[6:0]						RTCEN	RTCSEL[1:0]	ASOS	ASOE	CCO	LSEBYP	LSEREDY	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
[15:9]	RTCCAL[6:0]	RW	RTC 校准值寄存器，这个寄存器的值表示在每 2^{20} 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
8	RTCEN	RW	RTC 时钟使能控制：	0

			1: 使能 RTC 时钟; 0: 关闭 RTC 时钟。	
[7:6]	RTCSEL[1:0]	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 振荡器经过 512 分频后作为 RTC 时钟。	0
5	ASOS	RW	PC13 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
4	ASOE	RW	PC13 引脚使能脉冲输出位: 1: 使能输出闹钟脉冲或者秒脉冲; 0: 禁止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
3	CCO	RW	校准时钟输出选择位: 1: PC13 引脚输出经 64 分频的 RTC 时钟; 0: 不输出校准时钟。 注: 当 V_{DD33} 供电断开时, 该位被清除。	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: 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
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、CCO、ASOE、ASOS、RTCSEL、RTCEN 和 RTCCAL 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。这些位可由后备域复位清除。任何内部或外部复位都不会影响这些位。

3.4.11 控制/状态寄存器 (RCC_RSTSCKR)

偏移地址: 0x28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LKUP 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	LKUPRSTF	R0	LOCKUP 复位标志： 1：发生 LOCKUP 导致系统复位； 0：正常。 发生 LOCKUP 复位时由硬件置 1；软件写 RMVF 位清除。	0
30	WWDGRSTF	R0	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	R0	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	R0	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	R0	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	R0	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。 在 NRST 引脚复位发生时由硬件置 1；软件写 RMVF 位清除。	0
25	Reserved	R0	保留。	0
24	RMVF	W0	清除复位标志控制： 1：清除复位标志； 0：无作用。	0
[23:2]	Reserved	R0	保留。	0
1	LSIRDY	R0	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 BIT1 由上电复位清除，其他写清除复位标志可以清除。

3.4.12 HB 外设复位寄存器（RCC_HBRSTR）

偏移地址：0x2C

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

Reserved									PIOC RST	Reser ved	SERDE SRST	USBPD RST	UHSIF RST	OTGFS RST	Reser ved
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reser ved	ETHMA CRST	DVP RST	USBSS RST	USBHS RST	SDMMC RST	RNG RST	FMC RST	Reserved						DMA2 RST	DMA1 RST

位	名称	访问	描述	复位值
[31:22]	Reserved	RO	保留。	0
22	PIOCRST	RW	PIOC 模块复位控制： 1：复位模块； 0：无作用。	0
21	Reserved	RO	保留。	0
20	SERDESrst	RW	SERDES 模块复位控制： 1：复位模块； 0：无作用。	0
19	USBPDRST	RW	USBPD 模块复位控制： 1：复位模块； 0：无作用。	0
18	UHSIFRST	RW	UHSIF 模块复位控制： 1：复位模块； 0：无作用。	0
17	OTGFSRST	RW	USBOTG_FS 模块复位控制： 1：复位模块； 0：无作用。	0
[16:15]	Reserved	RO	保留。	0
14	ETHMACRST	RW	以太网 MAC 模块复位控制： 1：复位模块； 0：无作用。	0
13	DVPRST	RO	DVP 模块复位控制： 1：复位模块； 0：无作用。	0
12	USBSSRST	RW	USBSS 模块复位控制： 1：复位模块； 0：无作用。	0
11	USBHSRST	RW	USBHS 模块复位控制： 1：复位模块； 0：无作用。	0
10	SDMMCRST	RW	SDMMC 模块复位控制： 1：复位模块； 0：无作用。	0
9	RNGRST	RW	RNG 模块复位控制： 1：复位模块； 0：无作用。	0
8	FMC RST	RW	FMC 模块复位控制： 1：复位模块； 0：无作用。	0
[7:2]	Reserved	RO	保留。	0
1	DMA2RST	RW	DMA2 模块复位控制： 1：复位模块； 0：无作用。	0
0	DMA1RST	RW	DMA1 模块复位控制： 1：复位模块； 0：无作用。	0

3.4.13 时钟配置寄存器 2 (RCC_CFGR2)

偏移地址：0x30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ETH1GSR		HSADCSR		Reserved		I2S3S	I2S2S	RNGSR	Reserved		USBFS	USBFSDIV[3:0]			

[1:0]	[1:0]		RC	RC	C		SRC								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LTDCSRC [1:0]	LTDCDIV[5:0]					UHSIFSRC [1:0]		UHSIFDIV[5:0]							

位	名称	访问	描述	复位值
[31:30]	ETH1GSRC[1:0]	RW	千兆以太网 125M 时钟源选择： 00: PLL 时钟 (PLL_CLK)； 01: USBSS_PLL 时钟； 10: ETH_PLL 时钟 4 分频后输出； 11: SERDES_PLL 时钟 8 分频后输出。	0
[29:28]	HSADCSRC[1:0]	RW	HSADC 时钟源选择： 00: 系统时钟 (SYSCLK)； 01: PLL 时钟 (PLL_CLK)； 10: USBHS_PLL 时钟 (480MHz)； 11: ETH_PLL 时钟 (500MHz)。	0
[27:26]	Reserved	RO	保留。	0
25	I2S3SRC	RW	I2S3 时钟源： 1: PLL 时钟 (PLL_CLK)； 0: 系统时钟 (SYSCLK)。	0
24	I2S2SRC	RW	I2S2 时钟源： 1: PLL 时钟 (PLL_CLK)； 0: 系统时钟 (SYSCLK)。	0
23	RNGSRC	RW	RNG 时钟源选择： 1: PLL 时钟 (PLL_CLK)； 0: 系统时钟 (SYSCLK)。	0
[22:21]	Reserved	RO	保留。	
20	USBFSRC	RW	USBFS 48M 时钟源选择： 1: USBHS_PLL 时钟； 0: PLL 时钟。	0
[19:16]	USBFS DIV[3:0]	RW	USBFS 48M 时钟源预分频： 0000: 不分频； 0001: 2 分频； 0010: 3 分频； 0011: 4 分频； 0100: 5 分频； 0101: 6 分频； 0110: 8 分频； 0111: 10 分频； 1000: 1.5 分频； 1001: 2.5 分频； 1010: 3.5 分频； 1011: 4.5 分频； 1100: 5.5 分频； 1101: 6.5 分频； 1110: 7.5 分频； 1111: 9.5 分频。	0
[15:14]	LTDCSRC[1:0]	RW	LTDC 时钟源选择： 00: PLL 时钟 (PLL_CLK)； 01: SERDES_PLL 时钟 2 分频； 10: ETH_PLL 时钟； 11: USBHS_PLL 时钟。	0
[13:8]	LTDCDIV[5:0]	RW	LTDC 时钟来源预分频控制：	0

			000000: 不分频; 000001: 2 分频; 000010: 3 分频; 000011: 4 分频; 000100: 5 分频; ... 111110: 63 分频; 111111: 64 分频。	
[7:6]	UHSIFSRC[1:0]	RW	UHSIF 时钟源选择: 00: 系统时钟 (SYSCLK) ; 01: PLL 时钟 (PLL_CLK) ; 10: USBHS_PLL 时钟 (480MHz) ; 11: ETH_PLL 时钟 (500MHz) 。	0
[5:0]	UHSIFDIV[5:0]	RW	UHSIF 时钟来源预分频控制: 000000: 不分频; 000001: 2 分频; 000010: 3 分频; 000011: 4 分频; 000100: 5 分频; ... 111110: 63 分频; 111111: 64 分频。	0

3.4.14 PLL 配置寄存器 2 (RCC_PLLCFGR2)

偏移地址: 0x34

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

位	名称	访问	描述	复位值
[31:20]	Reserved	R0	保留。	0
[19:16]	SERDESPLL_MUL[3:0]	RW	SERDES_PLL 时钟倍频系数, 只有在 SERDES_PLLON 为 0 时才允许写: 0000: 25 倍频输出; 0001: 28 倍频输出; 0010: 30 倍频输出; 0011: 32 倍频输出; 0100: 35 倍频输出; 0101: 38 倍频输出; 0110: 40 倍频输出; 0111: 45 倍频输出; 1000: 50 倍频输出; 1001: 56 倍频输出; 1010: 60 倍频输出; 1011: 64 倍频输出; 1100: 70 倍频输出; 1101: 76 倍频输出; 1110: 80 倍频输出; 1111: 90 倍频输出。	1000b
[15:13]	Reserved	R0	保留。	0

[12:8]	USBSPLL_IN_DIV[4:0]	RW	USBHS_PLL 输入时钟源 SYS_PLL 的分频设置，只有在 USBHS_PLLON 为 0 时才允许写： 00000：不分频； 00001：2 分频； ... 11111：32 分频。	0
7	Reserved	RO	保留。	0
[6:4]	USBSSPLL_REFSEL[1:0]	RW	USBSS_PLL 参考时钟频率选择，只有在 USBSS_PLLON 为 0 时才允许写： 000：20MHz； 001：24MHz； 010：25MHz； 011：30MHz； 100：32MHz； 101：40MHz； 110：60MHz； 111：80MHz。	010b
[3:2]	USBHSPLL_REFSEL[1:0]	RW	USBHS_PLL 参考时钟频率选择，只有在 USBHS_PLLON 为 0 时才允许写： 00：25MHz； 01：20MHz； 10：24MHz； 11：32MHz。	0
[1:0]	USBHSPLLSRC[1:0]	RW	USBHS_PLL 时钟源选择： 00：HSE； 01：HSI； 10：ETHCLK_20M； 11：分频后的 SYS_PLL_CLK，分频系数参考 RCC_PLLCFGR2[12:8]。	0

第 4 章 中央处理器（CPU）

CH32H417 系列集成双内核结构，RISC-V5F 与 RISC-V3F 双内核均支持 32 位 I、C、M、A、B、F 和扩展指令集。

RISC-V3F 内核（C0/内核 0）内部采用顺序单发架构。取指模式可配，支持流水线模式和非流水线模式，在非流水线模式下可降低内核功耗。

RISC-V5F 内核（C1/内核 1）内部采用乱序双发射架构，并融合使用多种分支预测器，支持指令缓存，使得 V5 具有更出色的性能。此外 RISC-V5 实现了 WCH-X 扩展指令集，在代码密度，计算性能上均有所提高。

私有外设系统定时器、核间通信和硬件信号量模块；并支持外部调试器，调试器支持单/双线模式通信，单线通信速度四档可调。

4.1 主要特征

RISC-V3F

- 顺序单发架构
- 3 级流水线
- 低主频，低功耗，侧重于控制
- 提供一个不可屏蔽中断 NMI
- 4 通道物理内存保护（PMP）
- 4 通道硬件断点
- 2 级中断嵌套
- 快速可编程中断控制器（PFIC）
- 自定义扩展指令
- 硬件压栈

RISC-V5F

- 乱序多发架构
- 内置 32KB 的指令缓存
- 高主频，高性能，侧重于运算
- 提供一个不可屏蔽中断 NMI
- 4 通道物理内存保护（PMP）
- 4 通道硬件断点
- 8 级中断嵌套
- 快速可编程中断控制器（PFIC）
- 自定义扩展指令
- 硬件压栈

4.1.1 V5F 内核内存操作模型

V5F 内核是一款乱序超标量处理，其对存储器的访问具有重排序功能。

V5F 的存储器类型划分为 2 大类：

- **Normal**
CPU 对此类存储器的访问，会进行重排序，以提高性能。
- **Device and Strongly-ordered**
CPU 对此类存储器的访问，将严格按照程序顺序执行。

表 4-1 列出了存储器区域的默认属性及类型。内存保护单元（PMP）可对读（R）、写（W）、执行（X）的属性进行关闭。

表 4-1 存储器区域的默认属性及类型

	CODE	ITCM	DTCM	SRAM	PERI	MEM0	MEM1	DEV0	DEV1	PPB	RSV
属性	RWX	RWX	RWX	RWX	RW	RWX	RWX	RW	RW	RW	RW
总线	HB	TCM_BUS	TCM_BUS	HB	HB	HB	HB	HB	HB	HB	HB
类型	Normal	Normal	Normal	Normal	Device	Device	Device	Normal	Device	Device	Device

注：（1）TCM_BUS 是指在本芯片自定义的、适配与 TCM 存储器的总线。

（2）PPB 区域访问为固定优先级，V5 内核高于 V3 内核，须注意带宽限制对 V3 性能的影响。

各存储器区域的地址范围如表 4-2 所示：

表 4-2 存储器区域的地址范围

内存区域	地址范围
CODE	0x00000000 – 0x1FFFFFFF
ITCM	0x200A0000 – 0x200BFFFF
DTCM	0x200C0000 – 0x200FFFFF
SRAM	0x20100000 – 0x3FFFFFFF
PERI	0x40000000 – 0x5FFFFFFF
MEM0	0x60000000 – 0x7FFFFFFF
MEM1	0x80000000 – 0x9FFFFFFF
DEV0	0xA0000000 – 0xBFFFFFFF
DEV1	0xC0000000 – 0xDFFFFFFF
PPB	0xE0000000 – 0xE00FFFFF
RSV	其他地址

乱序内核中，实际的存储器访问顺序并非总是和程序中的访存顺序一致，为了提高访问效率，内核中会采取重排序，缓冲区，多总线接口等方式，在不违背数据相关性的前提下加快访存速度。

对于程序中的两条没有相关性的访存指令 A1, A2，假定 A1 指令顺序早于 A2 指令，且之间没有指令/数据同步指令，则 A1/A2 的实际请求顺序取决于 A1/A2 访问的内存类型（Normal、Device）及其指令类型（Load、Store）。可能存在的如表 4-3 的情况。

表 4-3 A1/A2 访问的内存类型及其指令类型

A1 \ A2	store device	store normal	load device	load normal
store device	order	re-order	order	re-order
store normal	re-order	re-order	re-order	re-order
load device	order	order	order	re-order
load normal	order	order	re-order	re-order

注：上文所述顺序、重排序均以 CPU 顶层接口总线操作顺序为准，对于所有乱序访问，均可通过插入 fence 指令强制顺序。

4.2 特权寄存器（CSR）

RISC-V3F 内核与 RISC-V5F 内核均符合 RISC-V 标准特权结构，支持 M 和 U 模式，调试模式等，内核中定义了部分标准寄存器和自定义寄存器。

4.2.1 通用特权寄存器

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。

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。

表 4-4 通用特权寄存器列表

名称	CSR 地址	描述	复位值
MVENDORID	0xF11	芯片制造商 ID 寄存器	0x00000000
MARCHID	0xF12	芯片结构 ID 寄存器	V3F: 0xDC68D86D V5F: 0xDC68D8AE
MIMPID	0xF13	内核 ID 寄存器	V3F: 0xDC688002 V5F: 0xDC688001
MHARTID	0xF14	内核 ID 寄存器	V3F: 0x00000000 V5F: 0x00000001
MSTATUS	0x300	机器状态寄存器	0x00000000
MISA	0x301	机器指令集寄存器	0x40901127
MTVEC	0x305	机器模式异常基地址寄存器	0x00000000
MSCRATCH	0x340	机器模式临时寄存器	0xFFFFFFFF
MEPC	0x341	机器异常程序计数器寄存器	0xFFFFFFFF
MCAUSE	0x342	机器异常寄存器	0x000000XX
MTVAL	0x343	机器自陷值寄存器	0xFFFFFFFF
TINFO	0x7A4	触发器信息寄存器	0x00000004
DPC	0x7B1	调试 PC 地址寄存器	0xFFFFFFFF
DSCRATCH0	0x7B2	调试模式临时寄存器 0	0xFFFFFFFF
DSCRATCH1	0x7B3	调试模式临时寄存器 1	0xFFFFFFFF
DBGMCU_0	0x7C0	调试寄存器 0	0x00000000
FFLAGS	0x001	浮点累计异常标记寄存器	0x00000000
FRM	0x002	浮点舍入模式寄存器	0x00000000
FCSR	0x003	浮点控制和状态寄存器	0x00000000
UACCES_MSTATUS	0x800	用户访问机器状态寄存器	0x00000000
HW_POPDM_CTLR	0x804	硬件压栈控制寄存器	0x00000004
TSELECT	0x7A0	触发器通道选择寄存器	0x0000000X
TDATA1	0x7A1	跟踪触发数据寄存器	0x23E00048
TDATA2	0x7A2	保存触发数据寄存器	0xFFFFFFFF

4.2.1.1 芯片制造商 ID 寄存器（MVENDORID）

CSR 地址：0xF11

位	名称	访问	描述	复位值
[31:0]	mvendorid[31:0]	MRO	芯片制造商 ID，固定值为 0。	0

4.2.1.2 芯片结构 ID 寄存器（MARCHID）

CSR 地址：0xF12

位	名称	访问	描述	复位值
[31:0]	marchid[31:0]	MRO	芯片结构 ID，固定值不可修改。	V3F: 0xDC68D86D

			V5F: 0xDC68D8AE

4.2.1.3 机器实现 ID 寄存器 (MIMPID)

CSR 地址: 0xF13

位	名称	访问	描述	复位值
[31:0]	mimpid[31:0]	MRO	机器实现 ID, 固定值不可修改。	V3F: 0xDC688002 V5F: 0xDC688001

4.2.1.4 内核 ID 寄存器 (MHARTID)

CSR 地址: 0xF14

位	名称	访问	描述	复位值
[31:0]	mhartid[31:0]	MRO	内核 ID, 表明本内核在芯片结构中的内核 ID 号。	V3F: 0 V5F: 1

4.2.1.5 机器状态寄存器 (MSTATUS)

CSR 地址: 0x300

位	名称	访问	描述	复位值
31	SD	MRO	用于标记浮点指令是否处于脏状态。	0
[30:15]	Reserved	MRO	保留。	0
[14:13]	FS[1:0]	MRW	浮点指令集状态, 值不为 0 时允许单精度浮点指令 (F) 执行: 00: OFF; 01: Initial; 10: Clean; 11: Dirty。	0
[12:11]	MPP[1:0]	MRW	保存/恢复进入异常前的机器状态, 进入异常时更新为异常前的机器状态: 00: 退出异常时机器状态设置为 U 模式; 01: 保留; 10: 保留; 11: 退出异常时机器状态设置为 M 模式。	0
[10:8]	Reserved	MRO	保留。	0
7	MPIE	MRW	保存进入中断时的中断使能状态: 1: 退出中断后使能全局中断; 0: 退出中断后关闭全局中断。	0
[6:4]	Reserved	MRO	保留。	0
3	MIE	MRW	全局中断使能, 退出中断时更新为 MPIE 值: 1: 全局中断使能, 允许处理器响应中断请求; 0: 全局中断关闭, 不允许处理器响应中断请求。	0
[2:0]	Reserved	MRO	保留。	0

4.2.1.6 机器指令集寄存器 (MISA)

CSR 地址: 0x301

位	名称	访问	描述	复位值
[31:30]	XLEN[1:0]	MRO	机器基本整数位宽和基本指令位宽信息, 32 位宽内核, 固定值为 1。	0x1
[29:24]	Reserved	MRO	保留。	0
23	X	MRO	本内核支持执行自定义指令集。	1
[22:21]	Reserved	MRO	保留。	0
20	U	MRO	支持用户模式运行。	1
[19:13]	Reserved	MRO	保留。	0
12	M	MRO	本内核支持执行乘法指令集。	1
[11:9]	Reserved	MRO	保留。	0
8	I	MRO	本内核支持执行基本整数指令集。	1
[7:6]	Reserved	MRO	保留。	0
5	F	MRO	本内核支持执行单精度浮点指令集。	1
[4:3]	Reserved	MRO	保留。	0
2	C	MRO	本内核支持执行压缩指令集。	1
1	B	MRO	本内核支持执行位操作指令集。	1
0	A	MRO	本内核支持执行原子指令集。	1

4.2.1.7 机器模式异常基地址寄存器 (MTVEC)

CSR 地址: 0x305

位	名称	访问	描述	复位值
[31:10]	mtvec_base[21:0]	MRW	处理器发生异常时跳转的基地址高位, 1kB 对齐, 实际 32 位基地址 BASE = {mtvec_base, 10'h0}。	0
[9:2]	Reserved	MRO	保留。	0
1	mode1	RW	向量表模式 1: 1: 中断向量表中存放跳转地址; 0: 中断向量表中存放跳转指令。	0
0	mode0	RW	向量表模式 0: 1: 所有中断/异常独立入口, 跳转地址为 BASE+4*IRQ/EXC ID; 0: 所有中断/异常统一入口, 均跳转到异常跳转基地址。	0

4.2.1.8 机器模式临时寄存器 (MSCRATCH)

CSR 地址: 0x340

位	名称	访问	描述	复位值
[31:0]	mscratch[31:0]	MRW	机器模式临时寄存器。	x

4.2.1.9 机器异常程序计数器寄存器 (MEPC)

CSR 地址: 0x341

位	名称	访问	描述	复位值

[31:0]	mepc[31:0]	MRW	保存退出中断/异常时恢复的 PC 地址， mepc[0] 固定为 0。	x

4.2.1.10 机器异常寄存器 (MCAUSE)

CSR 地址：0x342

位	名称	访问	描述	复位值
31	Interrupt	RW	中断/异常标志位： 1：异常/中断由外部产生； 0：异常/中断由内核产生。	x
[30:8]	Reserved	RO	保留。	0
[7:0]	Exception Code[7:0]	RW	保存产生异常或中断的源信息。	x

中断异常源编码：

Interrupt/Exception	Exception Code	中断/异常源
1	2	保留 NMI
1	3	机器模式软件中断
1	4	用户模式定时器中断
1	5	监视模式定时器中断
1	6	保留
1	7	机器模式定时器中断
1	8	用户模式外部中断
1	11	机器模式外部中断
1	12	系统定时器中断
1	≥16	保留平台中断 id
0	0	指令地址非对齐
0	1	指令访问错误
0	2	非法指令
0	3	断点
0	4	读地址非对齐
0	5	读访问错误
0	6	写/原子操作地址非对齐
0	7	写/原子操作访问错误
0	8	用户模式 ecall
0	11	机器模式 ecall
0	≥16	保留

4.2.1.11 机器自陷值寄存器 (MTVAL)

CSR 地址：0x343

位	名称	访问	描述	复位值
[31:0]	mtval[31:0]	RW	保存自陷的附加信息。	x

mtval 寄存器在 trap 中保存的附加信息：

异常类型	Mcause	保存信息
指令地址非对齐	0	指令 PC

指令访问失败	1	指令 PC
非法指令	2	指令机器码
读地址非对齐	4	读数据地址
读访问失败	5	读数据地址
写/原子访问失败	6	写数据地址
写/原子地址非对齐	7	写数据地址

4.2.1.12 触发器信息寄存器 (TINFO)

CSR 地址: 0x7A4

位	名称	访问	描述	复位值
[31:0]	tinfo[31:0]	MRO	触发器信息寄存器, 对应 TDATA1 寄存器 type 值, type 值为 2, tinfo 的第 2 位置 1。	0x4

4.2.1.13 调试 PC 地址寄存器 (DPC)

CSR 地址: 0x7B1

位	名称	访问	描述	复位值
[31:0]	dpc[31:0]	RW	保存退出调试模式时恢复的 PC 地址。	x

不同调试请求源引起的跳转到调试模式, dpc 保存的地址如下表:

诱因	保存地址
ebreak	Ebreak 的指令地址。
单步	保存在非调试模式时下一条应该执行的指令地址 (仅 V5F)。
触发模块	tdata1.timing=0: 触发断点的指令地址 (仅 V5F)。
暂停请求	进入调试模式时下一条应该执行的指令地址。

4.2.1.14 调试模式临时寄存器 0 (DSCRATCH0)

CSR 地址: 0x7B2

位	名称	访问	描述	复位值
[31:0]	Dscratch0[31:0]	RW	调试模式临时寄存器 0。	x

4.2.1.15 调试模式临时寄存器 1 (DSCRATCH1)

CSR 地址: 0x7B3

位	名称	访问	描述	复位值
[31:0]	Dscratch1[31:0]	RW	调试模式临时寄存器 1。	x

4.2.1.16 调试寄存器 0 (DBGMCU_0)

CSR 地址: 0x7C0

位	名称	访问	描述	复位值
[31:8]	dbg_mode_stop	MRW	进入调试模式时注册模块停止工作: 1: 注册的外设在内核进入调试模式时停止工作; 0: 注册的外设在内核进入调试模式时正常工作。	0

[7:0]	保留	MRO	保留。	0

4.2.1.17 浮点累计异常标记寄存器 (FFLAGS)

CSR 地址：0x001

位	名称	访问	描述	复位值																		
[31:5]	Reserved	R0	保留。	0																		
[4:0]	FFlags[4:0]	RW	累计异常标记，用于表示自上次重置以来（复位或软件清除）在浮点运算指令上出现的异常。 <table><tr><th>name</th><th>bit</th><th>description</th></tr><tr><td>NV</td><td>4</td><td>非法操作</td></tr><tr><td>DZ</td><td>3</td><td>除数为零</td></tr><tr><td>OF</td><td>2</td><td>向上溢出</td></tr><tr><td>UF</td><td>1</td><td>向下溢出</td></tr><tr><td>NX</td><td>0</td><td>不精确</td></tr></table>	name	bit	description	NV	4	非法操作	DZ	3	除数为零	OF	2	向上溢出	UF	1	向下溢出	NX	0	不精确	0
name	bit	description																				
NV	4	非法操作																				
DZ	3	除数为零																				
OF	2	向上溢出																				
UF	1	向下溢出																				
NX	0	不精确																				

4.2.1.18 浮点舍入模式寄存器 (FRM)

CSR 地址：0x002

位	名称	访问	描述	复位值
[31:3]	Reserved	RO	保留。	0
[2:0]	Frm[2:0]	RW	浮点操作舍入模式寄存器： 000：向最近值舍入（处于中间值时向偶数方向舍入）； 001：向零舍入； 010：向下舍入； 011：向上舍入； 100：向最近值舍入（处于中间值时向最大量级舍入）； 101、110：保留； 111：动态舍入，根据指令中的 FRM 位配置值选择舍入模式。	0

4.2.1.19 浮点控制和状态寄存器 (FCSR)

CSR 地址：0x003

位	名称	访问	描述	复位值
[31:8]	Reserved	RO	保留。	0
[7:5]	Frm[2:0]	RW	浮点操作舍入模式寄存器。	0
[4:0]	FFlags[4:0]	RW	累计异常标记。	0

4.2.1.20 用户访问机器状态寄存器 (UACCES_MSTATUS)

CSR 地址：0x800

位	名称	访问	描述	复位值
31	SD	RO	用于标记浮点指令是否处于脏状态。	0
[30:15]	Reserved	RO	保留。	0

[14:13]	FS[1:0]	RO	浮点指令集状态，值不为 0 时允许单精度浮点指令（F）执行： 00: OFF; 01: Initial; 10: Clean; 11: Dirty。	0
[12:11]	MPP[1:0]	RO	保存/恢复进入异常前的机器状态，进入异常时更新为异常前的机器状态： 00: 退出异常时机器状态设置为 U 模式； 01: 保留； 10: 保留； 11: 退出异常时机器状态设置为 M 模式。	0
[10:8]	Reserved	RO	保留。	0
7	MPIE	RW	保存进入中断时的中断使能状态： 1: 退出中断后使能全局中断； 0: 退出中断后关闭全局中断。	0
[6:4]	Reserved	RO	保留。	0
3	MIE	RW	全局中断使能，退出中断时更新为 MPIE 值： 1: 全局中断使能，允许处理器响应中断请求； 0: 全局中断关闭，不允许处理器响应中断请求。	0
[2:0]	Reserved	RO	保留。	0

注：本寄存器用于内核在用户模式运行时配置 MSTATUS 寄存器，当 CPU_RUN_CTLR 寄存器 non_usta 位使能后本地址可用，寄存器设置与 MSTATUS（0x300）寄存器相同。

4.2.1.20 硬件压栈控制寄存器（HW_POPDM_CTLR）

CSR 地址：0x804

位	名称	访问	描述	复位值
31	lock	RW	用户模式锁定标志，锁定后用户模式不可改写本组寄存器配置： 1: 锁定，仅机器模式可配置； 0: 非锁定，机器/用户模式可配置。	0
[30:6]	Reserved	RO	保留。	0
5	hw_pop_off	RW	一次性硬件出栈关闭，退出中断后自动复位： 1: 下一次退出中断时屏蔽硬件出栈； 0: 不屏蔽硬件出栈。	0
4	Reserved	RO	保留。	0
[3:2]	preempt[1:0]	RW	抢占优先级位宽配置寄存器，用于配置中断优先级中抢占优先级位宽： 00: 抢占位宽为 0，任何优先级的中断无法嵌套；	0x1

			01: 抢占位宽为 1, 即中断优先级寄存器的[7]位为抢占优先级; 10: 抢占位宽为 2, 即中断优先级寄存器的[7:6]位为抢占优先级; 11: 抢占位宽为 3, 即中断优先级寄存器的[7:5]位为抢占优先级。	
1	nest_en	RW	中断嵌套使能寄存器: 1: 允许中断嵌套 0: 不允许中断嵌套	0
0	hw_stk_en	RW	硬件堆栈保护使能: 1: 进出中断时使用硬件堆栈保护; 0: 进出中断时不使用硬件堆栈保护。	0

4.2.1.22 触发器通道选择寄存器 (TSELECT)

CSR 地址: 0x7A0

位	名称	访问	描述	复位值
[31:2]	Reserved	MRO	保留。	0
[1:0]	mscratch[1:0]	MRW	触发器通道选择寄存器。	x

注: (1) 内核 1 均实现四通道触发器, 通过 TSELECT 寄存器选择触发器通道进行配置, TSELECT 触发器低 2 位有效, 其余位恒为 0。

(2) 内核 0 的 TSELECT 寄存器仅适用于批号第五位不为 0 的芯片。

4.2.1.23 跟踪触发数据寄存器 (TDATA1)

CSR 地址: 0x7A1

位	名称	访问	描述	复位值
[31:28]	type[3:0]	MRO	触发器类型定义。	0x2
27	dmode	MRO	机器模式和调试模式均可修改触发器相关寄存器。	0
[26:21]	maskmax	MRO	当 mach=1 时, 允许匹配的最大指数幂范围。	0x1F
[20:19]	Reserved	MRO	保留。	0
18	timing	MRO	在指令/操作执行前触发。	0
[17:13]	Reserved	MRO	保留。	0
12	action	MRW	设置触发时采取的处理模式: 1: 触发时进入到调试模式, dpc 保存退出调试时需要恢复的指令地址; 0: 触发时产生一个将线程引导到 M 模式的异常。	0
[11:8]	Reserved	MRO	保留。	0
7	match	MRO	匹配策略配置: 允许比较值与 tdata2 值相等的地址触发。	0
6	m	MRO	M 模式下触发器使能: 1: 使能; 0: 关闭。	1

[5:4]	Reserved	MRO	保留。	0
3	u	MRO	U 模式下触发器使能： 1：使能； 0：关闭。	1
2	execute	MRW	根据执行指令的虚拟地址触发 1：触发器比较指令地址； 0：触发器不比较指令地址。	0
1	store	MRW	根据存储数据的虚拟地址触发 1：触发器比较写数据地址； 0：触发器不比较写数据地址。	0
0	load	MRW	根据读取数据的虚拟地址触发 1：触发器比较读数据地址； 0：触发器不比较读数据地址	0

注：（1）处理器触发配置符合 RISC-V 调试标准，触发器配置类型为 2，根据访问的虚拟地址的最低位进行匹配，在匹配的指令或读写数据前执行触发操作。

（2）内核 0 的 TDATA1 寄存器仅适用于批号第五位不为 0 的芯片。

4.2.1.24 保存触发数据寄存器（TDATA2）

CSR 地址：0x7A2

位	名称	访问	描述	复位值
[31:0]	tdata2[31:0]	MRW	用于保存触发数据（指令 PC 或数据地址）。	x

注：内核 0 的 TDATA2 寄存器仅适用于批号第五位不为 0 的芯片。

4.2.2 RISC-V3F 内核自定义寄存器

表 4-5 RISC-V3F 内核自定义寄存器列表

名称	CSR 地址	描述	复位值
CPU_RUN_CTLR	0xBC0	处理器运行控制寄存器	0x12370000
INEST_CTLR	0xBC1	中断嵌套控制寄存器	0x00000000
MIE	0xBC8	机器中断寄存器	0x00000000
DCSR	0x7B0	调试控制和状态寄存器	0x40000000

4.2.2.1 处理器运行控制寄存器（CPU_RUN_CTLR）

CSR 地址：0xBC0

位	名称	访问	描述	复位值
[31:28]	fadd_clkdiv[3:0]	MRW	浮点加指令分频系数，默认值为 1，即浮点加指令频率为主频的 2 分频。	0x1
[27:24]	fmul_clkdiv[3:0]	MRW	浮点乘指令分频系数，默认值为 2，即浮点乘指令频率为主频的 3 分频。	0x2
[23:20]	fmac_clkdiv[3:0]	MRW	浮点乘加指令分频系数，默认值为 3，即浮点乘加指令频率为主频的 4 分频。	0x3
[19:16]	fdiv_clkdiv[3:0]	MRW	浮点除指令分频系数，默认值为 7，即浮点除指令频率为主频的 8 分频。	0x7
7	nmi_ie	MRW	不可屏蔽中断使能： 1：发生异常导致嵌套溢出时进入 NMI 中断；	0

			0: 发生异常导致嵌套溢出时不进入 NMI 中断。	
6	int_fence	MRW	fence 屏蔽中断使能: 1: 执行 fence 指令时清除中断请求; 0: 执行 fence 指令时不清除中断请求。	0
5	non_usta	MRW	UACCES_MSTATUS 寄存器使能: 1: CSR 地址 0x800 的位 3 和位 7 分别映射为 MSTATUS 寄存器的位 MIE 和 MSTATUS 寄存器的位 MPIE; 0: CSR 地址 0x800 为只读寄存器, 返回值为 MSTATUS 的值。	0
[4:2]	Reserved	MRO	保留。	0
[1:0]	pipe_acc[1:0]	MRW	取指模式: 00: 预取关闭。指令预取功能关闭, 避免无效的取指操作, CPU 流水线上至多存在 1 条有效指令。此模式功耗最低, 性能下降约 2~3 倍。 01: 预取使能。指令预取功能打开, CPU 将持续对指令存储器进行访问, 直到内部指令缓冲器待执行指令数量超过一定数量, 或者指令缓冲器已满时, 暂停取指令, 此模式功耗高, 性能强; (CPU 预测失败将引入冗余取指操作, 在部分情况下, 执行单元将额外引入 0~2 个周期的气泡, 大部分程序性能下降不明显); 其他: 保留。	0

内核	fadd_freq (max)	fmul_freq (max)	fmac_freq (max)	fdiv_freq (max)
V3F	80MHz	53MHz	40MHz	20MHz

浮点算数指令为多周期指令, 对于不同类型的浮点指令, 最高运行频率如上表所示。根据处理器的运行频率, 用户需配置浮点运算分频系数, 使分频后的频率不高于上表所示的最大运行频率。

$$f_{xxx\_freq(max)} \geq \text{内核主频} / f_{xxx\_clkdiv}$$

4.2.2.2 中断嵌套控制寄存器 (INEST_CTLR)

CSR 地址: 0xBC1

位	名称	访问	描述	复位值
31	Reserved	R0	保留。	0
30	nest_ovr	MRW	中断/异常嵌套溢出标志位, 当嵌套深度超过最大深度时被置 1, 写 1 清零。 注: 中断溢出仅会在执行二级中断服务函数产生指令异常和 NMI 中断时发生。此时异常和 NMI 中断正常进入, 但是 CPU 堆栈溢出, 不可从此异常和 NMI 中断退出。	0
[29:12]	Reserved	MRO	保留。	0
[11:8]	nest_sta[3:0]	MRO	嵌套状态标志位: 0000: 无中断; 0001: 一级中断;	0

			0011：二级中断嵌套； 0111：三级中断嵌套，溢出； 其他：保留。	
[7:2]	Reserved	MRO	保留。	0
[1:0]	nest_max[1:0]	MRW	最大嵌套等级： 00：禁止嵌套，（关闭嵌套功能）； 01：二级嵌套，（打开嵌套功能）； 10：无法写入； 11：无法写入。 注：（1）对该域写 10b 或者 11b，寄存器被置为 01b； （2）写入 11b 时，读该寄存器，可获得芯片的最高嵌套等级。	0

4.2.2.3 机器中断寄存器（MIE）

CSR 地址：0xBC8

位	名称	访问	描述	复位值
[31:5]	Reserved	R0	保留。	0
[4:0]	nest_mie[4:0]	RW	机器中断寄存器,保存多级中断嵌套中每级中断的 mie 信息： status.mie=nest_mie[0]； status.mpie=nest_mie[1]。	0

4.2.2.4 调试控制和状态寄存器（DCSR）

CSR 地址：0x7B0

位	名称	访问	描述	复位值
[31:28]	exdebug[3:0]	MRO	存在符合调试器协议的外部调试器。	0x4
[27:16]	Reserved	MRO	保留。	0
15	ebreakm	MRW	机器模式下 ebreak 指令进入调试模式使能： 1：机器模式下运行 ebreak 指令，处理器进入调试模式； 0：机器模式下运行 ebreak 指令，处理器进入异常。	0
[14:13]	Reserved	MRO	保留。	0
12	ebreaku	MRW	用户模式下 ebreak 指令进入调试模式使能： 1：用户模式下运行 ebreak 指令，处理器进入调试模式； 0：用户模式下运行 ebreak 指令，处理器进入异常。	0
[11:10]	Reserved	MRO	保留。	0
9	stoptime	MRW	计时器暂停使能： 1：调试模式中计数器停止计数； 0：调试模式中计数器正常工作。	0

[8:6]	dcause[2:0]	MRW	调试源寄存器，用于标记引起进入调试模式的具体事件 000：复位值； 001：程序中 ebreak 指令触发调试（优先级 3）； 010：触发模块引起一个断点异常（优先级 4）（仅 V5F）； 011：调试器产生暂停请求（优先级 1）； 100：单步模式暂停（优先级 0，最低）（仅 V5F）； 101：未实现； 110：未实现； 111：保留。	0
[5:4]	Reserved	MRO	保留。	0
3	nmip	MRO	内核 NMI 中断挂起标志。	0
2	Reserved	MRO	保留。	0
[1:0]	prv[1:0]	MRW	记录进入到调试模式前的机器状态： 00：从用户模式进入调试模式，恢复时处理器恢复到用户模式； 01、10：保留； 11：从机器模式进入调试模式，恢复时处理器恢复到机器模式。	0

4.2.3 RISC-V5F 内核自定义寄存器

表 4-6 RISC-V5F 内核自定义寄存器列表

名称	CSR 地址	描述	复位值
MCOUNT_INHIBIT	0x320	计数器屏蔽寄存器	0x00000000
DCSR	0x7B0	调试控制和状态寄存器	0x40000000
MCYCLE	0xB00	机器周期计数器寄存器	0x00000000
MINSTRET	0xB02	机器指令计数器寄存器	0x00000000
UCYCLE	0xC00	用户模式周期计数器寄存器	0x00000000
UINSTRET	0xC02	用户模式指令计数器寄存器	0x00000000
CPU_RUN_CTLR	0xBC0	处理器运行控制寄存器	0x12370300
INEST_CTLR	0xBC1	中断嵌套控制寄存器	0x00000000
CACHE_STRTG_CTLR	0xBC2	缓存策略控制寄存器	0x0F000003
CACHE_PMP_OVR	0xBC3	缓存策略 PMP 覆盖寄存器	0x00000000
HW_POPDM_ADDR	0xBC4	硬件压栈地址寄存器	0x20060000
MEMORY_CFGR	0xBC5	存储器配置寄存器	0x600F0FFF
TCM_RRDUTY_CFGR	0xBC6	TCM 优先访问时长配置寄存器	0x00000000
MIE	0xBC8	机器中断寄存器	0x00000000
OPCACHE_CTLR	0xBD0	缓存操作寄存器	0x00000000
MEMINFO	0xFC0	内存信息寄存器	0x12220030

4.2.3.1 计数器屏蔽寄存器（MCOUNT_INHIBIT）

CSR 地址：0x320

位	名称	访问	描述	复位值
[31:3]	Reserved	MRO	保留。	0
2	IR	MRW	指令计数器屏蔽寄存器： 1：屏蔽指令计数器，minstret 和 unstret 停止计数； 0：不屏蔽指令计数器，minstret 和 unstret 正常计数。	0
1	Reserved	MRO	保留。	0
0	CY	MRW	周期计数器屏蔽寄存器： 1：屏蔽指令计数器，mcycle 和 ucycle 停止计数； 0：不屏蔽周期计数器，mcycle 和 ucycle 正常计数。	0

4.2.3.2 调试控制和状态寄存器（DCSR）

CSR 地址：0x7B0

位	名称	访问	描述	复位值
[31:28]	exdebug[3:0]	MRO	存在符合调试器协议的外部调试器。	0x4
[27:16]	Reserved	MRO	保留。	0
15	ebreakm	MRW	机器模式下 ebreak 指令进入调试模式使能： 1：机器模式下运行 ebreak 指令，处理器进入调试模式； 0：机器模式下运行 ebreak 指令，处理器进入异常。	0
[14:13]	Reserved	MRO	保留。	0
12	ebreaku	MRW	用户模式下 ebreak 指令进入调试模式使能： 1：用户模式下运行 ebreak 指令，处理器进入调试模式； 0：用户模式下运行 ebreak 指令，处理器进入异常。	0
11	stepie	MRW	单步模式中断使能： 1：在单步模式中允许响应中断； 0：在单步模式中禁止响应中断。	0
10	Reserved	MRO	保留。	0
9	stoptime	MRW	计时器暂停使能： 1：调试模式中计数器停止计数； 0：调试模式中计数器正常工作。	0
[8:6]	dcause[2:0]	MRW	调试源寄存器，用于标记引起进入调试模式的具体事件 000：复位值； 001：程序中 ebreak 指令触发调试（优先级 3）； 010：触发模块引起一个断点异常（优先	0

			级 4) (仅 V5F) ; 011: 调试器产生暂停请求 (优先级 1) ; 100: 单步模式暂停 (优先级 0, 最低) (仅 V5F) ; 101: 未实现; 110: 未实现; 111: 保留。	
[5:4]	Reserved	MRO	保留。	0
3	nmip	MRO	内核 NMI 中断挂起标志。	0
2	step	MRW	单步模式: 1: 单步模式; 0: 非单步模式。	0
[1:0]	prv[1:0]	MRW	记录进入到调试模式前的机器状态: 00: 从用户模式进入调试模式, 恢复时处理器恢复到用户模式; 01、10: 保留; 11: 从机器模式进入调试模式, 恢复时处理器恢复到机器模式。	0

4.2.3.3 机器周期计数器寄存器 (MCYCLE)

CSR 地址: 0xB00

位	名称	访问	描述	复位值
[31:0]	mcycle[31:0]	MRW	机器周期计数器, 用于计数机器工作的时钟周期, MCOUNT_INHIBIT 寄存器 CY 位为 0 时计数, 调试模式中计数器停止时不计数。	0

4.2.3.4 机器指令计数器寄存器 (MINSTRET)

CSR 地址: 0xB02

位	名称	访问	描述	复位值
[31:0]	minstret[31:0]	MRW	指令计数器, 用于计数机器正确执行完成的指令数, MCOUNT_INHIBIT 寄存器 IR 位为 0 时计数, 调试模式中计数器停止时不计数。	0

4.2.3.5 用户模式周期计数器寄存器 (UCYCLE)

CSR 地址: 0xC00

位	名称	访问	描述	复位值
[31:0]	ucycle[31:0]	RO	用户模式周期计数器, 用于计数机器工作的时钟周期, MCOUNT_INHIBIT 寄存器 CY 位为 0 时计数, 调试模式中计数器停止时不计数。	0

注: 用户模式周期计数器, 功能同 *mcycle* 寄存器, 用于在用户模式下读取机器运行周期数。

4.2.3.6 用户模式指令计数器寄存器 (UINSTRET)

CSR 地址: 0xC02

位	名称	访问	描述	复位值
[31:0]	uinstret[31:0]	RO	指令计数器, 用于计数机器正确执行完成的指	0

			令数，MCOUNT_INHIBIT 寄存器 IR 位为 0 时计数，调试模式中计数器停止时不计数。	

注：用户模式指令计数器，功能同 *minstret* 寄存器，用于在用户模式下读取机器正确执行的指令数。

4.2.3.7 处理器运行控制寄存器（CPU_RUN_CTLR）

CSR 地址：0xBC0

位	名称	访问	描述	复位值
[31:28]	fadd_clkdiv[3:0]	MRW	浮点加指令分频系数，默认值为 1，即浮点加指令频率为主频的 2 分频。	0x1
[27:24]	fmul_clkdiv[3:0]	MRW	浮点乘指令分频系数，默认值为 2，即浮点乘指令频率为主频的 3 分频。	0x2
[23:20]	fmac_clkdiv[3:0]	MRW	浮点乘加指令分频系数，默认值为 3，即浮点乘加指令频率为主频的 4 分频。	0x3
[19:16]	fdiv_clkdiv[3:0]	MRW	浮点除指令分频系数，默认值为 7，即浮点除指令频率为主频的 8 分频。	0x7
15	nlp_en	MRW	下一行分支预测功能使能： 1：使能； 0：关闭。	0
14	ghr_en	MRW	全局历史寄存器功能使能： 1：分支全局历史寄存器使能； 0：分支全局历史寄存器禁用。	0
[13:11]	Reserved	MRO	保留。	0
10	lsu_dual	MRW	访存指令并行发射使能，置 1 时允许发射单元并行发射访存指令，提高访存指令发射速度。	0
[9:8]	Reserved	MRO	保留。	0
7	nmi_ie	MRW	不可屏蔽中断使能： 1：发生异常导致嵌套溢出时进入 NMI 中断； 0：发生异常导致嵌套溢出时不进入 NMI 中断。	0
6	int_fence	MRW	fence 屏蔽中断使能： 1：执行 fence 指令时清除中断请求； 0：执行 fence 指令时不清除中断请求。	0
5	non_usta	MRW	UACCES_MSTATUS 寄存器使能： 1：CSR 地址 0x800 的位 3 和位 7 分别映射为 MSTATUS 寄存器的位 MIE 和 MSTATUS 寄存器的位 MPIE； 0：CSR 地址 0x800 为只读寄存器，返回值为 MSTATUS 的值。	0
[4:0]	Reserved	MRO	保留。	0

内核	fadd_freq (max)	fmul_freq (max)	fmac_freq (max)	fdiv_freq (max)
V5F	128MHz	96MHz	76MHz	38MHz

浮点算数指令为多周期指令，对于不同类型的浮点指令，最高运行频率如上表所示。根据处理器的运行频率，用户需配置浮点运算分频系数，使分频后的频率不高于上表所示的最大运行频率。

$$f_{xxx\_freq(max)} \geq \text{内核主频} / f_{xxx\_clkdiv}$$

4.2.3.8 中断嵌套控制寄存器（INEST_CTLR）

CSR 地址：0xBC1

位	名称	访问	描述	复位值
31	lsu_nmi_sta	MRW	访存单元产生不可屏蔽中断标记（通常为写总线返回错误），写 1 复位。	0
30	nest_ovr_sta	MRW	中断/异常嵌套溢出标志位，当嵌套深度超过最大深度时被置 1，写 1 清零。 注：中断溢出仅会在执行内核允许的最大中断级数服务函数产生指令异常和 NMI 中断时发生。此时异常和 NMI 中断正常进入，但是 CPU 堆栈溢出，不可从此异常和 NMI 中断退出。	0
[29:16]	Reserved	MRO	保留。	0
[15:8]	nest_sta	MRO	嵌套状态标志位： [0x00]：无中断； [0x01]：一级中断； [0x03]：二级中断嵌套； [0x07]：三级中断嵌套； [0x0F]：四级中断嵌套； [0x1F]：五级中断嵌套； [0x3F]：六级中断嵌套； [0x7F]：七级中断嵌套； [0xFF]：八级中断嵌套； 其他：保留。	0
[7:3]	Reserved	MRO	保留。	0
[2:0]	nest_max	MRW	最大嵌套等级： 000：禁止嵌套，（关闭嵌套功能）； 001：允许二级嵌套； 010：允许三级嵌套； 011：允许四级嵌套； 100：允许五级嵌套； 101：允许六级嵌套； 110：允许七级嵌套； 111：允许八级嵌套；	0

4.2.3.9 缓存策略控制寄存器（CACHE_STRTG_CTLR）

CSR 地址：0xBC2

位	名称	访问	描述	复位值
[31:28]	Reserved	RO	保留。	0
27	ic_mem1_strtg	MRW	对 0x80000000-0xffffffff 区域的指令缓存使能： 1：允许对本区域内指令进行缓存；	1

			0: 禁止对本区域内指令进行缓存。	
26	ic_mem0_strtg	MRW	对 0x60000000-0x7fffffff 区域的指令缓存使能。	1
25	ic_sram_strtg	MRW	对 0x20000000-0x3fffffff 区域的指令缓存使能。	1
24	ic_code_strtg	MRW	对 0x00000000-0x1fffffff 区域的指令缓存使能。	1
[23:2]	Reserved	RO	保留。	0
1	ic_disable	MRW	指令缓存失能标志位, 为 0 开启指令缓存功能。	1
0	Reserved	RO	保留。	1

4.2.3.10 缓存策略 PMP 覆盖寄存器 (CACHE_PMP_OVR)

CSR 地址: 0xBC3

位	名称	访问	描述	复位值
[31:13]	Reserved	MRO	保留。	0
12	ic_pmp3cache_strtg	MRW	PMP 通道 3 地址区域的指令缓存策略寄存器。	0
[11:9]	Reserved	MRO	保留。	0
8	ic_pmp2cache_strtg	MRW	PMP 通道 2 地址区域的指令缓存策略寄存器。	0
[7:5]	Reserved	MRO	保留。	0
4	ic_pmp1cache_strtg	MRW	PMP 通道 1 地址区域的指令缓存策略寄存器。	0
[3:1]	Reserved	MRO	保留。	0
0	ic_pmp0cache_strtg	MRW	PMP 通道 0 地址区域的指令缓存策略寄存器: 1: 当指令地址受 PMP 通道 0 保护时, 当前指令可缓存; 0: 当指令地址受 PMP 通道 0 保护时, 当前指令不可缓存。	0

注: 缓存策略 PMP 覆盖寄存器, 当指令或数据地址与受 PMP 通道控制的地址匹配时, 不执行缓存策略控制寄存器 (0xBC2) 中的策略, 执行本寄存器中与 PMP 通道匹配的策略。

4.2.3.11 硬件压栈地址寄存器 (HW_POPDM_ADDR)

CSR 地址: 0xBC4

位	名称	访问	描述	复位值
[31:20]	hw_stk_vector	MRO	DTCM 区域的固定首地址。	0x200
[19:9]		MRW	DTCM 区域块地址, 512bit 对齐。	0x600
[8:0]		MRO	固定为 0。	0

注: 硬件压栈地址寄存器, 指向 DTCM 区域中, 当硬件压栈功能使能且进入中断/异常时, 将堆栈值快速保存到指定 DTCM 区域中, 通常指向 DTCM 区域的底部。

4.2.3.12 存储器配置寄存器 (MEMORY_CFGR)

CSR 地址: 0xBC5

位	名称	访问	描述	复位值
[31:20]	MEM0_addr_hi[11:0]	MRW	分支预测器地址匹配寄存器, 配置为外部存储器的高 12 位地址信息, 当外部存储器地址高位不匹配时, 部分分支预测功能受限。 注: 请勿配置为内部存储器的高位地址 (0x000,	0x600

			0x200, 0x201)。	
[19:18]	dtcm_rr_mode[1:0]	MRW	DTCM 存储器访问优先级模式： 00：固定优先级，其他请求（DMA, C0_lsu, C0_ifu）高于 C1 请求； 01：固定优先级，C1 请求高于其他请求（DMA, C0_lsu, C0_ifu）； 10：轮询优先级，一通道最多连续获得一段时间的优先权，超时后切换到另一通道，当获得优先权的通道无操作请求而令一通道有请求时优先权切换到另一通道并重新计时，优先时长配置见 TCM_RRDUTY_CFGR 寄存器。 11：强制轮询优先级，一通道强制获得一段时间的优先权，超时后切换到另一通道，优先时长配置见 TCM_RRDUTY_CFGR 寄存器。	0x3
[17:16]	itcm_rr_mode[1:0]	MRW	ITCM 存储器访问优先级模式： 00：固定优先级，其他请求（DMA, C0_lsu, C0_ifu）高于 C1 请求 01：固定优先级，C1 请求高于其他请求（DMA, C0_lsu, C0_ifu） 10：轮询优先级，一通道最多连续获得一段时间的优先权，超时后切换到另一通道，当获得优先权的通道无操作请求而令一通道有请求时优先权切换到另一通道并重新计时，优先时长配置见 TCM_RRDUTY_CFGR 寄存器。 11：强制轮询优先级，一通道强制获得一段时间的优先权，超时后切换到另一通道，优先时长配置见 TCM_RRDUTY_CFGR 寄存器。	0x3
[15:12]	Reserved	RO	保留。	0
[11:10]	C0_ifu_dtcn_mode	MRW	C0_IFU 对 DTCM 存储器访问权限配置： C0_ifu_dtcn_mode[1]： 1：读使能； 0：读禁止。 C0_ifu_dtcn_mode[0]： 1：写使能； 0：写禁止。	0x3
[9:8]	C0_ifu_itcm_mode	MRW	C0_IFU 对 ITCM 存储器访问权限配置： C0_ifu_itcm_mode[1]： 1：读使能； 0：读禁止。 C0_ifu_itcm_mode[0]： 1：写使能； 0：写禁止。	0x3
[7:6]	C0_lsu_dtcn_mode	MRW	C0_LSU 对 DTCM 存储器访问权限配置： C0_lsu_dtcn_mode[1]： 1：读使能； 0：读禁止。	0x3

			C0_lsu_dtcn_mode[0]: 1: 写使能; 0: 写禁止。	
[5:4]	C0_lsu_itcm_mode	MRW	C0_LSU 对 ITCM 存储器访问权限配置: C0_lsu_itcm_mode[1]: 1: 读使能; 0: 读禁止。 C0_lsu_itcm_mode[0]: 1: 写使能; 0: 写禁止。	0x3
[3:2]	DMA_dtcn_mode	MRW	DMA 对 DTCM 存储器访问权限配置: DMA_dtcn_mode[1]: 1: 读使能; 0: 读禁止。 DMA_dtcn_mode[0]: 1: 写使能; 0: 写禁止。	0x3
[1:0]	DMA_itcm_mode	MRW	DMA 对 ITCM 存储器访问权限配置: DMA_itcm_mode[1]: 1: 读使能; 0: 读禁止。 DMA_itcm_mode[0]: 1: 写使能; 0: 写禁止。	0x3

4.2.3.13 TCM 优先访问时长配置寄存器 (TCM_RRDUTY_CFGR)

CSR 地址: 0xBC6

位	名称	访问	描述	复位值
[31:24]	C1_dtcn_rr_cycle [7:0]	MRW	来自 C1 对 dtcm 请求高优先级的持续时钟周期配置。	0
[23:16]	Cx_dtcn_rr_cycle [7:0]	MRW	来自非 C1 (C0, dma) 对 dtcm 请求高优先级的持续时钟周期配置。	0
[15:8]	C1_itcm_rr_cycle [7:0]	MRW	来自 C1 对 itcm 请求高优先级的持续时钟周期配置。	0
[7:0]	Cx_itcm_rr_cycle [7:0]	MRW	来自非 C1 (C0, dma) 对 itcm 请求高优先级的持续时钟周期配置。	0

注: TCM 区域访问优先级持续时间配置, 当 TCM_CFGR 中 itcm_rr_mode, dtcm_rr_mode 配置为 10b 或 11b 时生效, 用于控制对应通道高优先级的持续时间。

4.2.3.14 机器中断寄存器 (MIE)

CSR 地址: 0xBC8

位	名称	访问	描述	复位值
[31:9]	Reserved	RO	保留。	0
[8:0]	nest_mie	RW	机器中断寄存器, 保存多级中断嵌套中每级中断的 mie 信息:	0

			MSTATUS 寄存器 mie 位 = nest_mie[0]; MSTATUS 寄存器 mpie 位 = nest_mie[1]。	

4.2.3.15 缓存操作寄存器 (OPCACHE_CTLR)

CSR 地址: 0xBD0

位	名称	访问	描述	复位值
[31:5]	Vaddr[27:0]	WO	操作地址或索引信息。	0
[4:3]	Reserved	RO	保留。	0
2	Idx_mode	WO	索引模式 1: 以 vaddr 值为地址信息执行操作; 0: 以 vaddr 值中的索引信息为首地址执行操作。	0
[1:0]	Opcode[1:0]	WO	00: lcache invalidate; 其他: 无效。	0

4.2.3.16 内存信息寄存器 (MEMINFO)

CSR 地址: 0xFC0

位	名称	访问	描述	复位值
[31:30]	Reserved	RO	保留。	0
[29:26]	DTCM datasize[3:0]	MRO	数据内存容量信息: $64KB * 2^{DTCM\ datasize} = 256KB$	0100b
[25:24]	DTCM LineSize[1:0]	MRO	数据内存行长度信息: 00: 8 字节; 01: 16 字节; 10: 32 字节; 11: 64 字节。	10b
[23:22]	Reserved	RO	保留。	0
[21:18]	ITCM datasize[3:0]	MRO	指令内存容量信息: $64KB * 2^{ITCM\ datasize} = 128KB$ 。	0010b
[17:16]	ITCM LineSize[1:0]	MRO	指令内存行长度信息: 00: 8 字节; 01: 16 字节; 10: 32 字节; 11: 64 字节。	10b
[15:7]	Reserved	RO	保留。	0
[6:5]	lcache_way[1:0]	MRO	指令缓存路数: 00: 1-way; 01: 2-way; 10: 4-way; 11: 8-way。	01b
[4:2]	lcache_datasize[2:0]	MRO	指令缓存容量信息: 000: NONE; 001: 4KB; 010: 8KB; 011: 16KB;	100b

			100: 32KB; 101: 64KB; 110: 128KB; 111: 256KB。	
[1:0]	lcache linesize [1:0]	MRO	指令缓存行长度信息: 00: 8byte; 01: 16byte; 10: 32byte; 11: 64byte。	0

4.3 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）。

内核 1（RISC-V5F）的 PMP 单元包含 4 组 8 位的配置寄存器（CSR 地址为 0x3A0）和 4 组地址寄存器（CSR 地址为 0x3B0~0x3B3），需要使用 csr 指令进行访问，并且在机器模式下进行。

内核 0（RISC-V3F）的 PMP 单元包含 4 组 8 位的配置寄存器（CSR 地址为 0x3A0）和 4 组地址寄存器（CSR 地址为 0x3B0~0x3B3），需要使用 csr 指令进行访问，并且在机器模式下进行。

注：内核 0 的 PMP 功能仅适用于批号第五位不为 0 的芯片。

4.3.1 PMP 寄存器

表 4-7 PMP 寄存器列表

名称	CSR 地址	描述	复位值
PMPCFG0	0x3A0	PMP 配置寄存器	0x00000000
PMPADDR0	0x3B0	PMP 地址 0 寄存器	0x00000000
PMPADDR1	0x3B1	PMP 地址 1 寄存器	0x00000000
PMPADDR2	0x3B2	PMP 地址 2 寄存器	0x00000000
PMPADDR3	0x3B3	PMP 地址 3 寄存器	0x00000000

4.3.1.1 PMP 配置寄存器（PMPCFG0）

CSR 地址：0x3A0

位	名称	访问	描述	复位值
[31:24]	pmp3cfg[7:0]	MRW	见 pmp0cfg。	0
[23:16]	pmp2cfg[7:0]	MRW	见 pmp0cfg。	0
[15:8]	pmp1cfg[7:0]	MRW	见 pmp0cfg。	0

[7:0]	pmp0cfg[7:0]	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	-	保留。	
			5	IC_Str	地址匹配时执行指令缓存策略， 策略值与 CACHE_PMP_OVR 寄存器 对应。	
			[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 $A\_ADDR \leq region < B\_ADDR$ 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 $B\_ADDR - A\_ADDR == 2^{(G+2)}$ ， $G \geq 1$ ，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： $region = BUS\_ADDR \gg 2$; pmp0cfg 下， $0 \leq region < pmpaddr0$; pmp1cfg 下， $pmpaddr0 \leq region < pmpaddr1$; pmp2cfg 下， $pmpaddr1 \leq region < pmpaddr2$; pmp3cfg 下， $pmpaddr2 \leq region < pmpaddr3$ 。 $pmpaddr_{i-1} = A\_ADDR \gg 2$; $pmpaddr_i = B\_ADDR \gg 2$ 。
10b	NA4	固定 4 字节区域保护。 pmp0cfg~pmp3cfg 对应 pmpaddr0~pmpaddr3 作为起始地址。 $pmpaddr_i = A\_ADDR \gg 2$ 。
11b	NAPOT	保护 $2^{(G+2)}$ 区域， $G \geq 1$ ，此时 A_ADDR 为 $2^{(G+2)}$ 对齐。 $pmpaddr_i = ((A\_ADDR \gg 2) (2^{(G-1)} - 1))$ 。

4.3.1.2 PMP 地址 0 寄存器（PMPADDR0）

CSR 地址：0x3B0

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

4.3.1.3 PMP 地址 1 寄存器（PMPADDR1）

CSR 地址：0x3B1

位	名称	访问	描述	复位值

[31:0]	ADDR1[31:0]	MRW	PMP 设置地址 1 的 bit[33:2], 实际高 2 位未用。	0

4.3.1.4 PMP 地址 2 寄存器 (PMPADDR2)

CSR 地址: 0x3B2

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

4.3.1.5 PMP 地址 3 寄存器 (PMPADDR3)

CSR 地址: 0x3B3

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

4.4 核间通信 (IPC)

核间通信功能提供四路通信通道, 每通道 8 位状态位, 状态位独立使能, 两种中断方式, 32bit*4 通信数据寄存器。

- 两种通道中断方式: 接收端中断和发送端中断。
- 每通道 8 位状态位, 状态位独立使能, 使能自动更新。
- 接收中断和发送中断独立使能, 中断内核独立可配。
- 四组核间信息寄存器。

4.4.1 IPC 功能描述

本模块提供两种中断类型:

- 接收端中断
- 发送端中断

IPC 通道中断受中断使能位 (RX_IERx/TX_IERx), 状态使能位 (IPC_ENAx[7:0]) 和状态位 (IPC_STSx[7:0]) 共同控制。

接收中断

当状态位值为 1 时, 对应状态使能位值为 1, IPC_ISR 寄存器相应位值为 1; 若通道接收中断使能有效 (RX_IERx = 1), 可向通道配置的 RX_CID 内核产生中断请求, IPC_ISM 寄存器相应位值为 1。

发送中断

当状态位值为 0 时, 对应状态使能位值为 1, IPC_ISR 寄存器相应位值为 1; 若通道发送中断使能有效 (TX_IERx = 1), 可向通道配置的 TX_CID 内核产生中断请求, IPC_ISM 寄存器相应位值为 1。

当 AUTOENx 位配置为自动更新时, 发送端写/置位 IPC_STS 寄存器时, IPC_ENA 寄存器同步更新为相同值; 若通道接收中断使能有效, 可以立即向接收端产生接收中断。接收中断响应完成后清除 IPC_STS 标志不会影响 IPC_ENA 寄存器, 若通道发送中断有效, 可以立即向发送端产生发送中断, 发送端在中断中关闭 IPC_ENA 使能, 清除发送端中断标志。

当 AUTOENx 位配置为非自动更新时, 发送端先写状态使能位、后写 IPC_STS 的操作顺序可能引入额外的发送端中断。

4.4.2 IPC 寄存器

表 4-8 IPC 相关寄存器列表

名称	访问地址	描述	复位值
R32_IPC_CTLR	0xE000D000	IPC 控制寄存器	0x00000000
R32_IPC_ISR	0xE000D004	IPC 中断状态寄存器	0x00000000
R32_IPC_ISM	0xE000D008	IPC ISM 通道中断掩膜寄存器	0x00000000
R32_IPC_ENA	0xE000D010	IPC 状态位中断使能寄存器	0x00000000
R32_IPC_STS	0xE000D014	IPC 通道状态寄存器	0xFFFFFFFF
R32_IPC_SET	0xE000D018	IPC 状态标志位置寄存器	0x00000000
R32_IPC_CLR	0xE000D01C	IPC 状态标志位复位寄存器	0x00000000
R32_IPC_MSG0	0xE000D020	IPC 信息寄存器 0	0xFFFFFFFF
R32_IPC_MSG1	0xE000D024	IPC 信息寄存器 1	0xFFFFFFFF
R32_IPC_MSG2	0xE000D028	IPC 信息寄存器 2	0xFFFFFFFF
R32_IPC_MSG3	0xE000D02C	IPC 信息寄存器 3	0xFFFFFFFF

4.4.2.1 IPC 控制寄存器 (IPC_CTLR)

偏移地址: 0x00

位	名称	访问	描述	复位值
31	LOCK3	RW	通道 3 配置锁定, 写 1 后仅复位可清零, 锁定后通道 3 的其他控制寄存器不可改写。 1: 通道 3 控制锁定; 0: 通道 3 控制非锁定。	0
30	AUTOEN3	RW	通道 3 状态位使能自动更新, 置 1 时当发送端写 IPC_STS 时, IPC_ENA 自动更新为相同值, 发送端写 IPC_SET 置位 IPC_STS 时 IPC_ENA 相应位置位。 1: IPC_ENA 寄存器自动更新; 0: IPC_ENA 寄存器自动更新屏蔽。	0
29	RX_IER3	RW	通道 3 接收端中断使能, 置 1 时通道 3 可以产生接收中断: 1: 通道 3 接收端中断使能; 0: 通道 3 接收端中断屏蔽。	0
28	TX_IER3	RW	通道 3 发送端中断使能, 置 1 时通道 3 可以产生发送中断: 1: 通道 3 发送端中断使能; 0: 通道 3 发送端中断屏蔽。	0
27	Reserved	RO	保留。	0
26	RX_CID3	RW	接收端内核 ID, 配置响应接收中断的内核: 1: 向内核 1 产生接收中断请求; 0: 向内核 0 产生接收中断请求。	0
25	Reserved	RO	保留。	0
24	TX_CID3	RW	发送端内核 ID, 配置响应发送中断的内核: 1: 向内核 1 产生发送中断请求; 0: 向内核 0 产生发送中断请求。	0
23	LOCK2	RW	通道 2 配置锁定。	0

22	AUTOEN2	RW	通道 2 状态位使能自动更新。	0
21	RX_IER2	RW	通道 2 接收端中断使能，置 1 时通道 2 可以产生接收中断。	0
20	TX_IER2	RW	通道 2 发送端中断使能，置 1 时通道 2 可以产生发送中断。	0
19	Reserved	RO	保留。	0
18	RX_CID2	RW	接收端内核 ID，配置响应接收中断的内核。	0
17	Reserved	RO	保留。	0
16	TX_CID2	RW	发送端内核 ID，配置响应发送中断的内核。	0
15	LOCK1	RW	通道 1 配置锁定。	0
14	AUTOEN1	RW	通道 1 状态位使能自动更新。	0
13	RX_IER1	RW	通道 1 接收端中断使能，置 1 时通道 1 可以产生接收中断。	0
12	TX_IER1	RW	通道 1 发送端中断使能，置 1 时通道 1 可以产生发送中断。	0
11	Reserved	RO	保留。	0
10	RX_CID1	RW	接收端内核 ID，配置响应接收中断的内核。	0
9	Reserved	RO	保留。	0
8	TX_CID1	RW	发送端内核 ID，配置响应发送中断的内核。	0
7	LOCK0	RW	通道 0 配置锁定。	0
6	AUTOEN0	RW	通道 0 状态位使能自动更新。	0
5	RX_IER0	RW	通道 0 接收端中断使能，置 1 时通道 0 可以产生接收中断。	0
4	TX_IER0	RW	通道 0 发送端中断使能，置 1 时通道 0 可以产生发送中断。	0
3	Reserved	RO	保留。	0
2	RX_CID0	RW	接收端内核 ID，配置响应接收中断的内核	0
1	Reserved	RO	保留。	0
0	TX_CID0	RW	发送端内核 ID，配置响应发送中断的内核。	0

4.4.2.2 IPC 中断状态寄存器（IPC_ISR）

偏移地址：0x04

位	名称	访问	描述	复位值
[31:24]	IPC_ISR3[7:0]	RO	IPC 通道 3 中断状态寄存器，值为非 0 时 IPC 通道 3 存在中断。 当 IPC_ENA 使能时，IPC_STS 寄存器值为 0 时产生发送端中断；IPC_STS 寄存器值为 1 时产生接收端中断。	0
[23:16]	IPC_ISR2[7:0]	RO	IPC 通道 2 中断状态寄存器。	0
[15:8]	IPC_ISR1[7:0]	RO	IPC 通道 1 中断状态寄存器。	0
[7:0]	IPC_ISR0[7:0]	RO	IPC 通道 0 中断状态寄存器。	0

4.4.2.3 IPC ISM 通道中断掩膜寄存器（IPC_ISM）

偏移地址：0x08

位	名称	访问	描述	复位值

[31:24]	IPC_ISM3[7:0]	RO	IPC 通道 3 中断掩膜寄存器，值为非 0 时 IPC 通道 3 产生中断请求。 当 RX_IER 使能时，IPC_STS 值为 1，IPC_ENA 值为 1 的位向 RX_CID 产生接收端中断。 当 TX_IER 使能时，IPC_STS 值为 0，IPC_ENA 值为 1 的位向 TX_CID 产生发送端中断。	0
[23:16]	IPC_ISM2[7:0]	RO	IPC 通道 2 中断掩膜寄存器。	0
[15:8]	IPC_ISM1[7:0]	RO	IPC 通道 1 中断掩膜寄存器。	0
[7:0]	IPC_ISM0[7:0]	RO	IPC 通道 0 中断掩膜寄存器。	0

4.4.2.4 IPC 状态位中断使能寄存器 (IPC_ENA)

偏移地址：0x10

位	名称	访问	描述	复位值
[31:24]	IPC_ENA3[7:0]	RW	IPC 通道 3 标志位中断使能寄存器，值为 1 时使能相应位的中断功能。 IPC_STS 值为 0，IPC_ENA 值为 1 的位 TX_CID 产生发送端中断。 IPC_STS 值为 1，IPC_ENA 值为 1 的位 RX_CID 产生接收端中断。	0
[23:16]	IPC_ENA2[7:0]	RW	IPC 通道 2 标志位中断使能寄存器。	0
[15:8]	IPC_ENA1[7:0]	RW	IPC 通道 1 标志位中断使能寄存器。	0
[7:0]	IPC_ENA0[7:0]	RW	IPC 通道 0 标志位中断使能寄存器。	0

4.4.2.5 IPC 通道状态寄存器 (IPC_STS)

偏移地址：0x14

位	名称	访问	描述	复位值
[31:24]	IPC_STS3[7:0]	RW	IPC 通道 3 状态寄存器： IPC_STS 值为 1 的位可产生接收中断； IPC_STS 值为 0 的位可产生发送中断。	X
[23:16]	IPC_STS2[7:0]	RW	IPC 通道 2 状态寄存器。	X
[15:8]	IPC_STS1[7:0]	RW	IPC 通道 1 状态寄存器。	X
[7:0]	IPC_STS0[7:0]	RW	IPC 通道 0 状态寄存器。	X

4.4.2.6 IPC 状态标志位置位寄存器 (IPC_SET)

偏移地址：0x18

位	名称	访问	描述	复位值
[31:24]	IPC_SET3[7:0]	WO	IPC 通道 3 状态标志位置位寄存器，写 1 后置位 IPC_STS 寄存器对应位，如果通道的 AUTOEN 位使能，则同时置位 IPC_ENA 寄存器对应位。	0
[23:16]	IPC_SET2[7:0]	WO	IPC 通道 2 状态标志位置位寄存器。	0
[15:8]	IPC_SET1[7:0]	WO	IPC 通道 1 状态标志位置位寄存器。	0
[7:0]	IPC_SET0[7:0]	WO	IPC 通道 0 状态标志位置位寄存器。	0

4.4.2.7 IPC 状态标志位复位寄存器 (IPC_CLR)

偏移地址：0x1C

位	名称	访问	描述	复位值
[31:24]	IPC_CLR3[7:0]	WO	IPC 通道 3 状态标志位清除寄存器，写 1 后复位 IPC_STS 寄存器对应位。	0
[23:16]	IPC_CLR2[7:0]	WO	IPC 通道 2 状态标志位清除寄存器，写 1 后复位 IPC_STS 寄存器对应位。	0
[15:8]	IPC_CLR1[7:0]	WO	IPC 通道 1 状态标志位清除寄存器，写 1 后复位 IPC_STS 寄存器对应位。	0
[7:0]	IPC_CLR0[7:0]	WO	IPC 通道 0 状态标志位清除寄存器，写 1 后复位 IPC_STS 寄存器对应位。	0

4.4.2.8 IPC 信息寄存器（IPC_MSGx）（x=0/1/2/3）

偏移地址：0x20+0x4*n

位	名称	访问	描述	复位值
[31:0]	IPC_MSGx[31:0]	RW	IPC 信息寄存器，用于存储和传递核间通信需要传递的信息。	X

4.5 硬件信号量（HSEM）

硬件信号量功能提供 32 路硬件信号量通道。信号量功能可以用于在多核中不同进程运行间的同步操作，并提供非阻塞单指令锁定模式。功能包含如下：

- 信号量通道释放时产生中断。
- 信号量单通道解锁：解锁通道需操作内核与 CID 匹配，写数据与 PID 值匹配。
- 信号量通道批量解锁：通过关键值、CID 和 PID 匹配方式批量解锁。
- 两种信号量通道上锁方式：
 - 2-step 锁定：通过向 HSEM 通道写入 PID 锁定，CID 自动更新为获得锁的内核 ID 值，随后读数据进行检查。
 - 1-step 锁定：通过读 HSEM 通道获得锁，CID 自动更新为获得锁的内核 ID 值。

4.5.1 HSEM 功能描述

4.5.1.1 上锁方式

本模块提供两种上锁方式：

- 2-step (write) lock
- 1-step (read) lock

HSEM 通道 LOCK 寄存器值为 0 表示通道空闲，此时该通道的 CID 和 PID 值也为 0。当通道 LOCK 寄存器值为 1 时表示通道锁定，CID 值更新为锁定通道的内核 ID，PID 寄存器值为总线写入值。

当写锁定通道时，CID 更新为锁定通道的内核 ID，PID 值更新为写数据。当读锁定通道时，CID 更新为锁定通道的内核 ID，PID 值保持为 0。在写锁定时可以写入任意 PID 值。

2-step (write) 锁流程

2-step 锁流程通过向 HSEM_RX 寄存器写入锁信息锁定通道，随后通过读返回数据与写数据相同检查锁定成功。

- 向 HSEM_RX 寄存器中写入 PID，LOCK = 1，通道空闲时锁定成功，PID、CID、LOCK 值更新。
- 读数据检查，读数据 PID 与写数据匹配，CID 与内核 ID 匹配，LOCK 值为 1 表示操作成功。

- 否则重复上述操作。
HSEM 通道只有在空闲时可被锁定。
对已锁定的通道再次进行锁定操作时操作无效。

1-step (read) 锁流程

1-step 锁流程通过读 HSEM_RLRX 寄存器并比较返回值检查锁定成功。

- 读 HSEM_RLRX 寄存器获得返回值。
- 如果读数据为 0，则说明锁定成功，为非零值则锁定失败，该通道已经被其他进程锁定，锁定成功的通道 HSEM_RLRX 寄存器值更新为 LOCK = 1，CID = 操作的内核 ID，PID = 0。
- 否则重复上述操作。
- 通道解锁流程：
- 写解锁：

向锁定的 HSEM 通道对应的 HSEM_RX 寄存器中写 PID，LOCK=0，如果该通道由本内核锁定，写 PID 与通道 PID 相等，则解锁本通道，PID、CID、LOCK 清零。当通道中断使能时，可以产生中断请求。

- 匹配解锁：

向 HSEM_CLR 寄存器中写入关键值，配置匹配规则，与规则匹配的通道可以被批量解锁。当通道中断使能时，可以产生中断请求。

4.5.1.2 HSEM 中断

HSEM 通道被解锁时可以产生中断，HSEM 的中断具有以下特征：

- 各 HSEM 通道均可独立使能。
- 各 HSEM 通道具有独立中断标志。
- 各 HSEM 通道中断标志可单独清除。
- 各内核具有独立的中断使能，中断标志和清除位。

当通道中断被使能（HSEM_IER）时，该通道被其他内核解锁时可产生中断请求，对应的通道中断标志寄存器（HSEM_ISR）置 1。

向通道中断标志（HSEM_ISR）相应位写 1 可清除中断标志。

通道中断掩膜寄存器（HSEM_ISM）可用于获取产生中断的源通道，通道中断使能且中断标志置 1 的通道才可以产生中断请求，对应的 HSEM_ISM 位值为 1。

4.5.2 HSEM 寄存器

表 4-9 HSEM 相关寄存器列表

名称	访问地址	描述	复位值
R32_HSEM_RX0	0xE000C000	HSEM 锁定通道寄存器 0	0x00000000
R32_HSEM_RX1	0xE000C004	HSEM 锁定通道寄存器 1	0x00000000
R32_HSEM_RX2	0xE000C008	HSEM 锁定通道寄存器 2	0x00000000
R32_HSEM_RX3	0xE000C00C	HSEM 锁定通道寄存器 3	0x00000000
R32_HSEM_RX4	0xE000C010	HSEM 锁定通道寄存器 4	0x00000000
R32_HSEM_RX5	0xE000C014	HSEM 锁定通道寄存器 5	0x00000000
R32_HSEM_RX6	0xE000C018	HSEM 锁定通道寄存器 6	0x00000000
R32_HSEM_RX7	0xE000C01C	HSEM 锁定通道寄存器 7	0x00000000
R32_HSEM_RX8	0xE000C020	HSEM 锁定通道寄存器 8	0x00000000

R32_HSEM_RX9	0xE000C024	HSEM 锁定通道寄存器 9	0x00000000
R32_HSEM_RX10	0xE000C028	HSEM 锁定通道寄存器 10	0x00000000
R32_HSEM_RX11	0xE000C02C	HSEM 锁定通道寄存器 11	0x00000000
R32_HSEM_RX12	0xE000C030	HSEM 锁定通道寄存器 12	0x00000000
R32_HSEM_RX13	0xE000C034	HSEM 锁定通道寄存器 13	0x00000000
R32_HSEM_RX14	0xE000C038	HSEM 锁定通道寄存器 14	0x00000000
R32_HSEM_RX15	0xE000C03C	HSEM 锁定通道寄存器 15	0x00000000
R32_HSEM_RX16	0xE000C040	HSEM 锁定通道寄存器 16	0x00000000
R32_HSEM_RX17	0xE000C044	HSEM 锁定通道寄存器 17	0x00000000
R32_HSEM_RX18	0xE000C048	HSEM 锁定通道寄存器 18	0x00000000
R32_HSEM_RX19	0xE000C04C	HSEM 锁定通道寄存器 19	0x00000000
R32_HSEM_RX20	0xE000C050	HSEM 锁定通道寄存器 20	0x00000000
R32_HSEM_RX21	0xE000C054	HSEM 锁定通道寄存器 21	0x00000000
R32_HSEM_RX22	0xE000C058	HSEM 锁定通道寄存器 22	0x00000000
R32_HSEM_RX23	0xE000C05C	HSEM 锁定通道寄存器 23	0x00000000
R32_HSEM_RX24	0xE000C060	HSEM 锁定通道寄存器 24	0x00000000
R32_HSEM_RX25	0xE000C064	HSEM 锁定通道寄存器 25	0x00000000
R32_HSEM_RX26	0xE000C068	HSEM 锁定通道寄存器 26	0x00000000
R32_HSEM_RX27	0xE000C06C	HSEM 锁定通道寄存器 27	0x00000000
R32_HSEM_RX28	0xE000C070	HSEM 锁定通道寄存器 28	0x00000000
R32_HSEM_RX29	0xE000C074	HSEM 锁定通道寄存器 29	0x00000000
R32_HSEM_RX30	0xE000C078	HSEM 锁定通道寄存器 30	0x00000000
R32_HSEM_RX31	0xE000C07C	HSEM 锁定通道寄存器 31	0x00000000
R32_HSEM_RLRX0	0xE000C100	HSEM 读锁定寄存器 0	0x00000000
R32_HSEM_RLRX1	0xE000C104	HSEM 读锁定寄存器 1	0x00000000
R32_HSEM_RLRX2	0xE000C108	HSEM 读锁定寄存器 2	0x00000000
R32_HSEM_RLRX3	0xE000C10C	HSEM 读锁定寄存器 3	0x00000000
R32_HSEM_RLRX4	0xE000C110	HSEM 读锁定寄存器 4	0x00000000
R32_HSEM_RLRX5	0xE000C114	HSEM 读锁定寄存器 5	0x00000000
R32_HSEM_RLRX6	0xE000C118	HSEM 读锁定寄存器 6	0x00000000
R32_HSEM_RLRX7	0xE000C11C	HSEM 读锁定寄存器 7	0x00000000
R32_HSEM_RLRX8	0xE000C120	HSEM 读锁定寄存器 8	0x00000000
R32_HSEM_RLRX9	0xE000C124	HSEM 读锁定寄存器 9	0x00000000
R32_HSEM_RLRX10	0xE000C128	HSEM 读锁定寄存器 10	0x00000000
R32_HSEM_RLRX11	0xE000C12C	HSEM 读锁定寄存器 11	0x00000000
R32_HSEM_RLRX12	0xE000C130	HSEM 读锁定寄存器 12	0x00000000
R32_HSEM_RLRX13	0xE000C134	HSEM 读锁定寄存器 13	0x00000000
R32_HSEM_RLRX14	0xE000C138	HSEM 读锁定寄存器 14	0x00000000
R32_HSEM_RLRX15	0xE000C13C	HSEM 读锁定寄存器 15	0x00000000
R32_HSEM_RLRX16	0xE000C140	HSEM 读锁定寄存器 16	0x00000000
R32_HSEM_RLRX17	0xE000C144	HSEM 读锁定寄存器 17	0x00000000
R32_HSEM_RLRX18	0xE000C148	HSEM 读锁定寄存器 18	0x00000000
R32_HSEM_RLRX19	0xE000C14C	HSEM 读锁定寄存器 19	0x00000000
R32_HSEM_RLRX20	0xE000C150	HSEM 读锁定寄存器 20	0x00000000
R32_HSEM_RLRX21	0xE000C154	HSEM 读锁定寄存器 21	0x00000000

R32_HSEM_RLRX22	0xE000C158	HSEM 读锁定寄存器 22	0x00000000
R32_HSEM_RLRX23	0xE000C15C	HSEM 读锁定寄存器 23	0x00000000
R32_HSEM_RLRX24	0xE000C160	HSEM 读锁定寄存器 24	0x00000000
R32_HSEM_RLRX25	0xE000C164	HSEM 读锁定寄存器 25	0x00000000
R32_HSEM_RLRX26	0xE000C168	HSEM 读锁定寄存器 26	0x00000000
R32_HSEM_RLRX27	0xE000C16C	HSEM 读锁定寄存器 27	0x00000000
R32_HSEM_RLRX28	0xE000C170	HSEM 读锁定寄存器 28	0x00000000
R32_HSEM_RLRX29	0xE000C174	HSEM 读锁定寄存器 29	0x00000000
R32_HSEM_RLRX30	0xE000C178	HSEM 读锁定寄存器 30	0x00000000
R32_HSEM_RLRX31	0xE000C17C	HSEM 读锁定寄存器 31	0x00000000
R32_HSEM_LSE	0xE000C200	HSEM 锁状态寄存器	0x00000000
R32_HSEM_CLR	0xE000C208	HSEM 解锁寄存器	0x00000000
R32_HSEM_KEY	0xE000C20C	HSEM 解锁关键字寄存器	0x5AA50003
R32_HSEM_IER	0xE000C300	HSEM 中断使能寄存器	0x00000000
R32_HSEM_ISR	0xE000C308	HSEM 中断状态寄存器	0x00000000
R32_HSEM_ISM	0xE000C310	HSEM 中断掩膜寄存器	0x00000000
R32_HSEM_LSM	0xE000C318	HSEM 锁状态掩膜寄存器	0x00000000

注：寄存器中标注有“内核私有寄存器”的为外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.5.2.1 HSEM 锁定通道寄存器 y (HSEM_RXy) (y=0-31)

偏移地址：0x000 + 4y

位	名称	访问	描述	复位值
31	LOCK	RW	通道锁状态标志： 1：通道锁定； 0：通道空闲。	0
[30:10]	Reserved	RO	保留。	0
[9:8]	CID	RW	锁定通道的内核 ID： 1：通道被内核 1 锁定； 0：通道空闲或被内核 0 锁定。	0
[7:0]	PID	RW	锁定通道的进程 ID： 复位值或通道释放后装载为 0；写锁定锁定时更新为写入的总线数据。	0

4.5.2.2 HSEM 读锁定寄存器 y (HSEM_RLRXy) (y=0-31)

偏移地址：0x100 + 4y

位	名称	访问	描述	复位值
31	LOCK	RO	通道锁定状态标志： 1：通道锁定； 0：通道空闲。	0
[30:10]	Reserved	RO	保留。	0
[9:8]	CID	RO	锁定通道的内核 ID： 1：通道被内核 1 锁定； 0：通道空闲或被内核 0 锁定。	0
[7:0]	PID	RO	锁定通道的进程 ID，读锁定时为 0。	0

4.5.2.3 HSEM 锁状态寄存器 (HSEM_LSE)

偏移地址: 0x200

位	名称	访问	描述	复位值
[31:0]	HSEM_LSE	RO	HSEM 通道锁状态寄存器, 用于获取 32 通道的锁定状态: 1: 通道锁定; 0: 通道空闲。	0

4.5.2.4 HSEM 解锁寄存器 (HSEM_CLR)

偏移地址: 0x208

位	名称	访问	描述	复位值
[31:16]	CLR_KEY	WO	解锁关键字, 当写数据与匹配关键字相等时操作有效, 否则无效。键值为 0x5AA5。	0
[15:14]	Reserved	RO	保留。	0
13	CID_MASK	WO	CID 匹配屏蔽: 1: 屏蔽 CID 匹配, 所有 CID 值均解锁; 0: 不屏蔽 CID 匹配, 锁定的通道 CID 值与 match_CID 值相同时解锁。	0
12	PID_MASK	WO	PID 匹配屏蔽: 1: 屏蔽 PID 匹配, 所有 PID 值均解锁; 0: 不屏蔽 PID 匹配, 锁定的通道 PID 值与 match_PID 值相同时解锁。	0
[11:10]	Reserved	RO	保留。	0
[9:8]	match_CID	WO	用于匹配解锁的 CID 值, 与各锁定通道的 CID 值比较。	0
[7:0]	match_PID	WO	用于匹配解锁的 PID 值, 与各锁定通道的 PID 值比较。	0

4.5.2.5 HSEM 解锁关键字寄存器 (HSEM_KEY)

偏移地址: 0x20C

位	名称	访问	描述	复位值
[31:16]	KEY_VALUE	RO	解锁关键字, 用于匹配解锁方式关键字比较。	0x5AA5
[15:2]	Reserved	RO	保留。	0
1	auto_2step_clr	RW	使用 2step 方式锁定通道失败时自动清除对应通道中断状态位使能: 1: 锁定失败时自动清除对应通道中断状态位; 0: 锁定失败不影响对应通道中断状态位。	1
0	auto_1step_clr	RW	使用 1step 方式锁定通道失败时自动清除对应通道中断状态位使能: 1: 锁定失败时自动清除对应通道中断状态位; 0: 锁定失败不影响对应通道中断状态位。	1

4.5.2.6 HSEM 中断使能寄存器 (HSEM_IER)

偏移地址: 0x300

位	名称	访问	描述	复位值
[31:0]	HSEM_IER	RW	HSEM 通道中断使能寄存器，内核私有寄存器，置 1 后对应通道被其他内核解锁时允许产生中断。与 HSEM_ISR 寄存器共同作用可产生中断。	0

4.5.2.7 HSEM 中断状态寄存器 (HSEM_ISR)

偏移地址：0x308

位	名称	访问	描述	复位值
[31:0]	HSEM_ISR	RW1Z	HSEM 通道中断状态寄存器，内核私有寄存器，当锁定的通道被其他内核解锁时对应位被置位。与 HSEM_IER 寄存器共同作用可产生中断。	0

4.5.2.8 HSEM 中断掩膜寄存器 (HSEM_ISM)

偏移地址：0x310

位	名称	访问	描述	复位值
[31:0]	HSEM_ISM	RO	HSEM 通道中断掩膜寄存器，内核私有寄存器，值等于 HSEM_ISR & HSEM_IER，值为非 0 时表示产生了 SEM 中断，产生中断的通道对应位值为 1。	0

4.5.2.9 HSEM 锁状态掩膜寄存器 (HSEM_LSM)

偏移地址：0x318

位	名称	访问	描述	复位值
[31:0]	HSEM_LSM	RO	HSEM 通道锁状态寄存器，用于获取 HSEM 通道是否被本内核锁定： 1：通道被本内核锁定； 0：通道空闲或被其他内核锁定。	0

4.6 系统计数器 (SysTick)

内核自带了 2 个 32 位加减计数器 (SysTick)，可独立产生中断信号，可配置向指定内核产生中断请求。系统计数器计数频率与 HCLK 频率 (V3F 内核频率) 相同，RTC 计数频率为 HCLK/8。

- 双路 32 位定时器
- 向上向下计数
- 主频时钟和 RTC 时钟计数可配
- 中断请求内核可配
- 自动重装载可配

4.6.1 RISC-V-SysTick 寄存器

表 4-10 STK 相关寄存器列表

名称	访问地址	描述	复位值
R32_STK_CTLR_0	0xE000F000	系统计数控制寄存器 0	0x00000000
R32_STK_ISR	0xE000F004	系统计数器中断状态寄存器	0x00000000

R32_STK_CNT_0	0xE000F008	系统计数器 0 计数寄存器	0x00000000
R32_STK_CMP_0	0xE000F010	系统计数器 0 比较寄存器	0x00000000
R32_STK_CTLR_1	0xE000F080	系统计数控制寄存器 1	0x00000000
R32_STK_CNT_1	0xE000F088	系统计数器 1 计数寄存器	0x00000000
R32_STK_CMP_1	0xE000F090	系统计数器 1 比较寄存器	0x00000000

4.6.1.1 系统计数控制寄存器 0 (STK_CTLR_0)

偏移地址：0x00

位	名称	访问	描述	复位值
[31:7]	Reserved	RO	保留。	0
6	CID_0	RW	计数器 0 分配内核寄存器，用于配置计数器 0 向指定内核产生中断请求，在计数器 0 使能被配置有效时自动更新为操作的内核 ID。	0
5	Reserved	RO	保留。	0
4	DOWN_MODE_0	RW	计数模式： 1：向下计数； 0：向上计数。	0
3	AUTO_RELOAD_0	RW	自动重载计数使能位： 1：自动重载； 0：不自动重载。	0
2	NO_RTC_0	RW	计数器时钟源选择位： 1：使用 HCLK 做时基； 0：使用 HCLK/8 做时基。	0
1	IE_0	RW	计数器 0 中断使能控制位： 1：使能计数器中断； 0：关闭计数器中断。	0
0	EN_0	RW	系统计数器 0 使能控制位： 1：启动系统计数器 STK，且置 1 后仅 CID 与操作内核匹配时可以修改 TIM_CTL_0 的配置； 0：关闭系统计数器 STK，计数器停止计数。	0

4.6.1.2 系统计数器中断状态寄存器 (STK_ISR)

偏移地址：0x04

位	名称	访问	描述	复位值
[31:2]	Reserved	RO	保留。	0
1	STK_ISR1	RW	Systick1 中断标志： 1：定时器 1 存在未处理的中断请求； 0：定时器 1 不存在中断请求。	0
0	STK_ISR0	RW	Systick0 中断标志： 1：定时器 0 存在未处理的中断请求； 0：定时器 0 不存在中断请求。	0

4.6.1.3 系统计数器 0 计数寄存器 (STK_CNT_0)

偏移地址：0x08

位	名称	访问	描述	复位值

[31:0]	CNT_0[31:0]	RW	Systick0 计数寄存器, 计数使能后根据配置累加/减和自动重装载。	0

4.6.1.4 系统计数器 0 比较寄存器 (STK_CMP_0)

偏移地址: 0x10

位	名称	访问	描述	复位值
[31:0]	CMP_0[31:0]	RW	Systick0 比较寄存器, 用于向上计数时比较和向下计数时自动重装载。	0

4.6.1.5 系统计数控制寄存器 1 (STK_CTLR_1)

偏移地址: 0x80

位	名称	访问	描述	复位值
[31:7]	Reserved	RO	保留。	0
6	CID_1	RW	计数器 1 分配内核寄存器, 用于配置计数器 1 向指定内核产生中断请求, 在计数器 1 使能被配置有效时自动更新为操作的内核 ID。	0
5	Reserved	RO	保留。	0
4	DOWN_MODE_1	RW	计数模式: 1: 向下计数; 0: 向上计数。	0
3	AUTO_RELOAD_1	RW	自动重装载计数使能位: 1: 自动重装载; 0: 不自动重装载。	0
2	NO_RTC_1	RW	计数器时钟源选择位: 1: 使用 HCLK 做时基; 0: 使用 HCLK/8 做时基。	0
1	IE_1	RW	计数器 1 中断使能控制位: 1: 使能计数器中断; 0: 关闭计数器中断。	0
0	EN_1	RW	系统计数器 1 使能控制位: 1: 启动系统计数器 STK, 且置 1 后仅 CID 与操作内核匹配时可以修改 TIM_CTL_0 的配置; 0: 关闭系统计数器 STK, 计数器停止计数。	0

4.6.1.6 系统计数器 1 计数寄存器 (STK_CNT_1)

偏移地址: 0x88

位	名称	访问	描述	复位值
[31:0]	CNT_1[31:0]	RW	Systick1 计数寄存器, 计数使能后根据配置累加/减和自动重装载。	0

4.6.1.7 系统计数器 1 比较寄存器 (STK_CMP_1)

偏移地址: 0x90

位	名称	访问	描述	复位值
[31:0]	CMP_1[31:0]	RW	Systick1 比较寄存器, 用于向上计数时比较和向下计数时自动重装载。	0

4.7 中断和事件（PFIC）

内置可编程快速中断控制器（PFIC - Programmable Fast Interrupt Controller），最多支持 256 个中断向量，其中 RISC-V3F、RISC-V5F 内核各自管理 32 个内核私有中断，并共享管理其他中断。可通过配置中断分配寄存器分配给其中一个内核响应，低 32 号中断中部分外设如 IPC、SysTick 等需通过配置外设中的寄存器分配给指定内核，一次中断只能向一个内核产生中断请求。

中断控制平台可以控制内核睡眠状态和唤醒，当内核执行 WFI 指令后，内核进入低功耗状态，中断控制平台将置位内核睡眠标志、并监测中断或事件，当有中断或事件产生时，中断控制平台唤醒内核执行中断和其他程序。

4.7.1 PFIC 控制器主要特征

- 最多支持 256 路中断仲裁，低 32 位私有中断
- 共享中断分配
- 支持硬件中断堆栈（HPE），无需指令开销
- 提供 4 路免表中断（VTF）
- 支持低功耗睡眠和唤醒管理
- 内核 RISC-V3F 最高支持可配置的 2 级中断嵌套，内核 RISC-V5F 最高支持可配置的 8 级中断嵌套

4.7.2 中断和异常的向量表

表 4-11 CH32H417 系列芯片向量表

编号	优先级	类型	名称	描述	偏移地址
0	-6	固定	RESET	复位中断	0x000
1	-	-	-	保留	0x004
2	-5	固定	NMI	不可屏蔽中断	0x008
3	-4	固定	HardFault	异常中断	0x00C
4	-	-	-	保留	0x010
5	-3	固定	Ecall-M	机器模式回调中断	0x014
6-7	-	-	-	保留	0x018-0x01C
8	-2	固定	Ecall-U	用户模式回调中断	0x020
9	-1	固定	BreakPoint	断点回调中断	0x024
10-11	-	-	-	保留	0x028-0x02C
12	0	可编程	SysTick0	系统定时器中断 0	0x030
13	1	可编程	SysTick1	系统定时器中断 1	0x034
14	2	可编程	SW	软件中断	0x038
15	-	-	-	保留	0x03C
16	3	可编程	IPC_CH0	核间通信中断 0	0x040
17	4	可编程	IPC_CH1	核间通信中断 1	0x044
18	5	可编程	IPC_CH2	核间通信中断 2	0x048
19	6	可编程	IPC_CH3	核间通信中断 3	0x04C
20-27	-	-	-	保留	0x050-0x06C
28	7	可编程	HSEM	硬件信号量中断	0x070
29-31	-	-	-	保留	0x074-0x07C
32	8	可编程	WWDG	窗口定时器中断	0x080
33	9	可编程	EXTI15_8	EXTI 线[15:8]中断	0x084

34	10	可编程	FLASH	闪存全局中断	0x088
35	11	可编程	RCC	复位和时钟控制中断	0x08C
36	12	可编程	EXTI7_0	EXTI 线[7:0]中断	0x090
37	13	可编程	SPI1	SPI1 全局中断	0x094
38	14	可编程	DMA1_CH2	DMA1 通道 2 全局中断	0x098
39	15	可编程	DMA1_CH3	DMA1 通道 3 全局中断	0x09C
40	16	可编程	DMA1_CH4	DMA1 通道 4 全局中断	0x0A0
41	17	可编程	DMA1_CH5	DMA1 通道 5 全局中断	0x0A4
42	18	可编程	DMA1_CH6	DMA1 通道 6 全局中断	0x0A8
43	19	可编程	DMA1_CH7	DMA1 通道 7 全局中断	0x0AC
44	20	可编程	DMA1_CH8	DMA1 通道 8 全局中断	0x0B0
45	21	可编程	USART2	USART2 全局中断	0x0B4
46	22	可编程	I2C1_EV	I ² C1 事件中断	0x0B8
47	23	可编程	I2C1_ER	I ² C1 错误中断	0x0BC
48	24	可编程	USART1	USART1 全局中断	0x0C0
49	25	可编程	SPI2	SPI2 全局中断	0x0C4
50	26	可编程	SPI3	SPI3 全局中断	0x0C8
51	27	可编程	SPI4	SPI4 全局中断	0x0CC
52	28	可编程	I2C2_EV	I ² C2 事件中断	0x0D0
53	29	可编程	I2C2_ER	I ² C2 错误中断	0x0D4
54	30	可编程	USBPD	USBPD 全局中断	0x0D8
55	31	可编程	USBPDWakeUP	USBPD 唤醒中断	0x0DC
56	32	可编程	USBHS	USBHS 全局中断	0x0E0
57	33	可编程	DMA1_CH1	DMA1 通道 1 全局中断	0x0E4
58	34	可编程	CAN1_SCE	CAN1_SCE 全局中断	0x0E8
59	35	可编程	CAN1_TX	CAN1_TX 全局中断	0x0EC
60	36	可编程	CAN1_RX0	CAN1_RX0 全局中断	0x0F0
61	37	可编程	CAN1_RX1	CAN1_RX1 全局中断	0x0F4
62	38	可编程	USBSS	USBSS 全局中断	0x0F8
63	39	可编程	USBSS_LINK	USBSS LINK 中断	0x0FC
64	40	可编程	USBHSWakeUP	USBHS 唤醒中断	0x100
65	41	可编程	USBSSWakeUP	USBSS 唤醒中断	0x104
66	42	可编程	RTCAlarm	RTC 闹钟中断 (EXTI)	0x108
67	43	可编程	USBFS	USBFS 全局中断	0x10C
68	44	可编程	USBFSWakeUP	USBFS 唤醒中断	0x110
69	45	可编程	ADC1_2	ADC1_2 全局中断	0x114
70	46	可编程	TIM1_BRK	TIM1 刹车中断	0x118
71	47	可编程	TIM1_UP	TIM1 更新中断	0x11C
72	48	可编程	TIM1_TRG_COM	TIM1 触发和通信中断	0x120
73	49	可编程	TIM1_CC	TIM1 捕获比较中断	0x124
74	50	可编程	TIM2	TIM2 全局中断	0x128
75	51	可编程	TIM3	TIM3 全局中断	0x12C
76	52	可编程	TIM4	TIM4 全局中断	0x130
77	53	可编程	TIM5	TIM5 全局中断	0x134
78	54	可编程	I2C3_EV	I ² C3 事件中断	0x138

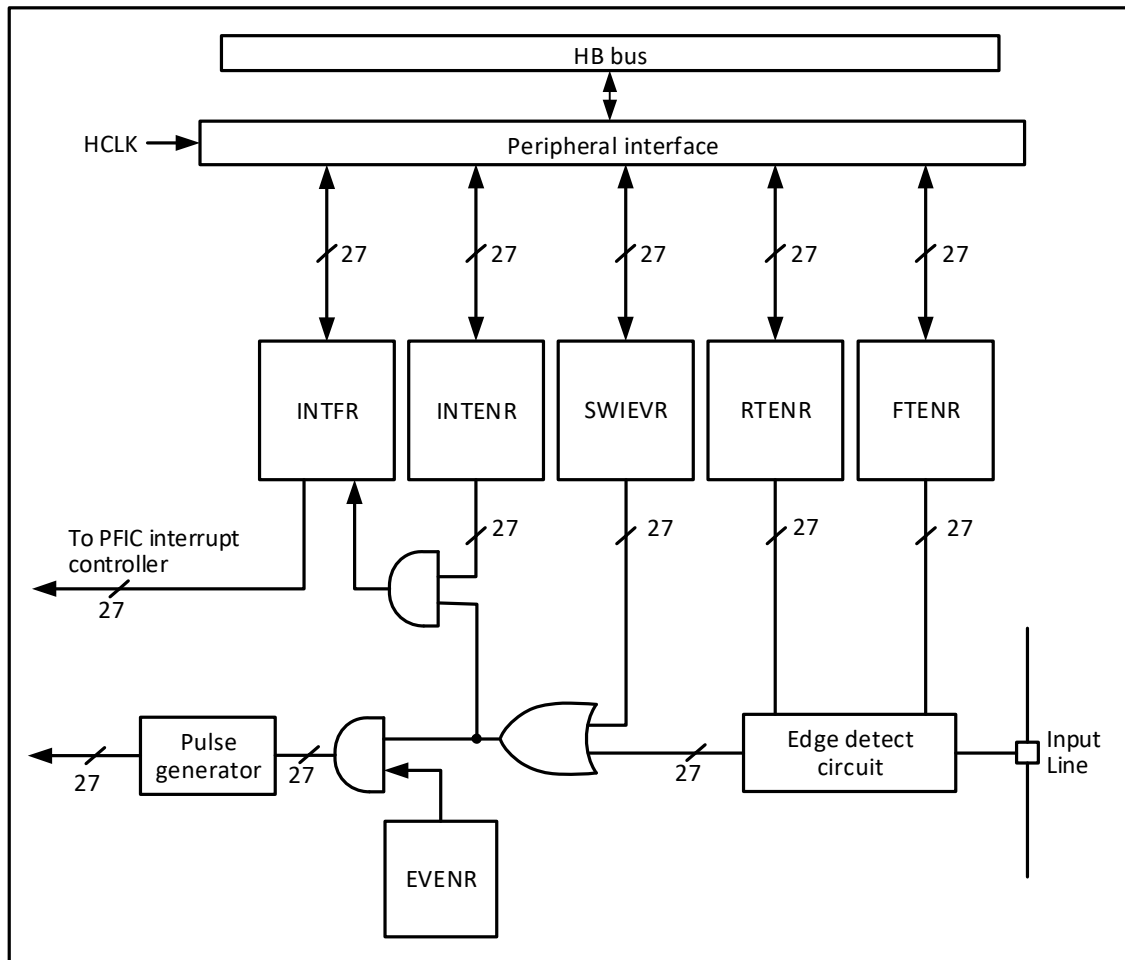
79	55	可编程	I2C3_ER	I ² C3 错误中断	0x13C
80	56	可编程	I2C4_EV	I ² C4 事件中断	0x140
81	57	可编程	I2C4_ER	I ² C4 错误中断	0x144
82	58	可编程	QSPI1	QSPI1 全局中断	0x148
83	59	可编程	SERDES	SERDES 全局中断	0x14C
84	60	可编程	USART3	USART3 全局中断	0x150
85	61	可编程	USART4	USART4 全局中断	0x154
86	62	可编程	TIM8_BRK	TIM8 刹车中断	0x158
87	63	可编程	TIM8_UP	TIM8 更新中断	0x15C
88	64	可编程	TIM8_TRG_COM	TIM8 触发和通信中断	0x160
89	65	可编程	TIM8_CC	TIM8 捕获比较中断	0x164
90	66	可编程	TIM9	TIM9 全局中断	0x168
91	67	可编程	TIM10	TIM10 全局中断	0x16C
92	68	可编程	TIM11	TIM11 全局中断	0x170
93	69	可编程	TIM12	TIM12 全局中断	0x174
94	70	可编程	FMC	FMC 全局中断	0x178
95	71	可编程	SDMMC	SDMMC 全局中断	0x17C
96	72	可编程	LPTIM1	LPTIM1 全局中断	0x180
97	73	可编程	LPTIM2	LPTIM2 全局中断	0x184
98	74	可编程	USART5	USART5 全局中断	0x188
99	75	可编程	USART6	USART6 全局中断	0x18C
100	76	可编程	TIM6	TIM6 全局中断	0x190
101	77	可编程	TIM7	TIM7 全局中断	0x194
102	78	可编程	DMA2_CH1	DMA2 通道 1 全局中断	0x198
103	79	可编程	DMA2_CH2	DMA2 通道 2 全局中断	0x19C
104	80	可编程	DMA2_CH3	DMA2 通道 3 全局中断	0x1A0
105	81	可编程	DMA2_CH4	DMA2 通道 4 全局中断	0x1A4
106	82	可编程	DMA2_CH5	DMA2 通道 5 全局中断	0x1A8
107	83	可编程	DMA2_CH6	DMA2 通道 6 全局中断	0x1AC
108	84	可编程	DMA2_CH7	DMA2 通道 7 全局中断	0x1B0
109	85	可编程	DMA2_CH8	DMA2 通道 8 全局中断	0x1B4
110	86	可编程	ETH	ETH 全局中断	0x1B8
111	87	可编程	ETH_WKUP	ETH 唤醒中断	0x1BC
112	88	可编程	CAN2_SCE	CAN2_SCE 全局中断	0x1C0
113	89	可编程	CAN2_TX	CAN2_TX 全局中断	0x1C4
114	90	可编程	CAN2_RX0	CAN2_RX0 全局中断	0x1C8
115	91	可编程	CAN2_RX1	CAN2_RX1 全局中断	0x1CC
116	92	可编程	USART7	USART7 全局中断	0x1D0
117	93	可编程	USART8	USART8 全局中断	0x1D4
118	94	可编程	I3C_EV	I ³ C 事件中断	0x1D8
119	95	可编程	I3C_ER	I ³ C 错误中断	0x1DC
120	96	可编程	DVP	DVP 全局中断	0x1E0
121	97	可编程	ECDC	ECDC 全局中断	0x1E4
122	98	可编程	PIOC	PIOC 全局中断	0x1E8
123	99	可编程	SAI	SAI 全局中断	0x1EC

124	100	可编程	LTDC	LTDC 全局中断	0x1F0
125	101	可编程	GPHA	GPHA 全局中断	0x1F4
126	—	—	—	保留	0x1F8
127	102	可编程	DFSDM0	DFSDM0 全局中断	0x1FC
128	103	可编程	DFSDM1	DFSDM1 全局中断	0x200
129	—	—	—	保留	0x204
130	—	—	—	保留	0x208
131	104	可编程	SWPMI	SWPMI 全局中断	0x20C
132	—	—	—	保留	0x210
133	—	—	—	保留	0x214
134	105	可编程	QSPI2	QSPI2 全局中断	0x218
135	106	可编程	SWPMIWakeUP	SWPMI 唤醒中断	0x21C
136	107	可编程	CAN3_SCE	CAN3_SCE 全局中断	0x220
137	108	可编程	CAN3_TX	CAN3_TX 全局中断	0x224
138	109	可编程	CAN3_RX0	CAN3_RX0 全局中断	0x228
139	110	可编程	CAN3_RX1	CAN3_RX1 全局中断	0x22C
140	111	可编程	LPTIM2WakeUP	LPTIM2 唤醒中断	0x230
141	112	可编程	LPTIM1WakeUP	LPTIM1 唤醒中断	0x234
142	113	可编程	I3CWakeUP	I ³ C 唤醒中断	0x238
143	114	可编程	RTC	RTC 全局中断	0x23C
144	115	可编程	HSADC	HSADC 全局中断	0x240
145	116	可编程	UHSIF	UHSIF 全局中断	0x244
146	117	可编程	RNG	RNG 全局中断	0x248
147	118	可编程	SDIO	SDIO 全局中断	0x24C
148	119	可编程	USARTWakeUP	USART 唤醒中断	0x250

4.7.3 外部中断和事件控制器 (EXTI)

4.7.3.1 概述

图 4-1 外部中断 (EXTI) 接口框图



由图 4-1 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 HB 接口访问。

4.7.3.2 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。唤醒事件通过以下三种配置产生：

- 在外设的寄存器里使能一个中断，但不在内核的 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 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 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 PFIC 挂起位的操作。
- 由另一处于运行状态的内核写 SENDEVENT 寄存器产生一次写事件。

4.7.3.3 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 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_EVENTR)；
- 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_EVENTR)；
- 2) 如果使用中断服务函数，需要设置内核的 PFIC 里 EXTI 中断；
- 3) 设置软件中断触发 (EXTI_SWIEVR)，即会产生中断。

4.7.3.4 外部事件映射

表 4-12 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	USBHS 唤醒事件
EXTI17	RTC 闹钟事件
EXTI18	SWPMI 唤醒事件
EXTI19	USBFSOTG 唤醒事件
EXTI20	USBPD 唤醒事件
EXTI21	ETH 唤醒事件
EXTI22	USBSS 唤醒事件
EXTI23	LPTIM1 唤醒事件
EXTI24	LPTIM2 唤醒事件
EXTI25	I3C 唤醒事件
EXTI26	USART 唤醒事件

4.7.4 EXTI 寄存器描述

表 4-13 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_EVENTR	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

4.7.4.1 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					MR26	MR25	MR24	MR23	MR22	MR21	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
[31:27]	Reserved	R0	保留。	0
[26:0]	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

4.7.4.2 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					MR26	MR25	MR24	MR23	MR22	MR21	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
[31:27]	Reserved	R0	保留。	0
[26:0]	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

4.7.4.3 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					TR26	TR25	TR24	TR23	TR22	TR21	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
[31:27]	Reserved	R0	保留。	0
[26:0]	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

4.7.4.4 下降沿触发使能寄存器 (EXTI_FTEMR)

偏移地址: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					TR26	TR25	TR24	TR23	TR22	TR21	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
[31:27]	Reserved	R0	保留	0
[26:0]	TRx	RW	使能外部中断通道 x 的下降沿触发： 1: 使能此通道的下降沿触发； 0: 禁止此通道的下降沿触发。	0

4.7.4.5 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 ₂₆	SWIER ₂₅	SWIER ₂₄	SWIER ₂₃	SWIER ₂₂	SWIER ₂₁	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
[31:27]	Reserved	R0	保留。	0
[26:0]	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_EVENTR) 开启, 那么就会产生中断或事件。	0

4.7.4.6 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					IF26	IF25	IF24	IF23	IF22	IF21	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
[31:27]	Reserved	R0	保留。	0
[26:0]	IFx	RW1Z	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

4.7.5 PFIC 寄存器

表 4-14 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	0x0000032C
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_ISR5	0xE000E010	PFIC 中断使能状态寄存器 5	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_IPR5	0xE000E030	PFIC 中断挂起状态寄存器 5	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	0xFFFFFFFF
R32_PFIC_VTFADDRR0	0xE000E060	PFIC VTF 中断 0 偏移地址寄存器	0xFFFFFFFF
R32_PFIC_VTFADDRR1	0xE000E064	PFIC VTF 中断 1 偏移地址寄存器	0xFFFFFFFF
R32_PFIC_VTFADDRR2	0xE000E068	PFIC VTF 中断 2 偏移地址寄存器	0xFFFFFFFF
R32_PFIC_VTFADDRR3	0xE000E06C	PFIC VTF 中断 3 偏移地址寄存器	0xFFFFFFFF
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_IENR5	0xE000E110	PFIC 中断使能设置寄存器 5	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_IRER5	0xE000E190	PFIC 中断使能清除寄存器 5	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_IPSR5	0xE000E210	PFIC 中断挂起设置寄存器 5	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_IPRR5	0xE000E290	PFIC 中断挂起清除寄存器 5	0x00000000
R32_PFIC_IACR1	0xE000E300	PFIC 中断激活状态寄存器 1	0x00000000
R32_PFIC_IACR2	0xE000E304	PFIC 中断激活状态寄存器 2	0x00000000
R32_PFIC_IACR3	0xE000E308	PFIC 中断激活状态寄存器 3	0x00000000
R32_PFIC_IACR4	0xE000E30C	PFIC 中断激活状态寄存器 4	0x00000000

R32_PFIC_IACR5	0xE000E310	PFIC 中断激活状态寄存器 5	0x00000000
R32_PFIC_IPRIORx	0xE000E400	PFIC 中断优先级配置寄存器	0x00000000
R32_PFIC_IALL0CRx	0xE000E600	PFIC 中断分配寄存器	0xFFFFFFFF
R32_PFIC_IAUTR1	0xE000E700	PFIC 中断权柄寄存器 1	0xFFFFFFFF
R32_PFIC_IAUTR2	0xE000E704	PFIC 中断权柄寄存器 2	0xFFFFFFFF
R32_PFIC_IAUTR3	0xE000E708	PFIC 中断权柄寄存器 3	0xFFFFFFFF
R32_PFIC_IAUTR4	0xE000E70C	PFIC 中断权柄寄存器 4	0xFFFFFFFF
R32_PFIC_IAUTR5	0xE000E710	PFIC 中断权柄寄存器 5	0xFFFFFFFF
R32_PFIC_WAKEIP0	0xE000E720	PFIC 唤醒指令指针寄存器 0	0x00000001
R32_PFIC_WAKEIP1	0xE000E724	PFIC 唤醒指令指针寄存器 1	0x00000000
R32_PFIC_CSTAR0	0xE000E780	PFIC 内核状态寄存器 0	0x00000000
R32_PFIC_CSTAR1	0xE000E784	PFIC 内核状态寄存器 1	0x00000000
R32_PFIC_EENR	0xE000EC80	PFIC 事件使能寄存器	0xFFFFFFFF
R32_PFIC_EPR	0xE000EC84	PFIC 事件挂起寄存器	0x00000000
R32_PFIC_EWUPR	0xE000EC88	PFIC 事件唤醒寄存器	0x00000000
R32_PFIC_SCTLR	0xE000ED10	PFIC 系统控制寄存器	0x00000000

注：1. NMI、HardFault、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。

4.7.5.1 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
INTENSTA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSTA15	INTENSTA14	INTENSTA13	INTENSTA12	Reserved	INTENSTA9	INTENSTA8	Reserved	INTENSTA5	Reserved	INTENSTA3	INTENSTA2	Reserved			

位	名称	访问	描述	复位值
[31:12]	INTENSTA	RO	12#-31#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
[11:10]	Reserved	RO	保留。	0
[9:8]	INTENSTA	RO	8#-9#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。	0x3
[7:6]	Reserved	RO	保留。	0
5	INTENSTA	RO	5#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。	1
4	Reserved	RO	保留。	0
[3:2]	INTENSTA	RO	2#-3#中断当前使能状态。 1：当前编号中断已使能；	0x3

			0: 当前编号中断未启用。	
[1:0]	Reserved	R0	保留。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.2 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
INTENSTA[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSTA[47:32]															

位	名称	访问	描述	复位值
[31:0]	INTENSTA	R0	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

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.3 PFIC 中断使能状态寄存器 3 (PFIC_ISR3)

偏移地址：0x08

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

位	名称	访问	描述	复位值
[31:0]	INTENSTA	R0	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

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.4 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
INTENSTA[127:112]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSTA[111:96]															

位	名称	访问	描述	复位值

[31:0]	INTENSTA	R0	96#-127#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

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.5 PFIC 中断使能状态寄存器 5 (PFIC_ISR5)

偏移地址：0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											INTENSTA[148:144]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSTA[143:128]															

位	名称	访问	描述	复位值
[31:21]	Reserved	R0	保留。	0
[20:0]	INTENSTA	R0	128#-148#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

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.6 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
PENDSTA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDS TA15	PENDS TA14	PENDS TA13	PENDS TA12	Reserved		PENDS TA9	PENDS TA8	Reserved		PENDS TA5	Reser ved	PENDS TA3	PENDS TA2	Reserved	

位	名称	访问	描述	复位值
[31:12]	PENDSTA	R0	12#-31#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
[11:10]	Reserved	R0	保留。	0
[9:8]	PENDSTA	R0	8#-9#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
[7:6]	Reserved	R0	保留。	0
5	PENDSTA	R0	5#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
4	Reserved	R0	保留。	0
[3:2]	PENDSTA	R0	2#-3#中断当前挂起状态。	0

			1: 当前编号中断已挂起; 0: 当前编号中断未挂起。	
[1:0]	Reserved	R0	保留。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.7 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
PENDSTA[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSTA[47:32]															

位	名称	访问	描述	复位值
[31:0]	PENDSTA	R0	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

4.7.5.8 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
PENDSTA[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSTA[79:64]															

位	名称	访问	描述	复位值
[31:0]	PENDSTA	R0	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

4.7.5.9 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
PENDSTA[127:112]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSTA[111:96]															

位	名称	访问	描述	复位值
[31:0]	PENDSTA	R0	96#-127#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

4.7.5.10 PFIC 中断挂起状态寄存器 5 (PFIC_IPR5)

偏移地址: 0x30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											INTENSTA[148:144]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSTA[143:128]															

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留	0
[20:0]	PENDSTA	RO	128#-148#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

4.7.5.11 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
Reserved																								THRESHOLD[7:0]							

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。 中断优先级低于所设阈值的中断在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。	0

注: 该寄存器为内核私有寄存器, 即外设中存在多组功能相同, 数据相互独立, 数据不可被其他内核访问或改写的寄存器, 仅内核对其访问地址相同。

4.7.5.12 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								SYSRST	Reserved						

位	名称	访问	描述	复位值
[31:16]	KEYCODE[15:0]	WO	key 控制关键字, 写入 beef 时有效, 否则操作无效。	0
[15:8]	Reserved	RO	保留。	0
7	SYSRST	WO	复位寄存器, 系统复位写入 0xBEEF0080 时系统复	0

			位，其他值无效。 注：与 PFIC_SCTLR 寄存器 SYSRST 位作用相同。	
[6:0]	Reserved	R0	保留。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.13 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
EX_STATE [1:0]	LOCK _UP	DBG_ MODE	GLOB L_IE	Rese rved	GPEN DSTA	GACT STA	NESTSTA[7:0]								

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:14]	EX_STATE[1:0]	R0	内核状态寄存器，用于获取内核的运行状态 00：运行； 01：浅睡眠； 10：深睡眠； 11：深睡眠（锁定）。	0
13	LOCK_UP	R0	内核锁定状态寄存器，用于查询内核是否处于锁定态。 1：锁定； 0：非锁定。	0
12	DBG_MODE	R0	内核调试模式寄存器，用于查询内核是否处于调试模式 1：内核处于调试模式； 0：内核处于非调试模式。	0
11	GLOBL_IE	R0	内核全局中断使能寄存器，用于查询内核全局中断是否使能 1：内核全局中断使能开启； 0：内核全局中断使能关闭。	0
10	Reserved	R0	保留。	0
9	GPENDSTA	R0	当前是否有中断处于挂起： 1：有； 0：没有。	0
8	GACTSTA	R0	当前是否有中断被执行： 1：有； 0：没有。	0
[7:0]	NESTSTA	R0	当前中断嵌套状态，用于查询内核中断的嵌套状态。同 CSR 寄存器的 INEST_CTLR 寄存器 nest_sta 位。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.14 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。	X
[23:16]	VTFID2	RW	配置 VTF 中断 2 的中断编号。	X
[15:8]	VTFID1	RW	配置 VTF 中断 1 的中断编号。	X
[7:0]	VTFID0	RW	配置 VTF 中断 0 的中断编号。	X

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.15 PFIC VTF 中断 0 地址寄存器 (PFIC_VTFADDR0)

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]。	X
0	VTF0EN	RW	VTF 中断 0 使能位： 1：启用 VTF 中断 0 通道； 0：关闭。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.16 PFIC VTF 中断 1 地址寄存器 (PFIC_VTFADDR1)

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]。	X
0	VTF1EN	RW	VTF 中断 1 使能位：	0

			1: 启用 VTF 中断 1 通道; 0: 关闭。	

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.17 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]。	X
0	VTF2EN	RW	VTF 中断 2 使能位： 1: 启用 VTF 中断 2 通道； 0: 关闭。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.18 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]。	X
0	VTF3EN	RW	VTF 中断 3 使能位： 1: 启用 VTF 中断 3 通道； 0: 关闭。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.19 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
INTEN15	INTEN14	INTEN13	INTEN12	Reserved											

位	名称	访问	描述	复位值
[31:12]	INTEN	WO	12#-31#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

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.20 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

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.21 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

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.22 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
INTEN[127:112]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

INTEN[111:96]

位	名称	访问	描述	复位值
[31:0]	INTEN	WO	96#-127#中断使能控制。 1: 当前编号中断使能; 0: 无影响。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.23 PFIC 中断使能设置寄存器 5 (PFIC_IENR5)

偏移地址：0x110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											INTEN[148:144]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN[143:128]															

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:0]	INTEN	WO	128#-148#中断使能控制。 1: 当前编号中断使能; 0: 无影响。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.24 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
INTRSET[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRSET15	INTRSET14	INTRSET13	INTRSET12	Reserved											

位	名称	访问	描述	复位值
[31:12]	INTRSET	WO	12#-31#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

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.25 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
INTRSET [63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRSET [47:32]															

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

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.26 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
INTRSET [95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRSET [79:64]															

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

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.27 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
INTRSET [127:112]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRSET [111:96]															

位	名称	访问	描述	复位值
[31:0]	INTRSET	WO	96#-127#中断关闭控制。 1: 当前编号中断关闭; 0: 无影响。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.28 PFIC 中断使能清除寄存器 5 (PFIC_IRER5)

偏移地址: 0x190

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											INTRSET[148:144]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRSET[143:128]															

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:0]	INTRSET	WO	128#-148#中断关闭控制。 1: 当前编号中断关闭; 0: 无影响。	0

注: 该寄存器为内核私有寄存器, 即外设中存在多组功能相同, 数据相互独立, 数据不可被其他内核访问或改写的寄存器, 仅内核对其访问地址相同。

4.7.5.29 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 SET15	PEND SET14	PEND SET13	PEND SET12	Reserved		PEND SET9	PEND SET8	Reserved		PEND SET5	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
[11:10]	Reserved	RO	保留。	0
[9:8]	PENDSET	RO	8#-9#中断挂起设置。 1: 当前编号中断挂起; 0: 无影响。	0
[7:6]	Reserved	RO	保留。	0
5	PENDSET	RO	5#中断挂起设置。 1: 当前编号中断挂起; 0: 无影响。	0
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

注: 该寄存器为内核私有寄存器, 即外设中存在多组功能相同, 数据相互独立, 数据不可被其他内核访问或改写的寄存器, 仅内核对其访问地址相同。

4.7.5.30 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

4.7.5.31 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

4.7.5.32 PFIC 中断挂起设置寄存器 4 (PFIC_IPSR4)

偏移地址: 0x20C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSET [127:112]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSET [111:96]															

位	名称	访问	描述	复位值
[31:0]	PENDSET	WO	96#-127#中断挂起设置。 1: 当前编号中断挂起; 0: 无影响。	0

4.7.5.33 PFIC 中断挂起设置寄存器 5 (PFIC_IPSR5)

偏移地址: 0x210

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											PENDSET [148:144]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSET[143:128]															

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:0]	PENDSET	WO	128#-148#中断挂起设置。 1: 当前编号中断挂起; 0: 无影响。	0

4.7.5.34 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
PENDRST[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEND RST15	PEND RST14	PEND RST13	PEND RST12	Reserved	PEND RST9	PEND RST8	Reserved	PEND RST5	Reserved	PEND RST3	PEND RST2	Reserved			

位	名称	访问	描述	复位值
[31:12]	PENDRST	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
[11:10]	Reserved	RO	保留。	0
[9:8]	PENDRST	RO	8#-9#中断挂起清除。 1: 当前编号中断清除挂起状态; 0: 无影响。	0
[7:6]	Reserved	RO	保留。	0
5	PENDRST	RO	5#中断挂起清除。 1: 当前编号中断清除挂起状态; 0: 无影响。	0
4	Reserved	RO	保留。	0
[3:2]	PENDRST	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

注: 该寄存器为内核私有寄存器, 即外设中存在多组功能相同, 数据相互独立, 数据不可被其他内核访问或改写的寄存器, 仅内核对其访问地址相同。

4.7.5.35 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
PENDRST[63:48]															

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

PENDRST[47:32]

位	名称	访问	描述	复位值
[31:0]	PENDRST	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

4.7.5.36 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

PENDRST[95:80]

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

PENDRST[79:64]

位	名称	访问	描述	复位值
[31:0]	PENDRST	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

4.7.5.37 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

PENDRST[127:112]

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

PENDRST[111:96]

位	名称	访问	描述	复位值
[31:0]	PENDESET	WO	96#-127#中断挂起清除。 1: 当前编号中断清除挂起状态; 0: 无影响。	0

4.7.5.38 PFIC 中断挂起清除寄存器 5 (PFIC_IPRR5)

偏移地址: 0x290

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

Reserved

PENDRST[148:144]

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

PENDRST[143:128]

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:0]	PENDESET	WO	128#-148#中断挂起清除。	0

			1: 当前编号中断清除挂起状态; 0: 无影响。	

4.7.5.39 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
IACTS15	IACTS14	IACTS13	IACTS12	Reserved								IACTS3	IACTS2	Reserved	

位	名称	访问	描述	复位值
[31:12]	IACTS	R0	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	R0	保留。	0
[3:2]	IACTS	R0	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	R0	保留。	0

注: 该寄存器为内核私有寄存器, 即外设中存在多组功能相同, 数据相互独立, 数据不可被其他内核访问或改写的寄存器, 仅内核对其访问地址相同。

4.7.5.40 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	R0	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

4.7.5.41 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

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

偏移地址: 0x30C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IACTS[127:112]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IACTS[111:96]															

位	名称	访问	描述	复位值
[31:0]	IACTS	WO	96#-127#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

4.7.5.43 PFIC 中断激活状态寄存器 5 (PFIC_IACTR5)

偏移地址: 0x310

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											IACTS[148:144]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IACTS[143:128]															

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:0]	IACTS	WO	128#-148#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

4.7.5.44 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			
...	...				...			
IPRIORx	PRIO_ (4x+3)				PRIO_ (4x+2)			
...	...				...			
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，数值越小，优先级越高，其中抢占位受 CSR 寄存器 preempt 控制。 preempt 寄存器值为 0 时，无抢占位。 preemempt 寄存器值为 1 时，bit7 为抢占位。 preempt 寄存器值为 2 时，bit7-6 为抢占位。 preempt 寄存器值位 3 时，bit7-5 为抢占位。 V3F 内核最多支持 2 级嵌套，嵌套满时无法继续抢占。 V5F 内核最多支持 8 级嵌套，嵌套级数可配，嵌套满时无法继续抢占。 [3:0]位固定为 0。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.45 PFIC 中断分配寄存器 (PFIC_IALLOCx) (x=0~63)

偏移地址：0x600~0x6FF

	31	24	23	16	15	8	7	0
IALLOC63	IALLOC_255				IALLOC_252			
...	...				...			
IALLOCx	IALLOC_(4x+3)				IALLOC_(4x)			
...	...				...			
IALLOC0	IALLOC_3				IALLOC_0			

位	名称	访问	描述	复位值
[2047:2040]	IALLOC_255	RW	同 IALLOC_32 描述。	X
...	...	...	...	...
[263:256]	IALLOC_33	RW	同 IALLOC_32 描述。	X
[255:248]	IALLOC_32	RW	编号 32 中断分配寄存器，本系统存在两个内核，仅第 0 位可配置；当本中断的 ENA 被某一内核使能时，本寄存器自	X

			动更新为操作 ENA 寄存器的内核 ID 值。 1: 本中断分配到 C1 处理; 0: 本中断分配到 C0 处理。	
[247:240]	IALLOC_31	RO	同 IALLOC_0 描述。	X
...	...	...	...	...
[31:24]	IALLOC_3	RO	同 IALLOC_0 描述。	X
[23:16]	IALLOC_2	RO	同 IALLOC_0 描述。	X
[15:8]	IALLOC_1	RO	同 IALLOC_0 描述。	X
[7:0]	IALLOC_0	RO	编号 0 中断分配: 本系统存在两个内核, 仅第 0 位可配置, C0 内核读取值为 0, C1 内核读取值为 1 1: 本中断分配到 C1 处理; 0: 本中断分配到 C0 处理。	X

4.7.5.46 PFIC 中断权柄寄存器 1 (PFIC_IAUTR1)

偏移地址: 0x700

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

位	名称	访问	描述	复位值
[31:0]	IAUT	RO	31#-0#中断分配全部寄存器, 用于查询中断是否每分配到本内核响应。 1: 中断分配到本内核; 0: 中断未分配到本内核。	0xFFFF FFF

注: 该寄存器为内核私有寄存器, 即外设中存在多组功能相同, 数据相互独立, 数据不可被其他内核访问或改写的寄存器, 仅内核对其访问地址相同。

4.7.5.47 PFIC 中断权柄寄存器 2 (PFIC_IAUTR2)

偏移地址: 0x704

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

位	名称	访问	描述	复位值
[31:0]	IAUT	RO	32#-63#中断分配全部寄存器, 用于查询中断是否每分配到本内核响应。 1: 中断分配到本内核; 0: 中断未分配到本内核。	X

注: 该寄存器为内核私有寄存器, 即外设中存在多组功能相同, 数据相互独立, 数据不可被其他内核

核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.48 PFIC 中断权柄寄存器 3 (PFIC_IAUTR3)

偏移地址：0x708

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

位	名称	访问	描述	复位值
[31:0]	IAUT	RO	64#-95#中断分配全部寄存器，用于查询中断是否每分配到本内核响应。 1：中断分配到本内核； 0：中断未分配到本内核。	X

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.49 PFIC 中断权柄寄存器 4 (PFIC_IAUTR4)

偏移地址：0x70C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IAUT[127:112]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IAUT[111:96]															

位	名称	访问	描述	复位值
[31:0]	IAUT	RO	96#-127#中断分配全部寄存器，用于查询中断是否每分配到本内核响应。 1：中断分配到本内核； 0：中断未分配到本内核。	X

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.50 PFIC 中断权柄寄存器 5 (PFIC_IAUTR5)

偏移地址：0x710

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											IAUT[148:144]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IAUT[143:128]															

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0

[20:0]	IAUT	RO	128#-148#中断分配全部寄存器，用于查询中断是否每分配到本内核响应。 1：中断分配到本内核； 0：中断未分配到本内核。	X

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.51 PFIC 唤醒指令指针寄存器 0 (PFIC_WAKEIP0)

偏移地址：0x720

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IP_RELOAD0[30:15]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP_RELOAD0[14:0]														SHUTD OWN0	

位	名称	访问	描述	复位值
[31:1]	IP_RELOAD0	RW	内核 C0 唤醒时 PC 地址寄存器，用于上电唤醒和复位唤醒后 PC 值的重装载。	0
0	SHUTDOWN0	RW	内核 C0 深睡眠（锁定）取消寄存器，用于上电后进入深睡眠（锁定）状态的内核取消锁定，允许被时间唤醒： 1：锁定； 0：解锁锁定。	1

4.7.5.52 PFIC 唤醒指令指针寄存器 1 (PFIC_WAKEIP1)

偏移地址：0x724

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IP_RELOAD1[30:15]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP_RELOAD1[14:0]														SHUTD OWN1	

位	名称	访问	描述	复位值
[31:1]	IP_RELOAD1[30:0]	RW	内核 C1 醒时 PC 地址寄存器，用于上电唤醒和复位唤醒后 PC 值的重装载。	0
0	SHUTDOWN1	RW	内核 C1 睡眠（锁定）取消寄存器，用于上电后进入深睡眠（锁定）状态的内核取消锁定，允许被时间唤醒： 1：锁定； 0：解锁锁定。	0

4.7.5.53 PFIC 内核状态寄存器 0 (PFIC_CSTAR0)

偏移地址：0x780

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
CPU_ex_state_0	CPU_lock_up_0	CPU_dbg_mode_0	CPU_global_ie_0	Reserved	CPU_irq_pending_0	CPU_irq_active_0	CPU_nest_sta_0								

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:14]	CPU_ex_state_0	R0	内核 C0 状态寄存器，用于获取内核 C0 的运行状态 00: 运行； 01: 浅睡眠； 10: 深睡眠； 11: 深睡眠（锁定）。	0
13	CPU_lock_up_0	R0	内核 C0 锁定状态寄存器，用于查询内核 C0 是否处于锁定态： 1: 锁定； 0: 非锁定。	0
12	CPU_dbg_mode_0	R0	内核 C0 调试模式寄存器，用于查询内核 C0 是否处于调试模式： 1: 内核处于调试模式； 0: 内核处于非调试模式。	0
11	CPU_global_ie_0	R0	内核 C0 全局中断使能寄存器，用于查询内核 C0 全局中断是否使能： 1: 内核全局中断使能开启； 0: 内核全局中断使能关闭。	0
10	Reserved	R0	保留。	0
9	CPU_irq_pending_0	R0	内核 C0 中断挂起标志寄存器，用于查询内核 C0 是否存在未处理的中断： 1: 内核存在未处理的中断； 0: 内核不存在未处理的中断。	0
8	CPU_irq_active_0	R0	内核 C0 中断活跃标志寄存器，用于查询内核 C0 是否正在处理中断 1: 内核正在处理中断； 0: 内核未处理中断。	0
[7:0]	CPU_nest_sta_0	R0	内核 C0 中断嵌套状态寄存器，用于查询内核 C0 中断的嵌套状态。同 CSR 寄存器的 INEST_CTLR 寄存器 nest_sta 位。	0

4.7.5.54 PFIC 内核状态寄存器 1 (PFIC_CSTAR1)

偏移地址：0x784

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
CPU_ex_state_1	CPU_lock_up_1	CPU_dbg_mode_1	CPU_global_ie_1	Reserved	CPU_irq_pending_1	CPU_irq_active_1	CPU_nest_sta_1								

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:14]	CPU_ex_state_1	R0	内核 C1 状态寄存器，用于获取内核 C1 的运行状态： 00：运行； 01：浅睡眠； 10：深睡眠； 11：深睡眠（锁定）。	0
13	CPU_lock_up_1	R0	内核 C1 锁定状态寄存器，用于查询内核 C1 是否处于锁定态： 1：锁定； 0：非锁定。	0
12	CPU_dbg_mode_1	R0	内核 C1 调试模式寄存器，用于查询内核 C1 是否处于调试模式： 1：内核处于调试模式； 0：内核处于非调试模式。	0
11	CPU_global_ie_1	R0	内核 C1 全局中断使能寄存器，用于查询内核 C1 全局中断是否使能： 1：内核全局中断使能开启； 0：内核全局中断使能关闭。	0
10	Reserved	R0	保留。	0
9	CPU_irq_pending_1	R0	内核 C1 中断挂起标志寄存器，用于查询内核 C1 是否存在未处理的中断： 1：内核存在未处理的中断； 0：内核不存在未处理的中断。	0
8	CPU_irq_active_1	R0	内核 C1 中断活跃标志寄存器，用于查询内核 C1 是否正在处理中断： 1：内核正在处理中断； 0：内核未处理中断。	0
[7:0]	CPU_nest_sta_1	R0	内核 C1 中断嵌套状态寄存器，用于查询内核 C1 中断的嵌套状态。同 CSR 寄存器的 INEST_CTLR 寄存器 nest_sta 位。	0

4.7.5.55 PFIC 事件使能寄存器 (PFIC_EENR)

偏移地址：0xC80

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

位	名称	访问	描述	复位值
[31:0]	EVENTEN	RW	31#-0#事件唤醒使能： 1：允许对应事件在睡眠中唤醒内核； 0：屏蔽对应事件，本事件发生时不允许唤醒内核。 最多支持 32 个事件源，其中 0-7 为电平触发（高有效），8-31 为脉冲触发： 0#：内核暂停请求； 1#：内核中断请求； 7#-2#：保留； 8#：内核进入中断； 9#：内核退出中断； 10#：新的中断挂起；（对于共享中断，不区分该中断是否路由至本地内核） 11#：内核发送事件；（本地内核的发送事件） 12#：内核接收事件；（其它内核的发送事件） 13#：系统接收事件；（系统配置的事件，例如 GPIO 翻转事件…） 31#-14#：保留。	0xFFFF FFF

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.56 PFIC 事件挂起寄存器（PFIC_EPR）

偏移地址：0xC84

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

位	名称	访问	描述	复位值
[31:8]	EVENT_PEND	RW1Z	31#-8#事件挂起状态，写 1 可清零： 1：对应事件发生未被处理； 0：对应事件未发生。	0
[7:0]	EVENT_PEND	RO	7#-0#事件挂起状态，不可清除： 1：对应事件发生未被处理； 0：对应事件未发生。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

4.7.5.57 PFIC 事件唤醒寄存器（PFIC_EWUPR）

偏移地址：0xC88

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

位	名称	访问	描述	复位值
[31:0]	EVENT_WUP	RO	31#-0#事件唤醒寄存器： 1：对应事件将内核从睡眠中唤醒； 0：对应事件不是本次唤醒内核的事件。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

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

位	名称	访问	描述	复位值
31	SYSRST	RW1	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
[30:24]	Reserved	RO	保留。	0
[23:16]	HART_ID	RO	内核 ID 寄存器，用于获取读本寄存器的内核 ID： 1：本内核为 C1； 0：本内核为 C0。	0
[15:6]	Reserved	RO	保留。	0
5	SENDEVENT	WO	写事件，产生一次写事件，可用于唤醒其他内核。	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）都能唤醒系统系统； 0：只有启用的事件和启用的中断可以唤醒。	0
3	WFI TO WFE	RW	WFI 命令执行为 WFE，WFE 是自定义指令，本寄存器置 1 后，内核执行 WFI 指	0

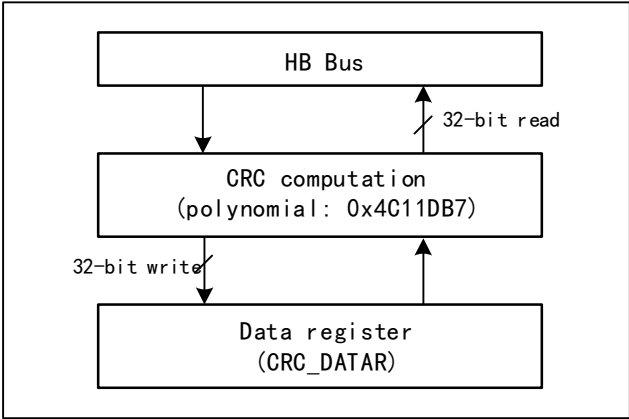
			令后进入低功耗状态，等待外部事件发生后可唤醒内核。 1：WFI 指令后等待事件唤醒内核； 0：WFI 指令后等待中断唤醒内核。	
2	SLEEPDEEP	RW	控制系统的低功耗模式： 1：deepsleep 0：sleep	0
1	SLEEPONEXIT	RW	控制离开中断服务程序后，系统状态： 1：系统进入低功耗模式； 0：系统进入主程序。	0
0	Reserved	RO	保留。	0

注：该寄存器为内核私有寄存器，即外设中存在多组功能相同，数据相互独立，数据不可被其他内核访问或改写的寄存器，仅内核对其访问地址相同。

第 5 章 循环冗余校验（CRC）

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 HB 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。

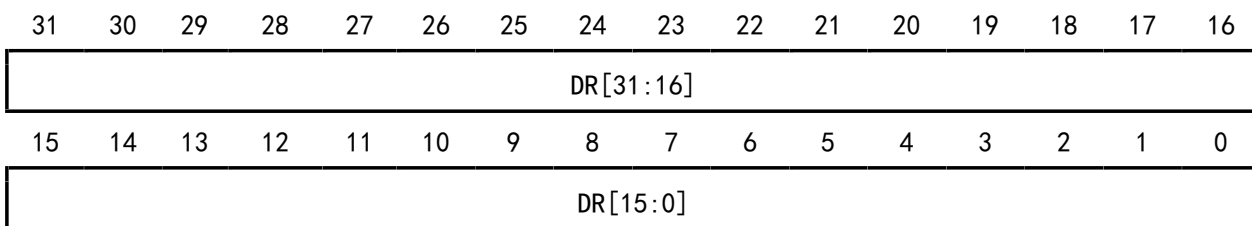
5.3 寄存器描述

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



位	名称	访问	描述	复位值
[31:0]	DR[31:0]	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

位	名称	访问	描述	复位值
[7:0]	IDR[7:0]	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



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)

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。

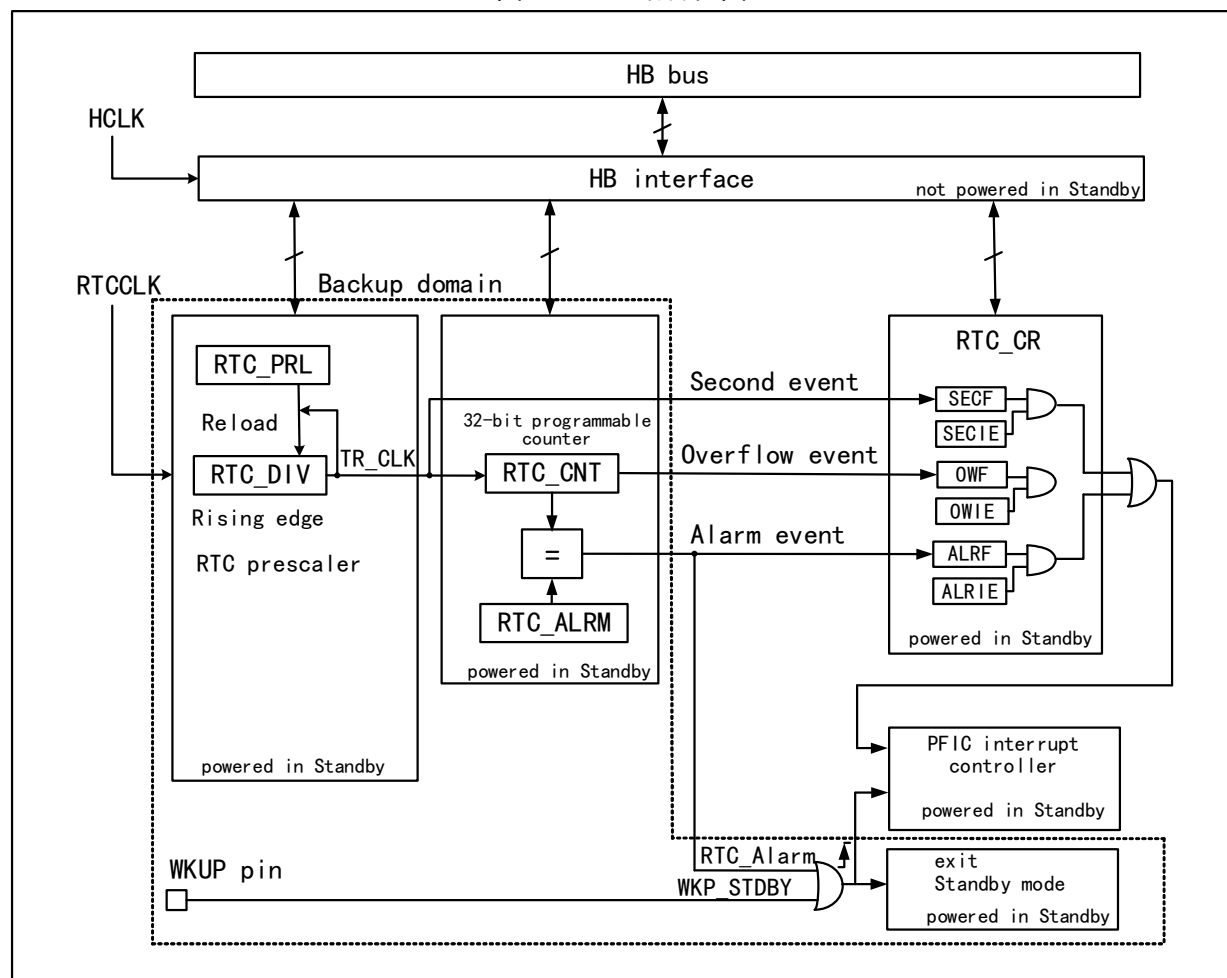
6.1 主要特征

- 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 HB 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。RTCCLK 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 HB 总线是独立的，HB 对 RTC 的读取不一定是实时的，通过 HB 读取 RTC 的寄存器必须在 HB 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、从停止模式唤醒后。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，HB 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	0x0020
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	0xFFFF
R16_RTC_DIVH	0x40002810	分频器寄存器高位	0xFFFF
R16_RTC_DIVL	0x40002814	分频器寄存器低位	0xFFFF
R16_RTC_CNTH	0x40002818	RTC 计数器高位	0xFFFF
R16_RTC_CNTL	0x4000281C	RTC 计数器低位	0xFFFF
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

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： 1：使能； 0：关闭。	0
1	ALRIE	RW	闹钟中断使能位： 1：使能； 0：关闭。	0

0	SECIE	RW	秒中断使能位： 1：使能； 0：关闭。	0

6.3.2 RTC 控制寄存器低位 (RTC_CTLRL)

偏移地址：0x04

位	名称	访问	描述	复位值
[15:6]	Reserved	R0	保留。	0
5	RTOFF	R0	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	RW0	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 HB 复位或 HB 时钟停止后，第一步应该将此位置位。 1：寄存器已被同步； 0：寄存器未被同步。	0
2	OWF	RW0	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	RW0	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	RW0	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中断。此位只能由软件清零，不能被软件置位。	0

6.3.3 预分频器重装值寄存器高位 (RTC_PSCRH)

偏移地址：0x08

位	名称	访问	描述	复位值

[15:4]	Reserved	RO	保留。	0
[3:0]	PRL[19:16]	WO	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

位	名称	访问	描述	复位值
[15:0]	PRL[15:0]	WO	重装值低位。实际的分频系数就是 (PRL[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。	X

6.3.5 分频器寄存器高位 (RTC_DIVH)

偏移地址：0x10

位	名称	访问	描述	复位值
[15:4]	Reserved	RO	保留。	0
[3:0]	DIV[19:16]	RO	分频器寄存器高位。	x

6.3.6 分频器寄存器低位 (RTC_DIVL)

偏移地址：0x14

位	名称	访问	描述	复位值
[15:0]	DIV[15:0]	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 中重装值。DIV 只能读取，读出的是当前分频器的计数器的剩余值。	X

6.3.7 RTC 计数器高位 (RTC_CNTH)

偏移地址：0x18

位	名称	访问	描述	复位值
[15:0]	CNT[31:16]	RW	计数器高位。	X

6.3.8 RTC 计数器低位 (RTC_CNTL)

偏移地址：0x1C

位	名称	访问	描述	复位值
[15:0]	CNT[15:0]	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。	X

6.3.9 闹钟寄存器高位 (RTC_ALRMH)

偏移地址：0x20

位	名称	访问	描述	复位值
[15:0]	ALR[31:16]	RW	闹钟寄存器高位。	X

6.3.10 闹钟寄存器低位（RTC_ALRML）

偏移地址：0x24

位	名称	访问	描述	复位值
[15:0]	ALR[15:0]	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。	X

第 7 章 独立看门狗（IWDG）

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

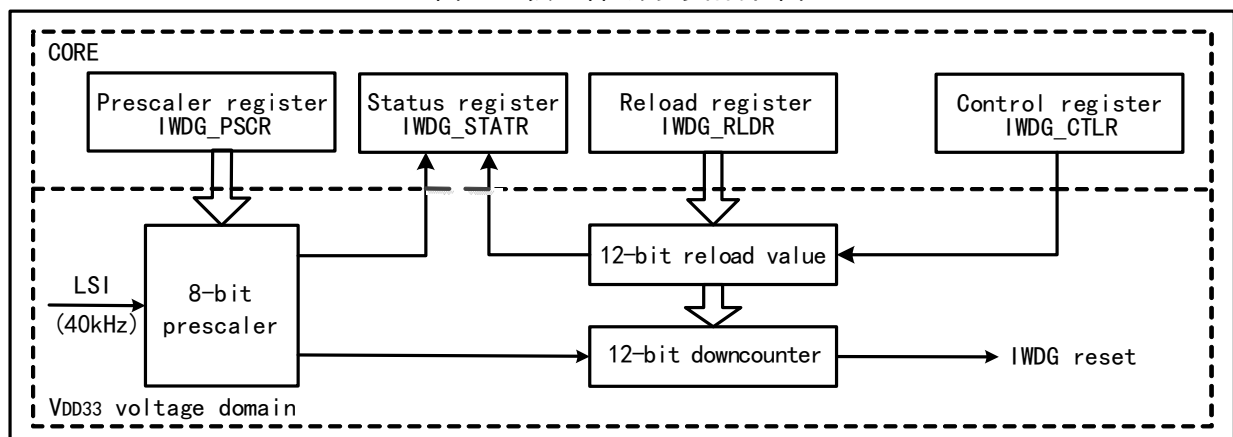
- 12 位自减型计数器
- 时钟来源 LSI 分频，可以在低功耗模式下运行
- 复位条件：计数器值减到 0

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，其功能在停止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 RVU 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	IWDG 控制寄存器	0x0000
R16_IWDG_PSCR	0x40003004	IWDG 分频因子寄存器	0x0000
R16_IWDG_RLDR	0x40003008	IWDG 重装载值寄存器	0x0FFF
R16_IWDG_STATR	0x4000300C	IWDG 状态寄存器	0x0000

7.3.1 IWDG 控制寄存器 (IWDG_CTLR)

偏移地址：0x00

位	名称	访问	描述	复位值
[15:0]	KEY[15:0]	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 IWDG 分频因子寄存器 (IWDG_PSCR)

偏移地址：0x04

位	名称	访问	描述	复位值
[15:3]	Reserved	RO	保留。	0
[2:0]	PR[2:0]	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。 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。	000b

7.3.3 IWDG 重装载值寄存器 (IWDG_RLDR)

偏移地址：0x08

位	名称	访问	描述	复位值
[15:12]	Reserved	RO	保留。	0

[11:0]	RL[11:0]	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 RVU 位为 0，否则读写此域无效。	0xFFFF

7.3.4 IWDG 状态寄存器（IWDG_STATR）

偏移地址：0x0C

位	名称	访问	描述	复位值
[15:2]	Reserved	R0	保留。	0
1	RVU	R0	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	R0	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)

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

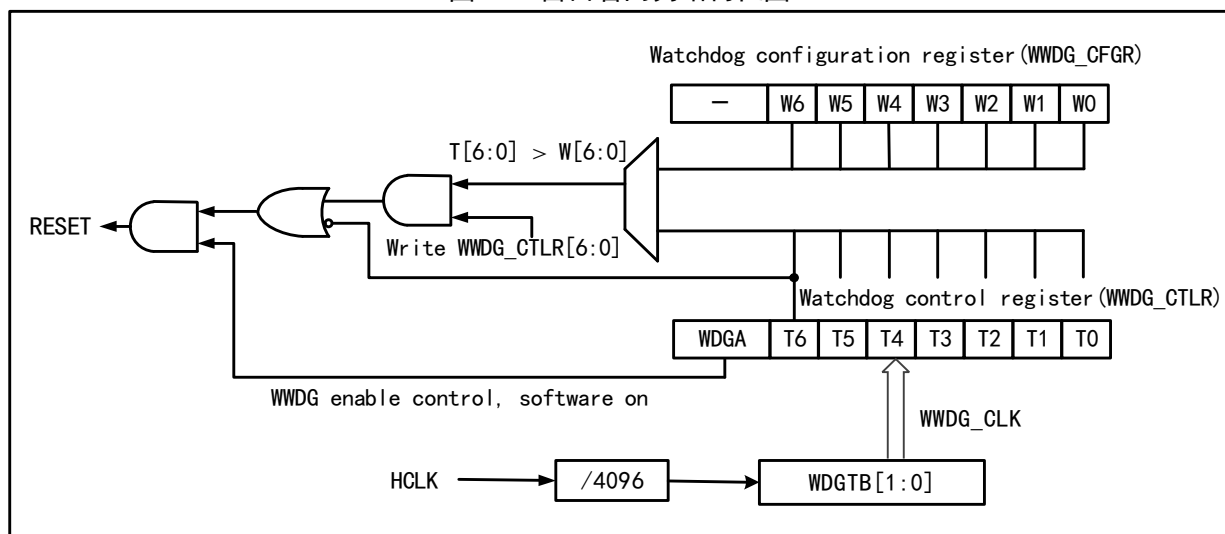
- 可编程的 7 位自减型计数器
- 双条件复位：当前计数器值小于 0x40，或者计数器值在窗口时间外被重装载
- 唤醒提前通知功能（EWI），用于及时喂狗动作防止系统复位

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 HB 总线下，计数时基 WWDG_CLK 来源（HCLK/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_HB1PCENR 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_HB1PRSTR 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：

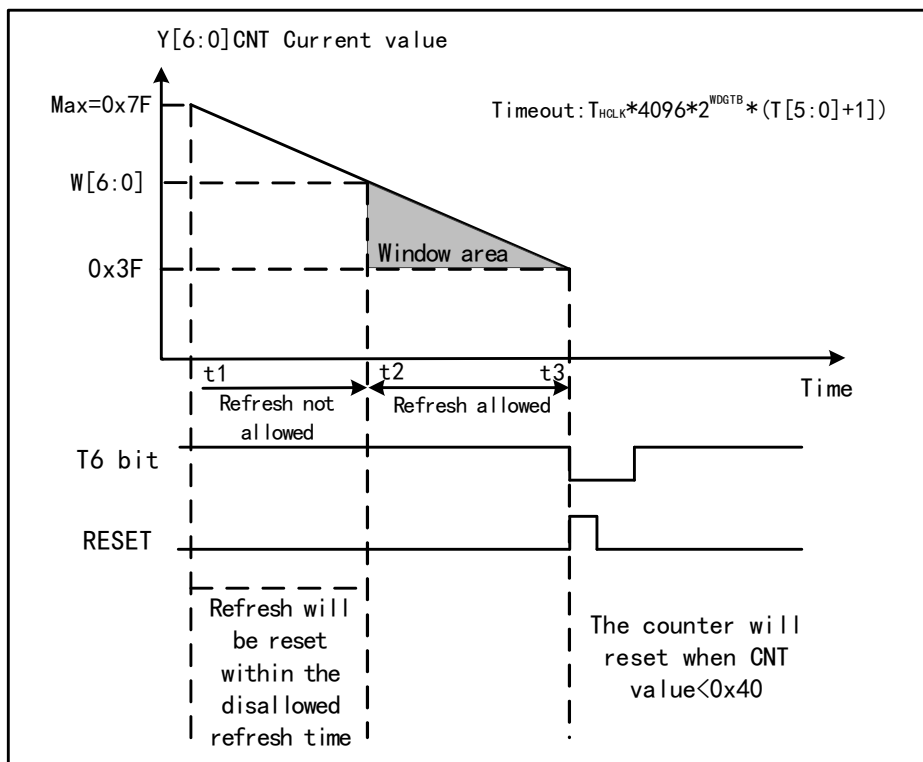
- 1) 计数时基：通过 WWDG_CFGR 寄存器的 WDG TB[1:0]位域，注意要开启 RCC 单元的 WWDG 模块时钟。
- 2) 窗口计数器：设置 WWDG_CFGR 寄存器的 W[6:0]位域，此计数器由硬件用作和当前计数器比较使用，数值由用户软件配置，不会改变。作为窗口时间的上限值。

- 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 t_2 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 t_3 对应当前计数器值达到 0x3F 的时间点。此区域时间内 $t_2 < t < t_3$ 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 $t_1 \leq t \leq t_2$ 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，EWIF 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	WWDG 控制寄存器	0x007F
R16_WWDG_CFGR	0x40002C04	WWDG 配置寄存器	0x007F
R16_WWDG_STATR	0x40002C08	WWDG 状态寄存器	0x0000

8.3.1 WWDG 控制寄存器 (WWDG_CTLR)

偏移地址：0x00

位	名称	访问	描述	复位值
[15:8]	Reserved	R0	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[6:0]	RW	7 位自减计数器，每 $4096 \times 2^{\text{WDGTB}}$ 个 HCLK 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。	0x7F

8.3.2 WWDG 配置寄存器 (WWDG_CFGR)

偏移地址：0x04

位	名称	访问	描述	复位值
[15:10]	Reserved	R0	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清 0。	0
[8:7]	WDGTB[1:0]	RW	窗口看门狗时钟分频选择： 00：1 分频，计数时基 = HCLK/4096； 01：2 分频，计数时基 = HCLK/4096/2； 10：4 分频，计数时基 = HCLK/4096/4； 11：8 分频，计数时基 = HCLK/4096/8。	0
[6:0]	W[6:0]	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。	0x7F

8.3.3 WWDG 状态寄存器 (WWDG_STATR)

偏移地址：0x08

位	名称	访问	描述	复位值
[15:1]	Reserved	WO	保留。	0
0	EWIF	RW0	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 GPIO 及其复用功能（GPIO/AFIO）

芯片内置 6 组 GPIO 端口（PA0~PA15、PB0~PB15、PC0~PC15、PD0~PD15、PE0~PE15、PF0~PF14），共 95 个 GPIO 引脚。多数引脚都可以由软件配置成输出（推挽或开漏）、输入（带或不带上拉或下拉）或复用的外设功能端口。

引脚 PA9~PA12、PC13~PC15、PE3~PE6 由 V_{DD33} 供电，额定 3.3V 供电。其中，在 V_{DD33} 掉电时，引脚 PC13~PC15 自动切换到由 V_{BAT} 供电。

引脚 PA0~PA8、PB2~PB7、PB15、PC4~PC5、PD8、PE2、PF6~PF10 由 V_{DD10} 供电，额定 3.3V 供电，支持 1.8V、2.5V、3.3V 电源。

高速引脚 PA13~PA15、PB0~PB1、PB10~PB14、PC0~PC3、PC6~PC12、PD0~PD7、PD9~PD15、PE0~PE1、PE7~PE15、PF0~PF5、PF11~PF14 由 V_{IO18} 供电，内置 I/O 引脚电压调节器，支持 1.2V、1.8V、2.5V、3.3V 电压并支持动态电源电压切换，支持 X0 引脚配置上电后的默认电压，具体配置信息请参考 PWR_CTLR 寄存器的[12:9]字段。

9.1 主要特征

端口的每个引脚可以配置成以下的多种模式之一：

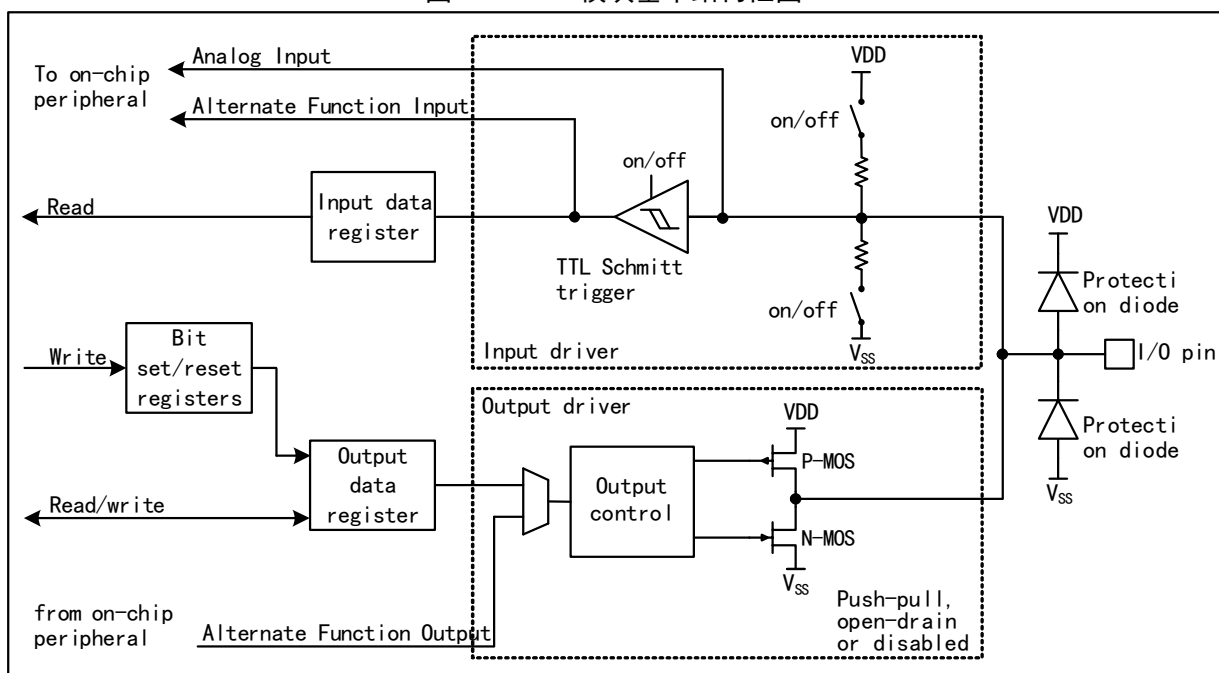
- 浮空输入
- 上拉输入
- 下拉输入
- 模拟输入
- 开漏输出
- 推挽输出
- 复用功能的输入和输出

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。

9.2 功能描述

9.2.1 概述

图 9-1 GPIO 模块基本结构框图



注：VDD 为供电电压，电压类型由具体引脚而定，详情请参考第 9 章。

如图 9-1 所示 I/O 口结构，每个引脚在芯片内部都有两只保护二极管，I/O 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 I/O 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。

9.2.2 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 I/O 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。

9.2.3 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，双方都是一对一的关系。

9.2.4 复用功能复用器和映射

复用功能

I/O 引脚通过一个复用器连接到板载外设/模块，该复用器一次仅允许一个外设的复用功能（AF）连接到 I/O 引脚。这可以确保共用同一个 I/O 引脚的外设之间不会发生冲突。

每个 I/O 引脚都有一个复用器，该复用器采用多达 16 路复用功能输入（AF0 到 AF15），可通过 GPIOx_AFLR 和 GPIOx_AFHR 寄存器对这些输入进行配置：

复位后，复用器选择为复用功能 0（AF0）。在复用模式下通过 GPIOx_CFGLR/GPIOx_CFGHR 寄存器配置 I/O。

CH32H417DS0 手册中详细说明了每个引脚的特定复用功能分配。

复用功能配置

- 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

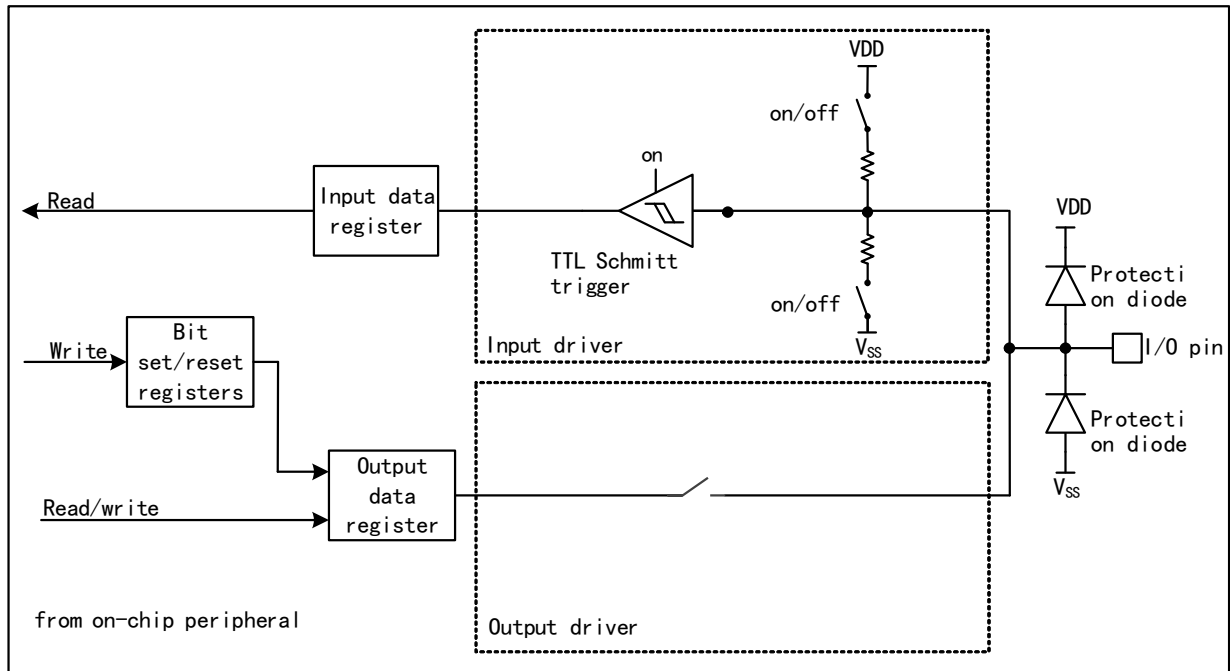
同一个 I/O 口可能有多个外设复用到此管脚，因此为了使各个外设都有最大的发挥空间，外设的复用引脚除了默认复用引脚，还可以进行重映射，重映射到其他的引脚，避开被占用的引脚。

9.2.5 锁定机制

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

9.2.6 输入配置

图 9-2 GPIO 模块输入配置结构框图

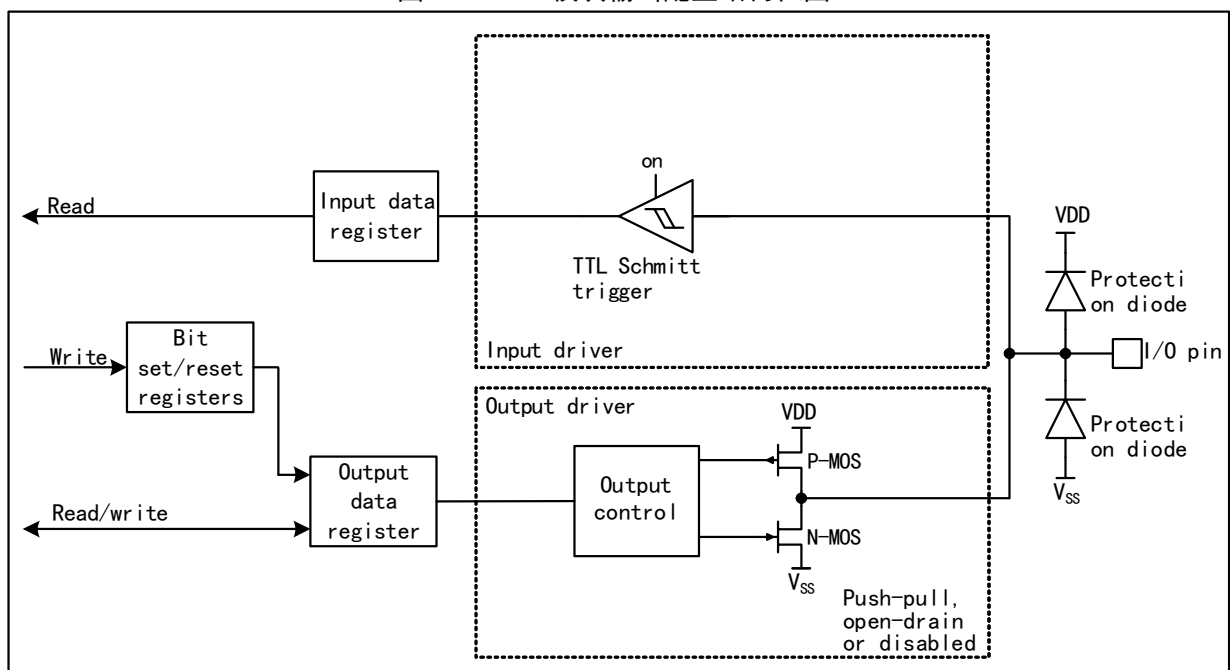


注：VDD 为供电电压，电压类型由具体引脚而定，详情请参考第 9 章。

当 I/O 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 I/O 口上的数据在每个 HB 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。

9.2.7 输出配置

图 9-3 GPIO 模块输出配置结构框图

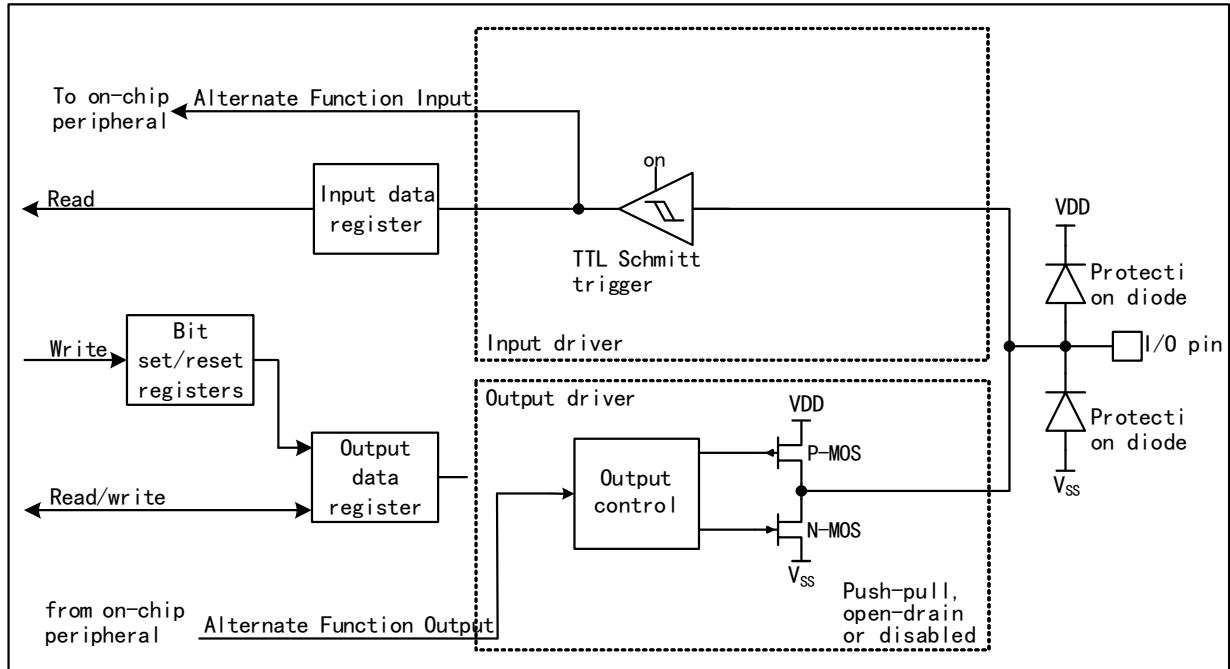


注：VDD 为供电电压，电压类型由具体引脚而定，详情请参考第 9 章。

当 I/O 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 I/O 引脚上的电平将会在每个 HB 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 I/O 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。

9.2.8 复用功能配置

图 9-4 GPIO 模块被其他外设复用时的结构框图

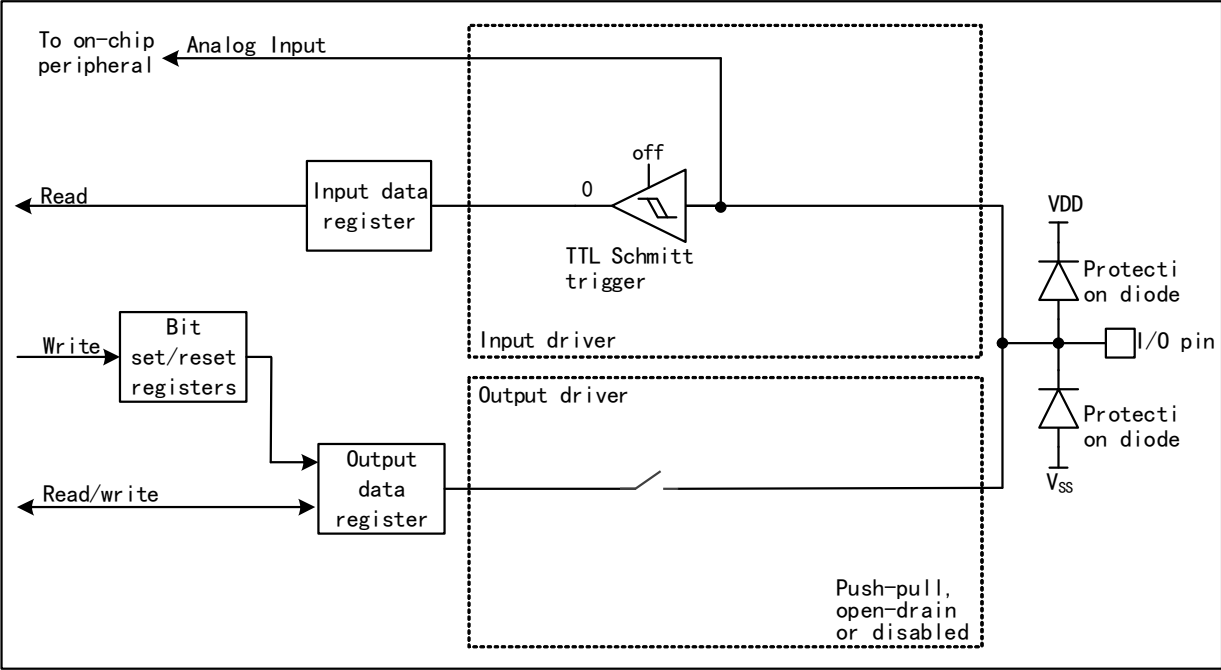


注：VDD 为供电电压，电压类型由具体引脚而定，详情请参考第 9 章。

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 I/O 引脚上的电平将会在每个 HB 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 I/O 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。

9.2.9 模拟输入配置

图 9-5 GPIO 模块作为模拟输入时的配置结构框图



注：VDD 为供电电压，电压类型由具体引脚而定，详情请参考第 9 章。

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。

9.2.10 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 9-1 高级定时器（TIM1/8）

TIMx	配置	GPIO 配置
TIMx_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
TIMx_CHxN	互补输出通道 x	推挽复用输出
TIMx_BKIN	刹车输入	浮空输入
TIMx_BKIN2	刹车输入	浮空输入
TIMx_ETR	外部触发时钟输入	浮空输入

表 9-2 通用定时器（TIM2/3/4/5/9/10/11/12）

TIMx 引脚	配置	GPIO 配置
TIMx_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
TIMx_ETR	外部触发时钟输入	浮空输入

表 9-3 低功耗定时器（LPTIM1/2）

LPTIMx 引脚	配置	GPIO 配置
LPTIMx_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

LPTIMx_ETR	外部触发时钟输入	浮空输入
LPTIMx_OC	PWM 输出	推挽复用输出

表 9-4 通用同步异步串行收发器 (USART1/2/3/4/5/6/7/8)

USARTx 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	开漏复用输出
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 9-5 串行外设接口 (SPI1/2/3/4) 模块

SPIx 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/NSS 输出使能	推挽复用输出
	软件模式	未使用

表 9-6 内置音频总线 (I2S2/3) 模块

I2Sx 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 9-7 内部集成总线 (I2C1/2/3/4) 模块

I ² Cx 引脚	配置	GPIO 配置
I ² Cx_SCL	I ² C 时钟	开漏复用输出
I ² Cx_SDA	I ² C 数据	开漏复用输出

表 9-8 I³C 总线 (I³C) 模块

I ³ C 引脚	配置	GPIO 配置
I ³ C_SCL	I ³ C 时钟	推挽复用输出
I ³ C_SDA	I ³ C 数据	推挽复用输出

表 9-9 控制器局域网 (CAN1/2/3) 模块

CANx 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 9-10 USB PD 控制器

USBPD 引脚	GPIO 配置
USBPD_CC1/USBPD_CC2	推挽复用输出

表 9-11 USB 2.0 高速主机设备 (USBHS) 控制器

USBHS 引脚	GPIO 配置
USBHS_DM/USBHS_DP	使能了 USB 模块之后, 复用 I/O 口会自动连接到内部 USBHS 收发器

表 9-12 USB 主机设备 (USBFS) 控制器

USBFS 引脚	GPIO 配置
USBFS_DM/USBFS_DP	使能了 USB 模块之后, 复用 I/O 口会自动连接到内部 USBFS 收发器

表 9-13 USB OTG_FS 控制器

USB OTG_FS 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 9-14 SDIO 控制器模块

SDIO 引脚	配置	GPIO 配置
CK	时钟	主模式: 推挽复用输出 从模式: 浮空输入
CMD	命令	推挽复用输出
D[7:0]	数据	推挽复用输出

表 9-15 SD/EMMC 控制器模块

SD/EMMC 引脚	配置	GPIO 配置
CK	时钟	主模式: 推挽复用输出 从模式: 推挽复用输出
CMD	命令	推挽复用输出

D[7:0]	数据	推挽复用输出
STR	返回时钟信号，用于 HS400 模式	推挽复用输出

表 9-16 数字滤波器，用于 $\Sigma \Delta$ 调制器（DFSDM）模块

DFSDM 引脚	配置	GPIO 配置
DFSDM_DATINx	串行数据输入	浮空输入
DFSDM_CKINx	串行时钟输入	浮空输入
DFSDM_CKOUT	内部时钟输出	推挽复用输出

表 9-17 串并互转控制器及收发器（SerDes）模块

SerDes 引脚	配置	GPIO 配置
SerDes_TXP	差分发送	浮空输入
SerDes_TXN		浮空输入
SerDes_RXP	差分接收	浮空输入
SerDes_RXN		浮空输入

表 9-18 灵活存储（FMC/SDRAM）控制器

FMC 引脚	GPIO 配置
FMC_A[25:0] FMC_D[31:0]	推挽复用输出
FMC_CK	推挽复用输出
FMC_NOE FMC_NWE	推挽复用输出
FMC_NE1 FMC_NCE2	推挽复用输出
FMC_NWAIT	浮空输入或带上拉输入
FMC_NBL[1:0]	推挽复用输出
FMC_SDCKE[1:0]	推挽复用输出
FMC_SDNE[1:0]	推挽复用输出
FMC_BA[1:0]	推挽复用输出
FMC_NRAS	推挽复用输出
FMC_NCAS	推挽复用输出
FMC_SDNWE	推挽复用输出
FMC_DQM[3:0]	推挽复用输出

表 9-19 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 9-20 高速模拟转数字转换器（HSADC）

HSADC 引脚	GPIO 配置
HSADC	模拟输入

表 9-21 数字图像接口（DVP）模块

DVP 引脚	配置	GPIO 配置
PCLK	像素时钟输入	浮空输入或带上拉输入
DATA[11:0]	像素数据输入	浮空输入或带上拉输入
VSNC	帧同步信号输入	浮空输入或带上拉输入
HSNC	行同步信号输入	浮空输入或带上拉输入

表 9-22 串行音频接口（SAI）模块

SAI 引脚	配置	GPIO 配置
SAI_SD	SD 数据线输入输出	复用推挽输出
SAI_SCK	位时钟输入输出	复用推挽输出
SAI_FS	FS 信号输入输出	复用推挽输出
SAI_MCLK	外部解码器的主时钟输出	复用推挽输出

表 9-23 QSPI 接口（QSPI）模块

QSPI 引脚	GPIO 配置
QSPI_SCK	复用推挽输出
QSPI_SCSN	复用推挽输出
QSPI_SIO[3:0]	复用推挽输出
QSPI_SCSXN	复用推挽输出
QSPI_SIOX[3:0]	复用推挽输出

表 9-24 单线协议主接口（SWPMI）模块

SWPMI 引脚	配置	GPIO 配置
SWP_RX	（非单线模式）接收信号输入	浮空输入
SWP_TX	（非单线模式）发送信号输出	复用推挽输出
SWP_SUP	（非单线模式）挂起信号输出	复用推挽输出
SWPMI_IO	（单线模式）输入输出	复用推挽输出

表 9-25 LCD-TFT 显示控制器（LTDC）模块

LTDC 引脚	配置	GPIO 配置
LCD_CLK	时钟输出	复用推挽输出
LCD_VSYNC	垂直同步输出	复用推挽输出
LCD_HSYNC	水平同步输出	复用推挽输出
LCD_DE	非数据使能输出	复用推挽输出
LCD_R[7:0]	数据：8 位红色数据输出	复用推挽输出
LCD_G[7:0]	数据：8 位绿色数据输出	复用推挽输出
LCD_B[7:0]	数据：8 位蓝色数据输出	复用推挽输出

注：必须通过用户程序配置 LCD-TFT 控制器引脚。未使用的引脚可用于其他功能。

对于高达 24 位（RGB888）的 LTDC 输出，如果使用低于 8bpp 的像素深度将 RGB565 或 RGB666 输出到 16 位或 18 位显示器，则 RGB 显示数据线必须连接到 LCD-TFT 控制器 RGB 数据线的 MSB。例如，当 LCD-TFT 控制器与 RGB565 16 位显示器相连时，LCD 显示器的 R[4:0]、G[5:0]、B[4:0] 数据线引脚必须至 LCD-TFT 控制器的 LCD_R[7:3]、LCD_G[7:2]、LCD_B[7:3]。

表 9-26 其他的 IO 功能设置

引脚	配置功能	GPIO 配置
(PC13) RTC	RTC 输出	硬件自动设置
MC0	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

9.2.11 复用功能重映射 GPIO 设置

9.2.11.1 OSC32_IN/OSC32_OUT 作为 GPIO 端口 PC14/PC15

当 LSEON=0 时, LSE 振荡器引脚 OSC32_IN/OSC32_OUT 可以分别用做 GPIO 的 PC14/PC15。

当 LSEON=1 时, 作为 LSE 引脚。

9.2.11.2 XI/XO 引脚作为 GPIO 端口 PD0/PD1

XI/XO 可以用做 GPIO 的 PD0/PD1, 通过设置重映射寄存器 1 (AFIO_PCFR1) 实现。

9.2.11.3 ADC 复用功能重映射

表 9-27 ADC1 外部触发注入转换复用功能重映射

复用功能	ADC1_ETRGINJ_RM=0 默认映射	ADC1_ETRGINJ_RM=1 重映射
ADC1 外部触发注入转换	ADC1 外部触发注入转换与 EXTI15 相连	ADC1 外部触发注入转换与 TIM8_CH4 相连

表 9-28 ADC1 外部触发规则转换复用功能重映射

复用功能	ADC1_ETRGREG_RM=0 默认映射	ADC1_ETRGREG_RM=1 重映射
ADC1 外部触发规则转换	ADC1 外部触发规则转换与 EXTI11 相连	ADC1 外部触发规则转换与 TIM8_TRGO 相连

表 9-29 ADC2 外部触发注入转换复用功能重映射

复用功能	ADC2_ETRGINJ_RM=0 默认映射	ADC2_ETRGINJ_RM=1 重映射
ADC2 外部触发注入转换	ADC2 外部触发注入转换与 EXTI15 相连	ADC2 外部触发注入转换与 TIM8_CH4 相连

表 9-30 ADC2 外部触发规则转换复用功能重映射

复用功能	ADC2_ETRGREG_RM=0 默认映射	ADC2_ETRGREG_RM=1 重映射
ADC2 外部触发规则转换	ADC2 外部触发规则转换与 EXTI11 相连	ADC2 外部触发规则转换与 TIM8_TRGO 相连

9.2.11.4 TIM2 内部触发 1 复用功能重映射

表 9-31 TIM2 ITR1 复用功能重映射

复用功能	TIM2_ITR1_RM=0	TIM2_ITR1_RM=1
TIM2 内部触发 1	在内部连接 RIM2_ITR1	在内部连接 RIM2_ITR1

	至以太网的 PTP 输出	至全速 USB OTG 的 SOF 输出

9.2.11.5 SDMMC 复用功能重映射

表 9-32 SDMMC 复用功能重映射

复用功能	SDMMC_RM=00 默认映射	SDMMC_RM=01 重映射	SDMMC_RM=1x 重映射
STS/CMD	PD2	PD12	PC10
SDCK/SLVCK	PC12	PD11	PC12
STR	PD3	PD10	PC11
D0	PC8	PB13	PD0
D1	PC9	PC9	PD1
D2	PC10	PB10	PD2
D3	PC11	PB11	PD3
D4	PA14	PA14	PD4
D5	PA15	PA15	PD5
D6	PC6	PC6	PD6
D7	PC7	PC7	PD7

9.2.11.6 UHSIF 复用功能重映射

表 9-33 UHSIF PORT 复用功能重映射

复用功能	UHSIF_PORT_RM=00 默认映射	UHSIF_PORT_RM=01 重映射	UHSIF_PORT_RM=1x 重映射
PORT0	PF12	PF12	PC1
PORT1	PF13	PF13	PC2
PORT2	PE7	PE7	PC3
PORT3	PE8	PC1	PB0
PORT4	PE9	PC2	PB1
PORT5	PE10	PC3	PE10
PORT6	PE11	PB0	PE11
PORT7	PE12	PB1	PE12
PORT8	PE13		
PORT9	PE14		
PORT10	PE15		
PORT11	PB10		
PORT12	PB11		
PORT13	PB12		
PORT14	PB13		
PORT15	PB14		
PORT16	PD10		

PORT17	PD11
PORT18	PD12
PORT19	PD13
PORT20	PD14
PORT21	PD15
PORT22	PF0
PORT23	PF1
PORT24	PF2
PORT25	PC6
PORT26	PC7
PORT27	PC8
PORT28	PC9
PORT29	PA13
PORT30	PA14
PORT31	PA15
PORT32	PC10
PORT33	PC11
PORT34	PC12
PORT35	PD0
PORT36	PD1
PORT37	PD2
PORT38	PD3
PORT39	PD4
PORT40	PD5
PORT41	PD6
PORT42	PD7
PORT43	PF3
PORT44	PF4
PORT45	PF5
PORT46	PE0
PORT47	PE1

表 9-34 UHSIF_CLK 复用功能重映射

复用功能	UHSIF_CLK_RM=00 默认映射	UHSIF_CLK_RM=01 重映射	UHSIF_CLK_RM=10 重映射	UHSIF_CLK_RM=11 重映射
UHSIF_CLK	PF11	PC0	PF14	PD9

9.3 寄存器描述

9.3.1 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 10-35 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_GPIOF_CFGLR	0x40011C00	PF 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_GPIOF_CFGHR	0x40011C04	PF 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_GPIOF_INDR	0x40011C08	PF 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_GPIOF_OUTDR	0x40011C0C	PF 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_GPIOF_BSHR	0x40011C10	PF 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_GPIOF_BCR	0x40011C14	PF 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
R32_GPIOF_LCKR	0x40011C18	PF 端口锁定配置寄存器	0x00000000
R32_GPIOA_SPEED	0x4001081C	PA 端口速度寄存器	0x00000000
R32_GPIOB_SPEED	0x40010C1C	PB 端口速度寄存器	0x00000000

R32_GPIOC_SPEED	0x4001101C	PC 端口速度寄存器	0x00000000
R32_GPIOD_SPEED	0x4001141C	PD 端口速度寄存器	0x00000000
R32_GPIOE_SPEED	0x4001181C	PE 端口速度寄存器	0x00000000
R32_GPIOF_SPEED	0x40011C1C	PF 端口速度寄存器	0x00000000

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

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] [23:22] [19:18] [15:14] [11:10] [7:6] [3:2]	CNFy[1:0]	RW	(y=0-7), 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] [5:4] [1:0]	MODEy[1:0]	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; 10: 输出模式; 11: 输出模式。	00b

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

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]	CNFy[1:0]	RW	(y=8-15), 端口 x 的配置位, 通过这些位配	01b

[27:26]			置相应的端口。	
[23:22]			在输入模式时 (MODE=00b) :	
[19:18]			00: 模拟输入模式;	
[15:14]			01: 浮空输入模式;	
[11:10]			10: 带有上下拉模式。	
[7:6]			11: 保留。	
[3:2]			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]			(y=8-15), 端口 x 的模式位, 通过这些位配	
[21:20]			置相应的端口。	
[17:16]			00: 输入模式;	
[13:12]	MODEy[1:0]	RW	01: 输出模式;	00b
[9:8]			10: 输出模式;	
[5:4]			11: 输出模式。	
[1:0]				

9.3.1.3 端口输入数据寄存器 (GPIOx_INDR) (x=A/B/C/D/E/F)

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

9.3.1.4 端口输出数据寄存器 (GPIOx_OUTDR) (x=A/B/C/D/E/F)

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	对于输出模式:	0

			(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。 对于带有上下拉的输入模式： 1：上拉输入； 0：下拉输入。	

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

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 位的形式访问。如果同时设置了 BR 和 BS 位，则 BS 位起作用。	0

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

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

9.3.1.7 锁定配置寄存器 (GPIOx_LCKR) (x=A/B/C/D/E/F)

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。

9.3.1.8 GPIO 端口速度寄存器 (GPIOx_SPEED) (x=A/B/C/D/E/F)

偏移地址：0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPEED15 [1:0]	SPEED14 [1:0]	SPEED13 [1:0]	SPEED12 [1:0]	SPEED11 [1:0]	SPEED10 [1:0]	SPEED9 [1:0]	SPEED8 [1:0]								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPEED7 [1:0]	SPEED6 [1:0]	SPEED5 [1:0]	SPEED4 [1:0]	SPEED3 [1:0]	SPEED2 [1:0]	SPEED1 [1:0]	SPEED0 [1:0]								

位	名称	访问	描述	复位值
[31:0]	SPEEDy[1:0]	RW	(y=0-15)，端口 x 速度选择，通过这些位配置相应的端口。 00: 10MHz; 01: 50MHz; 10: 100MHz; 11: 180MHz。	00b

9.3.2 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 10-36 AFIO 相关寄存器列表

名称	访问地址	描述	复位值
R32_AFIO_PCFR1	0x40010000	重映射寄存器 1	0x00000000
R32_AFIO_GPIOA_AFLR	0x40010004	PA 端口复用功能寄存器低位	0x00000000
R32_AFIO_GPIOA_AFHR	0x40010008	PA 端口复用功能寄存器高位	0x00000000

R32_AFIO_GPIOB_AFLR	0x4001000C	PB 端口复用功能寄存器低位	0x00000000
R32_AFIO_GPIOB_AFHR	0x40010010	PB 端口复用功能寄存器高位	0x00000000
R32_AFIO_GPIOC_AFLR	0x40010014	PC 端口复用功能寄存器低位	0x00000000
R32_AFIO_GPIOC_AFHR	0x40010018	PC 端口复用功能寄存器高位	0x00000000
R32_AFIO_GPIOD_AFLR	0x4001001C	PD 端口复用功能寄存器低位	0x00000000
R32_AFIO_GPIOD_AFHR	0x40010020	PD 端口复用功能寄存器高位	0x00000000
R32_AFIO_GPIOE_AFLR	0x40010024	PE 端口复用功能寄存器低位	0x00000000
R32_AFIO_GPIOE_AFHR	0x40010028	PE 端口复用功能寄存器高位	0x00000000
R32_AFIO_GPIOF_AFLR	0x4001002C	PF 端口复用功能寄存器低位	0x00000000
R32_AFIO_GPIOF_AFHR	0x40010030	PF 端口复用功能寄存器高位	0x00000000
R32_AFIO_EXTICR1	0x4001003C	外部中断配置寄存器 1	0x00000000
R32_AFIO_EXTICR2	0x40010040	外部中断配置寄存器 2	0x00000000

9.3.2.1 重映射寄存器 1 (AFIO_PCFR1)

偏移地址: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					SW_CFG[2:0]			Reserved			USBPD _CC_H _VT	Reser ved	VDD33 _IO_H _SLV	VDDIO _IO_H _SLV	VI018 _IO_H _SLV
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TIM2IT R1_RM	SDMMC_RM [1:0]		UHSIF_PORT _RM[1:0]		UHSIF_CLK_ RM[1:0]		Reser ved	ADC2_ ETRGI NJ_RM	ADC2_ ETRGR EG_RM	ADC1_ ETRGI NJ_RM	ADC1_ ETRGR EG_RM	PDOPD 1_RM

位	名称	访问	描述	复位值
[31:27]	Reserved	RO	保留。	0
[26:24]	SW_CFG[2:0]	RW	这些位用以配置 SW 功能和跟踪功能的 IO 口。SWD (SDI) 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 (SDI) ; 100: 关闭 SWD (SDI) , 作为 GPIO 功能; 其他: 无效。	000b
[23:21]	Reserved	RO	保留。	0
20	USBPD_CC_HVT	RW	CC 引脚输入通道阈值调节。 USBPD 的 CC 引脚高阈值输入模式使能: 1: 高阈值输入, 可降低 PD 通讯时的功耗; 0: 正常 GPIO 阈值输入。	0
19	Reserved	RO	保留。	0
18	VDD33_IO_HSLV	RW	VDD33 IO 在低电压下的速度配置: 1: IO 电压小于 2.7V 时, IO 速度可增强; 0: IO 工作在低压时, 速度不会被加强。	0
17	VDDIO_IO_HSLV	RW	VDDIO IO 在低电压下的速度配置: 1: IO 电压小于 2.7V 时, IO 速度可增强; 0: IO 工作在低压时, 速度不会被加强。	0

16	VI018_IO_HSLV	RW	VI018 IO 在低电压下的速度配置： 1: IO 电压小于 2.7V 时，IO 速度可增强； 0: IO 工作在低压时，速度不会被加强。	0
[15:13]	Reserved	RO	保留。	0
[12]	TIM2_ITR1_RM	RW	TIM2 内部触发 1 重映射。 1: 在内部连接 TIM2_ITR1 至全速 USB OTG 的 SOF 输出； 0: 在内部连接 TIM2_ITR1 至以太网的 PTP 输出。	0
[11:10]	SDMMC_RM[1:0]	RW	SDMMC 重映射： 00: 默认映射 (STS/CMD/PD2, SDCK/SLVCK/PC12, STR/PD3, D0/PC8, D1/PC9, D2/PC10, D3/PC11, D4/PA14, D5/PA15, D6/PC6, D7/PC7)； 01: 重映射 (STS/CMD/PD12, SDCK/SLVCK/PD11, STR/PD10, D0/PB13, D1/PC9, D2/PB10, D3/PB11, D4/PA14, D5/PA15, D6/PC6, D7/PC7)； 1x: 重映射 (STS/CMD/PC10, SDCK/SLVCK/PC12, STR/PC11, D0/PD0, D1/PD1, D2/PD2, D3/PD3, D4/PD4, D5/PPD5, D6/PD6, D7/PD7)。	00b
[9:8]	UHSIF_PORT_RM[1:0]	RW	UHSIF PORT 重映射： 00: 默认映射 (PORT0/PF12, PORT1/PF13, PORT2/PE7, PORT3/PE8, PORT4/PE9, PORT5/PE10, PORT6/PE11, PORT7/PE12, PORT8/PE13, PORT9/PE14, PORT10/PE15, PORT11/PB10, PORT12/PB11, PORT13/PB12, PORT14/PB13, PORT15/PB14, PORT16/PD10, PORT17/PD11, PORT18/PD12, PORT19/PD13, PORT20/PD14, PORT21/PPD15, PORT22/PF0, PORT23/PF1, PORT24/PF2, PORT25/PC6, PORT26/PC7, PORT27/PC8, PORT28/PC9, PORT29/PA13, PORT30/PA14, PORT31/PA15, PORT32/PC10, PORT33/PC11, PORT34/PC12, PORT35/PD0, PORT36/PD1, PORT37/PD2, PORT38/PD3, PORT39/PD4, PORT40/PD5, PORT41/PD6, PORT42/PD7, PORT43/PF3, PORT44/PF4, PORT45/PF5, PORT46/PE0, PORT47/PE1)； 01: 重映射 (PORT0/PF12, PORT1/PF13, PORT2/PE7, PORT3/PC1, PORT4/PC2, PORT5/PC3, PORT6/PB0, PORT7/PB1, PORT8/PE13, PORT9/PE14, PORT10/PE15, PORT11/PB10, PORT12/PB11, PORT13/PB12, PORT14/PB13, PORT15/PB14, PORT16/PD10, PORT17/PD11, PORT18/PD12, PORT19/PD13, PORT20/PD14, PORT21/PPD15, PORT22/PF0, PORT23/PF1, PORT24/PF2, PORT25/PC6, PORT26/PC7, PORT27/PC8, PORT28/PC9, PORT29/PA13, PORT30/PA14, PORT31/PA15, PORT32/PC10, PORT33/PC11,	00b

			PORT34/PC12 , PORT35/PD0 , PORT36/PD1 , PORT37/PD2 , PORT38/PD3 , PORT39/PD4 , PORT40/PD5 , PORT41/PD6 , PORT42/PD7 , PORT43/PF3 , PORT44/PF4 , PORT45/PF5 , PORT46/PE0, PORT47/PE1) ; 1x: 重映射 (PORT0/PC1, PORT1/PC2, PORT2/PC3, PORT3/PB0, PORT4/PB1, PORT5/PE10, PORT6/PE11, PORT7/PE12 , PORT8/PE13 , PORT9/PE14 , PORT10/PE15 , PORT11/PB10 , PORT12/PB11 , PORT13/PB12 , PORT14/PB13 , PORT15/PB14 , PORT16/PD10 , PORT17/PD11 , PORT18/PD12 , PORT19/PD13 , PORT20/PD14 , PORT21/PPD15 , PORT22/PF0 , PORT23/PF1 , PORT24/PF2 , PORT25/PC6 , PORT26/PC7 , PORT27/PC8 , PORT28/PC9 , PORT29/PA13 , PORT30/PA14 , PORT31/PA15 , PORT32/PC10 , PORT33/PC11 , PORT34/PC12 , PORT35/PD0 , PORT36/PD1 , PORT37/PD2 , PORT38/PD3 , PORT39/PD4 , PORT40/PD5 , PORT41/PD6 , PORT42/PD7 , PORT43/PF3 , PORT44/PF4 , PORT45/PF5 , PORT46/PE0, PORT47/PE1) 。 注: 封装为 56/68 引脚的芯片建议使用 01b 的映 射配置; 封装为 88 引脚的芯片建议使用 1xb 的映 射配置。	
[7:6]	UHSIF_CLK_RM[1:0]	RW	UHSIF_CLK 重映射: 00: 默认映射 (CLK/PF11) ; 01: 重映射 (CLK/PC0) ; 10: 重映射 (CLK/PF14) ; 11: 重映射 (CLK/PD9) 。	00b
5	Reserved	RO	保留。	0
4	ADC2_ETRGINJ_RM	RW	ADC2 外部触发注入转换的重映射位。 1: ADC2 外部触发注入转换与 TIM8_CH4 相连; 0: ADC2 外部触发注入转换与 EXTI15 相连。	0
3	ADC2_ETRGREG_RM	RW	ADC2 外部触发规则转换的重映射位。 1: ADC2 外部触发规则转换与 TIM8_TRG0 相连; 0: ADC2 外部触发规则转换与 EXTI11 相连。	0
2	ADC1_ETRGINJ_RM	RW	ADC1 外部触发注入转换的重映射位。 1: ADC1 外部触发注入转换与 TIM8_CH4 相连; 0: ADC1 外部触发注入转换与 EXTI15 相连。	0
1	ADC1_ETRGREG_RM	RW	ADC1 外部触发规则转换的重映射位。 1: ADC1 外部触发规则转换与 TIM8_TRG0 相连; 0: ADC1 外部触发规则转换与 EXTI11 相连。	0
0	PD0PD1_RM	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 XI&X0。 1: 引脚作为 GPIO 口使用;	0

			0: 引脚作为晶振引脚使用。	

9.3.3.2 端口复用功能寄存器低位 (GPIOx_AFLR) (x=A/B/C/D/E/F)

偏移地址: 0x04、0x0C、0x14、0x1C、0x24、0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			

位	名称	访问	描述	复位值
[31:28]	AFRy[3:0]	RW	(y=0-7), 端口 x 引脚 y 的复用功能选择:	0
[27:24]			0000: AF0; 0001: AF1;	
[23:20]			0010: AF2; 0011: AF3;	
[19:16]			0100: AF4; 0101: AF5;	
[15:12]			0110: AF6; 0111: AF7;	
[11:8]			1000: AF8; 1001: AF9;	
[7:4]			1010: AF10; 1011: AF11;	
[3:0]			1100: AF12; 1101: AF13;	
			1110: AF14; 1111: AF15。	

9.3.3.3 端口复用功能寄存器高位 (GPIOx_AFHR) (x=A/B/C/D/E/F)

偏移地址: 0x08、0x10、0x18、0x20、0x28、0x30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			

位	名称	访问	描述	复位值
[31:28]	AFRy[3:0]	RW	(y=8-15), 端口 x 引脚 y 的复用功能选择:	0
[27:24]			0000: AF0; 0001: AF1;	
[23:20]			0010: AF2; 0011: AF3;	
[19:16]			0100: AF4; 0101: AF5;	
[15:12]			0110: AF6; 0111: AF7;	
[11:8]			1000: AF8; 1001: AF9;	
[7:4]			1010: AF10; 1011: AF11;	
[3:0]			1100: AF12; 1101: AF13;	
			1110: AF14; 1111: AF15。	

9.3.2.4 外部中断配置寄存器 1 (AFIO_EXTICR1)

偏移地址: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			

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
[31:28] [27:24] [23:20] [19:16] [15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	(x=0-7), 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; 0101: PF 引脚的第 x 个引脚; 0110: CMP 输出; 其他: 保留。	0000b

9.3.2.5 外部中断配置寄存器 2 (AFIO_EXTICR2)

偏移地址: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
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
[31:28] [27:24] [23:20] [19:16] [15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	(x=8-15), 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; 0100: PE 引脚的第 x 个引脚; 0101: PF 引脚的第 x 个引脚; 0110: CMP 输出; 其他: 保留。	0000b

第 10 章 直接存储器访问控制（DMA）

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。

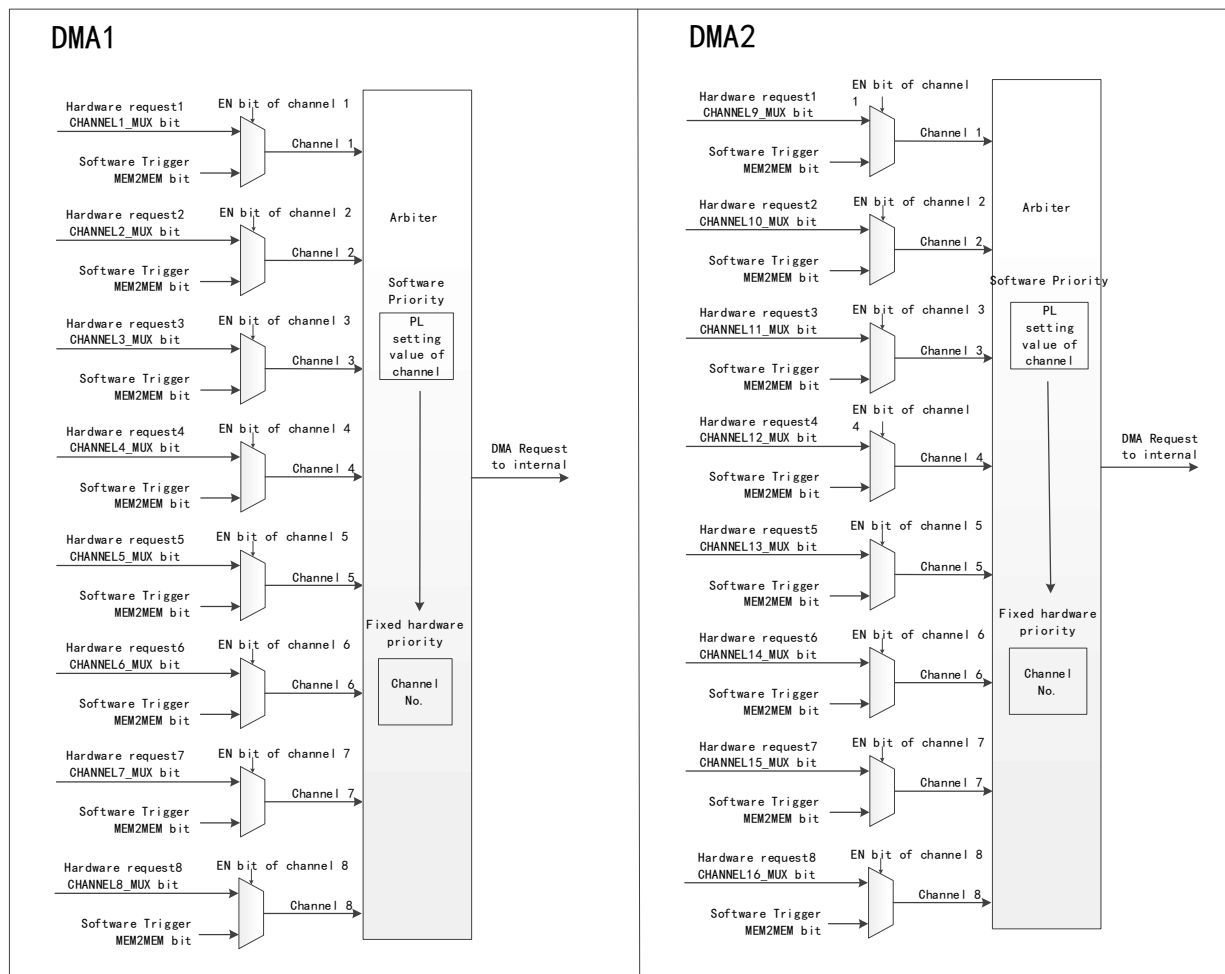
DMA 控制器支持外设与 DMA 之间重新配置 DMA 请求线。还有一个仲裁器来协调各通道之间的优先级。

10.1 主要特性

- 多个独立可配置通道
- 支持外设与 DMA 之间重新配置 DMA 请求线，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、HB 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

10.2 功能描述

图 10-1 DMA 通道工作示意图



10.2.1 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) 通过 FLAG_CUR_MEM 位选择存储器地址, 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_CFGRx 寄存器的 EN 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_CFGRx 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_CFGRx 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_CFGRx 寄存器的 EN 位，该通道被关闭。如果在 DMA_CFGRx 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 DMA 事件内容。

10.2.2 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 4 种选择：8 位、16 位、32 位、256 位。具体数据转移方式如下表：

表 10-1 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 0x01/B1 0x02/B2 0x03/B3	0x00/B0 0x01/B1 0x02/B2 0x03/B3	● 源端地址递增量与源端设置的数据位宽对齐，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	
8	256	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/00...0000B0 0x20/00...0000B1 0x40/00...0000B2 0x60/00...0000B3	
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	
16	256	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/00...00B1B0 0x20/00...00B3B2 0x40/00...00B5B4 0x60/00...00B7B6	
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	
32	256	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/00...B3B2B1B0 0x20/00...B7B6B5B4 0x40/00...BBBAB9B8 0x60/00...BFBEBDBC	
256	8	4	0x00/2F...13121110 0x20/4F...33323130 0x40/6F...53525150 0x60/8F...73727170	0x00/10 0x01/30 0x02/50 0x03/70	
256	16	4	0x00/2F...13121110 0x20/4F...33323130 0x40/6F...53525150 0x60/8F...73727170	0x00/1110 0x02/3130 0x04/5150 0x06/7170	
256	32	4	0x00/2F...13121110 0x20/4F...33323130 0x40/6F...53525150 0x60/8F...73727170	0x00/13121110 0x04/33323130 0x08/53525150 0x0C/73727170	
256	256	4	0x00/2F...13121110 0x20/4F...33323130 0x40/6F...53525150 0x60/8F...73727170	0x00/2F...13121110 0x20/4F...33323130 0x40/6F...53525150 0x60/8F...73727170	

10.2.3 DMA 请求复用器

DMAMUX 请求复用器可在芯片的外设和 DMA 控制器之间重新配置（路由）DMA 请求线。该路由功能通过可编程的多通道 DMA 请求线复用器来确保实现。每条通道可不受限制地选择一个 DMA 请求线，或者按照与来自其 DMAMUX 同步输入的事件相同步的方式选择一个 DMA 请求线。

DMA 控制器提供 16 个通道，其中 DMAMUX 通道 1 到 8 与 DMA1 通道 1 到 8 相连，DMAMUX 通道 9 到

16 与 DMA2 通道 1 到 8 相连。

表 10-2 DMA 复用器输入到资源的分配表

DMA 请求输入	外设	DMA 请求输入	外设	DMA 请求输入	外设
1	TIM1_CH1	42	TIM9_CH4	83	I3C_TX
2	TIM1_CH2	43	TIM9_UP	84	I3X_RX
3	TIM1_CH3	44	TIM9_TRIG	85	USART1_TX
4	TIM1_CH4	45	TIM10_CH1	86	USART1_RX
5	TIM1_UP	46	TIM10_CH2	87	USART2_TX
6	TIM1_COM	47	TIM10_CH3	88	USART2_RX
7	TIM1_TRIG	48	TIM10_CH4	89	USART3_TX
8	TIM2_CH1	49	TIM10_UP	90	USART3_RX
9	TIM2_CH2	50	TIM10_TRIG	91	USART4_TX
10	TIM2_CH3	51	TIM11_CH1	92	USART4_RX
11	TIM2_CH4	52	TIM11_CH2	93	USART5_TX
12	TIM2_UP	53	TIM11_CH3	94	USART5_RX
13	TIM2_TRIG	54	TIM11_CH4	95	USART6_TX
14	TIM3_CH1	55	TIM11_UP	96	USART6_RX
15	TIM3_CH2	56	TIM11_TRIG	97	USART7_TX
16	TIM3_CH3	57	TIM12_CH1	98	USART7_RX
17	TIM3_CH4	58	TIM12_CH2	99	USART8_TX
18	TIM3_UP	59	TIM12_CH3	100	USART8_RX
19	TIM3_TRIG	60	TIM12_CH4	101	SWPMI_TX
20	TIM4_CH1	61	TIM12_UP	102	SWPMI_RX
21	TIM4_CH2	62	TIM12_TRIG	103	DAC1
22	TIM4_CH3	63	SPI1_TX	104	DAC2
23	TIM4_CH4	64	SPI1_RX	105	保留
24	TIM4_UP	65	SPI2_TX	106	保留
25	TIM4_TRIG	66	SPI2_RX	107	DFSDM_DMA0
26	TIM5_CH1	67	SPI3_TX	108	DFSDM_DMA1
27	TIM5_CH2	68	SPI3_RX	109	保留
28	TIM5_CH3	69	SPI4_TX	110	保留
29	TIM5_CH4	70	SPI4_RX	111	SDIO
30	TIM5_UP	71	QSPI1_DMA	112	SAI_A_TX
31	TIM5_TRIG	72	QSPI2_DMA	113	SAI_A_RX
32	TIM8_CH1	73	I2C1_TX	114	SAI_B_TX
33	TIM8_CH2	74	I2C1_RX	115	SAI_B_RX
34	TIM8_CH3	75	I2C2_TX	116	保留
35	TIM8_CH4	76	I2C2_RX	117	保留
36	TIM8_UP	77	I2C3_TX	118	保留
37	TIM8_COM	78	I2C3_RX	119	保留
38	TIM8_TRIG	79	I2C4_TX	120	ADC1
39	TIM9_CH1	80	I2C4_RX	121	ADC2
40	TIM9_CH2	81	I3C_RS	122	TIM6_UP
41	TIM9_CH3	82	I3C_TC	123	TIM7_UP

10.2.4 双缓冲模式

双缓冲模式用于 DMA1 和 DMA2 的通道，双缓冲模式除了有两个存储器指针外，其通道的工作方式仍与单缓冲模式的一样。

通过设置 DMA_CFGRx 寄存器的 DOUBLE_MODE 位置 1，使能双缓冲模式，自动进入循环模式，并在完成一次循环后自动切换存储器地址（DMA_MADDRx 和 DMA_M1ADDRx 间交替切换）。

每次循环结束时 DMA 控制器都从一个存储器目标交换为另一个存储器目标，这样，软件在处理一个存储器区域的同时，DMA 传输还可以填充或使用第二个存储器区域。双缓冲通道可双向工作，如下表所示。

表 10-3 双缓冲模式下的源和目标地址寄存器

DMA_CFGRx.DIR	方向	源地址	目标地址
0	从外设读	DMA_PADDRx	DMA_MADDRx/DMA_M1ADDRx
1	从存储器读	DMA_MADDRx/DMA_M1ADDRx	DMA_PADDRx

注：使能双缓冲模式时禁止配置存储器到存储器模式。

10.3 寄存器描述

表 10-4 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_M1ADDR1	0x40020018	DMA1 通道 1 存储器地址寄存器 1	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_M1ADDR2	0x4002002C	DMA1 通道 2 存储器地址寄存器 1	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_M1ADDR3	0x40020040	DMA1 通道 3 存储器地址寄存器 1	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_M1ADDR4	0x40020054	DMA1 通道 4 存储器地址寄存器 1	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_M1ADDR5	0x40020068	DMA1 通道 5 存储器地址寄存器 1	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_M1ADDR6	0x4002007C	DMA1 通道 6 存储器地址寄存器 1	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_M1ADDR7	0x40020090	DMA1 通道 7 存储器地址寄存器 1	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
R32_DMA1_M1ADDR8	0x400200A4	DMA1 通道 8 存储器地址寄存器 1	0x00000000

表 10-5 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_M1ADDR1	0x40020418	DMA2 通道 1 存储器地址寄存器 1	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_M1ADDR2	0x4002042C	DMA2 通道 2 存储器地址寄存器 1	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_M1ADDR3	0x40020440	DMA2 通道 3 存储器地址寄存器 1	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_M1ADDR4	0x40020454	DMA2 通道 4 存储器地址寄存器 1	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_M1ADDR5	0x40020468	DMA2 通道 5 存储器地址寄存器 1	0x00000000
R32_DMA2_CFR6	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_M1ADDR6	0x4002047C	DMA2 通道 6 存储器地址寄存器 1	0x00000000
R32_DMA2_CFR7	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_M1ADDR7	0x40020490	DMA2 通道 7 存储器地址寄存器 1	0x00000000
R32_DMA2_CFR8	0x40020494	DMA2 通道 8 配置寄存器	0x00000000
R32_DMA2_CNTR8	0x40020498	DMA2 通道 8 传输数据数目寄存器	0x00000000
R32_DMA2_PADDR8	0x4002049C	DMA2 通道 8 外设地址寄存器	0x00000000
R32_DMA2_MADDR8	0x400204A0	DMA2 通道 8 存储器地址寄存器	0x00000000
R32_DMA2_M1ADDR8	0x400204A4	DMA2 通道 8 存储器地址寄存器 1	0x00000000

表 10-6 DMAMUX 寄存器列表

名称	访问地址	描述	复位值
R32_DMAMUX1_4_CFR	0x40020800	DMA 请求复用器通道 1-4 配置寄存器	0x00000000
R32_DMAMUX5_8_CFR	0x40020804	DMA 请求复用器通道 5-8 配置寄存器	0x00000000
R32_DMAMUX9_12_CFR	0x40020808	DMA 请求复用器通道 9-12 配置寄存器	0x00000000
R32_DMAMUX13_16_CFR	0x4002080C	DMA 请求复用器通道 13-16 配置寄存器	0x00000000

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

偏移地址: $0x00 + (x-1)*0x400$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEIF8	HTIF8	TCIF8	GIF8	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
31/27/23/ 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
30/26/22/ 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
29/25/21/ 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

28/24/20/ 16/12/8/4 /0	GIFx	R0	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

10.3.2 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
CTEIF8	CHTIF8	CTCIF8	CGIF8	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
31/27/23/ 19/15/11/ 7/3	CTEIFx	WO	清除通道 x 的传输错误标志 (x=1/2/3/4/5/6/7/8) : 1: 清除 DMA_INTFR 寄存器中的 TEIFx 标志; 0: 无作用。	0
30/26/22/ 18/14/10/ 6/2	CHTIFx	WO	清除通道 x 的传输过半标志 (x=1/2/3/4/5/6/7/8) : 1: 清除 DMA_INTFR 寄存器中的 HTIFx 标志; 0: 无作用。	0
29/25/21/ 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
28/24/20/ 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

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

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															FLAG_C UR_MEM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOUBLE _MODE	MEM2 MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCI E	EN

位	名称	访问	描述	复位值
[31:17]	Reserved	R0	保留。	0
16	FLAG_CUR_MEM	RW	存储器地址选择设置, 双缓冲模式时硬件会自动切换: 1: M1ADDR (使用 DMA_M1ADDRx 指针寻址); 0: MADDR (使用 DMA_MADDRx 指针寻址)。 注: 该位仅在双缓冲模式下使用, 在通道使能后, 此位 相当于一个状态标志, 用于指示作为当前目标的存储	0

			区。	
15	DOUBLE_MODE	RW	双缓冲模式使能： 1：使能双缓冲模式，会自动进循环模式，完成一次后会切换存储器地址； 0：关闭双缓冲模式。	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[1:0]	RW	通道优先级设置： 00：低； 01：中； 10：高； 11：最高。	00b
[11:10]	MSIZE[1:0]	RW	存储器地址数据宽度设置： 00：8 位； 01：16 位； 10：32 位； 11：256 位。	00b
[9:8]	PSIZE[1:0]	RW	外设地址数据宽度设置： 00：8 位； 01：16 位； 10：32 位； 11：256 位。	00b
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

10.3.4 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=1/2)

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[15:0]	RW	数据传输数量，范围 0~65535。 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：表示当前待传输数目。此寄存器只能在 EN=0 时更改，EN=1 时不可写。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。

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

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[31:0]	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。当 PSIZE[1:0]= '11'（256 位），模块自动忽略 bit[4:0]，操作地址自动 32 字节对齐。	0

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

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[31:0]	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。当 MSIZE[1:0]= '11'（256 位），模块自动忽略 bit[4:0]，操作地址自动 32 字节对齐。	0

10.3.7 DMAy 通道 x 存储器地址寄存器 1 (DMAy_M1ADDRx) (x=1/2/3/4/5/6/7/8, y=1/2)

偏移地址：0x18 + (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
M1A[31:0]																															

位	名称	访问	描述	复位值
[31:0]	M1A[31:0]	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。当 MSIZE[1:0]= '11'（256 位），模块自动忽略 bit[4:0]，操作地址自动 32 字节对齐。	0

10.3.8 DMA 请求复用器通道 1-4 配置寄存器 (DMAMUX1_4_CFGR)

偏移地址：0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CHANNEL4_MUX[6:0]						Reserved	CHANNEL3_MUX[6:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CHANNEL2_MUX[6:0]						Reserved	CHANNEL1_MUX[6:0]							

位	名称	访问	描述	复位值
31	Reserved	RO	保留。	0
[30:24]	CHANNEL4_MUX[6:0]	RW	DMAMUX 复用器通道 4 输入到资源的分配。	0
23	Reserved	RO	保留。	0
[22:16]	CHANNEL3_MUX[6:0]	RW	DMAMUX 复用器通道 3 输入到资源的分配。	0
15	Reserved	RO	保留。	0
[14:8]	CHANNEL2_MUX[6:0]	RW	DMAMUX 复用器通道 2 输入到资源的分配。	0
7	Reserved	RO	保留。	0
[6:0]	CHANNEL1_MUX[6:0]	RW	DMAMUX 复用器通道 1 输入到资源的分配： 0000000: DMA 请求输入 1； 0000001: DMA 请求输入 2； 0000010: DMA 请求输入 3； 1111010: DMA 请求输入 123； 其他：保留。	0

10.3.9 DMA 请求复用器通道 5-8 配置寄存器 (DMAMUX5_8_CFGR)

偏移地址：0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CHANNEL8_MUX[6:0]						Reserved	CHANNEL7_MUX[6:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CHANNEL6_MUX[6:0]						Reserved	CHANNEL5_MUX[6:0]							

ved		ved	

位	名称	访问	描述	复位值
31	Reserved	R0	保留。	0
[30:24]	CHANNEL8_MUX[6:0]	RW	DMAMUX 复用器通道 8 输入到资源的分配。	0
23	Reserved	R0	保留。	0
[22:16]	CHANNEL7_MUX[6:0]	RW	DMAMUX 复用器通道 7 输入到资源的分配。	0
15	Reserved	R0	保留。	0
[14:8]	CHANNEL6_MUX[6:0]	RW	DMAMUX 复用器通道 6 输入到资源的分配。	0
7	Reserved	R0	保留。	0
[6:0]	CHANNEL5_MUX[6:0]	RW	DMAMUX 复用器通道 5 输入到资源的分配： 0000000: DMA 请求输入 1； 0000001: DMA 请求输入 2； 0000010: DMA 请求输入 3； 1111010: DMA 请求输入 123； 其他：保留。	0

10.3.10 DMA 请求复用器通道 9–12 配置寄存器 (DMAMUX9_12_CFGR)

偏移地址：0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CHANNEL12_MUX[6:0]							Reserved	CHANNEL11_MUX[6:0]						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CHANNEL10_MUX[6:0]							Reserved	CHANNEL9_MUX[6:0]						

位	名称	访问	描述	复位值
31	Reserved	R0	保留。	0
[30:24]	CHANNEL12_MUX[6:0]	RW	DMAMUX 复用器通道 12 输入到资源的分配。	0
23	Reserved	R0	保留。	0
[22:16]	CHANNEL11_MUX[6:0]	RW	DMAMUX 复用器通道 11 输入到资源的分配。	0
15	Reserved	R0	保留。	0
[14:8]	CHANNEL10_MUX[6:0]	RW	DMAMUX 复用器通道 10 输入到资源的分配。	0
7	Reserved	R0	保留。	0
[6:0]	CHANNEL9_MUX[6:0]	RW	DMAMUX 复用器通道 9 输入到资源的分配： 0000000: DMA 请求输入 1； 0000001: DMA 请求输入 2； 0000010: DMA 请求输入 3； 1111010: DMA 请求输入 123； 其他：保留。	0

10.3.11 DMA 请求复用器通道 13–16 配置寄存器 (DMAMUX13_16_CFGR)

偏移地址：0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CHANNEL16_MUX[6:0]						Reserved	CHANNEL15_MUX[6:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CHANNEL14_MUX[6:0]						Reserved	CHANNEL13_MUX[6:0]							

位	名称	访问	描述	复位值
31	Reserved	RO	保留。	0
[30:24]	CHANNEL16_MUX[6:0]	RW	DMAMUX 复用器通道 16 输入到资源的分配。	0
23	Reserved	RO	保留。	0
[22:16]	CHANNEL15_MUX[6:0]	RW	DMAMUX 复用器通道 15 输入到资源的分配。	0
15	Reserved	RO	保留。	0
[14:8]	CHANNEL14_MUX[6:0]	RW	DMAMUX 复用器通道 14 输入到资源的分配。	0
7	Reserved	RO	保留。	0
[6:0]	CHANNEL13_MUX[6:0]	RW	DMAMUX 复用器通道 13 输入到资源的分配： 0000000: DMA 请求输入 1； 0000001: DMA 请求输入 2； 0000010: DMA 请求输入 3； 1111010: DMA 请求输入 123； 其他：保留。	0

第 11 章 模拟/数字转换（ADC）

ADC 模块包含 2 个 12 位的模拟数字转换器，最高 80MHz 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。

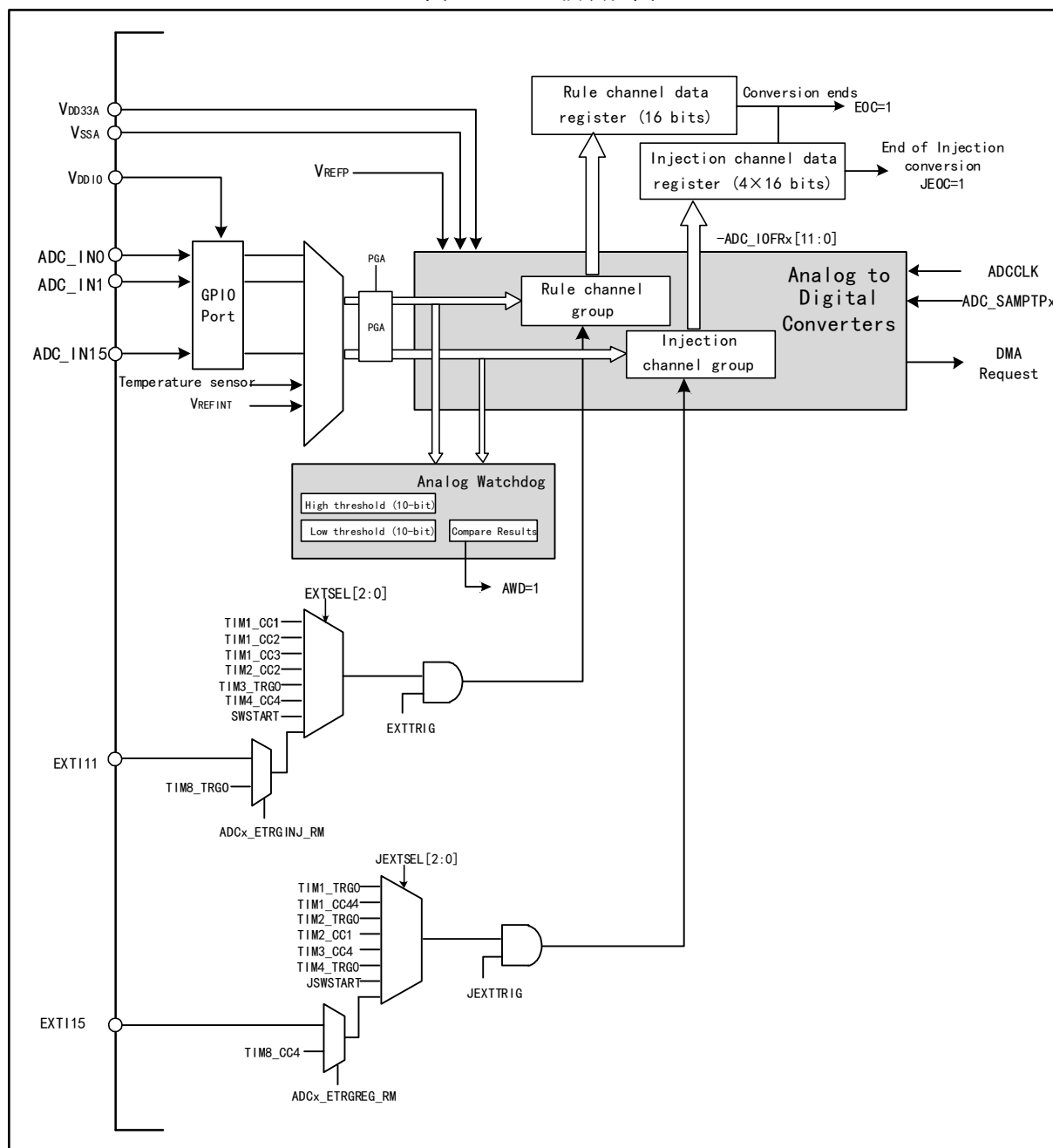
11.1 主要特性

- 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
- 双重模式
- ADC 通道输入范围： $V_{SS} \leq V_{IN} \leq V_{DD10}$
- 输入增益可调，可实现小信号放大采样

11.2 功能描述

11.2.1 模块结构

图 11-1 ADC 模块框图



11.2.2 ADC 配置

1) 模块上电

ADCx_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 HCLK (HB 总线) 时钟, 其转换单元的时钟基准 ADCCLK 与 HCLK 同步, 由

RCC_CFGR0 寄存器的 ADCPRE[1:0]域配置分频，最大不能超过 80MHz。

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 ADCx_RSQRx 寄存器中设置。规则组中转换的总数量应写入 ADCx_RSQR1 寄存器的 L[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 ADCx_ISQR 寄存器中设置。注入组里的转换总数量应写入 ADCx_ISQR 寄存器的 JL[1:0] 中。

注：如果 ADCx_RSQRx 或 ADCx_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

通过写 ADCx_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 ADCx_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。

通过写 ADCx_CTLR2 寄存器的 CAL_VOL[1:0] 位可配置校准电压值；置 CAL_AUTO 位为 1，使能硬件自动校准功能。

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。

注：ADC 在校准期间，采样时间至少要 1us。

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 ADCx_SAMPTR1 和 ADCx_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：

$$T_{\text{CONV}} = \text{采样时间} + 12.5T_{\text{ADCCLK}}$$

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 ADCx_RDATAR 中，为防止连续转换多个规则通道时，没有及时取走 ADCx_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 ADCx_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 ADCx_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

ADCx_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 ADCx_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 ADCx_IDATARx 是实际转换的数据减去 ADCx_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 11-2 数据左对齐

规则组数据寄存器

D11	D10	D9	D8	D7	D6	D5	D4	D3	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 11-3 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

11.2.3 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 ADCx_CTLR2 寄存器的 EXTTRIG 或 JEXTTRIG 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，EXTSEL[2:0] 和 JEXTSEL[2:0] 位的配置决定规则组和注入组的外部事件源。

注：当外部触发信号被选为 ADC 规则或注入转换时，只有它的上升沿可以启动转换。

表 11-1 规则组通道的外部触发源

EXTSEL[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 TRG0 事件	
101	定时器 4 的 CC4 事件	
110	EXTI 线 11/TIM8_TRG0	来自外部引脚/内部定时器信号
111	SWSTART 位置 1 软件触发	软件控制位

表 11-2 注入组通道的外部触发源

JEXTSEL[2:0]	触发源	类型
000	定时器 1 的 TRG0 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 TRG0 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 TRG0 事件	
110	EXTI 线 15/TIM8_CC4	来自外部引脚/内部定时器信号
111	JSWSTART 位置 1 软件触发	软件控制位

11.2.4 转换模式

表 11-3 转换模式组合

ADCx_CTLR1 和 ADCx_CTLR2 寄存器控制位					ADC 转换模式
CONT	SCAN	DISCEN/JDISCEN	JAUTO	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（ADCx_RSQRx）或所有注入组通道（ADCx_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（ADCx_RSQRx）或所有注入组通道（ADCx_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 (DISCEN 和 JDISCEN 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 DISCEN 和 JDISCEN，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。
			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 ADCx_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 ADCx_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 ADCx_IDATAR1 寄存器中，EOC 和 JEOC 标志被置位，如果设置了 JEOCIE 或 EOCIE 位，将触发 ADC 中断。

2) 单次扫描模式转换

通过设置 ADCx_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 ADCx_RSQRx 寄存器（对规则通道）或 ADCx_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 JAUTO 位的状态，又分为触发注入方式和自动注入方式。

● 触发注入

JAUTO 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

JAUTO 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 ADCx_RSQRx 和 ADCx_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（JEXTTRIG=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) 单次中断模式转换

通过设置 ADCx_CTLR1 寄存器的 DISCEN 或 JDISCEN 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 ($n \leq 8$) 定义在 ADCx_CTLR1 寄存器的 DISCNUM[2:0] 中，当 DISCEN 为 1，则是规则组的间断模式，待转换总长度定义在 ADCx_RSQR1 寄存器的 L[3:0] 中；当 JDISCEN 为 1，则是注入组的间断模式，待转换总长度定义在 ADCx_ISQR 寄存器的 JL[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：

DISCEN=1, DISCNUM[2:0]=3, L[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：

JDISCEN=1, JL[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 JE0C 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（JAUTO=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. 注入组的间断模式下，外部触发后要转换的注入通道数目为 1。

4) 连续转换

在连续转换模式中，当前面 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。此模式的启动事件包括外部触发事件和

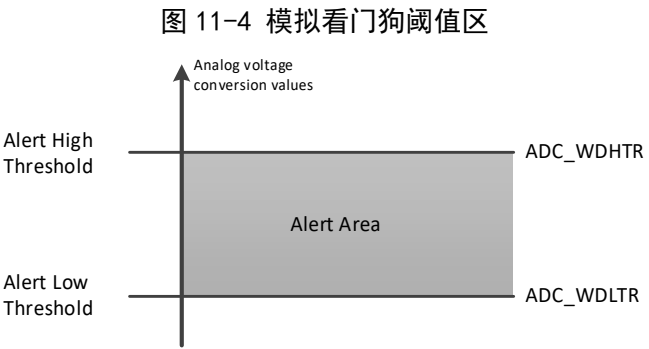
ADON 位置 1，设置启动后，需将 CONT 位置 1。

如果一个规则通道被转换，转换数据被存储于 ADCx_RDATAR 寄存器中，转换结束标志 EOC 被置位，如果设置了 EOCIE，则产生中断。

如果一个注入通道被转换，转换数据被存储于 ADCx_IDATARx 寄存器中，注入转换结束标志 JE0C 被置位，如果设置了 JE0CIE，则产生中断。

11.2.5 模拟看门狗

如果被 ADC 转换的模拟电压低于低阈值或高于高阈值，AWD 模拟看门狗状态位被设置。阈值设置位于 ADCx_WDHTR 和 ADCx_WDLTR 寄存器的最低 12 个有效位中。通过设置 ADCx_CTLR1 寄存器的 AWDIE 位以允许产生相应中断。



配置 ADCx_CTLR1 寄存器的 AWDSGL、AWDEN、JAWDEN 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 11-4 模拟看门狗通道选择

模拟看门狗警戒通道	ADCx_CTLR1 寄存器控制位			
	AWDSGL	AWDEN	JAWDEN	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

11.2.6 温度传感器

芯片内置温度传感器，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芯片内部温度，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，由于制造离散性，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 ADCx_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}-V₂₅）/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 V₂₅ 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。

11.2.7 双 ADC 模式

在有 2 个的 ADC 模块芯片中，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 DUALMOD[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。

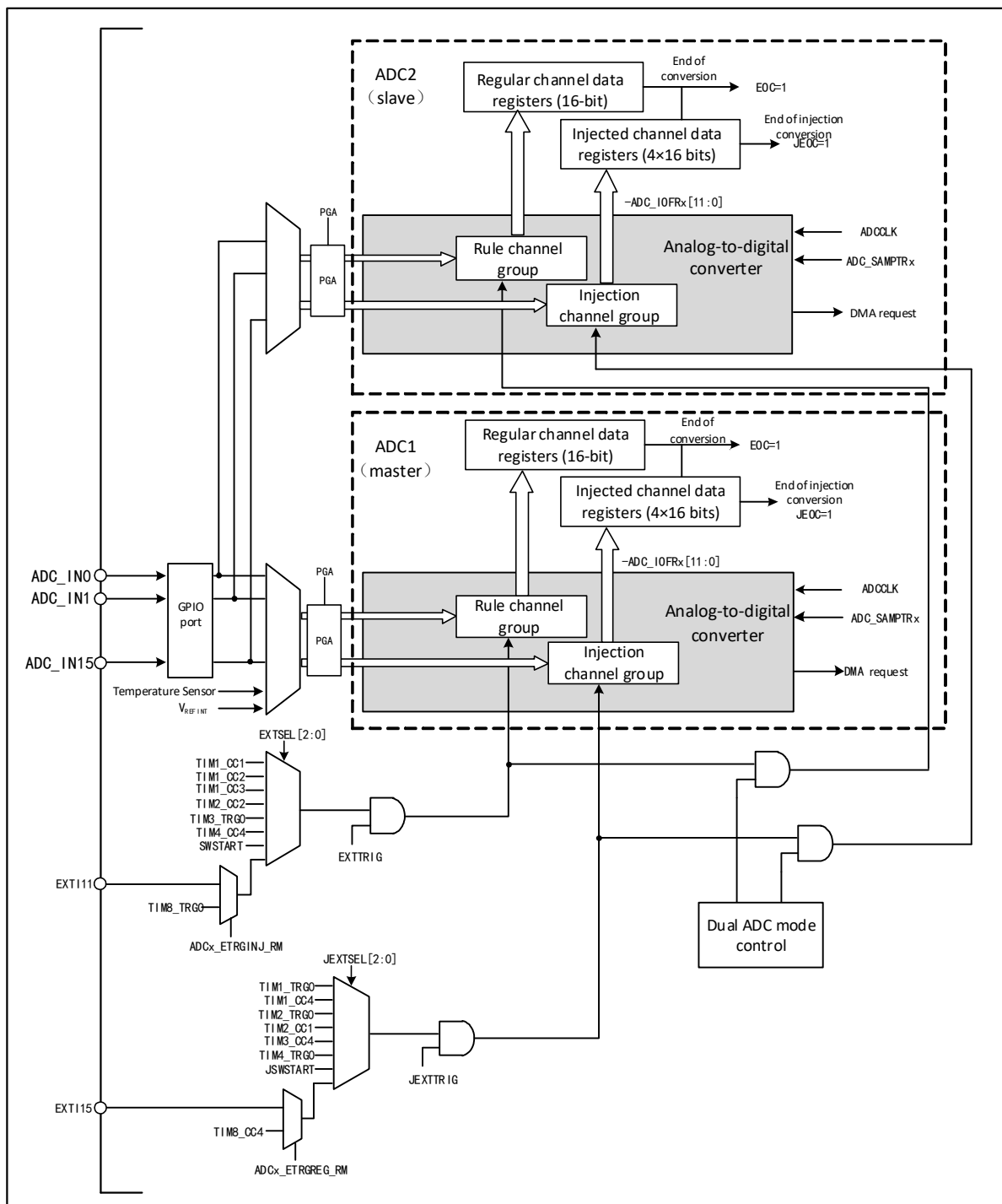
通过配置可以实现以下几种可能的模式：

- 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 11-5 双 ADC 框图



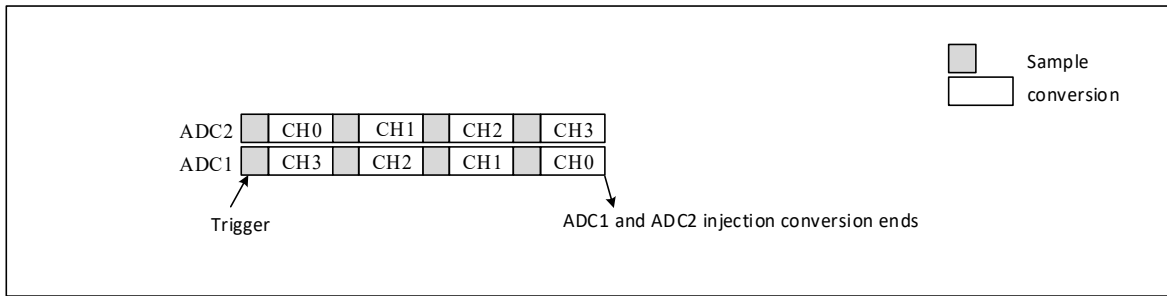
1) 独立模式

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_CTLR2 中 JEXTSEL[2:0]，以选择触发源，同时它也将用于同步触发 ADG2。转换完成时，转换后的数据分别存储在 ADCx 的 ADCx_IDATARx 中，且若使能任一 ADC 中断，转换结束后将产生 JE0C 中断。

图 11-6 4 通道同步注入转换



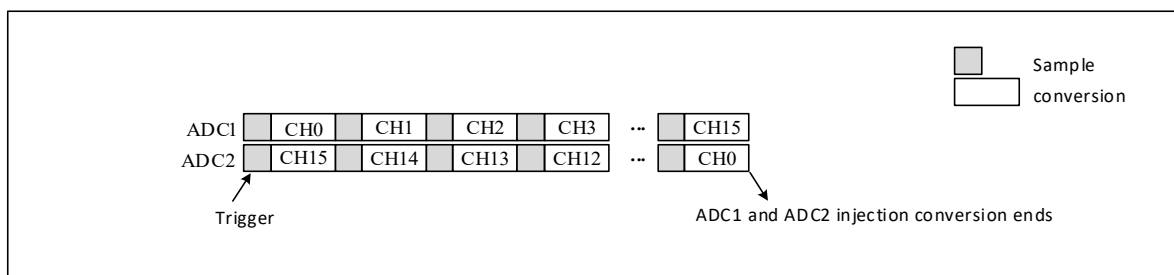
注：1. 同一时刻 ADC1 和 ADC2 的转换通道不应重合。

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_CTLR2 中 EXTSEL[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 11-7 16 通道同步规则转换



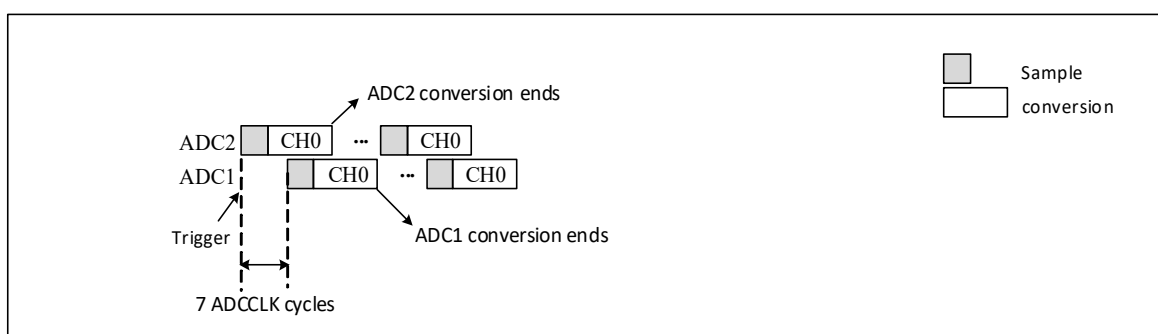
注：1. 同一时刻 ADC1 和 ADC2 的转换通道不应重合；

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_CTLR2 中 EXTSEL[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 11-8 单通道快速交替连续转换

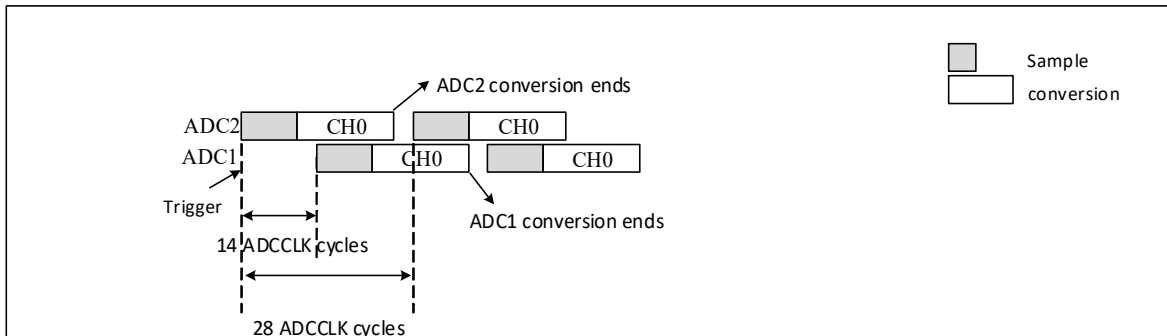


注：采样时间应小于 7 个 ADC 时钟周期，以避免 ADC1 和 ADC2 在采样同一个通道时，出现采样周期重合的问题。

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_CTLR2 中 EXTSEL[2:0]，以选择触发源，当触发生产生后 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 11-9 单通道慢速交替转换

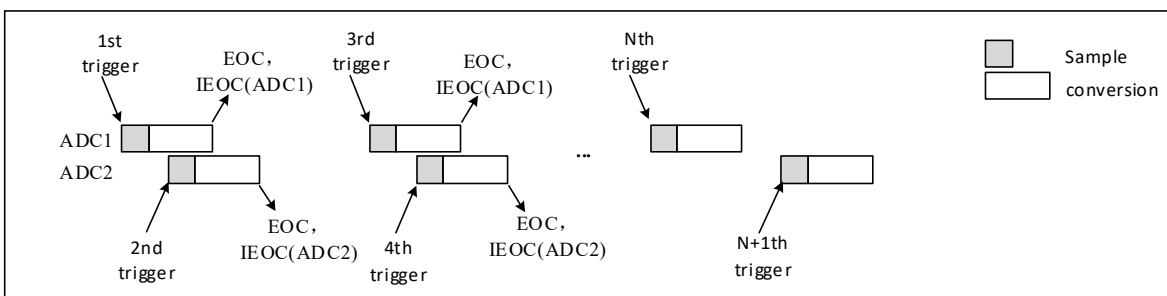


- 注：1. 采样时间应小于 14 个 ADC 时钟周期，以避免和下一次采样周期重合；
2. 28 个 ADC 时钟周期后自动启动新的 ADC2 转换；
3. 不需要设置 CONT 位。

6) 交替触发模式

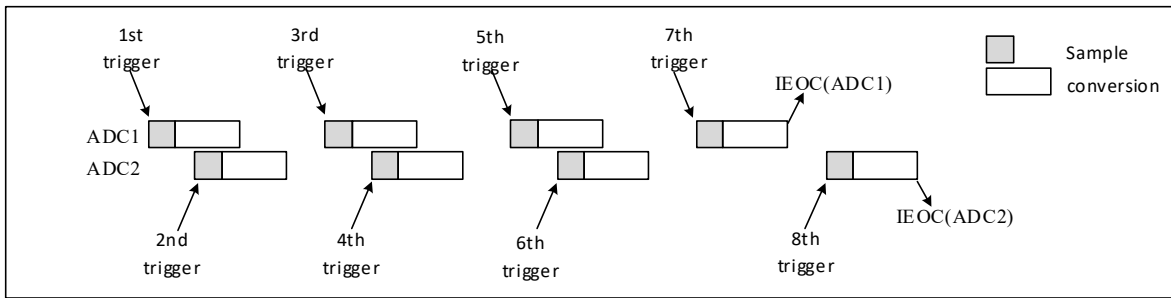
此模式仅适用于注入通道组，设置 ADC1_CTLR2 中 JEXTSEL[2:0]，以选择触发源。当第一次触发事件发生时，ADC1 上所有的注入通道被转换，当第二次触发事件发生时，ADC2 上所有的注入通道被转换。依次循环往复，若使能中断，则当 ADC1 所有注入通道转换完成后产生 JEOC 中断，ADC2 所有注入通道转换完成后产生 JEOC 中断。

图 11-10 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 JEOC 中断，当 ADC2 所有注入通道转换完成后产生 JEOC 中断。

图 11-11 间断模式下每个 ADC 注入通道交替触发转换



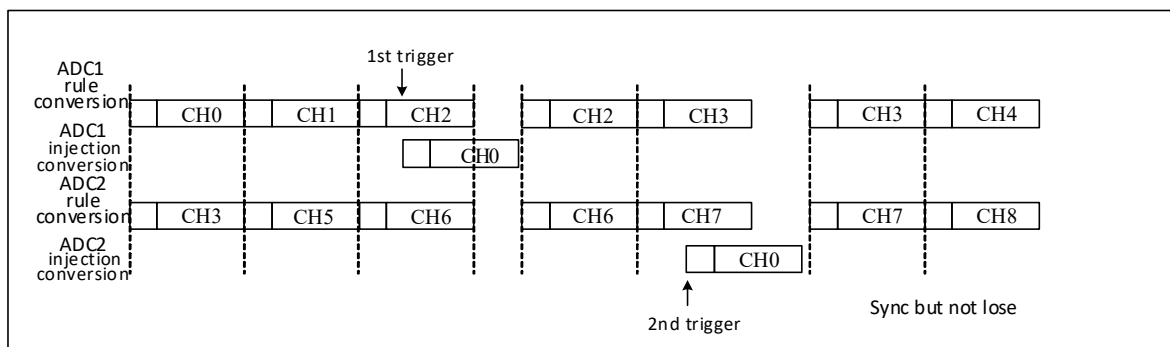
7) 同步规则模式+同步注入模式

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证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。

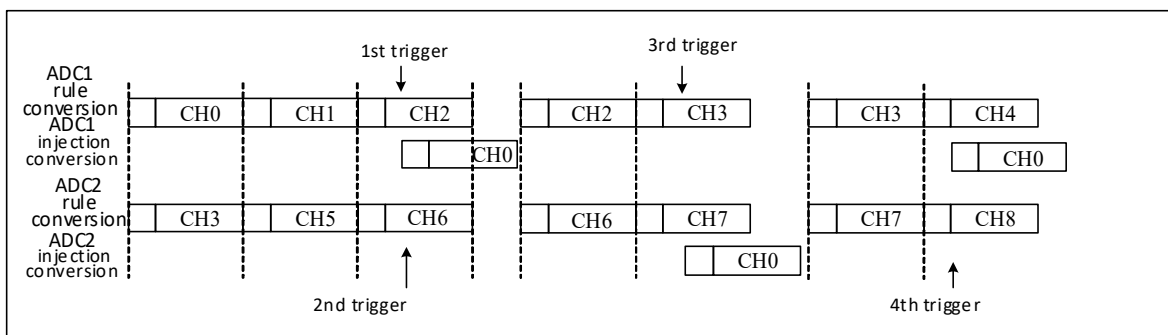
图 11-12 同步规则模式下交替触发注入通道转换



注：此模式下需要保证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 11-13 触发事件发生在注入转换期间

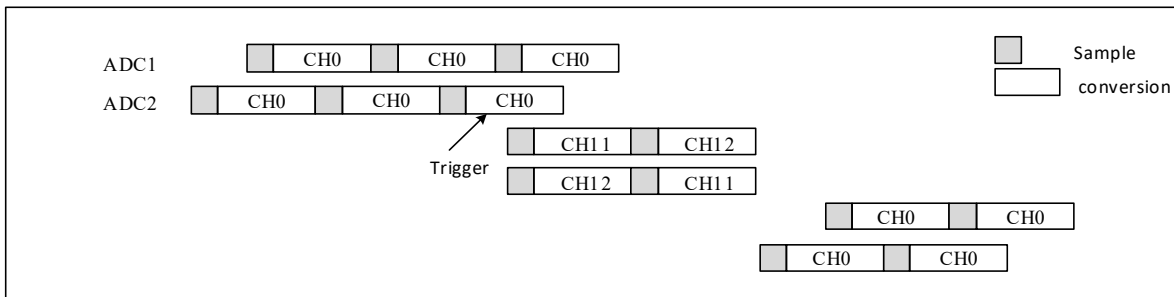


9) 同步注入模式+交替模式

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 11-14 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。

11.2.8 DFSDM 转换

ADC 转换结果可直接传送到具有 $\Sigma \Delta$ 调制器的数字滤波器(DFSDM)。在这种情况下,ADCx_AUX[31] 位必须置为 1, DMA 位必须清除为 0。ADC 会将规则数据寄存器数据的 12 个最低有效位传输到 DFSDM, 一旦传输完成, DFSDM 将重置 EOC 标志。

11.2.9 ADC 低功耗模式

通过 ADCx_CTLR2 寄存器 ADC_LP 位, 可配置 ADC 工作模式。默认为 0 的情况下, 功耗较高, 适用于 6M 及以上采样率, 且只适用于 V_{DD33A} 高于 3V; 置 1 的情况下为低功耗模式, 功耗较低, 适用于 2M 以下采样率。

11.3 寄存器描述

表 11-5 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	0x000000D0
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	0x00000FFF
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_ISR	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
R32_ADC1_AUX	0x40012454	ADC1 采样时间寄存器	0x00000000

表 11-6 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	0x000000D0
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	0x00000FFF
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
R32_ADC2_AUX	0x40012854	ADC2 采样时间寄存器	0x00000000
R32_ADC2_DRV	0x40012858	ADC2_TOUCHKEY 多路屏蔽寄存器	0x00000000

11.3.1 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
ADCRS TF	Reserved										STRT	JSTRT	JEOC	EOC	AWD

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
15	ADCRSTF	R0	ADC 复位状态指示： 1：复位结束； 0：复位中。	0
[14:5]	Reserved	R0	保留。	0
4	STRT	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	JSTRT	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	JEOC	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）。	0
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 ADCx_RDATAR 时清除。	0
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 ADCx_WDHTR 和 ADCx_WDLTR 寄存器范围），由软件清 0（写 1 无效）。	0

11.3.2 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
SW_P RE	ANA_ RST	Rese rved	PGA[1:0]	BUF EN	TKI TUNE	TKENABLE	AWDEN	JAWDEN	Reserved	DUALMOD[3:0]					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			JDISC EN	DISC EN	JAUTO	AWD SGL	SCAN	JEOC IE	AWDIE	EOC IE	AWDCH[4:0]				

位	名称	访问	描述	复位值
31	SW_PRE	RW	通道预切换关闭： 1：通道预切换关闭； 0：通道预切换开启。	0
30	ANA_RST	RW	模拟模块复位使能：	0

			1: 使能; 0: 关闭。 注: 在ADC转换期间复位ADC模拟电路将会破坏转换结果, 此位可与RCC的ADCxRST一起使用。	
29	Reserved	R0	保留。	0
[28:27]	PGA[1:0]	RW	ADC通道增益配置: 00: x1; 01: x4; 10: x16; 11: x64。 注: 输入增益可调, 可实现小信号放大采样。使用此功能需开启ADC_Buffer。	00b
26	BUFEN	RW	ADC BUFFER使能: 1: 使能输入Buffer; 0: 关闭输入Buffer。 注: 需要开启buffer的情况: 1. 当TKENABLE位或TSVREFE位置1, buffer默认开启不可关闭(故在这两种情况下, ADC校准需在TKENABLE位或TSVREFE位置1之前, 且buffer需要关闭)。 2. 当外部输入阻抗大于最大输入阻抗要求, 可开启buffer以改善ADC采集数据(此时ADC的采样时间不建议小于7.5T), 外部输入阻抗详见DS0数据手册。	0
25	TKITUNE	RW	TKEY模块充电电流配置: 1: 充电电流减半; 0: 充电电流为35uA。	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	AWDEN	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	JAWDEN	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	R0	保留。	0
[19:16]	DUALMOD[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。	0000b

			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。	
[15:13]	DISCNUM[2:0]	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。	000b
12	JDISCEN	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	DISCEN	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	JAUTO	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 ADCx_I0FRx 和 ADCx_RSQRx 选择的所有通道）； 0：关闭扫描模式。	0
7	JE0CIE	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（JE0C 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[4:0]	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； ... 10000：模拟输入通道 16； 10001：模拟输入通道 17。	00000b

11.3.3 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	SW START	JSW START	EXT TRIG	EXTSEL[2:0]			Reser ved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JEXT TRIG	JEXTSEL[2:0]			ALIGN	Reserved	DMA	ADC_LP	CAL_UTO	CAL_VOL[1:0]	RST CAL	CAL	CONT	ADON		

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	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。 注：该位仅适用于 ADC1。	0
22	SWSTART	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	JSWSTART	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	EXTTRIG	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]	EXTSEL[2:0]	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； 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：SWSTART 软件触发。	000b
16	Reserved	R0	保留。	0
15	JEXTTRIG	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]	JEXTSEL[2:0]	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：JSWSTART 软件触发。	000b
11	ALIGN	RW	数据对齐方式： 1：左对齐； 0：右对齐。	0

[10:9]	Reserved	R0	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	ADC_LP	RW	ADC 低功耗模式控制位： 1：低功耗模式，功耗较低，适用于 2M 以下采样率； 0：功耗较高，适用于 6M 及以上采样率，且只适用于 V_{DD33A} 高于 3V。	1
6	CAL_AUTO	RW	硬件自动校准功能使能： 1：ADC 结果将自动减去偏移值； 0：ADC 结果不会减去偏移值。	1
[5:4]	CAL_VOL[1:0]	RW	校准电压配置： 00：0.3*Vref（推荐使用）； 01：0.5*Vref； 10：0.75*Vref； 11：悬空。	01b
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，如果还有其他任意位发生变化，则不会启动新的转换。 1：开启 ADC 并启动转换； 0：关闭 ADC 转换/校准，并进入断电模式。	0

11.3.4 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	R0	保留。	0

[23:0]	SMPx[2:0]	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; SMPx[2:0]: 置 ADC_SMP_SELx 位时, 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: 2.5 周期; 101: 3.5 周期; 110: 4.5 周期; 111: 5.5 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。 注: 采样率高于 1MHz 时, 采样时间不建议小于 3.5 周期。	000b

11.3.5 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	R0	保留。	0
[29:0]	SMPx[2:0]	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; SMPx[2:0]: 置 ADC_SMP_SELx 位时, 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: 2.5 周期; 101: 3.5 周期; 110: 4.5 周期; 111: 5.5 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。 注: 采样率高于 1MHz 时, 采样时间不建议小于 3.5 周期。	000b

11.3.6 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				JOFFSETx[11:0]											

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0
[11:0]	JOFFSETx[11:0]	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 ADCx_IDATARx 寄存器中读出。	0

11.3.7 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	R0	保留。	0
[11:0]	HT[11:0]	RW	模拟看门狗高阈值设置值。	0xFFFF

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

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

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				LT[11:0]											

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0
[11:0]	LT[11:0]	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 WDLTR 的值, 但它们将在下次转换时生效。

11.3.9 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								L[3:0]				RSQ16[4:1]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ16[0]	SQ15[4:0]					SQ14[4:0]					SQ13[4:0]				

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:20]	L[3:0]	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]	SQ16[4:0]	RW	规则序列中的第 16 个转换通道的编号（0-17）。	0
[14:10]	SQ15[4:0]	RW	规则序列中的第 15 个转换通道的编号（0-17）。	0
[9:5]	SQ14[4:0]	RW	规则序列中的第 14 个转换通道的编号（0-17）。	0
[4:0]	SQ13[4:0]	RW	规则序列中的第 13 个转换通道的编号（0-17）。	0

11.3.10 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		SQ12[4:0]					SQ11[4:0]					SQ10[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ10[0]	SQ9[4:0]					SQ8[4:0]					SQ7[4:0]				

位	名称	访问	描述	复位值
[31:30]	Reserved	R0	保留。	0
[29:25]	SQ12[4:0]	RW	规则序列中的第 12 个转换通道的编号（0-17）。	0
[24:20]	SQ11[4:0]	RW	规则序列中的第 11 个转换通道的编号（0-17）。	0
[19:15]	SQ10[4:0]	RW	规则序列中的第 10 个转换通道的编号（0-17）。	0
[14:10]	SQ9[4:0]	RW	规则序列中的第 9 个转换通道的编号（0-17）。	0
[9:5]	SQ8[4:0]	RW	规则序列中的第 8 个转换通道的编号（0-17）。	0
[4:0]	SQ7[4:0]	RW	规则序列中的第 7 个转换通道的编号（0-17）。	0

11.3.11 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		SQ6[4:0]					SQ5[4:0]					SQ4[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4[0]	SQ3[4:0]					SQ2[4:0]					SQ1[4:0]				

位	名称	访问	描述	复位值
[31:30]	Reserved	R0	保留。	0
[29:25]	SQ6[4:0]	RW	规则序列中的第 6 个转换通道的编号（0-17）。	0
[24:20]	SQ5[4:0]	RW	规则序列中的第 5 个转换通道的编号（0-17）。	0
[19:15]	SQ4[4:0]	RW	规则序列中的第 4 个转换通道的编号（0-17）。	0
[14:10]	SQ3[4:0]	RW	规则序列中的第 3 个转换通道的编号（0-17）。	0

[9:5]	SQ2[4:0]	RW	规则序列中的第 2 个转换通道的编号 (0-17)。	0
[4:0]	SQ1[4:0]	RW	规则序列中的第 1 个转换通道的编号 (0-17)。	0

11.3.12 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										JL[1:0]		JSQ4[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ4[0]		JSQ3[4:0]				JSQ2[4:0]				JSQ1[4:0]					

位	名称	访问	描述	复位值
[31:22]	Reserved	RO	保留。	0
[21:20]	JL[1:0]	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]	JSQ4[4:0]	RW	注入序列中的第 4 个转换通道的编号 (0-17)。 注: 软件写入, 并将通道编号 (0-17) 分配为要转换的序列中的第 4 个。	0
[14:10]	JSQ3[4:0]	RW	注入序列中的第 3 个转换通道的编号 (0-17)。	0
[9:5]	JSQ2[4:0]	RW	注入序列中的第 2 个转换通道的编号 (0-17)。	0
[4:0]	JSQ1[4:0]	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 JL[1:0] 的长度小于 4, 则转换的序列顺序是从 (4-JL) 开始。

例如, 当 JL[1:0]=3 (定序器中有 4 次注入转换) 时, ADC 将按以下顺序转换通道: JSQ1[4:0]、JSQ2[4:0]、JSQ3[4:0] 和 JSQ4[4:0];

当 JL[1:0]=2 (定序器中有 3 次注入转换) 时, ADC 将按以下顺序转换通道: JSQ2[4:0]、JSQ3[4:0] 和 JSQ4[4:0];

当 JL[1:0]=1 (定序器中有 2 次注入转换) 时, ADC 转换通道的顺序为: 先是 JSQ3[4:0], 后是 JSQ4[4:0];

当 JL[1:0]=0 (定序器中有 1 次注入转换) 时, ADC 将仅转换 JSQ4[4:0] 通道。

如果 ADCx_ISQR[21:0]=10 00111 00011 00111 00010, ADC 将按以下顺序转换通道: JSQ2[4:0]、JSQ3[4:0] 和 JSQ4[4:0], 表示扫描转换按以下通道顺序进行: 7、3、7。

11.3.13 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
JDATA[15:0]															

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

11.3.14 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

ADC2DATA[15:0]															
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]	ADC2DATA[15:0]	R0	ADC2 转换的数据： 在 ADC1 中：双模式下，这些位包含了 ADC2 转换的规则通道数据。 在 ADC2 中：不使用这些位。	0
[15:0]	DATA[15:0]	R0	规则通道转换数据（数据左对齐或右对齐）。	0

11.3.15 ADCx 采样时间寄存器 (ADCx_AUX) (x=1/2)

偏移地址：0x54

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC_T O_DFS DM	Reserved													ADC_S MP_SE L17	ADC_S MP_SE L16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC_S MP_SE L15	ADC_S MP_SE L14	ADC_S MP_SE L13	ADC_S MP_SE L12	ADC_S MP_SE L11	ADC_S MP_SE L10	ADC_S MP_SE L9	ADC_S MP_SE L8	ADC_S MP_SE L7	ADC_S MP_SE L6	ADC_S MP_SE L5	ADC_S MP_SE L4	ADC_S MP_SE L3	ADC_S MP_SE L2	ADC_S MP_SE L1	ADC_S MP_SE L0

位	名称	访问	描述	复位值
31	ADC_TO_DFSDM	RW	ADC 模块采样数据送给 DFSDM 模块功能使能： 1：使能； 0：关闭。	0
[30:18]	Reserved	R0	保留。	0
[17:0]	ADC_SMP_SELx	RW	x=0-17，通道 x 的采样时间可选使能位（仅在 SMPx 为 100-111）： 1：SMPx=100：采样时间 2.5 个周期； SMPx=101：采样时间 3.5 个周期； SMPx=110：采样时间 4.5 个周期； SMPx=111：采样时间 5.5 个周期。 0：采样时间为对应 SMPx 配置周期。	0

11.3.16 ADCx_TOUCHKEY 多路屏蔽寄存器 (ADCx_DRV) (x=2)

偏移地址：0x58

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

位	名称	访问	描述	复位值
[31:17]	Reserved	R0	保留。	0
16	TKEY_DRV_EN	RW	TOUCHKEY 多路屏蔽使能： 1：使能； 0：禁止。	0
[15:0]	TKEY_DRV_OUTEN	RW	TOUCHKEY 多路屏蔽各通道使能： 1：使能； 0：禁止。	0

第 12 章 高速模拟/数字转换（HSADC）

芯片内置了 1 个 10 位的高速模拟/数字转换器（HSADC），提供 7 个外部通道采样，其采样速率高达 20MSPS，可以实现连续转换，支持使用 DMA 操作。

12.1 主要特征

- 支持 7 个外部通道采样
- 支持接收 FIFO
- 支持突发模式
- 支持 DMA 功能
- 支持乒乓存储模式
- 支持连续转换方式
- 提供多种传输中断标志和状态
- 提供 1 个中断向量

12.2 功能描述

12.2.1 HSADC 模块上电

R32_HSADC_CFGR 寄存器 EN 位为 1 表示 ADC 模块上电。当 ADC 模块从断电模式（EN=0）下进入上电状态（EN=1）后，需要延迟一段时间 t_{STAB} 用于模块稳定时间。之后写入 START 位为 1，用于作为软件启动 ADC 转换的启动信号。通过清除 EN 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。

12.2.2 HSADC 时钟

HSADC 时钟源可来自系统时钟（SYSCLK）、PLL 时钟、USBHS_PLL 时钟、ETH_PLL 时钟，可通过 RCC_CFGR2 寄存器 HSADCSRC[1:0] 位操作选择，还可通过 R32_HSADC_CFGR 寄存器 CLKDIV[5:0] 位对 HSADC 时钟进行分频。

采样周期总转换时间如下计算：

$$T_{CONV} = \text{采样时间} + 4T_{HSADCCLK} ;$$

采样时间固定为 $1T_{HSADCCLK}$ 。

12.2.3 HSADC 通道配置

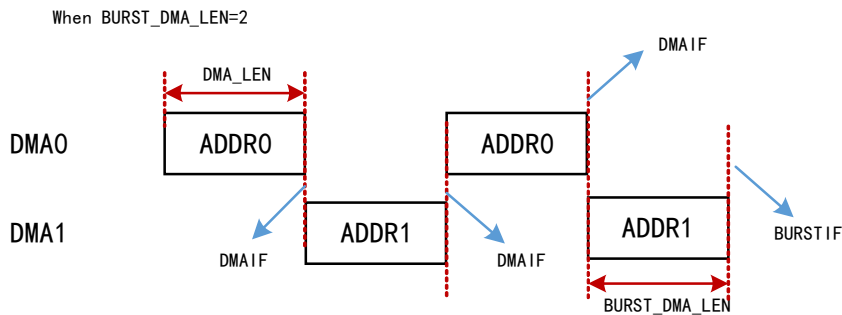
HSADC 模块提供了 7 个外部通道采样源。

12.2.4 HSADC DMA 功能

HSADC 的通道转换支持 DMA 功能。通道转换的值储存在一个仅有的数据寄存器 R32_HSADC_DATAR 中，没有及时取走 R32_HSADC_DATAR 寄存器中的数据，可以开启 ADC 的 DMA 功能（若不使能乒乓传输，仅可设置 DMA0）。硬件会在通道的转换结束时（EOC 置位）产生 DMA 请求，并将转换的数据从 R32_HSADC_DATAR 寄存器传输到用户指定的目的地址。

支持突发传输方式，突发传输时置位 PPMODE 位，开启乒乓存储模式，DMA 传输达到传输长度后会在目的地址 ADDR0/1 中交替；通过 R32_HSADC_CTLR 寄存器 BURST_END 位置 1，使能中止一次突发传输，在达到设置突发长度即中止传输，终次突发传输的 DMA 传输长度可根据需要在 R32_HSADC_CTLR2 寄存器 BURST_DMA_LEN 位设置。

下图以 BURST_DMA_LEN=2 为例展示突发传输功能：



12.2.5 HSADC 转换模式

HSADC 只支持连续转换，在连续转换模式中，当前面 ADC 转换一结束马上就启动另一次转换。此模式的启动事件通过设置 R32_HSADC_CTLR1 寄存器 START 位置 1 启动。一旦选择通道的转换完成将转换数据被储存在 10 位 R32_HSADC_DATAR 寄存器中，EOC 标志被置位，如果设置了 EOCIE 位，将触发 ADC 中断。

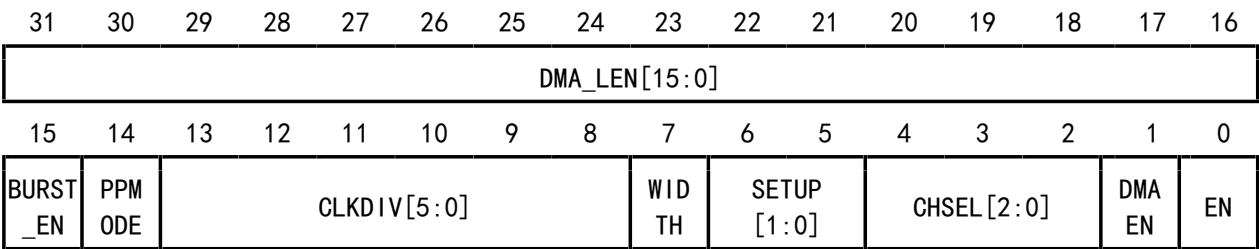
12.3 寄存器描述

表 12-1 HSADC 通道相关寄存器列表

名称	访问地址	描述	复位值
R32_HSADC_CFGR	0x40017400	HSADC 配置寄存器	0x00002400
R32_HSADC_CTLR1	0x40017404	HSADC 控制寄存器 1	0x00000000
R32_HSADC_CTLR2	0x40017408	HSADC 控制寄存器 2	0x00000000
R32_HSADC_STATR	0x4001740C	HSADC 状态寄存器	0x00000000
R32_HSADC_DATAR	0x40017410	HSADC 数据寄存器	0x00000000
R32_HSADC_ADDR0	0x40017414	HSADC DMA 接收地址寄存器 0	0x00000000
R32_HSADC_ADDR1	0x40017418	HSADC DMA 接收地址寄存器 1	0x00000000

12.3.1 HSADC 配置寄存器（R32_HSADC_CFGR）

偏移地址：0x00



位	名称	访问	描述	复位值
[31:16]	DMA_LEN[15:0]	RW	DMA 传输长度配置位。	0
15	BURST_EN	RW	突发传输使能： 1：使能； 0：关闭。	0
14	PPMODE	RW	乒乓存储模式使能： 1：使能； 0：关闭。	0

[13:8]	CLKDIV[5:0]	RW	高速 ADC 时钟分频配置位： 000000：不分频； 000001：2 分频； 000010：3 分频； 111111：64 分频。	0x9
7	WIDTH	RW	存储位宽配置： 1：8 位； 0：16 位。	0
[6:5]	SETUP[1:0]	RW	首次转换建立时间配置位： 00：8 个周期； 01：9 个周期； 10：10 个周期； 11：11 个周期。	0
[4:2]	CHSEL[2:0]	RW	通道选择位： 000：输入通道 0； 110：输入通道 6； 111：保留。	0
1	DMAEN	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
0	EN	RW	高速 ADC 使能： 1：使能； 0：关闭。	0

12.3.2 HSADC 控制寄存器 1（R32_HSADC_CTLR1）

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					BURSTIE	DMAIE	EOCIE	Reserved					BURST_END	START	

位	名称	访问	描述	复位值
[31:11]	Reserved	RO	保留。	0
10	BURSTIE	RW	突发传输完成中断使能： 1：使能； 0：关闭。	0
9	DMAIE	RW	DMA 传输完成中断使能： 1：使能； 0：关闭。	0
8	EOCIE	RW	转换完成中断使能：	0

			1: 使能; 0: 关闭。	
[7:2]	Reserved	R0	保留。	0
1	BURST_END	RW	中止一次突发传输。	0
0	START	WO	启动高速 ADC 转换: 1: 开启; 0: 关闭。	0

12.3.3 HSADC 控制寄存器 2 (R32_HSADC_CTLR2)

偏移地址: 0x08

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

位	名称	访问	描述	复位值
[31:16]	BURST_LEN[15:0]	RW	突发传输长度配置。	0
[15:0]	BURST_DMA_LEN[15:0]	RW	突发传输最终次 DMA 传输长度配置位。最终次突发传输数据数量与 DMA 传输长度非对齐时, DMA 传输长度配置位。	0

12.3.4 HSADC 状态寄存器 (R32_HSADC_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	FIFO_CNT[2:0]			FIF O_O V	FIF O_F ULL	FIF O_R DY	Reserved			PP_ ADD R	RXN E	BURST IF	DMA IF	EOC IF	

位	名称	访问	描述	复位值
[31:14]	Reserved	R0	保留。	0
[13:11]	FIFO_CNT[2:0]	R0	接收 FIFO 当前计数值。	0
10	FIFO_OV	R0	接收 FIFO 溢出状态标志位。	0
9	FIFO_FULL	R0	接收 FIFO 满状态标志。	0
8	FIFO_RDY	R0	接收 FIFO 非空状态标志。	0
[7:5]	Reserved	R0	保留。	0
4	PP_ADDR	R0	乒乓存储模式缓存地址指示位: 1: 数据存储在 DMA 接收地址 1; 0: 数据存储在 DMA 接收地址 0。	0
3	RXNE	R0	数据寄存器非空标志, 即转换完成数据存入数据寄存器中, 该位置位; 对数据寄存器的读操作可将该位清除。	0

2	BURSTIF	RW1	突发传输完成中断标志。	0
1	DMAIF	RW1	DMA 传输完成中断标志。	0
0	EOCIF	RW1	转换完成中断标志。	0

12.3.5 HSADC 数据寄存器 (R32_HSADC_DATAR)

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								DR[9:0]							

位	名称	访问	描述	复位值
[31:10]	Reserved	R0	保留。	0
[9:0]	DR[9:0]	RW	转换数据寄存器。	0

12.3.6 HSADC DMA 接收地址寄存器 0 (R32_HSADC_ADDR0)

偏移地址: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA_ADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_ADDR0[15:0]															

位	名称	访问	描述	复位值
[31:0]	DMA_ADDR0[31:0]	RW	DMA 传输地址 0 配置位。	0

12.3.7 HSADC DMA 接收地址寄存器 1 (R32_HSADC_ADDR1)

偏移地址: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA_ADDR1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_ADDR1[15:0]															

位	名称	访问	描述	复位值
31:0	DMA_ADDR1[31:0]	RW	DMA 传输地址 1 配置位。	0

第 13 章 触摸按键检测（TKEY）

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

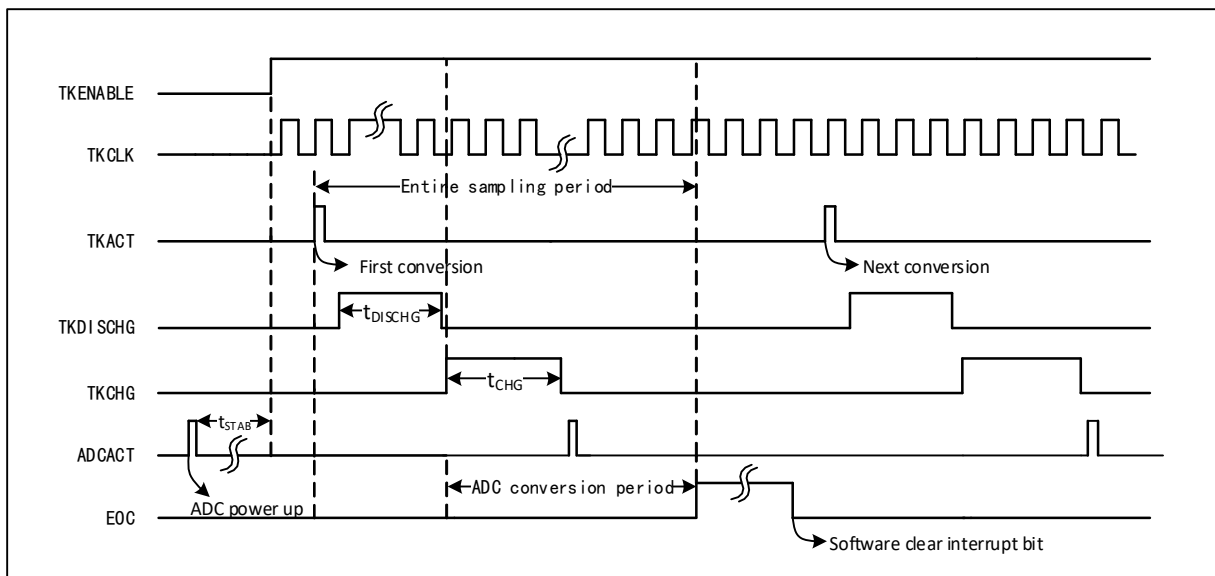
● TKEY 开启

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 HCLK 时钟周期（ t_{DISCHG} ）进行放电，然后再通过若干个 HCLK 周期（ t_{CHG} ）对通道进行充电进行电压采样，充电周期数为 TKEY_CHGOFFSET 值，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 L[3:0]为 1。

- 3) 写 TKEY_CHGOFFSET 寄存器，设置通道的充电时间（低 16 位有效），以调整充电时间。
- 4) 写 TKEY_ACT_DCG 寄存器，设置放电时间（低 16 位有效），并启动一次 TKEY 的采样和转换。
- 5) 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。
- 6) 如果需要进行下次转换，重复 2-5 步骤。如果不需要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_CHGOFFSET	0x4001243C	TKEY 充电时间寄存器	0x00000000
R32_TKEY1_ACT_DCG	0x4001244C	TKEY 启动和放电时间寄存器	0x00000000
R32_TKEY1_DR	0x4001244C	TKEY 数据寄存器	0x00000000

表 13-2 TKEY2 相关寄存器列表

名称	访问地址	描述	复位值
R32_TKEY2_CHGOFFSET	0x4001283C	TKEY 充电时间寄存器	0x00000000
R32_TKEY2_ACT_DCG	0x4001284C	TKEY 启动和放电时间寄存器	0x00000000
R32_TKEY2_DR	0x4001284C	TKEY 数据寄存器	0x00000000

13.3.1 TKEYx 充电时间寄存器（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
TKCGOFFSET[15:0]															

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	TKCGOFFSET[15:0]	WO	TKEY 充电时间配置值。 总充电时间 TCHG = TKCGOFFSET	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（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。

13.3.2 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
TKACT_DCG[15:0]															

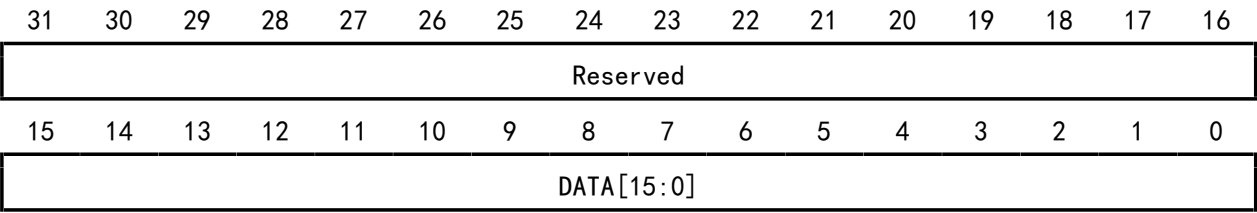
位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0

[15:0]	TKACT_DCG[15:0]	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）。

13.3.3 TKEYx 数据寄存器（TKEYx_DR）（x=1/2）

偏移地址：0x4C



位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	DATA[15:0]	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)

高级定时器模块包含 2 个 16 位的自动装载递增/递减计数器 (TIM1 和 TIM8)，具有 16 位可编程的预分频器。除了完整的通用定时器功能外，可以被看成是分配到 6 个通道的三相 PWM 发生器，具有带死区插入的互补 PWM 输出功能，允许在指定数目的计数器周期之后更新定时器进行重复计数周期，刹车功能等。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。

高级定时器的很多功能都与通用定时器相同，内部结构也相同，因此高级定时器可以通过定时器链接功能与其他 TIM 定时器协同操作，提供同步或事件链接功能。

14.1 主要特征

高级定时器 (TIM1/8) 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

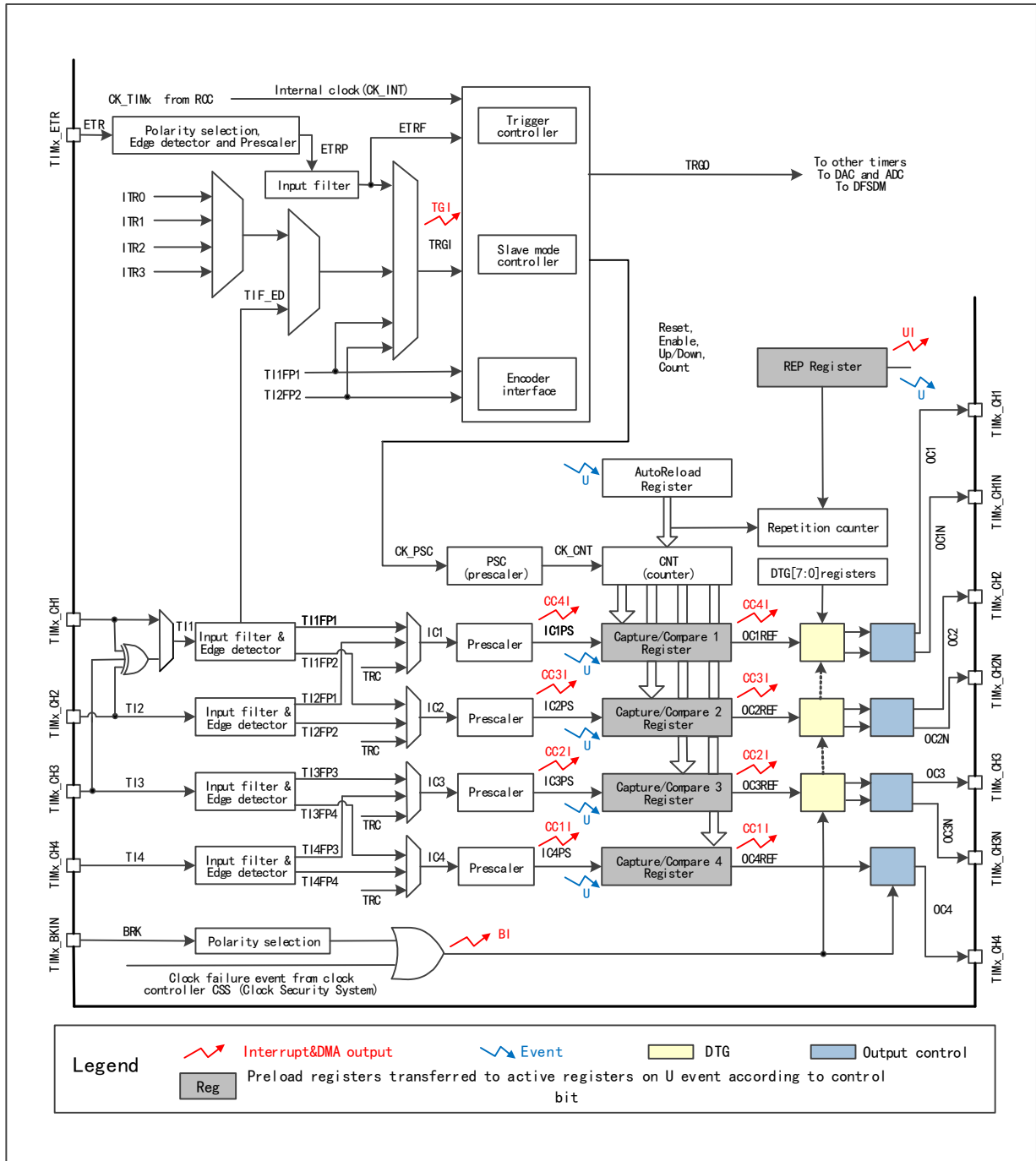
如图 14-1，高级定时器的结构大致可以分为三部分，即输入时钟部分，核心计数器部分和比较捕获通道部分。

高级定时器的时钟可以来自于 HB 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 重装初始值。另外还有个辅助计数器在一旁计数 ATRLR 为 CNT 重装初值的次数，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。

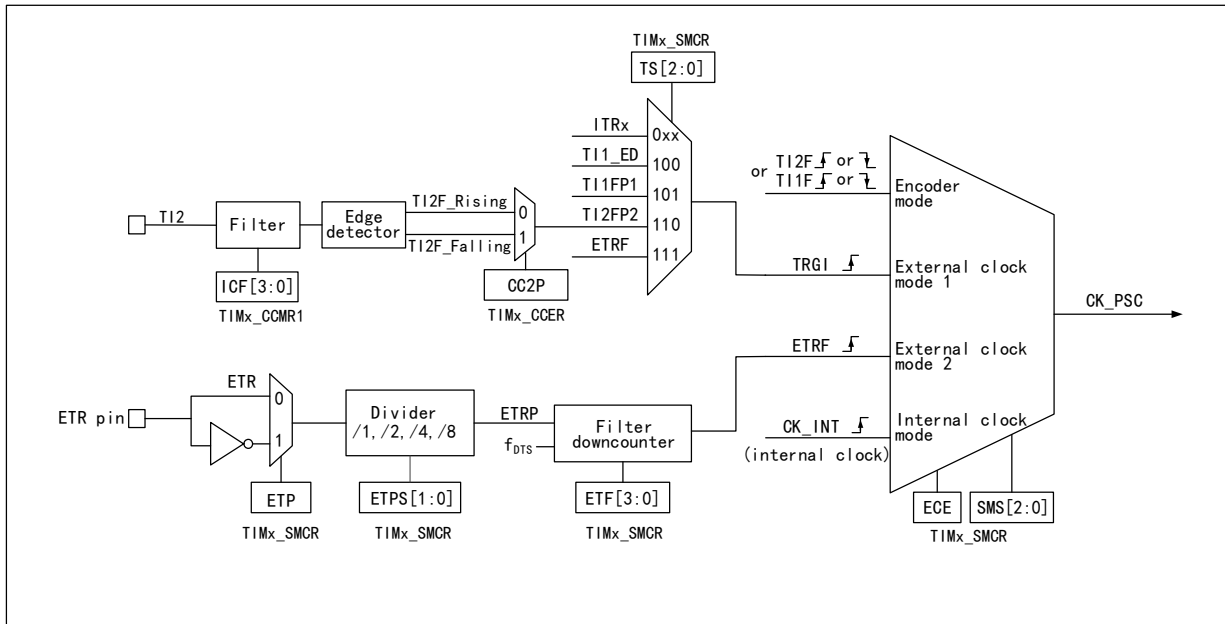
高级定时器拥有四组比较捕获通道，每组比较捕获通道都可以从专属的引脚上输入脉冲，也可以向引脚输出波形，即比较捕获通道支持输入和输出模式。比较捕获寄存器每个通道的输入都支持滤波、分频和边沿检测等操作，并支持通道间的互触发，还能为核心计数器 CNT 提供时钟。每个比较捕获通道都拥有一组比较捕获寄存器 (CHxCVR)，支持与主计数器 (CNT) 进行比较而输出脉冲。

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 HB 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 经过边沿检测器后的信号（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。

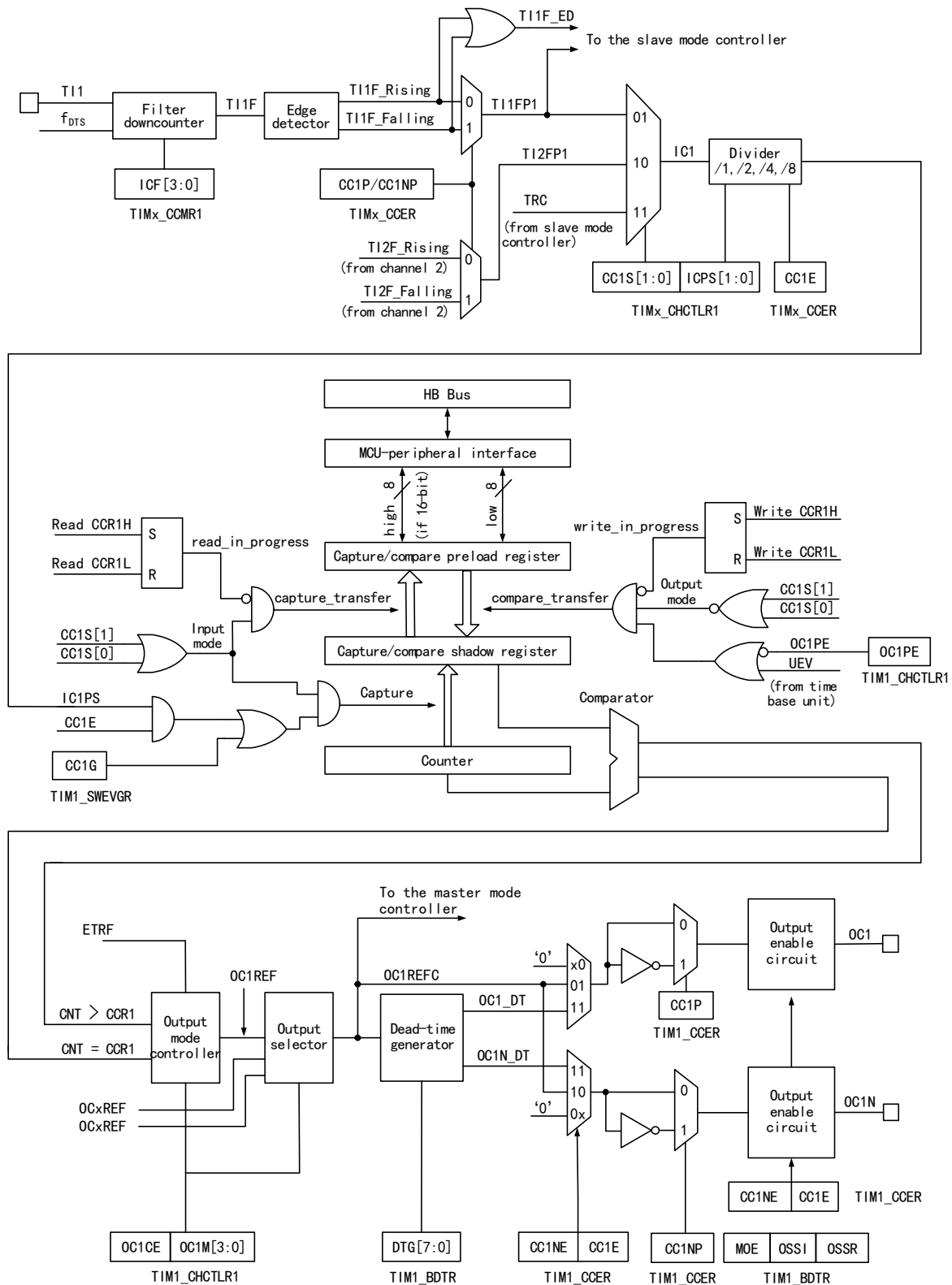
14.2.3 计数器和周边

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_CNT。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), 以 $TI1$ 举例: $TI1$ 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 CC1S 域，选择 IC1 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，CC1S 域默认是使比较捕获模块作为输出通道；
- 2) 配置 IC1F 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。
- 3) 配置 CC1P 位，设定 TI1FP1 的极性。比如保持 CC1P 位为低，选择上升沿跳变；
- 4) 配置 IC1PS 域，设定 IC1 信号成为 IC1PS 之间的分频系数。比如保持 IC1PS 为 00b，不分频；
- 5) 配置 CC1E 位，允许捕获核心计数器（CNT）的值到比较捕获寄存器中。
- 6) 根据是否需要中断或者 DMA 来配置 CC1IE 和 CC1DE 位。

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（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

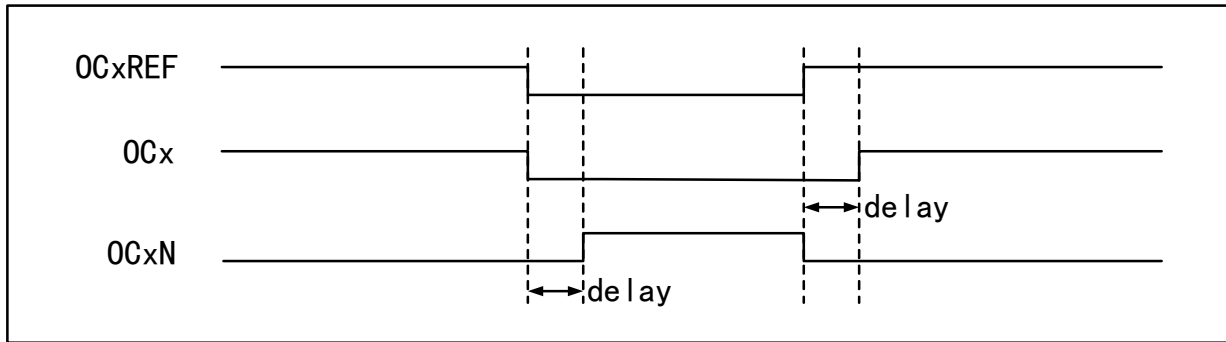
使用中央对齐模式时，核心计数器运行在增计数和减计数交替进行的模式下，OCxREF 在核心计数器和比较捕获寄存器的值一致时进行上升和下降的跳变。但比较标志在三种中央对齐模式下，置位的时机有所不同。在使用中央对齐模式时，最好在启动核心计数器之前产生一个软件更新标志（置 UG 位）。

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、OSS1、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 HB 时钟的延迟，因此需要等一个 HB 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：

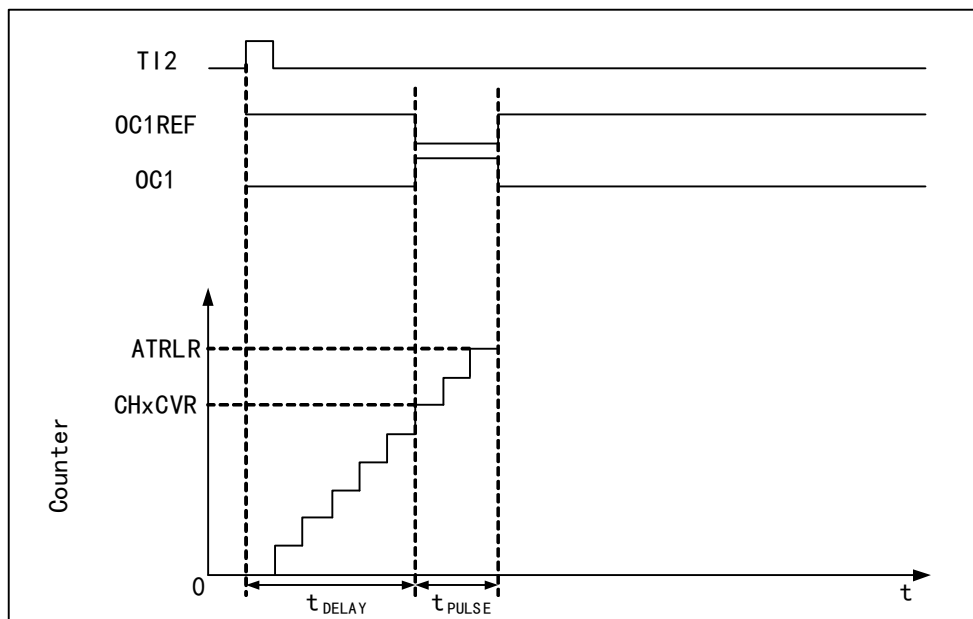
- 1) MOE 位被异步清零，根据 OSSI 位的设置将输出置为无效状态、空闲状态或复位状态；
- 2) 在 MOE 被清零后，每一个输出通道输出由 OISx 确定的电平；
- 3) 当使用互补输出时：输出被置于无效状态，具体取决于极性；
- 4) 如果 BIE 被置位，当 BIF 置位，会产生一个中断；如果设置了 BDE 位，则会产生一个 DMA 请求；
- 5) 如果 AOE 被置位，在下一个更新事件 UEV 时，MOE 位被自动置位。

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 t_{delay} 之后，在 OC1 上产生一个长度为 t_{pulse} 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

计数有效边沿	相对信号的电平	TI1FP1 信号边沿		TI2FP2 信号	
		上升沿	下降沿	上升沿	下降沿
仅在 TI1 边沿计数	高	向下计数	向上计数	不计数	
	低	向上计数	向下计数		
仅在 TI2 边沿计数	高	不计数		向上计数	向下计数
	低			向下计数	向上计数
在 TI1 和 TI2 双边沿计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

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

14.3.11 双边沿捕获模式

可通过 TIMx_AUX 寄存器 CAP_ED_CHx 位开启对应通道双边沿捕获对脉冲进行测量。

例如: 通过通道 2 来捕获脉冲宽度, 在捕获功能配置上, 选择 IC2 信号的来源 (TIMx_CHCTLR1 寄存器的 CC2S 位为 11b), 使能通道 2 的双边沿捕获功能 (TIMx_AUX 寄存器的 CAP_ED_CH2 位为 1)。至此已经将双边沿捕获配置完成。

可通过 TIMx_CH2CVR 寄存器的 CH2CVR 位读取捕获的脉冲的高低电平宽度值, 开启 CAPLVL 位可通过 TIMx_CH2CVR 寄存器 bit[16] 指示捕获值对应的电平。

14.3.12 死区时间不对称

在互补输出配置和死区控制基础上通过配置 DT_VLU2 位 (配置值大小需大于 0 且小于 TIMx_BDTR 寄存器 DTG[7:0] 中设置的 Tdtg 的个数), 配置 DTP_MODE 位或者 DTN_MODE 位选择 DT_VLU2 配置的死区时间发生在 OCXREF 的上升沿或下降沿。通过此方法实现死区时间不对称。

14.3.13 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_SMCFR	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	0xFFFF
R16_TIM1_RPTCR	0x40012C30	重复计数值寄存器	0x0000
R32_TIM1_CH1CVR	0x40012C34	比较/捕获寄存器 1	0x0000
R32_TIM1_CH2CVR	0x40012C38	比较/捕获寄存器 2	0x0000
R32_TIM1_CH3CVR	0x40012C3C	比较/捕获寄存器 3	0x0000
R32_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_DMACFR	0x40012C48	DMA 控制寄存器	0x0000
R16_TIM1_DMAADR	0x40012C4C	连续模式的 DMA 地址寄存器	0x0000
R16_TIM1_AUX	0x40012C50	双边沿捕获寄存器	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_SMCFR	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	0xFFFF
R16_TIM8_RPTCR	0x40013430	重复计数值寄存器	0x0000
R32_TIM8_CH1CVR	0x40013434	比较/捕获寄存器 1	0x0000

R32_TIM8_CH2CVR	0x40013438	比较/捕获寄存器 2	0x0000
R32_TIM8_CH3CVR	0x4001343C	比较/捕获寄存器 3	0x0000
R32_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_DMACHFR	0x40013448	DMA 控制寄存器	0x0000
R16_TIM8_DMAADR	0x4001344C	连续模式的 DMA 地址寄存器	0x0000
R16_TIM8_AUX	0x40013450	双边沿捕获寄存器	0x0000

14.4.1 控制寄存器 1 (TIMx_CTLR1) (x=1/8)

偏移地址: 0x00

位	名称	访问	描述	复位值
15	CAPLVL	RW	双沿捕获模式下, 捕获电平指示使能。 1: 使能指示功能; 0: 关闭指示功能。 <i>注: 使能后, CHxCVR 的[16]指示捕获值对应的电平。</i>	0
14	CAPOV	RW	捕获值模式配置。 1: 当捕获前产生计数器溢出时, CHxCVR 值为 0xFFFF; 0: 捕获值为实际计数器的值。	0
[13:10]	Reserved	RO	保留。	0
[9:8]	CKD[1:0]	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。	00b
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[1:0]	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。	00b

			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。	
4	DIR	RW	计数器方向： 1：计数器的计数模式为减计数； 0：计数器的计数模式为增计数。 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时，计数器停止（清除 CEN 位）； 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（ARR、PSC、CHxCVR）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。 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：禁止计数器。	0

14.4.2 控制寄存器 2（TIMx_CTLR2）（x=1/8）

偏移地址：0x04

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 OC4=1； 0：当 MOE=0 时，如果实施了 OC4N，则死区后 OC4=0。 注：已经设置了 LOCK（TIMx_BDTR 寄存器）级别 1、	0

			2 或 3 后, 该位不能被修改。	
13	OIS3N	RW	输出空闲状态 3: 1: 当 MOE=0 时, 死区后 OC3N=1; 0: 当 MOE=0 时, 死区后 OC3N=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	OIS2N	RW	输出空闲状态 2, 参见 OIS3N。	0
10	OIS2	RW	输出空闲状态 2, 参见 OIS4。	0
9	OIS1N	RW	输出空闲状态 1, 参见 OIS3N。	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[2:0]	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出 (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)。	000b
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: 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

14.4.3 从模式控制寄存器 (TIMx_SMCFGR) (x=1/8)

偏移地址：0x08

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[1:0]	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。	00b
[11:8]	ETF[3:0]	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 F _{sampling} =F _{ck_int} , N=2； 0010: 采样频率 F _{sampling} =F _{ck_int} , N=4； 0011: 采样频率 F _{sampling} =F _{ck_int} , N=8； 0100: 采样频率 F _{sampling} =Fdts/2, N=6； 0101: 采样频率 F _{sampling} =Fdts/2, N=8； 0110: 采样频率 F _{sampling} =Fdts/4, N=6； 0111: 采样频率 F _{sampling} =Fdts/4, N=8； 1000: 采样频率 F _{sampling} =Fdts/8, N=6；	0000b

			1001: 采样频率 $F_{\text{sampling}} = F_{\text{dts}}/8$, $N=8$; 1010: 采样频率 $F_{\text{sampling}} = F_{\text{dts}}/16$, $N=5$; 1011: 采样频率 $F_{\text{sampling}} = F_{\text{dts}}/16$, $N=6$; 1100: 采样频率 $F_{\text{sampling}} = F_{\text{dts}}/16$, $N=8$; 1101: 采样频率 $F_{\text{sampling}} = F_{\text{dts}}/32$, $N=5$; 1110: 采样频率 $F_{\text{sampling}} = F_{\text{dts}}/32$, $N=6$; 1111: 采样频率 $F_{\text{sampling}} = F_{\text{dts}}/32$, $N=8$ 。	
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[2:0]	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。	000b
3	Reserved	RO	保留。	0
[2:0]	SMS[2:0]	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。	000b

14.4.4 DMA/中断使能寄存器 (TIMx_DMAINTENR) (x=1/8)

偏移地址：0x0C

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	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： 1：允许更新中断； 0：禁止更新中断。	0

14.4.5 中断状态寄存器 (TIMx_INTFR) (x=1/8)

偏移地址：0x10

位	名称	访问	描述	复位值
[15:13]	Reserved	RO	保留。	0
12	CC40F	RWO	比较捕获通道 4 重复捕获标志位。	0
11	CC30F	RWO	比较捕获通道 3 重复捕获标志位。	0
10	CC20F	RWO	比较捕获通道 2 重复捕获标志位。	0
9	CC10F	RWO	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	Reserved	RO	保留。	0
7	BIF	RWO	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	RWO	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	RWO	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	RWO	比较捕获通道 4 中断标志位。	0
3	CC3IF	RWO	比较捕获通道 3 中断标志位。	0
2	CC2IF	RWO	比较捕获通道 2 中断标志位。	0
1	CC1IF	RWO	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： 当捕获事件发生时该位由硬件置位，它由软件清零	0

			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	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; 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)

偏移地址: 0x14

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: 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:	0

			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)

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[2:0]	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 OC2REF 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 2 的值相同时，强制 OC2REF 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 2 相同时，强制 OC2REF 为低； 011：翻转。当核心计数器与比较捕获寄存器 2 的值相同时，翻转 OC2REF 的电平； 100：强制为无效电平。强制 OC2REF 为低； 101：强制为有效电平。强制 OC2REF 为高； 110：PWM 模式 1：在向上计数时，一旦核心计数器	000b

			小于比较捕获寄存器的值时，通道 2 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 2 为无效电平（OC2REF=0），否则为有效电平（OC2REF=1）； 111：PWM 模式 2：在向上计数时，一旦核心计数器小于比较捕获寄存器的值时，通道 2 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 2 为有效电平（OC2REF=1），否则为无效电平（OC2REF=0）； 注：一旦 LOCK 级别设为 3 并且 CC2S=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，OC2REF 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 2 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 2 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 CC2S=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 1 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。 注：OC2FE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。	0
[9:8]	CC2S[1:0]	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）。	00b

			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[2:0]	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[1:0]	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[3:0]	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 $F_{\text{sampling}} = F_{\text{dts}}/8$, N=6; 0001：采样频率 $F_{\text{sampling}} = F_{\text{ck\_int}}$, N=2; 1001：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/8$, N=8; 0010：采样频率 $F_{\text{sampling}} = F_{\text{ck\_int}}$, N=4; 1010：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/16$, N=5; 0011：采样频率 $F_{\text{sampling}} = f = F_{\text{ck\_int}}$, N=8; 1011：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/16$, N=6; 0100：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/2$, N=6; 1100：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/16$, N=8; 0101：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/2$, N=8; 1101：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/32$, N=5; 0110：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/4$, N=6; 1110：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/32$, N=6; 0111：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/4$, N=8; 1111：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/32$, N=8。	0000b
[11:10]	IC2PSC[1:0]	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。	00b
[9:8]	CC2S[1:0]	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 2 通道被配置为输出； 01：比较捕获通道 2 通道被配置为输入，IC1 映射在 TI1 上； 10：比较捕获通道 2 通道被配置为输入，IC1 映射在 TI2 上；	00b

			11: 比较捕获通道 2 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: CC2S 仅在通道关闭时 (CC2E 为 0) 才是可写的。	
[7:4]	IC1F[3:0]	RW	输入捕获滤波器 1 配置域。	0
[3:2]	IC1PSC[1:0]	RW	比较捕获通道 1 预分频配置域。	0
[1:0]	CC1S[1:0]	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)

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
OC4GE	OC4M[2:0]			OC4PE	OC4FE	GC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	GC3S[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[2:0]	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[1:0]	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[2:0]	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[1:0]	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[3:0]	RW	输入捕获滤波器 4 配置域。	0
[11:10]	IC4PSC[1:0]	RW	比较捕获通道 4 预分频配置域。	0
[9:8]	CC4S[1:0]	RW	比较捕获通道 4 输入选择域。	0
[7:4]	IC3F[3:0]	RW	输入捕获滤波器 3 配置域。	0
[3:2]	IC3PSC[1:0]	RW	比较捕获通道 3 预分频配置域。	0
[1:0]	CC3S[1:0]	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)

偏移地址: 0x20

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。 CC1 通道配置为输出： 1: OC1 低电平有效； 0: OC1 高电平有效。 CC1 通道配置为输入： 该位选择是 IC1 还是 IC1 的反相信号作为触发或捕获信号。 1: 反相：捕获发生在 IC1 的下降沿；当用作外部触发器时，IC1 反相； 0: 不反相：捕获发生在 IC1 的上升沿；当用作外部触发器时，IC1 不反相。 注：一旦 LOCK 级别（TIMx_BDTR 寄存器中的 LOCK 位）设为 3 或 2，则该位不能被修改。	0
0	CC1E	RW	比较捕获通道 1 输出使能位。 CC1 通道配置为输出： 1: 开启。OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值； 0: 关闭。OC1 禁止输出，因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。 CC1 通道配置为输入： 该位决定了计数器的值是否能捕获入 TIMx_CH1CVR 寄存器。 1: 捕获使能； 0: 捕获禁止。	0

14.4.10 高级定时器的计数器（TIMx_CNT）（x=1/8）

偏移地址：0x24

位	名称	访问	描述	复位值
[15:0]	CNT[15:0]	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）

偏移地址：0x28

位	名称	访问	描述	复位值
[15:0]	PSC[15:0]	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）

偏移地址：0x2C

位	名称	访问	描述	复位值
[15:0]	ARR[15:0]	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。	FFFFh

14.4.13 重复计数值寄存器（TIMx_RPTCR）（x=1/8）

偏移地址：0x30

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
[7:0]	REP[7:0]	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）

偏移地址：0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL1
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
[31:17]	Reserved	RO	保留。	0
16	LEVEL1	RO	捕获值对应的电平指示 bit。	0
[15:0]	CH1CVR[15:0]	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）

偏移地址：0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL2
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
[31:17]	Reserved	RO	保留。	0
16	LEVEL2	RO	捕获值对应的电平指示 bit。	0
[15:0]	CH2CVR[15:0]	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）

偏移地址：0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL3
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
[31:17]	Reserved	R0	保留。	0
16	LEVEL3	R0	捕获值对应的电平指示 bit。	0
[15:0]	CH3CVR[15:0]	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)

偏移地址：0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL4
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
[31:17]	Reserved	R0	保留。	0
16	LEVEL4	R0	捕获值对应的电平指示 bit。	0
[15:0]	CH4CVR[15:0]	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)

偏移地址：0x44

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 HB 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	0

			该位的写需要一个 HB 时钟以后才能生效。	
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[1:0]	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。	00b
[7:0]	DTG[7:0]	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。 注: 一旦 LOCK 级别 (TIMx_BDTR 寄存器中的 LOCK[1:0]位) 设为 01b、10b 或 11b, 则不能修改这些位。	0

14.4.19 DMA 控制寄存器 (TIMx_DMACFGR) (x=1/8)

偏移地址: 0x48

位	名称	访问	描述	复位值
[15:13]	Reserved	RO	保留。	0
[12:8]	DBL[4:0]	RW	DMA 连续传送的长度, 实际值为此域的值+1。 当对 TIMx_DMAADR 寄存器进行读或写时, 定时器则进行一次连续传送), 即: 定义传输的次数, 传输可以是半字 (双字节) 或字节: 00000: 1 次传输; 00001: 2 次传输; 00010: 3 次传输;	0

			 10001: 18 次传输。 假如，我们进行这样的传输：DBL=7，DBA=TIM2_CTLR1。 如果 DBL=7，DBA=TIM2_CTLR1 表示待传输数据的地址，那么传输的地址由以下公式给出：(TIMx_CTLR1 的地址)+DBA+(DMA 索引)，其中 DMA 索引=DBL，其中 (TIMx_CTLR1 的地址)+DBA 再加上 7，给出了将要写入或者读出数据的地址，这样数据的传输将发生在从地址 (TIMx_CR1 的地址)+DBA 开始的 7 个寄存器。根据 DMA 数据长度的设置，可能发生以下情况： 1、如果设置数据为半字（16 位），那么数据就会传输给全部 7 个寄存器。 2、如果设置数据为字节，数据仍然会传输给全部 7 个寄存器：第一个寄存器包含第一个 MSB 字节，第二个寄存器包含第一个 LSB 字节，依此类推。因此对于定时器，用户必须指定由 DMA 传输的数据宽度。	
[7:5]	Reserved	RO	保留。	0
[4:0]	DBA[4:0]	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。 00000: TIMx_CTLR1; 00001: TIMx_CTLR2; 00010: TIMx_SMCGR; 	0

14.4.20 连续模式的 DMA 地址寄存器 (TIMx_DMAADR) (x=1/8)

偏移地址：0x4C

位	名称	访问	描述	复位值
[15:0]	DMAB[15:0]	RW	连续模式下，DMA 的地址。 对 TIMx_DMAADR 寄存器的读或写会导致对以下地址所在寄存器的存取操作：TIMx_CTLR1 地址+DBA+ DMA 索引，其中：“TIMx_CTLR1 地址”是控制寄存器 1 (TIMx_CTLR1) 所在的地址；“DBA”是 TIMx_DMACHGR 寄存器中定义的基地址；“DMA 索引”是由 DMA 自动控制的偏移量，它取决于 TIMx_DMACHGR 寄存器中定义的 DBL。	0

14.4.21 双边沿捕获寄存器 (TIMx_AUX) (x=1/8)

偏移地址：0x50

位	名称	访问	描述	复位值
[15:8]	DT_VLU2	RW	实现死区时间不对称功能时，设置时间小于 DTG 的死区持续时间，需配合 DTP_MODE 和 DTN_MODE 位使用。	0

			注：设置的死区持续时间为设置值乘 DTG 中设置的 $Tdtg$ ，其需大于 0 小于 DTG 设置的死区时间。	
7	DTN_MODE	RW	使能 DT_VLU2 设置的死区持续时间发生在 OCXREF 的下降沿。 1:DT_VLU2 设置的死区持续时间发生在 OCXREF 的下降沿； 0：功能关闭，OCXREF 的下降沿死区持续时间为原有 DTG 设置时间。	0
6	DTP_MODE	RW	使能 DT_VLU2 设置的死区持续时间发生在 OCXREF 的上升沿。 1:DT_VLU2 设置的死区持续时间发生在 OCXREF 的上升沿； 0：功能关闭，OCXREF 的上升沿死区持续时间为原有 DTG 设置时间。	0
[5:3]	BK_SEL	RW	刹车源选择位： 000: BKIN； 001: BKIN2； 01x: BKIN 或 BKIN2； 100: DFSDM_BK[0]； 101: DFSDM_BK[1]； 其他：保留。	0
2	CAP_ED_CH4	RW	通道 4 的双边沿捕获使能： 1：使能通道 4 的双边沿捕获使能； 0：关闭通道 4 的双边沿捕获使能。	0
1	CAP_ED_CH3	RW	通道 3 的双边沿捕获使能： 1：使能通道 3 的双边沿捕获使能； 0：关闭通道 3 的双边沿捕获使能。	0
0	CAP_ED_CH2	RW	通道 2 的双边沿捕获使能： 1：使能通道 2 的双边沿捕获使能； 0：关闭通道 2 的双边沿捕获使能。	0

第 15 章 通用定时器（GPTM）

通用定时器模块包含 4 个 16 位可自动重装的定时器（TIM2、TIM3、TIM4 和 TIM5）和 4 个 32 位可自动重装的定时器（TIM9、TIM10、TIM11 和 TIM12），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

16 位通用定时器的主要特征包括：

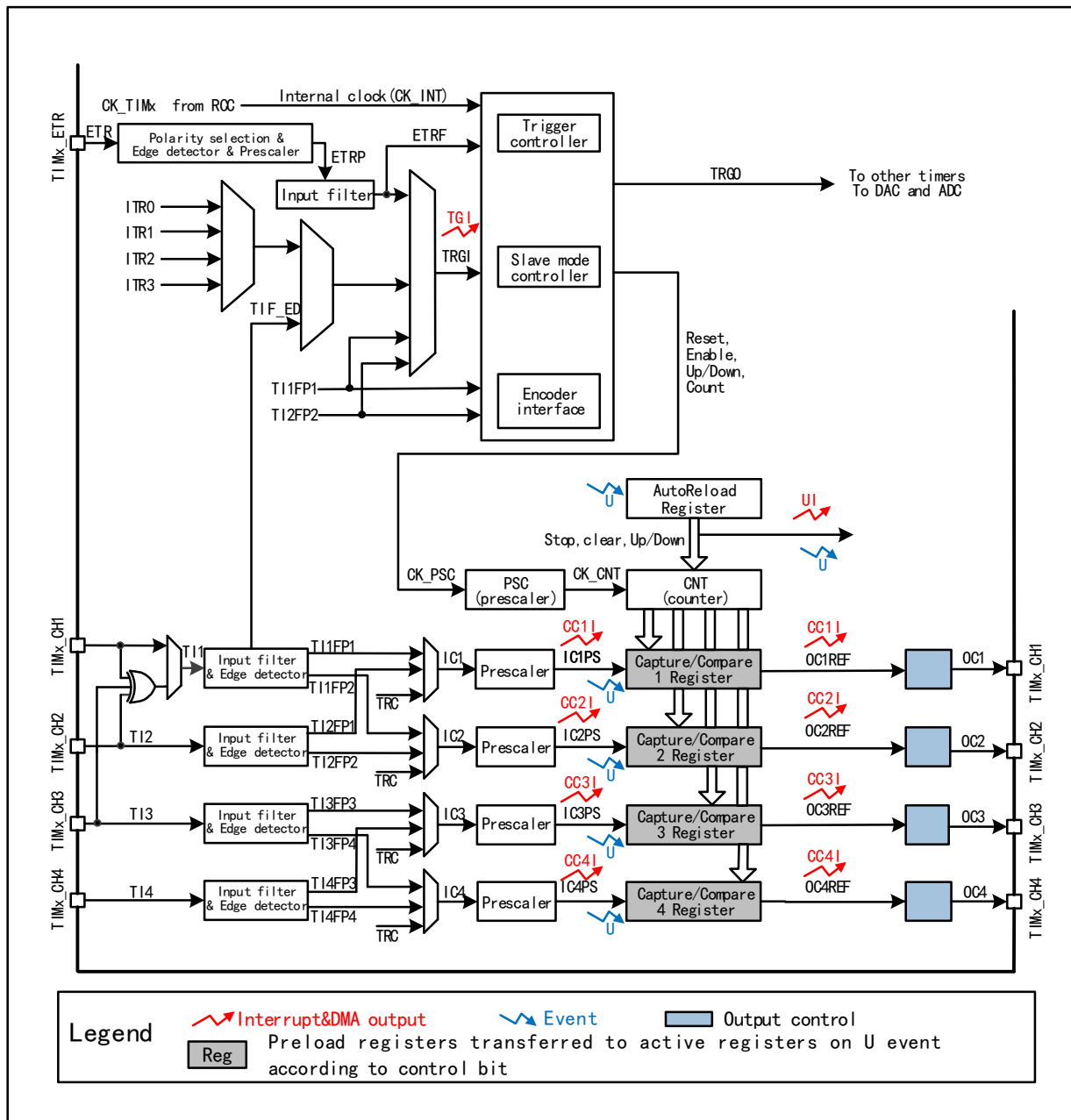
- 16 位自动重装计数器，支持增计数模式，减计数模式和增减计数模式
- 16 位预分频器，分频系数从 1~65536 之间动态可调
- 支持四路独立的比较捕获通道
- 每路比较捕获通道支持多种工作模式，比如：输入捕获、输出比较、PWM 生成和单脉冲输出
- 支持外部信号控制定时器
- 支持在多种模式下使用 DMA
- 支持增量式编码，定时器之间的级联和同步

32 位通用定时器的主要特征包括：

- 32 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 HB 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 位或 32 位的计数器（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 重装初始值。

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

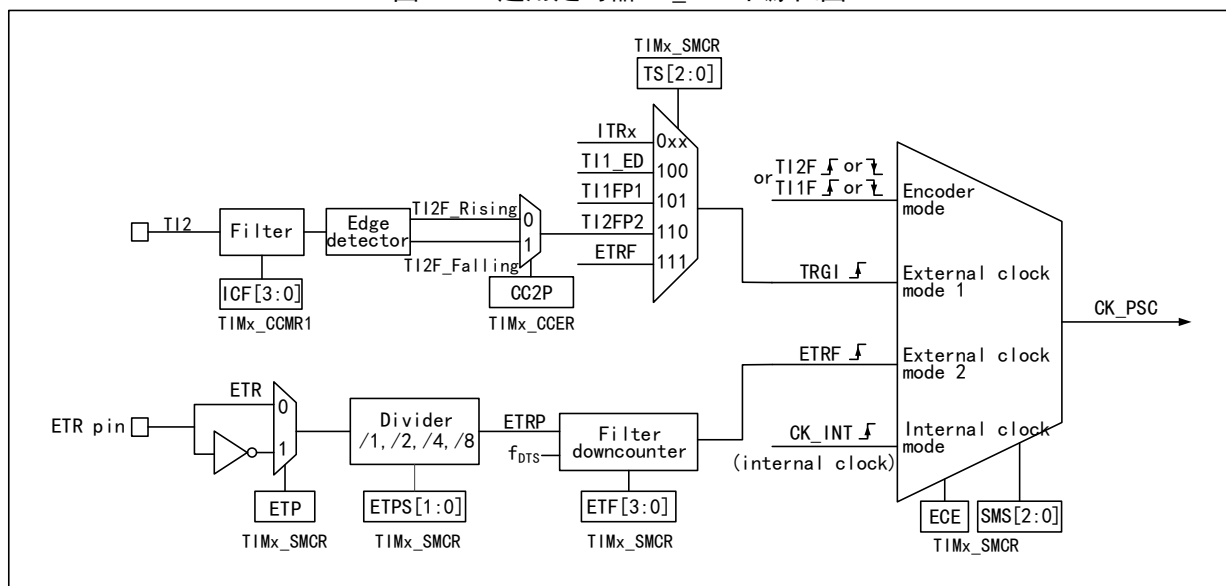
与高级定时器相比，通用定时器缺少以下功能：

- 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。

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 HB 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→Tl_x→Tl_xFP_x, 此路线也用于编码器模式;
- 4) 来自内部其他定时器的输入: ITR_x。

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 经过边沿检测器后的信号 (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 15.3.7 节。

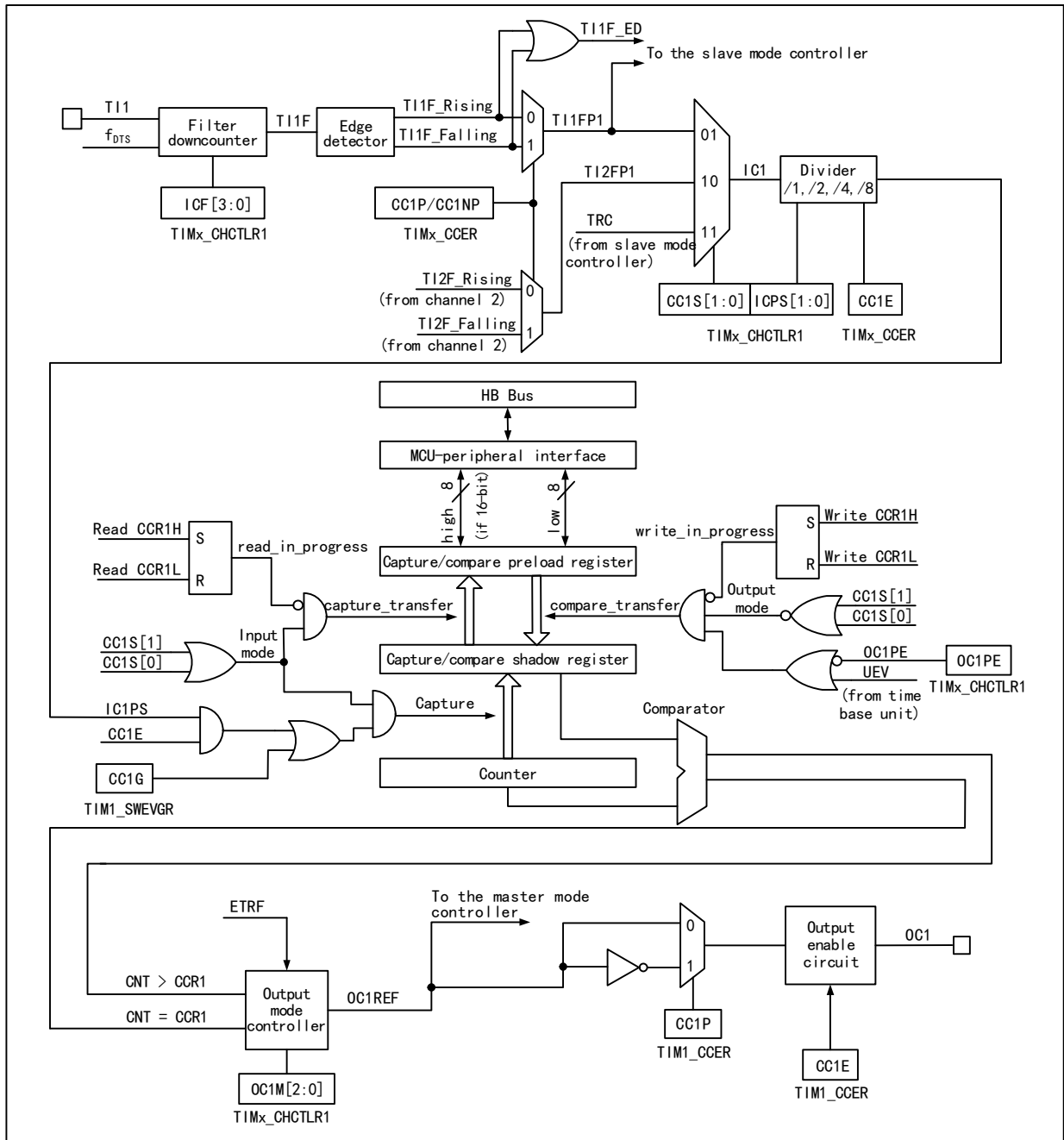
15.2.4 计数器和周边

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_CNT。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。并有一个自动重装值寄存器 (ATRLR) 可以设置计数器阈值，当 CNT 计数值达到自动重装值寄存器 (ATRLR) 里设置的计数器阈值时，会将 CNT 清零。

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 15-1)，以 $TI1$ 举例： $TI1$ 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 CC1S 域，选择 IC1 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，CC1S 域默认是使比较捕获模块作为输出通道；
- 2) 配置 IC1F 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。
- 3) 配置 CC1P 位，设定 TI1FP1 的极性。比如保持 CC1P 位为低，选择上升沿跳变；
- 4) 配置 IC1PS 域，设定 IC1 信号成为 IC1PS 之间的分频系数。比如保持 IC1PS 为 00b，不分频；
- 5) 配置 CC1E 位，允许捕获核心计数器（CNT）的值到比较捕获寄存器中。
- 6) 根据是否需要中断或者 DMA 来配置 CC1IE 和 CC1DE 位。

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。

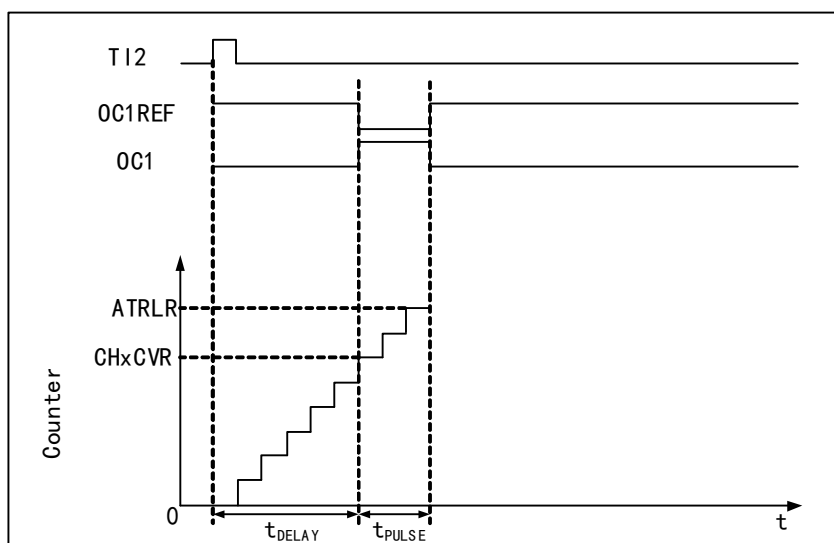
● 中央对齐

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

计数有效边沿	相对信号的电平	TI1FP1 信号边沿		TI2FP2 信号	
		上升沿	下降沿	上升沿	下降沿
仅在 TI1 边沿计数	高	向下计数	向上计数	不计数	
	低	向上计数	向下计数		
仅在 TI2 边沿计数	高	不计数		向上计数	向下计数
	低			向下计数	向上计数
在 TI1 和 TI2 双边沿计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

15.3.8 定时器同步模式

定时器能够输出时钟脉冲（TRG0），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 TRG0）是不一样的。定时器内部触发连接如表 15-2 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
TIM9	TIM1	TIM2	TIM10	TIM11
TIM10	TIM1	TIM8	TIM2	TIM9
TIM11	TIM1	TIM2	TIM9	TIM12
TIM12	TIM1	TIM2	TIM9	TIM11

15.3.9 双边沿捕获模式

可通过 TIMx_AUX 寄存器 CAP_ED_CHx 位开启对应通道双边沿捕获对脉冲进行测量。

例如：通过通道 2 来捕获脉冲宽度，在捕获功能配置上，选择 IC2 信号的来源（TIMx_CHCTLR1 寄存器的 CC2S 位为 11b），使能通道 2 的双边沿捕获功能（TIMx_AUX 寄存器的 CAP_ED_CH2 位为 1）。至此已经将双边沿捕获配置完成。

可通过 TIMx_CH2CVR 寄存器的 CH2CVR 位读取捕获的脉冲的高低电平宽度值，开启 CAPLVL 位可通过 TIMx_CH2CVR 寄存器 bit[16] 指示捕获值对应的电平。

注：此功能仅适用于 TIM2/TIM3/TIM4/TIM5。

15.3.10 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_SMCFR	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	0xFFFF
R32_TIM2_CH1CVR	0x40000034	TIM2 比较/捕获寄存器 1	0x0000
R32_TIM2_CH2CVR	0x40000038	TIM2 比较/捕获寄存器 2	0x0000
R32_TIM2_CH3CVR	0x4000003C	TIM2 比较/捕获寄存器 3	0x0000
R32_TIM2_CH4CVR	0x40000040	TIM2 比较/捕获寄存器 4	0x0000
R16_TIM2_DMACFR	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
R16_TIM2_AUX	0x40000050	TIM2 双边沿捕获寄存器	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_SMCFR	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	0xFFFF
R32_TIM3_CH1CVR	0x40000434	TIM3 比较/捕获寄存器 1	0x0000
R32_TIM3_CH2CVR	0x40000438	TIM3 比较/捕获寄存器 2	0x0000

R32_TIM3_CH3CVR	0x4000043C	TIM3 比较/捕获寄存器 3	0x0000
R32_TIM3_CH4CVR	0x40000440	TIM3 比较/捕获寄存器 4	0x0000
R16_TIM3_DMACHFR	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
R16_TIM3_AUX	0x40000450	TIM3 双边沿捕获寄存器	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_CHCTL1	0x40000818	TIM4 比较/捕获控制寄存器 1	0x0000
R16_TIM4_CHCTL2	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	0xFFFF
R32_TIM4_CH1CVR	0x40000834	TIM4 比较/捕获寄存器 1	0x0000
R32_TIM4_CH2CVR	0x40000838	TIM4 比较/捕获寄存器 2	0x0000
R32_TIM4_CH3CVR	0x4000083C	TIM4 比较/捕获寄存器 3	0x0000
R32_TIM4_CH4CVR	0x40000840	TIM4 比较/捕获寄存器 4	0x0000
R16_TIM4_DMACHFR	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
R16_TIM4_AUX	0x40000850	TIM4 双边沿捕获寄存器	0x0000

表 15-6 TIM5 相关寄存器列表

名称	偏移地址	描述	复位值
R16_TIM5_CTLR1	0x40000C00	TIM5 控制寄存器 1	0x0000
R16_TIM5_CTLR2	0x40000C04	TIM5 控制寄存器 2	0x0000
R16_TIM5_SMCFGR	0x40000C08	TIM5 从模式控制寄存器	0x0000
R16_TIM5_DMAINTENR	0x40000C0C	TIM5 DMA/中断使能寄存器	0x0000
R16_TIM5_INTFR	0x40000C10	TIM5 中断状态寄存器	0x0000
R16_TIM5_SWEVGR	0x40000C14	TIM5 事件产生寄存器	0x0000
R16_TIM5_CHCTL1	0x40000C18	TIM5 比较/捕获控制寄存器 1	0x0000
R16_TIM5_CHCTL2	0x40000C1C	TIM5 比较/捕获控制寄存器 2	0x0000
R16_TIM5_CCER	0x40000C20	TIM5 比较/捕获使能寄存器	0x0000
R16_TIM5_CNT	0x40000C24	TIM5 计数器	0x0000
R16_TIM5_PSC	0x40000C28	TIM5 计数时钟预分频器	0x0000
R16_TIM5_ATRLR	0x40000C2C	TIM5 自动重装值寄存器	0xFFFF
R32_TIM5_CH1CVR	0x40000C34	TIM5 比较/捕获寄存器 1	0x0000
R32_TIM5_CH2CVR	0x40000C38	TIM5 比较/捕获寄存器 2	0x0000

R32_TIM5_CH3CVR	0x40000C3C	TIM5 比较/捕获寄存器 3	0x0000
R32_TIM5_CH4CVR	0x40000C40	TIM5 比较/捕获寄存器 4	0x0000
R16_TIM5_DMACHFR	0x40000C48	TIM5 DMA 控制寄存器	0x0000
R16_TIM5_DMAADR	0x40000C4C	TIM5 连续模式的 DMA 地址寄存器	0x0000
R16_TIM5_AUX	0x40000C50	TIM5 双边沿捕获寄存器	0x0000

表 15-7 TIM9 相关寄存器列表

名称	偏移地址	描述	复位值
R16_TIM9_CTLR1	0x40014C00	TIM9 控制寄存器 1	0x0000
R16_TIM9_CTLR2	0x40014C04	TIM9 控制寄存器 2	0x0000
R16_TIM9_SMCFR	0x40014C08	TIM9 从模式控制寄存器	0x0000
R16_TIM9_DMAINTENR	0x40014C0C	TIM9 DMA/中断使能寄存器	0x0000
R16_TIM9_INTFR	0x40014C10	TIM9 中断状态寄存器	0x0000
R16_TIM9_SWEVGR	0x40014C14	TIM9 事件产生寄存器	0x0000
R16_TIM9_CHCTLR1	0x40014C18	TIM9 比较/捕获控制寄存器 1	0x0000
R16_TIM9_CHCTLR2	0x40014C1C	TIM9 比较/捕获控制寄存器 2	0x0000
R16_TIM9_CCER	0x40014C20	TIM9 比较/捕获使能寄存器	0x0000
R32_TIM9_CNT	0x40014C24	TIM9 计数器	0x00000000
R16_TIM9_PSC	0x40014C28	TIM9 计数时钟预分频器	0x0000
R32_TIM9_ATRLR	0x40014C2C	TIM9 自动重装值寄存器	0x0000FFFF
R32_TIM9_CH1CVR	0x40014C34	TIM9 比较/捕获寄存器 1	0x00000000
R32_TIM9_CH2CVR	0x40014C38	TIM9 比较/捕获寄存器 2	0x00000000
R32_TIM9_CH3CVR	0x40014C3C	TIM9 比较/捕获寄存器 3	0x00000000
R32_TIM9_CH4CVR	0x40014C40	TIM9 比较/捕获寄存器 4	0x00000000
R16_TIM9_DMACHFR	0x40014C48	TIM9 DMA 控制寄存器	0x0000
R16_TIM9_DMAADR	0x40014C4C	TIM9 连续模式的 DMA 地址寄存器	0x0000

表 15-8 TIM10 相关寄存器列表

名称	偏移地址	描述	复位值
R16_TIM10_CTLR1	0x40015000	TIM10 控制寄存器 1	0x0000
R16_TIM10_CTLR2	0x40015004	TIM10 控制寄存器 2	0x0000
R16_TIM10_SMCFR	0x40015008	TIM10 从模式控制寄存器	0x0000
R16_TIM10_DMAINTENR	0x4001500C	TIM10 DMA/中断使能寄存器	0x0000
R16_TIM10_INTFR	0x40015010	TIM10 中断状态寄存器	0x0000
R16_TIM10_SWEVGR	0x40015014	TIM10 事件产生寄存器	0x0000
R16_TIM10_CHCTLR1	0x40015018	TIM10 比较/捕获控制寄存器 1	0x0000
R16_TIM10_CHCTLR2	0x4001501C	TIM10 比较/捕获控制寄存器 2	0x0000
R16_TIM10_CCER	0x40015020	TIM10 比较/捕获使能寄存器	0x0000
R32_TIM10_CNT	0x40015024	TIM10 计数器	0x00000000
R16_TIM10_PSC	0x40015028	TIM10 计数时钟预分频器	0x0000
R32_TIM10_ATRLR	0x4001502C	TIM10 自动重装值寄存器	0x0000FFFF
R32_TIM10_CH1CVR	0x40015034	TIM10 比较/捕获寄存器 1	0x00000000
R32_TIM10_CH2CVR	0x40015038	TIM10 比较/捕获寄存器 2	0x00000000
R32_TIM10_CH3CVR	0x4001503C	TIM10 比较/捕获寄存器 3	0x00000000

R32_TIM10_CH4CVR	0x40015040	TIM10 比较/捕获寄存器 4	0x00000000
R16_TIM10_DMACHFR	0x40015048	TIM10 DMA 控制寄存器	0x0000
R16_TIM10_DMAADR	0x4001504C	TIM10 连续模式的 DMA 地址寄存器	0x0000

表 15-9 TIM11 相关寄存器列表

名称	偏移地址	描述	复位值
R16_TIM11_CTLR1	0x40015400	TIM11 控制寄存器 1	0x0000
R16_TIM11_CTLR2	0x40015404	TIM11 控制寄存器 2	0x0000
R16_TIM11_SMCFR	0x40015408	TIM11 从模式控制寄存器	0x0000
R16_TIM11_DMAINTENR	0x4001540C	TIM11 DMA/中断使能寄存器	0x0000
R16_TIM11_INTFR	0x40015410	TIM11 中断状态寄存器	0x0000
R16_TIM11_SWEVGR	0x40015414	TIM11 事件产生寄存器	0x0000
R16_TIM11_CHCTLR1	0x40015418	TIM11 比较/捕获控制寄存器 1	0x0000
R16_TIM11_CHCTLR2	0x4001541C	TIM11 比较/捕获控制寄存器 2	0x0000
R16_TIM11_CCER	0x40015420	TIM11 比较/捕获使能寄存器	0x0000
R32_TIM11_CNT	0x40015424	TIM11 计数器	0x00000000
R16_TIM11_PSC	0x40015428	TIM11 计数时钟预分频器	0x0000
R32_TIM11_ATRLR	0x4001542C	TIM11 自动重装值寄存器	0x0000FFFF
R32_TIM11_CH1CVR	0x40015434	TIM11 比较/捕获寄存器 1	0x00000000
R32_TIM11_CH2CVR	0x40015438	TIM11 比较/捕获寄存器 2	0x00000000
R32_TIM11_CH3CVR	0x4001543C	TIM11 比较/捕获寄存器 3	0x00000000
R32_TIM11_CH4CVR	0x40015440	TIM11 比较/捕获寄存器 4	0x00000000
R16_TIM11_DMACHFR	0x40015448	TIM11 DMA 控制寄存器	0x0000
R16_TIM11_DMAADR	0x4001544C	TIM11 连续模式的 DMA 地址寄存器	0x0000

表 15-10 TIM12 相关寄存器列表

名称	偏移地址	描述	复位值
R16_TIM12_CTLR1	0x40013C00	TIM12 控制寄存器 1	0x0000
R16_TIM12_CTLR2	0x40013C04	TIM12 控制寄存器 2	0x0000
R16_TIM12_SMCFR	0x40013C08	TIM12 从模式控制寄存器	0x0000
R16_TIM12_DMAINTENR	0x40013C0C	TIM12 DMA/中断使能寄存器	0x0000
R16_TIM12_INTFR	0x40013C10	TIM12 中断状态寄存器	0x0000
R16_TIM12_SWEVGR	0x40013C14	TIM12 事件产生寄存器	0x0000
R16_TIM12_CHCTLR1	0x40013C18	TIM12 比较/捕获控制寄存器 1	0x0000
R16_TIM12_CHCTLR2	0x40013C1C	TIM12 比较/捕获控制寄存器 2	0x0000
R16_TIM12_CCER	0x40013C20	TIM12 比较/捕获使能寄存器	0x0000
R32_TIM12_CNT	0x40013C24	TIM12 计数器	0x00000000
R16_TIM12_PSC	0x40013C28	TIM12 计数时钟预分频器	0x0000
R32_TIM12_ATRLR	0x40013C2C	TIM12 自动重装值寄存器	0x0000FFFF
R32_TIM12_CH1CVR	0x40013C34	TIM12 比较/捕获寄存器 1	0x00000000
R32_TIM12_CH2CVR	0x40013C38	TIM12 比较/捕获寄存器 2	0x00000000
R32_TIM12_CH3CVR	0x40013C3C	TIM12 比较/捕获寄存器 3	0x00000000
R32_TIM12_CH4CVR	0x40013C40	TIM12 比较/捕获寄存器 4	0x00000000
R16_TIM12_DMACHFR	0x40013C48	TIM12 DMA 控制寄存器	0x0000

R16_TIM12_DMAADR	0x40013C4C	TIM12 连续模式的 DMA 地址寄存器	0x0000

15.4.1 控制寄存器 1 (TIMx_CTLR1) (x=2/3/4/5/9/10/11/12)

偏移地址: 0x00

位	名称	访问	描述	复位值
15	CAPLVL	RW	双沿捕获模式下, 捕获电平指示使能。 1: 使能指示功能; 0: 关闭指示功能。 注: 使能后, CHxCVR 的[16]指示捕获值对应的电平。	0
14	CAPOV	RW	捕获值模式配置。 1: 当捕获前产生计数器溢出时, CHxCVR 值为 0xFFFF; 0: 捕获值为实际计数器的值。	0
[13:10]	Reserved	RO	保留。	0
[9:8]	CKD[1:0]	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。	00b
7	ARPE	RW	自动重装预装使能位: 0: 禁止自动重装值寄存器 (ATRLR); 1: 使能自动重装值寄存器 (ATRLR)。	0
[6:5]	CMS[1:0]	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。	00b
4	DIR	RW	计数器方向: 1: 计数器的计数模式为减计数; 0: 计数器的计数模式为增计数。 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: 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。 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=2/3/4/5/9/10/11/12）

偏移地址：0x04

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[2:0]	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；	000b

			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_SMCGR 寄存器中 MSM 位的描述); 010: 更新事件被选为触发输出 (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: 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
[2:1]	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_SMCGR) (x=2/3/4/5/9/10/11/12)

偏移地址: 0x08

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[1:0]	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;	00b

			10: ETRP 频率除以 4; 11: ETRP 频率除以 8。	
[11:8]	ETF[3:0]	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; 1111: 采样频率 Fsampling=Fdts/32, N=8。	0000b
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[2:0]	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。	000b
3	Reserved	R0	保留。	0
[2:0]	SMS[2:0]	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;	000b

			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。	

15.4.4 DMA/中断使能寄存器 (TIMx_DMAINTENR) (x=2/3/4/5/9/10/11/12)

偏移地址: 0x0C

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	Reserved	RO	保留。	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: 1: 允许比较捕获通道 3 中断;	0

			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=2/3/4/5/9/10/11/12)

偏移地址: 0x10

位	名称	访问	描述	复位值
[15:13]	Reserved	RO	保留。	0
12	CC40F	RWO	比较捕获通道 4 重复捕获标志位。	0
11	CC30F	RWO	比较捕获通道 3 重复捕获标志位。	0
10	CC20F	RWO	比较捕获通道 2 重复捕获标志位。	0
9	CC10F	RWO	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]	Reserved	RO	保留。	0
6	TIF	RWO	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	Reserved	RO	保留。	0
4	CC4IF	RWO	比较捕获通道 4 中断标志位。	0
3	CC3IF	RWO	比较捕获通道 3 中断标志位。	0
2	CC2IF	RWO	比较捕获通道 2 中断标志位。	0
1	CC1IF	RWO	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, 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。	0

			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

15.4.6 事件产生寄存器 (TIMx_SWEVGR) (x=2/3/4/5/9/10/11/12)

偏移地址: 0x14

位	名称	访问	描述	复位值
[15:7]	Reserved	R0	保留。	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	Reserved	R0	保留。	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: 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 CC10F 位。 0: 无动作;	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

15.4.7 比较/捕获控制寄存器 1 (TIMx_CHCTLR1) (x=2/3/4/5/9/10/11/12)

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； 0：OC2REF 不受 ETRF 输入的影响。	0
[14:12]	OC2M[2:0]	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 OC2REF 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 2 的值相同时，强制 OC2REF 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 2 相同时，强制 OC2REF 为低； 011：翻转。当核心计数器与比较捕获寄存器 2 的值相同时，翻转 OC2REF 的电平； 100：强制为无效电平。强制 OC2REF 为低； 101：强制为有效电平。强制 OC2REF 为高； 110：PWM 模式 1：在向上计数时，一旦核心计数器小于比较捕获寄存器的值时，通道 2 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 2 为无效电平（OC2REF=0），否则为有效电平（OC2REF=1）； 111：PWM 模式 2：在向上计数时，一旦核心计数器小于比较捕获寄存器的值时，通道 2 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 2 为有效电平（OC2REF=1），否则为无效电平（OC2REF=0）。 注：一旦 LOCK 级别设为 3 并且 CC2S=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	000b

			换到 PWM 模式时, OC2REF 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 2 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 2 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 CG2S=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 1 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。 注: OC2FE 只在通道被配置成 PWM1 或 PWM2 模式时起作用。	0
[9:8]	CC2S[1:0]	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)。 注: 比较捕获通道 2 仅在通道关闭时 (CC2E 为零时) 才是可写的。	00b
7	OC1CE	RW	比较捕获通道 1 清零使能位。	0
[6:4]	OC1M[2:0]	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[1:0]	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[3:0]	RW	输入捕获滤波器 2 配置域, 这几位设置了 TI1 输入	0000b

			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 $F_{\text{sampling}} = F_{\text{dts}}/8$，N=6； 0001：采样频率 $F_{\text{sampling}} = F_{\text{ck\_int}}$，N=2； 1001：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/8$，N=8； 0010：采样频率 $F_{\text{sampling}} = F_{\text{ck\_int}}$，N=4； 1010：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/16$，N=5； 0011：采样频率 $F_{\text{sampling}} = f = F_{\text{ck\_int}}$，N=8； 1011：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/16$，N=6； 0100：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/2$，N=6； 1100：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/16$，N=8； 0101：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/2$，N=8； 1101：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/32$，N=5； 0110：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/4$，N=6； 1110：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/32$，N=6； 0111：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/4$，N=8； 1111：采样频率 $F_{\text{sampling}} = F_{\text{dts}}/32$，N=8。	
[11:10]	IC2PSC[1:0]	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。	00b
[9:8]	CC2S[1:0]	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时（由 TS 位选择）。 注：CC1S 仅在通道关闭时（CC1E 为 0）才是可写的。	00b
[7:4]	IC1F[3:0]	RW	输入捕获滤波器 1 配置域。	0
[3:2]	IC1PSC[1:0]	RW	比较捕获通道 1 预分频配置域。	0
[1:0]	CC1S[1:0]	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=2/3/4/5/9/10/11/12)

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[2:0]	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[1:0]	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[2:0]	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[1:0]	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[3:0]	RW	输入捕获滤波器 4 配置域。	0
[11:10]	IC4PSC[1:0]	RW	比较捕获通道 4 预分频配置域。	0
[9:8]	CC4S[1:0]	RW	比较捕获通道 4 输入选择域。	0
[7:4]	IC3F[3:0]	RW	输入捕获滤波器 3 配置域。	0
[3:2]	IC3PSC[1:0]	RW	比较捕获通道 3 预分频配置域。	0
[1:0]	CC3S[1:0]	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=2/3/4/5/9/10/11/12）

偏移地址：0x20

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]	Reserved	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]	Reserved	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]	Reserved	RO	保留。	0
1	CC1P	RW	比较捕获通道 1 输出极性设置位。 CC1 通道配置为输出： 1：OC1 低电平有效；	0

			0: 0C1 高电平有效。 CC1 通道配置为输入： 该位选择是 IC1 还是 IC1 的反相信号作为触发或捕获信号。 1: 反相：捕获发生在 IC1 的下降沿；当用作外部触发器时，IC1 反相； 0: 不反相：捕获发生在 IC1 的上升沿；当用作外部触发器时，IC1 不反相。	
0	CC1E	RW	比较捕获通道 1 输出使能位。 CC1 通道配置为输出： 1: 开启：0C1 信号输出到对应的输出引脚； 0: 关闭：0C1 禁止输出。 CC1 通道配置为输入： 该位决定了计数器的值是否能捕获入 TIMx_CH1CVR 寄存器。 1: 捕获使能； 0: 捕获禁止。	0

15.4.10 通用定时器的计数器 (TIMx_CNT) (x=2/3/4/5)

偏移地址：0x24

位	名称	访问	描述	复位值
[15:0]	CNT[15:0]	RW	定时器的计数器的实时值。	0

15.4.11 通用定时器的计数器 (TIMx_CNT) (x=9/10/11/12)

偏移地址：0x24

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]	CNT[31:0]	RW	定时器的计数器的实时值。	0

15.4.12 计数时钟预分频器 (TIMx_PSC) (x=2/3/4/5/9/10/11/12)

偏移地址：0x28

位	名称	访问	描述	复位值
[15:0]	PSC[15:0]	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

15.4.13 自动重装值寄存器 (TIMx_ATRLR) (x=2/3/4/5)

偏移地址：0x2C

位	名称	访问	描述	复位值
[15:0]	ARR[15:0]	RW	计数器阈值，ATRLR 何时动作和更新请阅读 15.2.4 节；ATRLR 为空时，计数器停止。	0xFFFF

15.4.14 自动重装值寄存器 (TIMx_ATRLR) (x=9/10/11/12)

偏移地址: 0x2C

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

位	名称	访问	描述	复位值
[31:0]	ARR[31:0]	RW	计数器阈值, ATRLR 何时动作和更新请阅读 15.2.4 节; ATRLR 为空时, 计数器停止。	0xFFFF FFFF

15.4.15 比较/捕获寄存器 1 (TIMx_CH1CVR) (x=2/3/4/5)

偏移地址: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL1
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
[31:17]	Reserved	RO	保留。	0
16	LEVEL1	RO	捕获值对应的电平指示 bit。	0
[15:0]	CH1CVR[15:0]	RW	比较捕获寄存器通道 1 的值。	0

15.4.16 比较/捕获寄存器 1 (TIMx_CH1CVR) (x=9/10/11/12)

偏移地址: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1CVR[31:16]															
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
[31:0]	CH1CVR[31:0]	RW	比较捕获寄存器通道 1 的值。	0

15.4.17 比较/捕获寄存器 2 (TIMx_CH2CVR) (x=2/3/4/5)

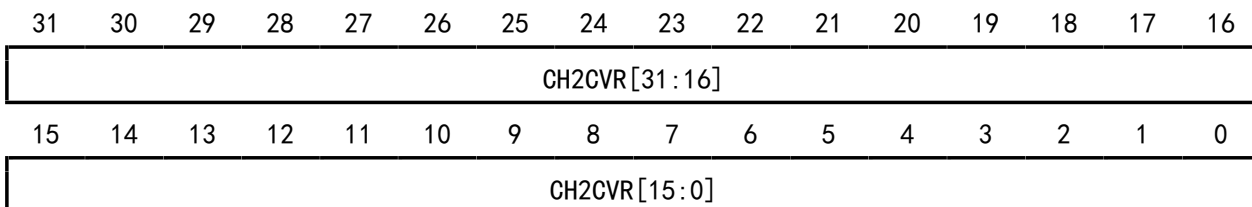
偏移地址: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL2
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
[31:17]	Reserved	R0	保留。	0
16	LEVEL2	R0	捕获值对应的电平指示 bit。	0
[15:0]	CH2CVR[15:0]	RW	比较捕获寄存器通道 2 的值。	0

15.4.18 比较/捕获寄存器 2 (TIMx_CH2CVR) (x=9/10/11/12)

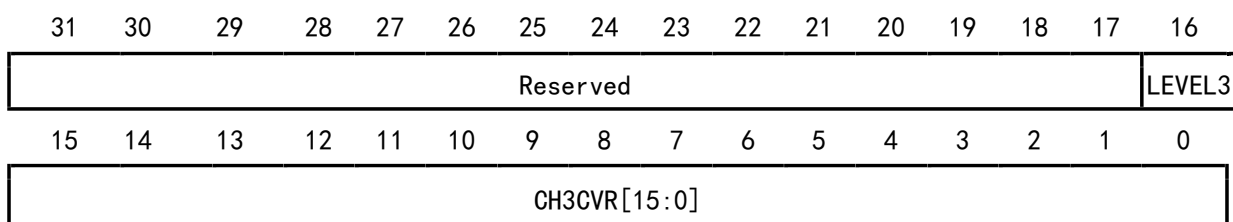
偏移地址: 0x38



位	名称	访问	描述	复位值
[31:0]	CH2CVR[31:0]	RW	比较捕获寄存器通道 2 的值。	0

15.4.19 比较/捕获寄存器 3 (TIMx_CH3CVR) (x=2/3/4/5)

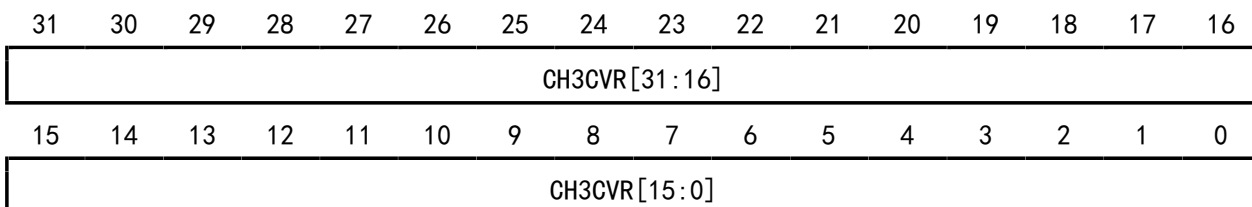
偏移地址: 0x3C



位	名称	访问	描述	复位值
[31:17]	Reserved	R0	保留。	0
16	LEVEL3	R0	捕获值对应的电平指示 bit。	0
[15:0]	CH3CVR[15:0]	RW	比较捕获寄存器通道 3 的值。	0

15.4.20 比较/捕获寄存器 3 (TIMx_CH3CVR) (x=9/10/11/12)

偏移地址: 0x3C



位	名称	访问	描述	复位值
[31:0]	CH3CVR[31:0]	RW	比较捕获寄存器通道 3 的值。	0

15.4.21 比较/捕获寄存器 4 (TIMx_CH4CVR) (x=2/3/4/5)

偏移地址: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL4
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
[31:17]	Reserved	R0	保留。	0
16	LEVEL4	R0	捕获值对应的电平指示 bit。	0
[15:0]	CH4CVR[15:0]	RW	比较捕获寄存器通道 4 的值。	0

15.4.22 比较/捕获寄存器 4 (TIMx_CH4CVR) (x=9/10/11/12)

偏移地址: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH4CVR[31:16]															
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
[31:0]	CH4CVR[31:0]	RW	比较捕获寄存器通道 4 的值。	0

15.4.23 DMA 控制寄存器 (TIMx_DMACHFR) (x=2/3/4/5/9/10/11/12)

偏移地址: 0x48

位	名称	访问	描述	复位值
[15:13]	Reserved	R0	保留。	0
[12:8]	DBL[4:0]	RW	DMA 连续传送的长度, 实际值为此域的值+1。	0
[7:5]	Reserved	R0	保留。	0
[4:0]	DBA[4:0]	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

15.4.24 连续模式的 DMA 地址寄存器 (TIMx_DMAADR) (x=2/3/4/5/9/10/11/12)

偏移地址: 0x4C

位	名称	访问	描述	复位值
[15:0]	DMAB[15:0]	RW	连续模式下, DMA 的地址。	0

15.4.25 双边沿捕获寄存器 (TIMx_AUX) (x=2/3/4/5)

偏移地址: 0x50

位	名称	访问	描述	复位值
[15:3]	Reserved	R0	保留。	0
2	CAP_ED_CH4	RW	通道 4 的双边沿捕获使能: 1: 使能通道 4 的双边沿捕获使能; 0: 关闭通道 4 的双边沿捕获使能。	0
1	CAP_ED_CH3	RW	通道 3 的双边沿捕获使能:	0

			1：使能通道 3 的双边沿捕获使能； 0：关闭通道 3 的双边沿捕获使能。	
0	CAP_ED_CH2	RW	通道 2 的双边沿捕获使能： 1：使能通道 2 的双边沿捕获使能； 0：关闭通道 2 的双边沿捕获使能。	0

第 16 章 基本定时器（BCTM）

基本定时器模块包含两个 16 位可自动重装的定时器（TIM6 和 TIM7），用于计数和在更新事件产生中断或 DMA 请求。TIM6 和 TIM7 支持 16 位可编程预分频器。可以为数模转换（DAC）提供时钟，触发 DAC 的同步电路。基本定时器之间是互相独立的，互不共享任何资源。

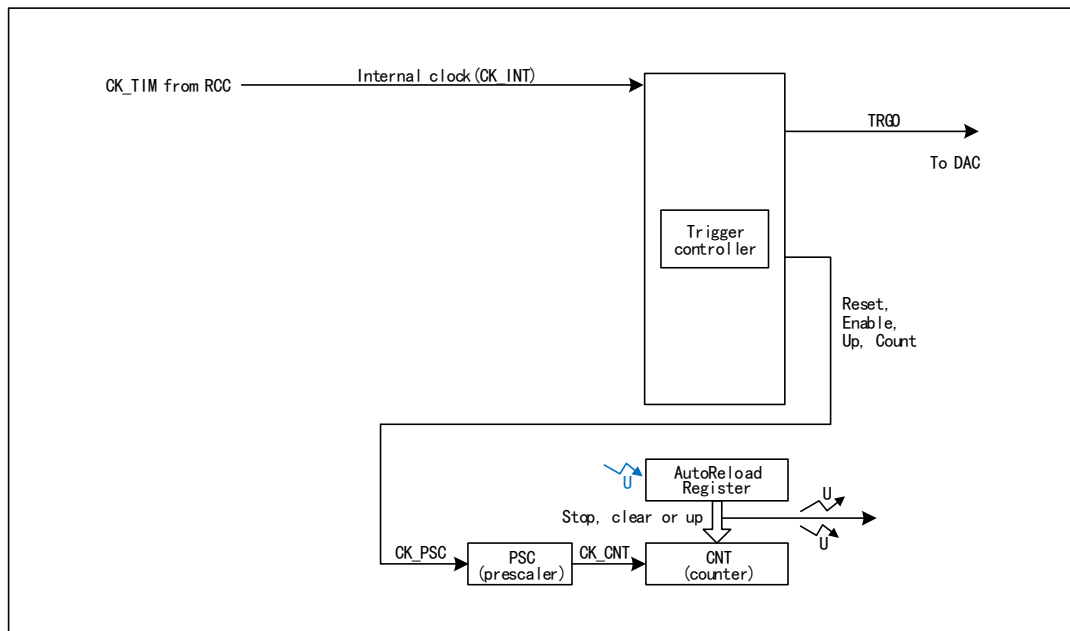
16.1 主要特征

基本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 16-1 所示，基本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 HB 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 重装载初始值。

16.2.2 基本定时器和通用定时器的区别

与通用定时器相比，基本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_CNT。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。并有一个自动重装值寄存器 (ATRLR) 可以设置计数器阈值, 当 CNT 计数值达到自动重装值寄存器 (ATRLR) 里设置的计数器阈值时, 会将 CNT 清零。

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	0xFFFF

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	0xFFFF

16.4.1 控制寄存器 1 (TIMx_CTLR1) (x=6/7)

偏移地址: 0x00

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

			0: 禁止自动重装值寄存器 (ATRLR)。	
[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

位	名称	访问	描述	复位值
[15:7]	Reserved	RO	保留。	0
[6:4]	MMS[2:0]	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_SMCFR 寄存器中 MSM 位的描述);	0

			010: 更新事件被选为触发输出 (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。	
[3:0]	Reserved	R0	保留。	0

16.4.3 DMA/中断使能寄存器 (TIMx_DMAINTENR) (x=6/7)

偏移地址: 0x0C

位	名称	访问	描述	复位值
[15:9]	Reserved	R0	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	R0	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

位	名称	访问	描述	复位值
[15:1]	Reserved	R0	保留。	0
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; 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: 无更新事件产生。	0

16.4.5 事件产生寄存器 (TIMx_SWEVGR) (x=6/7)

偏移地址: 0x14

位	名称	访问	描述	复位值
[15:1]	Reserved	R0	保留。	0
0	UG	W0	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。若在增计数模式下则核心计数器被清零。	0

16.4.6 通用定时器的计数器 (TIMx_CNT) (x=6/7)

偏移地址: 0x24

位	名称	访问	描述	复位值

[15:0]	CNT[15:0]	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

位	名称	访问	描述	复位值
[15:0]	PSC[15:0]	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

16.4.8 自动重装值寄存器 (TIMx_ATRLR) (x=6/7)

偏移地址: 0x2C

位	名称	访问	描述	复位值
[15:0]	ARR[15:0]	RW	计数器阈值, ATRLR 何时动作和更新请阅读 16.2.4 节; ATRLR 为空时, 计数器停止。	0xFFFF

第 17 章 低功耗定时器 (LPTIM)

低功耗定时器模块包含两个 16 位上行计数的定时器。LPTIM 具有多种可选的时钟源，使得 LPTIM 能在所有电源模式下运行。LPTIM 在没有内部时钟源的情况下也能运行，依此可以将 LPTIM 当作“脉冲计数器”使用。除此之外，LPTIM 还能将系统从低功耗模式唤醒，所以 LPTIM 很适合以极低的功耗实现“超时功能”。

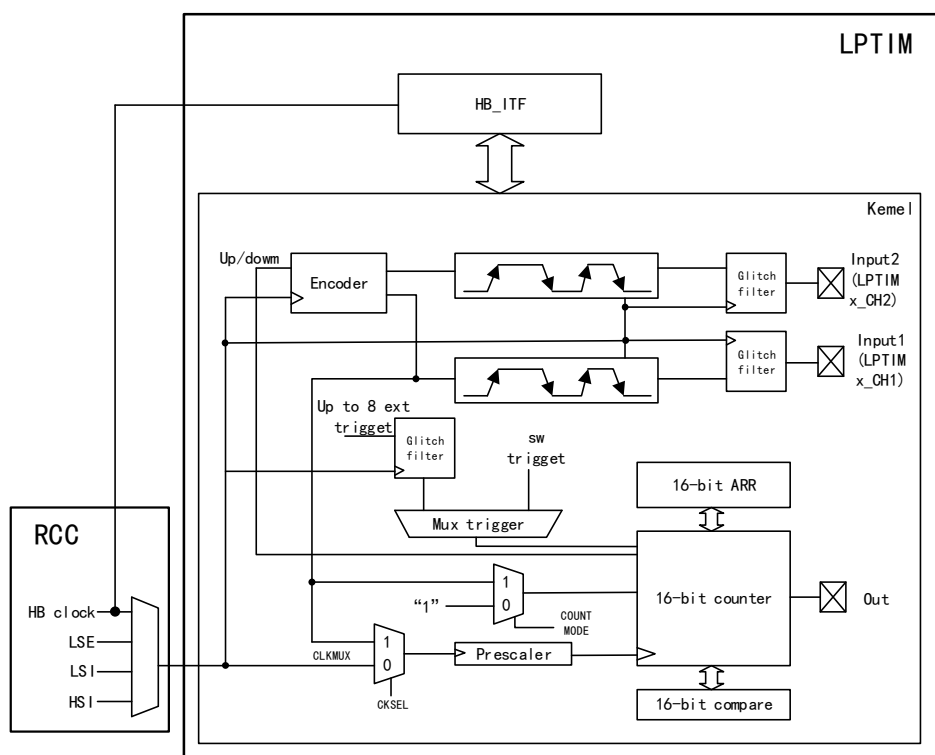
17.1 主要特征

- 16 位上行计数器
- 3 位预分频器，支持 8 种分频系数（1、2、4、8、16、32、64、128）
- 可选时钟源
内部时钟源：LSE、LSI、HSI 或 HB 时钟
外部时钟源：LPTIM 输入上的外部时钟
- 16 位 ARR 自动重装载寄存器
- 16 位比较寄存器
- 连续/单触发模式
- 可选择的软件/硬件输入触发器
- 可编程数字干扰滤波器
- 可配置输出 PWM 波
- 可配置 I/O 极性
- 编码器模式

17.2 LPTIM 功能描述

17.2.1 LPTIM 框图

图 17-1 低功耗定时器框图



17.3 LPTIM 触发器映射

LPTIM 外部触发器连接的信息如下所示：

表 17-1 LPTIMx 外部触发器连接

TRIGSEL[1:0]	外部触发
LPTIMx_TRG_00	LPTIMx_ETR
LPTIMx_TRG_01	RTC_ALARM

17.3.1 LPTIM 复位和时钟

LPTIM 模块复位由 RCC_HB1PRSTR 寄存器的 LPTIMRST 位控制，置 0 时无作用，置 1 时复位模块。

LPTIM 模块时钟使能由 RCC_HB1PCENR 寄存器的 LPTIMEN 位控制，置 0 时模块时钟关闭，置 1 时模块时钟开启。

LPTIM 的计数时钟由多个可选的时钟源提供，可分为内部时钟源和外部时钟源。

当使用内部时钟源计数时，内部时钟源可以通过 LPTIMx_CFGR 寄存器的 CLKMX_SEL 位中对 HB, LSI, LSE, HSI 四种时钟源进行选择。此外，LPTIM 可以使用外部输入 LPTIMx_CH1 上注入的外部时钟信号来计时。

表 17-2 LPTIM 内部时钟源

寄存器 LPTIMx_CFGR[26:25]	时钟源
00	TIM_CLK (来源于 HB_CLK)
01	HSI_CLK
10	LSE_CLK
11	LSI_CLK

当使用外部时钟源计数时，LPTIM 可能以以下两种配置之一运行：

第一种配置是由外部信号对 LPTIM 提供时钟，内部时钟信号同时由可配置的内部时钟源(HB、LSI、LSE、HSI)等提供。

第二种配置是 LPTIM 仅由外部时钟源通过其外部输入通道 1，该配置在进入低功耗模式后所有内部时钟源关闭时，实现脉冲计数器功能或超时功能。

对 CKSEL 和 COUNTMODE 位进行编程可以控制 LPTIM 时钟是使用外部时钟源还是内部时钟源。当配置为选择外部时钟源时，CKPOL 位用于选择外部时钟信号有效边沿。如果有效边沿设置为上升沿和下降沿（双边沿），则还要提供内部时钟信号。在这种情况下，内部时钟信号频率应该比外部时钟信号频率高四倍。

17.3.2 滤波器

LPTIM 输入无论是外部输入还是内部输入，都受到数字滤波器的保护，以防止任何毛刺和噪声干扰在 LPTIM 内部传播，进而避免产生意外计数或触发。

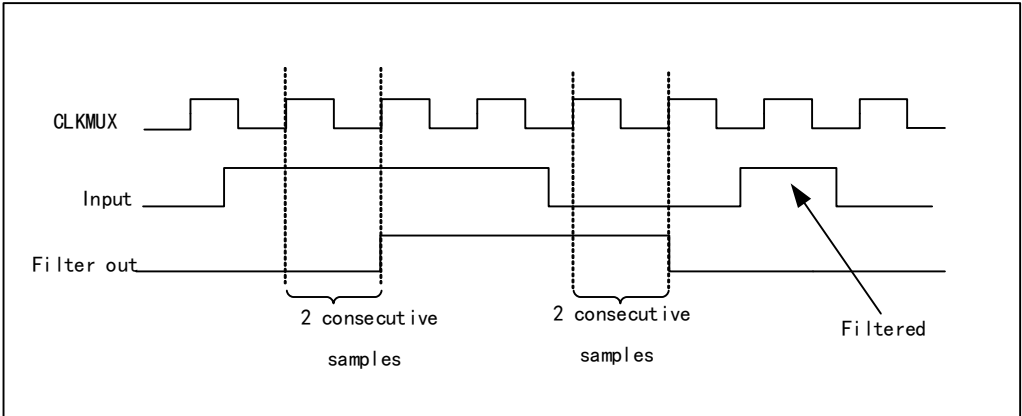
在激活数字滤波器之前，应先向 LPTIM 提供内部时钟源，保证过滤器正常运行。

数字滤波器分为两组：

一组为数字滤波器保护 LPTIM 外部输入，由 CKFLT 位控制数字滤波器的灵敏度；另一组数字滤波器保护 LPTIM 内部触发输入，由 TRGFLT 位控制数字滤波器的灵敏度。

滤波器的灵敏度会影响相同的连续采样的数量，在其中一个 LPTIM 输入上检测到此类连续的采样时，才能将某信号电平变化视为有效切换。下图给出了编程 2 个连续采样时，干扰滤波器行为的示例。

图 17-2 干扰滤波器时序框图



注：在不使用内部时钟信号的时候，数字滤波器的停用必须通过将 CKFLT 和 TRGFLT 位清零来实现，并使用外部模拟滤波器避免 LPTIM 外部输入带来的干扰。

17.3.3 预分频器

LPTIM 16 位计数器前面要有一个可配置的 2ⁿ 预分频器。由 PRESC[2:0] 3 位字段控制预分频器的分频比，表 17-3 列出了所有可能的分频比。

表 17-3 预分频器的分频比

PRESC[2:0]	分频比
000	1
001	1/2
010	1/4
011	1/8
100	1/16
101	1/32
110	1/64
111	1/128

17.3.4 触发器多路复用器

LPTIM 计数器有两种启动方式，一是通过软件启动，二是在检测到触发输入中一个以上的有效边沿后启动。LPTIM 触发方式由 TRIGEN[1:0] 控制，触发源由 TRIGSEL[1:0] 位控制。

表 17-4 触发方式

TRIGEN[1:0]	触发方式
00	无效
01	上升沿
10	下降沿
11	双边沿

表 17-5 触发源

TRIGSEL[1:0]	触发源
00	LPTIMx_ETR
01	RTC_ALARM

10	无效
11	无效

17.3.5 操作模式

LPTIM 具有两种操作模式：

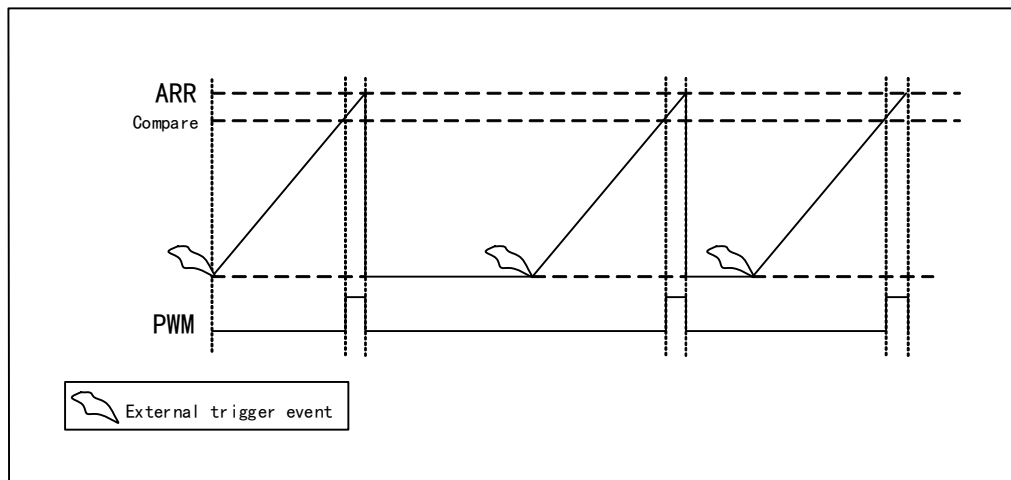
连续模式：计时器自由运行，从触发事件启动，直到计时器被禁用才停止。

单触发模式：计时器从触发事件启动，当达到 ARR 值时停止。

在单触发模式中，要启用单次计数，必须将 SNGSTRT 位置 1，一个新的触发事件将重新启动计时器，在计数器启动之后至计数器达到 ARR 之前发生的任何触发事件都将被丢失。

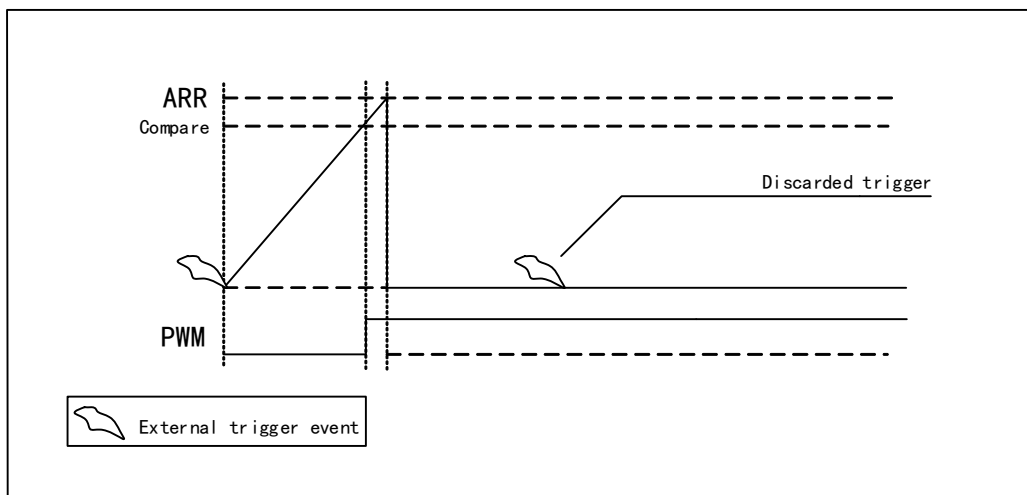
在选择用外部触发时，则在置位 SNGSTRT 位后以及计数器寄存器停止后到达的每个外部触发器事件将启动计数器进行新的一次计数循环。

图 17-3 单次计数模式配置时的 LPTIM 输出波形



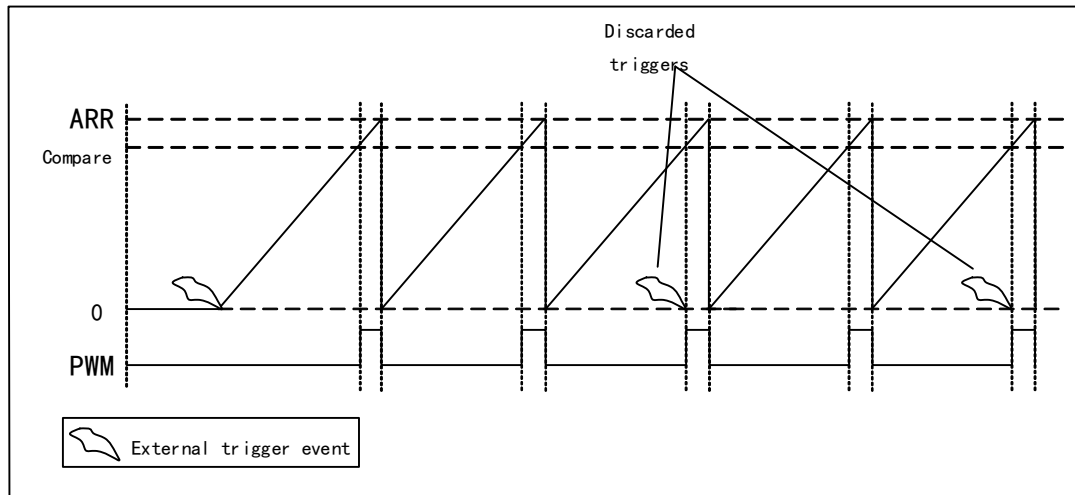
激活单触发设置模式，应该注意的是，在单触发模式下，当 LPTIMx_CFGR 寄存器中的 WAVE 位字段被设置时，一次设置模式被激活。在这种情况下，计数器仅在第一个触发后启动一次，任何后续的触发事件均被丢弃。

图 17-4 单次计数模式配置且激活一次设置模式时的 LPTIM 输出波形



连续模式，要启用连续计数，必须设置 CNTSTRT 位，如果选择了外部触发器，则在设置 CNTSTRT 之后到达的外部触发器事件将启动计数器进行连续计数。任何后续的外部触发事件都将被丢弃，在软件启动的情况下（TRIGEN=00）设置 CNTSTRT 将启动计数器连续计数。

图 17-5 连续计数模式配置时的 LPTIM 输出波形



SNGSTRT 和 CNTSTRT 位只能在定时器启用时设置（ENABLE 位设置为 1），可以将更改 LPTIM 计数器模式，如果先前选择了连续模式，则设置 SNGSTRT 将把 LPTIM 切换到单触发模式，计数器达到 ARR 值停止计数，如果先前选择了单触发模式，则设置 CNTSTRT 会将 LPTIM 切换到连续模式。计数器达到 ARR 值时立即重新启动。

17.3.6 超时功能

在一个选定的触发输入上检测到有效边沿可用于重置计数器，此功能通过 TIMOUT 位进行控制。第一个触发事件将启动计数器，任何连续的触发事件都将重置计数器并且计时器将重新启动，可以实现低功耗超时功能，如果没有触发事件发生，则 MCU 被比较匹配事件唤醒。

17.3.7 波形产生

两个 16 位寄存器 LPTIMx_ARR 和 LPTIMx_CMP，用于在 LPTIM 输出上生成几个不同的波形。

计时器可以生成以下波形：

- (1) PWM 模式：一旦 LPTIMx_CNT 中的计数器值超过 LPTIMx_CMP 中的比较值，就设置 LPTIM 输出。一旦 LPTIMx_ARR 和 LPTIMx_CNT 寄存器值相等，LPTIM 输出就会重置。
- (2) 单脉冲模式：输出波形与第一个脉冲的 PWM 模式相似，然后永久复位。
- (3) 一次设置模式：输出波形与单脉冲模式相似，只是输出保持在最后一个信号电平（取决于输出配置的极性）。

上述模式要求 LPTIMx_ARR 寄存器值严格大于 LPTIMx_CMP 寄存器值。

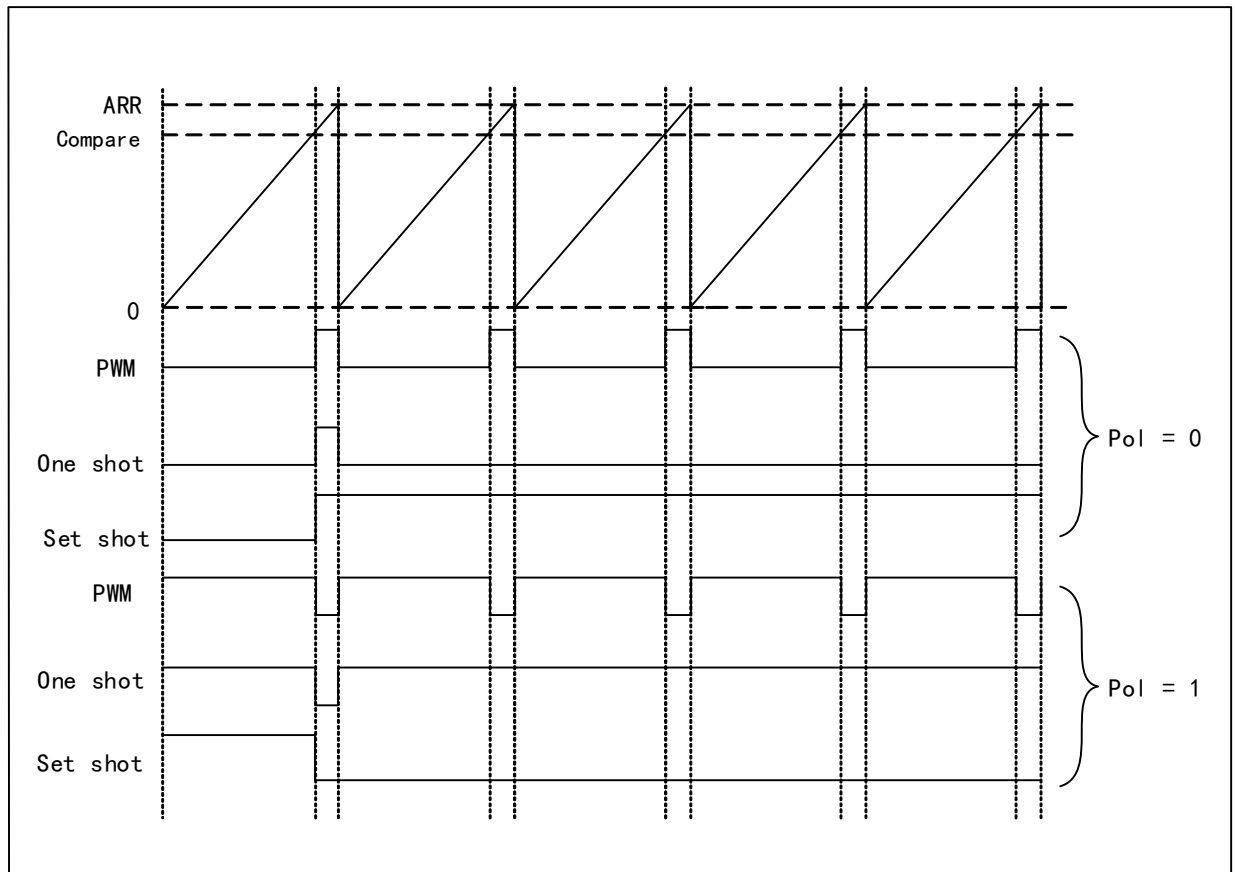
LPTIM 输出波形可以通过 WAVE 位配置如下：

- (1) 将 WAVE 位重置为 0 将强制 LPTIM 生成 PWM 波形或单脉冲波形，具体取决于设置的位：CNTSTRT 或 SNGSTRT。
- (2) 将 WAVE 位设置为 1 将强制 LPTIM 生成一次设置模式。

WAVPOL 位控制 LPTIM 输出极性，更改会立即生效，因此在重新配置极性后，甚至在启用计时器之前，输出默认值也会立即更改。

生成的信号的频率高达 LPTIM 时钟频率 2 分频。图 17-6 给出了可能在 LPTIM 输出上生成的三种波形。此外，此图还显示了通过 WAVPOL 位更改极性所产生的效果。

图 17-6 生成波形



17.3.8 寄存器更新

LPTIMx_ARR 寄存器和 LPTIMx_CMP 寄存器在 HB 总线写入操作之后立即更新，或者如果计时器已经启动，则在当前周期结束时更新。

PRELOAD 位控制 LPTIMx_ARR 和 LPTIMx_CMP 寄存器的更新方式：

当 PRELOAD 位设置为 0 时，LPTIMx_ARR 和 LPTIMx_CMP 寄存器将在任何写入访问后立即更新。

当 PRELOAD 位设置为 1 时，如果计时器已经启动，则在当前周期结束时更新 LPTIMx_ARR 和 LPTIMx_CMP 寄存器。

LPTIM HB 接口和 LPTIM 逻辑使用不同的时钟，因此在 HB 写入和这些值可用于计数器比较器之间存在一些延迟，在这个延迟期内，必须避免对这些寄存器进行任何额外的写入。

LPTIMx_ISR 寄存器中的 ARROK 标志和 CMPOK 标志分别指示何时完成对 LPTIMx_ARR 寄存器和 LPTIMx_CMP 寄存器的写入操作。

在对 LPTIMx_ARR 寄存器或 LPTIMx_CMP 寄存器进行写入之后，只有在前一次写入操作完成时，才能对同一寄存器执行新的写入操作。

在设置 ARROK 标志或 CMPOK 标志之前的任何连续写入都将导致不可预测的结果。

17.3.9 计数器模式

LPTIM 计数器可用于对 LPTIM_CH1 上的外部事件进行计数，也可用于对内部时钟周期进行计数。CKSEL 和 COUNTMODE 位控制哪个源将用于更新计数器。

如果 LPTIM 被配置为 LPTIM_CH1 上的外部事件进行计数，则可以根据写入 CKPOL[1:0] 位的值在上升沿，下降沿或双边沿之后更新计数。

根据 CKSEL 和 COUNTMODE 值，可以选择以下计数模式：

- (1) CKSEL=0: LPTIM 时钟由内部时钟源计数

COUNTMODE=0, LPTIM 被配置为由内部时钟源计数时, 并且 LPTIM 计数器被配置为跟随每个内部时钟脉冲之后更新。

COUNTMODE=1, 使用提供给 LPTIM 的内部时钟对 LPTIM_CH1 上的数据进行采样。因此为了不遗漏任何事件, LPTIM_CH1 上的信号的变化频率不应超过 LPTIM 内部时钟的频率。

(2) CKSEL=1: LPTIM 由外部时钟源计数时, COUNTMODE 的值无影响。

在此配置中, LPTIM 不需要内部时钟源(除非启用数字滤波器), 在 LPTIM_CH1 上注入的信号用作 LPTIM 的系统时钟。此配置适用于未启用内置振荡器的工作模式。

对于此配置, LPTIM 计数器可以在 LPTIM_CH1 时钟信号的上升沿或下降沿更新, 但不能在双边沿更新。由于在 LPTIM_CH1 注入的信号也用于 LPTIM 计数时, 因此在计数器递增之前会有一些初始延迟(在启用 LPTIM 之后)。准确来说, 在使能 LPTIM 之后, LPTIM 外部 LPTIM_CH1 的前五个有效边沿将丢失。

17.3.10 定时器启用

位于 LPTIMx_CR 寄存器中的 ENABLE 位用于启用/禁用 LPTIM 逻辑, 在设置 ENABLE 位后, 在实际启用 LPTIM 之前需要两个计数器时钟的延迟。

17.3.11 编码器模式

该模式允许处理来自用于检测旋转元件的位置的正交编码器的信号。编码器接口模式只是作为一个有方向选择的外部时钟。这意味着计数器仅在 0 和 LPTIMx_ARR 寄存器中的自动重载值之间连续计数。因此, 在启动之前必须配置 LPTIMx_ARR 根据两个外部输入信号 LPTIMx_CH1 和 LPTIMx_CH2 生成时钟信号以对 LPTIM 计数器进行计时。这两个信号之间的相位决定了计数方向。

只有当 LPTIM 由内部时钟源计数时, 编码器模式才可用。LPTIMx_CH1 和 LPTIMx_CH2 上的信号频率不得超过 LPTIM 内部时钟的频率 4 分频。满足以上条件才能确保 LPTIM 正常运行。

方向变化由 LPTIMx_ISR 寄存器中的两个向上和向下标志发出信号。此外, 如果通过 DOWNIE 位启用, 则可以为两个方向改变事件生成中断。

要激活编码器模式, 必须将 ENC 位设置为 1, 必须首先将 LPTIM 配置为连续模式。

当编码器模式处于激活状态, LPTIM 计数器会根据增量编码器的速度和方向自动修改。因此, 它的内容总是代表编码器的位置。由向上和向下标志指示计数方向, 对应于编码器转子的旋转方向。根据 CKPOL[1:0] 位配置的边缘灵敏度, 得到了以下可能的组合方案, LPTIMx_CH1 和 LPTIMx_CH2 不同时切换。

表 17-6 配置的边缘灵敏度组合方案

活动边沿 (CKPOL[1:0])	相反信号的电平 (LPTIMx_CH1 对应的输入是 LPTIMx_CH2, LPTIMx_CH2 对应的输入是 LPTIMx_CH1)	LPTIMx_CH1		LPTIMx_CH2	
		上升	下降	上升	下降
上升沿 (00)	高	递减	不计数	递增	不计数
	低	递增	不计数	递减	不计数
下降沿 (01)	高	不计数	递增	不计数	递减
	低	不计数	递减	不计数	递增
双边沿 (10)	高	递减	递增	递增	递减
	低	递增	递减	递减	递增
11	无效				

17.3.12 调试模式

当 MCU 进入调式模式（内核停止），LPTIM 继续正常工作。

17.3.13 LPTIM 低功耗模式

表 17-7 低功耗模式对 LPTIM 的影响

模式	描述
睡眠模式	无影响，LPTIM 中断会导致设备退出睡眠模式
停止模式	LPTIM 外围设备在由 LSE 或 LSI 计时时处于活动状态，LPTIM 中断导致设备退出停止模式

17.4 LPTIM 中断

如果通过 LPTIMx_IER 寄存器使能以下事件，则会产生中断/唤醒：

- 比较匹配
- 自动重载匹配
- 外部触发事件
- 自动重载寄存器写入完成
- 比较寄存器写入完成
- 方向改变（编码器模式）

表 17-8 中断事件

中断事件	描述
比较匹配	当计数器寄存器 LPTIMx_CNT 的值与比较寄存器 LPTIMx_CMP 的值相等时会产生中断标志
自动重载匹配	当计数器寄存器 LPTIMx_CNT 的值与自动重载寄存器 LPTIMx_ARR 的值相等时会产生中断标志
外部触发事件	当检测到外部触发事件时会产生中断标志
自动重载寄存器写入完成	对 LPTIMx_ARR 寄存器的写入操作完成时会产生中断标志
比较寄存器写入完成	对 LPTIMx_CMP 寄存器的写入操作完成时会产生中断标志
计数方向改变	在编码器模式下使用，两个中断标志嵌入到信号的方向改变： 向上标志：表示向上计数方向改变； 向下标志：表示向下计数方向改变。

17.5 寄存器描述

表 17-9 LPTIM1 相关寄存器列表

名称	访问地址	描述	复位值
R32_LPTIM1_ISR	0x40002400	LPTIM1 中断状态寄存器	0x00000000
R32_LPTIM1_ICR	0x40002404	LPTIM1 中断清除寄存器	0x00000000
R32_LPTIM1_IER	0x40002408	LPTIM1 中断使能寄存器	0x00000000
R32_LPTIM1_CFGR	0x4000240C	LPTIM1 配置寄存器	0x00000000
R32_LPTIM1_CR	0x40002410	LPTIM1 控制寄存器	0x00000000
R16_LPTIM1_CMR	0x40002414	LPTIM1 比较寄存器	0x00000000
R16_LPTIM1_ARR	0x40002418	LPTIM1 自动重载寄存器	0x00000001
R16_LPTIM1_CNT	0x4000241C	LPTIM1 计数寄存器	0x00000000

表 17-10 LPTIM2 相关寄存器列表

名称	访问地址	描述	复位值
R32_LPTIM2_ISR	0x40003400	LPTIM2 中断状态寄存器	0x00000000
R32_LPTIM2_ICR	0x40003404	LPTIM2 中断清除寄存器	0x00000000
R32_LPTIM2_IER	0x40003408	LPTIM2 中断使能寄存器	0x00000000
R32_LPTIM2_CFGR	0x4000340C	LPTIM2 配置寄存器	0x00000000
R32_LPTIM2_CR	0x40003410	LPTIM2 控制寄存器	0x00000000
R16_LPTIM2_CMR	0x40003414	LPTIM2 比较寄存器	0x00000000
R16_LPTIM2_ARR	0x40003418	LPTIM2 自动重载寄存器	0x00000001
R16_LPTIM2_CNT	0x4000341C	LPTIM2 计数寄存器	0x00000000

17.5.1 中断状态寄存器 (LPTIMx_ISR) (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								DIR_S YNC	DOW N	UP	ARR OK	CMP OK	EXTTR IG	ARR M	CMP M

位	名称	访问	描述	复位值
[31:8]	Reserved	R0	保留。	0
7	DIR_SYNC	R0	编码器模式下, 计数器计数方向: 1: 向下计数; 0: 向上计数。 该位只在编码器模式下有效, 即 ENC 位置 1, 使用前还需将 DIR_EXTEN 置 1 使能计数方向功能。	0
6	DOWN	R0	向下计数: 1: 计数方向从上到下; 0: 无效。 在编码器模式下, 该位由硬件设置, 以通知应用程序计数器方向从上到下变化, 为 1 之后可以通过向 LPTIMx_ICR[6]位写 1 来清除。	0
5	UP	R0	向上计数: 1: 计数方向从下到上; 0: 无效。 在编码器模式下, 该位由硬件设置, 以通知应用程序计数器方向从下到上变化, 为 1 之后可以通过向 LPTIMx_ICR[5]位写 1 来清除。	0
4	ARROK	R0	自动重加载寄存器数据更新成功: 1: 数据更新成功; 0: 无效。 该位由硬件设置, 以通知应用程序 HB 总线对 LPTIMx_ARR 的写入操作已成功完成, 为 1 之后可以通	0

			过向 LPTIMx_ICR[4]位写 1 来清除。	
3	CMPOK	RO	比较寄存器数据更新成功： 1：数据更新成功； 0：无效。 该位由硬件设置，以通知应用程序 HB 总线对 LPTIMx_CMR 的写入操作已成功完成，为 1 之后可以通过向 LPTIMx_ICR[3]位写 1 来清除。	0
2	EXTTRIG	RO	外部触发边缘事件： 1：有效边缘输入已发生； 0：无效。 该位由硬件设置，以通知应用程序所选外部触发器上的有效边缘输入已发生，如果由于计时器已经启动而忽略触发器，则此标志为未设置。为 1 之后可以通过向 LPTIMx_ICR[2]位写 1 来清除。	0
1	ARRM	RO	自动重加载寄存器数据与 LPTIMx_CNT 寄存器的数据匹配： 1：匹配成功； 0：无效。 该位由硬件设置，以通知应用程序 LPTIMx_CNT 寄存器的值达到 LPTIMx_ARR 寄存器的值，为 1 之后可以通过向 LPTIMx_ICR[1]位写 1 来清除。	0
0	CMPM	RO	比较寄存器的数据与 LPTIMx_CNT 寄存器的数据匹配： 1：匹配成功； 0：无效。 该位由硬件设置，以通知应用程序 LPTIMx_CNT 寄存器的值达到 LPTIMx_CMR 寄存器的值，为 1 之后可以通过向 LPTIMx_ICR[0]位写 1 来清除。	0

17.5.2 中断清除寄存器 (LPTIMx_ICR) (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									DOW NCF	UPN CF	ARRO KNCF	CMP OKN CF	EXTT RIGN CF	ARR MNC F	CMPM NCF

位	名称	访问	描述	复位值
[31:7]	Reserved	RO	保留。	0
6	DOWNCF	RW1	清除向下标志位。 注：写 0 无效，写 1 清 0 状态寄存器对应位。	x
5	UPCF	RW1	清除向上标志位。 注：写 0 无效，写 1 清 0 状态寄存器对应位。	x
4	ARROKCF	RW1	清除自动重加载数据更新标志位。	x

			注：写 0 无效，写 1 清 0 状态寄存器对应位。	
3	CMPOKCF	RW1	清除比较器数据更新标志位。 注：写 0 无效，写 1 清 0 状态寄存器对应位。	x
2	EXTTRIGCF	RW1	清除外部触发器边缘事件标志位。 注：写 0 无效，写 1 清 0 状态寄存器对应位。	x
1	ARRMCF	RW1	清除自动重加载寄存器匹配标志位。 注：写 0 无效，写 1 清 0 状态寄存器对应位。	x
0	CMPCMCF	RW1	清除比较寄存器匹配标志位。 注：写 0 无效，写 1 清 0 状态寄存器对应位。	x

17.5.3 中断使能寄存器 (LPTIMx_IER) (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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE	

位	名称	访问	描述	复位值
[31:7]	Reserved	RO	保留。	0
6	DOWNIE	RW	向下中断使能： 1：开启； 0：关闭。	0
5	UPIE	RW	向上中断使能： 1：开启； 0：关闭。	0
4	ARROKIE	RW	自动重加载寄存器数据更新成功中断使能： 1：开启； 0：关闭。	0
3	CMPOKIE	RW	比较寄存器数据更新成功中断使能： 1：开启； 0：关闭。	0
2	EXTTRIGIE	RW	外部触发边缘事件输入中断使能： 1：开启； 0：关闭。	0
1	ARRMIE	RW	自动重加载寄存器数据匹配成功中断使能： 1：开启； 0：关闭。	0
0	CMPMIE	RW	比较寄存器数据匹配成功中断使能： 1：开启； 0：关闭。	0

17.5.4 配置寄存器 (LPTIMx_CFGR) (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				FORCE_PWM	CLKMX_SEL		ENC	COUNT_MODE	PRELOAD	WAVPOL	WAVE	TIMOUT	TRIGEN		Reserved
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRIGSEL			Reserved	PRESC		Reserved	TRGFLT		Reserved	CKFLT		CKPOL		CKSEL

位	名称	访问	描述	复位值
[31:28]	Reserved	RO	保留。	0
27	FORCE_PWM	RW	强制 PWM 输出： 1：强制 PWM 输出高电平； 0：无效。	0
[26:25]	CLKMX_SEL[1:0]	RW	LPTIM 计数器内部时钟源选择： 00：TIM_CLK（来源于 HB_CLK）； 01：HSI_CLK； 10：LSE_CLK； 11：LSI_CLK。	0
24	ENC	RW	编码器模式使能： 1：开启； 0：关闭。	0
23	COUNTMODE	RW	计数器模式，选择 LPTIM 使用哪个时钟源为计数器计时： 1：当 LPTIM 外部输入每个有效脉冲后，计数器递增； 0：计数器在每个内部时钟脉冲后递增。	0
22	PRELOAD	RW	寄存器更新模式，控制 LPTIMx_ARR 和 LPTIMx_CMP 寄存器更新模式： 1：寄存器在当前 LPTIM 周期结束时更新； 0：每次 HB 总线写入访问后更新寄存器。	0
21	WAVPOL	RW	PWM 波形极性： 1：输出反映 LPTIMx_ARR 和 LPTIMx_CMP 寄存器之间比较结果的取反； 0：输出反映 LPTIMx_ARR 和 LPTIMx_CMP 寄存器之间比较结果。 注：如果计数器值大于比较器值比较结果为 1，否则为 0。	0
20	WAVE	RW	PWM 波形： 1：设置一次模式（一个脉冲波形）； 0：关闭一次模式。	0
19	TIMOUT	RW	超时启用，控制超时功能： 1：计时器已启动时到达的触发事件将重置并重新启动计数器； 0：计时器已启动时到达的触发事件被忽略。	0
[18:17]	TRIGEN[1:0]	RW	触发启用和极性，控制 LPTIM 计数器是否由外部触发器启动，如果选择了外部触发器选项，则触发器活动边缘有三种配置。 00：软件触发（计数器启动由软件启动）；	0

			01: 上升沿触发; 10: 下降沿触发; 11: 双边沿触发。	
[16:15]	Reserved	R0	保留。	0
[14:13]	TRIGSEL[1:0]	RW	触发源选择: 00: LPTIM_ETR; 01: RTC_ALARM; 10: 无效; 11: 无效。	0
12	Reserved	R0	保留	0
[11:9]	PRESC[2:0]	RW	时钟预分频器, 配置预分频系数: 000: 1 分频; 001: 2 分频; 010: 4 分频; 011: 8 分频; 100: 16 分频; 101: 32 分频; 110: 64 分频; 111: 128 分频。	0
8	Reserved	R0	保留。	0
[7:6]	TRGFLT[1:0]	RW	用于触发器的可配置数字滤波器: TRGFLT 值设置在内部触发器上发生变化时应检测到的连续相等样本的数量, 然后将其视为有效的电平转换。必须有内部时钟源才能使用此功能。 00: 任何触发器的更改都被视为有效触发器; 01: 触发激活电平变化必须稳定至少 2 个时钟周期, 然后才视为有效触发器; 10: 触发激活电平变化必须稳定至少 4 个时钟周期, 然后才视为有效触发器; 11: 触发激活电平变化必须稳定至少 8 个时钟周期, 然后才视为有效触发器。	0
5	Reserved	R0	保留。	0
[4:3]	CKFLT[1:0]	RW	外部时钟的可配置数字滤波器: CKFLT 值用于设置连续相等样本的数量。当外部时钟信号发生电平变化时应检测到此类连续样本, 才会将此电平变化视为有效电平转换。必须有内部时钟源才能使用此功能。 00: 任何外部时钟信号电平变化都被视为有效转换; 01: 外部时钟信号电平变化必须稳定至少 2 个时钟周期, 然后才视为有效; 10: 外部时钟信号电平变化必须稳定至少 4 个时钟周期, 然后才视为有效; 11: 外部时钟信号电平变化必须稳定至少 8 个时钟周期, 然后才视为有效。	0
[2:1]	CKPOL[1:0]	RW	如果 LPTIM 选择外部时钟源计数, CKPOL 位用于配置有效边沿:	0

			00: 上升沿用于计数; 01: 下降沿用于计数; 10: 双边沿用于计数, 当外部时钟信号的双边沿都有效时, LPTIM 也必须由频率至少等于外部时钟频率四倍的内部时钟源进行计时; 11: 无效。 如果 LPTIM 配置为编码器模式时 (ENC 位置 1), CKPOL 位用于选择编码器模式: 00: 编码器模式 1 激活; 01: 编码器模式 2 激活; 10: 编码器模式 3 激活; 11: 无效。	
0	CKSEL	RW	时钟选择器, 用于选择 LPTIM 使用的时钟: 1: LPTIM 使用外部时钟; 0: LPTIM 使用内部时钟。	0

17.5.5 控制寄存器 (LPTIMx_CR) (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											DIR_EXTEN	OUTEN	CNTSTRT	SNGSTRT	ENABLE

位	名称	访问	描述	复位值
[31:5]	Reserved	R0	保留。	0
4	DIR_EXTEN	RW	外部触发计数方向使能: 1: 开启; 0: 关闭。 注: 编码器模式开启后使用。	0
3	OUTEN	RW	PWM 波输出使能: 1: 开启; 0: 关闭。 注: 在非编码器模式下, 该位使能输出 PWM 波需要将 ENABLE 同时使能才有效。	0
2	CNTSTRT	RW	在连续模式下启动: 该位由软件设置, 并由硬件清除, 在软件启动的情况下 (TRIGEN==00), 设置此位以连续模式启动 LPTIM, 如果软件启动被禁用 (TRIGEN!=00) 则设置此位会在检测到外部触发器后立即以连续模式启动 LPTIM。如果在 LPTIM 处于单脉冲计数模式时设置此位, 则 LPTIM 将在计数器达到 ARR 值时不会停止计数。 注: ENABLE=1 时才可以写。	0
1	SNGSTRT	RW	LPTIM 在单触发模式下启动:	0

			该位由软件设置,并由硬件清除,在软件启动的情况下 (TRIGEN==00), 设置此位以单脉冲模式启动 LPTIM, 如果软件启动被禁用 (TRIGEN!=00) 则设置此位会在检测到外部触发器后立即以单脉冲模式启动 LPTIM。如果在 LPTIM 处于连续计数模式时设置此位, 则 LPTIM 将在计数器达到 ARR 值时停止。 注: ENABLE=1 时才可以写。	
0	ENABLE	RW	定时器使能, 当定时器使能位被清 0 时, 将会复位内部逻辑, 并且无法操作 LPT_CR[2:1] 位。 1: 低功耗定时器开启; 0: 低功耗定时器禁止并且复位内部逻辑。	0

17.6.6 比较寄存器 (LPTIMx_CMP) (x=1/2)

偏移地址: 0x14

位	名称	访问	描述	复位值
[15:0]	CMP	RW	比较值, 当计数器的值等于比较寄存器中的值, 定时器比较标志会被置 1, 如果在置 1 之前打开相应的使能信号会产生中断源和低功耗唤醒信号。	0

17.5.7 自动重载寄存器 (LPTIMx_ARR) (x=1/2)

偏移地址: 0x18

位	名称	访问	描述	复位值
[15:0]	ARR	RW	计数寄存器重新装载的计数值, 向上计数时如果计数值等于重新装载的计数值, 则计数器从 0 开始计数; 向下计数时如果计数器的值等于 0 则从重新装载的计数值开始计数。	1

17.5.8 计数寄存器 (LPTIMx_CNT) (x=1/2)

偏移地址: 0x1C

位	名称	访问	描述	复位值
[15:0]	COUNT	RO	当前定时器计数器寄存器的计数值: 当 LPTIM 使用异步时钟运行时, 读取 LPTIMx_CNT 寄存器可能返回不可靠的值, 因此, 在这种情况下, 有必要执行两次连续的读取访问, 并验证两个返回值是否相同。应该注意的是, 对于可靠的 LPTIMx_CNT 寄存器读取访问, 两次连续读取必须执行访问并进行比较。	0

第 18 章 数字/模拟转换（DAC）

数字/模拟转换模块（DAC），包含 2 个 12 位电压输出数字/模拟转换器（DAC），转换 2 路数字信号为 2 路模拟电压信号并输出，支持双 DAC 通道独立或同步转换，支持 12 位数据左对齐或右对齐，支持 12 位或 8 位数据，支持外部事件触发转换。可实现三角波、噪声生成。支持 DMA 功能。。

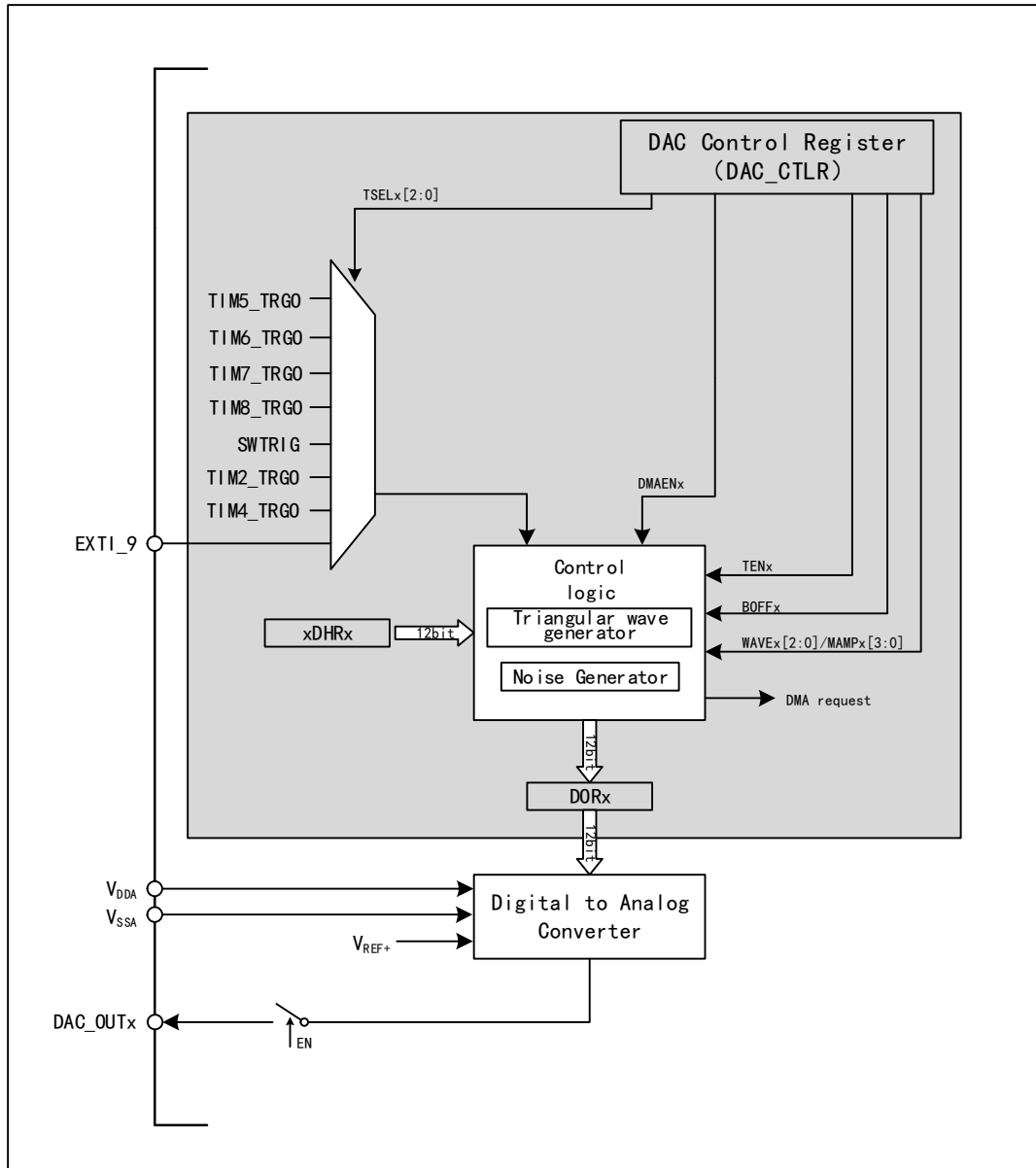
18.1 主要特性

- 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

18.2 功能描述

18.2.1 DAC 模块结构

图 18-1 DAC 模块框图



18.2.2 DAC 通道配置

18.2.2.1 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。

18.2.2.2 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。

18.2.2.3 数据格式：

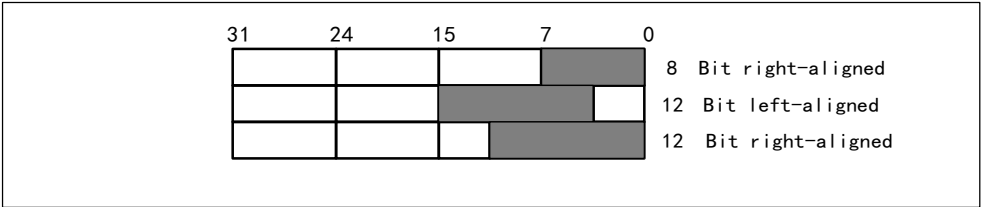
单 DAC 通道模式下，包括 8 位数据右对齐、12 位数据左对齐、12 位数据右对齐。

8 位数据右对齐时，写入数据到 DAC_R8BDHRx[7:0]，模块将加载（1 个 HB 时钟周期后）其左移数据到数据输出寄存器 DAC_DORx[11:4]。

12 位数据右对齐时，写入数据到 DAC_R12BDHRx[11:0]，模块将加载（1 个 HB 时钟周期后）右对齐数据到数据输出寄存器 DAC_DORx[11:0]。

12 数据左对齐时，写入数据到 DAC_L12BDHRx[15:4]，模块经过相应的移位后，将加载（1 个 HB 时钟周期后）左对齐数据到数据输出寄存器 DAC_DORx[11:0]。

图 18-2 单通道数据格式

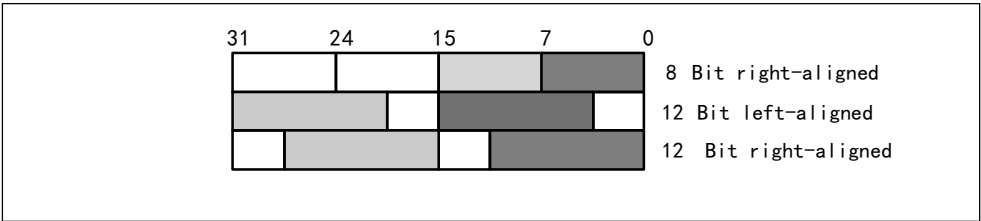


双 DAC 通道模式下，同样包括 8 位数据右对齐、12 数据左对齐、12 位数据右对齐三种方式。
8 位数据右对齐时，写入数据到 DAC_RD8BDHR[7:0]，模块将加载（1 个 HB 时钟周期后）位[7:0]移位后到 DAC_DOR1[11:4]，位[15:8]移位后到 DAC_DOR2[11:4]。

12 数据左对齐时，写入数据到 DAC_LD12BDHR[31:0]，模块将加载（1 个 HB 时钟周期后）位[15:4]数据移位后到 DAC_DOR1[11:0]，位[31:20]数据移位后到 DAC_DOR2[11:0]。

12 位数据右对齐，写入数据到 DAC_RD12BDHR[31:0]，模块将加载（1 个 HB 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 18-3 双通道数据格式



18.2.2.4 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。

18.2.2.5 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 18-1 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 HB 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 HB 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。

18.2.3 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 HB 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 HB 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 HB 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_{DD33A} 。任一 DAC 通道引脚上的输出电压满足下面的关系：

$$\text{DAC 输出电压} = V_{\text{DD33A}} * (\text{DAC\_DORx} / 4096)。$$

18.2.4 DAC 三角波生成器

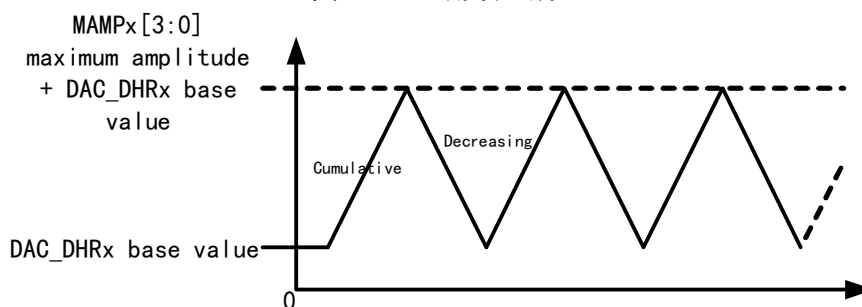
模块内置了一个三角波生成器，可以在基准信号上加上一个小幅度的三角波。设置 WAVEx[1:0] 位为 10b，选择 DAC 的三角波生成功能。设置 DAC_CTLR 寄存器的 MAMPx[3:0] 位来选择三角波的幅度。

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 HB 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 18-4 三角波生成



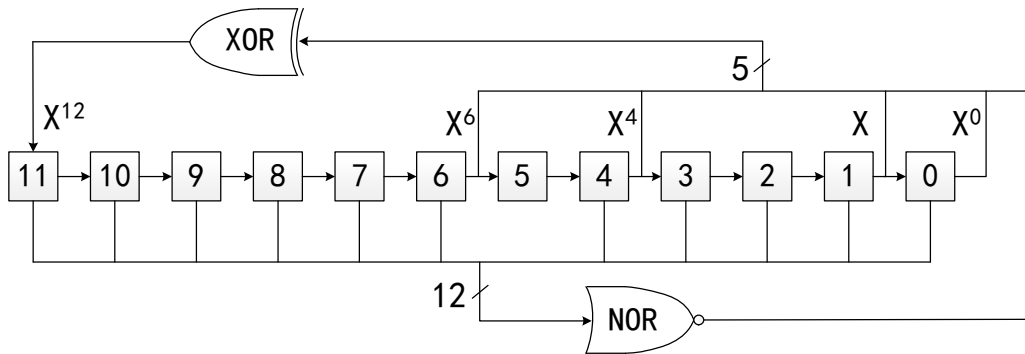
18.2.5 DAC 噪声生成器

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 01b，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 HB 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 00b，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 18-5 LFSR 寄存器算法



18.3 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。

18.3.1 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 HB 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 HB 时钟后送给 DAC_DOR2，用于转换，并更新 LFSR2。

18.3.2 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 HB 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 HB 时钟后送给 DAC_DOR2，用于转换，并更新 LFSR2。

18.3.3 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 HB 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 HB 时钟后送给 DAC_DOR2，用于转换，并更新通道 2 三角波计数器。

18.3.4 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 HB 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 HB 时钟后送给 DAC_DOR2，

用于转换，并更新通道 2 三角波计数器。

18.3.5 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 HB 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 HB 时钟后送给 DAC_DOR2，用于转换。

18.3.6 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 HB 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。

18.3.7 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 HB 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 HB 时钟后送给 DAC_DOR2，用于转换，并更新 LFSR2。

18.3.8 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 HB 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 HB 时钟后送给 DAC_DOR2，用于转换，并更新 LFSR2。

18.3.9 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 HB 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 HB 时钟后送给 DAC_DOR2，用于转换，并更新通道 2 三角波计数器值。

18.3.10 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 HB 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 HB 时钟后送给 DAC_DOR2，用于转换，并更新通道 2 三角波计数器值。

18.3.11 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 HB 时钟后分别送给 DAC_DOR1 和 DAC_DOR2 用于 DAC 转换。

18.4 寄存器描述

表 18-2 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

18.4.1 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[3:0]	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。	0000b
[23:22]	WAVE2[1:0]	RW	DAC 通道 2 的噪声/三角波生成使能：	00b

			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。	
[21:19]	TSEL2[2:0]	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。	000b
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 HB 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 HB 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 HB 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[3:0]	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。	0000b
[7:6]	WAVE1[1:0]	RW	DAC 通道 1 的噪声/三角波生成使能。	00b

			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。	
[5:3]	TSEL1[2:0]	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。	000b
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 HB 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 HB 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 HB 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

18.4.2 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														SW TRIG2	SW TRIG1

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 HB 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 HB 时钟周期后) 送入 DAC_DOR1 寄存器, 该位将硬件清 0。	

18.4.3 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[11:0]	RW	DAC 通道 1 的 12 位右对齐数据。	0

18.4.4 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
[31:16]	Reserved	RO	保留。	0
[15:4]	DACC1DHR[11:0]	RW	DAC 通道 1 的 12 位左对齐数据。	0
[3:0]	Reserved	RO	保留。	0

18.4.5 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[7:0]	RW	DAC 通道 1 的 8 位右对齐数据。	0

18.4.6 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[11:0]	RW	DAC 通道 2 的 12 位右对齐数据。	0

18.4.7 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[11:0]	RW	DAC 通道 2 的 12 位左对齐数据。	0
[3:0]	Reserved	RO	保留。	0

18.4.8 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[7:0]	RW	DAC 通道 2 的 8 位右对齐数据。	0

18.4.9 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[11:0]	RW	DAC 通道 2 的 12 位右对齐数据。	0
[15:12]	Reserved	RO	保留。	0
[11:0]	DACC1DHR[11:0]	RW	DAC 通道 1 的 12 位右对齐数据。	0

18.4.10 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[11:0]	RW	DAC 通道 2 的 12 位左对齐数据。	0
[19:16]	Reserved	RO	保留。	0
[15:4]	DACC1DHR[11:0]	RW	DAC 通道 1 的 12 位左对齐数据。	0
[3:0]	Reserved	RO	保留。	0

18.4.11 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[7:0]	RW	DAC 通道 2 的 8 位右对齐数据。	0
[7:0]	DACC1DHR[7:0]	RW	DAC 通道 1 的 8 位右对齐数据。	0

18.4.12 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[11:0]	R0	DAC 通道 1 输出数据。	0

18. 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	R0	保留。	0
[11:0]	DACC2DOR[11:0]	R0	DAC 通道 2 输出数据。	0

第 19 章 通用同步异步收发器 (USART)

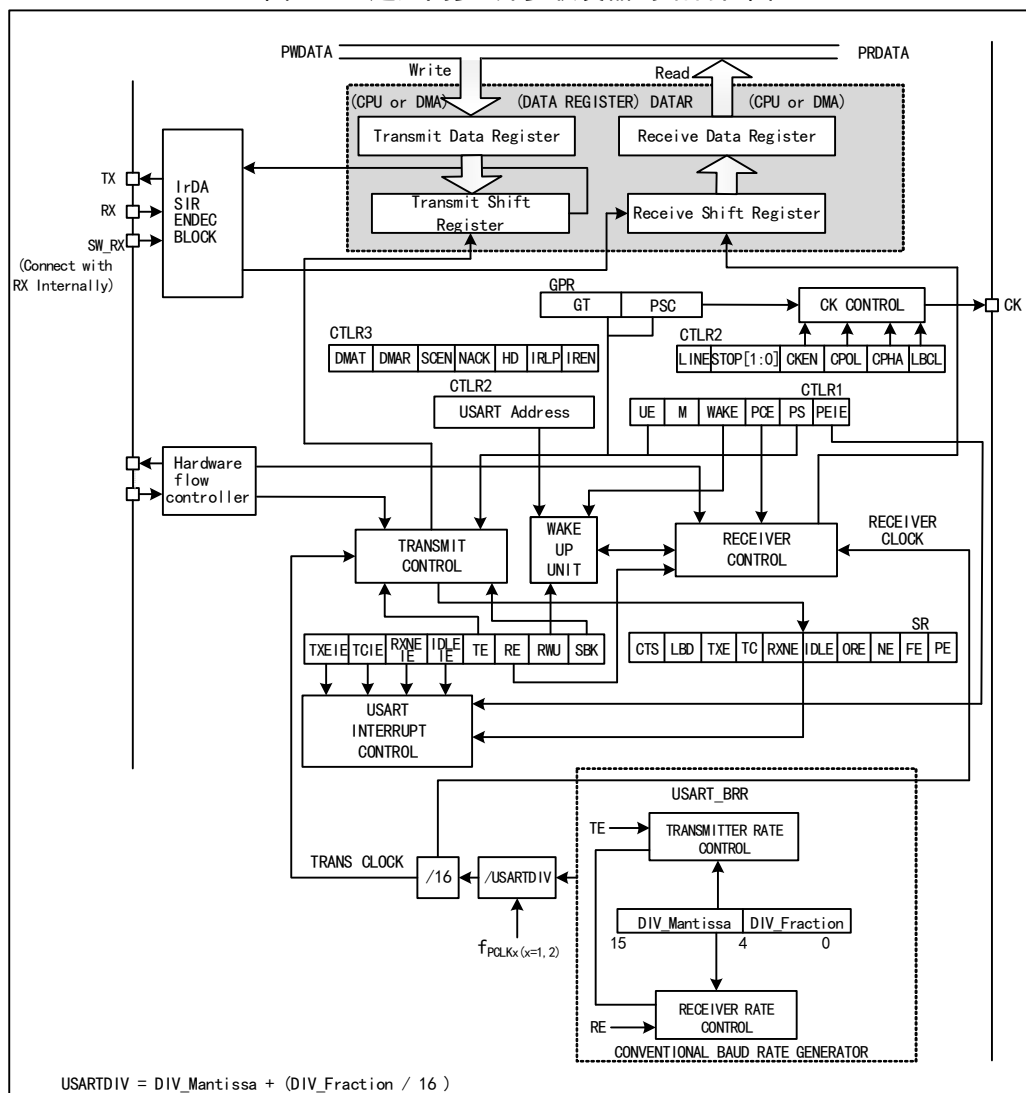
该模块包含 8 个通用同步异步收发器 (USART1/2/3/4/5/6/7/8)。

19.1 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
- 支持低功耗模式下唤醒功能

19.2 概述

图 19-1 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。

19.3 波特率发生器

收发器的波特率 = $HCLK / (16 * USARTDIV)$ 。USARTDIV 的值是根据 USARTx_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 HB 的时钟。比如设定波特率为 9600bps 的时，USARTDIV 的值设为 78.125，在 12MHz 时可以得到 9600bps 的波特率，但是如果你需要 115200bps 的波特率时，计算的 USARTDIV 是 6.51，但是实际上在 USART_BRR 里填入的值最接近只能是 6.5，实际产生的波特率是 115384bps，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%。

19.4 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 19-2。

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 19-3。

图 19-2 USART 时钟时序示例 (M=0)

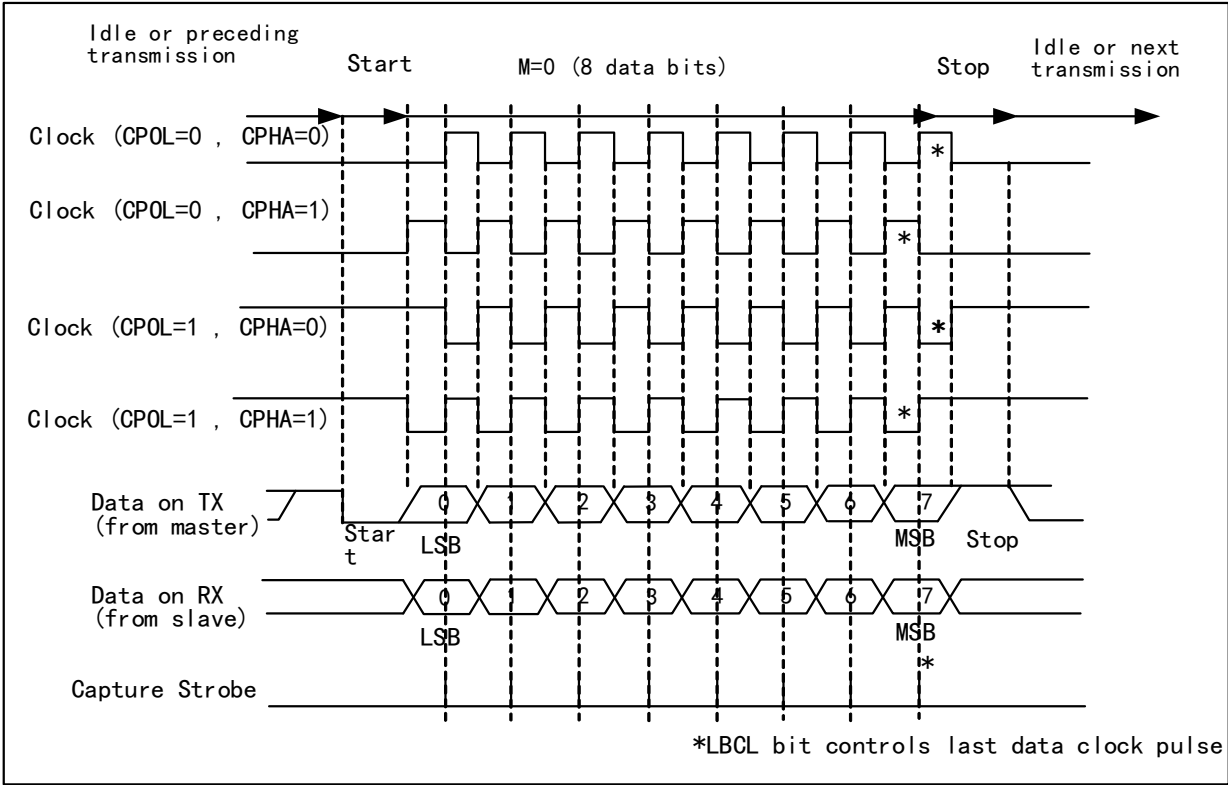
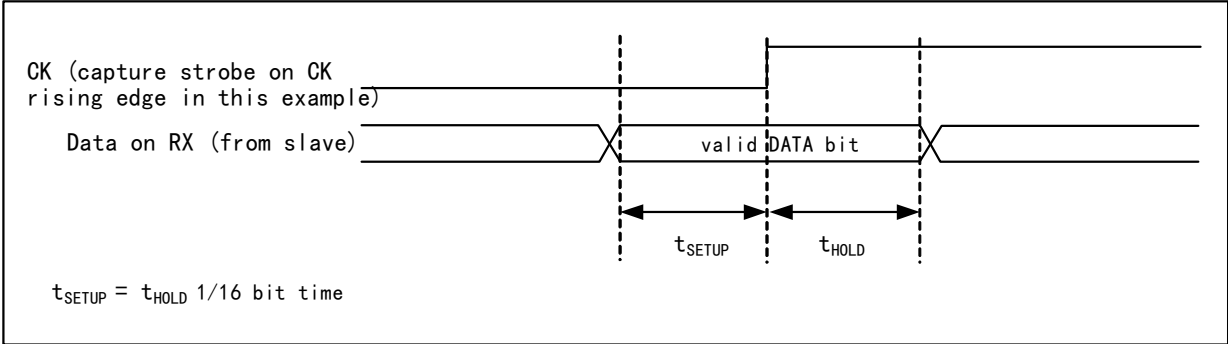


图 19-3 数据采样保持时间



19.5 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。

19.6 智能卡

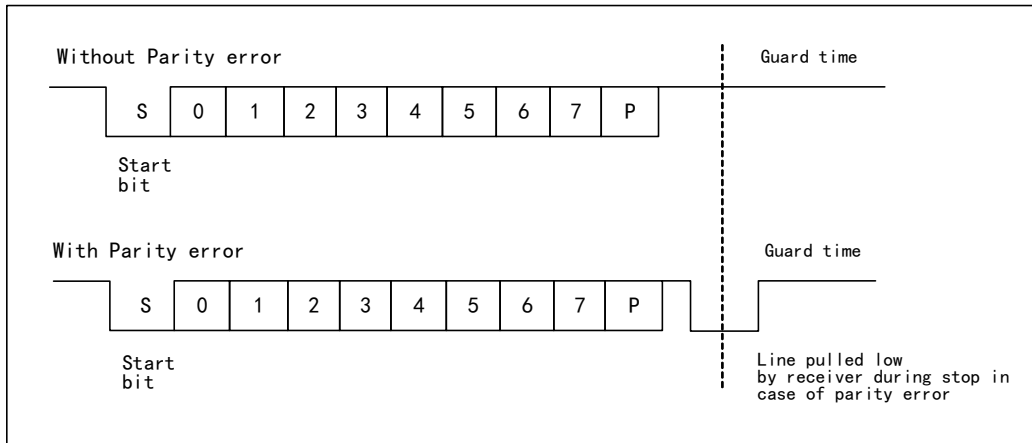
智能卡模式支持 ISO7816-3 协议访问智能卡控制器。

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 19-4 （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 HB 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）。

19.7 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 USART 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。

19.8 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。

19.9 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（RXNE）、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（ORE）和帧错误（FE）等等。

表 19-1 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 (RXNE)	RXNEIE
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 (ORE)	
多缓冲通信的帧错误 (FE)	

19.10 低功耗唤醒

通过 USARTx_CTLR3 寄存器 LPWKUP_EN 位使能低功耗唤醒功能, 并由 USARTx_STATR 寄存器的 LPWKUP_ACT_FLAG 位读取低功耗唤醒开始的指示标志; 低功耗唤醒的时钟源通过 USARTx_CTLR3 寄存器 LPWKUP_CK_SRC 位来选择, 可选 LSE 或 LSI, LPWKUP_DLY_CFG 位可配置低功耗唤醒接收延迟的时钟源周期。

19.11 寄存器描述

表 19-2 USART1 相关寄存器列表

名称	访问地址	描述	复位值
R32_USART1_STATR	0x40013800	USART1 状态寄存器	0x000000C0
R32_USART1_DATAR	0x40013804	USART1 数据寄存器	0x00000000
R32_USART1_BRR	0x40013808	USART1 波特率寄存器	0x00000000
R32_USART1_CTLR1	0x4001380C	USART1 控制寄存器 1	0x00000000
R32_USART1_CTLR2	0x40013810	USART1 控制寄存器 2	0x00000000
R32_USART1_CTLR3	0x40013814	USART1 控制寄存器 3	0x00000000
R32_USART1_GPR	0x40013818	USART1 保护时间和预分频寄存器	0x00000000
R32_USART1_CTLR4	0x4001381C	USART1 控制寄存器 4	0x00000000

表 19-3 USART2 相关寄存器列表

名称	访问地址	描述	复位值
R32_USART2_STATR	0x40004400	USART2 状态寄存器	0x000000C0
R32_USART2_DATAR	0x40004404	USART2 数据寄存器	0x00000000
R32_USART2_BRR	0x40004408	USART2 波特率寄存器	0x00000000
R32_USART2_CTLR1	0x4000440C	USART2 控制寄存器 1	0x00000000
R32_USART2_CTLR2	0x40004410	USART2 控制寄存器 2	0x00000000
R32_USART2_CTLR3	0x40004414	USART2 控制寄存器 3	0x00000000
R32_USART2_GPR	0x40004418	USART2 保护时间和预分频寄存器	0x00000000
R32_USART2_CTLR4	0x4000441C	USART2 控制寄存器 4	0x00000000

表 19-4 USART3 相关寄存器列表

名称	访问地址	描述	复位值
R32_USART3_STATR	0x40004800	USART3 状态寄存器	0x000000C0
R32_USART3_Datar	0x40004804	USART3 数据寄存器	0x00000000
R32_USART3_BRR	0x40004808	USART3 波特率寄存器	0x00000000
R32_USART3_CTLR1	0x4000480C	USART3 控制寄存器 1	0x00000000
R32_USART3_CTLR2	0x40004810	USART3 控制寄存器 2	0x00000000
R32_USART3_CTLR3	0x40004814	USART3 控制寄存器 3	0x00000000
R32_USART3_GPR	0x40004818	USART3 保护时间和预分频寄存器	0x00000000
R32_USART3_CTLR4	0x4000481C	USART3 控制寄存器 4	0x00000000

表 19-5 USART4 相关寄存器列表

名称	访问地址	描述	复位值
R32_USART4_STATR	0x40004C00	USART4 状态寄存器	0x000000C0
R32_USART4_Datar	0x40004C04	USART4 数据寄存器	0x00000000
R32_USART4_BRR	0x40004C08	USART4 波特率寄存器	0x00000000
R32_USART4_CTLR1	0x40004C0C	USART4 控制寄存器 1	0x00000000
R32_USART4_CTLR2	0x40004C10	USART4 控制寄存器 2	0x00000000
R32_USART4_CTLR3	0x40004C14	USART4 控制寄存器 3	0x00000000
R32_USART4_GPR	0x40004C18	USART4 保护时间和预分频寄存器	0x00000000
R32_USART4_CTLR4	0x40004C1C	USART4 控制寄存器 4	0x00000000

表 19-6 USART5 相关寄存器列表

名称	访问地址	描述	复位值
R32_USART5_STATR	0x40005000	USART5 状态寄存器	0x000000C0
R32_USART5_Datar	0x40005004	USART5 数据寄存器	0x00000000
R32_USART5_BRR	0x40005008	USART5 波特率寄存器	0x00000000
R32_USART5_CTLR1	0x4000500C	USART5 控制寄存器 1	0x00000000
R32_USART5_CTLR2	0x40005010	USART5 控制寄存器 2	0x00000000
R32_USART5_CTLR3	0x40005014	USART5 控制寄存器 3	0x00000000
R32_USART5_GPR	0x40005018	USART5 保护时间和预分频寄存器	0x00000000
R32_USART5_CTLR4	0x4000501C	USART5 控制寄存器 4	0x00000000

表 19-7 USART6 相关寄存器列表

名称	访问地址	描述	复位值
R32_USART6_STATR	0x40001800	USART6 状态寄存器	0x000000C0
R32_USART6_Datar	0x40001804	USART6 数据寄存器	0x00000000
R32_USART6_BRR	0x40001808	USART6 波特率寄存器	0x00000000
R32_USART6_CTLR1	0x4000180C	USART6 控制寄存器 1	0x00000000
R32_USART6_CTLR2	0x40001810	USART6 控制寄存器 2	0x00000000
R32_USART6_CTLR3	0x40001814	USART6 控制寄存器 3	0x00000000
R32_USART6_GPR	0x40001818	USART6 保护时间和预分频寄存器	0x00000000
R32_USART6_CTLR4	0x4000181C	USART6 控制寄存器 4	0x00000000

表 19-8 USART7 相关寄存器列表

名称	访问地址	描述	复位值
R32_USART7_STATR	0x40001C00	USART7 状态寄存器	0x000000C0
R32_USART7_DATAR	0x40001C04	USART7 数据寄存器	0x00000000
R32_USART7_BRR	0x40001C08	USART7 波特率寄存器	0x00000000
R32_USART7_CTLR1	0x40001C0C	USART7 控制寄存器 1	0x00000000
R32_USART7_CTLR2	0x40001C10	USART7 控制寄存器 2	0x00000000
R32_USART7_CTLR3	0x40001C14	USART7 控制寄存器 3	0x00000000
R32_USART7_GPR	0x40001C18	USART7 保护时间和预分频寄存器	0x00000000
R32_USART7_CTLR4	0x40001C1C	USART7 控制寄存器 4	0x00000000

表 19-9 USART8 相关寄存器列表

名称	访问地址	描述	复位值
R32_USART8_STATR	0x40002000	USART8 状态寄存器	0x000000C0
R32_USART8_DATAR	0x40002004	USART8 数据寄存器	0x00000000
R32_USART8_BRR	0x40002008	USART8 波特率寄存器	0x00000000
R32_USART8_CTLR1	0x4000200C	USART8 控制寄存器 1	0x00000000
R32_USART8_CTLR2	0x40002010	USART8 控制寄存器 2	0x00000000
R32_USART8_CTLR3	0x40002014	USART8 控制寄存器 3	0x00000000
R32_USART8_GPR	0x40002018	USART8 保护时间和预分频寄存器	0x00000000
R32_USART8_CTLR4	0x4000201C	USART8 控制寄存器 4	0x00000000

19.11.1 USART 状态寄存器 (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
LPWKUP_ACT_FLAG	Reserved			MS_ERR	RX_BUSY	CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
15	LPWKUP_ACT_FLAG	RW1Z	低功耗唤醒开始指示位, 写 1 清零。	0
[14:12]	Reserved	R0	保留。	0
11	MS_ERR	R0	MARK 或 SPACE 校验错误标志。在接收模式下, 如果产生 MARK 或 SPACE 校验错误, 硬件置位此位。读此位再读数据寄存器的操作会复位此位。如果之前 MS_ERRIE 被置位, 那么此位被置位会产生相应中断。	0
10	RX_BUSY	R0	接收状态指示位: 1: 工作在接收状态时, 此位置位; 0: 处于非接收状态。	0

9	CTS	RW0	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	RW0	LIN Break 检测标志。当检测到 LIN Break 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 Break; 0: 没有检测待 LIN Break。	0
7	TXE	RO	发送数据寄存器空标志。当 TDR 寄存器中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	RW0	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	RW0	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。	0
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	0

			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。	
2	NE	R0	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	R0	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	R0	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

19.11.2 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	R0	保留。	0
[8:0]	DR[8:0]	RW	数据寄存器。这个寄存器实际上是接收数据寄存器（RDR）和发送数据寄存器（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）。	0

19.11.3 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_Mantissa[11:0]												DIV_Fraction[3:0]			

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:4]	DIV_Mantissa [11:0]	RW	这 12 位定义了分频器除法因子的整数部分。	0
[3:0]	DIV_Fraction[3:0]	RW	这 4 位定义了分频器除法因子的小数部分。	0

19.11.4 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
M_EXT	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK	

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:14]	M_EXT	RW	字长扩展位: 00: 无效, M 位决定数据长度。 01: 7 个数据位; 10: 6 个数据位; 11: 5 个数据位。	0
13	UE	RW	USART 使能位。 1: 使能 USART; 0: 禁止 USART。 当此位被清零后, 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。 1: 校验位使能; 0: 校验位关闭。 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： 1：奇校验； 0：偶校验。 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。 1：允许产生奇偶检验错误中断； 0：禁止产生奇偶检验错误中断。	0
7	TXEIE	RW	发送缓冲区空中断使能： 1：允许产生发送缓冲区空中断； 0：禁止产生发送缓冲区空中断。	0
6	TCIE	RW	发送完成中断使能： 1：允许产生发送完成中断； 0：禁止产生发送完成中断。	0
5	RXNEIE	RW	接收缓冲区非空中断使能： 1：允许产生接收缓冲区非空中断； 0：禁止产生接收缓冲区非空中断。	0
4	IDLEIE	RW	总线空闲中断使能： 1：允许产生总线空闲中断； 0：禁止产生总线空闲中断。	0
3	TE	RW	发送使能。 1：使能发送器； 0：关闭发送器。	0
2	RE	RW	接收使能。 1：使能接收器，接收器开始检测 RX 引脚上的起始位； 0：关闭接收器。	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

19.11.5 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[1:0]	CLKEN	CPOL	CPHA	LBCL	Reserved	LBDIE	LBDL	Reserved	ADD[3:0]				

位	名称	访问	描述	复位值
[31:15]	Reserved	RO	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 同步断开符号，以及检测 LIN 同步断开符。	0
[13:12]	STOP[1:0]	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。	00b
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 Break 检测中断使能： 1: 使能 LBD 引起的中断； 0: 关闭 LBD 引起的中断。	0
5	LBDL	RW	LIN Break 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[3:0]	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

19.11.6 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
LPWKUP_DLY_CFG [2:0]	LPWKUP_CK_SRC	LPWKUP_EN	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE		

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:13]	LPWKUP_DLY_CFG [2:0]	RW	低功耗唤醒接收延迟配置： 000：无延迟； 001：延迟 1 个低功耗唤醒时钟源周期； ... 111：延迟 7 个低功耗唤醒时钟源周期。	0
12	LPWKUP_CK_SRC	RW	低功耗唤醒时钟源选择： 1：LSE； 0：LSI。	0
11	LPWKUP_EN	RW	低功耗唤醒使能： 1：使能低功耗唤醒； 0：关闭低功耗唤醒。	0
10	CTSIE	RW	CTSIE 中断使能位： 1：在 CTS 被置位时会产生中断； 0：在 CTS 被置位时不产生中断。	0
9	CTSE	RW	CTS 使能位： 1：使能 CTS 流控； 0：关闭 CTS 流控。	0
8	RTSE	RW	RTS 使能位： 1：使能 RTS 流控； 0：关闭 RTS 流控。	0
7	DMAT	RW	DMA 发送使能位： 1：在发送时使用 DMA； 0：在发送时不使用 DMA。	0
6	DMAR	RW	DMA 接收使能位： 1：在接收时使用 DMA； 0：在接收时不使用 DMA。	0
5	SCEN	RW	智能卡模式使能位： 1：使能智能卡模式；	0

			0: 关闭智能卡模式。	
4	NACK	RW	智能卡 NACK 使能位: 1: 在校验错误出现时, 发送 NACK; 0: 在校验错误出现时, 不发送 NACK。	0
3	HDSEL	RW	半双工模式选择位: 1: 开启半双工模式; 0: 关闭半双工模式。	0
2	IRLP	RW	红外低功耗选择位: 1: 在选择红外线时, 启用低功耗模式; 0: 在选择红外线时, 关闭低功耗模式。	0
1	IREN	RW	红外线使能位: 1: 使能红外模式; 0: 关闭红外模式。	0
0	EIE	RW	错误中断位使能: 1: 使能错误中断; 0: 关闭错误中断。 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

19.11.7 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[7:0]	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[7:0]	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

19.11.8 USART 控制寄存器 4 (USARTx_CTLR4) (x=1/2/3/4/5/6/7/8)

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												CHECK_SEL [1:0]	MS_ERRIE	Reserved	

位	名称	访问	描述	复位值
[31:4]	Reserved	R0	保留。	0
[3:2]	CHECK_SEL[1:0]	RW	校验功能选择位： 0x：关闭 MARK 和 SPACE 校验； 10：使能 MARK 校验； 11：使能 SPACE 校验。	0
1	MS_ERRIE	RW	SPACE 或 MARK 校验使能位： 1：允许产生 SPACE 或 MARK 校验错误中断； 0：禁止产生 SPACE 或 MARK 校验错误中断。	0
0	Reserved	R0	保留。	0

第 20 章 串并互转控制器及收发器 (SerDes)

本章模块描述仅适用于 CH32H417 微控制器产品。

系统内置了一组支持信号隔离和远距离传输的 SerDes 模块，该模块包含两个独立的 SerDes 控制器，支持 1.5Gbps 高速差分信号 (SERDES_RXP、SERDES_RXN、SERDES_TXP、SERDES_TXN 引脚)，可以通过光纤模块或网线中的一个差分对等传输媒体，进行远距离数据传输。

SerDes 模块集成全双工 PHY (物理层接口) 及两路独立控制单元，其中两路控制单元可分别与 PHY 的接收 (Rx) 通道、发送 (Tx) 通道进行功能对接。需明确：单路控制单元不得同时对接收与发送通道，通过配置两个控制器到两个通道，可实现 SerDes 全双工通信功能。具体实现方案及配置流程，请参见官方提供的 EVT 参考例程。

两个控制器的基地址配置如下：

SerDes 控制器 1 基地址：0x40027C00

SerDes 控制器 2 基地址：0x40027C40

20.1 主要特征

- 可编程的数据收发速率，最高支持 1.5Gbps
- 支持网络变压器高压信号隔离，也支持电容低压隔离
- 内置 8bit/10bit 编解码和 CRC 校验，支持序列号匹配
- 内置 FIFO，支持收发双缓冲模式
- 支持 DMA 功能，访问地址支持字节对齐
- 提供多种传输中断标志和状态，及时向应用层反馈信息
- 差分收发，可直接驱动光纤模块

20.2 功能描述

20.2.1 发送模式 (查询模式)

1. 设置 SERDESx_CTRL 寄存器
ALIGN_EN/PWR_UP/TX_PWR_UP/LINK_RESET/ALL_CLR 为 1；
2. 等待 PLL 锁定 (PLL_LOCK=1)；
3. 设置 SERDESx_CTRL 寄存器
ALIGN_EN/PWR_UP/TX_PWR_UP/INT_BUSY/DMA_EN/TX_EN 为 1；
4. 设置 LINK_INIT 为 1，启动发送初始化包，延迟 100ms (软件延迟)，清除 LINK_INIT；
5. 设置发送 DMA 地址 (单缓冲配置 DMA0 即可)；
6. 配置 SERDESx_DATA0 (单缓冲模式)；
7. 配置 SERDESx_TX_LEN；
8. 查询 LINK_FREE 为 1 (可选)；
9. 置 TX_VLD 为 1，启动发送 (5、6、7 顺序可以任意)；
10. 查询发送完成中断标志为 1，关闭发送，查询 TX_VLD 为 0 时清除中断标志。

注：步骤 4 的软件延迟时间以确保接收方准备好接收即可。

20.2.2 接收模式 (中断模式)

1. 设置 SERDESx_CTRL 寄存器
PWR_UP/RX_PWR_UP/LINK_RESET/ALL_CLR 为 1；
2. 等待 PLL 锁定 (PLL_LOCK=1)；
3. 设置 SERDESx_CTRL 寄存器

PWR_UP/TX_PWR_UP/DMA_EN/RX_EN 为 1；

4. 设置接收 DMA 地址（建议使用双缓冲模式）；
5. 设置中断使能 RECV_DONE_IE/COMINIT_IE/FIFO_OV_IE；
6. 接收完成后查询 RECV_CRC_OK/RX_SEQ_MATCH，如果双缓冲模式，需要查询 RX_SEQ_NUM 最低位得知当前接收包使用的寄存器 DATA0/DMA0/RX_LEN0 或 DATA1/DMA1/RX_LEN1；
RX_SEQ_NUM[0] 为 1，表示当前接收包使用 DATA0/DMA0/RX_LEN0；
RX_SEQ_NUM[0] 为 0，表示当前接收包使用 DATA1/DMA1/RX_LEN1。

注：RX_SEQ_NUM 在接收完成后自动加 1，所以接收完成查询到的 RX_SEQ_NUM 减 1 为当前接收包的 SEQ_NUM

20.2.3 软件注意事项：

- 1、控制器内部的时钟频率 SERDES_CLK 为 PLL 频率除以 40，例如 1.2G 时 SERDES_CLK 为 30MHz；
- 2、LINK_RESET、RX_EN、TX_VLD 的返回值经 SERDES_CLK 同步，读出会有延迟；
- 3、发送或接收的数据包长度不包括 DATA0/DATA1 的 4Bytes 长度；
- 4、发送完成后应先清零 TX_VLD，查询该位返回值为 0 后才能清除发送完成中断标志，否则会有异步问题。

20.3 寄存器描述

表 20-1 SERDES1 相关寄存器列表

名称	访问地址	描述	复位值
R32_SERDES1_CTRL	0x40027C00	SERDES1 控制寄存器	0x00041207
R32_SERDES1_INT_EN	0x40027C04	SERDES1 中断使能寄存器	0x00000000
R32_SERDES1_INT_FS	0x40027C08	SERDES1 中断标志/状态寄存器	0x00000000
R32_SERDES1_RTX_CTRL	0x40027C0C	SERDES1 收发控制器	0x00000000
R32_SERDES1_RX_LEN0	0x40027C10	SERDES1 接收长度 0 寄存器	0x00000000
R32_SERDES1_DATA0	0x40027C14	SERDES1 数据 0 缓冲区寄存器	0xFFFFFFFF
R32_SERDES1_DMA0	0x40027C18	DMA0 缓冲区的起始地址寄存器	0xFFFFFFFF
R32_SERDES1_RX_LEN1	0x40027C1C	SERDES1 接收长度 1 寄存器	0x00000000
R32_SERDES1_DATA1	0x40027C20	SERDES1 数据 1 缓冲区寄存器	0xFFFFFFFF
R32_SERDES1_DMA1	0x40027C24	DMA1 缓冲区的起始地址寄存器	0xFFFFFFFF

表 20-2 SERDES2 相关寄存器列表

名称	访问地址	描述	复位值
R32_SERDES2_CTRL	0x40027C40	SERDES2 控制寄存器	0x00041207
R32_SERDES2_INT_EN	0x40027C44	SERDES2 中断使能寄存器	0x00000000
R32_SERDES2_INT_FS	0x40027C48	SERDES2 中断标志/状态寄存器	0x00000000
R32_SERDES2_RTX_CTRL	0x40027C4C	SERDES2 收发控制器	0x00000000
R32_SERDES2_RX_LEN0	0x40027C50	SERDES2 接收长度 0 寄存器	0x00000000
R32_SERDES2_DATA0	0x40027C54	SERDES2 数据 0 缓冲区寄存器	0xFFFFFFFF
R32_SERDES2_DMA0	0x40027C58	DMA0 缓冲区的起始地址寄存器	0xFFFFFFFF
R32_SERDES2_RX_LEN1	0x40027C5C	SERDES2 接收长度 1 寄存器	0x00000000
R32_SERDES2_DATA1	0x40027C60	SERDES2 数据 1 缓冲区寄存器	0xFFFFFFFF
R32_SERDES2_DMA1	0x40027C64	DMA1 缓冲区的起始地址寄存器	0xFFFFFFFF

表 20-3 SERDES 相关寄存器列表

名称	访问地址	描述	复位值
R32_SYS_CFGR	0x5003C010	系统配置寄存器	0x0054422B

20.3.1 SERDESx 控制寄存器 (R32_SERDESx_CTRL) (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													ALIGN_EN	CONT_EN	Reserved
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_PWR_UP	RX_PWR_UP	Reserved						DMA_EN	TX_EN	RX_EN	RX_POLARITY	INT_BUSY_EN	RESET_PHY	RESET_LINK	CLR_ALL

位	名称	访问	描述	复位值
[31:19]	Reserved	RO	保留。	0
18	ALIGN_EN	RW	发送时插入 ALIGN 信号, 为了解决频偏, 高有效。	1
17	CONT_EN	RW	将发送空闲时的 SYNC 用 CONT 和扰码数据代替, 降低 EMI。	0
16	Reserved	RO	保留。	0
15	TX_PWR_UP	RW	发送模块上电使能位: 1: 使能; 0: 关闭。	0
14	RX_PWR_UP	RW	接收模块上电使能位: 1: 使能; 0: 关闭。	0
[13:8]	Reserved	RO	保留。	0x12
7	DMA_EN	RW	DMA 使能: 1: 使能; 0: 关闭。	0
6	TX_EN	RW	发送使能: 1: 使能; 0: 关闭。	0
5	RX_EN	RW	接收使能: 1: 使能; 0: 关闭。	0
4	RX_POLARITY	RW	RXN/RXP 信号交换。	0
3	INT_BUSY_EN	RW	SERDES 发送完成中断标志未清零前自动暂停发送使能位: 1: 使能; 0: 关闭。	0
2	RESET_PHY	RW	SERDES 物理层软件复位控制位:	1

			1: 使能; 0: 关闭。	
1	RESET_LINK	RW	SERDES 链路层软件复位控制位: 1: 使能; 0: 关闭。	1
0	CLR_ALL	RW	清空 SERDES 中断标志和 FIFO: 1: 使能; 0: 关闭。	1

20.3.2 SERDESx 中断使能寄存器 (R32_SERDESx_INT_EN) (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										COMINIT_IE	Reserved	FIFO_OV_IE	RECV_DONE_IE	TRAN_DONE_IE RECV_ERR_IE	PHYRDY_IE

位	名称	访问	描述	复位值
[31:6]	Reserved	RO	保留。	0
5	COMINIT_IE	RW	COMINIT 接收中断使能: 1: 使能; 0: 关闭。	0
4	Reserved	RO	保留。	0
3	FIFO_OV_IE	RW	FIFO 溢出中断使能: 1: 使能; 0: 关闭。	0
2	RECV_DONE_IE	RW	接收完成中断使能: 1: 使能; 0: 关闭。	0
1	TRAN_DONE_IE RECV_ERR_IE	RW	发送模式: 发送完成中断使能: 1: 使能; 0: 关闭。 接收模式: 接收 CRC 错误中断使能: 1: 使能; 0: 关闭。	0
0	PHYRDY_IE	RW	物理层 READY 中断使能: 1: 使能; 0: 关闭。	0

20.3.3 SERDESx 中断标志/状态寄存器 (R32_SERDESx_INT_FS) (x=1/2)

偏移地址: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TX_SEQ_NUM[3:0]				RX_SEQ_NUM[3:0]				Reserved		R_FIFO_RDY	LINK_FREE	PLL_LOCK	RCV_CRC_OK	RX_SEQ_MATCH	PHYRDY
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										COMINIT_IF	Reserved	FIFO_OV_IF	RCV_DONE_IF	TRAN_DONE_IF RCV_ERR_IF	PHYRDY_IF

位	名称	访问	描述	复位值
[31:28]	TX_SEQ_NUM[3:0]	R0	LINK_INIT 有效时自动清零。	0
[27:24]	RX_SEQ_NUM[3:0]	R0	接收模式下, 接收到 COMINT 自动清零。	0
[23:22]	Reserved	R0	保留。	0
21	R_FIFO_RDY	R0	SERDES 接收 FIFO 数据就绪状态: 1: 接收 FIFO 非空; 0: 接收 FIFO 空。	0
20	LINK_FREE	R0	SERDES 链路层发送空闲状态位。	0
19	PLL_LOCK	R0	PLL 锁定状态位, 该位为 1 后开始收发。	0
18	RCV_CRC_OK	R0	接收数据帧校验状态位, RCV_DONE 中断后检测该位, 在包接收完成后更新。	0
17	RX_SEQ_MATCH	R0	接收 SEQ_NUM 匹配状态位。	0
16	PHYRDY	R0	物理层 READY 状态位。	0
[15:6]	Reserved	R0	保留。	0
5	COMINIT_IF	RW1Z	COMINIT 接收中断标志位。	0
4	Reserved	R0	保留。	0
3	FIFO_OV_IF	RW1Z	FIFO 溢出中断标志位。	0
2	RCV_DONE_IF	RW1Z	接收完成中断标志位。	0
1	TRAN_DONE_IF RCV_ERR_IF	RW1Z	发送模式: 发送完成中断标志位。 接收模式: 接收 CRC 错误中断标志位。	0
0	PHYRDY_IF	RW1Z	物理层 READY 中断标志位。	0

20.3.4 SERDESx 收发控制器 (R32_SERDESx_RTX_CTRL) (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													BUF_MODE	TX_VLD	LINK_INIT
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERDES_TX_LEN[15:0]															

位	名称	访问	描述	复位值

[31:19]	Reserved	RO	保留。	0
18	BUF_MODE	RW	双缓冲控制位： 1：使能； 0：关闭。	0
17	TX_VLD	RW	发送数据包有效，由软件清零，该位返回和 PHY_CLK 同步。	0
16	LINK_INIT	RW	该位为 1，发送 LINK 初始化包。	0
[15:0]	SERDES_TX_LEN [15:0]	RW	发送的数据字节数，低 2 位固定为 0（4 字节对齐）	x

20.3.5 SERDESx 接收长度 0 寄存器（R32_SERDESx_RX_LEN0）（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
SERDES_RX_LEN0[15:0]															

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	SERDES_RX_LEN0 [15:0]	RO	最后接收到的数据的包长，最低 2 位固定为 0。 非双缓冲模式下，表示当前接收的包长； 双缓冲模式下，RX_SEQ_NUM 为偶数的包长度。	0

20.3.6 SERDESx 数据 0 缓冲区寄存器（R32_SERDESx_DATA0）（x=1/2）

偏移地址：0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SERDES_DATA0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERDES_DATA0[15:0]															

位	名称	访问	描述	复位值
[31:0]	SERDES_DATA0 [31:0]	RW	SERDES 数据缓存区 0 发送模式：LINK 收发的第一个 4Byte 数据存放在该寄存器，高 4bit 被替换成 TX_SEQ_NUM，用户可以使用低 28bit 作为自定义信息。 接收模式：数据包的前 4Byte 数据存入该寄存器，后面的数据存入 DMA。	x

20.3.7 DMA0 缓冲区的起始地址寄存器（R32_SERDESx_DMA0）（x=1/2）

偏移地址：0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SERDES_DMA0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERDES_DMA0[15:0]															

位	名称	访问	描述	复位值
[31:0]	SERDES_DMA0 [31:0]	RW	缓冲区 0 起始地址，接收模式下，软件需要将低 4 位固定位 0（16 字节对齐），发送模式不需要 16 字节对齐。 注： 1. 单缓冲区模式下，只使用 DMA0； 2. 在双缓冲区模式下，TX_SEQ_NUM/RX_SEQ_NUM 为偶数时使用该 DMA0。	X

20.3.8 SERDESx 接收长度 1 寄存器（R32_SERDESx_RX_LEN1）（x=1/2）

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
SERDES_RX_LEN1[15:0]															

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
[15:0]	SERDES_RX_LEN1 [15:0]	RO	最后接收到的数据的包长，最低 2 位固定为 0，只有在双缓冲模式下，RX_LEN1 才有效。表示 RX_SEQ_NUM 为奇数的包长度。	0

20.3.9 SERDESx 数据 1 缓冲区寄存器（R32_SERDESx_DATA1）（x=1/2）

偏移地址：0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SERDES_DATA1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERDES_DATA1[15:0]															

位	名称	访问	描述	复位值
[31:0]	SERDES_DATA1 [31:0]	RW	SERDES 数据缓存区 1 发送模式：LINK 收发的第一个 4B 数据存放在该寄存器，高 4bit 被替换成 TX_SEQ_NUM，用户可以使用低 28bit 作为自定义信息。 接收模式：数据包的前 4Byte 数据存入该寄存器，后面的数据存入 DMA。	X

			注：在双缓冲区模式下，且 RX_SEQ_NUM 为奇数时使用该缓冲区。	

20.3.10 DMA1 缓冲区的起始地址寄存器 (R32_SERDESx_DMA1) (x=1/2)

偏移地址：0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SERDES_DMA1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERDES_DMA1[15:0]															

位	名称	访问	描述	复位值
[31:0]	SERDES_DMA1	RW	缓冲区 0 起始地址： 接收模式下，软件需要将低 4 位固定位 0（16 字节对齐）； 发送模式不需要 16 字节对齐。 注：在双缓冲区模式下，TX_SEQ_NUM/RX_SEQ_NUM 为奇数时使用该 DMA。	X

20.3.11 系统配置寄存器 (R32_SYS_CFGR)

访问地址：0x5003C010

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_FBCAP_ADJ [1:0]		RX_FBRES_ADJ [1:0]		Reserved		TX_DE_EMPH ASIS[1:0]		TX_OUTPUT_SWING [1:0]		Reserved			

位	名称	访问	描述	复位值
[31:14]	Reserved	RW	保留，写入时必须保持原值。	0x151
[13:11]	RX_FBCAP_ADJ [1:0]	RW	接收器反馈电容调节位，详细参数参考 CH32H417DS0。	000b
[10:9]	RX_FBRES_ADJ [1:0]	RW	接收器反馈电阻调节位，详细参数参考 CH32H417DS0。	01b
[8:7]	Reserved	RW	保留，写入时必须保持原值。	00b
[6:5]	TX_DE_EMPHASIS [1:0]	RW	发送预加重幅度调节位，详细参数参考 CH32H417DS0。	01b
[4:3]	TX_OUTPUT_SWING [1:0]	RW	发送幅度调节位，详细参数参考 CH32H417DS0。	01b
[2:0]	Reserved	RW	保留，写入时必须保持原值。	011b

注：长线时建议将此寄存器的值调整为 0x542E7B。

第 21 章 两线通信总线（I2C）

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。

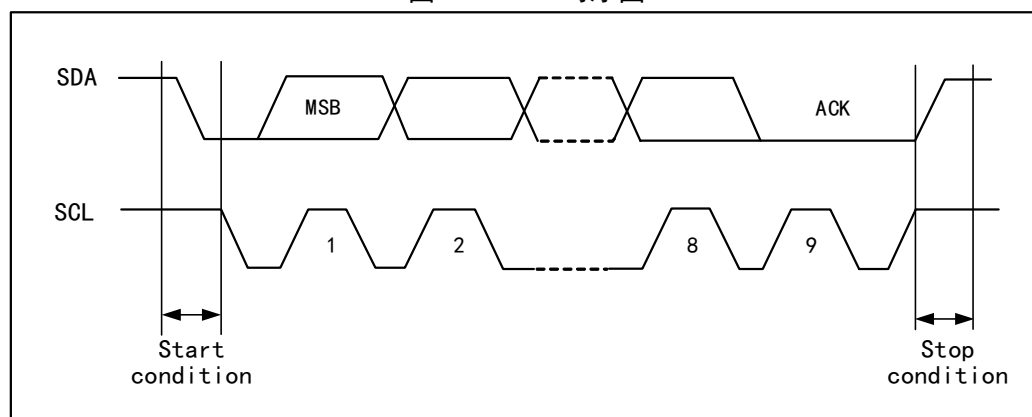
21.1 主要特征

- 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

21.2 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 21-1 所示。

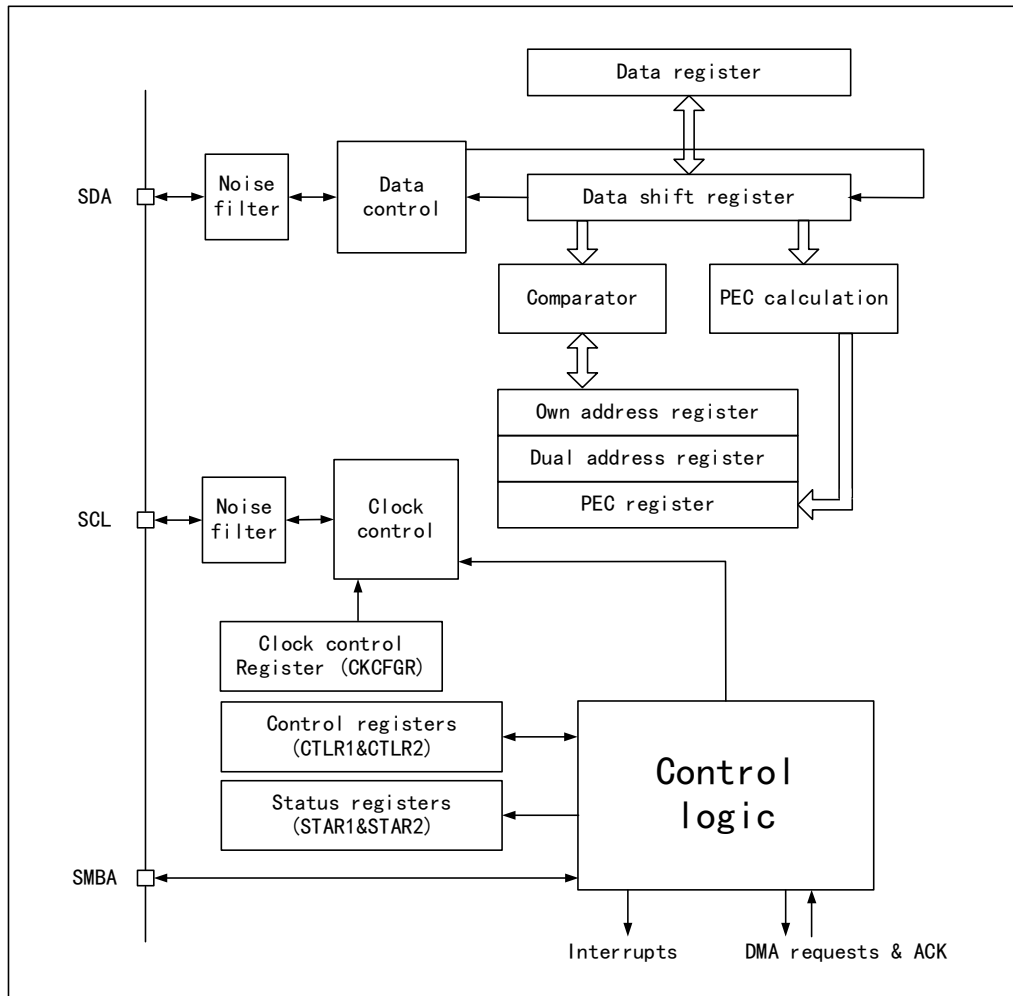
图 21-1 I2C 时序图



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 21-2 是 I2C 模块功能框图。

图 21-2 I2C 功能框图



21.3 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；

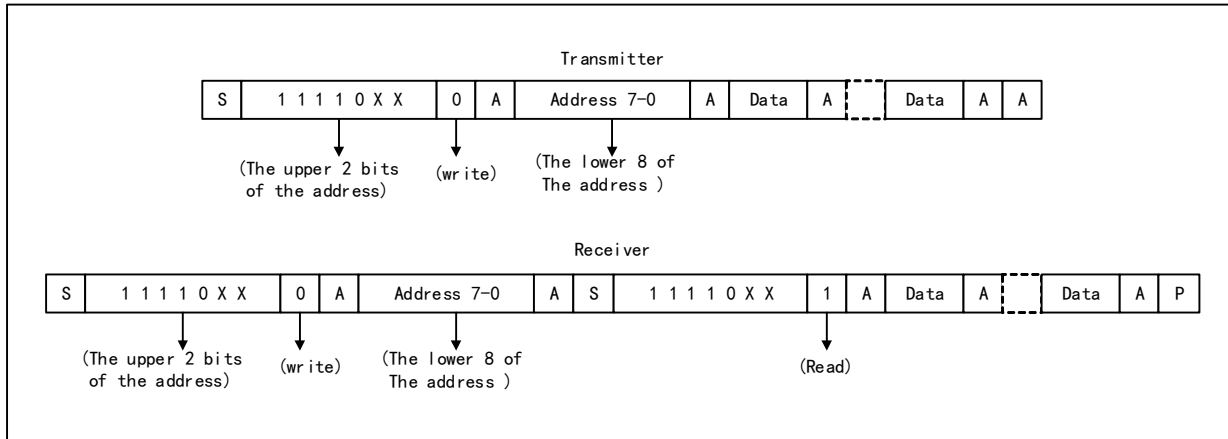
如果使用的是 7 位地址模式，那么写数据寄存器发送地址字节，在发送完地址字节后，状态寄存器的 ADDR 位会被置位，如果 ITEVTEN 位已经置位，则会产生中断，此时应读取 R16_I2Cx_STAR1 寄存

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 21-3 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 21-3 10 位地址时主机收发数据示意图

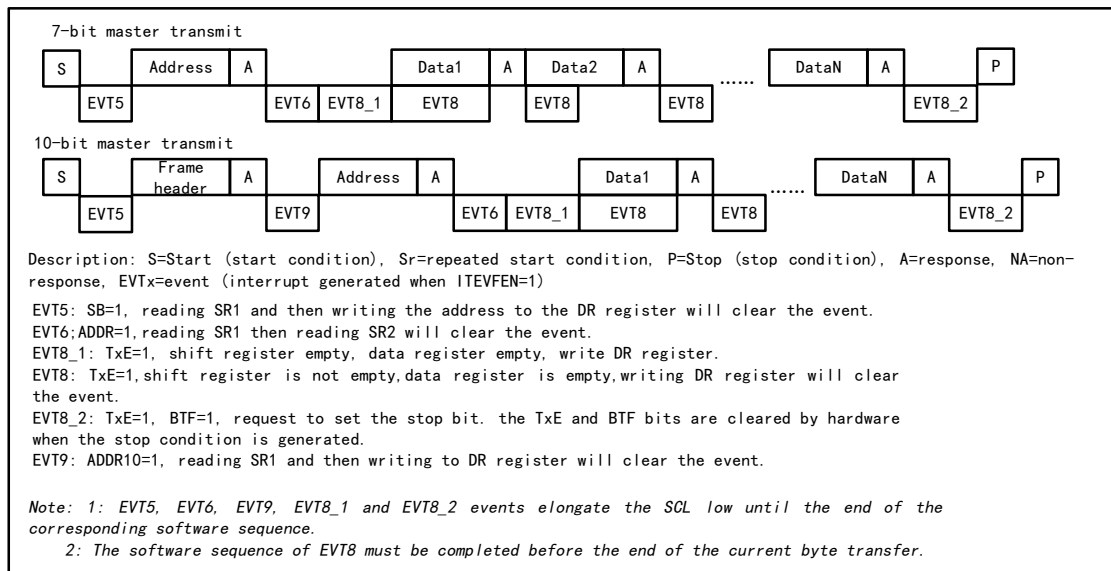


主发送模式：

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。

图 21-4 主发送器传送序列图

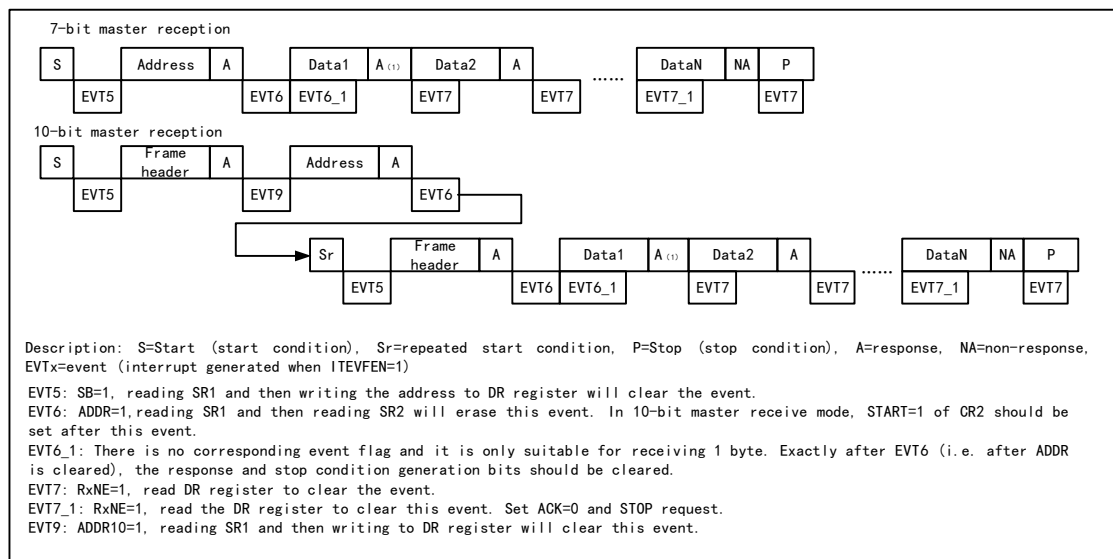


主接收模式：

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。

图 21-5 接收器传送序列图



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，I2C 将切换至从模式。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，接收到 NACK 后，从设备释放对 SCL 和 SDA 线的控制；主设备就可以发送一个停止/重起始条件。注意，产生停止条件后，I2C 模块将会自动切换至从模式。

21.4 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（ENGCG 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；如果第二个字节的地址也匹配或者 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；

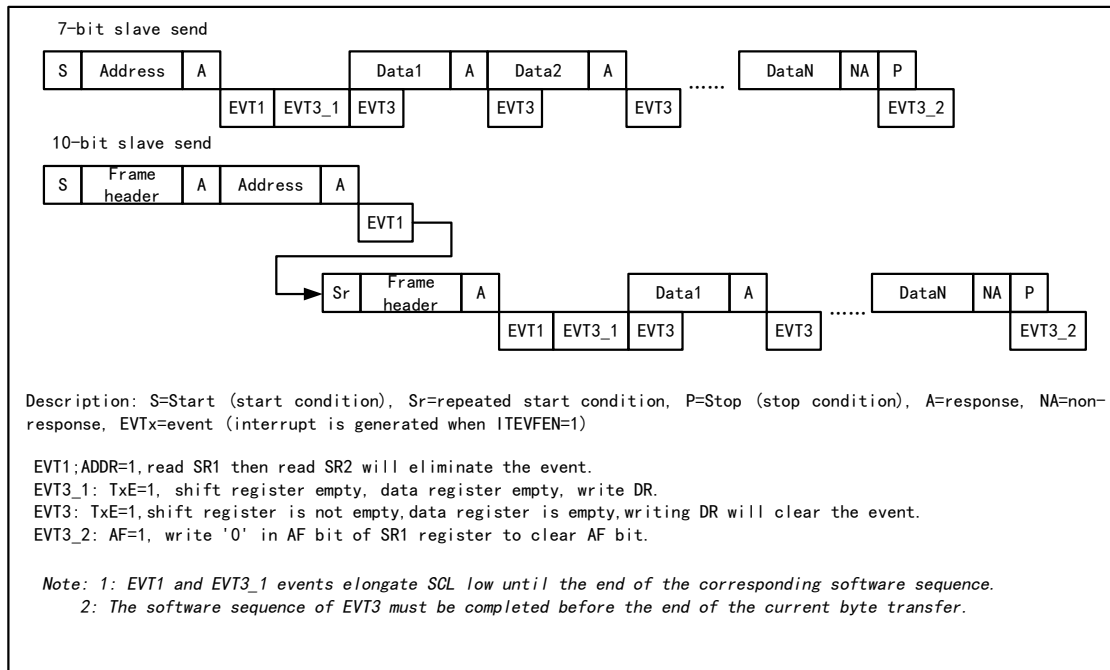
如果使用的是双地址模式（ENDUAL 位被置位），还需要读取 DUALF 位来判断主机唤起的是哪一个地址。

从模式默认是接收模式，在接收的头序列的最后一位为 1，或者 7 位地址最后一位为 1 后（取决于第一次接收到头序列还是普通的 7 位地址），当接收到重复的起始条件时，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。

从发送模式：

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。从设备保持 SCL 为低电平，直到 ADDR 位被清除且待发送数据已写入数据寄存器。（见下图中的 EVT1 和 EVT3）。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。

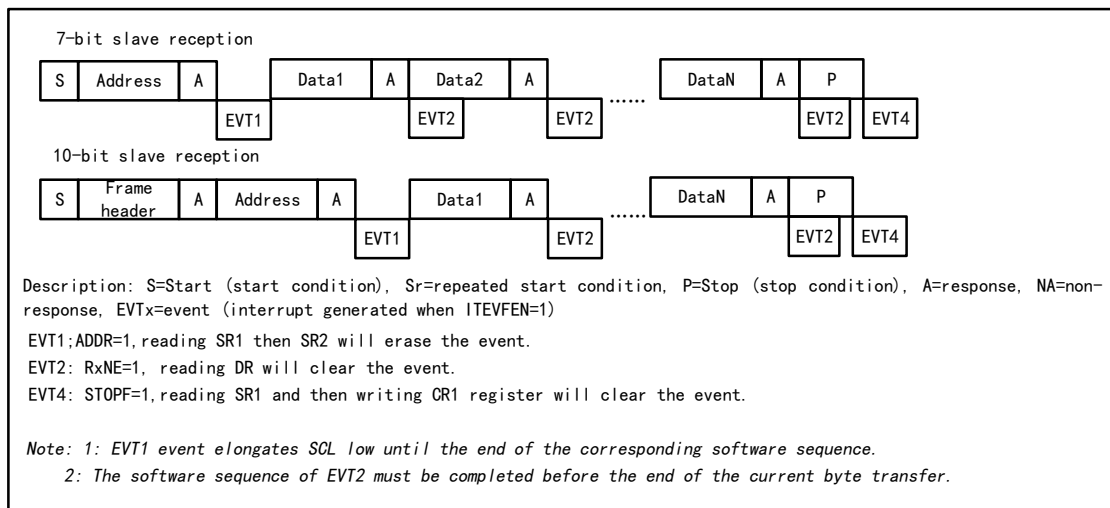
图 21-6 从发送器的传送序列图



从接收模式:

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。

图 21-7 从接收器的传送序列图



主设备在传输完最后一个数据字节后, 将产生一个停止条件, 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。(见上图中的 EVT4)。

21.5 错误

21.5.1 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。

21.5.2 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。

21.5.3 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。

21.5.4 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。

21.6 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。

21.7 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：

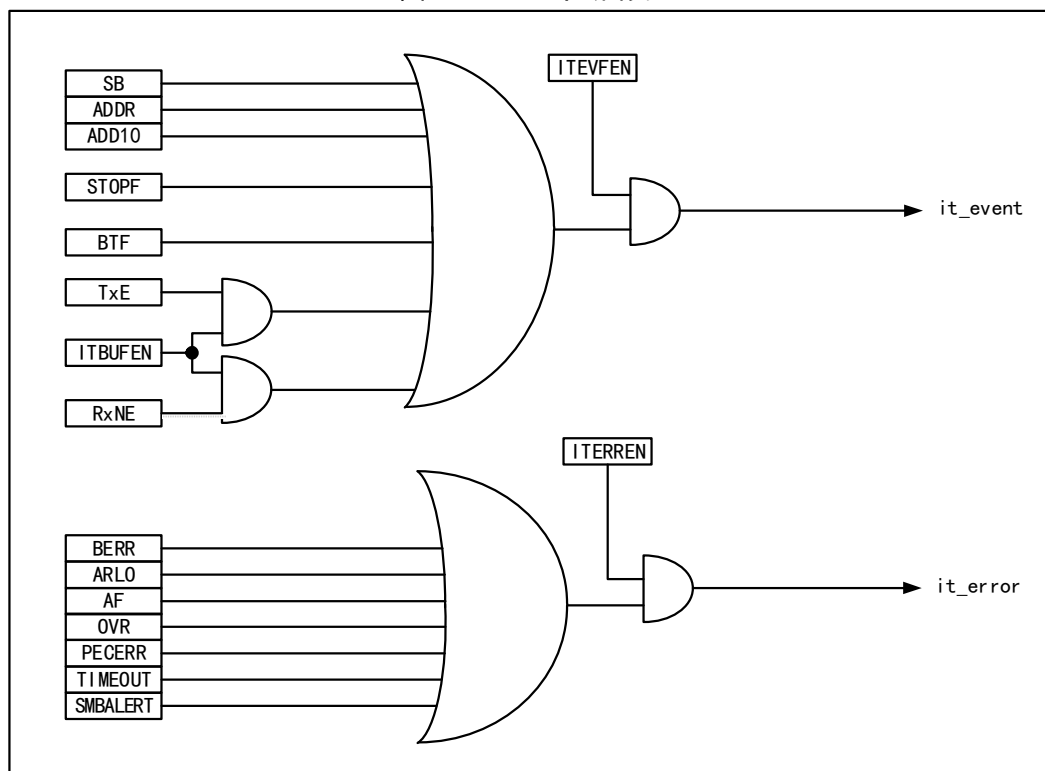
- 1) I2C 支持的速度最高 400kHz，而 SMBus 支持的最高是 100kHz，且 SMBus 有最小 10kHz 的速度限制；
- 2) SMBus 的时钟为低超过 35ms 时，会报超时，但 I2C 无此限制；
- 3) SMBus 有固定的逻辑电平，而 I2C 没有，取决于 V_{DD33} ；
- 4) SMBus 有总线协议，而 I2C 没有。

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。

21.8 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 21-8 的中断源。

图 21-8 I2C 中断请求



21.9 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 CTRL2 寄存器的 DMAEN 位置位可以激活 DMA 模式。只要 Tx E 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 Tx E 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 Tx E 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。

21.10 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。

21.11 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。

21.12 寄存器描述

表 21-1 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	0x40005420	I2C1 上升时间寄存器	0x0002

表 21-2 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	0x40005820	I2C2 上升时间寄存器	0x0002

表 21-3 I2C3 相关寄存器列表

名称	访问地址	描述	复位值
R16_I2C3_CTLR1	0x40005C00	I2C3 控制寄存器 1	0x0000
R16_I2C3_CTLR2	0x40005C04	I2C3 控制寄存器 2	0x0000
R16_I2C3_OADDR1	0x40005C08	I2C3 地址寄存器 1	0x0000
R16_I2C3_OADDR2	0x40005C0C	I2C3 地址寄存器 2	0x0000
R16_I2C3_DATAR	0x40005C10	I2C3 数据寄存器	0x0000
R16_I2C3_STAR1	0x40005C14	I2C3 状态寄存器 1	0x0000
R16_I2C3_STAR2	0x40005C18	I2C3 状态寄存器 2	0x0000
R16_I2C3_CKCFGR	0x40005C1C	I2C3 时钟寄存器	0x0000
R16_I2C3_RTR	0x40005C20	I2C3 上升时间寄存器	0x0002

表 21-4 I2C4 相关寄存器列表

名称	访问地址	描述	复位值
R16_I2C4_CTLR1	0x40014000	I2C4 控制寄存器 1	0x0000
R16_I2C4_CTLR2	0x40014004	I2C4 控制寄存器 2	0x0000
R16_I2C4_OADDR1	0x40014008	I2C4 地址寄存器 1	0x0000
R16_I2C4_OADDR2	0x4001400C	I2C4 地址寄存器 2	0x0000
R16_I2C4_DATAR	0x40014010	I2C4 数据寄存器	0x0000
R16_I2C4_STAR1	0x40014014	I2C4 状态寄存器 1	0x0000
R16_I2C4_STAR2	0x40014018	I2C4 状态寄存器 2	0x0000
R16_I2C4_CKCFGR	0x4001401C	I2C4 时钟寄存器	0x0000
R16_I2C4_RTR	0x40014020	I2C4 上升时间寄存器	0x0002

21.12.1 I2C 控制寄存器 (I2Cx_CTLR1) (x=1/2/3/4)

偏移地址: 0x00

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	0

			是 busy 位为 1 时, 重置 I2C 模块。	
14	Reserved	R0	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, 响应地址头应紧跟在 NACK 信号后面。	0
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。	0

			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。	
7	NOSTRETCH	RW	禁止时钟延长位，此位用于在 ADDB 或 BTF 标志被置位的情况下，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。	0
6	ENGCG	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： 1：开启 PEC 计算； 0：关闭 PEC 计算。	0
4	ENARP	RW	ARP 使能位： 1：使能 ARP； 0：关闭 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： 1：SMBus 主设备； 0：SMBus 从设备。	0
2	Reserved	RO	保留。	0
1	SMBUS	RW	SMBus 模式选择位： 1：使用 SMBus 模式； 0：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

21.12.2 I2C 控制寄存器 2 (I2Cx_CTLR2) (x=1/2/3/4)

偏移地址：0x04

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： 1：在 TxNE 或 RxNE 被置位时允许 DMA 请求； 0：在 TxNE 或 RxNE 被置位时禁止 DMA 请求。	0
10	ITBUFEN	RW	缓冲器中断使能位。	0

			1: 当 TxNE 或 RxNE 被置位时, 产生事件中断; 0: 当 TxNE 或 RxNE 被置位时, 不产生中断。	
9	ITEVTEN	RW	事件中断使能位: 1: 使能事件中断; 0: 关闭事件中断。 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 TxNE 或 RxNE 事件; 如果 ITBUFEN=1, TxNE 事件为 1; 如果 ITBUFEN=1, RxNE 事件为 1。	0
8	ITERREN	RW	出错中断使能位: 1: 允许出错中断; 0: 禁止出错中断; 在下列条件下, 将产生该中断; BERR=1; ARL0=1; AF=1; OVR=1; PECERR=1; TIMEOUT=1; SMBAlert=1。	0
[7:6]	Reserved	RO	保留。	0
[5:0]	FREQ[5:0]	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 4~60MHz 之间。必须设置在 000100b 到 111100b 之间, 单位为 MHz。	0

21.12.3 I2C 地址寄存器 1 (I2Cx_OADDR1) (x=1/2/3/4)

偏移地址: 0x08

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
[14:10]	Reserved	RO	保留。	0
[9:8]	ADD[9:8]	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]	ADD[7:1]	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

21.12.4 I2C 地址寄存器 2 (I2Cx_OADDR2) (x=1/2/3/4)

偏移地址: 0x0C

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
[7:1]	ADD2[7:1]	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

21.12.5 I2C 数据寄存器 (I2Cx_DATAR) (x=1/2/3/4)

偏移地址: 0x10

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
[7:0]	DR[7:0]	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

21.12.6 I2C 状态寄存器 1 (I2Cx_STAR1) (x=1/2/3/4)

偏移地址: 0x14

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。 1: 应答错误; 0: 应答正常。	0

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： 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	0

			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	R0	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

21.12.7 I2C 状态寄存器 2 (I2Cx_STAR2) (x=1/2/3/4)

偏移地址: 0x18

位	名称	访问	描述	复位值
[15:8]	PEC[7:0]	R0	包错误检查域, 当 PEC 使能时 (ENPEC 置位), 此域存放 PEC 的值。	0
7	DUALF	R0	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	R0	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	R0	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	R0	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 ENG=1 时, 收到广播呼叫的地址; 0: 未收到广播呼叫地址。	0
3	Reserved	R0	保留。	0
2	TRA	R0	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。 该位根据地址字节的 R/W 位来决定。	0
1	BUSY	R0	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

			0: 总线空闲无通讯。	
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

21.12.8 I2C 时钟寄存器（I2Cx_CKCFGR）（x=1/2/3/4）

偏移地址：0x1C

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	快速模式时的占空比： 1: $T_{\text{低电平}}/T_{\text{高电平}} = 16/9$ ； 0: $T_{\text{低电平}}/T_{\text{高电平}} = 2$ 。	0
[13:12]	Reserved	RO	保留。	0
[11:0]	CCR[11:0]	RW	时钟分频系数域，决定 SCL 时钟的频率波形。	0

21.12.9 I2C 上升时间寄存器（I2Cx_RTR）（x=1/2/3/4）

偏移地址：0x20

位	名称	访问	描述	复位值
[15:6]	Reserved	RO	保留。	0
[5:0]	TRISE[5:0]	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 22 章 I3C 总线（I3C）

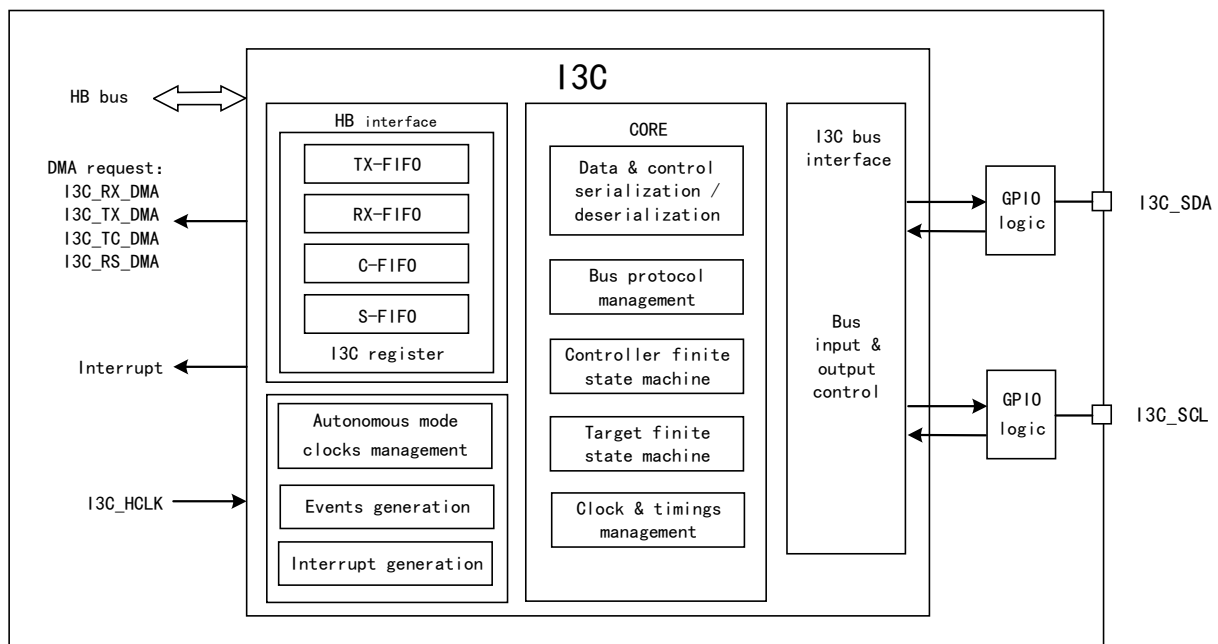
I3C 总线是一种双线制串行单端多分支总线，旨在对传统的 I2C 总线进行改进。I3C 接口负责处理本设备与连接在 I3C 总线上的其他设备之间的通信。它本身支持作为主设备和从设备，当作为主设备时，能够增强 I2C 接口的功能，同时保持一定程度的向后兼容性。

22.1 主要特征

- 支持主设备和从设备
- 支持 MIPI I3C 规范 v1.1
- 支持多主机功能
- 支持 DMA
- 带内中断（IBI）功能
- 内置错误检测和恢复
- I3C SCL 总线时钟最高可达 12.5MHz
- 支持动态分配地址，直接和广播通用命令代码（CCC）和私有读写传输

22.2 概述

图 22-1 I3C 的结构框图



22.3 功能描述

22.3.1 I3C 外设状态

I3C 外设既可以用作 I3C 主设备，也可以用作 I3C 从设备。在任何情况下，外设都处于以下状态之一：

- **禁止状态**

操作条件：I3C 外设复位（RCC 模块中的 I3CRST 位置 1 进行 I3C 复位）后，外设处于禁止状态。

禁止状态至空闲状态的转换：当软件将 R32_I3C_CFGR 寄存器 EN 位置 1 时，外设在完成内部配置参数的校验后，由禁止状态切换至空闲状态。

● 空闲状态

在空闲状态下，软件可以对 I3C 外设配置进行部分更新。

空闲状态至工作状态的转换：

- 1) 软件控制触发：
在主设备模式下，软件通过写操作将 R32_I3C_CFGR 寄存器 TSFSET 位置 1 时启动帧传输时触发；
在从设备模式下，软件发出 IBI/CR/HJ 请求时触发；
- 2) 硬件事件触发：
主设备在总线仲裁期间接收到来自从设备的启动请求和最大 T_{CAS} 时间之后触发；
从设备通过 SDA/SCL 线检测到广播/直接 CCC 或私有读/写时触发。

● 工作状态

在工作状态下，外设在线上执行传输。

在工作状态下，在请求的传输完成时，将通过 R32_I3C_EVR 寄存器中的事件通知软件，并通过 R32_I3C_INTENR 寄存器使能相应的中断。外设从工作状态切换到空闲状态后，仍可在总线上进行通信，并且可以进行部分重新配置。

- 1) 作为主设备：在工作状态下，在请求的传输完成后生成的事件/标志可以是帧完成 (FCF)、IBI/主设备角色/热加入请求完成 (IBIF/CRF/HJF) 或传输错误 (ERRF)；
- 2) 作为从设备：在工作状态下，在请求的传输完成后生成的事件/标志可以是动态地址分配完成 (DAUPDF)、IBI 完成 (IBIENDF)、主设备角色获得完成 (CRUPDF)、广播/直接 CCC 完成 (xxUPDF/RSTF/GETF/STAF)、私有读/写完成 (FCF) 或传输错误 (ERRF)。

22.3.2 I3C 主设备状态和编程序列

在本小节中，我们提供了 I3C 外设作为主设备时的完整编程序列，涵盖了状态转换、主要子任务以及条件。

主设备初始化

当主设备处于禁止状态 (R32_I3C_CFGR 寄存器 EN 位为 0) 时，软件必须遵循以下步骤进行初始化：

- 配置 R32_I3C_CFGR 寄存器，置 CRINIT 为 1，标记主设备角色
- 清除 R32_I3C_RESET 寄存器中 HST_SIE_RST 位
- 配置 I3C 总线时序：
 - 1) 配置 R32_I3C_TIMINGR0 寄存器，设置传统 I2C 和 I3C 开漏/推挽阶段的 SCL 时钟高电平时间周期和低电平时间周期
 - 2) 配置 R32_I3C_TIMINGR1 寄存器，设置推挽阶段的 SDA 保持时间 (t_{HD_PP})、总线空闲条件时间 (I3C t_{CAS} 、传统 I2C t_{BUF})、I3C 重复起始位时序 (t_{CASr} 、 t_{OBSr})、I3C 停止时序 (t_{CBP})、SCL 时钟低电平最大停顿时间 (针对 ENTDA CCC ($t_{STALLDA}$)、传统 I2C 的 ACK/NACK、I3C 传输的地址阶段、写数据传输的奇偶校验位、传统 I2C 写传输的 ACK/NACK 数据阶段、I3C 读传输的转换位或传统 I2C 写传输的 ACK/NACK 阶段 (t_{STALL}))，可供用作主设备的外设根据自身需要调整 SCL 时钟低电平停顿时间、 $t_{NEWORLOCK}$ ，以适应主设备角色切换过程 (在 GETACCCR CCC 命令后)
 - 3) 配置 R32_I3C_TIMINGR2 寄存器，以设定 SCL 时钟低电平的停顿时间 (各个阶段均可单独使能或禁用止，从而调整 SCL 时钟低电平的停顿时间 (以满足被寻址的 I3C 从设备或传统 I2C 从设备在 SDA 切换过程中所需))
- 配置 R32_I3C_DEVR0 寄存器 DA[6:0] 位，设置自身动态地址
- 配置 R32_I3C_DEVRx 寄存器，设置任意从设备 x 的相关管理 ($x = 1/2/3/4$)

- 配置 R32_I3C_CFGR 寄存器，设置帧传输或目标请求的传输的执行模式
- 配置 R32_I3C_INTENR 寄存器，设置任意事件的中断生成或轮询模式
- 设置初始化结束后，EN 位置 1，即可使能 I3C 外设

启动主设备发起的帧传输

当主设备处于使能状态（R32_I3C_CFGR 寄存器 EN 位为 1）时，软件可以通过以下任意配置方法启动帧传输：

1. 软件触发：通过执行写操作将 R32_I3C_CFGR 寄存器 TSFSET 位设置为 1；
 - 此操作将促使硬件将 R32_I3C_EVR 寄存器 CFNFF 标志置位，以请求写入首个控制字给 R32_I3C_CTLR 寄存器。
2. 无触发：通过软件直接写入第一个控制字给 R32_I3C_CTLR 寄存器。

无论采用何种启动方式，I3C 外设随后都会切换至工作状态。当控制字并非 I3C 帧的最后一条消息时（即 R32_I3C_CTLR 寄存器 MEND 位未被置位），且未出现任何传输错误时（即 R32_I3C_EVR 寄存器 ERRF 位置 1），硬件将持续请求下一个控制字，并继续执行帧传输：

- 1) 若 C-FIFO 为非 DMA 模式（R32_I3C_CFGR 寄存器 CDMAEN 位为 0），则软件将在 R32_I3C_EVR 寄存器 CFNFF 标志置位或相应的中断被触发（R32_I3C_INTENR 寄存器 CFNFIE 位置 1）后写入下一个控制字；
- 2) 若 C-FIFO 为 DMA 模式（R32_I3C_CFGR 寄存器 CDMAEN 位为 1），则与之相对应的 I3C DMA 请求将由分配的 DMA 通道自动处理，并写入下一个控制字。

启动和接收从设备发起的传输

当主设备处于使能状态（R32_I3C_CFGR 寄存器 EN 位为 1）时，主设备可以发起的传输，从设备亦可通过发出启动请求（将 SDA 驱动为低电平）来启动传输，但前提是主设备已通过 R32_I3C_DEVRO 寄存器允许热加入请求、IBI 请求或主设备角色请求。

在这种情况下，即便主设备软件未计划启动帧传输，硬件也会自动进入工作状态（在最大 t_{CAS} 时间（可定义为 $1\mu s$ 、 $100\mu s$ 、 $2ms$ 或 $50ms$ ，分别对应于总线工作状态 0、1、2 或 3）之前激活 SCL 时钟）以便接收来自从设备的热加入/带内中断/主设备角色请求。

执行主设备发起的帧传输

在总线上，主设备持续执行帧传输，直至最后一条消息传输完成（R32_I3C_EVR 寄存器 FCF 位为 1），或者传输过程中出现错误（R32_I3C_EVR 寄存器 ERRF 位为 1），并触发相应的中断（若已使能）。这一过程依赖于 R32_I3C_CTLR 和 R32_I3C_TDW/TDR 寄存器（由软件显式写入或由分配的 DMA 通道自动推送）以及 R32_I3C_RDW/RDR（由软件或由分配的 DMA 通道显式读取）。完成传输后，I3C 主设备返回空闲状态。

当 S-FIFO 被禁用（R32_I3C_CFGR 寄存器 SMODE 位为 0）时，每条无传输错误完成的消息，硬件都会更新 I3C 状态寄存器（R32_I3C_STATR），表明 I3C 总线上完成了一次交换消息的传输，软件可以选择读取或忽略该寄存器。

在 R32_I3C_STATR 寄存器的 S-FIFO 被禁用（R32_I3C_CFGR 寄存器 SMODE 位为 0）的情况下，除了最后一条消息传输完成（R32_I3C_EVR 寄存器 FCF 位为 1）或产生传输错误（R32_I3C_EVR 寄存器 ERRF 位为 1）并产生相应的中断（若已使能）外，在执行直接 CCC 读或私有读传输时，如果从设备提前终止读传输，也会通过 R32_I3C_EVR 寄存器 RXTGTENDF 位置 1 和产生相应的中断（若已使能）来通知软件。软件随后可读取 R32_I3C_STATR 寄存器，以获取更多有关于已执行传输的信息。

如果使能了 S-FIFO（R32_I3C_CFGR 寄存器 SMODE 位置 1），则无论从设备是否提前结束读传输，每词消息传输结束后都必须读取一次状态寄存器 R32_I3C_STATR，软件可读取（由 R32_I3C_EVR 寄存器 SFNEF 位为 1 和相应的中断（若已使能）通知），或通过 DMA 读取（如果 R32_I3C_CFGR 寄存器

SDMAEN 位置 1)。只有在读取最后一条消息的状态(S-FIFO 为空)后才会报告帧传输完成(R32_I3C_EVR 寄存器 FCF 位置 1)。

更新主设备传输配置

软件在返回到空闲状态后，可在下一次传输前更新 I3C 外设的配置：

- 配置 R32_I3C_TIMINGR2 寄存器，修改 SCL 时钟停顿；
- 配置 R32_I3C_INTENR 寄存器，修改中断/轮询模式策略；
- 配置 R32_I3C_CFGR 寄存器 TXTHRES、RXTHRES、TMODE、SMODE、TXDMAEN、CDMAEN、RXDMAEN、SDMAEN、EXITPTRN、RSTPTRN、NOARBH 位；

通过软件和/或 DMA 修改/准备要执行的下一次帧传输的控制字、状态字、读/写数据。

通常，在广播/直接 DISEC/ENEC CCC 发出并完成后：

- 通过 R32_I3C_CFGR 寄存器 HJACK 位修改热加入确认策略；
- 通过 R32_I3C_DEVRx 寄存器，修改对任意从设备 x 的 IBI/CR 确认策略。

22.3.3 I3C 从设备状态和编程序列

在本小节中，我们提供了 I3C 外设用作从设备时的完整编程序列，涵盖了状态转换、主要子任务和条件。

从设备初始化

当从设备处于禁止状态(R32_I3C_CFGR 寄存器 EN 位为 0)时，软件必须按如下所述进行初始化：

- 配置 R32_I3C_CFGR 寄存器，置 CRINIT 为 0，将 I3C 外设设置为从设备
- 清除 R32_I3C_RESET 寄存器中 DEV_SIE_RST 位
- 通过 I3C 时序寄存器 1 (R32_I3C_TIMINGR1) 设置 I3C 总线时序：写入 AVAL[7:0]以设置：
 - 总线可用条件时间 (t_{AVAL})，针对 IBI 或主设备角色请求
 - 总线空闲条件时间 (t_{IDLE})，针对热加入请求
 - $t_{NEWORLock}$ ，针对主设备角色切换过程 (GETACCCRCCC 后)
- 配置从设备发起的请求：写入 I3C 自身器件特性寄存器 (R32_I3C_DEVR0) 中的以下位：
 - IBIEN：带内中断 (也称为 IBI) 请求使能/禁止
 - CREN：主设备角色请求使能/禁止
 - HJEN：热加入请求使能/禁止
- 初始化从设备特性和功能
- 配置传输的执行模式：

R32_I3C_CFGR 寄存器中 TXDMAEN、RXDMAEN 分别为 TX-FIFO 和 RX-FIFO 使能/禁止 DMA 模式；TXTHRES、RXTHRES 分别为 TX-FIFO 和 RX-FIFO 的字节/字阈值；SMODE 位置 0 禁止 S-FIFO
- 配置 R32_I3C_INTENR 寄存器，修改任意事件的中断生成或轮询模式
- 设置初始化结束后，EN 位置 1，即可使能 I3C 外设

从主设备接收 (广播 CCC、直接读/写 CCC 或私有读/写) 消息

当从设备处于空闲状态 (R32_I3C_CFGR 寄存器 EN 位为 1) 时，即表示已准备好在 I3C 总线上接收来自主设备的通信消息，并且已准备好切换到工作状态。

在工作状态下，从设备会先接收广播 ENTDAACCC (可能在接收到可选的广播 ENEC/DISEC CCC 后)，然后再被分配动态地址。R32_I3C_EVR 寄存器中的事件 DAUPF 置 1，并生成相关中断 (若已使能)，从设备随后返回空闲状态。

之后，处于空闲状态的从设备便准备好从主设备接收任何其他广播 CCC 消息、直接读/写 CCC 或私有读/写消息。

读取消息状态寄存器

对于每条无错误完成接收的消息，硬件都会通过更新 I3C 状态寄存器（R32_I3C_STATR）来报告 I3C 总线上完成了一次交换消息的传输，软件则通过 I3C 事件寄存器（R32_I3C_EVR）中的相应标志或 I3C 中断使能寄存器（R32_I3C_INTENR）中的相应中断（若已使能）来获得通知后读取该寄存器。

若出现以下消息，软件必须读取 I3C 状态寄存器（R32_I3C_STATR）：

- 私有读：获取交换的数据字节数，因为主设备可能会比从设备预期的更早结束传输（若 R32_I3C_STATR 寄存器 XDCNT[15:0] 小于 R32_I3C_TGTTDR 寄存器 TGTTDCNT[15:0]）。
- 此时，软件必须将 R32_I3C_CFGR 寄存器中的 TXFLUSH 置 1，清空 TX-FIFO。
- DEFTGTS CCC 或 DEFGRPA CCC：获取 RX-FIFO 中接收到的数据字节数。

启动从设备发起的传输

当从设备从禁止状态进入空闲状态（软件在 R32_I3C_CFGR 寄存器 EN 位写 1），并且同时能接收主设备广播 CCC 时，软件可以发出热加入请求（软件在 R32_I3C_CTLR 寄存器 MTYPE[3:0] 位写 1000b）以申请资格加入下一个 ENTDAACCC，前提是热加入请求已使能（R32_I3C_DEVRO 寄存器 HJEN 位置 1）。

分配动态地址（R32_I3C_EVR 寄存器 DAUPF 位置 1）后（与主设备发出的帧传输同时进行的情况更为常见），软件可通过将相关控制字写入 R32_I3C_CTLR 寄存器来向主设备发出 IBI（带内中断请求）或主设备角色请求。

更新 I3C 外设的配置（作为从设备）

返回到空闲状态后，软件可以在下一次传输前更新 I3C 从设备的配置：

- 配置 R32_I3C_INTENR 寄存器，修改中断/轮询模式策略
- 配置 R32_I3C_CFGR 寄存器 TXTHRES、RXTHRES、TXDMAEN、RXDMAEN 位
- 修改/准备 I3CIBI 有效负载数据寄存器（R32_I3C_IBIDR），如果有任何有效负载（R32_I3C_BCR 寄存器 BCR2 位为 1 时），应在启动 IBI 传输（向 I3C 消息控制寄存器[复用]（R32_I3C_CTLR）写入 MTYPE[3:0] 位为 1010b）前执行该操作
- 修改/准备 I3C 从设备发送配置寄存器（R32_I3C_TGTTDR），在从主设备接收私有读或直接 CCC 读（GETSTATUS CCC 除外）消息之前，禁止或使能 TX-FIFO 预装载指定数量的待发送数据字节。

22.3.4 I3C 总线传输和编程

22.3.4.1 I3C 命令集（CCC）（作为主设备/从设备）

表 22-1 提供了支持的 I3C 命令集列表（例如 CCC 列表、通用命令代码），以及外设作为主设备或从设备时如何处理该命令集的概述。

表 22-1 支持的 I3C CCC 列表（作为主设备/从设备）

CCC 名称	CCC 值	读 / 写	有/无释义字节 有/无子命令字节	有/无可选数据字节	用作主设备	用作从设备，生成 I3C_EVR 事件	作为从设备时：特定操作
ENEC	0x00	写	无释义字节/子命令字节	有一个数据字节（使能从设备事件字节）	X	X, INTUPDF	更新并使能 R32_I3C_DEVRO 寄存器：HJEN、CREN、IBIEN（如果有）
DISEC	0x01			有一个数据字节（禁止从设备事件字节）	X	X, INTUPDF	更新并禁止 R32_I3C_DEVRO 寄存器：HJEN、

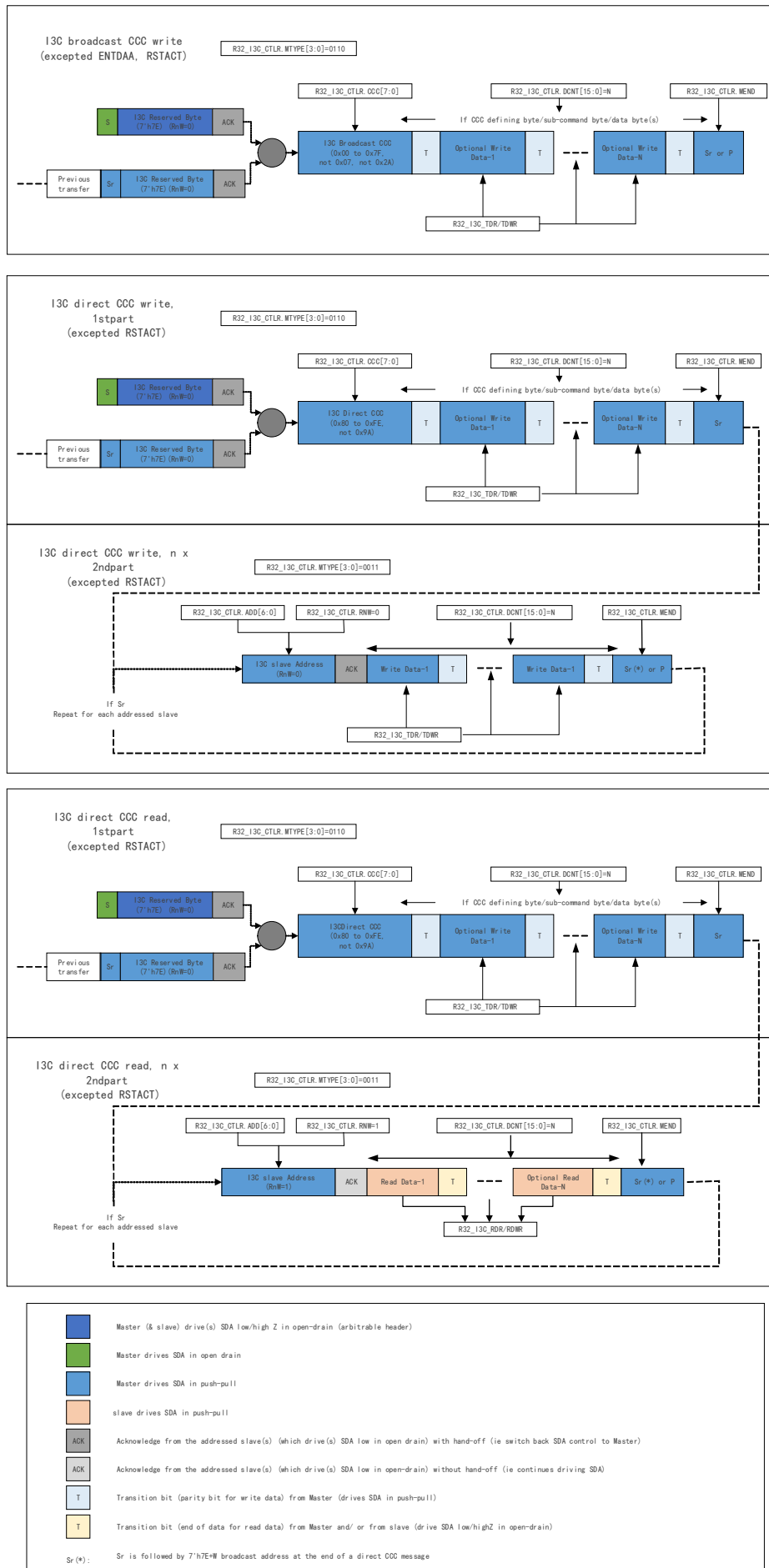
						CREN、IBIEN (如果有)
ENTAS _x ($x=0\cdots3$)	0x02 ... 0x05		无数据字节	X	X, ASUPDF	更新 R32_I3C_DEVR0 寄存器 AS[1:0] 位
RSTDAA	0x06		—	X	X, DAUPDF	将 R32_I3C_DEVR0 寄存器 DAVAL 位 清零
ENTDAA	0x07		—	X	X, DAUPDF	更新 R32_I3C_DEVR0 寄存器 DA[6:0] 位并将 DAVAL 位 置 1
DEFTGTS	0x08		有 $[1+4x(1+n)]$ 个数据字节 (n —从设备数量)	X	X, DEFF	更新 R32_I3C_RDR/R 32_I3C_RDWR 寄 存器
SETMWL	0x09		有两个数据字节	X	X, MWLUPDF	更新 R32_I3C_MAXWLR 寄存器
SETMRL	0x0A		有两个或三个数据 字节	X	X, MRLUPDF	更新 R32_I3C_MAXRLR 寄存器
ENTTM	0x0B		有一个数据字节	X	—	—
RSTACT	0x2A	有释义字节 (0x00、0x01或0x02)	无数据字节	X	X, 检测到 复位时序 后置位 RSTF	更新 R32_I3C_DEVR0 寄存器 RSTACT[1:0]位 并将 RSTVAL 位 置 1
DEFGRPA	0x2B	无释义字节/子 命令字节	有多个数据字节	X	X, GRPF	更新 R32_I3C_RDR/RD WR 寄存器
RSTGRPA	0x2C		无数据字节	X	—	—
直接 CCC						ACK 时执行操作 (如果 I3C 从设 备地址与 R32_I3C_DEVR0 寄存器 DA[6:0] 匹配且 R32_I3C_DEVR0 寄存器 DAVAL=1) (否 则为 NACK)

ENEC	0x80	写	无释义字节/子命令字节	有一个数据字节 (使能从设备事件字节)	X	X, INTUPDF	更新并使能 R32_I3C_DEVR 0寄存器: HJEN、CREN、 IBIEN (如果有)
DISEC	0x81			有一个数据字节 (禁止从设备事件字节)	X	X, INTUPDF	更新并禁止 R32_I3C_DEVR 0寄存器: HJEN、CREN、 IBIEN (如果有)
ENTASx (x=0...3)	0x82 ... 0x85			无数据字节	X	X, ASUPDF	更新 R32_I3C_DEVR0 寄存器 AS[1:0] 位
SETDASA	0x87			无数据字节	X	-	-
SETNEWDA	0x88			有一个数据字节	X	X, DAUPDF	更新 R32_I3C_DEVR0 寄存器DA[6:0] 位, 并将DAVAL 置1
SETMWL	0x89			有两个数据字节	X	X, MWLUPDF	更新 R32_I3C_MAXWLR 寄存器
SETMRL	0x8A			有两个或三个数据 字节	X	X, MRLUPDF	更新 R32_I3C_MAXRLR 寄存器
GETMWL	0x8B			有两个数据字节	X	X, GETF	从 R32_I3C_MAXWL 寄存器 RMWL[15:0]返回 数据字节
GETMRL	0x8C	读		有两个或三个数据 字节	X	X, GETF	从 R32_I3C_MAXRLR寄存器 MRL[15:0]返回 数据字节, 如果R32_I3C_BCR 寄存器BCR2=1, 则从 R32_I3C_MAXRLR寄存器 IBIP[2:0]返回 第三个字节
GETPID	0x8D			有六个数据字节	X	X, GETF	从 R32_I3C_EPIDR 寄存器返回数据 字节
GETBCR	0x8E			有一个数据字节	X	X, GETF	从R32_I3C_BCR 寄存器

						BCR[7:0]返回数据字节
GETDCR	0x8F			X	X, GETF	返回 R32_I3C_DCR 寄存器 BCR[7:0]
GETSTATUS	0x90		有/无释义字节 (TGTSTAT、PRECR)	有两个数据字节 (格式1或格式2 (有PRECR时))	X X, STAF (格式1时) X, GETF (格式2时)	返回两个数据字节
GETACCCR	0x91		无释义字节/子命令字节	有一个数据字节	X X, CRUPDF	从 R32_I3C_DEVR0 寄存器 DA[6:0] 返回带奇偶校验位的数据字节
GETMXDS	0x94		有/无释义字节 (WRRDTURN、CRHDLY)	有两个数据字节 (格式1)、五个数据字节 (格式2或格式3 (有WRRDTURN时)) 或一个数据字节 (格式3 (有CRHDLY时))	X X, GETF	从 R32_I3C_GETMXDSR 寄存器返回数据字节
GETCAPS	0x95	读	有/无释义字节 (TGTSTAT、CRCAPS)	有三个数据字节 (格式1或格式2 (有TGTSTAT时)) 或两个数据字节 (格式2 (有CRCAPS时))	X X, GETF	从 R32_I3C_GETCAPR 寄存器返回3个GETCAPx数据字节或者从 R32_I3C_CRCAPR 寄存器返回2个CRCAPx数据字节
RSTACT	0x9A	读 / 写	有释义字节 (0x00、0x01或0x02)	有释义字节 (0x00、0x01或0x02)	X X, 检测到复位时序后置位 RSTF	读: 从 R32_I3C_DEVR0 寄存器 RSTACT[1:0] 返回数据字节 写: 更新 R32_I3C_DEVR0 寄存器 RSTACT[1:0] 并将RSTVAL位置1
SETGRPA	0x9B	写	无释义字节/子命令字节	无释义字节/子命令字节	X	-
RSTGRPA	0x9C				X	

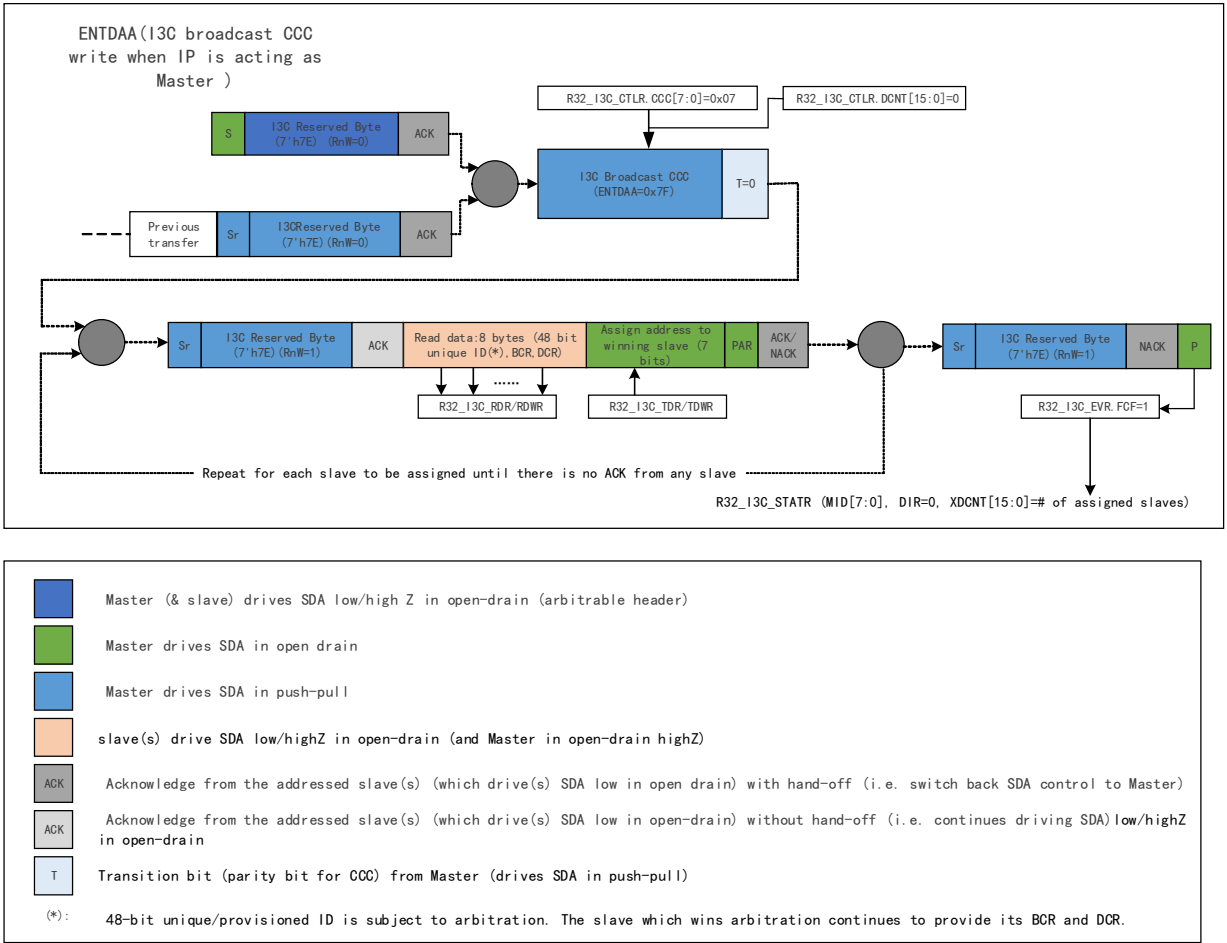
22.3.4.2 I3C 广播/直接 CCC 传输 (ENTDAA 和 RSTACT 除外) (作为主设备)

图 22-2 I3C CCC 消息



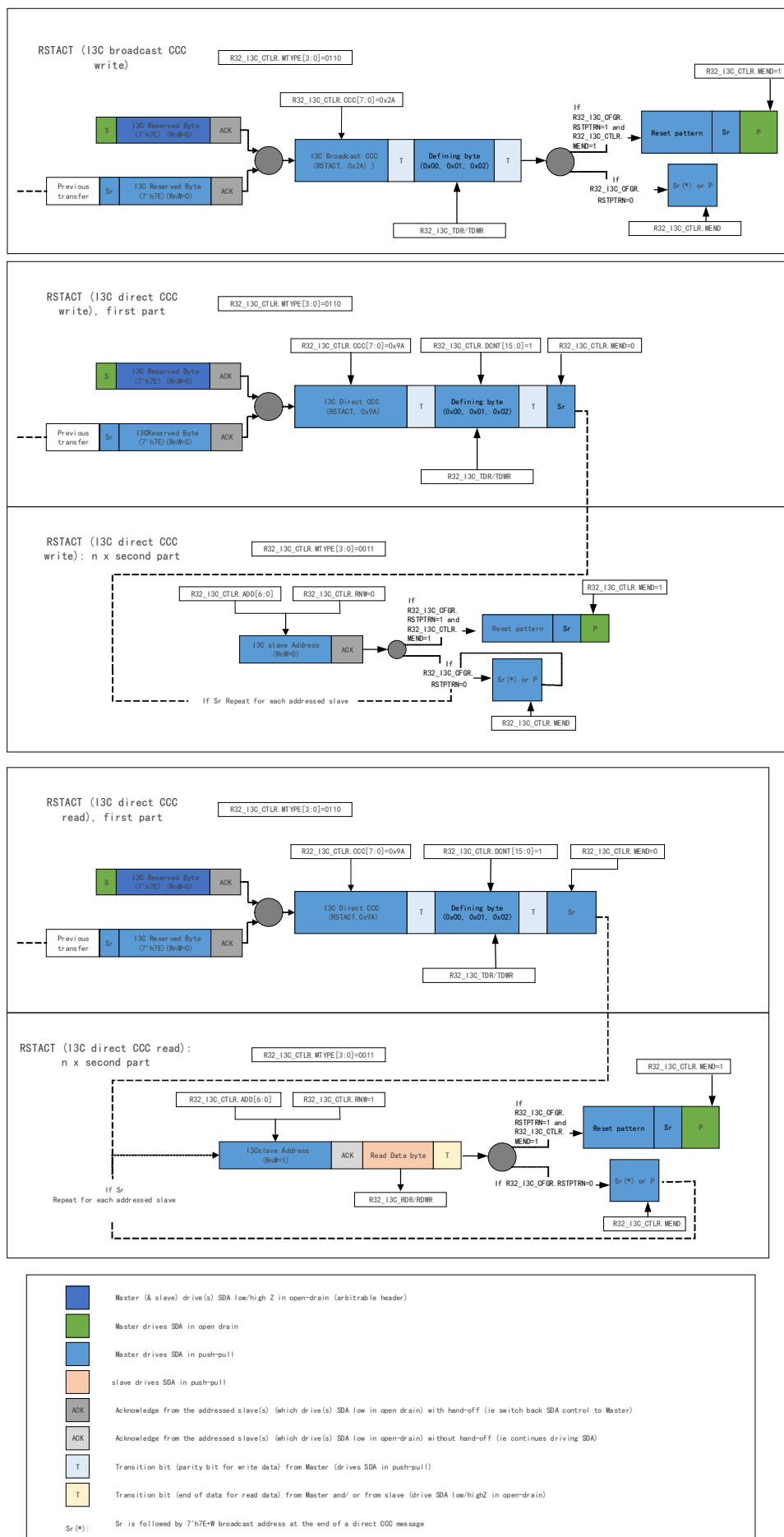
22.3.4.3 I3C 广播 ENTDAACCC 传输（作为主设备）

图 22-3 I3C 广播 ENTDAACCC 传输（作为主设备）



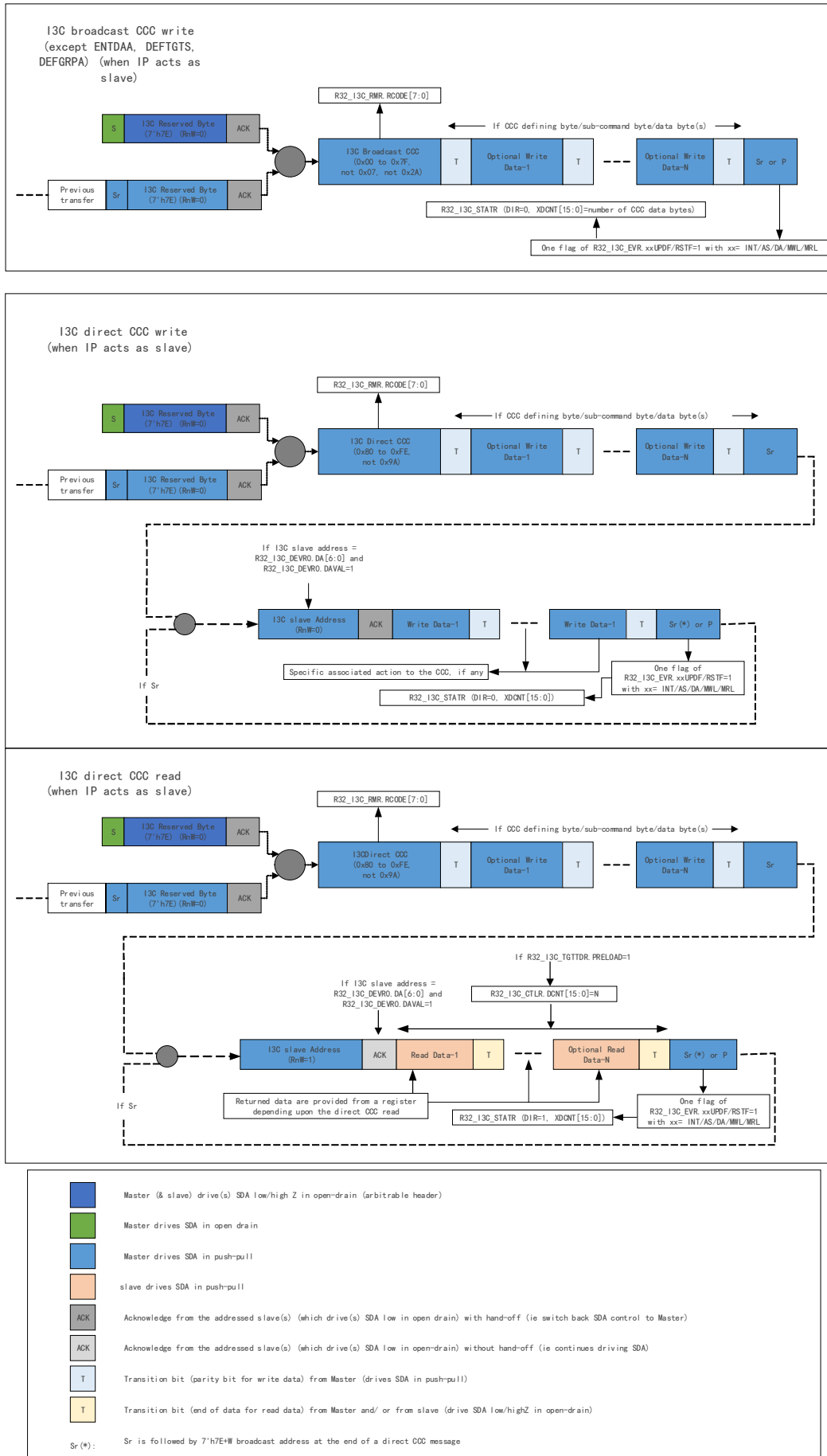
22.3.4.4 I3C 广播/直接 RSTACT CCC 传输（作为主设备）

图 22-4 13C 广播、直接读/写 RSTACT CCC (作为主设备)



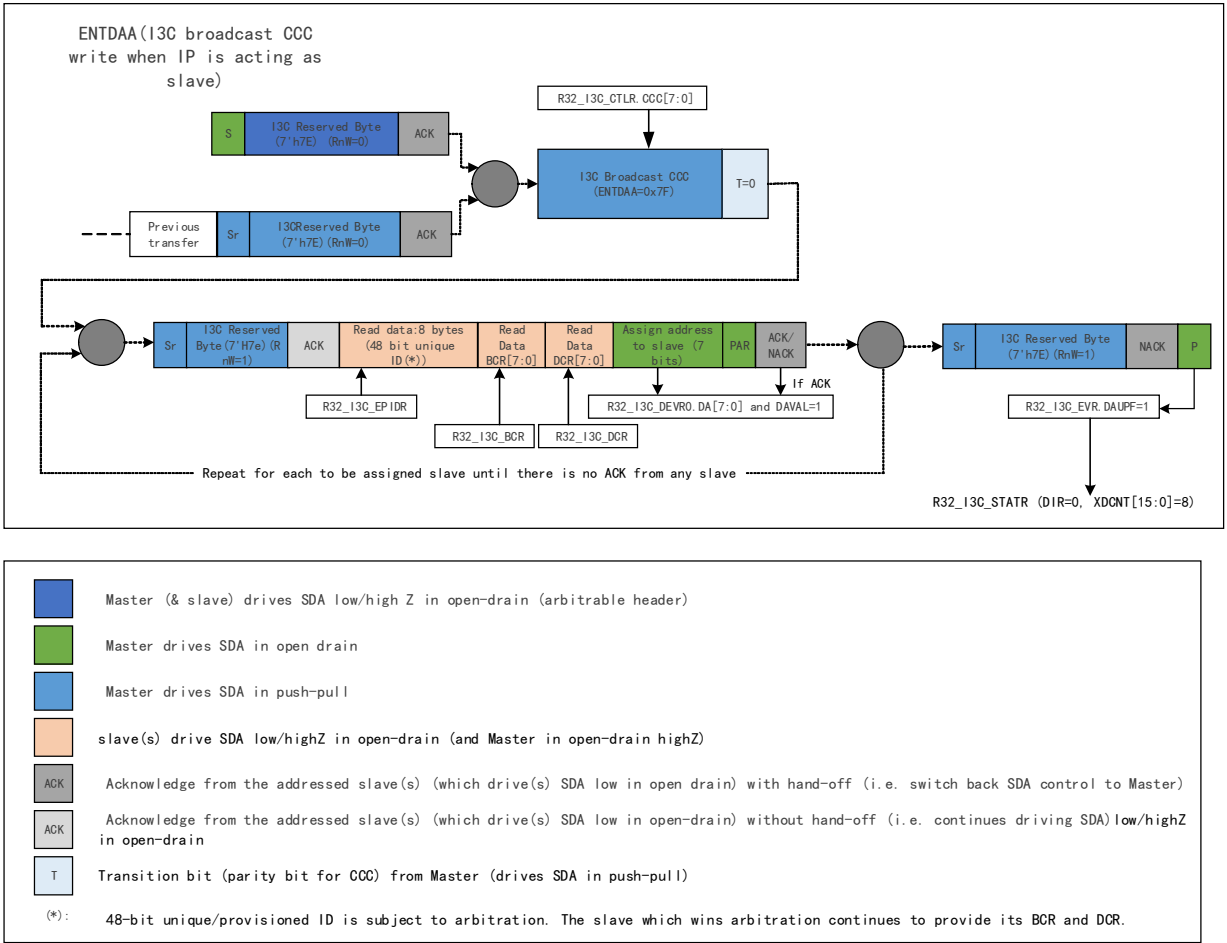
22.3.4.5 I3C 广播/直接 CCC 传输（ENTDAA、DEFTGTS、DEFGRPA 除外）（作为从设备）

图 22-5 I3C CCC 消息（作为从设备）



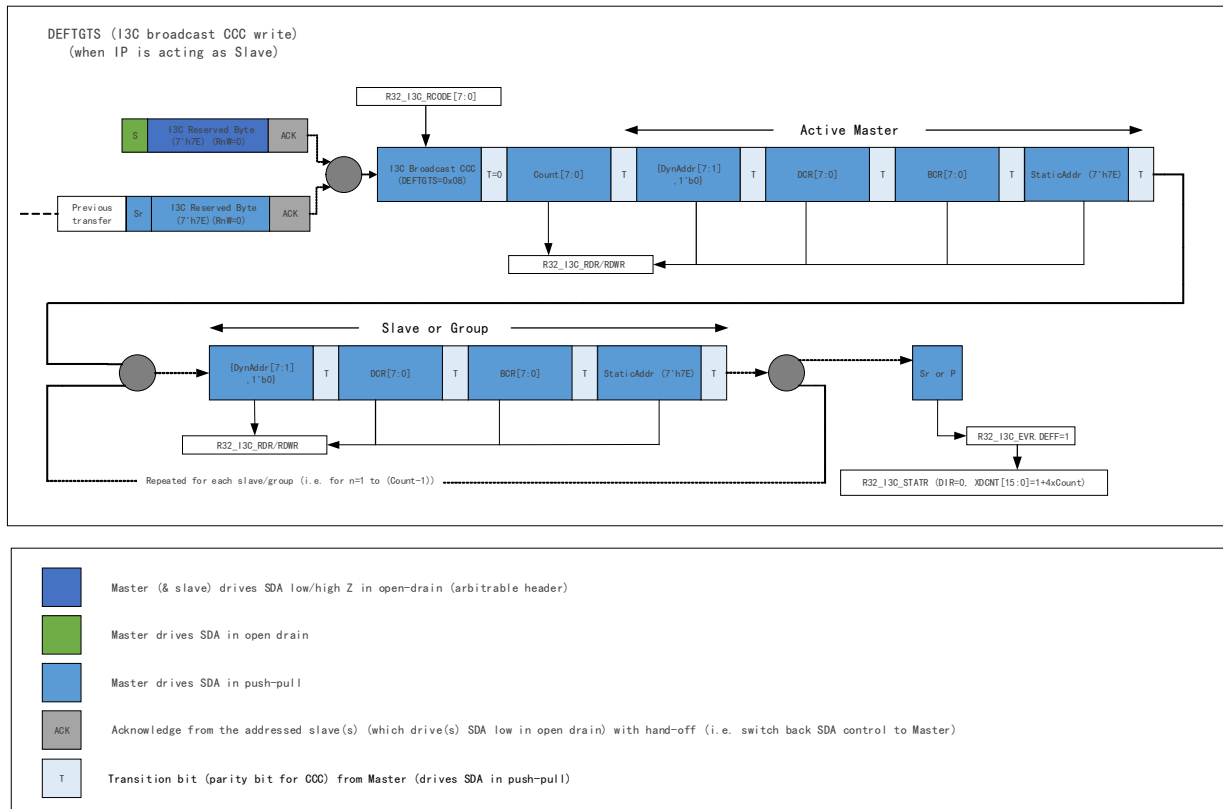
22.3.4.6 I3C 广播 ENTDAACCC 传输（作为从设备）

图 22-6 I3C 广播 ENTDAACCC 传输（作为从设备）



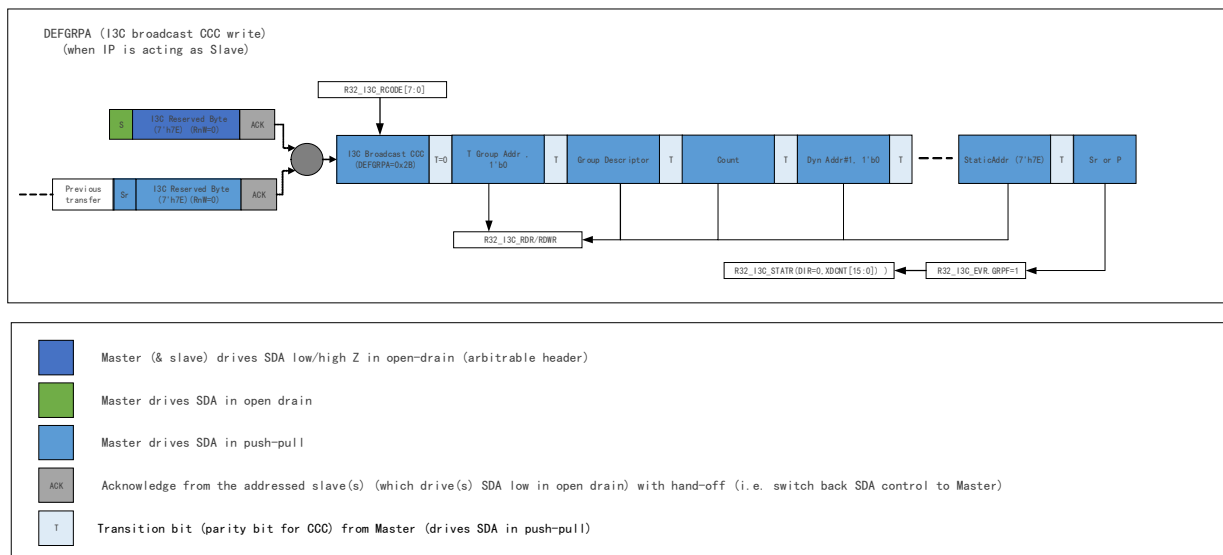
22.3.4.7 I3C 广播 DEFTGTS CCC 传输（作为从设备）

图 22-7 I3C 广播 DEFTGTS CCC 传输（作为从设备）



22.3.4.8 I3C 广播 DEFGRPA CCC 传输（作为从设备）

图 22-8 I3C 广播 DEFGRPA CCC 传输（作为从设备）



22.3.4.9 I3C 直接 GETSTATUS CCC 传输（作为从设备）

当 I3C 作为从设备时,硬件会在接收到 GETSTATUS CCC 时,按照格式 1(无释义字节或释义字节 TGTSTAT = 0x00) 或格式 2(释义字节 PRECR = 0x91) 返回两个数据字节。

I3C 总线上,以格式 1 返回的 2 字节 STATUS[15:0]如下:

STATUS[15:14] = 00 (未使用);

STATUS[13] = 1 (如果自接收到前一个 GETSTATUS CCC 后检测到丢失的起始位) 或 0 (其他情况);

STATUS[12] = 1（如果自接收到前一个 GETSTATUS CCC 后检测到上溢/下溢错误）或 0（其他情况）；

STATUS[11] = 1（如果自接收到前一个 GETSTATUS CCC 后，在 SDR 读传输期间检测到 SCL 的稳定时间超过 125 μ s）或 0（其他情况）；

STATUS[10:8] = 000 到 110（如果自接收到前一个 GETSTATUS CCC 后检测到协议错误（前提是 STATUS[5] = 1），编码值 $x = 0$ 到 6，对应于从设备错误 TEx）或 000（其他情况）；

STATUS[7:6] = 00（准备好执行切换过程）；

STATUS[5] = 1（如果自接收到前一个 GETSTATUS CCC 后检测到协议错误）或 0（其他情况）；

STATUS[4] = 0（保留）；

STATUS[3:1] = 000（未使用）；

STATUS[0] = 1（如果存在挂起的中断（前提是在 R32_I3C_CTLR 寄存器中配置了 IBI，R32_I3C_DEVR0 寄存器中的 IBIEN = 1 且 DAVAL = 1，而且 IBI 尚未被主设备确认也未通过 DISEC 禁止）或 0（其他情况）。

I3C 总线上，以格式 2 返回的 2 字节 STATUS[15:0] 如下：

STATUS[15:8] = 00000000（未使用）；

STATUS[7:2] = 00000（未使用）；

STATUS[1] = 1（如果软件仍在处理接收到的 DEFTGTS 或接收到的 DEFGRPACCC，并且尚未将相关事件清除（R32_I3C_EVR 寄存器中的 DEFF = 1 或 GRPF = 1））；主设备必须等待软件完成上述操作后再发出 GETACCR CCC（否则不会被确认）□；

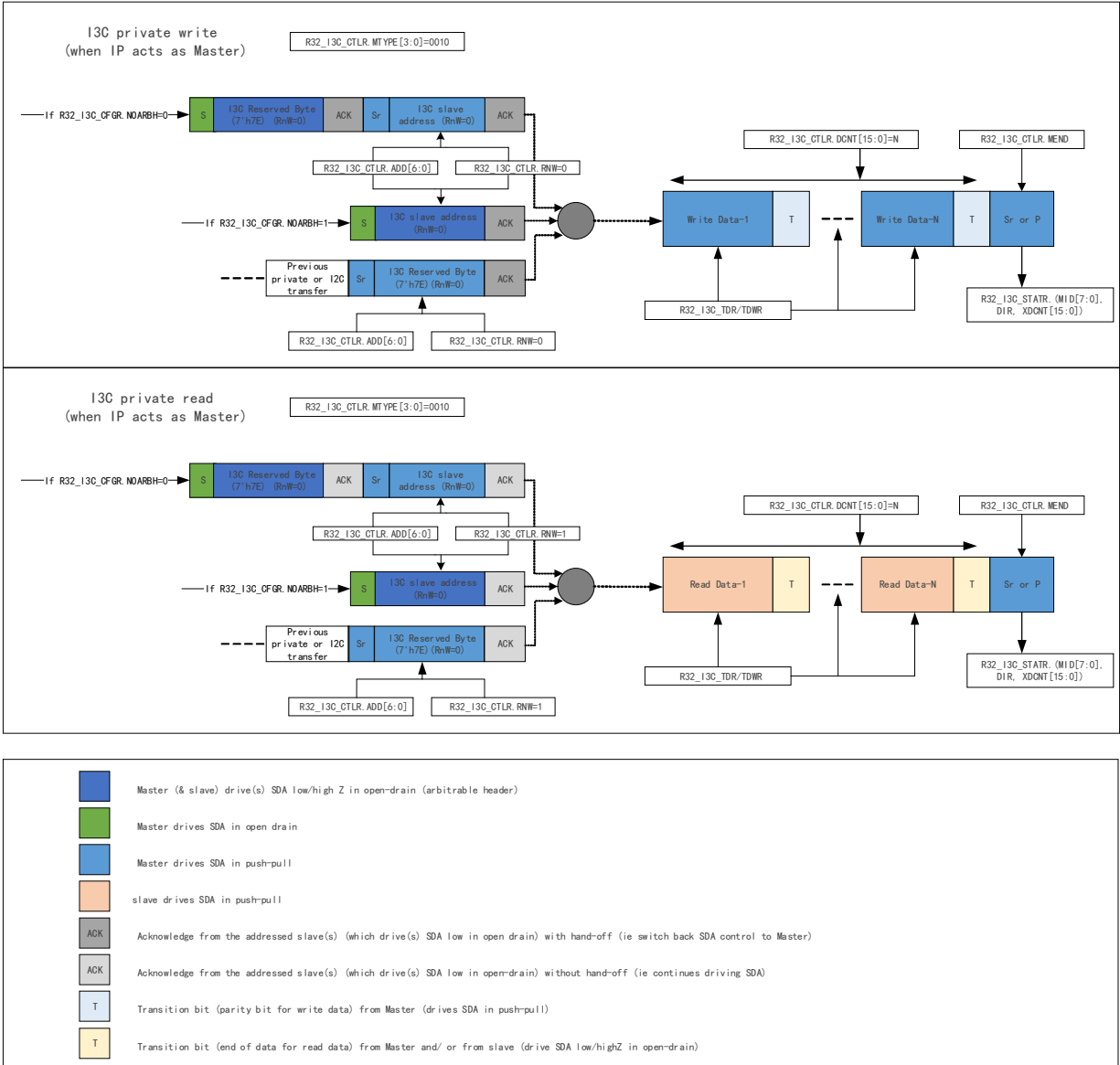
STATUS[0] = 1（如果 DEFTGTS 或 DEFGRPA CCC 可能已丢失）。如果检测到丢失的起始位（R32_I3C_EVR 寄存器中的 WKPF = 1），该位将会置 1；如果 R32_I3C_EVR 寄存器中的 DEFF = 1 或 GRPF = 1，该位将会清零。

当格式 1 的 GETSTATUS CCC 完成后，将通过 R32_I3C_EVR 寄存器中的 STAF = 1 和相应的中断（如果使能，即 R32_I3C_INTENR 寄存器中的 STAIE = 1）进行报告。

当格式 2 的 GETSTATUS CCC 完成后，将通过 R32_I3C_EVR 寄存器中的 GETF=1 和相应的中断（如果使能，即 R32_I3C_INTENR 寄存器中的 GETIE = 1）进行报告。

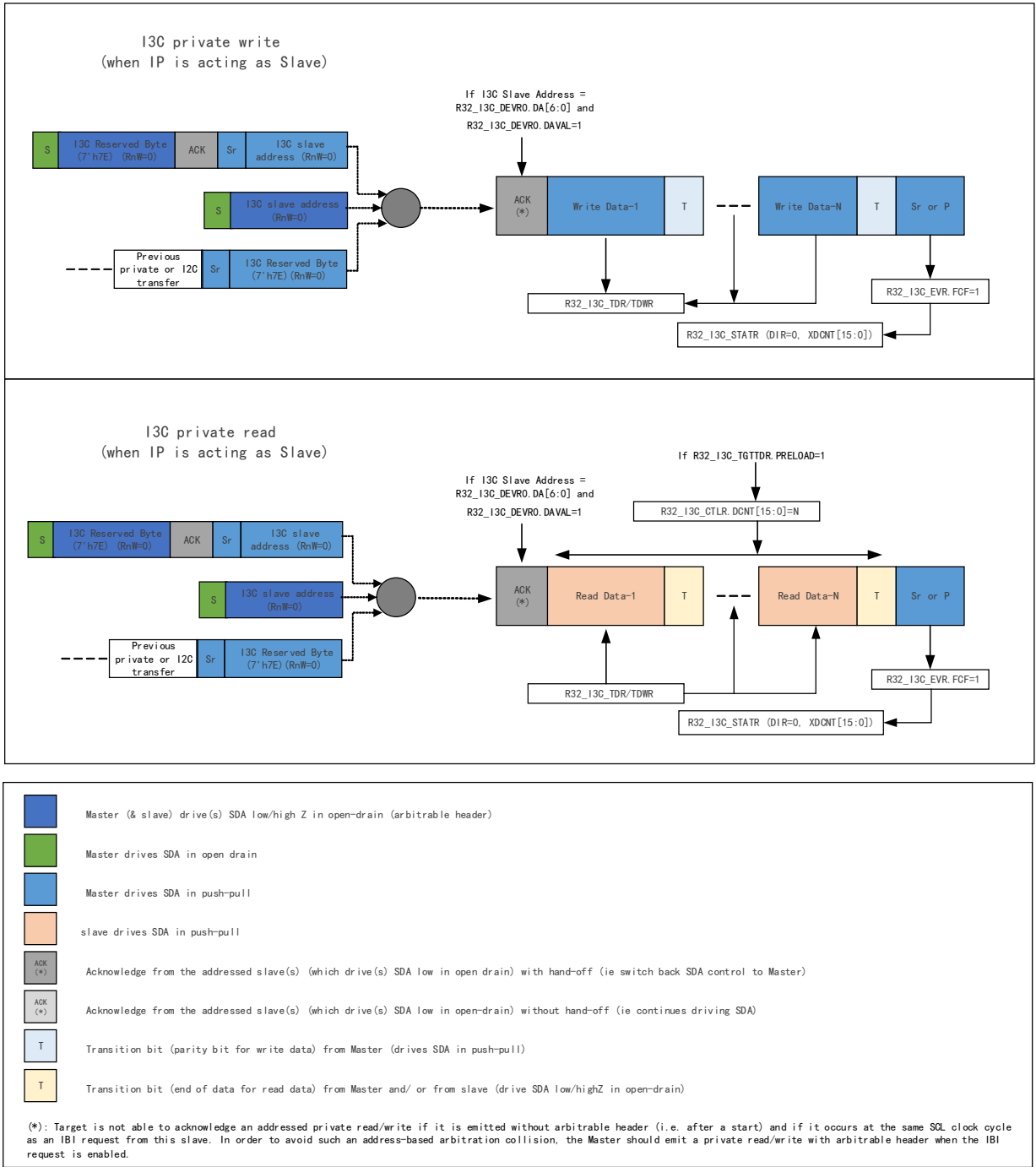
22.3.4.10 I3C 私有读/写传输（作为主设备）

图 22-9 I3C 私有读/写消息（作为主设备）



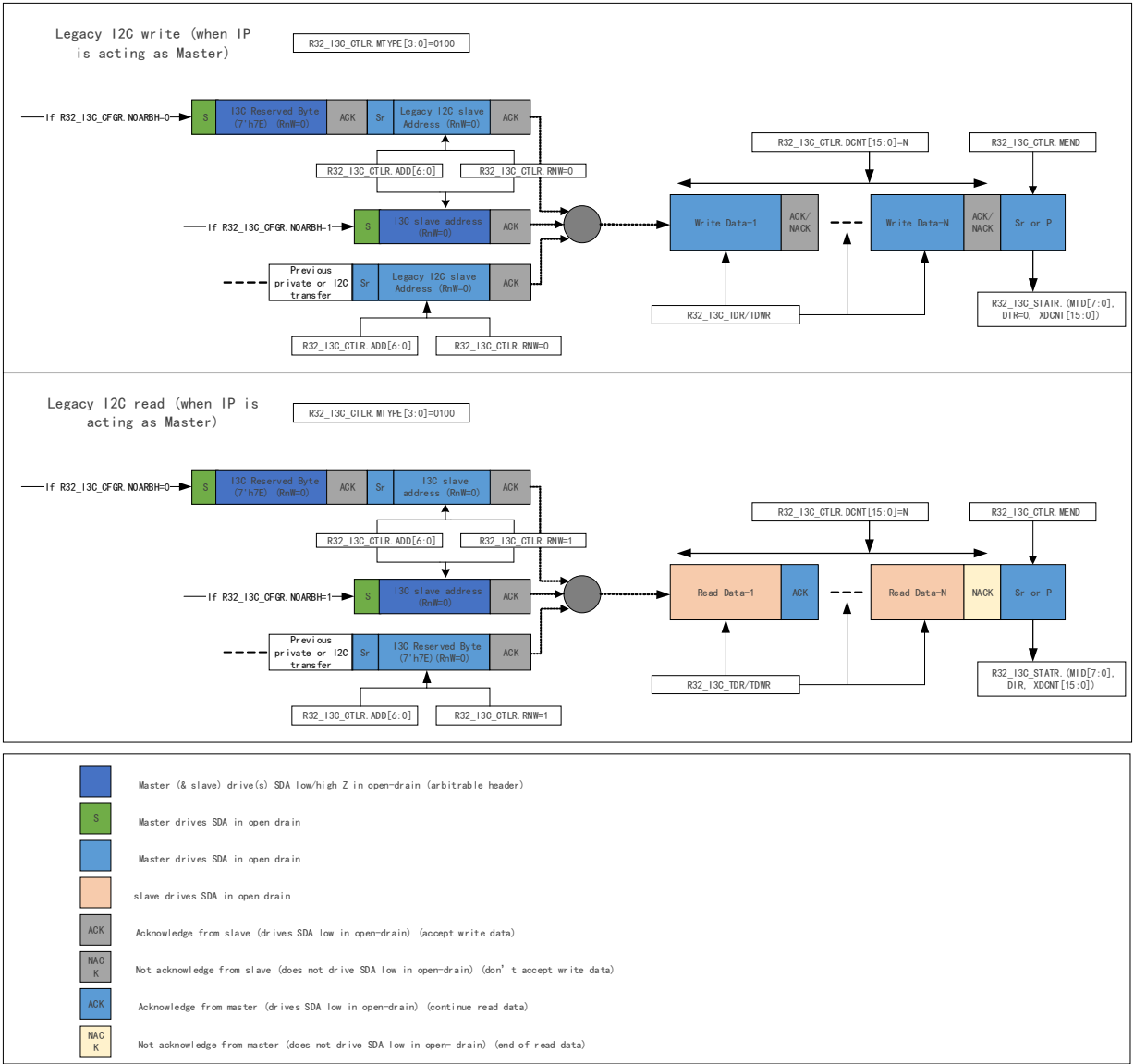
22.3.4.11 I3C 私有读/写传输（作为从设备）

图 22-10 I3C 私有读/写传输（作为从设备）



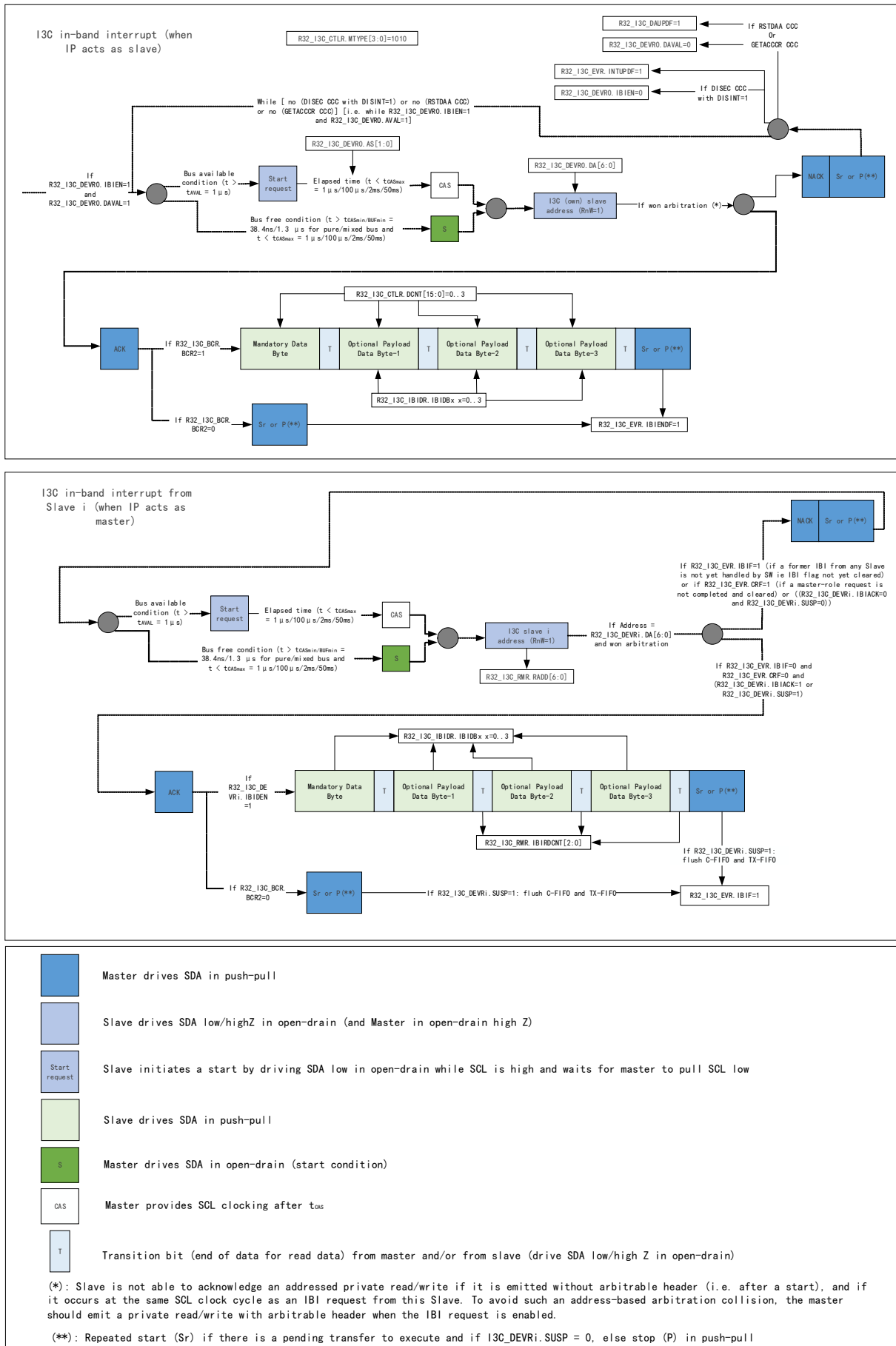
22.3.4.12 传统 I2C 读/写传输（作为主设备）

图 22-11 传统 I2C 读/写传输（作为主设备）



22.3.4.13 I3C IBI 传输（作为主设备/从设备）

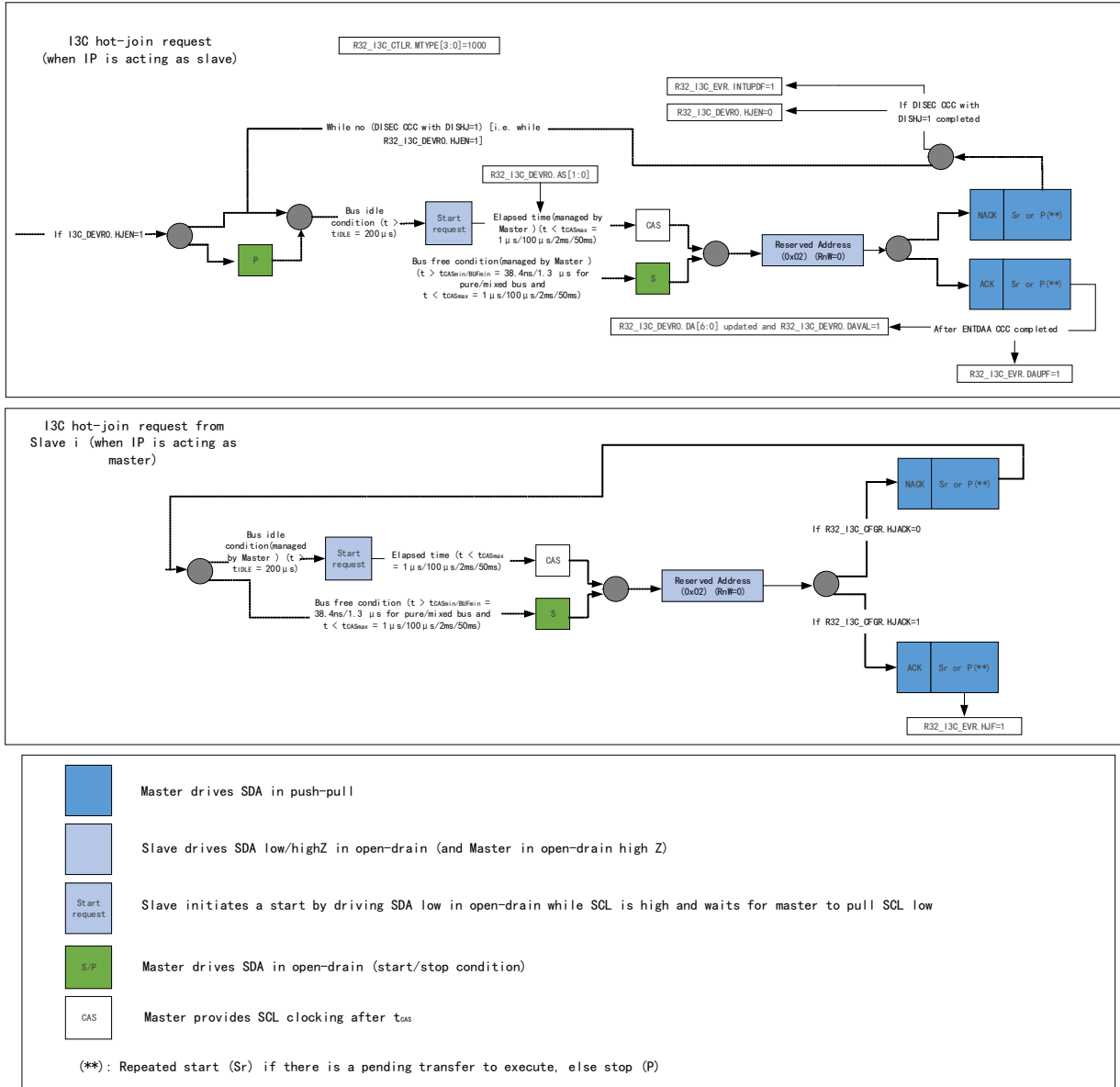
图 22-12 I3C IBI 传输（作为主设备/从设备）



当外设用作主设备时，R32_I3C_IBIDR 寄存器用于接收 IBI 数据有效负载。因此，来自从设备的 IBI 请求不得超过 4 字节的数据有效负载。如果需要在该带内中断的上下文中交换更多信息，则主设备软件必须发出私有读控制字。

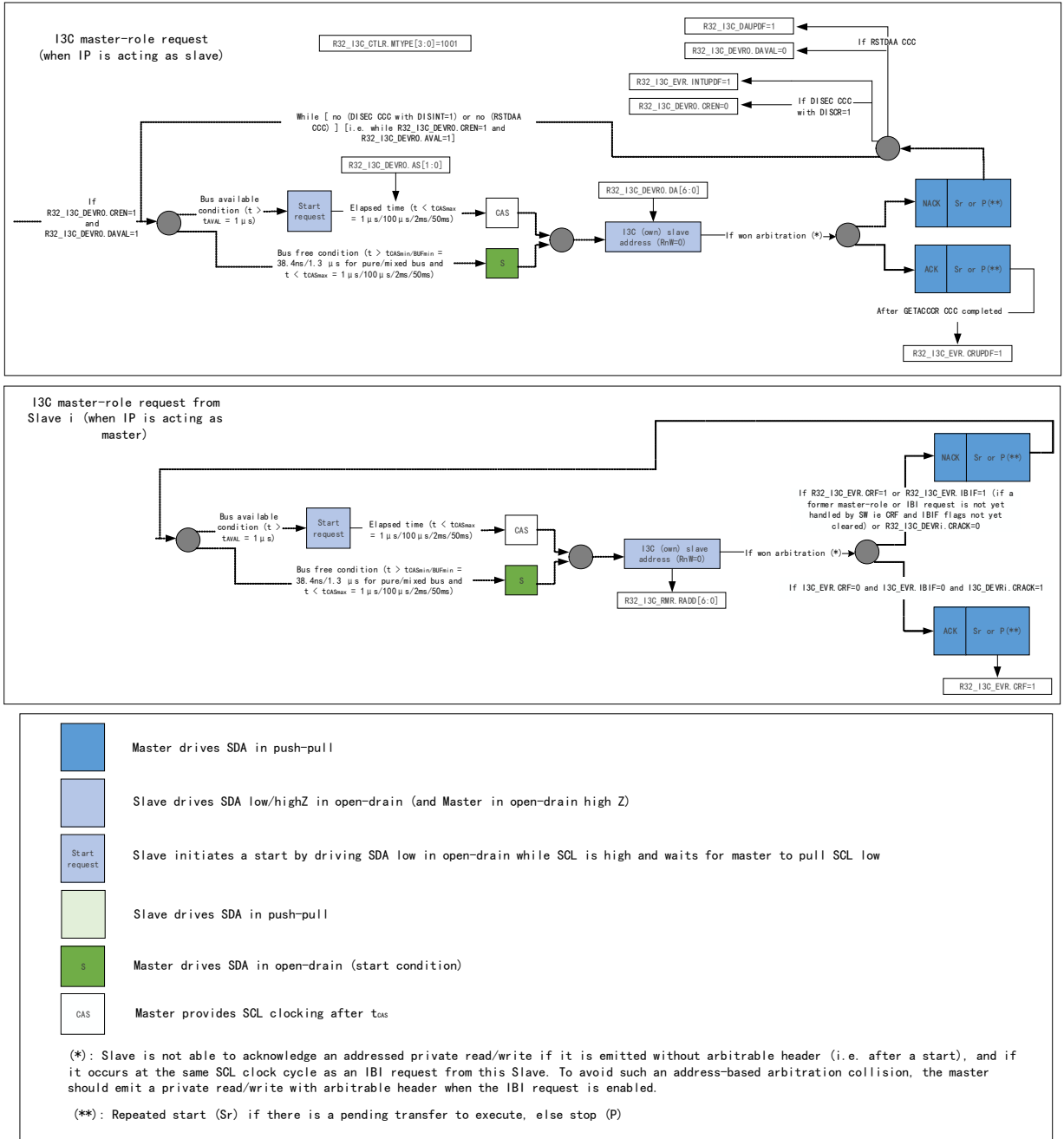
22.3.4.14 I3C 热加入请求传输（作为主设备/从设备）

图 22-13 I3C 热加入请求传输（作为主设备/从设备）



22.3.4.15 I3C 主设备角色请求传输（作为主设备/从设备）

图 22-14 I3C 主设备角色请求传输（作为主设备/从设备）



22.3.5 I3C FIFO 管理

表 22-2 I3C FIFO 实现

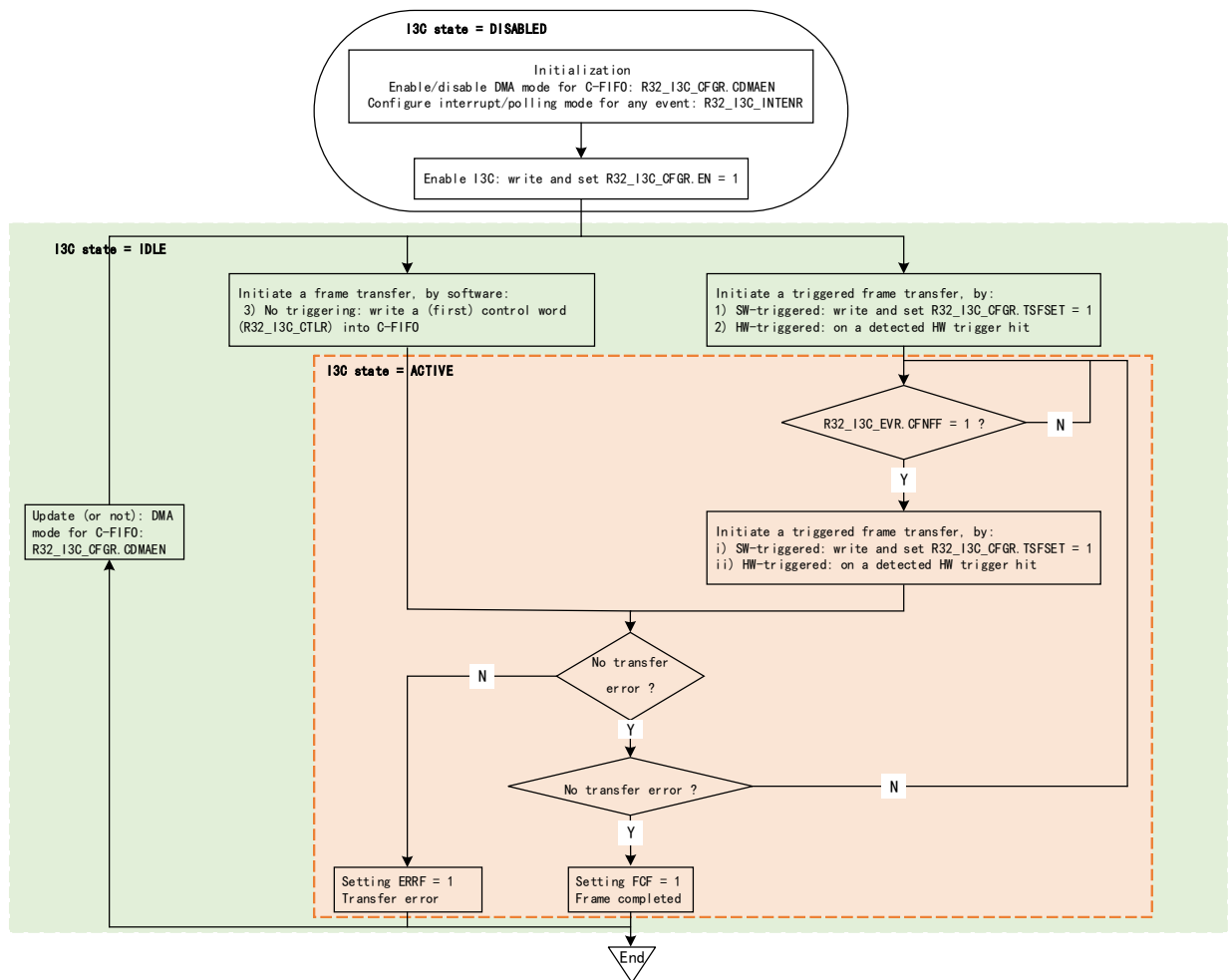
FIFO	内容	单位	大小	用作主设备/从设备（基本原理）
C-FIFO	32 位控制字	字	2	主设备（一个帧可基于多个控制字；而用作从设备时则不然）
S-FIFO	32 位状态字			主设备（从设备：仅限寄存器模式下的状态）
TX-FIFO	发送的数据	字节	8	主设备和从设备
RX-FIFO	接收的数据			

22.3.5.1 C-FIFO 管理（作为主设备）

用作主设备时，可以在 C-FIFO 广播 CCC、直接读/写 CCC、私有读/写、传统 I2C 读/写、软件发起的错误恢复期间使用。

图 22-15 说明了当 I3C 外设用作主设备时，如何管理 C-FIFO 以对 I3C 总线上的控制字进行排队。

图 22-15 C-FIFO 管理（作为主设备）



首先，软件必须通过 R32_I3C_CFGR 寄存器 CDMAEN 来初始化 C-FIFO 管理，具体步骤如下：

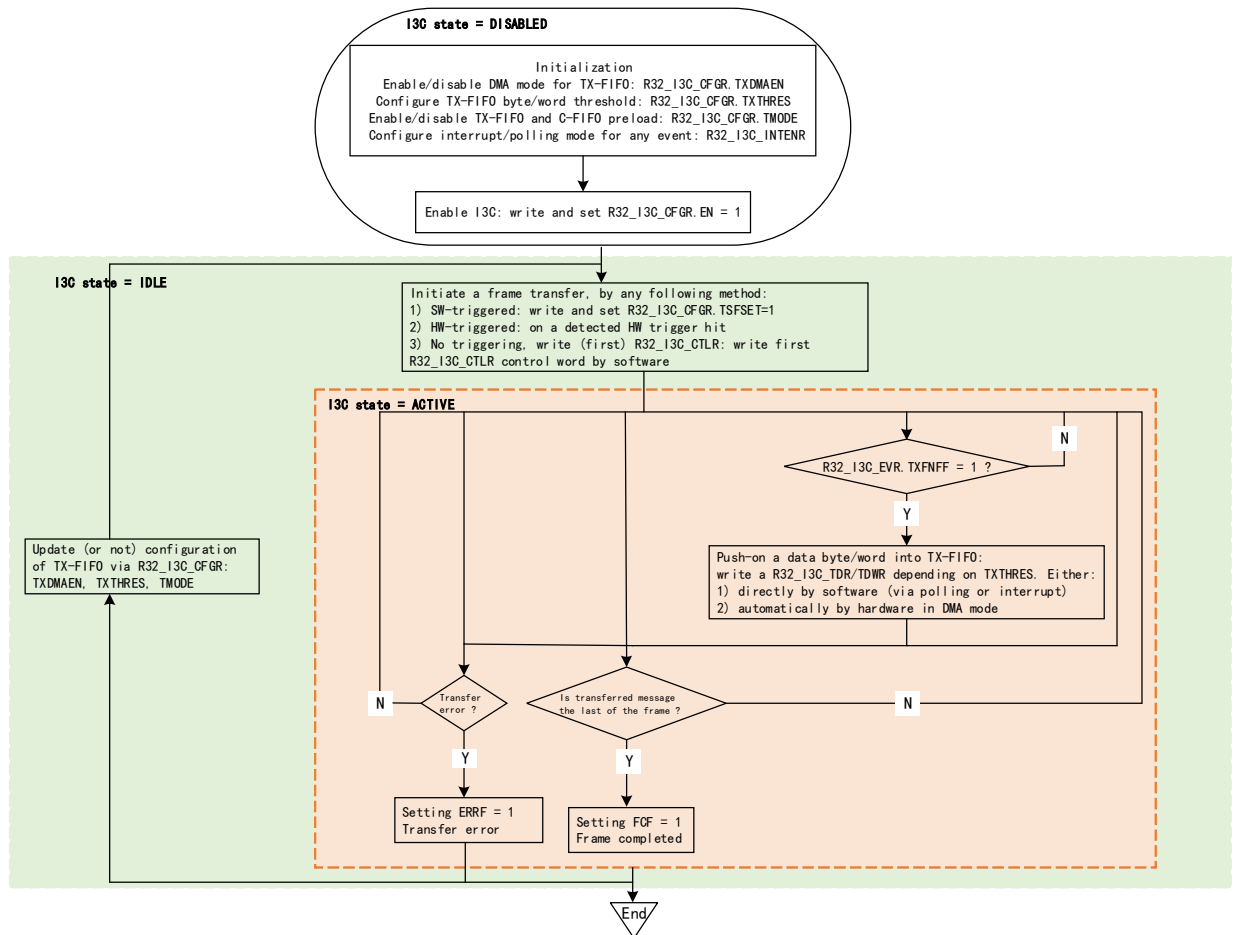
- 软件直接按控制字进行写入（当 CDMAEN=0 时）：
 - 通过轮询模式（设置 R32_I3C_INTENR 寄存器中的 CFNFIE=0）：在向 R32_I3C_CTLR 寄存器显式写入之前，等待硬件请求下一个控制字（R32_I3C_INTENR 寄存器中的 CFNFF=1）
 - 通过使能的中断通知（当 CFNFIE=1 时）：
- 由分配的 DMA 通道写入（当 CDMAEN=1 时），以使能相应的 I3C 外设 DMA 请求：
 - 根据模块级别的配置，DMA 会自动将控制字从其存储器源缓冲区推入/写入 R32_I3C_CTLR 寄存器，直到帧完成（在帧的最后一条消息后，I3C 总线上会发出停止位），但发生传输错误时除外。
- 在任何情况下，只要 C-FIFO 为空并且必须发出新控制字的重复起始位，都会报告 C-FIFO 下溢（R32_I3C_EVR 寄存器中的 ERRF=1 且 R32_I3C_STATERR 寄存器中的 COVR=1）。如果使能了 R32_I3C_INTENR 寄存器 ERRIE 位，则会生成中断。
- 当 I3C 外设未处于工作状态时，可以修改 C-FIFO 管理的 DMA 模式配置。
- 用作主设备时，如果发生传输错误（R32_I3C_EVR 寄存器中的 ERRF=1），硬件将会自动清空 C-FIFO。

22.3.5.2 TX-FIFO 管理（作为主设备）

用作主设备时，可以在广播或直接 CCC（包括 ENTDA 和 RSTACT），如果存在释义字节/子命令字节或数据字节、私有写、传统 I2C 写期间使用 TX-FIFO

图 22-16 说明了当 I3C 外设用作主设备时，如何管理 TX-FIFO 以对要在 I3C 总线上发送的数据字节或字进行排队。

图 22-16 TX-FIFO 管理（作为主设备）



首先，必须通过 R32_I3C_CFGR 寄存器中的以下位域初始化 TX-FIFO 管理：

- TXDMAEN：使能/禁止 TX-FIFO 的 DMA 模式
- TXTHRES：将数据字节或字推入 TX-FIFO
- TMODE：使能/禁止 TX-FIFO 和 C-FIFO 预装载

然后，根据 R32_I3C_CFGR 寄存器 TXDMAEN 位，有两种方法可以向 TX-FIFO 写入数据：

- 软件直接按字节/字写入（当 TXDMAEN=0 时）：
 - 通过轮询模式(R32_I3C_INTENR 寄存器 TXFNFI=0)：在显式写入 R32_I3C_TDR 或 R32_I3C_TDWR 寄存器之前，等待硬件请求下一个数据字节/字（R32_I3C_INTENR 寄存器 TXFNFF = 1），具体取决于 R32_I3C_CFGR 寄存器 TXTHRES 位
 - 通过使能的中断通知（当 TXFNFI = 1 时）
- 由分配的 DMA 通道写入（当 TXDMAEN = 1 时），以使能相应的 I3C 外设 DMA 请求：
 - 根据 DMA 模块级别的配置，DMA 会自动将数据字节/字从其存储器源缓冲区推入/写入 R32_I3C_TDR 或 R32_I3C_TDWR 寄存器（具体取决于 R32_I3C_CFGR 寄存器 TXTHRES 位），直到帧完成（在帧的最后一条消息后，I3C 总线上会发出停止位），但发生传输错误时除外。

I3C 消息以起始位或重复起始位开始，以停止位或重复起始位结束。在消息级别，要发送的最后一个数据字节/字由 R32_I3C_EVR 寄存器 TXLASTF = 1 进行指示。当 I3C 帧包含多条消息（由重复起

始位分隔) 时, 软件可以使用该事件来更新相应的指针, 使之指向用于存储下一条消息的字节/字的缓冲区。

当报告帧完成时 (R32_I3C_EVR 寄存器 FCF = 1, 且相应的中断已使能), TX-FIFO 为空。如果 TX-FIFO 为空并且必须在 I3C 总线上发送数据字节, 则会报告 TX-FIFO 下溢 (R32_I3C_EVR 寄存器 ERRF=1 且 R32_I3C_STATER 寄存器 DOVR = 1)。如果使能了相应的中断 (R32_I3C_INTENR 寄存器 ERRIE = 1), 则会生成中断。当 I3C 外设未处于工作状态时, 可以修改 TX-FIFO 管理的配置。

用作主设备时, 如果发生传输错误 (ERRF = 1), 硬件将会自动清空 TX-FIFO。

无 C-FIFO/TX-FIFO 预装载

C-FIFO 大小为 2 个字, TX-FIFO 大小为 8 个字节。

在未启用 C-FIFO/TX-FIFO 预装载功能的条件下, 即 R32_I3C_CFGR 寄存器 TMODE=0, 一旦 C-FIFO 接收到首个控制字, I3C 外设便会立即在 I3C 总线上发出起始位。随后, 外设将对 R32_I3C_CTLR 寄存器进行解码, 并根据需要写入下一个数据字节或字。一旦检测到硬件要求在 I3C 总线上发送下一个控制字 (例如, 需要在 I3C 总线上发出重复起始位或 C-FIFO 需获取可用空间), 外设将立即请求将该控制字写入 C-FIFO, 直至最后一条消息 (即 R32_I3C_CTLR 寄存器 MEND = 1)。同样地, 一旦检测到硬件要求在 I3C 总线上发送另一个数据字节或字 (例如, 需要在 I3C 总线上发出重复起始位, TX-FIFO 未滿, 或者必须在该 I3C 消息期间发送数据字节或字), 则必须立即将该数据字节/字写入 TX-FIFO。

C-FIFO 和 TX-FIFO 预装载

当配置了 C-FIFO/TX-FIFO 预装载时 (R32_I3C_CFGR 寄存器 TMODE = 1), 在总线上发出起始位之前, I3C 外设需要尽可能地等待 C-FIFO 和 TX-FIFO 装载, 具体如下:

等待第一个控制字写入 C-FIFO □ 等待数据字节/字写入 TX-FIFO (如果有), 如第一个控制字所定义 (如果 R32_I3C_CTLR 寄存器中的 RNW=0 且 DCNT[15:0]=0), 最多可写入 TX-FIFO 大小 □ 如果 TX-FIFO 未滿并且第一个控制字不是帧的最后一条消息 (R32_I3C_CTLR 寄存器中的 MEND=0):

- 等待第二个控制字写入 C-FIFO, 随即 C-FIFO 装满。

- 如果 TX-FIFO 未滿, 则等待数据字节/字写入 TX-FIFO (如果有), 如第二个控制字所定义 (R32_I3C_CTLR 寄存器中的 RNW=0 且 DCNT[15:0]), 最多可写入 TX-FIFO 大小。

然后, 只要检测到硬件要求在 I3C 总线上发送下一个控制字 (如果必须在 I3C 总线上发出重复起始位), 就会立即请求将该控制字写入 C-FIFO, 直到最后一条消息 (R32_I3C_CTLR 寄存器中的 MEND=1)。

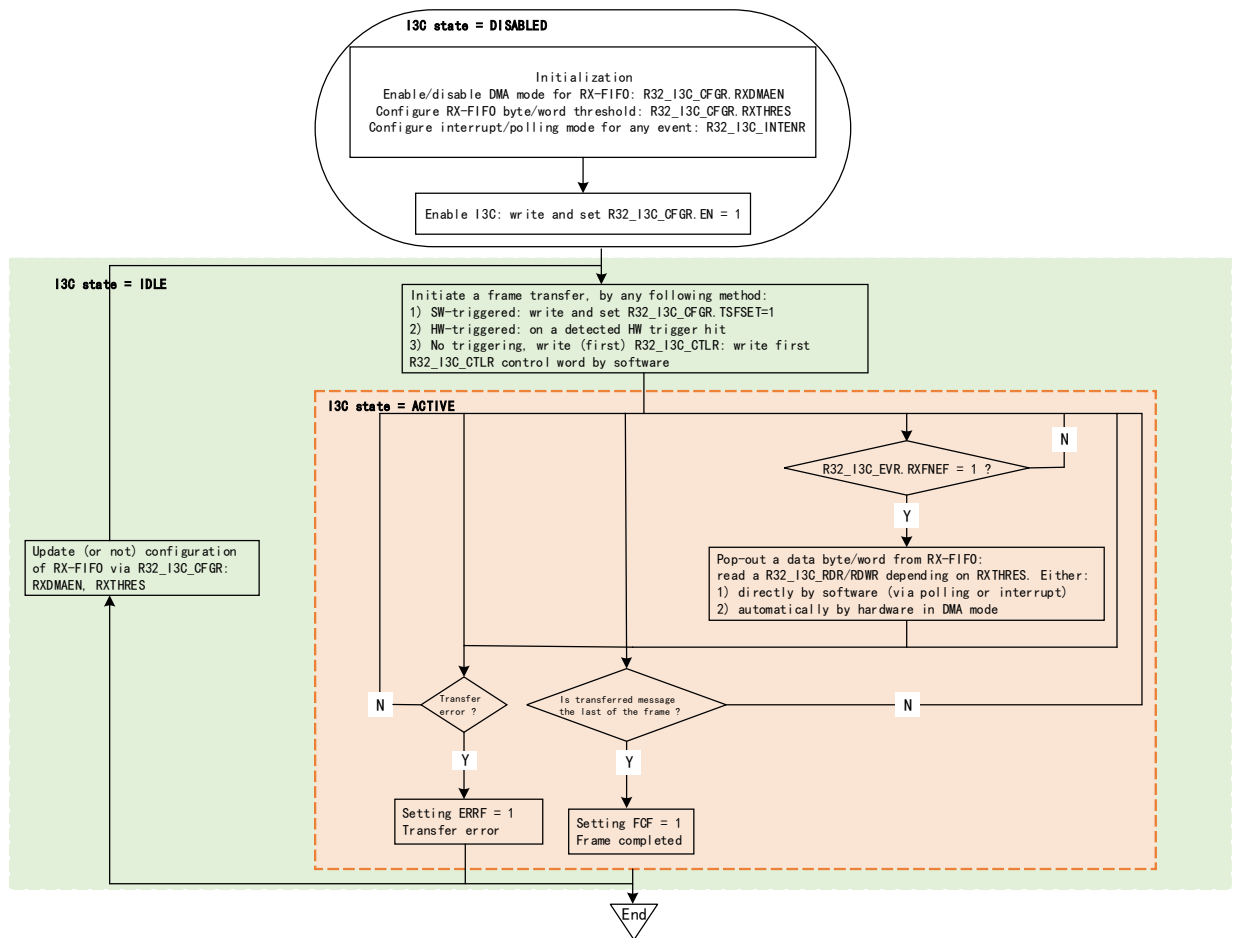
同理, 只要检测到硬件要求在 I3C 总线上发送下一个数据字节/字 (如果必须在 I3C 总线上发出重复起始位, TX-FIFO 未滿, 或者将在该 I3C 消息期间发送数据字节/字), 必须立即将该数据字节/字写入 TX-FIFO。

22.3.5.3 RX-FIFO 管理 (作为主设备)

用作主设备时, 可以在广播 ENTDA CCCC、直接 CCCC 读 (含直接 RSTACT CCCC 读)、私有读、传统 I2C 读期间使用 RX-FIFO。

图 22-17 说明了当 I3C 外设用作主设备时, 如何管理 RX-FIFO 以对要在 I3C 总线接收到的数据字节或字进行排队和弹出。

图 22-17 RX-FIFO 管理（作为主设备）



首先，软件必须通过 R32_I3C_CFGR 寄存器的以下位域初始化 RX-FIFO 管理：

- RXDMAEN：使能/禁止 RX-FIFO 的 DMA 模式
- RXTHRES：从 RX-FIFO 弹出数据字节或字

然后，根据 RXDMAEN 位读取 RX-FIFO：

- 软件直接按字节/字读取（RXDMAEN=0）：
 - 通过轮询模式（R32_I3C_INTENR 寄存器中的 RXFNEIE=0）：在显式读取 R32_I3C_RDR 或 R32_I3C_RDWR 寄存器之前等待硬件请求下一个数据字节/字（R32_I3C_INTENR 寄存器中的 RXFNEF=1），具体取决于 R32_I3C_CFGR 寄存器中的 RXTHRES 位
 - 通过使能的中断通知（如果 R32_I3C_INTENR 寄存器 RXFNEIE=1）
- 由分配的 DMA 通道读取（如果 RXDMAEN=1），以使能相应的 I3C 外设 DMA 请求：
 - 根据 DMA 模块级别的配置，DMA 会自动从 R32_I3C_RDR 或 R32_I3C_RDWR 寄存器弹出/读取数据字节/字（具体取决于 RXTHRES 位），并将它们写入其存储器从设备缓冲区，直到帧完成（在帧的最后一条消息后，I3C 总线上会发出停止位），但发生传输错误时除外。

I3C 消息以起始位或重复起始位开始，以停止位或重复起始位结束。在消息级别，从 I3C 总线接收到的最后一个数据字节/字由 I3C_EVR 寄存器中的 IRLASTF=1 进行指示。当 I3C 帧包含多条消息（由重复起始位分隔）时，软件可以使用该事件来更新相应的指针，使之指向用于存储下一条消息的数据字节/字的缓冲区。

如果 RX-FIFO 已满并且在 I3C 总线上接收到数据字节，则会报告 RX-FIFO 上溢（R32_I3C_EVR 寄存器中的 ERRF=1 且 R32_I3C_STATERR 寄存器中的 DOVR=1）。如果使能了相应的中断（R32_I3C_INTENR 寄存器中的 ERRIE=1），则会生成中断。

当 I3C 外设未处于工作状态时，可以修改 RX-FIFO 管理的配置。

从设备提前终止读传输

禁止了 S-FIFO（R32_I3C_CFGR 寄存器中的 SMODE=0），将通过 R32_I3C_EVR 寄存器中的 RXTGTENDF=1 和相应的中断（如果使能）通知软件发生读传输提前终止。然后，软件可以读取状态寄存器 R32_I3C_STATR 来检查与最后一条消息相关的信息，并获取提前结束的读传输中接收到的数据字节数（R32_I3C_STATR 寄存器中的 XDCNT[15:0]）。

在任何情况下，如果使能了 S-FIFO（R32_I3C_CFGR 寄存器中的 SMODE=1），软件或 DMA（具体取决于 R32_I3C_CFGR 寄存器中的 SDMAEN）必须针对每一条消息读取一次状态寄存器 R32_I3C_STATR。提前结束的读消息中接收到的有效数据字节数由 R32_I3C_STATR 寄存器中的 XDCNT[15:0]（随后 ABT=1）报告。

22.3.5.4 S-FIFO 管理（作为主设备）

用作主设备时，软件可以使用 S-FIFO 为每条传输的消息读取 I3C 状态寄存器（R32_I3C_STATR）。

禁止 S-FIFO 时，状态寄存器的读取

当 R32_I3C_CFGR 寄存器 SMODE 位设置为 0，则 S-FIFO 被禁用，状态寄存器的读取方式与普通寄存器相同：

- 在传输新消息时，寄存器的内容由硬件覆盖
- R32_I3C_STATR 包含传输的最后一条消息的状态
- 如果未读取状态寄存器，SCL 时钟不会停顿

当 SMODE=0，即在特定情况下从设备提前结束私有读传输时：

- 通过 R32_I3C_EVR 寄存器 RXTGTENDF 位为 1 和相应的中断（如果使能）来通知软件
- 然后，在软件通过写操作将 R32_I3C_CEVR 寄存器中的 CRXTGTENDF 位置 1 以清零事件标志之前：
 - I3C 总线上不再接收数据字节并由硬件写入 R32_I3C_RDR/R32_I3C_RDWR 寄存器
 - 不能更新 R32_I3C_STATR
 - 使 SCL 时钟停顿（如果需要）

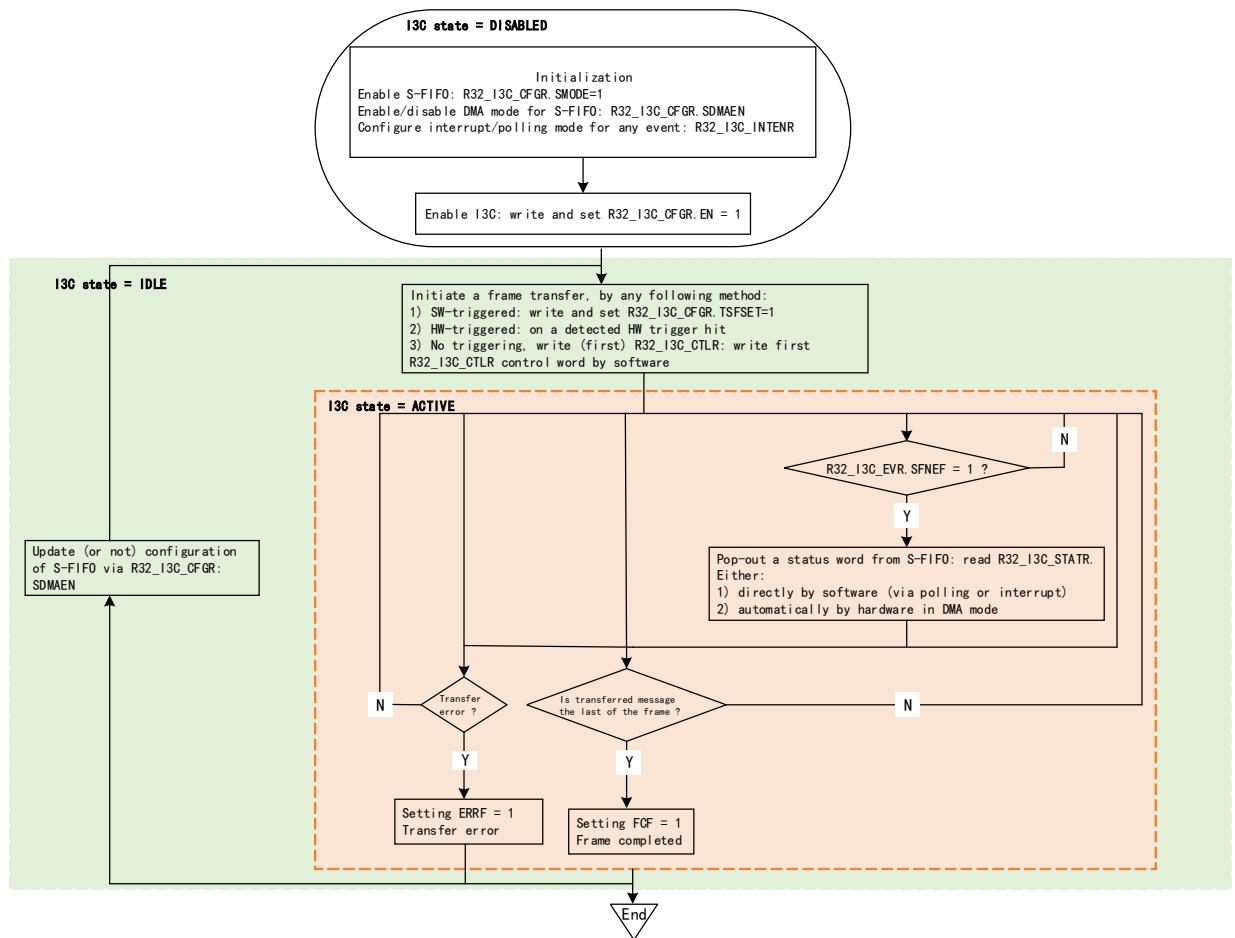
通常情况下，当 R32_I3C_EVR 寄存器的 FCF=1、ERRF=1 或 RXTGTENDF=1 时，可以读取 R32_I3C_STATR 寄存器。在从设备提前终止读传输之后，可以读取 R32_I3C_STATR 寄存器中的 XDCNT[15:0] 以获取私有读传输中接收到的有效数据字节数。

使能 S-FIFO 时读取状态寄存器

如果 R32_I3C_CFGR 寄存器 SMODE 位设置为 1，则 S-FIFO 处于使能状态。

图 22-18 说明了当 I3C 外设用作主设备时，如何管理 S-FIFO 以对 I3C 总线上执行的每条消息的状态字进行排队和弹出。

图 22-18 S-FIFO 管理（作为主设备）



首先，软件必须通过 R32_I3C_CFGR 寄存器中的 SDMAEN 位域（使能/禁止 S-FIFO 的 DMA 模式）初始化 S-FIFO 管理。然后，根据 SDMAEN 位读取 S-FIFO：

- 直接由软件读取（如果 SDMAEN=0）：
 - 通过轮询模式（R32_I3C_INTENR 寄存器中的 SFNEIE=0）：在显式读取 I3C_STATR 寄存器之前等待硬件请求下一个状态字（R32_I3C_INTENR 寄存器中的 SFNEF=1）
 - 通过使能的中断通知（SFNEIE=1）
- 由分配的 DMA 通道读取（如果 SDMAEN=1），以使能相应的 I3C 外设 DMA 请求（i3c_rs_dma）：
 - 根据 DMA 模块级别的配置，DMA 会自动从 R32_I3C_STATR 寄存器弹出/读取状态字并将它们写入其存储器从设备缓冲区，直到帧完成（在帧的最后一条消息后，I3C 总线上会发出停止位），但发生传输错误时除外。

必须读取每一个消息状态，否则当 S-FIFO 已满并且必须写入下一个消息状态时，硬件会将上溢错误标志置 1（R32_I3C_EVR 寄存器中的 ERRF=1 且 R32_I3C_STATR 寄存器中的 COVR=1）。如果使能了相应的中断（R32_I3C_INTENR 寄存器中的 ERRIE=1），则会生成中断。

只有在 S-FIFO 为空时才会报告帧完成（R32_I3C_EVR 寄存器中的 FCF=1）。

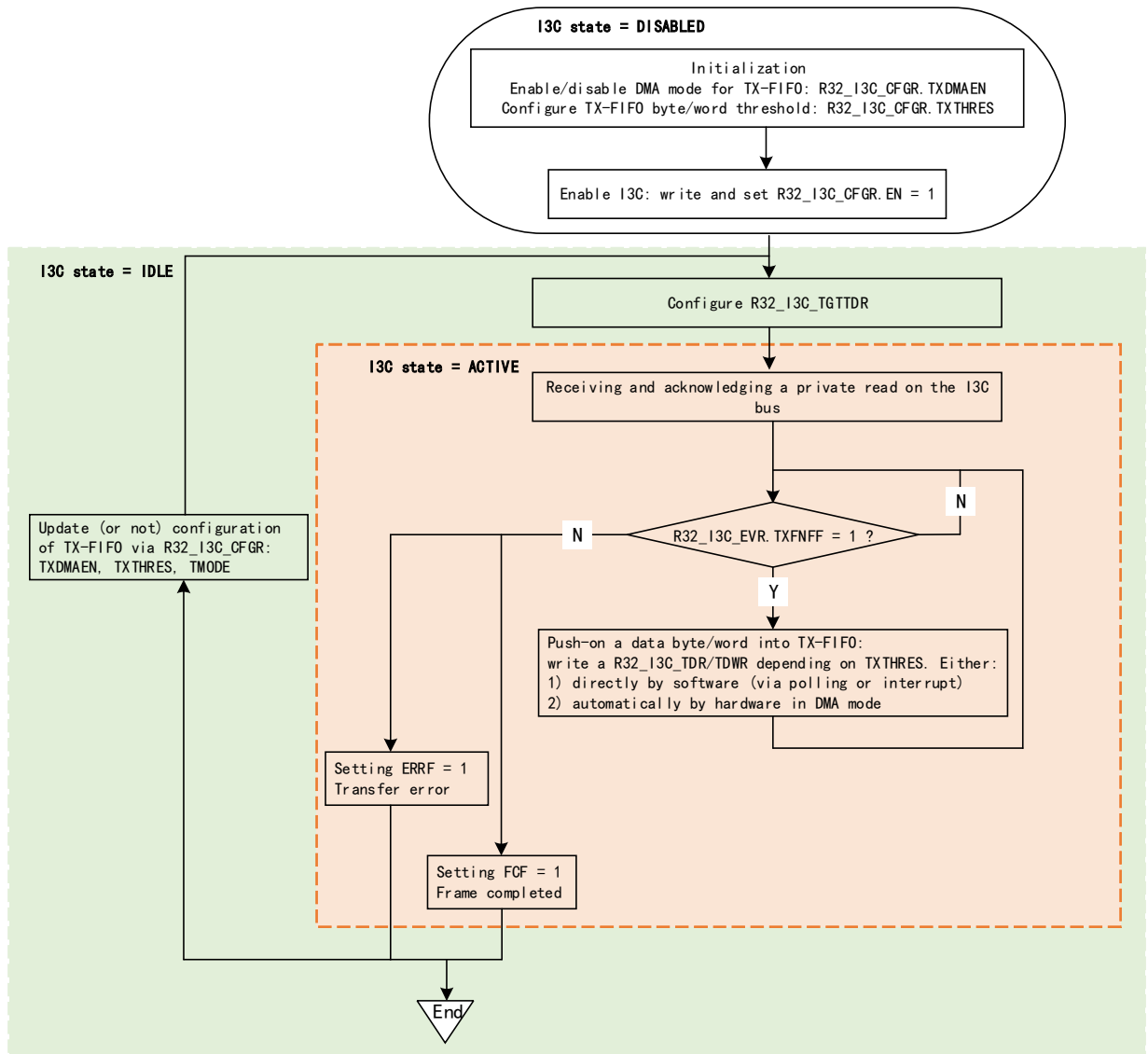
当 I3C 外设未处于工作状态时，可以修改 S-FIFO 管理的配置。

22.3.5.5 TX-FIFO 管理（作为从设备）

用作从设备时，只能在私有读传输期间使用 TX-FIFO。

图 22-19 说明了当 I3C 外设用作从设备时，如何管理 TX-FIFO 以对要在 I3C 总线上发送的数据字节或字进行排队和推入。

图 22-19 TX-FIFO 管理（作为从设备）



首先，软件必须初始化 TX-FIFO 管理并写入 R32_I3C_CFGR 中的以下位：

- TXDMAEN：使能/禁止 TX-FIFO 的 DMA 模式
- TXTHRES：将数据字节或字推入 TX-FIFO

然后，在 I3C 总线上接收私有读传输之前，软件必须配置 I3C 从设备发送配置寄存器（R32_I3C_TGTTDR）以向 TX-FIFO 预装载一定数量的数据字节（在单次访问中写入 TGTTDCNT[15:0]和 PRELOAD=1），从而准备好在 I3C 总线上发送来自从设备的数据字节：

- 如果 PRELOAD=1 且 TGTTDCNT[15:0]>TX-FIFO 大小，则 TX-FIFO 会先预装载至 FIFO 大小。
- 如果 PRELOAD=1 且 TGTTDCNT[15:0]≤TX-FIFO 大小，则 TX-FIFO 会预装载至 TGTTDCNT[15:0]。

根据 TXDMAEN 位预装载 TX-FIFO：

- 直接由软件按字节/字预装载（TXDMAEN=0）：
 - 通过轮询模式（R32_I3C_INTENR 寄存器 TXFNFF=0）：在显式写入 R32_I3C_TDR 或 R32_I3C_TDWR 寄存器之前等待硬件请求下一个数据字节/字（R32_I3C_INTENR 寄存器 TXFNFF=1），具体是字节还是字取决于 R32_I3C_CFGR 寄存器中的 TXTHRES
 - 通过使能的中断通知（如果 TXFNFF=1）
- 由分配的 DMA 通道预装载（TXDMAEN=1），以使能相应的 I3C 外设 DMA 请求（i3c_tx_dma）：

- 根据 DMA 模块级别的配置，DMA 会自动将数据字节/字从其存储器源缓冲区推入/写入 R32_I3C_TDR 或 R32_I3C_TDWR 寄存器（具体取决于 TXTHRES），直到传输完成（R32_I3C_EVR 寄存器 FCF=1），但发生传输错误时除外。

然后，当 R32_I3C_TGTTDR 寄存器的 TGTTDCNT[15:0] 中还有其余数据要装载到 TX-FIFO 以继续私有读传输（PRELOAD 置 1 且 TGTTDCNT[15:0] > TX-FIFO 大小）时，如果主设备尚未完成私有读传输，则采用同样的方式进行预装载（直接通过软件或通过分配的 DMA 通道）：

- 如果 TXTHRES=0：当在 I3C 总线上发送一个字节时，将下一个字节预装载到 TX-FIFO
- 如果 TXTHRES=1：当在 I3C 总线上发送四个字节时，将下一个字预装载到 TX-FIFO。

当从设备或主设备先终止数据字节传输时，私有读传输完成（R32_I3C_EVR 寄存器中的 FCF=1）。传输完成后，软件可以：

- 读取 R32_I3C_STATR 寄存器中的 XDCNT[15:0]：发送的有效数据字节数
- 读取 R32_I3C_TGTTDR 寄存器中的 TGTTDCNT[15:0]：I3C 总线上待装载和发送的剩余字节数
- 清空 TX-FIFO：是否通过写操作将 R32_I3C_CFGR 寄存器中的 TXFLUSH 置 1 并继续进行另一次私有读传输，取决于用户应用

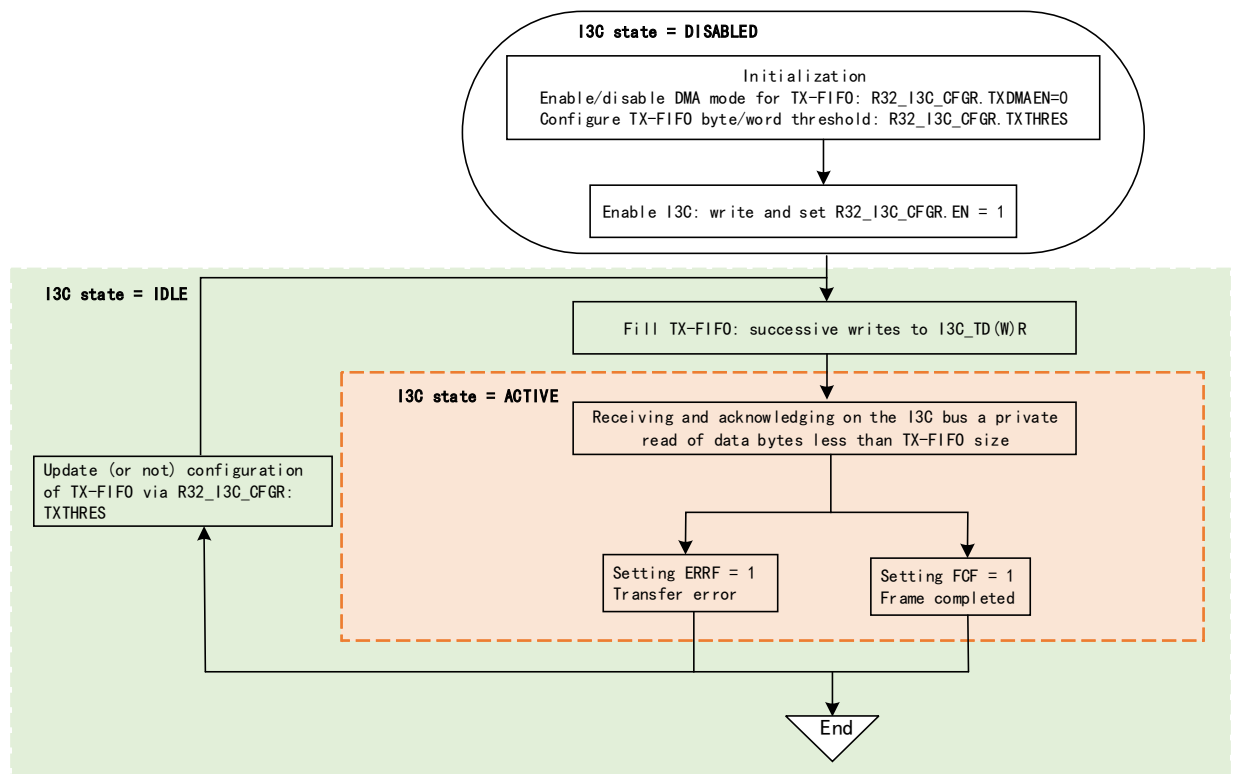
如果 TX-FIFO 为空并且必须在 I3C 总线上发送数据字节，则会报告 TX-FIFO 下溢（R32_I3C_EVR 寄存器 ERRF=1 且 R32_I3C_STATER 寄存器 DOVR=1）。如果使能了相应的中断（R32_I3C_EVR 寄存器 ERRIE=1），则会生成中断。

当 I3C 外设未处于工作状态时，可以修改 TX-FIFO 管理的配置。

如果字节数小于 TX-FIFO 大小，可以选择不使用 R32_I3C_TGTTDR

除了使用 R32_I3C_TGTTDR 寄存器之外，如图 22-20 所示，当不使用 DMA（TDMAEN=0）时，如果 I3C 总线上要读取的数据字节数小于 TX-FIFO 大小，软件可通过连续写入 R32_I3C_TDR 或 R32_I3C_TDWR（具体取决于 TXTHRES），直接使用自身直接读取的字节数来准备并填充 TX-FIFO。

图 22-20 如果读取的字节数小于 TX-FIFO 大小，则不使用 I3C_TGTTDR 而通过软件管理 TX-FIFO（作为从设备）

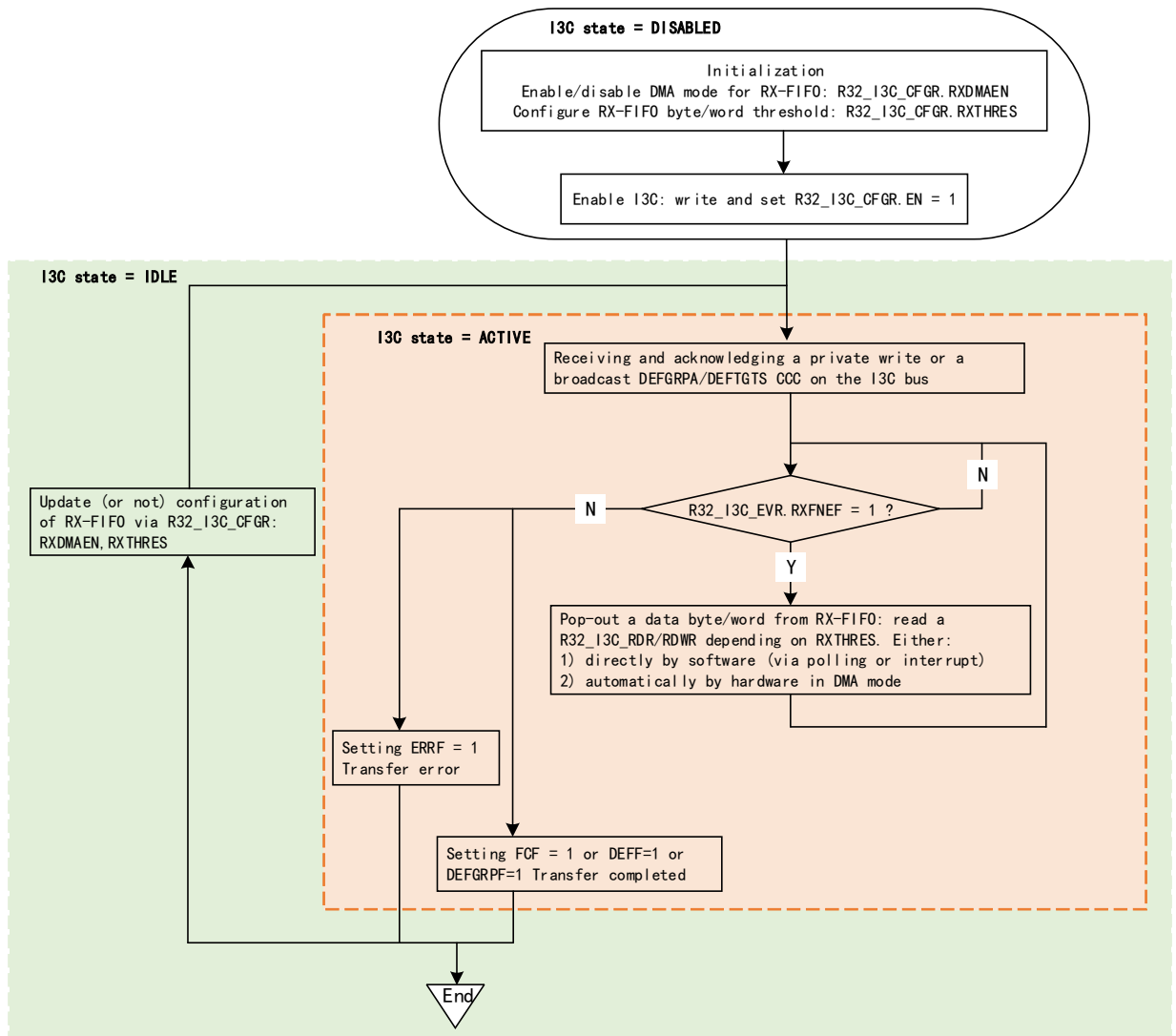


22.3.5.6 RX-FIFO 管理（作为从设备）

用作从设备时，可以在广播 DEFTGTS CCC、广播 DEFGRPA CCC、私有写的已接收和确认的传输期间使用 RX-FIFO：

图 22-21 说明了当 I3C 外设用作从设备时，如何管理 RX-FIFO 以对从 I3C 总线接收到的数据字节/字进行排队和弹出。

图 22-21 RX-FIFO 管理（作为从设备）



首先，软件必须通过以下 R32_I3C_CFGR 寄存器位域初始化 RX-FIFO 管理：

- RXDMAEN：使能/禁止 RX-FIFO 的 DMA 模式；
- RXTHRES：从 RX-FIFO 弹出数据字节或字；

然后，根据 RXDMAEN 位读取 RX-FIFO：

- 直接由软件按字节/字读取（如果 RXDMAEN=0）：
 - 通过轮询模式（R32_I3C_INTENR 寄存器中的 RXFNEIE=0）：在显式读取 R32_I3C_RDR 或 R32_I3C_RDWR 寄存器之前等待硬件请求下一个数据字节/字（R32_I3C_INTENR 寄存器中的 RXFNEF=1），具体是字节还是字取决于 R32_I3C_CFGR 寄存器中的 RXTHRES
 - 通过使能的中断通知（如果 RXFNEIE=1）
- 由分配的 DMA 通道读取（如果 RXDMAEN=1），以使能相应的 I3C 外设 DMA 请求（I3C_RX_DMA）：
 - 根据 DMA 模块级别的配置，DMA 会自动从 R32_I3C_RDR 或 R32_I3C_RDWR 寄存器弹出/读取数据字节/字（具体取决于 RXTHRES），并将它们写入其存储器从设备缓冲区，直到传输完成（R32_I3C_EVR 寄存器中的 FCF=1（私有写）、GRPF=1（DEFGRPA CCC）或 DEFF=1（DEFTGTS CCC）），但发生传输错

误时除外。

如果 RX-FIFO 已满并且在 I3C 总线上接收到新的数据字节，则会报告 RX-FIFO 上溢（R32_I3C_EVR 寄存器中的 ERRF=1 且 R32_I3C_STATER 寄存器中的 DOVR=1），并生成相应的中断（如果使能）。

当传输完成时（R32_I3C_EVR 寄存器中的 FCF=1、GRPF=1 或 DEFF=1）：

- RX-FIFO 为空
- 在软件尚未处理完与已完成的私有写/DEFTGTS/DEFGRPA 传输相对应的 RX 数据缓冲区（也就是软件还未通过写操作将 R32_I3C_CEVR 寄存器中的 CFCF/CDEFF/CGRPF 置 1 以清零相应的标志）之前，如果接收到下一个私有写/DEFTGTS CCC/DEFGRPA CCC，并且硬件必须将数据字节写入 RX-FIFO，则会报告 RX-FIFO 上溢错误（R32_I3C_EVR 寄存器中的 ERRF=1 且 R32_I3C_STATER 寄存器 DOVR=1）并生成相应的中断（如果使能）。

当 I3C 外设未处于工作状态时，可以修改 RX-FIFO 管理的配置。

22.3.6 I3C 错误管理

22.3.6.1 主设备错误管理

表 22-3 主设备错误管理

错误类型	说明	错误检测	主设备操作	报告错误
CE0	非法格式化CCC（例如返回的数据字节较少）	直接 CCC 读传输时从设备提前结束读取数据	硬件发出停止位	ERRF=1、PERR=1 且 CODERR[3:0]=0000b
CE1	监视错误	在传统 I2C 读传输结束时检测到不正确的 ACK	硬件使 SCL 保持运行状态并持续九个时钟周期，可以交换另一个字节，然后再次发出 NACK，后跟停止位	ERRF=1、PERR=1 且 CODERR[3:0]=0001b
		I3C SDR 读传输结束时无法生成重复起始位或停止位 检测到 CE1 错误后，无法生成起始位	SCL 保持运行状态。软件可通过向 R32_I3C_CTLR 寄存器写入控制字 MTYPE[3:0]=0000（报头消息）来停止 SCL，然后通常必须等待至少 150μs 才能发出另一条消息	
CE2	对广播地址无响应（1111110b）	在消息发送过程中（升级故障或复位时序消息除外），检测到报头（1111110b+RnW=0）被 NACK	硬件发出 HDR 退出时序，后跟停止位	ERRF=1、PERR=1 且 CODERR[3:0]=0010b
CE3	主设备切换失败	新主设备在 SDA 变为低电平后（通过测试报头或来自从设备的启动请求启动后）以及在其工作状态定义的延迟	硬件发出起始位 +1111110b+RnW=0，后跟来自从设备的 ACK/NACK，最后是停止位	ERRF=1、PERR=1 且 CODERR[3:0]=0011b

		时间结束后尚未将 SCL 驱动为低电平		
-	在 GETACCCR CCC 中返回的带奇偶校验位的 7 位从设备地址不正确	检测到不正确的动态地址和/或奇偶校验位	硬件取消 GETACCCR CCC，方法是发出重新起始位 +1111110b+RnW=0，后跟来自从设备的 ACK/NACK，最后是停止位。	ERRF=1 且 DERR=1
-	被寻址的从设备首次在直接 CCC 读传输中被 NACK	检测到 NACK	硬件执行单次重试，方法是发出重复起始位+7 位相同从设备地址 +RnW=1。	-
-	被寻址的从设备第二次在直接 CCC 写传输、I3C 私有读/写传输、传统 I2C 传输或直接 CCC 读传输中被 NACK	检测到 NACK	硬件发出停止位。	ERRF=1 且 ANACK=1
-	分配/发送的带奇偶校验位的动态地址首次在 ENTDAACCC 中被 NACK	检测到 NACK	硬件执行单次重试分配循环，方法是发出重复起始位 +1111110b+RnW=1，后跟来自（最高优先级）从设备的 ACK 和 8 字节读数据，后面是分配的地址+奇偶校验位。	-
-	分配/发送的带奇偶校验位的动态地址第二次在 ENTDAACCC 中被 NACK	检测到 NACK	硬件发出停止位。	ERRF=1 且 DNACK=1
-	写数据在传统 I2C 写传输中被 NACK			
-	相对于 I3C 总线时序，未及时写入/读取控制字、状态字、发送数据或读数据	SCL 停顿超时	硬件发出停止位。	ERRF=1 且 (COVR=1 或 DOVR=1)

22.3.6.2 从设备错误管理

表 22-4 从设备错误管理

错误类型	说明	错误检测	主设备操作	报告错误

TE0	DAA分配后的广播地址（1111110b+RnW=0）无效或7位动态地址+RnW=1无效	在起始位或重复起始位后检测到禁止使用的地址	硬件等待 HDR 退出时序	ERRF=1、PERR=1且 ODERR[3:0]=1000b
TE1	CCC 代码	检测到 CCC 代码中有奇偶校验错误		ERRF=1、PERR=1且 ODERR[3:0]=1001b
TE2	写数据	在 I3C 私有写消息中检测到写数据字节中有奇偶校验错误	硬件等待停止位或重复起始位	ERRF=1、PERR=1且 ODERR[3:0]=1010b
TE3	动态地址仲裁期间分配的地址	在 ENTDAACCC 中检测到分配的动态地址中有奇偶校验错误		ERRF=1、PERR=1且 ODERR[3:0]=1011b
TE4	动态地址仲裁期间 Sr 位后缺少 1111110b+RnW=1	检测到动态地址仲裁期间 Sr 位后缺少 {1111110b+RnW=1}		ERRF=1、PERR=1且 ODERR[3:0]=1100b
TE5	检测到 CCC 后的事务	在 CCC 直接读/写传输中，检测到无效的 CCC 方向（相对于地址阶段提供的方向）		ERRF=1、PERR=1且 ODERR[3:0]=1101b
TE6	监视错误	在 SDR 数据读传输期间（在直接 CCC 读、私有读或 IBI 传输期间）检测到以异常值驱动 SDA		ERRF=1、PERR=1且 ODERR[3:0]=1110b
-	SDR 数据读传输期间监视 SCL	在 SDR 数据读传输期间（在直接 CCC 读、私有读或 IBI 传输期间）检测到 SCL 的稳定时间超过 125μs	硬件释放 SDA，然后等待停止位或重复起始位	ERRF=1 且 STALL=1
-	待发送的数据未及时写入/不可用（相对于 I3C 总线时序）	在私有读传输中，软件或 DMA 未及时写入待发送的数据	硬件检测到停止位或重复起始位	ERRF=1且DOVR=1
-	无法及时读取接收到的数据（相对于 I3C总线时序）	在私有写、DEFTGTS 或 DEFGRPA CCC 中，软件或 DMA 未及时读取接收到的数据		

22.3.7 I3C 低功耗模式

低功耗模式下的 I3C 处于睡眠模式时不受其影响，I3C 中断可使器件退出睡眠模式。

当处于低功耗模式时，SCL 和 SDA 总线上的变化会唤醒系统，退出低功耗模式。

22.3.8 I3C 中断

表 22-5 I3C 中断请求

中断事件	中断划分	用作主设备	用作从设备	中断使能控制	事件标志	事件清除控制
请求控制字	请求	×	—	CFNFIE	CFNFF	—
状态字可用		×	—	SFNEIE	SFNEF	—
请求待发送的数据		×	×	TXFNFI E	TXFNFF	—
请求待发送的数据		×	×	RXFNEIE	RXFNEF	—
主设备：帧传输完成 从设备：私有传输完成		×	×	FCIE	FCF	CFCF
从设备提前结束私有读传输（并且 R32_I3C_CFGR 寄存器 SMODE 位为 0）		×	—	RXTGTENDIE	RXTGTENDF	CRXTGTENDF
接收到 IBI 请求		×	—	IBIIE	IBIF	CIBIF
接收到主设备角色请求		×	—	CRIE	CRF	CCRIF
接收到热加入请求		×	—	HJIE	HJF	CHJF
IBI 请求完成		—	×	IBIENDIE	IBIENDF	CIBIENDF
I3C 总线起始位丢失		—	×	WKPIE	WKPF	CWKPF
接收到直接 GETACCR CCC		—	×	CRUPDIE	CRUPDF	CCRUPDF
接收到直接 GETSTATUS CCC		—	×	STAIE	STAF	CSTAF
接收到任何直接 GETxxx CCC（GETSTATUS 除外）		—	×	GETIE	GETF	CGETF
动态地址更新（接收到广播 ENTDA A、广播 RSTDAA 或直接 SETNEWDA）		—	×	DAUPDIE	DAUPDF	CDAUPDF
接收到直接 SETMWL CCC		—	×	MWLUPDIE	MWLUPDF	CMWLUPDF
接收到直接 SETMRL CCC		—	×	MRLUPDIE	MRLUPDF	CMRLUPDF
检测到复位时序		—	×	RSTIE	RSTF	CRSTF
总线工作状态更新（接收到直接/广播 ENTASx CCC）		—	×	RSTIE	ASUPDF	CASUPDF
接收到广播 / 直接 ENEC/DISEC CCC		—	×	INTUPDIE	INTUPDF	CINTUPDF
接收到广播 DEFTGTS CCC		—	×	DEFIE	DEFF	CDEFF
接收到广播 DEFGRPA CCC		—	×	GRPIE	GRPF	CGRPF
发生错误	错误	×	×	ERRIE	ERRF	CERRF

22.4 寄存器描述

表 22-6 I3C 相关寄存器列表

名称	访问地址	描述	复位值
R32_I3C_CTLR	0x40014400	I3C 控制寄存器	0x00000000

R32_I3C_CFGR	0x40014404	I3C 配置寄存器	0x00000000
R32_I3C_RDR	0x40014410	I3C 接收数据字节寄存器	0x00000000
R32_I3C_RDWR	0x40014414	I3C 接收数据字节寄存器	0x00000000
R32_I3C_TDR	0x40014418	I3C 发送数据字节寄存器	0x00000000
R32_I3C_TDWR	0x4001441C	I3C 发送数据字节寄存器	0x00000000
R32_I3C_IBIDR	0x40014420	I3C IBI 有效负载数据寄存器	0x00000000
R32_I3C_TGTTDR	0x40014424	I3C 从设备发送配置寄存器	0x00000000
R32_I3C_RESET	0x4001442C	I3C 复位寄存器	0x03000000
R32_I3C_STATR	0x40014430	I3C 状态寄存器	0x00000000
R32_I3C_STATER	0x40014434	I3C 状态错误寄存器	0x00000000
R32_I3C_RMR	0x40014440	I3C 接收消息寄存器	0x00000000
R32_I3C_EVR	0x40014450	I3C 事件寄存器	0x00000003
R32_I3C_INTENR	0x40014454	I3C 中断使能寄存器	0x00000000
R32_I3C_CEV	0x40014458	I3C 清零事件寄存器	0x00000000
R32_I3C_DEVR0	0x40014460	I3C 设备寄存器 0	0x00000000
R32_I3C_DEVR1	0x40014464	I3C 设备寄存器 1	0x00000000
R32_I3C_DEVR2	0x40014468	I3C 设备寄存器 2	0x00000000
R32_I3C_DEVR3	0x4001446C	I3C 设备寄存器 3	0x00000000
R32_I3C_DEVR4	0x40014470	I3C 设备寄存器 4	0x00000000
R32_I3C_MAXRLR	0x40014490	I3C 最大读取长度寄存器	0x00000000
R32_I3C_MAXWLR	0x40014494	I3C 最大写入长度寄存器	0x00000000
R32_I3C_TIMINGR0	0x400144A0	I3C 时序寄存器 0	0x00000000
R32_I3C_TIMINGR1	0x400144A4	I3C 时序寄存器 1	0x00000000
R32_I3C_TIMINGR2	0x400144A8	I3C 时序寄存器 2	0x00000000
R32_I3C_BCR	0x400144C0	I3C 总线特征寄存器	0x00000000
R32_I3C_DCR	0x400144C4	I3C 设备特征寄存器	0x00000000
R32_I3C_GETCAPR	0x400144C8	I3C 获取功能寄存器	0x00000000
R32_I3C_CRCAPR	0x400144CC	I3C 主设备角色功能寄存器	0x00000000
R32_I3C_GETMSR	0x400144D0	I3C 获取最大数据速度寄存器	0x00000000
R32_I3C_EPIDR	0x400144D4	I3C 扩展配置 ID 寄存器	0x00000000

22.4.1 I3C 控制寄存器 (R32_I3C_CTLR)

偏移地址: 0x00

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

位	名称	访问	描述	复位值
31	MEND	WO	消息结束类型/帧的最后一条消息： 1: 来自主设备的此消息以停止位结束，是帧的最后一条消息； 0: 来自主设备的此消息后跟重复起始位，然后才能发出另一条消息。	0

			注：此位仅适用于主设备。	
[30:27]	MTYPE[3:0]	WO	消息类型。 当 I3C 用作主设备时： 0000：SCL 时钟强制停止直到执行下一个控制字； 0001：报头消息； 0010：私有消息； 0011：直接消息（I3C SDR 直接 CCC 命令的第二部分）； 0100：传统 I2C 消息； 其他：保留。 当 I3C 用作从设备时： 1000：热加入请求； 1001：主设备角色请求； 1010：IBI（带内中断）请求； 其他：保留。	0
[26:24]	Reserved	RO	保留。	0
[23:17]	ADD[6:0]	WO	7 位 I3C 动态/I2C 静态从设备地址。 注：当 I3C 用作主设备时，如果 MTYPE[3:0] 为 0010b、0011b 或 0100b 时使用该位域。	0
16	RNW	WO	读/非写消息。 1：读消息； 0：写消息。 注：当 I3C 用作主设备时，如果 MTYPE[3:0] 为 0010b、0011b 或 0100b 时使用该位域。	0
[15:0]	DCNT[15:0]	WO	读/写消息期间要传输的数据计数，以字节为单位： 0x0000：没有数据要传输； 0x0001：1 字节； 0x0002：2 字节； 0xFFFF：64KB - 1 字节。 当 I3C 用作主设备时，如果 MTYPE[3:0] 为 0010b、0011b 或 0100b 时使用该位域来设置在总线上交换的数据字节数。如果是私有或传统 I2C 读/写消息，则该位域必须非空。 当 I3C 用作从设备时，如果 MTYPE[3:0] 为 1010b 且存在任何 IBI 数据有效负载（I3C_BCR 寄存器 BCR2 位为 1 时要发送的数据），则使用该位域来设置 IBI 数据有效负载的字节数（1、2、3 或 4）。	0

22.4.2 I3C 控制寄存器[复用] (R32_I3C_CTLR)

偏移地址：0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MEND	MTYPE[3:0]			Reserved			CCC[7:0]								

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

位	名称	访问	描述	复位值
31	MEND	WO	消息结束类型/帧的最后一条消息： 1：来自主设备的此消息以停止位结束，是帧的最后一条消息； 0：来自主设备的此消息后跟重复起始位，然后才能发出另一条消息。 注：此位仅适用于主设备。	0
[30:27]	MTYPE[3:0]	WO	消息类型。 当 I3C 用作主设备时： 0110：广播/直接 CCC 命令； 位[23:16]（CCC[7:0]）是发出的 8 位 CCC 代码； 位[15:0]（DCNT[15:0]）是 CCC 释义字节、CCC 子命令字节或 CCC 数据字节的数量。 如果位[23] = CCC[7] = 1：消息为 I3C SDR 直接 CCC 命令的第一部分传输的直接 CCC 命令（第一部分）消息为： - {S/S + 0b111_1110 + RnW = 0/Sr+*} + （直接 CCC + T） + （8 位数据 + T）* + Sr； - 在 S（起始位）之后，根据 R32_I3C_CFGR 寄存器 NOARBH 位决定是否插入可仲裁头（0b111_1110 + RnW = 0）。 - Sr+*：在 Sr（重复起始位）之后，硬件会自动插入（0b111_1110 + R/W）。 如果位[23] = CCC[7] = 0：消息为 I3C SDR 广播 CCC 命令；传输的广播 CCC 命令消息为： - {S/S + 0b111_1110 + RnW = 0/Sr + *} + （广播 CCC + T） + （8 位数据 + T）* + Sr/P - 在 S（起始位）之后，根据 NOARBH 位决定是否插入可仲裁头（0b111_1110 + RnW = 0）。 - Sr+*：在 Sr（重复起始位）之后，硬件会自动插入（0b111_1110 + R/W）。 其他值：保留。	0
[26:24]	Reserved	RO	保留。	0
[23:16]	CCC[7:0]	WO	8 位 CCC 代码： 如果位[23] = CCC[7] = 1，则消息为 I3C SDR 直接 CCC 命令的第一部分。 如果位[23] = CCC[7] = 0，则消息为 I3C SDR 广播 CCC 命令（包括 ENTDA）。 注：此位仅适用于主设备。	0
[15:0]	DCNT[15:0]	WO	要作为 CCC 释义字节、CCC 子命令字节或 CCC 数据字节传输的 CCC 命令相关数据的计数（以字节为单位）：	0

		线性编码最长可达 64KB - 1。 0x0000: 没有数据要传输; 注: 发出 ENTDA 广播 CCC 时强制使用的值。 0x0001: 1 字节; 注: 发出 RSTACT 直接/广播 CCC 时强制使用的值。 0x0002: 2 字节; ... 0xFFFF: 64KB - 1 字节。	

22.4.3 I3C 配置寄存器 (R32_I3C_CFGR)

偏移地址: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TSFSET	Reserved								CFLUSH	CDMAEN	TMODE	SMODE	SFLUSH	SDMAEN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TXT HRES	TXF LUSH	TXD MAEN	Reserved	RXT HRES	RXF LUSH	RXD MAEN	HJACK	Reserved		EXIPTRN	RSTPTRN	NOARBH	CRINIT	EN

位	名称	访问	描述	复位值
31	Reserved	RO	保留。	0
30	TSFSET	WO	帧传输设置: 1: 硬件置位 CFNFF 位, 以启动帧传输; 0: 不执行任何操作。 注: (1) 此位仅适用于主设备。 (2) 如果该位未置 1, 可通过软件启动帧传输, 在 C-FIFO 为空 (R32_I3C_EVR 寄存器 CFEF 位为 1) 时直接写入第一个控制字寄存器 (R32_I3C_CTLR)。然后, 如果第一个写入的控制字未标记为消息结束 (R32_I3C_CTLR 寄存器 MEND 位为 0), 则会导致硬件将 CFNFF 置 1。 注: 此位仅适用于主设备。	0
[29:22]	Reserved	RO	保留。	0
21	CFLUSH	WO	C-FIFO 清空: 1: 清空 C-FIFO; 0: 不执行任何操作。 注: 此位仅适用于主设备。	0
20	CDMAEN	RW	C-FIFO DMA 请求使能: 1: 针对 C-FIFO 使能 DMA 模式; 0: 针对 C-FIFO 禁止 DMA 模式。 注: 此位仅适用于主设备。	0
19	TMODE	RW	I3C 总线上发出的帧来管理 C-FIFO 和 TX-FIFO: 1: 在开始发起帧传输之前, 先预装载 C-FIFO 和 TX-FIFO。 0: 在开始发起帧传输之前, 不预装载 C-FIFO 和	0

			TX-FIFO。第一个控制字出现在 C-FIFO 中时，帧传输立即开始。 注：此位仅适用于主设备。	
18	SMODE	RW	S-FIFO 使能/状态接收模式： 当 I3C 用作主设备时： 1：使能 S-FIFO； 0：禁止 S-FIFO。 当 I3C 用作从设备时，该位必须清零。	0
17	SFLUSH	WO	S-FIFO 清空： 1：清空 S-FIFO； 0：不执行任何操作。 注：此位仅适用于主设备。	0
16	SDMAEN	RW	S-FIFO DMA 请求使能： 1：使能通过 DMA 模式读取状态寄存器 R32_I3C_STATER； 0：禁止通过 DMA 模式读取状态寄存器 R32_I3C_STATER。 注：此位仅适用于主设备。	0
15	Reserved	RO	保留。	0
14	TXTHRES	RW	TX-FIFO 阈值： 1：1 字/4 字节阈值； 0：1 字节阈值。	0
13	TXFLUSH	WO	TX-FIFO 清空： 1：清空 TX-FIFO； 0：不执行任何操作。	0
12	TXDMAEN	RW	TX-FIFO DMA 请求使能： 1：针对 TX-FIFO 使能 DMA 模式； 0：针对 TX-FIFO 禁止 DMA 模式。	0
11	Reserved	RO	保留。	0
10	RXTHRES	RW	RX-FIFO 阈值： 1：1 字/4 字节阈值； 0：1 字节阈值。	0
9	RXFLUSH	WO	RX-FIFO 清空： 1：清空 RX-FIFO； 0：不执行任何操作。	0
8	RXDMAEN	RW	RX-FIFO DMA 请求使能： 1：针对 RX-FIFO 使能 DMA 模式； 0：针对 RX-FIFO 禁止 DMA 模式。	0
7	HJACK	RW	热加入请求确认： 1：热加入请求被确认； 0：热加入请求不被确认。 注：此位仅适用于主设备。	0
[6:5]	Reserved	RO	保留。	0
4	EXITPTRN	RW	HDR 退出时序使能： 1：发出消息报头（R32_I3C_CTLR 寄存器	0

			MTYPE[3:0]位为 0001b) 后发送 HDR 退出时序。这用于在从设备未响应私有消息或直接读 CCC 时进行主设备错误检测和升级处理。 0: 发出消息报头 (R32_I3C_CTLR 寄存器 MTYPE[3:0]位为 0001b) 后不发送 HDR 退出时序。这用于发送报头, 在主设备角色切换后, 如果怀疑存在问题 (新主设备未在工作状态定义的延迟时间内通过访问前一个主设备来使其主设备角色), 可以以此测试总线的所有权。 注: 此位仅适用于主设备。	
3	RSTPTRN	RW	HDR 复位时序使能: 1: 在任何发出的帧 (包含 RSTACT CCC 命令) 的停止位前都插入 HDR 复位时序; 0: 在帧结束时发出标准停止位。 注: 此位仅适用于主设备。	0
2	NOARBH	RW	起始位后无可仲裁头: 1: 无可仲裁头 - 对于传统 I2C 消息或 I3C SDR 私有读/写消息, 在起始位后直接发出从设备地址。 - 这是一种更高效的选项 (当发出 0x7E 可仲裁头无用), 但只能在主设备确定被寻址的从设备器件无法同时发出 IBI 或主设备角色请求时才可以使用 (以确保对于 IBI 或 MR, 主设备在开漏模式下发出的地址与从设备器件在起始位后可以发出的同一地址之间不会出现错误解析和潜在冲突)。 0: 在起始位之后、在传统 I2C 消息或 I3C SDR 私有读/写消息之前 (默认) 发出可仲裁头 (0b111_1110 + RnW = 0)。 注: 此位仅适用于主设备。	0
1	CRINIT	RW	标记主设备/从设备角色: 1: 主设备角色; 0: 从设备角色。	0
0	EN	RW	I3C 使能: 1: 使能 I3C; 0: 禁止 I3C。	0

22.4.4 I3C 接收数据字节寄存器 (R32_I3C_RDR)

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								RDB0[7:0]							

位	名称	访问	描述	复位值

[31:8]	Reserved	R0	保留。	0
[7:0]	RDB0[7:0]	R0	I3C 总线上的 8 位接收数据。	0

22.4.5 I3C 接收数据字寄存器 (R32_I3C_RDWR)

偏移地址：0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDB3[7:0]								RDB2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDB1[7:0]								RDB0[7:0]							

位	名称	访问	描述	复位值
[31:24]	RDB3[7:0]	R0	8 位接收数据 (I3C 总线上的最新字节)。	0
[23:16]	RDB2[7:0]	R0	8 位接收数据 (I3C 总线上跟在 RDB1 之后的字节)。	0
[15:8]	RDB1[7:0]	R0	8 位接收数据 (I3C 总线上跟在 RDB0 之后的字节)。	0
[7:0]	RDB0[7:0]	R0	8 位接收数据 (I3C 总线上最早的字节)。	0

22.4.6 I3C 发送数据字节寄存器 (R32_I3C_TDR)

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
Reserved								TDB0[7:0]							

位	名称	访问	描述	复位值
[31:8]	Reserved	R0	保留。	0
[7:0]	TDB0[7:0]	R0	I3C 总线上的 8 位发送数据。	0

22.4.7 I3C 发送数据字寄存器 (R32_I3C_TDWR)

偏移地址：0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDB3[7:0]								TDB2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDB1[7:0]								TDB0[7:0]							

位	名称	访问	描述	复位值
[31:24]	TDB3[7:0]	WO	8 位发送数据 (I3C 总线上的最新字节)。	0
[23:16]	TDB2[7:0]	WO	8 位发送数据 (I3C 总线上跟在 TDB1 之后的字节)。	0
[15:8]	TDB1[7:0]	WO	8 位发送数据 (I3C 总线上跟在 TDB0 之后的字节)。	0
[7:0]	TDB0[7:0]	WO	8 位发送数据 (I3C 总线上最早的字节)。	0

22.4.8 I3C IBI 有效负载数据寄存器 (R32_I3C_IBIDR)

偏移地址：0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IBIDB3[7:0]								IBIDB2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IBIDB1[7:0]								IBIDB0[7:0]							

位	名称	访问	描述	复位值
[31:24]	IBIDB3[7:0]	RW	8 位 IBI 有效负载数据（I3C 总线上的最新字节）。	0
[23:16]	IBIDB2[7:0]	RW	8 位 IBI 有效负载数据（I3C 总线上跟在 IBIDB1[7:0]之后的字节）。	0
[15:8]	IBIDB1[7:0]	RW	8 位 IBI 有效负载数据（I3C 总线上跟在 IBIDB0[7:0]之后的字节）。	0
[7:0]	IBIDB0[7:0]	RW	8 位 IBI 有效负载数据（I3C 总线上最早的字节，MDB[7:0]强制数据字节）。	0

22.4.9 I3C 从设备发送配置寄存器（R32_I3C_TGTTDR）

偏移地址：0x24

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

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
16	PRELOAD	RW	TX-FIFO 预装载： 1：执行 TX-FIFO 预装载； 0：不执行 TX-FIFO 预装载。 注：此位仅适用于从设备。	0
[15:0]	TGTTDCNT[15:0]	RW	发送数据计数器，以字节为单位 当 PRELOAD 置 1 时，该位域必须由软件在同一访问中写入，以便定义要预装载和发送的字节数。 该位域由硬件更新，读取该位域会报告要装载到 TX-FIFO 中的剩余字节数。 注：此位仅适用于从设备。	0

22.4.10 I3C 复位寄存器（R32_I3C_RESET）

偏移地址：0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						TGT _SI E_R ST	HST _SI E_R ST	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
[31:26]	Reserved	R0	保留。	0
25	TGT_SIE_RST	RW	从设备的复位寄存： 1：从设备的协议层处于复位状态； 0：从设备的协议层处于工作状态；	1
24	HST_SIE_RST	RW	主设备的复位寄存： 1：主设备的协议层处于复位状态； 0：主设备的协议层处于工作状态；	1
[23:0]	Reserved	R0	保留，初始化时必须保持 0x1000。	0

22.4.11 I3C 状态寄存器 (R32_I3C_STATR)

偏移地址：0x30

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

位	名称	访问	描述	复位值
[31:24]	MID[7:0]	R0	消息标识符/给定帧的计数器： 当 I3C 用作主设备时，该位域用于标识 R32_I3C_STATR 状态寄存器所引用的控制字消息 (R32_I3C_CTLR)。帧的第一条消息用 MID[7:0] = 0 标识。 当 I3C 总线上的新消息控制字 (R32_I3C_CTLR) 完成时，该位域 (由硬件) 递增。每次开始新帧时都会复位该位域。 注：此位仅适用于主设备。	0
[23:19]	Reserved	R0	保留。	0
18	DIR	R0	消息方向： 1：读； 0：写。 注：ENTDAA CCC 被视为写命令。	0
17	ABT	R0	从设备提前结束私有读消息： 1：从设备提前完成； 0：从设备未提前完成。 注：此位仅适用于主设备。	0
16	Reserved	R0	保留。	0
[15:0]	XDCNT[15:0]	R0	数据计数器：	0

		条件：动态地址分配过程中（ENTDAA CCC） - 当 I3C 用作主设备时：总线上检测到的从设备数； - 当 I3C 用作从设备时：发送的字节数。 条件：对于消息期间的其他传输 - 无论 I3C 用作主设备还是从设备：消息期间从 I3C 总线读取或在 I3C 总线上发送的数据字节数。	

22.4.12 I3C 状态错误寄存器（R32_I3C_STATERR）

偏移地址：0x34

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						DER R	DNA CK	ANA CK	COV R	DOV R	STA LL	PER R	CODERR[3:0]		

位	名称	访问	描述	复位值
[31:11]	Reserved	R0	保留。	0
10	DERR	R0	数据错误： 1：当接收到的从设备地址或/和奇偶校验位不匹配时，主设备在主设备角色切换过程中（GETACCCR CCC，以前称为 GETACCMST）检测到数据错误。活动主设备保留主设备角色。 0：未检测到错误。 注：此位仅适用于主设备。	0
9	DNACK	R0	数据未确认： 1：主设备在以下过程中检测到从设备未确认数据字节： i) 传统 I2C 写传输； ii) 第二次尝试在 ENTDAA 过程中发送动态地址。 0：未检测到错误。 注：此位仅适用于主设备。	0
8	ANACK	R0	地址未确认： 1：主设备在以下过程中检测到从设备未确认静态/动态地址： i) 传统 I2C 读/写传输； ii) 直接 CCC 写传输； iii) 第二次尝试直接 CCC 读传输； iv) 私有读/写传输； 0：未检测到错误。 注：此位仅适用于主设备。	0
7	COVR	R0	C-FIFO 下溢或 S-FIFO 上溢： 1：主设备检测到： i) C-FIFO 下溢：控制 FIFO 为空，必须发出重复	0

			起始位; ii) S-FIFO 上溢: S-FIFO 已满, 新消息结束; 0: 未检测到错误。 <i>注: 此位仅适用于主设备。</i>	
6	DOVR	R0	RX-FIFO 上溢或 TX-FIFO 下溢: 1: 无论用作主设备还是从设备, 硬件都检测到: i) TX-FIFO 下溢: TX-FIFO 为空并且必须发送写数据字节; ii) RX-FIFO 上溢: RX-FIFO 已满并且接收到新的数据字节; 0: 未检测到错误。	0
5	STALL	R0	SCL 停顿错误: 1: 从设备在 I3C SDR 数据读传输期间(在直接 CCC 读、私有读或 IB 传输期间)检测到 SCL 的稳定时间超过 125 μ s; 0: 未检测到错误。 <i>注: 此位仅适用于从设备。</i>	0
4	PERR	R0	协议错误: 1: 无论用作主设备还是从设备, 硬件都检测到协议错误; 0: 未检测到错误。	0
[3:0]	CODERR[3:0]	R0	0000: CE0 错误(发送 CCC 后的事务): 主设备检测到非法格式化 CCC; 0001: CE1 错误(监视错误): 主设备检测到总线上发送的数据与预期不符; 0010: CE2 错误(对广播地址无响应): 主设备检测到未确认的广播地址(0b111_1110); 0011: CE3 错误(主设备角色切换失败): 主设备检测到新主设备在主设备角色切换后未驱动总线; 1000: TE0 错误(无效广播地址 0b111_1110+W): 从设备检测到无效的广播地址 0b111_1110+W; 1001: TE1 错误(CCC 代码): 从设备通过奇偶校验检测到 CCC 代码中有奇偶校验错误(相对于 T 位); 1010: TE2 错误(写数据): 从设备通过奇偶校验检测到写数据中有奇偶校验错误(相对于 T 位); 1011: TE3 错误(动态地址仲裁期间分配的地址): 从设备在动态地址仲裁期间通过奇偶校验检测到分配的地址中有奇偶校验错误(相对于 PAR 位); 1100: TE4 错误(动态地址仲裁期间 Sr 位后缺少 0b111_1110+R): 从设备在动态地址仲裁期间检测到 Sr 位后缺少 0b111_1110 + R;	0

			1101: TE5 错误 (检测到 CCC 后的事务): 从设备检测到非法格式化 CCC; 1110: TE6 错误 (监视错误): 从设备检测到总线上发送的数据与预期不符; 其他值: 保留。	

22.4.13 I3C 接收消息寄存器 (R32_I3C_RMR)

偏移地址: 0x40

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

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:17]	RADD[6:0]	R0	接收到的从设备地址: 当 I3C 配置为主设备时, 该位域用于记录在已确认的 IBI 或主设备角色请求期间从设备接收到的动态地址。 注: 此位仅适用于主设备。	0
16	Reserved	R0	保留。	0
[15:8]	RCODE[7:0]	R0	接收到的 CCC 代码: 当 I3C 配置为从设备时, 该位域用于记录接收到的 CCC 代码。 注: 此位仅适用于从设备。	0
[7:3]	Reserved	R0	保留。	0
[2:0]	IBIRDCNT[2:0]	R0	IBI 接收到的有效负载数据计数: 当 I3C 配置为主设备时, 该位域用于记录 R32_I3C_IBIDR 寄存器中有效接收的数据字节数。 注: 此位仅适用于主设备。	0

22.4.14 I3C 事件寄存器 (R32_I3C_EVR)

偏移地址: 0x50

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GRPF	DEF F	INT UPD F	ASU PDF	RST F	MRL UPD F	MWL UPD F	DAU PDF	STA F	GET F	WKP F	Reserved	HJF	CRU PDF	CRF	IBI ENDF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IBIF	Reserved			ERR F	RXT GTE NDF	FCF	Reserved	RXL AST F	TXL AST F	RXF NEF	TXF NFF	SFN EF	CFN FF	TXF EF	CFE F

位	名称	访问	描述	复位值
31	GRPF	R0	组寻址标志： 当 I3C 用作从设备（通常支持主设备角色）时，该标志由硬件置 1，用于指示已接收到广播 DEFGRPA CCC（定义组地址列表）。然后，软件可以存储接收到的数据，以便在获得主设备角色时使用。 该标志的清零方式是通过软件向 R32_I3C_CTLR 寄存器 CGRPF 位写入 1。 注：此位仅适用于从设备。	0
30	DEFF	R0	DEFTGTS 标志： 当 I3C 用作从设备（通常支持主设备角色）时，该标志由硬件置 1，用于指示已接收到广播 DEFTGTS CCC（定义从设备列表）。然后，软件可以存储接收到的数据，以便在获得主设备角色时使用。 该标志的清零方式是通过软件向 R32_I3C_CEVR 寄存器 CDEFF 位写入 1。 注：此位仅适用于从设备。	0
29	INTUPDF	R0	中断/主设备角色/热加入更新标志： 当 I3C 用作从设备时，该标志由硬件置 1，用于指示已接收到直接或广播 ENEC/DISEC CCC（使能/禁止从设备事件），其中从设备事件为中断/IBI 请求、主设备角色请求或热加入请求。 然后，软件必须分别读取 R32_I3C_DEVR0 寄存器中的 IBIEN、CREN 或 HJEN。 该标志的清零方式是通过软件向 R32_I3C_CEVR 寄存器 CINTUPDF 位写入 1。 注：此位仅适用于从设备。	0
28	ASUPDF	R0	工作状态更新标志： 当 I3C 用作从设备时，该标志由硬件置 1，用于指示已接收到直接或广播 ENTAS_x CCC（其中 x = 0...3）。然后，软件必须读取 R32_I3C_DEVR0 寄存器 AS[1:0]。 该标志的清零方式是通过软件向 R32_I3C_CEVR 寄存器 CASUPDF 位写入 1。 注：此位仅适用于从设备。	0
27	RSTF	R0	复位时序标志： 当 I3C 用作从设备时，该标志由硬件置 1，用于指示已检测到复位时序（SCL 为低电平时发生 14 次 SDA 跳变，后跟重复起始位和停止位）。 然后，当不处于停止模式时，软件必须读取 I3C_DEVR0 寄存器中的 RSTACT[1:0] 和 RSTVAL，以获取所需的复位级别。 – 如果 RSTVAL = 1：当 RSTF 置 1（和/或生成相应的中断（如果使能））时，R32_I3C_DEVR0 寄存器中的 RSTACT[1:0] 指示软件要执行的复位操作	0

			(如果有)。 - 如果 RSTVAL = 0: 当 RSTF 置 1 (和/或生成相应的中断(如果使能))时, 软件必须在第一个检测到的复位时序后发出 I3C 复位, 并在第二个检测到的复位时序后发出系统复位。 当处于停止模式时, 可以使用相应的中断来唤醒器件。 该标志的清零方式是通过软件向 R32_I3C_CEVCR 寄存器中的相应 CRSTF 位写入 1。 注: 此位仅适用于从设备。	
26	MRLUPDF	R0	最大读取长度更新标志: 当 I3C 用作从设备时, 该标志由硬件置 1, 用于指示已接收到直接 SETMRL CCC (设置最大读取长度)。然后, 软件必须读取 R32_I3C_MAXRLR 寄存器 MRL[15:0]位以获取最大读取长度值。 该标志的清零方式是通过软件向 R32_I3C_CEVCR 寄存器 CMRLUPDF 位写入 1。 注: 此位仅适用于从设备。	0
25	MWLUPDF	R0	最大写入长度更新标志: 当 I3C 用作从设备时, 该标志由硬件置 1, 用于指示已接收到直接 SETMWL CCC (设置最大写入长度)。然后, 软件必须读取 R32_I3C_MAXRLR 寄存器 MWL[15:0]位以获取最大写入长度值。 该标志的清零方式是通过软件向 R32_I3C_CEVCR 寄存器 CMWLUPDF 位写入 1。 注: 此位仅适用于从设备。	0
24	DAUPDF	R0	动态地址更新标志: 当 I3C 用作从设备时, 该标志由硬件置 1, 用于指示已通过广播 ENTDA、RSTDA 和直接 SETNEWDA CCC 中的任何一个接收到动态地址更新。然后, 软件必须读取 R32_I3C_DEVRO 寄存器 DA[6:0]和 DAVAL 以获取动态地址更新。 该标志的清零方式是通过软件向 R32_I3C_CEVCR 寄存器 CDAUPDF 位写入 1。 注: 此位仅适用于从设备。	0
23	STAF	R0	获取状态标志: 当 I3C 用作从设备时, 该标志由硬件置 1, 用于指示已接收到格式 1 的直接 GETSTATUS CCC (无释义字节或有释义字节 TGTSTAT)。 该标志的清零方式是通过软件向 R32_I3C_CEVCR 寄存器 CSTAF 位写入 1。 注: 此位仅适用于从设备。	0
22	GETF	R0	获取标志: 当 I3C 用作从设备时, 该标志由硬件置 1, 用于指示已接收到除格式 1 的 GETSTATUS 之外 (但包括格式 2 的 GETSTATUS) 的任意获取类型的直接 CCC	0

			(GET*** CCC)。 该标志的清零方式是通过软件向 R32_I3C_CEVR 寄存器 CGETF 位写入 1。 注：此位仅适用于从设备。	
21	WKPF	R0	唤醒/丢失起始位标志： 当 I3C 用作从设备时，该标志由硬件置 1，用于指示已检测到起始位(SDA 下降沿后跟 SCL 下降沿)，但在下一个 SCL 下降沿，I3C 内核时钟（仍然）被门控。因此，从设备可能已丢失 I3C 总线事务。相应的中断可用于将器件从低功耗（睡眠或停止）模式唤醒。 该标志的清零方式是通过软件向 R32_I3C_CEVR 寄存器 CWKPF 位写入 1。 注：此位仅适用于从设备。	0
20	Reserved	R0	保留。	0
19	HJF	R0	热加入标志： 当 I3C 用作主设备时，该标志由硬件置 1，用于指示已接收到热加入请求。 该标志的清零方式是通过软件向 R32_I3C_CEVR 寄存器 CHJF 位写入 1。 注：此位仅适用于主设备。	0
18	CRUPDF	R0	主设备角色更新标志： 当 I3C 用作从设备时，该标志由硬件置 1，用于指示在完成主设备角色切换过程后 I3C 现已获得主设备角色。 该标志的清零方式是通过软件向 R32_I3C_CEVR 寄存器 CCRUPDF 位写入 1。 注：此位仅适用于从设备。	0
17	CRF	R0	主设备角色请求标志： 当 I3C 用作主设备时，该标志由硬件置 1，用于指示硬件已确认并完成主设备角色请求。然后，软件必须为主设备角色切换过程发出 GETACCCR CCC（获取接受主设备角色）。 该标志的清零方式是通过软件向 R32_I3C_CEVR 寄存器 CCRF 位写入 1。 注：此位仅适用于主设备。	0
16	IBIENDF	R0	IBI 结束标志： 当 I3C 用作从设备时，该标志由硬件置 1，用于指示已接收到并完成 IBI 传输（主设备确认 IBI 并读取 IBI 数据字节（如果有））。 该标志的清零方式是通过软件向 R32_I3C_CEVR 寄存器 CIBIENDF 位写入 1。 注：此位仅适用于从设备。	0
15	IBIF	R0	IBI 标志： 当 I3C 用作主设备时，该标志由硬件置 1，用于指示已接收到 IBI 请求。	0

			该标志的清零方式是通过软件向 R32_I3C_CEVCR 寄存器 CIBIF 位写入 1。 注：（1）此位仅适用于主设备； （2）接收从机发送的不带数据的 IBI 请求时，该位不生效。此类包的处理具体请参考官网 EVT 例程。	
[14:12]	Reserved	R0	保留。	0
11	ERRF	R0	标志： 该标志由硬件置 1，用于指示发生错误。然后，软件必须读取 R32_I3C_STATCR 以获取错误类型。 该标志的清零方式是通过软件向 R32_I3C_CEVCR 寄存器 CERRF 位写入 1。	0
10	RXTGTENDF	R0	从设备发起的读结束标志： 当 I3C 用作主设备时，并且只有在禁止 S-FIFO（R32_I3C_CFGR 寄存器 SMODE 位为 0）时，该标志才会由硬件置 1，用于指示从设备已提前结束读传输。然后，软件必须读取状态寄存器 R32_I3C_STATR 以检查与最后一条消息相关的信息，并获取提前结束的读传输中接收到的数据字节数（R32_I3C_STATR 寄存器 XDCNT）。 该标志的清零方式是通过软件向 R32_I3C_CEVCR 寄存器 CRXTGTENDF 位写入 1。 注：此位仅适用于主设备。	0
9	FCF	R0	帧完成标志： 当 I3C 用作主设备时，该标志由硬件置 1，用于指示 I3C 总线上的帧已（正常）完成，例如在发出停止位时。 当 I3C 用作从设备时，该标志由硬件置 1，用于指示在 I3C 总线上发送至该从设备或由该从设备寻址的消息已（正常）完成，例如在主设备发出下一个停止位或重复起始位时。 该标志的清零方式是通过软件向 R32_I3C_CEVCR 寄存器 CFCF 位写入 1。	0
8	Reserved	R0	保留。	0
7	RXLASTF	R0	最后读取的数据字节/字标志： 当 I3C 用作主设备时，该标志由硬件置 1，用于指示必须从 RX-FIFO 读取的消息的最后一个数据字节/字（具体取决于 I3C_CFGR 寄存器 RXTHRES 位）。当读取消息的最后一个数据字节/字时，该标志由硬件清零。 注：此位仅适用于主设备。	0
6	TXLASTF	R0	最后写入的数据字节/字标志： 该标志由硬件置 1，用于指示必须写入 TX-FIFO 的消息的最后一个数据字节/字（具体取决于 R32_I3C_CFGR 寄存器 TXTHRES 位）。当写入消息	0

			的最后一个数据字节/字时，该标志由硬件清零。	
5	RXFNEF	R0	RX-FIFO 非空标志： 该标志由硬件置 1/清零，用于指示必须/不得从 RX-FIFO 读取数据字节。 注：在从 RX-FIFO 读取数据字节（读取 R32_I3C_RDR 或 R32_I3C_RDWR，具体取决于 RXTHRES）之前，软件必须等待 RXFNEF = 1（通过轮询或通过使能的中断）。	0
4	TXFNFF	R0	TX-FIFO 未满载标志： 该标志由硬件置 1/清零，用于指示必须/不得向 TX-FIFO 写入数据字节。 注：（1）在写入 TX-FIFO（写入 R32_I3C_TDR 或 R32_I3C_TDWR，具体取决于 TXTHRES）之前，软件必须等待 TXFNFF = 1（通过轮询或通过使能的中断）。 （2）当 I3C 用作从设备时，如果软件要使用 TXFNFF 标志写入 R32_I3C_TDR/R32_I3C_TDWR，则必须配置 TX-FIFO 预装载（将 I3C_TGTTDR 寄存器 PRELOAD 位置 1）。	0
3	SFNEF	R0	S-FIFO 非空标志： 当 I3C 用作主设备时，如果 S-FIFO 已使能（R32_I3C_CFGR 寄存器 SMODE 位为 1），则该标志由硬件置 1，用于指示必须从 S-FIFO 读取状态字。该标志由硬件清零，用于指示不从 SFIFO 读取状态字。 注：此位仅适用于主设备。	0
2	CFNFF	R0	C-FIFO 未满载标志： 当 I3C 用作主设备时，该标志由硬件置 1，用于指示必须向 C-FIFO 写入控制字。该标志由硬件清零，用于指示不向 C-FIFO 写入控制字。 注：在写入 C-FIFO（写入 R32_I3C_CTLR）之前，软件必须等待 CFNFF = 1（通过轮询或通过使能的中断）。 注：此位仅适用于主设备。	0
1	TXFEF	R0	TX-FIFO 为空标志： 该标志由硬件置 1，用于指示 TX-FIFO 为空；该标志由硬件清零，用于指示 TX-FIFO 不为空。	1
0	CFEF	R0	C-FIFO 为空标志： 用作主设备时： 该标志由硬件置 1，用于指示 C-FIFO 为空；该标志由硬件清零，用于指示 C-FIFO 不为空。 用作从设备时： 该标志由硬件置 1，用于 R32_I3C_CTLR 寄存器中不包含控制字（无 IBI/CR/HJ 请求）；该标志由硬件清零，用于 R32_I3C_CTLR 寄存器中	1

			包含一个控制字（待处理的 IBI/CR/HJ 请求）。	
			注：当 I3C 用作主设备时，如果配置了 C-FIFO 和 TX-FIFO 预装载（R32_I3C_CFGR 寄存器 TMODE 位为 1），则软件必须等待 TXFEF = 1 和 CFEF = 1，然后才能开始新的帧传输。	

22.4.15 I3C 中断使能寄存器（R32_I3C_INTENR）

偏移地址：0x54

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GRPIE	DEFIE	INTUPDIE	ASUPDIE	RSTIE	MRLUPDIE	MWLUPDIE	DAUPDIE	STALIE	GETIE	WKP	Reserved	HJIE	CRUPDIE	CRIE	IBIENDIE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IBIIE	Reserved			ERRIE	RXTGTENDIE	FCIE	Reserved			RXFNEIE	TXFNEIE	SFNEIE	CFNFIE	Reserved	

位	名称	访问	描述	复位值
31	GRPIE	RW	DEFGRPA CCC 中断使能： 1：使能； 0：禁止。 注：此位仅适用于从设备。	0
30	DEFIE	RW	DEFTGTS CCC 中断使能： 1：使能； 0：禁止。 注：此位仅适用于从设备。	0
29	INTUPDIE	RW	ENEC/DISEC CCC 中断使能： 1：使能； 0：禁止。 注：此位仅适用于从设备。	0
28	ASUPDIE	RW	ENTASx CCC 中断使能： 1：使能； 0：禁止。 注：此位仅适用于从设备。	0
27	RSTIE	RW	复位时序中断使能： 1：使能； 0：禁止。 注：此位仅适用于从设备。	0
26	MRLUPDIE	RW	SETMRL CCC 中断使能： 1：使能； 0：禁止。 注：此位仅适用于从设备。	0
25	MWLUPDIE	RW	SETMWL CCC 中断使能：	0

			1: 使能; 0: 禁止。 注: 此位仅适用于从设备。	
24	DAUPDIE	RW	ENTDAA/RSTDAA/SETNEWDA CCC 中断使能: 1: 使能; 0: 禁止。 注: 此位仅适用于从设备。	0
23	STAIE	RW	格式 1 GETSTATUS CCC 中断使能: 1: 使能; 0: 禁止。 注: 此位仅适用于从设备。	0
22	GETIE	RW	GETxxx CCC 中断使能 (格式 1 的 GETSTATUS 除外): 1: 使能; 0: 禁止。 注: 此位仅适用于从设备。	0
21	WKPIE	RW	唤醒中断使能: 1: 使能; 0: 禁止。 注: 此位仅适用于从设备。	0
20	Reserved	RO	保留。	0
19	HJIE	RW	热加入中断使能: 1: 使能; 0: 禁止。 注: 此位仅适用于主设备。	0
18	CRUPDIE	RW	主设备角色更新中断使能: 1: 使能; 0: 禁止。 注: 此位仅适用于从设备。	0
17	CRIE	RW	主设备角色请求中断使能: 1: 使能; 0: 禁止。 注: 此位仅适用于主设备。	0
16	IBIENDIE	RW	IBI 结束中断使能: 1: 使能; 0: 禁止。 注: 此位仅适用于从设备。	0
15	IBIIE	RW	IBI 请求中断使能: 1: 使能; 0: 禁止。 注: 此位仅适用于主设备。	0
[14:12]	Reserved	RO	保留。	0
11	ERRIE	RW	错误中断使能: 1: 使能; 0: 禁止。	0
10	RXTGTENDIE	RW	从设备发起的读结束中断使能:	0

			1: 使能; 0: 禁止。 注: 此位仅适用于主设备。	
9	FCIE	RW	帧完成中断使能: 1: 使能; 0: 禁止。	0
[8:6]	Reserved	R0	保留。	0
5	RXFNEIE	RW	RX-FIFO 非空中断使能: 1: 使能; 0: 禁止。	0
4	RXFNEIE	RW	TX-FIFO 未中断使能: 1: 使能; 0: 禁止。	0
3	SFNEIE	RW	S-FIFO 非空中断使能: 1: 使能; 0: 禁止。 注: 此位仅适用于主设备。	0
2	CFNFIE	RW	C-FIFO 未中断使能: 1: 使能; 0: 禁止。 注: 此位仅适用于主设备。	0
[1:0]	Reserved	R0	保留。	0

22.4.16 I3C 清零事件寄存器 (R32_I3C_CEV)

偏移地址: 0x58

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CGRPF	CDEFF	CINTUPDF	CASUPDF	CRSTF	CMRLUPDF	CMWLUPDF	CDAUPDF	CSTAF	CGETF	CWKPF	Reserved	CHJF	CCRUPDF	CCRF	CIBIENDF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CIBIF	Reserved			CERF	CRXTGTENDF	CFCF	Reserved								

位	名称	访问	描述	复位值
31	CGRPF	WO	清零 DEFGRPA CCC 标志: 1: 将 GRPF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
30	CDEFF	WO	清零 DEFTGTS CCC 标志: 1: 将 DEFF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
29	CINTUPDF	WO	清零 ENEC/DISEC CCC 标志:	0

			1: 将 INTUPDF 清零; 0: 无影响。 注: 此位仅适用于从设备。	
28	CASUPDF	WO	清零 ENTASx CCC 标志: 1: 将 ASUPDF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
27	CRSTF	WO	清零复位时序标志: 1: 将 RSTF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
26	CMRLUPDF	WO	清零 SETMRL CCC 标志: 1: 将 MRLUPDF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
25	CMWLUPDF	WO	清零 SETMWL CCC 标志: 1: 将 MWLUPDF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
24	CDAUPDF	WO	清零 ENTDA/ RSTDA/ SETNEWDA CCC 标志: 1: 将 DAUPDF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
23	CSTAF	WO	清零格式 1 GETSTATUS CCC 标志: 1: 将 STAF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
22	CGETF	WO	清零 GETxxx CCC 标志 (格式 1 的 GETSTATUS 除外): 1: 将 GETF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
21	CWKPF	WO	清零唤醒标志: 1: 将 WKPF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
20	Reserved	RO	保留。	0
19	CHJF	WO	清零热加入标志: 1: 将 HJF 清零; 0: 无影响。 注: 此位仅适用于主设备。	0
18	CCRUPDF	WO	清零主设备角色更新标志: 1: 将 CRUPDF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
17	CCRF	WO	清零主设备角色请求标志:	0

			1: 将 CRF 清零; 0: 无影响。 注: 此位仅适用于主设备。	
16	CIBIENDF	WO	清零 IBI 结束标志: 1: 将 IBIENDF 清零; 0: 无影响。 注: 此位仅适用于从设备。	0
15	CIBIF	WO	清零 IBI 请求标志: 1: 将 IBIF 清零; 0: 无影响。 注: 此位仅适用于主设备。	0
[14:12]	Reserved	RO	保留。	0
11	CERRF	WO	清零错误标志: 1: 将 ERRF 清零; 0: 无影响。	0
10	CRXTGTENDF	WO	清零从设备发起的读结束标志: 1: 将 RXTGTENDF 清零; 0: 无影响。 注: 此位仅适用于主设备。	0
9	CFCF	WO	清零帧完成标志: 1: 将 FCF 清零; 0: 无影响。	0
[8:0]	Reserved	RO	保留。	0

22.4.17 I3C 设备寄存器 0 (R32_I3C_DEVR0)

偏移地址: 0x60

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

位	名称	访问	描述	复位值
[31:25]	Reserved	RO	保留。	0
24	RSTVAL	RO	复位操作有效: 该位由硬件置 1, 用于指示 RSTACT[1:0]位域已在接收到广播或直接写 RSTACT CGC (从设备复位操作) 时更新并生效。 当从设备接收到帧起始时, 该位由硬件清零。 当器件不处于停止模式时: - 如果 RSTVAL = 1: 当 R32_I3C_EVR 寄存器 RSTF 位置 1 (和/或生成相应的中断 (如果使能)) 时,	0

			R32_I3C_DEVR0 寄存器 RSTACT[1:0] 指示软件要执行的复位操作（如果有）。 如果 RSTVAL = 0：当 RSTF 置 1（和/或生成相应的中断（如果使能））时，软件必须在第一个检测到的复位时序后发出 I3C 复位，并在第二个检测到的复位时序后发出系统复位。 当处于停止模式时，可以使用相应的中断来唤醒器件。 注：此位仅适用于从设备。	
[23:22]	RSTACT[1:0]	R0	接收到复位时序时的复位操作/级别： 00：不执行任何复位操作 01：第一级复位：应用软件必须： a) 通过将 I3C 配置寄存器的使能位清零（写入 EN = 0）对外设进行部分复位。这会复位 I3C 总线接口和 I3C 内核的一部分，但不会修改 I3C HB 寄存器的内容（EN 位除外）。 b) 通过写操作将 RCC（复位和时钟主设备）寄存器的 I3C 复位控制位置 1，对外设（包括其所有寄存器）进行完全复位。 10：第二级复位：应用软件必须发出热复位，也称为系统复位。这与引脚复位（NRST = 0）的作用相同： 11：不执行任何复位操作。 注：此位仅适用于从设备。	0
[21:20]	AS[1:0]	R0	工作状态： 00：工作状态 0； 01：工作状态 1； 10：工作状态 2； 11：工作状态 3。 注：此位仅适用于从设备。	0
19	HJEN	RW	热加入请求使能： 1：使能； 0：禁止。 注：此位仅适用于从设备。	0
18	Reserved	R0	保留。	0
17	CREN	RW	主设备角色请求使能： 1：使能； 0：禁止。 注：此位仅适用于从设备。	0
16	IBIEN	RW	IBI 请求使能： 1：使能； 0：禁止。 注：此位仅适用于从设备。	0
[15:8]	Reserved	R0	保留。	0
[7:1]	DA[6:0]	RW	7 位动态地址： 当 I3C 用作主设备时，该位域可由软件写入，用	0

			于定义其自身的动态地址。 当 I3C 用作从设备时，该位域由硬件在接收到广播 ENTDA A CCC 或直接 SETNEWDA CCC 时更新。	
0	DAVAL	RW	动态地址有效： 当 I3C 用作主设备时，该位可由软件写入，用于验证其自身的动态地址，例如在主设备角色切换之前。 当 I3C 用作从设备时，该位由硬件在确认广播 ENTDA A CCC 或直接 SETNEWDA CCC 时置 1，在确认广播 RSTDA A CCC 时清零。	0

22.4.18 I3C 设备寄存器 x (R32_I3C_DEVRx) (x=1/2/3/4)

偏移地址：0x64, 0x68, 0x6C, 0x70

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIS	Reserved											SUS P	IBI DEN	CRA CK	IBI ACK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												DA[6:0]		Res erv ed	

位	名称	访问	描述	复位值
31	DIS	RO	DA[6:0]写禁止： 1：禁止/锁定对 DA[7:0]和 IBIDEN 的写操作； 0：允许对 R32_I3C_DEVRx 寄存器 DA[7:0]和 IBIDEN 的写操作。 <i>注：此位仅适用于主设备。</i>	0
[30:20]	Reserved	RO	保留。	0
19	SUSP	RW	收到 IBI 时挂起/停止 I3C 传输： 1：接收到来自从设备 x 的 IBI 请求后，停止 I3C 传输，并清空 C-FIFO 和 TX-FIFO； 0：确认并完成来自从设备 x 的 IBI 请求后，不清空 C-FIFO 和 TX-FIFO，并根据是否存在下一个控制字，发出重复起始位或停止位。 <i>注：此位仅适用于主设备。</i>	0
18	IBIDEN	RW	IBI 数据使能： 1：来自从设备 x 的已确认 IBI 后跟强制数据字节 MDB[7:0]； 0：来自从设备 x 的已确认 IBI 后没有数据字节。 <i>注：此位仅适用于主设备。</i>	0
17	CRACK	RW	主设备角色请求确认： 1：必须 ACK 来自从设备 x 的主设备角色请求（通过 7 位动态地址 DA[6:0]） -DIS 位域由硬件置 1，用于保护 DA[6:0]免受软件修改，同时硬件可以在内部将当前的 DA[6:0]存储	0

			到内核时钟域中。 –ACK 后，消息按照初始编程的方式继续传输。软件通过主设备角色请求标志 (R32_I3C_EVR 寄存器 CRF 位置 1) 和/或相应的中断 (如果使能) 获得通知；为了将主设备角色有效地授予请求的辅助主设备，软件必须发出 GETACCCR (以前称为 GETACCMST)，然后发出停止位。 –不管该器件或其他器件的 CRACK 配置为多少，其他主设备角色请求都将被 NACK，直到 I3C_EVR 寄存器中的主设备角色请求标志 (CRF) 和 IBI 标志 (IBIF) 都清零。 0: 必须 NACK 来自从设备 x 的主设备角色请求 NACK 后，消息按照初始编程的方式继续传输 (从设备已接收到 NACK，稍后可发出另一个主设备角色请求)。 注：此位仅适用于主设备。	
16	IBIACK	RW	IBI 请求确认： 1: 必须 ACK 来自从设备 x 的 IBI 请求 (通过 7 位动态地址 DA[6:0]) –DIS 位域由硬件置 1，用于保护 DA[6:0] 免受软件修改，同时硬件可以在内部将当前的 DA[6:0] 存储到内核时钟域中。 –ACK 后，主设备根据 I3C_DEVRx. IBIDEN 记录 IBI 有效负载数据 (如果有)。 –软件通过 IBI 标志 (IBIF=1) 和/或相应的中断 (如果使能) 获得通知； –不管该器件或其他器件的 IBIACK 配置为多少，其他 IBI 请求都将被 NACK，直到 IBI 请求标志 (IBIF) 和主设备角色请求标志 (CRF) 都清零。 0: 必须 NACK 来自从设备 x 的 IBI 请求 NACK 后，消息按照初始编程的方式继续传输 (从设备已接收到 NACK，稍后可发出另一个 IBI 请求) 注：此位仅适用于主设备。	0
[15:8]	Reserved	R0	保留。	0
[7:1]	DA[6:0]	RW	为从设备 x 分配的 I3C 动态地址： 当 I3C 用作主设备时，该位域必须由软件写入，用于存储主设备通过从设备 x 确认的广播 ENTDA 或直接 SETNEWDA CCC 发送的 7 位动态地址。 注：此位仅适用于主设备。	0
0	Reserved	R0	保留。	0

22.4.19 I3C 最大读取长度寄存器 (R32_I3C_MAXRLR)

偏移地址：0x90

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

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

位	名称	访问	描述	复位值
[31:19]	Reserved	R0	保留。	0
[18:16]	IBIP[2:0]	RW	IBI 负载数据最大大小，以字节为单位： 000：空负载数据大小（仅当 R32_I3C_BCR 寄存器 BCR2 位为 0 时允许）； 001：1 字节（强制数据字节 MDB[7:0]）； 010：2 字节（包括第一个 MDB[7:0]）； 011：3 字节（包括第一个 MDB[7:0]）； 其他：4 字节（包括第一个 MDB[7:0]）。 注：此位仅适用于从设备。	0
[15:0]	MRL[15:0]	RW	最大数据读取长度： 通过 MRLUPF 和相应的中断（如果使能）向软件通知 MRL 更新。 当从设备接收到 GETMRL CCC 时，硬件使用该位域在 I3C 总线上返回值。 注：此位仅适用于从设备。	0

22.4.20 I3C 最大写入长度寄存器（R32_I3C_MAXWLR）

偏移地址：0x94

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
MWL[15:0]															

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:0]	MWL[15:0]	RW	最大数据写入长度： 该位域最初由软件在 EN = 0 时写入，在接收到 SETMWL 命令时由硬件更新。 通过 R32_I3C_EVR 寄存器 MWLUPF 和相应的中断（如果使能）向软件通知 MWL 更新。 当从设备接收到 GETMWL CCC 时，硬件使用该位域在 I3C 总线上返回值。 注：此位仅适用于从设备。	0

22.4.21 I3C 时序寄存器 0（R32_I3C_TIMINGR0）

偏移地址：0xA0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCLH_I2C[7:0]								SCLL_OD[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH_I3C[7:0]								SCLL_PP[7:0]							

位	名称	访问	描述	复位值
[31:24]	SCLH_I2C[7:0]	RW	SCL 高电平持续时间, 用于传统 I2C 消息, 以内核时钟周期数为单位: $t_{SCL\_I2C} = (SCLH\_I2C + 1) \times t_{HCLK}$	0
[23:16]	SCLL_OD[7:0]	RW	开漏阶段的 SCL 低电平持续时间, 用于传统 I2C 消息和 I3C 开漏阶段 (起始位后的地址阶段、主设备发起的消息期间的 ACK 阶段以及直接/私有/IBI 有效负载期间的 T 位阶段), 以内核时钟周期数为单位: $t_{SCLL\_OD} = (SCLL\_OD + 1) \times t_{HCLK}$	
[15:8]	SCLH_I3C[7:0]	RW	SCL 高电平持续时间, 用于 I3C 消息 (在推挽和开漏阶段), 以内核时钟周期数为单位: $t_{SCLH\_I3C} = (SCLH\_I3C + 1) \times t_{HCLK}$	
[7:0]	SCLL_PP[7:0]	RW	I3C 推挽阶段的 SCL 低电平持续时间, 以内核时钟周期数为单位: $t_{SCLL\_PP} = (SCLL\_PP + 1) \times t_{HCLK}$	

22.4.22 I3C 时序寄存器 1 (R32_I3C_TIMINGR1)

偏移地址: 0xA4

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

位	名称	访问	描述	复位值
[31:29]	Reserved	R0	保留。	0
28	SDA_HD	RW	SDA 保持时间, 以内核时钟周期数为单位。 SDA 保持时间 = $(SDA\_HD + 0.5) \times t_{HCLK}$	0
[27:23]	Reserved	R0	保留。	0
[22:16]	FREE[6:0]	RW	用于设置某些 MIPI 时序 (例如总线空闲条件时间) 的内核时钟周期数。 注: 此位仅适用于主设备。	0
[15:10]	Reserved	R0	保留。	0
[9:8]	ASNCR[1:0]	RW	新主设备的工作状态: 在作为新从设备进行访问之前需要等待的时间, 请参见 AVAL[7:0]。 注: 此位仅适用于主设备。	0
[7:0]	AVAL[7:0]	RW	用于设置 1 μs 时间单位的内核时钟周期数:	0

		硬件随后使用该时间单位构建一些内部定时器，对应于以下 MIPI I3C 时序： 当 I3C 用作从设备时： 1. 对于总线可用条件时间：在停止位之后和发出 IBI 启动请求或主设备角色请求之前，必须等待一段（总线可用条件）时间（总线空闲条件至少持续 t_{AVAL}）。请参见 MIPI 时序 $t_{AVAL} = 1 \mu s$。该时序定义为：$t_{AVAL} = (AVAL[7:0] + 2) \times t_{HCLK}$ 2. 对于总线空闲条件时间：在 SDA 和 SCL 都为高电平且保持稳定之后和发出热加入事件之前，必须等待一段（总线空闲条件）时间。请参见 MIPI v1.1 时序 $t_{IDLE} = 200 \mu s$。该时序定义为：$t_{IDLE} = (AVAL[7:0] + 2) \times 200 \times t_{HCLK}$ 当 I3C 用作主设备时，其停顿时钟的时间不得超过最大停顿时间（将 SCL 时钟停顿为低电平），具体如下： 1. 对于动态地址分配期间分配的地址的第一位：停顿时钟的时间不得超过 MIPI 时序 $t_{STALLDAA} = 15ms$。该时序定义为：$t_{STALLDAAmax} = (AVAL[7:0] + 1) \times 15000 \times t_{HCLK}$ 2. 对于 I3C/I2C 传输的 ACK/NACK 阶段、写数据传输的奇偶校验位、I3C 读传输的转换位：停顿时钟的时间不得超过 MIPI 时序 $t_{STALL} = 100 \mu s$。该时序定义为：$t_{STALLmax} = (AVAL[7:0] + 1) \times 100 \times t_{HCLK}$ 无论 I3C 用作主设备还是（支持主设备角色的）从设备，在主设备角色切换过程中： 1. 新主设备必须等待 $t_{NEWORLock}$，然后才能在完成 GETACCR CCC 后将 SDA 拉低（发出起始位）。然后，新主设备可以在 t_{CAS} 内将 SCL 拉低以激活 SCL 时钟。在测试新主设备是否已通过拉低 SDA 获得总线控制权之前，活动主设备必须等待相同的 $t_{NEWORLock}$ 时间，或至少 $100 \mu s$。等待的时间取决于 I3C_TIMINGR1 寄存器中 ANSCR[1:0] 的值： - ANSCR[1:0] = 00: $t_{NEWORLock} = (AVAL[7:0] + 1) \times t_{HCLK}$ - ANSCR[1:0] = 01: $t_{NEWORLock} = (AVAL[7:0] + 1) \times 100 \times t_{HCLK}$ - ANSCR[1:0] = 10: $t_{NEWORLock} = (AVAL[7:0] + 1) \times 2000 \times t_{HCLK}$ - ANSCR[1:0] = 11: $t_{NEWORLock} = (AVAL[7:0] + 1) \times 50000 \times t_{HCLK}$	

22.4.23 I3C 时序寄存器 2 (R32_I3C_TIMINGR2)

偏移地址：0xA8

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
STALL[7:0]								Reserved				STA LLA	STA LLC	STA LLD	STA LLT

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:8]	STALL[7:0]	RW	主设备时钟停顿时间，以内核时钟周期数为单位： $t_{SOLL\_STALL} = STALL \times t_{HCLK}$	0
[7:4]	Reserved	R0	保留。	0
3	STALLA	RW	ACK 阶段的主设备时钟停顿使能： 1：使能停顿； 0：禁止停顿。	0
2	STALLC	RW	CCC PAR 阶段的主设备时钟停顿使能： 1：使能停顿； 0：禁止停顿。	0
1	STALLD	RW	数据 PAR 阶段的主设备时钟停顿使能： 1：使能停顿； 0：禁止停顿。	0
0	STALLT	RW	数据 T 位阶段（和传统 I2C 读消息的数据字节的 ACK/NACK 阶段）的主设备时钟停顿使能： 1：使能停顿； 0：禁止停顿。	0

22.4.24 I3C 总线特征寄存器（R32_I3C_BCR）

偏移地址：0xC0

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									BCR 6	Reserved			BCR 2	Res erv ed	BCR 0

位	名称	访问	描述	复位值
[31:7]	Reserved	R0	保留。	0
6	BCR6	RW	支持主设备角色： 1：支持 I3C 主设备角色； 0：I3C 从设备（不支持主设备角色）。	0
[5:3]	Reserved	R0	保留。	0
2	BCR2	RW	带内中断（IBI）有效负载： 1：接受的 IBI 后至少有 1 个强制数据字节（最多 4 个数据字节）； 0：接受的 IBI 后没有数据字节。	0

1	Reserved	R0	保留。	0
0	BCR0	RW	最大数据速度限制： 1：限制； 0：无限制。	0

22.4.25 I3C 设备特征寄存器 (R32_I3C_DCR)

偏移地址：0xC4

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								DCR[7:0]							

位	名称	访问	描述	复位值
[31:8]	Reserved	R0	保留。	0
[7:0]	DCR[7:0]	RW	设备特性 ID： 0x00：通用设备（适用于 v1.0 设备）； 其他值：描述 I3C 传感器/设备类型的 ID。	0

22.4.26 I3C 获取功能寄存器 (R32_I3C_GETCAPR)

偏移地址：0xC8

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	CAPPEND	Reserved													

位	名称	访问	描述	复位值
[31:15]	Reserved	R0	保留。	0
14	CAPPEND	RW	IBI MDB 支持挂起读通知： 1：用作从设备时，I3C 将发送带有指示挂起读通知的强制数据字节值 (MDB[7:5]=101) 的 IBI 请求； 0：用作从设备时，I3C 将发送不带有指示挂起读通知的强制数据字节值的 IBI 请求。	0
[13:0]	Reserved	R0	保留。	0

22.4.27 I3C 主设备角色功能寄存器 (R32_I3C_CRCAPR)

偏移地址：0xCC

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	CAP GRP	Reserved	CAP DHO FF	Reserved

位	名称	访问	描述	复位值
[31:10]	Reserved	R0	保留。	0
9	CAPGRP	RW	组管理支持： 1：支持组地址功能（成为主设备后）； 0：不支持组地址功能。 注：此位仅适用于主设备。	0
[8:4]	Reserved	R0	保留。	0
3	CAPDHOFF	RW	延迟主设备角色切换： 1：需要额外的时间来处理主设备角色切换； 0：不需要额外的时间来处理主设备角色切换。	0
[2:0]	Reserved	R0	保留。	0

22. 4. 28 I3C 获取最大数据速度寄存器（R32_I3C_GETMSR）

偏移地址：0xD0

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

位	名称	访问	描述	复位值
[31:25]	Reserved	R0	保留。	0
24	TSC0	RW	时钟到数据转向时间 (t_{SCO})： 1: $t_{SCO} > 12ns$; 0: $t_{SCO} \leq 12ns$ 。	0
[23:16]	RDTURN[7:0]	RW	写入 3 字节 MaxRdTurn 的所选字节的值（通过 FMT[1:0] 位域），返回该值可对最大读取切换转向时间进行编码以响应 GETMXDS CCC。	0
[15:10]	Reserved	R0	保留。	0
[9:8]	FMT[1:0]	RW	GETMXDS CCC 格式： 00：格式 1（2 字节：MaxWr（无释义字节）、MaxRd）； 01：格式 2：（5 字节：MaxWr（无释义字节）、MaxRd、MaxRdTurn）； -返回 3 字节 MaxRdTurn，其中 MSB=0，中间字节=0，LSB=RDTURN[7:0]； - 最大读取转向时间小于 $256 \mu s$ 。 10：格式 2（5 字节：MaxWr（无释义字节）、MaxRd 和 MaxRdTurn 的中间字节）； -返回 3 字节 MaxRdTurn，其中 MSB=0，中间字节	0

			=RDTURN[7:0], LSB=0; - 最大读取转向时间介于 256 到 65535 μ s 之间。 11: 格式 2 (5 字节: MaxWr (无释义字节)、MaxRd、MaxRdTurn 的 MSB) ; -返回 3 字节 MaxRdTurn, 其中 MSB=RDTURN[7:0], 中间字节=0, LSB=0; -最大读取转向时间介于 65535 μ s 到 16s 之间。	
[7:2]	Reserved	R0	保留。	0
[1:0]	HOFFAS[1:0]	RW	主设备切换工作状态: 00: 工作状态 0 是 I3C 在成为主设备之前和成为主设备时的初始工作状态; 01: 工作状态 1 是 I3C 成为主设备时的初始工作状态; 10: 工作状态 2 是 I3C 成为主设备时的初始工作状态; 11: 工作状态 3 是 I3C 成为主设备时的初始工作状态。	0

22.4.29 I3C 扩展配置 ID 寄存器 (R32_I3C_EPIDR)

偏移地址: 0xD4

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

位	名称	访问	描述	复位值
[31:17]	MIPI MID[14:0]	R0	15 位 MIPI 制造商 ID。	0x0104
16	IDTSEL	R0	配置 ID 类型选择器: 该位域设置为 0。 该位域代表 48 位配置 ID 的位[32]。 注: 配置 ID 的位[31:16]可以为 0。	0
[15:12]	MIPI ID[3:0]	RW	4 位 MIPI 实例 ID: 该位域由软件写入, 用于在单个 I3C 总线上通过特定编号单独设置和标识该 I3C IP 的每个实例。 该位域代表 48 位配置 ID 的位[15:12]。 注: 配置 ID 的位[11:0]可以为 0。	0
[11:0]	Reserved	R0	保留。	0

第 23 章 串行外设接口（SPI/I2S）

芯片内置 4 组串行外设 SPI 接口（SPI1/2/3/4），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 也是三线的同步串行接口通信协议，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。

23.1 主要特征

23.1.1 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 F_{HCLK} 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

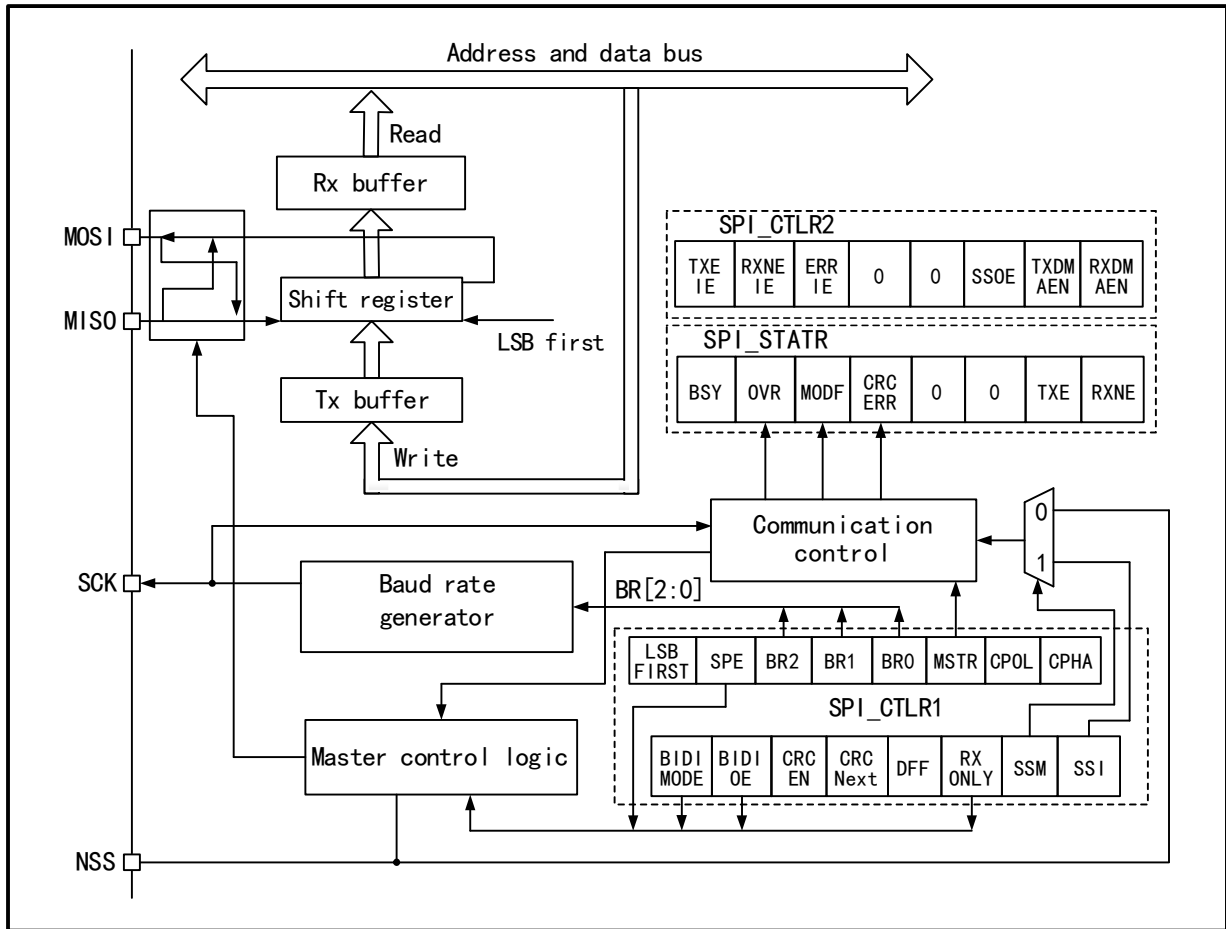
23.1.2 I2S 特征

- 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~562.2kHz
- 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、LSB 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

23.2 SPI 功能描述

23.2.1 概述

图 23-1 SPI 结构框图

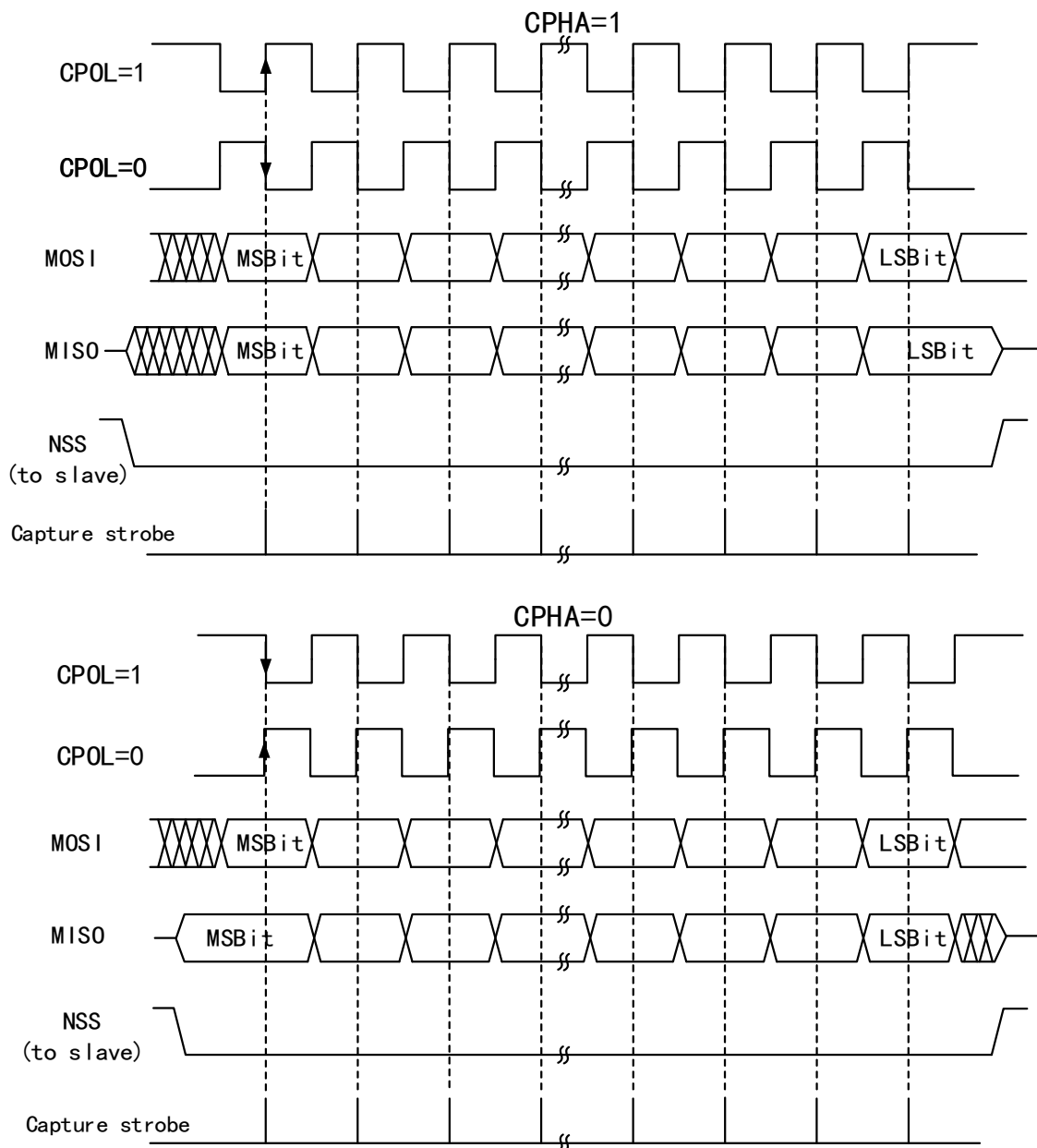


由图 23-1 可以看出，与 SPI 相关的主要是 MISO、MOSI、SCK 和 NSS 四个引脚。其中 MISO 引脚在 SPI 模块工作在主模式下时，是数据输入引脚；工作在从模式下时，是数据输出引脚。MOSI 引脚工作在主模式下时，是数据输出引脚；工作在从模式时，是数据输入引脚。SCK 是时钟引脚，时钟信号一直由主机输出，从机接收时钟信号并同步数据收发。NSS 引脚是片选引脚，有以下用法：

- 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，如果不能成功拉低 NSS 脚，说明主线上还有其他主设备正在通信，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 23-2。

图 23-2 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。

23.2.2 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 SS0E 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。

23.2.3 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。

23.2.4 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。也可以不理睬接收的数据将 SPI 置成只发送的模式。

23.2.5 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。

23.2.6 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。

23.2.7 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。

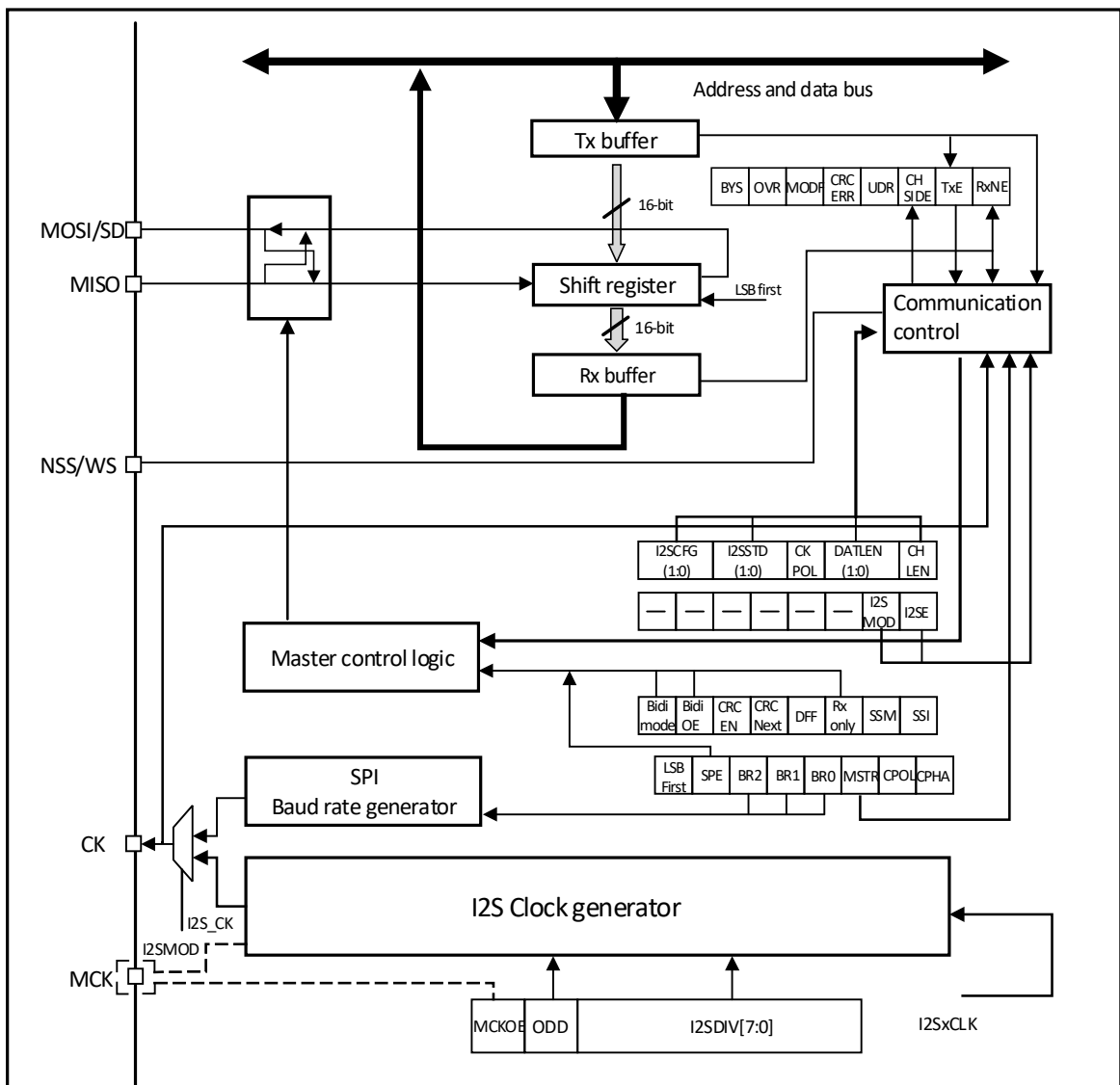
23.2.8 中断

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。

23.3 I2S 功能描述

23.3.1 I2S 概述

图 23-3 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 $256 \times F_s$ ，其中 F_s 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、时钟极性参数等）。在 I2S 模式下不使用寄存器 CTLR1 和所有的 CROR 寄存器。同样，I2S 模式下也不使用寄

寄存器 CTLR2 的 SS0E 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。

23.3.2 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）是有意义的数字，后 16 位（LSB）被强制为 0，该操作不需要软件干预，也不需要 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。

23.3.2.1 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 23-4 飞利浦协议波形（16/32 全精度，CPOL=0）

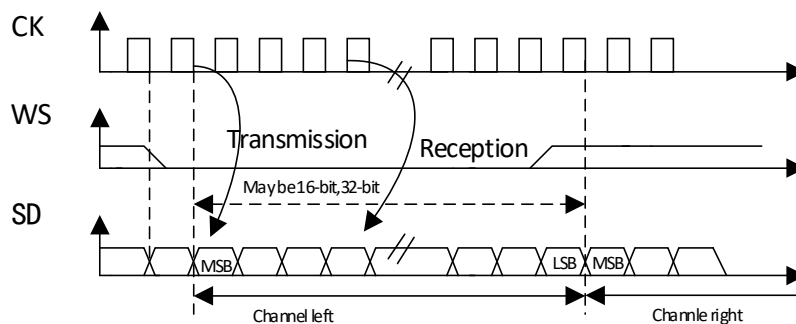
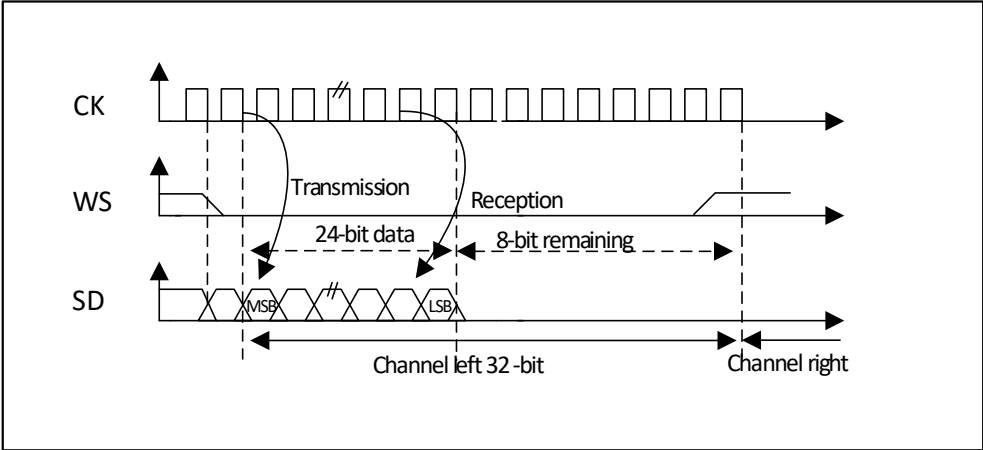
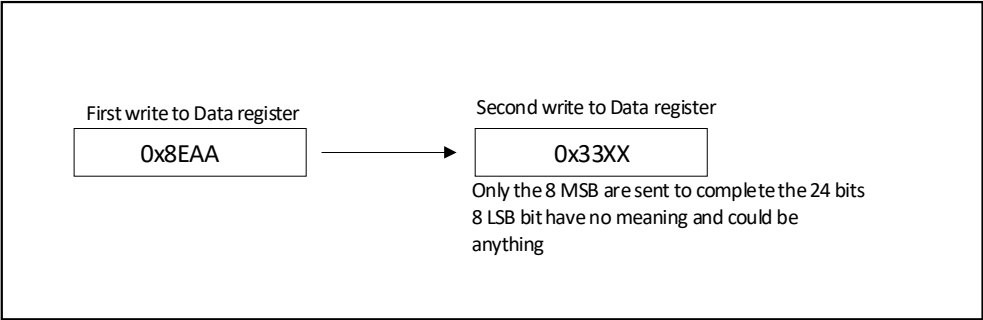


图 23-5 飞利浦协议波形（24 位帧，CPOL=0）



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：

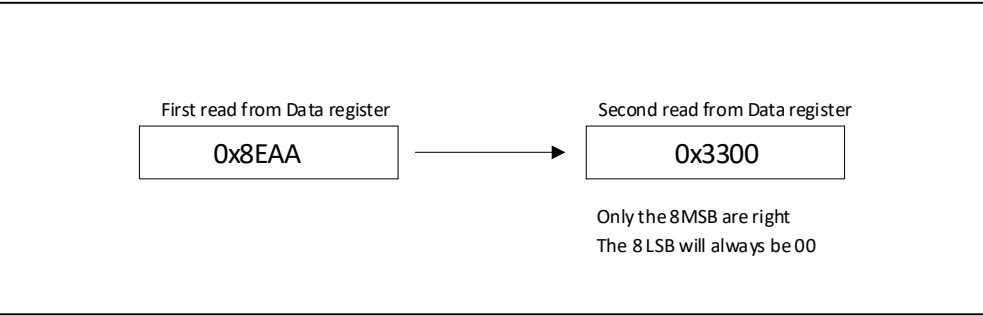
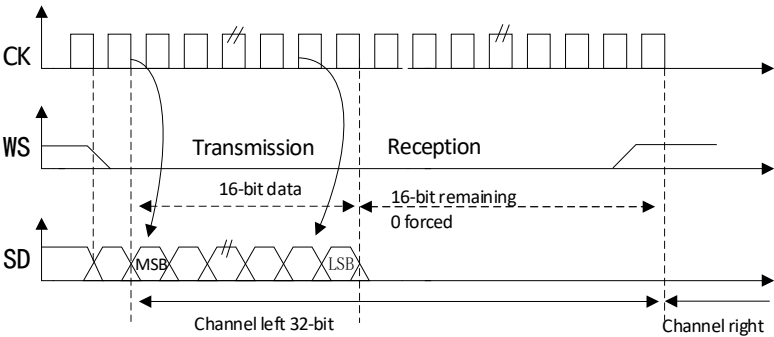


图 23-6 飞利浦协议标准波形（16 位扩展至 32 位包帧，CPOL=0）



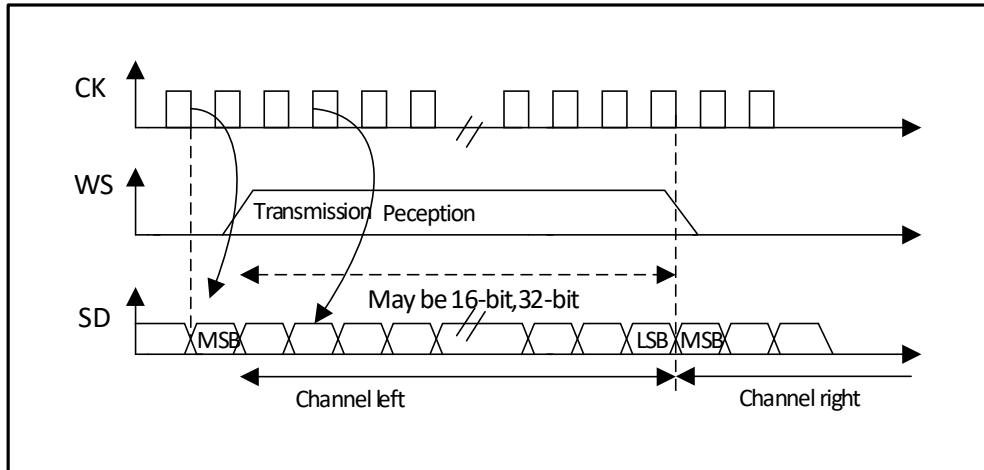
在 I2S 配置阶段，如果选择将 16 位数据扩展到 32 位声道帧，只需要访问一次寄存器 DATAR 用来扩展到 32 位的低 16 位被硬件置为 0x0000。如果待传输或接收的数据是 0x76A3（扩展到 32 位是 0x76A30000），只需要操作一次 DATAR。在发送时需要将 MSB 写入寄存器 DATAR；标志位 TXE 为 1 表

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。

23.3.2.2 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 23-7 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 23-8 MSB 对齐 24 位数据，CPOL = 0

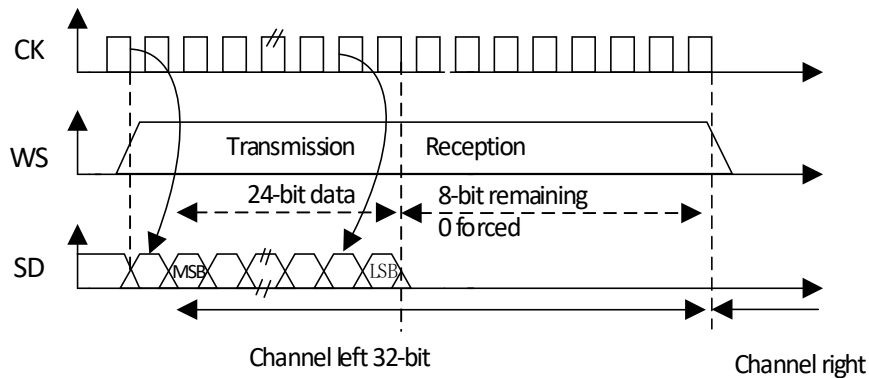
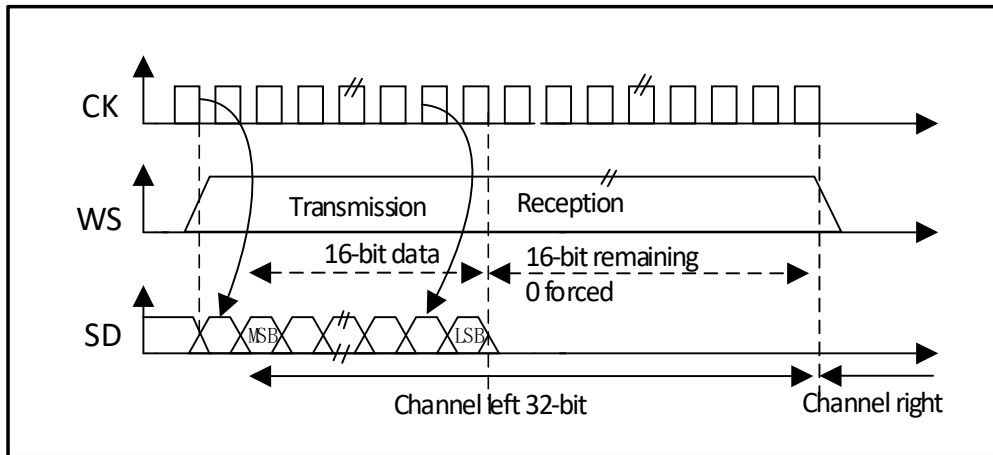


图 23-9 MSB 对齐 16 位数据扩展到 32 位包帧, CPOL = 0



23.3.2.3 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 23-10 LSB 对齐 16 或 32 位全精度, CPOL = 0

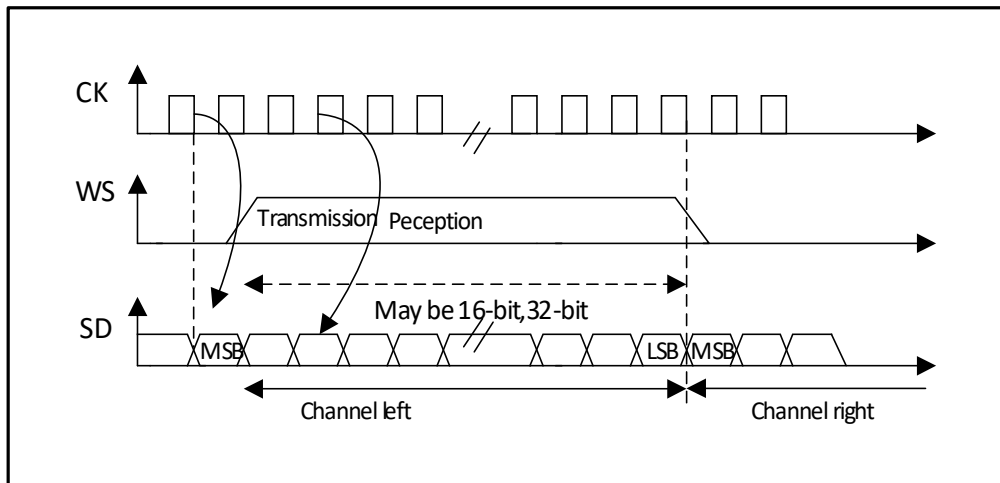
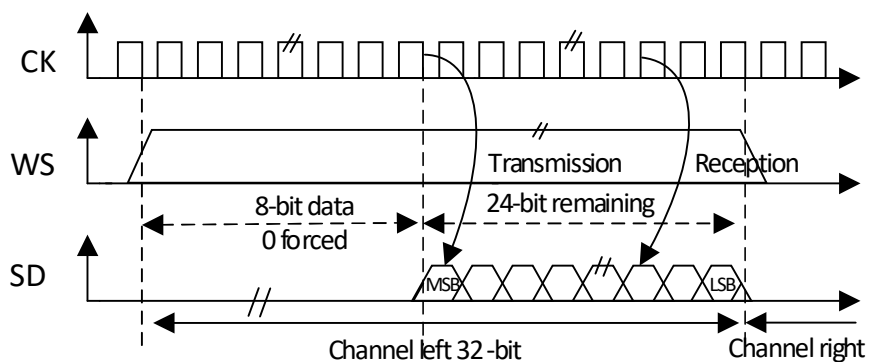
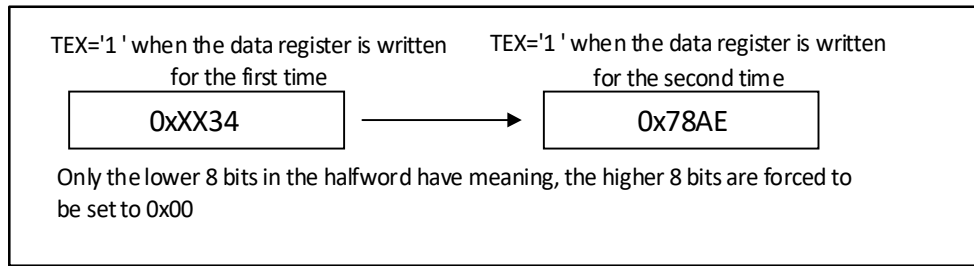


图 23-11 LSB 对齐 24 位数据, CPOL = 0



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。

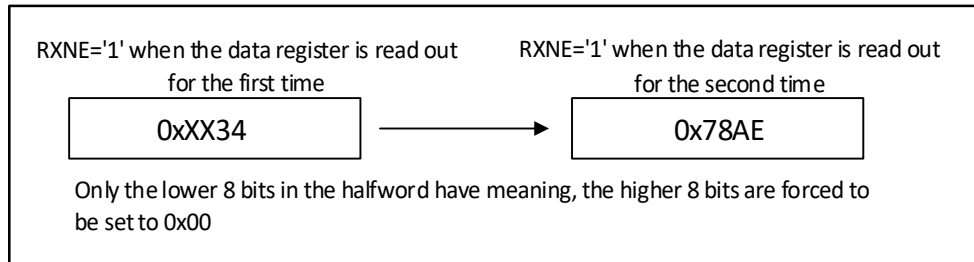
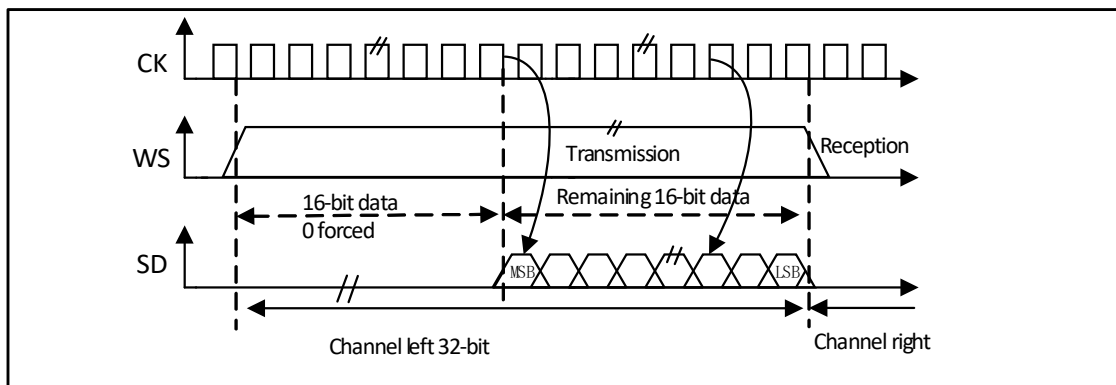


图 23-12 LSB 对齐 16 位数据扩展到 32 位包帧，CPOL = 0



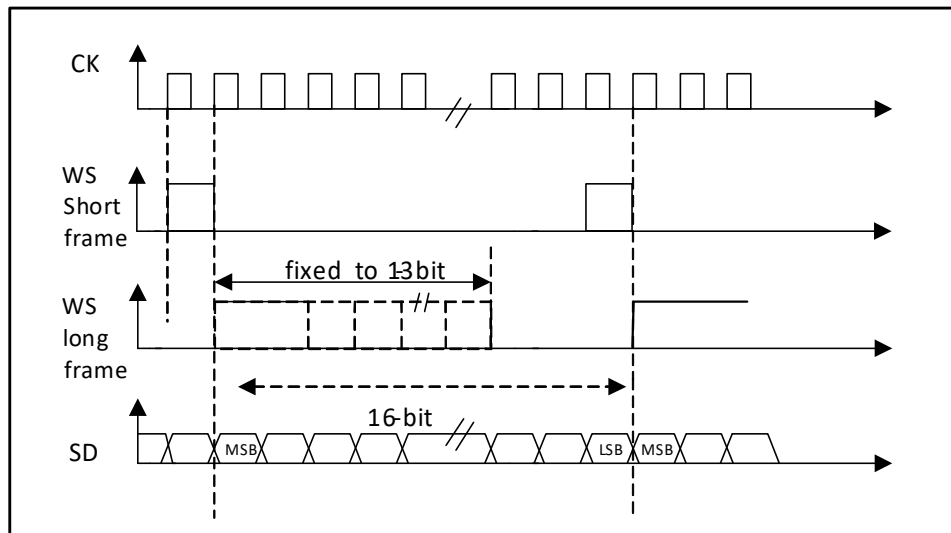
在 I2S 配置阶段，如果选择将 16 位数据扩展到 32 声道帧，只需要访问一次寄存器 DATAR。此时，扩展到 32 位后的高半字（16 位 MSB）被硬件置为 0x0000。

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。

23.3.2.4 PCM 标准

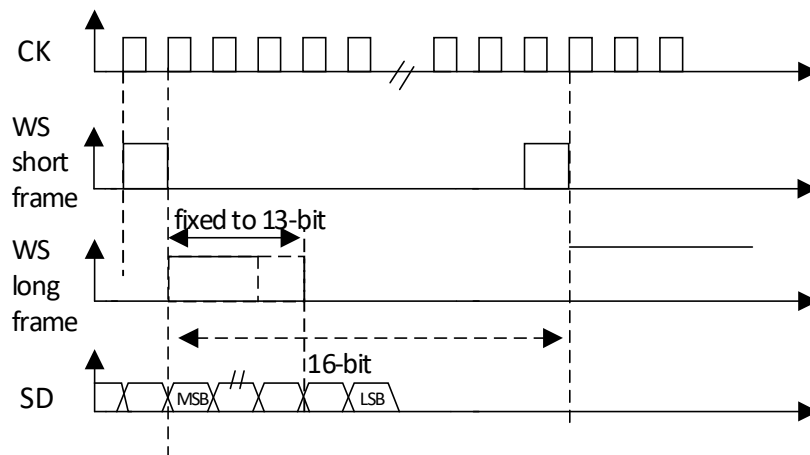
在 PCM 标准下，不存在声道选择的信息。PCM 标准有 2 种可用的帧结构，短帧或长帧，可以通过设置寄存器 I2SCFGR 的 PCMSYNC 位来选择。

图 23-13 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 23-14 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。

23.3.3 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。

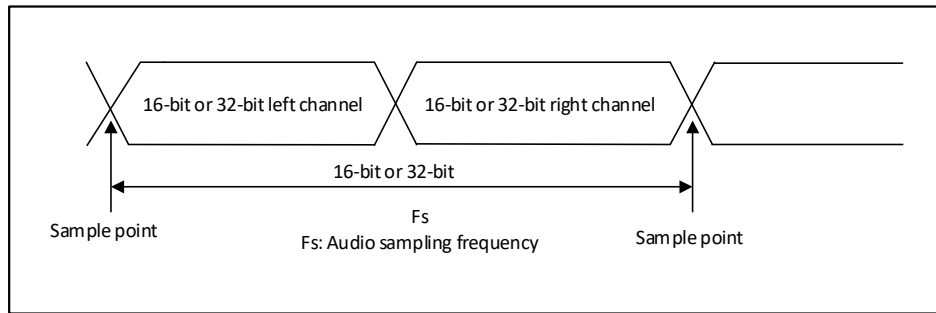
对于一个具有左右声道和 16 位音频信号，I2S 比特率计算如下：

$$\text{I2S 比特率} = 16 \times 2 \times F_s$$

如果包长为 32 位，I2S 比特率计算如下：

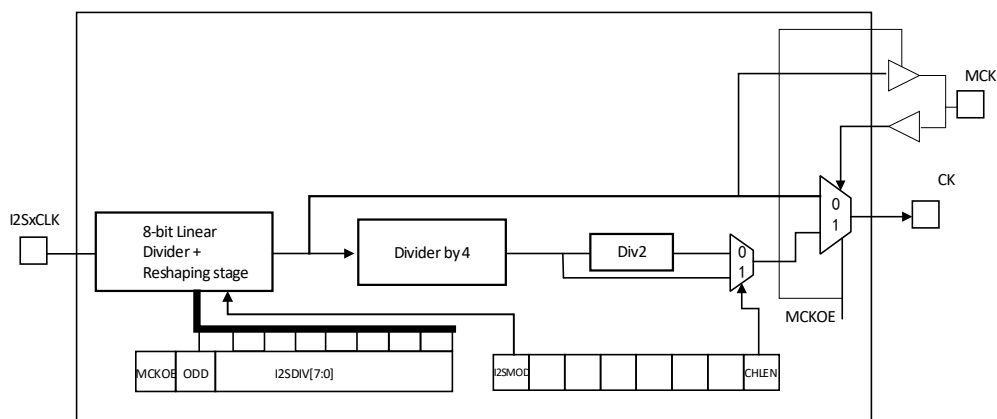
$$\text{I2S 比特率} = 32 \times 2 \times F_s$$

图 23-15 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 23-16 I2S 时钟发生器结构



图中 I2SxCLK 的时钟源是系统时钟（即驱动 HB 时钟的 HSI、HSE 或 PLL）。I2SxCLK 可以来自 SYSCCLK，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， $F_s = I2SxCLK / [(16 \times 2) * ((2 * I2SDIV) + ODD) * 8]$

声道的帧长为 32 位时， $F_s = I2SxCLK / [(32 \times 2) * ((2 * I2SDIV) + ODD) * 4]$

当关闭主时钟时（MCKOE 位为 '0'）：

声道的帧长为 16 位时， $F_s = I2SxCLK / [(16 \times 2) * ((2 * I2SDIV) + ODD)]$

声道的帧长为 32 位时， $F_s = I2SxCLK / [(32 \times 2) * ((2 * I2SDIV) + ODD)]$

23.3.4 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）。

23.3.4.1 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。

23.3.4.2 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前的顺序从引脚 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。

23.3.4.3 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。

23.3.5 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 interrupt 功能和 DMA 功能。
- 必须设置寄存器 I2SCFGR 的 I2SE 位为 1。

23.3.5.1 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。

23.3.5.2 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。

23.3.6 状态标志位

有 3 个状态标志位供用户监控 I2S 总线的状态。

23.3.6.1 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。

23.3.6.2 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。

23.3.6.3 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。

23.3.6.4 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。

23.3.7 错误标志位

23.3.7.1 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。

23.3.7.2 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。

23.3.8 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。

23.3.9 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。

23.4 寄存器描述

表 23-1 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_CRCCR	0x40013010	SPI1 多项式寄存器	0x0007
R16_SPI1_RCRCCR	0x40013014	SPI1 接收 CRC 寄存器	0x0000
R16_SPI1_TCRCCR	0x40013018	SPI1 发送 CRC 寄存器	0x0000
R16_SPI1_HSCR	0x40013024	SPI1 高速控制寄存器	0x0000

表 23-2 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_I2S_CFGR	0x4000381C	SPI2_I2S 配置寄存器	0x0000
R16_SPI2_I2SPR	0x40003820	SPI2_I2S 预分频寄存器	0x0000
R16_SPI2_HSCR	0x40003824	SPI2 高速控制寄存器	0x0000

表 23-3 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_I2S_CFGR	0x40003C1C	SPI3_I2S 配置寄存器	0x0000
R16_SPI3_I2SPR	0x40003C20	SPI3_I2S 预分频寄存器	0x0000
R16_SPI3_HSCR	0x40003C24	SPI3 高速控制寄存器	0x0000

表 23-4 SPI4 相关寄存器列表

名称	访问地址	描述	复位值
R16_SPI4_CTLR1	0x40004000	SPI4 控制寄存器 1	0x0000
R16_SPI4_CTLR2	0x40004004	SPI4 控制寄存器 2	0x0000
R16_SPI4_STATR	0x40004008	SPI4 状态寄存器	0x0002
R16_SPI4_DATAR	0x4000400C	SPI4 数据寄存器	0x0000
R16_SPI4_CRCR	0x40004010	SPI4 多项式寄存器	0x0007
R16_SPI4_RCRCR	0x40004014	SPI4 接收 CRC 寄存器	0x0000
R16_SPI4_TCRCR	0x40004018	SPI4 发送 CRC 寄存器	0x0000
R16_SPI4_HSCR	0x40004024	SPI4 高速控制寄存器	0x0000

23.4.1 SPI 控制寄存器 1 (SPIx_CTLR1) (x=1/2/3/4)

偏移地址：0x00

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。	0
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[2:0]	RW	波特率设置域，在通讯时不可以修改此域。 当 HSRXEN 位为 0 时，SCK 频率为： 000: $F_{HCLK}/2$ ； 001: $F_{HCLK}/4$ ； 010: $F_{HCLK}/8$ ； 011: $F_{HCLK}/16$ ； 100: $F_{HCLK}/32$ ； 101: $F_{HCLK}/64$ ； 110: $F_{HCLK}/128$ ； 111: $F_{HCLK}/256$ 。 当 HSRXEN 位为 1 时，SCK 频率为： 000: $F_{HCLK}/2$ ； 001: $F_{HCLK}/3$ ； 010: $F_{HCLK}/4$ ； 011: $F_{HCLK}/5$ ； 100: $F_{HCLK}/6$ ； 101: $F_{HCLK}/7$ ； 110: $F_{HCLK}/8$ ； 111: $F_{HCLK}/9$ 。	000b
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

			0: 配置为从设备。	
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

23.4.2 SPI 控制寄存器 2 (SPIx_CTLR2) (x=1/2/3/4)

偏移地址: 0x04

位	名称	访问	描述	复位值
[15:8]	Reserved	R0	保留。	0
7	TXEIE	RW	发送缓冲区空中断使能位: 1: 允许 TXE 被置位时产生中断; 0: 禁止 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: 1: 允许 RXNE 被置位时产生中断; 0: 禁止 RXNE 被置位时产生中断。	0
5	ERRIE	RW	错误中断使能位: 1: 允许在产生错误 (CRCERR, OVR, MODF) 时产生中断; 0: 禁止在产生错误 (CRCERR, OVR, MODF) 时产生中断。	0
[4:3]	Reserved	R0	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

23.4.3 SPI 状态寄存器 (SPIx_STATR) (x=1/2/3/4)

偏移地址: 0x08

位	名称	访问	描述	复位值
[15:8]	Reserved	R0	保留。	0
7	BSY	R0	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	R0	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	CHSIDE	RO	声道，该位由硬件置位，软件复位。 1：需要传输或接收右声道； 0：需要传输或接收左声道。	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。 注：读 DATAR，自动清零。	0

23.4.4 SPI 数据寄存器 (SPIx_DATAR) (x=1/2/3/4)

偏移地址：0x0C

位	名称	访问	描述	复位值
[15:0]	DR[15:0]	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，其中读对应接收缓冲区，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

23.4.5 SPI 多项式寄存器 (SPIx_CRCCR) (x=1/2/3/4)

偏移地址：0x10

位	名称	访问	描述	复位值
[15:0]	CRCPOLY[15:0]	RW	CRC 多项式。此域定义 CRC 计算用到的多项式。	0x0007

23.4.6 SPI 接收 CRC 寄存器 (SPIx_RCRCR) (x=1/2/3/4)

偏移地址：0x14

位	名称	访问	描述	复位值
[15:0]	RXCRC[15:0]	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 置位会复位该寄存器。计算方法使用 CRCPOLY 用到的多项式。	0

			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。	

23.4.7 发送 CRC 寄存器 (SPIx_TCRCR) (x=1/2/3/4)

偏移地址：0x18

位	名称	访问	描述	复位值
[15:0]	TXCRC[15:0]	RO	发送 CRC 值。存储着计算出来的已经发送出去的字节 CRC 校验的结果。对 CRGEN 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

23.4.8 SPI_I2S 配置寄存器 (SPIx_I2S_CFGR) (x=2/3)

偏移地址：0x1C

位	名称	访问	描述	复位值
[15:12]	Reserved	RO	保留。	0
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。	0
10	I2SE	RW	I2S 使能, 在 SPI 模式下不使用。 1: I2S 使能; 0: 关闭 I2S。	0
[9:8]	I2SCFG[1:0]	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。	0
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。	0
6	Reserved	RO	保留。	0
[5:4]	I2SSTD[1:0]	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。	0
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

			0: I2S 时钟静止态为低电平。	
[2:1]	DATLEN[1:0]	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
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。	0

23.4.9 SPI_I2S 预分频寄存器 (SPIx_I2SPR) (x=2/3)

偏移地址: 0x20

位	名称	访问	描述	复位值
[15:10]	Reserved	RO	保留。	0
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。	0
8	ODD	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。	0
[7:0]	I2SDIV[7:0]	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。	0

23.4.10 SPI 高速控制寄存器 (SPIx_HSCR) (x=1/2/3/4)

偏移地址: 0x24

位	名称	访问	描述	复位值
[15:3]	Reserved	RO	保留。	0
2	HSRXEN2	RW	SPI 高速读模式 2: 该模式仅在开启 HSRXEN 且 SPI_HCLK 大于等于 120M 时使用, 小于 120M 不建议使用此位。可以获得比仅开启 HSRXEN 更高的主机读速度。 1: 使能高速读模式 2; 0: 关闭高速读模式 2。	0
1	Reserved	RO	保留。	0
0	HSRXEN	WO	高速读模式使能位: 1: 使能高速读模式;	0

		0: 关闭高速读模式。	

第 24 章 USB PD 控制器（USBPD）

24.1 USB PD 控制器简介

芯片内置 USB Power Delivery 控制器和 PD 物理层收发器 PHY，支持 USB Type-C 主从检测，自动 BMC 编解码和 CRC，CC 引脚支持硬件边沿控制。

支持 USB PD2.0 和 PD3.0 以及 PD3.2 电力传送控制，支持 SPR 和 EPR，支持 100W 或 240W 快充，支持 PD 受电端 Sink 和 PD 供电端 Source 以及 DRP 应用。支持 PDUSB，支持 UFP 和 DFP 以及 DRD 应用。

支持 SOP、SOP'、SOP'' 等 PD 包，支持 USB PD 复位信号帧硬件复位，支持最大包长度 510 字节，支持 DMA。

24.2 寄存器描述

表 24-1 USBPD 相关寄存器列表

名称	访问地址	描述	复位值
R32_USBPD_CONFIG	0x40024400	PD 配置寄存器	0x00000002
R16_CONFIG	0x40024400	PD 中断使能寄存器	0x0002
R16_BMC_CLK_CNT	0x40024402	BMC 采样时钟计数器	0x0000
R32_USBPD_CONTROL	0x40024404	PD 控制寄存器	0x00000000
R16_CONTROL	0x40024404	PD 收发控制寄存器	0x0000
R8_CONTROL	0x40024404	PD 收发使能寄存器	0x00
R8_TX_SEL	0x40024405	PD 发送 SOP 选择寄存器	0x00
R16_BMC_TX_SZ	0x40024406	PD 发送长度寄存器	0x0000
R32_USBPD_STATUS	0x40024408	PD 状态寄存器	0x000000XX
R16_STATUS	0x40024408	PD 中断和数据寄存器	0x00XX
R8_DATA_BUF	0x40024408	DMA 缓存数据寄存器	0xXX
R8_STATUS	0x40024409	PD 中断标志寄存器	0x00
R16_BMC_BYTE_CNT	0x4002440A	字节计数器	0x0000
R32_USBPD_PORT	0x4002440C	端口控制寄存器	0x00030003
R16_PORT_CC1	0x4002440C	CC1 端口控制寄存器	0x0003
R16_PORT_CC2	0x4002440E	CC2 端口控制寄存器	0x0003
R32_USBPD_DMA	0x40024410	DMA 缓存地址寄存器	0x00XXXXXX

24.2.1 PD 配置寄存器（R32_USBPD_CONFIG）

偏移地址：0x00

位	名称
[31:16]	R16_BMC_CLK_CNT
[15:0]	R16_CONFIG

24.2.2 PD 中断使能寄存器（R16_CONFIG）

偏移地址：0x00

位	名称	访问	描述	复位值
15	IE_TX_END	RW	发送结束中断使能。	0
14	IE_RX_RESET	RW	接收复位中断使能。	0

13	IE_RX_ACT	RW	接收完成中断使能。	0
12	IE_RX_BYTE	RW	接收字节中断使能。	0
11	IE_RX_BIT	RW	接收 bit 中断使能。	0
10	IE_PD_IO	RW	PD IO 中断使能。	0
9	Reserved	RO	保留。	0
8	RX_MULT1_0	RO	连续接收多位 0 指示位： 1：接收到连续多位 0； 0：未接收到连续多位 0。	0
7	CC_INC	RW	CC 偏置电流调节位： 1：偏置电流+10%； 0：默认挡位。	0
6	CC_TR	RW	CC LV 波形的 Tr 调节： 1：Tr 提高 5%； 0：Tr 不变。	0
5	WAKE_POLAR	RW	PD 端口唤醒电平： 1：高电平有效； 0：低电平有效。	0
4	PD_RST_EN	RW	PD 模式复位命令使能： 1：复位； 0：无效。	0
3	PD_DMA_EN	RW	使能 USBPD 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
2	CC_SEL	RW	选择当前 PD 通讯端口： 1：使用 CC2 端口通讯； 0：使用 CC1 端口通讯。	0
1	PD_ALL_CLR	RW	PD 模式清除所有的中断标志位： 1：清除中断标志位； 0：无效。	1
0	PD_FILT_EN	RW	控制 PD 引脚的输入滤波使能： 1：打开滤波； 0：关闭滤波。	0

24.2.3 BMC 采样时钟计数器 (R16_BMC_CLK_CNT)

偏移地址：0x02

位	名称	访问	描述	复位值
[15:9]	Reserved	RO	保留。	0
[8:0]	BMC_CLK_CNT	RW	BMC 发送或者接收采样时钟计数器。	0

注：发送时， $BMC\_CLK\_CNT = (HCLK/CC \text{ 通信频率})/2$

接收时， $BMC\_CLK\_CNT = (HCLK/CC \text{ 通信频率})/4*3$

24.2.4 PD 控制寄存器 (R32_USBPD_CONTROL)

偏移地址：0x04

位	名称

[31:16]	R16_BMC_TX_SZ
[15:0]	R16_CONTROL

24.2.5 PD 收发控制寄存器 (R16_CONTROL)

偏移地址：0x04

位	名称
[15:8]	R8_TX_SEL
[7:0]	R8_CONTROL

24.2.6 PD 收发使能寄存器 (R8_CONTROL)

偏移地址：0x04

位	名称	访问	描述	复位值
7	BMC_BYTE_HI	RO	指示当前 PD 数据收发时的半字节状态： 1：表示正在处理高 4 位； 0：正在处理低 4 位。	0
6	TX_BIT_BACK	RO	指示当前 BMC 发送编码时的位状态： 1：表示正在发送 BMC 字节； 0：空闲。	0
5	DATA_FLAG	RO	缓存数据有效标志位。	0
[4:2]	RX_STATE[2:0]	RO	PD 接收状态标识 000：接收初始状态 001：开始接收 SOP 010：接收复位 011：接收 SOP 100：接收结束 101：接收未使用 110：接收 EOP 111：接收字节。	0
1	BMC_START	RW	BMC 发送开始信号。	0
0	PD_TX_EN	RW	USBPD 收发模式和发送使能： 1：PD 发送使能； 0：PD 接收使能。	0

24.2.7 PD 发送 SOP 选择寄存器 (R8_TX_SEL)

偏移地址：0x05

位	名称	访问	描述	复位值
[7:6]	TX_SEL4[1:0]	RW	PD 发送模式下 K-CODE4 类型选择： 00：SYNC2； 01：SYNC3； 1x：RST2。	0
[5:4]	TX_SEL3[1:0]	RW	PD 发送模式下 K-CODE3 类型选择： 00：SYNC1； 01：SYNC3； 1x：RST1。	0
[3:2]	TX_SEL2[1:0]	RW	PD 发送模式下 K-CODE2 类型选择： 00：SYNC1； 01：SYNC3； 1x：RST1。	0
1	Reserved	RO	保留。	0

0	TX_SEL1	RW	PD 发送模式下 K-CODE1 类型选择： 1: RST1; 0: SYNC1。	0

24.2.8 PD 发送长度寄存器 (R16_BMC_TX_SZ)

偏移地址: 0x06

位	名称	访问	描述	复位值
[15:9]	Reserved	R0	保留。	0
[8:0]	BMC_TX_SZ	RW	PD 模式下发送的总长度。	0

24.2.9 PD 状态寄存器 (R32_USBPDP_STATUS)

偏移地址: 0x08

位	名称
[31:16]	R16_BMC_BYTE_CNT
[15:0]	R16_STATUS

24.2.10 PD 中断和数据寄存器 (R16_STATUS)

偏移地址: 0x08

位	名称
[15:8]	R8_STATUS
[7:0]	R8_DATA_BUF

24.2.11 DMA 缓存数据寄存器 (R8_DATA_BUF)

偏移地址: 0x08

位	名称	访问	描述	复位值
[7:0]	DATA_BUF	R0	DMA 缓存数据。	X

24.2.12 PD 中断标志寄存器 (R8_STATUS)

偏移地址: 0x09

位	名称	访问	描述	复位值
7	IF_TX_END	RW1Z	传送完成中断标志, 写 1 清 0, 写 0 无效。	0
6	IF_RX_RESET	RW1Z	接收复位中断标志, 写 1 清 0, 写 0 无效。	0
5	IF_RX_ACT	RW1Z	接收完成中断标志, 写 1 清 0, 写 0 无效。	0
4	IF_RX_BYTE	RW1Z	接收字节或者 SOP 中断标志, 写 1 清 0, 写 0 无效。	0
3	IF_RX_BIT	RW1Z	接收 bit 或者 5bit 中断标志, 写 1 清 0, 写 0 无效。	0
2	BUF_ERR	RW1Z	BUFFER 或者 DMA 错误中断标志, 写 1 清 0, 写 0 无效。	0
[1:0]	BMC_AUX[1:0]	R0	指示当前 PD 状态: 在 PD 接收时或者接收完成后, 状态如下: 00: 接收空闲或者没有接收到有效的数据包; 01: 接收到 SOP 即 SOP0; 10: 接收到 SOP' 即 SOP1 或者 Hard Reset;	0

			11: 接收到 SOP” 即 SOP2 或者 Cable Reset。 在 PD 发送时, 状态如下: 00: 正在发送CRC32[7:0]; 01: 正在发送CRC32[15:8]; 10: 正在发送 CRC32[23:16]; 11: 正在发送 CRC32[31:24]。	

24.2.13 字节计数器 (R16_BMC_BYTE_CNT)

偏移地址: 0x0A

位	名称	访问	描述	复位值
[15:9]	Reserved	RO	保留。	0
[8:0]	BMC_BYTE_CNT	RO	字节计数器。	0

24.2.14 端口控制寄存器 (R32_USBDPD_PORT)

偏移地址: 0x0C

位	名称
[31:16]	R16_PORT_CC2
[15:0]	R16_PORT_CC1

24.2.15 CC1 端口控制寄存器 (R16_PORT_CC1)

偏移地址: 0x0C

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
[7:5]	CC1_GE	RW	CC1 端口电压比较器的使能: 000: 关闭; 001: 保留; 010: 0.22V; 011: 0.45V; 100: 0.55V; 101: 0.66V; 110: 0.95V; 111: 1.23V。	0
4	CC1_LVE	RW	CC1 端口输出低电压使能: 1: 低电压驱动输出; 0: 正常 V _{DD33} 电压驱动输出。	0
[3:2]	CC1_PU[1:0]	RW	CC1 端口上拉电流选择: 00: 禁止上拉电流; 01: 330uA; 10: 180uA; 11: 80uA。	0
1	CC1_PD	RW	CC1 端口下拉电阻 Rd 使能: 1: 启用 Rd 下拉电阻, 约 5.1K Ω ; 0: 关闭下拉电阻。 注: 关闭后仍有约 600k Ω 弱下拉。	1
0	PA_CC1_AI	RO	CC1 端口比较器模拟输入。	1

24.2.16 CC2 端口控制寄存器 (R16_PORT_CC2)

偏移地址: 0x0E

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
[7:5]	CC2_CE	RW	CC2 端口电压比较器的使能: 000: 关闭; 001: 保留; 010: 0.22V; 011: 0.45V; 100: 0.55V; 101: 0.66V; 110: 0.95V; 111: 1.23V。	0
4	CC2_LVE	RW	CC2 端口输出低电压使能: 1: 低电压驱动输出; 0: 正常 V_{DD33} 电压驱动输出。	0
[3:2]	CC2_PU	RW	CC2 端口上拉电流选择: 00: 禁止上拉电流; 01: 330uA; 10: 180uA; 11: 80uA。	0
1	CC2_PD	RW	CC2 端口下拉电阻 R_d 使能: 1: 启用 R_d 下拉电阻, 约 5.1K Ω ; 0: 关闭下拉电阻。 注: 关闭后仍有约 600k Ω 弱下拉。	1
0	PA_CC2_AI	RO	CC2 端口比较器模拟输入。	1

24.2.17 DMA 缓存地址寄存器 (R32_USBDMA_DMA)

偏移地址: 0x10

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:0]	USBDMA_DMA_ADDR	RW	USBDMA_DMA 缓存地址。 低 21 位有效, 地址必须 4 字节对齐。	X

第 25 章 USB 高速主机/设备控制器（USBHS）

USB2.0 高速控制器具有主机控制器和设备控制器双重角色，内置 480Mbps 的 USB-PHY 物理层收发器。当作为主机控制器时，它可支持低速、全速和高速的 USB 设备。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。

25.1 USB 功能描述

- 支持 USB 2.1、USB 2.0、USB 1.1、USB 1.0 协议规范
- 支持 USB Host 主机功能和 USB Device 设备功能
- 支持控制传输、批量传输、中断传输、实时/同步传输
- 提供总线复位、挂起、唤醒和恢复功能
- 主机支持 USB HUB
- 非 0 端点均支持最大 1024 字节数据包，内置 FIFO，支持中断和 DMA
- 支持 USART 串口或 I2C 引脚映射，兼用于两线调试

25.2 寄存器描述

USB 相关寄存器分为 2 个部分：

- USB 高速设备控制寄存器
- USB 高速主机控制寄存器

25.2.1 设备寄存器描述

表 25-1 设备相关寄存器列表

名称	访问地址	描述	复位值
R8_USB_CTRL	0x40030000	USBHS 控制寄存器	0x07
R8_USB_BASE_MODE	0x40030001	USBHS 模式控制寄存器	0x00
R8_USB_INT_EN	0x40030002	USBHS 中断使能寄存器	0x00
R8_USB_DEV_AD	0x40030003	USBHS 设备地址寄存器	0x00
R8_USB_WAKE_CTRL	0x40030004	USBHS 远程唤醒寄存器	0x00
R8_USB_TEST_MODE	0x40030005	USBHS 测试模式寄存器	0x00
R16_USB_LPM_DATA	0x40030006	USBHS 电源管理寄存器	0x8000
R8_USB_INT_FG	0x40030008	USBHS 中断标志寄存器	0x00
R8_USB_INT_ST	0x40030009	USBHS 中断状态寄存器	0x00
R8_USB_MIS_ST	0x4003000A	USBHS 杂项状态寄存器	0x00
R16_USB_FRAME_NO	0x4003000C	USBHS 帧号寄存器	0x0000
R16_USB_BUS	0x4003000E	USBHS 总线状态寄存器	0x0000
R16_UEP_TX_EN	0x40030010	USBHS 端点发送使能寄存器	0x0000
R16_UEP_RX_EN	0x40030012	USBHS 端点接收使能寄存器	0x0000
R16_UEP_T_TOG_AUTO	0x40030014	USBHS 端点发送自动翻转使能寄存器	0x0000
R16_UEP_R_TOG_AUTO	0x40030016	USBHS 端点接收自动翻转使能寄存器	0x0000
R8_UEP_T_BURST	0x40030018	USBHS 端点发送突发寄存器	0x00
R8_UEP_T_BURST_MODE	0x40030019	USBHS 端点发送突发模式寄存器	0x00
R8_UEP_R_BURST	0x4003001A	USBHS 端点接收突发寄存器	0x00
R8_UEP_R_RES_MODE	0x4003001B	USBHS 端点接收回复模式寄存器	0x00

R32_UEP_AF_MODE	0x4003001C	USBHS 端点复用寄存器	0x00000000
R32_UEP0_DMA	0x40030020	端点 0 缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP1_RX_DMA	0x40030024	端点 1 接收缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP2_RX_DMA	0x40030028	端点 2 接收缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP3_RX_DMA	0x4003002C	端点 3 接收缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP4_RX_DMA	0x40030030	端点 4 接收缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP5_RX_DMA	0x40030034	端点 5 接收缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP6_RX_DMA	0x40030038	端点 6 接收缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP7_RX_DMA	0x4003003C	端点 7 接收缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP1_TX_DMA	0x40030040	端点 1 发送缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP2_TX_DMA	0x40030044	端点 2 发送缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP3_TX_DMA	0x40030048	端点 3 发送缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP4_TX_DMA	0x4003004C	端点 4 发送缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP5_TX_DMA	0x40030050	端点 5 发送缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP6_TX_DMA	0x40030054	端点 6 发送缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP7_TX_DMA	0x40030058	端点 7 发送缓冲区的起始地址寄存器	0x000XXXXX
R32_UEP0_MAX_LEN	0x4003005C	端点 0 最大长度包寄存器	0x000000XX
R32_UEP1_MAX_LEN	0x40030060	端点 1 最大长度包寄存器	0x000000XX
R32_UEP2_MAX_LEN	0x40030064	端点 2 最大长度包寄存器	0x000000XX
R32_UEP3_MAX_LEN	0x40030068	端点 3 最大长度包寄存器	0x000000XX
R32_UEP4_MAX_LEN	0x4003006C	端点 4 最大长度包寄存器	0x000000XX
R32_UEP5_MAX_LEN	0x40030070	端点 5 最大长度包寄存器	0x000000XX
R32_UEP6_MAX_LEN	0x40030074	端点 6 最大长度包寄存器	0x000000XX
R32_UEP7_MAX_LEN	0x40030078	端点 7 最大长度包寄存器	0x000000XX
R16_UEP0_RX_LEN	0x4003007C	端点 0 接收长度寄存器	0x00XX
R16_UEP1_RX_LEN	0x40030080	端点 1 单次接收长度寄存器	0xFFFF
R16_UEP1_R_SIZE	0x40030082	端点 1 总接收数据长度寄存器	0xFFFF
R16_UEP2_RX_LEN	0x40030084	端点 2 单次接收长度寄存器	0xFFFF
R16_UEP2_R_SIZE	0x40030086	端点 2 总接收数据长度寄存器	0xFFFF
R16_UEP3_RX_LEN	0x40030088	端点 3 单次接收长度寄存器	0xFFFF
R16_UEP3_R_SIZE	0x4003008A	端点 3 总接收数据长度寄存器	0xFFFF
R16_UEP4_RX_LEN	0x4003008C	端点 4 单次接收长度寄存器	0xFFFF
R16_UEP4_R_SIZE	0x4003008E	端点 4 总接收数据长度寄存器	0xFFFF
R16_UEP5_RX_LEN	0x40030090	端点 5 单次接收长度寄存器	0xFFFF
R16_UEP5_R_SIZE	0x40030092	端点 5 总接收数据长度寄存器	0xFFFF
R16_UEP6_RX_LEN	0x40030094	端点 6 单次接收长度寄存器	0xFFFF
R16_UEP6_R_SIZE	0x40030096	端点 6 总接收数据长度寄存器	0xFFFF
R16_UEP7_RX_LEN	0x40030098	端点 7 单次接收长度寄存器	0xFFFF
R16_UEP7_R_SIZE	0x4003009A	端点 7 总接收数据长度寄存器	0xFFFF
R16_UEP0_T_LEN	0x4003009C	端点 0 发送长度寄存器	0x00XX
R8_UEP0_TX_CTRL	0x4003009E	端点 0 发送控制寄存器	0x00
R8_UEP0_RX_CTRL	0x4003009F	端点 0 接收控制寄存器	0x00
R16_UEP1_T_LEN	0x400300A0	端点 1 发送长度寄存器	0x0000
R8_UEP1_TX_CTRL	0x400300A2	端点 1 发送控制寄存器	0x00
R8_UEP1_RX_CTRL	0x400300A3	端点 1 接收控制寄存器	0x00

R16_UEP2_T_LEN	0x400300A4	端点 2 发送长度寄存器	0x0000
R8_UEP2_TX_CTRL	0x400300A6	端点 2 发送控制寄存器	0x00
R8_UEP2_RX_CTRL	0x400300A7	端点 2 接收控制寄存器	0x00
R16_UEP3_T_LEN	0x400300A8	端点 3 发送长度寄存器	0x0000
R8_UEP3_TX_CTRL	0x400300AA	端点 3 发送控制寄存器	0x00
R8_UEP3_RX_CTRL	0x400300AB	端点 3 接收控制寄存器	0x00
R16_UEP4_T_LEN	0x400300AC	端点 4 发送长度寄存器	0x0000
R8_UEP4_TX_CTRL	0x400300AE	端点 4 发送控制寄存器	0x00
R8_UEP4_RX_CTRL	0x400300AF	端点 4 接收控制寄存器	0x00
R16_UEP5_T_LEN	0x400300B0	端点 5 发送长度寄存器	0x0000
R8_UEP5_TX_CTRL	0x400300B2	端点 5 发送控制寄存器	0x00
R8_UEP5_RX_CTRL	0x400300B3	端点 5 接收控制寄存器	0x00
R16_UEP6_T_LEN	0x400300B4	端点 6 发送长度寄存器	0x0000
R8_UEP6_TX_CTRL	0x400300B6	端点 6 发送控制寄存器	0x00
R8_UEP6_RX_CTRL	0x400300B7	端点 6 接收控制寄存器	0x00
R16_UEP7_T_LEN	0x400300B8	端点 7 发送长度寄存器	0x0000
R8_UEP7_TX_CTRL	0x400300BA	端点 7 发送控制寄存器	0x00
R8_UEP7_RX_CTRL	0x400300BB	端点 7 接收控制寄存器	0x00
R16_UEP_T_ISO	0x400300BC	USBHS 端点发送同步模式使能寄存器	0x0000
R16_UEP_R_ISO	0x400300BE	USBHS 端点接收同步模式使能寄存器	0x0000
R32_UEP1_RX_FIFO	0x400300C0	USBHS 端点 1 的接收 FIFO 地址	0x00000000
R32_UEP2_RX_FIFO	0x400300C4	USBHS 端点 2 的接收 FIFO 地址	0x00000000
R32_UEP3_RX_FIFO	0x400300C8	USBHS 端点 3 的接收 FIFO 地址	0x00000000
R32_UEP4_RX_FIFO	0x400300CC	USBHS 端点 4 的接收 FIFO 地址	0x00000000
R32_UEP5_RX_FIFO	0x400300D0	USBHS 端点 5 的接收 FIFO 地址	0x00000000
R32_UEP6_RX_FIFO	0x400300D4	USBHS 端点 6 的接收 FIFO 地址	0x00000000
R32_UEP7_RX_FIFO	0x400300D8	USBHS 端点 7 的接收 FIFO 地址	0x00000000
R32_UEP1_TX_FIFO	0x400300DC	USBHS 端点 1 的接收 FIFO 地址	0x00000000
R32_UEP2_TX_FIFO	0x400300E0	USBHS 端点 2 的发送 FIFO 地址	0x00000000
R32_UEP3_TX_FIFO	0x400300E4	USBHS 端点 3 的发送 FIFO 地址	0x00000000
R32_UEP4_TX_FIFO	0x400300E8	USBHS 端点 4 的发送 FIFO 地址	0x00000000
R32_UEP5_TX_FIFO	0x400300EC	USBHS 端点 5 的发送 FIFO 地址	0x00000000
R32_UEP6_TX_FIFO	0x400300F0	USBHS 端点 6 的发送 FIFO 地址	0x00000000
R32_UEP7_TX_FIFO	0x400300F4	USBHS 端点 7 的发送 FIFO 地址	0x00000000

25.2.1.1 USBHS 控制寄存器 (R8_USB_CTRL)

偏移地址：0x00

位	名称	访问	描述	复位值
7	RB_UD_LPM_EN	RW	LPM 使能： 1：使能； 0：禁止。	0
6	Reserved	RO	保留。	0
5	RB_UD_DEV_EN	RW	USB 设备使能： 1：使能； 0：禁止。	0

4	RB_UD_DMA_EN	RW	DMA 传输使能： 1：使能； 0：禁止。	0
3	RB_UD_PHY_SUSPENDM	RW	USB PHY 挂起： 1：正常工作； 0：挂起。	0
2	RB_UD_CLR_ALL	RW	清除所有中断标志： 1：清空 USB 中断标志和 FIFO，需要软件清零； 0：不清空。	1
1	RB_UD_RST_SIE	RW	USB 协议处理器复位： 1：强制复位 USB 协议处理器（SIE），包括端点相关寄存器，需要软件清零； 0：不复位。	1
0	RB_UD_RST_LINK	RW	LINK 层复位： 1：USB Link 层复位； 0：不复位。	1

25.2.1.2 USBHS 模式控制寄存器（R8_USB_BASE_MODE）

偏移地址：0x01

位	名称	访问	描述	复位值
[7:2]	Reserved	RO	保留。	0x0
[1:0]	RB_UD_SPEED_TYPE[1:0]	RW	设备期望的速度模式： 00：全速设备； 01：高速设备； 10：低速设备； 11：保留。	0

25.2.1.3 USBHS 中断使能寄存器（R8_USB_INT_EN）

偏移地址：0x02

位	名称	访问	描述	复位值
7	RB_UDIE_FIFO_OVER	RW	FIFO 溢出中断。 1：使能中断； 0：禁止中断。	0
6	RB_UDIE_LINK_RDY	RW	USB 连接中断使能。 1：使能中断； 0：禁止中断。	0
5	RB_UDIE_SOF_ACT	RW	接收 SOF 包中断使能。 1：使能中断； 0：禁止中断。	0
4	RB_UDIE_TRANSFER	RW	USB 传输结束中断使能。 1：使能中断； 0：禁止中断。	0
3	RB_UDIE_LPM_ACT	RW	LPM 传输结束中断使能。 1：使能中断；	0

			0: 禁止中断。	
2	RB_UDIE_BUS_SLEEP	RW	USB 总线睡眠中断使能。 1: 使能中断; 0: 禁止中断。	0
1	RB_UDIE_SUSPEND	RW	USB 总线暂停中断使能。 1: 使能中断; 0: 禁止中断。	0
0	RB_UDIE_BUS_RST	RW	USB 总线复位中断使能。 1: 使能中断; 0: 禁止中断。	0

25.2.1.4 USBHS 设备地址寄存器 (R8_USB_DEV_AD)

偏移地址: 0x03

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
[6:0]	RB_UD_DEV_ADDR[6:0]	RW	USB 设备地址。	0

25.2.1.5 USBHS 远程唤醒寄存器 (R8_USB_WAKE_CTRL)

偏移地址: 0x04

位	名称	访问	描述	复位值
[7:1]	Reserved	RO	保留。	0
0	RB_UD_REMOTE_WKUP	RW1Z	远程唤醒, 硬件自动清零。	0

25.2.1.6 USBHS 测试模式寄存器 (R8_USB_TEST_MODE)

偏移地址: 0x05

位	名称	访问	描述	复位值
7	RB_UD_TEST_EN	RW	测试模式使能。	0
[6:4]	Reserved	RO	保留。	0
3	RB_UD_TEST_SEONAK	RW	测试模式, 输出 SE0。	0
2	RB_UD_TEST_PKT	RW	测试模式, 输出一个包。 包使用端点 4 的数据地址以及长度, TOG 为 DATA0。	0
1	RB_UD_TEST_K	RW	测试模式, 输出 K。	0
0	RB_UD_TEST_J	RW	测试模式, 输出 J。	0

25.2.1.7 USBHS 电源管理寄存器 (R16_USB_LPM_DATA)

偏移地址: 0x06

位	名称	访问	描述	复位值
15	RB_UD_LPM_BUSY	RW	电源管理忙。	1
[14:11]	Reserved	RO	保留。	0
[10:0]	RB_UD_LPM_DATA[10:0]	RO	电源管理数据。	0

25.2.1.8 USB 中断标志寄存器 (R8_USB_INT_FG)

偏移地址: 0x08

位	名称	访问	描述	复位值
7	RB_UDIF_FIFO_OV	RW1Z	FIFO 溢出中断标志位，写 1 清零。 1: FIFO 溢出触发； 0: 无事件。	0
6	RB_UDIF_LINK_RDY	RW1Z	USB 连接中断标志位，写 1 清零。 1: USB 连接事件触发； 0: 无事件。	0
5	RB_UDIF_RX_SOF	RW1Z	接收 SOF 包中断标志位，写 1 清零。 1: 接收 SOF 包事件触发； 0: 无事件。	0
4	RB_UDIF_RTX_ACT	R0	USB 传输结束中断标志位： 接收由 RB_UEP_R_DONE 清除； 发送由 RB_UEP_T_DONE 清除。	0
3	RB_UDIF_LPM_ACT	RW1Z	LPM 传输结束中断标志位，写 1 清零。 1: LPM 传输结束事件触发； 0: 无事件。	0
2	RB_UDIF_BUS_SLEEP	RW1Z	USB 总线睡眠中断标志位，当启用 LPM 时，进入 L1 (Sleep) 状态会触发，写 1 清零。 1: USB 总线睡眠事件触发； 0: 无事件。	0
1	RB_UDIF_SUSPEND	RW1Z	USB 总线挂起中断标志位，写 1 清零。 1: USB 挂起事件触发； 0: 无事件。	0
0	RB_UDIF_BUS_RST	RW1Z	USB 总线复位中断标志位，写 1 清零。 1: USB 总线复位事件触发； 0: 无事件。	0

25.2.1.9 USB 中断状态寄存器 (R8_USB_INT_ST)

偏移地址：0x09

位	名称	访问	描述	复位值
[7:5]	Reserved	R0	保留。	0
4	RB_UDIS_EP_DIR	R0	端点数据传输方向： 1: 端点 IN 数据； 0: 端点 OUT/SETUP 数据。	0
3	Reserved	R0	保留。	0
[2:0]	RB_UDIS_EP_ID_MASK [2:0]	R0	发生数据传输的端点号。	0

25.2.1.10 USB 杂项状态寄存器 (R8_USB_MIS_ST)

偏移地址：0x0A

位	名称	访问	描述	复位值
7	RB_UDMS_HS_MOD	R0	主机是否为高速。	0
[6:5]	Reserved	R0	保留。	0
4	RB_UDMS_SUSP_REQ	R0	USB 挂起请求。	0

3	RB_UDMS_SIE_FREE	R0	USB 空闲状态。	0
2	RB_UDMS_SLEEP	R0	USB 睡眠状态。	0
1	RB_UDMS_SUSPEND	R0	USB 挂起状态。 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
0	RB_UDMS_READY	R0	USB 连接状态。	0

25.2.1.11 USB 帧寄存器 (R16_USB_FRAME_NO)

偏移地址: 0x0C

位	名称	访问	描述	复位值
[15:13]	RB_UD_MFRAME_NO[2:0]	R0	接收到的微帧号。	0
[12:11]	Reserved	R0	保留。	0
[10:0]	RB_UD_FRAME_NO[10:0]	R0	接收到的帧号。	0

25.2.1.12 USB 总线状态寄存器 (R16_USB_BUS)

偏移地址: 0x0E

位	名称	访问	描述	复位值
[15:4]	Reserved	R0	保留。	0
3	RB_USB_DM_ST	R0	UDM 状态。	0
2	RB_USB_DP_ST	R0	UDP 状态。	0
1	Reserved	R0	保留。	0
0	RB_USB_WAKEUP	R0	USB 唤醒 (高有效)。	0

25.2.1.13 USBHS 端点发送使能寄存器 (R16_UEP_TX_EN)

偏移地址: 0x10

位	名称	访问	描述	复位值
[15:0]	RB_UEP_TX_EN[15:0]	RW	0~15 端点发送使能: 1: 使能; 0: 禁止。	0

25.2.1.14 USBHS 端点接收使能寄存器 (R16_UEP_RX_EN)

偏移地址: 0x12

位	名称	访问	描述	复位值
[15:0]	RB_UEP_RX_EN[15:0]	RW	0~15 端点接收使能: 1: 使能; 0: 禁止。	0

25.2.1.15 USBHS 端点发送自动翻转使能寄存器 (R16_UEP_T_TOG_AUTO)

偏移地址: 0x14

位	名称	访问	描述	复位值
[15:8]	Reserved	R0	保留。	0

[7:0]	RB_UEP_T_TOG_AUTO[7:0]	RW	0~7 端点同步触发位自动翻转使能： 1：数据发送成功后自动翻转； 0：手动控制翻转。 注：端点 0 仅支持手动控制翻转。	0

25.2.1.16 USBHS 端点接收自动翻转使能寄存器 (R16_UEP_R_TOG_AUTO)

偏移地址：0x16

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留	0
[7:0]	RB_UEP_R_TOG_AUTO[7:0]	RW	0~7 端点同步触发位自动翻转使能： 1：数据接收成功后自动翻转； 0：手动控制翻转。 注：端点 0 仅支持手动控制翻转。	0

25.2.1.17 USBHS 端点发送突发寄存器 (R8_UEP_T_BURST)

偏移地址：0x18

位	名称	访问	描述	复位值
[7:0]	RB_UEP_T_BURST_EN[7:0]	RW	0~7 端点 burst 发送使能： 1：使能； 0：禁止。	0

25.2.1.18 USBHS 端点发送模式寄存器 (R8_UEP_T_BURST_MODE)

偏移地址：0x19

位	名称	访问	描述	复位值
[7:0]	RB_UEP_T_BURST_MODE[7:0]	RW	0~7 端点 burst 发送模式： 1：突发模式下不发送 0 长度数据； 0：突发模式下发送 0 长度数据。	0

25.2.1.19 USBHS 端点接收突发寄存器 (R8_UEP_R_BURST)

偏移地址：0x1A

位	名称	访问	描述	复位值
[7:0]	RB_UEP_R_BURST_EN[7:0]	RW	0~7 端点 burst 接收使能： 1：使能； 0：禁止。	0

25.2.1.20 USBHS 端点回复模式寄存器 (R8_UEP_R_RES_MODE)

偏移地址：0x1B

位	名称	访问	描述	复位值
[7:0]	RB_UEP_R_RES_MODE[7:0]	RW	0~7 端点接收返回模式： 1：NYET； 0：ACK。	0

25.2.1.21 USBHS 端点复用寄存器 (R32_UEP_AF_MODE)

偏移地址：0x1C

位	名称	访问	描述	复位值
[31:8]	Reserved	R0	保留。	0
[7:1]	RB_UEP_T_AF[6:0]	RW	1~7 端点复用使能： 1：复用 9-15； 0：复用 1-7。	0
0	Reserved	R0	保留。	0

25.2.1.22 端点 0 缓冲区的起始地址寄存器 (R32_UEP0_DMA)

偏移地址：0x20

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:0]	UEP0_DMA[23:0]	RW	端点 0 的缓冲区起始地址。	X

25.2.1.23 端点 n 接收缓冲区的起始地址寄存器 (R32_UEPn_RX_DMA) (n=1~7)

偏移地址：0x24 + 4* (n-1)

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:0]	UEPn_RX_DMA[23:0]	RW	端点 n 的接收缓冲区起始地址。地址必须 4 字节对齐。	X

25.2.1.24 端点 n 发送缓冲区的起始地址寄存器 (R32_UEPn_TX_DMA) (n=1~7)

偏移地址：0x40 + 4* (n-1)

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:0]	UEPn_TX_DMA[23:0]	RW	端点 n 的发送缓冲区起始地址。地址必须 4 字节对齐。	X

25.2.1.25 端点 n 最大长度包寄存器 (R32_UEPn_MAX_LEN) (n=0)

偏移地址：0x5C

位	名称	访问	描述	复位值
[31:7]	Reserved	R0	保留。	0
[6:0]	UEPn_MAX_LEN[6:0]	RW	端点 n DMA 可偏移的最大限制长度。	X

25.2.1.26 端点 n 最大长度包寄存器 (R32_UEPn_MAX_LEN) (n=1~7)

偏移地址：0x60 + 4 (n-1)

位	名称	访问	描述	复位值
[31:11]	Reserved	R0	保留。	0
[10:0]	UEPn_MAX_LEN[10:0]	RW	端点 n DMA 可偏移的最大限制长度。	X

25.2.1.27 端点 0 接收长度寄存器 (R16_UEP0_RX_LEN)

偏移地址：0x7C

位	名称	访问	描述	复位值
[15:7]	Reserved	R0	保留。	0
[6:0]	UEP0_RX_LEN[6:0]	RW	端点 0 接收数据长度。	X

25.2.1.28 端点 n 单次接收长度寄存器 (R16_UEPn_RX_LEN) (n=1~7)偏移地址: $0x80 + 4 * (n-1)$

位	名称	访问	描述	复位值
[15:0]	UEPn_RX_LEN[15:0]	RW	端点 n 单次接收数据长度。	X

25.2.1.29 端点 n 总接收数据长度寄存器 (R16_UEPn_R_SIZE) (n=1~7)偏移地址: $0x82 + 4 * (n-1)$

位	名称	访问	描述	复位值
[15:0]	UEPn_R_SIZE[15:0]	RW	端点 n 总接收数据长度。	X

25.2.1.30 端点 0 发送长度寄存器 (R16_UEP0_T_LEN)偏移地址: $0x9C$

位	名称	访问	描述	复位值
[15:7]	Reserved	RO	保留。	0
[6:0]	UEP0_T_LEN[6:0]	RW	端点 0 发送数据长度。	X

25.2.1.31 端点 0 发送控制寄存器 (R8_UEP0_TX_CTRL)偏移地址: $0x9E$

位	名称	访问	描述	复位值
7	RB_UEP_T_DONE	RWO	端点 0 发送结束标志, 写 0 清除。	0
6	RB_UEP_T_NAK_ACT	RWO	端点 0 发送 NAK 结束标志, 写 0 清除。	0
[5:3]	Reserved	RO	保留。	0
2	RB_UEP_T_TOG_MASK	RW	端点 0 的发送期望的同步触发位: 1: DATA1; 0: DATA0。	0
[1:0]	RB_UEP_T_RES_MASK[1:0]	RW	端点 0 对于发送的响应控制: 00: 应答 NAK 或忙; 01: 应答 STALL 或错误; 10: 应答 ACK; 11: 保留。	0

25.2.1.32 端点 0 接收控制寄存器 (R8_UEP0_RX_CTRL)偏移地址: $0x9F$

位	名称	访问	描述	复位值
7	RB_UEP_R_DONE	RWO	端点 0 接收结束标志, 写 0 清除。	0
6	RB_UEP_R_NAK_ACT	RWO	端点 0 接收 NAK 结束标志, 写 0 清除。	0
5	RB_UEP_R_NAK_TOG	RO	端点 0 对于接收的返回 NAK, 数据包类型: 1: DATA1; 0: DATA0。	0
4	RB_UEP_R_TOG_MATCH	RO	接收的同步触发位与期望的同步触发位匹配状态: 1: 同步;	0

			0: 不同步。	
3	RB_UEP_R_SETUP_IS	R0	端点 0 接收的是否为 SETUP 事务。	0
2	RB_UEP_R_TOG_MASK	RW	端点 0 的接收期望的同步触发位: 1: DATA1; 0: DATA0。	0
[1:0]	RB_UEP_R_RES_MASK[1:0]	RW	端点 0 对于接收的响应控制: 00: 应答 NAK 或忙; 01: 应答 STALL 或错误; 10: 应答 ACK; 11: 保留。	0

25.2.1.33 端点 n 发送长度寄存器 (R16_UEPn_T_LEN) (n=1~7)

偏移地址: $0xA0 + 4 * (n-1)$

位	名称	访问	描述	复位值
[15:0]	UEPn_T_LEN[15:0]	RW	端点 n 发送数据长度。	0

25.2.1.34 端点 n 发送控制寄存器 (R8_UEPn_TX_CTRL) (n=1~7)

偏移地址: $0xA2 + 4 * (n-1)$

位	名称	访问	描述	复位值
7	RB_UEP_T_DONE	RW0	端点 n 发送结束标志, 写 0 清除。	0
6	RB_UEP_T_NAK_ACT	RW0	端点 n 发送 NAK 结束标志, 写 0 清除。	0
[5:4]	Reserved	R0	保留。	0
[3:2]	RB_UEP_T_TOG_MASK[1:0]	RW	端点 n 的发送期望的同步触发位: 00: DATA0; 01: DATA1; 10: DATA2; 11: MDATA。 <i>注: 此位手动模式下可配置, 自动模式下不可配置。</i>	0
[1:0]	RB_UEP_T_RES_MASK[1:0]	RW	端点 n 对于发送的响应控制: 00: 应答 NAK 或忙; 01: 应答 STALL 或错误; 10: 应答 ACK; 11: 保留。	0

25.2.1.35 端点 n 接收控制寄存器 (R8_UEPn_RX_CTRL) (n=1~7)

偏移地址: $0xA3 + 4 * (n-1)$

位	名称	访问	描述	复位值
7	RB_UEP_R_DONE	RW0	端点 n 接收结束标志, 写 0 清除。	0
6	RB_UEP_R_NAK_ACT	RW0	端点 n 接收 NAK 结束标志, 写 0 清除。	0
5	RB_UEP_R_NAK_TOG	R0	端点 n 对于接收的返回 NAK, 数据包类型: 1: DATA1; 0: DATA0。	0
4	RB_UEP_R_TOG_MATCH	R0	端点 n 接收的同步触发位与期望的同	0

			步触发位匹配状态： 1：同步； 0：不同步。	
[3:2]	RB_UEP_R_TOG_MASK[1:0]	RW	端点 n 的接收期望的同步触发位： 00：DATA0； 01：DATA1； 10：DATA2； 11：MDATA。 注：此位手动模式下可配置，自动模式下不可配置。	0
[1:0]	RB_UEP_R_RES_MASK[1:0]	RW	端点 n 对于接收的响应控制： 00：应答 NAK 或忙； 01：应答 STALL 或错误； 10：应答 ACK； 11：保留。	0

25.2.1.36 USBHS 端点发送同步模式使能寄存器 (R16_UEP_T_ISO)

偏移地址：0xBC

位	名称	访问	描述	复位值
[15:9]	RB_UEP_T_FIFO_EN[6:0]	RO	端点的 TX 的 FIFO 模式使能。	0
8	Reserved	RO	保留。	0
[7:1]	RB_UEPn_T_ISO_EN[6:0]	RW	上传端点 (IN) 同步模式使能。	0
0	Reserved	RO	保留。	0

25.2.1.37 USBHS 端点接收同步模式使能寄存器 (R16_UEP_R_ISO)

偏移地址：0xBE

位	名称	访问	描述	复位值
[15:9]	RB_UEP_R_FIFO_EN[6:0]	RO	端点的 RX 的 FIFO 模式使能。	0
8	Reserved	RO	保留。	0
[7:1]	RB_UEPn_R_ISO_EN[6:0]	RW	下传端点 (OUT) 同步模式使能。	0
0	Reserved	RO	保留。	0

25.2.1.38 USBHS 端点 n 的接收 FIFO 地址 (R32_UEPn_RX_FIFO) (n=1~7)

偏移地址：0xC0 + 4* (n-1)

位	名称	访问	描述	复位值
[31:16]	RB_UEP_RX_FIFO_E	RW	端点接收的 FIFO 结束地址。	0
[15:0]	RB_UEP_RX_FIFO_S	RW	端点接收的 FIFO 起始地址。	0

25.2.1.39 USBHS 端点 n 的发送 FIFO 地址 (R32_UEPn_TX_FIFO) (n=1~7)

偏移地址：0xDC + 4* (n-1)

位	名称	访问	描述	复位值

[31:16]	RB_UEP_TX_FIFO_E	RW	端点发送的 FIFO 结束地址。	0
[15:0]	RB_UEP_TX_FIFO_S	RW	端点发送的 FIFO 起始地址。	0

25.2.2 主机寄存器描述

表 25-2 主机相关寄存器列表

名称	访问地址	描述	复位值
R8_UH_CFG	0x40030100	USBHS 主机配置寄存器	0x07
R8_UH_INT_EN	0x40030102	USBHS 主机中断使能寄存器	0x00
R8_UH_DEV_AD	0x40030103	USBHS 主机设备地址寄存器	0x00
R32_UH_CONTROL	0x40030104	USBHS 主机控制寄存器	0x00XXXXXX
R8_UH_INT_FLAG	0x40030108	USBHS 主机中断标志寄存器	0x00
R8_UH_INT_ST	0x40030109	USBHS 主机中断状态寄存器	0xFF
R8_UH_MIS_ST	0x4003010A	USBHS 主机杂项状态寄存器	0xFF
R32_UH_LPM_DATA	0x4003010C	USBHS 主机电源管理数据寄存器	0x00000XXX
R32_UH_SPLIT_DATA	0x40030110	USBHS 主机 SPLIT 数据寄存器	0x000XXXXX
R32_UH_FRAME	0x40030114	USBHS 主机帧寄存器	0x00000000
R32_UH_TX_LEN	0x40030118	USBHS 主机发送长度寄存器	0x00000XXX
R32_UH_RX_LEN	0x4003011C	USBHS 主机接收长度寄存器	0x00000XXX
R32_UH_RX_MAX_LEN	0x40030120	USBHS 主机接收最大长度寄存器	0x00000XXX
R32_UH_RX_DMA	0x40030124	DMA 接收地址寄存器	0x000XXXXX
R32_UH_TX_DMA	0x40030128	DMA 发送地址寄存器	0x000XXXXX
R32_UH_PORT_CTRL	0x4003012C	USBHS 主机端口控制寄存器	0x0000X000
R8_UH_PORT_CFG	0x40030130	USBHS 主机端口配置寄存器	0x00
R8_UH_PORT_INT_EN	0x40030132	USBHS 主机端口中断使能寄存器	0x00
R8_UH_PORT_TEST_CT	0x40030133	USBHS 主机端口测试模式寄存器	0x00
R16_UH_PORT_ST	0x40030134	USBHS 主机端口状态寄存器	0x0010
R8_UH_PORT_CHG	0x40030136	USBHS 主机端口状态变化寄存器	0x00
R32_UH_BC_CTRL	0x4003013C	USBHS 主机 BC 充电控制寄存器	0x00000000

25.2.2.1 USBHS 主机配置寄存器 (R8_UH_CFG)

偏移地址: 0x00

位	名称	访问	描述	复位值
7	RB_UH_LPM_EN	RW	LPM 使能。	0
6	RB_UH_FORCE_FS	RW	强制使用 USB_FS。	0
5	RB_UH_SOF_EN	RW	使能 SOF 包发送。	0
4	RB_UH_DMA_EN	RW	使能 DMA 发送。	0
3	RB_UH_PHY_SUSPENDM	RW	PHY 挂起。	0
2	RB_UH_CLR_ALL	RW	清除中断标志以及 FIFO, 需要软件清零。	1
1	RB_UH_RST_SIE	RW	USB 协议处理器复位, 需要软件清零。	1
0	RB_UH_RST_LINK	RW	USB 连接控制模块复位。	1

25.2.2.2 USBHS 主机中断使能寄存器 (R8_UH_INT_EN)

偏移地址: 0x02

位	名称	访问	描述	复位值

7	RB_UHIE_FIFO_OVER	RW	FIFO 溢出中断使能。	0
6	RB_UHIE_TX_HALT	RW	发送暂停中断使能。	0
5	RB_UHIE_SOF_ACT	RW	SOF 包发送中断使能。	0
4	RB_UHIE_TRANSFER	RW	USB 结束或传输完成中断使能。	0
3	RB_UHIE_RESUME_ACT	RW	总线恢复中断使能。	0
2	RB_UHIE_WKUP_ACT	RW	唤醒中断使能。	0
[1:0]	Reserved	RO	保留。	0

25.2.2.3 USBHS 主机设备地址寄存器 (R8_UH_DEV_AD)

偏移地址：0x03

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	0
[6:0]	RB_UH_DEV_ADDR[6:0]	RW	当前操作的 USB 设备的地址。	0

25.2.2.4 USBHS 主机控制寄存器 (R32_UH_CONTROL)

偏移地址：0x04

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
23	RB_UH_RX_NO_RES	RW	IN-DATA 无应答, 用于同步传输或高速 SPLIT 包。	X
22	RB_UH_TX_NO_RES	RW	OUT/SETUP - DATA 不期望应答, 用于同步传输或高速 SPLIT 包。	X
21	RB_UH_RX_NO_DATA	RW	IN 令牌包后不期望数据包, 用于高速 SPLIT 包。	X
20	RB_UH_TX_NO_DATA	RW	OUT/SETUP 令牌包后无数据包, 用于高速 SPLIT 包。	X
19	RB_UH_PRE_PID_EN	RW	低速前导包 PRE PID 使能位, 当端口工作于全速, 需要发送低速包 (PRE) 需开启此位。 1: 使能, 用于通过外部全速 HUB 与低速 USB 设备通讯; 0: 禁用低速前导包。	X
18	RB_UH_SPLIT_VALID	RW	发送 SPLIT 包有效。	X
17	RB_UH_LPM_VALID	RW	发送 LPM 包有效。	X
16	RB_UH_HOST_ACTION	RW	HOST 执行事务使能, 事务完成后该位自动清零。	0
[15:11]	Reserved	RO	保留。	0
10	RB_UH_BUF_MODE	RW	数据缓存区控制位。 1: 发送时使用 R32_UH_RX_DMA, 接收使用 R32_UH_TX_DMA; 0: 发送时使用 R32_UH_TX_DMA, 接收使用 R32_UH_RX_DMA。	X
[9:8]	RB_UH_T_TOG_MASK [1:0]	RW	发送的数据 PID: 00: PID_DATA0; 01: PID_DATA1;	X

			10: PID_DATA2; 11: PID_MDATA。	
[7:4]	RB_UH_T_ENDP_MASK [3:0]	RW	主机发送的事务令牌包端点号。	X
[3:0]	RB_UH_T_TOKEN_MASK [3:0]	RW	主机发送的事务令牌包 PID。	X

25.2.2.5 USBHS 主机中断标志寄存器 (R8_UH_INT_FLAG)

偏移地址: 0x08

位	名称	访问	描述	复位值
7	RB_UHIF_FIFO_OVER	RW1Z	FIFO 溢出中断标志。	0
6	RB_UHIF_TX_HALT	RW1Z	发送中止中断标志, 主机模式下用于下传中读数据溢出, 或在 EOF2 时间点发送未结束, 中止发送, 写 1 清零。	0
5	RB_UHIF_SOF_ACT	RW1Z	SOF 包发送完成中断标志, 写 1 清零。	0
4	RB_UHIF_TRANSFER	RW1Z	USB 事务传输完成中断标志, 写 1 清零。	0
3	RB_UHIF_RESUME_ACT_IF	RW1Z	总线恢复中断标志, 写 1 清零。	0
2	RB_UHIF_WKUP_ACT	RW1Z	唤醒中断标志, 写 1 清零。	0
[1:0]	Reserved	RO	保留。	0

25.2.2.6 USBHS 主机中断状态寄存器 (R8_UH_INT_ST)

偏移地址: 0x09

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	X
4	RB_UHIS_PORT_RX_RESUME	RW	该位为 1 表示端口接收到唤醒信号。	X
[3:0]	RB_UH_R_TOKEN_MASK [3:0]	RO	接收到的 PID。	X

25.2.2.7 USBHS 主机杂项状态寄存器 (R8_UH_MIS_ST)

偏移地址: 0x0A

位	名称	访问	描述	复位值
7	RB_UHMS_BUS_SE0	RO	USB 总线上为 SE0。	X
6	RB_UHMS_BUS_J	RO	USB 总线上为 J。	X
[5:4]	RB_UHMS_LINESTATE[1:0]	RO	PHY 的 Linestate 信号。	X
3	RB_UHMS_USB_WAKEUP	RO	USB 总线唤醒状态: 1: 正在进行总线唤醒; 0: 未在进行总线唤醒。	X
2	RB_UHMS_SOF_ACT	RO	USB 总线 SOF 包传输状态位: 1: 正在发出 SOF 包; 0: 发送完成或空闲。	X
1	RB_UHMS_SOF_PRE	RO	USB 总线 SOF 包预示状态为: 1: 将要发送 SOF 包;	X

			0: 无 SOF 包发送。	
0	RB_UHMS_SOF_FREE	RO	端口使能状态: 1: 端口使能; 0: 端口未使能。	X

25.2.2.8 USBHS 主机电源管理数据寄存器 (R32_UH_LPM_DATA)

偏移地址: 0x0C

位	名称	访问	描述	复位值
[31:11]	Reserved	RO	保留。	0
[10:0]	RB_UH_LPM_DATA[10:0]	RW	链路电源管理包数据内容。	X

25.2.2.9 USBHS 主机 SPLIT 寄存器 (R32_UH_SPLIT_DATA)

偏移地址: 0x10

位	名称	访问	描述	复位值
[31:19]	Reserved	RO	保留。	0
[18:0]	RB_UH_SPLIT_DATA[18:0]	RW	主机端点发送 SPLIT 包的数据内容, 从高位到低位依次为 ET、E、S、Port、SC、Hub Addr。	X

25.2.2.10 USBHS 主机帧寄存器 (R32_UH_FRAME)

偏移地址: 0x14

位	名称	访问	描述	复位值
[31:26]	Reserved	RO	保留。	0
25	RB_UH_SOF_CNT_CLR	RW	SOF 计数清零。	0
24	RB_UH_SOF_CNT_EN	RW	SOF 计数使能。	0
[23:19]	Reserved	RO	保留。	0
[18:16]	RB_UH_MFRAME_NO[2:0]	RO	即将发送的 SOF 包的微帧号。	0
[15:11]	Reserved	RO	保留。	0
[10:0]	RB_UH_FRAME_NO[10:0]	RW	即将发送的 SOF 包的帧号。	0

25.2.2.11 USBHS 主机发送长度寄存器 (R32_UH_TX_LEN)

偏移地址: 0x18

位	名称	访问	描述	复位值
[31:11]	Reserved	RO	保留。	0
[10:0]	RB_UH_TX_LEN[10:0]	RW	发送数据长度。	X

25.2.2.12 USBHS 主机接收长度寄存器 (R32_UH_RX_LEN)

偏移地址: 0x1C

位	名称	访问	描述	复位值
[31:11]	Reserved	RO	保留。	0
[10:0]	RB_UH_RX_LEN[10:0]	RW	接收数据长度。	X

25.2.2.13 USBHS 主机接收最大长度寄存器 (R32_UH_RX_MAX_LEN)

偏移地址: 0x20

位	名称	访问	描述	复位值
[31:11]	Reserved	RO	保留。	0
[10:0]	RB_UH_RX_MAX_LEN[10:0]	RW	接收最大长度。	X

25.2.2.14 DMA 接收地址寄存器 (R32_UH_RX_DMA)

偏移地址: 0x24

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:0]	R32_UH_RX_DMA[23:0]	RW	接收地址, 低两位无效, 需要 4 字节对齐。	X

25.2.2.15 DMA 发送地址寄存器 (R32_UH_TX_DMA)

偏移地址: 0x28

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:0]	R32_UH_TX_DMA[23:0]	RW	发送地址, 低两位无效, 需要 4 字节对齐。	X

25.2.2.16 USBHS 主机端口控制寄存器 (R32_UH_PORT_CTRL)

偏移地址: 0x2C

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
16	RB_UH_BUS_RST_LONG	RW	总线复位时间选择: 1: 较长复位; 0: 正常复位。	0
[15:12]	RB_UH_PORT_SLEEP_BESL[3:0]	RW	唤醒时间控制: 0000: 125us; 1000: 3ms; 0001: 150us; 1001: 4ms; 0010: 200us; 1010: 5ms; 0011: 300us; 1011: 6ms; 0100: 400us; 1100: 7ms; 0101: 500us; 1101: 8ms; 0110: 1ms; 1110: 9ms; 0111: 2ms; 1111: 10ms。	X
[11:9]	Reserved	RO	保留。	0
8	RB_UH_CLR_PORT_SLEEP	WO	PORT 退出 SLEEP 状态 (LPM)。	0
[7:6]	Reserved	RO	保留。	0
5	RB_UH_CLR_PORT_CONNECT	WO	使端口进入端口状态。	0
4	RB_UH_CLR_PORT_EN	WO	PORT 退出使能状态, 进入 DISABLED 状态。	0

			态。	
3	RB_UH_SET_PORT_SLEEP	WO	PORT 进入 SLEEP 状态 (LPM)。	0
2	RB_UH_CLR_PORT_SUSP	WO	PORT 退出挂起状态。	0
1	RB_UH_SET_PORT_SUSP	WO	PORT 进入挂起状态。	0
0	RB_UH_SET_PORT_RESET	WO	端口发送复位。	0

25.2.2.17 USBHS 主机端口配置寄存器 (R8_UH_PORT_CFG)

偏移地址: 0x30

位	名称	访问	描述	复位值
7	RB_UH_PD_EN	RW	主机模式下 15K 电阻下拉使能。	0
[6:1]	Reserved	RO	保留。	0
0	RB_UH_HOST_EN	RW	USB 端口模式选择: 1: 端口为主机模式; 0: 端口为设备模式。	0

25.2.2.18 USBHS 主机端口中断使能寄存器 (R8_UH_PORT_INT_EN)

偏移地址: 0x32

位	名称	访问	描述	复位值
[7:6]	Reserved	RO	保留。	0
5	RB_UHIE_PORT_SLP	RW	端口睡眠状态变化中断使能。	0
4	RB_UHIE_PORT_RESET	RW	端口复位状态变化中断使能。	0
3	Reserved	RO	保留。	0
2	RB_UHIE_PORT_SUSP	RW	端口暂停状态变化中断使能。	0
1	RB_UHIE_PORT_EN	RW	端口使能状态变化中断使能。	0
0	RB_UHIE_PORT_CONNECTED	RW	端口连接状态变化中断使能。	0

25.2.2.19 USBHS 主机端口测试模式寄存器 (R8_UH_PORT_TEST_CT)

偏移地址: 0x33

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
4	RB_UH_TEST_SEO_NAK	RW	测试 TEST_SEO_NAK。	0
3	RB_UH_TEST_PACKET	RW	测试 TEST_PKT。	0
2	RB_UH_TEST_FORCE_EN	RW	测试模式使能。	0
1	RB_UH_TEST_K	RW	测试输出 K。	0
0	RB_UH_TEST_J	RW	测试输出 J。	0

25.2.2.20 USBHS 主机端口状态寄存器 (R16_UH_PORT_ST)

偏移地址: 0x34

位	名称	访问	描述	复位值
[15:12]	Reserved	RO	保留。	0
11	RB_UHIS_PORT_TEST	RO	端口是否在 test 模式。	0

10	RB_UHIS_PORT_HS	R0	端口连接速度是否为高速。	0
9	RB_UHIS_PORT_LS	R0	端口连接速度是否为低速。	0
[8:6]	Reserved	R0	保留。	0
5	RB_UHIS_PORT_SLP	R0	端口睡眠。	0
4	RB_UHIS_PORT_RST	R0	端口复位状态。	1
3	Reserved	R0	保留。	0
2	RB_UHIS_PORT_SUSP	R0	端口暂停状态。	0
1	RB_UHIS_PORT_EN	R0	端口使能。	0
0	RB_UHIS_PORT_CONNEC T	R0	端口连接状态。	0

25.2.2.21 USBHS 主机端口状态变化寄存器 (R8_UH_PORT_CHG)

偏移地址: 0x36

位	名称	访问	描述	复位值
[7:6]	Reserved	R0	保留。	0
5	RB_UHIF_PORT_SLP	RW1Z	端口睡眠状态变化, 写 1 清零。	0
4	RB_UHIF_PORT_RESET	RW1Z	端口复位状态变化, 写 1 清零。	0
3	Reserved	R0	保留。	0
2	RB_UHIF_PORT_SUSP	RW1Z	端口暂停状态变化, 写 1 清零。	0
1	RB_UHIF_PORT_EN	RW1Z	端口使能状态变化, 写 1 清零。	0
0	RB_UHIF_PORT_CONNEC T	RW1Z	端口连接状态变化, 写 1 清零。	0

25.2.2.22 USBHS 主机 BC 充电控制寄存器 (R32_UH_BC_CTRL)

偏移地址: 0x3C

位	名称	访问	描述	复位值
[31:11]	Reserved	R0	保留。	0
10	RB_UDM_VSRC_ACT	R0	自动模式下, 该位表示 UDM 输出 V_{BC_SRC} , 否则由 RB_UDM_BC_CMPE 控制。 注: UDP 高于 V_{BC_SRC} 但是 UDP 不是高电平时该位为 1, 否则为 0。	0
9	RB_UDM_BC_VSRC	RW	UDM 引脚 BC 协议源电压使能: 1: UDM 引脚输出 BC 协议源电压 V_{BC_SRC} ; 0: 禁止输出。	0
8	RB_UDP_BC_VSRC	RW	UDP 引脚 BC 协议源电压使能: 1: UDP 引脚输出 BC 协议源电压 V_{BC_SRC} ; 0: 禁止输出。	0
7	Reserved	R0	保留。	0
6	RB_BC_AUTO_MODE	RW	自动模式使能: 1: 使能; 0: 关闭。	0
5	RB_UDM_BC_CMPE	RW	UDM 引脚 BC 协议比较器使能: 1: 使能; 0: 关闭。	0
4	RB_UDP_BC_CMPE	RW	UDP 引脚 BC 协议比较器使能:	0

			1: 使能; 0: 关闭。	
[3:2]	Reserved	R0	保留。	0
1	RB_UDM_BC_CMP0	R0	UDM 引脚 BC 协议比较器状态: 1:UDM 引脚电压高于 BC 协议参考值 V_{BC_REF} ; 0:UDM 引脚电压低于 BC 协议参考值 V_{BC_REF} 。	0
0	RB_UDP_BC_CMP0	R0	UDP 引脚 BC 协议比较器状态: 1:UDP 引脚电压高于 BC 协议参考值 V_{BC_REF} ; 0:UDP 引脚电压低于 BC 协议参考值 V_{BC_REF} 。	0

第 26 章 USB 全速主机/设备控制器（USBFS/OTG_FS）

26.1 USB 控制器简介

USB 2.0 全速主机控制器和设备控制器（USBFS），遵循 USB 2.0 Fullspeed 标准。提供 16 个可配置的 USB 设备端点及一组主机端点。支持控制/批量/同步/中断传输，双缓冲区机制，USB 总线挂起/恢复操作，并提供待机/唤醒功能。

OTG_FS 是双重角色 USB 控制器，支持主机端和设备端的功能，兼容 On-The-Go Supplement to the USB 2.0 规范。同时，该控制器也可配置为仅支持主机端或仅支持设备端功能的控制器，兼容 USB 2.0 全速规范。

主要特性如下：

- 支持在（OTG_FS 控制器的物理层）USB On-The-Go Supplement, Revision1.3 规范中定义为可选项目 OTG 协议
- 通过软件可配置 USB 全速主机、USB 全速/低速设备、USB 双重角色设备
- 提供省电功能
- 支持控制传输、批量传输、中断传输、实时/同步传输
- 提供总线复位、挂起、唤醒和恢复功能
- 支持最大 64 字节的数据包，内置 FIFO，支持中断和 DMA

26.2 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

26.2.1 全局寄存器描述

表 26-1 USBFS 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	0x00
R8_USB_DEV_AD	0x40023403	USB 设备地址寄存器	0x00
R8_USB_MIS_ST	0x40023405	USB 杂项状态寄存器	0xXX
R8_USB_INT_FG	0x40023406	USB 中断标志寄存器	0x20
R8_USB_INT_ST	0x40023407	USB 中断状态寄存器	0xXX
R16_USB_RX_LEN	0x40023408	USB 接收长度寄存器	0x0XXX
R32_USB_OTG_CR	0x40023454	USB OTG 控制寄存器	0x00000000
R32_USB_OTG_SR	0x40023458	USB OTG 状态寄存器	0x0000000X

26.2.1.1 USB 控制寄存器（R8_USB_CTRL）

偏移地址：0x00

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_RST_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 26-2 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。

26.2.1.2 USB 中断使能寄存器 (R8_USB_INT_EN)

偏移地址: 0x02

位	名称	访问	描述	复位值
7	Reserved	RO	保留。	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	RB_U_1WIRE_MODE	RW	USB 单线模式使能： 1: 使能 USB 单线模式;	0

			0: 禁止 USB 单线模式。	
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

26.2.1.3 USB 设备地址寄存器 (R8_USB_DEV_AD)

偏移地址: 0x03

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

26.2.1.4 USB 杂项状态寄存器 (R8_USB_MIS_ST)

偏移地址: 0x05

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_RST	RO	USB 总线复位状态位:	X

			1: 当前 USB 总线处于复位态; 0: 当前 USB 总线处于非复位态。	
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

26.2.1.5 USB 中断标志寄存器 (R8_USB_INT_FG)

偏移地址: 0x06

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

			0: 无事件。	
	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

26.2.1.6 USB 中断状态寄存器 (R8_USB_INT_ST)

偏移地址: 0x07

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。	X
[3:0]	MASK_UIS_ENDP	RO	设备模式下, 当前 USB 传输事务的端点号。	X
	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。	X

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 保留; 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。

26.2.1.7 USB 接收长度寄存器 (R16_USB_RX_LEN)

偏移地址: 0x08

位	名称	访问	描述	复位值
[15:10]	Reserved	RO	保留。	0
[9:0]	R16_USB_RX_LEN	RO	当前 USB 端点接收的数据字节数	X

26.2.1.8 USB OTG 控制寄存器 (R32_USB_OTG_CR)

偏移地址: 0x54

位	名称	访问	描述	复位值
[31:6]	Reserved	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:	0

			1: VBUS_VLD 电平为 4.4V; 0: VBUS_VLD 电平为 4.8V。	
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

23.2.1.9 USB OTG 状态寄存器 (R32_USB_OTG_SR)

偏移地址: 0x58

位	名称	访问	描述	复位值
[31:4]	Reserved	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

26.2.2 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。

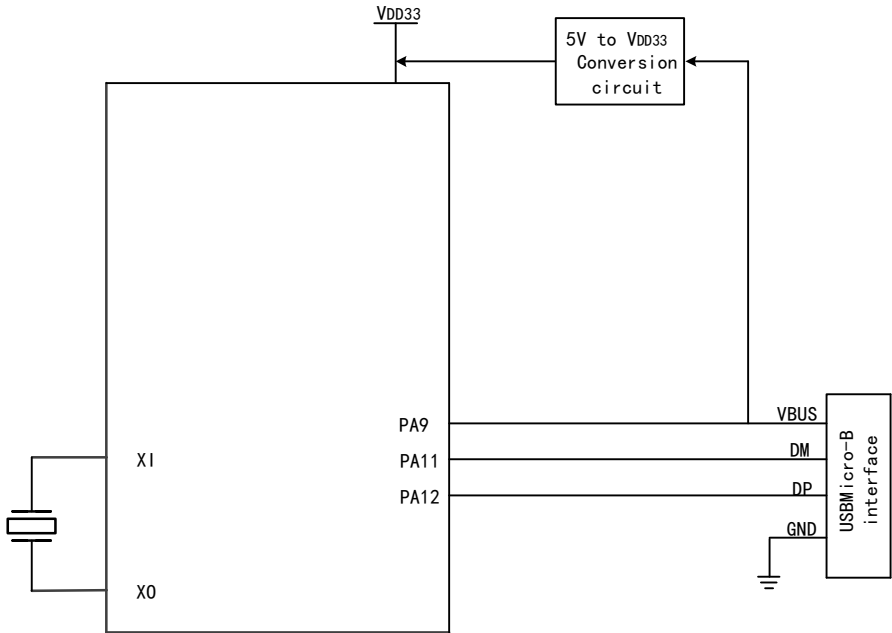
每组端点都具有一个控制寄存器 R8_UEPn_CTRL 和发送长度寄存器 R16_UEPn_T_LEN, 用于设定该端点的同步触发位、对 OUT 事务和 IN 事务的响应以及发送数据的长度等。

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 $V_{\text{SESS\_VLD, min}}$ ，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 26-1 所示。

图 26-1 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_AUTO_TOG 或 RB_UEP_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。

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 26-3 设备相关寄存器列表

名称	访问地址	描述	复位值
R8_UDEV_CTRL	0x40023401	USB 设备物理端口控制寄存器	0x00
R8_UEP4_1_MOD	0x4002340C	端点 1 (9) / 4 (8/12) 模式控制寄存器	0x00
R8_UEP2_3_MOD	0x4002340D	端点 2 (10) / 3 (11) 模式控制寄存器	0x00

R8_UEP5_6_MOD	0x4002340E	端点 5（13）/6（14）模式控制寄存器	0x00
R8_UEP7_MOD	0x4002340F	端点 7（15）模式控制寄存器	0x00
R32_UEP0_DMA	0x40023410	端点 0 缓冲区起始地址	0xFFFFFFFF
R32_UEP1_DMA	0x40023414	端点 1（9）缓冲区起始地址	0xFFFFFFFF
R32_UEP2_DMA	0x40023418	端点 2（10）缓冲区起始地址	0xFFFFFFFF
R32_UEP3_DMA	0x4002341C	端点 3（11）缓冲区起始地址	0xFFFFFFFF
R32_UEP4_DMA	0x40023420	端点 4（8/12）缓冲区起始地址	0xFFFFFFFF
R32_UEP5_DMA	0x40023424	端点 5（13）缓冲区起始地址	0xFFFFFFFF
R32_UEP6_DMA	0x40023428	端点 6（14）缓冲区起始地址	0xFFFFFFFF
R32_UEP7_DMA	0x4002342C	端点 7（15）缓冲区起始地址	0xFFFFFFFF
R8_UEP0_T_LEN	0x40023430	端点 0 发送长度寄存器	0xFF
R8_UEP0_TX_CTRL	0x40023432	端点 0 发送控制寄存器	0x00
R8_UEP0_RX_CTRL	0x40023433	端点 0 接收控制寄存器	0x00
R8_UEP1_T_LEN	0x40023434	端点 1（9）发送长度寄存器	0xFF
R8_UEP1_TX_CTRL	0x40023436	端点 1（9）发送控制寄存器	0x00
R8_UEP1_RX_CTRL	0x40023437	端点 1（9）接收控制寄存器	0x00
R8_UEP2_T_LEN	0x40023438	端点 2（10）发送长度寄存器	0xFF
R8_UEP2_TX_CTRL	0x4002343A	端点 2（10）发送控制寄存器	0x00
R8_UEP2_RX_CTRL	0x4002343B	端点 2（10）接收控制寄存器	0x00
R16_UEP3_T_LEN	0x4002343C	端点 3（11）发送长度寄存器	0xFF
R8_UEP3_TX_CTRL	0x4002343E	端点 3（11）发送控制寄存器	0x00
R8_UEP3_RX_CTRL	0x4002343F	端点 3（11）接收控制寄存器	0x00
R8_UEP4_T_LEN	0x40023440	端点 4（8/12）发送长度寄存器	0xFF
R8_UEP4_TX_CTRL	0x40023442	端点 4（8/12）发送控制寄存器	0x00
R8_UEP4_RX_CTRL	0x40023443	端点 4（8/12）接收控制寄存器	0x00
R8_UEP5_T_LEN	0x40023444	端点 5（13）发送长度寄存器	0xFF
R8_UEP5_TX_CTRL	0x40023446	端点 5（13）发送控制寄存器	0x00
R8_UEP5_RX_CTRL	0x40023447	端点 5（13）接收控制寄存器	0x00
R8_UEP6_T_LEN	0x40023448	端点 6（14）发送长度寄存器	0xFF
R8_UEP6_TX_CTRL	0x4002344A	端点 6（14）发送控制寄存器	0x00
R8_UEP6_RX_CTRL	0x4002344B	端点 6（14）接收控制寄存器	0x00
R8_UEP7_T_LEN	0x4002344C	端点 7（15）发送长度寄存器	0xFF
R8_UEP7_TX_CTRL	0x4002344E	端点 7（15）发送控制寄存器	0x00
R8_UEP7_RX_CTRL	0x4002344F	端点 7（15）接收控制寄存器	0x00

26.2.2.1 USB 设备物理端口控制寄存器（R8_UDEV_CTRL）

偏移地址：0x01

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

26.2.2.2 端点 1（9）/4（8/12）模式控制寄存器（R8_UEP4_1_MOD）

偏移地址：0x0C

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。	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。	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。

26.2.2.3 端点 2（10）/3（11）模式控制寄存器（R8_UEP2_3_MOD）

偏移地址：0x0D

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。	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	R0	保留。	0
0	RB_UEP2_BUF_MOD	RW	端点 2（10）数据缓冲区模式控制位。	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。

26.2.2.4 端点 5（13）/6（14）模式控制寄存器（R8_UEP5_6_MOD）

偏移地址：0x0E

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	R0	保留。	0
4	RB_UEP6_BUF_MOD	RW	端点 6（14）数据缓冲区模式控制位。	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	R0	保留。	0
0	RB_UEP5_BUF_MOD	RW	端点 5（13）数据缓冲区模式控制位。	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。

26.2.2.5 端点 7（15）模式控制寄存器（R8_UEP7_MOD）

偏移地址：0x0F

位	名称	访问	描述	复位值
[7:4]	Reserved	R0	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	R0	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 26-4。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。需要注意的是：在同步传输时，（RB_UEPn_RX_EN, RB_UEPn_TX_EN, RB_UEPn_BUF_MOD）仅为（1, 0, 0）或（0, 1, 0）时，端点 3 支持最大 1023 字节。

表 26-4 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_T0G 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_T0G 选择， 双 64 字节发送缓冲区 (IN)，通过 RB_UEP_T_T0G 选择。 全部 256 字节排列如下： UEPn_DMA+0 地址：RB_UEP_R_T0G=0 时端点接收地址； UEPn_DMA+64 地址：RB_UEP_R_T0G=1 时端点接收地址； UEPn_DMA+128 地址：RB_UEP_T_T0G=0 时端点发送地址； UEPn_DMA+192 地址：RB_UEP_T_T0G=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。

26.2.2.6 端点 n 缓冲区起始地址 (R32_UEPn_DMA) ($n=0-7$)

偏移地址： $0x10 + (4*n)$

位	名称	访问	描述	复位值
[31:0]	R32_UEPn_DMA	RW	端点 n 缓冲区起始地址。 地址必须 4 字节对齐。	X

注：1. 接收数据的缓冲区的长度 $\geq \min$ (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。

26.2.2.7 端点 n 发送长度寄存器 (R8_UEPn_T_LEN) ($n=0-2, 4-7$)

偏移地址： $0x30 + (4*n)$

位	名称	访问	描述	复位值
[7:0]	R8_UEPn_T_LEN	RW	设置 USB 端点 n 准备发送的数据字节数 $n=0, 1, 2, 4, 5, 6, 7$ 。	X

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)。

26.2.2.8 端点 n 发送长度寄存器 (R16_UEPn_T_LEN) ($n=3$)

偏移地址：0x3C

位	名称	访问	描述	复位值
[15:10]	Reserved	RO	保留。	0
[9:0]	R16_UEP3_T_LEN	RW	设置 USB 端点 3 准备发送的数据字节数。	X

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)。

26.2.2.9 端点 n 控制寄存器 (R8_UEPn_TX_CTRL) ($n=0-7$)

偏移地址： $0x32 + (4*n)$

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0
3	RB_UEP_T_AUTO_T0G	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。	0
[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。

26.2.2.10 端点 n 控制寄存器（R8_UEPn_RX_CTRL）（n=0-7）

偏移地址：0x33 + (4*n)

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。	0
[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

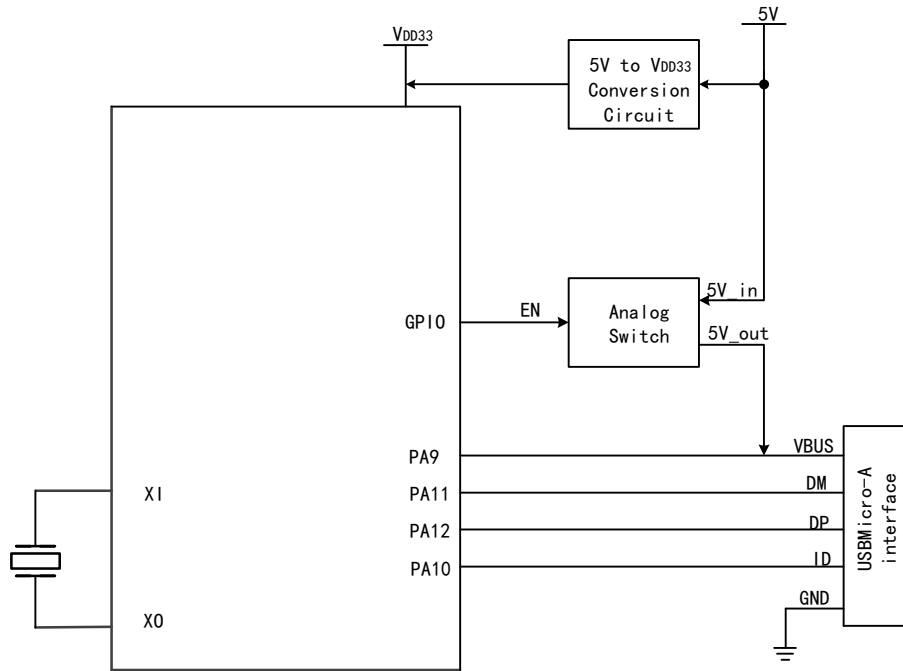
注：端点配置支持 0-7 端点，可映射配置端点 8-15 端点。

26.2.3 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 26-2 所示。

图 26-2 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 R32_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，接收到数据存放在 R32_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 26-5 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 主机物理端口控制寄存器	0x00
R32_UH_EP_MOD	0x4002340D	USB 主机端点模式控制寄存器	0x00000000
R32_UH_RX_DMA	0x40023418	USB 主机接收缓冲区起始地址	X
R32_UH_TX_DMA	0x4002341C	USB 主机发送缓冲区起始地址	X
R16_UH_SETUP	0x40023436	USB 主机辅助设置寄存器	0x0000
R8_UH_EP_PID	0x40023438	USB 主机令牌设置寄存器	0x00
R8_UH_RX_CTRL	0x4002343B	USB 主机接收端点控制寄存器	0x00
R16_UH_TX_LEN	0x4002343C	USB 主机发送长度寄存器	X
R8_UH_TX_CTRL	0x4002343E	USB 主机发送端点控制寄存器	0x00

23.2.3.1 USB 主机物理端口控制寄存器 (R8_UHOST_CTRL)

偏移地址: 0x01

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_RST	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

23.2.3.2 USB 主机端点模式控制寄存器 (R32_UH_EP_MOD)

偏移地址: 0x0D

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 26-6 主机发送缓冲区模式

RB_UH_EP_TX_EN	RB_UH_EP_TBUF_MOD	描述：以 R32_UH_TX_DMA 为起始地址
0	X	端点被禁用，未用到 R32_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 26-7 主机接收缓冲区模式

RB_UH_EP_RX_EN	RB_UH_EP_RBUF_MOD	结构描述：以 R32_UH_TX_DMA 为起始地址
0	X	端点被禁用，未用到 R32_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。

26.2.3.3 USB 主机接收缓冲区起始地址（R32_UH_RX_DMA）

偏移地址：0x18

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:0]	R32_UH_RX_DMA	RW	主机端点数据接收缓冲区起始地址。 低 20 位有效，地址必须 4 字节对齐。	x

26.2.3.4 USB 主机发送缓冲区起始地址（R32_UH_TX_DMA）

偏移地址：0x1C

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:0]	R32_UH_TX_DMA	RW	主机端点数据发送缓冲区起始地址。 低 20 位有效，地址必须 4 字节对齐。	x

26.2.3.5 USB 主机辅助设置寄存器（R16_UH_SETUP）

偏移地址：0x36

位	名称	访问	描述	复位值
[15:11]	Reserved	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]	Reserved	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]	Reserved	R0	保留。	0

23.2.3.6 USB 主机令牌设置寄存器 (R8_UH_EP_PID)

偏移地址: 0x38

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

26.2.3.7 USB 主机接收端点控制寄存器 (R8_UH_RX_CTRL)

偏移地址: 0x3B

位	名称	访问	描述	复位值
[7:4]	Reserved	R0	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	Reserved	R0	保留。	0
0	RB_UH_R_RES	R0	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

26.2.3.8 USB 主机发送长度寄存器 (R16_UH_TX_LEN)

偏移地址: 0x3C

位	名称	访问	描述	复位值
[15:0]	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

26.2.3.9 USB 主机发送端点控制寄存器 (R8_UH_TX_CTRL)

偏移地址: 0x3E

位	名称	访问	描述	复位值
[7:4]	Reserved	R0	保留。	0
3	RB_UH_T_AUTO_TOG	R0	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	R0	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：期望应答 ACK。	0

第 27 章 USB 3.0 超高速主机/设备控制器（USBSS）

本章模块描述仅适用于 CH32H417、CH32H416 微控制器产品。

USB 3.0 超高速控制器具有主机控制器和设备控制器双重角色，内置超高速的 USB PHY 物理层收发器，可实现 USB3.0 接口产品功能，支持 5Gbps 的 USBSS 超高速信号。

USBSS 模块为应用代码提供了链接层寄存器访问接口，用于管理设备的连接和断开、总线状态、电源模式。提供了主机（HOST）功能访问接口，设备（DEVICE）功能访问接口，用于实现 USB 3.0 协议规范的各种数据传输及上层协议。

USBSS 模块外设基地址：0x40034000

27.1 主要特征

- 支持 USB 3.0 协议规范和 USB 3.2 Gen1
- 支持 USB Host 主机功能和 USB Device 设备功能
- 电源管理模式支持 U1/U2/U3 低功耗状态
- 支持驱动 USB 3.0 HUB
- 支持控制传输、批量传输、中断传输、实时/同步传输
- 非 0 端点均支持最大 1024 字节的数据包，支持突发模式
- 支持 DMA 方式直接访问各端点缓冲区的数据
- 自研控制器和收发器，高速一体化设计，实测每秒 450Mbytes

27.2 寄存器描述

USBSS 相关寄存器分为 4 个部分：

- USBSS 全局寄存器
- USBSS 设备寄存器
- USBSS 主机控制寄存器
- USBSS LINK 控制寄存器

27.2.1 全局寄存器描述

表 27-1 USBSS 全局寄存器列表

名称	访问地址	描述	复位值
R32_USBSS_CTRL	0x40034070	USBSS 控制寄存器	0x00000006
R32_USBSS_STATUS	0x40034074	USBSS 状态寄存器	0x00000000
R32_UH_TX_DMA R32_UEP0_TX_DMA	0x4003408C	发送缓冲区地址寄存器	0xFFFFFFFF
R32_UH_RX_DMA R32_UEP0_RX_DMA	0x40034090	接收缓冲区地址寄存器	0xFFFFFFFF

27.2.1.1 USBSS 控制寄存器（R32_USBSS_CTRL）

偏移地址：0x70

位	名称	访问	描述	复位值
31	Reserved	RO	保留。	0
[30:24]	RB_DEV_ADDR	RW	主机模式： 当前操作的 USB 设备的地址； 设备模式：	0

			该 USB 设备的地址。	
23	RB_UIE_FIFO_RXOV	RW	接收 FIFO 溢出中断使能： 1：使能； 0：关闭。	0
22	RB_UIE_FIFO_TXOV	RW	发送 FIFO 溢出中断使能： 1：使能； 0：关闭。	0
21	Reserved	RO	保留。	0
20	RB_UIE_ITP	RW	发送 ITP 完成中断使能： 1：使能； 0：关闭。	0
19	RB_UIE_RX_PING	RW	接收 PING-TP 完成中断使能： 1：使能； 0：关闭。	0
18	RB_UDIE_STATUS	RW	设备模式：接收 STATUS 事务完成中断使能： 1：使能； 0：关闭。	0
18	RB_UHIE_NOTIF	RW	主机模式：接收 DEV_NOTIF-TP 完成中断使能： 1：使能； 0：关闭。	0
17	RB_UDIE_SETUP	RW	设备模式：接收 SETUP 事务完成中断使能： 1：使能； 0：关闭。	0
17	RB_UHIE_ERDY	RW	主机模式：接收 SETUP 事务完成中断使能： 1：使能； 0：关闭。	0
16	RB_UIE_TRANSFER	RW	USB 事务完成中断使能： 1：使能； 0：关闭。	0
15	Reserved	RO	保留。	0
14	RB_TX_ERDY_MODE	RW	设备模式下使用： 1：禁止 ERDY-TP 插入到两个 DPH 包之间； 0：允许 ERDY-TP 插入到两个 DPH 包之间。	0
[13:10]	Reserved	RO	保留。	0
[9:8]	RB_REG_HP_PEND[1:0]	RW	发送 TP/DP 包的 Packet Pending 控制位： 在突发模式下，低位表示最后一个包的 PP 位状态，高位表示非最后一个包的 PP 位状态。 非突发模式下，仅低位有效	0
7	RB_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_ITP_EN	RW	主机模式下，发送 ITP 使能： 1：使能； 0：关闭。	0
5	RB_SETUP_FLOW	RW	SETUP 事务流控	0

			1: 接收到 SETUP 事务后, 进入流控, 需要设备; 0: 接收到 SETUP 事务后, 不需要流控。	
4	Reserved	RO	保留。	0
3	RB_DMA_MODE	RW	1: 进行突发传输 DPH 时, 不预取下一包数据到 FIFO 中; 0: 进行突发传输 DPH 时, 预取下一包数据到 FIFO 中。 注: 开启预取可以让突发的 DPH 包间隔最小, 提高传输带宽, 默认预取。	0
2	RB_FORCE_RST	RW	协议层及 FIFO 模块复位, 需要软件清零: 1: 复位; 0: 不复位。	1
1	RB_USB_CLR_ALL	RW	复位所有软件配置寄存器, 高有效, 需要软件清零; 清除所有中断标志、清除设备地址, 清除所有端点的配置。 注: 设备模式下, 应在收到 HOT_RESET/WARM_RESET 后置位该位, 然后清零。	1
0	RB_DMA_EN	RW	使能 DMA: 1: 使能, 控制器可访问 SRAM 中数据; 0: 禁止, 控制器无法访问 SRAM 中数据。	0

27.2.1.2 USBSS 状态寄存器 (R32_USBSS_STATUS)

偏移地址: 0x74

位	名称	访问	描述	复位值
[31:30]	RB_HRX_RES [1:0]	RW	接收到来自设备返回的应答 TP: 00: ACK-TP; 01: RX_FAILED; 10: NRDY-TP; 11: STALL-TP。 注: TX_FAILED/RX_FAILED 意味着当主机发起事务时, HUB 下行口处于低功耗状态, HUB 将标记改包为 deferred, 同时返回主机, 非同步事务主机应等到接收到设备的 ERDY-TP 之后重新进行该事务。	0
[29:24]	Reserved	RO	保留。	0
[23:22]	RB_HTX_RES [1:0]	RW	接收到来自主机发送的 TP: 00: ACK-TP; 01: TX_FAILED; 10: NRDY-TP; 11: STALL-TP。 注: TX_FAILED/RX_FAILED 意味着当主机发起事务时, HUB 下行口处于低功耗状态, HUB 将标记改包为 deferred, 同时返回主机, 非同	0

			步事务主机应等到接收到设备的 ERDY-TP 之后重新进行该事务。	
21	Reserved	R0	保留。	0
[20:16]	RB_HOST_ACK_NUMP	RW	该位在同步传输下无效，在控制传输、块传输、中断传输中有如下定义： 发送 DPH+DPP (OUT/SETUP 事务) 后接收到 ACK-TP 中的 NUMP 字段，该字段表示设备该端点接下来能接收的包的个数，如果 NUMP=0 表示该端点无法继续接收 DPP，等到设备的该端点可以接收 DPH+DPP，设备会发送 ERDY-TP 包通知主机。但并不影响主机此时向其他端点发送数据。 特别地，如果 SETUP 事务的应答 ACK-TP 的 NUMP 为 0，表明进入流控状态，主机应该暂停数据阶段或状态阶段，直到接收到来自设备的 ERDY-TP。另外主机可以发送 SETUP 事务解除流控状态。当然处于流控状态时，主机也是可以进行数据阶段和状态阶段，不会产生错误，只是当设备的该端点没有准备好时会返回 NRDY-TP。	0
[15:13]	Reserved	R0	保留。	0
12	RB_EP_DIR	RW	当前有传输完成中断标志的端点的方向，如果收发同时有中断标志，那么该位为 1，即优先处理发送。（SETUP/STATUS 独立中断） 1：发送（IN）； 0：接收（OUT）。	0
11	Reserved	R0	保留。	0
[10:8]	RB_EP_ID	RW	当前有传输完成中断标志的端点号，如果多个端点同时有中断标志，按照端点类型优先级顺序如下（高→低）：同步传输>控制传输>块传输/中断传输，同一端点类型端点号越小优先级越高。	0
7	RB_UIF_FIFO_RXOV	RW1Z	接收 FIFO 溢出中断标志位： 1：中断已使能； 0：中断未使能。	0
6	RB_UIF_FIFO_TXOV	RW1Z	发送 FIFO 溢出中断标志位： 1：中断已使能； 0：中断未使能。	0
5	Reserved	R0	保留。	0
4	RB_UIF_ITP	RW1Z	发送 ITP 完成中断标志位： 1：中断已使能； 0：中断未使能。	0
3	RB_UIF_RX_PING	RW1Z	接收 PING-TP 完成中断标志位： 1：中断已使能； 0：中断未使能。	0
2	RB_UHIF_NOTIF	RW1Z	主机模式：接收 DEV_NOTIF-TP 完成中断标志	

			位： 1：中断已使能； 0：中断未使能。	
2	RB_UDIF_STATUS	RW1Z	设备模式：接收 STATUS 事务完成中断标志位： 1：中断已使能； 0：中断未使能。	0
1	RB_UHIF_ERDY	RW1Z	主机模式：接收 ERDY-TP 完成中断标志位： 1：中断已使能； 0：中断未使能。	
1	RB_UDIF_SETUP	RW1Z	设备模式：接收 SETUP 事务完成中断标志位： 1：中断已使能； 0：中断未使能。	0
0	RB_UIF_TRANSFER	RW1Z	USB 事务完成中断标志位： 1：中断已使能； 0：中断未使能。	0

27.2.1.3 发送缓冲区地址寄存器 (R32_UH_TX_DMA/U3EP0_TX_DMA)

偏移地址：0x8C

位	名称	访问	描述	复位值
[31:0]	U3EP0_TX_DMA	RW	主机模式下： 主机发送缓冲区的起始地址。 设备模式下： 端点 0 发送缓冲区起始地址。	X

27.2.1.4 接收缓冲区地址寄存器 (R32_UH_RX_DMA/U3EP0_RX_DMA)

偏移地址：0x90

位	名称	访问	描述	复位值
[31:0]	U3EP0_RX_DMA	RW	主机模式下： 主机接收缓冲区的起始地址。 设备模式下： 端点 0 接收缓冲区起始地址。	X

27.2.2 设备寄存器描述

表 27-2 USBSS 设备寄存器列表

名称	访问地址	描述	复位值
R32_USBSS_ITP	0x40034078	USB 中 ITP 包的 Interval 寄存器	0x00000000
R32_USBSS_ITP_ADJ	0x4003407C	USB 中 ITP 包的 Interval 自适应寄存器	0x00000000
R16_UEP_TX_EN	0x40034080	端点发送使能寄存器	0x00000000
R16_UEP_RX_EN	0x40034082	端点接收使能寄存器	0x00000000
R32_UEP0_TX_CTRL	0x40034084	端点 0 发送控制寄存器	0x00000000
R32_UEP0_RX_CTRL	0x40034088	端点 0 接收控制寄存器	0x00000000

27.2.2.1 USB 中 ITP 包的 Interval 寄存器 (R32_USBSS_ITP)

偏移地址: 0x78

位	名称	访问	描述	复位值
[31:14]	Reserved	RO	保留。	0
[13:0]	REG_ITP_INTERVAL	RO	接收的 ITP 中 Bus interval Counter 字段。 注: Bus interval Counter 每 125us 加一次。	0

27.2.2.2 USB 中 ITP 包的 Interval 自适应寄存器 (R32_USBSS_ITP_ADJ)

偏移地址: 0x7C

位	名称	访问	描述	复位值
[31:21]	Reserved	RO	保留。	0
[20:8]	ITP_DELTA	RO	设备模式下接收的 ITP 中 ITS 的高 13bit (Delta), 表示从当前 ITP 包的开始到前一个总线间隔边界的时间差值。如果主机精确的每 125us 发送一次 ITP, 主机应设置该位为 0。	0
7	ITP_DELAYED	RO	设备模式下, 接收的 ITP 中 Link Control Word 的 Delayed 位, 当 ITP 被延迟传输时为 1。	0
[6:0]	ITP_ADJ_CR	RO	设备模式下, 接收的 ITP 中 Bus Interval Adjustment Control 字段, 上电复位或者设备断开后, 主机发送的 ITP 的该字段应该为 0。	0

27.2.2.3 端点发送使能寄存器 (R16_UEP_TX_EN)

偏移地址: 0x80

位	名称	访问	描述	复位值
[15:1]	RB_EP_TX_EN	RW	端点 1~15 上传使能: 1: 使能; 0: 禁止。	0
0	Reserved	RO	保留。	1

27.2.2.4 端点接收使能寄存器 (R16_UEP_RX_EN)

偏移地址: 0x82

位	名称	访问	描述	复位值
[15:1]	RB_EP_RX_EN	RW	端点 1~15 下传使能: 1: 使能; 0: 禁止。	0
0	Reserved	RO	保留。	1

27.2.2.5 端点 0 发送控制寄存器 (R32_UEP0_TX_CTRL)

偏移地址: 0x84

位	名称	访问	描述	复位值
31	RB_UIF_EP0_TX_AC	RW	上传事务完成中断标志, 软件写 0 清零, 由	0

	T		硬件置 1。	
[20:26]	Reserved	RO	保留。	0
25	RB_EP0_TX_FLOW	RO	完成 NRDY-TP 发送（应答）的标志。	0
24	RB_EP0_TX_PP	RO	接收的 ACK-TP 中的 PP 位。	0
23	RB_EP0_TX_ERDY	RW	该位写 1，发送 ERDY，ERDY 发送完成后该位硬件自动清零。	0
[22:21]	RB_EP0_TX_RES[1:0]	RW	对 ACK-TP (nump ≠ 0) 的应答： 00: NRDY-TP； 01: DPH； 10: STALL-TP； 11: 不应答（无效）。 IN 事务完成后自动清零。	0
[20:16]	RB_EP0_TX_SEQ[4:0]	RW	端点当前序列号，接收到 SETUP 事务后，该位自动清零。	0
[15:11]	Reserved	RO	保留。	0
[10:0]	RB_EP0_TX_LEN[10:0]	RW	端点发送长度寄存器，最大值为 512B。	x

27.2.2.6 端点 0 接收控制寄存器（R32_UEP0_RX_CTRL）

偏移地址：0x88

位	名称	访问	描述	复位值
31	RB_UIF_EP0_RX_ACT	RW	下传事务完成中断标志，软件写 0 清零，由硬件置 1。	0
[30:25]	Reserved	RO	保留。	0
24	RB_EP0_RX_PP	RO	接收的 DPH 中的 PP 位。	0
23	RB_EP0_RX_ERDY	RW	该位写 1，发送 ERDY，ERDY 发送完成后该位硬件自动清零。	0
[22:21]	RB_EP0_RX_RES[1:0]	RW	对 DPH 的应答： 00: NRDY-TP； 01: ACK-TP； 10: STALL-TP； 11: 不应答（无效）。 SETUP/OUT 事务完成后 EPn_R_RES 自动清零； 另外 SETUP 包直接应答 ACK-TP，与此无关。	0
[20:16]	RB_EP0_RX_SEQ[4:0]	RW	端点期望接收的序列号。	0
[15:11]	Reserved	RO	保留。	0
[10:0]	RB_EP0_RX_LEN[10:0]	RO	端点接收长度寄存器。	x

27.2.3 设备寄存器—端点 1~7 发送相关寄存器

表 27-3 端点 1 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP1_TX_CFG	0x400340C0	端点 1 配置寄存器	0x86
R8_UEP1_TX_CR	0x400340C1	端点 1 控制寄存器	0x10

R8_UEP1_TX_SEQ	0x400340C2	端点 1 序列号寄存器	0x00
R8_UEP1_TX_ST	0x400340C3	端点 1 状态寄存器	0x00
R8_UEP1_TX_CHAIN_CR	0x400340C4	端点 1 CHAIN 控制寄存器	0x00
R8_UEP1_TX_CHAIN_ST	0x400340C5	端点 1 CHAIN 状态寄存器	0x00
R16_UEP1_TX_CHAIN_LEN	0x400340C6	端点 1 CHAIN 发送最后一包长度	0x00
R8_UEP1_TX_CHAIN_EXP_NUMP	0x400340C8	端点 1 期望发送 NUMP 个数	0x00
R8_UEP1_TX_CHAIN_NUMP	0x400340C9	端点 1 已经发送 NUMP 个数	0x00
R16_UEP1_TX_DMA_OFS	0x400340CA	端点 1 的 DMA 偏移长度	0x00
R32_UEP1_TX_DMA	0x400340CC	端点 1 的 DMA 起始地址	0x0400

表 27-4 端点 2 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP2_TX_CFG	0x400340E0	端点 2 配置寄存器	0x86
R8_UEP2_TX_CR	0x400340E1	端点 2 控制寄存器	0x10
R8_UEP2_TX_SEQ	0x400340E2	端点 2 序列号寄存器	0x00
R8_UEP2_TX_ST	0x400340E3	端点 2 状态寄存器	0x00
R8_UEP2_TX_CHAIN_CR	0x400340E4	端点 2 CHAIN 控制寄存器	0x00
R8_UEP2_TX_CHAIN_ST	0x400340E5	端点 2 CHAIN 状态寄存器	0x00
R16_UEP2_TX_CHAIN_LEN	0x400340E6	端点 2 CHAIN 发送最后一包长度	0x00
R8_UEP2_TX_CHAIN_EXP_NUMP	0x400340E8	端点 2 期望发送 NUMP 个数	0x00
R8_UEP2_TX_CHAIN_NUMP	0x400340E9	端点 2 已经发送 NUMP 个数	0x00
R16_UEP2_TX_DMA_OFS	0x400340EA	端点 2 的 DMA 偏移长度	0x00
R32_UEP2_TX_DMA	0x400340EC	端点 2 的 DMA 起始地址	0x0400

表 27-5 端点 3 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP3_TX_CFG	0x40034100	端点 3 配置寄存器	0x86
R8_UEP3_TX_CR	0x40034101	端点 3 控制寄存器	0x10
R8_UEP3_TX_SEQ	0x40034102	端点 3 序列号寄存器	0x00
R8_UEP3_TX_ST	0x40034103	端点 3 状态寄存器	0x00
R8_UEP3_TX_CHAIN_CR	0x40034104	端点 3 CHAIN 控制寄存器	0x00
R8_UEP3_TX_CHAIN_ST	0x40034105	端点 3 CHAIN 状态寄存器	0x00
R16_UEP3_TX_CHAIN_LEN	0x40034106	端点 3 CHAIN 发送最后一包长度	0x00
R8_UEP3_TX_CHAIN_EXP_NUMP	0x40034108	端点 3 期望发送 NUMP 个数	0x00
R8_UEP3_TX_CHAIN_NUMP	0x40034109	端点 3 已经发送 NUMP 个数	0x00
R16_UEP3_TX_DMA_OFS	0x4003410A	端点 3 的 DMA 偏移长度	0x00
R32_UEP3_TX_DMA	0x4003410C	端点 3 的 DMA 起始地址	0x0400

表 27-6 端点 4 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP4_TX_CFG	0x40034120	端点 4 配置寄存器	0x86
R8_UEP4_TX_CR	0x40034121	端点 4 控制寄存器	0x10
R8_UEP4_TX_SEQ	0x40034122	端点 4 序列号寄存器	0x00
R8_UEP4_TX_ST	0x40034123	端点 4 状态寄存器	0x00

R8_UEP4_TX_CHAIN_CR	0x40034124	端点 4 CHAIN 控制寄存器	0x00
R8_UEP4_TX_CHAIN_ST	0x40034125	端点 4 CHAIN 状态寄存器	0x00
R16_UEP4_TX_CHAIN_LEN	0x40034126	端点 4 CHAIN 发送最后一包长度	0x00
R8_UEP4_TX_CHAIN_EXP_NUMP	0x40034128	端点 4 期望发送 NUMP 个数	0x00
R8_UEP4_TX_CHAIN_NUMP	0x40034129	端点 4 已经发送 NUMP 个数	0x00
R16_UEP4_TX_DMA_OFS	0x4003412A	端点 4 的 DMA 偏移长度	0x00
R32_UEP4_TX_DMA	0x4003412C	端点 4 的 DMA 起始地址	0x0400

表 27-7 端点 5 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP5_TX_CFG	0x40034140	端点 5 配置寄存器	0x86
R8_UEP5_TX_CR	0x40034141	端点 5 控制寄存器	0x10
R8_UEP5_TX_SEQ	0x40034142	端点 5 序列号寄存器	0x00
R8_UEP5_TX_ST	0x40034143	端点 5 状态寄存器	0x00
R8_UEP5_TX_CHAIN_CR	0x40034144	端点 5 CHAIN 控制寄存器	0x00
R8_UEP5_TX_CHAIN_ST	0x40034145	端点 5 CHAIN 状态寄存器	0x00
R16_UEP5_TX_CHAIN_LEN	0x40034146	端点 5 CHAIN 发送最后一包长度	0x00
R8_UEP5_TX_CHAIN_EXP_NUMP	0x40034148	端点 5 期望发送 NUMP 个数	0x00
R8_UEP5_TX_CHAIN_NUMP	0x40034149	端点 5 已经发送 NUMP 个数	0x00
R16_UEP5_TX_DMA_OFS	0x4003414A	端点 5 的 DMA 偏移长度	0x00
R32_UEP5_TX_DMA	0x4003414C	端点 5 的 DMA 起始地址	0x0400

表 27-8 端点 6 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP6_TX_CFG	0x40034160	端点 6 配置寄存器	0x86
R8_UEP6_TX_CR	0x40034161	端点 6 控制寄存器	0x10
R8_UEP6_TX_SEQ	0x40034162	端点 6 序列号寄存器	0x00
R8_UEP6_TX_ST	0x40034163	端点 6 状态寄存器	0x00
R8_UEP6_TX_CHAIN_CR	0x40034164	端点 6 CHAIN 控制寄存器	0x00
R8_UEP6_TX_CHAIN_ST	0x40034165	端点 6 CHAIN 状态寄存器	0x00
R16_UEP6_TX_CHAIN_LEN	0x40034166	端点 6 CHAIN 发送最后一包长度	0x00
R8_UEP6_TX_CHAIN_EXP_NUMP	0x40034168	端点 6 期望发送 NUMP 个数	0x00
R8_UEP6_TX_CHAIN_NUMP	0x40034169	端点 6 已经发送 NUMP 个数	0x00
R16_UEP6_TX_DMA_OFS	0x4003416A	端点 6 的 DMA 偏移长度	0x00
R32_UEP6_TX_DMA	0x4003416C	端点 6 的 DMA 起始地址	0x0400

表 27-9 端点 7 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP7_TX_CFG	0x40034180	端点 7 配置寄存器	0x86
R8_UEP7_TX_CR	0x40034181	端点 7 控制寄存器	0x10
R8_UEP7_TX_SEQ	0x40034182	端点 7 序列号寄存器	0x00
R8_UEP7_TX_ST	0x40034183	端点 7 状态寄存器	0x00
R8_UEP7_TX_CHAIN_CR	0x40034184	端点 7 CHAIN 控制寄存器	0x00
R8_UEP7_TX_CHAIN_ST	0x40034185	端点 7 CHAIN 状态寄存器	0x00

R16_UEP7_TX_CHAIN_LEN	0x40034186	端点 7 CHAIN 发送最后一包长度	0x00
R8_UEP7_TX_CHAIN_EXP_NUMP	0x40034188	端点 7 期望发送 NUMP 个数	0x00
R8_UEP7_TX_CHAIN_NUMP	0x40034189	端点 7 已经发送 NUMP 个数	0x00
R16_UEP7_TX_DMA_OFS	0x4003418A	端点 7 的 DMA 偏移长度	0x00
R32_UEP7_TX_DMA	0x4003418C	端点 7 的 DMA 起始地址	0x0400

27.2.3.1 端点 n 配置寄存器 (R8_UEPn_TX_CFG) (n=1-7)

偏移地址: $0xC0 + (n-1) * 0x20$

位	名称	访问	描述	复位值
7	RB_EP_TX_CHAIN_AUTO	RW	CHAIN 自动切换模式, 推荐使用该模式。	1
6	RB_EP_TX_FIFO_MODE	RW	该位为 1 表示当前端点使用 FIFO 模式, 在该位为 1 前, 应先配置好 FIFO 的起始地址和结束地址。	0
5	RB_EP_TX_FIFO_CFG	RW	1: 访问偏移地址 $0xC \sim 0xF$, 操作对象为 UEP_DMA[31:16]/UEP_DMA[15:0]; 0: 访问偏移地址 $0xC \sim 0xF$, 操作对象为 UEP_DMA。	0
4	Reserved	RO	保留。	0
3	RB_EP_TX_EOB_MODE	RW	1: 发送短包, 则 ACK-TP 中 EOB/LPF=0; 0: 发送短包, 则 ACK-TP 中 EOB/LPF=1; 注: CHAIN_ST 的 EOB_LPF 与该模式置位的 EOB/LPF 是或的关系; 该位只是针对短包, 而 CHAIN_ST→EOB_LPF 针对于该 CHAIN 的最后一包, 与包长无关。	0
2	RB_EP_TX_ERDY_AUTO	RW	ERDY 自动模式, 硬件会发送 ERDY, 无需软件控制; 同步端点建议清零该位。	1
1	RB_EP_TX_SEQ_AUTO	RW	1: 禁止软件写 R16_EPn_ST→EP_SEQ_NUM; 0: 允许软件写 R16_EPn_ST→EP_SEQ_NUM。	1
0	RB_EP_TX_ISO_MODE	RW	为 1 表示当前端点为同步端点。	0

27.2.3.2 端点 n 控制寄存器 (R8_UEPn_TX_CR) (n=1-7)

偏移地址: $0xC1 + (n-1) * 0x20$

位	名称	访问	描述	复位值
7	RB_EP_TX_HALT	RW	端点停止, 高有效, 端点停止对 DPH 应答 STALL。	0
6	RB_EP_TX_CLR	RW1	写 1 清除端点所有配置值及状态, 除了 UEP_CFG。	0
5	RB_EP_TX_CHAIN_CLR	RW1	写 1 清除所有 CHAIN 配置值及状态。	0
[4:0]	RB_EP_TX_ERDY_NUMP	RW	硬件发送 ERDY-TP 的 nump 字段, 一般配置值为支持的突发个数; 例如支持 16 级突发, 该字段配置值为 16。	0x10

27.2.3.3 端点 n 序列号寄存器 (R8_UEPn_TX_SEQ) (n=1-7)偏移地址: $0xC2 + (n-1) * 0x20$

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
[4:0]	RB_TX_EP_SEQ_NUM	RW	端点当前序列号, 在非 SEQ_AUTO 模式下可写, SEQ_AUTO 模式下只读。	0

27.2.3.4 端点 n 状态寄存器 (R8_UEPn_TX_ST) (n=1-7)偏移地址: $0xC3 + (n-1) * 0x20$

位	名称	访问	描述	复位值
7	RB_TX_EP_INT_FLAG	RO	端点当前中断标志, 该位只读, 所有的 CHAIN_IF 为 0 则该位为 0, 否则为 1。	0
6	RB_TX_EP_FC_ST	RW1Z	端点当前流控状态, 写 1 清零。	0
5	RB_TX_EP_ERDY_REQ	RW1	表示当前处于发送 ERDY 中; 对该位写 1 将发送 ERDY, 在非 ERDY_AUTO 模式下使用。	0
4	RB_TX_CHAIN_RES	RO	CHAIN 应答状态, 对应于 4 个独立的 CHAIN: 1: 应答 ACK_TP; 0: 应答 NRDY-TP。 注: 端点停止模式下 ($EP\_HALT=1$), 应答 STALL, 该位不起作用。	0
[3:0]	RB_TX_CHAIN_EN	RO	CHAIN 使能状态, 对应于 4 个独立的 CHAIN。	0

27.2.3.5 端点 n CHAIN 控制寄存器 (R8_UEPn_TX_CHAIN_CR) (n=1-7)偏移地址: $0xC4 + (n-1) * 0x20$

位	名称	访问	描述	复位值
[7:6]	RB_TX_CUR_USE	RO	当前使用的 CHAIN 序列号。	0
[5:4]	RB_TX_CUR_CFG	RO	当前配置的 CHAIN 序列号。	0
3	Reserved	RO	保留。	0
2	RB_TX_FORCE_RET	RW	该位有效强制返回选中的 CHAIN 状态机配置。	0
[1:0]	RB_TX_RET_SEL	RW	当 FORCE_RET 有效时, 该位表示返回的 CHAIN 状态及配置。	0

27.2.3.6 端点 n CHAIN 状态寄存器 (R8_UEPn_TX_CHAIN_ST) (n=1-7)偏移地址: $0xC5 + (n-1) * 0x20$

位	名称	访问	描述	复位值
7	RB_TX_CHAIN_EN	RO	当前使用的 CHAIN 使能, 在配置完 UEP_CHAIN_NUMP 寄存器后自动硬件自动置 1, 该 CHAIN 传输结束后自动清零。	0
6	RB_TX_CHAIN_IF	WO	该位只写, 写 1 释放当前 CHAIN_IF; 当前 CHAIN 的 NUMP_EMPTY 为 1 产生该中断。	0
5	RB_TX_EOB_LPF	RW	当前 CHAIN 的最后一包 DPH 中 EOB/LPF 位: 对于批量传输, 该位置 1 后, CHAIN 传输结束后进入流控(End Of Block), 不会和下一个	0

			CHAIN 链接到一起发送； 对于同步传输，该位置 1 后，CHAIN 传输结束后，该微帧不在发送 DPH，表示最后一包为此微帧最后一包（LAST PACKET FLAG）。	
4	Reserved	RO	保留。	0
3	RB_TX_NUMP_EMPTY	RO	当前 CHAIN 中 NUMP 为 0，则该位置 1。	0
2	RB_TX_DPH_PP	RW	当前收到的 DPH 中 PP 位状态。	0
[1:0]	RB_TX_CHAIN_NO	RW	当前产生中断的 CHAIN 序列号。	0

27.2.3.7 端点 n CHAIN 发送最后一包长度（R16_UEPn_TX_CHAIN_LEN）（n=1-7）

偏移地址：0xC6 + (n-1) * 0x20

位	名称	访问	描述	复位值
[15:11]	Reserved	RO	保留。	0
[10:0]	CHAIN_TX_LEN	RW	该 CHAIN 发送最后一包的长度。	0

27.2.3.8 端点 n 期望发送 NUMP 个数（R8_UEPn_TX_CHAIN_EXP_NUMP）（n=1-7）

偏移地址：0xC8 + (n-1) * 0x20

位	名称	访问	描述	复位值
[7:0]	TX_CHAIN_EXP_NUMP	RW	当前完成的 CHAIN 可发送的 DPP 包个数。	0

27.2.3.9 端点 n 已经发送 NUMP 个数（R8_UEPn_TX_CHAIN_NUMP）（n=1-7）

偏移地址：0xC9 + (n-1) * 0x20

位	名称	访问	描述	复位值
[7:0]	TX_CHAIN_NUMP	RW	当前已经传输 DPP 包个数；对在 CHIN_EN 为 0 时，写 UEP_CHAIN_NUMP 同时该寄存器会自动清零。	0

27.2.3.10 端点 n 的 DMA 偏移长度（R16_UEPn_TX_DMA_OFS）（n=1-7）

偏移地址：0xCA + (n-1) * 0x20

位	名称	访问	描述	复位值
[15:0]	CHAIN_TX_DMA_OFS	RW	该 CHAIN 中 DPP 的偏移地址。	0x0400

27.2.3.11 端点 n 的 DMA 起始地址（R32_UEPn_TX_DMA）（n=1-7）

偏移地址：0xCC + (n-1) * 0x20

位	名称	访问	描述	复位值
[31:0]	CHAIN_TX_DMA	RW	普通模式下： 该 CHAIN 发送数据的 DMA 起始地址。 FIFO 模式下： FIFO 起始地址在 SRAM 中的映射地址的 16~23 位； FIFO 结束地址在 SRAM 中的映射地址的 16~23 位。	0

27.2.4 设备寄存器--端点 1~7 接收相关寄存器

表 27-10 端点 1 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP1_RX_CFG	0x400340D0	端点 1 配置寄存器	0x86
R8_UEP1_RX_CR	0x400340D1	端点 1 控制寄存器	0x10
R8_UEP1_RX_SEQ	0x400340D2	端点 1 序列号寄存器	0x00
R8_UEP1_RX_ST	0x400340D3	端点 1 状态寄存器	0x00
R8_UEP1_RX_CHAIN_CR	0x400340D4	端点 1 CHAIN 控制寄存器	0x00
R8_UEP1_RX_CHAIN_ST	0x400340D5	端点 1 CHAIN 状态寄存器	0x00
R16_UEP1_RX_CHAIN_LEN	0x400340D6	端点 1 CHAIN 接收到的最后一包长度	0x00
R8_UEP1_RX_CHAIN_MAX_NUMP	0x400340D8	端点 1 可接收 NUMP 个数	0x00
R8_UEP1_RX_CHAIN_NUMP	0x400340D9	端点 1 已经接收到的 NUMP 个数	0x00
R16_UEP1_RX_DMA_OFS	0x400340DA	端点 1 的 DMA 偏移长度	0x0400
R32_UEP1_RX_DMA	0x400340DC	端点 1 的 DMA 起始地址	0x00

表 27-11 端点 2 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP2_RX_CFG	0x400340F0	端点 2 配置寄存器	0x86
R8_UEP2_RX_CR	0x400340F1	端点 2 控制寄存器	0x10
R8_UEP2_RX_SEQ	0x400340F2	端点 2 序列号寄存器	0x00
R8_UEP2_RX_ST	0x400340F3	端点 2 状态寄存器	0x00
R8_UEP2_RX_CHAIN_CR	0x400340F4	端点 2 CHAIN 控制寄存器	0x00
R8_UEP2_RX_CHAIN_ST	0x400340F5	端点 2 CHAIN 状态寄存器	0x00
R16_UEP2_RX_CHAIN_LEN	0x400340F6	端点 2 CHAIN 接收到的最后一包长度	0x00
R8_UEP2_RX_CHAIN_MAX_NUMP	0x400340F8	端点 2 可接收 NUMP 个数	0x00
R8_UEP2_RX_CHAIN_NUMP	0x400340F9	端点 2 已经接收到的 NUMP 个数	0x00
R16_UEP2_RX_DMA_OFS	0x400340FA	端点 2 的 DMA 偏移长度	0x0400
R32_UEP2_RX_DMA	0x400340FC	端点 2 的 DMA 起始地址	0x00

表 27-12 端点 3 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP3_RX_CFG	0x40034110	端点 3 配置寄存器	0x86
R8_UEP3_RX_CR	0x40034111	端点 3 控制寄存器	0x10
R8_UEP3_RX_SEQ	0x40034112	端点 3 序列号寄存器	0x00
R8_UEP3_RX_ST	0x40034113	端点 3 状态寄存器	0x00
R8_UEP3_RX_CHAIN_CR	0x40034114	端点 3 CHAIN 控制寄存器	0x00
R8_UEP3_RX_CHAIN_ST	0x40034115	端点 3 CHAIN 状态寄存器	0x00
R16_UEP3_RX_CHAIN_LEN	0x40034116	端点 3 CHAIN 接收到的最后一包长度	0x00
R8_UEP3_RX_CHAIN_MAX_NUMP	0x40034118	端点 3 可接收 NUMP 个数	0x00
R8_UEP3_RX_CHAIN_NUMP	0x40034119	端点 3 已经接收到的 NUMP 个数	0x00
R16_UEP3_RX_DMA_OFS	0x4003411A	端点 3 的 DMA 偏移长度	0x0400

R32_UEP3_RX_DMA	0x4003411C	端点 3 的 DMA 起始地址	0x00

表 27-13 端点 4 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP4_RX_CFG	0x40034130	端点 4 配置寄存器	0x86
R8_UEP4_RX_CR	0x40034131	端点 4 控制寄存器	0x10
R8_UEP4_RX_SEQ	0x40034132	端点 4 序列号寄存器	0x00
R8_UEP4_RX_ST	0x40034133	端点 4 状态寄存器	0x00
R8_UEP4_RX_CHAIN_CR	0x40034134	端点 4 CHAIN 控制寄存器	0x00
R8_UEP4_RX_CHAIN_ST	0x40034135	端点 4 CHAIN 状态寄存器	0x00
R16_UEP4_RX_CHAIN_LEN	0x40034136	端点 4 CHAIN 接收到的最后一包长度	0x00
R8_UEP4_RX_CHAIN_MAX_NUMP	0x40034138	端点 4 可接收 NUMP 个数	0x00
R8_UEP4_RX_CHAIN_NUMP	0x40034139	端点 4 已经接收到的 NUMP 个数	0x00
R16_UEP4_RX_DMA_OFS	0x4003413A	端点 4 的 DMA 偏移长度	0x0400
R32_UEP4_RX_DMA	0x4003413C	端点 4 的 DMA 起始地址	0x00

表 27-14 端点 5 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP5_RX_CFG	0x40034150	端点 5 配置寄存器	0x86
R8_UEP5_RX_CR	0x40034151	端点 5 控制寄存器	0x10
R8_UEP5_RX_SEQ	0x40034152	端点 5 序列号寄存器	0x00
R8_UEP5_RX_ST	0x40034153	端点 5 状态寄存器	0x00
R8_UEP5_RX_CHAIN_CR	0x40034154	端点 5 CHAIN 控制寄存器	0x00
R8_UEP5_RX_CHAIN_ST	0x40034155	端点 5 CHAIN 状态寄存器	0x00
R16_UEP5_RX_CHAIN_LEN	0x40034156	端点 5 CHAIN 接收到的最后一包长度	0x00
R8_UEP5_RX_CHAIN_MAX_NUMP	0x40034158	端点 5 可接收 NUMP 个数	0x00
R8_UEP5_RX_CHAIN_NUMP	0x40034159	端点 5 已经接收到的 NUMP 个数	0x00
R16_UEP5_RX_DMA_OFS	0x4003415A	端点 5 的 DMA 偏移长度	0x0400
R32_UEP5_RX_DMA	0x4003415C	端点 5 的 DMA 起始地址	0x00

表 27-15 端点 6 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP6_RX_CFG	0x40034170	端点 6 配置寄存器	0x86
R8_UEP6_RX_CR	0x40034171	端点 6 控制寄存器	0x10
R8_UEP6_RX_SEQ	0x40034172	端点 6 序列号寄存器	0x00
R8_UEP6_RX_ST	0x40034173	端点 6 状态寄存器	0x00
R8_UEP6_RX_CHAIN_CR	0x40034174	端点 6 CHAIN 控制寄存器	0x00
R8_UEP6_RX_CHAIN_ST	0x40034175	端点 6 CHAIN 状态寄存器	0x00
R16_UEP6_RX_CHAIN_LEN	0x40034176	端点 6 CHAIN 接收到的最后一包长度	0x00
R8_UEP6_RX_CHAIN_MAX_NUMP	0x40034178	端点 6 可接收 NUMP 个数	0x00
R8_UEP6_RX_CHAIN_NUMP	0x40034179	端点 6 已经接收到的 NUMP 个数	0x00

R16_UEP6_RX_DMA_OFS	0x4003417A	端点 6 的 DMA 偏移长度	0x0400
R32_UEP6_RX_DMA	0x4003417C	端点 6 的 DMA 起始地址	0x00

表 27-16 端点 7 接收相关寄存器列表

名称	访问地址	描述	复位值
R8_UEP7_RX_CFG	0x40034190	端点 7 配置寄存器	0x86
R8_UEP7_RX_CR	0x40034191	端点 7 控制寄存器	0x10
R8_UEP7_RX_SEQ	0x40034192	端点 7 序列号寄存器	0x00
R8_UEP7_RX_ST	0x40034193	端点 7 状态寄存器	0x00
R8_UEP7_RX_CHAIN_CR	0x40034194	端点 7 CHAIN 控制寄存器	0x00
R8_UEP7_RX_CHAIN_ST	0x40034195	端点 7 CHAIN 状态寄存器	0x00
R16_UEP7_RX_CHAIN_LEN	0x40034196	端点 7 CHAIN 接收到的最后一包长度	0x00
R8_UEP7_RX_CHAIN_MAX_NUMP	0x40034198	端点 7 可接收 NUMP 个数	0x00
R8_UEP7_RX_CHAIN_NUMP	0x40034199	端点 7 已经接收到的 NUMP 个数	0x00
R16_UEP7_RX_DMA_OFS	0x4003419A	端点 7 的 DMA 偏移长度	0x0400
R32_UEP7_RX_DMA	0x4003419C	端点 7 的 DMA 起始地址	0x00

27.2.4.1 端点 n 配置寄存器 (R8_UEPn_RX_CFG) (n=1-7)

偏移地址: $0xD0 + (n-1) * 0x20$

位	名称	访问	描述	复位值
7	RB_EP_RX_CHAIN_AUTO	RW	CHAIN 自动切换模式, 推荐使用该模式。	1
6	RB_EP_RX_FIFO_MODE	RW	该位为 1 表示当前端点使用 FIFO 模式, 在该位为 1 前, 应先配置好 FIFO 的起始地址和结束地址。	0
5	RB_EP_RX_FIFO_CFG	RW	1: 访问偏移地址 $0xC \sim 0xF$, 操作对象为 UEP_DMA; 0: 访问偏移地址 $0xC \sim 0xF$, 操作对象为 R16_EPn_FIFO_START/R16_EPn_FIFO_DEST。	0
4	RB_EP_RX_TOUT_MODE	RW	该位为 1, 当接收连续突发包之后, PP=1 且不是短包, 超时未收到 DPH, 产生 TOUT_IF 中断。	0
3	RB_EP_RX_NRDY_MODE	RW	该位为 1, 当接收到短包或 DPH (PP=0), 清除所有 CHAIN_EN, 将导致对后面的 DPH 应答 NRDY; 需要软件重新配置。	0
2	RB_EP_RX_ERDY_AUTO	RW	ERDY 自动模式, 硬件会发送 ERDY, 无需软件控制; 同步端点建议清零该位;	1
1	RB_EP_RX_SEQ_AUTO	RW	1: 禁止软件写 R16_EPn_ST→EP_SEQ_NUM; 0: 允许软件写 R16_EPn_ST→EP_SEQ_NUM。	1
0	RB_EP_RX_ISO_MODE	RW	为 1 表示当前端点为同步端点	0

27.2.4.2 端点 n 控制寄存器 (R8_UEPn_RX_CR) (n=1-7)

偏移地址: $0xD1 + (n-1) * 0x20$

位	名称	访问	描述	复位值
7	RB_EP_RX_HALT	RW	端点停止，高有效，端点停止对 DPH 应答 STALL。	0
6	RB_EP_RX_CLR	RW1Z	写 1 清除端点所有配置值及状态，除了 UEP_CFG。	0
5	RB_EP_RX_CHAIN_CLR	RW1Z	写 1 清除所有 CHAIN 配置值及状态。	0
[4:0]	RB_EP_RX_ERDY_NUMP	RW	硬件发送 ERDY-TP 的 nump 字段，一般配置值为支持的突发个数；例如支持 16 级突发，该字段配置值为 16。	0x10

27.2.4.3 端点 n 序列号寄存器 (R8_UEPn_RX_SEQ) (n=1-7)

偏移地址：0xD2 + (n-1) * 0x20

位	名称	访问	描述	复位值
[7:5]	Reserved	RO	保留。	0
[4:0]	RB_EP_RX_SEQ_NUM	RW	端点当前序列号，在非 SEQ_AUTO 模式下可写，SEQ_AUTO 模式下只读。 注：一般情况下，该字段由硬件自动控制，软件无需写 EP_SEQ_NUM。	0

27.2.4.4 端点 n 状态寄存器 (R8_UEPn_RX_ST) (n=1-7)

偏移地址：0xD3 + (n-1) * 0x20

位	名称	访问	描述	复位值
7	RB_EP_RX_INT_FLAG	RO	端点当前中断标志，该位只读，所有的 CHAIN_IF 为 0 则该位为 0，否则为 1。	0
6	RB_EP_RX_FC_ST	RW1Z	端点当前流控状态，写 1 清零。	0
5	RB_EP_RX_ERDY_REQ	RW1	表示当前处于发送 ERDY 中； 对该位写 1 将发送 ERDY，在非 ERDY_AUTO 模式下使用。	0
4	RB_EP_RX_CHAIN_RES	RO	CHAIN 应答状态，对应于 4 个独立的 CHAIN： 1：应答 ACK_TP； 0：应答 NRDY-TP。 注：端点停止模式下 (EP_HALT=1)，应答 STALL，该位不起作用。	0
[3:0]	RB_EP_RX_CHAIN_EN	RO	CHAIN 使能状态，对应于 4 个独立的 CHAIN。	0

27.2.4.5 端点 n CHAIN 控制寄存器 (R8_UEPn_RX_CHAIN_CR) (n=1-7)

偏移地址：0xD4 + (n-1) * 0x20

位	名称	访问	描述	复位值
[7:6]	RB_EP_RX_CUR_USE	RO	当前使用的 CHAIN 序列号。	0
[5:4]	RB_EP_RX_CUR_CFG	RO	当前配置的 CHAIN 序列号。	0
3	Reserved	RO	保留。	0
2	RB_EP_RX_FORCE_RET	RW	该位有效强制返回选中的 CHAIN 状态机配置。	0

[1:0]	RB_EP_RX_RET_SEL	RW	当 FORCE_RET 有效时,该位表示返回的 CHAIN 状态及配置。	0

27.2.4.6 端点 n CHAIN 状态寄存器 (R8_UEPn_RX_CHAIN_ST) (n=1-7)

偏移地址: $0xD5 + (n-1) * 0x20$

位	名称	访问	描述	复位值
7	RB_EP_RX_CHAIN_EN	RO	当前使用的 CHAIN 使能, 在配置完 UEP_CHAIN_NUMP 寄存器后自动硬件自动置 1, 该 CHAIN 传输结束后自动清零。	0
6	RB_EP_RX_CHAIN_IF	RW1	该位只写, 写 1 释放当前 CHAIN_IF; 满足一下任一条件, 则产生该中断: 1 当前接收的包长度小于 1024; 2 非同步端点接收的 DPH 中 PP 为 0; 3 同步端点接收的 DPH 中 LPF 为 1; 4 当前 CHAIN 的 NUMP_EMPTY 为 1; 5 当 DPH 完成之后 4us 未收到 DPH 产生中断 (可选)。 当收到短包, 可选硬件自动 NRDY 或关闭所有 chain; 当收到 PP=0, 可选硬件自动 NRDY 或关闭所有 chain; 超时未收到 DPP, 可选硬件自动 NRDY 或关闭所有 chain。 同步模式如果没有收到预期的包个数(crc 错误)	0
5	RB_EP_RX_LPF_FLAG	RO	仅同步下传端点使用, 当前接收的 DPH 中 LPF 状态。	0
4	RB_EP_RX_ISO_PKT_ERR	RO	仅同步下传端点使用, 当前接收的 DPP 中有 CRC32 错误。	0
3	RB_EP_RX_NUMP_EMPTY	RO	当前 CHAIN 中 NUMP 为 0, 则该位置 1。	0
2	RB_EP_RX_DPH_PP	RO	当前收到的 DPH 中 PP 位状态。	0
[1:0]	RB_EP_RX_CHAIN_NO	RO	当前产生中断的 CHAIN 序列号。	0

27.2.4.7 端点 n CHAIN 接收到的最后一包长度 (R16_UEPn_RX_CHAIN_LEN) (n=1-7)

偏移地址: $0xD6 + (n-1) * 0x20$

位	名称	访问	描述	复位值
[15:11]	Reserved	RO	保留。	0
[10:0]	RB_EP_RX_CHAIN_RX_LEN	RO	当前完成的 CHAIN 接收到最后一包的长度。	0

27.2.4.8 端点 n 可接收 NUMP 个数 (R8_UEPn_RX_CHAIN_MAX_NUMP) (n=1-7)

偏移地址: $0xD8 + (n-1) * 0x20$

位	名称	访问	描述	复位值
[7:0]	RX_CHAIN_MAX_NUM P	RW	该 CHAIN 可接收的 DPP 包个数。	0

27.2.4.9 端点 n 已经接收到的 NUMP 个数 (R8_UEPn_RX_CHAIN_NUMP) (n=1-7)

偏移地址: $0xD9 + (n-1) * 0x20$

位	名称	访问	描述	复位值
[7:0]	RX_CHAIN_NUMP	RO	该 CHAIN 已经接收到的 DPP 包个数。	0

27.2.4.10 端点 n 的 DMA 偏移长度 (R16_UEPn_RX_DMA_OFS) (n=1-7)

偏移地址: $0xDA + (n-1) * 0x20$

位	名称	访问	描述	复位值
[15:0]	CHAIN_DMA_OFS	RW	该 CHAIN 中 DPP 的偏移地址。	0x0400

27.2.4.11 端点 n 的 DMA 起始地址 (R32_UEPn_RX_DMA) (n=1-7)

偏移地址: $0xDC + (n-1) * 0x20$

位	名称	访问	描述	复位值
[31:0]	CHAIN_DMA_BA	RW	普通模式下: 该 CHAIN 接收数据的 DMA 起始地址。 FIFO 模式下: FIFO 起始地址在 SRAM 中的映射地址的 16~23 位; FIFO 结束地址在 SRAM 中的映射地址的 16~23 位。	0

27.2.5 主机寄存器描述

表 27-17 USBSS 主机寄存器列表

名称	访问地址	描述	复位值
R32_UH_TX_CTRL	0x40034084	主机发送控制寄存器	0x0000XXXX
R32_UH_RX_CTRL	0x40034088	主机接收控制寄存器	0xXX00XXXX
R32_UH_TX_DMA_OFS	0x40034094	主机发送地址偏移寄存器	0x00000400
R32_UH_RX_DMA_OFS	0x40034098	主机接收地址偏移寄存器	0x00000400
R16_HOST_TX_NUMP	0x4003409C	主机发送 NUMP 寄存器	0x0000
R16_HOST_RX_NUMP	0x4003409E	主机接收 NUMP 寄存器	0xXX00
R32_HOST_STATUS	0x400340A0	主机状态寄存器	0x00000100
R16_HOST_TX_FC_STATUS	0x400340A4	主机端点发送流控寄存器	0x0000
R16_HOST_RX_FC_STATUS	0x400340A6	主机端点接收流控寄存器	0x0000
R32_TP_RX_DATA0	0x400340A8	DEV_NOTIFICATION 的 DATA0 寄存器	0xFFFFFFFF
R32_TP_RX_DATA1	0x400340AC	DEV_NOTIFICATION 的 DATA1 寄存器	0xFFFFFFFF
R32_TP_RX_DATA2	0x400340B0	DEV_NOTIFICATION 的 DATA2 寄存器	0xFFFFFFFF

27.2.5.1 主机发送控制寄存器 (R32_UH_TX_CTRL)

偏移地址: 0x84

位	名称	访问	描述	复位值

31	RB_UH_TX_ACT	RW	IN 事务完成中断标志，软件写 0 清零，由硬件置 1。	0
30	RB_UH_TX_ISO	RW	主机准备发送 ISO 数据包。	0
29	RB_UH_TX_SETUP	RW	表示主机发送的数据包为 Setup 数据包，设置 setup 标志。	0
28	RB_UH_TX_STATUS	RW	表示主机发送的包为 STATUS TP。	0
[27:24]	Reserved	RO	保留。	0
23	RB_UH_TX_LPF	RW	对于突发传输，该位只是表示最后一个包的 LPF/EOB，前面的包 LPF/EOB 使用固定值 0。 对于同步端点有如下定义：Last Packet Flag--IN 事务中返回 DPH 包中的 LPF，该位为 1 表示该包为此帧间隔中最后一包。 对于非同步端点有如下定义：End Of Block--IN 事务中返回 DPH 包中的 EOB，该位为 1 时该端点进入流控状态，主机等待设备的该端点接收 ERDY-TP 后重新开始 IN 事务。	0
[22:21]	RB_UH_TX_RES	RW	对 DPH+DPP 的应答： 00: NRDY-TP； 01: ACK-TP； 10: STALL-TP； 11: 不应答（无效）。 SETUP/OUT 事务完成后 EPn_R_RES 自动清零 EPn_R_RES。	0
[20:16]	RB_UH_TX_SEQ	RW	端点接收数据包的 SEQ_NUM，硬件自动加 1，端点 0 除外： 对于同步端点，接收到 ITP 包后硬件自动清零，由于协议要求帧间隔的第一个包的 SEQ 为 0。	0
[15:12]	RB_UH_TX_EP	RW	指示主机模式下发送的包的目的地（设备的目标端点号）	x
11	Reserved	RO	保留	0
[10:0]	RB_UH_TX_LEN	RW	端点接收长度寄存器，对于突发传输，该长度表示突发传输最后一包的数据包长度，其他包按照协议规定必须为 1024B。	x

27.2.5.2 主机接收控制寄存器（R32_UH_RX_CTRL）

偏移地址：0x88

位	名称	访问	描述	复位值
31	RB_UH_RX_ACT	RW	OUT 事务完成中断标志，软件写 0 清零，由硬件置 1。	0
30	RB_UH_RX_ISO	RO	接收到数据包 (DPP) 是同步传输。	0
29	Reserved	RO	保留。	0
[28:24]	RB_UH_RX_NUMP	RW	端点能够接收的数据包 (DPP) 个数 (突发传输)。	x
23	Reserved	RO	保留。	0

[22:21]	RB_UH_RX_RES	RO	对 DPH+DPP 或 STATUS-TP 的应答： 00: NRDY-TP; 01: ACK-TP; 10: STALL-TP; 11: 不应答（无效）。 SETUP/OUT/STATUS 事务完成后自动清零。	0
[20:16]	RB_UH_RX_SEQ	RW	端点期望接收的 SEQ_NUM, 硬件自动加 1, 端点 0 除外。 对于同步端点, 接收到 ITP 包后硬件自动清零, 由于协议要求帧间隔的第一个包的 SEQ 为 0。	0
[15:12]	RB_UH_RX_EP	RW	指示主机模式下接收到包的源头（设备端点号）。	x
11	Reserved	RO	保留。	0
[10:0]	RB_UH_RX_LEN	RO	端点接收长度寄存器, 对于突发传输, 该长度表示突发传输最后一包的数据包长度, 其他包按照协议规定必须为 1024B。	x

27.2.5.3 主机发送地址偏移寄存器 (R32_UH_TX_DMA_OFS)

偏移地址: 0x94

位	名称	访问	描述	复位值
[31:0]	RB_UH_TX_DMA_OFS	RW	主机发送完成后, 将 DMA 的地址偏移的大小。	0x00000 400

27.2.5.4 主机接收地址偏移寄存器 (R32_UH_RX_DMA_OFS)

偏移地址: 0x98

位	名称	访问	描述	复位值
[31:0]	RB_UH_RX_DMA_OFS	RW	主机接收完成后, 将 DMA 的地址偏移的大小。	0x00000 400

27.2.5.5 主机发送 NUMP 寄存器 (R16_HOST_TX_NUMP)

偏移地址: 0x9C

位	名称	访问	描述	复位值
[15:8]	Reserved	RO	保留。	0
[7:0]	RB_UH_TX_NUMP	RW	主机期望发送的包数量, 每发送一个, 硬件自动减 1。	0

27.2.5.6 主机接收 NUMP 寄存器 (R16_HOST_RX_NUMP)

偏移地址: 0x9E

位	名称	访问	描述	复位值
[15:8]	RB_UH_RX_DPP_NUM	RW	主机已经接受的 DPP 的数量。	x
[7:0]	RB_UH_RX_NUMP	RW	主机期望接收的包数量, 如果是同步传输, 每接收一个, 硬件自动减 1。	0

27.2.5.7 主机状态寄存器 (R32_HOST_STATUS)

偏移地址: 0xA0

位	名称	访问	描述	复位值
[31:20]	Reserved	R0	保留。	0
[19:18]	RB_UH_1TP_PRESAGE	R0	主机模式下, 该位表示发送 ITP 包的时间 00: 9us 内主机不会发送 ITP 包; 01: 9us 内主机发送 ITP 包; 11: 6us 内主机发送 ITP 包; 10: 3us 内主机发送 ITP 包。	0
17	RB_UH_RX_ISO_PKT_ERR	R0	同步传输时, 接收到数据包(DPP)的 CRC 错误。	0
16	RB_UH_RX_EOB_LPF	R0	该位表示接收到的包中的 EOB/LPF 状态。 对于突发传输, 该位只是表示最后一个包的 LPF/EOB, 前面的包 LPF/EOB 使用固定值 0。 对于同步端点有如下定义: Last Packet Flag--IN 事务中返回 DPH 包中的 LPF, 该位为 1 表示该包为此帧间隔中最后一包。 对于非同步端点有如下定义: End Of Block--IN 事务中返回 DPH 包中的 EOB, 该位为 1 时该端点进入流控状态, 主机等待设备的该端点发送 ERDY-TP 后重新开始 IN 事务。	0
15	RB_UH_RX_ERDY_DIR	R0	接收到来自设备的 ERDY, 该段表示端点的方向。	0
[14:13]	Reserved	R0	保留。	0
[12:8]	RB_UH_RX_ERDY_NUM	R0	接收到来自设备的 ERDY, 该段表示设备对应端点可以发送/接收的数据包个数。	0x1
[7:4]	RB_UH_RX_ERDY_EP	R0	接收到来自设备的 ERDY, 该段表示设备 ERDY 的端点号。	0
[3:0]	Reserved	R0	保留。	0

27.2.5.8 主机端点发送流控寄存器 (R16_HOST_TX_FC_STATUS)

偏移地址: 0xA4

位	名称	访问	描述	复位值
[15:1]	EPx_TX_FC	RW1Z	主机对发送端点 1-15 的流控状态。	0
0	Reserved	R0	保留。	0

27.2.5.9 主机端点接收流控寄存器 (R16_HOST_RX_FC_STATUS)

偏移地址: 0xA6

位	名称	访问	描述	复位值
[15:1]	EPx_RX_FC	RW1Z	主机对接收端点 1-15 的流控状态。	0
0	Reserved	R0	保留。	0

27.2.5.10 DEV_NOTIF-TP 数据 0 寄存器 (R32_TP_RX_DATA0)

偏移地址: 0xA8

位	名称	访问	描述	复位值

[31:0]	USB3_NOTIF_DATA0	RO	DEV_NOTIF-TP 接收完成后, HP 数据存放该寄存器。	x

27.2.5.11 DEV_NOTIF-TP 数据 1 寄存器 (R32_TP_RX_DATA1)

偏移地址: 0xAC

位	名称	访问	描述	复位值
[31:0]	USB3_NOTIF_DATA1	RO	DEV_NOTIF-TP 接收完成后, HP 数据存放该寄存器。	x

27.2.5.12 DEV_NOTIF-TP 数据 2 寄存器 (R32_TP_RX_DATA2)

偏移地址: 0xB0

位	名称	访问	描述	复位值
[31:0]	USB3_NOTIF_DATA2	RO	DEV_NOTIF-TP 接收完成后, HP 数据存放该寄存器。	x

27.2.6 USBSS LINK 寄存器描述

表 27-18 USBSS LINK 寄存器列表

名称	访问地址	描述	复位值
R32_LINK_CFG	0x40034000	LINK 配置寄存器	0xC0000128
R32_LINK_CTRL	0x40034004	LINK 控制寄存器	0xFFFF0013
R32_LINK_INT_CTRL	0x40034008	LINK 中断使能寄存器	0x00000000
R32_LINK_INT_FLAG	0x4003400C	LINK 中断标志寄存器	0x00000000
R32_LINK_STATUS	0x40034010	LINK 状态寄存器	0x0000430
R8_LINK_ITP_PRE	0x40034017	LINK ITP 超时模式寄存器	0x00
R8_LINK_U2_INACT_TIME	0x4003401D	LINK U2 不活动超时计数器阈值寄存器	0x00
R8_LINK_U1_WKUP_FILTER	0x40034028	U1 唤醒 LFPS 时长配置寄存器	0x4A
R8_LINK_U2_WKUP_FILTER	0x4003402C	U2 唤醒 LFPS 时长配置寄存器	0x02
R8_LINK_U3_WKUP_FILTER	0x40034030	U3 唤醒 LFPS 时长配置寄存器	0x64
R16_LINK_ISO_DLY	0x40034040	LINK 同步延迟寄存器	0x0028
R16_LINK_LPM_CR	0x40034050	链路电源管理寄存器	0x0180
R32_LINK_LMP_PORT_CAP	0x40034054	PORT_CAP 寄存器	0xFFFF0000
R32_LINK_LMP_RX_DATA0	0x40034058	LMP 接收数据 0 寄存器	0xFFFFFFFF
R32_LINK_LMP_RX_DATA1	0x4003405C	LMP 接收数据 1 寄存器	0xFFFFFFFF
R32_LINK_LMP_RX_DATA2	0x40034060	LMP 接收数据 2 寄存器	0xFFFFFFFF
R32_LINK_LMP_TX_DATA0	0x40034064	USB 自定义 HP 数据 0 寄存器	0xFFFFFFFF
R32_LINK_LMP_TX_DATA1	0x40034068	USB 自定义 HP 数据 1 寄存器	0xFFFFFFFF
R32_LINK_LMP_TX_DATA2	0x4003406C	USB 自定义 HP 数据 2 寄存器	0xFFFFFFFF

27.2.6.1 LINK 配置寄存器 (R32_LINK_CFG)

偏移地址: 0x00

位	名称	访问	描述	复位值
31	RB_LINK_RESET	RW	LINK 复位, 包括复位状态机及所有中断标志, 高有效。	1

[30:22]	Reserved	RO	保留。	0
21	RB_LINK_TOUT_MODE	RW	1: 3.2 SPEC; 0: 3.0 SPEC。 注: 协议版本不同影响 flow control 中的计时器时长: PM_LC_TIMER: 3.2_SPEC: 4us/3.0_SPEC: 3us; PM_ENTRY_TIMER: 3.2_SPEC: 8us/3.0_SPEC: 6us; PEND_HP_TIMER: 3.2_SPEC: 10us/3.0_SPEC: 3us。	0
20	RB_LINK_U1_PING_EN	RW	U1 下发送 PING_FPFS 使能: 1: 发送; 0: 不发送。	0
[19:18]	RB_LINK_SKP_MODE	RW	用于解扰中的 SKP 剔除控制: 00: 低两字节中的低字节为 SKP, 关闭低两字节的解扰使能; 高两字节中的低字节为 SKP, 关闭高两字节的解扰使能; 01/11: 低两字节都为 SKP, 关闭低两字节的解扰使能; 高两字节都为 SKP, 关闭高两字节的解扰使能; 10: 低两字节中的任一字节为 SKP, 关闭低两字节的解扰使能; 高两字节中的任一字节为 SKP, 关闭高两字节的解扰使能。	0
17	RB_LINK_U2_ALLOW	RW	高有效, 在接收到 LGO_U2 后, 响应 LAU 允许进入 U2 状态, 否则在接收到 LGO_U2 后, 响应 LXU 拒绝进入 U2 状态。 注: 对于下行口来说, 该使能跟 U2_TIMEOUT 值有确定关系, U2_TIMEOUT>0, 配置 U2_ALLOW 有效, 反之无效。	0
16	RB_LINK_U1_ALLOW	RW	高有效, 在接收到 LGO_U1 后, 响应 LAU 允许进入 U1 状态, 否则在接收到 LGO_U1 后, 响应 LXU 拒绝进入 U1 状态。 注: 对于下行口来说, 该使能跟 U1_TIMEOUT 值有确定关系, U1_TIMEOUT>0, 配置 U1_ALLOW 有效, 反正无效。	0
15	RB_LINK_LTSSM_MODE	RW	链路状态机进入 DISABLE 模式: 1: 按照协议在超时、检测次数等情况下在进入 DISABLED; 0: 不按照协议在超时、检测次数等情况下在进入 DISABLED;	0
14	RB_LINK_LOOPBACK_ACT	RW	用于 LOOPBACK 模式中用于 LOOPBACK 主机控制码型发送 (LOOPBACK 从机保持 0), ACT 为持续一段时间的高电平信号。 1、LOOPBACK 主机需要在此位拉高时发送 BRST 至 LOOPBACK 从机复位 LOOPBACK 从机的错误计数器 (从机回环 BRST)。	0

			2、在 ACT 处于高电平时 LOOPBACK 主机发送 BDAT 至从机，从机比较接收到的 BDAT 和加扰编码的 0 是否相等进行错误计数（从机回环 BDAT）。 3、在 ACT 拉低时，LOOPBACK 主机发送 BERC 至从机（从机返回错误计数 BCNT 至主机）。	
13	RB_LINK_LOOPBACK_EN	RW	允许进入 LOOPBACK 使能位，高有效，配合 TX/RX_TS_CFG[3] 中的 LOOKBACK 使能使用，两者都有效，LINK 才能进入 LOOKBACK 模式（用于使能 LOOPBACK 从机的数据回环及错误计数，LOOPBACK 主机保持 0）。	0
12	RB_LINK_U2_DET_EN	RW	U2 状态下连接设备的检测方式： 1：软件主动拉起 U2 状态下连接设备的检测； 0：硬件自动计时检测 U2 状态下是否连接设备。	0
[11:10]	RB_LINK_CP78_SEL	RW	在 Compliance Pattern 7/8 时，发送连续 0 或连续 1 的长度： 00: 190 bits； 01: 120 bits； 10: 50 bits； 11: 250 bits。	0
[9:8]	RB_LINK_TX_DEEMPH	RW	发送器 de-emphasis 控制： 00: -6.0 dB de-emphasis； 01: -3.5 dB de-emphasis（默认值）； 10: No de-emphasis； 11: 保留。 注：正常工作模式只能为 0/1，其他值用于 COMPLIANCE。	01b
7	RB_LINK_TX_SWING	RW	发送器信号摆幅控制，低摆幅功耗低，但是影响传输距离 1: Low Swing (400mv-1200mv)； 0: Full Swing (800mv-1200mv)。	0
6	RB_LINK_RX_EQ_EN	RW	接收器均衡使能控制，协议规定可选。	0
5	RB_LINK_LFPS_RX_PD	RW	LFPS 接收控制，该位为 1 表示禁用 LFPS 接收	1
4	RB_LINK_COMPLIANCE_EN	RW	POLLING_LFPS 超时是否进入 COMPLIANCE 模式： 1: 超时进入； 0: 超时不进入。	0
3	RB_LINK_PHY_RESET	RW	PIPE 接口复位 1: 复位； 0: 清除复位。	1
2	RB_LINK_SS_PLR_SWAP	RW	交换 SSTX 和 SSRX 极性，如下： 1: SSTXA (SSTXM)，SSTXB (SSTXP)；SSRXA (SSRXM)，SSRXB (SSRXP)。 0: SSTXA (SSTXP)，SSTXB (SSTXM)，	0

			SSRXA (SSRXP), SSRXB (SSRXM)。 注：该位于自适应的 <i>rx polarity</i> 不同的是， <i>rx polarity</i> 只交换 SSRXM 和 SSRXP。	
1	RB_LINK_RX_TERM_EN	RW	接收终端电阻控制： 1：连接端接电阻； 0：移除端接电阻。	0
0	RB_LINK_DOWN_MODE	RW	外设类型： 1：downstream； 0：upstream。 主机模式时，应配置为 downstream，设备模式时配置成 upstream。 特别地：当需要支持 OTG 模式时，应配置成 downstream，Port_Capability (LMP) 中的 Tiebreaker 应设置为 0，表明当链接到主机时，该端口将成为设备，连接到设备时，该端口为主机。协议规定当端口双方都为 downstream 或 upstream 时，Tiebreaker 为 1 的端口将成为 downstream（主机），随后双方需要重新交换 Port_Capability (LMP)。	0

27.2.6.2 LINK 控制寄存器 (R32_LINK_CTRL)

偏移地址：0x04

位	名称	访问	描述	复位值
[31:24]	LINK_RX_TS_CFG	RO	接收到的链路控制： BIT0：1：复位； 0：正常传输。 BIT1：保留。 BIT2：1：使能 Loopback；0：禁用 Loopback。 BIT3：1：禁用扰码； 0：使能扰码。 BIT4~7：保留。	0
[23:16]	LINK_TX_TS_CFG	RW	发送 TS1/TS2 训练序列的 link configuration： BIT0：1：HOT_RESET； 0：正常训练模式。 BIT1：保留。 BIT2：1：使能 Loopback；0：禁用 Loopback。 BIT3：1：禁用扰码； 0：使能扰码。 BIT4~7：保留。 特别地，保留位必须设为 0，BIT0 和 BIT2 不能同时为 1； 设备模式下，如果接收到 HOT_RESET 有效，产生 HOT_RESET 中断标志，同时 BIT0 由硬件自动置 1，以响应主机的 HOT_RESET。	0
15	LINK_TX_LGO_U3	RW	该位有效后发送 LGO_U3，高有效，硬件自动清零； 协议规定接收方不能拒绝进入 U3 状态； 查询 LINK 状态为 U3，表明进入 U3 低功耗状态，此时软件应改变当前电源模式为 P3。	0

14	LINK_TX_LGO_U2	RW	该位有效后发送 LGO_U2，高有效，硬件自动清零； 如果对方以 LXU 回应，表明拒绝进入 U2 状态；如果查询 LINK 状态为 U2，表明对方接受进入 U2 低功耗状态，此时软件应改变当前电源模式为 P2。	0
13	LINK_TX_LGO_U1	RW	该位有效后发送 LGO_U1，高有效，硬件自动清零； 如果对方以 LXU 回应，表明拒绝进入 U1 模式；如果查询 LINK 状态为 U1，表明对方接受进入 U1 低功耗状态，此时软件应改变当前电源模式为 P1。	0
12	LINK_POLLING_EN	RW	高有效，如果 SS.RX_DETECT 检测到 TERM，将在软件设置该位有效后进行 POLLING 握手。 注：AUTO_MODE[9]为 1 时，如果 AUTO_MODE[8]为 1 且当前电源为 P2，开始 POLLING_LFPS 握手； AUTO_MODE[9]为 1 时，如果 AUTO_MODE[7]为 1 且当前电源为 P0，开始 POLLING_LFPS 握手。	0
11	LINK_REG_ROUT_EN	RW	使能 HUB 的路由功能，高有效，有寄存器无接口： 1：上行口可正常路由头包到下行口； 0：上行口无法路由头包到下行口。	0
10	LINK_LUP_LDN_EN	RW	U0 状态，无数据情况下，每 10us 是否发送 LUP、LDN 包： 1：使能发送； 0：禁止发送。	0
9	LINK_TX_UX_EXIT	RW1Z	高有效，硬件自动清零： 1. 如果 LINK 处于 U3 状态，相当于执行 U3_WAKEUP，U3_WAKEUP 握手失败后，软件应该在至少 100ms 后重新发起 U3_WAKUP，主机模式下，如果连续 3 次握手失败，软件应设置 LINK 进入 INACTIVE，检测 TERM，设备可能已经断开。如果接收到 U3_WAKEUP 后，由于某种原因无法返回握手信号，当准备好进入 U0 后，应主动发送 U3_WAKEUP。 2. 如果 LINK 处于 U2 状态，相当于执行 U2_EXIT (LFPS)。 3. 如果 LINK 处于 U1 状态，相当于执行 U1_EXIT (LFPS)。 4. 如果执行 U1_EXIT/U2_EXIT 握手失败，LINK 进入 INACTIVE。	0
8	LINK_TX_WARM_RST	RW1Z	高有效，由软件清零。该位有效将会发送 warm-reset。 主机模式下：由软件拉高，可选择软件或硬	0

			件拉低 1、由软件决定 warm_reset 的持续时间，主机发送 warm_reset 的持续时间应在 80ms~120ms 之间（典型 100ms），当设备检测到 warm_reset 后，应在 tResetDelay（18ms~50ms）时间范围内发送 warm_reset 回应，直到检测到主机发送 warm_reset 结束，设备发送 warm_reset 结束。 2、在软件拉高后，由硬件计时 100ms 后自动拉低（需要配置对应的硬件超时使能 reg_auto_mode[13]） 设备模式下：设备控制器在检测到主机持续发送 18ms 后的 warm_reset 后会自动申明检测到了 warm_reset 并进行复位。 特别地，协议规定只有主机有能力主动发送 warm_reset，设备模式只能被动响应 warm_reset。	
7	LINK_GO_RX_DET	RW1Z	设置该位前应设置 PD_MODE 为 P2 模式，设置 LINK 进入 SS.RX_DETECT，软件查询 TERM_PRESENT 得知是否有连接或连接是否断开。高有效，自动清零。	0
6	LINK_GO_RECOVERY	RW1Z	设置 LINK 进入 SS.RECOVERY，高有效，硬件自动清零。	0
5	LINK_GO_INACTIVE	RW1Z	设置 LINK 进入 SS.INACTIVE，高有效，硬件自动清零。	0
4	LINK_GO_DISABLED	RW	设置 LINK 进入 SS.DISABLED，高有效，需要软件清零。 如果硬件自动进入 SS.DISABLED，那么该位由硬件自动置 1。 设备模式时，如果接收到 USB2.0 总线复位，软件应先设置 PD_MODE 为 P2 模式，然后清零此位，LTSSM 将自动进入 SS.RX_DETECT，设备进行 8 次检测（每 12ms 检测一次），如果，没有检测到 TERM，进入 DISABLE 状态，同时以 USB2.0 模式进行通信。 在主机模式时，应该通过 USB2.0 控制器发送 BUS_RESET 信号，软件应先设置 PD_MODE 为 P2 模式，然后清零此位，如果检测到 TERM，将进行 POLLING_LFPS 握手。否则软件应设置电源模式为 P3，以 USB2.0 模式进行通信。	1
[3:2]	Reserved	RO	保留。	0
[1:0]	LINK_PD_MODE	RW	配置 PHY 当前电源模式，对应于 PIPE 中的 P0/P1/P2/P3： 00：P0：正常工作模式； 01：P1：低功耗模式，处于该模式下端口双方	11b

			定期发送 PING-LFPS 保持连接； 10: P2: 低功耗模式，该模式比 P1 模式功耗更低，同时退出该模式的时间也会增加； 11: P3: 超低功耗模式，该模式下 PHY 的时钟（125MHz）将被关闭。 改变该配置值，将导致 LINK_BUSY 拉高，软件应查询 LINK_BUSY 为 0，表示切换完成。 查询 LINK_STATUS 中 PD_MODE_CURRENT 得到 PHY 当前的电源模式（可选操作）。	

27.2.6.3 LINK 中断使能寄存器（R32_LINK_INT_CTRL）

偏移地址：0x08

位	名称	访问	描述	复位值
31	LINK_IE_STATE_CHANGE	RW	链路状态机发生变化标志中断使能： 1：使能； 0：禁止。	0
30	LINK_IE_U1_TOUT	RW	U1 超时中断使能： 1：使能； 0：禁止。	0
29	LINK_IE_U2_TOUT	RW	U2 超时中断使能： 1：使能； 0：禁止。	0
28	LINK_IE_UX_FAIL	RW	UX 转换失败中断使能： 1：使能； 0：禁止。	0
27	LINK_IE_TX_WARMRESET	RW	发送 warm_reset 结束中断使能： 1：使能； 0：禁止。	0
26	LINK_IE_UX_EXIT_FAIL	RW	退出 Ux 失败中断使能： 1：使能； 0：禁止。	0
[25:24]	Reserved	RO	保留	0
23	LINK_IE_RX_LMP_TOUT	RW	接收 LMP 超时中断使能： 1：使能； 0：禁止。	0
22	LINK_IE_TX_LMP	RW	成功进入 U0，可以发送 HP 包中断使能： 1：使能； 0：禁止。	0
21	LINK_IE_RX_LMP	RW	接收到链路命令标志中断使能： 1：使能； 0：禁止。	0
20	LINK_IE_RX_DET	RW	链路进入 Rx. Detect 状态中断使能： 1：使能； 0：禁止。	0
19	LINK_IE_LOOPBACK	RW	链路进入回环模式，用于测试以及错误隔离	0

			中断使能： 1：使能； 0：禁止。	
18	LINK_IE_COMPLIANCE	RW	链路进入服从性测试，用兼容性测试或物理层一致性测试中断使能： 1：使能； 0：禁止。	0
17	LINK_IE_HPBUF_FULL	RW	hp_buf 发送 fifo 满中断使能： 1：使能； 0：禁止。	0
16	LINK_IE_HPBUF_EMPTY	RW	hp_buf 发送 fifo 空中断使能： 1：使能； 0：禁止。	0
15	LINK_IE_HOT_RST	RW	hot reset (热复位)，使用 TS1/TS2 有序集的复位中断使能： 1：使能； 0：禁止。	0
14	LINK_IE_U3_WAKEUP	RW	处于 U3 状态，接收到低频周期信号 (LFPS) 唤醒中断使能： 1：使能； 0：禁止。	0
13	LINK_IE_WARM_RST	RW	使用 LPFS 进行暖复位（未连接）中断使能： 1：使能； 0：禁止。	0
12	LINK_IE_UX_EXIT	RW	收到退出 Ux 的请求 中断使能： 1：使能； 0：禁止。	0
11	LINK_IE_TXEQ	RW	LINK 进入 POLLING_RXEQ，用于接收器均衡训练中断使能： 1：使能； 0：禁止。	0
10	LINK_IE_TERM_PRESS	RW	LINK 进入 RX_DETECT，用于检测远端接收器追踪段阻抗检测中断使能： 1：使能； 0：禁止。	0
9	LINK_IE_UX_REJ	RW	发送 LG0_Ux，接收到拒绝进入 Ux 发送的 LXU 中断使能： 1：使能； 0：禁止。	0
8	LINK_IE_U3_WK_TIMEOUT	RW	在请求退出 U3 命令超时中断使能： 1：使能； 0：禁止。	0
7	LINK_IE_GO_U0	RW	LINK 处于 U0 中断使能： 1：使能； 0：禁止。	0
6	LINK_IE_GO_U1	RW	LINK 处于 U1 中断使能：	0

			1: 使能; 0: 禁止。	
5	LINK_IE_GO_U2	RW	LINK 处于 U2 中断使能: 1: 使能; 0: 禁止。	0
4	LINK_IE_GO_U3	RW	LINK 处于 U3 中断使能: 1: 使能; 0: 禁止。	0
3	LINK_IE_DISABLE	RW	LINK 处于 DISABLED 中断使能: 1: 使能; 0: 禁止。	0
2	LINK_IE_INACTIVE	RW	LINK 处于 INACTIVE 中断使能: 1: 使能; 0: 禁止。	0
1	LINK_IE_RECOVERY	RW	由于错误 LINK 处于 RECOVERY 中断使能: 1: 使能; 0: 禁止。	0
0	LINK_IE_READY	RW	LINK 完成初始化, 包括两个端口之前进行 (Header Sequence Number Advertisement)、(RX Headr Buffer Credit Advertisement) 中断使能: 1: 使能; 0: 禁止。	0

27.2.6.4 LINK 中断标志寄存器 (R32_LINK_INT_FLAG)

偏移地址: 0x0C

位	名称	访问	描述	复位值
31	LINK_IF_STATE_CHG	RW1Z	link_reset 复位为 0 (写 1 清零): 1: 链路状态机发生变化标志; 0: 链路状态机未发生变化。	0
30	LINK_IF_U1_TOUT	RW1Z	U1 超时中断; (写 1 清零): 1: 超时; 0: 未超时;	0
29	LINK_IF_U2_TOUT	RW1Z	U2 超时中断; (写 1 清零): 1: 超时; 0: 未超时;	0
28	LINK_IF_UX_FAIL	RW1Z	进入 Ux 失败中断; (写 1 清零): 1: 进入 Ux 失败; 0: 无动作。	0
27	LINK_IF_TX_WARMRST	RW1Z	发送 warm_reset 结束中断; (写 1 清零): 1: 发送 warm_reset 结束; 0: 无动作。	0
26	LINK_IF_UX_EXIT_FAIL	RW1Z	退出 Ux 失败中断; (写 1 清零): 1: 退出 Ux 失败; 0: 无动作。	0

[25:24]	Reserved	R0	保留。	0
23	LINK_IF_RX_LMP_TOUT	RW1Z	LINK 初始化完成后交换 Port Capabilities/Configuration LMPs 超时 (20us) 中断标志。	0
22	LINK_IF_TX_LMP	RW1Z	LINK 初始化完成中断标志, 该信号拉高后软件配置发送 Port Capabilities LMP 进行端口能力交换。	0
21	LINK_IF_RX_LMP	RW1Z	接收 LMP 中断标志: 在端口能力交换阶段控制相关 LMP 发送: 下行口接收到 Port Capabilities LMP, 软件配置发送 Port Configuration LMP; 上行口接收到 Port Capabilities LMP, 软件配置发送 Port Configuration Response LMP。	0
20	LINK_IF_RX_DET	RW1Z	进入 RX_DETECT 状态中断标志, 此位置 1 后, 软件将 PWR_MODE 设置为 P2, 准备进行 RX_DETECT。	0
19	LINK_IF_LOOPBACK	RW1Z	LINK 进入 LOOPBACK 状态中断标志。	0
18	LINK_IF_COMPLIANCE	RW1Z	LINK 进入 COMPLIANCE 状态中断标志。	0
17	LINK_IF_HPBUF_FULL	RW1Z	Header Packet 缓冲区满中断标志。	0
16	LINK_IF_HPBUF_EMPTY	RW1Z	Header Packet 缓冲区空中断标志。	0
15	LINK_IF_HOT_RST	RW1Z	设备接收到 HOT RESET 中断标志: 在该信号拉高后, 软件应该清除 TX_TS_CFG[0] 结束 (完成) HOTRESET 过程。	0
14	LINK_IF_WAKEUP	RW1Z	电源处于 P3 模式下, 检测到 LFPS 信号中断标志。	0
13	LINK_IF_WARM_RST	RW1Z	设备接收到的 WARM RESET 状态变化 (有效→无效或无效→有效) 中断标志。	0
12	LINK_IF_UX_EXIT	RW1Z	LINK 接收到 LFPS 准备退出 U1/U2/U3 请求中断标志, 此位置 1 后, 软件应将 PWR_MODE 设置为 P0, 在将 PWR_MODE 设置为 P0 后, Ux 退出计时器开始计时, 在达到 U1_EXIT_TIMER、U2_EXIT_TIMER、U3_WAKE_TIMER 之后, LINK 退出 Ux 状态, 进入 Recovery。	0
11	LINK_IF_TXEQ	RW1Z	LINK 进入 TXEQ 状态中断标志: 表明完成 POLLING 握手, 等待软件将 PWR_MODE 设置为 P0, 以便进入 TXEQ 阶段。	0
10	LINK_IF_TERM_PRE	RW1Z	检测到 TERM 断开或连接中断标志。	0
9	LINK_IF_UX_REJ	RW1Z	LINK 拒绝进入低功耗模式 (U1/U2) 中断标志。	0
8	LINK_IF_U3_WK_TOUT	RW1Z	从 U3 唤醒超时中断标志 (10ms)。	0
7	LINK_IF_GO_U0	RW1Z	LINK 进入 U0 状态中断标志。	0

6	LINK_IF_GO_U1	RW1Z	LINK 进入 U1 状态中断标志。	0
5	LINK_IF_GO_U2	RW1Z	LINK 进入 U2 状态中断标志。	0
4	LINK_IF_GO_U3	RW1Z	LINK 进入 U3 状态中断标志。	0
3	LINK_IF_DISABLE	RW1Z	LINK 进入 SS. DISABLE 状态中断标志： 注：软件直接配置 GO_DISABLE 使能 (link_ctrl)，导致 LINK 直接进入 DISABLE，中断不会拉高。	0
2	LINK_IF_INACTIVE	RW1Z	LINK 进入 SS. INACTIVE 状态中断标志。 注：软件直接配置 GO_INACTIVE 使能 (link_ctrl)，导致 LINK 直接进入 INACTIVE，中断不会拉高。	0
1	LINK_IF_RECOVERY	RW1Z	LINK 进入 SS. RECOVERY 状态中断标志。 该位置 1 后，应查询 LINK_ERR_STATUS 获取 LINK 进入 SS. RECOVERY 的原因。 注：软件直接配置 GO_RECOVERY 使能 (link_ctrl)，导致 LINK 直接进入 RECOVERY，中断不会拉高。	0
0	LINK_IF_READY	RW1Z	LINK 进入 U0 状态并完成 LINK 初始化中断标志。	0

27.2.6.5 LINK 状态寄存器 (R32_LINK_STATUS)

偏移地址：0x10

位	名称	访问	描述	复位值
31	LINK_HPBUF_EMPTY	RO	HP 缓冲区空。	x
30	LINK_HPBUF_FULL	RO	HP 缓冲区满。	x
29	LINK_HPBUF_IDLE	RO	hp_buf 发送 fifo idle 的状态： 1: hp_buf 发送 fifo 有数据； 0: hp_buf 发送 fifo 为空。	x
[28:23]	RESERVE	RO	保留。	0
22	LINK_U3_SLEEP_ALLOW	RO	链路处于 U3 状态，允许睡眠。	x
21	LINK_U2_SLEEP_ALLOW	RO	链路处于 U2 状态，允许睡眠。	x
20	LINK_RXDET_SLEEP_ALLOW	RO	链路处于 RXDET 状态，允许睡眠。	x
19	LINK_WAKUP	RO	收到链路唤醒信号。	x
18	LINK_RX_LFPS	RO	链路接收到 LFPS 信号。	x
17	LINK_RX_DETECT	RO	链路处于 P2，正在进行 rx_detect。	x
16	LINK_RX_UX_EXIT_REQ	RO	接收到退出 Ux 的请求： 1: LINK 接收到有效的 Ux exit LFPS 信号，准备退出 U1/U2/U3，软件应该配置电源模式为 P0（在电源模式转为 P0 后，Ux 退出计时器开始计时并在达到预设值后 LINK 进入 RECOVERY）； 0: 无变化。	0

[15:12]	Reserved	RO	保留。	0
[11:8]	LINK_STATE	RO	链路状态： 0000: STATE_U0; 0001: STATE_U1; 0010: STATE_U2; 0011: STATE_U3; 0100: STATE_DISABLED; 0101: STATE_RXDET; 0110: STATE_INACTIVE; 0111: STATE_POLLING; 1000: STATE_RECOVERY; 1001: STATE_HOTRST; 1010: STATE_COMPLIANCE; 1011: STATE_LOOPBACK; 其他: 保留。	4h
7	Reserved	RO	保留。	0
6	LINK_TXEQ	RO	链路处于 POLLING_RXEQ。	0
[5:4]	LINK_PD_MODE_MASK	RO	链路电源状态： 00: P0 状态; 01: P1 状态; 10: P2 状态; 11: 默认状态 P3	10b
3	LINK_READY	RO	当 LINK 进入 U0 状态后, 完成初始化 (广播) 后该位置 1, 退出 U0 状态该位自动清零。	0
2	LINK_BUSY	RO	LINK 忙状态, 当切换 PD_MODE 时该位为 1, 完成切换后由硬件自动清零。	0
1	LINK_RX_WARM_RST	RO	接收到主机有效的 warm-reset 信号 (接收到主机的 warm_reset 18ms 后硬件自动拉起)。	0
0	LINK_RX_TERM_PRES	RO	RX_DETECT 后, 如果有接收终端电阻存在, 该位为 1。	0

27.2.6.6 LINK ITP 超时模式寄存器 (R8_LINK_ITP_PRE)

偏移地址: 0x17

位	名称	访问	描述	复位值
[7:0]	ITP_PRE	RO	ITP 超时模式： 1: 忽略上下行端口 ITP 的收发对链路空闲状态的影响; 0: 下行端口发送非延迟 ITP/上行端口接收非延迟 ITP 时, 链路为非空闲状态。	0x80

27.2.6.7 LINK U2 不活动超时计数器阈值寄存器 (R8_LINK_U2_INACT_TIMER)

偏移地址: 0x1D

位	名称	访问	描述	复位值
[7:0]	U2_INACTIVE_TIMER	RW	U2 的不活动超时计数器阈值的值: U2 Timeout 值 (该值表示处于 U0/U1 不活动	0

			状态的最大允许时间，超时进入 U2 状态），取值范围：0x00~0xff，256us/LSB。 注：当 U1/U2 Timeout 值取 0xff 时，表示链路不活动状态的超时时间为无穷大，计时器记到 0xff 也不会主动发起到 U1/U2 的转换请求。	

27.2.6.8 U1 唤醒 LFPS 有效时长寄存器 (R8_LINK_U1_WKUP_FILTER)

偏移地址：0x28

位	名称	访问	描述	复位值
[7:0]	U1_WKUP_FILTER	RW	U1 退出时接收到的 LFPS 的有效时长。端口发送 LFPS 进行握手，当接收 LFPS 达到该时间，可认为接收的 U1 EXIT 有效，拉高发送的 LFPS_last，握手成功，默认为 600ns U1_WKUP_FILTER[7] : 控制 U1_WKUP_FILTER[6:0]的时间单位； 1: U1_WKUP_FILTER[6:0]单位为 us，时长为 (n)us； 0: U1_WKUP_FILTER[6:0]单位为 8ns，时长为 (8n)ns，n 取值 0~124； 注：时长大于 1us 时必须使用 U1_WKUP_FILTER[7]置 1 的配置方法，否则 U1 无法唤醒。	0x4A

27.2.6.9 U2 唤醒 LFPS 有效时长寄存器 (R8_LINK_U2_WKUP_FILTER)

偏移地址：0x2C

位	名称	访问	描述	复位值
[7:0]	U2_WKUP_FILTER	RW	U2 唤醒和 loopback 退出时接收到的 LFPS 的有效判断时长，端口发送 LFPS 进行握手，当接收 LFPS 达到该时间，发送的 LFPS_last 拉高，握手成功，可认为接收的 U2 EXIT 有效，时长为 (n)us，默认 2us。	0x02

27.2.6.10 U3 唤醒 LFPS 有效时长寄存器 (R8_LINK_U3_WKUP_FILTER)

偏移地址：0x30

位	名称	访问	描述	复位值
[7:0]	U3_WKUP_FILTER	RW	U3 唤醒时接收到的 LFPS 的有效判断时长，端口发送 LFPS 进行握手，当接收 LFPS 达到该时间，发送的 LFPS_last 拉高，握手成功，可认为接收的 U3 EXIT 有效，时长为 (n)us，默认 100us。	0x64

27.2.6.11 LINK 同步延迟寄存器 (R16_LINK_ISO_DLY)

偏移地址：0x40

位	名称	访问	描述	复位值

[15:0]	LINK_ISOCH_DLY	RW	串行比特流解析到并行数据的延迟时间，默认 40ns，枚举时 SET ISOCH DELAY 主机将此信息发送到设备，作为设备将延迟信息写入该寄存器，该寄存器低 3 位无效（因时钟频率为 125MHz，延迟时间需设置为 8*n）。	0x28

27.2.6.12 链路电源管理寄存器（R16_LINK_LPM_CR）

偏移地址：0x50

位	名称	访问	描述	复位值
[15:14]	Reserved	RO	保留。	0
13	PHY_CHSEL_AUTO	RW	PHY 层收发通道类型自动选择： 1：设备检测完成之后如果连接状态标志不变的话切换至另一通道类型，默认通道类型为普通 USB 接口； 0：PHY 层禁止自动切换收发通道类型，只能使用软件设置收发类型，默认普通 USB 接口。	0
12	LPM_RXDET_EN	RO	当 lpm 计数达到期望值 RXDET_EXP，确认有设备连接在 PHY 上，此位拉高。	0
11	LPM_TERM_PRESENT	RO	PHY 层检测到设备连接。	0
10	LPM_TERM_CHG	RO	连接设备发生变化，插入设备或拔出设备。	0
9	LPM_EN	RW	CLKLPM 固定为 12MHz，当 LINK_PD_MODE=3 & LPM_EN =0 & OSCOUTEN =0 & PLLALIV=0，CLKOSCO 将停止，晶振停振； 1：使能 CLKLPM 在 L1 模式下； 0：禁止。 注：该位与 USBSS 控制器无关，系统进入低功耗需要关闭晶振，应清零此位。	0
8	LPM_RST	RW	LPM 相关寄存器的复位信号： 1：复位 LPM 计数器、LPM 终端检测； 0：不复位。	1
[7:0]	RXDET_EXP	RW	检测外部设备计数器寄存器。	0x80

27.2.6.13 PORT_CAP 寄存器（R32_LINK_LMP_PORT_CAP）

偏移地址：0x54

位	名称	访问	描述	复位值
31	LINK_LMP_RX_CAP_VLD	RO	接收到有效的 PORT_CAP-LMP，协议规定在进入 U0 状态后 20us 内 LINK 双方交换 PORT_CAP-LMP。 注：此标志信号会在 LINK 状态发生改变进入 port_state_polling 时置 0。	0
30	LINK_LMP_TX_CAP_VLD	RO	PORT Capability 配置完成标志（软件操作，写 1 清 0）： 1：表示 PORT Capability 配置未完成； 0：表示 PORT Capability 配置完成（主机与设备分两种情况）。	0

			主机在接收到设备的端口配置响应信息 (PORT Configuration Response) 后清 0; 设备在接收到主机的端口配置 (PORT Configuration) 后回复响应 (PORT Configuration Response) 并清 0。	
[29:24]	LINK_SPEED	RO	[24] 位置 1, 表示支持设备支持 USB3.2 Gen1 (5Gbps)。	x
[23:0]	LINK_REG_PORT_CAP	RO	1、LINK_REG_PORT_CAP[23:20]: Tiebreaker, 用于两个同时具有上行、下行能力的 PORT (LINK_REG_PORT_CAP[17:16]=11) 连接时的 PORT 类型仲裁, 值更高的端口成为下行端口, 如果两个 PORT 的 Tiebreaker 相等, 将会重新交换 PORT_CAP。 2、LINK_REG_PORT_CAP[17:16]: Direction LINK_REG_PORT_CAP[17]: 此位为 1, 表示端口可以被配置为上行端口; LINK_REG_PORT_CAP[16]: 此位为 1, 表示端口可以被配置为下行端口; 3、LINK_REG_PORT_CAP[7:0]: Nub HP Buffers, 表示该设备支持的 headr packet buffers 的个数; 4、其他位保留。	0

27.2.6.14 LMP 接收数据 0 寄存器 (R32_LINK_LMP_RX_DATA0)

偏移地址: 0x58

位	名称	访问	描述	复位值
[31:0]	LINK_LMP_RX_DATA0	RO	LMP 接收完成后, HP 数据存放该寄存器 [31:0]。	x

27.2.6.15 LMP 接收数据 1 寄存器 (R32_LINK_LMP_RX_DATA1)

偏移地址: 0x5C

位	名称	访问	描述	复位值
[31:0]	LINK_LMP_RX_DATA1	RO	LMP 接收完成后, HP 数据存放该寄存器 [63:32]。	x

27.2.6.16 LMP 接收数据 2 寄存器 (R32_LINK_LMP_RX_DATA2)

偏移地址: 0x60

位	名称	访问	描述	复位值
[31:0]	LINK_LMP_RX_DATA2	RO	LMP 接收完成后, HP 数据存放该寄存器 [95:64]。	x

27.2.6.17 USB 自定义 HP 数据 0 寄存器 (R32_LINK_LMP_TX_DATA0)

偏移地址: 0x64

位	名称	访问	描述	复位值
[31:0]	LINK_LMP_TX_DATA	RW	用户自定义发送 HP 的数据 [31:0]。	x

	0			

27.2.6.18 USB 自定义 HP 数据 1 寄存器 (R32_LINK_LMP_TX_DATA1)

偏移地址: 0x68

位	名称	访问	描述	复位值
[31:0]	LINK_LMP_TX_DATA 1	RW	用户自定义发送 HP 的数据[63:32]。	x

27.2.6.19 USB 自定义 HP 数据 2 寄存器 (R32_LINK_LMP_TX_DATA2)

偏移地址: 0x6C

位	名称	访问	描述	复位值
[31:0]	LINK_LMP_TX_DATA 2	RW	用户自定义发送 HP 的数据[95:64]。	x

注:

- 1、在写入 LINK_LMP_TX_DATA2 后, 硬件会自动产生一个触发信号将 DATA0/DATA1/DATA2 的信息发送出去。
- 2、自定义 HP 的发送应用于发送 LMP (TX_PORT_CAP/TX_PORT_CFG 等)/PING-TP 等寄存器未定义包的发送。

第 28 章 控制器局域网（CAN）

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(主动)。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。

28.1 主要特性

- 提供 3 组 CAN 控制器，兼容 CAN 规范 2.0A 和 2.0B(主动)
- 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
- 每组 CAN 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
- 每组 CAN 支持三级邮箱深度的 2 个接收 FIFO，三个 CAN 总共 42 组过滤器组可供配置，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

28.2 CAN 控制器工作模式

CAN 控制器可以对寄存器 CANx_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。

28.2.1 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 CANx_CTLR 中的 INRQ 位置 1，请求 CAN 控制器进入初始化模式，当寄存器 CANx_STATR 的 INAK 位自动置 1 则成功进入初始化状态。同样对寄存器 CANx_CTLR 中的 INRQ 位清零，请求 CAN 控制器退出初始化模式，当寄存器 CANx_STATR 的 INAK 位自动清 0 则成功退出初始化状态。

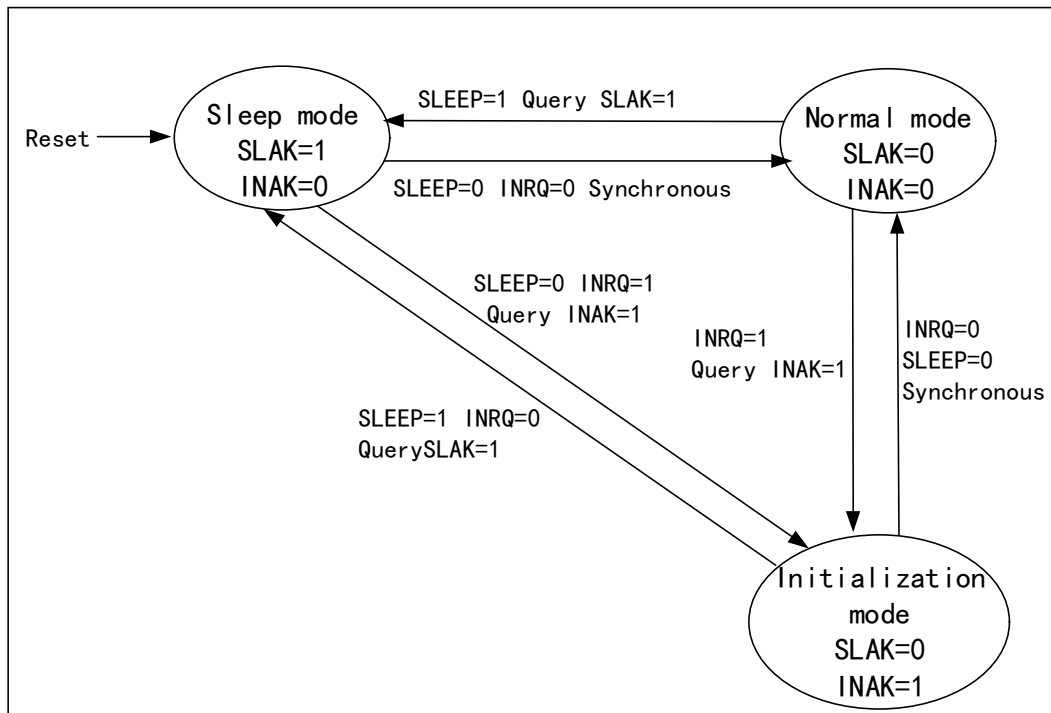
对过滤器组进行初始化，可以在非初始化模式下进行，不过必须对寄存器 CANx_FCTLr 的 FINIT 位进行置 1，此时禁止接收报文。

28.2.2 睡眠模式

对寄存器 CANx_CTLR 中的 SLEEP 位置 1，请求 CAN 控制器进入睡眠模式，当寄存器 CANx_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 CANx_CTLR 的 SLEEP 位清 0，INRQ 位置 1，当寄存器 CANx_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 CANx_CTLR 的 SLEEP 位清 0，当寄存器 CANx_STATR 的 SNAK 位自动清 0 则进入正常模式。

图 28-1 CAN 工作模式切换



28.3 CAN 控制器测试模式

在初始化模式下，对寄存器 CANx_BTMR 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 CANx_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。

28.3.1 静默模式

对寄存器 CANx_BTMR 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。

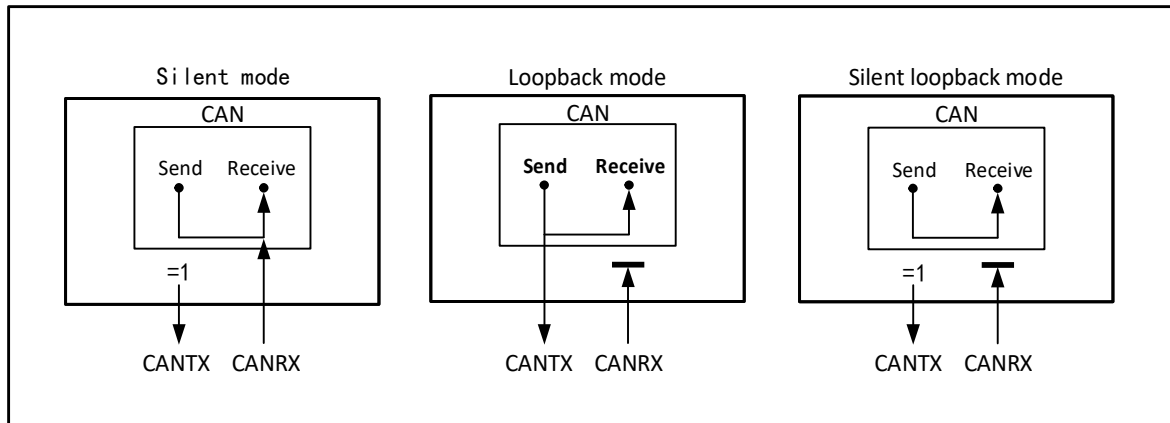
28.3.2 环回模式

对寄存器 CANx_BTMR 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。

28.3.3 静默环回模式

对寄存器 CANx_BTMR 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 28-2 CAN 总线的三种测试模式



28.4 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。

28.5 CAN 控制器功能描述

28.5.1 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 `CANx_TXMIDyR`、`CANx_TXMDTyR`、`CANx_TXMDLyR` 和 `CANx_TXMDHyR` 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 `CANx_TXMIDyR` 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 `CANx_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 `CANx_TSTATR` 的 `ALST` 位置 1，若发送错误，则 `TERR` 位置 1。

28.5.2 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 `CANx_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。

28.5.3 发送中止处理

若对寄存器 `CANx_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），也有可能失败（发送完成），结果可由寄存器 `CANx_TSTATR` 的 `TXOK` 位来查询。

28.5.4 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 `CANx_TTCTL` 寄存器的 `MODE` 位来选择默认模式和增强模式。对寄存器 `CANx_CTLR` 的 `TTCM` 和 `NART` 位置 1，使能时间触发模式并禁止自动重传，`CANx_TTCTL` 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

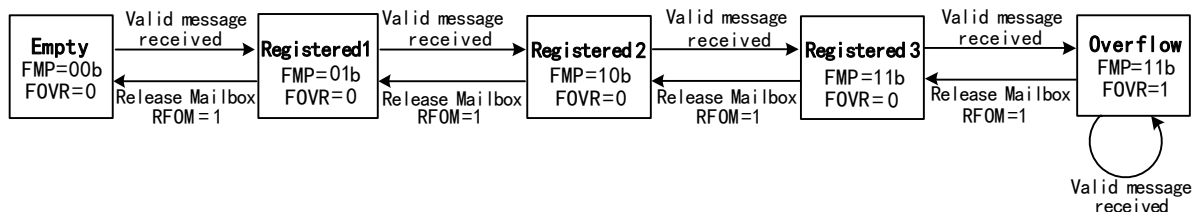
的帧起始位的采样点位置被采样并产生时间戳。若使用增强模式，需要将配置 CANx_TTCTLR 寄存器的 MODE 位为 1，来开启增强模式，使用该模式，在整个 CAN 网络中，必须存在三个或三个以上的节点，其中一个节点发送时间基准，其他节点收到该基准节点的时间戳后，通过向 CANx_TTCTLR 寄存器的 TIMRST 位写 1 复位内部计数器，将内部计数器进行同步，这样除了发送时间基准的节点之外，其余 CAN 节点实现了时间同步，之后将待发送的数据写入发送邮箱，依次配置各节点的时间触发计数值（CANx_TTCNT 寄存器的 TIMCNT）和内部计数器计数终值（CANx_TTCTLR 寄存器 TIMCMV），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。

28.5.5 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 CANx_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 CANx_RFIF0x 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 CANx_RFIF0x 的 FMR[1:0] 设置二进制 01b；若此时读取邮箱数据寄存器 CANx_RXMDLyR 和 CANx_RXMDHyR，通过对接收 FIFO 寄存器 CANx_RFIF0x 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 CANx_RFIF0x 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 28-3 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 CANx_RFIF0x 的 FOVR 位会硬件自动置 1，以供溢出查询。寄存器 CANx_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 CANx_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 CANx_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 28.6 节 CAN 中断。

28.5.6 接收报文标识符过滤

模块中有着多达 42 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 CANx_FxR0 和 CANx_FxR1 组成。过滤器组的位宽都可以通过设置寄存器 CANx_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 CANx_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 CANx_FWR 的各个位选择启用或禁用。设置寄存器 CANx_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 28-1 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 28-1 32 位屏蔽位模式

标识符寄存器	CANx_FxR1[31:24]	CANx_FxR1[23:16]		CANx_FxR1[15:8]	CANx_FxR1[7:0]			
屏蔽位寄存器	CANx_FxR2[31:24]	CANx_FxR2[23:16]		CANx_FxR2[15:8]	CANx_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，接收报文标识符必须与其中一个寄存器保持一致才能通过筛选。

表 28-2 32 位标识符列表模式

标识符寄存器	CANx_FxR1[31:24]	CANx_FxR1[23:16]		CANx_FxR1[15:8]	CANx_FxR1[7:0]			
屏蔽位寄存器	CANx_FxR2[31:24]	CANx_FxR2[23:16]		CANx_FxR2[15:8]	CANx_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 28-3 16 位屏蔽位模式

标识符寄存器 n	CANx_FxR1[15:8]	CANx_FxR1[7:0]			
屏蔽位寄存器 n	CANx_FxR1[31:24]	CANx_FxR1[23:16]			
标识符寄存器 n+1	CANx_FxR2[15:8]	CANx_FxR2[7:0]			
屏蔽位寄存器 n+1	CANx_FxR2[31:24]	CANx_FxR2[23:16]			
映射	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]

表 28-4 16 位标识符列表模式

标识符寄存器 n	CANx_FxR1[15:8]	CANx_FxR1[7:0]			
屏蔽位寄存器 n	CANx_FxR1[31:24]	CANx_FxR1[23:16]			
标识符寄存器 n+1	CANx_FxR2[15:8]	CANx_FxR2[7:0]			
屏蔽位寄存器 n+1	CANx_FxR2[31:24]	CANx_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 CANx_RXMDTyR 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 28-4 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

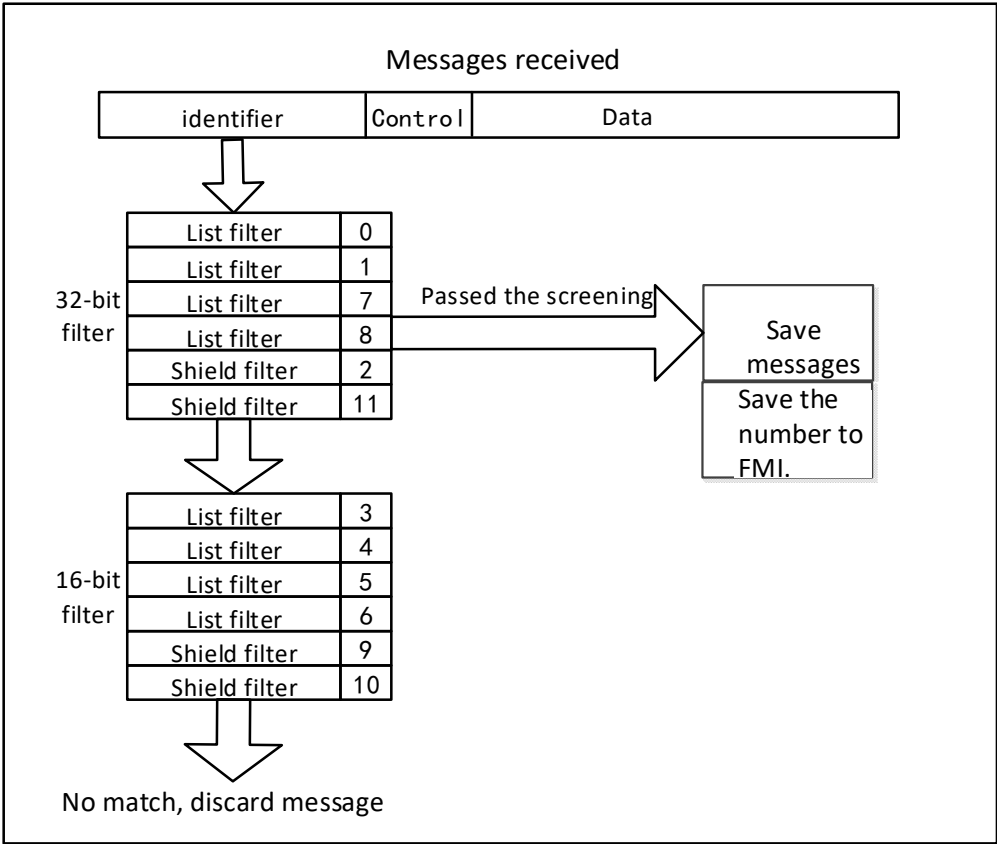
如图 28-5 所示：在接收报文时，先把标识符与 32 位标识符列表模式过滤器进行匹配筛选，没有匹配再与 32 位屏蔽位模式过滤器进行匹配筛选，没有匹配则继续与 16 位标识符列表模式过滤器进行

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 CANx_RXMDTyR 的 FMI 中。

图 28-4 过滤器编号的示例

Filter group number	FIFO0	Filter number	Filter group number	FIFO1	Filter number
0	32-bit shield mode	0	1	16-bit list mode	0 1 2 3
2	16-bit list mode	1 2 3 4	4	16-bit shield mode	4 5
3	32-bit list mode	5 6	6	32-bit list mode	6 7
5	Disabled 16-bit shield Mode	7 8	9	32-bit shield mode	8
7	32-bit shield mode	9	11	Disabled 16-bit shield Mode	9 10
8	32-bit list mode	10 11	12	32-bit list mode	11 12
10	16-bit shield mode	12 13	13	32-bit list mode	13 14

图 28-5 过滤器过滤示例



28.5.7 出错处理

CAN 控制器依靠状态错误寄存器 CANx_ERRSR，对于总线上的出错管理。状态错误寄存器 CANx_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。

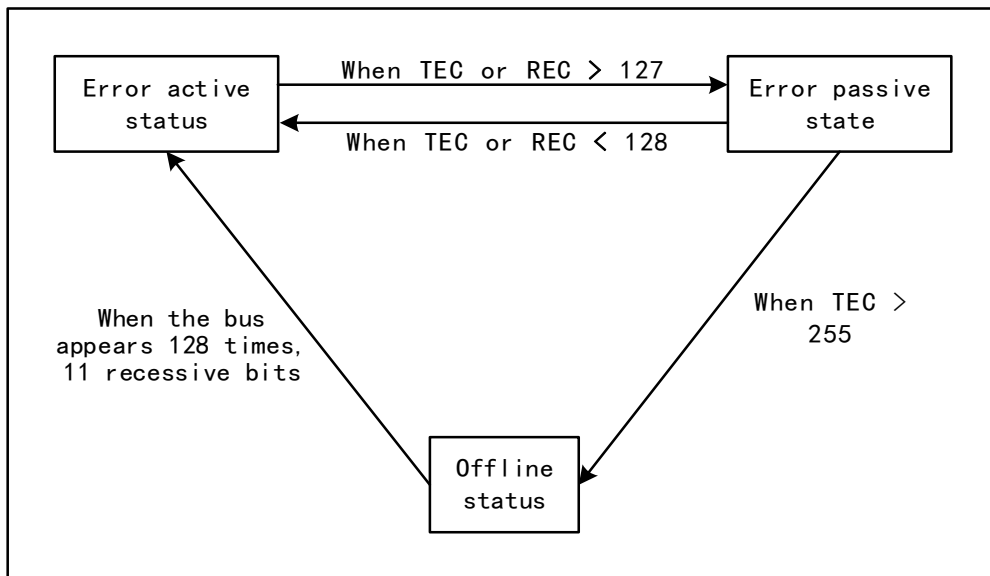
当状态错误寄存器 CANx_ERRSR 里的 TEC 和 REC 小于 128 时，当前 CAN 节点处于错误主动状态，可以正常参与总线通信，并且在检测到错误的时候发出主动错误标志。

当状态错误寄存器 CANx_ERRSR 里的 TEC 和 REC 大于 127 时，当前 CAN 节点处于错误被动状态，并且在检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 CANx_ERRSR 里的 TEC 大于 255 时，当前 CAN 节点进入离线状态。

当总线监测到 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 CANx_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 位进入初始化模式，随后退出初始化，才能退出离线状态。

图 28-6 CAN 错误状态切换图



28.5.8 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 T_q 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。

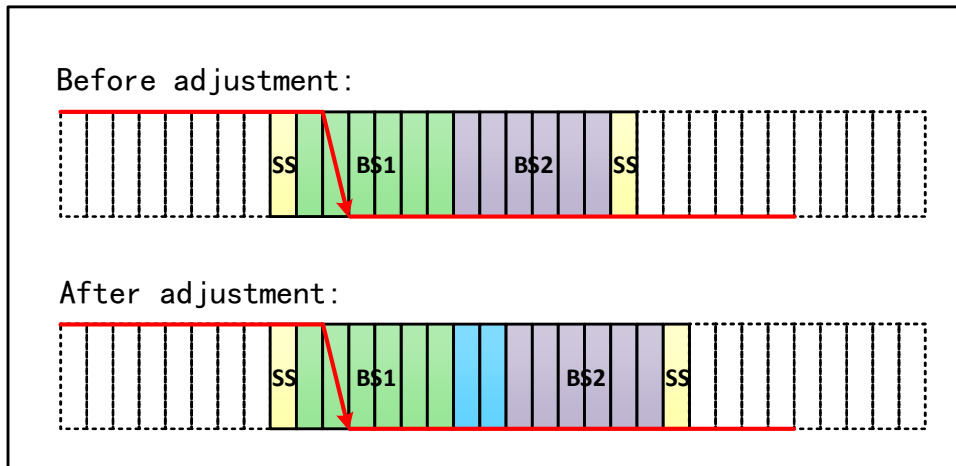
CAN 控制器把上述四段重新划分为三段，分别为：

- 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。

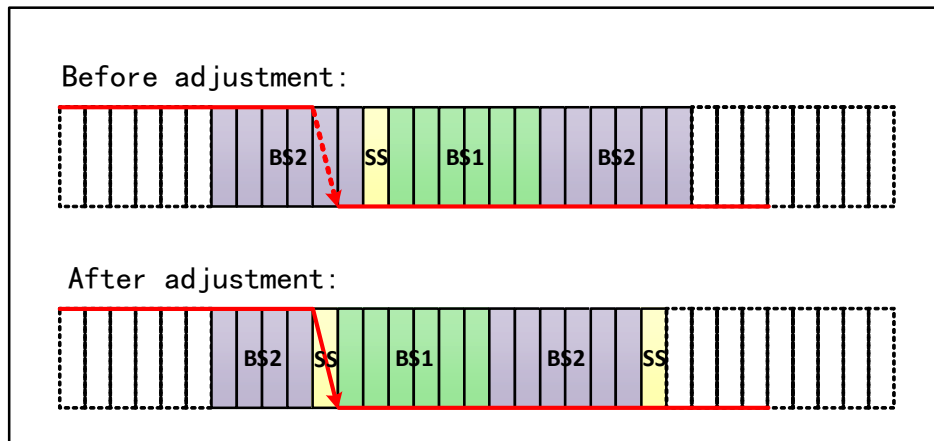
上述参数都可以在 CAN 总线时序寄存器 CANx_BTIMR 里配置。

图 28-7 跳变出现在 BS1 中



如图 28-7，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 28-8 跳变出现在 BS2 中



如图 28-8，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。

28.6 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 CANx_INTENR，可以允许或禁用各个中断源。

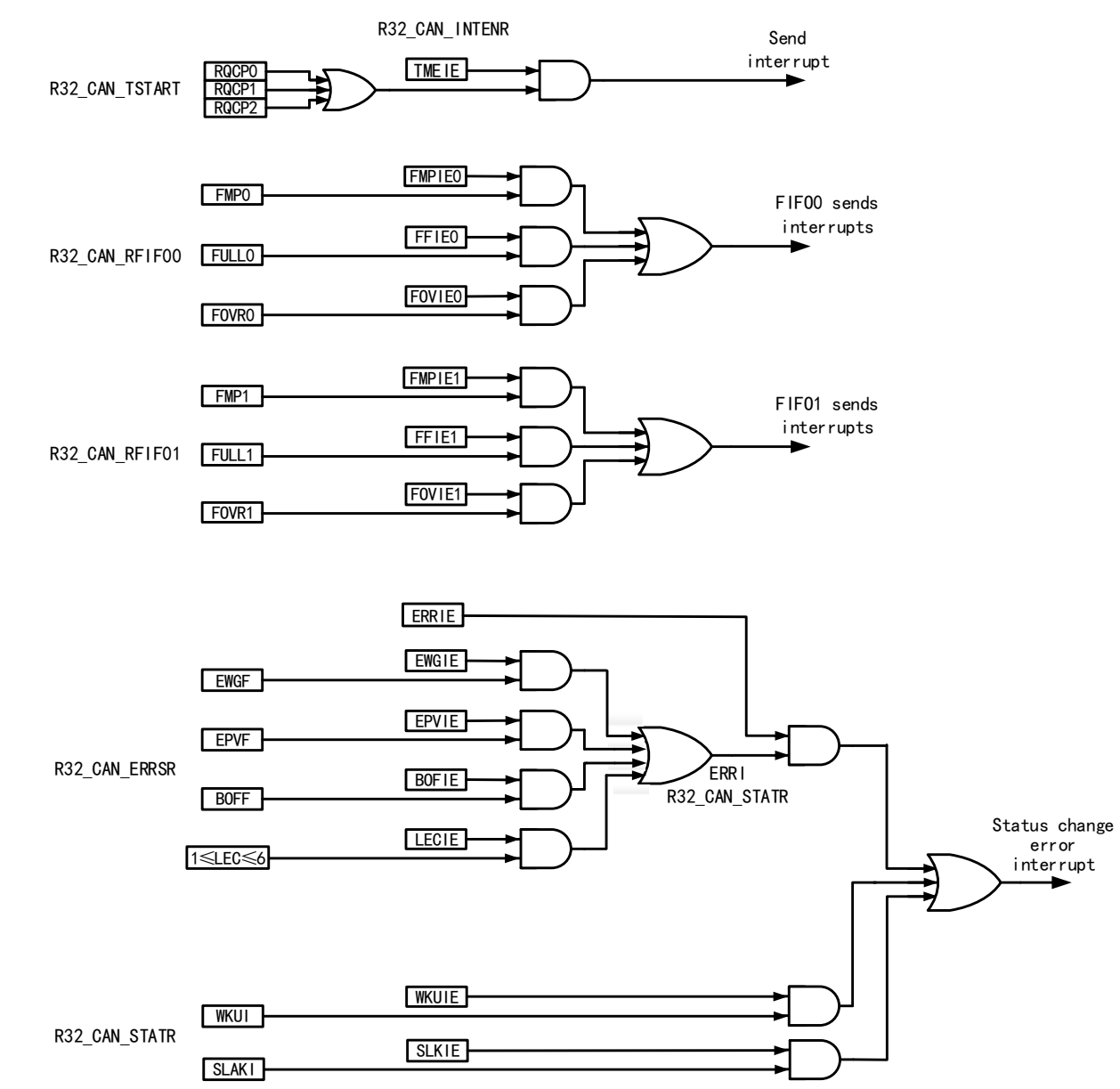
发送中断由发送邮箱变空事件产生，中断产生后，查询寄存器 CANx_TSTATR 的 RQCP0、RQCP1 和 RQCP2 位来判断是哪个邮箱变空事件产生。

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 CANx_RFIF00 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 CANx_RFIF01 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 28-9 CAN 中断逻辑图



28.7 寄存器描述

CAN 控制器相关的寄存器必须用 32 位字的方式来操作且操作时必须确保 CAN 的外设时钟已使能。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 CANx_BTIMR。

表 28-6 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	0x0000C002
R32_CAN1_TSTATR	0x40006408	CAN1 发送状态寄存器	0x1C000000
R32_CAN1_RFIF00	0x4000640C	CAN1 接收 FIF00 控制和状态寄存器	0x00000000
R32_CAN1_RFIF01	0x40006410	CAN1 接收 FIF01 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
R32_CAN1_TERR_CNT	0x40006428	CAN1 离线恢复错误计数器	0x00000000

表 28-7 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	0x0000C002
R32_CAN2_TSTATR	0x40006808	CAN2 发送状态寄存器	0x1C000000
R32_CAN2_RFIF00	0x4000680C	CAN2 接收 FIFO0 控制和状态寄存器	0x00000000
R32_CAN2_RFIF01	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	CAN 时间触发计数值寄存器	0x00000000
R32_CAN2_TERR_CNT	0x40006828	CAN2 离线恢复错误计数器	0x00000000

表 28-8 CAN3 相关寄存器列表

名称	访问地址	描述	复位值
R32_CAN3_CTLR	0x40007800	CAN3 主控制寄存器	0x00010002
R32_CAN3_STATR	0x40007804	CAN3 主状态寄存器	0x0000C002
R32_CAN3_TSTATR	0x40007808	CAN3 发送状态寄存器	0x1C000000
R32_CAN3_RFIF00	0x4000780C	CAN3 接收 FIFO0 控制和状态寄存器	0x00000000
R32_CAN3_RFIF01	0x40007810	CAN3 接收 FIFO1 控制和状态寄存器	0x00000000
R32_CAN3_INTENR	0x40007814	CAN3 中断使能寄存器	0x00000000
R32_CAN3_ERRSR	0x40007818	CAN3 错误状态寄存器	0x00000000
R32_CAN3_BTIMR	0x4000781C	CAN3 位时序寄存器	0x01230000
R32_CAN3_TTCTLR	0x40007820	CAN3 时间触发控制寄存器	0x0000FFFF
R32_CAN3_TTCNT	0x40007824	CAN3 时间触发计数值寄存器	0x00000000
R32_CAN3_TERR_CNT	0x40007828	CAN3 离线恢复错误计数器	0x00000000

表 28-9 CAN1 邮箱相关寄存器列表

名称	访问地址	描述	复位值
R32_CAN1_TXMI0R	0x40006580	CAN1 发送邮箱 0 标识符寄存器	X
R32_CAN1_TXMDT0R	0x40006584	CAN1 发送邮箱 0 数据长度和时间戳寄存器	X
R32_CAN1_TXMDL0R	0x40006588	CAN1 发送邮箱 0 低字节数据寄存器	X
R32_CAN1_TXMDH0R	0x4000658C	CAN1 发送邮箱 0 高字节数据寄存器	X
R32_CAN1_TXMI1R	0x40006590	CAN1 发送邮箱 1 标识符寄存器	X
R32_CAN1_TXMDT1R	0x40006594	CAN1 发送邮箱 1 数据长度和时间戳寄存器	X
R32_CAN1_TXMDL1R	0x40006598	CAN1 发送邮箱 1 低字节数据寄存器	X
R32_CAN1_TXMDH1R	0x4000659C	CAN1 发送邮箱 1 高字节数据寄存器	X

R32_CAN1_TXMI2R	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_TXMDH2R	0x400065AC	CAN1 发送邮箱 2 高字节数据寄存器	X
R32_CAN1_RXMI0R	0x400065B0	CAN1 接收 FIFO0 邮箱标识符寄存器	X
R32_CAN1_RXMDTOR	0x400065B4	CAN1 接收 FIFO0 邮箱数据长度和时间戳寄存器	X
R32_CAN1_RXMDLOR	0x400065B8	CAN1 接收 FIFO0 邮箱低字节数据寄存器	X
R32_CAN1_RXMDHOR	0x400065BC	CAN1 接收 FIFO0 邮箱高字节数据寄存器	X
R32_CAN1_RXMI1R	0x400065C0	CAN1 接收 FIFO1 邮箱标识符寄存器	X
R32_CAN1_RXMDT1R	0x400065C4	CAN1 接收 FIFO1 邮箱数据长度和时间戳寄存器	X
R32_CAN1_RXMDL1R	0x400065C8	CAN1 接收 FIFO1 邮箱低字节数据寄存器	X
R32_CAN1_RXMDH1R	0x400065CC	CAN1 接收 FIFO1 邮箱高字节数据寄存器	X

表 28-10 CAN2 邮箱相关寄存器列表

名称	访问地址	描述	复位值
R32_CAN2_TXMI0R	0x40006980	CAN2 发送邮箱 0 标识符寄存器	X
R32_CAN2_TXMDTOR	0x40006984	CAN2 发送邮箱 0 数据长度和时间戳寄存器	X
R32_CAN2_TXMDLOR	0x40006988	CAN2 发送邮箱 0 低字节数据寄存器	X
R32_CAN2_TXMDHOR	0x4000698C	CAN2 发送邮箱 0 高字节数据寄存器	X
R32_CAN2_TXMI1R	0x40006990	CAN2 发送邮箱 1 标识符寄存器	X
R32_CAN2_TXMDT1R	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_TXMDH1R	0x4000699C	CAN2 发送邮箱 1 高字节数据寄存器	X
R32_CAN2_TXMI2R	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_TXMDH2R	0x400069AC	CAN2 发送邮箱 2 高字节数据寄存器	X
R32_CAN2_RXMI0R	0x400069B0	CAN2 接收 FIFO0 邮箱标识符寄存器	X
R32_CAN2_RXMDTOR	0x400069B4	CAN2 接收 FIFO0 邮箱数据长度和时间戳寄存器	X
R32_CAN2_RXMDLOR	0x400069B8	CAN2 接收 FIFO0 邮箱低字节数据寄存器	X
R32_CAN2_RXMDHOR	0x400069BC	CAN2 接收 FIFO0 邮箱高字节数据寄存器	X
R32_CAN2_RXMI1R	0x400069C0	CAN2 接收 FIFO1 邮箱标识符寄存器	X
R32_CAN2_RXMDT1R	0x400069C4	CAN2 接收 FIFO1 邮箱数据长度和时间戳寄存器	X
R32_CAN2_RXMDL1R	0x400069C8	CAN2 接收 FIFO1 邮箱低字节数据寄存器	X
R32_CAN2_RXMDH1R	0x400069CC	CAN2 接收 FIFO1 邮箱高字节数据寄存器	X

表 28-11 CAN3 邮箱相关寄存器列表

名称	访问地址	描述	复位值
R32_CAN3_TXMI0R	0x40007980	CAN3 发送邮箱 0 标识符寄存器	X
R32_CAN3_TXMDTOR	0x40007984	CAN3 发送邮箱 0 数据长度和时间戳寄存器	X

R32_CAN3_TXMDLOR	0x40007988	CAN3 发送邮箱 0 低字节数据寄存器	X
R32_CAN3_TXMDHOR	0x4000798C	CAN3 发送邮箱 0 高字节数据寄存器	X
R32_CAN3_TXMI1R	0x40007990	CAN3 发送邮箱 1 标识符寄存器	X
R32_CAN3_TXMDT1R	0x40007994	CAN3 发送邮箱 1 数据长度和时间戳寄存器	X
R32_CAN3_TXMDLR1	0x40007998	CAN3 发送邮箱 1 低字节数据寄存器	X
R32_CAN3_TXMDH1R	0x4000799C	CAN3 发送邮箱 1 高字节数据寄存器	X
R32_CAN3_TXMI2R	0x400079A0	CAN3 发送邮箱 2 标识符寄存器	X
R32_CAN3_TXMDT2R	0x400079A4	CAN3 发送邮箱 2 数据长度和时间戳寄存器	X
R32_CAN3_TXMDL2R	0x400079A8	CAN3 发送邮箱 2 低字节数据寄存器	X
R32_CAN3_TXMDH2R	0x400079AC	CAN3 发送邮箱 2 高字节数据寄存器	X
R32_CAN3_RXMI0R	0x400079B0	CAN3 接收 FIFO0 邮箱标识符寄存器	X
R32_CAN3_RXMDTOR	0x400079B4	CAN3 接收 FIFO0 邮箱数据长度和时间戳寄存器	X
R32_CAN3_RXMDLOR	0x400079B8	CAN3 接收 FIFO0 邮箱低字节数据寄存器	X
R32_CAN3_RXMDHOR	0x400079BC	CAN3 接收 FIFO0 邮箱高字节数据寄存器	X
R32_CAN3_RXMI1R	0x400079C0	CAN3 接收 FIFO1 邮箱标识符寄存器	X
R32_CAN3_RXMDT1R	0x400079C4	CAN3 接收 FIFO1 邮箱数据长度和时间戳寄存器	X
R32_CAN3_RXMDL1R	0x400079C8	CAN3 接收 FIFO1 邮箱低字节数据寄存器	X
R32_CAN3_RXMDH1R	0x400079CC	CAN3 接收 FIFO1 邮箱高字节数据寄存器	X

表 28-12 CAN 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_FMCFGR1	0x40006608	CAN1 过滤器模式寄存器 1	0x00000000
R32_CAN1_FSCFGR	0x4000660C	CAN1 过滤器位宽寄存器	0x00000000
R32_CAN1_FSCFGR1	0x40006610	CAN1 过滤器位宽寄存器 1	0x00000000
R32_CAN1_FAFIFOR	0x40006614	CAN1 过滤器 FIFO 关联寄存器	0x00000000
R32_CAN1_FAFIFOR1	0x40006618	CAN1 过滤器 FIFO 关联寄存器 1	0x00000000
R32_CAN1_FWR	0x4000661C	CAN1 过滤器激活寄存器	0x00000000
R32_CAN1_FWR1	0x40006620	CAN1 过滤器激活寄存器 1	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
R32_CAN1_F28R1	0x40006720	CAN1 过滤器组 28 寄存器 1	X

R32_CAN1_F28R2	0x40006724	CAN1 过滤器组 28 寄存器 2	X
R32_CAN1_F29R1	0x40006728	CAN1 过滤器组 29 寄存器 1	X
R32_CAN1_F29R2	0x4000672C	CAN1 过滤器组 29 寄存器 2	X
R32_CAN1_F30R1	0x40006730	CAN1 过滤器组 30 寄存器 1	X
R32_CAN1_F30R2	0x40006734	CAN1 过滤器组 30 寄存器 2	X
R32_CAN1_F31R1	0x40006738	CAN1 过滤器组 31 寄存器 1	X
R32_CAN1_F31R2	0x4000673C	CAN1 过滤器组 31 寄存器 2	X
R32_CAN1_F32R1	0x40006740	CAN1 过滤器组 32 寄存器 1	X
R32_CAN1_F32R2	0x40006744	CAN1 过滤器组 32 寄存器 2	X
R32_CAN1_F33R1	0x40006748	CAN1 过滤器组 33 寄存器 1	X
R32_CAN1_F33R2	0x4000674C	CAN1 过滤器组 33 寄存器 2	X
R32_CAN1_F34R1	0x40006750	CAN1 过滤器组 34 寄存器 1	X
R32_CAN1_F34R2	0x40006754	CAN1 过滤器组 34 寄存器 2	X
R32_CAN1_F35R1	0x40006758	CAN1 过滤器组 35 寄存器 1	X
R32_CAN1_F35R2	0x4000675C	CAN1 过滤器组 35 寄存器 2	X
R32_CAN1_F36R1	0x40006760	CAN1 过滤器组 36 寄存器 1	X
R32_CAN1_F36R2	0x40006764	CAN1 过滤器组 36 寄存器 2	X
R32_CAN1_F37R1	0x40006768	CAN1 过滤器组 37 寄存器 1	X
R32_CAN1_F37R2	0x4000676C	CAN1 过滤器组 37 寄存器 2	X
R32_CAN1_F38R1	0x40006770	CAN1 过滤器组 38 寄存器 1	X
R32_CAN1_F38R2	0x40006774	CAN1 过滤器组 38 寄存器 2	X
R32_CAN1_F39R1	0x40006778	CAN1 过滤器组 39 寄存器 1	X
R32_CAN1_F39R2	0x4000677C	CAN1 过滤器组 39 寄存器 2	X
R32_CAN1_F40R1	0x40006780	CAN1 过滤器组 40 寄存器 1	X
R32_CAN1_F40R2	0x40006784	CAN1 过滤器组 40 寄存器 2	X
R32_CAN1_F41R1	0x40006788	CAN1 过滤器组 41 寄存器 1	X
R32_CAN1_F41R2	0x4000678C	CAN1 过滤器组 41 寄存器 2	X

注：CAN 过滤表的时钟由 CAN1 提供。因此，要配置或者使用过滤表功能，必须使能 CAN1 的时钟。

28.7.1 CANx 主控制寄存器 (CANx_CTLR) (x=1/2/3)

偏移地址：0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														CFGCANM	DBF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RST	Reserved							TTCM	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ

位	名称	访问	描述	复位值
[31:18]	Reserved	RO	保留。	0
17	CFGCANM	RW	配置 CAN 离线恢复时间： 1：从离线恢复到正常符合 CAN 协议； 0：从离线恢复到正常更快一些。	0
16	DBF	RW	调试是否禁止 CAN 总线工作：	1

			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。	
15	RST	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: CAN 控制器一直重传至发送成功为止。	0
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。 注: 该位只用于传统 CAN。	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

			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。	

28.7.2 CANx 主状态寄存器 (CANx_STATR) (x=1/2/3)

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	RW1Z	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。 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。	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

28.7.3 CANx 发送状态寄存器 (CANx_TSTATR) (x=1/2/3)

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	表示发送邮箱 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	表示发送邮箱 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	表示发送邮箱 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	表示发送邮箱 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[1:0]	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，表示优先级最低的邮箱号。	0
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	RW1	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	0

			件置 1 清 0，软件写 0 无效。	
15	ABRQ1	RW1	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	R0	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	RW1	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	RW1	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	R0	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：上次发送失败。 软件置 1 清 0，软件写 0 无效。	0
0	RQCP0	RW1	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

28.7.4 CANx 接收 FIFO 0 状态寄存器 (CANx_RFIF00) (x=1/2/3)

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	R0	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	R0	保留。	0
[1:0]	FMP0[1:0]	R0	接收 FIFO_0 报文数目。	0

28.7.5 CANx 接收 FIFO 1 状态寄存器 (CANx_RFIF01) (x=1/2/3)

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	R0	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[1:0]	R0	接收 FIFO_1 报文数目。	0

28.7.6 CANx 中断使能寄存器 (CANx_INTENR) (x=1/2/3)

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 位，不触发错误中断。	0
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：	0

			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。	
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

28.7.7 CANx 错误状态寄存器 (CANx_ERRSR) (x=1/2/3)

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[7:0]	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[7:0]	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
[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;	0

			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。	
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

28.7.8 CANx 位时序寄存器 (CANx_BTMR) (x=1/2/3)

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[3:0]				TS2[3:0]				TS1[3:0]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTR_TS1_T				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
[29:28]	Reserved	RO	保留。	0
[27:24]	SJW[3:0]	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小时间单元数量上限, 实际值为 (SJW[3:0]+1), 范围可设置为 1 到 16 个最小时间单元。	0001b
[23:20]	TS2[3:0]	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[3:0]+1)。	0010b
[19:16]	TS1[3:0]	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[3:0]+1)。	0011b
[15:12]	BTR_TS1_T	RW	对于传统 CAN, CLAS_LONG_TS1=0, 则 TS1 为 TS[3:0] (4bit); CLAS_LONG_TS1=1, 则 TS1 为 TS[1:0]+BTR_TS1_T[15:12] (6bit)。	0
[11:10]	Reserved	RO	保留。	0

[9:0]	BRP[9:0]	RW	最小时间单元长度设置值 $T_q = (BRP[9:0] + 1) \times t_{HCLKS}$	0

28.7.9 CANx 时间触发控制寄存器 (CANx_TTCTLR) (x=1/2/3)

偏移地址: 0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														MODE	TIMRST
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMCMV[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[15:0]	RW	内部计数器计数终值。	0xFFFF

28.7.10 CANx 时间触发计数值寄存器 (CANx_TTCNT) (x=1/2/3)

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[15:0]	RW	时间触发计数值。	0

28.7.11 CANx 离线恢复错误计数器 (CANx_TERR_CNT) (x=1/2/3)

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								TX_ERR_CNT							

位	名称	访问	描述	复位值
[31:9]	Reserved	RO	保留。	0
[8:0]	TX_ERR_CNT	RW	当前离线恢复错误计数值, 修改该计数值可	0

			从离线立即恢复。	

28.7.12 CANx 发送邮箱标识符寄存器 (CANx_TXMlyR) (x=1/2/3, 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[10:0]/EXID[28:18]	RW	标准标识符或扩展标识符的高 11 位。	X
[20:3]	EXID[17:0]	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

28.7.13 CANx 发送邮箱数据长度和时间戳寄存器 (CANx_TXMDTyR) (x=1/2/3, 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[15:0]	RW	用于发送报文 SOF 时刻的 16 位定时器值。	X
[15:9]	Reserved	RO	保留。	0
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。	0
[3:0]	DLC[3:0]	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 64。	0

28.7.14 CANx 发送邮箱低字节数据寄存器 (CANx_TXMDLyR) (x=1/2/3, 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[7:0]	RW	发送数据字节 3 的内容。	X
[23:16]	DATA2[7:0]	RW	发送数据字节 2 的内容。	X
[15:8]	DATA1[7:0]	RW	发送数据字节 1 的内容。	X
[7:0]	DATA0[7:0]	RW	发送数据字节 0 的内容。	X

28.7.15 CANx 发送邮箱高字节数据寄存器 (CANx_TXMDHyR) (x=1/2/3, 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[7:0]	RW	发送数据字节 7 的内容。	X
[23:16]	DATA6[7:0]	RW	发送数据字节 6 的内容。	X
[15:8]	DATA5[7:0]	RW	发送数据字节 5 的内容。	X
[7:0]	DATA4[7:0]	RW	发送数据字节 4 的内容。	X

28.7.16 CANx 接收邮箱标识符寄存器 (CANx_RXMlyR) (x=1/2/3, 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	Reserved	

位	名称	访问	描述	复位值
[31:21]	STID[10:0]/EXIDH[28:18]	RO	标准标识符或扩展标识符的高 11 位。	X
[20:3]	EXIDL[17:0]	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;	X

			0: 选用标准标识符。	
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。	0

28.7.17 CANx 接收邮箱数据长度和时间戳寄存器（CANx_RXMDTyR）（x=1/2/3, 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]								RES	ESI	BRS	DLC[3:0]				

位	名称	访问	描述	复位值
[31:16]	TIME[15:0]	RO	用于接收报文 SOF 时刻的 16 位定时器值。	0
[15:7]	FMI[7:0]	RO	报文所匹配的过滤器编号。	X
6	RES	RO	当前接收帧的 RES 位。	0
5	ESI	RO	当前接收帧的 ESI 位。	0
4	BRS	RO	当前接收帧的 BRS 位。	0
[3:0]	DLC[3:0]	RO	接收报文数据长度。 DLC=0-8: 表示数据帧长度 0 到 8, 远程帧为 0。	X

28.7.18 CANx 接收邮箱低字节数据寄存器（CANx_RXMDLyR）（x=1/2/3, y=0/1）

偏移地址：0x1B8

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

28.7.19 CANx 接收邮箱高字节数据寄存器（CANx_RXMDHyR）（x=1/2/3, y=0/1）

偏移地址：0x1BC

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

28.7.20 CANx 过滤器主控制寄存器 (CANx_FCTL) (x=1)

偏移地址: 0x200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										CAN3SB[5:0]					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CAN2SB[4:0]						Reserved					FINIT	

位	名称	访问	描述	复位值
[31:22]	Reserved	RO	保留。	0x2A
[21:16]	CAN3SB[5:0]	RW	CAN3 过滤器开始组 (CAN3SB[5:0]+1)。 000000: 保留; 000001: CAN3 过滤器组 2~41; 000010: CAN3 过滤器组 3~41; 100111: CAN3 过滤器组 40~41; 101000: CAN3 过滤器组 41; 101001: 保留; 101010: CAN3 不参与过滤器组分配; 其他: 保留。	0x1C
[15:13]	Reserved	RO	保留。	0
[12:8]	CAN2SB[4:0]	RW	CAN2 过滤器开始组 (CAN2SB[4:0])。 00000: CAN1 不参与过滤器组分配; CAN2 过滤器组 0~CAN3SB; 00001: CAN1 过滤器组 0; CAN2 过滤器组 1~CAN2EB; 00010: CAN1 过滤器组 0~1; CAN2 过滤器组 2~CAN2EB; 00011: CAN1 过滤器组 0~2; CAN2 过滤器组 3~CAN2EB; 11011: CAN1 过滤器组 0~26; CAN2 过滤器组 27~CAN2EB; 11100: CAN1 过滤器组 0~27; CAN2 不参与过滤器组分配; 11101: CAN1 过滤器组 0~27; CAN2 过滤器组 29~CAN2EB;	0xE

			 11111: CAN1 过滤器组 0~27; CAN2 过滤器组 31~CAN2EB。 注: CAN3SB[5:0]=42 时, CAN2EB=41; CAN3SB[5:0] 不等于 42 时, CAN2EB=CAN3SB[5:0]。	
[7:1]	Reserved	RO	保留。	0
0	FINIT	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

注: 1. CAN2SB[4:0] 等于 28 时, CAN3SB[5:0] 取值范围 27~40 和 42;

2. CAN2SB[4:0] 不等于 28 时, CAN2SB[4:0] 的取值必需小于等于 CAN3SB[5:0] 的取值。

28.7.21 CANx 过滤器模式寄存器 (CANx_FMCFGR) (x=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, FINIT 为 1 才能写入。 1: 过滤器组 x 的寄存器为标识符列表模式; 0: 过滤器组 x 的寄存器为屏蔽位模式。	0

28.7.22 CANx 过滤器模式寄存器 (CANx_FMCFGR1) (x=1)

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	FBM41	FBM40	FBM39	FBM38	FBM37	FBM36	FBM35	FBM34	FBM33	FBM32	FBM31	FBM30	FBM29	FBM28	

位	名称	访问	描述	复位值
[31:14]	Reserved	RO	保留。	0
[13:0]	FBMx	RW	过滤器组 x 的工作模式控制位, FINIT 为 1 才能写入。 1: 过滤器组 x 的寄存器为标识符列表模式; 0: 过滤器组 x 的寄存器为屏蔽位模式。	0

28.7.23 CANx 过滤器位宽寄存器 (CANx_FSCFGR) (x=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，FINIT 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

28.7.24 CANx 过滤器位宽寄存器 (CANx_FSCFGR1) (x=1)

偏移地址：0x210

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	FSC41	FSC40	FSC39	FSC38	FSC37	FSC36	FSC35	FSC34	FSC33	FSC32	FSC31	FSC30	FSC29	FSC28	

位	名称	访问	描述	复位值
[31:14]	Reserved	RO	保留。	0
[13:0]	FSCx	RW	过滤器组 x 的位宽控制位，FINIT 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

28.7.25 CANx 过滤器 FIFO 关联寄存器 (CANx_FAFIFOR) (x=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，FINIT 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

28.7.26 CANx 过滤器 FIFO 关联寄存器 (CANx_FAFIFOR1) (x=1)

偏移地址: 0x218

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	FFA41	FFA40	FFA39	FFA38	FFA37	FFA36	FFA35	FFA34	FFA33	FFA32	FFA31	FFA30	FFA29	FFA28	

位	名称	访问	描述	复位值
[31:14]	Reserved	RO	保留。	0
[13:0]	FFAx	RW	过滤器组 x 的关联 FIFO 控制位, FINIT 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

28.7.27 CANx 过滤器激活寄存器 (CANx_FWR) (x=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

28.7.28 CANx 过滤器激活寄存器 (CANx_FWR1) (x=1)

偏移地址: 0x220

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	FACT 41	FACT 40	FACT 39	FACT 38	FACT 37	FACT 36	FACT 35	FACT 34	FACT 33	FACT 32	FACT 31	FACT 30	FACT 29	FACT 28	

位	名称	访问	描述	复位值
[31:14]	Reserved	RO	保留。	0
[13:0]	FACTx	RW	过滤器组 x 的激活控制位。 1: 过滤器组 x 激活;	0

			0: 过滤器组 x 禁用。	

28. 7. 29 CANx 过滤器组的过滤寄存器（CANx_FiRy）（x=1, i=0-41, y=1/2）

偏移地址：0x240-0x38C

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，FINIT 为 1 才能写入。 标识符模式： 1：对应位期望电平为隐性位； 0：对应位期望电平为显性位。 屏蔽位模式： 1：必须和对应的标识符寄存器位一致； 0：不需要和对应的标识符寄存器位一致。	0

第 29 章 数字图像接口（DVP）

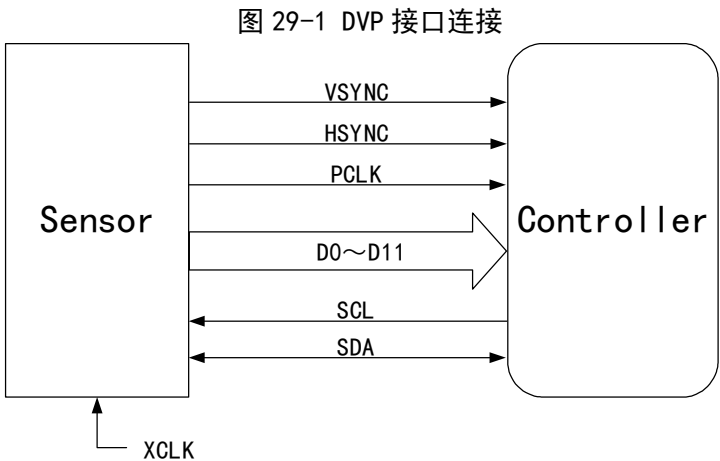
数字图像接口 DVP（Digital Video Port）用来连接摄像头模块获取图像数据流。提供了 8/10/12 位并行接口方式通讯，支持最高 150MHz 像素时钟输入频率。支持按原始的行、帧格式组织的图像数据，如 YUV、RGB 等，也支持如 JPEG 格式的压缩图像数据，能够接收外部 8 位、10 位、12 位的摄像头模块输出的高速并行数据流。接收时，主要依靠 VSYNC 和 HSYNC 信号同步。支持图像裁剪功能。

29.1 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

29.2 功能描述

29.2.1 与传感器相连



- PCLK（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 HCLK 时钟最大支持 150MHz。
- HSYNC（horizontal 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 29-1 DVP 引脚

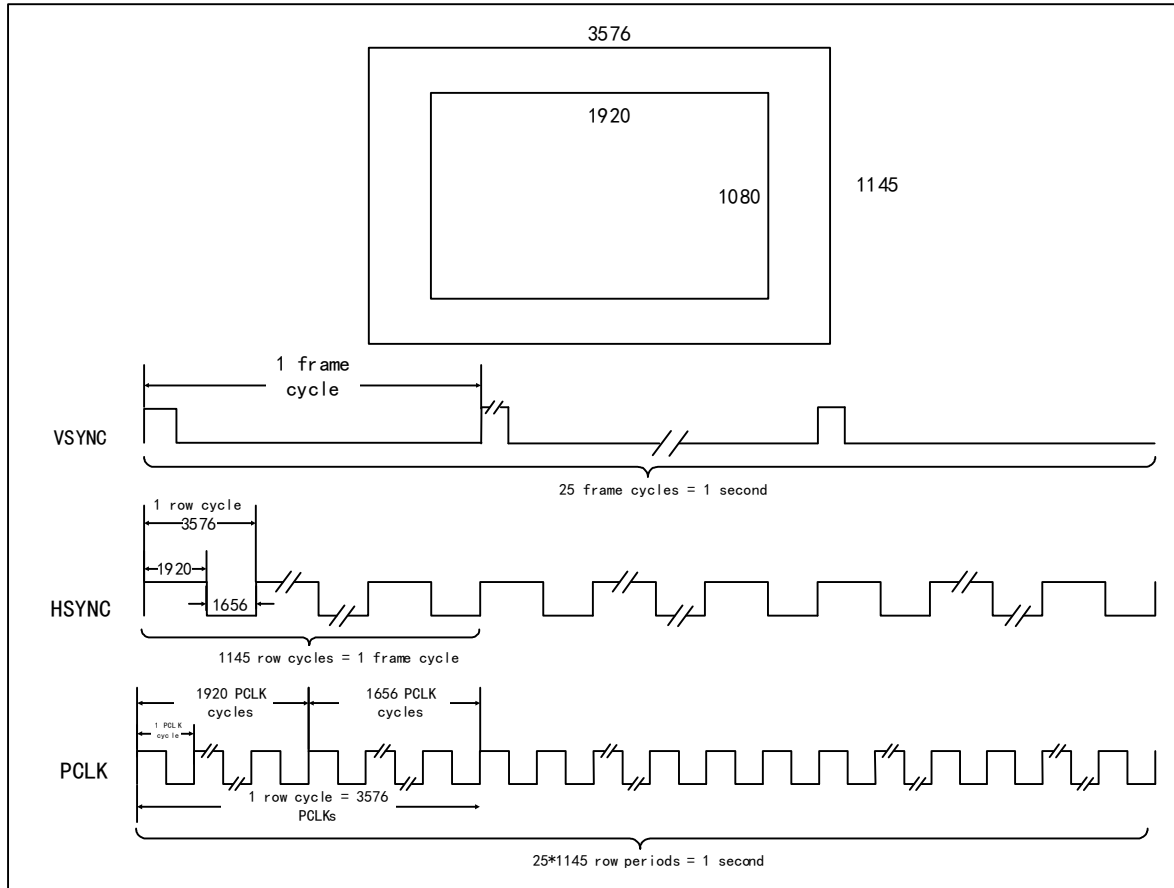
名称	信号类型描述
PCLK	像素时钟输入
DATA[11:0]	像素数据输入
VSYNC	帧同步信号输入
HSYNC	行同步信号输入

29.2.2 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 29-2 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 29-2 时序描述



HCLK 是一个像素传输的时间，所以 HSYNC 是 HCLK 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 HCLK 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。

29.2.3 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）。

29.3 数字图像接口应用

29.3.1 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 HCLK 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。

29.3.2 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、HCLK 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 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。

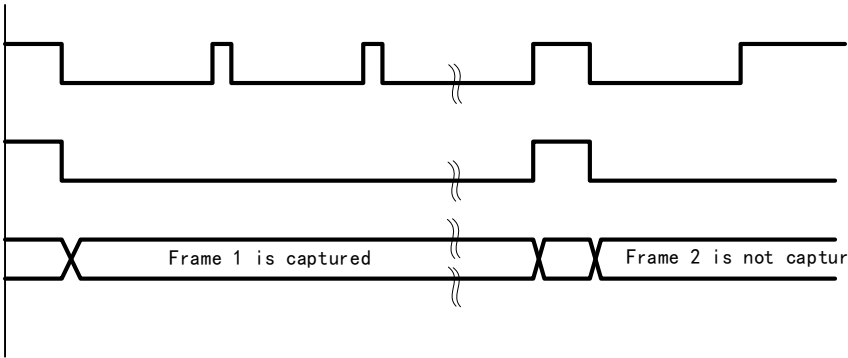
29.3.3 捕获模式说明

DVP 接口支持两种捕获模式：快照（单帧）模式和连续模式。

29.3.3.1 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 29-3 所示。

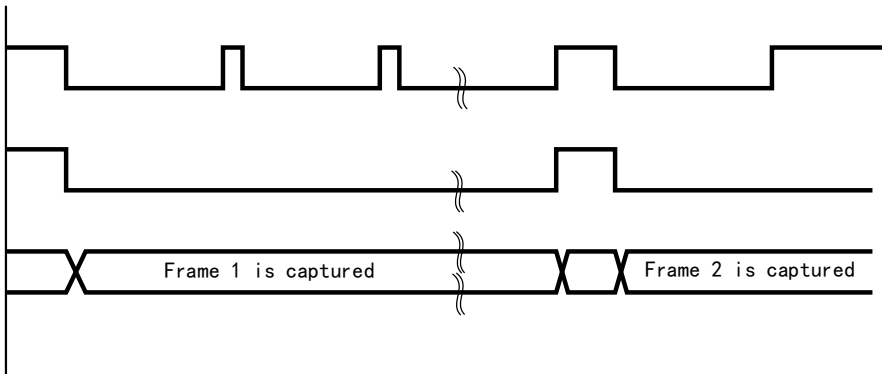
图 29-3 快照模式下的单帧捕获波形



29.3.3.2 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 29-4 所示。

图 29-4 连续模式下帧捕获波形



29.3.4 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 29-5 所示。裁剪窗口数据捕获波形如图 29-6 所示。

图 29-5 裁剪窗口的坐标和大小

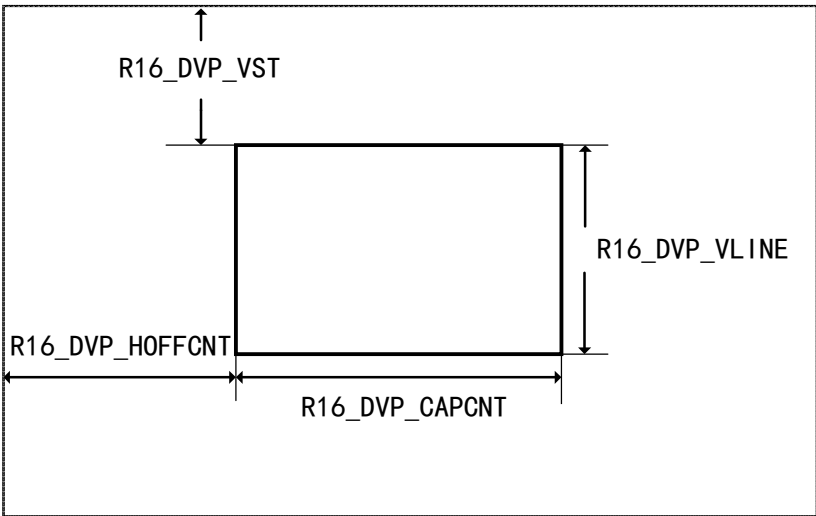
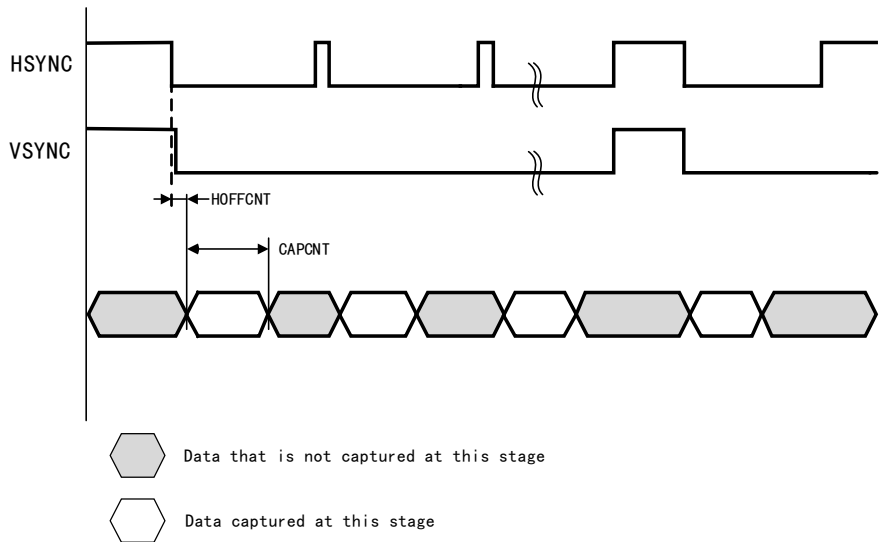


图 29-6 裁剪窗口数据捕获波形



29.4 寄存器描述

表 29-2 DVP 相关寄存器列表

名称	访问地址	描述	复位值
R8_DVP_CR0	0x40025800	DVP 控制寄存器 0	0x00
R8_DVP_CR1	0x40025801	DVP 控制寄存器 1	0x06
R8_DVP_IER	0x40025802	DVP 中断使能寄存器	0x00
R16_DVP_ROW_NUM	0x40025804	图像行数配置寄存器	0x0000
R16_DVP_COL_NUM	0x40025806	图像列数配置寄存器	0x0000
R32_DVP_DMA_BUF0	0x40025808	DVP DMA 地址 0 寄存器	0x00000000
R32_DVP_DMA_BUF1	0x4002580C	DVP DMA 地址 1 寄存器	0x00000000
R8_DVP_IFR	0x40025810	DVP 中断标志寄存器	0x00
R8_DVP_STATUS	0x40025811	DVP 接收 FIFO 状态寄存器	0x00
R16_DVP_ROW_CNT	0x40025814	DVP 行计数器寄存器	0x0000
R16_DVP_HOFFCNT	0x40025818	窗口开始的水平方向位移寄存器	0x0000
R16_DVP_VST	0x4002581A	窗口开始的行数寄存器	0x0000
R16_DVP_CAPCNT	0x4002581C	捕获计数寄存器	0x0000
R16_DVP_VLINE	0x4002581E	垂直行计数寄存器	0x0000
R32_DVP_DR	0x40025820	数据寄存器	0x00000000

29.4.1 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[1:0]	RW	DVP 数据位宽配置： 00：8 位模式； 01：10 位模式； 1x：12 位模式。	00b

3	RB_DVP_P_POLAR	RW	HCLK 极性配置： 1：在 HCLK 下降沿采样数据； 0：在 HCLK 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	VSYSN 极性配置： 1：VSYSN 高电平数据有效； 0：VSYSN 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

29.4.2 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[1:0]	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。	00b
5	RB_DVP_CROP	RW	裁剪功能控制： 1：仅捕获剪裁寄存器所指定的窗口中的数据； 0：捕获完整图像。	0
4	RB_DVP_CM	RW	捕获模式： 1：快照模式； 0：连续模式。	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：数据存储在接受地址 1； 0：数据存储在接受地址 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

29.4.3 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。	0
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 ROW/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

29.4.4 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[15:0]	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

29.4.5 DVP 图像有效列配置寄存器 (R16_DVP_COL_NUM)

偏移地址: 0x06

位	名称	访问	描述	复位值
[15:0]	RB_DVP_COL_NUM[15:0]	RW	在 RGB 模式下, 表示一行数据内包含的 HCLK 周期数。在 JPEG 模式下, 用于配置 DMA 接收长度。	0

29.4.6 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
[31:0]	RB_DVP_DMA_BUF0[31:0]	RW	DMA 接收地址 0 (32 字节对齐)。	0

29.4.7 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
[31:0]	RB_DVP_DMA_BUF1[31:0]	RW	DMA 接收地址 1（32 字节对齐）。	0

29.4.8 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	R0	保留。	0
4	RB_DVP_IF_STP_FRM	RW1Z	帧结束中断标志，高有效，写 1 清除。	0
3	RB_DVP_IF_FIFO_OV	RW1Z	接收 FIFO 溢出中断标志，高有效，写 1 清除。	0
2	RB_DVP_IF_FRM_DONE	RW1Z	帧接收完成中断标志，高有效，写 1 清除。	0
1	RB_DVP_IF_ROW_DONE	RW1Z	行结束中断标志，高有效，写 1 清除。	0
0	RB_DVP_IF_STR_FRM	RW1Z	帧开始中断标志，高有效，写 1 清除。	0

29.4.9 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	R0	保留。	0
[6:4]	RB_DVP_FIFO_CNT[2:0]	R0	FIFO 计数器。	0
3	Reserved	R0	保留。	0
2	RB_DVP_FIFO_OV	R0	FIFO 溢出状态： 1：FIFO 溢出； 0：FIFO 未溢出。	0
1	RB_DVP_FIFO_FULL	R0	FIFO 满状态： 1：缓存已满； 0：FIFO 未滿。	0
0	RB_DVP_FIFO_RDY	R0	FIFO 就绪状态： 1：FIFO 中有数据； 0：FIFO 中无数据。	0

29.4.10 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[15:0]	R0	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

29.4.11 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[15:0]	RW	窗口行内，每行在捕获数据前需要空出 HCLK 周期数。低 14 位有效。	0

29.4.12 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[15:0]	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

29.4.13 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[15:0]	RW	裁剪窗口内需要捕获的 HCLK 周期数。低 14 位有效。	0

29.4.14 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[15:0]	RW	裁剪窗口内需要捕获的行数。低 14 位有效。	0

29.4.15 DVP 数据寄存器 (R32_DVP_DR)

偏移地址: 0x20

位	名称	访问	描述	复位值
[31:0]	RB_DVP_DR[31:0]	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

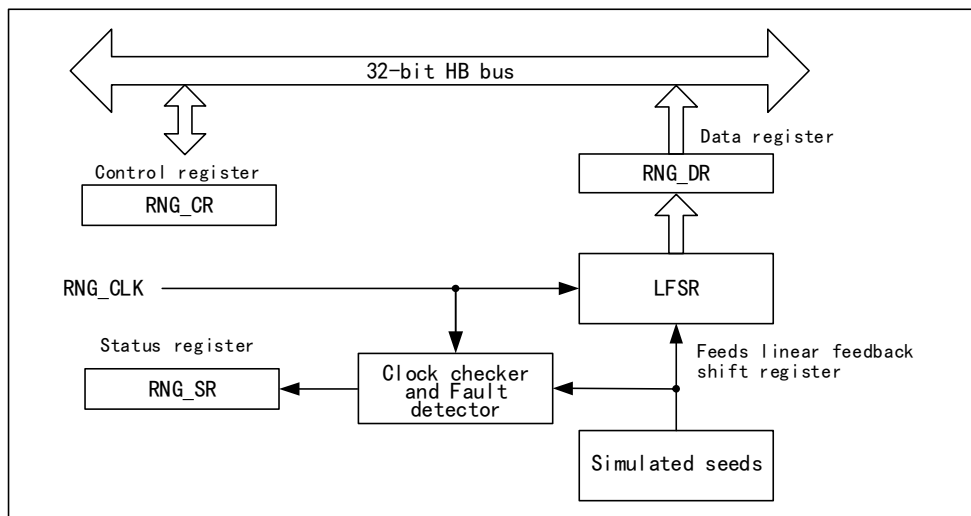
第 30 章 随机数发生器（RNG）

芯片内置一个硬件随机数发生器，以连续模拟噪声为基础，它通过内部的模拟电路提供一个 32 位的随机数。

30.1 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 30-1 RNG 模块框图



30.2 功能描述

随机数发生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（R32_RNG_DR）。

30.2.1 RNG 操作

RNG 具体操作步骤如下：

- 1) 若使能中断，需通过将 R32_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 R32_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 R32_RNG_SR 寄存器中的 SEIS 和 CEIS 位为 0 确定未出现错误且 DRDY 为 1 确定随机数已准备就绪。之后即可读取 R32_RNG_DR 寄存器中的内容。

30.2.2 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 R32_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。

30.3 寄存器描述

表 30-1 RNG 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

30.3.1 RNG 控制寄存器 (R32_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: 1: 使能 RNG 中断; 0: 禁止 RNG 中断。	0
2	RNGEN	RW	随机数发生器使能: 1: 使能随机数发生器; 0: 禁止随机数发生器。	0
[1:0]	Reserved	RW	保留。	0

30.3.2 RNG 状态寄存器 (R32_RNG_SR)

偏移地址: 0x04

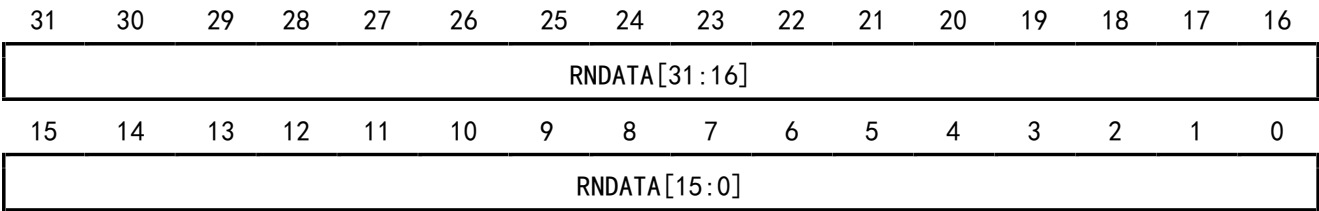
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									SEIS	CEIS	Reserved	SECS	CECS	DRDY	

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): 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: 未检测倒错误序列。	0
5	CEIS	RW	时钟错误中断状态 (此位与 CECS 同时设置): 1: 未正确检测到 PLL48CLK 时钟; 0: 正确检测到 PLL48CLK 时钟。	0

[4:3]	Reserved	R0	保留。	0
2	SECS	R0	种子错误当前状态： 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：未检测出错误序列。	0
1	CECS	R0	时钟错误当前状态： 1：未正确检测到 PLL48CLK 时钟； 0：正确检测到 PLL48CLK 时钟。	0
0	DRDY	R0	数据就绪（读取 R32_RNG_DR 寄存器后，该位清 0）： 1：R32_RNG_DR 寄存器有效，此随机数可用； 0：R32_RNG_DR 寄存器无效，此随机数不可用。	0

30.3.3 RNG 数据寄存器（R32_RNG_DR）

偏移地址：0x08



位	名称	访问	描述	复位值
[31:0]	RNDATA	R0	32 位随机数。	0

第 31 章 以太网收发器（ETH）

本章模块描述仅适用于 CH32H417 微控制器产品。

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。

31.1 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）、256 位宽的 DMA 控制器、管理计数器（MMC）、精确时间协议控制器（PTP）、一个十兆/百兆速度的以太网物理层（10M/100M 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 RGMII 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 256 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。

31.1.1 MAC 特征

- 支持 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 10M/100M/1000Mbps 的数据传输速率
- 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
- 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、HASH 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
- 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
- 内置预分配的全球唯一 MAC 地址
- 应用可选以太网控制器 MAC+外置 1Gbps PHY，或以太网控制器 MAC+内置 10Mbps/100Mbps PHY

31.1.2 DMA 特征

- 256 位宽的 MAC 专享 DMA

- 最大程度减小 CPU 的操作
- 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

31.1.3 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

31.1.4 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

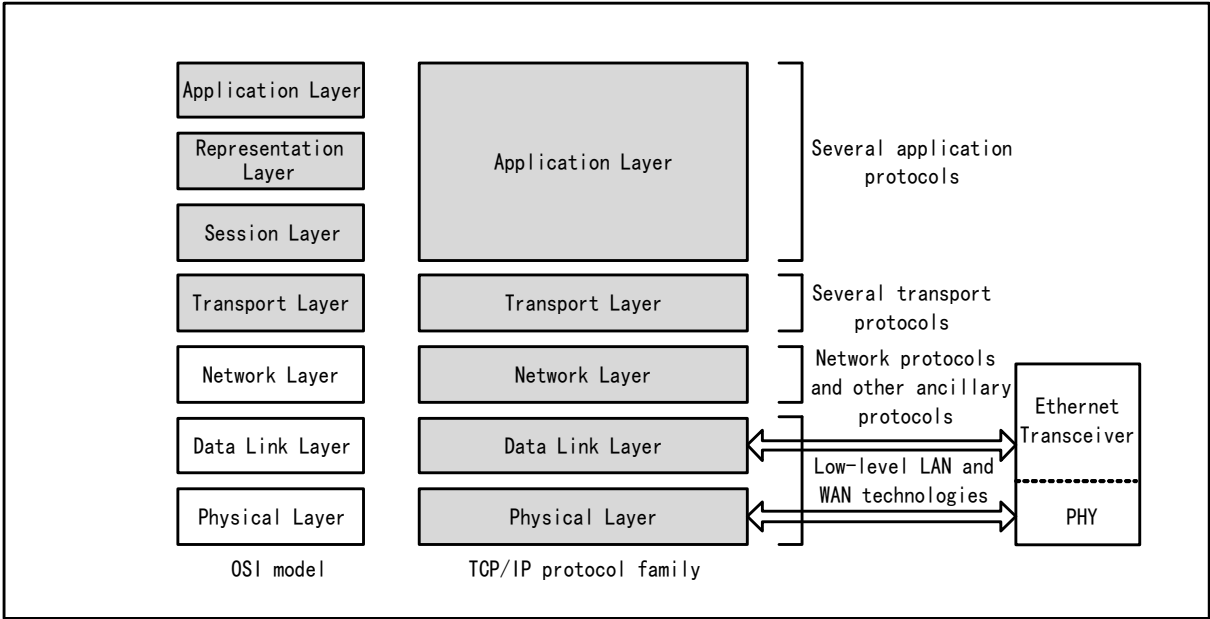
31.1.5 内部 10M/100M 物理层特征

- 支持 10BASE-T 和 100BASE-TX 及自动协商
- 支持 Auto-MDIX 交换 TX/RX，自动识别正负信号线
- 支持支持 WOL 网络唤醒
- 支持中断功能
- 支持半双工、全双工
- 五种网络状态 LED

31.2 概述

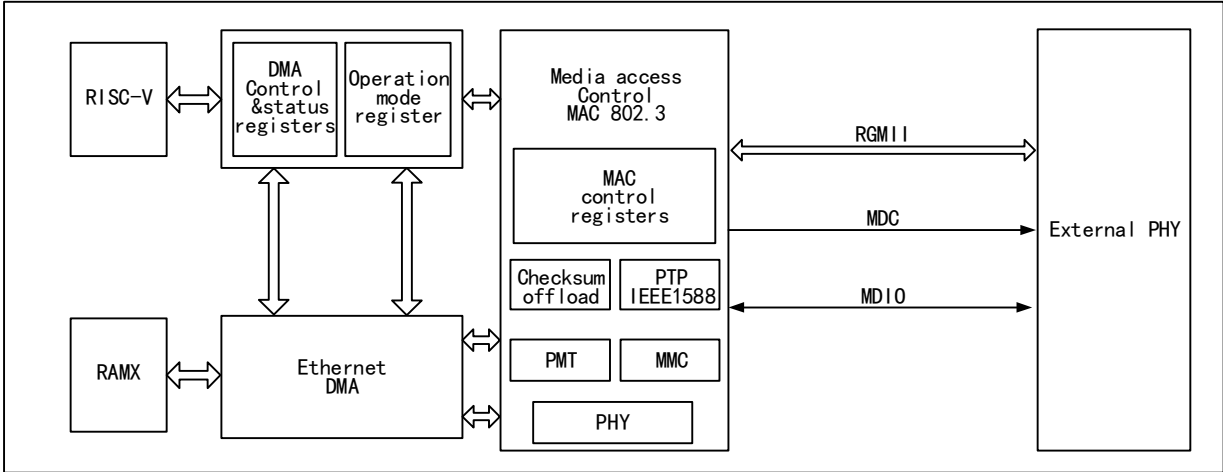
以太网收发器工作在 OSI 七层模型中的数据链路层和物理层。为了能在广泛使用的以太网中建立 IP、TCP 和 UDP 等协议的通讯，用户还需要用软件实现 TCP/IP 协议栈。以太网收发器由媒体访问控制层（MAC）、搭配的 DMA 及二者的控制寄存器及内部物理层组成。

图 31-1 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 256 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 HB 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。以太网收发器的 RGMII 接口的管脚是复用的，具体参看 31.3 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，HB 总线的时钟不能低于 50MHz。

图 31-2 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。

31.3 以太网收发器使用引脚的分布和配置

微控制器支持 RGMII 接口。RGMII 接口可用于十兆、百兆和千兆以太网。下表显示了标准 RGMII 接口及内部物理层在封装引脚上的分布。

表 31-1 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
PD15	GTXC	推挽复用输出
PA14	RXDV	浮空输入
PA15	GRXC	浮空输入
PC6	RXD3	浮空输入
PD13	TXD0	推挽复用输出
PD12	TXD1	推挽复用输出
PD11	TXD2	推挽复用输出
PD10	TXD3	推挽复用输出
PC8	RXD1	浮空输入
PC0	MDC	推挽复用输出
PC1	MDIO	推挽复用输出
PC9	RXD0	浮空输入
PC7	RXD2	浮空输入
PD14	TXEN	推挽复用输出
PC2	PPS	推挽复用输出
-	MDITP	无需 IO 配置
-	MDITN	无需 IO 配置
-	MDIRP	无需 IO 配置
-	MDIRN	无需 IO 配置

31.4 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。微控制器在 RGMII 模式下使用不同管脚引出 SMI 接口。

31.4.1 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_MACMII DR）。

31.4.1.1 帧格式

SMI 接口管理帧的格式如下表 31-2 所示。

表 31-2 SMI 帧格式

	Preamble	STR	OP	PHY ADR	REG ADR	T	DATA	P
读	32 个 1	01	10	PPPPP	RRRRR	Z0	DDDDDDDDDDDDDDDDDD	Z
写	32 个 1	01	01	PPPPP	RRRRR	10	DDDDDDDDDDDDDDDDDD	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。

31.4.1.2 读写时序

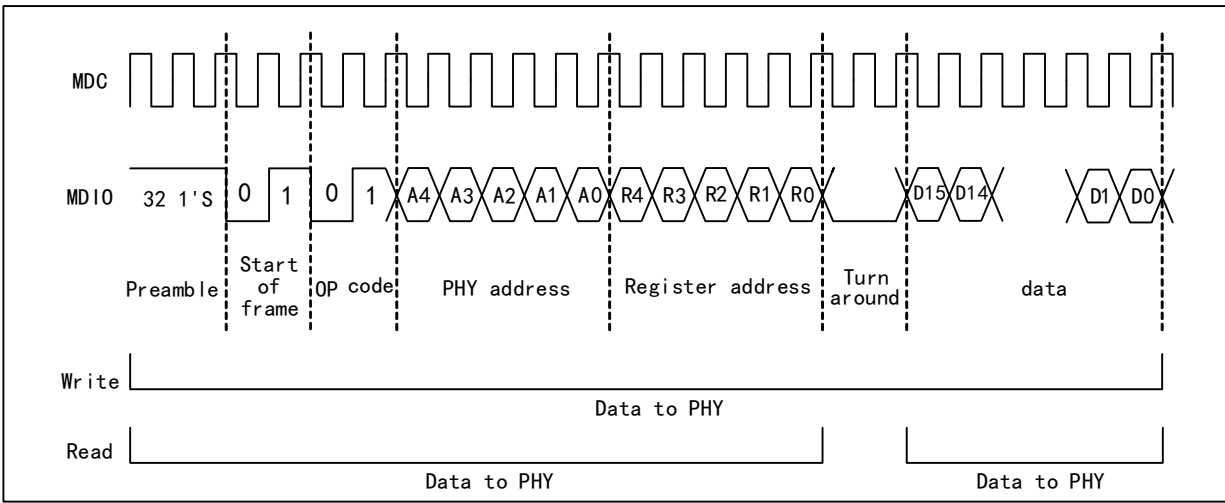
写 PHY 寄存器操作如下:

当用户设置了 MII 写位 (R32_ETH_MACMIAR: MW) 和忙位 (R32_ETH_MACMIAR: MB), SMI 接口会向 PHY 发送 PHY 地址和寄存器地址, 然后发送数据 (R32_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 31-3 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 (R32_ETH_MACMIAR) 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 31-3 的读部分。

图 31-3 SMI 接口读写时序图



31.4.1.3 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 (R32_ETH_MACMIAR) 的 CR 域从 HB 时钟分频得到。下表显示和 SMI 时钟和 HB 时钟的分频关系。默认选择 42 分频。

表 31-3 SMI 时钟和 HB 时钟的分频关系

	适合搭配的 HB 时钟的范围	SMI 时钟
0000b	60MHz 以上	HB 时钟/42
0010b	20-35MHz	HB 时钟/16
0011b	35-60MHz	HB 时钟/26

其它值	无意义

31.4.2 RGMII 接口

31.4.2.1 引脚

RGMII 的引脚和功能如下表

表 31-4 RGMII 的引脚和功能

引脚	功能
GTXC	发送时钟
TXCTL	发送数据控制
TXD0	发送数据线[0:3]
TXD1	
TXD2	
TXD3	
GRXC	接收时钟
RXCTL	接收数据控制
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。

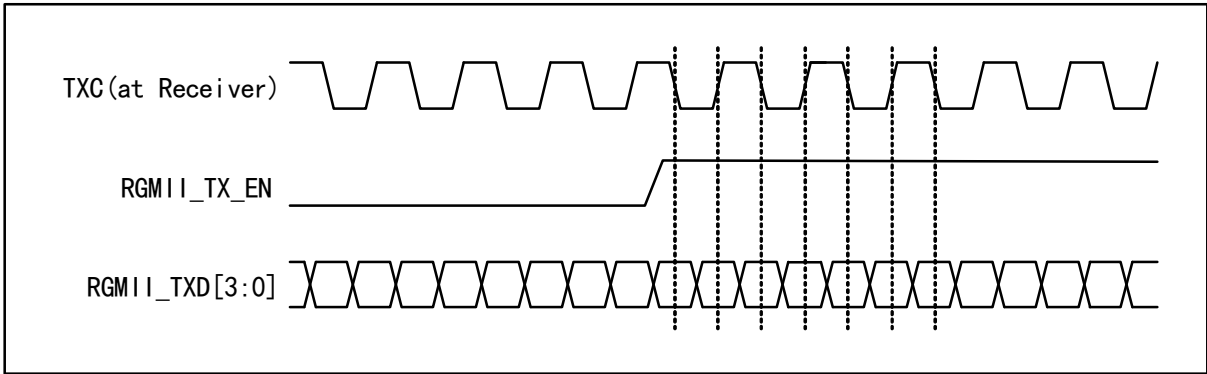
31.4.2.2 时序

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。

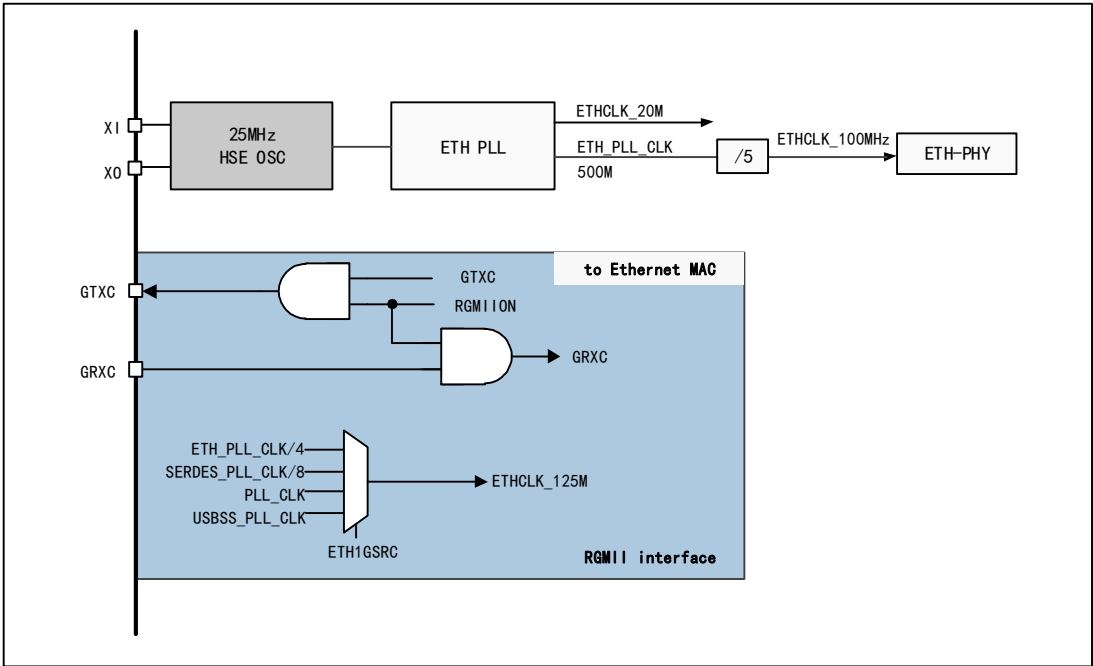
RGMII 的接收方接收到的时钟应相对数据滞后 90° 以保证正确采样。以太网收发器设计了发送时钟延迟输出和相位翻转的功能。

图 31-4 RGMII 的时序示意图



31. 4. 3 外设时钟源配置

图 31-5 以太网外设时钟树



在使用 RGMII 时，需将 RCC_CFGR0[21]置 1，以使能 RGMII 模式，同时设置 RCC_CFG2[31:30]选择合适的 125MHz 时钟源提供给 MAC。RGMII 的发送时钟 GTXC 由 MAC 产生，接收时钟 GRXC 由 PHY 产生。详见官网 EVT 例程。

31. 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。

31. 5. 1 帧格式

以太网帧的帧格式如表 31-7 所示。

表 31-7 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
7Bytes	1Byte	6Bytes	6Bytes	2Bytes	46-1500 Bytes	4Bytes
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。

31.5.2 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 31.5.1 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，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，会自动在数据域后加上填充，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：

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

在使能了 IEEE 1588（PTP）功能的情况下，需要使用增强型描述符进行时间戳的储存。

31.5.3 帧接收

以太网收发器在接收以太网帧的时候，以太网帧从 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，需要使用增强型描述符进行时间戳的储存。

31.5.4 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。

31.5.4.1 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。

31.5.4.2 多播过滤

MAC 通过 HPF 位和 HM 位来确定多播帧进行 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。

31.5.4.3 广播过滤

通过置位 BFD 位，MAC 可以阻断所有的广播包。

31.5.4.4 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。

31.5.4.5 小结

对目标地址和源地址的过滤设置分别如表 31-8 和表 31-9。

表 31-8 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	PM	HPF	HU	DAIF	HM	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	–	–	HASH 滤波匹配时通过
	0	0	1	1	–	–	HASH 滤波匹配时不通过
	0	1	1	0	–	–	完美滤波或 HASH 滤波匹配时通过
	0	1	1	1	–	–	完美滤波或 HASH 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	HASH 滤波匹配时通过
	0	1	–	0	1	0	完美滤波或 HASH 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	HASH 滤波匹配时不通过
	0	1	–	1	1	0	完美滤波或 HASH 滤波匹配时不通过

表 31-9 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	完美过滤器匹配时通过，标记但不丢弃未通过的帧
	0	1	0	完美过滤器匹配时不通过，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。

31.5.5 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（MMCRGUFGR）获取当前接收到的完好的帧的数量，通过发送“好”帧计数器（MMCTGFGFR）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（MMCRFCEGR）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。

31.5.6 PMT

31.5.6.1 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。

31.5.6.2 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 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 84 c2 e4 01 02 02 xx xx xx xx xx xx (有的网卡要求魔法帧最后附上密码)。
```

31.5.6.3 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。要是通过远程唤醒帧过滤寄存器（R32_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 31-10 远程唤醒帧过滤寄存器组的结构

过滤寄存器 0	过滤寄存器 0 的字节掩码							
过滤寄存器 1	过滤寄存器 1 的字节掩码							
过滤寄存器 2	过滤寄存器 2 的字节掩码							
过滤寄存器 3	过滤寄存器 3 的字节掩码							
过滤寄存器 4	保留	过滤器 3 命令	保留	过滤器 2 命令	保留	过滤器 1 命令	保留	过滤器 0 命令
过滤寄存器 5	过滤器 3 偏移		过滤器 2 偏移		过滤器 1 偏移		过滤器 0 偏移	
过滤寄存器 6	过滤器 1 CRC-16				过滤器 0 CRC-16			
过滤寄存器 7	过滤器 3 CRC-16				过滤器 2 CRC-16			

由上表可以看出，过滤寄存器的字节掩码、命令、偏移、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。

32 位的字节掩码最高位必须为 0，bit[30:0] 是对应由偏移域定义的开头的 31 字节数据，此中若有 bit 置 1，则表明对应字节参与 CRC-16 校验，且最多有 31 个字节参与计算；

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，如若偏移域为 12，则为从开头第 13 个字节开始计算 CRC16 的值（参数模型使用 CRC-16-IBM，参数默认）。

过滤器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，如果一致，则认定为（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 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。

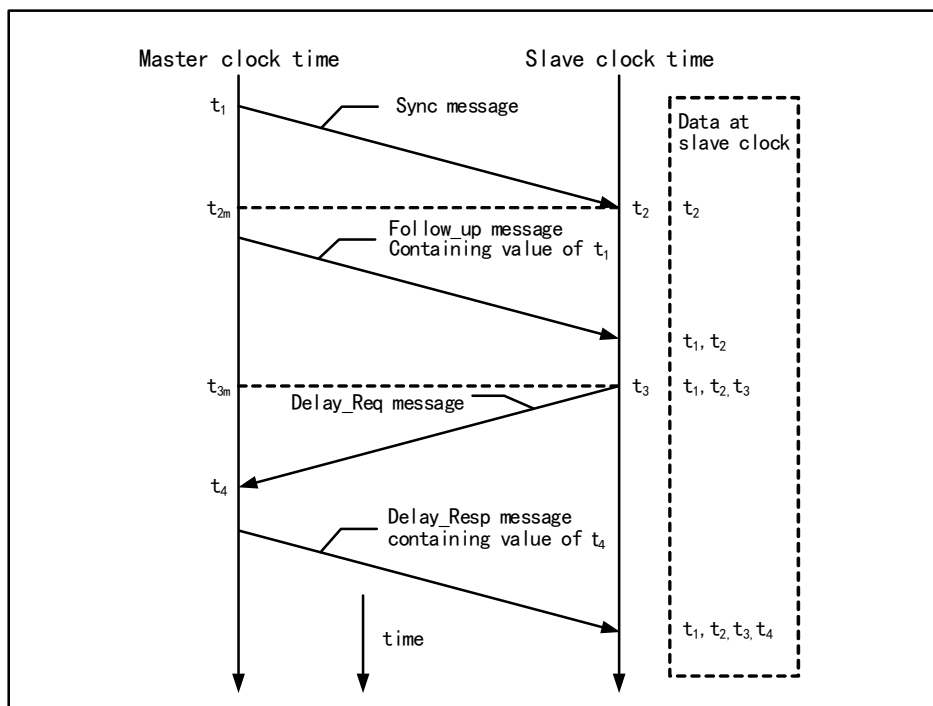
31.5.7 IEEE1588 PTP

31.5.7.1 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 31-6 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 t_2 ；

- 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 t_1 ;
- 从机向主机发送 delay_req 消息，从机记录下发送时间 t_3 ;
- 主机向从机发送 delay_resq 消息，包含 delay_req 消息的接收时间 t_4 ;

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 T_{delay} ，然后算出主机时间和从机时间的时间偏移。

$$T_{\text{delay}} = \frac{(t_2 - t_1) + (t_4 - t_3)}{2}$$

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。

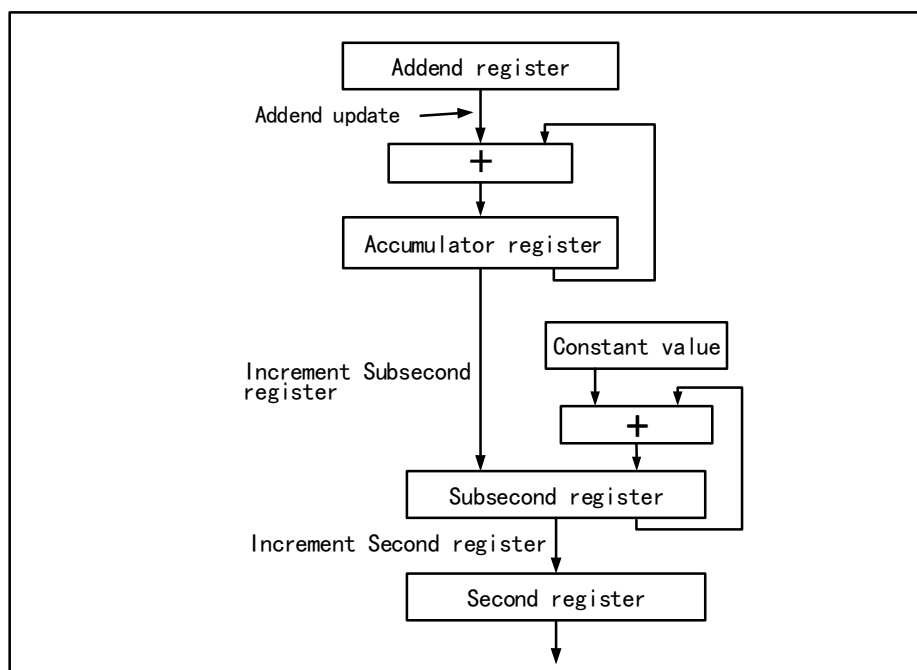
31.5.7.2 本地时间更新与校正

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 31-7 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 Accumulator 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 $1/(2^{31}) = 0.46566128730 \dots$ 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。

31.6 DMA 操作

31.6.1 概述

在以太网收发器中，数据从 RGMII 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 256 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 256 位宽的，所以要求收发描述符队列在 RAM 中的基地址是 32 字节对齐的。

31.6.2 DMA 描述符（DMA Descriptor）

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 USART 或 SPI，都有一个数据寄存器来暂存收发数据，有一组 DMA 将收到的所有数据统一保存到特定的地址空间。

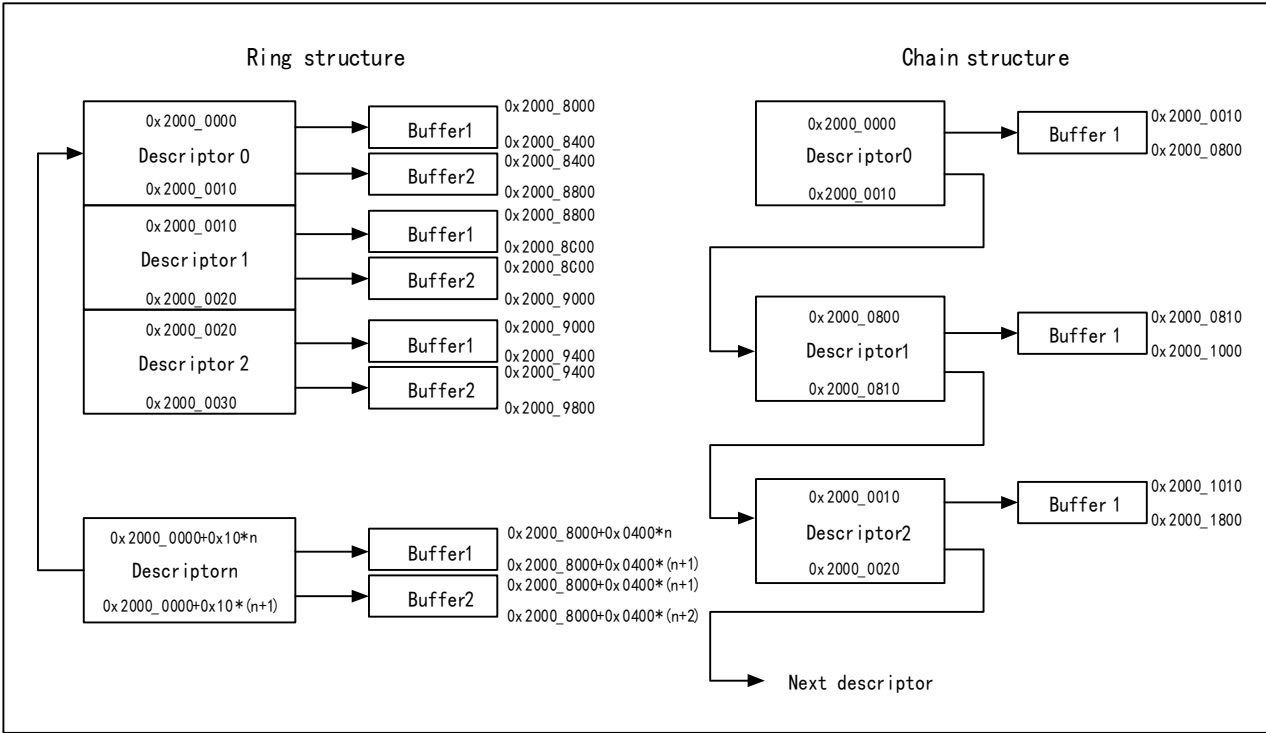
而以太网收发器以其极高的数据传输速度、极大的数据吞吐量和独特的数据帧组织形式，使得它的操作不同于传统的通讯收发器。以太网收发器尽可能密集地接收大量的数据，它必须将收到的数据流以帧为单位存到单独的内存空间，自行完成接收和发送操作，使 CPU 能以最少的操作就能读取并处理掉这些数据，释放出对应的内存空间，并保证 DMA 控制器和 CPU 不会就内存的使用权限起冲突。内存中被开辟用来暂存以太网帧的缓冲区单个大小一般设置为以太网帧的最大包长，IEEE802.3 规定为 1518 个字节（包括源止硬件地址、长度或类型域和 CRC32 校验域共 18 个字节）；以太网帧收发缓冲区的数量根据实际交互的频度和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 32 字节对齐，单个常规描述符的大小为 16 字节，单个增强描述符的大小为 32 字节。

图 31-8 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。

图 31-8 环式结构和链式结构的缓冲区分配方案



31.6.2.1 发送 DMA 描述符

表 31-11、31-12 表明了发送描述符的结构。

表 31-11 常规发送描述符的结构

	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							
TDes3	缓冲区 2 地址/下一个描述符的地址							

表 31-12 增强的发送描述符的结构

	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								
TDes3	缓冲区 2 地址/下一个描述符的地址								
TDes4	保留								

TDes5	保留
TDes6	时间戳低位
TDes7	时间戳高位

由上表可以看出常规发送描述符由四个 32 位字组成，分别是 TDes0、TDes1、TDes2 和 TDes3，其中 TDes0 用作控制和返回发送状态，TDes1 用以指示发送长度，TDes2 用以表示发送缓冲区的位置，TDes3 用以表示用于第二个发送缓冲区的地址（TCH 未置位时）。

增强发送描述符与常规发送描述符相比，增加了 TDes4、TDes5、TDes6、TDes7 四个 32 位字。其中，在启用 IEEE1588 功能时，TDes6 用以表示时间戳低位，TDes7 用以表示时间戳高位。

各 32 位字的描述如下。

表 31-13 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	LC0	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。 1：此描述符归 DMA 所有，CPU 无权限修改此描述符； 0：此描述符归 CPU 所有，CPU 可以修改此描述符的值。 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 控制器将在当前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校验和的计算和插入, 使能计算伪报头校验和。
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, 并存放于 TDes6 和 TDes7 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 控制器会检查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 (Jabber timeout); IHE[TDES0:16] IP 报头错误位。
14	JT	啰嗦超时位 (Jabber timeout)。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 31-14 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。（长度要求为 32 的倍数）
[15:13]	Reserved	未使用。
[12:0]	TBS1	发送缓冲区 1 的大小。（长度要求为 32 的倍数）

表 31-15 TDes2 的各位定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TBAD1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBAD1															

位	名称	描述
[31:0]	TBAD1	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。（缓冲区起始地址需 32 字节对齐）

表 31-16 TDes3 的各位定义

31	30	29	28	27	26	25	24	23	22	21	20	19	20	17	16
TDAD2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDAD2															

位	名称	描述
[31:0]	TDAD2	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。（缓冲区起始地址需 32 字节对齐）

31.6.2.2 接收 DMA 描述符

表 31-17 常规接收描述符的结构

	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					
RDes3	缓冲区 2 地址、下一个描述符的地址					

表 31-18 增强的接收描述符的结构

	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						
RDes3	缓冲区 2 地址、下一个描述符的地址						
RDes4	保留						
RDes5	保留						
RDes6	时间戳低位						
RDes7	时间戳高位						

由上表可以看出常规接收描述符也是由 RDes0、RDes1、RDes2、RDes3 四个 32 位字组成的，其中 RDes0 主要是返回接收时的状态，RDes1 包含了接收的数据长度，RDes2 定义了接收缓冲区的地址，RDes3 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）。

增强接收描述符与常规接收描述符相比，增加了 RDes4、RDes5、RDes6、RDes7 四个 32 位字。其中，在启用 IEEE1588 功能时，RDes6 用以表示时间戳低位，RDes7 用以表示时间戳高位。

接收描述符各字的各位意义如下：

表 31-19 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	LC0	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。 1：此描述符归 DMA 所有，CPU 无权限修改此描述符； 0：此描述符归 CPU 所有，CPU 可以修改此描述符的值。 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[RDs0:15]为 0 时有效。在 LS 位[RDs0:8]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[RDs0:1]：CRC 错误； RE[RDs0:3]：接收错误； RWT[RDs0:4]：看门狗超时； IPHCE[RDs0:7]：巨型帧（注意，报 IPHCE 时需要分辨到底是巨型帧还是由 IP 头错误）； OE[RDs0:11]：溢出错误； DE[RDs0:14]：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[RDs0:8]被置位时才有效。
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[RDs0:5]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	LC0	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 时表示接收到的帧为以太网类型封装的帧（RFC 894）。此位为 0 时表示接收到的帧为 IEEE802.3 类型封装的帧（RFC1042）。当帧长度小于 14 字节时，此位无效。 注：FT 具有的特殊含义参考表 31-18。
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 的整数倍，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 31-20 RDes0:7/5/0 的值和接收到的帧状态的关系

RDes0:5	RDes0:7	RDes0:0	帧状态
帧类型指示位 (FT)	IP 报头校验和错误标志位 (IPHCE)	负载校验和错误 (PCE)	
1	0	0	IP 类型帧，未检测到 IP 报头和负载校验和错误
1	0	1	IP 类型帧，负载校验和错误
1	1	0	IP 类型帧，IP 报头校验和错误
1	1	1	IP 类型帧，IP 报头校验和错误，负载校验和错误
0	0	0	IEEE802.3 类型封装帧（RFC1042）
0	0	1	IP 类型帧，未检测到 IP 报头校验和错误，因负载校验不受支持而未检测
0	1	1	不是 IP 类型帧（例如 ARP 帧）
0	1	0	保留

表 31-21 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。（长度要求为 32 的倍数）
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（R32_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。（长度要求为 32 的倍数）

表 31-22 RDes2 的各位定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RBAD1															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBAD1															

位	名称	描述
[31:0]	RBAD	RDes2 全部 32 位作为一个单元用来存放接收缓冲区的地址。（缓冲区起始地址需 32 字节对齐）

表 31-23 RDes3 的各位定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDAD2															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDAD2															

位	名称	描述
[31:0]	RDAD	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。（缓冲区起始地址需 32 字节对齐）

31.6.3 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 256 位的，因此设置收发描述符队列和缓冲区都需要保证它们的起始地址在 32 字节对齐的位置。

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，因为缓冲区的长度要求为 32 的倍数，所以一个缓冲区可以设置为 1536 字节。

31.6.4 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。用户需要初始化 DMA 后手动启动 DMA 才能继续运转。

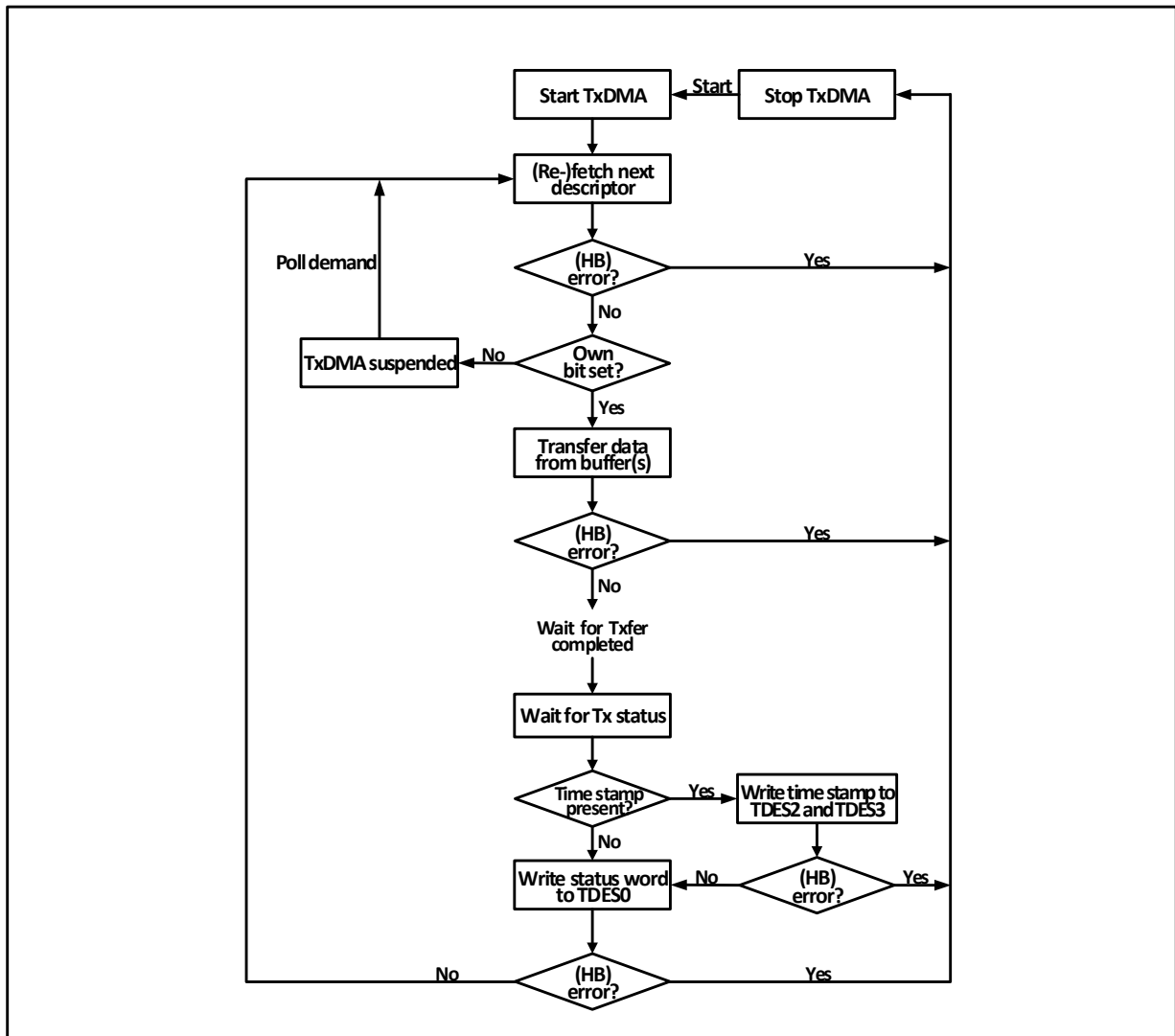
31.6.4.1 发送 DMA 配置

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 模式并使能增强描述符，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 TDes6 和 TDes7 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 31-9 发送流程



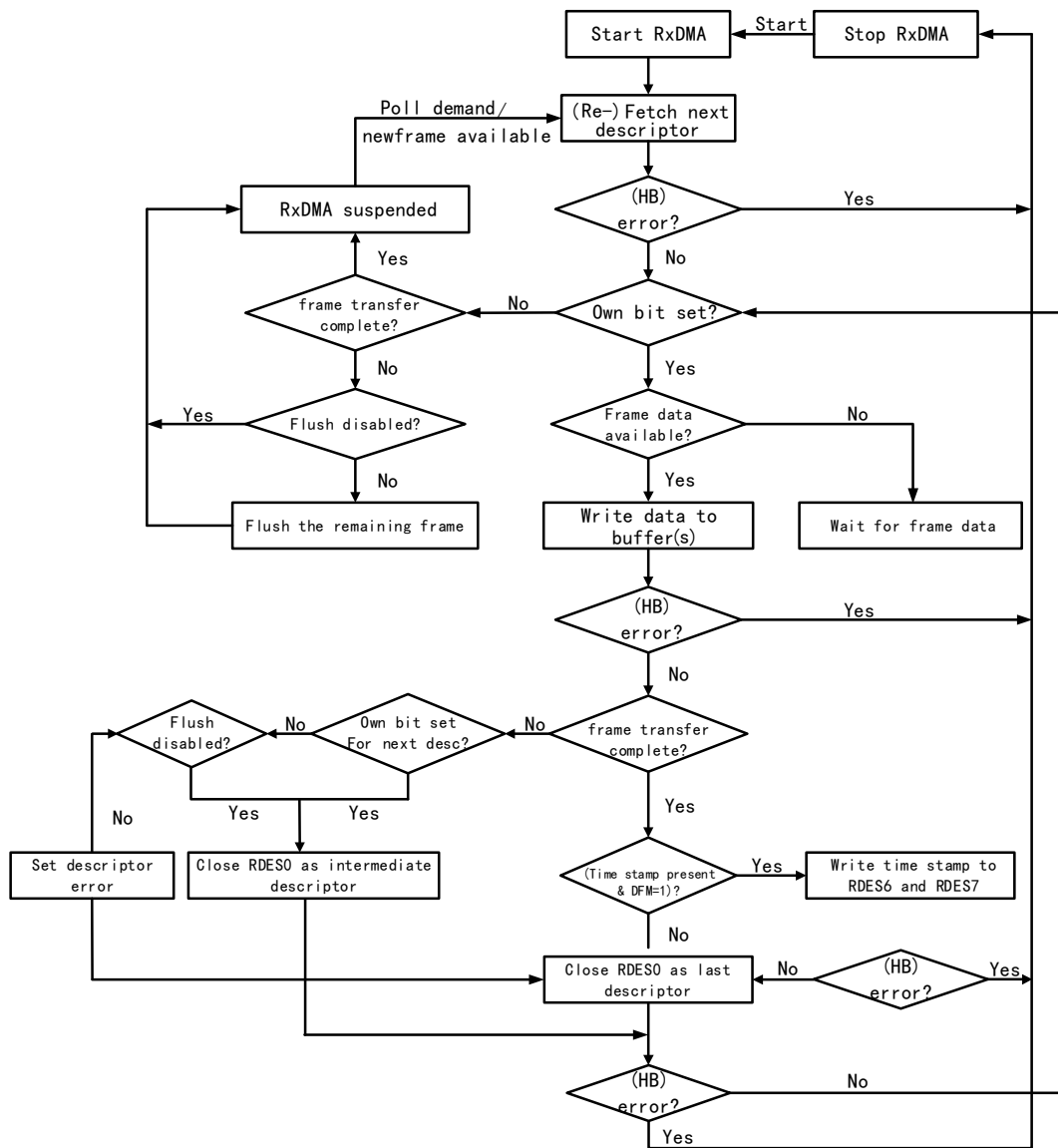
31.6.4.2 接收 DMA 配置

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 MAC 开启了 IEEE1588 PTP 模式并使能增强描述符，在 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。

下图展示了默认接收流转机制：

图 31-10 接收流程



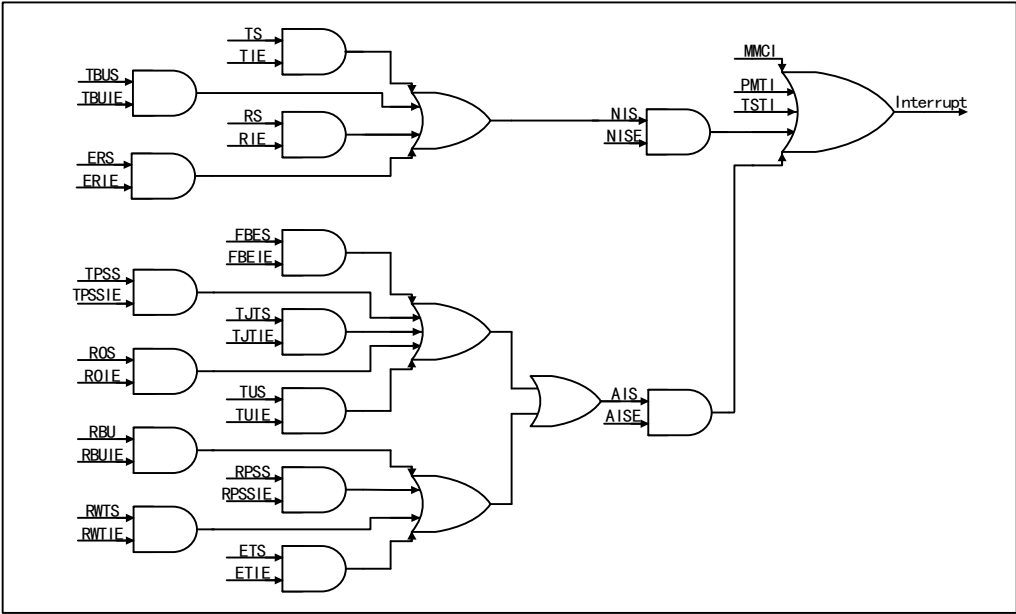
31.7 中断

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。

31.7.1 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 31-11 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。

31.7.2 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。

31.7.3 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。

31.8 寄存器描述

表 31-24 以太网相关寄存器列表

MAC 控制相关寄存器地址映射

名称	访问地址	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_MACRWUFR	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_MACA0HR	0x40028040	MAC 地址寄存器 0 高 32 位	0x8000FFFF
R32_ETH_MACA0LR	0x40028044	MAC 地址寄存器 0 低 32 位	0xFFFFFFFF
R32_ETH_MACA1HR	0x40028048	MAC 地址寄存器 1 高 32 位	0x0000FFFF
R32_ETH_MACA1LR	0x4002804C	MAC 地址寄存器 1 低 32 位	0xFFFFFFFF
R32_ETH_MACA2HR	0x40028050	MAC 地址寄存器 2 高 32 位	0x0000FFFF
R32_ETH_MACA2LR	0x40028054	MAC 地址寄存器 2 低 32 位	0xFFFFFFFF
R32_ETH_MACA3HR	0x40028058	MAC 地址寄存器 3 高 32 位	0x0000FFFF
R32_ETH_MACA3LR	0x4002805C	MAC 地址寄存器 3 低 32 位	0xFFFFFFFF
R32_ETH_PHY_CFGR	0x40028080	MAC_PHY 配置寄存器	0x40000001

MMC 控制相关寄存器地址映射，注意：地址非连续

名称	访问地址	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 一次冲突后发送好帧计数器	0x00000000
R32_ETH_MMCTGFMSCCR	0x40028150	MMC 多次冲突后发送好帧计数器	0x00000000
R32_ETH_MMCTGFCCR	0x40028168	MMC 发送好帧计数寄存器	0x00000000
R32_ETH_MMCRFCECR	0x40028194	MMC 接收 CRC 有误帧计数寄存器	0x00000000
R32_ETH_MMCRFAECR	0x40028198	MMC 接收对齐错误帧计数寄存器	0x00000000
R32_ETH_MMCRAFCCR	0x4002819C	MMC 接收所有帧计数寄存器	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	访问地址	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	访问地址	描述	复位值
R32_ETH_DMABMR	0x40029000	DMA 总线模式寄存器	0x00000001
R32_ETH_DMATPDR	0x40029004	DMA 发送查询寄存器	0x00000000
R32_ETH_DMARPDR	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/100M 物理层相关寄存器地址

名称	偏移地址	描述	复位值
R16_BMCR	0x00	基本控制寄存器	0x3100
R16_BMSR	0x01	基本状态寄存器	0x7849
R16_ANAR	0x04	自动协商消息寄存器	0x01E1
R16_ANALPAR	0x05	自动协商链接方能力寄存器	0x0001
R16_PAGE_SEL	0x1F	页选择寄存器	0x0000

注：内部物理层寄存器的偏移地址在 SMI 接口中使用。

PAGE0 相关寄存器地址

名称	偏移地址	描述	复位值
R16_PHY_STATUS	0x1A	PHY 状态寄存器	0x0020
R16_INTERRUPT_IND	0x1E	中断指示寄存器	0x0000

PAGE7 相关寄存器地址

名称	偏移地址	描述	复位值
R16_INTERRUPT_MASK	0x13	中断/LED 功能寄存器	0x0030

31.8.1 MAC 控制相关寄存器各位域

31.8.1.1 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[2:0]			Reserved					WD	JD	Reserved			IFG[2:0]		Reserved
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FES[1:0]		Reserved	LM	DM	IPCO	Reserved		APCS	Reserved			TE	RE	TCF	Reserved

位	名称	访问	描述	复位值
[31:29]	TCD[2:0]	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 TCF 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： $T_{delay} = TCD(ETH\_MACCR[31:29]) * 0.5ns$;	0
[28:24]	Reserved	RO	保留。	0

23	WD	RW	看门狗设置位： 1: MAC 关闭开门狗，能接收最大 16384 个字节的以太网帧； 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。	0
22	JD	RW	Jabber 设置位： 1: MAC 关闭 Jabber 定时器，最多能发送 16384 字节的以太网帧； 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。	0
[21:20]	Reserved	RO	保留。	0
[19:17]	IFG[2:0]	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。	0
16	Reserved	RO	保留。	0
[15:14]	FES[1:0]	RW	以太网速度设置域： 00: 10Mbit/s； 01: 100Mbit/s； 10: 1Gbit/s； 11: 保留，未使用。	0
13	Reserved	RO	保留。	0
12	LM	RW	自循环模式使能位。置该位使能自循环模式。	0
11	DM	RW	双工模式使能位。置该位使能全双工模式。	0
10	IPC0	RW	IPv4 校验检验使能位： 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； 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。	0
[9:8]	Reserved	RO	保留。	0
7	APCS	RW	填充&CRC 自动剥离使能位： 1: MAC 仅在长度/类型字段值小于或等于 1500 字节时，去除帧上的 Pad/FCS 字段。如果长度/类型字段大于或等于 1501 字节，不会更改帧内容； 0: MAC 不修改帧内容。	0
[6:4]	Reserved	RO	保留。	0
3	TE	RW	发送使能位： 1: 使能 MAC 发送器； 0: MAC 在发送完当前帧之后，关闭发送器，不再发送任何帧。	0

2	RE	RW	接收使能位： 1：使能 MAC 接收器； 0：MAC 在接收完当前帧之后，关闭接收器，不再接收任何帧。	0
1	TCF	RW	发送时钟翻转设置位： 1：将芯片内部的发送时钟反相作为芯片输出的 GTXC； 0：将芯片内部的发送时钟直接作为芯片输出的 GTXC。	0
0	Reserved	R0	保留。	0

31.8.1.2 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[1:0]	BFD	PAM	DAIF	HM	HU	PM	

位	名称	访问	描述	复位值
31	RA	RW	接收全部位： 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； 0：MAC 只把通过了过滤器的帧转发到接收队列中。	0
[30:11]	Reserved	R0	保留。	0
10	HPF	RW	HASH 过滤或者完美过滤选择位： 1：在 HM 或者 HU 置位的前提下，符合 HASH 过滤器或完美过滤器，都能通过地址过滤； 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。	0
9	SAF	RW	源地址过滤选择位： 1：MAC 会直接丢弃未通过源 MAC 地址过滤的帧； 0：MAC 对未通过源 MAC 地址过滤的帧进行标记。	0
8	SAIF	RW	源地址过滤结果颠倒位： 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； 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。	0
[7:6]	PCF[1:0]	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： 1：丢弃所有广播帧；	0

			0: 接收所有广播帧。	
4	PAM	RW	通过全部多播帧控制位: 1: 全部的多播帧都能通过地址过滤; 0: 多播帧能否通过过滤取决于 HM 的值。	0
3	DAIF	RW	目的 MAC 地址过滤器过滤结果颠倒控制位: 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; 0: 过滤器结果正常生效。	0
2	HM	RW	多播帧过滤模式选择位: 1: 进行 HASH 地址过滤; 0: 进行完美地址过滤。	0
1	HU	RW	单播帧过滤模式选择位: 1: 进行 HASH 地址过滤; 0: 进行完美地址过滤。	0
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。	0

31.8.1.3 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[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTH[15:0]															

偏移地址: 0x0C

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

位	名称	访问	描述	复位值
[31:0]	HTH[31:0]	RW	哈希列表高 32 位。	0
[31:0]	HTL[31:0]	RW	哈希列表低 32 位。	0

31.8.1.4 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[4:0]				MR[4:0]				Reserved				CR[2:0]		MW	MB

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:11]	PA[4:0]	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[4:0]	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	R0	保留。	0
[4:2]	CR[2:0]	RW	时钟范围设定域。 000: HCLK/42; 001: HCLK/62; 010: HCLK/16; 011: HCLK/26; 100: HCLK/102; 101: HCLK/74; 其他: 保留。	0
1	MW	RW	读写设定位: 1: 对物理层进行写操作; 0: 对物理层进行读操作。	0
0	MB	RW1Z	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。	0

31.8.1.5 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[15:0]															

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:0]	MD[15:0]	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

31.8.1.6 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[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												UPFD	RFCE	TFCE	FCB

位	名称	访问	描述	复位值
[31:16]	PT[15:0]	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
[15:4]	Reserved	RO	保留。	0
3	UPFD	RW	单播 Pause 帧检测位： 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； 0: MAC 只接收带协议规范定义的唯一地址的 Pause 帧。	0
2	RFCE	RW	接收流控使能位： 1: MAC 解析 Pause 帧并关闭发送器一段时间； 0: MAC 不解析 Pause 帧。	0
1	TFCE	RW	发送流控使能位： 1: MAC 使能发送流控，可以发送 Pause 帧； 0: MAC 关闭发送流控，不发送 Pause 帧。	0
0	FCB	RW1Z	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。	0

31.8.1.7 VLAN 标签寄存器 (R32_ETH_MACVLAN)

偏移地址：0x10

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

位	名称	访问	描述	复位值
[31:17]	Reserved	RO	保留。	0
16	VLANT	RW	标签比较控制位： 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: 将 VLAN 帧的第 15, 16 字节的全部 16 位数据都与 VLANTI 域进行比较。	0
[15:0]	VLANTI[15:0]	RW	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

31.8.1.8 唤醒帧过滤寄存器 (R32_ETH_MACRWUFR)

偏移地址：0x28

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

位	名称	访问	描述	复位值
[31: 0]	RWUFR[31: 0]	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 31.5.6.3 节的描述。	0

31.8.1.9 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	PHY_PMEBR	WFR	MPR	Reserved	PHY_PMEB_EN	WFE	MPE	PD

位	名称	访问	描述	复位值
31	WFFRP	RW1Z	唤醒帧过滤寄存器指针复位。置此位会将 MACRWUFR 寄存器全部清零。此位会在一个时钟周期自动清零。	0
[30:10]	Reserved	R0	保留。	0
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。	0
8	Reserved	R0	保留。	0
7	PHY_PMEBR	RZ	当收到 PHY 的唤醒帧，此位会被置位。读取自动清零。	0
6	WFR	RZ	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。	0
5	MPR	RZ	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。	0
4	Reserved	R0	保留。	0
3	PHY_PMEB_EN	RW	PHY_PMEB 使能位。置此位允许 PHY 收到唤醒帧时产生 PMT 事件	0
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。	0
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。	0

0	PD	RW1Z	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。	0

31.8.1.10 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	R0	保留。	0
9	TSTS	RZ	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。	0
[8:7]	Reserved	R0	保留。	0
6	MMCTS	R0	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。	0
5	MMCRS	R0	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。	0
4	MMCS	R0	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。	0
3	PMTS	R0	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。	0
[2:0]	Reserved	R0	保留。	0

31.8.1.11 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	R0	保留。	0
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	R0	保留。	0

3	PMTIM	RW	PMT 中断屏蔽位。置此位将屏蔽 PMT 中断。	0
[2:0]	Reserved	R0	保留。	0

31.8.1.12 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
MACA0H[15:0]															

位	名称	访问	描述	复位值
31	MO	R0	总是为 1。	1
[30:16]	Reserved	R0	保留。	0
[15:0]	MACA0H[15:0]	RW	MAC 地址的高 16 位, 即 47:32 位。	0xFFFF

31.8.1.13 MAC 地址寄存器 0 低 32 位 (R32_ETH_MACA0LR)

偏移地址: 0x44

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACA0L[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA0L[15:0]															

位	名称	访问	描述	复位值
[31:0]	MACA0L[31:0]	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。	0xFFFF FFFF

31.8.1.14 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[5:0]						Reserved							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA1H[15:0]															

位	名称	访问	描述	复位值
31	AE	RW	地址过滤使能位: 1: 地址过滤器使用 MAC 地址 1 来进行完美过滤; 0: 地址过滤器忽略 MAC 地址 1。	0
30	SA	RW	地址角色选择位: 1: MAC 地址 1 被用来和接收到的帧的源地址对比; 0: MAC 地址 1 被用来和接收到的帧的目标地址对比	0

			比。	
[29:24]	MBC[5:0]	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]。	0
[23:16]	Reserved	RO	保留。	0
[15:0]	MACA1H[15:0]	RW	MAC 地址的高 16 位，即 47:32 位。	0xFFFF

31.8.1.15 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[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA1L[15:0]															

位	名称	访问	描述	复位值
[31:0]	MACA1L[31:0]	RW	MAC 地址的低 32 位，即 31:0 位。	0xFFFF FFFF

31.8.1.16 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[5:0]						Reserved							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA2H[15:0]															

位	名称	访问	描述	复位值
31	AE	RW	地址过滤使能位： 1：地址过滤器使用 MAC 地址 2 来进行完美过滤； 0：地址过滤器忽略 MAC 地址 2。	0
30	SA	RW	地址角色选择位： 1：MAC 地址 2 被用来和接收到的帧的源地址对比； 0：MAC 地址 2 被用来和接收到的帧的目标地址对比。	0
[29:24]	MBC[5:0]	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。	0

			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]。	
[23:16]	Reserved	R0	保留。	0
[15:0]	MACA2H[15:0]	RW	MAC 地址的高 16 位, 即 47:32 位。	0xFFFF

31.8.1.17 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[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA2L[15:0]															

位	名称	访问	描述	复位值
[31:0]	MACA2L[31:0]	RW	MAC 地址的低 32 位, 即 31:0 位。	0xFFFF FFFF

31.8.1.18 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[5:0]						Reserved							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA3H[15:0]															

位	名称	访问	描述	复位值
31	AE	RW	地址过滤使能位: 1: 地址过滤器使用 MAC 地址 3 来进行完美过滤; 0: 地址过滤器忽略 MAC 地址 3。	0
30	SA	RW	地址角色选择位: 1: MAC 地址 3 被用来和接收到的帧的源地址对比; 0: MAC 地址 3 被用来和接收到的帧的目标地址对比。	0
[29:24]	MBC[5:0]	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];	0

			R32_ETH_MACA3HR[25]置位则屏蔽 MACA3L[15:8]; R32_ETH_MACA3HR[24]置位则屏蔽 MACA3L[7:0]。	
[23:16]	Reserved	RO	保留。	0
[15:0]	MACA3H[15:0]	RW	MAC 地址的高 16 位, 即 47:32 位。	0xFFFF

31.8.1.19 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[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA3L[15:0]															

位	名称	访问	描述	复位值
[31:0]	MACA3L[31:0]	RW	MAC 地址的低 32 位, 即 31:0 位。	0xFFFF FFFF

31.8.1.20 ETH_PHY 配置寄存器 (R32_ETH_PHY_CFGR)

偏移地址: 0x80

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

位	名称	访问	描述	复位值
31	PHY_RSTN	RW	以太网 PHY 全局复位信号: 1: 正常工作模式; 0: 复位以太网 PHY。	0
30	PHY_PD	RW	掉电模式下 MAC 可以通过 MDIO 接口访问 PHY 的寄存器: 1: 掉电模式; 0: 正常工作模式。	1
[29:11]	Reserved	RO	保留。	0
10	DUPIEX	RO	1: 连接为全双工; 0: 连接为半双工。	0
9	SPEED	RO	1: 连接为百兆; 0: 连接为十兆。	0
8	LINK_STATUS	RO	1: PHY 连接成功; 0: 未连接。	0
7	PHYADDR_EN	RW	重新配置 PHY 地址: 1: bit4-0 作为 PHY 的地址。 0: PHY 地址默认为 1。	0
[6:5]	Reserved	RO	保留。	0

[4:0]	REPHYADDR[4:0]	RW	当 bit7 为 1 时, bit[4:0] 作为 PHY 的地址。	0x1

31.8.2 MMC 相关寄存器各位域

31.8.2.1 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	R0	保留。	0
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。	0
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。	0
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会自动归零。	0
0	CR	RW1Z	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。	0

31.8.2.2 MMC 接收中断寄存器 (R32_ETH_MMCRIR)

偏移地址: 0x0104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														RGUFS	Reserved
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	R0	保留。	0
17	RGUFS	RZ	接收好帧数量过半时会置此位。	0
[16:6]	Reserved	R0	保留。	0
5	RFCES	RZ	接收 CRC 校验错误的帧数量过半时会置此位。	0
[4:0]	Reserved	R0	保留。	0

31.8.2.3 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	R0	保留。	0
21	TGFS	RZ	发送帧数过半时此位会置位。	0
[20:0]	Reserved	R0	保留。	0

31.8.2.4 MMC 接收中断屏蔽寄存器 (R32_ETH_MMCRIMR)

偏移地址: 0x010C

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

位	名称	访问	描述	复位值
[31:18]	Reserved	R0	保留。	0
17	RGUFM	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。	0
[16:6]	Reserved	R0	保留。	0
5	RFCM	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。	0
[4:0]	Reserved	R0	保留。	0

31.8.2.5 MMC 发送中断屏蔽寄存器 (R32_ETH_MMCTIMR)

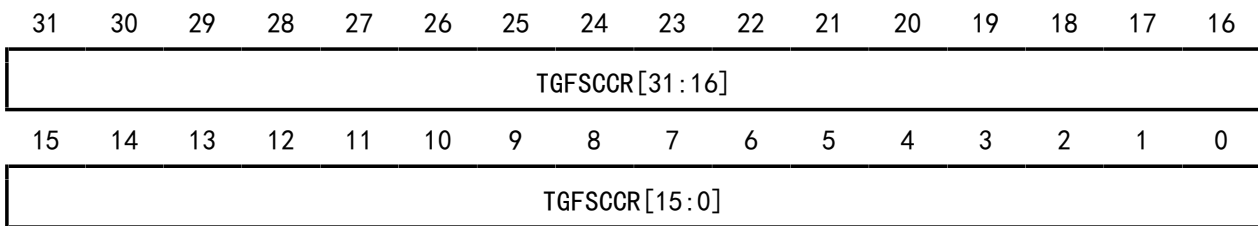
偏移地址: 0x0110

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	R0	保留。	0
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。	0
[20:0]	Reserved	R0	保留。	0

31.8.2.6 MMC 一次冲突后发送好帧计数器 (R32_ETH_MMCTGFSCCR)

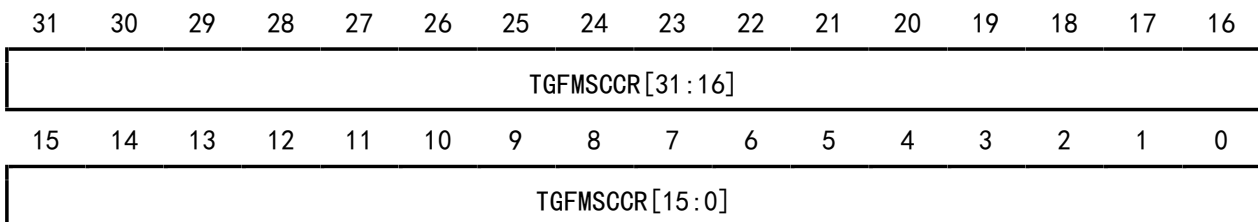
偏移地址: 0x014C



位	名称	访问	描述	复位值
[31:0]	TGFSCCR[31:0]	RO	这个域包含在半双工模式下, 单个冲突后成功发送的帧的数量, 即单个冲突后发送好帧的计数器。	0

31.8.2.7 MMC 多次冲突后发送好帧计数器 (R32_ETH_MMCTGFMSCCR)

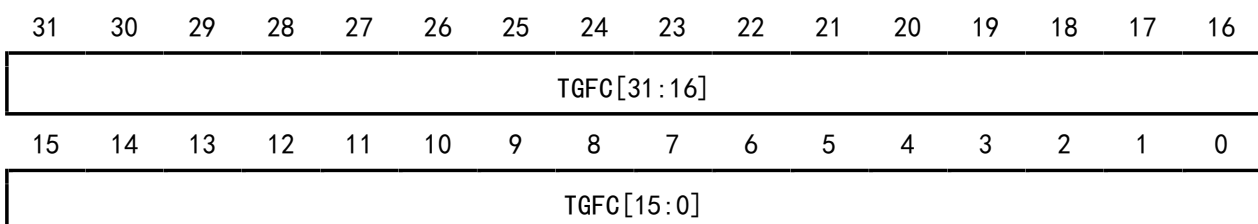
偏移地址: 0x0150



位	名称	访问	描述	复位值
[31:0]	TGFMSCCR[31:0]	RO	这个域包含在半双工模式下, 一个以上冲突后成功发送的帧的数量, 即多个冲突后发送好帧的计数器。	0

31.8.2.8 MMC 发送好帧计数寄存器 (R32_ETH_MMCTGFCR)

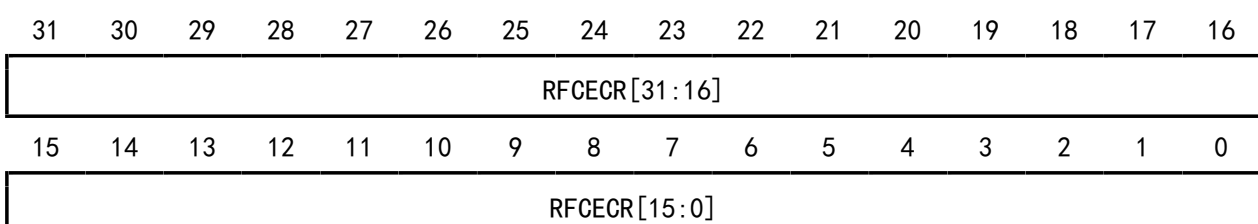
偏移地址: 0x0168



位	名称	访问	描述	复位值
[31:0]	TGFC[31:0]	RO	MAC 发送出去的正确帧的计数数量。	0

31.8.2.9 MMC 接收 CRC 有误帧计数寄存器 (R32_ETH_MMCRFCECR)

偏移地址: 0x0194



位	名称	访问	描述	复位值
[31:0]	RFECR[31:0]	R0	MAC 接收到的存在 CRC 校验错误的帧的计数器。	0

31.8.2.10 MMC 接收对齐错误计数寄存器 (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[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFAECR[15:0]															

位	名称	访问	描述	复位值
[31:0]	RFAECR[31:0]	R0	MAC 接收到的存在对齐错误的帧的计数器。	0

31.8.2.11 MMC 接收所有帧计数寄存器 (R32_ETH_MMCRAFCR)

偏移地址: 0x019C

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

位	名称	访问	描述	复位值
[31:0]	RAFCR[31:0]	R0	MAC 接收到的所有帧的数量, 不论是否有错误。	0

31.8.2.12 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[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RGUFCR[15:0]															

位	名称	访问	描述	复位值
[31:0]	RGUFCR[31:0]	R0	MAC 接收到的正常的帧的数量。	0

31.8.3 IEEE 1588 (PTP) 相关寄存器各位域

31.8.3.1 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	R0	保留。	0
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。	0
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。	0
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。	0
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。	0
1	TSFCU	RW	更新模式选择控制位： 1：表示使用精调模式； 0：表示使用粗调模式。	0
0	TSE	RW	附加时间戳使能位： 1：发送或接收完成后，向描述符中添加时间戳； 0：不向描述符中添加时间戳。	0

31.8.3.2 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[7:0]							

位	名称	访问	描述	复位值
[31:8]	Reserved	R0	保留。	0
[7:0]	STSSI[7:0]	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

31.8.3.3 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[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STS[15:0]															

位	名称	访问	描述	复位值
[31:0]	TSHR[31:0]	R0	PTP 系统时间值，实时值，单位为秒。	0

31.8.3.4 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[30:16]													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STSS[15:0]															

位	名称	访问	描述	复位值
31	STPNS	R0	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[30:0]	R0	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

31.8.3.5 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[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSUS[15:0]															

位	名称	访问	描述	复位值
[31:0]	TSUS[31:0]	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

31.8.3.6 PTP 时间戳更新寄存器低 32 位（R32_ETH_PTPTSLUR）

偏移地址：0x0714

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

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	0

			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。	
[30:0]	TSUSS[30:0]	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

31.8.3.7 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[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSA[15:0]															

位	名称	访问	描述	复位值
[31:0]	TSA[31:0]	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

31.8.3.8 PTP 目标时间寄存器高 32 位 (R32_ETH_PTPTHR)

偏移地址：0x071C

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

位	名称	访问	描述	复位值
[31:0]	TTSH[31:0]	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

31.8.3.9 PTP 目标时间寄存器低 32 位 (R32_ETH_PTPTLR)

偏移地址：0x0720

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

位	名称	访问	描述	复位值
[31:0]	TTSL[31:0]	RW	目标时间亚秒值。如果 PTP 系统时间达到或超过这	0

			个值，且使能了相关中断，那么会产生一个中断。	

31.8.4 DMA 控制相关寄存器各位域

31.8.4.1 DMA 总线模式寄存器 (R32_ETH_DMABMR)

偏移地址：0x1000

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

位	名称	访问	描述	复位值
31	ERX_RST	RW	写 1 复位 MAC 接收模块，写 0 撤销复位。	0
30	ETX_RST	RW	写 1 复位 MAC 发送模块，写 0 撤销复位。	0
29	DMARX_RST	RW	写 1 复位 DMA 接收模块，写 0 撤销复位。	0
28	DMATX_RST	RW	写 1 复位 DMA 发送模块，写 0 撤销复位。	0
[27:8]	Reserved	RO	保留。	0
7	DFM	RW	描述符模式： 1：增强描述符模式； 0：常规描述符模式。	0
[6:2]	DSL[4:0]	RW	描述符跳跃长度： 这些位定义了 2 个不以链式结构连接的描述符之间的跳跃距离，单位为字（32 位）。地址跳跃是指从当前描述符的结尾到下一个描述符开头的地址差值。当 DSL 域为 0 时，在环形结构下，DMA 认为描述符是相邻地连续排列的。	0
1	Reserved	RO	保留。	0
0	SR	RW	软件复位： 置 1 时，MAC 的 DMA 控制器复位 MAC 所有子系统的内部寄存器和逻辑电路。在 MAC 内部不同时钟域模块完成复位操作后，自动清除该位。在重新写 MAC 的寄存器前，应当确保该位为 0。	1

31.8.4.2 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[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TPDR[15:0]

位	名称	访问	描述	复位值
[31:0]	TPDR[31:0]	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

31.8.4.3 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[31:16]

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

RPDR[15:0]

位	名称	访问	描述	复位值
[31:0]	RPDR[31:0]	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

31.8.4.4 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[31:16]

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

RDLAR[15:0]

位	名称	访问	描述	复位值
[31:0]	RDLAR[31:16]	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

31.8.4.5 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[31:16]

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

TDLAR[15:0]

位	名称	访问	描述	复位值

[31:0]	TDLAR[31:0]	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

31.8.4.6 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
Reserved	TSTS	PMTS	MMCS	Reserved	EBS[2:0]	TPS[2:0]	RPS[2:0]	NIS							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIS	ERS	Reserved	PLS	ETS	RWTS	RPSS	RBUS	RS	TUS	ROS	TJTS	TBUS	TPSS	TS	

位	名称	访问	描述	复位值
[31:30]	Reserved	R0	保留。	0
29	TSTS	R0	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。	0
28	PMTS	R0	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。	0
27	MMCS	R0	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。	0
26	Reserved	R0	保留。	0
[25:23]	EBS[2:0]	R0	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; 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[2:0]	R0	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[2:0]	R0	接收流程状态域。这个域用来表示当前接收 DMA 的	0

			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。	
16	NIS	RW1Z	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; -DMASR[11]: 内部物理层连接状态改变。 NIS 位为黏着位, 当导致 NIS 位置 1 的对应中断状态位被清零时, 必须通过写 1 的方式将 NIS 位也清零。	0
15	AIS	RW1Z	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 (Jabber) 超时; -DMASR[4]: 接收 FIFO 溢出; -DMASR[5]: 发送数据下溢; -DMASR[7]: 接收缓存不可用; -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。 AIS 位为黏着位, 当导致 AIS 位置 1 的对应中断状态位被清零时, 必须通过写 1 的方式将 AIS 位也清零。	0
14	ERS	RW1Z	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。	0
[13:12]	Reserved	R0	保留。	0
11	PLS	RW1Z	内部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。	0
10	ETS	RW1Z	早发送状态位。置位时表示发送帧已经全部压入 FIFO。	0
9	RWTS	RW1Z	看门狗超时标志位。置位时表示帧长已经超过了 2048 字节。	0
8	RPSS	RW1Z	接收流程停止状态位。该位置位表示接收流程已经停止。	0
7	RBUS	RW1Z	接收缓存不可用状态位。该位置位表示接收描述符	0

			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。	
6	RS	RW1Z	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。	0
5	TUS	RW1Z	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。	0
4	ROS	RW1Z	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。	0
3	TJTS	RW1Z	发送啰嗦（Jabber）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（Jabber）超时位已经置位。	0
2	TBUS	RW1Z	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。	0
1	TPSS	RW1Z	发送流程停止状态位。该位被置位表示发送流程已经停止。	0
0	TS	RW1Z	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。	0

31.8.4.7 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	Reserved

位	名称	访问	描述	复位值
[31:27]	Reserved	R0	保留。	0
26	DTCEFD	RW	不丢弃 TCP/IP 校验和错误帧控制位： 1：发现 TCP/IP/ICMP/UDP 之类协议的检验和存在错误时，不丢弃该帧； 0：如果 FEF 位为 0，MAC 会丢弃所有有错误的帧。	0
[25:22]	Reserved	R0	保留。	0
21	TSF	RW	发送存储转发控制位： 1：发送流程将完整的帧全部写入 FIFO 后才会启动发送； 0：发送流程写入 FIFO 的数据达到确定值之后就会启动发送。	0
20	FTF	RW	发送 FIFO 清空控制位。置此位将复位发送 FIFO。	0

[19:14]	Reserved	R0	保留。	0
13	ST	RW	开始或停止发送控制位： 1：把发送进程置为运行状态； 0：发送完当前帧之后，发送进程进入停止状态。	0
[12:8]	Reserved	R0	保留。	0
7	FEF	RW	转发错误帧控制位： 1：除了过短的帧之外，所有的帧都会转发给 DMA； 0：接收 FIFO 会丢弃有错误的帧。	0
6	FUGF	RW	转发过短的帧控制位： 1：接收 FIFO 转发长度过短的帧； 0：接收 FIFO 丢弃所有长度小于 64 字节的帧。	0
[5:2]	Reserved	R0	保留。	0
1	SR	RW	开始或停止接收控制位： 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； 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。	0
0	Reserved	R0	保留。	0

31.8.4.8 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
Reserved															NISE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AISE	ERS	Reserved	PLE	ETIE	RWTIE	RPSIE	RBUIE	RIE	TUIE	ROIIE	TJTIE	TBUIE	TPSIE	TIE	

位	名称	访问	描述	复位值
[31:17]	Reserved	R0	保留。	0
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。	0
15	AISE	RW	异常中断。使能该位将使能以下中断。 -DMASR[1]：发送流程停止； -DMASR[3]：发送啰嗦（Jabber）超时； -DMASR[4]：接收 FIFO 溢出； -DMASR[5]：发送数据下溢； -DMASR[7]：接收缓存不可用； -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。	0
14	ERS	RW	早接收中断使能位。使能该位即可以产生早接收中	0

			断。NISE 须置位。AISE 须置位。	
[13:12]	Reserved	RO	保留。	0
11	PLE	RW	内部 100M 物理层连接状态改变中断使能位。	0
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。	0
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。	0
8	RPSIE	RW	接收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。	0
7	RBUIE	RW	接收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。	0
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。	0
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。	0
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。	0
3	TJTIE	RW	发送啰嗦（Jabber）超时中断使能。使能该位即可以产生发送啰嗦（Jabber）超时中断。AISE 须置位。	0
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	0
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。	0
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。	0

31.8.4.9 DMA 丢失帧寄存器（R32_ETH_DMAMFBCR）

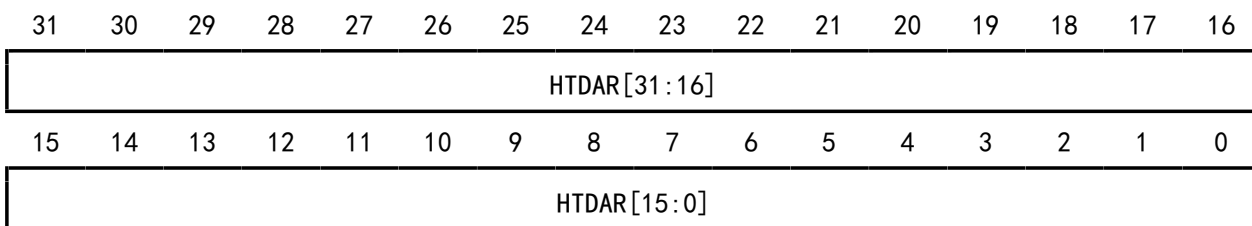
偏移地址：0x1020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			OF0C	MFA[10:0]											OMFC
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFC[15:0]															

位	名称	访问	描述	复位值
[31:29]	Reserved	RO	保留。	0
28	OF0C	RZ	FIFO 上溢计数器的上溢位。	0
[27:17]	MFA[10:0]	RZ	应用程序所丢失的帧。	0
16	OMFC	RZ	丢失帧计数器溢出位。	0
[15:0]	MFC[15:0]	RZ	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

31.8.4.10 DMA 当前发送描述符寄存器（R32_ETH_DMACHTDR）

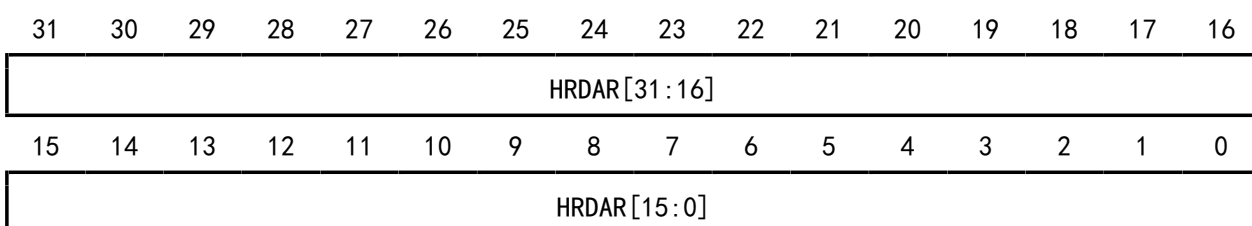
偏移地址：0x1048



位	名称	访问	描述	复位值
[31:0]	HTDAR[31:0]	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

31.8.4.11 DMA 当前接收描述符寄存器 (R32_ETH_DMACHRDR)

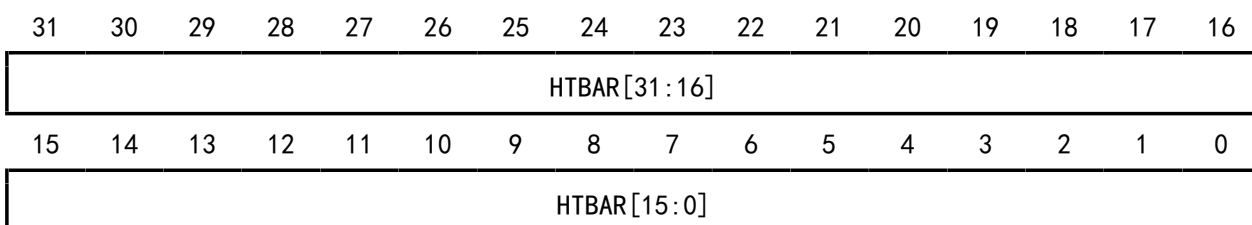
偏移地址: 0x104C



位	名称	访问	描述	复位值
[31:0]	HRDAR[31:0]	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

31.8.4.12 DMA 当前发送缓冲区寄存器 (R32_ETH_DMACHTBR)

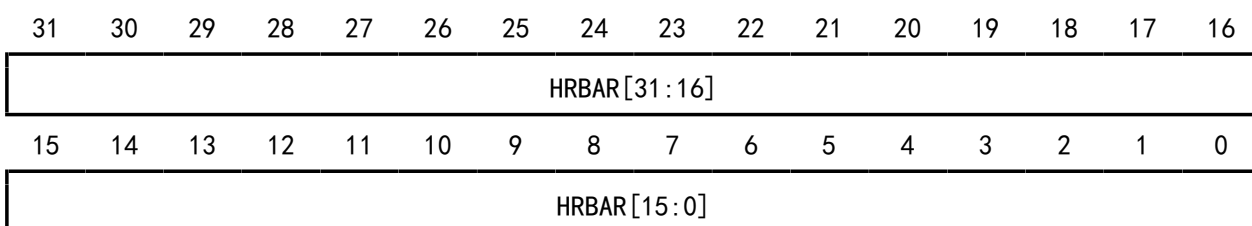
偏移地址: 0x1050



位	名称	访问	描述	复位值
[31:0]	HTBAR[31:0]	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

31.8.4.13 DMA 当前接收缓冲区寄存器 (R32_ETH_DMACHRBR)

偏移地址: 0x1054



位	名称	访问	描述	复位值
[31:0]	HRBAR[31:0]	R0	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

31.8.5 内部 10M/100M 物理层相关寄存器地址

31.8.5.1 基础控制寄存器 (R16_BMCR)

偏移地址: 0x00

位	名称	访问	描述	复位值
15	RST	RW/SC	1: PHY 复位; 0: 正常操作。	0
14	LOOPBACK	RW	1: 使能回环; 0: 正常操作。	0
13	SPEED_SELECTION	RW	1: 100Mb/s; 0: 10Mb/s。 自动协商后显示协商工作速度状态。	1
12	AUTO_NEGOTIATION	RW	1: 使能自动协商; 0: 禁止自动协商。	1
11	POWER_DOWN	RW	1: 关闭电源; 0: 正常运行。	0
10	ISOLATE	RW	1: MII/RMII 接口与 PHY 隔离, PHY 依旧能够响应 MDC/MDIO; 0: 正常工作。	0
9	RESTART_AUTO_NEGOTIATION	RW/SC	1: 重新自动协商; 0: 正常操作。	0
8	DUPLEX_MODE	RW	1: 全双工; 0: 半双工。	1
7	COLLISION_TEST	RW	碰撞测试。 1: 冲突检测使能; 0: 正常操作。	0
[6:0]	Reserved	R0	保留。	0

31.8.5.2 基础状态寄存器 (R16_BMSR)

地址偏移: 0x01

位	名称	访问	描述	复位值
15	100BASE_T4	R0	1: 支持启用100BASE-T4; 0: 不支持 100BASE-T4。	0
14	100BASE_TX_FULL_DUPLEX	R0	1: 支持启用100BASE-TX全双工; 0: 不支持 100BASE-TX 全双工。	1
13	100BASE_TX_HALF_DUPLEX	R0	1: 支持启用100BASE-TX半双工; 0: 不支持 100BASE-TX 半双工。	1
12	10BASE_T_FULL_DUPLEX	R0	1: 支持启用10BASE-T全双工; 0: 不支持 10BASE-T 全双工。	1
11	10BASE_T_HALF_DUPLEX	R0	1: 支持启用10BASE-T半双工; 0: 不支持 10BASE-T 半双工。	1
[10:7]	Reserved	R0	保留。	0

6	MF_PREAMBLE_SUPPRESSION	R0	允许接收前导码抑制的管理帧。	1
5	AUTO_NEGOTIATION_COMPLETE	R0	1: 自动协商完成; 0: 自动协商未完成。	0
4	REMOTE_FAULT	R0	1: 检测到远程错误情况 (读取清除); 0: 未检测到远程错误情况。	0
3	AUTO_NEGOTIATION_ABILITY	R0	1: PHY能执行自协商; 0: PHY 不能执行自协商。	1
2	LINK_STATUS	R0	1: 已建立有效链接; 0: 未建立有效链接。	0
1	JABBER_DETECT	R0	1: 检测到Jabber条件; 0: 未检测到 Jabber 条件。	0
0	EXTENDED_CAPABILITY	R0	1: 扩展寄存器功能; 0: 无扩展寄存器功能。	1

31.8.5.3 自动协商消息寄存器 (R16_ANAR)

偏移地址: 0x04

位	名称	访问	描述	复位值
15	NP	RW	下一页位: 1: 发送协议规则数据页; 0: 发送首要能力数据页。	0
14	ACK	R0	1: 确认接收到链路伙伴能力数据字; 0: 未接收到确认信号。	0
13	RF	RW	1: 通知远程错误检测能力; 0: 不通知远程错误检测能力。	0
12	Reserved	R0	保留。	0
11	ASYPAUSE	RW	1: 支持非对称流控; 0: 不支持非对称流控。	0
10	PAUSE	RW	1: 支持流控; 0: 不支持流控。	0
9	100BASE_T4	R0	1: 本地节点支持 100Base-T4; 0: 本地节点不支持 100Base-T4。	0
8	100BASE_TX_FD	RW	1: 本地节点支持 100Base-TX 全双工; 0: 本地节点不支持 100Base-TX 全双工。	1
7	100BASE_TX	RW	1: 本地节点支持 100Base-TX; 0: 本地节点不支持 100Base-TX。	1
6	10BASE_T_FD	RW	1: 本地节点支持 10Base-TX 全双工; 0: 本地节点不支持 10Base-TX 全双工。	1
5	10BASE_T	RW	1: 本地节点支持 10Base-T 半双工; 0: 本地节点不支持 10Base-T 半双工。	1

[4:0]	SELECT[4:0]	R0	此节点支持二进制编码选择器目前只有 CSMA/CD 00001，没有其他协议支持。	1

31.8.5.4 自动协商链接方能力寄存器 (R16_ANLPAR)

偏移地址：0x05

位	名称	访问	描述	复位值
15	NP	RW	下一页位。 1：发送协议细节数据页； 0：发送基本能力数据页。	0
14	ACK	R0	1：确认接收到对端自动协商功能寄存器； 0：未接收到确认信号。	0
13	RF	RW	1：通知远程错误检测功能； 0：不通知远程错误检测功能。	0
12	Reserved	R0	保留。	0
11	ASYPAUSE	R0	1：支持通知非对称暂停； 0：不支持非对称暂停。 启用自动协商时，此位反映链接方的能力。	0
10	PAUSE	R0	1：支持通知对称暂停； 0：不支持对称暂停。	0
9	100BASE_T4	R0	1：PHY 支持 100BASE-T4； 0：PHY 不支持 100BASE-T4。	0
8	100BASE_TX_FD	R0	1：PHY 支持 100BASE-TX 全双工； 0：PHY 不支持 100BASE-TX 全双工。	0
7	100BASE_TX	R0	1：PHY 支持 100BASE-TX； 0：PHY 不支持 100BASE-TX。	0
6	10BASE_T_FD	R0	1：PHY 支持 10BASE-T 全双工； 0：PHY 不支持 10BASE-T 全双工。	0
5	10BASE_T	R0	1：PHY 支持 10BASE-T； 0：PHY 不支持 10BASE-T。	0
[4:0]	SELECT[4:0]	R0	PHY支持二进制编码选择器。 目前只有CSMA/CD 00001特殊，没有其他协议支持。	0x01

31.8.5.5 页选择寄存器 (R16_PAGE_SEL)

偏移地址：0x1F

位	名称	访问	描述	复位值
[15:8]	Reserved	R0	保留。	0
[7:0]	SEL_PAGE_ADDR[7:0]	RW	选择页地址：00000000：11111111。	0

31.8.6 PAGE0 相关寄存器

31.8.6.1 PHY 状态寄存器 (R16_PHY_STATUS)

偏移地址: 0x1A

位	名称	访问	描述	复位值
15	INCOMPATIBLE_LINK	R0	自动协商优先级解析后本地设备与对端设备能力兼容性状态: 1: 不兼容; 0: 兼容。	0
[14:7]	Reserved	R0	保留。	0
6	SIGNAL_STATUS	R0	100M 信号状态: 1: 信号状态良好; 0: 无信号。	0
5	MDI_MDIX_STATUS	R0	MDI/MDIX 状态: 1: MDI; 0: MDIX。	1
4	POLARITY_STATUS	R0	极性状态: 1: 极性反转; 0: 极性正常。	0
[3:0]	PHY_SPEED_DUP[3:0]	R0	PHY 运行速率和全半双工工作模式。	0

PHY_SPEED_DUP 详情表				
3	PHY100_INDICATE	R0	运行速率 100M。	0
2	PHY10_INDICATE	R0	运行速率 10M。	0
1	FULL_INDICATE	R0	全双工。	0
0	HALFFULL_INDICATE	R0	半双工。	0

31.8.6.2 中断指示寄存器 (R16_INTERRUPT_IND)

偏移地址: 0x1E

位	名称	访问	描述	复位值
[15:12]	Reserved	R0	保留。	0
11	LINKSTATUS_CHG	RZ	链路状态改变提示。	0
[10:0]	Reserved	R0	保留。	0

31.8.7 PAGE7 相关寄存器

31.8.7.1 中断/LED 标志寄存器 (R16_INTERRUPT_MASK)

PAGE7 偏移地址: 0x13

位	名称	访问	描述	复位值
[15:14]	Reserved	R0	保留。	0
13	INT_LINKCHG	RW	链路改变中断 MASK。	0

[12:10]	Reserved	R0	保留。	0
9	LED_EN	RW	LED 功能使能。	0
[8:0]	Reserved	R0	保留。	0x30

第 32 章 SDIO 接口 (SDIO)

本章模块描述仅适用于 CH32H417、CH32H415 微控制器产品。

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 HB 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 卡，或控制其他使用 SDIO 作为通讯接口的设备，比如 WiFi/4G 模块。

32.1 主要特征

32.1.1 特征

- 支持 SD 卡、SDIO 卡和 MMC 卡
- 支持 1 位、4 位和 8 位总线
- 最高通讯时钟可达 100MHz
- 兼容 MMC 规范 4.5（向前兼容）
- 兼容 SD 卡规范 2.0，SDIO 卡规范 2.0
- 不兼容 SPI 或 QSPI

32.1.2 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。

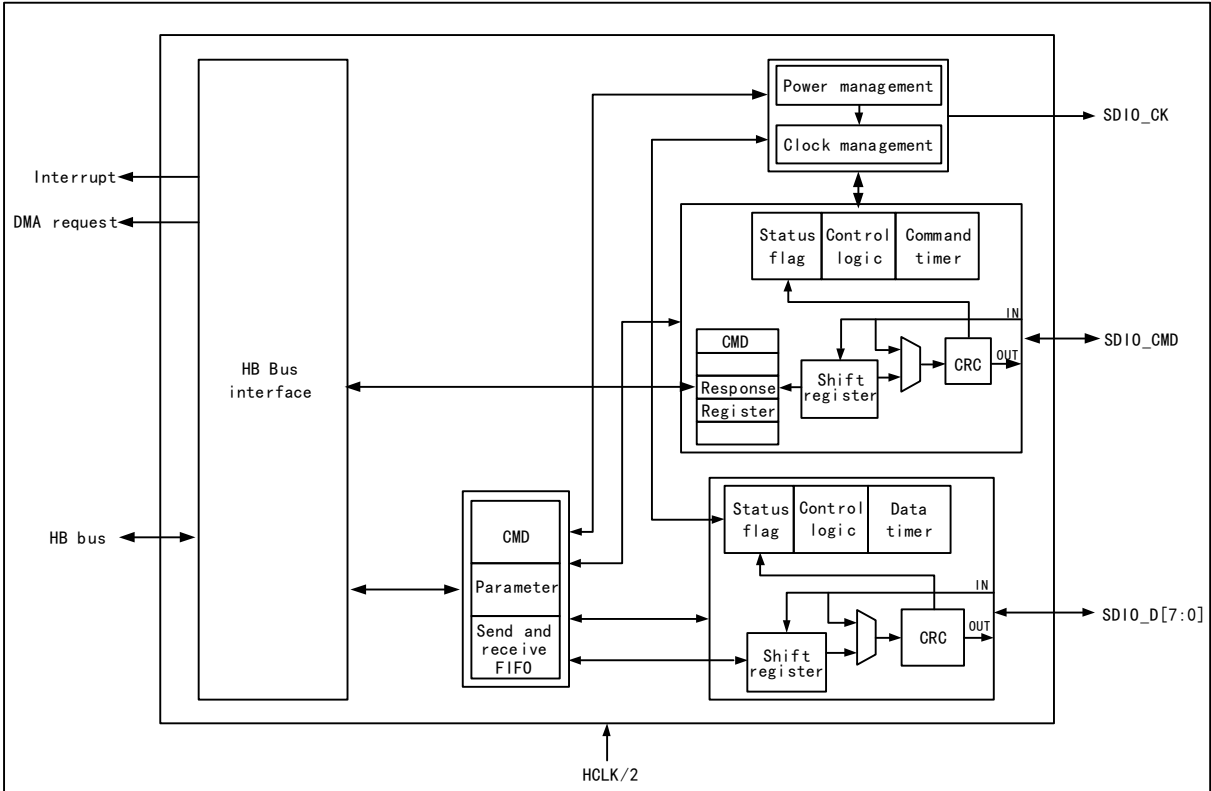
32.2 接口和时钟

SDIO 通过 HB 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。

32.2.1 外设结构

SDIO 的结构如图 32-1 所示。

图 32-1 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 HB 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。

32.2.2 引脚及其配置

SDIO 需要配置的引脚及其模式见表 32-1。

表 32-1 SDIO 的引脚配置

SDIO 复用功能	需要配置成的引脚模式
SDIO_D[0:3]	推挽复用输出
SDIO_CK	推挽复用输出
SDIO_CMD	推挽复用输出
SDIO_D4	推挽复用输出
SDIO_D5	推挽复用输出
SDIO_D6	推挽复用输出
SDIO_D7	推挽复用输出

注：从机模式时，SDIO_CK 引脚应配置为浮空输入。

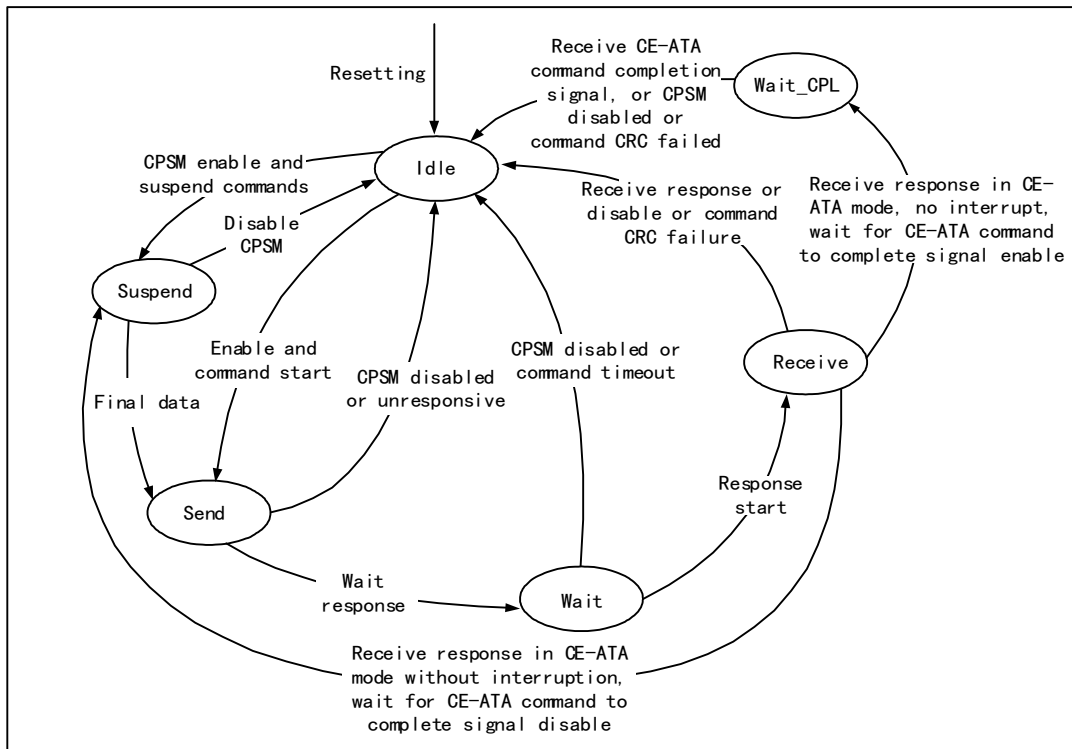
32.2.3 时钟

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。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。

32.2.4 命令状态机

SDIO 的命令工作流程遵循如下图的状态机。

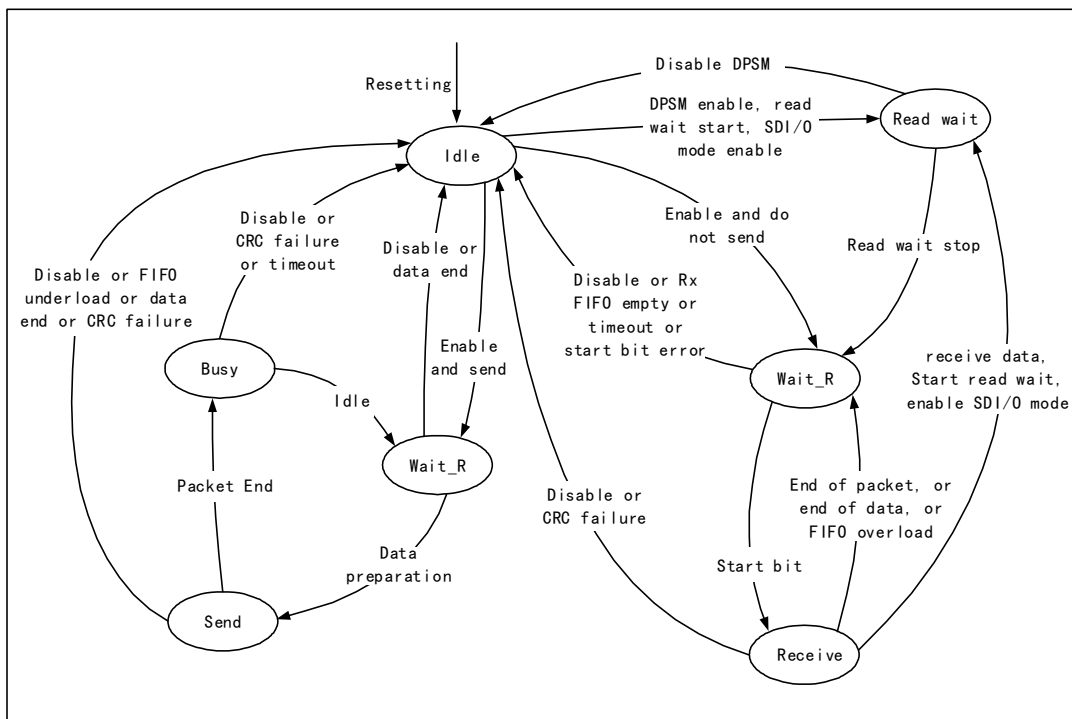
图 32-2 命令状态机



32.2.5 数据状态机

SDIO 的数据工作流程遵循如下图的状态机。

图 32-3 数据状态机



32.3 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。

32.3.1 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 32-4 SDIO 的无响应时序和无数据时序（以 SD 卡为例）

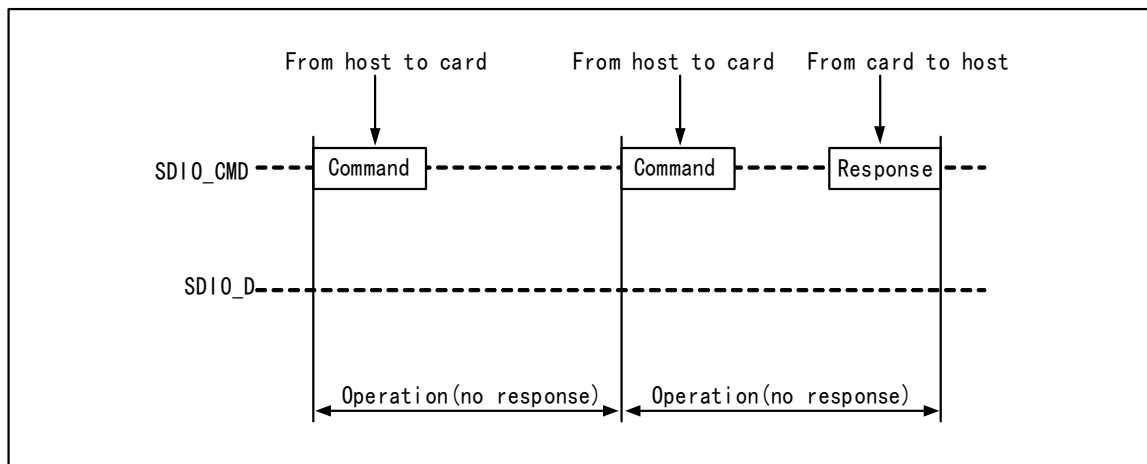


图 32-5 SDIO 的多块读时序（以 SD 卡为例）

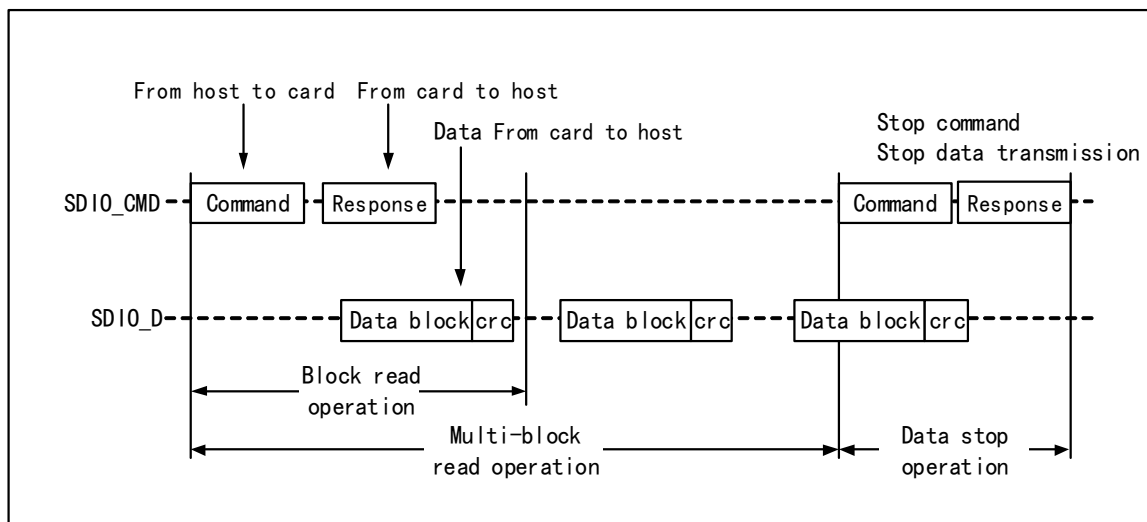


图 32-6 SDIO 的多块写时序（以 SD 卡为例）

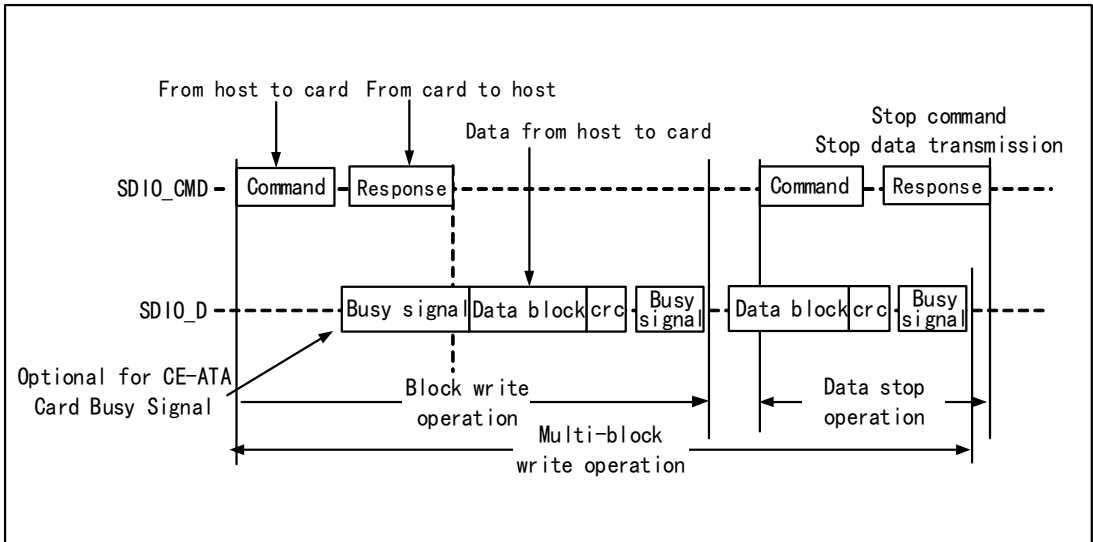


图 32-7 SDIO 的数据流读时序（以 SDIO 卡为例）

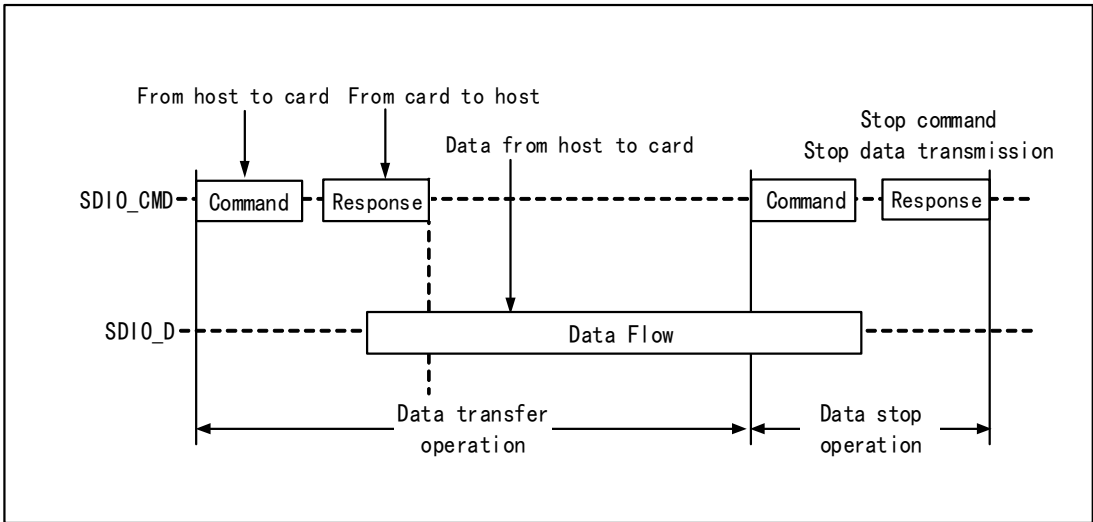
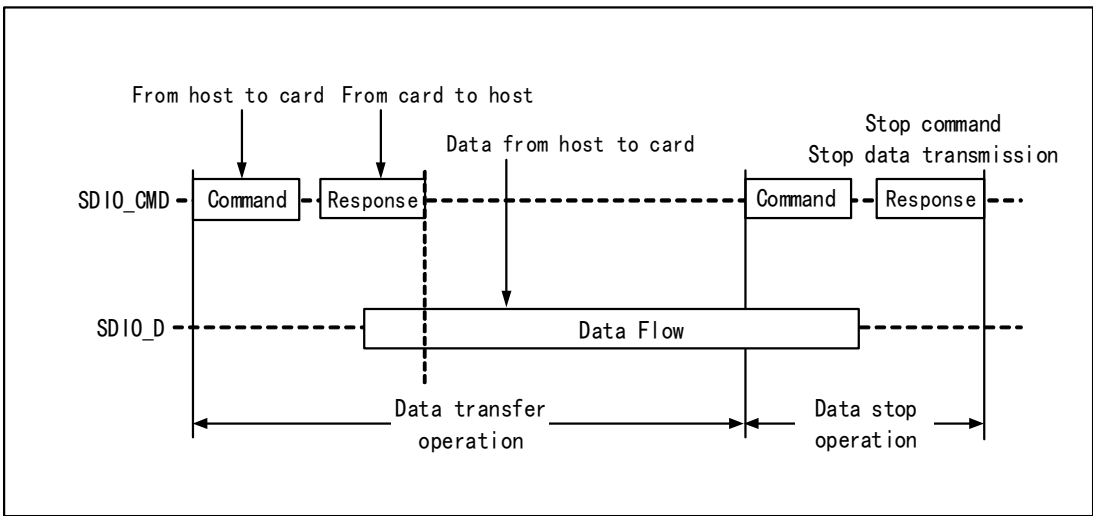


图 32-8 SDIO 的数据流写时序（以 SDIO 卡为例）



32.3.2 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 32-2 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]/6bits	[39:8]/32bits	[7:1]/7bits	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。

32.3.2.1 基础命令

基础命令是 SD 卡所支持的一些较基本的功能。

表 32-3 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。

32.3.2.2 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 32-4 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。

32.3.2.3 块传输的读命令

表 32-5 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。

32.3.2.4 块传输的写命令

表 32-6 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。

32.3.3 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。

32.3.3.1 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 32-7 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]/6bits	[39:8]/32bits	[7:1]/7bits	0
值	0b	0b	跟随 CMD 索引	X	X	1b

32.3.3.2 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。

32.3.3.3 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 32.4.2 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 32-8 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

32.3.3.4 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 32-9 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

32.3.3.5 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 32-10 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
I/O 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
I/O 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。

32.3.3.6 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 32-11 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。

32.3.3.7 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 32-12 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

32.3.3.8 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 32-13 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

32.3.4 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。

开启 RANDOM_LEN_EN 位，配置 DBLOCKSIZE2 表示的字节长度，配置 R32_SDIO_DLEN 表示的传输数据总长度，使能 DTEN 位，填写数据至 R32_SDIO_FIFO，即可完成任意字节长度且带 CRC 校验的块传输。

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 32-9 SD 卡单总线数据传输字节的格式

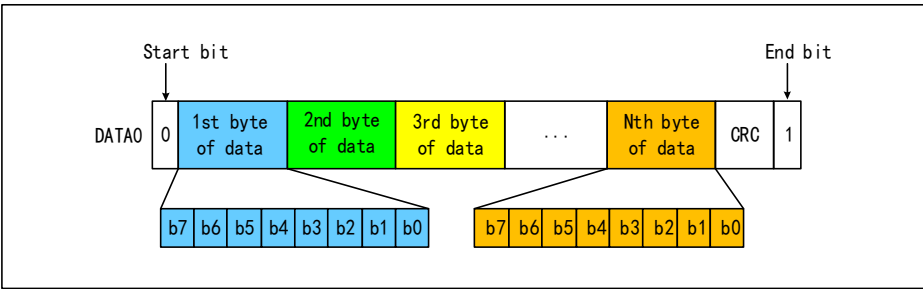


图 32-10 SD 卡 4 总线数据传输字节的格式

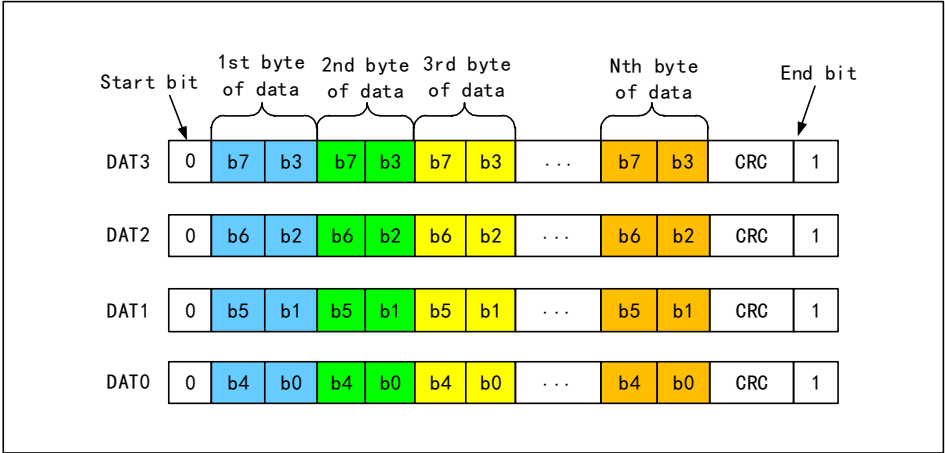


图 32-11 SD 卡单总线数据传输一个 512 位字的格式

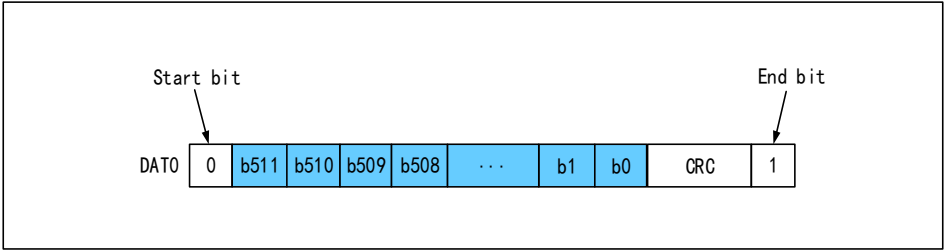
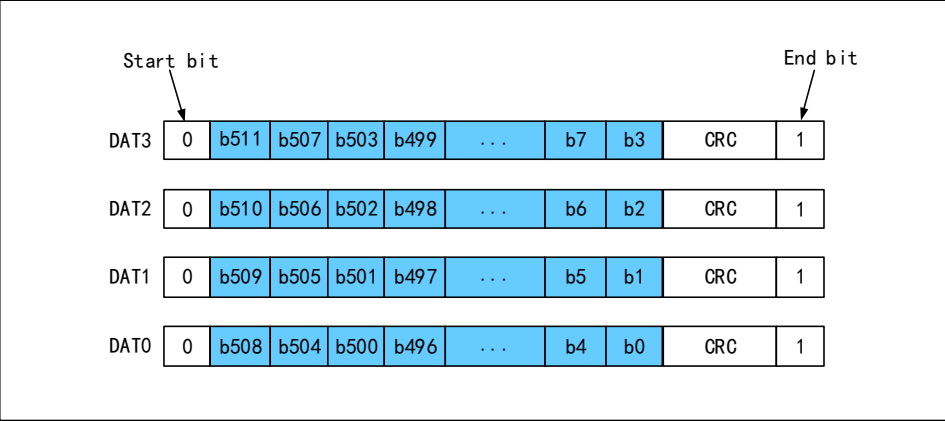


图 32-12 SD 卡 4 总线数据传输一个 512 位字的格式



32.3.5 从模式

开启 SLV_MODE 位，即处于等待主机命令的状态，接收到的命令参数会放在 RESP1，接收到的命令索引会放在 RESP2，同时自动向主机应答 R1 类型的应答，应答参数为 CMDARG 的值。

从模式下的数据收发参考主模式，但从机可以使用 SLV_FORCE_ERR 位强制数据块 CRC 错，从而表示不期望的数据读写。

注：从机只支持 R1 类型的应答，主从通信所用的命令索引和参数含义均由软件定义。

32.4 应用

32.4.1 设备初始化和设备寄存器

32.4.1.1 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 32-14 OCR 寄存器位定义

位次	位定义	描述
[0:3]	保留	支持的 V_{DD33} 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-III 卡状态位	此位被置位表示此卡支持 UHS-III 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

32.4.1.2 CID 寄存器

CID 存储了一些身份识别信息。

表 32-15 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]

32.4.1.3 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 32-16 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	R0	[127:126]
保留	无	6	00_0000b	R0	[125:120]
读数访问时间	TAAC	8	0Eh	R0	[119:112]
用时钟周期表示的读数访问时间	NSAC	8	00h	R0	[111:104]
最大数据发送速度	TRAN_SPEED	8	32h5Ah0Bh2Bh	R0	[103:96]
卡命令类	CCC	12	X1X1101101X1b	R0	[95:84]
读数据块最大长度	READ_BL_LEN	4	9	R0	[83:80]
允许块部分读	READ_BL_PARTIAL	1	0	R0	[79]
块写不对齐	WRITE_BLK_MISALIGN	1	0	R0	[78]
块读不对齐	READ_BLK_MISALIGN	1	0	R0	[77]
执行的 DSR	DSR_IMP	1	X	R0	[76]
保留	无	6	00_0000b	R0	[75:70]
设备大小	C_SIZE	22	XXXXXXh	R0	[69:48]
保留	无	1	0	R0	[47]
单块擦使能	ERASE_BLK_EN	1	1	R0	[46]
擦扇区尺寸	SECTOR_SIZE	7	7Fh	R0	[45:39]
写保护组大小	WP_GRP_SIZE	7	0000000b	R0	[38:32]
写保护组使能	WP_GRP_ENABLE	1	0	R0	[31]
保留	无	2	00b	R0	[30:29]
写速度因素	R2W_FACTOR	3	010b	R0	[28:26]
最大写数据块长度	WRITE_BL_LEN	4	9	R0	[25:22]
允许块部分写	WRITE_BL_PARTIAL	1	0	R0	[21]
保留	无	5	00000b	R0	[20:16]
文件格式组	FILE_FORMAT_GRP	1	0	R0	[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	R0	[11:10]
保留	无	2	00b	R0	[9:8]
CRC	CRC	7	0000000b	RW	[7:1]
未使用，必须使用 1	无	1	1b	R0	[0]

32.4.1.4 RCA 寄存器

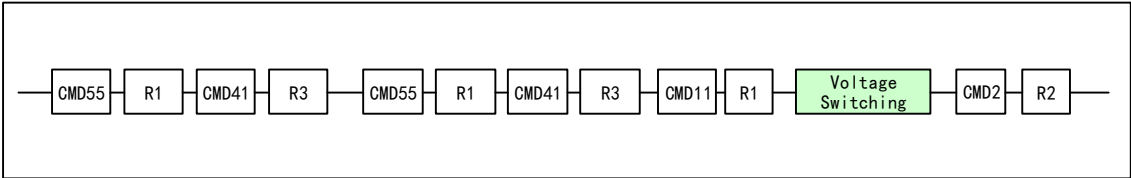
相对卡地址寄存器存储了卡的地址，为 16 位，默认值为 0。

32.4.2 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 32-13 电压切换序列



32.4.3 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。

32.5 中断

32.5.1 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。

32.5.2 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_STA 寄存器的 SDIOIT 位可以得到 SDIO 主机是否收到中断。

32.6 寄存器描述

表 32-17 SDIO 相关寄存器列表

名称	访问地址	描述	复位值
R32_SDIO_POWER	0x40018000	电源寄存器	0x00000000
R32_SDIO_CLKCR	0x40018004	时钟寄存器	0x00000000
R32_SDIO_ARG	0x40018008	命令参数寄存器	0x00000000
R32_SDIO_CMD	0x4001800C	命令寄存器	0x00000000

R32_SDIO_RESPCMD	0x40018010	响应寄存器	0x00000000
R128_SDIO_RESPX	0x40018014	响应参数寄存器	0x00000000
R32_SDIO_DTIMER	0x40018024	数据定时寄存器	0x00000000
R32_SDIO_DLEN	0x40018028	传输长度寄存器	0x00000000
R32_SDIO_DCTLR	0x4001802C	数据控制寄存器	0x00000000
R32_SDIO_DCOUNT	0x40018030	传输计数寄存器	0x00000000
R32_SDIO_STA	0x40018034	状态寄存器	0x00000000
R32_SDIO_ICR	0x40018038	中断清除寄存器	0x00000000
R32_SDIO_MASK	0x4001803C	中断使能寄存器	0x00000000
R32_SDIO_FIFOCNT	0x40018048	FIFO 计数器	0x00000000
R32_SDIO_DCTRL2	0x40018060	数据控制寄存器 2	0x00000000
R32_SDIO_FIFO	0x40018080	FIFO 寄存器	0x00000000

32.6.1 电源寄存器 (R32_SDIO_POWER)

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														PWRCTRL [1:0]	

位	名称	访问	描述	复位值
[31:2]	Reserved	R0	保留。	0
[1:0]	PWRCTRL[1:0]	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

32.6.2 时钟寄存器 (R32_SDIO_CLKCR)

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	HWFC_EN	NEGE_DGE	WIDBUS[1:0]	BYPASS	PWRSAV	CLKEN	CLKDIV[7:0]								

位	名称	访问	描述	复位值
[31:15]	Reserved	R0	保留。	0
14	HWFC_EN	RW	硬件流控使能位, 此位置位后, TXFIFOE 和 RXFIFOE 信号才起作用。 1: 开启硬件流控; 0: 关闭硬件流控。	0

13	NEGEDGE	RW	SDIO_CK 相位选择位： 1：在 HCLK 的下降沿产生 SDIO_CK； 0：在 HCLK 的上升沿产生 SDIO_CK。	0
[12:11]	WIDBUS[1:0]	RW	总线宽度配置域： 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	BYPASS	RW	时钟旁路使能位： 1：SDIO_CK 直接接到 HCLK 上； 0：SDIO_CK 通过分频器分频得出。 注：使用该位时需使能 HWFC_EN 位。	0
9	PWRSV	RW	空闲时钟状态配置位。此位置位后，总线空闲时关闭 SDIO_CK 输出以节省电能。 1：SDIO_CK 只在需要时输出； 0：SDIO_CK 始终输出。	0
8	CLKEN	RW	时钟使能位： 1：SDIO_CK 被允许输出； 0：SDIO_CK 被禁止输出。	0
[7:0]	CLKDIV[7:0]	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：（1）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；

（2）从机时无需配置时钟寄存器。

32.6.3 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[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMDARG[15:0]															

位	名称	访问	描述	复位值
[31:0]	CMDARG	RW	主机模式：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。 从机模式：从机回复应答的 32 位响应参数寄存器。	0

32.6.4 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

Reserved	ATACMD	NIEN	ENCMDCompI	SDIOSuspend	CPSMEN	WAITPEND	WAITINT	WAITRESP[1:0]	CMDINDEX[5:0]

位	名称	访问	描述	复位值
[31:15]	Reserved	RO	保留。	0
14	ATACMD	RW	进行 CE-ATA 命令。如果置此位, CPSM 将转到 CMD61。	0
13	NIEN	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	ENCMDCompI	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	SDIOSuspend	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	CPSMEN	RW	CPSM (命令通道状态机) 使能位。如果置此位, 将使能 CPSM。	0
9	WAITPEND	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	WAITINT	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]	WAITRESP[1:0]	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]	CMDINDEX[5:0]	RW	命令索引域。此域表明具体的命令值。	0

32.6.5 响应寄存器 (R32_SDIO_RESPCMD)

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										RESPCMD[5:0]					

位	名称	访问	描述	复位值
[31:6]	Reserved	RO	保留。	0
[5:0]	RESPCMD[5:0]	RO	这个域记录了收到的响应的索引值。	0

32.6.6 响应参数寄存器 (R128_SDIO_RESPX)

位	名称	访问	描述	复位值
[127:0]	CARDSTATUSx	RO	主机模式: 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	0

			R128_SDIO_RESX 最低位开始存储。	

32.6.6.1 响应参数寄存器高 32 位 (R128_SDIO_RESP1[127:96])

偏移地址: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARDSTATUS1[127:112]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS1[111:96]															

位	名称	访问	描述	复位值
[31:0]	CARDSTATUS1	R0	主机模式: 长响应: 卡状态[127:96]; 短响应: 卡状态[31:0]。 从机模式: 接收命令收到的命令参数。	0

32.6.6.2 响应参数寄存器次高 32 位 (R128_SDIO_RESP2[95:64])

偏移地址: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARDSTATUS2[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS2[79:64]															

位	名称	访问	描述	复位值
[31:0]	CARDSTATUS2	R0	主机模式: 卡状态[95:64]; 从机模式: 接收命令收到的命令索引。	0

32.6.6.3 响应参数寄存器次低 32 位 (R128_SDIO_RESP3[63:32])

偏移地址: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARDSTATUS3[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS3[47:32]															

位	名称	访问	描述	复位值
[31:0]	CARDSTATUS3	R0	主机模式: 卡状态[63:32]。	0

32.6.6.4 响应参数寄存器低 32 位 (R128_SDIO_RESP4[31:0])

偏移地址: 0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARDSTATUS4[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS4[15:0]															

位	名称	访问	描述	复位值
[31:0]	CARDSTATUS4	R0	主机模式：卡状态[31:0]。	0

32.6.7 数据定时寄存器 (R32_SDIO_DTIMER)

偏移地址：0x24

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

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

位	名称	访问	描述	复位值
[31:0]	DATATIME	RW	数据超时时长。以 SDIO_CK 的周期数为单位。	0

32.6.8 传输长度寄存器 (R32_SDIO_DLEN)

偏移地址：0x28

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

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

位	名称	访问	描述	复位值
[31:25]	Reserved	R0	保留。	0
[24:0]	DALENGTH[24:0]	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

32.6.9 数据控制寄存器 (R32_SDIO_DCTRL)

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				SDIO EN	RW MOD	RW STO P	RW STA RT	DBLOCKSIZE[3:0]				DMA EN	DTMO DE	DTD IR	DTEN

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0
11	SDIOEN	RW	SDIO 使能位，此位被置位后，DPSM 可以执行 SDIO	0

			卡特定的一些操作。	
10	RWMOD	RW	读等待模式： 1：使用 SDIO_D2 控制读等待； 0：停止 SDIO_CK 控制读等待。	0
9	RWSTOP	RW	读等待停止位，如果 RWSTART 位被置位，那么读等待将被停止。	0
8	RWSTART	RW	读等待开始位，置此位将执行读等待操作。	0
[7:4]	DBLOCKSIZE[3:0]	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： 1：使能 DMA； 0：关闭 DMA。	0
2	DTMODE	RW	传输模式设置位，置此位设置传输模式。 1：流传输； 0：块传输。	0
1	DTDIR	RW	传输方向设置位。置此位设置传输方向。 1：卡到控制器； 0：控制器到卡；	0
0	DTEN	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

32.6.10 传输计数寄存器（R32_SDIO_DCOUNT）

偏移地址：0x30

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

位	名称	访问	描述	复位值
[31:25]	Reserved	RO	保留。	0
[24:0]	DATACOUNT[24:0]	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

32.6.11 状态寄存器（R32_SDIO_STA）

偏移地址：0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CEAT AEND	SDIOI T	RXDAV L	TXDAV L	RXFI FOE	TXFI FOE	RXFI FOF	TXFI FOF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RXFIFOHF	TXFIFOHE	RXACT	TXACT	CMDACT	DBCKEND	STBITERR	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
23	CEATAEND	R0	此位被置位时, CMD61 接收到 CE-ATA 完成信号。	0
22	SDIOIT	R0	此位被置位时, SDIO 收到设备中断。	0
21	RXDAVL	R0	此位被置位时, 接收 FIFO 数据可用。	0
20	TXDAVL	R0	此位被置位时, 发送 FIFO 中的数据可用。	0
19	RXFIFOE	R0	此位被置位时, 接收 FIFO 空。	0
18	TXFIFOE	R0	此位被置位时, 发送 FIFO 空。	0
17	RXFIFO	R0	此位被置位时, 接收 FIFO 满。	0
16	TXFIFO	R0	此位被置位时, 发送 FIFO 满。	0
15	RXFIFOHF	R0	此位被置位时, 接收 FIFO 半满。	0
14	TXFIFOHE	R0	此位被置位时, 发送 FIFO 半空。	0
13	RXACT	R0	此位被置位时, 正在接收数据。	0
12	TXACT	R0	此位被置位时, 正在发送数据。	0
11	CMDACT	R0	此位被置位时, 正在传输命令。	0
10	DBCKEND	R0	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	STBITERR	R0	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	DATAEND	R0	此位被置位时, 数据传输结束(传输计数器为零)。	0
7	CMDSENT	R0	此位被置位时, 命令已经发送出去。	0
6	CMDREND	R0	此位被置位时, 已经收到响应, CRC 检验成功。	0
5	RXOVERR	R0	此位被置位时, 接收 FIFO 上溢。	0
4	TXUNDERR	R0	此位被置位时, 发送 FIFO 下溢。	0
3	DTIMEOUT	R0	此位被置位时, 数据超时。	0
2	CTIMEOUT	R0	此位被置位时, 命令超时超过了 64 个 SDIO_CLK 周期。	0
1	DCRCFAIL	R0	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	CCRCFAIL	R0	此位被置位时, 已经收到响应, 但 CRC 检验失败。	0

32.6.12 中断清除寄存器 (R32_SDIO_ICR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CEATA ENDC	SDIOI TC	Reserved					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DBCKE NDC	STBIT ERRC	DATA ENDC	CMDSENTC	CMDRE ENDC	RXOV ERRC	TXUND ERRC	DTIME OUTC	CTIME OUTC	DCRCF AILC	CCRCF AILC

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
23	CEATAENDC	RW	置此位清除状态寄存器中的 CEATAEND 位。	0
22	SDIOITC	RW	置此位清除状态寄存器中的 SDIOIT 位。	0
[21:11]	Reserved	RW	保留。	0
10	DBCKENDC	RW	置此位清除状态寄存器中的 DBCKEND 位。	0
9	STBITERRC	RW	置此位清除状态寄存器中的 STBITERR 位。	0
8	DATAENDC	RW	置此位清除状态寄存器中的 DATAEND 位。	0
7	CMDSENTC	RW	置此位清除状态寄存器中的 CMDSENT 位。	0
6	CMDREND	RW	置此位清除状态寄存器中的 CMDREND 位。	0
5	RXOVERRC	RW	置此位清除状态寄存器中的 RXOVERR 位。	0
4	TXUNDERRC	RW	置此位清除状态寄存器中的 TXUNDERR 位。	0
3	DTIMEOUTC	RW	置此位清除状态寄存器中的 DTIMEOUT 位。	0
2	CTIMEOUTC	RW	置此位清除状态寄存器中的 CTIMEOUT 位。	0
1	DCRCFAILC	RW	置此位清除状态寄存器中的 DCRCFAIL 位。	0
0	CCRCFAILC	RW	置此位清除状态寄存器中的 CCRCFAIL 位。	0

32.6.13 中断使能寄存器 (R32_SDIO_MASK)

偏移地址: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CEATA ENDIE	SDIOI TIE	RXDAV LIE	TXDAV LIE	RXFIF OEIE	TXFIF OEIE	RXFIF OFIE	TXFIF OFIE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFIF OHFIE	TXFIF OHEIE	RXACT IE	TXACT IE	CMDAC TIE	DBCKE NDIE	STBIT ERRIE	DATAE NDIE	CMDSE NTIE	CMDRE NDIE	RXOVE RRIE	TXUND ERRIE	DTIME OUTIE	CTIME OUTIE	DCRCF AILIE	CCRCF AILIE

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
23	CEATAENDIE	RW	置此位后, 状态寄存器在置 CEATAEND 位时将产生中断。	0
22	SDIOITIE	RW	置此位后, 状态寄存器在置 SDIOIT 位时将产生中断。	0
21	RXDAVLIE	RW	置此位后, 状态寄存器在置 RXDAVL 位时将产生中断。	0
20	TXDAVLIE	RW	置此位后, 状态寄存器在置 TXDAVL 位时将产生中断。	0
19	RXFIFOEIE	RW	置此位后, 状态寄存器在置 RXFIFOE 位时将产生中断。	0
18	TXFIFOEIE	RW	置此位后, 状态寄存器在置 TXFIFOE 位时将产生中断。	0
17	RXFIFOFIE	RW	置此位后, 状态寄存器在置 RXFIFOF 位时将产生中断。	0
16	TXFIFOFIE	RW	置此位后, 状态寄存器在置 TXFIFOF 位时将产生中断。	0
15	RXFIFOHFIE	RW	置此位后, 状态寄存器在置 RXFIFOHF 位时将产生中断。	0

14	TXFIFOHEIE	RW	置此位后, 状态寄存器在置 TXFIFOHE 位时将产生中断。	0
13	RXACTIE	RW	置此位后, 状态寄存器在置 RXACT 位时将产生中断。	0
12	TXACTIE	RW	置此位后, 状态寄存器在置 TXACT 位时将产生中断。	0
11	CMDACTIE	RW	置此位后, 状态寄存器在置 CMDACT 位时将产生中断。	0
10	DBCKENDIE	RW	置此位后, 状态寄存器在置 DBCKEND 位时将产生中断。	0
9	STBITERRIE	RW	置此位后, 状态寄存器在置 STBITERR 位时将产生中断。	0
8	DATAENDIE	RW	置此位后, 状态寄存器在置 DATAEND 位时将产生中断。	0
7	CMDSENTIE	RW	置此位后, 状态寄存器在置 CMDSENT 位时将产生中断。	0
6	CMDRENDIE	RW	置此位后, 状态寄存器在置 CMDREND 位时将产生中断。	0
5	RXOVERRIE	RW	置此位后, 状态寄存器在置 RXOVERR 位时将产生中断。	0
4	TXUNDERRIE	RW	置此位后, 状态寄存器在置 TXUNDERR 位时将产生中断。	0
3	DTIMEOUTIE	RW	置此位后, 状态寄存器在置 DTIMEOUT 位时将产生中断。	0
2	CTIMEOUTIE	RW	置此位后, 状态寄存器在置 CTIMEOUT 位时将产生中断。	0
1	DCRCFAILIE	RW	置此位后, 状态寄存器在置 DCRCFAIL 位时将产生中断。	0
0	CCRCFAILIE	RW	置此位后, 状态寄存器在置 CCRCFAIL 位时将产生中断。	0

32.6.14 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
FIFOCOUNT[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFOCOUNT[15:0]															

位	名称	访问	描述	复位值
[31:0]	FIFOCOUNT[31:0]	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

32.6.15 数据控制寄存器 2 (R32_SDIO_DCTRL2)

偏移地址：0x60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					SLV_CK_PHASE	SLV_FORCE_ERR	SLV_MODE	Reserved							RANDOM_LEN_EN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DBLOCKSIZE2										

位	名称	访问	描述	复位值
[31:27]	Reserved	R0	保留。	0
26	SLV_CK_PHASE	RW	SLV_MODE 为 1 时，从模式输出 DATA 时的相位选择位： 1：在内部 SDCK 下降沿输出 DATA； 0：在内部 SDCK 上升沿输出 DATA。 SLV_MODE 为 0 时： 1：DTEN 位写 0 不清空 FIFO； 0：DTEN 位写 0 清空 FIFO。	0
25	SLV_FORCE_ERR	RW	从机模式下软件强制数据块 CRC 错误 1：软件强制数据块 CRC 为错误值； 0：使用硬件正确的 CRC 值。	0
24	SLV_MODE	RW	从模式使能位： 1：控制器 CPSM 和 DPSM 工作在从模式下； 0：控制器工作在主机模式下。	0
[23:17]	Reserved	R0	保留。	0
16	RANDOM_LEN_EN	RW	数据块任意字节长度使能位： 1：数据块可以是 0 - 2048 的任意长度，长度为 DBLOCKSIZE2； 0：数据块只能是 $2^{DBLOCKSIZE}$ 字节。	0
[15:12]	Reserved	R0	保留。	0
[11:0]	DBLOCKSIZE2	RW	任意字节长度模式的数据块长度域。	0

32.6.16 FIFO 寄存器 (R32_SDIO_FIFO)

偏移地址：0x80

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

位	名称	访问	描述	复位值
[31:0]	FIFODATA[31:0]	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 33 章 SD/EMMC 控制器 (SDMMC)

本章模块描述仅适用于 CH32H417、CH32H416 微控制器产品。

系统提供 1 组 SDMMC 控制器主机/从机接口,传输时钟最高可达 200MHz,支持 1/4/8 线通讯模式,支持双沿采样,可外接 SD/TF 卡、EMMC 卡等器件。应用程序代码可灵活设置数据收发的各种命令、应答包、有效数据包的模式和长度,双缓冲长度切换界限等参数。

注: SDMMC 在使用单线或者四线模式时,未使用的数据线对应的 GPIO 引脚不能做复用输出使用,可用于复用输入,也可用于通用 GPIO 输出。

33.1 主要特征

- 支持 SD 物理层 1.0、2.0 规范,支持 SD3.0 规范的 UHS-I SDR50、DDR50 和 SDR104 模式
- 符合 EMMC 卡 4.4 和 4.5.1 规范;支持 EMMC 卡 5.0 规范的 HS200, HS400
- 通讯模式支持单线、四线、八线模式
- 最高通讯时钟单沿可达 200MHz、双沿可达 180MHz
- 支持双沿采样
- 灵活可设置的数据包长度、命令格式、应答状态
- 提供硬件自动在数据块间隔时停止时钟功能
- 支持 SD 卡、SDIO 卡、EMMC 卡等符合 SD 接口协议的设备
- 支持 SDIO 从机接口,可以完成支持 SDIO 主机接口的芯片的数据交换
- DMA 双缓冲功能

33.2 功能描述

33.2.1 通讯时钟

- 频率配置

控制器接口提供了 2 种时钟模式:低速模式和高速模式。一般 SD 协议规定对于 SD 接口的设备的前期初始化使用 400KHz 左右的通讯时钟来保证更好的通讯兼容性,当获取了外接设备的参数信息后,可以根据其内部支持的配置,切换到更高的时钟频率进行通讯。

设置 R16_EMMC_CLK_DIV 寄存器的 bit9 选择时钟模式,设置 bit[4:0]分频值可得到最终的 SDIO 接口输出时钟。bit8 控制 EMMC 外设是否输出时钟信号到 MSDCK 引脚,需要保证 MSDCK 引脚的 IO 模式配置为推挽输出。

- 相位配置

在和外部 SDIO 接口设备通讯时,由于较高的时钟频率、硬件走线、器件特性等因素,会导致信号采样出错。

系统 SDIO 控制器主机接口提供了输出时钟相位翻转(偏移 180°)功能,通过设置寄存器 R16_EMMC_CLK_DIV 的 bit10 位写 1,可以让输出的时钟信号物理翻转,但是控制器内部的时钟仍然保持不变。这种方式可以调节通讯时序。

注:在较高 SDCLK 频率下,若推荐的时序调节寄存器值不能稳定通信,用户需发起总线采样 tuning 序列寻找更佳的采样点。

33.2.2 命令发送及响应

SDIO 主机接口向从机发出的命令长度固定的 48bit 数据包,在 MCMD 线上串行传送。有的命令不需要从机响应,有的命令需要从机响应不同长度的应答包或数据包。

- 一般的命令格式：开始位（0）+传输位（1）+命令索引+参数+CRC7+结束位（1）

其中开始位、传输位、CRC7、结束位由硬件自动填充，命令索引和参数根据命令的不同而不同，需要应用代码传给硬件控制器。通过写 R32_EMMC_ARGUMENT 寄存器，配置 SD 命令中的参数域；写 R16_EMMC_CMD_SET 寄存器的 bit[5:0]完成命令索引的填充，同时该寄存器的 bit[11:8]配置硬件此命令期望的应答包检测。对 R16_EMMC_CMD_SET 寄存器的写操作将触发硬件在 MCMD 信号线上完成命令包序列传输。

- 短响应格式：开始位（0）+传输位（0）+命令索引+参数+CRC7+结束位（1）
- 长响应格式：开始位（0）+传输位（0）+保留域（111111b）+数据+结束位（1）

从机响应的应答包按长度分为 48 位和 136 位，其中对用户有效数据位分别位 32 位和 128 位。当主机命令发送后，可以通过查询 R16_EMMC_INT_FG 寄存器，如果标志位 RB_EMMC_IF_CMDDONE 置 1，即可读取 R32_EMMC_RESPONSE 寄存器获取从机接口应答包中的有效数据。如果应答包异常，寄存器 R16_EMMC_INT_FG 的 bit[2:0]有相应标志置位。

33.2.3 连续读取多个数据块

对于 SD 协议命令中对于读取数据（通过数据线传输）的操作，需要如下配置：

1) 初始化：配置 R16_EMMC_CONTROL 寄存器，设置命令和数据的采样边沿，收发的数据线宽度，启用 DMA 功能。配置 R8_EMMC_TIMEOUT 寄存器 RB_EMMC_TOCNT_MASK 位设置命令超时时间和数据超时时间。配置 R32_EMMC_DMA_BEG1 寄存器，设置读出数据保存的 DMA 起始地址。配置 R32_EMMC_BLOCK_CFG 寄存器，设置传输过程中单次数据块长度和总共需要读取的数据块个数。配置 R32_EMMC_TRAN_MODE 寄存器，设置 DMA 传输方向为“SD 到控制器”，可选设置是否需要块传输间隔时间自动停止时钟，给应用代码足够的时间进行处理工作，之后软件再启动传输。

2) 发送命令给 SDIO 从机接口设备，此命令是需要从机设备从数据线传输数据给主机接口的命令，例如 SD 协议中的 CMD17、CMD18。

3) 控制器接口从数据线读取数据：当从机接口设备响应了主机命令后，如果设备准备好数据传输，会基于主机时钟信号从数据线上输出数据，起始信号为低电平。主机控制器端采样信号并由内部 DMA 转移到用户配置的 SRAM 区域。每转移 RB_EMMC_BKSIZE_MASK 设置大小的数据，表示完成一个数据块传输，同时内部总传输数据块数目减 1，直到所有的数据块传输结束，内部 DMA 将不再进行数据转移。此过程中，每完成一个数据块的读取，硬件将会置位 RB_EMMC_IF_BKGAP 标志，当所有的数据块读完，硬件将会置位 RB_EMMC_IF_TRANDONE 标志，如果控制器在等待接收过程中发生数据超时或者接收数据的 CRC 错误，将会分别置位 RB_EMMC_IF_DATTMO 和 RB_EMMC_IF_TRANERR 标志。应用代码可以配置中断使能寄存器 R16_EMMC_INT_EN 的相应中断控制位，当对应中断标志置位时，可以触发 EMMC 的中断服务。

4) 检测状态：R32_EMMC_STATUS 寄存器可是实时查询数据线 MD0 和命令线 MCMD 的电平状态，在一些协议规定的时序上做判断，MASK_BLOCK_NUM 域记录了当前命令传输中已经成被 DMA 成功转移的数据块个数。

33.2.4 连续写入多个数据块

1) 初始化：配置 R16_EMMC_CONTROL 寄存器，设置命令和数据的采样边沿，收发的数据线宽度，启用 DMA 功能。配置 R8_EMMC_TIMEOUT 寄存器 RB_EMMC_TOCNT_MASK 位设置命令超时时间和数据超时时间。命令发送成功后，再配置 R32_EMMC_BLOCK_CFG 寄存器，设置传输过程中单次数据块长度和总共需要读取的数据块个数，即启动数据传输。配置 R32_EMMC_TRAN_MODE 寄存器，设置 DMA 传输方向为“控制器到 SD”。

2) 发送命令给 SDIO 从机接口设备，此命令是需要主机接口通过数据线向从机设备传输数据的命令，例如 SD 协议中的 CMD24、CMD25。

3) 控制器接口向数据线写数据：当从机接口设备响应了主机命令后，作出响应包并释放了数据线状态（非低电平），此时主机可以驱动数据线输出一个数据块。通过写 DMA 寄存器 R32_EMMC_DMA_BEG1

或者向 R32_EMMC_WRITE_CONT 寄存器执行写操作，都会启动主机接口驱动数据线，区别在于，操作寄存器 R32_EMMC_DMA_BEG1，硬件会将此时 R32_EMMC_DMA_BEG1 寄存器的值加载到内部，并从此地址取数据送入控制器发送数据；而对 R32_EMMC_WRITE_CONT 寄存器执行写操作，硬件将使用内部当前 DMA 移动到的位置开始向后继续取 SRAM 数据送入控制器，所以如果需要改变数据的 DMA 地址（与之前非连续）需要重新写 R32_EMMC_DMA_BEG1 寄存器执行继续数据线数据输出，如果与之前写数据的地址连续，硬件内部会自动取数据地址进行移动，可以直接对 R32_EMMC_WRITE_CONT 寄存器执行写操作来让控制器继续数据线数据输出。每输出 RB_EMMC_BKSIZE_MASK 设置大小的数据，表示完成一个数据块传输，同时内部总传输数据块数目减 1，直到所有的数据块传输结束，内部 DMA 将不再进行数据输出。此过程中，每完成一个数据块的读取，硬件将会置位 RB_EMMC_IF_BKGAP 标志，当所有的数据块读完，硬件将会置位 RB_EMMC_IF_TRANDONE 标志，如果控制器在等待接收过程中发生数据超时或者接收数据的 CRC 错误，将会分别置位 RB_EMMC_IF_DATTIME 和 RB_EMMC_IF_TRANERR 标志。应用代码可以配置中断使能寄存器 R16_EMMC_INT_EN 的相应中断控制位，当对应中断标志置位时，可以触发 EMMC 的中断服务。

注 1：控制器接口向外部设备写入数据块时，需要等待命令发送并应答完成后才能设置 DMA 地址，因为设置 DMA 地址的操作即通知硬件驱动数据线，此时命令如果还没有发送完成，设备将不会接受数据。

注 2：在某次命令执行的传输过程中（包括数据读和数据写），如果出现 CRC 错误（RB_EMMC_IF_TRANERR），硬件将不计入成功计数，也不会对总传输块计数进行递减，会继续下一个块的传输。但是 DMA 仍然会将此 CRC 错误的数据块放入相应的 SRAM 地址区域。应用代码要注意此刻的处理。

33.2.5 DMA 双缓冲功能

系统提供了 DMA 双缓冲功能，有 2 个 DMA 地址寄存器 R32_EMMC_DMA_BEG1 和 R32_EMMC_DMA_BEG2，以及 DMA 切换的数据块长度阈值 RB_EMMC_DMATN_CNT。当 R32_EMMC_DMA_BEG1 寄存器地址开始的数据块长度阈值传输完成，硬件将切换到 R32_EMMC_DMA_BEG2 设置地址区域进行数据传输，并在达到传输阈值后再次切换到 R32_EMMC_DMA_BEG1 区，以此循环，直到 RB_EMMC_BKNUM_MASK 配置的总数据块数量全部传输完成。

每次对 R32_EMMC_BLOCK_CFG 寄存器的 RB_EMMC_BKNUM_MASK 写入非 0 数据，将会启动数据传输，同时硬件也会将 DMA 指向 R32_EMMC_DMA_BEG1 地址区域。传输过程中的标志置位条件和前面描述一致。应用代码可选择开启相应中断位触发 EMMC 中断服务。

33.2.6 从机模式

从机模式操作

从机模式硬件只支持 R1 类型的应答，因此 R16_EMMC_CMD_SET 寄存器在从机模式下不起作用。

1) 主机查询从机数据状态：

从机提前将自己期望接收/发送数据的状态填入 ARGUMENT 寄存器，主机发送 CMD13 查询从机状态，主机根据应答决定下一步是读从机还是写从机。

注：是否用 CMD13 以及从机响应中的位定义均由软件自定义。

2) 主机发送 CMD18 读多块：

主机的 CMD18 命令参数中不再是扇区地址，而是本次要读的块数，从机收到接收命令中断后，配置 DMA 地址，配置 R32_EMMC_TRAN_MODE，将接收命令参数寄存器中的块数读出填入 R32_EMMC_BLOCK_CFG 寄存器。

3) 主机发送 CMD25 写多块：

主机的 CMD25 命令参数中不再是扇区地址，而是本次要写的块数，从机收到接收命令中断后，配置 DMA 地址，配置 R32_EMMC_TRAN_MODE，将接收命令参数寄存器中的块数读出填入 R32_EMMC_BLOCK_CFG 寄存器。

从机错误处理

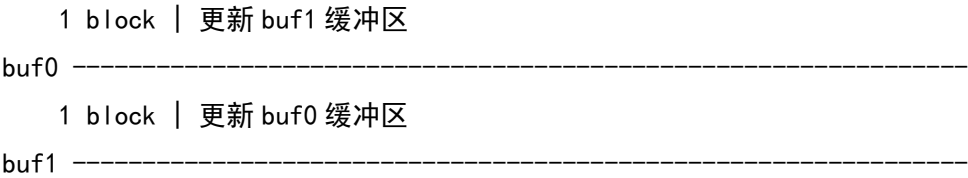
从机会对所有主机的命令进行 R1 类型的应答，响应参数为从机 ARGUMENT 寄存器的值，从机可能因为接收命令时出现 CRC 错误不响应主机的应答，则主机应该重试当前命令。

如果从机在收发过程中出现内部 FIFO 溢出，硬件会自动强制本次收发的数据块为 CRC 错误，以此通知主机传输有问题；软件也可以设置 RB_SLV_FORCE_ERR 来强制传输错误的 CRC 块。

从机硬件会解析一个特殊的命令 CMD63，用来简化从机软件在传输出错时的处理。若主机收发过程中出现 CRC 错误，则主机立刻发送 CMD63，从机在收到 CMD63 时，软件需要去复位掉数据状态机，硬件会自动将 DMA 地址指向上一个错误包的起始地址，这样等主从协商好可以继续传输时，对于读操作，从机软件只用设置 R32_EMMC_BLOCK_CFG 中的块数，对于写操作，软件设置完 R32_EMMC_BLOCK_CFG 块数并且写 RESPONSE3 寄存器，即可完成重传并继续传输。

双缓冲模式

如果从机使用双缓冲模式，若主机读取 buf0 的最后一块数据时发生错误，则主机会在 buf1 的第一块数据期间发送 CMD63 以指示上一包 CRC 错误，因此从机应该在 buf1 的第一个块成功传输后，才更新掉 buf0 的数据，新增的中断位 RB_SIF_SLV_BUF_RELEASE 即在新 buf 的第一个扇区成功传输时产生。



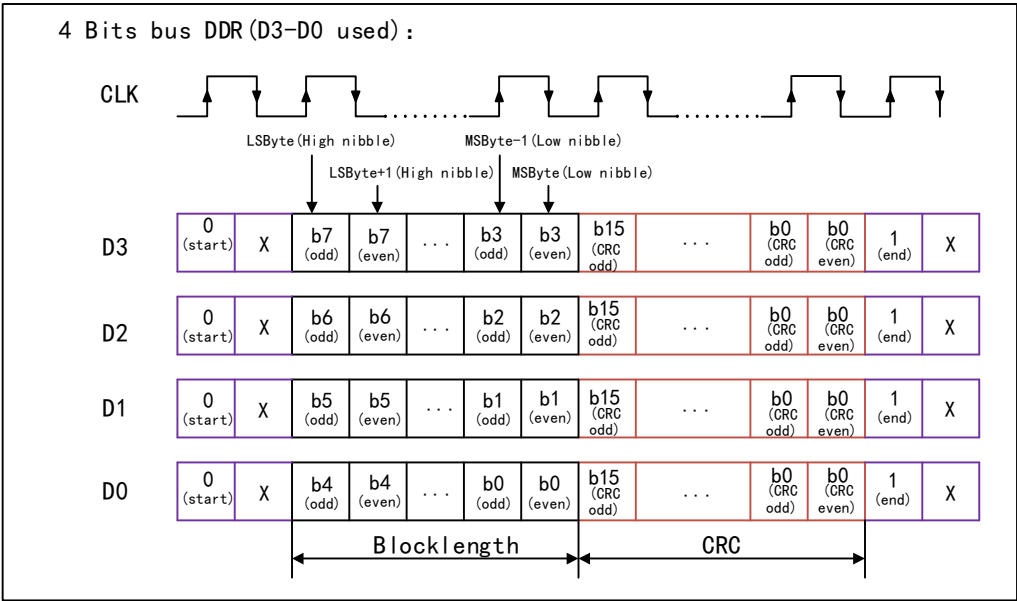
DDR 模式

支持 4 线和 8 线下的 DDR 模式。

1) 4 线 DDR 模式：

即通过时钟的上升沿和下降沿，输出和采样数据，提高数据传输速率，应用时，应使能 4 线模式寄存器和 DDR 模式寄存器；

图 33-1 4 线的 DDR 模式

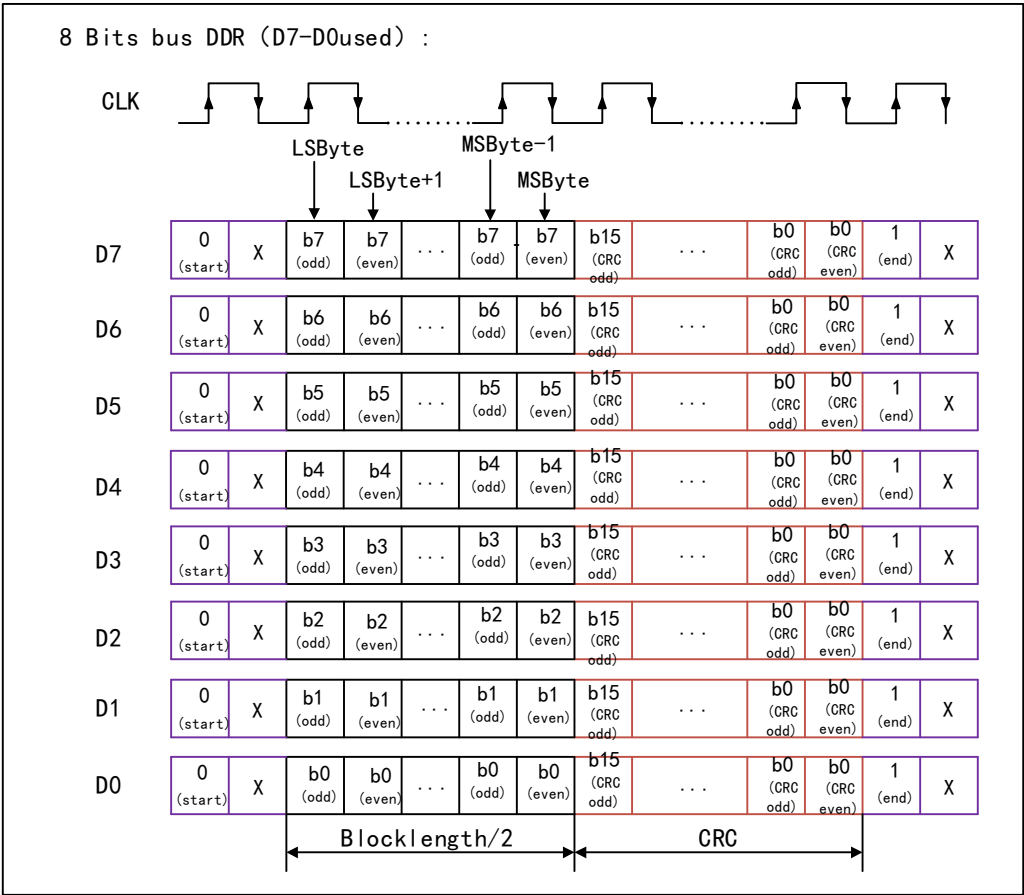


2) 8 线的 DDR 模式：

即通过时钟的上升沿和下降沿，输出和采样数据，提高数据传输速率，应用时，应使能 8 线模式

寄存器和 DDR 模式寄存器；

图 33-2 8 线的 DDR 模式



R16_EMMC_CLK_DIV	0x40024038	时钟配置寄存器	0x0213
R32_EMMC_DMA_BEG2	0x4002403C	DMA 起始地址 2 寄存器	0x00000000
R32_EMMC_TUNE_DAT0	0x40024040	数据输出延迟寄存器	0x00000000
R32_EMMC_TUNE_DAT1	0x40024044	数据输入延迟寄存器	0x00000000
R32_EMMC_TUNE_CLK_CMD	0x40024048	时钟和命令延迟寄存器	0x00000000

33.3.1 命令参数寄存器 (R32_EMMC_ARGUMENT)

偏移地址: 0x00

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

位	名称	访问	描述	复位值
[31:0]	EMMC_ARGUMENT	RW	主机模式: SD/eMMC 的 32 位命令参数寄存器。 从机模式: 从机回复应答的 32 位响应参数寄存器。	0

33.3.2 命令设置寄存器 (R16_EMMC_CMD_SET)

偏移地址: 0x04

位	名称	访问	描述	复位值
[15:12]	Reserved	RO	保留。	0
11	RB_EMMC_CKIDX	RW	校验应答的命令索引: 1: 需要; 0: 不需要。	0
10	RB_EMMC_CKCRC	RW	校验应答的 CRC: 1: 需要; 0: 不需要。	0
[9:8]	RB_EMMC_RPTY_MASK	RW	期望应答类型: 00: 无应答; 01: 应答长度为 136 位; 10: 应答长度为 48 位; 11: 应答长度为 48 位, 且为 R1b 类型应答。	0
[7:6]	Reserved	RO	保留。	0
[5:0]	RB_EMMC_CMDIDX_MASK	RW	当前发送命令的索引号。	0

33.3.3 应答参数寄存器 (R32_EMMC_RESPONSE)

偏移地址: 0x08、0x0C、0x10、0x14、0x18

位	名称	访问	描述	复位值
[31:0]	R32_EMMC_RESPONSE0	RO	应答参数寄存器 0。	0
[63:32]	R32_EMMC_RESPONSE1	RO	应答参数寄存器 1。	0
[95:64]	R32_EMMC_RESPONSE2	RO	主机模式: 应答参数寄存器 2; 从机模式: 接收命令收到的命令索引。	0
[127:96]	R32_EMMC_RESPONSE3	RO	主机模式: 应答参数寄存器 3; 从机模式: 接收命令参数寄存器。	0

[127:96]	R32_EMMC_WRITE_CONT	WO	复用 EMMC_RESPONSE3 寄存器，用于在多块写过程中，启动写操作。	0

注：应答长度为 136 位时，有效数据为 128 位，应答长度为 48 位时，有效数据长度为 32 位，R32_EMMC_RESPONSE3 寄存器存放的是应答的有效数据参数。

R32_EMMC_RESPONSE3 寄存器存在复用，寄存器复用为：在使用 CMD25 命令向卡连续写入多块数据时，在块传输完成时，写此寄存器的动作，控制器将自动进行地址偏移块长度，继续写数据。如果需要更改 DMA 地址，则写 DMA 地址寄存器，就会启动写操作，就不用再用写寄存器的方式启动。

33.3.4 控制寄存器（R16_EMMC_CONTROL）

偏移地址：0x18

位	名称	访问	描述	复位值
[15:10]	Reserved	RO	保留。	0
9	RB_SLV_FORCE_ERR	RW	从机模式下软件强制数据块 CRC 错误： 1：软件强制数据块 CRC 为错误值； 0：使用硬件正确的 CRC 值。	0
8	RB_SLV_MODE	RW	从机模式使能位： 1：从机模式，SDCK 输入； 0：主机模式，SDCK 输出。	0
[7:6]	Reserved	RO	保留。	0
5	RB_EMMC_NEGSMP	RW	CMD 和 Data 信号线采样模式选择位： 1：下降沿采样； 0：上升沿采样。	0
4	RB_EMMC_RST_LGC	RW	模块内部逻辑收发复位： 1：执行复位； 0：正常工作；	1
3	RB_EMMC_DMAEN	RW	控制器 DMA 使能： 1：使能 DMA； 0：关闭 DMA。	0
2	RB_EMMC_ALL_CLR	RW	SD 控制器逻辑复位： 1：复位 SD 控制器逻辑，需要软件清零； 0：正常工作。	1
[1:0]	RB_EMMC_LW_MASK	RW	收发数据有效数据宽度： 00：收发器仅使用 D[0]，单数据线； 01：收发器使用 D[3:0]，4 数据线； 10：收发器使用 D[7:0]，8 数据线。	01b

33.3.5 超时控制寄存器（R8_EMMC_TIMEOUT）

偏移地址：0x1C

位	名称	访问	描述	复位值
[7:4]	Reserved	RO	保留。	0
[3:0]	RB_EMMC_TOCNT_MASK	RW	应答/数据超时配置。 非 0：设置超时时间，有效数值 1-15； 计算：模块时钟周期 * 4194304 * RB_EMMC_TOCNT_MASK。 例如：SDCLK 周期为 10ns，写入 12，则超时时间	0xC

			为 $10\text{ns} * (4194304) * (12) = 503\text{ms}$ 。 0: 不可用。	

注1: 上述数据超时包含下列4种情况:

- 1) R1b 应答之后的 D[0] busy 超时;
- 2) 写数据块时, CRC status 之后的 D[0] busy 超时;
- 3) 写数据块时, 等待 CRC status 超时;
- 4) 读数据块时, 等待起始位超时。

注2: 命令的应答同样支持超时机制, 如果应答超时, 则由中断寄存器中的 R16_EMMC_INT_FG 中断给出。命令超时采用协议给出的超时最大值: $64T_{sdclk}$ 。

33.3.6 状态指示寄存器 (R32_EMMC_STATUS)

偏移地址: 0x20

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

位	名称	访问	描述	复位值
[32:18]	Reserved	R0	保留。	0
17	RB_EMMC_DATA_TOSTA	R0	D0 数据线当前电平状态: 1: 高电平; 0: 低电平。	0
16	RB_EMMC_CMD_DSTA	R0	CMD 信号线当前电平状态: 1: 高电平; 0: 低电平。	0
[15:0]	MASK_BLOCK_NUM	R0	当前多块传输操作中, 已经成功传输的块数。	0

33.3.7 中断标志寄存器 (R16_EMMC_INT_FG)

偏移地址: 0x24

位	名称	访问	描述	复位值
[15:11]	Reserved	R0	保留。	0
10	RB_SIF_SLV_BUF_RELEASE	RW1Z	从机双缓冲模式下, BUF 释放标志位, 写 1 清零: 1: 软件可以更新掉 (写入/读取) 对应 buf 的数据; 0: 对应 buf 数据仍在使用中。	0
9	RB_EMMC_IF_SDIOINT	RW1Z	SDIO 卡中断标志, 写 1 清零: 1: SDIO 卡产生卡中断; 0: 无事件。	0
8	RB_EMMC_IF_FIFO_OV	RW1Z	FIFO 溢出标志, 写 1 清零: 1: FIFO 溢出触发; 0: 无事件。	0
7	RB_EMMC_IF_BKGAP	RW1Z	单块传输完成标志, 写 1 清零: 1: 单块收发完成触发; 0: 无事件。	0

6	RB_EMMC_IF_TRANDONE	RW1Z	所有块数传输完成标志, 写 1 清零: 1: 请求块数都传输完成触发; 0: 无事件。	0
5	RB_EMMC_IF_TRANERR	RW1Z	数据传输 CRC 错误标志, 写 1 清零: 1: CRC 错误触发; 0: 无事件。	0
4	RB_EMMC_IF_DATTMO	RW1Z	数据超时标志, 写 1 清零: 1: 数据超时触发; 0: 无事件。	0
3	RB_EMMC_IF_CMDDONE	RW1Z	命令完成标志位/从机接收命令完成标志, 写 1 清零: 1: 主机发送命令, 并且收到应答完毕; 从机接收命令完毕; 0: 无事件。	0
2	RB_EMMC_IF_REIDX_ER	RW1Z	应答索引号校验错误标志, 写 1 清零: 1: 应答索引号校验错误触发; 0: 无事件。	0
1	RB_EMMC_IF_RECRC_WR	RW1Z	应答 CRC 校验错误标志, 写 1 清零: 1: 应答 CRC 校验错误触发; 0: 无事件。	0
0	RB_EMMC_IF_RE_TMOUT	RW1Z	接收应答超时标志, 写 1 清零: 1: 应答超时触发; 0: 无事件。	0

33.3.8 中断使能寄存器 (R16_EMMC_INT_EN)

偏移地址: 0x28

位	名称	访问	描述	复位值
[15:11]	Reserved	RO	保留。	0
10	RB_SIE_SLV_BUF_RELEASE	RW	从机双缓冲模式下, BUF 释放中断使能: 1: 使能相应中断; 0: 禁止相应中断。	0
9	RB_EMMC_IE_SDIOINT	RW	SDIO 卡中断使能: 1: 使能相应中断; 0: 禁止相应中断。	0
8	RB_EMMC_IE_FIFO_OV	RW	FIFO 溢出中断使能: 1: 使能相应中断; 0: 禁止相应中断。	0
7	RB_EMMC_IE_BKGAP	RW	单块传输完成中断使能: 1: 使能相应中断; 0: 禁止相应中断。	0
6	RB_EMMC_IE_TRANDONE	RW	所有块传输完成中断使能: 1: 使能相应中断; 0: 禁止相应中断。	0
5	RB_EMMC_IE_TRANERR	RW	块传输 CRC 错误中断: 1: 使能相应中断;	0

			0: 禁止相应中断。	
4	RB_EMMC_IE_DATTMO	RW	数据超时中断: 1: 使能相应中断; 0: 禁止相应中断。	0
3	RB_EMMC_IE_CMDDONE	RW	命令完成中断: 1: 使能相应中断; 0: 禁止相应中断。	0
2	RB_EMMC_IE_REIDX_ER	RW	应答索引校验错误中断: 1: 使能相应中断; 0: 禁止相应中断。	0
1	RB_EMMC_IE_RECRC_WR	RW	应答 CRC 校验错误中断: 1: 使能相应中断; 0: 禁止相应中断。	0
0	RB_EMMC_IE_RE_TMOUT	RW	命令应答超时中断: 1: 使能相应中断; 0: 禁止相应中断。	0

33.3.9 数据块 DMA 起始地址寄存器 (R32_EMMC_DMA_BEG1)

偏移地址: 0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB_EMMC_DMAAD1_MASK[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB_EMMC_DMAAD1_MASK[15:0]															

位	名称	访问	描述	复位值
[31:0]	RB_EMMC_DMAAD1_MASK [31:0]	RW	读写数据缓存区起始地址, 低 4 位固定为 0 (16 字节对齐)。该寄存器设置完成后, 则正式启动接收或发送。	0

注: 此寄存器在从 SD 中读数据时, 保存着读到的数据在 SRAM 中的起始地址。在写向 SD 卡写数据时, 保存着将要写的数据在 SRAM 中的起始地址。寻址在 SRAM 区域。

如果执行连续的多块读写 SD 操作, 那么在单块传输完成后 (RB_EMMC_IF_BKGAP), 用户根据需要可以写 DMA_BEG1 寄存器来更改 DMA 地址。不可在传输过程中更改 DMA 地址, 否则可能引起数据计数错误。在进行连续多块写时, 单块传输完成后, 需要写 R32_EMMC_WRITE_CONT 或 DMA_BEG1 寄存器的方式启动继续写操作。多块读不需要。

33.3.10 传输块配置寄存器 (R32_EMMC_BLOCK_CFG)

偏移地址: 0x30

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

位	名称	访问	描述	复位值

[31:28]	Reserved	R0	保留。	0
[27:16]	RB_EMMC_BKSIZE_MASK	RW	单块传输大小（1~2048 字节）。	0
[15:0]	RB_EMMC_BKNUM_MASK	RW	本次 DMA 要传输的块计数（1~65535 块），内部自动清零，块数不为零则产生接收或发送信号。	0

33.3.11 传输模式寄存器（R32_EMMC_TRAN_MODE）

偏移地址：0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											RB_SW	RB_CARE_NEG	RB_DDR_MODE	RB_EMMC_DULEDMA_EN	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RB_EMMC_DMATN_CNT							Reserved	RB_EMMC_AUTOGAPSTOP	Reserved	RB_EMMC_MODE_BOOT	RB_EMMC_GAP_STOP	RB_EMMC_MC_DMA_DIR		

位	名称	访问	描述	复位值
[31:21]	Reserved	R0	保留。	0
[20:19]	RB_SW	RW	DDR模式下，时钟切换： 00：时钟不切换； 01：自动切换； 10：手动切换； 11：保留。	0
18	RB_CARE_NEG	RW	在 DDR 模式下，对时钟下降沿数据检查： 1：检查； 0：不检查。	0
17	RB_DDR_MODE	RW	使能 DDR 模式： 1：使能； 0：关闭。	0
16	RB_EMMC_DULEDMA_EN	RW	使能 DMA 双缓冲功能： 1：使能； 0：关闭。	0
15	Reserved	R0	保留。	0
[14:8]	RB_EMMC_DMATN_CNT	RW	双缓冲模式下，设置缓冲区切换的块计数值。	0x10
[7:5]	Reserved	RW	保留，Bit5 必须写 0。	010b
4	RB_EMMC_AUTOGAPSTOP	RW	硬件自动数据单块停时钟使能： 1：使能； 0：关闭。	0
3	Reserved	RW	必须写 0。	0
2	RB_EMMC_MODE_BOOT	RW	设置eMMC卡传输模式： 1：引导模式； 0：正常模式。 注：仅用于eMMC卡。	0
1	RB_EMMC_GAP_STOP	RW	数据块完成时钟停止模式。读块数据时，将此位置 1，则计数一个数据块的采样后硬件自动关闭时钟输出，开启新的块传输，需要软件清 0，延	0

			时 1 个 SCLK 时间后, 重新置 1 再次计数。 如果使能了 RB_EMMC_AUTOGAPSTOP 功能, 此位硬件置 1, 软件只需清 0。	
0	RB_EMMC_DMA_DIR	RW	DMA 传输的方向: 1: 控制器到 SD; 0: SD 到控制器。	0

33.3.12 时钟配置寄存器 (R16_EMMC_CLK_DIV)

偏移地址: 0x38

位	名称	访问	描述	复位值
[15:11]	Reserved	RO	保留。	0
10	RB_EMMC_PHASE_INV	RW	SDCK 时钟输出相位反转。	0
9	RB_EMMC_CLKMode	RW	时钟频率模式选择位: 1: 高速模式, 25M-200MHz; 0: 低速模式, 5MHz 以下。	1
8	RB_EMMC_CLKOE	RW	SD 物理时钟信号线输出控制位: 1: 打开, 输出通讯时钟; 0: 关闭。	0
[7:5]	Reserved	RO	保留。	0
[4:0]	RB_EMMC_DIV_MASK	RW	SD 控制器时钟 (SDCLK) 分频因子: 当 RB_EMMC_CLKMode = 1, 则 SDCLK = SYSPLL_SEL/RB_EMMC_DIV_MASK; 当 RB_EMMC_CLKMode = 0, 则 SDCLK = SYSPLL_SEL/RB_EMMC_DIV_MASK/64。 最小值为 2, 写 1 等效关闭 SDC 模时钟。	10011b

注: 从模式时 RB_EMMC_DIV_MASK 需配置成最小值 2, 该值不影响输入时钟值。

33.3.13 数据块 DMA 起始地址寄存器 (R32_EMMC_DMA_BEG2)

偏移地址: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RB_EMMC_DMAAD2_MASK[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB_EMMC_DMAAD2_MASK[15:0]															

位	名称	访问	描述	复位值
[31:0]	RB_EMMC_DMAAD2_MASK [31:0]	RW	读写数据缓存区起始地址, 低 4 位固定为 0 (16 字节对齐)。	0

注: 此寄存器在从 SD 中读数据时, 保存着读到的数据在 SRAM 中的起始地址。在写向 SD 卡写数据时, 保存着将要写的数据在 SRAM 中的起始地址。寻址在 SRAM 区域。

33.3.14 数据输出延迟寄存器 (R32_EMMC_TUNE_DAT0)

偏移地址: 0x40

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

RB_TUNNE_DAT7_0[3:0]	RB_TUNNE_DAT6_0[3:0]	RB_TUNNE_DAT5_0[3:0]	RB_TUNNE_DAT4_0[3:0]
15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
RB_TUNNE_DAT3_0[3:0]	RB_TUNNE_DAT2_0[3:0]	RB_TUNNE_DAT1_0[3:0]	RB_TUNNE_DAT0_0[3:0]

位	名称	访问	描述	复位值
[31:28]	RB_TUNNE_DAT7_0 [3:0]	RW	D7 输出的延迟。	0
[27:24]	RB_TUNNE_DAT6_0 [3:0]	RW	D6 输出的延迟。	0
[23:20]	RB_TUNNE_DAT5_0 [3:0]	RW	D5 输出的延迟。	0
[19:16]	RB_TUNNE_DAT4_0 [3:0]	RW	D4 输出的延迟。	0
[15:12]	RB_TUNNE_DAT3_0 [3:0]	RW	D3 输出的延迟。	0
[11:8]	RB_TUNNE_DAT2_0 [3:0]	RW	D2 输出的延迟。	0
[7:4]	RB_TUNNE_DAT1_0 [3:0]	RW	D1 输出的延迟。	0
[3:0]	RB_TUNNE_DAT0_0 [3:0]	RW	D0 输出的延迟。	0

注：输出延迟=0.1ns * RB_TUNNE_DATx_0。

33.3.15 数据输入延迟寄存器 (R32_EMMC_TUNE_DATI)

偏移地址：0x44

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16
RB_TUNNE_DAT7_I[3:0]	RB_TUNNE_DAT6_I[3:0]	RB_TUNNE_DAT5_I[3:0]	RB_TUNNE_DAT4_I[3:0]
15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
RB_TUNNE_DAT3_I[3:0]	RB_TUNNE_DAT2_I[3:0]	RB_TUNNE_DAT1_I[3:0]	RB_TUNNE_DAT0_I[3:0]

位	名称	访问	描述	复位值
[31:28]	RB_TUNNE_DAT7_I [3:0]	RW	D7 输入的延迟。	0
[27:24]	RB_TUNNE_DAT6_I [3:0]	RW	D6 输入的延迟。	0
[23:20]	RB_TUNNE_DAT5_I [3:0]	RW	D5 输入的延迟。	0
[19:16]	RB_TUNNE_DAT4_I [3:0]	RW	D4 输入的延迟。	0
[15:12]	RB_TUNNE_DAT3_I [3:0]	RW	D3 输入的延迟。	0
[11:8]	RB_TUNNE_DAT2_I [3:0]	RW	D2 输入的延迟。	0

[7:4]	RB_TUNNE_DAT1_I [3:0]	RW	D1 输入的延迟。	0
[3:0]	RB_TUNNE_DAT0_I [3:0]	RW	D0 输入的延迟。	0

注：输入延迟=0.1ns * RB_TUNNE_DATx_I。

33.3.16 时钟和命令延迟寄存器（R32_EMMC_TUNE_CLK_CMD）

偏移地址：0x48

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								RB_TUNNE_CMD_I[3:0]				RB_TUNNE_CMD_0[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RB_TUNNE_CLK_I[3:0]				RB_TUNNE_CLK_0[3:0]			

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:20]	RB_TUNNE_CMD_I [3:0]	RW	CMD 输入的延迟： 延迟值 = 0.2ns * RB_TUNNE_CMD_I。	0
[19:16]	RB_TUNNE_CMD_0 [3:0]	RW	CMD 输出的延迟： 延迟值 = 0.2ns * RB_TUNNE_CMD_0。	0
[15:8]	Reserved	RO	保留。	0
[7:4]	RB_TUNNE_CLK_I [3:0]	RW	CLK 输入的延迟： 延迟值 = 0.2ns * RB_TUNNE_CLK_I。 注：该时钟输入的延迟只延迟 emmc 卡的 strobe 信号，在使用没有此信号的卡时，该字段无效。	0
[3:0]	RB_TUNNE_CLK_0 [3:0]	RW	CLK 输出的延迟： 延迟 = 0.2ns * RB_TUNNE_CLK_0。	0

第 34 章 可编程协议 I/O 微控制器 (PIOC)

芯片内嵌了一个可编程协议 I/O 微控制器，该微控制器基于单时钟周期的专用精简指令集 RISC 内核，运行于系统主频，具有 2K 指令的程序 ROM 和 49 个 SFR 寄存器及 PWM 定时/计数器，支持 2 个 I/O 引脚的协议控制。

34.1 主要特征

- RISC 内核，优化的单周期位操作指令集，全静态指令集
- 复用了 4K 字节的系统 SRAM 作为 2K 字容量的程序 ROM，支持程序暂停和动态加载
- 提供 33 字节的双向和单向各 1 个寄存器，提供 6 级独立堆栈
- 支持 2 个通用双向 I/O 协议控制，支持输入电平变化检测
- 通过动态加载不同的协议程序，可以支持多种协议规格的单线接口和两线接口
- 支持单线 ARGB 芯片 1024 级串联

34.2 指令和程序空间及堆栈

34.2.1 指令集

eMCU 采用精简指令集 RISC8B 内核，数据宽度为 8 位，指令宽度为 16 位。

共 66 条指令，除了跳转指令是双时钟周期、程序空间读写指令是双周期之外，其余指令都是单时钟周期。更多介绍请参考 RISC8B 内核指令集和汇编工具文档 CHRISC8B.PDF。

34.2.2 程序 ROM

eMCU 的程序 ROM 来自 4K 字节的系统 SRAM 复用，当 SB_MST_CLK_GATE 为 0 时专用于主机侧的系统 SRAM，当 SB_MST_CLK_GATE 为 1 时专用于 eMCU 的程序 ROM。

eMCU 的程序空间是 2048 字，程序空间地址从 0x0000 到 0x07FF。程序 ROM 中 0 地址的指令具有特殊用途，该指令不会被执行。

eMCU 芯片支持对程序空间进行读操作。步骤是：先将目标地址低 8 位写入 SFR_INDIR_ADDR 中，目标地址高 3 位写入 A 寄存器中，然后执行 RDCODE 指令，可以一次读出程序空间中的 16 位数据，分为高低两个字节，其中低字节返回在 A 寄存器中，高字节返回在 SFR_INDIR_ADDR 中。该操作可以用于双字节查表。

eMCU 芯片不支持对程序空间进行写操作。

34.2.3 堆栈

eMCU 具有 6 级深度的堆栈存储器，数据宽度是 11 位。用于在子程序调用时保存程序返回地址，也可以用于 PUSHAS 指令将程序运行中的变量和状态等数据保存到堆栈中

34.2.4 睡眠与唤醒

当 SB_MST_CLK_GATE 为 0 时，eMCU 程序将暂停运行。当执行 SLEEP 或 SLEEPX 指令后，eMCU 将进入睡眠状态。eMCU 为全静态设计，程序暂停与睡眠具有等同效果。

eMCU 睡眠后支持两种唤醒方式：开启电平变化检测的引脚的电平变化唤醒，SB_MST_CLK_GATE 从 0 到 1 的变化唤醒。

34.2.5 事件等待指令

WAITB 指令支持 8 种事件等待，当事件无效时一直持续执行 WAITB 等待，程序计数器 PC 保持不变；当检测到目标事件后退出等待，执行下一条指令。等待过程中可选开启定时器超时复位。

指令参数 WB_DATA_SW_MR_0 表示等待直到 SB_DATA_SW_MR 为 0 时退出，如已是 0 直接退出；

指令参数 WB_BIT_CYC_TAIL_1 表示等待直到 SB_BIT_CYC_TAIL 为 1 时退出，如已是 1 直接退出；

指令参数 WB_PORT_I0_FALL 表示等待直到 SB_PORT_IN0 检测到下降沿退出，如之前记录了已检测到也直接退出，退出时清除下降沿检测记录；

指令参数 WB_PORT_I0_RISE 表示等待直到 SB_PORT_IN0 检测到上升沿退出，如之前记录了已检测到也直接退出，退出时清除上升沿检测记录；

指令参数 WB_DATA_MW_SR_1 表示等待直到 SB_DATA_MW_SR 为 1 时退出，如已是 1 直接退出；

指令参数 WB_PORT_XOR1_1 表示等待直到 SB_PORT_XOR1 为 1 时退出，如已是 1 直接退出；

指令参数 WB_PORT_XOR0_0 表示等待直到 SB_PORT_XOR0 为 0 时退出，如已是 0 直接退出；

指令参数 WB_PORT_XOR0_1 表示等待直到 SB_PORT_XOR0 为 1 时退出，如已是 1 直接退出。

34.2.6 位传送指令

位传送指令支持 2 个操作寄存器：BP1F 和 BG1F 对应寄存器 SFR_INDIR_ADDR，BP2F 和 BG2F 对应寄存器 SFR_DATA_EXCH。

位传送指令 BP1F 和 BP2F 支持 4 个独立位输出：0#参数 BIO_FLAG_C 对应 SB_FLAG_C 位，1#参数 B0_BIT_TX_00 对应 SB_BIT_TX_00，2#参数 B0_PORT_OUT0 对应 SB_PORT_OUT0，3#参数 B0_PORT_OUT1 对应 SB_PORT_OUT1。

位传送指令 BG1F 和 BG2F 支持 4 个独立位输入：0#参数 BIO_FLAG_C 对应 SB_FLAG_C 位，1#参数 BI_BIT_RX_I0 对应 SB_BIT_RX_I0，2#参数 BI_PORT_IN0 对应 SB_PORT_IN0，3#参数 BI_PORT_IN1 对应 SB_PORT_IN1。

位传送指令 BCTC 支持 4 个独立位输入：0#参数 BI_C_XOR_IN0 则位 SB_FLAG_C 更新为 SB_FLAG_C 与 SB_PORT_IN0 的异或结果，1#参数 BI_BIT_RX_I0 对应 SB_BIT_RX_I0，2#参数 BI_PORT_IN0 对应 SB_PORT_IN0，3#参数 BI_PORT_IN1 对应 SB_PORT_IN1。

34.3 数据空间和寄存器及寻址

eMCU 的数据空间包括 49 个专用功能寄存器 SFR，通过 8 位地址进行寻址，范围是 0x00-0x3F。所有寄存器的数据宽度都是 1 个字节，也就是 8 位数据。

部分 SFR 在主机侧可以读写或者只读，主机侧支持 8 位、16 位或者 32 位宽度进行读写。主机侧具有写优先权，当 eMCU 与主机同时写入同一寄存器时，eMCU 写操作被自动丢弃。

eMCU 的寻址方式包括：立即数寻址、立即数快速寻址、普通直接寻址、扩展直接寻址、间接寻址、位寻址，后 4 种用于寻址寄存器。

间接寻址的寻址范围是 0x000-0x0FF，覆盖 eMCU 的所有寄存器。eMCU 共提供了两组间接寻址寄存器，每组间接寻址寄存器由一个地址寄存器和一个数据读写端口组成。先向地址寄存器写入目的寄存器的地址，再通过读写数据读写端口就可以读写目的寄存器。

扩展直接寻址的寻址范围是 0x000-0x1FF，由指令直接提供 9 位寄存器地址，仅适用于以下两条直接读写寄存器的指令：MOV register, A 或 F 指令，MOVA register 指令。

普通直接寻址的寻址范围是 0x000-0x0FF，由指令直接提供 8 位寄存器地址，适用于除上述扩展直接寻址指令之外的所有直接地址指令，例如 CLR register、ADD register、BS register, bit 等。需要寻址更大范围时应该采用间接寻址方式。

立即数快速寻址用于无需通过 A 中转而将指令码中的操作数快速写入目标寄存器。MOVA1F 用于快速设置 1#寄存器 SFR_PORT_DIR，MOVA2F 用于快速设置 2#寄存器 SFR_PORT_I0。MOVA1P 和 MOVA2P 通过间接寻址适用于所有寄存器的快速设置，相当于单周期内先立即数寻址再间接寻址。

位寻址由指令直接提供 3 位位地址，而对寄存器的寻址可分别采用普通直接寻址或者间接寻址，分别实现 0x000-0x0FF 的寻址范围，从而可以寻址任何一个寄存器的任何一位。

34.4 专用功能寄存器 SFR

部分 SFR 或部分位未真正实现，是保留位，读出是 0，写入必须保持原值或者写 0。

名称	访问地址	描述	主机侧读写	复位值
R8_SFR_INDIR_PORT	0x40025C00	间接寻址的数据读写端口	UUUUUUUU	XXXXXXXX
R8_SFR_INDIR_PORT2	0x40025C01	间接寻址 2 的数据读写端口	UUUUUUUU	XXXXXXXX
R8_SFR_PRG_COUNT	0x40025C02	程序计数器 PC 的低字节	UUUUUUUU	00000000
R8_SFR_STATUS_REG	0x40025C03	状态寄存器	UUUUSUSU	0000-0-0
R8_SFR_INDIR_ADDR	0x40025C04	间接寻址的地址寄存器	RRRRRRRR	XXXXXXXX
R8_SFR_TMRO_COUNT	0x40025C05	定时器 0 的计数寄存器	RRSSRRRR	00000000
R8_SFR_TIMER_CTRL	0x40025C06	定时器的控制寄存器	RRRRRRRR	00000000
R8_SFR_TMRO_INIT	0x40025C07	定时器 0 的初值寄存器	RRRRRRRR	00000000
R8_SFR_BIT_CYCLE	0x40025C08	编码位周期寄存器	WWWWWWW	00000000
R8_SFR_INDIR_ADDR2	0x40025C09	间接寻址 2 的地址寄存器	RRRRRRRR	00000000
R8_SFR_PORT_DIR	0x40025C0A	端口方向设置寄存器	RRRRRRRR	00000000
R8_SFR_PORT_IO	0x40025C0B	端口输入输出寄存器	RRRRRRRR	XXXXXX00
R8_SFR_BIT_CONFIG	0x40025C0C	编码位配置寄存器	WWRRRRRR	00010000
R8_SFR_SYS_CFG	0x40025C1C	系统配置寄存器	RRRWWWWW	00000000
R8_SFR_CTRL_RD	0x40025C1D	eMCU 读写且主机只读寄存器	RRRRRRRR	00000000
R8_SFR_CTRL_WR	0x40025C1E	主机读写且 eMCU 只读寄存器	WWWWWWW	00000000
R8_SFR_DATA_EXCH	0x40025C1F	数据交换寄存器	WWWWWWW	00000000
R8_SFR_DATA_REG0	0x40025C20	数据寄存器 0	WWWWWWW	00000000
R8_SFR_DATA_REG1	0x40025C21	数据寄存器 1	WWWWWWW	00000000
R8_SFR_DATA_REG2	0x40025C22	数据寄存器 2	WWWWWWW	00000000
R8_SFR_DATA_REG3	0x40025C23	数据寄存器 3	WWWWWWW	00000000
R8_SFR_DATA_REG4	0x40025C24	数据寄存器 4	WWWWWWW	00000000
R8_SFR_DATA_REG5	0x40025C25	数据寄存器 5	WWWWWWW	00000000
R8_SFR_DATA_REG6	0x40025C26	数据寄存器 6	WWWWWWW	00000000
R8_SFR_DATA_REG7	0x40025C27	数据寄存器 7	WWWWWWW	00000000
R8_SFR_DATA_REG8	0x40025C28	数据寄存器 8	WWWWWWW	00000000
R8_SFR_DATA_REG9	0x40025C29	数据寄存器 9	WWWWWWW	00000000
R8_SFR_DATA_REG10	0x40025C2A	数据寄存器 10	WWWWWWW	00000000
R8_SFR_DATA_REG11	0x40025C2B	数据寄存器 11	WWWWWWW	00000000
R8_SFR_DATA_REG12	0x40025C2C	数据寄存器 12	WWWWWWW	00000000
R8_SFR_DATA_REG13	0x40025C2D	数据寄存器 13	WWWWWWW	00000000
R8_SFR_DATA_REG14	0x40025C2E	数据寄存器 14	WWWWWWW	00000000
R8_SFR_DATA_REG15	0x40025C2F	数据寄存器 15	WWWWWWW	00000000
R8_SFR_DATA_REG16	0x40025C30	数据寄存器 16	WWWWWWW	00000000
R8_SFR_DATA_REG17	0x40025C31	数据寄存器 17	WWWWWWW	00000000
R8_SFR_DATA_REG18	0x40025C32	数据寄存器 18	WWWWWWW	00000000
R8_SFR_DATA_REG19	0x40025C33	数据寄存器 19	WWWWWWW	00000000
R8_SFR_DATA_REG20	0x40025C34	数据寄存器 20	WWWWWWW	00000000

R8_SFR_DATA_REG21	0x40025C35	数据寄存器 21	WWWWWWWW	00000000
R8_SFR_DATA_REG22	0x40025C36	数据寄存器 22	WWWWWWWW	00000000
R8_SFR_DATA_REG23	0x40025C37	数据寄存器 23	WWWWWWWW	00000000
R8_SFR_DATA_REG24	0x40025C38	数据寄存器 24	WWWWWWWW	00000000
R8_SFR_DATA_REG25	0x40025C39	数据寄存器 25	WWWWWWWW	00000000
R8_SFR_DATA_REG26	0x40025C3A	数据寄存器 26	WWWWWWWW	00000000
R8_SFR_DATA_REG27	0x40025C3B	数据寄存器 27	WWWWWWWW	00000000
R8_SFR_DATA_REG28	0x40025C3C	数据寄存器 28	WWWWWWWW	00000000
R8_SFR_DATA_REG29	0x40025C3D	数据寄存器 29	WWWWWWWW	00000000
R8_SFR_DATA_REG30	0x40025C3E	数据寄存器 30	WWWWWWWW	00000000
R8_SFR_DATA_REG31	0x40025C3F	数据寄存器 31	WWWWWWWW	00000000

复位后的默认值都是以二进制数表示，位值说明如下：

0：复位后总是 0；

1：复位后总是 1；

X：复位不影响数据，数据初值不确定；

-：上电复位后总是清 0，系统复位或主机强制复位不影响数据。

主机侧读写的说明如下：

W：可以读或者写的位；

R：只读位；

U：不可见的位，不可读写；

S：只读的交换位，主机侧可见的 SFR_TIMER_CTRL 与 eMCU 侧部分不同，其中 SB_TMRO_ENABLE 和 SB_TMRO_OUT_EN 两个位分别被 SB_GP_BIT_Y 和 SB_GP_BIT_X 代替。

34.4.1 间接寻址的数据读写端口（SFR_INDIR_PORT）

34.4.2 间接寻址的地址寄存器（SFR_INDIR_ADDR）

34.4.3 间接寻址 2 的数据读写端口（SFR_INDIR_PORT2）

34.4.4 间接寻址 2 的地址寄存器（SFR_INDIR_ADDR2）

这是两组间接寻址寄存器，每组间接寻址寄存器由一个地址寄存器和一个数据读写端口组成。先向地址寄存器写入目的寄存器的地址，再通过读写数据读写端口就可以读写目的寄存器。

读写间接寻址的数据读写端口 SFR_INDIR_PORT，就是读写由间接寻址的地址寄存器 SFR_INDIR_ADDR 指定地址的目的寄存器。读写间接寻址的数据读写端口 SFR_INDIR_PORT2，就是读写由间接寻址的地址寄存器 SFR_INDIR_ADDR2 指定地址的目的寄存器。

专用指令 MOVIP 用于向 SFR_INDIR_ADDR 加载 8 位目标地址，并且 SFR_INDIR_ADDR 也可以被其它非专用指令独立修改。

专用指令 MOVIA 用于向 SFR_INDIR_ADDR2 加载 8 位目标地址，并且 SFR_INDIR_ADDR2 也可以被其它非专用指令独立修改。

间接寻址 2 的地址寄存器 SFR_INDIR_ADDR2 具有自动增量特性，当执行“MOV SFR_INDIR_PORT2, A”读指令或者对 RAM 执行“MOVA SFR_INDIR_PORT2”写指令时，SFR_INDIR_ADDR2 将在指令执行完成后自动加 1。

专用指令 MOVA1P 和 MOVA2P 用于直接将指令码中的操作数快速写入目标寄存器，相当于单周期内先立即数寻址再间接寻址。

34.4.5 状态寄存器 (SFR_STATUS_REG)

状态寄存器包含 ALU 的结果状态、通用的位变量及堆栈使用标志等。当执行算术或者逻辑操作后，eMCU 会根据结果设置相应的状态位，以便程序判断后再做出进一步处理。通用位变量可以由应用程序定义其用途。

定时器超时复位用于监控 WAITB 等待指令的超时，关闭 SB_TMR0_ENABLE 并启用 SB_EN_TOUT_RST 后，在 WAITB 指令执行期间，定时器进行计数，溢出时 SB_TMR0_CYCLE 为 1 则导致超时复位。每次当 WAITB 指令因满足条件而退出 WAITB 等待时自动重新加载定时器初值。在 SB_TMR0_ENABLE 为 0 情况下的非 WAITB 指令的执行期间不计数，即定时器超时复位仅针对 WAITB 指令。

位	名称	描述	复位值
7	Reserved	保留。	0
6	Reserved	保留。	0
5	SB_STACK_USED	当前堆栈使用标志： 1：已压栈； 0：尚未使用或已完全出栈。	0
4	SB_EN_TOUT_RST	定时器超时复位使能： 1：允许在定时器超时溢出时复位 eMCU； 0：禁用。	0
3	SB_GP_BIT_Y	通用的位变量 Y，由应用程序定义和使用， 上电复位后为 0，不受系统复位或主机强制复位的影响	—
2	SB_FLAG_Z	ALU 零标志，结果是否为 00H： 1：结果是 0； 0：结果不是 0。	0
1	SB_GP_BIT_X	通用的位变量 X，由应用程序定义和使用， 上电复位后为 0，不受系统复位或主机强制复位的影响	—
0	SB_FLAG_C	ALU 进位标志，结果是否进位或者由移位产生： 1：结果产生进位； 0：结果没有进位。	0

34.4.6 程序计数器 PC 的低字节 (SFR_PRG_COUNT)

eMCU 的程序计数器 PC 的宽度是 11 位，通过读写 SFR_PRG_COUNT 可以修改 PC 的低 8 位，但是不会影响 PC 的高 3 位，从而可以在 256 字节范围内进行短跳转和查表。

34.4.7 端口方向设置寄存器 (SFR_PORT_DIR)

下表是端口方向设置寄存器 SFR_PORT_DIR 的说明。

位	名称	描述	复位值
7	SB_PORT_MOD3	引脚模式控制，主机侧定义用途。	0
6	SB_PORT_MOD2		0
5	SB_PORT_MOD1		0
4	SB_PORT_MOD0		0
3	SB_PORT_PU1	I01 引脚上拉使能： 1：允许； 0：禁用。	0
2	SB_PORT_PU0	I00 引脚上拉使能： 1：允许； 0：禁用。	0

1	SB_PORT_DIR1	I01 引脚方向控制： 1：输出； 0：输入。	0
0	SB_PORT_DIR0	I00 引脚方向控制： 1：输出； 0：输入。	0

34.4.8 端口输入输出寄存器 (SFR_PORT_IO)

端口包含 2 个双向输入输出引脚，可以由各自的方向设置位 SB_PORT_DIR 独立设定为输入方向或输出方向，如果是输出方向，则输出 SB_PORT_OUT 中的数据到引脚。复位后的默认方向是输入，SB_PORT_IN 用于获取当前引脚输入电平（当 SB_PORT_IN_EDGE=0 时）或者前一个周期的引脚输入电平（当 SB_PORT_IN_EDGE=1 时），后者比前者提前约半个时钟周期。

2 个引脚均具有电平变化检测功能，当输入电平与输出数据寄存器 SB_PORT_OUT 中的数据不一致时，对应的 SB_PORT_XOR 变为 1。

I00 引脚的可用输入指令和方法更多，相比 I01 更适宜用于输入或双向数据。

下表是端口输入输出寄存器 SFR_PORT_IO 的说明。

位	名称	描述	复位值
7	SB_PORT_IN_XOR	I01 引脚输入状态与 I00 引脚输入状态的异或结果： 1：I01 与 I00 电平不同； 0：I01 与 I00 电平相同。	X
6	SB_BIT_RX_I0	I00 引脚的位接收状态解码后的数据。	X
5	SB_PORT_IN1	I01 引脚输入状态（参考 SB_PORT_IN_EDGE）。	X
4	SB_PORT_IN0	I00 引脚输入状态（参考 SB_PORT_IN_EDGE）。	X
3	SB_PORT_XOR1	I01 引脚输入状态与 SB_PORT_OUT1 的异或结果： 1：I01 电平与 SB_PORT_OUT1 不同，用于检测电平变化； 0：I01 电平 SB_PORT_IN1 与 SB_PORT_OUT1 相同。	X
2	SB_PORT_XOR0	I00 引脚输入状态与 SB_PORT_OUT0 的异或结果： 1：I00 电平与 SB_PORT_OUT0 不同，用于检测电平变化； 0：I00 电平 SB_PORT_IN0 与 SB_PORT_OUT0 相同。	X
1	SB_PORT_OUT1	I01 引脚输出数据： 1：输出高； 0：输出低。	0
0	SB_PORT_OUT0	I00 引脚输出数据： 1：输出高； 0：输出低。 定时器输出信号优先，当 SB_TMRO_OUT_EN 为 1 时，输出 SB_TMRO_CYCLE 或者其二分频信号。	0

34.4.9 定时器 0 的计数寄存器 (SFR_TMRO_COUNT)

34.4.10 定时器 0 的初值寄存器 (SFR_TMRO_INIT)

34.4.11 定时器的控制寄存器 (SFR_TIMER_CTRL)

定时器 0 是 8 位宽度的计数器。计数值在寄存器 SFR_TMRO_COUNT 中，SFR_TMRO_COUNT 支持直接

写入数据以加载新值。定时器 0 的时钟频率由 SB_TMRO_FREQ 进行选择。定时器 0 可以工作于定时器模式或脉宽调制器模式。

定时器 0 在定时器模式下，计数初值事先保存在初值寄存器 SFR_TMRO_INIT 中，定时器 0 的计数值总是从计数初值计数到 0FFH，当从 0FFH 计数溢出时，定时器 0 自动从初值寄存器 SFR_TMRO_INIT 中重新加载初值，然后再从计数初值开始向 0FFH 计数。当计数值从 0FEH 计数到 0FFH 时，定时器 0 自动将计数周期信号 SB_TMRO_CYCLE 设置为 1；当计数值从 0FFH 计数溢出重新回到计数初值时，定时器 0 自动将计数周期信号 SB_TMRO_CYCLE 清除为 0。SB_TMRO_CYCLE 的频率是 (SB_TMRO_FREQ 时钟频率) / (256 - SFR_TMRO_INIT)，其占空比是 1 / (256 - SFR_TMRO_INIT)，也就是每个周期中有 1 个时钟期间 SB_TMRO_CYCLE 为 1。

定时器 0 在脉宽调制器模式下，定时器 0 总是从 00H 计数到 0FFH，再从 0FFH 计数溢出，重新回到 00H 开始新的计数，其周期总是 256 个时钟。当计数值大于定时器 0 的初值寄存器 SFR_TMRO_INIT 中的数值时，定时器 0 自动将计数周期信号 SB_TMRO_CYCLE 设置为 1；当计数值从 0FFH 计数溢出回到 00H 时，定时器 0 自动将计数周期信号 SB_TMRO_CYCLE 清除为 0。SB_TMRO_CYCLE 的占空比是 (255 - SFR_TMRO_INIT) / 256。

SB_TMRO_OUT_EN 用于设定内部信号 SB_TMRO_CYCLE 或者其二分频信号是否输出到 I00 引脚，内部信号为 1 时引脚输出高电平，否则输出低电平。如果计数周期信号输出允许，那么在定时器模式下，SB_TMRO_CYCLE 的二分频信号（频率降低为一半并且占空比为 50%）将通过 I00 引脚输出；在脉宽调制器模式下，SB_TMRO_CYCLE 信号将通过 I00 引脚输出。

下表是定时器的控制寄存器 SFR_TIMER_CTRL 的说明。

位	名称	描述	复位值
7	SB_EN_LEVEL1	I01 引脚电平变化激活中断标志和唤醒使能： 1：允许激活中断标志和唤醒； 0：不激活、不唤醒。	0
6	SB_EN_LEVEL0	I00 引脚电平变化激活中断标志和唤醒使能： 1：允许激活中断标志和唤醒； 0：不激活、不唤醒。	0
5	SB_TMRO_ENABLE	定时器 0 的计数使能： 1：允许计数； 0：禁止计数。	0
4	SB_TMRO_OUT_EN	定时器 0 的计数周期信号输出允许： 1：允许输出到 I00 引脚； 0：禁止输出。	0
3	SB_TMRO_MODE	定时器 0 的工作模式： 1：脉宽调制器模式； 0：定时器模式。	0
[2:0]	SB_TMRO_FREQ	选择定时器 0 的时钟频率（以系统主频的分频数）： 000：1024X； 001：256X； 010：64X； 011：16X； 100：8X； 101：4X； 110：2X； 111：1X。	000

34.4.12 系统配置寄存器（SFR_SYS_CFG）

34.4.13 主机读写且 eMCU 只读寄存器（SFR_CTRL_WR）

34.4.14 eMCU 读写且主机只读寄存器（SFR_CTRL_RD）

系统配置寄存器提供了主机的一些控制位和双向数据交换的状态位。

SB_INT_REQ 由 eMCU 完全控制，用于向主机请求中断或取消请求，但主机可以通过对只读寄存器 SFR_CTRL_RD 的假写操作实现对 SB_INT_REQ 清 0，主机写 SFR_CTRL_RD 对其中数据无实际影响。

eMCU 写入 SFR_CTRL_RD 后 SB_DATA_SW_MR 自动置 1，主机读取 SFR_CTRL_RD 后 SB_DATA_SW_MR 自动清 0。

主机写入 SFR_CTRL_WR 后 SB_DATA_MW_SR 自动置 1，eMCU 读取 SFR_CTRL_WR 后 SB_DATA_MW_SR 自动清 0。

SB_MST_CLK_GATE 还同时控制 eMCU 程序 ROM 的复用，当 SB_MST_CLK_GATE 为 0 时专用于主机侧的系统 SRAM，共 4K 字节，可以为 eMCU 动态加载新的程序；当 SB_MST_CLK_GATE 为 1 时专用于 eMCU 的程序 ROM，共 2K 指令。

系统配置寄存器的位 0~位 4 由主机完全控制，eMCU 侧为只读。

下表是系统配置寄存器 SFR_SYS_CFG 的说明。

位	名称	描述	复位值
7	SB_INT_REQ	中断请求激活位： 1：向主机请求中断； 0：取消中断请求。	0
6	SB_DATA_SW_MR	SFR_CTRL_RD 等待读取状态位： 1：eMCU 已写入但主机侧尚未读取； 0：eMCU 尚未写入或者主机侧已经读取。	0
5	SB_DATA_MW_SR	SFR_CTRL_WR 等待读取状态位： 1：主机侧已写入但 eMCU 侧尚未读取； 0：主机侧尚未写入或者 eMCU 侧已经读取。	0
4	SB_MST_CFG_B4	配置信息位，软件定义用途	0
3	SB_MST_IO_EN1	I01 引脚的模式和输出控制开关： 1：由 eMCU 控制； 0：由主机控制。	0
2	SB_MST_IO_EN0	I00 引脚的模式和输出控制开关： 1：由 eMCU 控制； 0：由主机控制。	0
1	SB_MST_RESET	强制 eMCU 复位： 1：主机额外强制 eMCU 单独复位； 0：不强制复位，eMCU 仅跟随主机一起复位。	0
0	SB_MST_CLK_GATE	eMCU 的全局时钟控制： 1：开启 eMCU 的时钟； 0：关闭 eMCU 的时钟，其程序暂停，相当于睡眠。	0

34.4.15 编码位周期寄存器 (SFR_BIT_CYCLE)

34.4.16 编码位配置寄存器 (SFR_BIT_CONFIG)

eMCU 支持两种位调制方式：PWM 占空比调制和 Manchester 曼彻斯特调制。

设置 SB_BIT_CYCLE 大于 3，则开启编解码。SB_BIT_TX_EN=0 时接收，根据预设宽度，从 I00 接收输入状态，解码后的数据放在 SB_BIT_RX_I0 中。SB_BIT_TX_EN=1 时发送，在 SB_BIT_TX_I0 中放置将要发送的原始数据，编码后自动送到 SB_PORT_OUT0 供引脚输出。

SB_BIT_CYC_CNT 和 SB_BIT_CYC_TAIL 标识编码位的周期状态，当 SB_BIT_CYC_TAIL=1 时，表示正处于位周期的后 25%，说明 SB_BIT_RX_I0 已完成接收解码，发送编码已完成并达到 SB_PORT_OUT0。

设置 SB_BIT_CYCLE=0 可以强制 SB_BIT_CYC_TAIL 置为 1，便于 Manchester 开启首位编解码。

设置 SFR_BIT_CYCLE 将清除 I/O 的上升沿和下降沿的检测记录，便于 WAITB 指令重新等待。

下表是编码位周期寄存器 SFR_BIT_CYCLE 的说明。

位	名称	描述	复位值
7	SB_BIT_TX_00	将要发送的编码位的原始位数据，该位为双缓冲结构，当新的位周期开始后，此位即可提前加载下一个位数据	0
[6:0]	SB_BIT_CYCLE	设定以时钟周期为单位的编码位的宽度，是位的实际时钟数减 1。其高 5 位为 0 则关闭编解码，非 0 则开启。	0000000

下表是编码位配置寄存器 SFR_BIT_CONFIG 的说明。

位	名称	描述	复位值
7	SB_BIT_TX_EN	编码位的发送使能： 1：开启发送，将 SB_BIT_TX_00 编码后送到 SB_PORT_OUT0； 0：禁止发送，如果 SB_BIT_CYCLE 有效则可以接收。	0
6	SB_BIT_CODE_MOD	编码位的调制方式： 1：Manchester 调制，0 在周期内不翻转，1 在周期内翻转； 0：PWM 占空比调制，0 的占空比 25%，1 的占空比 75%。	0
5	SB_PORT_IN_EDGE	引脚输入电平采样时点选择： 1：在前一个周期的边沿处采样，相当于提前约半个周期； 0：在本时钟周期的中间点采样，相对更实时一点。	0
4	SB_BIT_CYC_TAIL	编码位的周期状态： 1：处于位周期的后 25%，说明位编解码已完成； 0：处于位周期的前 75%。	1
3	SB_BIT_CYC_CNT6	编码位的周期计时状态，位周期计时以时钟周期为单位，是 7 位计数器，SB_BIT_CYC_CNT6~3 是计数器的位 [6:3]。	0
2	SB_BIT_CYC_CNT5		0
1	SB_BIT_CYC_CNT4		0
0	SB_BIT_CYC_CNT3		0

34.4.17 数据寄存器（SFR_DATA_x）（x=0~31）

SFR_DATA_x 均为双侧可读可写的数据寄存器，主机侧写操作优先，软件定义用途。

其中，SFR_DATA_EXCH 还支持单周期的位传送指令。

34.5 应用

eMCU 与主机侧主频相同，便于动态交换数据，其位操作指令丰富，单周期设置 I/O，单周期采集 I/O 状态或复制位数据，可以硬件编解码 PWM 或 Manchester 调制的位数据，适用于各种中低速通讯协议的 I/O 实现，以及需要精确定时的 I/O 控制。

第 35 章 运算放大器（OPA）和比较器（CMP）

该模块包含 3 个可独立配置的运算放大器（OPA1/OPA2/OPA3）和 1 个可独立配置的电压比较器（CMP），其中运算放大器支持低失调输入电压，支持增益选择，也可改用于电压比较器。

运算放大器的输入和输出可连接至 I/O 口，且输入通道（引脚）和增益可选择，输出通道可选择配置到通用 I/O 口，额外有一个内部输出通道直连到比较器输入端，支持高速模式，可通过设置高速模式提高压摆率。

每个电压比较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 TIM 内部采样通道（不占用 I/O 引脚）。

35.1 主要特征

- OPA 输入/输出通道可选择
- OPA1 输出通道直连到比较器输入端
- OPA 支持电压缓冲器（BUFFER）及可编程增益运放（PGA）
- CMP 输入引脚可选择
- CMP 正输入端支持可选的 DAC 输出，负输入端支持可选的内部自偏置电压
- CMP 支持可选数字滤波及迟滞
- CMP 输出引脚可选择通用 I/O 口或 TIM 内部采样通道

35.2 功能描述

35.2.1 运算放大器 1/2/3（OPA1/OPA2/OPA3）

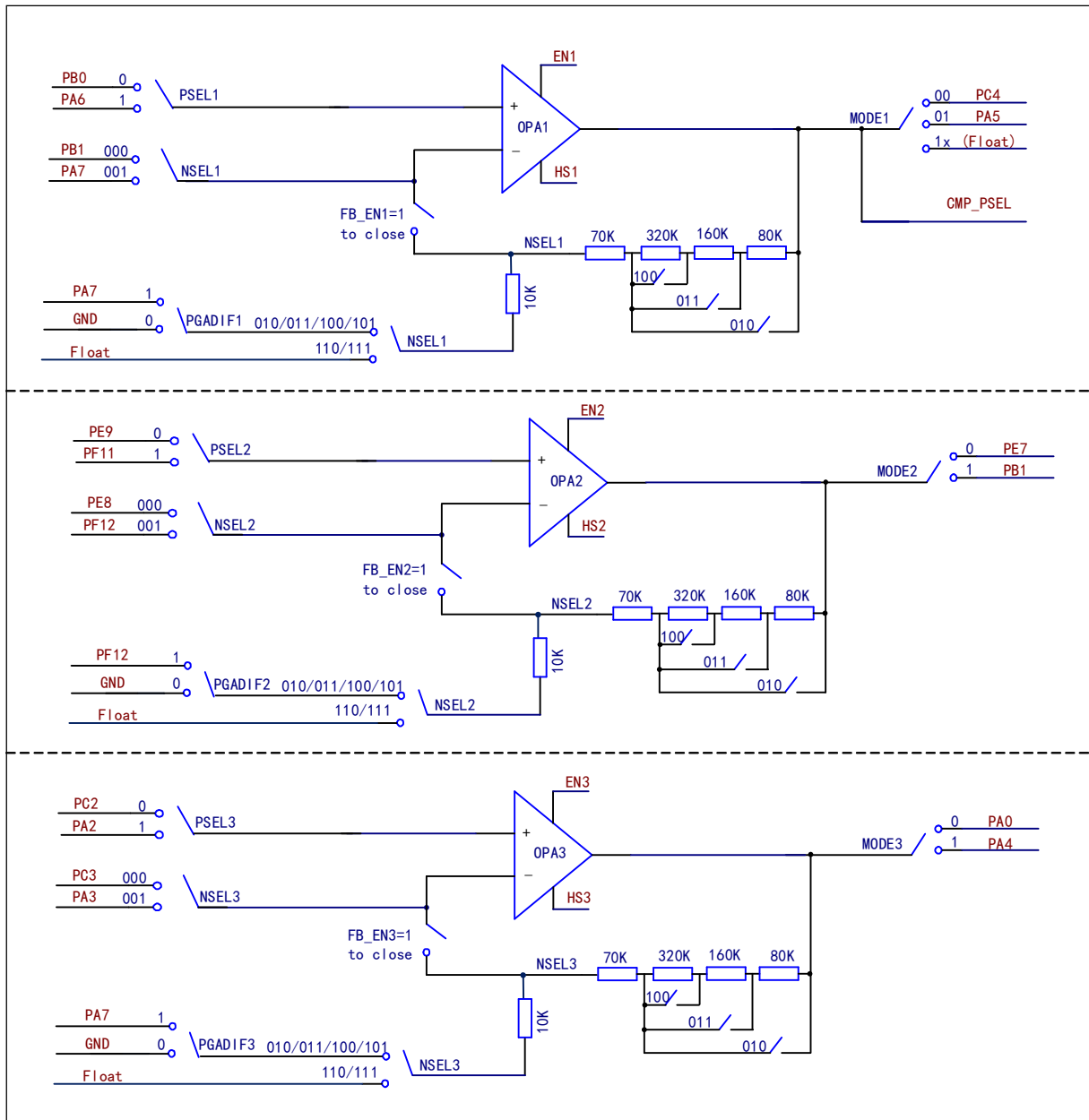
OPA1、OPA2、OPA3 支持电压缓冲器（BUFFER）及可编程增益运放（PGA）。如图 35-1 所示，这三个 OPA 的功能基本一致，其中 OPA1 的输出相较于 OPA2/OPA3，增加了一个输出通道连接至电压比较器 CMP_P2 输入端。

在 R32_OPA_CTLR1 寄存器中：配置 EN1 位，可使能 OPA1；配置 PSEL1 位，可选择 OPA1 正端输入通道；配置 NSEL1 位，可选择 OPA1 负端输入通道或选择 PGA 模式、设置 OPA 内部增益大小；配置 FB_EN1 位，可使能 OPA1 的 PGA 模式反馈；配置 PGADIF1 位，可使能差分输入 PGA 模式，使 OPA1 配合 NSEL1 作为 PGA 使用，且 N 端连接 PA7；配置 HS1 位，可使能高速模式，在此模式下可提高压摆率；配置 MODE1[1:0] 位，可选择输出通道为 GPIO 或比较器正输入端。

在 R32_OPA_CTLR2 寄存器中：配置 EN2 位，可使能 OPA2；配置 PSEL2 位，可选择 OPA2 正端输入通道；配置 NSEL2 位，可选择 OPA2 负端输入通道或选择 PGA 模式、设置 OPA 内部增益大小；配置 FB_EN2 位，可使能 OPA2 的 PGA 模式反馈；配置 PGADIF2 位，可使能差分输入 PGA 模式，使 OPA2 配合 NSEL2 作为 PGA 使用，且 N 端连接 PF12；配置 HS2 位，可使能高速模式，在此模式下可提高压摆率；配置 MODE2 位，可选择输出通道。

在 R32_OPA_CTLR3 寄存器中：配置 EN3 位，可使能 OPA3；配置 PSEL3 位，可选择 OPA3 正端输入通道；配置 NSEL3 位，可选择 OPA3 负端输入通道或选择 PGA 模式、设置 OPA 内部增益大小；配置 FB_EN3 位，可使能 OPA3 的 PGA 模式反馈；配置 PGADIF3 位，可使能差分输入 PGA 模式，使 OPA3 配合 NSEL3 作为 PGA 使用，且 N 端连接 PA3；配置 HS3 位，可使能高速模式，在此模式下可提高压摆率；配置 MODE3 位，可选择输出通道。

图 35-1 OPA1/OPA2/OPA3 的结构图



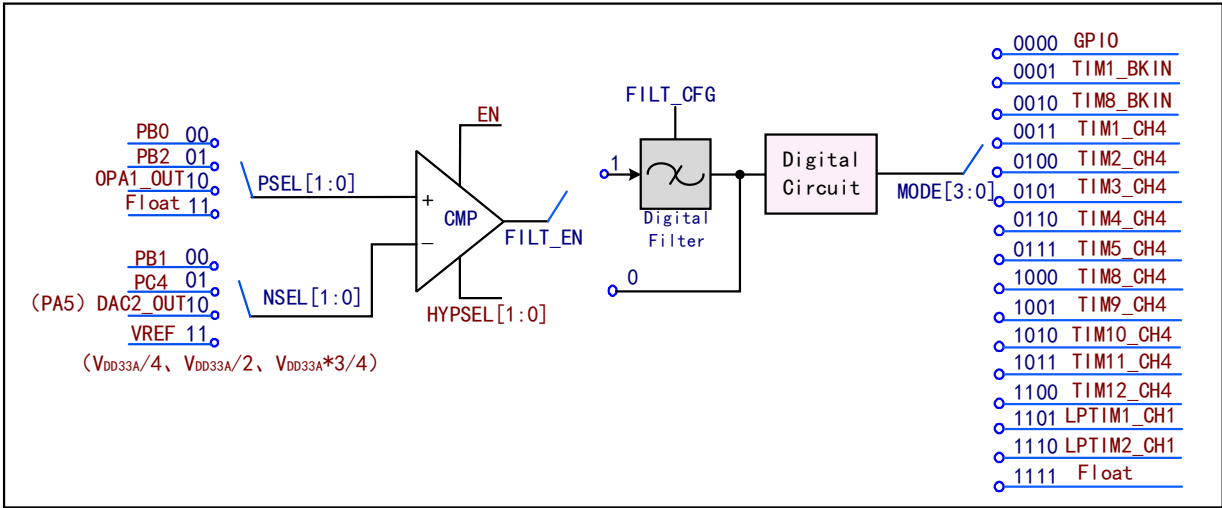
35.2.2 比较器 (CMP)

CMP 支持可选迟滞特性，支持输出端数字滤波功能，且输出滤波可选，结构图如图 35-2 所示。

在 R32_CMP_CTLR 寄存器中：配置 EN 位，可使能 CMP 模块；配置 PSEL[1:0] 位，可选择 P 端通道由 GPIO 输入或者在芯片内部与 OPA 运放连接；配置 NSEL[1:0] 位，可选择 N 端通道由 GPIO 输入或者在芯片内部与 DAC 输出连接或者可选的内部自偏置电压；配置 HYPSEL 位，可选择比较器迟滞电压；配置 VREF 位，可选择 CMP 内部偏置电压大小；配置 FILT_EN 位，使能 CMP 输出数字滤波功能；配置 FILT_CFG[8:0] 位，可选择 CMP 输出数字滤波采样间隔，采样间隔设置要大于噪声宽度，小于两倍噪声宽度；配置 FILT_BASE[2:0] 位，可配置 CMP 输出数字滤波采样的时间基准；配置 MODE[3:0] 位，可选择 CMP2 输出内部连接至 IO 或输出连接到定时器刹车输入端/定时器内部采样通道。

在 R32_CMP_STATR 寄存器中：通过 OUT_FILT 位，可读取 CMP 滤波功能关闭/开启的输出结果。

图 35-2 CMP 结构图



35.3 寄存器描述

表 35-4 OPA 相关寄存器列表

名称	访问地址	描述	复位值
R32_OPA_CTLR1	0x40017800	OPA 控制寄存器 1	0x00000076
R32_OPA_CTLR2	0x40017804	OPA 控制寄存器 2	0x00000070
R32_OPA_CTLR3	0x40017808	OPA 控制寄存器 3	0x00000070
R32_CMP_CTLR	0x4001780C	CMP 控制寄存器	0x000000FF
R32_CMP_STATR	0x40017810	CMP 状态寄存器	0x00000000

35.3.1 OPA 控制寄存器 1 (R32_OPA_CTLR1)

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					HS1	PGADIF1	FB_EN1	Reserved	NSEL1[2:0]			PSEL1	MODE1[1:0]		EN1

位	名称	访问	描述	复位值
[31:11]	Reserved	RO	保留。	0
10	HS1	RW	OPA1 高速模式使能: 1: 使能; 0: 禁止。	0
9	PGADIF1	RW	OPA1 配合 NSEL1 作为 PGA 使用, 且 N 端连接 OPA1_CHN1 (PA7): 1: 开启差分输入 PGA 模式; 0: 关闭差分输入 PGA 模式。	0
8	FB_EN1	RW	OPA1 的 PGA 模式反馈使能: 1: 使能;	0

			0: 禁止。	
7	Reserved	R0	保留。	0
[6:4]	NSEL1[2:0]	RW	OPA1 负端通道选择与 PGA 增益选择: 000: OPA1_CHN0 (PB1) ; 001: OPA1_CHN1 (PA7) ; 010: PGA 模式, 无负端输入通道, 内部增益为 8; 011: PGA 模式, 无负端输入通道, 内部增益为 16; 100: PGA 模式, 无负端输入通道, 内部增益为 32; 101: PGA 模式, 无负端输入通道, 内部增益为 64; 其他: 关断。	111b
3	PSEL1	RW	OPA1 正端通道选择: 1: OPA1_CHP1 (PA6) ; 0: OPA1_CHP0 (PB0) 。	0
[2:1]	MODE1[1:0]	RW	OPA1 输出通道选择: 00: 输出信号通过 PC4 输出; 01: 输出信号通过 PA5 输出; 1X: 输出信号不连接至 IO, 但仍连接至比较器正输入端。	11b
0	EN1	RW	OPA1 使能: 1: 使能; 0: 禁止。	0

35.3.2 OPA 控制寄存器 2 (R32_OPA_CTLR2)

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					HS2	PGADIF2	FB_EN2	Reserved	NSEL2[2:0]			PSEL2	Reserved	MODE2	EN2

位	名称	访问	描述	复位值
[31:11]	Reserved	R0	保留。	0
10	HS2	RW	OPA2 高速模式使能: 1: 使能; 0: 禁止。	0
9	PGADIF2	RW	OPA2 配合 NSEL2 作为 PGA 使用, 且 N 端连接 OPA2_CHN1 (PF14) : 1: 开启差分输入 PGA 模式; 0: 关闭差分输入 PGA 模式。	0
8	FB_EN2	RW	OPA2 的 PGA 模式反馈使能: 1: 使能; 0: 禁止。	0
7	Reserved	R0	保留。	0
[6:4]	NSEL2[2:0]	RW	OPA2 负端通道选择与 PGA 增益选择:	111b

			000: OPA2_CHN0 (PE8) ; 001: OPA2_CHN1 (PF12) ; 010: PGA 模式, 无负端输入通道, 内部增益为 8; 011: PGA 模式, 无负端输入通道, 内部增益为 16; 100: PGA 模式, 无负端输入通道, 内部增益为 32; 101: PGA 模式, 无负端输入通道, 内部增益为 64; 其他: 关断。	
3	PSEL2	RW	OPA2 正端通道选择: 1: OPA2_CHP1 (PF11) ; 0: OPA2_CHP0 (PE9) 。	0
2	Reserved	RO	保留。	0
1	MODE2	RW	OPA2 输出通道选择: 1: 输出信号通过 PB1 输出; 0: 输出信号通过 PE7 输出。	0
0	EN2	RW	OPA2 使能: 1: 使能; 0: 禁止。	0

35.3.3 OPA 控制寄存器 3 (R32_OPA_CTLR3)

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					HS3	PGADIF3	FB_EN3	Reserved	NSEL3[2:0]			PSEL3	Reserved	MODE3	EN3

位	名称	访问	描述	复位值
[31:11]	Reserved	RO	保留。	0
10	HS3	RW	OPA3 高速模式使能: 1: 使能; 0: 禁止。	0
9	PGADIF3	RW	OPA3 配合 NSEL3 作为 PGA 使用, 且 N 端连接 OPA3_CHN1 (PA3) : 1: 开启差分输入 PGA 模式; 0: 关闭差分输入 PGA 模式。	0
8	FB_EN3	RW	OPA3 的 PGA 模式反馈使能: 1: 使能; 0: 禁止。	0
7	Reserved	RO	保留。	0
[6:4]	NSEL3[2:0]	RW	OPA3 负端通道选择与 PGA 增益选择: 000: OPA3_CHN0 (PC3) ; 001: OPA3_CHN1 (PA3) ; 010: PGA 模式, 无负端输入通道, 内部增益为 8; 011: PGA 模式, 无负端输入通道, 内部增益为 16;	111b

			100: PGA 模式, 无负端输入通道, 内部增益为 32; 101: PGA 模式, 无负端输入通道, 内部增益为 64; 其他: 关断。	
3	PSEL3	RW	OPA3 正端通道选择: 1: OPA3_CHP1 (PA2) ; 0: OPA3_CHP0 (PC2) 。	0
2	Reserved	RO	保留。	0
1	MODE3	RW	OPA3 输出通道选择: 1: 输出信号通过 PA4 输出; 0: 输出信号通过 PA0 输出。	0
0	EN3	RW	OPA3 使能: 1: 使能; 0: 禁止。	0

35.3.4 CMP 控制寄存器 (R32_CMP_CTLR)

偏移地址: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	FILT_BASE[2:0]			Reserved			FILT_CFG[8:0]								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FILT_EN	VREF[1:0]		HYPSEL[1:0]		EN	MODE[3:0]			NSEL[1:0]		PSEL[1:0]		

位	名称	访问	描述	复位值
31	Reserved	RO	保留。	0
[30:28]	FILT_BASE[2:0]	RW	CMP 滤波采样时基配置。	0
[27:25]	Reserved	RO	保留。	0
[24:16]	FILT_CFG[8:0]	RW	CMP 滤波采样间隔配置。	0
[15:14]	Reserved	RO	保留。	0
13	FILT_EN	RW	CMP 数字滤波使能: 1: 开启; 0: 关闭。	0
[12:11]	VREF[1:0]	RW	CMP 内部偏置电压选择位: 00: 无效; 01: $1/4V_{DD33A}$; 10: $1/2V_{DD33A}$; 11: $3/4V_{DD33A}$ 。	0
[10:9]	HYPSEL[1:0]	RW	CMP 迟滞电压选择位: 00: 0mV; 01: 10mV; 10: 20mV; 11: 30mV。	0
8	EN	RW	CMP 使能: 1: 使能;	0

			0: 关闭。	
[7:4]	MODE[3:0]	RW	CMP 输出通道选择: 0000: 连接至 GPIO (PA6: AF10; PA8: AF12; PB12: AF13; PC5: AF13; PE6: AF11; PE12: AF13; PE15: AF13); 0001: TIM1_BKIN; 0010: TIM8_BKIN; 0011: TIM1_CH4; 0100: TIM2_CH4; 0101: TIM3_CH4; 0110: TIM4_CH4; 0111: TIM5_CH4; 1000: TIM8_CH4; 1001: TIM9_CH4; 1010: TIM10_CH4; 1011: TIM11_CH4; 1100: TIM12_CH4; 1101: LPTIM1_CH1; 1110: LPTIM2_CH1; 1111: 关闭输出功能。	1111b
[3:2]	NSEL[1:0]	RW	CMP 负端通道选择位: 00: CHN0 (PB1); 01: CHN1 (PC4); 10: DAC2 输出端 (PA5); 11: CMP 内部偏置电压。	11b
[1:0]	PSEL[1:0]	RW	CMP 正端通道选择位: 00: CHP0 (PB0); 01: CHP1 (PB2); 10: OPA1 输出; 11: 保留。	11b

35.3.5 CMP 状态寄存器 (R32_CMP_STATR)

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														OUT_F ILT	

位	名称	访问	描述	复位值
[31:1]	Reserved	RO	保留。	0
0	OUT_FILT	RO	CMP 输出: 当 CMP 数字滤波使能 (FILT_EN = 1) 时, 输出的是滤波值; 当 CMP 数字滤波关闭 (FILT_EN = 0) 时, 输出的是正常值。	0

第 36 章 串行音频接口（SAI）

串行音频接口（SAI）支持各种通用的音频协议，如 I2S、PCM/DSP、AC' 97、LSB 或 MSB 对齐和 TDM，它适用于单声道和立体声应用。为了实现 SAI 接口的灵活性和可配置性，它采用两个完全独立的音频子模块，每个音频子模块包含多达 4 个 IO 引脚（SD，SCK，FS 和 MCLK）。

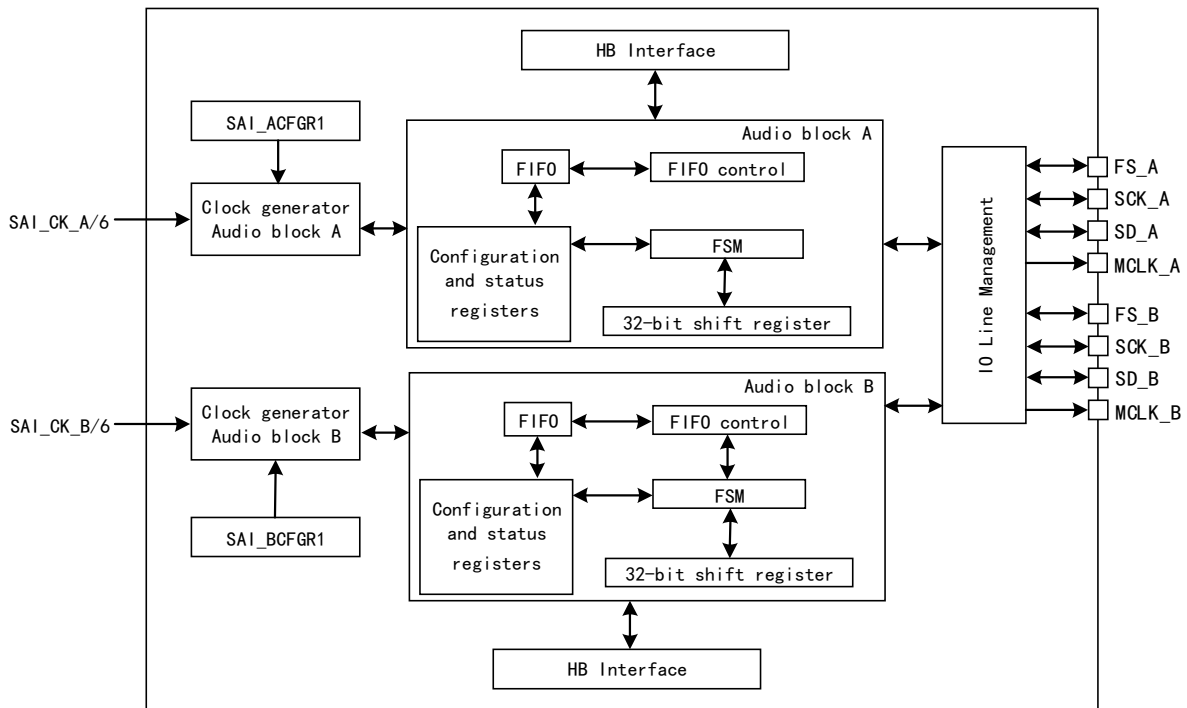
SAI 可以配置成主/从、发送/接收的任何组合，根据音频子模块同步/异步配置，可以设置其操作模式为全双工/单工。

36.1 主要特征

- 提供 2 组独立的控制模块，每个音频子模块可以配置成主/从、发送/接收的任何组合，并都具有一个 8 字的 FIFO
- 两个音频子模块间可以是同步或异步模式，主/从配置相互独立
- 灵活的配置 Slot 长度，并且可以配置 Slot 为有效或无效
- 帧同步配置（有效电平、有效长度和偏移）
- 每个音频帧包含多达 16 个可配置的 Slot
- 高达 16 个 Slot，每个 Slot 能支持一个大小为 8 位、10 位、16 位、20 位、24 位或 32 位的数据
- 支持 LSB 或 MSB 数据传输
- 支持 I2S 标准、LSB 或 MSB 对齐、PCM/DSP、TDM 和 AC' 97 等多种音频协议
- 具有单声道/立体声音频功能，支持静音设置
- 串行时钟选通边沿选择（SCK）
- 每个音频子模块都有 2 个独立的 DMA 接口，支持频率高达 4MHz 的从机模式
- 错误标志位和中断源：
 - FIFO 上溢和下溢；
 - 从模式时，帧同步提前检测；
 - 从模式时，帧同步滞后检测；
 - AC' 97 编解码器未就绪；
 - 时钟配置错误。

36.2 概述

图 36-1 SAI 的结构框图



SAI 主要由两个独立的音频子模块组成，并通过一个 I/O 管理模块连接到输出，I/O 线控制器管理 SAI 中指定音频模块的 4 个专用引脚（SD、SCK、FS、MCLK）。如果将两个子模块声明为同步模块，则可以共用其中某些引脚，从而留出一些引脚用作通用 I/O。MCLK 引脚是否可用作输出引脚取决于实际应用和解码要求以及音频模块是否配置为主模式。

每个音频模块集成一个 32 位移位寄存器，该寄存器由模块自身的功能状态机控制。数据存储和读取都是通过专用的 FIFO 来完成。FIFO 可通过 CPU 访问，也可通过 DMA 访问以减轻 CPU 的通信负担。

音频子模块在主模式或从模式下均可用作发送器或接收器。主模式意味着从 SAI 生成 SCK 位时钟和帧同步信号，而从模式则意味着 SCK 位时钟和帧同步信号来自另一外部或内部主器件。在特殊情况下，FS 信号方向与主模式或从模式定义不直接相关。在 AC' 97 协议中，即使 SAI（链接控制器）设置为消耗 SCK 时钟，FS 信号也会是 SAI 输出（从模式下也是如此）。

36.3 功能描述

36.3.1 时钟发生器

SAI 的两个独立的音频子模块都有自己的时钟发生器，但这两个时钟发生器在功能上没有任何差异。

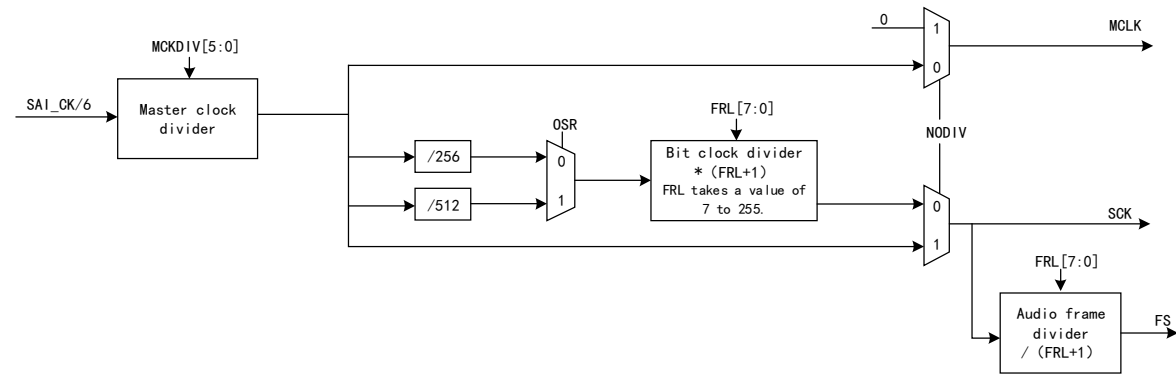
当音频模块被配置为主模式时，时钟发生器将负责提供位时钟（SCK）以及用于外部解码器的主时钟（MCLK）。将音频模块定义为从模式时，时钟发生器将关闭。

SCK 上的位时钟选通边沿可通过 R32_SAI_xCFGR1 寄存器 CKSTR 位配置。该位对主模式和从模式均适用。

建议选择 PLL 输出 295MHz 作为 SAI 的输入时钟。

音频模块时钟发生器的架构如图 36-2 所示。

图 36-2 音频模块时钟发生器的架构



注：当 NODIV 置 1 时，如果 MCLK 引脚配置为 GPIO 外设中的 SAI 引脚，则其信号电平将置 0。

时钟发生器的时钟源来自乘积时钟控制器。SAI_CLK_x 时钟等效于主时钟，可通过 MCKDIV[5:0] 位为外部解码器分频：

如果 MCKDIV[5:0] 不等于 0000b，则 $MCLK_x = SAI\_CK_x / (MCKDIV[5:0] * 2)$

如果 MCKDIV[5:0] 等于 0000b，则 $MCLK_x = SAI\_CK_x$

MCLK 信号仅在 TDM 中使用。

分频必须均匀，使 MCLK 输出上和 SCK 时钟上都保持 50% 的占空比。如果 MCKDIV[5:0] 位设置为 0000b，采用一分频可使 MCLK 等于 SAI_CLK_x。

在 SAI 中，使用单一比率 $MCLK/FS = 256$ 。大多数情况下，将遇到三个频率范围，如表 36-1 所示。

表 36-1 可能的音频采样范围示例

输入 SAI_CLK_x 时钟频率	可获得的常见音频采样频率	MCKDIV[5:0]
192kHz * 256	192kHz	000000
	96kHz	000001
	48kHz	000010
	16kHz	000110
	8kHz	001100
44.1kHz * 256	44.1kHz	000000
	22.05kHz	000001
	11.025kHz	000010
SAI_CLK_x = MCLK	MCLK	000000

注：当乘积时钟控制器选择一个外部时钟源而非 PLL 时钟时，会出现这种情况。

如果相应音频模块作为主模块且 R32_SAI_xCFGR1 寄存器 NODIV 位设置为 0，就可通过 I/O 口为外部解码器生成主时钟。将音频模块定义为从模式时，时钟发生器将关闭。NODIV、MCKDIV 位的值将被忽略。此外，MCLK I/O 引脚也会被释放，可作为通用 I/O 使用。

位时钟通过主时钟导出。位时钟分频器按照以下公式设置位时钟（SCK）和主时钟（SCK）间的分频系数：

$$SCK = MCLK * (FRL[7:0] + 1) / 256$$

其中：

256 是 MCLK 和音频采样频率之间的固定比率；

FRL[7:0] 是音频帧中的位时钟周期 - 1，在 R32_SAI_xFRCCR 寄存器中配置。

主模式下，(FRL[7:0]+1) 必须等于 2 的几次幂，以通过位时钟周期获得偶数个 MCLK 脉冲。位时钟（SCK）上将保证 50% 的占空比。

SAI_CLK_x 时钟还可等于位时钟频率。在这种情况下，R32_SAI_xCFGR1 寄存器中的 NODIV 位位置

1, MCKDIV 分频器内的值以及位时钟分频器内的值将被忽略。此时, 每个帧的位数完全可配置, 而无需等于 2 的几次幂。

36.3.2 操作模式

SAI 音频子模块可以通过 R32_SAI_xCFGR1 寄存器 MODE[1:0]位配置成主/从、发送/接收任何组合的操作模式。

SAI 中的每个音频模块均通过 R32_SAI_xCFGR1 寄存器 SAIEN 位使能。在从模式下, 此位一经激活, 发送器或接收器便会对时钟线、数据线和同步线上的活动敏感。

在主 TX 模式下, 即使 FIFO 中没有数据, 使能音频模块也会立即为外部从模块产生位时钟, 但 FS 信号的产生受 FIFO 中是否存在数据的控制。FIFO 接收到要发送的第一个数据后, 此数据将输出到外部从模块。如果 FIFO 中没有要发送的数据, 则随后将在音频帧中传送值 0, 并会产生一个下溢标志。

在从模式下, 使能音频模块时和检测到 SOF 位时开始音频帧。

在从 TX 模式下, 使能音频模块后的第一个帧上不可能出现下溢事件, 因为此时的强制操作顺序如下:

- 1) 通过软件或 DMA 写入 R32_SAI_xDATAR;
- 2) 等待至 FIFO 阈值 (FLTH) 标志与 000b (FIFO 为空) 不同;
- 3) 使能音频模块为从发送模式。

主模式

在主模式下, SAI 会向连接外部器件提供时钟信号:

- 1) 位时钟和帧同步分别在引脚 SCK 和 FS 上输出;
- 2) 如有需要, SAI 也可以在 MCLK 引脚上生成主时钟。

FS 在 FIFO 不为空且帧开始时生成, 而 CK 和 MCLK 的产生不受 FIFO 是否为空的制约, 只要音频子模块被使能就会生成。

从模式

在从模式下, SAI 从外部器件接收时钟信号:

- 1) 将 SAI 子模块配置为异步模式, 则 SCK 引脚和 FS 引脚将被配置为输入。
- 2) 如果将 SAI 子模块配置为与第二个音频子模块同步运行, 则会释放相应的 SCK 引脚和 FS 引脚以用作通用 I/O。

在从模式下, 不使用 MCLK 引脚, 可将其分配给其它功能。建议在使能主器件前先使能从器件。

36.3.3 内部同步模式

音频子模块可以和同一 SAI 中的另一个音频子模块同步运行, 也叫内部同步模式。在这种情况下, 二者将共用位时钟和帧同步信号, 以减少通信时占用的外部引脚数。配置为同步模式的音频模块将释放其 SCK、FS 和 MCLK 引脚以用作 GPIO, 而配置为异步模式的音频模块将使用其 SCK、FS 和 MCLK I/O 引脚 (如果该音频模块被配置为主模块)。

通常, 同步模式下的音频模块可用于在全双工模式下配置 SAI。两个音频模块中的一个可配置为主模块, 另一个为从模块; 也可将两个均配置为从模块; 一个模块为异步模块 (R32_SAI_xCFGR1 寄存器 SYNCEN[1:0]位设置为 00b), 另一个为同步模块 (R32_SAI_xCFGR1 寄存器 SYNCEN[1:0]位设置为 01b)。

注: 由于存在内部重新同步阶段, 因此 HCLK 频率必须大于比特率时钟频率的二倍。

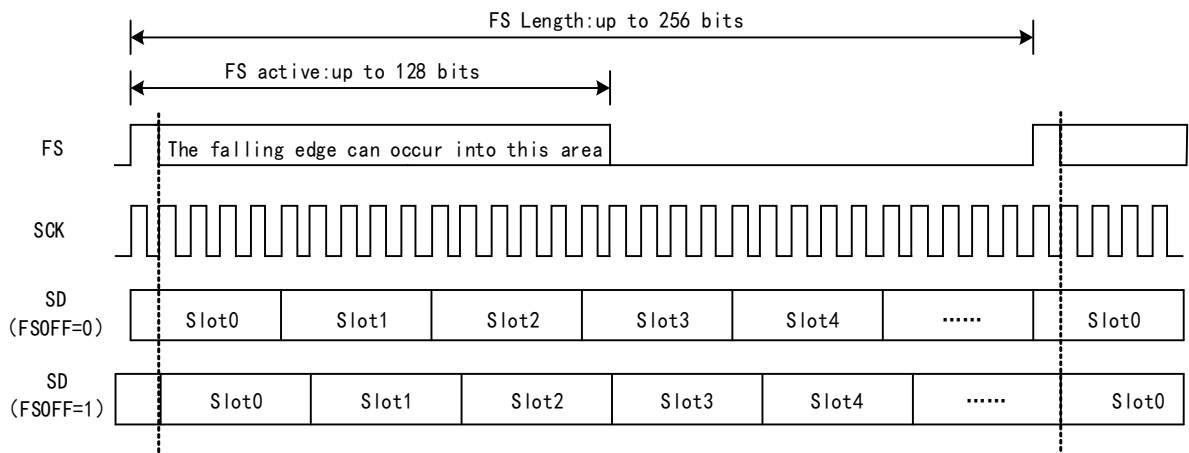
36.3.4 帧配置

36.3.4.1 帧同步

FS 信号用作音频中的帧同步信号 (SOF)。可通过配置 R32_SAI_xFCR 寄存器来实现此信号的波

形完全可配置，可在帧同步时，支持各种具有特殊规格的音频协议。下图展示了音频帧的可配置性。

图 36-3 音频帧



在 AC' 97 模式或 SPDIF 模式下（R32_SAI_xCFGR1 寄存器 PRTCFG[1:0] 位为 10b 或 01b），帧同步信号的波形被强制配置为支持 AC' 97 协议。R32_SAI_xFRCR 寄存器值被忽略。

每个音频模块相互独立，因此均需要特定的配置。

36.3.4.2 帧长度

主模式

将 R32_SAI_xFRCR 寄存器 FRL[7:0] 字段置 1，可将音频帧的长度配置为最长 256 个位时钟周期。

如果帧长度大于为该帧声明的 Slot 数，则要发送的剩余位将用 0 填充，或者 SD 线将释放为高阻态，具体取决于 R32_SAI_xCFGR2 寄存器 TRIS 位的状态。在接收模式下，剩余位被忽略。

如果将 NODIV 位清零，（FRL+1）必须等于 2 的几次幂（从 8 到 256）以确保音频帧的每个位时钟周期都包含整数个 MCLK 脉冲。如果将 NODIV 位置 1，则（FRL+1）字段可采用从 8 到 256 的任意值。

从模式

音频帧的长度主要用于指定由外部主模块向从模块发送的每个音频帧的位时钟周期数。它主要用于从主模块中检测音频帧传输期间出现的提前或滞后帧同步信号。这种情况下会出现错误。

在从模式下，R32_SAI_xFRCR 寄存器 FRL[7:0] 位的配置不受任何限制。

36.3.4.3 帧同步极性

通过 R32_SAI_xFRCR 寄存器 FSPOL 位设置 FS 引脚的有效极性，通过该极性来启动帧。SOF 信号对边沿敏感。

在从模式下，音频模块属于等待状态，直至接收到一个有效的帧信号来执行发送或接收操作，SOF 信号与该帧信号同步。仅当在通信过程中未检测到 SOF 信号并且 SOF 信号与预期 SOF 信号一致时，帧同步极性才有效。

在主模式下，每次音频帧完成时，都会发送帧同步信号，直至 R32_SAI_xCTLR1 寄存器 SAIEN 位被清零。如果前一个音频帧结束时 FIFO 中没有数据，则将按照 36.3.12.1 小节所述处理下溢情况，但音频通信流不会因此中断。

36.3.4.4 帧同步有效电平长度

R32_SAI_xFRCR 寄存器 FSALL[6:0] 位用于配置帧同步信号的有效电平的长度。该长度可设置为 1 到 128 个位时钟周期。

在 L2S、LSB 或 MSB 对齐模式下，有效长度设置为帧长的一半；而在 PCM/DSP 或 TDM 模式下，有效长度设置为 1 位。

36.3.4.5 帧同步偏移

根据应用中所支持的音频协议（例如 I2S 标准协议和 MSB 对齐协议），设置音频帧同步偏移，可在发送的最后一位或第一位时，将帧同步信号置为有效。可通过 R32_SAI_xFRCR 寄存器 FS0FF 位选择两个配置之一。

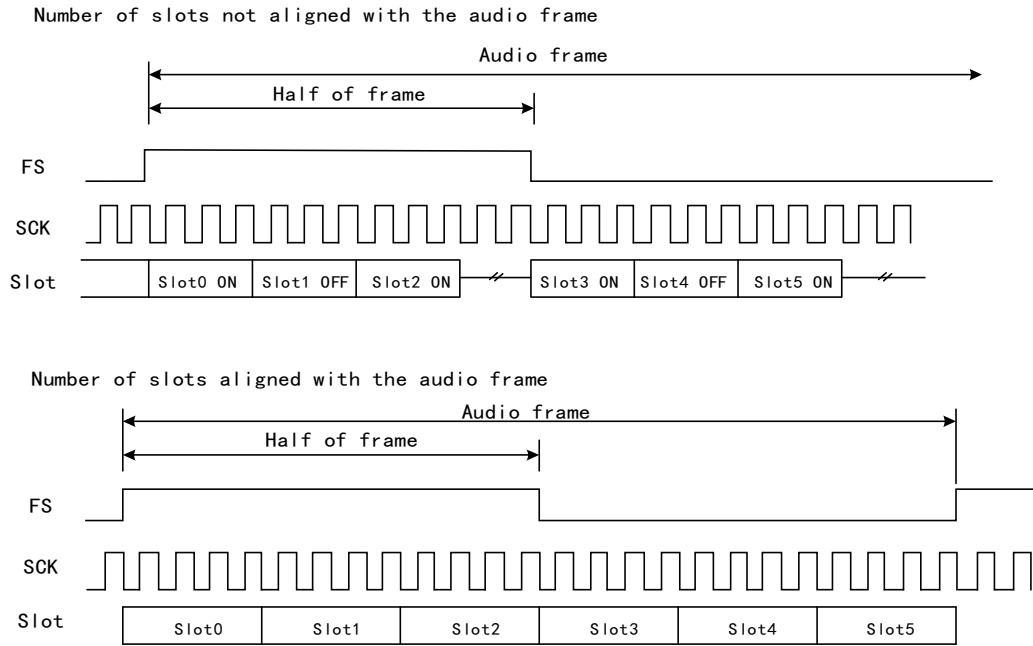
36.3.4.6 FS 信号的作用

FS 信号的含义因 FS 的功能的不同而异，在 R32_SAI_xFRCR 寄存器 FSDEF 位可配置 FS 信号的含义：

- 当 FSDEF 位为 1 时，FS 信号同时承载帧内的 S0F 信号和通道识别信号，适用于 I2S、MSB 或 LSB 对齐协议等音频帧结构；
- 当 FSDEF 位为 0 时，FS 信号表现为 S0F 信号，适用于 PCM/DSP、TDM、AC’ 97 以及音频协议。

在将 FS 信号解释为帧内的 S0F 信号和通道识别信号时，声明的 Slot 数量应平均分配给左通道和右通道。若半个音频帧上的位时钟周期数大于某个通道的专用 Slot 数，且 TRIS 位设置为 0，则 R32_SAI_xCFGR2 寄存器中剩余的位时钟周期将输出 0。反之，若 TRIS 位设置为 1，则 SD 线将进入高阻态。在接收时，只有在通道发生变化时，才会考虑剩余的位时钟周期。

图 36-4 FS 的作用是 S0F 信号+通道识别信号



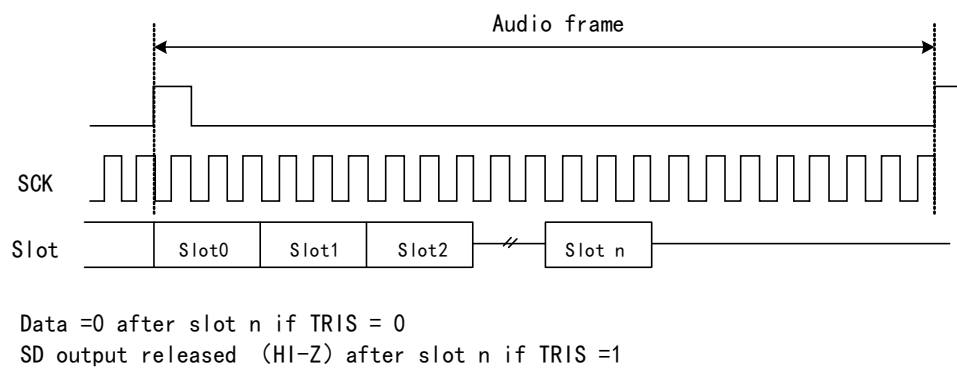
注：帧长度应为偶数。

如果 R32_SAI_xFRCR 寄存器 FSDEF 位保持清零状态，则 FS 信号等效于 S0F 信号，如果 R32_SAI_xSLOTR 寄存器 NBSLOT[3:0]位定义的 Slot 数乘以 R32_SAI_xSLOTR 寄存器 SLOTSZ[1:0]位配置的 Slot 位数所得的结果小于帧大小（R32_SAI_xFRCR 寄存器 FRL[7:0]位），则：

如果 R32_SAI_xCFGR2 寄存器 TRIS 位设置为 0，则最后一个 Slot 后的剩余位将强制为 0，直至发送器的帧结束。

如果 R32_SAI_xCFGR2 寄存器 TRIS 位设置为 1，则传输这些剩余位时，数据线将释放为高阻态。在接收模式下，这些位被丢弃。

图 36-5 FS 的作用是 SOF 信号



在发送模式下，音频模块被配置为捕获 SD 线上的 SPDIF 输出时，将不涉及 FS 信号的使用。相应的 FS I/O 将被释放，以便于其他用途。

36.3.5 Slot 配置

音频帧的基本单元为 Slot，每个音频帧最多可分为 16 个 Slot。音频帧中的 Slot 的数量由 NBSLOT[3:0]+1 来确定。

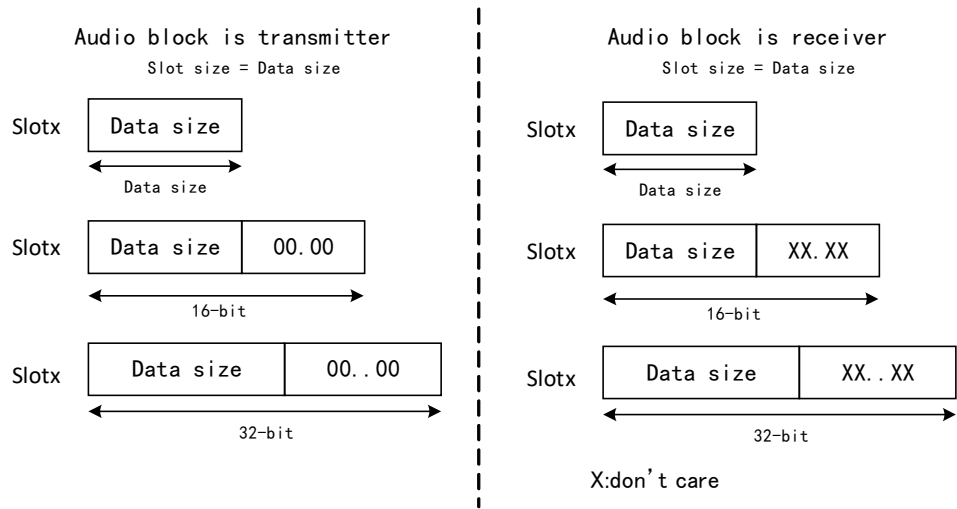
对于 AC’97 协议或 SPDIF 协议（当 PRTCFG[1:0]位设置为 10b 或 01b 时），Slot 的数量将自动遵循相应的协议规范进行配置，此时 NBSLOT[3:0]的值将被忽略。

通过设置 R32_SAI_xSLOTR 寄存器 SLOTEN[15:0]位，可以将各个 Slot 设置为有效或无效状态。

当传输一个无效 Slot 时，SD 数据线将被强制置为 0 或置于高阻态，具体取决于在发送模式下 TRIS 位的配置（详见无效 Slot 输出数据线管理部分）。在接收模式下，从该无效 Slot 结束后接收的数据将被忽略。因此不会有 FIFO 访问，也不会有与该无效 Slot 状态相关的 FIFO 读/写请求。

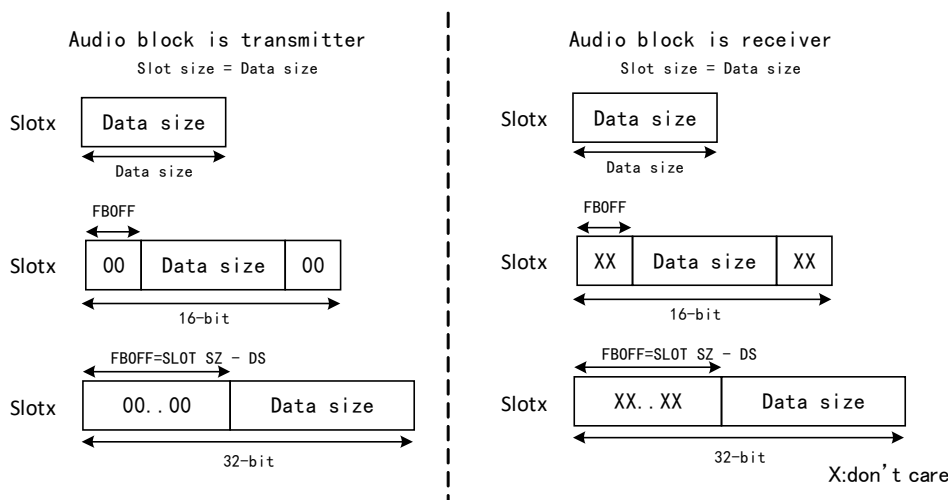
此外，Slot 的大小可以通过将 R32_SAI_xSLOTR 寄存器 SLOTSZ[1:0]位置 1 来选择，如图 36-6 所示。

图 36-6 FB0FF 位为 0 时的 Slot 大小配置



通过 R32_SAI_xSLOTR 寄存器 FB0FF[4:0]位可配置 Slot 内要传输的第一个数据位的位置，即第一个位的偏移值。如图 36-7 所示，在发送模式下，将从 Slot 起始位置注入 0 值，直至达到预设的偏移位置；而在接收模式下，位于偏移阶段中的位将被忽略。该特性适用于 LSB 对齐协议（如果偏移等于 Slot 大小减去数据大小）。

图 36-7 第一位偏移



要避免出现故障 SAI 行为，必须遵循以下条件：

$$FBOFF \leq (SLOTSZ - DS),$$

$$DS \leq SLOTSZ,$$

$$NBSLOT \times SLOTSZ \leq FRL \text{ (帧长度)},$$

在 R32_SAI_xFRCR 寄存器 FSDEF 位置 1 时，Slot 数必须为偶数。

在 AC' 97 和 SPDIF 协议 (PRTCFG[1:0]位设置为 10b 或 01b) 中，Slot 大小按照第 36.3.7 节：AC' 97 链路控制器中的定义自动设置。

36.3.6 内部 FIFO

在每个 SAI 音频子模块中都有一个独立的 8 字宽的同步 FIFO，以提高传输效率。这些 FIFO 可以被 CPU 或是 DMA 访问，FIFO 请求中断机制用于请求 CPU 和 DMA 访问。

根据音频模块被定义为发送器还是接收器，相应地执行对 FIFO 的读取或写入。无论访问大小为何，每次对 FIFO 进行读写时，均针对 1 个字的 FIFO 单元进行操作。每个 FIFO 字包含一个音频 Slot。每次访问 R32_SAI_xDATAR 寄存器后，FIFO 指针递增一个字。因此只存在一个 FIFO 请求与 R32_SAI_xSR 寄存器 FREQ 位 (FIFO 请求标志) 相关联。

当 R32_SAI_xINTENR 寄存器 FREQIE 位使能，将产生中断。该中断的触发条件取决于操作模式、FIFO 阈值、FIFO 状态和 DMA 突发传输大小。

具体触发中断的机制将在表 36-2 和后续段落“在发送模式下产生中断”和“在接收模式下产生中断”中详细阐述。

表 36-2 FIFO 请求中断的产生条件

发送：MODE[1:0]=x0				接收：MODE[0]=x1			
FIFO 阈值	FTH[2:0]	FIFO 状态	FLTH[2:0]	FIFO 阈值	FTH[2:0]	FIFO 状态	FLTH[2:0]
空	=000	空	=000	空	=000	非空	≥001
1/4 满	=001	<1/4 满	<010	1/4 满	=001	≥1/4 满	≥010
1/2 满	=010	<1/2 满	<011	1/2 满	=010	≥1/2 满	≥011
3/4 满	=011	<3/4 满	<100	3/4 满	=011	≥3/4 满	≥100
全满	=100	非满	<101	全满	=100	全满	=101

注：（1）如果根本条件不满足，则中断请求就会被硬件清除。

（2）将 R32_SAI_xCFGR2 寄存器 FFLUSH 位置 1 禁止 SAI 后，可重新初始化 FIFO 指针。如果 FFLUSH 在 SAI 使能情况下置 1，则 FIFO 中的数据将自动清除，读写指针复位到 0。

在发送模式下产生中断

根据发送模式下的 FIFO 配置产生中断：

当 R32_SAI_xCFGR2 寄存器中的 FIFO 阈值位配置为 FIFO 为空时 (FTH[2:0] 置为 000b)，若寄存器中没有数据 (R32_SAI_xSR 寄存器 FLTH[2:0] 位小于 001b)，将产生中断 (R32_SAI_xSR 寄存器 FREQ 位由硬件置 1)。当 FIFO 不再为空时 (R32_SAI_xSR 中的 FLTH[2:0] 位不是 000b)，即 FIFO 中存储一个或多个数据时，此中断 (R32_SAI_xSR 寄存器 FREQ 位) 由硬件清零。

当 R32_SAI_xCFGR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之一满时 (FTH[2:0] 置为 001b)，如果不到四分之一的 FIFO 包含数据 (R32_SAI_xSR 中的 FLTH[2:0] 位小于 010b)，将产生中断 (R32_SAI_xSR 寄存器 FREQ 位由硬件置 1)。当至少四分之一的 FIFO 包含数据时 (R32_SAI_xSR 寄存器 FLTH[2:0] 位大于或等于 010b)，此中断 (R32_SAI_xSR 寄存器 FREQ 位) 由硬件清零。

当 R32_SAI_xCFGR2 寄存器中的 FIFO 阈值位配置为 FIFO 半满时 (FTH[2:0] 置为 010b)，如果不到一半的 FIFO 包含数据 (R32_SAI_xSR 寄存器 FLTH[2:0] 位小于 011b)，将产生中断 (R32_SAI_xSR 寄存器 FREQ 位由硬件置 1)。当至少一半的 FIFO 包含数据时 (R32_SAI_xSR 中的 FLTH[2:0] 位大于或等于 011b)，此中断 (R32_SAI_xSR 寄存器 FREQ 位) 由硬件清零。

当 R32_SAI_xCFGR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之三满时 (FTH[2:0] 置为 011b)，如果不到四分之三的 FIFO 包含数据 (R32_SAI_xSR 寄存器 FLTH[2:0] 位小于 100b)，将产生中断 (R32_SAI_xSR 寄存器 FREQ 位由硬件置 1)。当至少四分之三的 FIFO 包含数据时 (R32_SAI_xSR 寄存器 FLTH[2:0] 位大于或等于 100b)，此中断 (R32_SAI_xSR 寄存器 FREQ 位) 由硬件清零。

当 R32_SAI_xCFGR2 寄存器中的 FIFO 阈值位配置为 FIFO 为满时 (FTH[2:0] 置为 100b)，如果 FIFO 不满 (R32_SAI_xSR 寄存器 FLTH[2:0] 位小于 101b)，将产生中断 (R32_SAI_xSR 寄存器 FREQ 位由硬件置 1)。当 FIFO 已满时 (R32_SAI_xSR 寄存器 FLTH[2:0] 位等于值 101b)，此中断 (R32_SAI_xSR 寄存器 FREQ 位) 由硬件清零。

在接收模式下产生中断

根据接收模式下的 FIFO 配置产生中断：

当 R32_SAI_xCFGR2 寄存器中的 FIFO 阈值位配置为 FIFO 为空时 (FTH[2:0] 置为 000b)，如果 R32_SAI_xDATAR 寄存器中至少有一个数据 (R32_SAI_xSR 寄存器 FLTH[2:0] 位大于或等于 001b)，将产生中断 (R32_SAI_xSR 寄存器 FREQ 位由硬件置 1)。当 FIFO 变为空时 (R32_SAI_xSR 寄存器 FLTH[2:0] 位等于 000b)，即 FIFO 中未存储数据时，此中断 (R32_SAI_xSR 寄存器 FREQ 位) 由硬件清零。

当 R32_SAI_xCFGR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之一满时 (FTH[2:0] 置为 001b)，如果至少有四分之一的 FIFO 数据单元可用 (R32_SAI_xSR 寄存器 FLTH[2:0] 位大于或等于 010b)，将产生中断 (R32_SAI_xSR 寄存器 FREQ 位由硬件置 1)。当不到四分之一的 FIFO 数据单元可用时 (R32_SAI_xSR 中的 FLTH[2:0] 位小于 010b)，此中断 (R32_SAI_xSR 寄存器 FREQ 位) 由硬件清零。

当 R32_SAI_xCFGR2 寄存器中的 FIFO 阈值位配置为 FIFO 半满时 (FTH[2:0] 置为 010b)，如果至少有一半的 FIFO 数据单元可用 (R32_SAI_xSR 寄存器 FLTH[2:0] 位大于或等于 011b)，将产生中断 (R32_SAI_xSR 寄存器 FREQ 位由硬件置 1)。当不到一半的 FIFO 数据单元可用时 (R32_SAI_xSR 寄存器 FLTH[2:0] 位小于 011b)，此中断 (R32_SAI_xSR 寄存器 FREQ 位) 由硬件清零。

当 R32_SAI_xCFGR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之三满时 (FTH[2:0] 置为 011b)，如果至少有四分之三的 FIFO 数据单元可用 (SAI_xSR 中的 FLTH[2:0] 位大于或等于 100b)，将产生中断 (R32_SAI_xSR 寄存器 FREQ 位由硬件置 1)。当不到四分之三的 FIFO 数据单元可用时 (R32_SAI_xSR 寄存器 FLTH[2:0] 位小于 100b)，此中断 (R32_SAI_xSR 寄存器 FREQ 位) 由硬件清零。

当 R32_SAI_xCFGR2 寄存器中的 FIFO 阈值位配置为 FIFO 满时 (FTH[2:0] 置为 100b)，如果 FIFO 已满 (R32_SAI_xSR 寄存器 FLTH[2:0] 位等于 101b)，将产生中断 (R32_SAI_xSR 寄存器 FREQ 位由硬件置 1)。当 FIFO 不满时 (R32_SAI_xSR 寄存器 FLTH[2:0] 位小于 101b)，此中断 (R32_SAI_xSR 寄存器 FREQ 位) 由硬件清零。

注：与中断的产生类似，如果 R32_SAI_xCFGR1 寄存器 DMAEN 位置 1，则 SAI 可使用 DMA。

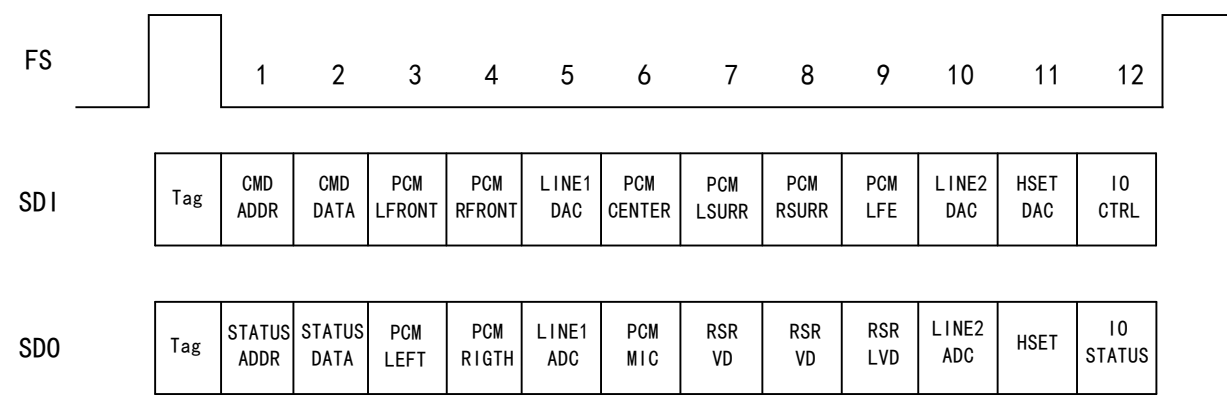
36.3.7 AC’ 97 链路控制器

SAI 可用作 AC’ 97 链路控制器，该模式是通过 R32_SAI_xCFGR1 寄存器 PRTCFG[1:0]位配置的，当选择了此协议，有许多配置字段会被忽略，所以 Slot 数和 Slot 大小是固定的，帧同步信号定义完善并且波形固定，否则将无法保证 SAI 运行正常。

- NBSLOT[3:0]位和 SLOTSZ[1:0]位将被忽略；
- Slot 数固定为 13。第一个 Slot 为 16 位宽，所有其它 Slot 均为 20 位宽（数据 Slot）；
- SAI_xSLOTR 寄存器 FBOFF[4:0]位被忽略；
- SAI_xFRCR 寄存器被忽略；
- 未使用 MCLK。

无论采用主配置还是从配置，由于 AC’ 97 链路控制器驱动 FS 信号，异步模块发出的 FS 信号将自动配置为输出。

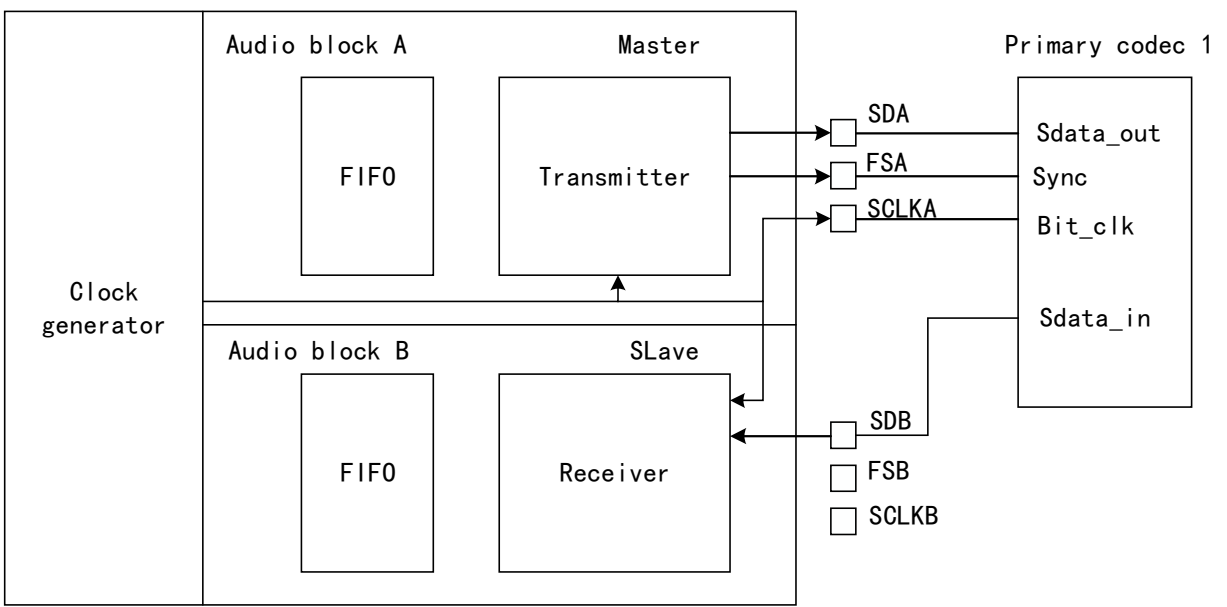
图 36-8 AC’ 97 音频帧



注：在 AC’ 97 协议中，TAG 的位 2 将保留（始终为 0），因此无论 SAI FIFO 中写入何值，TAG 的位 2 均强制为 0。有关 TAG 表示的详细信息，请参见 AC’ 97 协议标准。

可将一个 SAI 用于 AC’ 97 点对点通信，如图 36-9 所示。在 SAI 中，必须将音频模块 A 声明为异步主发送器，而将音频模块 B 定义为从接收器，并在内部与音频模块 A 同步。

图 36-9 SAI 用于 AC’ 97 点对点通信



在接收器模式下，用作 AC’ 97 链路控制器的 SAI 无需 FIFO 请求，因此当 Slot0 中的编码解码器就绪位解码为低电平时，FIFO 中不存储任何数据。如果 R32_SAI_xINTENR 寄存器 CNRDYIE 位使能，则 R32_SAI_xSR 寄存器中的标志 CNRDY 将置 1 并会产生一个中断。此标志专用于 AC’ 97 协议。

AC' 97 模式下的时钟发生器编程

在AC' 97模式下，帧长度固定为256位，其频率应设置为48kHz。第36.3.1节：SAI时钟发生器中给出的规则应在FRL=255时使用，以生成适当的帧速率（F_{FS}）。

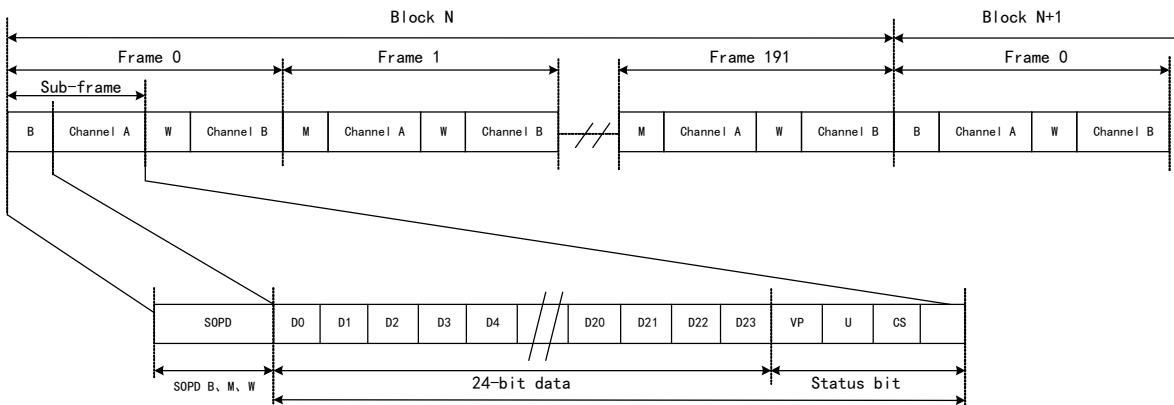
36.3.8 SPDIF 输出

SPDIF（索尼/飞利浦数字接口）是一种用于消费音频设备的数字音频互连，用于在合理的短距离内输出音频。SPDIF 接口仅在发送模式下可用，它支持 IEC 60958 标准。可通过将 R32_SAI_xCFGR1 寄存器 PRTCFG[1:0]位置为 01b，选择 SPDIF 模式。

对于 SPDIF 协议：

- 1) 仅使能 SD 数据线；
- 2) 释放 FS、SCK、MCLK I/O 引脚；
- 3) 为使能 SAI 的时钟发生器并管理 SD 线上的数据速率，强制将 MODE[1]位设置为 0，以选择主模式；
- 4) 数据大小强制设置为 24 位。对 R32_SAI_xCFGR1 寄存器的 DS[2:0]位中所设置的值忽略；
- 5) 必须配置时钟发生器以定义符号率，已知位时钟应为符号率的两倍。数据采用曼彻斯特协议进行编码；
- 6) 忽略 SAI_xFRCR 和 SAI_xSLOTR 寄存器。在内部配置 SAI 使其符合 SPDIF 协议要求，如图 36-0 所示。

图 36-10 SPDIF 格式



SPDIF 块包含 192 个帧。每个帧由两个 32 位的子帧构成，通常一个子帧用于左通道，一个用于右通道。每个子帧由一个 SOPD 模式（4 位）构成，用于指定该子帧是否为块的开始（并用于识别通道 A），或者是否标识块中某处的通道 A，或者是否指代通道 B（请参见表 36-3）。接下来的 28 位是通道信息，由 24 个数据位和 4 个状态位组成。

表 36-3 SOPD 模式

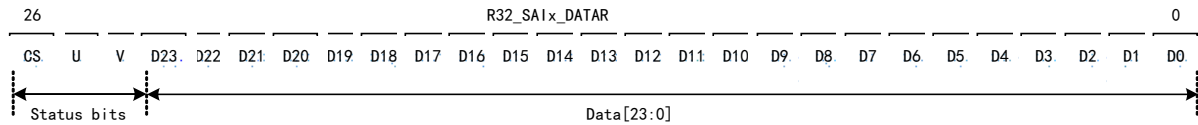
SOPD	报头编码		描述
	最后一位是 0	最后一位是 1	
B	11101000	00010111	通道 A 数据，且为一个块的起始子帧
W	11100100	00011011	通道 B 数据
M	11100010	00011101	通道 A 数据

SPDIF 的数据传输在 R32_SAI_xDATAR 寄存器的数据填充应遵循：

- R32_SAI_xDATAR[26:24] 包含通道状态位、用户位和有效性位；
- R32_SAI_xDATAR[23:0] 包含所考虑通道的 24 位数据。

注：（1）如果数据大小为 20 位，应将数据映射到 R32_SAI_xDATAR[23:4] 上。
（2）如果数据大小为 16 位，应将数据映射到 R32_SAI_xDATAR[23:8] 上。

图 36-11 R32_SAI_xDATAR 寄存器数据示意图



注：R32_SAI_xDATAR[23] 始终代表 MSB。执行传输时，LSB 始终优先。

SAI 首先在块中发送每个子帧的适当报头，随后在 SD 线上发送 R32_SAI_xDATAR（以曼彻斯特协议进行编码）。SAI 通过传输按表 36-4 所述计算的奇偶校验位来结束子帧。

表 36-4 校验位奇数

R32_SAI_xDATAR[26:0]	传输校验位 P 的值
奇数个 0	0
奇数个 1	1

在 SPDIF 模式下，下溢是在 R32_SAI_xSR 寄存器唯一用到的错误标志。因为 SAI 只能在发送模式下工作，所以，要从通过下溢中断或下溢状态位检测到的下溢错误中恢复，应按顺序执行以下步骤：

- 1) 如果已使用 DMA，则需要通过 DMA 外设禁止 DMA 流。
- 2) 禁止 SAI 操作，并通过轮询 R32_SAI_xCFGR1 寄存器中的 SAIEN 位确认已从物理上禁止外设。
- 3) 通过将 R32_SAI_xSR 寄存器 OVRUDR 位写 1 清除标志位。
- 4) 通过将 R32_SAI_xCFGR2 寄存器 FFLUSH 位置 1 来刷新 FIFO。软件需要指向与新数据块开始（报头 B 的数据）相对应的后续数据地址。如果已使用 DMA，应相应地更新 DMA 源起始地址指针。
- 5) 如果 DMA 控制器根据新的源起始地址指针重新开始传输数据，则重新使能 DMA 流。
- 6) 通过将 R32_SAI_xCFGR1 寄存器中的 SAIEN 位置 1，重新使能 SAI。

SPDIF 发生器模式下的时钟发生器编程

对于 SPDIF 发生器，SAI 应提供一个符号率相等的位时钟：

表 36-5 音频采样频率与符号率（SHARK）

音频采样频率（F _s ）	符号率
44.1kHz	2.8224MHz
48kHz	3.072MHz
96kHz	6.144MHz
192kHz	12.288MHz

通常，音频采样率（F_s）与位时钟速率（F_{SCK}）之间的关系由以下公式给出：

$$F_s = \frac{F_{SAI\_CK\_x}}{64}$$

36.3.9 立体声/单声道

在发送模式下，如果 Slot 数等于 2（R32_SAI_xSLOTR 寄存器 NBSLOT[3:0] 位设置为 0001b），则可寻址单声道模式，而无需在存储器中进行任何数据预处理。在这种情况下，由于发送时 Slot0 的数据被复制到数据 Slot1 中，FIFO 的访问时间将减少一半。

要使能单声道模式，需要：

- 1) 将 R32_SAI_xCFGR1 寄存器 MONO 位置 1；
- 2) 将 R32_SAI_xSLOTR 寄存器 NBSLOT 置 1，并将 SLOTEN 位设置为 3。

在接收模式下，仅当 Slot 数等于 2（与发送模式下相同）时，MONO 位的置 1 才有意义。当该位

置 1 时，只有 Slot0 的数据将存储到 FIFO 中，Slot1 的数据由于被认为与前一个 Slot 的数据相同而被丢弃。如果接收模式下的数据流量是左声道数据和右声道数据明显不同的真立体声音频流，则 MONO 位没有意义。由软件完成从输出立体声文件到等效单声道文件的转换。

36.3.10 静音模式

当音频子模块用作发送器或接收器时，可使用静音模式。

发送模式下的音频子模块

在发送模式下，可随时选择静音模式。静音模式对于全部音频帧均有效。在帧传输过程中通过将 R32_SAI_xCFGR2 寄存器 MUTE 位置为 1，即可使能静音模式。该静音模式位仅在帧结束时选通。如果在帧结束时将 MUTE 位置 1，静音模式将在新的音频帧开始时激活，并持续整个帧长度，直至下次帧结束；然后将选通该位，以确定下一帧是否仍为静音帧。

如果通过 R32_SAI_xSLOTR 寄存器 NBSLOT[3:0] 位设置的 Slot 数小于或等于 2，可指定静音模式下发送的值是否为 0 或该值是否为每个 Slot 的最后一个值。通过 R32_SAI_xCFGR2 寄存器 MUTEVAL 位进行选择。

如果在 R32_SAI_xSLOTR 寄存器 NBSLOT[3:0] 位中设置的 Slot 数大于 2，由于在各 Slot 的每位上都发送值 0，因此 R32_SAI_xCFGR2 寄存器 MUTEVAL 位没有意义。

在静音模式下，FIFO 指针仍递增，这意味着将丢弃 FIFO 中请求在静音模式下传输的数据。

接收模式下的音频子模块

在接收模式下，对于给定数量的连续音频帧（R32_SAI_xCFGR2 寄存器 MTECNT[5:0] 位），如果在音频帧所有声明的有效 Slot 接收到 0，则可检测到从外部发送器发来的静音模式。检测到相应数量的静音帧时，R32_SAI_xSR 寄存器 MTEDET 标志置 1 并会在 R32_SAI_xCFGR2 寄存器 MTEDETIE 位置 1 情况下产生中断。

在音频子模块禁止时或在有效 Slot 内至少接收到音频帧中的一个数据时，静音帧计数器清零。计数器达到 MTECNT[5:0] 位中指定的值时，仅产生一次中断。随后中断事件在计数器清零时重新初始化。

注：静音模式不可用于 SPDIF 音频模块。

36.3.11 压缩扩展模式

在移动通信领域，数据的传输前的处理往往涉及数据压缩与扩展的算法的应用。

应用可根据 R32_SAI_xCFGR2 寄存器 COMP[1:0] 位（仅当选择 TDM 模式时使用），选择在 SD 串行输出线（压缩）发送数据前是否处理数据，以及是否在 SD 串行输入线（扩展）接收数据后扩展数据，如图 36-11 所示。所支持的两个压扩模式是 μ -Law 和 A-Law，二者均为 CCITT G.711 推荐的标准。

美国和日本采用的压扩标准是 μ -Law，该标准允许 14 位动态范围（R32_SAI_xCFGR2 寄存器 COMP[1:0] 位为 10b）。欧洲压扩标准是 A-Law，该标准支持 13 位动态范围（R32_SAI_xCFGR2 寄存器 COMP[1:0] 位为 11b）。

关于 μ -Law 或 A-Law 压扩标准的计算方法，其依据 R32_SAI_xCFGR2 寄存器 CPL 位的设置，可采用 1 的补码或 2 的补码来进行计算。

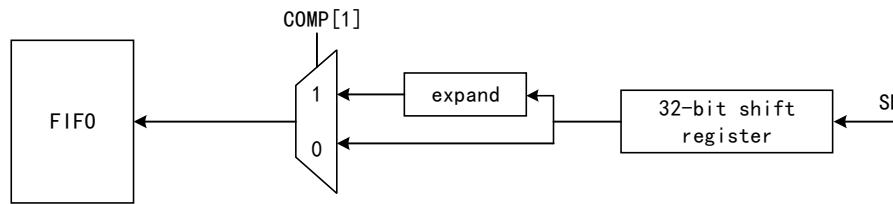
在 μ -Law 和 A-Law 标准中，数据被编码为采用 MSB 对齐的 8 位格式，经过压扩处理的数据宽度始终为 8 位。

因此，当 SAI 音频模块使能（R32_SAI_xCFGR1 寄存器 SAIE 位置 1）并且通过 COMP[1:0] 位选择这两个压扩模式之一时，R32_SAI_xCFGR1 寄存器 DS[2:0] 位将强制设为 010b。如果无需压扩处理，则 COMP[1:0] 位应保持清零。

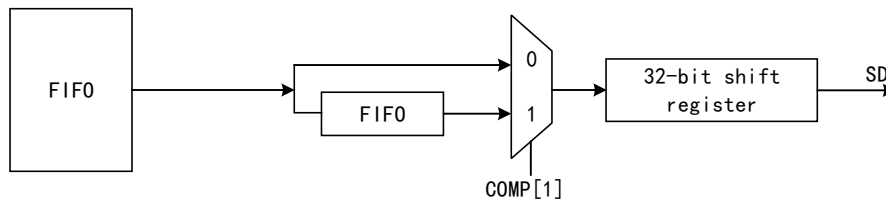
其中，如果 SAI 音频模块配置为发送器，且 R32_SAI_xCFGR2 寄存器 COMP[1] 位置 1，将采用压缩模式；如果 SAI 音频模块声明为接收器，将采用扩展算法。

图 36-12 SAI 的音频模块中的数据压扩硬件

Receiver mode (bit MODE[0]=1 in R32_SAI_xCFGR1)



Transmitter mode (bit MODE[0]=0 in R32_SAI_xCFGR1)



注：选择 AC' 97 时不适用。

无效 Slot 上的输出数据线管理

在发送模式下，当在数据线上发送无效 Slot 时，可通过 TRIS 位控制 SD 线输出的行为。

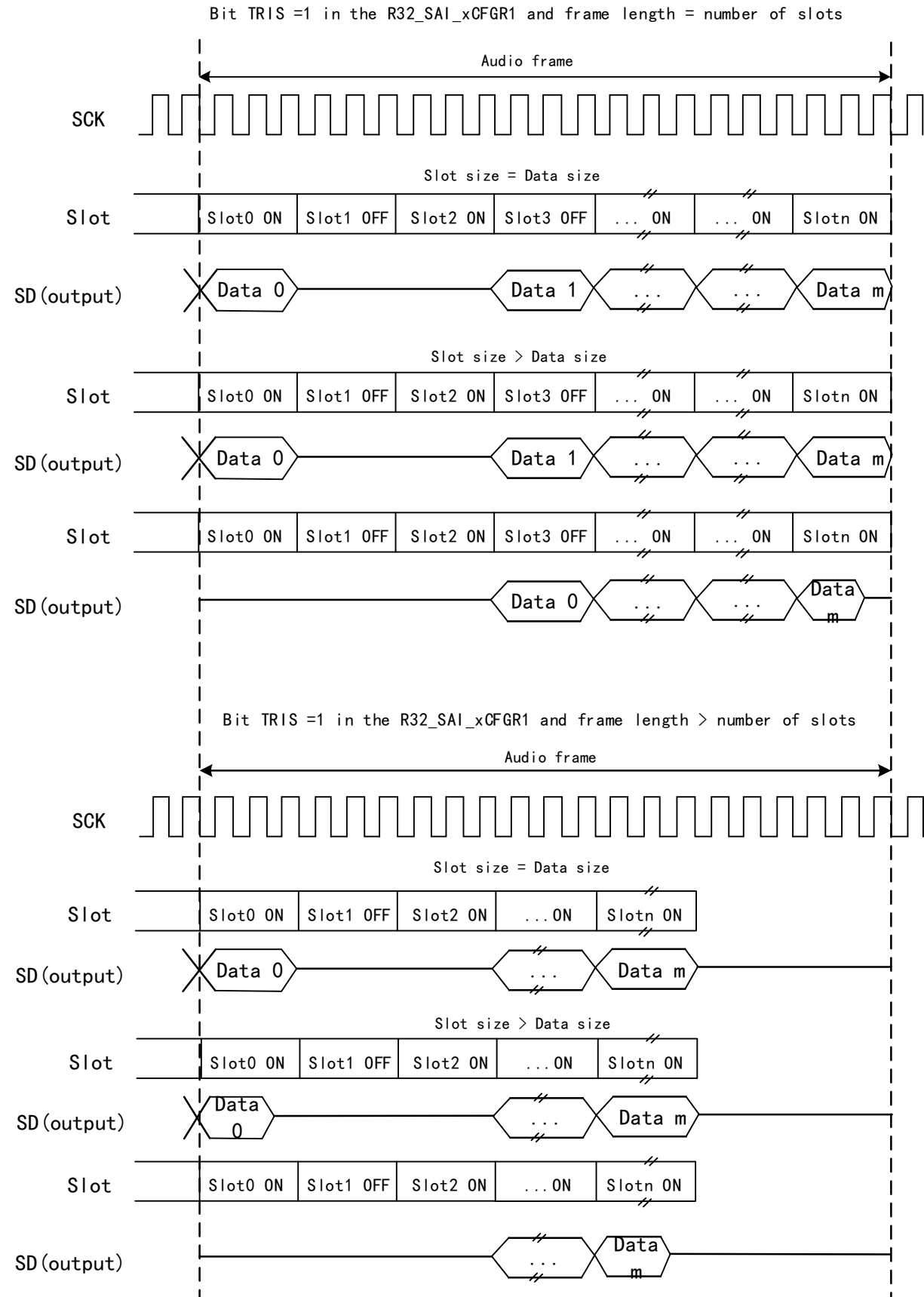
- 发送无效 Slot 时，SAI 将 SD 输出线强制为 0。
- 在最后一个数据位传输完毕后，SD 输出线将被置于高阻态，以便释放该数据线供其他连接此节点的发送器使用。

必须避免两个发送器同时驱动同一个 SD 输出引脚，以防止短路现象的发生。为确保发送过程中的间隙存在，如果数据低于 32 位，可在 R32_SAI_xSLOTR 寄存器中设置 SLOTSZ[1:0] 位为 10b，将数据扩展到 32 位。随后，如果下一个 Slot 被声明为无效，则 SD 输出引脚将在有效 Slot 的 LSD 结束时（将数据扩展到 32 位的填 0 阶段）被置为三态。

此外，如果 Slot 数与 Slot 大小的乘积小于帧长度，则在通过填 0 来补充音频帧结束时，SD 输出线将被置为三态。

图 36-13 说明了这些行为。

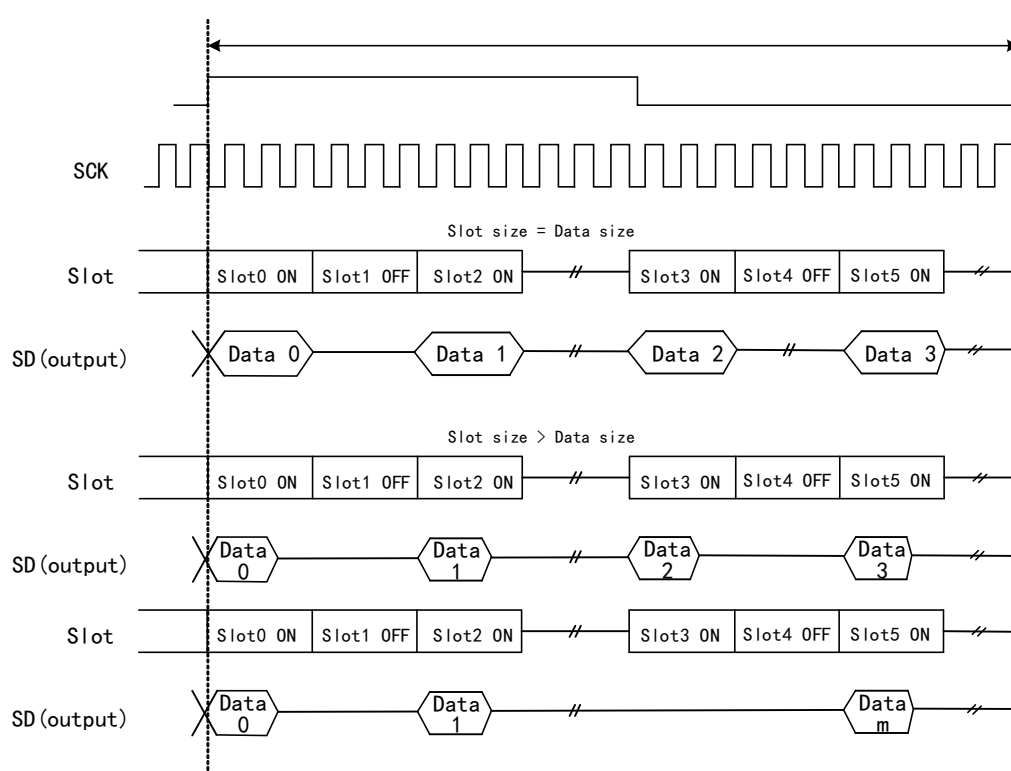
图 36-13 发送无效 Slot 时 SD 输出线上的三态策略



当所选音频协议使用 FS 信号作为 SOF 信号或通道识别信号（R32_SAI_xFCR 寄存器 FSDEF 位置 1）时，将按照图 36-14 管理三态模式（其中，R32_SAI_xCFGR1 寄存器 TRIS 位置 1，FSDEF 位置 1，

半帧长大于 Slot 数/2 且 NBSLOT=6)。

图 36-14 采用 I2S 等协议时输出数据线上的三态策略



如果 R32_SAI_xCFGR2 寄存器 TRIS 位清零，图 36-13 和图 36-14 上的 SD 输出线上的所有高阻态将替换为使用值 0 驱动。

36.3.12 错误标志

SAI 使用以下错误标志：

- FIFO 上溢/下溢；
- 帧同步提前检测；
- 帧同步滞后检测；
- 编解码器未就绪（仅限 AC' 97）；
- 主模式时钟配置错误。

36.3.12.1 FIFO 上溢/下溢（OVRUDR）

FIFO 上溢/下溢位是 R32_SAI_xSR 寄存器 OVRUDR 位。

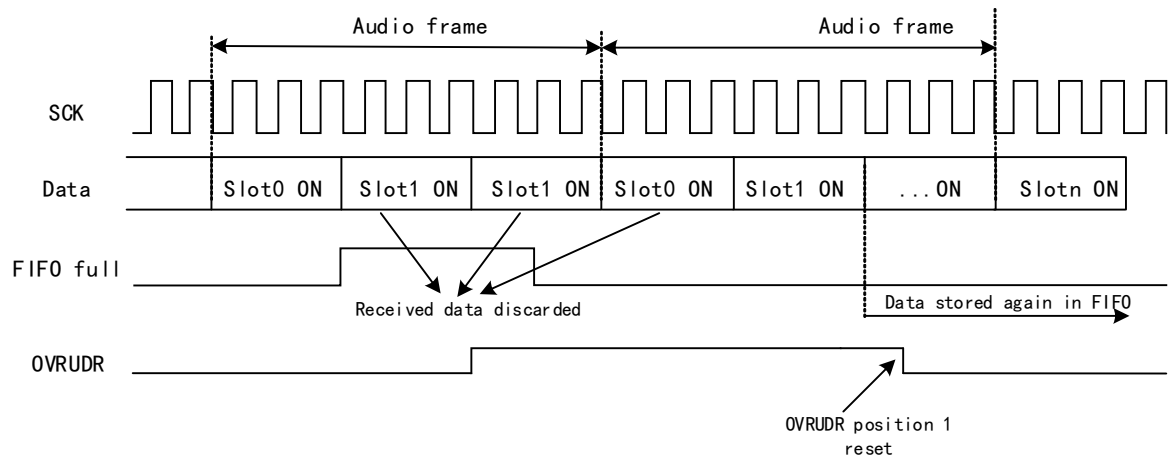
由于音频模块既可作为接收器，又可作为发送器，并且指定 SAI 中的每个音频模块都具有自己的 R32_SAI_xSR 寄存器，因此上溢或下溢错误共用同一位。

上溢

若音频模块配置为接收器，则在 FIFO 已满且无法再存储接收数据的情况下又收到音频帧数据时，将出现上溢情况。这种情况下，接收数据将丢失，R32_SAI_xSR 寄存器 OVRUDR 标志置 1；如果 R32_SAI_xINTENR 寄存器 OVRUDRIE 位置 1，还将生成中断。内部将记录发生上溢时的 Slot 编号。FIFO 无法再存储更多数据，直至释放出空间存储新数据为止。在 FIFO 释放了至少一个数据的空间时，SAI 音频模块接收器将从检测到上溢后内部记录的 Slot 编号开始接收来自新音频帧的新数据，这样可避免目标存储器中出现数据 Slot 不对齐的情况，具体请参见图 36-15。

要清除 OVRUDR 标志，可将 R32_SAI_xSR 寄存器 OVRUDR 位写 1 即可。

图 36-15 上溢错误检测



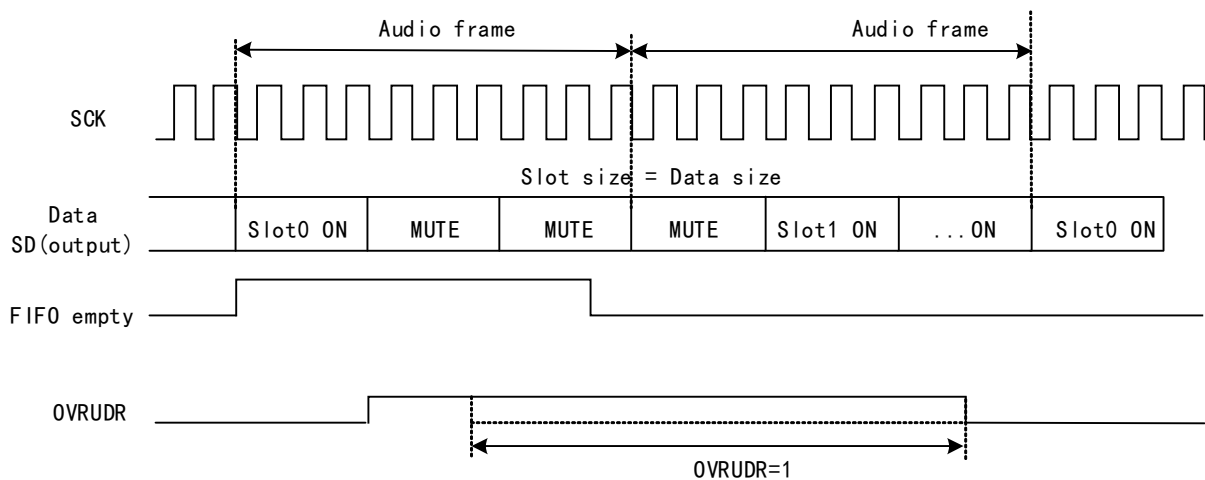
下溢

当 SAI 中的音频模块用作发送器时，如果需要发送数据时 FIFO 为空，则可能出现下溢。如果检测到下溢，则将存储发生事件的 Slot 编号并发送 MUTE 值 (00b)，直到 FIFO 准备好发送与检测到下溢的 Slot 对应的数据，具体请参见图 36-16。这样可避免存储器指针与音频帧中的 Slot 之间发生同步失效。

下溢事件会使 R32_SAI_xSR 寄存器 OVRUDR 标志置 1，如果 R32_SAI_xINTENR 寄存器 OVRUDRIE 位置 1，还将生成中断。要清除该标志，可将 R32_SAI_xSR 寄存器 OVRUDR 位写 1 即可。

将音频子模块配置为主模式或从模式时，会发生下溢事件。

图 36-16 FIFO 下溢事件



36.3.12.2 帧同步提前检测 (AFSDDET)

AFSDDET 标志仅在当 SAI 音频模块用作从模式时使用，在主模式下该标志将被禁用。在 R32_SAI_xFRCR 寄存器中，帧长度、帧极性和帧偏移配置均已已知，该标志用于指示是否比预期更早检测到帧同步 (FS) 信号。当发生帧检测提前时，R32_SAI_xSR 寄存器 AFSDDET 标志将置 1。

对 FS 提前不敏感的当前音频帧，次检测不会产生影响。换言之，FS 信号的“寄生”事件将被标记，但不会干扰当前音频帧的处理。

如果 R32_SAI_xINTENR 寄存器 AFSDETIE 位置 1，则会产生中断。清除 AFSDDET 标志需要必须将该位写 1。

为在发生帧检测提前错误后重新与主模块同步，必须遵循以下步骤：

- 1) 通过将 R32_SAI_xCFGR1 寄存器 SAIEN 位复位，禁止 SAI 模块。为确保 SAI 被禁止，需读回

SAIEN 位并确认其值已置为 0。

2) 通过将 R32_SAI_xCFGR2 寄存器 FFLUSH 位置 1，刷新 FIFO。

3) 通过将 R32_SAI_xCFGR1 寄存器 SAIEN 位置 1，重新使能 SAI 外设。

4) SAI 模块将等待 FS 使能，以重新开始与主模块同步。

注：AC' 97 模式下不使能 SAIEN 标志，原因在于 SAI 音频模块作为链路控制器，即使在被声明为从模块时，也会生成 FS 信号。由于 AC' 97 协议未使用 FS 信号，因此该功能在 SPDIF 模式下无实际意义。

36.3.12.3 帧同步滞后检测 (LFSDET)

LFSDET 标志仅在当 SAI 音频模块用作从模块时使用，在主模式下该标志将被禁用。在 R32_SAI_xFRCR 寄存器中，帧长度、帧极性和帧偏移配置均已知，如果外部主模块未在预定时间发送 FS 信号或 FS 信号生成过晚，LFSDET 标志将置 1，如果 R32_SAI_xINTENR 寄存器 LFSDETIE 位置 1，还将产生中断。如果将 R32_SAI_xSR 寄存器 LFSDET 位写 1，将可清除 LFSDET 标志。

在检测到相应错误时帧同步滞后检测标志置 1，SAI 需要重新与主模块同步，具体请参见第 36.3.12.2 小节：帧同步提前检测 (AFSDet) 中所述的顺序。

在噪声环境中，音频模块的状态机可能会错误检测到对 SCK 时钟的干扰，并将 SAI 数据移位到错误的帧位置。SAI 会检测到此事件，并将其报告为帧同步滞后检测错误。

如果外部主模块不是在连续模式下管理音频数据帧发送，就不会对帧造成破坏，大多数应用都不会出现这种情况。若是处在这种情况下，LFSDET 标志将置 1。

注：AC' 97 模式下不使能 LFSDET 标志，原因在于 SAI 音频模块作为链路控制器，即使在被声明为从模块时，也会生成 FS 信号。由于 AC' 97 协议未使用 FS 信号，因此该功能在 SPDIF 模式下无实际意义。

36.3.12.4 编解码器未就绪 (CNRDY AC' 97)

在 SAI 音频模块被配置为在 AC' 97 模式下工作时（即 R32_SAI_xCFGR1 寄存器 PRTCFG[1:0] 位设置为 10b），R32_SAI_xSR 寄存器 CNRDY 标志才有意义。如果 R32_SAI_xINTENR 寄存器 CNRDYIE 位置 1，则当 CNRDY 标志置 1 时，将生成中断。

在接收 AC' 97 音频帧的 TAG0 (slot0) 期间，若编解码器尚未准备就绪以进行通信，CNRDY 标志位将被置 1。在此情况下，直至 TAG0 指示编解码器就绪之前，数据都不会自动存储到 FIFO，原因在于编解码器尚未准备就绪。一旦编解码器就绪，将捕获 R32_SAI_xSLOTR 寄存器中定义的所有有效 Slot。

为清除 CNRDY 标志，必须将 R32_SAI_xSR 寄存器 CNRDY 位写 1。

主模式时钟配置错误 (NODIV=0)

当音频模块在主模式下工作 (MODE[1:0]=0x) 且 NODIV 位置为 0 时，如果满足以下条件，则只要使能 SAI，WCKCFG 标志位将被置为 1：

- (FRL+1) 不是 2 的几次幂；
- (FRL+1) 不在 8 和 256 之间。

MODE 位、NODIV 位和 SAIEN 位属于 R32_SAI_xCFGR1 寄存器，FRL 位属于 R32_SAI_xFRCR 寄存器。

如果 WCKCFGIE 位置 1，则当 R32_SAI_xSR 寄存器 WCKCFG 标志置 1 时将生成中断。若要清除该标志，可将 R32_SAI_xSR 寄存器 WCKCFG 位写 1 即可。

当 WCKCFG 位置 1 时，音频模块会自动禁止，因此将对 SAIEN 位执行硬件清零。

36.3.13 DMA 接口

为减轻 CPU 负担和优化总线带宽，每个 SAI 音频模块都具有独立的 DMA 接口，以便对 R32_SAI_xDATAR 寄存器进行读/写操作（访问内部 FIFO）。每个音频子模块均配有支持基本 DMA 请求/应答协议的 DMA 通道。

在配置 DMA 传输以服务音频子模块时，需将 R32_SAI_xCFGR1 寄存器 DMAEN 位置 1。DMA 请求由

FIFO 控制器直接管理，其产生条件取决于 FIFO 阈值。当音频子模块配成发送模式时，FIFO 阈值必须设成一个特定的值，以保证在最坏的情况下也有足够的剩余空间来实现一个完整的 DMA 突发写操作，否则有可能出现 FIFO 上溢错误。当音频子模块配成接收模式时，FIFO 阈值必须设成一个特定的值，以保证 FIFO 中有足够的数据来实现一个完整的 DMA 突发读操作，从而避免出现 FIFO 下溢错误。

DMA 传输方向与 SAI 音频子模块配置相关：

– 若音频模块用作发送器，则音频模块的 FIFO 控制器将输出 DMA 请求以向 FIFO 加载 SAI_xDATAR 寄存器中写入的数据；

– 如果音频模块用作接收器，则 DMA 请求与来自 R32_SAI_xDATAR 寄存器的读取操作相关。

按以下顺序将 SAI 接口配置为 DMA 模式：

- 1) 配置 SAI 和 FIFO 阈值以指定何时启动 DMA 请求。
- 2) 配置 SAIDMA 通道。
- 3) 使能 DMA。
- 4) 使能 SAI 接口。

注：配置 SAI 模块前，必须禁止 SAI DMA 通道。

36.3.14 SAI 中断

表 36-6 SAI 中断请求

中断源	中断划分	中断出现条件	中断使能控制	中断清除控制
FREQ	请求	MODE[1:0]为任意模式	FREQIE	取决于： - FIFO 阈值设置 - 通信传输方向 详见 36.3.6 小节
MUTEDET	静音	MODE[1:0]为接收器	MUTEDETIE	CMUTEDET = 1
OVRUDR	错误	MODE[1:0]为任意模式	OVRUDRIE	COVRUDR = 1
AFSDDET	错误	MODE[1:0]为从模式（不适用于 AC' 97 模和 SPDIF 模式）	AFSDDETIIE	CAFSDET = 1
LFSDET	错误	MODE[1:0]为从模式（不适用于 AC' 97 模和 SPDIF 模式）	LFSDETIIE	CLFSDET = 1
CNRDY	错误	MODE[1:0]为从模式（仅限 AC' 97 模式）	CNRDYIE	CCNRDY = 1
WCKCFG	错误	MODE[1:0]为主模式且 NODIV = 0	WCKCFGIE	CWCKCFG = 1

按照以下顺序使能中断：

- 1) 禁止 SAI 中断；
- 2) 配置 SAI 功能寄存器；
- 3) 使能 SAI 中断源；
- 4) 使能 SAI。

36.4 寄存器描述

表 36-7 SAI 相关寄存器列表

名称	访问地址	描述	复位值
R32_SAI_ACFGR1	0x40015804	SAI 模块 A 配置寄存器 1	0x00000040
R32_SAI_ACFGR2	0x40015808	SAI 模块 A 配置寄存器 2	0x00000000
R32_SAI_AFRCR	0x4001580C	SAI 模块 A 帧配置寄存器	0x00000007

R32_SAI_ASLOTR	0x40015810	SAI 模块 A Slot 寄存器	0x00000000
R32_SAI_AINTENR	0x40015814	SAI 模块 A 中断使能寄存器	0x00000000
R32_SAI_ASR	0x40015818	SAI 模块 A 状态寄存器	0x00000008
R32_SAI_ADATAR	0x40015820	SAI 模块 A 数据寄存器	0x00000000
R32_SAI_BCFGR1	0x40015824	SAI 模块 B 配置寄存器 1	0x00000040
R32_SAI_BCFGR2	0x40015828	SAI 模块 B 配置寄存器 2	0x00000000
R32_SAI_BFRCR	0x4001582C	SAI 模块 B 帧配置寄存器	0x00000007
R32_SAI_BSL0TR	0x40015830	SAI 模块 B Slot 寄存器	0x00000000
R32_SAI_BINTENR	0x40015834	SAI 模块 B 中断使能寄存器	0x00000000
R32_SAI_BSR	0x40015838	SAI 模块 B 状态寄存器	0x00000008
R32_SAI_BDATAR	0x40015840	SAI 模块 B 数据寄存器	0x00000000

36.4.1 SAI 模块 x 配置寄存器 1 (R32_SAI_xCFGR1) (x=A/B)

偏移地址: 0x04, 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					OSR	MCKDIV[5:0]					NODIV	Reserved	DMAEN	SAIEN	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				MONO	SYNCEN[1:0]	CKSTR	LSBFIRST	DS[2:0]		Reserved	PRTCFCG[1:0]		MODE[1:0]		

位	名称	访问	描述	复位值
[31:27]	Reserved	R0	保留。	0
26	OSR	RW	主时钟的过采样率: 1: $F_{FS} \times 512$; 0: $F_{FS} \times 256$ 。	0
[25:20]	MCKDIV[5:0]	RW	主时钟分频器: 000000: 主时钟输入 1 分频; 其他: 主时钟频率将根据以下公式计算: $F_{SCK\_x} = \frac{F_{SAI\_CK\_x}}{MCKDIV}$ 注: 仅在音频模块禁止情况下配置该位。	0
19	NODIV	RW	无分频器: 1: 在时钟发生器中不使用分频器 (此时主时钟分频器位不起作用); 0: 使能主时钟发生器。	0
18	Reserved	R0	保留。	0
17	DMAEN	RW	DMA 使能: 1: 使能; 0: 关闭。 注: 在接收模式下, 必须在 DMAEN 位置 1 前配置 MODE[1:0] 位, 以避免 DMA 请求, 原因是复位后音频模块将默认以发送模式工作。	0
16	SAIEN	RW	音频模块使能:	0

			1: 使能 SAIx 音频模块; 0: 禁止 SAIx 音频模块。 注: 当 SAI 模块 (A 或 B) 配置为主模式时, SAI 模块输入中必须有时钟, 然后才能将 SAIEN 位置 1。	
[15:13]	Reserved	R0	保留。	0
12	MONO	RW	单声道模式选择: 1: 单声道模式; 0: 立体声模式。 注: 仅当 Slot 数为 2 时该位才有意义。	0
[11:10]	SYNCEN[1:0]	RW	同步使能: 00: 音频子模块处于异步模式; 01: 音频子模块与另一个内部音频子模块同步。这种情况下, 必须将该音频子模块配置为从模式; 其他: 保留。 注: (1) 使能 SPDIF 模式后, 应将音频子模块配置为异步; (2) 仅在音频模块禁止情况下配置该位。	0
9	CKSTR	RW	时钟选通边沿: 1: 在 SCK 下降沿更改 SAI 生成的信号, 而在 SCK 上升沿对 SAI 接收的信号进行采样; 0: 在 SCK 上升沿更改 SAI 生成的信号, 而在 SCK 下降沿对 SAI 接收的信号进行采样。 注: 仅在音频模块禁止情况下配置该位。	0
8	LSBFIRST	RW	最低有效位优先: 1: 优先传输数据的 LSB; 0: 优先传输数据的 MSB。 注: (1) 仅在音频模块禁止情况下配置该位。 (2) 当使用 LSB 模式时, 接收数据在高位: 如接收数据大小位 16 位, 数据内容为 0x1234, 实际接收到的数据为 0x12340000。	0
[7:5]	DS[2:0]	RW	数据大小选择: 000/001: 保留; 010: 8 位; 011: 10 位; 100: 16 位; 101: 20 位; 110: 24 位; 111: 32 位。 注: 仅在音频模块禁止情况下配置该位。	010b
4	Reserved	R0	保留。	0
[3:2]	PRTCFG[1:0]	RW	协议配置: 00: 自由协议。通过设置大部分配置寄存器位以及帧配置寄存器, 自由协议允许用户使用音频模块这一强大的配置功能来处理特定的音频协议 (如 I2S、LSB/MSB 对齐、TDM、PCM/DSP...);	0

			01: SPDIF 协议; 10: AC' 97 协议; 11: 保留。 注: 仅在音频模块禁止情况下配置该位。	
[1:0]	MODE[1:0]	RW	SAIx 音频模块模式: 00: 主发送器; 01: 主接收器; 10: 从发送器; 11: 从接收器。 注: (1) 如果将音频模块配置为 SPDIF 模式, 则将强制设置主发送模式 (MODE[1:0] = 00)。在主发送模式下, 音频模块将立即开始生成 FS 和时钟; (2) 仅在音频模块禁止情况下配置该位。	0

36.4.2 SAI 模块 x 配置寄存器 2 (R32_SAI_xCFGR2) (x=A/B)

偏移地址: 0x08, 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
COMP[1:0]		CPL	MUTECNT[5:0]						MUTEVAL	MUTE	TRIS	FFLUSH	FTH[2:0]		

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:14]	COMP[1:0]	RW	压扩模式: 00: 不支持压扩算法; 01: 保留; 10: μ -Law 算法; 11: A-Law 算法。 注: 仅在选择了 TDM 协议时才能使用压扩模式。	0
13	CPL	RW	补码位: 1: 2 的补码表示; 0: 1 的补码表示。 注: 仅当压扩模式为 μ -Law 算法或 A-Law 算法时该位才有效。	0
[12:7]	MUTECNT[5:0]	RW	静音计数器: 这些位中所设置的值将与接收模式下检测到的连续静音帧数量进行比较。当静音帧数量与该值相等时, MUTEDET 标志置 1, 并且在 MUTEDETIE 位置 1 的情况下, 还将生成中断。 注: 该域仅用于接收模式。	0
6	MUTEVAL	RW	静音值设置: 1: 静音模式期间发送上一个值; 0: 静音模式期间发送位值 0。	0

			注：该位对 SPDIF 音频模块无意义，从而也不使用。	
5	MUTE	RW	静音模式使能： 1：使能静音模式； 0：禁止静音模式。 注：该位对 SPDIF 音频模块无意义，从而也不使用。	0
4	TRIS	RW	数据线的三态管理： 1：SD 输出线将在上一个有效 Slot（下一个 Slot 无效）的最后一个数据位传输结束时释放（高阻态）； 0：Slot 无效时，SD 输出线仍由 SAI 驱动。	0
3	FFLUSH	RW	FIFO 刷新： 1：FIFO 刷新。将此位编程为 1 可触发 FIFO 刷新。所有的内部 FIFO 指针（读和写）将清零。这种情况下，仍存留在 FIFO 中的数据丢失（发送或接收数据不会继续丢失）。刷新 SAI 前，必须禁止 DMA 数据流/中断； 0：禁止 FIFO 刷新。	0
[2:0]	FTH[2:0]	RW	FIFO 阈值设置： 000：FIFO 为空； 001： $\frac{1}{4}$ FIFO； 010： $\frac{1}{2}$ FIFO； 011： $\frac{3}{4}$ FIFO； 100：FIFO 已满。 其他：保留。	0

36.4.3 SAI 模块 x 帧配置寄存器（R32_SAI_xFRCR）（x=A/B）

偏移地址：0x0C, 0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													FS0 FF	FSP OL	FSD EF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reser ved	FSALL[6:0]						FRL[7:0]								

位	名称	访问	描述	复位值
[31:19]	Reserved	R0	保留。	0
18	FSOFF	RW	帧同步偏移： 1：在 Slot 0 第一位的前一位上使能 FS； 0：在 Slot 0 的第一位上使能 FS。	0

17	FSPOL	RW	帧同步极性： 1：FS 为高电平有效（上升沿）； 0：FS 为低电平有效（下降沿）。 注：FSPOL=0 时须外接上拉电阻。	0
16	FSDEF	RO	帧同步定义： 1：FS 信号为 SOF 信号 + 通道识别信号； 0：FS 信号为起始帧信号。 此位置 1 时，R32_SAI_xSLOTR 寄存器中定义的 Slot 数必须为偶数。这意味着有半数 Slot 将用于左通道，其它 Slot 用于右通道（例如，对于 I2S 或 MSB/LSB 对齐等协议，该位必须置 1）。	0
15	Reserved	RO	保留。	0
[14:8]	FSALL[6:0]	RW	帧同步有效电平长度： 指定音频帧中 FS 信号的有效电平长度，以位时钟数（SCK）+ 1（FSALL[6:0] + 1）表示。	0
[7:0]	FRL[7:0]	RW	定义以 SCK 时钟周期数表示的音频帧长度：帧中的位数等于 FRL[7:0] + 1。 音频帧中发送的位数必须大于或等于 8，否则音频模块将出现操作异常。数据大小为 8 位且在 R32_SAI_xSLOTR 寄存器的 NBSLOT[4:0] 中只定义了一个 Slot（NBSLOT[3:0] = 0000）时便属于这种情况。 在主模式下，如果使用主时钟（MCLK_x 引脚上提供），则帧长度应为 8 到 256 之间的一个等于 2 的几次幂的数。不使用主时钟（NODIV = 1）时，建议将帧长度编程为 8 到 256 之间的值。	0x07

注：R32_SAI_AFRCR 和 R32_SAI_BFRCR 寄存器必须在音频模块禁止的情况下配置，且对于 AC' 97 和 SPDIF 音频协议无意义。

36.4.4 SAI 模块 x Slot 寄存器（R32_SAI_xSLOTR）（x=A/B）

偏移地址：0x10，0x30

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

位	名称	访问	描述	复位值
[31:16]	SLOTEN[15:0]	RW	Slot 使能： 1：有效 Slot； 0：无效 Slot。 每个 SLOTEN 位都对应于从 0 到 15 的一个 Slot 位置（最多 16 个 Slot）。	0
[15:12]	Reserved	RO	保留。	0

[11:8]	NBSLOT[3:0]	RW	音频帧中的 Slot 数： 设置的值表示音频帧中的 Slot 数 + 1（包括无效 Slot 数）。Slot 数最大值为 16。R32_SAI_xFRCR 寄存器 FSDEF 位置 1 时，Slot 数应为偶数。	0
[7:6]	SLOTSZ[1:0]	RW	Slot 大小： 00: Slot 大小与数据大小（在 R32_SAI_xCFGR1 寄存器 DS[3:0] 位中指定）相当； 01: 16 位； 10: 32 位； 11: 保留。 Slot 大小必须大于或等于数据大小。如果不满足该条件，SAI 的行为将不确定。	0
5	Reserved	RO	保留。	0
[4:0]	FBOFF[4:0]	RW	第一个位偏移： 设置的值定义 Slot 中第一个数据传输位的位置。它表示一个偏移值。在发送模式下，此数据字段以外的位将强制清零。在接收模式下，将丢弃额外接收的位。	0

注：R32_SAI_ASLOTR 和 R32_SAI_BSL0TR 寄存器必须在音频模块禁止的情况下配置，且对于 AC' 97 和 SPDIF 音频协议无意义。

36.4.5 SAI 模块 x 中断使能寄存器（R32_SAI_xINTENR）（x=A/B）

偏移地址：0x14, 0x34

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									LFSDETIE	AFSDETIE	CNRDYIE	FREQIE	WCKCFGIE	MUTETIE	OVRURIE

位	名称	访问	描述	复位值
[31:7]	Reserved	RO	保留。	0
6	LFSDETIE	RW	帧同步滞后检测中断使能： 1: 使能中断； 0: 禁止中断。	0
5	AFSDETIE	RW	帧同步提前检测中断使能： 1: 使能中断； 0: 禁止中断。	0
4	CNRDYIE	RW	编解码器未就绪中断使能： 1: 使能中断； 0: 禁止中断。	0
3	FREQIE	RW	FIFO 请求中断使能： 1: 使能中断； 0: 禁止中断。	0
2	WCKCFGIE	RW	时钟配置错误中断使能：	0

			1: 使能中断; 0: 禁止中断。	
1	MUTEDETIE	RW	静音检测中断使能: 1: 使能中断; 0: 禁止中断。	0
0	OVRUDRIE	RW	上溢/下溢中断使能: 1: 使能中断; 0: 禁止中断。	0

36.4.6 SAI 模块 x 状态寄存器 (R32_SAI_xSR) (x=A/B)

偏移地址: 0x18, 0x38

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

位	名称	访问	描述	复位值
[31:19]	Reserved	R0	保留。	0
[18:16]	FLTH[2:0]	R0	FIFO 阈值标志: 如果将 SAI 模块配置为发送器: 000: FIFO 为空; 001: FIFO ≤ 1/4, 但非空; 010: 1/4 < FIFO ≤ 1/2; 011: 1/2 < FIFO ≤ 3/4; 100: 3/4 < FIFO, 但未满; 101: FIFO 已满; 其他: 保留。 如果将 SAI 模块配置为接收器: 000: FIFO 为空; 001: FIFO < 1/4, 但非空; 010: 1/4 ≤ FIFO < 1/2; 011: 1/2 ≤ FIFO < 3/4; 100: 3/4 ≤ FIFO, 但未满; 101: FIFO 已满; 其他: 保留。	0
[15:7]	Reserved	R0	保留。	0
6	LFSDET	RW1Z	帧同步滞后检测: 1: 帧同步信号未在正确的时刻出现; 0: 无错误。 注: 写 1 清 0, 写 0 无效。	0
5	AFSDET	RW1Z	帧同步提前检测: 1: 提前检测到帧同步信号;	0

			0: 无错误。 注: 写 1 清 0, 写 0 无效。	
4	CNRDY	RW1Z	编解码器未就绪: 1: 外部 AC' 97 编解码器未就绪; 0: 外部 AC' 97 编解码器已就绪。 注: (1) 仅当在 R32_SAI_xCFGR1 寄存器中选择了 AC' 97 音频模块并且音频模块配置为接收器模式时, 才使用该位。 (2) 写 1 清 0, 写 0 无效。	0
3	FREQ	RW1Z	FIFO 请求标志: 1: FIFO 请求读取或写入 R32_SAI_xDATAR; 0: 无 FIFO 请求。 请求内容取决于音频模块的配置: - 如果模块配置为发送模式, 则 FIFO 请求与向 R32_SAI_xDATAR 中写入相关。 - 如果音频模块配置为接收模式, 则 FIFO 请求与从 R32_SAI_xDATAR 中读取相关。 注: 写 1 清 0, 写 0 无效。	1
2	WCKCFG	RW1Z	时钟配置错误标志: 1: 时钟配置不符合帧同步中定义的帧长度规范 (R32_SAI_xFRCR 寄存器 FRL[7:0] 位的配置); 0: 时钟配置正确。 注: 写 1 清 0, 写 0 无效。	0
1	MUTEDET	RW1Z	静音检测标志: 1: 在 SD 输入线上检测到指定数量的连续音频帧中的 MUTE 值 (0 值); 0: SD 输入线上未检测到 MUTE 值。 注: 写 1 清 0, 写 0 无效。	0
0	OVRUDR	RW1Z	上溢/下溢错误标志: 1: 检测到上溢/下溢错误; 0: 无上溢/下溢错误。 注: 写 1 清 0, 写 0 无效。	0

36.4.7 SAI 模块 x 数据寄存器 (R32_SAI_xDATAR) (x=A/B)

偏移地址: 0x20, 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
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[31:0]	RW	数据: 若 FIFO 未滿, 写入该寄存器可向 FIFO 加载数据。 若 FIFO 非空, 读取该寄存器可清空 FIFO。	0

第 37 章 QSPI 接口 (QuadSPI)

本章模块描述仅适用于 CH32H417、CH32H416 微控制器产品。

芯片内置两组专用的 QSPI 通信接口 (QuadSPI)，连接单、双或四 (条数据线) SPI 的 FLASH 存储介质。该接口的工作模式主要包括以下几种：

- 间接模式：在这种模式下，QSPI 接口通过使用 QSPI 寄存器执行所有操作，该模式适用于需要频繁读写操作的应用场景。
- 状态轮询模式：在这种模式下，系统会周期性地读取外部 FLASH 状态寄存器。当标志位置 1 时（如擦除或烧写完成），系统会产生中断。该模式适用于需要实时监控外部 Flash 状态的应用场景。

内存映射模式：在这种模式下，外部 Flash 被映射到微控制器的地址空间中，系统将其视为内部存储器进行访问。该内存映射的地址只能用于数据访问。

采用双闪存模式时，系统可以同时访问两个 Quad-SPI FLASH，吞吐量和容量均可提高二倍。这种模式适用于需要大量数据处理的应用场景。

37.1 主要特征

- 三种功能模式：间接模式、状态轮询模式和内存映射模式
- 双闪存模式，通过并行访问两个 FLASH
- 集成 FIFO，用于发送和接收
- 支持 SDR 模式
- 针对间接模式和内存映射模式，完全可编程帧格式
- 针对间接模式和内存映射模式，完全可编程操作码
- 在达到 FIFO 阈值和传输完成时生成 DMA 触发信号
- 在达到 FIFO 阈值、超时、操作完成以及发生访问错误时产生中断

37.2 概述

图 37-1 QSPI 功能框图（双闪存模式禁止）

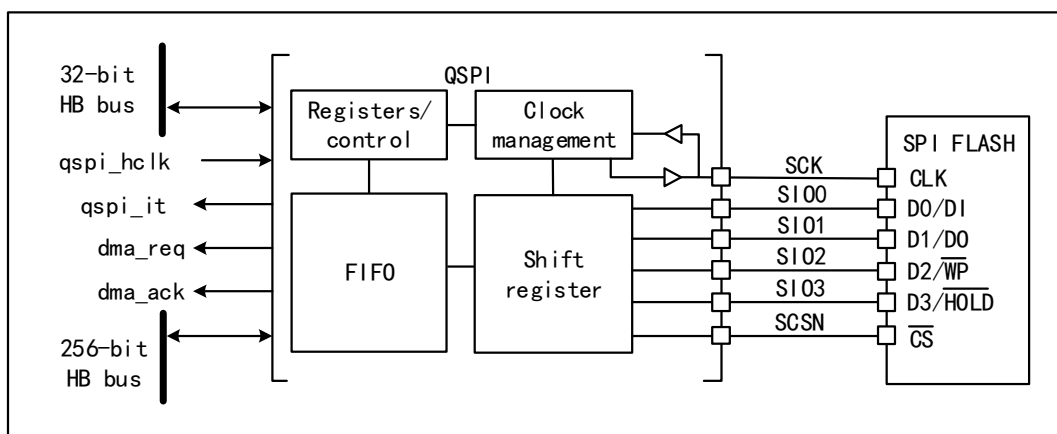
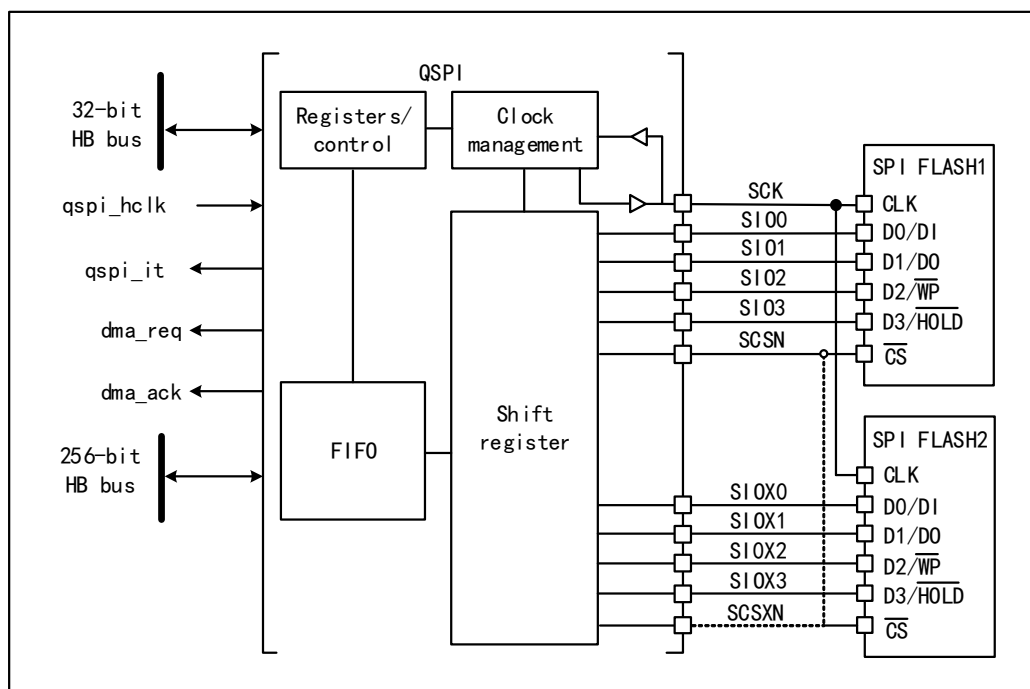


图 37-2 QSPI 功能框图（双闪存模式使能）

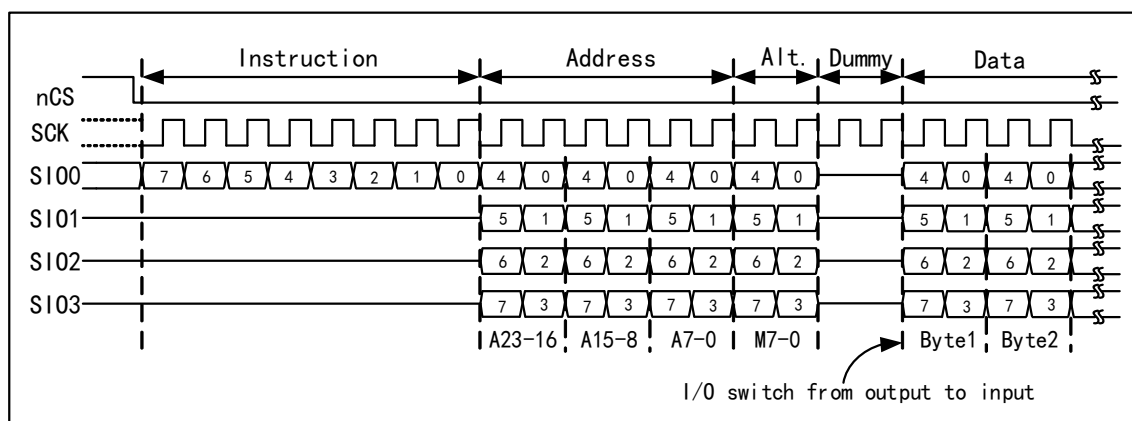


37.3 功能描述

37.3.1 QSPI 命令序列

QSPI 通过命令与 FLASH 通信，每条命令包括五个阶段：指令、地址、交替字节、空指令和数据。任一阶段均可跳过，但至少需保留指令、地址、交替字节或数据阶段中的一个。nCS 在每条指令开始前下降，在每条指令完成后再次上升。

图 37-3 四线模式下的读命令示例



指令阶段

在指令阶段，通过在 R32_QSPIx_CCR 寄存器的 INSTRUCTION[7:0] 字段中配置的一条 8 位指令发送到 FLASH，实现对待执行操作的类型的指定。大部分 FLASH 仅能通过 SIO0 信号（单线 SPI 模式）以每次 1 位的方式接收指令。然而，通过配置 R32_QSPIx_CCR 寄存器中的 IMODE[1:0] 字段，可以实现指令阶段的选择性发送，即一次发送 2 位（在双线 SPI 模式中通过 SIO0/SIO1）或一次发送 4 位（在四线 SPI 模式中通过 SIO0/SIO1/SIO2/SIO3）。若 IMODE=00b，则指令阶段被跳过，命令序列从地址阶段（若存在）开始。

地址阶段

在地址阶段，为指示操作地址，可将 1-4 字节发送到 FLASH。在 R32_QSPIx_CCR 寄存器的 ADSize[1:0] 字段中配置待发送的地址字节数。在间接模式和自动轮询模式下，待发送的地址字节在 R32_QSPIx_AR 寄存器的 ADDRESS[31:0] 中指定。而在内存映射模式下，地址则是由系统直接提供。

通过配置 R32_QSPIx_CCR 寄存器中的 ADMODE[1:0] 字段，地址阶段可实现一次发送 1 位（在单线 SPI 模式中通过 SI00）、2 位（在双线 SPI 模式中通过 SI00/SI01）或 4 位（在四线 SPI 模式中通过 SI00/SI01/SI02/SI03）。若 ADMODE=00b，则地址阶段被跳过，命令序列直接进入下一阶段（若存在）。

交替字节阶段

在交替字节阶段，1-4 字节数据被发送到 FLASH，通常用于控制操作模式。在 R32_QSPIx_CCR 寄存器的 ABSIZE[1:0] 字段中配置待发送的交替字节数。待发送字节的指定在 R32_QSPIx_ABR 寄存器中完成。

通过 R32_QSPIx_CCR 寄存器中的 ABMODE[1:0] 字段进行配置，交替字节阶段可一次发送 1 位（在单线 SPI 模式中通过 SI00）、2 位（在双线 SPI 模式中通过 SI00/SI01）或 4 位（在四线 SPI 模式中通过 SI00/SI01/SI02/SI03）。若 ABMODE=00b，则交替字节阶段被跳过，命令序列直接进入下一阶段（若存在）。

在交替字节阶段，有时只需发送单个半字节而非一个全字节。例如：在双线模式下，仅使用两个周期发送交替字节时。此时，固件可采用四线模式（ABMODE=11b）并发送一个字节，方法是 ALTERNATE 的位[7]和[3]置 1（SI03 保持高电平）且位[6]和[2]置 0（SI02 线保持低电平）。此时，半字节的高 2 位存放在 ALTERNATE 的位[4:3]，低 2 位存放在位[1]和[0]中。例如，如果半字节 2（0010）通过 SI00/SI01 发送，则 ALTERNATE 应设置为 0x8A（1000_1010）。

空指令周期阶段

在空指令周期阶段，当使用更高的时钟频率时，为确保 FLASH 有足够的时间准备数据，需在给定的 1-31 个周期内不进行任何数据的发送或接收。该阶段的周期数由 R32_QSPIx_CCR[22:18] 寄存器的 DCYC[4:0] 字段指定。

在 SDR 模式下，持续时间由一定个数的全时钟周期决定。若 DCYC 为 0，则跳过空指令周期阶段，命令序列直接进入数据阶段（若存在）。

空指令周期阶段的操作模式固定为单线模式。

数据阶段

在数据阶段，可实现从 FLASH 接收或向其发送任意数量的字节。在间接模式和自动轮询模式下，在 R32_QSPIx_DLR 寄存器中指定待接收/发送的字节数。在间接写入模式下，需在 R32_QSPIx_DR 寄存器中写入需发送至 FLASH 的数据。在间接读取模式下，从 FLASH 接收的数据通过读取 R32_QSPIx_DR 寄存器获得。

在内存映射模式下，读取的数据直接送回到系统。

通过配置 R32_QSPIx_CCR 寄存器中的 DMODE[1:0] 字段，数据阶段可一次发送/接收 1 位（在单线 SPI 模式中通过 SI00）、2 位（在双线 SPI 模式中通过 SI00/SI01）或 4 位（在四线 SPI 模式中通过 SI00/SI01/SI02/SI03）。若 DMODE=00b，则数据阶段被跳过，命令序列在拉高 nCS 时立即完成。这一配置仅适用于间接写入模式。

37.3.2 QSPI 信号接口协议模式

单线 SPI 模式

传统 SPI 模式允许串行发送/接收单独的 1 位。在此模式下，数据通过 SI00 发送到 FLASH。从 FLASH 接收到的数据通过 SI01 送达。

为使用此单个位模式，可通过将 R32_QSPIx_CCR 中的 IMODE/ADMODE/ABMODE/DMODE 字段设置为 01b，对不同的命令阶段分别进行配置。

在每个已配置为单线模式的阶段中：

- SI00 处于输出模式
- SI01 处于输入模式（高阻抗）

- SI02 处于输出模式并强制置 0（以禁用“写保护”功能）
- SI03 处于输出模式并强制置 1（以禁用“保持”功能）

若 DMODE=01b，这对于空指令阶段也如上所示。

双线 SPI 模式

在双线模式下，通过 SI00/SI01 信号同时发送/接收两位。通过将 R32_QSPIx_CCR 寄存器的 IMODE/ADMODE/ABMODE/DMODE 字段配置为 10b，可以对不同的命令阶段进行相应的设置。

在每个已配置为双线模式的阶段中：

- SI00/SI01 在数据阶段进行读取操作时处于高阻态（输入），在其他情况下为输出
- SI02 处于输出模式并强制置 0
- SI03 处于输出模式并强制置 1

在空指令阶段，若 DMODE=01b，则 SI00/SI01 始终保持高阻态。

四线 SPI 模式

在四线模式下，通过 SI00/SI01/SI02/SI03 信号同时发送/接收四位数据。通过将 R32_QSPIx_CCR 寄存器的 IMODE/ADMODE/ABMODE/DMODE 字段配置为 11b，可以对不同的命令阶段进行相应的设置。

在每个已配置为四线模式的阶段中：

- SI00/SI01/SI02/SI03 在数据阶段进行读取操作时均处于高阻态（输入），在其他情况下为输出
- SI02 和 SI03 仅用于 QSPI 模式。若未配置任何阶段使用四线 SPI 模式，即使 QSPI 激活，对应 SI02 和 SI03 的引脚也可用于其他功能

在空指令阶段中，若 DMODE=11，则 SI00/SI01/SI02/SI03 均为高阻态

SDR 模式

QSPI 在单倍数据速率（SDR）模式下工作。

在 SDR 模式下，当 QSPI 驱动 SI00、SI01、SI02、SI03 信号时，这些信号仅在 SCK 的下降沿发生转变。

在 SDR 模式下，接收数据时，QSPI 假定 FLASH 通过 SCK 的下降沿发送数据，使用 SCK 后续的边沿（上升沿）对信号进行采样。

双闪存模式

当 DFM 位（R32_QSPIx_CR[6]）被设置为 1 时，QSPI 进入双闪存模式。QSPI 使用两个外部四线 SPIFLASH（FLASH1 和 FLASH2），每个周期中发送/接收 8 位数据，有效地将吞吐量和容量翻倍。每个 FLASH 使用同一个 SCK 并可选择使用同一个 nCS 信号，但其 SI00、SI01、SI02 和 SI03 信号是各自独立的。

双闪存模式可与单比特模式、双比特模式、四比特模式以及 SDR 模式结合使用。

在 FSIZE[4:0]（R32_QSPIx_DCR[20:16]）中指定 FLASH 的大小，指定的参数应能够反映 FLASH 的总容量，即单个组件容量的 2 倍。当地址 X 为偶数时，QSPI 赋给地址 X 的字节是存放于 FLASH1 的地址 X/2 中的字节，QSPI 赋给地址 X+1 的字节是存放于 FLASH2 的地址 X/2 中的字节。即偶地址中的字节存储于 FLASH1，奇地址中的字节存储于 FLASH2。

在双闪存模式下，读取 FLASH 状态寄存器时，需要读取的字节数是单闪存模式下的 2 倍。即每个 FLASH 执行状态寄存器的获取指令后能提供 8 个有效位，那么 QSPI 必须配置为 2 个字节（16 位）的数据长度，并且 QSPI 从每个 FLASH 接收 1 个字节。如果每个 FLASH 提供一个 16 位的状态信息，则 QSPI 必须配置为读取 4 字节，以便在双闪存模式下可获取两个 FLASH 的所有状态位。结果数据寄存器中的最低有效字节是 FLASH1 状态寄存器的最低有效字节，而下一个字节是 FLASH2 状态寄存器的最低有效字节。数据寄存器的第三个字节是 FLASH1 的第二个字节，第四个字节是 FLASH2 的第二个字节（在 FLASH 具有 16 位状态寄存器的情况下）。

在双闪存模式下，必须始终访问偶数个字节。因此，当 DRM=1 时，数据长度字段（R32_QSPIx_DLR[0]）的位 0 始终保持为 1。

在双闪存模式下，FLASH1 接口信号的行为与正常模式下基本相同。在指令、地址、交替字节以及空指令周期阶段，FLASH2 接口信号与 FLASH1 接口信号的波形完全相同。即每个 FLASH 总是接收相同

的指令与地址。然后，在数据阶段，SIOx 和 SIOx 总线并行传输数据，但发送到 FLASH1（或从其接收）的数据与 FLASH2 中的不同。

37.3.3 QSPI 间接模式

在间接模式下，通过向 QSPI 寄存器写入来触发命令；数据通过读写数据寄存器进行传输，与其他通信外设的方式相同。

当 FMODE=00b（R32_QSPIx_CCR[27:26]）时，QSPI 处于间接写入模式，在数据阶段中将字节发送到 FLASH。数据通过写入数据寄存器（R32_QSPIx_DR）的方式来提供；当 FMODE=01b 时，QSPI 处于间接读取模式，在数据阶段中将字节从 FLASH 接收。数据通过读取数据寄存器（R32_QSPIx_DR）的方式来获取。

在数据长度寄存器（R32_QSPIx_DLR）中指定读取/写入的字节数。若 R32_QSPIx_DLR=0xFFFF_FFFF（全置 1），则数据长度视为未定义，QSPI 将继续传输数据，直到到达（由 FSIZE 定义的）FLASH 的结尾。若不传输任何字节，R32_QSPIx_CCR 寄存器 DMODE 位设置为 00b。若 R32_QSPIx_DLR=0xFFFF_FFFF 并且 FSIZE=0x1F（数据长度为 4G bytes），接模式访问 flash 时，配置应保证 flash address + DLR ≤ FSIZE，则传输将无限继续下去，仅在 QSPI 被禁止或出现终止请求后停止。在读取完最后一个存储器地址（地址为 0xFFFF_FFFF）后，将从地址=0x0000_0000 开始继续读取。

当达到编程设定要发送或接收的字节数时，若 TCIE=1，则 TCF 置 1 并产生中断。当数据数量不确定时，系统会根据 R32_QSPIx_CR 寄存器中定义的 FLASH 大小，在达到外部 SPI 的限制时，TCF 同样会被置 1。

QSPI 支持写流控，即当 DLR 所填长度未发完，同时 FIFO 为空时，暂停发送移位寄存器；在此模式下，出于效率的考虑，软件应当先写 DR 寄存器往 FIFO 填数据，或置 DMAEN 为 1，自动向 DMA 请求到若干数据后，再写 START 位开始发送。

触发命令启动

本质上，一旦固件提供了命令所需的最后信息，命令即会启动。根据 QSPI 的配置，在间接模式下，触发命令启动的三种方式如下：

- 1) 对 R32_QSPIx_CCR 寄存器 INSTRUCTION[7:0] 执行写入操作，如果没有地址是必需的（当 ADMODE=00b）并且无需固件提供数据（当 FMODE=01b 或 DMODE=00b）；
- 2) 对 R32_QSPIx_AR 寄存器 ADDRESS[31:0] 执行写入操作，如果地址是必需的（当 ADMODE=00b）并且无需固件提供数据（当 FMODE=01b 或 DMODE=00b）；
- 3) 对 R32_QSPIx_DR 寄存器 DATA[31:0] 执行写入操作，如果地址是必需的（当 ADMODE!=00b）并且需要固件提供数据（当 FMODE=00b 并且 DMODE!=00b）。

写入交替字节寄存器（R32_QSPIx_ABR）始终不会触发命令启动。如果需要交替字节，必须预先进行编程。一旦命令启动，R32_QSPIx_SR 寄存器 BUSY 位将自动置 1。

FIFO 和数据管理

在间接模式下，数据将通过 QSPI 内部的 32 字节 FIFO 传输。R32_QSPIx_SR 寄存器 FLEVEL[5:0] 指示 FIFO 当前存储了多少字节。

在间接写入模式下（FMODE=00b），当固件向 R32_QSPIx_DR 寄存器写入数据时，数据将被加入到 FIFO 中。字写入将在 FIFO 中增加 4 个字节，半字写入增加 2 个字节，而字节写入仅增加 1 个字节。若固件在 FIFO 中加入的数据超过了 DL[31:0] 所指示的值，那么在写操作完成时（TCF 置 1），超出的字节将从 FIFO 中清除。对 R32_QSPIx_DR 寄存器的字节/半字访问必须严格限于该 32 位寄存器的最低有效字节/半字。

FTHRES[4:0] 用于定义 FIFO 的阈值。当达到阈值时，FTF（FIFO 阈值标志）置 1。在间接读取模式下，在 FIFO 中读取的有效字节数超过阈值时，FTF 置 1。从 FLASH 中读取最后一个字节后，若 FIFO 中仍有数据，则无论 FTHRES 的设置为多少，FTF 都会置 1。在间接写入模式下，当 FIFO 中的空字节数超过阈值时，FTF 置 1。若 FTIE=1，则 FTF 置 1 时产生中断。一旦不再满足阈值条件，即 CPU 或 DMA 传输了足够的数据后，那么 HW 将清零 FTF。

在间接模式下，QSPI 在 FIFO 已满时将暂停从 Flash 读取字节，以避免上溢。只有当 FIFO 中的 4 个字节为空（FLEVEL≤11）时，才会恢复读取 FLASH。因此，若 FTHRES≥13，应用程序必须读取足够数量的字节，以确保 QSPI 能够继续从 FLASH 中检索数据。反之，只要 11<FLEVEL<FTHRES，FTF 标志将始终为 0。

37.3.4 QSPI 状态标志轮询模式

在自动轮询模式下，QSPI 周期性启动命令以读取一定数量的状态字节（最多 4 个）。通过屏蔽接收的字节，可以隔离一些状态位。当选定的位具有预设的值时，可产生中断。

对 FLASH 的访问开始方式与在间接读取模式相同：如果不需要地址（AMODE=00b），则在写入 R32_QSPIx_CCR 时即开始访问。反之，若需要地址，首次访问将在写入 R32_QSPIx_AR 时开始。此时 BUSY 变为高电平，并且即使在周期性访问期间也保持不变。

在自动轮询模式下，R32_QSPIx_PSMAR 寄存器 MASK[31:0] 的内容用于屏蔽来自 FLASH 的数据。如果 MASK[n]=0，则结果中的位 n 被屏蔽，不予以考虑。若 MASK[n]=1，并且位 [n] 的内容与 R32_QSPIx_PSMAR 寄存器 MATCH[n] 相同，说明位 n 存在匹配。

若轮询匹配模式位（PMM，R32_QSPIx_CR 的位 23）为 0，将激活“AND”匹配模式，即只有在所有未屏蔽的位都匹配时，状态匹配标志（SMF）才会被置 1；如果 PMM=1，则激活“OR”匹配模式，即若在任意未屏蔽位上匹配成功，SMF 置 1；如果 SMIE=1，则在 SMF 置 1 时会调用一个中断；如果自动轮询模式停止，即 APMS 位置 1，则操作停止，并且在检测到匹配时 BUSY 位清零。反之，BUSY 位保持为 1，继续进行周期性访问，直到发生中止或禁止 QSPI（EN=0）。

数据寄存器（R32_QSPIx_DR）包含最新接收的状态字节（FIFO 停用）。数据寄存器的内容不受匹配逻辑所用屏蔽方法的影响。FTF 状态位在新一次状态读取完成后置 1，一旦读取数据，FTF 清零。

37.3.5 QSPI 内存映射模式

当配置为内存映射模式时，外部 SPI 器件被视为是内部存储器。若未对 QSPI 外设进行正确配置与使能，将无法访问 QSPI Flash 的存储区域。即便 FLASH 容量更大，其寻址空间亦受限于 256MB 的上限。

注意：内存映射的地址只能用于数据访问，不能用于指令访问。

支持字节、半字和字访问类型。QSPI 接受下一个微控制器访问并提前加载后面地址中的字节。如果之后访问的是连续地址，由于数据已经预取，访问过程得以加速完成。

在默认情况下，QSPI 从不中断其预取操作，即使长时间未对 FLASH 进行访问，也会保持之前的读取操作，且 nCS 将维持在低电平。然而，当 nCS 保持低电平时，FLASH 功耗增加。因此，应用程序可能会激活超时计数器（TCEN=1，R32_QSPIx_CR 的位 3）。从而在 FIFO 中写满预取的数据后，如果没有在 TIMEOUT[15:0]（R32_QSPIx_LPTR）个周期的时长内访问，则释放 nCS。一旦发生第一个存储器映射访问，BUSY 信号将变为高电平。由于预取操作的存在，BUSY 信号仅在发生超时、中止或外设禁用后才会下降。

37.3.6 QSPI FLASH 配置

设备配置寄存器（R32_QSPIx_DCR）被用于定义外部 SPI FLASH 的特性参数。其中，FSIZE[4:0] 字段依据以下公式确定外部存储器的容量：

$$\text{FLASH 中的字节数} = 2^{[FSIZE+1]}$$

FSIZE+1 是对 FLASH 寻址所需的地址位数。在间接模式下，FLASH 容量最高可达 4GB（采用 32 位进行寻址），但在内存映射模式下的可寻址空间限制为 256MB。若 DFM=1，FSIZE 表示两个 FLASH 容量的总和。

当 QSPI 连续执行两条命令时，它在两条命令之间将片选信号（nCS）置为高电平，默认仅一个 SCK 周期时长。如果 FLASH 需要命令之间的时间更长，可使用片选高电平时间（CSHT）字段指定 nCS 必须保持高电平的最少 SCK 周期数（最大为 8）。时钟模式（CKMODE）位指示在命令之间（当 nCS=1 时）

SCK 信号的逻辑电平。

37.3.7 QSPI 配置

QSPI 配置分两个阶段：QSPI IP 配置、QSPI FLASH 配置。QSPI 一旦配置并使能，即可使用其三种操作模式之一：间接模式、状态轮询模式和内存映射模式。

QSPI IP 配置：通过 R32_QSPIx_CR 配置。用户必须配置传入数据的时钟预分频器的分频系数以及采样移位设置。生成 DMA 触发信号或生成中断的 FIFO 电平在 FTHRES 位中进行编程。如果需要超时计数器，则可将 TCEN 位置 1 并在 R32_QSPIx_LPTR 寄存器中编程超时值。双闪存模式可通过将 DFM 置 1 来激活。

QSPI FLASH 配置：通过 R32_QSPIx_DCR 寄存器配置与外部目标 FLASH 相关的参数。用户必须在 FSIZE 位中编程 FLASH 的大小、在 CSHT 位中编程片选保持高电平的最短时间，并在 MODE 位中设置功能模式（模式 0 或模式 3）。

37.3.8 QSPI 的用法

通过 R32_QSPIx_CCR 寄存器 FMODE[1:0] 选择操作模式。

间接模式

当 FMODE=00b 时，选择间接写入模式，可以将数据发送到 FLASH。当 FMODE=01b 时，选择间接读取模式，读取 FLASH 中的数据。

QSPI 用于间接模式时，采用以下方式构建帧：

- 1) 在 R32_QSPIx_AR 寄存器中指定目标地址；
- 2) 在 R32_QSPIx_CR 寄存器中指定工作模式；
- 3) 在 R32_QSPIx_DLR 寄存器中指定待读取或写入的字节数；
- 4) 在 R32_QSPIx_CCR 寄存器中指定帧格式、模式和指令代码；
- 5) 通过 R32_QSPIx_DR 寄存器从 FIFO 读取数据/向 FIFO 写入数据；
- 6) 在 R32_QSPIx_ABR 寄存器中指定要在地址阶段后立即发送的可选交替字节。

在写入控制寄存器（R32_QSPIx_CR）时，用户可指定以下设置：

- 使能位（EN）设置为 1
- 超时计数器使能位（TCEN）
- FIFO 阈值（FTHRES），以指示 FTF 标志在何时置 1
- 中断使能
- 自动轮询模式参数：匹配模式和停止模式（在 FMODE=11b 时有效）
- 时钟预分频器

在写入通信配置寄存器（R32_QSPIx_CCR）时，用户指定以下参数：

- 通过 DCYC 位指定空指令字节数
- 通过 DBMODE 位指定是否存在空指令字节
- 通过 INSTRUCTION 位指定指令字节
- 通过 ABSIZE 位指定交替字节数（1/2/3/4）
- 通过 IMODE 位指定指令发送方式（1/2/4 线）
- 通过 ADSIZE 位指定地址长度（8/16/24/32 位）
- 通过 ADMODE 位指定地址发送方式（无/1/2/4 线）
- 通过 ABMODE 位指定交替字节发送方式（无/1/2/4 线）
- 通过 DMODE 位指定数据发送/接收方式（无/1/2/4 线）

若不需要为某个命令更新数据寄存器（R32_QSPIx_DR）和地址寄存器（R32_QSPIx_AR），那么在写入 R32_QSPIx_CCR 寄存器时，该命令序列便立即启动。在 ADMODE 和 DMODE 都为 00b 时，或在间接读取模式（FMODE=01b）下仅 ADMODE=00b 时，便属于此情况。当需要地址（ADMODE!=00）且数据寄存器不需要写入（FMODE=01b 或 DMODE=00b）时，一旦通过写入 R32_QSPIx_AR 寄存器更新地址，命令序

列便立即启动。在数据传输（FMODE=00b 并且 DMODE!=00b）的情况下，通过 R32_QSPIx_DR 寄存器向 FIFO 写入会触发通信启动。

状态标志轮询模式

使能状态标志轮询模式：将 R32_QSPIx_CCR 寄存器 FMODE 字段设置为 10b。在此模式下，会发送编程的帧并周期性检索数据。每帧中读取的最大数据量为 4 字节。若 R32_QSPIx_DLR 寄存器请求的数据量超过 4 个字节，则会被忽略，只读取 4 字节。周期性由 R32_QSPIx_PISR 寄存器指定。

在检索到状态数据后，可在内部进行处理，以达到以下目的：

- 自动停止周期性检索状态字节
- 将状态匹配标志位置 1，并在使能时生成中断

可通过存储于 R32_QSPIx_PSMKR 寄存器中的值将接收到的值进行屏蔽，并与存储在 R32_QSPIx_PSMAR 寄存器中的值进行或运算、与运算。若存在匹配，则状态匹配标志置 1，并且在使能了中断的情况下还将产生中断；如果 AMPS 位置 1，则 QSPI 自动停止。在所有情况下，最新的检索值都在 R32_QSPIx_DR 寄存器中可用。

内存映射模式

在内存映射模式下，尽管存在访问延迟，外部 FLASH 仍被视为内部存储器。在此模式下，仅支持对外部 FLASH 执行读取操作。通过将 R32_QSPIx_CCR 寄存器中的 FMODE 设置为 11b，可以激活内存映射模式。当主器件访问存储器映射空间时，将发送已配置的帧和指令。

FIFO 用作预取缓冲区，以支持连续的线性读取操作。在此模式中，任何对于 R32_QSPIx_DR 寄存器的访问均返回零。在内存映射模式中，数据长度寄存器（R32_QSPIx_DLR）无效。

37.3.9 指令仅发送一次

某些 FLASH（例如 Winbound）能够提供一种模式，在该模式下，必须仅用第一个命令序列发送指令，而后续的命令根据地址直接执行。用户通过使用 R32_QSPIx_CCR 寄存器 S100 位来实现这一特性。

S100 位适用于所有的功能模式：间接模式、状态轮询模式和内存映射模式。当 S100 位置 1，仅第一条命令发送指令，且对 R32_QSPIx_CCR 寄存器执行写操作。后续命令序列都将跳过指令阶段，直到 R32_QSPIx_CCR 寄存器被再次写入。然而，在 IMODE=00b（无指令）时，S100 位无效。

37.3.10 QSPI 差错管理

可能产生错误的情况有：

- 在间接模式或状态标志轮询模式下，若在 R32_QSPIx_AR 寄存器中设置了错误的地址，即根据 R32_QSPIx_DCR 寄存器中 FSIZE[4:0]定义的 FLASH 大小：TEF 将置 1，如果使能，还将产生中断。
- 在间接模式下，若地址加数据的长度超过 FLASH 的大小，TEF 将在访问被触发时置 1。

37.3.11 QSPI 的繁忙位和中止功能

当 QSPI 开始对 FLASH 进行操作时，R32_QSPIx_SR 寄存器中的 BUSY 位自动置 1。在间接模式下，QSPI 完成了请求的命令序列，并且 FIFO 为空时，BUSY 位复位；在自动轮询模式下，只有在最后一次周期性访问完成后，由于 APMS=1 时发生匹配，或者由于中止操作，BUSY 位才变为低电平。在内存映射模式下，首次访问后，BUSY 位只有在发生超时事件或中止时变为低电平。

通过将 R32_QSPIx_CR 寄存器中的 ABORT 位置 1 可将任何操作中中止。一旦中止完成，BUSY 位和 ABORT 位自动复位，FIFO 清空。

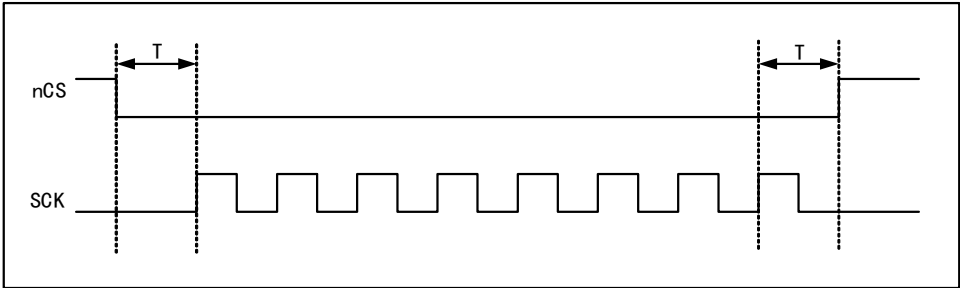
注：若中止对状态寄存器的写操作，有些 FLASH 可能发生异常行为。

37.3.12 nCS 行为

默认情况下，nCS 为高电平，取消选择外部 FLASH。在操作开始前 nCS 会下降，并在操作完成后立即上升。

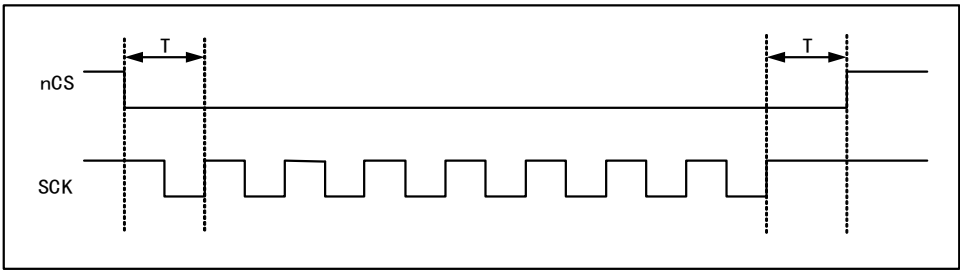
当 CKMODE=0（“模式 0”，即在没有进行操作时 SCK 保持低电平），nCS 在操作首次升高 SCK 边沿时的一个 SCK 周期前降低至低电平，在操作最后一次升高 SCK 边沿时的一个 SCK 周期后升高至高电平，如下图所示。

图 37-4 CKMODE=0 时的 nCS（T=SCK 周期）



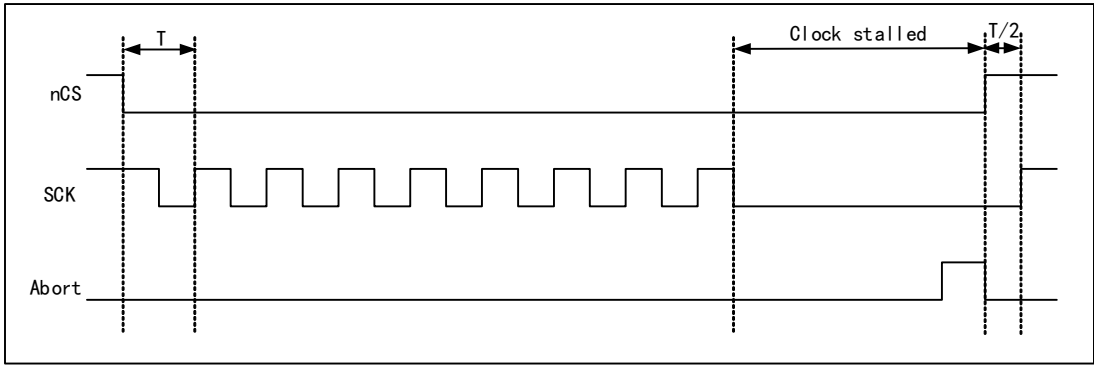
当 CKMODE=1（“模式 3”，在未进行任何操作时 SCK 升高至高电平）时，nCS 仍在操作首次升高 SCK 边沿时的一个 SCK 周期前降低至低电平，在操作最后一次升高 SCK 边沿时的一个 SCK 周期后升高至高电平，如下图所示。

图 37-5 CKMODE=1 时的 nCS（T=SCK 周期）



若 FIFO 在读取操作中保持写满状态或在写入操作中保持为空，则在固件干预 FIFO 前，操作停止并且 SCK 保持低电平。若操作停止时发生中止，则 nCS 在请求中止后即升高至高电平，SCK 则在半个周期后升高至高电平，如下图所示。

图 37-6 CKMODE=1 且发生中止时的 nCS（T=SCK 周期）



当不处于双闪存模式（DFM=0）且 FSEL=0（默认值）时，仅访问 FLASH1。因此，若 FSEL=1，仅访问 FLASH1，nCS 保持高电平。在双闪存模式下，FLASH1 和 FLASH2 的 nCS 行为完全相同。因此，如果存在 FLASH2，并且应用程序始终处于双闪存模式，那么 FLASH1 的 nCS 信号也可以用于 FLASH2，而 FLASH2 的 nCS 引脚输出可用于其他功能。

37.3.13 中断

表 37-1 中断请求

中断事件	事件标志	使能控制位

超时	TOF	TOIE
状态匹配	SMF	SMIE
FIFO 阈值	FTF	FTIE
传输完成	TCF	TCIE
传输错误	TEF	TEIE

37.4 寄存器描述

表 37-2 QSPI1 相关寄存器列表

名称	访问地址	描述	复位值
R32_QSPI1_CR	0x40024C00	QSPI1 控制寄存器	0x00000000
R32_QSPI1_DCR	0x40024C04	QSPI1 器件配置寄存器	0x00000000
R32_QSPI1_SR	0x40024C08	QSPI1 状态寄存器	0x00000000
R32_QSPI1_FCR	0x40024C0C	QSPI1 标志清零寄存器	0x00000000
R32_QSPI1_DLR	0x40024C10	QSPI1 数据长度寄存器	0x00000000
R32_QSPI1_CCR	0x40024C14	QSPI1 通信配置寄存器	0x00000000
R32_QSPI1_AR	0x40024C18	QSPI1 地址寄存器	0x00000000
R32_QSPI1_ABR	0x40024C1C	QSPI1 交替字节寄存器	0x00000000
R32_QSPI1_DR	0x40024C20	QSPI1 数据寄存器	0x00000000
R32_QSPI1_PSMKR	0x40024C24	QSPI1 轮询状态屏蔽寄存器	0x00000000
R32_QSPI1_PSMAR	0x40024C28	QSPI1 轮询状态匹配寄存器	0x00000000
R32_QSPI1_PIR	0x40024C2C	QSPI1 轮询间隔寄存器	0x00000000
R32_QSPI1_LPTR	0x40024C30	QSPI1 低功耗超时寄存器	0x00000000

表 37-3 QSPI2 相关寄存器列表

名称	访问地址	描述	复位值
R32_QSPI2_CR	0x40025000	QSPI2 控制寄存器	0x00000000
R32_QSPI2_DCR	0x40025004	QSPI2 器件配置寄存器	0x00000000
R32_QSPI2_SR	0x40025008	QSPI2 状态寄存器	0x00000000
R32_QSPI2_FCR	0x4002500C	QSPI2 标志清零寄存器	0x00000000
R32_QSPI2_DLR	0x40025010	QSPI2 数据长度寄存器	0x00000000
R32_QSPI2_CCR	0x40025014	QSPI2 通信配置寄存器	0x00000000
R32_QSPI2_AR	0x40025018	QSPI2 地址寄存器	0x00000000
R32_QSPI2_ABR	0x4002501C	QSPI2 交替字节寄存器	0x00000000
R32_QSPI2_DR	0x40025020	QSPI2 数据寄存器	0x00000000
R32_QSPI2_PSMKR	0x40025024	QSPI2 轮询状态屏蔽寄存器	0x00000000
R32_QSPI2_PSMAR	0x40025028	QSPI2 轮询状态匹配寄存器	0x00000000
R32_QSPI2_PIR	0x4002502C	QSPI2 轮询间隔寄存器	0x00000000
R32_QSPI2_LPTR	0x40025030	QSPI2 低功耗超时寄存器	0x00000000

37.4.1 QSPI 控制寄存器 (R32_QSPIx_CR) (x=1/2)

偏移地址: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESCALER[7:0]								PMM	APMS	Reserved	TOIE	SMIE	FTIE	TCIE	TEIE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SIO XEN	FTHRES[4:0]					FSEL	DFM	START	Reser ved	TCEN	DMAEN	ABORT	EN

位	名称	访问	描述	复位值
[31:24]	PRESCALER[7:0]	RW	时钟预分频器： 该字段定义基于 HB 总线时钟生成 SCK 所用的分频系数（值+1）。 0x0: $F_{SCK}=F_{HCLK}$，HCLK 时钟直接用作 QSPI SCK（预分频器被旁路）； 0x1: $F_{SCK}=F_{HCLK}/2$； 0x2: $F_{SCK}=F_{HCLK}/3$； ... 0xFF: $F_{SCK}=F_{HCLK}/256$。 对于奇数时钟分频系数，SCK 的占空比并非 50%。时钟信号的低电平持续时间比高电平持续时间多一个周期。 仅可在 BUSY=0 时修改该字段。	0
23	PMM	RW	轮询匹配模式： 1: OR 匹配模式。如果从 FLASH 接收的任意一个未屏蔽位与匹配寄存器中的对应位相匹配，则 SMF 置 1； 0: AND 匹配模式。如果从 FLASH 接收的所有未屏蔽位均与匹配寄存器中的对应位相匹配，则 SMF 置 1。 仅可在 BUSY=0 时修改该位。	0
22	APMS	RW	自动轮询模式停止： 1: 发生匹配时，自动轮询模式停止； 0: 仅通过中止或禁用 QSPI 停止自动轮询模式。 仅可在 BUSY=0 时修改该位。	0
21	Reserved	RO	保留。	0
20	TOIE	RW	TimeOut 中断使能： 1: 使能中断； 0: 禁止中断。	0
19	SMIE	RW	状态匹配中断使能： 1: 使能中断； 0: 禁止中断。	0
18	FTIE	RW	FIFO 阈值中断使能： 1: 使能中断； 0: 禁止中断。	0
17	TCIE	RW	传输完成中断使能： 1: 使能中断； 0: 禁止中断。	0
16	TEIE	RW	传输错误中断使能： 1: 使能中断；	0

			0: 禁止中断。	
[15:14]	Reserved	R0	保留。	0
13	SIOXEN	RW	SIOX 使能: 1: 4 线 IO 输出使能; 0: 4 线 IO 输出关闭。	0
[12:8]	FTHRES[4:0]	RW	FIFO 阈值级别: 定义在间接模式下 FIFO 中将导致 FIFO 阈值标志 (FTF, R32_QSPIx_SR[2]) 置 1 的字节数阈值。 在间接写入模式下 (FMODE=00): 00000: 如果 FIFO 中存在 1 个或更多空闲字节可供写入, 则 FTF 置 1; 00001: 如果 FIFO 中存在 2 个或更多空闲字节可供写入, 则 FTF 置 1; ... 11111: 如果 FIFO 中存在 32 个空闲字节可供写入, 则 FTF 置 1。 在间接读取模式下 (FMODE=01): 00000: 如果 FIFO 中存在 1 个或更多有效字节可供读取, 则 FTF 置 1; 00001: 如果 FIFO 中存在 2 个或更多有效字节可供读取, 则 FTF 置 1; ... 11111: 如果 FIFO 中存在 32 个有效字节可供读取, 则 FTF 置 1。	0
7	FSEL	RW	FLASH 选择: 1: 选择 FLASH2; 0: 选择 FLASH1。 仅可在 BUSY=0 时修改该位。在 DFM=1 时忽略该位。	0
6	DFM	RW	双闪存模式: 1: 使能双闪存模式; 0: 禁止双闪存模式。 仅可在 BUSY=0 时修改该位。	0
5	START	WZ	在间接模式和状态轮询模式下, 所需寄存器已配置好后, 对 START 位写 1, 即可启动 flash 命令序列, 该位硬件自动清零。	0
4	Reserved	R0	保留。	0
3	TCEN	RW	超时计数器使能: 该位仅在内存映射模式 (FMODE=11) 下有效。激活该位后, 如果在一段时间 (通过 TIMEOUT[15:0] (R32_QSPIx_LPTR) 定义) 内一直没有进行访问, 将释放片选 (nCS), 从而降低功耗。 使能超时计数器。 默认情况下, 即便在很长时间内不访问 FLASH, QSPI 也不会停止预取操作, 之前的读取操作将保	0

			持激活状态并且 nCS 保持低电平。由于 nCS 保持低电平时，FLASH 功耗增加，应用程序可能会激活超时计数器（TCEN=1，R32_QSPIx_CR 的位 3）。从而在 FIFO 中写满预取的数据后，若在 TIMEOUT[15:0]（R32_QSPIx_LPTR）个周期的时长内没有访问，则释放 nCS。 1：使能超时计数器，在内存映射模式下，FLASH 持续不活动 TIMEOUT[15:0] 个周期后释放片选（nCS）； 0：禁止超时计数器，在内存映射模式中进行访问后，片选（nCS）保持激活。 仅可在 BUSY=0 时修改该位。	
2	DMAEN	RW	DMA 模式使能： 1：使能 DMA； 0：禁止 DMA。	0
1	ABORT	RW	中止请求/清空 FIFO： 1：请求中止/清空 FIFO； 0：不请求中止/不清空 FIFO。 中止请求：该位中止执行中的命令序列。在中止完成时自动复位。该位可停止当前的传输。 在轮询模式或内存映射模式下，该位也用以复位 APMS 位或 DMAEN 位。 清空 FIFO：将此位编程为 1 可清空 FIFO。所有的内部 FIFO 指针（读和写）将清零。这种情况下，仍存留在 FIFO 中的数据丢失（发送或接收数据不会继续丢失）。 当 QSPI 读写方向转换时，需要清空一次 FIFO。当 R32_QSPIx_SR 寄存器的 IDLEF 位置 1 时，ABORT 位使能，则会产生一次传输完成标志，即 R32_QSPIx_SR 寄存器 TCF 位置位。	0
0	EN	RW	使能： 1：使能 QSPI； 0：禁止 QSPI。	0

37.4.2 QSPI 器件配置寄存器（R32_QSPIx_DCR）（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											FSIZE[4:0]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CSHT[2:0]			Reserved					CKMODE		

位	名称	访问	描述	复位值

[31:21]	Reserved	R0	保留。	0
[20:16]	FSIZE[4:0]	RW	FLASH 大小： FLASH 中的字节数= $2^{[FSIZE+1]}$ ， FSIZE+1 是对 FLASH 寻址所需的地址位数。在间接模式下，FLASH 容量最高可达 4GB（使用 32 位进行寻址），但在内存映射模式下的可寻址空间限制为 256MB。 如果 DFM=1，FSIZE 表示两个 FLASH 容量的总和。 仅可在 BUSY=0 时修改该字段。	0
[15:11]	Reserved	R0	保留。	0
[10:8]	CSHT[2:0]	RW	片选高电平时间： CSHT+1 定义片选（nCS）在发送至 FLASH 的命令之间必须保持高电平的最少 SCK 周期数。 当 PRESCALER[7:0] 位配置为非 0x2 时： 000：nCS 在 FLASH 命令之间保持高电平至少 1 个周期； 001：nCS 在 FLASH 命令之间保持高电平至少 2 个周期； ... 111：nCS 在 FLASH 命令之间保持高电平至少 8 个周期。 当 PRESCALER[7:0] 位配置为 0x2 时： 000：nCS 在 FLASH 命令之间保持高电平至少 1 个周期； 001：nCS 在 FLASH 命令之间保持高电平至少 2 个周期； ... 110：nCS 在 FLASH 命令之间保持高电平至少 7 个周期； 111：保留。 仅可在 BUSY=0 时修改该字段。	0
[7:1]	Reserved	R0	保留。	0
0	CKMODE	RW	模式 0/模式 3： 1：nCS 为高电平（片选释放）时，SCK 必须保持高电平。这称为模式 3； 0：nCS 为高电平（片选释放）时，SCK 必须保持低电平。这称为模式 0。 仅可在 BUSY=0 时修改该字段。	0

37.4.3 QSPI 状态寄存器（R32_QSPIx_SR）（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															IDL EF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FLEVEL[5:0]						Reserved	BUSY	TOF	SMF	FTF	TCF	TEF	

位	名称	访问	描述	复位值
[31:17]	Reserved	R0	保留。	0
16	IDLEF	R0	空闲标志： 当此位置 1 时，QSPI 总线处于空闲状态。	0
[15:14]	Reserved	R0	保留。	0
[13:8]	FLEVEL[5:0]	R0	FIFO 级别： 该字段给出 FIFO 中的有效字节数。FIFO 为空时 FLEVEL=0，写满时 FLEVEL=32。 在内存映射模式和自动状态轮询模式下，FLEVEL 为零。	0
[7:6]	Reserved	R0	保留。	0
5	BUSY	R0	忙： 操作进行时，该位置 1。在对 FLASH 的操作完成并且 FIFO 为空时，该位自动清零。	0
4	TOF	R0	超时标志： 发生超时该位置 1。向 CTOF 写入 1 可将该位清零。	0
3	SMF	R0	状态匹配标志： 在自动轮询模式下，若未屏蔽的接收数据与匹配寄存器（R32_QSPIx_PSMAR）中的对应位相匹配，则该位置 1。向 CSMF 写入 1 可将该位清零。	0
2	FTF	R0	FIFO 阈值标志： 在间接模式下，若达到 FIFO 阈值，或从 FLASH 读取完成后，FIFO 中留有数据时，该位置 1。只要阈值条件不再为“真”，该位就自动清零。 在自动轮询模式下，每次读取状态寄存器时，该位即置 1；读取数据寄存器时，该位清零。	0
1	TCF	R0	传输完成标志： 在间接模式下，当传输的数据数量达到编程设定值，或在任何模式下传输中止时，该位置 1。向 CTCF 写入 1 时，该位清零。	0
0	TEF	R0	传输错误标志： 在间接模式下访问无效地址时，该位置 1。向 CTEF 写入 1 可将该位清零。	0

37.4.4 QSPI 标志清零寄存器（R32_QSPIx_FCR）（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	CTOF	CSMF	Reserved	CTCF	CTEF

位	名称	访问	描述	复位值
[31:5]	Reserved	R0	保留。	0
4	CTOF	RW1Z	清除超时标志： 写入 1 可将 R32_QSPIx_SR 寄存器中的 TOF 标志清零。	0
3	CSMF	RW1Z	清除状态匹配标志： 写入 1 可将 R32_QSPIx_SR 寄存器中的 SMF 标志清零。	0
2	Reserved	R0	保留。	0
1	CTCF	RW1Z	清除传输完成标志： 写入 1 可将 R32_QSPIx_SR 寄存器中的 TCF 标志清零。	0
0	CTEF	RW1Z	清除传输错误标志： 写入 1 可将 R32_QSPIx_SR 寄存器中的 TEF 标志清零。	0

37.4.5 QSPI 数据长度寄存器 (R32_QSPIx_DLR) (x=1/2)

偏移地址：0x10

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

位	名称	访问	描述	复位值
[31:0]	DL[31:0]	RW	数据长度： 在间接模式和状态轮询模式下待检索的数据数量（值+1）。对状态轮询模式应使用不大于 3 的值（表示 4 字节）。 在间接模式下，所有位置 1 表示未定义长度，QSPI 将继续传输数据直到到达由 FSIZE 定义的存储器末尾。 0x0000_0000：传输 1 个字节； 0x0000_0001：传输 2 个字节； 0x0000_0002：传输 3 个字节； 0x0000_0003：传输 4 个字节； ... 0xFFFF_FFFD：传输 4, 294, 967, 294 (4G-2) 个字节； 0xFFFF_FFFE：传输 4, 294, 967, 295 (4G-1) 个字节； 0xFFFF_FFFF：未定义长度——传输所有字节直到到	0

		达由 FSIZE 定义的 FLASH 的结尾。如果 FSIZE=0x1F, 则读取无限继续下去。 在双闪存模式 (DFM=1) 下, 即使该位写入 0, DL[0] 也始终保持为 1, 因此保证了每次访问均传输偶数个字节。 该字段在内存映射模式 (FMODE=10) 下不起作用; 仅可在 BUSY=0 时写入该字段。	

37.4.6 QSPI 通信配置寄存器 (R32_QSPIx_CCR) (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			SI00	FMODE [1:0]		DMODE [1:0]		Reserved	DCYC[4:0]				ABSIZE [1:0]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABMODE [1:0]		ADSIZE [1:0]		ADMODE [1:0]		IMODE [1:0]		INSTRUCTION[7:0]							

位	名称	访问	描述	复位值
[31:29]	Reserved	R0	保留。	0
28	SI00	RW	仅发送指令一次模式: IMODE=00 时, 该位不起作用。 1: 仅为第一条命令发送指令; 0: 在每个事务中发送指令。 仅可在 BUSY=0 时写入该字段。	0
[27:26]	FMODE[1:0]	RW	功能模式: 00: 间接写入模式; 01: 间接读取模式; 10: 自动轮询模式; 11: 内存映射模式。 仅可在 BUSY=0 时写入该字段。	0
[25:24]	DMODE[1:0]	RW	数据模式: 00: 无数据; 01: 单线传输数据; 10: 双线传输数据; 11: 四线传输数据。 该字段还定义空指令阶段。 仅可在 BUSY=0 时写入该字段。	0
23	Reserved	R0	保留。	0
[22:18]	DCYC[4:0]	RW	空指令周期数: 该字段定义空指令阶段的持续时间。在 SDR 模式下, 它指定 SCK 周期数 (0-31)。 仅可在 BUSY=0 时写入该字段。	0
[17:16]	ABSIZE[1:0]	RW	交替字节长度: 00: 8 位交替字节;	0

			01: 16 位交替字节; 10: 24 位交替字节; 11: 32 位交替字节。 仅可在 BUSY=0 时写入该字段。	
[15:14]	ABMODE[1:0]	RW	交替字节模式: 00: 无交替字节; 01: 单线传输交替字节; 10: 双线传输交替字节; 11: 四线传输交替字节。 仅可在 BUSY=0 时写入该字段。	0
[13:12]	ADSIZE[1:0]	RW	地址长度: 00: 8 位地址; 01: 16 位地址; 10: 24 位地址; 11: 32 位地址。 仅可在 BUSY=0 时写入该字段。	0
[11:10]	ADMODE[1:0]	RW	地址模式: 00: 无地址; 01: 单线传输地址; 10: 双线传输地址; 11: 四线传输地址。 仅可在 BUSY=0 时写入该字段。	0
[9:8]	IMODE[1:0]	RW	指令模式: 00: 无指令; 01: 单线传输指令; 10: 双线传输指令; 11: 四线传输指令; 仅可在 BUSY=0 时写入该字段。	0
[7:0]	INSTRUCTION[7:0]	RW	指令: 指定要发送到外部 SPI 设备的指令。 仅可在 BUSY=0 时写入该字段。	0

37.4.7 QSPI 地址寄存器 (R32_QSPIx_AR) (x=1/2)

偏移地址: 0x18

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

位	名称	访问	描述	复位值
[31:0]	ADDRESS[31:0]	RW	地址: 指定要发送到外部 FLASH 的地址。 BUSY=0 或 FMODE=11 (内存映射模式) 时, 将忽略写入该字段。	0

			在双闪存模式下，由于地址始终为偶地址，ADDRESS[0]自动保持为0。	

37.4.8 QSPI 交替字节寄存器 (R32_QSPIx_ABR) (x=1/2)

偏移地址：0x1C

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

位	名称	访问	描述	复位值
[31:0]	ALTERNATE[31:0]	RW	交替字节： 指定要在地址后立即发送到外部 SPI 设备的可选数据。 仅可在 BUSY=0 时写入该字段。	0

37.4.9 QSPI 数据寄存器 (R32_QSPIx_DR) (x=1/2)

偏移地址：0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA[31:16]															
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[31:0]	RW	数据： 指定要与外部 SPI 设备交换的数据。 在间接写入模式下，写入该寄存器的数据在数据阶段发送到 FLASH，在此之前则存储于 FIFO。如果 FIFO 太满，将暂停写入，直到 FIFO 具有足够的空间接受要写入的数据才继续。 在间接模式下，读取该寄存器可获得（通过 FIFO）已从 FLASH 接收的数据。如果 FIFO 所含字节数比读取操作要求的字节数少并且 BUSY=1，将暂停读取，直到足够的数据出现或传输完成（不分先后）才继续。 在自动轮询模式下，该寄存器包含最后从 FLASH 读取的数据（未进行屏蔽）。 支持对该寄存器进行字、半字以及字节访问。在间接写入模式下，字节写入将在 FIFO 中增加 1 个字节，半字写入增加 2 个，而字写入则增加 4 个。类似地，在间接读取模式下，字节读取将擦除 FIFO 中的 1 个字节，半字读取擦除 2 个，而字读取则擦除 4 个。	0

		间接模式下的访问必须与此寄存器的最低位对齐：字节读取必须读取 DATA[7:0] 而半字读取必须读取 DATA[15:0]。	

37.4.10 QSPI 轮询状态屏蔽寄存器 (R32_QSPIx_PSMKR) (x=1/2)

偏移地址：0x24

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

位	名称	访问	描述	复位值
[31:0]	MASK[31:0]	RW	状态屏蔽： 对在轮询模式下接收的状态字节进行屏蔽。 对于位 n： 1：不屏蔽在自动轮询模式下所接收数据的位 n，在匹配逻辑中考虑其值； 0：屏蔽在自动轮询模式下所接收数据的位 n，在匹配逻辑中不考虑其值。 仅可在 BUSY=0 时写入该字段。	0

37.4.11 QSPI 轮询状态匹配寄存器 (R32_QSPIx_PSMAR) (x=1/2)

偏移地址：0x28

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

位	名称	访问	描述	复位值
[31:0]	MATCH[31:0]	RW	状态匹配： 该值将与屏蔽状态寄存器比较以进行匹配。 仅可在 BUSY=0 时写入该字段。	0

37.4.12 QSPI 轮询间隔寄存器 (R32_QSPIx_PIR) (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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERVAL[15:0]															

位	名称	访问	描述	复位值

[31:16]	Reserved	R0	保留。	0
[15:0]	INTERVAL[15:0]	RW	轮询间隔： 自动轮询阶段读取操作之间的 SCK 周期数。 仅可在 BUSY=0 时写入该字段。	0

37.4.13 QSPI 低功耗超时寄存器 (R32_QSPIx_LPTR) (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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT[15:0]															

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:0]	TIMEOUT[15:0]	RW	超时时长： 在内存映射模式下，每次访问结束后，QSPI 将预 取后续字节并将其存放在 FIFO 中。该字段指示在 FIFO 写满后，QSPI 等待多少个 SCK 时钟周期才让 nCS 升至高电平将 FLASH 置为低功耗状态。 仅可在 BUSY=0 时写入该字段。	0

第 38 章 单线协议主接口（SWPMI）

单线协议主接口（SWPMI）是一种全双工单线通信的技术，该技术基于 ETSI TS 102 613 标准规范实现单线协议（SWP）通信。

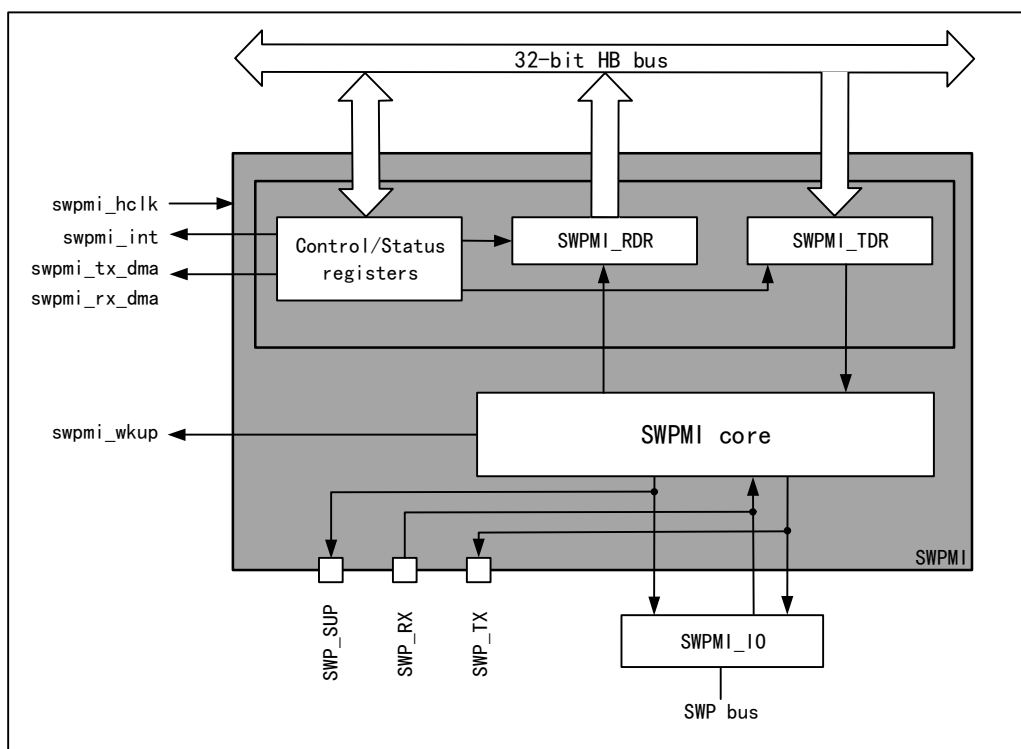
在 SWPMI 中，数据可以通过两种物理方式传输：第一种是通过电压域（S1 信号）来实现从主设备到从设备的数据传输；第二种是通过电流域（S2 信号）来实现从从设备到主设备的数据传输。S1 信号使用脉冲宽度调制的数字调制方式传输，而 S2 信号则从电流变化中传输数据。

38.1 主要特征

- 支持全双工通信模式
- 支持自动处理填充位
- 支持自动 SWP 总线状态管理
- 提供回送模式用于测试
- 支持自动处理帧起始（SOF）和自动处理帧结束（EOF）
- 支持比特率可配置，高达 2Mbit/s，支持中断可配置
- 提供 CRC 错误、下溢、上溢和从器件恢复监测标志
- 支持 CRC-16 的计算、生成与检验

38.2 概述

图 38-1 单线协议主接口的结构框图



38.3 功能描述

38.3.1 SWP 初始化和激活

SWP 初始化和激活可实现 SWPMI_IO 状态从低电平转为高电平。使用内部收发器时，软件操作按以

下流程进行：

- 1) 通过 R32_SWPMI_CR 寄存器 SWPTEN 位置 1，使能 SWPMI_I0 收发器，并将 SWPMI_I0 设为低电平（SWP 总线“已禁用”）；
- 2) 等待 R32_SWPMI_SR 寄存器 RDYF 位置 1（轮询标志或使用 R32_SWPMI_IER 寄存器 RDYIE 位使能中断）；
- 3) 通过 R32_SWPMI_CR 寄存器 SWPACT 位置 1，SWP 总线从“已禁用”转换到“已挂起”状态，即激活 SWP。

38.3.2 SWP 总线状态

SWP 总线有 3 种状态：“已禁用”、“已挂起”和“已激活”。

状态转换有：

- 激活：由“已禁用”状态转换到“已挂起”状态；
- 挂起：由“已激活”状态转换到“已挂起”状态；
- 禁用：由“已挂起”状态转换到“已禁用”状态；
- 主器件恢复：由主器件发起从“已挂起”状态到“已激活”状态的转换；
- 从器件恢复：由从器件发起从“已挂起”状态到“已激活”状态的转换。

激活

在复位阶段和刚完成复位后，SWPMI_I0 配置为模拟模式。激活 SWP 总线相关流程请参考 38.3.1 小节。

挂起

若 SWP 总线与从器件存在通信（无论发送或接收模式下），SWP 总线始终保持“已激活”状态。若后续没有发送或接收请求，则在 7 个空闲位后，SWP 总线将切换回“已挂起”状态。

禁用

1) 禁用请求

SWP 处于“已挂起”状态时，如果不再需要通信，用户可通过禁止 SWPMI 外设来请求将 SWP 切换到“已禁用”状态。将 R32_SWPMI_CR 寄存器 DEACT 位置 1 才可切换到“已禁用”状态。若 SWPMI 未检测到“从器件恢复”状态，则 R32_SWPMI_ISR 寄存器 DEACTF 位将置 1，同时 R32_SWPMI_ICR 寄存器 SWPACT 位将清零。若 DEACT 位置 1 后，SWPMI 检测到“从器件恢复”状态，那么 R32_SWPMI_ISR 寄存器 SRF 标志将置 1，DEACTF 保持清零，SWPACT 保持置 1，DEACT 位清零。将 R32_SWPMI_CR 寄存器 DEACT 位清零，然后将 SWPACT 位置 1，即可重新激活 SWP。

2) 禁用模式

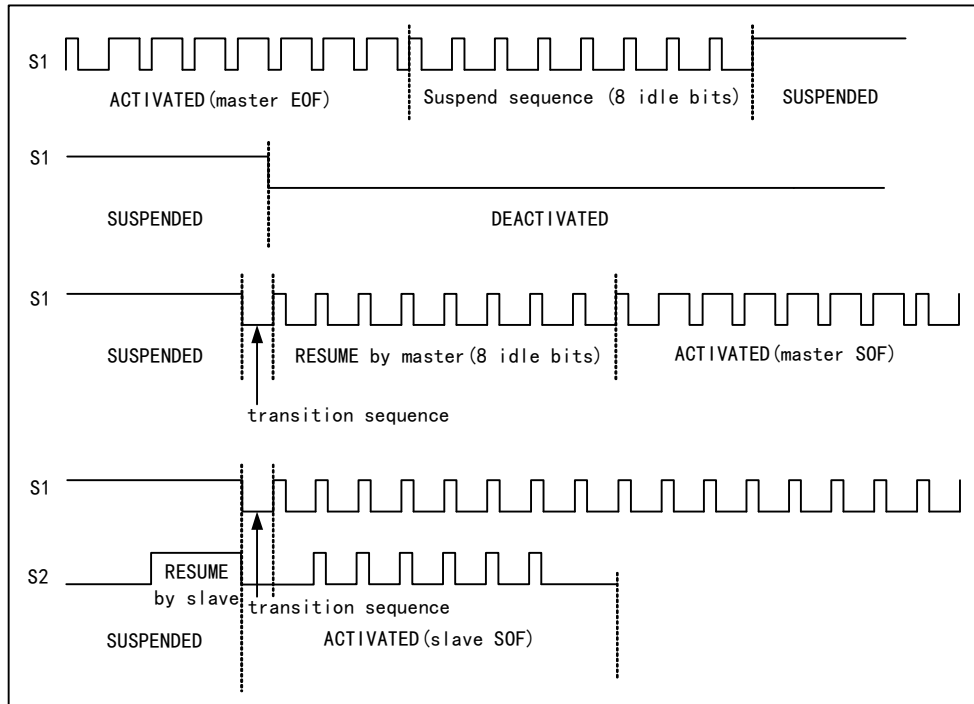
用户必须将 R32_SWPMI_CR 寄存器 SWPACT 位清零，SWP 才可立即切换到“已禁用”模式，并忽略可能传入的“从器件恢复”状态。

注：为了进一步降低电流消耗，可将 GPIO 控制器中的 SWPMI_I0 端口配置为推挽输出模式，并将其置于低电平状态，最后将 R32_SWPMI_CR 寄存器 SWPTEN 位清零。

主器件恢复

SWPMI 使能后，用户可请求 SWPMI 帧发送。首先，SWPMI 发送转换序列和 8 个空闲位，即主器件恢复，然后开始帧发送。“主器件恢复”后，SWP 从“已挂起”状态转换到“已激活”状态，请参考图 38-2：SWP 总线状态。

图 38-2 SWP 总线状态图



从器件恢复

SWPMI 使能后，若 SWPMI 接收到从器件的恢复状态信号，SWP 会从“已挂起”状态转换到“已激活”状态。“从器件恢复”将触发 R32_SWPMI_ISR 寄存器中的 SRF 标志位置 1。

38.3.3 SWPMI_I0（内部收发器）旁路

微控制器中集成了符合 ETSI TS 102 613 技术规范的 SWPMI_I0（内部收发器）。为实现将该收发器旁路，可通过将 R32_SWPMI_OR 寄存器 SWP_TBYP 位置 1。在旁路模式下，SWPMI_I0 被禁止，SWP_RX、SWP_TX 和 SWPMI_SUP 信号作为这三个 GPIO 上的复用功能可供使用。此配置可用于连接外部收发器。

注：在 SWPMI_I0 旁路模式下，R32_SWPMI_CR 寄存器 SWPTEN 位必须清零。

38.3.4 SWPMI 比特率

在 R32_SWPMI_BRR 寄存器中，根据以下公式设置比特率：

$$F_{SWP} = F_{HCLK} / ((BR[7:0] + 1) \times 4)$$

注：最大比特率为 2Mbit/s。

38.3.5 SWPMI 帧处理

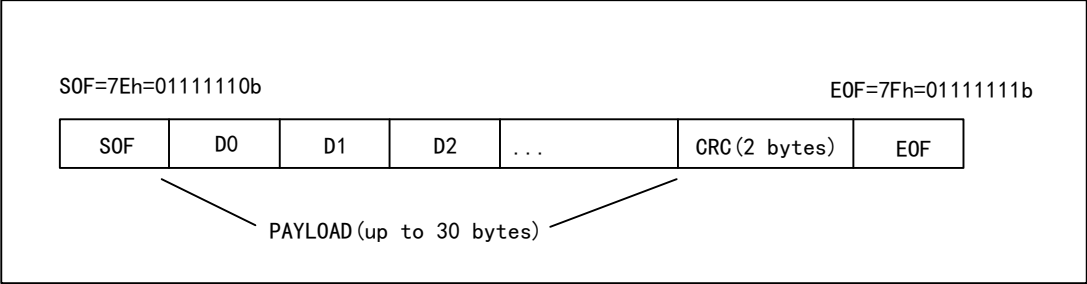
SWP 帧由以下几个部分组成：帧起始（SOF）、1 到 30 字节的有效负载、16 位 CRC（循环冗余校验）和 EOF（帧结束）。如图 38-3。

SWPMI 集成了一个 32 位数据发送寄存器（R32_SWPMI_TDR）和一个 32 位数据接收寄存器（R32_SWPMI_RDR）。

在数据发送模式下，SOF 插入、EOF 插入、CRC 计算和插入均由 SWPMI 自动管理。用户仅需提供有效负载的大小和内容。向 R32_SWPMI_TDR 寄存器中写入数据后，会立即进行帧发送。将通过特定的标志位来指示发送数据寄存器为空、数据帧发送完成事件。

在数据接收模式下，SOF 删除、EOF 删除、CRC 计算和检查均由 SWPMI 自动管理。用户仅需读取有效负载的大小和内容。将通过特定的标志位来指示接收数据完成帧接收、寄存器为满、可能的 CRC 错误事件。

图 38-3 SWP 帧结构



填充位插入（发送模式下）和删除（接收模式下）均由 SWPMI 内核自动管理。这些操作对用户是透明的。

38.3.6 发送过程

在进行帧发送之前，用户必须先激活 SWP，请参考 38.3.1。帧传输的软件实现方式有多种，包括无软件缓冲区模式、单软件缓冲区模式以及多软件缓冲区模式。在采用软件缓冲区的情况下，数据通过 DMA 通道从 RAM 存储器的软件缓冲区传输到 SWPMI 外设发送数据寄存器。

无软件缓冲区模式

无软件缓冲区模式下，无需启用 DMA。主循环或 SWPMI 中断程序内轮询状态标志可处理 SWP 帧发送。SWPMI 包含一个 32 位发送数据寄存器（R32_SWPMI_TDR），若向该寄存器进行写操作将触发多达 4 个字节的发送操作。

通过将 R32_SWPMI_CR 寄存器 TXDMA 位清零，可选择无软件缓冲区模式。

首次向 R32_SWPMI_TDR 寄存器写入数据时，可触发帧发送。在向 R32_SWPMI_TDR 寄存器写入的第一个字中，位[7:0]表示有效负载中的数据字节数，位[15:8]、位[23:16]、位[31:24]分别包含有效负载的前 3 个字节，其中位[15:8]包含有效负载的第一个字节，位[23:16]包含第二个字节，位[31:24]包含第三个字节。随后对 R32_SWPMI_TDR 寄存器进行的写操作将仅包含后续的有效负载数据字节（每次写操作最多 4 个字节）。

注：向 R32_SWPMI_TDR 寄存器写入的第一个 32 位字的低有效字节用于对有效负载中的数据字节数进行编码。该数值范围应介于 1~30 之间。若最低有效字节中的值超出此范围，将不予考虑，且不会触发数据发送。

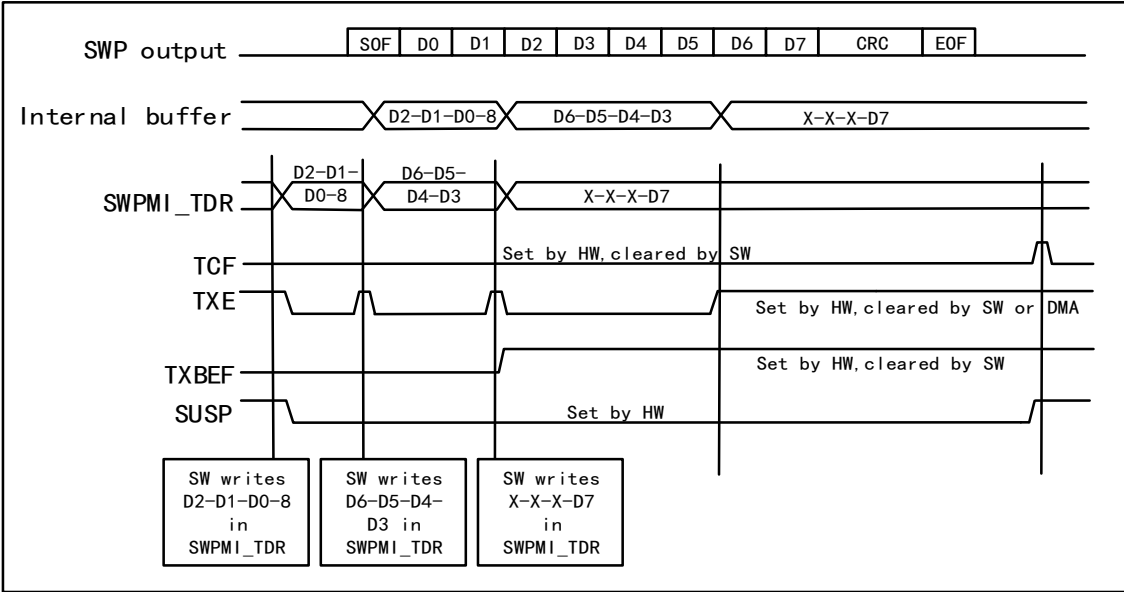
对 R32_SWPMI_TDR 寄存器进行写操作将触发以下一系列操作：

- 1) 当 SWP 总线为“已挂起”状态，系统将发送转换序列以及 8 个空闲位（主器件恢复）。相反，若 SWP 总线为“已激活”状态，则不会触发此操作；
- 2) 发送帧起始；
- 3) 根据 R32_SWPMI_TRD 寄存器的内容，发送有效负载。若有效负载中的字节数超过 3，只要 R32_SWPMI_ISR 寄存器 TXBEF 标志位未置 1，每当 R32_SWPMI_ISR 寄存器的 TXE 标志位置 1 时，R32_SWPMI_TDR 就需要由软件重新填充；
- 4) 发送 16 位 CRC（由 SWPMI 内核自动计算）；
- 5) 发送帧结束（EOF）。

软件对 R32_SWPMI_TDR 寄存器进行写操作时，将自动清零 TXE 标志。

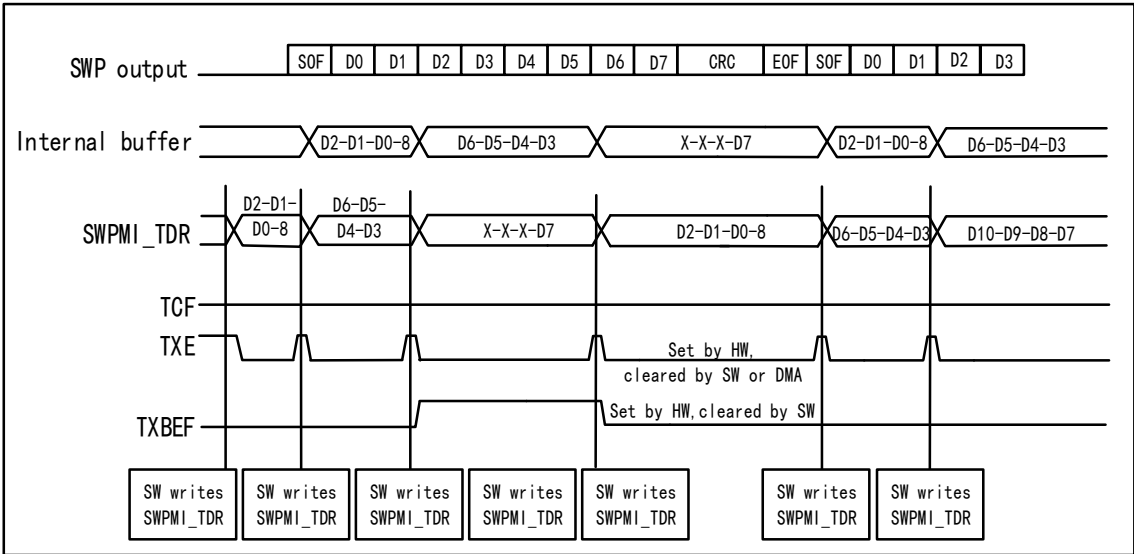
完整的帧发送成功后，若尚未请求其他帧发送（即在 TXBEF 标志置 1 后未再次对 R32_SWPMI_TDR 进行写操作），那么在 EOF 发送完成后，再经过 7 个空闲位，R32_SWPMI_ISR 寄存器中的 TCF 和 SUSP 标志将会置 1。此外，如果将 R32_SWPMI_IER 寄存器中将 TCIE 位置 1，将会触发中断。如图 38-4。

图 38-4 SWPMI 无软件缓冲区模式发送



若在 EOF 发送结束之前，再次发起帧发送请求，那么 TCF 标志位不会置 1，且该帧将与前一个帧衔接（二者之间只有一个空闲位，请参考图 38-5）。

图 38-5 SWPMI 无软件缓冲区模式发送，连续帧



单软件缓冲区模式

单软件缓冲区模式下可通过 DMA 发送完整 SWP 帧，且不需要 CPU 介入。DMA 将重新填充 32 位 R32_SWPMI_TDR 寄存器，软件可使用 SWPMI_TXBEF 标志轮询帧发送是否结束。选择单软件缓冲区模式：将 R32_SWPMI_CR 寄存器 TXDMA 位置 1、TXMODE 位清零。

DMA 通道或数据流必须配置为以下模式（请参考 DMA 部分）：

- 存储器大小设为 32 位；
- 外设大小设为 32 位；
- 使能存储器递增模式；
- 禁止循环模式；
- 禁止外设递增模式；
- 禁止存储器到存储器模式；

- 要传输的字数必须根据 SWP 帧长度进行设置；
- 目标地址为 R32_SWPMI_TDR 寄存器；
- 数据传输方向设为从存储器读取。
- 源地址是 RAM 中的 SWP 帧缓冲区。

随后，用户进行以下操作：

- 1) 将 R32_SWPMI_CR 寄存器 TXDMA 位置 1；
- 2) 将 R32_SWPMI_IER 寄存器 TXBEIE 位置 1；
- 3) 对 RAM 存储器中的缓冲区进行填充（有效负载中的数据字节数由第一个字的最低有效字节表示）；
- 4) 通过使能 DMA 模块中的数据流或通道，以触发 DMA 传输和帧发送。

当 R32_SWPMI_ISR 中的 TXE 标志置 1 时，SWPMI 将发起 DMA 请求。在 DMA 对 R32_SWPMI_TDR 寄存器进行写操作时，TXE 标志位将被自动清零。

在 SWPMI 中断程序中，用户需要查看 R32_SWPMI_ISR 寄存器 TXBEF 位。若该位置 1，并且还有其他帧需要发送，则用户必须进行以下操作：

- 1) 禁止 DMA 模块中的数据流或通道；
- 2) 使用要发送的下一帧的内容更新 RAM 存储器中的缓冲区；
- 3) 在 DMA 模块中设置要传输的总字数；
- 4) 通过使能 DMA 模块中的数据流或通道，以触发下一次帧发送；
- 5) 通过将 R32_SWPMI_ICR 寄存器 CTXBEF 位置 1，以清零 TXBEF 标志。

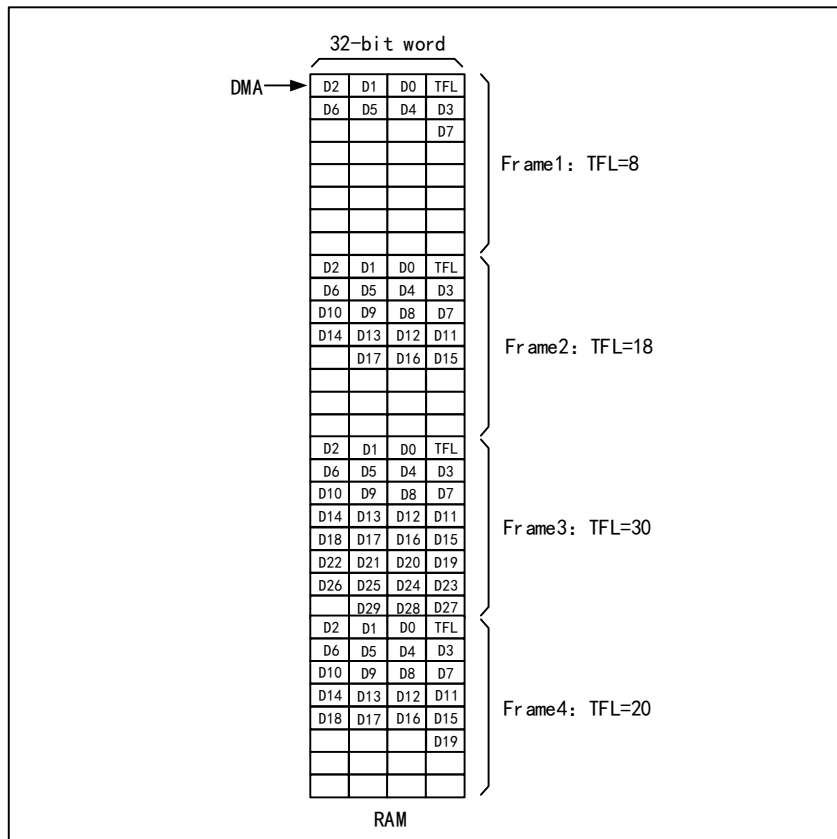
多软件缓冲区模式

多软件缓冲区模式支持在 RAM 存储器中配置多个帧缓冲区，以保障数据的连续传输、维持极低 CPU 负载，并为软件提供更多延迟以执行缓冲区更新（使用 DMA）。软件能够随时监控 DMA 计数器，并相应地在 RAM 存储器中更新 SWP 帧。

为实现最佳效果，多软件缓冲区模式应当与循环模式下的 DMA 结合使用。

在 SWP 帧有效负载中，无论字节数量多少，RAM 存储器中的每个发送缓冲区都应具备固定的长度，即八个 32 位字。这些发送缓冲区需要由软件进行填充，两个连续缓冲区之间保持 8 字节的偏移量。缓冲区的第一个数据字节表示帧有效负载的字节数。如下图所示。

图 38-6 SWPMI 多软件缓冲区模式发送



通过将 R32_SWPMI_CR 寄存器中的 TXDMA 和 TXMODE 位置 1 来选择多软件缓冲区模式。若要使用 4 个发送缓冲区，用户必须如下配置 DMA：

DMA 通道或数据流必须配置为以下模式（请参考 DMA 部分）：

- 存储器大小设为 32 位；
- 外设大小设为 32 位；
- 必须将要传输的字数设为 32（每个缓冲区 8 个字）；
- 数据传输方向设为从存储器读取；
- 禁止存储器到存储器模式；
- 禁止外设递增模式；
- 使能存储器递增模式；
- 使能循环模式；
- 源地址是 RAM 中的缓冲区 1；
- 目标地址为 R32_SWPMI_TDR 寄存器。

随后，用户进行以下操作：

- 1) 将 R32_SWPMI_CR 寄存器 TXDMA 位置 1；
- 2) 将 R32_SWPMI_IER 寄存器 TXBEIE 位置 1；
- 3) 对 RAM 存储器中的缓冲区 1、缓冲区 2、缓冲区 3 和缓冲区 4 进行填充（有效负载中的数据字节数由第一个字的最低有效字节表示）；
- 4) 通过使能 DMA 模块中的数据流或通道，以触发 DMA 传输和帧发送。

在 SWPMI 中断例程中，用户必须检查 R32_SWPMI_ISR 寄存器 TXBEF 位。若该位置 1，用户应将 R32_SWPMI_ICR 寄存器 CTXBEF 位置 1，以清零 TXBEF 标志，并且更新 RAM 存储器中的缓冲区 1。当下一次 SWPMI 中断发生时，用户将更新缓冲区 2，如此循环往复。

软件还可读取 DMA 寄存器中 DMA 计数器（待传输的数据量），以检索已从 RAM 存储器传输并发送

的帧。例如，软件使用 4 个发送缓冲区，且 DMA 计数器值等于 17，则表示 RAM 区域中有两个缓冲区已准备好进行更新。这一功能可有效解决此情况：软件处理 SWPMI 中断前需要先发送多个帧。若发生这种情况，软件必须更新多个缓冲区。

当没有帧需要发送时，用户必须禁止 DMA 模块的循环模式。发送将在缓冲区 4 发送结束时停止。

若提前停止发送，如在缓冲区 2 结束时，用户必须将缓冲区 3 和缓冲区 4 中第一个字的低有效字节设为 0。

当读取到第一个字的最低有效字节中有效负载的数据字节数为 0 时，R32_SWPMI_CR 寄存器 TXDMA 位将由硬件清零。

38.3.7 接收过程

在进行帧发送之前，用户必须激活 SWP。请参考 38.3.1。R32_SWPMI_CR 寄存器 SWPACT 位置 1 后，“从器件恢复”状态会将 R32_SWPMI_ISR 寄存器中的 SRF 标志置 1，并自动使能 SWPMI 以进行帧接收。

若 SWP 总线处于“已激活”状态，如正在进行帧发送，则 SWPMI 内核无需“从器件恢复”状态，可立即进行接收。

帧接收的软件实现方式有多种，包括无软件缓冲区模式、单软件缓冲区模式以及多软件缓冲区模式。软件缓冲区的使用需要通过 DMA 通道，以便将数据从 SWPMI 外设中的接收数据寄存器传输到 RAM 存储器中的软件缓冲区。

无软件缓冲区模式

无软件缓冲区模式下，无需启用 DMA。主循环或 SWPMI 中断程序内轮询状态标志可处理 SWP 帧发送。SWPMI 包含一个 32 位接收数据寄存器（R32_SWPMI_RDR），允许在读取该寄存器之前接收多达 4 个字节。

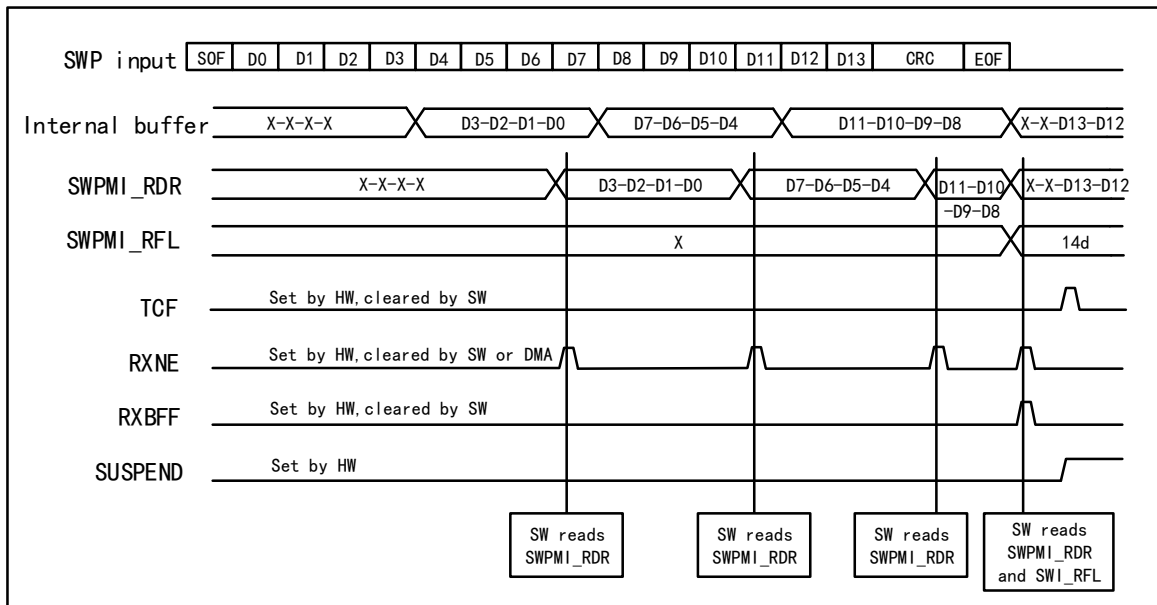
通过将 R32_SWPMI_CR 寄存器 RXDMA 位复位，可选择无软件缓冲区模式。

接收到帧起始（SOF）后，接下来的有效负载字节将被存储于 R32_SWPMI_RDR 寄存器。当 R32_SWPMI_RDR 已满时，R32_SWPMI_ISR 中的 RXNE 标志将置 1。此外，如果 R32_SWPMI_IER 寄存器中的 RIE 位置 1，还将生成中断。用户可读取 R32_SWPMI_RDR 寄存器来获取数据，当软件读取 R32_SWPMI_RDR 寄存器时，RXNE 标志会自动清零。

一旦接收到完整的帧（CRC、EOF），R32_SWPMI_ISR 寄存器中的 RXNE 和 RXBFF 标志便会置 1。用户必须读取 R32_SWPMI_RDR 寄存器中有效负载的最后一个或几个字节，并通过 R32_SWPMI_ICR 中的 CRXBFF 标志置 1，以将 RXBFF 标志清零。有效负载中的数据字节数可在 R32_SWPMI_RFL 寄存器中获取。当软件读取 R32_SWPMI_RDR 寄存器时，RXNE 标志将再次自动复位。如下图所示。

当 RXNE 清零时，读取 R32_SWPMI_RDR 寄存器将返回 0。

图 38-7 SWPMI 无软件缓冲区模式接收



单软件缓冲区模式

单软件缓冲区模式下可通过 DMA 发送完整 SWP 帧，且不需要 CPU 介入。DMA 将接收到的数据从 32 位 R32_SWPMI_RDR 寄存器传输到 RAM 存储器，软件可使用 SWPMI_RBFF 标志轮询帧接收是否结束。单软件缓冲区模式：将 R32_SWPMI_CR 寄存器 RXDMA 位置 1、RXMODE 位清零。

DMA 通道或数据流必须配置为以下模式（请参考 DMA 部分）：

- 存储器大小设为 32 位；
- 外设大小设为 32 位；
- 使能存储器递增模式；
- 禁止循环模式；
- 禁止外设递增模式；
- 禁止存储器到存储器模式；
- 传输的字数设置为 8；
- 目标地址是 RAM 中的 SWP 帧缓冲区；
- 数据传输方向设为从存储器读取；
- 源地址是 R32_SWPMI_RDR 寄存器。

随后，用户进行以下操作：

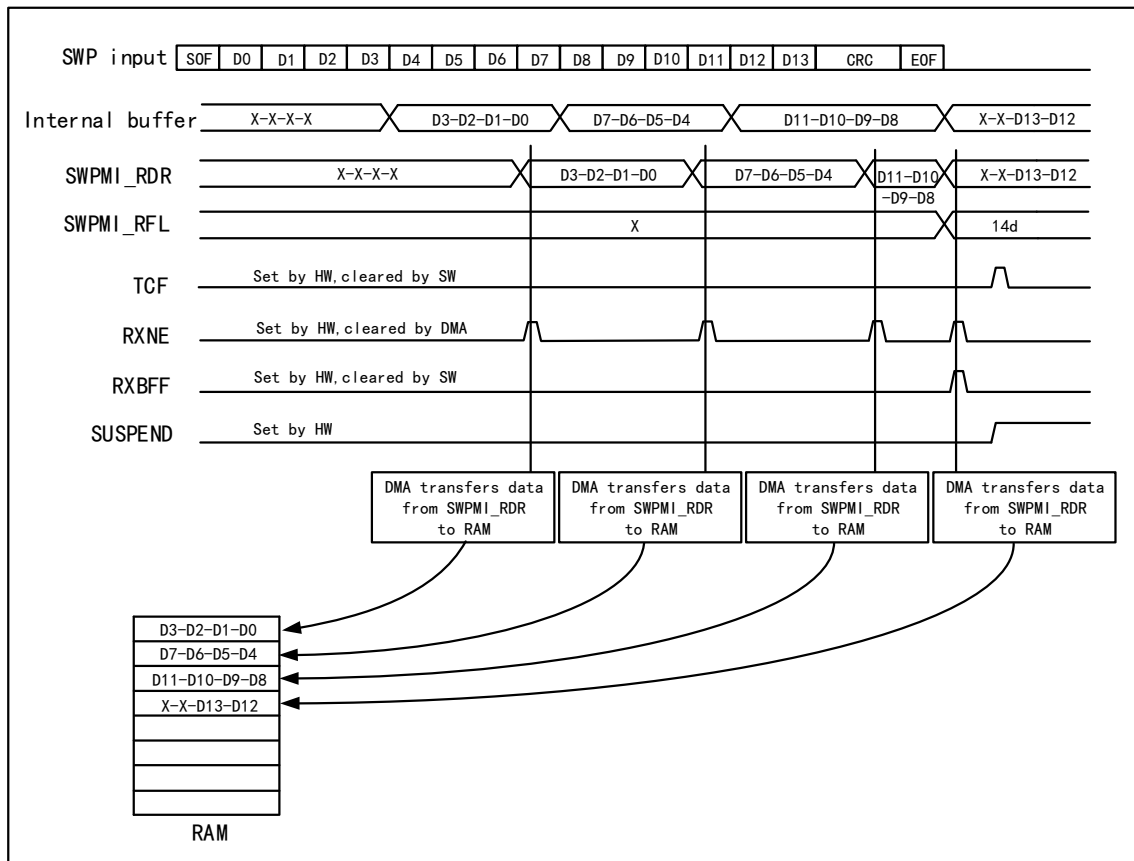
- 1) 将 R32_SWPMI_CR 寄存器 RXDMA 位置 1；
- 2) 将 R32_SWPMI_IER 寄存器 RXBEIE 位置 1；
- 3) 通过使能 DMA 模块中的数据流或通道。

当 R32_SWPMI_ISR 中的 RXNE 标志置 1 时，SWPMI 将发起 DMA 请求。在 DMA 对 R32_SWPMI_RDR 寄存器进行写操作时，RXNE 标志位将被自动清零。

在 SWPMI 中断程序中，用户需要查看 R32_SWPMI_ISR 寄存器 RXBEF 位。若该位置 1，并且还有其他帧需要发送，则用户必须进行以下操作：

- 1) 禁止 DMA 模块中的数据流或通道；
- 2) 读取 R32_SWPMI_RFL 寄存器中已接收的帧有效负载的字节数；
- 3) 读取 RAM 缓冲区中的帧有效负载；
- 4) 使能 DMA 模块中的数据流或通道；
- 5) 通过将 R32_SWPMI_ICR 寄存器 CTXBEF 位置 1，以清零 RXBEF 标志。详情请参考图 38-8。

图 38-8 SWPMI 单软件缓冲区模式接收



多软件缓冲区模式

多软件缓冲区模式支持在 RAM 存储器中配置多个帧缓冲区，以保障数据的连续传输、维持极低 CPU 负载。帧有效负载与帧状态标志一起存储在 RAM 存储器中。软件可随时检查 DMA 计数器和状态标志，以处理 RAM 存储器中接收到的 SWP 帧。

为实现最佳效果，多软件缓冲区模式应当与循环模式下的 DMA 结合使用。选择多软件缓冲区模式：将 R32_SWPMI_CR 寄存器中的 RXDMA 和 RXMODE 位置 1。

为实现在 RAM 中使用 n 个接收缓冲区，DMA 通道或数据流必须配置为以下模式（请参考 DMA 部分）：

- 存储器大小设为 32 位；
- 外设大小设为 32 位；
- 必须将要传输的字数设为 8*n（每个缓冲区 8 个字）；
- 数据传输方向设为从外设读取；
- 禁止存储器到存储器模式；
- 禁止外设递增模式；
- 使能存储器递增模式；
- 使能循环模式；
- 源地址是 R32_SWPMI_TDR 寄存器；
- 目标地址为 RAM 中的缓冲区 1。

随后，用户进行以下操作：

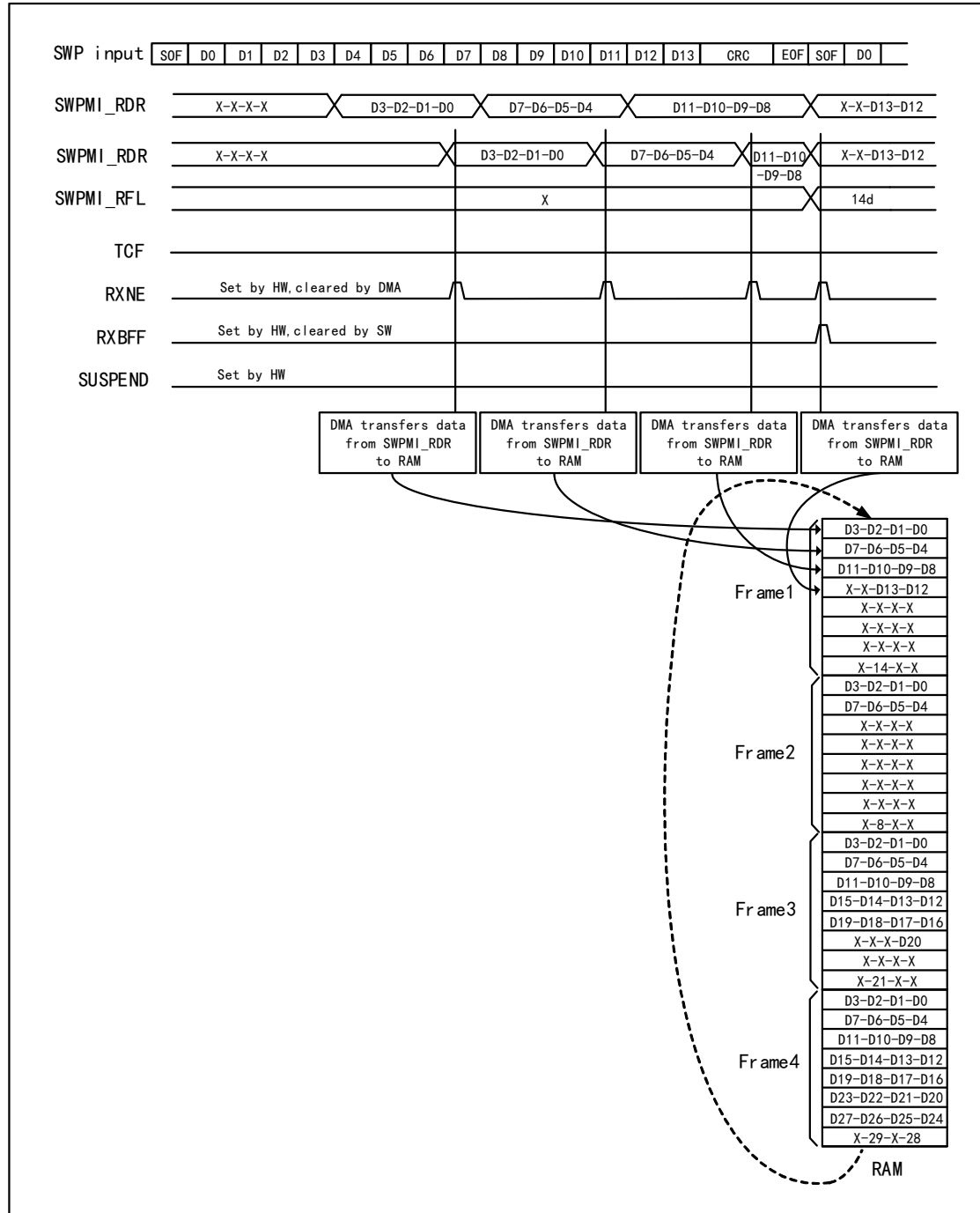
- 1) 将 R32_SWPMI_CR 寄存器 RXDMA 位置 1；
- 2) 将 R32_SWPMI_IER 寄存器 RXBEIE 位置 1；
- 3) 使能 DMA 模块中的数据流或通道。

在 SWPMI 中断例程中，用户必须检查 R32_SWPMI_ISR 寄存器 RXBEF 位。若该位置 1，用户应将

R32_SWPMI_ICR 寄存器 CTXBEF 位置 1, 以清零 RXBEF 标志, 并且可读取在第一个缓冲区(在 DMA2_CMAR1 中设置的 RAM 地址)中接收到的第一个帧有效负载。有效负载的数据字节数可在倒数第 8 个字的位 [23:16] 中获取。

在下一个 SWPMI 中断例程发生时, 用户将读取在第二个缓冲区(在 DMA2_CMAR1 中设置的地址+8)中接收到的第二个帧, 如此循环往复。如图 38-9。

图 38-9 SWPMI 多软件缓冲区模式接收



若应用软件无法保证在接收到下一次帧之前处理 SWPMI 中断, 那么每个缓冲区状态将在第 8 个缓冲区字的最高有效字节中呈现:

- 接收上溢标志: R32_SWPMI_ISR 寄存器中的 RXOVRF 标志, 由第 8 个字的位 25 表示。有关上溢错误的说明, 请参考第 38.3.8 节。
- 接收缓冲区已满标志: R32_SWPMI_ISR 寄存器中的 RXBFF 标志, 由第 8 个字的位 26 表示。

- CRC 错误标志：R32_SWPMI_ISR 寄存器中的 RXBERF 标志，由第 8 个字的位 24 表示。有关 CRC 错误的说明，请参见第 38.3.8 节。

当发生 CRC 错误时，RXBFF 和 RXBERF 标志均设置为 1，这意味着位 24 和位 26 都将被置 1。当发生上溢时，上溢标志位会被置 1，即位 25 会被置 1。仅在接收最后一个字节时发生上溢的情况下，接收缓冲区已满标志才会置 1；随后，对于当前和下一次帧接收，位 25 和位 26 将被置 1。

软件能够读取 DMA 寄存器中的 DMA 计数器，从而检索通过 DMA 传输并存储在 RAM 的帧。例如，当软件使用 4 个接收缓冲区，且 DMA 计数器值等于 17 时，这表明 RAM 区域中有两个缓冲区已准备就绪，可供软件读取。在多软件缓冲区接收模式下，若软件正在读取第 8 个字的位 24、25 和 26，那么在每次接收帧之后，将无需清零 RXBERF、RXOVRF 和 RXBFF 标志位。

38.3.8 错误管理

有效负载发送期间发生下溢

在发送帧有效负载过程中，若出现发送下溢，R32_SWPMI_ISR 寄存器的 TXUNRF 标志将会被触发以示警告。若 R32_SWPMI_IER 寄存器 TXBUNREIE 位置 1，将会产生中断。当发生发送下溢时，SWPMI 将会停止发送有效负载，并发送损坏的 CRC（发送的第一个 CRC 字节的第一个位被反转），紧接着是 EOF 标志。若使用 DMA，R32_SWPMI_CR 寄存器中的 TXDMA 位将自动清零。

当 TXUNRF 置 1 时，对 R32_SWPMI_TDR 寄存器进行的任何写操作均会被忽略。用户必须通过将 R32_SWPMI_ICR 寄存器 CTXUNRF 位置 1，以实现 TXUNRF 标志清零。

有效负载接收期间发生上溢

在接收帧有效负载过程中，若出现接收上溢，R32_SWPMI_ISR 寄存器的 RXOVRF 标志将会被触发以示警告。当发生接收上溢时，传入的数据将不会被 SWPMI 利用来更新 R32_SWPMI_RDR，因此传入的数据将丢失。

在 EOF 之前持续进行数据接收，一旦上溢条件消失，RXBFF 标志将会置 1。当 RXBFF 标志置 1 时，用户可检查 RXOVRF 标志。为实现清零 RXOVRF 标志，用户需将 R32_SWPMI_ICR 寄存器中的 CRXOVRF 位置 1。若用户需立即检测上溢情况，可将 R32_SWPMI_IER 寄存器中的 RXBOVREIE 位置 1，从而确保在发生上溢时立即触发中断。

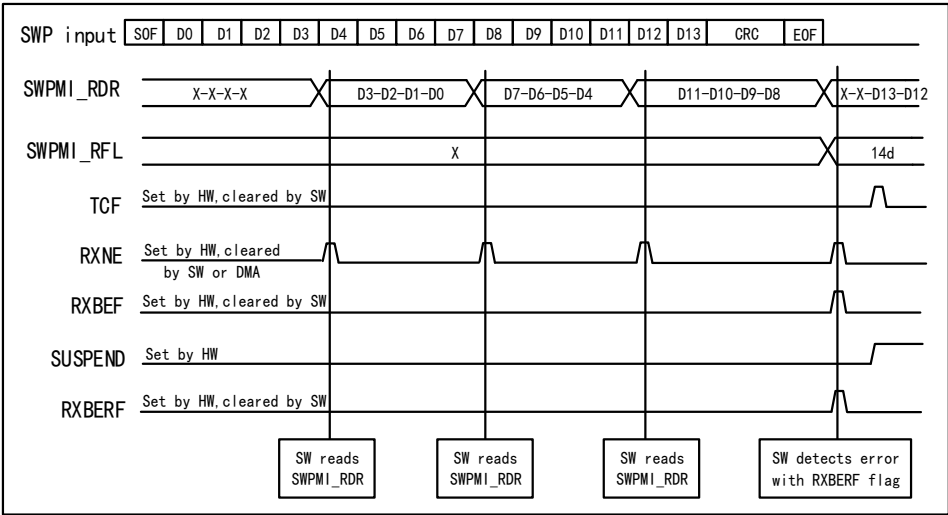
当 RXOVRF 标志与 RXNE 标志同时置 1，在上溢事件发生后，将执行两次 R32_SWPMI_RDR 读取操作。此现象表明：至少有一个接收到的字节已丢失，且 R32_SWPMI_RDR 中加载的字节包含了上溢发生前刚接收到的字节。

在多软件缓冲区模式下，如果已接收帧的最后一个字将 RXOVRF 标志置 1，那么当前帧和下一帧将上溢位（第 8 个字的位 25）置 1。

有效负载接收期间发生 CRC 错误

当接收到两个 CRC 字节后，若 CRC 错误，那么在完成 EOF 接收后，R32_SWPMI_ISR 寄存器 RXBERF 标志会置 1。此时，如果 R32_SWPMI_IER 寄存器中的 RXBEIE 位置 1，那么将会触发一个中断，请参考图 38-10。为清零 RXBERF 标志，用户必须将 R32_SWPMI_ICR 中的 CRXBERF 位置 1。

图 38-10 SWPMI 单缓冲区模式接收（带 CRC 错误）



有效负载接收期间丢失或损坏填充位

当有效负载中的填充位出现丢失或损坏的情况，在接收到 EOF 后，R32_SWPMI_ISR 中的 RXBERF 和 RXBFF 标志会置 1。

接收损坏的 EOF

在接收到 SOF 后，SWPMI 将开始累计接收到的字节，直至接收到 EOF（忽略任何可能的 SOF）。接收到 EOF 后，SWPMI 将准备就绪，开始新一轮的帧接收，并等待下一个 SOF。若 EOF 损坏，SWPMI_ISR 寄存器中的 RXBERF 和 RXBFF 标志将在下一个 EOF 完成接收后置 1。

注：当接收到损坏的 EOF 时，仍会继续接收有效负载。因此，若接收到的字节数超过 30，有效负载中的字节数可取值 31。有效负载中的字节数在 R32_SWPMI_RFL 寄存器中读取，或者根据工作模式不同，在 RAM 存储器缓冲区的第 8 个字的位[23:16]中读取。

38.3.9 回送模式

可以采用回送模式进行测试。用户必须将 R32_SWPMI_CR 寄存器中的 LPBK 位置 1，以启用回送模式。启用回送模式后，SWP_TX 和 SWP_RX 信号将被连接。因此，所有通过 SWPMI 发送的帧将被收回。

38.3.10 SWPMI 中断

所有 SWPMI 中断均连接至 PFIC。要使能 SWPMI 中断，需按照以下流程操作：

- 1) 配置 PFIC 中的 SWPMI 中断通道并将其使能；
- 2) 配置 SWPMI 以生成 SWPMI 中断（请参考 R32_SWPMI_IER 寄存器）。

表 38-2 中断请求

中断事件	中断事件标志	中断使能控制位
接收缓冲区已满	RXBFF	RXBFIE
发送缓冲区为空	TXBEF	TXBEIE
接收缓冲区错误 (CRC 错误)	RXBERF	RXBEIE
接收缓冲区上溢	RXOVRF	RXBOVERIE
发送缓冲区下溢	TXUNRF	TXBUNREIE
接收数据寄存器非空	RXNE	RIE
发送数据寄存器已满	TXE	TIE
传输完成标志	TCF	TCIE

从器件恢复标志	SRF	SRIE
收发器就绪标志	RDYF	RDYIE

38.4 寄存器描述

表 38-3 SWPMI 相关寄存器列表

名称	访问地址	描述	复位值
R32_SWPMI_CR	0x40008400	SWPMI 配置/控制寄存器	0x00000000
R32_SWPMI_BRR	0x40008404	SWPMI 比特率寄存器	0x00000001
R32_SWPMI_ISR	0x4000840C	SWPMI 中断和状态寄存器	0x000002C2
R32_SWPMI_ICR	0x40008410	SWPMI 中断标志清零寄存器	0x00000000
R32_SWPMI_IER	0x40008414	SWPMI 中断使能寄存器	0x00000000
R32_SWPMI_RFL	0x40008418	SWPMI 接收帧长度寄存器	0x00000000
R32_SWPMI_TDR	0x4000841C	SWPMI 发送数据寄存器	0x00000000
R32_SWPMI_RDR	0x40008420	SWPMI 接收数据寄存器	0x00000000
R32_SWPMI_OR	0x40008424	SWPMI 选项寄存器	0x00000000

38.4.1 SWPMI 配置/控制寄存器 (R32_SWPMI_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				SWPT EN	DE ACT	Reserved			SWP ACT	LP BK	TXM ODE	RXM ODE	TX DMA	RX DMA	

位	名称	访问	描述	复位值
[31:12]	Reserved	RO	保留。	0
11	SWPTEN	RW	单线协议主收发器使能: 1: SWPMI_I0 收发器由 SWPMI 控制; 0: SPWMI_I0 引脚由 GPIO 控制器控制。	0
10	DEACT	RW	单线协议主接口禁用,用于请求 SWP “已禁用”状态。除了可能传入的“从器件恢复”状态会使 SWP 保持为“已激活”状态外,将该位置 1 与清零 SWPACT 的作用相同。	0
[9:6]	Reserved	RO	保留。	0
5	SWPACT	RW	单线协议主接口激活: 1: SWPMI_I0 被释放, SWP 总线切换到“已挂起”状态; 0: SWPMI_I0 被下拉至地, SWP 总线切换到“已禁用”状态。 将 SWPACT 位置 1, 必须先清零 DEACT 位。	0
4	LPBK	RW	回送模式使能: 1: 使能回送模式; 0: 禁止回送模式。	0

3	TXMODE	RW	发送缓冲模式： 1：针对发送将 SWPMI 配置为多软件缓冲区模式； 0：针对发送将 SWPMI 配置为单软件缓冲区模式。 注：SWPACT 位置 1 时，该位无法写入。	0
2	RXMODE	RW	接收缓冲模式： 1：针对接收将 SWPMI 配置为多软件缓冲区模式； 0：针对接收将 SWPMI 配置为单软件缓冲区模式。 注：SWPACT 位置 1 时，该位无法写入。	0
1	TXDMA	RW	发送 DMA 使能： 1：针对发送使能 DMA； 0：针对发送禁止 DMA。 注：如果将已发送帧的有效负载大小指定为 0x00（在帧的第一个字的 TDR 最低有效字节中），TXDMA 将自动清零。TXDMA 在发生下溢事件时（R32_SWP_ISR 寄存器的 TXUNRF 标志置 1 时）也会自动清零。	0
0	RXDMA	RW	接收 DMA 使能： 1：针对接收使能 DMA； 0：针对接收禁止 DMA。	0

表 38-4 针对发送/接收选择缓冲区模式

缓冲区模式	无软件缓冲区	单软件缓冲区	多软件缓冲区
RXMODE/TXMODE	X	0	1
RXDMA/TXDMA	0	1	1

38.4.2 SWPMI 比特率寄存器 (R32_SWPMI_BRR)

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								BR[7:0]							

位	名称	访问	描述	复位值
[31:8]	Reserved	R0	保留。	0
[7:0]	BR[7:0]	RW	比特率预分频器，必须按照以下公式，在考虑到 RCC（复位和时钟控制）中编程的 F_{HCLK} 的情况下编程该位域以设置 SWP 总线比特率： $FSWP = F_{HCLK} / ((BR[7:0] + 1) \times 4)$ 注：编程的比特率必须处于以下范围内：100 kbit/s~2Mbit/s。 R32_SWPMI_CR 寄存器中的 SWPACT 位置 1 时，不能写入 BR[7:0]。	0x01

38.4.3 SWPMI 中断和状态寄存器 (R32_SWPMI_ISR)

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				RD YF	DEA CTF	SUSP	SRF	TCF	TXE	RXNE	TXU NRF	RXO VRF	RXB ERF	TXB EF	RXB FF

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0
11	RDYF	R0	收发器就绪标志：当收发器就绪时，该位立即由硬件置 1。通过将 R32_SWPMI_CR 寄存器中的 SWPTEN 位置 1 以使能 SWPMI_I0 收发器后，软件必须等待此标志置 1 后再将 SWPACT 位置 1 以激活 SWP 总线。 1：收发器已就绪； 0：收发器未就绪。	0
10	DEACTF	R0	已禁用标志： 1：SWP 总线处于“已禁用”状态； 0：SWP 总线处于“已激活”状态或“已挂起”状态。	0
9	SUSP	R0	挂起标志： 1：SWP 总线处于“已挂起”或“已禁用”状态； 0：SWP 总线处于“已激活”状态。	1
8	SRF	R0	从器件恢复标志：该位由硬件置 1，用于指示检测到“从器件恢复”状态。该位由软件清零，方法是向 R32_SWPMI_ICR 寄存器中的 CSRF 位写入 1。 1：在 SWP 总线处于“已挂起”状态时检测到“从器件恢复”状态； 0：未检测到“从器件恢复”状态。	0
7	TCF	R0	传输完成标志：当发送和接收均已完成并且 SWP 切换到“已挂起”状态时，该标志将立即由硬件置 1。该位由软件清零，方法是向 R32_SWPMI_ICR 寄存器中的 CTCF 位写入 1。 1：发送和接收均已完成并且 SWP 切换到“已挂起”状态； 0：发送或接收未完成。	1
6	TXE	R0	发送数据寄存器为空： 1：已发送写入到发送数据寄存器 R32_SWPMI_TDR 中的数据，可再次对 R32_SWPMI_TDR 进行写操作； 0：尚未发送写入到发送数据寄存器 R32_SWPMI_TDR 中的数据。	1
5	RXNE	R0	接收数据寄存器非空： 1：R32_SWPMI_RDR 寄存器中接收到的数据已准备就绪可供读取；	0

			0: R32_SWPMI_RDR 寄存器中未接收到数据。	
4	TXUNRF	R0	发送下溢错误标志: 该标志由硬件置 1, 用于指示在有效负载发送期间发生下溢, 即软件或 DMA 未及时对 R32_SWPMI_TDR 进行写操作。该标志由软件清零, 方法是向 R32_SWPMI_ICR 寄存器中的 CTXUNRF 位写入 1。 1: 在发送期间检测到下溢错误; 0: 发送期间无下溢错误。	0
3	RXOVRF	R0	接收上溢错误标志: 该标志由硬件置 1, 用于指示在有效负载接收期间发生上溢, 即软件或 DMA 未及时对 R32_SWPMI_RDR 进行读操作。该标志由软件清零, 方法是向 R32_SWPMI_ICR 寄存器中的 CRXOVRF 位写入 1。 1: 在接收期间检测到上溢错误; 0: 接收期间无上溢错误。	0
2	RXBERF	R0	接收 CRC 错误标志: 该标志由硬件置 1, 用于指示接收到的帧中存在 CRC 错误。它与 RXBFF 标志同步置 1。该位由软件清零, 方法是向 R32_SWPMI_ICR 寄存器中的 CRXBERF 位写入 1。 1: 在接收期间检测到 CRC 错误; 0: 接收期间无 CRC 错误。	0
1	TXBEF	R0	发送缓冲区为空标志: 该标志由硬件置 1, 用于指示无需进一步进行 R32_SWPMI_TDR 更新来完成当前帧发送。该位由软件清零, 方法是向 R32_SWPMI_ICR 寄存器中的 CTXBEF 位写入 1。 1: 帧发送缓冲区已清空; 0: 帧发送缓冲区尚未清空。	1
0	RXBFF	R0	接收缓冲区已满标志: 当 R32_SWPMI_RDR 中包含接收帧的最后一个字时, 该标志由硬件置 1。该标志由软件清零, 方法是向 R32_SWPMI_ICR 寄存器中的 CRXBFF 位写入 1。 1: 接收帧的最后一个字已传到 R32_SWPMI_RDR; 0: 接收帧的最后一个字尚未传到 R32_SWPMI_RDR。	0

38.4.4 SWPMI 中断标志清零寄存器 (R32_SWPMI_ICR)

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				CRDYF	Reserved		CSRF	CTCF	Reserved		CTXU NRF	CRXO VRF	CRXBE RF	CTX BEF	CRX BFF

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0

11	CRDYF	RW1Z	清零收发器就绪标志：将 1 写入此位时，R32_SWPMI_ISR 寄存器中的 RDYF 标志将清零向该位，写入 0 无任何影响。	0
[10:9]	Reserved	R0	保留。	0
8	CSRF	RW1Z	清零从器件恢复标志：将 1 写入此位时，R32_SWPMI_ISR 寄存器中的 SRF 标志将清零向该位，写入 0 无任何影响。	0
7	CTCF	RW1Z	清零传输完成标志：将 1 写入此位时，R32_SWPMI_ISR 寄存器中的 TCF 标志将清零向该位，写入 0 无任何影响。	0
[6:5]	Reserved	R0	保留。	0
4	CTXUNRF	RW1Z	清零发送下溢错误标志：将 1 写入此位时，R32_SWPMI_ISR 寄存器中的 TXUNRF 标志将清零向该位，写入 0 无任何影响。	0
3	CRXOVRF	RW1Z	清零接收上溢错误标志：将 1 写入此位时，R32_SWPMI_ISR 寄存器中的 RXBOCREF 标志将清零向该位，写入 0 无任何影响。	0
2	CRXBERF	RW1Z	清零接收 CRC 错误标志：将 1 写入此位时，R32_SWPMI_ISR 寄存器中的 RXBERF 标志将清零向该位，写入 0 无任何影响。	0
1	CTXBEF	RW1Z	清零发送缓冲区为空标志：将 1 写入此位时，R32_SWPMI_ISR 寄存器中的 TXBEF 标志将清零向该位，写入 0 无任何影响。	0
0	CRXBFF	RW1Z	清零接收缓冲区已满标志：将 1 写入此位时，R32_SWPMI_ISR 寄存器中的 RXBFF 标志将清零向该位，写入 0 无任何影响。	0

38.4.5 SWPMI 中断使能寄存器 (R32_SMPMI_IER)

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				RDYIE	Reserved		SRIE	TCIE	TIE	RIE	TXUN RIE	RXOV RIE	RXBE RIE	TXB EIE	RXB FIE

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0
11	RDYIE	RW	收发器就绪中断使能： 1: R32_SWPMI_ISR 寄存器中的 RDYF 标志置 1 时生成 SWPMI 中断； 0: 禁止中断。	0
[10:9]	Reserved	R0	保留。	0
8	SRIE	RW	从器件恢复中断使能： 1: R32_SWPMI_ISR 寄存器中的 SRF 标志置 1 时生	0

			成 SWPMI 中断; 0: 禁止中断。	
7	TCIE	RW	发送完成中断使能: 1: R32_SWPMI_ISR 寄存器中的 TCF 标志置 1 时生 SWPMI 中断; 0: 禁止中断。	0
6	TIE	RW	发送中断使能: 1: R32_SWPMI_ISR 寄存器中的 TXE 标志置 1 时生 成 SWPMI 中断; 0: 禁止中断。	0
5	RIE	RW	接收中断使能: 1: R32_SWPMI_ISR 寄存器中的 RXNE 标志置 1 时生 成 SWPMI 中断; 0: 禁止中断。	0
4	TXUNRIE	RW	发送下溢错误中断使能: 1: R32_SWPMI_ISR 寄存器中的 TXBUNRF 标志置 1 时生成 SWPMI 中断; 0: 禁止中断。	0
3	RXOVRIE	RW	接收上溢错误中断使能: 1: R32_SWPMI_ISR 寄存器中的 RXBOVRF 标志置 1 时生成 SWPMI 中断; 0: 禁止中断。	0
2	RXBERIE	RW	接收 CRC 错误中断使能: 1: R32_SWPMI_ISR 寄存器中的 RXBERF 标志置 1 时 生成 SWPMI 中断; 0: 禁止中断。	0
1	TXBEIE	RW	发送缓冲区为空中断使能: 1: R32_SWPMI_ISR 寄存器中的 TXBEF 标志置 1 时 生成 SWPMI 中断; 0: 禁止中断。	0
0	RXBFIIE	RW	接收缓冲区已满中断使能: 1: R32_SWPMI_ISR 寄存器中的 RXBFF 标志置 1 时 生成 SWPMI 中断; 0: 禁止中断。	0

38.4.6 SWPMI 接收帧长度寄存器 (R32_SWPMI_RFL)

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
Reserved											RFL[4:0]				

位	名称	访问	描述	复位值
[31:5]	Reserved	RO	保留。	0

[4:0]	RFL[4:0]	R0	接收帧长度：已接收帧的有效负载中的数据字节数。两个最低有效位 RFL[1:0] 表示最后一次 R32_SWPMI_RDR 寄存器读取操作的相关字节数。	0

38.4.7 SWPMI 发送数据寄存器 (R32_SWPMI_TDR)

偏移地址：0x1C

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

位	名称	访问	描述	复位值
[31:0]	TD[31:0]	WO	发送数据：对该寄存器进行写操作会触发 SOF 发送或下一次有效负载数据发送，并会清零 TXE 标志。	0

38.4.8 SWPMI 接收数据寄存器 (R32_SWPMI_RDR)

偏移地址：0x20

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

位	名称	访问	描述	复位值
[31:0]	RD[31:0]	R0	已接收数据：包含接收到的数据，读取该寄存器会清零 RXNE 标志。	0

38.4.9 SWPMI 选项寄存器 (R32_SWPMI_OR)

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														Reserved	SWP_TBYP

位	名称	访问	描述	复位值
[31:2]	Reserved	R0	保留。	0
1	Reserved	R0	保留。	0
0	SWP_TBYP	RW	禁止内部收发器： 1：禁止内部收发器。SWP_RX、SWP_TX 和 SWPMI_SUP 信号作为 GPIO 上的复用功能可供使用。通过选择	0

			此配置，可以连接外部收发器； 0：使能内部收发器。SWPMI 的外部接口为 SWPMI_IO(SWP_RX、SWP_TX 和 SWP_SUP 信号在 GPIO 上不可用)。	

第 39 章 电子签名 (ESIG)

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。

39.1 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。

39.2 寄存器描述

表 39-1 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	0xFFFF
R32_ESIG_UNIID1	0x1FFFF7E8	UID 寄存器 1	0xFFFFFFFF
R32_ESIG_UNIID2	0x1FFFF7EC	UID 寄存器 2	0xFFFFFFFF
R32_ESIG_UNIID3	0x1FFFF7F0	UID 寄存器 3	0xFFFFFFFF

39.2.1 闪存容量寄存器 (R16_ESIG_FLACAP)

位	名称	访问	描述	复位值
[15:0]	F_SIZE[15:0]	RO	以 Kbyte 为单位的闪存容量。 例：0x0080 = 128K 字节	X

39.2.2 UID 寄存器 (R32_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[31:0]	RO	UID 的 0-31 位。	X

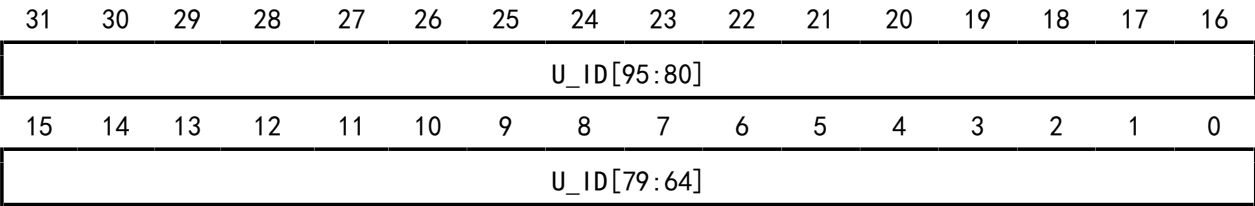
39.2.3 UID 寄存器 (R32_ESIG_UNIID2)

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[63:32]	R0	UID 的 32-63 位。	X

39. 2. 4 UID 寄存器 (R32_ESIG_UNIID3)



位	名称	访问	描述	复位值
[31:0]	U_ID[95:64]	R0	UID 的 64-95 位。	X

第 40 章 灵活存储控制器（FMC）

本章模块描述仅适用于 CH32H417 微控制器产品。

灵活存储控制器（FMC）是一种用于管理外部存储器的控制器，包含：NAND 存储控制器、NOR/PSRAM 存储控制器、同步 DRAM（SDRAM/Mobile LPDDR SDRAM）控制器。

40.1 主要特征

FMC 用于管理扩展可连接不同类型的存储器，包括：SDRAM、NAND Flash、同步/异步静态存储器。其主要目的是：满足外部存储器设备的访问时间要求；将 HB 数据通信事务转换为适当的外部设备协议。

所有外部存储器共享地址、数据和控制信号，但各自拥有独立的片选信号。FMC 一次只能访问一个外部设备。

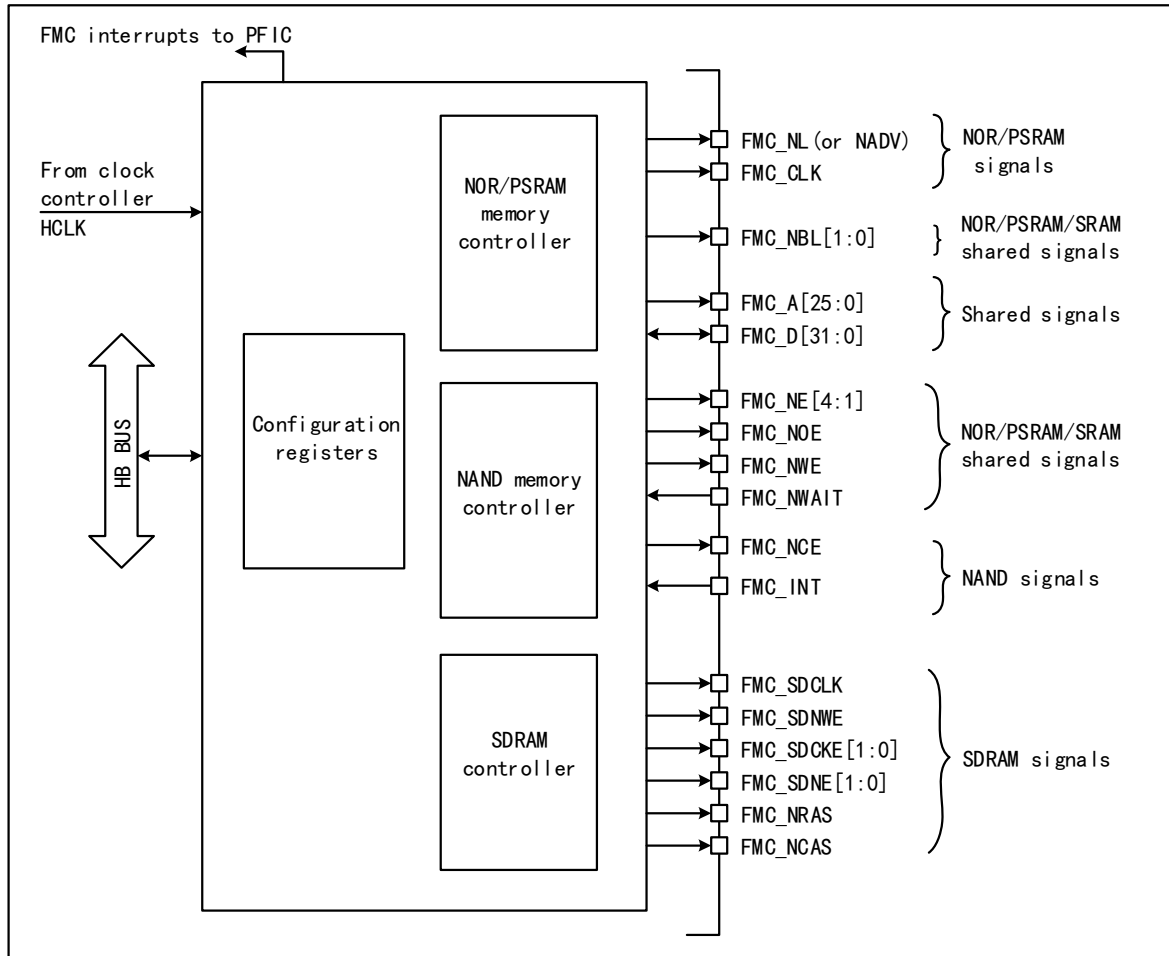
FMC 控制器的主要特征如下：

- 连接静态存储器映射的器件：
 - 静态随机访问存储器（SRAM）
 - PSRAM（4 个存储区域）
 - NOR Flash/OneNAND Flash
 - 带有硬件 ECC 的 NAND Flash 存储器，可检查多达 8KB 的数据
- 连接同步 DRAM（SDRAM/Mobile LPDDR SDRAM）存储器
- 外部异步等待控制
- 16*32 位深度写 FIFO
- 每个存储区域可独立配置
- 每个存储区域有独立片选控制
- 具有 8 位、16 位或 32 位宽的存储器数据总线
- 可编程连续时钟输出以支持异步和同步访问
- SDRAM 控制器具有可缓存的 6*32 位深度读 FIFO（6*14 位地址标记）
- 写使能和字节通道选择输出，可配合 PSRAM、SRAM 和 SDRAM 器件使用
- 支持突发模式，能够更快速地访问同步设备（如 NOR Flash、PSRAM 和 SDRAM）

40.2 概述

如下图所示，FMC 包含 4 个主要模块：HB 接口、SDRAM 控制器、NAND 控制器、NOR Flash/PSRAM/SRAM 控制器。

图 40-1 FMC 的结构框图



40.3 HB 接口

HB 从设备接口允许 CPU 和其它 HB 总线主设备访问外部存储器。HB 事务会转换为外部器件协议。特别是，当所选外部存储器的宽度为 16 位或 8 位时，那么在 HB 中的 32 位宽事务将被分割成多个连续的 16 或 8 位访问。连续访问之间，FMC 片选（FMC_NEx）不会翻转，除非在启用扩展模式的访问模式 D 情况下。

FMC 在以下情况下会产生 HB 错误：

- 读取或写入未使能的 FMC 存储区域。
- 违反 SDRAM 地址范围（访问保留的地址范围）时。
- 向写保护的 SDRAM 存储区域（R32_FMC_SDCRx 寄存器中 WP 位置 1）写入时。
- 在 R32_FMC_BCRx 寄存器中的 FACCEN 位复位时读取或写入 NOR Flash 存储区域。

此 HB 错误的影响具体取决于尝试进行读、写访问的 HB 主设备。

- 如果为 DMA 控制器，则会生成 DMA 传输错误，并会自动禁止相应的 DMA 通道。

HB 时钟（HCLK）是 FMC 的参考时钟。

40.3.1 支持的存储器和事务

通用事务规则

所请求的 HB 事务数据宽度为 8、16、32 或 256 位，然而访问的外部器件的数据宽度是固定的。这种差异可能会引起数据宽度不匹配的问题。为避免上述问题的发生，需遵循的事务规则如下：

- HB 事务数据宽度和存储器数据宽度相等：正常运行。

- HB 事务数据宽度大于存储器宽度：FMC 会将 HB 事务拆分成多个较小的连续存储器访问，以匹配外部数据宽度。在连续访问之间，FMC 片选（FMC_NEx）不会翻转。
- HB 事务数据宽度小于存储器宽度：传输的一致性可能会受到影响，这取决于外部设备的类型。
 - 访问具备字节选择功能的设备（ROM、SRAM、SDRAM、PSRAM）：FMC 允许读、写事务，并通过其字节选择通道 NBL[1:0]访问正确的数据；通过 NBL[1:0]寻址要写入的字节；读取所有存储器字节（在读取事务期间 NBL[1:0]保持低电平），然后丢弃无效字节。
 - 访问不具有字节选择功能的器件（NOR、NAND Flash）：当需要对 16 位宽的 FLASH 存储器进行字节访问时，会出现这种情况。由于这些器件不支持字节模式访问（只能向 Flash 读取或写入 16 位字），因此允许读事务而不允许写事务（控制器会读取全部 16 位存储器字，但只使用所需字节）。

NOR Flash/PSRAM 和 SDRAM 的回卷支持

由于并非所有主器件都能发出回卷事务，因此同步存储器必须按未定义长度的线性突发模式进行配置。如果主器件生成 HB 回卷事务：

- 读操作将分为两个线性突发事务。
- 如果使能写 FIFO，写操作将分为两个线性突发事务；如果禁止写 FIFO，写操作将分为多个线性突发事务。

40.4 外部器件地址映射

从 FMC 角度，外部存储器被划分为每个 256MB 的固定大小的存储区域，如下图所示：

图 40-2 FMC 存储区域

Address	Bank	Supported memory type
0X6000 0000	Bank 1 4×64 MB	NOR/PSRAM
0X6FFF FFFF		
0X7000 0000	Bank 2 FMC Not used	
0X7FFF FFFF		
0X8000 0000	Bank 3 4×64 MB	NAND Flash memory
0X8FFF FFFF		
0X9000 0000	Bank 4 FMC Not used	
0X9FFF FFFF		
0XC000 0000	SDRAM Bank 1 4×64 MB	SDRAM
0XCFFF FFFF		
0XD000 0000		
0XDFFF FFFF		
	SDRAM Bank 2 4×64 MB	

存储区域 1 可连接多达 4 个 NOR Flash 或 PSRAM 器件。此存储区域被划分为 4 个 NOR/PSRAM/SR 子区域，带 4 个专用片选信号：存储区域 1 NOR/PSRAM1、存储区域 1 NOR/PSRAM2、存储区域 1 NOR/PSRAM3、存储区域 1 NOR/PSRAM4。存储区域 3 用于连接 NAND Flash 器件。此空间的 MPU 存储器特性必须由软件重新配置到器件中。存储区域 4 和 5 用于连接 SDRAM 器件（每个存储区域一个器件）。

对于每个存储区域，所要使用的存储器类型可由用户应用程序通过配置寄存器配置。

FMC 存储区域映射可通过 R32_FMC_BCR1 寄存器中的 BMP[1:0]位修改。下表显示了用于交换 NOR/PSRAM 存储区域与 SDRAM 存储区域或者重映射 SDRAM 存储区域 2 的配置，从而允许在两个不同的地址映射中访问 SDRAM 存储区域。

表 40-1 FMC 存储区域映射选项

起始地址-结束地址	BMP[1:0]=00 (默认映射)	BMP[1:0]=01 交换 NOR/PSRAM 与 SDRAM 存储区域
0x60000000-0x6FFFFFFF	NOR/PSRAM 存储区域	SDRAM 存储区域 1
0x70000000-0x7FFFFFFF	FMC 未使用	
0x80000000-0x8FFFFFFF	NAND 存储区域	NAND 存储区域
0x90000000-0x9FFFFFFF	FMC 未使用	
0xC0000000-0xCFFFFFFF	SDRAM 存储区域 1	NOR/PSRAM 存储区域
0xD0000000-0xDFFFFFFF	SDRAM 存储区域 2	SDRAM 存储区域 2

40.4.1 NOR/PSRAM 地址映射

HADDR[27:26]位用于从表 40-1 中所示的四个存储区域之中选择其中一个存储区域。

表 40-2 NOR/PSRAM 存储区域选择

HADDR[27:26] ⁽¹⁾	选择的存储区域
00	存储区域 1NOR/PSRAM1
01	存储区域 1NOR/PSRAM2
10	存储区域 1NOR/PSRAM3
11	存储区域 1NOR/PSRAM4

注 1: HADDR 是 HB 内部地址线，但也会参与对外部存储器的寻址。

HADDR[25:0]位包含外部存储器地址。由于 HADDR 为字节地址，而存储器以字为单位寻址，所以实际向存储器发送的地址会根据存储器数据宽度不同而变化，如下表所示。

表 40-3 NOR/PSRAM 存储区域地址

存储器宽度 ⁽¹⁾	向存储器发出的数据地址	最大存储器容量(位)
8 位	HADDR[25:0]	64MB×8=512Mb
16 位	HADDR[25:1]>>1	64MB/2×16=512Mb
32 位	HADDR[25:2]>>2	64MB/4×32=512Mb

注 1: 当外部存储器的宽度为 16 位，FMC 将使用内部的 HADDR[25:1]地址来作为对外部存储器的寻址地址 FMC_A[24:0]。当存储器宽度为 32 位，FMC 将使用内部的 HADDR[25:2]地址进行外部寻址。无论外部存储器的宽度是多少，FMC_A[0]都应连接到外部存储器地址 A[0]。

40.4.2 NAND Flash 地址映射

NAND 存储区域分为几个存储器区域，如下表所示。

表 40-4 NAND 存储映射和时序寄存器

起始地址	结束地址	FMC 存储区域	存储空间	时序寄存器
0x88000000	0x8BFFFFFF	存储区域 3-NAND Flash	特性区	R32_FMC_PATT (0x8C)

0x80000000	0x83FFFFFF		通用区	R32_FMC_PMEM (0x88)

对于 NAND Flash 存储器，通用区和特性区存储空间分为三个部分，均位于低位 256KB 中，如下表所示：

表 40-5 NAND 存储区域选择

部分名称	HADDR[17:16]	地址范围
地址区域	1X	0x020000-0x03FFFF
命令区域	01	0x010000-0x01FFFF
数据区域	00	0x000000-0x0FFFFF

数据区域：通用/特性存储空间中的第一个 64KB；命令区域：通用/特性存储空间中的第二个 64KB；地址区域：通用/特性存储空间中的下一个 128KB。

应用程序软件使用这 3 个区域来访问 NAND Flash 存储器：

- 向 NAND Flash 发送命令，软件必须将命令值写入命令区域中的任意存储单元。
- 指定读取或写入的 NAND Flash 地址，软件必须将地址值写入地址区域中的任意存储单元。由于地址的长度可以是 4 或 5 个字节（具体取决于实际存储器大小），要指定完整的地址，需要对地址区域执行多个连续写入操作。
- 读取或写入数据，软件将从数据区域中的任意存储单元读取数据，或向其写入数据。

由于 NAND Flash 存储器会自动递增地址，因此无需递增数据区域的地址即可访问连续存储器位置。

40.4.3 SDRAM 地址映射

HADDR[28] 位（内部 HB 地址线 28）用于选择两个存储区域之一，如表 40-6 所示。

表 40-6 SDRAM 存储区域选择

HADDR[28]	选择的存储区域	控制寄存器	时序寄存器
0	SDRAM Bank1	R32_FMC_SDCR1	R32_FMC_SDTR1
1	SDRAM Bank2	R32_FMC_SDCR2	R32_FMC_SDTR2

下表显示了 13 位行和 11 位列配置的 SDRAM 映射。

表 40-7 SDRAM 地址映射

存储器宽度 ⁽¹⁾	内部存储区域	行地址	列地址 ⁽²⁾	最大存储器容量 (MB)
8 位	HADDR[25:24]	HADDR[23:11]	HADDR[10:0]	64MB:4x8Kx2K
16 位	HADDR[26:25]	HADDR[24:12]	HADDR[11:1]	128MB:4x8Kx2Kx2
32 位	HADDR[27:26]	HADDR[25:13]	HADDR[12:2]	256MB:4x8Kx2Kx4

注 1：连接 16 位存储器时，FMC 内部使用 HADDR[11:1] 内部 HB 地址线进行外部寻址。连接 32 位存储器时，FMC 内部使用 HADDR[12:2] 地址线进行外部寻址。无论外部存储器的宽度是多少，FMC_A[0] 都必须连接到外部存储器地址 A[0]。

注 2：不支持 AutoPrecharge。FMC_A[10] 必须连接到外部存储器地址 A[10]，但始终保持为“低电平”。

HADDR[27:0] 位根据 SDRAM 控制器配置，将转换为外部 SDRAM 地址。数据大小：8、16 或 32 位；列大小：8、9、10 或 11 位；行大小：11、12 或 13 位；内部存储区域数量：2 个或 4 个内部存储区域。下表显示了不同 SDRAM 控制器配置的 SDRAM 地址映射。

表 40-8 8 位数据总线宽度时的 SDRAM 地址映射^{(1) (2)}

行大小	HADDR (HB 内部地址线)

配置	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
11 位 行大小 配置	Res.							Bank [1:0]		Row[10:0]										Column[7:0]								
	Res.					Bank [1:0]		Row[10:0]										Column[8:0]										
	Res.				Bank [1:0]		Row[10:0]										Column[9:0]											
	Res.			Bank [1:0]		Row[10:0]										Column[10:0]												
12 位 行大小 配置	Res.					Bank [1:0]		Row[11:0]										Column[7:0]										
	Res.				Bank [1:0]		Row[11:0]										Column[8:0]											
	Res.			Bank [1:0]		Row[11:0]										Column[9:0]												
	Res.		Bank [1:0]		Row[11:0]										Column[10:0]													
13 位 行大小 配置	Res.				Bank [1:0]		Row[12:0]										Column[7:0]											
	Res.			Bank [1:0]		Row[12:0]										Column[8:0]												
	Res.		Bank [1:0]		Row[12:0]										Column[9:0]													
	Res.	Bank [1:0]		Row[12:0]										Column[10:0]														

注 1: BANK[1:0] 为存储区域地址 BA[1:0]。当仅使用 2 个内部存储区域时, BA1 必须始终设置为 0。

注 2: 访问保留的 (Res.) 地址范围会生成 HB 错误。

表 40-9 16 位数据总线宽度时的 SDRAM 地址映射^{(1) (2)}

行大小 配置	HADDR (HB 内部地址线)																											
	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
11 位 行大小 配置	Res.						Bank [1:0]		Row[10:0]											Column[7:0]							BMO ⁽³⁾	
	Res.					Bank [1:0]		Row[10:0]											Column[8:0]							BMO		
	Res.				Bank [1:0]		Row[10:0]											Column[9:0]							BMO			
	Res.			Bank [1:0]		Row[10:0]											Column[10:0]							BMO				
12 位 行大小 配置	Res.					Bank [1:0]		Row[11:0]											Column[7:0]							BMO		
	Res.				Bank [1:0]		Row[11:0]											Column[8:0]							BMO			
	Res.			Bank [1:0]		Row[11:0]											Column[9:0]							BMO				

	Res.	Bank [1:0]	Row[11:0]	Column[10:0]	BMO
13 位 行大小 配置	Res.	Bank [1:0]	Row[12:0]	Column[7:0]	BMO
	Res.	Bank [1:0]	Row[12:0]	Column[8:0]	BMO
	Res.	Bank [1:0]	Row[12:0]	Column[9:0]	BMO
	Res.	Bank [1:0]	Row[12:0]	Column[10:0]	BMO

注 1: BANK[1:0] 为存储区域地址 BA[1:0]。当仅使用 2 个内部存储区域时, BA1 必须始终设置为 0。

注 2: 访问保留的空间 (Res.) 会生成 HB 错误。

注 3: BMO 为 16 位访问的字节屏蔽。

表 40-10 32 位数据总线宽度时的 SDRAM 地址映射 ^{(1) (2)}

行大小 配置	HADDR（HB 内部地址线）																										
	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
11 位 行大小 配置	Res.				Bank [1:0]		Row[10:0]										Column[7:0]				BM[1: 0] ⁽³⁾						
	Res.			Bank [1:0]		Row[10:0]										Column[8:0]				BM[1: 0]							
	Res.		Bank [1:0]		Row[10:0]										Column[9:0]				BM[1: 0]								
	Res.	Bank [1:0]		Row[10:0]										Column[10:0]				BM[1: 0]									
12 位 行大小 配置	Res.			Bank [1:0]		Row[11:0]										Column[7:0]				BM[1: 0]							
	Res.		Bank [1:0]		Row[11:0]										Column[8:0]				BM[1: 0]								
	Res.	Bank [1:0]		Row[11:0]										Column[9:0]				BM[1: 0]									
	Res.	Bank [1:0]		Row[11:0]										Column[10:0]				BM[1: 0]									
13 位 行大小 配置	Res.		Bank [1:0]		Row[12:0]										Column[7:0]				BM[1: 0]								
	Res.		Bank [1:0]		Row[12:0]										Column[8:0]				BM[1: 0]								
	Res.	Bank [1:0]		Row[12:0]										Column[9:0]				BM[1: 0]									
	Bank [1:0]		Row[12:0]										Column[10:0]				BM[1: 0]										

注 1: BANK[1:0] 为存储区域地址 BA[1:0]。当仅使用 2 个内部存储区域时, BA1 必须始终设置为 0。

注 2: 访问保留的空间 (Res.) 会生成 HB 错误。

注 3: BM[1:0] 为 32 位访问的字节屏蔽。

40.5 NOR Flash/PSRAM 控制器

FMC 生成适当的信号时序来驱动以下类型的存储器：

- 异步 SRAM 和 ROM：8 位、16 位、32 位；
- PSRAM（Cellular RAM）：异步模式、同步访问的突发模式、复用或非复用；
- NOR Flash：异步模式、同步访问的突发模式、复用或非复用。

FMC 会为每个存储区域输出唯一的片选信号 NE[4:1]。所有其它信号（地址、数据和控制）均为共享信号。FMC 通过可编程时序支持多种器件，其中：

- 可编程连续时钟（FMC_CLK）输出
- 等待周期可编程（最多 15 个时钟周期）
- 总线周转周期可编程（最多 15 个时钟周期）
- 独立的读和写时序和协议，以支持各种存储器和时序
- 输出使能和写入使能延迟可编程（最多 15 个时钟周期）

为生成 FMC_CLK 连续时钟，存储区域 1 还必须配置为支持同步模式（请参考第 40.5.6 节：NOR/PSRAM 控制寄存器）。由于所有同步存储器均使用同一个时钟，因此在生成连续输出时钟和执行同步访问时，HB 数据大小必须与存储数据宽度（MWID）相同，否则 FMC_CLK 频率将根据 HB 数据事务发生变化（有关 FMC_CLK 分频比公式的信息请参考第 40.5.5 节：同步事务）。

每个存储区域的大小固定，均为 64MB。每个存储区域都通过专用的寄存器配置（请参考第 40.5.6 节：NOR/PSRAM 控制寄存器）。存储器的可编程参数包括访问时间（如下表所示）以及对等待管理的支持（用于在突发模式下访问 NOR Flash 和 PSRAM）。

使用 NOR Flash/PSRAM 控制器读取数据时，只能使用 DMA 256 位读取。

表 40-11 NOR/PSRAM 的可编程访问参数

参数	功能	访问模式	最大值	最小值	单位
地址建立	地址建立阶段的持续时间	异步	0	15	HB 时钟周期（HCLK）
地址保持	地址保持阶段的持续时间	异步，复用 I/O	1	15	HB 时钟周期（HCLK）
数据建立	数据建立阶段的持续时间	异步	1	256	HB 时钟周期（HCLK）
总线周转	总线周转阶段的持续阶段	异步和同步读/写	0	15	HB 时钟周期（HCLK）
时钟分频比	构建一个存储器时钟周期（CLK）所需的 HB 时钟周期（HCLK）数量	同步	2	16	HB 时钟周期（HCLK）
数据延迟	在发出突发的第一个数据前向存储器发出的时钟周期数量	同步	2	17	存储器时钟周期（HCLK）

40.5.1 外部存储器接口信号

通常用于连接 NOR Flash、SRAM 和 PSRAM 的信号请参考表 40-12、表 40-13、表 40-14、表 40-15，前缀“N”标识低电平有效的信号。

表 40-12 非复用 I/O NOR Flash

FMC 信号名称	I/O	功能
CLK	0	时钟（用于同步访问）
A[25:0]	0	地址总线

D[31:0]	I/O	双向数据总线
NE[x]	0	片选, x=1~4
NOE	0	输出使能
NWE	0	写入使能
NL (=NADV)	0	锁存使能 (对于部分 NOR Flash 器件, 此信号也称为地址有效 (NADV))
NWAIT	1	FMC 的 NOR Flash 等待输入信号

最大容量为 512Mb (26 个地址线)。

表 40-13 16 位复用 I/O NOR Flash

FMC 信号名称	I/O	功能
CLK	0	时钟 (用于同步访问)
A[25:16]	0	地址总线
AD[15:0]	I/O	16 位复用, 双向地址/数据总线 (16 位地址 A[15:0] 和数据总线上复用)
NE[x]	0	片选, x=1~4
NOE	0	输出使能
NWE	0	写入使能
NL (=NADV)	0	锁存使能 (对于部分 NOR Flash 器件, 此信号也称为地址有效 (NADV))
NWAIT	1	FMC 的 NOR Flash 等待输入信号

最大容量为 512Mb。

表 40-14 非复用 I/O PSRAM/SRAM

FMC 信号名称	I/O	功能
CLK	0	时钟 (仅用于 PSRAM 同步访问)
A[25:0]	0	地址总线
D[31:0]	I/O	双向数据总线
NE[x]	0	片选, x=1~4 (在 PSRAM 应用中被称作 NCE (Cellular RAM, 即 CRAM))
NOE	0	输出使能
NWE	0	写入使能
NL (=NADV)	0	仅用于 PSRAM 输入的地址有效信号 (存储器信号名称: NADV)
NWAIT	1	PSRAM 发送给 FMC 等待输入信号
NBL[1:0]	0	字节通道输出。字节 0 到字节 3 控制 (高字节和低字节使能)

最大容量为 512Mb。

表 40-15 16 位复用 I/O PSRAM

FMC 信号名称	I/O	功能
CLK	0	时钟 (用于同步访问)
A[25:16]	0	地址总线
AD[15:0]	I/O	16 位复用, 双向地址/数据总线

		(16 位地址 A[15:0] 和数据 D[15:0] 在数据总线上复用)
NE[x]	0	片选, x=1~4 (在 PSRAM 应用中被称作 NCE (Cellular RAM, 即 CRAM))
NOE	0	输出使能
NWE	0	写入使能
NL (=NADV)	0	用于 PSRAM 输入的地址有效信号 (存储器信号名称: NADV)
NWAIT	1	PSRAM 发送给 FMC 等待输入信号
NBL[1:0]	0	字节通道输出。字节 0 到字节 1 控制 (高字节和低字节使能)

最大容量为 512Mb (26 个地址线)。

40.5.2 支持的存储器和事务

当 NOR Flash、PSRAM 和 SRAM 的存储器数据总线宽度为 16 位时所支持的设备、访问模式和事务的示例如下表所示, 本示例中 FMC 不允许 (或不支持) 的事务以灰色显示。

表 40-16 NOR Flash/PSRAM: 支持的存储器和事务示例

设备	模式	R/W	HB 数据大小	存储器 数据大小	是否允许	注释
NOR Flash (复用 I/O 和非复 用 I/O)	异步	R	8	16	是	-
	异步	W	8	16	否	-
	异步	R	16	16	是	-
	异步	W	16	16	是	-
	异步	R	32	16	是	分为 2 次 FMC 访问
	异步	W	32	16	是	分为 2 次 FMC 访问
	异步页	R	-	16	否	不支持该模式
	同步	R	8	16	否	-
	同步	R	16	16	是	-
	同步	R	32	16	是	-
PSRAM (复用 I/O 和 非复用 I/O)	异步	R	8	16	是	-
	异步	W	8	16	是	使用字节通道 NBL[1:0]
	异步	R	16	16	是	-
	异步	W	16	16	是	-
	异步	R	32	16	是	分为 2 次 FMC 访问
	异步	W	32	16	是	分为 2 次 FMC 访问
	异步页	R	-	16	否	不支持该模式
	同步	R	8	16	否	-
	同步	R	16	16	是	-
	同步	R	32	16	是	-
	同步	W	8	16	是	使用字节通道 NBL[1:0]
	同步	W	16/32	16	是	-
SRAM 和 ROM	异步	R	8/16	16	是	-
	异步	W	8/16	16	是	使用字节通道 NBL[1:0]
	异步	R	32	16	是	分为 2 次 FMC 访问
	异步	W	32	16	是	分为 2 次 FMC 访问, 使用字节通道 NBL[1:0]

40.5.3 通用时序规则

信号同步

- 所有控制器输出信号在内部时钟（HCLK）的上升沿变化；
- 在同步模式（读/写）下，所有输出信号在 HCLK 的上升沿变化。无论 CLKDIV 值为何，所有输出信号变化如下：
 - NOEL/NWEL/NEL/NADVL/NADVH/NBLL 地址有效输出在 FMC_CLK 时钟的下降沿变化。
 - NOEH/NWEH/NEH/NOEH/NBLH 地址有效输出在 FMC_CLK 时钟的上升沿变化。

40.5.4 NOR Flash/PSRAM 控制器异步事务

异步静态存储器（NOR Flash、PSRAM、SRAM）

- 信号通过内部时钟 HCLK 进行同步。此时钟不会发送到存储器；
- FMC 总是会先对数据采样，而后再禁止片选信号 NE。这样可以确保符合存储器数据保持时序的要求（片选使能为高到数据转换最短时间通常为 0nS）。
- 如果使能扩展模式（R32_FMC_BCRx 寄存器中的 EXTMOD 位置 1），则最多可提供四种扩展模式（A、B、C、D）。可以混合使用 A、B、C 和 D 模式来进行读取和写入操作。例如，可以在模式 A 下执行读取操作，而在模式 B 下执行写入操作。
- 如果禁用扩展模式（R32_FMC_BCRx 寄存器中的 EXTMOD 位复位），则 FMC 可以在模式 1 或模式 2 下进行运行，如下所述：
 - 当选择 SRAM/PSRAM 存储器类型时，模式 1 为默认模式（R32_FMC_BCRx 寄存器中 MTYP=0x0 或 0x01）；
 - 当选择 NOR 存储器类型时，模式 2 为默认模式（R32_FMC_BCRx 寄存器中 MTYP=0x10）。

模式 1-SRAM/PSRAM（GRAM）

如下图所示，所支持模式的读取和写入事务以及所需的 R32_FMC_BCRx 和 R32_FMC_BTRx/R32_FMC_BWTRx 寄存器配置。

图 40-3 模式 1 读取访问波形

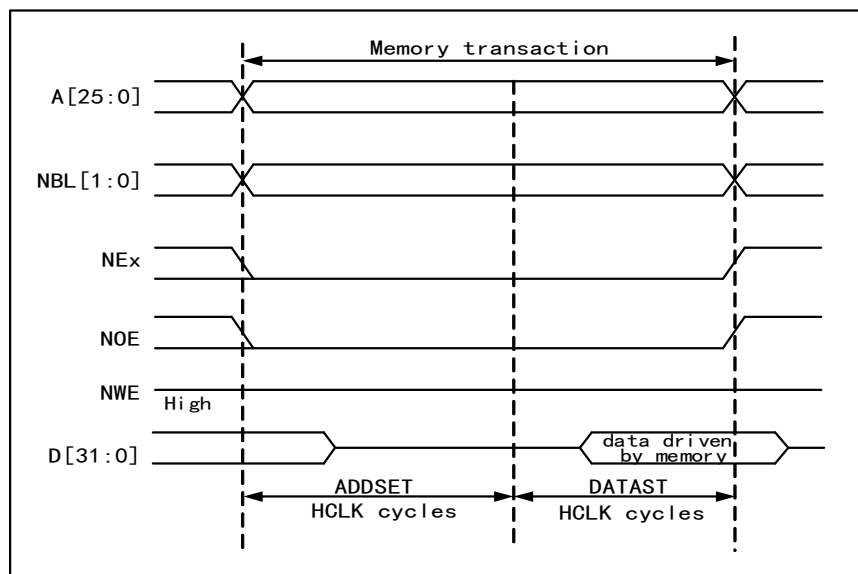
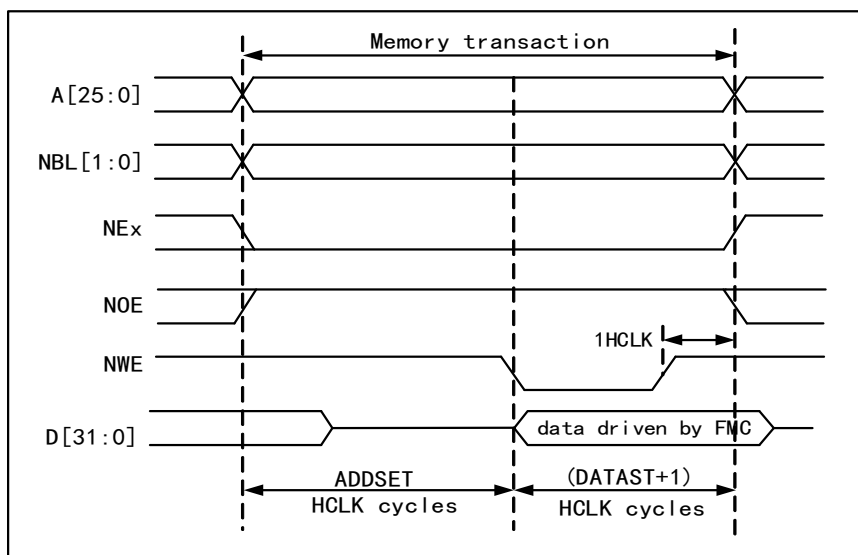


图 40-4 模式 1 写入访问波形



写事务结束时的一个 HCLK 周期利于确保 NEW 上升沿之后的地址和数据保持时间。由于存在此 HCLK 周期, DATAST 值必须大于 0 (DATAST>0)。

表 40-17 R32 FMC BCRx 位字段

位号	位名	要设置的值
31	FMC_EN	0x1
[30:26]	保留	0x00
[25:24]	BMP	0x0 或 0x1
[23:20]	保留	0x0
19	CBURSTRW	0x0（对异步模式没有影响）
[18:16]	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0。
14	EXTMOD	0x0
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	保留	0x0
10	WRAPMOD	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACGEN	无关
[5:4]	MWID	根据需要进行设置
[3:2]	MTYP	根据需要进行设置，0x2 除外（NOR Flash）
1	MUXE	0x0
0	MBKEN	0x1

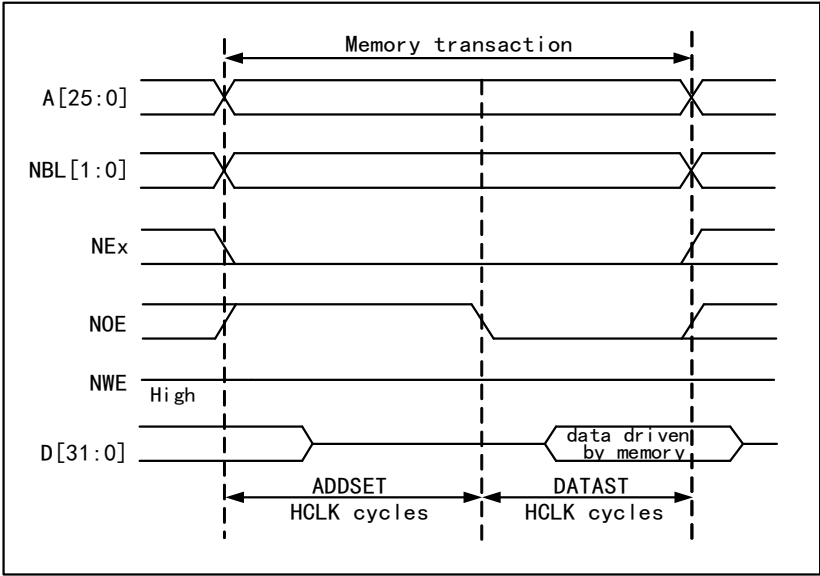
表 40-18 FMC BTR_x 位字段

位号	位名	要设置的值
[31:30]	保留	0x00
[29:28]	ACCMOD	无关

[27:24]	DATLAT	无关
[23:20]	CLKDIV	无关
[19:16]	BUSTURN	NE _x 变为高电平到 NE _x 变为低电平之间的时间(BUSTURN HCLK)
[15:8]	DATAST	第二个访问阶段的持续时间（写入访问为 DATAST+1 个 HCLK 周期，读取访问为 DATAST 个 HCLK 周期）。
[7:4]	ADDHLD	无关
[3:0]	ADDSET	第一个访问阶段的持续时间（ADDSET 个 HCLK 周期）。ADDSET 最小值为 0。

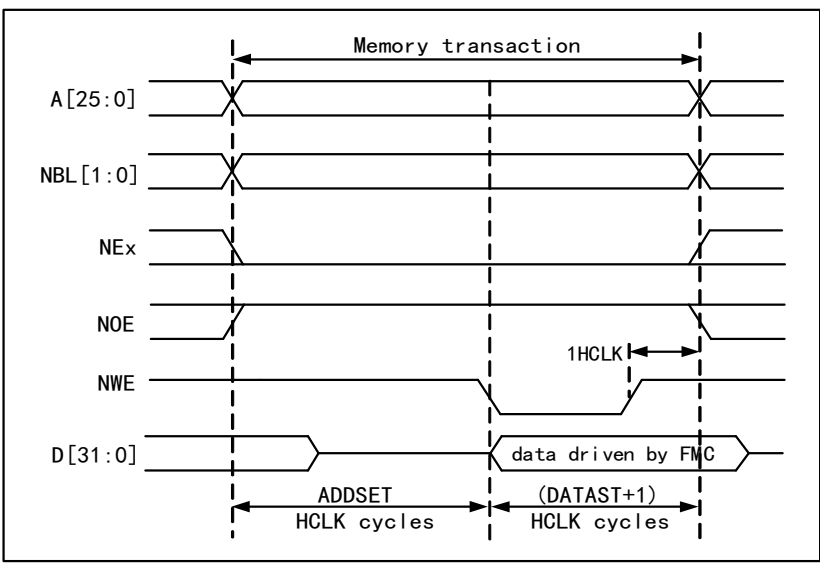
模式 A-SRAM/PSRAM（GRAM）OE 切换

图 40-5 模式 A 读取访问波形



注：NBL[1:0]在进行读取访问时为低电平。

图 40-6 模式 A 写入访问波形



与模式 1 的不同之处在于 NOE 的切换与独立的读取和写入时序。

表 40-19 R32_FMC_BCRx 位字段

位号	位名	要设置的值
31	FMC_EN	0x1
[30:26]	保留	0x00
[25:24]	BMP	0x0 或 0x1
[23:20]	保留	0x0
19	CBURSTRW	0x0 (对异步模式没有影响)
[18:16]	CPSIZE	0x0 (对异步模式没有影响)
15	ASYNCAWAIT	如果存储器支持该特性, 则置为 1。否则, 保持为 0。
14	EXTMOD	0x1
13	WAITEN	0x0 (对异步模式没有影响)
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	无关
[5:4]	MWID	根据需要进行设置
[3:2]	MTYP	根据需要进行设置, 0x2 除外 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

表 40-20 R32_FMC_BTRx 位字段

位号	位名	要设置的值
[31:30]	保留	0x00
[29:28]	ACCMOD	无关
[27:24]	DATLAT	无关
[23:20]	CLKDIV	无关
[19:16]	BUSTURN	NE _x 变为高电平到 NE _x 变为低电平之间的时间 (BUSTURN HCLK)
[15:8]	DATAST	写入访问第二个阶段的持续时间 (DATAST+1 个 HCLK 周期)。
[7:4]	ADDHLD	无关
[3:0]	ADDSET	写入访问第一个阶段的持续时间 (ADDSET+1 个 HCLK 周期)。ADDSET 最小值为 0。

表 40-21 R32_FMC_BWTRx 位字段

位号	位名	要设置的值
[31:30]	保留	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	无关
[23:20]	CLKDIV	无关
[19:16]	BUSTURN	NE _x 变为高电平到 NE _x 变为低电平之间的时间 (BUSTURN

		HCLK)
[15:8]	DATAST	写入访问第二个阶段的持续时间 (DATAST 个 HCLK 周期)。
[7:4]	ADDHLD	无关
[3:0]	ADDSET	写入访问第一个阶段的持续时间 (ADDSET 个 HCLK 周期)。ADDSET 最小值为 0。

模式 2/B-NOR Flash

图 40-7 模式 2 和模式 B 读取访问波形

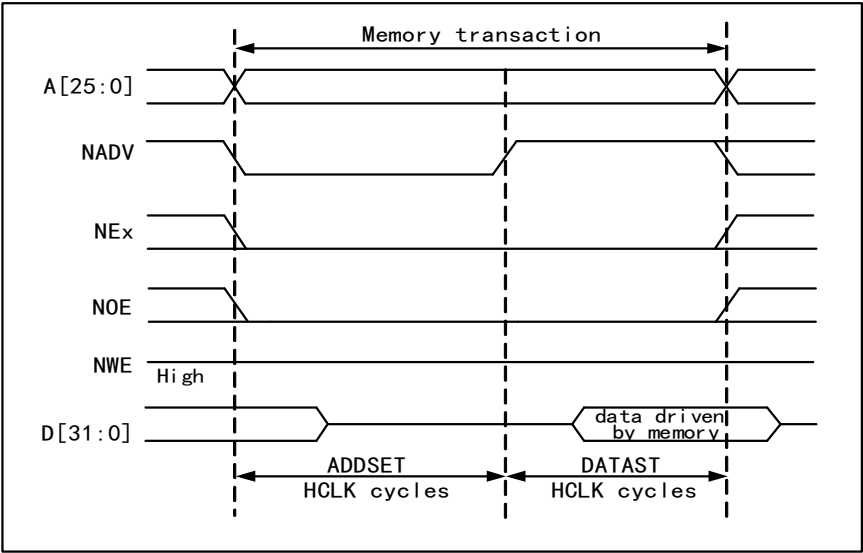


图 40-8 模式 2 写入访问波形

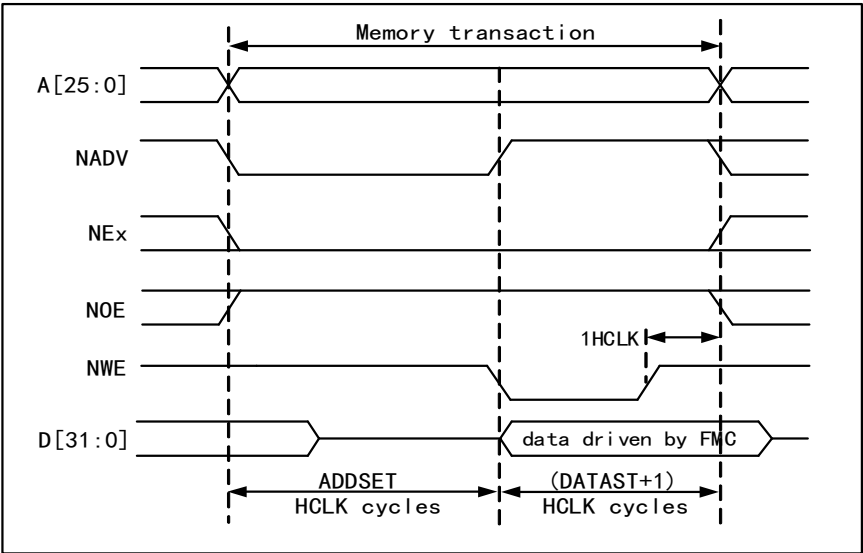
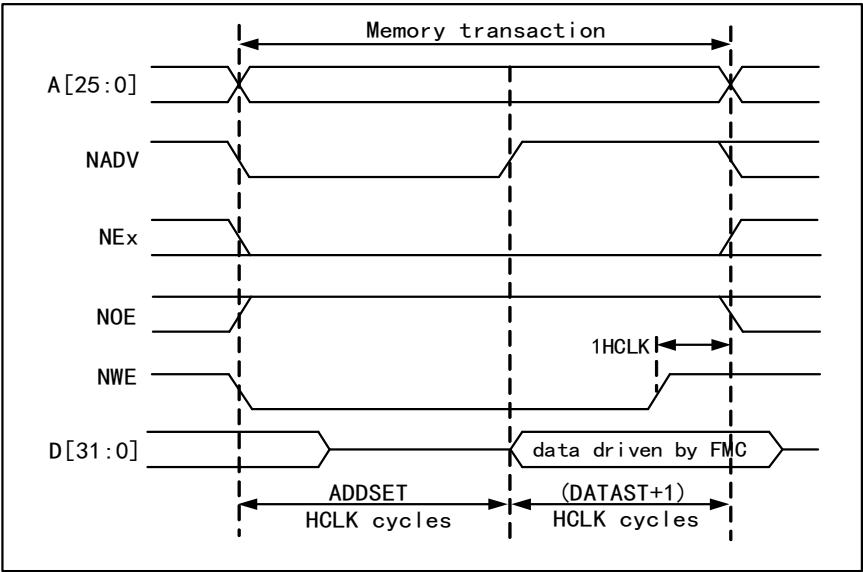


图 40-9 模式 B 写入访问波形



与模式 1 的不同之处在于设置扩展模式（模式 B）时的 NWE 切换与独立的读取和写入时序。

表 40-22 R32_FMC_BCRx 位字段

位号	位名	要设置的值
31	FMC_EN	0x1
[30:26]	保留	0x00
[25:24]	BMP	0x0 或 0x1
[23:20]	保留	0x0
19	CBURSTRW	0x0（对异步模式没有影响）
[18:16]	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0。
14	EXTMOD	模式 B 为 0x1，模式 2 为 0x0
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	0x1
[5:4]	MWID	根据需要进行设置
[3:2]	MTYP	0x2（NOR Flash）
1	MUXEN	0x0
0	MBKEN	0x1

表 40-23 R32_FMC_BTRx 位字段

位号	位名	要设置的值
[31:30]	保留	0x00
[29:28]	ACCMOD	若设置为扩展模式，则为 0x1
[27:24]	DATLAT	无关

[23:20]	CLKDIV	无关
[19:16]	BUSTURN	NE _x 变为高电平到 NE _x 变为低电平之间的时间(BUSTURN HCLK)
[15:8]	DATAST	读取访问第二个阶段的持续时间 (DATAST 个 HCLK 周期)。
[7:4]	ADDHLD	无关
[3:0]	ADDSET	读取访问第一个阶段的持续时间 (ADDSET 个 HCLK 周期)。ADDSET 最小值为 0。

表 40-24 R32_FMC_BWTRx 位字段

位号	位名	要设置的值
[31:30]	保留	0x0
[29:28]	ACCMOD	若设置为扩展模式，则为 0x1
[27:24]	DATLAT	无关
[23:20]	CLKDIV	无关
[19:16]	BUSTURN	NE _x 变为高电平到 NE _x 变为低电平之间的时间(BUSTURN HCLK)
[15:8]	DATAST	读取访问第二个阶段的持续时间 (DATAST 个 HCLK 周期)。
[7:4]	ADDHLD	无关
[3:0]	ADDSET	读取访问第一个阶段的持续时间 (ADDSET 个 HCLK 周期)。ADDSET 最小值为 0。

注：仅当设置了扩展模式（模式 B）时，R32_FMC_BWTRx 寄存器才有效，否则其内容均为“无关”。

模式 C-NOR Flash-OE 切换

图 40-10 模式 C 读取访问波形

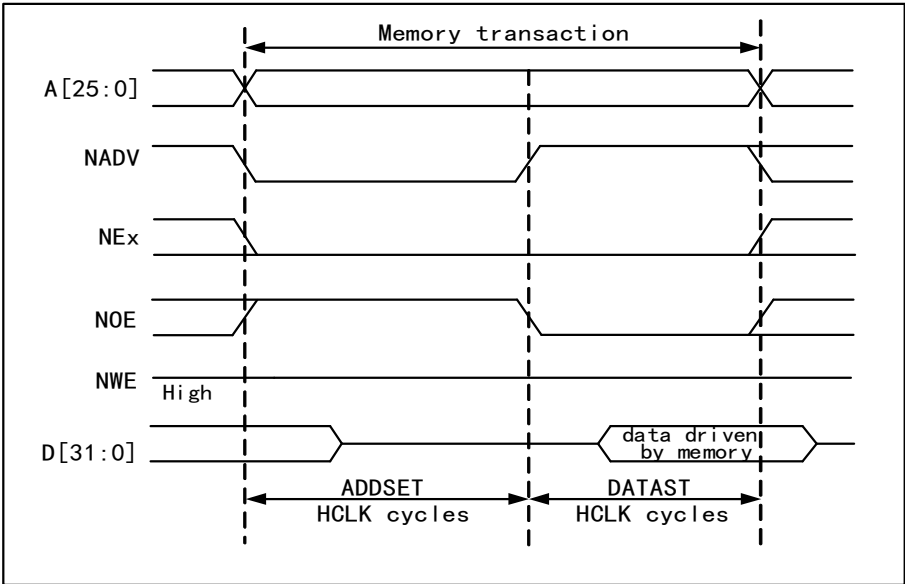
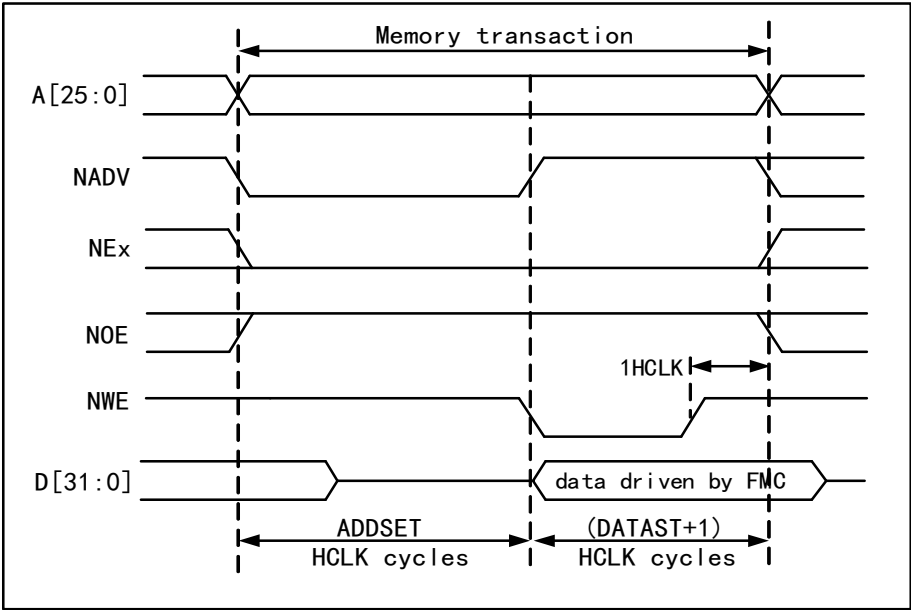


图 40-11 模式 C 写入访问波形



与模式 1 的不同之处在于 NOE 的切换与独立的读取和写入时序。

表 40-25 R32_FMC_BCRx 位字段

位号	位名	要设置的值
31	FMC_EN	0x1
[30:26]	保留	0x00
[25:24]	BMP	0x0 或 0x1
[23:20]	保留	0x0
19	CBURSTRW	0x0 (对异步模式没有影响)
[18:16]	CPSIZE	0x0 (对异步模式没有影响)
15	ASYNCWAIT	如果存储器支持该特性, 则置为 1。否则, 保持为 0。
14	EXTMOD	0x1
13	WAITEN	0x0 (对异步模式没有影响)
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	0x1
[5:4]	MWID	根据需要进行设置
[3:2]	MTYP	0x2 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

表 40-26 R32_FMC_BTRx 位字段

位号	位名	要设置的值
[31:30]	保留	0x0
[29:28]	ACCMOD	0x2

[27:24]	DATLAT	0x0
[23:20]	CLKDIV	0x0
[19:16]	BUSTURN	NE _x 变为高电平到 NE _x 变为低电平之间的时间(BUSTURN HCLK)
[15:8]	DATAST	写入访问第二个阶段的持续时间 (DATAST+1 个 HCLK 周期)
[7:4]	ADDHLD	无关
[3:0]	ADDSET	写入访问第一个阶段的持续时间 (ADDSET+1 个 HCLK 周期)。ADDSET 最小值为 0。

表 40-27 R32_FMC_BWTRx 位字段

位号	位名	要设置的值
[31:30]	保留	0x0
[29:28]	ACCMOD	0x2
[27:24]	DATLAT	无关
[23:20]	CLKDIV	无关
[19:16]	BUSTURN	NE _x 变为高电平到 NE _x 变为低电平之间的时间(BUSTURN HCLK)
[15:8]	DATAST	写入访问第二个阶段的持续时间 (DATAST 个 HCLK 周期)。
[7:4]	ADDHLD	无关
[3:0]	ADDSET	写入访问第一个阶段的持续时间 (ADDSET 个 HCLK 周期)。ADDSET 最小值为 0。

模式 D-扩展地址异步访问

图 40-12 模式 D 读取访问波形

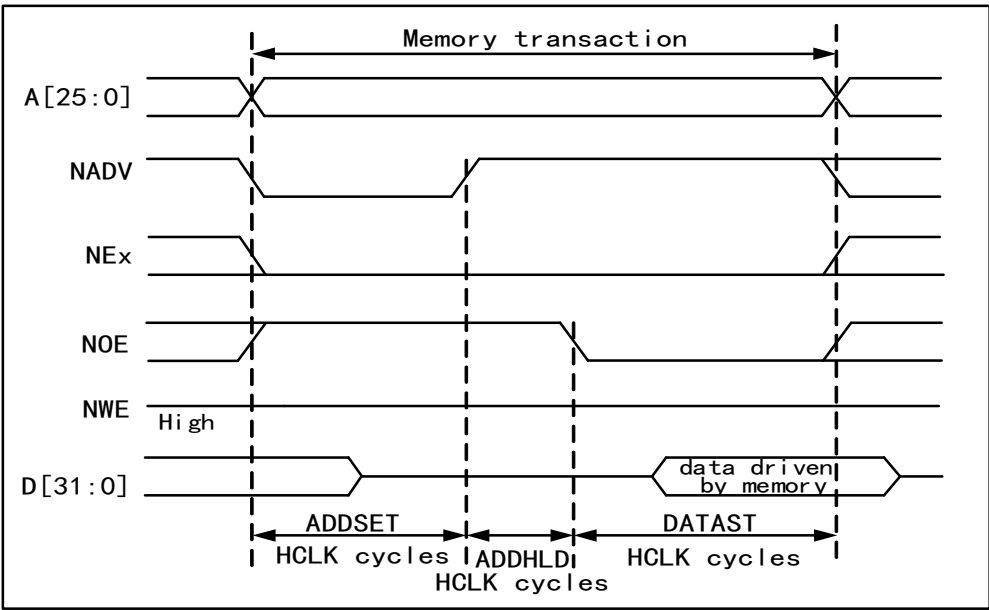
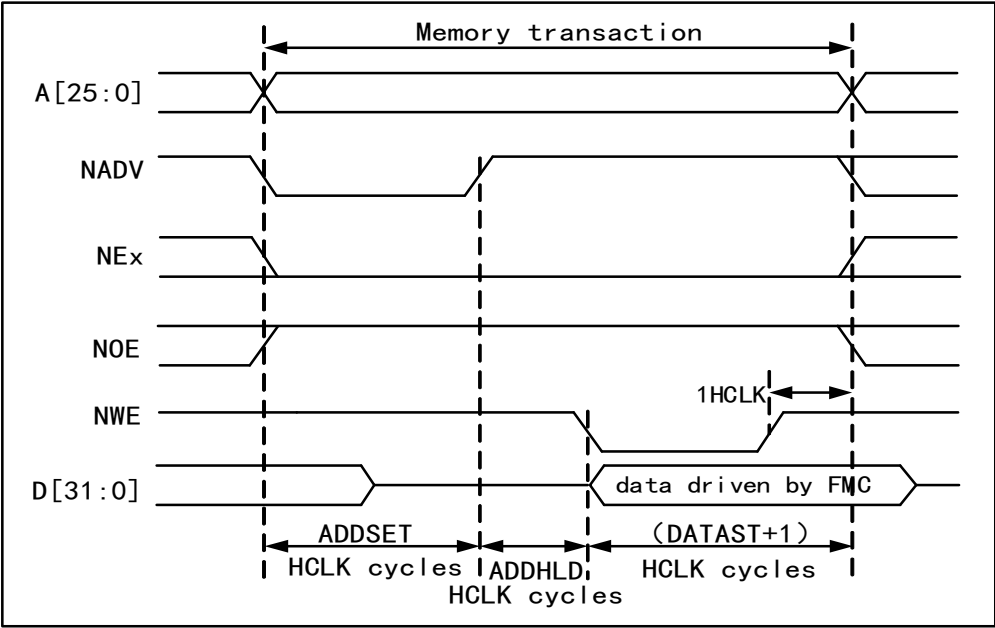


图 40-13 模式 D 写入访问波形



与模式 1 的不同之处在于 NADV 变化后 NOE 的切换与独立的读取和写入时序。

表 40-28 R32_FMC_BCRx 位字段

位号	位名	要设置的值
31	FMC_EN	0x1
[30:26]	保留	0x00
[25:24]	BMP	0x0 或 0x1
[23:20]	保留	0x0
19	CBURSTRW	0x0 (对异步模式没有影响)
[18:16]	CPSIZE	0x0 (对异步模式没有影响)
15	ASYNCWAIT	如果存储器支持该特性, 则置为 1。否则, 保持为 0。
14	EXTMOD	0x1
13	WAITEN	0x0 (对异步模式没有影响)
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	根据存储器支持情况进行设置
[5:4]	MWID	根据需要进行设置
[3:2]	MTYP	根据需要进行设置
1	MUXEN	0x0
0	MBKEN	0x1

表 40-29 R32_FMC_BTRx 位字段

位号	位名	要设置的值
[31:30]	保留	0x0

[29:28]	ACCMOD	0x3
[27:24]	DATLAT	无关
[23:20]	CLKDIV	无关
[19:16]	BUSTURN	NE _x 变为高电平到 NE _x 变为低电平之间的时间(BUSTURN HCLK)。
[15:8]	DATAST	写入访问第二个阶段的持续时间(DATAST+1 个 HCLK 周期)。
[7:4]	ADDHLD	读取访问中间阶段的持续时间(ADDHLD 个 HCLK 周期)。
[3:0]	ADDSET	写入访问第一个阶段的持续时间(ADDSET+1 个 HCLK 周期)。ADDSET 最小值为 1。

表 40-30 R32_FMC_BWTRx 位字段

位号	位名	要设置的值
[31:30]	保留	0x0
[29:28]	ACCMOD	0x3
[27:24]	DATLAT	无关
[23:20]	CLKDIV	无关
[19:16]	BUSTURN	NE _x 变为高电平到 NE _x 变为低电平之间的时间(BUSTURN HCLK)。
[15:8]	DATAST	第二个访问阶段的持续时间(写入访问为 DATAST+1 个 HCLK 周期)。
[7:4]	ADDHLD	写入访问中间阶段的持续时间(ADDHLD 个 HCLK 周期)。
[3:0]	ADDSET	写入访问第一个阶段的持续时间(ADDSET 个 HCLK 周期)。ADDSET 最小值为 0。

复用模式-复用异步访问 NOR Flash

图 40-14 复用读取访问波形

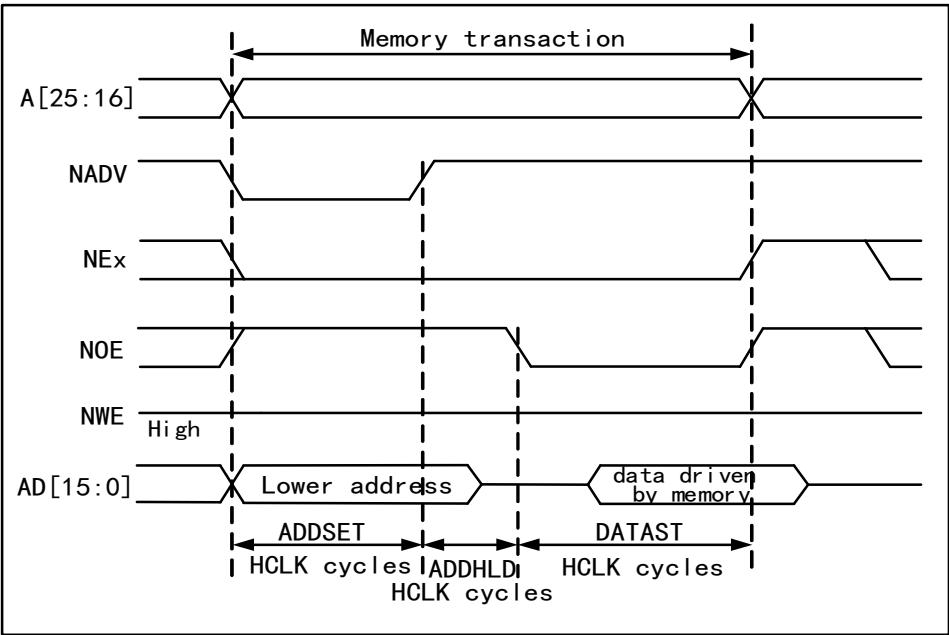
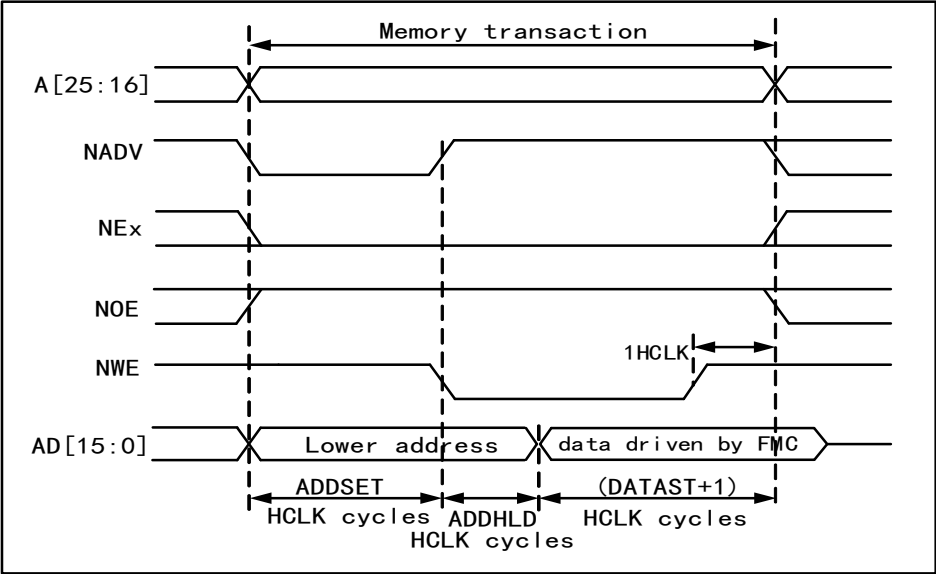


图 40-15 复用写入访问波形



与模式 D 的不同之处在于在数据总线上驱动低地址字节。

表 40-31 R32_FMC_BCRx 位字段

位号	位名	要设置的值
31	FMC_EN	0x1
[30:26]	保留	0x00
[25:24]	BMP	0x0 或 0x1
[23:20]	保留	0x0
19	CBURSTRW	0x0 (对异步模式没有影响)
[18:16]	CPSIZE	0x0 (对异步模式没有影响)
15	ASYNCWAIT	如果存储器支持该特性, 则置为 1。否则, 保持为 0。
14	EXTMOD	0x0
13	WAITEN	0x0 (对异步模式没有影响)
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	0x1
[5:4]	MWID	根据需要进行设置
[3:2]	MTYP	0x2 (NOR Flash)
1	MUXEN	0x1
0	MBKEN	0x1

表 40-32 R32_FMC_BTRx 位字段

位号	位名	要设置的值
[31:30]	保留	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	无关

[23:20]	CLKDIV	无关
[19:16]	BUSTURN	NE _x 变为高电平到 NE _x 变为低电平之间的时间(BUSTURN HCLK)。
[15:8]	DATAST	第二个访问阶段的持续时间（读取访问为 DATAST 个 HCLK 周期，写入访问为 DATAST+1 个 HCLK 周期）。
[7:4]	ADDHLD	访问中间阶段的持续时间（ADDHLD 个 HCLK 周期）。
[3:0]	ADDSET	第一个访问阶段的持续时间（ADDSET 个 HCLK 周期）。ADDSET 的最小值为 1。

异步访问中的 WAIT 管理

如果异步存储器发出 WAIT 信号，指示尚未准备好接受或提供数据，则 R32_FMC_BCR_x 寄存器中的 ASYNCWAIT 位必须置 1。

如果 WAIT 信号处于有效状态（电平高低取决于 WAITPOL 位），则由 DATAST 位控制的第二个访问阶段（数据建立阶段）将延长，直到 WAIT 变为无效状态。与数据建立阶段不同，由 ADDSET 和 ADDHLD 位控制的第一个访问阶段（地址建立和地址保持阶段）对 WAIT 不敏感，因此第一个访问阶段不会延长。

必须配置数据建立阶段，以便在存储器事务结束前 4 个 HCLK 周期检测到 WAIT。必须考虑以下情况：

- 1) 存储器发出的 WAIT 信号和 NOE/NEW 信号对齐：

$$DATAST \geq (4 * HCLK) + \max\_wait\_assertion\_time$$

- 2) 存储器发出的 WAIT 信号和 NE_x 对齐（或者 NOE/NWE 信号不翻转）：

如果

$$\max\_wait\_assertion\_time > address\_phase + hold\_phase$$

那么：

$$DATAST \geq (4 * HCLK) + (\max\_wait\_assertion\_time - address\_phase - hold\_phase)$$

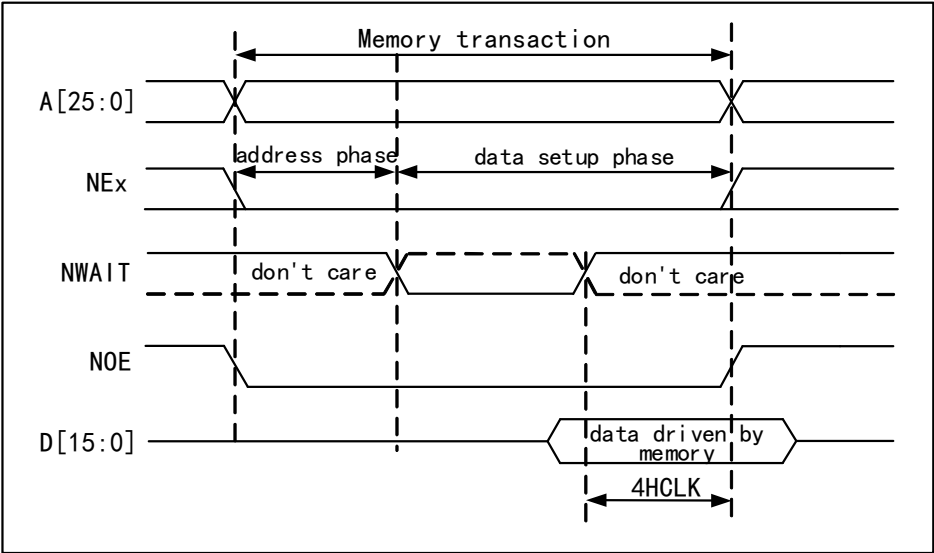
否则

$$DATAST \geq 4 * HCLK$$

其中，max_wait_assertion_time 是在 NE_x/NOE/NWE 变为低电平后存储器使能 WAIT 信号所花费的最长时间。

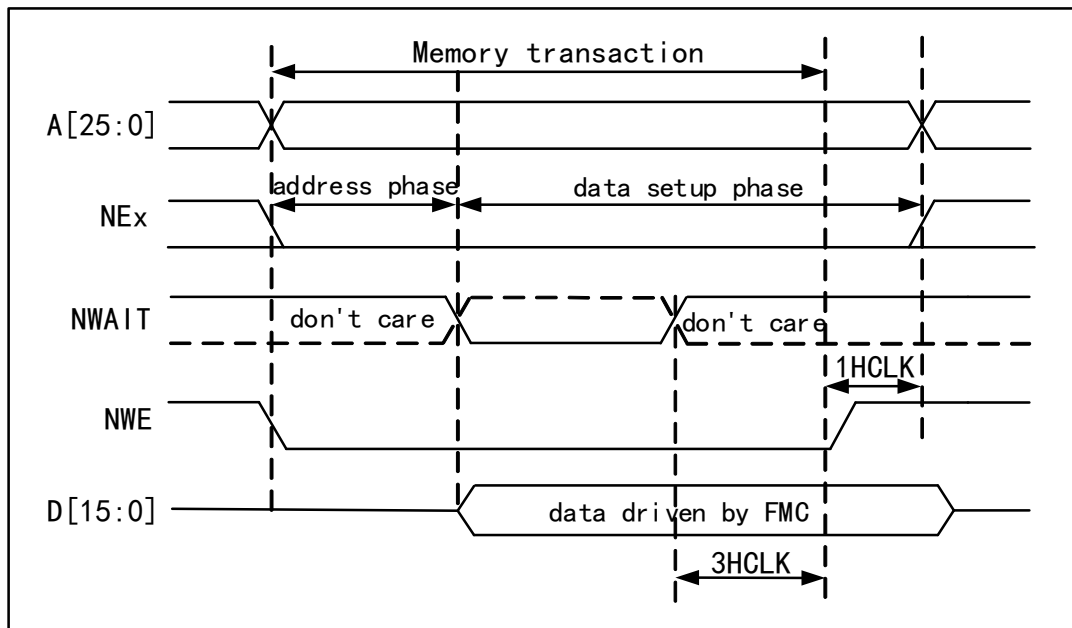
图 40-15 和图 40-16 显示了异步存储器释放 WAIT 之后，在存储器访问阶段增加的 HCLK 时钟周期的个数（与上述情况无关）。

图 40-16 读取访问波形期间的异步等待



注：NWAIT 极性取决于 R32_FMC_BCRx 寄存器中的 WAITPOL 位设置。

图 40-17 写入访问波形期间的异步等待



注：NWAIT 极性取决于 R32_FMC_BCRx 寄存器中的 WAITPOL 位设置。

40.5.5 同步事务

存储器时钟 FMC_CLK 是 HCLK 的约数。它取决于 CLKDIV 的值和 MWID/HB 数据大小，如以下公式所示：

$$\text{FMC\_CLK 分频比} = \max(\text{CLKDIV}+1, \text{MWID (HB 数据大小)})$$

MWID 为 16 位或 8 位时，FMC_CLK 分频比始终由 CLKDIV 的设定值定义。

MWID 为 32 位时，FMC_CLK 分频比还取决于 HB 数据大小。

示例：

- CLKDIV=1, MWID=32 位, HB 数据大小=8 位时, FMC_CLK=HCLK/4。
- CLKDIV=1, MWID=16 位, HB 数据大小=8 位时, FMC_CLK=HCLK/2。

NOR Flash 指定了从 NADV 使能到 CLK 高电平的最短时间。为了符合这一限制，FMC 不会在同步访问的第一个内部时钟周期内（NADV 使能之前）将时钟发到存储器。这样可以确保存储器时钟的上升沿出现在 NADV 低脉冲的中间。

数据延迟与 NOR 延迟

数据延迟是对数据进行采样之前需要等待的周期数。DATLAT 的值必须与 NOR FLASH 配置寄存器中指定的延迟值一致。当数据延迟计数中 NADV 为低电平时，FMC 不会计入时钟周期。

一些 NOR Flash 将 NADV 低电平周期计入数据延迟计数，这样 NOR Flash 延迟和 FMC DATLAT 参数之间的确切关系可以是以下任一种：

- NOR Flash 延迟= (DATLAT+2) 个 CLK 时钟周期
- NOR Flash 延迟= (DATLAT+3) 个 CLK 时钟周期

近来有一些存储器会在延迟阶段使能 NWAIT。在这种情况下，可以将 DATLAT 设置为最小值。然后，FMC 会对数据进行采样，并且等待足够长的时间来评估数据是否有效。这样，FMC 就能检测到存储器存在延迟的时间，从而处理真实数据。

其他存储器不会在延迟期间使能 NWAIT。在这种情况下，必须正确设置 FMC 和存储器的延迟，否则可能会将无效数据误用为有效数据，或者在存储器访问初始阶段丢失有效数据。

单次突发传输

当所选存储区域配置为同步突发模式时，例如，如果向 16 位存储器请求了一个 HB 单次突发事务，则 FMC 会执行长度为 1 的突发事务（如果 HB 传输为 16 位）或者长度为 2 的突发事务（如果 HB 传输为 32 位），然后在最后一个数据选通时禁止片选信号。

与异步读取操作相比，就周期而言这并不是最有效的传输方法。但是，随机异步读取需要先重新编程存储器访问模式，这样总时间会更长。

越过 Cellular RAM1.5 的边界页

Cellular RAM1.5 不允许突发访问越过页边界。根据存储器页大小配置 R32_FMC_BCR1 寄存器中的 CPSIZE 位来达到存储器页大小时，FMC 控制器会自动分离突发访问。

等待管理

对于同步 NOR Flash，会在配置的延迟周期（相当于 $(DATLAT+2)$ 个 CLK 时钟周期）之后对 NWAIT 进行评估。

如果 NWAIT 有效（WAITPOL=0 时为低电平，WAITPOL=1 时为高电平），会插入等待状态，直到 NWAIT 无效（WAITPOL=0 时为高电平，WAITPOL=1 时为低电平）。

当 NWAIT 无效时，数据将立即（位 WAITCFG=1）或在下一个时钟边沿（位 WAITCFG=0）被视为有效。

通过 NWAIT 信号插入等待周期期间，控制器会继续将时钟脉冲发送到存储器，保持片选和输出使能信号有效。但不将数据视为有效。

突发模式下，NOR Flash NWAIT 信号有两种时序配置：

- Flash 在等待周期之前一个数据周期发出 NWAIT 信号（复位后的默认值）。
- Flash 在等待周期期间发出 NWAIT 信号。

FMC 支持这两种 NOR Flash 等待周期配置，通过 R32_FMC_BCRx 寄存器的 WAITCFG 位（x=0..3）针对每个片选进行配置。

图 40-18 等待配置波形

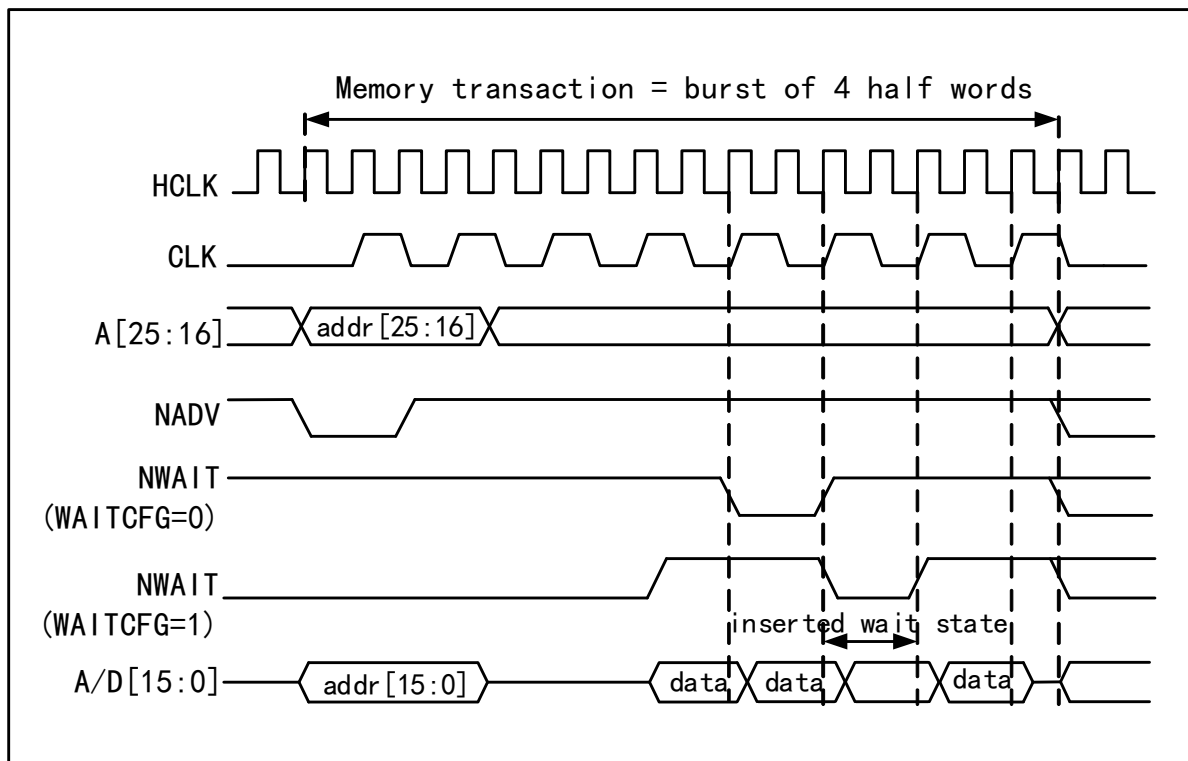
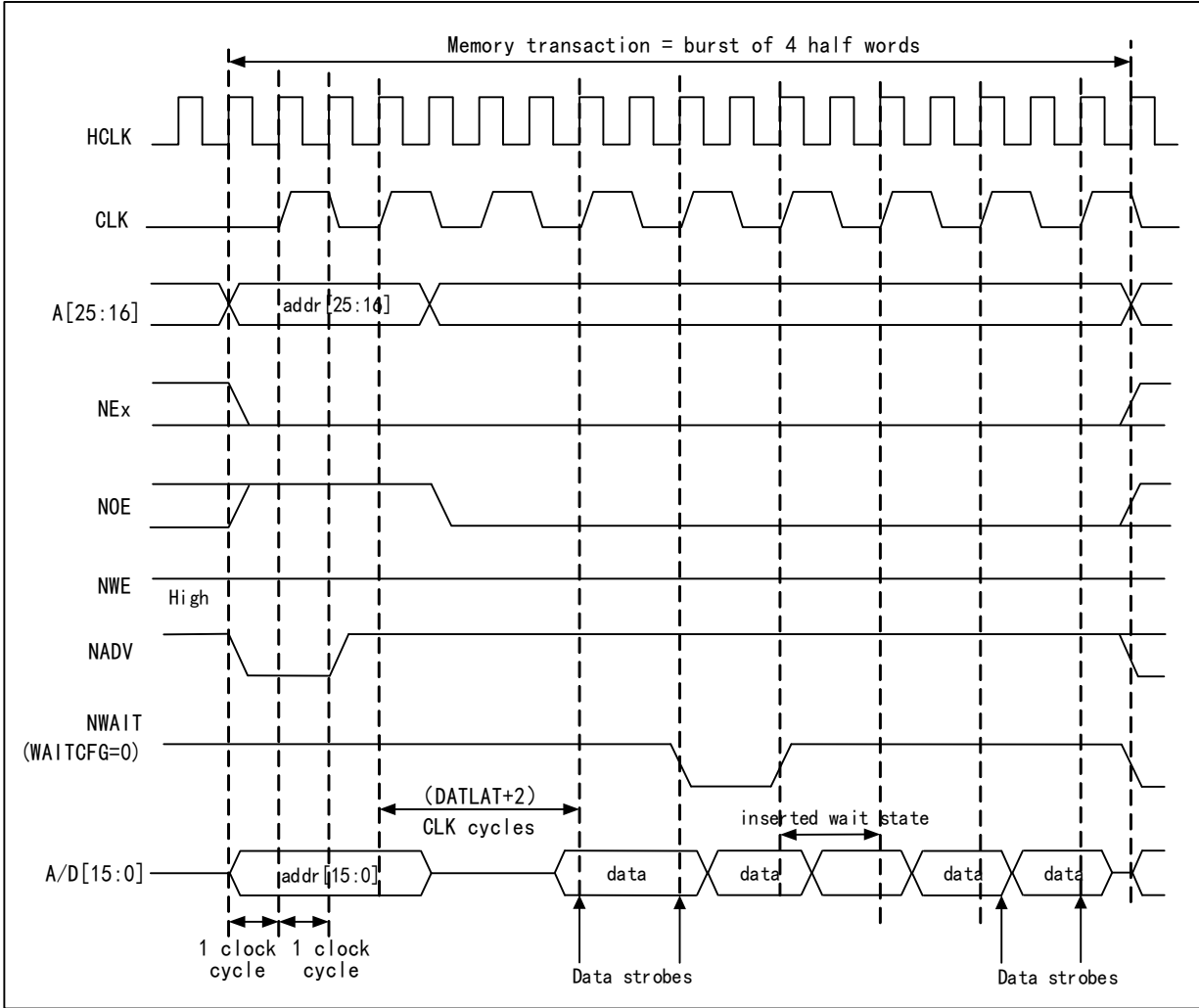


图 40-19 同步复用读取模式波形-NOR、PSRAM (CRAM)



注：字节通道输出 NBL 未显示，它们对于 NOR 访问保持高电平，对于 PSRAM (CRAM) 访问则保持低电平。

表 40-33 R32_FMC_BCRx 位字段

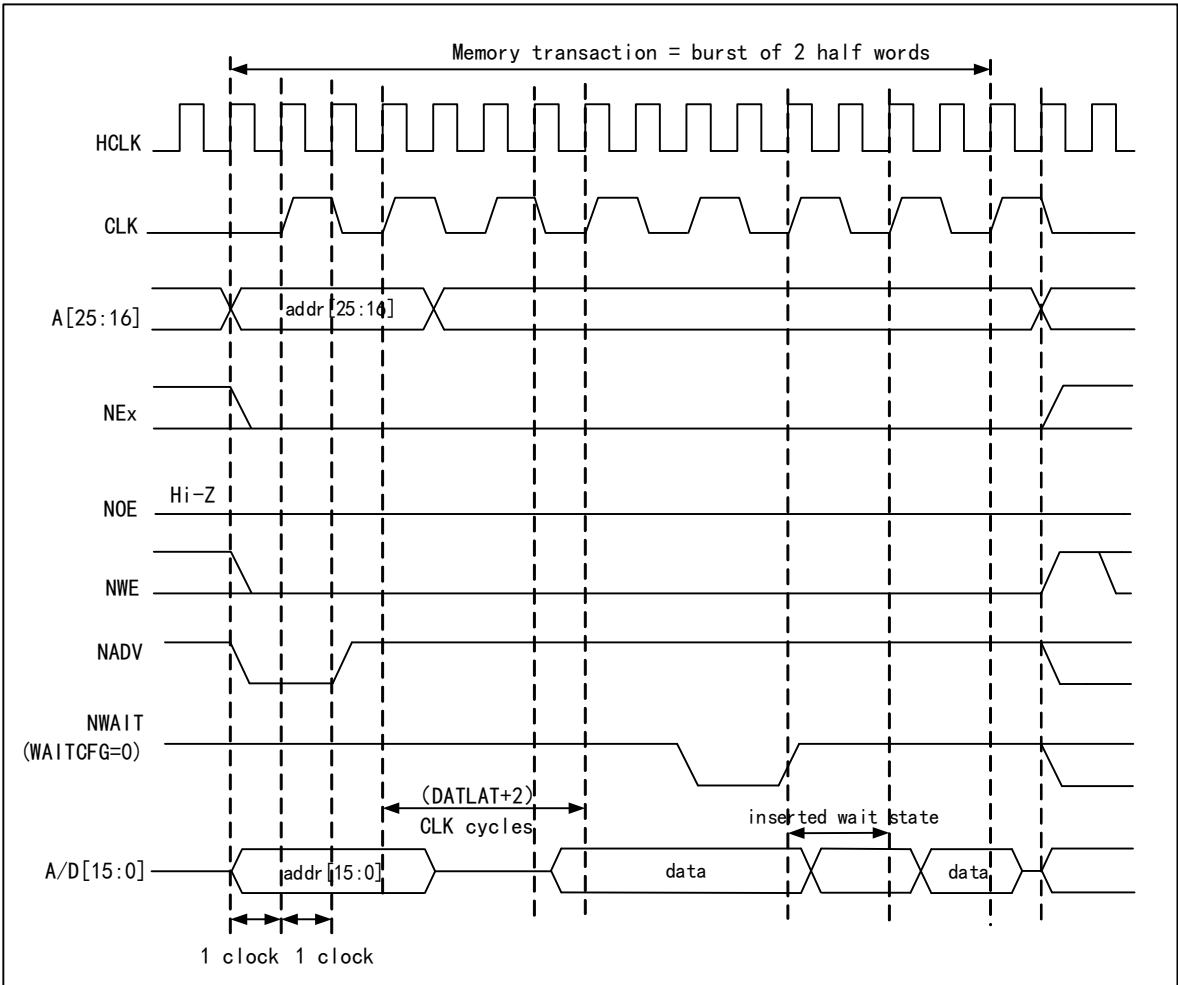
位号	位名	要设置的值
31	FMC_EN	0x1
[30:26]	保留	0x00
[25:24]	BMP	0x0 或 0x1
[23:20]	保留	0x0
19	CBURSTRW	对同步读取没有影响
[18:16]	CPSIZE	0x0 (对异步模式没有影响)
15	ASYNCAWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	在存储器支持该特性的情况下置位为 1，否则保持为 0
12	WREN	对同步读取没有影响
11	WAITCFG	是否置位视存储器情况而定
10	保留	0x0
9	WAITPOL	是否置位视存储器情况而定
8	BURSTEN	0x1
7	保留	0x1

6	FACCEN	在存储器支持的情况下置位（NOR Flash）
[5:4]	MWID	根据需要进行设置
[3:2]	MTYP	0x1 或 0x2
1	MUXEN	根据需要进行设置
0	MBKEN	0x1

表 40-34 R32_FMC_BTRx 位字段

位号	位名	要设置的值
[31:30]	保留	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	数据延迟
[23:20]	CLKDIV	0x0, 使 CLK=HCLK（不支持） 0x1, 使 CLK=2*HCLK ...
[19:16]	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间(BUSTURN HCLK)
[15:8]	DATAST	无关
[7:4]	ADDHLD	无关
[3:0]	ADDSET	无关

图 40-20 同步复用写入模式波形-PSRAM（GRAM）



注 1: 存储器必须提前一个周期发出 *NWAIT* 信号, 相应地 *WAITCFG* 必须编程为 0。

注 2: 字节通道 (NBL) 输出未显示, 当 *NE_x* 有效时它们保持低电平。

表 40-35 R32_FMC_BCR_x 位字段

位号	位名	要设置的值
31	FMC_EN	0x1
[30:26]	保留	0x00
[25:24]	BMP	0x0 或 0x1
[23:20]	保留	0x0
19	CBURSTRW	0x1
[18:16]	CPSIZE	根据需要进行设置 (CRAM1.5 为 0x1)
15	ASYNCAWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	在存储器支持该特性的情况下置位为 1, 否则保持为 0
12	WREN	0x1
11	WAITCFG	0x0
10	保留	0x0
9	WAITPOL	是否置位视存储器情况而定
8	BURSTEN	对同步写入没有影响
7	保留	0x1
6	FACCEN	根据存储器支持情况进行设置
[5:4]	MWID	根据需要进行设置
[3:2]	MTYP	0x1
1	MUXEN	根据需要进行设置
0	MBKEN	0x1

表 40-36 R32_FMC_BTR_x 位字段

位号	位名	要设置的值
[31:30]	保留	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	数据延迟
[23:20]	CLKDIV	0x0, 使 CLK=HCLK (不支持) 0x1, 使 CLK=2*HCLK
[19:16]	BUSTURN	NE _x 变为高电平到 NE _x 变为低电平之间的时间 (BUSTURN HCLK)
[15:8]	DATAST	无关
[7:4]	ADDHLD	无关
[3:0]	ADDSET	无关

40.5.6 NOR/PSRAM 控制寄存器

表 40-37 NOR/PSRAM 相关寄存器列表

名称	访问地址	描述	复位值
R32_FMC_BCR1	0x40025400	SRAM/NOR-Flash 片选控制寄存器 1	0x00003052
R32_FMC_BTR1	0x40025404	SRAM/NOR-Flash 片选时序寄存器 1	0x0FFFFFFF
R32_FMC_BCR2	0x40025408	SRAM/NOR-Flash 片选控制寄存器 2	0x000030D2

R32_FMC_BTR2	0x4002540C	SRAM/NOR-Flash 片选时序寄存器 2	0x0FFFFFFF
R32_FMC_BCR3	0x40025410	SRAM/NOR-Flash 片选控制寄存器 3	0x000030D2
R32_FMC_BTR3	0x40025414	SRAM/NOR-Flash 片选时序寄存器 3	0x0FFFFFFF
R32_FMC_BCR4	0x40025418	SRAM/NOR-Flash 片选控制寄存器 4	0x000030D2
R32_FMC_BTR4	0x4002541C	SRAM/NOR-Flash 片选时序寄存器 4	0x0FFFFFFF
R32_FMC_BWTR1	0x40025504	SRAM/NOR-Flash 写入时序寄存器 1	0x0FFFFFFF
R32_FMC_BWTR2	0x4002550C	SRAM/NOR-Flash 写入时序寄存器 2	0x0FFFFFFF
R32_FMC_BWTR3	0x40025514	SRAM/NOR-Flash 写入时序寄存器 3	0x0FFFFFFF
R32_FMC_BWTR4	0x4002551C	SRAM/NOR-Flash 写入时序寄存器 4	0x0FFFFFFF

40.5.6.1 SRAM/NOR-Flash 片选控制寄存器 1 (R32_FMC_BCR1)

偏移地址: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FMC_EN	Reserved					BMP	Reserved					CBURSTRW	CPSIZE[2:0]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCAWAIT	EXTMOD	WAITEN	WR	WAITCFG	Reserved	WAITPOL	BURSTEN	Reserved	FACEN	MWID[1:0]		MTYP[1:0]		MUXEN	MBKEN

位	名称	访问	描述	复位值
31	FMC_EN	RW	NAND/NOR/PSRAM 使能位: 1: 开启; 0: 关闭,	0
[30:26]	Reserved	RO	保留。	0
[25:24]	BMP	RW	地址重映射, 实现表 40-1 地址切换功能。	0
[23:20]	Reserved	RO	保留。	0
19	CBURSTRW	RW	同步写使能: 1: 在同步模式下写入; 0: 始终在异步模式下写入。	0
[18:16]	CPSIZE[2:0]	RW	GRAM 页大小: 000: 越过页边界后不进行突发分离; 001: 128 字节; 010: 256 字节; 011: 512 字节; 100: 1024 字节; 其它: 保留。	0
15	ASYNCAWAIT	RW	异步传输期间的等待信号: 该位可使能/禁止 FMC 使用等待信号, 即使在异步协议期间也有效。 1: 运行异步协议时考虑 NWAIT 信号; 0: 运行异步协议时不考虑 NWAIT 信号。	0
14	EXTMOD	RW	扩展模式使能: FMC 可对 R32_FMC_BWTR 寄存器中非复用异步访问的写入时序进行配置, 此配置由 EXTMOD 位使能,	0

			进而使读取和写入操作采用不同时序。 1: 考虑 R32_FMC_BWTR 寄存器中的值; 0: 不考虑 R32_FMC_BWTR 寄存器中的值。 注: 如果禁用扩展模式, FMC 可以在模式 1 或模式 2 下运行, 如下所述: - 当选择 SRAM/PSRAM 存储器类型时, 模式 1 为默认模式 (MTYP=0x0 或 0x01); - 当选择 NOR 存储器类型时, 模式 2 为默认模式 (MTYP=0x10)。	
13	WAITEN	RW	等待使能位: 1: 使能 NWAIT 信号 (考虑其电平, 如果使能, 在配置过的延迟周期后插入等待周期); 0: 禁止 NWAIT 信号 (不考虑其电平, 不在配置过的 Flash 延迟周期后插入等待周期)。	1
12	WREN	RW	写入使能位: 1: FMC 使能在存储区域内写入; 0: FMC 禁止在存储区域内写入, 如果进行写操作将报告 HB 错误。	1
11	WAITCFG	RW	等待时序配置: 1: NWAIT 信号在等待周期期间有效 (不适用于 PSRAM); 0: NWAIT 信号在等待周期之前的一个数据周期有效。	0
10	Reserved	RO	保留。	0
9	WAITPOL	RW	等待信号极性位: 1: NWAIT 高电平有效; 0: NWAIT 低电平有效。	0
8	BURSTEN	RW	同步读使能位: 1: 使能同步读模式。在同步模式下进行读取访问; 0: 禁止同步读模式。在异步模式下进行读取访问。	0
7	Reserved	RO	保留。	0
6	FACCEN	RW	Flash 访问使能: 1: 使能相应的 NOR Flash 访问; 0: 禁止相应的 NOR Flash 访问。	1
[5:4]	MWID[1:0]	RW	存储器数据总线宽度: 00: 8 位; 01: 16 位; 10: 32 位; 11: 保留。	01b
[3:2]	MTYP[1:0]	RW	存储器类型: 00: SRAM (对于存储区域 2~4, 复位后的默认值); 01: PSRAM (CRAM); 10: NOR Flash/OneNAND Flash (对于存储区域 1, 复位后的默认值); 11: 保留。	00b
1	MUXEN	RW	地址/数据复用使能位:	1

			该位置 1 时，地址和数据值在数据总线上复用，仅对 NOR 和 PSRAM 存储器有效。 1：地址/数据在数据总线上复用； 0：地址/数据非复用。	
0	MBKEN	RW	存储区域使能位： 使能存储区域。复位后使能存储区域 1，其它存储区域均禁止。访问禁止的存储区域会引起 HB 总线上的错误。 1：使能相应的存储区域； 0：禁止相应的存储区域。	0

40.5.6.2 SRAM/NOR-Flash 片选控制寄存器 x (R32_FMC_BCRx) (x=2/3/4)

偏移地址：0x8* (x - 1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CBURSTRW	CPSIZE[2:0]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWAIT	EXTMOD	WAITEN	WR EN	WAITCFG	Reserved	WAITPOL	BURSTEN	Reserved	MWID[1:0]		MTYP[1:0]		MUXEN	MBKEN	

位	名称	访问	描述	复位值
31	Reserved	RW	保留。	0
[30:26]	Reserved	RO	保留。	0
[25:24]	Reserved	RW	保留。	0
[23:20]	Reserved	RO	保留。	0
19	CBURSTRW	RW	同步写使能： 1：在同步模式下写入； 0：始终在异步模式下写入。	0
[18:16]	CPSIZE[2:0]	RW	GRAM 页大小： 这些位用于 Cellular RAM1.5，Cellular RAM1.5 不允许突发访问越过页面之间的地址边界。如果配置这些位，FMC 控制器会在达到存储器页大小后自动分离突发访问。 000：越过页边界后不进行突发分离； 001：128 字节； 010：256 字节； 011：512 字节； 100：1024 字节； 其它：保留。	0
15	ASYNCWAIT	RW	异步传输期间的等待信号： 该位可使能/禁止 FMC 使用等待信号，即使在异步协议期间也有效。 1：运行异步协议时考虑 NWAIT 信号； 0：运行异步协议时不考虑 NWAIT 信号。	0

14	EXTMOD	RW	扩展模式使能： FMC 可对 R32_FMC_BWTR 寄存器中非复用异步访问的写入时序进行配置，此配置由 EXTMOD 位使能，进而使读取和写入操作采用不同时序。 1：考虑 R32_FMC_BWTR 寄存器中的值； 0：不考虑 R32_FMC_BWTR 寄存器中的值。 注：如果禁用扩展模式，FMC 可以在模式 1 或模式 2 下运行，如下所述： - 当选择 SRAM/PSRAM 存储器类型时，模式 1 为默认模式（MTYP=0x0 或 0x01）； - 当选择 NOR 存储器类型时，模式 2 为默认模式（MTYP=0x10）。	0
13	WAITEN	RW	等待使能位： 1：使能 NWAIT 信号（考虑其电平，如果使能，在配置过的延迟周期后插入等待周期）； 0：禁止 NWAIT 信号（不考虑其电平，不在配置过的 Flash 延迟周期后插入等待周期）。	1
12	WREN	RW	写入使能位： 1：FMC 使能在存储区域内写入； 0：FMC 禁止在存储区域内写入，如果进行写操作将报告 HB 错误。	1
11	WAITCFG	RW	等待时序配置： 1：NWAIT 信号在等待周期期间有效（不适用于 PSRAM）； 0：NWAIT 信号在等待周期之前的一个数据周期有效。	0
10	Reserved	RO	保留。	0
9	WAITPOL	RW	等待信号极性位： 1：NWAIT 高电平有效； 0：NWAIT 低电平有效。	0
8	BURSTEN	RW	同步读使能位： 1：使能同步读模式。在同步模式下进行读取访问； 0：禁止同步读模式。在异步模式下进行读取访问。	0
7	Reserved	RO	保留。	1
6	Reserved	RW	保留。	1
[5:4]	MWID[1:0]	RW	存储器数据总线宽度： 00：8 位； 01：16 位； 10：32 位； 11：保留。	01b
[3:2]	MTYP[1:0]	RW	存储器类型： 00：SRAM（对于存储区域 2...4，复位后的默认值）； 01：PSRAM（CRAM）； 10：NOR Flash/OneNAND Flash（对于存储区域 1，复位后的默认值）； 11：保留。	0

1	MUXEN	RW	地址/数据复用使能位： 该位置 1 时，地址和数据值在数据总线上复用， 仅对 NOR 和 PSRAM 存储器有效。 1：地址/数据在数据总线上复用； 0：地址/数据非复用。	1
0	MBKEN	RW	存储区域使能位： 使能存储区域。复位后使能存储区域 1，其它存储 区域均禁止。访问禁止的存储区域会引起 HB 总线 上的错误。 1：使能相应的存储区域； 0：禁止相应的存储区域。	0

40.5.6.3 SRAM/NOR-Flash 片选时序寄存器 x (R32_FMC_BTRx) (x=1/2/3/4)

偏移地址：0x04 + 0x8* (x - 1)

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 [1:0]	RW	访问模式： 指定异步访问模式，如时序图所示。仅当 R32_FMC_BCRx 寄存器中的 EXTMOD 位置 1 时，这些 位才有效。 00：访问模式 A； 01：访问模式 B； 10：访问模式 C； 11：访问模式 D。	0
[27:24]	DATLAT [3:0]	RW	同步存储器的数据延迟： 对于使能了读/写突发模式（BURSTEN/CBURSTRW 位置 1）的同步访问，该字段定义读写首个数据前 要发送给存储器的存储器时钟周期数（+2）： 该时序参数以 FMC_CLK 周期而非 HCLK 周期表示。 该值与异步访问模式无关。 0000：首次突发访问时，2 个 CLK 时钟周期的数据 延迟； 1111：首次突发访问时，17 个 CLK 时钟周期的数 据延迟。	1111b
[23:20]	CLKDIV [3:0]	RW	FMC_CLK 信号的时钟分频比： 0000：保留； 0001：FMC_CLK 周期=2*HCLK 周期； 0010：FMC_CLK 周期=3*HCLK 周期；	1111b

			1111: FMC_CLK 周期=16*HCLK 周期。 在异步 NOR Flash、SRAM 或 PSRAM 访问模式下, 该值为无关值。 注: 有关 FMC_CLK 分频比公式的信息, 请参考第 40.5.5 节: 同步事务。	
[19:16]	BUSTURN[3:0]	RW	总线周转阶段的持续时间: 通过软件写入这些位可在写-读(和读-写)事务结束时添加延迟。 总线周转延迟还会插入对不同存储区域进行的两个连续的读/写事务之间。当两个连续的读事务指向同一个 FMC 存储区域时, 无论所配置的 BUSTURN 时序为何, 均只会插入 1 个 HCLK 时钟周期。当两个连续的写事务指向同一个 FMC 存储区域时, 不会插入总线周转延迟。 总线周转延迟可以匹配连续事务之间的最短时间 (t_{EHEL} 由 NE_x 高电平变为 NE_x 低电平) 以及存储器在读取访问后释放数据总线所需的最长时间 (t_{EHQZ}): (BUSTURN+1) HCLK 周期 $\geq t_{EHELmin}$ (BUSTURN+2) HCLK 周期 $\geq t_{EHQZmax}$ (若 EXTMOD=0) (BUSTURN+2) HCLK 周期 $\geq \max(t_{EHELmin}, t_{EHQZmax})$ (若 EXTMOD=1)。 0000: BUSTURN 阶段的持续时间=增加 0 个 HCLK 时钟周期; ... 1111: BUSTURN 阶段的持续时间=15*HCLK 时钟周期。	1111b
[15:8]	DATAST[7:0]	RW	数据阶段的持续时间: 通过软件写入这些位可定义数据阶段的持续时间 (请参考图 40-3~图 40-15), 适用于异步访问模式: 0000 0000: 保留; 0000 0001: DATAST 阶段的持续时间=1*HCLK 时钟周期; 0000 0010: DATAST 阶段的持续时间=2*HCLK 时钟周期; ... 1111 1111: DATAST 阶段的持续时间=255*HCLK 时钟周期。 有关每种存储器类型和访问模式数据阶段持续时间的信息 (请参考图 40-3~图 40-15)。 例如: 在模式 1、写入访问以及 DATAST=1 条件下, 数据阶段的持续时间=DATAST+1=2 个 HCLK 时钟周期。 注: 在同步访问模式下, 该值为无关值。	1111 1111b
[7:4]	ADDHLD[3:0]	RW	地址保持阶段的持续时间:	1111b

			通过软件写入这些位可定义地址保持阶段的持续时间（请参考图 40-3～图 40-15），适用于模式 D 或复用访问： 0000：保留； 0001：ADDHLD 阶段的持续时间=1*HCLK 时钟周期； 0010：ADDHLD 阶段的持续时间=2*HCLK 时钟周期； ... 1111：ADDHLD 阶段的持续时间=15*HCLK 时钟周期。 有关每种访问模式地址保持阶段持续时间的信息，请参见相应图片（请参考图 40-3～图 40-15）。 注：在同步访问模式下，该值不使用，因为地址保持阶段的持续时间始终是 1 个存储器时钟周期。	
[3:0]	ADDSET[3:0]	RW	地址设置阶段的持续时间： 通过软件写入这些位可定义地址设置阶段的持续时间（请参考图 40-3～图 40-15），适用于 SRAM、ROM、异步 NOR Flash 和 PSRAM： 0000：ADDSET 阶段的持续时间=0*HCLK 时钟周期； ... 1111：ADDSET 阶段的持续时间=15*HCLK 时钟周期； 有关每种访问模式地址设置阶段持续时间的信息，请参见相应图片（请参考图 40-3～图 40-15）。 注：在同步访问模式下，该值为无关值。 复用模式或模式 D 下，ADDSET 的最小值为 1。	1111b

注：PSRAM (CRAM) 由于内部刷新而导致数据延时时间长度不确定。因此，这些存储器会在整个延迟阶段发送 NWAIT 信号，以便按照需要延长延迟。对于 PSRAM (CRAM)，字段 DATLAT 必须设置为 0，这样 FMC 会立即退出延迟阶段，开始对存储器中的 NWAIT 采样，然后在存储器准备就绪后开始读取或写入。此方法也适用于最新一代的同步 Flash，同早期 Flash 不同的是：此类 Flash 会发送 NWAIT 信号（检查所用的具体 Flash 数据表）。

40.5.6.4 SRAM/NOR-Flash 写入时序寄存器 x (R32_FMC_BWTRx) (x=1/2/3/4)

偏移地址：0x104 + 0x8* (x - 1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		ACCMOD [1:0]		Reserved								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	R0	保留。	0
[29:28]	ACCMOD[1:0]	RW	访问模式： 指定异步访问模式，如下一个时序图所示。仅当 R32_FMC_BCRx 寄存器中的 EXTMOD 位置 1 时，这些位才有效。 00：访问模式 A；	0

			01: 访问模式 B; 10: 访问模式 C; 11: 访问模式 D。	
[27:20]	Reserved	R0	保留。	0
[19:16]	BUSTURN[3:0]	RW	总线周转阶段的持续时间: 通过软件写入这些位可在写-读事务结束时增加一个延迟, 进而匹配两个连续事务之间的最小时间 (t_{EHEL} 由 ENx 高电平变为 ENx 低电平): (BUSTURN+1) HCLK 周期 $\geq t_{\text{EHELmin}}$ 0000: BUSTURN 阶段的持续时间=增加 0 个 HCLK 时钟周期; ... 1111: BUSTURN 阶段的持续时间=增加 15 个 HCLK 时钟周期。	1111b
[15:8]	DATAST[7:0]	RW	数据阶段的持续时间: 通过软件写入这些位可定义数据阶段的持续时间 (请参考图 40-3~图 40-15), 适用于异步 SRAM、PSRAM 和 NOR Flash 访问模式: 0000 0000: DATAST 阶段的持续时间=1*HCLK 时钟周期; 0000 0001: DATAST 阶段的持续时间=2*HCLK 时钟周期; 0000 0010: DATAST 阶段的持续时间=3*HCLK 时钟周期; ... 1111 1111: DATAST 阶段的持续时间=256*HCLK 时钟周期。	1111 1111b
[7:4]	ADDHLD[3:0]	RW	地址保持阶段的持续时间: 通过软件写入这些位可定义地址保持阶段的持续时间 (请参考图 40-3~图 40-15), 适用于异步复用访问: 0000: 保留; 0001: ADDHLD 阶段的持续时间=1*HCLK 时钟周期; 0010: ADDHLD 阶段的持续时间=2*HCLK 时钟周期; ... 1111: ADDHLD 阶段的持续时间=15*HCLK 时钟周期。 注: 在同步 NOR Flash 访问模式下, 该值不使用, 因为地址保持阶段的持续时间始终是 1 个 Flash 时钟周期。	1111b
[3:0]	ADDSET[3:0]	RW	地址建立阶段的持续时间: 通过软件写入这些位可定义以 HCLK 周期表示的地址建立阶段持续时间 (请参考图 40-3~图 40-15), 适用于异步访问模式: 0000: ADDSET 阶段的持续时间=0*HCLK 时钟周期; ... 1111: ADDSET 阶段的持续时间=15*HCLK 时钟周期。	1111b

			注：在同步访问模式下，该值不使用，因为地址建立阶段的持续时间始终是 1 个 Flash 时钟周期。 复用模式下，ADDSET 的最小值为 1。	

40.6 NAND Flash 控制器

FMC 会生成相应的信号时序，用于驱动以下类型的设备：8 位和 16 位 NAND Flash。

NAND 存储区域通过专用的寄存器配置（第 40.6.7 节）。可编程的存储器参数包括访问时序（如下表所示）和 ECC 配置。

表 40-38 可编程的 NAND Flash 访问参数

参数	功能	访问模式	单位	最小值	最大值
存储器建立时间	命令使能前地址建立 所需时钟周期（HCLK）数	读/写	HB 时钟周期 （HCLK）	1	255
存储器等待	命令使能的最小持续时间 （按 HCLK 时钟周期计）	读/写	HB 时钟周期 （HCLK）	2	255
存储器保持	命令禁止后，必须保持地址 （如果进行了写访问还需保持数 据）的时钟周期（HCLK）数	读/写	HB 时钟周期 （HCLK）	1	254
存储器数据总线 高阻态	开始进行写访问后，数据总线 保持高阻状态期间的时钟 周期（HCLK）数	写	HB 时钟周期 （HCLK）	1	255

40.6.1 外部存储器接口信号

8 位 NAND Flash

表 40-39 8 位 NAND Flash

FMC 信号名称	I/O	功能
A[17]	0	NAND Flash 地址锁存使能（ALE）信号
A[16]	0	NAND Flash 命令锁存使能（CLE）信号
D[7:0]	I/O	8 位复用双向地址/数据总线
NCE	0	片选
NOE（=NRE）	0	输出使能（存储器信号名称：读取使能，NRE）
NWE	0	写入使能
NWAIT/INT	I	输入 FMC 的 NAND Flash 就绪/繁忙信号

由于 FMC 能够管理足够多的地址周期，因此理论上不存在容量限制。

注：前缀“N”标识低电平有效的信号。

16 位 NAND Flash

表 40-40 16 位 NAND Flash

FMC 信号名称	I/O	功能
A[17]	0	NAND Flash 地址锁存使能（ALE）信号
A[16]	0	NAND Flash 命令锁存使能（CLE）信号
D[15:0]	I/O	16 位复用双向地址/数据总线
NCE	0	片选
NOE（=NRE）	0	输出使能（存储器信号名称：读取使能，NRE）
NWE	0	写入使能

NWAIT/INT	I	输入 FMC 的 NAND Flash 就绪/繁忙信号

由于 FMC 能够管理足够多的地址周期，因此理论上不存在容量限制。

注：前缀“N”标识低电平有效的信号。

40.6.2 NAND Flash 支持的存储器和事务

下表介绍了所支持的设备、访问模式和事务。NAND Flash 控制器不允许（或不支持）的事务以灰色显示。

表 40-41 支持的存储器和事务

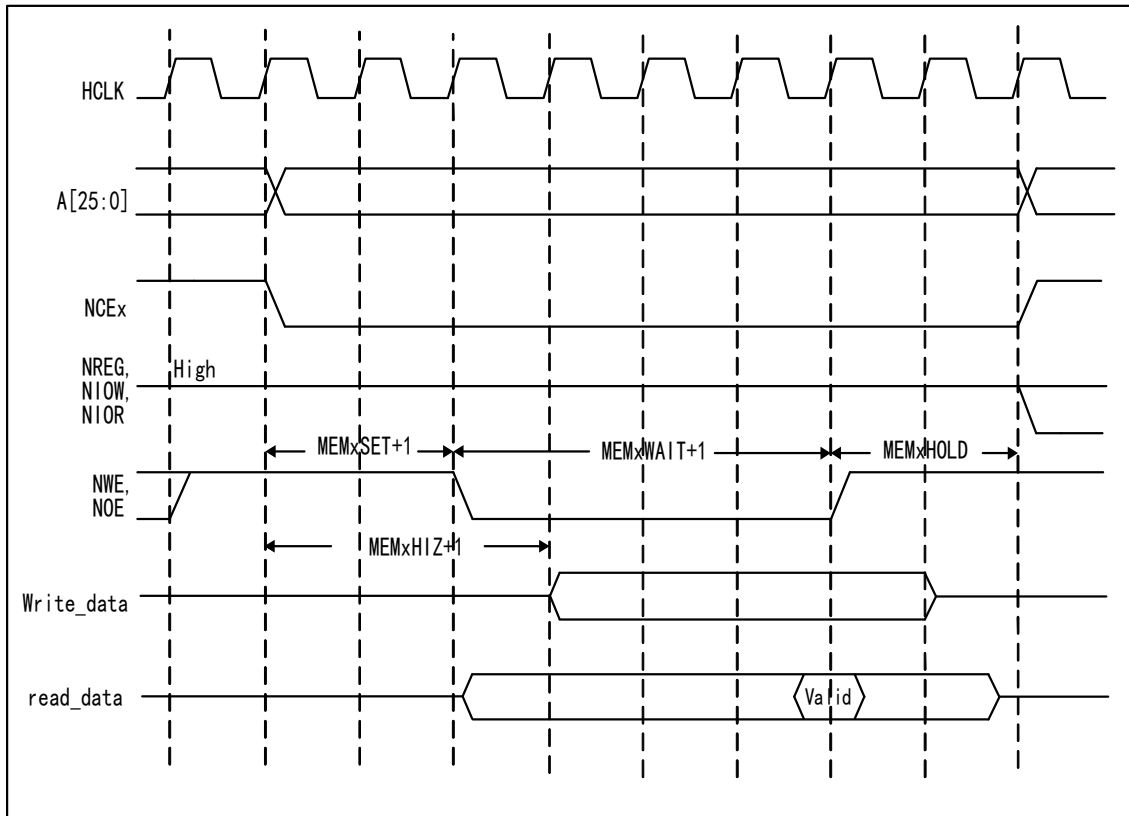
设备	模式	R/W	HB 数据大小	存储器 数据大小	允许/不允许	注释
8 位 NAND	异步	R	8	8	是	—
	异步	W	8	8	是	—
	异步	R	16	8	是	分为 2 次 FMC 访问
	异步	W	16	8	是	分为 2 次 FMC 访问
	异步	R	32	8	是	分为 4 次 FMC 访问
	异步	W	32	8	是	分为 4 次 FMC 访问
16 位 NAND	异步	R	8	16	是	—
	异步	W	8	16	否	—
	异步	R	16	16	是	—
	异步	W	16	16	是	—
	异步	R	32	16	是	分为 2 次 FMC 访问
	异步	W	32	16	是	分为 2 次 FMC 访问

40.6.3 NAND Flash 的时序图

NAND Flash 存储区域通过以下一组寄存器进行管理：

- 控制寄存器：FMC_PCR
- 中断状态寄存器：FMC_SR
- ECC 寄存器：FMC_ECCR
- 通用存储器空间的时序寄存器：R32_FMC_PMEM
- 特性存储器空间的时序寄存器：R32_FMC_PATT

图 40-21 NAND Flash 控制器的通用存储器访问波形



注：进行写访问期间 NOE 保持高电平（无效）。进行读访问期间 NWE 保持高电平（无效）。

每个时序配置寄存器包含四个参数，其中三个参数用于定义 NAND Flash 的三个访问阶段的 HCLK 周期数，另一个参数用于定义进行写访问时开始驱动数据总线的时序。上图介绍了通用存储器访问的时序参数定义，特性存储器空间的访问时序与此类似。

40.6.4 NAND Flash 操作

NAND Flash 设备的命令锁存使能 (CLE) 信号和地址锁存使能 (ALE) 信号由 FMC 控制器的地址信号驱动。这意味着要向 NAND Flash 发送命令或地址，CPU 必须对其存储器空间中的特定地址执行写操作。

从 NAND Flash 设备进行的典型页读取操作需要执行以下步骤：

1) 根据 NAND Flash 的特性 (PWID 位指示 NAND Flash 数据总线宽度, PTP=1, 根据需要 PWAITEN=0 或 1; 有关时序配置, 请参考第 40.4.2 节: NAND Flash 地址映射), 配置 FMC_PCR 和 FMC_PMEM (对于某些设备, 配置 R32_FMC_PATT, 请参考第 40.6.5 节: NAND Flash 预等待功能) 寄存器, 进而配置和使能相应的存储区域。

2) CPU 向通用存储器空间执行字节写操作, 此时数据字节等于一个 Flash 命令字节 (例如 Samsung NAND Flash 设备的字节为 0x00)。在写入选通期间 (NWE 上为低电平脉冲), NAND Flash 的 LE 输入有效, 因此将已写入的字节视为 NAND Flash 的一个命令。该命令由存储器设备锁存后, 随后进行页读取操作时, 不需要再次写入该命令。

3) 通过在通用存储器空间或特性空间写入 STARTAD[7:0]、STARTAD[16:9]、STARTAD[24:17] 以及 STARTAD[25] (针对 64Mb*8 位的 NAND Flash) 这 4 个字节 (对于容量较小的设备写入 3 个字节), CPU 能够发送读操作的起始地址 (STARTAD)。在写入选通期间 (NWE 上为低电平脉冲), NAND Flash 设备的 ALE 输入有效, 因此将已写入的字节视为读操作的起始地址。借助特性存储器空间, 可使用 FMC 的另一种不同时序配置, 实现某些 NAND Flash 所需的预等待功能 (有关详细信息, 请参考第 40.6.5 节: NAND Flash 预等待功能)。

4) 控制器在对同一个或另一个存储区域进行新访问之前, 等待 NAND Flash 准备好 (R/NB 信号处

于高电平)。等待期间, 控制器保持 NCE 信号有效(低电平)。

5) 然后 CPU 能够从通用存储器空间执行字节读操作, 进而按字节读取 NAND Flash 页(数据字段+备用字段)。

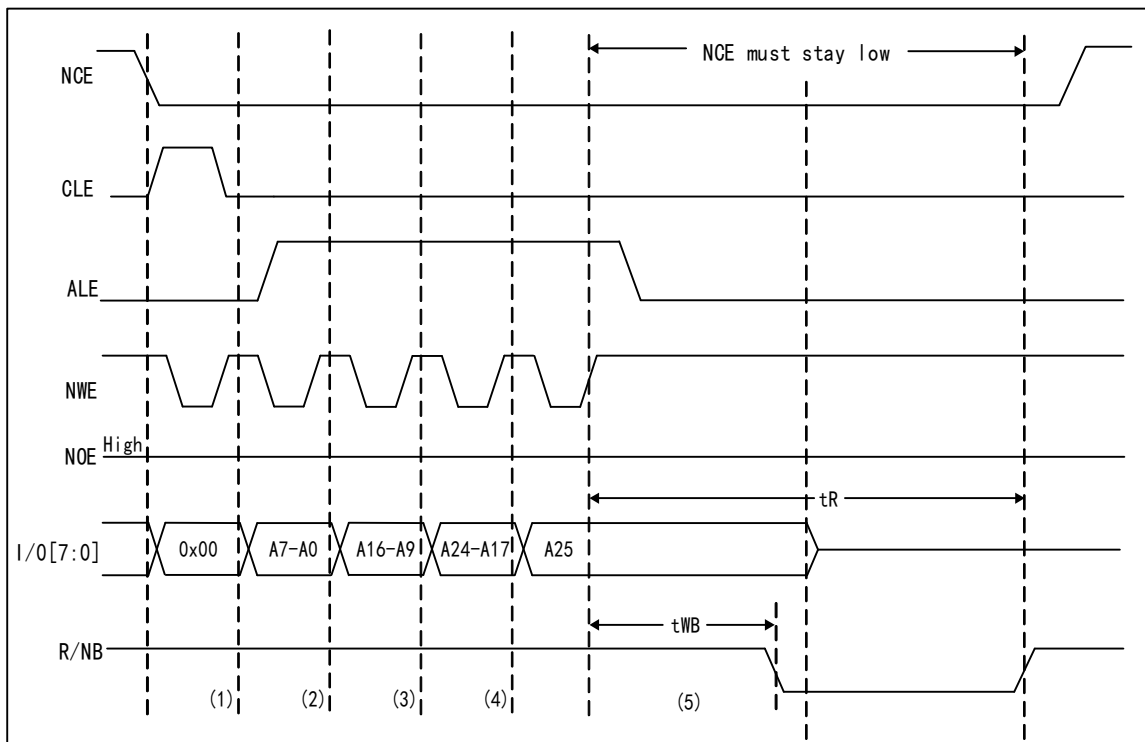
6) 可读取下一个 NAND Flash 页, 而无需任何 CPU 命令或地址写操作。可采用以下三种不同的方式实现:

- 执行步骤 5 中介绍的操作
- 通过重新开始步骤 3 中的操作随机访问一个新地址
- 通过重新开始步骤 2 向 NAND Flash 设备发送新命令

40.6.5 NAND Flash 预等待功能

一些 NAND Flash 设备需要在写入地址的最后一部分后, 控制器等待 R/NB 信号变为低电平, 如下图所示。

图 40-22 访问非“CE 无关”NAND-Flash



- 1) CPU 在地址 0x70010000 处写入字节 0x00。
- 2) CPU 在地址 0x70020000 处写入字节 A7~A0。
- 3) CPU 在地址 0x70020000 处写入字节 A16~A9。
- 4) CPU 在地址 0x70020000 处写入字节 A24~A17。

5) CPU 在地址 0x78020000 处写入字节 A25: FMC 通过 R32_FMC_PATT2 时序定义执行写访问, 其中 $ATTHOLD \geq 7$ (假设 $(7+1) * HCLK = 112ns > t_{WB}$ 最大值)。这可确保 NCE 保持低电平, 直到 R/NB 再次变为低电平后变为高电平(只有 NCE 信号对之有作用的 NAND Flash 才需要此功能)。

当需要此功能时, 可通过编程 MEMHOLD 值确保满足 t_{WB} 时序。然而 CPU 对 NAND Flash 进行的所有读或写访问均会经过 $(MEMHOLD+1)$ 个 HCLK 周期的保持延迟(该延迟插入在 NWE 信号的上升沿与下一访问之间)。

要克服该时序限制, 可使用特性存储器空间, 将其时序寄存器编程为满足 t_{WB} 时序的 ATTHOLD 值并将 MEMHOLD 保持为其最小值。然后, CPU 必须使用通用存储器空间进行所有的 NAND Flash 读和写访问, 向 NAND Flash 设备写入最后一个地址字节时除外, 此时 CPU 必须对特性存储器空间执行写操作。

40.6.6 纠错码（ECC）计算（NAND Flash）

FMC NAND 卡控制器包括两个纠错码计算硬件模块，每个存储区域各有一个。这些模块可在软件处理 ECC 时减少主机 CPU 工作负载。

这两个 ECC 模块相同，分别与存储区域 2 和存储区域 3 相关联。因此，硬件 ECC 计算均不适用于与存储区域 4 相连的存储器。

对 NAND Flash 执行读取或写入操作时，相应每 256、512、1024、2048、4096 或 8192 个字节，FMC 中使用的 ECC 算法可修正 1 位错误并且检测出 2 位错误。该操作基于 Hamming 编码算法，并且包括计算行和列奇偶校验。

每当 NAND Flash 存储区域处于激活状态时，ECC 模块均会监视 NAND Flash 数据总线和读/写信号（NCE 和 NWE）。

ECC 按如下说明操作：

- 当访问 NAND Flash 存储区域 2 或存储区域 3 时，将锁存 D[15:0] 总线上出现的数据并将其用于 ECC 计算。
- 当访问 NAND Flash 中的任何其它地址时，ECC 逻辑会进入空闲状态，不执行任何操作。因此，进行 ECC 计算时，用于定义 NAND Flash 命令或地址的写操作无效。

主机 CPU 对 NAND Flash 完成所需字节数的读取/写入操作后，必须读取 R32_FMC_ECCR 寄存器，才能检索计算出的值。读取后，应通过将 ECCEN 位复位为零来将这些寄存器清零。要计算新的数据块，必须将 FMC_PCR 寄存器中的 ECCEN 位置 1。

要进行 ECC 计算：

- 1) 使能 R32_FMC_PCR 寄存器中的 ECCEN 位。
- 2) 将数据写入 NAND Flash 页。在写入 NAND 页期间，ECC 模块将计算 ECC 值。
- 3) 读取 R32_FMC_ECCR 寄存器中所提供的 ECC 值，并将其存储到变量。
- 4) 将 R32_FMC_PCR 寄存器中的 ECCEN 位清零后使能，然后从 NAND 页回读写入的数据。在读取 NAND 页期间，ECC 模块将计算 ECC 值。
- 5) 读取 R32_FMC_ECCR 寄存器中所提供的新 ECC 值。
- 6) 如果两次读取的 ECC 值相同，则无需校正，否则说明存在 ECC 错误，并且软件校正例程将返回有关该错误是否能够得到校正的信息。

40.6.7 NAND Flash 控制寄存器

表 40-42 NAND Flash 相关寄存器列表

名称	访问地址	描述	复位值
R32_FMC_PCR	0x40025480	NAND Flash 控制寄存器	0x00000018
R32_FMC_SR	0x40025484	FIFO 状态和中断寄存器	0x00000040
R32_FMC_PMEM	0x40025488	通用存储器空间时序寄存器	0xFCFCFCFC
R32_FMC_PATT	0x4002548C	特性存储器空间时序寄存器	0xFCFCFCFC
R32_FMC_ECCR	0x40025494	ECC 结果寄存器	0x00000000

40.6.7.1 NAND Flash 控制寄存器（R32_FMC_PCR）

偏移地址：0x80

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	PWAI	Reser	

							TEN	ved

位	名称	访问	描述	复位值
[31:20]	Reserved	RO	保留。	0
[19:17]	ECCPS[2:0]	RW	ECC 页大小： 000: 256 字节； 001: 512 字节； 010: 1024 字节； 011: 2048 字节； 100: 4096 字节； 101: 8192 字节。	0
[16:13]	TAR[3:0]	RW	ALE 到 RE 的延迟： 以 HB 时钟周期数 (HCLK) 设置从 ALE 低电平到 RE 低电平的时间： $t_{ar} = (TAR + SET + 2) * THCLK$ ，其中 THCLK 是 HCLK 时钟周期。 0000: 1 个 HCLK 周期（默认）； 1111: 16 个 HCLK 周期。 注：根据寻址空间，SET 为 MEMSET 或 ATTSET。	0
[12:9]	TCLR[3:0]	RW	CLE 到 RE 的延迟： 以 HB 时钟周期数 (HCLK) 设置从 CLE 低电平到 RE 低电平的时间： $t_{clr} = (TCLR + SET + 2) * THCLK$ ，其中 THCLK 是 HCLK 时钟周期。 0000: 1 个 HCLK 周期（默认）； 1111: 16 个 HCLK 周期。 注：根据寻址空间，SET 为 MEMSET 或 ATTSET。	0
[8:7]	Reserved	RO	保留。	0
6	ECCEN	RW	ECC 计算逻辑使能位： 1: 使能 ECC 逻辑； 0: 禁止和复位 ECC 逻辑（复位后为默认值）。	0
[5:4]	PWID[1:0]	RW	数据总线宽度： 00: 8 位； 01: 16 位； 10: 保留； 11: 保留。	01b
3	PTYP	RW	存储器类型： 定义附加到相应存储区域的设备的类型： 1: NAND Flash（复位后为默认值）； 0: 保留，必须保持复位值。	1
2	PBKEN	RW	NAND Flash 存储区域使能位： 使能存储区域。访问禁止的存储区域会引起 HB 总线上的错误。 1: 使能相应的存储区域； 0: 禁止相应的存储区域（复位后为默认值）。	0
1	PWAITEN	RW	等待特性使能位： 1: 使能；	0

			0: 禁用。	
0	Reserved	R0	保留。	0

40.6.7.2 FIFO 状态和中断寄存器 (R32_FMC_SR)

偏移地址: 0x84

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	R0	保留。	0
6	FEMPT	R0	FIFO 为空: 1: FIFO 为空; 0: FIFO 非空。	1
5	IFEN	RW	中断下降沿检测使能位: 1: 中断下降沿检测请求使能; 0: 中断下降沿检测请求禁止。	0
4	ILEN	RW	中断高电平检测使能位: 1: 中断高电平检测请求使能; 0: 中断高电平检测请求禁止。	0
3	IREN	RW	中断上升沿检测使能位: 1: 中断上升沿检测请求使能; 0: 中断上升沿检测请求禁止。	0
2	IFS	RWO	中断下降沿状态: 此标志由硬件置 1, 由软件复位, 写 0 清除。 1: 出现中断下降沿; 0: 未出现中断下降沿。	0
1	ILS	RWO	中断高电平状态: 此标志由硬件置 1, 由软件复位, 写 0 清除。 1: 出现中断高电平; 0: 未出现中断高电平。	0
0	IRS	RWO	中断上升沿状态: 此标志由硬件置 1, 由软件复位, 写 0 清除。 1: 出现中断上升沿; 0: 未出现中断上升沿。	0

40.6.7.3 通用存储器空间时序寄存器 (R32_FMC_PMEM)

偏移地址: 0x88

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

位	名称	访问	描述	复位值
[31:24]	MEMHIZ[7:0]	RW	通用存储器数据总线高阻态时间： 00000000：1 个 HCLK 周期； 11111110：255 个 HCLK 周期； 11111111：保留。	1111 1100b
[23:16]	MEMHOLD[7:0]	RW	通用存储器保持时间： 00000000：保留； 00000001：1 个 HCLK 周期； 11111110：254 个 HCLK 周期； 11111111：保留。	1111 1100b
[15:8]	MEMWAIT[7:0]	RW	通用存储器等待时间： 00000000：保留； 00000001：2 个 HCLK 周期（+禁止 NWAIT 时引入的等待周期）； 11111110：255 个 HCLK 周期（+禁止 NWAIT 时引入的等待周期）； 11111111：保留。	1111 1100b
[7:0]	MEMSET[7:0]	RW	通用存储器建立时间： 00000000：1 个 HCLK 周期； 11111110：255 个 HCLK 周期； 11111111：保留。	1111 1100b

40.6.7.4 特性存储器空间时序寄存器（R32_FMC_PATT）

偏移地址：0x8C

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

位	名称	访问	描述	复位值
[31:24]	ATTHIZ[7:0]	RW	特性存储器数据总线高阻态时间： 00000000：0 个 HCLK 周期； 11111110：255 个 HCLK 周期； 11111111：保留。	1111 1100b
[23:16]	ATTHOLD[7:0]	RW	特性存储器保持时间： 00000000：保留； 00000001：1 个 HCLK 周期； 11111110：254 个 HCLK 周期； 11111111：保留。	1111 1100b
[15:8]	ATTWAIT[7:0]	RW	特性存储器等待时间： 00000000：保留； 00000001：2 个 HCLK 周期（+禁止 NWAIT 时引入的等待周期）；	1111 1100b

			11111110: 255 个 HCLK 周期 (+禁止 NWAIT 时引入的等待周期) ; 11111111: 保留。	
[7:0]	ATTSET[7:0]	RW	特性存储器建立时间: 00000000: 1 个 HCLK 周期; 11111110: 255 个 HCLK 周期; 11111111: 保留。	1111 1100b

40. 6. 7. 5 ECC 结果寄存器 (R32_FMC_ECCR)

偏移地址: 0x94

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

位	名称	访问	描述	复位值																					
[31:0]	ECC	R0	ECC 结果： 该字段包含由 ECC 计算逻辑计算所得的值，如下表所示。	0																					
			<table><tr><th>ECCPS[2:0]</th><th>以字节为单位的页大小</th><th>ECC 位</th></tr><tr><td>000</td><td>256</td><td>ECC[21:0]</td></tr><tr><td>001</td><td>512</td><td>ECC[23:0]</td></tr><tr><td>010</td><td>1024</td><td>ECC[25:0]</td></tr><tr><td>011</td><td>2048</td><td>ECC[27:0]</td></tr><tr><td>100</td><td>4096</td><td>ECC[29:0]</td></tr><tr><td>101</td><td>8192</td><td>ECC[31:0]</td></tr></table>		ECCPS[2:0]	以字节为单位的页大小	ECC 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]
			ECCPS[2:0]		以字节为单位的页大小	ECC 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]																							

40. 7 SDRAM 控制器

40. 7. 1 SDRAM 控制器主要特性

SDRAM 控制器的主要特性如下:

- 支持字、半字和字节访问
- 掉电模式
- CAS 延迟 1, 2, 3
- 自刷新模式
- 可编程时序参数
- 通过软件进行 SDRAM 上电初始化
- 两个 SDRAM 存储区域, 可独立配置
- 多存储区域乒乓访问
- SDRAM 时钟可以是 HCLK、HCLK/2 或 HCLK/3
- 支持自动刷新操作, 可编程刷新速率
- 8 位、16 位、32 位存储器数据总线宽度
- 自动进行和存储区域边界管理
- 读 FIFO 可缓存, 支持 6 行*32 位深度 (6*14 位地址标记)

- 13 位地址行，11 位地址列，4 个内部存储区域：4*16M*32bit（256MB）、4*16M*16bit（128MB）、4*16M*8bit（64MB）

40.7.2 SDRAM 外部存储器接口信号

启动时，必须通过用户应用程序对用于连接 FMC SDRAM 控制器与外部 SDRAM 设备的 SDRAM I/O 引脚进行配置。应用程序未使用的 SDRAM 控制器 I/O 引脚可用于其它用途。

表 40-43 SDRAM 信号

SDRAM 信号	I/O 类型	说明	复用功能
SDCLK	0	SDRAM 时钟	—
SDCKE[1:0]	0	SDCKE0: SDRAM 存储区域 1 时钟使能 SDCKE1: SDRAM 存储区域 2 时钟使能	—
SDNE[1:0]	0	SDNE0: SDRAM 存储区域 1 芯片使能 SDNE1: SDRAM 存储区域 2 芯片使能	—
A[12:0]	0	地址	FMC_A[12:0]
D[31:0]	I/O	双向数据总线	FMC_D[31:0]
BA[1:0]	0	存储区域地址	FMC_A[15:14]
NRAS	0	行地址选通	—
NCAS	0	列地址选通	—
SDNWE	0	写入使能	—
DQM[3:0]	0	写访问的输出字节屏蔽 (存储器信号名称: DQM[3:0])	—

40.7.3 SDRAM 控制器功能说明

所有 SDRAM 控制器输出（信号、地址和数据）可配置为在存储器时钟（FMC_SDCLK）的下降沿上变化。

SDRAM 初始化

初始化序列通过软件进行管理。如果使用了两个存储区域，则必须将 R32_FMC_SDCMR 寄存器中的目标存储区域位 CTB1 和 CTB2 置 1，同时为存储区域 1 和存储区域 2 生成初始化序列：

1) 将存储器设备的特性编程到 R32_FMC_SDCRx 寄存器中。SDRAM 时钟频率、RBURST 和 RPIPE 特性必须编程到 R32_FMC_SDCR1 寄存器中。

2) 将存储器设备的时序编程到 R32_FMC_SDTRx 寄存器中。TRP 和 TRC 时序必须编程到 R32_FMC_SDTR1 寄存器中。

3) 将 R32_FMC_SDCMR 寄存器中的 MODE[2:0] 位置为“001”并配置 R32_FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）以开始为存储器提供时钟信号（SDCKE 驱动为高电平）。

4) 等待指定延迟周期。典型延迟为 100us。

5) 将 R32_FMC_SDCMR 寄存器中的 MODE[2:0] 位置为“010”并配置 R32_FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）以发送“全部预充电”命令。

6) 将 R32_FMC_SDCMR 寄存器中的 MODE[2:0] 位置为“011”并配置 R32_FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）和连续自动刷新命令（NRFS）的数量。

7) 根据 SDRAM 设备配置 MRD 字段，将 R32_FMC_SDCMR 寄存器中的 MODE[2:0] 位置为“100”并配置 R32_FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）以发送“加载模式寄存器”命令并对 SDRAM 设备进行编程。尤其是：

a) 必须根据 R32_FMC_SDCR1/2 中的值来选择 CAS 延迟；

b) 必须通过 R32_FMC_SDCMR 寄存器中的 MRD[2:0] 位设置为 000 来将突发长度（BL）设置为 1。

如果两个 SDRAM 存储区域的模式寄存器不同，则此步骤必须重复两次，每个存储区域各一次且目

标存储区域位相应置 1。

8) 编程 R32_FMC_SDRTR 寄存器中的刷新速率。刷新速率对应于刷新周期之间的延迟。其值必须与 SDRAM 设备相适应。

在这一阶段，SDRAM 设备已做好接受命令的准备。如果进行 SDRAM 访问期间发生系统复位，则数据总线仍可能由 SDRAM 设备驱动。因此，必须在复位后重新初始化 SDRAM 设备，NOR Flash/PSRAM/SRAM 或 NAND Flash 控制器才能发送新的访问命令。

注：如果有两个 SDRAM 设备连接到 FMC，对命令模式寄存器同时访问这两个器件（加载模式寄存器命令）的情况，将按 R32_FMC_SDR1 寄存器中为 SDRAM 存储区域 1 配置的时序参数（TMRD 和 TRAS 时序）发出访问命令。

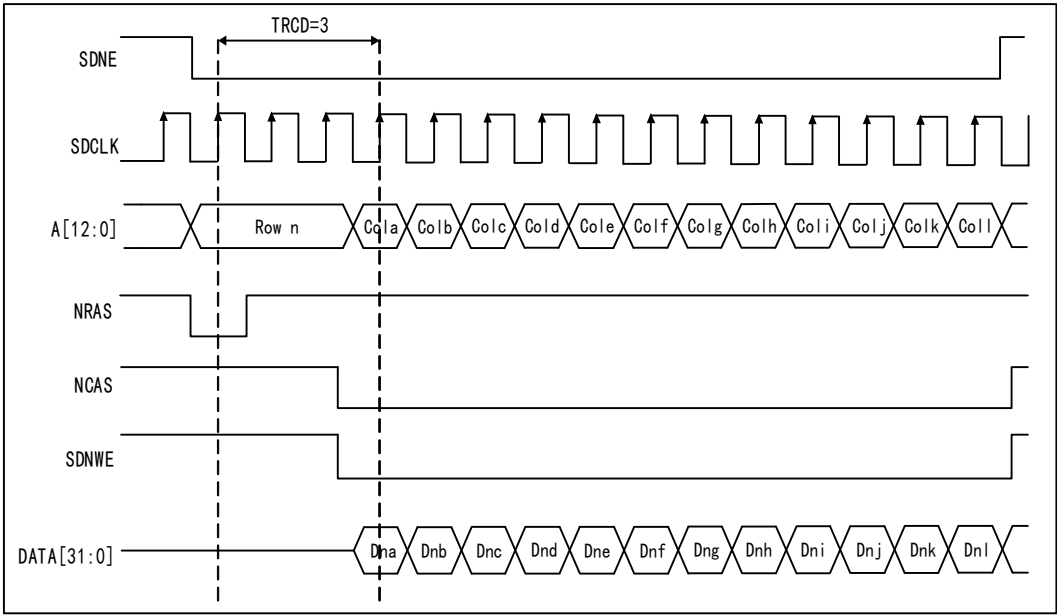
SDRAM 控制器写周期

SDRAM 控制器可接收单次的和突发的写请求，并将其视为单次存储器访问。在这两种情况下，SDRAM 控制器都会跟踪各存储区域的有效行，以能够对不同的存储区域进行连续的写访问（多存储区域乒乓访问）。执行任何写访问前，必须将 R32_FMC_SDCRx 寄存器中的 WP 位清零，禁止 SDRAM 存储区域的写保护。

SDRAM 控制器始终会检查下一个访问：

- 1) 如果下一个访问发生在同一行或在其它有效行，则直接执行写操作。
- 2) 如果下一个访问指向一个无效行，则 SDRAM 控制器将生成预充电命令、激活该新行并初始化写命令。

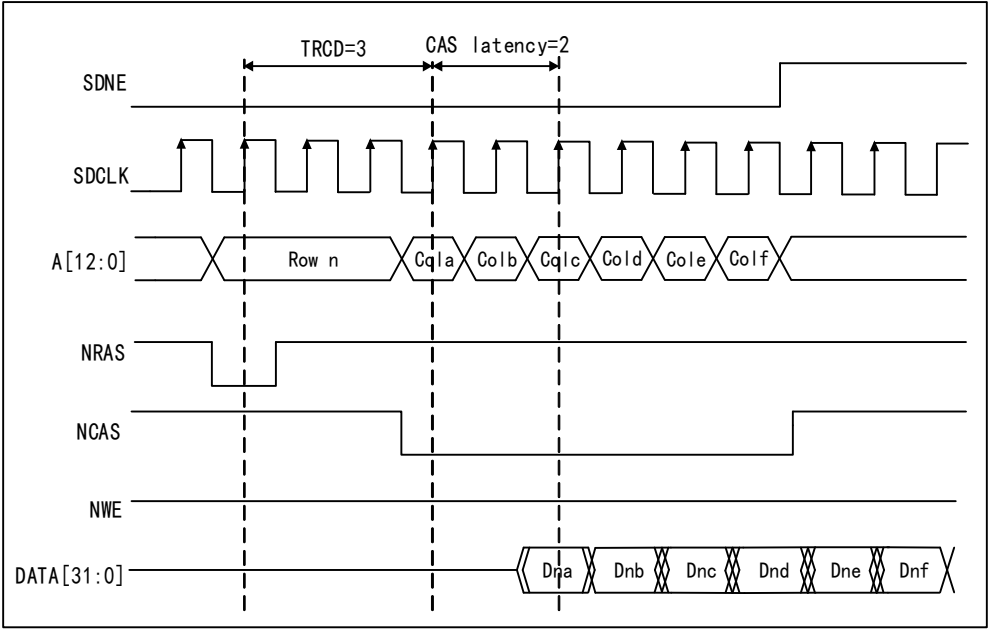
图 40-23 突发写入 SDRAM 访问波形



SDRAM 控制器读周期

SDRAM 控制器可接收单次的和突发的读请求，并将其视为单次存储器访问。在这两种情况下，SDRAM 控制器都会跟踪各存储区域的有效行，以能够对不同的存储区域进行连续的读访问（多存储区域乒乓访问）。

图 40-24 突发读 SDRAM 访问



FMC SDRAM 控制器具有可缓存的读 FIFO（6 行*32 位），它用于根据以下公式在 CAS 延迟周期和 RPIPE 延迟期间提前存储读取的数据。必须将 R32_FMC_SDCR1 寄存器中的 RBURST 位置 1 才能接受下一个读访问。预期的数据数量=CAS 延迟+1+（RPIPE 延迟）/2，示例：

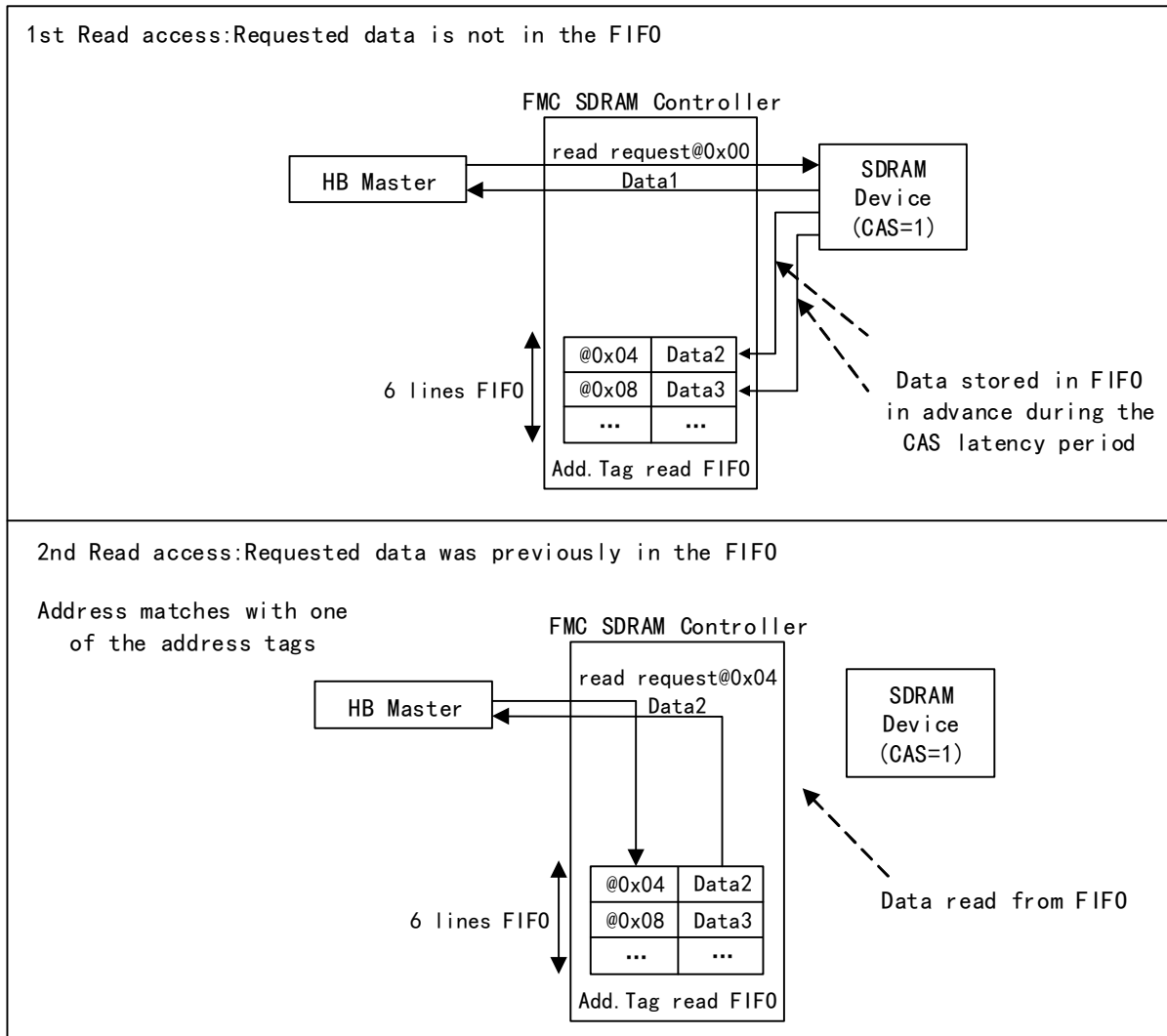
- CAS 延迟=3，RPIPE 延迟=0：四个数据（未提交）存储在 FIFO 中。
- CAS 延迟=3，RPIPE 延迟=2：五个数据（未提交）存储在 FIFO 中。

读 FIFO 的每行都具有 14 位地址标记用于标识自身内容：11 位用于表示列地址，2 位用于表示选择内部存储区域和有效行，1 位用于选择 SDRAM 设备，在 HB 突发读取期间，如果提前到达行末尾，则提前读取的数据（未提交）不存储到读 FIFO。对于单次读访问，数据将正确存储到读 FIFO。

每当出现读请求时，SDRAM 控制器将检查：

- 地址与地址标记之一是否匹配，如果找到匹配，则直接从 FIFO 读取数据并清空相应地址标记/行内容，同时压缩 FIFO 中的剩余数据以避免空行。
- 否则，向存储器发送新的读命令并用新数据更新 FIFO。如果 FIFO 已满，则较早的数据将丢失。

图 40-25 RBURST 位置 1 时的读访问逻辑图 (CAS=1, RPIPE=0)



读访问或预充电命令期间，读 FIFO 将刷新并做好填充新数据的准备。

接到第一个读请求后，如果当前访问还未进行到行边界，则 SDRAM 控制器将在 CAS 延迟周期和 RPIPE 延迟（如果已配置）期间接受下一个读访问。这将通过递增存储器地址来实现。必须满足以下条件：

- R32_FMC_SDCR1 寄存器中的 RBURST 控制位必须置 1
地址管理取决于下一个 HB 请求：
 - 下一个 HB 请求是连续的（HB 突发）
这种情况下，SDRAM 控制器将递增地址。
 - 下一个 HB 请求不连续
 - 若新的读请求指向与上一请求相同的行或另一个有效行，则新地址将传送给存储器，同时主设备在 CAS 延迟周期停止工作，等待从存储器获取新数据。
 - 若新的读请求指向无效行，则 SDRAM 控制器生成预充电命令、激活该新行并初始化读命令。
- 如果 RBURST 位置 0，则不使用读 FIFO。

行和存储区域边界管理

当读/写访问跨越了行边界时，如果下一个读/写访问是连续的并且当前访问已执行到行界，则 SDRAM 控制器将执行以下操作：

- 1) 对有效行进行预充电；

2) 激活新行;

3) 启动读/写命令。

对于各种列和数据总线宽度配置，都支持在行边界自动激活下一行。

SDRAM 控制器可根据需要在以下命令之间插入附加时钟周期:

- 在预充电和激活命令之间插入以匹配 TRP 参数（仅当下一个访问指向同一存储区域中的其它行时）
- 在激活和读命令之间插入以匹配 TRCD 参数。

这些参数在 FMC_SDTRx 寄存器中定义。

有关跨越行边界读取和突发写访问的信息，请参考下图。

图 40-26 跨行边界的读访问

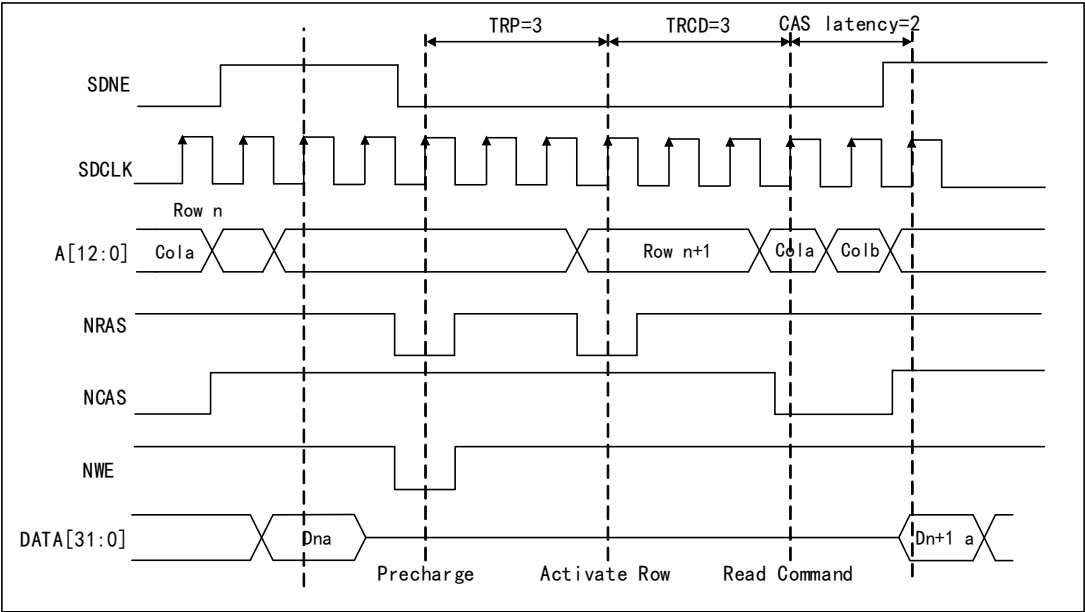
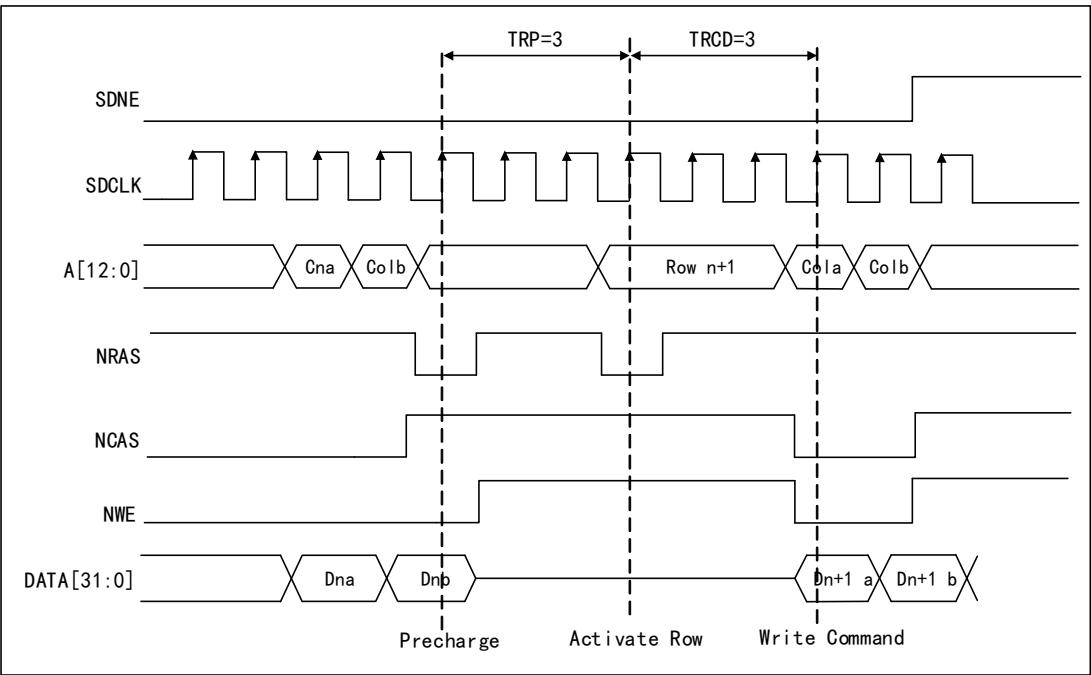


图 40-27 跨行边界的写访问



如果下一个访问是连续的并且当前访问跨越了存储区域边界，则 SDRAM 控制器将激活下一个存储

区域的第一个行并发出新的读/写命令。可能存在以下两种情况：

- 如果当前存储区域不是最后一个存储区域，则必须对新存储区域中的激活行进行预充电。对于各种行/列和数据总线宽度配置，都支持在存储区域边界自动激活下一行。
- 如果当前存储区域是最后一个存储区域，则仅在对 13 位行、11 位列、4 个内部存储区域和 32 位数据总线的 SDRAM 设备寻址时，支持自动激活下一行。否则，将违反 SDRAM 地址范围并生成 HB 错误。
- 对于 13 位行地址、11 位列地址、4 个内部存储区域和总线宽度为 32 位的 SDRAM 存储器，SDRAM 控制器将通过第二个 SDRAM 设备继续进行读/写操作（假设第二个 SDRAM 设备已初始化）：
 - SDRAM 控制器将激活第一行（在对有效行预充电之后，假设第一个内部存储区域中存在有效行）并发出新的读/写命令；
 - 如果第一行已激活，则 SDRAM 控制器将仅发出读/写命令。

SDRAM 控制器刷新周期

自动刷新命令用于刷新 SDRAM 设备的内容。SDRAM 控制器会定期发送自动刷新命令。它使用一个内部计数器装载 R32_FMC_SDRTR 寄存器中的 COUNT 值。该值定义刷新周期之间的存储器时钟周期个数（刷新速率）。该计数器的值达到零时将生成一个内部脉冲。

如果存在进行中的存储器访问，则会延迟自动刷新请求。不过，在存储器访问和自动刷新请求同时出现时，则优先处理自动刷新请求。

如果在自动刷新期间访问存储器，则会缓存访问请求并在自动刷新完成后进行处理。

如果在上一个自动刷新请求尚未完成的情况下又出现了新的自动刷新请求，则状态寄存器中的 RE（刷新错误）位将置 1。该位如果已使能（REIE=1），将生成中断。

如果 SDRAM 的行不是空闲状态（并非所有行都已关闭），则 SDRAM 控制器将生成 PALL（全部预充电）命令，然后再进行自动刷新。

如果由 R32_FMC_SDCMR 命令模式寄存器（模式位=“011”）生成自动刷新命令，则必须先发出 PALL 命令（模式位=“010”）。

40.7.4 低功耗模式

可使用两种低功耗模式：

- 自刷新模式
 - 由 SDRAM 设备自身执行自动刷新循环以保留数据，无需外部时钟。
- 掉电模式
 - 由 SDRAM 控制器执行自动刷新循环。

自刷新模式

通过将 MODE 位置为“101”并配置 R32_FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）来选择该模式。

SDRAM 时钟在 TRAS 延迟后停止运行，而内部刷新定时器只有在满足以下条件之一时才停止计数：

- 向两个设备都发出了自刷新命令
- 其中一个设备未激活（SDRAM 存储区域未初始化）。

进入自刷新模式前，SDRAM 控制器会自动发送 PALL 命令。

如果写数据 FIFO 非空，则所有数据都将在自刷新模式激活前发送到存储器并且 BUSY 状态标志保持置 1。

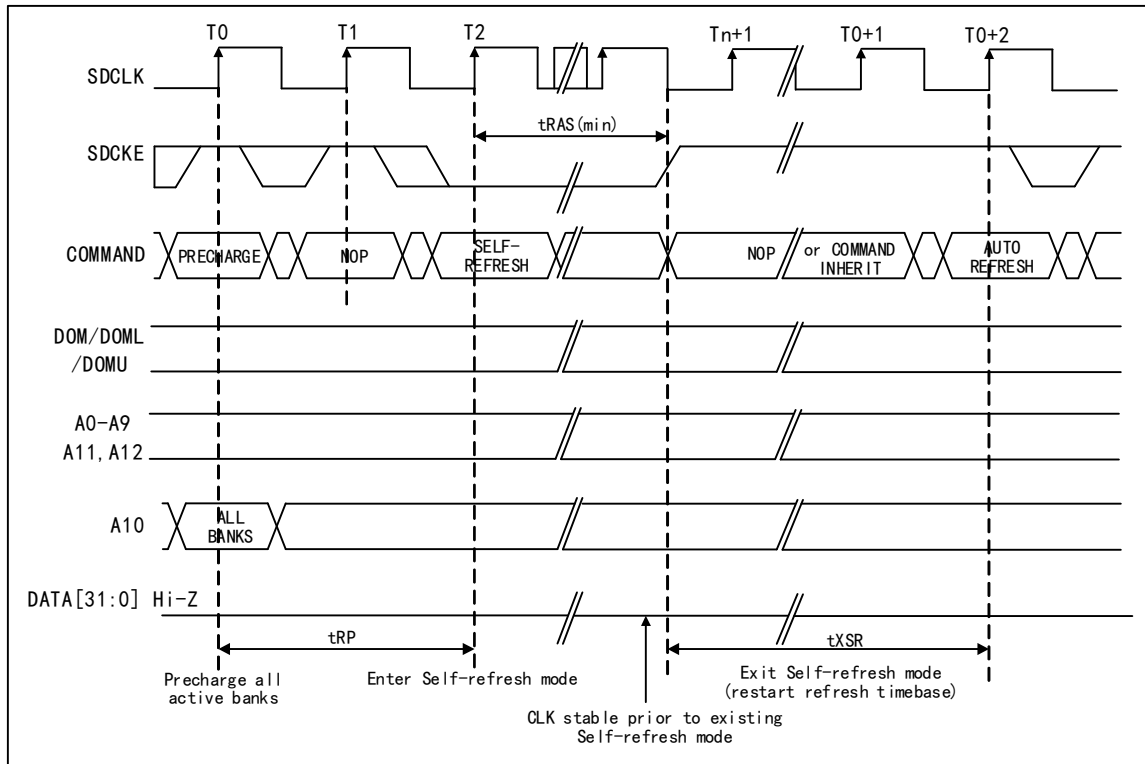
在自刷新模式下，除保持低电平的 SDCKE 外，SDRAM 设备的所有输入都无效。

SDRAM 设备必须处于自刷新模式最短为 TRAS 时间，且能够在更长的时间内始终处于自刷新模式。为保证这一最短时长，在自刷新激活后的 TRAS 延迟期间，BUSY 状态标志将保持高电平。SDRAM 控制器会在有 SDRAM 设备被选定后立即生成一个命令序列以退出自刷新模式。存储器访问完成

后，选定的设备将保持正常模式。

要退出自刷新模式，必须将 MODE 位置为“000”（正常模式）并配置 R32_FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）。

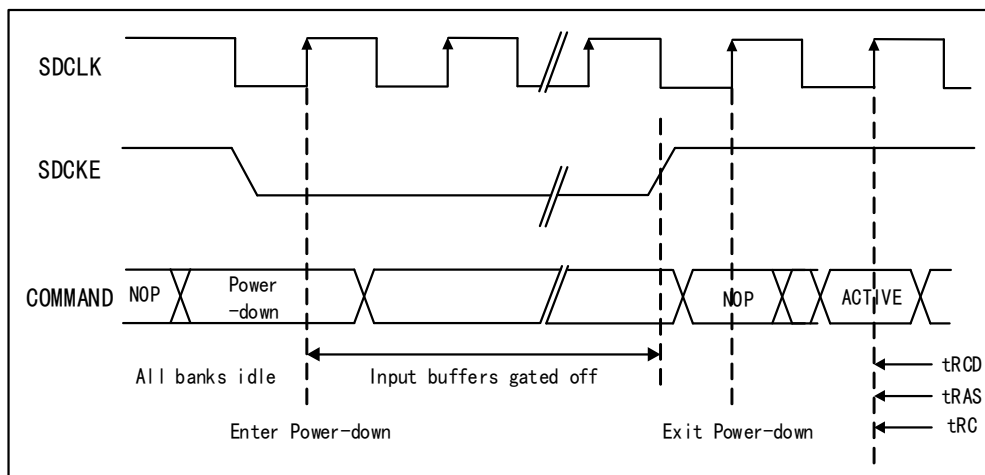
图 40-28 自刷新模式



掉电模式

通过将 MODE 位置为“110”并配置 R32_FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）来选择该模式。

图 40-29 掉电模式



如果写数据 FIFO 非空，则所有数据都将在掉电模式激活前发送到存储器。

SDRAM 控制器会在有 SDRAM 设备被选定后立即退出掉电模式。存储器访问完成后，选定的 SDRAM 设备将保持正常模式。

在掉电模式期间，将禁用 SDRAM 设备的所有输入/输出缓冲区，只有保持低电平 of SDCKE 除外。

SDRAM 设备保持掉电模式的时间不会长于刷新周期，并且自身无法执行自刷新循环。因此，SDRAM

控制器通过以下操作执行刷新：

- 1) 退出掉电模式并将 SDCKE 驱动为高电平
- 2) 生成 PALL 命令（前提是在掉电模式下存在激活行）
- 3) 生成自动刷新命令
- 4) 再次将 SDCKE 驱动为低电平以返回掉电模式。

要退出掉电模式，必须将 MODE 位置为“000”（正常模式）并配置 R32_FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）。

40.7.5 SDRAM 控制寄存器

表 40-44 SDRAM 相关寄存器列表

名称	访问地址	描述	复位值
R32_FMC_SDCR1	0x40025540	SDRAM 控制寄存器 1	0x000002D0
R32_FMC_SDCR2	0x40025544	SDRAM 控制寄存器 2	0x000002D0
R32_FMC_SDTR1	0x40025548	SDRAM 时序寄存器 1	0x0FFFFFFF
R32_FMC_SDTR2	0x4002554C	SDRAM 时序寄存器 2	0x0FFFFFFF
R32_FMC_SDCMR	0x40025550	SDRAM 命令模式寄存器	0x00000000
R32_FMC_SDRTR	0x40025554	SDRAM 刷新定时器寄存器	0x00000000
R32_FMC_SDSR	0x40025558	SDRAM 状态寄存器	0x00000000
R32_SDRAM_MISC	0x40025580	SDRAM 自定义寄存器	0x00000000

40.7.5.1 SDRAM 控制寄存器 x（R32_FMC_SDCRx）（x=1/2）

偏移地址：0x140+4*（x-1）

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
Reser ved	RPIPE[1:0]		RBU RST	SDCLK[1:0]		WP	CAS[1:0]		NB	MWID[1:0]		NR[1:0]		NC[1:0]	

位	名称	访问	描述	复位值
[31:15]	Reserved	RO	保留。	0
[14:13]	RPIPE[1:0]	RW	读管道： 00：0 个 HCLK 时钟周期延迟； 01：1 个 HCLK 时钟周期延迟； 10：2 个 HCLK 时钟周期延迟； 11：保留。 注：R32_FMC_SDCR2 寄存器中的相应位为无关位。	0
12	RBURST	RW	突发读： 0：不将单次读请求作为突发请求管理； 1：始终将单次读请求作为突发请求管理。 注：R32_FMC_SDCR2 寄存器中的相应位为无关位。	0
[11:10]	SDCLK[1:0]	RW	SDRAM 时钟配置： 00：禁止 SDCLK 时钟； 01：SDCLK 周期=HCLK 周期； 10：SDCLK 周期=2*HCLK 周期；	11b

			11: SDCLK 周期=3*HCLK 周期; 注: R32_FMC_SDCR2 寄存器中的相应位为无关位。	
9	WP	RW	写保护: 1: 忽略写访问; 0: 允许写访问。	0
[8:7]	CAS[1:0]	RW	CAS 延迟: 00: 保留; 01: 1 个周期; 10: 2 个周期; 11: 3 个周期。	10b
6	NB	RW	内部存储区域数量: 1: 4 个内部存储区域; 0: 2 个内部存储区域。	0
[5:4]	MWID[1:0]	RW	存储器数据总线宽度: 00: 8 位; 01: 16 位; 10: 32 位; 11: 保留。	10b
[3:2]	NR[1:0]	RW	行地址位数: 00: 11 位; 01: 12 位; 10: 13 位; 11: 保留。	0
[1:0]	NC[1:0]	RW	列地址位数: 00: 8 位; 01: 9 位; 10: 10 位; 11: 11 位。	0

注: 修改 RBURST 或 RPIPE 设置或者禁止 SDCLK 时钟之前, 用户必须先发送 PALL 命令以确保先完成正在进行的操作。

40.7.5.2 SDRAM 时序寄存器 x (R32_FMC_SDTRx) (x=1/2)

偏移地址: $0x148 + 4 * (x - 1)$

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

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:24]	TRCD[3:0]	RW	行到列延迟: 0000: 1 个周期; 0001: 2 个周期;	1111b

			1111: 16 个周期。	
[23:20]	TRP[3:0]	RW	行预充电延迟: 仅在 R32_FMC_SDTR1 寄存器中配置 TRP 时序。如果使用了两个 SDRAM 设备, 则必须使用最慢设备的时序配置 TRP。 0000: 1 个周期; 0001: 2 个周期; 1111: 16 个周期。 注: R32_FMC_SDTR2 寄存器中的相应位为无关位。	1111b
[19:16]	TWR[3:0]	RW	恢复延迟: 0000: 1 个周期; 0001: 2 个周期; 1111: 16 个周期。 注: TWR 必须设置成与 SDRAM 数据手册中定义的写恢复时间 (t_{WR}) 相匹配, 保证: $TWR \geq TRAS - TRCD$, 并且 $TWR \geq TRC - TRCD - TRP$ 例如: $TRAS=4$ 个周期, $TRCD=2$ 个周期。因此, $TWR \geq 2$ 个周期。TWR 必须设置为 0x1。 如果使用了两个 SDRAM 设备, 则必须为 R32_FMC_SDTR1 和 R32_FMC_SDTR2 配置相同的 TWR 时序 (对应于较慢的 SDRAM 设备)。	1111b
[15:12]	TRC[3:0]	RW	行循环延迟: 仅在 R32_FMC_SDTR1 寄存器中配置 TRC 时序。如果使用了两个 SDRAM 设备, 则必须使用最慢设备的时序配置 TRC。 0000: 1 个周期; 0001: 2 个周期; 1111: 16 个周期。 注: TRC 必须与 SDRAM 设备数据手册中定义的 TRC 和 TRFC (自动刷新周期) 时序相匹配。 注: R32_FMC_SDTR2 寄存器中的相应位为无关位。	1111b
[11:8]	TRAS[3:0]	RW	自刷新时间: 0000: 1 个周期; 0001: 2 个周期; 1111: 16 个周期。	1111b
[7:4]	TXSR[3:0]	RW	退出自刷新延迟: 0000: 1 个周期; 0001: 2 个周期; 1111: 16 个周期。 注: 如果使用了两个 SDRAM 设备, 则必须为 R32_FMC_SDTR1 和 R32_FMC_SDTR2 配置相同的	1111b

			TXSR 时序（对应于较慢的 SDRAM 设备）。	
[3:0]	TMRD[3:0]	RW	加载模式寄存器到激活： 0000：1 个周期； 0001：2 个周期； 1111：16 个周期。	1111b

注：如果连接了两个 SDRAM 设备，对于命令模式寄存器同时访问这两个设备（加载模式寄存器命令）的情况，将按 R32_FMC_SDTR1 寄存器中为存储区域 1 配置的时序参数（TMRD 和 TRAS 时序）发出访问命令。仅在 R32_FMC_SDTR1 寄存器中配置 TRP 和 TRC 时序。如果使用了两个 SDRAM 设备，则必须使用最慢设备的时序配置 TRP 和 TRC 时序。

40.7.5.3 SDRAM 命令模式寄存器（R32_FMC_SDCMR）

偏移地址：0x150

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											MRD[12:7]				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRD[6:0]						NRFS[3:0]				CTB1	CTB2	MODE[2:0]			

位	名称	访问	描述	复位值
[31:22]	Reserved	R0	保留。	0
[21:9]	MRD[12:0]	RW	模式寄存器定义： 此 13 位字段定义 SDRAM 模式寄存器内容。可通过加载模式寄存器命令对模式寄存器进行编程。	0
[8:5]	NRFS[3:0]	RW	自动刷新次数： 这些位定义 MODE=“011”时所发出的连续自刷新命令个数。 0000：1 个自刷新周期； 0001：2 个自刷新周期； 1110：15 个自刷新周期； 1111：16 个自刷新周期。	0
4	CTB1	RW	命令目标存储区域 1： 1：命令已发送到 SDRAM 存储区域 1； 0：命令未发送到 SDRAM 存储区域 1。	0
3	CTB2	RW	命令目标存储区域 2： 1：命令已发送到 SDRAM 存储区域 2。 0：命令未发送到 SDRAM 存储区域 2；	0
[2:0]	MODE[2:0]	RW	命令模式： 000：正常模式； 001：时钟配置使能； 010：PALL（“预充电所有存储区域”）命令； 011：自动刷新命令； 100：加载模式寄存器； 101：自刷新命令；	0

		110: 掉电命令; 111: 保留。 注: 命令发出后, 至少一个命令目标存储区域位 (CTB1 或 CTB2) 必须置 1, 否则该命令将被忽略。 注: 如果使用两个 SDRAM 存储区域, 则必须通过将 CTB1 和 CTB2 位置 1 来向两个器件同时发送自刷新和 PALL 命令, 否则该命令将被忽略。 注: 如果只使用一个 SDRAM 存储区域, 并通过将其关联的 CTB 位置 1 来发出命令, 则未使用存储区域的另一个 CTB 位必须保持为 0。	

40.7.5.4 SDRAM 刷新定时器寄存器 (R32_FMC_SDRTR)

偏移地址: 0x154

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	REIE	COUNT[12:0]												CRE	

位	名称	访问	描述	复位值
[31:15]	Reserved	R0	保留。	0
14	REIE	RW	RES 中断使能: 1: RE=1 时生成中断; 0: 禁止中断。	0
[13:1]	COUNT[12:0]	RW	刷新定时器计数: 该 13 位字段定义 SDRAM 设备的刷新速率, 以存储器时钟周期数表示。该字段必须设置为至少 41 个 SDRAM 时钟周期。 刷新速率= (COUNT+1) *SDRAM 频率时钟; COUNT= (SDRAM 刷新周期/行数) -20。	0
0	CRE	R0	清除刷新错误标志: 该位用于清除状态寄存器中的刷新错误标志 (RE)。 1: 清除刷新错误标志; 0: 无操作。	0

注: 所编程的 COUNT 值不可等于以下时序之和: TWR+TRP+TRC+TRCD+4 个存储器时钟周期。

40.7.5.5 SDRAM 状态寄存器 (R32_FMC_SDSR)

偏移地址: 0x158

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										BUSY	MODES2[1:0]	MODES1[1:0]	RE		

位	名称	访问	描述	复位值
[31:6]	Reserved	R0	保留。	0
5	BUSY	R0	繁忙状态： 1：SDRAM 控制器未准备好接受新请求； 0：SDRAM 控制器已准备好接受新请求。	0
[4:3]	MODES2[1:0]	R0	存储区域 2 的状态模式： 00：正常模式； 01：自刷新模式； 10：掉电模式。	0
[2:1]	MODES1[1:0]	R0	存储区域1的状态模式： 00：正常模式； 01：自刷新模式； 10：掉电模式。	0
0	RE	R0	刷新错误标志： 1：检测到刷新错误； 0：未检测到刷新错误。 在 REIE=1 且 RE=1 时会生成中断	0

40.7.5.6 SDRAM 自定义寄存器 (R32_SDRAM_MISC)

偏移地址：0x180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														En_Bank2	En_Bank1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Enhance_read_mode	Reserved						Phase_Sel				NRFS_CNT				

位	名称	访问	描述	复位值
[31:18]	Reserved	R0	保留。	0
17	En_Bank2	RW	使能 SDRAM2。	0
16	En_Bank1	RW	使能 SDRAM1。	0
15	Enhance_read_mode	RW	1：读增强/预取模式； 0：正常读模式。 控制器会预读取下面 8、16、32 个地址的数据（根据 SDRAM 的位宽而定，要取够 256bit 的数据），当地址连续时可以提高读取的速度。 注：该位开启时需关闭 RBURST 位。	0
[14:8]	Reserved	RW	保留。	0
[7:4]	Phase_Sel[3:0]	RW	输出 SDR_CLK 的相位偏移： Bit[7]：输出相位反相； Bit[6:4]：相位依次增加 0.4ns 左右。	0
[3:0]	NRFS_CNT[3:0]	RW	定义控制器每次自刷新的次数：	0

			0000: 1 个周期; 0001: 2 个周期; 1110: 15 个周期; 1111: 16 个周期。	

第 41 章 加密模块（ECDC）

系统内置了分组密码算法模块，支持 AES 和 SM4 两种分组密码算法以及电子密码本（ECB）和计数器（CTR）模式。模块以 128 位数据大小为基本单位完成一次加解密过程，提供了对内存中数据以 DMA 方式加解密以及 SFR 寄存器单次加解密模式。

41.1 主要特征

- SM4算法128位密钥的ECB模式和CTR模式
- AES算法128/192/256位密钥的ECB模式和CTR模式
- 支持软件写SFR的方式直接加密单个128位数据
- 支持DMA的方式（存储器到存储器）加解密软件指定长度的数据块

41.2 AES/SM4 算法

AES（Advanced Encryption Standard）算法是一种区块加密法，采用对称分组密码体制，是对称密钥加密中最流行的算法之一。SM4 分组密码算法一般是用于无线局域网和可信计算机的专用分组密码算法，同时也可用于其它环境下的数据加密保护。

在数据加解密过程中，需要载入密钥。对于 AES 算法，根据设置密钥长度为 128/192/256 位，使用时，需分别将用户密钥扩展为 11×128 / 13×128 / 15×128 位的扩展密钥。而 SM4 算法，将 128 位的用户密钥扩展成为 32×32 位的扩展密钥。这些扩展密钥被保存在内部寄存器中，以方便在加解密时使用。

41.3 ECB 与 CTR 模式

AES/SM4 支持两种模式，电子密码本（ECB）模式和计数器（CTR）模式，其中 CTR 模式的安全性能要高于 ECB 模式。如图 41-1 和 41-2 所示的工作框图。

图 41-1 ECB 模式加解密过程

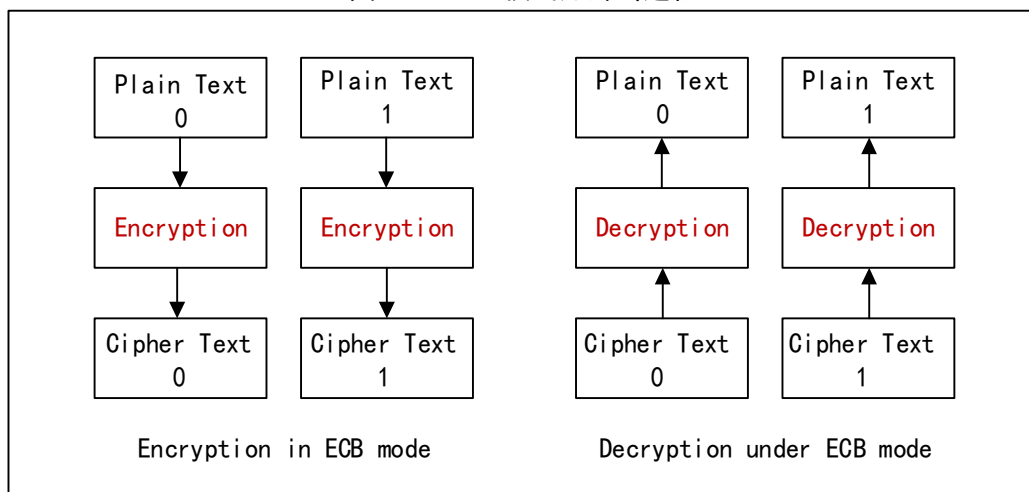
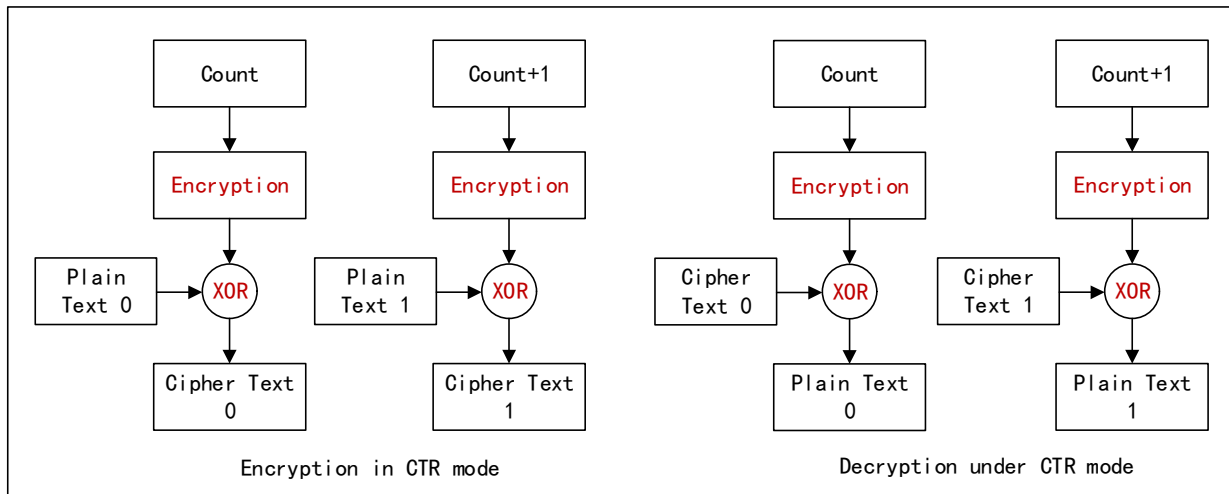


图 41-2 CTR 模式加解密过程



在 ECB 模式下，明文（plain text）与密文（cipher text）是一一对应的，加密之后的明文直接作为密文；而在 CTR 模式下，需要预先载入 128 位的计数值，将计数值加密，加密之后的计数值与明文异或作为密文。值得注意的是，CTR 解密模式下，也是只对计数值进行加密，而非解密。

41.4 模块应用

41.4.1 加解密模块初始化

- 时钟配置：设置 R32_ECEC_CTRL 寄存器的 RB_ECDC_CLKDIV_MASK 位，配置模块运行的时钟。如果不使用此模块可以将其设置为 0，关闭模块降低功耗。
- 数据大小端：设置 R32_ECEC_CTRL 寄存器的 RB_ECDC_DAT_MOD 位，可选择加解密数据的大小端存储方式。按 128bits（16 字节）一包大小端存储。
- 密钥扩展：加解密过程中使用的不是用户密钥而是扩展密钥。当设置完用户密钥长度和内容后，需要通知硬件执行密钥扩展动作，当密钥扩展后会被保存到模块内部寄存器。之后才可以正常进行数据加解密工作。设置 R32_ECEC_CTRL 寄存器 RB_ECDC_WRSRAM_EN、RB_ECDC_ALGRM_MOD、RB_ECDC_CIPHER_MOD、RB_ECDC_KLEN_MASK，使能 SRAM 数据加解密、选择加解密算法、分组密码模式、密钥长度；然后配置用户密钥寄存器组 R32_ECDC_KEY，填入用户密钥内容。将 R32_ECEC_CTRL 寄存器的 RB_ECDC_KEYEX_EN 置 1 再清 0 后，硬件开始执行密钥扩展功能。
- 计数器初值：如果使用 CTR 模式进行加解密，还需一个初始的计数器值。设置 CTR 模式计数值寄存器组 R32_ECDC_IV 填入 128 位计数器值，每次成功完成一次数据加解密（128bit 数据）模块内部的计数器值将加 1。ECB 模式无需此计数器值。

41.4.2 加解密配置

模块的加解密数据流包含存储器到存储器、单次 128bit 数据加解密两种方式，具体可以参考表 41-1 的配置。

表 41-1 不同数据流模式配置

加解密数据流	RB_ECDC_WRSRAM_EN	RB_ECDC_NORMAL_EN	RB_ECDC_MODE_SEL	其他
SRAM 数据加密	1	1	0	需配置寄存器 R32_ECDC_SRC_ADDR, R32_ECDC_DST_ADDR 和 R32_ECDC_SRAM_LEN
SRAM 数据解密	1	1	1	

128bits 数据单次加密	-	1	0	写 R32_ECDC_SGSD 寄存器, 填入原始数据(明文或密文) 读 R32_ECDC_SGRT 寄存器, 读出加解密后的结果数据。
128bits 数据单次解密	-	1	1	

存储器到存储器之间的数据转换开始由 R32_ECDC_SRAM_LEN 长度寄存器写入非 0 数据启动, 通过查询 R32_ECDC_INT_FG 寄存器的 RB_ECDC_IF_WRSRAM 位, 如果置 1 表示存储器到存储器数据转换完成, 清除 RB_ECDC_IF_WRSRAM 位状态。如果开启了 R32_ECDC_CTRL 寄存器的 RB_ECDC_IE_WRSRAM 功能, 在 RB_ECDC_IF_WRSRAM 标志置位的同时将触发中断服务。

单次加解密方式必须配置为大端方式 (RB_ECDC_DAT_MOD=1) 执行。128bits 数据单次加解密转换开始向 R32_ECDC_SGSD 寄存器的最高 32 位写入操作启动, 通过查询 R32_ECDC_INT_FG 寄存器的 RB_ECDC_IF_SINGLE 位, 如果置 1 表示单次 128bits 数据转换完成, 清除 RB_ECDC_IF_SINGLE 位状态。如果开启了 R32_ECEC_CTRL 寄存器的 RB_ECDC_IE_SINGLE 功能, 在 RB_ECDC_IF_SINGLE 标志置位的同时将触发中断服务。

41.5 寄存器描述

表 41-2 ECDC 模块寄存器列表

名称	访问地址	描述	复位值
R32_ECDC_CTRL	0x40016C00	ECDC 控制寄存器	0x00000020
R32_ECDC_INT_FG	0x40016C04	ECDC 中断标志寄存器	0x00000000
R32_ECDC_KEY_255T224	0x40016C08	用户密钥寄存器 7	0xFFFFFFFF
R32_ECDC_KEY_223T192	0x40016C0C	用户密钥寄存器 6	0xFFFFFFFF
R32_ECDC_KEY_191T160	0x40016C10	用户密钥寄存器 5	0xFFFFFFFF
R32_ECDC_KEY_159T128	0x40016C14	用户密钥寄存器 4	0xFFFFFFFF
R32_ECDC_KEY_127T96	0x40016C18	用户密钥寄存器 3	0xFFFFFFFF
R32_ECDC_KEY_95T64	0x40016C1C	用户密钥寄存器 2	0xFFFFFFFF
R32_ECDC_KEY_63T32	0x40016C20	用户密钥寄存器 1	0xFFFFFFFF
R32_ECDC_KEY_31T0	0x40016C24	用户密钥寄存器 0	0xFFFFFFFF
R32_ECDC_IV_127T96	0x40016C28	CTR 模式计数值寄存器 3	0xFFFFFFFF
R32_ECDC_IV_95T64	0x40016C2C	CTR 模式计数值寄存器 2	0xFFFFFFFF
R32_ECDC_IV_63T32	0x40016C30	CTR 模式计数值寄存器 1	0xFFFFFFFF
R32_ECDC_IV_31T0	0x40016C34	CTR 模式计数值寄存器 0	0xFFFFFFFF
R32_ECDC_SGSD_127T96	0x40016C40	单次加解密原始数据 3	0x00000000
R32_ECDC_SGSD_95T64	0x40016C44	单次加解密原始数据 2	0x00000000
R32_ECDC_SGSD_63T32	0x40016C48	单次加解密原始数据 1	0x00000000
R32_ECDC_SGSD_31T0	0x40016C4C	单次加解密原始数据 0	0x00000000
R32_ECDC_SGRT_127T96	0x40016C50	单次加解密结果 3	0x00000000
R32_ECDC_SGRT_95T64	0x40016C54	单次加解密结果 2	0x00000000
R32_ECDC_SGRT_63T32	0x40016C58	单次加解密结果 1	0x00000000
R32_ECDC_SGRT_31T0	0x40016C5C	单次加解密结果 0	0x00000000
R32_ECDC_SRC_ADDR	0x40016C60	加解密 SRAM 区域源地址	0x00000000
R32_ECDC_DST_ADDR	0x40016C64	加解密 SRAM 目的地址	0x00000000

R32_ECDC_SRAM_LEN	0x40016C68	加解密 SRAM 大小	0x00000000

41.5.1 ECDC 控制寄存器 (R32_ECDC_CTRL)

偏移地址: 0x00

位	名称	访问	描述	复位值
[31:26]	Reserved	R0	保留。	0
25	RB_ECDC_AES/SM4_CLOCK_EN	RW	1: 开启加解密硬件时钟; 0: 关闭加解密硬件时钟; 注: 使用加解密时, 此位置 1; 不使用时, 此位置 0 且可节省功耗。	0
24	RB_ECDC_CLOCK_SELECT	RW	1: 加解密工作在 pllclk 下 (最大支持 400M); 0: 工作频率为 [6:4] 所指的分频时钟下。 注: ECDC 时钟须大于等于 HCLK 时钟。	0
[23:19]	Reserved	R0	保留。	0
18	RB_ECDC_IE_WRSRAM	RW	存储器到存储器进行加解密完成中断使能。 1: 使能相应中断; 0: 禁止相应中断。	0
17	RB_ECDC_IE_SINGLE	RW	单次加解密完成中断使能。针对 128bit 寄存器转换完成。 1: 使能相应中断; 0: 禁止相应中断。	0
[16:14]	Reserved	R0	保留。	0
13	RB_ECDC_DAT_MOD	WO	加解密数据大小端方式选择。 1: 大端方式; 0: 小端方式。 注: 单次 128bits 加解密下必须配置为大端方式。	0
12	Reserved	R0	保留。	0
[11:10]	RB_ECDC_KLEN_MASK	RW	密钥长度设置。 00: 128 位; 01: 192 位; 10: 256 位; 11: 保留。	0
9	RB_ECDC_CIPHER_MOD	RW	分组密码模式选择。 1: CTR 模式; 0: ECB 模式。	0
8	RB_ECDC_ALGRM_MOD	RW	加解密算法模式选择。 1: AES; 0: SM4。	0
7	RB_ECDC_WRSRAM_EN	RW	使能 SRAM 数据加解密。需要控制位 RB_ECDC_KEYEX_EN 配合使用。 1: 开启; 0: 关闭。	0
[6:4]	RB_ECDC_CLKDIV_MASK	RW	加解密时钟分频系数。 计算: $EDclk = PLLCLK / ED\_CLK\_PRE$ 。	010b

			最小值 2，写 1 等效关闭 ECDC 模块运算时钟，写 0 关闭 ECDC 模块运算时钟。 注：ECDC 时钟须大于等于 HCLK 时钟。	
3	RB_ECDC_MODE_SEL	RW	加解密模式选择。 1：解密模式； 0：加密模式。	0
2	Reserved	RW	保留	0
1	RB_ECDC_NORMAL_EN	RW	使能加解密工作在正常加解密模式，软件在检测到密钥扩展完成后，开始加解密前将此位置 1。 1：加解密使能； 0：不工作。	0
0	RB_ECDC_KEYEX_EN	RW	密钥扩展功能使能控制位，高电平脉冲启动。在配置了密钥长度后，需要此位置 1，并等待密钥扩展完成才能进行加解密过程。 注：应用代码驱动需要将其置高再置低。	0

41.5.2 ECDC 中断标志寄存器 (R32_ECDC_INT_FG)

偏移地址：0x04

位	名称	访问	描述	复位值
[31:19]	Reserved	R0	保留。	0
18	RB_ECDC_IF_WRSRAM	RW1Z	存储器到存储器加解密完成中断标志，写 1 清 0。 1：加解密完成事件； 0：无事件。	0
17	RB_ECDC_IF_SINGLE	RW1Z	单次加解密完成中断标志，写 1 清 0。 1：单次加解密完成事件； 0：无事件。	0
[16:0]	Reserved	RW	保留	0

41.5.3 用户密钥寄存器组 (R32_ECDC_KEY)

偏移地址：0x08、0x0C、0x10、0x14、0x18、0x1C、0x20、0x24

位	名称	访问	描述	复位值
[31:0]	R32_ECDC_KEY_255T224	RW	用户密钥 224-256 位。	X
[31:0]	R32_ECDC_KEY_223T192	RW	用户密钥 192-223 位。	X
[31:0]	R32_ECDC_KEY_191T160	RW	用户密钥 160-191 位。	X
[31:0]	R32_ECDC_KEY_159T128	RW	用户密钥 128-159 位。	X
[31:0]	R32_ECDC_KEY_127T96	RW	用户密钥 96-127 位。	X
[31:0]	R32_ECDC_KEY_95T64	RW	用户密钥 64-95 位。	X
[31:0]	R32_ECDC_KEY_63T32	RW	用户密钥 32-63 位。	X
[31:0]	R32_ECDC_KEY_31T0	RW	用户密钥 0-31 位。	X

41.5.4 CTR 模式计数值寄存器组 (R32_ECDC_IV)

偏移地址：0x28、0x2C、0x30、0x34

位	名称	访问	描述	复位值
[31:0]	R32_ECDC_IV_127T96	RW	计数值 96-127 位。	X

[31:0]	R32_ECDC_IV_95T64	RW	计数值 64-95 位。	X
[31:0]	R32_ECDC_IV_63T32	RW	计数值 32-63 位。	X
[31:0]	R32_ECDC_IV_31T0	RW	计数值 0-31 位。	X

41.5.5 单次加解密原始数据寄存器组 (R32_ECDC_SGSD)

偏移地址：0x40、0x44、0x48、0x4C

位	名称	访问	描述	复位值
[31:0]	R32_ECDC_SGSD_127T96	RW	原始数据 96-127 位。	0
[31:0]	R32_ECDC_SGSD_95T64	RW	原始数据 64-95 位。	0
[31:0]	R32_ECDC_SGSD_63T32	RW	原始数据 32-63 位。	0
[31:0]	R32_ECDC_SGSD_31T0	RW	原始数据 0-31 位。	0

注：内部加解密固定 128bit 大小进行。当执行完 R32_ECDC_SGSD_31T0 寄存器的写操作，硬件自动开始进行单次加解密转换。通过 RB_ECDC_IF_SINGLE 表示判断转换是否完成。

41.5.6 单次加解密结果寄存器组 (R32_ECDC_SGRT)

偏移地址：0x50、0x54、0x58、0x5C

位	名称	访问	描述	复位值
[31:0]	R32_ECDC_SGRT_127T96	RW	数据 96-127 位。	0
[31:0]	R32_ECDC_SGRT_95T64	RW	数据 64-95 位。	0
[31:0]	R32_ECDC_SGRT_63T32	RW	数据 32-63 位。	0
[31:0]	R32_ECDC_SGRT_31T0	RW	数据 0-31 位。	0

注：加解密结果数据固定大端方式存储，128bits 结构。

41.5.7 加解密 SRAM 区域源地址 (R32_ECDC_SRC_ADDR)

偏移地址：0x60

位	名称	访问	描述	复位值
[31:0]	ECDC_SRAM_SRC_ADDR	RW	加解密 SRAM 数据的源起始地址。	0

注：此 DMA 寻址在 SRAM 区域。

41.5.8 加解密 SRAM 目的地址 (R32_ECDC_DST_ADDR)

偏移地址：0x64

位	名称	访问	描述	复位值
[31:0]	ECDC_SRAM_DST_ADDR	RW	加解密 SRAM 数据的目的起始地址，单位为 128bit。当写入非零值时，将启动 SRAM 数据的加解密过程。（32 字节对齐）	0

41.5.9 加解密 SRAM 大小 (R32_ECDC_SRAM_LEN)

偏移地址：0x68

位	名称	访问	描述	复位值
[31:13]	Reserved	RO	保留。	0
[12:0]	ECDC_SRAM_LEN	RW	加解密 SRAM 数据长度，单位为 128bit。当写入非零值时，将启动 SRAM 数据的加解密过程。	0

第 42 章 数字滤波器，用于 $\Sigma \Delta$ 调制器 (DFSDM)

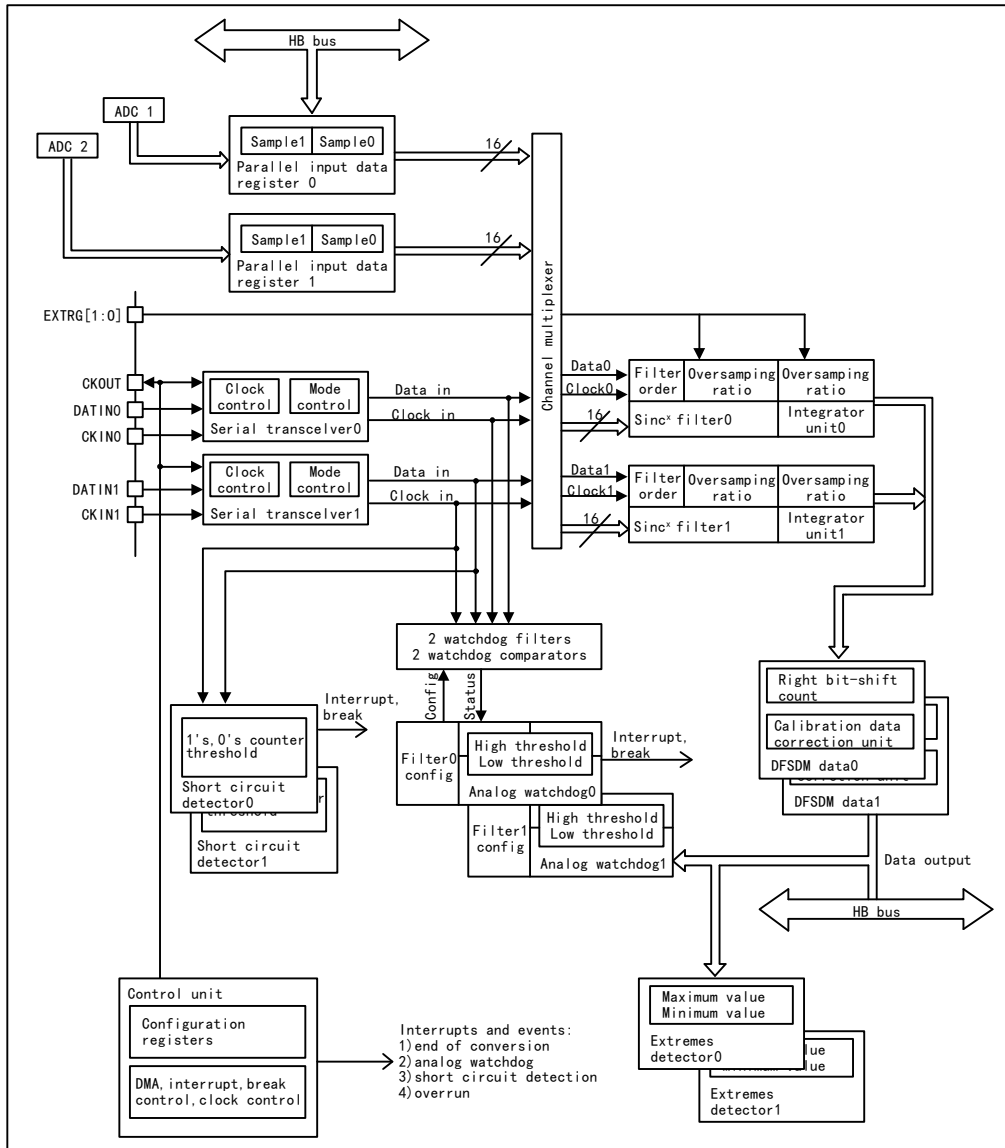
DFSDM 是一种专用于将外部 $\Sigma \Delta$ 调制器连接到 MCU 的高性能模块。它包含 2 个外部数字串行接口和 2 个数字滤波器，具有灵活的 $\Sigma \Delta$ 数字处理选项，能提供高达 24 位的 ADC 最终分辨率。DFSDM 还特有可选择来自内部 ADC 外设或者设备存储器的并行数据流输入。

42.1 主要特征

- 提供 2 个复用输入数字串行通道
- 2 个内部数字并行通道支持可选的输入
- 支持可调节的数字信号处理
- 支持高达 24 位的输出数据分辨率
- 支持有符号的数据格式
- 支持自动的数据偏移校正（偏移值由用户存储在寄存器里）
- 包含两个转换模式：单个转换模式和连续转换模式
- 可通过软件触发、内部定时器、外部事件、使用第一个 DFSDM 滤波器同步开启转换
- 支持模拟看门狗
- 内置短路检测器来检测饱和的模拟输入值（下限和上限）
- 支持发生模拟看门狗事件和短路监测事件时产生断路
- 内置极值检测器
- 支持中断和 DMA

42.2 概述

图 42-1 DFSDM 的结构框图



42.3 功能描述

42.3.1 DFSDM 复位和时钟

DFSDM 开关控制

通过将 R32_DFSDM_CH0CFGR1 寄存器中的 DFSDMEN 位设置为 1, 实现 DFSDM 接口的全局使能。DFSDM 接口全局使能后, 所有配置的输入通道 y ($y=0/1$) 以及数字滤波器 DFSDM_FLT x ($x=0/1$) 将在其各自的使能位 (R32_DFSDM_CH y CFGR1 中的通道使能位 CHEN 和 R32_DFSDM_FLT x CR1 中的 DFSDM_FLT x 使能位 DFEN) 置 1 时开始工作。

数字滤波器 DFSDM_FLT x ($x=0/1$) 通过将 R32_DFSDM_FLT x CR1 寄存器的 DFEN 置 1 来使能。使能 DFSDM_FLT x (DFEN=1) 后, Sinc x 数字滤波器单元和积分器单元会重新初始化。

通过将 DFEN 清零, 正在进行的所有转换都会立即停止, DFSDM_FLT x 会置于停止模式。除 R32_DFSDM_FLT x AWSR 和 R32_DFSDM_FLT x ISR (均被复位) 外, 所有寄存器设置保持不变。

通过将 R32_DFSDM_CH y CFGR1 寄存器的 CHEN 置 1 使能通道 y ($y=0/1$)。通道使能后, 会从外部 $\Sigma \Delta$ 调制器或并行内部数据源 (ADC 或存储器的 CPU/DMA 写操作) 接收串行数据。

在停止系统时钟，使器件进入停止模式之前，必须全局禁止 DFSDM（通过在 R32_DFSDM_CH0CFGR1 中设置 DFSDMEN=0）。

DFSDM 时钟

内部 DFSDM 时钟 f_{DFSDMCLK} 用于驱动通道收发器、数字处理模块（包括数字滤波器、积分器）以及后续附加模块（如模拟看门狗、短路检测器、极值检测器、控制模块），该时钟由 RCC 模块生成，源自外设时钟 HCLK 或音频时钟。DFSDM 时钟在停止模式下自动停止（对于所有 DFSDM_FLT $_x$ ， $x=0/1$ ，DFEN=0 时）。

DFSDM 串行通道收发器可以接收外部串行时钟，以采样外部串行数据流。如果使用标准 SPI 编码，内部 DFSDM 时钟必须至少比外部串行时钟快 4 倍；

DFSDM 支持提供一个外部输出时钟信号，用于驱动外部 $\Sigma \Delta$ 调制器的时钟输入。该输出时钟信号通过 CKOUT 引脚输出。输出时钟信号的频率必须符合器件手册中规定的范围，并且可以通过分频器从 DFSDM 时钟或音频时钟（参见 R32_DFSDM_CH0CFGR1 寄存器中的 CKOUTSRC 位）中获取。分频器的可编程范围为 2 至 256（R32_DFSDM_CH0CFGR1 寄存器中的 CKOUTDIV 位）。音频时钟通过 RCC 配置中的 SYSPLL_SEL[2:0] 字段选择 PLL_CLK，进而配置 PLLCLK 获得。

42.3.2 串行通道收发器

该外设具有 2 个复用串行数据通道，可以通过各个滤波器、模拟看门狗或者短路检测器将其选择用于转换。这些串行收发器接收来自外部 $\Sigma \Delta$ 调制器的数据流。数据流能以 SPI 格式进行发送（请参考 R32_DFSDM_CHyCFGR1 寄存器的 SITP 位）。

通过将 R32_DFSDM_CHyCFGR1 寄存器的 CHEN 置 1 可以使能通道。

通道输入选择

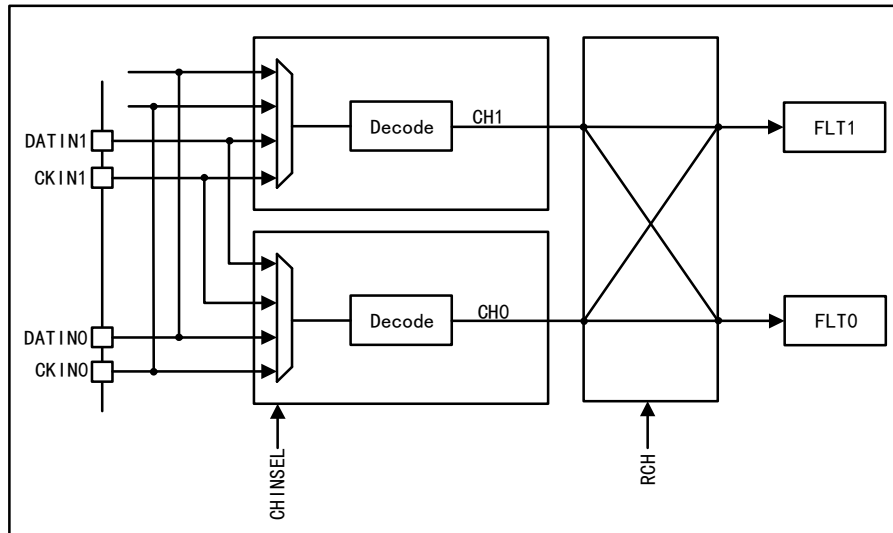
DATIN $_y$ 和 CKIN $_y$ 引脚的串行输入（数据和时钟信号）可从后续通道引脚重定向。串行输入通道的重定向功能是通过 R32_DFSDM_CHyCFGR1 寄存器中的 CHINSEL 位进行配置的。

通道重定向可用于从 PDM（脉冲密度调制）立体声麦克风中采集音频数据。PDM 立体声麦克风包含一个数据和一个时钟信号，其中数据信号为左右音频通道提供信息（时钟上升沿采样用于左通道，时钟下降沿采样用于右通道）。

PDM 麦克风输入串行通道的配置：

- PDM 麦克风信号（数据、时钟）将连接到 DFSDM 输入串行通道 y （DATIN $_y$ ，CKOUT）引脚
- 将通道 y 进行以下配置：CHINSEL=0（输入来自给定通道引脚：DATIN $_y$ 和 CKIN $_y$ ）
- 将通道 $(y-1)$ （模 2）进行以下配置：
 - CHINSEL=1（输入来自后续通道 $((y-1)+1)$ 引脚：DATIN $_y$ 和 CKIN $_y$ ）
- 通道 y ：SITP=0（上升沿选通数据）即通道 y 的左音频通道
- 通道 $(y-1)$ ：SITP=1（下降沿选通数据）即通道 $y-1$ 的右音频通道
- 两个 DFSDM 滤波器将分配至通道 y 和通道 $(y-1)$ （用于对 PDM 麦克风的左右通道进行滤波）

图 42-2 输入通道引脚重定向



时钟信号输出

可在 CKOUT 引脚上提供一个时钟信号，用于驱动外部 $\Sigma \Delta$ 调制器时钟输入。CKOUT 信号的频率通过预分频器（参考 R32_DFSDM_CH0CFGR1 寄存器 CKOUTDIV 位）从 DFSDM 时钟或音频时钟（参考 R32_DFSDM_CH0CFGR1 寄存器中的 CKOUTSRC 位）中获取。如果输出时钟停止，则 CKOUT 信号会被置于低电平状态（通过在 R32_DFSDM_CHyCFGR1 寄存器中设置 CKOUTDIV=0 或在 R32_DFSDM_CH0CFGR1 寄存器中设置 DFSDMEN=0，可以停止输出时钟）。在以下时间执行输出时钟停止：

- 当 CKOUTSRC=0 时，DFSDMEN 清零后 4 个系统时钟周期
- 当 CKOUTSRC=1 时，DFSDMEN 清零后 1 个系统时钟周期和 3 个音频时钟周期

在更改 CKOUTSRC 之前，软件必须等待 CKOUT 停止，以避免在 CKOUT 引脚上产生毛刺信号。输出时钟信号频率必须在 0~20MHz 范围内。

SPI 数据输入格式操作

在 SPI 格式下，数据流通过数据和时钟信号以串行格式进行传输。数据信号始终由 DATINy 引脚提供。时钟信号可以通过 CKINy 引脚从外部提供，也可以通过取自 CKOUT 信号源的信号从内部提供。

若选择外部时钟源（SPICKSEL[1:0]=0），则根据 R32_DFSDM_CHyCFGR1 寄存器中的 SITP 位设置，在（CKINy 引脚）时钟的上升沿或下降沿采样（DATINy 引脚）数据信号。

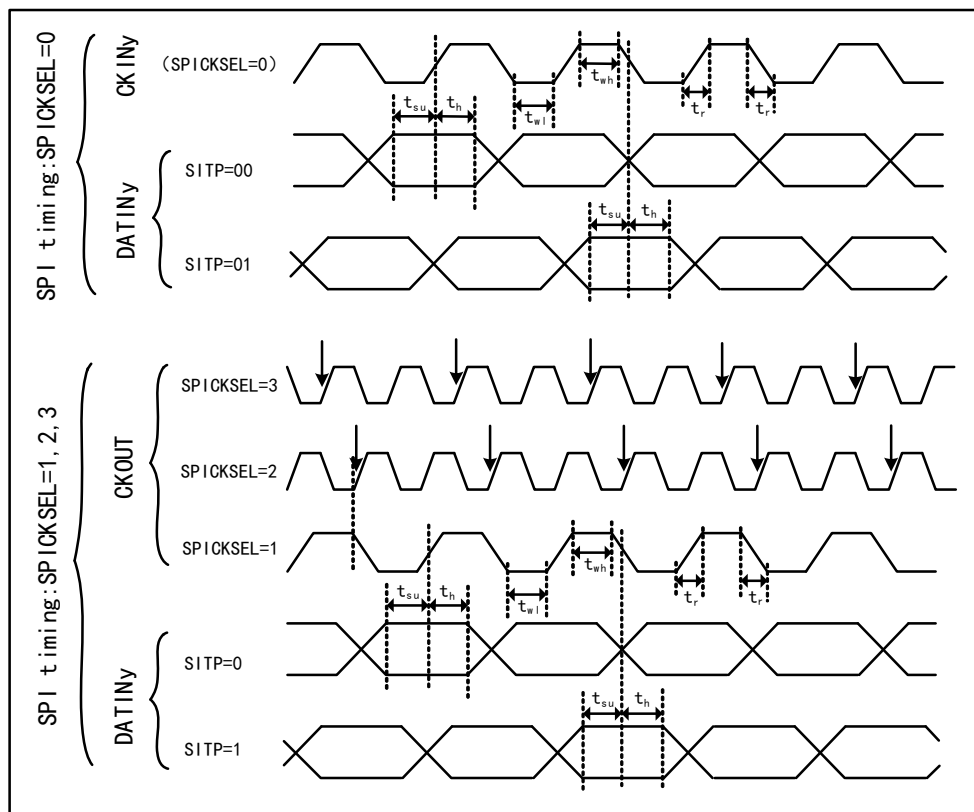
内部时钟源——参见 R32_DFSDM_CHyCFGR1 寄存器中的 SPICKSEL[1:0]：

- CKOUT 信号：
 - 用于连接外部 $\Sigma \Delta$ 调制器，调制器直接使用其时钟输入（来自 CKOUT）来生成输出串行通信时钟。
 - 采样点：上升沿/下降沿，具体取决于 SITP 设置。
- CKOUT/2 信号（在 CKOUT 上升沿生成）：
 - 用于连接外部 $\Sigma \Delta$ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入上升沿有效）。
 - 采样点：每第二个 CKOUT 下降沿。
- CKOUT/2 信号（在 CKOUT 下降沿生成）：
 - 用于连接外部 $\Sigma \Delta$ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入下降沿有效）。
 - 采样点：每第二个 CKOUT 上升沿。

注：只有在外部的 $\Sigma \Delta$ 调制器将 CKOUT 信号用作时钟输入（以实现同步时钟和数据操作）时，才能使用内部时钟源。使用内部时钟源可省去 CKINy 引脚连接（CKINy 引脚可用于其他用途）。如果采

用 SPI 编码，时钟源信号频率必须在 $0\sim 20\text{MHz}$ 范围内，且小于 $f_{\text{DFSDMCLK}}/4$ 。

图 42-3 通道收发器时序图



时钟缺失检测

该功能能够检测通道串行时钟输入的时钟缺失或存在情况，确保转换过程和错误报告的正常运作。通过 R32_DFSDM_CHyCFGR1 寄存器 CKABEN 位在各个输入通道 y 上使能或禁止时钟缺失检测。如果使能了给定通道的时钟缺失检测，则会对其连续执行时钟缺失检测。如果出现输入时钟错误，则时钟缺失标志会置 1 (CKABF[y]=1)，并会调用中断 (CKABIE=1 时)。时钟缺失标志清零 (R32_DFSDM_FLT0ICR 寄存器的 CLRCKABF) 后，会刷新时钟缺失标志。相应通道 y 禁止时，则时钟缺失状态位 CKABF[y] 还会由硬件置 1 (如果 CHEN[y]=0，则 CKABF[y] 保持在置 1 状态)。

一旦发生时钟缺失事件，数据转换 (和/或模拟看门狗和短路检测器) 将提供错误数据。在报告时钟缺失时，用户应处理此事件并丢弃相关数据。

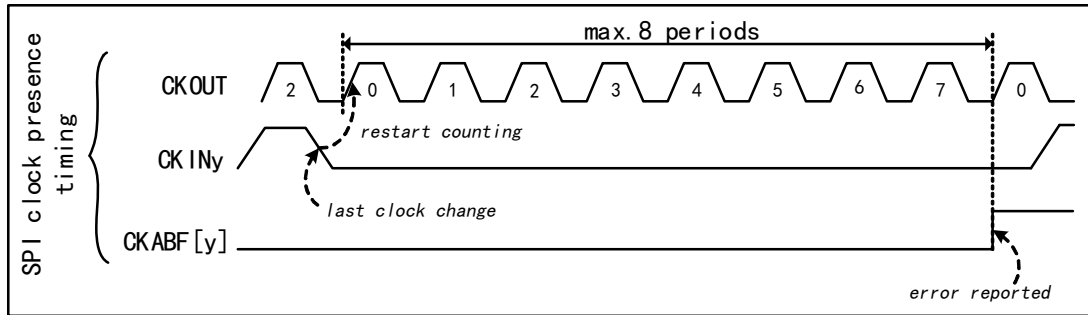
时钟缺失功能仅在将系统时钟用于 CKOUT 信号 (R32_DFSDM_CH0CFGR1 寄存器中的 CKOUTSRC=0) 时可用。

当收发器未同步时，时钟缺失标志将被置位，且无法通过 R32_DFSDM_FLT0ICR 寄存器中的 CLRCKABF[y] 位清零。与时钟缺失检测功能相关的软件操作序列应如下：

- 通过设置 CHEN=1 启用指定通道
- 尝试清零时钟缺失标志 (通过设置 CLRCKABF=1)，直到时钟缺失标志实际清零 (CKABF=0)。此时，收发器已同步 (信号时钟有效)，能够接收数据
- 使能时钟缺失功能 CKABEN=1 且相关中断 CKABIE=1，以检测 SPI 时钟是否丢失

若使用 SPI 数据格式，时钟缺失检测将基于外部输入时钟与输出时钟生成的 CKOUT 信号进行比较。输入通道中的外部输入时钟信号必须每经过 CKOUT 信号 (由 R32_DFSDM_CH0CFGR1 寄存器 CKOUTDIV 字段控制) 的 8 个信号周期更改一次。

图 42-4 SPI 时钟缺失时序图



如果出现输入时钟恢复错误，则时钟缺失标志会置 1（CKABF[y]=1），并会调用中断（CKABIE=1 时）。时钟缺失标志清零（通过 R32_DFSDM_FLT0ICR 寄存器的 CLRCKABF）后，会刷新时钟缺失标志。

外部串行时钟频率测量

通过测量通道的串行时钟输入频率，可以实时获取来自外部 $\Sigma \Delta$ 调制器的数据速率，这对于应用目的至关重要。

外部串行时钟输入频率可以通过在一个转换持续时间内对 DFSDM 时钟（ f_{DFSDMCLK} ）进行计数的定时器来测量。计数在转换触发（常规或注入）后的第一个输入数据时钟处开始，并在转换结束（转换结束标志置 1）前的最后一个输入数据时钟处结束。转换完成（JEOCF=1 或 REOCF=1）时，会在寄存器 R32_DFSDM_FLTxCNVTIMR 的计数器 CNVCNT[27:0] 中更新每次转换的时间（第一个串行采样和最后一个串行采样之间的时间）。然后用户可以根据数字滤波器设置（FORD、FOSR、IOSR、FAST）计算数据速率。只有在滤波器被旁路（FOSR=0，只有积分器有效，R32_DFSDM_FLTxCNVTIMR 寄存器 CNVCNT[27:0]=0）时，才会停止外部串行频率测量。

对于并行数据输入（参见第 42.3.4 节：并行数据输入），测量的频率代表了一次转换期间的平均输入数据速率。

注：转换被中断（例如，通过禁止/使能所选通道）时，还会在 CNVCNT[27:0] 中对中断时间进行计数。因此，为得到正确的转换时间结果，建议不要中断转换。建议在连续快速模式下测量时钟。

转换时间：

FAST=0 时的注入转换或常规转换（或者 FAST=1 时的首次转换）：

对于 Sinc^x 滤波器（x=1..5）：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * (2^{10\text{SR}} + F_{\text{ORD}} + 1) + 2] / f_{\text{CKIN}}$$

对于 FastSinc 滤波器：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * (F_{\text{OSR}} * (2^{10\text{SR}} + 5) + 2)] / f_{\text{CKIN}}$$

FAST=1 时的常规转换（首次转换除外）：

对于 Sinc^x 和 FastSinc 滤波器：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * 2^{10\text{SR}}] / f_{\text{CKIN}}$$

如果 FOSR=FOSR[9:0]+1（滤波器旁路，仅积分器有效），则：

$$t = 2^{10\text{SR}} / f_{\text{CKIN}} \quad (\dots \text{但是 } \text{CNVCNT}=0)$$

其中：

- 1) f_{CKIN} 为（给定通道 CKINy 引脚上的）通道输入时钟频率或（并行数据输入时的）输入数据速率；
- 2) F_{OSR} 为滤波器过采样率： $F_{\text{OSR}} = \text{FOSR}[9:0] + 1$ （参考 R32_DFSDM_FLTxFCR 寄存器）；
- 3) $2^{10\text{SR}}$ 为积分器过采样率（参考 R32_DFSDM_FLTxFCR 寄存器）；
- 4) F_{ORD} 为滤波器阶数： $F_{\text{ORD}} = \text{FORD}[2:0]$ （参考 R32_DFSDM_FLTxFCR 寄存器）。

通道偏移设置

每个通道都有自身的偏移设置（在寄存器中），该偏移值最终会从给定通道的各个转换结果（注

入或常规)中减去。在数据右移位后执行偏移校正。偏移值以 24 位有符号值的形式存储在 R32_DFSDM_CHyCFGR2 寄存器的 OFFSET[23:0]字段中。

数据右移位

要使结果与 24 位值对齐,每个通道均定义了一系列右移位,这些右移位将应用于给定通道的各个转换结果(注入或常规)。右移位数存储在 R32_DFSDM_CHyCFGR2 寄存器的 DTRBS[4:0]位中。

右移位将结果四舍五入为最接近的整数值。移位结果的符号保留,以便得到有效 24 位有符号格式的结果数据。

42.3.3 配置输入串行接口

必须为输入串行接口配置以下参数:

- 输出时钟预分频器。通过可编程预分频器,可以从 DFSDM 时钟生成输出时钟(范围为 2 至 256)。此设置由 R32_DFSDM_CH0CFGR1 寄存器中的 CKOUTDIV[7:0]位决定。
- 串行接口类型及输入时钟相位。选择 SPI 接口以及输入时钟的采样边沿。此选项由 R32_DFSDM_CHyCFGR1 寄存器中的 SITP 位指定。
- 输入时钟源选择。可选择来自 CKINy 引脚的外部源或来自 CKOUT 引脚的内部源。此选择由 R32_DFSDM_CHyCFGR1 寄存器中的 SPICKSEL[1:0]字段决定。
- 最终数据右移位设置。定义最终数据右移位数,以确保结果与 24 位值对齐。此配置由 R32_DFSDM_CHyCFGR2 寄存器中的 DTRBS[4:0]位定义。
- 通道偏移量配置。为特定串行通道设定模拟偏移量(即所连接外部 $\Sigma \Delta$ 调制器的偏移)。此值由 R32_DFSDM_CHyCFGR2 寄存器中的 OFFSET[23:0]位设定。
- 通道短路检测器与时钟缺失功能的启用。在 R32_DFSDM_CHyCFGR1 寄存器中,通过 SCDEN 位启用或禁用给定串行通道的短路检测器,通过 CKABEN 位启用或禁用时钟缺失检测。
- 模拟看门狗滤波器与短路检测器阈值配置。用于设定通道模拟看门狗滤波器参数和通道短路检测器参数。这些配置在 R32_DFSDM_CHyAWSCDR 寄存器中定义。

42.3.4 并行数据输入

每个输入通道均为 16 位并行数据输入提供一个寄存器(除了串行数据输入外)。每个 16 位并行输入只能源自内部数据源:

- 内部 ADC 结果(仅限 ADC1 和 ADC2)
- 直接 CPU/DMA 写操作

通过 R32_DFSDM_CHyCFGR1 寄存器的 DATMPX[1:0]字段选择是对给定通道使用串行还是并行数据输入。DATMPX[1:0]中还定义了并行数据源:内部 ADC 或直接由 CPU/DMA 进行的写操作。

每个通道均包含一个 32 位数据输入寄存器(R32_DFSDM_CHyDATINR),在该寄存器中可写入 16 位数据。数据采用 16 位有符号格式。这些数据还可用作接受 16 位并行数据的数字滤波器的输入。

如果选择串行数据输入(DATMPX[1:0]=0),则 R32_DFSDM_CHyDATINR 寄存器被写保护。

来自内部 ADC 的输入(仅限 ADC1 和 ADC2)

对于并行 ADC 数据输入(DATMPX[1:0]=1),ADC[y+1]结果被分配至通道 y 输入(ADC1 填充 R32_DFSDM_CH0DATINR 寄存器,ADC2 填充 R32_DFSDM_CH1DATINR 寄存器。来自 ADC[y+1]的转换结束事件会导致更新通道 y 的数据(来自 ADC[y+1]的并行数据被用作数字滤波器的下一个采样)。转换结束事件发生时,来自 ADC[y+1]的数据被写入 R32_DFSDM_CHyDATINR 寄存器(INDAT0[15:0]字段)。

数据封装模式设置(R32_DFSDM_CHyCFGR1 寄存器中的 DATPACK[1:0])对 ADC 数据输入无影响。

注:ADC 扩展规范:将内部 ADC 配置为交错模式时(例如,ADC1 与 ADC2 交错-请参考 ADC 规范),来自 ADC1 或 ADC2 的每个结果均会传送至同一 16 位总线(即 ADC1 的总线),该总线连至 DFSDM 通道 0 (固定连接)。因此,DFSDM 通道 0 中将存在双倍输入数据速率(偶数采样来自 ADC1,奇数采样来自

ADC2)。与 ADC2 相关的通道 1 将处于空闲状态。

来自存储器的输入（直接由 CPU/DMA 写入）

当 CPU 或 DMA (DATMPX[1:0]=2) 直接将数据写入 R32_DFSDM_CHyDATINR 寄存器时，这些数据可以作为输入，用于处理来自存储器或外设的数字数据流。

数据可由 CPU 或 DMA 写入到 R32_DFSDM_CHyDATINR 寄存器中：

1) CPU 数据写入：

输入数据直接由 CPU 写入到 R32_DFSDM_CHyDATINR 寄存器中。

2) DMA 数据写入：

DMA 应配置为存储器到存储器传输模式，以将存储器缓冲区中的数据传输至 R32_DFSDM_CHyDATINR 寄存器中。目标存储器地址为 R32_DFSDM_CHyDATINR 寄存器的地址。数据以 DMA 传输速度从存储器传输至 DFSDM 并行输入。

此 DMA 不同于读取 DFSDM 转换结果所使用的 DMA。这两个 DMA 可同时使用，第一个 DMA（配置为存储器到存储器传输）用于输入数据写入，第二个 DMA（配置为外设到存储器传输）用于数据结果读取。

对 R32_DFSDM_CHyDATINR 的访问可为 16 位或 32 位宽，允许分别在一次写操作中加载一个或两个采样。32 位输入数据寄存器 (R32_DFSDM_CHyDATINR) 可用一个或两个 16 位数据采样进行填充，具体取决于在 R32_DFSDM_CHyCFGR1 寄存器的 DATPACK[1:0] 字段中定义的数据封装操作模式：

1) 标准模式 (DATPACK[1:0]=00b)：

只有一个样本存储在 R32_DFSDM_CHyDATINR 寄存器的 INDAT0[15:0] 字段中，其用作通道 y 的输入数据。高 16 位 (INDAT1[15:0]) 被忽略，且被写保护。数字滤波器必须执行一次输入采样（通过 INDAT0[15:0]），以便清空由 CPU/DMA 填充后的数据寄存器。该模式与对 R32_DFSDM_CHyDATINR 寄存器的 16 位 CPU/DMA 访问结合使用，以在每次写操作时加载一个采样。

2) 交错模式 (DATPACK[1:0]=01b)：

R32_DFSDM_CHyDATINR 寄存器被用作两个采样缓冲区。第一个采样存储在 INDAT0[15:0] 中，第二个采样存储在 INDAT1[15:0] 中。数字滤波器必须通过通道 y 执行两次输入采样，以清空 R32_DFSDM_CHyDATINR 寄存器。该模式与对 R32_DFSDM_CHyDATINR 寄存器的 32 位 CPU/DMA 访问结合使用，以在每次写操作时加载两个采样。

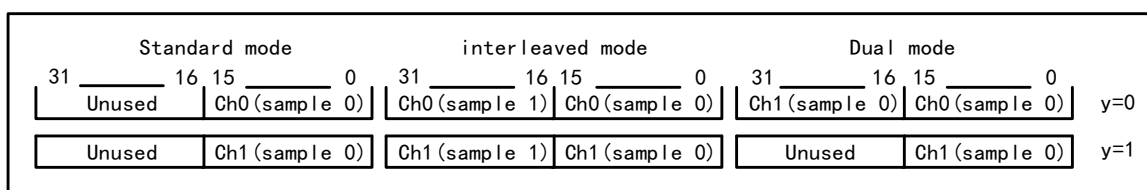
3) 双通道模式 (DATPACK[1:0]=10b)：

将两个采样写入到 R32_DFSDM_CHyDATINR 寄存器中。INDAT0[15:0] 中的数据用于通道 y，INDAT1[15:0] 中的数据用于通道 y+1。INDAT1[15:0] 中的数据会被自动复制到下一 (y+1) 通道数据寄存器 DFSDM_CH[y+1]DATINR 的 INDAT0[15:0] 中。数字滤波器必须执行两次采样（一次来自通道 y，另一次来自通道 (y+1)），以清空 R32_DFSDM_CHyDATINR 寄存器。

只能针对偶数通道数 (y=0) 进行双通道模式设置 (DATPACK[1:0]=10b)。如果将奇数通道 (y=1) 设为双通道模式，则对于该通道，INDAT0[15:0] INDAT1[15:0] 部分均被写保护。如果偶数通道被设为双通道模式，则下一奇数通道必须设置成标准模式 (DATPACK[1:0]=00b)，以便与偶数通道正确配合。

有关 R32_DFSDM_CHyDATINR 寄存器数据模式和通道数据样本分配，请参考下图。

图 42-5 R32_DFSDM_CHyDATINR 寄存器操作模式和分配



首先使能所选输入通道（通道 y），以便进行数据采集（启动通道 y 转换），然后必须向 R32_DFSDM_CHyDATINR 寄存器进行写操作，以加载一个或两个采样。否则，在下次处理时会丢失写入的数据。

例如：对于单次转换和交错模式，在启动单次转换前，不要开始向 R32_DFSDM_CHyDATINR 写入成对数据样本（开始转换前 R32_DFSDM_CHyDATINR 中存在的数据均会被丢弃）。

42.3.5 通道选择

存在 2 个复用通道，可选择用于注入通道组转换和/或常规通道转换。

注入通道组可以选择 2 个通道中的任一通道或其全部。R32_DFSDM_FLTxJCHGR 寄存器中的 JCHG[1:0] 选择注入组通道，其中 JCHG[y]=1 表示已选择通道 y。

注入转换可工作在扫描模式（JSCAN=1）或单个模式（JSCAN=0）下。在扫描模式下，每个所选通道都会依次转换。最低通道（通道 0，如果已选择）最先转换，紧接着转换下一个较高通道，直至 JCHG[1:0] 选择的所有通道均被转换。在单个模式（JSCAN=0）下，只会转换所选的一个通道，通道选择会移至下一通道。JSCAN=0 时，对 JCHG[1:0] 进行写入操作会将通道选择复位为选择的最低通道。

注入转换可由软件或触发器启动。此类转换从不会被常规转换中断。

常规通道只选择 2 个通道的其中一个。R32_DFSDM_FLTxCR1 寄存器中的 RCH 位指示所选通道。

常规转换只能由软件启动（不能由触发器启动）。请求注入转换时，会暂时中断连续常规转换序列。

对已禁止通道（R32_DFSDM_CHyCFGR1 寄存器中的 CHEN=0）执行转换会导致转换将永远不会结束，这是因为未提供输入数据的原因（无时钟信号）。在这种情况下，需要使能给定通道（R32_DFSDM_CHyCFGR1 寄存器中的 CHEN=1），或停止转换（通过 R32_DFSDM_FLTxCR1 寄存器的 DFEN=0）。

42.3.6 数字滤波器配置

DFSDM 包含 Sinc^x 类型数字滤波器实现。此 Sinc^x 滤波器执行输入数字数据流滤波，这会导致减小输出数据速率（抽取）以及增大输出数据分辨率。可对 Sinc^x 数字滤波器进行配置，以达到所需输出数据速率和所需输出数据分辨率。可配置参数有：

- 滤波器阶数/类型：（参考 R32_DFSDM_FLTxFCR 寄存器中的 FORD[2:0] 位）：

- FastSinc
- Sinc¹
- Sinc²
- Sinc³
- Sinc⁴
- Sinc⁵

- 滤波器过采样/抽取率：（请参考 R32_DFSDM_FLTxFCR 寄存器中的 FOSR[9:0] 位）：

- FOSR=1-1024——适用于 FastSinc 滤波器和 Sinc^x 滤波器 x=F_{ORD}=1..3
- FOSR=1-215——适用于 Sinc^x 滤波器 x=F_{ORD}=4
- FOSR=1-73——适用于 Sinc^x 滤波器 x=F_{ORD}=5

滤波器支持以下传递函数（H 域中的冲激响应）：

- Sinc^x 滤波器类型：

$$H(z) = \left(\frac{1-z^{-FOSR}}{1-z^{-1}} \right)^x$$

- FastSinc 滤波器类型：

$$H(z) = \left(\frac{1-z^{-FOSR}}{1-z^{-1}} \right)^2 \times (1 + z^{-(2 \times FOSR)})$$

图 42-6 示例：Sinc3 滤波器响应

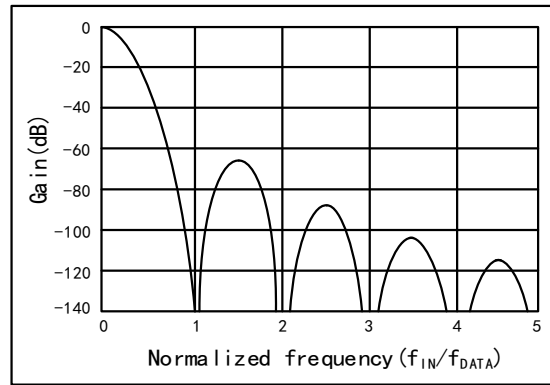


表 42-1 滤波器最大输出分辨率（来自滤波器输出的峰值数据值），基于某些 FOSR 值

FOSR	Sinc ¹	Sinc ²	FastSinc	Sinc ³	Sinc ⁴	Sinc ⁵
x	+/-x	+/-x ²	+/-2x ²	+/-x ³	+/-x ⁴	+/-x ⁵
4	+/-4	+/-16	+/-32	+/-64	+/-256	+/-1024
8	+/-8	+/-64	+/-128	+/-512	+/-4096	-
32	+/-32	+/-1024	+/-2048	+/-32768	+/-1048576	+/-33554432
64	+/-64	+/-4096	+/-8192	+/-262144	+/-16777216	+/-1073741824
128	+/-128	+/-16384	+/-32768	+/-2097152	+/-268435456	
256	+/-256	+/-65536	+/-131072	+/-16777216	在满量程输入的条件下，结果会溢出（>32 位有符号整数）	
1024	+/-1024	+/-1048576	+/-2097152	+/-1073741824		

42.3.7 积分器单元

积分器会对来自数字滤波器的数据进一步提升抽取率和分辨率。对于来自滤波器的给定数量的数据采样，积分器对其进行简单的求和。

积分器过采样率参数定义了将多少个数据的和作为积分器的一个数据输出。IOSR 可设置为介于 1~256 范围内（参考 R32_DFSDM_FLTxFCR 寄存器中的 IOSR[3:0] 位说明）。

表 42-2 积分器最大输出分辨率（来自积分器输出的峰值数据值），
基于某些 IOSR 值，FOSR=256 以及 Sinc3 滤波器类型（最大数据）

IOSR	Sinc ¹	Sinc ²	FastSinc	Sinc ³	Sinc ⁴	Sinc ⁵
x	+/-FOSR · x	+/-FOSR ² · x	+/-2FOSR ² · x	+/-FOSR ³ · x	+/-FOSR ⁴ · x	+/-FOSR ⁵ · x
4	-	-	-	+/-67108864	-	-
32	-	-	-	+/-536870912	-	-
128	-	-	-	+/-2147483648	-	-
256	-	-	-	+/-2 ³²	-	-

42.3.8 模拟看门狗

模拟看门狗用于在达到或超出给定阈值上限或者达到或低于给定阈值下限时，触发外部信号（断路或中断）。随即可调用中断/事件/断路生成。

每个模拟看门狗都会根据（R32_DFSDM_FLTxCR1 寄存器中）AWFSEL 位设置监视串行数据接收器输出（在每个通道上模拟看门狗滤波器之后）或数据输出寄存器（当前注入或常规转换结果）。是否需要通过模拟看门狗 x 监视的输入通道可以通过 R32_DFSDM_FLTxCR2 寄存器中的 AWDCH[1:0] 选择。

输入通道上的模拟看门狗转换与标准转换无关。在标准转换模式下，模拟看门狗在每个输入通道上使用自己的滤波器和信号处理，与注入或常规主转换无关。在连续转换模式下，对所选输入通道执行模拟看门狗转换，以便在注入或常规主转换暂停（RCIP=0，JCIP=0）时也仍能对通道进行监视。

阈值上限和阈值下限寄存器与给定数据值（通过 R32_DFSDM_FLTxAWHTR 寄存器的 AWHT[23:0]位和 R32_DFSDM_FLTxAWLTR 寄存器的 AWLT[23:0]位进行设置）进行比较。

可以选择 2 种情况将阈值寄存器与数据值进行比较：

- 选择 1：在这种情况下，输入数据取自最终输出数据寄存器（AWFSEL=0）。该选择具有以下特性：
 - 高输入数据分辨率（高达 24 位）
 - 响应时间慢——不适用于过流检测等快速响应应用
 - 为了进行比较，最终数据需要经过移位和偏移数据校正
 - 最终数据仅在执行常规或注入主转换后才可用
 - 可在并行输入数据源（R32_DFSDM_CHyCFGR1 寄存器中的 DATMPX[1:0] ≠ 0）情况下使用
- 选择 2：在这种情况下，输入数据取自任一串行数据接收器输出（AWFSEL=1）。该选择具有以下特性：
 - 输入串行数据通过专用模拟看门狗 Sinc^x通道滤波器进行处理，滤波器的过采样率（1..32 和滤波器阶数（1..3）可以配置（请参考 R32_DFSDM_CHyAWSCDR 寄存器中的 AWFOSR[4:0]和 AWFORD[1:0]位设置）
 - 较低分辨率（最高为 16 位）
 - 响应时间快——适用于需要快速响应的应用，如过流/过压检测
 - 数据以连续模式提供，与常规或注入主转换活动无关

进行输入通道监视（AWFSEL=1）时，用于跟阈值进行比较的数据取自 AWDCH[1:0] 字段（R32_DFSDM_FLTxCR2 寄存器）所选的通道。将每个所选通道滤波器结果与一个阈值对（AWHT[23:0]/AWLT[23:0]）进行比较。此时，只有高 16 位（AWHT[23:8]/AWLT[23:8]）会定义与模拟看门狗滤波器输出进行比较的 16 位阈值，因为来自模拟看门狗滤波器的数据的分辨率最高为 16 位。在这种情况下（AWFSEL=1），不对 AWHT[7:0]/AWLT[7:0]位进行比较。

在 R32_DFSDM_CHyAWSCDR 寄存器中设置各个输入通道的模拟看门狗滤波器配置参数（滤波器阶数 AWFORD[1:0]和滤波器过采样率 AWFOSR[4:0]）。

给定通道 y 的模拟看门狗滤波器数据可通过固件从 DFSDM_CHyWDATR 寄存器 WDATA[15:0] 字段中读取。如果 R32_DFSDM_CHyCFGR1 寄存器中的 CHEN=1，则该模拟看门狗滤波器数据被连续转换，其数据速率由模拟看门狗滤波器设置和通道输入时钟频率给定。

模拟看门狗滤波器转换的工作方式类似于常规的快速连续转换（无积分器）。模拟看门狗滤波器输出（通道输入时钟频率为 f_{CKIN} ）一个结果所需的串行采样数量：

首次转换：

对于 Sinc^x滤波器（ $x=1..5$ ）：采样数= $[F_{OSR} * (F_{ORD}+1) + 2]$

对于 FastSinc 滤波器：采样数= $[F_{OSR} * 4 + 2 + 1]$

后续转换：

对于 Sinc^x和 FastSinc 滤波器：采样数= $[FOSR * FORD]$

其中：

F_{OSR} 即滤波器过采样率： $F_{OSR}=AWFOSR[4:0]+1$ （参考 R32_DFSDM_CHyAWSCDR 寄存器）

F_{ORD} 即滤波器阶数： $F_{ORD}=AWFORD[1:0]$ （参考 R32_DFSDM_CHyAWSCDR 寄存器）

对于输出数据寄存器监视（AWFSEL=0），在完成最终数据的右移位和偏移校正（请参考 R32_DFSDM_CHyCFGR2 寄存器的 OFFSET[23:0] 和 DTRBS[4:0] 字段）后进行比较。在由（R32_DFSDM_FLTxCR2 寄存器中）AWDCH[1:0] 字段所选通道的每次注入或常规转换结束后，进行比较。

模拟看门狗事件的状态反映在 R32_DFSDM_FLTxAWSR 寄存器中，该寄存器中锁定了给定事件。AWHTF[y]=1 标志指示在通道 y 上是否超出了 AWHT[23:0] 值。AWLTF[y]=1 标志指示在通道 y 上是否超出了 AWLT[23:0] 值。对于 R32_DFSDM_FLTxAWSR 寄存器中锁存的事件，将通过向 R32_DFSDM_FLTxAWCFR 寄存器中的相应清零位 CLRAWHTF[y] 或 CLRAWLTF[y] 写入 1 来清除。

模拟看门狗的全局状态通过 R32_DFSDM_FLTxISR 寄存器中的 AWDF 标志位来反映（其用于快速检测中断源）。AWDF=1 表示至少发生了一个看门狗事件（至少对于一个通道，AWHTF[y]=1 或 AWLTF[y]=1）。

所有 AWHTF[1:0] 和 AWLTF[1:0] 均清零时, AWDF 位才清零。

可将模拟看门狗事件分配至断路输出信号。可将两个断路输出分配至超出阈值上限或低于阈值下限的事件 (DFSDM_BK[1:0])。通过 R32_DFSDM_FLTxAWHTR 寄存器和 R32_DFSDM_FLTxAWLTR 寄存器中的 BKAWH[1:0] 和 BKAWL[1:0] 字段将断路信号分配至给定模拟看门狗事件。

42.3.9 短路检测器

使用短路检测器的目的在于, 当模拟信号达到饱和值 (超出满量程范围) 并保持给定的时间时, 短路检测器可以在极快的响应时间内发出信号。该特性可检测短路或断路故障 (例如过流或过压)。可调用中断/事件/断路生成。

短路检测器的输入数据取自通道收发器输出。

各个输入通道上都有一个递增计数器, 用于对串行数据接收器输出上的连续 0 或 1 进行计数。如果接收的数据流发生变化 (数据信号从 1 到 0 或从 0 到 1), 则会重新启动计数器。如果该计数器达到短路阈值寄存器值 (R32_DFSDM_CHyAWSCDR 寄存器中的 SCDT[7:0] 位), 则会调用短路事件。每个输入通道都配有短路检测器。通过将 SCDEn 位 (R32_DFSDM_CHyCFGR1 寄存器中) 置 1, 可选择对任一通道进行连续监视, 每个通道都有其自己的短路检测器设置 (阈值位 SCDT[7:0]、状态位 SCDF[1:0]、状态清零位 CLRSCDF[1:0])。禁止相应通道 y (CHEN[y]=0) 时, 状态标志 SCDF[y] 还可由硬件清零。

对于各个通道, 可将短路检测器事件分配至断路输出信号 DFSDM_BK[1:0]。可将两个断路输出分配至短路检测器事件。通过 R32_DFSDM_CHyAWSCDR 寄存器中的 BKSCD[1:0] 字段将断路信号分配至给定通道短路检测器事件。

如果选择了并行输入数据通道 (R32_DFSDM_CHyCFGR1 寄存器中的 DATMPX[1:0] ≠ 0), 则无法使用短路检测器。

总共有两个断路输出 (与模拟看门狗功能共享)。

42.3.10 极值检测器

极值检测器用于采集最终输出数据字的最小值和最大值 (峰峰值)。

如果输出数据字高于极值检测器最大值寄存器 (R32_DFSDM_FLTxEVMAX 寄存器中的 EVMAX[23:0] 位) 中存储的值, 则该寄存器将用当前输出数据字值进行更新, 且存储其数据的通道位于 (R32_DFSDM_FLTxEVMAX 寄存器的) EVMAXCH 位中。

如果输出数据字低于极值检测器最小值寄存器 (R32_DFSDM_FLTxEVMIN 寄存器中的 EVMIN[23:0] 位) 中存储的值, 则该寄存器将用当前输出数据字值进行更新, 且存储其数据的通道位于 (R32_DFSDM_FLTxEVMIN 寄存器的) EVMINCH 位中。

最小值和最大值寄存器值可由软件 (通过读取 R32_DFSDM_FLTxEVMAX 或 R32_DFSDM_FLTxEVMIN 寄存器) 进行刷新。刷新后, 极值检测器最小值数据寄存器 (R32_DFSDM_FLTxEVMIN) 用 0x7FFFFFFF (最大正值) 进行填充, 极值检测器最大值寄存器 (R32_DFSDM_FLTxEVMAX) 用 0x800000 (最小负值) 进行填充。

极值检测器在完成右移位和偏移数据校正之后, 才执行比较操作。对于每个极值检测器, 计算极值时涉及的输入通道由 (R32_DFSDM_FLTxCR2 寄存器中的) EXCH[1:0] 位选定。

42.3.11 数据单元模块

数据单元模块是整个处理路径的最后一个模块: 外部 $\Sigma \Delta$ 调制器——串行收发器——Sinc 滤波器——积分器——数据单元块。

输出数据速率取决于串行数据流速率以及滤波器和积分器设置。最大输出数据速率为:

$$\text{数据速率 [采样数/秒]} = \frac{f_{\text{CKIN}}}{F_{\text{OSR}} * (2^{I_{\text{OSR}}} + F_{\text{ORD}} + 1) + 2}; \text{FAST}=0, \text{Sinc}^x \text{ 滤波器}$$

$$\text{数据速率 [采样数/秒]} = \frac{f_{\text{CKIN}}}{F_{\text{OSR}} * (2^{I_{\text{OSR}}} + 5) + 2}; \text{FAST}=0, \text{FastSinc 滤波器}$$

或

$$\text{数据速率[采样数/秒]} = \frac{f_{\text{CKIN}}}{F_{\text{OSR}} * 2^{I_{\text{OSR}}}}; \text{FAST}=1$$

并行数据输入时的最大输出数据速率:

$$\text{数据速率[采样数/秒]} = \frac{f_{\text{DATAIN\_RATE}}}{F_{\text{OSR}} * (2^{I_{\text{OSR}}} + F_{\text{ORD}} + 1) + 2}; \text{FAST}=0, \text{Sinc}^x \text{ 滤波器}$$

或

$$\text{数据速率[采样数/秒]} = \frac{f_{\text{DATAIN\_RATE}}}{F_{\text{OSR}} * (2^{I_{\text{OSR}}} + 5) + 2}; \text{FAST}=0, \text{FastSinc 滤波器}$$

或

$$\text{数据速率[采样数/秒]} = \frac{f_{\text{DATAIN\_RATE}}}{F_{\text{OSR}} * 2^{I_{\text{OSR}}}}; \text{FAST}=1 \text{ 或任一旁路情形 } (F_{\text{OSR}}=1)$$

其中: $f_{\text{DATAIN\_RATE}}$ 即 ADC 或 CPU/DMA 的输入数据速率。

在该模块中执行最终数据的右移位, 因为最终数据宽度为 24 位, 而来自处理路径的数据最高可为 32 位。对于每个所选输入通道, 此右移位均可在 0~31 位的范围内进行配置 (请参考 R32_DFSDM_CHyCFGR2 寄存器的 DTRBS[4:0] 位)。右移位将结果四舍五入为最接近的整数值。移位结果的符号保留, 以便得到有效 24 位有符号格式的结果数据。

接下来对结果进行偏移校正。从给定通道的输出数据中减去偏移校正 (存储在寄存器 R32_DFSDM_CHyCFGR2 中的 OFFSET[23:0])。OFFSET[23:0] 字段中的数据由软件通过相应的校准程序进行设置。

因数字处理过程中的所有操作均在 32 位有符号寄存器上执行, 因此为保证结果不会发生溢出, 必须满足以下条件:

- 1) $F_{\text{OSR}}^{F_{\text{ORD}}} \cdot 2^{I_{\text{OSR}}} \leq 2^{31}$, 适用于 Sinc^x 滤波器 ($x=1..5$);
- 2) $F_{\text{OSR}}^2 \cdot 2^{I_{\text{OSR}}} \leq 2^{31}$, 适用于 FastSinc 滤波器。

注: 滤波器和积分器旁路时 ($I_{\text{OSR}}[3:0]=0$, $F_{\text{OSR}}[9:0]=0$), 输入数据速率 ($f_{\text{DATAIN\_RATE}}$) 必须被限制为能够读取所有输出数据: $f_{\text{DATAIN\_RATE}} \leq f_{\text{HCLK}}$; 其中, f_{HCLK} 为 DFSDM 外设连接的总线频率。

42.3.12 有符号数据格式

DFSDM 输入串行通道的每一通道均能与外部 $\Sigma \Delta$ 调制器相连接。一个外部 $\Sigma \Delta$ 调制器配置有两个差分输入 (正负), 这两个输入既可进行差分测量也可用于单端信号测量。

始终假设 $\Sigma \Delta$ 调制器的输出采用有符号格式 (即来自 $\Sigma \Delta$ 调制器的 0/1 数据流代表数值 -1 和 +1)。

寄存器的有符号数据格式: 在最终输出数据、模拟看门狗、极值检测器以及偏移校正的寄存器中所存储的数据均为有符号格式。输出数据字的最高有效位 (msb) 用于表示数值的符号 (采用二进制补码格式)。

42.3.13 启动转换

启动注入转换的方法如下:

- 软件: 向 R32_DFSDM_FLTxCRI 寄存器中的 JSWSTART 写入 1。
- 触发: 在 JEXTEN 激活的情况下, 通过 JEXTSEL[3:0] 选择触发信号, 并确定有效边沿 (参考 R32_DFSDM_FLTxCRI 寄存器)。
- 若 JSYNC 置为 1, 则 DFSDM_FLT0 将同步启动注入转换: 对于 DFSDM_FLTx ($x>0$), 注入转换将在 DFSDM_FLT0 中自动开始; 注入转换由软件触发 (通过 R32_DFSDM_FLT0CR2 寄存器的 JSWSTART 位设置为 1)。DFSDM_FLTx ($x>0$) 中的每个注入转换都将根据其本地设置 (如 JSCAN 和 JCHG 等) 执行。

在特定时间内, 仅允许执行一次注入转换。因此, 若已有注入转换请求发出但尚未完成, 任何后续的启动请求都将被忽略。

启动常规转换的方法如下：

- 软件：向 R32_DFSDM_FLTxCR1 寄存器中的 RSWSTART 写入 1。
- 如果 RSYNC=1，则使用 DFSDM_FLT0 同步启动：对于 DFSDM_FLTx (x>0)，常规转换会在 DFSDM_FLT0 中自动启动；常规转换由软件触发（通过 R32_DFSDM_FLT0CR2 寄存器的 RSWSTART 位设置为 1）。DFSDM_FLTx (x>0) 中的每个常规转换都将根据其本地配置设置（RCONT 和 RCH 等）执行。

42.3.14 连续和快速连续模式

如果将 R32_DFSDM_FLTxCR1 寄存器中的 RCONT 置 1，则会在连续模式下执行常规转换。RCONT=1 表示向 RSWSTART 写入 1 后，重复转换 RCH 位所选的通道。

可通过向 RCONT 写入 0 停止在连续模式下执行的常规转换。将 RCONT 清零后，会立即停止正在进行的转换。

在连续模式下，可通过将 R32_DFSDM_FLTxCR1 寄存器的 FAST 位置 1 来提高数据速率。此时，如果对一个通道进行连续转换，则滤波器无需新的刷新数据重新填充，因为滤波器内部数据有效，这些数据是先前通过对连续数据进行采样而得。速度提升与所选的滤波器阶数有关。通过 RSWSTART=1 启动连续转换后，将充分进行快速模式（FAST=1）下的首次转换（与 FAST=0 时相同），之后会在较短的间隔内完成各个后续转换。

连续模式下的转换时间：

FAST=0 时（或 FAST=1 时的快速转换）：

对于 Sinc^x 滤波器：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * (2^{10\text{SR}} + F_{\text{ORD}} + 1) + 2] / f_{\text{CKIN}}$$

对于 FastSinc 滤波器：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * (2^{10\text{SR}} + 5) + 2] / f_{\text{CKIN}}$$

FAST=1 时（首次转换除外）：

对于 Sinc^x 和 FastSinc 滤波器：

$$t = \text{CNVCNT} / f_{\text{DFSDMCLK}} = [F_{\text{OSR}} * 2^{10\text{SR}}] / f_{\text{CKIN}}$$

FOSR=FOSR[9:0]+1=1 时（滤波器旁路，仅积分器有效）：

$$t = 2^{10\text{SR}} / f_{\text{CKIN}} \quad (\dots \text{但是 } \text{CNVCNT}=0)$$

连续模式不适用于注入转换。注入转换可通过定时器触发进行启动，以便使用精确时序仿真连续模式。

如果常规连续转换正在进行（RCONT=1），并且如果对请求常规连续转换（RCONT=1）的 R32_DFSDM_FLTxCR1 寄存器进行写访问，则会从下一转换周期重新启动常规连续转换（类似于新的常规连续转换应用于新通道选择，即使 R32_DFSDM_FLTxCR1 寄存器中无更改，亦如此）。

42.3.15 请求优先级

注入转换的优先级高于常规转换。正在进行的常规转换会立即被注入转换请求中断；在注入转换完成后将重新启动此常规转换。

如果某个注入转换处于待处理状态或正在进行，则不能启动其他注入转换：只要（R32_DFSDM_FLTxISR 寄存器中）位 JCIP 为 1，就会忽略启动注入转换（通过 JSWSTART 或触发）的请求。

同理，如果某个常规转换处于待处理状态或正在进行，则不能启动其他常规转换：只要（R32_DFSDM_FLTxISR 寄存器中）位 RCIP 为 1，就会忽略启动常规转换（使用 RSWSTART）的请求。

不过，如果在进行某个常规转换时请求注入转换，则常规转换会立即停止，并会启动注入转换。常规转换之后会重新启动，并且此延迟的重新启动会通过位 RPEND 进行表示。

注入转换的优先级高于常规转换，因为注入转换可暂时中断连续的常规转换序列。完成注入转换序列时，如果 RCONT 仍置 1 则会重新启动连续常规转换（RPEND 位将发出基于首次常规转换结果的延迟启动信号）。

通过对 DFSDM 进行相同的写操作启动相关操作时，或者，如果在其他操作结束时多个操作处于待处理状态，则优先级也至关重要。例如，假设注入转换正在进行（JCIP=1），则对 R32_DFSDM_FLTxCR1 进行单次写操作将向 RSWSTART 写入 1，以请求常规转换。完成注入序列时，优先级指示接下来执行常规转换，并且通过 RPEND 位指示相应的延迟启动。

42.3.16 在运行模式下的功耗优化

为降低功耗，DFSDM 滤波器和积分器会在未用于转换时自动进入空闲状态（RCIP=0，JCIP=0）。

42.3.17 触发信号与触发源

表 42-3 触发信号与触发源连接

触发器名称	触发源	触发器名称	触发源
DFSDM_JTRG0	TIM1_TRG0	DFSDM_JTRG8	TIM9_TRG0
DFSDM_JTRG1	TIM2_TRG0	DFSDM_JTRG9	TIM10_TRG0
DFSDM_JTRG2	TIM3_TRG0	DFSDM_JTRG10	TIM11_TRG0
DFSDM_JTRG3	TIM4_TRG0	DFSDM_JTRG11	TIM12_TRG0
DFSDM_JTRG4	TIM5_TRG0	DFSDM_JTRG12	EXTI11
DFSDM_JTRG5	TIM6_TRG0	DFSDM_JTRG13	EXTI15
DFSDM_JTRG6	TIM7_TRG0	DFSDM_JTRG14	LPTIM1
DFSDM_JTRG7	TIM8_TRG0	DFSDM_JTRG15	LPTIM2

42.3.18 DFSDM 断路连接

表 42-4 DFSDM 断路连接

断路名称	断路目标
DFSDM_BK[0]	TIM1 断路 1
DFSDM_BK[1]	TIM1 断路 2

42.4 中断

为增强中央处理单元（CPU）性能，已实现了一组与 CPU 事件发生相关的中断：

- 注入转换结束中断：
 - 通过 R32_DFSDM_FLTxCR2 寄存器 JE0CIE 位使能
 - 通过 R32_DFSDM_FLTxISR 寄存器 JE0CF 位进行指示
 - 通过读取 R32_DFSDM_FLTxJDATAR 寄存器（注入数据）进行清除
 - 指示哪一个通道发生了转换结束，并在 R32_DFSDM_FLTxJDATAR 寄存器 JDATACH 位中进行报告
- 常规转换结束中断：
 - 通过 R32_DFSDM_FLTxCR2 寄存器 RE0CIE 位使能
 - 通过 R32_DFSDM_FLTxISR 寄存器 RE0CF 位进行指示
 - 通过读取 R32_DFSDM_FLTxRDATAR 寄存器（常规数据）进行清除
 - 指示哪一个通道发生了转换结束，并在 R32_DFSDM_FLTxRDATAR 寄存器 RDATACH 位中进行报告
- 注入转换数据溢出中断：
 - 当注入的转换数据未从 R32_DFSDM_FLTxJDATAR 寄存器中读取（通过 CPU 或 DMA），并被新注入转换覆盖时，会发生此中断
 - 通过 R32_DFSDM_FLTxCR2 寄存器 JOVRIE 位使能

- 通过 R32_DFSDM_FLTISR 寄存器 JOVRF 位进行指示
- 通过向 R32_DFSDM_FLTICR 寄存器 CLRJOVRF 位写入 1 进行清除
- 常规转换数据溢出中断：
 - 当常规转换数据未从 R32_DFSDM_FLTxDATAR 寄存器中读取（通过 CPU 或 DMA），并被新常规转换覆盖时，会发生此中断
 - 通过 R32_DFSDM_FLTCR2 寄存器 ROVRIE 位使能
 - 通过 R32_DFSDM_FLTISR 寄存器 ROVRF 位进行指示
 - 通过向 R32_DFSDM_FLTICR 寄存器 CLRROVRF 位写入 1 进行清除
- 模拟看门狗中断：
 - 当转换数据（输出数据或模拟看门狗滤波器中的数据——具体取决于 R32_DFSDM_FLTCR1 寄存器中的 AWFSEL 位设置）超出 R32_DFSDM_FLTAWHTR/R32_DFSDM_FLTAWLTR 寄存器中的阈值上限或低于相应阈值下限时，会发生此中断
 - 通过（所选通道 AWDCH[1:0] 的）R32_DFSDM_FLTCR2 寄存器 AWDIE 位使能
 - 由 R32_DFSDM_FLTISR 寄存器 AWDF 位指示
 - 通过 R32_DFSDM_FLTAWSR 寄存器 AWHTF[1:0] 和 AWLTF[1:0] 字段单独指示模拟看门狗阈值上限或下限错误
 - 通过向 R32_DFSDM_FLTAWCFR 寄存器相应 CLRAWHTF[1:0] 或 CLRAWLTF[1:0] 位写入 1 进行清除
- 短路检测器中断：
 - 稳定的数据数超出 R32_DFSDM_CHyAWSCDR 寄存器中的阈值时会发生此中断
 - 通过（通过 R32_DFSDM_CHyCFGR1 寄存器 SCDEN 位选择的通道的）R32_DFSDM_FLTCR2 寄存器的 SCDIE 位使能
 - 通过 R32_DFSDM_FLTISR 寄存器 SCDF[1:0] 位指示（还会报告发生短路检测器事件的通道）
 - 通过向 R32_DFSDM_FLTICR 寄存器中的相应 CLRSCDF[1:0] 位写入 1 进行清除
- 通道时钟缺失中断：
 - CKINy 引脚上出现时钟缺失时会发生此中断（参考第 42.3.2 节：串行通道收发器的时钟缺失检测）
 - 通过（通过 R32_DFSDM_CHyCFGR1 寄存器 CKABEN 位选择的通道的）R32_DFSDM_FLTCR2 寄存器的 CKABIE 位使能
 - 通过 R32_DFSDM_FLTISR 寄存器 CKABF[y] 位进行指示
 - 通过向 R32_DFSDM_FLTICR 寄存器 CLRCKABF[y] 位写入 1 进行清除

表 42-5 DFSDM 中断请求

中断事件	事件标志	事件/中断清除方法	中断使能控制位
注入转换结束	JE0CF	读取 R32_DFSDM_FLTxDATAR	JE0CIE
常规转换结束	RE0CF	读取 R32_DFSDM_FLTxDATAR	RE0CIE
注入数据溢出	JOVRF	写入 CLRJOVRF=1	JOVRIE
常规数据溢出	ROVRF	写入 CLRROVRF=1	ROVRIE
模拟看门狗	AWDF、 AWHTF[1:0]、 AWLTF[1:0]	写入 CLRAWHTF[1:0]=1 写入 CLRAWLTF[1:0]=1	AWDIE、 (AWDCH[1:0])
短路检测器	SCDF[1:0]	写入 CLRSCDF[1:0]=1	SCDIE、 (SCDEN)
通道时钟缺失	CKABF[1:0]	写入 CLRCKABF[1:0]=1	CKABIE、 (CKABEN)

42.5 DFSDM DMA 传输

为降低 CPU 干预，可使用 DMA 传输将转换传输至存储器。注入转换的 DMA 传输通过将 R32_DFSDM_FLTxCR1 寄存器的 JDMAEN 位置 1 使能。常规转换的 DMA 传输通过将 R32_DFSDM_FLTxCR1 寄存器的 RDMAEN 位置 1 使能。

注：通过 DMA 传输，中断标志在注入或常规转换结束时自动清零（R32_DFSDM_FLTxISR 寄存器中的 JE0CF 或 RE0CF 位），因为 DMA 正在读取 R32_DFSDM_FLTxJDATAR 或 R32_DFSDM_FLTxRDATAR 寄存器。

42.6 寄存器描述

表 42-6 DFSDM 通道相关寄存器列表

名称	访问地址	描述	复位值
R32_DFSDM_CH0CFGR1	0x40017000	DFSDM 通道 0 配置寄存器 1	0x00000000
R32_DFSDM_CH0CFGR2	0x40017008	DFSDM 通道 0 配置寄存器 2	0x00000000
R32_DFSDM_CH0AWSCDR	0x40017010	DFSDM 通道 0 模拟看门狗和短路检测器寄存器	0x00000000
R32_DFSDM_CH0WDATR	0x40017018	DFSDM 通道 0 看门狗滤波器数据寄存器	0x00000000
R32_DFSDM_CH0DATINR	0x40017020	DFSDM 通道 0 数据输入寄存器	0x00000000
R32_DFSDM_CH1CFGR1	0x40017004	DFSDM 通道 1 配置寄存器 1	0x00000000
R32_DFSDM_CH1CFGR2	0x4001700C	DFSDM 通道 1 配置寄存器 2	0x00000000
R32_DFSDM_CH1WSCDR	0x40017114	DFSDM 通道 1 模拟看门狗和短路检测器寄存器	0x00000000
R32_DFSDM_CH1WDATR	0x4001711C	DFSDM 通道 1 看门狗滤波器数据寄存器	0x00000000
R32_DFSDM_CH1DATINR	0x40017024	DFSDM 通道 1 数据输入寄存器	0x00000000

表 42-7 DFSDM 滤波器相关寄存器列表

名称	访问地址	描述	复位值
R32_DFSDM_FLT0CR1	0x40017028	DFSDM 滤波器 0 控制寄存器 1	0x00000000
R32_DFSDM_FLT0CR2	0x40017030	DFSDM 滤波器 0 控制寄存器 2	0x00000000
R32_DFSDM_FLT0ISR	0x40017038	DFSDM 滤波器 0 中断和状态寄存器	0x00000000
R32_DFSDM_FLT0ICR	0x40017040	DFSDM 滤波器 0 中断标志清零寄存器	0x00000000
R32_DFSDM_FLT0JCHGR	0x40017048	DFSDM 滤波器 0 注入通道组选择寄存器	0x00000001
R32_DFSDM_FLT0FCR3	0x40017050	DFSDM 滤波器 0 控制寄存器 3	0x00000000
R32_DFSDM_FLT0JDATAR	0x40017058	DFSDM 滤波器 0 注入组数据寄存器	0x00000000
R32_DFSDM_FLT0RDATAR	0x40017060	DFSDM 滤波器 0 常规通道数据寄存器	0x00000000
R32_DFSDM_FLT0AWHTR	0x40017068	DFSDM 滤波器 0 模拟看门狗阈值上限寄存器	0x00000000
R32_DFSDM_FLT0AWLTR	0x40017070	DFSDM 滤波器 0 模拟看门狗阈值	0x00000000

		下限寄存器	
R32_DFSDM_FLT0AWSR	0x40017078	DFSDM 滤波器 0 模拟看门狗状态寄存器	0x00000000
R32_DFSDM_FLT0AWCFR	0x40017080	DFSDM 滤波器 0 模拟看门狗清零标志寄存器	0x00000000
R32_DFSDM_FLT0EXMAX	0x40017088	DFSDM 滤波器 0 极值检测器最大值寄存器	0x80000000
R32_DFSDM_FLT0EXMIN	0x40017090	DFSDM 滤波器 0 极值检测器最小值寄存器	0x00000000
R32_DFSDM_FLT0CNVTIMR	0x40017098	DFSDM 滤波器 0 转换定时器寄存器	0x00000000
R32_DFSDM_FLT1CR1	0x4001702C	DFSDM 滤波器 1 控制寄存器 1	0x00000000
R32_DFSDM_FLT1CR2	0x40017034	DFSDM 滤波器 1 控制寄存器 2	0x00000000
R32_DFSDM_FLT1ISR	0x4001703C	DFSDM 滤波器 1 中断和状态寄存器	0x00000000
R32_DFSDM_FLT1ICR	0x40017044	DFSDM 滤波器 1 中断标志清零寄存器	0x00000000
R32_DFSDM_FLT1JCHGR	0x4001704C	DFSDM 滤波器 1 注入通道组选择寄存器	0x00000001
R32_DFSDM_FLT1FCR3	0x40017054	DFSDM 滤波器 1 控制寄存器 3	0x00000000
R32_DFSDM_FLT1JDATAR	0x4001705C	DFSDM 滤波器 1 注入组数据寄存器	0x00000000
R32_DFSDM_FLT1RDATAR	0x40017064	DFSDM 滤波器 1 常规通道数据寄存器	0x00000000
R32_DFSDM_FLT1AWHTR	0x4001706C	DFSDM 滤波器 1 模拟看门狗阈值上限寄存器	0x00000000
R32_DFSDM_FLT1AWLTR	0x40017074	DFSDM 滤波器 1 模拟看门狗阈值下限寄存器	0x00000000
R32_DFSDM_FLT1AWSR	0x4001707C	DFSDM 滤波器 1 模拟看门狗状态寄存器	0x00000000
R32_DFSDM_FLT1AWCFR	0x40017084	DFSDM 滤波器 1 模拟看门狗清零标志寄存器	0x00000000
R32_DFSDM_FLT1EXMAX	0x4001708C	DFSDM 滤波器 1 极值检测器最大值寄存器	0x80000000
R32_DFSDM_FLT1EXMIN	0x40017094	DFSDM 滤波器 1 极值检测器最小值寄存器	0x00000000
R32_DFSDM_FLT1CNVTIMR	0x4001709C	DFSDM 滤波器 1 转换定时器寄存器	0x00000000

42.6.1 DFSDM 通道 y 配置寄存器 1 (R32_DFSDM_CHyCFGR1) (y=0/1)

偏移地址: $0x00 + 0x04 * y$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DFSDM EN	CKOUT SRC	Reserved						CKOUTDIV[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATPACK [1:0]	DATMPX [1:0]	Reserved			CHINS EL	CHE N	CKA BEN	SCD EN	Reser ved	SPICKSEL [1:0]	Rese rved	SI TP			

位	名称	访问	描述	复位值
31	DFSDMEN	RW	DFSDM 接口全局使能： 1：使能 DFSDM 接口； 0：禁止 DFSDM 接口。 如果已使能 DFSDM 接口，则该接口将根据使能的 y 通道和使能的 x 滤波器设置（R32_DFSDM_CHyCFGR1 的 CHEN 位和 R32_DFSDM_FLTxCr1 的 DFEN 位）进行工作。通过设置 DFSDMEN=0 清除数据： - 所有寄存器 R32_DFSDM_FLTxISR 均被设为复位状态（x=0、1）； - 所有寄存器 R32_DFSDM_FLTxAWSR 均被设为复位状态（x=0、1）。 注：DFSDMEN 仅存在于 R32_DFSDM_CH0CFGR1 寄存器（通道 y=0）中。	0
30	CKOUTSRC	RW	输出串行时钟源选择： 1：输出时钟的源来自音频时钟； 0：输出时钟的源来自外设时钟。 只有在（R32_DFSDM_CH0CFGR1 寄存器中）DFSDMEN=0 时，才能修改该值。 注：CKOUTSRC 仅存在于 R32_DFSDM_CH0CFGR1 寄存器（通道 y=0）中。	0
[29:24]	Reserved	R0	保留。	0
[23:16]	CKOUTDIV[7:0]	RW	输出串行时钟分频器： 0：禁止输出时钟生成（CKOUT 信号被设为低电平状态）； 1~255：为 CKOUT 信号定义串行时钟输出的系统时钟分频，范围为 2~256（分频器值=CKOUTDIV+1）。 CKOUTDIV 还定义了时钟缺失检测的阈值。 只有在（R32_DFSDM_CH0CFGR1 寄存器中）DFSDMEN=0 时，才能修改该值。 如果 DFSDMEN=0（R32_DFSDM_CH0CFGR1 寄存器中），则 CKOUT 信号被置为低电平状态（在 DFSDMEN=0 后的一个 DFSDM 时钟周期内执行设置）。 注：CKOUTDIV 仅存在于 R32_DFSDM_CH0CFGR1 寄存器（通道 y=0）中。	0
[15:14]	DATPACK[1:0]	RW	R32_DFSDM_CHyDATINR 寄存器数据封装模式： 00：标准：R32_DFSDM_CHyDATINR 寄存器中的输入数据仅存储在 INDAT0[15:0] 中。要清空 R32_DFSDM_CHyDATINR 寄存器，必须由 DFSDM 滤波	0

			器从通道 y 中读取一个采样。 01: 交错: R32_DFSDM_CHyDATINR 寄存器中的输入数据存储为两个采样: - 第一个采样位于 INDAT0[15:0] 中 (已分配至通道 y); - 第二个采样位于 INDAT1[15:0] 中 (已分配至通道 y)。 要清空 R32_DFSDM_CHyDATINR 寄存器, 必须由数字滤波器从通道 y 中读取两个采样(INDAT0[15:0] 部分被读为第一个采样, 而 INDAT1[15:0] 部分则被读为下一个采样)。 10: 双通道: R32_DFSDM_CHyDATINR 寄存器中的输入数据存储为两个采样: - 第一个采样位于 INDAT0[15:0] 中 (已分配至通道 y); - 第二个样采样于 INDAT1[15:0] 中 (已分配至通道 y+1)。 要清空 R32_DFSDM_CHyDATINR 寄存器, 必须由数字滤波器从通道 y 中读取第一个样本, 且必须由另一数字滤波器从通道 y+1 中读取第二个样本。只有在通道数为偶数 (y=0) 时才支持双通道模式, 对于通道数为奇数 (y=1) 的情况, R32_DFSDM_CHyDATINR 被写保护。如果偶数通道被设为双通道模式, 则下一奇数通道必须进入标准模式 (DATPACK[1:0]=0), 以便与偶数通道正确配合。 11: 保留, 只有在 (R32_DFSDM_CHyCFGR1 寄存器中) CHEN=0 时, 才能修改该值。	
[13:12]	DATMPX[1:0]	RW	通道 y 输入数据复用器: 00: 通道 y 输入数据取自外部串行输入, 为 1 位值。R32_DFSDM_CHyDATINR 寄存器被写保护; 01: 通道 y 输入数据取自内部模数转换器 ADCy+1 输出寄存器更新, 为 16 位值 (若 ADCy+1 存在)。ADC 输出数据被写入 R32_DFSDM_CHyDATINR 寄存器的 INDAT0[15:0] 部分; 10: 通道 y 输入数据取自内部 R32_DFSDM_CHyDATINR 寄存器 (通过 CPU/DMA 直接写操作实现)。可写入一个或两个 16 位数据采样, 具体取决于 DATPACK[1:0] 位域设置; 11: 保留。 只有在 (R32_DFSDM_CHyCFGR1 寄存器中) CHEN=0 时, 才能修改该值。 注: DATMPX[1:0]=1 仅适用于 ADC1 和 ADC2。	0
[11:9]	Reserved	RO	保留。	0
8	CHINSEL	RW	通道输入选择: 1: 通道输入取自下一通道 (通道 (y+1) 取 2 的	0

			模)的引脚; 0: 通道输入取自同一通道 y 的引脚。 只有在 (R32_DFSDM_CHyCFGR1 寄存器中) CHEN=0 时, 才能修改该值。	
7	CHEN	RW	通道 y 使能: 1: 使能通道 y; 0: 禁止通道 y。 如果使能了通道 y, 则会根据给定通道设置开始接收串行数据。	0
6	CKABEN	RW	通道 y 时钟缺失检测器使能: 1: 使能通道 y 时钟缺失检测器; 0: 禁止通道 y 时钟缺失检测器。	0
5	SCDEN	RW	通道 y 短路检测器使能: 1: 输入通道 y 将持续受短路检测器保护; 0: 输入通道 y 将不受短路检测器保护。	0
4	Reserved	RO	保留。	0
[3:2]	SPICKSEL[1:0]	RW	通道 y SPI 时钟选择: 00: 时钟来自外部 CKINy 输入——基于 SITP 的采样点; 01: 时钟来自内部 CKOUT 输出——基于 SITP 的采样点; 10: 时钟来自内部 CKOUT——在 CKOUT 每个第二个下降沿的采样点。 用于连接外部 $\Sigma \Delta$ 调制器, 调制器将时钟输入(来自 CKOUT)除以 2, 来生成输出串行通信时钟(该输出时钟变化在各个时钟输入上升沿有效)。 11: 时钟来自内部 CKOUT 输出——在 CKOUT 每个第二个上升沿的采样点。 用于连接外部 $\Sigma \Delta$ 调制器, 调制器将时钟输入(来自 CKOUT)除以 2, 来生成输出串行通信时钟(该输出时钟变化在各个时钟输入下降沿有效)。 只有在 (R32_DFSDM_CHyCFGR1 寄存器中) CHEN=0 时, 才能修改该值。	0
1	Reserved	RO	保留。	0
0	SITP	RW	通道 y 串行接口类型: 1: 下降沿选通数据 SPI; 0: 上升沿选通数据 SPI。 只有在 (R32_DFSDM_CHyCFGR1 寄存器中的) CHEN=0 时, 才能修改该值。	0

42. 6. 2 DFSDM 通道 y 配置寄存器 2 (R32_DFSDM_CHyCFGR2) (y=0/1)

偏移地址: $0x08 + 0x04 * y$

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

OFFSET[7:0]	DTRBS[4:0]	Reserved

位	名称	访问	描述	复位值
[31:8]	OFFSET[23:0]	RW	通道 y 24 位校准偏移： 对于通道 y 的每一次转换结果，都执行 OFFSET。 该值由软件设置。	0
[7:3]	DTRBS[4:0]	RW	通道 y 数据右移位： 00000-01000 有效，定义积分器输出数据结果的移位——将向右移多少位以获得最终结果。在进行偏移校正之前执行移位。数据移位将结果四舍五入为最接近的整数值。移位结果的符号保留（以便得到有效 24 位有符号格式的结果数据）。 只有在（R32_DFSDM_CHyCFGR1 寄存器中）CHEN=0 时，才能修改该值。 其他：无效。	0
[2:0]	Reserved	R0	保留。	0

42. 6.3 DFSDM 通道 y 模拟看门狗和短路检测器寄存器（R32_DFSDM_CHyAWSCDR）

(y=0/1)

偏移地址：0x10 + 0x04 * x

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

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:22]	AWFORD[1:0]	RW	通道 y 模拟看门狗 Sinc 滤波器阶数： 00: FastSinc 滤波器类型； 01: Sinc ¹ 滤波器类型； 10: Sinc ² 滤波器类型； 11: Sinc ³ 滤波器类型。 只有在（R32_DFSDM_CHyCFGR1 寄存器中）CHEN=0 时，才能修改该位。	0
21	Reserved	R0	保留。	0
[20:16]	AWFOSR[4:0]	RW	通道 y 模拟看门狗滤波器过采样率： 0~31: 定义 Sinc 类型滤波器的长度，长度范围为 1~32（AWFOSR+1）。该数字还表示模拟数据速率的抽取率。 只有在（R32_DFSDM_CHyCFGR1 寄存器中）CHEN=0 时，才能修改该位。 注：如果 AWFOSR=0，则滤波器无影响（滤波器旁	0

			路)。	
[15:14]	Reserved	R0	保留。	0
[13:12]	BKSCD[1:0]	RW	通道 y 短路检测器断路信号分配： 00：断路信号未分配至通道短路检测器； 01：断路 0 信号分配至通道 0 短路检测器； 10：断路 1 信号分配至通道 1 短路检测器； 11：断路信号均分配至对应的通道短路检测器。	0
[11:8]	Reserved	R0	保留。	0
[7:0]	SCDT[7:0]	RW	通道 y 的短路检测器阈值： 这些位可由软件写入，用来定义短路检测器的计数器阈值。如果达到该值，则在给定通道上发生短路检测器事件。	0

42.6.4 DFSDM 通道 y 看门狗滤波器数据寄存器 (R32_DFSDM_CHyWDATR) (y=0/1)

偏移地址：0x18 + 0x04 * y

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
WDATA[15:0]															

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:0]	WDATA[15:0]	R0	输入通道 y 看门狗数据： 由输入通道 y 模拟看门狗滤波器转换的数据。该通道进行连续的数据转换（无触发），转换分辨率受一定限制（OSR=1...32/sinc 阶数=1...3）。	0

42.6.5 DFSDM 通道 y 数据输入寄存器 (R32_DFSDM_CHyDAT1NR) (y=0/1)

偏移地址：0x20 + 0x04 * y

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INDAT1[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDAT0[15:0]															

位	名称	访问	描述	复位值
[31:16]	INDAT1[15:0]	RW	通道 y 或通道 y+1 的输入数据： 当 DATMPX[1:0]=1 或 DATMPX[1:0]=2 时，输入并行通道数据将通过数字滤波器进行处理。数据可由 CPU/DMA 写入（当 DATMPX[1:0]=2 时），也可以由内部 ADC 直接写入（当 DATMPX[1:0]=1 时）。当 DATPACK[1:0]=0（标准模式）时 INDAT0[15:0] 被写保护（不用于输入采样）。	0

			当 DATPACK[1:0]=1（交错模式）时，第二个通道 y 数据采样存储到 INDAT1[15:0]中。第一个通道 y 数据采样存储到 INDAT0[15:0]中。DFSDM_FLTx 滤波器依次读取该两个采样，作为两个通道 y 数据采样。 当 DATPACK[1:0]=2（双模式）时，对于偶数 y 通道：INDAT1[15:0]中的采样自动复制到通道（y+1）的 INDAT0[15:0]中。对于奇数 y 通道：INDAT1[15:0]被写保护。INDAT0[15:1]采用 16 位带符号格式。	
[15:0]	INDAT0[15:0]	RW	通道 y 输入数据： 当 DATMPX[1:0]=1 或 DATMPX[1:0]=2 时，输入并行通道数据将通过数字滤波器进行处理。数据可由 CPU/DMA 写入（当 DATMPX[1:0]=2 时），也可以由内部 ADC 直接写入（当 DATMPX[1:0]=1 时）。 当 DATPACK[1:0]=0（标准模式）时，通道 y 数据采样存储到 INDAT0[15:0]中。 当 DATPACK[1:0]=1（交错模式）时，第一个通道 y 数据采样存储到 INDAT0[15:0]中；第二个通道 y 数据采样存储到 INDAT1[15:0]中。DFSDM_FLTx 滤波器依次读取该两个采样，作为两个通道 y 数据采样。 当 DATPACK[1:0]=2（双模式）时，对于偶数 y 通道：通道 y 数据采样存储到 INDAT0[15:0]中。对于奇数 y 通道：INDAT0[15:0]被写保护。INDAT0[15:0]采用 16 位带符号格式。	0

42.6.6 DFSDM 滤波器 x 控制寄存器 1 (R32_DFSDM_FLTxCR1) (x=0/1)

偏移地址：0x28 + 0x04 * x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	AWFSEL	FAS T	Reserved				RCH	Reserved	RDM AEN	Reserved	RSY NC	RCO NT	RSWST ART	Reserved	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	JEXTEN [1:0]		Reserved	JEXTSEL[3:0]				Reserved	JDM AEN	JSC AN	JSY NC	Reserved	JSWST ART	DFEN	

位	名称	访问	描述	复位值
31	Reserved	R0	保留。	0
30	AWFSEL	RW	模拟看门狗快速模式选择： 1：通道收发器值的模拟看门狗（在看门狗滤波器之后）； 0：数据输出值的模拟看门狗（在数字滤波器之	0

			后)。在偏移校正和移位后进行比较。	
29	FAST	RW	常规转换的快速转换模式选择： 1：使能快速转换模式； 0：禁止快速转换模式。 在连续模式下进行常规转换时，如果使能了快速模式，则每次转换（第一次转换除外）都比在标准模式下的转换执行得快。该位对于不连续的转换没有影响。 只有在 DFEN=0 (R32_DFSDM_FLTxCR1) 时才能修改该位。	0
[28:25]	Reserved	R0	保留。	0
24	RCH	RW	常规通道选择： 1：将通道 1 选作常规通道； 0：将通道 0 选作常规通道。 当 RCIP=1 时写入这些位，并在下一个常规转换开始时生效。这在连续模式下（当 RCONT=1 时）特别有用。它还会影响待处理的常规转换（因正在进行注入转换而暂停）。	0
[23:22]	Reserved	R0	保留。	0
21	RDMAEN	RW	使能 DMA 通道以读取常规转换数据： 1：已使能 DMA 通道来读取常规数据； 0：未使能 DMA 通道来读取常规数据。 只有在 DFEN=0 (R32_DFSDM_FLTxCR1) 时才能修改该位。	0
20	Reserved	R0	保留。	0
19	RSYNC	RW	使用 DFSDM_FLT0 同步启动常规转换： 1：在 DFSDM_FLT0 中启动常规转换的同时，在该 DFSDM_FLTx 中启动常规转换； 0：不使用 DFSDM_FLT0 同步启动常规转换。 只有在 DFEN=0 (R32_DFSDM_FLTxCR1) 时才能修改该位。	0
18	RCONT	RW	常规转换的连续模式选择： 1：在发出每个转换请求后，重复转换常规通道； 0：对于每个转换请求，只转换一次常规通道。 在连续常规转换进行过程中，向该位写入 0 将立即停止连续模式。	0
17	RSWSTART	RW1T	软件启动常规通道转换： 1：写入 1 会发出启动常规通道转换的请求，并会使 RCIP 变为 1。如果已经设置 RCIP=1，则对 RSWSTART 进行的写操作无效。如果 RSYNC=1，则写入 1 无效； 0：写入 0 无影响。 此位始终读为 0。	0
[16:15]	Reserved	R0	保留。	0
[14:13]	JEXTEN[1:0]	RW	注入转换触发使能和触发边沿选择： 00：禁止触发检测；	0

			01：所选触发的每个上升沿都会发出启动注入转换的请求； 10：所选触发的每个下降沿都会发出启动注入转换的请求； 11：所选触发的上升沿和下降沿都会发出启动注入转换的请求。 只有在 DFEN=0（R32_DFSDM_FLTxCR1）时才能修改该位。																			
12	Reserved	R0	保留。	0																		
[11:8]	JEXTSEL[3:0]	RW	启动注入转换的触发信号选择： 0x0~0xF：根据下表选择的触发输入（内部或外部触发）。 只有在 DFEN=0（R32_DFSDM_FLTxCR1）时才能修改该位。 注：同步触发的延迟最大为一个 $f_{DFSDMCLK}$ 时钟周期（存在确定的抖动），异步触发的延迟为 2-3 个 $f_{DFSDMCLK}$ 时钟周期（抖动长达 1 个周期）。 <table><tr><th></th><th>DFSDM_FLT0</th><th>DFSDM_FLT1</th></tr><tr><td>0x00</td><td>dfsdm_jtrg0</td><td>dfsdm_jtrg0</td></tr><tr><td>0x01</td><td>dfsdm_jtrg1</td><td>dfsdm_jtrg1</td></tr><tr><td>...</td><td></td><td></td></tr><tr><td>0xE</td><td>dfsdm_jtrg14</td><td>dfsdm_jtrg14</td></tr><tr><td>0xF</td><td>dfsdm_jtrg15</td><td>dfsdm_jtrg15</td></tr></table>		DFSDM_FLT0	DFSDM_FLT1	0x00	dfsdm_jtrg0	dfsdm_jtrg0	0x01	dfsdm_jtrg1	dfsdm_jtrg1	...			0xE	dfsdm_jtrg14	dfsdm_jtrg14	0xF	dfsdm_jtrg15	dfsdm_jtrg15	0
	DFSDM_FLT0	DFSDM_FLT1																				
0x00	dfsdm_jtrg0	dfsdm_jtrg0																				
0x01	dfsdm_jtrg1	dfsdm_jtrg1																				
...																						
0xE	dfsdm_jtrg14	dfsdm_jtrg14																				
0xF	dfsdm_jtrg15	dfsdm_jtrg15																				
[7:6]	Reserved	R0	保留。	0																		
5	JDMAEN	RW	使能 DMA 通道以读取注入通道组的数据： 1：已使能 DMA 通道来读取注入数据； 0：未使能 DMA 通道来读取注入数据。 只有在 DFEN=0（R32_DFSDM_FLTxCR1）时才能修改该位。	0																		
4	JSCAN	RW	注入转换的扫描转换模式： 1：从选择的最低通道开始，对注入通道组执行连续转换； 0：对注入通道组执行一次通道转换，然后选择该通道组下一个通道。 只有在 DFEN=0（R32_DFSDM_FLTxCR1）时才能修改该位。 如果 JSCAN=0，则对 JCHG 进行写入操作会将通道选择复位为最小所选通道。	0																		

3	JSYNC	RW	使用 DFSDM_FLT0 JSWSTART 触发同步启动注入转换： 1：在 DFSDM_FLT0 中通过 JSWSTART 触发启动注入转换的同时，在该 DFSDM_FLTx 中启动注入转换； 0：不使用 DFSDM_FLT0 同步启动注入转换。 只有在 DFEN=0 (R32_DFSDM_FLTxCR1) 时才能修改该位。	0
2	Reserved	R0	保留。	0
1	JSWSTART	RW1T	启动注入通道组转换： 1：写入 1 会发出转换注入通道组请求，同时会使 JCIP 变为 1。如果已经设置 JCIP=1，则对 JSWSTART 进行的写操作无效。如果 JSYNC=1，则写入 1 无效； 0：写入 0 无影响。 此位始终读为 0	0
0	DFEN	RW	DFSDM_FLTx 使能： 1：使能 DFSDM_FLTx。如果使能了 DFSDM_FLTx，则 DFSDM_FLTx 会根据其设置开始工作； 0：禁止 DFSDM_FLTx。给定 DFSDM_FLTx 的所有转换均立即停止，DFSDM_FLTx 的所有功能均停止。 通过设置 DFEN=0 清零的数据： - 寄存器 R32_DFSDM_FLTxISR 被设为复位状态； - 寄存器 R32_DFSDM_FLTxAWSR 被设为复位状态。	0

42.6.7 DFSDM 滤波器 x 控制寄存器 2 (R32_DFSDM_FLTxCR2) (x=0/1)

偏移地址：0x30 + 0x04 * x

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														AWDCH[1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						EXCH[1:0]		Reserved	CKA BIE	SCD IE	AWD IE	ROV RIE	JOV RIE	REO CIE	JE0 CIE

位	名称	访问	描述	复位值
[31:18]	Reserved	R0	保留。	0
[17:16]	AWDCH[1:0]	RW	模拟看门狗通道选择： 这些位用于选择由模拟看门狗持续监控的输入通道： 00：在通道 0 和通道 1 上禁止模拟看门狗； 01：在通道 0 上使能模拟看门狗； 10：在通道 1 上使能模拟看门狗； 11：在通道 0 和通道 1 上使能模拟看门狗。	0
[15:10]	Reserved	R0	保留。	0
[9:8]	EXCH[1:0]	RW	极值检测器通道选择： 这些位选择极值检测器要采用的输入通道。	0

			00: 极值检测器不接受通道 0 和通道 1 的数据; 01: 极值检测器接受通道 0 的数据; 10: 极值检测器接受通道 1 的数据; 11: 极值检测器接受通道 0 和通道 1 的数据。	
7	Reserved	R0	保留。	0
6	CKABIE	RW	时钟缺失中断使能: 1: 使能检测通道输入时钟缺失中断; 0: 禁止检测通道输入时钟缺失中断。 请参考 R32_DFSDM_FLT _x ISR 中的 CKABF[1:0] 说明。 注: CKABIE 仅存在于 R32_DFSDM_FLT0CR2 寄存器中 (滤波器 x=0)。	0
5	SCDIE	RW	短路检测器中断使能: 1: 使能短路检测器中断; 0: 禁止短路检测器中断。 请参考 R32_DFSDM_FLT _x ISR 中的 SCDF[1:0] 说明。 注: SCDIE 仅存在于 R32_DFSDM_FLT0CR2 寄存器中 (滤波器 x=0)。	0
4	AWDIE	RW	模拟看门狗中断使能: 1: 使能模拟看门狗中断; 0: 禁止模拟看门狗中断。 请参考 R32_DFSDM_FLT _x ISR 中的 AWDF 说明。	0
3	ROVRIE	RW	常规数据溢出中断使能: 1: 使能常规数据溢出中断; 0: 禁止常规数据溢出中断。 请参考 R32_DFSDM_FLT _x ISR 中的 ROVRF 说明。	0
2	JOVRIE	RW	注入数据溢出中断使能: 1: 使能注入数据溢出中断; 0: 禁止注入数据溢出中断。 请参考 R32_DFSDM_FLT _x ISR 中的 JOVRF 说明。	0
1	REOCIE	RW	常规转换结束中断使能: 1: 使能常规转换结束中断; 0: 禁止常规转换结束中断。 请参考 R32_DFSDM_FLT _x ISR 中的 REOCF 说明。	0
0	JEOCIE	RW	注入转换结束中断使能: 1: 使能注入转换结束中断; 0: 禁止注入转换结束中断。 请参考 R32_DFSDM_FLT _x ISR 中的 JEOCF 说明。	0

42. 6. 8 DFSDM 滤波器 x 中断和状态寄存器 (R32_DFSDM_FLT_xISR) (x=0/1)

偏移地址: $0x38 + 0x04 * x$

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

Reserved	RCIP	JCIP	Reserved	AWDF	ROVRF	JOVRF	REOCF	JEOCF

位	名称	访问	描述	复位值
[31:26]	Reserved	R0	保留。	0
[25:24]	SCDF[1:0]	R0	短路检测器标志： 00：通道 0 和通道 1 上均未发生任何短路检测器事件； 01：通道 0 上的短路检测器计数器达到在 R32_DFSDM_CHyAWSCDR 寄存器中编程的值； 10：通道 1 上的短路检测器计数器达到在 R32_DFSDM_CHyAWSCDR 寄存器中编程的值； 11：通道 0 和通道 1 上的短路检测器计数器均达到在 R32_DFSDM_CHyAWSCDR 寄存器中编程的值； 此位将由硬件置 1，该位可由软件使用 R32_DFSDM_FLTxICR 寄存器中的 CLRSCDF 位清零。 当 CHEN[y]=0（给定通道被禁止）时，SCDF 还可以由硬件清零。 注：SCDF[1:0] 仅存在于 R32_DFSDM_FLT0ISR 寄存器中（滤波器 x=0）。	0
[23:18]	Reserved	R0	保留。	0
[17:16]	CKABF[1:0]	RW	时钟缺失标志： 00：通道 0 和通道 1 上均存在时钟信号； 01：通道 0 上不存在时钟信号； 10：通道 1 上不存在时钟信号； 11：通道 0 和通道 1 上均不存在时钟信号。 当检测到通道上不存在时钟信号时，该位由硬件置 1。当 CHEN=0 时（请参见 R32_DFSDM_CHyCFGR1 寄存器），它由硬件保持为时钟缺失状态。当收发器尚未被同步时，它由硬件保持为时钟缺失状态。它可由软件使用 R32_DFSDM_FLTxICR 寄存器中的相应 CLRCKABF 位清零。 注：CKABF[1:0] 仅存在于 R32_DFSDM_FLT0ISR 寄存器中（滤波器 x=0）。	0
15	Reserved	R0	保留。	0
14	RCIP	RW	常规转换正在进行状态： 1：常规通道转换正在进行，或一个常规转换请求正在等待处理； 0：未发出转换常规通道的请求。 当 RCIP=1 时，忽略启动常规转换的请求。	0
13	JCIP	RW	注入转换正在进行状态： 1：注入通道组转换处于正在进行状态（向 JSWSTART 写入了 1）或者一个注入转换请求正在等待处理（由于触发检测）； 0：未发出转换注入通道组的请求（软件和触发器	0

			均未发出请求)。 当 JCIP=1 时, 忽略启动注入转换的请求。	
[12:5]	Reserved	R0	保留。	0
4	AWDF	RW	模拟看门狗: 1: 模拟看门狗模块检测到电压超出在 R32_DFSDM_FLTxAWLTR 或 R32_DFSDM_FLTxAWHTR 寄存器中编程的值; 0: 未发生模拟看门狗事件。 此位将由硬件置 1, 它由软件通过以下方式清零: 将 R32_DFSDM_FLTxAWSR 寄存器中的所有源标志位 AWHTF[1:0] 和 AWLTF[1:0] 清零 (通过向 R32_DFSDM_FLTxAWCFR 寄存器中的清零位写入 1 实现)。	0
3	ROVRF	RW	常规转换溢出标志: 1: 发生了常规转换溢出, 也就是说, 常规转换已经完成, 但 REOCF 仍然为 1。RDATA 不受溢出影响; 0: 未发生常规转换溢出。 此位将由硬件置 1, 该位可由软件使用 R32_DFSDM_FLTxICR 寄存器中的 CLRROVRF 位清零。	0
2	JOVRF	RW	注入转换溢出标志: 1: 发生了注入转换溢出, 也就是说, 注入转换已经完成, 但 JEOCF 仍然为 1。JDATA 不受溢出影响; 0: 未发生注入转换溢出。 此位将由硬件置 1, 该位可由软件使用 R32_DFSDM_FLTxICR 寄存器中的 CLRJOVRF 位清零。	0
1	REOCF	RW	常规转换结束标志: 1: 已完成常规转换并可以读取其数据; 0: 未完成常规转换。 此位将由硬件置 1, 当软件或 DMA 读取 R32_DFSDM_FLTxRDATAR 时, 该位清零。	0
0	JEOCF	RW	注入转换结束标志: 1: 已完成注入转换并可以读取其数据; 0: 未完成注入转换。 此位将由硬件置 1, 当软件或 DMA 读取 R32_DFSDM_FLTxJDATAR 时, 该位清零。	0

注: 对于每个标志位, 可通过将 R32_DFSDM_FLTxCR2 中的相应位置 1 来使能中断。如果调用了中断, 则必须在退出中断服务程序之前将该标志清零。当 DFEN=0 时, R32_DFSDM_FLTxISR 的所有位都会自动复位。

42.6.9 DFSDM 滤波器 x 中断标志清零寄存器 (R32_DFSDM_FLTxICR) (x=0/1)

偏移地址: $0x40 + 0x04 * x$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						CLRSCDF [1:0]		Reserved						CLRCKABF [1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CLRR0 VRF	CLRJO VRF	Reserved	

位	名称	访问	描述	复位值
[31:26]	Reserved	R0	保留。	0
[25:24]	CLRSCDF [1:0]	RW	清零短路检测器标志： 00：写入 0 无影响； 01：通道 0 上短路检测器标志清零； 10：通道 1 上短路检测器标志清零； 11：通道 0 和通道 1 上短路检测器标志清零。 注：CLRSCDF [1:0] 仅存在于 R32_DFSDM_FLT0ICR 寄存器中（滤波器 x=0）。	0
[23:18]	Reserved	R0	保留。	0
[17:16]	CLRCKABF [1:0]	RW	清零时钟缺失标志： 00：写入 0 无影响； 01：通道 0 上时钟缺失标志清零； 10：通道 1 上时钟缺失标志清零； 11：通道 0 和通道 1 上时钟缺失标志清零。 当收发器尚未被同步时，时钟缺失标志会置 1，且无法由 CLRCKABF 清零。 注：CLRCKABF [1:0] 仅存在于 R32_DFSDM_FLT0ICR 寄存器中（滤波器 x=0）。	0
[15:4]	Reserved	R0	保留。	0
3	CLRR0VRF	RW	将常规转换溢出标志清零： 1：写入 1 会将 R32_DFSDM_FLTxISR 寄存器中的 ROVRF 位清零； 0：写入 0 无影响。	0
2	CLRJOVRF	RW	将注入转换溢出标志清零： 1：写入 1 会将 R32_DFSDM_FLTxISR 寄存器中的 JOVRF 位清零； 0：写入 0 无影响。	0
[1:0]	Reserved	R0	保留。	0

注：R32_DFSDM_FLTxICR 的位始终读为 0。

42. 6. 10 DFSDM 滤波器 x 注入通道组选择寄存器（R32_DFSDM_FLTxJCHGR）（x=0/1）

偏移地址：0x48 + 0x04 * x

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	JCHG[1:0]

位	名称	访问	描述	复位值
[31:2]	Reserved	R0	保留。	0
[1:0]	JCHG[1:0]	RW	注入通道组选择： 00：通道 0 和通道 1 均不属于注入组； 01：通道 0 属于注入组； 10：通道 1 属于注入组； 11：通道 0 和通道 1 均属于注入组。 如果 JSCAN=1，则依次转换每一个所选通道。首先转换最小通道（通道 0，如果已选择），最后转换最大所选通道。 如果 JSCAN=0，则只转换其中一个所选通道，然后通道选择移至下一个通道。如果 JSCAN=0，则对 JCHG 进行写入操作会将通道选择复位为最小所选通道。 必须始终至少为注入组选择一个通道。如果写操作导致所有 JCHG 位均变为零，则忽略写操作。	01b

42. 6. 11 DFSDM 滤波器 x 控制寄存器 3 (R32_DFSDM_FLTxFCR3) (x=0/1)

偏移地址：0x50 + 0x04 * x

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

位	名称	访问	描述	复位值
[31:29]	FORD[2:0]	R0	Sinc 滤波器阶数： 000：FastSinc 滤波器类型； 001：Sinc ¹ 滤波器类型； 010：Sinc ² 滤波器类型； 011：Sinc ³ 滤波器类型； 100：Sinc ⁴ 滤波器类型； 101：Sinc ⁵ 滤波器类型； 110、111：保留。 只有在 DFEN=0 (R32_DFSDM_FLTxCR1) 时才能修改该位。	0
[28:26]	Reserved	R0	保留。	0
[25:16]	FOSR[9:0]	RW	Sinc 滤波器过采样率： 0~1023：定义 Sinc 类型滤波器的长度，长度范围为 1~1024 (FOSR=FOSR[9:0]+1)。该数字还表示滤波器输出数据速率的抽取率。 只有在 DFEN=0 (R32_DFSDM_FLTxCR1) 时才能修改该位	0

			注：如果 $FOSR=0$ ，则滤波器无影响（滤波器旁路）。	
[15:4]	Reserved	R0	保留。	0
[3:0]	IOSR[3:0]	RW	积分器过采样率为 2^{IOSR} 。定义一个积分器输出数据采样是由多少个 Sinc 滤波器采样而合成的。积分器的输出数据速率将被减去该数值（额外数据抽取率）。 只有在 $DFEN=0$ ($R32\_DFSDM\_FLTxCR1$) 时才能修改该位。 注：（1）如果 $IOSR=0$ ，则积分器无影响（积分器旁路）。 （2）在连续非快速模式下， $IOSR$ 需大于 0。	0

42.6.12 DFSDM 滤波器 x 注入组数据寄存器 (R32_DFSDM_FLTxDATAR) (x=0/1)

偏移地址：0x58 + 0x04 * x

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

位	名称	访问	描述	复位值
[31:8]	JDATA[23:0]	R0	注入组转换数据： 每次转换完注入组中的一个通道时，得到的数据都会存储在该字段中。JEOCF=1 时数据有效。读取该寄存器会将相应的 JEOCF 清零。	0
[7:1]	Reserved	R0	保留。	0
0	JDATA[0]	R0	最近转换的注入通道： 每次转换完注入组中的一个通道时，都会更新 JDATA[0] 以指示转换了哪一个通道。因此，JDATA[23:0] 保持的数据对应于 JDATA[0] 所指示的通道。	0

注：可以使用 DMA 从该寄存器中读取数据。可以使用半字访问只读取转换数据的最高有效位 (MSB)。读取该寄存器还会将 $R32\_DFSDM\_FLT_xISR$ 的 JEOCF 清零。因此，如果为从该寄存器中读取数据而激活了 DMA，则固件不能读取该寄存器。

42.6.13 DFSDM 滤波器 x 常规通道数据寄存器 (R32_DFSDM_FLTxDATAR) (x=0/1)

偏移地址：0x60 + 0x04 * x

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

				H

位	名称	访问	描述	复位值
[31:8]	RDATA[23:0]	R0	常规通道转换数据： 每次完成常规转换时，相应的数据都会存储在该寄存器中。当 REOCF=1 时，数据有效。读取该寄存器会将相应的 REOCF 清零。	0
[7:5]	Reserved	R0	保留。	0
4	RPEND	R0	常规通道待处理数据： 因在转换期间发生注入通道触发而导致 RDATA[23:0] 中的常规数据被延迟处理。	0
[3:1]	Reserved	R0	保留。	0
0	RDATAACH	R0	最近转换的常规通道： 每次完成常规转换时，都会更新 RDATAACH 以指示转换了哪一个通道（因为在常规转换期间，可更新 R32_DFSDM_FLTxCR1 寄存器中的常规通道选择 RCH）。因此，RDATA[23:0] 保持的数据对应于 RDATAACH 所指示的通道。	0

注：可以使用半字访问只读取转换数据的最高有效位（MSB）。读取该寄存器还会将 R32_DFSDM_FLTxISR 的 REOCF 清零。

42. 6. 14 DFSDM 滤波器 x 模拟看门狗阈值上限寄存器（R32_DFSDM_FLTxAWHTR）（x=0/1）

偏移地址：0x68 + 0x04 * x

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

位	名称	访问	描述	复位值
[31:8]	AWHT[23:0]	RW	模拟看门狗阈值上限： 这些位由软件写入，用于定义模拟看门狗的阈值上限。	0
[7:2]	Reserved	R0	保留。	0
[1:0]	BKAHW[1:0]	RW	模拟看门狗阈值上限事件的断路信号分配： 00：断路信号不分配到模拟看门狗阈值上限事件； 01：断路 0 信号分配到模拟看门狗阈值上限事件； 10：断路 1 信号分配到模拟看门狗阈值上限事件； 11：断路信号均分配到模拟看门狗阈值上限事件。	0

42. 6. 15 DFSDM 滤波器 x 模拟看门狗阈值下限寄存器（R32_DFSDM_FLTxAWLTR）（x=0/1）

偏移地址：0x70 + 0x04 * x

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

AWLT[23:8]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWLT[7:0]								Reserved					BKAWL[1:0]		

位	名称	访问	描述	复位值
[31:8]	AWLT[23:0]	RW	模拟看门狗阈值下限： 这些位由软件写入，用于定义模拟看门狗的阈值下限。 注：对于通道收发器监视（AWFSEL=1），仅高 16 位（AWLT[23:8]）定义 16 位阈值，该阈值与模拟看门狗滤波器输出进行比较（因为来自模拟看门狗滤波器的数据的分辨率最高为 16 位）。在这种情况下，不对位 AWLT[7:0] 进行比较。	0
[7:2]	Reserved	R0	保留。	0
[1:0]	BKAWL[1:0]	RW	模拟看门狗阈值下限事件的断路信号分配： 00：断路信号不分配到模拟看门狗阈值下限事件； 01：断路 0 信号分配到模拟看门狗阈值下限事件； 10：断路 1 信号分配到模拟看门狗阈值下限事件； 11：断路信号均分配到模拟看门狗阈值下限事件。	0

42.6.16 DFSDM 滤波器 x 模拟看门狗状态寄存器（R32_DFSDM_FLTxAWSR）（x=0/1）

偏移地址：0x78 + 0x04 * x

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						AWHTF[1:0]		Reserved					AWLTF[1:0]		

位	名称	访问	描述	复位值
[31:10]	Reserved	R0	保留。	0
[9:8]	AWHTF[1:0]	R0	模拟看门狗阈值上限标志： 00：通道 0 和通道 1 上均不存在阈值上限错误； 01：通道 0 上存在阈值上限错误； 10：通道 1 上存在阈值上限错误； 11：通道 0 和通道 1 上均存在阈值上限错误。 该位由硬件置 1。该位可由软件使用 R32_DFSDM_FLTxAWCFR 寄存器中的相应 CLRAWHTF 位清零。	0
[7:2]	Reserved	R0	保留。	0
[1:0]	AWLTF[1:0]	R0	模拟看门狗阈值下限标志： 00：通道 0 和通道 1 上均不存在阈值下限错误； 01：通道 0 上存在阈值下限错误； 10：通道 1 上存在阈值下限错误；	0

			11: 通道 0 和通道 1 上均存在阈值下限错误。 该位由硬件置 1。该位可由软件使用 R32_DFSDM_FLTxAWSR 寄存器中的相应 CLRAWLTF 位清零。	

注: DFEN=0 时, R32_DFSDM_FLTxAWSR 的所有位都会自动复位。

42. 6. 17 DFSDM 滤波器 x 模拟看门狗清零标志寄存器 (R32_DFSDM_FLTxAWCFR) (x=0/1)

偏移地址: $0x80 + 0x04 * x$

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						CLRAWHTF [1:0]		Reserved						CLRAWLTF [1:0]	

位	名称	访问	描述	复位值
[31:10]	Reserved	R0	保留。	0
[9:8]	CLRAWHTF[1:0]	RW1Z	清零模拟看门狗阈值上限标志: 00: 写入 0 无影响; 01: 通道 0 阈值上限错误标志清零; 10: 通道 1 阈值上限错误标志清零; 11: 通道 0 和通道 1 阈值上限错误标志清零。	0
[7:2]	Reserved	R0	保留。	0
[1:0]	CLRAWLTF[1:0]	RW1Z	清零模拟看门狗阈值下限标志: 00: 写入 0 无影响; 01: 通道 0 阈值下限错误标志清零; 10: 通道 1 阈值下限错误标志清零; 11: 通道 0 和通道 1 阈值下限错误标志清零。	0

42. 6. 18 DFSDM 滤波器 x 极值检测器最大值寄存器 (R32_DFSDM_FLTxEVMAX) (x=0/1)

偏移地址: $0x88 + 0x04 * x$

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

位	名称	访问	描述	复位值
[31:8]	EXMAX[23:0]	RZ	极值检测器最大值: 这些位由硬件置 1, 用于指示 DFSDM_FLTx 所转换 的最大值。通过读取该寄存器将 EXMAX[23:0] 位复 位为值 (0x800000)。	0
[7:1]	Reserved	R0	保留。	0

0	EXMAXCH	RZ	极值检测器最大值数据通道： 这些位包含的信息涉及哪一个通道的数据被存储到 EXMAX[23:0]。可通过读取该寄存器将这些位清零。	0

42. 6. 19 DFSDM 滤波器 x 极值检测器最小值寄存器 (R32_DFSDM_FLT_xEXMIN) (x=0/1)

偏移地址：0x90 + 0x04 * x

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

位	名称	访问	描述	复位值
[31:8]	EXMIN[23:0]	R0	极值检测器最小值： 这些位由硬件置 1，用于指示 DFSDM_FLT _x 所转换的最小值。通过读取该寄存器将 EXMIN[23:0] 位复位为值 (0x7FFFFFFF)。	0
[7:1]	Reserved	R0	保留。	0
0	EXMINCH	R0	极值检测器最小值数据通道： 这些位包含的信息涉及哪一个通道的数据被存储到 EXMIN[23:0]。可通过读取该寄存器将这些位清零。	0

42. 6. 20 DFSDM 滤波器 x 转换定时器寄存器 (R32_DFSDM_FLT_xCNVTIMR) (x=0/1)

偏移地址：0x98 + 0x04 * x

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

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:0]	CNVCNT[27:0]	R0	28 位定时器计数转换时间： CNVCNT = (CNVCNT[27:0] + 1) $t = \text{CNVCNT} / f_{\text{DFSDMCLK}}$; 定时器的输入时钟来自 DFSDM 时钟 (系统时钟 f_{DFSDMCLK})。转换时间测量始于每次转换开始，并止于每 36 次转换结束 (即第一次和最后一次串行采样之间的间隔)。只有在滤波器旁路 (FOSR[9:0]=0) 的情况下，转换时间测量才会停止，且 CNVCNT[27:0]=0。时间计时如下：	0

		如果 FAST=0 (或者 FAST=1 连续模式下的第一次转换)，则： $t = [FOSR * (2^{IOSR} + FORD + 1)] / f_{CKIN}$ (适用于 Sinc^x 滤波器)； $t = [FOSR * (2^{IOSR} + 5)] / f_{CKIN}$ (适用于 FastSinc 滤波器)； 如果在连续模式下 FAST=1 (第一次转换除外)，则： $t = [FOSR * 2^{IOSR}] / f_{CKIN}$； 如果 FOSR=FOSR[9:0]+1=1 (滤波器旁路，仅积分器有效)，则： CNVCNT=0 (时间计时停止，转换时间：$t = 2^{IOSR} / f_{CKIN}$)； 其中，f_{CKIN} 为 (给定通道 CKIN_y 引脚上的) 通道输入时钟频率；在并行数据输入 (来自内部 ADC 或来自 CPU/DMA 写操作) 的情况下，其表示输入数据速率。 注：当转换被中断时 (例如通过禁止/使能所选通道)，定时器也将该中断时间计算在内。	

第 43 章 LCD-TFT 显示控制器 (LTDC)

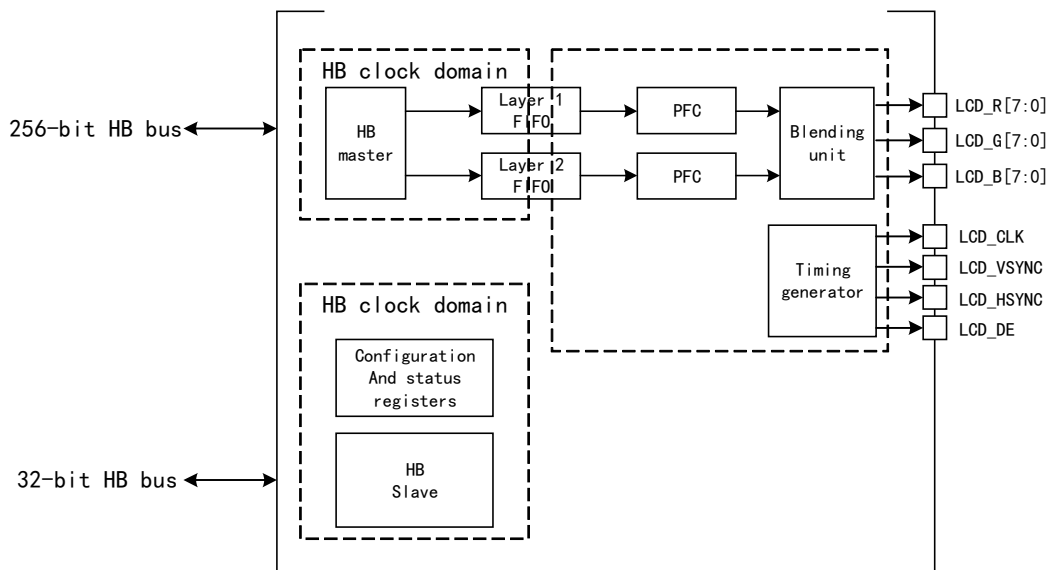
LCD-TFT (薄膜晶体管-液晶屏显示器) 显示控制器 (LTDC) 主要提供并行数字 RGB 以及水平同步、垂直同步、像素时钟和数据使能的信号, 可作为输出信号到不同 LCD 和 TFT 面板接口。

43.1 主要特征

- 提供 2 个显示层, 包含了专有的 8*256 位 FIFO
- 支持查色表 (CLUT), 每个图层高达 256 种颜色
- 支持可编程不同显示面板的时序
- 支持可编程背景色
- 支持可编程 HSYNC, VSYNC 和数据使能信号的极性
- 每个显示层可选择高达 8 个输入颜色的格式
- 支持使用 alpha 值 (每像素或常数) 的两层之间进行灵活的混合
- 支持色键 (透明颜色)
- 支持可编程窗口位置和大小
- 支持薄膜晶体管 (TFT) 彩色显示器
- 支持高达 3 个可编程中断时间

43.2 概述

图 43-1 LTDC 结构框图



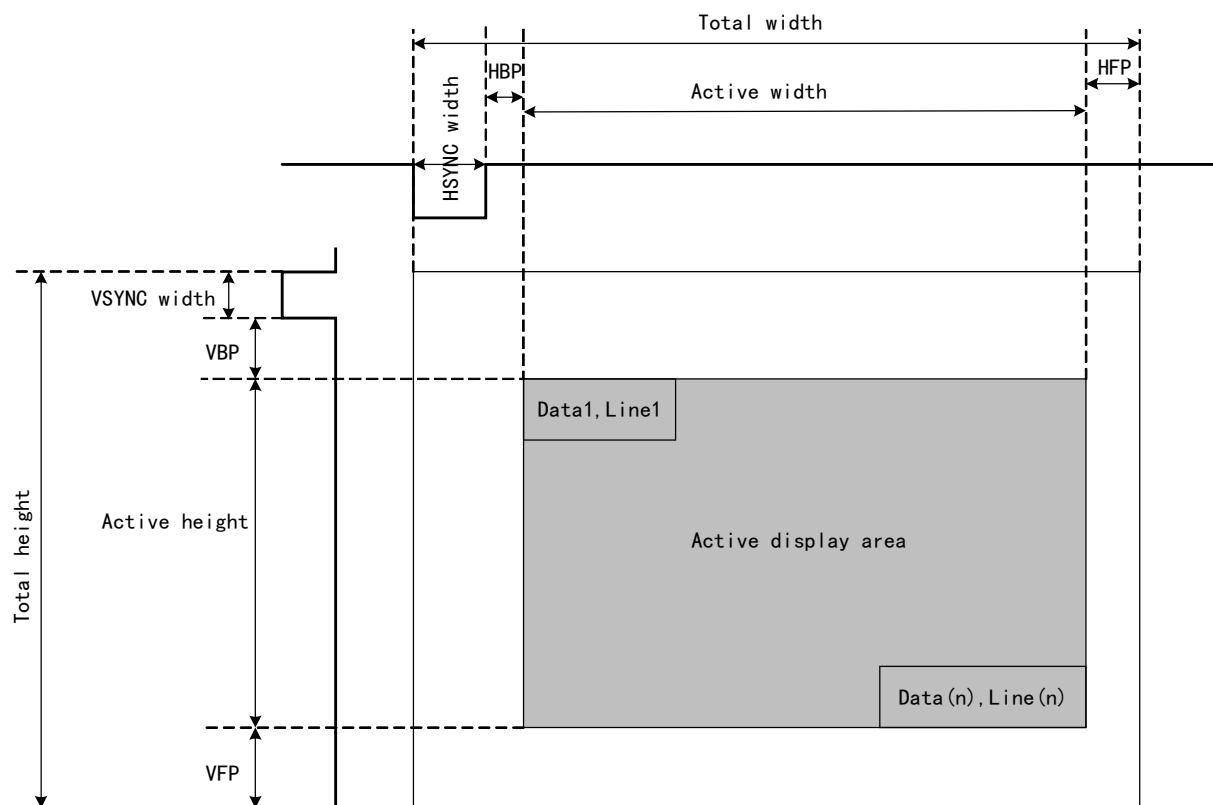
- (1) PFC: pixel format converter, performing the pixel format conversion from the selected input pixel format of a layer to words.
(2) ltcd_ker_ck: pixel clock.

43.3 功能描述

43.3.1 LTDC 同步时序

LTDC 接口是一个同步数据接口, 包括像素时钟, 像素数据以及水平和垂直同步信号。图 43-2 展示了 LTDC 同步时序发生器模块生成的可配置时序参数。

图 43-2 LTDC 同步时序



注：（1）HSYNC 和 VSYNC 宽度分别为水平和垂直同步宽度；HBP 和 HFP 分别为水平后沿周期和水平前沿周期；VBP 和 VFP 分别为垂直后沿周期和垂直前沿周期。

(2) 使能 LTDC 时, 产生的时序以 X/Y=0/0 位置作为垂直同步区域中的第一个水平同步像素, 随后是后沿、有效数据显示区域和前沿。

(3) 禁止 LTDC 时, 时序发生器模块复位为 $X=\text{总宽度}-1$ 、 $Y=\text{总高度}-1$, 并在垂直同步阶段和 FIFO 刷新前保持上一个像素。因此, 仅连续输出消隐数据。

同步时序配置示例如下所示：

- 1) LCD-TFT 时序（应从面板数据手册中提取）；
- 2) 水平和垂直同步宽度：0xA 像素，0x2-1 行；
- 3) 水平和垂直后沿：0x14 像素，0x2 行；
- 4) 有效宽度和有效高度：0x140 像素，0xF0 行（320x240）；
- 5) 水平前沿：0xA 像素；
- 6) 垂直前沿：0x4 行；

LTDC 时序寄存器中编程的值将为:

- 1) R32_LTDC_SSCR 寄存器：将编程为 0x00090001。（HSW[11:0]为 0xA-1 且 VSH[10:0]为 0x2-1）；
- 2) R32_LTDC_BPCR 寄存器：将编程为 0x001D0003。（AHBP[11:0]为 0x1D (0xA + 0x14-1)，AVBP[10:0]为 0x3 (0x2 + 0x2-1)）；
- 3) R32_LTDC_AWCR 寄存器：将编程为 0x015D00F3；（AAW[11:0]为 0x15D (0xA + 0x14 + 0x140-1)，AAH[10:0]为 0xF3 (0x2 + 0x2 + 0xF0-1)）；
- 4) R32_LTDC_TWCR 寄存器：将编程为 0x00000167。（TOTALW[11:0]为 0x167 (0xA + 0x14 + 0x140 + 0xA-1)）；
- 5) R32_LTDC_TWCR 寄存器：将编程为 0x000000F7。（TOTALH[10:0]为 0xF7 (0x2 + 0x2 + 0xF0 + 0x4-1)）。

43.3.2 像素格式

可编程像素格式用于层的帧缓冲区中存储的数据。可通过 R32_LTDC_LxPFCR 寄存器为每个层配置多达 8 个输入像素格式。

ARGB8888 格式要求每通道 (Alpha、Red、Green 和 Blue) 有 8 位数据。但是 ARGB1555 和 ARGB4444 格式的某些通道是少于 8 位的。像素处理单元通过拷贝高位填充到低位的方式, 将其转换成 ARGB8888。当处理 RGB888 和 RGB565 格式时, 像素处理单元假设 Alpha=255, 并且如果通道的位数少于 8, 也将拷贝高位填充到低位。

43.3.3 查色表 (CLUT)

可在运行时通过 R32_LTDC_LxCR 寄存器 CLUTEN 位为每个层使能 CLUT, CLUT 仅在使用 L8、AL44 和 AL88 输入像素格式时适用于索引色。

首先, CLUT 必须加载用于替换相应像素 (索引色) 的原始 R、G 和 B 值的 R、G 和 B 值。每个颜色 (RGB 值) 在 CLUT 内都有自己对应的地址。R、G 和 B 值及其各自的地址均通过 R32_LTDC_LxCLUTWR 寄存器编程。

在使用 L8 和 AL88 输入像素格式时, CLUT 必须加载 256 个颜色。各颜色的地址在 R32_LTDC_LxCLUTWR 寄存器 CLUTADD 位中配置。

在使用 AL44 输入像素格式时, CLUT 必须仅加载 16 个颜色。各颜色的地址必须通过将 4 位 L 通道重复为 8 位进行填充, 具体如下:

- L0 (索引色 0), 地址 0x00 处;
- L1, 地址 0x11 处;
- L2, 地址 0x22 处;
-;
- L15, 地址 0xFF 处。

43.3.4 颜色帧缓冲区

43.3.4.1 颜色帧缓冲区地址

每个层的颜色帧缓冲区均有一个起始地址, 该地址通过 R32_LTDC_LxCFBAR 寄存器进行配置。当使能某个层时, 将从颜色帧缓冲区中获取该数据。

43.3.4.2 颜色帧缓冲区长度

每层均设置颜色帧缓冲区的总行长 (单位为字节) 和行数, 二者可分别通过 R32_LTDC_LxCFBLR 和 R32_LTDC_LxCFBLNR 寄存器进行配置。

行长和行数的设置用于阻止超出帧缓存结尾的数据被预取到层对应的 FIFO 中。

如果设置为低于所需字节, 则会产生 FIFO 下溢中断 (如果之前已使能)。

如果设置为高于实际所需字节, 则将丢弃从 FIFO 中读取的无用数据且不会显示。

43.3.4.3 颜色帧缓冲区间距

每层的颜色帧缓冲区均具有可配置间距, 此间距是一行的开始与下一行开始的距离 (以字节为单位)。它通过 R32_LTDC_LxCFBLR 寄存器配置。

43.3.5 层窗口和混合功能

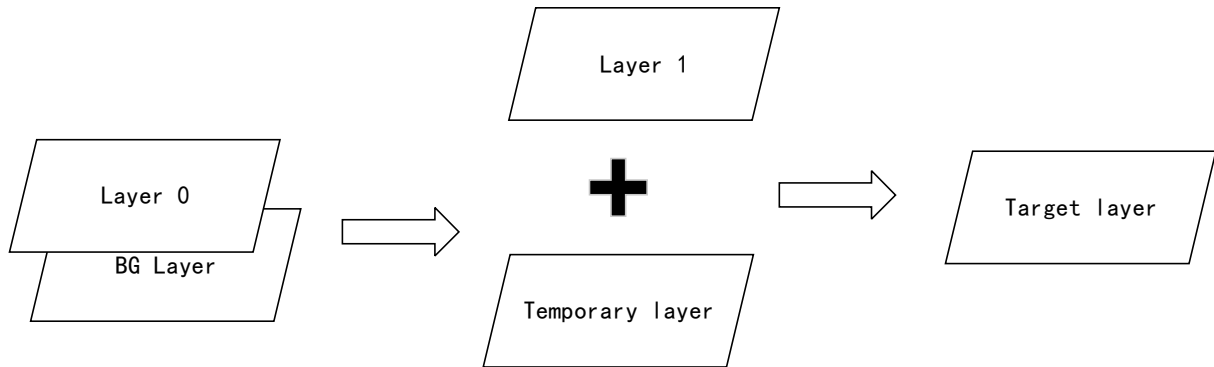
每层都支持窗口功能以及两层的混合功能。LTDC 首先执行每层的窗口操作, 然后将两层混合成一帧。

窗口功能定义了一个显示窗口, 每层有独立的窗口位置配置寄存器 R32_LTDC_LxWHPCR 和 R32_LTDC_LxWVPCR。窗口配置定义了一层内的显示窗。窗口内的像素将保持它的原值, 但是窗口外的

像素值将被 R32_LTDC_LxDCCR 寄存器定义的像素值替代。

混合顺序固定，即由下至上。混合单元首先混合层 0 和背景层得到一个临时层，然后混合层 1 和临时层，得到目的层。背景层的 ARGB 值在 R32_LTDC_BCCR 寄存器定义。如果某层被禁止，混合功能使用该层的默认颜色。若要层禁止时不显示默认颜色，需将 R32_LTDC_LxBFCR 寄存器中此层的混合系数设置为其复位值。混合过程如图 43-3 所示。

图 43-3 混合过程框图



通用混合公式： $BC = BF1 * C + BF2 * CS$;

BC 为混合的颜色；BF1 为混合因子 1；C 为当前层颜色；BF2 为混合因子 2；CS 下一级层混合的颜色。

其中，当前像素的混合因子有两种取值，由寄存器配置。一种是像素 Alpha 乘以常数 Alpha，另一种是常数 Alpha。

注：常数 Alpha 值是在寄存器中编程的值，由硬件实现 255 等分。

例如：仅使能第一层，BF1 配置为常数 Alpha，BF2 配置为 1-常数 Alpha

常数 Alpha：在 LxCACR 寄存器中编程的常数 Alpha 为 240 (0xF0)。

因此，常数 Alpha 值为 $240/255=0.94$

C：当前层红色通道为 0x80

Cs：背景红色通道为 0x30

第一层与背景色混合。

红色通道最终结果 = 常数 Alpha * C + (1-常数 Alpha) * Cs

= $0.94 * 0x80 + (1 - 0.94) * 0x30 = 0x7B$ 。

绿色和蓝色通道同理。

43.3.6 色键

色键 (RGB) 可配置为代表透明像素。运行时，可配置色键值并用其替换像素 RGB 值。

使能色键后，当前像素 (格式转换后、CLUT 分别混合前的像素) 将与色键进行比较。如果当前像素与编程的 RGB 值相匹配，则该像素的所有通道 (ARGB) 均设置为 0。

色键通过 LTDC_LxCKCR 寄存器配置。编程值取决于像素格式，因为它会在像素格式转换为 ARGB888 后与当前像素进行比较。

例如，将中黄色 (50%红 + 50%绿) 用作透明色键，则：

- 在 RGB565 中，中黄色为 0x8400。将 LTDC_LxCKCR 设为 0x848200。

- 在 ARGB8888 中，中黄色为 0x808000，将 LTDC_LxCKCR 设为 0x808000。

- 在所有基于 CLUT 的颜色模式 (L8、AL88、AL44) 下，将其中一个调色板条目设为中黄色 0x808000，将 LTDC_LxCKCR 设为 0x808000。

43.3.7 重载影子寄存器

一些配置寄存器执行影子操作。对活动寄存器执行写操作时，或在 R32_LTDC_SRCR 寄存器配置阶段之后的垂直消隐周期开始时，可将影子寄存器值立即重载到活动寄存器中。如果选择了立即重载配置，则只应在所有新寄存器完成写操作后激活重载。

不应在重载完成前再次修改影子寄存器。读取影子寄存器将返回实际有效值。新写入的值只能在重载发生后读取。

如果在 R32_LTDC_IER 寄存器中相应使能，则可产生寄存器重载中断。

影子寄存器均为第 1 层和第 2 层寄存器，但 R32_LTDC_LxCLUTWR 寄存器除外。

43.3.8 中断

LTDC 提供 3 个可屏蔽中断，这些中断经逻辑或运算到 1 个中断向量。

中断源可通过 R32_LTDC_IER 寄存器单独使能或禁止；通过 R32_LTDC_ISR 寄存器可读取相应中断标志；通过将 R32_LTDC_ICR 寄存器置 1 可清除相应的中断标志位。

发生如下事件时会产生中断：

- 行中断：达到被编程的行时产生。行中断的位置在 R32_LTDC_LIPCR 寄存器中编程；
- 寄存器重载中断：在垂直消隐周期内执行影子寄存器重载时产生；
- FIFO 下溢中断：从空的层 FIFO 中请求像素时产生。

43.4 寄存器描述

表 38-1 LTDC 相关寄存器列表

名称	访问地址	描述	复位值
R32_LTDC_SSCR	0x40014800	LTDC 同步大小配置寄存器	0x00000000
R32_LTDC_BPCR	0x40014804	LTDC 后沿配置寄存器	0x00000000
R32_LTDC_AWCR	0x40014808	LTDC 有效宽度配置寄存器	0x00000000
R32_LTDC_TWCR	0x4001480C	LTDC 总宽度配置寄存器	0x00000000
R32_LTDC_GCR	0x40014810	LTDC 全局控制寄存器	0x00002220
R32_LTDC_SRCR	0x40014814	LTDC 影子重载配置寄存器	0x00000000
R32_LTDC_BCCR	0x40014818	LTDC 背景色配置寄存器	0x00000000
R32_LTDC_IER	0x4001481C	LTDC 中断使能寄存器	0x00000000
R32_LTDC_ISR	0x40014820	LTDC 中断状态寄存器	0x00000000
R32_LTDC_ICR	0x40014824	LTDC 中断清零寄存器	0x00000000
R32_LTDC_LIPCR	0x40014828	LTDC 行中断位置配置寄存器	0x00000000
R32_LTDC_CPSR	0x4001482C	LTDC 当前位置状态寄存器	0x00000000
R32_LTDC_CDSR	0x40014830	LTDC 当前显示状态寄存器	0x0000000F
R32_LTDC_L1CR	0x40014834	LTDC 第 1 层控制寄存器	0x00000000
R32_LTDC_L1WHPCR	0x40014838	LTDC 第 1 层窗口水平位置配置寄存器	0x00000000
R32_LTDC_L1WVPCR	0x4001483C	LTDC 第 1 层窗口垂直位置配置寄存器	0x00000000
R32_LTDC_L1CKCR	0x40014840	LTDC 第 1 层色键配置寄存器	0x00000000
R32_LTDC_L1PFCR	0x40014844	LTDC 第 1 层像素格式配置寄存器	0x00000000
R32_LTDC_L1CACR	0x40014848	LTDC 第 1 层常数 Alpha 配置寄存器	0x00000000
R32_LTDC_L1DCCR	0x4001484C	LTDC 第 1 层默认颜色配置寄存器	0x00000000
R32_LTDC_L1BFGR	0x40014850	LTDC 第 1 层混合系数配置寄存器	0x00000600

R32_LTDC_L1CFBAR	0x40014854	LTDC 第 1 层颜色帧缓冲区地址寄存器	0x00000000
R32_LTDC_L1CFBLR	0x40014858	LTDC 第 1 层颜色帧缓冲区长度寄存器	0x00000000
R32_LTDC_L1CFBLNR	0x4001485C	LTDC 第 1 层颜色帧缓冲区行数寄存器	0x00000000
R32_LTDC_L1CLUTWR	0x40014860	LTDC 第 1 层 CLUT 写寄存器	0x00000000
R32_LTDC_L2CR	0x40014864	LTDC 第 2 层控制寄存器	0x00000000
R32_LTDC_L2WHPCR	0x40014868	LTDC 第 2 层窗口水平位置配置寄存器	0x00000000
R32_LTDC_L2WVPCR	0x4001486C	LTDC 第 2 层窗口垂直位置配置寄存器	0x00000000
R32_LTDC_L2CKCR	0x40014870	LTDC 第 2 层色键配置寄存器	0x00000000
R32_LTDC_L2PFCR	0x40014874	LTDC 第 2 层像素格式配置寄存器	0x00000000
R32_LTDC_L2CACR	0x40014878	LTDC 第 2 层常数 Alpha 配置寄存器	0x00000000
R32_LTDC_L2DCCR	0x4001487C	LTDC 第 2 层默认颜色配置寄存器	0x00000000
R32_LTDC_L2BFCR	0x40014880	LTDC 第 2 层混合系数配置寄存器	0x00000600
R32_LTDC_L2CFBAR	0x40014884	LTDC 第 2 层颜色帧缓冲区地址寄存器	0x00000000
R32_LTDC_L2CFBLR	0x40014888	LTDC 第 2 层颜色帧缓冲区长度寄存器	0x00000000
R32_LTDC_L2CFBLNR	0x4001488C	LTDC 第 2 层颜色帧缓冲区行数寄存器	0x00000000
R32_LTDC_L2CLUTWR	0x40014890	LTDC 第 2 层 CLUT 写寄存器	0x00000000

43.4.1 LTDC 同步大小配置寄存器 (R32_LTDC_SSCR)

偏移地址: 0x00

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

位	名称	访问	描述	复位值
[31:28]	Reserved	RO	保留。	0
[27:16]	HSW[11:0]	RW	水平同步宽度（以像素时钟周期为单位）： 此域定义水平同步像素数减 1。	0
[15:11]	Reserved	RO	保留。	0
[10:0]	VSH[10:0]	RW	垂直同步高度（以水平扫描行为单位）： 此域定义垂直同步高度减 1。它表示水平同步行数。	0

43.4.2 LTDC 后沿配置寄存器 (R32_LTDC_BPCR)

偏移地址: 0x04

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

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:16]	AHBP[11:0]	RW	累加水平后沿（以像素时钟周期为单位）： 此域定义累加水平后沿宽度（水平同步像素加水平后沿像素减1）。 水平后沿是水平同步信号变为无效到下一扫描行的有效显示开始之间的间隔。	0
[15:11]	Reserved	R0	保留。	0
[10:0]	AVBP[10:0]	RW	累加垂直后沿（以水平扫描行为单位）： 此域定义累加垂直后沿宽度（垂直同步行加垂直后沿行减1）。 垂直后沿是帧开始到下一帧的首个有效扫描行开始所包含的水平扫描行的数量。	0

43.4.3 LTDC 有效宽度配置寄存器（R32_LTDC_AWCR）

偏移地址：0x08

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

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:16]	AAW[11:0]	RW	累加有效宽度（以像素时钟周期为单位）： 此域定义累加有效宽度（水平同步像素加水平后沿像素加有效像素减1）。 有效宽度是面板扫描行的有效显示区中的像素数。	0
[15:11]	Reserved	R0	保留。	0
[10:0]	AAH[10:0]	RW	累加有效高度（以水平扫描行为单位）： 此域定义累加高度（垂直同步行加垂直后沿行加有效高度行减1）。有效高度是面板中的有效行数。	0

43.4.4 LTDC 总宽度配置寄存器（R32_LTDC_TWCR）

偏移地址：0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TOTALW[11:0]											

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

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:16]	TOTALW[11:0]	RW	总宽度（以像素时钟周期为单位）： 此域定义累加总宽度（水平同步像素加水平后沿像素加有效宽度加水平前沿像素减1）。	0
[15:11]	Reserved	R0	保留。	0
[10:0]	TOTALH[10:0]	RW	总高度（以水平扫描行为单位）： 此域定义累加高度（垂直同步行加垂直后沿行加有效高度加垂直前沿行减1）。	0

43.4.5 LTDC 全局控制寄存器（R32_LTDC_GCR）

偏移地址：0x10

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

位	名称	访问	描述	复位值
31	HSPOL	RW	水平同步极性： 1：水平同步极性高电平有效； 0：水平同步极性低电平有效。	0
30	VSPOL	RW	垂直同步极性： 1：垂直同步高电平有效； 0：垂直同步低电平有效。	0
29	DEPOL	RW	非数据使能极性： 1：非数据使能极性高电平有效； 0：非数据使能极性低电平有效。	0
28	PCPOL	RW	像素时钟极性： 1：像素时钟高电平有效； 0：像素时钟极性低电平有效。	0
[27:1]	Reserved	R0	保留。	0x444
0	LTDCEN	RW	LCD-TFT 控制器使能位： 1：使能； 0：禁止。	0

43.4.6 LTDC 影子重载配置寄存器（R32_LTDC_SRCR）

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														VBR	IMR

位	名称	访问	描述	复位值
[31:2]	Reserved	R0	保留。	0
1	VBR	RW	垂直消隐重载： 1：影子寄存器在垂直消隐周期（有效显示区后的第一行开始时）内重载； 0：无影响。 注：此位由软件置1，只有重载后才由硬件清零。	0
0	IMR	RW	立即重载： 1：影子寄存器立即重载； 0：无影响。 注：此位由软件置1，只有重载后才由硬件清零。	0

注：影子寄存器回读有效值时，直至重载完成，才会读取旧值。

43.4.7 LTDC 背景色配置寄存器（R32_LTDC_BCCR）

偏移地址：0x18

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

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:16]	BCRED[7:0]	RW	背景红色值。	0
[15:8]	BCGREEN[7:0]	RW	背景绿色值。	0
[7:0]	BCBLUE[7:0]	RW	背景蓝色值。	0

43.4.8 LTDC 中断使能寄存器（R32_LTDC_IER）

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												RRIE	Reserved	FUIE	LIE

位	名称	访问	描述	复位值
[31:4]	Reserved	R0	保留。	0
3	RRIE	RW	寄存器重载中断使能： 1：使能； 0：禁止。	0

2	Reserved	R0	保留。	0
1	FUIE	RW	FIFO 下溢中断使能： 1：使能； 0：禁止。	0
0	LIE	RW	行中断使能： 1：使能； 0：禁止。	0

43.4.9 LTDC 中断状态寄存器 (R32_LTDC_ISR)

偏移地址：0x20

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												RRIF	Reserved	FUIF	LIF

位	名称	访问	描述	复位值
[31:4]	Reserved	R0	保留。	0
3	RRIF	R0	寄存器重载中断标志： 1：发生垂直消隐重载时（以及到达有效区域后的第一行时）产生寄存器重载中断； 0：未产生寄存器重载中断。	0
2	Reserved	R0	保留。	0
1	FUIF	R0	FIFO 下溢中断标志： 1：当其中一个层 FIFO 为空并从 FIFO 读取像素数据时，将产生 FIFO 下溢中断； 0：未产生 FIFO 下溢中断。	0
0	LIF	R0	行中断标志： 1：到达编程的行时产生行中断； 0：未产生行中断。	0

43.4.10 LTDC 中断清零寄存器 (R32_LTDC_ICR)

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												CRRIF	Reserved	CFUIF	CLIF

位	名称	访问	描述	复位值
[31:4]	Reserved	R0	保留。	0
3	CRRIF	RW1Z	寄存器重载中断清零标志，写 1 清 0，写 0 无效。	0
2	Reserved	R0	保留。	0

1	CFUIF	RW1Z	FIFO 下溢中断清零标志, 写 1 清 0, 写 0 无效。	0
0	CLIF	RW1Z	行中断清零标志, 写 1 清 0, 写 0 无效。	0

43.4.11 LTDC 行中断位置配置寄存器 (R32_LTDC_LIPCR)

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						LIP0S[10:0]									

位	名称	访问	描述	复位值
[31:4]	Reserved	R0	保留。	0
[10:0]	LIP0S[10:0]	RW	行中断位置。	0

43.4.12 LTDC 当前位置状态寄存器 (R32_LTDC_CPSR)

偏移地址: 0x2C

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

位	名称	访问	描述	复位值
[31:16]	CXPOS[15:0]	RW	当前 X 位置。	0
[15:0]	CYPOS[15:0]	RW	当前 Y 位置。	0

43.4.13 LTDC 当前显示状态寄存器 (R32_LTDC_CDSR)

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												HSYNCS	VSYNCS	HDES	VDES

位	名称	访问	描述	复位值
[31:4]	Reserved	R0	保留。	0
3	HSYNCS	R0	水平同步显示状态: 1: 高电平有效; 0: 低电平有效。	1
2	VSYNCS	R0	垂直同步显示状态: 1: 高电平有效; 0: 低电平有效。	1
1	HDES	R0	水平数据使能显示状态:	1

			1: 高电平有效; 0: 低电平有效。	
0	VDES	R0	垂直数据使能显示状态: 1: 高电平有效; 0: 低电平有效。	1

43.4.14 LTDC 第 x 层控制寄存器 (R32_LTDC_LxCR) (x=1/2)

偏移地址: $0x34 + 0x30 \times (x-1)$

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											CLUTEN	Reserved	COLKEN	LEN	

位	名称	访问	描述	复位值
[31:5]	Reserved	R0	保留。	0
4	CLUTEN	RW	查色表使能: 1: 使能; 0: 禁止。 注: CLUT 仅对 L8、AL44 和 AL88 像素格式有意义。	0
[3:2]	Reserved	R0	保留。	0
1	COLKEN	RW	色键使能: 1: 使能; 0: 禁止。	0
0	LEN	RW	层使能: 1: 使能; 0: 禁止。	0

43.4.15 LTDC 第 x 层窗口水平位置配置寄存器 (R32_LTDC_LxWHPCR) (x=1/2)

偏移地址: $0x38 + 0x30 \times (x-1)$

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

位	名称	访问	描述	复位值
[31:28]	Reserved	R0	保留。	0
[27:16]	WHSPPOS[11:0]	RW	窗口水平结束位置: $WHSPPOS[11:0] \geq AHBP[11:0] + 1$ 。	0
[15:12]	Reserved	R0	保留。	0
[11:0]	WHSTPOS[11:0]	RW	窗口水平起始位置: $WHSTPOS[11:0] \leq AAW[11:0]$ 。	0

例如, R32_LTDC_BPCR 寄存器配置为 0x000E0005 (AHBP[11:0] 为 0xE), R32_LTDC_AWCR 寄存器配置

为 0x028E01E5 (AAW[11:0] 为 0x28E)。要配置大小为 630x460 的窗口的水平位置 (有效数据区域中的水平起始偏移为 5 个像素)。

- 1) 层窗口的第一个像素: WHSTPOS[11:0] 应编程为 0x14 (0xE+1+0x5) ;
- 2) 层窗口的最后一个像素: WHSPPOS[11:0] 应编程为 0x28A。

43.4.16 LTDC 第 x 层窗口垂直位置配置寄存器 (R32_LTDC_LxWVPCR) (x=1/2)

偏移地址: 0x3C + 0x30*(x-1)

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

位	名称	访问	描述	复位值
[31:27]	Reserved	R0	保留。	0
[26:16]	WVSPPOS[10:0]	RW	窗口垂直结束位置: WVSPPOS[10:0] >= AVBP[10:0] + 1。	0
[15:11]	Reserved	R0	保留。	0
[10:0]	WVSTPOS[10:0]	RW	窗口垂直起始位置: WVSTPOS[10:0] <= AAW[10:0]。	0

例如, R32_LTDC_BPCR 寄存器配置为 0x000E0005 (AVBP[10:0] 为 0x5), R32_LTDC_AWCR 寄存器配置为 0x028E01E5 (AAH[10:0] 为 0x1E5)。要配置大小为 630x460 的窗口的垂直位置 (有效数据区域中的垂直起始偏移为 8 行) :

- 1) 层窗口的第一行: WVSTPOS[10:0] 应编程为 0xE (0x5 + 1 + 0x8) ;
- 2) 层窗口的最后一行: WVSPPOS[10:0] 应编程为 0x1DA。

43.4.17 LTDC 第 x 层色键配置寄存器 (R32_LTDC_LxCKCR) (x=1/2)

偏移地址: 0x40 + 0x30*(x-1)

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

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:16]	CKRED[7:0]	RW	色键红色值。	0
[15:8]	CKGREEN[7:0]	R0	色键绿色值。	0
[7:0]	CKBLUE[7:0]	RW	色键蓝色值。	0

43.4.18 LTDC 第 x 层像素格式配置寄存器 (R32_LTDC_LxPFCR) (x=1/2)

偏移地址: 0x44 + 0x30*(x-1)

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													PF[2:0]		

位	名称	访问	描述	复位值
[31:3]	Reserved	R0	保留。	0
[2:0]	PF[2:0]	RW	像素格式： 000: ARGB8888; 001: RGB888; 010: RGB565; 011: ARGB1555; 100: ARGB4444; 101: L8 (8 位 Luminance) ; 110: AL44 (4 位 Alpha, 4 位 Luminance) ; 111: AL88 (8 位 Alpha, 8 位 Luminance) 。	0

43.4.19 LTDC 第 x 层常数 Alpha 配置寄存器 (R32_LTDC_LxCACR) (x=1/2)

偏移地址: $0x48 + 0x30 \times (x-1)$

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								CONSTA[7:0]							

位	名称	访问	描述	复位值
[31:8]	Reserved	R0	保留。	0
[7:0]	CONSTA[7:0]	RW	常数 Alpha: 此域配置混合时使用的常数 Alpha。常数 Alpha 由硬件实现 255 等分。 示例: 如果编程的常数 Alpha 为 0xFF, 则常数 Alpha 值为 $255/255=1$ 。	0xFF

43.4.20 LTDC 第 x 层默认颜色配置寄存器 (R32_LTDC_LxDCCR) (x=1/2)

偏移地址: $0x4C + 0x30 \times (x-1)$

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

位	名称	访问	描述	复位值
[31:24]	DCALPHA[7:0]	RW	默认 Alpha 值。	0
[23:16]	DCRED[7:0]	RW	默认颜色红色。	0
[15:8]	DCGREEN[7:0]	RW	默认颜色绿色。	0
[7:0]	DCBLUE[7:0]	RW	默认颜色蓝色。	0

43.4.21 LTDC 第 x 层混合系数配置寄存器 (R32_LTDC_LxBFGR) (x=1/2)偏移地址: $0x50 + 0x30 \times (x-1)$

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					BF1[2:0]			Reserved					BF2[2:0]		

位	名称	访问	描述	复位值
[31:11]	Reserved	R0	保留。	0
[10:8]	BF1[2:0]	RW	混合系数 1: 100: 常数 Alpha; 110: 像素 Alpha * 常数 Alpha; 其他: 保留。	110b
[7:3]	Reserved	R0	保留。	0
[2:0]	BF2[2:0]	RW	混合系数 2: 101: 1-常数 Alpha; 111: 1-像素 Alpha * 常数 Alpha; 其他: 保留。	0

43.4.22 LTDC 第 x 层颜色帧缓冲区地址寄存器 (R32_LTDC_LxCFBAR) (x=1/2)偏移地址: $0x54 + 0x30 \times (x-1)$

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

位	名称	访问	描述	复位值
[31:0]	CFBADD[31:0]	RW	颜色帧缓冲区起始地址。	0

43.4.23 LTDC 第 x 层颜色帧缓冲区长度寄存器 (R32_LTDC_LxCFBLR) (x=1/2)偏移地址: $0x58 + 0x30 \times (x-1)$

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

位	名称	访问	描述	复位值
[31:29]	Reserved	R0	保留。	0
[28:16]	CFBP[12:0]	RW	颜色帧缓冲区间距 (以字节为单位): 此域定义从像素某行的起始处到下一行的起始处	0

			的增量。	
[15:13]	Reserved	R0	保留。	0
[12:0]	CFBLN[12:0]	RW	颜色帧缓冲区行长： 定义一行像素的长度（以字节为单位）+ 31。 行长的计算方法为：有效宽度 × 每像素的字节数 + 31。	0

43. 4. 24 LTDC 第 x 层颜色帧缓冲区行数寄存器（R32_LTDC_LxCFBLNR）（x=1/2）

偏移地址：0x5C + 0x30*(x-1)

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						CFBLNBR[10:0]									

位	名称	访问	描述	复位值
[31:12]	Reserved	R0	保留。	0
[10:0]	CFBLNBR[10:0]	RW	帧缓冲区行数： 此域定义缓冲区中的行数，行数跟有效高度对应。	0

注：行数和行长设置定义针对每层从每个帧中获取的数据量。如果配置为低于所需字节，则会产生 FIFO 下溢中断（如果使能）。另一方面，起始地址和间距设置定义存储器中每行的正确起始位置。

43. 4. 25 LTDC 第 x 层 CLUT 写寄存器（R32_LTDC_LxCLUTWR）（x=1/2）

偏移地址：0x60 + 0x30*(x-1)

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

位	名称	访问	描述	复位值
[31:24]	CLUTADD[7:0]	WO	配置每个 RGB 值的 CLUT 地址。	0
[23:16]	RED[7:0]	WO	红色值。	0
[15:8]	GREEN[7:0]	WO	绿色值。	0
[7:0]	BLUE[7:0]	WO	蓝色值。	0

注：（1）CLUT 写寄存器只应在消隐周期内或在层禁止时配置。

（2）CLUT 可通过 R32_LTDC_LxCR 寄存器使能和禁止。

（3）CLUT 仅对 L8、AL44 和 AL88 像素格式有意义。

第 44 章 图形处理硬件加速器（GPHA）

图像处理硬件加速器（GPHA）是一种专门用于图像处理的 DMA，支持索引颜色模式或直接颜色模式，支持所有经典颜色编码方案。此外，GPHA 模块还带有自身专有的 CLUT（颜色查找表）。

44.1 主要特征

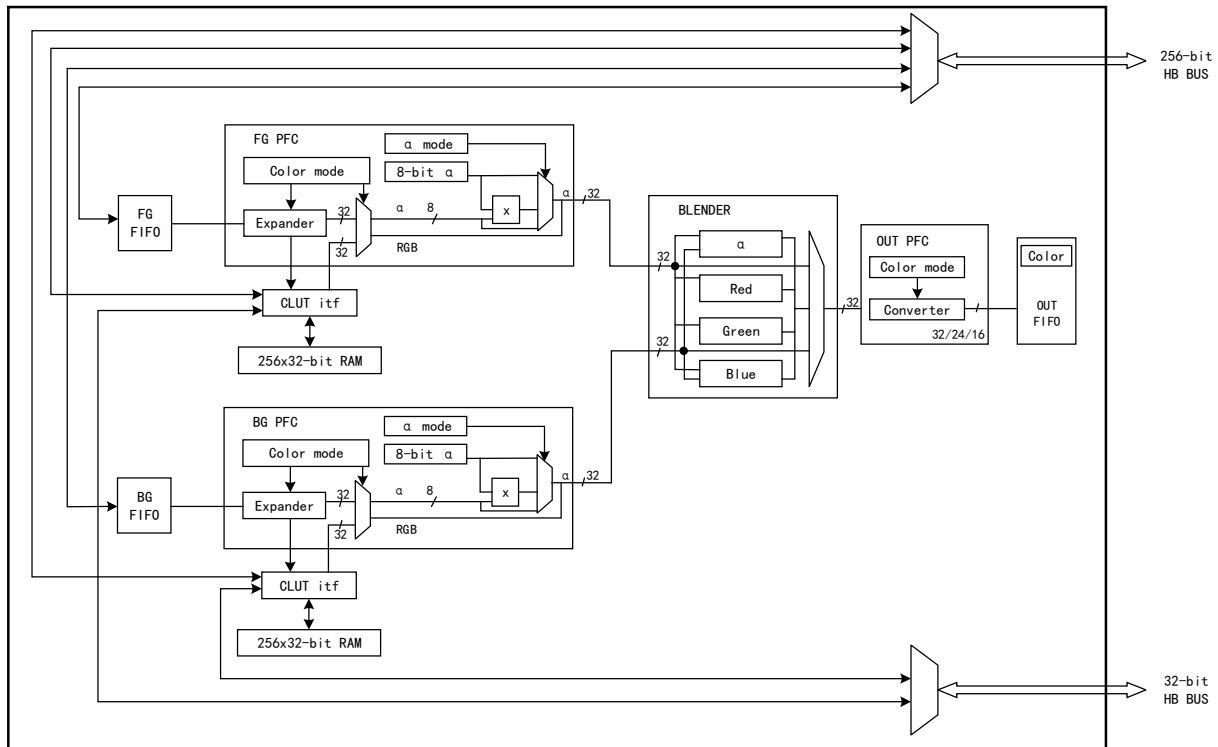
- 支持单 HB 主设备总线架构
- HB 从设备编程接口支持 8/16/32 位访问（除 32 位的 CLUT 访问外）
- 支持可编程源区域和目标区域的大小和偏移
- 支持可编程整个存储空间的源地址和目标地址
- 支持 Alpha 值可调（源值、固定值、调制值）
- 支持高达 2 个源的混合操作
- 支持可编程源颜色格式和目标颜色格式，支持多达 11 种颜色格式且每像素最高达 32 位
- 在间接颜色模式下，支持 2 个内部存储器用于存储 CLUT
- 支持可编程 CLUT 大小，可通过 CPU 自动加载 CLUT 或编程 CLUT
- 支持内部定时器控制 HB 的带宽
- 支持 3 种工作模式：寄存器到存储器、存储器到存储器且支持像素格式转换、存储器到存储器且支持像素格式转换和混合
- 支持对目标图像的指定部分或全部区域填充特定颜色
- 支持源图像的部分或全部内容复制到目标图像对应的部分或全部中
- 支持源图像的部分或全部内容至目标图像的部分或全部内容的像素格式转换复制
- 支持将像素格式不同的两个源图像部分和/或全部混合，再将结果复制到颜色格式不同的部分或整个目标图像中
- 支持中止、挂起 GPHA 操作
- 支持在出现总线错误或者访问冲突时生成中断
- 支持在流程完成时生成中断

44.2 功能描述

44.2.1 概述

GPHA 前景层（FG）FIFO 和背景层（BG）FIFO 获取待复制以及待处理的输入数据，这些 FIFO 根据相应像素格式转换器（PFC）中预设的颜色格式获取像素，进行颜色格式转换，其中前景层和背景层的可单独处理，也可以混合处理，最后输出转换后的颜色数据。如图 44-1 所示，其详细的展示了 GPHA 的结构框图。

图 44-1 GPHA 结构框图



GPHA 控制器执行直接存储器传输，它作为一个 HB 主设备可以控制 HB 总线矩阵来启动 HB 事务。HB 从设备端口用于编程 GPHA 控制器。

GPHA 支持以下 3 种模式下工作：

- 寄存器到存储器；
- 存储器到存储器并执行像素格式转换；
- 存储器到存储器并执行像素格式转换和混合。

GPHA 功能包括以下操作：

- 支持对目标图像的指定部分或全部区域填充特定颜色；
- 支持源图像的部分或全部内容复制到目标图像对应的部分或全部中；
- 支持源图像的部分或全部内容至目标图像的部分或全部内容的像素格式转换复制；
- 支持将像素格式不同的两个源图像部分和/或全部混合，再将结果复制到颜色格式不同的部分或整个目标图像中。

44.2.2 GPHA 控制

通过 R32_GPHA_CTLR 寄存器可配置 GPHA 控制器，可以执行以下操作：

- 1) 选择 GPHA 工作模式；
- 2) 使能/禁止 GPHA 中断；
- 3) 启动/挂起/中止进行中的数据传送。

44.2.3 GPHA 前景层 FIFO 和背景层 FIFO

GPHA 前景层（FG）FIFO 和背景层（BG）FIFO 的协同工作是 GPHA 高效图像处理的关键，它们获取待复制以及待处理的输入数据，这些 FIFO 会根据相应像素格式转换器（PFC）中预设的颜色格式获取像素。可通过以下寄存器对它们进行编程：

- GPHA 前景层存储器地址寄存器（R32_GPHA_FGMAR）
- GPHA 前景层偏移寄存器（R32_GPHA_FGOR）

- GPHA 背景层存储器地址寄存器 (R32_GPHA_BGMAR)
- GPHA 背景层偏移寄存器 (R32_GPHA_BGBOR)
- GPHA 行数寄存器 (行数和每行像素数) (R32_GPHA_NLR)

GPHA 工作在寄存器到存储器模式下，不激活任何 FIFO。GPHA 工作在存储器到存储器模式下并支持像素格式转换时（无混合操作），不会激活 BGFIFO。

44. 2. 4 GPHA 前景层和背景层像素格式转换器 (PFC)

GPHA 前景层和背景层像素格式转换器 (PFC) 负责执行像素格式转换，以生成每像素 32 位的值。此外，PFC 还具备修改 alpha 通道的能力。

该转换器在第一阶段转换颜色格式。通过 R32_GPHA_FGPFCCR 寄存器的 CM[3:0] 位和 R32_GPHA_BGPFCCR 寄存器的 CM[3:0] 位，分别对前景层像素和背景层像素的原始颜色格式进行配置。其输入时支持的颜色模式由表 44-1 所示。

表 44-1 输入时支持的颜色模式

CM[3:0]	颜色模式
0000	ARGB8888
0001	RGB888
0010	RGB565
0011	ARGB1555
0100	ARGB4444
0101	L8
0110	AL44
0111	AL88
1000	L4
1001	A8
1010	A4

Alpha 通道是图像中用于表示每个像素透明度的一个额外通道。它常与红色、绿色、蓝色 (RGB) 通道一起使用，形成 RGBA 模型。

以下为表 44-1 中出现的颜色格式的编码方式的说明：

Alpha 值字段：透明，0xFF 值对应不透明像素，0x00 对应透明像素；

R 字段：代表红色；

G 字段：代表绿色；

B 字段：代表蓝色；

L 字段：代表亮度，该字段是 CLUT 的索引，用于检索三个或四个 RGB/ARGB 分量。

若原始格式为直接颜色模式，则通过将 MSB 复制到 LSB 扩展为每通道 8 位。这可以确保转换具有良好的线性。

若原始格式不包括 alpha 通道，则会自动将 alpha 值设为 0xFF（不透明）。

若原始格式为间接颜色模式，则需要使用 CLUT，并且每个像素格式转换器与一个 256 个 32 位条目的 CLUT 相关联。

对于特定的 alpha 模式 A4 和 A8，既不存储颜色信息，也不编制索引。用于生成图像的颜色是固定的，并且在 R32_GPHA_FGCOLR 寄存器中定义前景层像素的颜色，在 R32_GPHA_BGCOLR 寄存器中定义背景层像素的颜色。

系统存储器中的字段顺序如表 44-2 所示。

表 44-2 存储器中的数据顺序

颜色模式	@+3	@+2	@+1	@+0
ARGB8888	A ₀ [7:0]	R ₀ [7:0]	G ₀ [7:0]	B ₀ [7:0]
RGB888	B ₁ [7:0]	R ₀ [7:0]	G ₀ [7:0]	B ₀ [7:0]
	G ₂ [7:0]	B ₂ [7:0]	R ₁ [7:0]	G ₁ [7:0]
	R ₃ [7:0]	G ₃ [7:0]	B ₃ [7:0]	R ₂ [7:0]
RGB565	R ₁ [4:0]G ₁ [5:3]	G ₁ [2:0]B ₁ [4:0]	R ₀ [4:0]G ₀ [5:3]	G ₀ [2:0]B ₀ [4:0]
ARGB1555	A ₁ [0]R ₁ [4:0]G ₁ [4:3]	G ₁ [2:0]B ₁ [4:0]	A ₀ [0]R ₀ [4:0]G ₀ [4:3]	G ₀ [2:0]B ₀ [4:0]
ARGB4444	A ₁ [3:0]R ₁ [3:0]	G ₁ [3:0]B ₁ [3:0]	A ₀ [3:0]R ₀ [3:0]	G ₀ [3:0]B ₀ [3:0]
L8	L ₃ [7:0]	L ₂ [7:0]	L ₁ [7:0]	L ₀ [7:0]
AL44	A ₃ [3:0]L ₃ [3:0]	A ₂ [3:0]L ₂ [3:0]	A ₁ [3:0]L ₁ [3:0]	A ₀ [3:0]L ₀ [3:0]
AL88	A ₁ [7:0]	L ₁ [7:0]	A ₀ [7:0]	L ₀ [7:0]
L4	L ₇ [3:0]L ₆ [3:0]	L ₅ [3:0]L ₄ [3:0]	L ₃ [3:0]L ₂ [3:0]	L ₁ [3:0]L ₀ [3:0]
A8	A ₃ [7:0]	A ₂ [7:0]	A ₁ [7:0]	A ₀ [7:0]
A4	A ₇ [3:0]A ₆ [3:0]	A ₅ [3:0]A ₄ [3:0]	A ₃ [3:0]A ₂ [3:0]	A ₁ [3:0]A ₀ [3:0]

采用 ARGB8888 模式支持按 32 位对齐 24 位 RGB888，在生成 32 位值后，可根据 R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR 寄存器 AM[1:0]位，对 alpha 通道进行调整，具体配置详见表 44-3 所示。

表 44-3 Alpha 模式配置

AM[1:0]	Alpha 模式
00	不做调整
01	替换为 R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR 寄存器 ALPHA[7:0]值
10	替换为原始 alpha 值与 R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR 寄存器 ALPHA[7:0]值除以 255 所得商的乘积。
11	保留

注：为实现备用格式的支持，需将 R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR 寄存器 AI 位设置为 1，以便对即将输入 alpha 的值进行取反操作。此操作亦适用于 R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR 寄存器和 CLUT 中存储的 Alpha 值。此外，还可通过将 R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR 寄存器 RBS 位设置为 1，来实现 R 字段和 B 字段的互换。此操作亦适用于 CLUT 以及 R32_GPHA_FGCOLR/R32_GPHA_BGCOLOR 寄存器中使用的 RGB 顺序。

44.2.5 GPHA 前景层 FIFO 和背景层 CLUT 接口

CLUT 是一个用于存储颜色映射的表，它将每个像素的索引值映射到具体的颜色值。CLUT 接口可管理对 CLUT 存储器的访问以及 CLUT 的自动加载。CLUT 接口支持以下访问方式：

- PFC 在像素格式转换期间读取 CLUT；
- CPU 通过 HB 从设备端口访问 CLUT，进行数据的读取或写入；
- 执行自动加载 CLUT 时，通过 HB 主设备端口进行 CLUT 写入。

CLUT 存储器加载可通过自动加载和手动加载的方式实现：

自动加载

- 1) 将 CLUT 地址编程到 R32_GPHA_FGCMAR 寄存器（前景层 CLUT）或 R32_GPHA_BGCMAR 寄存器（背

景层 CLUT)。

2) 将 CLUT 大小编程到 R32_GPHA_FGPFCCR 寄存器 (前景层 CLUT) 或 R32_GPHA_BGPFCCR 寄存器 (背景层 CLUT) 的 CS[7:0]位。

3) 将 R32_GPHA_FGPFCCR 寄存器 (前景层 CLUT) 或 R32_GPHA_BGPFCCR 寄存器 (背景层 CLUT) 的 START 位设置为 1, 以启动传输。在自动加载期间, CPU 不得访问 CLUT。若出现冲突, 且 R32_GPHA_CTLR 寄存器 CAEIE 位被置 1, 则发生 CLUT 访问错误中断。

手动加载

应用程序必须通过 GPHA HB 从设备端口手动编程, 将本地 CLUT 存储器映射到的 CLUT。

1) 将 CLUT 像素地址编程到 R32_GPHA_FGCRWS 寄存器 FG_CLUT_INDEX 位 (前景层 CLUT) 或 R32_GPHA_FGCRWS 寄存器 BG_CLUT_INDEX 位 (背景层 CLUT)。

2) 将 CLUT 像素值编程到 R32_GPHA_FGCDAT 寄存器 (前景层 CLUT) 或 R32_GPHA_BGCDAT 寄存器 (背景层 CLUT), CPU 自动将该像素值加载至对应的 CLUT 存储器。

CLUT 格式支持 24 位或 32 位。通过 R32_GPHA_FGPFCCR 寄存器 (前景层 CLUT) 或 R32_GPHA_BGPFCCR 寄存器 (背景层 CLUT) 的 CCM 位, 可配置 CLUT 颜色模式, 具体配置方式详见表 44-4。

表 44-4 支持的 CLUT 颜色模式

CCM	CLUT 颜色模式
0	32 位 ARGB8888
1	24 位 RGB888

表 44-5 给出了系统存储器中 CLUT 数据的组织方式。

表 44-5 存储器中的 CLUT 数据顺序

CLUT 颜色模式	@+3	@+2	@+1	@+0
ARGB8888	A ₀ [7:0]	R ₀ [7:0]	G ₀ [7:0]	B ₀ [7:0]
RGB888	B ₁ [7:0]	R ₀ [7:0]	G ₀ [7:0]	B ₀ [7:0]
	G ₂ [7:0]	B ₂ [7:0]	R ₁ [7:0]	G ₁ [7:0]
	R ₃ [7:0]	G ₃ [7:0]	B ₃ [7:0]	R ₂ [7:0]

44.2.6 GPHA 混合器

GPHA 混合器成对混合源像素以计算结果像素。

混合将按以下公式执行:

$$C_{OUT} = \frac{C_{FG}\alpha_{FG} + C_{BG}\alpha_{BG} - C_{BG}\alpha_{Mult}}{\alpha_{OUT}} \quad (\text{商将向下取整});$$

其中 C=R 或 G 或 B,

$$\alpha_{Mult} = \frac{\alpha_{FG}\alpha_{BG}}{255}$$

$$\alpha_{OUT} = \alpha_{FG} + \alpha_{BG} - \alpha_{Mult}$$

混合器不需要任何配置寄存器。是否使用混合器取决于 R32_GPHA_CTLR 寄存器的 MODE[1:0]位中定义的 GPHA 工作模式。

44.2.7 GPHA 输出 PFC

输出 PFC 将像素格式从 32 位转换为指定的输出格式, 输出格式可通过 R32_GPHA_OPFCCR 寄存器 CM[2:0]位进行配置。

表 44-6 输出时支持的颜色模式

CM[2:0]	颜色模式

000	ARGB8888
001	RGB888
010	RGB565
011	ARGB1555
100	ARGB4444

注：为实现备用格式的支持，需将 R32_GPHA_OPFCCR 寄存器 AI 位设置为 1，以便对即将计算 alpha 的值进行取反操作。此操作亦适用于 R32_GPHA_OPFCCR 寄存器中使用的 Alpha 值。此外，还可通过将 R32_GPHA_OPFCCR 寄存器 RBS 位设置为 1，来实现 R 字段和 B 字段的互换。此操作亦适用于 R32_GPHA_OCOLR 寄存器中使用的 RGB 顺序。

44.2.8 GPHA 输出 FIFO

输出 FIFO 根据输出 PFC 中定义的颜色格式对像素进行编程。可通过以下寄存器定义目标区域：

- GPHA 输出存储器地址寄存器 (R32_GPHA_OMAR)
- GPHA 输出偏移寄存器 (R32_GPHA_OOR)
- GPHA 行数寄存器 (行数和每行像素数) (R32_GPHA_NLR)

若 GPHA 工作在寄存器到存储器模式下，则配置的输出矩形将以 GPHA 输出颜色寄存器 (R32_GPHA_OCOLR) 中指定的颜色填充，该寄存器中包含固定的 32 位、24 位或 16 位值。颜色格式的选择则可通过 R32_GPHA_OPFCCR 寄存器 CM[2:0] 位来配置。

系统存储器中的字段顺序如表 44-7 所示。

表 44-7 存储器中的数据顺序

颜色模式	@+3	@+2	@+1	@+0
ARGB8888	A ₀ [7:0]	R ₀ [7:0]	G ₀ [7:0]	B ₀ [7:0]
RGB888	B ₁ [7:0]	R ₀ [7:0]	G ₀ [7:0]	B ₀ [7:0]
	G ₂ [7:0]	B ₂ [7:0]	R ₁ [7:0]	G ₁ [7:0]
	R ₃ [7:0]	G ₃ [7:0]	B ₃ [7:0]	R ₂ [7:0]
RGB565	R ₁ [4:0] G ₁ [5:3]	G ₁ [2:0] B ₁ [4:0]	R ₀ [4:0] G ₀ [5:3]	G ₀ [2:0] B ₀ [4:0]
ARGB1555	A ₁ [0] R ₁ [4:0] G ₁ [4:3]	G ₁ [2:0] B ₁ [4:0]	A ₀ [0] R ₀ [4:0] G ₀ [4:3]	G ₀ [2:0] B ₀ [4:0]
ARGB4444	A ₁ [3:0] R ₁ [3:0]	G ₁ [3:0] B ₁ [3:0]	A ₀ [3:0] R ₀ [3:0]	G ₀ [3:0] B ₀ [3:0]

注：可通过 ARGB8888 模式支持按 32 位对齐 RGB888。

44.2.9 GPHA HB 主设备端口定时器

HB 主设备端口内嵌一个 8 位定时器，旨在实现可选限制交叉开关矩阵的带宽。此定时器由 HB 时钟驱动，负责对两个连续访问之间的死区进行计数，从而实现对带宽使用的限制。

可通过 R32_GPHA_AMTCR 寄存器的 EN 位和 DT[7:0] 位对定时器的使能以及死区值进行配置，HB 死区的设置请参考 44.3.16 章节。

44.2.10 GPHA 事务

GPHA 事务由一系列预设数量的数据传输序列构成。数据的数量与宽度可通过软件进行编程。

每个 GPHA 数据传输工程最多需要 4 个步骤：

1) 从 R32_GPHA_FGMAR 寄存器寻址的存储单元加载数据，并按照 R32_GPHA_FGPFCCR 中的定义进行像素格式转换。

2) 从 R32_GPHA_BGMAR 寄存器寻址的存储单元加载数据，并按照 R32_GPHA_BGPFCCR 中的定义进行像素格式转换。

3) 利用 PFC 操作处理 alpha 值，获取 alpha 通道信息，并将所有检索到的像素进行混合。

4) 根据 R32_GPHA_OPFCCR 寄存器对合成像素进行像素格式转换，然后将转换后的数据编程到通

过 R32_GPHA_OMAR 寄存器寻址的存储单元。

44.2.11 GPHA 配置

源和目标数据传输在整个 4GB 区域（地址范围在 0x00000000 和 0xFFFFFFFF 之间）都可以寻址外设和存储器。

GPHA 可在以下 3 种模式下工作，通过 R32_GPHA_CTLR 寄存器 MODE[1:0] 位选择工作模式：

- 寄存器到存储器
- 存储器到存储器并执行 PFC
- 存储器到存储器并执行 PFC 和混合

44.2.11.1 寄存器到存储器

寄存器到存储器模式用于以预定义颜色填充用户自定义区域。

颜色格式在 R32_GPHA_OPFCCR 寄存器中设置。

GPHA 不从任何源获取数据。它只将 R32_GPHA_OCOLR 寄存器中定义的颜色写入通过 R32_GPHA_OMAR 寻址以及 R32_GPHA_NLR 和 R32_GPHA_OOR 定义的区域。

44.2.11.2 存储器到存储器并执行 PFC

在此模式下，GPHA 对源数据执行像素格式转换并将结果存储在目标存储单元。

对于要传输的区域大小，源区域大小由 R32_GPHA_NLR 和 R32_GPHA_FGOR 寄存器定义，目标区域大小则由 R32_GPHA_NLR 和 R32_GPHA_OOR 寄存器定义。

从 R32_GPHA_FGMAR 寄存器定义的位置获取数据，并由前景层 PFC 进行处理。原始像素格式通过 GPHA_FGPFCCR 寄存器配置。

如果原始像素格式是直接颜色模式，则所有颜色通道都扩展到 8 位。

如果像素格式是间接颜色模式，则必须将相关 CLUT 加载到 CLUT 存储器中。

CLUT 加载可按如下顺序自动完成：

- 1) 在 R32_GPHA_FGCMAR 寄存器中设置 CLUT 地址。
- 2) 在 R32_GPHA_FGPFCCR 寄存器的 CS[7:0] 位设置 CLUT 大小。
- 3) 在 R32_GPHA_FGPFCCR 寄存器的 CCM 位设置 CLUT 格式（24 或 32 位）。
- 4) 将 R32_GPHA_FGPFCCR 寄存器的 START 位置 1 启动 CLUT 加载。

CLUT 加载完成时，R32_GPHA_ISR 寄存器将置位 CTCIF 标志；如果 R32_GPHA_CTLR 寄存器中的 CTCIE 位置 1，还将产生中断。CLUT 自动加载过程无法与传统的 GPHA 传输同时进行。

CLUT 还可由 CPU 填充，或由任意其它主设备通过 HB 端口填充。在 GPHA 传输进行期间和使用 CLUT（间接颜色格式）时无法访问 CLUT。

44.2.11.3 存储器到存储器并执行 PFC 和混合

此模式下，将在前景层 FIFO 和背景层 FIFO（分别在 R32_GPHA_FGMAR 寄存器和 R32_GPHA_BGMAR 寄存器中定义）获取 2 个源图像。

必须按存储器到存储器模式中所述配置两个像素格式转换器。由于这两个像素格式转换器各自独立且自身具有 CLUT 存储器，因此其配置可以不同。

在每个像素都通过相应的 PFC 转换为 32 位后，将根据 44.3.6 中的公式进行混合，输出 PFC 将根据指定的输出格式对得到的 32 位像素值进行编码，并且编码数据将写入 GPHA_OMAR 寻址的目标存储单元。

44.2.11.4 配置错误检测

GPHA 将在每次执行传输前检查配置是否正确。开始新的传输/自动加载时，如果检测到配置错误，硬件将设置配置错误中断标志。如果 R32_GPHA_CTLR 寄存器 CEIE 位置 1，还将产生中断。

可检测到的错误配置如下：

- 存储器传输（寄存器到存储器模式除外）：R32_GPHA_FGMAR 寄存器 MA 位与 R32_GPHA_FGPFCCR 寄存器 CM 位不匹配；
- 存储器传输（寄存器到存储器模式除外）：R32_GPHA_FGPFCCR 寄存器中的 CM 位无效；
- 存储器传输（寄存器到存储器模式除外）：R32_GPHA_NLR 寄存器 PL 位为奇，而 R32_GPHA_FGPFCCR 寄存器 CM 位为 A4 或 L4；
- 存储器传输（寄存器到存储器模式除外）：R32_GPHA_FGOR 寄存器中的 L0 位为奇，而 R32_GPHA_FGPFCCR 寄存器的 CM 位为 A4 或 L4；
- 存储器传输（仅限混合模式）：R32_GPHA_BGMAR 寄存器 MA 位与 R32_GPHA_BGPFCCR 寄存器 CM 位不一致；
- 存储器传输：R32_GPHA_BGPFCCR 寄存器 CM 位无效（仅限混合模式）；
- 存储器传输（仅限混合模式）：R32_GPHA_NLR 寄存器 PL 位为奇，而 R32_GPHA_BGPFCCR 寄存器 CM 位为 A4 或 L4；
- 存储器传输（仅限混合模式）：R32_GPHA_BGOR 寄存器 L0 位为奇，而 R32_GPHA_BGPFCCR 寄存器 CM 位为 A4 或 L4；
- 存储器传输：R32_GPHA_OMAR 寄存器 MA 位与 R32_GPHA_OPFCCR 寄存器 CM 位不匹配；
- 存储器传输：R32_GPHA_OPFCCR 寄存器 CM 位无效；
- 存储器传输：R32_GPHA_NLR 寄存器中的 NL 位=0；
- 存储器传输：R32_GPHA_NLR 寄存器中的 PL 位=0；

44.2.12 GPHA 传输控制（启动、挂起、中止和完成）

配置 GPHA 后，通过将 R32_GPHA_CTLR 寄存器 START 位置 1，可启动传输。传输完成时，START 位自动复位且 R32_GPHA_ISR 寄存器置位 TCIF 标志。如果 R32_GPHA_CTLR 寄存器 TCIE 位置 1，还将产生中断。

用户应用程序随时都可通过将 R32_GPHA_CTLR 寄存器 SUSP 位置 1 来挂起 GPHA。随即可通过将 GPHA_CTLR 寄存器 ABORT 位置 1 来中止事务，或通过将 R32_GPHA_CTLR 寄存器 SUSP 位复位重新启动事务。

用户应用程序随时都可通过将 R32_GPHA_CTLR 寄存器 ABORT 位置 1 来中止处理中的事务。在这种情况下，不会置位 TCIF 标志。

还可通过 R32_GPHA_FGPFCCR 寄存器和 R32_GPHA_BGPFCCR 寄存器自身的 START 位中止或挂起 CLUT 自动传输过程。

44.2.13 水印

可对水印编程，以在指定行的最后一个像素写入目标存储区域时产生中断。

行号在 R32_GPHA_LWR 寄存器 LW[15:0]位中定义。

在该行的最后一个像素传输完成时，R32_GPHA_ISR 将置位 TWIF 标志；在 R32_GPHA_CTLR 寄存器 TWIE 位置 1 的情况下，还将产生中断。

44.2.14 错误管理

可触发错误：CLUT 访问引发的冲突（CPU 在 CLUT 加载或 GPHA 传输执行期间尝试访问 CLUT），通过 R32_GPHA_ISR 寄存器 CAEIF 标志指示。

这个标志与其在 R32_GPHA_CTLR 寄存器中的中断使能标志（控制在需要时产生中断的 TEIE 和 CAEIE）相关。

44.2.15 HB 死区

为了限制 HB 带宽的使用，可在两个连续的 HB 访问之间编程一个死区。

死区功能可通过将 R32_GPHA_AMTCR 寄存器 EN 位置 1 使能。

死区值存储在 R32_GPHA_AMTCR 寄存器 DT[7:0]位中。该值表示 HB 总线上两个连续事务之间允许占用的最少周期数。

在 GPHA 运行过程中对死区值所做的更新将在下一次 HB 传输时生效。

44.2.16 中断

表 44-8 中断请求

中断事件	事件标志	使能控制位
配置错误	CEIF	CEIE
CLUT 传输完成	CTCIF	CTCIE
CLUT 访问错误	CAEIF	CAEIE
传输水印	TWF	TWIE
传输完成	TCIF	TCIE

44.3 寄存器描述

表 44-9 GPHA 相关寄存器列表

名称	访问地址	描述	复位值
R32_GPHA_CTLR	0x40016800	GPHA 控制寄存器	0x00000000
R32_GPHA_ISR	0x40016804	GPHA 中断状态寄存器	0x00000000
R32_GPHA_IFCR	0x40016808	GPHA 中断标志清零寄存器	0x00000000
R32_GPHA_FGMAR	0x4001680C	GPHA 前景层存储器地址寄存器	0x00000000
R32_GPHA_FGOR	0x40016810	GPHA 前景层偏移寄存器	0x00000000
R32_GPHA_BGMR	0x40016814	GPHA 背景层存储器地址寄存器	0x00000000
R32_GPHA_BGOR	0x40016818	GPHA 背景层偏移寄存器	0x00000000
R32_GPHA_FGPFCCR	0x4001681C	GPHA 前景层 PFC 控制寄存器	0x00000000
R32_GPHA_FGCOLR	0x40016820	GPHA 前景层颜色寄存器	0x00000000
R32_GPHA_BGPFCCR	0x40016824	GPHA 背景层 PFC 控制寄存器	0x00000000
R32_GPHA_BGCOLR	0x40016828	GPHA 背景层颜色寄存器	0x00000000
R32_GPHA_FGCMAR	0x4001682C	GPHA 前景层 CLUT 存储器地址寄存器	0x00000000
R32_GPHA_BGCMAR	0x40016830	GPHA 背景层 CLUT 存储器地址寄存器	0x00000000
R32_GPHA_OPFCCR	0x40016834	GPHA 输出 PFC 控制寄存器	0x00000000
R32_GPHA_OCOLR	0x40016838	GPHA 输出颜色寄存器	0x00000000
R32_GPHA_OMAR	0x4001683C	GPHA 输出存储器地址寄存器	0x00000000
R32_GPHA_OOR	0x40016840	GPHA 输出偏移寄存器	0x00000000
R32_GPHA_NLR	0x40016844	GPHA 行数寄存器	0x00000000
R32_GPHA_LWR	0x40016848	GPHA 行水印寄存器	0x00000000
R32_GPHA_AMTCR	0x4001684C	GPHA HB 主设备定时器配置寄存器	0x00000000
R32_GPHA_FGCWRS	0x40016850	GPHA 前景层 CLUT 读写设置寄存器	0x00000000
R32_GPHA_FGCDAT	0x40016854	GPHA 前景层 CLUT 读写数据寄存器	0x00000000
R32_GPHA_BGCWRS	0x40016858	GPHA 背景层 CLUT 读写设置寄存器	0x00000000
R32_GPHA_BGCDAT	0x4001685C	GPHA 背景层 CLUT 读写数据寄存器	0x00000000

44.3.1 GPHA 控制寄存器 (R32_GPHA_CTLR)

偏移地址: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													MODE[2:0]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CEIE	CTCIE	CAEIE	TWIE	TCIE	Reserved							ABORT	SUSP	START

位	名称	访问	描述	复位值
[31:19]	Reserved	R0	保留。	0
[18:16]	MODE[2:0]	RW	GPHA 工作模式： 001: 存储器到存储器并执行 PFC（仅限 FG PFC 激活时的 FG 获取）； 010: 存储器到存储器并执行混合（执行 PFC 和混合时的 FG 和 BG 获取）； 011: 寄存器到存储器（无 FG 和 BG，仅输出阶段激活）； 其他：保留。	0
[15:14]	Reserved	R0	保留。	0
13	CEIE	RW	配置错误中断使能： 1: CE 中断使能； 0: CE 中断禁止。	0
12	CTCIE	RW	CLUT 传输完成中断使能： 1: CTC 中断使能； 0: CTC 中断禁止。	0
11	CAEIE	RW	CLUT 访问错误中断使能： 1: CAE 中断使能； 0: CAE 中断禁止。	0
10	TWIE	RW	传输水印中断使能： 1: TW 中断使能； 0: TW 中断禁止。	0
9	TCIE	RW	传输完成中断使能： 1: TC 中断使能； 0: TC 中断禁止。	0
[8:3]	Reserved	R0	保留。	0
2	ABORT	RW	中止： 1: 请求传输中止； 0: 不请求传输中止。	0
1	SUSP	RW	挂起： 1: 传输挂起； 0: 传输不挂起。	0
0	START	RW	启动： 此位可用于根据各种配置寄存器中加载的参数启动 GPHA。在下列情况下将自动复位此位： - 传输结束时； - 通过用户应用程序将 GPHA_CTLR 寄存器 ABORT 位置 1 中止数据传输时；	0

			- 数据传输出错时； - 因配置错误或已经在进行其它传输操作（CLUT 自动加载）导致数据传输未启动时。	

44.3.2 GPHA 中断状态寄存器（R32_GPHA_ISR）

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										CEIF	CTCIF	CAEIF	TWIF	TCIF	Reserved

位	名称	访问	描述	复位值
[31:6]	Reserved	R0	保留。	0
5	CEIF	R0	配置错误中断标志： R32_GPHA_CTLR 、 R32_GPHA_FGPFCCR 或 R32_GPHA_BGPFCCR 寄存器的 START 位置 1 以及编程了错误的配置时，此位置 1。	0
4	CTCIF	R0	CLUT 传输完成中断标志： 完成将 CLUT 从系统存储区复制到 GPHA 内部存储器时，此位置 1。	0
3	CAEIF	R0	CLUT 访问错误中断标志： 在从系统存储器自动将 CLUT 复制到 GPHA 内部存储器期间，CPU 若访问 CLUT，此位将置 1。	0
2	TWIF	R0	传输水印中断标志： 带水印行的最后一个像素完成传输时，此位置 1。	0
1	TCIF	R0	传输完成中断标志： GPHA 传输操作完成（仅限数据传输）时此位置 1。	0
0	Reserved	R0	保留。	0

44.3.3 GPHA 中断标志清零寄存器（R32_GPHA_IFCR）

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										CCEIF	CCTCIF	CAECIF	CTWIF	CTCIF	Reserved

位	名称	访问	描述	复位值
[31:6]	Reserved	R0	保留。	0
5	CCEIF	RW1Z	清除配置错误中断标志，写 1 清 0，写 0 无效。	0
4	CCTCIF	RW1Z	清除 CLUT 传输完成中断标志，写 1 清 0，写 0 无效。	0

3	CAECIF	RW1Z	清除 CLUT 访问错误中断标志，写 1 清 0，写 0 无效。	0
2	CTWIF	RW1Z	清除传输水印中断标志，写 1 清 0，写 0 无效。	0
1	CTCIF	RW1Z	清除传输完成中断标志，写 1 清 0，写 0 无效。	0
0	Reserved	R0	保留。	0

44.3.4 GPHA 前景层存储器地址寄存器 (R32_GPHA_FGMAR)

偏移地址：0x0C

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

位	名称	访问	描述	复位值
[31:0]	MA[31:0]	RW	前景层图像的存储地址： 该寄存器仅在数据传输被禁止时允许写入操作。 数据传输一旦启动，寄存器将转变为只读状态。 地址对齐需与图像格式相匹配，例如，采用每像素 32 位格式时，地址对齐必须为 32 位；采用每像素 16 位格式时，地址对齐必须为 16 位；而采用每像素 4 位格式时，地址对齐必须为 8 位。	0

44.3.5 GPHA 前景层偏移寄存器 (R32_GPHA_FGOR)

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		L0[13:0]													

位	名称	访问	描述	复位值
[31:14]	Reserved	R0	保留。	0
[13:0]	L0[13:0]	RW	前景层图像的行偏移： 此值用于生成地址。行偏移将添加到各行末尾，用于确定下一行的起始地址。 该寄存器仅在数据传输被禁止时允许写入操作。 数据传输一旦启动，该域将变为只读。 如果图像格式为每像素 4 位，则行偏移值必须为偶数。	0

44.3.6 GPHA 背景层存储器地址寄存器 (R32_GPHA_BGMAR)

偏移地址：0x14

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

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

位	名称	访问	描述	复位值
[31:19]	MA[31:0]	RW	背景层图像的存储地址： 该寄存器仅在数据传输被禁止时允许写入操作。 一旦数据传输过程被激活，寄存器将转变为只读状态。 地址对齐需与图像格式相匹配，例如，采用每像素 32 位格式时，地址对齐必须为 32 位；采用每像素 16 位格式时，地址对齐必须为 16 位；而采用每像素 4 位格式时，地址对齐必须为 8 位。	0

44.3.7 GPHA 背景层偏移寄存器 (R32_GPHA_BGOR)

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
Reserved		L0[13:0]													

位	名称	访问	描述	复位值
[31:14]	Reserved	R0	保留。	0
[13:0]	L0[13:0]	RW	背景层图像的行偏移： 此值用于生成地址。行偏移将添加到各行末尾，用于确定下一行的起始地址。 该寄存器仅在数据传输被禁止时允许写入操作。 数据传输一旦启动，该域将变为只读。 如果图像格式为每像素 4 位，则行偏移值必须为偶数。	0

44.3.8 GPHA 前景层 PFC 控制寄存器 (R32_GPHA_FGPFCCR)

偏移地址：0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA[7:0]								Reserved		RBS	AI	Reserved		AM[1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS[7:0]								Reserved		STA RT	CCM	CM[3:0]			

位	名称	访问	描述	复位值
[31:24]	ALPHA[7:0]	RW	Alpha 值： 此域定义 alpha 值，该值可作为原始 alpha 值的	0

			替代或与之相乘, 具体取决于通过 AM[1:0] 位选择的 alpha 值。 注: 该域仅在数据传输被禁止时允许写入操作。数据传输一旦启动, 该域将变为只读。	
[23:22]	Reserved	R0	保留。	0
21	RBS	RW	红蓝交换: 1: 交换模式 (BGR 或 ABGR); 0: 常规模式 (RGB 或 ARGB)。	0
20	AI	RW	将 Alpha 取反: 1: 取反 Alpha; 0: 常规 Alpha。 注: 传输一旦启动, 此位将变为只读。	0
[19:18]	Reserved	R0	保留。	0
[17:16]	AM[1:0]	RW	Alpha 模式: 00: 不修改前景层图像的 alpha 通道值; 01: 原始前景层图像的 alpha 通道值替换为 ALPHA[7:0]; 10: 原始前景层图像的 alpha 通道值替换为 ALPHA[7:0] 与原始 alpha 通道值的乘积; 其他: 保留。 注: 该域仅在数据传输被禁止时允许写入操作。数据传输一旦启动, 该域将变为只读。	0
[15:8]	CS[7:0]	RW	CLUT 大小: 前景层图像所用的 CLUT 条目数等于 CS[7:0] + 1。 注: CLUT 传输一旦启动, 该域将变为只读。	0
[7:6]	Reserved	R0	保留。	0
5	START	RW	启动 CLUT 的自动加载过程: 1: 启动; 0: 关闭。 该位在以下情况下自动复位: - 传输结束时; - 通过用户应用程序将 R32_GPHA_CTLR 寄存器 ABORT 位置 1 中止传输时; - 传输出错时; - 因配置错误或已经在进行其它传输操作 (数据传输或自动背景层 CLUT 传输) 导致传输未启动时。	0
4	CCM	RW	CLUT 颜色模式: 1: RGB888; 0: ARGB8888。 注: 该域仅在数据传输被禁止时允许写入操作。传输一旦启动, 该位将变为只读。	0
[3:0]	CM[3:0]	RW	颜色模式: 0000: ARGB8888; 0001: RGB888; 0010: RGB565; 0011: ARGB1555;	0

		0100: ARGB4444; 0101: L8; 0110: AL44; 0111: AL88; 1000: L4; 1001: A8; 1010: A4; 其他: 保留。 注: 该域仅在数据传输被禁止时允许写入操作。传输一旦启动, 该位将变为只读。	

44.3.9 GPHA 前景层颜色寄存器 (R32_GPHA_FGCOLR)

偏移地址: 0x20

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

位	名称	访问	描述	复位值
[31:24]	Reserved	RO	保留。	0
[23:16]	RED[7:0]	RW	前景层图像的 A4 或 A8 模式的红色值。	0
[15:8]	GREEN[7:0]	RW	前景层图像的 A4 或 A8 模式的绿色值。	0
[7:0]	BLUE[7:0]	RW	前景层图像的 A4 或 A8 模式的蓝色值。	0

注: R32_GPHA_FGCOLR 寄存器仅在数据传输被禁止时允许写入操作。传输一旦启动, 该位将变为只读。

44.3.10 GPHA 背景层 PFC 控制寄存器 (R32_GPHA_BGPFCCR)

偏移地址: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA[7:0]								Reserved	RBS	AI	Reserved	AM[1:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS[7:0]								Reserved	STA RT	CCM	CM[3:0]				

位	名称	访问	描述	复位值
[31:24]	ALPHA[7:0]	RW	Alpha 值: 此域定义 alpha 值, 该值可作为原始 alpha 值的替代或与之相乘, 具体取决于通过 AM[1:0]位选择的 alpha 值。 注: 该域仅在数据传输被禁止时允许写入操作。数据传输一旦启动, 该域将变为只读。	0
[23:22]	Reserved	RO	保留。	0
21	RBS	RW	红蓝交换: 1: 交换模式 (BGR 或 ABGR);	0

			0: 常规模式 (RGB 或 ARGB)。 注: 传输一旦启动, 此位将变为只读。	
20	AI	RW	将Alpha取反: 1: 取反Alpha; 0: 常规Alpha。 注: 传输一旦启动, 此位将变为只读。	0
[19:18]	Reserved	R0	保留。	0
[17:16]	AM[1:0]	RW	Alpha 模式: 00: 不修改背景层图像的 alpha 通道值; 01: 原始背景层图像的 alpha 通道值替换为 ALPHA[7:0]; 10: 原始背景层图像的 alpha 通道值替换为 ALPHA[7:0]与原始 alpha 通道值的乘积; 其他: 保留。 注: 该域仅在数据传输被禁止时允许写入操作。数据传输一旦启动, 这域将变为只读。	0
[15:8]	CS[7:0]	RW	CLUT 大小: 背景层图像所用的 CLUT 条目数等于 CS[7:0] + 1。 注: CLUT 传输一旦启动, 该域将变为只读。	0
[7:6]	Reserved	R0	保留。	0
5	START	RW	启动 CLUT 的自动加载过程: 1: 启动; 0: 关闭。 该位在以下情况下自动复位: - 传输结束时; - 通过用户应用程序将 R32_GPHA_CTLR 寄存器 ABORT 位置 1 中止传输时; - 传输出错时; - 因配置错误或已经在进行其它传输操作 (数据传输或自动背景层 CLUT 传输)导致传输未启动时。	0
4	CCM	RW	CLUT 颜色模式: 1: RGB888; 0: ARGB8888。 注: 该域仅在数据传输被禁止时允许写入操作。传输一旦启动, 该位将变为只读。	0
[3:0]	CM[3:0]	RW	颜色模式: 0000: ARGB8888; 0001: RGB888; 0010: RGB565; 0011: ARGB1555; 0100: ARGB4444; 0101: L8; 0110: AL44; 0111: AL88; 1000: L4; 1001: A8;	0

		1010: A4; 其他: 保留。 注: 该域仅在数据传输被禁止时允许写入操作。传输一旦启动, 该位将变为只读。	

44.3.11 GPHA 背景层颜色寄存器 (R32_GPHA_BGCOLOR)

偏移地址: 0x28

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

位	名称	访问	描述	复位值
[31:24]	Reserved	R0	保留。	0
[23:16]	RED[7:0]	RW	背景层图像的 A4 或 A8 模式的红色值。	0
[15:8]	GREEN[7:0]	RW	背景层图像的 A4 或 A8 模式的绿色值。	0
[7:0]	BLUE[7:0]	RW	背景层图像的 A4 或 A8 模式的蓝色值。	0

注: R32_GPHA_BGCOLOR 寄存器仅在数据传输被禁止时允许写入操作。传输一旦启动, 该位将变为只读。

44.3.12 GPHA 前景层 CLUT 存储器地址寄存器 (R32_GPHA_FGCMAR)

偏移地址: 0x2C

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

位	名称	访问	描述	复位值
[31:0]	MA[31:0]	RW	前景层图像的 CLUT 存储地址: 该寄存器仅在数据传输被禁止时允许写入操作。 CLUT 传输一旦启动, 寄存器将转变为只读状态。 若前景层 CLUT 格式是 32 位, 则地址必须是 32 位对齐。	0

44.3.13 GPHA 背景层 CLUT 存储器地址寄存器 (R32_GPHA_BGCMAR)

偏移地址: 0x30

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

位	名称	访问	描述	复位值

[31:19]	MA[31:0]	RW	背景层图像的 CLUT 存储地址： 该寄存器仅在数据传输被禁止时允许写入操作。 CLUT 传输一旦启动，寄存器将转变为只读状态。 若前景层 CLUT 格式是 32 位，则地址必须是 32 位对齐。	0

44.3.14 GPHA 输出 PFC 控制寄存器 (R32_GPHA_OPFCCR)

偏移地址：0x34

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

位	名称	访问	描述	复位值
[31:22]	Reserved	R0	保留。	0
21	RBS	RW	红蓝交换： 1：交换模式（BGR 或 ABGR）； 0：常规模式（RGB 或 ARGB）。 注：传输一旦启动，此位将变为只读。	0
20	AI	RW	将Alpha取反： 1：取反Alpha； 0：常规Alpha。 注：传输一旦启动，此位将变为只读。	0
[19:3]	Reserved	R0	保留。	0
[2:0]	CM[2:0]	RW	颜色模式： 000：ARGB8888； 001：RGB888； 010：RGB565； 011：ARGB1555； 100：ARGB4444； 其他：保留。 注：该寄存器仅在数据传输被禁止时允许写入操作。传输一旦启动，寄存器将转变为只读状态。	0

44.3.15 GPHA 输出颜色寄存器 (R32_GPHA_OCOLR)

偏移地址：0x38

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

位	名称	访问	描述	复位值
[31:24]	ALPHA[7:0]	RW	输出颜色的 alpha 通道。	0

[23:16]	RED[7:0]	RW	输出图像的红色值。	0
[15:8]	GREEN[7:0]	RW	输出图像的绿色值。	0
[7:0]	BLUE[7:0]	RW	输出图像的蓝色值。	0

44.3.16 GPHA 输出存储器地址寄存器 (R32_GPHA_OMAR)

偏移地址: 0x3C

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

位	名称	访问	描述	复位值
[31:19]	MA[31:0]	RW	输出 FIFO 的存储地址： 该寄存器仅在数据传输被禁止时允许写入操作。 一旦数据传输过程被激活，寄存器将转变为只读状态。 地址对齐需与图像格式相匹配，例如，采用每像素 32 位格式时，地址对齐必须为 32 位；采用每像素 16 位格式时，地址对齐必须为 16 位。 注：若输出的像素格式为 RGB888，则输出像素数必须为 4 的倍数。	0

44.3.17 GPHA 输出偏移寄存器 (R32_GPHA_OOR)

偏移地址: 0x40

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		LO[13:0]													

位	名称	访问	描述	复位值
[31:14]	Reserved	RO	保留。	0
[13:0]	LO[13:0]	RW	输出的行偏移： 此值用于生成地址。行偏移将添加到各行末尾，用于确定下一行的起始地址。 该寄存器仅在数据传输被禁止时允许写入操作。 数据传输一旦启动，该域将变为只读。	0

44.3.18 GPHA 行数寄存器 (R32_GPHA_NLR)

偏移地址: 0x44

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

NL[15:0]

位	名称	访问	描述	复位值
[31:30]	Reserved	R0	保留。	0
[29:16]	PL[13:0]	RW	待传输区域的每行像素数： 该寄存器仅在数据传输被禁止时允许写入操作。 数据传输一旦启动，该域将变为只读。 如果图像格式为每像素 4 位，则行偏移值必须为偶数。 注：当 R32_GPHA_CTLR 寄存器 MODE[2:0]=011b 时 PL[13:0]位的值应大于 0。	0
[15:0]	NL[15:0]	RW	待传输区域的行数： 该寄存器仅在数据传输被禁止时允许写入操作。 数据传输一旦启动，该域将变为只读。	0

44. 3. 19 GPHA 行水印寄存器 (R32_GPHA_LWR)

偏移地址：0x48

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
LW[15:0]															

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:0]	LW[15:0]	RW	行水印： 在带水印行的最后一个像素传输完成时产生中断。仅在数据传输被禁止时允许写入操作。数据传输一旦启动，该域将变为只读。	0

44. 3. 20 GPHA HB 主设备定时器配置寄存器 (R32_GPHA_AMTCR)

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
DT[7:0]								Reserved							EN

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
[15:8]	DT[7:0]	RW	在 HB 主设备端口上两个连续访问之间所插入的死区值，以 HB 时钟周期数表示。这些位表示两个连续 HB 访问之间允许占用的最少周期数。	0
[7:1]	Reserved	R0	保留。	0

0	EN	RW	使能死区功能： 1：使能； 0：禁止。	0

44. 3. 21 GPHA 前景层 CLUT 读写设置寄存器（R32_GPHA_FGCWRS）

偏移地址：0x50

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							FG_CLUT_EN	FG_CLUT_INDEX[7:0]							

位	名称	访问	描述	复位值
[31:9]	Reserved	R0	保留。	0
8	FG_CLUT_EN	RW	CPU 读写查找表使能： 1：使能； 0：关闭。	0
[7:0]	FG_CLUT_INDEX[7:0]	RW	设置查找表地址。	0

44. 3. 22 GPHA 前景层 CLUT 读写数据寄存器（R32_GPHA_FGCDAT）

偏移地址：0x54

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

位	名称	访问	描述	复位值
[31:0]	FG_CLUT_DATA[31:0]	RW	读数据或写查找表 SRAM 数据。	0

44. 3. 23 GPHA 背景层 CLUT 读写设置寄存器（R32_GPHA_BGCWRS）

偏移地址：0x58

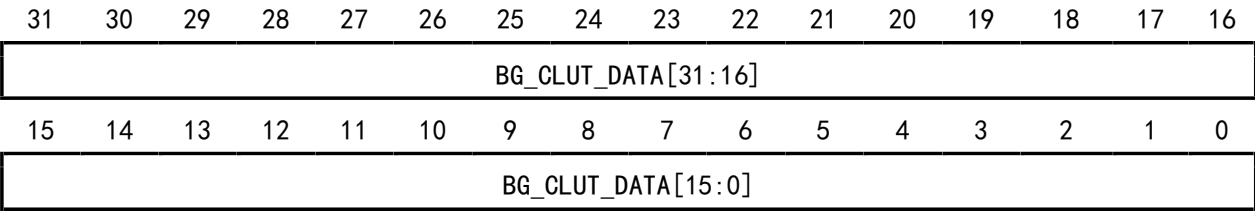
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							BG_CLUT_EN	BG_CLUT_INDEX[7:0]							

位	名称	访问	描述	复位值
[31:9]	Reserved	R0	保留。	0
8	BG_CLUT_EN	RW	CPU 读写查找表使能： 1：使能；	0

			0：关闭。	
[7:0]	BG_CLUT_INDEX[7:0]	RW	设置查找表地址。	0

44. 3. 24 GPHA 背景层 CLUT 读写数据寄存器（R32_GPHA_BGCDAT）

偏移地址：0x5C



位	名称	访问	描述	复位值
[31:0]	BG_CLUT_DATA[31:0]	RW	读数据或写查找表 SRAM 数据。	0

第 45 章 调试支持（DBG）

45.1 主要特征

此寄存器允许在调试状态下配置 MCU。包括：

- 支持独立看门狗（IWDG）的计数器
- 支持窗口看门狗（WWDG）的计数器
- 支持定时器的计数器
- 支持 I2CSMBus 的超时控制
- 支持 CAN 通信

45.2 寄存器描述

45.2.1 RISC-V 调试 MCU 配置寄存器（R32_DBGMCU_CR）

地址：0x7C0（GSR）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CAN3_STOP	CAN2_STOP	CAN1_STOP	I2C4_SMBUS_TIMEOUT	I2C3_SMBUS_TIMEOUT	LPTIM2_STOP	LPTIM1_STOP	TIM12_STOP	TIM11_STOP	TIM10_STOP	TIM9_STOP	TIM8_STOP	TIM7_STOP	TIM6_STOP	TIM5_STOP
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM4_STOP	TIM3_STOP	TIM2_STOP	TIM1_STOP	I2C2_SMBUS_TIMEOUT	I2C1_SMBUS_TIMEOUT	WWDG_STOP	IWDG_STOP	Reserved							

位	名称	访问	描述	复位值
31	Reserved	RW	保留。	0
30	CAN3_STOP	RW	CAN3 调试停止位。当内核进入调试状态时 CAN2 停止运行。 1：CAN3 的接收寄存器不继续接收数据； 0：CAN3 仍然正常运行。	0
29	CAN2_STOP	RW	CAN2 调试停止位。当内核进入调试状态时 CAN2 停止运行。 1：CAN2 的接收寄存器不继续接收数据； 0：CAN2 仍然正常运行。	0
28	CAN1_STOP	RW	CAN1 调试停止位。当内核进入调试状态时 CAN2 停止运行。 1：CAN1 的接收寄存器不继续接收数据； 0：CAN1 仍然正常运行。	0
27	I2C4_SMBUS_TIMEOUT	RW	I2C4_SMBUS 超时模式调试停止位。当内核进入调试状态时停止 SMBUS 超时模式。 1：冻结 SMBUS 的超时控制； 0：与正常模式操作相同。	0
26	I2C3_SMBUS_TIMEOUT	RW	I2C3_SMBUS 超时模式调试停止位。当内核进入调试	0

			状态时停止 SMBUS 超时模式。 1: 冻结 SMBUS 的超时控制; 0: 与正常模式操作相同。	
25	LPTIM2_STOP	RW	低功耗定时器 2 调试停止位。当内核进入调试状态时计数器停止工作。 1: 低功耗定时器 2 的计数器停止工作; 0: 低功耗定时器 2 的计数器仍然正常工作。	0
24	LPTIM1_STOP	RW	低功耗定时器 1 调试停止位。当内核进入调试状态时计数器停止工作。 1: 低功耗定时器 10 的计数器停止工作; 0: 低功耗定时器 1 的计数器仍然正常工作。	0
23	TIM12_STOP	RW	定时器 12 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 12 的计数器停止工作; 0: 定时器 12 的计数器仍然正常工作。	0
22	TIM11_STOP	RW	定时器 11 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 11 的计数器停止工作; 0: 定时器 11 的计数器仍然正常工作。	0
21	TIM10_STOP	RW	定时器 10 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 10 的计数器停止工作; 0: 定时器 10 的计数器仍然正常工作。	0
20	TIM9_STOP	RW	定时器 9 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 9 的计数器停止工作; 0: 定时器 9 的计数器仍然正常工作。	0
19	TIM8_STOP	RW	定时器 8 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 8 的计数器停止工作; 0: 定时器 8 的计数器仍然正常工作。	0
18	TIM7_STOP	RW	定时器 7 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 7 的计数器停止工作; 0: 定时器 7 的计数器仍然正常工作。	0
17	TIM6_STOP	RW	定时器 6 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 6 的计数器停止工作; 0: 定时器 6 的计数器仍然正常工作。	0
16	TIM5_STOP	RW	定时器 5 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 5 的计数器停止工作; 0: 定时器 5 的计数器仍然正常工作。	0
15	TIM4_STOP	RW	定时器 4 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 4 的计数器停止工作;	0

			0: 定时器 4 的计数器仍然正常工作。	
14	TIM3_STOP	RW	定时器 3 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 3 的计数器停止工作; 0: 定时器 3 的计数器仍然正常工作。	0
13	TIM2_STOP	RW	定时器 2 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 2 的计数器停止工作; 0: 定时器 2 的计数器仍然正常工作。	0
12	TIM1_STOP	RW	定时器 1 调试停止位。当内核进入调试状态时计数器停止工作。 1: 定时器 1 的计数器停止工作; 0: 定时器 1 的计数器仍然正常工作。	0
11	I2C2_SMBUS_TIMEOUT	RW	I2C2_SMBUS 超时模式调试停止位。当内核进入调试状态时停止 SMBUS 超时模式。 1: 冻结 SMBUS 的超时控制; 0: 与正常模式操作相同。	0
10	I2C1_SMBUS_TIMEOUT	RW	I2C1_SMBUS 超时模式调试停止位。当内核进入调试状态时停止 SMBUS 超时模式。 1: 冻结 SMBUS 的超时控制; 0: 与正常模式操作相同。	0
9	WWDG_STOP	RW	窗口看门狗调试停止位。当内核进入调试状态时调试窗口看门狗停止工作。 1: 窗口看门狗计数器停止工作; 0: 窗口看门狗计数器仍然正常工作。	0
8	IWDG_STOP	RW	独立看门狗调试停止位。当内核进入调试状态时看门狗停止工作。 1: 看门狗计数器停止工作; 0: 看门狗计数器仍然正常工作。	0
[7:0]	Reserved	RW	保留。	0

第 46 章 闪存及用户选择字（FLASH）

46.1 闪存组织

芯片内部闪存组织结构如下：

表 46-1 闪存组织结构 1（用户区 960K+B00T 区 56K）

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 3839	0x080EFF00 - 0x080EFFFF	256
信息块	系统引导代码存储	0x1FFF0000 - 0x1FFFDFFF	56K
	用户选择字	0x1FFFF800 - 0x1FFFF8FF	256

注：上述主存储器区域用于用户的应用程序存储，以 8K 字节（32 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。

表 46-2 闪存组织结构 2（用户区 480K+B00T 区 28K）

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	0x08077F00 - 0x08077FFF	256
信息块	系统引导代码存储	0x1FFF0000 - 0x1FFF6FFF	28K
	用户选择字	0x1FFFF800 - 0x1FFFF8FF	256

注：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。

46.2 闪存编程及安全性

46.2.1 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，当 DBMODE=1 时芯片单次 8K 字节执行擦除操作，当 DBMODE=0 时芯片单次 4K 字节执行擦除操作。
- 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，

当 DBMODE=1 时芯片单次 64K 字节执行擦除操作，当 DBMODE=0 时芯片单次 32K 字节执行擦除操作。

46.2.2 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) 当 DBMODE=1 时芯片主存储器 0-31 页（8K 字节）自动写保护状态，不受 R32_FLASH_WPR 寄存器控制；解除读保护状态，所有主存储页都由 R32_FLASH_WPR 寄存器控制。
 - 2) 当 DBMODE=0 时芯片主存储器 0-15 页（4K 字节）自动写保护状态，不受 R32_FLASH_WPR 寄存器控制；解除读保护状态，所有主存储页都由 R32_FLASH_WPR 寄存器控制。
 - 3) 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。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。

46.3 FLASH 增强读模式

FLASH 增强读模式适用于用户程序运行在 FLASH 中开启该模式，可提高 FLASH 访问效率。开启该模式需将 R32_FLASH_ACTLR 寄存器的 EHMOD 位置 1，此时读 EHMODS 位，EHMODS 位置 1 即增强读模式已生效，关闭该模式需先将 EHMOD 位清 0，再将 RSENACT 置 1。

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，否则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，否则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。

46.4 寄存器描述

表 46-3 FLASH 相关寄存器列表

名称	访问地址	描述	复位值
R32_FLASH_ACTLR	0x40022000	访问控制寄存器	0x00000000
R32_FLASH_KEYR	0x40022004	FPEC 键寄存器	0xFFFFFFFF
R32_FLASH_OBKEYR	0x40022008	OBKEY 寄存器	0xFFFFFFFF
R32_FLASH_STATR	0x4002200C	状态寄存器	0x00008000
R32_FLASH_CTLR	0x40022010	控制寄存器	0x00008080
R32_FLASH_ADDR	0x40022014	地址寄存器	0x00000000
R32_FLASH_OBR	0x4002201C	选择字寄存器	0xFFFFFFFFFE
R32_FLASH_WPR	0x40022020	写保护寄存器	0xFFFFFFFF
R32_FLASH_MODEKEYR	0x40022024	扩展键寄存器	0xFFFFFFFF
R32_BOOT_MODEKEYR	0x40022028	BOOT 键寄存器	0xFFFFFFFF
R32_FLASH_CFGRO	0x4002202C	配置寄存器 1	0xFFFFFFFF

46.4.1 访问控制寄存器（R32_FLASH_ACTLR）

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

FLASH_ST	FLASH_READY	Reserved	FLASH_RD_MD	Reserved	FLASH_LP	EHMOD	ENHANCE_STATUS	Reserved	SCK_CFG

位	名称	访问	描述	复位值
[31:16]	Reserved	R0	保留。	0
15	FLASH_ST	R0	FLASH 低功耗状态指示位： 1：FLASH 处于低功耗模式； 0：FLASH 处于正常工作模式。	0
14	FLASH_READY	R0	FLASH 就绪指示位： 1：FLASH 就位； 0：FLASH 未准备好。	0
[13:12]	Reserved	R0	保留。	0
11	FLASH_RD_MD	RW	FLASH 读模式控制位： 1：连续读模式； 0：单次读模式。 此位为普通解锁。	0
[10:9]	Reserved	R0	保留。	0
8	FLASH_LP	RW	FLASH 低功耗模式控制位： 1：FLASH 低功耗模式； 0：FLASH 退出低功耗模式。	0
7	EHMOD	RW	FLASH 增强读模式控制位： 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 RSENACT 位一起操作，退出步骤先将 EHM0D 位清 0，再将 RSENACT 置 1。 此位为普通解锁。	0
6	ENHANCE_STATUS	R0	增强状态： 1：读出的命令发送已完成； 0：等待。	0
[5:2]	Reserved	R0	保留。	0
[1:0]	SCK_CFG	RW	FLASH 等待状态数： 00：HCLK； FLASH 读操作--建议 HCLK≤75MHz， FLASH 擦写操作--建议 HCLK≤60MHz； 01：HCLK/2； FLASH 读操作--建议 HCLK≤150MHz， FLASH 擦写操作--建议 HCLK≤120MHz； 10：HCLK/4； FLASH 读操作--HCLK 最大为 150MHz，不建议使用该档位， FLASH 擦写操作--建议 HCLK≤150MHz； 11：保留。	0

46.4.2 FPEC 键寄存器 (R32_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[31:0]	WO	FPEC 键, 用于输入 FPEC 的解锁键包括: RDPRT 键 = 0x000000A5; KEY1 = 0x45670123; KEY2 = 0xCDEF89AB。	X

46.4.3 OBKEY 寄存器 (R32_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[31:0]	WO	选择字键, 用于输入选择字键解除 OPTWRE。 KEY1 = 0x45670123; KEY2 = 0xCDEF89AB。	X

46.4.4 状态寄存器 (R32_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
BOOT_LOCK	BOOT_MODE	BOOT_STATUS	BOOT_AVA	Reserved						EOP	WRPRT_ERR	Reserved		WRBSY	BSY

位	名称	访问	描述	复位值
[31:16]	Reserved	RO	保留。	0
15	BOOT_LOCK	RW1	BOOT 区锁住: 1: 锁住; 0: 解锁。	1
14	BOOT_MODE	RW	和 BOOT_AVA 结合可以控制用户区和 BOOT 区之间的切换: 1: 软件复位之后可以切换到 BOOT 区或者用	0

			户区; 0: 软件复位之后可以切换到用户区。	
13	BOOT_STATUS	RW0	当前执行的程序来源, 写 0 清零: 1: 表示从 BOOT 区加载的程序; 0: 表示从用户区加载的程序。	0
12	BOOT_AVA	RO	程序初始化状态: 1: 表示从 BOOT 区启动; 0: 表示从用户区启动。	0
[11:6]	Reserved	RO	保留。	0
5	EOP	RW1Z	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	WRPRERR	RW1Z	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
[3:2]	Reserved	RO	保留。	0
1	WRBSY	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

46.4.5 控制寄存器 (R32_FLASH_CTLR)

偏移地址: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									RSEN ACT	PGSTR T	Reserved	BER	Rese rved	FTPG	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLOCK	Reserved	EOPIE	Rese rved	ERRIE	OBWRE	Rese rved	LOCK	STRT	OBERR	OBPG	Reserved	PER	PG		

位	名称	访问	描述	复位值
[31:23]	Reserved	RO	保留。	0
22	RSENACT	WO	退出增强读模式, 硬件自动清除, 需配合 EHMOD 位一起操作, 退出步骤先将 EHMOD 位清 0, 再将 RSENACT 置 1。	0
21	PGSTRT	WO	开始。置 1 启动一次页编程, 硬件自动清除。 此位为普通解锁。	0
[20:19]	Reserved	RO	保留。	0
18	BER	RW	当 DBMODE=1 时, 执行 64KB 擦除。 当 DBMODE=0 时, 执行 32KB 擦除。 此位为普通解锁+快速解锁。	0
17	Reserved	RO	保留。	0
16	FTPG	RW	执行快速页编程操作。 此位为普通解锁+快速解锁。	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	1

			确的解锁序列后，硬件清除此位为‘0’。 软件置 1，重新加锁。 此位为普通解锁。	
[14:13]	Reserved	RO	保留。	0
12	EOPIE	RW	操作完成中断控制(R32_FLASH_STATR 寄存器中 EOP 置位)： 1：允许产生中断； 0：禁止产生中断。 此位为普通解锁。	0
11	Reserved	RO	保留。	0
10	ERRIE	RW	错误状态中断控制(R32_FLASH_STATR 寄存器中 PGERR/WRPTERR 置位)： 1：允许产生中断； 0：禁止产生中断。 此位为普通解锁。	0
9	OBWRE	RW0	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 R32_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。 此位为普通解锁。	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 R32_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。 此位为普通解锁。	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’)。 此位为普通解锁。	0
5	OBBER	RW	执行用户选择字擦除。 此位为普通解锁。	0
4	OBPG	RW	执行用户选择字编程。 此位为普通解锁。	0
[3:2]	Reserved	RO	保留。	0
1	PER	RW	当 DBMODE=1 时，执行扇区 (8KB) 擦除操作。 当 DBMODE=0 时，执行扇区 (4KB) 擦除操作。 此位为普通解锁。	0
0	PG	RW	执行标准编程操作。 此位为普通解锁。	0

46.4.6 地址寄存器 (R32_FLASH_ADDR)

偏移地址：0x14

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

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

位	名称	访问	描述	复位值
[31:0]	FAR[31:0]	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 R32_FLASH_STATR 寄存器中的 BSY 位为 '1' 时, 不能写此寄存器。	0

46.4.8 选择字寄存器 (R32_FLASH_OBR)

偏移地址: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DATA1[7:0]						DATA0[7:6]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0[5:0]						USA RTD LEN	USB FSD LEN	Reserved				IWDG _SW	RDPRT	OBERR	

位	名称	访问	描述	复位值
[31:26]	Reserved	R0	保留。	0
[25:18]	DATA1[7:0]	R0	数据字节 1。	X
[17:10]	DATA0[7:0]	R0	数据字节 0。	X
9	USER	USARTDLEN	R0 BOOT 使能 USART 免按键下载功能: 1: 开启; 0: 关闭。	1
8		USBFSDLEN	R0 BOOT 使能 USBFS 下载功能: 1: 开启; 0: 关闭。	1
[7:3]		Reserved	R0 保留。	11111b
2		IWDG_SW	R0 独立看门狗 (IWDG) 硬件使能位, 低电平有效。	1
1	RDPRT		R0 读保护状态。 1: 表示闪存当前读保护有效。	1
0	OBERR		R0 选择字错误。 1: 表示选择字和它的反码不匹配。	0

46.4.9 写保护寄存器 (R32_FLASH_WPR)

偏移地址: 0x20

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

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

位	名称	访问	描述	复位值
[31:0]	WRP[31:0]	RO	闪存写保护状态。 1: 写保护失效; 0: 写保护有效。 当 DBMODE=1 时, 芯片每个比特位代表 8K 字节 (32 页) 存储写保护状态, 在读保护状态前 8K 字节处于写保护状态。 当 DBMODE=0 时, 芯片每个比特位代表 4K 字节 (16 页) 存储写保护状态, 在读保护状态前 4K 字节处于写保护状态。	X

46.4.10 扩展键寄存器 (R32_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[31:0]	WO	输入下面序列解锁快速编程/擦除模式: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB。	X

46.4.11 BOOT 键寄存器 (R32_BOOT_MODEKEYR)

偏移地址: 0x28

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

位	名称	访问	描述	复位值
[31:0]	BOOTKEYR[31:0]	WO	输入下面序列解锁 BOOT 锁: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB。	X

46.4.12 配置寄存器 1 (R32_FLASH_CFGR0)

偏移地址: 0x2C

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

Reserved			DBM ODE	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
[31:29]	Reserved	R0	保留。	x
28	DBMODE	R0	FLASH 模式状态位： 1：双倍 FLASH 模式； 0：单倍 FLASH 模式。	x
[27:0]	Reserved	R0	保留。	x

46.5 闪存操作流程

46.5.1 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。

46.5.2 解除闪存锁

系统复位后，闪存控制器（FPEC）和 R32_FLASH_CTLR 寄存器是被锁定的，不可访问。通过写入序列到 R32_FLASH_KEYR 寄存器可解锁闪存控制器模块。

解锁序列：

- 1) 向 R32_FLASH_KEYR 寄存器写入 KEY1 = 0x45670123（第 1 步必须是 KEY1）；
- 2) 向 R32_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 R32_FLASH_CTLR 寄存器并产生总线错误，直到下次系统复位。

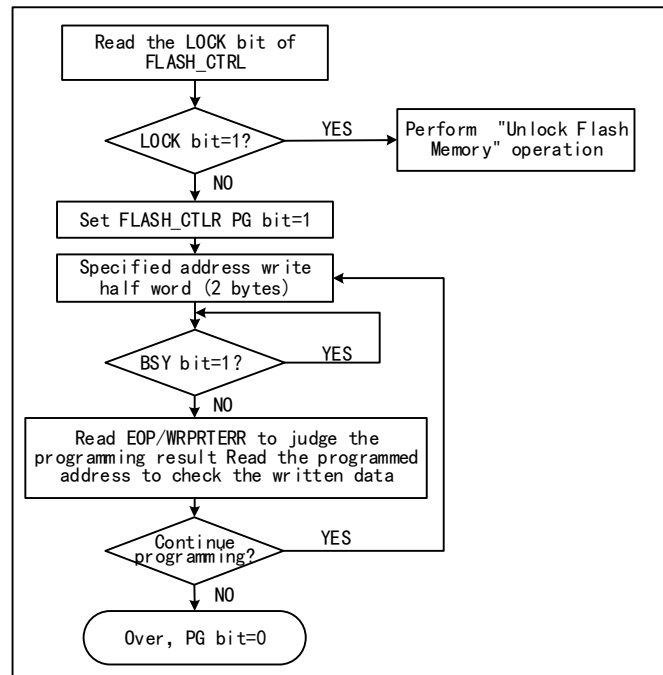
闪存控制器（FPEC）和 R32_FLASH_CTLR 寄存器可以通过将 R32_FLASH_CTLR 寄存器的“LOCK”位，置 1 来再次锁定。

46.5.3 主存储器标准编程

标准编程每次可以写入 2 字节。当 R32_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 46-1 FLASH 编程

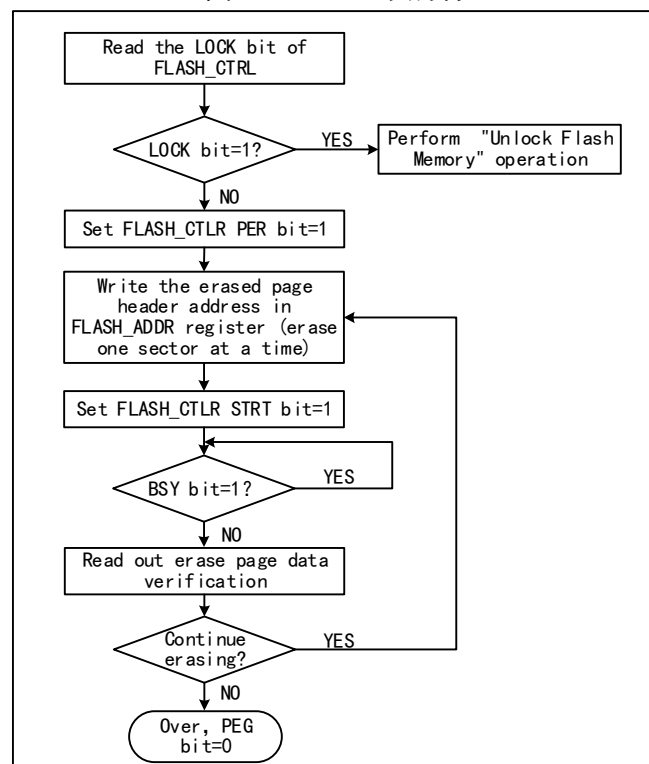


- 1) 检查 R32_FLASH_CTRL 寄存器 LOCK，如果为 1，需要执行“解除闪存锁”操作。
- 2) 设置 R32_FLASH_CTRL 寄存器的 PG 位为 ‘1’，开启标准编程模式。
- 3) 向指定闪存地址（偶地址）写入要编程的半字。
- 4) 等待 BSY 位变为 ‘0’ 或 R32_FLASH_STATR 寄存器的 EOP 位为 ‘1’ 表示编程结束，将 EOP 位清 0。
- 5) 查询 R32_FLASH_STATR 寄存器看是否有错误，或者读编程地址数据校验。
- 6) 继续编程可以重复 3-5 步骤，结束编程将 PG 位清 0。

46.5.4 主存储器标准擦除

闪存按扇区擦除。

图 46-2 FLASH 页擦除



- 1) 检查 R32_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 R32_FLASH_CTLR 寄存器的 PER 位为 ‘1’, 开启扇区擦除模式。
 - 3) 向 R32_FLASH_ADDR 寄存器写入选择擦除的页首地址。
 - 4) 设置 R32_FLASH_CTLR 寄存器的 STRT 位为 ‘1’, 启动一次擦除动作。
 - 5) 等待 BSY 位变为 ‘0’ 或 R32_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) 继续扇区擦除可以重复 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。

46.5.5 快速编程模式解锁

通过写入序列到 R32_FLASH_MODEKEYR 寄存器可解锁快速编程模式操作。解锁后, R32_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 R32_FLASH_CTLR 寄存器的 “FLOCK” 位软件置 1 来再次锁定。

解锁序列:

- 1) 向 R32_FLASH_MODEKEYR 寄存器写入 KEY1 = 0x45670123;
- 2) 向 R32_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。

46.5.6 主存储器快速编程

快速编程按页 (256 字节) 进行编程。

- 1) 检查 R32_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 R32_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 R32_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 R32_FLASH_CTLR 寄存器的 FTPG 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*) 0x8000000 = 0x12345678;
- 6) 等待 R32_FLASH_STATR 寄存器的 WRBSY 为 ‘0’, 写入下个数据。
- 7) 重复步骤 5-6 共 64 次。
- 8) 设置 R32_FLASH_CTLR 寄存器的 FTPG 位为 ‘1’, 启动快速页编程。
- 9) 等待 BSY 位变为 ‘0’ 或 R32_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 R32_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 FTPG 位清 0。

46.5.7 主存储器快速擦除

快速擦除按块进行擦除。

- 1) 检查 R32_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 R32_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 R32_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 R32_FLASH_CTLR 寄存器的 BER 位为 ‘1’, 开启快速块擦除模式功能。
- 5) 向 R32_FLASH_ADDR 寄存器写入快速擦除块的首地址。
- 6) 设置 R32_FLASH_CTLR 寄存器的 STRT 位为 ‘1’, 启动一次快速块擦除动作。
- 7) 等待 BSY 位变为 ‘0’ 或 R32_FLASH_STATR 寄存器的 EOP 位为 ‘1’ 表示擦除结束, 将 EOP 位清 0。
- 8) 查询 R32_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 BER 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。

46.6 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 46-4 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 46-5 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，当 DBMODE=1 时芯片 0-31 页（8K）将自动写保护，不受 WRPR0 控制；当 DBMODE=0 时芯片 0-15 页（4K）将自动写保护，不受 WRPR0 控制。	0xA5
USER	7	USARTDLEN	BOOT 使能 USART 免按键下载功能 ^[1] ： 1：开启； 0：关闭。	1
	6	USBFSLEN	BOOT 使能 USBFS 下载功能 ^[1] ： 1：开启； 0：关闭。	1
	[5:1]	Reserved	保留。	0x1F
	0	IWDGSW	独立看门狗（IWDG）硬件使能位： 0：IWDG 功能由硬件开启（随 LSI 时钟决定）； 1：IWDG 功能由软件开启，禁止硬件开启。	1
Data0 - Data1			存储用户数据 2 字节。	X
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（当 DBMODE=1 时：8K 字节/扇区；当 DBMODE=0 时：4K 字节/扇区）的写保护状态： 0：启用写保护； 1：关闭写保护。	0xFFFF FFFF

	4 个字节用于保护主存储器。 WRPRO: 第 0-7 扇区存储写保护控制; WRPR1: 第 8-15 扇区存储写保护控制; WRPR2: 第 16-23 扇区存储写保护控制; WRPR3: 位 0-6 提供第 24-30 扇区的写保护; 位 7 提供第 31-119 扇区的写保护。	

注[1]: 此功能的具体操作方式和注意事项请参考 CH32H417 评估板说明书。

46. 6. 1 用户选择字解锁

通过写入序列到 R32_FLASH_OBKEYR 寄存器可解锁用户选择字操作。解锁后, R32_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 R32_FLASH_CTLR 寄存器的“OBWRE”位, 软件清 0 来再次锁定。

解锁序列:

- 1) 向 R32_FLASH_OBKEYR 寄存器写入 KEY1 = 0x45670123;
 - 2) 向 R32_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。

46. 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 R32_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 R32_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 R32_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 R32_FLASH_CTLR 寄存器的 OBPG 位为 ‘1’。
 - 5) 写入要编程的半字(2 字节)到指定地址。
 - 6) 等待 BSY 位变为 ‘0’ 或 R32_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。

46. 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 R32_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 R32_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 R32_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 R32_FLASH_CTLR 寄存器的 OBER 位为 ‘1’, 之后设置 R32_FLASH_CTLR 寄存器的 STRT 位为 ‘1’, 开启用户选择字擦除。
 - 5) 等待 BSY 位变为 ‘0’ 或 R32_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。

46. 6. 4 解除读保护

闪存是否读保护, 由用户选择字决定。读取 R32_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。

46.6.5 解除写保护

闪存是否写保护，由用户选择字决定。读取 R32_FLASH_WPR 寄存器，每个比特位代表一个扇区闪存空间，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。

第 47 章 通用高速接口（UHSIF）

本章模块描述仅适用于 CH32H417 微控制器产品。

芯片内置一组通用高速接口 UHSIF，传输时钟最高可达 125MHz，支持 8 位、16 位或 32 位数据宽度，理论最高速度 500Mbytes/s。

具体应用请参考和调用提供的子程序库。