基于 ARM®32 位的 Cortex®-M4F 微控制器，配有 64K 字节到 256K 字节闪存、sLib、15 个定时器、1 个 ADC、18 个通信接口（CAN 和 OTGFS）

- **内核：带有FPU的ARM®32位的Cortex®-M4F CPU**
  - 最高150 MHz工作频率，带存储器保护单元(MPU)，内建单周期乘法和硬件除法
  - 内建浮点运算(FPU)
  - 具有DSP指令集
- **存储器**
  - 64 K到256 K字节的闪存存储器
  - 20 K字节的启动程序代码区作启动加载程序（Bootloader）用，可一次性配置成一般用户区
  - sLib：将指定之主存储区设为执行代码安全库区，此区代码仅能调用无法读取
  - 高达48 K字节的SRAM
  - 具有16位数据总线的外部存储器控制器（XMC）：支持总线复用PSRAM、NOR存储器
- **XMC作为LCD并口，兼容8080/6800模式**
- **电源控制（PWC）**
  - 2.4至3.6V供电
  - 上电复位（POR）、低电压复位(LVR)、电源电压监测器(PVM)
  - 低功耗模式：睡眠、深度睡眠和待机
  - 20个32位的电池供电寄存器（BPR）
- **时钟和复位管理（CRM）**
  - 4至25 MHz晶体（HEXT）
  - 内嵌经出厂调校的48 MHz高速时钟（HICK），25 °C达1%精度，-40 °C至+105 °C达2.5%精度，带自动时钟校准（ACC）功能
  - 32 kHz晶体（LEXT）
  - 低速内部时钟（LICK）
- **模拟模块**
  - 1个12位5.33 MSPS A/D转换器，多达24个外部输入通道；分辨率12/10/8/6位可调；硬件过采样最高达16位分辨率
  - 温度传感器（V_{TS}）、内部参考电压（V_{INTR}）
  - 2个12位D/A转换器
- **DMA：14通道DMA控制器**
- **多达87个快速GPIO端口**
  - 所有GPIO口可以映像到16个外部中断（EXINT）
  - 几乎所有GPIO口可容忍5V输入信号
- **多达15个定时器（TMR）**
  - 1个16位7通道高级定时器，包括3对互补通道PWM输出，带死区控制和紧急停止功能
  - 多达8个16位和1个32位通用定时器，每个定时器最多达4个用于输入捕获/输出比较/PWM或脉冲计数的通道和增量编码器输入
  - 2个16位基本定时器
  - 2个看门狗定时器（WDT和WWDT）
  - 系统滴答定时器：24位递减计数器
- **ERTC：增强型RTC，具有自动唤醒、闹钟、亚秒级精度及硬件日历，带校准功能**
- **多达18个通信接口**
  - 多达3个I^2C接口，支持SMBus/PMBus
  - 3个SPI接口（36 M位/秒），均可复用为半双工I^2S接口；任意2个半双工I^2S可以组合为1个全双工I^2S
  - 8个USART接口；支持主同步SPI和调制解调器控制；具有ISO7816接口、LIN、IrDA、和RS485驱动使能；支持TX/RX可配置引脚互换
  - 2个CAN接口（2.0B主动），每个CAN内置256字节的专用缓存
  - USB OTG全速控制器含片上PHY，内置1280字节的专用缓存，设备模式时支持无晶振(crystal-less)
  - 红外发射器（IRTMR）
- **CRC计算单元**
- **96位的芯片唯一代码（UID）**
- **调试模式**
  - 串行单线调试(SWD)和JTAG接口
- **温度范围：-40至+105℃**
- **封装**
  - LQFP100 14 x 14 mm　LQFP64 10 x 10 mm
    LQFP64 7 x 7 mm　LQFP48 7 x 7 mm
    QFN48 6 x 6 mm　QFN36 6 x 6 mm
    QFN32 4 x 4 mm
- **选型列表**

闪存存储器	型号
64 K字节	AT32F423K8U7-4 AT32F423T8U7 AT32F423C8U7 AT32F423C8T7 AT32F423R8T7-7 AT32F423R8T7 AT32F423V8T7
128 K字节	AT32F423KBU7-4 AT32F423TBU7 AT32F423CBU7 AT32F423CBT7 AT32F423RBT7-7 AT32F423RBT7 AT32F423VBT7
256 K字节	AT32F423KCU7-4 AT32F423TCU7 AT32F423CCU7 AT32F423CCT7 AT32F423RCT7-7 AT32F423RCT7 AT32F423VCT7

目 录

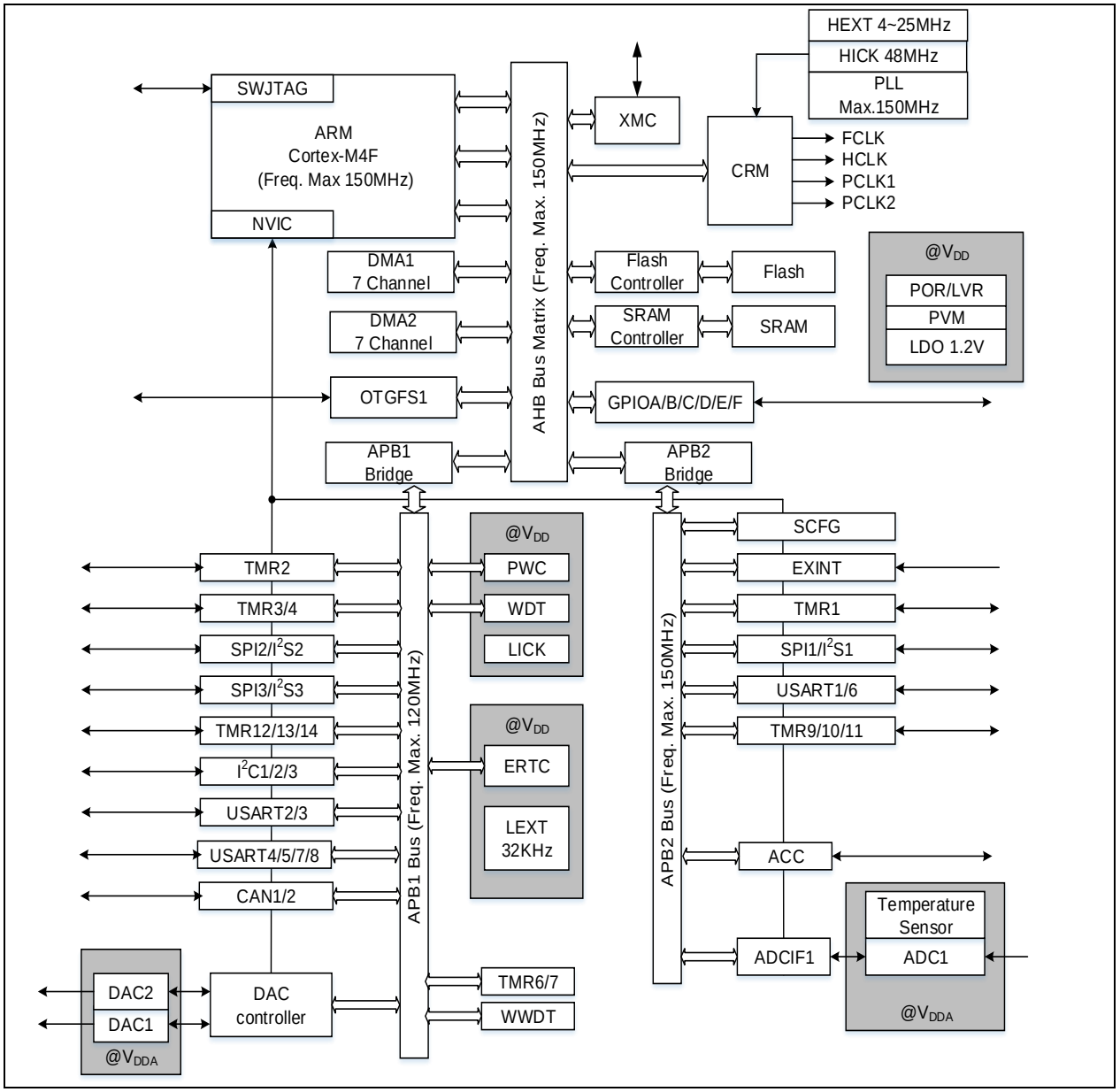
图目录

表目录

1 系统架构

AT32F423 系列微控制器内部集成了：32 位 ARM® Cortex®-M4F 处理器，多个 16 位和 32 位的定时器，红外线接口 IRTMR，DMA 控制器，实时时钟 ERTC，SPI 通信接口，I²C 通信接口，USART 通信接口，CAN 总线控制器，外部存储控制器 XMC，USB2.0 OTG 全速接口，HICK 自动时钟校准 ACC，12 位 ADC，12 位 DAC 和 PVM 模块等外设。大量的外设和存储器。Cortex®-M4F 处理器支持增强的高效 DSP 指令集，包含扩展的单周期 16/32 位乘法累加器（MAC）、双 16 位 MAC 指令、优化的 8/16 位 SIMD 运算及饱和运算指令，并且具有单精度（IEEE-754）浮点运算单元（FPU）。系统详细架构见下图。

图 1-1 AT32F423系列微控制器系统架构



1.1 系统概述

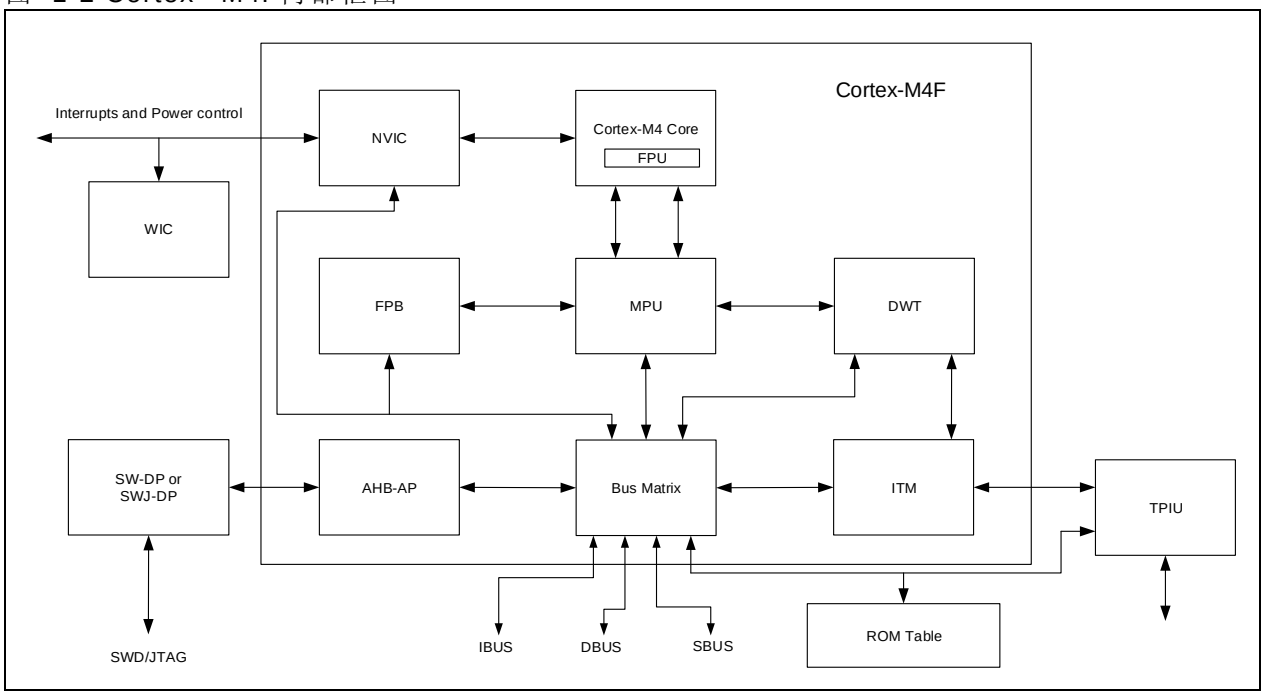
1.1.1 ARM Cortex®-M4F处理器

Cortex®-M4F 处理器是一款低功耗处理器，具有低门数，低中断延迟和低成本调试的特点。支持包括 DSP 指令集与浮点运算功能，特别适合用于深度嵌入式应用程序需要快速中断响应功能。Cortex®-M4F 处理器是基于 ARMv7-M 架构，既支持 Thumb 指令集也支持 DSP 指令集。

下图为 Cortex®-M4F 处理器的内部框图，请参阅《ARM® Cortex-M4 技术参考手册》了解关于 Cortex®-

M4F 更详尽信息

图 1-2 Cortex®-M4F内部框图



1.1.2 位带

利用位带操作，可以使用普通的加载/存储操作来对单一比特进行读写访问。在 Cortex®-M4F 中提供了两个位带区：SRAM 最低 1M 字节空间和外设区间的最低 1M 字节空间。这两个区中的地址除了可以像普通存储器一样访问外，还可以通过它们各自的位带别名区来快捷访问这两个区中任意地址的任意比特位，位带别名区将位带区每个比特膨胀成一个 32 位的字。当你访问位带别名区的一个地址时，等同于直接访问位带区的一个比特位。

图 1-3 位带区与位带别名区的膨胀关系图A

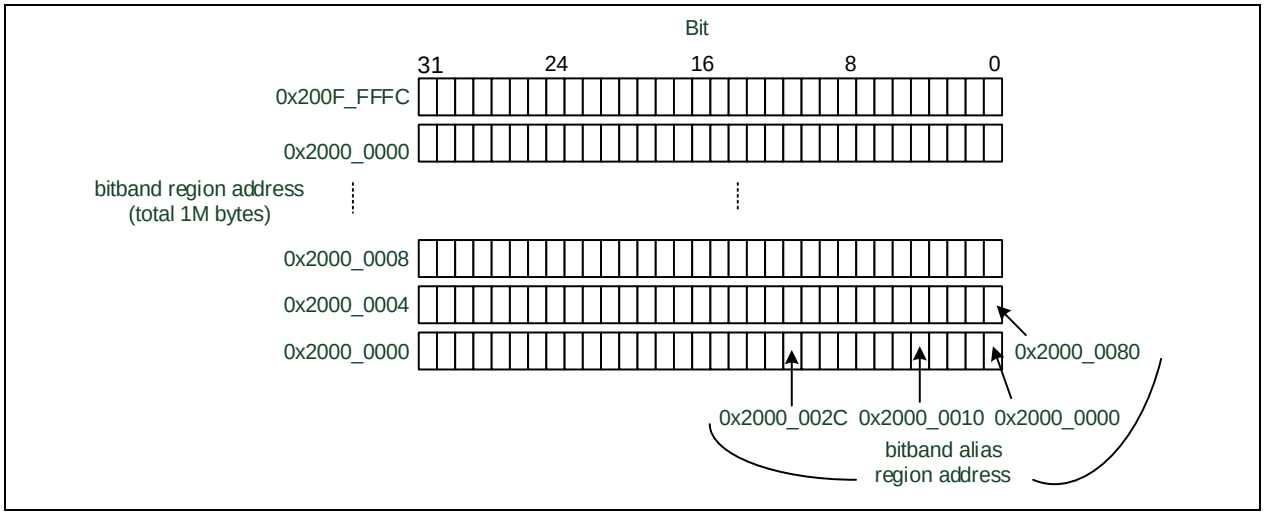
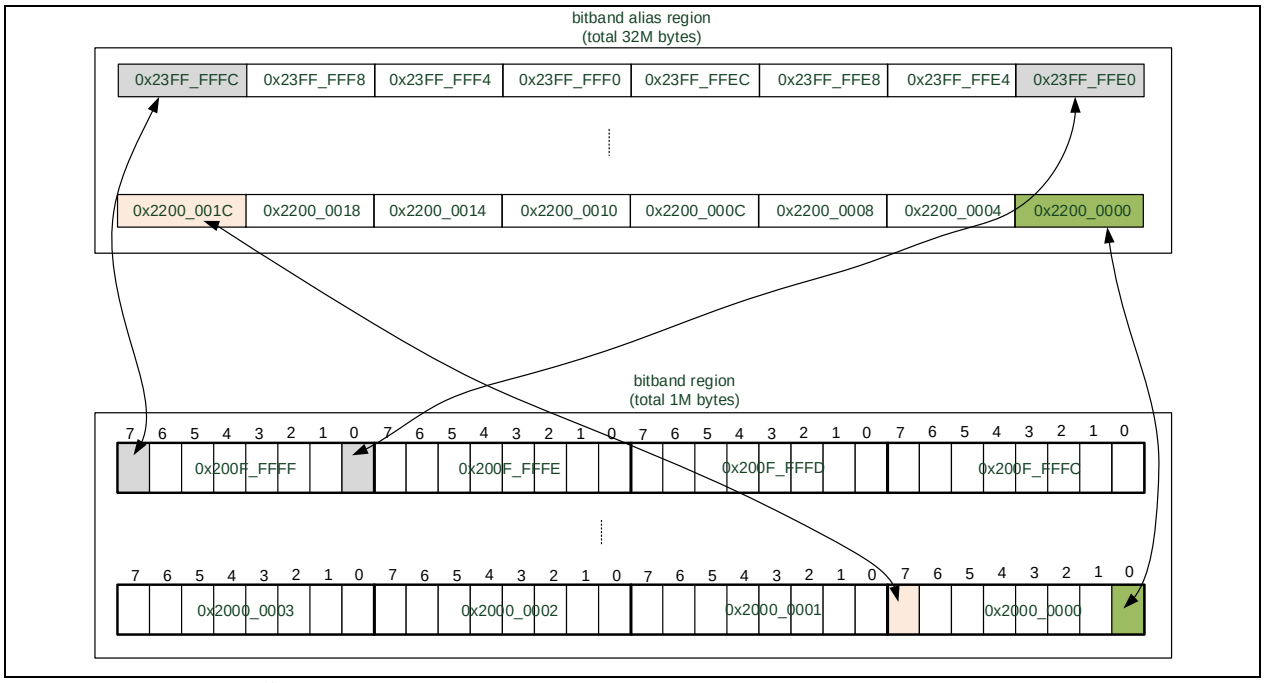


图 1-4 位带区与位带别名区的膨胀关系图B



位带区：支持位带操作的地址区

位带别名区：对别名区地址的访问最终作用到位带区的访问上

在位带区中，每个比特都映射到别名地址区的一个字（这是只有 LSB 有效的字）。当一个位带别名区地址被访问时，会先把该地址变换成位带区地址。对于读操作，读取位带区地址中的一个字，再把需要的位右移到 LSB，并把 LSB 返回。对于写操作，把需要写的位左移到对应的位序号处，然后执行一个比特级的“读-改-写”过程。

支持位带操作的两个内存区的地址范围为：

SRAM 区中的最低 1M 字节：0x2000_0000~0x200F_FFFF

外设区间的最低 1M 字节： 0x4000_0000~0x400F_FFFF

对于 SRAM 位带区的某个比特，如果所在字节地址为 A，位序号为 n(0<=n<=7)，则该比特在别名区的地址为：

AliasAddr = 0x2200_0000+ (A-0x2000_0000)*32+n*4

对于外设区间位带区的某个比特，如果所在字节地址为 A，位序号为 n(0<=n<=7)，则该比特在别名区的地址为：

AliasAddr = 0x4200_0000+ (A-0x4000_0000)*32+n*4

对于 SRAM 区中，位带区与位带别名区的映射如下表所示：

表 1-1 SRAM区中的位带地址映射

位带区	等效别名区地址
0x2000_0000.0	0x2200_0000.0
0x2000_0000.1	0x2200_0004.0
0x2000_0000.2	0x2200_0008.0
…	…
0x2000_0000.31	0x2200_007C.0
0x2000_0004.0	0x2200_0080.0
0x2000_0004.1	0x2200_0084.0
0x2000_0004.2	0x2200_0088.0
…	…
0x200F_FFFC.31	0x23FF_FFFC.0

对于外设区中，位带区与位带别名区的映射如下表所示：

表 1-2 外设区中的位带地址映射

位带区	等效别名区地址
0x4000_0000.0	0x4200_0000.0
0x4000_0000.1	0x4200_0004.0
0x4000_0000.2	0x4200_0008.0
…	…
0x4000_0000.31	0x4200_007C.0
0x4000_0004.0	0x4200_0080.0
0x4000_0004.1	0x4200_0084.0
0x4000_0004.2	0x4200_0088.0
…	…
0x400F_FFFC.31	0x43FF_FFFC.0

位带操作的优越性最容易想到的是通过 GPIO 的管脚来单独控制每盏 LED 的点亮与熄灭。另一方面，也对操作串行接口提供很大的方便。总之，位带操作对于硬件 I/O 密集型的底层程序最有用处。

位带操作还能简化跳转的判断。当跳转依据是某个位时，以前必须这样做：

- 读取整个寄存器
- 屏蔽不需要的位
- 比较并跳转

现在只需要：

- 从位带别名区读取该位的状态
- 比较并跳转

使代码更简洁，这只是位带操作优越性的初步体现，位带操作还有一个重要的好处是在多任务以及多任务环境中，将以前的读-改-写需要的三条指令，做成了一个硬件级别支持的原子操作，消除了以前读-改-写可能被中断，导致出现紊乱的情况。

1.1.3 中断和异常向量

下面列出了 AT32F423 产品的向量表。

表 1-3 AT32F423产品的向量表

位置	优先级	优先级类型	名称	说明	地址
	-	-	-	保留	0x0000_0000
	-3	固定	Reset	复位	0x0000_0004
	-2	固定	NMI	不可屏蔽中断 CRM 时钟失效检测（CFD）连接到 NMI 向量	0x0000_0008
	-1	固定	硬件失效（HardFault）	所有类型的失效	0x0000_000C
	0	可设置	存储管理（MemoryManage）	存储器管理	0x0000_0010
	1	可设置	总线错误（BusFault）	预取指失败，存储器访问失败	0x0000_0014
	2	可设置	错误应用（UsageFault）	未定义的指令或非法状态	0x0000_0018
	-	-	-	保留	0x0000_001C ~0x0000_002B
	3	可设置	SVCall	通过 SWI 指令的系统服务调用	0x0000_002C

	4	可设置	调试监控（DebugLENonitor）	调试监控器	0x0000_0030
	-	-	-	保留	0x0000_0034
	5	可设置	PendSV	可挂起的系统服务	0x0000_0038
	6	可设置	SysCNTRick	系统嘀嗒定时器	0x0000_003C
0	7	可设置	WWDT	窗口定时器中断	0x0000_0040
1	8	可设置	PVM	连到 EXINT 的电源电压检测（PVM）中断	0x0000_0044
2	9	可设置	TAMPER	侵入检测中断	0x0000_0048
3	10	可设置	ERTC_WKUP	实时时钟（ERTC）唤醒中断	0x0000_004C
4	11	可设置	FLASH	闪存全局中断	0x0000_0050
5	12	可设置	CRM	时钟和复位管理（CRM）中断	0x0000_0054
6	13	可设置	EXINT0	EXINT 线 0 中断	0x0000_0058
7	14	可设置	EXINT1	EXINT 线 1 中断	0x0000_005C
8	15	可设置	EXINT2	EXINT 线 2 中断	0x0000_0060
9	16	可设置	EXINT3	EXINT 线 3 中断	0x0000_0064
10	17	可设置	EXINT4	EXINT 线 4 中断	0x0000_0068
11	18	可设置	DMA1 通道 1	DMA1 通道 1 全局中断	0x0000_006C
12	19	可设置	DMA1 通道 2	DMA1 通道 2 全局中断	0x0000_0070
13	20	可设置	DMA1 通道 3	DMA1 通道 3 全局中断	0x0000_0074
14	21	可设置	DMA1 通道 4	DMA1 通道 4 全局中断	0x0000_0078
15	22	可设置	DMA1 通道 5	DMA1 通道 5 全局中断	0x0000_007C
16	23	可设置	DMA1 通道 6	DMA1 通道 6 全局中断	0x0000_0080
17	24	可设置	DMA1 通道 7	DMA1 通道 7 全局中断	0x0000_0084
18	25	可设置	ADC	ADC 的全局中断	0x0000_0088
19	26	可设置	CAN1_TX	CAN1 发送中断	0x0000_008C
20	27	可设置	CAN1_RX0	CAN1 接收 0 中断	0x0000_0090
21	28	可设置	CAN1_RX1	CAN1 接收 1 中断	0x0000_0094
22	29	可设置	CAN1_SE	CAN 状态错误中断	0x0000_0098
23	30	可设置	EXINT9_5	EXINT 线[9：5]中断	0x0000_009C
24	31	可设置	TMR1_BRK_TMR9	TMR1 停止中断和 TMR9 全局中断	0x0000_00A0
25	32	可设置	TMR1_OVF_TMR10	TMR1 溢出中断和 TMR10 全局中断	0x0000_00A4
26	33	可设置	TMR1_TRG_HALL_TMR11	TMR1 触发和 HALL 中断和 TMR11 全局中断	0x0000_00A8
27	34	可设置	TMR1_CH	TMR1 通道中断	0x0000_00AC
28	35	可设置	TMR2	TMR2 全局中断	0x0000_00B0
29	36	可设置	TMR3	TMR3 全局中断	0x0000_00B4

30	37	可设置	TMR4	TMR4 全局中断	0x0000_00B8
31	38	可设置	I2C1_EVT	I^2C1 事件中断	0x0000_00BC
32	39	可设置	I2C1_ERR	I^2C1 错误中断	0x0000_00C0
33	40	可设置	I2C2_EVT	I^2C2 事件中断	0x0000_00C4
34	41	可设置	I2C2_ERR	I^2C2 错误中断	0x0000_00C8
35	42	可设置	SPI1	SPI1 全局中断	0x0000_00CC
36	43	可设置	SPI2	SPI2 全局中断	0x0000_00D0
37	44	可设置	USART1	USART1 全局中断	0x0000_00D4
38	45	可设置	USART2	USART2 全局中断	0x0000_00D8
39	46	可设置	USART3	USART3 全局中断	0x0000_00DC
40	47	可设置	EXINT15_10	EXINT 线[15：10]中断	0x0000_00E0
41	48	可设置	ERTCAlarm	连到 EXINT 的 ERTC 闹钟中断	0x0000_00E4
42	49	可设置	OTGFS1_WKUP	连到 EXINT 的 OTGFS1 待机唤醒中断	0x0000_00E8
43	50	可设置	TMR12	TMR12 全局中断	0x0000_00EC
44	51	可设置	TMR13	TMR13 全局中断	0x0000_00F0
45	52	可设置	TMR14	TMR14 全局中断	0x0000_00F4
46	53	-	-	-	0x0000_00F8
47	54	-	-	-	0x0000_00FC
48	55	-	-	-	0x0000_0100
49	56	-	-	-	0x0000_0104
50	57	-	-	-	0x0000_0108
51	58	可设置	SPI3	SPI3 全局中断	0x0000_010C
52	59	可设置	USART4	USART4 全局中断	0x0000_0110
53	60	可设置	USART5	USART5 全局中断	0x0000_0114
54	61	可设置	TMR6_DAC	TMR6 全局中断 DAC1 和 DAC2 下溢错误中断	0x0000_0118
55	62	可设置	TMR7	TMR7全局中断	0x0000_011C
56	63	可设置	DMA2 通道 1	DMA2通道1全局中断	0x0000_0120
57	64	可设置	DMA2 通道 2	DMA2通道2全局中断	0x0000_0124
58	65	可设置	DMA2 通道 3	DMA2通道3全局中断	0x0000_0128
59	66	可设置	DMA2 通道 4	DMA2通道4全局中断	0x0000_012C
60	67	可设置	DMA2 通道 5	DMA2通道5全局中断	0x0000_0130
61	68	-	-	-	0x0000_0134
62	69	-	-	-	0x0000_0138
63	70	可设置	CAN2_TX	CAN2发送中断	0x0000_013C
64	71	可设置	CAN2_RX0	CAN2接收0中断	0x0000_0140

65	72	可设置	CAN2_RX1	CAN2接收1中断	0x0000_0144
66	73	可设置	CAN2_SE	CAN2状态错误中断	0x0000_0148
67	74	可设置	OTGFS1	OTGFS1全局中断	0x0000_014C
68	75	可设置	DMA2 通道 6	DMA2通道6全局中断	0x0000_0150
69	76	可设置	DMA2 通道 7	DMA2通道7全局中断	0x0000_0154
70	77	-	-	-	0x0000_0158
71	78	可设置	USART6	USART6全局中断	0x0000_015C
72	79	可设置	I2C3_EVT	I2C2事件中断	0x0000_0160
73	80	可设置	I2C3_ERR	I2C2错误中断	0x0000_0164
74	81	-	-	-	0x0000_0168
75	82	-	-	-	0x0000_016C
76	83	-	-	-	0x0000_0170
77	84	-	-	-	0x0000_0174
78	85	-	-	-	0x0000_0178
79	86	-	-	-	0x0-000_017C
80	87	-	-	-	0x0000_0180
81	88	可设置	FPU	FPU例外中断	0x0000_0184
82	89	可设置	USART7	USART7全局中断	0x0000_0188
83	90	可设置	USART8	USART8全局中断	0x0000_018C
84	91	-	-	-	0x0000_0190
85	92	-	-	-	0x0000_0194
86	93	-	-	-	0x0000_0198
87	94	-	-	-	0x0000_019C
88	95	-	-	-	0x0000_01A0
89	96	-	-	-	0x0000_01A4
90	97	-	-	-	0x0000_01A8
91	98	-	-	-	0x0000_01AC
92	99	-	-	-	0x0000_01B0
93	100	-	-	-	0x0000_01B4
94	101	可设置	DMAMUX	DMAMUX溢出中断	0x0000_01B8
95	102	-	-	-	0x0000_01BC
96	103	-	-	-	0x0000_01C0
97	104	-	-	-	0x0000_01C4
98	105	-	-	-	0x0000_01C8
99	106	-	-	-	0x0000_01CC

100	107	-	-	-	0x0000_01D0
101	108	-	-	-	0x0000_01D4
102	109	-	-	-	0x0000_01D8
103	110	可设置	ACC	ACC全局中断	0x0000_01DC

1.1.4 系统嘀嗒定时器（SysTick）

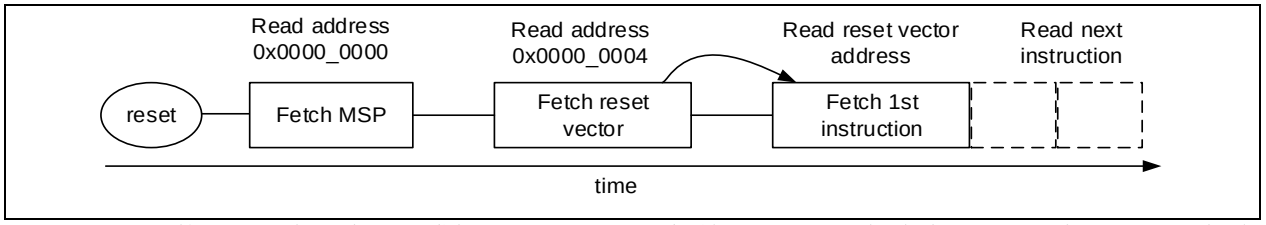
系统嘀嗒定时器是一个 24 位递减计数器，递减至零可自动重载计数初值。可产生周期性异常，用作嵌入式操作系统的多任务调度计数器，或对于无嵌入式操作系统，可用于调用需周期性执行的任务。系统嘀嗒定时器校准值固定值 9000，当系统嘀嗒时钟设定为 9MHz，产生 1ms 时间基准。

1.1.5 复位流程

系统复位后以及处理器开始执行程序前，处理器会从 CODE 存储器中读出前两个字。

- 从地址0x0000_0000处取出主栈指针（MSP）的初始值。
- 从地址0x0000_0004处取出程序计数器（PC）的初始值，这个值是复位向量，LSB必须是1。然后从这个值所对应的地址处取指。

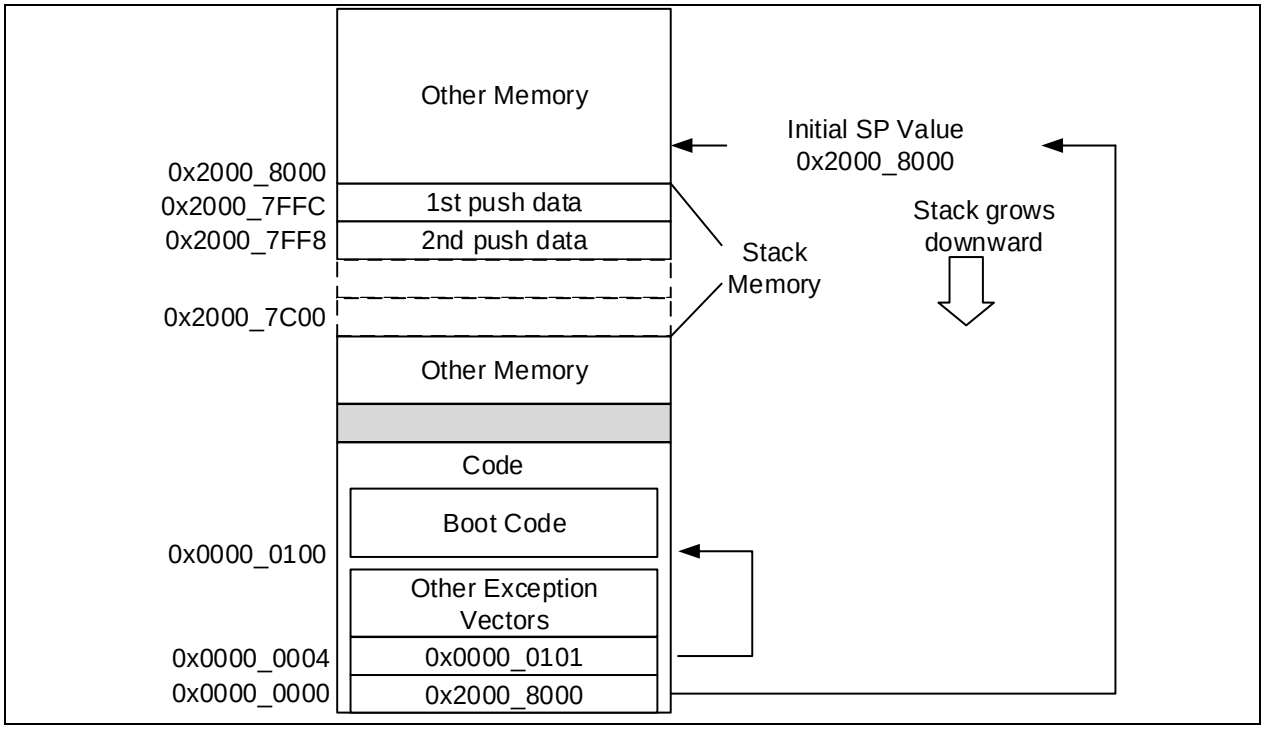
图 1-5 复位流程



Cortex®-M4F 使用的是向下生长的满栈，所以 MSP 的初始值必须是堆栈内存的末地址加 1。举例来说，堆栈区域设定在 0x2000_7C00~0x2000_7FFF 之间，那么 MSP 的初始值必须是 0x2000_8000。

向量表跟随在 MSP 的初始值之后。Cortex®-M4F 是在 Thumb 态下执行，所以向量表中的每个数值都必须将 LSB 置 1，所以，下图中使用 0x0000_0101 来表示地址 0x0000_0100。当 0x0000_0100 处的指令得到执行后，就正式开始程序的执行。在此之前初始化 MSP 是必须的，因为可能第一条指令还没执行就会被 NMI 或是其他 fault 打断。MSP 初始化好后就可以为它们的服务程序准备好堆栈空间。

图 1-6 MSP及PC初始化的一个范例



在 AT32F423 中，可以将主闪存存储器、启动程序存储器或片上 SRAM 这三块存储器重映射到 0x0000_0000~0x07FF_FFFF 的 CODE 区，nBOOT1 对应用户系统数据区（USD）里面系统配置字节

（SSB）位 nBOOT1 的值，由 nBOOT1 和 BOOT0 管脚来设定 CODE 从哪块存储器启动：
当{nBOOT1，BOOT0}=00/10 时，CODE 从主闪存存储器启动。
当{nBOOT1，BOOT0}=11 时，CODE 从启动程序存储器启动。
当{nBOOT1，BOOT0}=01 时，CODE 从片上 SRAM 启动。
系统复位后或从待机模式退出时，nBOOT1 和 BOOT0 管脚值都会被重新锁存。
当加载为片上 SRAM 启动模式后，BOOT 状态会被锁定，此时系统复位无法加载新的启动模式，必须上电复位后才能重新加载新的启动模式。
启动程序存储器中包含内嵌的 Bootloader 程序，可提供 flash 编程功能，通过 USART1、USART2 或 USB 接口对 flash 进行重新编程；也可以提供通信协议栈等额外的固件，可被软件开发人员通过 API 调用。

1.2 寄存器描述缩写说明

表 1-4 寄存器描述缩写说明

寄存器类型	说明
rw	可以读或写这些位
ro	只能读这些位
wo	只能写这些位；如果读这些位，则返回它们的复位值
rrc	可以读，读取这些位时，自动清除这些位
rw0c	可以读并写'0'清除这些位，写'1'将不对该位产生影响
rw1c	可以读并写'1'清除这些位，写'0'将不对该位产生影响
rw1s	可以读并写'1'设置这些位，写'0'将不对该位产生影响
tog	可以读，写'1'将翻转此位值，写'0'将不对该位产生影响
rwt	可以读，写任何值时，将触发事件
resd	保留

1.3 器件特征信息

表 1-5 器件特征信息相关寄存器地址和复位值

寄存器简称	基地址	复位值
F_SIZE	0x1FFF F7E0	0xXXXX
UID[31:0]	0x1FFF F7E8	0xXXXX XXXX
UID[63:32]	0x1FFF F7EC	0xXXXX XXXX
UID[95:64]	0x1FFF F7F0	0xXXXX XXXX

1.3.1 闪存容量寄存器

闪存容量寄存器提通该芯片闪存容量信息，用户可透过该寄存器取得闪存容量。

域	简称	复位值	类型	功能
位 15：0	F_SIZE	0xXXXX	ro	闪存容量，以 KByte 为单位 例如：0x0040 = 64KByte

1.3.2 器件电子签名

器件电子签名包含产品容量信息和器件唯一 ID（96 位 UID），它位于闪存的信息区块中。96 位器件唯一 ID 对任何器件来说都是独一无二的，且用户不可更改。ID 可以用来作为下列用途：

- 序列号；例如 USB 字串序列
- 或者做为密钥的一部分

域	简称	复位值	类型	功能
位 31：0	UID[31：0]	0xXXXX XXXX	ro	UID 的 bit31 到 bit0 信息

域	简称	复位值	类型	功能
位 31：0	UID[63：32]	0xXXXX XXXX	ro	UID 的 bit63 到 bit32 信息

域	简称	复位值	类型	功能
位 31：0	UID[95：64]	0xXXXX XXXX	ro	UID 的 bit95 到 bit64 信息

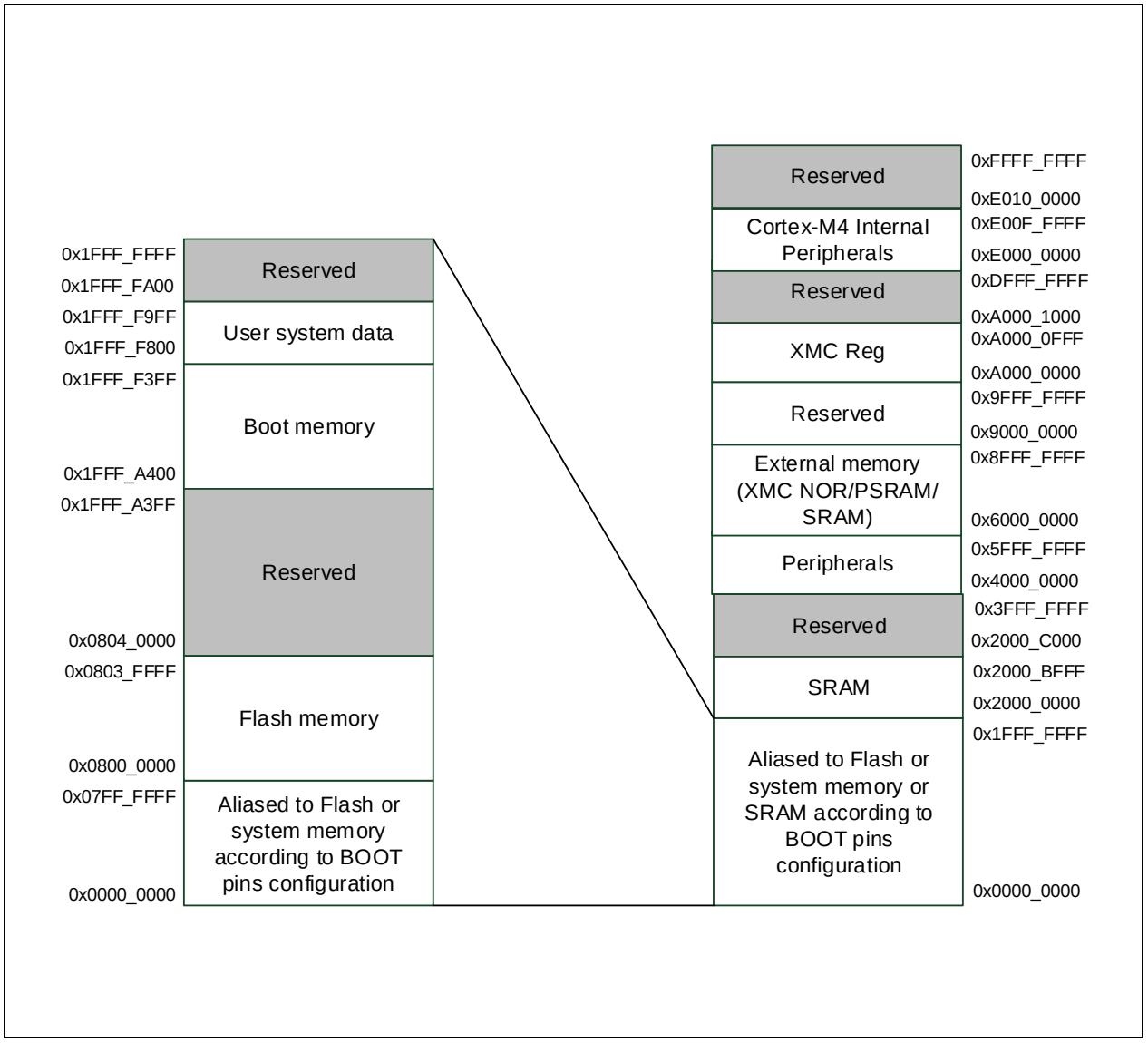
注：UID[95:88]为 Series ID，为 0x12。

2 存储器资源

2.1 内部存储器地址映射

芯片内部存储器包括程序存储器 flash，数据存储器 SRAM，外设寄存器和内核寄存器等。各区域地址映射如下图：

图 2-1 AT32F423地址配置



2.2 Flash存储器

AT32F423 系列提供最大 256KB 的片上闪存，支持单周期最大 32 位读取操作。

闪存存储器由闪存控制器操作，有关闪存控制器的操作与寄存器配置信息请参考第 5 章节。

闪存存储结构（256K）

主存储器只有闪存容量为 256K 字节的片 1 闪存，包含 128 个扇区，每扇区大小为 2K 字节。

表 2-1 256KB闪存模块的组织

结构		名称	地址范围
主存储器	片 1 (Bank1) 256KB	扇区 0	0x0800 0000 – 0x0800 07FF
		扇区 1	0x0800 0800 – 0x0800 0FFF
		扇区 2	0x0800 1000 – 0x0800 17FF

		扇区 3	0x0800 1800 – 0x0800 1FFF
		扇区 4	0x0800 2000 – 0x0800 27FF
		.	.
		扇区 127	0x0803 F800 – 0x0803 FFFF
信息块		启动程序代码区 20KB	0x1FFF A400 – 0x1FFF F3FF
		用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

闪存存储结构（128K）

主存储器只有闪存容量为 128K 字节的片 1 闪存，包含 128 个扇区，每扇区大小为 1K 字节。

表 2-2 128KB闪存模块的组织

结构		名称	地址范围
主存储器	片 1 (Bank1) 128KB	扇区 0	0x0800 0000 – 0x0800 03FF
		扇区 1	0x0800 0400 – 0x0800 07FF
		扇区 2	0x0800 0800 – 0x0800 0BFF
		扇区 3	0x0800 0C00 – 0x0800 0FFF
		扇区 4	0x0800 1000 – 0x0800 13FF
		.	.
		扇区 127	0x0801 FC00 – 0x0801 FFFF
信息块		启动程序代码区 20KB	0x1FFF A400 – 0x1FFF F3FF
		用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

闪存存储结构（64K）

主存储器只有闪存容量为 64K 字节的片 1 闪存，包含 64 个扇区，每扇区大小为 1K 字节。

表 2-3 64KB闪存模块的组织

结构		名称	地址范围
主存储器	片 1 (Bank1) 64KB	扇区 0	0x0800 0000 – 0x0800 03FF
		扇区 1	0x0800 0400 – 0x0800 07FF
		扇区 2	0x0800 0800 – 0x0800 0BFF
		扇区 3	0x0800 0C00 – 0x0800 0FFF
		扇区 4	0x0800 1000 – 0x0800 13FF
		.	.
		扇区 63	0x0800 FC00 – 0x0800 FFFF
信息块		启动程序代码区 20KB	0x1FFF A400 – 0x1FFF F3FF
		用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

2.3 SRAM存储器

AT32F423 系列内置最高可达 48K 字节的片上 SRAM，起始地址为 0x2000_0000。它可以以字节、半字（16 位）或全字（32 位）访问。

2.4 外设地址映射

表 2-4 各外设起始地址

总线	起始地址	外设
AHB	0xC000 0000 - 0xFFFF FFFF	保留
	0xB000 0000 - 0xBFFF FFFF	保留
	0xA000 1000 - 0xAFFF FFFF	保留
	0xA000 0000 - 0xA000 0FFF	XMC_REG

	0x9000 0000 - 0x9FFF FFFF	保留
	0x6000 0000 - 0x8FFF FFFF	XMC
	0x5004 0000 - 0x5FFF FFFF	保留
	0x5000 0000 - 0x5003 FFFF	OTG_FS1
	0x4002 6800 - 0x4FFF FFFF	保留
	0x4002 6400 - 0x4002 67FF	DMA2
	0x4002 6000 - 0x4002 63FF	DMA1
	0x4002 4000 - 0x4002 5FFF	保留
	0x4002 3C00 - 0x4002 3FFF	闪存存储器接口（FLASH）
	0x4002 3800 - 0x4002 3BFF	时钟和复位管理（CRM）
	0x4002 3400 - 0x4002 37FF	保留
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2000 - 0x4002 2FFF	保留
	0x4002 1C00 - 0x4002 1FFF	保留
	0x4002 1800 - 0x4002 1BFF	保留
	0x4002 1400 - 0x4002 17FF	GPIO 端口 F
	0x4002 1000 - 0x4002 13FF	GPIO 端口 E
	0x4002 0C00 - 0x4002 0FFF	GPIO 端口 D
	0x4002 0800 - 0x4002 0BFF	GPIO 端口 C
	0x4002 0400 - 0x4002 07FF	GPIO 端口 B
	0x4002 0000 - 0x4002 03FF	GPIO 端口 A
APB2	0x4001 8000 - 0x4001 FFFF	保留
	0x4001 7C00 - 0x4001 7FFF	保留
	0x4001 7800 - 0x4001 7BFF	保留
	0x4001 7400 - 0x4001 77FF	ACC
	0x4001 4C00 - 0x4001 73FF	保留
	0x4001 4800 - 0x4001 4BFF	TMR11 定时器
	0x4001 4400 - 0x4001 47FF	TMR10 定时器
	0x4001 4000 - 0x4001 43FF	TMR9 定时器
	0x4001 3C00 - 0x4001 3FFF	EXINT
	0x4001 3800 - 0x4001 3BFF	SCFG
	0x4001 3400 - 0x4001 37FF	保留
	0x4001 3000 - 0x4001 33FF	SPI1/I2S1
	0x4001 2400 - 0x4001 2FFF	保留
	0x4001 2000 - 0x4001 23FF	ADC
	0x4001 1800 - 0x4001 1FFF	保留
	0x4001 1400 - 0x4001 17FF	USART6
	0x4001 1000 - 0x4001 13FF	USART1
	0x4001 0800 - 0x4001 0FFF	保留
	0x4001 0400 - 0x4001 07FF	保留
	0x4001 0000 - 0x4001 03FF	TMR1 定时器

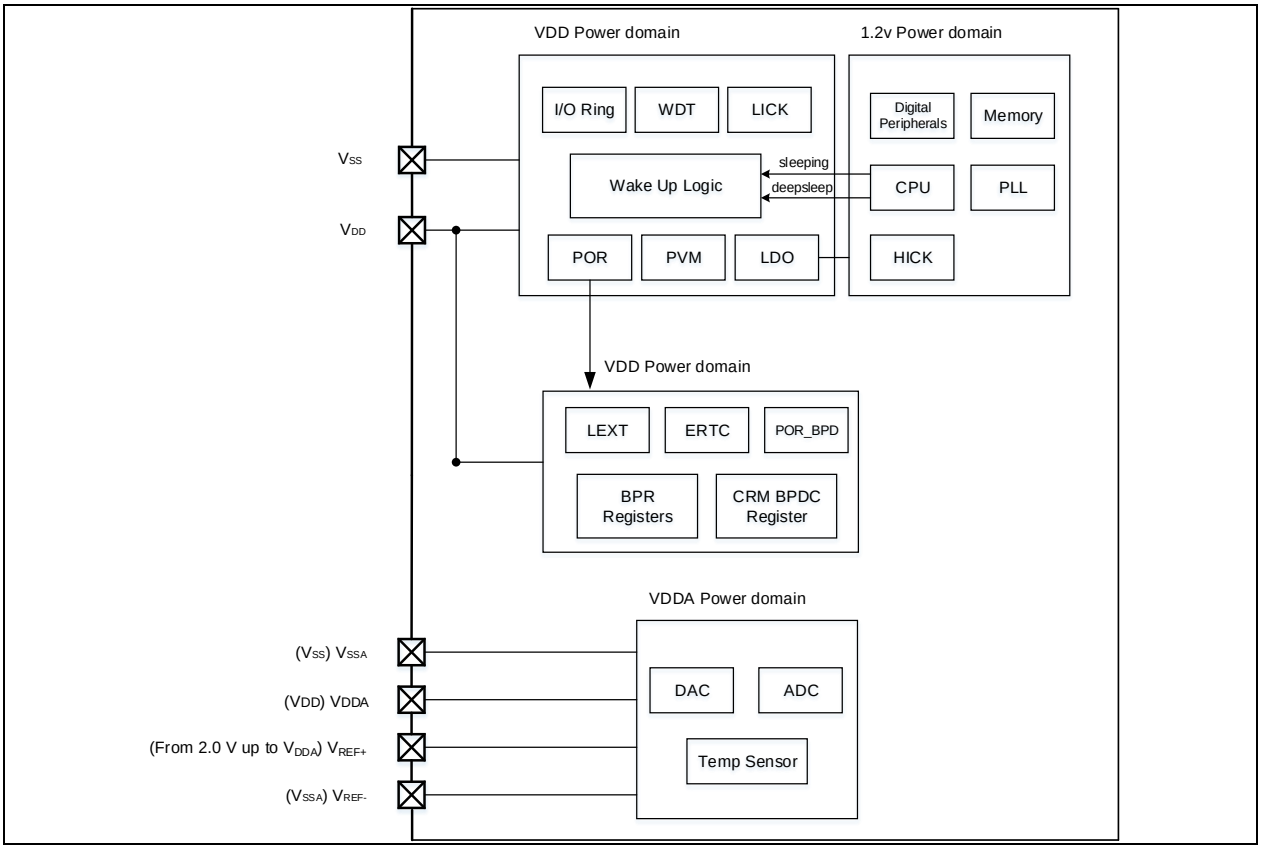
APB1	0x4000 8000 - 0x4000 FFFF	保留
	0x4000 7C00 - 0x4000 7FFF	USART8
	0x4000 7800 - 0x4000 7BFF	USART7
	0x4000 7400 - 0x4000 77FF	DAC
	0x4000 7000 - 0x4000 73FF	电源控制（PWC）
	0x4000 6C00 - 0x4000 6FFF	保留
	0x4000 6800 - 0x4000 6BFF	CAN2
	0x4000 6400 - 0x4000 67FF	CAN1
	0x4000 6000 - 0x4000 63FF	保留
	0x4000 5C00 - 0x4000 5FFF	I2C3
	0x4000 5800 - 0x4000 5BFF	I2C2
	0x4000 5400 - 0x4000 57FF	I2C1
	0x4000 5000 - 0x4000 53FF	USART5
	0x4000 4C00 - 0x4000 4FFF	USART4
	0x4000 4800 - 0x4000 4BFF	USART3
	0x4000 4400 - 0x4000 47FF	USART2
	0x4000 4000 - 0x4000 43FF	保留
	0x4000 3C00 - 0x4000 3FFF	SPI3/I2S3
	0x4000 3800 - 0x4000 3BFF	SPI2/I2S2
	0x4000 3400 - 0x4000 37FF	保留
	0x4000 3000 - 0x4000 33FF	看门狗（WDT）
	0x4000 2C00 - 0x4000 2FFF	窗口看门狗（WWDT）
	0x4000 2800 - 0x4000 2BFF	ERTC
	0x4000 2400 - 0x4000 27FF	保留
	0x4000 2000 - 0x4000 23FF	TMR14 定时器
	0x4000 1C00 - 0x4000 1FFF	TMR13 定时器
	0x4000 1800 - 0x4000 1BFF	TMR12 定时器
	0x4000 1400 - 0x4000 17FF	TMR7 定时器
	0x4000 1000 - 0x4000 13FF	TMR6 定时器
	0x4000 0C00 - 0x4000 0FFF	保留
	0x4000 0800 - 0x4000 0BFF	TMR4 定时器
	0x4000 0400 - 0x4000 07FF	TMR3 定时器
	0x4000 0000 - 0x4000 03FF	TMR2 定时器

3 电源控制（PWC）

3.1 简介

AT32F423 系列设备工作电压范围为 2.4V 至 3.6V，正常工作温度范围为-40~+105℃。AT32F423 系列设备为了降低功耗，使用户可以在 CPU 运行时间要求、速度和功耗进行折中取舍，提供了三种省电模式——睡眠模式，深度睡眠模式和待机模式。AT32F423 系列设备有两个电源域——VDD/VDDA 域，1.2V 域。其中 VDD/VDDA 域由电源直接供电，1.2V 域由 VDD/VDDA 域中嵌入的 LDO 供电。

图 3-1各电源域框图



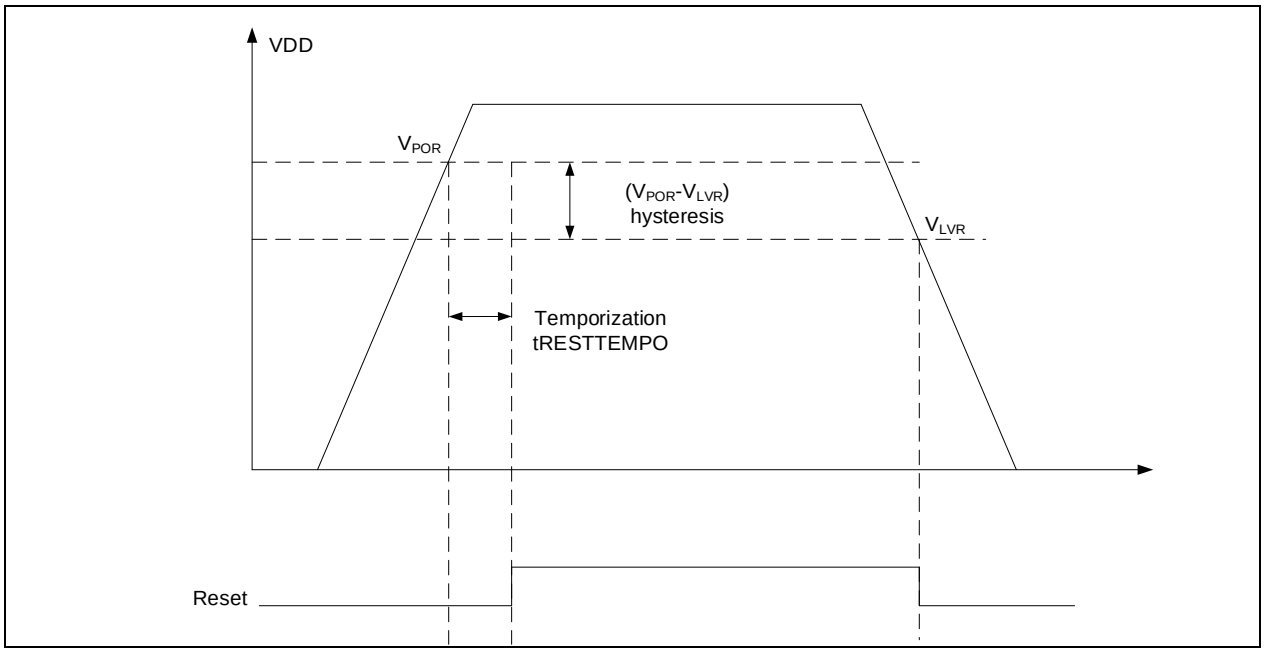
3.2 主要特点

- 具备两个电源域：VDD/VDDA 域、1.2V 内核域。
- 支持三种省电模式：睡眠模式、深度睡眠模式和待机模式。
- 内建电压调节器，提供 1.2V 给内核域。
- 提供电压监测器，能在电压低于阈值或高于阈值时产生中断或事件。
- VDD/VDDA 采用独立的数字和模拟地，用于隔离电源噪声。

3.3 上电下电复位

VDD/VDDA 域内置一个 POR 模拟模块用于产生电源复位，当 VDD 由 0V 上升至工作电压过程中，电源复位信号在 V_{POR} 时刻被上电释放。当 VDD 由工作电压下降至 0V 过程中，电源复位信号在 V_{LVR} 时刻被低电压复位。上电复位过程，复位信号的释放相较于 VDD 升压过程存在一定的时间延迟，同时上电低电压复位具有一定迟滞。

图 3-2上电/低电压复位波形图

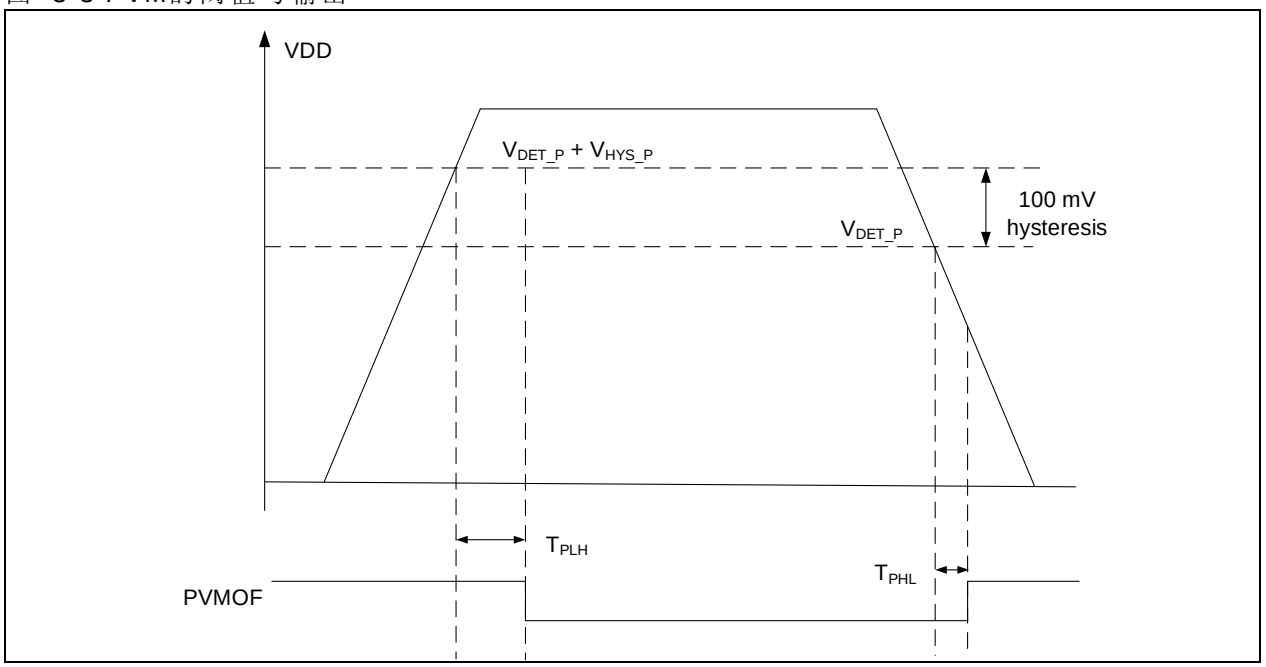


3.4 电压监测器（PVM）

电压监测器 PVM 主要用来监控供电电源的跳变，可通过电源控制寄存器（PWC_CTRL）中的 PVMEN 位开启电压监测功能，并通过 PVMSEL[2：0]来选择监控阈值。

电压监测器开启后，电源控制及状态寄存器（PWC_CTRLSTS）中的 PVMOF 位会指示 VDD 与设定阈值比较的结果，迟滞电压 VHYS_P 为 100mv。当 VDD 越过 PVM 阈值边界时，产生的 PVMOF 位电平变化可以通过外部中断第 16 号线产生 PVM 中断。

图 3-3 PVM的阈值与输出



3.5 电源域划分

1.2V 域

1.2V 内核域包括 CPU 内核、存储器 SRAM、内嵌数字外设以及时钟锁相环 PLL，由 LDO（电压调节器）供电。

VDD/VDDA 域

VDD/VDDA 域包括 VDD 域和 VDDA 域两部分。VDD 域包括 I/O 电路、省电模式唤醒电路、看门狗 WDT、上电/低电压复位（POR/LVR）、电压调节器 LDO、ERTC 电路、LEXT 振荡器以及所有 PAD 电路等。VDDA 域包括 DAC/ADC（DA/AD 转换器）、温度传感器 Temp Sensor 等。
一般来说，为保证低电压时 ADC/DAC 的高精度，数字电路由 VDD 供电，模拟电路由 VDDA 独立供电，在 64 PIN 封装及以下型号中，外部参考电压 VREF+连接至 VDDA 管脚，VREF-连接至 VSSA 管脚。
在芯片正常工作（Run Mode）时，电压调节器（LDO）为 1.2V 域供电。可配置 LDO 调校寄存器（PWC_LDOOV）来选择 LDO 的输出电压，不同的输出电压，系统可以工作的最高工作频率不同，详细参见 AT32F423 数据手册。
注意：限定在系统时钟使用 HEXT 或 HICK 时方可改变 LDO 的输出电压。

LDO 输出电压调节
1） 系统时钟切换至HICK或HEXT
2） 修改LDO电压（ LDOOVSEL[1：0]）
3） 设置闪存性能选择寄存器（FLASH_PSR）
4） 设置PLL相关寄存器至目标频率，开启PLL，等待PLL_STBL
5） 设置AHB及APB预除频系数
6） 若PLL频率大于108MHz，打开顺滑切换
7） 切换系统时钟至PLL

3.6 省电模式

当 CPU 无需继续运行时，AT32F423 支持三种低功耗模式（睡眠模式、深度睡眠模式、待机模式）可以实现更低的功耗。用户可以在启动时间，唤醒源，电源消耗等方面进行折中。此外在运行模式下，还可以通过降低系统时钟或关闭 APB 和 AHB 总线上未被使用的外设时钟来降低功耗。

睡眠模式（Sleep Mode）
执行 WFI 或 WFE 指令可以进入睡眠状态。结合 Cortex®-M4F 系统控制寄存器中的 SLEEPONEXIT 位的设定，提供两种进入睡眠模式的机制：
SLEEP-NOW 模式
当 SLEEPDEEP=0，SLEEPONEXIT=0 时，执行 WFI 或 WFE 指令，此时可立即进入睡眠模式。
SLEEP-ON-EXIT 模式
当 SLEEPDEEP=0，SLEEPONEXIT=1 时，执行 WFI 指令，此时当系统从最低优先级的中断处理程序中退出时，可立即进入睡眠模式。
在睡眠模式下，CPU 时钟关闭，其他时钟均正常工作，电压调节器正常工作，所有的 I/O 管脚都保持它们在运行模式时的状态，调节器 LDO 以正常功耗模式提供电源（CPU 内核、内存和内嵌外设），由 LDO 调校寄存器（PWC_LDOOV）来配置 LDO 的输出电压。
1) 执行WFI指令进入睡眠模式时，只要产生外设中断，都能使系统退出睡眠模式。
2) 执行WFE指令进入睡眠模式时，存在两种方式的唤醒事件，使系统退出睡眠模式：
- 使能任一外设中断（未在 NVIC 中使能）且使能 SEVONPEND 位可以产生唤醒事件。系统唤醒后，需清除外设中断挂起位和 NVIC 通道挂起位。
- 配置内部 EXINT 线为事件模式来产生唤醒事件。
- 从执行WFE指令进入睡眠模式唤醒所需的时间最短，因为没有时间损失在中断的进入或退出上。

深度睡眠模式（Deepsleep Mode）
通过设置 Cortex®-M4F 系统控制寄存器中的 SLEEPDEEP 位，清除电源控制寄存器（PWC_CTRL）中的 LPSEL 位，再执行 WFI 或 WFE 指令即可进入深度睡眠模式。
还可以通过设置电源控制寄存器（PWC_CTRL）中 VRSEL 位选择深度睡眠模式下电压调节器的工作状态。当 VRSEL=0，电压调节器正常工作，当 VRSEL=1，电压调节器处于低功耗模式。
另外，深度睡眠模式下，在设置 VRSEL=1，控制电压调节器处于低功耗模式的基础上，可以通过设置内部电压调节器额外低功耗模式使能位（VREXLPEN），进一步降低深度睡眠模式下整个系统的功耗。
在深度睡眠模式下，所有 1.2V 时钟关闭，HICK 和 HEXT 振荡器都被关闭，电压调节器以正常工作或低功耗工作状态给 1.2V 域供电，所有 I/O 管脚都保持它们在运行模式时的状态，SRAM 和寄存器内容保持。

1) 执行WFI指令进入深度睡眠模式，任一外部中断线在中断模式下产生的中断，即可使系统退出深度睡眠模式。
2) 如果执行WFE指令进入深度睡眠模式，任一外部中断线在事件模式下产生的事件，即可使系统退出深度睡眠模式。

系统从深度睡眠模式退出时，HICK RC 振荡器开启并在稳定后被选为系统时钟。当电压调节器处于低功耗模式时，退出深度睡眠模式时，需要额外等待电压调节器稳定，从而会增加一段额外的唤醒时间。

低功耗 deepsleep LDO 电压调节流程（注：sleep 和 standby 没有此限制）

1） 系统时钟切换至HICK
2） 修改LDO 电压（LDOOVSEL[1：0]）为1.0V以及VREXLPEN位
3） 配置LDO是否工作在低功耗模式（VRSEL）
4） 系统进入deepsleep状态
5） 系统退出deepsleep状态（满足唤醒条件后）
6） 修改LDO 电压（LDOOVSEL[1：0]）
7） 设置闪存性能选择寄存器（FLASH_PSR）
8） 若PLL时钟源为HEXT，需要开启HEXT并等待HEXTSTBL
9） 设置PLL相关寄存器至目标频率
10） 开启PLL，等待PLL_STBL
11） 设置AHB及APB预除频系数
12） 若PLL频率大于108MHz，打开顺滑切换
13） 切换系统时钟至PLL

注：若期望低功耗唤醒后保持进低功耗前的时钟状态，前述 7/9/11 步骤可省略。

待机模式（Standby Mode）

待机模式可最大限度的降低系统功耗，在该模式下，电压调节器关闭，只有 VDD/VDDA 域维持供电，其他的 1.2V 供电区域，PLL、HICK 和 HEXT 振荡器都被断电。寄存器和 SRAM 中的内容也会丢失。

通过设置 Cortex®-M4F 系统控制寄存器中的 SLEEPDEEP 位，设置电源控制寄存器(PWC_CTRL)中 LPSEL 位，并清除电源控制及状态寄存器（PWC_CTRLSTS）中的 SWEF 位的情况下，最后执行 WFI 或 WFE 指令即可进入待机模式。

在待机模式下，电池供电域数据寄存器、ERTC 相关寄存器和待机电路维持供电。

在待机模式下，除了复位管脚、被设置为防侵入或校准输出时的 TAMPER 管脚和被使能的唤醒管脚之外，所有的 I/O 管脚处于高阻态。

当产生 WKUP 管脚的上升沿、ERTC 闹钟事件的上升沿、ERTC 入侵事件、ERTC 时间戳、ERTC 周期性自动唤醒、NRST 管脚上外部复位、WDT 复位时，微控制器将退出待机模式。

调试配置

默认情况下，在进行调试时，微处理器一旦进入深度睡眠或待机模式，会因为 Cortex®-M4F 的内核失去了时钟而失去调试连接。只需通过设置 DEBUG 控制寄存器（DEBUG_CTRL）中的某些配置位，就可以在低功耗模式下继续调试软件。

3.7 PWC寄存器

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

表 3-1 PWC寄存器的映像和复位值

寄存器简称	基址偏移量	复位值
PWC_CTRL	0x00	0x0000 0000
PWC_CTRLSTS	0x04	0x0000 0000
PWC_LDOOV	0x10	0x0000 0012

3.7.1 电源控制寄存器（PWC_CTRL）

域	简称	复位值	类型	功能
位 31：9	保留	0x000000	resd	保持默认值。
位 8	BPWEN	0x0	rw	电池供电区域的写入使能（Battery powered domain write enable） 0：关闭； 1：开启。 注： 复位后，电池供电区域禁止写入。要对电池供电区域进行写操作的话，需先设置这位为允许写入状态。
位 7：5	PVMSEL	0x0	rw	电压监测临界值选择（Power voltage monitoring boundary select） 000：未用，禁止配置。 001：2.3V 010：2.4V 011：2.5V 100：2.6V 101：2.7V 110：2.8V 111：2.9V
位 4	PVMEN	0x0	rw	电压监测使能（Power voltage monitoring enable） 0：关闭； 1：开启。
位 3	CLSEF	0x0	wo	清除 SEF 标志（Clear SEF flag） 0：无效； 1：清除 SEF 标志。 注：该位在清除 SEF 后由硬件将其清零，且任何时刻读取该位返回值均是零。
位 2	CLSWEF	0x0	wo	清除 SWEF 标志（Clear SWEF flag） 0：无效； 1：清除 SWEF 标志。 注： 实际 SWEF 标志的清除大约需要 2 个系统时钟周期； 该位在清除 SWEF 后由硬件将其清零，且任何时刻读取该位返回值均是零。
位 1	LPSEL	0x0	rw	SLEEPDEEP 状态下的低功耗模式选择位（Low power mode select when Cortex®-M4F sleepdeep） 0：进入 DEEPSLEEP 模式； 1：进入待机模式。
位 0	VRSEL	0x0	rw	DEEPSLEEP 模式下电压调节器状态选择（Voltage regulator state select when deepsleep mode） 0：正常开启 ； 1：低功耗模式。

3.7.2 电源控制及状态寄存器（PWC_CTRLSTS）

域	简称	复位值	类型	功能
位 31：15	保留	0x0000 0	resd	保持默认值。
位 14	SWPEN7	0x0	rw	待机唤醒引脚 7 使能（Standby wake-up pin7 enable） 0：关闭（该引脚可用作通用 I/O）； 1：开启（该引脚被强置为输入下拉模式，且无法再用作通用 I/O）。 注：在系统复位时硬件将清除这一位。

位 13	SWPEN6	0x0	rw	待机唤醒引脚 6 使能（Standby wake-up pin6 enable） 0：关闭（该引脚可用作通用 I/O）； 1：开启（该引脚被强置为输入下拉模式，且无法再用作通用 I/O）。 注：在系统复位时硬件将清除这一位。
位 12：10	保留	0x0	resd	保持默认值。
位 9	SWPEN2	0x0	rw	待机唤醒引脚 2 使能（Standby wake-up pin2 enable） 0：关闭（该引脚可用作通用 I/O）； 1：开启（该引脚被强置为输入下拉模式，且无法再用作通用 I/O）。 注：在系统复位时硬件将清除这一位。
位 8	SWPEN1	0x0	rw	待机唤醒引脚 1 使能（Standby wake-up pin1 enable） 0：关闭（该引脚可用作通用 I/O）； 1：开启（该引脚被强置为输入下拉模式，且无法再用作通用 I/O）。 注：在系统复位时硬件将清除这一位。
位 7：3	保留	0x00	resd	保持默认值。
位 2	PVMOF	0x0	ro	电源电压监测输出标志（Power voltage monitoring output flag） 0：电源电压高于临界值； 1：电源电压低于临界值。 注：待机模式下电压监测停止工作。
位 1	SEF	0x0	ro	进入待机模式标志（Standby mode entry flag） 0：未进过待机模式； 1：有进过待机模式。 注：该位被硬件置起（进入待机模式时），由 POR/LVR 或写 CLSEF 位将其清零。
位 0	SWEF	0x0	ro	待机唤醒事件标志（Standby wake-up event flag） 0：无唤醒事件产生； 1：有唤醒事件产生。 注： 该位被硬件置起（产生唤醒事件时），由 POR/LVR 或写 CLSWEF 位将其清零。 唤醒事件将由以下几种情况产生： 在待机唤醒引脚上出现上升沿时，将产生唤醒事件； 出现 ERTC 闹钟事件时，将产生唤醒事件； 待机唤醒引脚保持高电平期间使能该待机唤醒引脚，将产生唤醒事件。

3.7.3 LDO调校寄存器（PWC_LDOOV）

域	简称	复位值	类型	功能
位 31：5	保留	0x0000000	resd	保持默认值。
位 4	VREXLPEN	0x1	rw	内部电压调节器额外低功耗模式使能位（Voltage regulator extra low power mode enable） 与电源控制寄存器（PWC_CTRL）的 LPSEL 和 VRSEL 位协同工作，在 VRSEL = 1 时，且芯片进入深度睡眠模式时才有效。 0：内部电压调节器额外低功耗模式关闭。 1：内部电压调节器额外低功耗模式开启。 注：如需启动额外低功耗模式，请先配置 VREXLPEN，再配置 LPSEL 和 VRSEL 位。

位 3：2	保留	0x0	resd	保持默认值。
位 1：0	LDOOVSEL	0x2	rw	LDOOVSEL：内部电压调节器 LDO 输出电压选择（LDO output voltage select） 00：1.0V 01：保留 10：1.2V 11：1.3V

4 时钟和复位管理（CRM）

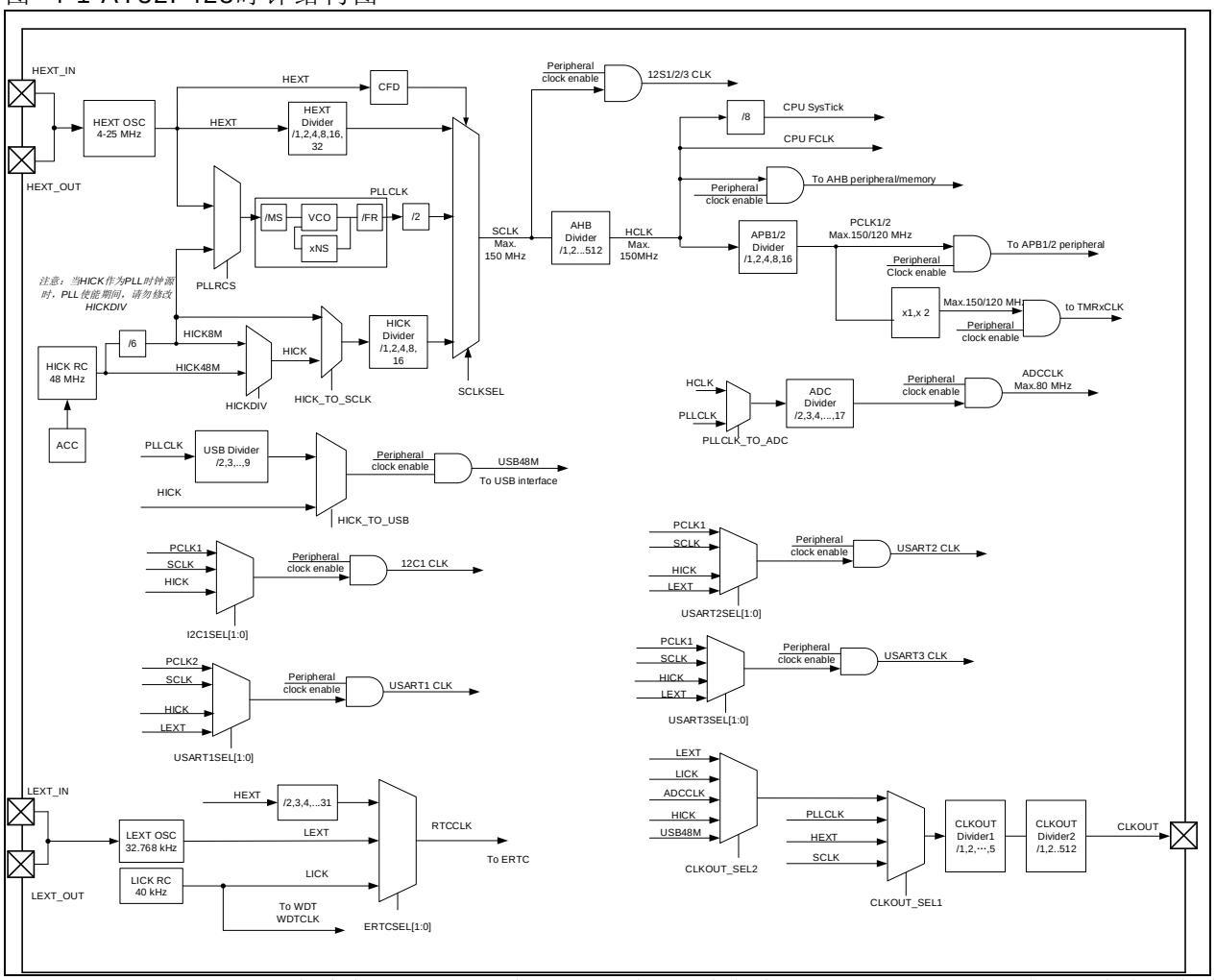
4.1 时钟

AT32F423 的时钟源包含：

- HEXT(high speed external crystal)高速外部晶振
- HICK (high speed internal clock) 高速内部时钟
- PLL(phase-locked loops)锁相环时钟
- LEXT(low speed external crystal) 低速外部晶振
- LICK (low speed internal clock) 低速内部时钟

时钟结构如下：

图 4-1 AT32F423时钟结构图



AHB、APB2 和 APB1 的频率都支持多种分频。AHB 域的最大频率是 150MHz，APB1 域的最大允许频率是 120MHz，APB2 域的最大允许频率是 150MHz。

4.1.1 时钟源

- HEXT 振荡器时钟

包括 HEXT 晶体/陶瓷谐振器和 HEXT 旁路时钟两个时钟源。

HEXT 晶体/陶瓷谐振器外接一颗频率范围为 4~25MHz HEXT 的晶体，可为系统提供高精度的时钟。HEXT 时钟直到时钟稳定后才会被释放出来。

HEXT 旁路时钟可以提供频率高达 25MHz 的外部时钟。外部时钟信号必须连到 HEXT_IN 引脚，HEXT_OUT 引脚可以释放给 GPIO 控制。

- HICK 振荡器时钟

HICK 振荡器时钟由芯片内的高速 RC 振荡器提供。HICK 时钟的内部频率为 48MHz，频率精度较差，但启动时间比 HEXT 晶体振荡器短，每颗芯片的 HICK 时钟频率在出厂前已经被校准到 1%（25° C），工厂校准值被装载到时钟控制寄存器的 HICKCAL[7：0]位。考虑不同的电压或环境温度对 HICK 的 RC 振荡器的影响，用户可以通过时钟控制寄存器里的 HICKTRIM[5：0]位来调整 HICK 频率。
HICK 时钟直到稳定后才会被释放出来。

当 HICK 作为系统时钟时，去更改 CRM_MISC2 寄存器中的 HICK_TO_SCLK 位，需要遵循如下的配置流程：
1. 配置 CRM_MISC2 寄存器中的 HICK_TO_SCLK_DIV 位为 0x4
2. 等待 50 个系统时钟
3. 更改 CRM_MISC2 寄存器中的 HICK_TO_SCLK 值
4. 重新配置 HICK_TO_SCLK_DIV 位为需要设定的值

当 HICK 作为系统时钟时，去更改 CRM_MISC1 寄存器中的 HICKDIV 位，需要遵循如下的配置流程：
1. 配置 CRM_MISC2 寄存器中的 HICK_TO_SCLK_DIV 位为 0x4
2. 等待 50 个系统时钟
3. 配置 CRM_MISC2 寄存器中的 HICK_TO_SCLK 位为 0x1
4. 更改 CRM_MISC1 寄存器中的 HICKDIV 值
5. 恢复 HICK_TO_SCLK 位为原配置值
6. 重新配置 HICK_TO_SCLK_DIV 位为需要设定的值

- PLL 时钟

PLL 可以选择 HICK 时钟或 HEXT 时钟作为输入时钟，PLL 的输入时钟在 PLL 内部经过预分频器分频后送给 VCO 倍频，VCO 输出频率经过后分频器分频后输出。其中预分频后时钟需保证在 2M~16MHz 之间，VCO 的工作频率需保证在 500MHz~1000MHz 之间。使用 PLL 前，一定要先配置 PLL 参数，否则，PLL 使能后，这些参数将无法改动。PLL 时钟直到稳定后才会被释放出来。
PLL 公式如下：
PLL 输出时钟 = PLL 输入时钟 x PLL 倍频系数 /（PLL 预分频系数 x PLL 后分频系数）
500MHz <= PLL 输入时钟 x PLL 倍频系数 / PLL 预分频系数 <= 1000MHz
2MHz <= PLL 输入时钟 / PLL 预分频系数 <= 16MHz
例如：当 PLL 输入时钟为 25 MHz 时，可以配置 PLL 输出频率= 25 x 192 / (5 x 4) = 240 MHz

- LEXT 振荡器时钟

LEXT 振荡器时钟包括 LEXT 晶体/陶瓷谐振器和 LEXT 旁路时钟两个时钟源。
LEXT 晶体/陶瓷谐振器
LEXT 晶体/陶瓷谐振器提供一个低功耗且精确的 32.768KHz 低速时钟源。LEXT 时钟直到稳定后，才会被释放出来。

- LEXT 旁路时钟

在 LEXT 旁路模式下，可以提供最高频率达 32.768kHz 的外部时钟源。外部时钟信号必须连到 LEXT_IN 引脚，并且 LEXT_OUT 引脚也一定要保持悬空。

- LICK 振荡器时钟

LICK 振荡器时钟由芯片内的低速 RC 振荡器提供，作为一个频率在 30kHz 和 60kHz 之间的低功耗时钟源，它可以为看门狗和自动唤醒单元提供时钟，并能在深度睡眠和待机模式下保持运行。
LICK 时钟直到稳定后，才会被释放出来。

4.1.2 系统时钟

系统复位以后，系统时钟使用 HICK 时钟作为默认时钟。系统时钟可在 HICK 振荡器时钟、HEXT 振荡器时钟和 PLL 时钟之间进行灵活切换，只有当目标时钟源稳定后，系统时钟切换才会发生。当 HICK 振荡器时钟直接作为系统时钟或间接通过 PLL 作为系统时钟时，它将无法被停止。

4.1.3 外设时钟

大多数外设使用系统时钟 HCLK、PCLK1 或 PCLK2 时钟。个别外设还有专用时钟。
系统嘀嗒定时器（SysTick）使用 CPU FCLK(HCLK)或 CPU systick(HCLK 的 8 分频)作为时钟。

ADC 使用 HCLK 或是 PLL 时钟的 2、3、4、5…17 分频作为时钟。
定时器使用 APB1/2 作为时钟，特别地，当 APB 预分频系数是 1 时，定时器的时钟频率等于 APB1/2 的时钟频率；当 APB 预分频系数不为 1 时，定时器的时钟频率等于 APB1/2 时钟频率的 2 倍。
USB 时钟可在 HICK 和 PLL 分频时钟之间切换。当选 HICK 时钟源时，需配置 USB 时钟为 48MHz 时钟；当选 PLL 分频时钟时，USB 分频器提供 48MHz 的 USBCLK 时钟，PLL 需设置为（48*N）MHz（N=2,3,4,5…）。
ERTC 的时钟源有：HEXT 振荡器分频时钟，LEXT 振荡器时钟和 LICK 振荡器时钟。ERTC 的时钟源一旦选择后就不可再更改，只有复位后才能重新配置 ERTC 时钟源。当 VDD 掉电时，由于 HEXT、LEXT 和 LICK 均掉电，会导致 ERTC 状态不定。
I^2C1 的时钟源有：系统时钟 SCLK，PCLK1 时钟和 HICK 振荡器时钟。
USART1 的时钟源有：系统时钟 SCLK，PCLK2 时钟，HICK 振荡器时钟和 LEXT 振荡器时钟。
USART2/3 的时钟源有：系统时钟 SCLK，PCLK1 时钟，HICK 振荡器时钟和 LEXT 振荡器时钟。
看门狗使用 LICK 振荡器时钟作为时钟源。硬件选项或软件开启看门狗后，将强制打开 LICK 振荡器，LICK 振荡器稳定后，才给看门狗提供时钟。

4.1.4 时钟失效检测

当 HEXT 时钟直接或间接作为系统时钟时，为防止 HEXT 时钟出现故障，特设计了时钟失效检测模块（CFD）。当 HEXT 时钟出现故障，CFD 侦测到失效后，将时钟失效事件送到 TMR1/9/10/11/12/13/14 的刹车输入端，并产生 CFD 中断，此 CFD 中断直接连到 CPU 的 NMI 中断，供软件完成营救操作。NMI 中断将一直重复执行，直到 CFD 中断挂起位被清除为止，所以在 NMI 的处理程序中必须清除 CFD 中断。
当 HEXT 时钟出现故障时，将导致系统时钟切换到 HICK 时钟，同时关闭 CFD，关闭 HEXT 时钟，如果 HEXT 时钟通过 PLL 做为系统时钟时，也会关闭 PLL 模块。

4.1.5 自动滑顺频率切换

当系统时钟源切换或是 AHB 分频因子更改时，为了使系统稳定顺滑切换，特设计了自动顺滑频率切换功能。当自动顺滑频率切换功能开启时，硬件会暂停 AHB 总线，直到整个自动顺滑频率切换才恢复。此期间 DMA 仍正常工作，中断事件会被记忆并待 AHB 总线恢复后 NVIC 即可处理。

4.1.6 内部时钟输出

微控制器允许输出时钟信号到外部 CLKOUT 引脚。ADCCLK、USB48M、SCLK、LICK、LEXT、HICK、HEXT、PLLCLK 等时钟信号可作为 CLKOUT 时钟。作为 CLKOUT 时钟输出脚时，相应的 GPIO 端口寄存器必须被配置为相应功能。

4.1.7 中断

微控制器为每个时钟源设计了一个稳定标志，当用户开启一个时钟源后，可查询对应的时钟源的的稳定标志来判断时钟是否稳定。当用户开启对应时钟源的中断使能的话，将产生中断请求。
当 HEXT 时钟出现故障，CFD 侦测到失效后，将产生 CFD 中断，此中断直接连到 CPU 的 NMI 中断。

4.2 复位

4.2.1 系统复位

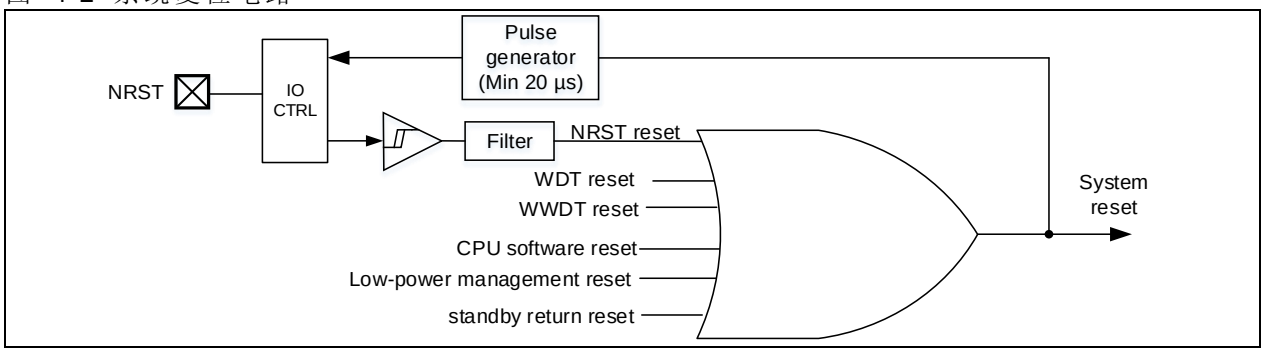
AT32F423 系统复位包括以下复位源：

- NRST 复位：外部 NRST 管脚复位
- WDT 复位：看门狗溢出复位
- WWDT 复位：窗口看门狗溢出复位
- CPU 软件复位：Cortex®-M4F 软件复位
- 低功耗管理复位：将用户系统数据中的 nSTDBY_RST 位清 0 并进入待机模式，将产生低功耗管理复位；将用户系统数据中的 nDEPSLP_RST 位清 0 并进入深度睡眠模式，也将产生低功耗管理复位。
- POR 复位：上电复位
- LVR 复位：掉电复位
- 从待机模式中返回等事件产生复位。

NRST 复位，WDT 复位，WWDT 复位，软件复位和低功耗管理复位将复位所有寄存器至它们的复位状

态，时钟控制器的控制/状态寄存器（CRM_CTRLSTS）和电池供电域中的寄存器除外；从待机模式中返回等事件产生复位会复位所有寄存器至复位状态，电池供电寄存器除外。

图 4-2 系统复位电路



4.2.2 电池供电域复位

电池供电域复位包括以下复位源：

- 电池供电域软件复位：设置电池供电域控制寄存器（CRM_BPDC）中的 BPDRST 位来产生复位
- 在 VDD 掉电的前提下，VDD 再上电将产生复位。

电池供电域软件复位只影响电池供电域。

4.3 CRM寄存器描述

下表列出了 CRM 寄存器的映像和复位值。

可以用字节（8 位）、半字（16 位）或字（32 位）的方式操作这些外设寄存器。

表 4-1 CRM寄存器的映像和复位值

寄存器简称	基址偏移量	复位值
CRM_CTRL	0x000	0x0000 XX83
CRM_PLLCFG	0x004	0x0003 3002
CRM_CFG	0x008	0x4000 0000
CRM_CLKINT	0x00C	0x0000 0000
CRM_AHBRST1	0x010	0x0000 0000
CRM_AHBRST2	0x014	0x0000 0000
CRM_AHBRST3	0x018	0x0000 0000
CRM_APB1RST	0x020	0x0000 0000
CRM_APB2RST	0x024	0x0000 0000
CRM_AHBEN1	0x030	0x0000 0000
CRM_AHBEN2	0x034	0x0000 0000
CRM_AHBEN3	0x038	0x0000 0000
CRM_APB1EN	0x040	0x0000 0000
CRM_APB2EN	0x044	0x0000 0000
CRM_AHBLPEN1	0x050	0x0141 903F
CRM_AHBLPEN2	0x054	0x0000 0080
CRM_AHBLPEN3	0x058	0x0000 0001
CRM_APB1LPEN	0x060	0xF6FE C9F7
CRM_APB2LPEN	0x064	0x2007 5131
CRM_PICLKS	0x068	0x0000 0000

	CRM_BPDC	0x070	0x0000 0000
	CRM_CTRLSTS	0x074	0x0C00 0000
	CRM_MISC1	0x0A0	0x000F 0000
	CRM_MISC2	0x0A4	0x0000 000D

4.3.1 时钟控制寄存器（CRM_CTRL）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：26	保留	0x00	resd	请保持默认值。
位 25	PLLSTBL	0x0	ro	PLL 时钟稳定（PLL clock stable） 该位待 PLL 稳定后由硬件置起。 0：未稳定； 1：已稳定。
位 24	PLLEN	0x0	rw	PLL 使能（PLL enable） 该位可由软件置起或清除，也可在进入待机或深度睡眠模式时，由硬件清除。当系统时钟为 PLL 时钟时，该位无法清除。 0：关闭； 1：开启。
位 23：20	保留	0x0	resd	保持默认值。
位 19	CFDEN	0x0	rw	时钟失效检测使能（Clock Failure Detection enable） 0：关闭； 1：开启。
位 18	HEXTBYPS	0x0	rw	HEXT 旁路使能（High speed external crystal bypass） 只有在 HEXT 关闭时，软件才能操作该位。 0：关闭； 1：开启。
位 17	HEXTSTBL	0x0	ro	HEXT 时钟稳定（High speed external crystal stable） 该位待 HEXT 稳定后由硬件置起。 0：未稳定； 1：已稳定。
位 16	HEXTEN	0x0	rw	HEXT 使能（High speed external crystal enable） 该位可由软件置起或清除，也可在进入待机或深度睡眠模式时，由硬件清除。当系统时钟有用到 HEXT 时，该位无法清除。 0：关闭； 1：开启。
位 15：8	HICKCAL	0xXX	rw	HICK 时钟校准值（High speed internal clock calibration） 默认值为出厂校准初始值。 HICK 输出频率为 48 MHz 时，每 HICKCAL 数值的变化对应频率调整 240 kHz（设计值）；HICK 输出频率是 8 MHz 时，每 HICKCAL 数值的变化对应频率调整 40 kHz（设计值）。 注意：此位只有在 HICKCAL_KEY[7：0]为 0x5A 的时候可被写入。
位 7：2	HICKTRIM	0x20	rw	HICK 时钟调整值（High speed internal clock trimming） 该数值和 HICKCAL[7：0]数值共同决定 HICK 振荡器的频率，默认数值为 32，可以把 HICK 调整到精度±1%。
位 1	HICKSTBL	0x1	ro	HICK 时钟稳定（High speed internal clock stable） 该位待 HICK 稳定后由硬件置起。 0：未稳定； 1：已稳定。

位 0	HICKEN	0x1	rw	HICK 使能（High speed internal clock enable） 该位可由软件置起或清除，在退出待机或深度睡眠模式，或 HEXT 发生故障时，该位也可被硬件置起。当系统时钟有用到 HICK 时，该位无法清除。 0：关闭； 1：开启。

4.3.2 PLL时钟配置寄存器（CRM_PLLCFG）

访问：无等待周期，字，半字和字节访问。

域	简称	复位值	类型	功能
位 31：23	保留	0x000	resd	请保持为复位值。
位 22	PLLRCS	0x0	rw	PLL 输入时钟选择（PLL reference clock select） 由软件置'1'或清'0'来选择 PLL 输入时钟源。只能在关闭 PLL 时才能写入此位。 0：HICK 时钟作为 PLL 输入时钟； 1：HEXT 时钟作为 PLL 输入时钟源。
位 21：19	保留	0x0	resd	请保持为复位值。
位 18：16	PLL_FR	0x3	rw	PLL 后分频配置值（PLL post-division） PLL_FR 范围（0~5） 000：PLL 后分频系数为 1，1 分频输出； 001：PLL 后分频系数为 2，2 分频输出； 010：PLL 后分频系数为 4，4 分频输出； 011：PLL 后分频系数为 8，8 分频输出； 100：PLL 后分频系数为 16，16 分频输出； 101：PLL 后分频系数为 32，32 分频输出； 其他：保留 请注意 PLL_FR 值和后分频系数对应关系
位 15	保留	0x0	resd	请保持为复位值。
位 14：6	PLL_NS	0x0C0	rw	PLL 倍频系数(PLL Multiplication Factor) PLL_NS 适用范围（31~500）适用 000000000 ~ 000011110： 禁止使用 000011111： 31 000100000： 32 000100001： 33 …… 111110011：499 111110100：500 111110101~111111111：禁止使用
位 5：4	保留	0x0	resd	请保持为复位值。
位 3：0	PLL_MS	0x2	rw	PLL 预分频系数(PLL pre-division) PLL_MS 适用范围（1~15） 0000：禁止使用 0001：1 0010：2 0011：3 …… 1110：14 1111：15

注意：PLL 时钟计算公式

PLL 输出时钟 = PLL 输入时钟 x PLL 倍频系数 / （PLL 预分频系数 x PLL 后分频

系数）
500MHz <= PLL 输入时钟 x PLL 倍频系数 / PLL 预分频系数 <= 1000MHz
2MHz <= PLL 输入时钟 / PLL 预分频系数 <= 16MHz

4.3.3 时钟配置寄存器（CRM_CFG）

访问：0 到 2 个等待周期，字，半字和字节访问，只有当访问发生在时钟切换时，才会插入 1 或 2 个等待周期。

域	简称	复位值	类型	功能
位 31：30	CLKOUT_SEL1	0x1	rw	内部输出时钟选择 1（clock output selecction 1） 由软件置'1'或清零。 00： 系统时钟（SCLK）输出； 01： 二级时钟输出，由 CRM_MISC1 寄存器的 CLKOUT_SEL2 位选择二级时钟输出 10： 外部振荡器时钟（HEXT）输出； 11： PLL 时钟输出； 注意：- 该时钟输出在启动和切换 CLKOUT 时钟源时可能会被截断。 - 在系统时钟作为输出至 CLKOUT 引脚时，请保证输出时钟频率不超过 50MHz（I/O 口最高频率）。
位 29：27	CLKOUTDIV1	0x0	rw	CLKOUT 分频因子 1（Clock output division1） 0xx： CLKOUT 100： CLKOUT/2 101： CLKOUT/3 110： CLKOUT/4 111： CLKOUT/5
位 26：21	保留	0x00	resd	请保持为复位值。
位 20：16	ERTCDIV	0x00	rw	ERTC 时钟源 HEXT 分频因子 （HEXT division for ERTC clock） 由软件置'1'或清'0'来控制 ERTC 时钟源 HEXT 的预分频系数。 此控制位必须在 ERTC 时钟源之前设定。 00000：禁止使用 00001：禁止使用 00010：HEXT/2 00011：HEXT/3 00100：HEXT/4 … 11110：HEXT/30 11111：HEXT/31
位 15：13	APB2DIV	0x0	rw	APB2 分频因子（APB2 division） HCLK 分频后作为 APB2 时钟。 0xx：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频
位 12：10	APB1DIV	0x0	rw	APB1 分频因子（APB1 division） HCLK 分频后作为 APB1 时钟。 0xx：不分频 100：2 分频 101：4 分频 110：8 分频 111：16 分频
位 9：8	保留	0x0	resd	请保持为复位值。

位 7：4	AHBDIV	0x0	rw	AHB 分频因子（AHB division） 0xxx：SCLK 不分频 1000：SCLK 2 分频　1100：SCLK 64 分频 1001：SCLK 4 分频　1101：SCLK 128 分频 1010：SCLK 8 分频　1110：SCLK 256 分频 1011：SCLK 16 分频 1111：SCLK 512 分频 注意：确保自动顺滑 AUTO_STEP_EN 开启后，再更改 AHBDIV
位 3：2	SCLKSTS	0x0	ro	系统时钟选择状态位（System clock select status） 00：HICK； 01：HEXT； 10：PLL/2； 11：保留，保持默认值。
位 1：0	SCLKSEL	0x0	rw	系统时钟选择（System clock select） 00：HICK； 01：HEXT； 10：PLL/2； 11：保留，保持默认值。 注意：确保自动顺滑 AUTO_STEP_EN 开启后，再更改 SCLKSEL

4.3.4 时钟中断寄存器 （CRM_CLKINT）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：24	保留	0x00	resd	请保持为复位值。
位 23	CFDFC	0x0	wo	清除时钟失效标志（Clock failure detection interrupt clear） 由软件置'1'来清除 CFDF。 0：无作用； 1：清除 CFDF 位。
位 22：21	保留	0x0	resd	请保持为复位值。
位 20	PLLSTBLFC	0x0	wo	清除 PLL 稳定标志（PLL stable flag clear） 由软件写'1'清除 PLLSTBLF。 0：不清除； 1：清除。
位 19	HEXTSTBLFC	0x0	wo	清除 HEXT 稳定标志（HEXT stable flag clear） 由软件写'1'清除 HEXTSTBLF。 0：不清除； 1：清除。
位 18	HICKSTBLFC	0x0	wo	清除 HICK 稳定标志（HICK stable flag clear） 由软件写'1'清除 HICKSTBLF。 0：不清除； 1：清除。
位 17	LEXTSTBLFC	0x0	wo	清除 LEXT 稳定标志（LEXT stable flag clear） 由软件写'1'清除 LEXTSTBLF。 0：不清除； 1：清除。
位 16	LICKSTBLFC	0x0	wo	清除 LICK 稳定标志（LICK stable flag clear） 由软件写'1'清除 LICKSTBLF。 0：不清除； 1：清除。
位 15：13	保留	0x0	resd	请保持为复位值。
位 12	PLLSTBLIEN	0x0	rw	PLL 稳定中断使能（PLL stable interrupt enable） 0：关闭； 1：开启。
位 11	HEXTSTBLIEN	0x0	rw	HEXT 稳定中断使能（HEXT stable interrupt enable） 0：关闭； 1：开启。
位 10	HICKSTBLIEN	0x0	rw	HICK 稳定中断使能（HICK stable interrupt enable）

				0：关闭； 1：开启。
位 9	LEXTSTBLIEN	0x0	rw	LEXT 稳定中断使能（LEXT stable interrupt enable） 0：关闭； 1：开启。
位 8	LICKSTBLIEN	0x0	rw	LICK 稳定中断使能（LICK stable interrupt enable） 0：关闭； 1：开启。
位 7	CFDF	0x0	ro	时钟失效标志（Clock Failure Detection flag） 在 HEXT 时钟出现故障时，由硬件置起。 0：未出现； 1：出现。
位 6：5	保留	0x0	resd	请保持为复位值。
位 4	PLLSTBLF	0x0	ro	PLL 稳定标志（PLL stable flag） 由硬件置起。 0：未稳定； 1：已稳定。
位 3	HEXTSTBLF	0x0	ro	HEXT 稳定标志（HEXT stable flag） 由硬件置起。 0：未稳定； 1：已稳定。
位 2	HICKSTBLF	0x0	ro	HICK 稳定标志（HICK stable flag） 由硬件置起。 0：未稳定； 1：已稳定。
位 1	LEXTSTBLF	0x0	ro	LEXT 稳定标志（LEXT stable flag） 由硬件置起。 0：未稳定； 1：已稳定。
位 0	LICKSTBLF	0x0	ro	LICK 稳定中断标志（LICK stable flag） 由硬件置起。 0：未稳定； 1：已稳定。

4.3.5 AHB外设复位寄存器1（CRM_AHBRST1）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：25	保留	0x00	resd	请保持为复位值。
位 24	DMA2RST	0x0	rw	DMA2 复位 （DMA2 reset） 0：无复位； 1：复位。
位 23	保留	0x0	resd	请保持为复位值。
位 22	DMA1RST	0x0	rw	DMA1 复位 （DMA1 reset） 0：无复位； 1：复位。
位 21：13	保留	0x000	resd	请保持为复位值。
位 12	CRCRST	0x0	rw	CRC 复位 （CRC reset） 0：无复位； 1：复位。
位 11：6	保留	0x00	resd	请保持为复位值。
位 5	GPIOFRST	0x0	rw	IO 端口 F 复位 （IO port F reset） 0：无复位； 1：复位。
位 4	GPIOERST	0x0	rw	IO 端口 E 复位 （IO port E reset） 0：无复位； 1：复位。
位 3	GPIODRST	0x0	rw	IO 端口 D 复位 （IO port D reset） 0：无复位； 1：复位。

位 2	GPIOCRST	0x0	rw	IO 端口 C 复位 （IO port C reset） 0：无复位； 1：复位。
位 1	GPIOBRST	0x0	rw	IO 端口 B 复位 （IO port B reset） 0：无复位； 1：复位。
位 0	GPIOARST	0x0	rw	IO 端口 A 复位 （IO port A reset） 0：无复位； 1：复位。

4.3.6 AHB外设复位寄存器2（CRM_AHBRST2）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	resd	请保持为复位值。
位 7	OTGFS1RST	0x0	rw	OTGFS1 复位 （OTGFS1 reset） 0：无复位； 1：复位。
位 6：0	保留	0x00	resd	请保持为复位值。

4.3.7 AHB外设复位寄存器3（CRM_AHBRST3）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：1	保留	0x00000000	resd	请保持为复位值。
位 0	XMCRST	0x0	rw	XMC 复位 （XMC reset） 0：无复位； 1：复位。

4.3.8 APB1外设复位寄存器（CRM_APB1RST）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31	USART8RST	0x0	rw	USART8 复位 （USART8 reset） 0：无复位； 1：复位。
位 30	USART7RST	0x0	rw	USART7 复位 （USART7 reset） 0：无复位； 1：复位。
位 29	DACRST	0x0	rw	DAC 接口复位 （DAC interface reset） 0：无复位； 1：复位。
位 28	PWCRST	0x0	rw	电源接口复位 （Power interface reset） 0：无复位； 1：复位。
位 27	保留	0x0	resd	请保持为复位值。
位 26	CAN2RST	0x0	rw	CAN2 复位 （CAN2 reset） 0：无复位； 1：复位。
位 25	CAN1RST	0x0	rw	CAN1 复位 （CAN1 reset） 0：无复位； 1：复位。
位 24	保留	0x0	resd	请保持为复位值。
位 23	I2C3RST	0x0	rw	I2C3 复位 （I2C3 reset） 0：无复位； 1：复位。
位 22	I2C2RST	0x0	rw	I2C2 复位 （I2C2 reset） 0：无复位；

				1：复位。
位 21	I2C1RST	0x0	rw	I2C1 复位 （I2C1 reset） 0：无复位； 1：复位。
位 20	USART5RST	0x0	rw	USART5 复位 （USART5 reset） 0：无复位； 1：复位。
位 19	USART4RST	0x0	rw	USART4 复位 （USART4 reset） 0：无复位； 1：复位。
位 18	USART3RST	0x0	rw	USART3 复位 （USART3 reset） 由软件置’1’或清’0’ 0：无作用； 1：复位 USART3。
位 17	USART2RST	0x0	rw	USART2 复位 （USART2 reset） 0：无复位； 1：复位。
位 16	保留	0x0	resd	请保持为复位值。
位 15	SPI3RST	0x0	rw	SPI3 复位 （SPI3 reset） 0：无复位； 1：复位。
位 14	SPI2RST	0x0	rw	SPI2 复位 （SPI2 reset） 0：无复位； 1：复位。
位 13：12	保留	0x0	resd	请保持为复位值。
位 11	WWDTRST	0x0	rw	窗口看门狗复位 （Window watchdog reset） 0：无复位； 1：复位。
位 10：9	保留	0x0	resd	请保持为复位值。
位 8	TMR14RST	0x0	rw	定时器 14 复位 （Timer14 reset） 0：无复位； 1：复位。
位 7	TMR13RST	0x0	rw	定时器 13 复位 （Timer13 reset） 0：无复位； 1：复位。
位 6	TMR12RST	0x0	rw	定时器 12 复位 （Timer12 reset） 0：无复位； 1：复位。
位 5	TMR7RST	0x0	rw	定时器 7 复位 （Timer7 reset） 0：无复位； 1：复位。
位 4	TMR6RST	0x0	rw	定时器 6 复位 （Timer6 reset） 0：无复位； 1：复位。
位 3	保留	0x0	resd	请保持为复位值。
位 2	TMR4RST	0x0	rw	定时器 4 复位 （Timer4 reset） 0：无复位； 1：复位。
位 1	TMR3RST	0x0	rw	定时器 3 复位 （Timer3 reset） 0：无复位； 1：复位。
位 0	TMR2RST	0x0	rw	定时器 2 复位 （Timer2 reset） 0：无复位； 1：复位。

4.3.9 APB2外设复位寄存器（CRM_APB2RST）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	请保持为复位值。

位 29	ACCRST	0x0	rw	ACC 复位 （ACC reset） 0：无复位； 1：复位。
位 28：19	保留	0x000	resd	请保持为复位值。
位 18	TMR11RST	0x0	rw	定时器 11 复位 （Timer11 reset） 0：无复位； 1：复位。
位 17	TMR10RST	0x0	rw	定时器 10 复位 （Timer10 reset） 0：无复位； 1：复位。
位 16	TMR9RST	0x0	rw	定时器 9 复位 （Timer9 reset） 0：无复位； 1：复位。
位 15	保留	0x0	resd	请保持为复位值。
位 14	SCFGRST	0x0	rw	SCFG 复位 （SCFG reset） 0：无复位； 1：复位。
位 13	保留	0x0	resd	请保持为复位值。
位 12	SPI1RST	0x0	rw	SPI1 复位 （SPI1 reset） 0：无复位； 1：复位。
位 11：9	保留	0x0	resd	请保持为复位值。
位 8	ADCRST	0x0	rw	ADC 接口复位 （ADC interface reset） 0：无复位； 1：复位。
位 7：6	保留	0x0	resd	请保持为复位值。
位 5	USART6RST	0x0	rw	USART6 复位 （USART6 reset） 0：无复位； 1：复位。
位 4	USART1RST	0x0	rw	USART1 复位 （USART1 reset） 0：无复位； 1：复位。
位 3：1	保留	0x0	resd	请保持为复位值。
位 0	TMR1RST	0x0	rw	TMR1 定时器复位 （TMR1 timer reset） 0：无复位； 1：复位。

4.3.10 AHB外设时钟使能寄存器1（CRM_AHBEN1）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：25	保留	0x00	resd	请保持为复位值。
位 24	DMA2EN	0x0	rw	DMA2 时钟使能（DMA2 clock enable） 0：关闭； 1：开启。
位 23	保留	0x0	resd	请保持为复位值。
位 22	DMA1EN	0x0	rw	DMA1 时钟使能（DMA1 clock enable） 0：关闭； 1：开启。
位 21：13	保留	0x000	resd	请保持为复位值。
位 12	CRCEN	0x0	rw	CRC 时钟使能（CRC clock enable） 0：关闭； 1：开启。
位 11：6	保留	0x00	resd	请保持为复位值。
位 5	GPIOFEN	0x0	rw	IO 端口 F 时钟使能（IO port F clock enable） 0：关闭； 1：开启。
位 4	GPIOEEN	0x0	rw	IO 端口 E 时钟使能（IO port E clock enable） 0：关闭；

				1：开启。
位 3	GPIODEN	0x0	rw	IO 端口 D 时钟使能（IO port D clock enable） 0：关闭； 1：开启。
位 2	GPIOCEN	0x0	rw	IO 端口 C 时钟使能（IO port C clock enable） 0：关闭； 1：开启。
位 1	GPIOBEN	0x0	rw	IO 端口 B 时钟使能（IO port B clock enable） 0：关闭； 1：开启。
位 0	GPIOAEN	0x0	rw	IO 端口 A 时钟使能（IO port A clock enable） 0：关闭； 1：开启。

4.3.11 AHB外设时钟使能寄存器2（CRM_AHBEN2）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	resd	请保持为复位值。
位 7	OTGFS1EN	0x0	rw	OTGFS1 时钟使能 （OTGFS1 clock enable） 0：关闭； 1：开启。
位 6：0	保留	0x00	resd	请保持为复位值。

4.3.12 AHB外设时钟使能寄存器3（CRM_AHBEN3）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：1	保留	0x00000000	resd	请保持为复位值。
位 0	XMCEN	0x0	rw	XMC 时钟使能 （XMC clock enable） 0：关闭； 1：开启。

4.3.13 APB1外设时钟使能寄存器（CRM_APB1EN）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31	USART8EN	0x0	rw	USART8 时钟使能 （USART8 clock enable） 0：关闭； 1：开启。
位 30	USART7EN	0x0	rw	USART7 时钟使能 （USART7 clock enable） 0：关闭； 1：开启。
位 29	DACEN	0x0	rw	DAC 接口时钟使能 （DAC interface clock enable） 0：关闭； 1：开启。
位 28	PWCEN	0x0	rw	电源接口时钟使能 （Power interface clock enable） 0：关闭； 1：开启。
位 27	保留	0x0	resd	请保持为复位值。
位 26	CAN2EN	0x0	rw	CAN2 时钟使能 （CAN2 clock enable） 0：关闭； 1：开启。
位 25	CAN1EN	0x0	rw	CAN1 时钟使能 （CAN1 clock enable） 0：关闭； 1：开启。
位 24	保留	0x0	resd	请保持为复位值。

位 23	I2C3EN	0x0	rw	I2C3 时钟使能 （I2C3 clock enable） 0：关闭； 1：开启。
位 22	I2C2EN	0x0	rw	I2C2 时钟使能 （I2C2 clock enable） 0：关闭； 1：开启。
位 21	I2C1EN	0x0	rw	I2C1 时钟使能 （I2C1 clock enable） 0：关闭； 1：开启。
位 20	USART5EN	0x0	rw	USART5 时钟使能 （USART5 clock enable） 0：关闭； 1：开启。
位 19	USART4EN	0x0	rw	USART4 时钟使能 （USART4 clock enable） 0：关闭； 1：开启。
位 18	USART3EN	0x0	rw	USART3 时钟使能 （USART3 clock enable） 0：关闭； 1：开启。
位 17	USART2EN	0x0	rw	USART2 时钟使能 （USART2 clock enable） 0：关闭； 1：开启。
位 16	保留	0x0	resd	请保持为复位值。
位 15	SPI3EN	0x0	rw	SPI3 时钟使能 （SPI3 clock enable） 0：关闭； 1：开启。
位 14	SPI2EN	0x0	rw	SPI2 时钟使能 （SPI 2 clock enable） 0：关闭； 1：开启。
位 13：12	保留	0x0	resd	请保持为复位值。
位 11	WWDTEN	0x0	rw	窗口看门狗时钟使能 （Window watchdog clock enable） 0：关闭； 1：开启。
位 10：9	保留	0x0	resd	请保持为复位值。
位 8	TMR14EN	0x0	rw	定时器 14 时钟使能 （Timer14 clock enable） 0：关闭； 1：开启。
位 7	TMR13EN	0x0	rw	定时器 13 时钟使能 （Timer13 clock enable） 0：关闭； 1：开启。
位 6	TMR12EN	0x0	rw	定时器 12 时钟使能 （Timer12 clock enable） 0：关闭； 1：开启。
位 5	TMR7EN	0x0	rw	定时器 7 时钟使能 （Timer 7 clock enable） 0：关闭； 1：开启。
位 4	TMR6EN	0x0	rw	定时器 6 时钟使能 （Timer 6 clock enable） 0：关闭； 1：开启。
位 3	保留	0x0	resd	请保持为复位值。
位 2	TMR4EN	0x0	rw	定时器 4 时钟使能 （Timer 4 clock enable） 0：关闭； 1：开启。
位 1	TMR3EN	0x0	rw	定时器 3 时钟使能 （Timer 3 clock enable） 0：关闭； 1：开启。
位 0	TMR2EN	0x0	rw	定时器 2 时钟使能 （Timer 2 clock enable） 0：关闭； 1：开启。

4.3.14 APB2外设时钟使能寄存器（CRM_APB2EN）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	请保持为复位值。
位 29	ACCEN	0x0	rw	ACC 时钟使能 （ACC clock enable） 0：关闭； 1：开启。
位 28：19	保留	0x00	resd	请保持为复位值。
位 18	TMR11EN	0x0	rw	定时器 11 时钟使能 （Timer11 clock enable） 0：关闭； 1：开启。
位 17	TMR10EN	0x0	rw	定时器 10 时钟使能 （Timer10 clock enable） 0：关闭； 1：开启。
位 16	TMR9EN	0x0	rw	定时器 9 时钟使能 （Timer9 clock enable） 0：关闭； 1：开启。
位 15	保留	0x0	resd	请保持为复位值。
位 14	SCFGEN	0x0	rw	SCFG 时钟使能 （SCFG clock enable） 0：关闭； 1：开启。
位 13	保留	0x0	resd	请保持为复位值。
位 12	SPI1EN	0x0	rw	SPI1 时钟使能 （SPI1 clock enable） 0：关闭； 1：开启。
位 11：9	保留	0x0	resd	请保持为复位值。
位 8	ADCEN	0x0	rw	ADC 接口时钟使能 （ADC interface clock enable） 0：关闭； 1：开启。
位 7：6	保留	0x0	resd	请保持为复位值。
位 5	USART6EN	0x0	rw	USART6 时钟使能 （USART6 clock enable） 0：关闭； 1：开启。
位 4	USART1EN	0x0	rw	USART1 时钟使能 （USART1 clock enable） 0：关闭； 1：开启。
位 3：1	保留	0x0	resd	请保持为复位值。
位 0	TMR1EN	0x0	rw	TMR1 定时器时钟使能 （TMR1 timer clock enable） 0：关闭； 1：开启。

4.3.15 AHB外设时钟低功耗使能寄存器1（CRM_AHBLPEN1）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：25	保留	0x00	resd	请保持为复位值。
位 24	DMA2LPEN	0x1	rw	睡眠模式下 DMA2 时钟使能（DMA2 clock enable during sleep mode） 0：关闭； 1：开启。
位 23	保留	0x0	resd	请保持为复位值。
位 22	DMA1LPEN	0x1	rw	睡眠模式下 DMA1 时钟使能（DMA1 clock enable during sleep mode） 0：关闭； 1：开启。
位 21：17	保留	0x00	resd	请保持为复位值。
位 16	SRAMLPEN	0x1	rw	睡眠模式下 SRAM 时钟使能（SRAM clock enable during sleep mode）

				0：关闭； 1：开启。
位 15	FLASHLPEN	0x1	rw	睡眠模式下 FLASH 时钟使能（FLASH clock enable during sleep mode） 0：关闭； 1：开启。
位 14：13	保留	0x0	resd	请保持为复位值。
位 12	CRCLPEN	0x1	rw	睡眠模式下 CRC 时钟使能（CRC clock enable during sleep mode） 0：关闭； 1：开启。
位 11：6	保留	0x0	resd	请保持为复位值。
位 5	GPIOFLPEN	0x1	rw	睡眠模式下 IO 端口 F 时钟使能（IO port F clock enable during sleep mode） 0：关闭； 1：开启。
位 4	GPIOELPEN	0x1	rw	睡眠模式下 IO 端口 E 时钟使能（IO port E clock enable during sleep mode） 0：关闭； 1：开启。
位 3	GPIODLPEN	0x1	rw	睡眠模式下 IO 端口 D 时钟使能（IO port D clock enable during sleep mode） 0：关闭； 1：开启。
位 2	GPIOCLPEN	0x1	rw	睡眠模式下 IO 端口 C 时钟使能（IO port C clock enable during sleep mode） 0：关闭； 1：开启。
位 1	GPIOBLPEN	0x1	rw	睡眠模式下 IO 端口 B 时钟使能（IO port B clock enable during sleep mode） 0：关闭； 1：开启。
位 0	GPIOALPEN	0x1	rw	睡眠模式下 IO 端口 A 时钟使能（IO port A clock enable during sleep mode） 0：关闭； 1：开启。

4.3.16 AHB外设时钟低功耗使能寄存器2（CRM_AHBLPEN2）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	resd	请保持为复位值。
位 7	OTGFS1LPEN	0x1	rw	睡眠模式下 OTGFS1 时钟使能 （OTGFS1 clock enable during sleep mode） 0：关闭； 1：开启。
位 6：0	保留	0x00	resd	请保持为复位值。

4.3.17 AHB外设时钟低功耗使能寄存器3（CRM_AHBLPEN3）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：1	保留	0x00000000	resd	请保持为复位值。
位 0	XMCLPEN	0x1	rw	睡眠模式下 XMC 时钟使能 （XMC clock enable during sleep mode） 0：关闭； 1：开启。

4.3.18 APB1外设时钟低功耗使能寄存器（CRM_APB1LPEN）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31	USART8LPEN	0x1	rw	睡眠模式下 USART8 时钟使能 （USART8 clock enable during sleep mode） 0：关闭； 1：开启。
位 30	USART7LPEN	0x1	rw	睡眠模式下 USART7 时钟使能 （USART7 clock enable during sleep mode） 0：关闭； 1：开启。
位 29	DACLPEN	0x1	rw	睡眠模式下 DAC 接口时钟使能 （DAC interface clock enable during sleep mode） 0：关闭； 1：开启。
位 28	PWCLPEN	0x1	rw	睡眠模式下 PWC 时钟使能 （Power interface clock enable during sleep mode） 0：关闭； 1：开启。
位 27	保留	0x0	resd	请保持为复位值。
位 26	CAN2LPEN	0x1	rw	睡眠模式下 CAN2 时钟使能 （CAN2 clock enable during sleep mode） 0：关闭； 1：开启。
位 25	CAN1LPEN	0x1	rw	睡眠模式下 CAN1 时钟使能 （CAN1 clock enable during sleep mode） 0：关闭； 1：开启。
位 24	保留	0x0	resd	请保持为复位值。
位 23	I2C3LPEN	0x1	rw	睡眠模式下 I2C3 时钟使能 （I2C3 clock enable during sleep mode） 0：关闭； 1：开启。
位 22	I2C2LPEN	0x1	rw	睡眠模式下 I2C2 时钟使能 （I2C2 clock enable during sleep mode） 0：关闭； 1：开启。
位 21	I2C1LPEN	0x1	rw	睡眠模式下 I2C1 时钟使能 （I2C1 clock enable during sleep mode） 0：关闭； 1：开启。
位 20	USART5LPEN	0x1	rw	睡眠模式下 USART5 时钟使能 （USART5 clock enable during sleep mode） 0：关闭； 1：开启。
位 19	USART4LPEN	0x1	rw	睡眠模式下 USART4 时钟使能 （USART4 clock enable during sleep mode） 0：关闭； 1：开启。
位 18	USART3LPEN	0x1	rw	睡眠模式下 USART3 时钟使能 （USART3 clock enable during sleep mode） 0：关闭； 1：开启。
位 17	USART2LPEN	0x1	rw	睡眠模式下 USART2 时钟使能 （USART2 clock enable during sleep mode） 0：关闭； 1：开启。
位 16	保留	0x0	resd	请保持为复位值。

位 15	SPI3LPEN	0x1	rw	睡眠模式下 SPI3 时钟使能 （SPI3 clock enable during sleep mode） 0：关闭； 1：开启。
位 14	SPI2LPEN	0x1	rw	睡眠模式下 SPI2 时钟使能 （SPI 2 clock enable during sleep mode） 0：关闭； 1：开启。
位 13：12	保留	0x0	resd	请保持为复位值。
位 11	WWDTLPEN	0x1	rw	睡眠模式下 WWDT 时钟使能 （Window watchdog clock enable during sleep mode） 0：关闭； 1：开启。
位 10：9	保留	0x0	resd	请保持为复位值。
位 8	TMR14LPEN	0x1	rw	睡眠模式下 TMR14 时钟使能 （Timer14 clock enable during sleep mode） 0：关闭； 1：开启。
位 7	TMR13LPEN	0x1	rw	睡眠模式下 TMR13 时钟使能 （Timer13 clock enable during sleep mode） 0：关闭； 1：开启。
位 6	TMR12LPEN	0x1	rw	睡眠模式下 TMR12 时钟使能 （Timer12 clock enable during sleep mode） 0：关闭； 1：开启。
位 5	TMR7LPEN	0x1	rw	睡眠模式下 TMR7 时钟使能 （Timer 7 clock enable during sleep mode） 0：关闭； 1：开启。
位 4	TMR6LPEN	0x1	rw	睡眠模式下 TMR6 时钟使能 （Timer 6 clock enable during sleep mode） 0：关闭； 1：开启。
位 3	保留	0x0	resd	请保持为复位值。
位 2	TMR4LPEN	0x1	rw	睡眠模式下 TMR4 时钟使能 （Timer 4 clock enable during sleep mode） 0：关闭； 1：开启。
位 1	TMR3LPEN	0x1	rw	睡眠模式下 TMR3 时钟使能 （Timer 3 clock enable during sleep mode） 0：关闭； 1：开启。
位 0	TMR2LPEN	0x1	rw	睡眠模式下 TMR2 时钟使能 （Timer 2 clock enable during sleep mode） 0：关闭； 1：开启。

4.3.19 APB2外设时钟低功耗使能寄存器（CRM_APB2LPEN）

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	请保持为复位值。
位 29	ACCLPEN	0x1	rw	睡眠模式下 AC 时钟使能 （ACC clock enable during sleep mode） 0：关闭； 1：开启。
位 28：19	保留	0x000	resd	请保持为复位值。
位 18	TMR11LPEN	0x1	rw	睡眠模式下定时器 11 时钟使能 （Timer11 clock enable during sleep mode）

				0：关闭； 1：开启。
位 17	TMR10LPEN	0x1	rw	睡眠模式下定时器 10 时钟使能 （Timer10 clock enable during sleep mode） 0：关闭； 1：开启。
位 16	TMR9LPEN	0x1	rw	睡眠模式下定时器 9 时钟使能 （Timer9 clock enable during sleep mode） 0：关闭； 1：开启。
位 15	保留	0x0	resd	请保持为复位值。
位 14	SCFGLPEN	0x1	rw	睡眠模式下 SCFG 时钟使能 （SCFG clock enable during sleep mode） 0：关闭； 1：开启。
位 13	保留	0x0	resd	请保持为复位值。
位 12	SPI1LPEN	0x1	rw	睡眠模式下 SPI1 时钟使能 （SPI1 clock enable during sleep mode） 0：关闭； 1：开启。
位 11：9	保留	0x0	resd	请保持为复位值。
位 8	ADCLPEN	0x1	rw	睡眠模式下 ADC 接口时钟使能 （ADC interface clock enable during sleep mode） 0：关闭； 1：开启。
位 7：6	保留	0x0	resd	请保持为复位值。
位 5	USART6LPEN	0x1	rw	睡眠模式下 USART6 时钟使能 （USART6 clock enable during sleep mode） 0：关闭； 1：开启。
位 4	USART1LPEN	0x1	rw	睡眠模式下 USART1 时钟使能 （USART1 clock enable during sleep mode） 0：关闭； 1：开启。
位 3：1	保留	0x0	resd	请保持为复位值。
位 0	TMR1LPEN	0x1	rw	睡眠模式下 TMR1 定时器时钟使能 （TMR1 timer clock enable during sleep mode） 0：关闭； 1：开启。

4.3.20 外设独立时钟选择寄存器（CRM_PICLKS）

域	简称	复位值	类型	功能
位 31：14	保留	0x0000	resd	请保持为复位值。
位 13：12	I2C1SEL	0x0	rw	I2C1 时钟源选择（I2C1 clock source select） 00：PCLK1 01：SCLK 10：HICK 11：保留
位 11：6	保留	0x00	resd	请保持为复位值。
位 5：4	USART3SEL	0x0	rw	USART3 时钟源选择（USART3 clock source select） 00：PCLK1 01：SCLK 10：HICK 11：LEXT
位 3：2	USART2SEL	0x0	rw	USART2 时钟源选择（USART2 clock source select） 00：PCLK1 01：SCLK 10：HICK 11：LEXT

位 1：0	USART1SEL	0x0	rw	USART1 时钟源选择（USART1 clock source select） 00：PCLK2 01：SCLK 10：HICK 11：LEXT

4.3.21 电池供电域控制寄存器（CRM_BPDC）

只能由电池供电域复位有效复位

访问：0 到 3 等待周期，字、半字和字节访问；当连续对该寄存器进行访问时，将插入等待状态。

注意：电池供电域控制寄存器中（CRM_BPDC）LEXTEN、LEXTBYPS、ERTCSEL 和 ERTCEN 位处于电池供电域。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（PWC_CTRL）中的 BPWEN 位置位后才能对这些位进行改动。这些位只能由电池供电域软件复位清除。任何内部或外部复位都不会影响这些位。

域	简称	复位值	类型	功能
位 31：17	保留	0x0000	resd	请保持为复位值。
位 16	BPDRST	0x0	rw	电池供电域软件复位 （Battery powered domain software reset） 0：无复位； 1：复位。
位 15	ERTCEN	0x0	rw	ERTC 时钟使能 （ERTC clock enable） 由软件置'1'或清'0' 0：关闭 ERTC 时钟； 1：开启 ERTC 时钟。
位 14：10	保留	0x00	resd	请保持为复位值。
位 9：8	ERTCSEL	0x0	rw	ERTC 时钟源选择 （ERTC clock source selection） 确定了 ERTC 时钟选择后，如果想要再次更改，必须设置 BPDRST 位复位后，才能重新改写 ERTC 时钟选择。 00：无； 01：LEXT； 10：LICK； 11：HEXT 分频时钟（分频由 CRM_CFG 的 ERTC_DIV 设定）。
位 7：3	保留	0x00	resd	请保持为复位值。
位 2	LEXTBYPS	0x0	rw	LEXT 旁路使能 （Low speed external crystal bypass） 0：关闭； 1：开启。
位 1	LEXTSTBL	0x0	ro	LEXT 稳定（External low-speed oscillator stable） 该位待 LEXT 稳定后由硬件置起。 0：未稳定； 1：已稳定。
位 0	LEXTEN	0x0	rw	LEXT 使能 （External low-speed oscillator enable） 0：关闭； 1：开启。

4.3.22 控制/状态寄存器（CRM_CTRLSTS）

除复位标志外由系统复位清除，复位标志能由电源复位或写 RSTFC 位进行清除。访问：0 到 3 等待周期，字、半字和字节访问；当连续对该寄存器进行访问时，将插入等待状态。

域	简称	复位值	类型	功能
位 31	LPRSTF	0x0	ro	低功耗复位标志 （Low-power reset flag） 该位由硬件置起，软件写 RSTFC 位清除。 0：无； 1：有。
位 30	WWDTRSTF	0x0	ro	窗口看门狗复位标志 （WWDT reset flag） 该位由硬件置起，软件写 RSTFC 位清除。 0：无；

				1：有。
位 29	WDTRSTF	0x0	ro	看门狗复位标志 （WDT reset flag） 该位由硬件置起，软件写 RSTFC 位清除。 0：无； 1：有。
位 28	SWRSTF	0x0	ro	软件复位标志 （Software reset flag） 该位由硬件置起，软件写 RSTFC 位清除。 0：无； 1：有。
位 27	PORRSTF	0x1	ro	上电/低电压复位标志 （POR/LVR reset flag） 该位由硬件置起，软件写 RSTFC 位清除。 0：无； 1：有。
位 26	NRSTF	0x1	ro	NRST 引脚复位标志 （NRST reset flag） 该位由硬件置起，软件写 RSTFC 位清除。 0：无； 1：有。
位 25	保留	0x0	resd	请保持为复位值。
位 24	RSTFC	0x0	rw	清除复位标志 （Reset flag clear） 由软件置'1'来清除复位标志。 0：无作用； 1：清除复位标志。
位 23：2	保留	0x000000	resd	请保持为复位值。
位 1	LICKSTBL	0x0	ro	LICK 稳定 （LICK stable） 0：未稳定； 1：已稳定。
位 0	LICKEN	0x0	rw	LICK 使能 （LICK enable） 0：关闭； 1：开启。

4.3.23 额外寄存器（CRM_MISC1）

访问：无等待周期，字，半字和字节访问。

域	简称	复位值	类型	功能
位 31：28	CLKOUTDIV2	0x0	rw	CLKOUT 分频因子 2（Clock output division2） 0xxx： 不分频 1000：2 分频； 1001：4 分频； 1010：8 分频； 1011：16 分频； 1100：64 分频； 1101：128 分频； 1110：256 分频； 1111：512 分频。
位 27：20	保留	0x00	resd	请保持为复位值。
位 19：16	CLKOUT_SEL2	0xF	rw	CLKOUT 时钟选择位 2 (Clock output sel2) 0000：USB 时钟输出 0001：ADC 时钟输出 0010：内部 RC 振荡器时钟（HICK）除频输出 0011：LICK 时钟输出 0100：LEXT 时钟输出 0101~0111： 保留 1000~1111： 保留
位 15	PLLCLK_TO_ADC	0x0	rw	ADC 时钟源选择位（ADC clock source select） 0：ADC 时钟源是 HCLK 1：ADC 时钟源是 PLLCLK 注意：当选择 PLL 作为 ADC 时钟源时，此时必须保证系统时钟 SCLKSEL 选择 PLL/2
位 14	HICK_TO_SCLK	0x0	rw	HICK 作为系统时钟的频率选择位（HICK as system clock frequency select）

				当 SCLKSEL 选择 HICK 为时钟源时，SCLK 的频率为 0：固定是 8MHz，即选择 HICK 时钟的 6 分频； 1：根据 HICKDIV 设定可能为 48M 或 8M。
位 13	HICK_TO_USB	0x0	rw	USB 48M 时钟源选择位 （USB 48MHz clock source select） 0：USB 48M 时钟源是 PLL 或是其分频； 1：USB 48M 时钟源来自 HICK 或是其 6 分频。 注意：由于 USB 必须工作在 48M，此时必须保证 HICKDIV=1，使 USB 48M 时钟选择 HICK 的 48MHz 输出。
位 12	HICKDIV	0x0	rw	HICK 6 分频（HICK 6 divider selection） 该位选择使用 HICK 时钟还是 HICK 的 6 分频时钟，选择 HICK 的 6 分频时钟的话，时钟频率为 8 MHz，选择不分频的话，时钟频率为 48 MHz。 0：分频； 1：不分频。 注意： 1、当 HICK 作为 PLL 时钟源时，PLL 使能期间，请勿修改 HICKDIV；
位 11：8	保留	0x0	resd	请保持为复位值。
位 7：0	HICKCAL_KEY	0x00	rw	HICKCAL 写入键值（HICK calibration key） 此字段为 0x5A 时，HICKCAL [7：0]才可被写入。

4.3.24 额外寄存器2（CRM_MISC2）

访问：无等待周期，字，半字和字节访问。

域	简称	复位值	类型	功能
位 31：22	保留	0x000	resd	请保持为复位值。
位 21：19	HEXT_TO_SCLK_DIV	0x0	rw	HEXT 作为系统时钟的分频因子（HEXT as system clock frequency division） 000：HEXT 001：HEXT/2 010：HEXT/4 011：HEXT/8 100：HEXT/16 101：HEXT/32 其他：保留
位 18：16	HICK_TO_SCLK_DIV	0x0	rw	HICK 作为系统时钟的分频因子（HICK as system clock frequency division） 000：HICK 001：HICK/2 010：HICK/4 011：HICK/8 100：HICK/16 其他：保留
位 15：12	USBDIV	0x0	rw	USB 分频因子 （USB division） PLL 时钟分频后作为 USB 时钟。 0000：3 倍分频 0001：2 倍分频 0010：5 倍分频 0011：4 倍分频 0100：7 倍分频 0101：6 倍分频 0110：9 倍分频 0111：8 倍分频 其他：保留
位 11：6	保留	0x0	resd	请保持为复位值。
位 5：4	AUTO_STEP_EN	0x0	rw	自动滑顺频率切换（auto step system clock switch enable）

				为使切换系统时钟源或是切换 AHB 分频因子时平顺，建议启动自动滑顺频率切换。 当自动滑顺频率切换功能作用时，硬件会暂停 AHB 总线，直到整个自动滑顺频率切换完成才恢复。此期间 DMA 仍正常工作，中断事件会被记忆并待 AHB 总线恢复后 NVIC 即可处理。 00：关闭； 01：保留； 10：保留； 11：开启，当 AHBDIV 或 SCLKSEL 这两个控制位被改动时，会自动触发自动滑顺频率切换功能。
位 3：0	保留	0xD	resd	固定为 0xD，请勿修改。

5 闪存控制器（FLASH）

5.1 FLASH介绍

闪存由主存储器、信息块、闪存寄存器这三个部分组成。

- 主存储器容量可达256K字节
- 信息块由20K字节的系统启动程序代码区和用户系统数据区组成。系统启动程序使用USART1、USART2或USB实现ISP编程

主存储器只有闪存容量为256K字节的片1闪存，包含128扇区，每扇区大小为2K字节。

表 5-1 闪存存储结构（256K）

结构		名称	地址范围
主存储器	片 1 (Bank1) 256KB	扇区 0	0x0800 0000 – 0x0800 07FF
		扇区 1	0x0800 0800 – 0x0800 0FFF
		扇区 2	0x0800 1000 – 0x0800 17FF
		…	…
		扇区 127	0x0803 F800 – 0x0803 FFFF
信息块		启动程序代码区 20KB	0x1FFF A400 – 0x1FFF F3FF
		用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

主存储器只有闪存容量为128K字节的片1闪存，包含128扇区，每扇区大小为1K字节。

表 5-2 闪存存储组织（128K）

结构		名称	地址范围
主存储器	片 1 (Bank1) 128KB	扇区 0	0x0800 0000 – 0x0800 03FF
		扇区 1	0x0800 0400 – 0x0800 07FF
		扇区 2	0x0800 0800 – 0x0800 0BFF
		…	…
		扇区 127	0x0801 FC00 – 0x0801 FFFF
信息块		启动程序代码区 20KB	0x1FFF A400 – 0x1FFF F3FF
		用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

主存储器只有闪存容量为64K字节的片1闪存，包含64扇区，每扇区大小为1K字节。

表 5-3 闪存存储组织（64K）

结构		名称	地址范围
主存储器	片 1 (Bank1) 64KB	扇区 0	0x0800 0000 – 0x0800 03FF
		扇区 1	0x0800 0400 – 0x0800 07FF
		扇区 2	0x0800 0800 – 0x0800 0BFF
		…	…
		扇区 63	0x0800 FC00 – 0x0800 FFFF
信息块		启动程序代码区 20KB	0x1FFF A400 – 0x1FFF F3FF
		用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

用户系统数据区

每次系统复位后将从闪存信息块中读出系统数据信息并保存在FLASH_USD以及FLASH_EPPS寄存器中。

每个系统数据实际占用2个字节，低字节对应系统数据的内容，高字节对应系统数据的反码，用于验证选择位的正确性。当读出的高字节不等于低字节的反码时（高字节及低字节均为0xFF时除外），系统数据装载器会产生一个系统数据错误的标志（USDERR），并把对应的系统数据及其反码都强置为0xFF。

注意：用户系统数据内容的更新需要一次系统复位才能真正实现。

表 5-4 用户系统数据说明

地址	位	内容

0x1FFF_F800	[7:0]	**FAP[7:0]**：闪存访问保护（访问保护启动/解除结果存放在寄存器 FLASH_USD[1]以及[26]） 0xA5: 闪存访问保护解除 0xCC：高级闪存访问保护启动 其他值：低级闪存访问保护启动	
	[15:8]	**nFAP[7:0]**：FAP[7:0]的反码	
	[23:16]	**SSB[7:0]**：系统配置字节（存放在寄存器 FLASH_USD[9:2]）	
		位 7	保留不用
		位 6（nSTDBY_WDT）	0：进入待机模式时停止计数 1：进入待机模式时不停止计数
		位 5（nDEPSLP_WDT）	0：进入深度睡眠模式时停止计数 1：进入深度睡眠模式时不停止计数
		位 4（nBOOT1）	nBOOT1：和 BOOT0 引脚一起决定启动模式，当 BOOT0 = 1 时： 0：由 SRAM 启动 1：由系统启动程序代码区启动
		位 3	保留不用
		位 2（nSTDBY_RST）	0：进入待机模式时产生复位 1：进入待机模式时不产生复位
		位 1（nDEPSLP_RST）	0：进入深度睡眠模式时产生复位 1：进入深度睡眠模式时不产生复位
		位 0（nWDT_ATO_EN）	0：看门狗自启动开启 1：看门狗自启动关闭
	[31:24]	**nSSB[7:0]**：SSB[7:0]的反码	
0x1FFF_F804	[7:0]	**Data0[7:0]**：用户数据 0（存放在寄存器 FLASH_USD[17:10]）	
	[15:8]	**nData0[7:0]**：Data0[7:0]的反码	
	[23:16]	**Data1[7:0]**：用户数据 1（存放在寄存器 FLASH_USD[25:18]）	
	[31:24]	**nData1[7:0]**：Data1[7:0]的反码	
0x1FFF_F808	[7:0]	**EPP0[7:0]**：闪存擦写保护字节 0（存放在寄存器 FLASH_EPPS[7:0]） 用于保护 256KB 主闪存存储器的扇区 0 ~ 扇区 15，128KB 与 64KB 主闪存存储器的扇区 0 ~ 扇区 31，每个比特位保护 4K 字节 0：擦写保护启动 1：擦写保护解除	
	[15:8]	**nEPP0[7:0]**：EPP0[7:0]的反码	
	[23:16]	**EPP1[7:0]**：闪存擦写保护字节 1（存放在寄存器 FLASH_EPPS[15:8]） 用于保护 256KB 主闪存存储器的扇区 16 ~ 扇区 31，128KB 与 64KB 主闪存存储器的扇区 32 ~ 扇区 63，每个比特位保护 4K 字节 0：擦写保护启动 1：擦写保护解除	
	[31:24]	**nEPP1[7:0]**：EPP1[7:0]的反码	
0x1FFF_F80C	[7:0]	**EPP2[7:0]**：闪存擦写保护字节 2（存放在寄存器 FLASH_EPPS[23:16]） 用于保护 256KB 主闪存存储器的扇区 32~ 扇区 47，128KB 主闪存存储器的扇区 64~ 扇区 95，64KB 主闪存存储器此字节保留不用，每个比特位保护 4K 字节 0：擦写保护启动 1：擦写保护解除	
	[15:8]	**nEPP2[7:0]**：EPP2[7:0]的反码	
	[23:16]	**EPP3[7:0]**：闪存擦写保护字节 3（存放在寄存器 FLASH_EPPS[31:24]） 位 6 到位 0 用于保护 256KB 主闪存存储器的扇区 48~ 扇区 61，128KB 主闪存存储器的扇区 96~ 扇区 123，64KB 主闪存存储器位 6 到位 0 保留不用，每个比特位保护 4K 字节 位 7 用于保护 256KB 主闪存存储器的扇区 62~ 扇区 127，128KB 主闪存存储器的扇区 124~ 扇区 127，位 7 同时用于保护 256KB、128KB 与 64KB 主闪存存储器的主存扩展区 0：擦写保护启动 1：擦写保护解除	
	[31:24]	**nEPP3[7:0]**：EPP3[7:0]的反码	
0x1FFF_F810	[7:0]	**Data2[7:0]**：用户数据 2	
	[15:8]	**nData2[7:0]**：Data2[7:0]的反码	

	[23:16]	**Data3[7:0]**：用户数据 3
	[31:24]	**nData3[7:0]**：Data3[7:0]的反码
0x1FFF_F814	[7:0]	**Data4[7:0]**：用户数据 4
	[15:8]	**nData4[7:0]**：Data4[7:0]的反码
	[23:16]	**Data5[7:0]**：用户数据 5
	[31:24]	**nData5[7:0]**：Data5[7:0]的反码
. .	. .	. .
0x1FFF_F9FC	[7:0]	**Data248[7:0]**：用户数据 248
	[15:8]	**nData248[7:0]**：Data248[7:0]的反码
	[23:16]	**Data249[7:0]**：用户数据 249
	[31:24]	**nData249[7:0]**：Data249[7:0]的反码

5.2 主存储器操作

5.2.1 解锁/锁定

复位后，主存储器默认是被锁定的，此时不允许配置 FLASH_CTRL 寄存器，需要对闪存解锁后才能成功实现对闪存的写入与擦除操作。

解锁流程：

对 FLASH_UNLOCK 寄存器顺序写入键值 KEY1（0x45670123）和键值 KEY2（0xCDEF89AB），能够解锁对应区域闪存。

注意：解锁必须顺序写入正确的键值，否则会产生总线错误并且闪存会被锁死，直到下一次复位才能恢复。

锁定流程：

软件置起闪存控制寄存器（FLASH_CTRL）中的 OPLK 位，锁定对应区域闪存。

5.2.2 擦除

编程之前必须先进行擦除操作，主存储器有扇区擦除和整片擦除两种擦除方式。

扇区擦除

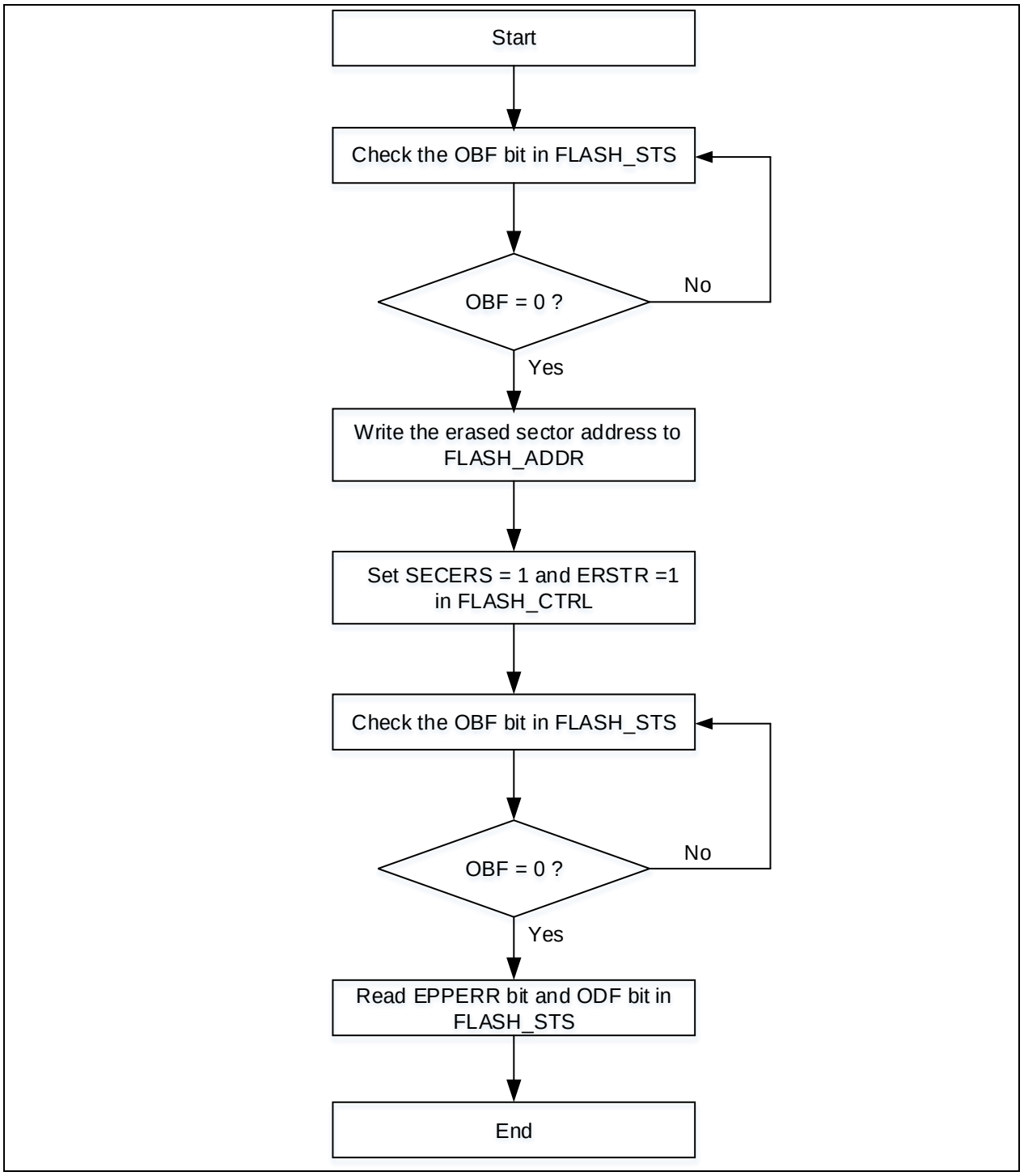
主闪存存储器的每一扇区以及主存扩展区都可以使用扇区擦除功能独立擦除。

擦除流程如下：

- 检查闪存状态寄存器（FLASH_STS）的OBF位，确认没有正在进行的闪存操作；
- 对FLASH_ADDR寄存器写入要擦除的扇区地址；
- 对闪存控制寄存器（FLASH_CTRL）的SECERS位以及ERSTR位均置1，启动扇区擦除；
- 等待闪存状态寄存器（FLASH_STS）的OBF位变为‘0’，并查询闪存状态寄存器（FLASH_STS）的EPPERR位和ODF位，确认擦除结果。

注意：当启动程序代码区域是设定为主存扩展区时，执行扇区擦除实际上是对整个主存扩展区的擦除。

图 5-1 主存储器扇区擦除流程图



整片擦除

主闪存存储器可以使用整片擦除功能直接擦除。

擦除流程如下：

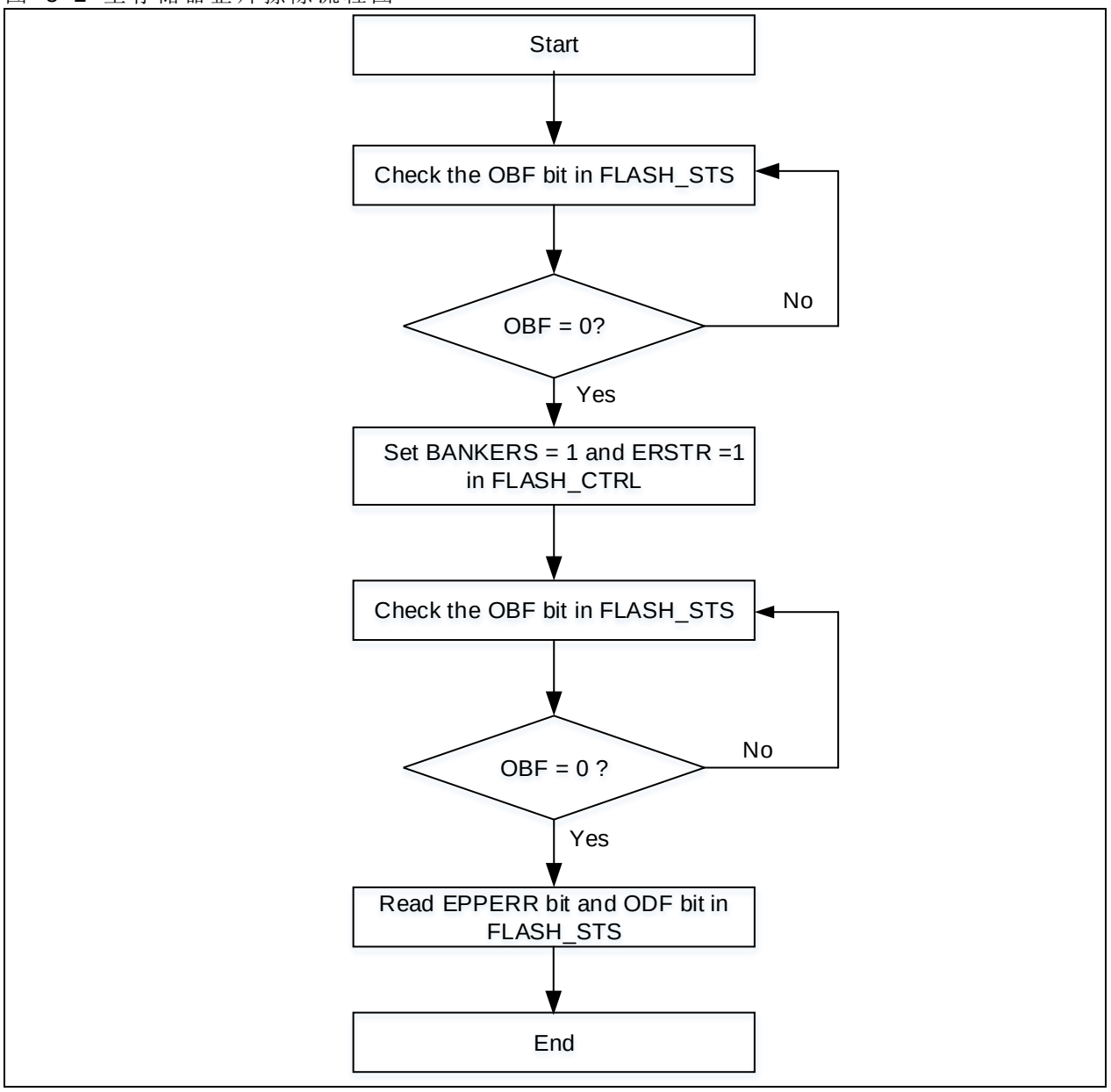
- 检查闪存状态寄存器（FLASH_STS）的OBF位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器（FLASH_CTRL）的BANKERS位以及ERSTR位均置1，启动整片擦除；
- 等待闪存状态寄存器（FLASH_STS）的OBF位变为‘0’，并查询闪存状态寄存器（FLASH_STS）的EPPERR位和ODF位，确认擦除结果。

注意：

1）当启动程序代码区域是设定为主存扩展区时，执行整片擦除操作会自动擦除主闪存以及主存扩展区。

2）擦除期间进行读闪存的操作，将导致 CPU 会被暂停直到擦除完成才处理读闪存操作。
3）擦除操作前必须保证内部的 HICK 有打开

图 5-2 主存储器整片擦除流程图



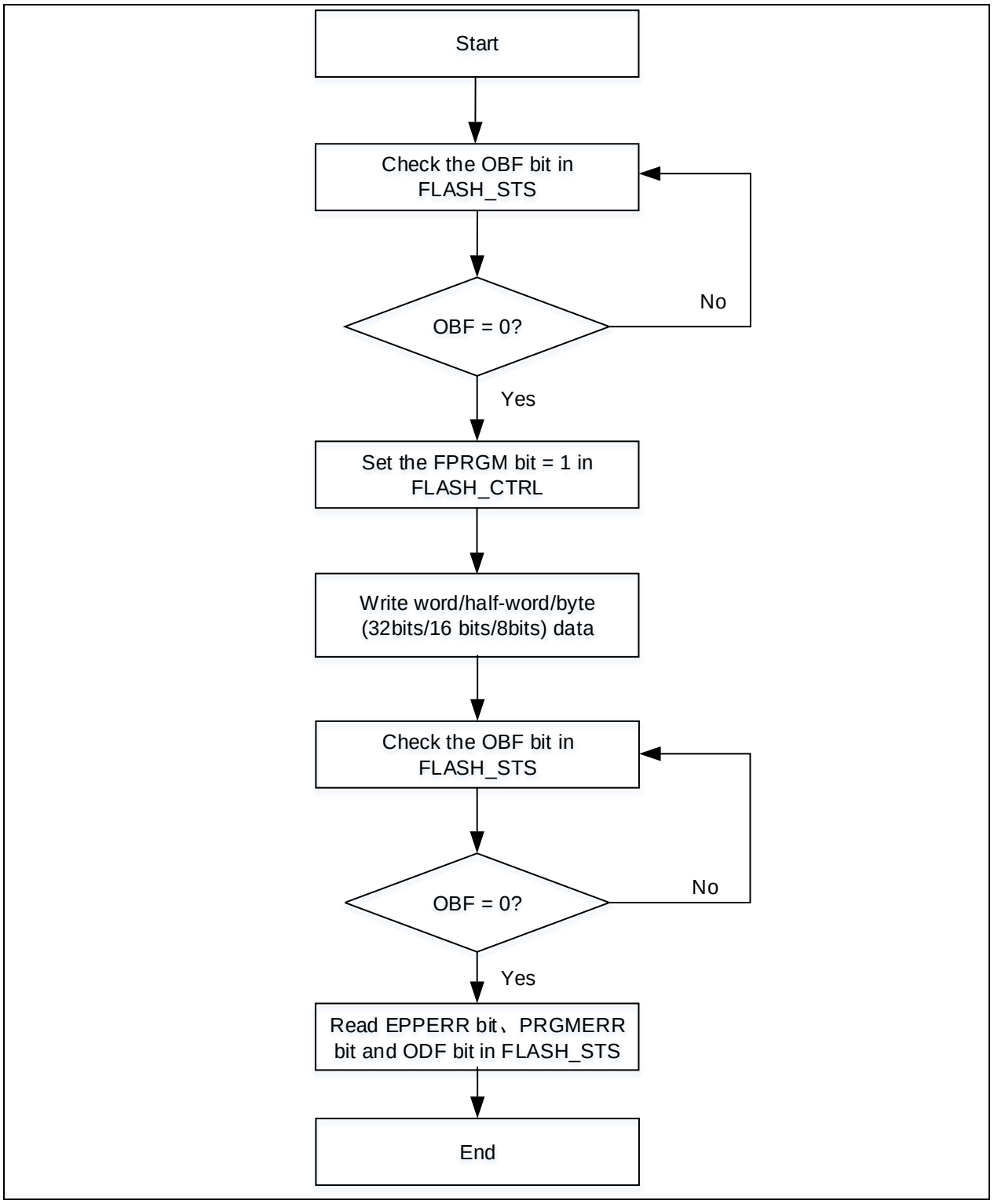
5.2.3 编程

当想要改写主存储器的内容时，可以通过主存储器编程流程完成一次写入 32 位、16 位或 8 位的数据。
主存储器编程流程：

- 检查闪存状态寄存器（FLASH_STS）的OBF位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器（FLASH_CTRL）的FPRGM位置1，此时可以接受对主闪存的编程指令；
- 对指定的地址写入要编程的数据（任意字/半字/字节）；
- 等待闪存状态寄存器（FLASH_STS）的OBF位变为‘0’，并查询闪存状态寄存器（FLASH_STS）的EPPERR位、PRGMERR位和ODF位，确认编程结果。

注意：
1）当要写入的地址未被提前擦除时，除非要写入的数据值是全 0，否则编程不被执行，并置位闪存状态寄存器（FLASH_STS）的 PRGMERR 位来告知编程发生错误。
2）编程期间，CPU 会被暂停直到编程完成才处理后续操作。
3）编程操作前必须保证内部的 HICK 有打开

图 5-3 主存储器编程流程图



5.2.4 读取

通过 CPU 的 AHB 总线可以直接寻址访问主闪存存储区。

5.3 主存扩展区操作

启动程序代码区也可以设定为主存扩展区存放用户应用代码。当作为主存扩展区时，其操作方法，包括读取、解锁、擦除、编程都跟主存储器相同。

5.4 用户系统数据区操作

5.4.1 解锁/锁定

复位后，用户系统数据区默认是锁定的，需要在闪存解锁后再对用户系统数据区解锁才能成功实现写入与擦除操作。

解锁流程：

对 FLASH_UNLOCK 寄存器顺序写入键值 KEY1（0x45670123）和键值 KEY2（0xCDEF89AB）；

对 FLASH_USD_UNLOCK 寄存器顺序写入键值 KEY1（0x45670123）和键值 KEY2（0xCDEF89AB），闪存控制寄存器（FLASH_CTRL）中的 USDULKS 位将被硬件自动置起，表示允许对用户系统数据区的写、擦除操作。

注意：解锁必须顺序写入正确的键值，否则会产生总线错误并且闪存会被锁死，直到下一次复位才能恢复。

锁定流程：

软件清除闪存控制寄存器（FLASH_CTRL）中的 USDULKS 位，锁定用户系统数据区。

5.4.2 擦除

在编程之前必须先进行擦除操作，用户系统数据区域可单独实现擦除功能。

擦除流程如下：

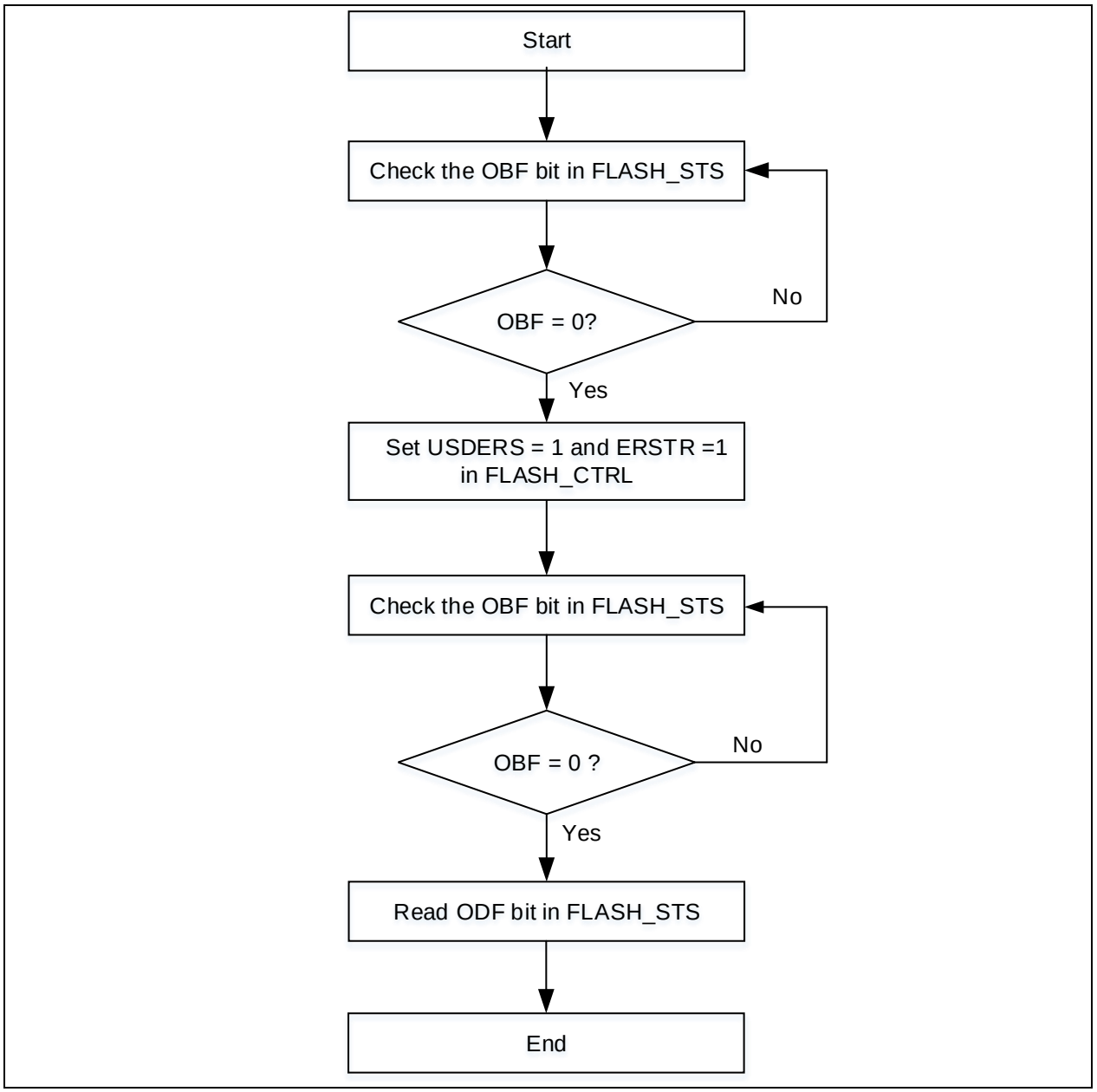
- 检查闪存状态寄存器（FLASH_STS）的OBF位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器（FLASH_CTRL）的USDERS位以及ERSTR位均置1，启动整块系统数据区擦除；
- 等待闪存状态寄存器（FLASH_STS）的OBF位变为‘0’，并查询闪存状态寄存器（FLASH_STS）的ODF位，确认擦除结果。

注意：

擦除期间进行读闪存的操作，将导致 CPU 会被暂停直到擦除完成才处理读闪存操作。

擦除操作前必须保证内部的 HICK 有打开。

图 5-4 系统数据区擦除图



5.4.3 编程

当想要改写用户系统数据区域的内容时，可以通过用户系统数据区编程流程完成一次写入 32 位或 16 位数据。

系统数据区的编程流程：

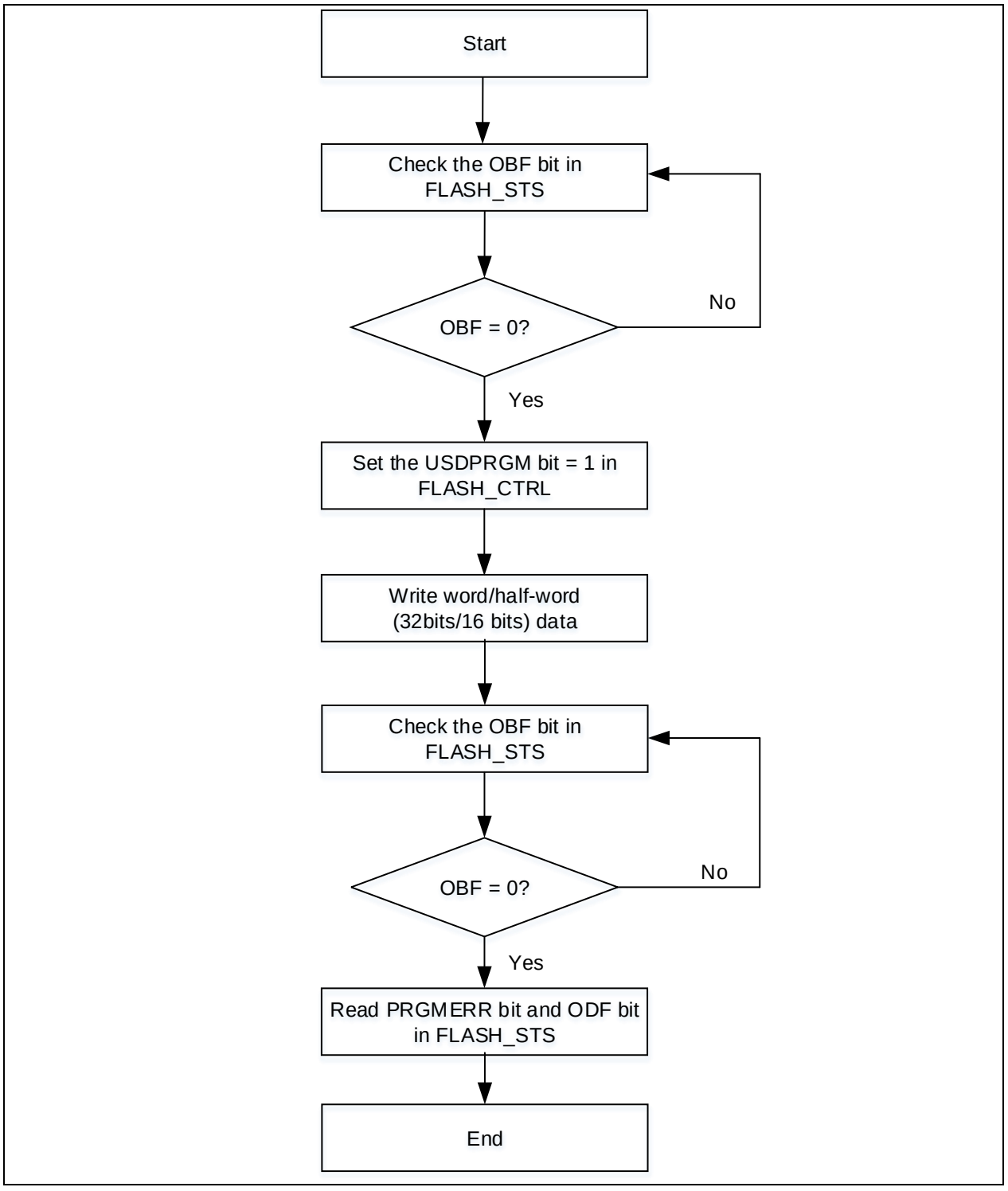
- 检查闪存状态寄存器（FLASH_STS）的OBF位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器（FLASH_CTRL）的USDPRGM位置1，此时可以接受对用户系统数据区的编程指令；
- 对指定的地址写入要编程的数据（任意字/半字）；
- 等待闪存状态寄存器（FLASH_STS）的OBF位变为‘0’，并查询闪存状态寄存器（FLASH_STS）的PRGMERR位和ODF位，确认编程结果。

注意：

编程期间，CPU 会被暂停直到编程完成才处理后续操作。

编程操作前必须保证内部的 HICK 有打开。

图 5-5 系统数据区编程图



5.4.4 读取

通过 CPU 的 AHB 总线可以直接寻址访问用户系统数据区。

5.5 闪存保护

闪存存储器有访问保护以及擦写保护两种保护方式。

5.5.1 访问保护

闪存访问保护分为两类：闪存低级访问保护，闪存高级访问保护。

访问保护启动后，只允许闪存程序对闪存存储器数据进行读出访问，禁止在调试模式下或是从非主闪存存

储器启动对闪存存储器数据的读出访问。

闪存低级访问保护

当 nFAP 字节和 FAP 字节存放的内容不等于 0x5A 和 0xA5 以及不等于 0x33 和 0xCC 时，闪存在系统复位后，将启动闪存低级访问保护。

此保护下，用户可以重新擦除系统数据区，并对 FAP 字节写入 0xA5 解除闪存低级访问保护（从低级保护状态变为未保护状态，将自动产生对主闪存以及主存扩展区的整片擦除操作），最后进行系统复位，系统数据装载器重新加载系统数据信息，更新闪存访问保护解除信息（FAP 字节）。

闪存高级访问保护

当 nFAP 字节存放的内容等于 0x33，并且 FAP 字节存放的内容等于 0xCC 时，闪存在系统复位后，将启动闪存高级访问保护。

一旦此保护启动后，将不能被解除，并且禁止用户以任何方式重新擦除以及写入系统数据区。

注意：

1）主存扩展区也支持访问保护功能

2）如果访问保护被置位的时候仍然处于调试模式，必须用 POR（上电复位）代替系统复位清除调试模式，才能恢复闪存程序访问闪存存储器数据的权限。

下表是启动闪存访问保护后，闪存不同区域访问权限说明：

表 5-5 闪存访问权限

区域	保护等级	访问权限					
		调试模式或是从 SRAM 启动以及从启动程序代码区启动			从主闪存启动		
		读	写	擦除	读	写	擦除
主闪存区	低级访问保护	禁止		禁止（1）（2）	允许		
	高级访问保护	无（3）			允许		
用户系统数据区	低级访问保护	禁止	允许		允许		
	高级访问保护	无（3）			允许	禁止	

（1） 主闪存区会在解除闪存访问保护时被硬件自动擦除

（2） 只禁止扇区擦除，允许整片擦除

（3） 高级访问保护开启时，系统自动从主闪存启动

5.5.2 擦写保护

擦写保护的基本单位为 4KB。通过擦写保护可以防止程序在跑飞时闪存存储器的内容被意外更改。

在下面列出的情况下，擦写将不被允许，并会置位 EPPERR 位：

- 对被设置为擦写保护的扇区（主闪存以及闪存扩展区）做扇区擦除操作以及编程操作将不被允许
- 对存在任一扇区被设置为擦写保护的主闪存以及闪存扩展区做整片擦除将不被允许
- 闪存访问保护启动后，256KB主闪存扇区0~1、128KB与64KB主闪存扇区0~3将被自动擦写保护，不允许做扇区擦除操作以及编程操作
- 闪存访问保护启动后，主存储器和主存扩展区在调试模式或是从非主闪存存储器启动下被自动擦写保护，不允许做扇区擦除操作以及编程操作

5.6 读取性能

提升系统时钟频率前须先按照闪存性能选择寄存器（FLASH_PSR）的 WTCYC 位说明配置读取闪存须插入的延迟时间。

使能闪存性能选择寄存器（FLASH_PSR）的 PFT_EN 位、PFT_EN2 位与 PFT_LAT_DIS 位可降低去闪存读取次数。

5.7 特殊功能

5.7.1 安全库区设定

设定以密码保护主存中指定范围的程序区，即安全库区，仅能被执行，无法写入与删除，除非输入指定密码。安全库区划分为 指令区 与 唯读区。指令区无法读取，唯独区可被读取。

设定安全库区的益处：

以密码保护安全库区，方案商可刻录核心算法到此区域；
安全库区仅能执行，无法被读取，除非输入方案商指定密码，也无法删除（包含 ISP/IAP/SWD）；
其余空白程序区可以提供给方案商客户进行二次开发；
方案商可以藉由安全库功能销售核心算法，不需要每个客户都开发完整方案；
设定安全库区，可防止蓄意破坏或更改终端产品应用程序代码。

注意：
安全库区代码必须以扇区为单位进行烧录，并且起始地址与主存地址对齐；
仅允许 CPU 指令读取指令库区；
写入或擦除安全库区代码（指令区和唯读区），将在闪存状态寄存器（FLASH_STS）的 EPPERR 位置'1'提出警告；
执行主存的整片擦除时，将不会擦除安全库区。

默认状态下，安全库区设定寄存器始终是不可读且被锁定的。要想对安全库区设定寄存器进行写操作，首先要对安全库区解锁，对 SLIB_UNLOCK 寄存器写入 0xA35F6D24 值，通过查看闪存安全库区额外状态寄存器（SLIB_MISC_STS）的位 SLIB_ULKF 确认解锁成功，随后对安全库区设定寄存器写入设定值。
启动安全库区的流程如下：

- 检查闪存状态寄存器（FLASH_STS）的OBF位，以确认没有其他正在进行的编程操作；
- 对SLIB_UNLOCK寄存器写入0xA35F6D24，以进行安全库区解锁；
- 检查闪存安全库区额外状态寄存器（SLIB_MISC_STS）的SLIB_ULKF位，以确认解锁成功；
- 如果安全库区设在主闪存内，需要在SLIB_SET_RANGE寄存器设定要保护的扇区，包含指令区与唯读区的地址；如果安全库区设在主存扩展区域，需要设定EM_SLIB_SET寄存器。
- 等待OBF位变为'0'；
- 在SLIB_SET_PWD寄存器设定安全库区密码；
- 等待OBF位变为'0'；
- 烧录将存入安全库区的代码；
- 进行系统复位，重装载安全库区设定字；
- 读出SLIB_STS0/STS1寄存器用于判断安全库区设定结果。

注意：
不支持同时设定主闪存和主存扩展区域为安全库区；
启动安全库区的流程需要在闪存访问保护未启动时执行

解除安全库区的流程是：

- 在SLIB_PWD_CLR寄存器写入先前设置的安全区域密码；
- 等待OBF位变为'0'；
- 进行系统复位，重装载安全库区设定字；
- 读出SLIB_STS0寄存器用于判断安全库区解除结果。

注意：解除安全库区将会自动执行主存及主存扩展区的整片擦除，以及安全库设定块擦除。

5.7.2 启动程序代码区域作为主存扩展使用

用户只有一次机会将启动程序代码区域作为主存扩展使用。一旦设定成功，主存扩展区将具有主闪存特性。设定启动程序代码区域作为主存扩展使用的流程是：

- 用户读取SLIB_STS0的位0，获知启动程序代码区域当前的模式

- 对SLIB_UNLOCK寄存器写入0xA35F6D24，以进行启动程序代码区域模式设定解锁
- 写非0xFF到BTM_MODE_SET寄存器的位7-0
- 等待OBF位变为'0'；
- 进行系统复位，重装载设定字；
- 读出SLIB_STS0寄存器用于判断设定结果。

注意：启动设定主存扩展区的流程需要在闪存访问保护未启动时执行
当开启该功能后，原启动程序存储器启动模式将强制变为主闪存存储器启动模式。

5.7.3 CRC校验

以扇区为单位对安全库区代码或用户代码进行可选的 CRC 校验。

- CRC生成多项式: 0x4C11DB7，
  即$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$
- CRC初始值：0x00000000

校验流程如下：

- 检查闪存状态寄存器（FLASH_STS）的OBF位，以确认没有其他正在进行的编程操作；
- 在FLASH_CRC_ADDR寄存器设定要校验的代码起始地址；
- 在FLASH_CRC_CTRL寄存器位15-0，设定要校验的代码数量（单位是扇区）；
- 在FLASH_CRC_CTRL寄存器设置位16，启动CRC校验；
- 等待OBF位变为'0'；
- 读出FLASH_CRC_CHKR寄存器用于判断CRC校验结果。

注意：
FLASH_CRC_ADDR 寄存器设定值必须与扇区起始地址对齐；
不允许跨主存及主存扩展区的 CRC 校验。

5.8 FLASH寄存器

必须用字（32 位）的方式操作这些外设寄存器。
表 5-6 闪存接口寄存器映射和复位值

寄存器简称	基址偏移量	复位值
FLASH_PSR	0x00	0x0000 01F0
FLASH_UNLOCK	0x04	0xXXXX XXXX
FLASH_USD_UNLOCK	0x08	0xXXXX XXXX
FLASH_STS	0x0C	0x0000 0000
FLASH_CTRL	0x10	0x0000 0080
FLASH_ADDR	0x14	0x0000 0000
FLASH_USD	0x1C	0x03FF FFFC
FLASH_EPPS	0x20	0xFFFF FFFF
SLIB_STS0	0x74	0x00FF 0000
SLIB_STS1	0x78	0xFFFF FFFF
SLIB_PWD_CLR	0x7C	0xFFFF FFFF
SLIB_MISC_STS	0x80	0x0000 0000
FLASH_CRC_ADDR	0x84	0x0000 0000
FLASH_CRC_CTRL	0x88	0x0000 0000
FLASH_CRC_CHKR	0x8C	0x0000 0000
SLIB_SET_PWD	0x160	0x0000 0000
SLIB_SET_RANGE	0x164	0x0000 0000

EM_SLIB_SET	0x168	0x0000 0000
BTM_MODE_SET	0x16C	0x0000 0000
SLIB_UNLOCK	0x170	0x0000 0000

5.8.1 闪存性能选择寄存器（FLASH_PSR）

域	简称	复位值	类型	功能
位 31:9	保留	0x000000	resd	保持为默认值。
位 8	PFT_LAT_DIS	1	rw	预取时延无效（Prefetch latency disable） 0：闪存预取缓冲区时延开启，访问缓冲区需要等待 1 个系统时钟周期； 1：闪存预取缓冲区时延关闭，访问缓冲区零等待。 推荐此位配置为 1，后续请勿修改。
位 7	PFT_ENF2	1	ro	预取使能标志 2（Prefetch enabled flag2） 该位置起时，表示启动闪存预取缓冲区第二数据块
位 6	PFT_EN2	1	rw	预取使能 2（Prefetch enable2） 0：闪存预取缓冲区第二数据块关闭； 1：闪存预取缓冲区第二数据块开启。 推荐此位配置为 1，后续请勿修改。
位 5	PFT_ENF	1	ro	预取使能标志（Prefetch enabled flag） 该位置起时，表示启动闪存预取缓冲区
位 4	PFT_EN	1	rw	预取使能（Prefetch enable） 0：闪存预取缓冲区关闭； 1：闪存预取缓冲区开启。
位 3	保留	0	resd	须保持为 0。
位 2:0	WTCYC	0x0	rw	等待周期（Wait cycle） 需要根据系统时钟大小来设定闪存访问的等待周期，以系统时钟为单位。 000：零个等待周期，0MHz<系统时钟≤32MHz 使用； 001：一个等待周期，32MHz<系统时钟≤64MHz 使用； 010：两个等待周期，64MHz<系统时钟≤96MHz 使用； 011：三个等待周期，96MHz<系统时钟≤128MHz 使用； 100：四个等待周期，128MHz<系统时钟≤150MHz 使用。

5.8.2 闪存解锁寄存器（FLASH_UNLOCK）

域	简称	复位值	类型	功能
位 31:0	UKVAL	0xXXXX XXXX	wo	解锁键值（Unlock key value） 该寄存器用于解锁主闪存及闪存扩展区。

注意：所有这些位是只写的，读出时返回 0。

5.8.3 闪存用户系统数据解锁寄存器（FLASH_USD_UNLOCK）

域	简称	复位值	类型	功能
位 31:0	USD_UKVAL	0xXXXX XXXX	wo	用户系统数据解锁键值（User system data unlock key value）

注意：所有这些位是只写的，读出时返回 0。

5.8.4 闪存状态寄存器（FLASH_STS）

域	简称	复位值	类型	功能
位 31:6	保留	0x0000000	resd	保持为默认值
位 5	ODF	0	rwc1	操作完成标志（Operation done flag） 当闪存操作（编程/擦除）成功完成时，硬件会置起该位，软件写'1'可以清除。
位 4	EPPERR	0	rwc1	擦写保护错误（Erase/Program protection error） 当擦除或编程的闪存地址在擦写保护设定范围内时，硬件会置起该位，软件写'1'可以清除。

位 3	保留	0	resd	保持为默认值
位 2	PRGMERR	0	rwc1	编程错误（Program error） 当编程的闪存地址的值为非擦除状态时，硬件会置起该位，软件写'1'可以清除。
位 1	保留	0	resd	保持为默认值
位 0	OBF	0	ro	操作忙标志（Operation busy flag） 该位置起表示闪存操作正在进行，该位清除表示操作结束。

5.8.5 闪存控制寄存器（FLASH_CTRL）

域	简称	复位值	类型	功能
位 31:13	保留	0x0000	resd	保持为默认值
位 12	ODFIE	0	rw	操作完成中断使能（Operation done flag interrupt enable） 0：关闭； 1：开启。
位 11,8,3	保留	0	resd	保持为默认值
位 10	ERRIE	0	rw	错误中断使能（Error interrupt enable） 开启后 EPPERR 或 PRGMERR 都会产生中断。 0：关闭； 1：开启。
位 9	USDULKS	0	rw	用户系统数据解锁成功（User system data unlock success） 一旦用户系统数据区解锁成功，该位将被硬件自动置起，表示允许对用户系统数据的编程/擦除操作。软件写'0'可以清除此位，重新锁定用户系统数据区。
位 7	OPLK	1	rw	操作锁定（Operation lock） 该位默认处于置起状态，表示闪存锁定，锁定时不允许操作，解锁成功后，硬件会自动清除此位，表示允许闪存编程/擦除操作。软件写'1'可以重新锁定闪存操作。
位 6	ERSTR	0	rw	擦除开始（Erasing start） 软件置起该位，开始执行擦除操作。擦除完成后硬件自动清除该位。
位 5	USDERS	0	rw	用户系统数据擦除（User system data erase） 用户系统数据区擦除。
位 4	USDPRGM	0	rw	用户系统数据编程（User system data program） 用户系统数据编程。
位 2	BANKERS	0	rw	片擦除（Bank erase） 擦除片操作。
位 1	SECERS	0	rw	扇区擦除（Sector erase） 擦除扇区操作。
位 0	FPRGM	0	rw	闪存编程（Flash program） 编程操作。

5.8.6 闪存地址寄存器（FLASH_ADDR）

域	简称	复位值	类型	功能
位 31:0	FA	0x0000 0000	wo	闪存地址（Flash address） 扇区擦除时选择对应的闪存扇区地址。

5.8.7 用户系统数据寄存器（FLASH_USD）

域	简称	复位值	类型	功能
位 31:27	保留	0x00	resd	保持为默认值
位 26	FAP_HL	0	ro	闪存访问保护高级（Flash access protection high level） 闪存访问保护状态使用 {位 26，位 1}联合判断。 00：未启动访问保护，且 FAP 值=0xA5 01：启动低级访问保护，且 FAP 值非 0xCC 以及 0xA5

				10：保留 11：启动高级访问保护，且 FAP 值=0xCC
位 25:18	USER_D1	0xFF	ro	用户数据 1
位 17:10	USER_D0	0xFF	ro	用户数据 0
位 9:2	SSB	0xFF	ro	系统配置字节（System setting byte） 这里包含加载的用户系统数据区中的系统配置字节 位 9：未用 位 8：nSTDBY_WDT 位 7：nDEPSLP_WDT 位 6：nBOOT1 位 5：未用 位 4：nSTDBY_RST 位 3：nDEPSLP_RST 位 2：nWDT_ATO_EN
位 1	FAP	0	ro	闪存访问保护（Flash access protection）
位 0	USDERR	0	ro	用户系统数据错误（User system data error） 该位置起表示用户系统数据中某字节和它的反码不匹配。此时该字节和它的反码读出值将被硬件自动强制置为 0xFF

5.8.8 擦除编程保护状态寄存器（FLASH_EPPS）

域	简称	复位值	类型	功能
位 31:0	EPPS	0xFFFF FFFF	ro	擦除/编程保护状态（Erase/Program protection status） 该寄存器反映的是加载的用户系统数据中的擦写保护字节状态。

5.8.9 闪存安全库区状态寄存器0（SLIB_STS0）

专用于闪存安全库区。

域	简称	复位值	类型	功能
位 31:24	保留	0x00	resd	保持为默认值
位 23:16	EM_SLIB_INST_SS	0xFF	ro	主存扩展存储区安全库区指令起始扇区（Extension memory sLib instruction start sector） 00000000：扇区 0 00000001：扇区 1 00000010：扇区 2 … 00001001：扇区 9（256KB 主闪存存储器的最后扇区） … 00010011：扇区 19（64KB 与 128KB 主闪存存储器的最后扇区） 11111111：无指令安全区 其余设定值：无效
位 15:4	保留	0x000	resd	保持为默认值
位 3	SLIB_ENF	0	ro	sLib 使能标志（sLib enabled flag） 该位置起时，表示闪存主存区域部分或是全部（依照 SLIB_STS1 设定）作为安全库代码。
位 2	EM_SLIB_ENF	0	ro	主存扩展存储区 sLib 使能标志（Extension memory sLib enabled flag） 该位置起时，表示启动程序代码区域是作为主闪存扩展区域（BTM_AP_ENF 置起），并且存放的应用代码为安全库代码
位 1	保留	0	resd	保持为默认值
位 0	BTM_AP_ENF	0	ro	启动程序代码区域存放应用代码使能标志（Boot memory store application code enabled flag） 该位置起时，表示启动程序代码区域可以作为主存扩展区域存放用户应用代码；否则仅用于存放系统启动代码

5.8.10 闪存安全库区状态寄存器1（SLIB_STS1）

专用于闪存安全库区。

域	简称	复位值	类型	功能
位 31:22	SLIB_ES	0x3FF	ro	主存安全库区结束扇区（sLib end sector） 0000000000：扇区 0 0000000001：扇区 1 0000000010：扇区 2 … 0000111111：扇区 63（64KB 主闪存存储器的最后扇区） … 0001111111：扇区 127（256KB 与 128KB 主闪存存储器的最后扇区）
位 21:11	SLIB_INST_SS	0x7FF	ro	主存安全库区指令区起始扇区（sLib instruction start sector） 00000000000：扇区 0 00000000001：扇区 1 00000000010：扇区 2 … 00000111111：扇区 63（64KB 主闪存存储器的最后扇区） … 00001111111：扇区 127（256KB 与 128KB 主闪存存储器的最后扇区） 11111111111：无安全库区指令区
位 10:0	SLIB_SS	0x7FF	ro	主存安全库区起始扇区（sLib start sector） 00000000000：扇区 0 00000000001：扇区 1 00000000010：扇区 2 … 00000111111：扇区 63（64KB 主闪存存储器的最后扇区） … 00001111111：扇区 127（256KB 与 128KB 主闪存存储器的最后扇区）

5.8.11 闪存安全库区密码清除寄存器（SLIB_PWD_CLR）

专用于闪存安全库区。

域	简称	复位值	类型	功能
位 31:0	SLIB_PCLR_VAL	0x0000 0000	wo	安全库区密码清除（sLib password clear value） 用于写入正确的安全库区密码，将实现解除安全库区功能。 此寄存器写入状态将在闪存安全库区额外状态寄存器（SLIB_MISC_STS）位 0 与位 1 中体现。

5.8.12 闪存安全库区额外状态寄存器（SLIB_MISC_STS）

专用于闪存安全库区。

域	简称	复位值	类型	功能
位 31:3	保留	0x0000000	resd	保持为默认值
位 2	SLIB_ULKF	0	ro	SLib 解锁标志（sLib unlock flag） 当该位置起时表示 SLib 相关设定寄存器允许配置。
位 1	SLIB_PWD_OK	0	ro	密码正确（sLib password ok） 当密码正确，该位被硬件置起。
位 0	SLIB_PWD_ERR	0	ro	密码错误（sLib password error） 当密码错误，并且设定的密码清除寄存器的值不等于 0xFFFF FFFF，该位被硬件置起。

				注意：当该位置起后，硬件将不再接受重新设定密码清除寄存器，直到再次复位。

5.8.13 闪存CRC校验地址寄存器（FLASH_CRC_ADDR）

专用于主闪存以及主存扩展区域。

域	简称	复位值	类型	功能
位 31:0	CRC_ADDR	0x0000 0000	wo	CRC 地址（CRC address） 选择要校验的闪存扇区起始地址。 注意：必须与扇区起始地址对齐

注意：所有这些位是只写的，读出无反应。

5.8.14 闪存CRC校验控制寄存器（FLASH_CRC_CTRL）

专用于主闪存以及主存扩展区域。

域	简称	复位值	类型	功能
位 31:17	保留	0x0000	resd	保持为默认值
位 16	CRC_STRT	0	wo	启动 CRC 校验（CRC start） 设置该位去启动用户代码或是安全库代码的 CRC 校验功能。 硬件启动 CRC 后，会自动清除该位。 注意： 校验数据从 CRC_ADDR ~ CRC_ADDR+CRC_SN*1 扇区
位 15:0	CRC_SN	0x0000	wo	CRC 校验扇区数量（CRC sector numbler） 设定本次 CRC 校验的数据量，单位是扇区

5.8.15 闪存CRC校验结果寄存器（FLASH_CRC_CHKR）

专用于主闪存以及主存扩展区域。

域	简称	复位值	类型	功能
位 31:0	CRC_CHKR	0x0000 0000	ro	CRC 校验结果（CRC check result）

注意：所有这些位是只读的，写入无反应。

5.8.16 闪存安全库区密码设定寄存器（SLIB_SET_PWD）

专用于闪存安全库区密码设定。

域	简称	复位值	类型	功能
位 31:0	SLIB_PSET_VAL	0x0000 0000	wo	安全库区密码（sLib password setting value） 注意：在解除安全库区锁定后，此寄存器才允许被写入，用于设定安全库区启动密码。但写入 0xFFFF_FFFF 以及 0x0000_0000 值无效。

注意：所有这些位是只写入，读出为 0。

5.8.17 闪存安全库区地址设定寄存器（SLIB_SET_RANGE）

专用于主存安全库区地址设定。

域	简称	复位值	类型	功能
位 31:22	SLIB_ES_SET	0x000	wo	主存安全库区结束扇区设定（sLib end sector setting） 用于设定启动安全库区时的安全库区结束扇区位置 0000000000：扇区 0 0000000001：扇区 1 0000000010：扇区 2 …

				0000111111：扇区 63（64KB 主闪存存储器的最后扇区） … 0001111111：扇区 127（256KB 与 128KB 主闪存存储器的最后扇区）
位 21:11	SLIB_ISS_SET	0x000	wo	主存安全库区指令区起始扇区设定（sLib instruction start sector setting） 用于设定启动安全库区时的指令区起始扇区位置 00000000000：扇区 0 00000000001：扇区 1 00000000010：扇区 2 … 00000111111：扇区 63（64KB 主闪存存储器的最后扇区） … 00001111111：扇区 127（256KB 与 128KB 主闪存存储器的最后扇区） 11111111111：无安全库区指令区
位 10:0	SLIB_SS_SET	0x000	wo	主存安全库区起始扇区设定（sLib start sector setting） 用于设定启动安全库区时的安全库区起始扇区位置 00000000000：扇区 0 00000000001：扇区 1 00000000010：扇区 2 … 00000111111：扇区 63（64KB 主闪存存储器的最后扇区） … 00001111111：扇区 127（256KB 与 128KB 主闪存存储器的最后扇区）

注意：
所有这些位是只写入，读出为 0。
在解除安全库区锁定后，此寄存器才允许被写入。
超过主闪存存储地址范围均是无效设定。

5.8.18 主存扩展存储区域安全库区设定寄存器（EM_SLIB_SET）

专用于主存扩展区域。

域	简称	复位值	类型	功能
位 31:24	保留	0x00	resd	保持为默认值
位 23:16	EM_SLIB_ISS_SET	0x000	wo	主存扩展区安全库区指令区起始扇区设定（Extension memory sLib instruction start sector setting） 用于设定启动安全库区时的指令区起始扇区位置 00000000：扇区 0 00000001：扇区 1 00000010：扇区 2 … 00001001：扇区 9（256KB 主闪存存储器的最后扇区） … 00010011：扇区 19（64KB 与 128KB 主闪存存储器的最后扇区） 11111111：无指令安全区 其余设定值：无效 注意： 当设为 0xFF，表示主存扩展区从扇区 0 至扇区 3 都为安全库区，且整个安全库区作为唯读取；
位 15:0	EM_SLIB_SET	0x000	wo	主存扩展存储区 sLib 设定（Extension memory sLib setting） 写入 0x5AA5 将启动主存扩展区作为存放安全库区代码功能

注意：　所有这些位是只写的，读出无反应。

5.8.19 启动程序代码区模式设定寄存器（BTM_MODE_SET）

专用于启动程序代码区域。

域	简称	复位值	类型	功能
位 31:8	保留	0x000000	resd	保持为默认值
位 7:0	BTM_MODE_SET	0x00	wo	启动程序代码区模式设定（Boot memory mode setting） 0xFF：启动程序代码区域作为系统区域，存放系统启动代码功能 其他值：启动程序代码区域作为主存扩展区域，存放应用代码功能 注意：此寄存器的设定需要在闪存访问保护未启动下进行

注意：所有这些位是只写的，读出无反应。

5.8.20 闪存安全库区解锁寄存器（SLIB_UNLOCK）

专用于安全库区寄存器的解锁设定。

域	简称	复位值	类型	功能
位 31:0	SLIB_UKVAL	0x0000 0000	wo	安全库区解锁键值（sLib unlock key value） 固定键值 0xA35F_6D24，用于安全库区设定寄存器的解锁。

注意：所有这些位是只写入，读出为 0。

6 通用和复用功能 I/O（GPIO 和 IOMUX）

6.1 简介

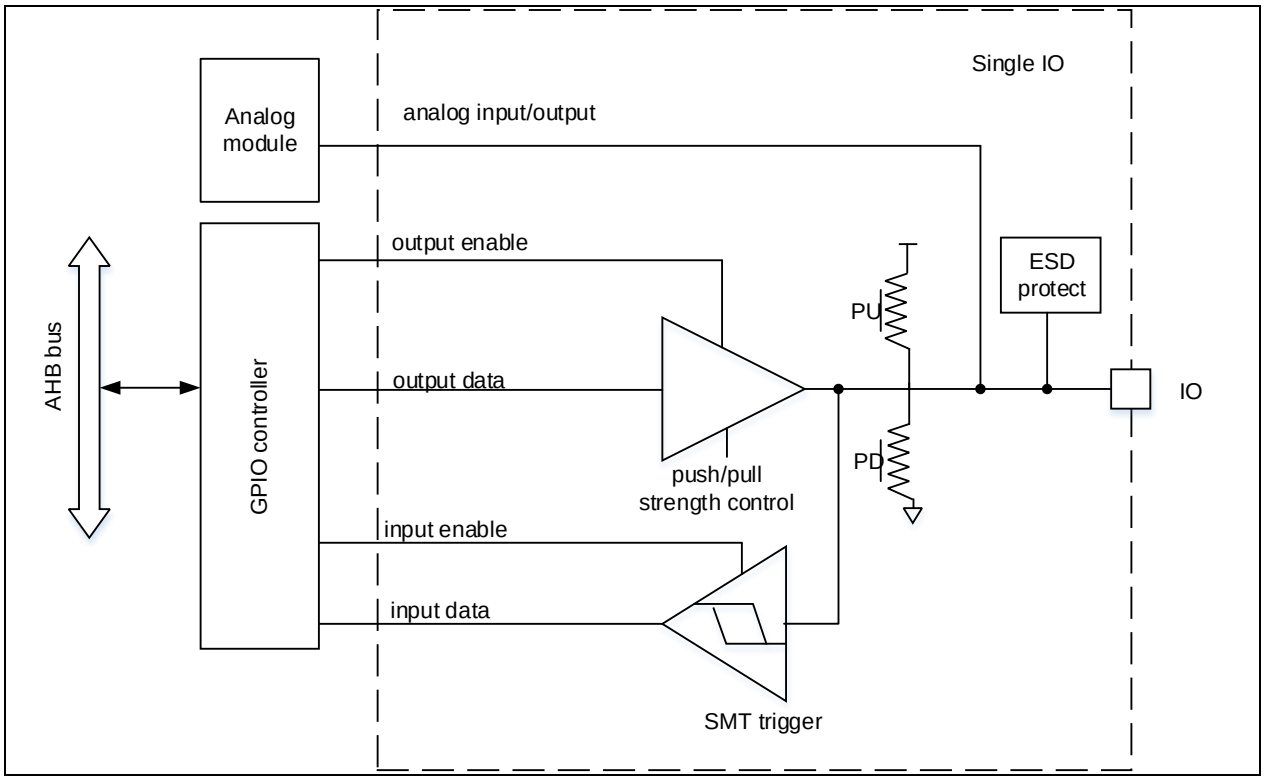
AT32F423 支持多达 87 个双向 I/O 引脚，分别为 PA0-PA15、PB0-PB15、PC0-PC15、PD0-PD15、PE0-PE15、PF0-PF2、PF6、PF8-PF10、每个引脚都可以实现与外部的通讯、控制以及数据采集的功能。
每个引脚都可以软件配置成通用输入、通用输出、复用功能和模拟引脚。当配置成通用输出或复用功能时，可选择推挽或是开漏输出。
每个引脚都有独立的弱上拉/下拉功能。
每个引脚都可以软件配置输出驱动能力。
每个引脚都可以配置为外部中断输入。
每个引脚都支持配置写保护功能。

6.2 功能描述

6.2.1 GPIO结构

每个引脚对应的寄存器允许字、半字或字节访问，每个引脚可以自由编程。下图给出了一个 I/O 端口位的基本结构。

图 6-1 GPIO基本结构



注意： PC13 所对应的 GPIO 功能以及相关的 RTC 功能初始上电时不能直接使用，如要使用请参考 ES0010_AT32F423_Errata_Sheet GPIO 章节

6.2.2 GPIO复位状态

系统上电或复位后，所有引脚除了 JTAG 相关引脚以外，都被配置为浮空输入模式。JTAG 相关引脚则配置为：PA15/JTDI 、PA13/JTMS 和 PB4/JNTRST 为复用功能上拉模式，PA14/JTCK 为复用功能下拉模式，PB3/TDO 为复用功能无上拉下拉模式。

6.2.3 通用功能输入配置

配置模式	IOMC	PULL
通用浮空输入	00	00
通用下拉输入		10
通用上拉输入		01

当引脚被配置为通用输入时：

- 引脚状态可通过对输入数据寄存器的读访问得到
- 施密特触发器有效

注意： 如果是浮空输入模式，为避免复杂环境下，没有使用的引脚有干扰，导致漏电，建议引脚如不使用，则配置为模拟模式。

6.2.4 模拟配置

配置模式	IOMC	PULL
模拟	11	不使用

当引脚被配置为模拟时：

- 施密特触发无效
- 输出数据寄存器无效
- 设置上拉/下拉无效
- 引脚状态不可通过输入数据寄存器访问得到

6.2.5 通用输出配置

配置模式	IOMC	OM	HDRV	ODRV[1：0]	PULL
通用推挽无上拉/下拉	01	0	000：输出模式，适中电流推动/吸入能力 001：输出模式，较大电流推动/吸入能力 010：输出模式，适中电流推动/吸入能力 011：输出模式，适中电流推动/吸入能力 1xx：输出模式，极大电流推动/吸入能力		00 或 11
通用推挽上拉	01	0			01
通用推挽下拉	01	0			10
通用开漏无上拉/下拉	01	1			00 或 11
通用开漏上拉	01	1			01
通用开漏下拉	01	1			10

当引脚被配置为通用输出时：

- 施密特触发器有效
- 在开漏输出模式时，当输出数据寄存器配置为数字0时，引脚输出数字0；当输出数据寄存器配置为数字1时，无内部上拉和下拉时，引脚为高阻状态，有内部上拉时，引脚上拉到数字1，有内部下拉时，引脚下拉到数字0
- 在推挽输出模式时，可通过输出数据寄存器输出数字0/1
- 引脚状态可通过对输入数据寄存器的读访问得到
- 通过GPIO设置/清除寄存器控制对应的GPIO数据输出寄存器的设置/清除

注意： GPIO 设置/清除寄存器 对应同一个引脚的 IOCB/IOSB 同时写 1，IOSB 优先级高于 IOCB

6.2.6 GPIO端口写保护

为了防止误操作导致 GPIO 功能混乱，提供每个对应引脚的的写保护机制。一旦设定写保护，在下次复位或者上电之前都不能进行对应引脚的 GPIO 配置。

6.2.7 IOMUX功能结构

大多数引脚支持多个外设的输出功能映射，通过 IOMUX 功能输入/输出章节查找表来选择每个引脚对应的外设输入输出功能。通过引脚所对应的 GPIOx_MUXL（从引脚 0 到引脚 7）或 GPIOx_MUXH(从引脚 8 到引脚 15）进行对应的设置,单一引脚有多达 16 种不同的 IOMUX 映射方案，方便灵活选用

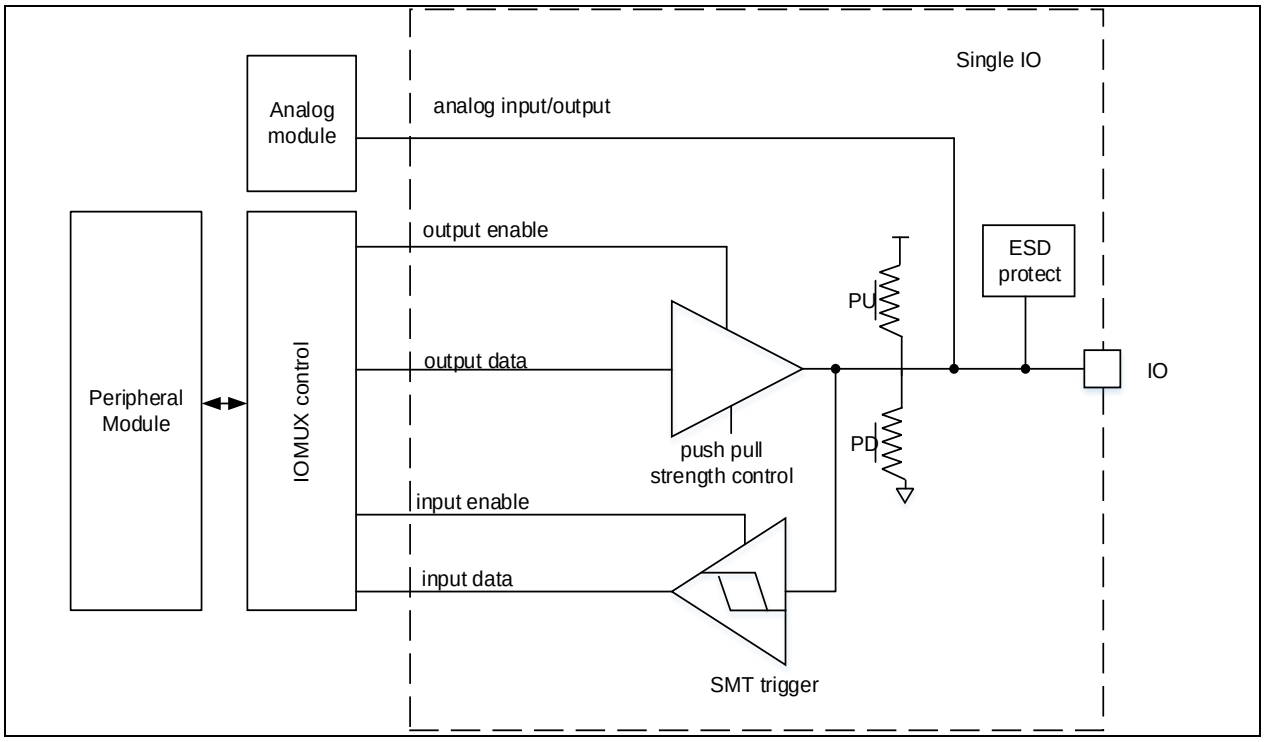
每个引脚通过设定 GPIOx_MUXL 或 GPIOx_MUXH，只会和单一外设的单一脚进行对应，不存在单一引脚多个外设抢占的冲突。

引脚作为复用功能时，引脚和 GPIO 控制器断开，由 IOMUX 控制器进行控制。

引脚作为复用输入时，配置 GPIOx_CFGR 将该引脚设定为复用模式，根据外设特性可配置 GPIOx_PULL 将引脚配置为浮空、上拉或下拉。

引脚作为复用输出或双向复用功能时，配置 GPIOx_CFGR 将该引脚设定为复用模式，根据外设特性可配置 GPIOx_OMODE 将引脚配置为推挽或开漏模式，可配置 GPIOx_ODRVR 和 GPIOx_HDRV 配置引脚的输出驱动能力以及 GPIOx_PULL 配置引脚的上下拉。

图 6-2 IOMXU复用结构



6.2.8 复用功能配置

配置模式	IOMC	OM	HDRV	ODRV[1：0]	PULL
复用推挽无上拉/下拉	10	0	000：输出模式，适中电流推动/吸入能力 001：输出模式，较大电流推动/吸入能力 010：输出模式，适中电流推动/吸入能力 011：输出模式，适中电流推动/吸入能力 1xx：输出模式，极大电流推动/吸入能力 仅对复用输出或双向复用功能起作用		00 或 11
复用推挽上拉	10	0			01
复用推挽下拉	10	0			10
复用开漏无上拉/下拉	10	1			00 或 11
复用开漏上拉	10	1			01
复用开漏下拉	10	1			10
复用输入无上拉/下拉	10	x			00 或 11
复用输入上拉	10	x			01
复用输入下拉	10	x			10

当引脚被配置为复用功能时：

- 施密特触发器有效
- 在开漏输出模式时，当外设输出为数字0时，引脚输出数字0；当外设输出为数字1时，无内部上拉和下拉时，引脚为高阻状态，有内部上拉时，引脚上拉到数字1，有内部下拉时，引脚下拉到数字0
- 在推挽输出模式时，根据外设的输出引脚输出数字0/1
- 引脚状态可通过对输入数据寄存器的读访问得到

6.2.9 IOMUX功能输入/输出

选择每个端口线的有效复用功能是通过 GPIOx_MUXL（从引脚 0 到引脚 7）或 GPIOx_MUXH(从引脚 8 到引脚 15）进行设置。

表 6-1 通过GPIO_MUX*寄存器配置端口A的复用功能

引脚名	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PA0		TMR2_CH1 TMR2_EXT		TMR9_CH2C	I2C2_SCL		USART2_RX	USART2_CTS
PA1		TMR2_CH2		TMR9_CH1C	I2C2_SDA	I2C1_SMBA	SPI3_CS/I2S3_WS	USART2_RTS_DE
PA2		TMR2_CH3		TMR9_CH1				USART2_TX
PA3		TMR2_CH4		TMR9_CH2		I2S2_MCK		USART2_RX
PA4					I2C1_SCL	SPI1_CS/I2S1_WS	SPI3_CS/I2S3_WS	USART2_CK
PA5		TMR2_CH1 TMR2_EXT				SPI1_SCK/I2S1_CK	USART3_CK	USART3_RX
PA6		TMR1_BRK	TMR3_CH1			SPI1_MISO/I2S1_MCK	I2S2_MCK	USART3_CTS
PA7		TMR1_CH1C	TMR3_CH2		I2C3_SCL	SPI1_MOSI/I2S1_SD		USART3_TX
PA8	CLKOUT	TMR1_CH1		TMR9_BRK	I2C3_SCL			USART1_CK
PA9	CLKOUT	TMR1_CH2			I2C3_SMBA	SPI2_SCK/I2S2_CK		USART1_TX
PA10	ERTC_REFIN	TMR1_CH3				SPI2_MOSI/I2S2_SD		USART1_RX
PA11		TMR1_CH4			I2C2_SCL	SPI2_CS/I2S2_WS	I2C1_SMBA	USART1_CTS
PA12		TMR1_EXT			I2C2_SDA	SPI2_MISO/I2S2_MCK		USART1_RTS_DE
PA13	JTMS SWDIO	IR_OUT			I2C1_SDA	I2S_SDEXT	SPI3_MISO/I2S3_MCK	
PA14	JTCK SWCLK				I2C1_SMBA		SPI3_MOSI/I2S3_SD	
PA15	JTDI	TMR2_CH1 TMR2_EXT				SPI1_CS/I2S1_WS	SPI3_CS/I2S3_WS	USART1_TX

引脚名	MUX8	MUX9	MUX10	MUX11	MUX12	MUX13	MUX14	MUX15
PA0	USART4_TX							EVENTOUT
PA1	USART4_RX							EVENTOUT
PA2		CAN2_RX			XMC_D4			EVENTOUT
PA3		CAN2_TX			XMC_D5			EVENTOUT
PA4	USART6_TX	TMR14_CH1	OTGFS_OE		XMC_D6			EVENTOUT
PA5	USART6_RX	TMR13_CH1C			XMC_D7			EVENTOUT
PA6	USART3_RX	TMR13_CH1						EVENTOUT
PA7		TMR14_CH1						EVENTOUT
PA8	USART2_TX	USART7_RX	OTGFS_SOF					EVENTOUT
PA9	I2C1_SCL	TMR14_BRK	OTGFS_VBUS					EVENTOUT
PA10	I2C1_SDA		OTGFS_ID					EVENTOUT
PA11	USART6_TX	CAN1_RX						EVENTOUT
PA12	USART6_RX	CAN1_TX						EVENTOUT
PA13			OTGFS_OE					EVENTOUT
PA14	USART2_TX							EVENTOUT
PA15	USART2_RX	USART7_TX	USART4_RTS_DE		XMC_NE2			EVENTOUT

表 6-2 通过GPIO_MUX*寄存器配置端口B的复用功能

引脚名	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PB0		TMR1_CH2C	TMR3_CH3			SPI1_MISO / I2S1_MCK	SPI3_MOSI/I2S3_SD	USART2_RX
PB1		TMR1_CH3C	TMR3_CH4			SPI1_MOSI / I2S1_SD	SPI2_SCK/I2S2_CK	USART2_CK
PB2		TMR2_CH4	TMR3_EXT		I2C3_SMBA		SPI3_MOSI/I2S3_SD	
PB3	JTDO SWO	TMR2_CH2			I2C2_SDA	SPI1_SCK/I2S1_CK	SPI3_SCK/I2S3_CK	USART1_RX
PB4	JNTRST		TMR3_CH1	TMR11_BRK	I2C3_SDA	SPI1_MISO/I2S1_MCK	SPI3_MISO/I2S3_MCK	USART1_CTS
PB5			TMR3_CH2	TMR10_BRK	I2C3_SMBA	SPI1_MOSI/I2S1_SD	SPI3_MOSI/I2S3_SD	USART1_CK
PB6			TMR4_CH1	TMR10_CH1C	I2C1_SCL	I2S1_MCK	SPI3_CS / I2S3_WS	USART1_TX
PB7			TMR4_CH2	TMR11_CH1C	I2C1_SDA		SPI3_SCK / I2S3_CK	USART1_RX
PB8		TMR2_CH1 TMR2_EXT	TMR4_CH3	TMR10_CH1	I2C1_SCL		SPI3_MISO / I2S3_MCK	USART1_TX
PB9	IR_OUT	TMR2_CH2	TMR4_CH4	TMR11_CH1	I2C1_SDA	SPI2_CS/I2S2_WS	SPI3_MOSI / I2S3_SD	I2C2_SDA
PB10		TMR2_CH3			I2C2_SCL	SPI2_SCK/I2S2_CK	I2S3_MCK	USART3_TX
PB11		TMR2_CH4			I2C2_SDA			USART3_RX
PB12		TMR1_BRK		TMR12_BRK	I2C2_SMBA	SPI2_CS/I2S2_WS	SPI3_SCK/I2S3_CK	
PB13	CLKOUT	TMR1_CH1C		TMR12_CH1C	I2C3_SMBA	SPI2_SCK/I2S2_CK		I2C3_SCL
PB14		TMR1_CH2C			I2C3_SDA	SPI2_MISO/I2S2_MCK	I2S_SDEXT	USART3_RTS_DE
PB15	ERTC_REFIN	TMR1_CH3C		TMR12_CH1C	I2C3_SCL	SPI2_MOSI/I2S2_SD		

引脚名	MUX8	MUX9	MUX10	MUX11	MUX12	MUX13	MUX14	MUX15
PB0	USART3_CK							EVENTOUT
PB1	USART3_RTS_DE	TMR14_CH1						EVENTOUT
PB2		TMR14_CH1C						EVENTOUT
PB3	USART1_RTS_DE	USART7_RX	USART5_TX					EVENTOUT
PB4	I2S_SDEXT	USART7_TX	USART5_RX					EVENTOUT
PB5	USART5_RX	CAN2_RX	USART5_RTS_DE					EVENTOUT
PB6	USART5_TX	CAN2_TX	USART4_CK					EVENTOUT
PB7	USART4_CTS				XMC_NADV			EVENTOUT
PB8	USART5_RX	CAN1_RX						EVENTOUT
PB9	USART5_TX	CAN1_TX	I2S1_MCK					EVENTOUT
PB10					XMC_NOE			EVENTOUT
PB11		TMR13_BRK						EVENTOUT
PB12	USART3_CK	CAN2_RX			XMC_D13			EVENTOUT
PB13	USART3_CTS	CAN2_TX						EVENTOUT
PB14		TMR12_CH1			XMC_D0			EVENTOUT
PB15		TMR12_CH2						EVENTOUT

表 6-3 通过GPIO_MUX*寄存器配置端口C的复用功

引脚名	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PC0					I2C3_SCL			I2C1_SCL
PC1					I2C3_SDA	SPI3_MOSI/I2S3_SD	SPI2_MOSI/I2S2_SD	I2C1_SDA
PC2						SPI2_MISO/I2S2_MCK	I2S_SDEXT	
PC3						SPI2_MOSI/I2S2_SD		
PC4				TMR9_CH1		I2S1_MCK		USART3_TX
PC5				TMR9_CH2	I2C1_SMBA			USART3_RX
PC6		TMR1_CH1	TMR3_CH1		I2C1_SCL	I2S2_MCK		
PC7		TMR1_CH2	TMR3_CH2		I2C1_SDA	SPI2_SCK/I2S2_CK	I2S3_MCK	
PC8		TMR1_CH3	TMR3_CH3					USART8_TX
PC9	CLKOUT	TMR1_CH4	TMR3_CH4		I2C3_SDA			USART8_RX
PC10							SPI3_SCK/I2S3_CK	USART3_TX
PC11						I2S_SDEXT	SPI3_MISO/I2S3_MCK	USART3_RX
PC12				TMR11_CH1	I2C2_SDA		SPI3_MOSI/I2S3_SD	USART3_CK
PC13								
PC14								
PC15								

引脚名	MUX8	MUX9	MUX10	MUX11	MUX12	MUX13	MUX14	MUX15
PC0	USART6_TX	USART7_TX						EVENTOUT
PC1	USART6_RX	USART7_RX						EVENTOUT
PC2	USART8_TX				XMC_NWE			EVENTOUT
PC3	USART8_RX				XMC_A0			EVENTOUT
PC4		TMR13_CH1			XMC_NE4			EVENTOUT
PC5		TMR13_CH1C			XMC_NOE			EVENTOUT
PC6	USART6_TX	USART7_TX			XMC_D1			EVENTOUT
PC7	USART6_RX	USART7_RX			XMC_NADV			EVENTOUT
PC8	USART6_CK							EVENTOUT
PC9	I2C1_SDA		OTGFS_OE					EVENTOUT
PC10	USART4_TX							EVENTOUT
PC11	USART4_RX				XMC_D2			EVENTOUT
PC12	USART4_CK		USART5_TX		XMC_D3			EVENTOUT
PC13								EVENTOUT
PC14								EVENTOUT
PC15								EVENTOUT

表 6-4 通过GPIO_MUX*寄存器配置端口D的复用功能

引脚名	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PD0							SPI3_MOSI/I2S3_SD	SPI2_CS/I2S2_WS
PD1							SPI2_SCK/I2S2_CK	SPI2_CS/I2S2_WS
PD2			TMR3_EXT					USART3_RTS_DE
PD3						SPI2_SCK/I2S2_CK	SPI2_MISO/I2S2_MCK	USART2_CTS
PD4							SPI2_MOSI/I2S2_SD	USART2_RTS_DE
PD5								USART2_TX
PD6						SPI3_MOSI/I2S3_SD		USART2_RX
PD7								USART2_CK
PD8								USART3_TX
PD9								USART3_RX
PD10								USART3_CK
PD11					I2C2_SMBA			USART3_CTS
PD12			TMR4_CH1		I2C2_SCL			USART3_RTS_DE
PD13			TMR4_CH2		I2C2_SDA			
PD14			TMR4_CH3		I2C3_SCL			
PD15			TMR4_CH4		I2C3_SDA			

引脚名	MUX8	MUX9	MUX10	MUX11	MUX12	MUX13	MUX14	MUX15
PD0	USART4_RX	CAN1_RX			XMC_D2			EVENTOUT
PD1	USART4_TX	CAN1_TX			XMC_D3			EVENTOUT
PD2	USART5_RX				XMC_NWE			EVENTOUT
PD3					XMC_CLK			EVENTOUT
PD4					XMC_NOE			EVENTOUT
PD5					XMC_NWE			EVENTOUT
PD6					XMC_NWAIT			EVENTOUT
PD7					XMC_NE1			EVENTOUT
PD8		TMR12_CH2C			XMC_D13			EVENTOUT
PD9					XMC_D14			EVENTOUT
PD10	USART4_TX				XMC_D15			EVENTOUT
PD11					XMC_A16			EVENTOUT
PD12	USART8_CK_RTS_DE				XMC_A17			EVENTOUT
PD13	USART8_TX				XMC_A18			EVENTOUT
PD14	USART8_RX				XMC_D0			EVENTOUT
PD15		USART7_CK_RTS_DE			XMC_D1			EVENTOUT

表 6-5通过GPIO_MUX*寄存器配置端口E的复用功能

引脚名	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PE0			TMR4_EXT					
PE1		TMR1_CH2C						
PE2			TMR3_EXT	TMR9_BRK				
PE3			TMR3_CH1	TMR9_CH2C				
PE4			TMR3_CH2	TMR9_CH1C				
PE5			TMR3_CH3	TMR9_CH1				
PE6			TMR3_CH4	TMR9_CH2				
PE7		TMR1_EXT						
PE8		TMR1_CH1C						
PE9		TMR1_CH1						
PE10		TMR1_CH2C						
PE11		TMR1_CH2						
PE12		TMR1_CH3C				SPI1_CS/I2S1_WS		
PE13		TMR1_CH3				SPI1_SCK/I2S1_CK		
PE14		TMR1_CH4				SPI1_MISO/I2S1_MCK		
PE15		TMR1_BRK				SPI1_MOSI/I2S1_SD		

引脚名	MUX8	MUX9	MUX10	MUX11	MUX12	MUX13	MUX14	MUX15
PE0	USART8_RX	TMR13_CH1			XMC_LB			EVENTOUT
PE1	USART8_TX	TMR14_CH1			XMC_UB			EVENTOUT
PE2		TMR14_CH1C			XMC_A23			EVENTOUT
PE3		TMR14_BRK			XMC_A19			EVENTOUT
PE4					XMC_A20			EVENTOUT
PE5					XMC_A21			EVENTOUT
PE6					XMC_A22			EVENTOUT
PE7	USART5_CK	USART7_RX			XMC_D4			EVENTOUT
PE8	USART4_TX	USART7_TX			XMC_D5			EVENTOUT
PE9	USART4_RX				XMC_D6			EVENTOUT
PE10	USART5_TX				XMC_D7			EVENTOUT
PE11	USART5_RX				XMC_D8			EVENTOUT
PE12					XMC_D9			EVENTOUT
PE13					XMC_D10			EVENTOUT
PE14					XMC_D11			EVENTOUT
PE15					XMC_D12			EVENTOUT

表 6-6通过GPIO_MUX*寄存器配置端口F的复用功能

引脚名	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PF0		TMR1_CH1			I2C1_SDA			
PF1		TMR1_CH2C			I2C1_SCL	SPI2_CS / I2S2_WS		
PF2						SPI2_SCK/I2S2_CK		
PF6		TMR2_CH1			I2C2_SCL			
PF8		TMR2_CH2			I2C2_SDA			
PF9			TMR4_CH1					
PF10			TMR4_CH2					

引脚名	MUX8	MUX9	MUX10	MUX11	MUX12	MUX13	MUX14	MUX15
PF0								EVENTOUT
PF1								EVENTOUT
PF2		USART7_CK_RTS_DE						EVENTOUT
PF6		USART7_RX						EVENTOUT
PF8		USART7_TX						EVENTOUT
PF9	USART6_TX	TMR12_CH1						EVENTOUT
PF10	USART6_RX	TMR12_CH2						EVENTOUT

注意：EVENTOUT 是 Cortex-M 的 TXEV 信号

6.2.10 外设复用功能引脚配置

当外设需要使用 IOMUX 复用功能时：

- 如果外设引脚需要作为复用输出则对应的引脚配置成复用推挽/开漏输出
- 如果外设引脚需要作为复用输入则对应的引脚配置成复用模式（浮空/上拉/下拉）
- ADC外设需要将模拟通道对应的引脚配置为模拟输入/输出模式
- I^2C外设需要对应引脚作为双向复用功能时，需把对应的引脚配置复用开漏模式
- USB的OTGFS_ID引脚只需配置IOMUX对应的复用，以及CRM 对应的时钟使能就可以，不需要配置GPIO 状态。

6.2.11 IOMUX映射优先级

除了极个别引脚可能会被硬件直接抢占，其他外设都可通过配置 GPIOx_MUXL/GPIOx_MUXH 寄存器得到唯一外设复用。

某些引脚不管GPIO配置为任何模式，都会被特定的硬件功能直接占用。

表 6-7 硬件抢占功能

引脚名字	抢占使能位	说明
PA0	PWC_CTRLSTS[8] =1	抢占使能位有效之后，PA0 引脚直接作为 PWC 的 WKUP1 功能使用
PC13	PWC_CTRLSTS[9] = 1	抢占使能位有效之后，PC13 引脚直接作为 PWC 的 WKUP2 功能使用
PB5	PWC_CTRLSTS[13] =1	抢占使能位有效之后，PB5 引脚直接作为 PWC 的 WKUP6 功能使用
PB15	PWC_CTRLSTS[14] =1	抢占使能位有效之后，PB15 引脚直接作为 PWC 的 WKUP7 功能使用
PC13	(ERTC_CTRL[23]=1)| (ERTC_CTRL[22：21]!=00)| (ERTC_CTRL[11]=1& ERTC_TAMP[17]=0)| (ERTC_TAMP[0]=1& ERTC_TAMP[16]=0)	抢占使能位有效之后，PC13 作为 RTC 通道使用
PA0	(ERTC_CTRL[11]=1& ERTC_TAMP[17]=1)| (ERTC_TAMP[0]=1& ERTC_TAMP[16]=1)| (ERTC_TAMP[3]=1)	抢占使能位有效之后，PA0 引脚直接作为 TAMPER2_BPR 功能使用
PC14	CRM_BPDC[0]=1	抢占使能位有效之后，PC14 作为 LEXT 通道使用
PC15	CRM_BPDC[0]=1 & CRM_BPDC[2]=0	抢占使能位有效之后，PC15 作为 LEXT 通道使用
PA4	DAC_CTRL[2] =1	抢占使能位有效之后，PA4 作为 DAC1 模拟通道使用
PA5	DAC_CTRL[18] =1	抢占使能位有效之后，PA5 作为 DAC2 模拟通道使用
PF0	CRM_CTRL[16]=1	抢占使能位有效之后，PF0 作为 HEXT 通道使用
PF1	CRM_CTRL[16]=1& CRM_CTRL[18]=0	抢占使能位有效之后，PF1 作为 HEXT 通道使用
PA11	CRM_AHBEN2[7] & OTGFS_GCCFG[16]	抢占使能位有效之后，PA11 作为 OTGFS_D-通道使用
PA12	CRM_AHBEN2[7] & OTGFS_GCCFG[16]	抢占使能位有效之后，PA12 作为 OTGFS_D+通道使用

注意： PA0 或者 PC13 不能同时使能 TAMPER_BPR 功能和 PWC 的 WKUP 功能。

6.2.12 外部中断/唤醒线

每个引脚都支持作为外部中断的输入，对应的引脚须配置为输入模式。

6.3 GPIO寄存器

下面列出了 GPIO 寄存器映像和复位数值。
可以用字节（8 位）、半字（16 位）或字（32 位）的方式操作这些外设寄存器。
表 6-8 GPIO寄存器地址映像和复位值

寄存器简称	基址偏移量	复位值
GPIOA_CFGR	0x00	0xA800 0000
GPIOx_CFGR(x =B,C,F)	0x00	0x0000 0280(B) 0x0000 0000
GPIOx_OMODER	0x04	0x0000 0000
GPIOx_ODRVR	0x08	0x0000 00C0(B) 0x0000 0000
GPIOA_PULL	0x0C	0x6400 0000(A)
GPIOx_PULL(x = B,C,F)	0x0C	0x0000 0100(B) 0x0000 0000
GPIOx_IDT	0x10	0x0000 XXXX
GPIOx_ODT	0x14	0x0000 0000
GPIOx_SCR	0x18	0x0000 0000
GPIOx_WPR	0x1C	0x0000 0000
GPIOx_MUXL	0x20	0x0000 0000
GPIOx_MUXH	0x24	0x0000 0000
GPIOx_CLR	0x28	0x0000 0000
GPIOx_TGR	0x2C	0x0000 0000
GPIOx_HDRV	0x3C	0x0000 0000

6.3.1 GPIO配置寄存器（GPIOx_CFGR）（x=A..F）

复位值：0xa8000000 端口 A
0x0000 0280 端口 B
0x00000000 其它端口

域	简称	复位值	类型	功能
位 2y+1：2y	IOMCy	0xA800 0000	rw	GPIOx 模式配置（y=0~15）（GPIOx mode configurate） 用于配置 GPIOx 的工作模式： 00：输入（复位后的模式） 01：通用输出 10：复用功能 11：模拟

6.3.2 GPIO输出模式寄存器（GPIOx_OMODE）（x=A..F）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	始终读为 0。
位 15：0	OM	0x0000	rw	GPIOx 的输出模式配置（y=0..15）（GPIOx output mode configurate） 当 GPIOx 用作输出时，可选择以下两种输出模式： 0：推挽（复位状态） 1：开漏

6.3.3 GPIO电流推动/吸入能力切换控制寄存器（GPIOx_ODRVR）

（x=A..F）

复位值：0x0000 00C0 端口 B
0x00000000 其它端口

域	简称	复位值	类型	功能
位 2y+1：2y	ODRVy	0x0000 0000	rw	GPIOx 的驱动能力配置（y=0…15）（GPIOx drive capability） 用于配置相应的 I/O 端口电流能力： x0：适中电流推动/吸入能力 01：较大电流推动/吸入能力 11：适中电流推动/吸入能力

6.3.4 GPIO上/下拉寄存器（GPIOx_PULL）（x=A..F）

复位值：0x6400 0000 端口 A
0x0000 0100 端口 B
0x00000000 其它端口

域	简称	复位值	类型	功能
位 2y+1：2y	PULLy	0x6400 0000	rw	GPIOx 的上下拉配置（y=0…15）（GPIOx pull configurate） 用于配置相应的 I/O 上拉或下拉。 00：无作用 01：上拉 10：下拉

6.3.5 GPIO输入数据寄存器（GPIOx_IDT）（x=A..F）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	始终读为 0。
位 15：0	IDT	0xXXXX	ro	GPIOx 输入的数据（GPIOx input data） GPIOx 对应 IO 口的输入电平状态，每一位对应 GPIOx 的一个 IO。

6.3.6 GPIO输出数据寄存器（GPIOx_ODT）（x=A..F）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	始终读为 0。
位 15：0	ODT	0x0000	rw	GPIOx 输出的数据（IO output data）。 每一位对应 GPIOx 的一个 IO。 GPIOx 对应 IO 口的输出电平状态。 0：低电平； 1：高电平。

6.3.7 GPIO设置/清除寄存器（GPIOx_SCR）（x=A..F）

域	简称	复位值	类型	功能
位 31：16	IOCB	0x0000	wo	清除 GPIOx 位（GPIOx clear bit）

				写'1'的位其对应 ODT 寄存器位会清除，写'0'的位其对应 ODT 寄存器位维持不变，相当于 ODT 寄存器的位操作。 0：对应位不变； 1：对应位清除。
位 15：0	IOSB	0x0000	wo	设置 GPIOx 位（GPIOx set bit） 写'1'的位其对应 ODT 寄存器位会置起，写'0'的位其对应 ODT 寄存器位维持不变，相当于 ODT 寄存器的位操作。 如果 IOCB 和 IOSB 同一个位都写'1'，那么优先级更高的 IOSB 会生效。 0：对应位不变； 1：对应位置起。

6.3.8 GPIO写保护寄存器（GPIOx_WPR）（x=A..F）

域	简称	复位值	类型	功能
位 31：17	保留	0x0000	resd	保持为默认值。
位 16	WPSEQ	0x0	rw	写保护使能序列（Write protect sequence） 想保护某些 IO 位不被写入，需配合同时操作写保护使能序列位和 WPEN 位。 写保护使能位操作按照以下方式操作 4 次，写'1' ->写'0' ->写'1' ->读，操作期间 WPEN 位值不可修改。
位 15：0	WPEN	0x0000	rw	写保护使能（Write protect enable） 每一位对应 GPIOx 的一个 IO。 0：无写保护； 1：写保护。

6.3.9 GPIO复用低位寄存器（GPIOx_MUXL）（x=A..F）

域	简称	复位值	类型	功能
位 4y+3：4y	MUXLy	0x0	rw	GPIOx 引脚 y 的复用功能选择（y=0…7）（GPIOx pin y muxing） 用于配置对应 IO 口的复用功能。 0000：MUX0 0001：MUX1 0010：MUX2 0011：MUX3 0100：MUX4 0101：MUX5 0110：MUX6 0111：MUX7 1000：MUX8 1001：MUX9 1010：MUX10 1011：MUX11 1100：MUX12 1101：MUX13 1110：MUX14 1111：MUX15

6.3.10 GPIO复用高位寄存器（GPIOx_MUXH）（x=A..F）

域	简称	复位值	类型	功能
位 4y+3：4y	MUXHy	0x0	rw	端口 x 引脚 y 的复用功能选择（y=8…15）（GPIOx pin y muxing） 用于配置对应 IO 口的复用功能。

				0000：MUX0 0001：MUX1 0010：MUX2 0011：MUX3 0100：MUX4 0101：MUX5 0110：MUX6 0111：MUX7 1000：MUX8 1001：MUX9 1010：MUX10 1011：MUX11 1100：MUX12 1101：MUX13 1110：MUX14 1111：MUX15

6.3.11 GPIO位清除寄存器（GPIOx_CLR）（x=A..F）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持为默认值。
位 15：0	IOCB	0x0000	wo	清除 GPIOx 的位（GPIOx clear bit） 写'1'的位其对应 ODT 寄存器位会清除，写'0'的位其对应 ODT 寄存器位维持不变，相当于 ODT 寄存器的位操作。 0：对应位不变； 1：对应位清除。

6.3.12 GPIO位翻转寄存器（GPIOx_TOGR）（x=A..F）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持为默认值。
位 15：0	IOTB	0x0000	wo	翻转 GPIOx 位（GPIOx toggle bit） 写'1'的位其对应 ODT 寄存器位会翻转原有的值，写'0'的位其对应 ODT 寄存器位维持不变，相当于 ODT 寄存器的位操作。 0：对应位不变； 1：对应位翻转。

6.3.13 极大电流推动/吸入能力切换控制寄存器（GPIOx_HDRV）（x=A..F）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持为默认值。
位 15：0	HDRV	0x0000	rw	极大电流推动/吸入能力切换控制寄存器 0：无效 1：GPIO 切换为极大电流推动/吸入能力

7 系统配置控制器（SCFG）

7.1 简介

该器件具有一组配置寄存器。系统配置控制器的主要用途如下：

- 管理连接到 GPIO 口的外部中断
- 管理存储器映射方式
- 管理部分 IRTMR GPIO 配置

7.2 SCFG寄存器

下面列出了 SCFG 寄存器映像和复位数值。
必须以字（32 位）的方式操作这些外设寄存器。
表 7-1 SCFG寄存器地址映像和复位值

寄存器简称	基址偏移量	复位值
SCFG_CFG1	0x00	0x0000 000X
SCFG_CFG2	0x04	0x0000 0000
SCFG_EXINTC1	0x08	0x0000 0000
SCFG_EXINTC2	0x0C	0x0000 0000
SCFG_EXINTC3	0x10	0x0000 0000
SCFG_EXINTC4	0x14	0x0000 0000
SCFG_UHDRV	0x2C	0x0000 0000

7.2.1 SCFG配置寄存器1（SCFG_CFG1）

域	简称	复位值		功能
位 31：8	保留	0x00000 00	resd	请保持为复位值。
位 7：6	IR_SRC_SEL	0x0	rw	红外调制包络信号源选择（Infrared modulation envelope signal source selection） 用于选择红外调制包络信号源： 00：TMR10 01：保留 10：保留 11：保留
位 5	IR_POL	0x0	rw	红外输出极性选择（Infrared output polarity selection） 0：红外线发射输出（IR_OUT）不反向 1：红外线发射输出（IR_OUT）反向
位 4：2	保留	0x0	resd	请保持为复位值。
位 1:0	MEM_MAP_SEL	0xX	ro	启动模式状态位 此位仅供读取，显示复位后的启动区域。 X0：从主存存储器启动 01：从系统存储器启动 11：从内置 SRAM 启动

7.2.2 SCFG配置寄存器2（SCFG_CFG2）

域	简称	复位值	类型	功能
位 31：30	I2S_FD	0x0	rw	I^2S 全双工配置位(I2S full duplex) 用于配置任两组 I^2S 组成全双工模式。 若非 I^2S 全双工需求，此位域必须保持为 00，否则结果不可预期。详细的组合规则请参考 13.3.2 章节描述。

				00：SPI/I²S1~3 各自独立操作 01：I²S1 和 I²S3 组成全双工模式 10：I²S2 和 I²S3 组成全双工模式 11：I²S1 和 I²S2 组成全双工模式
位 29：3	保留	0x0000 000	resd	请保持为复位值。
位 2	PVM_LK	0x0	rw	PVM 锁使能位(PVM lock enable) 0：断开 PVM 中断与 TMR1/TMR9/TMR10/11/12/13/14 刹车输入的连接。PVMSEL 和 PVMEN 位可以被软件修改。 1：使能 PVM 中断与 TMR1/TMR9/TMR10/11/12/13/14 刹车输入的连接。PVMSEL 和 PVMEN 位为只读位，不能被软件修改
位 1	保留	0x0	resd	请保持为复位值。
位 0	CPU_LK	0x0	rw	CPU 锁使能位(CPU lock enable) 0：断开 CPU 锁与 TMR1/TMR9/TMR10/11/12/13/14 刹车输入的连接。 1：使能 CPU 锁与 TMR1/TMR9/TMR10/11/12/13/14 刹车输入的连接。

7.2.3 SCFG外部中断配置寄存器1（SCFG_EXINTC1）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持默认值。
位 15：12	EXINT3	0x0	rw	配置 EXINT3 的输入源（configure EXINT3 source） 选择 EXINT3 外部中断的输入源。 0000：GPIOA 引脚 3 0001：GPIOB 引脚 3 0010：GPIOC 引脚 3 0011：GPIOD 引脚 3 0100：GPIOE 引脚 3 其他：保留
位 11：8	EXINT2	0x0	rw	配置 EXINT2 的输入源（configure EXINT2 source） 选择 EXINT2 外部中断的输入源。 0000：GPIOA 引脚 2 0001：GPIOB 引脚 2 0010：GPIOC 引脚 2 0011：GPIOD 引脚 2 0100：GPIOE 引脚 2 0101：GPIOF 引脚 2 其他：保留
位 7：4	EXINT1	0x0	rw	配置 EXINT1 的输入源（configure EXINT1 source） 选择 EXINT1 外部中断的输入源。 0000：GPIOA 引脚 1 0001：GPIOB 引脚 1 0010：GPIOC 引脚 1 0011：GPIOD 引脚 1 0100：GPIOE 引脚 1 0101：GPIOF 引脚 1 其他：保留
位 3：0	EXINT0	0x0	rw	配置 EXINT0 的输入源（configure EXINT0 source） 选择 EXINT0 外部中断的输入源。 0000：GPIOA 引脚 0 0001：GPIOB 引脚 0 0010：GPIOC 引脚 0 0011：GPIOD 引脚 0 0100：GPIOE 引脚 0 0101：GPIOF 引脚 0 其他：保留

7.2.4 SCFG外部中断配置寄存器2（SCFG_EXINTC2）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持默认值。
位 15：12	EXINT7	0x0	rw	配置 EXINT7 的输入源（configure EXINT7 source） 选择 EXINT7 外部中断的输入源。 0000：GPIOA 引脚 7 0001：GPIOB 引脚 7 0010：GPIOC 引脚 7 0011：GPIOD 引脚 7 0100：GPIOE 引脚 7 其他：保留
位 11：8	EXINT6	0x0	rw	配置 EXINT6 的输入源（configure EXINT6 source） 选择 EXINT6 外部中断的输入源。 0000：GPIOA 引脚 6 0001：GPIOB 引脚 6 0010：GPIOC 引脚 6 0011：GPIOD 引脚 6 0100：GPIOE 引脚 6 0101：GPIOF 引脚 6 其他：保留
位 7：4	EXINT5	0x0	rw	配置 EXINT5 的输入源（configure EXINT5 source） 选择 EXINT5 外部中断的输入源。 0000：GPIOA 引脚 5 0001：GPIOB 引脚 5 0010：GPIOC 引脚 5 0011：GPIOD 引脚 5 0100：GPIOE 引脚 5 其他：保留
位 3：0	EXINT4	0x0	rw	配置 EXINT4 的输入源（configure EXINT4 source） 选择 EXINT4 外部中断的输入源。 0000：GPIOA 引脚 4 0001：GPIOB 引脚 4 0010：GPIOC 引脚 4 0011：GPIOD 引脚 4 0100：GPIOE 引脚 4 其他：保留

7.2.5 SCFG外部中断配置寄存器3（SCFG_EXINTC3）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持默认值。
位 15：12	EXINT11	0x0	rw	配置 EXINT11 的输入源（configure EXINT11 source） 选择 EXINT11 外部中断的输入源。 0000：GPIOA 引脚 11 0001：GPIOB 引脚 11 0010：GPIOC 引脚 11 0011：GPIOD 引脚 11 0100：GPIOE 引脚 11 其他：保留
位 11：8	EXINT10	0x0	rw	配置 EXINT10 的输入源（configure EXINT10 source） 选择 EXINT10 外部中断的输入源。 0000：GPIOA 引脚 10 0001：GPIOB 引脚 10 0010：GPIOC 引脚 10 0011：GPIOD 引脚 10 0100：GPIOE 引脚 10 0101：GPIOF 引脚 10 其他：保留
位 7：4	EXINT9	0x0	rw	配置 EXINT9 的输入源（configure EXINT9 source）

				选择 EXINT9 外部中断的输入源。 0000：GPIOA 引脚 9 0001：GPIOB 引脚 9 0010：GPIOC 引脚 9 0011：GPIOD 引脚 9 0100：GPIOE 引脚 9 0101：GPIOF 引脚 9 其他：保留
位 3：0	EXINT8	0x0	rw	配置 EXINT8 的输入源（configure EXINT8 source） 选择 EXINT8 外部中断的输入源。 0000：GPIOA 引脚 8 0001：GPIOB 引脚 8 0010：GPIOC 引脚 8 0011：GPIOD 引脚 8 0100：GPIOE 引脚 8 0101：GPIOF 引脚 8 其他：保留

7.2.6 SCFG外部中断配置寄存器4（SCFG_EXINTC4）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持默认值。
位 15：12	EXINT15	0x0	rw	配置 EXINT15 的输入源（configure EXINT15 source） 选择 EXINT15 外部中断的输入源。 0000：GPIOA 引脚 15 0001：GPIOB 引脚 15 0010：GPIOC 引脚 15 0011：GPIOD 引脚 15 0100：GPIOE 引脚 15 其他：保留
位 11：8	EXINT14	0x0	rw	配置 EXINT14 的输入源（configure EXINT14 source） 选择 EXINT14 外部中断的输入源。 0000：GPIOA 引脚 14 0001：GPIOB 引脚 14 0010：GPIOC 引脚 14 0011：GPIOD 引脚 14 0100：GPIOE 引脚 14 其他：保留
位 7：4	EXINT13	0x0	rw	配置 EXINT13 的输入源（configure EXINT13 source） 选择 EXINT13 外部中断的输入源。 0000：GPIOA 引脚 13 0001：GPIOB 引脚 13 0010：GPIOC 引脚 13 0011：GPIOD 引脚 13 0100：GPIOE 引脚 13 其他：保留
位 3：0	EXINT12	0x0	rw	配置 EXINT12 的输入源（configure EXINT12 source） 选择 EXINT12 外部中断的输入源。 0000：GPIOA 引脚 12 0001：GPIOB 引脚 12 0010：GPIOC 引脚 12 0011：GPIOD 引脚 12 0100：GPIOE 引脚 12 其他：保留

7.2.7 SCFG超高电流推动/吸入能力（SCFG_UHDRV）

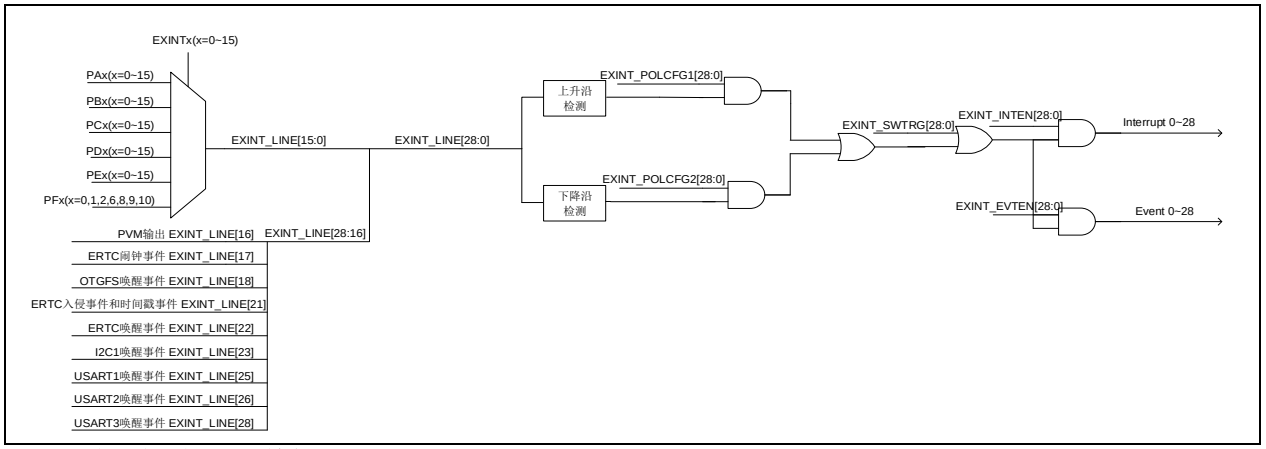
域	简称	复位值	类型	功能
位 31：7	保留	0x0000 000	resd	保持默认值。
位 6	PD13_UH	0x0	rw	PD13 超高电流推动/吸入能力（PD13 Ultra high sourcing/sinking strength） 这些位可由软件读写，用于控制 PD13 PAD 电流推动/吸入能力。 0：无效 1：对应的 GPIO 切换为超高电流推动/吸入能力 当该位使能时，所对应的 GPIO 电流推动/吸入能力（GPIOx_ODRVR&GPIOx_HDRV）控制 bit 无效
位 5	PD12_UH	0x0	rw	PD12 超高电流推动/吸入能力（PD12 Ultra high sourcing/sinking strength） 这些位可由软件读写，用于控制 PD12 PAD 电流推动/吸入能力。 0：无效 1：对应的 GPIO 切换为超高电流推动/吸入能力 当该位使能时，所对应的 GPIO 电流推动/吸入能力（GPIOx_ODRVR&GPIOx_HDRV）控制 bit 无效
位 4	保留	0x0	resd	保持默认值。
位 3	PB8_UH	0x0	rw	PB8 超高电流推动/吸入能力（PB8 Ultra high sourcing/sinking strength） 这些位可由软件读写，用于控制 PB8 PAD 电流推动/吸入能力。 0：无效 1：对应的 GPIO 切换为超高电流推动/吸入能力 当该位使能时，所对应的 GPIO 电流推动/吸入能力（GPIOx_ODRVR&GPIOx_HDRV）控制 bit 无效
位 2	保留	0x0	resd	保持默认值。
位 1	PB9_UH	0x0	rw	PB9 超高电流推动/吸入能力（PB9 Ultra high sourcing/sinking strength） 这些位可由软件读写，用于控制 PB9 PAD 电流推动/吸入能力。 0：无效 1：对应的 GPIO 切换为超高电流推动/吸入能力 当该位使能时，所对应的 GPIO 电流推动/吸入能力（GPIOx_ODRVR&GPIOx_HDRV）控制 bit 无效
位 0	保留	0x0	resd	保持默认值。

8 外部中断/事件控制器（EXINT）

8.1 EXINT介绍

EXINT 共计有 25 条中断线 EXINT_LINE[28:0] (其中位 19、20、24、27 为保留位)，每条中断线均支持通过边沿检测触发和软件触发来产生中断或事件。EXINT 可以根据软件配置，独立的使能或禁止中断或事件，并采取不同的边沿检测方式（检测上升沿或检测下降沿或同时检测上升沿和下降沿）以及触发方式（边沿检测触发或软件触发或边沿检测和软件同时触发）响应触发源独立的产生中断或事件。

图 8-1 外部中断/事件控制器框图



EXINT 控制器的主要特性：

- 中断线 0~15 所映射的 IO 可以独立的配置
- 每个中断线都有独立的触发方式选择
- 每个中断都有独立的使能位
- 每个事件都有独立的使能位
- 共 25 个可独立产生和清除的软件触发
- 每个中断都有独立的状态位
- 每个中断都可以被独立的清除

8.2 功能描述和配置流程

EXINT 共计有 25 条中断线 EXINT_LINE[28:0] (其中位 19、位 20、位 24、位 27 为保留位)，可以通过边沿检测的方式分别检测来自 GPIO 的外部中断源以及包括 PVM 输出，ERTC 闹钟事件，ERTC 入侵事件和时间戳事件，ERTC 唤醒事件，OTGFS 唤醒事件，USART1/USART2/USART3 唤醒事件以及 I2C1 唤醒事件共九种芯片内部的中断源，其中来自 GPIO 的中断源可以通过软件编程配置 SCFG 中的复用外部中断配置寄存器 x（SCFG_EXINTCx）灵活的选择，需要注意的是这些输入源是互斥的，例如 EXINT_LINE0 只能选择 PA0/PB0/PC0/PD0…中的某一个，而不能同时选择 PA0 和 PB0 作为输入源。

EXINT 支持多种边沿检测方式，每条中断线可以通过软件编程配置极性配置寄存器 1（EXINT_POLCFG1）和极性配置寄存器 2（EXINT_POLCFG2）独立的选择上升沿检测或下降沿检测或同时进行上升沿和下降沿检测，中断线上检测到的有效边沿触发可以用于产生事件或中断。

EXINT 支持独立的软件触发产生中断或事件，即除了来自中断线上的有效边沿外，用户可以通过软件编程配置软件触发寄存器（EXINT_SWTRG）对应位来产生对应的中断或事件。

EXINT 具备独立的中断和事件使能位，用户可以通过软件编程配置中断使能寄存器（EXINT_INTEN）和事件使能寄存器（EXINT_EVTEN）来使能或关闭对应的中断或事件，这意味着无论是通过边沿检测还是软件触发产生中断或事件，都需要提前使能对应的中断或事件。

EXINT 具备独立的中断状态位，用户可以通过中断状态寄存器（EXINT_INTSTS）读取对应的中断状态并通过对该寄存器相应位写 1 来清除已置位的状态标志。

中断初始化流程

1. **选择中断源**

   即配置复用外部中断配置寄存器 x（SCFG_EXINTCx）(如果需要使用 GPIO 作为中断源需要该步骤)。

2. **选择触发方式**
   即配置极性配置寄存器 1（EXINT_POLCFG1）和极性配置寄存器 2（EXINT_POLCFG2）。
3. **使能中断或事件**
   即配置中断使能寄存器（EXINT_INTEN）和事件使能寄存器（EXINT_EVTEN）。
4. **产生软件触发**
   即配置软件触发寄存器（EXINT_SWTRG）产生软件触发（此步骤仅适用于需软件触发产生中断的应用）。

注意：若需要更改中断源配置，应先关闭中断使能寄存器和事件使能寄存器后，再重新开始中断初始化流程的配置。

中断清除流程

- 清除标志，即对中断状态寄存器（EXINT_INTSTS）对应位写 1 来清除已产生的中断，同时该操作会同步清除软件触发寄存器（EXINT_SWTRG）中的对应位。

8.3 EXINT寄存器描述

下表列出了 EXINT 寄存器的映像和复位值。
必须以字（32 位）的方式操作这些外设寄存器。
表 8-1 外部中断/事件控制器寄存器映像和复位值

寄存器简称	基址偏移量	复位值
EXINT_INTEN	0x00	0x0000 0000
EXINT_EVTEN	0x04	0x0000 0000
EXINT_ POLCFG1	0x08	0x0000 0000
EXINT_ POLCFG2	0x0C	0x0000 0000
EXINT_ SWTRG	0x10	0x0000 0000
EXINT_ INTSTS	0x14	0x0000 0000

8.3.1 中断使能寄存器（EXINT_INTEN）

域	简称	复位值	类型	功能
位 31：29	保留	0x0	resd	硬件强制为 0。
位 28：0	INTENx	0x00000	rw	线 x 上的中断使能/禁止位 （Interrupt enable or disable on line x） 0：禁止中断请求； 1：使能中断请求。 注：位 19、20、24、27 为保留位，未使用。

8.3.2 事件使能寄存器（EXINT_EVTEN）

域	简称	复位值	类型	功能
位 31：29	保留	0x0	resd	硬件强制为 0。
位 28：0	EVTENx	0x00000	rw	线 x 上的事件使能/禁止位（Event enable or disable on line x） 0：禁止事件请求； 1：使能事件请求。 注：位 19、20、24、27 为保留位，未使用。

8.3.3 极性配置寄存器1（EXINT_POLCFG1）

域	简称	复位值	类型	功能
位 31：29	保留	0x0	resd	硬件强制为 0。
位 28：0	RPx	0x00000	rw	线 x 上的上升沿触发事件配置位 （Rising polarity configuration bit of line x） 这些位用于选择线 x 由上升沿触发中断和事件。 0：禁止上升沿触发； 1：使能上升沿触发。

				注：位 19、20、24、27 为保留位，未使用。

8.3.4 极性配置寄存器2（EXINT_POLCFG2）

域	简称	复位值	类型	功能
位 31：29	保留	0x0	resd	硬件强制为 0。
位 28：0	FRx	0x00000	rw	线 x 上的下降沿触发事件配置位 （Falling polarity event configuration bit of line x） 这些位用于选择线 x 由下降沿触发中断和事件。 0：禁止下降沿触发； 1：允许下降沿触发。 注：位 19、20、24、27 为保留位，未使用。

8.3.5 软件触发寄存器（EXINT_SWTRG）

域	简称	复位值	类型	功能
位 31：29	保留	0x0	resd	硬件强制为 0。
位 28：0	SWTx	0x00000	rw	软件触发线 x （Software triggle on line x） 当 EXINT_INTEN 寄存器中的对应位为 1，则软件写此位硬件将自动置起 EXINT_INTSTS 寄存器中的对应位并产生中断。 当 EXINT_EVTEN 寄存器中的对应位为 1，则软件写此位硬件将自动产生对应中断线上的的事件。 0：默认值； 1：产生软件触发。 注：通过清除 EXINT_INTSTS 的对应位（写入 1），可以清除该位为 0。 注：位 19、20、24、27 为保留位，未使用。

8.3.6 中断状态寄存器（EXINT_INTSTS）

域	简称	复位值	类型	功能
位 31：29	保留	0x0	resd	硬件强制为 0。
位 28：0	LINEx	0x00000	rw1c	线 x 状态位 （Line x state bit） 0：没有发生中断； 1：发生了中断。 注：在该位中写入’1’可以清除它。 注：位 19、20、24、27 为保留位，未使用。

9 DMA 控制器（DMA）

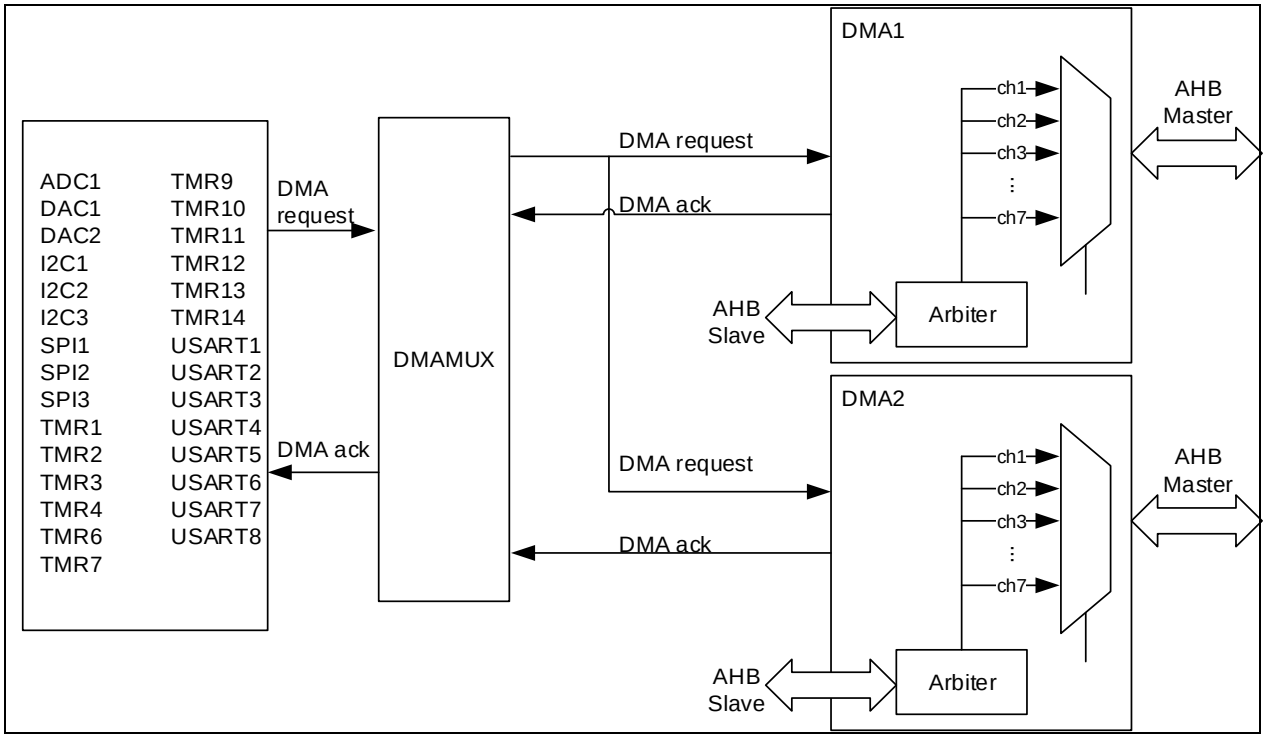
9.1 简介

直接存储器存取（DMA）用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输。无须CPU 干预，数据可以通过 DMA 快速地移动，这就节省了 CPU 的资源来做其他操作。
一个处理器包含 2 个 DMA 控制器。每个控制器各有 7 个 DMA 通道，管理来自于外设对存储器访问的请求，并由仲裁器来协调各个 DMA 请求的优先权。

9.2 特性

- 符合 AMBA 规范（Rev. 2.0）
- 仅支持 AHB OKAY 和 ERROR 响应
- 不支持 AHB 主接口的 HBUSREQ 和 HGRANT
- 支持 7 个通道
- 支持外设到存储器，存储器到外设和存储器到存储器的传输
- 支持硬件握手
- 支持 8 位，16 位和 32 位数据宽度传输
- 传输数据长度最大为 65535，可由编程配置
- 支持多路复用器

图 9-1 DMA框图



注意： 根据不同型号，图中 DMA 外设可能会有所减少。

9.3 功能描述

9.3.1 通道配置

1. 设置外设地址（DMA_CPBAx寄存器）

数据传输的初始外设地址，在传输过程中不会被改变。

2. 设置存储器地址（DMA_CMBAx寄存器）

数据传输的初始存储器地址，在传输过程中不会被改变。

3. 配置数据传输量（DMA_DTCNTx寄存器）

可编程的传输数据长度最大为65535。在传输过程中，该传输数据量的值会逐渐递减。

4. 配置通道设定（DMA_CHCTRLx寄存器）

包含通道优先级，数据传输的方向、宽度，地址增量模式、循环模式和中断方式。

通道优先级 (CHPL)

分为 4 个等级，最高优先级、高优先级、中等优先级和低优先级。

若有 2 个通道优先级设定相同，则较低编号的通道有较高的优先权。举例，通道 1 优先于通道 2。

数据传输方向（DTD）

分为存储器到外设（M2P），外设到存储器（P2M）。

地址增量模式 (PINCM/MINCM)

当设置为增量模式时，下一笔传输的地址将是前一笔传输地址加上传输宽度（PWIDTH/MWIDTH）。

循环模式 (LM)

当通道配置设定为循环模式时，在最后一次传输后 DMA_DTCNTx 寄存器的内容会恢复成初始值。

存储器到存储器模式 (M2M)

存储器到存储器模式是 DMA 在没有外设请求的情况下进行数据传输。

循环模式与存储器到存储器模式不能同时使用。

5. 使能该通道的DMA传输（DMA_CHCTRLx寄存器的CHEN位）

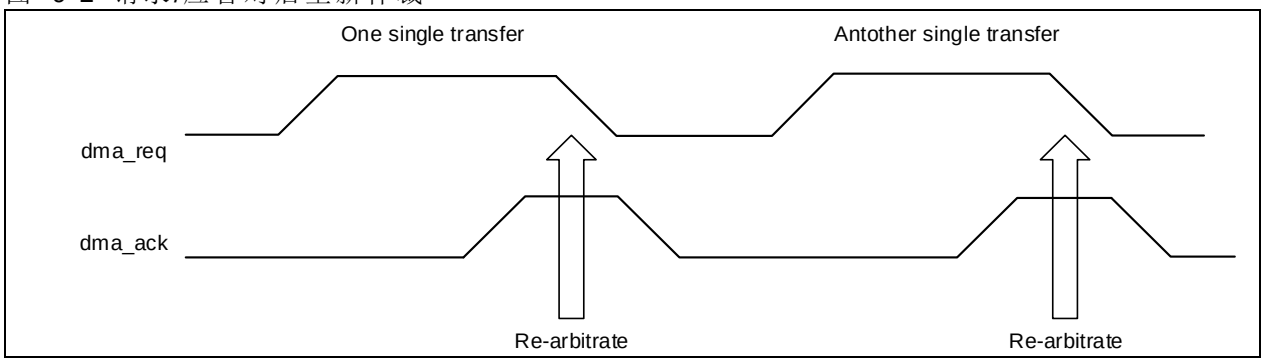
9.3.2 握手机制

在 P2M 和 M2P 传输模式，外设需要向 DMA 控制器发送请求信号。该通道将发出外设传输（单次），直到请求信号被应答为止。外设传输完成后，DMA 控制器将应答信号发送到外设。外设从 DMA 控制器获得应答信号后立即释放其请求。一旦外设取消了请求，DMA 控制器将释放应答信号。

9.3.3 仲裁

当同时启用多个通道时，仲裁器将在主控制器完全传输数据后重新进行仲裁。优先级最高的通道将在先前的通道完成占用主控制器之后，具有主控制器优先级。每当通道以外设主控制器的优先级完成一个单次传输后，外设主控制器就会重新仲裁以服务其他通道。

图 9-2 请求/应答对后重新仲裁



9.3.4 可编程数据传输宽度

通过 DMA_CHCTRLx 中的 PWIDTH 和 MWIDTH 位可以对源数据和目标数据的数据宽度进行编程，当 PWIDTH 不等于 MWIDTH 时，会依据 PWIDTH/ MWIDTH 设定将资料对齐。

图 9-3 PWIDTH： byte, MWIDTH： half-word

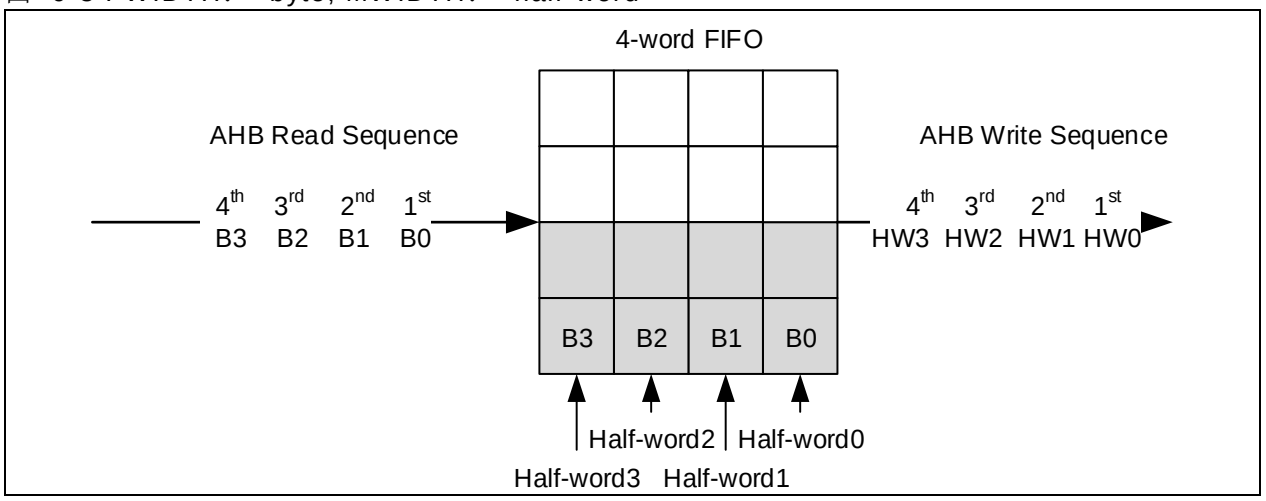


图 9-4 PWIDTH： half-word, MWIDTH： word

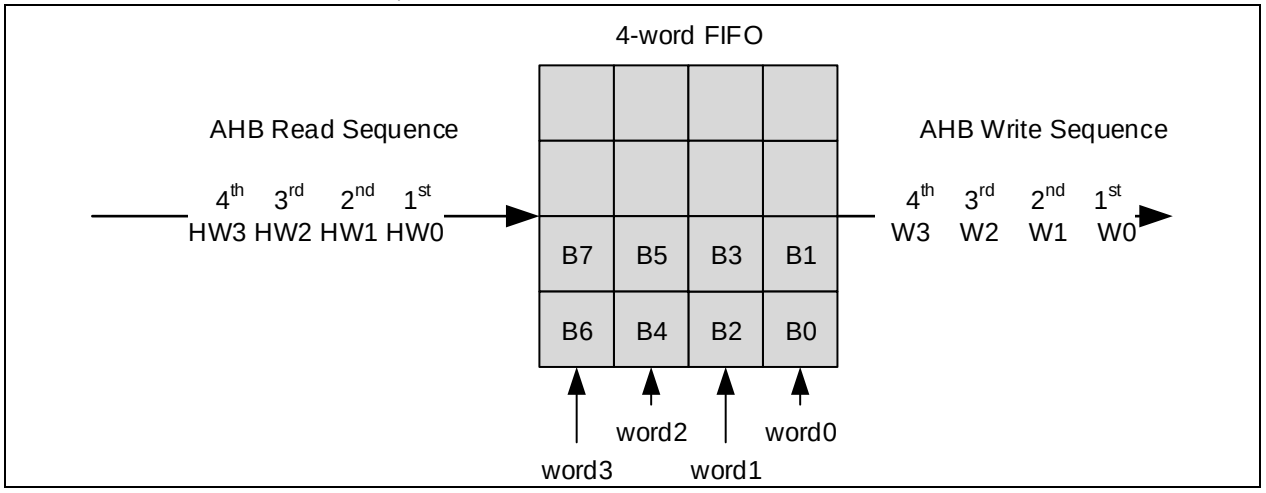
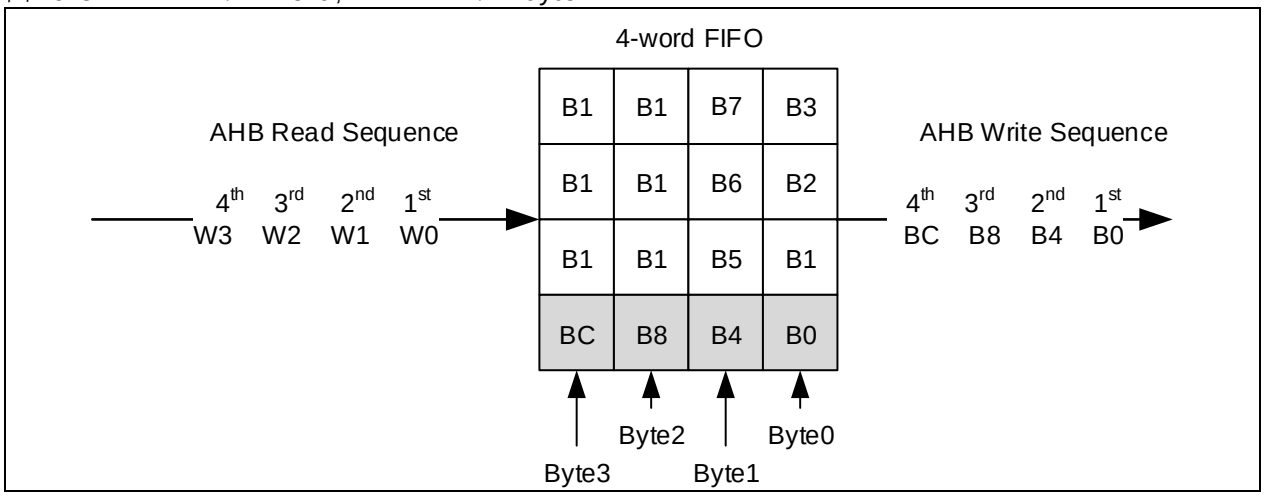


图 9-5 PWIDTH： word, MWIDTH： byte



9.3.5 错误事件

表 9-1 DMA错误事件

错误事件	
传输错误	DMA 读/写访问期间发生 AHB 响应错误

9.3.6 中断

DMA 可在传输过半、传输完成和传输错误时产生中断。每个通道的中断都有专用标志，清除和使能位如下表所示。

表 9-2 DMA中断

中断事件	事件标志位	清除控制位	使能控制位
半传输	HDTF	HDTFC	HDTIEN
传输完成	FDTF	FDTFC	FDTIEN
传输错误	DTERRF	DTERRFC	DTERRIEN

9.4 多路复用器（DMAMUX）

DMAMUX 在外设和 DMA 控制器之间路由 DMA 请求/确认。

DMA 控制器通过 DMA_MUXSEL 中的 TBL_SEL 位选择 DMA 映射表。每个 DMA 控制器流可从弹性映射表中选择一个唯一的 DMA 请求。在弹性映射中，每个通道可以通过 DMA_MUXCxCTRL 中的 REQSEL [6：0]字段旁路或同步来自外设或生成器的 127 个可能的通道请求。

9.4.1 DMAMUX功能描述

DMAMUX 具有两个功能：请求生成器和请求多路复用器。

每个 DMAMUX 生成器通道 x 在 DMA_MUXGxCTRL 中都有一个使能 GEN 位。通过 SIGSEL 字段选择 DMAMUX 生成器的触发输入，通常 DMA 请求的数量为 GREQCNT + 1。通过 DMA_MUXGxCTRL 中的 GPOL 字段选择触发事件，该事件可以是上升沿，下降沿或任一沿。

每个 DMAMUX 多路复用器流 x 来自弹性映射 all_req [127：1] 请求组输入。

在弹性映射中，可以通过 DMA_MUXSxCTRL 中的 SYNCEN 位来同步选定的 DMA 请求输入。当通道处于同步模式时， 通过 DMA_MUXSxCTRL 中的 SYNCSEL 字段选择同步输入， 然后一旦通过 DMA_MUXSxCTRL 中的 SYNCPOL [1：0]字段检测到同步输入的有效沿，则所选的 DMA 请求输入将传播到多路复用器请求输出 chx_mux_req [7：0]。另外，当通过 DMA_MUXCxCTRL 中的 EGE 位启用流的事件生成时，可编程请求计数器 REQCNT 用于请求输出生成和事件生成。

图 9-6 DMAMUX 框图

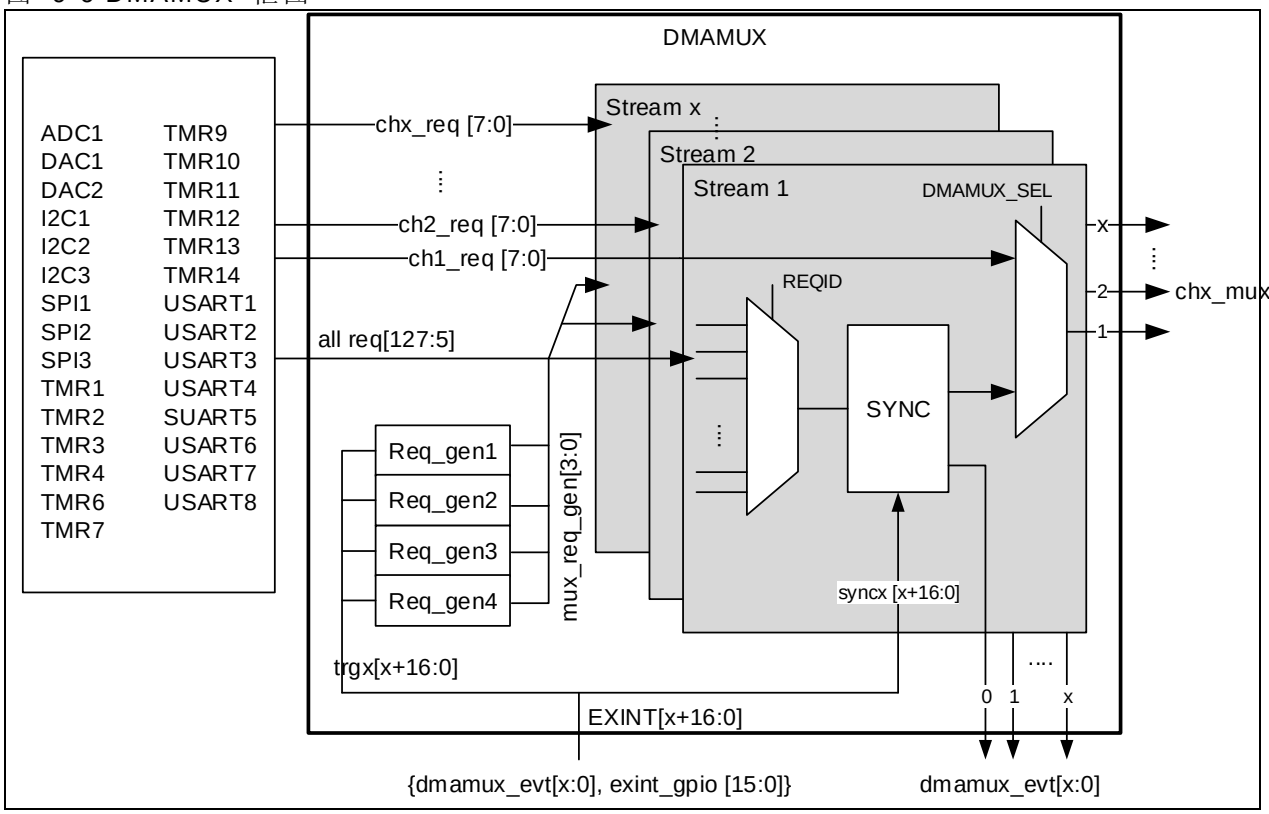


表 9-3 DMA1 / DMA2请求弹性映射

CHx_SRC	请求来源	CHx_SRC	请求来源	CHx_SRC	请求来源	CHx_SRC	请求来源
1	DMA_MUXREQG1	33	USART5_TX	65	TMR3_OVERFLOW	97	TMR12_TRIG
2	DMA_MUXREQG2	34	reserved	66	TMR3_TRIG	98	TMR12_HALL
3	DMA_MUXREQG3	35	reserved	67	TMR4_CH1	99	reserved
4	DMA_MUXREQG4	36	reserved	68	TMR4_CH2	100	reserved
5	ADC1	37	reserved	69	TMR4_CH3	101	reserved
6	DAC1	38	reserved	70	TMR4_CH4	102	reserved
7	reserved	39	reserved	71	TMR4_OVERFLOW	103	reserved
8	TMR6_OVERFLOW	40	reserved	72	reserved	104	reserved
9	TMR7_OVERFLOW	41	DAC2	73	reserved	105	reserved
10	SPI1_RX	42	TMR1_CH1	74	reserved	106	reserved
11	SPI1_TX	43	TMR1_CH2	75	reserved	107	reserved
12	SPI2_RX	44	TMR1_CH3	76	reserved	108	reserved
13	SPI2_TX	45	TMR1_CH4	77	reserved	109	reserved
14	SPI3_RX	46	TMR1_OVERFLOW	78	TMR9_CH1	110	reserved
15	SPI3_TX	47	TMR1_TRIG	79	TMR9_OVERFLOW	111	reserved
16	I2C1_RX	48	TMR1_HALL	80	TMR9_TRIG	112	reserved
17	I2C1_TX	49	reserved	81	TMR9_HALL	113	reserved
18	I2C2_RX	50	reserved	82	TMR10_CH1	114	USART6_RX
19	I2C2_TX	51	reserved	83	TMR10_OVERFLOW	115	USART6_TX
20	I2C3_RX	52	reserved	84	TMR11_CH1	116	USART7_RX
21	I2C3_TX	53	reserved	85	TMR11_OVERFLOW	117	USART7_TX
22	reserved	54	reserved	86	reserved	118	USART8_RX
23	reserved	55	reserved	87	reserved	119	USART8_TX
24	USART1_RX	56	TMR2_CH1	88	reserved	120	TMR13_CH1
25	USART1_TX	57	TMR2_CH2	89	reserved	121	TMR13_OVERFLOW
26	USART2_RX	58	TMR2_CH3	90	reserved	122	TMR14_CH1
27	USART2_TX	59	TMR2_CH4	91	reserved	123	TMR14_OVERFLOW
28	USART3_RX	60	TMR2_OVERFLOW	92	reserved	124	TMR9_CH2
29	USART3_TX	61	TMR3_CH1	93	reserved	125	TMR12_CH2
30	USART4_RX	62	TMR3_CH2	94	reserved	126	TMR2_TRIG
31	USART4_TX	63	TMR3_CH3	95	TMR12_CH1	127	TMR4_TRIG
32	USART5_RX	64	TMR3_CH4	96	TMR12_OVERFLOW		

表 9-4 DMAMUX EXINT LINE用于触发输入和同步输入

EXINT LINE	来源	EXINT LINE	来源	EXINT LINE	来源	EXINT LINE	来源
0	exint_gpio[0]	8	exint_gpio[8]	16	DMA_MUXevt1	24	reserved
1	exint_gpio[1]	9	exint_gpio[9]	17	DMA_MUXevt2	25	reserved
2	exint_gpio[2]	10	exint_gpio[10]	18	DMA_MUXevt3	26	reserved
3	exint_gpio[3]	11	exint_gpio[11]	19	DMA_MUXevt4	27	reserved
4	exint_gpio[4]	12	exint_gpio[12]	20	DMA_MUXevt5	28	reserved
5	exint_gpio[5]	13	exint_gpio[13]	21	DMA_MUXevt6	39	reserved
6	exint_gpio[6]	14	exint_gpio[14]	22	DMA_MUXevt7	30	reserved
7	exint_gpio[7]	15	exint_gpio[15]	23	reserved	31	reserved

9.4.2 DMAMUX 溢出中断

在 DMAMUX 请求生成中，如果在请求生成器计数器 GREQCNT 下溢之前发生了新的触发输入，则在 DMA_MUXGSTS 中声明请求触发溢出标志位 TRGOVFx。通过将 DMA_MUXGCLR 中的相关位 TRGOVFCx 置 1，可以清除溢出标志 TRGOVFx。

如果在 DMA_MUXGxCTRL 中设置了中断允许位 TRGOVIEN，则 DMAMUX 请求触发溢出标志会产生一个中断。

在 DMAMUX 同步模式下，如果在请求计数器 REQCNT 下溢之前发生了新的同步输入，则在 DMA_MUXSYNCSTS 中声明同步溢出标志位 SYNCOVFx。通过将 DMA_MUXSYNCCLR 中的相关位 SYNCOVFCx 置 1，可以清除溢出标志 SYNCOVFx。

如果在 DMA_MUXSxCTRL 中设置了中断允许位 SYNCOVIEN，则 DMAMUX 同步溢出标志会产生一个中断。

图 9-7 DMAMUX请求同步模式

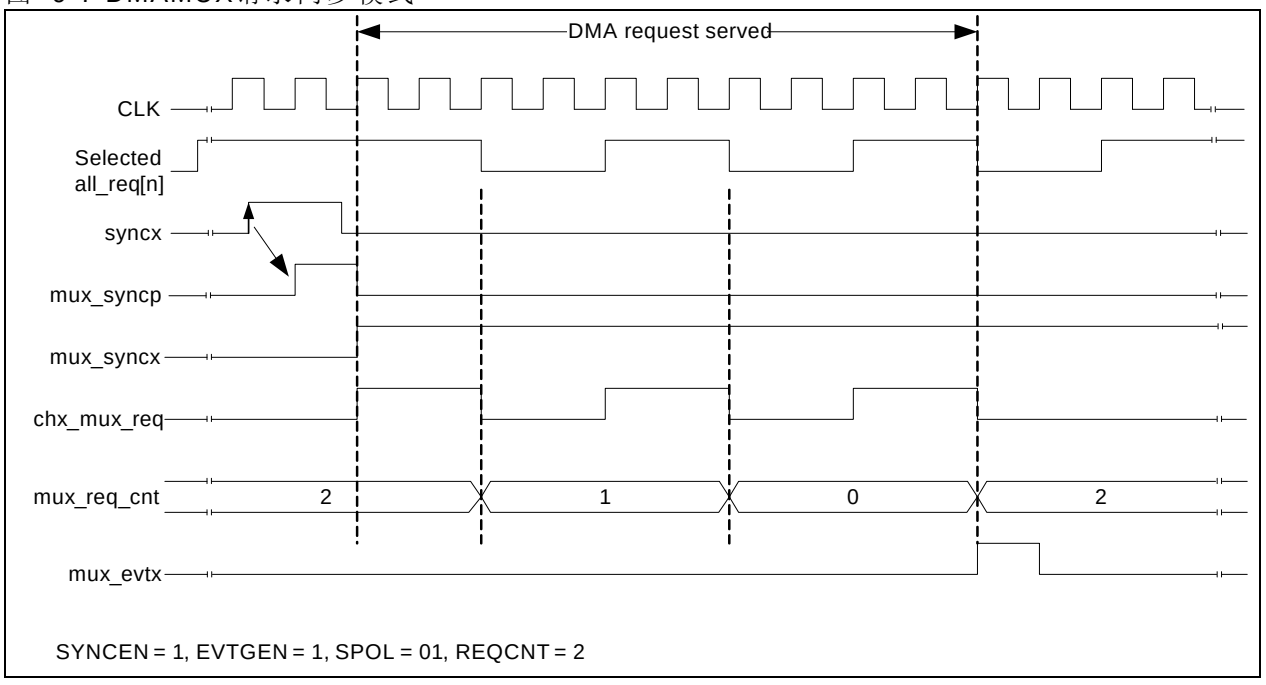
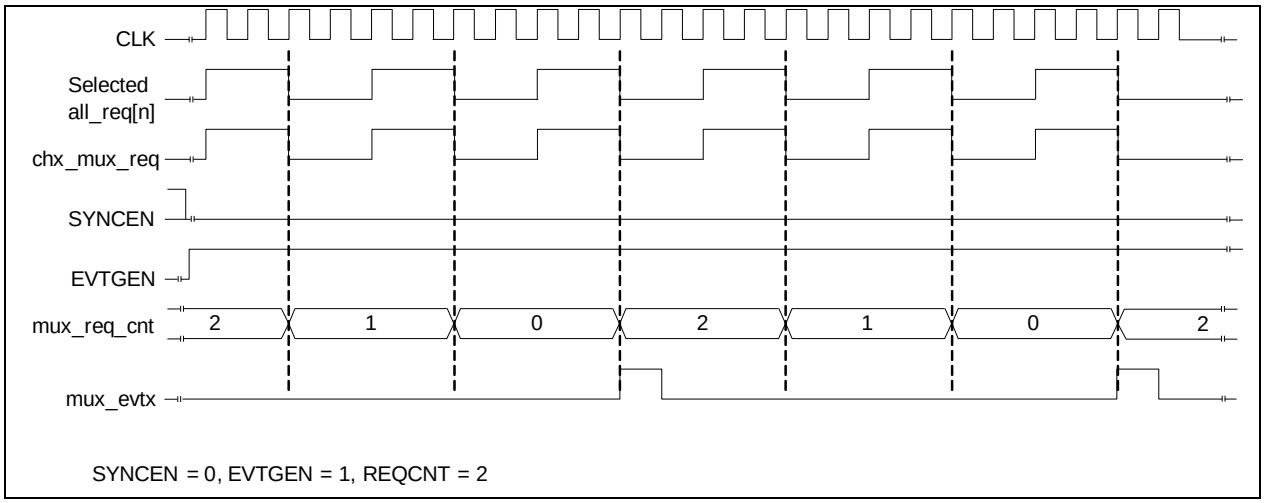


图 9-8 DMAMUX事件生成



9.5 DMA寄存器

下表列出了 DMA 寄存器的映像和复位值。
可以用字节（8 位）、半字（16 位）或字（32 位）的方式操作这些外设寄存器。

表 9-5 DMA寄存器的映像和复位值

寄存器简称	基址偏移量	复位值
DMA_STS	0x00	0x0000 0000
DMA_CLR	0x04	0x0000 0000
DMA_C1CTRL	0x08	0x0000 0000
DMA_C1DTCNT	0x0c	0x0000 0000
DMA_C1PADDR	0x10	0x0000 0000
DMA_C1MADDR	0x14	0x0000 0000
DMA_C2CTRL	0x1c	0x0000 0000
DMA_C2DTCNT	0x20	0x0000 0000
DMA_C2PADDR	0x24	0x0000 0000
DMA_C2MADDR	0x28	0x0000 0000
DMA_C3CTRL	0x30	0x0000 0000
DMA_C3DTCNT	0x34	0x0000 0000
DMA_C3PADDR	0x38	0x0000 0000
DMA_C3MADDR	0x3c	0x0000 0000
DMA_C4CTRL	0x44	0x0000 0000
DMA_C4DTCNT	0x48	0x0000 0000
DMA_C4PADDR	0x4c	0x0000 0000
DMA_C4MADDR	0x50	0x0000 0000
DMA_C5CTRL	0x58	0x0000 0000
DMA_C5DTCNT	0x5c	0x0000 0000
DMA_C5PADDR	0x60	0x0000 0000
DMA_C5MADDR	0x64	0x0000 0000
DMA_C6CTRL	0x6c	0x0000 0000
DMA_C6DTCNT	0x70	0x0000 0000

DMA_C6PADDR	0x74	0x0000 0000
DMA_C6MADDR	0x78	0x0000 0000
DMA_C7CTRL	0x80	0x0000 0000
DMA_C7DTCNT	0x84	0x0000 0000
DMA_C7PADDR	0x88	0x0000 0000
DMA_C7MADDR	0x8c	0x0000 0000
DMA_MUXSEL	0x100	0x0000 0000
DMA_MUXC1CTRL	0x104	0x0000 0000
DMA_MUXC2CTRL	0x108	0x0000 0000
DMA_MUXC3CTRL	0x10c	0x0000 0000
DMA_MUXC4CTRL	0x110	0x0000 0000
DMA_MUXC5CTRL	0x114	0x0000 0000
DMA_MUXC6CTRL	0x118	0x0000 0000
DMA_MUXC7CTRL	0x11c	0x0000 0000
DMA_MUXG1CTRL	0x120	0x0000 0000
DMA_MUXG2CTRL	0x124	0x0000 0000
DMA_MUXG3CTRL	0x128	0x0000 0000
DMA_MUXG4CTRL	0x12c	0x0000 0000
DMA_MUXSYNCSTS	0x130	0x0000 0000
DMA_MUXSYNCCLR	0x134	0x0000 0000
DMA_MUXGSTS	0x138	0x0000 0000
DMA_MUXGCLR	0x13c	0x0000 0000

9.5.1 DMA状态寄存器（DMA_STS）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：28	保留	0x0	resd	保持默认值。
位 27	DTERRF7	0x0	ro	通道 7 数据传输错误事件标志（Data transfer error event flag） 0：未发生错误传输事件 1：发生错误传输事件
位 26	HDTF7	0x0	ro	通道 7 半数据传输事件标志（Half data transfer event flag） 0：未发生半传输事件 1：发生半传输事件
位 25	FDTF7	0x0	ro	通道 7 数据传输完成事件标志（Full data transfer event flag） 0：未发生传输完成事件 1：发生传输完成事件
位 24	GF7	0x0	ro	通道 7 全局事件标志（Global event flag） 0：未发生传输错误、半传输完成或传输完成事件 1：发生传输错误、半传输完成或传输完成事件
位 23	DTERRF6	0x0	ro	通道 6 数据传输错误事件标志（Data transfer error event flag） 0：未发生错误传输事件 1：发生错误传输事件
位 22	HDTF6	0x0	ro	通道 6 半数据传输事件标志（Half data transfer event flag） 0：未发生半传输事件 1：发生半传输事件
位 21	FDTF6	0x0	ro	通道 6 数据传输完成事件标志（Full data transfer event flag） 0：未发生传输完成事件 1：发生传输完成事件
位 20	GF6	0x0	ro	通道 6 全局事件标志（Global event flag） 0：未发生传输错误、半传输完成或传输完成事件 1：发生传输错误、半传输完成或传输完成事件
位 19	DTERRF5	0x0	ro	通道 5 数据传输错误事件标志（Data transfer error event flag） 0：未发生错误传输事件 1：发生错误传输事件
位 18	HDTF5	0x0	ro	通道 5 半数据传输事件标志（Half data transfer event flag） 0：未发生半传输事件 1：发生半传输事件
位 17	FDTF5	0x0	ro	通道 5 数据传输完成事件标志（Full data transfer event flag） 0：未发生传输完成事件 1：发生传输完成事件
位 16	GF5	0x0	ro	通道 5 全局事件标志（Global event flag） 0：未发生传输错误、半传输完成或传输完成事件 1：发生传输错误、半传输完成或传输完成事件
位 15	DTERRF4	0x0	ro	通道 4 数据传输错误事件标志（Data transfer error event flag） 0：未发生错误传输事件 1：发生错误传输事件

位 14	HDTF4	0x0	ro	通道 4 半数据传输事件标志（Half data transfer event flag） 0：未发生半传输事件 1：发生半传输事件
位 13	FDTF4	0x0	ro	通道 4 数据传输完成事件标志（Full data transfer event flag） 0：未发生传输完成事件 1：发生传输完成事件
位 12	GF4	0x0	ro	通道 4 全局事件标志（Global event flag） 0：未发生传输错误、半传输完成或传输完成事件 1：发生传输错误、半传输完成或传输完成事件
位 11	DTERRF3	0x0	ro	通道 3 数据传输错误事件标志（Data transfer error event flag） 0：未发生错误传输事件 1：发生错误传输事件
位 10	HDTF3	0x0	ro	通道 3 半数据传输事件标志（Half data transfer event flag） 0：未发生半传输事件 1：发生半传输事件
位 9	FDTF3	0x0	ro	通道 3 数据传输完成事件标志（Full data transfer event flag） 0：未发生传输完成事件 1：发生传输完成事件
位 8	GF3	0x0	ro	通道 3 全局事件标志（Global event flag） 0：未发生传输错误、半传输完成或传输完成事件 1：发生传输错误、半传输完成或传输完成事件
位 7	DTERRF2	0x0	ro	通道 2 数据传输错误事件标志（Data transfer error event flag） 0：未发生错误传输事件 1：发生错误传输事件
位 6	HDTF2	0x0	ro	通道 2 半数据传输事件标志（Half data transfer event flag） 0：未发生半传输事件 1：发生半传输事件
位 5	FDTF2	0x0	ro	通道 2 数据传输完成事件标志（Full data transfer event flag） 0：未发生传输完成事件 1：发生传输完成事件
位 4	GF2	0x0	ro	通道 2 全局事件标志（Global event flag） 0：未发生传输错误、半传输完成或传输完成事件 1：发生传输错误、半传输完成或传输完成事件
位 3	DTERRF1	0x0	ro	通道 1 数据传输错误事件标志（Data transfer error event flag） 0：未发生错误传输事件 1：发生错误传输事件
位 2	HDTF1	0x0	ro	通道 1 半数据传输事件标志（Half data transfer event flag） 0：未发生半传输事件 1：发生半传输事件
位 1	FDTF1	0x0	ro	通道 1 数据传输完成事件标志（Full data transfer event flag） 0：未发生传输完成事件 1：发生传输完成事件
位 0	GF1	0x0	ro	通道 1 全局事件标志（Global event flag） 0：未发生传输错误、半传输完成或传输完成事件 1：发生传输错误、半传输完成或传输完成事件

9.5.2 DMA标志清除寄存器（DMA_CLR）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：28	保留	0x0	resd	保持默认值。
位 27	DTERRFC7	0x0	rw1c	清除通道 7 的数据传输错误标志（Data transfer error flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF7 标志
位 26	HDTFC7	0x0	rw1c	清除通道 7 的半数据传输标志（Half data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 HDTF7 标志
位 25	FDTFC7	0x0	rw1c	清除通道 7 的数据传输完成标志（Full data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 FDTF7 标志
位 24	GFC7	0x0	rw1c	清除通道 7 的全局中断标志（Global flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF7、HDTF7 FDTF7 和 GF7 标志
位 23	DTERRFC6	0x0	rw1c	清除通道 6 的数据传输错误标志（Data transfer error flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF6 标志
位 22	HDTFC6	0x0	rw1c	清除通道 6 的半数据传输标志（Half data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 HDTF6 标志
位 21	FDTFC6	0x0	rw1c	清除通道 6 的数据传输完成标志（Full data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 FDTF6 标志
位 20	GFC6	0x0	rw1c	清除通道 6 的全局中断标志（Global flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF6、HDTF6 FDTF6 和 GF6 标志
位 19	DTERRFC5	0x0	rw1c	清除通道 5 的数据传输错误标志（Data transfer error flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF5 标志
位 18	HDTFC5	0x0	rw1c	清除通道 5 的半数据传输标志（Half data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 HDTF5 标志
位 17	FDTFC5	0x0	rw1c	清除通道 5 的数据传输完成标志（Full data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 FDTF5 标志
位 16	GFC5	0x0	rw1c	清除通道 5 的全局中断标志（Global flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF5、HDTF5 FDTF5 和 GF5 标志
位 15	DTERRFC4	0x0	rw1c	清除通道 4 的数据传输错误标志（Data transfer error flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF4 标志

位 14	HDTFC4	0x0	rw1c	清除通道 4 的半数据传输标志（Half data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 HDTF4 标志
位 13	FDTFC4	0x0	rw1c	清除通道 4 的数据传输完成标志（Full data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 FDTF4 标志
位 12	GFC4	0x0	rw1c	清除通道 4 的全局中断标志（Global flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF4、HDTF4 FDTF4 和 GF4 标志
位 11	DTERRFC3	0x0	rw1c	清除通道 3 的数据传输错误标志（Data transfer error flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF7 标志
位 10	HDTFC3	0x0	rw1c	清除通道 3 的半数据传输标志（Half data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 HDTF7 标志
位 9	FDTFC3	0x0	rw1c	清除通道 3 的数据传输完成标志（Full data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 FDTF3 标志
位 8	GFC3	0x0	rw1c	清除通道 3 的全局中断标志（Global flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF3、HDTF3 FDTF3 和 GF3 标志
位 7	DTERRFC2	0x0	rw1c	清除通道 2 的数据传输错误标志（Data transfer error flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF2 标志
位 6	HDTFC2	0x0	rw1c	清除通道 2 的半数据传输标志（Half data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 HDTF2 标志
位 5	FDTFC2	0x0	rw1c	清除通道 2 的数据传输完成标志（Full data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 FDTF2 标志
位 4	GFC2	0x0	rw1c	清除通道 2 的全局中断标志（Global flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF2、HDTF2 FDTF2 和 GF2 标志
位 3	DTERRFC1	0x0	rw1c	清除通道 1 的数据传输错误标志（Data transfer error flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF1 标志
位 2	HDTFC1	0x0	rw1c	清除通道 1 的半数据传输标志（Half data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 HDTF1 标志
位 1	FDTFC1	0x0	rw1c	清除通道 1 的数据传输完成标志（Full data transfer flag clear） 0：无效 1：清除 DMA_STS 寄存器中 FDTF1 标志

位 0	GFC1	0x0	rw1c	清除通道 1 的全局中断标志（Global flag clear） 0：无效 1：清除 DMA_ISTS 寄存器中 DTERRF1、HDTF1 FDTF1 和 GF1 标志

9.5.3 DMA通道x配置寄存器（DMA_CxCTRL）（x = 1…7）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：15	保留	0x00000	resd	保持默认值。
位 14	M2M	0x0	rw	存储器到存储器模式（Memory to memory mode） 0：关闭 1：开启
位 13：12	CHPL	0x0	rw	通道优先级（Channel preemptive level） 00：低优先级 01：中优先级 10：高优先级 11：最高优先级
位 11：10	MWIDTH	0x0	rw	存储器数据宽度（Memory data bit width） 00：8 bit 位宽 01：16 bit 位宽 10：32 bit 位宽 11：保留
位 9：8	PWIDTH	0x0	rw	外设数据宽度（Peripheral data bit width） 00：8 bit 位宽 01：16 bit 位宽 10：32 bit 位宽 11：保留
位 7	MINCM	0x0	rw	存储器地址递增模式（Memory address increment mode） 0：关闭 1：开启
位 6	PINCM	0x0	rw	外设地址递增模式（Peripheral address increment mode） 0：关闭 1：开启
位 5	LM	0x0	rw	循环模式（Loop mode） 0：关闭 1：开启
位 4	DTD	0x0	rw	数据传输方向（Data transfer direction） 0：外设为源 1：存储器为源
位 3	DTERRIEN	0x0	rw	允许数据传输错误中断（Data transfer error interrupt enable） 0：禁止数据传输错误中断 1：允许数据传输错误中断
位 2	HDTIEN	0x0	rw	允许半数据传输中断（Half data transfer interrupt enable） 0：禁止半数据传输中断 1：允许半数据传输中断
位 1	FDTIEN	0x0	rw	允许数据传输完成中断（Full data transfer interrupt enable） 0：禁止数据传输完成中断 1：允许数据传输完成中断
位 0	CHEN	0x0	rw	通道使能（Channel enable） 0：关闭 1：开启

9.5.4 DMA通道x数据传输量寄存器（DMA_CxDTCNT）（x = 1…7）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持默认值。
位 15：0	CNT	0x0000	rw	DMA 通道数据传输个数（Number of data to transfer） DMA 通道传输数据个数范围为 0x0~0xFFFF，在更改 DMA 通道传输数据个数时需要确保对应通道的 CHEN 位为 0，否则无法写入；DMA 控制器每传输完一笔数据，此值会硬件减 1。 注：此寄存器为传输数据个数，不是传输数据量大小；传输数据量大小需要根据数据宽度换算得到。

9.5.5 DMA通道x外设地址寄存器（DMA_CxPADDR）（x = 1…7）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：0	PADDR	0x0000 0000	rw	外设端基地址 （Peripheral base address） 外设数据寄存器的基地址，作为数据传输的源或目标。 注：确保对应通道的 CHEN 位为 0，否则无法写入。

9.5.6 DMA通道x存储器地址寄存器（DMA_CxMADDR）（x = 1…7）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：0	MADDR	0x0000 0000	rw	储存器端基地址（Memory base address） 储存器地址作为数据传输的源或目标。 注：确保对应通道的 CHEN 位为 0，否则无法写入。

9.5.7 DMAMUX选择寄存器（DMA_MUXSEL）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
31：1	保留	0x0000 0000	resd	保持默认值。
位 0	TBL_SEL	0x0	rw	多路复用器表选择（multiplexer table select） 0x1：弹性映射表

9.5.8 DMAMUX通道x控制寄存器（DMA_MUXCxCTRL）（x = 1…7）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31:25	保留	0x00	resd	保持默认值。
位 28：24	SYNCSEL	0x00	rw	同步标识（Synchronization select）
位 23：19	REQCNT	0x00	rw	DMA 请求数（Request count） 定义同步事件之后向 DMA 控制器的 DMA 请求数，和/或生成输出事件之前的 DMA 请求数。 仅当 SYNCEN 和 EVTGEN 位均为 LOW 时，才写入字段保留。
位 18：17	SYNCPOL	0x0	rw	同步极性（Synchronization polarity） 定义所选同步输入的边缘极性。 0x0：无事件 0x1：上升沿 0x2：下降沿 0x3：上升沿和下降沿
位 16	SYNCEN	0x0	rw	同步启用（Synchronization enable） 0：禁用同步 1：同步使能
位 15：10	保留	0x00	resd	保持默认值。
位 9	EVTGEN	0x0		启用事件（Event generate enable） 0：禁用事件生成 1：启用事件生成
位 8	SYNCOVIEN	0x0		同步溢出中断启用（Synchronization overrun interrupt enable） 0：禁止中断 1：中断使能
位 7	保留	0x0	resd	保持默认值。
位 6：0	REQSEL	0x00		DMA 请求标识（DMA Request select） 选择输入的 DMA 请求。请参考 DMAMUX 表以了解资源的多路复用器输入的分配。

9.5.9 DMAMUX生成器x控制寄存器（DMA_MUXGxCTRL）（x = 1…4）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：24	保留	0x00	resd	保持默认值。
位 23：19	GREQCNT	0x00	rw	DMA 请求生成数（Request generation count） 定义触发事件后将生成的 DMA 请求（GNBREQ + 1）的数量。 仅当禁用 GEN 位时，才写入字段保留。
位 18：17	GPOL	0x0	rw	DMA 请求生成器触发极性（Generation polarity） 定义所选触发输入的边缘极性。 0x0：无事件 0x1：上升沿 0x2：下降沿 0x3：上升沿和下降沿
位 16	GEN	0x0	rw	DMA 请求生成器（DMA request generation enable） 0：禁用 DMA 请求生成器 1：启用 DMA 请求生成器
位 15：9	保留	0x00	resd	保持默认值。
位 8	TRGOVIEN	0x0	rw	触发溢出中断使能（Trigger overrun interrupt enable） 0：禁止中断 1：中断使能
位 7：5	保留	0x0	resd	保持默认值。
位 4：0	SIGSEL	0x00	rw	信号识别（Signal select） 选择用于 DMA 请求生成器的 DMA 触发输入。

9.5.10 DMAMUX通道同步状态寄存器（DMA_MUXSYNCSTS）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：8	保留	0x0000 00	resd	保持默认值。
位 7：0	SYNCOVF	0x00	ro	同步溢出中断标志（Synchroniztion overrun interrupt flag） 当 DMA 请求计数器低于 REQCNT 时发生，如果发生新的同步事件，将设置该标志。

9.5.11 DMAMUX通道中断清除标志寄存器（DMA_MUXSYNCCLR）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：8	保留	0x0000 00	resd	保持默认值。
位 7：0	SYNCOVFC	0x00	rw1c	清除同步溢出中断标志（Synchronization overrun interrupt flag clear） 在每个位写入 1 以清除 MUXSYNCSTS 寄存器中的相应溢出标志 SYNCOVF。

9.5.12 DMAMUX发生器中断状态寄存器（DMA_MUXGSTS）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31：4	保留	0x0000 000	resd	保持默认值。
位 3：0	TRGOVF	0x00	ro	触发溢出中断标志（Trigger overrun interrupt flag） 当 DMA 请求计数器低于 GREQCNT 时，如果发生新的触发事件，将设置该标志。

9.5.13 DMAMUX发生器中断清除标志寄存器（DMA_MUXGCLR）

访问：无等待状态，字，半字和字节访问

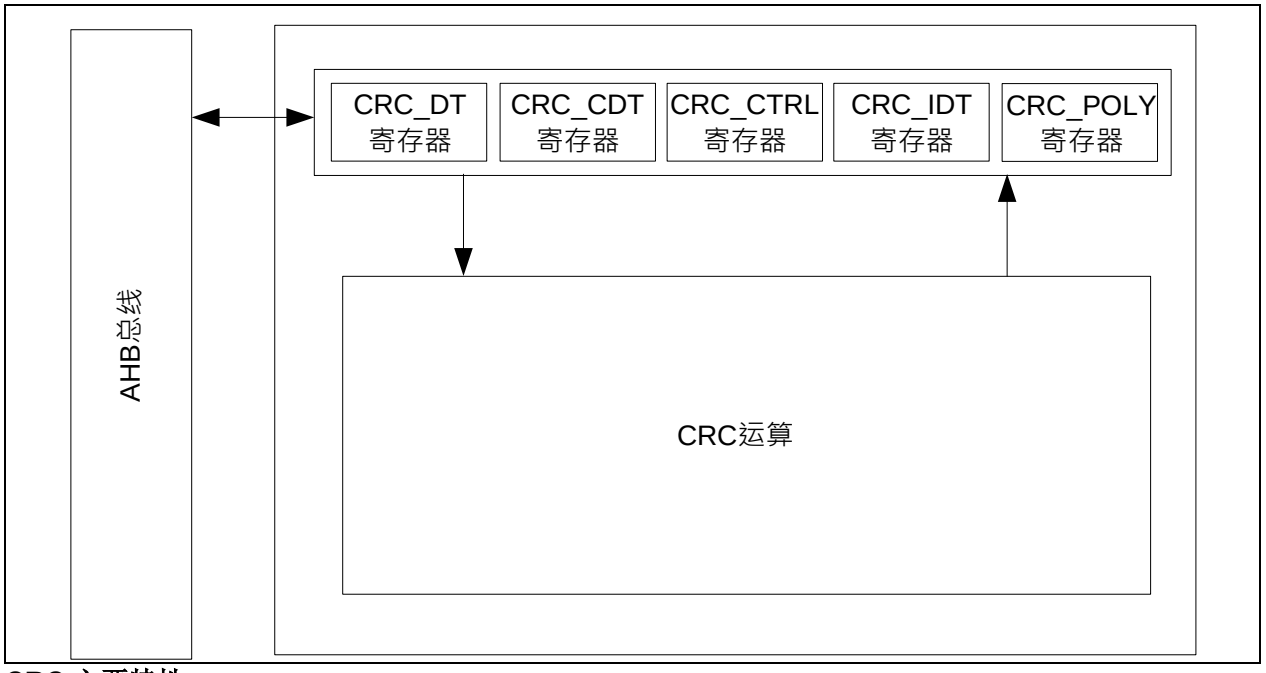
域	简称	复位值	类型	功能
位 31：4	保留	0x0000 000	resd	保持默认值。
位 3：0	TRGOVFC	0x00	rw1c	清除触发器溢出中断标志（Trigger overrun interrupt flag clear） 在每个位写入 1，以清除 DMA_MUXGSTS 寄存器中的相应溢出标志 TRGOVF。

10 CRC 计算单元（CRC）

10.1 CRC介绍

CRC 计算单元是一个独立的具备 CRC 计算功能的外设，CRC 计算单元采用 CRC32/MPEG-2。
用户可以通过软件编程配置 CRC_CTRL 选择是否进行输出数据翻转（全字翻转，REVOD=1）或输入数据翻转(字节翻转，REVID=01；半字翻转，REVID=10；全字翻转，REVID=11)，CRC 计算单元还提供初始化功能，每次 RESET 操作后，CRC 计算单元会将 CRC_IDT 中的值搬入 CRC_DT。CRC_POLY 寄存器可让用户软件编程不同的生成多项式系数，并透过 CRC_CTRL 的 POLY_SIZE 将生成多项式的大小配置为 7/8/16/32 位。
用户通过写和读 CRC_DT 寄存器的方式，写入想要进行计算的值，读出计算的结果，注意每次的 CRC 计算结果是前一次计算结果与当前待计算值的组合。

图 10-1 CRC计算单元框图



CRC 主要特性：

- 预设采用 CRC-32 标准
- 可编程生成多项式
- 一次 CRC 计算需要 4 个 HCLK
- 输入输出数据格式可翻转
- 待计算值的写入和计算结果的读出都通过写和读 CRC_DT 实现
- 配置 CRC_IDT 写入初始化值，在每次 CRC 复位后该值会加载到 CRC_DT

10.2 CRC功能说明

CRC 的计算原理是将输入数据做为被除数，与作为除数的生成多项式进行模二除法，得到的余数即为 CRC 值。

CRC 运算流程

- 输入翻转，即数据输入后，先依据 CRC_CTRL 的 REVID 值进行输入数据翻转
- 初始化，首次计算会与 CRC_IDT 设定的初始值做 XOR。若非首次计算，则初始值为上次的计算结果。
- CRC 计算，与生成多项式(0x4C11DB7)进行模二除法，所得余数为 CRC 值
- 输出翻转，依据 CRC_CTRL 的 REVOD 决定是否将 CRC 值执行全字翻转后再输出。
- 对结果进行 XOR 运算，结果异或值固定为 0x0000 0000。

CRC-32/MPEG-2 参数说明

- 生成多项式: 0x4C11DB7，

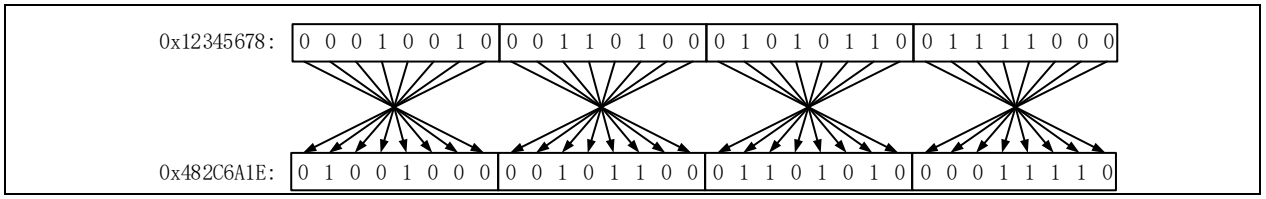
即$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$

- 初始值:0xFFFF FFFF，目的为避免待测数据为 1 字节 0x00 和多字节 0x00 的结果相同。
- 结果异或值:0x0000 0000,此值表示不对 CRC 结果再进行一次 XOR 运算

翻转功能说明

- 选择以字节翻转，则 8bit 为一组，组内排列顺序颠倒。如下图所示，若原数据为 0x12345678，翻转后为 0x482C6A1E。
- 选择以半字翻转，则 16bit 为一组，组内排列顺序颠倒。
- 选择以字翻转，则 32bit 为一组，组内排列顺序颠倒。

图 10-2 字节翻转示意图



10.3 CRC寄存器

除 CRC_DT 可以用字节（8 位）、半字（16 位）或字（32 位）的方式操作之外，其他寄存器必须以字（32 位）的方式操作。

表 10-1 CRC计算单元寄存器映像

寄存器简称	基址偏移量	复位值
CRC_DT	0x00	0xFFFF FFFF
CRC_CDT	0x04	0x0000 0000
CRC_CTRL	0x08	0x0000 0000
CRC_IDT	0x10	0xFFFF FFFF
CRC_POLY	0x14	0x04C1 1DB7

10.3.1 数据寄存器（CRC_DT）

域	简称	复位值	类型	功能
位 31：0	DT	0xFFFF FFFF	rw	数据寄存器位（Data value） 写入 CRC 计算器的新数据时，作为输入寄存器读取时返回 CRC 计算的结果。

10.3.2 通用数据寄存器（CRC_CDT）

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	resd	保持默认值。
位 7：0	CDT	0x00	rw	通用 8 位数据寄存器位（Common 8-bit data value） 可用于临时存放 1 字节的数据。寄存器 CRC_CTRL 的 RST 位产生的 CRC 复位对本寄存器没有影响。

10.3.3 控制寄存器（CRC_CTRL）

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	resd	保持默认值。
位 7	REVOD	0x0	resd	输出数据翻转（Reverse output data） 由软件置起或清零。该位控制是否翻转输出数据。 0：不翻转； 1：全字翻转。
位 6：5	REVID	0x0	rw	输入数据翻转（Reverse input data） 由软件置起或清零。该位控制如何翻转输入数据。 00：不翻转； 01：字节翻转； 10：半字翻转； 11：全字翻转。
位 4：3	POLY_SIZE	0x0	rw	生成多项式位宽(Polynomial size) 该位控制生成多项式的位宽大小，与 CRC_POLY 寄存器相配合。 00：位宽为 32 位 01：位宽为 16 位 10：位宽为 8 位 11：位宽为 7 位
位 2：1	保留	0x0	resd	保持默认值。
位 0	RST	0x0	wo	RESET 位（Reset CRC calculation unit） 由软件置起，由硬件自动清零。复位 CRC 计算单元，设置数据寄存器为 0xFFFF FFFF。 0：无作用； 1：复位。

10.3.4 初始化寄存器（CRC_IDT）

域	简称	复位值	类型	功能
位 31：0	IDT	0xFFFF FFFF	rw	初始化数据寄存器（Initial data value） 当 CRC_CTRL 寄存器的 RST 位产生的 CRC 复位时，初始化寄存器中的数值将作为 CRC_DT 寄存器的初始值写入。

10.3.5 生成多项式系数寄存器（CRC_POLY）

域	简称	复位值	类型	功能
位 31：0	POLY	0x04C1 1DB7	rw	生成多项式系数寄存器（polynomial coefficient） 生成多项式为 CRC 计算中的除数，预设使用 CRC32 参数模型，所以系数为 0x4C11DB7。用户亦可自行编程该生成多项式。

11 I²C 接口

11.1 I²C 简介

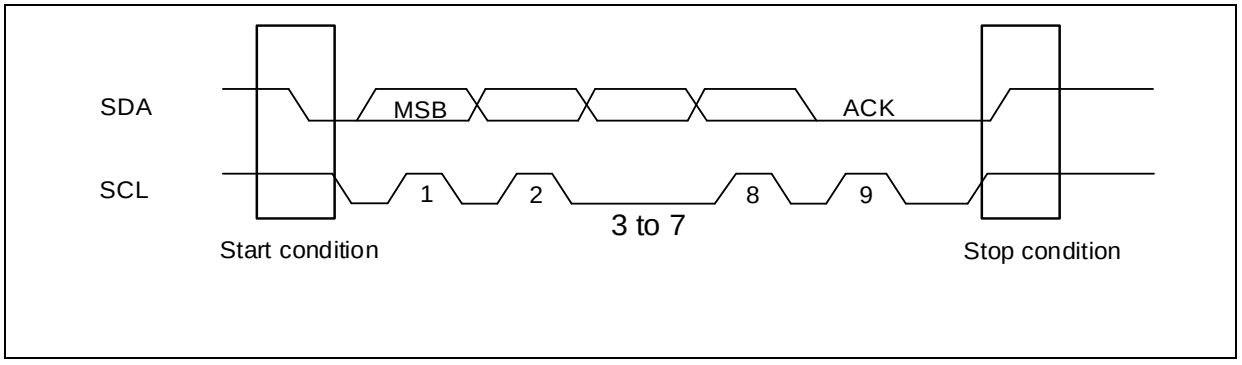
I²C 总线接口处理微控制器和串行 I²C 总线之间的通信，支持主机和从机模式，最大通信速度为 1Mbit/s(增强快速模式)。

11.2 I²C 的主要特点

- I2C 总线
  - 主机和从机模式
  - 多主机功能
  - 标准模式（standard mode，最高 100kHz）、快速模式（fast mode，最高 400kHz）和增强快速模式（fast mode plus，最高 1 MHz）
  - 7-bit 和 10-bit 地址模式
  - 两组 7 位从地址（2 个地址，其中一个可屏蔽）
  - 广播呼叫
  - 可编程数据建立和保持时间
  - 时钟延展功能
- 支持 DMA 功能
- 可编程数字噪声滤波器
- 可用地址匹配事件从 deep sleep mode 唤醒
- 支持 SMBus 2.0 版协议
  - PEC 产生和检查
  - 命令和数据的应答控制
  - ARP(地址解析协议)
  - 主机功能
  - 设备功能
  - SMBus 提醒功能
  - 超时检测
  - 空闲检测
- PMBus

11.3 I²C总线特性

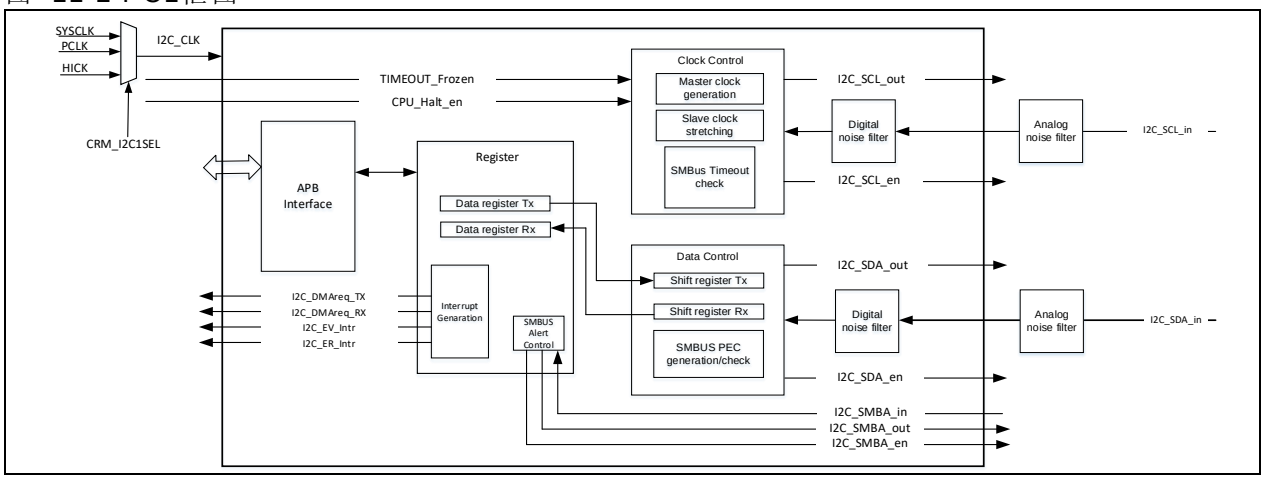
I²C 总线是由数据线 SDA 和时钟线 SCL 构成，在标准模式下通信速度可达到 100kHz，快速模式下则可以达到 400kHz，增强快速模式可达到 1MHz。一帧数据传输从开始信号开始，在结束信号后停止，在收到开始信号后总线被认为是繁忙的，当收到结束信号后，总线被认为再次空闲。
开始信号：SCL 为高电平时，SDA 由高电平变为低电平；
结束信号：SCL 为高电平时，SDA 由低电平变为高电平。
图 11-1 I2C总线协议



11.4 I²C 接口

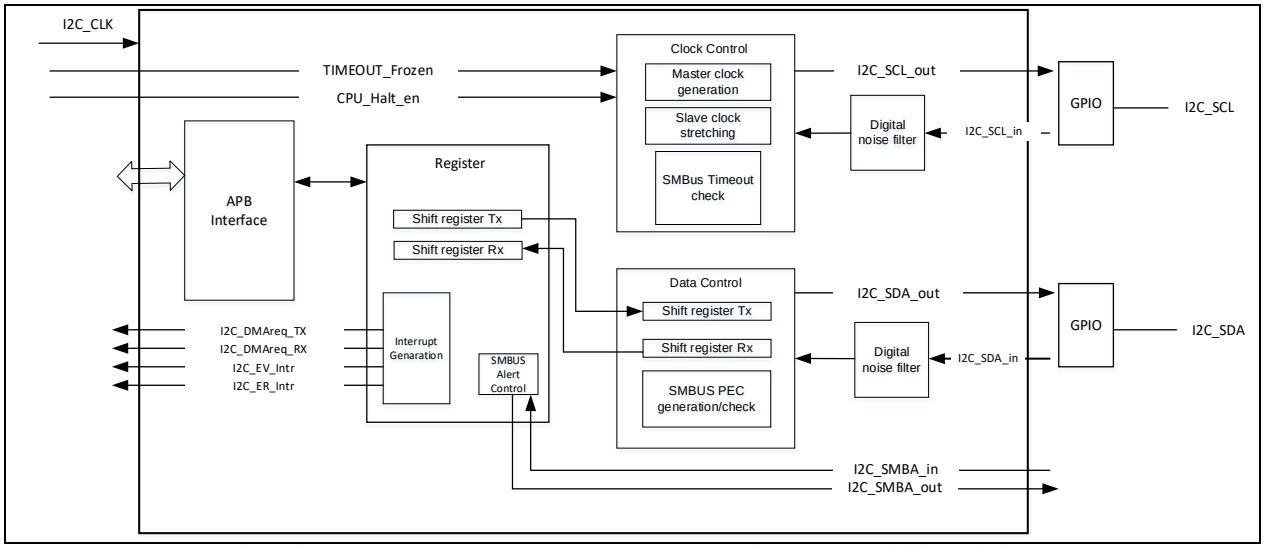
I²C 接口的功能框图示于下图：

图 11-2 I²C1框图



I2C1 可透过配置 CRM 中 PICLKS 寄存器的 I2C1SEL 位，时钟来源可选择来自 SYSCLK、PCLK 和 HICK，并且支持从 Deepsleep mode 唤醒， I2C1 有模拟滤波器，可以滤掉 50ns 内的噪声。

图 11-3 I²C2、I²C3框图



I2C2 、I2C3 时钟来源为 PCLK，不支持 Deepsleep mode 唤醒并且没有模拟滤波器。

1. 接口工作模式

I²C 总线接口可以工作在主机模式与从机模式，并且可以相互切换。默认情况下处于从机模式，当产生了一个起始信号后，I²C 总线接口切换成主模式，当数据传输完成之后，也就是结束信号产生了之后，I²C 总线接口自动返回为从机模式。

2. 通信流程

- 主机模式通信流程：
  1. 产生开始信号
  2. 发送地址
  3. 发送或接收数据
  4. 产生结束信号
  5. 通信结束
- 从机模式通信流程：
  1. 等待地址匹配
  2. 发送或接收数据

3. 等待结束信号产生
4. 通信结束

3. 数字滤波功能

SCL 和 SDA 总线上均有数字滤波器，数字滤波器可以有效降低总线上的干扰，通过配置 I2C_CTRL1 的 DFLT[3：0]位（范围 0~15），可以启用数字滤波器，滤波时间为 DFLT x t_{I2C_CLK}，当 I²C 启用时不允许更改数字滤波器的配置。

4. 地址控制

主机和从机都支持 7 位和 10 位地址模式

从机地址模式：

- 7 位地址模式（ADDR1MODE=0）
  - 单地址模式ADDR1EN=1，ADDR2EN=0：此时只匹配OADDR1；
  - 双地址模式ADDR1EN=1，ADDR2EN=1：此时匹配OADDR1和OADDR2。
- 10 位地址模式（ADDR1MODE=1）
  - 只支持单地址模式ADDR1EN=1，ADDR2EN=0，匹配OADDR1

从机地址屏蔽功能

从机地址 2（OADDR2）支持地址屏蔽功能，通过设置 ADDR2MASK[2：0]启用地址屏蔽功能

- 0：匹配地址位[7：1]；
- 1：只匹配地址位[7：2]；
- 2：只匹配地址位[7：3]；
- 3：只匹配地址位[7：4]；
- 4：只匹配地址位[7：5]；
- 5：只匹配地址位[7：6]；
- 6：只匹配地址位[7]；
- 7：所有非I²C保留地址都会响应。

从机特殊地址支持：

- 广播地址（0b0000000x）：当 GCAEN=1 时该地址启用；
- SMBus 设备默认地址（0b1100001x）：当在 SMBus 设备模式下（DEVADDREN = 1）该地址启用，该地址用于 SMBus 地址解析协议；
- SMBus 主机默认地址（0b0001000x）：当在 SMBus 主机模式下（HADDREN = 1）该地址启用，该地址用于 SMBus 主机通知协议；
- SMBus 提醒地址（0b0001100x）：当在 SMBus 在设备模式下，并且拉低 SMBbus ALERT 引脚（SMBALERT = 1）下该地址启用，该地址用于 SMBus 提醒响应协议。

关于 SMBus 协议更详细的信息请参考 SMBus2.0 协议。

从机地址匹配流程：

- 收到开始信号
- 匹配地址
- 若地址成功匹配，从机回一个 ACK
- 此时 ADDRF 置 1，SDIR 指示传输方向
  - 如果SDIR=0从机进入接收模式，开始接收数据
  - 如果SDIR=1从机进入发送模式，开始发送数据

5. 时钟延展功能

在默认情况下，从机的时钟延展功能是打开的，即 I2C_CTRL1 的 STRETCH 位为 0，从机可以根据需要将 SCL 信号拉低，延长 SCL 信号的低电平时间以便执行软件的操作，如果主机不支持时钟延展功能，则 I2C_CTRL1 的 STRETCH 位需配置为 1。需要注意的是 I²C 从机的时钟延展模式必须在在使能 I²C 外设之前配置。

从机时钟延展

I²C 从机在以下情况下延展 SCL 时钟：

- 地址接收阶段：从机接收到的地址与启用的本机地址匹配（I2C_STS 的 ADDRF=1）时会将 SCL 线拉低，直到软件将 I2C_CLR 的ADDRC 位置 1 清掉 ADDRF 时释放时钟延展；
- 数据接收阶段：I2C_RXDT 寄存器的数据尚未被读取，移位寄存器也接收到一个新的数据，这时会将 SCL 线拉低直到 I2C_RXDT 寄存器的数据被读取；

- 数据发送阶段：ADDRF 被清除后未写入数据时，I2C_STS 的 TDBE= 1，这时会将 SCL 线拉低直到数据写入 I2C_TXDT；
- 数据发送阶段：上一笔的数据传输已完成，尚未有新的数据写到 I2C_TXDT，这时会将 SCL 线拉低直到数据写入 I2C_TXDT；
- 当启用从机字节控制模式(I2C_CTRL1 的 SCTRL 为 1)且 I2C_CTRL2 的 RLDEN 位为 1，如果 TCRLD = 1，代表此时最后一个数据字节已经传输完成，直到向 I2C_CTRL2 的 CNT 位写入一个非零值，硬件自动将 TCRLD 清 0，从机释放 SCL 线。

从机不带时钟延展

当 I2C_CTRL1 的 STRETCH 位为 1 时，I²C 从机不会延展 SCL 信号，需要特别注意下列情况：

- 地址接收阶段：从机接收到的地址与启用的本机地址匹配（I2C_STS 的 ADDRF=1）时 SCL 时钟不会延展；
- 数据接收阶段：下一个字节的 ACK 发生前，尚未将数据从 I2C_RXDT 寄存器读走，会发生上溢，I2C_STS 的 OUF 位会被置 1；
- 数据发送阶段：上一笔的数据传输已完成，尚未有新的数据写到 I2C_TXDT，会发生下溢，I2C_STS 的 OUF 位会被置 1。

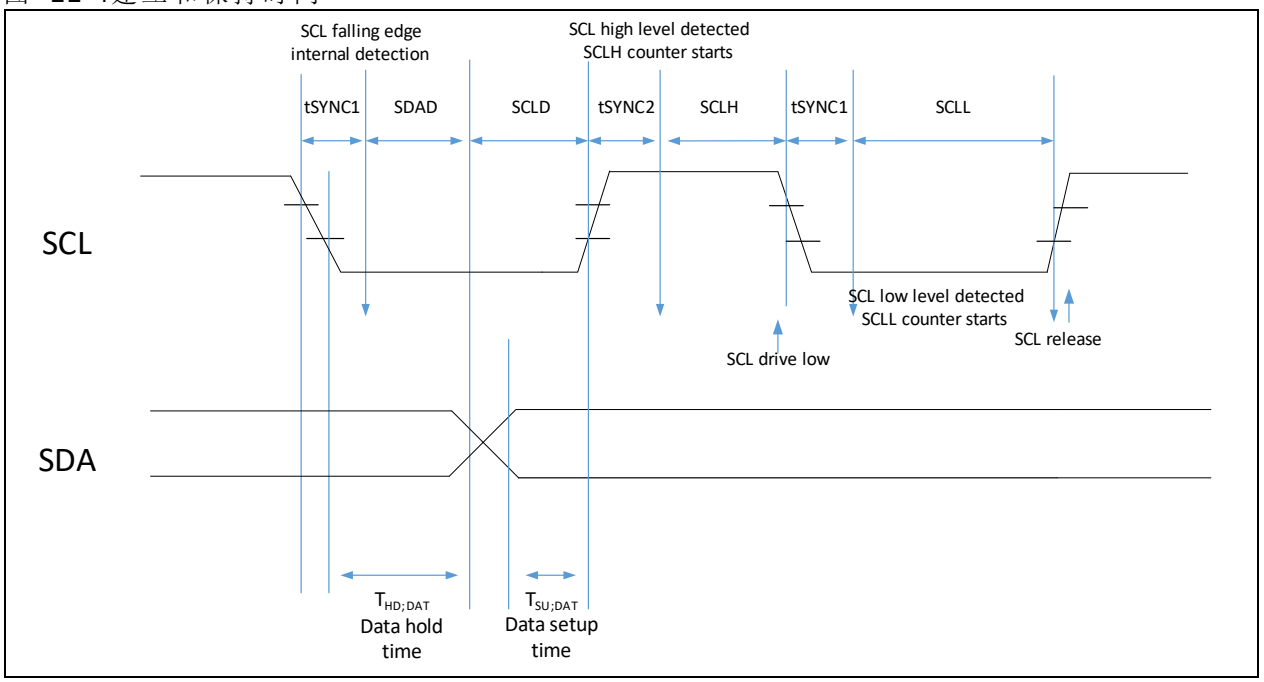
11.4.1 I²C 时序控制

I²C 内核由 I2C_CLK 提供时钟，I2C_CLK 由 PCLK1 提供，PCLK1 周期需满足小于 4/3 SCL 周期。通过 I2C_CLKCTRL 寄存器的各个位，配置各个时序。

— DIV[7：0]：I²C时钟分频；
— SDAD[3：0]：数据保持时间（$t_{HD;DAT}$）；
— SCLD[3：0]：数据建立时间（$t_{SU;DAT}$）；
— SCLH[7：0]：SCL高电平时间；
— SCLL[7：0]：SCL低电平时间。

注：当 I²C 启用时不允许更改时序配置。

图 11-4建立和保持时间



通过配置 I2C_CLKCTRL 的 DIV[7：0]、SDAD[3：0]、SCLD[3：0]，可以灵活的配置数据保持时间（$t_{HD;DAT}$）和数据建立时间（$t_{SU;DAT}$）

- 数据保持时间（$t_{HD;DAT}$）：SCL 下降沿到 SDA 输出的延时

  $t_{HD;DAT} = t_{SDAD} + t_{SYNC}$

  $t_{SDAD} = SDAD \times (DIV + 1) \times t_{I2C\_CLK}$

  $t_{SYNC} = (DFLT + 3) \times t_{I2C\_CLK} - t_f$

  t_{SYNC} 由以下几部分组成

- SCL下降沿时间t_f
- 数字滤波器的输入延迟（DFLT x t_{I2C_CLK}）
- SCL和I2C_CLK的同步延时（2~3个I2C_CLK）

- 数据建立时间（$t_{SU;DAT}$）：SDA 输出后到 SCL 上升沿的延时

  $t_{SU;DAT}$ = SCLD x (DIV+1) x t_{I2C_CLK} – t_r

在主机模式下，通过配置 I2C_CLKCTRL 的 DIV[7：0]、SCLH[7：0]、SCLL[7：0]，可以灵活的配置 SCL 高电平宽度和低电平宽度

低电平控制：当检测到 SCL 总线为低电平时，内部 SCLL 计数器开始计数，当计数值达到 SCLL 值时，释放 SCL 线，SCL 线变为高电平。

高电平控制：当检测到 SCL 总线为高电平时，内部 SCLH 计数器开始计数，当计数值达到 SCLH 值时，拉低 SCL 线，SCL 线变为低电平，当在高电平期间，如果被外部总线拉低，那么内部 SCLH 计数器停止计数，并开始低电平计数，这为时钟同步提供了条件。

- 高电平宽度：

  t_{HIGH} = (SCLH + 1) x (DIV + 1) x t_{I2C_CLK}

- 低电平宽度：

  T_{Low} = (SCLL + 1) x (DIV + 1) x t_{I2C_CLK}

表 11-1 I^2C时间规范

参数		标准模式 Standard mode		快速模式 Fast mode		增强快速模式 Fast mode plus		SMBus 模式	
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
f_{SCL} (kHz)	SCL 时钟频率		100		400		1000		100
t_{LOW} (us)	SCL 时钟低电平	4.7		1.3		0.5		4.7	
t_{HIGH} (us)	SCL 时钟高电平	4.0		0.6		0.26		4.0	50
tHD;DAT (us)	数据保持时间	0		0	0.9	0	0.45	300	
tSU;DAT (ns)	数据建立时间	250		100		50		250	
tr (ns)	SCL、SDA 上升沿		1000		300		120		1000
tf (ns)	SCL、SDA 下降沿		300		300		120		300

11.4.2 数据传输管理

I^2C 内部有一个管理数据传输的计数器，用来管理通信流程，主要应用于以下地方：

- NACK发送：主机接收模式；
- STOP发送：主机接收/发送模式；
- RESTART产生：主机接收/发送模式；
- ACK控制：从机模式（SMBus下）；
- PEC发送/接收：主机/从机模式。

通常字节传输管理计数器（I2C_CTRL2 的 CNT[7:0]位配置）应用于主模式，在从机模式下是关闭的，只有在 SMBus 模式下，从机为了对每一个字节进行 ACK 的控制和 PEC 的接收，才会使用此计数器，在 SMBus 模式下从机可以通过 I2C_CTRL2 的 SCTRL 位来启用字节计数器功能。

主机字节控制

I2C_CTRL2 的 CNT[7:0]用于配置发送字节个数，配置范围为 1~255，当一次传输的数据超过 255 个以上需要将 I2C_CTRL2 的 RLDEN 位置 1，使能重载模式，所以发送数据个数分为≤255 字节和＞255 字节两种情况：

- ≤255字节，例如要传输数据个数为100个字节
  - 步骤 1：配置 RLDEN=0，关闭重载模式；
  - 步骤 2：配置 CNT[7:0]=100；
- ＞255字节，例如要传输数据个数为600个字节

- 步骤 1：配置 RLDEN=1，使能重载模式；
- 步骤 2：配置 CNT[7:0]=255，此时还剩下 600-255=345 字节；
- 步骤 3：当这 255 字节传输完成后，I2C_STS 的 TCRLD 位置 1，然后配置 CNT[7:0]=255，继续传输，此时还剩下 345-255=90；
- 步骤 4：当这 255 字节传输完成后，I2C_STS 的 TCRLD 位置 1，然后配置 RLDEN=0，禁止重载模式，再配置 CNT[7:0]=90，继续传输。

当是最后一笔数据传输时（重载模式禁止，RLDEN=0），有两种结束数据传输模式

- 自动结束模式（I2C_CTRL2的ASTOPEN=1）
  - 当发送了 CNT[7:0]个字节后，主机自动发送 STOP 条件。
- 软件结束模式（I2C_CTRL2的ASTOPEN=0）
  - 当发送了 CNT[7:0]个字节后，I2C_STS 的 TDC 位置 1，此时 SCL 被拉低，如果使能了 TDCIEN，那么可以产生一个中断，此时软件可以设置 I2C_CTRL2 的 GENSTOP=1 产生一个 STOP 条件,也可以设置 I2C_CTRL2 的 GENSTART=1 产生一个 RESTART 条件，然后软件清除 TDC 标志。

从机字节控制

通过 I2C_CTRL2 的 SCTRL 位来启用从机字节控制功能，该功能可以让从机对接收到的每一个字节进行单独的 ACK/NACK 控制。

- 使用步骤：
  - 设置 SCTRL=1，启用从机字节控制功能；
  - 在从机地址匹配后（ADDRF=1），使能重载模式（RLDEN=1），并设置 CNT[7:0]=1；
  - 当接收到一个字节后，I2C_STS 的 TCRLD 置 1，从机在 SCL 的第 8 个和第 9 个时钟沿中间拉低 SCL 总线，此时用户读取 RXDT 寄存器，然后根据需要配置 I2C_CTRL2 的 NACKEN 位，来产生一个 ACK 或 NACK；
  - 如果产生一个 NACK，那么通信结束；
  - 如果产生一个 ACK，通信继续，此时写入 CNT[7:0]=1，硬件自动清除 TCRLD 标志，从机释放 SCL 总线，继续接收下一个字节。

当然 CNT[7:0]值不仅仅局限于 1，例如想接收 8 个数据，但只想控制第 8 个数据的 ACK/NACK，此时就可以设置 CNT[7:0]=8，那么从机就会连续接收 7 个数据，并且回复 ACK，在第 8 个数据接收完后，从机拉低总线，然后同上述步骤，决定 ACK/NACK 的回复。

需要注意的是：要使用从机字节控制模式必须要使能时钟延展（I2C_CTRL1 的 STRETCH 位 = 0）。

表 11-2 I²C配置表

功能	RLDEN	ASTOPEN	SCTRL
主机发送/接收 RESTART	0	0	×
主机发送/接收 STOP	0	1	×
从机接收（每个字节 ACK/NACK 控制）	1	×	1
从机发送/接收（所有字节响应 ACK）	×	×	0

11.4.3 I²C 主机通信流程

1. I²C时钟初始化（配置I2C_CLKCTRL寄存器）

- I²C 时钟分频：DIV[7：0]
- 数据保持时间（$t_{HD;DAT}$）：SDAD[3：0]
- 数据建立时间（$t_{SU;DAT}$）：设置 SCLD[3：0]
- SCL 高电平时间：设置 SCLH[7：0]
- SCL 低电平时间：设置 SCLL[7：0]

该寄存器的配置可以使用Artery_I2C_Timing_Configuration时钟配置工具计算。

2. 设置传输字节数

- ≤255 字节

  配置 I2C_CTRL2 的 RLDEN=0，关闭重载模式

配置 I2C_CTRL2 的 CNT[7:0]=N
- ＞255 字节
  配置 I2C_CTRL2 的 RLDEN=1，使能重载模式
  配置 I2C_CTRL2 的 CNT[7:0]=255
  剩余传输字节数 N=N-255

3. 设置传输结束模式
- ASTOPEN=0：软件结束模式，当数据传输完成后，I2C_STS 的 TDC 标志置 1，软件设置 GENSTOP=1 或者 GENSTART=1，发送 STOP 条件或者 START 条件。
- ASTOPEN=1：自动结束模式，当数据传输完成后，自动发送 STOP 条件。

4. 设置从机地址
- 设置寻址的从机地址值（I2C_CTRL2 的 SADDR）
- 设置从机地址模式（I2C_CTRL2 的 ADDR10）：
  ADDR10=0：7 位地址模式
  ADDR10=1：10 位地址模式

5. 设置传输方向（I2C_CTRL2的DIR）
- DIR=0：主机接收数据
- DIR=1：主机发送数据

6. 开始传输

设置I2C_CTRL2的GENSTART=1，主机开始在总线上发送START条件和从机地址，当从机响应了地址之后，主机的I2C_STS的ADDRF=1，设置I2C_CLR的ADDRC=1清除ADDRF标志后，开始数据传输。

7. 主机发送数据
1. I2C_TXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 TDIS=1；
2. 向 TXDT 数据寄存器写入数据 1，然后数据将被立即放进移位寄存器；
3. 此时 TXDT 数据寄存器为空，TDIS 又置 1；
4. 向 TXDT 数据寄存器写入数据 2，TDIS 被清 0；
5. 重复 2、3 步骤直到发送 CNT[7:0]个数据；
6. 如果此时 I2C_STS 的 TCRLD=1（重载模式），分为以下两种情况：
   剩余字节数 N>255：向 CNT 写入 255，N=N-255，TCRLD 被自动清 0，传输继续；
   剩余字节数 N≤255：关闭重载模式（RLDEN=0），向 CNT 写入 N，TCRLD 被自动清 0，传输继续。

8. 主机接收数据
1. 当正确匹配了从机地址之后，主机的 I2C_STS 的 ADDRF=1，设置 I2C_CLR 的 ADDRC=1 清除 ADDRF 标志后，开始数据传输；
2. 当收到数据后，RDBF=1，读取 RXDT 数据寄存器，RDBF 被自动清零；
3. 重复步骤 2 直到接收 CNT[7:0]个数据；
4. 如果此时 I2C_STS 的 TCRLD=1（重载模式），分为以下两种情况：
   剩余字节数 N>255：向 CNT 写入 255，N=N-255，TCRLD 被自动清 0，传输继续；
   剩余字节数 N≤255：关闭重载模式（RLDEN=0），向 CNT 写入 N，TCRLD 被自动清 0，传输继续；
5. 当在接收到最后一个字节时，主机会自动发送一个 NACK。

9. 结束时序
- 停止条件产生：
  软件结束模式（ASTOPEN=0）：此时 I2C_STS 的 TDC 置 1，设置 GENSTOP=1 产生 STOP 条件；
  自动结束模式（ASTOPEN=1）：自动产生 STOP 条件。
- 等待产生 STOP 条件，当 STOP 条件产生时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPC 写 1，清除 STOPF 标志，传输结束。

主机在传输过程中，如果收到了 NACK，那么此时 I2C_STS 的 ACKFAIL 置 1，并自动发送 STOP 条件结束通信，无论当前是软件结束模式（ASTOPEN=0）还是自动结束模式（ASTOPEN=1）。

主机发送流程

图 11-5 I²C 主机发送流程图

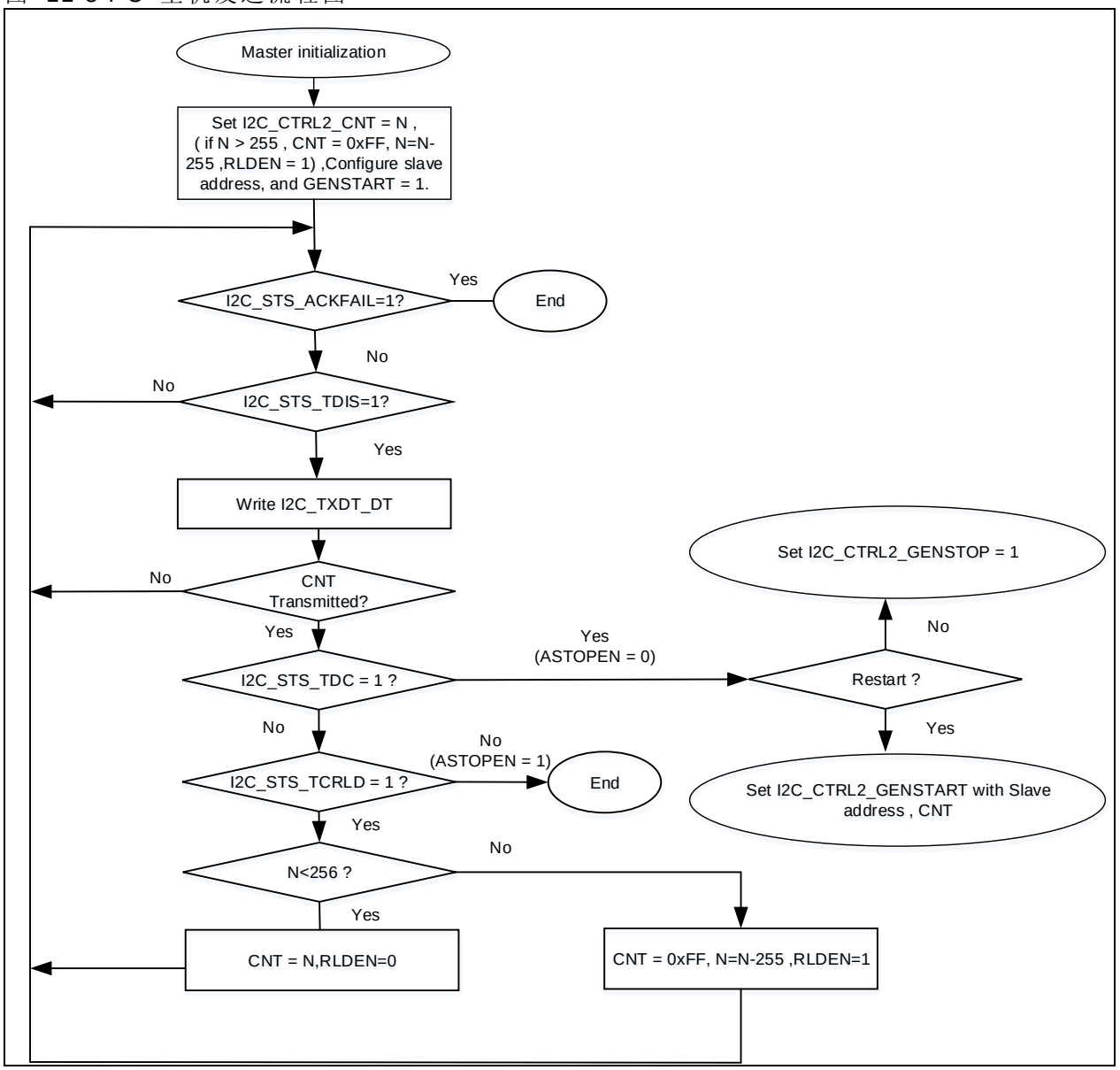
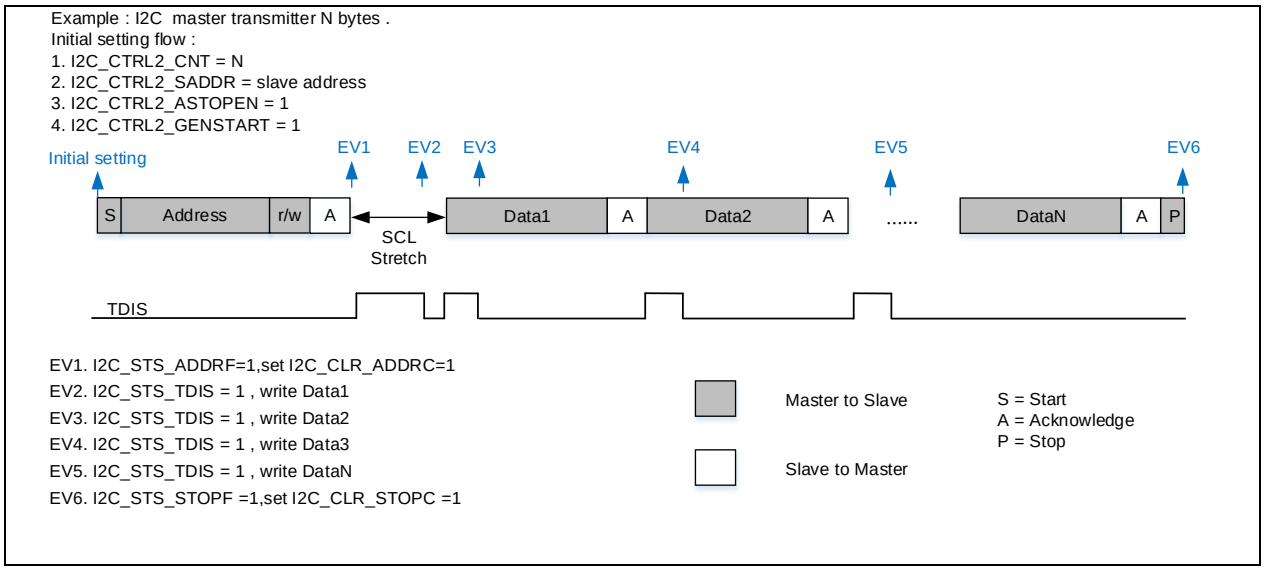


图 11-6 I²C主机发送时序图



主机接收流程

图 11-7 I²C主机接收流程图

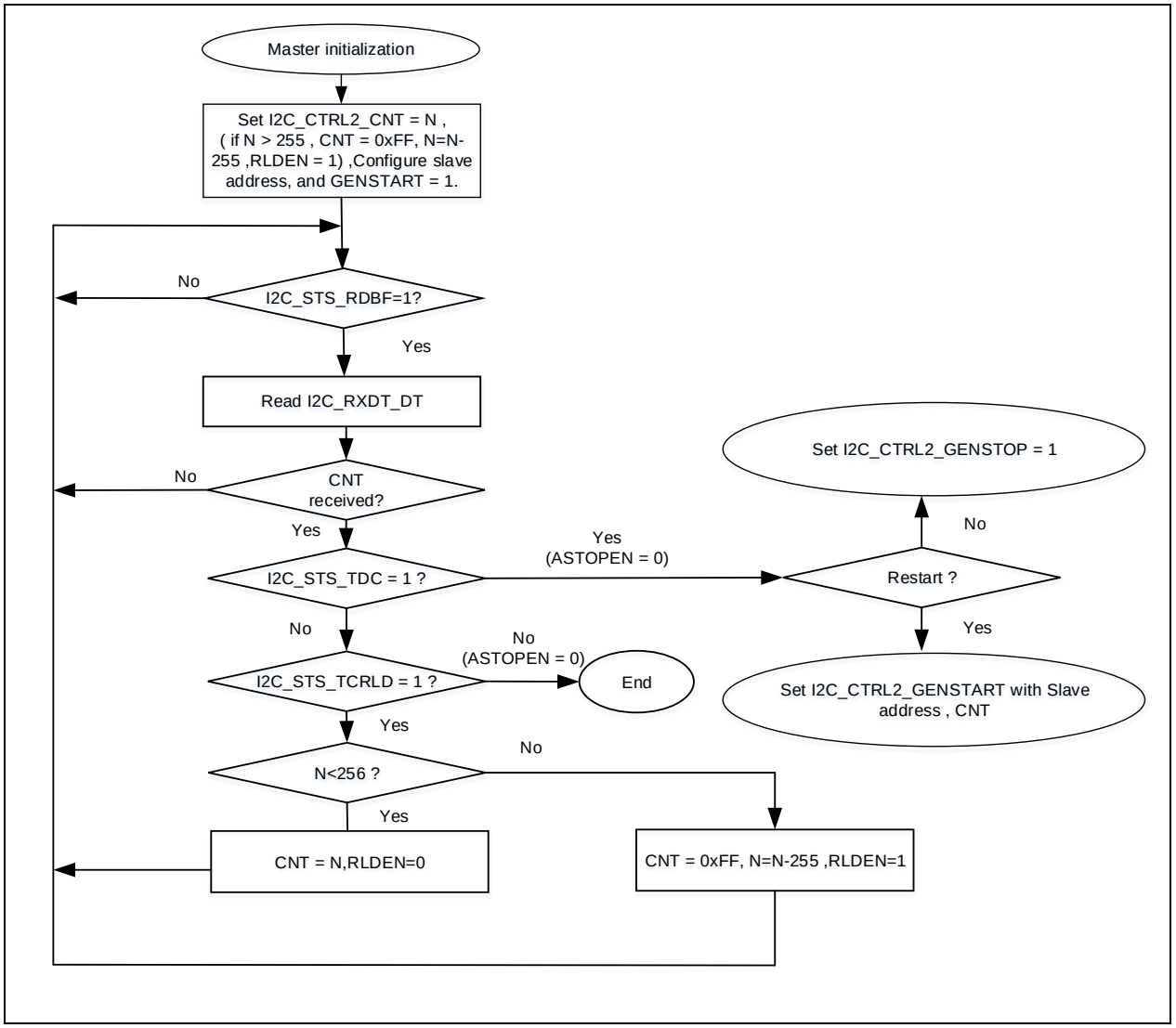
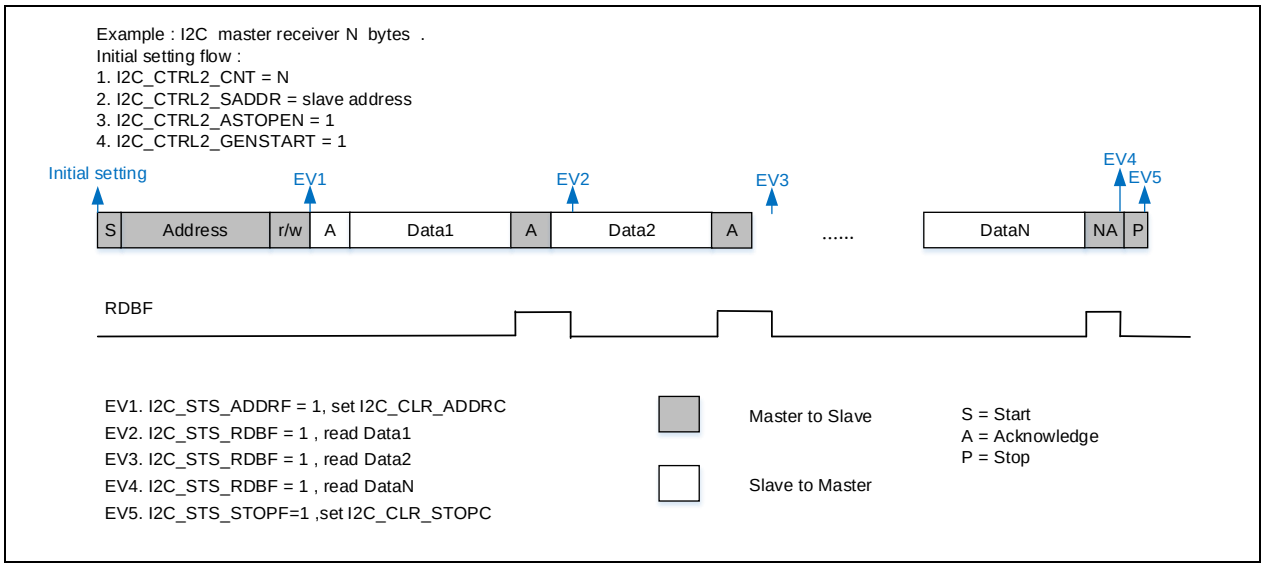


图 11-8 I²C 主机接收时序图



主机特殊传输时序支持

在 10 位地址传输模式下，I2C_CTRL2 的 READH10 用于产生特殊时序，当 READH10=1 时，支持如下传输序：主机先发送数据给从机，然后再从从机读取数据，传输时序图如下图所示：

使用方法：

主机在软件结束模式（ASTOPEN = 0）下，发送数据到从机，当数据发送完成后设置 READH10=0，然后再从从机接收数据。

图 11-9 10位地址读访问READH10 = 1

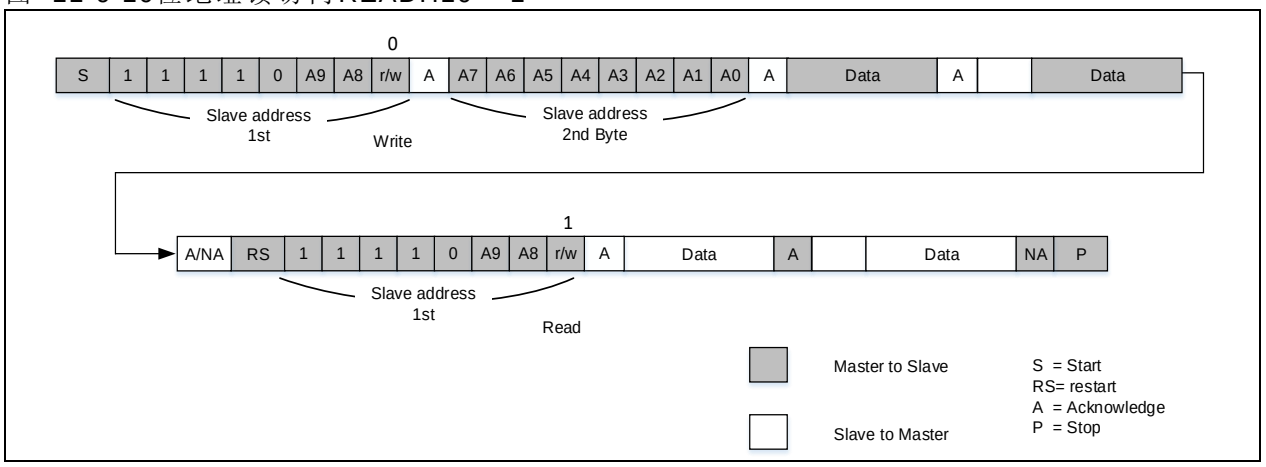
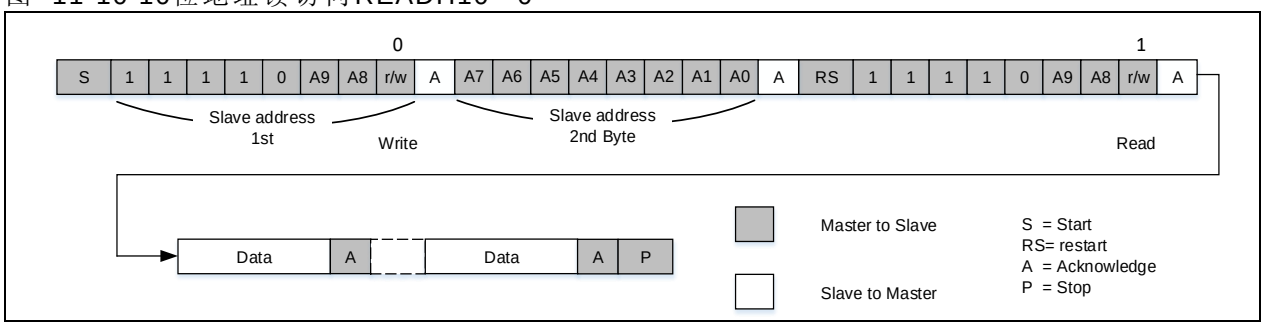


图 11-10 10位地址读访问READH10= 0



11.4.4 I²C从机通信流程

1. I2C时钟初始化（配置I2C_CLKCTRL寄存器）

— I²C 时钟分频：DIV[7：0]

- 数据保持时间（$t_{HD;DAT}$）：SDAD[3：0]
- 数据建立时间（$t_{SU;DAT}$）：设置 SCLD[3：0]

该寄存器的配置可以使用Artery_I2C_Timing_Configuration时钟配置工具计算。

2. 设置本机地址1

- 设置地址模式：
  7 位地址：设置 I2C_OADDR1 的 ADDR1MODE = 0
  10 位地址：设置 I2C_OADDR1 的 ADDR1MODE = 1
- 设置地址 1：设置 I2C_OADDR1 的 ADDR1
- 使能地址 1：设置 I2C_OADDR1 的 ADDR1EN=1

3. 设置本机地址2

- 设置地址 2：设置 I2C_OADDR2 的 ADDR2
- 设置地址 2 屏蔽位：设置 I2C_OADDR2 的 ADDR2MASK
- 使能地址 2：设置 I2C_OADDR2 的 ADDR2EN=1

需要注意的是，地址 2 只支持 7 位地址模式，并且可以通过 ADDR2MASK 位来灵活的屏蔽一些地址位，使从机响应一些特定的地址，关于 ADDR2MASK 位的详细用法，请参照章节 14.2。

在只使用一个地址的情况下，此地址不用配置，只需要配置地址 1。

4. 等待地址匹配

当接收到本机地址后，I2C_STS的ADDRF标志置1，此时可以读取I2C_STS的SDIR位，得到数据传输方向，当SDIR=0时数据传输方向为从机接收数据，当SDIR=1时，数据方向为从机发送数据，通过读取I2C_STS的ADDR[6:0]标志，可以知道接收到的地址是多少，这在双地址模式以及使用了地址2的屏蔽功能模式下，比较有用。

当通过设置I2C_CLR的ADDRC=1清除ADDRF标志后，开始数据传输。

5. 传输数据（从机发送，开启时钟延展，STRETCH=0）

当在地址匹配后：

1. I2C_TXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 TDIS=1；
2. 向 TXDT 数据寄存器写入数据 1，然后数据将被立即放进移位寄存器；
3. 此时 TXDT 数据寄存器为空，TDIS 又置 1；
4. 向 TXDT 数据寄存器写入数据 2，TDIS 被清 0；
5. 重复 3、4 步骤直到数据发送完成；
6. 等待收到 NACK 条件，当收到 NACK 条件时，I2C_STS 的 ACKFAILF 置 1，将 I2C_CLR 的 ACKFAILC 写 1，清除 ACKFAILF 标志；
7. 等待收到 STOP 条件，当收到 STOP 条件时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPC 写 1，清除 STOPF 标志，传输结束。

需要注意的是，在时钟延展关闭（STRETCH=1）的情况下，如果在将要传输数据的第一个 Bit 位开始发送之前，也就是 SDA 边沿产生之前，如果数据还未写入 TXDT 数据寄存器，那么会发生欠载错误，此时 I2C_STS 的 OUF 将会置 1，并将 0xFF 发送到总线。

为了能及时的写入数据，可以在通信开始前，先将数据写入到 DT 寄存器，写入的方式有如下两种：

- 直接写入：软件先将TDBE置1，目的是为了清空TXDT寄存器的数据，然后将第一个数据写入TXDT寄存器，此时TDBE清零；
- 通过中断或者DMA写入：软件先将TDBE置1，目的是为了清空TXDT寄存器的数据，再将TDIS置1，目的是为了产生一个TDIS事件，TDIS事件可以产生一个中断或者DMA请求，此时就可以通过DMA或者在中断函数内将数据写入TXDT寄存器。

6. 传输数据（从机接收，开启时钟延展，STRETCH=0）

当在地址匹配后：

1. I2C_RXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 RDBF=0；
2. 当收到数据后，RDBF=1，读取 RXDT 数据寄存器，RDBF 被自动清零；
3. 重复步骤 2 直到所有数据接收完成；
4. 等待收到 STOP 条件，当收到 STOP 条件时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPC 写 1，清除 STOPF 标志，传输结束。

在从机接收模式下，可以选择使用从机字节控制模式（SCTRL=1）接收数据，在从机字节控制模式下，可以对每一个接收到的字节进行 ACK/NACK 的控制，这种模式通常使用在 SMBus 协议中，关于从机字节控制模式的详细用法，请参照 11.4.2 数据传输管理章节。

需要注意的是，在时钟延展关闭（STRETCH=1）的情况下，在收到数据后从机应该及时的将数据读走，如果在已经收到 1 个字节后，在下一个字节接收完成之前，如果还未将数据读走，从机将产生过载错误，此时 I2C_STS 的 OUF 将会置 1，并自动回复 NACK。

上述提到的标志，均可通过相应的中断使能位，产生中断，具体的对应关系，请见中断章节。

从机发送流程

图 11-11 I²C 从机发送流程图

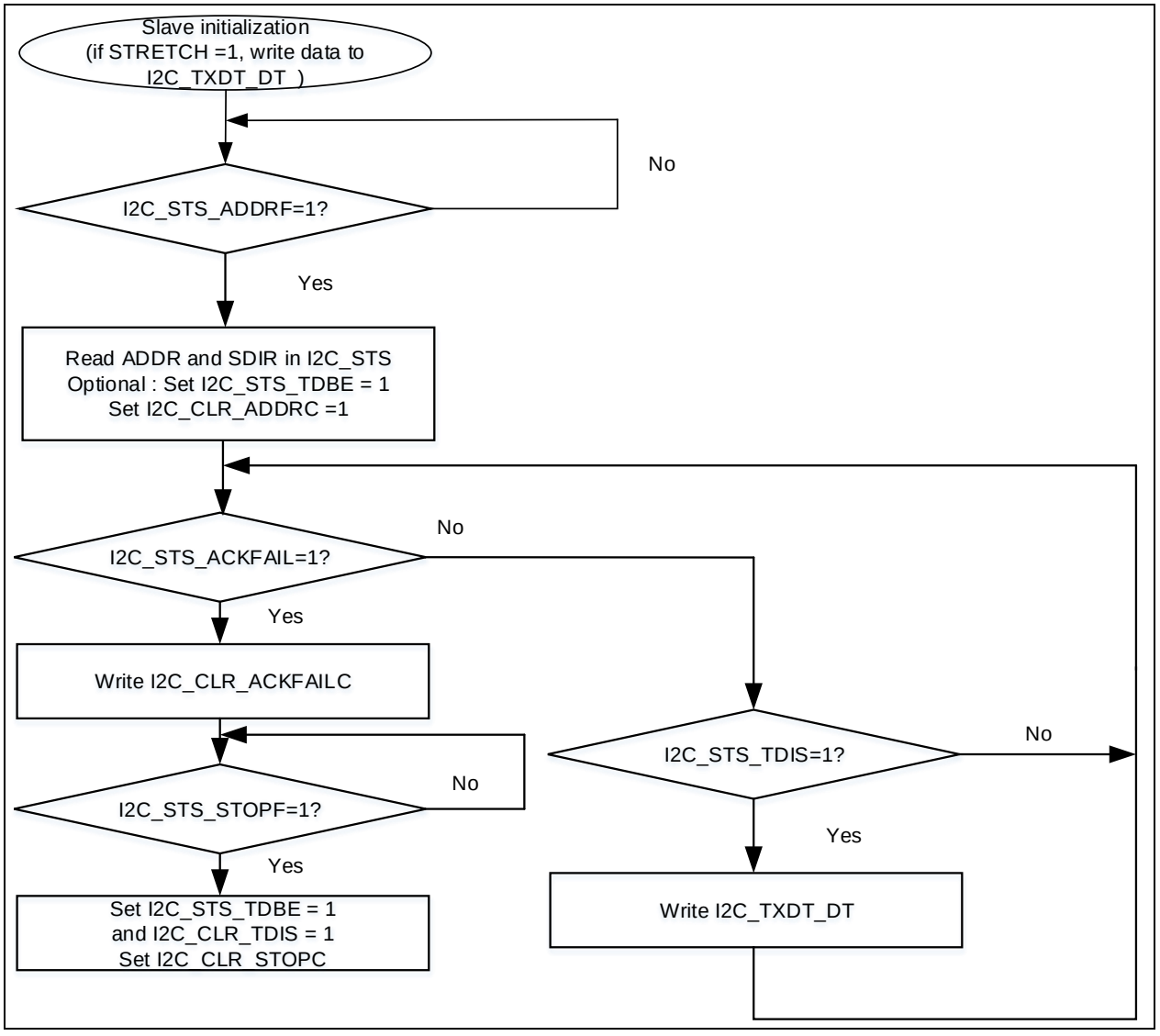
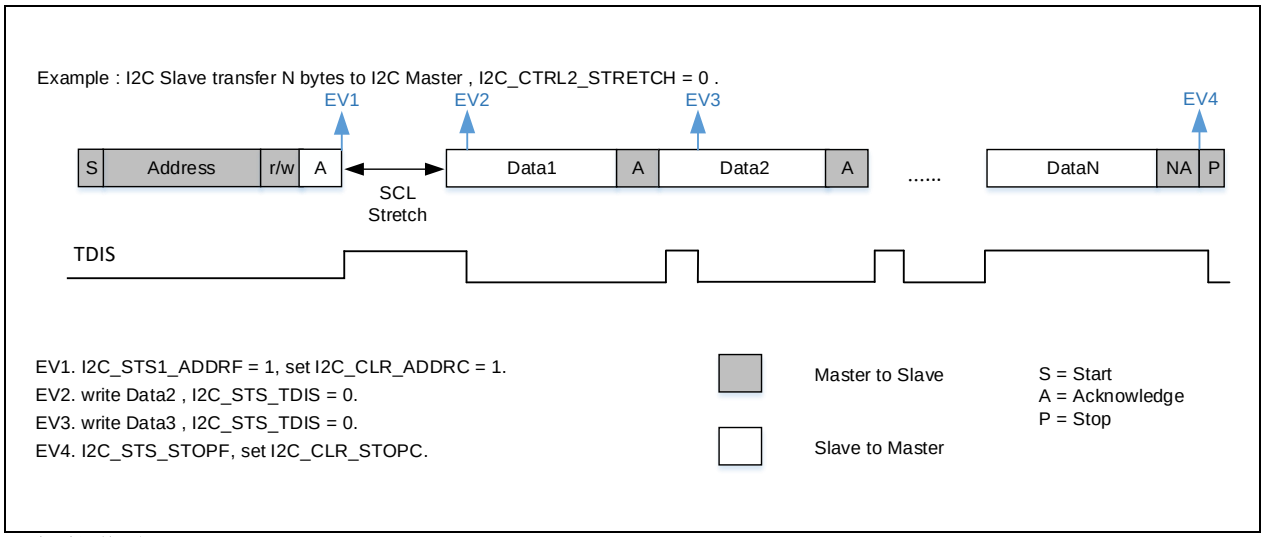


图 11-12 I²C 从机发送时序图



从机接收流程

图 11-13 I²C 从机接收流程图

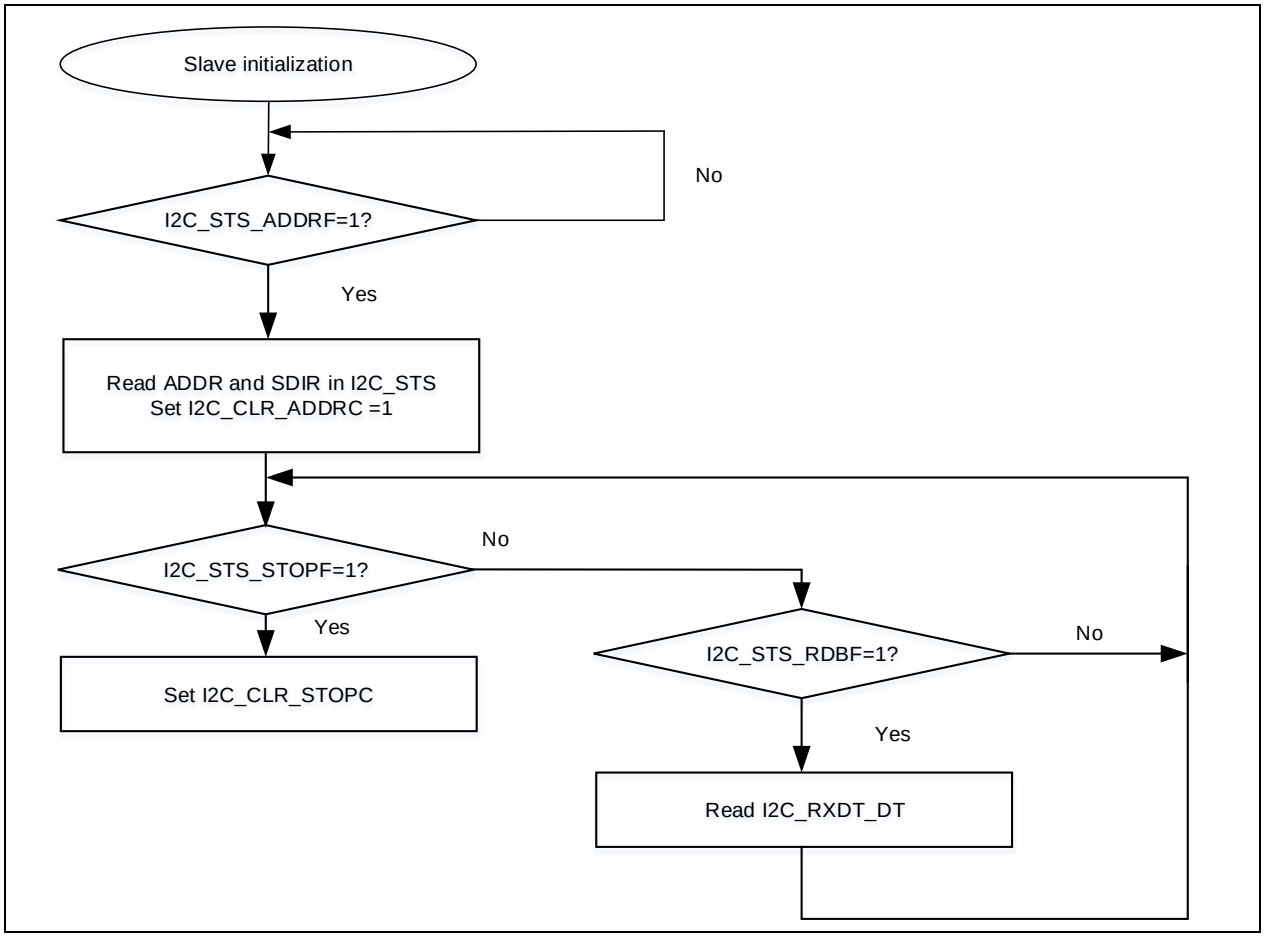
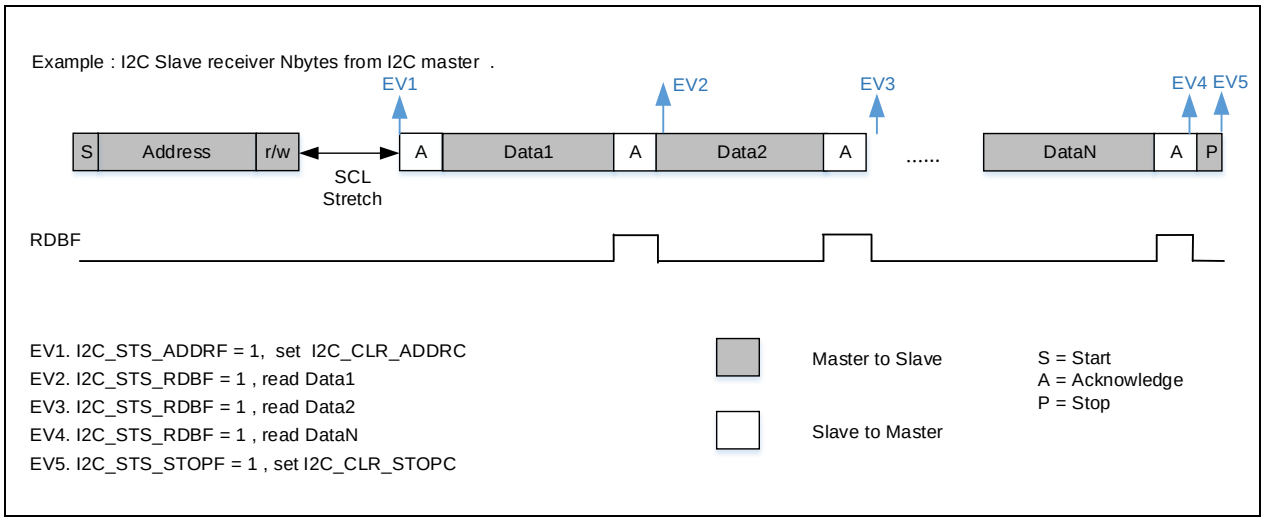


图 11-14 I²C 从机接收时序图



11.4.5 SMBus功能

SMBus即系统管理总线是一双线制总线，基于I²C的操作原理，系统中各设备之间通过SMBus总线传送和接收讯息，通过SMBus总线，设备可以提供制造商信息，告诉系统型号，报告不同类型错误，接受控制参数等。关于SMBus更加详细的信息请参考SMBus2.0协议。

SMBus 和 I²C 的差异

1. SMBus需维持最低10kHz以上的运作频率主要为了管理监控，只要在保持一定传速运作的情况下加入参数，就可轻松获知总线目前是否处于闲置（Idle）中，省去逐一侦测传输过程中的停断（STOP）信号，或持续保有停断侦测并辅以额外参数侦测，I²C则无
2. SMBus传输速度从最小10kHz到最大100kHz，I²C则是无最小传输速度，根据不同模式有不同的最大传输速度，分为标准模式(100kHz)和快速模式(400kHz)
3. SMBus有超时机制，I²C则无超时机制

SMBus 地址解析协议(ARP)

通过 ARP 协议可以给总线上的设备动态的分配一个唯一的新地址，解决地址冲突问题。关于 ARP 协议更详细的信息请参考 SMBus2.0 协议。

通过使能 I2C_CTRL1 寄存器的 DEVADDREN 位，可以使能 I²C 接口对设备默认地址（0b1100001x）的识别，但是像唯一设备标识（UDID）以及具体的协议实现过程，需要由软件来处理。

SMBus 主机通知协议

通过 SMBus 主机通知协议，可让从设备发送数据到主设备，例如从机可以通过此协议通知主机进行 ARP。关于 SMBus 主机通知协议更详细的信息请参考 SMBus2.0 协议。

当在主机模式（HADDREN =1）下，I²C 接口使能对主机默认地址（0b0001000x）的识别。

SMBus 提醒协议（SMBus Alert）

SMBALERT 是一个可选信号，连接主机和从机的 ALERT 引脚，用于从机通知主机访问从机，SMBALERT 是一个线与信号。关于 SMBus 提醒协议更详细的信息请参考 SMBus2.0 协议。

操作流程如下：

SMBus 主机

1. 启用SMBus提醒模式（SMBALERT=1）；
2. 根据实际需求启用ALERT中断；
3. 当ALERT引脚上产生了提醒事件时（ALERT引脚电平由高变低）；
4. 如果使能了中断，主机将产生ALERT中断；
5. 主机处理该中断并向从机发送提醒响应地址ARA（Alert Response Address）地址（0001100x），访问所有设备，获取取从机地址，只有那些将SMBALERT拉低的设备才会应答；
6. 主机通过获取到的从机地址进行下一步操作。

SMBus 从机

1. 产生提醒事件，ALERT引脚由高变低（SMBALERT=1），此时从机响应ARA（Alert Response Address）地址（0001100x）；
2. 等待主机通过发送ARA地址获取从机地址；

3. 上报自己的地址，如果发生了仲裁丢失，继续等待；
4. 地址上报成功，释放ALERT引脚（SMBALERT=0）。

包错误检查

包错误校验(PEC)用于保证数据传输的正确性和完成性，使用 CRC-8 进行校验，多项式为：

$C(x) = x^8 + x^2 + x + 1$

PEC 计算：当 I2C_CTRL1 的 PECEN=1 时启动 PEC 计算，检验数据包括地址以及数据。

PEC 传输：

- 主机：当I2C_CTRL2的PECTEN=1时，启动PEC传输使能，当数据传输个数达到N-1（CNT=N）了之后，主机会自动发送PEC；
- 从机：当I2C_CTRL2的PECTEN=1时，启动PEC传输使能，当数据传输个数达到N-1（CNT=N）了之后，从机会自动将第N个数据作为PEC检验，如果PEC校验不正确将回复NACK，并且I2C_STS的PECERR标志将会置1，当在从机发送模式下，无论校验是否正确，都将回复NACK。

SMBus 超时

在 SMBus 协议里面，主要有三种超时检测：

- 低电平超时$t_{TIMEOUT}$：时钟线SCL单次低电平时间（主机从机都会检测，主动被动拉低都会计算）；
- 从机低电平累积超时$t_{LOW:SEXT}$：从机在START条件到STOP条件之间的主动拉低SCL时间总和；
- 主机低电平累积超时$t_{LOW:MEXT}$：主机在上一个字节的ACK到下一次数据的第8个BIT位之间（单个字节），主动拉低拉低SCL时间总和。

需要注意的是$t_{LOW:SEXT}$和$t_{LOW:MEXT}$和只计算自己主动拉低的时间，外部拉低的时间不计算在内，而$t_{TIMEOUT}$主动、被动拉低都会计算。

表 11-3 SMBus超时规范

超时类型	最小值	最大值	单位
低电平超时 tTIMEOUT	25	35	ms
从机低电平累积超时 tLOW:SEXT	-	25	ms
主机低电平累积超时 tLOW:MEXT	-	10	ms

I2C 外设内置两个计数器用于检测超时，在 I2C_TIMEOUT 寄存器中配置，当发生超时事件时 I2C_STS 的 TMOUT 置 1，将 I2C_CLR 的 TMOUTC 置 1 可清 0。

- EXTTIME：用于主机、从机低电平累积超累计超时检测
  超时时间= (EXTTIME + 1) x 2048 x T_{I2C_CLK}
- TOTIME：时钟电平超时检测，可以通过TOMODE位来选择检测电平
  TOMODE=0：低电平超时检测，超时时间=(TOTIME + 1) x 2048 x T_{I2C_CLK}
  TOMODE=1：高电平超时检测，超时时间=(TOTIME + 1) x 4 x T_{I2C_CLK}

表 11-4 SMBus超时检测配置

超时类型	其他配置	使能	超时时间计算
低电平超时 tTIMEOUT	TOMODE=0	TOEN=1	(TOTIME + 1) x 2048 x TI2C_CLK
从机低电平累积超时 tLOW:SEXT	-	EXTEN=1	(EXTTIME + 1) x 2048 x TI2C_CLK
主机低电平累积超时 tLOW:MEXT	-	EXTEN=1	(EXTTIME + 1) x 2048 x TI2C_CLK

从机字节控制

在从机接收模式下，如果需要对接收到的每一个字节进行 ACK/NACK 的控制，可以选择从机接收数据控制模式（SCTRL=1）接收数据，关于从机字节控制模式的用法，请参照 11.4.2 数据传输管理章节。

表 11-5 SMBus模式配置表

传输模式	PECEN	PECTEN	RLDEN	ASTOPEN	SCTRL
主机发送/接收+STOP	1	1	0	1	-
主机发送/接收+RESTART	1	1	0	0	-
从机接收	1	1	1	-	1
从机发送	1	1	0	-	-

SMBus 使用流程

1. 设置SMBus默认地址响应：

   如果HADDREN=1：响应主机默认地址（0b0001000x）；
   如果DEVADDREN=1：响应设备默认地址（0b1100001x）。

2. PEC配置；
3. 如果是从机，可以根据需要使能从机字节控制模式（I2C_CTRL1的SCTRL）；
4. 其他配置和I²C使用配置一样。

需要注意的是各种 SMBus 协议需要由软件来实现，I²C 接口只提供了这些协议的地址识别。

11.4.6 SMBus主机通信流程

SMBus 主机通信流程和 I²C 主机通信流程类似。

1. **I²C时钟初始化（配置I2C_CLKCTRL寄存器）**
   - I²C 时钟分频：DIV[7：0]
   - 数据保持时间（$t_{HD;DAT}$）：SDAD[3：0]
   - 数据建立时间（$t_{SU;DAT}$）：设置 SCLD[3：0]
   - SCL 高电平时间：设置 SCLH[7：0]
   - SCL 低电平时间：设置 SCLL[7：0]

   该寄存器的配置可以使用Artery_I2C_Timing_Configuration时钟配置工具计算。

2. **SMBus相关初始化**
   - 选择为 SMBus 主机：设置 HADDREN=1，响应主机默认地址（0b0001000x）
   - 使能 PEC 计算：设置 I2C_CTRL1 的 PECEN=1
   - 使能 PEC 传输：设置 I2C_CTRL2 的 PECTEN=1

3. **设置传输字节数**
   - 配置 I2C_CTRL2 的 RLDEN=0，关闭重载模式
   - 配置 I2C_CTRL2 的 CNT[7:0]=N

   SMBus模式下单次传输字节数<255

4. **设置传输结束模式**
   - ASTOPEN=0：软件结束模式，当数据传输完成后，I2C_STS 的 TDC 标志置 1，软件设置 GENSTOP=1 或者 GENSTART=1，发送 STOP 条件或者 START 条件。
   - ASTOPEN=1：自动结束模式，当数据传输完成后，自动发送 STOP 条件。

5. **设置从机地址**
   - 设置寻址的从机地址值（I2C_CTRL2 的 SADDR）
   - 设置从机地址模式为 7 位地址模式（I2C_CTRL2 的 ADDR10=0）

6. **设置传输方向（I2C_CTRL2的DIR）**
   - DIR=0：主机接收数据
   - DIR=1：主机发送数据

7. **开始传输**

设置I2C_CTRL2的GENSTART=1，主机开始在总线上发送START条件和从机地址，当从机响应了地址之后，主机的I2C_STS的ADDRF=1，设置I2C_CLR的ADDRC=1清除ADDRF标志后，开始数据传输。

8. 主机发送数据

1. I2C_TXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 TDIS=1；
2. 向 TXDT 数据寄存器写入数据 1，然后数据将被立即放进移位寄存器；
3. 此时 TXDT 数据寄存器为空，TDIS 又置 1；
4. 向 TXDT 数据寄存器写入数据 2，TDIS 被清 0；
5. 重复 2、3 步骤直到发送 N-1 个数据；
6. 此时主机将自动发送第 N 个数据，也就是 PEC。

9. 主机接收数据

1. 当收到数据后，RDBF=1，读取 RXDT 数据寄存器，RDBF 被自动清零；
2. 重复步骤 1 直到接收 N 个数据，第 N 个数据为 PEC，在接收第 N 个字节，也就是 PEC 时，无论 PEC 是否正确，主机会自动发送一个 NACK。

10. 结束时序

— 停止条件产生：
软件结束模式（ASTOPEN=0）：此时 I2C_STS 的 TDC 置 1，设置 GENSTOP=1 产生 STOP 条件；
自动结束模式（ASTOPEN=1）：自动产生 STOP 条件。

— 等待产生 STOP 条件，当 STOP 条件产生时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPC 写 1，清除 STOPF 标志，传输结束。

SMBus 主机发送流程

图 11-15 SMBus主机发送流程图

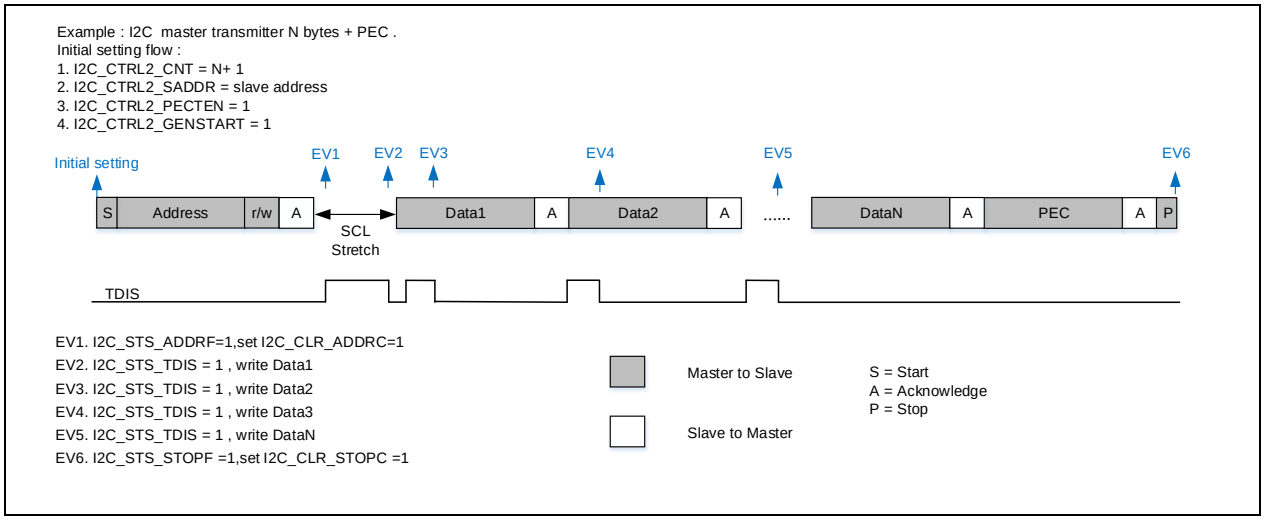
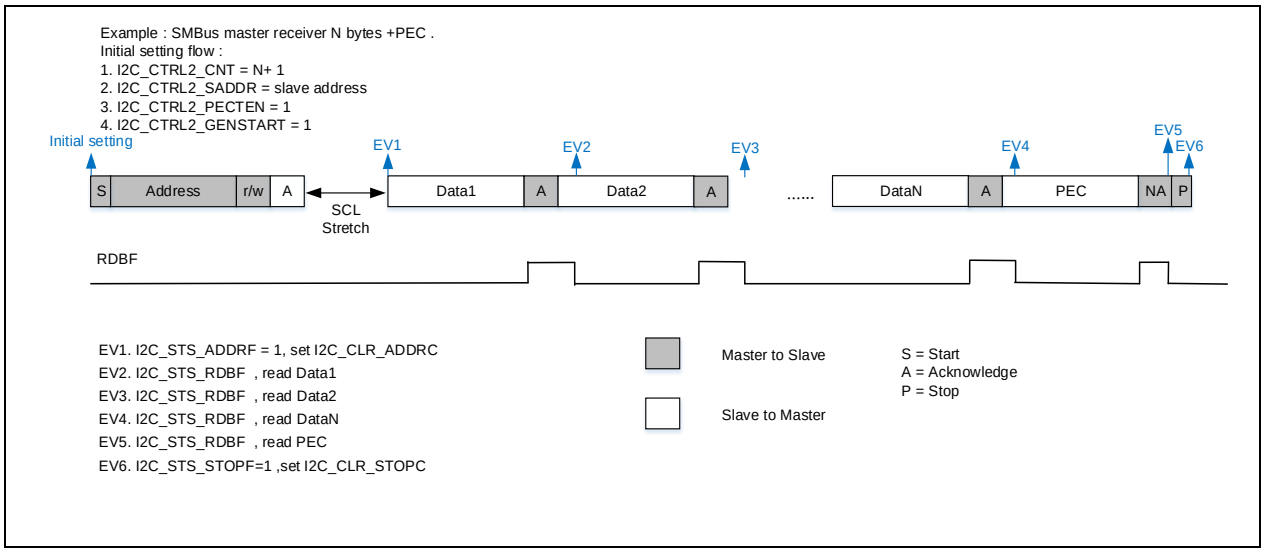


图 11-16 SMBus主机发送时序图



SMBus 主机接收流程

图 11-17 SMBus主机接收流程图

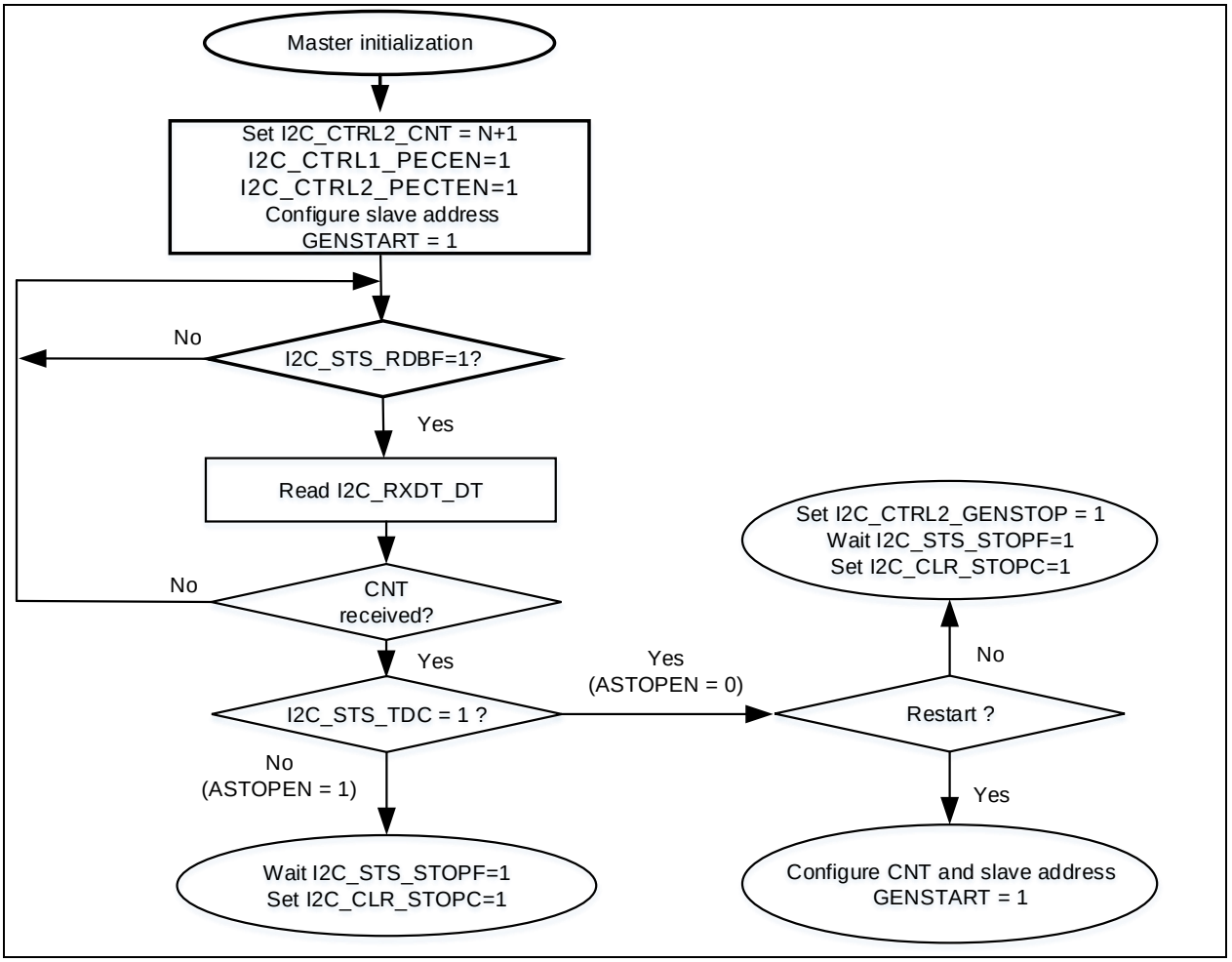
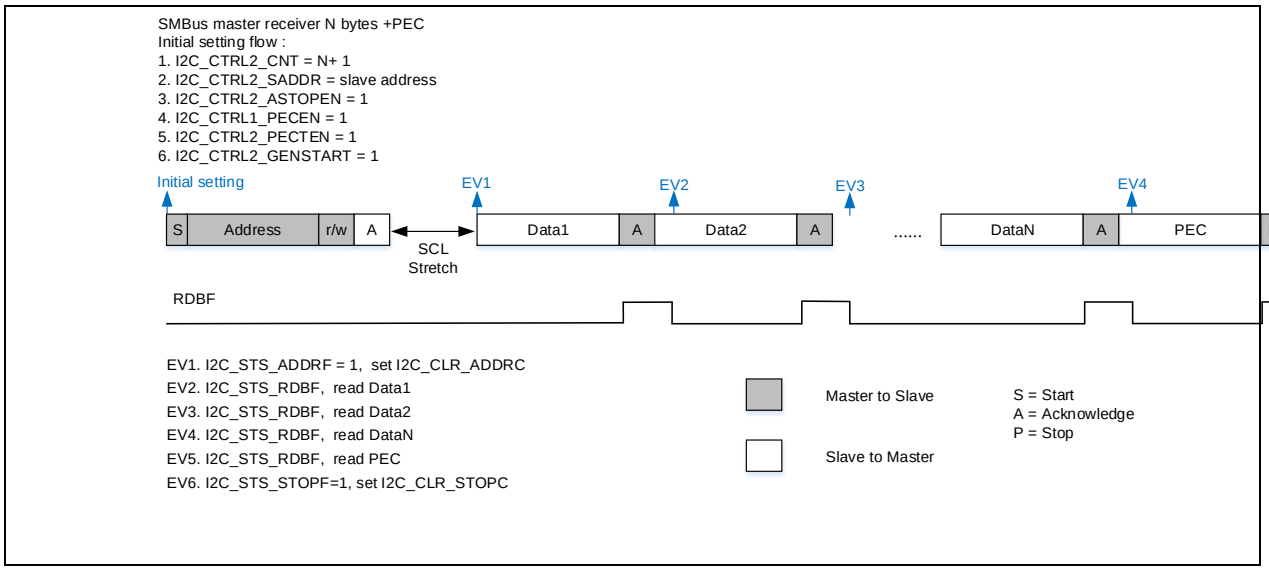


图 11-18 SMBus主机接收时序图



11.4.7 SMBus从机通信流程

SMBus 从机通信流程和 I²C 从机通信流程类似。

1. I2C时钟初始化（配置I2C_CLKCTRL寄存器）

- I²C 时钟分频：DIV[7：0]；
- 数据保持时间（$t_{HD;DAT}$）：SDAD[3：0]；
- 数据建立时间（$t_{SU;DAT}$）：设置 SCLD[3：0]；

该寄存器的配置可以使用Artery_I2C_Timing_Configuration时钟配置工具计算。

2. 设置本机地址

- 设置地址模式为 7 位：设置 I2C_OADDR1 的 ADDR1MODE = 0；
- 设置地址 1：设置 I2C_OADDR1 的 ADDR1；
- 使能地址 1：设置 I2C_OADDR1 的 ADDR1EN=1。

3. SMBus相关初始化

- 选择为 SMBus 设备：设置 DEVADDREN=1，响应设备默认地址（0b1100001x）；
- 使能 PEC 计算：设置 I2C_CTRL1 的 PECEN=1；
- 设置从机字节控制模式：

  从机发送：关闭字节控制模式，设置 I2C_CTRL1 的 SCTRL=0；

  从机接收：使能字节控制模式，设置 I2C_CTRL1 的 SCTRL=1。

4. 等待地址匹配

当接收到本机地址后，I2C_STS的ADDRF标志置1，此时可以读取I2C_STS的SDIR位，得到数据传输方向，当SDIR=0时数据传输方向为从机接收数据，当SDIR=1时，数据方向为从机发送数据，通过读取I2C_STS的ADDR[6:0]标志，可以知道接收到的地址是多少。

使能PEC传输：设置I2C_CTRL2的PECTEN=1。

设置传输个数：

- 从机发送：设置I2C_CTRL2的CNT=N；
- 从机接收：设置I2C_CTRL2的CNT=1；

重载模式设置：

- 从机发送：设置I2C_CTRL2的RLDEN =0；
- 从机接收：设置I2C_CTRL2的RLDEN =1；

设置I2C_CLR的ADDRC=1，清除ADDRF标志，开始数据传输。

5. 传输数据（从机发送，开启时钟延展，STRETCH=0）

当在地址匹配后：

1. I2C_TXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 TDIS=1；
2. 向 TXDT 数据寄存器写入数据 1，然后数据将被立即放进移位寄存器；
3. 此时 TXDT 数据寄存器为空，TDIS 又置 1；
4. 向 TXDT 数据寄存器写入数据 2，TDIS 被清 0；
5. 重复 3、4 步骤直到数据发送 N-1 个数据；
6. 此时从机将自动发送第 N 个数据，也就是 PEC；
7. 等待收到 NACK 条件，当收到 NACK 条件时，I2C_STS 的 ACKFAILF 置 1，将 I2C_CLR 的 ACKFAILC 写 1，清除 ACKFAILF 标志；
8. 等待收到 STOP 条件，当收到 STOP 条件时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPC 写 1，清除 STOPF 标志，传输结束。

6. 传输数据（从机接收，开启时钟延展，STRETCH=0）

当在地址匹配后：

1. I2C_RXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 RDBF=0；
2. 当收到一个字节后，RDBF=1，TCRLD 置 1，从机拉住 SCL 总线；
3. 读取 RXDT 数据寄存器，RDBF 被自动清零；
4. 根据需要配置 I2C_CTRL2 的 NACKEN 位，来产生一个 ACK 或 NACK；
   如果产生一个NACK，通信结束；
   如果产生一个ACK，通信继续，此时写入CNT=1，硬件自动清除TCRLD标志，从机释放SCL总线，继续接收下一个字节；
5. 重复步骤 2、3、4 直到接收 N-1 个数据；
6. 设置 I2C_CTRL2 的 RLDEN =0，关闭重载模式，设置 CNT=1，重复 2、3 步骤接收 PEC
   如果 PEC 校验错误，PECERR 标志将会置 1；
7. 等待收到 STOP 条件，当收到 STOP 条件时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPC 写 1，清除 STOPF 标志，传输结束。

SMBus 从机发送

图 11-19 SMBus从机发送流程图

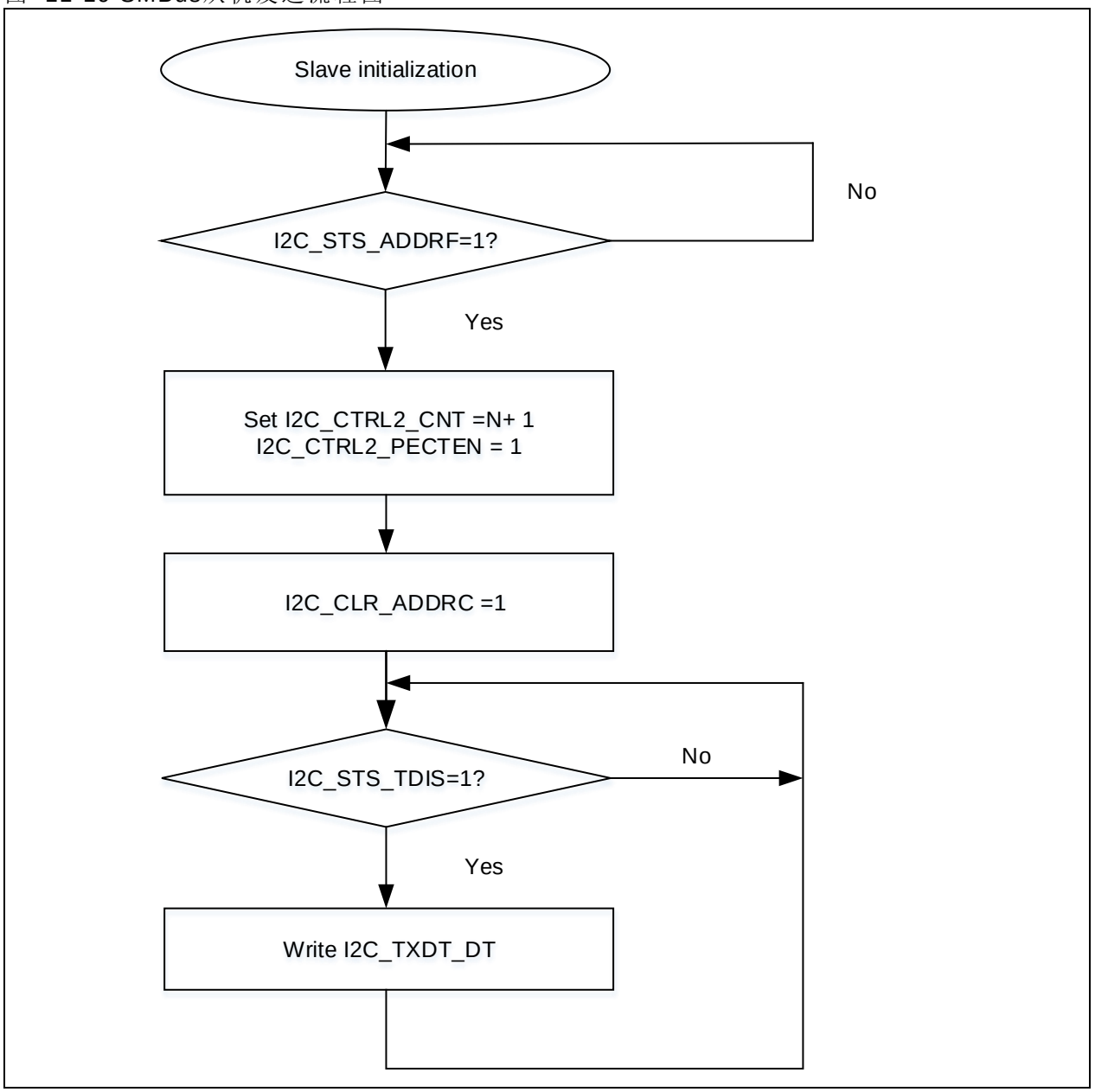
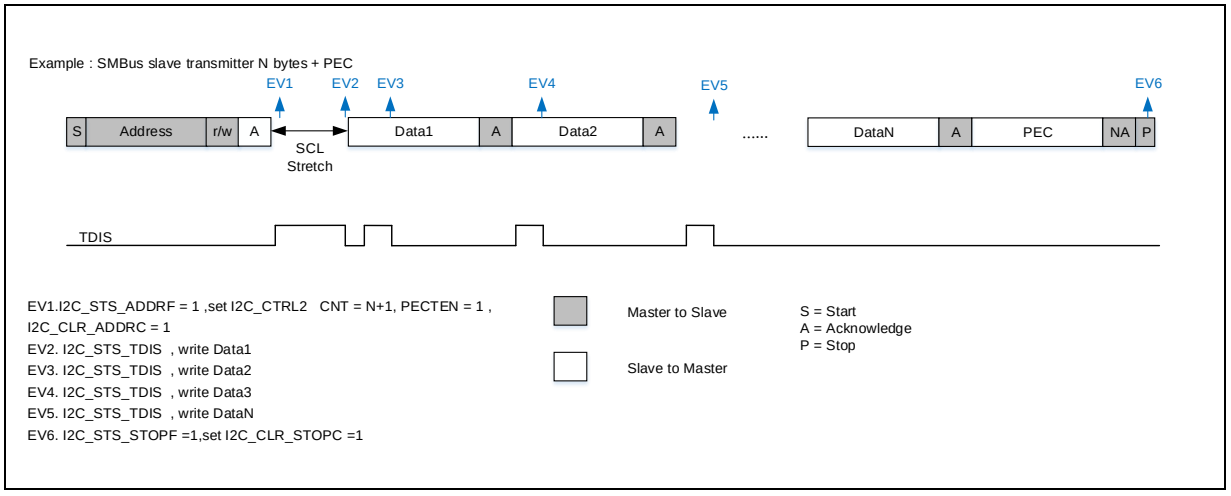


图 11-20 SMBus从机发送时序图



SMBus 从机接收

图 11-21 SMBus从机接收流程图

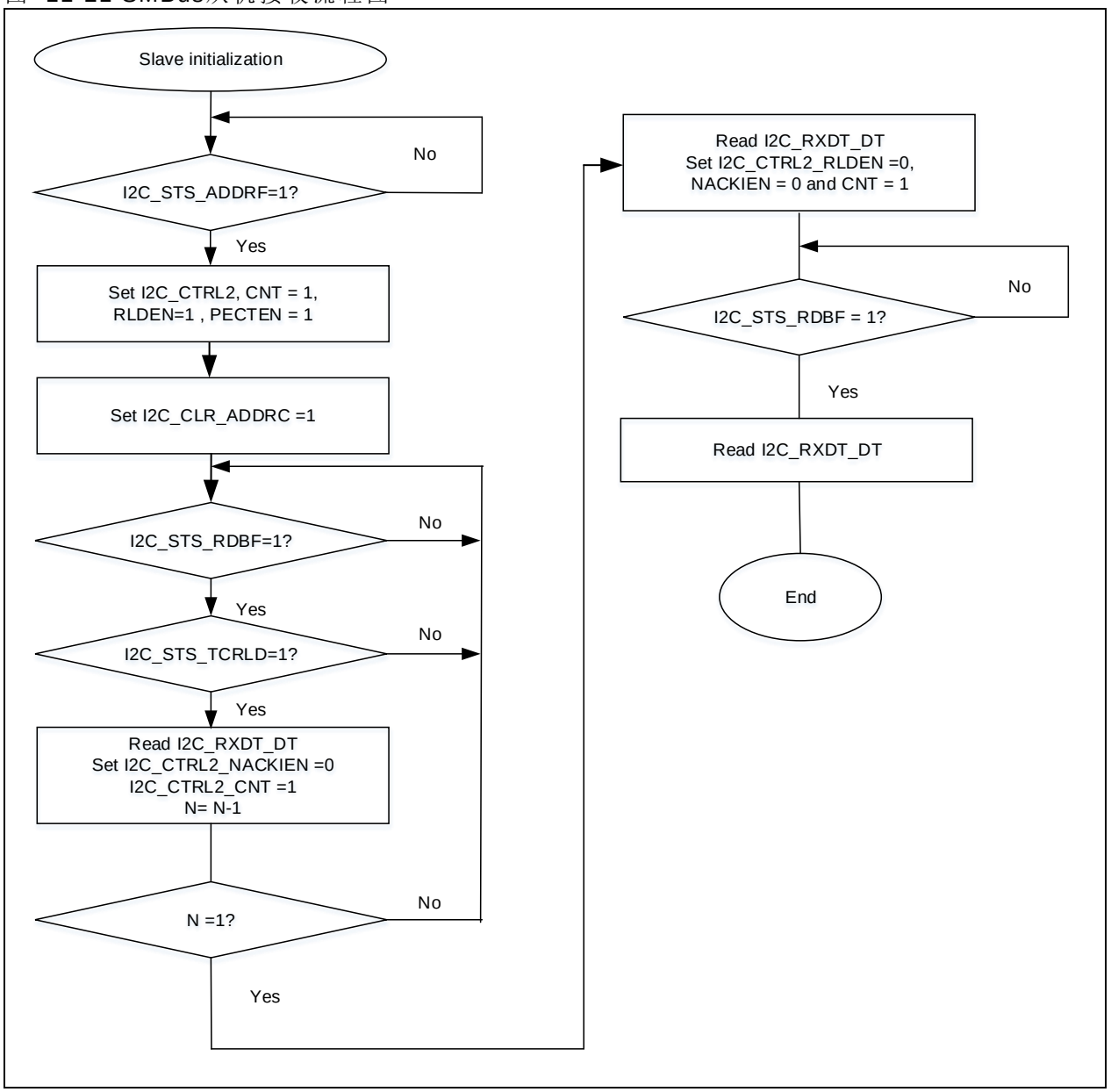
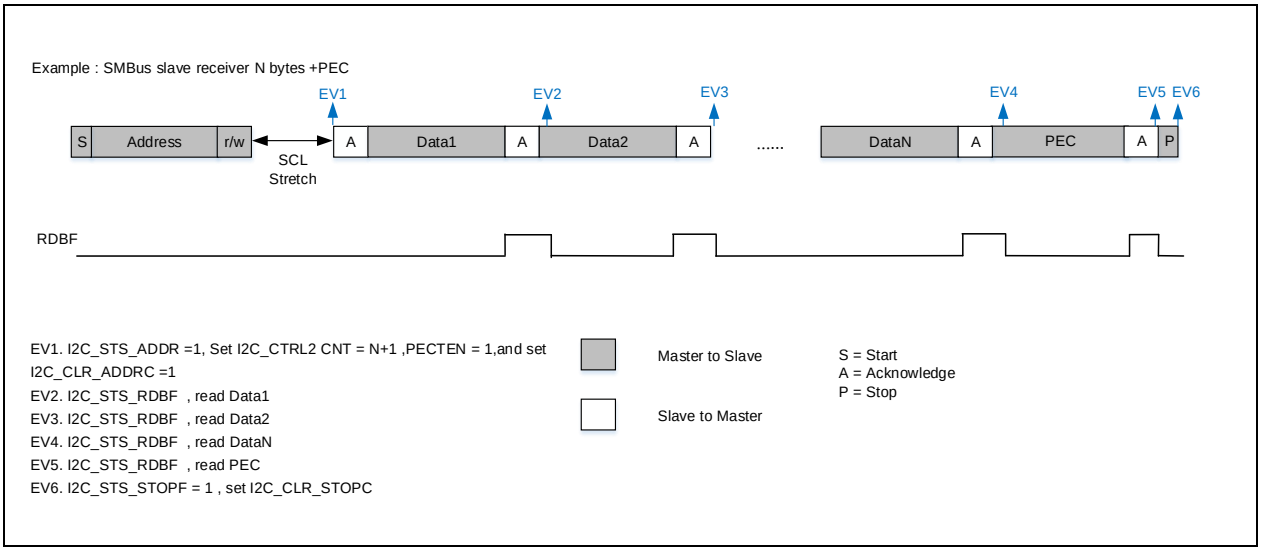


图 11-22 SMBus 从机接收时序图



11.4.8 DMA传输

I²C 可以使用 DMA 进行数据传输，降低 CPU 负担，在使用 DMA 传输时，TDIEN 和 RDIEN 应该保持关闭。

DMA 发送（DMATEN=1）

1. 设置外设地址（DMA通道x外设地址寄存器（DMA_CxPADDR）=数据寄存器(I2C_TXDT)地址）
2. 设置数据存储地址（DMA通道x存储器地址寄存器（DMA_CxMADDR）=数据存储地址）
3. 设置传输方向为内存到外设（DMA_CHCTRL的DTD=1）
4. 设置传输字节数（DMA通道x数据传输量寄存器（DMA_CxDTCNT））
5. 设置DMA通道的其他配置，例如：优先级、存储器数据宽度、外设数据宽度、中断等（DMA_CHCTRL）
6. 使能DMA通道（DMA通道x配置寄存器（DMA_CxCTRL）的CHEN=1）。
7. 使能I²C DMA请求（控制寄存器2（I2C_CTRL2）的DMAEN=1），当状态寄存器1（I2C_STS）的TDIS位被置1时，DMA将数据从内存地址传输到数据寄存器(I2C_TXDT)
8. 等待传输字节数DMA通道x数据传输量寄存器（DMA_CxDTCNT）=0时，数据传输完成，（可以通过DMA传输完成中断来等待）。
9. 主机发送模式：停止时序见I²C主机通信流程章节。
   从机发送模式：停止时序见I²C从机机通信流程章节。

DMA 接收（DMAREN=1）

1. 设置外设地址（DMA通道x外设地址寄存器（DMA_CxPADDR）=数据寄存器(I2C_RXDT)地址）；
2. 设置数据存储地址（DMA通道x存储器地址寄存器（DMA_CxMADDR）=数据存储地址）；
3. 设置传输方向为外设到内存（DMA_CHCTRL的DTD=0）；
4. 设置传输字节数（DMA通道x数据传输量寄存器（DMA_CxDTCNT））；
5. 设置DMA通道的其他配置，例如：优先级、存储器数据宽度、外设数据宽度、中断等（DMA_CHCTRL）；
6. 使能DMA通道（DMA通道x配置寄存器（DMA_CxCTRL）的CHEN=1）；
7. 使能I²C DMA请求（控制寄存器2（I2C_CTRL2）的DMAEN=1），当状态寄存器1（I2C_STS）的RDBF位被置1时，DMA将数据从I2C_DT寄存器传输到数据存储地址；
8. 等待传输字节数DMA_TCNTx=0时，数据传输完成，（可以通过DMA传输完成中断来等待）；
9. 主机接收模式：停止时序见I²C主机通信流程章节。
   从机接收模式：停止时序见I²C从机机通信流程章节。

11.4.9 错误管理

I²C 内部有多种错误管理，可以极大的提高通信的可靠性，支持错误管理的事件如下：

表 11-6 I²C 错事件

错误事件	事件标志	中断使能位	事件清除位
SMBus 提醒	ALERTF	ERRIEN	ALERTC
超时错误	TMOUT	ERRIEN	TMOUTC
PEC 错误	PECERR	ERRIEN	PECERRC
过载/欠载	OUF	ERRIEN	OUFC
仲裁丢失	ARLOST	ERRIEN	ARLOSTC
总线错误	BUSERR	ERRIEN	BUSERRC

过载或者欠载（OUF）

只有在从机模式下，且关闭时钟延展（I2C_CTRL1 的 STRETCH=1）时，才有可能出现欠载或者过载错误。

从机发送模式：如果在将要传输数据的第一个 Bit 位开始发送之前，也就是 SDA 边沿产生之前，如果数据还未写入 TXDT 数据寄存器，那么会发生欠载错误，此时 I2C_STS 的 OUF 将会置 1，并将 0xFF 发送到总线。

从机接收模式：在收到数据后从机应该及时的将数据读走，如果在已经收到 1 个字节后，在下一个字节接收完成之前，如果还未将数据读走，从机将产生过载错误，此时 I2C_STS 的 OUF 将会置 1，并自动回复 NACK。

仲裁丢失（ARLOST）

当设备控制 SDA 线输出高电平，但是总线上实际输出低电平时，发生仲裁丢失事件。

- 主机发送数据：仲裁可以发生在地址传输、数据传输阶段；
- 主机接收数据：仲裁可以发生在地址传输、响应ACK阶段；
- 从机发送数据：仲裁可以发生在数据传输阶段；
- 从机接收数据：仲裁可以发生在响应ACK阶段。

当在发生仲裁丢失后，硬件自动将 I2C_STS 的 ARLOST 置 1，无论是主机还是从机，都将会立即释放 SCL、SDA 总线，并自动回到从机状态。

总线错误(BUSERR)

在数据传输阶段，在 SCL 高周期区间 SDA 线必须保持稳定，当 SCL 信号为低时，SDA 才能改变，否则将会出现总线错误，当 SCL 为高电平时：

- SDA从1变成0：错误的开始条件；
- SDA从0变成1：错误的停止条件；

以上两种情况都会触发总线错误，硬件自动将 I2C_STS 的 BUSERR 置 1。

PEC 错误（PECERR）

只有在 SMBus 模式下才存在 PEC，当在主机接收模式和从机接收模式下，如果接收到的 PEC 和内部计算的 PEC 不相等时，会出现 PEC 错误，此时硬件自动将 I2C_STS 的 PECERR 置 1。

在从机接收模式下，如果 PEC 不正确，从机将回复 NACK；

当在主机接收模式下，无论 PEC 是否正确，主机都将回复 NACK。

SMBus 提醒（ALERTF）

在 SMBus 主机模式下（HADDREN=1）且启用了 SMBus 提醒模式（SMBALERT=1）时，SMBus 提醒功能可以使用，当 ALERT 引脚上产生了提醒事件时（ALERT 引脚电平由高变低），硬件自动将 I2C_STS 的 ALERTF 置 1。

超时错误（TMOUT）

超时错误是 SMBus 协议所定义的，用来提高系统稳定性的机制，用来避免主机或者从机出现故障时一直拉低总线，导致总线无法使用的情况。当发生了超时事件时（SMBus 章节所定义的），硬件自动将 I2C_STS 的 TMOUT 置 1。如果是在从机模式下发生超时事件时，从机将立即释放 SCL 和 SDA 总线；如果是主机发生超时事件时，主机将自动发送一个 STOP 条件结束通信 。

11.4.10 地址匹配事件从Deepsleep mode 唤醒

I2C1 支持在被寻址到时将系统从 Deepsleep mode 唤醒，要开启此项功能需在进入 Deepsleep mode 前设置内部寄存器

- I2C1_CTRL1的WAKEUPEN位置1
- I2C1_CTRL1的DFLT位设为0
- I2C1_CTRL1的STRETCH位设为0
- CRM_PICLKS的I2C1SEL位选择HICK

Deepsleep mode 唤醒流程 ：

1. 设置好上述位后系统进入Deepsleep mode，此时HICK是关闭的
2. 当检测到I2C总在线的开始条件，I2C接口启动HICK并将SCL总线拉低
3. 待HICK打开后开始接收地址
   - 地址匹配 ：唤醒系统，在系统唤醒期间I2C接口会持续将SCL总线拉低，直到系统处理地址匹配中断并清除ADDRF标志
   - 地址不匹配 ：关闭HICK，系统也不会被唤醒
4. SCL总线被释放，进入正常传输状态
   - 禁止I2C在作为主机传输数据或作为从机被寻址到后进入Deepsleep mode

11.5 I²C 中断

下表列出了所有的 I²C 中断请求。

表 11-7 I²C 中断请求

中断事件	事件标志	中断使能位
地址匹配	ADDRF	ADDRIEN
应答失败	ACKFAIL	ACKFAILIEN
停止条件产生完成	STOPF	STOPIEN
发送中断状态	TDIS	TDIEN
接收数据缓冲器满	RDBF	RDIEN
传输完成等待加载数据	TCRLD	TDCIEN
数据传输完成	TDC	
SMBus 提醒	ALERTF	ERRIEN
超时错误	TMOUT	
PEC 错误	PECERR	
过载/欠载	OUF	
仲裁丢失	ARLOST	
总线错误	BUSERR	

11.6 I²C 调试模式

当微控制器进入调试模式(Cortex®-M4 核心处于停止状态)时，根据 DEBUG 模块中的 I2Cx_SMBUS_TIMEOUT 配置位，SMBUS 超时控制可以继续正常工作或者可以停止。

11.7 I²C 寄存器

下表给出了 I²C 寄存器映像和复位值。
必须以字（32 位）的方式操作这些外设寄存器。
表 11-8 寄存器映像和复位值

寄存器简称	基址偏移量	复位值
I2C_CTRL1	0x00	0x00000000
I2C_CTRL2	0x04	0x00000000
I2C_OADDR1	0x08	0x00000000
I2C_OADDR2	0x0C	0x00000000
I2C_CLKCTRL	0x10	0x00000000
I2C_TIMEOUT	0x14	0x00000000
I2C_STS	0x18	0x00000000
I2C_CLR	0x1C	0x00000000
I2C_PEC	0x20	0x00000000
I2C_RXDT	0x24	0x00000000
I2C_TXDT	0x28	0x00000000

11.7.1 控制寄存器1（I2C_CTRL1）

域	简称	复位值	类型	功能
位 31：24	保留	0x00	res	保持默认值。
位 23	PECEN	0x0	rw	PEC 计算使能（PEC calculation enable） 0：关闭； 1：开启。
位 22	SMBALERT	0x0	rw	SMBus 提醒功能使能/引脚设置（SMBus alert enable / pin set） SMBus 主机，提醒功能使能： 0：关闭； 1：开启。 SMBus 从机，提醒地址使能： 0：置高； 1：置低，响应 0001100x。
位 21	DEVADDREN	0x0	rw	SMBus 设备默认地址使能（SMBus device default address enable） 0：关闭； 1：开启，响应设备默认地址 1100001x。
位 20	HADDREN	0x0	rw	SMBus 主机地址使能（SMBus host address enable） 0：关闭； 1：开启，响应主机地址 0001000x。
位 19	GCAEN	0x0	rw	广播地址使能（General call address enable） 0：关闭； 1：开启，响应地址 0000000x。
位 18	WAKEUPEN	0x0	rw	从 Deepsleep mode 唤醒使能(Deepsleep mode wakeup enable) 0：关闭； 1：开启。
位 17	STRETCH	0x0	rw	时钟延展模式（Clock stretching mode） 0：开启； 1：关闭。 注：只在从机模式下有效。
位 16	SCTRL	0x0	rw	从机接收数据控制（Slave receiving data control） 0：关闭； 1：开启。

位 15	DMAREN	0x0	rw	DMA 数据接收使能（DMA receive data request enable） 0：关闭； 1：开启。
位 14	DMATEN	0x0	rw	DMA 数据发送使能（DMA Transmit data request enable） 0：关闭； 1：开启。
位 13	保留	0x0	res	保持默认值。
位 12	ANGNFOFF	0x0	rw	模拟滤波关闭(Analog filter off) 0：开启； 1：关闭。
位 11：8	DFLT	0x0	rw	数字滤波值（Digital filter value） SCL 总线上小于此宽度的毛刺将被滤除，滤波时间 = DFLT x T_{I2C_CLK}。
位 7	ERRIEN	0x0	rw	错误中断使能（Error interrupt enable） 0：关闭； 1：开启。
位 6	TDCIEN	0x0	rw	数据传输完成中断使能（Transfer data complete interrupt enable） 0：关闭； 1：开启。
位 5	STOPIEN	0x0	rw	停止条件产生完成中断使能（Stop generation complete interrupt enable） 0：关闭； 1：开启。
位 4	ACKFAILIEN	0x0	rw	应答失败中断使能（Acknowledge fail interrupt enable） 0：关闭； 1：开启。
位 3	ADDRIEN	0x0	rw	地址匹配中断使能（Address match interrupt enable） 0：关闭； 1：开启。
位 2	RDIEN	0x0	rw	数据接收中断使能（Receive data interrupt enable） 0：关闭； 1：开启。
位 1	TDIEN	0x0	rw	数据发送中断使能（Transmit data interrupt enable） 0：关闭； 1：开启。
位 0	I2CEN	0x0	rw	I^2C 外设使能（I^2C peripheral enable） 0：关闭； 1：开启。

11.7.2 控制寄存器2（I2C_CTRL2）

域	简称	复位值	类型	功能
位 31：27	保留	0x00	res	保持默认值。
位 26	PECTEN	0x0	rw	请求 PEC 传输使能（Request PEC transmission enable） 0：停止传输； 1：启动传输。
位 25	ASTOPEN	0x0	rw	自动发送停止条件使能（Automatically send stop condition enable） 0：关闭（软件结束模式）； 1：开启（自动结束模式）。
位 24	RLDEN	0x0	rw	发送数据重载模式使能（Send data reload mode enable） 0：关闭； 1：开启。
位 23：16	CNT[7：0]	0x00	rw	发送数据个数（Transmit data counter）
位 15	NACKEN	0x0	rw	不应答使能（Not acknowledge enable） 0：应答；

				1：不应答。
位 14	GENSTOP	0x0	rw	产生停止条件（Generate stop condition） 0：未产生； 1：产生。
位 13	GENSTART	0x0	rw	产生起始条件（Generate start condition） 0：未产生； 1：产生。
位 12	READH10	0x0	rw	10 位地址头读取时序使能（10-bit address header read enable） 0：关闭； 1：开启。
位 11	ADDR10	0x0	rw	主机发送 10 位地址模式使能（Host send 10-bit address mode enable） 0：7 位地址； 1：10 位地址。
位 10	DIR	0x0	rw	主机数据传输方向（Master data transmission direction） 0：发送； 1：接收。
位 9：0	SADDR[9：0]	0x000	rw	主机发送的从机地址（The slave address sent by the master） 当在 7 位地址模式下时 BIT0 以及 BIT[9：8]不关心。

11.7.3 地址寄存器1（I2C_OADDR1）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	res	保持默认值。
位 15	ADDR1EN	0x0	rw	本机地址 1 使能（Own Address 1 enable） 0：关闭； 1：开启。
位 14：11	保留	0x0	res	保持默认值。
位 10	ADDR1MODE	0x0	rw	本机地址 1 模式（Own Address mode） 0：7 位地址； 1：10 位地址。
位 9：0	ADDR1[9：0]	0x000	rw	本机地址 1（Own address 1） 当在 7 位地址模式下时 BIT0 以及 BIT[9：8]不关心。

11.7.4 地址寄存器2（I2C_OADDR2）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	res	保持默认值。
位 15	ADDR2EN	0x0	rw	本机地址 2 使能（Own address 2 enable） 0：关闭； 1：开启。
位 14：11	保留	0x0	res	保持默认值。
位 10：8	ADDR2MASK[2：0]	0x0	rw	本机地址 2 位屏蔽（Own address 2-bit mask） 000：匹配地址位[7：1]； 001：只匹配地址位[7：2]； 010：只匹配地址位[7：3]； 011：只匹配地址位[7：4]； 100：只匹配地址位[7：5]； 101：只匹配地址位[7：6]； 110：只匹配地址位[7]； 111：所有非 I²C 保留地址都会响应。
位 7：1	ADDR2[7：1]	0x00	rw	本机地址 2（Own address 2） 7 位地址。
位 0	保留	0x0	res	保持默认值。

11.7.5 时序寄存器（I2C_CLKCTRL）

域	简称	复位值	类型	功能
位 31：28	DIVL[3：0]	0x0	rw	时钟分频值低 4 位（Low 4 bits of clock divider value）
位 27：24	DIVH[7：4]	0x0	rw	时钟分频值高 4 位（High 4 bits of clock divider value）

				DIV = (DIVH << 4) + DIVL。
位 23：20	SCLD[3：0]	0x0	rw	SCL 输出延时（SCL output delay） T_{SCLD} = (SCLD + 1) x (DIV + 1) x T_{I2C_CLK}。
位 19：16	SDAD[3：0]	0x0	rw	SDA 输出延时（SDA output delay） T_{SDAD} = (SDAD + 1) x (DIV + 1) x T_{I2C_CLK}。
位 15：8	SCLH[7：0]	0x00	rw	SCL 高电平（SCL high level） T_{SCLH} = (SCLH + 1) x (DIV + 1) x T_{I2C_CLK}。
位 7：0	SCLL[7：0]	0x00	rw	SCL 低电平（SCL low level） T_{SCLL} = (SCLL + 1) x (DIV + 1) x T_{I2C_CLK}。

11.7.6 超时寄存器（I2C_TIMEOUT）

域	简称	复位值	类型	功能
位 31	EXTEN	0x0	rw	累计时钟延展超时使能（Cumulative clock low extend timeout enable） 0：关闭； 1：开启。 SMBus 协议里面的 $T_{LOW:SEXT}$ / $T_{LOW:MEXT}$。
位 30：28	保留	0x0	res	保持默认值。
位 27：16	EXTTIME[11:0]	0x000	rw	累计时钟延展超时时间（Cumulative clock low extend timeout value） 超时时间 = (EXTTIME+ 1) x 2048 x T_{I2C_CLK}。
位 15	TOEN	0x0	rw	时钟电平超时检测使能（Detect clock low/high timeout enable） 0：关闭； 1：开启。 SMBus 协议里面的 $T_{TIMEOUT}$。
位 14：13	保留	0x0	res	保持默认值。
位 12	TOMODE	0x0	rw	时钟电平超时检测模式（Clock timeout detection mode） 0：检测低电平； 1：检测高电平。
位 11：0	TOTIME[11:0]	0x000	rw	时钟电平超时检测时间（Clock timeout detection time） 当检测低电平（TOMODE = 0）时： 超时时间 = (TOTIME + 1) x 2048 x T_{I2C_CLK}。 当检测高电平（TOMODE = 1）时： 超时时间 = (TOTIME+ 1) x 4 x T_{I2C_CLK}。

11.7.7 状态寄存器（I2C_STS）

域	简称	复位值	类型	功能
位 31：24	保留	0x00	res	保持默认值。
位 23：17	ADDR[6：0]	0x00	r	从机地址匹配值（Slave address matching value） 7 位地址下：从机接收到的地址。 10 位地址下：从机接收到的 10 位地址头。
位 16	SDIR	0x0	r	从机数据传输方向（Slave data transmit direction） 0：接收数据； 1：发送数据。
位 15	BUSYF	0x0	r	总线忙标志（Bus busy flag transmission mode） 0：空闲； 1：忙。 当检测到 START 条件置起，检测到停止条件自动清零。
位 14	保留	0x00	res	保持默认值。
位 13	ALERTF	0x0	r	SMBus 提醒标志（SMBus alert flag） SMBus 主机：指示提醒信号接收（ALERT 引脚由高变低） 0：未收到； 1：收到。 SMBus 从机：指示设备默认地址接收（0001100x） 0：未收到； 1：收到。

位 12	TMOUT	0x0	r	SMBus 超时标志（SMBus timeout flag） 0：未超时； 1：超时。
位 11	PECERR	0x0	r	PEC 接收错误标志（PEC receive error flag） 0：正确； 1：错误。
位 10	OUF	0x0	r	过载或者欠载标志（Overflow or underflow flag） 当传输方向为发送数据时： 0：正常； 1：欠载。 当传输方向为接收数据时： 0：正常； 1：过载。
位 9	ARLOST	0x0	r	仲裁丢失标志（Arbitration lost flag） 0：正常； 1：仲裁丢失。
位 8	BUSERR	0x0	r	总线错误标志（Bus error flag） 0：正常； 1：错误。
位 7	TCRLD	0x0	r	传输完成，等待加载数据（Transmission is complete, waiting to load data） 0：未完成； 1：已完成。 当数据被发送完成（CNT = 1），并且使能了重载模式（RLDEN=1）时被置起，当写入 CNT 值时自动清零。 在主机模式或者当从机在 SCTRL=1 时使用。
位 6	TDC	0x0	r	数据传输完成标志（Transmit data complete flag） 0：未完成（移位寄存器还有数据）； 1：已完成（移位寄存器空,数据已经完全发送到总线上）。 当在软件结束模式，并且数据传输完成后置起（ASTOPEN = 0，RLDEN = 0，CNT = 0）。 收到 START 或者 STOP 条件后自动清零。
位 5	STOPF	0x0	r	停止条件产生完成标志（Stop condition generation complete flag） 0：未产生； 1：已产生。
位 4	ACKFAILF	0x0	r	应答失败标志（Acknowledge failure flag） 0：正常； 1：失败。
位 3	ADDRF	0x0	r	地址匹配标志（0~7 bit address match flag） 0：未匹配； 1：已匹配。
位 2	RDBF	0x0	r	接收数据缓冲器满（Receive data buffer full flag） 0：数据寄存器（DT）未接收到数据； 1：数据寄存器（DT）接收到数据。
位 1	TDIS	0x0	rw1s	发送中断状态（Transmit data interrupt status） 0：数据已写入 I2C_TXDT； 1：数据已从 I2C_TXDT 发送到移位寄存器，I2C_TXDT 为空，这时必须把要发的数据写到 I2C_ TXDT 寄存器。 在禁止时钟延展模式下，可以向此位写 1，以生成一个 TDIS 事件以便提前写数据到 I2C_TXDT 寄存器。
位 0	TDBE	0x0	rw1s	发送数据寄存器空标志（Transmit data buffer empty flag） 0：I2C_TXDT 有数据，不为空； 1：I2C_TXDT 无数据，为空。 该位只用以表示当前 I2C_TXDT 的状态，并可以通过软件写 1，清除 I2C_TXDT 寄存器里的数据。

11.7.8 状态清除寄存器（I2C_CLR）

域	简称	复位值	类型	功能
位 31：14	保留	0x00000	res	保持默认值。
位 13	ALERTC	0x0	w	清除 SMBus 提醒标志（Clear SMBus alert flag） 写 1 清除。
位 12	TMOUTC	0x0	w	清除 SMBus 超时标志（Clear SMBus timeout flag） 写 1 清除。
位 11	PECERRC	0x0	w	清除 PEC 接收错误标志（Clear PEC receive error flag） 写 1 清除。
位 10	OUFC	0x0	w	清除溢出标志（Clear overload / underload flag） 写 1 清除。
位 9	ARLOSTC	0x0	w	清除仲裁丢失标志（Clear arbitration lost flag） 写 1 清除。
位 8	BUSERRC	0x0	w	清除总线错误标志（Clear bus error flag） 写 1 清除。
位 7：6	保留	0x0	res	保持默认值。
位 5	STOPC	0x0	w	清除停止条件产生完成标志（Clear stop condition generation complete flag） 写 1 清除。
位 4	ACKFAILC	0x0	w	清除应答失败标志（Clear acknowledge failure flag） 写 1 清除。
位 3	ADDRC	0x0	w	清除地址匹配标志（Clear 0~7 bit address match flag） 写 1 清除。
位 2：0	保留	0x0	res	保持默认值。

11.7.9 PEC寄存器（I2C_PEC）

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	res	保持默认值。
位 7：0	PECVAL[7：0]	0x00	r	PEC 值（PEC value）

11.7.10 接收寄存器（I2C_RXDT）

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	res	保持默认值。
位 7：0	DT[7：0]	0x00	r	接收数据寄存器（Receive data register）

11.7.11 发送寄存器（I2C_TXDT）

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	res	保持默认值。
位 7：0	DT[7：0]	0x00	rw	发送数据寄存器（Transmit data register）

12 通用同步异步收发器（USART）

12.1 USART介绍

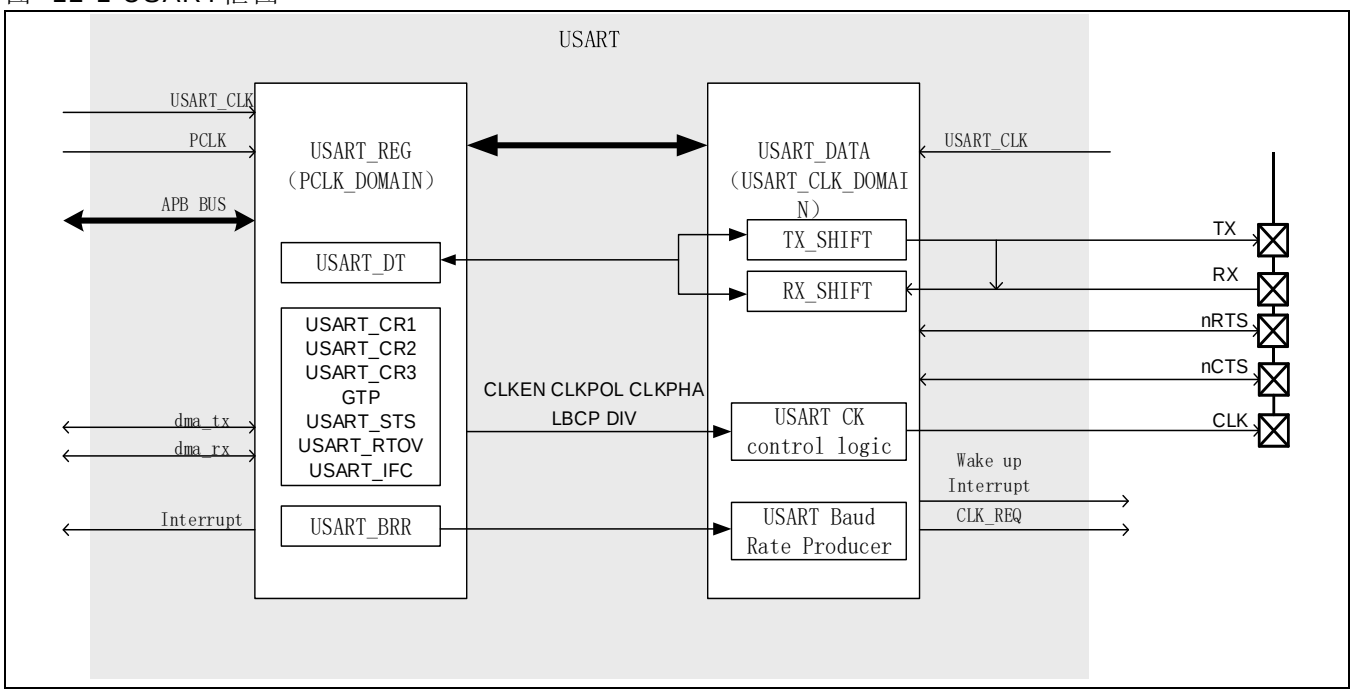
通用同步异步收发器（USART）是一个能通过多种不同的配置与使用不同的数据格式的外设进行通信的通用接口，同时支持异步全双工，异步半双工以及同步传输。USART 提供了可编程的波特率发生器，根据系统频率以及分频系数的不同，用户可以通过配置系统时钟以及分频系数以此产生所需要的特定通信频率。

USART 除了支持标准的 NRZ 异步以及同步收发通信协议外，还支持一些常用的其他类型的串行通信协议，如 LIN(局域互联网)，IrDA（红外数据组织）SIRENDEC 规范，ISO7816-3 标准的异步智能卡协议，以及 CTS/RTS（Clear To Send/Request To Send）硬件流操作，RS485 和 Modbus。

USART 还支持多处理器通信，以及可配置通过空闲帧或地址匹配唤醒的静默模式，以此搭建 USART 网络，并且同时支持使用 DMA 进行数据的收发，以此实现高速通信。

USART 支持双时钟域，PCLK 有系统时钟分频后得到，USART_CLK 来源可以是 PCLK 或 HICK 或 LEXT，这使得 USART 可以工作在 DEEPSLEEP 模式，并支持低功耗唤醒功能。

图 12-1 USART框图



USART 主要特性如下所列：

- 可编程配置的全双工或半双工通信
  —全双工异步通信
  —单线半双工通信
- 可编程配置的通信模式
  —NRZ 标准格式（Mark/Space）
  —LIN（局域互联网）
  —IrDA SIR（串行红外）
  —ISO7816-3 标准里定义的异步智能卡协议： 智能卡模式支持 0.5 或 1.5 个停止位
  —RS-232 CTS/RTS（Clear To Send/Request To Send）硬件流操作
  —RS-485
  —通过静默模式实现多处理器通信（具有地址匹配和总线空闲两种可编程配置的唤醒方式）
  —同步模式
- 可编程配置的波特率发生器
  — 发送和接收共用的可编程波特率
- 可编程配置的帧格式
  —可编程的数据位位数（7 位或 8 位或 9 位）

—可编程的停止位位数-支持 1 或 2 个停止位
—可编程的校验控制：发送方具备发送校验位的能力，接收方具备对接收到的数据进行校验的能力
—可编程配置的数据发送顺序（MSB/LSB）
—可编程配置的 Tx/Rx 引脚极性
—可编程配置的 DT 数据极性

- 可编程配置的 DMA 多缓冲器通信
- 可编程配置的独立的发送器和接收器使能位
- 可编程配置的输出 CLK 的相位和极性以及频率
- 检测标志
  — 接收缓冲器满
  — 发送缓冲器空
  — 传输结束标志
- 四个错误检测标志
  — 溢出错误
  — 噪音错误
  — 帧错误
  — 校验错误
- 可编程配置的 13 个带标志的中断源
  — CTSF 改变
  — LIN 断开符检测
  — 发送数据寄存器空
  — 发送完成
  — 接收数据寄存器满
  — 检测到总线为空闲
  — 溢出错误
  — 帧错误
  — 噪音错误
  — 校验错误
  — 接收器超时检测
  — 字节匹配检测
  — 低功耗唤醒

12.2 全双工半双工选择器简述和配置流程

USART 全双工半双工选择器通过软件编程配置相应寄存器的方式，使得 USART 可以采用全双工或半双工的方式和外设进行数据交换。

USART 默认选择使用双线单向全双工时，TX 引脚用于数据输出，RX 引脚用于数据输入，USART 接收器和发送器相互独立，这使得 USART 可以同时进行数据发送和数据接收，以此实现全双工通信。

USART 在 HALFSEL 位置 1 时选择使用单线双向半双工的方式进行数据通信，在此条件下，LINEN 位，CLKEN 位，SCMEN 位以及 IRDAEN 位需置 0，此时在 USART 内部，RX 引脚无效，TX 引脚和 SW_RX 引脚互连，对 USART 来说，TX 引脚用于数据输出，SW_RX 用于数据输入，对外设来说，数据都从 TX 引脚映射的 IO 双向传输。

12.3 模式选择器简述和配置流程

12.3.1 模式选择器简述

USART 模式选择器通过软件编程配置相应寄存器的方式，使得 USART 可以根据软件的不同配置工作在不同的工作模式下，以此能与使用不同通信协议的外设之间实现数据交换。

USART 默认支持 NRZ 标准格式（Mark/Space），根据 USART 模式选择器配置的不同，USART 还可以支持 LIN（局域互联网），IrDA SIR（串行红外），ISO7816-3 标准里定义的异步智能卡协议，RS-232 CTS/RTS（Clear To Send/Request To Send）硬件流操作以及静默模式和同步模式。

12.3.2 模式选择器配置方法

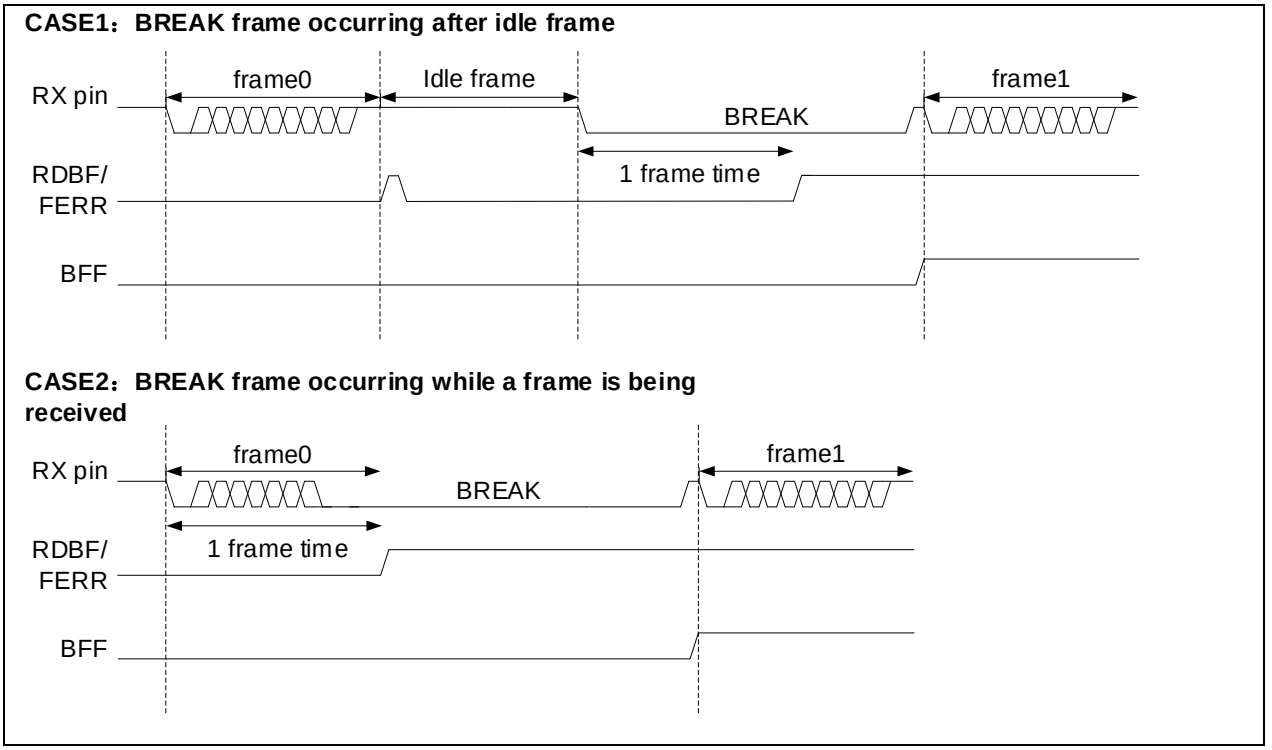
用户可以通过不同的配置以此选择不同的工作模式，配置方法分列如下所列，请将如下配置方法配合本章后述的接收器和发送器配置方法结合使用以完成 USART 初始化配置。

1. **LIN模式**

基础设置：LINEN位置1，CLKEN位置0，STOPBN[1：0]位置0，SCMEN位置0，SLBEN位置0，IRDAEN位置0，DBN[1：0]=00。

LIN主机有发送间隔帧的能力，可以使用SBF位置1发送13位低电平的LIN同步间隔帧。同时LIN从机也有检测间隔帧的能力，可以选择BFBN位置1或0来选择是11位还是10位间隔帧检测。

图 12-2 LIN模式下的BFF检测与FERR检测



2. **智能卡模式**

基础设置：SCMEN位置1，LINEN位置0，SLBEN位置0，IRDAEN位置0，CLKEN位置1，DBN[1：0]=01，PEN位置1，STOPBN[1：0]=11。

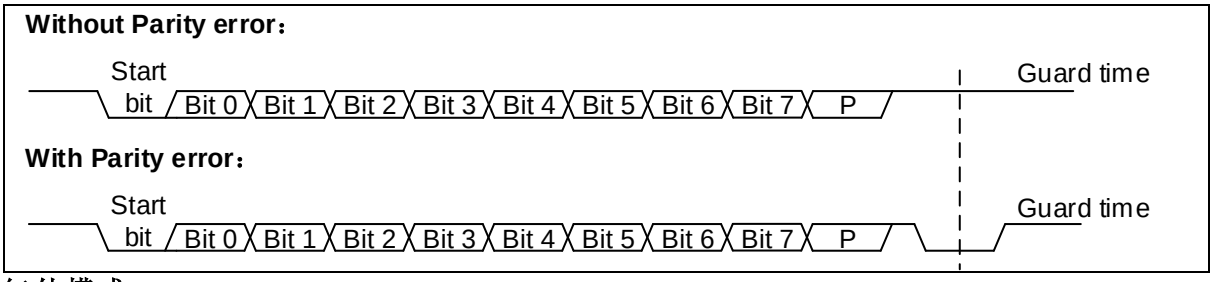
可以选择配置CLKPOL位和CLKPHA位以及LBCP位以满足不同的时钟极性以及时钟相位和时钟脉冲个数，具体可见同步模式部分。

通过配置SCGT[7：0]位选择保护时间，使TDC标志的置起可以得到延时，直到保护时间计数器向上计数到SCGT[7：0]的值，TDC才得以置起。

而智能卡属于单线双向半双工通信，可以通过配置SCNACKEN位选择是否在校验出错时发送NACK，以告知数据没有被正确接收。

注意：发送和接收时，都只可配置为1.5个停止位，且必须确保智能卡保护时间皆满足1.5位长才能正常接收连续数据。

图 12-3 Smartcard frame format



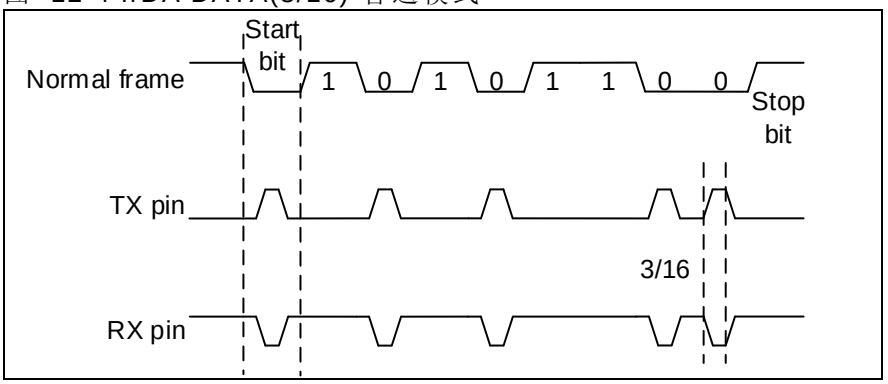
3. **红外模式**

基础设置：IRDAEN位置1，CLKEN位置0，STOPBN[1：0]位置0，SCMEN位置0，SLBEN

位置0。

可以选择IRDALP位置1以开启红外低功耗模式，在普通模式下持续时间为3/16位，在红外低功耗模式下位持续时间可调，并配合ISDIV[7：0]配置想要产生的低功耗频率。

图 12-4 IrDA DATA(3/16)-普通模式



4. **Modbus**

USART仅提供实现Modbus/RTU和Modbus/ASCII所需的基础的硬件支持，这意味着协议的控制部分必须由软件完成，USART只是提供块结束侦测。

在Modbus/RTU中，块结束侦测通过可编程的超时功能识别接收线为空闲状态的时间大于2个字节时间实现，用户可以通过配置RTOV寄存器设定需要的超时值（时间单位是1bit位宽），通过RTODEN位置1开启超时检测，当USART接收器检测到接收线为空闲的时间等于设定的超时值时，USART会置起RTODF，如果RTODIE位置1，产生中断，可以通过RTODCF位写1清除RTODF位；

在Modbus/ASCII中，块结束的侦测通过字节匹配功能识别特殊的字节序列（CR/LF），通过软件编程将LF ASCII码写入ID[7：0],通过CMDIE位置1开启字节匹配功能，当USART接收器接收到的数据和ID[7：0]匹配时，USART会置起CMDF,如果CMDIE位置1，产生中断，可以通过CMDCF位写1清除CMDF位。

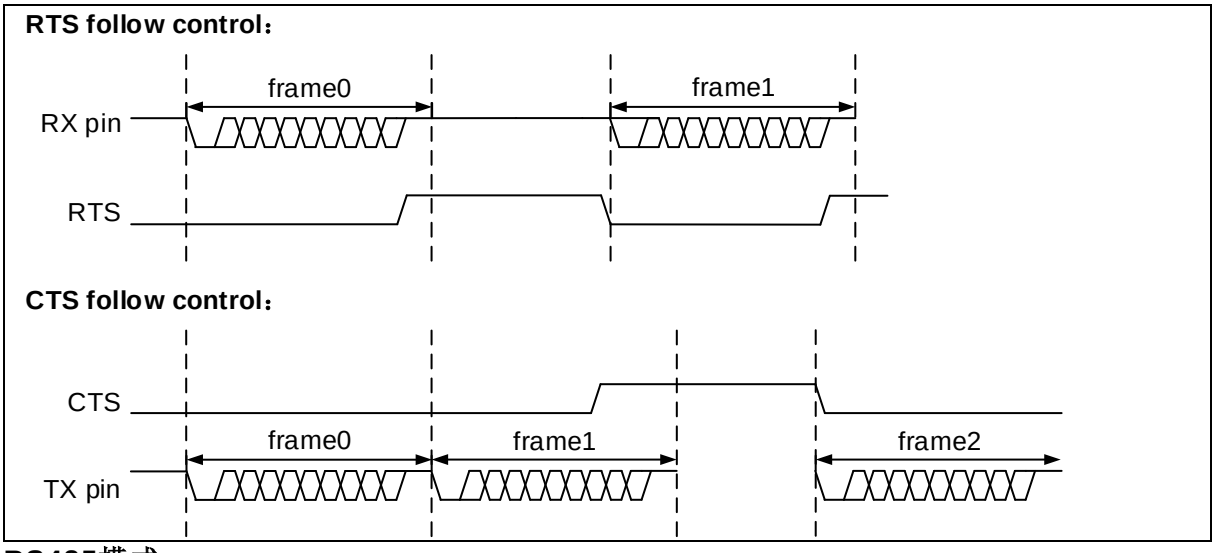
5. **硬件流控制模式**

RTSEN位置1和CTSEN位置1可以分别开启RTS和CTS流控制。

RTS流控制：USART接收器准备好接收新的数据，RTS就变成有效（下拉为低电平）。当接收寄存器内有数据到达时（在每个stop位开始时），RTS被置位，由此表明希望在当前帧结束时停止数据传输。

CTS流控制：USART发送器在发送下一帧前检查CTS输入。如果CTS有效（也即CTS为低电平），则下一个数据被发送；若CTS在传输期间被变成无效(也即CTS为高电平)，当前的传输完成后停止发送。

图 12-5 Hardware flow control



6. **RS485模式**

RS485EN位置1开启RS485模式，驱动使能信号从RTS引脚输出，用户可以通过配置DEP位选择DE信号的极性，用户可以通过分别配置TSDT[4：0]和TCDT[4：0]设置发送器开始

发送起始位前延迟的时间，和发送器发送完最后一笔数据的停止位后置起TC标志前延迟的时间。

7. 静默模式

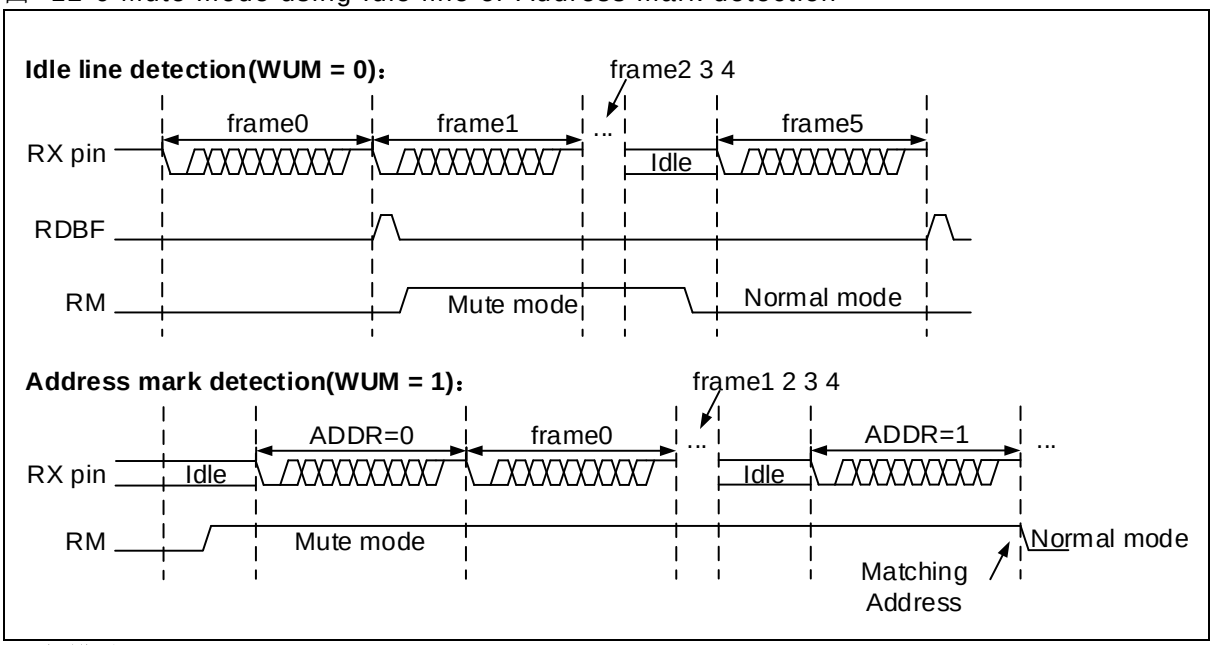
RM位置1进入静默模式，根据WUM位置1和置0，可以分别通过ID匹配和空闲总线从静默模式中唤醒，其中ID号ID[7：0]可编程配置，并且可以通过配置IDBN选择使用ID[7：:0]或ID[3：0]，当选择ID匹配时，数据位的MSB为1表示当前数据是ID。

关闭奇偶校验功能时，当DBN[1：0]=10时，MSB是USART_DT[6]，当DBN[1：0]=00时，MSB是USART_DT[7]，当DBN[1：0]=01时，MSB是USART_DT[8]。

开启奇偶校验功能时，当DBN[1：0]=10时，MSB是USART_DT[5]，当DBN[1：0]=00时，MSB是USART_DT[6]，当DBN[1：0]=01时，MSB是USART_DT[7]。

当选择ID[3：0]时，数据位的4个LSB表示ID值；当选择ID[7：0]时，数据位除上述奇偶校验位以及MSB位以外，所有的LSB位表示ID值。

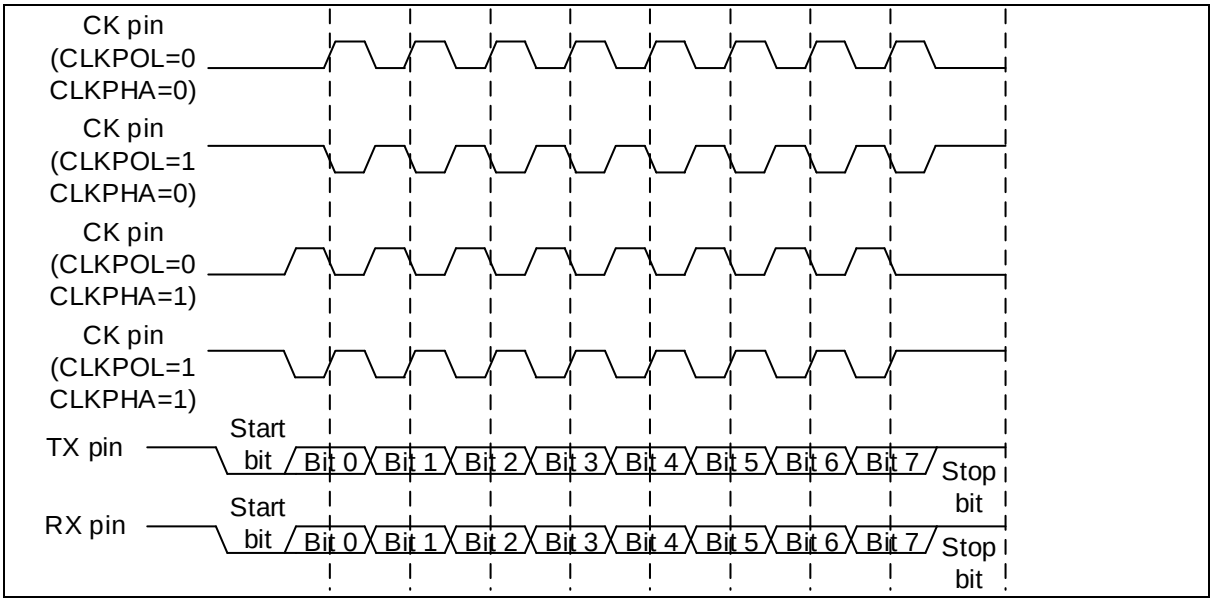
图 12-6 Mute mode using Idle line or Address mark detection



8. 同步模式

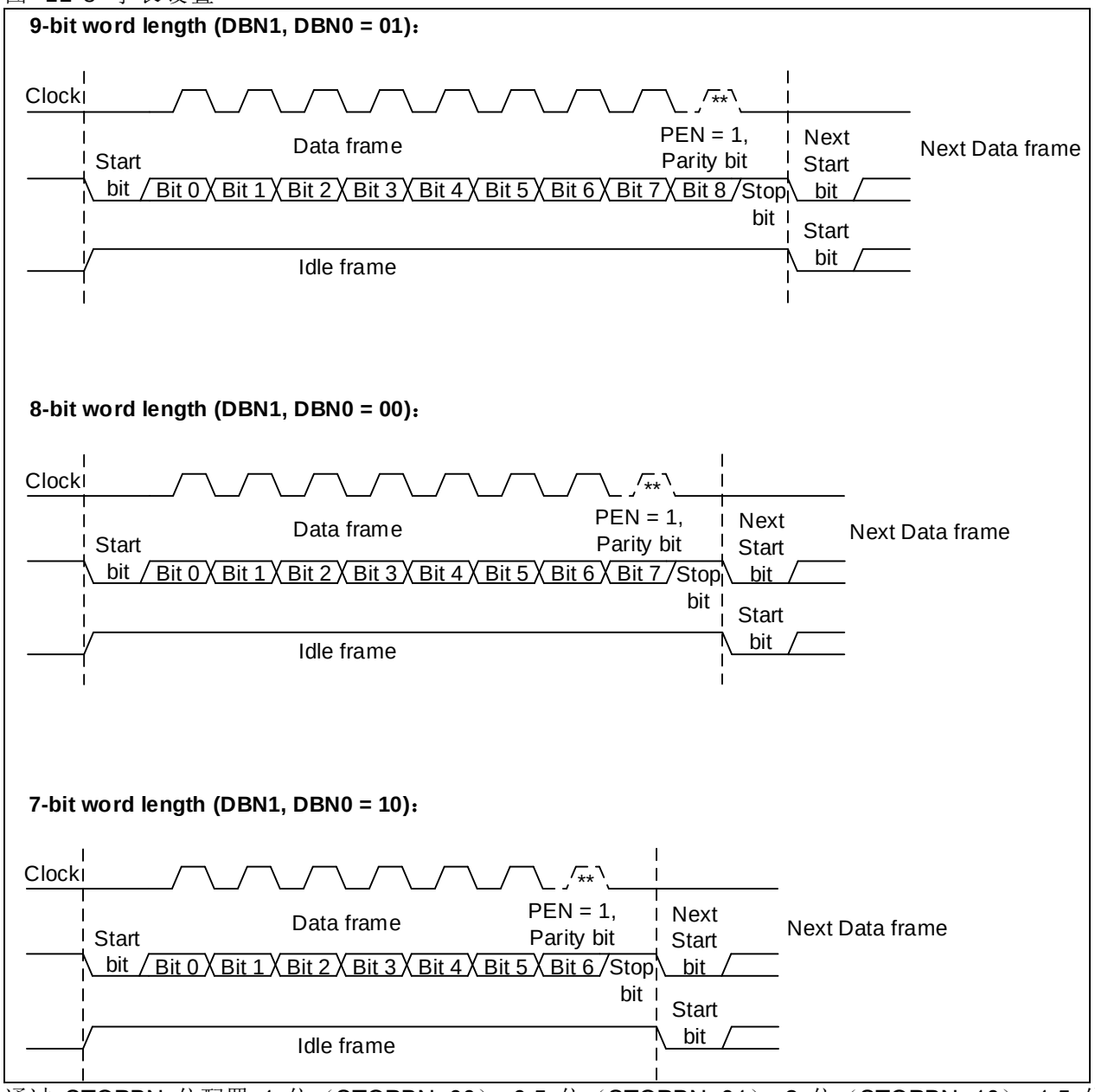
CLKEN位置1开启同步模式并使能时钟引脚输出，通过配置CLKPOL位置1或0可以选择空闲状态下CK引脚上的电平为高或低，通过配置CLKPHA位置1或0可以选择在时钟的第二个或第一个边沿开始采样数据，通过配置LBCP位置1或0可以选择最后一位数据是否输出时钟，通过配置ISDIV[4：0]可以选择想要输出的时钟频率。

图 12-7 8-bit format USART同步模式



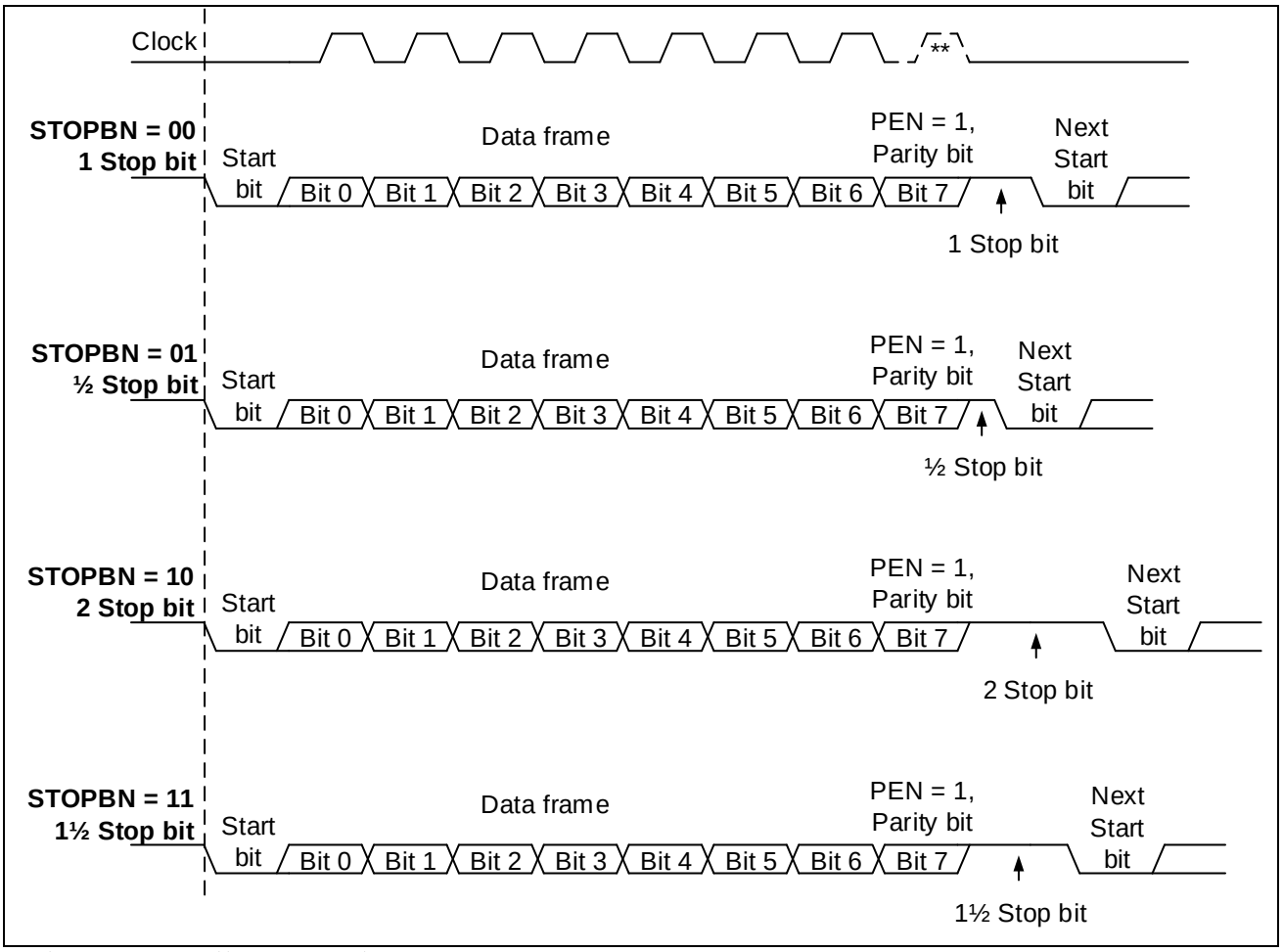
12.4 USART帧格式简述和配置流程

USART 一笔数据帧由起始位，数据位，停止位依次组成,最后一位数据位可以作为校验位。
USART 一笔空闲帧的长度等于当前配置下数据帧的长度，但所有位都为 1。
USART 一笔断开帧的长度等于当前配置下数据帧的长度加上停止位，停止位之前的所有位都等于 0。需要特别注意的是，在非 LIN 模式下， 发送和检测断开帧的长度都需遵守此规则，例 DBN[1：0]=00，那么发送和检测的断开帧长度就是 10 位低电平加停止位。LIN 模式请参考模式选择器简述和配置流程部分。
通过 DBN1 位和 DBN0 配置 7 位（DBN[1：0]=10）8 位（DBN[1：0]=00）或 9 位（DBN[1：0]=01）数据位。
图 12-8 字长设置



通过 STOPBN 位配置 1 位（STOPBN=00），0.5 位（STOPBN=01），2 位（STOPBN=10），1.5 位（STOPBN=11）停止位。
通过 PEN 位置“1”配置校验控制使能，通过 PSEL 位配置奇校验（PSEL=1）或偶检验（PSEL=0），校验控制使能后数据位的 MSB 将由奇偶校验位替代，即有效数据位减少一位。

图 12-9 配置停止位



通过 MTF 位配置数据是先传输 MSB（MTF=1）还是 LSB（MTF=0）。
通过 DTREV 位配置 USART_DT 是以 1=L,0=H（DTREV=1）还是 0=L,1=H（DTREV=0）的方式发送和接收。
通过 TXREV 位配置 USART_TX 引脚上的信号是以 VDD=0/mark,Gnd=1/idle（TXREV=1）还是 VDD=1/idle,Gnd=0/mark（TXREV=0）的方式传输。
通过 RXREV 位配置 USART_RX 引脚上的信号是以 VDD=0/mark,Gnd=1/idle（RXREV=1）还是 VDD=1/idle,Gnd=0/mark（RXREV=0）的方式传输。

12.5 DMA传输简述和配置流程

USART 可以使用 DMA 操作发送数据缓冲器和接收数据缓冲期以实现高速连续传输，USART 的 DMA 传输需要配合 DMA 使用，下方会简述配置流程，但具体和 DMA 配置相关部分请参见 DMA 章节的描述。

12.5.1 DMA发送配置流程

1. 选择DMA传输通道：在DMA章节DMA通道映射表中选择用于当前所用USART的DMA通道。
2. 配置DMA传输目标地址：在DMA控制寄存器中DMA传输目的地址位写入当前所使用的USART的USART_DT寄存器地址，DMA将会在接收到发送请求后将代发送的数据写入该地址。
3. 配置DMA传输源地址：在DMA控制寄存器中DMA传输源地址位写入代发送数据存放的地址，DMA将会在接收到发送请求后将该地址内的数据写入到目标地址中，即写入到当前所使用的USART的USART_DT寄存器中。
4. 配置DMA传输字节个数：在DMA控制寄存器相关位置配置期望传输的字节个数
5. 配置DMA传输通道优先级：在DMA控制寄存器相关位置配置当前所使用通道的USART的DMA传输通道优先级。
6. 配置DMA中断产生时机：在DMA控制寄存器相关位置配置是在传输完成或传输完成一半时产生DMA中断。

7. 使能DMA传输通道：在DMA控制寄存器相关位置使能当前所选用的DMA通道。

12.5.2 DMA接收配置流程

1. 选择DMA传输通道：在DMA章节DMA通道映射表中选择用于当前所用USART的DMA通道。
2. 配置DMA传输目标地址：在DMA控制寄存器中DMA传输目的地址位写入期望存放接收数据的地址，DMA将会在接收到接收请求后，将当前所使用的USART的USART_DT寄存器中的数据存放在目的地址中。
3. 配置DMA传输源地址：在DMA控制寄存器中DMA传输源地址位写入当前所使用的USART的USART_DT寄存器的地址，DMA将会在接收到接收请求后将该地址内的数据写入到目标地址中，即写入到期望存放接收数据的地址。
4. 配置DMA传输字节个数：在DMA控制寄存器相关位置配置期望传输的字节个数
5. 配置DMA传输通道优先级：在DMA控制寄存器相关位置配置当前所使用通道的USART的DMA传输通道优先级。
6. 配置DMA中断产生时机：在DMA控制寄存器相关位置配置是在传输完成或传输完成一半时产生DMA中断。
7. 使能DMA传输通道：在DMA控制寄存器相关位置使能当前所选用的DMA通道。

12.6波特率发生器简述及配置流程

12.6.1 波特率发生器简述

USART 波特率发生器通过使用内部计数器，以 PCLK 为基准，DIV（USART_BAUDR[15：0]）即为该计数器的溢出值，该计数器计满一次代表一位数据，所以每位数据位宽为 DIV 个 PCLK 周期。

由于 USART 的接收器和发送器共用同一个波特率发生器，并且接收器将每位数据拆分为 16 份等长的部分以此来实现过采样，所以数据位宽不得小于 16 个 PCLK 周期，即 DIV 中的值必须大于或等于 16。

12.6.2 波特率发生器配置方法

用户可通过配置不同的系统时钟以及在 USART_BAUDR 中写入不同的值以此产生特定的波特率，具体的运算关系见如下公式

$$\text{TX/RX 波特率} = \frac{f_{CK}}{\text{DIV}}$$

这里的f_{CK}是指 USART 的系统时钟（PCLK1 用于 USART2、3 和 USART4、5、7、8，PCLK2 用于 USART1、6）

注意：1.USART_BAUDR 中的值需要在 UEN 之前写入，且 UEN=1 时，不可更改这些位。

2.关闭 USART 接收器或发送器会使内部计数器复位，波特率发生中断。

表 12-1 设置波特率时的误差计算

波特率		fPCLK=36MHz			fPCLK=72MHz		
序号	Kbps	实际	置于波特率寄存器中的值	误差%	实际	置于波特率寄存器中的值	误差%
1	2.4	2.4	15000	0%	2.4	30000	0%
2	9.6	9.6	3750	0%	9.6	7500	0%
3	19.2	19.2	1875	0%	19.2	3750	0%
4	57.6	57.6	625	0%	57.6	1250	0%
5	115.2	115.384	312	0.15%	115.2	625	0%
6	230.4	230.769	156	0.16%	230.769	312	0.16%
7	460.8	461.538	78	0.16%	461.538	156	0.16%
8	921.6	923.076	39	0.16%	923.076	78	0.16%
9	2250	2250	16	0%	2250	32	0%
10	4500	不可能	不可能	不可能	4500	16	0%

以波特率 115.2Kbps 为例，假设 fPCLK 为 36MHz，此时波特率寄存器应设置为 312(0x138)，经由公式计算：36000000 / 312 = 115384 = 115.384Kbps

而它们的误差计算为(实际值 - 理论值) / 理论值 * 100%：(115.384 - 115.2) / 115.2 * 100% = 0.15%

12.7发送器简述和配置流程

12.7.1 发送器简述

USART 发送器具有独立的使能位 TEN，发送器与接收器共用同一个波特率且该波特率可编程配置，USART 具有一个发送数据缓冲器（TDR）和一个发送移位寄存器，当发送数据缓冲器（TDR）为空时，TDBE 置起，如果设置了 TDBEIEN 将会产生中断。

软件写入的值会先存储在发送数据缓冲器（TDR）中，当发送移位寄存器为空时，USART 会将发送数据缓冲器中的值移入到发送移位寄存器，USART 发送器将以 LSB 的方式将发送移位寄存器中的数据从 TX 脚输出，具体的输出格式取决于软件配置的帧格式。

如若选择了同步传输或者配置了时钟输出，USART 发送器将时钟脉冲从 CK 脚输出，如若选择了硬件流控制，USART 发送器将控制信号将从 CTS 引脚输入。

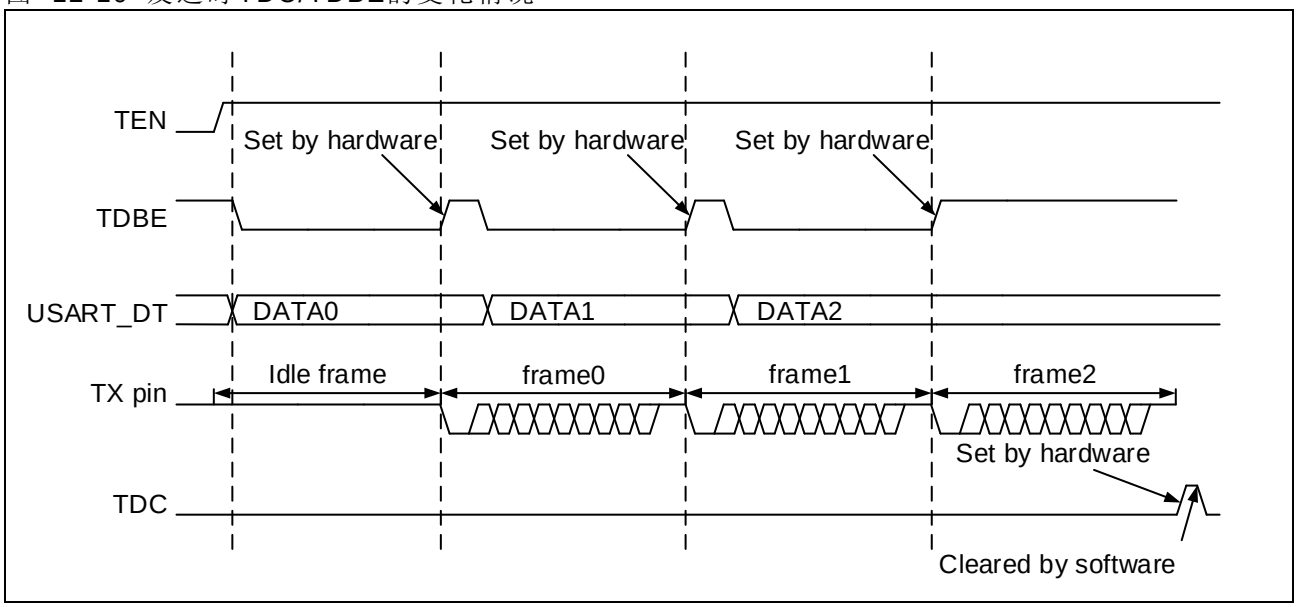
注意：1. 在数据传输期间不能复位 TEN 位，否则将破坏 TX 脚上的数据。

2. TEN 位被激活后,USART 将自动发送一个空闲帧。

12.7.2 发送器配置流程

1. USART使能：UEN位置1。
2. 全双工半双工配置：具体参见全双工半双工选择器配置部分（12.2）。
3. 模式配置：具体参见模式选择器配置部分（12.3）。
4. 帧格式配置：具体参见帧格式配置部分（12.4）。
5. 中断配置：具体参见中断发生器配置部分（12.11）。
6. DMA发送配置：如果选择使用DMA发送，DMATEN位USART_CTRL3[7])置1，并按照DMA传输中的描述配置DMA寄存器。
7. 波特率配置：具体参见波特率发生器配置部分（12.6）。
8. 发送器使能：TEN位置1，置1后USART发送器会自动发送一个空闲帧。
9. 数据写入：等待TDBE位置起后，将要发送的数据写入USART_DT寄存器（此操作会清除TDBE位），在非DMA模式下，重复此操作。
10. 在写入最后一个期望传输的数据后，等待TDC位置起，这表示最后一个数据帧的传输结束，在该标志置起前，禁止关闭USART，否则传输可能出错。
11. 在TDC=1后，可以采用先读一次USART_STS寄存器，再写一次USART_DT寄存器的方式来清除TDC；也可以采用软件对它写’0’来清除，但此方法只推荐在DMA模式下使用。

图 12-10 发送时TDC/TDBE的变化情况



注意：USART 连续数据发送时，两笔数据之间固定存在 2 个 USART_CLK 周期的空闲电平时间。

*以 USART clock = 72MHz 为例，clock period = 13.88ns，上一笔数据停止位传输完成后，TX pin 会再经历 13.88 * 2 = 27.76ns 的空闲时间才发送下一笔数据。*

12.8接收器简述和配置流程

12.8.1 接收器简述

USART 接收器具有独立的接收器使能位 REN(USART_CTRL1[2])，接收器和发送器共用同一个波特率且该波特率可编程配置，USART 具有一个接收数据缓冲器（RDR）和一个接收移位寄存器。
数据从 USART 的 RX 脚输入，当接收器判断到一个有效的起始位后，接收器会以 LSB 的方式将接收到的数据依次移入接收移位寄存器，并根据软件配置的帧格式，在接收到一个完整的数据帧后将接收移位寄存器中的值移入接收数据缓冲器并置起 RDBF，如果设置了 RDBFIEN 将会产生中断。
如若选择了硬件流控制，USART 接收器将控制信号将从 RTS 引脚输出。
在数据接收过程中，USART 接收器会根据软件的配置检测帧错误，溢出错误，奇偶校验错误以及噪声错误，并根据相应的中断使能位是否置位来判断是否产生相应的中断。

12.8.2 接收器配置流程

配置步骤：

1. USART使能：UEN位置1。
2. 全双工半双工配置：具体参见全双工半双工选择器配置部分（12.2）。
3. 模式配置：具体参见模式选择器配置部分（12.3）。
4. 帧格式配置：具体参见帧格式配置部分（12.4）。
5. 中断配置：具体参见中断发生器配置部分（12.11）。
6. DMA接收配置：如果选择使用DMA接收，DMAREN位置1，并按照DMA传输中的描述配置DMA寄存器。
7. 波特率配置：具体参见波特率发生器配置部分（12.6）。
8. 接收器使能：REN位置1。

当一字符被接收到时：

- RDBF 位被置位。它表明移位寄存器的内容被转移到 RDR（Receiver Data Register）。换句话说，数据已经被接收并且可以被读出（包括与之有关的错误标志）。
- 如果 RDBFIEN 位被设置，则产生中断。
- 在接收期间如果检测到帧错误，噪音或溢出错误，错误标志将被置起。
- 在 DMA 传输时，RDBF 在每个字节接收后被置起，并由 DMA 对数据寄存器的读操作而清零。
- 在非 DMA 传输时，由软件读 USART_DT 寄存器完成对 RDBF 位清除。RDBF 标志也可以通过对它写 0 来清除。RDBF 位必须在下一帧数据接收结束前被清零，以避免溢出错误。

当一个断开帧被接收到时：

- 非 LIN 模式：USART 接收器按照帧错误处理，并置起 FERR 位，若相应中断使能，中断产生，具体可见下方错误帧的描述。
- LIN 模式：USART 接收器按断开帧处理，并置起 BFF 位，若 BFIEN 置位，则中断产生。

当一个空闲帧被接收到时：

- USART 接收器按数据帧处理，并置起 IDLEF 位，若 IDLEIEN 置位，则中断产生。

当一个帧错误产生时：

- FERR 位置位。
- USART 接收器将错误的数据从接收移位寄存器转移到接收数据缓冲器。
- 在非 DMA 传输时，这个位和 RDBF 位同时置起，后者将产生中断。在 DMA 传输时，如果 ERRIEN 置位的话，将产生中断。

当一个溢出错误产生时：

- ROERR 位被置位。
- 接收数据缓冲器中的数据不会被覆盖，读 USART_DT 寄存器仍能得到先前的数据。
- 接收移位寄存器中的内容会被覆盖，随后接收到的数据都将丢失。
- 如果 RDBFIEN 位置位或 ERRIEN 和 DMAREN 位都被置位，中断产生。
- 先读 USART_STS，再读 USART_DT 寄存器，可清除 ROERR。

注意： 当 ROERR 置位时，表明至少有 1 个数据已经丢失。有两种可能性：

- *如果 RDBF=1，上一个有效数据还存储在接收数据缓冲器中，可以被读出。*
- *如果 RDBF=0，这意味着上一个有效数据已经从接收数据缓冲器中读走。*

注意：在接收数据时，REN 位不应该被复位。如果 REN 位在接收时被清零，当前字节的接收被丢失。

12.8.3 起始侦测和噪声检测

USART 接收器在 REN 位置位后便开始侦测起始位，USART 接收器通过过采样技术，在第 3、5、7、8、9、10 位共 6 个点进行数据采样，以此侦测有效起始位以及识别噪声，具体的噪声和有效起始位的判别方式可以参见下方检测起始位和噪声的数据采样。

表 12-2 检测起始位和噪声的数据采样

采样值（3 • 5 • 7）	采样值（8 • 9 • 10）	NERR 位	起始位有效性
000	000	0	有效
001/010/100	001/010/100	1	有效
001/010/100	000	1	有效
000	001/010/100	1	有效
111/110/101/011	任意值	0	无效
任意值	111/110/101/011	0	无效

注意：如果在第 3、5、7、8、9、10 位的采样值满足不了上表任意一种组合，则 USART 接收器认为没有接受到正确的起始位，将退出起始位侦测并回到空闲状态等待下降沿。

USART 接收器具备噪声检测功能，在非同步模式时，使用过采样技术，在第 7、8、9 采样点，根据不同的采样值，区别有效输入数据和噪音，并恢复数据和置起噪声错误标志位 NERR。具体的采样方法以及噪声和有效数据的判别方式可以参见下方检测有效数据和噪声的数据采样。

表 12-3 检测有效数据和噪声的数据采样

采样值	NERR 位	接收的位	数据有效性
000	0	0	有效
001	1	0	无效
010	1	0	无效
011	1	1	无效
100	1	0	无效
101	1	1	无效
110	1	1	无效
111	0	1	有效

USART 接收器在最大允许偏差下，皆可以正常接收数据，其值取决于 USART_CTRL1 的 DBN[1：0]以及 USART_BAUDR 的 DIV[3：0]。

注意：以下表格的最大允许偏差是以波特率 115.2Kbps 为基准进行计算，实际接收器最大允许偏差会随着波特率设定大小有所改变，波特率越大时其最大允许偏差会越小，反过来波特率越小其最大允许偏差会越大。

表 12-4 最大允许偏差

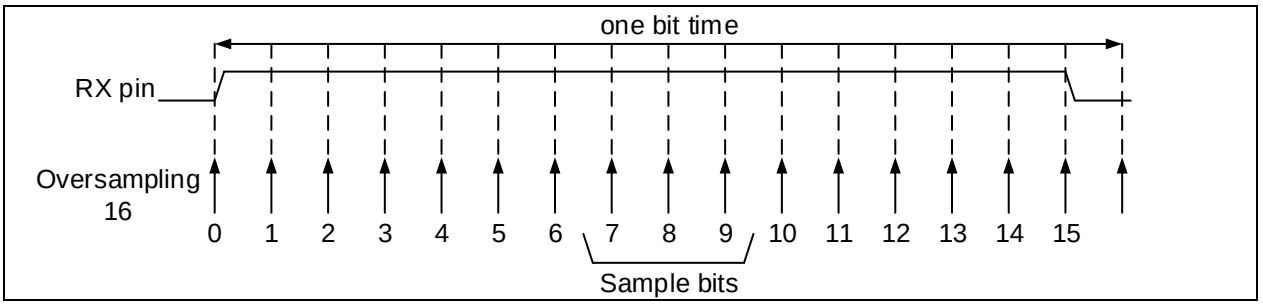
DBN[1:0]	DIV[3:0] = 0	DIV[3:0] != 0
00	3.75%	3.33%
01	3.41%	3.03%
10	4.16%	3.7%

当 USART 接收器在数据帧中检测到噪音时：

- 在 RDBF 位置起的同时置起 NERR 位。
- USART 接收器将错误数据从接收移位寄存器转移到接收数据缓冲器。
- 在非 DMA 传输时，没有噪声中断产生。然而，因为 NERR 位和 RDBF 位是同时置位，RDBF 将产生中断。在 DMA 传输时，如果 ERRIEN 位置位，中断产生。

先读 USART_STS，再读 USART_DT 寄存器，将清除 NERR 位。

图 12-11 检测噪声的数据采样



12.9低功耗唤醒简述和配置流程

USART 支持低功耗唤醒功能，在进入 DEEPSLEEP 模式之前，软件需要保证 USART_CLK 的时钟来源为 HICK 和 LEXT，并需要通过 OCCUPY 位判断 USART 此时没有进行传输，再通过 RXON 位确认 USART 接收器已初始化完成，最后需要置位 SMUSEN=1 以使能 DEEPSLEEP 模式下的 USART。
USART 在进入 DEEPSLEEP 模式后 USART_CLK 会关闭，USART 会侦测接收线上的下降沿，一旦有下降沿被侦测到，USART 会请求 MCU 打开 USART_CLK，USART_CLK 会至少持续到 USART 回到空闲状态，在此期间如果侦测到了唤醒源，USART 会产生中断唤醒 MCU，如果无唤醒源被侦测到，USART 请求 MCU 关闭 USART_CLK 并等待下一个下降沿。
USART 根据 LPWUM[1:0]不同的配置有三种唤醒方式，ID 匹配（LPWUM=00），起始位唤醒（LPWUM=10），RDBF 标志位唤醒（LPWUM=11），USART 在 DEEPSLEEP 模式期间如果检测到了设定的唤醒源，会置位 LPWUF 位，如果 LPWUFIE 位置位，则产生中断，需要注意此中断仅 DEEPSLEEP 模式有效，需要另外特别指出，如果选择 RDBF 标志位唤醒，也可以通过置位 RDBFIE 位使能中断。
由于进入 DEEPSLEEP 模式后系统时钟关闭，所以也需要软件提前配置唤醒方式，以及置位相应的中断使能位。

USART 处于静默模式下进入 DEEPSLEEP 模式需要注意：

1. *不能使用空闲总线唤醒静默模式。*
2. *如果使用ID匹配唤醒静默模式，那MCU低功耗唤醒方式应选择ID匹配。如果RDBF在进入DEEPSLEEP模式前置位，即使ID匹配，MCU退出DEEPSLEEP模式，但USART仍然处于静默模式。*
3. *如果使用起始位唤醒MCU退出DEEPSLEEP模式，LPWUF会置位，但RDBF不会置位。*

注意：USART 唤醒 DEEPSLEEP 时，软件除需清除 USART 的标志位外，还需要清除 EXINT 的 pending 标志；
USART 唤醒 DEEPSLEEP 时，推荐 EXINT 配置上升沿触发，若配置使用双边沿触发，需要独立清除 EXINT pending 标志。

注意：使用 USART 唤醒 DEEPSLEEP，对 USART 波特率有最大限制，取决于唤醒时间参数和 USART 最大允许偏差(表 12.4)。
以 DBN[1：0] = 01，DIV[3：0] = 0000 为例，接收器最大允许偏差为 D_{WUmax}，唤醒时间参数则是 $t_{WULPRUN}$，根据以下公式可计算出最大波特率：
如使用 HICK 用作 USART_CLK，需考虑 HICK 不精确的情况，HICK 不精确度常温下为 1%
$t_{WULPRUN}$ = 13.25us(硬件设计仿真值)
D_{WUmax} = 3.41% - 1% = 2.41%
$T_{bit\ min}$ = 13.25us / (11 x 2.41%) = 49.98us

波特率 = 1 / 49.98us ~= 20kbps。

12.10 Tx/Rx可配置引脚互换

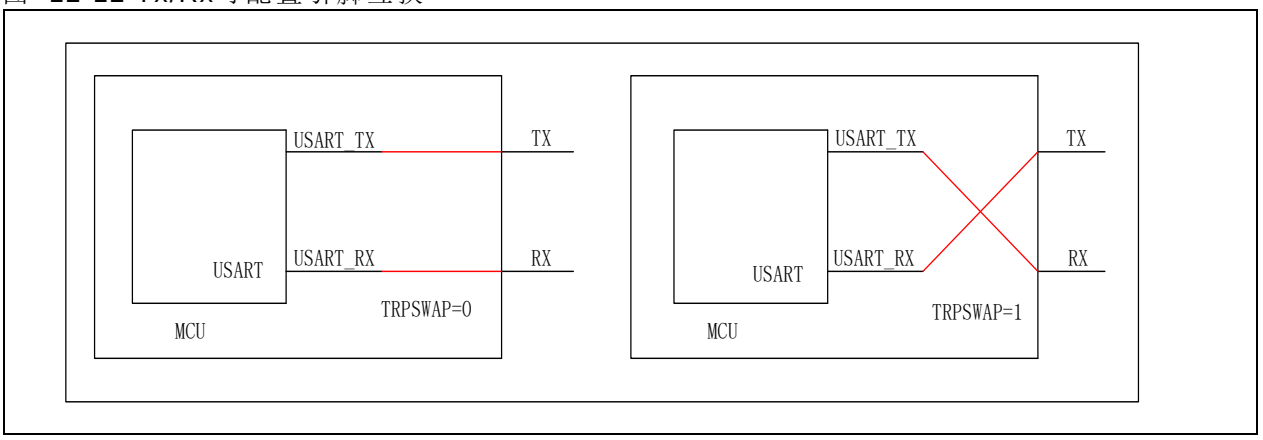
如果TRPSWAP（USART_CTRL2[15]）位被使能，MCU的Tx/Rx引脚顺序将被交换。以下举例两种常见应用场景：

- 若用户在外接 RS-232 芯片时不慎将 Tx/Rx 接反，可通过修改 TRPSWAP 位更换引脚顺序，无

需修改硬件连接。

- 若用户在全双工模式下只将主机的 Tx 和从机的 Rx 连接，在主机和从机互换后，也可通过 TRPSWAP 位更换引脚顺序，无需修改硬件连接。

图 12-12 Tx/Rx可配置引脚互换



注意：SWAP（USART_CTRL2[15]）位必须在 USART 未被使能（UEN=0）时才能被改写

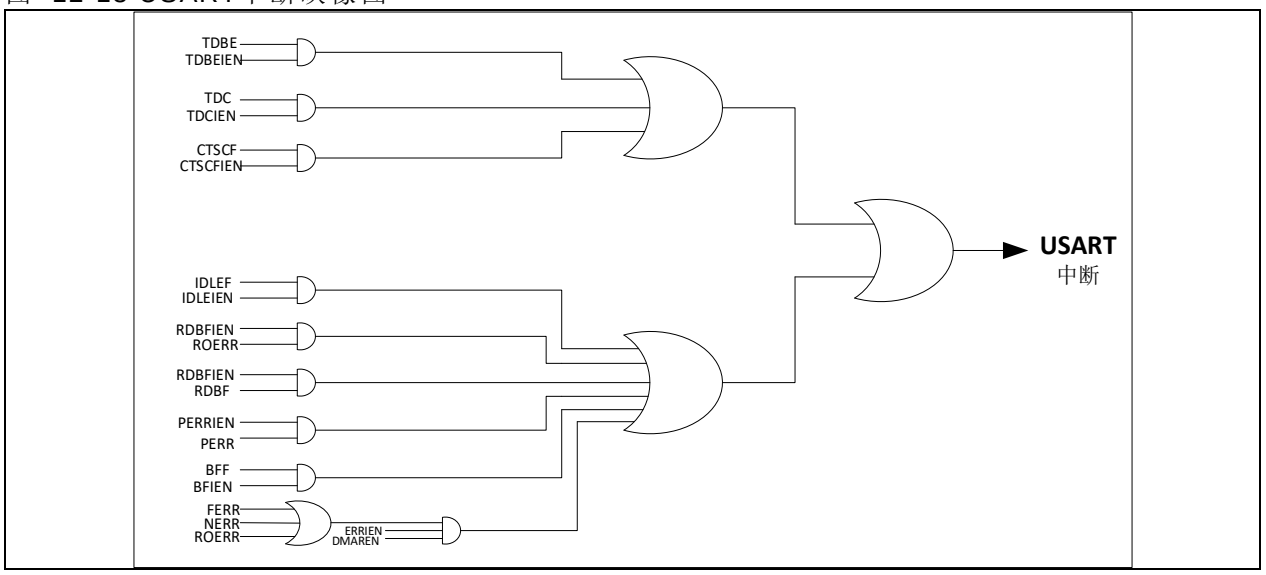
12.11 中断

USART 中断发生器是 USART 中断的控制中枢，USART 中断产生器会实时监测 USART 内部的中断源，并根据软件配置的相应中断源的中断使能位，以此决定是否产生中断，下表所示为 USART 的中断源以及相应的中断使能位，对相应的中断使能位置 1 时，即可在相应事件出现后产生中断。

表 12-5 USART中断请求

中断事件	事件标志	使能位
发送数据寄存器空	TDBE	TDBEIEN
CTS 标志	CTSCF	CTSCFIEN
发送完成	TDC	TDCIEN
接收数据就绪可读	RDBF	RDBFIEN
检测到数据溢出	ROERR	
检测到空闲线路	IDLEF	IDLEIEN
奇偶检验错	PERR	PERRIEN
断开标志	BFF	BFIEN
噪声标志，多缓冲通信中的溢出错误和帧错误	NERR 或 ROERR 或 FERR	ERRIEN（1）

图 12-13 USART中断映像图



12.12 I/O引脚控制

USART 通过五个接口外部设备进行通信，引脚定义如下：
RX：串行数据输入端。
TX：串行数据输出端。在单线半双工模式和智能卡模式里，TX 脚作为 I/O 使用，即用于发送数据也用于接收数据。
CK：发送器时钟输出。输出的 CLK 相位和极性以及频率均可编程配置。
CTS：发送器输入端，硬件流控制模式发送使能信号。
RTS：接收器输出端，硬件流控制模式发送请求信号。

12.13 USART寄存器描述

表 12-6 USART寄存器映像和复位值

寄存器简称	基址偏移量	复位值
USART_STS	0x00	0x00C0
USART_DT	0x04	0x0000
USART_BAUDR	0x08	0x0000
USART_CTRL1	0x0C	0x0000
USART_CTRL2	0x10	0x0000
USART_CTRL3	0x14	0x0000
USART_GDIV	0x18	0x0000
USART_RTOV	0x1C	0x0000
USART_IFC	0x20	0x0000

12.13.1状态寄存器（USART_STS）

域	简称	复位值	类型	功能
位 31：23 位 19：18 位 15：12 位 10	保留	0x000000	resd	硬件强制为 0。
位 22	RXON	0		接收器使能标志 0：接收器未完成使能 1：接收器已完成使能 注：USART4 至 USART8 不支持此位，默认为零。
位 21	TXON	0		发送器使能标志 0：发送器未完成使能 1：发送器已完成使能 注：USART4 至 USART8 不支持此位，默认为零。
位 20	LPWUF	0	r	低功耗唤醒标志 当检测到唤醒事件时，该位被硬件置起，由软件将其清零。 0：无； 1：有。 注：USART4 至 USART8 不支持此位，默认为零。
位 17	CMDF	0	r	字节匹配检测标志 当接收到由 ID[7:0]定义的字节时，该位被硬件置起，由软件将其清零。 0：无； 1：有。
位 16	OCCUPY	0		接收器占用标志 0：接收器空闲 1：接收器被占用 注：USART4 至 USART8 不支持此位，默认为零。
位 11	RTODF	0	r	接收器超时检测标志

				当超时值达到 RTOV 寄存器编程的值，若无任何通信，该位被硬件置起，由软件将其清零。 0：无； 1：有。
位 9	CTSCF	0x0	rw0c	CTS 变化标志（CTS change flag） 当 CTS 线发送变化时，该位被硬件置起，由软件将其清零。 0：无； 1：有。
位 8	BFF	0x0	rw0c	间隔帧标志（break frame flag） 当检测到间隔帧时，该位被硬件置起，由软件将其清零。 0：无； 1：有。
位 7	TDBE	0x1	ro	发送缓冲器空（Transmit data buffer empty） 当发送缓冲器为空，可以再次写入数据时，该位被硬件置起。对 USART_DT 的写操作，将清零该位。 0：非空； 1：空。
位 6	TDC	0x1	rw0c	发送数据完成（Transmit data CMPlete） 当发送数据完成，该位被硬件置起，由软件将其清零（方式 1：先读 USART_STS，再写 USART_DT；方式 2：操作该位写'0'）。 0：未完成； 1：完成。
位 5	RDBF	0x0	rw0c	接收数据缓冲器满（Receive data buffer full） 当接收到数据时，该位被硬件置起，由软件将其清零（方式 1：读 USART_DT；方式 2：操作该位写'0'）。 0：未收到； 1：收到。
位 4	IDLEF	0x0	ro	总线空闲（Idle flag） 当检测到总线空闲时，该位被硬件置起，由软件将其清零（先读 USART_STS，再读 USART_DT）。 0：无； 1：有。
位 3	ROERR	0x0	ro	接收器溢出错误（Receiver overflow error） 当 RDBF 仍然置起没有清除的时候，如果此时又收到数据，该位被硬件置起，由软件将其清零（先读 USART_STS，再读 USART_DT）。 0：无； 1：有。 注意：该位被置位时，DT 寄存器中的数据不会丢失，但是后续的数据会被覆盖。
位 2	NERR	0x0	ro	杂讯错误（Noise error） 接收到的数据有杂讯时，该位被硬件置起，由软件将其清零（先读 USART_STS，再读 USART_DT）。 0：无； 1：有。
位 1	FERR	0x0	ro	帧错误（Framing error） 当检测到停止位异常（检测到低电平）、过多的杂讯噪声或者检测到间隔帧，该位被硬件置起，由软件将其清零（先读 USART_STS，再读 USART_DT）。 0：无； 1：有。
位 0	PERR	0x0	ro	校验错误（Parity error） 接收如果出现奇偶校验错误，该位被硬件置起，由软件将其清零（先读 USART_STS，再读 USART_DT）。 0：无； 1：有。

12.13.2数据寄存器（USART_DT）

域	简称	复位值	类型	功能
位 31：9	保留位	0x000000	resd	硬件强制为 0。
位 8：0	DT	0x000	rw	数据值（Data value） 该寄存器包含读和写的功能。当奇偶校验位使能，发送操作时，写到 MSB 的值会被校验位取代。接收操作时，读到的 MSB 位是接收到的校验位。

12.13.3波特比率寄存器（USART_BAUDR）

注意：如果 TEN 或 REN 均被禁止，波特计数器停止计数。

域	简称	复位值	类型	功能
位 31：16	保留位	0x0000	resd	硬件强制为 0。
位 15：0	DIV	0x0000	rw	分频系数（Division） 这 16 位定义了 USART 分频系数。

12.13.4控制寄存器1（USART_CTRL1）

域	简称	复位值	类型	功能
位 31：29	保留位	0x00000	resd	硬件强制为 0。
位 28	DBN1	0x0	rw	数据位个数高位（Data bit num） 该位和 DBN0 位一起定义了数据位的个数。 10：7 位； 00：8 位； 01：9 位； 11：禁止写入，否则数据异常。
位 27	RTODEN	0	rw	接收器超时检测使能位（receiver time out detection enable） 0：关闭； 1：开启。
位 26	RETODIE	0	rw	接收器超时检测中断使能位（receiver time out detection interrupt enable） 0：关闭； 1：开启。
位 25：21	TSDT	0x00	rw	发送器开始延迟时间（transmit start delay time） RS485 模式下一系列连续发送的第一笔数据会在数据写入后延迟一段时间后发送，以确保外部收发器已将传输方向切换为发送，该时间由 TSDT 的值决定，时间单位为 1/16 个波特率周期。
位 20：16	TCDT	0x00	rw	发送器完成延迟时间（transmit complete delay time） RS485 模式下一系列连续发送的最后一笔数据会在数据最后一个停止位发送完成后延迟一段时间结束，以确保外部收发器转接器已将传输方向切换为接收，该时间由 TCDT 的值决定，时间单位为 1/16 个波特率周期。
位 15	保留	0	resd	保持默认值。
位 14	CMDIE	0	rw	字节匹配检测中断使能位（character match detection interrupt enable） 0：关闭； 1：开启。
位 13	UEN	0x0	rw	USART 使能（USART enable） 0：关闭； 1：开启。
位 12	DBN0	0x0	rw	数据位个数高位（Data bit num） 该位和 DBN1 位一起定义了数据位的个数。 10：7 位； 00：8 位； 01：9 位； 11：禁止写入，否则数据异常。
位 11	WUM	0x0	rw	唤醒方式（Wake up mode）

				该位定义静默状态下被唤醒的方式。 0：空闲帧唤醒； 1：ID 匹配唤醒。
位 10	PEN	0x0	rw	奇偶校验使能（Parity enable） 该位定义使能硬件奇偶校验（对于发送来说就是校验位的产生；对于接收来说就是校验位的检测）。当使能了该位，硬件将发送数据的最高位替换成校验位；对接收到的数据检查其校验位是否正确。 0：关闭； 1：开启。
位 9	PSEL	0x0	rw	奇偶校验选择（Parity selection） 该位定义是采用奇校验还是偶校验。 0：偶校验； 1：奇校验。
位 8	PERRIEN	0x0	rw	PERR 中断使能（PERR interrupt enable） 0：关闭； 1：开启。
位 7	TDBEIEN	0x0	rw	发送数据缓冲器空中断使能（TDBE interrupt enable） 0：关闭； 1：开启。
位 6	TDCIEN	0x0	rw	发送数据完成中断使能（TDC interrupt enable） 0：关闭； 1：开启。
位 5	RDBFIEN	0x0	rw	接收数据缓冲器满中断使能 （RDBF interrupt enable） 0：关闭； 1：开启。
位 4	IDLEIEN	0x0	rw	总线空闲中断使能 （IDLE interrupt enable） 0：关闭； 1：开启。
位 3	TEN	0x0	rw	发送使能 （Transmitter enable） 该位定义发送端的使能。 0：关闭； 1：开启。
位 2	REN	0x0	rw	接收使能 （Receiver enable） 该位定义接收端的使能。 0：关闭； 1：开启。
位 1	RM	0x0	rw	接收静默 （Receiver mute） 该位定义接收端静默的开启，可由软件置起或清零。当配置为空闲帧唤醒时，唤醒后硬件也会将其清零，当配置为匹配地址唤醒时，收到匹配地址唤醒后硬件会将其清零，收到不匹配地址后硬件会再次将其置起进入静默状态。 0：普通； 1：静默。
位 0	SBF	0x0	rw	发送间隔帧 （Send break frame） 使用该位来发送间隔帧。该位可以由软件置起或清零。常规用法是软件置起该位，间隔帧发送完成后，由硬件将该位清零。 0：无； 1：发送。

12.13.5控制寄存器2（USART_CTRL2）

域	简称	复位值	类型	功能
位 31：28	IDH	0x0	rw	USART 的 ID 号高 4 位（USART identification） 可配置的 USART 的 ID 号。
位 27：20	保留	0x000	resd	保持默认值。
位 19	MTF	0	rw	MSB 先传输（MSB transmit first） 该位用于选择数据先传输 MSB 还是 LSB。 0：LSB；

				1：MSB。 *注意：使能 MTF 时，不支持奇偶效验。*
位 18	DTREV	0	rw	DT 寄存器极性反向（DT register polarity reverse）0： 0：1=H，0=L 1：1=L，0=H
位 17	TXREV	0	rw	TX 引脚极性反向（TX polarity reverse） 0：VDD=1/idle,Gnd=0/mark 1：VDD=0/mark,Gnd=1/idle
位 16	RXREV	0	rw	RX 引脚极性反向（RX polarity reverse） 0：VDD=1/idle,Gnd=0/mark 1：VDD=0/mark,Gnd=1/idle
位 15	TRPSWAP	0x0	rw	收发管脚交换（Transmit receive pin swap） 0：关闭； 1：开启。
位 14	LINEN	0x0	rw	LIN 模式使能（LIN mode enable） 0：关闭； 1：开启。
位 13：12	STOPBN	0x0	rw	停止位个数（STOP bit num） 这 2 位用来设置停止位的个数 00：1 位； 01：0.5 位； 10：2 位； 11：1.5 位；
位 11	CLKEN	0x0	rw	时钟使能（Clock enable） 该位用来使能同步模式或智能卡模式的时钟引脚。 0：关闭； 1：开启。
位 10	CLKPOL	0x0	rw	时钟极性（Clock polarity） 在同步模式或智能卡模式下，可以用该位选择时钟引脚上总线空闲时时钟输出的极性。 0：低电平； 1：高电平。
位 9	CLKPHA	0x0	rw	时钟相位（Clock phase） 在同步模式或智能卡模式下，可以用该位选择时钟引脚上时钟输出的相位。 0：第一个边沿进行数据捕获； 1：第二个边沿进行数据捕获。
位 8	LBCP	0x0	rw	最后一位时钟脉冲（Last bit clock puLEXT） 在同步模式下，使用该位来控制是否在时钟引脚上输出数据的最后一位对应的时钟脉冲 0：不输出； 1：输出。
位 7	保留位	0x0	resd	保持默认值。
位 6	BFIEN	0x0	rw	间隔帧中断使能（break frame interrupt enable） 0：关闭； 1：开启。
位 5	BFBN	0x0	rw	间隔帧位数（break frame bit num） 该位用来选择是 11 位还是 10 位的间隔帧。 0：10 位； 1：11 位。
位 4	IDBN	0	rw	ID 号位数(Identification bit num) 此位用于选择 ID 号位数。 0：4 位； 1：数据位-1 位。 注意：当该位置’1’时，在 7、8 或 9 位数据模式下，ID 号位数分别为低 6、7 或者 8 位，如有配置奇偶校验则为 5、6 或者 7 位。
位 3：0	IDL	0x0	rw	USART 的 ID 号（USART identification） 可配置的 USART 的 ID 号。

注意： 在使能发送后不能改写这三个位（CLKPOL、CLKPHA、LBCP）。

12.13.6控制寄存器3（USART_CTRL3）

域	简称	复位值	类型	功能
位 31：18 位 12	保留位	0x000000	resd	硬件强制为 0。
位 17：16	LPWUM	0x0	rw	低功耗唤醒方式（low power wakeup method） 00：ID 匹配 01：保留 10：起始位 11：RDBF 注：USART4 至 USART8 不支持此位，默认为零。
位 15	DEP	0	rw	DE 信号极性选择 (DE polarity selection) 0：高电平有效。 1：低电平有效。
位 14	RS485EN	0	rw	RS485 使能 (RS485 enable) 此位用于使能 RS485 模式，RS485 模式下 USART 通过控制信号 DE 控制外部收发器传输方向。 0：禁止 RS485 模式，控制信号 DE 禁止输出，RTS 引脚作 RS232 模式使用。 1：使能 RS485 模式，控制信号 DE 在 RTS 引脚上输出。
位 13	LPWUFIE	0	rw	低功耗唤醒标志中断使能（low power wakeup flag interrupt enable） 0：关闭； 1：开启。 注：USART4 至 USART8 不支持此位，默认为零。
位 11	SMUSEN	0	rw	DEEPSLEEP 模式 USART 使能（Deepsleep mode usart enable） 0：关闭； 1：开启。 注：USART4 至 USART8 不支持此位，默认为零。
位 10	CTSCFIEN	0x0	rw	CTSCF 中断使能（CTSCF interrupt enable） 0：关闭； 1：开启。
位 9	CTSEN	0x0	rw	CTS 使能（CTS enable） 0：关闭； 1：开启。
位 8	RTSEN	0x0	rw	RTS 使能（RTS enable） 0：关闭； 1：开启。
位 7	DMATEN	0x0	rw	DMA 发送使能（DMA transmit enable） 0：关闭； 1：开启。
位 6	DMAREN	0x0	rw	DMA 接收使能（DMA receiver enable） 0：关闭； 1：开启。
位 5	SCMEN	0x0	rw	智能卡模式使能（Smart card mode enable） 0：关闭； 1：开启。
位 4	SCNACKEN	0x0	rw	智能卡 NACK 使能（Smart card NACK enable） 该位用于配置校验错误出现时，发送 NACK。 0：不发送； 1：发送。
位 3	SLBEN	0x0	rw	单线双向半双工模式使能（Single line bidirectional half-duplex enable） 0：关闭； 1：开启。
位 2	IRDALP	0x0	rw	红外低功耗模式配置（IrDA low-power mode） 该位用来配置红外低功耗模式。 0：关闭； 1：开启。

位 1	IRDAEN	0x0	rw	红外功能使能（IrDA enable） 0：关闭； 1：开启。
位 0	ERRIEN	0x0	rw	错误中断使能（Error interrupt enable） 当有帧错误、接收溢出错误或者杂讯错误时产生中断。 0：关闭； 1：开启。

12.13.7保护时间和预分频寄存器（GDIV）

域	简称	复位值	类型	功能
位 31：16	保留位	0x0000	resd	硬件强制为 0。
位 15：8	SCGT	0x00	rw	智能卡保护时间值（Smart card guard time） 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，这几位配置保护时间值。
位 7：0	ISDIV	0x00	rw	红外或者智能卡分频系数（IrDA/smaERTCard division） 红外（IrDA）模式： 8 位[7：0]有效，普通模式无效且只能设置为 00000001，低功耗模式分频系数对外设时钟进行分频，作为脉冲宽度的基数周期； 00000000：保留－不要写入该值； 00000001：1 分频； 00000010：2 分频； …… 智能卡模式： 低 5 位[4：0]有效，分频系数对外设时钟进行分频，给智能卡提供时钟。可以设置为如下值： 00000：保留－不要写入该值； 00001：2 分频； 00010：4 分频； 00011：6 分频； ……

12.13.8接收器超时检测值寄存器（RTOV）

域	简称	复位值	类型	功能
位 31：24	保留位	0x00	resd	硬件强制为 0。
位 23：0	RTOV	0x00	rw	接收器超时检测值（receiver time out value） 单位为 1bit 位宽

12.13.9中断标志位清除寄存器（IFC）

域	简称	复位值	类型	功能
位 31：21 位 19：18 位 16：12 位 10：0	保留位	0x00	resd	硬件强制为 0。
位 20	LPWUFC	0	w1	低功耗唤醒标志清除位（low power wake up flag clear） 注：USART4 至 USART8 不支持此位，默认为零。
位 17	CMDFC	0	w1	字节匹配检测标志清除位（character match detection flag clear）
位 11	RTODFC	0	w1	接收器超时检测标志清除位（receiver time out detection flag clear）

13 串行外设接口（SPI）

13.1 串行外设接口（SPI）简介

SPI 接口提供软件编程配置选项，根据软件编程配置方式不同，可以分别作为 SPI 和 I²S 使用。本章将分 SPI 和 I²S 分别介绍 SPI 作 SPI 或 I²S 的功能特性以及配置流程。

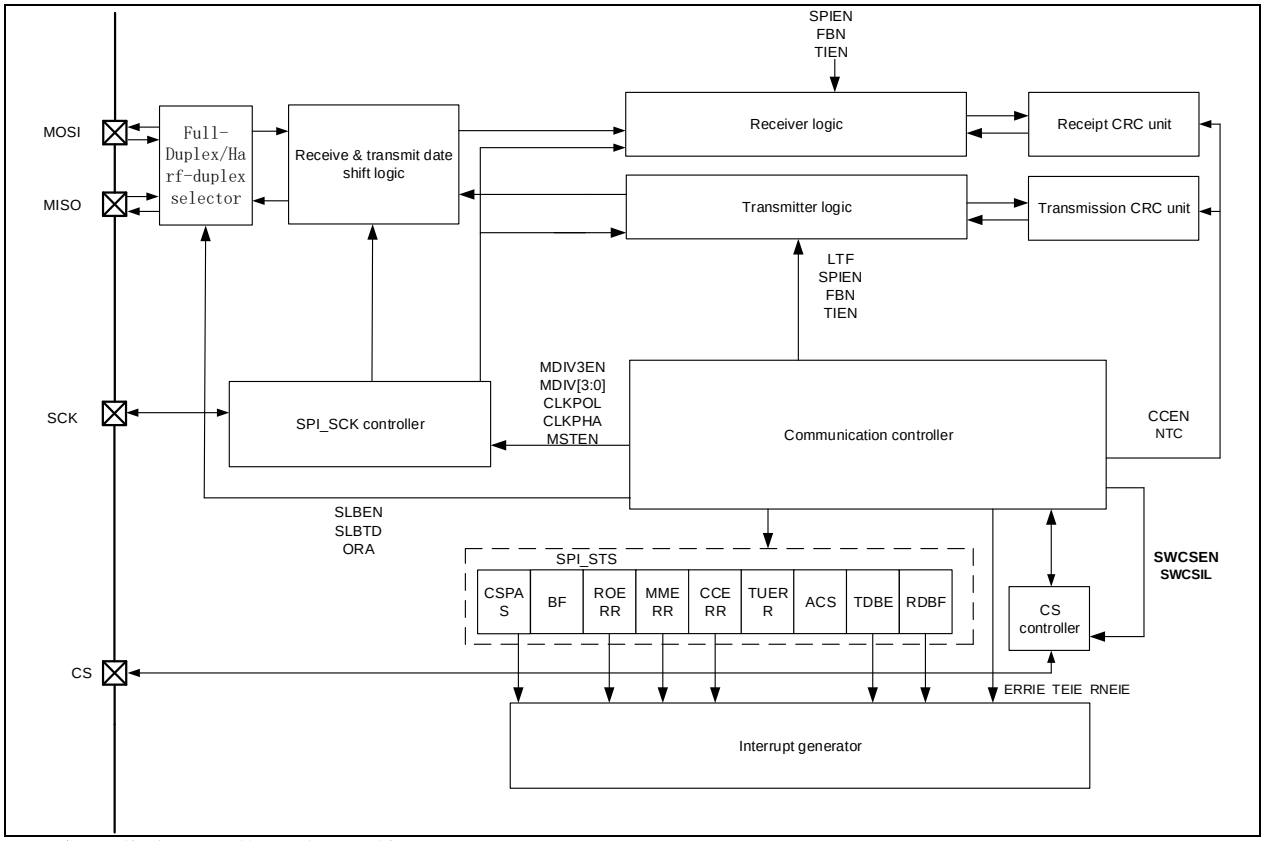
13.2 SPI功能描述

13.2.1 SPI简述

串行外设接口（SPI）根据软件编程配置的方式不同，可以分别作为主机和从机使用，又可以分别工作在全双工，全双工只收，半双工只发/只收四种不同的模式下，并且还提供 DMA 传输，SPI 内部硬件自动 CRC 计算和校验等功能，同时可以通过软件编程配置使 SPI 接口兼容 TI 协议。

SPI 的架构框图见下图：

图 13-1 SPI框图



SPI 接口作为 SPI 使用时主要特征如下：

- 可编程配置的全双工或半双工通信：
  - 全双工同步通信（可以选择全双工只收以此释放用于发送的IO）；
  - 半双工同步通信（可以根据软件编程配置选择传输方向：发送或接收）。
- 可编程配置主/从模式。
- 可编程配置的 CS 信号处理方式：
  - 硬件处理CS；
  - 软件处理CS。
- 可编程配置的 8 位或 16 位帧位数。
- 可编程配置的通信频率以及分频系数（最大分频系数为$f_{PCLK}/2$）。
- 可编程配置的时钟极性和相位。
- 可编程配置的数据传输顺序(先发 MSB/LSB)。
- 可编程配置的错误中断标志（CS脉冲异常，接收器溢出错误，主模式错误，CRC校验错

误）。

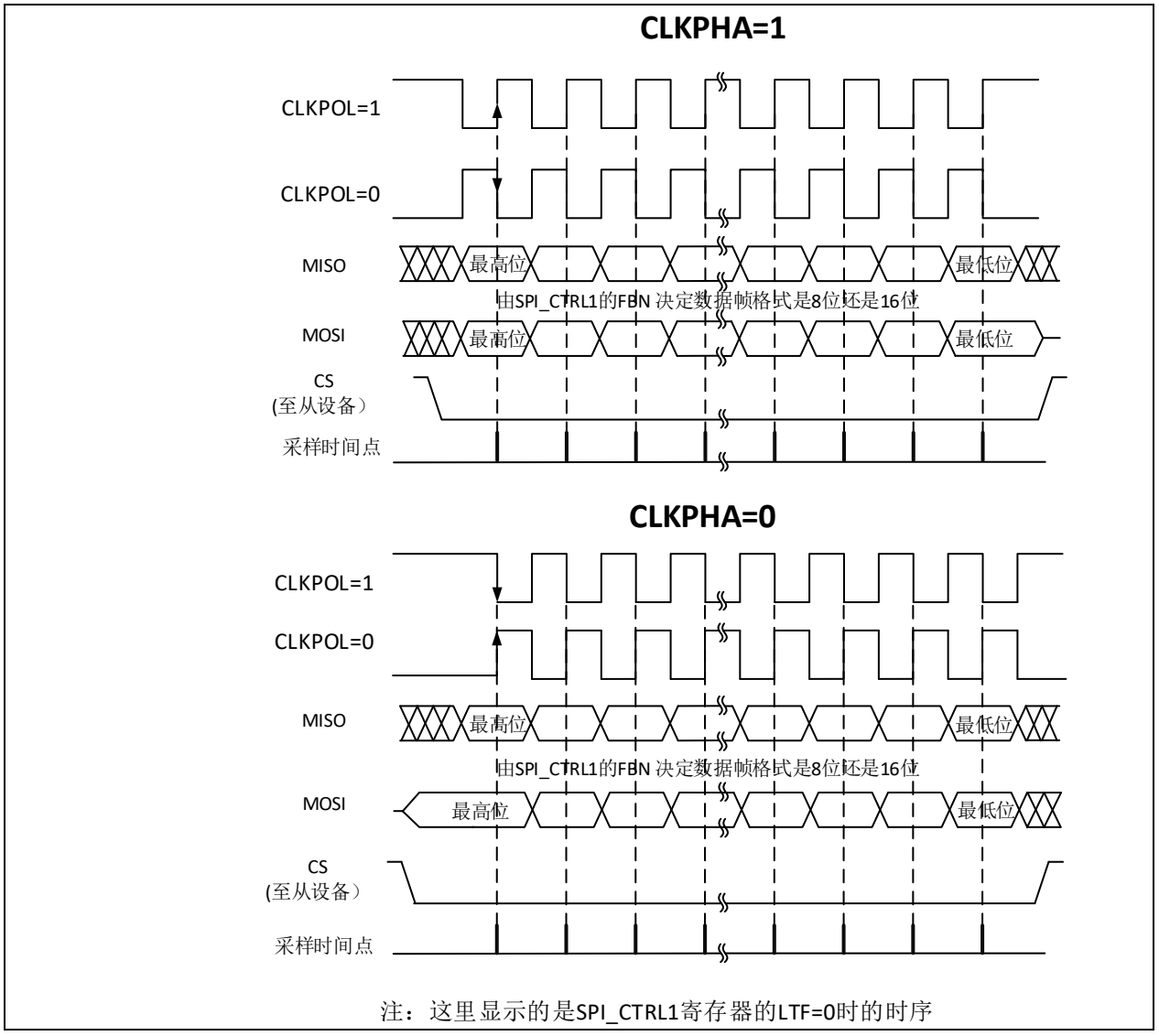
- 可编程配置的发送数据缓冲器空中断以及接收数据缓冲器满中断。
- 支持 DMA 发送和接收。
- 支持硬件 CRC 发送和校验。
- 具备通信忙标志。
- 兼容 TI 协议。

时钟信号的相位和极性

SPI_CTRL1 寄存器的 CLKPOL 和 CLKPHA 位，能够组合四种可能的时序关系。CLKPOL（时钟极性）位控制在没有数据传输时时钟的空闲状态电平，此位对主模式和从模式下的设备都有效。如果 CLKPOL 被清‘0’，SCK 引脚在空闲状态保持低电平；如果 CLKPOL 被置‘1’，SCK 引脚在空闲状态保持高电平。

如果 CLKPHA（时钟相位）位被置’1’，SCK 时钟的第二个边沿（CLKPOL 位为 0 时就是下降沿，CLKPOL 位为’1’时就是上升沿）进行数据位的采样，数据在第二个时钟边沿被锁存。如果 CLKPHA 位被清’0’，SCK 时钟的第一边沿（CLKPOL 位为’0’时就是上升沿，CLKPOL 位为’1’时就是下降沿）进行数据位采样，数据在第一个时钟边沿被锁存。

图 13-2 数据时钟时序图



注：这里显示的是SPI_CTRL1寄存器的LTF=0时的时序

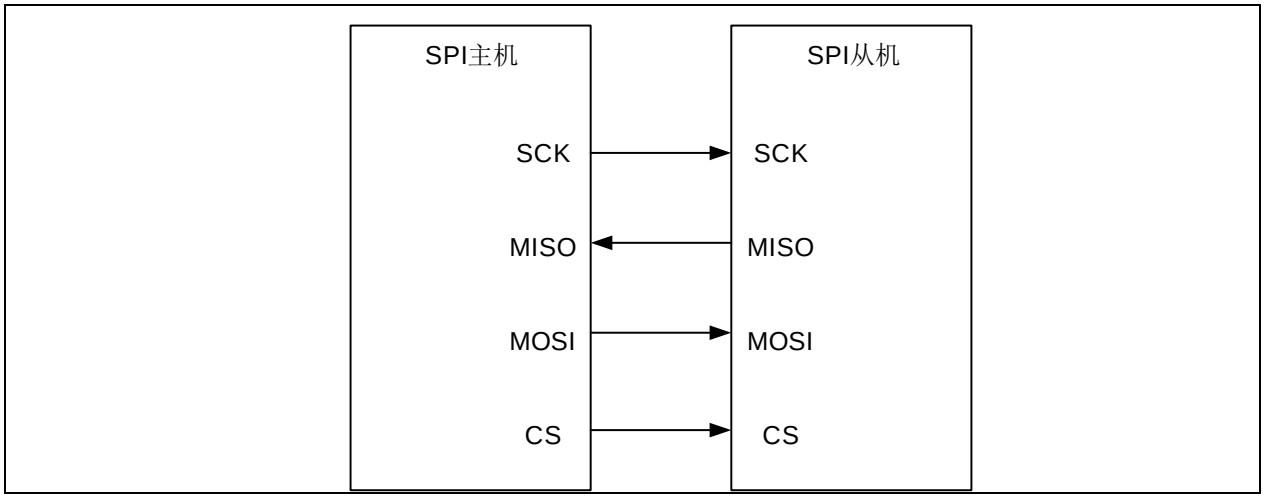
13.2.2 全双工半双工选择器简述和配置流程

SPI 全双工半双工选择器通过软件编程配置的方式，可以使 SPI 接口作为 SPI 使用时，可以工作在双线单向全双工，单线单向只收，单线双向半双工发送和单线双向半双工接收四种同步模式。

双线单向全双工模式配置方式以及 SPI IO 连接方式如下：

SLBEN 位置 0，ORA 置 0 时，SPI 工作在双线单向全双工，此时 SPI 可以同时进行数据的收发，IO 连接方式如下图。

图 13-3 SPI双线单向全双工连接示意图



SPI 作主机或从机在此模式下，关闭 SPI 或进入省电模式（或关闭 SPI 系统时钟）之前需要等待 RDBF 置位，TDBE 置位，并等待 BF=0。

单线单向只收模式配置方式以及 SPI IO 连接方式如下：

SLBEN 位置 0，ORA 置 1 时，SPI 工作在单线单向只收模式，此时 SPI 只能作为数据接收方，无法发送数据。作为主机时使用 MISO 接收数据，MOSI 管脚所映射的 IO 释放。作为从机时使用 MOSI 接收数据，MISO 管脚所映射的 IO 释放。

图 13-4 SPI作主机单线单向只收连接示意图

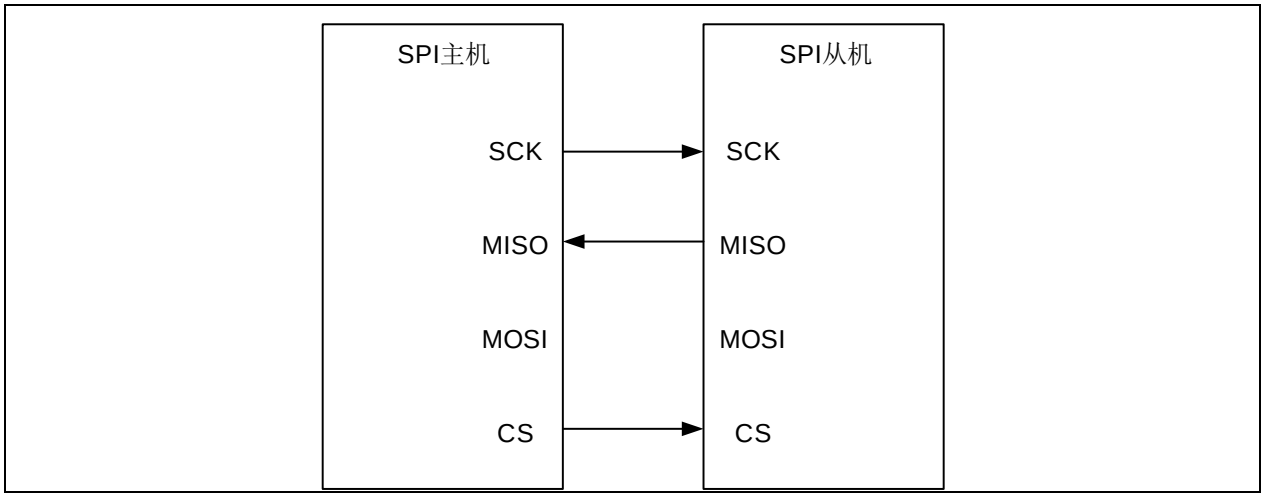
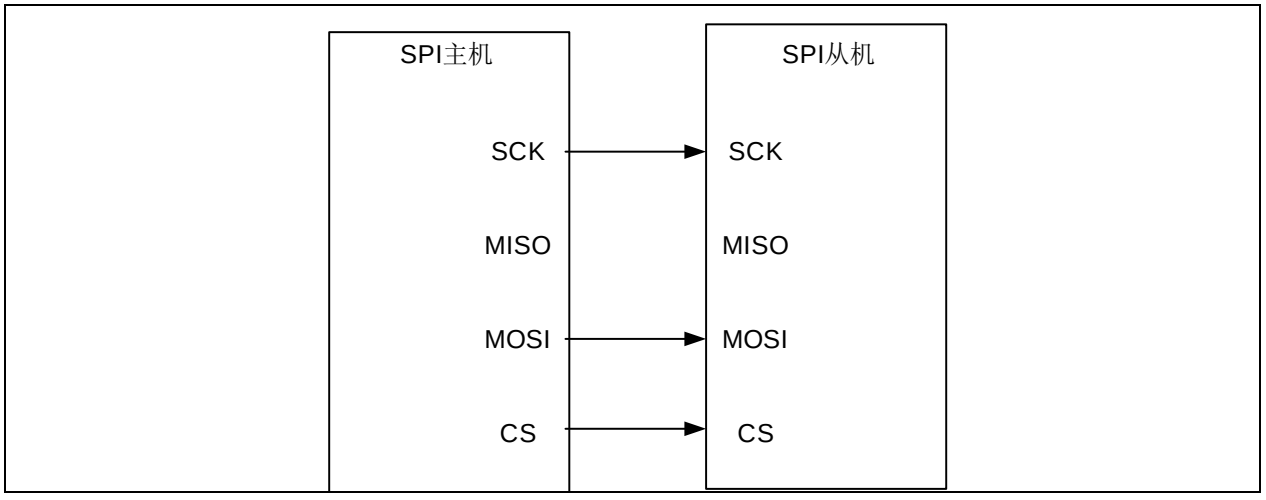


图 13-5 SPI作从机单线单向只收连接示意图

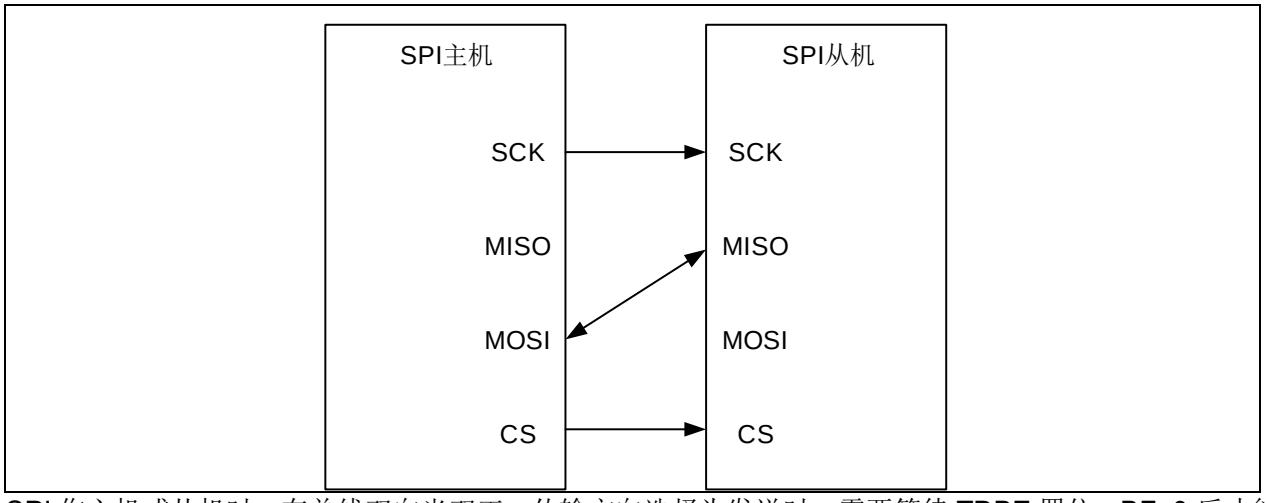


SPI 作主机时，在此模式下，需要等待倒数第二个 RDBF 置起，关闭 SPI 之前等待一个 SPI_SCK 周期，在进入省电模式(或关闭 SPI 系统时钟)之前等待最后一个 RDBF=1。
SPI 作从机时，在此模式下，关闭 SPI 无需判断任何标志，但是在进入省电模式(或关闭 SPI 系统时钟)之前需要等待 BF=0。

单线双向半双工模式配置方式以及 SPI IO 连接方式如下：

SLBEN 位置 1 时，SPI 工作在单线双向半双工模式，此时 SPI 可以分时进行数据收发。作为主机时使用 MOSI 收发数据，MISO 管脚所映射的 IO 释放。作为从机时使用 MISO 收发数据，MOSI 管脚所映射的 IO 释放。
软件通过编程控制 SLBTD 位控制传输方向，SLBTD 位置 1 时，SPI 只能发送数据，SLBTD 位置 0 时，SPI 只能接收数据。

图 13-6 SPI作单线双向半双工连接示意图



SPI 作主机或从机时，在单线双向半双工，传输方向选择为发送时，需要等待 TDBE 置位，BF=0 后才能关闭 SPI，在关闭 SPI 后才可以进入省电模式(或关闭 SPI 系统时钟)。
SPI 作主机时，在单线双向半双工，传输方向选择为接收时，需要等待倒数第二个 RDBF 置起，关闭 SPI 之前等待一个 SPI_SCK 周期，在进入省电模式(或关闭 SPI 系统时钟)之前等待最后一个 RDBF=1。
SPI 作从机时，在单线双向半双工，传输方向选择为接收时，关闭 SPI 无需判断任何标志，但是在进入省电模式(或关闭 SPI 系统时钟)之前需要等待 BF=0。

13.2.3 CS控制器简述和配置流程

SPI 的 CS 控制器提供通过软件可编程配置的方式选择硬件控制 CS 信号或软件控制 CS 信号，以此实现 CS 信号的控制，用于多处理器模式下主机从机选择，以及通过 CS 信号后于 SCK 信号使能，有效地屏蔽总线上的干扰，下面将分软件 CS 以及硬件 CS 来介绍 CS 控制器的配置流程，并会简述在主机和从机模式下软件和硬件 CS 的输入输出方式。

硬件 CS 配置流程：

当 SPI 作主机，硬件 CS 输出时，HWCSOE 位置 1，SWCSEN 置 0，开启硬件 CS 控制，SPI 在使能之后会在 CS 管脚上输出低电平，在 SPI 关闭并且发送完成后，释放 CS 信号。
当 SPI 作主机，硬件 CS 输入时，HWCSOE 位置 0，SWCSEN 置 0，开启硬件 CS 控制，此时一旦主机 SPI 检测到 CS 管脚为低电平时，SPI 硬件自动关闭 SPI 并进入从机模式，模式错误标志 MMERR 同时置位，若使能了错误中断（ERRIE=1），将会产生中断，在 MMERR 置位期间，硬件不允许软件置位 SPIEN 和 MSTEN 位，通过读或写 SPI_STS 再写 SPI_CTRL1 可以清除 MMERR。
当 SPI 作从机，硬件 CS 输入时，HWCSOE 位置 0，SWCSEN 置 0，开启硬件 CS 控制，从机根据 CS 管脚上的电平判断是否发送或接收数据，只有 CS 管脚上为低电平时，从机才会被选中并进行数据的收发。

软件 CS 配置流程：

当 SPI 作主机，软件 CS 输入时， SWCSEN 位置 1，开启软件 CS 控制，当 SWCSIL 位置 0 时，SPI 硬件自动关闭 SPI 并进入从机模式，模式错误标志 MMERR 同时置位，若使能了错误中断（ERRIE=1），将会产生中断，在 MMERR 置位期间，硬件不允许软件置位 SPIEN 和 MSTEN 位，通过读或写 SPI_STS 再写 SPI_CTRL1 可以清除 MMERR。
当 SPI 作从机，软件 CS 输入时， SWCSEN 位置 1，开启软件 CS 控制，SPI 根据 SWCSIL 位判断 CS

信号电平，不使用 CS 管脚，当 SWCSIL=0 时，从机才会被选中并进行数据的收发。

13.2.4 SPI_SCK控制器简述和配置流程

SPI 协议采用同步传输，所以 SPI 接口在作为 SPI 使用时，作主机时，需要产生通信时钟用于 SPI 接口的数据收发，并且需要将该通信时钟通过 IO 输出给从机，用于从机的数据收发；作从机时，需要外设提供通信时钟从 IO 输入到 SPI 接口内部作为通信时钟使用，所以实际上，SPI_SCK 控制器便是扮演着产生 SPI_SCK 以及分配 SPI_SCK 的角色，详细的配置方法如下所述。

SPI_SCK 控制器配置流程：

- 时钟极性相位选择：配置CLKPOL,CLKPHA选择需要的极性和相位。
- 时钟分频系数选择：配置CRM选择需要的PCLK频率，配置MDIV[3：0]或MDIV3EN选择需要的分频系数。
- 主机或从机选择：配置MSTEN选择SPI作主机或从机使用，注意主机只收模式在SPI使能后就会开始输出时钟，直到SPI被关闭且接收完成。

13.2.5 CRC简述和配置流程

SPI 接口内部具有独立的发送和接收 CRC 计算单元，通过软件编程配置，SPI 接口在作为 SPI 使用时，可以同时在用户使用 DMA 读写数据或 CPU 读写数据的情况下，自动进行 CRC 计算以及 CRC 校验，如果在传输过程中，硬件检测到接收到的数据与 SPI_RCRC 中的数据不符，且该笔数据又是 CRC 数据时，CCERR 位会置起，若使能了错误中断（ERRIE=1），将会产生中断。

下面分 DMA 和 CPU 操作数据寄存器分别描述 SPI 的 CRC 功能以及 CRC 配置流程。

CRC 配置流程

- CRC计算多项式配置：配置SPI_CPOLY选择CRC计算多项式。
- 使能CRC：置起CCEN位使能CRC计算，该操作将会复位SPI_RCRC以及SPI_TCRC。
- 根据DMA或CPU操作数据寄存器选择是否以及何时置位NTC位，具体请参见下方描述。

DMA 发送模式：

在采用 DMA 写入待发送的数据时，当使能 CCEN 后，硬件会根据 SPI_CPOLY 中的值以及每笔发送的数据自动计算 CRC 值，并在最后一笔数据发送完成后自动发送 CRC 值，该值即 SPI_TCRC 中的值。

DMA 接收模式：

在采用 DMA 读取待接收的数据时，当使能 CCEN 后，硬件会根据 SPI_CPOLY 中的值以及每笔接收的数据自动计算 CRC 值，并在最后一笔数据接收完成后等待 CRC 数据接收完成，并将收到的 CRC 值和 SPI_RCRC 中的值作比较，若校验出错，会置起 CCERR 标志，若使能了 ERRIE 位，则产生错误中断。

CPU 发送模式：

相较于 DMA 发送模式，该模式需要软件在写入最后一笔待发送的数据后，在最后一笔数据发送完成之前置起 NTC 位。

CPU 接收模式：

在双线单向全双工模式下，按照 CPU 发送模式操作 NTC 位，CPU 接收模式的 CRC 计算和校验会自动完成，在单线单向只接收以及单线双向只接收模式下，相较于 DMA 接收需要软件在接收到倒数第二笔数据之后，接收到最后一笔数据之前置起 NTC 位。

13.2.6 DMA传输简述和配置流程

SPI 接口支持使用 DMA 进行发送数据的写入，接收数据的读取，具体配置流程分别见下述的 DMA 发送配置流程以及 DMA 接收配置流程。

需要特别注意的是，在开启CRC计算和校验时，DMA发送数据的个数配置为待发送的数据个数，DMA读取数据的个数配置为待接收的数据个数+1，此时硬件在所有数据传输完毕后自动进行CRC传输，且接收方还会自动进行CRC校验，需要注意，接收到的CRC数据，硬件会搬到SPI_DT寄存器中，并置位RDBF，以及在开启了DMA传输时发出DMA读请求，所以这里推荐当CRC接收完毕后软件要去读DT寄存器来取走CRC值，防止后续传输出错。

DMA 发送配置流程：

- 选择DMA传输通道：根据DMA章节DMA弹性请求映射选择DMA通道用于当前SPI的DMA传输。

- 配置DMA传输目标地址：在DMA控制寄存器中DMA传输目的地址位写入当前所使用的SPI的SPI_DT寄存器地址，DMA将会在接收到发送请求后将待发送的数据写入该地址。
- 配置DMA传输源地址：在DMA控制寄存器中DMA传输源地址位写入待发送数据存放的地址，DMA将会在接收到发送请求后将该地址内的数据写入到目标地址中，即写入到当前所使用的SPI的SPI_DT寄存器中。
- 配置DMA传输数据个数：在DMA控制寄存器相关位置配置期望传输的数据个数。
- 配置DMA传输通道优先级：在DMA控制寄存器相关位置配置当前所使用通道的SPI的DMA传输通道优先级。
- 配置DMA中断产生时机：在DMA控制寄存器相关位置配置是在传输完成或传输完成一半时产生DMA中断。
- 使能DMA传输通道：在DMA控制寄存器相关位置使能当前所选用的DMA通道。

DMA接收配置流程：

- 选择DMA传输通道：根据DMA章节DMA弹性请求映射选择DMA通道用于当前SPI的DMA传输。
- 配置DMA传输目标地址：在DMA控制寄存器中DMA传输目的地址位写入期望存放接收数据的地址，DMA将会在接收到接收请求后，将当前所使用的SPI的SPI_DT寄存器中的数据存放在目的地址中。
- 配置DMA传输源地址：在DMA控制寄存器中DMA传输源地址位写入当前所使用的SPI的SPI_DT寄存器的地址，DMA将会在接收到接收请求后将该地址内的数据写入到目标地址中，即写入到期望存放接收数据的地址。
- 配置DMA传输数据个数：在DMA控制寄存器相关位置配置期望传输的数据个数。
- 配置DMA传输通道优先级：在DMA控制寄存器相关位置配置当前所使用通道的SPI的DMA传输通道优先级。
- 配置DMA中断产生时机：在DMA控制寄存器相关位置配置是在传输完成或传输完成一半时产生DMA中断。
- 使能DMA传输通道：在DMA控制寄存器相关位置使能当前所选用的DMA通道。

13.2.7 TI模式简述和配置流程

SPI 接口支持 TI 协议，用户可以通过将 TIEN 位置 1 使能 TI 模式。

使能 TI 模式后，SPI 接口将按照 TI 协议要求产生通信时钟 SPI_CLK，这意味着 SPI_CLK 的极性和相位会强制符合 TI 协议要求，用户无需也无法通过配置 CLKPOL 和 CLKPHA 来改变 SPI_CLK 的极性和相位。

使能 TI 模式后，SPI 接口将按照 TI 协议要求产生 CS 信号，这意味着 CS 的输出和输入将强制符合 TI 协议的要求，用户无需也无法通过配置 SWCSEN 位，SWCSIL 位，HWCSOE 位来管理 CS 信号。

使能 TI 模式后，SPI 从机只会在数据发送期间控制 MISO 引脚，这意味着 SPI 从机的 MISO 引脚将会在空闲状态保持高阻态。

使能 TI 模式后，SPI 接口作从机使用时，SPI 接口会在数据传输期间侦测错误的 CS 脉冲，并在侦测到错误的 CS 脉冲后置起 CSPAS 位(该位可以通过软件读 SPI_STS 清除)，此时 SPI 会忽略掉该错误的脉冲，但由于此时 CS 信号已经出现异常，软件应当关闭 SPI 从机，重新配置 SPI 主机，再打开 SPI 从机以重新开始通信。

13.2.8 发送器简述和配置流程

SPI 发送器时钟由 SPI_SCK 控制器提供，根据软件编程配置，发送器可以输出不同的数据帧格式，SPI 具有一个数据缓冲寄存器 SPI_DT，软件需要将待发送的数据先写入 SPI_DT，发送器在有时钟时，会把 SPI_DT 中的数据保存到发送器中的数据缓冲器(有别于 SPI_DT，SPI 发送器中的数据缓冲器由 SPI_SCK 驱动，且硬件自动控制，软件不可操作)，并按照配置好的帧格式将数据依次发出。

用户可以选择 DMA 或 CPU 来控制数据的写入，若选择 DMA 传输，详细配置请参见 DMA 传输章节，若选择 CPU 传输，则用户需要判断 TDBE 位，该位复位值为 1，代表 SPI_DT 为空，若 TDBEIE 置位，则产生中断，数据写入后，TDBE 拉低，直到数据被同步到发送器中的数据缓冲器后，TDBE 再次被拉起，即用户只可以在 TDBE 置位时写入待发送的数据。

发送器配置完成并使能 SPI 后，SPI 将进入数据发送状态，所以在此之前，应需要参考全双工半双工章节配置通信选用的是全双工或半双工等，并参考 CS 控制器章节配置选用的 CS 控制模式，还需要参考 SPI_SCK 控制章节配置通信时钟，若使用了 CRC 以及 DMA，还需参考 CRC 以及 DMA 传输章节配置 CRC 以及 DMA，如下为推荐的发送器配置流程。

发送器配置流程：

- 配置全双工半双工选择器。
- 配置CS控制器。
- 配置SPI_SCK控制器。
- 配置CRC（若需要使用CRC自动计算和校验功能）。
- 配置DMA传输（若需要使用DMA传输功能）。
- 若没有选择DMA传输功能，软件需要判断TDBE位，软件需要根据需求判断是否要打开发送数据中断，即置位TDBEIE。
- 配置帧格式：配置LTF位选择MSB/LSB格式，配置FBN选择8/16位数据。
- 置位SPIEN位使能SPI。

13.2.9 接收器简述和配置流程

SPI 接收器时钟由 SPI_SCK 控制器提供，根据软件编程配置，接收器可以接收不同的数据帧格式，SPI 接收器具有一个接收数据缓冲寄存器，该寄存器由 SPI_SCK 驱动，在每笔传输的最后一个 CLK，数据从移位寄存器压入该接收数据缓冲寄存器，随后发送器会给出数据接收完成的标志给到 SPI 的控制逻辑，SPI 的控制逻辑在检测到该标志后会自动把接收器中的数据缓冲器中的值压入 SPI_DT，RDBF 随之置起，这意味着有数据被收到，且该数据已被压入 SPI_DT，此时读 SPI_DT 可以读出该笔数据，同时 RDBF 随之清除。

用户可以选择 DMA 或 CPU 来控制数据的读出，若选择 DMA 传输，详细配置请参见 DMA 传输章节，若选择 CPU 传输，则用户需要判断 RDBF 位，该位复位值为 0，代表 SPI_DT 为空，当有数据被接收到，且数据被移入 SPI_DT 时，RDBF 置位，代表 SPI_DT 内有数据等待读取，此时若 RDBFIE 置位则产生中断。

若在下一笔接收器接收到的数据准备压入 SPI_DT 时，之前接收到的数据仍未被读走，即 RDBF 仍为 1，则代表数据溢出，在此之前接收到的数据不会丢失，但之后的数据都将丢失，此时 ROERR 置起，若 ERRIE 置位，则产生错误中断，依次读 SPI_DT 寄存器和 SPI_STS 寄存器可将 ROERR 清除，如下为推荐的接收器配置流程。

接收器配置流程：

- 配置全双工半双工选择器。
- 配置CS控制器。
- 配置SPI_SCK控制器。
- 配置CRC（若需要使用CRC自动计算和校验功能）。
- 配置DMA传输（若需要使用DMA传输功能）。
- 若没有选择DMA传输功能，软件需要判断RDBF位，软件需要根据需求判断是否要打开接收数据中断，即置位RDBFIE。
- 配置帧格式：配置LTF位选择MSB/LSB格式，配置FBN选择8/16位数据。
- 置位SPIEN位使能SPI。

13.2.10 Motorola模式通信时序

本节介绍 SPI 通信时序，包括全双工和半双工的主/从通信时序。

全双工通信-主机通信时序

其中主机端配置如下：

MSTEN=1：设备为主机；

SLBEN=0：全双工模式；

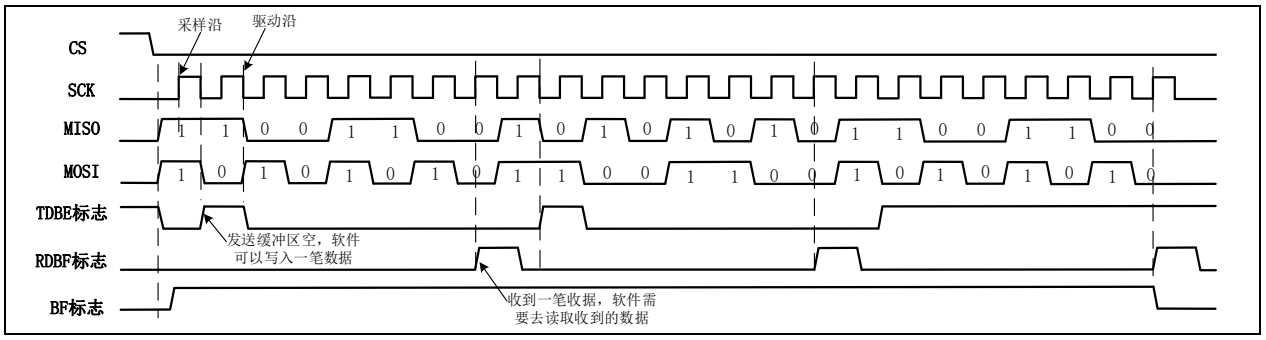
CLKPOL=0，CLKPHA=0：SCK 空闲输出低电平，第一个边沿作为采样边沿；

FBN=0：帧数据的长度为 8 位；

主机发送数据（MOSI）：0xaa, 0xcc, 0xaa；

从机发送数据（MISO）：0xcc, 0xaa, 0xcc；

图 13-7 主机全双工通信



全双工通信-从机通信时序

其中从机端配置如下：

MSTEN=0：设备为从机；

SLBEN=0：全双工模式；

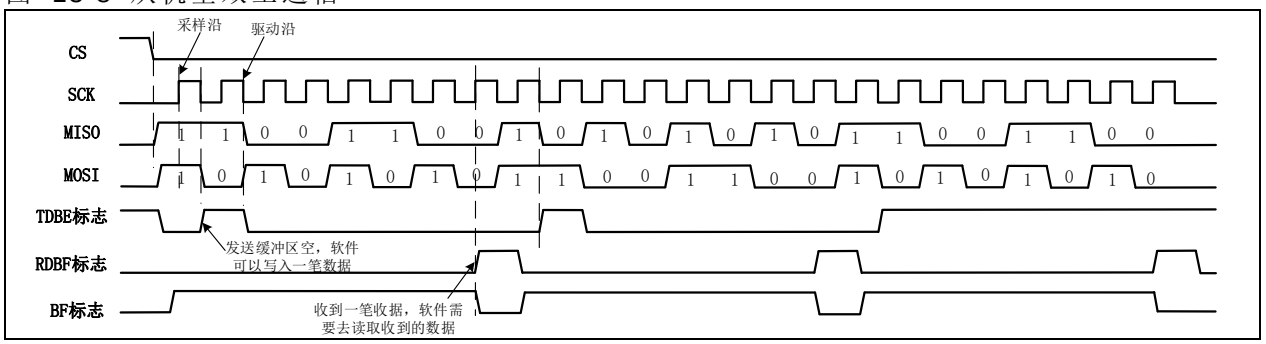
CLKPOL=0，CLKPHA=0：SCK 空闲输出低电平，第一个边沿作为采样边沿；

FBN=0：帧数据的长度为 8 位；

主机发送数据（MOSI）：0xaa, 0xcc, 0xaa；

从机发送数据（MISO）：0xcc, 0xaa, 0xcc；

图 13-8 从机全双工通信



半双工通信-主机发送时序

MSTEN=1：设备为主机；

SLBEN=1：单线双向模式；

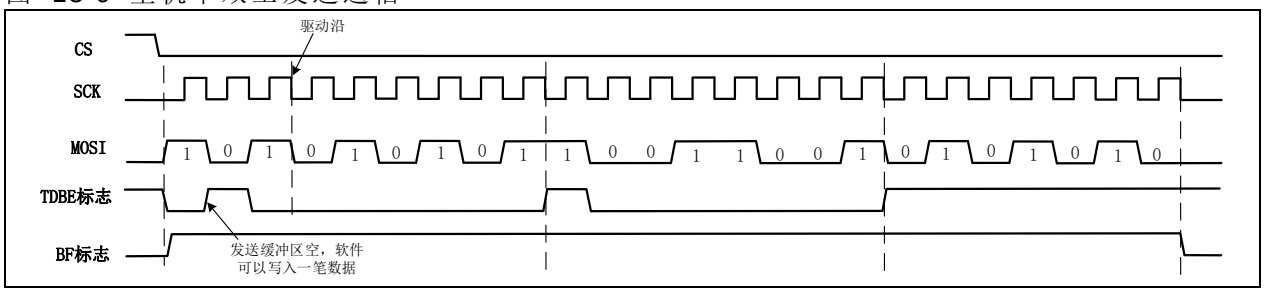
SLBTD=1：发送模式；

CLKPOL=0，CLKPHA=0：SCK 空闲输出低电平，第一个边沿为采样边沿；

FBN=0：帧数据的长度为 8 位；

主机发送数据：0xaa, 0xcc, 0xaa；

图 13-9 主机半双工发送通信



半双工通信-从机接收时序

MSTEN=0：设备为从机；

SLBEN=1：单线双向模式；

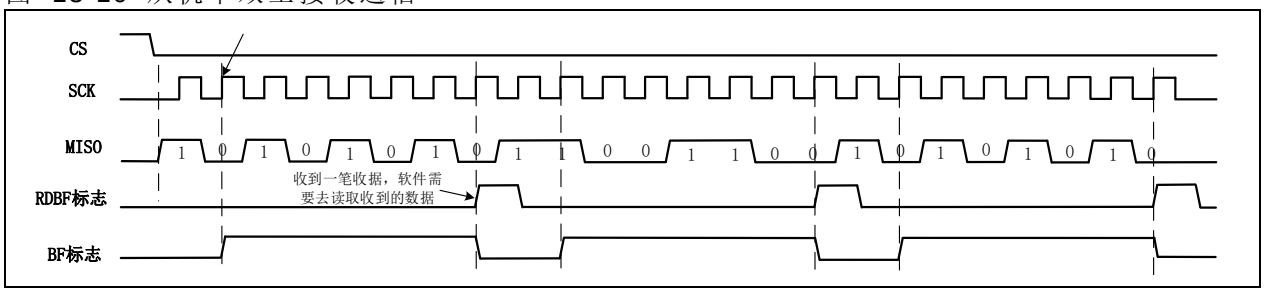
SLBTD=0：接收模式；

CLKPOL=0，CLKPHA=0：SCK 空闲输出低电平，第一个边沿为采样边沿；

FBN=0：帧数据的长度为 8 位；

从机接收数据：0xaa, 0xcc, 0xaa；

图 13-10 从机半双工接收通信



半双工通信-从机发送时序

MSTEN=0：设备为从机；

SLBEN=1：单线双向模式；

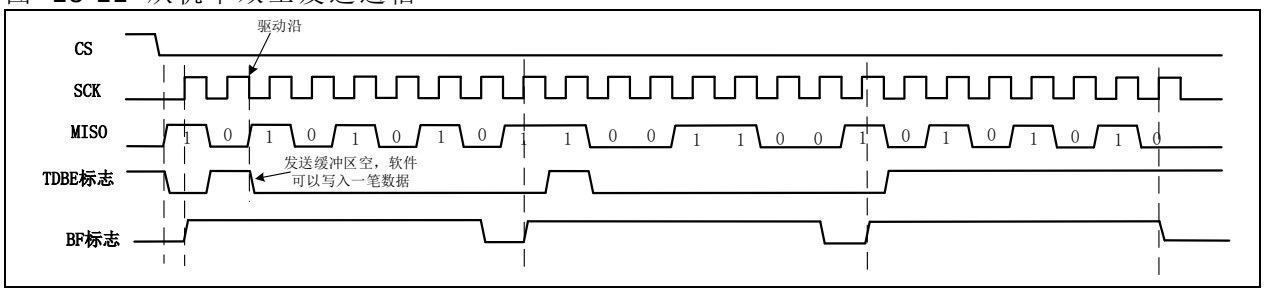
SLBTD=1：发送模式；

CLKPOL=0，CLKPHA=0：SCK 空闲输出低电平，第一个边沿为采样边沿；

FBN=0：帧数据的长度为 8 位；

从机发送数据：0xaa, 0xcc, 0xaa；

图 13-11 从机半双工发送通信



半双工通信-主机接收时序

MSTEN=1：设备为主机；

SLBEN=1：单线双向模式；

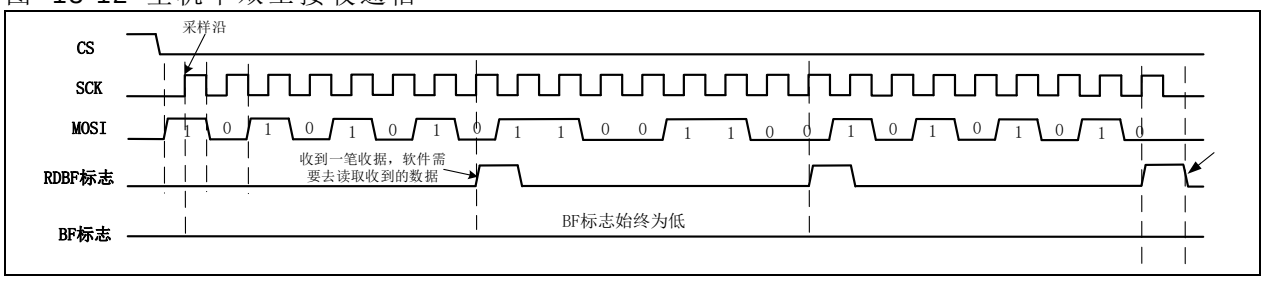
SLBTD=0：接收模式；

CLKPOL=0，CLKPHA=0：SCK 空闲输出低电平，第一个边沿为采样边沿；

FBN=0：帧数据的长度为 8 位；

主机接收数据：0xaa, 0xcc, 0xaa；

图 13-12 主机半双工接收通信

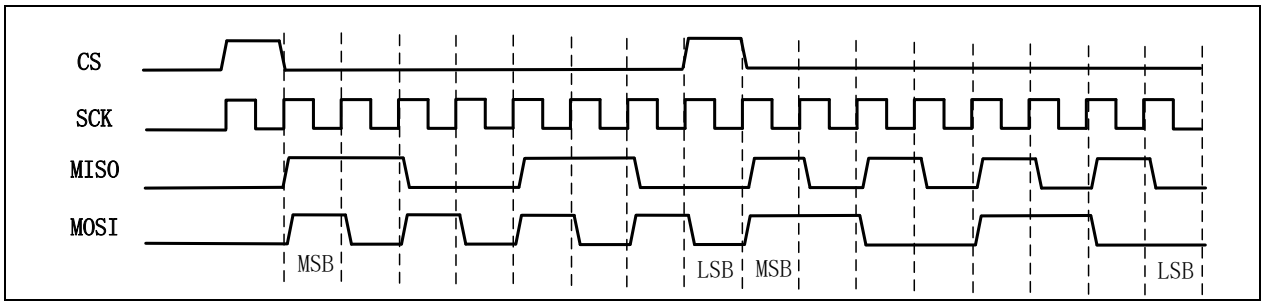


13.2.11 TI模式通信时序

SPI 接口支持 TI 模式，用户可以通过 TIEN 位置 1 使能该模式。

TI 模式下，连续与不连续通信的时序图稍有区别。当待发送数据在当前发送数据帧最后一位对应时钟 SCK 的上升沿之前写入，则通信连续，每笔数据的中间不存在 dummy CLK，主机在发送当前数据帧的最后一位数据的同时发出 CS 有效脉冲。

图 13-13 TI模式连续通信时序



TI 模式下，当待发送数据在当前发送数据帧最后一位对应时钟 SCK 的上升沿与下降沿之间写入，则每笔数据的中间存在一个 dummy CLK。

图 13-14 TI模式带dummy CLK的连续通信时序

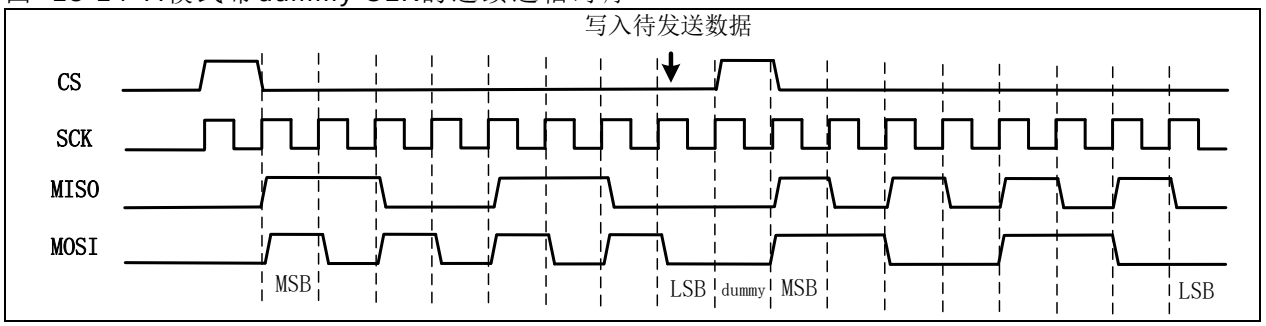
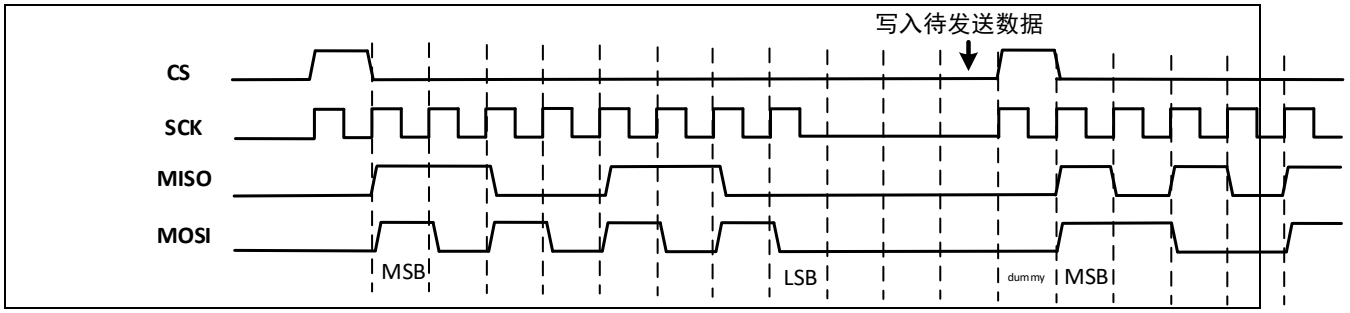
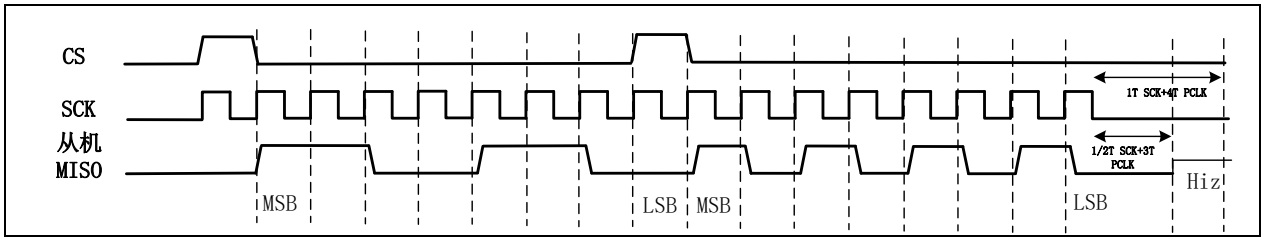


图 13-15 TI模式非连续通信时序



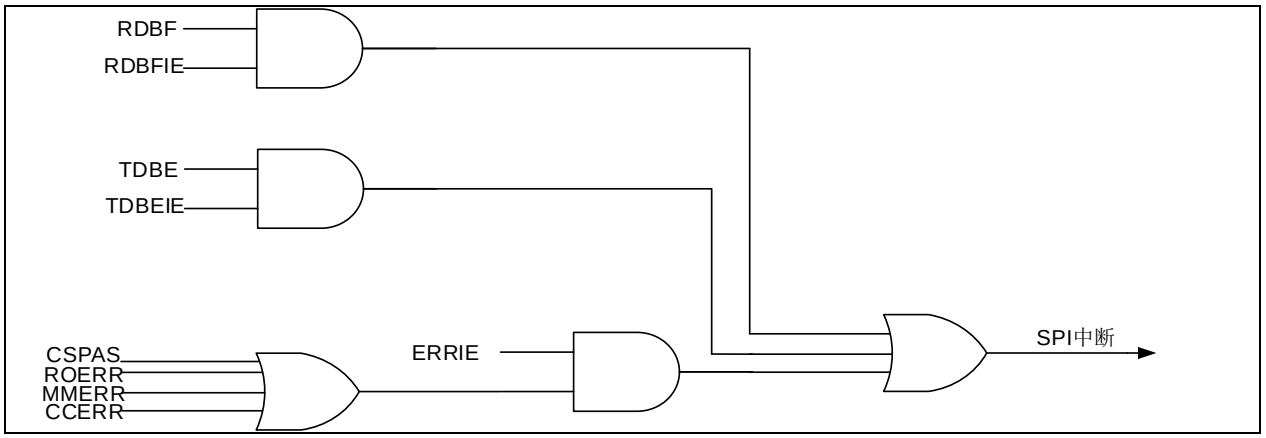
TI 模式下，当待发送数据在当前发送数据帧最后一位对应时钟 SCK 的下降沿之后写入，则主机固定在 1T SCK + 4T PCLK 后才能重新发出有效 SCK 时钟，从机在接收当前数据帧最后一位时，仍然没有检测到有效的 CS 脉冲，则在 1/2T SCK + 3T PCLK 后关闭 MISO 的输出功能，控制 MISO 浮空。

图 13-16 TI模式SCK及从机MISO释放时刻



13.2.12 中断

图 13-17 SPI中断



13.2.13 IO管脚控制

SPI 接口作为 SPI 使用时最多可有 4 根管脚与外设相连，各管脚的使用方法可以参见全双工半双工选择器简述和配置流程以及 CS 控制器简述和配置流程章节，各管脚的定义如下。

MISO：主机输入/从机输出管脚。在 SPI 接口作 SPI 主机使用时，从机送出的数据从该管脚输入。在 SPI 接口作 SPI 从机使用时，从机待发送的数据从该管脚输出。

MOSI：主设备输出/从设备输入管脚。在 SPI 接口作 SPI 主机使用时，主机待发送的数据从该管脚输出。在 SPI 接口作 SPI 从机使用时，主机送出的数据从该管脚输入。

SCK：SPI 的通信时钟管脚。在 SPI 接口作 SPI 主机使用时，通信时钟从此管脚输出送给外设。在 SPI 接口作 SPI 从机使用时，主机提供的通信时钟从该管脚输入以作为 SPI 接口的通信时钟。

CS：片选信号。这是一个可选的管脚，用来选中主/从设备，具体使用方式可以参见 CS 控制器章节。

13.2.14 注意事项

- CRC接收完成后要软件读DT寄存器来读出CRC值。

13.3 I²S功能描述

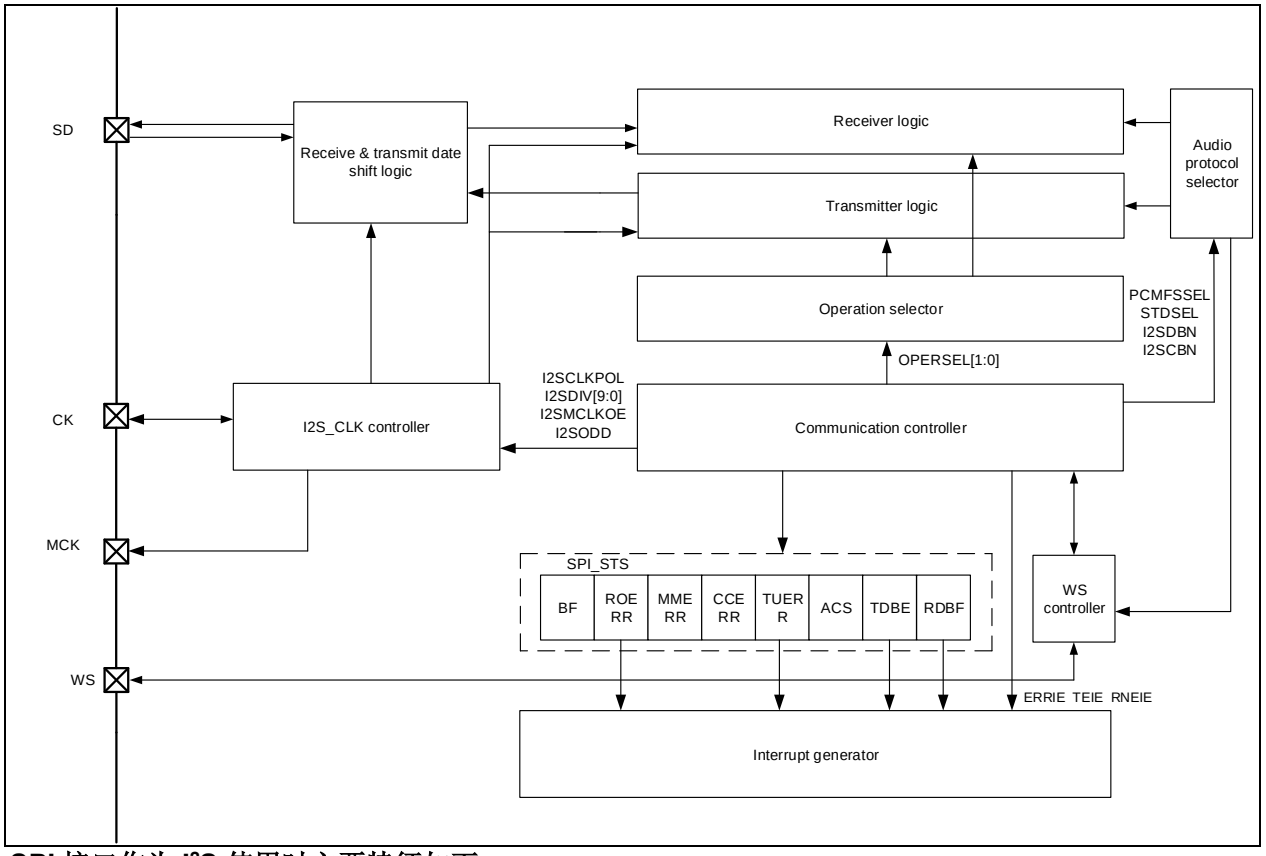
13.3.1 I²S简述

I²S 根据软件配置的不同，可以分别工作在主机接收，主机发送，从机接收，从机发送四种操作模式，并且可以分别支持包括飞利浦标准，高字节对齐标准，低字节对齐标准，PCM 标准在内的共四种音频标准，并同时支持 DMA 传输。

且通过两个 I²S 组合的形式，可以实现 I²S 全双工，具体使用方法请参考 I²S 全双工章节相关描述。

I²S 的框图如下图所示：

图 13-18 I²S框图



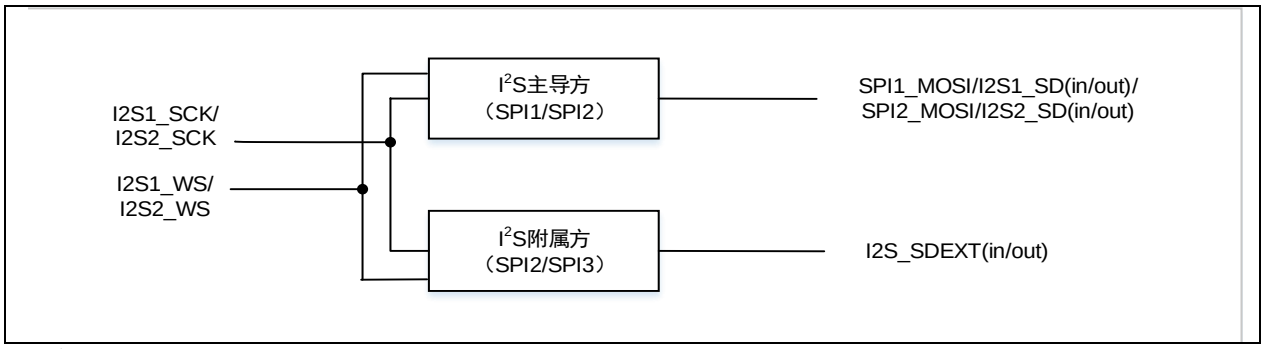
SPI 接口作为 I²S 使用时主要特征如下：

- 可编程配置的操作模式：
  - 从设备发送；
  - 从设备接收；
  - 主设备发送；
  - 主设备接收。
- 可编程配置的时钟极性。
- 可编程配置的时钟频率（8KHz到192KHz）。
- 可编程配置的数据位数（16位，24位，32位）。
- 可编程配置的声道位数（16位，32位）。
- 可编程配置的音频协议：
  - I²S飞利浦标准；
  - 高字节对齐标准（左对齐）；
  - 低字节对齐标准（右对齐）；
  - PCM标准（带长或短帧同步的通道帧）。
- 支持I²S全双工。
- 支持DMA传输。
- 支持提供频率固定比例为256倍Fs（音频采样频率）的外设主时钟。

13.3.2 I²S 全双工

可以通过设置SCFG中的SCFG_CFG2[31：30]将两个SPI组合在一起实现I²S全双工，三个SPI中，SPI1或SPI2可以被配置为全双工的主导方，SPI2或SPI3可以被配置为全双工的附属方，用户通过设置SCFG中的SCFG_CFG2[31：30]选择主导方和附属方的组合方式。在选择两个SPI组合成I²S全双工后，主导方的IO remap关系不变，附属方的SCK和WS在芯片内部和主导方的SCK和WS连在一起，附属方的SD线remap到I2S_SDEXT上。附属方原本的IO remap关系失效，对应IO被释放。

图 13-19 I²S全双工结构图



I²S 全双工主导方

可以配置为主机或从机模式，可以配置为发送或者接收方。

- I2Sx_WS线将参与通讯，用于实际通讯的WS信号交互；
- I2Sx_SCK线将参与通讯，用于实际通讯的时钟信号交互；
- I2Sx_SD线将参与通讯，用于实际通讯的主导方数据信息交互。

I²S 全双工附属方

只可以配置成从机模式，可以配置为发送或者接收方。

- I2Sy_WS不参与通讯，对应IO被释放；
- I2Sy_SCK线不参与通讯，对应IO被释放；
- I2Sy_SD线不参与通讯，对应IO被释放；
- I2S_SDEXT线将参与通讯，用于实际通讯的附属方数据信息交互。

注：x可以是1或2，y可以是2或3。

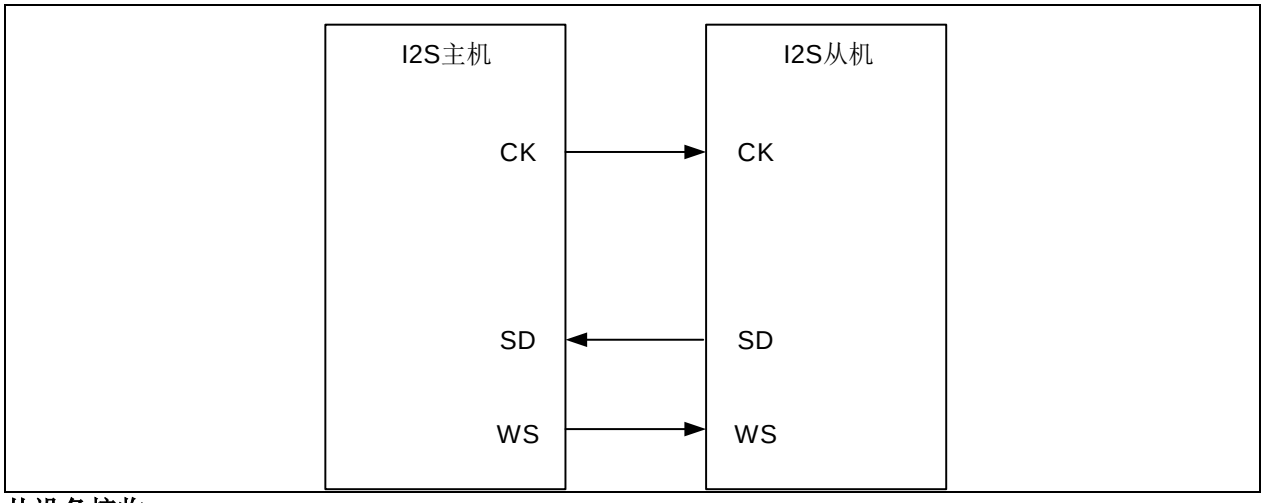
13.3.3 操作模式选择器简述和配置流程

SPI 接口作 I²S 选择器使用时提供了多种操作模式，用户可以通过软件编程控制操作模式选择器，选择需要的操作模式，本节会分从设备发送，从设备接收，主设备发送，主设备接收四种操作模式简单介绍配置流程以及连接方式。

从设备发送：

置位 I2SMSEL 位，配置 OPERSEL[1：0]位为 00，I²S 将工作在从设备发送模式下

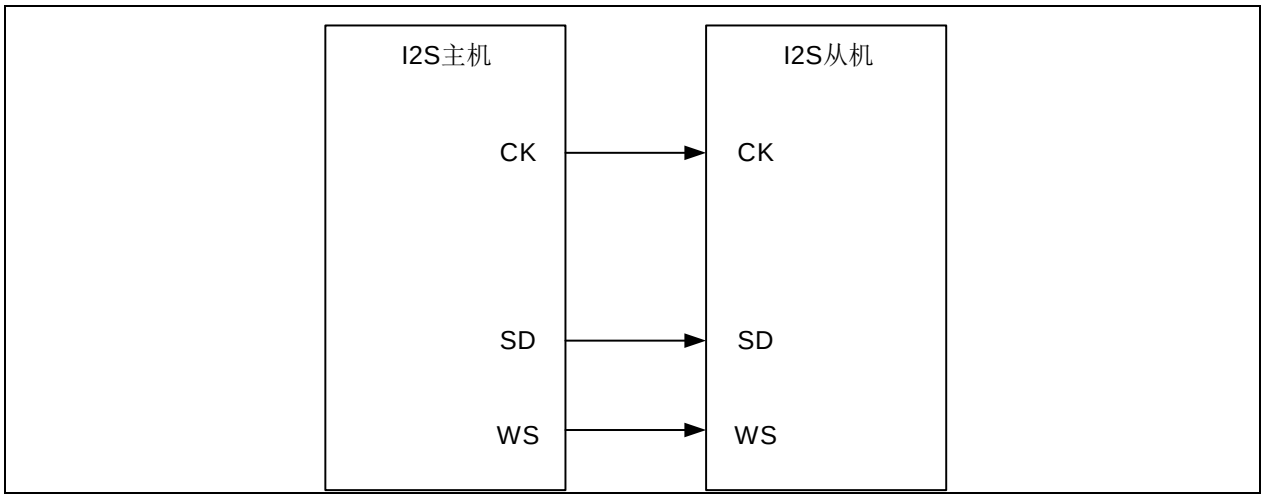
图 13-20 I²S从设备发送连接示意图



从设备接收：

置位 I2SMSEL 位，配置 OPERSEL[1：0]位为 01，I²S 将工作在从设备接收模式下。

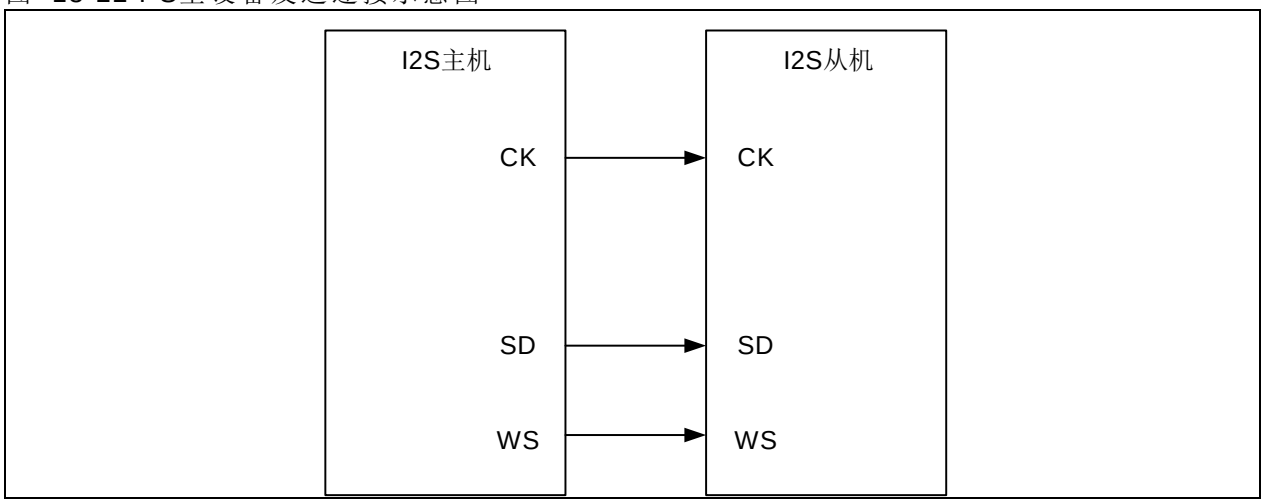
图 13-21 I²S从设备接收连接示意图



主设备发送：

置位 I2SMSEL 位，配置 OPERSEL[1：0]位为 10，I²S 将工作在主设备发送模式下。

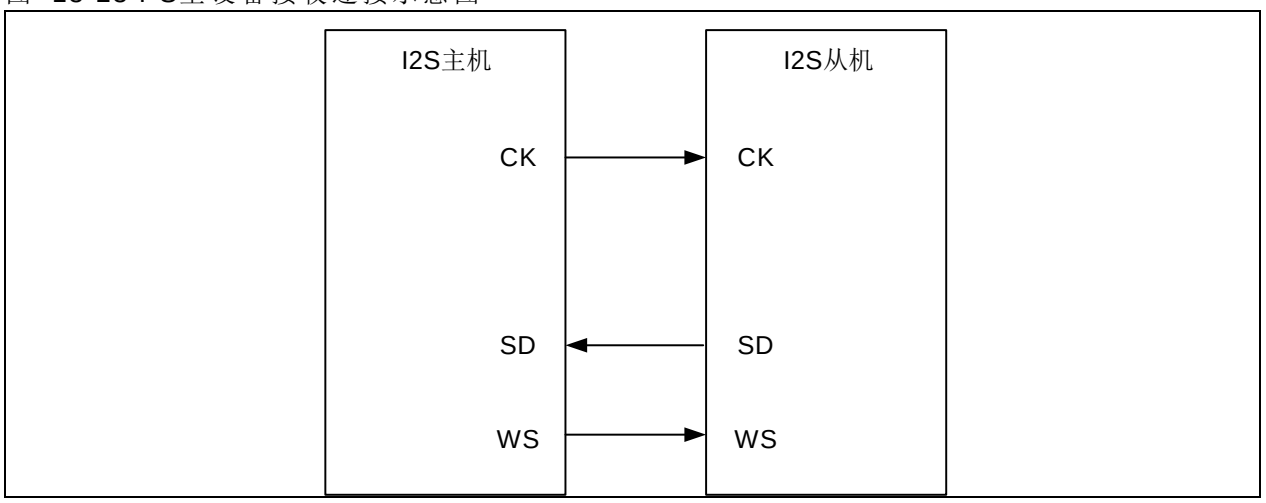
图 13-22 I²S主设备发送连接示意图



主设备接收：

置位 I2SMSEL 位，配置 OPERSEL[1：0]位为 11，I²S 将工作在主设备接收模式下。

图 13-23 I²S主设备接收连接示意图



13.3.4 音频协议选择器简述和配置流程

SPI接口作为I²S使用时支持多种音频协议，用户可以通过软件编程控制音频协议选择器选择需要的音频协议，数据位个数以及声道位个数同样由音频协议选择器控制，用户同样可以通过软件编程配置的方式选

择需要的数据位个数以及声道位个数，同时，音频协议选择器会自动控制WS控制器，输出或检测符合协议要求的WS信号，具体的配置流程如下。

- 音频协议选择：配置STDSEL位选择需要的音频协议：
  STDSLE=00：飞利浦标准；
  STDSLE=01：高字节对齐标准（左对齐）；
  STDSLE=10：低字节对齐标准（右对齐）；
  STDSLE=11：PCM标准。
- PCM帧同步格式选择：配置PCM长帧同步（PCMFSSEL=1）或短帧同步（PCMFSSEL=0）（该步骤在选择PCM协议时需要）。
- 数据位个数选择：配置I2SDBN位选择需要的数据位个数：
  I2SDBN=00：16位；
  I2SDBN =01：24位；
  I2SDBN =10：32位。
- 声道位个数选择：配置I2SCBN位选择需要的声道位个数：
  I2SDBN =0：16位；
  I2SDBN =1：32位。

需要注意的是，不同的音频协议以及不同的数据位数和声道位数组合所对应的数据写入方式存在较大不同，下面将依次罗列所有的允许的配置组合以及其数据的读写方式。

- 飞利浦标准或PCM标准或高字节或低字节标准，16位数据，16位声道
  数据位数和声道位数一致，每个声道只需读写一次SPI_DT寄存器，DMA传输个数为1。
- 飞利浦标准或PCM标准或高字节标准，16位数据，32位声道。
  数据位数和声道位数不一致，每个声道只需读写一次SPI_DT寄存器，DMA传输个数为1。只有前16位是有效数据，后16位数据硬件默认输出和接收0。
- 飞利浦标准或PCM标准或高字节标准，24位数据，32位声道。
  数据位数和声道位数不一致，每个声道需读写二次SPI_DT寄存器，DMA传输个数为2。前16位发送和接收第一笔16位数据，后16位发送和接收高8位数据，低8位数据硬件默认输出和接收0。
- 飞利浦标准或PCM标准或高字节或低字节标准，32位数据，32位声道。
  数据位数和声道位数一致，每个声道需读写二次SPI_DT寄存器，DMA传输个数为2。数据分两次，依次发送和接收16位数据。
- 低字节标准，16位数据，32位声道。
  数据位数和声道位数不一致，每个声道只需读写一次SPI_DT寄存器，DMA传输个数为1。只有后16位是有效数据，前16位数据硬件默认输出和接收0。
- 低字节标准，24位数据，32位声道。
  数据位数和声道位数不一致，每个声道需读写二次SPI_DT寄存器，DMA传输个数为2。前16位数据只有低八位有效，高八位数据硬件默认输出和接收0，后16位发送和接收第二笔16位数据

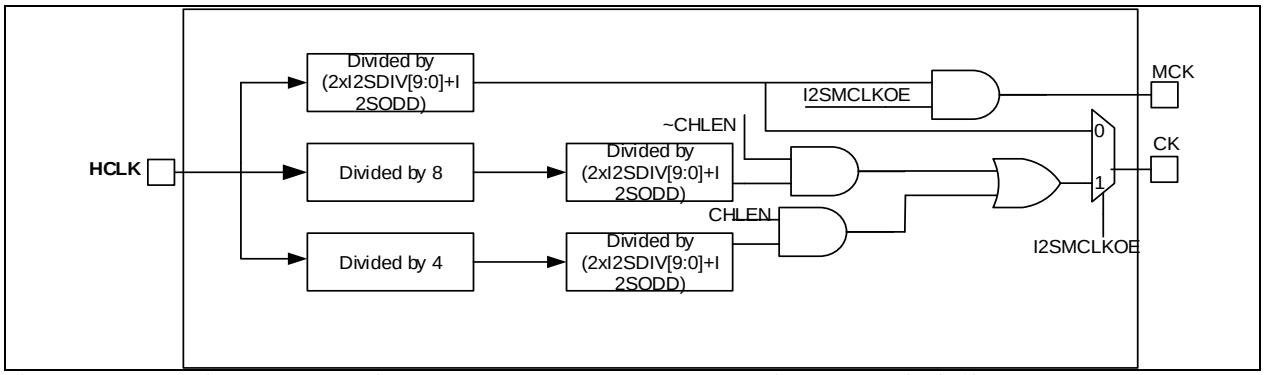
13.3.5 I2S_CLK控制器简述和配置流程

SPI 接口作 I²S 使用时，所有该接口支持的音频协议均为同步协议，作主机时，需要产生通信时钟用于 SPI 接口的数据收发，并且需要将该通信时钟通过 IO 输出给从机，用于从机的数据收发；作从机时，需要主机提供通信时钟从 IO 输入到 SPI 接口内部作为通信时钟使用，所以实际上，I2S_CLK 控制器便是扮演着产生 I2S_CLK 以及分配 I2S_CLK 的角色。

SPI 接口作 I²S 主机时支持提供通信时钟 CK 以及外设主时钟 MCK，CK 和 MCK 的来源如图 13-24 所示，CK 和 MCK 都是由 HCLK 分频得到，其中 MCK 的分频系数由 I2SDIV 以及 I2SODD 决定，具体计算公式见图 13-24。

CK 的分频系数与是否给外设提供主时钟有关，为了满足主时钟始终是音频采样频率的 256 倍，取决于是否提供主时钟以及声道位个数，当需要给外设提供主时钟时，CK 需要先做 8（I2SCBN=0 时）或 4（I2SCBN=1 时）的预分频，随后再做和 MCK 相同分频系数的分频得到最终的通信时钟 CK；如果不需要给外设提供主时钟，则 CK 的分频系数只由 I2SDIV 以及 I2SODD 决定，具体计算公式见图 13-24。

图 13-24 SPI作主机CK & MCK来源示意图



除了根据上面的描述自行配制想要的时钟外，我们也提供一些特定的时钟频率其对应的 I2SDIV, I2SODD 的值，以及相应的误差，用户可以直接按此表配置 I2SDIV 和 I2SODD。

表 13-1 使用系统时钟得到精确的音频频率

SCLK (MHz)	MCLK	Target Fs (Hz)	16bit				32bit			
			I2S DIV	I2S_ODD	RealFs	Error	I2S DIV	I2S_ODD	RealFs	Error
72	No	192000	6	0	187500	2.34%	3	0	187500	2.34%
72	No	96000	11	1	97826.09	1.90%	6	0	93750	2.34%
72	No	48000	32	1	34615.38	27.88%	11	1	48913.04	1.90%
72	No	44100	25	1	44117.65	0.04%	13	0	43269.23	1.88%
72	No	32000	35	0	32142.86	0.45%	17	1	32142.86	0.45%
72	No	22050	51	0	22058.82	0.04%	25	1	22058.82	0.04%
72	No	16000	70	1	15957.45	0.27%	35	0	16071.43	0.45%
72	No	11025	102	0	11029.41	0.04%	51	0	11029.41	0.04%
72	No	8000	140	1	8007.117	0.09%	70	1	7978.723	0.27%
72	Yes	96000	2	0	70312.5	26.76%	2	0	70312.5	26.76%
72	Yes	48000	3	0	46875	2.34%	3	0	46875	2.34%
72	Yes	44100	3	0	46875	6.29%	3	0	46875	6.29%
72	Yes	32000	4	1	31250	2.34%	4	1	31250	2.34%
72	Yes	22050	6	1	21634.62	1.88%	6	1	21634.62	1.88%
72	Yes	16000	9	0	15625	2.34%	9	0	15625	2.34%
72	Yes	11025	13	0	10817.31	1.88%	13	0	10817.31	1.88%
72	Yes	8000	17	1	8035.714	0.45%	17	1	8035.714	0.45%

13.3.6 DMA传输简述和配置流程

SPI 接口支持使用 DMA 进行发送数据的写入，接收数据的读取，由于无论 SPI 接口作 I²S 使用还是作 SPI 使用，对 DMA 来说，读写请求的来源都是同一个外设，所以实际上 SPI 接口作 I²S 使用时 DMA 传输的配置方法和作 SPI 使用并无不同，具体配置流程分别见下述的 DMA 发送配置流程以及 DMA 接收配置流程。

DMA 发送配置流程：

- 选择DMA传输通道：在DMA章节DMA通道映射表中选择用于当前所用SPI的DMA通道。
- 配置DMA传输目标地址：在DMA控制寄存器中DMA传输目的地址位写入当前所使用的SPI的SPI_DT寄存器地址，DMA将会在接收到发送请求后将待发送的数据写入该地址。
- 配置DMA传输源地址：在DMA控制寄存器中DMA传输源地址位写入待发送数据存放的地址，DMA将会在接收到发送请求后将该地址内的数据写入到目标地址中，即写入到当前所使用的SPI的SPI_DT寄存器中。
- 配置DMA传输数据个数：在DMA控制寄存器相关位置配置期望传输的数据个数。
- 配置DMA传输通道优先级：在DMA控制寄存器相关位置配置当前所使用通道的SPI的DMA传输通道优先级。

- 配置DMA中断产生时机：在DMA控制寄存器相关位置配置是在传输完成或传输完成一半时产生DMA中断。
- 使能DMA传输通道：在DMA控制寄存器相关位置使能当前所选用的DMA通道。

DMA接收配置流程：

- 选择DMA传输通道：在DMA章节DMA通道映射表中选择用于当前所用SPI的DMA通道。
- 配置DMA传输目标地址：在DMA控制寄存器中DMA传输目的地址位写入期望存放接收数据的地址，DMA将会在接收到接收请求后，将当前所使用的SPI的SPI_DT寄存器中的数据存放在目的地址中。
- 配置DMA传输源地址：在DMA控制寄存器中DMA传输源地址位写入当前所使用的SPI的SPI_DT寄存器的地址，DMA将会在接收到接收请求后将该地址内的数据写入到目标地址中，即写入到期望存放接收数据的地址。
- 配置DMA传输数据个数：在DMA控制寄存器相关位置配置期望传输的数据个数。
- 配置DMA传输通道优先级：在DMA控制寄存器相关位置配置当前所使用通道的SPI的DMA传输通道优先级。
- 配置DMA中断产生时机：在DMA控制寄存器相关位置配置是在传输完成或传输完成一半时产生DMA中断。
- 使能DMA传输通道：在DMA控制寄存器相关位置使能当前所选用的DMA通道。

13.3.7 发送器接收器简述和配置流程

由于无论 SPI 接口作 I²S 使用还是作 SPI 使用，对于 CPU 来说都是同一个外设，共用同一个基地址，并且 SPI 接口内部，作 I²S 使用和作 SPI 使用时，都共用同一个数据寄存器 SPI_DT，并且实际上发送器和接收器也是共用的，所以 SPI 接口的发送器和接收器只是根据通信控制器的配置发送和接收期望的数据帧格式，所以如 TDBE 和 RDBF 以及 ROERR 等状态标志，以及 TDBEIE 和 RDBFIE 以及 ERRIE 等中断使能位都是共用的。

但需要特别注意的是：

- I²S不支持CRC校验，所以和CRC有关的操作，以及CCERR标志和与之相对应的中断都不能使用。
- I²S协议需要解析当前的声道状态，用户可以根据ACS位判断当前传输是左声道（ACS=0）还是右声道（ACS=1）。
- I²S使用TUERR位表示当前是否发生欠载，TUERR=1，表示当前发送器出现了欠载错误，如果ERRIE置位，则产生中断。
- I²S在不同的音频协议和数据位数以及声道位数的组合下，操作SPI_DT寄存器的方式是不同的，具体可以参考音频协议选择器简述和配置流程部分描述。
- I²S的关闭方式同样需要特别注意，依据不同的配置方式罗列如下：
  - I2SDBN=00, I2SCBN=1，STDSLE=10：等待倒数第二个RDBF=1，等待17个CK周期，关闭I²S。
  - I2SDBN=00, I2SCBN=1，STDSLE=00或STDSLE=01或STDSLE=11：等待最后一个RDBF=1，等待一个CK时钟周期，关闭I²S。
  - 其它I2SDBN, I2SCBN,STDSLE组合： 等待倒数第二个RDBF=1，等待一个CK时钟周期，关闭I²S。

下面给出发送器和接收器的配置流程

I²S 发送器配置流程：

- 配置操作模式选择器。
- 配置音频协议选择器。
- 配置I2S_CLK控制器。
- 配置DMA(若需要开启DMA传输)。
- 置位I2SEN位开启I²S。
- 按上述方式配置I²SxEXT（若需要使用I²S全双工）。

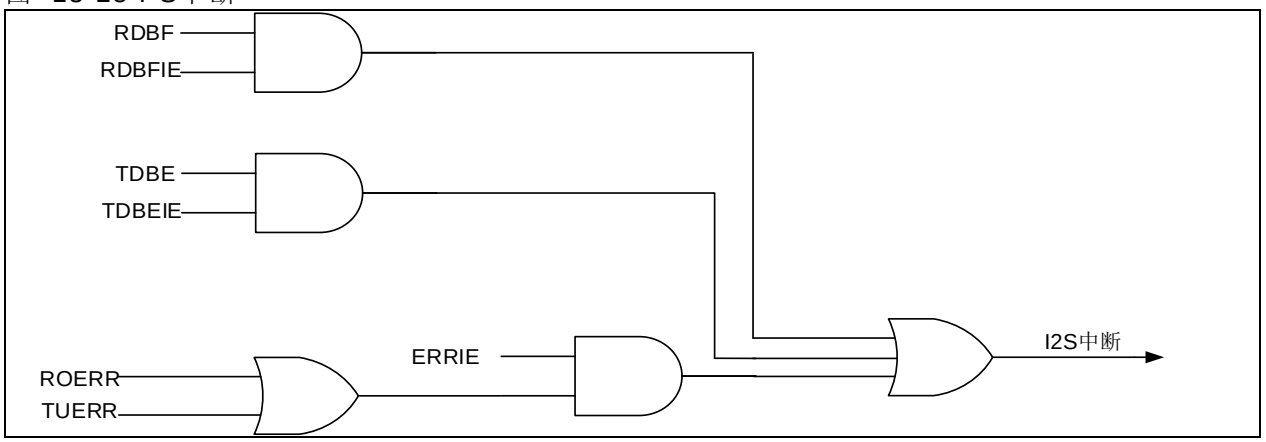
I²S 接收器配置流程：

- 配置操作模式选择器。
- 配置音频协议选择器。

- 配置I2S_CLK控制器。
- 配置DMA(若需要开启DMA传输)。
- 置位I2SEN位开启I²S。
- 按上述方式配置I²SxEXT（若需要使用I²S全双工）。

13.3.8 中断

图 13-25 I²S中断



13.3.9 I0管脚控制

SPI 接口作 I²S 使用时，I²S 传输需要三个管脚，分别是数据管脚 SD，同步管脚 WS，通信时钟管脚 CK，如果需要给外设提供主时钟还需要主时钟输出管脚 MCLK，由于一个 SPI 接口不可能同时作 I²S 和 SPI 使用，所以 I²S 和 SPI 部分管脚映射是共用的各管脚的映射和定义如下。

- SD：数据管脚（和MOSI管脚共用同样的GPIO映射关系），数据的双向收发管脚。
- WS：同步管脚（和CS管脚共用同样的GPIO映射关系），通信同步信号的双向控制管脚，主模式输出，从模式输入。
- CK：通信时钟管脚（和SCK管脚共用同样的GPIO映射关系），通信时钟双向输入输出管脚，主模式输出，从模式输入。
- MCLK：主时钟管脚（独立映射），主时钟输出管脚，用于给外设提供主时钟，输出的时钟频率固定为音频采样频率的256倍。

13.4SPI寄存器

必须用字（32 位）的方式操作这些外设寄存器。

表 13-2 SPI寄存器列表及其复位值

寄存器简称	基址偏移量	复位值
SPI_CTRL1	0x00	0x0000
SPI_CTRL2	0x04	0x0000
SPI_STS	0x08	0x0002
SPI_DT	0x0C	0x0000
SPI_CPOLY	0x10	0x0007
SPI_RCRC	0x14	0x0000
SPI_TCRC	0x18	0x0000
SPI_I2SCTRL	0x1C	0x0000
SPI_I2SCLKP	0x20	0x0002

13.4.1 SPI控制寄存器1（SPI_CTRL1）（I2S模式下不使用）

域	简称	复位值	类型	功能
位 15	SLBEN	0x0	rw	单线双向半双工模式使能（Single line bidirectional half-duplex enable） 0：关闭； 1：开启。
位 14	SLBTD	0x0	rw	单线双向半双工模式传输方向（Single line bidirectional half-duplex transmission direction） 和 SLBEN 位一起决定在“单线双向半双工”模式下数据的传输方向 0：接收模式； 1：发送模式。
位 13	CCEN	0x0	rw	CRC 校验使能（CRC calculation enable） 0：关闭； 1：开启。
位 12	NTC	0x0	rw	下一笔传输数据为 CRC（Next transmission CRC） 该位置起表示下一笔传输的数据为 CRC 数据。 0：普通数据； 1：CRC 数据。
位 11	FBN	0x0	rw	帧位个数（frame bit num） 该位配置发送/接收时数据帧位个数。 0：8 位； 1：16 位。
位 10	ORA	0x0	rw	仅接收有效（Only receive active） 在“双线单向”模式时，该位置起表示只有接收有效，发送被禁止。 0：发送和接收； 1：仅接收。
位 9	SWCSEN	0x0	rw	软件 CS 模式使能（Software CS enable） 当该位被置起时，CS 管脚上的电平由 SWCSIL 位的值决定，此时在 CS 管脚上的 I/O 电平状态无效。 0：关闭； 1：开启。
位 8	SWCSIL	0x0	rw	软件 CS 内部电平（Software CS internal level） 该位只在 SWCSEN 位置起时有意义，它决定了 CS 上的内部电平状态。 做主设备时，该位必须设置置起。 0：低电平； 1：高电平。
位 7	LTF	0x0	rw	LSB 先传输（LSB transmit first） 该位用于选择数据先传输 MSB 还是 LSB。 0：MSB； 1：LSB。
位 6	SPIEN	0x0	rw	SPI 使能（SPI enable） 0：关闭； 1：开启。
位 5：3	MDIV	0x0	rw	主模式时钟频率分频系数（Master clock frequency division） 作主模式时，分频系数对外设时钟进行分频，作为 SPI 时钟，MDIV[3]位在 SPI_CTRL2 寄存器，MDIV[3：0]： 0000：2 分频 0001：4 分频 0010：8 分频 0011：16 分频 0100：32 分频 0101：64 分频 0110：128 分频 0111：256 分频 1000：512 分频 1001：1024 分频

位 2	MSTEN	0x0	rw	主模式使能（Master enable） 0：关闭（从设备）； 1：开启（主设备）。
位 1	CLKPOL	0x0	rw	时钟极性（Clock polarity） 空闲时时钟输出的极性。 0：低电平； 1：高电平。
位 0	CLKPHA	0x0	rw	时钟相位（Clock phase） 0：第一个边沿进行数据捕获； 1：第二个边沿进行数据捕获。

注：在 I²S 模式下，SPI_CTRL1 寄存器需置 0。

13.4.2 SPI控制寄存器2（SPI_CTRL2）

域	简称	复位值	类型	功能
位 15：10	保留	0x00	resd	硬件强制为 0
位 9	MDIV3EN	0x0	rw	主模式时钟频率三分频使能（Master clock frequency3 division enable） 0：关闭； 1：开启。 注：该位开启时，MDIV[3：0]无效，SPI 时钟被强制为 PCLK 三分频。
位 8	MDIV[3]	0x0	rw	主模式时钟频率分频系数（Master clock frequency division） 详见 MDIV[2：0]在 SPI_CTRL1 寄存器。
位 7	TDBEIE	0x0	rw	发送数据缓冲器空中断使能（Transmit data buffer empty interrupt enable） 0：关闭； 1：开启。
位 6	RDBFIE	0x0	rw	接收数据缓冲器满中断使能（Receive data buffer full interrupt enable） 0：关闭； 1：开启。
位 5	ERRIE	0x0	rw	错误中断使能（Error interrupt enable） 当错误（CCERR、MMERR、ROERR、TUERR）产生时，该位控制是否产生中断 0：关闭； 1：开启。
位 4	TIEN	0x0	rw	TI 模式使能（TI mode enable） 0：关闭（Motorola 模式）； 1：开启（TI 模式）。 注：该位在 I2S 模式下没有用，在 I2S 模式下需保持为 0。
位 3	保留	0x0	resd	保持默认值。
位 2	HWCSOE	0x0	rw	硬件 CS 输出使能（Hardware CS output enable） 该位做主设备时才有意义，设置为’1’时，CS 脚 I/O 口输出低电平，设置为’0’时，必须保证 CS 脚 I/O 口输入为高电平。 0：关闭； 1：开启。
位 1	DMATEN	0x0	rw	DMA 发送使能（DMA transmit enable） 0：关闭； 1：开启。
位 0	DMAREN	0x0	rw	DMA 接收使能（DMA receive enable） 0：关闭； 1：开启。

13.4.3 SPI状态寄存器（SPI_STS）

域	简称	复位值	类型	功能
位 15：9	保留	0x00	resd	硬件强制为 0

位 8	CSPAS	0x0	ro	CS 脉冲异常置位标志（CS pulse abnormal setting flag） 0：无异常； 1：有异常置位； 注：该位用于 TI slave mode，由软件读 STS 寄存器清零。
位 7	BF	0x0	ro	通信忙标志（Busy flag） 0：通信空闲； 1：通信忙。
位 6	ROERR	0x0	ro	接收器溢出错误（Receiver overflow error） 0：无； 1：有。
位 5	MMERR	0x0	ro	主模式错误（Master mode error） 该位由硬件置位，软件清除（先读或写 SPI_STS 寄存器，再写 SPI_CTRL1 寄存器）。 0：无； 1：有。
位 4	CCERR	0x0	rw0c	CRC 校验错误（CRC calculation error） 该位由硬件置起，由软件清除。 0：正确； 1：错误。
位 3	TUERR	0x0	ro	发送器欠载错误（Transmitter underload error） 该位由硬件置起，软件清除（读 SPI_STS 寄存器）。 0：无； 1：有。 注：该位只在 I²S 模式使用。
位 2	ACS	0x0	ro	音频通道状态（Audio channel state） 该位表示当前传输的音频左右声道状态。 0：左声道； 1：右声道。 注：该位只在 I²S 模式使用。
位 1	TDBE	0x1	ro	发送数据缓冲器空（Transmit data buffer empty） 0：非空； 1：空。
位 0	RDBF	0x0	ro	接收数据缓冲器满（Receive data buffer full） 0：未满； 1：满。

13.4.4 SPI数据寄存器（SPI_DT）

域	简称	复位值	类型	功能
位 15：0	DT	0x0000	rw	数据值（Data value） 该寄存器包含读和写的功能，当数据位配置为 8 位时，该寄存器只有低 8 位[7：0]有效。

13.4.5 SPICRC多项式寄存器（SPI_CPOLY）（I2S模式下不使用）

域	简称	复位值	类型	功能
位 15：0	CPOLY	0x0007	rw	CRC 多项式寄存器（CRC polynomial） 该寄存器为 CRC 计算时用到的多项式，可以根据应用设置。 注：该寄存器只在 SPI 模式下使用。

13.4.6 SPIRxCRC寄存器（SPI_RCRC）（I2S模式下不使用）

域	简称	复位值	类型	功能
位 15：0	RCRC	0x0000	ro	接收 CRC 寄存器（receive CRC） CRC 使能后，该寄存器值为根据接收到的数据计算得到的 CRC 值，要复位该寄存器，需操作 SPI_CTRL1 的 CCEN 位先清除再置起。

				当数据位配置为 8 位时，该寄存器只有低 8 位[7：0]有效，按照 CRC8 计算；当数据位配置为 16 位时，按照 CRC16 计算。 注：该寄存器只在 SPI 模式下使用。

13.4.7 SPITxCRC寄存器（SPI_TCRC）

域	简称	复位值	类型	功能
位 15：0	TCRC	0x0000	ro	发送 CRC 寄存器（transmit CRC） CRC 使能后，该寄存器值为根据发送的数据计算得到的 CRC 值。要复位该寄存器，需操作 SPI_CTRL1 的 CCEN 位先清除再置起。 当数据位配置为 8 位时，该寄存器只有低 8 位[7：0]有效，按照 CRC8 计算；当数据位配置为 16 位时，按照 CRC16 计算。 注：该寄存器只在 SPI 模式下使用。

13.4.8 SPI_I2S配置寄存器（SPI_I2SCTRL）

域	简称	复位值	类型	功能
位 15：12	保留位	0x0	resd	硬件强制为 0
位 11	I2SMSEL	0x0	rw	I^2S 模式选择（I^2S mode select） 0：SPI 模式； 1：I^2S 模式。
位 10	I2SEN	0x0	rw	I^2S 使能（I^2S enable） 0：关闭； 1：开启。
位 9：8	OPERSEL	0x0	rw	I^2S 操作选择（I^2S operation select） 00：从设备发送； 01：从设备接收； 10：主设备发送； 11：主设备接收。
位 7	PCMFSSEL	0x0	rw	PCM 帧同步（PCM frame synchronization select） 该位只在使用 PCM 标准时才有意义。 0：短帧同步； 1：长帧同步。
位 6	保留位	0x0	resd	保持默认值。
位 5：4	STDSEL	0x0	rw	I^2S 标准选择（I^2S standard select） 00：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。
位 3	I2SCLKPOL	0x0	rw	I^2S 时钟极性（I^2S clock polarity） 时钟管脚上总线空闲时时钟输出的极性。 0：低电平； 1：高电平。
位 2：1	I2SDBN	0x0	rw	I^2S 数据位个数（I^2S data bit num） 00：16 位； 01：24 位； 10：32 位； 11：不允许。
位 0	I2SCBN	0x0	rw	I^2S 声道位个数（I^2S channel bit num） 该位只有在 I^2S 数据位个数为 16 位时配置才有意义，否则都由硬件固定为 32 位。 0：16 位宽； 1：32 位宽。

13.4.9 SPI_I2S预分频寄存器（SPI_I2SCLKP）

域	简称	复位值	类型	功能

位 15：12	保留位	0x0	resd	硬件强制为 0
位 9	I2SMCLKOE	0x0	rw	I²S 主设备时钟输出使能（I²S Master clock output enable） 0：关闭； 1：开启。
位 8	I2SODD	0x0	rw	I²S 分频系数配置奇数（Odd result for I²S division） 0：实际分频系数=I2SDIV*2； 1：实际分频系数=(I2SDIV*2)+1。
位 11：10 位 7：0	I2SDIV	0x02	rw	I²S 分频系数（I²S division） I2SDIV[9：0]禁止设置为 0 或者 1。

14 定时器（TIMER）

AT32F423 定时器种类有基本定时器、通用定时器、高级控制定时器，详细功能模式可参考 14.1~14.5 节说明，下表为各种类型定时器的功能总表。

表 14-1 TMR功能对比

Timer 类型	Timer	计数位数	计数方式	重复计数器	预分频系数	DMA 请求产生	捕获/比较通道	PWM 输入模式	ETR 输入	刹车输入
高级控制定时器	TMR1	16	向上 向下 向上/向下	16 位	1~65536	支持	4	支持	支持	支持
通用定时器	TMR2	16/32	向上 向下 向上/向下	不支持	1~65536	支持	4	支持	支持	不支持
	TMR3 TMR4	16	向上 向下 向上/向下	不支持	1~65536	支持	4	支持	支持	不支持
	TMR9 TMR12	16	向上 向下 向上/向下	8 位	1~65536	支持	2	支持	不支持	支持
	TMR10 TMR11 TMR13 TMR14	16	向上 向下 向上/向下	8 位	1~65536	支持	1	不支持	不支持	支持
基本定时器	TMR6 TMR7	16	向上	不支持	1~65536	支持	不支持	不支持	不支持	不支持

Timer 类型	Timer	计数位数	计数方式	PWM 输出	单周期输出	互补输出	死区	编码器接口连接	霍尔传感器接口连接	连动外设
高级控制定时器	TMR1	16	向上 向下 向上/向下	支持	支持	支持	支持	支持	支持	定时器同步/ADC
通用定时器	TMR2	16/32	向上 向下 向上/向下	支持	支持	不支持	不支持	支持	支持	定时器同步/ADC/DAC
	TMR3 TMR4	16	向上 向下 向上/向下	支持	支持	不支持	不支持	支持	支持	定时器同步/ADC/DAC
	TMR9 TMR12	16	向上 向下 向上/向下	支持	支持	支持	支持	支持	不支持	定时器同步 ADC/DAC
	TMR10 TMR11 TMR13 TMR14	16	向上 向下 向上/向下	支持	支持	支持	支持	不支持	不支持	定时器同步
基本定时器	TMR6 TMR7	16	向上	不支持	不支持	不支持	不支持	不支持	不支持	ADC/DAC

14.1基本定时器（TMR6和TMR7）

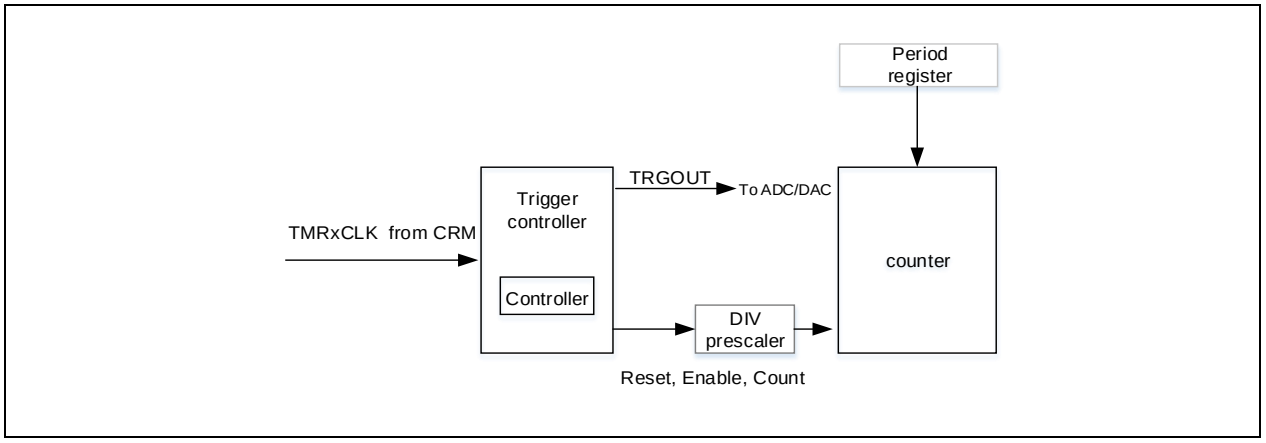
14.1.1 TMR6和TMR7简介

基本定时器（TMR6 和 TMR7）包含一个 16 位向上计数器以及对应的控制逻辑，没有外部 I/O 接入。可用于简单的定时功能。

14.1.2 TMR6和TMR7的主要功能

- 16位向上计数器，可自动装载
- 16位预分频器，用于对TMR_CLK时钟分频，分频系数为1～65536之间的任意数值
- 触发DAC的同步电路（TMR6和TMR7独有的特性）

图 14-1 基本定时器框图

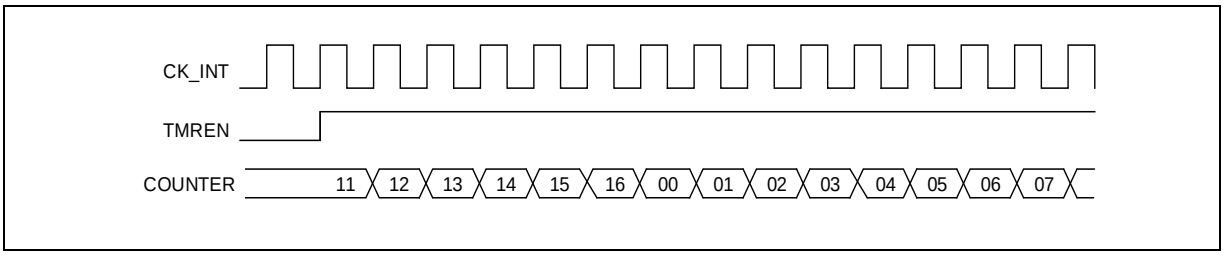


14.1.3 TMR6和TMR7的功能

14.1.3.1 计数时钟

TMR6 和 TMR7 由内部时钟源（CK_INT）经由预分频器提供计数器计数。当 TMR 对应的 APB 时钟预分频系数是 1 时，CK_INT 频率等于 APB 时钟频率，否则 CK_INT 频率等于 APB 时钟频率的 2 倍。

图 14-2 使用CK_INT且分频系数为1



14.1.3.2 计数模式

基本定时器仅提供向上计数模式。其内部拥有一个 16 位计数器。

TMRx_PR 寄存器用于设置计数器计数周期。默认 TMRx_PR 寄存器值会立即传入它的影子寄存器；当开启周期缓冲功能后（PRBEN 置 1），TMRx_PR 寄存器值在溢出事件发生时传入它的影子寄存器。

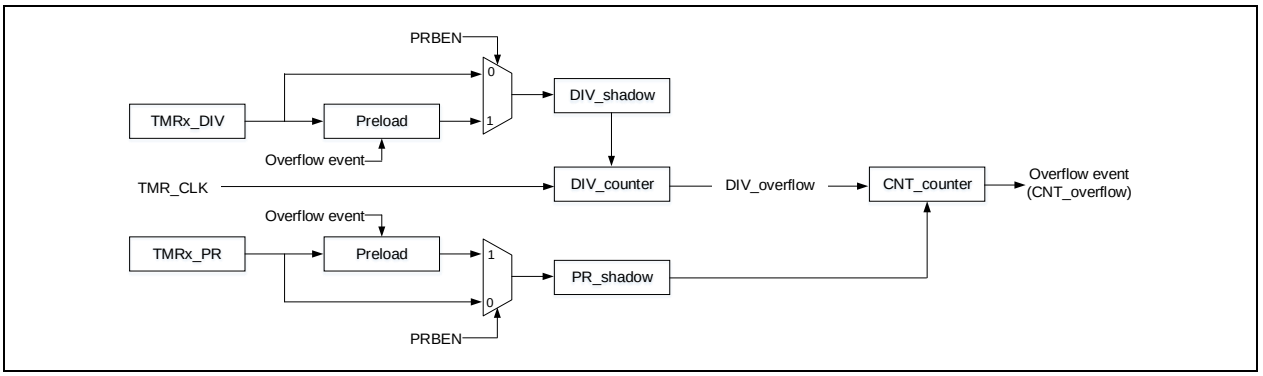
TMRx_DIV 寄存器用于设置计数器计数频率，每（DIV[15:0]+1）个计数时钟周期，计数器计数一次。和 TMRx_PR 寄存器类似，开启周期缓冲功能后，TMRx_DIV 寄存器值在溢出事件时更新至它的影子寄存器。

读取 TMRx_CNT 寄存器会返回当前计数器计数值，写入 TMRx_CNT 寄存器会更新计数器当前计数值为写入值。

默认允许产生溢出事件，设置 TMRx_CTRL1 寄存器 OVFEN=1 将禁止溢出事件产生。TMRx_CTRL1 寄存器 OVFS 用于选择溢出事件来源，默认计数器上溢或下溢、置位 OVFSWTR、复位模式次定时器控制器产生的复位信号产生溢出事件。置位 OVFS 后，只有计数器上溢或下溢产生溢出事件。

TMREN 位置 1 将使能定时器计数，由于同步逻辑，实际驱动计数器的使能信号 TMR_EN 相对于 TMREN 延迟一个时钟周期。

图 14-3 计数器基本结构



向上计数模式

上计数模式计数值达到 TMRx_PR 值时，重新从 0 向上计数，计数器上溢并产生溢出事件，同时 OVFIF 位置 1。若禁止产生溢出事件，计数器溢出后不再重载预分频值和周期值，否则预分频值和周期值在溢出事件后更新。

图 14-4 PRBEN=0时的溢出事件

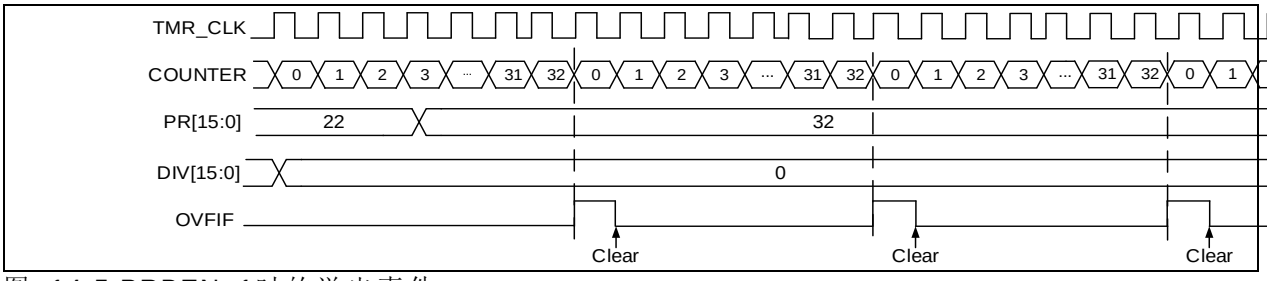


图 14-5 PRBEN=1时的溢出事件

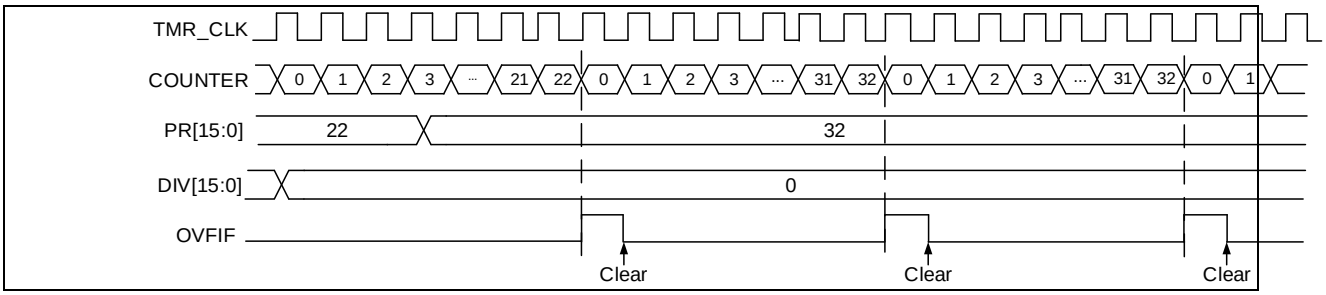
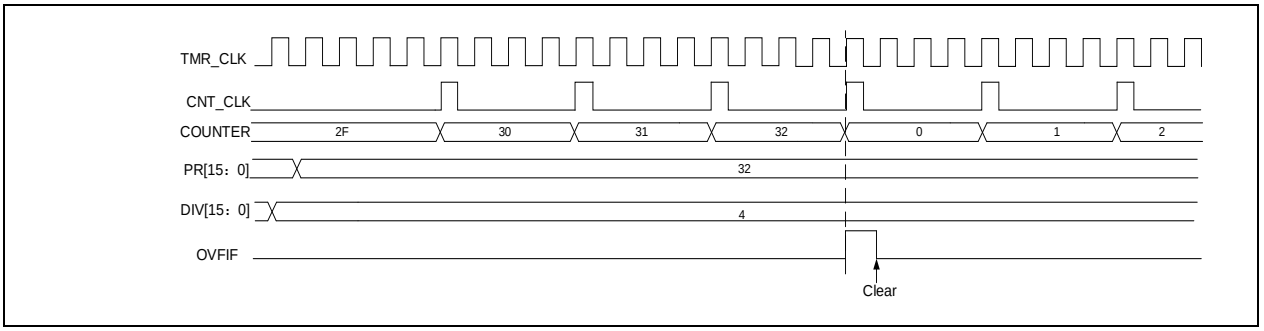


图 14-6 计数器时序图，内部时钟分频因子为4



14.1.3.3 调试模式

当微控制器进入调试模式（Cortex®-M4F 核心停止）时，将 DEBUG 模块中的 TMRx_PAUSE 置 1，可以使 TMRx 计数器暂停计数。

14.1.4 TMR6和TMR7寄存器

必须用字（32 位）的方式操作这些外设寄存器。

下表中将 TMR6、7 的所有寄存器映射到一个 16 位可寻址（编址）空间。

表 14-2 TMR6和TMR7寄存器和复位值

寄存器简称	基址偏移量	复位值
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000

14.1.4.1 TMR6和TMR7控制寄存器1（TMRx_CTRL1）

域	简称	复位值	类型	功能
位 15：8	保留	0x00	resd	保持默认值。
位 7	PRBEN	0x0	rw	周期缓冲使能（Period buffer enable） 0：缓冲关闭； 1：缓冲开启。
位 6：4	保留	0x0	resd	保持默认值。
位 3	OCMEN	0x0	rw	单周期使能 （One cycle mode enable） 该功能用于选择溢出事件后，计数器是否停止。 0：关闭； 1：开启。
位 2	OVFS	0x0	rw	溢出事件源选择 （Overflow event source） 配置溢出事件或 DMA 请求来源。 0：来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件； 1：只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 （Overflow event enable） 该位用于允许或禁止溢出事件（OEV）产生。 0：允许溢出事件产生，溢出事件可以由下列事件产生： - 计数器溢出 - 将 OVFSWTR 位置 1 - 通过次定时器控制器产生的溢出事件 1：禁止溢出事件产生。 如果将 OVFSWTR 位置 1 或次定时器控制器产生了一个硬件复位，则计数器和预分频器将被重新初始化。 注：该位由软件置 1 和清 0。
位 0	TMREN	0x0	rw	使能定时器（TMR enable） 0：关闭； 1：开启。

14.1.4.2 TMR6和TMR7控制寄存器2（TMRx_CTRL2）

域	简称	复位值	类型	功能
位 15：7	保留	0x000	resd	保持默认值。
位 6：4	PTOS	0x0	rw	主定时器输出信号选择 （Primary TMR output selection） TMRx 输出到次定时器的信号选择： 000：复位； 001：使能； 010：更新；
位 3：0	保留	0x0	resd	保持默认值。

14.1.4.3 TMR6和TMR7 DMA/中断使能寄存器（TMRx_IDEN）

域	简称	复位值	类型	功能
位 15：9	保留	0x00	resd	保持默认值。

位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能（overflow event DMA request enable） 0：关闭； 1：开启。
位 7：1	保留	0x00	resd	保持默认值。
位 0	OVFIEN	0x0	rw	溢出中断使能（Overflow interrupt enable） 0：关闭； 1：开启。

14.1.4.4 TMR6和TMR7中断状态寄存器（TMRx_ISTS）

域	简称	复位值	类型	功能
位 15：1	保留	0x0000	resd	保持默认值。
位 0	OVFIF	0x0	rw0c	溢出中断标记 （Overflow interrupt flag） 当溢出事件发生时由硬件置'1'，由软件清'0'。 0：无溢出事件发生； 1：发生溢出事件，若 TMRx_CTRL1 的 OVFEN=0、OVFS=0 时： – 当 TMRx_SWEVE 寄存器的 OVFG=1 时产生溢出事件； – 当计数值 CVAL 被触发事件重初始化时产生溢出事件。

14.1.4.5 TMR6和TMR7软件事件寄存器（TMRx_SWEVT）

域	简称	复位值	类型	功能
位 15：1	保留	0x0000	resd	保持默认值。
位 0	OVFSWTR	0x0	rw0c	软件触发溢出事件（Overflow event triggered by software） 通过软件触发一个溢出事件。 0：无作用； 1：制造一个溢出事件。

14.1.4.6 TMR6和TMR7计数值（TMRx_CVAL）

域	简称	复位值	类型	功能
位 15：0	CVAL	0x0000	rw	计数值 （Counter value）

14.1.4.7 TMR6和TMR7分频系数（TMRx_DIV）

域	简称	复位值	类型	功能
位 15：0	DIV	0x0000	rw	分频系数 （Divider value） 计数器时钟频率 $f_{CK\_CNT} = f_{TMR\_CLK}$ /（DIV[15：0]+1）。 DIV 为溢出事件发生时写入的分频系数。

14.1.4.8 TMR6和TMR7周期寄存器（TMRx_PR）

域	简称	复位值	类型	功能
位 15：0	PR	0x0000	rw	周期值（Period value） 定时器计数的周期值。当周期值为 0 时，定时器不工作。

14.2通用定时器（TMR2至TMR4）

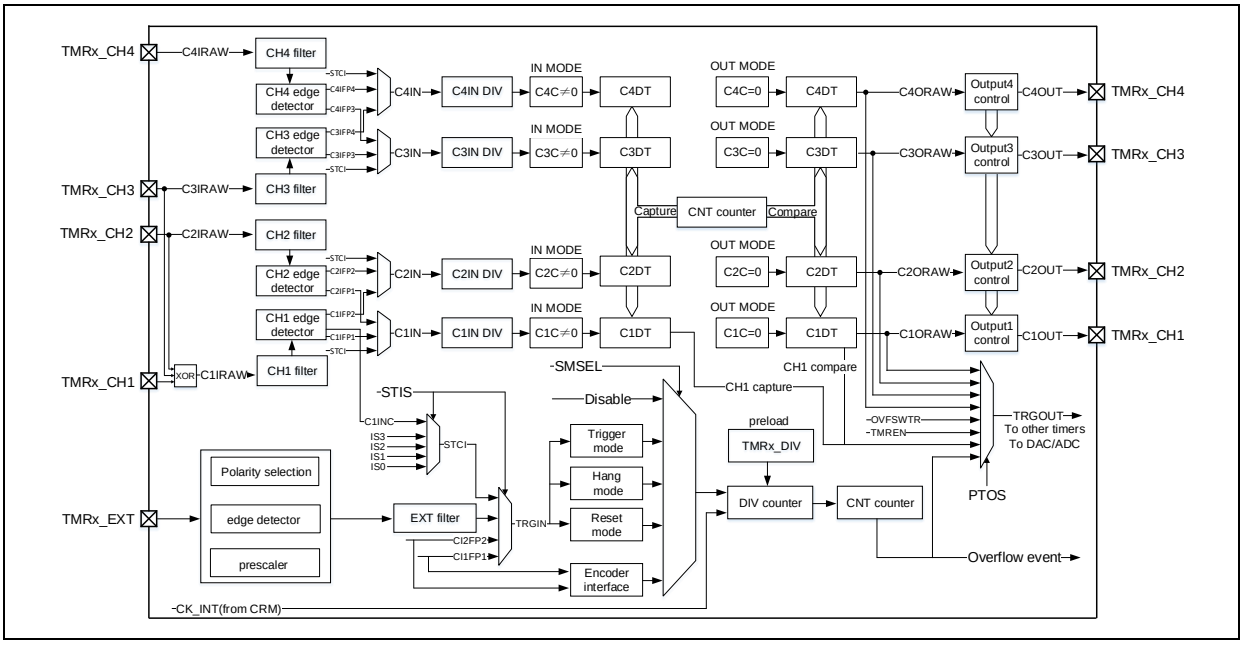
14.2.1 TMR2至TMR4简介

通用定时器 TMR2 至 TMR4 包含一个支持向上、向下、双向计数的 16 位计数器(TMR2 可扩展至 32 位)、4 个捕获/比较寄存器、4 组独立的通道。可实现输入捕获、可编程 PWM 输出。

14.2.2 TMR2至TMR4主要功能

- 可选内部、外部、内部触发输入用作计数时钟
- 16 位支持向上、向下、双向、编码器模式的计数器（TMR2 可扩展至 32 位）
- 4 组独立通道，支持输入捕获、输出比较、PWM 生成、单周期模式。
- 定时器之间可互联同步
- 支持溢出事件、触发事件、通道事件触发中断/DMA
- 支持 TMR burst DMA 传输

图 14-7 通用定时器框图

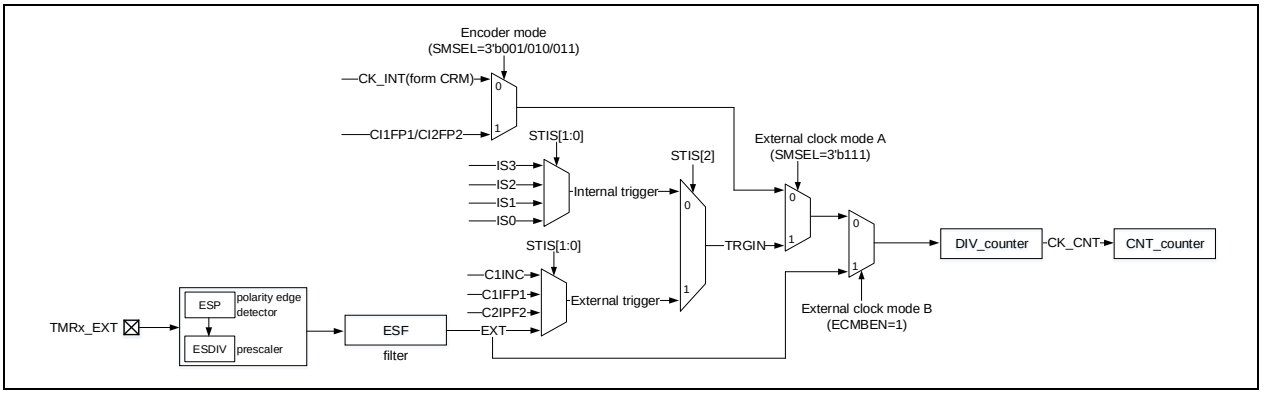


14.2.3 TMR2至TMR4功能描述

14.2.3.1 计数时钟

TMR2 至 TMR4 计数时钟可从内部时钟（CK_INT）、外部时钟（外部时钟模式 A、B）、内部触发输入（ISx）这些时钟源提供。

图 14-8 计数时钟

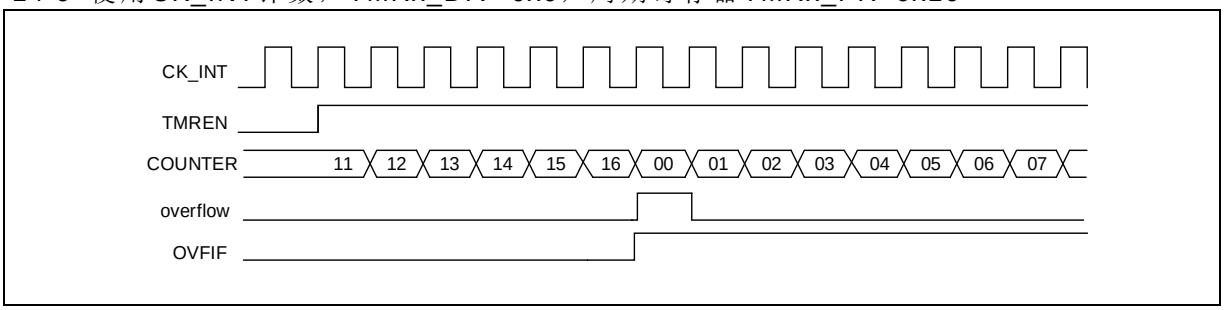


内部时钟（CK_INT）

默认下使用 CK_INT 经由预分频器驱动计数器计数，当 TMR 对应的 APB 时钟预分频系数是 1 时，CK_INT

频率等于 APB 时钟频率，否则 CK_INT 频率等于 APB 时钟频率的 2 倍。相关配置流程如下：
-配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]，选择计数模式，若选择单向对齐计数模式，还需配置 TMRx_CTRL1 寄存器 OWCDIR 选择计数方向。
-配置 TMRx_DIV 寄存器，设置计数器计数频率。
-配置 TMRx_PR 寄存器，设置计数器计数周期。
-配置 TMRx_CTRL1 寄存器 TMREN，使能计数器。

图 14-9 使用CK_INT计数，TMRx_DIV=0x0，周期寄存器TMRx_PR=0x16



外部时钟（TRGIN/EXT）

计数时钟可由两种外部时钟源提供，分别为 TRGIN 和 EXT 信号。

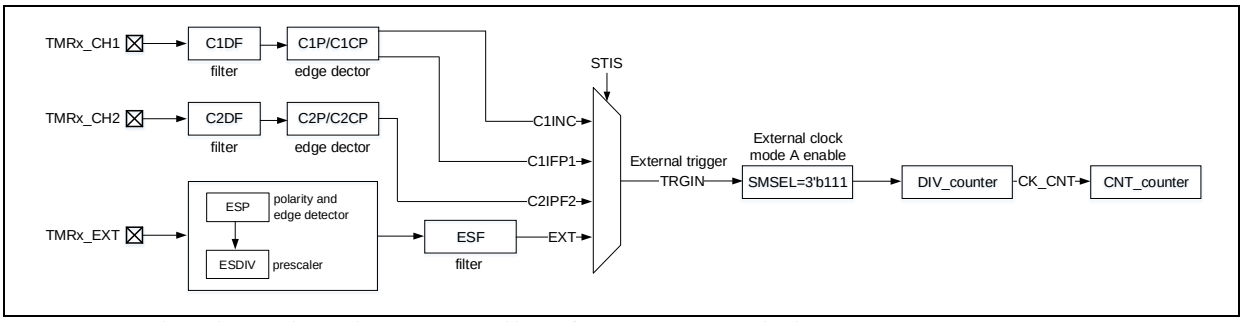
当 SMSEL=3'b111 时，外部时钟模式 A 被选中，配置 STIS[2：0]来选择外部时钟源 TRGIN 信号驱动计数器计数。外部时钟源 TRGIN 可选则 C1INC（STIS=3'b100，通道 1 上升沿和下降沿信号）、C1IFP1（STIS=3'b101，通道 1 滤波且极性选择后信号）、C2IFP2（STIS=3'b110，通道 2 滤波且极性选择后信号）和 EXT（STIS=3'b111，外部输入经极性选择、分频和滤波后信号）。

当 ECMBEN=1 时，外部时钟模式 B 被选中，计数器由外部输入经极性选择、分频和滤波后 EXT 信号驱动计数。外部时钟模式 B 等效于外部时钟模式 A 选择 EXT 信号作为外部时钟源 TRGIN。

若要使用外部时钟模式 A，可按如下步骤配置：
-配置外部时钟源 TRGIN 参数。
若选择 TRGIN 来源为 TMRx_CH1，需配置通道 1 输入滤波（TMRx_CM1 寄存器 C1DF[3:0]）和通道 1 输入极性（TMRx_CCTRL 寄存器 C1P/C1CP）。
若选择 TRGIN 来源为 TMRx_CH2，需配置通道 2 输入滤波（TMRx_CM1 寄存器 C2DF[3:0]）和通道 1 输入极性（TMRx_CCTRL 寄存器 C2P/C2CP）。
若选择 TRGIN 来源为 TMRx_EXT，需配置外部信号极性（TMRx_STCTRL 寄存器 ESP）、外部信号分频（TMRx_STCTRL 寄存器 ESDIV[1:0]）和外部信号滤波（TMRx_STCTRL 寄存器 ESF[3:0]）。
-配置 TMRx_STCTRL 寄存器 STIS[1:0]，设置 TRGIN 信号来源。
-配置 TMRx_STCTRL 寄存器 SMSEL=3'b111，使能外部时钟模式 A。
-配置 TMRx_DIV 寄存器 DIV[15:0]，设置计数器计数频率。
-配置 TMRx_PR 寄存器 PR[15:0]，设置计数器计数周期。
-配置 TMRx_CTRL1 寄存器 TMREN，使能计数器。

若要使用外部时钟模式 B，可按如下步骤配置：
-配置 TMRx_STCTRL 寄存器 ESP，设置外部信号极性。
-配置 TMRx_STCTRL 寄存器 ESDIV[1:0]，设置外部信号分频。
-配置 TMRx_STCTRL 寄存器 ESF[3:0]，设置外部信号滤波。
-配置 TMRx_STCTRL 寄存器 ECMBEN，使能外部时钟模式 B。
-配置 TMRx_DIV 寄存器 DIV[15:0]，设置计数器计数频率。
-配置 TMRx_PR 寄存器 PR[15:0]，设置计数器计数周期。
-配置 TMRx_CTRL1 寄存器 TMREN，使能计数器。

图 14-10 外部时钟模式A框图



注：由于同步逻辑，输入端信号与计数器实际时钟之间存在一定延时。

图 14-11 使用外部时钟模式A计数，PR=0x32，DIV=0x0

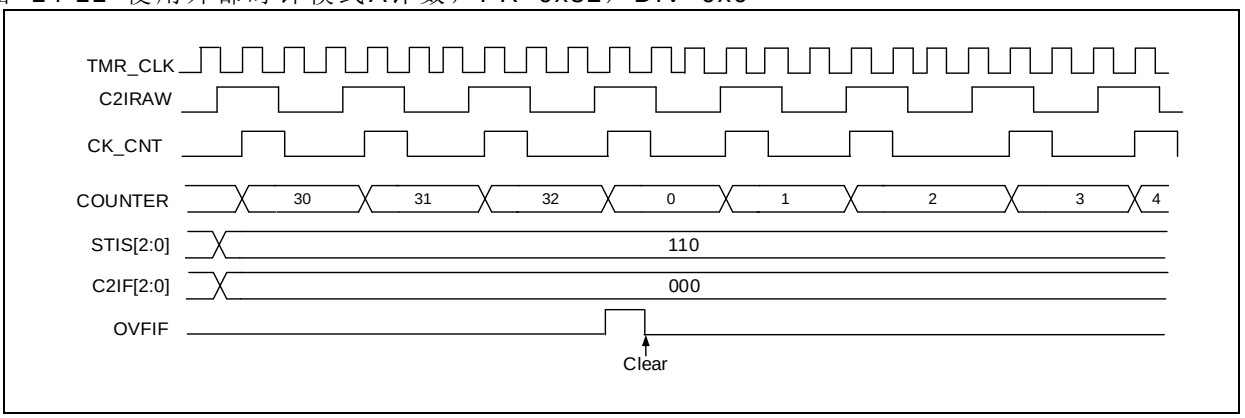
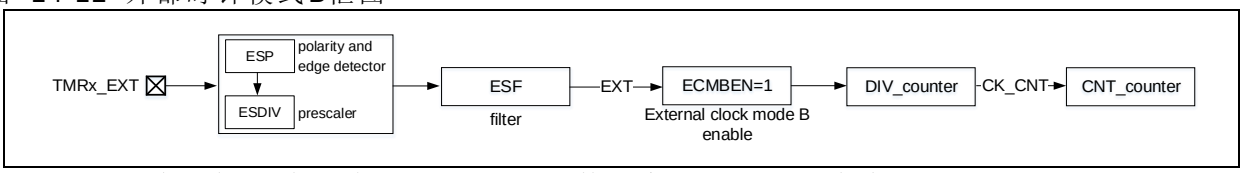
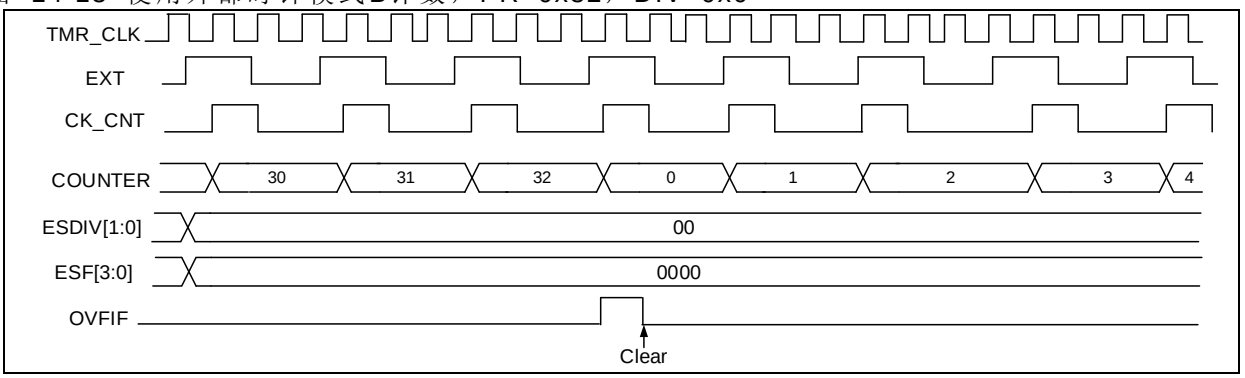


图 14-12 外部时钟模式B框图



注：由于同步逻辑。输入端 EXT 信号与计数器实际时钟之间存在一定延时。

图 14-13 使用外部时钟模式B计数，PR=0x32，DIV=0x0



内部触发输入（ISx）

定时器之间支持互联同步，因此一个定时器的 TMR_CLK 可由另一个定时器输出信号 TRGOUT 提供。配置 STIS[2：0]选择内部触发信号驱动计数器计数。

TMR2 至 TMR4 定时器内含一个 16 位预分频器，用于产生驱动计数器计数的时钟 CK_CNT，通过配置 TMRx_DIV 寄存器值，可灵活调整 CK_CNT 与 TMR_CLK 之间的分频关系。预分频值可在任何时刻修改，但只在下一个溢出事件发生时，新值才会生效。

内部触发输入配置流程如下：

- -配置 TMRx_PR 寄存器，设置计数器计数周期。
- -配置 TMRx_DIV 寄存器，设置计数器计数频率。
- -配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
- -配置 TMRx_STCTRL 寄存器 STIS[2:0]位范围为 3’b000~3’b011，选择内部触发。
- -配置 TMRx_STCTRL 寄存器 SMSEL[2:0]=3’b111，选择外部时钟模式 A。

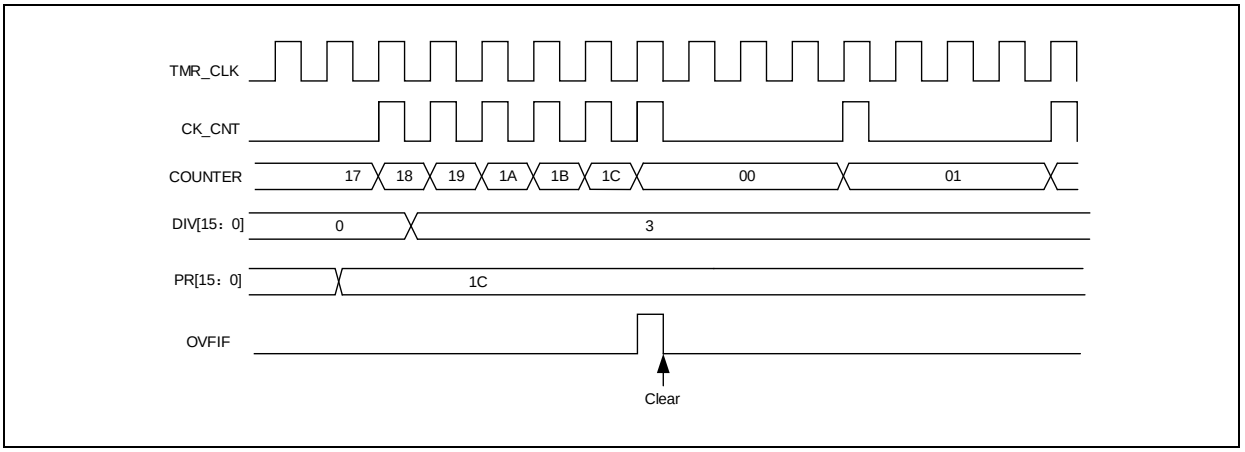
-配置 TMRx_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

表 14-3 TMRx内部触发连接

次定时器	IS0 （STIS = 000）	IS1 （STIS = 001）	IS2 （STIS = 010）	IS3 （STIS = 011）
TMR2	TMR1	TMR9	TMR3	OTGFS_SOF
TMR3	TMR1	TMR2	TMR9	TMR4
TMR4	TMR1	TMR2	TMR3	TMR9

注意 1：如果某个产品中没有相应的定时器，则对应的触发信号 ISx 也不存在。

图 14-14 当预分频器的参数从1变到4时，计数器的时序图



14.2.3.2 计数模式

TMR2 至 TMR4 定时器提供了多种计数模式，用来满足不同的应用场景。其内部拥有一个支持 16 位向上计、向下、中央双向对齐计数的计数器，TMR2 可通过将 PMEN 位置 1 扩展至 32 位。

TMRx_PR 寄存器用于设置计数器计数周期。默认 TMRx_PR 寄存器值会立即传入它的影子寄存器；当开启周期缓冲功能后（PRBEN 置 1），TMRx_PR 寄存器值在溢出事件发生时传入它的影子寄存器。

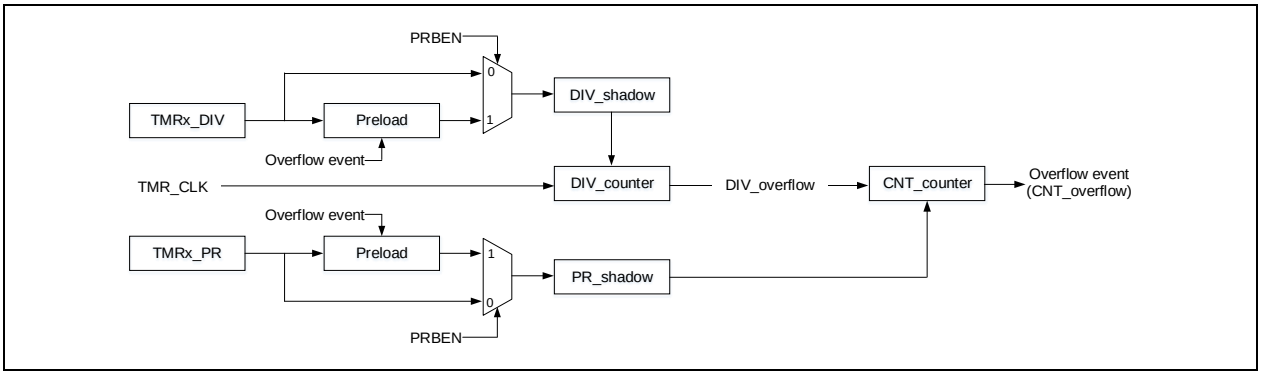
TMRx_DIV 寄存器用于设置计数器计数频率，每（DIV[15:0]+1）个计数时钟周期，计数器计数一次。和 TMRx_PR 寄存器类似，开启周期缓冲功能后，TMRx_DIV 寄存器值在溢出事件时更新至它的影子寄存器。

读取 TMRx_CNT 寄存器会返回当前计数器计数值，写入 TMRx_CNT 寄存器会更新计数器当前计数值为写入值。

默认允许产生溢出事件，设置 TMRx_CTRL1 寄存器 OVFEN=1 将禁止溢出事件产生。TMRx_CTRL1 寄存器 OVFS 用于选择溢出事件来源，默认计数器上溢或下溢、置位 OVFSWTR、复位模式次定时器控制器产生的复位信号产生溢出事件。置位 OVFS 后，只有计数器上溢或下溢产生溢出事件。

TMREN 位置 1 将使能定时器计数，由于同步逻辑，实际驱动计数器的使能信号 TMR_EN 相对于 TMREN 延迟一个时钟周期。

图 14-15 计数器基本结构



向上计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]=2’b00，OWCDIR=1’b0 开启向上计数模式，计数值达到

TMRx_PR 值时，重新从 0 向上计数，计数器上溢并产生溢出事件，同时 OVFIF 位置 1。若禁止产生溢出事件，计数器溢出后不再重载预分频值和周期值，否则预分频值和周期值在溢出事件后更新。

图 14-16 PRBEN=0时的溢出事件

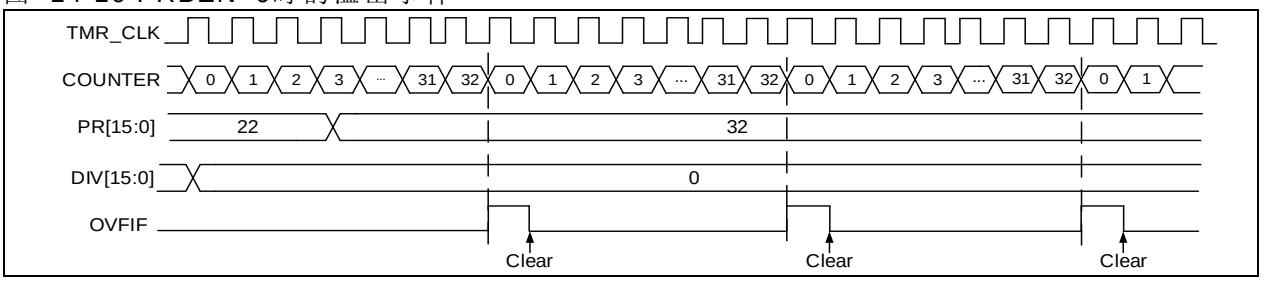
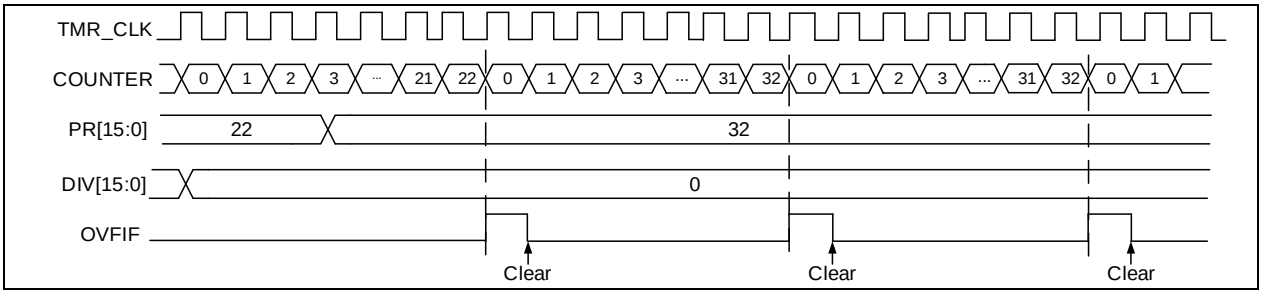


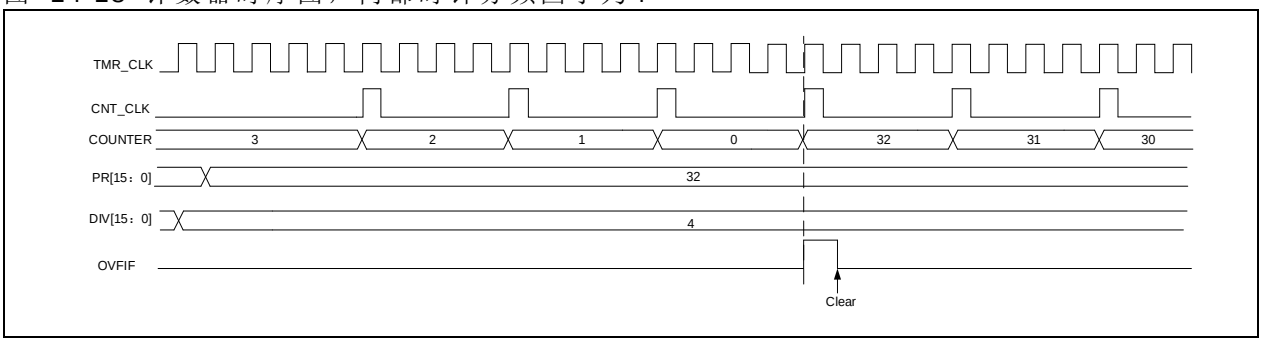
图 14-17 PRBEN=1时的溢出事件



向下计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]=2'b00，OWCDIR=1'b1 开启向下计数模式，计数值达到 0 值并重新从 TMRx_PR 向上下数时，计数器下溢并产生溢出事件。

图 14-18 计数器时序图，内部时钟分频因子为4



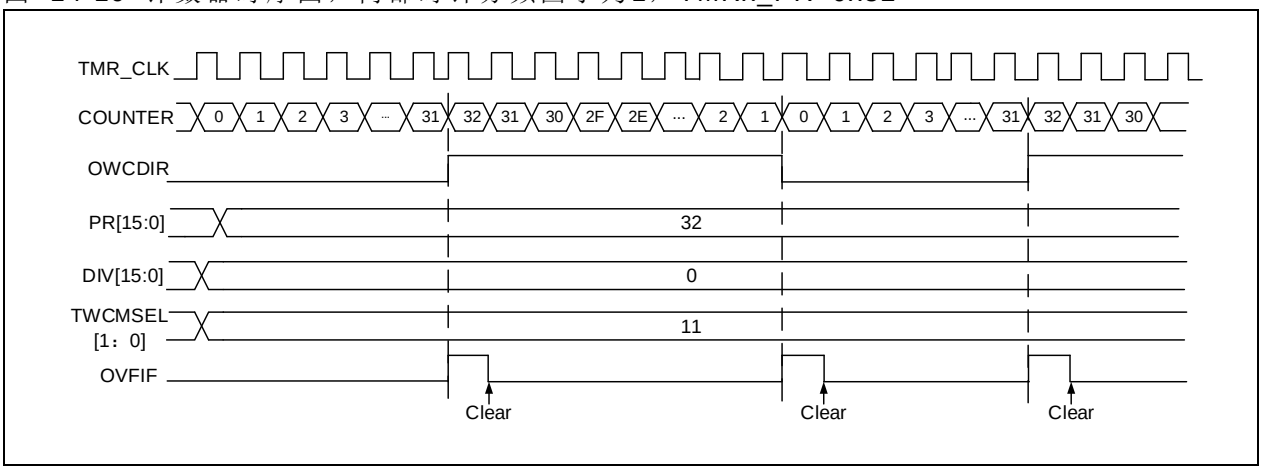
中央双向对齐计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]≠2'b00 开启中央双向对齐计数模式，中央双向对齐计数模式下计数器交替向上、向下计数。计数值从 TMRx_PR 值向下计数到 1 值，产生下溢事件，然后从 0 开始向上计数；向上计数到 TMRx_PR 值-1，产生上溢事件，之后从 TMRx_PR 值向下计数。计数器计数方向由计数器方向控制位（OWCDIR）实时查看。

TMRx_CTRL1 寄存器 TWCMSEL[1:0]位还用于选择中央双向对齐计数模式下 CxIF 标志置起方式，中央双向对齐计数模式 1（TWCMSEL[1:0]=2'b01）仅允许 CxIF 标志位在计数器向下计数时置起；双向对齐计数模式 2（TWCMSEL[1:0]=2'b10）仅允许 CxIF 标志位在计数器向上计数时置起；双向对齐计数模式 3（TWCMSEL[1:0]=2'b11）允许 CxIF 标志位在计数器向上和向下计数时置起。

注意：　中央双向对齐计数模式下，OWCDIR 位为只读位。

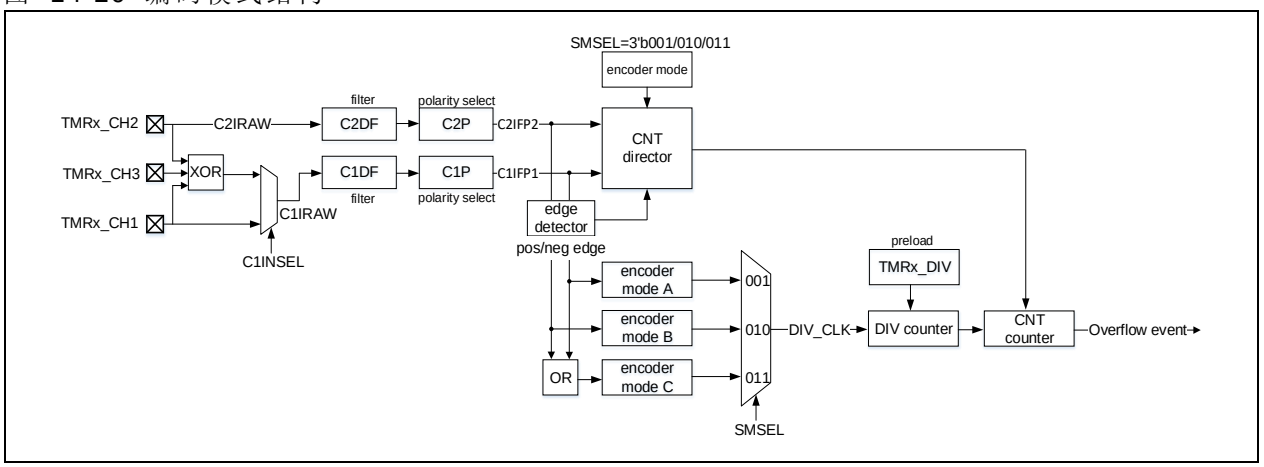
图 14-19 计数器时序图，内部时钟分频因子为1，TMRx_PR=0x32



编码器模式

编码器模式下需提供两组输入信号 TMRx_CH1 和 TMRx_CH2，根据一组输入信号电平值，计数器在另一组输入信号边沿向上或向下计数。计数方向由 OWCDIR 值指示。

图 14-20 编码模式结构



编码器模式 A：SMSEL=3’b001，计数器在 C1IFP1 边沿计数（上升沿和下降沿），计数方向由 C1IFP1 边沿方向和 C2IFP2 电平高低共同决定。

编码器模式 B：SMSEL=3’b010，计数器在 C2IFP2 边沿计数（上升沿和下降沿），计数方向由 C2IFP2 边沿方向和 C1IFP1 电平高低共同决定。

编码器模式 C：SMSEL=3’b011，计数器在 C1IFP1 和 C2IFP2 边沿计数（上升沿和下降沿），计数方向由 C1IFP1 边沿方向和 C2IFP2 电平高低、C2IFP2 边沿方向和 C1IFP1 电平高低共同决定共同决定。

若要使用编码器模式可按下面步骤配置：

-配置 TMRx_CM1 寄存器 C1DF[3:0]，设置通道 1 输入信号滤波；配置 TMRx_CCTRL 寄存器 C1P，设置通道 1 输入信号有效电平。

-配置 TMRx_CM1 寄存器 C2DF[3:0]，设置通道 2 输入信号滤波；配置 TMRx_CCTRL 寄存器 C2P，设置通道 2 输入信号有效电平。

-配置 TMRx_CM1 寄存器 C1C[1:0]，设置通道 1 为输入模式；配置 TMRx_CM1 寄存器 C2C[1:0]，设置通道 2 为输入模式；

-配置 TMRx_STCTRL 寄存器 SMSEL[2:0]，选择编码器模式 A（SMSEL=3’b001）、编码器模式 B（SMSEL=3’b010）或编码器模式 C（SMSEL=3’b011）。

-配置 TMRx_PR 寄存器 PR[15:0]，设置计数器计数周期。

-配置 TMRx_DIV 寄存器 DIV[15:0]，设置计数器计数频率。

-配置 TMRx_CH1 和 TMRx_CH2 对应 IO 为复用模式。

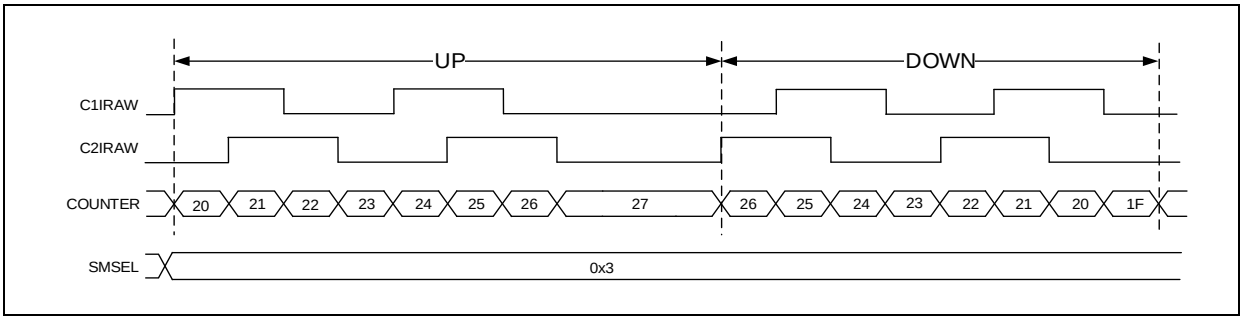
-配置 TMRx_CTRL1 寄存器 TMREN，使能计数器。

编码模式下计数器计数方向如下表所示：

表 14-4 计数方向与编码器信号的关系

计数边沿	计数边沿相对信号的电平（C1IFP1 边沿对应 C2IFP2 电平，C2IFP2 边沿对应 C1IFP1 电平）	C1IFP1 边沿方向		C2IFP2 边沿方向	
		上升	下降	上升	下降
C1IFP1	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
C2IFP2	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
C1IFP1 和 C2IFP2	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

图 14-21 编码模式计数实例（编码器模式C）



14.2.3.3 TMR输入部分

TMRx 拥有 4 个独立通道，每个通道可配置为输入或输出，当配置位输入时，每个通道输入信号依次经过以下处理：

-TMRx_CHx 经过预处理输出 CxIRAW。配置 C1INSEL 位，选择 C1IRAW 来源是 TMRx_CH1 或是 TMRx_CH1、TMRx_CH2、TMRx_CH3 异或。C2IRAW、C3IRAW、C4IRAW 来源是 TMRx_CH2、TMRx_CH3、TMRx_CH4。

-CxIRAW 输入数字滤波器，输出滤波后信号 CxIF。数字滤波器通过 CxDF 位配置采样频率和次数。

-CxIF 输入边沿检测器，输出边沿选择后信号 CxIFPx。边沿选择由 CxP 和 CxCP 位共同控制，可选择输入上升沿、下降沿或双边沿有效。

-CxIFPx 输入捕获信号选择器，输出选择后信号 CxIN。捕获信号选择器由 CxC 控制，可选择 CxIN 来源为 CxIFPx、CyIFPx、STCI。其中 CyIFPx（x≠y）是来自通道 y 的 CyIFPy 经通道 x 边沿检测器处理后的信号（例如 C1IFP2 是来自通道 1 的 C1IFP1 信号经过通道 2 边沿检测器处理后的信号）；STCI 来自次定时器控制器，由 STIS 位选择来源。

-CxIN 经由输入通道分频器，输出分频后信号 CxIPS。分频系数由 CxIDIV 位配置为不分频、2 分频、4 分频或 8 分频。

图 14-22 输入/输出通道 1 的主电路

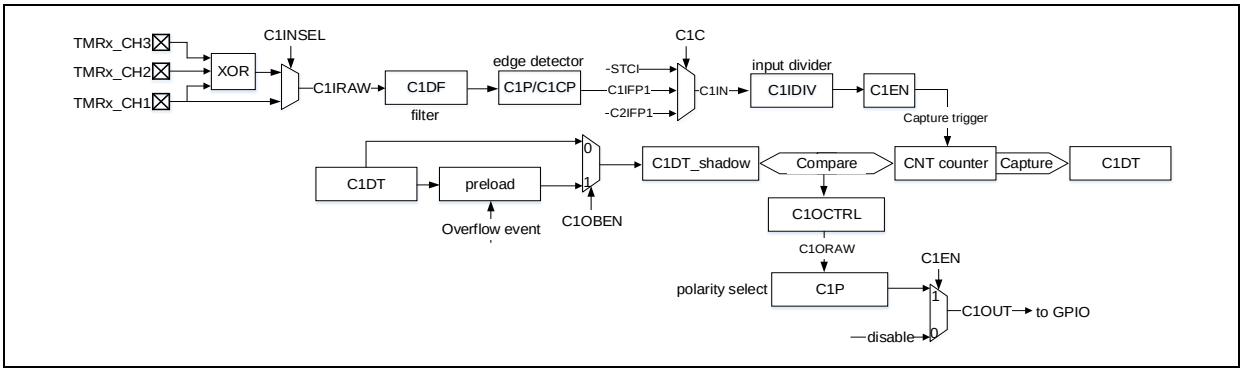
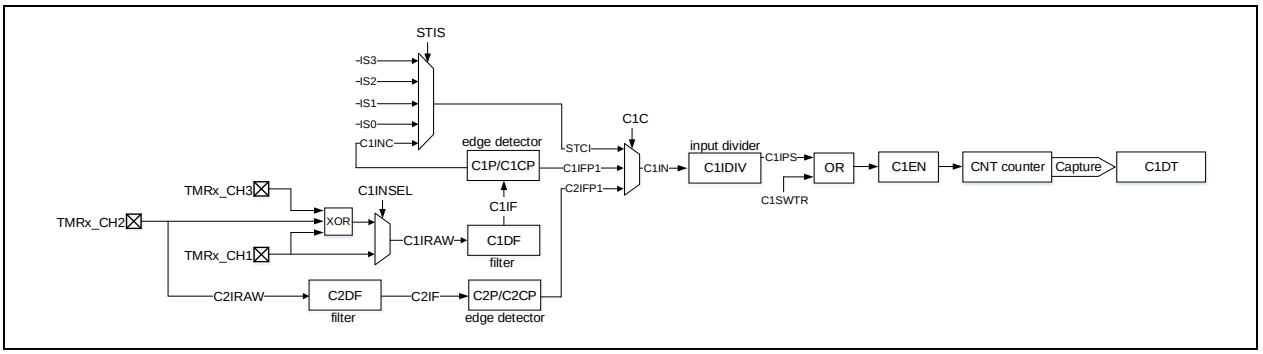


图 14-23 通道1输入部分



输入模式

此模式下，当选中的触发信号被检测到，通道寄存器（TMRx_CxDT）记录当前计数器计数值，并将捕获比较中断标志位（CxIF）置 1，若已使能通道中断（CxIEN）、通道 DMA 请求（CxDEN）则产生相应的中断和 DMA 请求。若在 CxIF 置 1 后检测到触发信号，将产生捕获溢出事件，TMRx_CxDT 会使用当前计数器计数值覆盖之前记录的计数器计数值，同时通道再捕获标志位（CxRF）置 1。

若要捕获 C1IN 输入的上升沿，可按如下进行配置：

- 通道模式寄存器 1（TMRx_CM1）中的 C1C 位配置为 01，选择 C1IN 作为通道 1 输入。
- 配置 C1IN 信号滤波器带宽（CxDF[3：0]）。
- 配置 C1IN 通道的有效沿，在 TMRx_CCTRL 寄存器中写入 C1P=0（上升沿）。
- 配置 C1IN 信号捕获分频（C1DIV[1：0]）。
- 使能通道 1 输入捕获（C1EN=1）。
- 根据需要设置 TMRx_IDEN 寄存器中的 C1IEN 位、TMRx_IDEN 寄存器中的 C1DEN 位，选择中断请求或 DMA 请求。

多输入异或

通道 1 的输入端可选择 TMRx_CH1、TMRx_CH2 和 TMRx_CH3 经异或逻辑后输入。将 TMRx_CTRL2 寄存器中的 C1INSEL 位置 1 可开启此功能。

多输入异或功能可用于连接霍尔传感器，例如，将异或输入的三个输入端分别连接到三个霍尔传感器，通过分析三路霍尔传感器信号可计算出转子的位置和速度。

PWM 输入

PWM 输入模式适用于通道 1 和 2，要使用此模式，需要将 C1IN 和 C2IN 映射到同一 TMRx_CHx，并且通道 1 或 2 的 CxIFPx 配置成触发次定时器控制器复位。

PWM 输入模式可用于测量输入信号的周期和占空比，如需测量通道 1 输入信号的周期和占空比，操作步骤如下：

- 配置C1C=2‘b01，选择C1IN为C1IFP1。
- 配置C1P=1’b0，选择C1IFP1上升沿有效。
- 配置C2C=2‘b10，选择C2IN为C1IFP2。
- 配置C2P=1’b1，选择C1IFP2下降沿有效。
- 配置STIS=3’b101，选择次定时器触发信号为C1IFP1。
- 配置SMSEL=3‘b100，选择次定时器模式为复位模式。
- 配置C1EN=1’b1，C2EN=1’b1。使能通道1和输入捕获。

上述配置下，通道 1 输入信号的上升沿会触发捕获并将捕获值存储到 C1DT 寄存器，同时通道 1 输入信号上升沿复位计数器。通道 1 输入信号下降沿触发捕获并将捕获值存储到 C2DT 寄存器。通道 1 输入信号的周期可通过 C1DT 计算，占空比可通过 C2DT 计算。

图 14-24 PWM输入模式配置实例

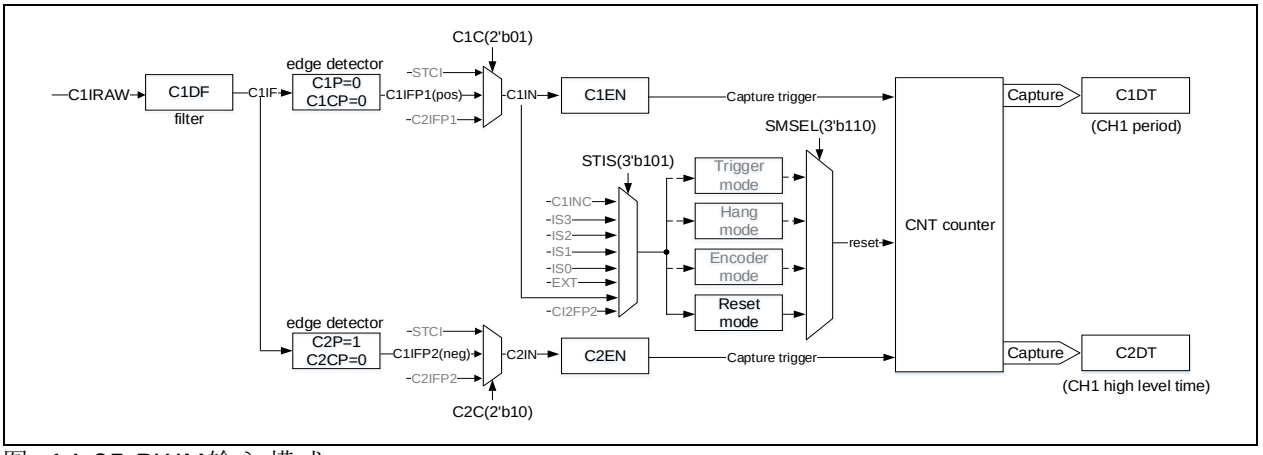
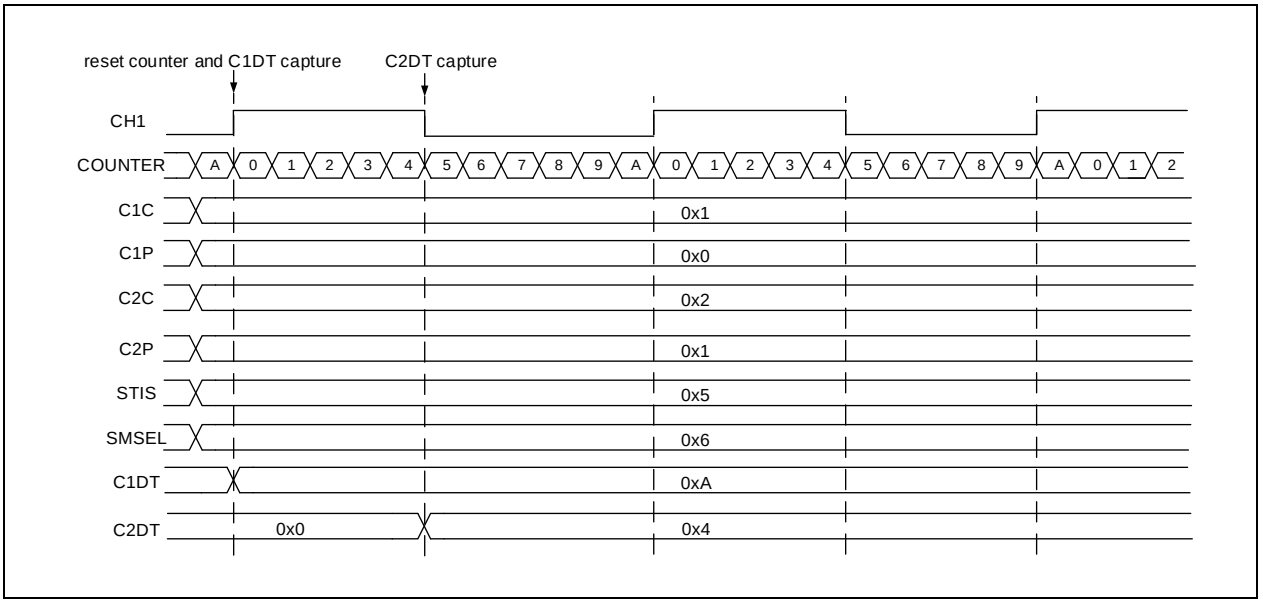


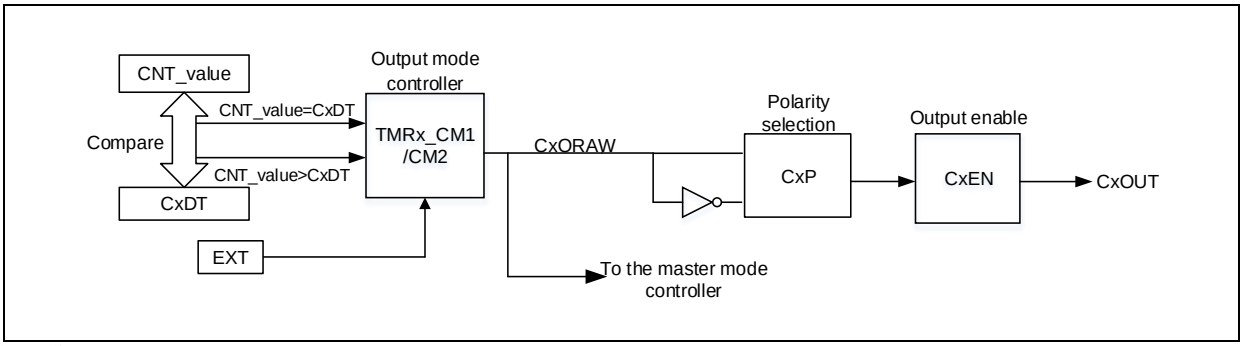
图 14-25 PWM输入模式



14.2.3.4 TMR输出部分

TMR 的输出部分由比较器和输出控制构成，用于编程输出信号的周期、占空比、极性。

图 14-26 捕获/比较通道的输出部分（通道 1 至 4）



输出模式

配置 CxC[1:0]≠2'b00 将通道配置为输出可实现多种输出模式，此时，计数器计数值将与 CxDT 寄存器值比较，并根据 CxOCTRL[2:0]位配置的输出模式，产生中间信号 CxORAW，再经过输出控制逻辑处理后输送到 IO。输出信号的周期由 TMRx_PR 寄存器值配置，占空比则由 CxDT 寄存器值配置。

输出比较模式有以下子类：

PWM 模式 A： CxOCTRL=3'b110 时，开启 PWM 模式 A。向上计数时，TMRx_C1DT>TMRx_CVAL 时 C1ORAW 输出高电平，否则为低电平；向下计数时，TMRx_C1DT<TMRx_CVAL 时 C1ORAW 输出低电平，否则为高电平。若要使用 PWM 模式 A，可按如下方式配置。

- -配置 TMRx_PR 寄存器，设置 PWM 周期。
- -配置 TMRx_CxDT 寄存器，设置 PWM 占空比。
- -配置 TMRx_CM1/CM2 寄存器 CxOCTRL 位为 3'b110，设置输出模式为 PWM 模式 A。
- -配置 TMRx_DIV 寄存器，设置计数器计数频率。
- -配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
- -配置 TMRx_CCTRL 寄存器 CxP 位、CxCP 位，设置输出极性。
- -配置 TMRx_CCTRL 寄存器 CxEN 位、CxCEN 位，使能通道输出。
- -配置 TMRx_BRK 寄存器 OEN 位，使能 TMRx 输出。
- -配置 TMR 输出通道对应 GPIO 为对应的复用模式。
- -配置 TMRx_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

- **PWM 模式 B**：CxOCTRL=3'b111 时，开启 PWM 模式 B。向上计数时，TMRx_C1DT>TMRx_CVAL 时 C1ORAW 输出低电平，否则为高电平；向下计数时，TMRx_C1DT<TMRx_CVAL 时 C1ORAW 输出高电平，否则为低电平。**强制输出模式**：CxOCTRL=3'b100/101 时，开启强制输出模式。此时，CxORAW 信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。
- **输出比较模式**：CxOCTRL=3'b001/010/011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 强制输出高电平（CxOCTRL=3'b001）、低电平（CxOCTRL=3'b010）或进行电平翻转（CxOCTRL=3'b011）。
- **单周期模式**：PWM 模式的特例，将 OCMEN 位置 1 可开启单周期模式，此模式下，仅在当前计数周期中进行比较匹配，完成当前计数后，TMREN 位清 0，因此仅输出一个脉冲。当配置为向上计数模式时，需要严格配置 CVAL<CxDT≤PR；向下计数时，需严格配置 CVAL>CxDT。
- **快速输出模式**：将 CxOIEN 位置 1 可开启此功能，开启后 CxORAW 电平值不再在计数值与 CxDT 匹配时变化，而是在当前计数周期开始时，也就是说，比较结果被提前了，计数器值与 CxDT 寄存器的比较结果将会提前决定 CxORAW 的电平。

图 14-27 展示了输出比较模式（翻转）的例子，C1DT=0x3，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-28 展示了计数器向上计数与 PWM 模式 A 配合的例子，PR=0x32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-29 展示了计数器中央双向对齐计数与 PWM 模式 A 配合的例子，PR=0x32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-30 展示了计数器向上计数与单周期模式下 PWM 模式 B 配合的例子，计数器仅计数了一个周期，输出信号在这个周期中只输出了一个脉冲。

图 14-27 计数值与C1DT值匹配时翻转C1ORAW

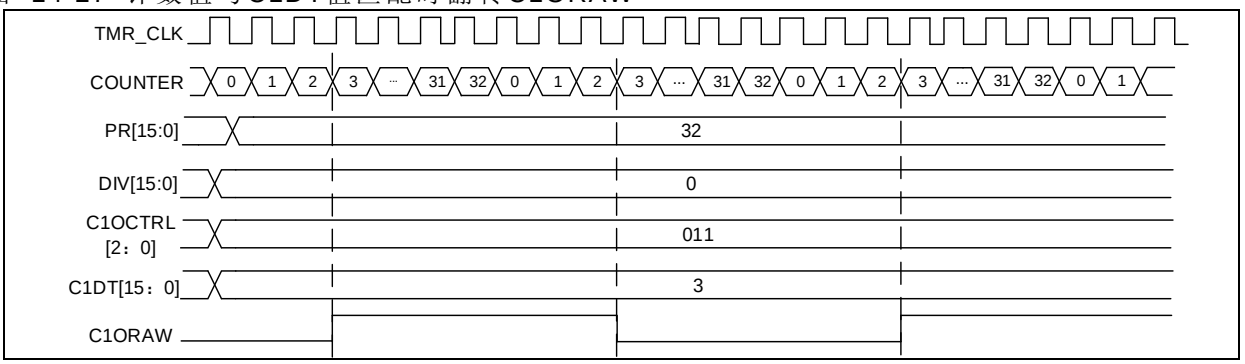


图 14-28 向上计数下PWM模式A

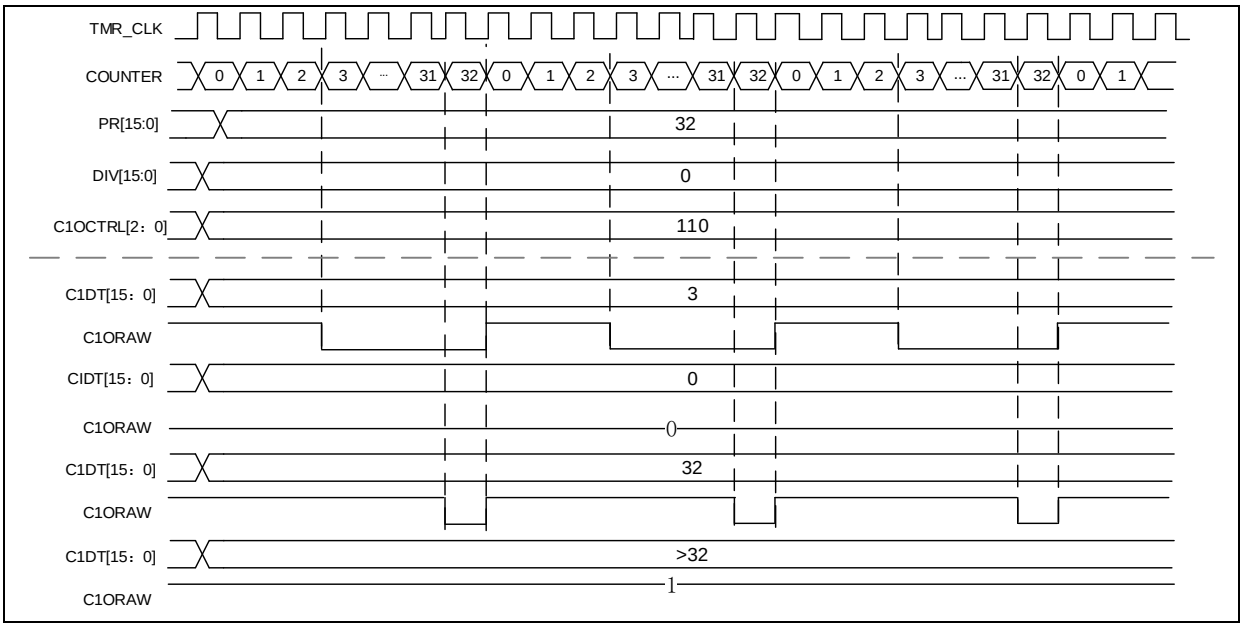


图 14-29中央双向对齐计数下PWM模式A

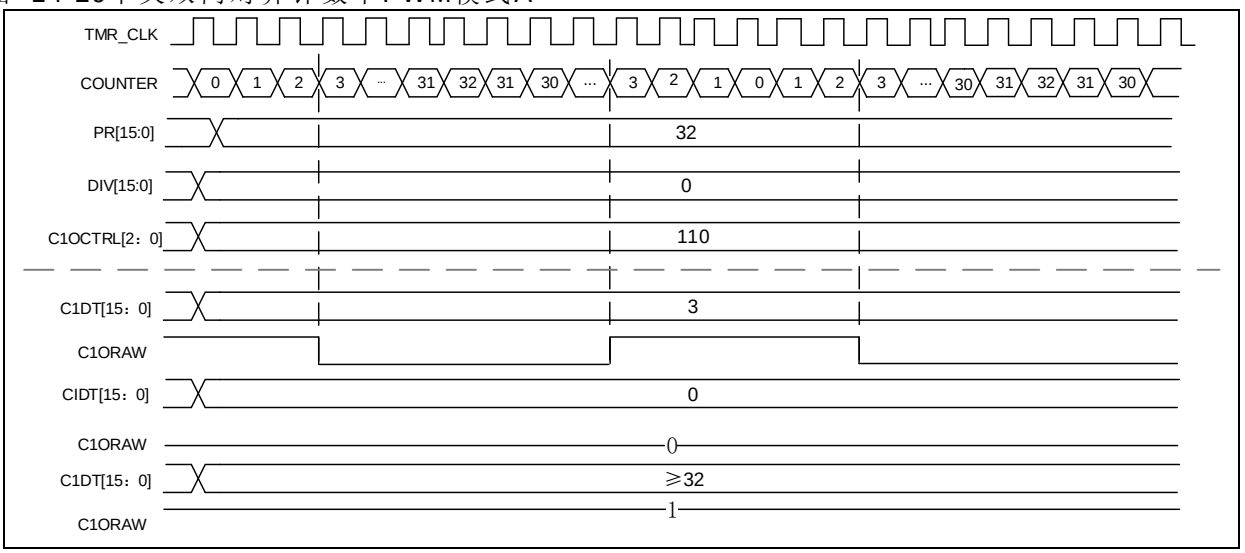
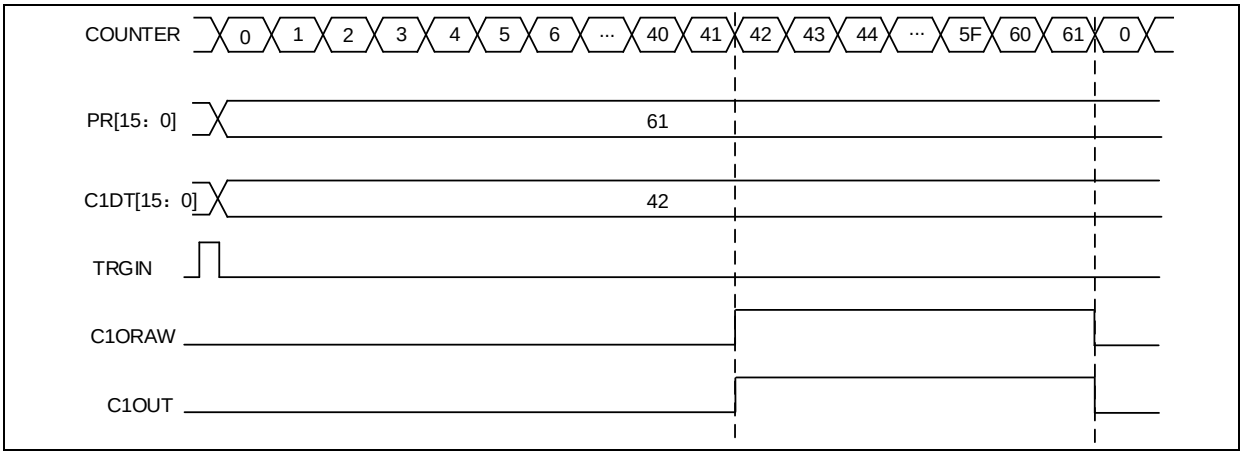


图 14-30 单周期模式



主定时器事件输出

当 TMR 作为主定时器时，可选择如下信号源作为 TRGOUT 信号输出到次定时器，选择信号为 TMRxCTRL2 寄存器 PTOS 位。

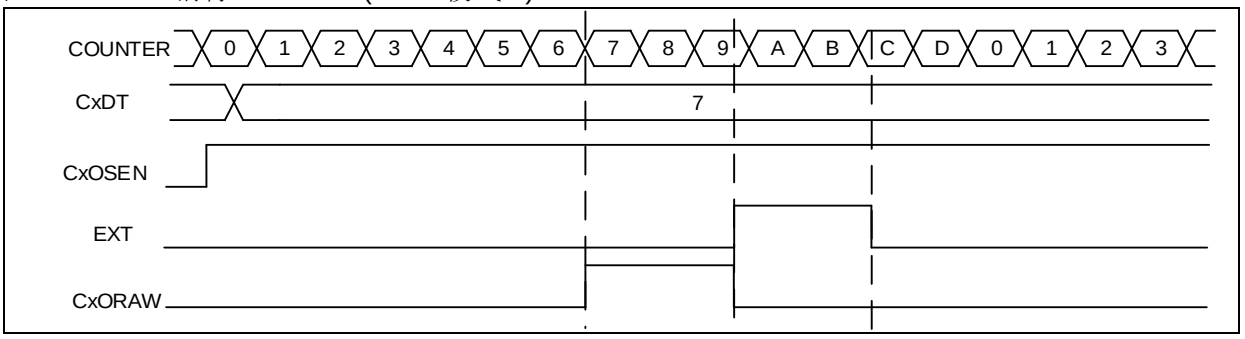
●-PTOS=3’b000，TRGOUT 输出软件溢出事件（TMRx_SWEVT 寄存器 OVFSWTR 位）。
●-PTOS=3’b001，TRGOUT 输出计数器使能信号。
●-PTOS=3’b010，TRGOUT 输出计数器溢出事件。
●-PTOS=3’b011，TRGOUT 输出捕获、比较事件。
●-PTOS=3’b100，TRGOUT 输出 C1ORAW 信号。
●-PTOS=3’b101，TRGOUT 输出 C2ORAW 信号。
●-PTOS=3’b110，TRGOUT 输出 C3ORAW 信号。
-PTOS=3’b111，TRGOUT 输出 C4ORAW 信号。

CxORAW 信号清除

将 CxOSEN 位置 1 后，指定通道的 CxORAW 信号由 EXT 高电平清 0，在下一次溢出事件发生前 CxORAW 信号无法被改变。

强制输出模式时，CxORAW 信号清除功能不可用，只有在输出比较模式或 PWM 模式，此功能有效。下图显示了使用 EXT 信号清除 CxORAW 的例子，当 EXT 为高电平期间，原本为高电平的 CxORAW 信号被拉低，当 EXT 为低电平时，CxORAW 根据计数值和 CxDT 比较结果输出电平。

图 14-31 EXT清除CxORAW(PWM模式A)



14.2.3.5 定时器同步

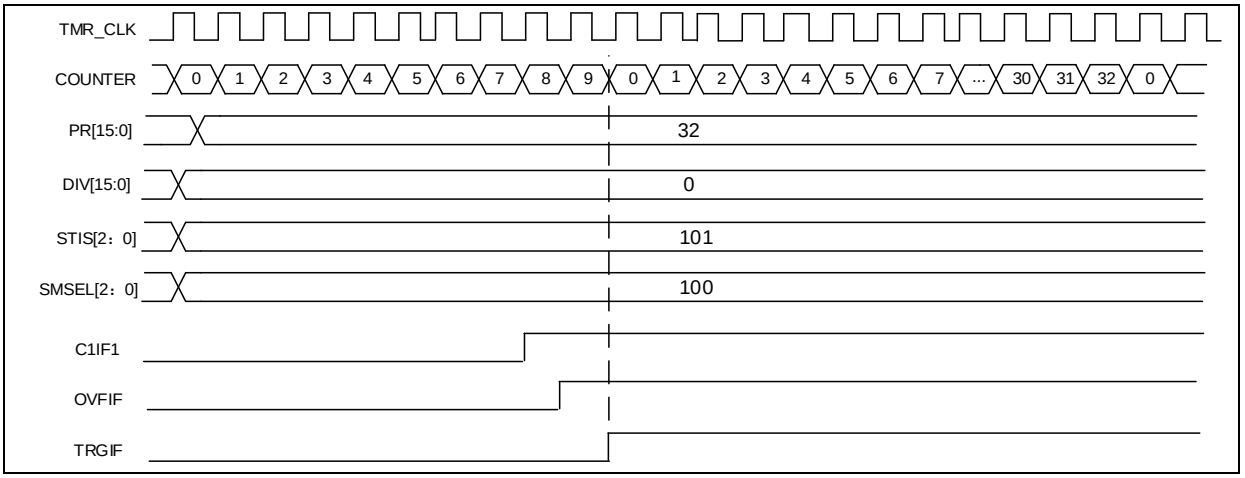
主次定时器之间可由内部连接信号进行同步。主定时器可由 PTOS[2：0]位选择主定时器输出，即同步信息；次定时器由 SMSEL[2：0]位选择从模式，即次定时器的工作模式。

定时器从模式有以下几种：

从模式：复位模式

选中的触发信号将复位计数器和预分频器，若 OVFS 位为 0，将产生一个溢出事件。

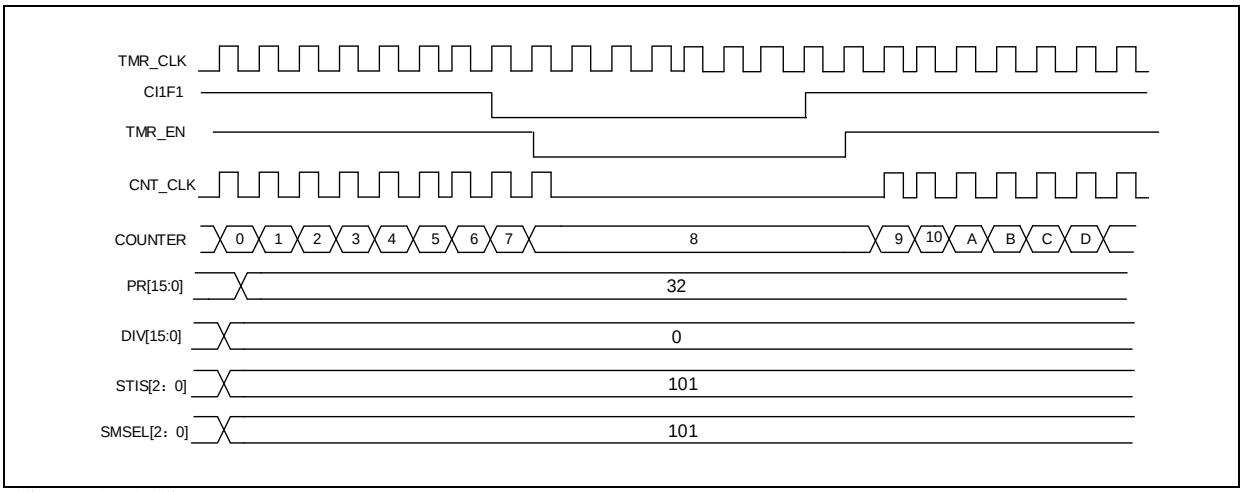
图 14-32 复位模式例子



从模式：挂起模式

挂起模式下，计数的计数和停止受选中触发输入信号控制，当触发输入为高电平时计数器开始计数；当为低电平时，计数器暂停计数。

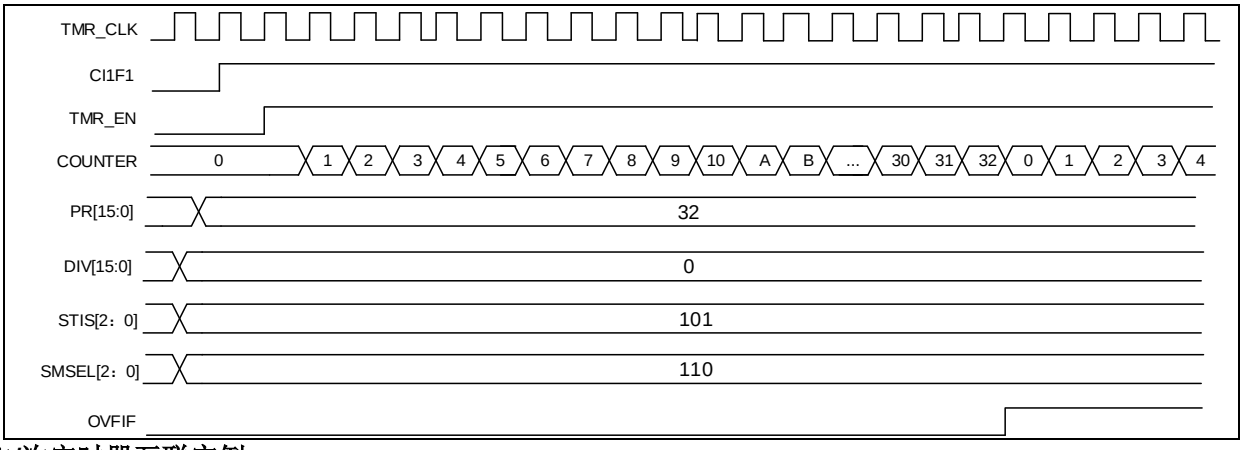
图 14-33 挂起模式下例子



从模式：触发模式

计数器将在选中的触发输入上升沿启动计数（将 TMR_EN 置 1）。

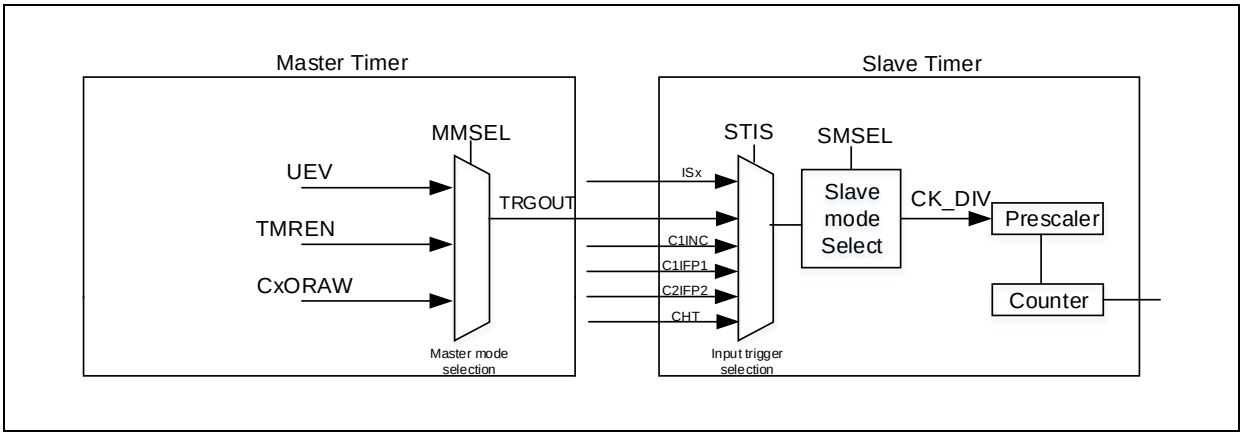
图 14-34 触发器模式例子



主/次定时器互联实例

主/次定时器可分别配置不同的主模式和从模式，两者搭配可实现多种功能，一下提供了一些定时器互联的例子。

图 14-35 主/次定时器连接框图



主定时器为次定时器提供时钟：

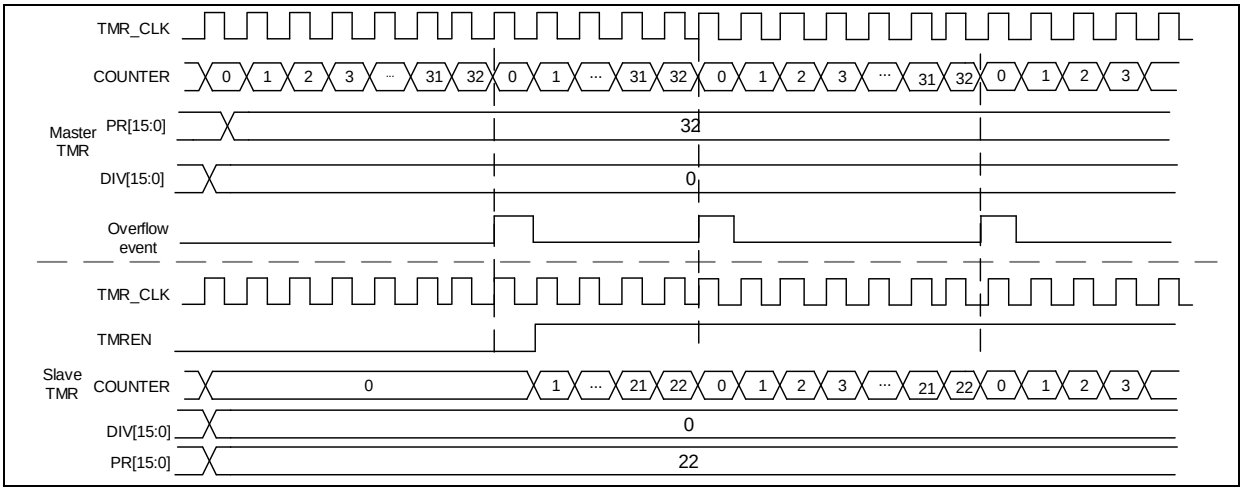
- 配置主定时器输出信号 TRGOUT 为溢出事件，配置 PTOS[2：0]=3’b010，主定时器每次计数器溢出输出一个脉冲信号，用作次定时器计数时钟。
- 配置主定时器计数周期（TMRx_PR 寄存器）。
- 配置次定时器触发输入信号 TRGIN 为主定时器输出（TMRx_STCTRL 寄存器的 STIS[2：0]）。
- 配置次定时器使用外部时钟模式 A（TMRx_STCTRL 寄存器的 SMSEL[2：0]=3’b111）。

- 将主定时器和次定时器的 TMREN 位置 1 启动定时器。

主定时器启动次定时器：

- 配置主定时器输出信号 TRGOUT 为溢出事件，配置 PTOS[2：0]=3’b010，主定时器每次计数器溢出输出一个脉冲信号，用作次定时器计数时钟。
- 配置主定时器计数周期（TMRx_PR 寄存器）。
- 配置次定时器触发输入 TRGIN 为主定时器输出。
- 置主定时器 TMREN=1 以启动主定时器。

图 14-36 主定时器启动次定时器例子

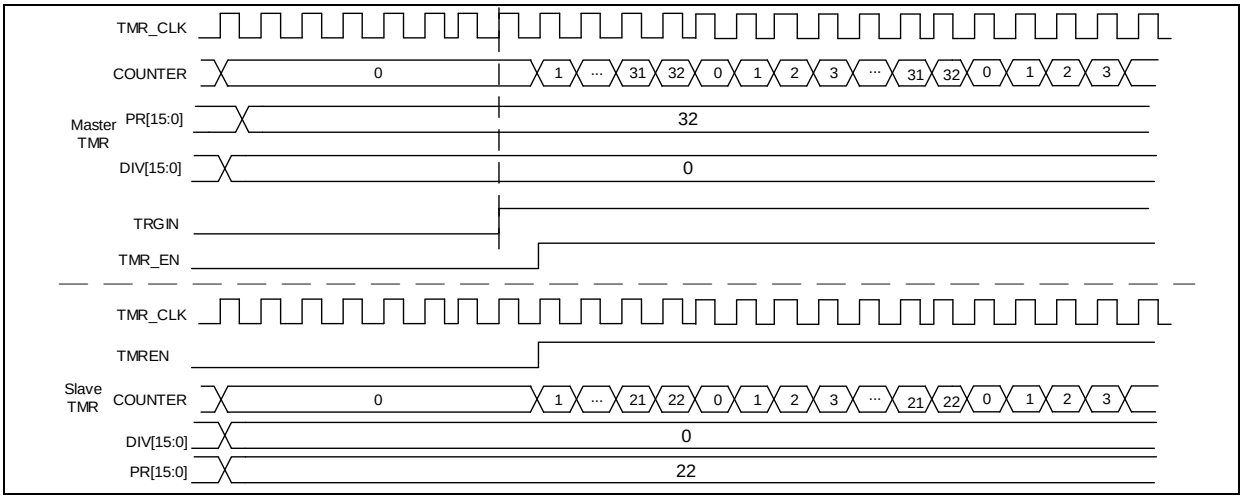


外部触发信号同步启动主、次定时器：

这个例子中，主定时器同时作为主定时器和次定时器，将主定时器的次定时器同步功能开启，此模式用于将主定时器和次定时器保持同步。

- 配置主定时器 STS 位为 1。
- 配置主定时器输出信号 TRGOUT 为溢出事件，配置 PTOS[2：0]=3’b010，主定时器每次计数器溢出输出一个脉冲信号，用作次定时器计数时钟。
- 配置主定时器的次定时模式为触发模式，触发源选择 C1IN。
- 配置次定时器触发输入 TRGIN 为主定时器输出。
- 配置次定时器为触发模式（TMR2_STCTRL 寄存器的 SMSEL=3’b110）。

图 14-37 外部触发同时启动主、次定时器



14.2.3.6 调试模式

当微控制器进入调试模式（Cortex®-M4F 核心停止）时，将 DEBUG 模块中的 TMRx_PAUSE 置 1，可以使 TMRx 计数器暂停计数。

14.2.4 TMR2至TMR4寄存器描述

必须以字（32 位）的方式操作这些外设寄存器。

下表中将 TMR2 至 TMR4 的所有寄存器映射到一个 16 位可寻址（编址）空间。

表 14-5 TMR2至TMR4寄存器图和复位值

寄存器简称	基址偏移量	复位值
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_STCTRL	0x08	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CM2	0x1C	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000 0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000 0000
TMRx_C1DT	0x34	0x0000 0000
TMRx_C2DT	0x38	0x0000 0000
TMRx_C3DT	0x3C	0x0000 0000
TMRx_C4DT	0x40	0x0000 0000
TMRx_DMACTRL	0x48	0x0000
TMRx_DMADT	0x4C	0x0000

14.2.4.1 控制寄存器1（TMRx_CTRL1）

域	简称	复位值	类型	功能
位 15：11	保留	0x00	resd	保持默认值。
位 10	PMEN	0x0	rw	增强模式使能（Plus Mode Enable） 开启 TMR2 增强模式，该模式下 TMR2_CVAL，TMR2_PR， TMR2_CxDT 由 16 位扩展为 32 位。 0：关闭； 1：开启。 注：TMR2 才具有此功能，其它 TMR 设置此位无效。在增强模式关闭状态下，TMRx_CVAL， TMRx_PR，TMRx_CxDT 寄存器只能写入 16 位值。
位 9：8	CLKDIV	0x0	rw	时钟除频（Clock divider） 此位用于设置数字滤波器采样频率 f_{DTS} 和定时器时钟频率 f_{CK_INT} 之间的分频比。 00：无除频，$f_{DTS}=f_{CK\_INT}$； 01：2 除频，$f_{DTS}=f_{CK\_INT}/2$； 10：4 除频，$f_{DTS}=f_{CK\_INT}/4$； 11：保留。
位 7	PRBEN	0x0	rw	周期缓冲使能（Period buffer enable） 0：缓冲关闭； 1：缓冲开启。
位 6：5	TWCMSEL	0x0	rw	中央双向对齐计数模式选择 （Two-way count mode selection） 00：单向对齐计数模式，方向由 OWCDIR 配置；

				01：中央双向对齐计数模式 1，上下交替计数，CxIF 位只在计数器向下计数时被置起； 10：中央双向对齐计数模式 2，上下交替计数，CxIF 位只在计数器向上计数时被置起； 11：中央双向对齐计数模式 3，上下交替计数，CxIF 位在计数器向上和向下计数时皆被置起。
位 4	OWCDIR	0x0	rw	单向计数方向 （One-way count direction） 0：向上； 1：向下。
位 3	OCMEN	0x0	rw	单周期使能 （One cycle mode enable） 该功能用于选择溢出事件后，计数器是否停止。 0：关闭； 1：开启。
位 2	OVFS	0x0	rw	溢出事件源选择 （Overflow event source） 配置溢出事件或 DMA 请求来源。 0：来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件； 1：只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 （Overflow event enable） 0：开启； 1：关闭。
位 0	TMREN	0x0	rw	使能定时器（TMR enable） 0：关闭； 1：开启。

14.2.4.2 控制寄存器2（TMRx_CTRL2）

域	简称	复位值	类型	功能
位 15：8	保留	0x00	resd	保持默认值。
位 7	C1INSEL	0x0	rw	C1IN 选择 （C1IN selection） 0：CH1 管脚连到 C1IRAW 输入； 1：CH1、CH2 和 CH3 管脚异或结果连到 C1IRAW 输入。
位 6：4	PTOS	0x0	rw	主定时器输出信号选择 （Primary TMR output selection） TMRx 输出到次定时器的信号选择： 000：复位； 001：使能； 010：更新； 011：比较脉冲； 100：C1ORAW 信号； 101：C2ORAW 信号； 110：C3ORAW 信号； 111：C4ORAW 信号。
位 3	DRS	0x0	rw	DMA 请求源（DMA request source） DMA 请求来源。 0：捕获/比较事件； 1：溢出事件。
位 2：0	保留	0x0	resd	保持默认值。

14.2.4.3 次定时器控制寄存器（TMRx_STCTRL）

域	简称	复位值	类型	功能
位 15	ESP	0x0	rw	外部信号极性（External signal polarity） 用于选择外部方式。 0：高电平或上升沿； 1：低电平或下降沿。
位 14	ECMBEN	0x0	rw	外部时钟模式 B 使能 （External clock mode 2 enable） 用于启用外部时钟模式 B 0：关闭； 1：开启。
位 13：12	ESDIV	0x0	rw	外部信号除频 （External trigger signal division）

				用于选择降低外部触发频率的除频。 00：关闭分频； 01：2 分频； 10：4 分频； 11：8 分频。
位 11：8	ESF	0x0	rw	外部信号滤波（External signal filter） 用于过滤外部信号，当外部信号产生了 N 次之后才能被采样。 0000：无滤波器，以f_{DTS}采样 0001：$f_{SAMPLING}$ =f_{CK_INT}，N=2； 0010：$f_{SAMPLING}$ =f_{CK_INT}，N=4； 0011：$f_{SAMPLING}$ =f_{CK_INT}，N=8； 0100：$f_{SAMPLING}$ =f_{DTS}/2，N=6； 0101：$f_{SAMPLING}$ =f_{DTS}/2，N=8； 0110：$f_{SAMPLING}$ =f_{DTS}/4，N=6； 0111：$f_{SAMPLING}$=f_{DTS}/4， N=8； 1000：$f_{SAMPLING}$=f_{DTS}/8， N=6； 1001：$f_{SAMPLING}$=f_{DTS}/8， N=8； 1010：$f_{SAMPLING}$=f_{DTS}/16，N=5； 1011：$f_{SAMPLING}$=f_{DTS}/16，N=6； 1100：$f_{SAMPLING}$=f_{DTS}/16，N=8； 1101：$f_{SAMPLING}$=f_{DTS}/32，N=5； 1110：$f_{SAMPLING}$=f_{DTS}/32，N=6； 1111：$f_{SAMPLING}$=f_{DTS}/32，N=8。
位 7	STS	0x0	rw	次定时器同步 （Subordinate TMR synchronization） 该位开启后，主次定时器可实现高度同步。 0：关闭； 1：开启。
位 6：4	STIS	0x0	rw	次定时器输入选择 （Subordinate TMR input selection） 用于次定时器的输入选择。 000：内部选择 0（IS0）； 001：内部选择 1（IS1）； 010：内部选择 2（IS2）； 011：内部选择 3（IS3）； 100：C1IRAW 的输入检测器（C1INC）； 101：滤波输入 1（C1IF1）； 110：滤波输入 2（C2IF2）； 111：外部输入（EXT）。 关于每个定时器中 ISx 的细节，参见表 14-3。
位 3	保留	0x0	resd	保持默认值。
位 2：0	SMSEL	0x0	rw	次定时器模式选择 （Subordinate TMR mode selection） 000：关闭从模式； 001：编码模式 A； 010：编码模式 B； 011：编码模式 C； 100：复位模式 - TRGIN 输入上升沿时，重新初始化计数器； 101：挂起模式 - TRGIN 输入高电平时，计数器计数； 110：触发模式 - TRGIN 输入上升沿时，产生触发事件； 111：外部时钟模式 A - TRGIN 输入上升沿提供时钟； 注：编码模式 A/B/C 配置方法请查看计数模式模式章节。

14.2.4.4 DMA/中断使能寄存器（TMRx_IDEN）

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值。
位 14	TDEN	0x0	rw	触发 DMA 请求使能（Trigger DMA request enable） 0：关闭； 1：开启。
位 13	保留	0x0	resd	保持默认值。

位 12	C4DEN	0x0	rw	通道 4 的 DMA 请求使能（Channel 4 DMA request enable） 0：关闭； 1：开启。
位 11	C3DEN	0x0	rw	通道 3 的 DMA 请求使能（Channel 3 DMA request enable） 0：关闭； 1：开启。
位 10	C2DEN	0x0	rw	通道 2 的 DMA 请求使能（Channel 2 DMA request enable） 0：关闭； 1：开启。
位 9	C1DEN	0x0	rw	通道 1 的 DMA 请求使能（Channel 1 DMA request enable） 0：关闭； 1：开启。
位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能（overflow event DMA request enable） 0：关闭； 1：开启。
位 7	保留	0x0	resd	保持默认值。
位 6	TIEN	0x0	rw	触发中断使能（Trigger interrupt enable） 0：关闭； 1：开启。
位 5	保留	0x0	resd	保持默认值。
位 4	C4IEN	0x0	rw	通道 4 中断使能（Channel 4 interrupt enable） 0：关闭； 1：开启。
位 3	C3IEN	0x0	rw	通道 3 中断使能（Channel 3 interrupt enable） 0：关闭； 1：开启。
位 2	C2IEN	0x0	rw	通道 2 中断使能（Channel 2 interrupt enable） 0：关闭； 1：开启。
位 1	C1IEN	0x0	rw	通道 1 中断使能（Channel 1 interrupt enable） 0：关闭； 1：开启。
位 0	OVFIEN	0x0	rw	溢出中断使能（Overflow interrupt enable） 0：关闭； 1：开启。

14.2.4.5 中断状态寄存器（TMRx_ISTS）

域	简称	复位值	类型	功能
位 15：13	保留	0x0	resd	保持默认值。
位 12	C4RF	0x0	rw0c	通道 4 再捕获标记 （Channel 4 recapture flag） 见 C1RF 的描述。
位 11	C3RF	0x0	rw0c	通道 3 再捕获标记 （Channel 3 recapture flag） 见 C1RF 的描述。
位 10	C2RF	0x0	rw0c	通道 2 再捕获标记 （Channel 2 recapture flag） 见 C1RF 的描述。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记 （Channel 1 recapture flag） C1IF 的状态已经为’1’时是否再次发生了捕获，由硬件置’1’，写’0’清除。 0：无捕获发生； 1：捕获发生。
位 8：7	保留	0x0	resd	保持默认值。
位 6	TRGIF	0x0	rw0c	触发中断标记 （Trigger interrupt flag） 当发生触发事件时由硬件置’1’，写’0’清除。 0：无触发事件发生； 1：发生触发事件。 触发事件：在 TRGIN 接收到有效边沿，或挂起模式下接收到任意边沿。
位 5	保留	0x0	resd	保持默认值。
位 4	C4IF	0x0	rw0c	通道 4 中断标记 （Channel 4 interrupt flag）

				参考 C1IF 描述。
位 3	C3IF	0x0	rw0c	通道 3 中断标记 （Channel 3 interrupt flag） 参考 C1IF 描述。
位 2	C2IF	0x0	rw0c	通道 2 中断标记 （Channel 2 interrupt flag） 参考 C1IF 描述。
位 1	C1IF	0x0	rw0c	通道 1 中断标记 （Channel 1 interrupt flag） 若通道 1 为输入模式时： 捕获事件发生时由硬件置’1’，由软件清’0’或读 TMRx_C1DT 清’0’。 0：无捕获事件发生； 1：发生捕获事件。 若通道 1 为输出模式时： 比较事件发生时由硬件置’1’，由软件清’0’。 0：无比较事件发生； 1：发生比较事件。
位 0	OVFIF	0x0	rw0c	溢出中断标记 （Overflow interrupt flag） 当溢出事件发生时由硬件置’1’，由软件清’0’。 0：无溢出事件发生； 1：发生溢出事件，若 TMRx_CTRL1 的 OVFEN=0、OVFS=0 时： – 当 TMRx_SWEVE 寄存器的 OVFG=1 时产生溢出事件； – 当计数值 CVAL 被触发事件重初始化时产生溢出事件。

14.2.4.6 软件事件寄存器（TMRx_SWEVT）

域	简称	复位值	类型	功能
位 15：7	保留	0x000	resd	保持默认值。
位 6	TRGSWTR	0x0	rw	软件触发触发事件（Trigger event triggered by software） 通过软件触发一个触发事件。 0：无作用； 1：制造一个触发事件。
位 5	保留	0x0	resd	保持默认值。
位 4	C4SWTR	0x0	wo	软件触发通道 4 事件（Channel 4 event triggered by software） 见 C1M 的描述。
位 3	C3SWTR	0x0	wo	软件触发通道 3 事件（Channel 3 event triggered by software） 见 C1M 的描述。
位 2	C2SWTR	0x0	wo	软件触发通道 2 事件（Channel 2 event triggered by software） 见 C1M 的描述。
位 1	C1SWTR	0x0	wo	软件触发通道 1 事件（Channel 1 event triggered by software） 通过软件触发一个通道 1 事件。 0：无作用； 1：制造一个通道 1 事件。
位 0	OVFSWTR	0x0	wo	软件触发溢出事件（Overflow event triggered by software） 通过软件触发一个溢出事件。 0：无作用； 1：制造一个溢出事件。

14.2.4.7 通道模式寄存器1（TMRx_CM1）

输出比较模式：

域	简称	复位值	类型	功能
位 15	C2OSEN	0x0	rw	通道 2 输出开关使能（Channel 2 output switch enable）
位 14：12	C2OCTRL	0x0	rw	通道 2 输出控制（Channel 2 output control）
位 11	C2OBEN	0x0	rw	通道 2 输出缓存使能（Channel 2 output buffer enable）

位 10	C2OIEN	0x0	rw	通道 2 输出立即使能（Channel 2 output immediately enable）
位 9：8	C2C	0x0	rw	通道 2 配置（Channel 2 configure） 当 C2EN=’0’时，这些位用于选择通道 2 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C2IN 映射在 C2IFP2 上； 10：输入，C2IN 映射在 C1IFP2 上； 11：输入，C2IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。
位 7	C1OSEN	0x0	rw	通道 1 输出开关使能（Channel 1 output switch enable） 0：EXT 输入不影响 C1ORAW； 1：当 EXT 输入高电平时，将 C1ORAW 清 0。
位 6：4	C1OCTRL	0x0	rw	通道 1 输出控制（Channel 1 output control） 这些位用于设置原始信号 C1ORAW 的工作状态。 000：断开。断开 C1ORAW 到 C1OUT 的输出； 001：设置 C1ORAW 为高：TMRx_CVAL=TMRx_C1DT 时。 010：设置 C1ORAW 为低：TMRx_CVAL=TMRx_C1DT 时。 011：切换 C1ORAW 的电平：当 TMRx_CVAL=TMRx_C1DT 时。 100：固定 C1ORAW 为低。 101：固定 C1ORAW 为高。 110：PWM 模式 A －OWCDIR=0，若 TMRx_C1DT>TMRx_CVAL 时设置 C1ORAW 为高，否则为低； －OWCDIR=1，若 TMRx_ C1DT <TMRx_CVAL 时设置 C1ORAW 为低，否则为高。 111：PWM 模式 B －OWCDIR=0，若 TMRx_ C1DT >TMRx_CVAL 时设置 C1ORAW 为低，否则为高； －OWCDIR=1，若 TMRx_ C1DT <TMRx_CVAL 时设置 C1ORAW 为高，否则为低。 *注：除’000’外，其余配置下 C1OUT 将连接到 C1ORAW，C1OUT 的输出电平除了会根据 C1ORAW 变化外，还与 CCTRL 所配置的输出极性有关。*
位 3	C1OBEN	0x0	rw	通道 1 输出缓存使能（Channel 1 output buffer enable） 0：关闭 TMRx_C1DT 的缓存功能，写入 TMRx_C1DT 的内容会立即生效。 1：启用 TMRx_C1DT 的缓存功能，写入 TMRx_C1DT 的内容将保存到缓存寄存器中，当发生溢出事件时再更新到 TMRx_C1DT 中。
位 2	C1OIEN	0x0	rw	通道 1 输出立即使能（Channel 1 output immediately enable） 在 PWM 模式 A 或模式 B 下，该位能够缩短触发事件到通道 1 的输出响应间的时间。 0：需要比较 CVAL 与 C1DT 的值之后再产生输出。 1：无需比较 CVAL 与 C1DT 的值，当发生触发事件时立即产生输出。
位 1：0	C1C	0x0	rw	通道 1 配置 （Channel 1 configure） 当 C1EN=’0’时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C1IN 映射在 C1IFP1 上； 10：输入，C1IN 映射在 C2IFP1 上； 11：输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

输入模式：

域	简称	复位值	类型	功能

位 15：12	C2DF	0x0	rw	通道 2 滤波器 （Channel 2 digital filter）
位 11：10	C2IDIV	0x0	rw	通道 2 分频系数 （Channel 2 input divider）
位 9：8	C2C	0x0	rw	通道 2 配置（Channel 2 configure） 当 C2EN=’0’时，这些位用于选择通道 2 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C2IN 映射在 C2IFP2 上； 10：输入，C2IN 映射在 C1IFP2 上； 11：输入，C2IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。
位 7：4	C1DF	0x0	rw	通道 1 滤波器 （Channel 1 digital filter） 这些位用于配置通道 1 的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器： 0000：无滤波器，以f_{DTS}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 $f_{SAMPLING}$=f_{DTS}/2，N=6 0101：采样频率 $f_{SAMPLING}$=f_{DTS}/2，N=8 0110：采样频率 $f_{SMPLING}$=f_{DTS}/4，N=6 0111：采样频率 $f_{SAMPLING}$=f_{DTS}/4，N=8 1000：采样频率$f_{SAMPLING}$=f_{DTS}/8，N=6 1001：采样频率$f_{SAMPLING}$=f_{DTS}/8，N=8 1010：采样频率$f_{SAMPLING}$=f_{DTS}/16，N=5 1011：采样频率$f_{SAMPLING}$=f_{DTS}/16，N=6 1100：采样频率$f_{SAMPLING}$=f_{DTS}/16，N=8 1101：采样频率$f_{SAMPLING}$=f_{DTS}/32，N=5 1110：采样频率$f_{SAMPLING}$=f_{DTS}/32，N=6 1111：采样频率$f_{SAMPLING}$=f_{DTS}/32，N=8
位 3：2	C1IDIV	0x0	rw	通道 1 分频系数 （Channel 1 input divider） 这些位定义了通道 1 的分频系数。 00：不分频，每一个有效的边沿都会产生一次输入； 01：每 2 个有效的边沿产生一次输入； 10：每 4 个有效的边沿产生一次输入； 11：每 8 个有效的边沿产生一次输入。 注：C1EN=’0’时，分频系数复位。
位 1：0	C1C	0x0	rw	通道 1 配置 （Channel 1 configure） 当 C1EN=’0’时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C1IN 映射在 C1IFP1 上； 10：输入，C1IN 映射在 C2IFP1 上； 11：输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

14.2.4.8 通道模式寄存器2（TMRx_CM2）

输出比较模式：

域	简称	复位值	类型	功能
位 15	C4OSEN	0x0	rw	通道 4 输出开关使能（Channel 4 output switch enable）
位 14：12	C4OCTRL	0x0	rw	通道 4 输出控制（Channel 4 output control）
位 11	C4OBEN	0x0	rw	通道 4 输出缓存使能（Channel 4 output buffer enable）
位 10	C4OIEN	0x0	rw	通道 4 输出立即使能（Channel 4 output immediately enable）
位 9：8	C4C	0x0	rw	通道 4 配置（Channel 4 configure）

				当 C4EN=’0’时，这些位用于选择通道 4 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C4IN 映射在 C4IFP4 上； 10：输入，C4IN 映射在 C3IFP4 上； 11：输入，C4IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。
位 7	C3OSEN	0x0	rw	通道 3 输出开关使能（Channel 3 output switch enable）
位 6：4	C3OCTRL	0x0	rw	通道 3 输出控制（Channel 3 output control）
位 3	C3OBEN	0x0	rw	通道 3 输出缓存使能（Channel 3 output buffer enable）
位 2	C3OIEN	0x0	rw	通道 3 输出立即使能（Channel 3 output immediately enable）
位 1：0	C3C	0x0	rw	通道 3 配置（Channel 3 configure） 当 C3EN=’0’时，这些位用于选择通道 3 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C3IN 映射在 C3IFP3 上； 10：输入，C3IN 映射在 C4IFP3 上； 11：输入，C3IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

输入模式：

域	简称	复位值	类型	功能
位 15：12	C4DF	0x0	rw	通道 4 滤波器 （Channel 4 digital filter）
位 11：10	C4IDIV	0x0	rw	通道 4 分频系数 （Channel 4 input divider）
位 9：8	C4C	0x0	rw	通道 4 配置（Channel 4 configure） 当 C4EN=’0’时，这些位用于选择通道 4 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C4IN 映射在 C4IFP4 上； 10：输入，C4IN 映射在 C3IFP4 上； 11：输入，C4IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。
位 7：4	C3DF	0x0	rw	通道 3 滤波器 （Channel 3 digital filter）
位 3：2	C3IDIV	0x0	rw	通道 3 分频系数 （Channel 3 input divider）
位 1：0	C3C	0x0	rw	通道 3 配置（Channel 3 configure） 当 C3EN=’0’时，这些位用于选择通道 3 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C3IN 映射在 C3IFP3 上； 10：输入，C3IN 映射在 C4IFP3 上； 11：输入，C3IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

14.2.4.9 通道控制寄存器（TMRx_CCTRL）

域	简称	复位值	类型	功能
位 15：14	保留	0x0	resd	保持默认值。
位 13	C4P	0x0	rw	通道 4 极性 （Channel 4 polarity） 见 C1P 的描述。
位 12	C4EN	0x0	rw	通道 4 使能 （Channel 4 enable） 见 C1EN 的描述。
位 11	C3CP	0x0	rw	通道 3 互补极性 （Channel 3 complementary polarity） 定义输入信号的有效沿，详见 C1P 位描述。
位 10	保留	0x0	resd	保持默认值。
位 9	C3P	0x0	rw	通道 3 极性 （Channel 3 polarity） 见 C1P 的描述。
位 8	C3EN	0x0	rw	通道 3 使能 （Channel 3 enable） 见 C1EN 的描述。
位 7	C2CP	0x0	rw	通道 2 互补极性 （Channel 2 complementary polarity） 定义输入信号的有效沿，详见 C1P 位描述。

位 6	保留	0x0	resd	保持默认值。
位 5	C2P	0x0	rw	通道 2 极性 （Channel 2 polarity） 见 C1P 的描述。
位 4	C2EN	0x0	rw	通道 2 使能 （Channel 2 enable） 见 C1EN 的描述。
位 3	C1CP	0x0	rw	通道 1 互补极性 （Channel 1 complementary polarity） 定义输入信号的有效沿，详见 C1P 位描述。
位 2	保留	0x0	resd	保持默认值。
位 1	C1P	0x0	rw	通道 1 极性 （Channel 1 polarity） 通道 1 配置为输出： 0：C1OUT 的有效电平为高 1：C1OUT 的有效电平为低 通道 1 配置为输入： C1CP/C1P 位共同定义输入信号有效沿。 00：C1IN 的有效边沿为上升沿；作为外部触发使用时，C1IN 不反相。 01：C1IN 的有效边沿为下降沿；作为外部触发使用时，C1IN 反相。 10：保留 11：C1IN 的有效边沿为上升沿和下降沿；作为外部触发使用时，C1IN 不反相。
位 0	C1EN	0x0	rw	通道 1 使能 （Channel 1 enable） 0：禁止输入或输出； 1：使能输入或输出。

表 14-6 标准CxOUT通道的输出控制位

CxEN 位	CxOUT 输出状态
0	禁止输出（CxOUT=0，Cx_EN=0）
1	CxOUT = CxORAW + 极性，Cx_EN=1

注意：连接到标准 CxOUT 通道的外部 I/O 管脚状态，取决于 CxOUT 通道状态和 GPIO 以及 AFIO 寄存器。

14.2.4.10 计数值（TMRx_CVAL）

域	简称	复位值	类型	功能
位 31：16	CVAL	0x0000	rw	计数值 （Counter value） 当 TMR2 开启增强模式时（TMR2_CTRL1 中的 PMEN 位），CVAL 被扩展为 32 位。
位 15：0	CVAL	0x0000	rw	计数值 （Counter value）

14.2.4.11 分频系数（TMRx_DIV）

域	简称	复位值	类型	功能
位 15：0	DIV	0x0000	rw	分频系数 （Divider value） 计数器时钟频率 $f_{CK\_CNT} = f_{TMR\_CLK} /$（DIV[15：0]+1）。 DIV 为溢出事件发生时写入的分频系数。

14.2.4.12 周期寄存器（TMRx_PR）

域	简称	复位值	类型	功能
位 31：16	PR	0x0000	rw	周期值（Period value） 当 TMR2 开启增强模式时（TMR2_CTRL1 中的 PMEN 位），PR 被扩展为 32 位。
位 15：0	PR	0x0000	rw	周期值（Period value） 定时器计数的周期值。当周期值为 0 时，定时器不工作。

14.2.4.13 通道1数据寄存器（TMRx_C1DT）

域	简称	复位值	类型	功能
位 31：16	C1DT	0x0000	rw	通道 1 数据寄存器值 （Channel 1 data register）

				当 TMR2 开启增强模式时（TMR2_CTRL1 中的 PMEN 位），C1DT 被扩展为 32 位。
位 15：0	C1DT	0x0000	rw	通道 1 数据寄存器值（Channel 1 data register） 若通道 1 配置为输入： C1DT 是前一次通道 1 输入事件（C1IN）所保存的 CVAL。 若通道 1 配置为输出： C1DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C1OBEN），并根据设置在 C1OUT 上产生相应的输出。

14.2.4.14 通道2数据寄存器（TMRx_C2DT）

域	简称	复位值	类型	功能
位 31：16	C2DT	0x0000	rw	通道 2 数据寄存器值 （Channel 2 data register） 当 TMR2 开启增强模式时（TMR2_CTRL1 中的 PMEN 位），C2DT 被扩展为 32 位。
位 15：0	C2DT	0x0000	rw	通道 2 数据寄存器值（Channel 2 data register） 若通道 2 配置为输入： C2DT 是前一次通道 2 输入事件（C2IN）所保存的 CVAL。 若通道 2 配置为输出： C2DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C2OBEN），并根据设置在 C2OUT 上产生相应的输出。

14.2.4.15 通道3数据寄存器（TMRx_C3DT）

域	简称	复位值	类型	功能
位 31：16	C3DT	0x0000	rw	通道 3 数据寄存器值 （Channel 3 data register） 当 TMR2 开启增强模式时（TMR2_CTRL1 中的 PMEN 位），C3DT 被扩展为 32 位。
位 15：0	C3DT	0x0000	rw	通道 3 数据寄存器值（Channel 3 data register） 若通道 3 配置为输入： C3DT 是前一次通道 3 输入事件（C3IN）所保存的 CVAL。 若通道 3 配置为输出： C3DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C3OBEN），并根据设置在 C3OUT 上产生相应的输出。

14.2.4.16 通道4数据寄存器（TMRx_C4DT）

域	简称	复位值	类型	功能
位 31：16	C4DT	0x0000	rw	通道 4 数据寄存器值 （Channel 4 data register） 当 TMR2 开启增强模式时（TMR2_CTRL1 中的 PMEN 位），C4DT 被扩展为 32 位。
位 15：0	C4DT	0x0000	rw	通道 4 数据寄存器值（Channel 4 data register） 若通道 4 配置为输入： C4DT 是前一次通道 4 输入事件（C4IN）所保存的 CVAL。 若通道 4 配置为输出： C4DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C4OBEN），并根据设置在 C4OUT 上产生相应的输出。

14.2.4.17 DMA控制寄存器（TMRx_DMACTRL）

域	简称	复位值	类型	功能
位 15：13	保留	0x0	resd	保持默认值。
位 12：8	DTB	0x00	rw	DMA 传输字节（DMA transfer bytes） 这些位定义了传输的字节个数： 00000：1 个字节　　00001：2 个字节 00010：3 个字节　　00011：4 个字节

				…… 10000：17 个字节 …… 10001：18 个字节
位 7：5	保留	0x0	resd	保持默认值。
位 4：0	ADDR	0x00	rw	DMA 传输地址偏移 （DMA transfer address offset） ADDR 定义了从 TMRx_CTRL1 所在地址开始的偏移量： 00000：TMRx_CTRL1， 00001：TMRx_CTRL2， 00010：TMRx_STCTRL， ……

14.2.4.18 DMA数据寄存器（TMRx_DMADT）

域	简称	复位值	类型	功能
位 15：0	DMADT	0x0000	rw	DMA 传输的数据寄存器（DMA data register） 通过对 DMADT 寄存器的读写能够实现对任意 TMR 寄存器的操作，其操作的寄存器地址范围是： TMRx 外设地址 + ADDR*4 至 TMRx 外设地址 + ADDR*4 + DTB*4。

14.3通用定时器（TMR9和TMR12）

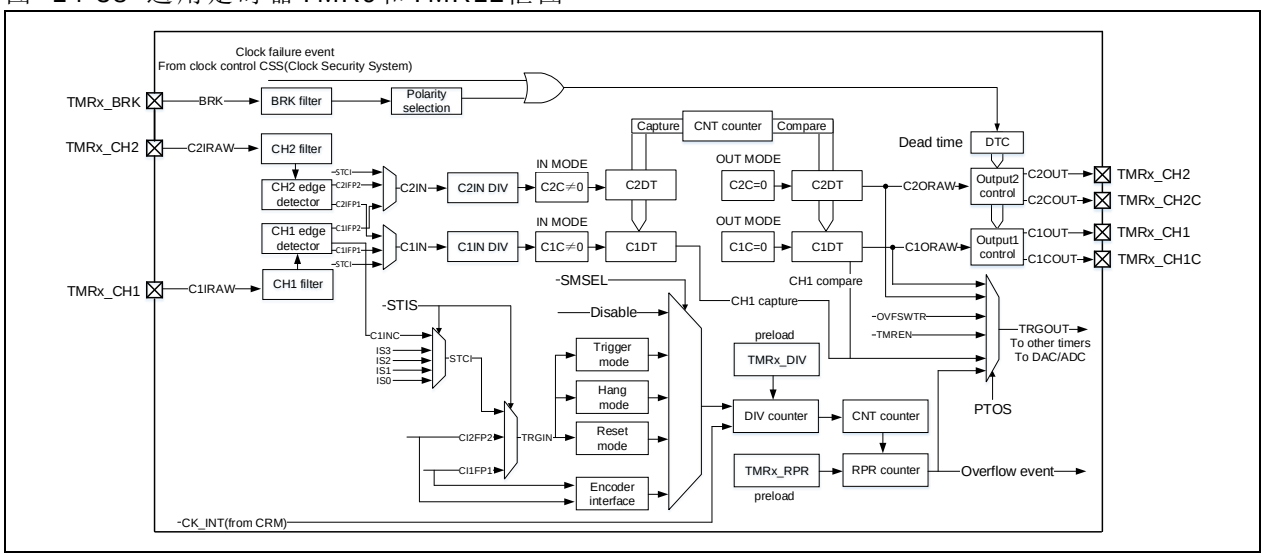
14.3.1 TMR9、TMR12简介

通用定时器 TMR9 和 TMR12 支持 16 位向上计数，2 个捕获/比较寄存器、2 组独立的通道。可实现嵌入死区、输入捕获、可编程 PWM 输出。

14.3.2 TMR9、TMR12主要特性

- 可选内部时钟、外部输入、内部触发输入用作计数时钟
- 16 位向上计数器、8 位重复计数计数器
- 2 组独立通道，支持输入捕获、输出比较、PWM 生成、单周期模式、死区插入。
- 2 组支持互补输出的独立通道
- 支持 TMR 刹车功能
- 定时器之间可互联同步
- 支持溢出事件、触发事件、刹车输入、通道事件触发中断/DMA

图 14-38 通用定时器TMR9和TMR12框图

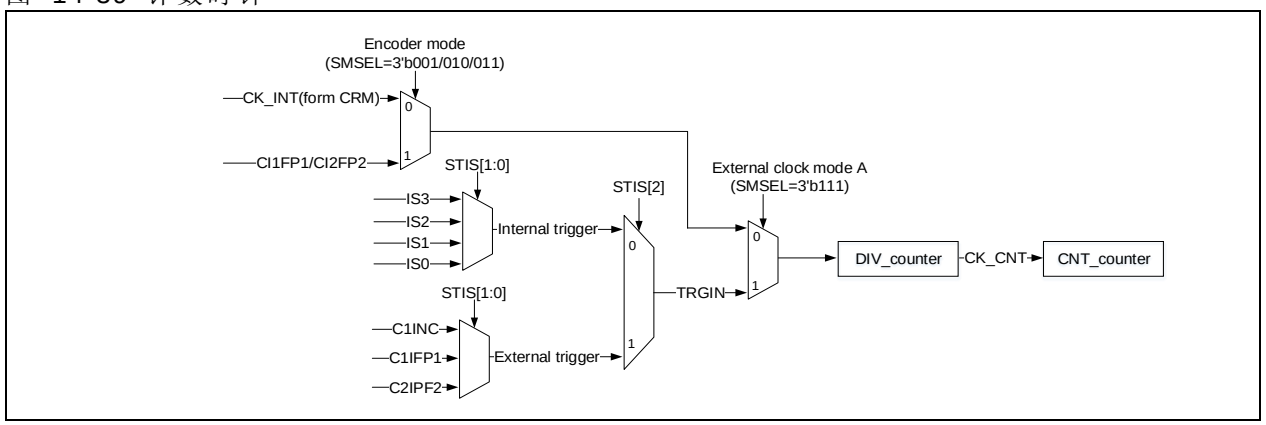


14.3.3 TMR9、TMR12功能描述

14.3.3.1 计数时钟

TMR9 和 TMR12 计数时钟可从内部时钟（CK_INT）、外部时钟（外部时钟模式 A）、内部触发输入（Isx）这些时钟源提供。

图 14-39 计数时钟

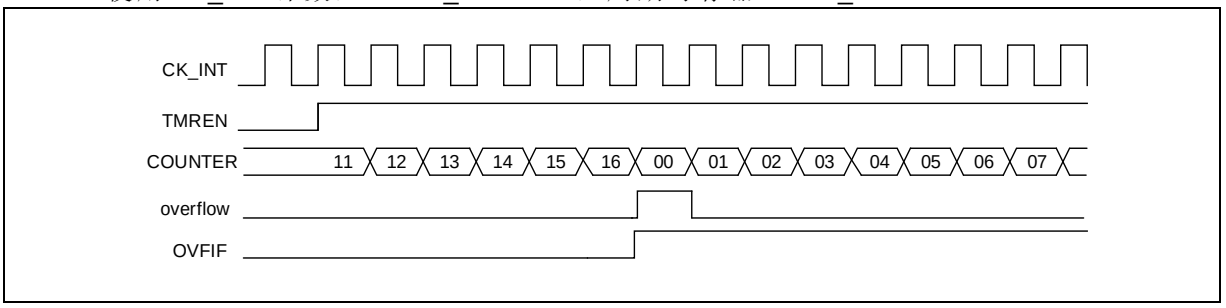


内部时钟（CK_INT）

默认下使用 CK_INT 经由预分频器驱动计数器计数，当 TMR 对应的 APB 时钟预分频系数是 1 时，CK_INT 频率等于 APB 时钟频率，否则 CK_INT 频率等于 APB 时钟频率的 2 倍。相关配置流程如下：

-配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]，选择计数模式，若选择单向对齐计数模式，还需配置 TMRx_CTRL1 寄存器 OWCDIR 选择计数方向。
-配置 TMRx_DIV 寄存器，设置计数器计数频率。
-配置 TMRx_PR 寄存器，设置计数器计数周期。
-配置TMRx_CTRL1寄存器TMREN，使能计数器。

图 14-40 使用CK_INT计数，TMRx_DIV=0x0，周期寄存器TMRx_PR=0x16



外部时钟（TRGIN/EXT）

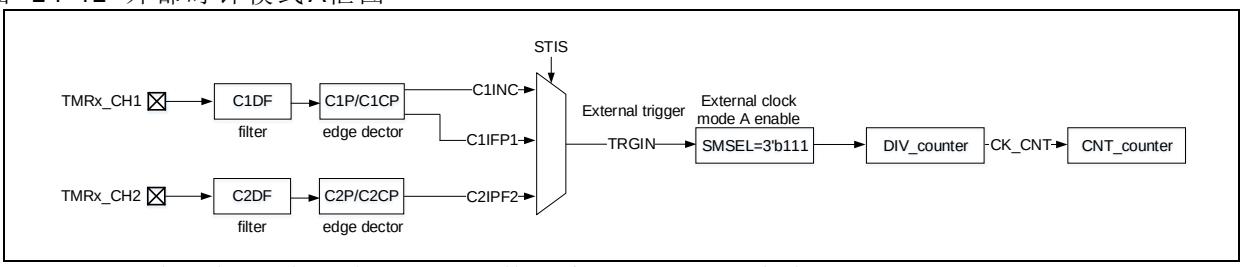
计数时钟由 TRGIN 外部时钟源提供。

当 SMSEL=3'b111 时，外部时钟模式 A 被选中，配置 STIS[2：0]来选择外部时钟源 TRGIN 信号驱动计数器计数。外部时钟源 TRGIN 可选则 C1INC（STIS=3'b100，通道 1 上升沿和下降沿信号）、C1IFP1（STIS=3'b101，通道 1 滤波且极性选择后信号）和 C2IFP2（STIS=3'b110，通道 2 滤波且极性选择后信号）。

若要使用外部时钟模式 A，可按如下步骤配置：

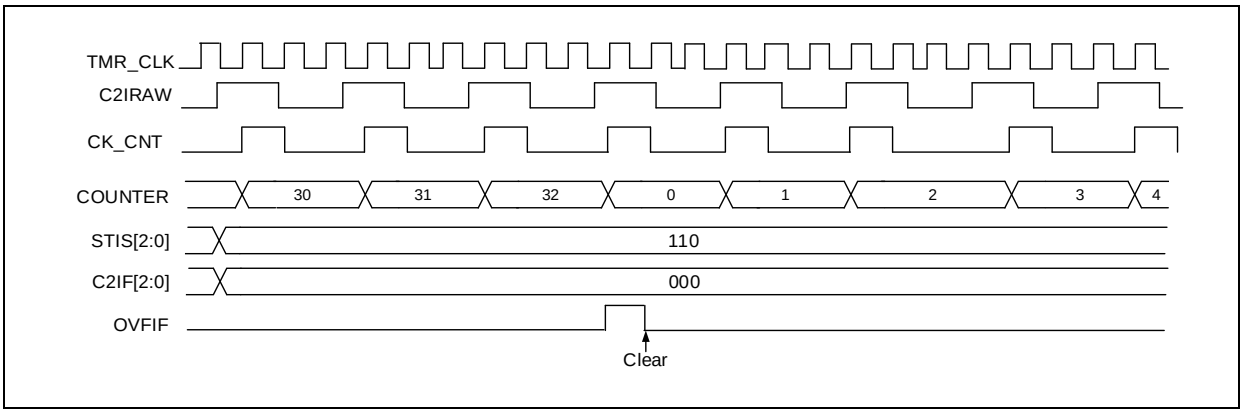
-配置外部时钟源 TRGIN 参数。
若选择 TRGIN 来源为 TMRx_CH1，需配置通道 1 输入滤波（TMRx_CM1 寄存器 C1DF[3:0]）和通道 1 输入极性（TMRx_CCTRL 寄存器 C1P/C1CP）。
若选择 TRGIN 来源为 TMRx_CH2，需配置通道 2 输入滤波（TMRx_CM1 寄存器 C2DF[3:0]）和通道 1 输入极性（TMRx_CCTRL 寄存器 C2P/C2CP）。
-配置 TMRx_STCTRL 寄存器 STIS[1:0]，设置 TRGIN 信号来源。
-配置 TMRx_STCTRL 寄存器 SMSEL=3'b111，使能外部时钟模式 A。
-配置 TMRx_DIV 寄存器 DIV[15:0]，设置计数器计数频率。
-配置 TMRx_PR 寄存器 PR[15:0]，设置计数器计数周期。
-配置 TMRx_CTRL1 寄存器 TMREN，使能计数器。

图 14-41 外部时钟模式A框图



注：由于同步逻辑，输入端信号与计数器实际时钟之间存在一定延时。

图 14-42 使用外部时钟模式A计数，PR=0x32，DIV=0x0



内部触发输入（Isx）

定时器之间支持互联同步，因此一个定时器的 TMR_CLK 可由另一个定时器输出信号 TRGOUT 提供。配置 STIS[2:0]选择内部触发信号驱动计数器计数。

TMR9 和 TMR12 定时器内含一个 16 位预分频器，用于产生驱动计数器计数的时钟 CK_CNT，通过配置 TMRx_DIV 寄存器值，可灵活调整 CK_CNT 与 TMR_CLK 之间的分频关系。预分频值可在任何时刻修改，但只在下一个溢出事件发生时，新值才会生效。

内部触发输入配置流程如下：

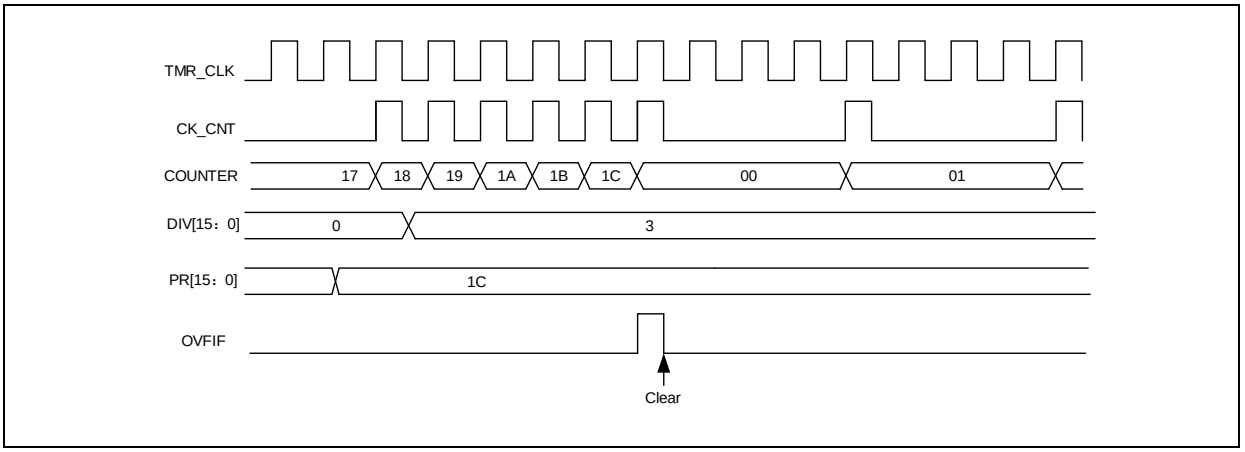
- -配置 TMRx_PR 寄存器，设置计数器计数周期。
- -配置 TMRx_DIV 寄存器，设置计数器计数频率。
- -配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
- -配置 TMRx_STCTRL 寄存器 STIS[2:0]位范围为 3'b000~3'b011，选择内部触发。
- -配置 TMRx_STCTRL 寄存器 SMSEL[2:0]=3'b111，选择外部时钟模式 A。

-配置 TMRx_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

表 14-7 TMRx内部触发连接

次定时器	IS0 （STIS=000）	IS1 （STIS=001）	IS2 （STIS=010）	IS3 （STIS=011）
TMR9	TMR2	TMR3	TMR10_C1OUT	TMR11_C1OUT
TMR12	TMR4	TMR2	TMR13_C1OUT	TMR14_C1OUT

图 14-43 当预分频器的参数从1变到4时，计数器的时序图



14.3.3.2 计数模式

TMR9 和 TMR12 定时器提供了多种计数模式，用来满足不同的应用场景。其内部拥有一个支持 16 位向上计计数的计数器。

TMRx_PR 寄存器用于设置计数器计数周期。默认 TMRx_PR 寄存器值会立即传入它的影子寄存器；当开启周期缓冲功能后（PRBEN 置 1），TMRx_PR 寄存器值在溢出事件发生时传入它的影子寄存器。

TMRx_DIV 寄存器用于设置计数器计数频率，每（DIV[15:0]+1）个计数时钟周期，计数器计数一次。和 TMRx_PR 寄存器类似，开启周期缓冲功能后，TMRx_DIV 寄存器值在溢出事件时更新至它的影子寄存

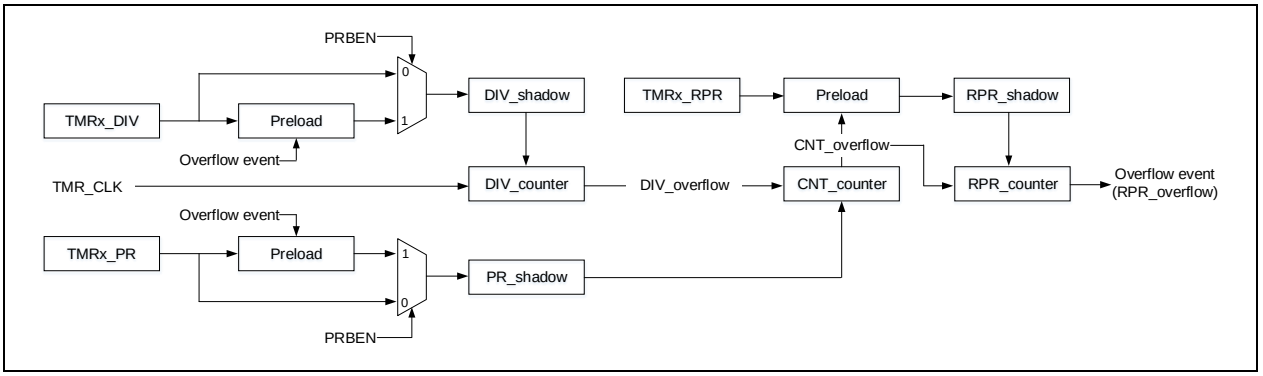
器。

读取 TMRx_CNT 寄存器会返回当前计数器计数值，写入 TMRx_CNT 寄存器会更新计数器当前计数值为写入值。

默认允许产生溢出事件，设置 TMRx_CTRL1 寄存器 OVFEN=1 将禁止溢出事件产生。TMRx_CTRL1 寄存器 OVFS 用于选择溢出事件来源，默认计数器上溢或下溢、置位 OVFSWTR、复位模式次定时器控制器产生的复位信号产生溢出事件。置位 OVFS 后，只有计数器上溢或下溢产生溢出事件。

TMREN 位置 1 将使能定时器计数，由于同步逻辑，实际驱动计数器的使能信号 TMR_EN 相对于 TMREN 延迟一个时钟周期。

图 14-44 计数器基本结构



向上计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]=2'b00，OWCDIR=1'b0 开启向上计数模式，计数值达到 TMRx_PR 值时，重新从 0 向上计数，计数器上溢并产生溢出事件，同时 OVFIF 位置 1。若禁止产生溢出事件，计数器溢出后不再重载预分频值和周期值，否则预分频值和周期值在溢出事件后更新。

图 14-45 PRBEN=0时的溢出事件

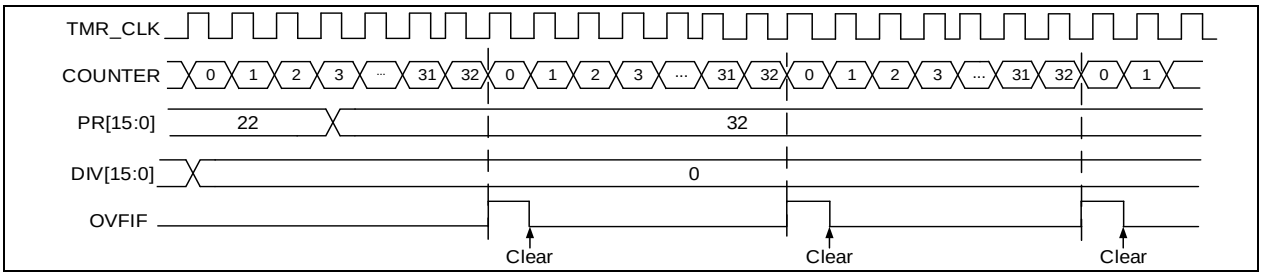
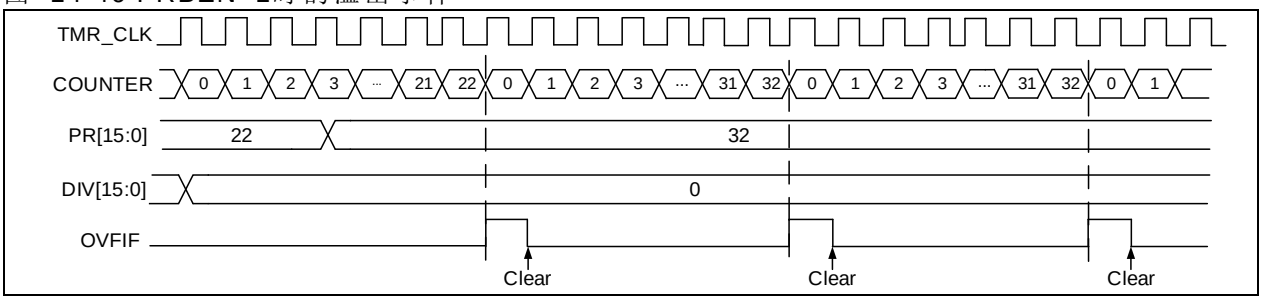


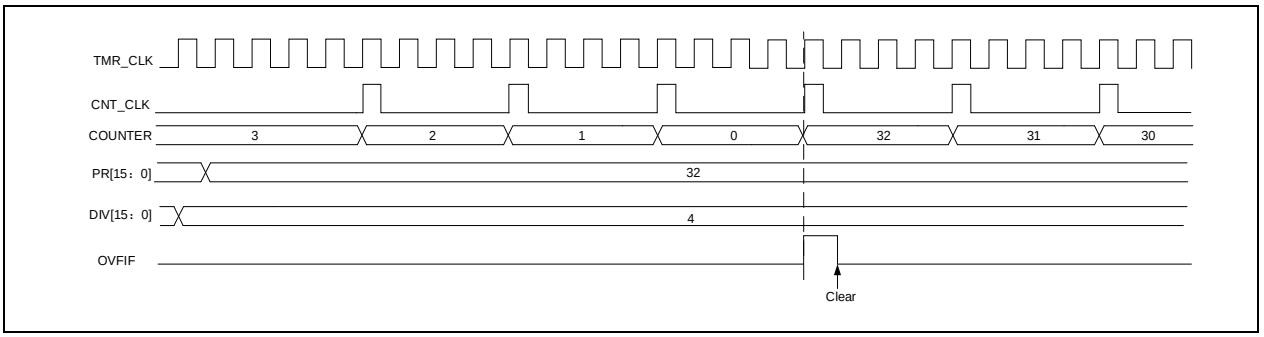
图 14-46 PRBEN=1时的溢出事件



向下计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]=2'b00，OWCDIR=1'b1 开启向下计数模式，计数值达到 0 值并重新从 TMRx_PR 向上下数时，计数器下溢并产生溢出事件。

图 14-47 计数器时序图，内部时钟分频因子为4



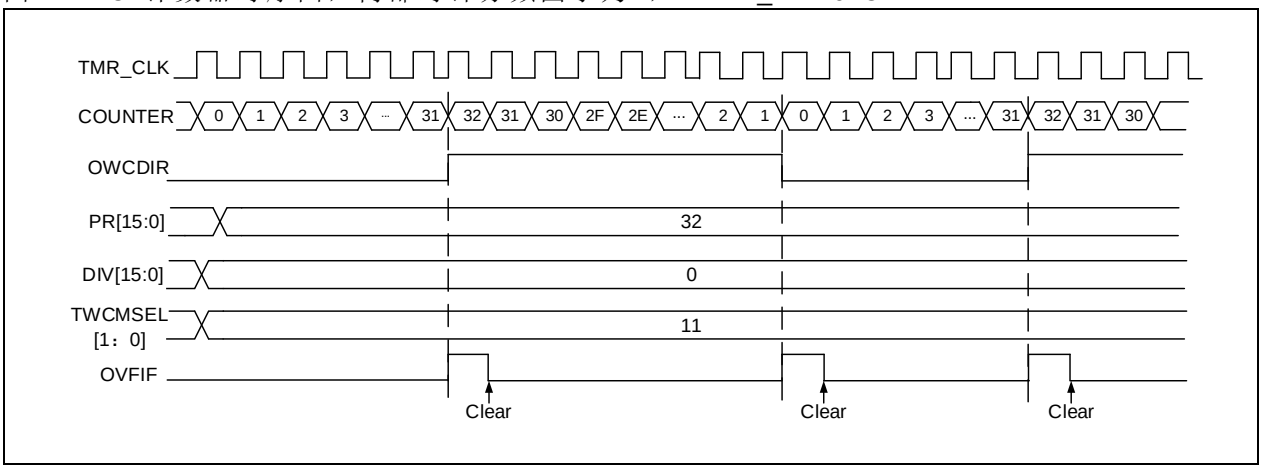
中央双向对齐计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]≠2’b00 开启中央双向对齐计数模式，中央双向对齐计数模式下计数器交替向上、向下计数。计数值从 TMRx_PR 值向下计数到 1 值，产生下溢事件，然后从 0 开始向上计数；向上计数到 TMRx_PR 值-1，产生上溢事件，之后从 TMRx_PR 值向下计数。计数器计数方向由计数器方向控制位（OWCDIR）实时查看。

TMRx_CTRL1 寄存器 TWCMSEL[1:0]位还用于选择中央双向对齐计数模式下 CxIF 标志置起方式，中央双向对齐计数模式 1（TWCMSEL[1:0]=2’b01）仅允许 CxIF 标志位在计数器向下计数时置起；双向对齐计数模式 2（TWCMSEL[1:0]=2’b10）仅允许 CxIF 标志位在计数器向上计数时置起；双向对齐计数模式 3（TWCMSEL[1:0]=2’b11）允许 CxIF 标志位在计数器向上和向下计数时置起。

注意： 中央双向对齐计数模式下，OWCDIR 位为只读位。

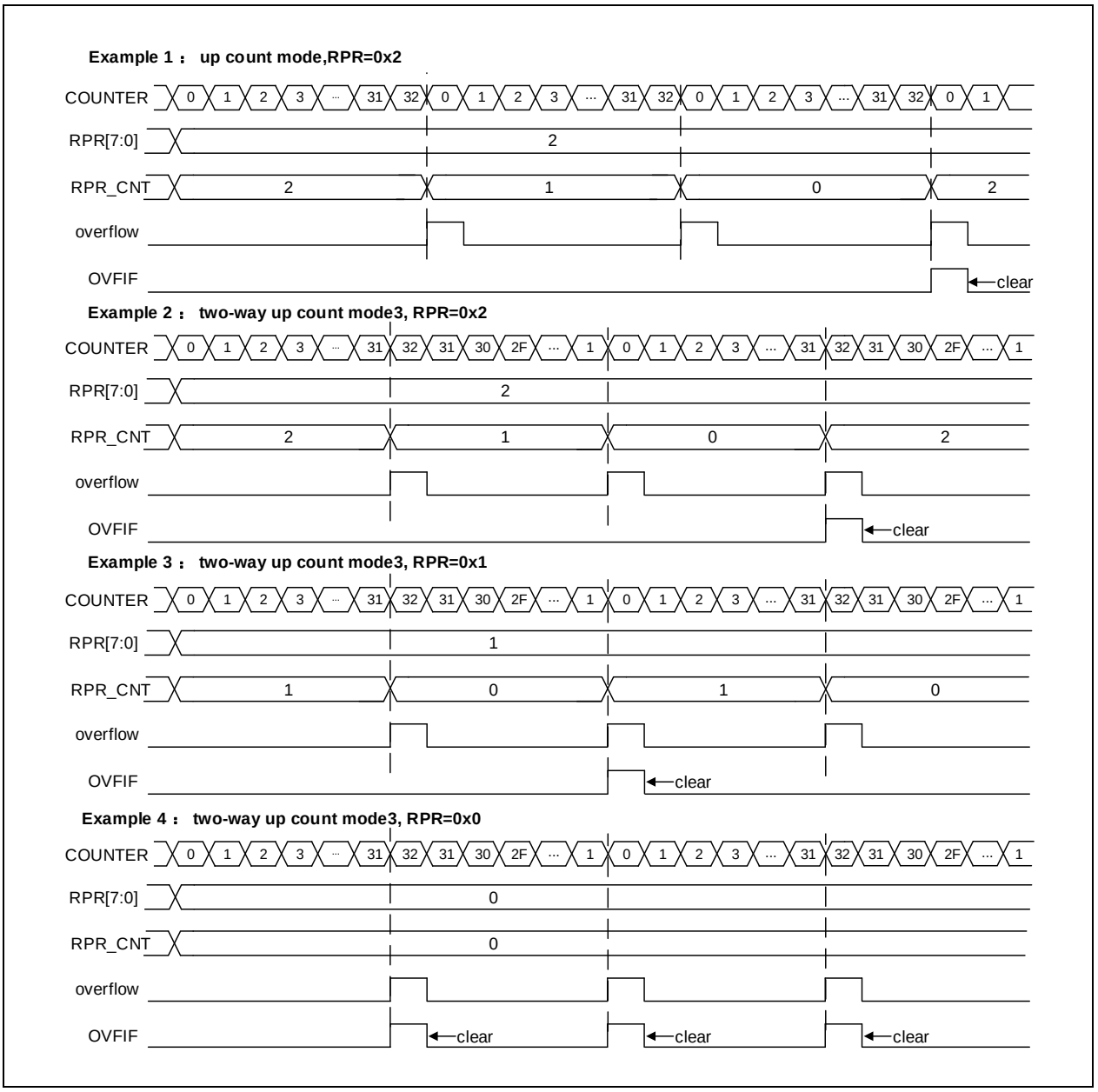
图 14-48 计数器时序图，内部时钟分频因子为1，TMRx_PR=0x32



重复计数模式：

TMRx_RPR 寄存器用于配置重复计数器计数周期，TMRx_RPR 寄存器为非 0 值时，重复计数模式启动。重复计数模式下，每（RPR[7:0]+1）次计数器溢出将产生一次溢出事件。每次计数器溢出，重复计数器递减，仅当重复计数器计数值等于 0 值时，计数器溢出会产生溢出事件。通过配置不同重复计数器值，可调整溢出事件产生的频率。

图 14-49 向上计数模式和中央双向对齐计数模式时OVFIF



14.3.3.3 TMR输入部分

TMR9 和 TMR12 拥有 2 个独立通道，每个通道可配置为输入或输出，当配置位输入时，当配置为输入时，每个通道输入信号依次经过以下处理：

-TMRx_CHx 经过预处理输出 CxIRAW。配置 C1INSEL 位，选择 CxIRAW 来源是 TMRx_CHx。

-CxIRAW 输入数字滤波器，输出滤波后信号 CxIF。数字滤波器通过 CxDF 位配置采样频率和次数。

-CxIF 输入边沿检测器，输出边沿选择后信号 CxIFPx。边沿选择由 CxP 和 CxCP 位共同控制，可选择输入上升沿、下降沿或双边沿有效。

-CxIFPx 输入捕获信号选择器，输出选择后信号 CxIN。捕获信号选择器由 CxC 控制，可选择 CxIN 来源为 CxIFPx、CyIFPx、STCI。其中 CyIFPx（x≠y）是来自通道 y 的 CyIFPy 经通道 x 边沿检测器处理后的信号（例如 C1IFP2 是来自通道 1 的 C1IFP1 信号经过通道 2 边沿检测器处理后的信号）；STCI 来自次定时器控制器，由 STIS 位选择来源。

-CxIN 经由输入通道分频器，输出分频后信号 CxIPS。分频系数由 CxIDIV 位配置为不分频、2 分频、4 分频或 8 分频。

图 14-50 输入/输出通道 1 的主电路

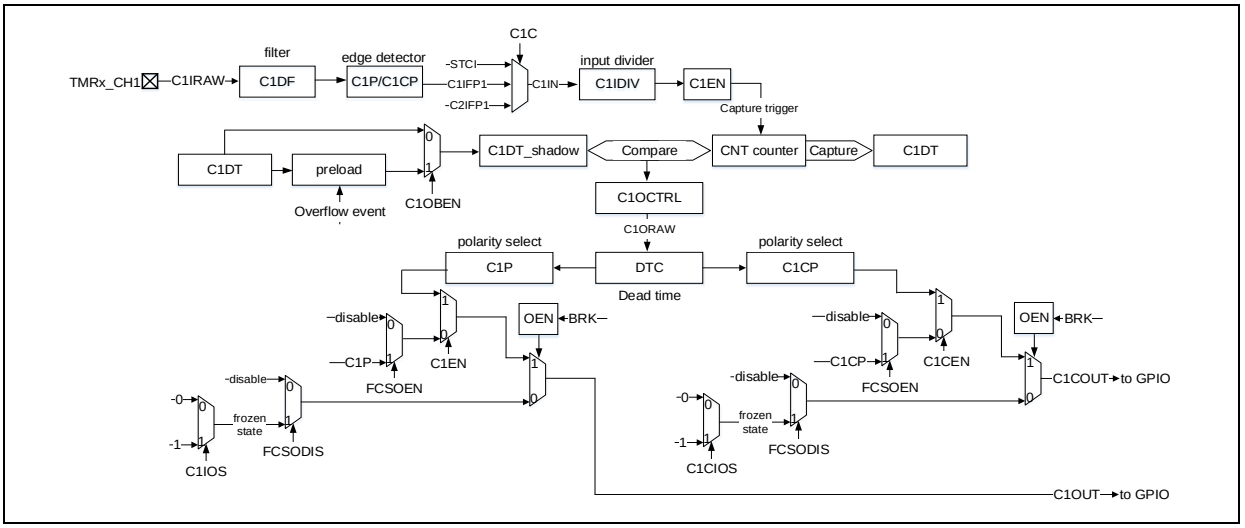
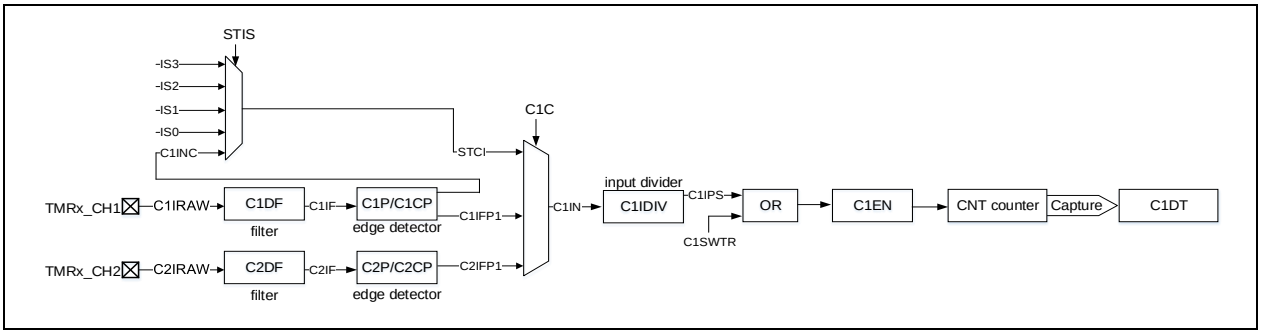


图 14-51 通道1输入部分



输入模式

此模式下，当选中的触发信号被检测到，通道寄存器（TMRx_CxDT）记录当前计数器计数值，并将捕获比较中断标志位（CxIF）置 1，若已使能通道中断（CxIEN）、通道 DMA 请求（CxDEN）则产生相应的中断和 DMA 请求。若在 CxIF 置 1 后检测到触发信号，将产生捕获溢出事件，TMRx_CxDT 会使用当前计数器计数值覆盖之前记录的计数器计数值，同时通道再捕获标志位（CxRF）置 1。

以若要捕获 C1IN 输入的上升沿，可按如下进行配置：

- 将通道模式寄存器 1（TMRx_CM1）中的 C1C 位配置为 01，选择 C1IN 作为通道 1 输入。
- 配置 C1IN 信号滤波器带宽（CxDF[3:0]）。
- 配置 C1IN 通道的有效沿，在 TMRx_CCTRL 寄存器中写入 C1P=0（上升沿）。
- 配置 C1IN 信号捕获分频（C1DIV[1:0]）。
- 使能通道 1 输入捕获（C1EN=1）。
- 根据需要设置 TMRx_IDEN 寄存器中的 C1IEN 位、TMRx_IDEN 寄存器中的 C1DEN 位，选择中断请求或 DMA 请求。

PWM 输入

PWM 输入模式适用于通道 1 和 2，要使用此模式，需要将 C1IN 和 C2IN 映射到同一 TMRx_CHx，并且通道 1 或 2 的 CxIFPx 配置成触发次定时器控制器复位。

PWM 输入模式可用于测量输入信号的周期和占空比，如需测量通道 1 输入信号的周期和占空比，操作步骤如下：

- 配置C1C=2‘b01，选择C1IN为C1IFP1。
- 配置C1P=1’b0，选择C1IFP1上升沿有效。
- 配置C2C=2‘b10，选择C2IN为C1IFP2。
- 配置C2P=1’b1，选择C1IFP2下降沿有效。
- 配置STIS=3’b101，选择次定时器触发信号为C1IFP1。
- 配置SMSEL=3‘b100，选择次定时器模式为复位模式。

- 配置C1EN=1’b1，C2EN=1’b1。使能通道1和输入捕获。

上述配置下，通道 1 输入信号的上升沿会触发捕获并将捕获值存储到 C1DT 寄存器，同时通道 1 输入信号上升沿复位计数器。通道 1 输入信号下降沿触发捕获并将捕获值存储到 C2DT 寄存器。通道 1 输入信号的周期可通过 C1DT 计算，占空比可通过 C2DT 计算。

图 14-52 PWM输入模式配置实例

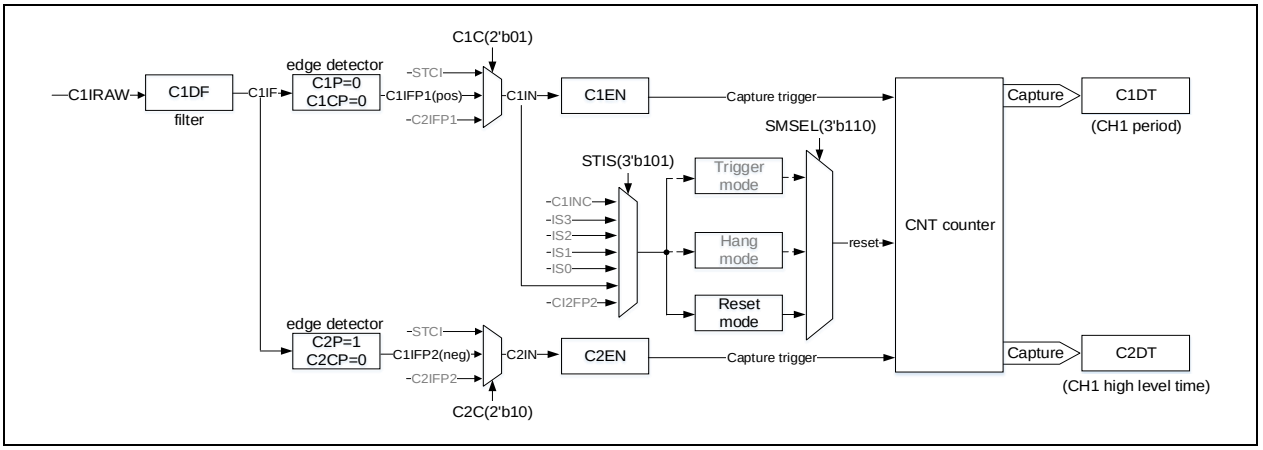
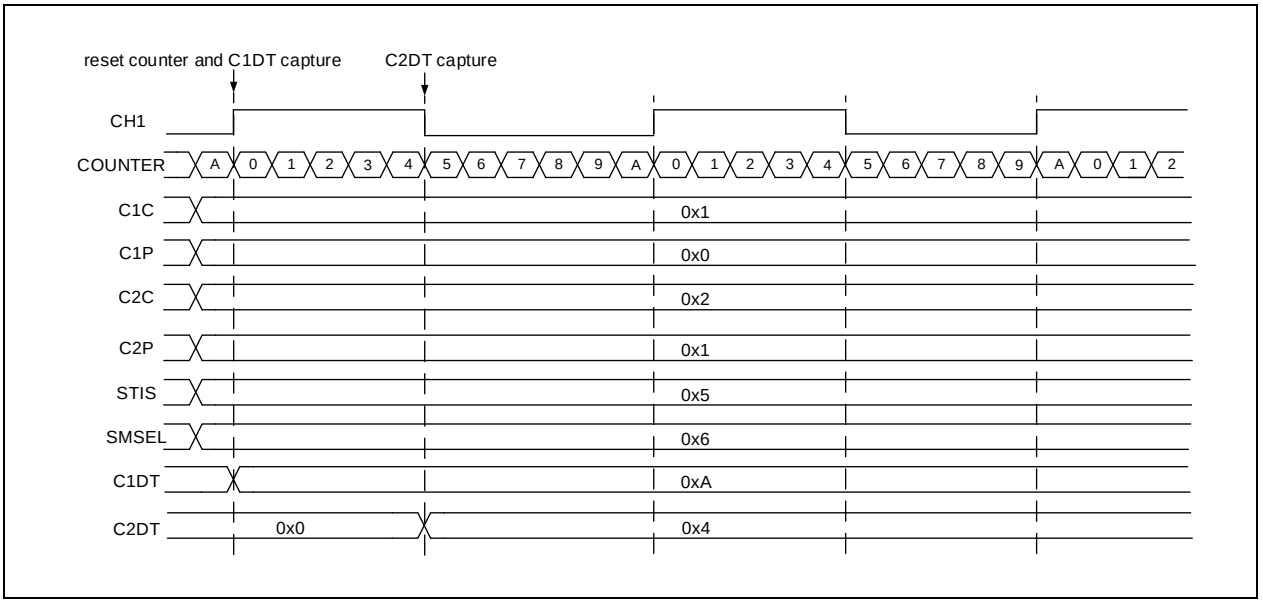


图 14-53 PWM输入模式



14.3.3.4 TMR输出部分

TMR 的输出部分由比较器和输出控制构成，用于编程输出信号的周期、占空比、极性。TMR9 和 TMR12 的输出部分在不同通道上有所不同，如下图所示：

图 14-54 通道1输出部分

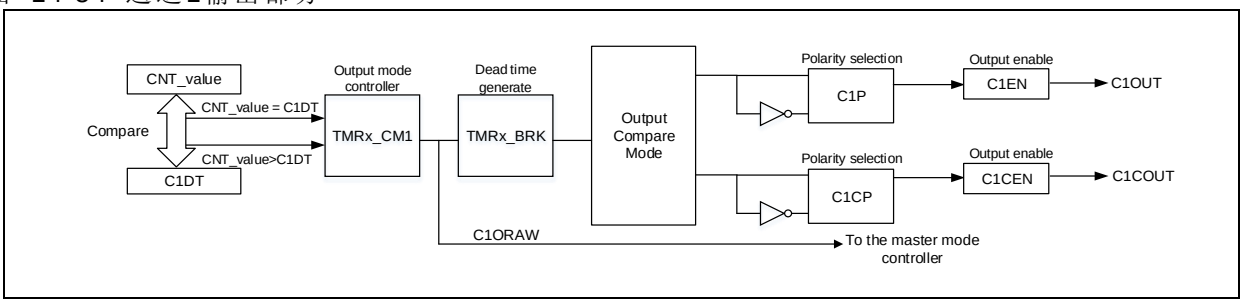
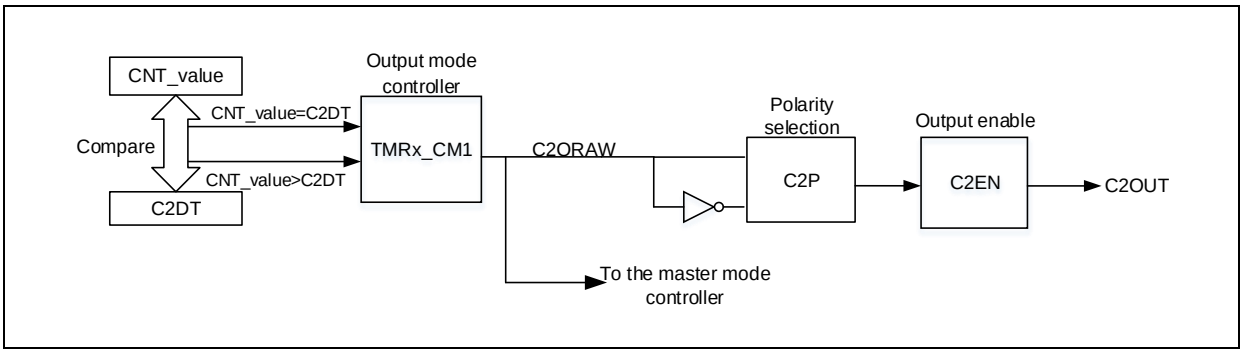


图 14-55 通道2输出部分



输出模式

配置 CxC[1:0]≠2'b00 将通道配置为输出可实现多种输出模式，此时，计数器计数值将与 CxDT 寄存器值比较，并根据 CxOCTRL[2:0]位配置的输出模式，产生中间信号 CxORAW，再经过输出控制逻辑处理后输送到 IO。输出信号的周期由 TMRx_PR 寄存器值配置，占空比则由 CxDT 寄存器值配置。

输出比较模式有以下子类：

PWM 模式 A：CxOCTRL=3'b110 时，开启 PWM 模式 A。向上计数时，TMRx_C1DT>TMRx_CVAL 时 C1ORAW 输出高电平，否则为低电平；向下计数时，TMRx_C1DT<TMRx_CVAL 时 C1ORAW 输出低电平，否则为高电平。若要使用 PWM 模式 A，可按如下方式配置。

- -配置 TMRx_PR 寄存器，设置 PWM 周期。
- -配置 TMRx_CxDT 寄存器，设置 PWM 占空比。
- -配置 TMRx_CM1/CM2 寄存器 CxOCTRL 位为 3'b110，设置输出模式为 PWM 模式 A。
- -配置 TMRx_DIV 寄存器，设置计数器计数频率。
- -配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
- -配置 TMRx_CCTRL 寄存器 CxP 位、CxCP 位，设置输出极性。
- -配置 TMRx_CCTRL 寄存器 CxEN 位、CxCEN 位，使能通道输出。
- -配置 TMRx_BRK 寄存器 OEN 位，使能 TMRx 输出。
- -配置 TMR 输出通道对应 GPIO 为对应的复用模式。
- -配置 TMRx_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

- **PWM 模式 B**：CxOCTRL=3'b111 时，开启 PWM 模式 B。向上计数时，TMRx_C1DT>TMRx_CVAL 时 C1ORAW 输出低电平，否则为高电平；向下计数时，TMRx_C1DT<TMRx_CVAL 时 C1ORAW 输出高电平，否则为低电平。
- **强制输出模式**：CxOCTRL=3'b100/101 时，开启强制输出模式。此时，CxORAW 信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。
- **输出比较模式**：CxOCTRL=3'b001/010/011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 强制输出高电平（CxOCTRL=3'b001）、低电平（CxOCTRL=3'b010）或进行电平翻转（CxOCTRL=3'b011）。
- **单周期模式**：PWM 模式的特例，将 OCMEN 位置 1 可开启单周期模式，此模式下，仅在当前计数周期中进行比较匹配，完成当前计数后，TMREN 位清 0，因此仅输出一个脉冲。当配置为向上计数模式时，需要严格配置 CVAL<CxDT≤PR；向下计数时，需严格配置 CVAL>CxDT。
- **快速输出模式**：将 CxOIEN 位置 1 可开启此功能，开启后 CxORAW 电平值不再在计数值与 CxDT 匹配时变化，而是在当前计数周期开始时，也就是说，比较结果被提前了，计数器值与 CxDT 寄存器的比较结果将会提前决定 CxORAW 的电平。

图 14-58 展示了输出比较模式（翻转）的例子，C1DT=0x3，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-59 展示了计数器向上计数与 PWM 模式 A 配合的例子，PR=0x32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-60 展示了计数器向上计数与单周期模式下 PWM 模式 B 配合的例子，计数器仅计数了一个周期，输出信号在这个周期中只输出了一个脉冲。

图 14-56 计数值与C1DT值匹配时翻转C1ORAW

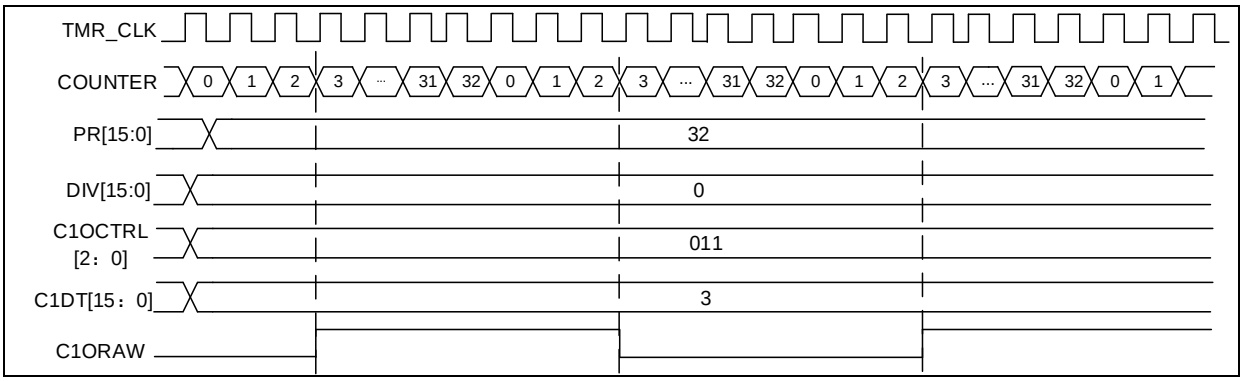


图 14-57 向上计数下PWM模式A

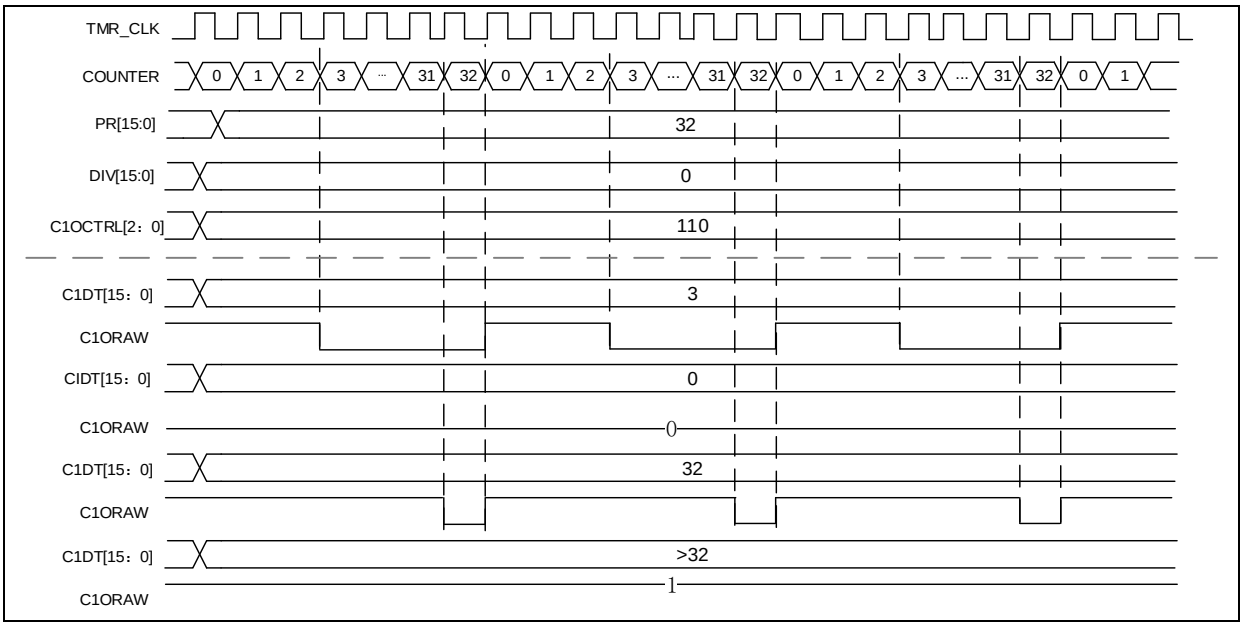
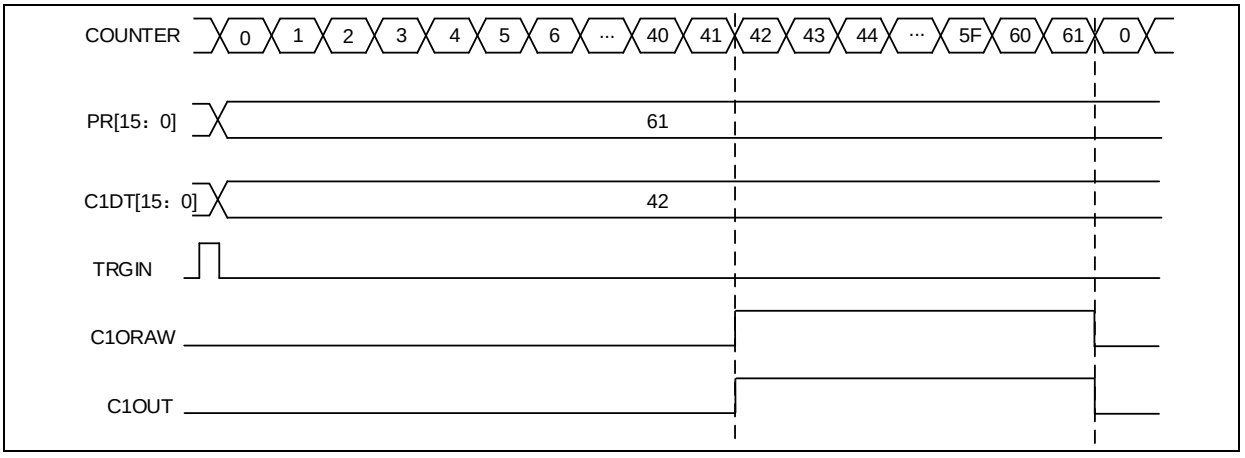


图 14-58 单周期模式



主定时器事件输出

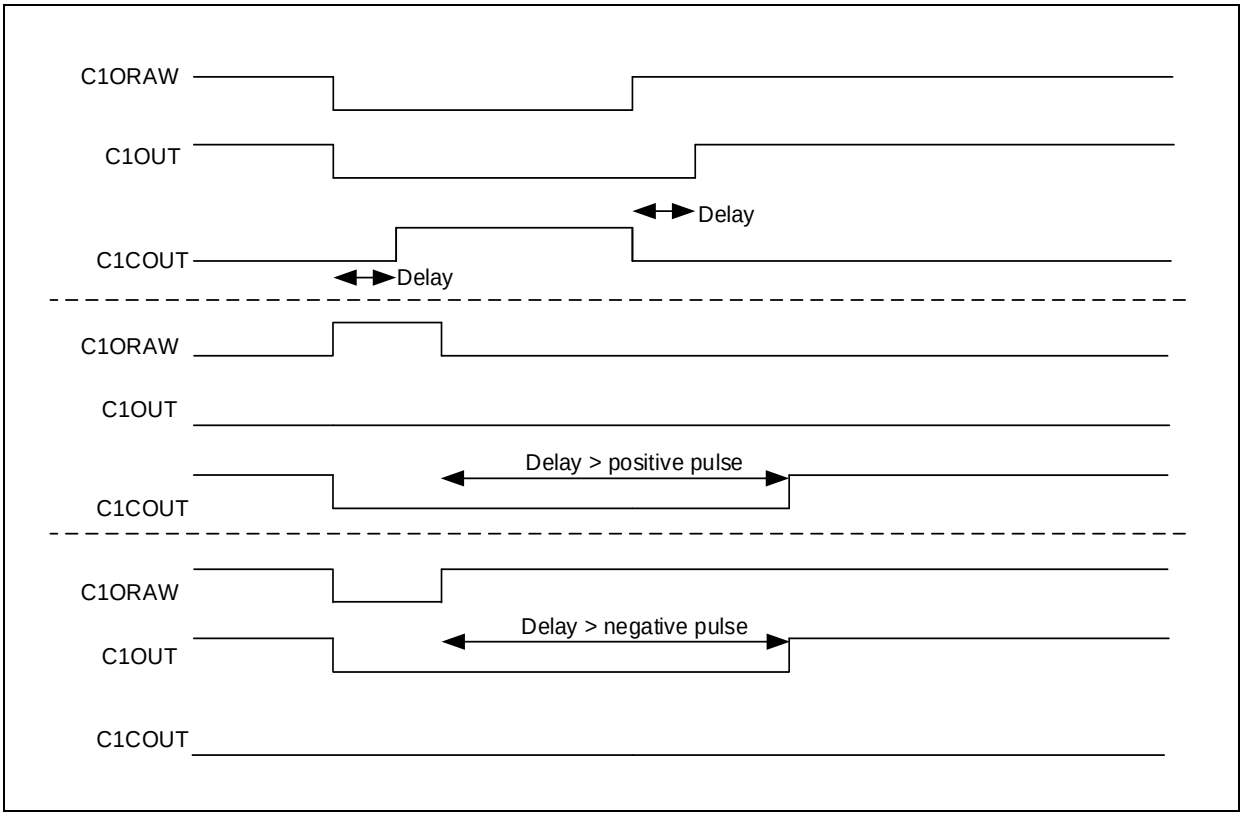
当 TMR 作为主定时器时，可选择如下信号源作为 TRGOUT 信号输出到次定时器，选择信号为 TMRxCTRL2 寄存器 PTOS 位。

- -PTOS=3’b000，TRGOUT 输出软件溢出事件（TMRx_SWEVT 寄存器 OVFSWTR 位）。
- -PTOS=3’b001，TRGOUT 输出计数器使能信号。
- -PTOS=3’b010，TRGOUT 输出计数器溢出事件。
- -PTOS=3’b011，TRGOUT 输出捕获、比较事件。
- -PTOS=3’b100，TRGOUT 输出 C1ORAW 信号。

- -PTOS=3’b101，TRGOUT 输出 C2ORAW 信号。

死区插入

TMR9 和 TMR12 通道 1 包含一组反向通道输出，通过 CxCEN 使能，通过 CxCP 配置极性。
当转换为 IDLEF 状态，即 OEN 下降到 0，死区被激活。
将 CxEN 和 CxCEN 位置 1 后，通过配置 DTC[7:0]死区发生器，可插入不同时长的死区。插入死区后，CxOUT 的上升沿延迟于参考信号的上升沿；CxCOUT 的上升沿延迟于参考信号的下降沿。
如果延迟大于当前有效的输出宽度，C1OUT 和 C1COUT 不会产生相应的脉冲，死区时间应小于有效的输出宽度。
下列图显示了 CxP=0、CxCP=0、OEN=1、CxEN=1 并且 CxCEN=1 时死区插入的例子
图 14-59 带死区插入的互补输出



14.3.3.5 TMR刹车功能

开启刹车功能后（BRKEN 位置 1），CxOUT 和 CxCOUT 由 OEN、FCSODIS、FCSOEN、CxIOS 和 CxCIOS 共同控制。但 CxOUT 和 CxCOUT 输出总是不能同时处于有效电平上的。
刹车信号来源可以是刹车输入引脚、时钟失效事件，刹车输入信号的极性由 BRKV 位控制。
当发生刹车事件时，有下述动作：

- OEN位异步清零，通道输出状态由FCSODIS位选择。关闭MCU的振荡器不影响该功能。
- OEN被清零后，通道输出电平由CxIOS位设定。如果FCSODIS=0，则定时器输出使能被禁止，否则输出使能始终为高。
- 当使用互补输出时：
  - 输出最开始处于复位状态，也就是无效的状态（取决于极性）。这是异步操作，定时器有无时钟并不影响此功能。
  - 定时器的时钟如果有效，会开启死区生成功能，CxIOS和CxCIOS位用来配置死区之后的电平。即使在这种情况下，CxOUT和CxCOUT也不能被同时驱动到有效的电平。

注意，由于OEN 位同步逻辑，死区时间较通常情况会延长一段时间（大约 2 个 clk_tmr 的时钟周期）。

  - 如果FCSODIS=0，定时器释放使能输出，否则保持使能输出；或一旦CxEN与CxCEN之一变高时，使能输出变为高。
- 如果开启了刹车中断或DMA功能，刹车状态标志将置1，并产生刹车中断或DMA请求。
- 如果将AOEN位置1，在下一个溢出事件时OEN位被自动置1。

注意：刹车输入电平有效时，OEN 不能被设置，状态标志 BRKIF 也不能被清除。

图 14-60 TMR输出控制

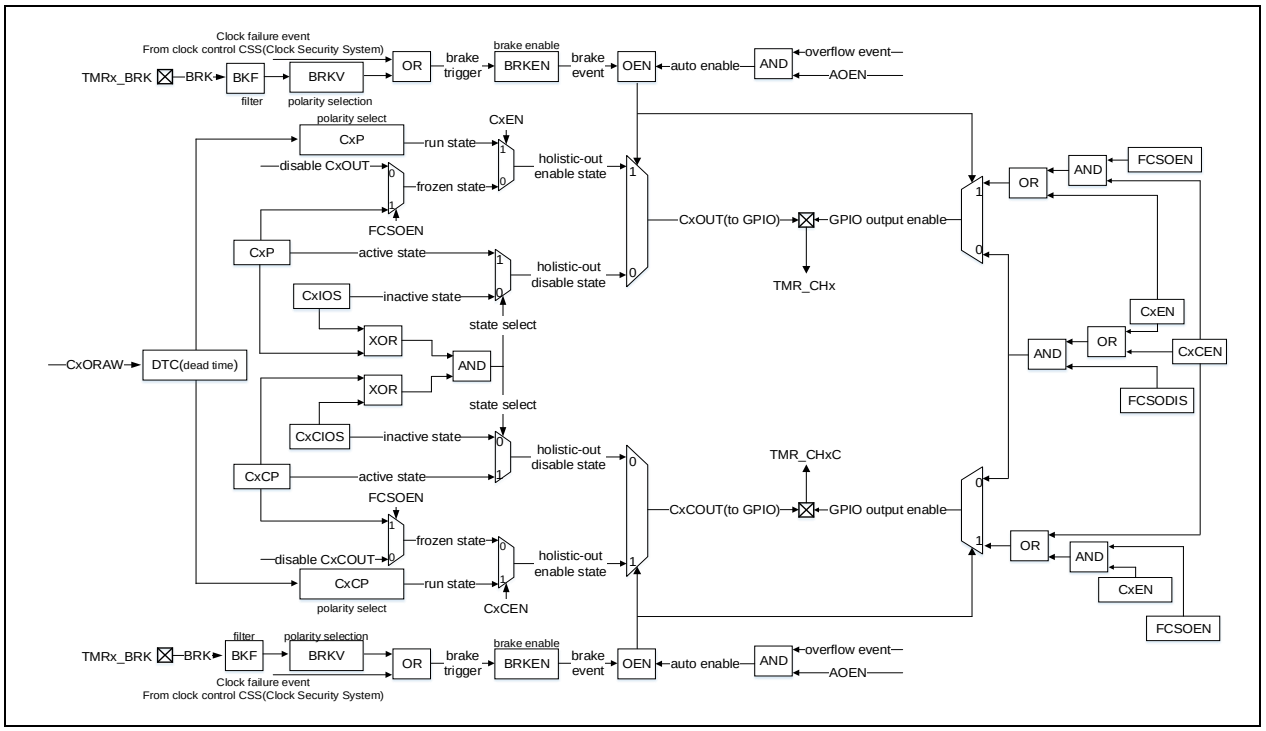
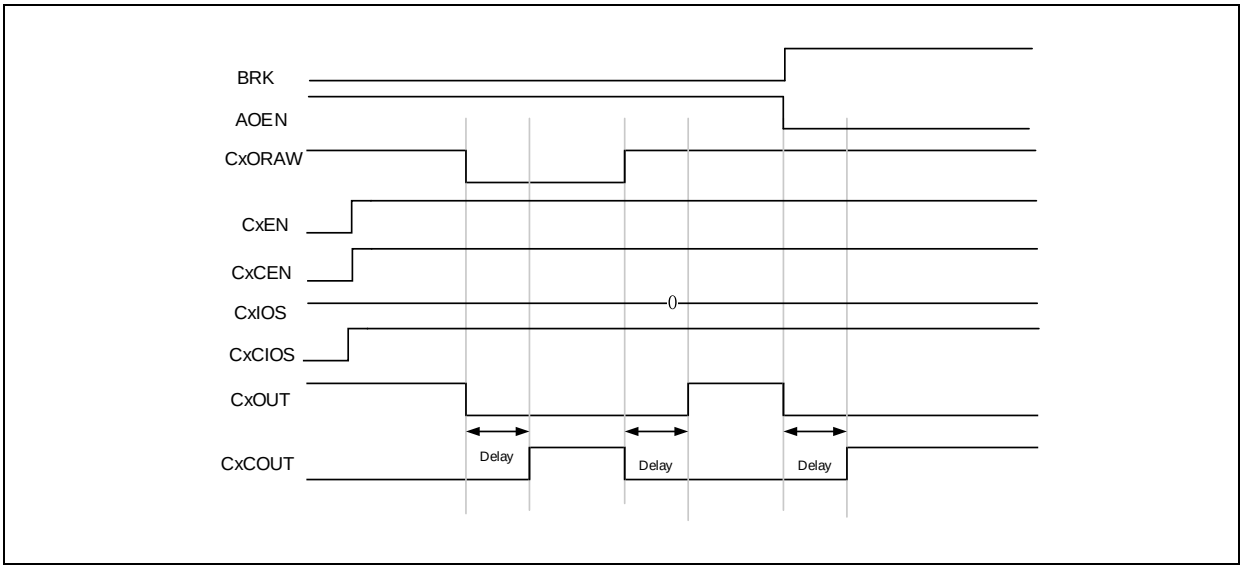


图 14-61 TMR刹车功能的例子



14.3.3.6 TMR同步

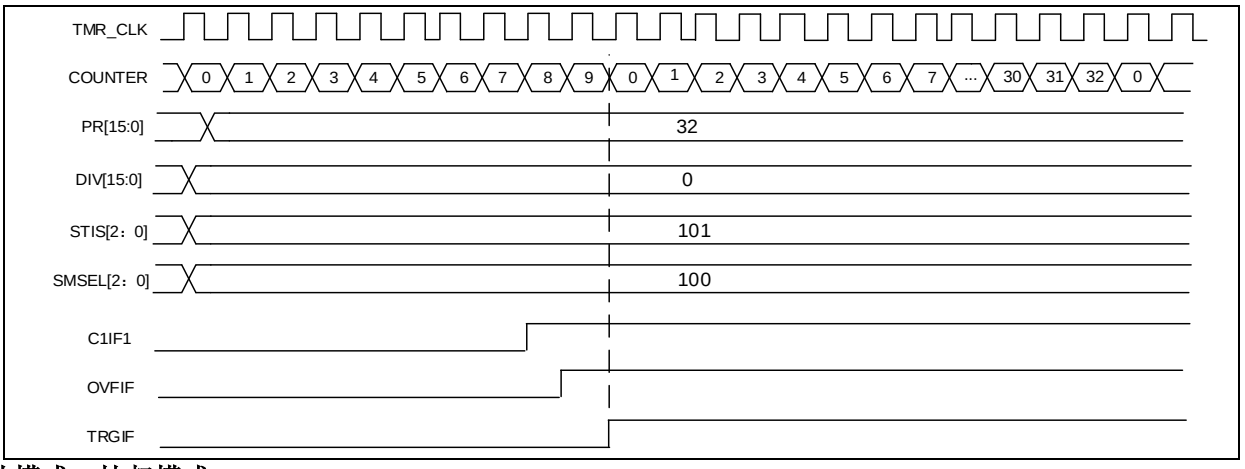
主次定时器之间可由内部连接信号进行同步。主定时器可由 PTOS[2:0]位选择主定时器输出，即同步信息；次定时器由 SMSEL[2:0]位选择从模式，即次定时器的工作模式。

定时器从模式有以下几种：

从模式：复位模式

选中的触发信号将复位计数器和预分频器，若 OVFS 位为 0，将产生一个溢出事件。

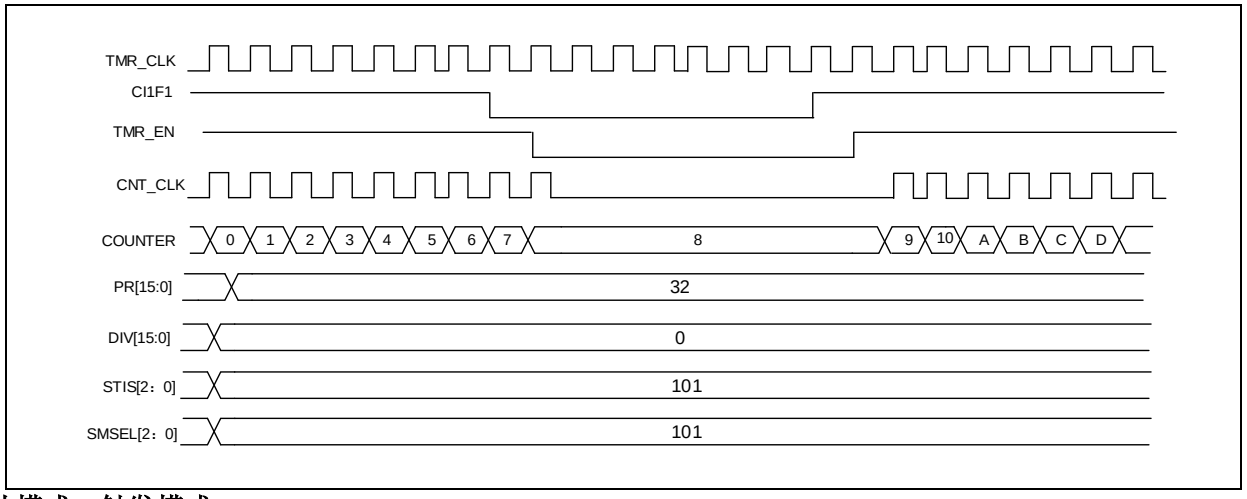
图 14-62 复位模式例子



从模式：挂起模式

挂起模式下，计数的计数和停止受选中触发输入信号控制，当触发输入为高电平时计数器开始计数；当为低电平时，计数器暂停计数。

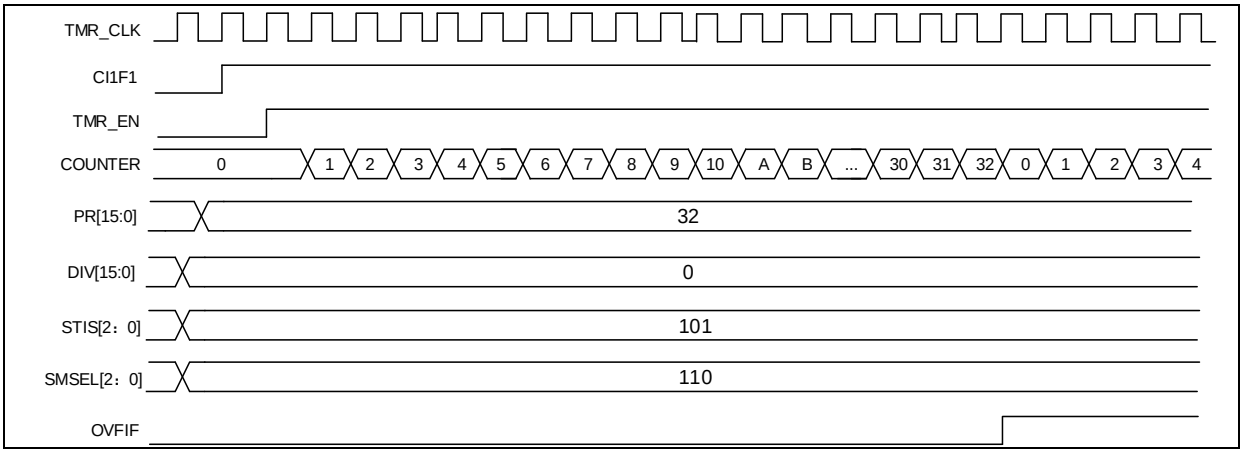
图 14-63 挂起模式下例子



从模式：触发模式

计数器将在选中的触发输入上升沿启动计数（将 TMR_EN 置 1）。

图 14-64 触发器模式例子



14.3.3.7 调试模式

当微控制器进入调试模式（Cortex®-M4F 核心停止）时，将 DEBUG 模块中的 TMRx_PAUSE 置 1，可以使 TMRx 计数器暂停计数。

14.3.4 TMR9、TMR12寄存器描述

表 14-8 TMR9和TMR12寄存器图和复位值

寄存器简称	基址偏移量	复位值
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_STCTRL	0x08	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_RPR	0x30	0x0000
TMRx_C1DT	0x34	0x0000
TMRx_C2DT	0x38	0x0000
TMRx_BRK	0x44	0x0000
TMRx_DMACTRL	0x48	0x0000
TMRx_DMADT	0x4C	0x0000

14.3.4.1 TMR9和TMR12控制寄存器1（TMRx_CTRL1）

域	简称	复位值	类型	功能
位 15：10	保留	0x00	resd	保持默认值。
位 9：8	CLKDIV	0x0	rw	时钟除频（Clock divider） 此位用于设置数字滤波器采样频率 f_{DTS} 和定时器时钟频率 f_{CK_INT} 之间的分频比，也用于调整死区时间的时基 T_{DTS} 和定时器时钟周期 T_{CK_INT} 的分频比。 00：无除频，$f_{DTS}=f_{CK\_INT}$； 01：2 除频，$f_{DTS}=f_{CK\_INT}/2$； 10：4 除频，$f_{DTS}=f_{CK\_INT}/4$； 11：保留。
位 7	PRBEN	0x0	rw	周期缓冲使能（Period buffer enable） 0：缓冲关闭； 1：缓冲开启。
位 6：5	TWCMSEL	0x0	rw	中央双向对齐计数模式选择 （Two-way count mode selection） 00：单向对齐计数模式，方向由 OWCDIR 配置； 01：中央双向对齐计数模式 1，上下交替计数，CxIF 位只在计数器向下计数时被置起； 10：中央双向对齐计数模式 2，上下交替计数，CxIF 位只在计数器向上计数时被置起； 11：中央双向对齐计数模式 3，上下交替计数，CxIF 位在计数器向上和向下计数时皆被置起。
位 4	OWCDIR	0x0	rw	单向对齐计数方向 （One-way count direction） 0：向上； 1：向下。
位 3	OCMEN	0x0	rw	单周期使能 （One cycle mode enable） 该功能用于选择溢出事件后，计数器是否停止。 0：关闭；

				1：开启。
位 2	OVFS	0x0	rw	溢出事件源选择 （Overflow event source） 配置溢出事件或 DMA 请求来源。 0：来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件； 1：只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 （Overflow event enable） 0：开启； 1：关闭。
位 0	TMREN	0x0	rw	使能定时器（TMR enable） 0：关闭； 1：开启。

14.3.4.2 TMR9和TMR12控制寄存器2（TMRx_CTRL2）

域	简称	复位值	类型	功能
位 15：12	保留	0x00	resd	保持默认值。
位 11	C2CIOS	0x0	rw	通道 2 互补空闲输出状态（Channel 2 complementary idle output state）
位 10	C2IOS	0x0	rw	通道 2 空闲输出状态（Channel 2 idle output state）
位 9	C1CIOS	0x0	rw	通道 1 互补空闲输出状态（Channel 1 complementary idle output state） 输出关闭（OEN = 0），死区发生后： 0：C1COUT=0； 1：C1COUT=1。
位 8	C1IOS	0x0	rw	通道 1 空闲输出状态（Channel 1 idle output state） 输出关闭（OEN = 0），死区发生后： 0：C1OUT=0。 1：C1OUT=1。
位 7	保留	0x0	resd	保持默认值。
位 6：4	PTOS	0x0	rw	主定时器输出信号选择 （Primary TMR output selection） TMRx 输出到次定时器的信号选择： 000：复位； 001：使能； 010：溢出； 011：比较脉冲； 100：C1ORAW 信号； 101：C2ORAW 信号； 110：C3ORAW 信号； 111：C4ORAW 信号。
位 3	DRS	0x0	rw	DMA 请求源（DMA request source） DMA 请求来源。 0：通道事件； 1：溢出事件。
位 2	CCFS	0x0	rw	通道控制位刷新选择（Channel control bit refresh select） 对具有互补输出的通道，如果通道控制位有缓存时： 0：通过设置 HALL 位刷新控制位； 1：通过设置 HALL 位或 TRGIN 的上升沿刷新控制位。
位 1	保留	0x0	resd	保持默认值。
位 0	CBCTRL	0x0	rw	通道缓存控制（Channel buffer control） 对具有互补输出的通道： 0：CxEN，CxCEN 和 CxOCTRL 位无缓存； 1：CxEN，CxCEN 和 CxOCTRL 位有缓存。

14.3.4.3 TMR9和TMR12次定时器控制寄存器（TMRx_STCTRL）

域	简称	复位值	类型	功能
位 15：8	保留	0x00	resd	保持默认值。
位 7	STS	0x0	rw	次定时器同步 （Subordinate TMR synchronization）

				该位开启后，主次定时器可实现高度同步。 0：关闭； 1：开启。
位 6：4	STIS	0x0	rw	次定时器输入选择 （Subordinate TMR input selection） 用于次定时器的输入选择。 000：内部选择 0（IS0）； 001：内部选择 1（IS1）； 010：内部选择 2（IS2）； 011：内部选择 3（IS3）； 100：C1IRAW 的输入检测器（C1INC）； 101：滤波输入 1（C1IF1）； 110：滤波输入 2（C2IF2）； 111：外部输入（EXT）。 关于每个定时器中 ISx 的细节，参见表 14-7。
位 3	保留	0x0	resd	保留，保持默认值。
位 2：0	SMSEL	0x0	rw	次定时器模式选择 （Subordinate TMR mode selection） 000：关闭从模式； 001：保留； 010：保留； 011：保留； 100：复位模式 - TRGIN 输入上升沿时，重新初始化计数器； 101：挂起模式 - TRGIN 输入高电平时，计数器计数； 110：触发模式 - TRGIN 输入上升沿时，产生触发事件； 111：外部时钟模式 A - TRGIN 输入上升沿提供时钟；

14.3.4.4 TMR9和TMR12 DMA中断使能寄存器（TMRx_IDEN）

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值。
位 14	TDEN	0x0	rw	触发 DMA 请求使能（Trigger DMA request enable） 0：关闭； 1：开启。
位 13	HALLDE	0x0	rw	HALL DMA 请求使能 （HALL DMA request enable） 0：关闭； 1：开启。
位 12：11	保留	0x00	resd	保持默认值。
位 10	C2DEN	0x0	rw	通道 2 的 DMA 请求使能 （Channel 2 DMA request enable） 0：关闭； 1：开启。
位 9	C1DEN	0x0	rw	通道 1 的 DMA 请求使能 （Channel 1 DMA request enable） 0：关闭； 1：开启。
位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能（overflow event DMA request enable） 0：关闭； 1：开启。
位 7	BRKIE	0x0	rw	刹车中断使能（Brake interrupt enable） 0：关闭； 1：开启。
位 6	TIEN	0x0	rw	触发中断使能（Trigger interrupt enable） 0：关闭； 1：开启。
位 5	HALLIEN	0x0	rw	HALL 中断使能（HALL interrupt enable） 0：关闭； 1：开启。
位 4：3	保留	0x00	resd	保持默认值。

位 2	C2IEN	0x0	rw	通道 2 中断使能（Channel 2 interrupt enable） 0：关闭； 1：开启。
位 1	C1IEN	0x0	rw	通道 1 中断使能（Channel 1 interrupt enable） 0：关闭； 1：开启。
位 0	OVFIEN	0x0	rw	溢出中断使能（Overflow interrupt enable） 0：关闭； 1：开启。

14.3.4.5 TMR9和TMR12中断状态寄存器（TMRx_ISTS）

域	简称	复位值	类型	功能
位 15：11	保留	0x00	resd	保持默认值。
位 10	C2RF	0x0	rw0c	通道 2 再捕获标记 （Channel 2 recapture flag） 见 C1RF 的描述。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记 （Channel 1 recapture flag） C1IF 的状态已经为’1’时是否再次发生了捕获，由硬件置’1’，写’0’清除。 0：无捕获发生； 1：捕获发生。
位 8	保留	0x0	resd	保持默认值。
位 7	BRKIF	0x0	rw0c	刹车中断标记 （Brake interrupt flag） 用于标记刹车输入的电平是否有效，由硬件置’1’，写’0’清除。 0：无效； 1：有效。
位 6	TRGIF	0x0	rw0c	触发中断标记 （Trigger interrupt flag） 当发生触发事件时由硬件置’1’，写’0’清除。 0：无触发事件发生； 1：发生触发事件。 触发事件：在 TRGIN 接收到有效边沿，或挂起模式下接收到任意边沿。
位 5	HALLIF	0x0	rw0c	HALL 中断标记 （HALL interrupt flag） 当发生触发事件时由硬件置’1’，写’0’清除。 0：无 HALL 事件发生； 1：发生 HALL 事件。 HALL 事件：CxEN、CxCEN、CxOCTRL 已被更新。
位 4：3	保留	0x0	resd	保持默认值。
位 2	C2IF	0x0	rw0c	通道 2 中断标记 （Channel 2 interrupt flag） 见 C1IF 的描述。
位 1	C1IF	0x0	rw0c	通道 1 中断标记 （Channel 1 interrupt flag） 若通道 1 为输入模式时： 捕获事件发生时由硬件置’1’，由软件清’0’或读 TMRx_C1DT 清’0’。 0：无捕获事件发生； 1：发生捕获事件。 若通道 1 为输出模式时： 比较事件发生时由硬件置’1’，由软件清’0’。 0：无比较事件发生； 1：发生比较事件。
位 0	OVFIF	0x0	rw0c	溢出中断标记 （Overflow interrupt flag） 当溢出事件发生时由硬件置’1’，由软件清’0’。 0：无溢出事件发生； 1：发生溢出事件，若 TMRx_CTRL1 的 OVFEN=0、OVFS=0 时： – 当 TMRx_SWEVE 寄存器的 OVFG=1 时产生溢出事件； – 当计数值 CVAL 被触发事件重初始化时产生溢出事件。

14.3.4.6 TMR9和TMR12软件事件寄存器（TMRx_SWEVT）

域	简称	复位值	类型	功能
位 15：8	保留	0x00	resd	保持默认值。
位 7	BRKSWTR	0x0	wo	软件触发刹车事件（Brake event triggered by software） 通过软件触发一个刹车事件。 0：无作用； 1：制造一个刹车事件。
位 6	TRGSWTR	0x0	wo	软件触发触发事件（Trigger event triggered by software） 通过软件触发一个触发事件。 0：无作用； 1：制造一个触发事件。
位 5	HALLSWTR	0x0	wo	软件触发 HALL 事件（HALL event triggered by software） 通过软件产生一个 HALL 事件。 0：无作用； 1：产生一个 HALL 事件。 注：该位只对拥有互补输出的通道有效。
位 4：3	保留	0x0	resd	保持默认值。
位 2	C2SWTR	0x0	wo	软件触发通道 2 事件（Channel 2 event triggered by software） 见 C1M 的描述。
位 1	C1SWTR	0x0	wo	软件触发通道 1 事件（Channel 1 event triggered by software） 通过软件触发一个通道 1 事件。 0：无作用； 1：制造一个通道 1 事件。
位 0	OVFSWTR	0x0	wo	软件触发溢出事件（Overflow event triggered by software） 通过软件触发一个溢出事件。 0：无作用； 1：制造一个溢出事件。

14.3.4.7 TMR9和TMR12通道模式寄存器1（TMRx_CM1）

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 CxC 定义。该寄存器其它位的作用在输入和输出模式下不同。CxOx 描述了通道在输出模式下的功能，CxIx 描述了通道在输出模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

输出比较模式：

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值。
位 14：12	C2OCTRL	0x0	rw	通道 2 输出控制（Channel 2 output control）
位 11	C2OBEN	0x0	rw	通道 2 输出缓存使能（Channel 2 output buffer enable）
位 10	C2OIEN	0x0	rw	通道 2 输出立即使能（Channel 2 output immediately enable）
位 9：8	C2C	0x0	rw	通道 2 配置（Channel 2 configure） 当 C2EN='0'时，这些位用于选择通道 2 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C2IN 映射在 C2IFP2 上； 10：输入，C2IN 映射在 C1IFP2 上； 11：输入，C2IN 映射在 STI 上，只有在 STIS 选择内部触发输入时才工作。
位 7	保留	0x0	resd	保持默认值。
位 6：4	C1OCTRL	0x0	rw	通道 1 输出控制（Channel 1 output control） 这些位用于设置原始信号 C1ORAW 的工作状态。 000：断开。断开 C1ORAW 到 C1OUT 的输出； 001：设置 C1ORAW 为高：TMRx_CVAL=TMRx_C1DT 时。

				010：设置 C1ORAW 为低：TMRx_CVAL=TMRx_C1DT 时。 011：切换 C1ORAW 的电平：当 TMRx_CVAL=TMRx_C1DT 时。 100：固定 C1ORAW 为低。 101：固定 C1ORAW 为高。 110：PWM 模式 A －OWCDIR=0，若 TMRx_C1DT>TMRx_CVAL 时设置 C1ORAW 为高，否则为低； －OWCDIR=1，若 TMRx_ C1DT <TMRx_CVAL 时设置 C1ORAW 为低，否则为高。 111：PWM 模式 B －OWCDIR=0，若 TMRx_ C1DT >TMRx_CVAL 时设置 C1ORAW 为低，否则为高； －OWCDIR=1，若 TMRx_ C1DT <TMRx_CVAL 时设置 C1ORAW 为高，否则为低。 *注：除'000'外，其余配置下 C1OUT 将连接到 C1ORAW，C1OUT 的输出电平除了会根据 C1ORAW 变化外，还与 CCTRL 所配置的输出极性有关。*
位 3	C1OBEN	0x0	rw	通道 1 输出缓存使能（Channel 1 output buffer enable） 0：关闭 TMRx_C1DT 的缓存功能，写入 TMRx_C1DT 的内容会立即生效。 1：启用 TMRx_C1DT 的缓存功能，写入 TMRx_C1DT 的内容将保存到缓存寄存器中，当发生溢出事件时再更新到 TMRx_C1DT 中。
位 2	C1OIEN	0x0	rw	通道 1 输出立即使能（Channel 1 output immediately enable） 在 PWM 模式 A 或模式 B 下，该位能够缩短触发事件到通道 1 的输出响应间的时间。 0：需要比较 CVAL 与 C1DT 的值之后再产生输出。 1：无需比较 CVAL 与 C1DT 的值，当发生触发事件时立即产生输出。
位 1：0	C1C	0x0	rw	通道 1 配置 （Channel 1 configure） 当 C1EN='0'时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C1IN 映射在 C1IFP1 上； 10：输入，C1IN 映射在 C2IFP1 上； 11：输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

输入捕获模式：

域	简称	复位值	类型	功能
位 15：12	C2DF	0x0	rw	通道 2 滤波器 （Channel 2 digital filter）
位 11：10	C2IDIV	0x0	rw	通道 2 分频系数 （Channel 2 input divider）
位 9：8	C2C	0x0	rw	通道 2 配置（Channel 2 configure） 当 C2EN='0'时，这些位用于选择通道 2 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C2IN 映射在 C2IFP2 上； 10：输入，C2IN 映射在 C1IFP2 上； 11：输入，C2IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。
位 7：4	C1DF	0x0	rw	通道 1 滤波器 （Channel 1 digital filter） 这些位用于配置通道 1 的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器： 0000：无滤波器，以f_{DTS}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 $f_{SAMPLING}=f_{DTS}/2$，N=6

				0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=8
位 3：2	C1IDIV	0x0	rw	通道 1 分频系数 （Channel 1 input divider） 这些位定义了通道 1 的分频系数。 00：不分频，每一个有效的边沿都会产生一次输入； 01：每 2 个有效的边沿产生一次输入； 10：每 4 个有效的边沿产生一次输入； 11：每 8 个有效的边沿产生一次输入。 注：C1EN=’0’时，分频系数复位。
位 1：0	C1C	0x0	rw	通道 1 配置 （Channel 1 configure） 当 C1EN=’0’时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C1IN 映射在 C1IFP1 上； 10：输入，C1IN 映射在 C2IFP1 上； 11：输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

14.3.4.8 TMR9和TMR12通道控制寄存器（TMRx_CCTRL）

域	简称	复位值	类型	功能
位 15：8	保留	0x00	resd	保持默认值。
位 7	C2CP	0x0	rw	通道 2 互补极性 （Channel 2 complementary polarity） 见 C1P 的描述。
位 6	C2CEN	0x0	rw	通道 2 互补使能 （Channel 2 complementary enable） 见 C1EN 的描述。
位 5	C2P	0x0	rw	通道 2 极性 （Channel 2 polarity） 见 C1P 的描述。
位 4	C2EN	0x0	rw	通道 2 使能 （Channel 2 enable） 见 C1EN 的描述。
位 3	C1CP	0x0	rw	通道 1 互补极性 （Channel 1 complementary polarity） 0：C1COUT 的有效电平为高 1：C1COUT 的有效电平为低
位 2	C1CEN	0x0	rw	通道 1 互补使能 （Channel 1 complementary enable） 0：禁止输出； 1：使能输出。
位 1	C1P	0x0	rw	通道 1 极性 （Channel 1 polarity） 通道 1 配置为输出： 0：C1OUT 的有效电平为高 1：C1OUT 的有效电平为低 通道 1 配置为输入： C1CP/C1P 位共同定义输入信号有效沿。 00：C1IN 的有效边沿为上升沿；作为外部触发使用时，C1IN 不反相。 01：C1IN 的有效边沿为下降沿；作为外部触发使用时，C1IN 反相。 10：保留 11：C1IN 的有效边沿为上升沿和下降沿；作为外部触发使用时，C1IN 不反相。
位 0	C1EN	0x0	rw	通道 1 使能 （Channel 1 enable） 0：禁止输入或输出； 1：使能输入或输出。

14.3.4.9 TMR9和TMR12计数值（TMRx_CVAL）

域	简称	复位值	类型	功能
位 15：0	CVAL	0x0000	rw	计数器的值 （Counter value）

14.3.4.10 TMR9和TMR12预分频器（TMRx_DIV）

域	简称	复位值	类型	功能
位 15：0	DIV	0x0000	rw	分频系数 （Divider value） 计数器时钟频率 $f_{CK\_CNT} = f_{TMR\_CLK}$ /（DIV[15：0]+1） 溢出事件发生时该寄存器值被传送到实际的预分频寄存器中。

14.3.4.11 TMR9和TMR12周期寄存器（TMRx_PR）

域	简称	复位值	类型	功能
位 15：0	PR	0x0000	rw	周期值 （Period value） 定时器计数的周期值。当周期值为 0 时，定时器不工作。

14.3.4.12 TMR9和TMR12重复周期寄存器（TMRx_RPR）

域	简称	复位值	类型	功能
位 15：8	保留	0x00	resd	保持默认值。
位 7：0	RPR	0x00	rw	重复周期的次数（Repetition of period value） 这些位用于减慢溢出事件发生的速率，当重复周期的次数减为 0 时才会发生溢出事件。

14.3.4.13 TMR9和TMR12通道1数据寄存器（TMRx_C1DT）

域	简称	复位值	类型	功能
位 15：0	C1DT	0x0000	rw	通道 1 数据寄存器值（Channel 1 data register） 若通道 1 配置为输入： C1DT 是前一次通道 1 输入事件（C1IN）所保存的 CVAL。 若通道 1 配置为输出： C1DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C1OBEN），并根据设置在 C1OUT 上产生相应的输出。

14.3.4.14 TMR9和TMR12通道2数据寄存器（TMRx_C2DT）

域	简称	复位值	类型	功能
位 15：0	C2DT	0x0000	rw	通道 2 数据寄存器值（Channel 2 data register） 若通道 2 配置为输入： C2DT 是前一次通道 2 输入事件（C2IN）所保存的 CVAL。 若通道 2 配置为输出： C2DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C2OBEN），并根据设置在 C2OUT 上产生相应的输出。

14.3.4.15 TMR9和TMR12刹车寄存器（TMRx_BRK）

域	简称	复位值	类型	功能
位 19:16	BKF	0x0	rw	刹车输入滤波（brake input filter） 这些位用于配置刹车输入的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器： 0000：无滤波器，以f_{DTS}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 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=6

				1001：采样频率$f_{SAMPLING}$=f_{DTS}/8，N=8 1010：采样频率$f_{SAMPLING}$=f_{DTS}/16，N=5 1011：采样频率$f_{SAMPLING}$=f_{DTS}/16，N=6 1100：采样频率$f_{SAMPLING}$=f_{DTS}/16，N=8 1101：采样频率$f_{SAMPLING}$=f_{DTS}/32，N=5 1110：采样频率$f_{SAMPLING}$=f_{DTS}/32，N=6 1111：采样频率$f_{SAMPLING}$=f_{DTS}/32，N=8
位 15	OEN	0x0	rw	输出使能（Output enable） 对配置为输出的通道，该位用于使能 CxOUT 和 CxCOUT 的输出。 0：关闭； 1：开启。
位 14	AOEN	0x0	rw	输出自动使能（Automatic output enable） 用于溢出事件时将 OEN 自动置'1' 0：关闭； 1：开启
位 13	BRKV	0x0	rw	刹车输入信号的有效性（Brake input validity） 用于选择刹车输入信号的输入有效电平： 0：低电平； 1：高电平。
位 12	BRKEN	0x0	rw	刹车功能使能（Brake enable） 用于开启刹车功能。 0：关闭； 1：开启。
位 11	FCSOEN	0x0	rw	总输出开时的冻结状态（Frozen channel status when holistic output enable） 该位用于配置具有互补输出的通道，在定时器不工作且 OEN=1 时的通道状态。 0：关闭 CxOUT/CxCOUT 输出； 1：开启 CxOUT/CxCOUT 输出，输出为无效电平。
位 10	FCSODIS	0x0	rw	总输出关时的冻结状态（Frozen channel status when holistic output disable） 该位用于配置具有互补输出的通道，在定时器不工作且 OEN=0 时的通道状态。 0 ：关闭 CxOUT/CxCOUT 输出； 1 ：开启 CxOUT/CxCOUT 输出，输出为空闲电平。
位 9：8	WPC	0x0	rw	写保护配置（Write protected configuration） 该位用于配置写保护。 00：写保护关闭； 01：3 级写保护，以下位受写保护： TMRx_BRK：DTC、BRKEN、BRKV 和 AOEN TMRx_CTRL2：CxIOS 和 CxCIOS 10：2 级写保护，除 3 级写保护的内容外，以下位也受写保护： TMRx_CCTRL：CxP 和 CxCP TMRx_BRK：FCSODIS 和 FCSOEN 11：1 级写保护，除 2 级写保护的内容外，以下位也受写保护： TMRx_CMx：C2OCTRL 和 C2OBEN *注：WPC>0 时将无法再次被修改，直到系统复位。*
位 7：0	DTC	0x00	rw	死区配置 （Dead-time configuration） 这些位用于配置死区时间。取 DTC[7：0]的高 3 位为功能选择位： 0xx：DT = DTC [7：0] * TDTS； 10x：DT =（64+ DTC [5：0]）* TDTS * 2； 110：DT =（32+ DTC [4：0]）* TDTS * 8； 111：DT =（32+ DTC [4：0]）* TDTS * 16；

14.3.4.16 TMR9和TMR12 DMA控制寄存器（TMRx_DMACTRL）

域	简称	复位值	类型	功能
位 15：13	保留	0x0	resd	保持默认值。
位 12：8	DTB	0x00	rw	DMA 传输字节（DMA transfer bytes） 这些位定义了传输的字节个数： 00000：1 个字节　　00001：2 个字节 00010：3 个字节　　00011：4 个字节 .....　　..... 10000：17 个字节　　10001：18 个字节
位 7：5	保留	0x0	resd	保持默认值。
位 4：0	ADDR	0x00	rw	DMA 传输地址偏移 （DMA transfer address offset） ADDR 定义了从 TMRx_CTRL1 所在地址开始的偏移量： 00000：TMRx_CTRL1， 00001：TMRx_CTRL2， 00010：TMRx_STCTRL， .....

14.3.4.17 TMR9和TMR12 DMA数据寄存器（TMRx_DMADT）

域	简称	复位值	类型	功能
位 15：0	DMADT	0x0000	rw	DMA 传输的数据寄存器（DMA data register） 通过对 DMADT 寄存器的读写能够实现对任意 TMR 寄存器的操作，其操作的寄存器地址范围是： TMRx 外设地址 + ADDR*4 至 TMRx 外设地址 + ADDR*4 + DTB*4。

14.4通用定时器（TMR10/11/13/14）

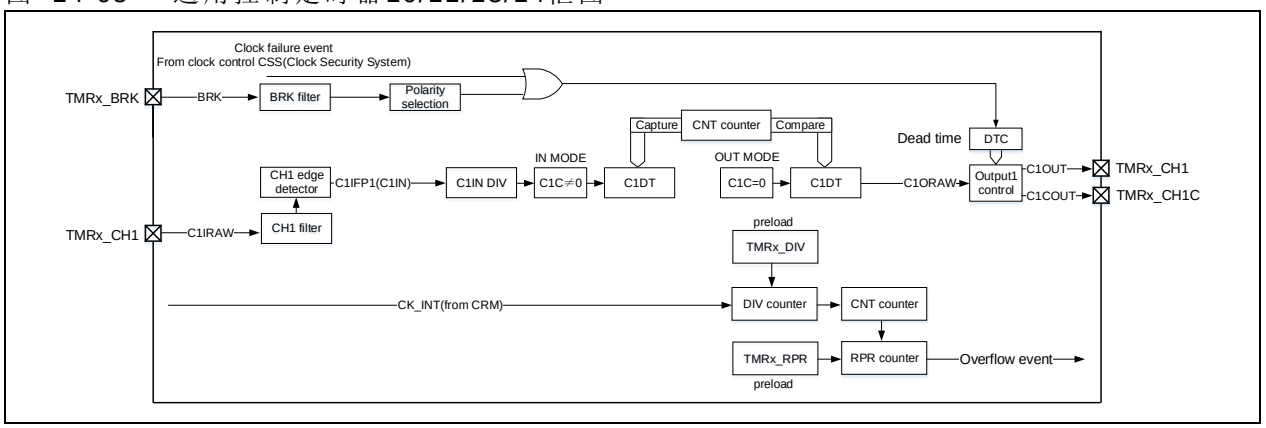
14.4.1 TMRx简介

通用定时器 TMR10/11/13/14 包含一个支持向上计数的 16 位计数器、1 个捕获/比较寄存器、1 组独立的通道。可实现嵌入死区、输入捕获、可编程 PWM 输出。

14.4.2 TMRx主要特性

- 16 位支持向上计数器、8 位重复计数计数器
- 1 组独立通道，支持输入捕获、输出比较、PWM 生成、单周期模式、死区插入。
- 1 组支持互补输出的独立通道
- 支持 TMR 刹车功能
- 定时器之间可互联同步
- 支持溢出事件、刹车输入、通道事件触发中断/DMA
- 支持 TMR burst DMA 传输

图 14-65 通用控制定时器10/11/13/14框图

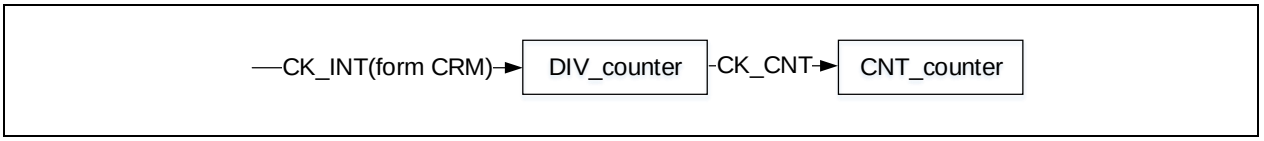


14.4.3 TMRx功能描述

14.4.3.1 计数时钟

TMR10/11/13/14 计数时钟仅能从内部时钟（CK_INT）时钟源提供。

图 14-66 计数时钟



内部时钟（CK_INT）

默认下使用 CK_INT 经由预分频器驱动计数器计数，当 TMR 对应的 APB 时钟预分频系数是 1 时，CK_INT 频率等于 APB 时钟频率，否则 CK_INT 频率等于 APB 时钟频率的 2 倍。相关配置流程如下：

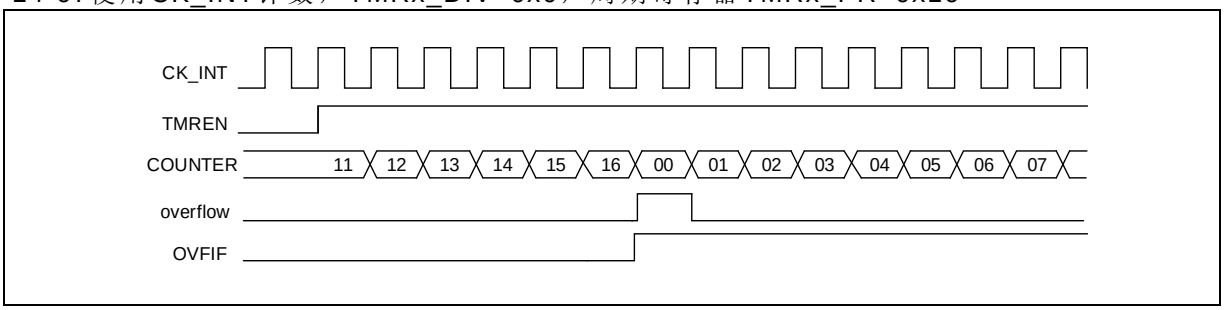
-配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]，选择计数模式，若选择单向对齐计数模式，还需配置 TMRx_CTRL1 寄存器 OWCDIR 选择计数方向。

-配置 TMRx_DIV 寄存器，设置计数器计数频率。

-配置 TMRx_PR 寄存器，设置计数器计数周期。

-配置TMRx_CTRL1寄存器TMREN，使能计数器。

图 14-67使用CK_INT计数，TMRx_DIV=0x0，周期寄存器TMRx_PR=0x16



14.4.3.2 计数模式

TMR10/11/13/14 提供了多种计数模式，用来满足不同的应用场景。其内部拥有一个支持 16 位向上计数的计数器。

TMRx_PR 寄存器用于设置计数器计数周期。默认 TMRx_PR 寄存器值会立即传入它的影子寄存器；当开启周期缓冲功能后（PRBEN 置 1），TMRx_PR 寄存器值在溢出事件发生时传入它的影子寄存器。

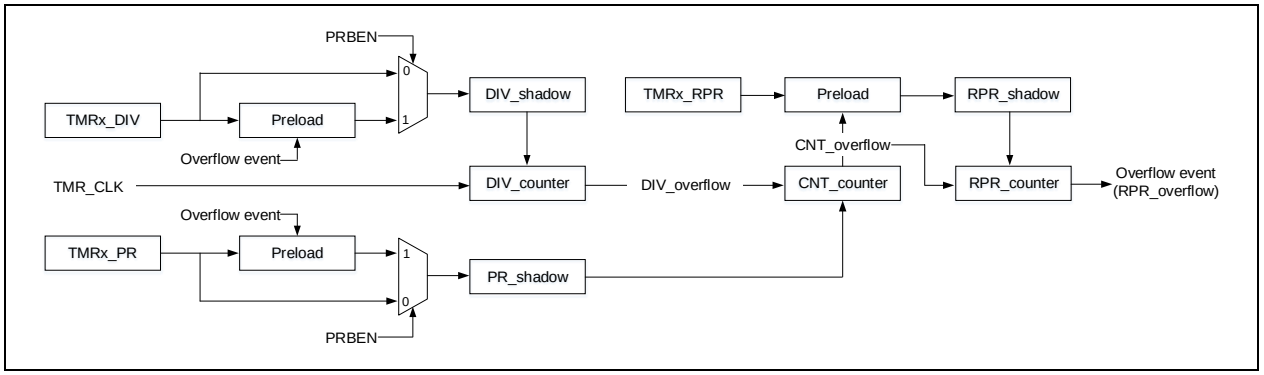
TMRx_DIV 寄存器用于设置计数器计数频率，每（DIV[15:0]+1）个计数时钟周期，计数器计数一次。和 TMRx_PR 寄存器类似，开启周期缓冲功能后，TMRx_DIV 寄存器值在溢出事件时更新至它的影子寄存器。

读取 TMRx_CNT 寄存器会返回当前计数器计数值，写入 TMRx_CNT 寄存器会更新计数器当前计数值为写入值。

默认允许产生溢出事件，设置 TMRx_CTRL1 寄存器 OVFEN=1 将禁止溢出事件产生。TMRx_CTRL1 寄存器 OVFS 用于选择溢出事件来源，默认计数器上溢或下溢、置位 OVFSWTR、复位模式次定时器控制器产生的复位信号产生溢出事件。置位 OVFS 后，只有计数器上溢或下溢产生溢出事件。

TMREN 位置 1 将使能定时器计数，由于同步逻辑，实际驱动计数器的使能信号 TMR_EN 相对于 TMREN 延迟一个时钟周期。

图 14-68 计数器基本结构



向上计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]=2'b00，OWCDIR=1'b0 开启向上计数模式，计数值达到 TMRx_PR 值时，重新从 0 向上计数，计数器上溢并产生溢出事件，同时 OVFIF 位置 1。若禁止产生溢出事件，计数器溢出后不再重载预分频值和周期值，否则预分频值和周期值在溢出事件后更新。

图 14-69 PRBEN=0时的溢出事件

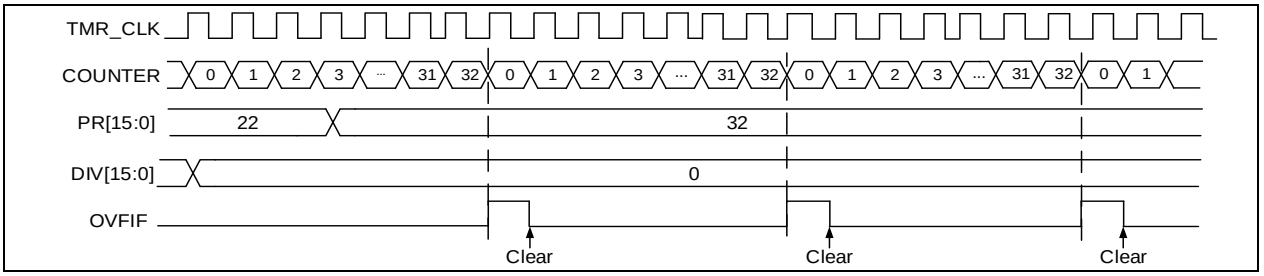
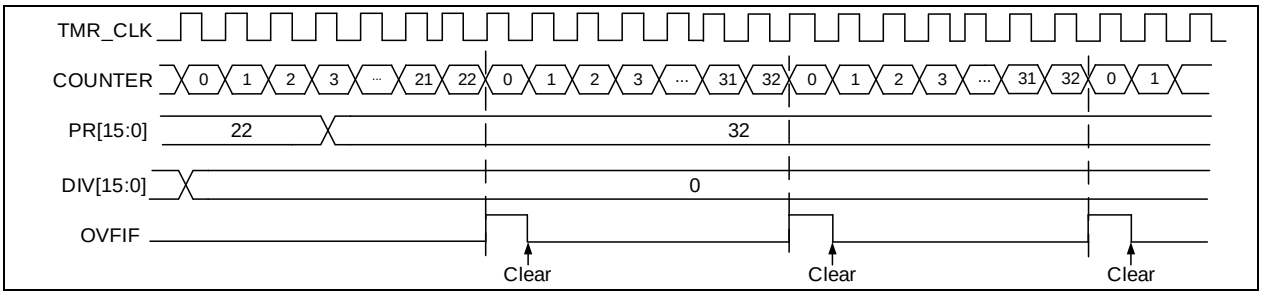


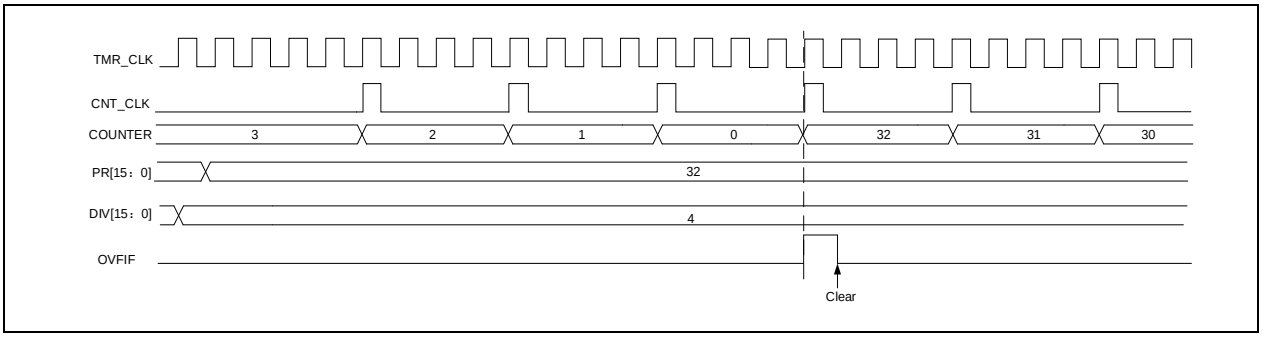
图 14-70 PRBEN=1时的溢出事件



向下计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]=2'b00，OWCDIR=1'b1 开启向下计数模式，计数值达到 0 值并重新从 TMRx_PR 向上下数时，计数器下溢并产生溢出事件。

图 14-71 计数器时序图，内部时钟分频因子为4



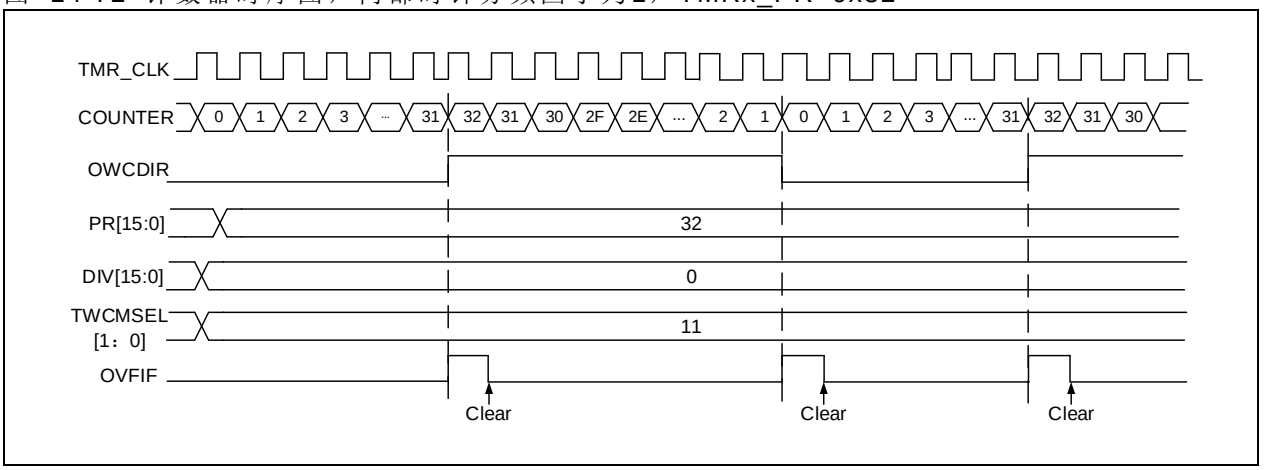
中央双向对齐计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]≠2'b00 开启中央双向对齐计数模式，中央双向对齐计数模式下计数器交替向上、向下计数。计数值从 TMRx_PR 值向下计数到 1 值，产生下溢事件，然后从 0 开始向上计数；向上计数到 TMRx_PR 值-1，产生上溢事件，之后从 TMRx_PR 值向下计数。计数器计数方向由计数器方向控制位（OWCDIR）实时查看。

TMRx_CTRL1 寄存器 TWCMSEL[1:0]位还用于选择中央双向对齐计数模式下 CxIF 标志置起方式，中央双向对齐计数模式 1（TWCMSEL[1:0]=2'b01）仅允许 CxIF 标志位在计数器向下计数时置起；双向对齐计数模式 2（TWCMSEL[1:0]=2'b10）仅允许 CxIF 标志位在计数器向上计数时置起；双向对齐计数模式 3（TWCMSEL[1:0]=2'b11）允许 CxIF 标志位在计数器向上和向下计数时置起。

注意： 中央双向对齐计数模式下，OWCDIR 位为只读位。

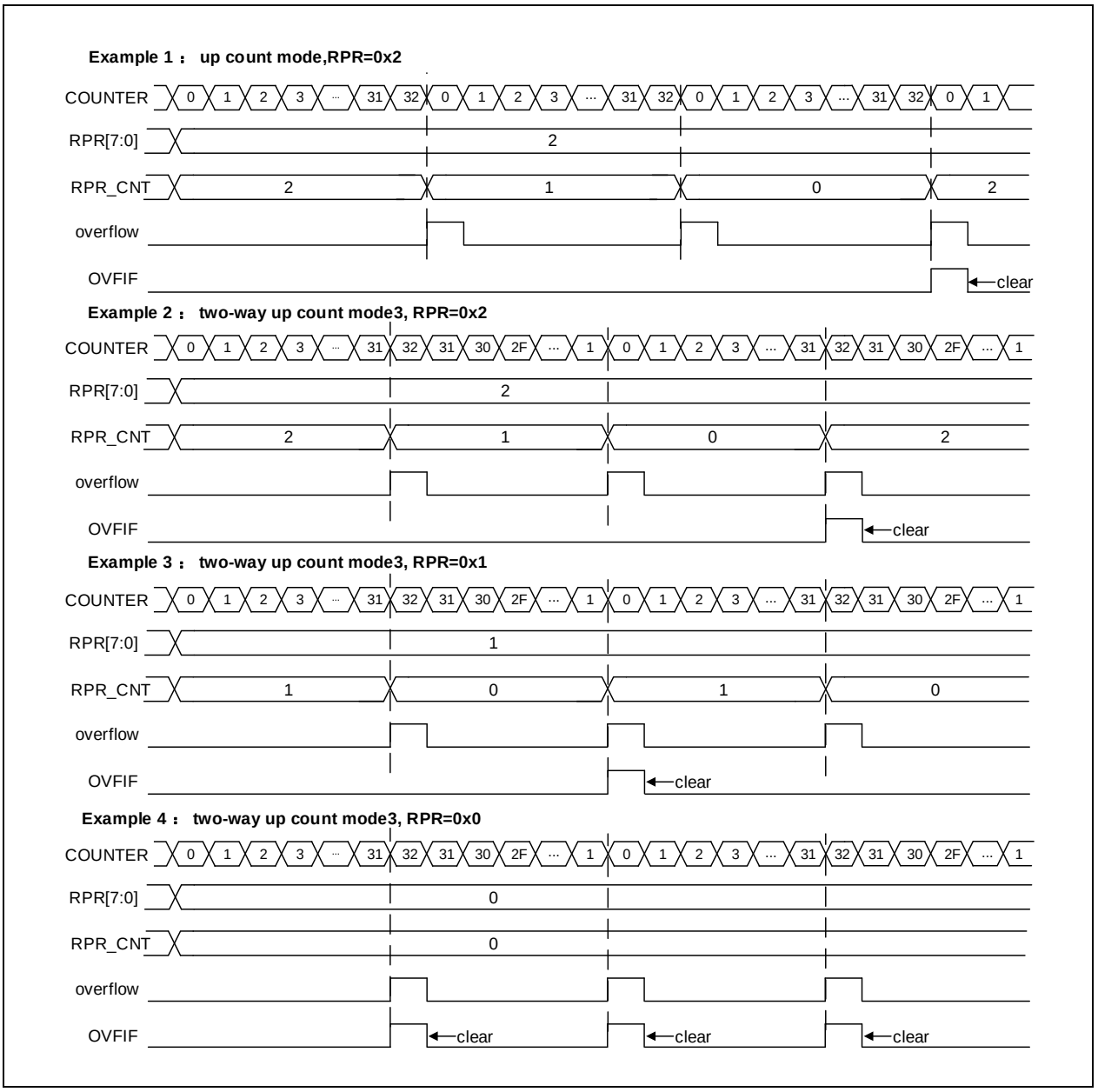
图 14-72 计数器时序图，内部时钟分频因子为1，TMRx_PR=0x32



重复计数模式：

TMRx_RPR 寄存器用于配置重复计数器计数周期，TMRx_RPR 寄存器为非 0 值时，重复计数模式启动。重复计数模式下，每（RPR[7:0]+1）次计数器溢出将产生一次溢出事件。每次计数器溢出，重复计数器递减，仅当重复计数器计数值等于 0 值时，计数器溢出会产生溢出事件。通过配置不同重复计数器值，可调整溢出事件产生的频率。

图 14-73 向上计数模式和中央双向对齐计数模式时OVFIF



14.4.3.3 TMR输入部分

TMR10/11/13/14 拥有 1 个独立通道，可配置为输入或输出，当配置位输入时，每个通道输入信号依次经过以下处理：

-TMRx_CHx 经过预处理输出 CxIRAW。配置 C1INSEL 位，选择 CxIRAW 来源是 TMRx_CHx。

-CxIRAW 输入数字滤波器，输出滤波后信号 CxIF。数字滤波器通过 CxDF 位配置采样频率和次数。

-CxIF 输入边沿检测器，输出边沿选择后信号 CxIFPx。边沿选择由 CxP 和 CxCP 位共同控制，可选择输入上升沿、下降沿或双边沿有效。

-CxIFPx 输入捕获信号选择器，输出选择后信号 CxIN。捕获信号选择器由 CxC 控制，可选择 CxIN 来源为 CxIFPx。

-CxIN 经由输入通道分频器，输出分频后信号 CxIPS。分频系数由 CxIDIV 位配置为不分频、2 分频、4 分频或 8 分频。

图 14-74 输入/输出通道1的主电路

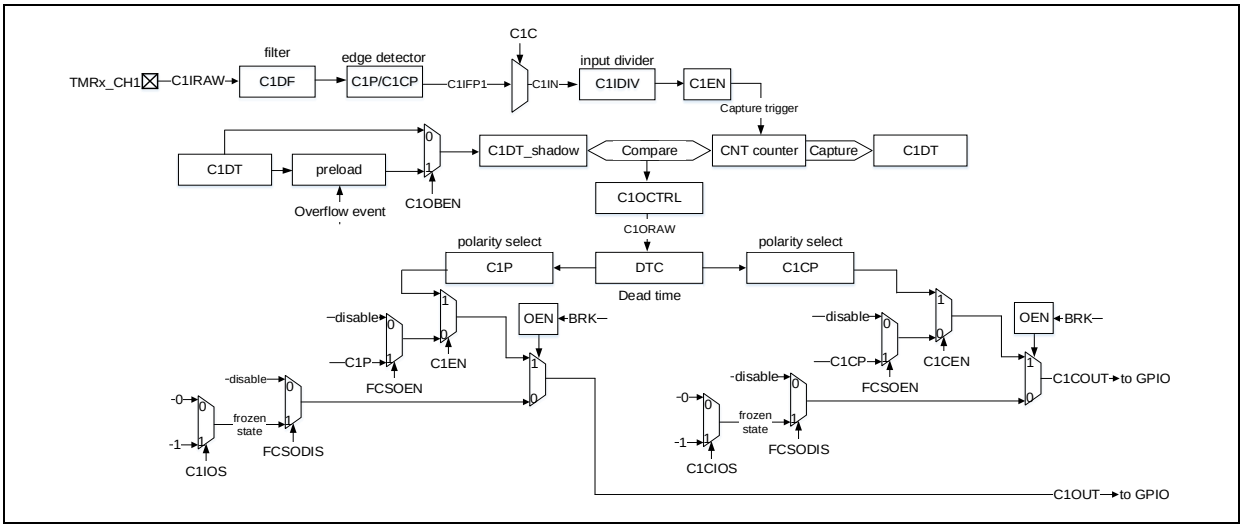
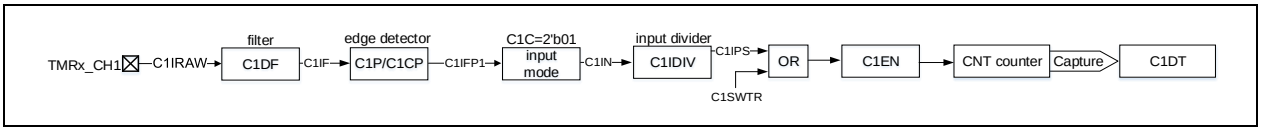


图 14-75 通道1输入部分



输入模式

此模式下，当选中的触发信号被检测到，通道寄存器（TMRx_CxDT）记录当前计数器计数值，并将捕获比较中断标志位（CxIF）置 1，若已使能通道中断（CxIEN）、通道 DMA 请求（CxDEN）则产生相应的中断和 DMA 请求。若在 CxIF 置 1 后检测到触发信号，将产生捕获溢出事件，TMRx_CxDT 会使用当前计数器计数值覆盖之前记录的计数器计数值，同时通道再捕获标志位（CxRF）置 1。

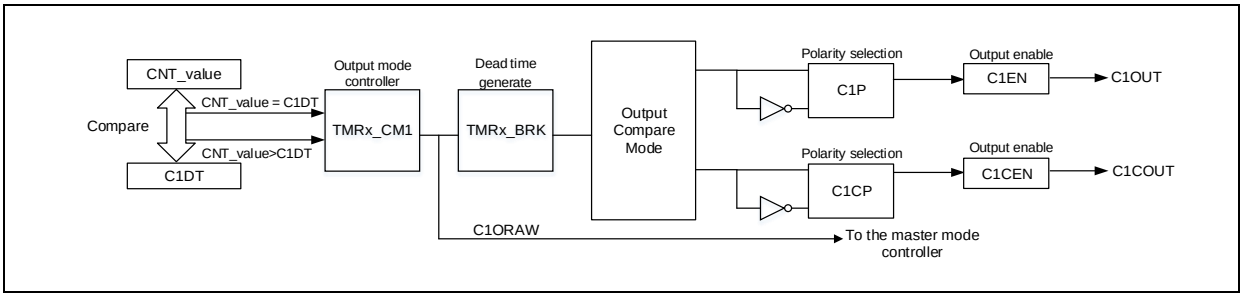
以若要捕获 C1IN 输入的上升沿，可按如下进行配置：

- 将通道模式寄存器 1（TMRx_CM1）中的 C1C 位配置为 01，选择 C1IN 作为通道 1 输入。
- 配置 C1IN 信号滤波器带宽（CxDF[3:0]）。
- 配置 C1IN 通道的有效沿，在 TMRx_CCTRL 寄存器中写入 C1P=0（上升沿）。
- 配置 C1IN 信号捕获分频（C1DIV[1:0]）。
- 使能通道 1 输入捕获（C1EN=1）。
- 根据需要设置TMRx_IDEN寄存器中的C1IEN位、TMRx_IDEN寄存器中的C1DEN位，选择中断请求或DMA请求。

14.4.3.4 TMR输出部分

TMRx 的输出部分由比较器和输出控制构成，用于编程输出信号的周期、占空比、极性。输出部分如下图所示：

图 14-76 通道1输出部分



输出模式

配置 CxC[1:0]≠2'b00 将通道配置为输出可实现多种输出模式，此时，计数器计数值将与 CxDT 寄存器值比较，并根据 CxOCTRL[2:0]位配置的输出模式，产生中间信号 CxORAW，再经过输出控制逻辑处理后输送到 IO。输出信号的周期由 TMRx_PR 寄存器值配置，占空比则由 CxDT 寄存器值配置。

输出比较模式有以下子类：

PWM 模式 A：CxOCTRL=3’b110 时，开启 PWM 模式 A。向上计数时，TMRx_C1DT>TMRx_CVAL 时 C1ORAW 输出高电平，否则为低电平；向下计数时，TMRx_C1DT<TMRx_CVAL 时 C1ORAW 输出低电平，否则为高电平。若要使用 PWM 模式 A，可按如下方式配置。

- -配置 TMRx_PR 寄存器，设置 PWM 周期。
- -配置 TMRx_CxDT 寄存器，设置 PWM 占空比。
- -配置 TMRx_CM1/CM2 寄存器 CxOCTRL 位为 3’b110，设置输出模式为 PWM 模式 A。
- -配置 TMRx_DIV 寄存器，设置计数器计数频率。
- -配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
- -配置 TMRx_CCTRL 寄存器 CxP 位、CxCP 位，设置输出极性。
- -配置 TMRx_CCTRL 寄存器 CxEN 位、CxCEN 位，使能通道输出。
- -配置 TMRx_BRK 寄存器 OEN 位，使能 TMRx 输出。
- -配置 TMR 输出通道对应 GPIO 为对应的复用模式。
- -配置 TMRx_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

PWM 模式 B：CxOCTRL=3’b111 时，开启 PWM 模式 B。向上计数时，TMRx_C1DT>TMRx_CVAL 时 C1ORAW 输出低电平，否则为高电平；向下计数时，TMRx_C1DT<TMRx_CVAL 时 C1ORAW 输出高电平，否则为低电平。

强制输出模式：CxOCTRL=3’b100/101 时，开启强制输出模式。此时，CxORAW 信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。

输出比较模式：CxOCTRL=3’b001/010/011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 强制输出高电平（CxOCTRL=3’b001）、低电平（CxOCTRL=3’b010）或进行电平翻转（CxOCTRL=3’b011）。

单周期模式：PWM 模式的特例，将 OCMEN 位置 1 可开启单周期模式，此模式下，仅在当前计数周期中进行比较匹配，完成当前计数后，TMREN 位清 0，因此仅输出一个脉冲。当配置为向上计数模式时，需要严格配置 CVAL<CxDT≤PR；向下计数时，需严格配置 CVAL>CxDT。

快速输出模式：将 CxOIEN 位置 1 可开启此功能，开启后 CxORAW 电平值不再在计数值与 CxDT 匹配时变化，而是在当前计数周期开始时，也就是说，比较结果被提前了，计数器值与 CxDT 寄存器的比较结果将会提前决定 CxORAW 的电平。

图 14-79 展示了输出比较模式（翻转）的例子，C1DT=0x3，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-80 展示了计数器向上计数与 PWM 模式 A 配合的例子，PR=0x32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-81 展示了计数器向上计数与单周期模式下 PWM 模式 B 配合的例子，计数器仅计数了一个周期，输出信号在这个周期中只输出了一个脉冲。

图 14-77 计数值与C1DT值匹配时翻转C1ORAW

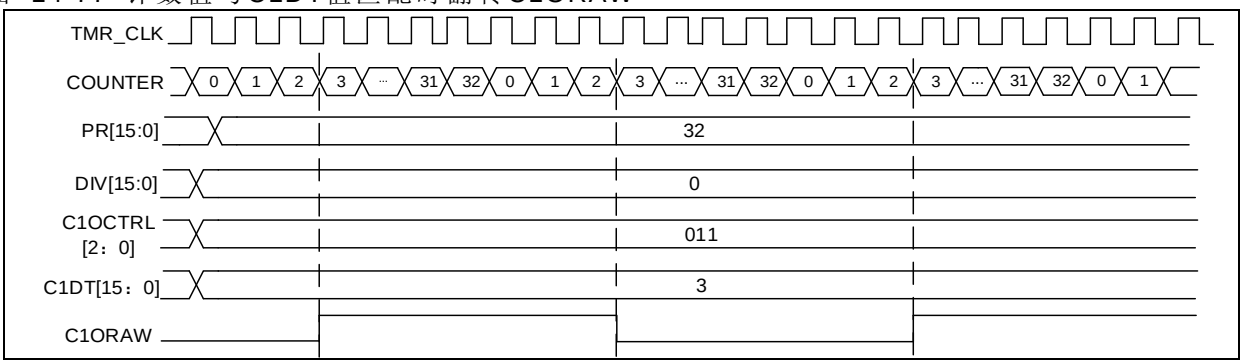


图 14-78 向上计数下PWM模式A

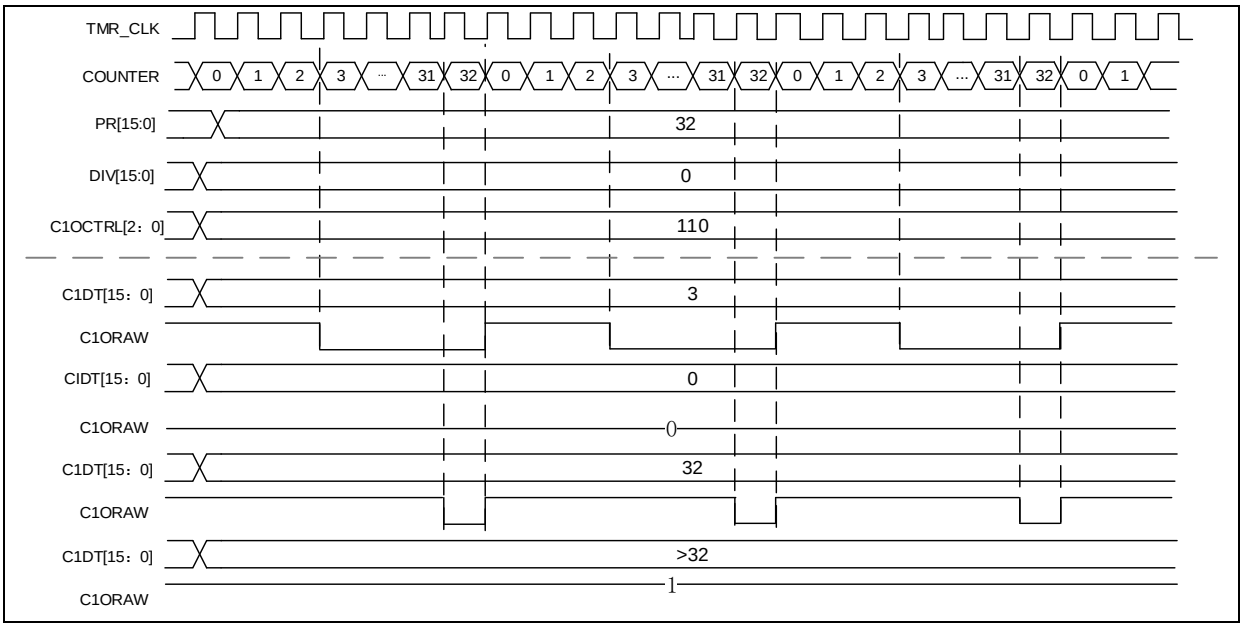
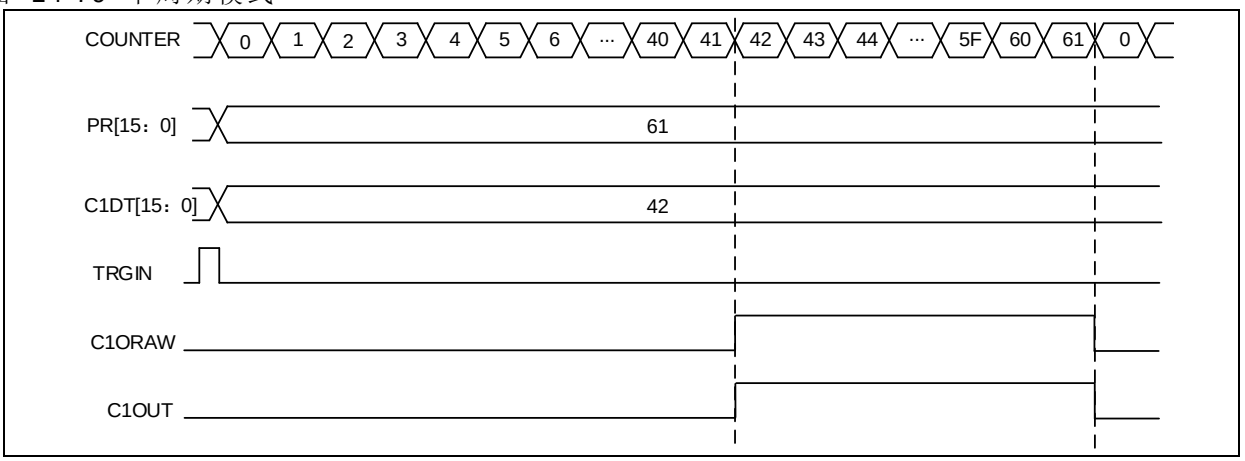


图 14-79 单周期模式



死区插入

TMRx 通道 1 包含一组反向通道输出，通过 CxCEN 使能，通过 CxCP 配置极性。CxOUT 和 CxCOUT 的输出状态见表 14-10。

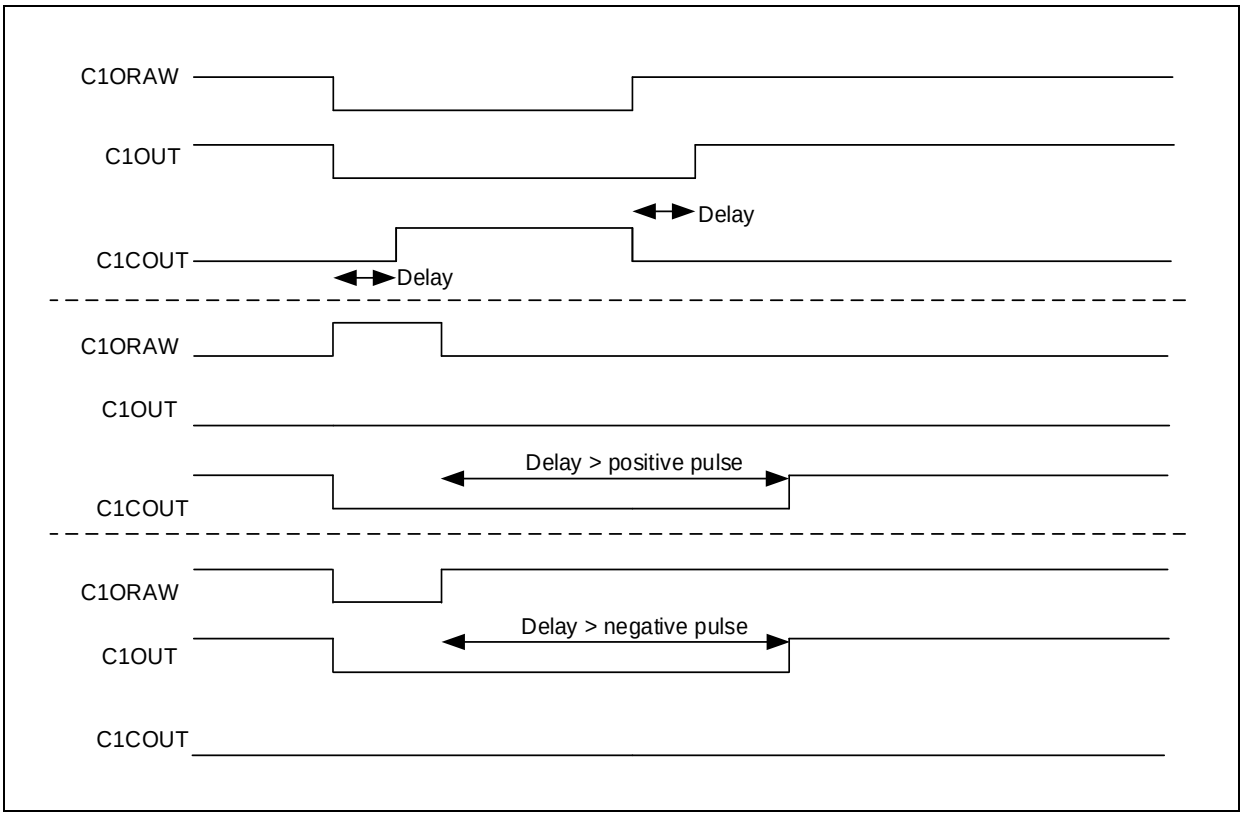
当转换为 IDLEF 状态，即 OEN 下降到 0，死区被激活。

将 CxEN 和 CxCEN 位置 1 后，通过配置 DTC[7:0]死区发生器，可插入不同时长的死区。插入死区后，CxOUT 的上升沿延迟于参考信号的上升沿；CxCOUT 的上升沿延迟于参考信号的下降沿。

如果延迟大于当前有效的输出宽度，C1OUT 和 C1COUT 不会产生相应的脉冲，死区时间应小于有效的输出宽度。

下列图显示了 CxP=0、CxCP=0、OEN=1、CxEN=1 并且 CxCEN=1 时死区插入的例子

图 14-80 带死区插入的互补输出



14.4.3.5 TMR刹车功能

开启刹车功能后（BRKEN 位置 1），CxOUT 和 CxCOUT 由 OEN、FCSODIS、FCSOEN、CxIOS 和 CxCIOS 共同控制。但 CxOUT 和 CxCOUT 输出总是不能同时处于有效电平上的。详见表 14-10 带刹车功能的互补输出通道 CxOUT 和 CxCOUT 的控制位。

刹车信号来源可以是刹车输入引脚、时钟失效事件，刹车输入信号的极性由 BRKV 位控制。

当发生刹车事件时，有下述动作：

- OEN 位异步清零，通道输出状态由 FCSODIS 位选择。关闭 MCU 的振荡器不影响该功能。
- OEN 被清零后，通道输出电平由 CxIOS 位设定。如果 FCSODIS=0，则定时器输出使能被禁止，否则输出使能始终为高。
- 当使用互补输出时：
  - 输出最开始处于复位状态，也就是无效的状态（取决于极性）。这是异步操作，定时器有无时钟并不影响此功能。
  - 定时器的时钟如果有效，会开启死区生成功能，CxIOS和CxCIOS位用来配置死区之后的电平。即使在这种情况下，CxOUT和CxCOUT也不能被同时驱动到有效的电平。

注意：由于 OEN 位同步逻辑，死区时间较通常情况会延长一段时间（大约 2 个 clk_tmr 的时钟周期）。

  - 如果FCSODIS=0，定时器释放使能输出，否则保持使能输出；或一旦CxEN与CxCEN之一变高时，使能输出变为高。
- 如果开启了刹车中断或 DMA 功能，刹车状态标志将置 1，并产生刹车中断或 DMA 请求。
- 如果将 AOEN 位置 1，在下一个溢出事件时 OEN 位被自动置 1。

注意：刹车输入电平有效时，OEN 不能被设置，状态标志 BRKIF 也不能被清除。

图 14-81 TMR输出控制

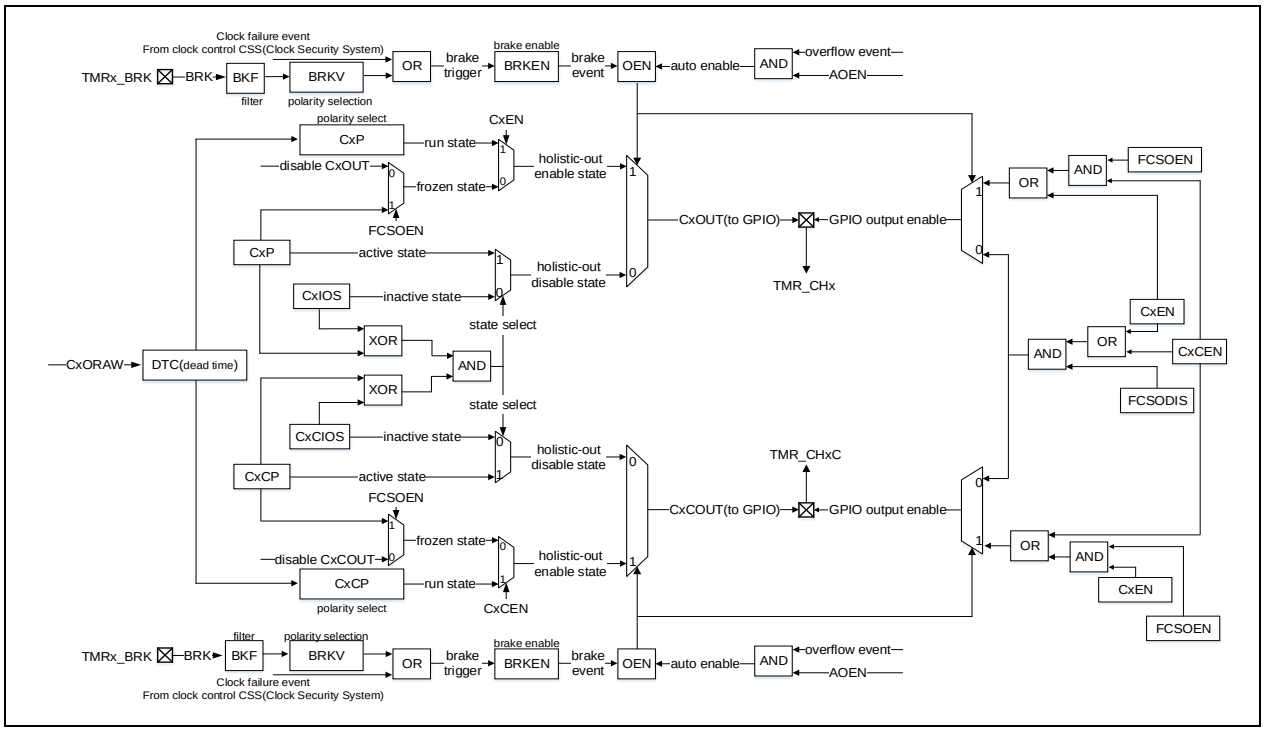
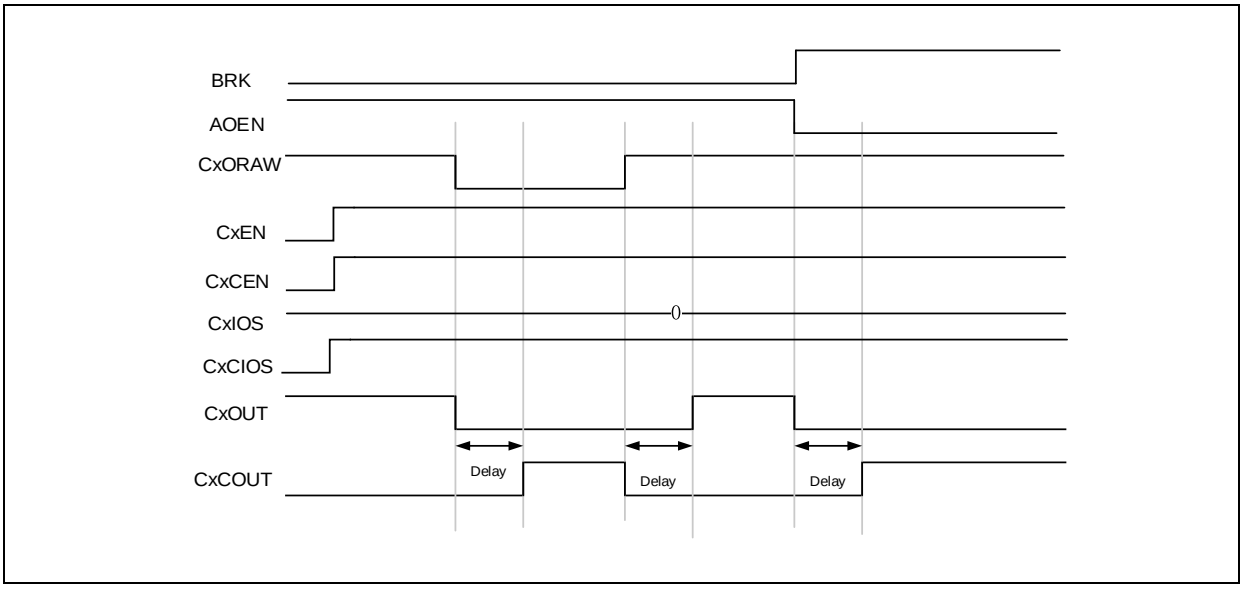


图 14-82 TMR刹车功能的例子



14.4.3.6 调试模式

当微控制器进入调试模式（Cortex®-M4F 核心停止）时，将 DEBUG 模块中的 TMRx_PAUSE 置 1，可以使 TMRx 计数器暂停计数。

14.4.4 TMRx寄存器描述

表 14-9 TMR10/11/13/14寄存器图和复位值

寄存器简称	基址偏移量	复位值
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000

TMRx_CM1	0x18	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_RPR	0x30	0x0000
TMRx_C1DT	0x34	0x0000
TMRx_BRK	0x44	0x0000
TMRx_DMACTRL	0x48	0x0000
TMRx_DMADT	0x4C	0x0000

14.4.4.1 TMRx控制寄存器1（TMRx_CTRL1）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 15：10	保留	0x00	resd	保持默认值。
位 9：8	CLKDIV	0x0	rw	时钟除频（Clock divider） 此位用于设置数字滤波器采样频率 f_{DTS} 和定时器时钟频率 f_{CK_INT} 之间的分频比，也用于调整死区时间的时基 T_{DTS} 和定时器时钟周期 T_{CK_INT} 的分频比。 00：无除频，$f_{DTS}=f_{CK\_INT}$； 01：2 除频，$f_{DTS}=f_{CK\_INT}/2$； 10：4 除频，$f_{DTS}=f_{CK\_INT}/4$； 11：保留。
位 7	PRBEN	0x0	rw	周期缓冲使能（Period buffer enable） 0：缓冲关闭； 1：缓冲开启。
位 6：5	TWCMSEL	0x0	rw	中央双向对齐计数模式选择 （Two-way count mode selection） 00：单向对齐计数模式，方向由 OWCDIR 配置； 01：中央双向对齐计数模式 1，上下交替计数，CxIF 位只在计数器向下计数时被置起； 10：中央双向对齐计数模式 2，上下交替计数，CxIF 位只在计数器向上计数时被置起； 11：中央双向对齐计数模式 3，上下交替计数，CxIF 位在计数器向上和向下计数时皆被置起。
位 4	OWCDIR	0x0	rw	单向计数方向 （One-way count direction） 0：向上； 1：向下。
位 3	OCMEN	0x0	rw	单周期使能 （One cycle mode enable） 该功能用于选择溢出事件后，计数器是否停止。 0：关闭； 1：开启。
位 2	OVFS	0x0	rw	溢出事件源选择 （Overflow event source） 配置溢出事件或 DMA 请求来源。 0：来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件； 1：只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 （Overflow event enable） 0：开启； 1：关闭。
位 0	TMREN	0x0	rw	使能定时器（TMR enable） 0：关闭； 1：开启。

14.4.4.2 TMRx控制寄存器2（TMRx_CTRL2）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 15：10	保留	0x00	resd	保持默认值。
位 9	C1CIOS	0x0	rw	通道 1 互补空闲输出状态（Channel 1 complementary idle output state） 输出关闭（OEN = 0），死区发生后： 0：C1COUT=0； 1：C1COUT=1。
位 8	C1IOS	0x0	rw	通道 1 空闲输出状态（Channel 1 idle output state） 输出关闭（HOEN = 0），死区发生后： 0：C1OUT=0。 1：C1OUT=1。
位 7：4	保留	0x0	resd	保持默认值。
位 3	DRS	0x0	rw	DMA 请求源（DMA request source） DMA 请求来源。 0：通道事件； 1：溢出事件。
位 2	CCFS	0x0	rw	通道控制位刷新选择（Channel control bit flash select） 对具有互补输出的通道，如果通道控制位有缓存时： 0：通过设置 HALL 位刷新控制位； 1：通过设置 HALL 位或 TRGIN 的上升沿刷新控制位。
位 1	保留	0x0	resd	保持默认值。
位 0	CBCTRL	0x0	rw	通道缓存控制（Channel buffer control） 对具有互补输出的通道： 0：CxEN，CxLEN 和 CxOCTRL 位无缓存； 1：CxEN，CxLEN 和 CxOCTRL 位有缓存。

14.4.4.3 TMRx DMA/中断使能寄存器（TMRx_IDEN）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 15：10	保留	0x00	resd	保持默认值。
位 9	C1DEN	0x0	rw	通道 1 的 DMA 请求使能 （Channel 1 DMA request enable） 0：关闭； 1：开启。
位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能（overflow event DMA request enable） 0：关闭； 1：开启。
位 7	BRKIE	0x0	rw	刹车中断使能（Brake interrupt enable） 0：关闭； 1：开启。
位 6	保留	0x0	resd	保持默认值。
位 5	HALLIEN	0x0	rw	HALL 中断使能（HALL interrupt enable） 0：关闭； 1：开启。
位 4：2	保留	0x0	resd	保持默认值。
位 1	C1IEN	0x0	rw	通道 1 中断使能（Channel 1 interrupt enable） 0：关闭； 1：开启。
位 0	OVFIEN	0x0	rw	溢出中断使能（Overflow interrupt enable） 0：关闭； 1：开启。

14.4.4.4 TMRx中断状态寄存器（TMRx_ISTS）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 15：10	保留	0x00	resd	保持默认值。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记 （Channel 1 recapture flag）

				C1IF 的状态已经为'1'时是否再次发生了捕获，由硬件置'1'，写'0'清除。 0：无捕获发生； 1：捕获发生。
位 8	保留	0x0	resd	保持默认值。
位 7	BRKIF	0x0	rw0c	刹车中断标记 （Brake interrupt flag） 用于标记刹车输入的电平是否有效，由硬件置'1'，写'0'清除。 0：无效； 1：有效。
位 6	保留	0x0	resd	保持默认值。
位 5	HALLIF	0x0	rw0c	HALL 中断标记 （HALL interrupt flag） 当发生触发事件时由硬件置'1'，写'0'清除。 0：无 HALL 事件发生； 1：发生 HALL 事件。 HALL 事件：CxEN、CxLEN、CxOCTRL 已被更新。
位 4：2	保留	0x0	resd	保持默认值。
位 1	C1IF	0x0	rw0c	通道 1 中断标记 （Channel 1 interrupt flag） 若通道 1 为输入模式时： 捕获事件发生时由硬件置'1'，由软件清'0'或读 TMRx_C1DT 清'0'。 0：无捕获事件发生； 1：发生捕获事件。 若通道 1 为输出模式时： 比较事件发生时由硬件置'1'，由软件清'0'。 0：无比较事件发生； 1：发生比较事件。
位 0	OVFIF	0x0	rw0c	溢出中断标记 （Overflow interrupt flag） 当溢出事件发生时由硬件置'1'，由软件清'0'。 0：无溢出事件发生； 1：发生溢出事件，若 TMRx_CTRL1 的 OVFEN=0、OVFS=0 时： – 当 TMRx_SWEVE 寄存器的 OVFG=1 时产生溢出事件； – 当计数值 CVAL 被触发事件重初始化时产生溢出事件。

14.4.4.5 TMRx软件事件寄存器（TMRx_SWEVT）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 15：8	保留	0x00	resd	保持默认值。
位 7	BRKSWTR	0x0	wo	软件触发刹车事件（Brake event triggered by software） 通过软件触发一个刹车事件。 0：无作用； 1：制造一个刹车事件。
位 6	保留	0x0	resd	保持默认值。
位 5	HALLSWTR	0x0	wo	软件触发 HALL 事件（HALL event triggered by software） 通过软件产生一个 HALL 事件。 0：无作用； 1：产生一个 HALL 事件。 注：该位只对拥有互补输出的通道有效。
位 4：2	保留	0x0	resd	保持默认值。
位 1	C1SWTR	0x0	wo	C1SWTR：软件触发通道 1 事件（Channel 1 event triggered by software） 通过软件触发一个通道 1 事件。 0：无作用； 1：制造一个通道 1 事件。
位 0	OVFSWTR	0x0	wo	软件触发溢出事件（Overflow event triggered by software） 通过软件触发一个溢出事件。 0：无作用； 1：制造一个溢出事件。

14.4.4.6 TMRx通道模式寄存器1（TMRx_CM1）（x=10/11/13/14）

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 CxC 位定义。该寄存器其它位的作用在输入和输出模式下不同。CxOx 描述了通道在输出模式下的功能，CxIx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

输出比较模式：

域	简称	复位值	类型	功能
位 15：7	保留	0x000	resd	保持默认值。
位 6：4	C1OCTRL	0x0	rw	通道 1 输出控制（Channel 1 output control） 这些位用于设置原始信号 C1ORAW 的工作状态。 000：断开。断开 C1ORAW 到 C1OUT 的输出； 001：设置 C1ORAW 为高：TMRx_CVAL=TMRx_C1DT 时。 010：设置 C1ORAW 为低：TMRx_CVAL=TMRx_C1DT 时。 011：切换 C1ORAW 电平：当 TMRx_CVAL=TMRx_C1DT 时。 100：固定 C1ORAW 为低。 101：固定 C1ORAW 为高。 110：PWM 模式 A －OWCDIR=0，若 TMRx_C1DT>TMRx_CVAL 时设置 C1ORAW 为高，否则为低； －OWCDIR=1，若 TMRx_C1DT <TMRx_CVAL 时设置 C1ORAW 为低，否则为高。 111：PWM 模式 B －OWCDIR=0，若 TMRx_C1DT >TMRx_CVAL 时设置 C1ORAW 为低，否则为高； －OWCDIR=1，若 TMRx_C1DT <TMRx_CVAL 时设置 C1ORAW 为高，否则为低。 注：除'000'外，其余配置下 C1OUT 将连接到 C1ORAW，C1OUT 的输出电平除了会根据 C1ORAW 变化外，还与 CCTRL 所配置的输出极性有关。
位 3	C1OBEN	0x0	rw	通道 1 输出缓存使能（Channel 1 output buffer enable） 0：关闭 TMRx_C1DT 的缓存功能，写入 TMRx_C1DT 的内容会立即生效。 1：启用 TMRx_C1DT 的缓存功能，写入 TMRx_C1DT 的内容将保存到缓存寄存器中，当发生溢出事件时再更新到 TMRx_C1DT 中。
位 2	C1OIEN	0x0	rw	通道 1 输出立即使能（Channel 1 output immediately enable） 在 PWM 模式 A 或模式 B 下，该位能够缩短触发事件到通道 1 的输出响应间的时间。 0：需要比较 CVAL 与 C1DT 的值之后再产生输出。 1：无需比较 CVAL 与 C1DT 的值，当发生触发事件时立即产生输出。
位 1：0	C1C	0x0	rw	通道 1 配置 （Channel 1 configure） 当 C1EN='0'时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C1IN 映射在 C1IFP1 上； 10：保留； 11：保留。

输入捕获模式：

域	简称	复位值	类型	功能
位 15：8	保留	0x00	resd	保持默认值。
位 7：4	C1DF	0x0	rw	通道 1 滤波器 （Channel 1 digital filter） 这些位用于配置通道 1 的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器： 0000：无滤波器，以f_{DTS}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 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=8
位 3：2	C1IDIV	0x0	rw	通道 1 分频系数 （Channel 1 input divider） 这些位定义了通道 1 的分频系数。 00：不分频，每一个有效的边沿都会产生一次输入； 01：每 2 个有效的边沿产生一次输入； 10：每 4 个有效的边沿产生一次输入； 11：每 8 个有效的边沿产生一次输入。 注：C1EN=’0’时，分频系数复位。
位 1：0	C1C	0x0	rw	通道 1 配置 （Channel 1 configure） 当 C1EN=’0’时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C1IN 映射在 C1IRAW 上； 10：输入，C1IN 映射在 C2IRAW 上； 11：输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

14.4.4.7 TMRx通道控制寄存器（TMRx_CCTRL）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 15：4	保留	0x000	resd	保持默认值。
位 3	C1CP	0x0	rw	通道 1 互补极性 （Channel 1 complementary polarity） 0：C1COUT 的有效电平为高 1：C1COUT 的有效电平为低
位 2	C1CEN	0x0	rw	通道 1 互补使能 （Channel 1 complementary enable） 0：禁止输出； 1：使能输出。
位 1	C1P	0x0	rw	通道 1 极性 （Channel 1 polarity） 通道 1 配置为输出： 0：C1OUT 的有效电平为高 1：C1OUT 的有效电平为低 通道 1 配置为输入： C1CP/C1P 位共同定义输入信号有效沿。 00：C1IN 的有效边沿为上升沿；作为外部触发使用时，C1IN 不反相。 01：C1IN 的有效边沿为下降沿；作为外部触发使用时，C1IN 反相。 10：保留 11：C1IN 的有效边沿为上升沿和下降沿；作为外部触发使用时，C1IN 不反相。
位 0	C1EN	0x0	rw	通道 1 使能 （Channel 1 enable）

	0：禁止输入或输出； 1：使能输入或输出。

表 14-10 带刹车功能的互补输出通道OCx和OCxN的控制位

控制位					输出状态（1）	
OEN 位	FCSODIS 位	FCSOEN 位	CxEN 位	CxCEN 位	CxOUT 输出状态	CxCOUT 输出状态
1	X	0	0	0	输出禁止 （与定时器断开） CxOUT=0，Cx_EN=0	输出禁止（与定时器断开） CxCOUT=0，CxCEN=0
		0	0	1	输出禁止 （与定时器断开） CxOUT=0，Cx_EN=0	CxORAW + 极性， CxCOUT= CxORAW xor CxCP， CxCEN=1
		0	1	0	CxORAW+极性， CxOUT= CxORAW xor CxP， Cx_EN=1	输出禁止（与定时器断开） CxCOUT=0，CxCEN=0
		0	1	1	CxORAW+极性+死区， Cx_EN=1	CxORAW 反相+极性+死区， CxCEN=1
		1	0	0	输出禁止（与定时器断开） CxOUT=CxP，Cx_EN=0	输出禁止（与定时器断开） CxCOUT=CxCP，CxCEN=0
		1	0	1	关闭状态 （输出使能且为无效电平） CxOUT=CxP，Cx_EN=1	CxORAW + 极性， CxCOUT= CxORAW xor CxCP，CxCEN=1
		1	1	0	CxORAW + 极性， CxOUT= CxORAW xor CxP， Cx_EN=1	关闭状态 （输出使能且为无效电平） CxCOUT=CxCP，CxCEN=1
		1	1	1	CxORAW+极性+死区， Cx_EN=1	CxORAW 反相+极性+死区， CxCEN=1
0	0	X	0	0	输出禁止（对应 IO 与定时器断开，IO 浮空） 异步地： CxOUT=CxP ，Cx_EN=0 ，CxCOUT=CxCP ，CxCEN=0； 若时钟存在：经过一个死区时间后 CxOUT=CxIOS，CxCOUT=CxCIOS，假设 CxIOS 与 CxCIOS 并不都对应 CxOUT 和 CxCOUT 的有效电平。	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0	CxEN=CxCEN=0 时：输出禁止（对应 IO 与定时器断开，IO 浮空）； 其它情况下：关闭状态（对应通道输出无效电平） 异步地：CxOUT =CxP，Cx_EN=1，CxCOUT=CxCP ，CxCEN=1； 若时钟存在：经过一个死区 时间后 CxOUT =C1IOS，CxCOUT=CxCIOS，假设 CxIOS 与 CxCIOS 并不都对应 CxOUT 和 CxCOUT 的有效电平。	
	1		0	1		
	1		1	0		
	1		1	1		

注意：如果一个通道的 2 个输出都没有使用（CxEN = CxCEN = 0），那么 CxIOS，CxCIOS，CxP 和 CxCP 都必须清零。

注意：引脚连接到互补的 CxOUT 和 CxCOUT 通道的外部 I/O 引脚的状态，取决于 CxOUT 和 CxCOUT 通道状态和 GPIO 以及 IOMUX 寄存器。

14.4.4.8 TMRx计数值（TMRx_CVAL）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 15：0	CVAL	0x0000	rw	计数器的值 （Counter value）

14.4.4.9 TMRx预分频器（TMRx_DIV）（x=10/11/13/14）

域	简称	复位值	类型	功能

位 15：0	DIV	0x0000	rw	分频系数 （Divider value） 计数器时钟频率 $f_{CK\_CNT} = f_{TMR\_CLK} /$（DIV[15：0]+1） 溢出事件发生时该寄存器值被传送到实际的预分频寄存器中。

14.4.4.10 TMRx周期寄存器（TMRx_PR）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 15：0	PR	0x0000	rw	周期值 （Period value） 定时器计数的周期值。当周期值为 0 时，定时器不工作。

14.4.4.11 TMRx周期寄存器（TMRx_RPR）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 15：8	保留	0x00	resd	保持默认值。
位 7：0	RPR	0x00	rw	重复周期的次数（Repetition of period value） 这些位用于减慢溢出事件发生的速率，当重复周期的次数减为 0 时才会发生溢出事件。

14.4.4.12 TMRx通道1数据寄存器（TMRx_C1DT）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 15：0	C1DT	0x0000	rw	通道 1 数据寄存器值（Channel 1 data register） 若通道 1 配置为输入： C1DT 是前一次通道 1 输入事件（C1IN）所保存的 CVAL。 若通道 1 配置为输出： C1DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C1OBEN），并根据设置在 C1OUT 上产生相应的输出。

14.4.4.13 TMRx刹车寄存器（TMRx_BRK）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 19：16	BKF	0x0	rw	刹车输入滤波（brake input filter） 这些位用于配置刹车输入的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器： 0000：无滤波器，以f_{DTS}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 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=8
位 15	OEN	0x0	rw	输出使能（Output enable） 对配置为输出的通道，该位用于使能 CxOUT 和 CxCOUT 的输出。 0：关闭； 1：开启。
位 14	AOEN	0x0	rw	输出自动使能（Automatic output enable） 用于溢出事件时将 OEN 自动置’1’ 0：关闭； 1：开启
位 13	BRKV	0x0	rw	刹车输入信号的有效性（Brake input validity） 用于选择刹车输入信号的输入有效电平：

				0：低电平； 1：高电平。
位 12	BRKEN	0x0	rw	刹车功能使能（Brake enable） 用于开启刹车功能。 0：关闭； 1：开启。
位 11	FCSOEN	0x0	rw	总输出开时的冻结状态（Frozen channel status when holistic output enable） 该位用于配置具有互补输出的通道，在定时器不工作且 OEN=1 时的通道状态。 0：关闭 CxOUT/CxCOUT 输出； 1：开启 CxOUT/CxCOUT 输出，输出为无效电平。
位 10	FCSODIS	0x0	rw	总输出关时的冻结状态（Frozen channel status when holistic output disable） 该位用于配置具有互补输出的通道，在定时器不工作且 OEN=0 时的通道状态。 0 ：关闭 CxOUT/CxCOUT 输出； 1 ：开启 CxOUT/CxCOUT 输出，输出为空闲电平。
位 9：8	WPC	0x0	rw	写保护配置（Write protected configuration） 该位用于配置写保护。 00：写保护关闭； 01：3 级写保护，以下位受写保护： TMRx_STOP：DTC、STPEN、STPV 和 HOAEN TMRx_CTRL2：CxIOS 和 CxIOSL 10：2 级写保护，除 3 级写保护的内容外，以下位也受写保护： TMRx_CCTRL：CxP 和 CxLP TMRx_STOP：FCSODIS 和 FCSOEN 11：1 级写保护，除 2 级写保护的内容外，以下位也受写保护： TMRx_CMx：C2OCTRL 和 C2OBEN 注：WPC>0 时将无法再次被修改，直到系统复位。
位 7：0	DTC	0x00	rw	死区配置 （Dead-time configuration） 这些位用于配置死区时间。取 DTC[7：0]的高 3 位为功能选择位： 0xx：DT = DTC [7：0] * TDTS； 10x：DT =（64+ DTC [5：0]）* TDTS * 2； 110：DT =（32+ DTC [4：0]）* TDTS * 8； 111：DT =（32+ DTC [4：0]）* TDTS * 16；

14.4.4.14 TMRx DMA控制寄存器（TMRx_DMACTRL）（x=10/11/13/14）

域	简称	复位值	类型	功能
位 15：13	保留	0x0	resd	保持默认值。
位 12：8	DTB	0x00	rw	DMA 传输字节（DMA transfer bytes） 这些位定义了传输的字节个数： 00000：1 个字节 00001：2 个字节 00010：3 个字节 00011：4 个字节 .... 10000：17 个字节 10001：18 个字节
位 7：5	保留	0x0	resd	保持默认值。
位 4：0	ADDR	0x00	rw	DMA 传输地址偏移 （DMA transfer address offset） ADDR 定义了从 TMRx_CTRL1 所在地址开始的偏移量： 00000：TMRx_CTRL1， 00001：TMRx_CTRL2， 00010：TMRx_STCTRL， .....

14.4.4.15 TMRx DMA数据寄存器（TMRx_DMADT）（x=10/11/13/14）

域	简称	复位值	类型	功能

位 15：0	DMADT	0x0000	rw	DMA 传输的数据寄存器（DMA data register） 通过对 DMADT 寄存器的读写能够实现对任意 TMR 寄存器的操作，其操作的寄存器地址范围是：TMRx 外设地址 + ADDR*4 至 TMRx 外设地址 + ADDR*4 + DTB*4。

14.4.4.16 TMR14通道输入重映射寄存器（TMR14_RMP）

	域简称	复位值	类型	功能
位 15：8	保留	0x00	resd	保持默认值。
位 7：6	TMR14_CH1_IRMP	0x0	rw	TMR14 通道 1 输入重映射（TMR14 channel 1 input remap） 00：TMR14 通道 1 输入与 GPIO 相连接 01：ERTC_CLK 10：ERTC 32 分频后 HEXT 11：CLK_OUT
位 5：0		0x00	resd	保持默认值

14.5高级控制定时器（TMR1）

14.5.1 TMR1简介

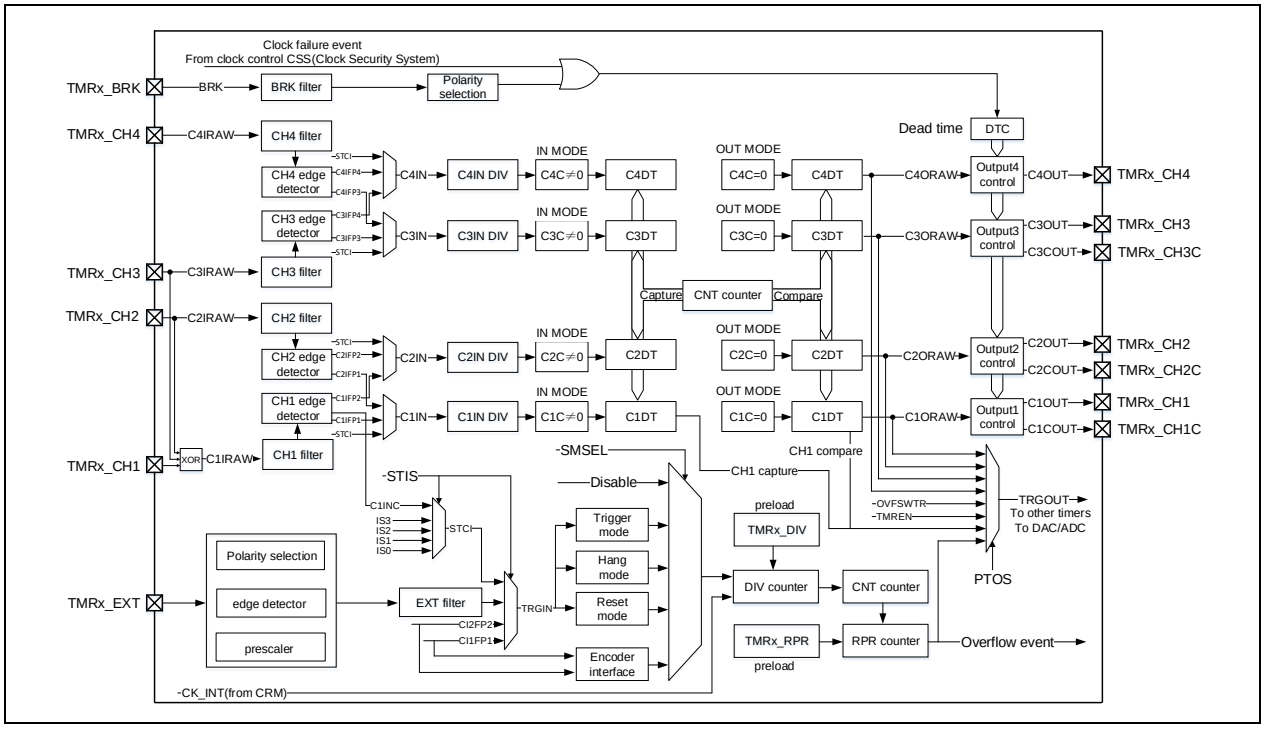
高级定时器 TMR1 包含一个支持向上、向下、双向计数的 16 位计数器、4 个通道寄存器、4 组独立的通道。可实现嵌入死区、输入捕获、可编程 PWM 输出。

14.5.2 TMR1主要特性

TMR1 定时器的功能包括：

- 可选内部、外部、内部触发输入用作计数时钟
- 16 位支持向上、向下、双向、重复计数、编码器模式的计数器
- 4 组独立通道，支持输入捕获、输出比较、PWM 生成、单周期模式、死区插入。
- 3 组支持互补输出的独立通道
- 支持 TMR 停止功能
- 定时器之间可互联同步
- 支持溢出事件、触发事件、停止输入、通道事件触发中断/DMA
- 支持 TMR burst DMA 传输

图 14-83 高级控制定时器框图

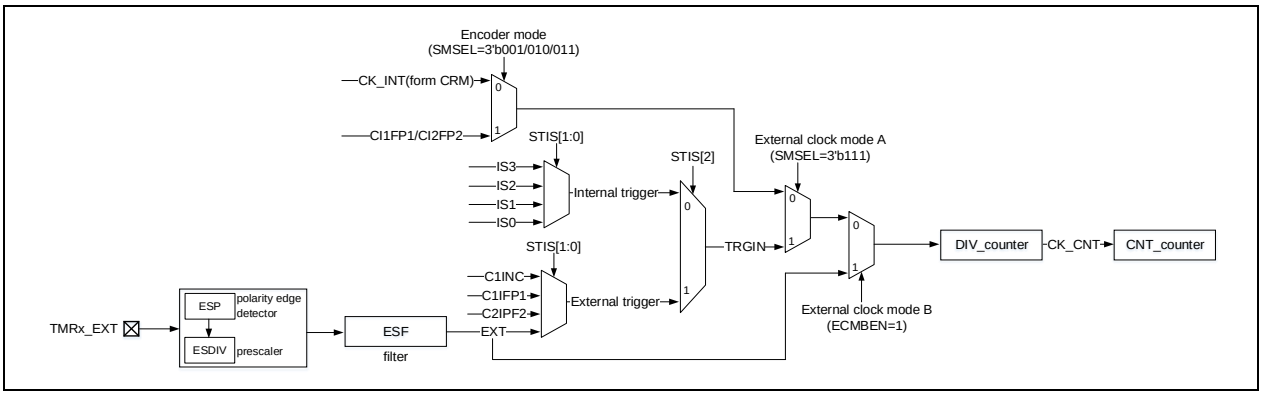


14.5.3 TMR1功能描述

14.5.3.1 计数时钟

TMR1 计数时钟可从内部时钟（CK_INT）、外部时钟（外部时钟模式 A、B）、内部触发输入（ISx）这些时钟源提供。

图 14-84 计数时钟



内部时钟（CK_INT）

默认下使用 CK_INT 经由预分频器驱动计数器计数，当 TMR 对应的 APB 时钟预分频系数是 1 时，CK_INT 频率等于 APB 时钟频率，否则 CK_INT 频率等于 APB 时钟频率的 2 倍。相关配置流程如下：

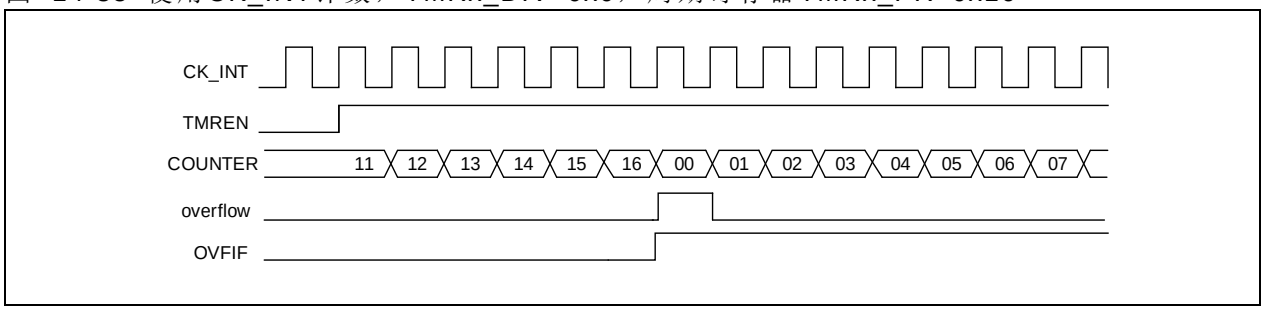
-配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]，选择计数模式，若选择单向对齐计数模式，还需配置 TMRx_CTRL1 寄存器 OWCDIR 选择计数方向。

-配置 TMRx_DIV 寄存器，设置计数器计数频率。

-配置 TMRx_PR 寄存器，设置计数器计数周期。

-配置TMRx_CTRL1寄存器TMREN，使能计数器。

图 14-85 使用CK_INT计数，TMRx_DIV=0x0，周期寄存器TMRx_PR=0x16



外部时钟（TRGIN/EXT）

计数时钟可由两种外部时钟源提供，分别为 TRGIN 和 EXT 信号。

当 SMSEL=3'b111 时，外部时钟模式 A 被选中，配置 STIS[2：0]来选择外部时钟源 TRGIN 信号驱动计数器计数。外部时钟源 TRGIN 可选则 C1INC（STIS=3'b100，通道 1 上升沿和下降沿信号）、C1IFP1（STIS=3'b101，通道 1 滤波且极性选择后信号）、C2IFP2（STIS=3'b110，通道 2 滤波且极性选择后信号）和 EXT（STIS=3'b111，外部输入经极性选择、分频和滤波后信号）。

当 ECMBEN=1 时，外部时钟模式 B 被选中，计数器由外部输入经极性选择、分频和滤波后 EXT 信号驱动计数。外部时钟模式 B 等效于外部时钟模式 A 选择 EXT 信号作为外部时钟源 TRGIN。

若要使用外部时钟模式 A，可按如下步骤配置：

-配置外部时钟源 TRGIN 参数。

若选择 TRGIN 来源为 TMRx_CH1，需配置通道 1 输入滤波（TMRx_CM1 寄存器 C1DF[3:0]）和通道 1 输入极性（TMRx_CCTRL 寄存器 C1P/C1CP）。

若选择 TRGIN 来源为 TMRx_CH2，需配置通道 2 输入滤波（TMRx_CM1 寄存器 C2DF[3:0]）和通道 1 输入极性（TMRx_CCTRL 寄存器 C2P/C2CP）。

若选择 TRGIN 来源为 TMRx_EXT，需配置外部信号极性（TMRx_STCTRL 寄存器 ESP）、外部信号分频（TMRx_STCTRL 寄存器 ESDIV[1:0]）和外部信号滤波（TMRx_STCTRL 寄存器 ESF[3:0]）。

-配置 TMRx_STCTRL 寄存器 STIS[1:0]，设置 TRGIN 信号来源。

-配置 TMRx_STCTRL 寄存器 SMSEL=3'b111，使能外部时钟模式 A。

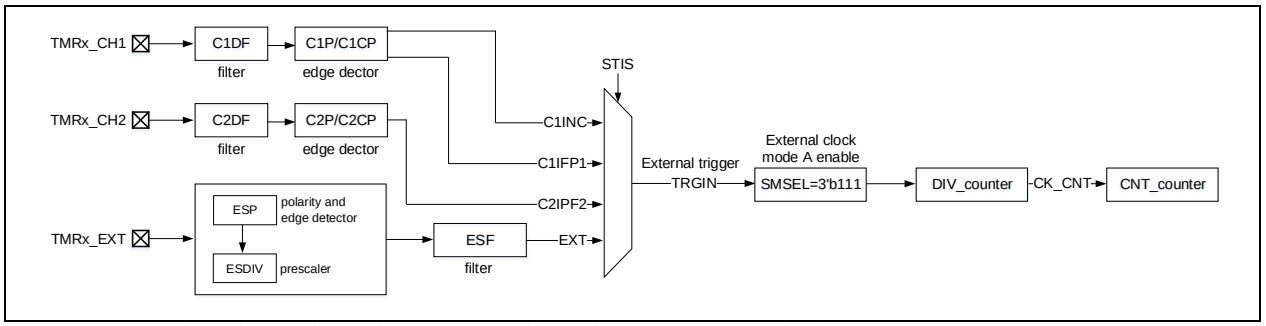
-配置 TMRx_DIV 寄存器 DIV[15:0]，设置计数器计数频率。

-配置 TMRx_PR 寄存器 PR[15:0]，设置计数器计数周期。

-配置 TMRx_CTRL1 寄存器 TMREN，使能计数器。

若要使用外部时钟模式 B，可按如下步骤配置：
-配置 TMRx_STCTRL 寄存器 ESP，设置外部信号极性。
-配置 TMRx_STCTRL 寄存器 ESDIV[1:0]，设置外部信号分频。
-配置 TMRx_STCTRL 寄存器 ESF[3:0]，设置外部信号滤波。
-配置 TMRx_STCTRL 寄存器 ECMBEN，使能外部时钟模式 B。
-配置 TMRx_DIV 寄存器 DIV[15:0]，设置计数器计数频率。
-配置 TMRx_PR 寄存器 PR[15:0]，设置计数器计数周期。
-配置 TMRx_CTRL1 寄存器 TMREN，使能计数器。

图 14-86 外部时钟模式A框图



注：由于同步逻辑，输入端信号与计数器实际时钟之间存在一定延时。

图 14-87 使用外部时钟模式A计数，PR=0x32，DIV=0x0

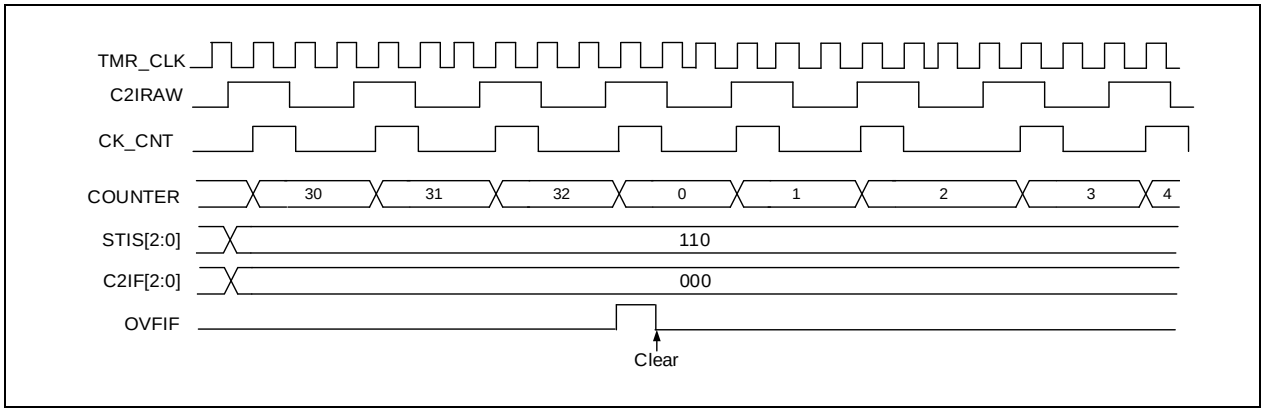
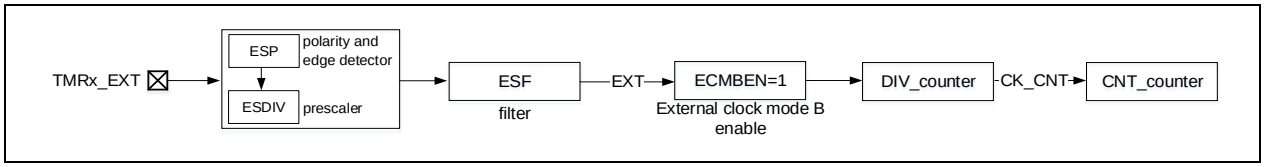
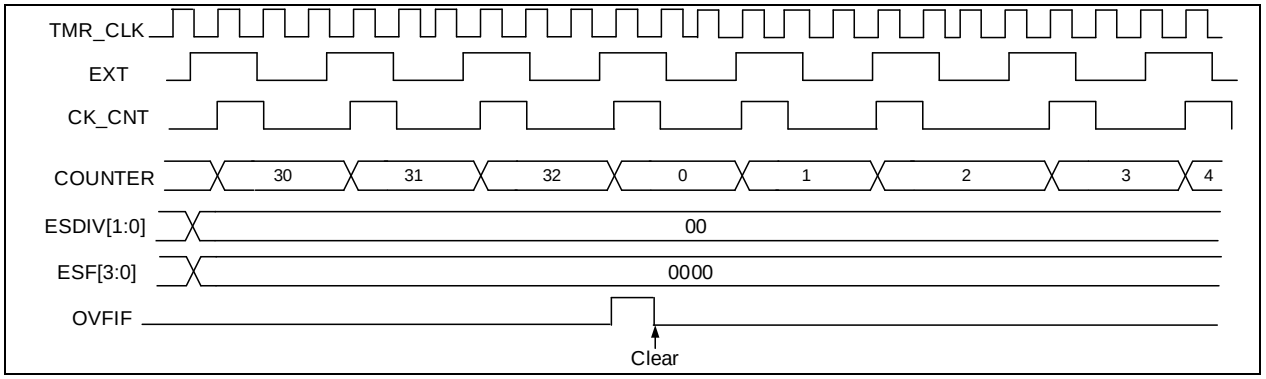


图 14-88 外部时钟模式B框图



注：由于同步逻辑。输入端 EXT 信号与计数器实际时钟之间存在一定延时。

图 14-89 使用外部时钟模式B计数，PR=0x32，DIV=0x0



内部触发输入（ISx）

定时器之间支持互联同步，因此一个定时器的 TMR_CLK 可由另一个定时器输出信号 TRGOUT 提供。配

置 STIS[2：0]选择内部触发信号驱动计数器计数。
高级定时器内含一个 16 位预分频器，用于产生驱动计数器计数的时钟 CK_CNT，通过配置 TMR1_DIV 寄存器值，可灵活调整 CK_CNT 与 TMR_CLK 之间的分频关系。预分频值可在任何时刻修改，但只在下一个溢出事件发生时，新值才会生效。
内部触发输入配置流程如下：

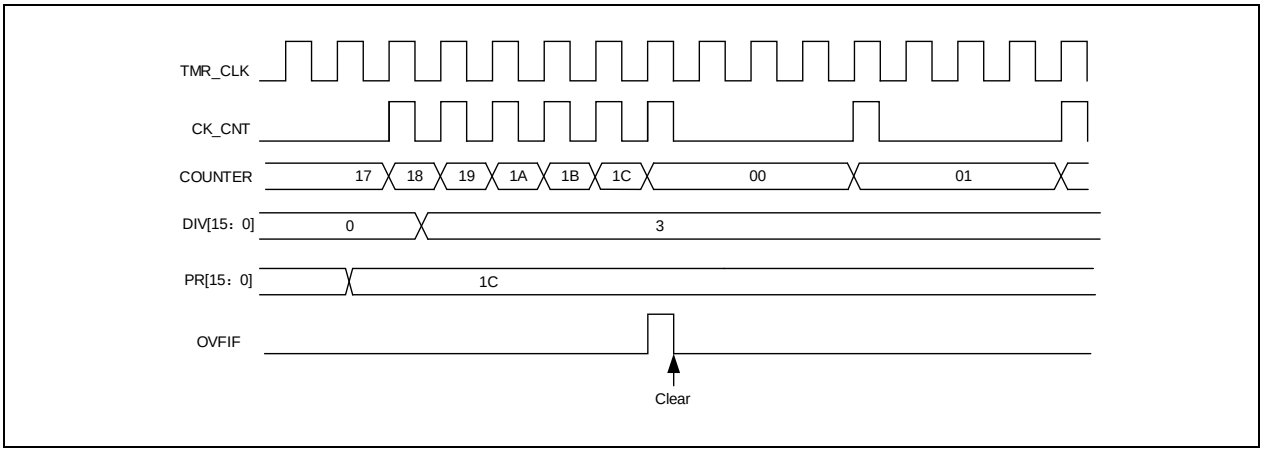
- -配置 TMRx_PR 寄存器，设置计数器计数周期。
- -配置 TMRx_DIV 寄存器，设置计数器计数频率。
- -配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
- -配置 TMRx_STCTRL 寄存器 STIS[2:0]位范围为 3’b000~3’b011，选择内部触发。
- -配置 TMRx_STCTRL 寄存器 SMSEL[2:0]=3’b111，选择外部时钟模式 A。

-配置 TMRx_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

表 14-11 TMR1内部触发连接

次定时器	IS0 （STIS=000）	IS1 （STIS=001）	IS2 （STIS=010）	IS3 （STIS=011）
TMR1	TMR9	TMR2	TMR3	TMR4

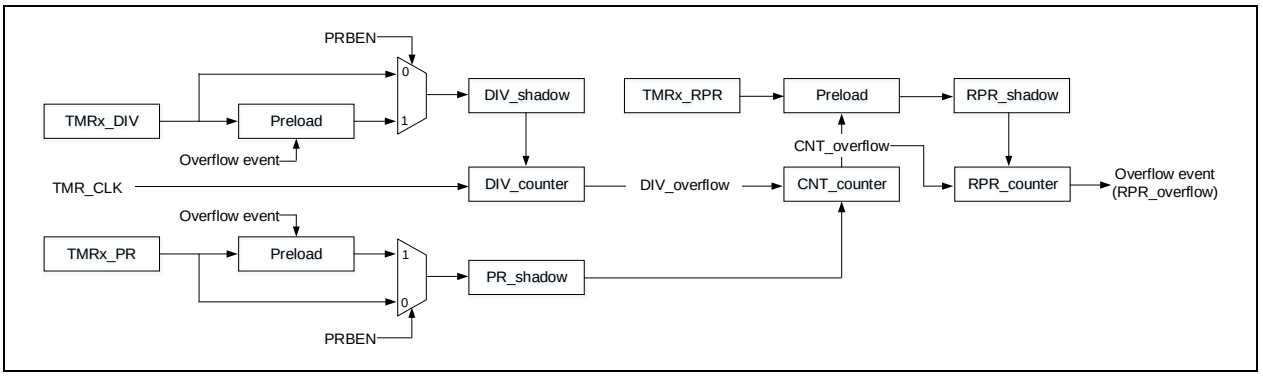
图 14-90 当预分频器的参数从1变到4时，计数器的时序图



14.5.3.2 计数模式

高级定时器提供了多种计数模式，用来满足不同的应用场景。其内部拥有一个支持 16 位向上计、向下、双向计数的计数器。
TMRx_PR 寄存器用于设置计数器计数周期。默认 TMRx_PR 寄存器值会立即传入它的影子寄存器；当开启周期缓冲功能后（PRBEN 置 1），TMRx_PR 寄存器值在溢出事件发生时传入它的影子寄存器。
TMRx_DIV 寄存器用于设置计数器计数频率，每（DIV[15:0]+1）个计数时钟周期，计数器计数一次。和 TMRx_PR 寄存器类似，开启周期缓冲功能后，TMRx_DIV 寄存器值在溢出事件时更新至它的影子寄存器。
读取 TMRx_CNT 寄存器会返回当前计数器计数值，写入 TMRx_CNT 寄存器会更新计数器当前计数值为写入值。
默认允许产生溢出事件，设置 TMRx_CTRL1 寄存器 OVFEN=1 将禁止溢出事件产生。TMRx_CTRL1 寄存器 OVFS 用于选择溢出事件来源，默认计数器上溢或下溢、置位 OVFSWTR、复位模式次定时器控制器产生的复位信号产生溢出事件。置位 OVFS 后，只有计数器上溢或下溢产生溢出事件。
TMREN 位置 1 将使能定时器计数，由于同步逻辑，实际驱动计数器的使能信号 TMR_EN 相对于 TMREN 延迟一个时钟周期。

图 14-91 计数器基本结构



向上计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]=2'b00，OWCDIR=1'b0 开启向上计数模式，计数值达到 TMRx_PR 值时，重新从 0 向上计数，计数器上溢并产生溢出事件，同时 OVFIF 位置 1。若禁止产生溢出事件，计数器溢出后不再重载预分频值和周期值，否则预分频值和周期值在溢出事件后更新。

图 14-92 PRBEN=0时的溢出事件

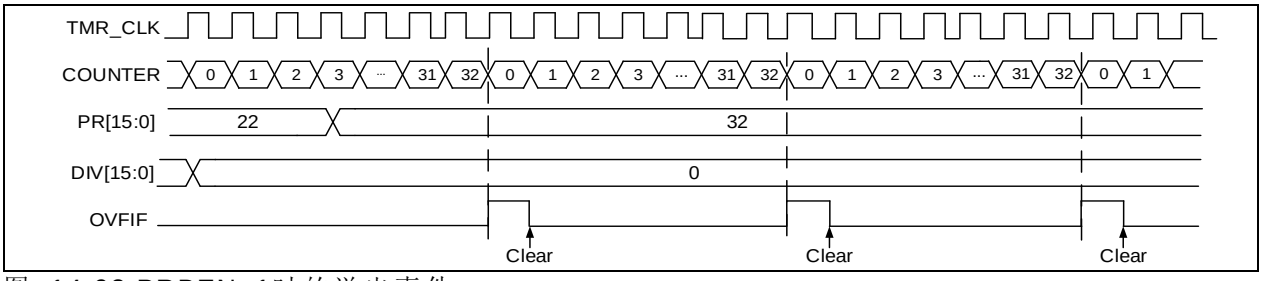
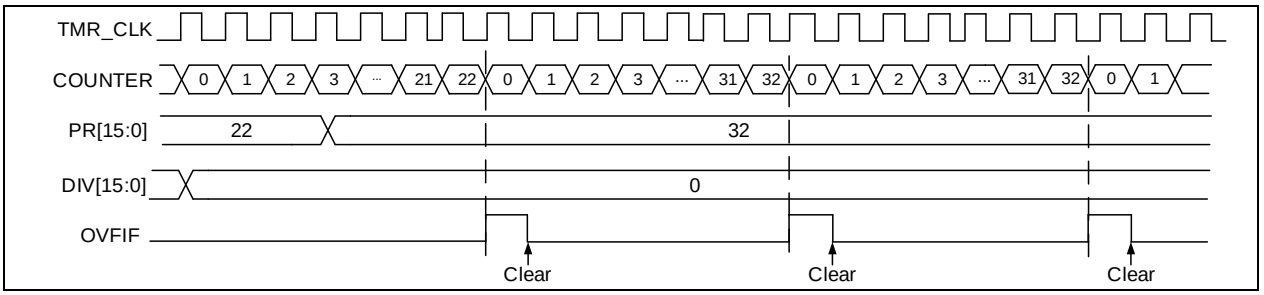


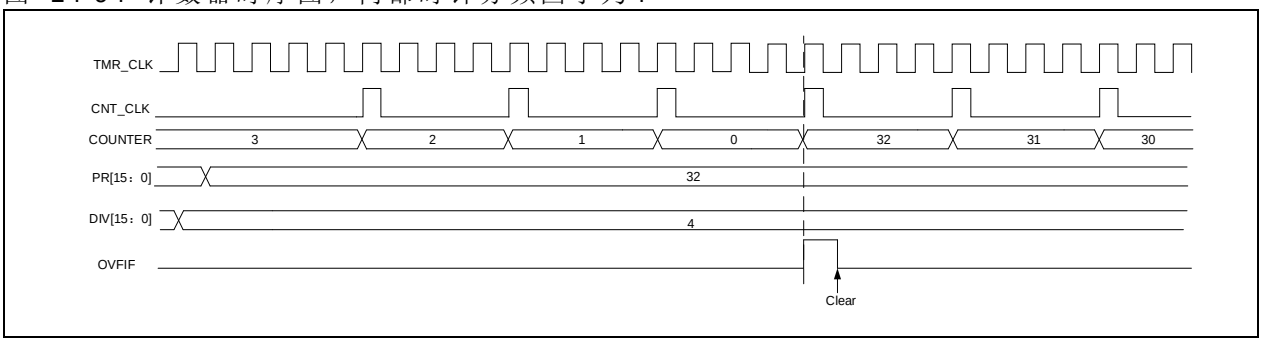
图 14-93 PRBEN=1时的溢出事件



向下计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]=2'b00，OWCDIR=1'b1 开启向下计数模式，计数值达到 0 值并重新从 TMRx_PR 向上下数时，计数器下溢并产生溢出事件。

图 14-94 计数器时序图，内部时钟分频因子为4



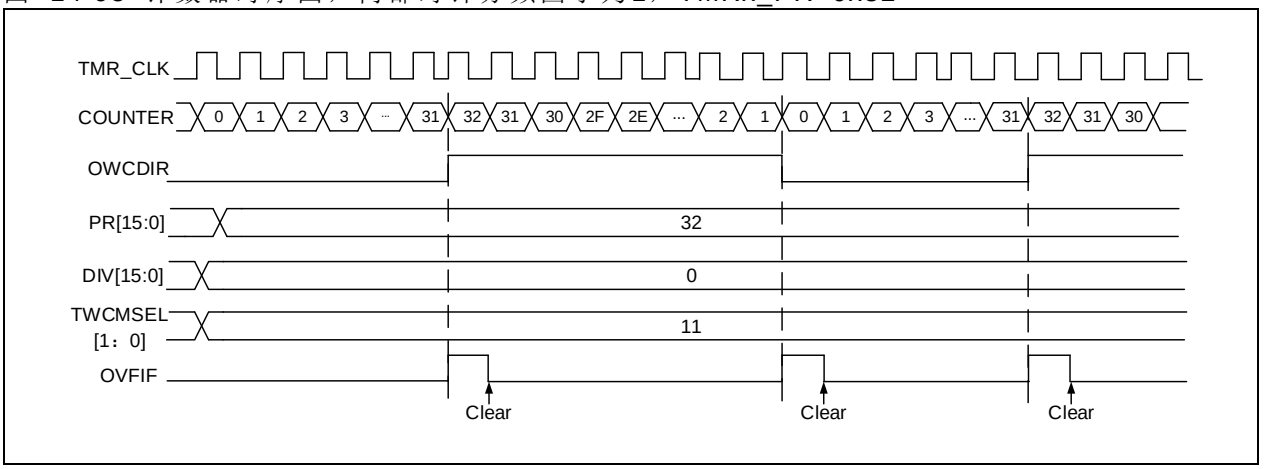
中央双向对齐计数模式

配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]≠2'b00 开启中央双向对齐计数模式，中央双向对齐计数模式下计数器交替向上、向下计数。计数值从 TMRx_PR 值向下计数到 1 值，产生下溢事件，然后从 0 开始向上计数；向上计数到 TMRx_PR 值-1，产生上溢事件，之后从 TMRx_PR 值向下计数。计数器计数方向由计数器方向控制位（OWCDIR）实时查看。

TMRx_CTRL1 寄存器 TWCMSEL[1:0]位还用于选择中央双向对齐计数模式下 CxIF 标志置起方式，中央双向对齐计数模式 1（TWCMSEL[1:0]=2'b01）仅允许 CxIF 标志位在计数器向下计数时置起；双向对齐计数模式 2（TWCMSEL[1:0]=2'b10）仅允许 CxIF 标志位在计数器向上计数时置起；双向对齐计数模式 3（TWCMSEL[1:0]=2'b11）允许 CxIF 标志位在计数器向上和向下计数时置起。

注意： 中央双向对齐计数模式下，OWCDIR 位为只读位。

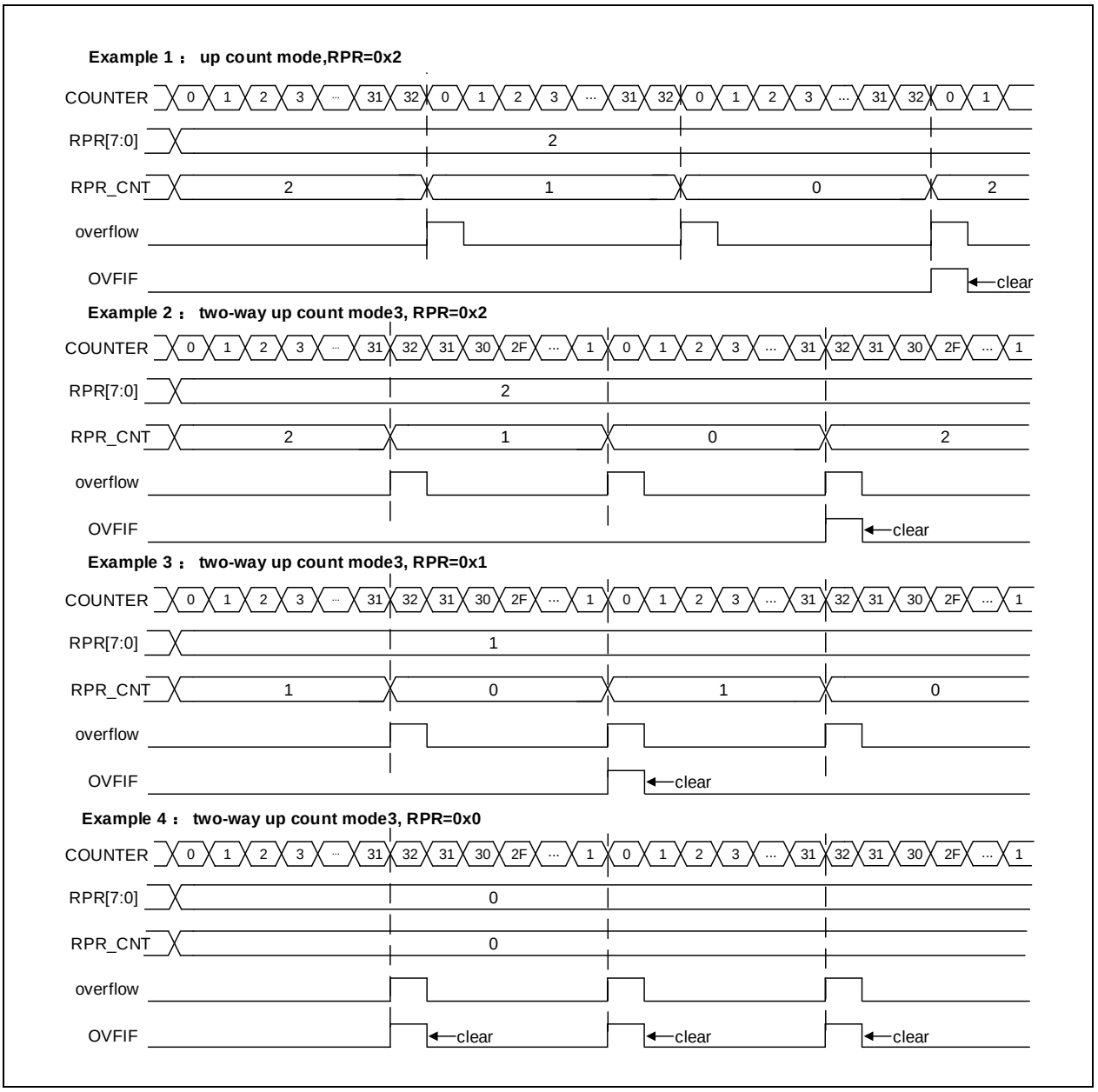
图 14-95 计数器时序图，内部时钟分频因子为1，TMRx_PR=0x32



重复计数模式：

TMRx_RPR 寄存器用于配置重复计数器计数周期，TMRx_RPR 寄存器为非 0 值时，重复计数模式启动。重复计数模式下，每（RPR[15:0]+1）次计数器溢出将产生一次溢出事件。每次计数器溢出，重复计数器递减，仅当重复计数器计数值等于 0 值时，计数器溢出会产生溢出事件。通过配置不同重复计数器值，可调整溢出事件产生的频率。

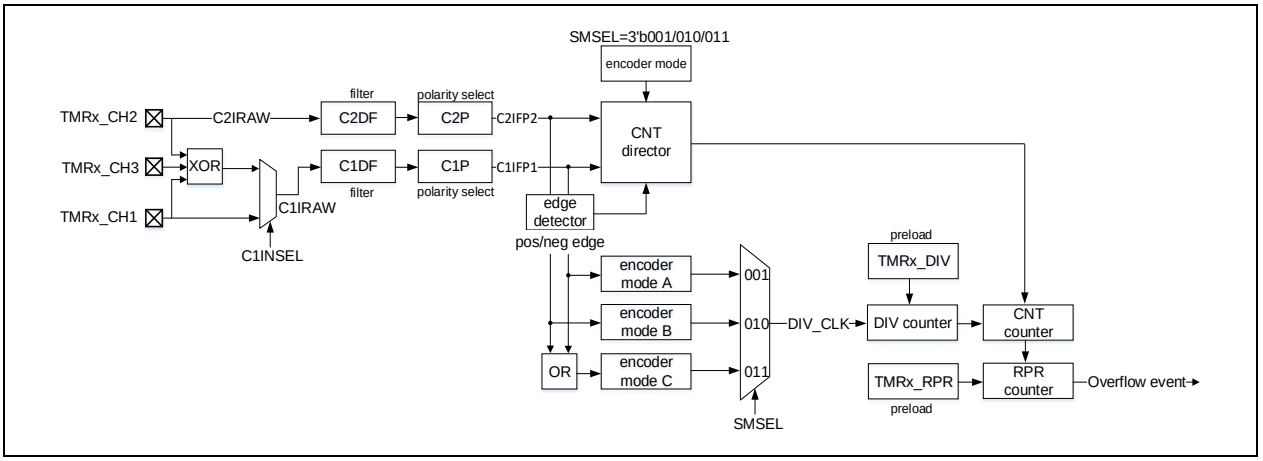
图 14-96 向上计数模式和中央双向对齐计数模式时OVFIF



编码器模式

编码器模式下需提供两组输入信号 TMRx_CH1 和 TMRx_CH2，根据一组输入信号电平值，计数器在另一组输入信号边沿向上或向下计数。计数方向由 OWCDIR 值指示。

图 14-97 编码模式结构



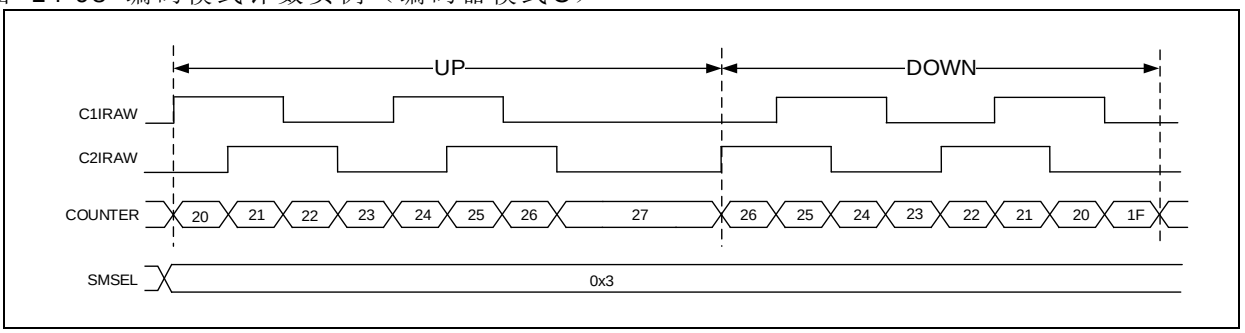
编码器模式 A：SMSEL=3’b001，计数器在 C1IFP1 边沿计数（上升沿和下降沿），计数方向由 C1IFP1 边沿方向和 C2IFP2 电平高低共同决定。
编码器模式 B：SMSEL=3’b010，计数器在 C2IFP2 边沿计数（上升沿和下降沿），计数方向由 C2IFP2 边沿方向和 C1IFP1 电平高低共同决定。
编码器模式 C：SMSEL=3’b011，计数器在 C1IFP1 和 C2IFP2 边沿计数（上升沿和下降沿），计数方向由 C1IFP1 边沿方向和 C2IFP2 电平高低、C2IFP2 边沿方向和 C1IFP1 电平高低共同决定共同决定。

若要使用编码器模式可按下面步骤配置：
-配置 TMRx_CM1 寄存器 C1DF[3:0]，设置通道 1 输入信号滤波；配置 TMRx_CCTRL 寄存器 C1P，设置通道 1 输入信号有效电平。
-配置 TMRx_CM1 寄存器 C2DF[3:0]，设置通道 2 输入信号滤波；配置 TMRx_CCTRL 寄存器 C2P，设置通道 2 输入信号有效电平。
-配置 TMRx_CM1 寄存器 C1C[1:0]，设置通道 1 为输入模式；配置 TMRx_CM1 寄存器 C2C[1:0]，设置通道 2 为输入模式；
-配置 TMRx_STCTRL 寄存器 SMSEL[2:0]，选择编码器模式 A（SMSEL=3’b001）、编码器模式 B（SMSEL=3’b010）或编码器模式 C（SMSEL=3’b011）。
-配置 TMRx_PR 寄存器 PR[15:0]，设置计数器计数周期。
-配置 TMRx_DIV 寄存器 DIV[15:0]，设置计数器计数频率。
-配置 TMRx_CH1 和 TMRx_CH2 对应 IO 为复用模式。
-配置 TMRx_CTRL1 寄存器 TMREN，使能计数器。

表 14-12 计数方向与编码器信号的关系

计数边沿	计数边沿相对信号的电平（C1IFP1 边沿对应 C2IFP2 电平，C2IFP2 边沿对应 C1IFP1 电平）	C1IFP1 边沿方向		C2IFP2 边沿方向	
		上升	下降	上升	下降
C1IFP1	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
C2IFP2	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
C1IFP1 和 C2IFP2	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

图 14-98 编码模式计数实例（编码器模式C）



14.5.3.3 TMR输入部分

TMR1 拥有 4 个独立通道，每个通道可配置为输入或输出，当配置位输入时，每个通道输入信号依次经过以下处理：

-TMRx_CHx 经过预处理输出 CxIRAW。配置 C1INSEL 位，选择 C1IRAW 来源是 TMRx_CH1 或是 TMRx_CH1、TMRx_CH2、TMRx_CH3 异或。C2IRAW、C3IRAW、C4IRAW 来源是 TMRx_CH2、TMRx_CH3、TMRx_CH4。

-CxIRAW 输入数字滤波器，输出滤波后信号 CxIF。数字滤波器通过 CxDF 位配置采样频率和次数。

-CxIF 输入边沿检测器，输出边沿选择后信号 CxIFPx。边沿选择由 CxP 和 CxCP 位共同控制，可选择输入上升沿、下降沿或双边沿有效。

-CxIFPx 输入捕获信号选择器，输出选择后信号 CxIN。捕获信号选择器由 CxC 控制，可选择 CxIN 来源为 CxIFPx、CyIFPx、STCI。其中 CyIFPx（x≠y）是来自通道 y 的 CyIFPy 经通道 x 边沿检测器处理后的信号（例如 C1IFP2 是来自通道 1 的 C1IFP1 信号经过通道 2 边沿检测器处理后的信号）；STCI 来自次定时器控制器，由 STIS 位选择来源。

-CxIN 经由输入通道分频器，输出分频后信号 CxIPS。分频系数由 CxIDIV 位配置为不分频、2 分频、4 分频或 8 分频。

图 14-99 输入/输出通道 1 的主电路

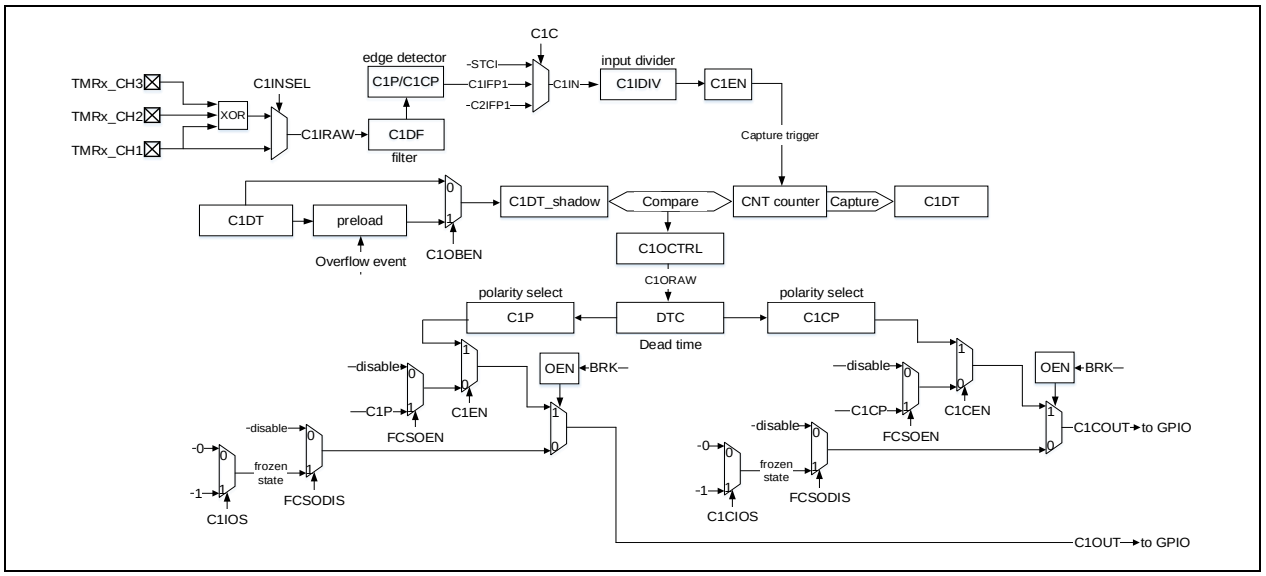
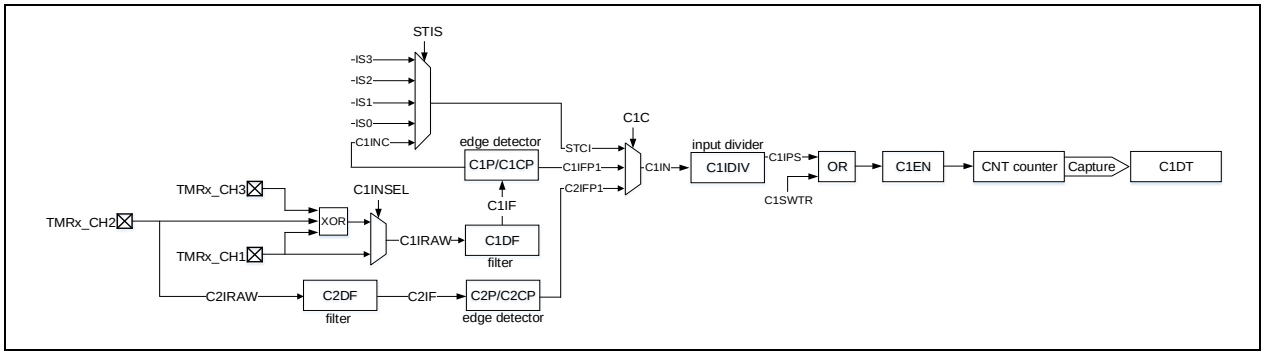


图 14-100 通道1输入部分



输入模式

此模式下，当选中的触发信号被检测到，通道寄存器（TMRx_CxDT）记录当前计数器计数值，并将捕获比较中断标志位（CxIF）置 1，若已使能通道中断（CxIEN）、通道 DMA 请求（CxDEN）则产生相应的中断和 DMA 请求。若在 CxIF 置 1 后检测到触发信号，将产生捕获溢出事件，TMRx_CxDT 会使用当前计数器计数值覆盖之前记录的计数器计数值，同时通道再捕获标志位（CxRF）置 1。

以若要捕获 C1IN 输入的上升沿，可按如下进行配置：

- 将通道模式寄存器 1（TMRx_CM1）中的 C1C 位配置为 01，选择 C1IN 作为通道 1 输入。
- 配置 C1IN 信号滤波器带宽（CxDF[3：0]）。
- 配置 C1IN 通道的有效沿，在 TMR1_CCTRL 寄存器中写入 C1P=0（上升沿）。
- 配置 C1IN 信号捕获分频（C1DIV[1：0]）。
- 使能通道 1 输入捕获（C1EN=1）。
- 根据需要设置 TMR1_IDEN 寄存器中的 C1IEN 位、TMR1_IDEN 寄存器中的 C1DEN 位，选择中断请求或 DMA 请求。

多输入异或

通道 1 的输入端可选择 TMR1_CH1、TMR1_CH2 和 TMR1_CH3 经异或逻辑后输入。将 TMR1_CTRL2 寄存器中的 C1INSEL 位置 1 可开启此功能。

多输入异或功能可用于连接霍尔传感器，例如，将异或输入的三个输入端分别连接到三个霍尔传感器，通过分析三路霍尔传感器信号可计算出转子的位置和速度。

PWM 输入

PWM 输入模式适用于通道 1 和 2，要使用此模式，需要将 C1IN 和 C2IN 映射到同一 TMRx_CHx，并且通道 1 或 2 的 CxIFPx 配置成触发次定时器控制器复位。

PWM 输入模式可用于测量输入信号的周期和占空比，如需测量通道 1 输入信号的周期和占空比，操作步骤如下：

- 配置C1C=2‘b01，选择C1IN为C1IFP1。
- 配置C1P=1’b0，选择C1IFP1上升沿有效。
- 配置C2C=2‘b10，选择C2IN为C1IFP2。
- 配置C2P=1’b1，选择C1IFP2下降沿有效。
- 配置STIS=3’b101，选择次定时器触发信号为C1IFP1。
- 配置SMSEL=3‘b100，选择次定时器模式为复位模式。
- 配置C1EN=1’b1，C2EN=1’b1。使能通道1和输入捕获。

上述配置下，通道 1 输入信号的上升沿会触发捕获并将捕获值存储到 C1DT 寄存器，同时通道 1 输入信号上升沿复位计数器。通道 1 输入信号下降沿触发捕获并将捕获值存储到 C2DT 寄存器。通道 1 输入信号的周期可通过 C1DT 计算，占空比可通过 C2DT 计算。

图 14-101 PWM输入模式配置实例

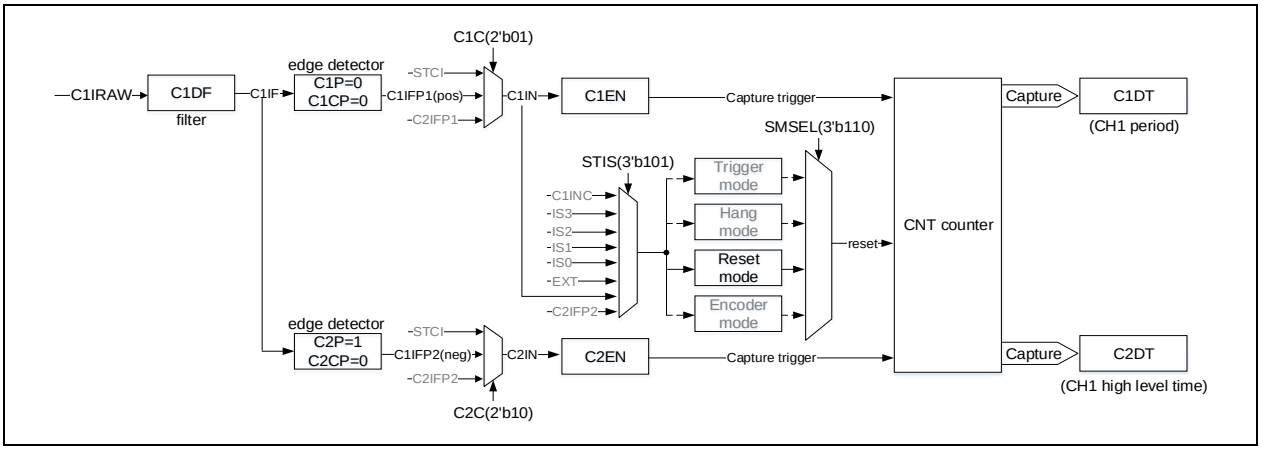
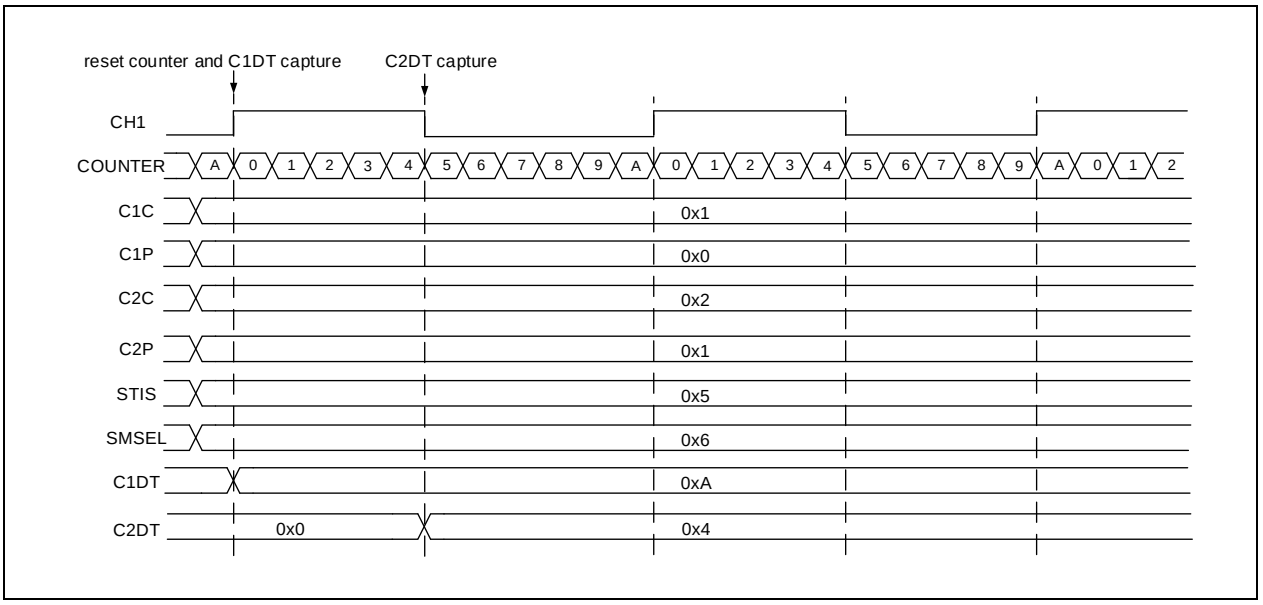


图 14-102 PWM输入模式



14.5.3.4 TMR输出部分

TMR 的输出部分由比较器和输出控制构成，用于编程输出信号的周期、占空比、极性。高级定时器的输出部分在不同通道上有所不同，如下图所示：

图 14-103 通道1至3输出部分

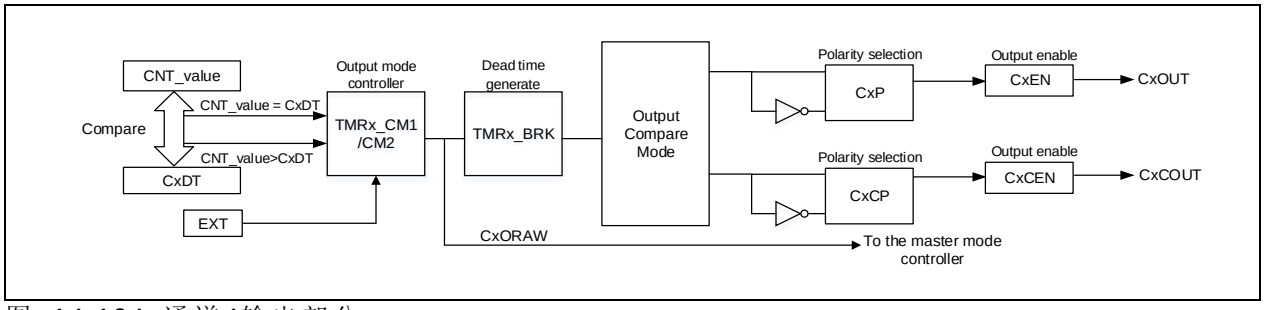
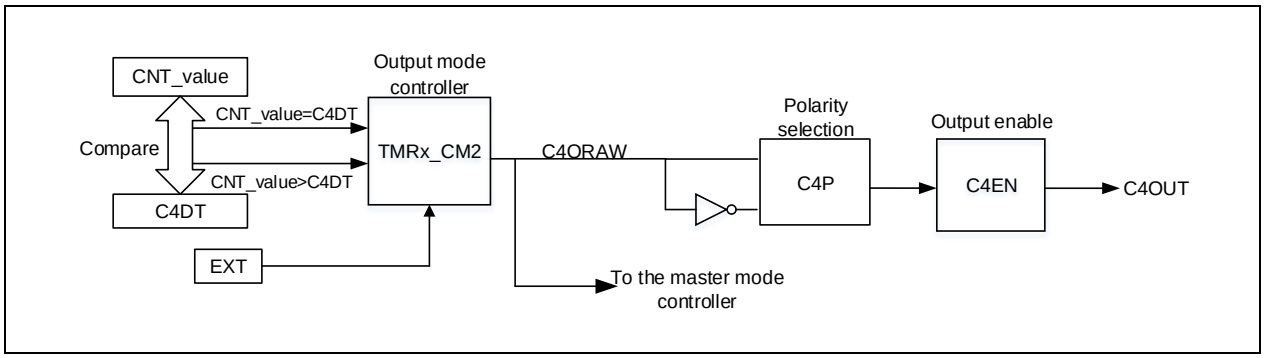


图 14-104 通道4输出部分



输出模式

配置 CxC[1：0]≠2'b00 将通道配置为输出可实现多种输出模式，此时，计数器计数值将与 CxDT 寄存器值比较，并根据 CxOCTRL[2：0]位配置的输出模式，产生中间信号 CxORAW，再经过输出控制逻辑处理后输送到 IO。输出信号的周期由 TMR1_PR 寄存器值配置，占空比则由 CxDT 寄存器值配置。

输出比较模式有以下子类：

PWM 模式 A：CxOCTRL=3'b110 时，开启 PWM 模式 A。向上计数时，TMRx_C1DT>TMRx_CVAL 时 C1ORAW 输出高电平，否则为低电平；向下计数时，TMRx_C1DT<TMRx_CVAL 时 C1ORAW 输出低电平，否则为高电平。若要使用 PWM 模式 A，可按如下方式配置。

- -配置 TMRx_PR 寄存器，设置 PWM 周期。
- -配置 TMRx_CxDT 寄存器，设置 PWM 占空比。

●-配置 TMRx_CM1/CM2 寄存器 CxOCTRL 位为 3’b110，设置输出模式为 PWM 模式 A。
●-配置 TMRx_DIV 寄存器，设置计数器计数频率。
●-配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
●-配置 TMRx_CCTRL 寄存器 CxP 位、CxCP 位，设置输出极性。
●-配置 TMRx_CCTRL 寄存器 CxEN 位、CxCEN 位，使能通道输出。
●-配置 TMRx_BRK 寄存器 OEN 位，使能 TMRx 输出。
●-配置 TMR 输出通道对应 GPIO 为对应的复用模式。
●-配置 TMRx_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

PWM 模式 B：CxOCTRL=3’b111 时，开启 PWM 模式 B。向上计数时，TMRx_C1DT>TMRx_CVAL 时 C1ORAW 输出低电平，否则为高电平；向下计数时，TMRx_C1DT<TMRx_CVAL 时 C1ORAW 输出高电平，否则为低电平。

强制输出模式：CxOCTRL=3’b100/101 时，开启强制输出模式。此时，CxORAW 信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。

● **输出比较模式：**CxOCTRL=3’b001/010/011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 强制输出高电平（CxOCTRL=3’b001）、低电平（CxOCTRL=3’b010）或进行电平翻转（CxOCTRL=3’b011）。

● **单周期模式：**PWM 模式的特例，将 OCMEN 位置 1 可开启单周期模式，此模式下，仅在当前计数周期中进行比较匹配，完成当前计数后，TMREN 位清 0，因此仅输出一个脉冲。当配置为向上计数模式时，需要严格配置 CVAL<CxDT≤PR；向下计数时，需严格配置 CVAL>CxDT。

● **快速输出模式：**将 CxOIEN 位置 1 可开启此功能，开启后 CxORAW 电平值不再在计数值与 CxDT 匹配时变化，而是在当前计数周期开始时，也就是说，比较结果被提前了，计数器值与 CxDT 寄存器的比较结果将会提前决定 CxORAW 的电平。

图 14-107 展示了输出比较模式（翻转）的例子，C1DT=0x3，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-108 展示了计数器向上计数与 PWM 模式 A 配合的例子，PR=0x32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-109 展示了计数器中央双向对齐计数与 PWM 模式 A 配合的例子，PR=0X32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-110 展示了计数器向上计数与单周期模式下 PWM 模式 B 配合的例子，计数器仅计数了一个周期，输出信号在这个周期中只输出了一个脉冲。

图 14-105 计数值与C1DT值匹配时翻转C1ORAW

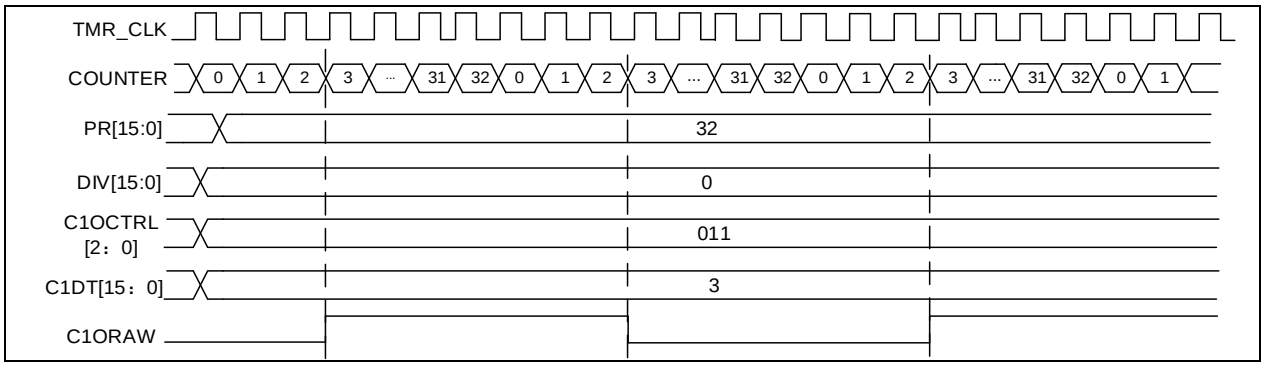


图 14-106 向上计数下PWM模式A

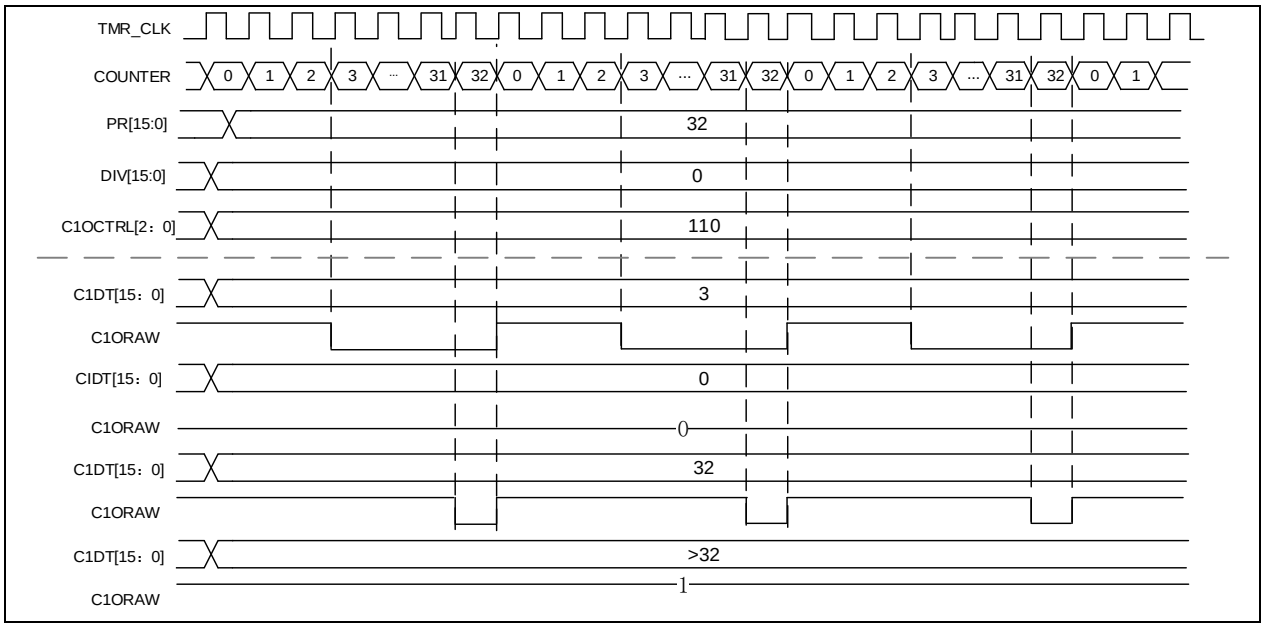


图 14-107中央双向对齐计数下PWM模式

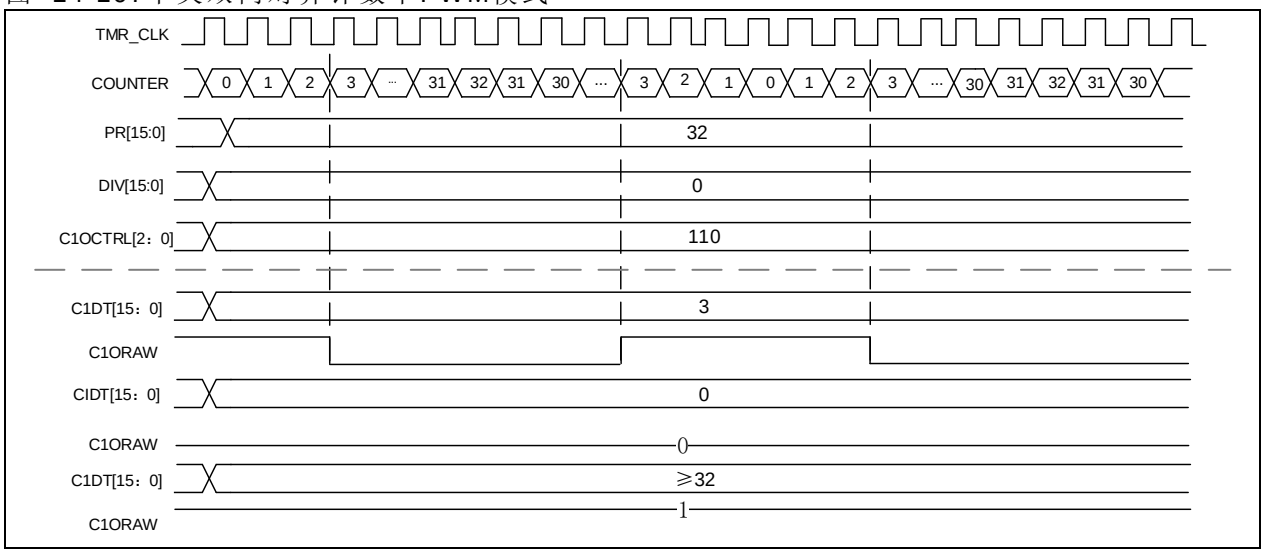
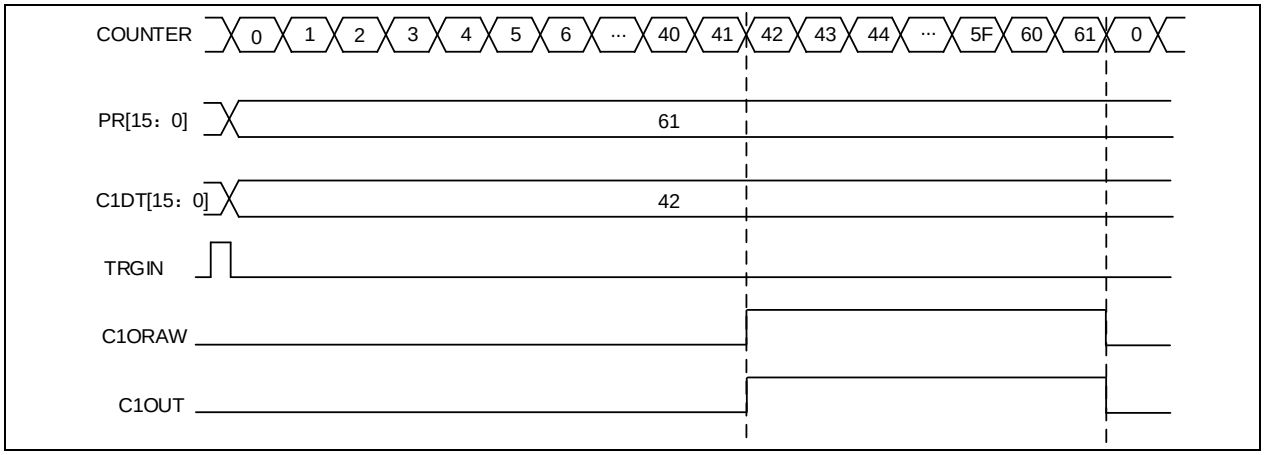


图 14-108 单周期模式



主定时器事件输出

当 TMR 作为主定时器时，可选择如下信号源作为 TRGOUT 信号输出到次定时器，选择信号为 TMRxCTRL2 寄存器 PTOS 位。

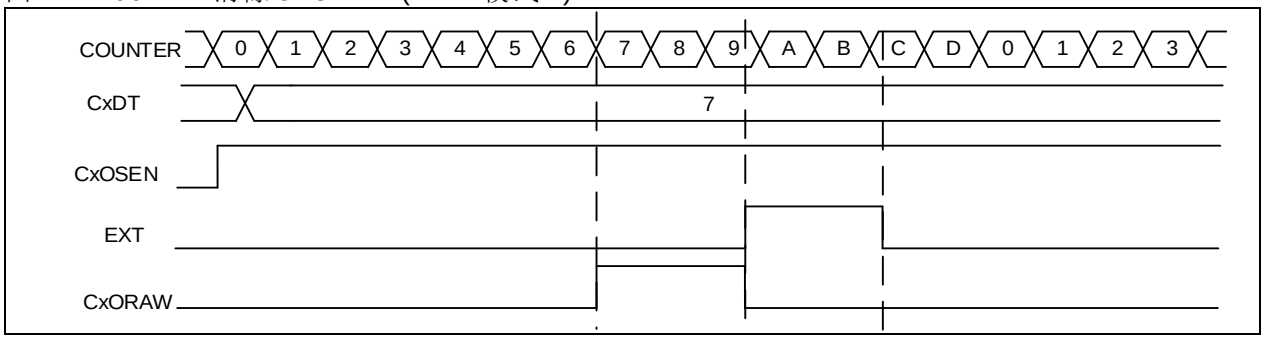
●-PTOS=3’b000，TRGOUT 输出软件溢出事件（TMRx_SWEVT 寄存器 OVFSWTR 位）。
●-PTOS=3’b001，TRGOUT 输出计数器使能信号。
●-PTOS=3’b010，TRGOUT 输出计数器溢出事件。
●-PTOS=3’b011，TRGOUT 输出捕获、比较事件。
●-PTOS=3’b100，TRGOUT 输出 C1ORAW 信号。
●-PTOS=3’b101，TRGOUT 输出 C2ORAW 信号。
●-PTOS=3’b110，TRGOUT 输出 C3ORAW 信号。
●-PTOS=3’b111，TRGOUT 输出 C4ORAW 信号。

CxORAW 信号清除

将 CxOSEN 位置 1 后，指定通道的 CxORAW 信号由 EXT 高电平清 0，在下一次溢出事件发生前 CxORAW 信号无法被改变。

强制输出模式时，CxORAW 信号清除功能不可用，只有在输出比较模式或 PWM 模式，此功能有效。下图显示了使用 EXT 信号清除 CxORAW 的例子，当 EXT 为高电平期间，原本为高电平的 CxORAW 信号被拉低，当 EXT 为低电平时，CxORAW 根据计数值和 CxDT 比较结果输出电平。

图 14-109 EXT清除CxORAW(PWM模式A)



死区插入

高级定时器通道 1 至 3 包含一组反向通道输出，通过 CxCEN 使能，通过 CxCP 配置极性。CxOUT 和 CxCOUT 的输出状态见表 14-14。

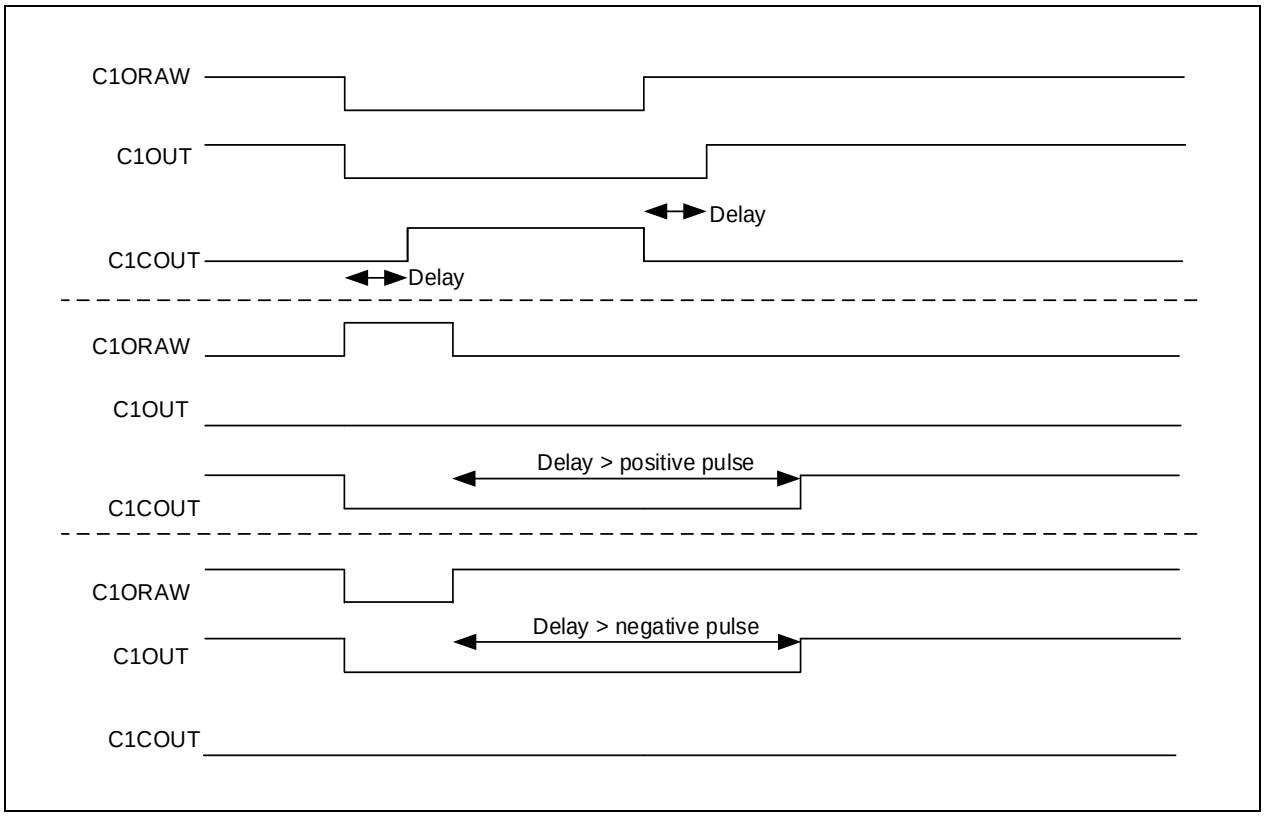
当转换为 IDLEF 状态，即 OEN 下降到 0，死区被激活。

将 CxEN 和 CxCEN 位置 1 后，通过配置 DTC[7：0]死区发生器，可插入不同时长的死区。插入死区后，CxOUT 的上升沿延迟于参考信号的上升沿；CxCOUT 的上升沿延迟于参考信号的下降沿。

如果延迟大于当前有效的输出宽度，C1OUT 和 C1COUT 不会产生相应的脉冲，死区时间应小于有效的输出宽度。

下列图显示了 CxP=0、CxCP=0、OEN=1、CxEN=1 并且 CxCEN=1 时死区插入的例子

图 14-110 带死区插入的互补输出



14.5.3.5 TMR刹车功能

开启停止功能后（STPEN 位置 1），CxOUT 和 CxCOUT 由 OEN、FCSODIS、FCSOEN、CxIOS 和 CxCIOS 共同控制。但 CxOUT 和 CxCOUT 输出总是不能同时处于有效电平上的。详见表 14-14 带停止功能的互补输出通道 CxOUT 和 CxCOUT 的控制位。

停止信号来源可以是停止输入管脚、时钟失效事件，停止输入信号的极性由 STPV 位控制。

当发生停止事件时，有下述动作：

- OEN 位异步清零，通道输出状态由 FCSODIS 位选择。关闭 MCU 的振荡器不影响该功能。
- OEN 被清零后，通道输出电平由 CxIOS 位设定。如果 FCSODIS=0，则定时器输出使能被禁止，否则输出使能始终为高。
- 当使用互补输出时：
  - 输出最开始处于复位状态，也就是无效的状态（取决于极性）。这是异步操作，定时器有无时钟并不影响此功能。
  - 定时器的时钟如果有效，会开启死区生成功能，CxIOS和CxCIOS位用来配置死区之后的电平。即使在这种情况下，CxOUT和CxCOUT也不能被同时驱动到有效的电平。

  注意，由于 OEN 位同步逻辑，死区时间较通常情况会延长一段时间（大约 2 个 ck_tim 的时钟周期）。
  - 如果FCSODIS=0，定时器释放使能输出，否则保持使能输出；或一旦CxEN与CxCEN之一变高时，使能输出变为高。
- 如果开启了停止中断或 DMA 功能，停止状态标志将置 1，并产生停止中断或 DMA 请求。
- 如果将 AOEN 位置 1，在下一个溢出事件时 OEN 位被自动置 1。

注意：停止输入电平有效时，OEN 不能被设置，状态标志 BRKIF 也不能被清除。

图 14-111 TMR输出控制

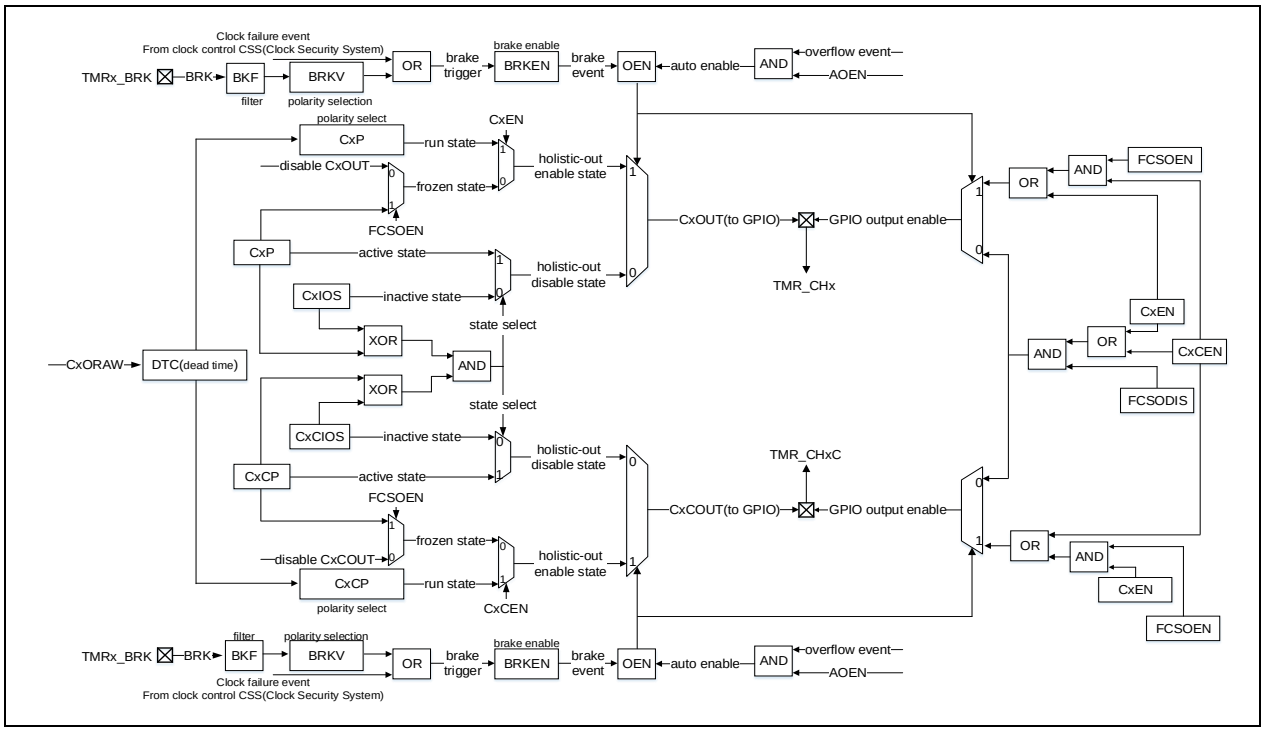
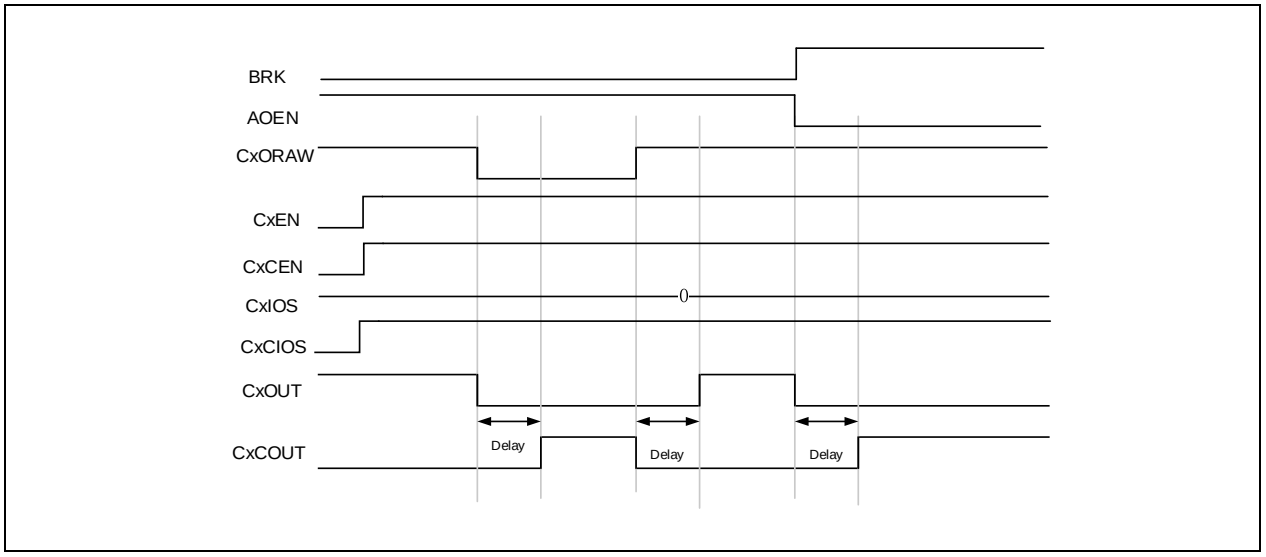


图 14-112 TMR停止功能的例子



14.5.3.6 TMR同步

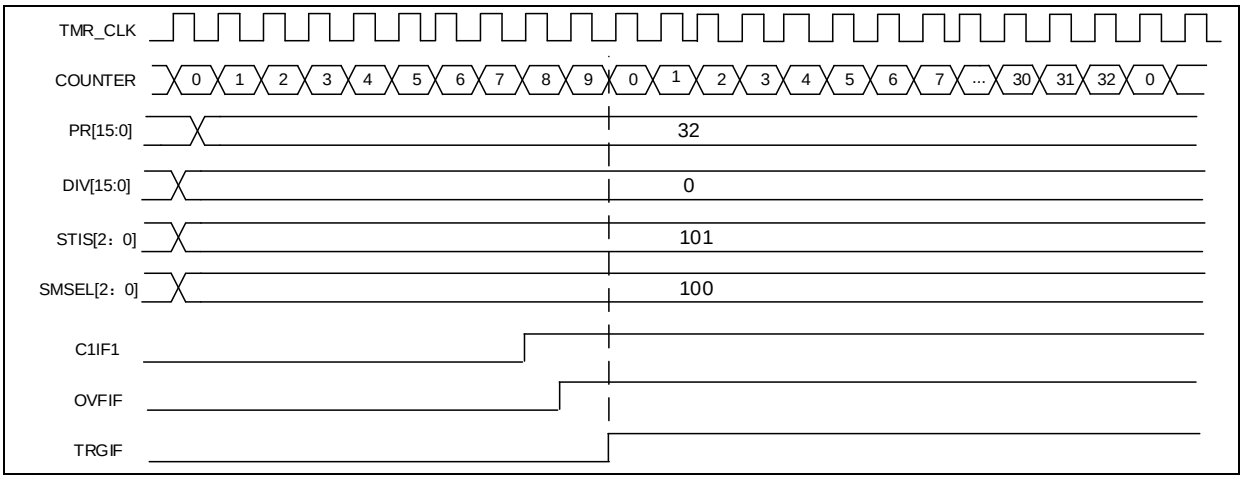
主次定时器之间可由内部连接信号进行同步。主定时器可由 PTOS[2：0]位选择主定时器输出，即同步信息；次定时器由 SMSEL[2：0]位选择从模式，即次定时器的工作模式。

定时器从模式有以下几种：

从模式：复位模式

选中的触发信号将复位计数器和预分频器，若 OVFS 位为 0，将产生一个溢出事件。

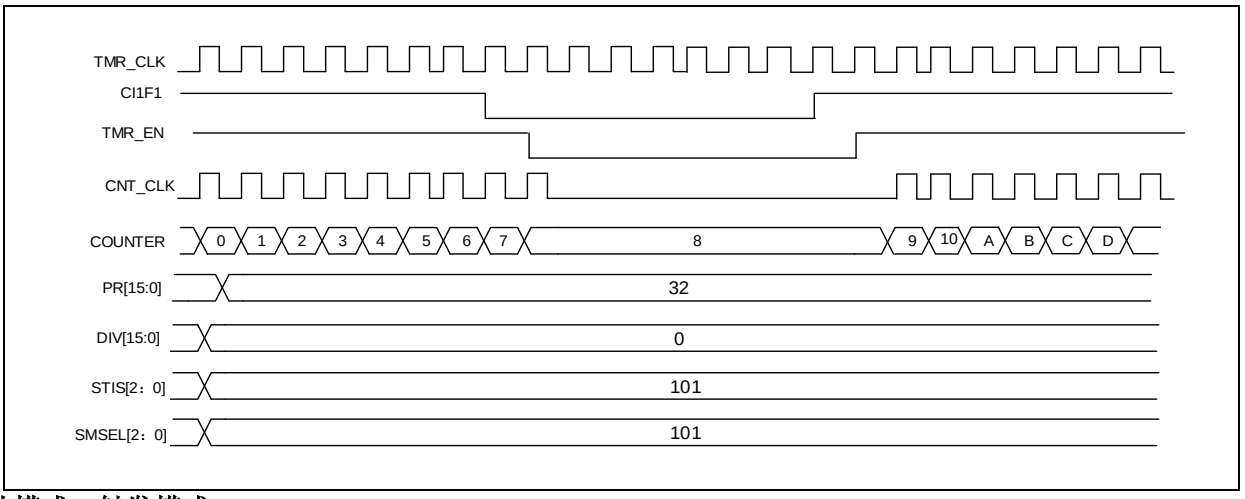
图 14-113 复位模式例子



从模式：挂起模式

挂起模式下，计数的计数和停止受选中触发输入信号控制，当触发输入为高电平时计数器开始计数；当为低电平时，计数器暂停计数。

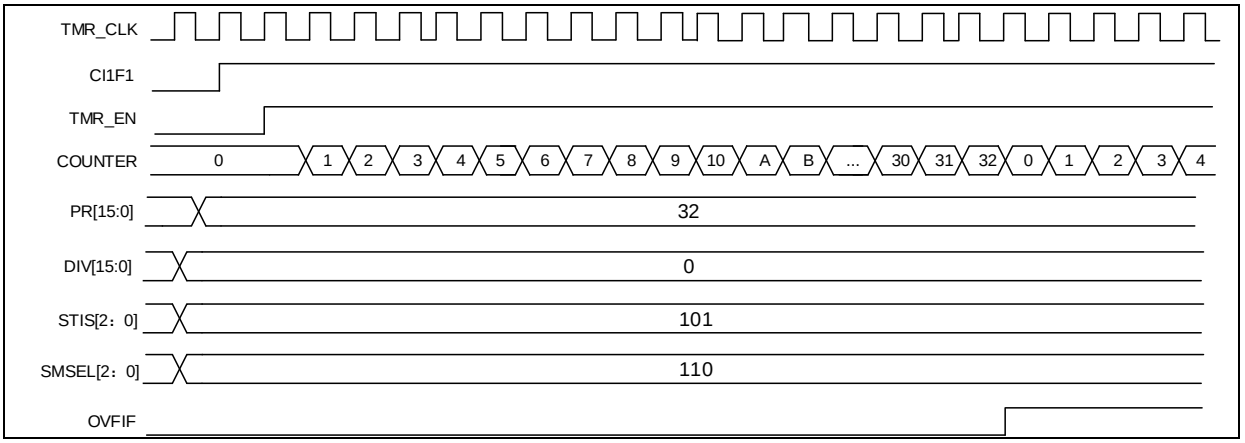
图 14-114 挂起模式下例子



从模式：触发模式

计数器将在选中的触发输入上升沿启动计数（将 TMR_EN 置 1）。

图 14-115 触发器模式例子



14.5.3.7 调试模式

当微控制器进入调试模式（Cortex®-M4F 核心停止）时，将 DEBUG 模块中的 TMR1_PAUSE 置 1，可以使 TMR1 计数器暂停计数。

14.5.4 TMR1寄存器描述

必须以字（32 位）的方式操作这些外设寄存器。

下表中将 TMR1 的所有寄存器映射到一个 16 位可寻址（编址）空间

表 14-13 TMR1寄存器图和复位值

寄存器简称	基址偏移量	复位值
TMR1_CTRL1	0x00	0x0000
TMR1_CTRL2	0x04	0x0000
TMR1_STCTRL	0x08	0x0000
TMR1_IDEN	0x0C	0x0000
TMR1_ISTS	0x10	0x0000
TMR1_SWEVT	0x14	0x0000
TMR1_CM1	0x18	0x0000
TMR1_CM2	0x1C	0x0000
TMR1_CCTRL	0x20	0x0000
TMR1_CVAL	0x24	0x0000
TMR1_DIV	0x28	0x0000
TMR1_PR	0x2C	0x0000
TMR1_RPR	0x30	0x0000
TMR1_C1DT	0x34	0x0000
TMR1_C2DT	0x38	0x0000
TMR1_C3DT	0x3C	0x0000
TMR1_C4DT	0x40	0x0000
TMR1_BRK	0x44	0x0000
TMR1_DMACTRL	0x48	0x0000
TMR1_DMADT	0x4C	0x0000

14.5.4.1 TMR1控制寄存器1（TMR1_CTRL1）

域	简称	复位值	类型	功能
位 15：10	保留	0x00	resd	保持默认值。
位 9：8	CLKDIV	0x0	rw	时钟除频（Clock divider） 此位用于设置数字滤波器采样频率 f_{DTS} 和定时器时钟频率 f_{CK_INT} 之间的分频比，也用于调整死区时间的时基 T_{DTS} 和定时器时钟周期 T_{CK_INT} 的分频比。 00：无除频，$f_{DTS}=f_{CK\_INT}$； 01：2 除频，$f_{DTS}=f_{CK\_INT}/2$； 10：4 除频，$f_{DTS}=f_{CK\_INT}/4$； 11：保留。
位 7	PRBEN	0x0	rw	周期缓冲使能（Period buffer enable） 0：缓冲关闭； 1：缓冲开启。
位 6：5	TWCMSEL	0x0	rw	中央双向对齐计数模式选择（Two-way count mode selection） 00：单向对齐计数模式，方向由 OWCDIR 配置； 01：中央双向对齐计数模式 1，上下交替计数，CxIF 位只在计数器向下计数时被置起； 10：中央双向对齐计数模式 2，上下交替计数，CxIF 位只在计数器向上计数时被置起； 11：中央双向对齐计数模式 3，上下交替计数，CxIF 位在计数器向上和向下计数时皆被置起。

位 4	OWCDIR	0x0	rw	单向计数方向 （One-way count direction） 0：向上； 1：向下。
位 3	OCMEN	0x0	rw	单周期使能 （One cycle mode enable） 该功能用于选择溢出事件后，计数器是否停止。 0：关闭； 1：开启。
位 2	OVFS	0x0	rw	溢出事件源选择 （Overflow event source） 配置溢出事件或 DMA 请求来源。 0：来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件； 1：只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 （Overflow event enable） 0：开启； 1：关闭。
位 0	TMREN	0x0	rw	使能定时器（TMR enable） 0：关闭； 1：开启。

14.5.4.2 TMR1控制寄存器2（TMR1_CTRL2）

域	简称	复位值	类型	功能
位 31：15	保留	0x0000	resd	保持默认值
位 14	C4IOS	0x0	rw	通道 4 空闲输出状态（Channel 4 idle output state）
位 13	C3CIOS	0x0	rw	通道 3 互补空闲输出状态（Channel 3 complementary idle output state）
位 12	C3IOS	0x0	rw	通道 3 空闲输出状态（Channel 3 idle output state）
位 11	C2CIOS	0x0	rw	通道 2 互补空闲输出状态（Channel 2 complementary idle output state）
位 10	C2IOS	0x0	rw	通道 2 空闲输出状态（Channel 2 idle output state）
位 9	C1CIOS	0x0	rw	通道 1 互补空闲输出状态（Channel 1 complementary idle output state） 输出关闭（OEN = 0），死区发生后： 0：C1COUT=0； 1：C1COUT=1。
位 8	C1IOS	0x0	rw	通道 1 空闲输出状态（Channel 1 idle output state） 输出关闭（OEN = 0），死区发生后： 0：C1OUT=0。 1：C1OUT=1。
位 7	C1INSEL	0x0	rw	C1IN 选择 （C1IN selection） 0：CH1 引脚连到 C1IRAW 输入； 1：CH1、CH2 和 CH3 引脚异或结果连到 C1IRAW 输入。
位 6：4	PTOS	0x0	rw	主定时器输出信号选择 （Primary TMR output selection） TMRx 输出到次定时器的信号选择： 000：复位； 001：使能； 010：溢出； 011：比较脉冲； 100：C1ORAW 信号； 101：C2ORAW 信号； 110：C3ORAW 信号； 111：C4ORAW 信号。
位 3	DRS	0x0	rw	DMA 请求源（DMA request source） DMA 请求来源。 0：通道事件； 1：溢出事件。
位 2	CCFS	0x0	rw	通道控制位刷新选择（Channel control bit refresh select） 对具有互补输出的通道，如果通道控制位有缓存时： 0：通过设置 HALL 位刷新控制位；

				1：通过设置 HALL 位或 TRGIN 的上升沿刷新控制位。
位 1	保留	0x0	resd	保持默认值。
位 0	CBCTRL	0x0	rw	通道缓存控制（Channel buffer control） 对具有互补输出的通道： 0：CxEN，CxCEN 和 CxOCTRL 位无缓存； 1：CxEN，CxCEN 和 CxOCTRL 位有缓存。

14.5.4.3 TMR1次定时器控制寄存器（TMR1_STCTRL）

域	简称	复位值	类型	功能
位 15	ESP	0x0	rw	外部信号极性（External signal polarity） 用于选择外部方式。 0：高电平或上升沿； 1：低电平或下降沿。
位 14	ECMBEN	0x0	rw	外部时钟模式 B 使能 （External clock mode B enable） 用于启用外部时钟模式 B 0：关闭； 1：开启。
位 13：12	ESDIV	0x0	rw	外部信号除频 （External signal divide） 用于选择降低外部触发频率的除频。 00：关闭分频； 01：2 分频； 10：4 分频； 11：8 分频。
位 11：8	ESF	0x0	rw	外部信号滤波（External signal filter） 用于过滤外部信号，当外部信号产生了 N 次之后才能被采样。 0000：无滤波器，以f_{DTS}采样 0001：$f_{SAMPLING}$ =f_{CK_INT}，N=2； 0010：$f_{SAMPLING}$ =f_{CK_INT}，N=4； 0011：$f_{SAMPLING}$ =f_{CK_INT}，N=8； 0100：$f_{SAMPLING}$ =f_{DTS}/2，N=6； 0101：$f_{SAMPLING}$ =f_{DTS}/2，N=8； 0110：$f_{SAMPLING}$ =f_{DTS}/4，N=6； 0111：$f_{SAMPLING}$=f_{DTS}/4， N=8； 1000：$f_{SAMPLING}$=f_{DTS}/8， N=6； 1001：$f_{SAMPLING}$=f_{DTS}/8， N=8； 1010：$f_{SAMPLING}$=f_{DTS}/16，N=5； 1011：$f_{SAMPLING}$=f_{DTS}/16，N=6； 1100：$f_{SAMPLING}$=f_{DTS}/16，N=8； 1101：$f_{SAMPLING}$=f_{DTS}/32，N=5； 1110：$f_{SAMPLING}$=f_{DTS}/32，N=6； 1111：$f_{SAMPLING}$=f_{DTS}/32，N=8。
位 7	STS	0x0	rw	次定时器同步 （Subordinate TMR synchronization） 该位开启后，主次定时器可实现高度同步。 0：关闭； 1：开启。
位 6：4	STIS	0x0	rw	次定时器输入选择 （Subordinate TMR input selection） 用于次定时器的输入选择。 000：内部选择 0（IS0）； 001：内部选择 1（IS1）； 010：内部选择 2（IS2）； 011：内部选择 3（IS3）； 100：C1IRAW 的输入检测器（C1INC）； 101：滤波输入 1（C1IF1）； 110：滤波输入 2（C2IF2）； 111：外部输入（EXT）。 关于每个定时器中 ISx 的细节，参见表 14-11。
位 3	保留	0x0	resd	保留，保持默认值。
位 2：0	SMSEL	0x0	rw	次定时器模式选择 （Subordinate TMR mode selection） 000：关闭从模式；

				001：编码模式 A； 010：编码模式 B； 011：编码模式 C； 100：复位模式 - TRGIN 输入上升沿时，重新初始化计数器； 101：挂起模式 - TRGIN 输入高电平时，计数器计数； 110：触发模式 - TRGIN 输入上升沿时，产生触发事件； 111：外部时钟模式 A - TRGIN 输入上升沿提供时钟； 注：编码器模式 A/B/C 配置方法请查看计数模式章节。

14.5.4.4 TMR1 DMA/中断使能寄存器（TMR1_IDEN）

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值。
位 14	TDEN	0x0	rw	触发 DMA 请求使能（Trigger DMA request enable） 0：关闭； 1：开启。
位 13	HALLDE	0x0	rw	HALL DMA 请求使能 （HALL DMA request enable） 0：关闭； 1：开启。
位 12	C4DEN	0x0	rw	通道 4 的 DMA 请求使能 （Channel 4 DMA request enable） 0：关闭； 1：开启。
位 11	C3DEN	0x0	rw	通道 3 的 DMA 请求使能 （Channel 3 DMA request enable） 0：关闭； 1：开启。
位 10	C2DEN	0x0	rw	通道 2 的 DMA 请求使能 （Channel 2 DMA request enable） 0：关闭； 1：开启。
位 9	C1DEN	0x0	rw	通道 1 的 DMA 请求使能 （Channel 1 DMA request enable） 0：关闭； 1：开启。
位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能（overflow event DMA request enable） 0：关闭； 1：开启。
位 7	BRKIE	0x0	rw	刹车中断使能（Brake interrupt enable） 0：关闭； 1：开启。
位 6	TIEN	0x0	rw	触发中断使能（Trigger interrupt enable） 0：关闭； 1：开启。
位 5	HALLIEN	0x0	rw	HALL 中断使能（HALL interrupt enable） 0：关闭； 1：开启。
位 4	C4IEN	0x0	rw	通道 4 中断使能（Channel 4 interrupt enable） 0：关闭； 1：开启。
位 3	C3IEN	0x0	rw	通道 3 中断使能（Channel 3 interrupt enable） 0：关闭； 1：开启。
位 2	C2IEN	0x0	rw	通道 2 中断使能（Channel 2 interrupt enable） 0：关闭； 1：开启。
位 1	C1IEN	0x0	rw	通道 1 中断使能（Channel 1 interrupt enable） 0：关闭； 1：开启。

位 0	OVFIEN	0x0	rw	溢出中断使能（Overflow interrupt enable） 0：关闭； 1：开启。

14.5.4.5 TMR1中断状态寄存器（TMR1_ISTS）

域	简称	复位值	类型	功能
位 15：13	保留	0x0	resd	保持默认值。
位 12	C4RF	0x0	rw0c	通道 4 再捕获标记 （Channel 4 recapture flag） 见 C1RF 的描述。
位 11	C3RF	0x0	rw0c	通道 3 再捕获标记 （Channel 3 recapture flag） 见 C1RF 的描述。
位 10	C2RF	0x0	rw0c	通道 2 再捕获标记 （Channel 2 recapture flag） 见 C1RF 的描述。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记 （Channel 1 recapture flag） C1IF 的状态已经为’1’时是否再次发生了捕获，由硬件置’1’，写’0’清除。 0：无捕获发生； 1：捕获发生。
位 8	保留	0x0	resd	保持默认值。
位 7	BRKIF	0x0	rw0c	刹车中断标记 （Brake interrupt flag） 用于标记刹车输入的电平是否有效，由硬件置’1’，写’0’清除。 0：无效； 1：有效。
位 6	TRGIF	0x0	rw0c	触发中断标记 （Trigger interrupt flag） 当发生触发事件时由硬件置’1’，写’0’清除。 0：无触发事件发生； 1：发生触发事件。 触发事件：在 TRGIN 接收到有效边沿，或挂起模式下接收到任意边沿。
位 5	HALLIF	0x0	rw0c	HALL 中断标记 （HALL interrupt flag） 当发生触发事件时由硬件置’1’，写’0’清除。 0：无 HALL 事件发生； 1：发生 HALL 事件。 HALL 事件：CxEN、CxLEN、CxOCTRL 已被更新。
位 4	C4IF	0x0	rw0c	通道 4 中断标记 （Channel 4 interrupt flag） 见 C1IF 的描述。
位 3	C3IF	0x0	rw0c	通道 3 中断标记 （Channel 3 interrupt flag） 见 C1IF 的描述。
位 2	C2IF	0x0	rw0c	通道 2 中断标记 （Channel 2 interrupt flag） 见 C1IF 的描述。
位 1	C1IF	0x0	rw0c	通道 1 中断标记 （Channel 1 interrupt flag） 若通道 1 为输入模式时： 捕获事件发生时由硬件置’1’，由软件清’0’或读 TMR1_C1DT 清’0’。 0：无捕获事件发生； 1：发生捕获事件。 若通道 1 为输出模式时： 比较事件发生时由硬件置’1’，由软件清’0’。 0：无比较事件发生； 1：发生比较事件。
位 0	OVFIF	0x0	rw0c	溢出中断标记 （Overflow interrupt flag） 当溢出事件发生时由硬件置’1’，由软件清’0’。 0：无溢出事件发生； 1：发生溢出事件，若 TMR1_CTRL1 的 OVFEN=0、OVFS=0 时： – 当 TMR1_SWEVE 寄存器的 OVFG=1 时产生溢出事件； – 当计数值 CVAL 被触发事件重初始化时产生溢出事件。

14.5.4.6 TMR1软件事件寄存器（TMR1_SWEVT）

域	简称	复位值	类型	功能
位 15：8	保留	0x00	resd	保持默认值。
位 7	BRKSWTR	0x0	wo	软件触发刹车事件（Brake event triggered by software） 通过软件触发一个刹车事件。 0：无作用； 1：制造一个刹车事件。
位 6	TRGSWTR	0x0	wo	软件触发触发事件（Trigger event triggered by software） 通过软件触发一个触发事件。 0：无作用； 1：制造一个触发事件。
位 5	HALLSWTR	0x0	wo	软件触发 HALL 事件（HALL event triggered by software） 通过软件产生一个 HALL 事件。 0：无作用； 1：产生一个 HALL 事件。 注：该位只对拥有互补输出的通道有效。
位 4	C4SWTR	0x0	wo	软件触发通道 4 事件（Channel 4 event triggered by software） 见 C1M 的描述。
位 3	C3SWTR	0x0	wo	软件触发通道 3 事件（Channel 3 event triggered by software） 见 C1M 的描述。
位 2	C2SWTR	0x0	wo	软件触发通道 2 事件（Channel 2 event triggered by software） 见 C1M 的描述。
位 1	C1SWTR	0x0	wo	C1SWTR：软件触发通道 1 事件（Channel 1 event triggered by software） 通过软件触发一个通道 1 事件。 0：无作用； 1：制造一个通道 1 事件。
位 0	OVFSWTR	0x0	wo	软件触发溢出事件（Overflow event triggered by software） 通过软件触发一个溢出事件。 0：无作用； 1：制造一个溢出事件。

14.5.4.7 TMR1通道模式寄存器1（TMR1_CM1）

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 CxC 位定义。该寄存器其它位的作用在输入和输出模式下不同。CxOx 描述了通道在输出模式下的功能，CxIx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

输出比较模式

域	简称	复位值	类型	功能
位 15	C2OSEN	0x0	rw	通道 2 输出开关使能（Channel 2 output switch enable）
位 14：12	C2OCTRL	0x0	rw	通道 2 输出控制（Channel 2 output control）
位 11	C2OBEN	0x0	rw	通道 2 输出缓存使能（Channel 2 output buffer enable）
位 10	C2OIEN	0x0	rw	通道 2 输出立即使能（Channel 2 output immediately enable）
位 9：8	C2C	0x0	rw	通道 2 配置（Channel 2 configure） 当 C2EN=’0’时，这些位用于选择通道 2 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C2IN 映射在 C2IFP2 上； 10：输入，C2IN 映射在 C1IFP2 上； 11：输入，C2IN 映射在 STI 上，只有在 STIS 选择内部触发输入时才工作。
位 7	C1OSEN	0x0	rw	通道 1 输出开关使能（Channel 1 output switch enable） 0：EXT 输入不影响 C1ORAW； 1：当 EXT 输入高电平时，将 C1ORAW 清 0。
位 6：4	C1OCTRL	0x0	rw	通道 1 输出控制（Channel 1 output control） 这些位用于设置原始信号 C1ORAW 的工作状态。

				000：断开。断开 C1ORAW 到 C1OUT 的输出； 001：设置 C1ORAW 为高：TMR1_CVAL=TMR1_C1DT 时。 010：设置 C1ORAW 为低：TMR1_CVAL=TMR1_C1DT 时。 011：切换 C1ORAW 的电平：当 TMR1_CVAL=TMR1_C1DT 时。 100：固定 C1ORAW 为低。 101：固定 C1ORAW 为高。 110：PWM 模式 A －OWCDIR=0，若 TMR1_C1DT>TMR1_CVAL 时设置 C1ORAW 为高，否则为低； －OWCDIR=1，若 TMR1_ C1DT <TMR1_CVAL 时设置 C1ORAW 为低，否则为高。 111：PWM 模式 B －OWCDIR=0，若 TMR1_ C1DT >TMR1_CVAL 时设置 C1ORAW 为低，否则为高； －OWCDIR=1，若 TMR1_ C1DT <TMR1_CVAL 时设置 C1ORAW 为高，否则为低。 注：除'000'外，其余配置下 C1OUT 将连接到 C1ORAW，C1OUT 的输出电平除了会根据 C1ORAW 变化外，还与 CCTRL 所配置的输出极性有关。
位 3	C1OBEN	0x0	rw	通道 1 输出缓存使能（Channel 1 output buffer enable） 0：关闭 TMR1_C1DT 的缓存功能，写入 TMR1_C1DT 的内容会立即生效。 1：启用 TMR1_C1DT 的缓存功能，写入 TMR1_C1DT 的内容将保存到缓存寄存器中，当发生溢出事件时再更新到 TMR1_C1DT 中。
位 2	C1OIEN	0x0	rw	通道 1 输出立即使能（Channel 1 output immediately enable） 在 PWM 模式 A 或模式 B 下，该位能够缩短触发事件到通道 1 的输出响应间的时间。 0：需要比较 CVAL 与 C1DT 的值之后再产生输出。 1：无需比较 CVAL 与 C1DT 的值，当发生触发事件时立即产生输出。
位 1：0	C1C	0x0	rw	通道 1 配置 （Channel 1 configure） 当 C1EN='0'时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C1IN 映射在 C1IFP1 上； 10：输入，C1IN 映射在 C2IFP1 上； 11：输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

输入模式

域	简称	复位值	类型	功能
位 15：12	C2DF	0x0	rw	通道 2 滤波器 （Channel 2 digital filter）
位 11：10	C2IDIV	0x0	rw	通道 2 分频系数 （Channel 2 input divider）
位 9：8	C2C	0x0	rw	通道 2 配置（Channel 2 configure） 当 C2EN='0'时，这些位用于选择通道 2 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C2IN 映射在 C2IFP2 上； 10：输入，C2IN 映射在 C1IFP2 上； 11：输入，C2IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。
位 7：4	C1DF	0x0	rw	通道 1 滤波器 （Channel 1 digital filter） 这些位用于配置通道 1 的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器： 0000：无滤波器，以f_{DTS}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 $f_{SAMPLING}=f_{DTS}/2$，N=6 0101：采样频率 $f_{SAMPLING}=f_{DTS}/2$，N=8 0110：采样频率 $f_{SMPLING}=f_{DTS}/4$，N=6 0111：采样频率 $f_{SAMPLING}=f_{DTS}/4$，N=8 1000：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=6 1001：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=8 1010：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=5 1011：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=6 1100：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=8 1101：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=5 1110：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=6 1111：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=8
位 3：2	C1IDIV	0x0	rw	通道 1 分频系数 （Channel 1 input divider） 这些位定义了通道 1 的分频系数。 00：不分频，每一个有效的边沿都会产生一次输入； 01：每 2 个有效的边沿产生一次输入； 10：每 4 个有效的边沿产生一次输入； 11：每 8 个有效的边沿产生一次输入。 注：C1EN=’0’时，分频系数复位。
位 1：0	C1C	0x0	rw	通道 1 配置 （Channel 1 configure） 当 C1EN=’0’时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C1IN 映射在 C1IFP1 上； 10：输入，C1IN 映射在 C2IFP1 上； 11：输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

14.5.4.8 TMR1通道模式寄存器2（TMR1_CM2）

参看以上 CM1 寄存器描述

输出比较模式

域	简称	复位值	类型	功能
位 15	C4OSEN	0x0	rw	通道 4 输出开关使能（Channel 4 output switch enable）
位 14：12	C4OCTRL	0x0	rw	通道 4 输出控制（Channel 4 output control）
位 11	C4OBEN	0x0	rw	通道 4 输出缓存使能（Channel 4 output buffer enable）
位 10	C4OIEN	0x0	rw	通道 4 输出立即使能（Channel 4 output immediately enable）
位 9：8	C4C	0x0	rw	通道 4 配置（Channel 4 configure） 当 C4EN=’0’时，这些位用于选择通道 4 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C4IN 映射在 C4IFP4 上； 10：输入，C4IN 映射在 C3IFP4 上； 11：输入，C4IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。
位 7	C3OSEN	0x0	rw	通道 3 输出开关使能（Channel 3 output switch enable）
位 6：4	C3OCTRL	0x0	rw	通道 3 输出控制（Channel 3 output control）
位 3	C3OBEN	0x0	rw	通道 3 输出缓存使能（Channel 3 output buffer enable）
位 2	C3OIEN	0x0	rw	通道 3 输出立即使能（Channel 3 output immediately enable）
位 1：0	C3C	0x0	rw	通道 3 配置（Channel 3 configure）

				当 C3EN=’0’时，这些位用于选择通道 3 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C3IN 映射在 C3IFP3 上； 10：输入，C3IN 映射在 C4IFP3 上； 11：输入，C3IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

输入模式

域	简称	复位值	类型	功能
位 15：12	C4DF	0x0	rw	通道 4 滤波器 （Channel 4 digital filter）
位 11：10	C4IDIV	0x0	rw	通道 4 分频系数 （Channel 4 input divider）
位 9：8	C4C	0x0	rw	通道 4 配置（Channel 4 configure） 当 C4EN=’0’时，这些位用于选择通道 4 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C4IN 映射在 C4IFP4 上； 10：输入，C4IN 映射在 C3IFP4 上； 11：输入，C4IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。
位 7：4	C3DF	0x0	rw	通道 3 滤波器 （Channel 3 digital filter）
位 3：2	C3IDIV	0x0	rw	通道 3 分频系数 （Channel 3 input divider）
位 1：0	C3C	0x0	rw	通道 3 配置（Channel 3 configure） 当 C3EN=’0’时，这些位用于选择通道 3 为输出或输入，以及输入时的映射选择： 00：输出； 01：输入，C3IN 映射在 C3IFP3 上； 10：输入，C3IN 映射在 C4IFP3 上； 11：输入，C3IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

14.5.4.9 TMR1通道控制寄存器（TMR1_CCTRL）

域	简称	复位值	类型	功能
位 15：14	保留	0x0	resd	保持默认值。
位 13	C4P	0x0	rw	通道 4 极性 （Channel 4 polarity） 见 C1P 的描述。
位 12	C4EN	0x0	rw	通道 4 使能 （Channel 4 enable） 见 C1EN 的描述。
位 11	C3CP	0x0	rw	通道 3 互补极性 （Channel 3 complementary polarity） 见 C1P 的描述。
位 10	C3CEN	0x0	rw	通道 3 互补使能 （Channel 3 complementary enable） 见 C1EN 的描述。
位 9	C3P	0x0	rw	通道 3 极性 （Channel 3 polarity） 见 C1P 的描述。
位 8	C3EN	0x0	rw	通道 3 使能 （Channel 3 enable） 见 C1EN 的描述。
位 7	C2CP	0x0	rw	通道 2 互补极性 （Channel 2 complementary polarity） 见 C1P 的描述。
位 6	C2CEN	0x0	rw	通道 2 互补使能 （Channel 2 complementary enable） 见 C1EN 的描述。
位 5	C2P	0x0	rw	通道 2 极性 （Channel 2 polarity） 见 C1P 的描述。
位 4	C2EN	0x0	rw	通道 2 使能 （Channel 2 enable） 见 C1EN 的描述。
位 3	C1CP	0x0	rw	通道 1 互补极性 （Channel 1 complementary polarity） 0：C1COUT 的有效电平为高 1：C1COUT 的有效电平为低
位 2	C1CEN	0x0	rw	通道 1 互补使能 （Channel 1 complementary enable） 0：禁止输出；

				1：使能输出。
位 1	C1P	0x0	rw	通道 1 极性 （Channel 1 polarity） 通道 1 配置为输出： 0：C1OUT 的有效电平为高 1：C1OUT 的有效电平为低 通道 1 配置为输入： C1CP/C1P 位共同定义输入信号有效沿。 00：C1IN 的有效边沿为上升沿；作为外部触发使用时，C1IN 不反相。 01：C1IN 的有效边沿为下降沿；作为外部触发使用时，C1IN 反相。 10：保留 11：C1IN 的有效边沿为上升沿和下降沿；作为外部触发使用时，C1IN 不反相。
位 0	C1EN	0x0	rw	通道 1 使能 （Channel 1 enable） 0：禁止输入或输出； 1：使能输入或输出。

表 14-14 带停止功能的互补输出通道CxOUT和CxCOUT的控制位

控制位					输出状态（1）	
OEN 位	FCSODIS 位	FCSOEN 位	CxEN 位	CxCEN 位	CxOUT 输出状态	CxCOUT 输出状态
1	X	0	0	0	输出禁止（与定时器断开） CxOUT=0，Cx_EN=0	输出禁止（与定时器断开） CxCOUT=0，CxCEN=0
		0	0	1	输出禁止（与定时器断开） CxOUT=0，Cx_EN=0	CxORAW + 极性， CxCOUT= CxORAW xor CxCP， CxCEN=1
		0	1	0	CxORAW+极性， CxOUT= CxORAW xor CxP， Cx_EN=1	输出禁止（与定时器断开） CxCOUT=0，CxCEN=0
		0	1	1	CxORAW+极性+死区， Cx_EN=1	CxORAW 反相+极性+死区， CxCEN=1
		1	0	0	输出禁止（与定时器断开） CxOUT=CxP，Cx_EN=0	输出禁止（与定时器断开） CxCOUT=CxCP，CxCEN=0
		1	0	1	关闭状态 （输出使能且为无效电平） CxOUT=CxP，Cx_EN=1	CxORAW + 极性， CxCOUT= CxORAW xor CxCP，CxCEN=1
		1	1	0	CxORAW + 极性， CxOUT= CxORAW xor CxP， Cx_EN=1	关闭状态 （输出使能且为无效电平） CxCOUT=CxCP，CxCEN=1
		1	1	1	CxORAW+极性+死区， Cx_EN=1	CxORAW 反相+极性+死区， CxCEN=1
0	0	X	0	0	输出禁止（对应 IO 与定时器断开，IO 浮空） 异步地： CxOUT=CxP ，Cx_EN=0 ，CxCOUT=CxCP ，CxCEN=0； 若时钟存在：经过一个死区时间后 CxOUT=CxIOS，CxCOUT=CxCIOS，假设 CxIOS 与 CxCIOS 并不都对应 CxOUT 和 CxCOUT 的有效电平。	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0	CxEN=CxCEN=0 时：输出禁止（对应 IO 与定时器断开，IO 浮空）； 其它情况下：关闭状态（对应通道输出无效电平） 异步地：CxOUT =CxP，Cx_EN=1，CxCOUT=CxCP ，CxCEN=1；	
	1		0	1		
	1		1	0		

	1		1	1	若时钟存在：经过一个死区 时间后 CxOUT =C1IOS，CxCOUT=CxCIOS，假设 CxIOS 与 CxCIOS 并不都对应 CxOUT 和 CxCOUT 的有效电平。

注意：如果一个通道的 2 个输出都没有使用（CxEN = CxCEN = 0），那么 CxIOS，CxCIOS，CxP 和 CxCP 都必须清零。

注意：引脚连接到互补的 CxOUT 和 CxCOUT 通道的外部 I/O 引脚的状态，取决于 CxOUT 和 CxCOUT 通道状态和 GPIO 以及 IOMUX 寄存器。

14.5.4.10 TMR1计数值（TMR1_CVAL）

域	简称	复位值	类型	功能
位 15：0	CVAL	0x0000	rw	计数值 （Counter value）

14.5.4.11 TMR1预分频器（TMR1_DIV）

域	简称	复位值	类型	功能
位 15：0	DIV	0x0000	rw	分频系数 （Divider value） 计数器时钟频率 $f_{CK\_CNT} = f_{TMR\_CLK}$ /（DIV[15：0]+1） 溢出事件发生时该寄存器值被传送到实际的预分频寄存器中。

14.5.4.12 TMR1周期寄存器（TMR1_PR）

域	简称	复位值	类型	功能
位 15：0	PR	0x0000	rw	周期值 （Period value） 定时器计数的周期值。当周期值为 0 时，定时器不工作。

14.5.4.13 TMR1重复周期寄存器（TMR1_RPR）

域	简称	复位值	类型	功能
位 15：0	RPR	0x00	rw	重复周期的次数（Repetition of period value） 这些位用于减慢溢出事件发生的速率，当重复周期的次数减为 0 时才会发生溢出事件。

14.5.4.14 TMR1通道1数据寄存器（TMR1_C1DT）

域	简称	复位值	类型	功能
位 15：0	C1DT	0x0000	rw	通道 1 数据寄存器值（Channel 1 data register） 若通道 1 配置为输入： C1DT 是前一次通道 1 输入事件（C1IN）所保存的 CVAL。 若通道 1 配置为输出： C1DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C1OBEN），并根据设置在 C1OUT 上产生相应的输出。

14.5.4.15 TMR1通道2数据寄存器（TMR1_C2DT）

域	简称	复位值	类型	功能
位 15：0	C2DT	0x0000	rw	通道 2 数据寄存器值（Channel 2 data register） 若通道 2 配置为输入： C2DT 是前一次通道 2 输入事件（C2IN）所保存的 CVAL。 若通道 2 配置为输出： C2DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C2OBEN），并根据设置在 C2OUT 上产生相应的输出。

14.5.4.16 TMR1通道3数据寄存器（TMR1_C3DT）

域	简称	复位值	类型	功能
位 15：0	C3DT	0x0000	rw	通道 3 数据寄存器值（Channel 3 data register） 若通道 3 配置为输入： C3DT 是前一次通道 3 输入事件（C3IN）所保存的 CVAL。 若通道 3 配置为输出：

				C3DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C3OBEN），并根据设置在 C3OUT 上产生相应的输出。

14.5.4.17 TMR1通道4数据寄存器（TMR1_C4DT）

域	简称	复位值	类型	功能
位 15：0	C4DT	0x0000	rw	通道 4 数据寄存器值（Channel 4 data register） 若通道 4 配置为输入： C4DT 是前一次通道 4 输入事件（C4IN）所保存的 CVAL。 若通道 4 配置为输出： C4DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C4OBEN），并根据设置在 C4OUT 上产生相应的输出。

14.5.4.18 TMR1刹车寄存器（TMR1_BRK）

域	简称	复位值	类型	功能
位 19：16	BKF	0x0	rw	刹车输入滤波（brake input filter） 这些位用于配置刹车输入的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器： 0000：无滤波器，以f_{DTS}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 $f_{SAMPLING}$=f_{DTS}/2，N=6 0101：采样频率 $f_{SAMPLING}$=f_{DTS}/2，N=8 0110：采样频率 $f_{SMPLING}$=f_{DTS}/4，N=6 0111：采样频率 $f_{SAMPLING}$=f_{DTS}/4，N=8 1000：采样频率$f_{SAMPLING}$=f_{DTS}/8，N=6 1001：采样频率$f_{SAMPLING}$=f_{DTS}/8，N=8 1010：采样频率$f_{SAMPLING}$=f_{DTS}/16，N=5 1011：采样频率$f_{SAMPLING}$=f_{DTS}/16，N=6 1100：采样频率$f_{SAMPLING}$=f_{DTS}/16，N=8 1101：采样频率$f_{SAMPLING}$=f_{DTS}/32，N=5 1110：采样频率$f_{SAMPLING}$=f_{DTS}/32，N=6 1111：采样频率$f_{SAMPLING}$=f_{DTS}/32，N=8
位 15	OEN	0x0	rw	输出使能（Output enable） 对配置为输出的通道，该位用于使能 CxOUT 和 CxCOUT 的输出。 0：关闭； 1：开启。
位 14	AOEN	0x0	rw	输出自动使能（Automatic output enable） 用于溢出事件时将 OEN 自动置'1' 0：关闭； 1：开启
位 13	BRKV	0x0	rw	刹车输入信号的有效性（Brake input validity） 用于选择刹车输入信号的输入有效电平： 0：低电平； 1：高电平。
位 12	BRKEN	0x0	rw	刹车功能使能（Brake enable） 用于开启刹车功能。 0：关闭； 1：开启。
位 11	FCSOEN	0x0	rw	总输出开时的冻结状态（Frozen channel status when holistic output enable） 该位用于配置具有互补输出的通道，在定时器不工作且 OEN=1 时的通道状态。 0：关闭 CxOUT/CxCOUT 输出； 1：开启 CxOUT/CxCOUT 输出，输出为无效电平。
位 10	FCSODIS	0x0	rw	总输出关时的冻结状态（Frozen channel status when holistic output disable）

				该位用于配置具有互补输出的通道，在定时器不工作且OEN=0 时的通道状态。 0 ：关闭 CxOUT/CxCOUT 输出； 1 ：开启 CxOUT/CxCOUT 输出，输出为空闲电平。
位 9：8	WPC	0x0	rw	写保护配置（Write protected configuration） 该位用于配置写保护。 00：写保护关闭； 01：3 级写保护，以下位受写保护： TMR1_STOP：DTC、STPEN、STPV 和 HOAEN TMR1_CTRL2：CxIOS 和 CxIOSL 10：2 级写保护，除 3 级写保护的内容外，以下位也受写保护： TMR1_CCTRL：CxP 和 CxLP TMR1_STOP：FCSODIS 和 FCSOEN 11：1 级写保护，除 2 级写保护的内容外，以下位也受写保护： TMR1_CMx：C2OCTRL 和 C2OBEN 注：WPC>0 时将无法再次被修改，直到系统复位。
位 7：0	DTC	0x00	rw	死区配置 （Dead-time configuration） 这些位用于配置死区时间。取 DTC[7：0]的高 3 位为功能选择位： 0xx：DT = DTC [7：0] * TDTS； 10x：DT =（64+ DTC [5：0]）* TDTS * 2； 110：DT =（32+ DTC [4：0]）* TDTS * 8； 111：DT =（32+ DTC [4：0]）* TDTS * 16；

注意：根据锁定设置，AOEN、BRKV、BRKEN、FCSODIS、FCSOEN 和 DTC[7：0]位均可被写保护，有必要在第一次写入 TMR1_BRK 寄存器时对它们进行配置。

14.5.4.19 TMR1 DMA控制寄存器（TMR1_DMACTRL）

域	简称	复位值	类型	功能
位 15：13	保留	0x0	resd	保持默认值。
位 12：8	DTB	0x00	rw	DMA 传输字节（DMA transfer bytes） 这些位定义了传输的字节个数： 00000：1 个字节 00001：2 个字节 00010：3 个字节 00011：4 个字节 .... 10000：17 个字节 10001：18 个字节
位 7：5	保留	0x0	resd	保持默认值。
位 4：0	ADDR	0x00	rw	DMA 传输地址偏移 （DMA transfer address offset） ADDR 定义了从 TMR1_CTRL1 所在地址开始的偏移量： 00000：TMR1_CTRL1， 00001：TMR1_CTRL2， 00010：TMR1_STCTRL， .....

14.5.4.20 TMR1 DMA数据寄存器（TMR1_DMADT）

域	简称	复位值	类型	功能
位 15：0	DMADT	0x0000	rw	DMA 传输的数据寄存器（DMA data register） 通过对 DMADT 寄存器的读写能够实现对任意 TMR 寄存器的操作，其操作的寄存器地址范围是： TMR1 外设地址 + ADDR*4 至 TMR1 外设地址 + ADDR*4 + DTB*4。

15 窗口看门狗（WWDT）

15.1 WWDT简介

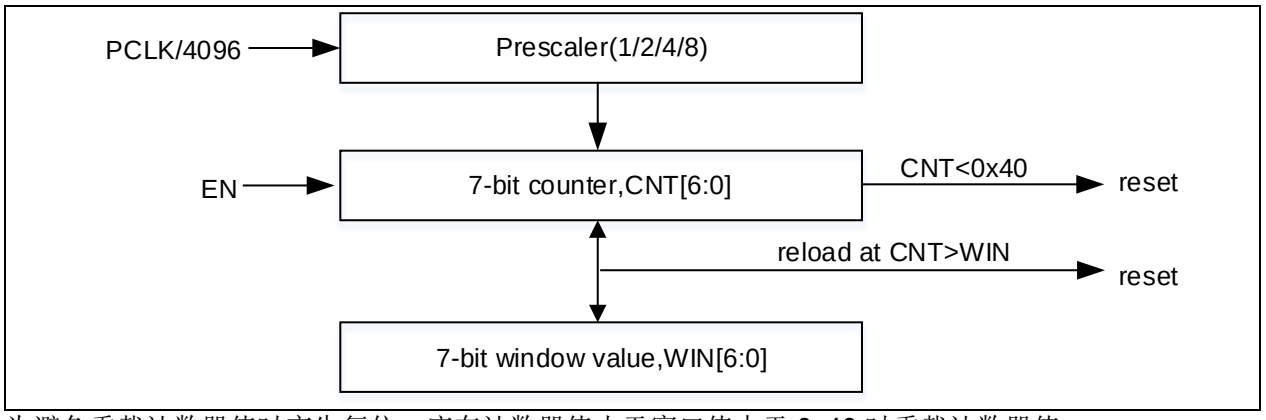
当程序正常运行时，需在一个有限的时间窗口内重载窗口看门狗递减计数器，用来避免看门狗电路产生系统复位，以此来监测系统是否正常运行。
窗口看门狗时钟由 APB1_CLK 分频而来，由于 APB1_CLK 的精确性，窗口看门狗可对有限的时间窗口精确控制。

15.2 WWDT主要特性

- 7位递减计数器
- 启动看门狗后，当递减计数器的值小于0x40或是在窗口外被重新装载产系统生复位。
- 可以通过重载计数器中断重装载计数器

15.3 WWDT功能描述

启动窗口看门狗后，窗口看门狗可在以下两种情况下产生系统复位：
第一种，7 位递减计数器值由 0x40 变为 0x3F。
第二种，7 位递减计数器值大于 7 位窗口值时，重载计数器值。
图 15 1 窗口看门狗框图



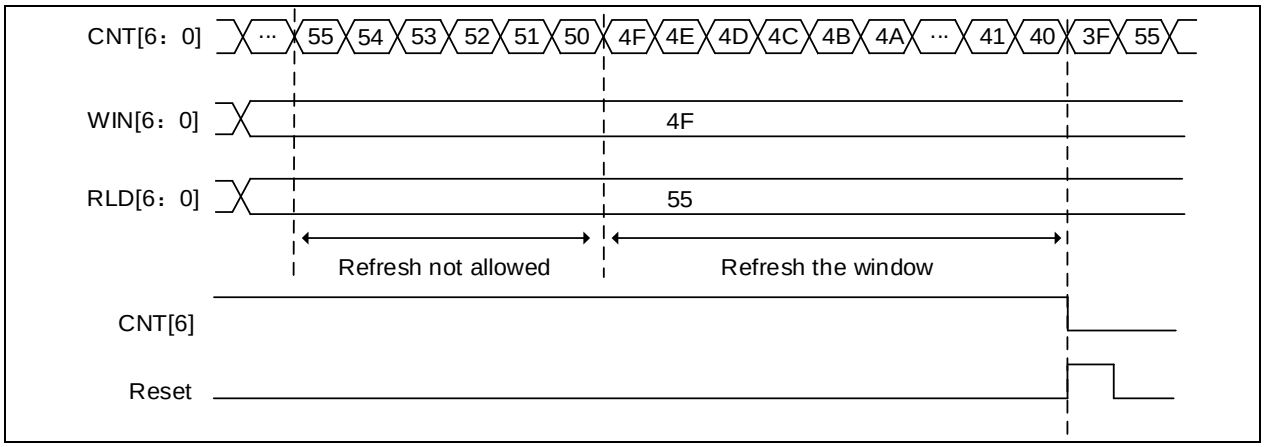
为避免重载计数器值时产生复位，应在计数器值小于窗口值大于 0x40 时重载计数器值。
WWDT 计数器时钟由 APB1_CLK 分频得到，分频系数可通过配置 WWDT_CFG 寄存器 DIV[1：0]改变。
计数器值决定了 WWDT 复位前的最大计数周期数，结合 WIN[6：0]可灵活的调整重载窗口。
WWDT 提供了重载计数器中断功能，开启后，WWDT 将在计数值达到 0x40h 时将 RLDF 标志位置 1，同时产生重载计数器中断，可在中断服务程序中重载计数器值，以避免发生系统复位。需要注意的是，若在 CNT[6]为 0 时，将 WWDTEN 置 1 会产生一个系统复位，因此当写入 WWDT_CTRL 寄存器时，应始终保持 CNT[6]为 1，避免使能窗口看门狗后立即产生一个系统复位。
窗口看门狗超时时间 T_{WWDT} 可由一下公式计算，其中 T_{PCLK1} 为 APB1 时钟周期，单位为 ms：

$$T_{WWDT} = T_{PCLK1} \times 4096 \times 2^{DIV[1:0]} \times （CNT[5:0] + 1）；\quad （ms）$$

表 15-1 给出了当 PCLK1 频率为 72MHz 时，最大和最小看门狗超时时间。
表 15-1 PCLK1频率为72MHz时，最大和最小看门狗超时时间

时钟预分频值	最小超时时间	最大超时时间
0	56.5 μs	3.64ms
1	113.5 μs	7.28ms
2	227.5 μs	14.56ms
3	455 μs	29.12ms

图 15-1 窗口看门狗时序图



15.4 调试模式

微控制器处于调试模式时，意味着 Cortex®-M4F 核心停止。将 DEBUG 模块中 WWDT_PAUSE 位置 1 可将 WWDT 计数器计数暂停。

15.5 WWDT寄存器

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

寄存器简称	基址偏移量	复位值
WWDT_CTRL	0x00	0x7F
WWDT_CFG	0x04	0x7F
WWDT_STS	0x08	0x00

15.5.1 控制寄存器（WWDT_CTRL）

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	resd	保持默认值。
位 7	WWDTEN	0x0	rw1s	窗口看门狗使能 （Window watchdog enable） 0：关闭； 1：开启。 该位由软件置起，只能在复位后自动清零。
位 6：0	CNT	0x7F	rw	递减计数器（Decrement counter） 当计数器递减到 0x3F 时产生复位。

15.5.2 配置寄存器（WWDT_CFG）

域	简称	复位值	类型	功能
位 31：10	保留	0x000000	resd	保持默认值。
位 9	RLDIEN	0x0	rw	重载计数器中断（Reload counter interrupt） 0：关闭； 1：开启。
位 8：7	DIV	0x0	rw	时钟预分频值（Clock division value） 00：PCLK1 除以 4096； 01：PCLK1 除以 8192； 10：PCLK1 除以 16384； 11：PCLK1 除以 32768。
位 6：0	WIN	0x7F	rw	窗口值（Window value） 当计数器值大于窗口值时，此时重载计数器会产生复位，重载计数器区间为 0x40~WIN[6：0]

15.5.3 状态寄存器（WWDT_STS）

域	简称	复位值	类型	功能
位 31：1	保留	0x0000 0000	resd	保持默认值。

位 0	RLDF	0x0	rw0c	重载计数器中断标志（Reload counter interrupt flag） 当递减计数器为 0x40 时，该标志会置位。 该位被硬件置起，由软件将其清零。

16 看门狗（WDT）

16.1 WDT简介

看门狗由专用低速时钟（LICK）驱动，由于 LICK 时钟精度较低，因此看门狗适用于低时间精度、能够独立于主程序之外的应用

16.2 WDT主要特性

- 12位递减计数器
- 计数器由LICK时钟驱动（可在深睡眠和待机模式下工作）
- 可选择在DEEPSLEEP、STANDBY模式下是否停止计数
- 支持两种复位方式：
  - 当递减计数器递减至0。
  - 当递减计数器在窗口外被重新装载。

16.3 WDT功能描述

WDT 启动方式：

WDT 的启动方式有两种，分别为软件启动和硬件启动。软件启动通过向 WDT_CMD 寄存器写入 0xCCCC 实现；硬件启动则需通过配置用户系统数据区来实现，使能硬件看门狗后，看门狗将在上电复位后自动开始运行。

WDT 复位条件：

当 WDT 计数器值递减至 0 时将产生 WDT 系统复位，因此需定时向 WDT_CMD 寄存器写入 0xAAAA 重载计数器值。此外，若将 WIN[11：0]设置为非默认值（0xFFF）将开启窗口看门狗功能，在计数值大于窗口值时重载计数器值将会产生系统复位。

WDT 写保护：

DT_DIV、WDT_RLD、WDT_WIN 寄存器受写保护，向 WDG_CMD 寄存器写入 0x5555 可解锁寄存器写保护，之后可对其进行配置。这三个寄存器的更新状态分别由 WDT_STS 寄存器中 DIVF、RLDF、WINF 指示。向 WDG_CMD 寄存器写入其它值将重新启动 WDT_DIV、WDT_RLD、WDT_WIN 寄存器写保护。向 WDG_CMD 寄存器写入 0xAAAA 也会启动寄存器写保护。

WDT 时钟：

WDT 计数器由 LICK 时钟驱动，LICK 是内部 RC 时钟，范围为 30kHz~60kHz 之间，所以超时时间也是在一定区间内，使用时应注意在超时时间配置上应该留有余量，如果需要获得较为精确的看门狗超时时间，可对 LICK 进行校准，有关 LICK 校准的问题，详见 4.1.1 节。

WDT 低功耗计数模式：

WDT 能够在 SLEEP、DEEPSLEEP、STANDBY 模式下运行，用户可选择进入 DEEPSLEEP、STANDBY 模式后计数器是否停止计数，可由用户系统数据区中的 nDEPSLP_WDT、nSTDBY_WDT 位配置。

如果设置了停止计数，当进入了 DEEPSLEEP、STANDBY 模式后，看门狗计数器停止递减，意味着看门狗在这两种低功耗模式下不会发生复位，当从这两种模式唤醒后，计数器从进入时的值继续递减。

图 16-1看门狗框图

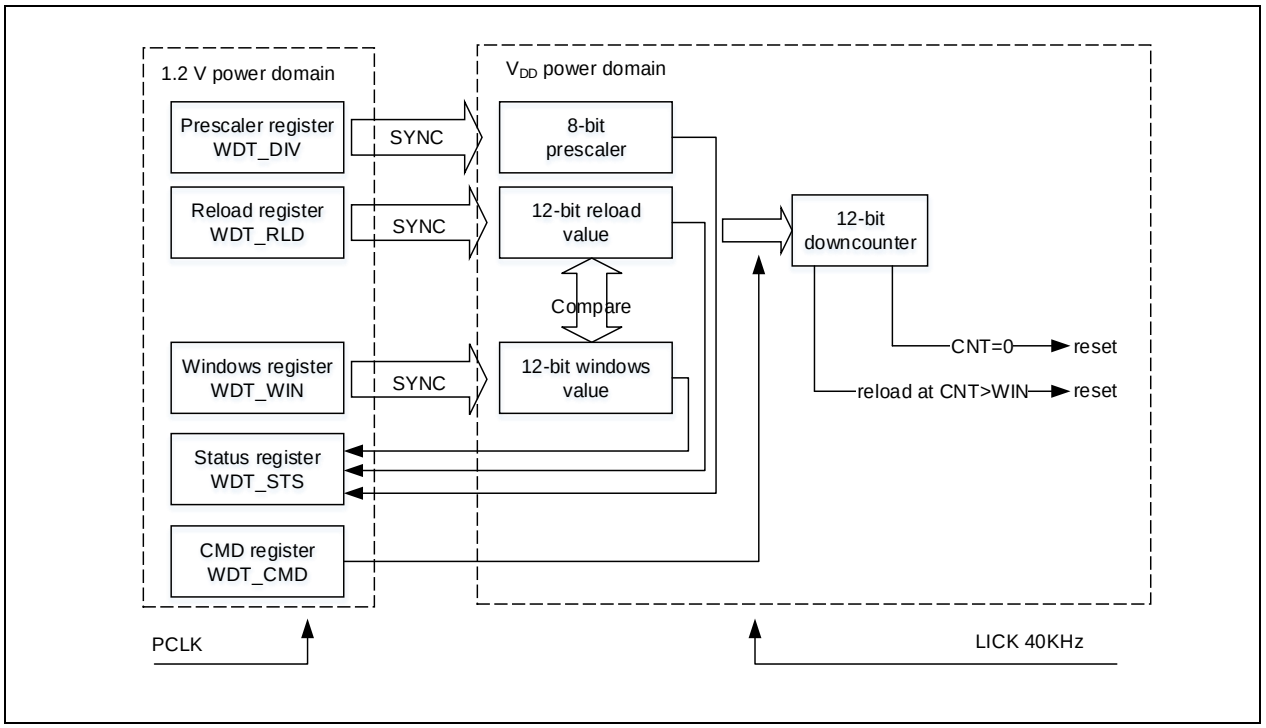


表 16-1看门狗超时时间（LICK=40kHz）

预分频系数	DIV[2：0]位	最短时间（ms） RLD[11：0] = 0x000	最长时间（ms） RLD[11：0] = 0xFFF
/4	0	0.1	409.6
/8	1	0.2	819.2
/16	2	0.4	1638.4
/32	3	0.8	3276.8
/64	4	1.6	6553.6
/128	5	3.2	13107.2
/256	（6 或 7）	6.4	26214.4

16.4调试模式

微控制器处于调试模式时，意味着 Cortex®-M4F 核心停止。此时将 DEBUG 模块中 WDT_PAUSE 位置 1 会暂停 WDT 计数器计数。详见 25.2 节

16.5WDT寄存器

必须以字（32 位）的方式操作这些外设寄存器。

表 16-2 WDT寄存器映像和复位值

寄存器简称	基址偏移量	复位值
WDT_CMD	0x00	0x0000 0000
WDT_DIV	0x04	0x0000 0000
WDT_RLD	0x08	0x0000 0FFF
WDT_STS	0x0C	0x0000 0000
WDT_WIN	0x10	0x0000 0FFF

16.5.1 命令寄存器（WDT_CMD）

（在待机模式复位）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持默认值。
位 15：0	CMD	0x0000	wo	命令寄存器（Command register） 0xAAAA：重载计数器； 0x5555：解锁 WDT_DIV 和 WDT_RLD、WDT_WIN 写保护； 0xCCCC：启动看门狗，如果使能了硬件看门狗，则不需要执行此操作。

16.5.2 预分频寄存器（WDT_DIV）

（待机模式时不复位）

域	简称	复位值	类型	功能
位 31：3	保留	0x0000 0000	resd	保持默认值。
位 2：0	DIV	0x0	rw	递减计数器时钟预分频值（Clock division value） 000：LICK 除以 4； 001：LICK 除以 8； 010：LICK 除以 16； 011：LICK 除以 32； 100：LICK 除以 64； 101：LICK 除以 128； 110：LICK 除以 256； 111：LICK 除以 256。 只有解锁写保护后才能写此寄存器，只有当 DIVF 为 0 时，才能读取此寄存器。

16.5.3 重装载寄存器（WDT_RLD）

（待机模式时不复位）

域	简称	复位值	类型	功能
位 31：12	保留	0x00000	resd	保持默认值。
位 11：0	RLD	0xFFF	rw	重载值（Reload value） 只有解锁写保护后才能写此寄存器，只有当 RLDF 为 0 时，才能读取此寄存器。

16.5.4 状态寄存器（WDT_STS）

（待机模式复位）

域	简称	复位值	类型	功能
位 31：3	保留	0x0000 0000	resd	保持默认值。
位 2	WINF	0x0	ro	窗口值更新完成标志（Window value update complete flag） 0：更新完成； 1：正在更新。 只有当 RLDF 为 0 时才能写 WDT_WIN 寄存器。
位 1	RLDF	0x0	ro	重载值更新完成标志（Reload value update CMPlete flag） 0：更新完成； 1：正在更新。 只有当 RLDF 为 0 时才能写 WDT_RLD 寄存器。
位 0	DIVF	0x0	ro	分频值更新完成标志（Division value update CMPlete flag） 0：更新完成； 1：正在更新。 只有当 DIVF 为 0 时才能写 WDT_DIV 寄存器。

16.5.5 窗口寄存器（WDT_WIN）

（待机模式不复位）

域	简称	复位值	类型	功能
位 31：12	保留	0x00000	resd	保持默认值。
位 11:0	WIN	0xFFF	rw	窗口值（Window value） 当计数器值大于窗口值时，此时重载计数器会产生复位，重载计数器区间为 0~窗口值。

17 实时时钟（ERTC）

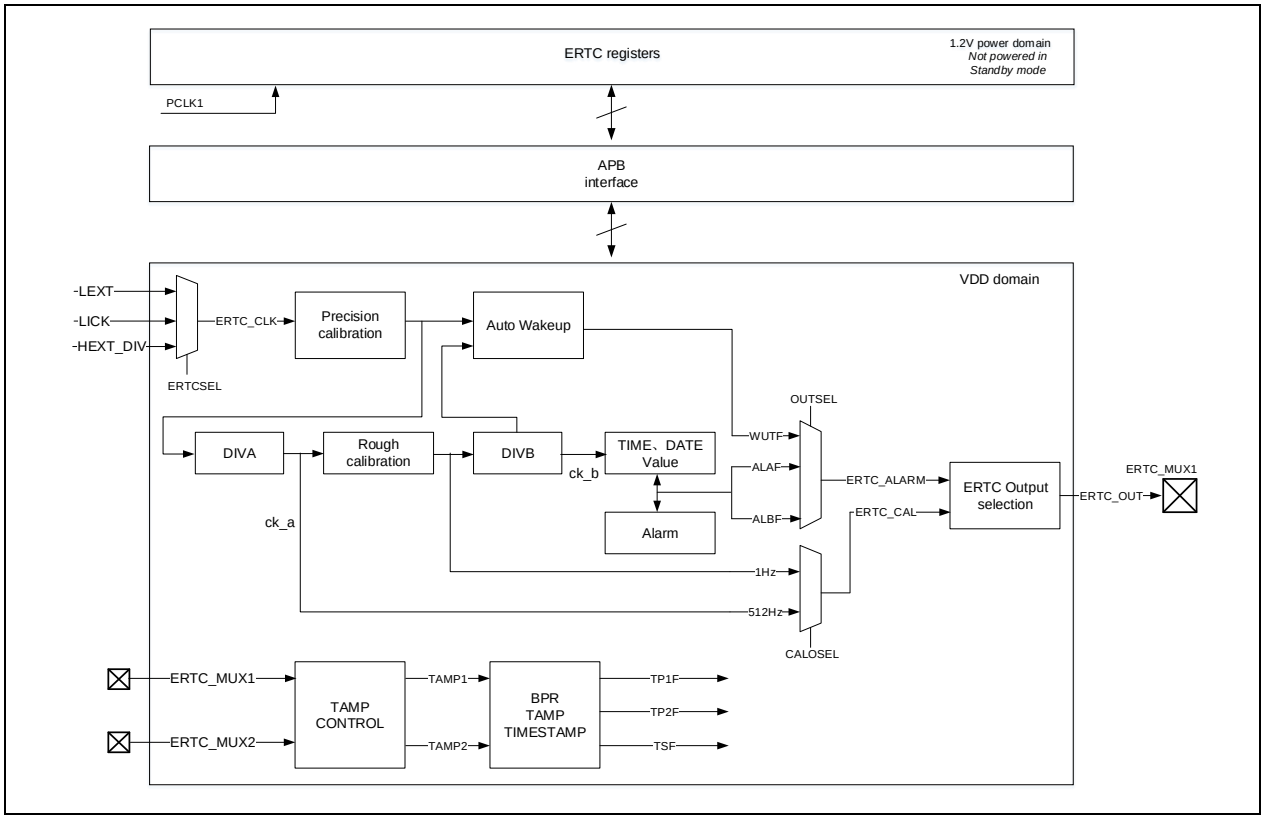
17.1ERTC简介

实时时钟用于配置日历时钟，修改 ERTC 中日历寄存器值可以修改系统的当前时间和日期。
ERTC 计数逻辑位于电池供电域，只要电池供电域有电（仅支持由 VDD 域供电），ERTC 便会一直运行，不受系统复位影响。

17.2ERTC主要特性

- 功能强大的实时日历，自动处理月份天数28（平年2月）、29（闰年2月）、30（小月）、31（大月），其中当年份寄存器是4的倍数时为闰年，支持两组闹钟
- 周期性唤醒
- 参考时钟检测
- 两组可配置入侵检测，支持时间戳功能
- 支持精密校准
- 20个电池供电寄存器
- 5组中断：闹钟A、闹钟B、周期性唤醒、入侵检测、时间戳
- 复用功能输出，校准时钟输出、闹钟事件或唤醒事件
- 复用功能输入，参考时钟输入、两路入侵检测、时间戳

图 17-1 ERTC框图



17.3ERTC功能说明

17.3.1 ERTC时钟

ERTC_CLK 可从 LEXT、LICK、分频后的 HEXT 中选择（由 CRM_BPDC 寄存器 ERTCSEL[1：0]配置），HEXT 的分频值由 CRM_CFG 寄存器 ERTC_DIV[4：0]配置。
ERTC 内置分频器 A 和分频器 B，分别由 DIVA[6：0]、DIVB[14：0]配置，推荐 DIVA 配置为较高的值，以最大程度降低功耗。ERTC_CLK 依次经由分频器 A、分频器 B 处理，得到 ck_a、ck_b 时钟，ck_a 用于更新亚秒，ck_b 用于更新日历和周期性唤醒。ck_a、ck_b 时钟频率可由下式计算：

$$f_{ck\_a} = \frac{f_{ERTC\_CLK}}{DIVA + 1}$$

$$f_{ck\_b} = \frac{f_{ERTC\_CLK}}{(DIVB + 1) \times (DIVA + 1)}$$

当配置 DIVA=127，DIVB=255，且 ERTC_CLK 选用 32.768kHz 的 LEXT 时，可得到 1Hz 的 ck_b，用于于更新日历。

注意：若 ERTC_CLK 选用分频后的 HEXT，应先配置 HEXT 分频值，再将时钟源切换为 HEXT。

17.3.2 ERTC初始化

寄存器解锁：

上电复位后所有 ERTC 寄存器处于写保护状态，需要先解除写保护，才能写配置 ERTC 寄存器（除 ERTC_STS[14：8]、ERTC_TAMP 和 ERTC_BPRx 寄存器外，其它寄存器位均受写保护，且写保护不受系统复位影响）。

解锁步骤：

1、使能电源接口时钟：CRM_APB1EN 的 PWCEN=1

2、解锁电池供电域写保护：PWC_CTRL 的 BPWEN=1

3、依次向 ERTC_WP 寄存器写入 0xCA，0x53，若向 ERTC_WP 寄存器写入错误的值，将重新激活写保护。

下表列出了需要解除写保护和进入初始化模式才可以配置的 ERTC 寄存器：

表 17-1 ERTC寄存器配置表

寄存器简称	是否受 ERTC_WP 写保护	是否需要进入初始化模式	其它
ERTC_TIME	是	是	-
ERTC_DATE	是	是	-
ERTC_CTRL	是	位 7、6、4 需要	-
ERTC_STS	除位[14：8]外	-	-
ERTC_DIV	是	是	-
ERTC_WAT	是	否	WATWF 为 1 时可配置
ERTC_ALA	是	否	ALAWF 为 1 时可配置
ERTC_ALB	是	否	ALBWF 为 1 时可配置
ERTC_WP	-	-	-
ERTC_SBS	-	-	-
ERTC_TADJ	是	否	TADJF 为 0 时可配置
ERTC_TSTM	-	-	-
ERTC_TSDT	-	-	-
ERTC_TSSBS	-	-	-
ERTC_SCAL	是	否	CALUPDF 为 0 时可配置
ERTC_TAMP	否	否	-
ERTC_ALASBS	是	否	ALAWF 为 1 时可配置
ERTC_ALBSBS	是	否	ALBWF 为 1 时可配置
ERTC_BPRx	否	否	-

时钟、日历初始化：

寄存器解锁后，时钟和日历的初始化配置可按以下步骤进行：

1. 将 IMEN 位置 1 进入初始化模式。
2. 等待初始化标志位 INITF 置 1。
3. 依次配置 DIVB、DIVA。

4. 配置时钟和日历值。
5. 将 IMEN 位清 0 退出初始化模式，等待 UPDF 置 1，表明日历值同步完成，日历开始计数。

为了方便时间微调，ERTC 还提供了夏令时和时间调整功能。
夏令时功能：用于增加（ADD1H=1）或减小（DEC1H=1）1 小时，而无需重新进行初始化配置。
时间调整功能：用于精确的调整当前时钟。若只配置 DECSBS[14：0]值，该值将会加到分频器 B 计数器值中，时钟因此产生延迟；若只将 ADD1S 位置 1，当前时钟将增加 1 秒；若同时配置 DECSBS[14：0]，ADD1S 位，时钟将增加零点几秒。
延迟时间（ADD1S=0）：延迟时间=DECSBS/(DIVB+1)；
提前时间（ADD1S=1）：提前时间=1-(DECSBS/(DIVB+1))。
注：设置时间调整寄存器前，必须先确认 SBS[15]=0，以免亚秒发生上溢。

日历读取：
ERTC 提供两种日历访问方式，分别为同步读取（DREN=0）和异步读取（DREN=1）。
同步读取时，ERTC 通过 PCLK1 访问同步的影子寄存器来获取时钟和日历值。影子寄存器值由位于电池供电域的 ERTC 日历值同步而来，同步周期为两个 ERTC_CLK，同步完成后 UPDF 将置 1。影子寄存器由系统复位来复位。为保证读取的 ERTC_SBS、ERTC_TIME、ERTC_DATE 寄存器值来自同一时刻，读取低阶寄存器时会将高阶寄存器值锁定，直到读取 ERTC_DATE 寄存器。例如读取 ERTC_SBS，会将 ERTC_TIME、ERTC_DATE 寄存器值锁定。
异步读取时，ERTC 通过 PCLK1 直接读取位于电池供电域的 ERTC 时钟和日历值，这样避免了由于同步时间带来的误差。异步读取时，UPDF 标志将由硬件清 0。为保证异步读取时钟和日历值的准确性，软件必须两次读取时钟和日历值，并比较两次结果是否一致，如果不一致应该再读，直到两次结果一致，另外，也可以只比较两次结果的最低位来判定。
注：在 STANDBY 和 DEEPSLEEP 模式下，当前日历值不会复制到影子寄存器中，当从这两种模式唤醒时，必须先设置 UPDF=0，然后等待 UPDF=1，以保证读取的日历值是最新的；在同步读取时，需保证 PCLK1 频率至少为 ERTC_CLK 频率 7 倍；异步读取时，需要额外一个 APB 周期才能完成读取日历寄存器的指令。

闹钟初始化：
ERTC 包含闹钟 A、闹钟 B 两个闹钟，并分别提供了闹钟 A 中断、闹钟 B 中断。
闹钟值由 ERTC_ALASBS/ERTC_ALA（ERTC_ALBSBS/ERTC_ALB）设定，开启闹钟后，当设定的闹钟值匹配日历值时将触发闹钟事件。通过 MASKx 位，可选择性的屏蔽日历字段，被屏蔽的字段不参与闹钟匹配。
闹钟 A、B 的配置可按以下步骤进行：
1. 关闭闹钟 A 或闹钟 B（设置 ALAEN=0 或 ALBEN=0）。
2. 等待闹钟 A 或闹钟 B 寄存器允许写（等待 ALAWF 或 ALBWF 位置 1）。
3. 配置闹钟 A 或闹钟 B 寄存器（ERTC_ALA/ERTC_ALASBS、ERTC_ALB/ERTC_ALBSBS）。
4. 使能闹钟 A 或闹钟 B（设置 ALAEN=1 或 ALBEN=1）。
注：当 ERTC_ALA 或 ERTC_ALB 中 MASK1 为 0 时，DIVB 至少为 3 才能使闹钟正常工作。

17.3.3 周期性自动唤醒
周期性唤醒功能用于 ERTC 周期性的自动唤醒低功耗模式，唤醒周期由 VAL[15：0]设定（WATCLK[2]=1 时扩展为 17 位，唤醒计数值为 VAL+2^{16}）。当唤醒计数器值由 VAL 值递减至 0 时，WATF 标志置 1，产生唤醒事件，同时唤醒计数器值重载 VAL 值。若使能周期性唤醒中断，将产生周期性唤醒中断。
驱动唤醒定时器的时钟通过 WATCLK[2:0]设定，可选 16/8/4/2 分频后的 ERTC_CLK 或 ck_b（通常 1Hz），结合唤醒计数值可灵活调整唤醒周期。
周期性唤醒功能可按以下步骤进行配置：
1. 关闭周期新唤醒（设置 WATEN=0）。
2. 等待唤醒自动重载定时器和 WATCLK[2：0]位允许写（WATWF 位置 1）。
3. 配置唤醒定时器计数值和唤醒时钟（VAL[15：0]、WATCLK[2：0]位）。
4. 使能定时器（设置 WATEN=1）。
注：系统复位以及低功耗模式（睡眠、深度睡眠和待机）对唤醒定时器没有任何影响。
注：DEBUG 模式下，若唤醒时钟选择 ERTC_CLK，用于周期性唤醒的计数器正常运行。

17.3.4 ERTC校准

ERTC 支持使用精密数字校准校准时钟。

精密数字校准：

精密校准的均匀性很好。开启精密校准校准功能后，将均匀增加或减少 ERTC_CLK 来达到校准的目的。当 ERTC_CLK 为 32.768kHz 时，精密校准周期约为 2^{20} 个 ERTC_CLK(32 秒)。DEC[8：0]值指定了 2^{20} 个 ERTC_CLK 中忽略的脉冲数，最多可忽略 511 个脉冲；将 ADD 置 1，可在 2^{20} 个 ERTC_CLK 中插入 512 个脉冲。两者搭配使用，可在 2^{20} 个 ERTC_CLK 周期进行-511~+512 的调整。

有效校准频率 F_{SCAL}：

$$F_{SCAL} = F_{ERTC\_CLK} \times [\,1 + \frac{ADD \times 512 - DEC}{2^{20} + DEC - ADD \times 512}\,]$$

当分频器A值小于3时，会按照ADD等于0校准。此时应降低分频器B值来实现每秒增加8个ERTC_CLK，也就是 32 秒增加 256 个 ERTC_CLK 搭配 DEC[8：0]位，可在 2^{20} 个 ERTC_CLK 周期进行-255~+256 的调整。

此时有效校准频率 F_{SCAL}：

$$F_{SCAL} = F_{ERTC\_CLK} \times [\,1 + \frac{256 - DEC}{2^{20} + DEC - 256}\,]$$

精密数字校准的校准周期还可选择 8 秒或 16 秒(由 CAL8 和 CAL16 配置)，8 秒校准周期的优先级更高，同时使能 8 秒和 16 秒校准周期，将优先选择 8 秒校准周期。

ERTC 提供了 CALUPDF 标志用来指示校准值的状态，当配置 ERTC_SCAL 寄存器时，CALUPDF 标志位将置 1，指示校准值正在更新；当校准值被成功应用后，标志位自动清 0，指示校准值更新完成。

17.3.5 参考时钟检测

为保证日历长时间运行的精确性，ERTC 提供了时钟同步功能（低功耗模式不可用），用精度更高的参考时钟（一般用 50Hz 或者 60Hz 的市电）校准更新日历的 1Hz 时钟。

参考时钟检测功能开启后，在每次更新日历值的前 7 个 ck_a 周期检测参考时钟边沿，若检测到边沿，将时用此边沿更新日历值，后续采用 3 个 ck_a 周期检测参考时钟边沿。每一次检测到参考时钟边沿时，都会将分频器 A 的值进行重载，这会使得内部 1Hz 的日历时钟与参考时钟边沿刚好对齐，当内部 1Hz 时钟出现微小偏移时，利用更精确的参考时钟，将 1Hz 时钟微调至与参考时钟边沿对齐。当没有检测到参考时钟边沿时，ERTC 会利用原来的时钟源更新日历。

需要注意的是，使能参考时钟功能后，需要将 DIVA、DIVB 设置为复位值（0x7F、0xFF）。

17.3.6 时间戳

时间戳功能用于在发生时间戳事件时（入侵引脚检测到有效边沿），将当前的日历值保存到时间戳寄存器中。

当发生时间戳时，TSF 位置 1，此时若再次发生时间戳事件，TSOF 标志位将置 1，但时间戳寄存器并不会更新，可以通过 TSIEN 位设置是否使能时间戳中断。

时间戳用法有两种

1. 单独的时间戳功能，此时入侵检测引脚用来检测时间戳，使用步骤：
- 选择时间戳引脚（设置 TSPIN）
- 选择上升沿还是下降沿触发（设置 TSEDG）
- 使能时间戳（设置 TSEN=1）

2. 发生入侵事件时保存时间戳，使用步骤：
- 配置入侵检测相关寄存器
- 使能发生入侵事件时保存时间戳（设置 TPTSEN=1）

注：发生时间戳事件后，TSF 在两个 ck_a 周期后置 1，建议在 TSF 已置 1 的情况下轮询 TSOF 位

17.3.7 入侵检测

ERTC 提供了两组入侵检测 TAMP1、TAMP2，且两组入侵检测可分别配置为滤波后的电平检测或边沿检测。TAMP1 通过 TSPIN 选择入侵引脚 ERTC_MUX1 或 ERTC_MUX2；TAMP2 只能选择 ERTC_MUX2。

当检测到有效的入侵事件后，TP1F 或 TP2F 将置 1，若已使能了入侵检测中断，将产生对应的中断；若

TPTSEN 位已置 1，将同时产生时间戳事件。为保证位于电池供电域中的电池供电寄存器数据安全，入侵事件发生时将复位电池供电寄存器。

边沿入侵检测配置步骤：

1. 选择入侵检测方式为边沿检测（TPFLT=00），并选择有效沿（TP1EDG 或 TP2EDG）。
2. 根据需要配置是否在入侵事件时激活时间戳（TPTSEN 置 1）。
3. 根据需要开启入侵中断使能（TPIEN 置 1）。
4. 若选择 TAMP1 进行入侵检测，选择 TAMP1 映射关系为 ERTC_MUX1 或是 ERTC_MUX2（TP1PIN）并将 TAMP1 使能（TP1EN 置 1）；若选择 TAMP2 进行入侵检测，则只需使能 TAMP2（TP2EN 置 1）。

电平入侵检测配置步骤：

1. 选择入侵检测方式为电平检测，并选择有效电平采样次数（TPFLT≠00）。
2. 选择入侵有效电平（TP1EDG 或 TP2EDG）。
3. 选择入侵检测采样频率（TPFREQ）。
4. 根据需要开启入侵检测上拉（TPPU 置 1），若开启，还需配置上拉电阻预充电时间（TPPR）。
5. 根据需要配置是否在入侵事件时激活时间戳（TPTSEN 置 1）。
6. 根据需要开启入侵中断使能（TPIEN 置 1）。
7. 若选择 TAMP1 进行入侵检测，选择 TAMP1 映射关系为 ERTC_MUX1 或是 ERTC_MUX2（TP1PIN）并将 TAMP1 使能（TP1EN 置 1）；若选择 TAMP2 进行入侵检测，则只需使能 TAMP2（TP2EN 置 1）

当配置为边沿检测时，有以下两点要注意：

1. 在使能入侵检测前，若入侵检测已被配置为上升沿有效，且入侵检测引脚已为高电平，在使能入侵检测后会立刻产生一个入侵检测事件。
2. 在使能入侵检测前，若入侵检测已被配置为下降沿有效，且入侵检测引脚已为低电平，在使能入侵检测后会立刻产生一个入侵检测事件。

注：当电池供电域电源关闭时，入侵检测失效。

注：standby 模式下，TAMP2（PA0 引脚）无法使用预充电功能。

17.3.8 复用功能输出

ERTC 提供了一组复用功能输出，可以输出以下事件：

1. 校准后的时钟（OUTSEL=0，CALOEN=1）
   - 输出 512Hz（CALOSEL=0）
   - 输出 1Hz（CALOSEL=1）
2. 闹钟A（OUTSEL=1）
3. 闹钟B（OUTSEL=2）
4. 唤醒事件（OUTSEL=3）

当输出闹钟事件或者唤醒事件时（OUTSEL≠0），可以通过 OUTTYPE 选择输出类型为开漏或是推挽，可以通过 OUTP 配置输出极性。

17.3.9 ERTC唤醒

ERTC 可由闹钟、周期性唤醒、时间戳、入侵事件进行唤醒，使能 ERTC 中断可按以下操作配置：

1. 将 ERTC 对应中断的 EXINT 线配置为中断模式并使能，有效沿选择上升沿。
2. 使能 ERTC 中断对应的 NVIC 通道。
3. 使能对应的 ERTC 中断。

下表说明了 ERTC 时钟源、事件以及中断对唤醒低功耗模式的影响：

表 17-2 ERTC唤醒低功耗模式

时钟源	事件	唤醒 SLEEP	唤醒 DEEPSLEEP	唤醒 STANDBY
HEXT	闹钟 A	√	×	×
	闹钟 B	√	×	×
	周期性唤醒	√	×	×
	时间戳	√	×	×
	入侵事件	√	×	×
LICK	闹钟 A	√	√	√
	闹钟 B	√	√	√
	周期性唤醒	√	√	√

	时间戳	√	√	√
	入侵事件	√	√	√
LEXT	闹钟 A	√	√	√
	闹钟 B	√	√	√
	周期性唤醒	√	√	√
	时间戳	√	√	√
	入侵事件	√	√	√

表 17-3中断控制位

中断事件	事件标志	中断使能位	EXINT 线
闹钟 A	ALAF	ALAIEN	17
闹钟 B	ALBF	ALBIEN	17
周期性唤醒	WATF	WATIEN	22
时间戳	TSF	TSIEN	21
入侵事件	TP1F/TP2F	TPIEN	21

17.4ERTC寄存器描述

必须以字（32 位）的方式操作这些外设寄存器。
ERTC 寄存器是 32 位可寻址寄存器，具体描述如下：
表 17-4 ERTC-寄存器映像和复位值

寄存器简称	基址偏移量	复位值
ERTC_TIME	0x00	0x0000 0000
ERTC_DATE	0x04	0x0000 2101
ERTC_CTRL	0x08	0x0000 0000
ERTC_STS	0x0C	0x0000 0007
ERTC_DIV	0x10	0x007F 00FF
ERTC_WAT	0x14	0x0000 FFFF
ERTC_ALA	0x1C	0x0000 0000
ERTC_ALB	0x20	0x0000 0000
ERTC_WP	0x24	0x0000 0000
ERTC_SBS	0x28	0x0000 0000
ERTC_TADJ	0x2C	0x0000 0000
ERTC_TSTM	0x30	0x0000 0000
ERTC_TSDT	0x34	0x0000 000D
ERTC_TSSBS	0x38	0x0000 0000
ERTC_SCAL	0x3C	0x0000 0000
ERTC_TAMP	0x40	0x0000 0000
ERTC_ALASBS	0x44	0x0000 0000
ERTC_ALBSBS	0x48	0x0000 0000
ERTC_BPRx	0x50-0x9C	0x0000 0000

17.4.1 ERTC时间寄存器(ERTC_TIME)

域	简称	复位值	类型	功能
位 31:23	保留	0x000	resd	保持默认值。
位 22	AMPM	0x0	rw	上午/下午（AM/PM） 0：上午； 1：下午。 注：该位只用于 12 小时制，24 小时制保持为 0。
位 21:20	HT	0x0	rw	小时十位（Hour tens）

位 19:16	HU	0x0	rw	小时个位（Hour units）
位 15	保留	0x0	resd	保持默认值。
位 14:12	MT	0x0	rw	分钟十位（Minute tens）
位 11:8	MU	0x0	rw	分钟个位（Minute units）
位 7	保留	0x0	resd	保持默认值
位 6:4	ST	0x0	rw	秒钟十位（Second tens）
位 3:0	SU	0x0	rw	秒钟个位（Second units）

17.4.2 ERTC日期寄存器(ERTC_DATE)

域	简称	复位值	类型	功能
位 31:24	保留	0x00	resd	保持默认值。
位 23:20	YT	0x0	rw	年份十位（Year tens）
位 19:16	YU	0x0	rw	年份个位（Year units）
位 15:13	WK	0x1	rw	星期（Week） 0：禁用； 1：星期一； 2：星期二； 3：星期三； 4：星期四； 5：星期五； 6：星期六； 7：星期日。
位 12	MT	0x0	rw	月份十位（Month tens）
位 11:8	MU	0x1	rw	月份个位（Month units）
位 7:6	保留	0x0	resd	保持默认值。
位 5:4	DT	0x0	rw	日期十位（Date tens）
位 3:0	DU	0x1	rw	日期个位（Date units）

17.4.3 ERTC控制寄存器(ERTC_CTRL)

域	简称	复位值	类型	功能
位 31:24	保留	0x00	resd	保持默认值。
位 23	CALOEN	0x0	rw	校准输出使能（Calibration output enable） 0：关闭； 1：开启。
位 22:21	OUTSEL	0x0	rw	输出源选择（Output source selection） 00：关闭； 01：闹钟 A； 10：闹钟 B； 11：唤醒事件。
位 20	OUTP	0x0	rw	输出极性（Output polarity） 0：高； 1：低。
位 19	CALOSEL	0x0	rw	校准输出选择（Calibration output selection） 0：512Hz； 1：1Hz。
位 18	BPR	0x0	rw	电池供电域数据寄存器（Battery power domain data register） 该位在电池供电域，不受系统复位影响，可用来存储夏令时操作或者一些其他需要一直保存的数据。
位 17	DEC1H	0x0	wo	减少 1 小时（Decrease 1 hour） 0：无作用； 1：减少一小时。 注：当小时不为 0 时才有效，该位置 1 后（不要在小时递增时置 1），下一秒生效。
位 16	ADD1H	0x0	wo	增加 1 小时（Add 1 hour） 0：无作用； 1：增加一小时。 注：该位置 1 后（不要在小时递增时置 1），下一秒生效。

位 15	TSIEN	0x0	rw	时间戳中断使能（Timestamp interrupt enable） 0：关闭； 1：开启。
位 14	WATIEN	0x0	rw	唤醒定时器中断使能（Wakeup timer interrupt enable） 0：关闭； 1：开启。
位 13	ALBIEN	0x0	rw	闹钟 B 中断使能（Alarm B interrupt enable） 0：关闭； 1：开启。
位 12	ALAIEN	0x0	rw	闹钟 A 中断使能（Alarm A interrupt enable） 0：关闭； 1：开启。
位 11	TSEN	0x0	rw	时间戳使能（Timestamp enable） 0：关闭； 1：开启。
位 10	WATEN	0x0	rw	唤醒定时器使能（Wakeup timer enable） 0：关闭； 1：开启。
位 9	ALBEN	0x0	rw	闹钟 B 使能（Alarm B enable） 0：关闭； 1：开启。
位 8	ALAEN	0x0	rw	闹钟 A 使能（Alarm A enable） 0：关闭； 1：开启。
位 7	保留	0x0	resd	保持默认值
位 6	HM	0x0	rw	小时模式（Hour mode） 0：24 小时制； 1：12 小时制。
位 5	DREN	0x0	rw	日期/时间寄存器直接读取使能（Date/time register direct read enable） 0：关闭，ERTC_TIME、ERTC_DATE、ERTC_SBS 值从同步寄存器获取，每两个 ERTC_CLK 更新一次； 1：开启，ERTC_TIME、ERTC_DATE、ERTC_SBS 值从电池供电域获取。
位 4	RCDEN	0x0	rw	参考时钟检测使能（Reference clock detection enable） 0：关闭； 1：开启。
位 3	TSEDG	0x0	rw	时间戳触发边沿（Timestamp trigger edge） 0：上升沿； 1：下降沿。
位 2:0	WATCLK	0x0	rw	唤醒定时器时钟选择（Wakeup timer clock selection） 000：ERTC_CLK/16； 001：ERTC_CLK/8； 010：ERTC_CLK/4； 011：ERTC_CLK/2； 10x：ck_b； 11x：ck_b，唤醒计数值增加 2^{16}，唤醒时间 =ERTC_WAT+2^{16}。 *注：在 WATEN=0 且 WATWF=1 时可对这些位进行写操作。*

17.4.4 ERTC初始化和状态寄存器(ERTC_STS)

域	简称	复位值	类型	功能
位 31:17	保留	0x0000	resd	保持默认值。
位 16	CALUPDF	0x0	ro	校准值更新完成标志（Calibration value update completed flag） 0：更新完成； 1：正在更新。

				当对精密校准寄存器 ERTC_SCAL 写时，该位自动置 1，当 ERTC 使用新的校准值时，该位自动清零，当该位为 1 时，不能写 ERTC_SCAL 寄存器。
位 15	保留	0x0	resd	保持默认值。
位 14	TP2F	0x0	rw0c	入侵检测 2 标志（Tamper detection 2 flag） 0：无入侵事件发生； 1：有入侵事件发生。
位 13	TP1F	0x0	rw0c	入侵检测 1 标志（Tamper detection 1 flag） 0：无入侵事件发生； 1：有入侵事件发生。
位 12	TSOF	0x0	rw0c	时间戳溢出标志（Timestamp overflow flag） 0：正常； 1：溢出。 当产生了时间戳事件（TSF 置 1）时，又产生了时间戳事件，该标志置 1。
位 11	TSF	0x0	rw0c	时间戳标志（Timestamp flag） 0：无时间戳事件发生； 1：有时间戳事件发生。 当读取了时间戳，并清除了 TSF 时，建议再检查 TSOF 标志，因为可能会存在当正在清除 TSF 时又产生了时间戳事件。 注：该位清 0 后 2 个 APB_CLK 后生效。
位 10	WATF	0x0	rw0c	唤醒定时器标志（Wakeup timer flag） 0：无唤醒事件发生； 1：有唤醒事件发生。 注：该位清 0 后 2 个 APB_CLK 后生效。
位 9	ALBF	0x0	rw0c	闹钟 B 标志（Alarm clock B flag） 0：无闹钟事件发生； 1：有闹钟事件发生。 注：该位清 0 后 2 个 APB_CLK 后生效。
位 8	ALAF	0x0	rw0c	闹钟 A 标志（Alarm clock A flag） 0：无闹钟事件发生； 1：有闹钟事件发生。 注：该位清 0 后 2 个 APB_CLK 后生效。
位 7	IMEN	0x0	rw	初始化模式使能（Initialization mode enable） 0：关闭； 1：开启。 当进入了初始化模式后，日历处于停止状态。
位 6	IMF	0x0	ro	进入初始化模式标志（Enter initialization mode flag） 0：未进入； 1：进入。 当使能了初始化模式（INITEN=1），并进入了初始化模式（INITEF=1）时，才能更改 ERTC_TIME、ERTC_DATE、ERTC_DIV 寄存器。
位 5	UPDF	0x0	rw0c	日历更新标志（Calendar update flag） 0：正在更新； 1：更新完成。 每当从电池供电域将日历更新到 ERTC_TIME、ERTC_DATE、ERTC_SBS 同步寄存器，该标志置 1，每两个 ERTC_CLK 更新一次。
位 4	INITF	0x0	ro	日历初始化标志（Calendar initialization flag） 0：未初始化； 1：已初始化。 当检测到 ERTC_DATE 里面的年份不为 0 时该位置 1，当年份为 0 时，该位清 0。
位 3	TADJF	0x0	ro	时间调整标志（Time adjustment flag） 0：无操作； 1：正在执行时间调整。 当对时间调整存器 ERTC_TADJ 写时，该位自动置 1，当时间调整结束后，该位自动清零。

位 2	WATWF	0x1	ro	唤醒定时器寄存器允许写标志（Wakeup timer register allows write flag） 0：不允许写； 1：允许写。
位 1	ALBWF	0x1	ro	闹钟 B 允许写标志（Alarm B register allows write flag） 0：不允许写； 1：允许写。
位 0	ALAWF	0x1	ro	闹钟 A 允许写标志（Alarm B register allows write flag） 0：不允许写 1：允许写

17.4.5 ERTC预分频器寄存器(ERTC_DIV)

域	简称	复位值	类型	功能
位 31:23	保留	0x000	resd	保持默认值。
位 22:16	DIVA	0x7F	rw	分频器 A（Diveder A）
位 15	保留	0x0	resd	保持默认值。
位 14:0	DIVB	0x00FF	rw	分频器 B（Diveder B） 日历时钟=ERTC_CLK/((DIVA+1)x(DIVB+1))。

17.4.6 ERTC唤醒定时器寄存器(ERTC_WAT)

域	简称	复位值	类型	功能
位 31:16	保留	0x0000	resd	保持默认值。
位 15:0	VAL	0xFFFF	rw	唤醒定时器重载值（Wakeup timer reload value）

17.4.7 ERTC闹钟A寄存器(ERTC_ALA)

域	简称	复位值	类型	功能
位 31	MASK4	0x0	rw	日期/星期屏蔽（Date/week mask） 0：无屏蔽； 1：闹钟和日期/星期无关。
位 30	WKSEL	0x0	rw	日期/星期选择（Date/week mode select） 0：日期； 1：星期（DT[1：0]不使用）。
位 29:28	DT	0x0	rw	日期十位（Date tens）
位 27:24	DU	0x0	rw	日期/星期个位（Date/week units）
位 23	MASK3	0x0	rw	小时屏蔽（Hour mask） 0：无屏蔽； 1：闹钟和小时无关。
位 22	AMPM	0x0	rw	上午/下午（AM/PM） 0：上午； 1：下午。 *注：该位只用于 12 小时制，24 小时制保持为 0。*
位 21:20	HT	0x0	rw	小时十位（Hour tens）
位 19:16	HU	0x0	rw	小时个位（Hour units）
位 15	MASK2	0x0	rw	分钟屏蔽（Minute mask） 0：无屏蔽； 1：闹钟和分钟无关。
位 14:12	MT	0x0	rw	分钟十位（Minute tens）
位 11:8	MU	0x0	rw	分钟个位（Minute units）
位 7	MASK1	0x0	rw	秒钟屏蔽（Second mask） 0：无屏蔽； 1：闹钟和秒钟无关。
位 6:4	ST	0x0	rw	秒钟十位（Second tens）
位 3:0	SU	0x0	rw	秒钟个位（Second units）

17.4.8 ERTC闹钟B寄存器(ERTC_ALB)

域	简称	复位值	类型	功能
位 31	MASK4	0x0	rw	日期/星期屏蔽（Date/week mask） 0：无屏蔽；

				1：闹钟和日期/星期无关。
位 30	WKSEL	0x0	rw	日期/星期选择（Date/week mode select） 0：日期； 1：星期（DT[1：0]不使用）。
位 29:28	DT	0x0	rw	日期十位（Date tens）
位 27:24	DU	0x0	rw	日期/星期个位（Date/week units）
位 23	MASK3	0x0	rw	小时屏蔽（Hour mask） 0：无屏蔽； 1：闹钟和小时无关。
位 22	AMPM	0x0	rw	上午/下午（AM/PM） 0：上午； 1：下午。 注：该位只用于 12 小时制，24 小时制保持为 0。
位 21:20	HT	0x0	rw	小时十位（Hour tens）
位 19:16	HU	0x0	rw	小时个位（Hour units）
位 15	MASK2	0x0	rw	分钟屏蔽（Minute mask） 0：无屏蔽； 1：闹钟和分钟无关。
位 14:12	MT	0x0	rw	分钟十位（Minute tens）
位 11:8	MU	0x0	rw	分钟个位（Minute units）
位 7	MASK1	0x0	rw	秒钟屏蔽（Second mask） 0：无屏蔽； 1：闹钟和秒钟无关。
位 6:4	ST	0x0	rw	秒钟十位（Second tens）
位 3:0	SU	0x0	rw	秒钟个位（Second units）

17.4.9 ERTC写保护寄存器(ERTC_WP)

域	简称	复位值	类型	功能
位 31:8	保留	0x000000	resd	保持默认值。
位 7:0	CMD	0x00	wo	命令寄存器（Command register） 依次写入 0xCA、0x53 解锁所有 ERTC 寄存器写保护，当写一个其他值时，将重新开启写保护。

17.4.10 ERTC亚秒寄存器(ERTC_SBS)

域	简称	复位值	类型	功能
位 31:16	保留	0x0000	resd	保持默认值。
位 15:0	SBS	0x0000	ro	亚秒值（Sub-second value） 亚秒为分频器 DIVB 的计数值，时钟频率为 ERTC_CLK/(DIVA+1)。

17.4.11 ERTC时间微调寄存器(ERTC_TADJ)

域	简称	复位值	类型	功能
位 31	ADD1S	0x0	wo	增加一秒（Add 1 second） 0：无效果； 1：增加 1 秒。 当 TADJF=0 时，才能写此寄存器，通常 ADD1S 与 DECSBS 配合使用，达到微调时间的效果。
位 30:15	保留	0x0000	resd	保持默认值。
位 14:0	DECSBS	0x0000	wo	DECSBS[14：0]：减少亚秒值（Decrease sub-second value） 延迟时间（ADD1S=0）：延迟=DECSBS/(DIVB+1)； 提前时间（ADD1S=1）：延迟=1-(DECSBS/(DIVB+1))。

17.4.12 ERTC时间戳时间寄存器(ERTC_TSTM)

域	简称	复位值	类型	功能
位 31:23	保留	0x000	resd	保持默认值。
位 22	AMPM	0x0	ro	上午/下午（AM/PM） 0：上午；

				1：下午。 注：该位只用于 12 小时制，24 小时制保持为 0。
位 21:20	HT	0x0	ro	小时十位（Hour tens）
位 19:16	HU	0x0	ro	小时个位（Hour units）
位 15	保留	0x0	resd	保持默认值
位 14:12	MT	0x0	ro	分钟十位（Minute tens）
位 11:8	MU	0x0	ro	分钟个位（Minute units）
位 7	保留	0x0	resd	保持默认值。
位 6:4	ST	0x0	ro	秒钟十位（Second tens）
位 3:0	SU	0x0	ro	秒钟个位（Second units）

注意：仅当 ERTC_STS 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

17.4.13 ERTC时间戳日期寄存器(ERTC_TSDT)

域	简称	复位值	类型	功能
位 31:16	保留	0x0000	resd	保持默认值。
位 15:13	WK	0x0	ro	星期（Week）
位 12	MT	0x0	ro	月十位（Month tens）
位 11:8	MU	0x0	ro	月个位（Month units）
位 7:6	保留	0x0	resd	保持默认值
位 5:4	DT	0x0	ro	日期十位（Date tens）
位 3:0	DU	0x0	ro	日期个位（Date units）

注意：仅当 ERTC_STS 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

17.4.14 ERTC时间戳亚秒寄存器(ERTC_TSSBS)

域	简称	复位值	类型	功能
位 31:16	保留	0x0000	resd	保持默认值。
位 15:0	SBS	0x0000	ro	亚秒值（Sub-second value）

注意：仅当 ERTC_STS/TSF 置 1 时，该寄存器的内容才有效。当 ERTC_STS/TSF 位复位时，清零该 寄存器。

17.4.15 ERTC精密校准寄存器(ERTC_SCAL)

域	简称	复位值	类型	功能
位 31:16	保留	0x0000	resd	保持默认值。
位 15	ADD	0x0	rw	增加 ERTC 时钟（Add ERTC clock） 0：无操作； 1：每 2^{11} 个 ERTC_CLK，插入一个 ERTC_CLK。
位 14	CAL8	0x0	rw	8 秒校准周期（8-second calibration period） 0：无效果； 1：8 秒校准周期。
位 13	CAL16	0x0	rw	16 秒校准周期（16 second calibration period） 0：无效果； 1：16 秒校准周期。
位 12:9	保留	0x0	resd	保持默认值
位 8:0	DEC	0x000	rw	减少 ERTC 时钟（Decrease ERTC clock） 在 2^{20} 个 ERTC_CLK 周期内，屏蔽 DEC 个 ERTC_CLK。 通常和 ADD 配合使用，当 ADD 为 1 时，在 2^{20} 个 ERTC_CLK 周期内，实际的 ERTC_CLK 个数为 2^{20}+512-DEC。

17.4.16 ERTC入侵配置寄存器(ERTC_TAMP)

域	简称	复位值	类型	功能
位 31:19	保留	0x0000	resd	保持默认值。
位 18	OUTTYPE	0x0	rw	输出类型（Output type） 0：开漏；

				1：推挽。
位 17	TSPIN	0x0	rw	时间戳检测引脚选择（Time stamp detection pin selection） 0：ERTC_MUX1； 1：ERTC_MUX2。
位 16	TP1PIN	0x0	rw	入侵检测 1 引脚选择（Tamper detection pin selection） 0：ERTC_MUX1； 1：ERTC_MUX2。
位 15	TPPU	0x0	rw	入侵检测上拉（Tamper detection pull-up） 0：开启； 1：关闭。
位 14:13	TPPR	0x0	rw	入侵检测预充电时间（Tamper detection pre-charge time） 0：1 个 ERTC_CLK； 1：2 个 ERTC_CLK； 2：4 个 ERTC_CLK； 3：8 个 ERTC_CLK。
位 12:11	TPFLT	0x0	rw	入侵检测滤波时间（Tamper detection filter time） 0：无滤波； 1：连续 2 次采样有效，判定入侵事件发生； 2：连续 4 次采样有效，判定入侵事件发生； 3：连续 8 次采样有效，判定入侵事件发生。
位 10:8	TPFREQ	0x0	rw	入侵检测检测频率（Tamper detection frequency） 0：ERTC_CLK/32768； 1：ERTC_CLK/16384； 2：ERTC_CLK/8192； 3：ERTC_CLK/4096； 4：ERTC_CLK/2048； 5：ERTC_CLK/1024； 6：ERTC_CLK/512； 7：ERTC_CLK/256。
位 7	TPTSEN	0x0	rw	入侵检测时间戳使能（Tamper detection timestamp enable） 0：关闭； 1：开启，当产生入侵事件时，保持时间戳。
位 6：5	保留	0x0	resd	保持默认值。
位 4	TP2EDG	0x0	rw	入侵检测 2 有效边沿（Tamper detection 2 valid edge） 当无滤波时（TPFLT=0）： 0：上升沿； 1：下降沿。 当有滤波时（TPFLT>0）： 0：低电平； 1：高电平。
位 3	TP2EN	0x0	rw	入侵检测 2 使能（Tamper detection 2 enable） 0：关闭； 1：开启。
位 2	TPIEN	0x0	rw	入侵检测中断使能（Tamper detection interrupt enable） 0：关闭； 1：开启。
位 1	TP1EDG	0x0	rw	入侵检测 1 有效边沿（Tamper detection 1 valid edge） 当无滤波时（TPFLT=0）： 0：上升沿； 1：下降沿。 当有滤波时（TPFLT>0）： 0：低电平； 1：高电平。
位 0	TP1EN	0x0	rw	入侵检测 1 使能（Tamper detection 1 enable） 0：关闭； 1：开启。

17.4.17 ERTC闹钟A亚秒寄存器(ERTC_ALASBS)

域	简称	复位值	类型	功能
位 31:28	保留	0x0	resd	保持默认值。
位 27:24	SBSMSK	0x0	rw	亚秒屏蔽（Sub-second mask） 0：不匹配亚秒，闹钟与亚秒无关； 1：只匹配 SBS[0]； 2：只匹配 SBS[1：0]； 3：只匹配 SBS[2：0]； ... 14：只匹配 SBS[13：0]； 15：匹配 SBS[14：0]。
位 23:15	保留	0x000	rw	保持默认值。
位 14:0	SBS	0x0000	rw	亚秒值（Sub-second value）

17.4.18 ERTC闹钟A亚秒寄存器(ERTC_ALASBS)

域	简称	复位值	类型	功能
位 31:28	保留	0x0	resd	保持默认值。
位 27:24	SBSMSK	0x0	rw	亚秒屏蔽（Sub-second mask） 0：不匹配亚秒，闹钟与亚秒无关； 1：只匹配 SBS[0]； 2：只匹配 SBS[1：0]； 3：只匹配 SBS[2：0]； ... 14：只匹配 SBS[13：0]； 15：匹配 SBS[14：0]。
位 23:15	保留	0x000	rw	保持默认值。
位 14:0	SBS	0x0000	rw	亚秒值（Sub-second value）

17.4.19 ERTC 电池供电数据寄存器(ERTC_BPRx)

域	简称	复位值	类型	功能
位 31:0	DT	0x0000 0000	rw	电池供电域数据（Battery powered domain data） BPR_DTx 寄存器，可以在只由电池供电域供电下保存数据，不会被系统复位所复位，只能通过电池供电域复位或入侵事件进行复位。

18 模拟/数字转换（ADC）

18.1 ADC简介

ADC 是一个将模拟输入信号转换为 12 位、10 位、8 位或 6 位的数字信号的外设。采样率最高可达 5.33MSPS。多达 26 个通道源（包括内部及外部通道）可进行采样及转换，总计 24 个外部输入通道源。

18.2 ADC主要特征

模拟方面有以下特征：

- 支持分辨率 12 位、10 位、8 位或 6 位的转换
- 自校准时间： 205 个 ADC 时钟周期
- ADC 转换时间
  - 快速通道: ADC时钟80MHz，分辨率12位时转换时间为0.1875 μs(5.33MSps)
  - 慢速通道: ADC时钟80MHz，分辨率12位时转换时间为0.2375 μs(4.21MSps)
  - 快速通道: ADC时钟80MHz，分辨率6位时转换时间为0.1126 μs(8.88MSps)
  - 慢速通道: ADC时钟80MHz，分辨率6位时转换时间为0.1626 μs(6.15MSps)
- ADC 供电要求：2.4V 到 3.6V
- ADC 输入范围：$V_{REF-} \leqslant V_{IN} \leqslant V_{REF+}$

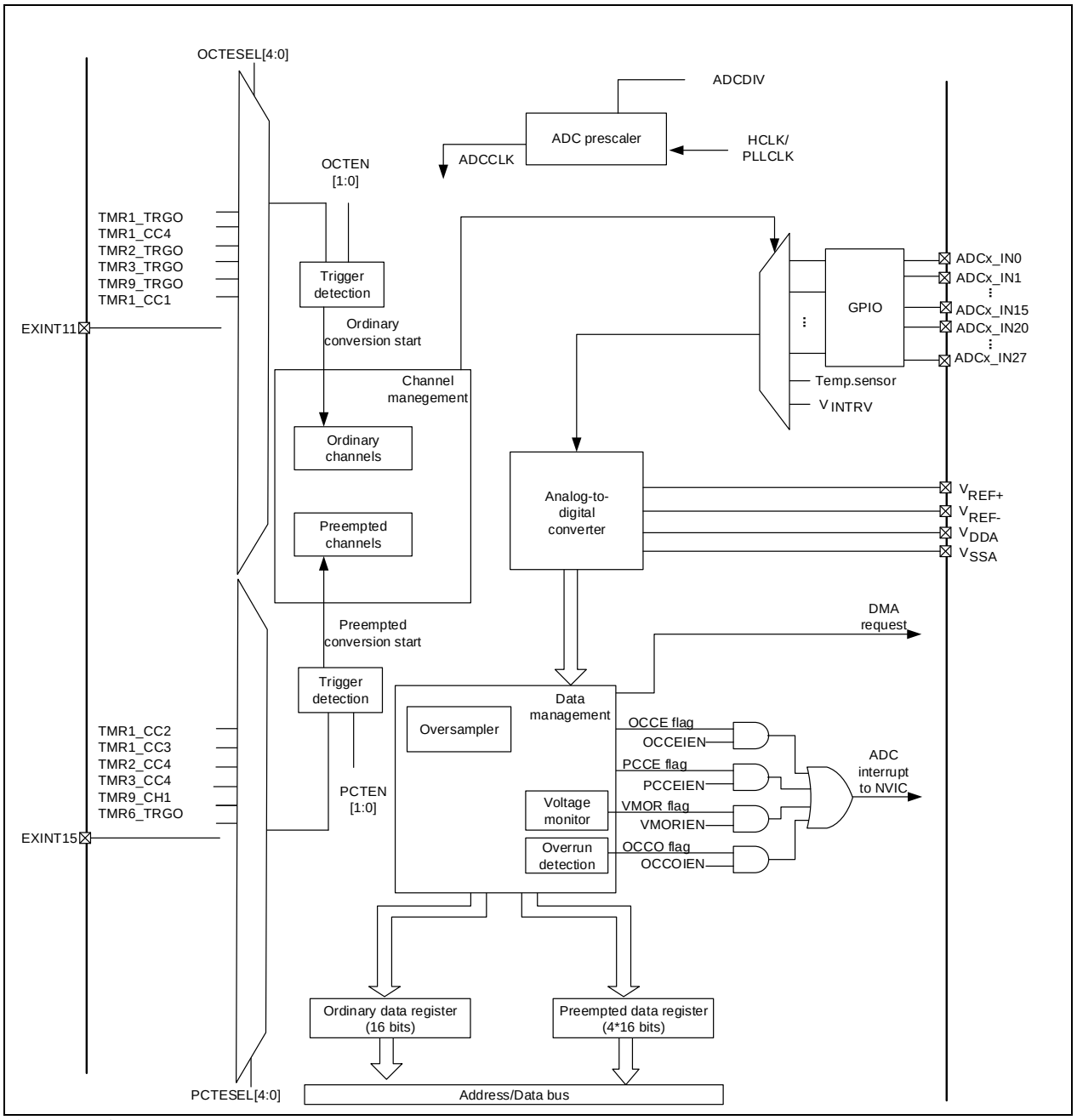
数字控制方面有以下特征：

- 通道管理区分优先权不同的普通通道与抢占通道
- 普通通道与抢占通道具备各自独立的触发侦测电路
- 各通道均可独立配置采样时间
- 转换顺序管理支持多种不同的多通道转换
- 过采样器：硬件过采样最高可实现 16 位分辨率
- 可选择的数据对齐方式
- 可配置的电压监测边界
- 支持 DMA 传输的普通通道数据
- 可设定以下事件发生时响应中断
  - 普通通道转换数据溢出
  - 抢占通道组转换结束
  - 普通通道转换结束
  - 电压监测超出范围

18.3 ADC架构

ADC1 的架构如图 18-1 ADC1 框图所示。

图 18-1 ADC1框图



输入引脚介绍：

- V_{DDA}：模拟电源，ADC 模拟电源，可与 V_{DD} 连接，或 $2.4V \le V_{DDA} \le V_{DD}$（3.6V）
- V_{SSA}：模拟电源地，ADC 模拟电源地，必须与 V_{SS} 连接
- V_{REF+}：模拟参考正极，ADC 使用的高端/正极模拟参考电压，$2.0V \le V_{REF+} \le V_{DDA}$
- V_{REF-}：模拟参考负极，ADC 使用的低端/负极参考电压，必须与 V_{SS} 连接
- ADCx_IN ：模拟输入信号通道

18.4 ADC功能介绍

18.4.1 通道管理

模拟信号通道输入

每个 ADC 拥有多达 26 个模拟信号通道输入，以 ADC_INx 表示，x=0 至 17、20 至 27。

- ADC_IN0 至 ADC_IN15 为外部模拟输入，ADC_IN16 为内部温度传感器，ADC_IN17 为内部参考电压，ADC_IN20 至 ADC_IN27 为外部模拟输入。

通道转换

转换区分为普通通道转换与抢占通道转换，抢占通道的转换优先权高于普通通道。
抢占通道触发若发生于普通通道转换途中，优先进行抢占通道的转换，普通通道于抢占通道转换结束后重新开始转换被打断的通道。普通通道触发若发生于抢占通道转换途中，普通通道的转换会等待抢占通道转换完成后才开始。
将通道（ADC_INx）编排进普通通道序列（ADC_OSQx）以及抢占通道序列（ADC_PSQ），相同通道可重复编排，序列总数由 OCLEN 与 PCLEN 定义，接着即可启动普通通道转换或抢占通道转换。

18.4.1.1 内部温度传感器

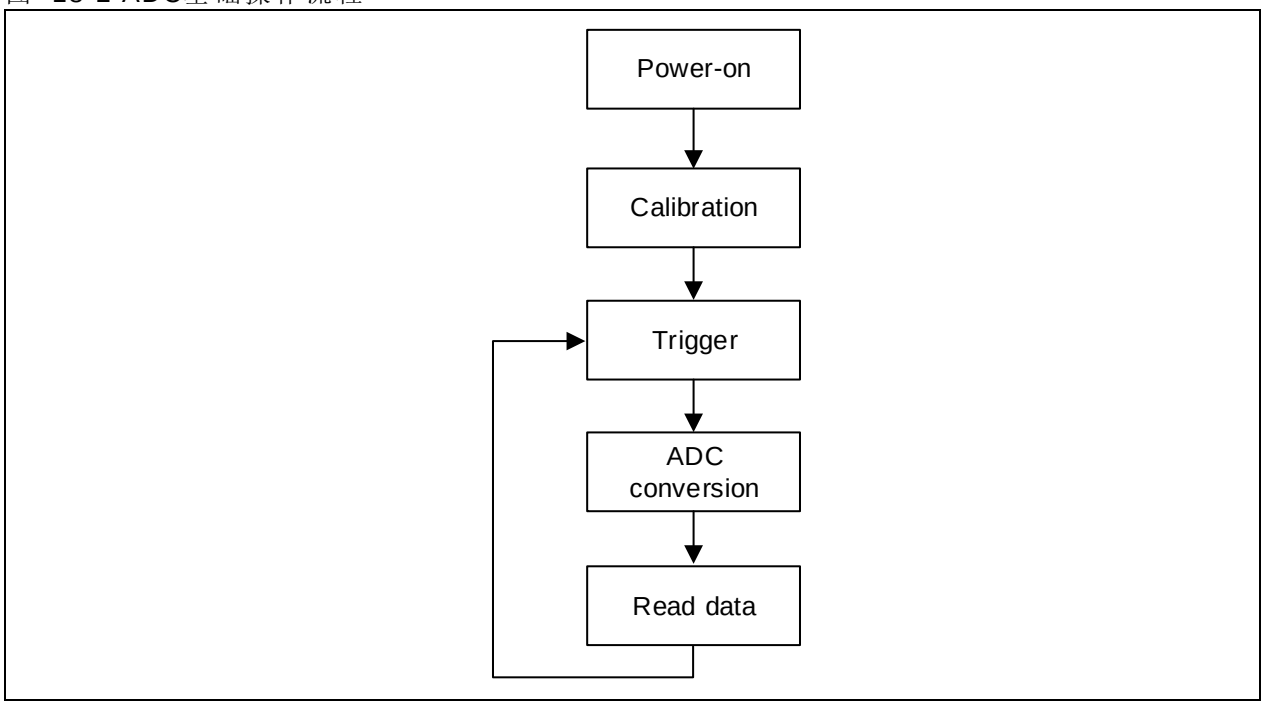
温度传感器接到 ADC1_IN16，必须先使能 ADC_CCTRL 的 ITSRVEN 位并且等待上电时间后才可对温度传感通道进行转换。
转换后获得的数据，搭配数据手册的电气特性章节提供的 25° C 的电压值与数据对温度斜率（Avg_Slope），即可推算温度。

18.4.1.2 内部参考电压

典型值 1.2V 的内部参考电压接到 ADC1_IN17，必须先使能 ADC_CCTRL 的 ITSRVEN 位后才可对内部参考电压通道进行转换。此通道的转换数据可用于推算外部参考电压。

18.4.2 ADC操作流程

ADC 的基础操作流程如下图所示，建议第一次上电后进行校准，以提升采样与转换准确度。待校准完成后可靠触发引起 ADC 采样转换，转换结束后即可读取数据。
图 18-2 ADC基础操作流程



18.4.2.1 上电与校准

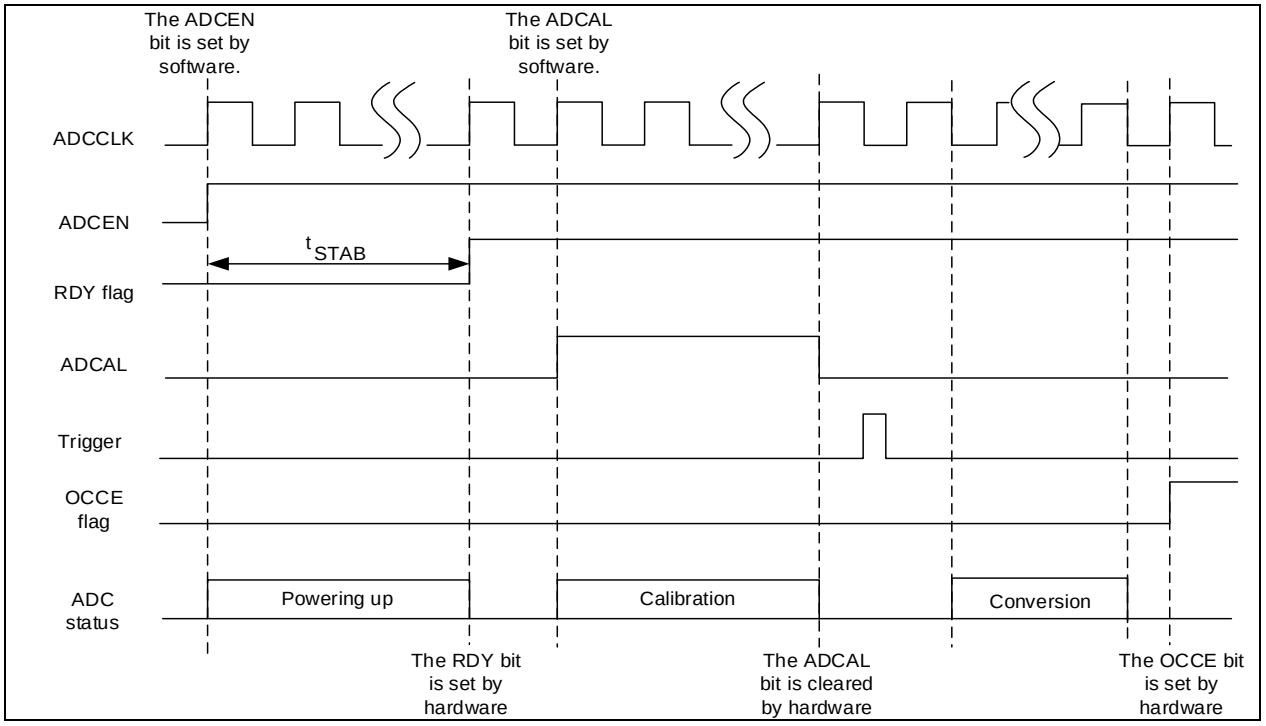
上电
用户须先使能 CRM_APB2EN 的 ADCxEN，以使能 ADC 的时钟：PCLK2 与 ADCCLK。
时钟使能后必须配置 ADC 预分频器（ADC_CCTRL 的 ADCDIV），将 ADCCLK 调整至需求的频率。
CRM_MISC1 的 PLLCLK_TO_ADC 位可选择 ADC 时钟来源，当选择 PLL 作为 ADC 时钟源时，此时必须保证系统时钟 SCLKSEL 选择 PLL/2。
注意： ADCCLK 不可大于 80MHz，ADCCLK 频率不可高于 PCLK2。
ADCCLK 频率调整完后，需配置 ADC_MISC 中 XTEST[5:0]为 0x9，之后使能 ADC_CTRL2 的 ADCEN 位使 ADC 上电，等待 RDY 标志位置起后才可对 ADC 进行后续操作。清除 ADCEN 会使 ADC 的转换中止并复位，同时 ADC 被断电以达到省电的效果。
校准
上电完成后可设置 ADC_CTRL2 的 ADCAL 使 ADC 进行校准，校准完成后硬件清除 ADCAL 位，软件即

可触发以进行转换。

每次校准后，校准值会被存放至 ADC_CALVAL 的低 7 位中，这个校准值自动反馈回 ADC 内部，以消除电容误差。该校准值的存放不会置位 OCCE 标志，不会产生中断或 DMA 请求。

注意： 只能在分辨率 12 位时进行校准，校准完成后才可切换分辨率，切换分辨率后等待 RDY 标志置起即可触发。

图 18-3 ADC上电与校准



18.4.2.2 触发

ADC 触发分为普通通道触发与抢占通道触发，普通通道触发引发普通通道转换，抢占通道触发引发抢占通道转换。针对外部触发来源的有效极性是由 ADC_CTRL2 的 OCETE 与 PCETE 选择，ADC 会检测触发来源并响应转换。

触发来源可分为软件写寄存器触发（ADC_CTRL2 的 OCSWTRG 与 PCSWTRG）以及外部触发，外部触发包含定时器触发与引脚触发，由 ADC_CTRL2 的 OCTESEL 与 PCTESEL 选择触发来源，如下表所示。

表 18-1普通通道与抢占通道触发来源

OCTESEL	触发来源	PCTESEL	触发来源
00000	TMR1_TRGOUT event	00000	TMR1_CH2 event
00001	TMR1_CH4 event	00001	TMR1_CH3 event
00010	TMR2_TRGOUT event	00010	TMR2_CH4 event
00011	TMR3_TRGOUT event	00011	TMR3_CH4 event
00100	TMR9_TRGOUT event	00100	TMR9_CH1 event
00101	TMR1_CH1 event	00101	TMR6_TRGOUT event
00110	EXINT line11 External pin	00110	EXINT line15 External pin
00111	OCSWTRG	00111	PCSWTRG
01000~1111	保留	01000~1111	保留

18.4.2.3 采样与转换时序

用户可于 ADC_SPT1 与 ADC_SPT2 的 CSPTx 配置各个通道（ADC_INx）的采样周期。通道 0/1/10/11/12/13 为快速通道，支持最小 2.5 个周期采样时间；其他通道为慢速通道，支持最小 6.5 个周期采样时间。藉由配置 ADC_CTRL1 的 CRSEL 可改变转换数据的分辨率，较低的分辨率所需的转换时间较短。一次转换所需的时间可利用以下公式推得：

$$\text{一次转换所需的时间}(\text{ADCCLK 的周期}) = \text{采样时间} + \text{分辨率位数} + 0.5$$

示例：

CSPTx 选择 2.5 周期，CRSEL 选择 12 位，一次转换需要 2.5+12.5=15 个 ADCCLK 周期。
CSPTx 选择 6.5 周期，CRSEL 选择 10 位，一次转换需要 6.5+10.5=17 个 ADCCLK 周期。

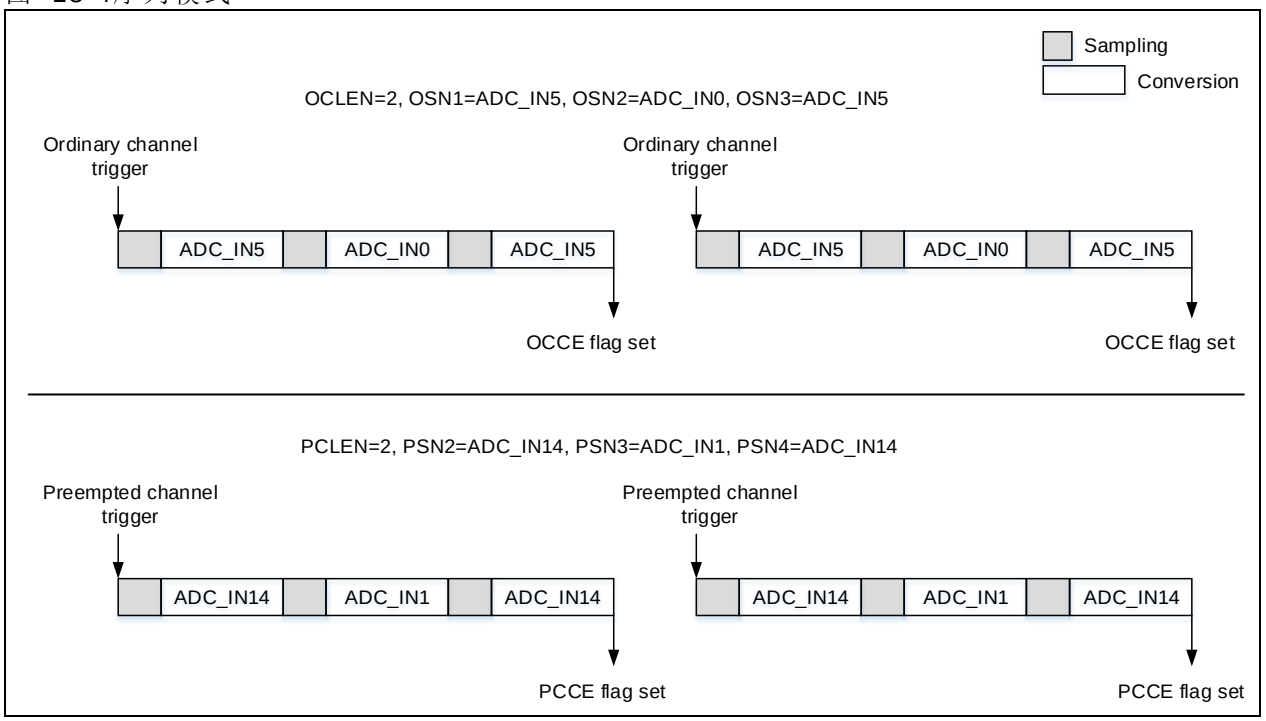
18.4.3 转换顺序管理

默认模式下，每次触发只会转换单个通道，即 OSN1（普通触发）或 PSN4（抢占触发）记录的通道。下面介绍不同的转换顺序模式，即可使多个通道以特定顺序做转换。

18.4.3.1 序列模式

使能 ADC_CTRL1 的 SQEN，即开启序列模式，用户于 ADC_OSQx 配置普通通道顺序与总数，于 ADC_PSQ 配置抢占通道顺序与总数，开启序列模式后，一次触发将序列中的通道依序转换一次。普通通道从 OSN1 开始转换起，抢占通道是从 PSNx 开始转换起，x=4-PCLEN，下图示范了序列模式的行为。

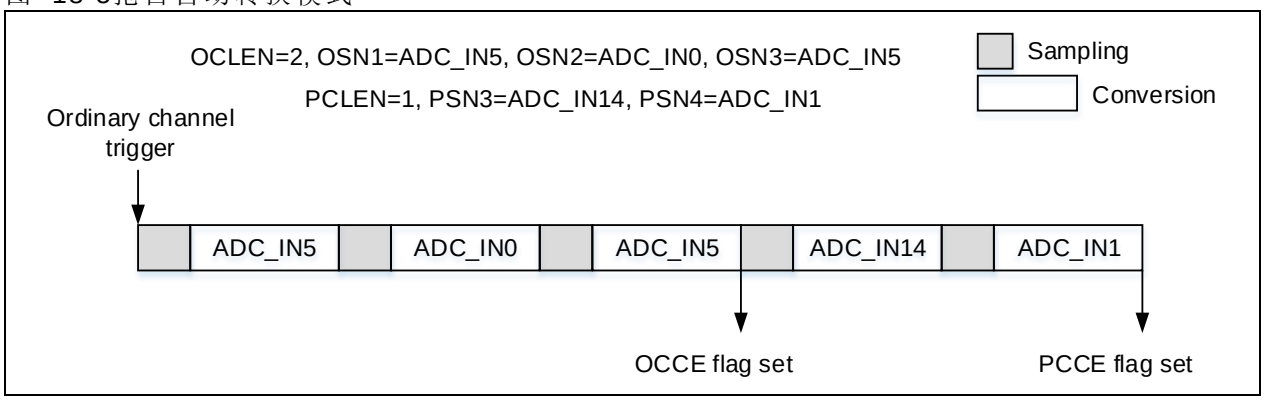
图 18-4序列模式



18.4.3.2 抢占自动转换模式

使能 ADC_CTRL1 的 PCAUTOEN，即开启抢占自动转换模式，当普通通道转换完成后，抢占通道将自动接续着转换。可与序列模式共用，当普通通道序列完成后，即会自动开始抢占序列的转换。下图示范了与序列模式共用的抢占自动转换模式行为。

图 18-5抢占自动转换模式

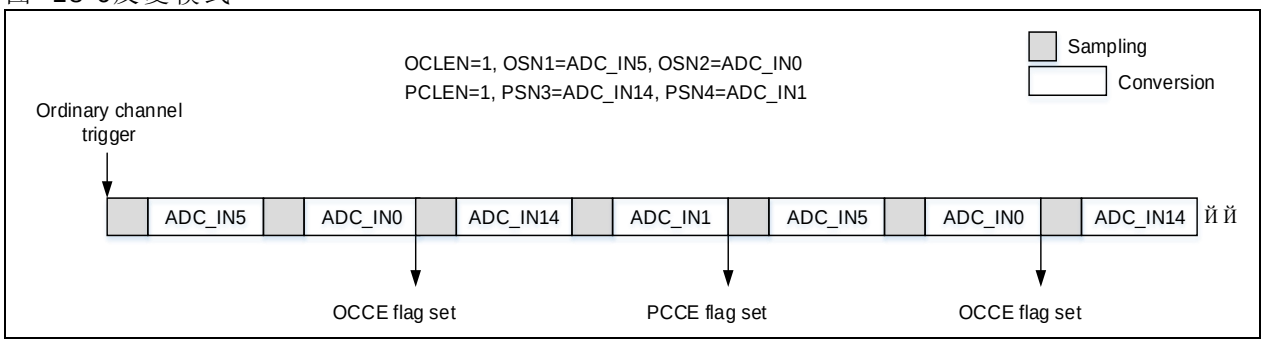


18.4.3.3 反复模式

使能 ADC_CTRL2 的 RPEN，即开启反复模式。当普通通道检测到触发后就即会反复不断地转换。可与序列模式下的普通通道转换共用，将反复地转换普通通道序列。也可与抢占自动转换模式共用，将依次反复地转换普通通道序列与抢占通道序列。下图示范了与序列模式及抢占自动转换模式共用的反复模式行

为。

图 18-6反复模式



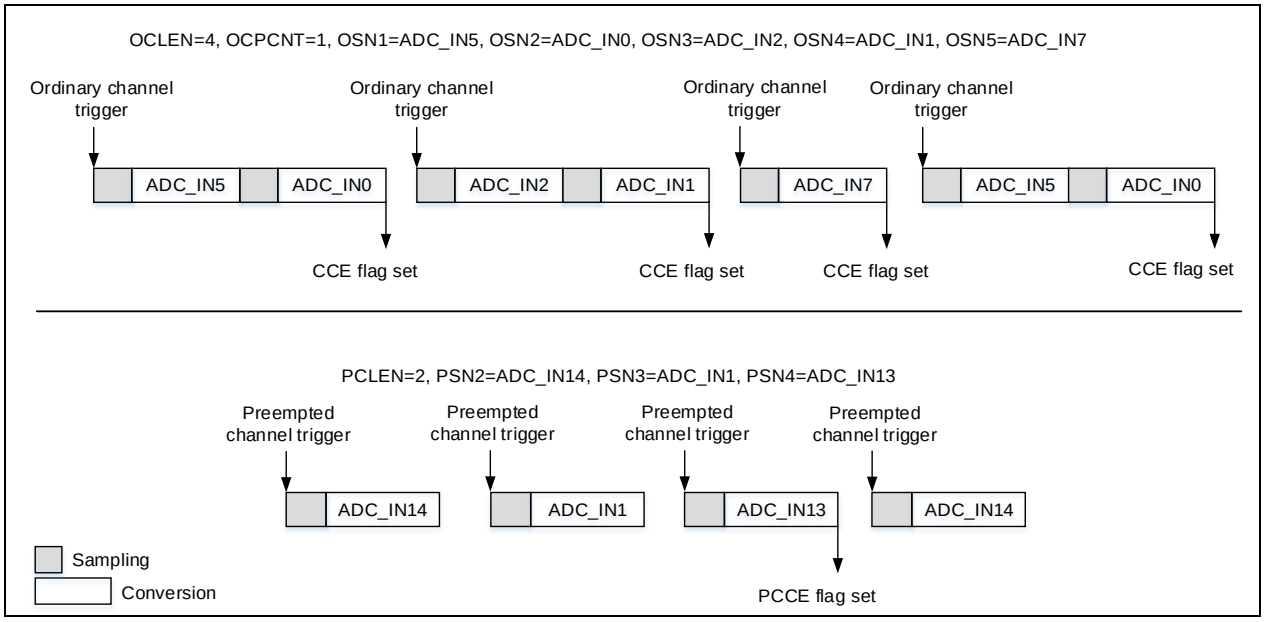
18.4.3.4 分割模式

使能 ADC_CTRL1 的 OCPEN，即开启普通通道的分割模式，此模式将 ADC_OSQ1 的 OCLEN 的序列长度分割成长度较小的子组别，子组别的通道数于 ADC_CTRL1 的 OCPCNT 配置，一次触发将转换子组别中的所有通道。每次触发会依序选择不同的子组别。

使能 ADC_CTRL1 的 PCPEN，即开启抢占通道的分割模式，此模式将 ADC_OSQ1 的 PCLEN 的序列长度分割成只有一个通道的子组别，一次触发将转换子组别中的通道。每次触发会依序选择不同的子组别。

分割模式与反复模式不可共用。下图分别示范了普通分割与抢占分割模式的行为。

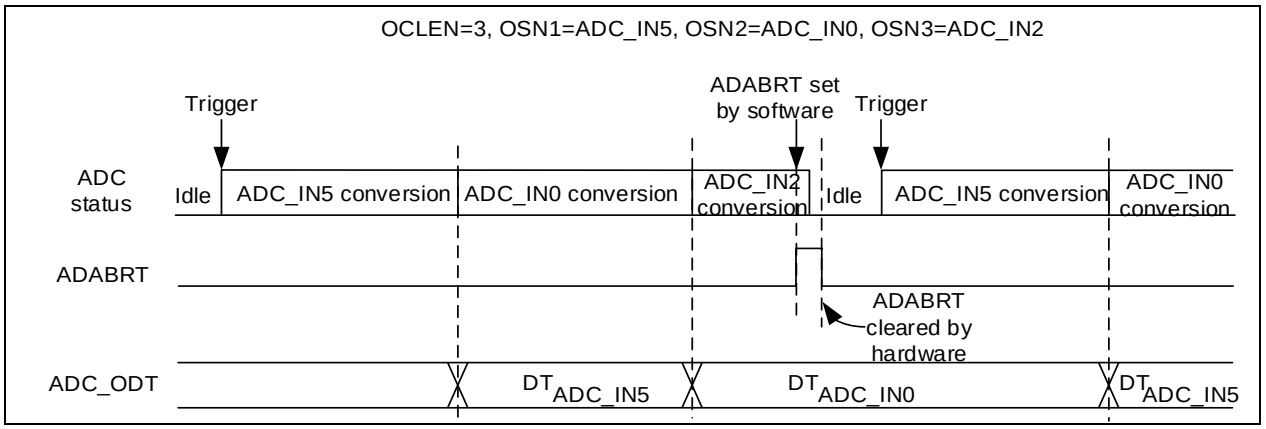
图 18-7分割模式



18.4.4 转换中止

用户可利用 ADC_CTRL2 的 ADABRT 使 ADC 停止转换，转换顺序回归第一个通道，用户即可重新编排通道顺序，再次触发 ADC 将从头开始新序列的转换。若 ADABRT 已置位，需等待 ADABRT 被硬件清零后进行其他 ADC 操作。图 18-8 置位 ADABRT 时的时序图。

图 18-8 ADABRT时序



18.4.5 过采样器

一次过采样转换数据是透过转换多次相同通道，累加转换数据后作平均实现的。

- 由ADC_OVSP的OSRSEL选择过采样率，此位用来定义过采样倍数，通过多次转换同个通道实现。
- 由ADC_OVSP的OSSSEL选择过采样移位，此位用来定义平均系数，通过右移位实现。

若平均后数据大于 16 位，只取靠右 16 位数据，放入 16 位数据寄存器。如表 18-2 所示。

示例：

若 OSRSEL 选择 4 倍，一次过采样转换同个通道转换 4 次，并将这 4 次转换数据累加。若 OSSSEL 选择 6 位，累加数据除以 2^6，以四舍五入进位。

表 18-2 最大累加数据与过采样倍数及位移系数关系

过采样率	2x	4x	8x	16x	32x	64x	128x	256x
最大累加数据	0x1FFE	0x3FFC	0x7FF8	0xFFF0	0x1FFE0	0x3FFC0	0x7FF80	0xFFF00
不移位	0x1FFE	0x3FFC	0x7FF8	0xFFF0	0xFFE0	0xFFC0	0xFF80	0xFF00
移 1 位	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFF0	0xFFE0	0xFFC0	0xFF00
移 2 位	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFF0	0xFFE0	0xFFC0
移 3 位	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFF0	0xFFE0
移 4 位	0x0200	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFF0
移 5 位	0x0100	0x0200	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8
移 6 位	0x0080	0x0100	0x0200	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC
移 7 位	0x0040	0x0080	0x0100	0x0200	0x0400	0x0800	0x0FFF	0x1FFE
移 8 位	0x020	0x0040	0x0080	0x0100	0x0200	0x0400	0x0800	0x0FFF

使用过采样时，忽视 DTALIGN 与 PCDTOx，数据一律靠右摆放。切换 CRSEL 位改变过采样转换中每一次通道转换结果的分辨率，但是不改变过采样数据运算方式。

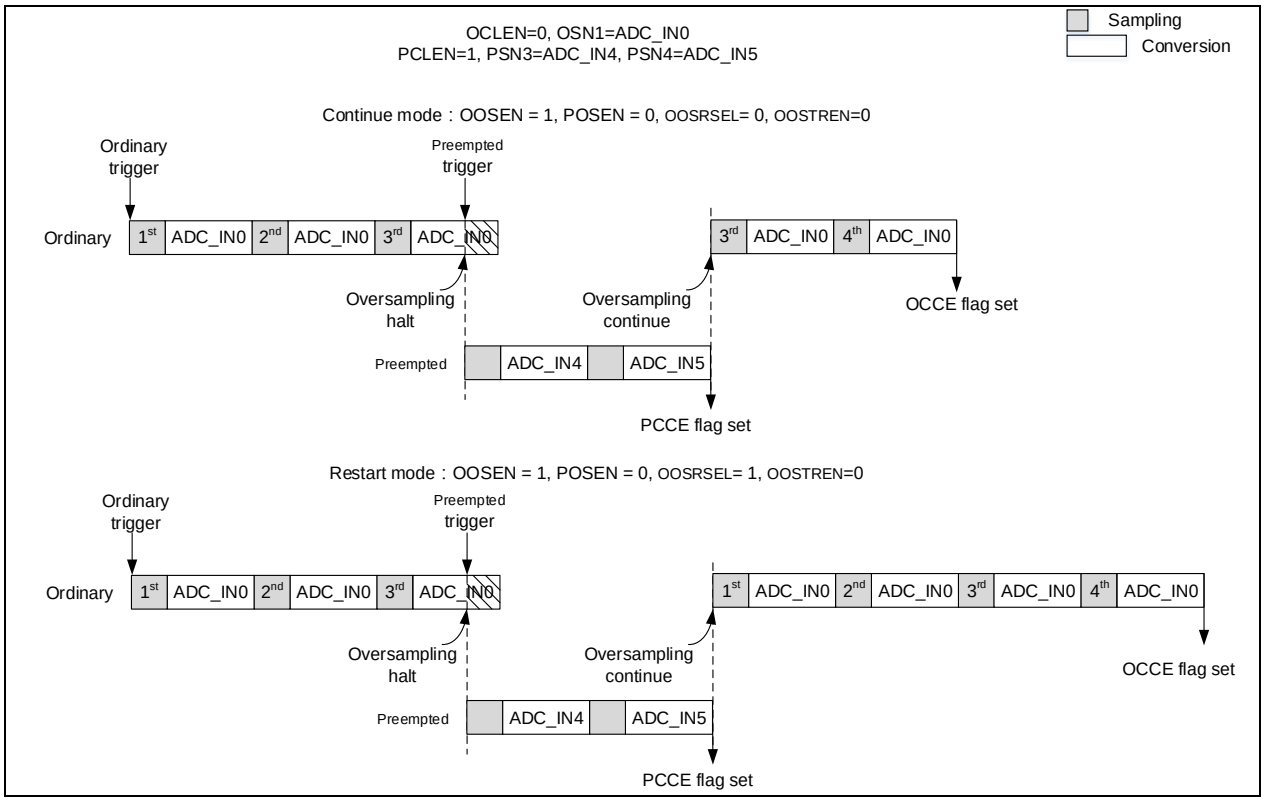
18.4.5.1 普通通道过采样

由 ADC_OVSP 的 OOSRSEL 控制普通过采样被打断后恢复的动作。

- OOSRSEL=0：接续模式，普通过采样中途被抢占通道转换插入后，保留已累加的数据，再次开始转换时将从打断处转换。
- OOSRSEL=1：重转模式。普通过采样中途被抢占通道转换插入后，累加的数据被清空。再次开始转换时是重新该通道的过采样转换。

下图以 4x 过采样率与序列模式下，示范了普通过采样接续模式与重转模式的差别。

图 18-9普通过采样重转模式选择

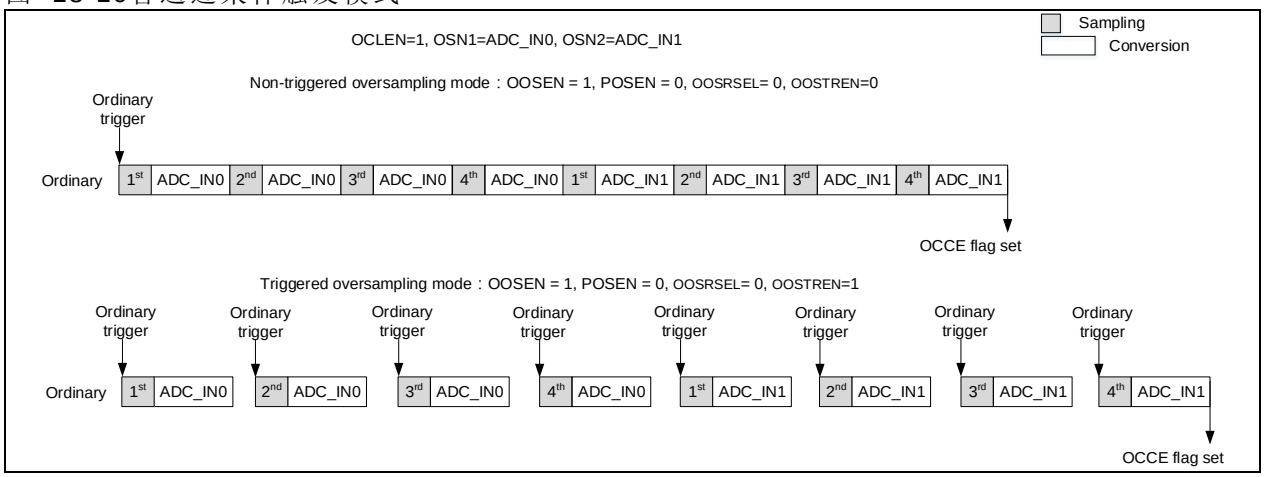


使能 ADC_OVSP 的 OOSTREN 即可使用触发模式。用户须触发普通过采样转换中每一次的普通通道转换。此模式下，中途被抢占通道触发打断后，须重新触发普通通道才会恢复转换普通通道过采样。

当触发模式与转换顺序管理模式搭配使用时，触发方式遵循触发模式，转换完成标志则遵循转换顺序管理模式。下图以 4x 过采样率与序列模式下，示范了普通过采样触发模式与恢复模式共用。

注意：触发模式与反复模式不可共用。

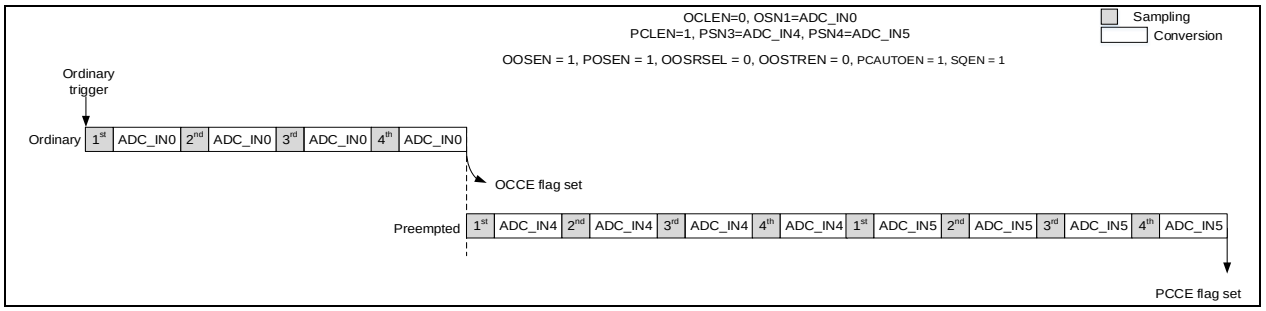
图 18-10普通过采样触发模式



18.4.5.2 抢占通道过采样

抢占过采样可与普通过采样同时使用，也可分别使用。抢占过采样不影响到普通过采样的各种模式。下图以 4x 过采样率与抢占自动转换模式下，示范了抢占过采样与普通过采样触发模式共用。

图 18-11抢占过采样



18.4.6 数据管理

普通通道转换完成后数据存储于普通数据寄存器（ADC_ODT），抢占通道转换完成后数据存储于抢占数据寄存器 x（ADC_PDTx）。

18.4.6.1 数据内容处理

由 ADC_CTRL2 的 DTALIGN 选择转换数据靠右或是靠左对齐放置于数据寄存器，除此之外，抢占通道的数据还会减去抢占数据偏移寄存器 x（ADC_PCDTOx）的偏移量，因此抢占通道数据有可能为负值，以 SIGN 作为符号。

分辨率 CRSEL 为 6 位时，数据存储方式以字节为基准摆放，其余皆以半字为基准摆放。如下图所示。

图 18-12数据内容处理

Ordinary channel data 12 bits

Right-alignment

0	0	0	0	DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]

Left-alignment

DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]	0	0	0	0

Ordinary channel data 6 bits

Right-alignment

0	0	0	0	0	0	0	0	0	0	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]

Left-alignment

0	0	0	0	0	0	0	0	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]	0	0

Preempted channel data 12 bits

Right-alignment

SIGN	SIGN	SIGN	SIGN	DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]

Left-alignment

SIGN	DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]	0	0	0

Preempted channel data 6 bits

Right-alignment

SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]

Left-alignment

SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	SIGN	DT[5]	DT[4]	DT[3]	DT[2]	DT[1]	DT[0]	0

18.4.6.2 数据获取

普通通道转换数据可藉由 CPU 或 DMA 读取普通数据寄存器（ADC_ODT）获得。抢占通道数据只可藉由 CPU 读取抢占数据寄存器 x（ADC_PDTx）获得。

由 ADC_CTRL2 的 EOCSFEN 选择普通通道转换结束标志置位于序列结束或是每次普通数据寄存器更新时。

使能 ADC_CTRL2 的 OCDMAEN 后，ADC 会在每次普通数据寄存器更新时请求 DMA。

当 EOCSFEN 或是 OCDMAEN 置位时，ADC 会自动开启溢出检测。若发生溢出事件，OCCO 溢出事件标志置起，ADC 转换停下，数据寄存器存储着最后一个有效数据。若使用 DMA，DMA 请求持续置起以请求 DMA 读取最后的有效数据。软件需将 OCCO 清除，并且重新触发 ADC，ADC 将从有效数据的下一个通道开始转换，如此即使中途发生溢出事件，所有被读取的数据皆是有效且按照顺序的。

配置 ADC_CTRL2 的 OCDRCEN 来选择是否要在 DMA 传输数量寄存器归零后持续请求 DMA。

18.4.7 电压监测

使能 ADC_CTRL1 的 OCVMEN（普通通道）或 PCVMEN（抢占通道）即可通过对转换结果的判定来实现电压监测。

当转换结果大于高边界 ADC_VMHB[11:0]寄存器或是小于低边界 ADC_VMLB[11:0]寄存器时，电压监测超出标志 VMOR 会置起。

若选择 10 位分辨率，ADC_VMHB[11:10]和 ADC_VMLB[11:10]要设置成 2’b00。

若选择 8 位分辨率，ADC_VMHB[11:8]和 ADC_VMLB[11:8]要设置成 4’b0000。

若选择 6 位分辨率，ADC_VMHB[11:6]和 ADC_VMLB[11:6]要设置成 6’b000000。

透过 VMSGEN 选择对单一特定通道或是所有通道监测。对单一通道监测的话，由 VMCSEL 配置通道。

电压监测一律以转换的原始数据与 12 位边界寄存器做比较，无视 CRSEL、PCDTOx 与 DTALIGN 位的设定。

若是使用过采样器，则是以 ADC_VMHB[15:0]与 ADC_VMLB[15:0]完整的 16 位寄存器与过采样数据作比较。

18.4.7.1 状态标志与中断

ADC 拥有自己的状态寄存器（ADC_STS）：准备就绪标志（RDY）、溢出事件标志（OCCO）、普通通道转换开始标志（OCCS）、抢占通道转换开始标志（PCCS）、抢占通道组转换结束标志（PCCE）、普通通道转换结束标志（OCCE）及电压监测超出标志（VMOR）。

其中溢出事件标志、抢占通道组转换结束标志、普通通道转换结束标志及电压监测超出标志拥有对应中断使能位，只要将中断使能，标志置起时便会对 CPU 发出中断。

18.5 ADC寄存器

下表列出了 ADC 寄存器的映像和复位值。

必须以字(32 位) 的方式操作这些外设寄存器。

表 18-3 ADC寄存器映像和复位值

寄存器简称	基址偏移量	复位值
ADC1_STS	0x000	0x0000 0000
ADC1_CTRL1	0x004	0x0000 0000
ADC1_CTRL2	0x008	0x0000 0000
ADC1_SPT1	0x00C	0x0000 0000
ADC1_SPT2	0x010	0x0000 0000
ADC1_PCDTO1	0x014	0x0000 0000
ADC1_PCDTO2	0x018	0x0000 0000
ADC1_PCDTO3	0x01C	0x0000 0000
ADC1_PCDTO4	0x020	0x0000 0000
ADC1_VMHB	0x024	0x0000 FFFF
ADC1_VMLB	0x028	0x0000 0000
ADC1_OSQ1	0x02C	0x0000 0000
ADC1_OSQ2	0x030	0x0000 0000
ADC1_OSQ3	0x034	0x0000 0000
ADC1_PSQ	0x038	0x0000 0000
ADC1_PDT1	0x03C	0x0000 0000
ADC1_PDT2	0x040	0x0000 0000
ADC1_PDT3	0x044	0x0000 0000
ADC1_PDT4	0x048	0x0000 0000
ADC1_ODT	0x04C	0x0000 0000

ADC1_SPT3	0x050	0x0000 0000
ADC1_OSQ4	0x054	0x0000 0000
ADC1_OSQ5	0x058	0x0000 0000
ADC1_OSQ6	0x05C	0x0000 0000
ADC1_OVSP	0x080	0x0000 0000
ADC1_CALVAL	0x0B4	0x0000 0000
ADC1_MISC	0x0D0	0x0000 000C
ADC_CCTRL	0x304	0x0000 0000

18.5.1 ADC状态寄存器（ADC_STS）

访问：字访问

域	简称	复位值	类型	功能
位 31：7	保留	0x0000000	resd	请保持默认值。
位 6	RDY	0x0	ro	ADC 准备就绪标志（ADC ready to conversion flag） 只读位，该位在 ADC 上电完毕后由硬件置位。 0：未就绪； 1：已就绪。
位 5	OCCO	0x0	rw0c	普通通道转换溢出标志 (Ordinary channel conversion overflow flag) 该位被硬件置起，由软件将其清零（对自身写零） 0：未发生溢出 1：发生溢出 注： 溢出检测仅在使能 DMA 传输或者 EOCSFEN =1 时有效
位 4	OCCS	0x0	rw0c	普通通道转换开始标志 （Ordinary channel conversion start flag） 该位被硬件置起，由软件将其清零（对自身写零）。 0：未开始； 1：已开始。
位 3	PCCS	0x0	rw0c	抢占通道转换开始标志 （Preempted channel conversion start flag） 该位被硬件置起，由软件将其清零（对自身写零）。 0：未开始； 1：已开始。
位 2	PCCE	0x0	rw0c	抢占通道组转换结束标志 （Preempted channels conversion end flag） 该位被硬件置起，由软件将其清零（对自身写零）。 0：未结束； 1：已结束。
位 1	OCCE	0x0	rw0c	普通通道转换结束标志 （Ordinary channels conversion end flag） 该位被硬件置起，由软件将其清零（对自身写零），或由读取 ADC_ODT 寄存器清零。 0：未结束； 1：已结束。
位 0	VMOR	0x0	rw0c	电压监测超出范围标志 （Voltage monitoring out of range flag） 该位被硬件置起，由软件将其清零（对自身写零）。 0：无超出； 1：有超出。

18.5.2 ADC控制寄存器1（ADC_CTRL1）

访问：字访问

域	简称	复位值	类型	功能
位 31：27	保留	0x00	resd	请保持默认值。
位 26	OCCOIE	0x0	rw	普通通道转换溢出中断使能 (Ordinary channel conversion overflow interrupt enable) 0：关闭； 1：开启。
位 25:24	CRSEL	0x0	rw	转换分辨率选择 (Conversion resolution select) 00：12 位； 01：10 位； 10：8 位； 11：6 位。
位 23	OCVMEN	0x0	rw	普通通道的电压监测使能 （Voltage monitoring enable on ordinary channels） 0：关闭； 1：开启。
位 22	PCVMEN	0x0	rw	抢占通道的电压监测使能 （Voltage monitoring enable on preempted channels） 0：关闭； 1：开启。
位 21：16	保留	0x0	resd	请保持默认值。
位 15：13	OCPCNT	0x0	rw	分割模式下每次触发转换的普通通道个数 （Partitioned mode conversion count of ordinary channels） 000：1 个通道； 001：2 个通道； …… 111：8 个通道。 注： 抢占组在分割模式下每次触发固定只转换一个通道。
位 12	PCPEN	0x0	rw	抢占通道上的分割模式使能（Partitioned mode enable on preempted channels） 0：关闭； 1：开启。
位 11	OCPEN	0x0	rw	普通通道上的分割模式使能（Partitioned mode enable on ordinary channels） 该位由软件设置和清除，用于开启或关闭普通通道组上的分割模式 0：关闭； 1：开启。
位 10	PCAUTOEN	0x0	rw	普通组转换结束后的抢占组自动转换使能（Preempted group automatic conversion enable after ordinary group） 0：关闭； 1：开启。
位 9	VMSGEN	0x0	rw	单个通道的电压监测使能 （Voltage monitoring enable on a single channel） 0：关闭（电压监测所有通道）； 1：开启（电压监测单一通道）。

位 8	SQEN	0x0	rw	序列模式使能 （Sequence mode enable） 0：关闭（转换选择的单一通道）； 1：开启（转换设定的多个通道）。 注：如果开启多通道模式，且开启了 CCEIEN 或 PCCEIEN 位，则只在最后一个通道转换完毕后才会产生 OCCE 或 PCCE 中断。
位 7	PCCEIEN	0x0	rw	抢占通道组转换结束中断使能 （conversion end interrupt enable for Preempted channels） 0：关闭； 1：开启。
位 6	VMORIEN	0x0	rw	电压监测超出范围中断使能 （Voltage monitoring out of range interrupt enable） 0：关闭； 1：开启。
位 5	OCCEIEN	0x0	rw	普通通道转换结束中断使能 （Ordinary channel conversion end interrupt enable） 0：关闭； 1：开启。
位 4：0	VMCSEL	0x00	rw	电压监测通道选择（Voltage monitoring channel select） 仅在 VMSGEN 开启时有效。 00000：ADC_IN0 通道； 00001：ADC_IN1 通道； …… 10001：ADC_IN17 通道； 10100：ADC_IN20 通道； 10101：ADC_IN21 通道； …… 11011：ADC_IN27 通道。 11011~11111：未用，禁止配置。

18.5.3 ADC控制寄存器2（ADC_CTRL2）

访问：字访问

域	简称	复位值	类型	功能
位 30：26	保留	0x00	resd	请保持默认值。
位 30	OCSWTRG	0x0	rw	软件触发普通通道转换（Conversion trigger by software of ordinary channels） 0：不触发； 1：触发转换（可由软件清除，或在转换开始后由硬件自动清除）。
位 29:28	OCETE	0x0	rw	普通通道组的外部触发边沿选择（Ordinary channel’s external trigger edge select） 00：禁止边沿触发； 01：上升沿触发； 01：下降沿触发； 11：任意边沿触发。
位 31 位 27:24	OCTESEL	0x00	rw	普通通道组转换的触发事件选择（Ordinary channel’s conversion trigger event select） 注： 各 bit 的定义参考 18.4.2.2 小节

位 22	PCSWTRG	0x0	rw	软件触发抢占通道转换（Conversion trigger by software of preempted channels） 0：不触发； 1：触发转换（可由软件清除，或在转换开始后由硬件自动清除）。
位 21:20	PCETE	0x0	rw	抢占通道组的外部触发边沿选择（Preempted channel’s external trigger edge select） 00：禁止边沿触发； 01：上升沿触发； 01：下降沿触发； 11：任意边沿触发。
位 23 位 19:16	PCTESEL	0x00	rw	抢占通道组转换的触发事件选择（Preempted channel’s conversion trigger event select） 注： 各 bit 的定义参考 18.4.2.2 小节
位 15:12	保留	0x0	resd	请保持默认值。
位 11	DTALIGN	0x0	rw	数据对齐方式 （Data alignment） 0：右对齐； 1：左对齐。
位 10	EOCSFEN	0x0	rw	每个普通通道转换置位 OCCE 标志使能（Each ordinary channel conversion set OCCE flag enable） 0：关闭； 1：开启。 注： 开启此位时，溢出检测被自动使能。
位 9	OCDRCEN	0x0	rw	普通通道数据的 DMA 请求接续使能 (Ordinary channel’s DMA request continuation enable for independent mode) 0：关闭（传输完 DMA 设定个数后，普通通道转换完毕不会再产生 DMA 请求）； 1：开启（不关心 DMA 设定的个数，每个普通通道转换完毕均产生 DMA 请求）。 注： 此位仅作用于设定为非主从模式，且 OCDMAEN = 1 时。
位 8	OCDMAEN	0x0	rw	普通通道转换数据的 DMA 传输使能 （DMA transfer enable of ordinary channels） 0：关闭； 1：开启。
位 7：5	保留	0x0	resd	请保持默认值。
位 4	ADABRT	0x0	rw	ADC 转换中止 (ADC conversion abort) 0：无中止转换命令执行； 1：中止当前正在执行的转换。 注： 当转换已中止时硬件将该位清零。该位被清零后，可重新进行触发转换。若 ADABRT 已置位，需等待 ADABRT 被硬件清零后进行其他 ADC 操作。
位 3	ADCALINIT	0x0	rw	A/D 初始化校准 （initialize A/D calibration） 该位由软件设置并由硬件清除。在校准寄存器被初始化后该位将被清除。 0：校准寄存器无初始化执行或初始化结束； 1：校准寄存器初始化或初始化进行中。

位 2	ADCAL	0x0	rw	A/D 校准 （A/D Calibration） 0：无校准执行或校准结束； 1：开始校准或校准进行中。
位 1	RPEN	0x0	rw	反复模式使能（Repeat mode enable） 0：关闭 SQEN=0 时，每次触发转换单个通道，SQEN=1 时，每次触发转换一组通道； 1：开启 SQEN =0 时，一次触发后将反复转换单个通道，SQEN =1 时，一次触发后将反复转换一组通道。直到 ADCEN 被清零。
位 0	ADCEN	0x0	rw	A/D 转换器使能 （A/D converter enable） 0：关闭（ADC 进入断电模式）； 1：开启。 注： 当该位为关闭状态时，写入开启命令将把 ADC 从断电模式下唤醒。 应用程序需注意，在转换器上电至转换开始有一个延迟 t_{STAB}。

18.5.4 ADC采样时间寄存器1（ADC_SPT1）

访问：字访问

域	简称	复位值	类型	功能
位 31：24	保留	0x00	resd	请保持默认值。
位 23：21	CSPT17	0x0	rw	选择 ADC_IN17 通道的采样时间 （Selection sample time of channel ADC_IN17） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 20：18	CSPT16	0x0	rw	选择 ADC_IN16 通道的采样时间 （Selection sample time of channel ADC_IN16） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。

位 17：15	CSPT15	0x0	rw	选择 ADC_IN15 通道的采样时间 （Selection sample time of channel ADC_IN15） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 14：12	CSPT14	0x0	rw	选择 ADC_IN14 通道的采样时间 （Selection sample time of channel ADC_IN14） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 11：9	CSPT13	0x0	rw	选择 ADC_IN13 通道的采样时间 （Selection sample time of channel ADC_IN13） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 8：6	CSPT12	0x0	rw	选择 ADC_IN12 通道的采样时间 （Selection sample time of channel ADC_IN12） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 5：3	CSPT11	0x0	rw	选择 ADC_IN11 通道的采样时间 （Selection sample time of channel ADC_IN11） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。

位 2：0	CSPT10	0x0	rw	选择 ADC_IN10 通道的采样时间 （Selection sample time of channel ADC_IN10） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。

18.5.5 ADC采样时间寄存器2（ADC_SPT2）

访问：字访问

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	请保持默认值。
位 29：27	CSPT9	0x0	rw	选择 ADC_IN9 通道的采样时间 （Selection sample time of channel ADC_IN9） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 26：24	CSPT8	0x0	rw	选择 ADC_IN8 通道的采样时间 （Selection sample time of channel ADC_IN8） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 23：21	CSPT7	0x0	rw	选择 ADC_IN7 通道的采样时间 （Selection sample time of channel ADC_IN7） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。

位 20：18	CSPT6	0x0	rw	选择 ADC_IN6 通道的采样时间 （Selection sample time of channel ADC_IN6） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 17：15	CSPT5	0x0	rw	选择 ADC_IN5 通道的采样时间 （Selection sample time of channel ADC_IN5） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 14：12	CSPT4	0x0	rw	选择 ADC_IN4 通道的采样时间 （Selection sample time of channel ADC_IN4） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 11：9	CSPT3	0x0	rw	选择 ADC_IN3 通道的采样时间 （Selection sample time of channel ADC_IN3） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 8：6	CSPT2	0x0	rw	选择 ADC_IN2 通道的采样时间 （Selection sample time of channel ADC_IN2） 000：保留； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。

位 5：3	CSPT1	0x0	rw	选择 ADC_IN1 通道的采样时间 （Selection sample time of channel ADC_IN1） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 2：0	CSPT0	0x0	rw	选择 ADC_IN0 通道的采样时间 （Selection sample time of channel ADC_IN0） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。

18.5.6 ADC抢占通道数据偏移寄存器x（ADC_PCDTOx）（x=1..4）

访问：字访问

域	简称	复位值		功能
位 31：12	保留	0x00000	resd	请保持默认值。
位 11：0	PCDTOx	0x000	rw	抢占通道 x 的数据偏移量设定 （Data offset for Preempted channel x） ADC_PDTx 内存放的转换数据 = 原始转换数据 - ADC_PCDTOx

18.5.7 ADC电压监测高边界寄存器（ADC_VMHB）

访问：字访问

域	简称	复位值	类型	功能
位 31：16	保留	0x00000	resd	请保持默认值。
位 15：0	VMHB	0xFFFF	rw	电压监测高边界设定（Voltage monitoring high boundary）

18.5.8 ADC电压监测低边界寄存器（ADC_VMLB）

访问：字访问

域	简称	复位值	类型	功能
位 31：16	保留	0x00000	resd	请保持默认值。
位 15：0	VMLB	0x0000	rw	电压监测低边界设定（Voltage monitoring low boundary）

18.5.9 ADC普通序列寄存器1（ADC_OSQ1）

访问：字访问

域	简称	复位值	类型	功能
位 31：24	保留	0x00	resd	请保持默认值。

位 23：20	OCLEN	0x00	rw	普通转换序列长度 （Ordinary conversion sequence length） 0000：1 个转换； 0001：2 个转换； …… 11111：32 个转换。
位 19：15	OSN16	0x00	rw	普通序列中第 16 个转换通道的编号 （number of 16th conversion in ordinary sequence）
位 14：10	OSN15	0x00	rw	普通序列中第 15 个转换通道的编号 （number of 15th conversion in ordinary sequence）
位 9：5	OSN14	0x00	rw	普通序列中第 14 个转换通道的编号 （number of 14th conversion in ordinary sequence）
位 4：0	OSN13	0x00	rw	普通序列中第 13 个转换通道的编号 （number of 13th conversion in ordinary sequence） 注：编号可设定 0~17、20~27，示例：设定为 3 就代表第 13 个转换的是 ADC_IN3 通道。

18.5.10 ADC普通序列寄存器2（ADC_OSQ2）

访问：字访问

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	请保持默认值。
位 29：25	OSN12	0x00	rw	普通序列中第 12 个转换通道的编号（number of 12th conversion in ordinary sequence）
位 24：20	OSN11	0x00	rw	普通序列中第 11 个转换通道的编号（number of 11th conversion in ordinary sequence）
位 19：15	OSN10	0x00	rw	普通序列中第 10 个转换通道的编号（number of 10th conversion in ordinary sequence）
位 14：10	OSN9	0x00	rw	普通序列中第 9 个转换通道的编号（number of 8th conversion in ordinary sequence）
位 9：5	OSN8	0x00	rw	普通序列中第 8 个转换通道的编号 （number of 8th conversion in ordinary sequence）
位 4：0	OSN7	0x00	rw	普通序列中第 7 个转换通道的编号 （number of 7th conversion in ordinary sequence） 注：编号可设定 0~17、20~27，示例：设定为 8 就代表第 7 个转换的是 ADC_IN8 通道。

18.5.11 ADC普通序列寄存器3（ADC_OSQ3）

访问：字访问

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	请保持默认值。
位 29：25	OSN6	0x00	rw	普通序列中第 6 个转换通道的编号（number of 6th conversion in ordinary sequence）
位 24：20	OSN5	0x00	rw	普通序列中第 5 个转换通道的编号（number of 5th conversion in ordinary sequence）

位 19：15	OSN4	0x00	rw	普通序列中第 4 个转换通道的编号（number of 4th conversion in ordinary sequence）
位 14：10	OSN3	0x00	rw	普通序列中第 3 个转换通道的编号（number of 3rd conversion in ordinary sequence）
位 9：5	OSN2	0x00	rw	普通序列中第 2 个转换通道的编号 （number of 2nd conversion in ordinary sequence）
位 4：0	OSN1	0x00	rw	普通序列中第 1 个转换通道的编号 （number of 1st conversion in ordinary sequence） 注：编号可设定 0~17、20~27，示例：设定为 17 就代表第 1 个转换的是 ADC_IN17 通道。

18.5.12 ADC抢占序列寄存器（ADC_PSQ）

访问：字访问

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	请保持默认值。
位 21：20	PCLEN	0x0	rw	抢占转换序列长度 （Preempted conversion sequence length） 00：1 个转换； 01：2 个转换； 10：3 个转换； 11：4 个转换。
位 19：15	PSN4	0x00	rw	抢占序列中第 4 个转换通道的编号（number of 4th conversion in Preempted sequence）
位 14：10	PSN3	0x00	rw	抢占序列中第 3 个转换通道的编号（number of 3rd conversion in Preempted sequence）
位 9：5	PSN2	0x00	rw	抢占序列中第 2 个转换通道的编号（number of 2nd conversion in Preempted sequence）
位 4：0	PSN1	0x00	rw	抢占序列中第 1 个转换通道的编号（number of 1st conversion in Preempted sequence） 注： 编号可设定 0~17、20~27，比如设定为 3 时其代表的就是 ADC_IN3 通道。 若 PCLEN 小于 4，则转换的序列顺序是从（4-PCLEN）开始。例如：ADC_PSQ[21：0] = 10 00110 00101 00100 00011，意味着扫描转换将按下列通道顺序执行：4、5、6，而不是 3、4、5。

18.5.13 ADC抢占数据寄存器x（ADC_PDTx）（x= 1..4）

访问：字访问

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	请保持默认值。
位 15：0	PDTx	0x0000	ro	抢占通道的转换数据（Conversion data of preempted channel）

18.5.14 ADC普通数据寄存器（ADC_ODT）

访问：字访问

域	简称	复位值	类型	功能

位 31：16	保留	0x0000	resd	请保持默认值。
位 15：0	ODT	0x0000	ro	普通通道的转换数据（Conversion data of ordinary channel）

18.5.15 ADC采样时间寄存器3（ADC_SPT3）

访问：字访问

域	简称	复位值	类型	功能
位 31：24	保留	0x0	resd	请保持默认值。
位 23：21	CSPT27	0x0	rw	选择 ADC_IN27 通道的采样时间 （Selection sample time of channel ADC_IN27） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 20：18	CSPT26	0x0	rw	选择 ADC_IN26 通道的采样时间 （Selection sample time of channel ADC_IN26） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 17：15	CSPT25	0x0	rw	选择 ADC_IN25 通道的采样时间 （Selection sample time of channel ADC_IN25） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 14：12	CSPT24	0x0	rw	选择 ADC_IN24 通道的采样时间 （Selection sample time of channel ADC_IN24） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。

位 11：9	CSPT23	0x0	rw	选择 ADC_IN23 通道的采样时间 （Selection sample time of channel ADC_IN23） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 8：6	CSPT22	0x0	rw	选择 ADC_IN22 通道的采样时间 （Selection sample time of channel ADC_IN22） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 5：3	CSPT21	0x0	rw	选择 ADC_IN21 通道的采样时间 （Selection sample time of channel ADC_IN21） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。
位 2：0	CSPT20	0x0	rw	选择 ADC_IN20 通道的采样时间 （Selection sample time of channel ADC_IN20） 000：2.5 周期； 001：6.5 周期； 010：12.5 周期； 011：24.5 周期； 100：47.5 周期； 101：92.5 周期； 110：247.5 周期； 111：640.5 周期。

18.5.16 ADC普通序列寄存器4（ADC_OSQ4）

访问：字访问

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	请保持默认值。
位 29：25	OSN22	0x00	rw	普通序列中第 22 个转换通道的编号（number of 6th conversion in ordinary sequence）
位 24：20	OSN21	0x00	rw	普通序列中第 21 个转换通道的编号（number of 5th conversion in ordinary sequence）
位 19：15	OSN20	0x00	rw	普通序列中第 20 个转换通道的编号（number of 4th conversion in ordinary sequence）

位 14：10	OSN19	0x00	rw	普通序列中第 19 个转换通道的编号（number of 3rd conversion in ordinary sequence）
位 9：5	OSN18	0x00	rw	普通序列中第 18 个转换通道的编号 （number of 2nd conversion in ordinary sequence）
位 4：0	OSN17	0x00	rw	普通序列中第 17 个转换通道的编号 （number of 1st conversion in ordinary sequence） 注：编号可设定 0~17、20~27，示例：设定为 17 就代表第 1 个转换的是 ADC_IN17 通道。

18.5.17 ADC普通序列寄存器5（ADC_OSQ5）

访问：字访问

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	请保持默认值。
位 29：25	OSN28	0x00	rw	普通序列中第 28 个转换通道的编号（number of 6th conversion in ordinary sequence）
位 24：20	OSN27	0x00	rw	普通序列中第 27 个转换通道的编号（number of 5th conversion in ordinary sequence）
位 19：15	OSN26	0x00	rw	普通序列中第 26 个转换通道的编号（number of 4th conversion in ordinary sequence）
位 14：10	OSN25	0x00	rw	普通序列中第 25 个转换通道的编号（number of 3rd conversion in ordinary sequence）
位 9：5	OSN24	0x00	rw	普通序列中第 24 个转换通道的编号 （number of 2nd conversion in ordinary sequence）
位 4：0	OSN23	0x00	rw	普通序列中第 23 个转换通道的编号 （number of 1st conversion in ordinary sequence） 注：编号可设定 0~17、20~27，示例：设定为 17 就代表第 1 个转换的是 ADC_IN17 通道。

18.5.18 ADC普通序列寄存器6（ADC_OSQ6）

访问：字访问

域	简称	复位值	类型	功能
位 31：20	保留	0x0	resd	请保持默认值。
位 19：15	OSN32	0x00	rw	普通序列中第 32 个转换通道的编号（number of 4th conversion in ordinary sequence）
位 14：10	OSN31	0x00	rw	普通序列中第 31 个转换通道的编号（number of 3rd conversion in ordinary sequence）
位 9：5	OSN30	0x00	rw	普通序列中第 30 个转换通道的编号 （number of 2nd conversion in ordinary sequence）
位 4：0	OSN29	0x00	rw	普通序列中第 29 个转换通道的编号 （number of 1st conversion in ordinary sequence） 注：编号可设定 0~17、20~27，示例：设定为 17 就代表第 1 个转换的是 ADC_IN17 通道。

18.5.19 ADC过采样寄存器（ADC_OVSP）

访问：字访问

域	简称	复位值	类型	功能
位 31：11	保留	0x0000	resd	请保持默认值。
位 10	OOSRSEL	0x0	rw	普通过采样重转模式选择 (Ordinary oversampling restart mode select) 当普通过采样中途被抢占通道转换插入后，依据如下设定，选择普通过采样从哪里开始恢复转换。 0：接续模式（普通过采样缓冲区会被保留）； 1：重转模式（普通过采样缓冲区会被清零，即当前通道之前的采样次数被清零）。
位 9	OOSTREN	0x0	rw	普通过采样触发模式使能 (Ordinary oversampling trigger mode enable) 0：关闭（通道的所有过采样转换仅需一次触发）； 1：开启（通道的每个过采样转换均需进行触发）。
位 8：5	OSSSEL	0x0	rw	过采样移位选择 (Oversampling shift select) 此位定义应用到最终过采样结果的右移位数。 0000：不进行移位； 0001：移 1 位； 0010：移 2 位； 0011：移 3 位； 0100：移 4 位； 0101：移 5 位； 0110：移 6 位； 0111：移 7 位； 1000：移 8 位； 1001~1111：未用，禁止配置。
位 4：2	OSRSEL	0x0	rw	过采样率选择 (Oversampling ratio select) 000：2x； 001：4x； 010：8x； 011：16x； 100：32x； 101：64x； 110：128x； 111：256x。
位 1	POSEN	0x0	rw	抢占过采样使能 (Preempted oversampling enable) 0：关闭； 1：开启。
位 0	OOSEN	0x0	rw	普通过采样使能 (Ordinary oversampling enable) 0：关闭； 1：开启。

18.5.20 ADC校准值寄存器（ADC_CALVAL）

访问：字访问

域	简称	复位值	类型	功能
位 31：7	保留	0x0000	resd	请保持默认值。
位 6：0	CALVAL	0x0	rw	A/D 校准值（A/D Calibration value）

18.5.21 ADC额外寄存器(ADC_MISC)

访问：字访问

域	简称	复位值	类型	功能
位 31：6	保留	0x0000	resd	请保持默认值。
位 5：0	XTEST	0xC	rw	ADC 测试值（A/D test value） 用于微调 ADC 转换结果，在 ADC 时钟开启但未使能 ADC 及校准之前，需配置 XTEST 值为 0x9，方可继续使用 ADC

18.5.22 ADC通用控制寄存器（ADC_CCTRL）

访问：字访问

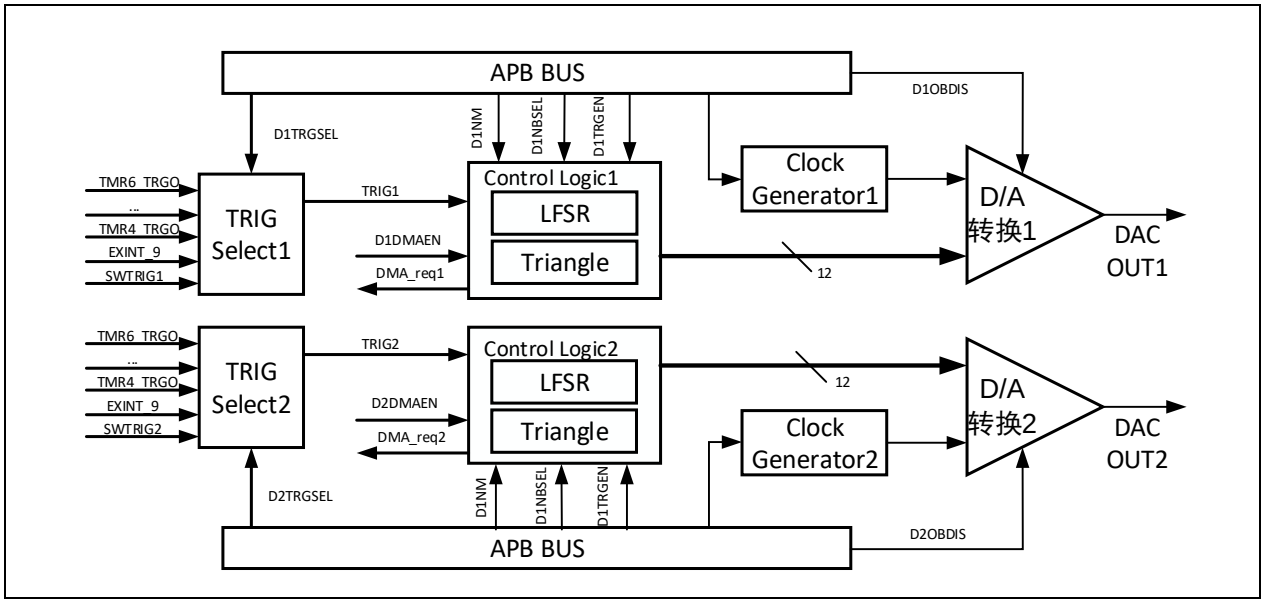
域	简称	复位值	类型	功能
位 31：24	保留	0x0000	resd	请保持默认值。
位 23	ITSRVEN	0x0	rw	内部温度传感器及 V_{INTRV} 使能 （Internal temperature sensor and V_{INTRV} enable） 0：关闭； 1：开启。
位 22：20	保留	0x0	resd	请保持默认值。
位 19：16	ADCDIV	0x0	rw	ADC 分频因子 (ADC division) 0000：ADC 输入时钟 2 分频 0001：ADC 输入时钟 3 分频 … 1111：ADC 输入时钟 17 分频 注： 经此分频后的时钟将被所有 ADC 共用； ADCCLK 额定最大为 80MHz，经分频后 ADCCLK 不能高于 PCLK2。
位 15：0	保留	0x0	resd	请保持默认值。

19 数字/模拟转换（DAC）

19.1 简介

数模转换器（DAC）采用 12 位数字输入，产生 0 至参考电压之间的模拟输出。数字部分可以配置为 8 位或者 12 位模式，支持单/双 DAC 的左对齐或者右对齐，同时可以与 DMA 配合使用。两个 DAC1/DAC2 各有一个数模转换器，每个 DAC1/DAC2 可以独立进行数模转换，也可以双 DAC 同时触发进行转换，输入参考电压 V_{REF+}可以使转换操作更加精确。

图 19-1 DAC1/DAC2模块框图



19.2 主要特性

- 单/双 DAC 8 位或者 12 位数字输入
- 数据支持左对齐或者右对齐模式
- 支持噪声波/三角波产生
- 双 DAC 或者单个 DAC1/DAC2 独立转换
- 每个 DAC1/DAC2 支持 DMA 模式
- 软件触发或者外部触发转换
- 支持输入参考电压 V_{REF+}

19.3 设计提示

DAC 有以下提示仅供设计参考。

- 模拟模块配置

DAC1/DAC2 的模拟部分由 DAC 控制寄存器（DAC_CTRL）ENx 位控制开启，数字部分则不受该位控制。另外 DAC 集成了 2 个输出增益，可以用来减少输出阻抗，无需外部运放即可直接驱动外部负载。每个 DAC1/DAC2 输出增益可以通过设置 DAC 控制寄存器（DAC_CTRL）的 DxOBDIS 位来使能或关闭。

- DMA 功能

任一 DAC1/DAC2 支持 DMA 功能，通过设置 DAC 控制寄存器（DAC_CTRL）的 DxDMAEN 位使能 DMA 请求。当触发使能位 DxTRGEN 有效，触发信号有效时，即产生 DMA 请求。DAC 的 DMA 请求不会累计，未来得及处理的 DMA 请求将被忽略，也不会产生错误信息。

- 在双 DAC 模式下，程序可以只使用一个 DMA 请求，一个 DMA 通道的情况下，处理工作在双 DAC 模式的 2 个 DAC1/DAC2。

- DMA 下溢处理

当开启 DAC 的 DMA 功能，如果第二个外部触发到达时尚未收到 DMA 反馈的第一个外部

触发的确认信号，即产生 DMA 下溢。此时将不会处理新的外部触发，并且不会发出新的 DMA 请求，同时 DAC 状态寄存器（DAC_SR）DxDMAUDRF 位指示发生 DMA 下溢错误，配置 DAC 控制寄存器（DAC_CTRL）DxDMAUDRIEN 位为‘1’，将产生相应的 DMA 下溢错误中断。

- 软件通过写入‘1’来清除 DAC 状态寄存器（DAC_SR）DxDMAUDRF 位，在重新开始 DMA 传输之前，应该先清除 DAC 控制寄存器（DAC_CTRL）的 DxDMAEN 位，然后重新初始化 DMA 和 DAC。
- 输入输出配置

数字输入经过 DAC 线性地转换为模拟电压输出，其范围为 0 至 V_{REF+}。模拟 DAC 模块采用 VDDA 供电，输入正模拟参考电压大小介于 2.0V 与 VDDA 之间，PA4 或者 PA5 作为模拟输出时，为避免寄生干扰和额外的功耗，需设置为模拟输入。

DAC 输出 = V_{REF+} x（DxODT[11：0]/ 4095）

19.4功能描述

19.4.1 触发事件

如果 DAC_CTRL 寄存器的 DxTRGEN 位被置 1，DAC 转换可以由某外部事件（定时器计数器、外部中断线）或者软件触发，触发事件源由 DxTRGSEL[2：0]进行选择。

表 19-1 触发源选择

触发源	DxTRGSEL [2：0]	说明
TMR6_TRGOUT	000	片上信号
TMR3_TRGOUT	001	
TMR7_TRGOUT	010	
TMR9_TRGOUT	011	
TMR2_TRGOUT	100	
TMR4_TRGOUT	101	
EXINT_9	110	外部信号
DxSWTRG	111	软件触发

当 DxTRGEN 位被置 1 时，每次 DAC 侦测到有效的的触发事件，存放在 HDRx 中的数据就会被传动到寄存器 DAC_DxODT 中，当选择软件触发时，触发标志 DxSWTRG 在被软件置‘1’后，硬件自动清零。一旦数据装入 DAC_DxODT 寄存器，经过一段时间，模拟的数模转换器输出即有效。

当 DxTRGEN 位被清‘0’时，每次写入数据寄存器值时，数据即被传送到寄存器 DAC_DxODT 中，无需等待触发事件。

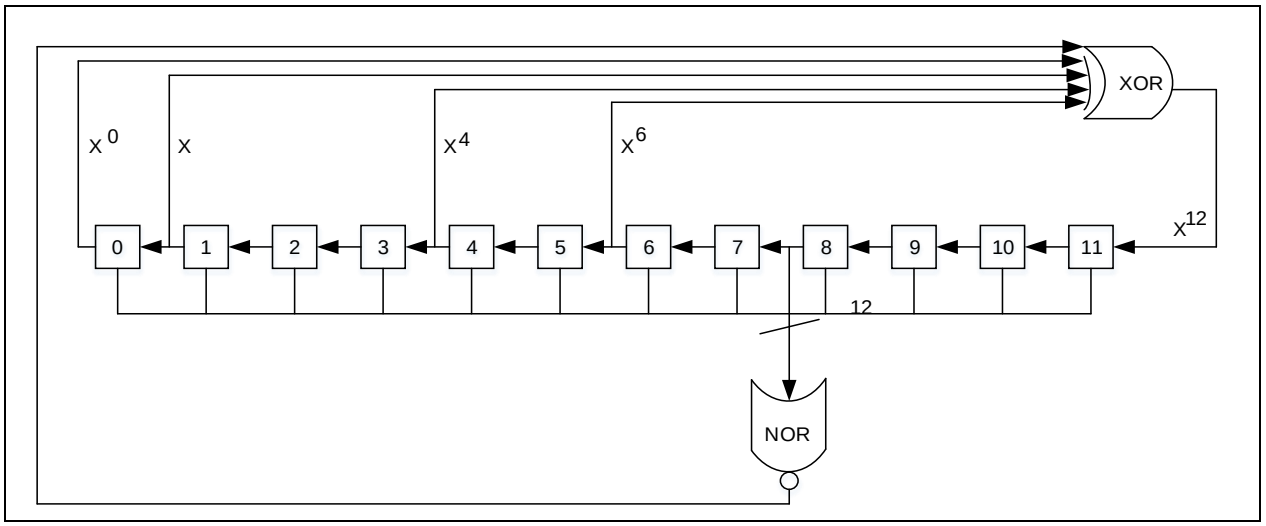
19.4.2 噪声/三角波生成

有噪声波和三角波两种波形可以叠加到 DAC 输出：分别利用线性回馈移位寄存器（LENinear Feedback Shift Register LFSR）产生幅度变化的伪噪声和通过三角波发生器（triangle）产生三角波。当设置 DxNM[1：0]位为‘01’使能 LFSR，输出幅度变化的伪噪声。当设置 DxNM[1：0]位为‘1x’使能三角波发生器，输出三角波。

LFSR 原理

寄存器 LFSR 的预装入值为 0xAAA，按照特定算法，在每次触发事件之后更新该寄存器的值。

图 19-2 DAC LFSR寄存器算法



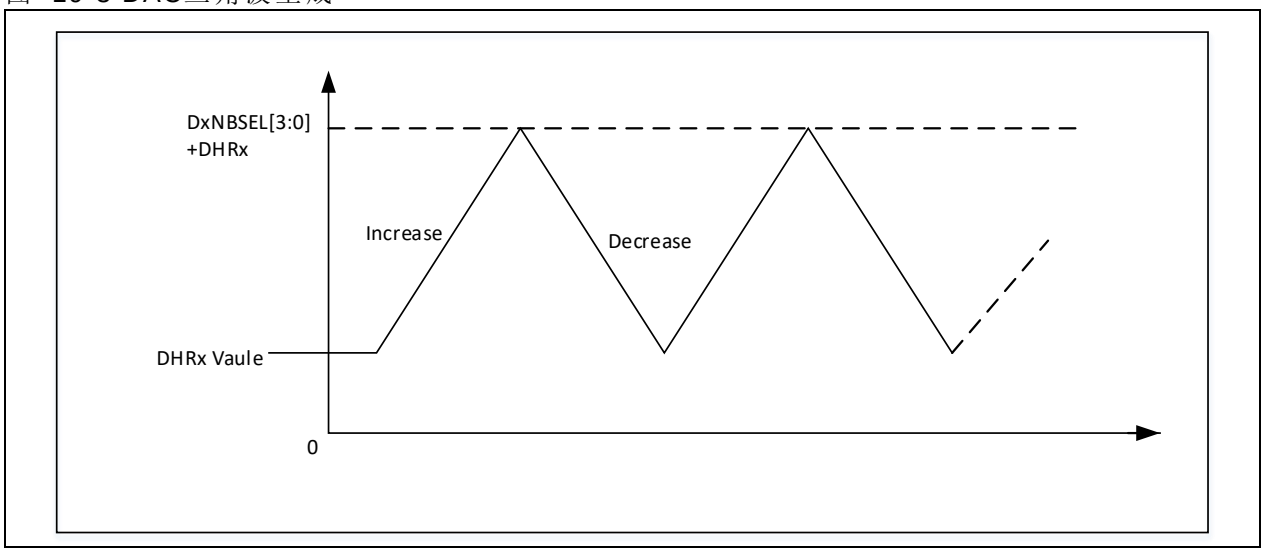
设置 DAC 控制寄存器（DAC_CTRL）的 DxNBSEL [3：0]位可以屏蔽部分或者全部 LFSR 的数据，这样的得到的 LSFR 值与 DHRx 的数值相加，去掉溢出位之后即被写入 DAC1/DAC2 数据输出寄存器（DAC_DxODT）。将 DxNM[1：0]位置’00’可以关掉 LFSR 功能，同时复位 LFSR 波形的生成算法。

三角波原理

当设置 DxNM[1：0]位为‘1x’，则选择 DAC 的三角波生成功能。三角波的幅度由 DAC 控制寄存器（DAC_CTRL）的 DxNBSEL [3：0]位设置，内部的三角波计数器每检测到一次触发事件累加 1，当达到 DxNBSEL [3：0]位定义的最大幅度时，则计数器开始递减，达到 0 后再开始累加，周而复始。

同时，计数器的值与 DHRx 寄存器的数值相加并丢弃溢出位后写入 DAC1/DAC2 数据输出寄存器（DAC_DxODT）。将 DxNM[1：0]设置‘00’，可以关掉三角波生成，同时复位三角波计数器。

图 19-3 DAC三角波生成



19.4.3 数据配置

DAC 支持单 DAC 或者双 DAC 模式，根据模式的不同，数据配置有以下方式：

单 DAC 数据配置方式：采用 8 位数据右对齐，或者 12 位数据左对齐，或者 12 位数据右对齐方式时，对寄存器 DAC_DxDTH8R [7：0]，或者 DAC_DxDTH12L [15：4]，或者 DAC_DxDTH12R [11：0]位写入值。

双 DAC 数据配置方式：采用 8 位数据右对齐，或者 12 位数据左对齐，或者 12 位数据右对齐方式时，对寄存器 DAC_DDTH8R [7：0]和 DAC_DDTH8R [15：8]，或者 DAC_DDTH12L [15：4]和 DAC_DDTH12L [31：20]，或者 DAC_DDTH12R [11：0]和 DAC_DDTH12R [27：16]位写入值。

写入的 8 位数据对应 DHRx[11：4]位，写入的 12 位数据对应 DHRx[11：0]位。

19.5DAC寄存器

下表列出了所有 DAC 寄存器。
必须以字（32 位）的方式操作这些外设寄存器。
表 19-2 DAC寄存器映像和复位值

寄存器简称	基址偏移量	复位值
DAC_CTRL	000h	0x0000 0000
DAC_SWTRG	004h	0x0000 0000
DAC_D1DTH12R	008h	0x0000 0000
DAC_D1DTH12L	00Ch	0x0000 0000
DAC_D1DTH8R	010h	0x0000 0000
DAC_D2DTH12R	014h	0x0000 0000
DAC_D2DTH12L	018h	0x0000 0000
DAC_D2DTH8R	01Ch	0x0000 0000
DAC_DDTH12R	020h	0x0000 0000
DAC_DDTH12L	024h	0x0000 0000
DAC_DDTH8R	028h	0x0000 0000
DAC_D1ODT	02Ch	0x0000 0000
DAC_D2ODT	030h	0x0000 0000
DAC_STS	034h	0x0000 0000

19.5.1 DAC控制寄存器（DAC_CTRL）

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	请保持默认值。
位 29	D2DMAUDRIEN	0x0	rw	DAC2 的 DMA 传输下溢中断使能 （DAC2 DMA transfer underrun interrupt enable） 该位由软件设置和清除。 0：关闭 DAC2 DMA 下溢中断； 1：使能 DAC2 DMA 下溢中断。
位 28	D2DMAEN	0x0	rw	DAC2 的 DMA 传输使能 （DAC2 DMA transfer enable） 该位由软件设置和清除。 0：关闭 DAC2 DMA 模式； 1：使能 DAC2 DMA 模式。
位 27：24	D2NBSEL	0x0	rw	DAC2 噪声位宽选择 （DAC2 noise bit select） 这些位用于在噪声生成模式下选择屏蔽位，在三角波生成模式下选择波形的幅值。 0000：不屏蔽 LSFR 位 0 / 三角波幅值等于 1； 0001：不屏蔽 LSFR 位[1：0] / 三角波幅值等于 3； 0010：不屏蔽 LSFR 位[2：0] / 三角波幅值等于 7； 0011：不屏蔽 LSFR 位[3：0] / 三角波幅值等于 15； 0100：不屏蔽 LSFR 位[4：0] / 三角波幅值等于 31； 0101：不屏蔽 LSFR 位[5：0] / 三角波幅值等于 63； 0110：不屏蔽 LSFR 位[6：0] / 三角波幅值等于 127； 0111：不屏蔽 LSFR 位[7：0] / 三角波幅值等于 255； 1000：不屏蔽 LSFR 位[8：0] / 三角波幅值等于 511； 1001：不屏蔽 LSFR 位[9：0] / 三角波幅值等于 1023； 1010：不屏蔽 LSFR 位[10：0] / 三角波幅值等于 2047； ≥1011：不屏蔽 LSFR 位[11：0] / 三角波幅值等于 4095。
位 23：22	D2NM	0x0	rw	DAC2 噪声/三角波生成选择 （DAC2 noise mode）

				00：关闭； 01：开启噪声波形发生器； 1x：开启三角波发生器。
位 21：19	D2TRGSEL	0x0	rw	DAC2 的触发事件选择 （DAC2 trigger select） 000：TMR6 TRGOUT 事件； 001：TMR3 TRGOUT 事件； 010：TMR7 TRGOUT 事件； 011：TMR9 TRGOUT 事件； 100：TMR2 TRGOUT 事件； 101：TMR4 TRGOUT 事件； 110：外部中断线 9； 111：软件触发。 注意：这些位只能在 D2TRGEN = 1 时设置。
位 18	D2TRGEN	0x0	rw	DAC2 触发使能 （DAC2 trigger enable） 0：关闭； 1：开启。 注： 关闭触发时，写入寄存器 DAC_D2DTHx 的数据在 1 个 APB1 时钟周期后传入寄存器 DAC_D2ODT。 开启触发时，写入寄存器 DAC_D2DTHx 的数据在 3 个 APB1 时钟周期后传入寄存器 DAC_ D2ODT。 如果选择软件触发，写入寄存器 DAC_D2DTHx 的资料只需要 1 个 APB1 时钟周期就可以传入寄存器 DAC_ D2ODT。
位 17	D2OBDIS	0x0	rw	关闭 DAC2 输出缓存 （DAC2 output buffer disable） 0：开启输出缓存； 1：关闭输出缓存。
位 16	D2EN	0x0	rw	DAC2 使能 （DAC2 enable） 0：关闭； 1：开启。
位 15：14	保留	0x0	resd	请保持默认值。
位 13	D1DMAUDRIEN	0x0	rw	DAC1 的 DMA 传输下溢中断使能 （DAC1 DMA transfer underrun interrupt enable） 该位由软件设置和清除。 0：关闭 DAC1 DMA 下溢中断； 1：使能 DAC1 DMA 下溢中断。
位 12	D1DMAEN	0x0	rw	DAC1 的 DMA 传输使能 （DAC1 DMA transfer enable） 该位由软件设置和清除。 0：关闭 DAC1 DMA 模式； 1：使能 DAC1 DMA 模式。
位 11：8	D1NBSEL	0x0	rw	DAC1 屏蔽/幅值选择（DAC1 noise bit select） 这些位用于在噪声生成模式下选择屏蔽位，在三角波生成模式下选择波形的幅值。 0000：不屏蔽 LSFR 位 0 / 三角波幅值等于 1； 0001：不屏蔽 LSFR 位[1：0] / 三角波幅值等于 3； 0010：不屏蔽 LSFR 位[2：0] / 三角波幅值等于 7； 0011：不屏蔽 LSFR 位[3：0] / 三角波幅值等于 15； 0100：不屏蔽 LSFR 位[4：0] / 三角波幅值等于 31； 0101：不屏蔽 LSFR 位[5：0] / 三角波幅值等于 63； 0110：不屏蔽 LSFR 位[6：0] / 三角波幅值等于 127； 0111：不屏蔽 LSFR 位[7：0] / 三角波幅值等于 255； 1000：不屏蔽 LSFR 位[8：0] / 三角波幅值等于 511； 1001：不屏蔽 LSFR 位[9：0] / 三角波幅值等于 1023； 1010：不屏蔽 LSFR 位[10：0] / 三角波幅值等于 2047； ≥1011：不屏蔽 LSFR 位[11：0] / 三角波幅值等于 4095。
位 7：6	D1NM	0x0	rw	DAC1 噪声/三角波生成选择 （DAC1 noise mode） 00：关闭； 01：开启噪声波形发生器；

				1x：开启三角波发生器。
位 5：3	D1TRGSEL	0x0	rw	DAC1 的触发事件选择 （DAC1 trigger select） 000：TMR6 TRGOUT 事件； 001：TMR3 TRGOUT 事件； 010：TMR7 TRGOUT 事件； 011：TMR9 TRGOUT 事件； 100：TMR2 TRGOUT 事件； 101：TMR4 TRGOUT 事件； 110：外部中断线 9； 111：软件触发。 注：这些位只能在 D1TRGEN= 1 时设置。
位 2	D1TRGEN	0x0	rw	DAC1 触发使能 （DAC1 trigger enable） 0：关闭； 1：开启。 注： 关闭触发时，写入寄存器 DAC_D1DTHx 的数据在 1 个 APB1 时钟周期后传入寄存器 DAC_D1ODT 开启触发时，写入寄存器 DAC_D1DTHx 的数据在 3 个 APB1 时钟周期后传入寄存器 DAC_ D1ODT。 如果选择软件触发，写入寄存器 DAC_D1DTHx 的资料只需要 1 个 APB1 时钟周期就可以传入寄存器 DAC_ D1ODT。
位 1	D1OBDIS	0x0	rw	关闭 DAC1 输出缓存 （DAC1 output buffer disable） 0：开启输出缓存； 1：关闭输出缓存。
位 0	D1EN	0x0	rw	DAC1 使能 （DAC1 enable） 0：关闭； 1：开启。

19.5.2 DAC软件触发寄存器（DAC_SWTRG）

域	简称	复位值	类型	功能
位 31：2	保留	0x0000 0000	resd	请保持默认值。
位 1	D2SWTRG	0x0	rw	DAC2 软件触发 （DAC2 software trigger） 0：不触发； 1：触发。 注：一旦寄存器 DAC_D2DTH 的数据传入寄存器 DAC_D2ODT，（1 个 APB1 时钟周期后）该位由硬件清零。
位 0	D1SWTRG	0x0	rw	DAC1 软件触发 （DAC1 software trigger） 0：不触发； 1：触发。 注：一旦寄存器 DAC_D1DTH 的数据传入寄存器 DAC_D1ODT，（1 个 APB1 时钟周期后）该位由硬件清零。

19.5.3 DAC1的12位右对齐数据保持寄存器（DAC_D1DTH12R）

域	简称	复位值	类型	功能
位 31：12	保留	0x00000	resd	请保持默认值。
位 11：0	D1DT12R	0x000	rw	DAC1 的 12 位右对齐数据 （DAC1 12-bit right-aligned data）

19.5.4 DAC1的12位左对齐数据保持寄存器（DAC_D1DTH12L）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	请保持默认值。

位 15：4	D1DT12L	0x000	rw	DAC1 的 12 位左对齐数据 （DAC1 12-bit left-aligned data）
位 3：0	保留	0x0	resd	请保持默认值。

19.5.5 DAC1的8位右对齐数据保持寄存器（DAC_D1DTH8R）

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	resd	请保持默认值。
位 7：0	D1DT8R	0x00	rw	DAC1 的 8 位右对齐数据 （DAC1 8-bit right-aligned data）

19.5.6 DAC2的12位右对齐数据保持寄存器（DAC_D2DTH12R）

域	简称	复位值	类型	功能
位 31：12	保留	0x00000	resd	请保持默认值。
位 11：0	D2DT12R	0x000	rw	DAC2 的 12 位右对齐数据 （DAC2 12-bit right-aligned data）

19.5.7 DAC2的12位左对齐数据保持寄存器（DAC_D2DTH12L）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	请保持默认值。
位 15：4	D2DT12L	0x000	rw	DAC2 的 12 位左对齐数据 （DAC2 12-bit left-aligned data）
位 3：0	保留	0x0	resd	请保持默认值。

19.5.8 DAC2的8位右对齐数据保持寄存器（DAC_D2DTH8R）

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	resd	请保持默认值。
位 7：0	D2DT8R	0x00	rw	DAC2 的 8 位右对齐数据 （DAC2 8-bit right-aligned data）

19.5.9 双DAC的12位右对齐数据保持寄存器（DAC_DDTH12R）

域	简称	复位值	类型	功能
位 31：28	保留	0x0	resd	请保持默认值。
位 27：16	DD2DT12R	0x000	rw	DAC2 的 12 位右对齐数据 （DAC2 12-bit right-aligned data）
位 15：12	保留	0x0	resd	请保持默认值。
位 11：0	DD1DT12R	0x000	rw	DAC1 的 12 位右对齐数据 （DAC1 12-bit right-aligned data）

19.5.10 双DAC的12位左对齐数据保持寄存器（DAC_DDTH12L）

域	简称	复位值	类型	功能
位 31：20	DD2DT12L	0x000	rw	DAC2 的 12 位左对齐数据 （DAC2 12-bit left-aligned data）
位 19：16	保留	0x0	resd	请保持默认值。
位 15：4	DD1DT12L	0x000	rw	DAC1 的 12 位左对齐数据 （DAC1 12-bit left-aligned data）
位 3：0	保留	0x0	resd	请保持默认值。

19.5.11双DAC的8位右对齐数据保持寄存器（DAC_DDTH8R）

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	请保持默认值。
位 15：8	DD2DT8R	0x00	rw	DAC2 的 8 位右对齐数据 （DAC2 8-bit right-aligned data）
位 7：0	DD1DT8R	0x00	rw	DAC1 的 8 位右对齐数据 （DAC1 8-bit right-aligned data）

19.5.12DAC1数据输出寄存器（DAC_D1ODT）

域	简称	复位值	类型	功能
位 31：12	保留	0x00000	resd	请保持默认值。
位 11：0	D1ODT	0x000	rw	DAC1 输出数据（DAC1 output data）

19.5.13DAC2数据输出寄存器（DAC_D2ODT）

域	简称	复位值	类型	功能
位 31：12	保留	0x00000	resd	请保持默认值。
位 11：0	D2ODT	0x000	rw	DAC2 输出数据（DAC2 output data）

19.5.14DAC状态寄存器（DAC_STS）

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	请保持默认值。
位 29	D2DMAUDRF	0x0	w1c	DAC2 的 DMA 传输下溢标志 （DAC2 DMA transfer underrun flag） 0：DAC2 未发生 DMA 传输下溢。 1：DAC2 发生 DMA 传输下溢。 注：软件写‘1’清除该标志。
位 28：14	保留	0x0000	resd	请保持默认值
位 13	D1DMAUDRF	0x0	w1c	DAC1 的 DMA 传输下溢标志 （DAC1 DMA transfer underrun flag） 0：DAC1 未发生 DMA 传输下溢。 1：DAC1 发生 DMA 传输下溢。 注：软件写‘1’清除该标志。
位 12：0	保留	0x0000	resd	请保持默认值

20 CAN 总线控制器（CAN）

20.1简介

CAN（Controller Area Network）是一种实现各节点之间实时、可靠数据通信的分布式串行通信协议，支持 CAN 协议 2.0A 和 2.0B.

20.2主要特性

- 波特率最高可达 1M bit/s/
- 支持时间触发通信
- 中断使能和屏蔽
- 自动重传功能可配

发送

- 3 个发送邮箱
- 发送优先级可配置
- 支持发送时间戳

接收

- 2 个深度为 3 的 FIFO
- 14 组过滤器组
- 支持标识符列表模式
- 支持标识符掩码模式
- 支持 FIFO 溢出管理

时间触发通信模式

- 16 位定时器
- 发送时间戳

20.3波特率设置

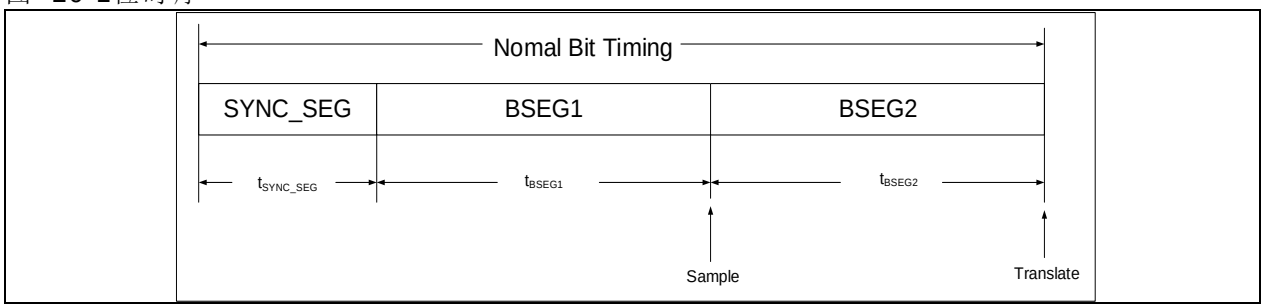
CAN 总线的额定位时间由 3 部分组成。

同步段(SYNC_SEG)，该段占用 1 时间单元，时间长度由 CAN 位时序寄存器（CAN_BTMG)的 BRDIV[11：0]位定义。

位段 1（BIT SEGMENT 1），包括 CAN 标准里的 PROP_SEG 和 PHASE_SEG1，记为 BSEG1，该段占用 1 至 16 时间单元，时间单元个数由 BTS1[3：0]位定义。

位段 2（BIT SEGMENT 2），包括 CAN 标准里的 PHASE_SEG2，记为 BSEG2，该段占用 1 至 8 时间单元，时间单元个数由 BTS2[2：0]位定义。

图 20-1位时序



波特率计算公式

$$BaudRate = \frac{1}{\text{Nomal Bit Timimg}}$$

$$Nomal\ \ Bit\ \ Timing = t_{SYNC\_SEG} + t_{BSEG1} + t_{BSEG2}$$

其中

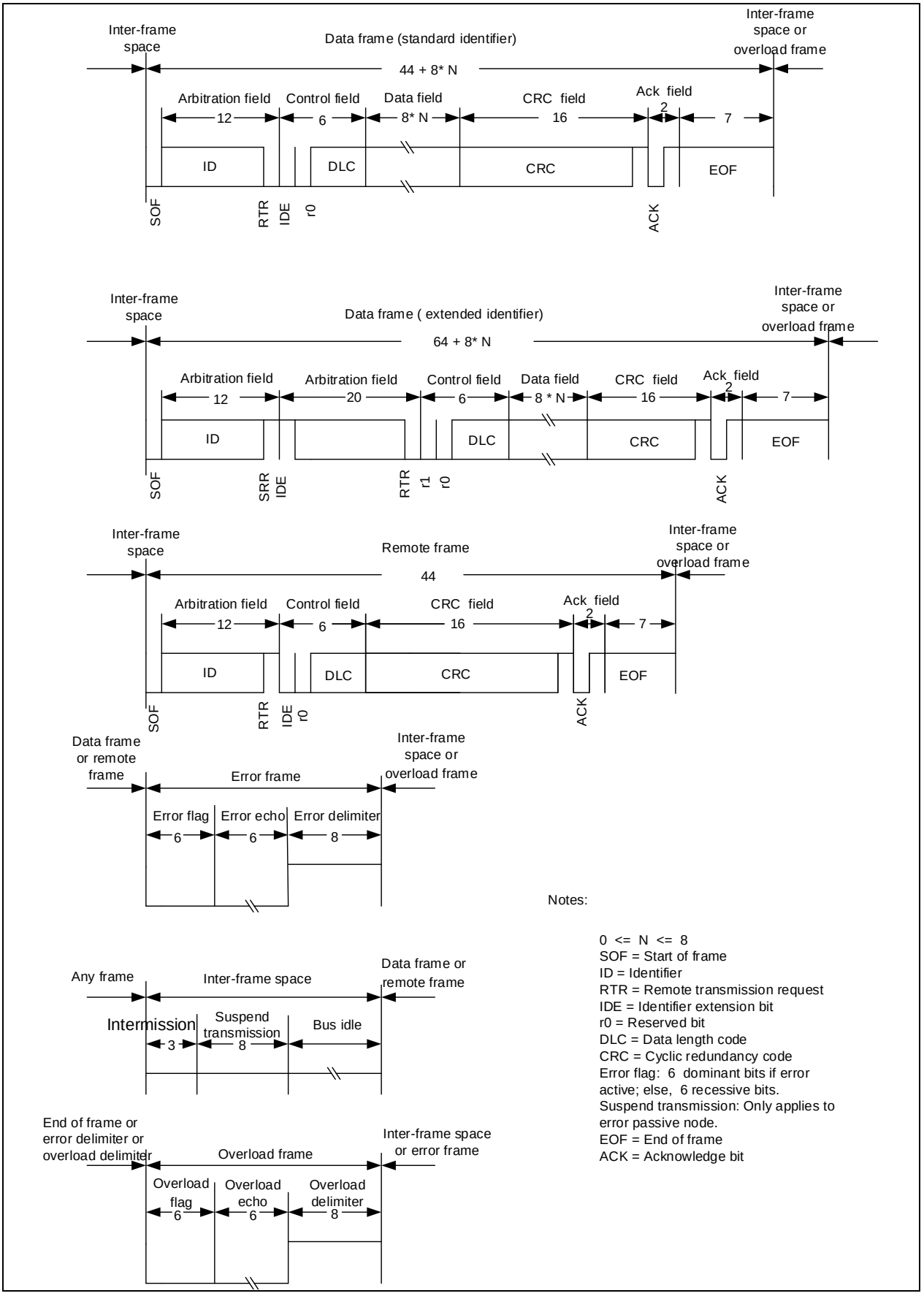
$$t_{SYNC\_SEG} = 1\ \ x\ \ t_q$$

$$t_{BSEG1} = (1 + \text{BTS1}[3:0]) \times t_q$$
$$t_{BSEG2} = (1 + \text{BTS2}[2:0]) \times t_q$$
$$t_q = (1 + \text{BRDIV}[11:0]) \times t_{pclk}$$

硬同步和重同步

默认情况下，CAN 节点的每一位的起始位置总是在同步段内，同时在位段 1 和位段 2 临界位置进行采样。但是由于节点振荡器漂移，网络节点之间的传播延迟以及噪声干扰等，实际的传输过程中，CAN 节点的每一位会存在一定的相位误差。为避免相位误差对通讯造成影响，可以通过帧起始位置的边沿以及后面的下降沿进行硬同步或者重同步，同步补偿的时间长度最长不超过重新同步调整宽度（1 至 4 个时间单元，RSAW[1：0]位设置）。

图 20-2帧类型



Notes:

0 <= N <= 8
SOF = Start of frame
ID = Identifier
RTR = Remote transmission request
IDE = Identifier extension bit
r0 = Reserved bit
DLC = Data length code
CRC = Cyclic redundancy code
Error flag: 6 dominant bits if error active; else, 6 recessive bits.
Suspend transmission: Only applies to error passive node.
EOF = End of frame
ACK = Acknowledge bit

20.4 中断管理

CAN 控制器具有 4 个中断向量，通过配置 CAN 中断使能寄存器 （CAN_INTEN），可以控制相应的中断开启或关闭。

图 20-3发送中断的产生

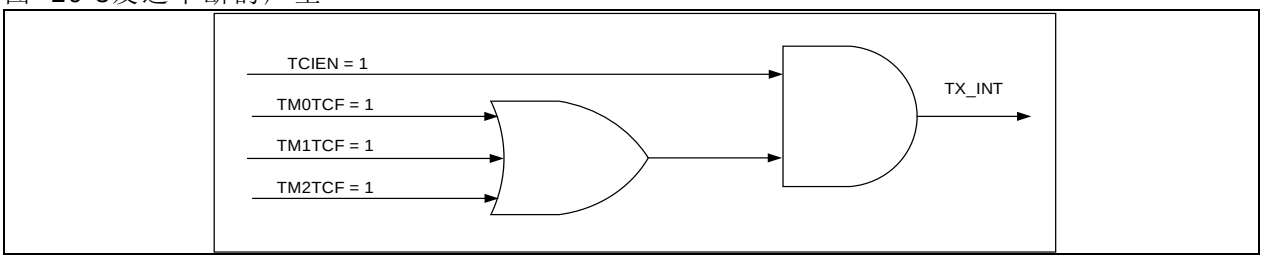


图 20-4接收中断0的产生

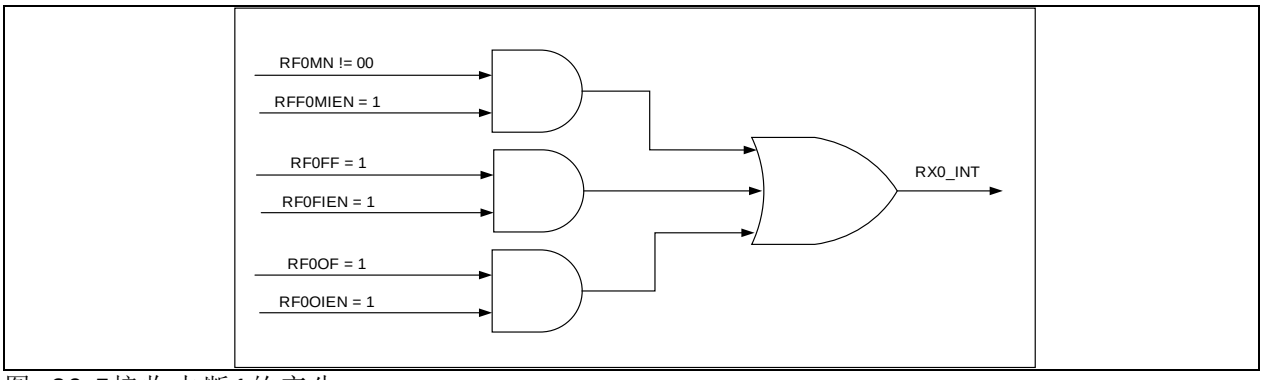


图 20-5接收中断1的产生

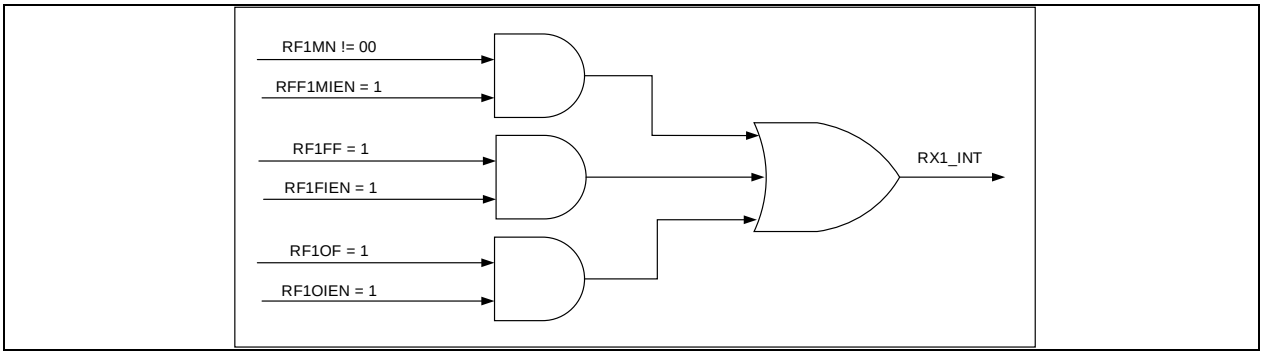
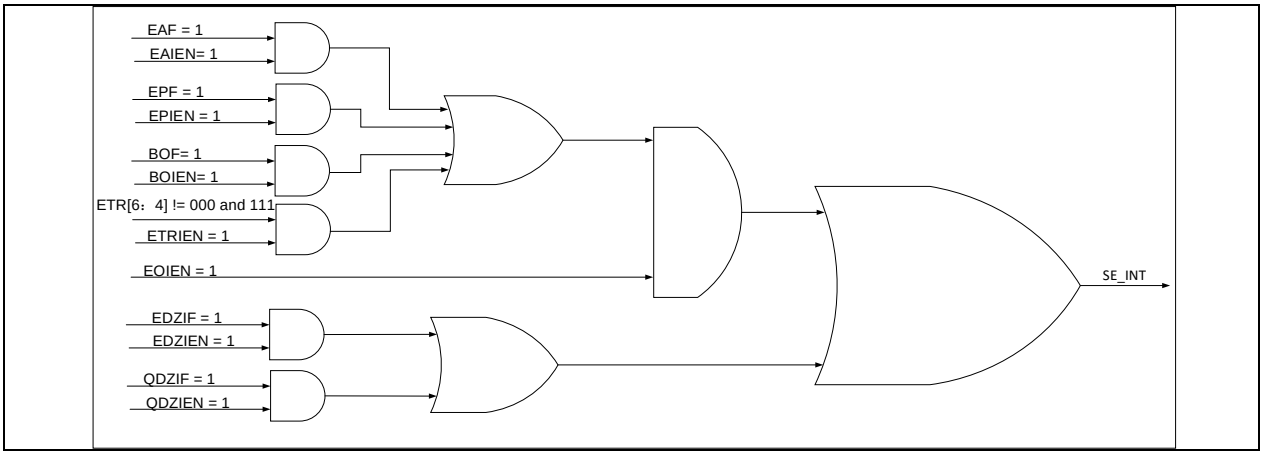


图 20-6状态错误中断的产生



20.5 设计提示

为便于 CAN 应用开发，设计时建议参考如下提示。

- 调试控制

当系统进入调试模式时，可以通过控制 MCU 调试寄存器 DEBUG_CTRL 的 CANx_PAUSE

以及 CAN 主控制寄存器（CAN_MCTRL）的 PTD 位控制 CAN 控制器处于停止状态或者正常发送接收状态。

- 时间触发通信

时间触发通信用于提高系统的实时性，避免总线竞争。当 CAN 主控制寄存器（CAN_MCTRL）的 TTCEN 位置‘1’，CAN 控制器的时间触发通信即被激活。内部 16 位定时器在每个 CAN 位累加，在帧起始位置被采样，生成时间戳，存储在接收 FIFO 邮箱数据长度和时间戳寄存器 （CAN_RFCx）/发送邮箱数据长度和时间戳寄存器（CAN_TMCx）中。

- 寄存器访问保护

CAN 位时序寄存器（CAN_BTMG）只能在冻结工作模式下进行修改。

CAN 节点发送错误数据对网络层不会带来问题，但却会对应用程序造成严重影响，因此只能在发送邮箱为空时改变它。

只有在设置过滤器为配置模式下（即 FCS=1），才能修改过滤器的设置，即修改 CAN 过滤器模式配置寄存器 （CAN_FMCFG），CAN 过滤器位宽配置寄存器（CAN_FBWCFG），CAN 过滤器 FIFO 关联寄存器 （CAN_FRF）。过滤位寄存器 x（CAN_FiFBx）只有在过滤器配置模式下（即 FCS=1）或者相应过滤器关闭情况下（即 FAENx=0）才能进行修改。

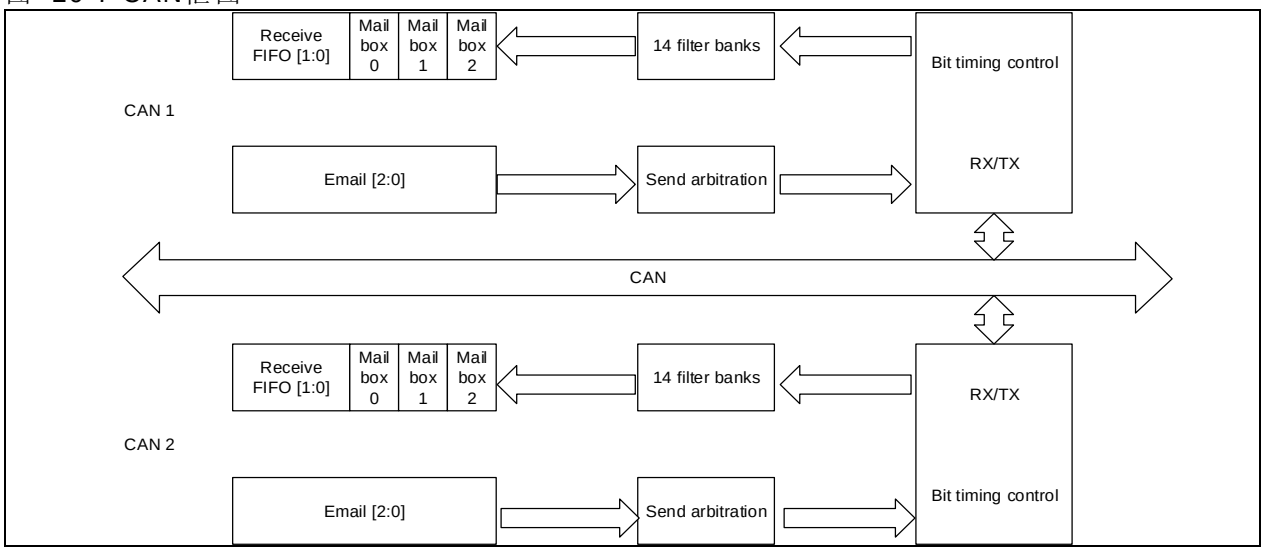
20.6功能描述

20.6.1 整体功能描述

随着 CAN 网络节点和报文数量的增加，需要一个增强的过滤机制处理各种类型的报文，减少接收报文的处理时间，采用 FIFO 的方案，使得 CPU 可以长时间处理应用层任务而不会丢失报文。同时发送报文由硬件配置发送优先级顺序，并且完全支持标准标识符（11 位）和扩展标识符（29 位）。

基于以上考虑，CAN 控制器提供 14 组位宽可变/可配置的标识符过滤器组，2 个接收 FIFO，每个 FIFO 都可以存放 3 个完整的报文，且完全由硬件管理，共有 3 个发送邮箱，发送调度器决定发送优先级顺序。

图 20-7 CAN框图



20.6.2 工作模式

CAN 控制器有 3 种工作模式：

- 睡眠模式

系统复位之后，CAN 控制器处于睡眠模式，该模式下 CAN 的时钟停止，因此可以节省电能，但软件仍然可以访问邮箱寄存器，同时 TX 管脚的内部上拉电阻被禁用。

软件通过对 CAN 主控制寄存器（CAN_MCTRL）的 DZEN 位置’1’，可以请求 bxCAN 进入睡眠模式，并且硬件对 CAN 主状态寄存器 （CAN_MSTS）的 DZC 位置‘1’进行确认。

有两种方式退出睡眠模式：配置 CAN 主控制寄存器（CAN_MCTRL）的 AEDEN 位为‘1’，一旦检测到 CAN 总线的活动，硬件自动对 DZEN 位清‘0’来唤醒 CAN 控制器。或者软件对 DZEN 位清‘0’可以退出睡眠状态。

睡眠工作模式进入冻结工作模式：对 CAN 主控制寄存器（CAN_MCTRL）的 FZEN 位置‘1’，并且同时对 DZEN 位清‘0’，然后硬件对 CAN 主状态寄存器 （CAN_MSTS）的 FZC 位置’1’来进行确认。
睡眠工作模式进入通讯工作模式：对 CAN 主控制寄存器（CAN_MCTRL）FZEN 和 DZEN 位清‘0’，并且 CAN 控制器必须跟总线取得同步，即在 CANRX 管脚上监测到 11 个连续的隐性位。

- 冻结模式

软件对 CAN 控制器的初始化，只能在冻结模式下进行，包括位 CAN 位时序寄存器 （CAN_BTMG）和 CAN 主控制寄存器（CAN_MCTRL）这 2 个寄存器。对 CAN 控制器的 14 组过滤器组（包括模式、位宽、FIFO 关联、激活和过滤器值）进行初始化可以在非冻结模式下进行。当 CAN 处于冻结模式时，禁止报文的接收和发送。
冻结工作模式进入通讯工作模式：对 CAN 主控制寄存器（CAN_MCTRL）FZEN 位清‘0’，硬件对 CAN 主状态寄存器 （CAN_MSTS）的 FZC 位清’0’就确认了冻结模式的退出，并且 CAN 控制器必须跟总线取得同步。
冻结工作模式进入睡眠工作模式：对 CAN 主控制寄存器（CAN_MCTRL）DZEN 位置‘1’ ，CAN 主控制寄存器（CAN_MCTRL）FZEN 位清‘0’，并且硬件对 CAN 主状态寄存器 （CAN_MSTS）的 DZC 位置‘1’进行确认。

- 通讯模式

在冻结工作模式配置完成 CAN 位时序寄存器 （CAN_BTMG）和 CAN 主控制寄存器（CAN_MCTRL）这两个寄存器后，控制 CAN 进入通讯工作模式，开始报文收发过程。
通讯工作模式进入睡眠工作模式：对 CAN 主控制寄存器（CAN_MCTRL）DZEN 位置‘1’，并等待当前 CAN 总线传输完成。
通讯工作模式进入冻结工作模式：对 CAN 主控制寄存器（CAN_MCTRL）FZEN 位置‘1’，并等待当前 CAN 总线传输完成。

20.6.3 测试方法

CAN 控制器定义了三种方法用于测试分析，包括只听方式、回环方式以及回环只听方式，可以通过 CAN 位时序寄存器 （CAN_BTMG）的 LOEN 位和 LBEN 位进行配置。

- 当 CAN_BTMG[31]位为‘1’时采用只听方式，此时 CAN 可以正常接收数据，但发送端 CANTX 固定隐性位输出。同时，发送端 CANTX 发出的显性位可以被接收端侦测到，但是不会影响到 CAN 总线。
- 当 CAN_BTMG[30]位为‘1’时采用回环方式，此时 CAN 只会接收本节点发送端 CANTX 的电平信号，同时 CAN 可以发送数据至外部总线，回环方式主要用于本节点的自我检测。
- 当 CAN_BTMG[31：30]位为‘11’时，只听方式和回环方式同时有效，此时 CAN 与总线网络断开，发送端 CANTX 固定隐性位输出，并且发送端直接与接收端相连。

20.6.4 报文过滤

在接收到的报文会根据其标识符（ID）进行过滤，通过过滤的报文会存储在对应的 FIFO 中，没有通过的报文则会被丢弃，整个过程由硬件自动完成，不会占用 CPU 开销。

过滤器的位宽

每个 CAN 控制器提供 14 个位宽可变、可配置的过滤器组（0~13），每个过滤器组由 2 个 32 位寄存器，CAN_FiFB1 和 CAN_FiFB2 组成，通过配置 CAN 过滤器位宽配置寄存器（CAN_FBWCFG）的对应位，设置过滤器位宽为 2 个 16 位或者单个 32 位。

32 位宽的过滤器寄存器 CAN_FiFBx 包括：SID[10：0]、EID[17：0]、IDT 和 RTR 位。

CAN_FiFB1[31：21]	CAN_FiFB1[20：3]	CAN_FiFB1[2：0]		
CAN_FiFB2[31：21]	CAN_FiFB2[20：3]	CAN_FiFB2[2：0]		
SID[10：0]/EID[28：18]	EID[17：0]	IDT	RTR	0

2 个 16 位宽的过滤器寄存器 CAN_FiFBx 包括：SID[10：0]、IDT、RTR 和 EID[17：15]位。

CAN_FiFB1[31：21]	CAN_FiFB1 [20：19]	CAN_FiFB1 [18：16]	CAN_FiFB1[15：5]	CAN_FiFB1 [4：3]	CAN_FiFB1 [2：0]
CAN_FiFB2[31：21]	CAN_FiFB2 [20：19]	CAN_FiFB2 [18：16]	CAN_FiFB2[15：5]	AN_FiFB2 [4：3]	CAN_FiFB2 [2：0]

SID[10：0]	IDT	RTR	EID[17：15]	SID[10：0]	IDT	RTR	EID[17：15]

过滤器模式

通过设置 CAN_FMCFG 寄存器的 FMSELx 位可以设置过滤器寄存器工作在标识符掩码模式或者标识符列表模式，掩码模式用来指定哪些位与预设标识符相同，哪些位无需比较，列表模式表示标识符（ID 号）必须与预设标识符一致。两种模式与过滤器位宽配合使用，可以有以下四种过滤方式：

图 20-8 32位宽标识符掩码模式

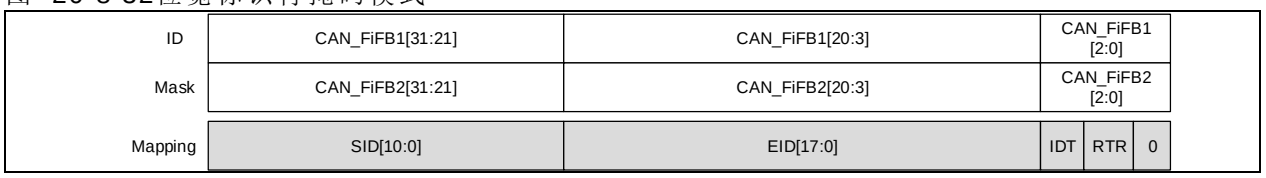


图 20-9 32位宽标识符列表模式

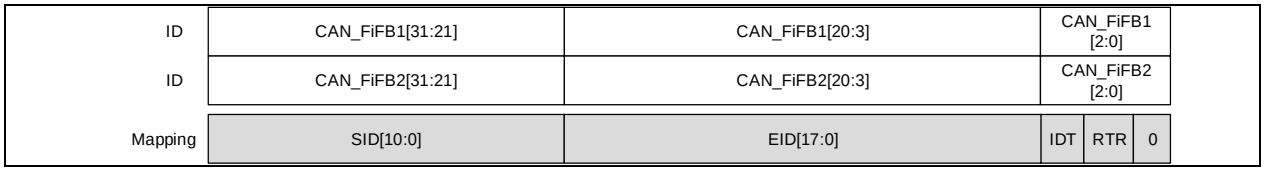


图 20-10 16位宽标识符掩码模式

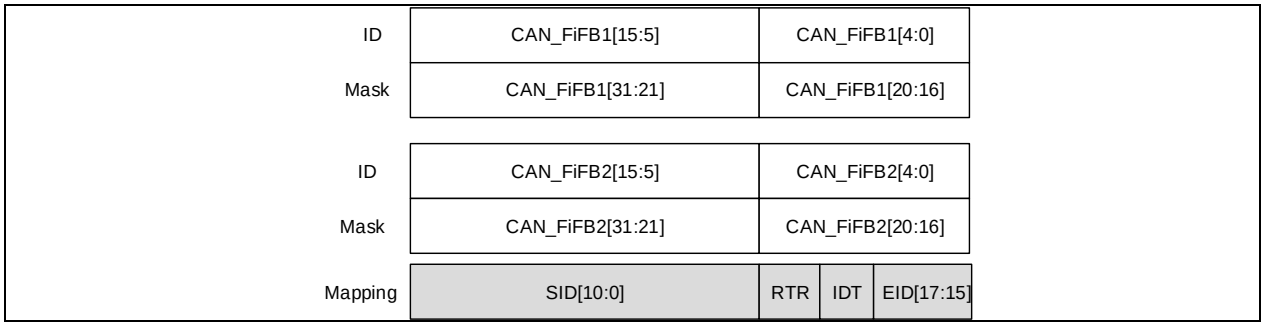
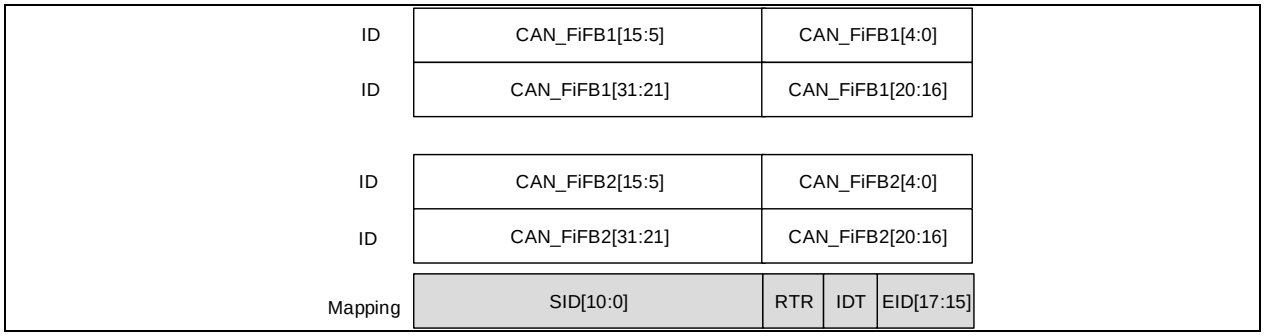


图 20-11 16位宽标识符列表模式



过滤器匹配序号

14 组过滤器组根据位宽模式的不同，具有不同的过滤效果，例如 32 位宽标识符掩码模式包含序号为 n 的过滤器，而 16 位宽标识符列表模式包含序号为 n、n+1、n+2 以及 n+3 的过滤器。一帧报文通过了某个序号（Filter Nnumber）N 的过滤器，则该帧的接收 FIFO 邮箱数据长度和时间戳寄存器 （CAN_RFCx）RFFMN[7：0]位存储该序号 N，过滤器序号的分配不关心对应的过滤器组是否处于激活状态。

下表为过滤器匹配序号的示例。

Filter bank	FIFO0	Active	Filter number
0	CAN_F0FB1[31：0]-ID	Yes	0
	CAN_F0FB2[31：0]-ID		1

Filter bank	FIFO1	Active	Filter number
3	CAN_F3FB1[15：0]-ID	Yes	0
	CAN_F3FB1[31：16]-ID		1

1	CAN_F1FB1[15：0]-ID	Yes	2
	CAN_F1FB1[31：16]-ID		3
	CAN_F1FB2[15：0]-ID		4
	CAN_F1FB2[31：16]-ID		5
2	CAN_F2FB1[31：0]-ID	Yes	6
	CAN_F2FB2[31：0]-Mask		
6	CAN_F6FB1[15：0]-ID	No	7
	CAN_F6FB1[31：16]-Mask		
	CAN_F6FB2[15：0]-ID		8
	CAN_F6FB2[31：16]-Mask		
9	CAN_F9FB1[31：0]-ID	No	9
	CAN_F9FB2[31：0]-ID		10
10	CAN_F10FB1[15：0]-ID	Yes	11
	CAN_F10FB1[31：16]-Mask		
	CAN_F10FB2[15：0]-ID		12
	CAN_F10FB2[31：16]-Mask		
12	CAN_F12FB1[15：0]-ID	No	13
	CAN_F12FB1[31：16]-ID		14
	CAN_F12FB2[15：0]-ID		15
	CAN_F12FB2[31：16]-ID		16

	CAN_F3FB2[15：0]-ID		2
	CAN_F3FB2[31：16]-ID		3
4	CAN_F4FB1[31：0]-ID	Yes	4
	CAN_F4FB2[31：0]-Mask		
5	CAN_F5FB1[15：0]-ID	No	5
	CAN_F5FB1[31：16]-Mask		
	CAN_F5FB2[15：0]-ID		6
	CAN_F5FB2[31：16]-Mask		
7	CAN_F7FB1[15：0]-ID	No	7
	CAN_F7FB1[31：16]-ID		8
	CAN_F7FB2[15：0]-ID		9
	CAN_F7FB2[31：16]-ID		10
8	CAN_F8FB1[31：0]-ID	Yes	11
	CAN_F8FB2[31：0]-Mask		
11	CAN_F11FB1[31：0]-ID	Yes	12
	CAN_F11FB2[31：0]-ID		13
13	CAN_F13FB1[15：0]-ID	Yes	14
	CAN_F13FB1[31：16]-ID		15
	CAN_F13FB2[15：0]-ID		16
	CAN_F13FB2[31：16]-ID		17

优先级匹配规则

CAN 控制器接收一帧报文，有可能能够通过多个过滤器的过滤，在这种情况下，存放在接收邮箱中的过滤器匹配序号，根据以下优先级规则确定。

- 位宽为 32 位的过滤器，优先级高于位宽为 16 位的过滤器。
- 在相同位宽的情况下，标识符列表模式的优先级高于标识符掩码模式。
- 在位宽和标识符模式都相同的情况下，标号越小的过滤器具有更高的优先级。

过滤器配置

- 将 CAN 过滤器控制寄存器 （CAN_FCTRL）FCS 位置‘1’，允许配置 CAN 过滤器。
- 写 CAN 过滤器模式配置寄存器 （CAN_FMCFG）FMSELx 位，控制过滤器工作模式为标识符掩码模式或者列表模式。
- 写 CAN 过滤器位宽配置寄存器 （CAN_FBWCFG）FBWSELx 位，控制过滤器位宽为 2 个 16 位或者单个 32 位。
- 写 CAN 过滤器 FIFO 关联寄存器 （CAN_FRF）FRFSELx 位，关联过滤器 x 到 FIFO0 或者 FIFO1。
- 将 CAN 过滤器激活控制寄存器 （CAN_FACFG）FAENx 位置‘1’，激活对应的过滤器组 x。
- 写 CAN_FiFBx（其中 i=0...13；x=1,2）,配置 0~13 组过滤器组。

将 CAN 过滤器控制寄存器 （CAN_FCTRL）FCS 位置‘0’，完成 CAN 过滤器配置过程。

20.6.5 报文发送

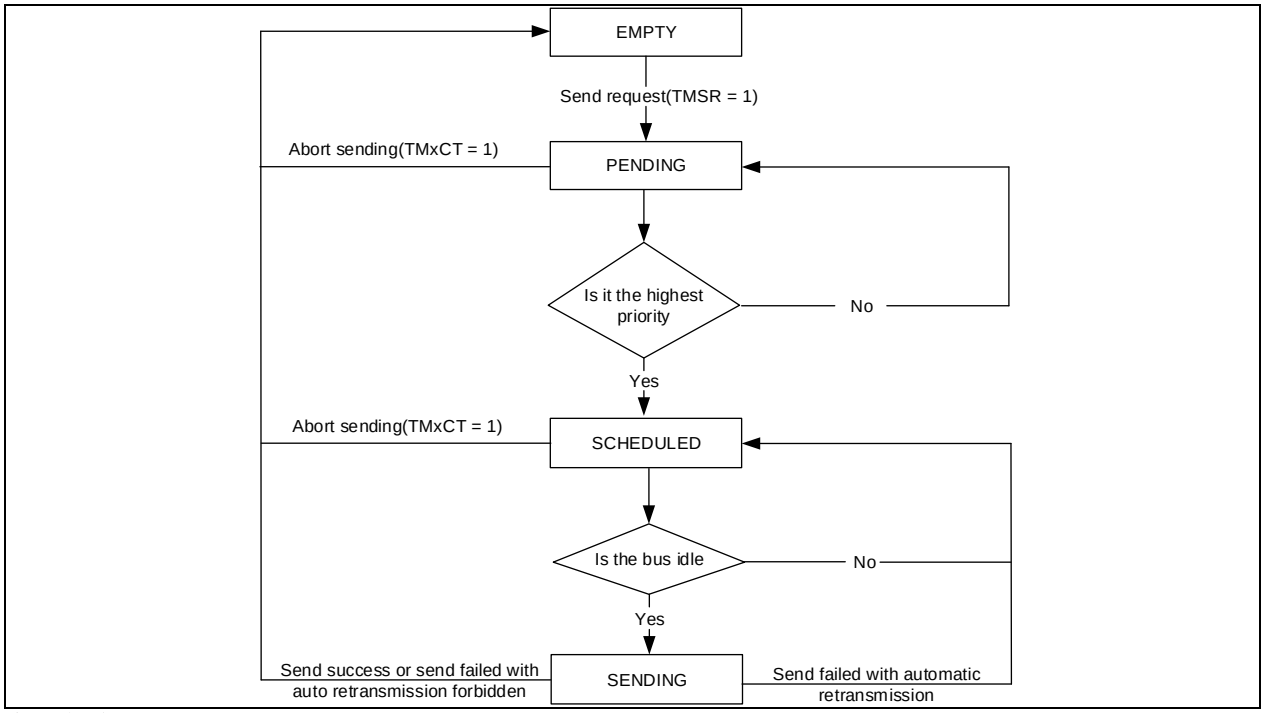
寄存器配置

数据发送首先需要选择发送邮箱进行配置，对应的寄存器为 CAN_TMIx、CAN_TMCx、CAN_TMDTLx 以及 CAN_TMDTHx。当邮箱配置完成后，对发送邮箱标识符寄存器 （CAN_TMIx）TMSR 位置‘1’控制 CAN 启动发送流程。

报文发送

当对应的邮箱配置完成且 CAN 控制器接收到发送请求后，该邮箱进入 PENDING 状态，此时 CAN 控制器会检查该邮箱是否处于最高优先级状态，如果是则进入 SCHEDULED 状态，否则停下等待该邮箱获取最高优先级。处于 SCHEDULED 状态的邮箱会实时监控 CAN 总线状态，只要总线空闲，预定发送邮箱中的报文就马上被发送。发送完成，该邮箱进入 EMPTY 状态。

图 20-12 发送邮箱状态转换



发送优先级配置

当有两个及以上发送邮箱处于 PENDING 状态时，需要决定邮箱的发送优先级。

由标识符决定：当 CAN 主控制寄存器（CAN_MCTRL）的 MMSSR 位置‘0’，发送顺序由邮箱中报文的标识符决定。标识符数值低的报文具有更高优先级，相同标识符的，邮箱号小的报文优先发送。

由发送请求顺序决定：当 CAN 主控制寄存器（CAN_MCTRL）的 MMSSR 位置‘1’，发送优先级由各邮箱的发送请求次序决定。

发送状态及错误信息

CAN 发送状态寄存器（CAN_TSTS）中的 TMxTCF、TMxTSF、TMxALF、TMxTEF 以及 TMxEF 用于显示发送状态和错误信息。

TMxTCF 位：发送完成标志。表示本次数据发送完成，置‘1’有效。

TMxTSF 位：无错误发送完成标志。表示本次数据发送完成且无错误，置‘1’有效。

TMxALF 位：发送仲裁丢失标志。表示本次数据发送仲裁失败，置‘1’有效。

TMxTEF 位：发送错误标志。表示本次数据发送检测到总线错误，且发送错误帧，置‘1’有效。

TMxEF 位：邮箱空标志。表示本次数据发送完成，邮箱变为空状态，置‘1’有效。

数据发送中止

可以通过将 CAN 发送状态寄存器（CAN_TSTS）的 TMxCT 位置‘1’中止当前邮箱的发送，具体情况需要分类讨论。

当前邮箱发送失败或者丢失仲裁，假如报文自动重传功能被禁止，则发送邮箱进入 EMPTY 状态；假如报文自动重传功能被使能，则发送邮箱进入 SCHEDULED 状态，接着邮箱发送被中止，进入 EMPTY 状态。

当前邮箱本次数据发送完成且无错误，邮箱进入 EMPTY 状态。

20.6.6 报文接收

寄存器配置

用户程序通过读接收 FIFO 邮箱标识符寄存器（CAN_RFIx）、接收 FIFO 邮箱数据长度和时间戳寄存器（CAN_RFCx）、接收 FIFO 邮箱低字节数据寄存器（CAN_RFDTLx）以及接收 FIFO 邮箱高字节数据寄存器（CAN_RFDTHx）获取接收到的有效报文。

报文接收

CAN 控制器具有两个深度为 3 的 FIFO 用于接收报文，采用先进先出的原则。当报文被正确接收且通过了标识符过滤，则被认为是有效报文并存储在对应的 FIFO 中。接收 FIFO 每接收到一帧有效报文，CAN_RFx 寄存器中的报文数目 RFxMN[1：0]就加 1，当 RFxMN[1：0]等于 3 的同时又接收到一帧有效报文，此时控制器会根据 CAN 主控制寄存器（CAN_MCTRL）的 MDRSEL 位选择覆盖接收到的原有的报文或者丢弃该报文。

同时，当用户每次读出一帧报文且控制 CAN_RFx 寄存器 RFxR 位置‘1’，则对应 FIFO 释放一个深度空间，并且 CAN_RFx 寄存器中的报文数目 RFxMN[1：0]减 1。

接收 FIFO 状态

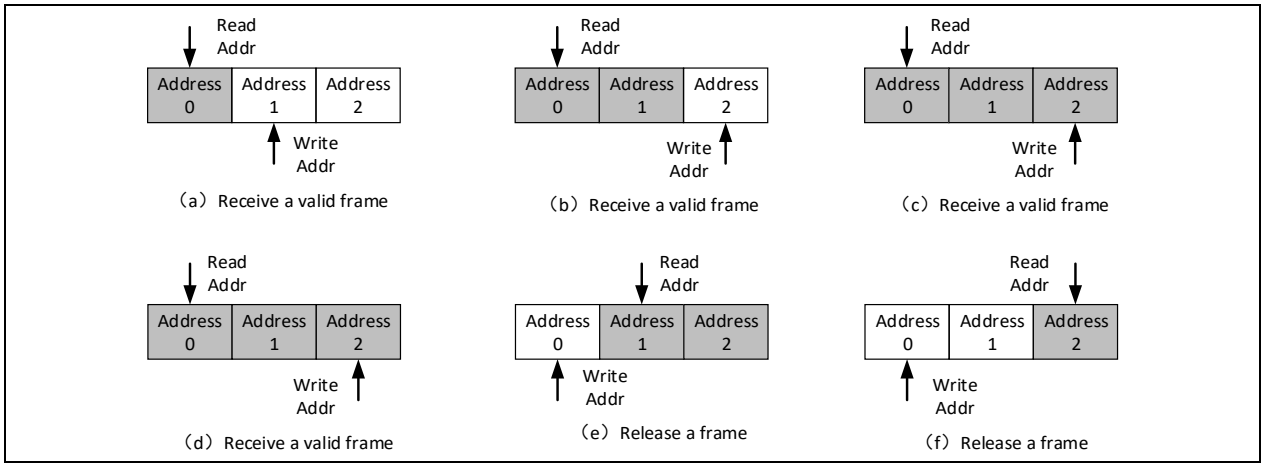
寄存器 RFx 中的 RFxMN[1：0]、RFxFF 以及 RFxOF 用于显示接收 FIFO 的状态信息。

RFxMN[1：0]：表示 FIFOx 中当前存储有效报文的数目。

RFxFF：表示 FIFOx 中当前存储 3 帧有效报文处于‘满’状态，如图（c）所示。

RFxOF：表示 FIFOx 中当前有 3 帧有效报文同时又接收到一帧有效报文，处于溢出状态，如图（d）所示。

图 20-13 接收FIFO状态



20.6.7 出错管理

CAN 总线的状态可以根据发送错误计数值 TEC 和接收错误计数值 REC 表明当前 CAN 节点所处的状态，同时 CAN 错误状态寄存器（CAN_ESTS）的 ETR[6：4]位用于记录上次错误的原因，这些错误状态在 CAN 中断使能寄存器（CAN_INTEN）控制下产生中断。

- 主动错误状态：当 TEC 且 REC 计数值都小于 128 时，系统处于主动错误状态，当检测到错误时发送主动错误标志。
- 被动错误状态：当 TEC 或 REC 计数值大于 127 时，系统处于被动错误状态，当检测到错误时发送被动错误标志。

离线状态：当 TEC 大于 255 时，系统进入离线状态，处于离线状态的节点不能发送接收报文，从离线状态恢复分两种情况。当 CAN 主控制寄存器（CAN_MCTRL）AEBOEN 位为‘0’时，通信模式下 CAN 节点 RX 检测到 128 次 11 个连续隐性位，接着软件请求进入冻结模式，随后从离线状态恢复。当 AEBOEN 位为‘1’时，通信模式下 CAN 节点 RX 检测到 128 次 11 个连续隐性位，随后自动从离线状态恢复。

20.7CAN寄存器

必须以字（32 位）的方式操作这些外设寄存器。

表 20-1 CAN寄存器映像和复位值

寄存器简称	基址偏移量	复位值
MCTRL	000h	0x0001 0002
MSTS	004h	0x0000 0C02
TSTS	008h	0x1C00 0000
RF0	00Ch	0x0000 0000
FR1	010h	0x0000 0000
INTEN	014h	0x0000 0000
ESTS	018h	0x0000 0000
BTMG	01Ch	0x0123 0000
保留	020h~17Fh	xx
TMI0	180h	0xXXXX XXXX
TMC0	184h	0xXXXX XXXX
TMDTL0	188h	0xXXXX XXXX
TMDTH0	18Ch	0xXXXX XXXX
TMI1	190h	0xXXXX XXXX
TMC1	194h	0xXXXX XXXX
TMDTL1	198h	0xXXXX XXXX
TMDTH1	19Ch	0xXXXX XXXX
TMI2	1A0h	0xXXXX XXXX
TMC2	1A4h	0xXXXX XXXX
TMDTL2	1A8h	0xXXXX XXXX
TMDTH2	1ACh	0xXXXX XXXX
RFI0	1B0h	0xXXXX XXXX
RFC0	1B4h	0xXXXX XXXX
RFDTL0	1B8h	0xXXXX XXXX
RFDTH0	1BCh	0xXXXX XXXX
RFI1	1C0h	0xXXXX XXXX
RFC1	1C4h	0xXXXX XXXX
RFDTL1	1C8h	0xXXXX XXXX
RFDTH1	1CCh	0xXXXX XXXX
保留	1D0h~1FFh	xx
FCTRL	200h	0x2A1C 0E01
FMCFG	204h	0x0000 0000
保留	208h	xx
FBWCFG	20Ch	0x0000 0000
保留	210h	xx
FRF	214h	0x0000 0000
保留	218h	xx
FACFG	21Ch	0x0000 0000
保留	220h~23Fh	xx

F0FB1	240h	0xXXXX XXXX
F0FB2	244h	0xXXXX XXXX
F1FB1	248h	0xXXXX XXXX
F1FB2	24Ch	0xXXXX XXXX
…	…	…
F13FB1	2A8h	0xXXXX XXXX
F13FB2	2ACh	0xXXXX XXXX

20.7.1 CAN控制和状态寄存器

20.7.1.1 CAN主控制寄存器（CAN_MCTRL）

域	简称	复位值	类型	功能
位 31：17	保留	0x0000	resd	请保持默认值。
位 16	PTD	0x1	rw	调试时禁止收发（Prohibit trans when debug） 0：不禁止； 1：禁止。仍然可以正常地读写和控制接收 FIFO。 注：仅 PTD 及 DEBUG_CTRL 寄存器的 CANx_PAUSE 同时置位时，才会实现禁止收发的效果。
位 15	SPRST	0x0	rw1s	部分软复位（Software partial reset） 0：不复位； 1：部分复位。 注： SPRST 指只复位接收 FIFO 及 MCTRL 寄存器。 复位后 CAN 进入睡眠模式。此后硬件自动对该位清零。
位 14：8	保留	0x00	resd	请保持默认值。
位 7	TTCEN	0x0	rw	时间触发通信模式使能（Time triggered communication mode enable） 0：关闭； 1：开启。
位 6	AEBOEN	0x0	rw	自动退出离线状态使能（Automatic exit bus-off enable） 0：关闭； 1：开启。 注： 当开启时，硬件只需检测到 CAN 总线上出现退出时序就自动退出； 当关闭时，需要软件执行一次额外的冻结模式的进入以及退出动作，接着在 CAN 总线上检测到退出时序时才会退出离线状态。
位 5	AEDEN	0x0	rw	自动退出睡眠模式使能（Automatic exit doze mode enable） 0：关闭； 1：开启。 注： 当关闭时，需软件写清睡眠请求命令来退出； 当开启时，无需软件干预，只要检测到 CAN 总线上出现报文时就立即退出睡眠模式。
位 4	PRSFEN	0x0	rw	发送失败时禁止重传使能（Prohibit retransmission when sending fails enable） 0：关闭； 1：开启。
位 3	MDRSEL	0x0	rw	接收溢出时报文丢弃规则选择 （Message discarding rule select when overflow） 0：最先收到的报文被丢弃； 1：最新收到的报文被丢弃。
位 2	MMSSR	0x0	rw	多报文发送顺序规则选择 （Multiple message sending sequence rule） 0：标识符最小的最先被发送；

				1：最先请求的最先被发送。
位 1	DZEN	0x1	rw	睡眠模式使能 （Doze mode enable） 0：关闭； 1：开启。 注： 当设置了 AEDEN，且检测到 CAN 总线上出现报文时，硬件会自动退出睡眠模式； 在 CAN 复位或部分软复位后，该位被硬件强制置位，即 CAN 默认将处于睡眠模式。
位 0	FZEN	0x0	rw	冻结模式使能 （Freeze mode enable） 0：关闭； 1：开启。 注： 当写关闭命令时，会在检测到接收管脚上出现连续的 11 个隐性位才会实际退出。因此软件需等待 FZC 被硬件清零来确认。 当写开启命令时，会在当前的 CAN 活动（发送或接收）结束后才会实际进入。因此软件需等待 FZC 被硬件置位来确认。

20.7.1.2 CAN主状态寄存器 （CAN_MSTS）

域	简称	复位值	类型	功能
位 31：12	保留	0x00000	resd	请保持默认值。
位 11	REALRX	0x1	ro	接收管脚实时电平（Real time level of RX pin） 0：低电平； 1：高电平。
位 10	LSAMPRX	0x1	ro	接收管脚上次采样电平（Last sample level of RX pin） 0：低电平； 1：高电平。 注：此值会跟随 REALRX 实时更新。
位 9	CURS	0x0	ro	当前的接收状态（Currently receiving status） 0：未接收； 1：正在接收。 注：在 CAN 开始接收时硬件置位此标志，接收完毕后硬件自动清除。
位 8	CUSS	0x0	ro	当前的发送状态（Currently sending status） 0：未发送； 1：正在发送。 注：在 CAN 开始发送时硬件置位此标志，发送完毕后硬件自动清除。
位 7：5	保留	0x0	resd	请保持默认值。
位 4	EDZIF	0x0	rw1c	进入睡眠模式的中断标志（Enter doze mode interrupt flag） 0：未进入或无标志置位条件； 1：已进入。 注： 只有当 EDZIEN=1，且 CAN 进入睡眠模式时才会由硬件置位此标志。此标志的置位将会产生一个状态改变中断。 此标志可由软件清零（对自身写一），或当 DZC 位被清零时硬件自动对本标志清零。
位 3	QDZIF	0x0	rw1c	退出睡眠模式的中断标志 （Quit doze mode interrupt flag） 0：未退出或无退出条件； 1：已退出或产生了退出条件。 注： 该位由软件将其清零（对自身写一）。 退出条件为检测到总线上出现帧起始位（SOF）。 如果 QDZIEN=1，此标志的置位将会产生一个状态改变中断。
位 2	EOIF	0x0	rw1c	出现错误的中断标志 （Error occur Interrupt flag）

				0：未出现或无标志置位条件； 1：已出现。 注： 该位由软件将其清零（对自身写一）。 仅当 CAN 错误状态寄存器（CAN_ESTS）中的某位被置位，且其对应的 CAN 中断使能寄存器（CAN_INTEN）的相应中断使能位处于使能状态时，该标志才会被硬件置位。此标志的置位将会产生一个状态改变中断。
位 1	DZC	0x1	ro	睡眠模式确认（Doze mode confirm） 0：未处于睡眠模式； 1：正处于睡眠模式中。 注： 该位用于确定 CAN 当前是否处于睡眠模式，是对软件请求进入睡眠模式的确认。 当进入睡眠模式时，会在当前的 CAN 活动（发送或接收）结束后才会实际进入。因此软件需等待本标志被硬件置位来确认进入睡眠模式。 当退出睡眠模式（即软件写关闭睡眠模式命令，或者自动退出睡眠模式使能状态下检测到 CAN 总线上出现报文）时，会在检测到 CAN 的 RX 管脚上出现连续的 11 位隐性位时才会实际退出。因此软件需等待本标志被硬件清零来确认退出睡眠模式。
位 0	FZC	0x0	ro	冻结模式确认（Freeze mode confirm） 0：未处于冻结模式； 1：正处于冻结模式中。 注： 该位用于确定 CAN 当前是否处于冻结模式，是对软件请求进入冻结模式的确认。 当进入冻结模式时，会在当前的 CAN 活动（发送或接收）结束后才会实际进入。因此软件需等待本标志被硬件置位来确认进入冻结模式。 当退出冻结模式时，会在检测到 CAN 的 RX 管脚上出现连续的 11 位隐性位时才会实际退出。因此软件需等待本标志被硬件清零来确认退出冻结模式。

20.7.1.3 CAN发送状态寄存器 （CAN_TSTS）

域	简称	复位值	类型	功能
位 31	TM2LPF	0x0	ro	邮箱 2 优先级最低标志（Transmit mailbox 2 lowest priority flag） 0：非最低优先级； 1：最低优先级（表明多个邮箱在等待发送报文时，邮箱 2 的优先级最低）。
位 30	TM1LPF	0x0	ro	邮箱 1 优先级最低标志（Transmit mailbox 1 lowest priority flag） 0：非最低优先级； 1：最低优先级（表明多个邮箱在等待发送报文时，邮箱 1 的优先级最低）。
位 29	TM0LPF	0x0	ro	邮箱 0 最低优先级标志（Transmit mailbox 0 lowest priority flag） 0：非最低优先级； 1：最低优先级（表明多个邮箱在等待发送报文时，邮箱 0 的优先级最低）。
位 28	TM2EF	0x1	ro	发送邮箱 2 空标志（Transmit mailbox 2 empty flag） 当发送邮箱 2 中没有等待发送的报文时，硬件置位该位。
位 27	TM1EF	0x1	ro	发送邮箱 1 空标志（Transmit mailbox 1 empty flag） 当发送邮箱 1 中没有等待发送的报文时，硬件置位该位。
位 26	TM0EF	0x1	ro	发送邮箱 0 空标志（Transmit mailbox 0 empty flag） 当发送邮箱 0 中没有等待发送的报文时，硬件置位该位。
位 25：24	TMNR	0x0	ro	发送邮箱号记录 （Transmit Mailbox number record） 注：

				当有发送邮箱为空时，这两位表示接下来将要使用的空置邮箱号。 示例：CAN 空闲状态下，写一个报文的发送命令后，这 2 位的值将变为 01。 当没有发送邮箱为空时，这两位表示优先级最低的那个发送邮箱号。 示例：3 个报文待发，报文标识符依次为，邮箱 0 为 0x400，邮箱 1 为 0x433，邮箱 2 为 0x411，此时这 2 位的值将变为 01。
位 23	TM2CT	0x0	rw1s	邮箱 2 取消发送 （Transmit mailbox 2 cancel transmit） 0：无意义； 1：取消发送。 注：软件设置此位可中断邮箱 2 的发送，硬件清除邮箱 2 的发送报文后同步清除该位。若邮箱 2 为空置邮箱时，软件置位该位没有任何意义。
位 22：20	保留	0x0	resd	保持默认值。
位 19	TM2TEF	0x0	rw1c	邮箱 2 发送错误标志 （Transmit mailbox 2 transmission error flag） 0：无错误； 1：出现错误。 注： 当邮箱 2 出现发送错误时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。
位 18	TM2ALF	0x0	rw1c	邮箱 2 仲裁丢失标志 （Transmit mailbox 2 arbitration lost flag） 0：无仲裁问题； 1：出现仲裁丢失。 注： 当邮箱 2 因仲裁丢失导致发送失败时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。
位 17	TM2TSF	0x0	rw1c	邮箱 2 发送成功标志 （Transmit mailbox 2 transmission success flag） 0：发送失败； 1：发送成功。 注： 该位实时指示每次邮箱 2 的发送结果。可由软件对该位写一清零。
位 16	TM2TCF	0x0	rw1c	邮箱 2 发送完成标志 （transmit mailbox 2 transmission completed flag） 0：正在发送； 1：发送完成。 注： 每次对邮箱 2 的请求（发送或中止）完成后，由硬件置位该位。 该位可由软件写一清零。或当接收到新的发送请求时由硬件自动清除。 当该位被清除时， 邮箱 2 的 TSMF2、ALMF2、TEMF2 也会同步被硬件清除。
位 15	TM1CT	0x0	rw1s	邮箱 1 取消发送 （Transmit mailbox 1 cancel transmit） 0：无意义； 1：取消发送。 注：软件设置此位可禁止邮箱 1 的发送，硬件清除邮箱 1 的发送报文后同步清除该位。若邮箱 1 为空置邮箱时，软件置位该位没有任何意义。
位 14：12	保留	0x0	resd	保持默认值。
位 11	TM1TEF	0x0	rw1c	邮箱 1 发送错误标志 （Transmit mailbox 1 transmission error flag） 0：无错误； 1：出现错误。

				注： 当邮箱 1 出现发送错误时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。
位 10	TM1ALF	0x0	rw1c	邮箱 1 仲裁丢失标志 （Transmit mailbox 1 arbitration lost flag） 0：无仲裁问题； 1：出现仲裁丢失。 注： 当邮箱 1 因仲裁丢失导致发送失败时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。
位 9	TM1TSF	0x0	rw1c	邮箱 1 发送成功标志 （Transmit mailbox 1 transmission success flag） 0：发送失败； 1：发送成功。 注： 该位实时指示每次邮箱 1 的发送结果。可由软件对该位写一清零。
位 8	TM1TCF	0x0	rw1c	邮箱 1 发送完成标志 （Transmit mailbox 1 transmission completed flag） 0：正在发送； 1：发送完成。 注： 每次对邮箱 1 的请求（发送或中止）完成后，由硬件置位该位。 该位可由软件写一清零。或当接收到新的发送请求时由硬件自动清除。 当该位被清除时， 邮箱 1 的 TSMF1、ALMF1、TEMF1 也会同步被硬件清除。
位 7	TM0CT	0x0	rw1s	邮箱 0 取消发送 （Transmit mailbox 0 cancel transmit） 0：无意义； 1：取消发送。 注：软件设置此位可禁止邮箱 0 的发送，硬件清除邮箱 0 的发送报文后同步清除该位。若邮箱 0 为空置邮箱时，软件置位该位没有任何意义。
位 6：4	保留	0x0	resd	保持默认值。
位 3	TM0TEF	0x0	rw1c	邮箱 0 发送错误标志 （Transmit mailbox 0 transmission error flag） 0：无错误； 1：出现错误。 注： 当邮箱 0 出现发送错误时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。
位 2	TM0ALF	0x0	rw1c	邮箱 0 仲裁丢失标志 （Transmit mailbox 0 arbitration lost flag） 0：无仲裁问题； 1：出现仲裁丢失。 注： 当邮箱 0 因仲裁丢失导致发送失败时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。
位 1	TM0TSF	0x0	rw1c	邮箱 0 发送成功标志 （Transmit mailbox 0 transmission success flag） 0：发送失败； 1：发送成功。 注： 该位实时指示每次邮箱 0 的发送结果。可由软件对该位写一清零。

位 0	TM0TCF	0x0	rw1c	邮箱 0 发送完成标志 （Transmit mailbox 0 transmission completed flag） 0：正在发送； 1：发送完成。 注： 每次对邮箱 0 的请求（发送或中止）完成后，由硬件置位该位。 该位可由软件写一清零。或当接收到新的发送请求时由硬件自动清除。 当该位被清除时， 邮箱 0 的 TSMF0、ALMF0、TEMF0 也会同步被硬件清除。

20.7.1.4 CAN接收FIFO 0寄存器 （CAN_RF0）

域	简称	复位值	类型	功能
位 31：6	保留	0x0000000	resd	保持默认值。
位 5	RF0R	0x0	rw1s	释放接收 FIFO 0 （Receive FIFO 0 release） 0：无意义； 1：释放 FIFO。 注： 软件设置此位可释放接收 FIFO 0，当 FIFO 0 被释放时，硬件对该位清零。 接收 FIFO 0 为空时，软件置位该位没有任何意义。 若 FIFO 0 中有 2 个以上的报文时，软件需要执行一次释放命令后才能访问第 2 个报文。
位 4	RF0OF	0x0	rw1c	接收 FIFO 0 溢出标志 （Receive FIFO 0 overflow flag） 0：无溢出； 1：有溢出。 注： 当 FIFO 0 已满时，又收到了新的符合过滤条件的报文，硬件将置位该位。 该位由软件写一清零。
位 3	RF0FF	0x0	rw1c	接收 FIFO 0 满标志 （Receive FIFO 0 full flag） 0：未满； 1：已满。 注： 当 FIFO 0 中存储 3 笔待读取的报文时，硬件将置位该位。 该位由软件写一清零。
位 2	保留	0x0	resd	保持默认值。
位 1：0	RF0MN	0x0	ro	FIFO 0 报文数目 （Receive FIFO 0 message num） 注： 这 2 位表示存储在 FIFO 0 中的待读取或者处理的报文数目。 当 FIFO 0 未满时，每收到了一笔新的符合过滤条件的报文，硬件就对 RF0MN 加 1。 每当软件对 RF0R 位写一来释放接收 FIFO 0 时，RF0MN 就会被减 1，直到其值为 0。

20.7.1.5 CAN接收FIFO 1寄存器（CAN_RF1）

域	简称	复位值	类型	功能
位 31：6	保留	0x0000000	resd	保持默认值。
位 5	RF1R	0x0	rw1s	释放接收 FIFO 1 （Receive FIFO 1 release） 0：无意义； 1：释放 FIFO。 注： 软件设置此位可释放接收 FIFO 1，当 FIFO 1 被释放时，硬件对该位清零。 接收 FIFO 1 为空时，软件置位该位没有任何意义。 若 FIFO 1 中有 2 个以上的报文时，软件需要执行一次释放命令后才能访问第 2 个报文。

位 4	RF1OF	0x0	rw1c	接收 FIFO 1 溢出标志 （Receive FIFO 1 overflow flag） 0：无溢出； 1：有溢出。 注： 当 FIFO 1 已满时，又收到了新的符合过滤条件的报文，硬件将置位该位。 该位由软件写一清零。
位 3	RF1FF	0x0	rw1c	接收 FIFO 1 满标志 （Receive FIFO 1 full flag） 0：未满； 1：已满。 注： 当 FIFO 1 中存储 3 笔待读取的报文时，硬件将置位该位。 该位由软件写一清零。
位 2	保留	0x0	resd	保持默认值。
位 1：0	RF1MN	0x0	ro	FIFO 1 报文数目 （Receive FIFO 1 message num） 注： 这 2 位表示存储在 FIFO 1 中的待读取或者处理的报文数目。 当 FIFO 1 未满时，每收到了一笔新的符合过滤条件的报文，硬件就对 RF1MN 加 1。 每当软件对 RF1R 位写一来释放接收 FIFO 1 时，RF1MN 就会被硬件减 1，直到其值为 0。

20.7.1.6 CAN中断使能寄存器 （CAN_INTEN）

域	简称	复位值	类型	功能
位 31：18	保留	0x0000	resd	保持默认值。
位 17	EDZIEN	0x0	rw	进入睡眠模式的中断使能 （Enter doze mode interrupt enable） 0：关闭； 1：开启。 注：此中断对应的标志位为 EDZIF，故仅本中断使能且 EDZIF 被置位时才会产生中断。
位 16	QDZIEN	0x0	rw	退出睡眠模式的中断使能 （Quit doze mode interrupt enable） 0：关闭； 1：开启。 注：此中断对应的标志位为 QDZIF，故仅本中断使能且 QDZIF 被置位时才会产生中断。
位 15	EOIEN	0x0	rw	出现错误的中断使能 （Error occur interrupt enable） 0：关闭； 1：开启。 注：此中断对应的标志位为 EOIF，故仅本中断使能且 EOIF 被置位时才会产生中断。
位 14：12	保留	0x0	resd	保持默认值。
位 11	ETRIEN	0x0	rw	错误类型记录中断使能 （Error type record interrupt enable） 0：关闭； 1：开启。 注：只有此中断使能后，硬件设置 ETR[2：0]时，才会同步设置 EOIF 位为’1’。
位 10	BOIEN	0x0	rw	总线关闭中断使能 （Bus-off interrupt enable） 0：关闭； 1：开启。 注：只有此中断使能后，硬件设置 BOF 时，才会同步设置 EOIF 位为’1’。
位 9	EPIEN	0x0	rw	错误被动中断使能 （Error passive interrupt enable） 0：关闭； 1：开启。

				注：只有此中断使能后，硬件设置 EPF 时，才会同步设置 EOIF 位为'1'。
位 8	EAIEN	0x0	rw	错误警告中断使能 （Error active interrupt enable） 0：关闭； 1：开启。 注：只有此中断使能后，硬件设置 EAF 时，才会同步设置 EOIF 位为'1'。
位 7	保留	0x0	resd	保持默认值。
位 6	RF1OIEN	0x0	rw	接收 FIFO 1 溢出中断使能 （Receive FIFO 1 overflow interrupt enable） 0：关闭； 1：开启。 注：此中断对应的标志位为 RF1OF，故仅本中断使能且 RF1OF 被置位时才会产生中断。
位 5	RF1FIEN	0x0	rw	接收 FIFO 1 满中断使能 （Receive FIFO 1 full interrupt enable） 0：关闭； 1：开启。 注：此中断对应的标志位为 RF1FF，故仅本中断使能且 RF1FF 被置位时才会产生中断。
位 4	RF1MIEN	0x0	rw	接收 FIFO 1 报文接收中断使能 （FIFO 1 receive message interrupt enable） 0：关闭； 1：开启。 注：此中断对应的标志位为 RF1MN，故仅本中断使能且 RF1MN 为非零时才会产生中断。
位 3	RF0OIEN	0x0	rw	接收 FIFO 0 溢出中断使能 （Receive FIFO 0 overflow interrupt enable） 0：关闭； 1：开启。 注：此中断对应的标志位为 RF0OF，故仅本中断使能且 RF0OF 被置位时才会产生中断。
位 2	RF0FIEN	0x0	rw	接收 FIFO 0 满中断使能 （Receive FIFO 0 full interrupt enable） 0：关闭； 1：开启。 注：此中断对应的标志位为 RF0FF，故仅本中断使能且 RF0FF 被置位时才会产生中断。
位 1	RF0MIEN	0x0	rw	接收 FIFO 0 报文接收中断使能 （FIFO 0 receive message interrupt enable） 0：关闭； 1：开启。 注：此中断对应的标志位为 RF0MN，故仅本中断使能且 RF0MN 为非零时才会产生中断。
位 0	TCIEN	0x0	rw	发送邮箱发送完成中断使能 （Transmit mailbox empty interrupt enable） 0：关闭； 1：开启。 注：此中断对应的标志位为 TMxTCF，故仅本中断使能且 TMxTCF 被置位时才会产生中断。

20.7.1.7 CAN错误状态寄存器 （CAN_ESTS）

域	简称	复位值	类型	功能
位 31：24	REC	0x00	ro	接收错误计数器 （Receive error counter） 这个计数器按照 CAN 协议的故障界定机制的接收部分实现。
位 23：16	TEC	0x00	ro	发送错误计数器 （Transmit error counter） 这个计数器按照 CAN 协议的故障界定机制的发送部分实现。
位 15：7	保留	0x00	resd	保持默认值。

位 6：4	ETR	0x0	rw	错误类型记录 （Error type record） 000： 没有错误； 001： 位填充错误； 010： 格式错误； 011： 确认错误； 100： 隐性位错误； 101： 显性位错误； 110： CRC 错误； 111： 由软件设置。 注： 这三位用于记录最新错误类型，由硬件根据 CAN 总线上的出错情况设置。当报文被正确发送或接收后，硬件自动将这三位清零。 硬件没有使用错误代码 7，软件可以设置该值，从而可以检测代码的更新。
位 3	保留	0x0	resd	保持默认值。
位 2	BOF	0x0	ro	总线关闭标志 （Bus-off flag） 0：未处于总线关闭状态； 1：处于总线关闭状态。 注：当发送错误计数器 TEC 溢出（即大于 255）时，CAN 进入总线关闭状态，硬件对该位置’1’。
位 1	EPF	0x0	ro	错误被动标志 （Error passive flag） 0：未处于错误被动状态； 1：处于错误被动状态。 注：当前记录的出错次数达到错误被动状态（即接收错误计数器或发送错误计数器的值>127）时，硬件对该位置’1’。
位 0	EAF	0x0	ro	错误主动标志 （Error active flag） 0：未处于错误主动状态； 1：处于错误主动状态。 注：当前记录的出错次数达到错误主动状态（即接收错误计数器或发送错误计数器的值≥96）时，硬件对该位置’1’。

20.7.1.8 CAN位时序寄存器 （CAN_BTMG）

域	简称	复位值	类型	功能
位 31	LOEN	0x0	rw	只听模式使能（Listen-Only mode） 0：关闭； 1：开启。
位 30	LBEN	0x0	rw	回环模式使能（Loop back mode） 0：关闭； 1：开启。
位 29：26	保留	0x0	resd	保持默认值。
位 25：24	RSAW	0x1	rw	重新同步调整宽度 （Resynchronization width） tRSAW = tCAN x （RSAW[1：0] + 1）。 注：该位域定义了 CAN 硬件在每位中可以延长或缩短多少个时间单元的上限。
位 23	保留	0x0	resd	保持默认值。
位 22：20	BTS2	0x2	rw	位时间段 2 （Bit time segment 2） tBTS2 = tCAN x （BTS2[2：0] + 1）。 注：该位域定义了位时间段 2 占用了多少个时间单元。
位 19：16	BTS1	0x3	rw	位时间段 1 （Bit time segment 1） tBTS1 = tCAN x （BTS1[3：0] + 1） 注：该位域定义了位时间段 1 占用了多少个时间单元。
位 15：12	保留	0x0	resd	保持默认值。
位 11：0	BRDIV	0x000	rw	波特率分频器 （Baud rate division） tq = （BRDIV[11：0]+1） x tPCLK 注：该位域定义了时间单元（tq）的时间长度。

20.7.2 CAN邮箱寄存器

本节描述发送和接收邮箱寄存器。关于寄存器映像的详细信息，请参考 20.6.5 节报文。

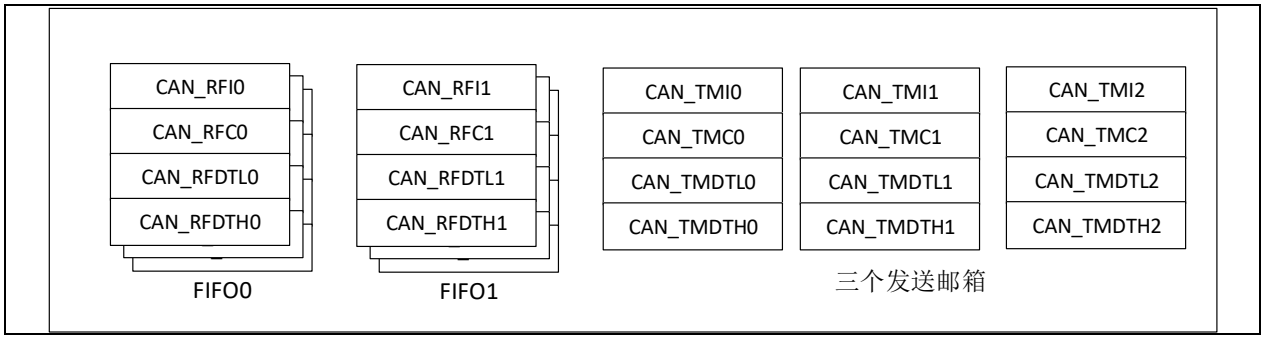
除了下述例外，发送和接收邮箱几乎一样：

- 接收 FIFO 邮箱数据长度和时间戳寄存器（CAN_RFCx）的 RFFMN 域；
- 接收邮箱是只读的；
- 发送邮箱只有在它为空时才是可写的，CAN 发送状态寄存器（CAN_TSTS）的相应 TM2S 位为'1'，表示发送邮箱为空。

共有 3 个发送邮箱和 2 个接收邮箱。每个接收邮箱为 3 级深度的 FIFO，并且只能访问 FIFO 中最先收到的报文。

每个邮箱包含 4 个寄存器。

图 20-14 发送和接收邮箱



20.7.2.1 发送邮箱标识符寄存器 （CAN_TMIx） （x=0..2）

注意：1.当其所属的邮箱处在等待发送的状态时，该寄存器是写保护的。
2.该寄存器实现了发送请求控制功能（第 0 位）一复位值为 0。

域	简称	复位值	类型	功能
位 31：21	TMSID/ TMEID	0xXXX	rw	发送邮箱标准标识符或扩展标识符高字节（Transmit mailbox standard identifier or extended identifier high bytes） 注：这 11 位为标准标识符或扩展标识符的高 11 位。
位 20：3	TMEID	0xXXXXX	rw	发送邮箱扩展标识符低字节（Ttransmit mailbox extended identifier） 注：这 18 位为扩展标识符的低 18 位。
位 2	TMIDSEL	0xX	rw	标识符类型选择（Transmit mailbox identifier type select） 0：标准标识符； 1：扩展标识符。
位 1	TMFRSEL	0xX	rw	发送邮箱帧类型选择（Transmit mailbox frame type select） 0：数据帧； 1：远程帧。
位 0	TMSR	0x0	rw	发送邮箱的数据发送请求（transmit mailbox send request） 0：无意义； 1：请求发送。 注：当数据发送完成，邮箱为空时，硬件对其清'0'。

20.7.2.2 发送邮箱数据长度和时间戳寄存器 （CAN_TMCx）（x=0..2）

当邮箱不在空置状态时，该寄存器的所有位为写保护。

域	简称	复位值	类型	功能
位 31：16	TMTS	0xXXXX	rw	发送邮箱的报文时间戳 （Transmit mailbox time stamp） 注：该时间戳为报文发送帧起始时刻采样到的 CAN 内部定时器的值。
位 15：9	保留	0xXX	resd	保持默认值。

位 8	TMTSTEN	0xX	rw	时间戳的发送使能（Transmit mailbox time stamp transmit enable） 0：不发送； 1：发送。 注： 只有时间触发通信模式使能后，该位才有意义。 时间戳 MTS[15：0]中，MTS[7：0]存放于 TMDT7，MTS[15：8]存放于 TMDT6。故为发送时间戳，发送数据长度需要被设定为 8。
位 7：4	保留	0xX	resd	保持默认值。
位 3：0	TMDTBL	0xX	rw	发送数据长度 （Transmit mailbox data byte length） 注：该域指定了发送报文的数据长度。其中 1 个报文可包含 0 到 8 个字节数据。

20.7.2.3 发送邮箱低字节数据寄存器 （CAN_TMDTLx） （x=0..2）

当邮箱不在空置状态时，该寄存器的所有位为写保护。

域	简称	复位值	类型	功能
位 31：24	TMDT3	0xXX	rw	发送邮箱数据字节 3 （Transmit mailbox data byte 3）
位 23：16	TMDT2	0xXX	rw	发送邮箱数据字节 2 （Transmit mailbox data byte 2）
位 15：8	TMDT1	0xXX	rw	发送邮箱数据字节 1 （Transmit mailbox data byte 1）
位 7：0	TMDT0	0xXX	rw	发送邮箱数据字节 0 （Transmit mailbox data byte 0）

20.7.2.4 发送邮箱高字节数据寄存器 （CAN_TMDTHx） （x=0..2）

当邮箱不在空置状态时，该寄存器的所有位为写保护。

域	简称	复位值	类型	功能
位 31：24	TMDT7	0xXX	rw	发送邮箱数据字节 7 （Transmit mailbox data byte 7）
位 23：16	TMDT6	0xXX	rw	发送邮箱数据字节 6 （Transmit mailbox data byte 6） 注：若时间触发通信模式使能，且对应的时间戳的发送使能，则此位将被 MTS[15：8]替代。
位 15：8	TMDT5	0xXX	rw	发送邮箱数据字节 5 （Transmit mailbox data byte 5）
位 7：0	TMDT4	0xXX	rw	发送邮箱数据字节 4 （Transmit mailbox data byte 4）

20.7.2.5 接收FIFO邮箱标识符寄存器 （CAN_RFIx） （x=0..1）

注意： 所有接收邮箱寄存器都是只读的。

域	简称	复位值	类型	功能
位 31：21	RFSID/RFEID	0xXXX	ro	接收 FIFO 的标准标识符或扩展标识符（Receive FIFO standard identifier or receive FIFO extended identifier） 注：这 11 位为标准标识符或扩展标识符的高 11 位。
位 20：3	RFEID	0xXXXXX	ro	接收 FIFO 的扩展标识符 （Receive FIFO extended identifier） 注：这 18 位为扩展标识符的低 18 位。
位 2	RFIDI	0xX	ro	接收 FIFO 的标识符类型指示（Receive FIFO identifier type indication） 0：使用标准标识符； 1：使用扩展标识符。
位 1	RFFRI	0xX	ro	接收 FIFO 的帧类型指示（Receive FIFO frame type indication） 0：数据帧； 1：远程帧。
位 0	保留	0x0	resd	保持默认值。

20.7.2.6 接收FIFO邮箱数据长度和时间戳寄存器 （CAN_RFCx） （x=0..1）

注意： 有接收邮箱寄存器都是只读的。

域	简称	复位值	类型	功能
位 31：16	RFTS	0xXXXX	ro	接收邮箱的报文时间戳 （Receive FIFO time stamp） 注：该时间戳为报文接收帧起始时刻采样到的 CAN 内部定时器的值。

位 15：8	RFFMN	0xXX	ro	过滤器匹配序号 （Receive FIFO filter match number） 注：此处存放的是报文通过的那个过滤器序号。
位 7：4	保留	0xX	resd	保持默认值。
位 3：0	RFDTL	0xX	ro	接收数据长度 （Receive FIFO data length） 注：该域指定了接收报文的数据长度。其中 1 个报文可包含 0 到 8 个字节数据。对于远程帧，数据长度 RFDTL 固定为 0。

20.7.2.7 接收FIFO邮箱低字节数据寄存器 （CAN_RFDTLx）（x=0..1）

注意： 所有接收邮箱寄存器都是只读的。

域	简称	复位值	类型	功能
位 31：24	RFDT3	0xXX	ro	接收 FIFO 数据字节 3 （Receive FIFO data byte 3）
位 23：16	RFDT2	0xXX	ro	接收 FIFO 数据字节 2 （Receive FIFO data byte 2）
位 15：8	RFDT1	0xXX	ro	接收 FIFO 数据字节 1 （Receive FIFO data byte 1）
位 7：0	RFDT0	0xXX	ro	接收 FIFO 数据字节 0 （Receive FIFO data byte 0）

20.7.2.8 接收FIFO邮箱高字节数据寄存器 （CAN_RFDTHx）（x=0..1）

注意： 所有接收邮箱寄存器都是只读的。

域	简称	复位值	类型	功能
位 31：24	RFDT7	0xXX	ro	接收 FIFO 数据字节 7 （Receive FIFO data byte 7）
位 23：16	RFDT6	0xXX	ro	接收 FIFO 数据字节 6 （Receive FIFO data byte 6）
位 15：8	RFDT5	0xXX	ro	接收 FIFO 数据字节 5 （Receive FIFO data byte 5）
位 7：0	RFDT4	0xXX	ro	接收 FIFO 数据字节 4 （Receive FIFO data byte 4）

20.7.3 CAN过滤器寄存器

20.7.3.1 CAN过滤器控制寄存器 （CAN_FCTRL）

注意： 该寄存器的非保留位完全由软件控制。

域	简称	复位值	类型	功能
位 31：1	保留	0x150E0700	resd	保持默认值。
位 0	FCS	0x1	rw	过滤器组配置控制开关 （Filters configure switch） 0：关闭（过滤器组处于工作状态）； 1：开启（过滤器组处于配置状态）。 注：过滤器组的初始化配置必须要在过滤器组工作在配置状态下进行。

20.7.3.2 CAN过滤器模式配置寄存器 （CAN_FMCFG）

注意： 只有在设置 CAN_FCTRL（FCS=1），使过滤器处于配置模式下，才能对该寄存器写入。

域	简称	复位值	类型	功能
位 31：14	保留	0x00000	resd	保持默认值。
位 13：0	FMSELx	0x0000	rw	过滤器组的模式选择 （Filter mode select） 每一位对应于一个过滤器组 0：掩码模式； 1：列表模式。

20.7.3.3 CAN过滤器位宽配置寄存器 （CAN_FBWCFG）

注意： 只有在设置 CAN_FCTRL（FCS=1），使过滤器处于配置模式下，才能对该寄存器写入。

域	简称	复位值	类型	功能
位 31：14	保留	0x00000	resd	保持默认值。
位 13：0	FBWSELx	0x0000	rw	过滤器组的位宽选择（Filter bit width select） 每一位对应于一个过滤器组 0：2 个 16 位；

				1：单个 32 位。

20.7.3.4 CAN过滤器FIFO关联寄存器 （CAN_FRF）

注意： 只有在设置 CAN_FCTRL（FCS=1），使过滤器处于初始化模式下，才能对该寄存器写入。

域	简称	复位值	类型	功能
位 31：14	保留	0x00000	resd	保持默认值。
位 13：0	FRFSELx	0x0000	rw	过滤器组关联 FIFO 选择（Filter relation FIFO select） 每一位对应于一个过滤器组 0：关联 FIFO0； 1：关联 FIFO1。

20.7.3.5 CAN过滤器激活控制寄存器 （CAN_FACFG）

域	简称	复位值	类型	功能
位 31：14	保留	0x00000	resd	保持默认值。
位 13：0	FAENx	0x0000	rw	过滤器组激活使能（Filter active enable） 每一位对应于一个过滤器组 0：关闭； 1：开启。

20.7.3.6 CAN过滤器组i的过滤位寄存器x （CAN_FiFBx） （其中i＝0..13；x=1..2）

注意： 共有 14 组过滤器：i=0..13。每组过滤器由 2 个 32 位的寄存器，CAN_FiFB[2：1]组成。

只有在 CAN 过滤器激活控制寄存器 （CAN_FACFG）相应的 FAENx 位清'0'，或 CAN 过滤器控制寄存器 （CAN_FCTRL）的 FCS 位为'1'时，才能修改相应的过滤器寄存器。

域	简称	复位值	类型	功能
位 31：0	FFDB	0x0000 0000	rw	过滤器过滤数据位 （Filters filter data bit） 列表模式： 寄存器配置值跟总线上接收到的数据对应位的电平完全一致（如果标准帧则忽略扩展帧对应位数值）。 掩码模式： 只有寄存器配置值为'1'的位才跟总线上接收到的数据对应位的电平一致，寄存器配置值为'0'的位不关心。

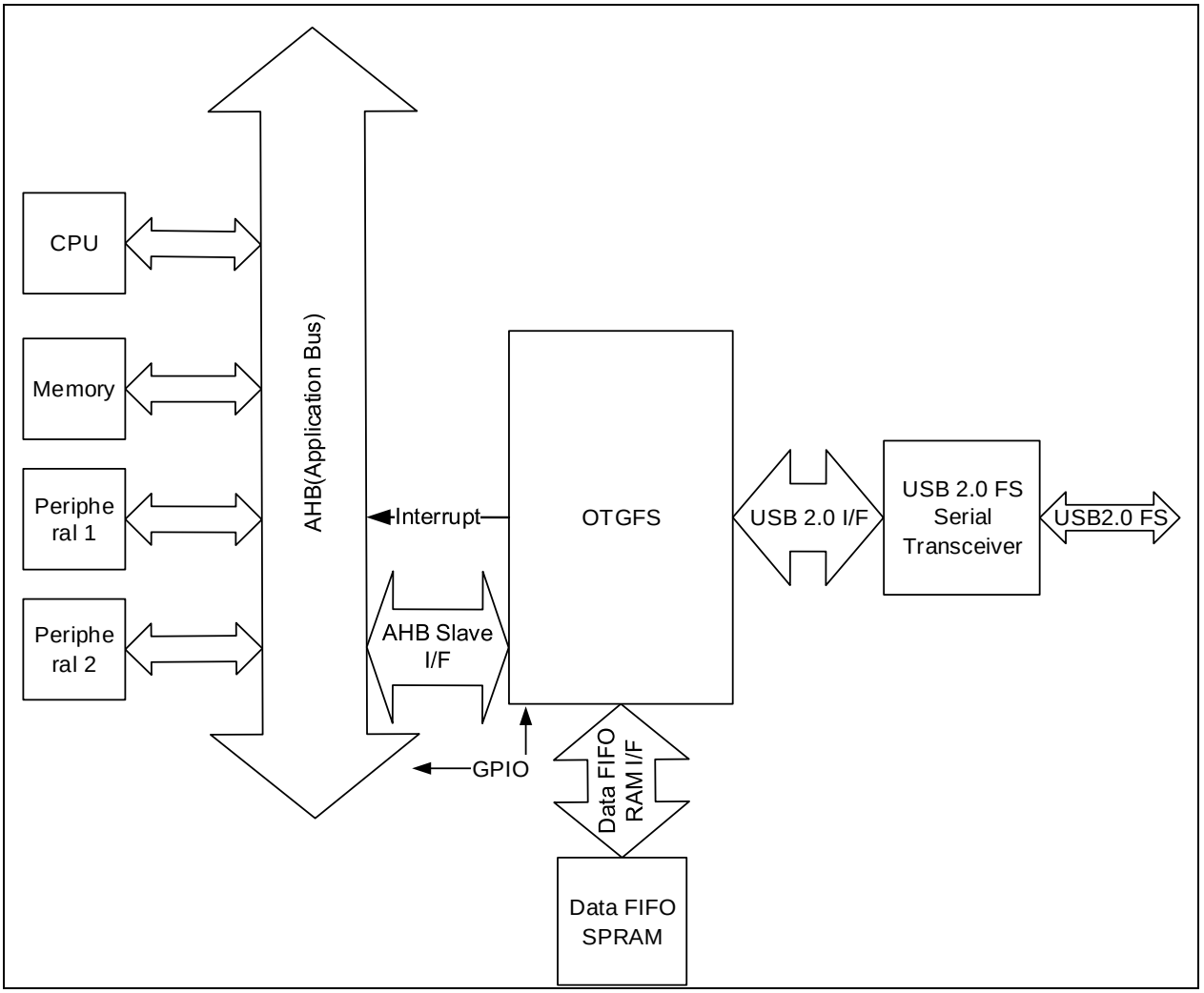
21 USB OTG 全速（OTGFS）

OTGFS 为全速双重角色设备，符合 Universal Serial Bus Specification，Revision2.0 标准。

21.1 OTGFS系统结构框图

下图为 OTGFS 的系统结构框图，OTGFS 模块挂载到 AHB 上，拥有独立的 1280Byte SRAM。

图 21-1 OTGFS的系统结构框图



21.2 OTGFS 功能概述

OTGFS 模块由 OTGFS controller、内置物理层（PHY）以及独立 1280 字节 SRAM 组成。

OTGFS 支持控制传输、大容量、中断和同步传输。

OTGFS 是 USB 全速双角色设备（DRD）控制器，由 ID 线的状态控制 OTGFS 为主机还是设备：当 ID 线浮空时，OTGFS 作为设备，当 ID 线接地时，OTGFS 为主机。OTG PHY 内置了 1.5KΩ上拉电阻和 15KΩ下拉电阻，以满足双角色设备需要。

OTGFS 作为设备时，支持 1 个双向控制端点，7 个 IN 端点、7 个 OUT 端点；做为主机时，支持 16 个主机通道。

OTGFS 支持 SOF 脉冲功能和 OE 脉冲功能：发生 SOF 包时产生 SOF 脉冲，SOF 脉冲可输出到 PIN 脚和定时器 2；OTGFS 输出数据时产生 OE 脉冲，OE 脉冲可输出到 PIN 脚。

OTGFS 支持挂起模式，进入挂起模式后 OTGFS 处于省电模式。

OTGFS 作为设备时，为所有 OUT 端点分配一个的 FIFO 缓存，为每个 IN 端点分配各自分配一个 FIFO 缓存；OTGFS 作为主机时，为所有的接受通道分配了一个统一的接收 FIFO，为所有非周期性发送通道分配一个统一的发送 FIFO，为所有周期性发送通道分配一个统一的发送 FIFO。

OTGFS 支持挂起模式，在时钟门控寄存器(OTGFS_PCGCCTL)的 STOPPCLK 位置位后，当连续 3ms 未收到总线信号时，OTGFS 会进入挂起模式；OTGFS 还可以通过配置 OTGFS 通用控制器配置寄存器(OTGFS_GCCFG)的 LP_MODE 位关闭 PHY 的接收功能，从而达到降低功耗效果。

21.3 OTGFS时钟与管脚配置

21.3.1 OTGFS时钟配置

OTGFS 接口中存在两个时钟：USB 控制时钟和 AHB 总线时钟。USB 全速设备总线速度标准为 12Mb/s ±0.25%。因此需要给 OTGFS 提供 48MHz±0.25%的时钟频率用于 USB 总线采样。

OTGFS 48M 时钟有两种配置来源：

- HICK 48M

使用 HICK 48M 时钟作为 USB 控制时钟时，建议开启 ACC 功能。

- 通过 PLL 分频

注意 PLL 的输出频率要满足 USBDIV（参考时钟配置寄存器（CRM_CFG））能够正常分频到 48MHz。

注意：使用 OTGFS 时，AHB 时钟频率必须大于 30Mhz

21.3.2 OTGFS管脚配置

OTGFS 的输入输出均与 GPIO 复用，当满足以下条件时 GPIO 被用于 OTGFS 功能。

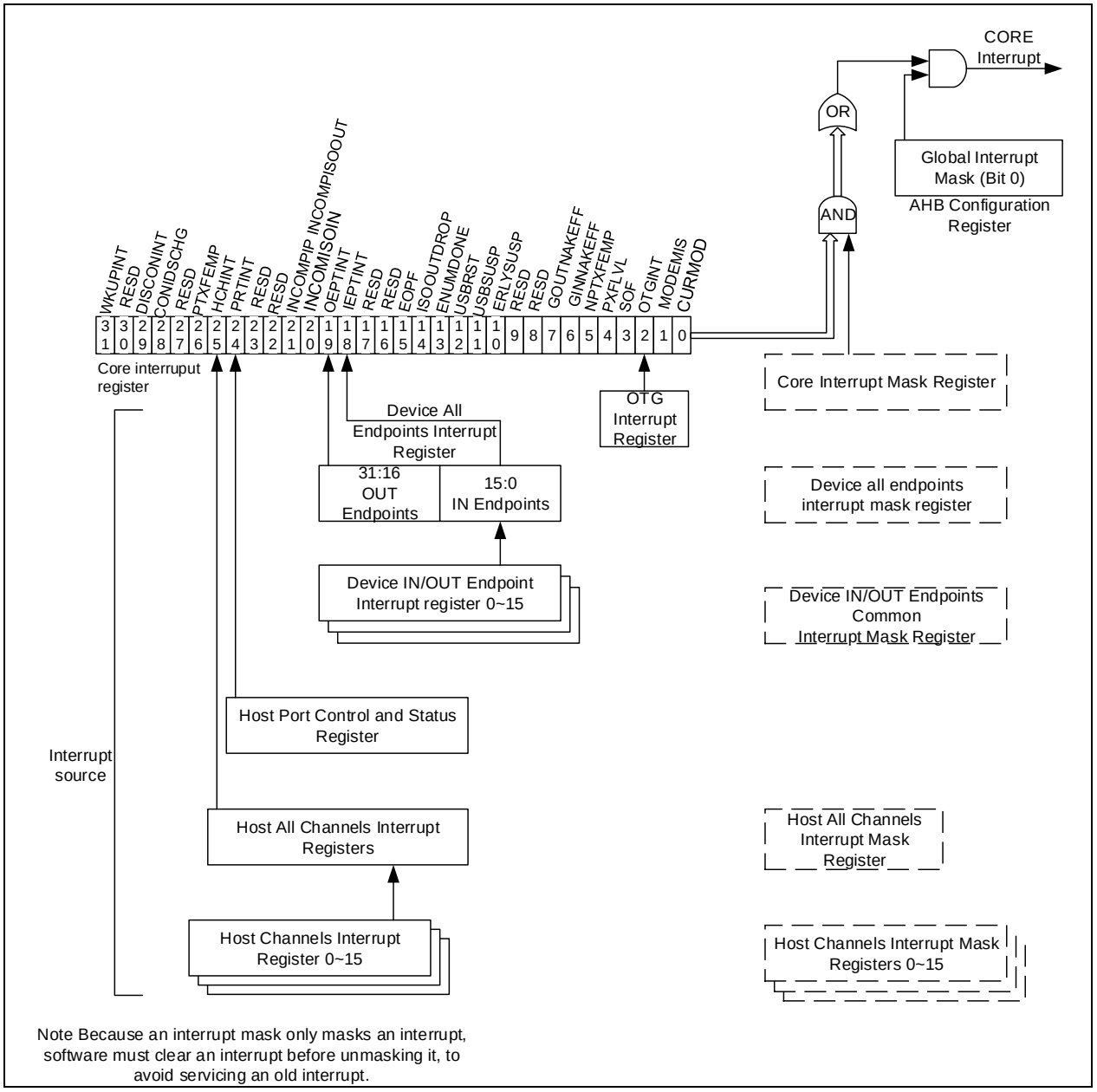
表 21-1 OTGFS输入/输出引脚

信号名称	GPIO	条件
OTGFS_SOF	PA8	在 CRM 中使能 OTG 模块，且 PA8 复用功能寄存器配置为 0xA
OTGFS_VBUS	PA9	配置 PA9 为复用功能模式，且将 PA9 复用功能寄存器配置为 0xA
OTGFS_ID	PA10	在 CRM 中使能 OTG 模块，配置 PA10 为复用功能模式，且将 PA10 复用功能寄存器配置为 0xA
OTGFS_D-	PA11	在 CRM 中使能 OTG 模块，且 PWRDOWN 位设置为 1
OTGFS_D+	PA12	在 CRM 中使能 OTG 模块，且 PWRDOWN 位设置为 1
OTGFS_OE	PA4	在 CRM 中使能 OTG 模块，且 PA4 复用功能寄存器配置为 0xA
	PA13	在 CRM 中使能 OTG 模块，且 PA13 复用功能寄存器配置为 0xA
	PC9	在 CRM 中使能 OTG 模块，且 PC9 复用功能寄存器配置为 0xA

21.4 OTGFS中断

OTGFS 中断结构示意图如下图所示，功能详见 OTGFS 中断寄存器(OTGFS_GINTSTS)和 OTGFS 中断屏蔽寄存器(OTGFS_GINTMSK)。

图 21-2 OTGFS中断结构示意图



21.5 OTGFS 功能操作

21.5.1 OTGFS初始化

如果上电时线缆已连接，那么控制器中断寄存器的当前操作模式位（CURMOD）指示当前工作模式。当连接 A 类插头时，OTGFS 控制器工作在主机模式；当连接 B 类插头时，控制器工作在设备模式。

本节介绍了上电后控制器的初始化操作。无论控制器处于主机模式还是设备模式，应用程序都必须遵循初始化顺序。所有的控制器全局寄存器都须按照控制器的配置进行初始化操作。

1、在全局 AHB 配置寄存器中设置以下位域：

- 全局中断屏蔽位= 0x1
- 非周期性发送 FIFO 空级别

- 周期性发送 FIFO 空级别

2、设置全局中断屏蔽寄存器的下列位域：

- OTGFS_GINTMSK.RXFLVLMSK = 0x0

3、设置 OTGFS_USB 配置寄存器(OTGFS_GUSBCFG)的下列位域：

- 全速超时标准位
- USB 周转时间位

4、软件必须解除 OTGFS 中断屏蔽寄存器(OTGFS_GINTMSK)中的下列位的中断屏蔽：

- OTG 中断屏蔽
- 模式不匹配中断屏蔽

5、软件通过读取当前运行模式寄存器(OTGFS_GINTSTS.CURMOD)位来确定 OTGFS 控制器是处于主机模式还是设备模式。

21.5.2 OTGFS FIFO配置

21.5.2.1 设备模式

在上电复位或 USB 复位时，需要进行动态 FIFO 重新分配。在设备模式下，应用程序在更改 FIFO 数据的 SRAM 分配之前，必须先确报满足下列条件：

- OTGFS_DIEPCTLx/ OTGFS_DOEPCTLx.EPENA = 0x0
- OTGFS_DIEPCTLx/ OTGFS_DOEPCTLx.NAKSTS = 0x1

通过发送 FIFO 编号寄存器（OTGFS_GRSTCTL.TXFNUM）位刷新控制器中的发送 FIFO。关于如何刷新发送 FIFO，详见“刷新控制器发送 FIFO”章节。

在为 FIFO 分配 SRAM 空间时，必须注意以下几点：

（1）接收 FIFO SRAM 分配

- 用于 SETUP 包的 SRAM：必须预留 13 个 WORDs 才能接收控制端点发出的 1 个 SETUP 包，这些位置是预留给 SETUP 写数据的，控制器并不会占用。
- 为全局 OUT NAK 预留 1 个 WORD
- 状态信息与每次收到的包一起写入到 FIFO。因而，必须分配至少一个(包最大长度 /4)+1 这样的空间来接收数据包。如果使能了多个同步端点，那么至少需要分配两个(包最大长度 / 4)+1 这样的空间来接收连续的数据包。一般情况下，建议分配两个(包最大长度 / 4)+1 的空间以确保当前一个数据包正传输给 AHB 时，USB 可以接收到下一个数据包。如果 AHB 延迟较长，必须配置足够的空间来接收多个数据包，以防止丢失同步数据包。
- 传输完成的状态信息，连同每个端点的最后一个数据包，一起被推送到 FIFO。
- 必须为每个端点的禁止状态位预留一个位置。
- 通常建议为每个 OUT 端点配置两个 WORDs。

（2）发送 FIFO SRAM 分配

每个 IN 端点发送 FIFO 所需的最小 SRAM 空间就是该 IN 端点的最大数据包长度。发送 IN 端点 FIFO 分配的空间越多，USB 性能越好，并且还能避开 AHB 线上的延迟。

表 21-2 OTGFS发送FIFO SRAM分配表

FIFO 名称	SRAM 大小
接收 FIFO	rx_fifo_size，包含 setup 包，OUT 端点控制信息以及数据 OUT 包。
发送 FIFO 0	tx_fifo_size[0]
发送 FIFO 1	tx_fifo_size[1]
发送 FIFO 2	tx_fifo_size[2]
……	……
发送 FIFO i	tx_fifo_size[i]

根据以上信息配置下列寄存器：

1. OTGFS 接收 FIFO 长度寄存器(OTGFS_GRXFSIZ)

- OTGFS_GRXFSIZ.RXFDEP = rx_fifo_size；

2. 端点 0 TX FIFO 长度寄存器(OTGFS_DIEPTXF0)

- OTGFS_DIEPTXF0.INEPT0TXDEP = tx_fifo_size[0]；
- OTGFS_DIEPTXF0.INEPT0TXSTADDR = rx_fifo_size；

3. 设备 IN 端点发送 FIFO#1 长度寄存器(OTGFS_DIEPTXF1)

- OTGFS_DIEPTXF1.INEPTXFSTADDR = OTGFS_DIEPTXF0.INEPT0TXSTADDR + tx_fifo_size[0]；

4. 设备 IN 端点发送 FIFO#2 长度寄存器(OTGFS_DIEPTXF2)

- OTGFS_DIEPTXF2.INEPTXFSTADDR = OTGFS_DIEPTXF1.INEPTXFSTADDR + tx_fifo_size[1]；

5. 设备 IN 端点发送 FIFO#i 长度寄存器(OTGFS_DIEPTXFi)

- OTGFS_DIEPTXFi.INEPTXFSTADDR = OTGFS_DIEPTXFi-1.INEPTXFSTADDR +tx_fifo_size[i-1]；

6. 完成 SRAM 分配后必须刷新发送 FIFO 和接收 FIFO，以确保 FIFO 正常运行。

- OTGFS_GRSTCTL.TXFNUM = 0x10
- OTGFS_GRSTCTL.TXFFLSH = 0x1
- OTGFS_GRSTCTL.RXFFLSH = 0x1

应用程序必须要等到 TXFFLSH 位和 RXFFLSH 位清除后才能在控制器上执行其它操作。

21.5.2.2 主机模式

在主机模式下，应用程序在更改 FIFO 数据的 SRAM 分配之前，必须先确认下列信息：

- 所有通道已禁用
- 所有 FIFO 为空

重新分配完 FIFO 数据的 SRAM 后，应用程序必须通过发送 FIFO 编号寄存器（OTGFS_GRSTCTL.TXFNUM）刷新位来刷新控制器中的所有 FIFO。

在完成重新分配后，必须通过刷新来复位 FIFO 中的指针以确保 FIFO 正常运行。关于刷新发送 FIFO 的详细信息，请参考“刷新控制器中的发送 FIFO”。

（1）接收 FIFO SRAM 分配

状态信息与每次接收的数据包一起写入 FIFO。因此，必须分配至少一个(包最大长度 / 4)+2 这样的空间来接收数据包。如果使能了多个同步端点，那么至少需要分配两个(包最大长度 /4)+2 这样的空间来接收背靠背数据包。一般情况下，建议分配两个(包最大长度 /4)+2 的空间以便当前一个数据包正传输给 AHB 时，USB 可以接收到下一个数据包。如果 AHB 延迟较长，必须配置足够的空间来接收多个数据包。

传输完成状态信息和通道中止信息会跟着每个主机通道的最后一个数据包一起推送给 FIFO。为此，必须分配两个 WORDs。

（2）发送 FIFO SRAM 分配

主机非周期性发送 FIFO 所需要的最小 SRAM 空间就是所有非周期性 OUT 通道的最大包长度。发送非周期性 FIFO 的空间越多，USB 性能越好，并且还能避开 AHB 线上的延迟。通常，建议分配两个最大包长度，以便当前一个数据包正传输给 USB 时，AHB 可以接收到下一个数据包。如果 AHB 延迟较长，必须配置足够的空间来缓冲多个数据包。

主机周期性发送 FIFO 所需的最小 SRAM 空间就是所有周期性 OUT 通道的最大包长度。

（3）内部存储空间分配

表 21-3 OTGFS内部存储空间分配表

FIFO Name	Data SRAM Size
接收数据 FIFO	rx_fifo_size
非周期性发送 FIFO	tx_fifo_size[0]
IN 端点发送 FIFO	tx_fifo_size[1]

根据这些信息来配置下列寄存器：

1. OTGFS 接收 FIFO 长度寄存器(OTGFS_GRXFSIZ)

- OTGFS_GRXFSIZ.RXFDEP = rx_fifo_size；

2. OTGFS 非周期性 TX FIFO 长度寄存器(OTGFS_GNPTXFSIZ)

- OTGFS_GNPTXFSIZ.NPTXFDEP = tx_fifo_size[0]；
- OTGFS_GNPTXFSIZ. NPTXFSTADDR = rx_fifo_size；

3. OTGFS 主机周期性发送 FIFO 长度寄存器(OTGFS_HPTXFSIZ)

- OTGFS_HPTXFSIZ.PTXFSIZE = tx_fifo_size[1]；
- OTGFS_HPTXFSIZ.PTXFSTADDR = OTGFS_GNPTXFSIZ.NPTXFSTADDR + tx_fifo_size[0]；

4. 在完成 SRAM 分配之后，必须刷新发送 FIFO 和接收 FIFO 以确保 FIFO 正常运行。

- OTGFS_GRSTCTL.TXFNUM = 0x10

- OTGFS_GRSTCTL.TXFFLSH = 0x1
- OTGFS_GRSTCTL.RXFFLSH = 0x1
- 应用程序必须等到 TXFFLSH 位和 RXFFLSH 位清除之后才能执行其它操作。

21.5.2.3 刷新控制器发送FIFO

应用程序通过 OTGFS_GRSTCTL.TXFFLSH 来刷新控制器中的所有发送 FIFO，流程如下：

- 如果该位已清除，则置设置全局 IN NAK 寄存器（OTGFS_DCTL.SGNPINNAK）为 0x1。NAK 有效中断置位表示控制器未读取 FIFO。
- 等待 OTGFS_GINTSTS.GINNAKEFF = 0x1，表示针对所有 IN 端点的 NAK 设置已生效。
- 轮询 AHB 主机空闲寄存器（OTGFS_GRSTCTL.AHBIDLE）直到其置为 1，AHBIDLE = H 表示控制器没有写入 FIFO。
- 确认 OTGFS_GRSTCTL.TXFFLSH = 0x0。如果置 0，则将你需要刷新的发送 FIFO 编号写入到发送 FIFO 编号寄存器（OTGFS_GRSTCTL.TXFNUM）。
- 设置 OTGFS_GRSTCTL.TXFFLSH = 0x1，然后等待清除。
- 置起清除全局 IN NAK 寄存器（OTGFS_DCTL.CGNPINNAK）位。

21.5.3 OTGFS主机模式

在 OTGFS 作为主机时，控制器内部不能为 VBUS 提供 5V 电压源，需要外部电压泵持续为 VBUS 供电。

21.5.3.1 主机初始化

应用程序必须遵循以下步骤来初始化控制器：

1. 将主机端口中断屏蔽寄存器（OTGFS_GINTMSK.PRTINTMSK）设置为解除中断屏蔽
2. 设置 OTGFS 主机模式配置寄存器(OTGFS_HCFG)，选择全速主机或高速主机模式
3. 设置 OTGFS_HPRT.PRTPWR 位为 0x1，驱动 USB 线上的 VBUS 供电
4. 等待端口连接检测寄存器（OTGFS_HPRT0.PRTCONDET）中断，该中断表示设备连接到端口。
5. 设置端口复位寄存器（OTGFS_HPRT.PRTRST）位为 0x1，发起复位操作。
6. 等待至少 10 ms，确保完成复位。
7. 设置端口复位寄存器（OTGFS_HPRT.PRTRST）位为 0x0
8. 等待端口使能/禁止状态改变寄存器（OTGFS_HPRT.PRTENCHNG）中断
9. 读取端口速度寄存器（OTGFS_HPRT.PRTSPD）位，获取枚举速度。
10. 根据已选择的 PHY 时钟值来配置 HFIR 寄存器。
11. 设置 OTGFS 接收 FIFO 长度寄存器(OTGFS_GRXFSIZ)，选择接收 FIFO 的长度
12. 设置 OTGFS 非周期性 TX FIFO 长度寄存器(OTGFS_GNPTXFSIZ)，选择非周期发送 FIFO 的起始地址和长度
13. 设置 OTGFS 主机周期性发送 FIFO 长度寄存器(OTGFS_HPTXFSIZ)，选择周期性发送 FIFO 的起始地址和和长度

为了与设备通信，应用程序必须按照“OTGFS 通道初始化”的要求使能和初始化至少一个通道。

21.5.3.2 OTGFS 通道初始化

为了与设备通信，应用程序必须使能和初始化至少一个通道。应用程序可遵循下列步骤来使能和初始化通道：

1. 设置 OTGFS 中断屏蔽寄存器(OTGFS_GINTMSK)解除下列中断屏蔽：
   - 用于 OUT 传输的非周期性发送 FIFO 空
   - 用于 OUT 传输的非周期性发送 FIFO 半空
2. 设置 OTGFS 主机所有通道中断屏蔽寄存器(OTGFS_HAINTMSK)解除所选通道的中断屏蔽
3. 设置 OTGFS 主机通道 x 中断屏蔽寄存器(OTGFS_HCINTMSKx)解除主机通道中断寄存器中与传输相关的中断屏蔽
4. 为所选通道的 OTGFS 主机通道 x 传输长度寄存器(OTGFS_HCTSIZx)配置传输总长度（以字节为单位），以及期望接收的包数目，包括短数据包。应用程序必须根据初始数据 PID (用于首个 OUT 传输的 PID，或者期望从首个 IN 传输收到的数据 PID)来配置 PID 位。
5. 设置传输长度，确保该通道的传输长度是最大包长度的倍数。
6. 根据设备端点特性，比如类型，速度，方向等来配置所选通道的 OTGFS 主机通道 x 特性寄存器(OTGFS_HCCHARx)（只有当应用程序准备好传输或接收数据包时，才能通过设置通道使能位为 1 来使

能通道）

21.5.3.3 中止通道

应用程序可以通过将 OTGFS 主机通道 x 特性寄存器(OTGFS_HCCHARx)的通道禁止寄存器（HCCHARx.CHDIS）和通道使能寄存器（OTGFS_HCCHARx.CHENA）位设置为 0x1 来中止通道。这个操作将使能主机刷新已提交的请求（如果有的话）并生成“通道中止”中断。应用程序必须等待通道中止寄存器（OTGFS_HCINTx.CHHLTD）中断生成之后，才能重新为其他传输事务分配通道。主机不会中断已经在 USB 线上启动的传输任务。

应用程序在中止通道前，必须确保非周期性请求队列（当中止非周期性通道时）或者周期性请求队列（当中止周期性通道时）中至少有一个可用空间。当请求队列满额时（在中止通道前），应用程序可以通过将 OTGFS 主机通道 x 特性寄存器(OTGFS_HCCHARx)中的通道禁止寄存器（HCCHARx.CHDIS）位设置为 0x1 以及将通道使能寄存器（OTGFS_HCCHARx.CHENA）位设置为 0 来刷新已提交的请求。

当请求队列里有传输输入时，控制器会触发 RXFLVL 中断。应用程序必须通过读取/弹出 OTGFS 状态读和 POP 寄存器(OTGFS_GRXSTSP)来生成“通道中止”中断。

应用程序应当在发生下列情况时中止通道：

- 在非周期性 IN 传输期间检测到了传输完成寄存器（OTGFS_HCINTx.XFERC）中断；
- 当 IN 或 OUT 通道收到了 STALL 响应已收到中断寄存器（OTGFS_HCINTx.STALL）、传输错误寄存器（OTGFS_HCINTx.XACTERR）、Babble Error 寄存器（OTGFS_HCINTx.BBLERR）、或者数据翻转错误寄存器（OTGFS_HCINTx.DTGLERR）中断信号时。
- 当收到检测到断开事件产生中断寄存器（OTGFS_GINTSTS.DISCONINT）（设备断开）中断事件时。应用程序必须检查端口连接状态寄存器（OTGFS_HPRT.PRTCONSTS）信号，这是因为当设备与主机断开时，端口连接状态寄存器（OTGFS_HPRT.PRTCONSTS）将复位。应用程序必须启动软件复位以确保所有通道已清除。当设备重新连接时，主机必须启动 USB 复位。
- 当应用程序需要在正常传输结束之前中止传输时。

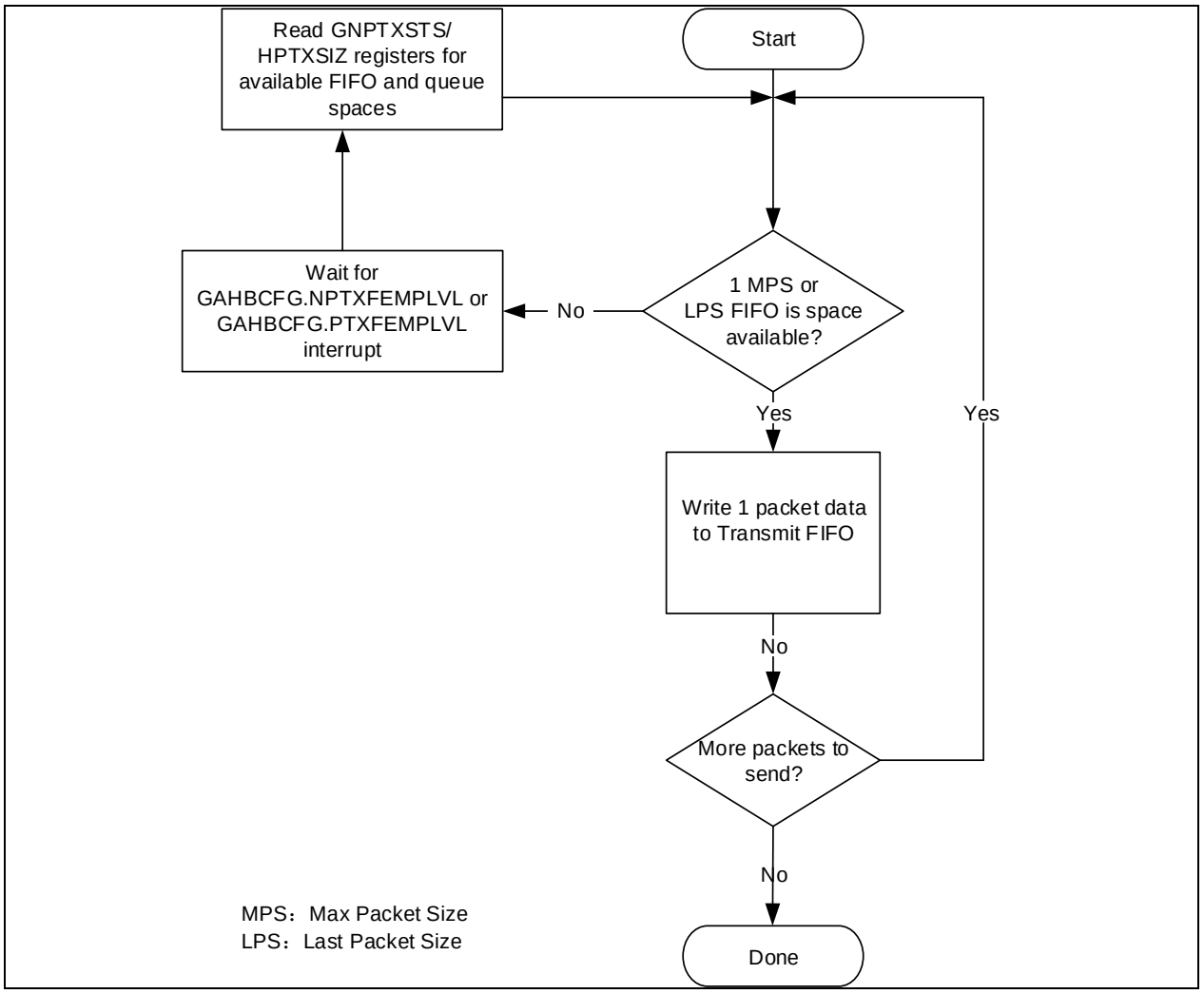
21.5.3.4 选择队列深度

在周期性硬件传输请求队列中支持最多 8 个中断和同步传输请求；在非周期性硬件传输请求队列中支持最多 8 个控制和大容量传输的传输请求。

- 写入发送 FIFO

下图显示写入发送 FIFO 的流程图。OTGFS 主机会在最后一个 WORD 写数据包时，自动将一个条目（OUT 请求）写入周期性/非周期性请求队列。在开始写入 FIFO 之前，应用程序必须保证周期性/非周期性请求队列里至少有一个可用空间。应用程序只能以 WORD 方式写入发送 FIFO。如果包长度没有对齐 WORD，那么应用程序必须填充数据位。OTGFS 主机根据所设置的最大包长度和传输长度来确定实际的包长度。

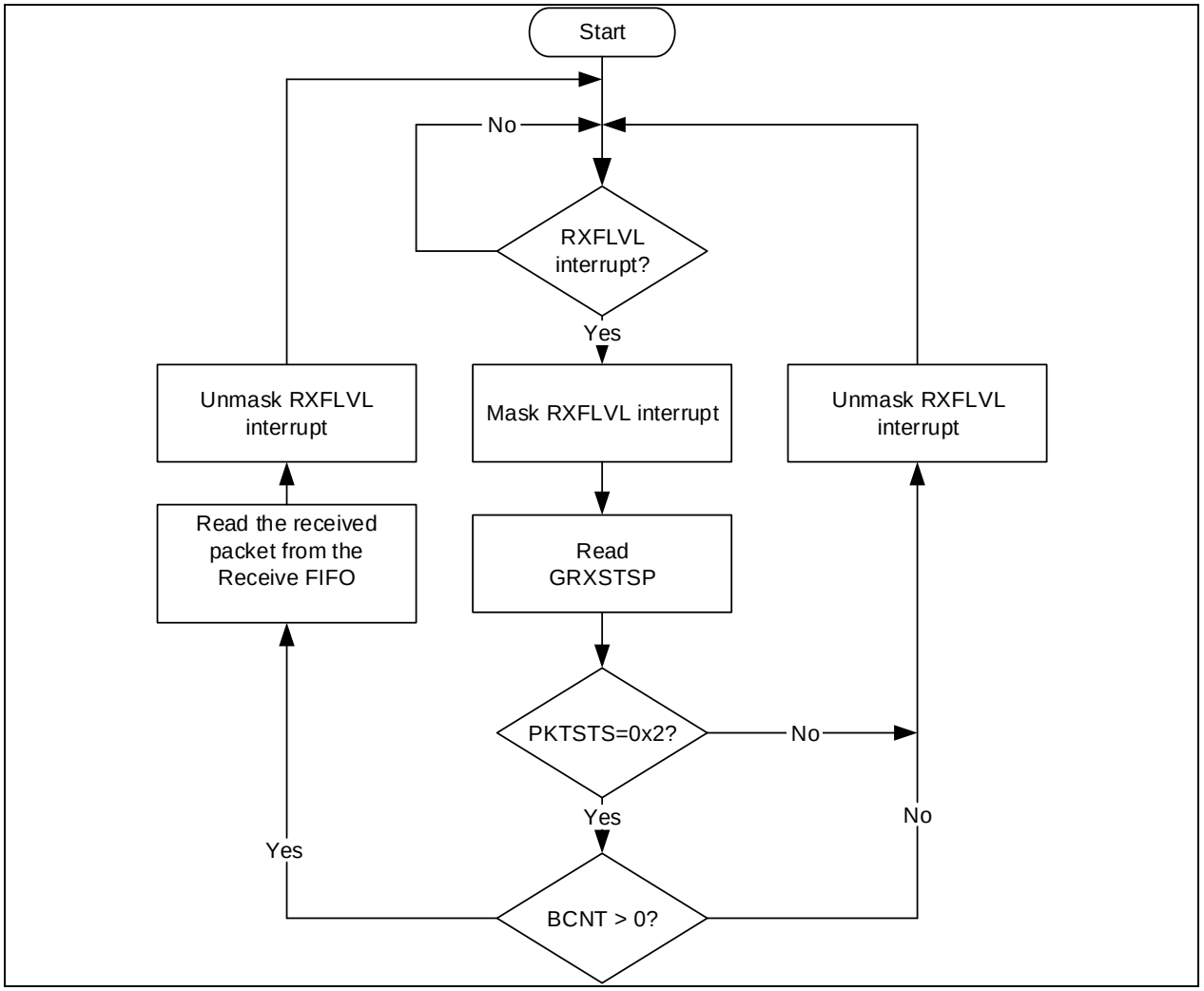
图 21-3写入发送FIFO流程图



- 读取接收 FIFO

下图显示读取接收 FIFO 的流程图。应用程序需要忽略除了 IN 数据包(0x0010)以外的所有包状态。

图 21-4读取接收FIFO流程图



21.5.3.5 特殊情况处理

（1）处理 Babble 状况

OTGFS 处理两种 babble 情况：即包 babble 和端口 babble。如果设备发出的数据超过该通道的最大包长度时将发生包 babble。如果控制器在 EOF2（即帧 2 末尾，非常接近 SOF）时持续接收设备发出的数据时会产生端口 Babble。

当检测到包 babble 时，OTGFS 将停止写入接收缓冲区并等待包结束。当检测到包结束时，OTGFS 将清空已写入接收缓冲区的数据并生成 Babble 中断。

当检测到端口 babble 时，OTGFS 会清空接收 FIFO 并禁用该端口，然后，控制器生成“端口禁用中断”。一旦收到该中断，应用程序需要通过确认端口过流有效位寄存器（HPRT.PRTOVRCACT）信号来确定端口中断并不是因为过流引发的（这是导致端口禁用中断的另一个原因），然后进行软件复位。控制器在检测到端口 babble 信号时不再发送令牌。

（2）处理设备断开

如果设备突然断开，则会生成一个检测到断开事件产生中断寄存器（OTGFS_GINTSTS.DISCONINT）中断。应用程序在收到该中断时，必须通过设置控制器软件复位寄存器（OTGFS_GRSTCTL.CSFTRST）来启动软件复位。

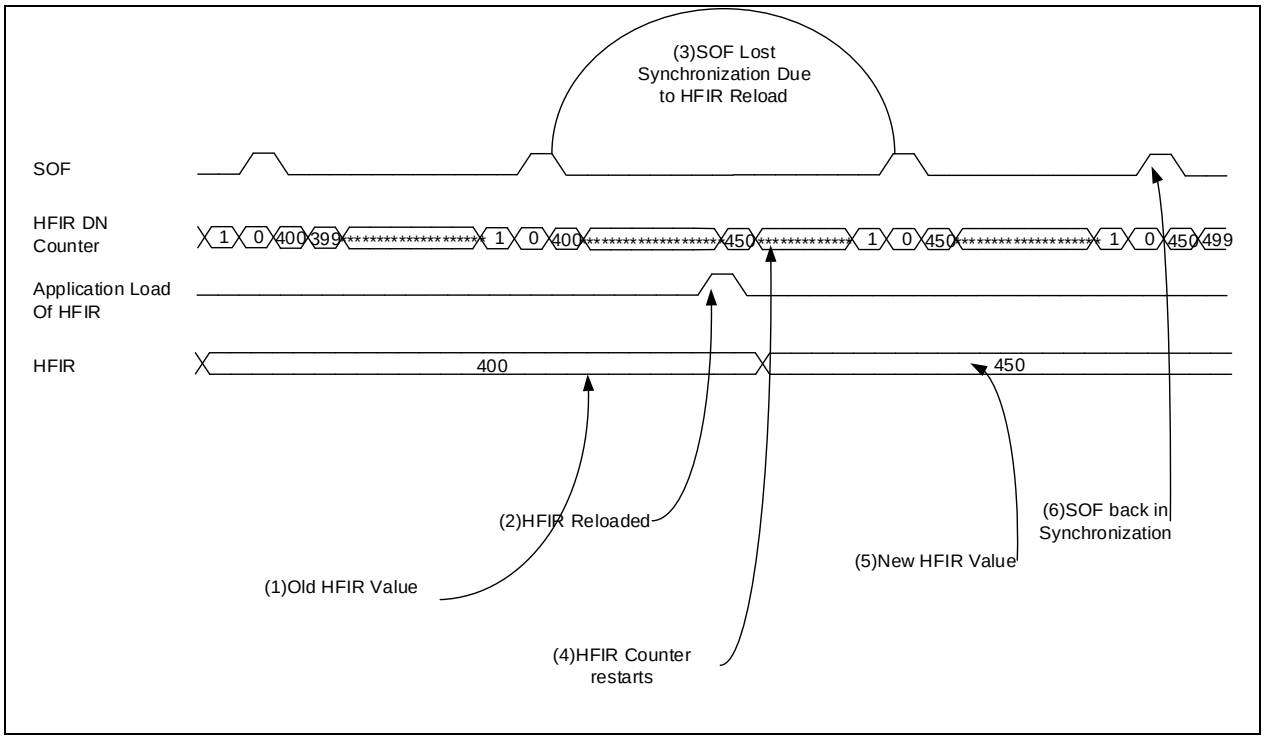
21.5.3.6 主机HFIR功能

主机帧间隔寄存器(HFIR)定义了两个连续 SOF(全速)或者 Keep-Alive 令牌之间的间隔。此位域包含了构成所需帧间隔的 PHY 时钟数，主要用于根据 PHY 时钟频率来调整 SOF 持续时间。

- 将重新加载控制寄存器（OTGFS_HFIR.HFIRRLDCTRL）设置为 0x0 时的 HFIR 行为

本章节使用下图所示波形描述了 OTGFS_HFIR.HFIRRLDCTRL 设置为 0x0 时控制器的行为。

图 21-5 HFIRRLDCTRL为0x0时的HFIR行为



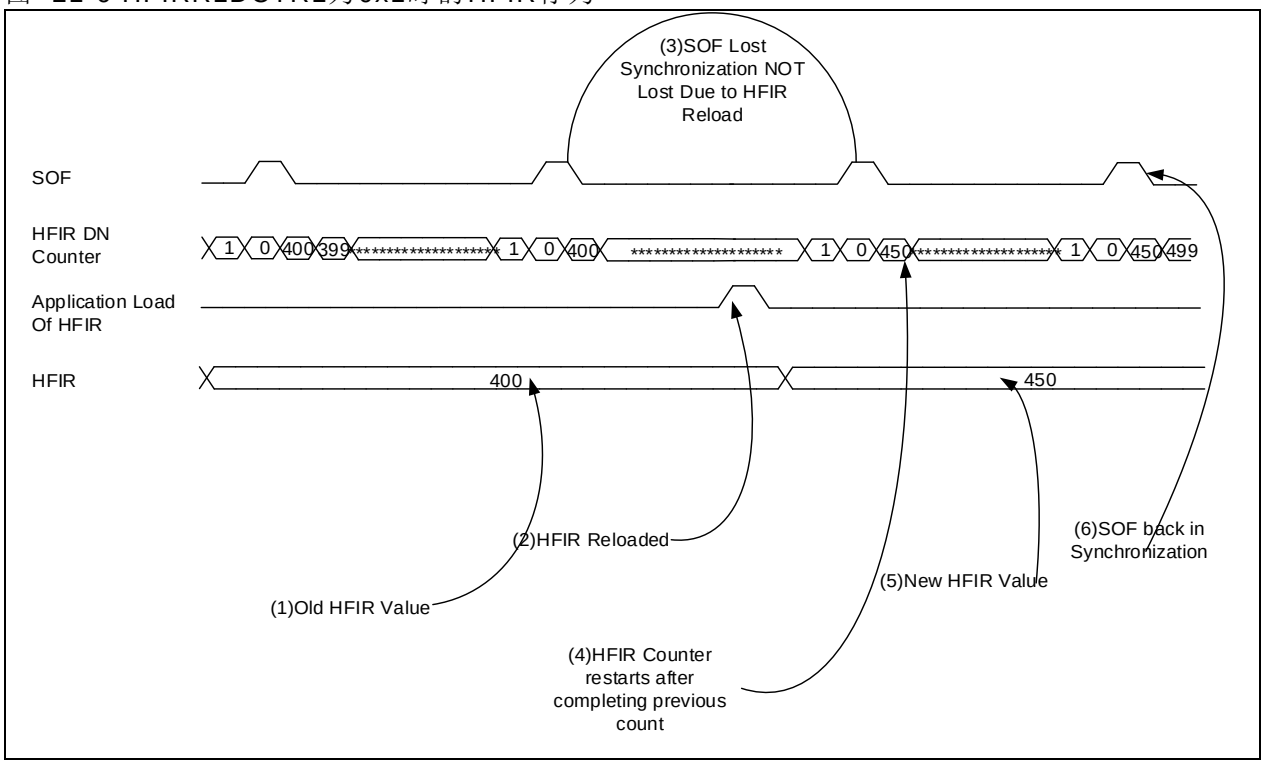
步骤如下所示：

1. 上电复位后，显示应用程序当前设定的 HFIR 值
2. 应用程序加载新值到 HFIR 寄存器
3. 由于 HFIR 向下计数器重新加载，它会立即开始重新计数，从而丢失 SOF 同步
4. 重启 HFIR 计数器
5. HFIR 寄存器接收到新的设定值
6. 使用 HFIR 新功能生成首个 SOF 后，又再次获得 SOF 同步。

● 当重新加载控制寄存器（OTGFS_HFIR.HFIRRLDCTRL）为 0x1 时的 HFIR 行为

本节使用下图所示波形介绍了当 HFIR.HFIRRLDCTRL = 0x1 时控制器的行为。

图 21-6 HFIRRLDCTRL为0x1时的HFIR行为



所示的步骤如下：
1.上电复位后，显示应用程序当前设定的 HFIR 值
2.应用程序加载新的 HFIR 值；HFIR 计数器不采用新值，而是继续计数直到计数器达到 0
3.当计数器在使用旧 HFIR 值计数到 0 时，生成 SOF
4.HFIR 计数器采用新值
5.新的 HFIR 值生效
经过以上步骤 SOF 恢复同步。

21.5.3.7 初始化批量IN传输/控制IN传输

图 21-7 显示典型的批量 IN 传输/控制 IN 传输的操作流程，详见通道 2(ch_2)。假设：

- 应用程序正在尝试接收两个最大长度的数据包（传输长度为 1024 字节）
- 接收 FIFO 包含至少一个最大包长度的数据包以及每个数据包的两个状态 WORD（对于全速传输有 72 字节）
- 非周期性请求队列深度为 4

（1）普通批量 IN 传输和控制 IN 传输的操作流程

图 21-7 所示的操作顺序如下：
1.初始化通道 2（按照“OTGFS 通道初始化”的要求）
2.置起通道使能寄存器（OTGFS_HCCHAR2.CHENA）位，向非周期性请求队列写入一个 IN 请求
3.控制器在完成当前 OUT 传输后会发送一个 IN 令牌
4.一旦将收到的数据包写入接收 FIFO，控制器即生成 RXFLVL 中断。
5.为处理 RXFLVL 中断，需要先屏蔽 RXFLVL 中断，并读取接收数据包状态以确认收到的字节数，然后再读取接收 FIFO。按照这个步骤可以解除 RXFLVL 中断屏蔽。
6.控制器在传输完成状态输入接收 FIFO 后会产生 RXFLVL 中断。
7.应用程序必须读取接收的数据包状态，当接收的数据包不是 IN 数据包时，则不理会它。
8. 控制器在接收到的数据包被读取之后会生成 XFERC 中断。
9. 为处理 XFERC 中断，需要先中止通道（详见“中止通道”），并停止写入 OTGFS 主机通道 2 特性寄存器(OTGFS_HCCHAR2)。控制器会在 OTGFS 主机通道 2 特性寄存器(OTGFS_HCCHAR2)被写入后向非周期性请求队列写入通道中止请求。
10. 控制器在中止状态写入接收 FIFO 后会生成 RXFLVL 中断。
11. 读取接收数据包状态，但不予处理。
12. 一旦从接收 FIFO 中读出中止状态，控制器即会生成 CHHLTD 中断。
13. 如果需要处理 CHHLTD 中断，则需要取消为其它传输分配通道。

（2）处理中断

下列代码描述了批量传输和 IN 传输流程中与通道相关的中断服务程序：

```
Unmask (XACTERR/XFERC/BBLERR/STALL/DATATGLERR)
if (XFERC)
    {
    Reset Error Count
    Unmask CHHLTD
    Disable Channel
    Reset Error Count
    Mask ACK
    }
else if (XACTERR or BBLERR or STALL)
    {
    Unmask CHHLTD
    Disable Channel
    if (XACTERR)
        {
        Increment Error Count
        Unmask ACK
        }
    }
else if (ChHltd)
    {
    Mask CHHLTD
    if (Transfer Done or (Error_count == 3))
```

```
            {
            De-allocate Channel
            }
        else
            {
            Re-initialize Channel
            }
        }
else if (ACK)
        {
        Reset Error Count
        Mask ACK
        }
else if (DATATGLERR)
        {
        Reset Error Count
        }
```

21.5.3.8 初始化批量和控制OUT/SETUP传输

图 21-7 介绍了典型的批量传输或控制传输的 OUT/SETUP 操作流程。详见通道 1 (ch_1)。需要发送 2 个批量传输的 OUT 数据包。控制传输的 SETUP 操作流程也相同，只是只有一个数据包。假设：

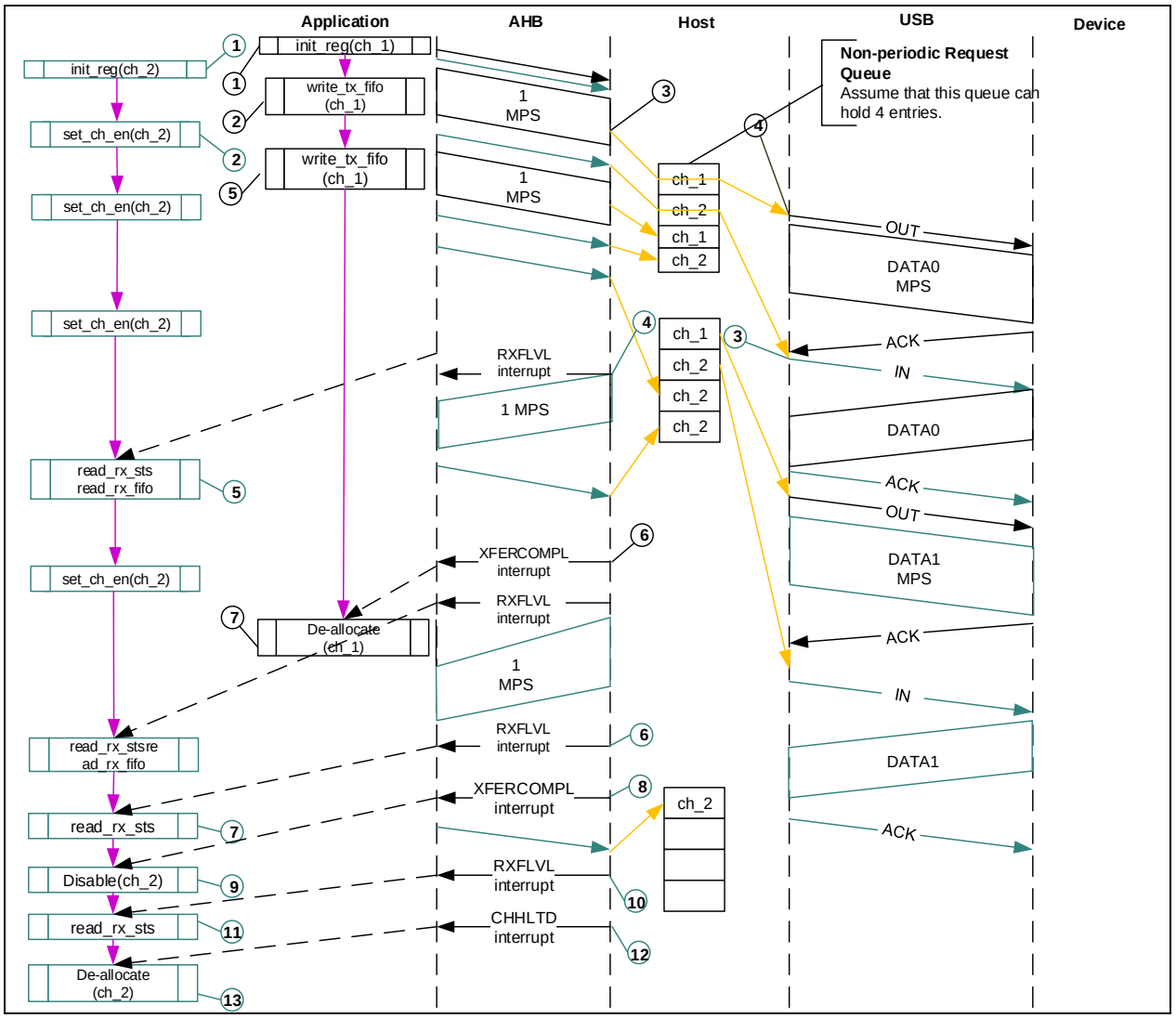
- 应用程序正在尝试发送两个最大包长度的数据包（传输长度为 1024 字节）
- 非周期性发送 FIFO 可以保存 2 个数据包（对于全速传输有 128 字节）
- 非周期性请求队列深度为 4。

（1）普通批量传输和控制传输的 OUT/SETUP 操作流程

图 21-7 所示的操作流程如下：

1. 初始化通道 1（按照“OTGFS 通道初始化”步骤）
2. 写通道 1 的第一个数据包
3. 随着最后一个 WORD 写入，控制器向非周期性请求队列写入一个条目。
4. 一旦非周期性队列变为非空，控制器就会在当前帧帧内发送一个 OUT 令牌。
5. 向通道 1 写入第二个（最后一个）数据包
6. 在前一个传输成功完成后，控制器就会生成 XFERC 中断。
7. 为响应 XFERC 中断，需要取消为其他传输分配通道。

图 21-7普通Bulk/Control OUT/SETUP和Bulk/Control IN传输例程示例图



（2） 处理中断

下列代码给出了批量传输、控制传输 OUT/SETUP 流程中与通道相关的中断服务程序：

```
Unmask (NAK/XACTERR/NYET/STALL/XFERC)
if (XFERC)
      {
      Reset Error Count
      Mask ACK
      De-allocate Channel
      }
else if (STALL)
      {
      Transfer Done = 1
      Unmask CHHLTD
      Disable Channel
      }
else if (NAK or XACTERR or NYET)
      {
      Rewind Buffer Pointers
      Unmask CHHLTD
      Disable Channel
      if (XactErr)
            {
            Increment Error Count
            Unmask ACK
            }
      else
```

```
        {
        Reset Error Count
        }
    }
else if (CHHLTD)
    {
    Mask CHHLTD
    if (Transfer Done or (Error_count == 3))
        {
        De-allocate Channel
        }
    else
        {
        Re-initialize Channel (Do ping protocol for HS)
        }
    }
else if (ACK)
    {
    Reset Error Count
    Mask ACK
    }
```

注意事项：

- 应用程序必须在发送 FIFO 和请求队列里有剩余空间时才能向发送 FIFO 写入数据包。应用程序需要通过非周期性发送 FIFO 空寄存器（OTGFS_GINTSTS.NPTXFEMP）中断来检查发送 FIFO 是否有可用空间。
- 应用程序必须在请求队列有可用空间才能写入请求直到收到 XFERC 中断。

21.5.3.9 初始化中断IN传输

图 21-8 显示了典型的中断 IN 传输操作流程。详见通道 2(ch_2)。假设：

- 应用程序正在尝试从奇数帧开始接收最大一个数据包长度的数据包（传输长度为 1024 字节）
- 接收 FIFO 可以保存至少一个最大数据包长度的包以及每个数据包的两个状态 WORD（对于全速传输有 1031 字节）
- 周期性请求队列深度为 4。

（1）普通的中断 IN 操作流程

图 21-8（通道 2）所描述的操作流程如下：

1. 初始化通道 2（按照“OTGFS 通道初始化”步骤）。应用程序必须设置奇数帧寄存器(OTGFS_HCCHAR2.ODDFRM) 位。
2. 设置通道使能寄存器（OTGFS_HCCHAR2.CHENA）位，向周期性请求队列写入 IN 请求。
3. 每次置起 OTGFS 主机通道 2 特性寄存器(OTGFS_HCCHAR2)的 CHENA 位时，OTGFS 主机都会向周期性请求队列写入一个 IN 请求。
4. OTGFS 主机尝试在下一个（奇数）帧时发送一个 IN 令牌。
5. 一旦收到 IN 数据包，并写入接收 FIFO 后，OTGFS 主机就会生成 RXFLVL 中断。
6. 为了处理 RXFLVL 中断，需要先读取接收的数据包状态以确认接收的字节数，然后再读取接收 FIFO。应用程序必须在读取接收 FIFO 之前先屏蔽 RXFLVL 中断，并在读完整个数据包之后解除中断屏蔽。
7. 控制器在传输完成状态写入接收 FIFO 后会生成 RXFLVL 中断。应用程序必须读取接收包状态并在检测到接收包状态不是 IN 数据包时忽略这个包。
8. 一旦读出接收数据包状态后，控制器即生成 XFERC 中断。
9. 为了处理 XFERC 中断，需要先读取包数目寄存器（OTGFS_HCTSIZ2.PKTCNT 位）。如果 OTGFS_HCTSIZ2.PKTCNT 不为 0，则需要中止该通道，然后重新初始化该通道进行下次传输。如果 OTGFS_HCTSIZ2.PKTCNT == 0，则需要重新初始化该通道进行下一个传输。此时，应用程序必须复位奇数帧寄存器(OTGFS_HCCHAR2.ODDFRM) 位。

（2）处理中断

下列代码描述了中断 IN 传输流程中与通道相关的中断服务程序。

```
Unmask (NAK/XACTERR/XFERC/BBLERR/STALL/FRMOVRUN/DATATGLERR)
if (XFERC)
  {
```

```
    Reset Error Count
    Mask ACK
    if (HCTSIZx.PKTCNT == 0)
      {
      De-allocate Channel
      }
    else
      {
      Transfer Done = 1
      Unmask CHHLTD
      Disable Channel
      }
  }
else if (STALL or FRMOVRUN or NAK or DATATGLERR or BBLERR)
  {
  Mask ACK
  Unmask CHHLTD
  Disable Channel
  if (STALL or BBLERR)
    {
    Reset Error Count
    Transfer Done = 1
    }
  else if (!FRMOVRUN)
    {
    Reset Error Count
    }
  }
else if (XACTERR)
  {
  Increment Error Count
  Unmask ACK
  Unmask CHHLTD
  Disable Channel
  }
else if (CHHLTD)
  {
  Mask CHHLTD
  if (Transfer Done or (Error_count == 3))
    {
    De-allocate Channel
    }
  else Re-initialize Channel (in next b_interval - 1 uF/F)
    }
  }
else if (ACK)
  {
  Reset Error Count
  Mask ACK
  }
```

应用程序必须在请求队列的剩余空间达到 MC 位域指定数目时才能向同一个通道写入请求，然后再切换到其它通道（如果有的话）。

21.5.3.10 初始化中断OUT传输

图 21-8 显示了典型的中断 OUT 操作流程。详见通道 1(ch_1)。假设：

- 应用程序正在尝试从奇数帧开始每个帧发送一个最大包长度的数据包（传输长度为 1024 字节）
- 周期性发送 FIFO 可保存 1 个包（对于全速模式为 1KB）
- 周期性请求队列长度为 4。

（1）普通的中断 OUT 操作

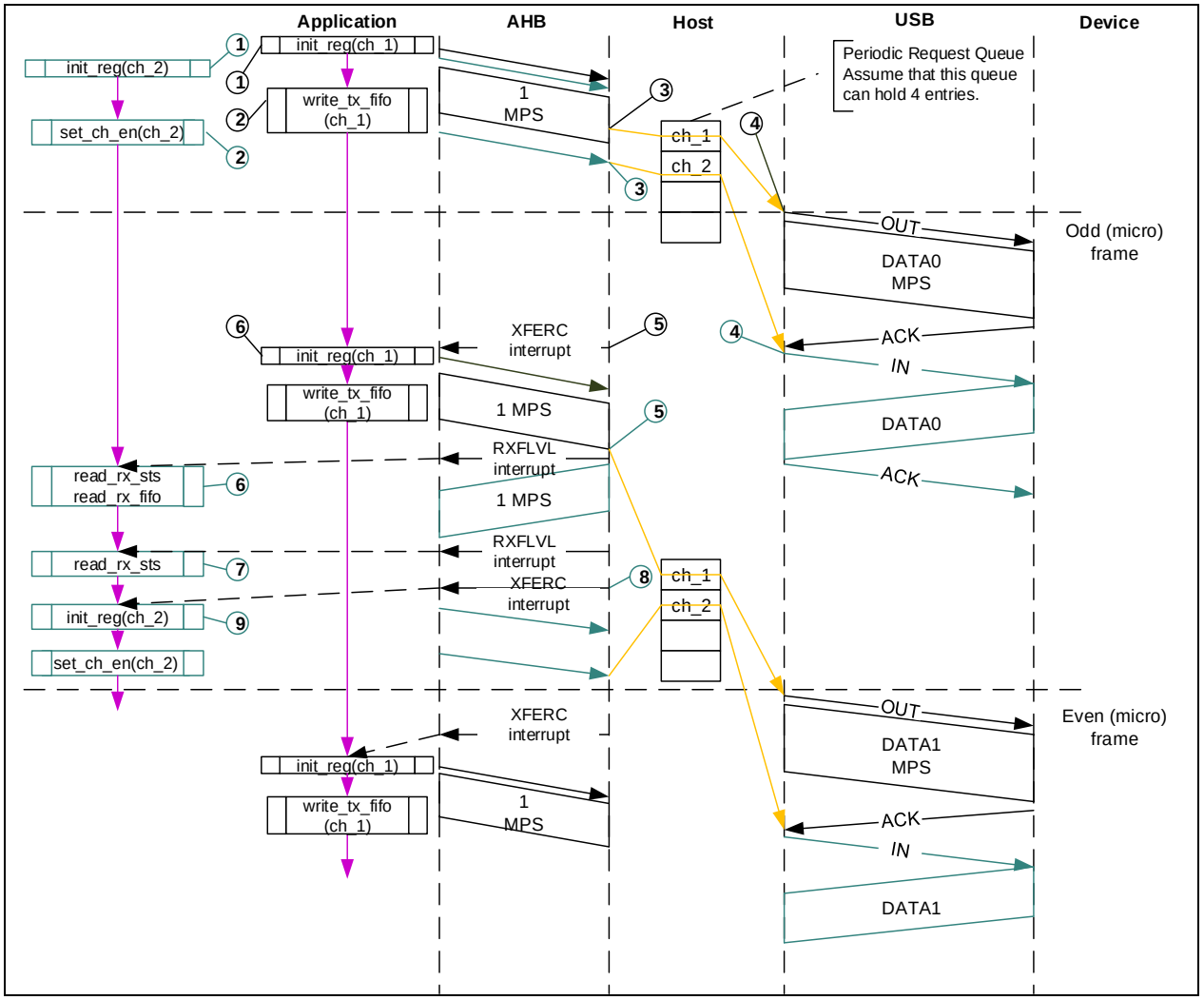
图 21-8（通道 1）所描述的操作流程如下：

1. 参考“OTGFS 通道初始化”使能并初始化通道 1。应用程序必须设置奇数帧寄存器 (OTGFS_HCCHAR1.ODDFRM) 位。
2. 向通道 1 写入第一个数据包。

3. 在写完每个包的最后一个 WORD 时，主机向周期性请求队列写入一个请求
4. 主机会在下一个帧时(奇数帧)时发送 OUT 令牌
5. 在最后一个包发送成功后，主机会生成 XFERC 中断
6. 为响应 XFERC 中断，需要重新初始化该通道进行下次传输。

（2）处理中断

图 21-8普通中断OUT/IN传输示例图



下列代码示例为中断 OUT 传输流程中与通道相关的中断服务程序

```
Unmask (NAK/XACTERR/STALL/XFERC/FRMOVRUN)
if (XFERC)
{
Reset Error Count
Mask ACK
De-allocate Channel
}
else if (STALL or FRMOVRUN)
{
Mask ACK
Unmask CHHLTD
Disable Channel
if (STALL)
{
Transfer Done = 1
}
}
else if (NAK or XACTERR)
{
Rewind Buffer Pointers
Reset Error Count
```

```
Mask ACK
Unmask CHHLTD
Disable Channel
}
else if (CHHLTD)
{
Mask CHHLTD
if (Transfer Done or (Error_count == 3))
{
De-allocate Channel
}
else
{
Re-initialize Channel (in next b_interval - 1 uF/F)
}
}
else if (ACK)
{
Reset Error Count
Mask ACK
}
```

在切换到其它通道之前，应用程序需要根据 MC 位域设定的数目，在发送 FIFO 有可用空间时向发送 FIFO 和请求队列写入数据包。应用程序通过非周期性发送 FIFO 空寄存器（OTGFS_GINTSTS.NPTXFEMP）中断来确认发送 FIFO 是否有可用空间。

21.5.3.11 初始化同步IN传输

图 21-9 显示了典型的同步 IN 传输流程。详见通道 22(ch_2)。假设：

- 应用程序正在尝试从下一个奇数帧开始每个帧发送一个最大数据包长度的数据包（传输长度为 1024 字节）
- 接收 FIFO 可保存至少一个最长包长度的数据包和两个状态 WORD（对于全速传输有 1031 字节）
- 周期性请求队列深度为 4。

（1）普通的同步 IN 传输

图 21-9（通道 2）所描述的操作流程如下：

1. 参考“OTGFS 通道初始化”初始化通道 2。应用程序必须设置奇数帧寄存器(OTGFS_HCCHAR2.ODDFRM) 位。
2. 设置通道使能寄存器（OTGFS_HCCHAR2.CHENA）位向周期性请求队列写入 IN 请求。
3. 每次设置 OTGFS 主机通道 2 特性寄存器(OTGFS_HCCHAR2)的 CHENA 位，主机就会向周期性请求队列写入一个 IN 请求
4. 主机会在下一个奇数帧时发送 IN 令牌
5. 当接收到 IN 数据包并写入接收 FIFO 后，主机会生成 RXFLVL 中断。
6. 为响应 RXFLVL 中断，需要读取接收的数据包状态以确认接收的字节数，然后读取接收 FIFO。应用程序在读取接收 FIFO 之前必须先屏蔽 RXFLVL 中断，并在读取了整个数据包之后解除中断屏蔽。
7. 控制器在传输完成状态输入到接收 FIFO 之后会生成 RXFLVL 中断。此时，应用程序必须读取接收数据包状态，并在检测到接收数据包状态不是 IN 数据包时丢齐这个包。(GRXSTSR.PKTSTS!= 0x0010)。
8. 控制器在读取了接收数据包状态后会生成 XFERC 中断。
9. 为响应 XFERC 中断，需要读取包数目寄存器（OTGFS_HCTSIZ2.PKTCNT）位。如果 OTGFS_HCTSIZ2.PKTCNT 不为 0，然后再重新初始化该通道前中止该通道，然后进行下次传输（如果有的话）。如果 OTGFS_HCTSIZ2.PKTCNT == 0，需重新初始化该通道以进行下一次传输。此时，应用程序必须读取奇数帧寄存器(OTGFS_HCCHAR2.ODDFRM) 位。

（2）处理中断

下列代码示例同步 IN 传输流程中与通道相关的中断服务程序。

```
Unmask (XACTERR/XFERC/FRMOVRUN/BBLERR)
if (XFERC or FRMOVRUN)
   {
   if (XFERC and (HCTSIZx.PKTCNT == 0))
      {
```

```
      Reset Error Count
      De-allocate Channel
      }
   else
      {
      Unmask CHHLTD
      Disable Channel
      }
   }
else if (XACTERR or BBLERR)
   {
   Increment Error Count
   Unmask CHHLTD
   Disable Channel
   }
else if (CHHLTD)
   {
   Mask CHHLTD
   if (Transfer Done or (Error_count == 3))
      {
      De-allocate Channel
      }
   else
      {
      Re-initialize Channel
      }
   }
```

21.5.3.12 初始化同步OUT传输

图 21-9 显示了典型的中步 OUT 传输操作流程。详见通道 1(ch_1)。假设：

- 应用程序正在尝试从下一个奇数帧开始每个帧发送一个最大数据包长度的数据包（传输长度为 1024 字节）
- 周期性发送 FIFO 可保存 1 个数据包（对于全速模式为 1KB）
- 周期性请求队列深度为 4。

（1）普通的同步 OUT 传输操作流程

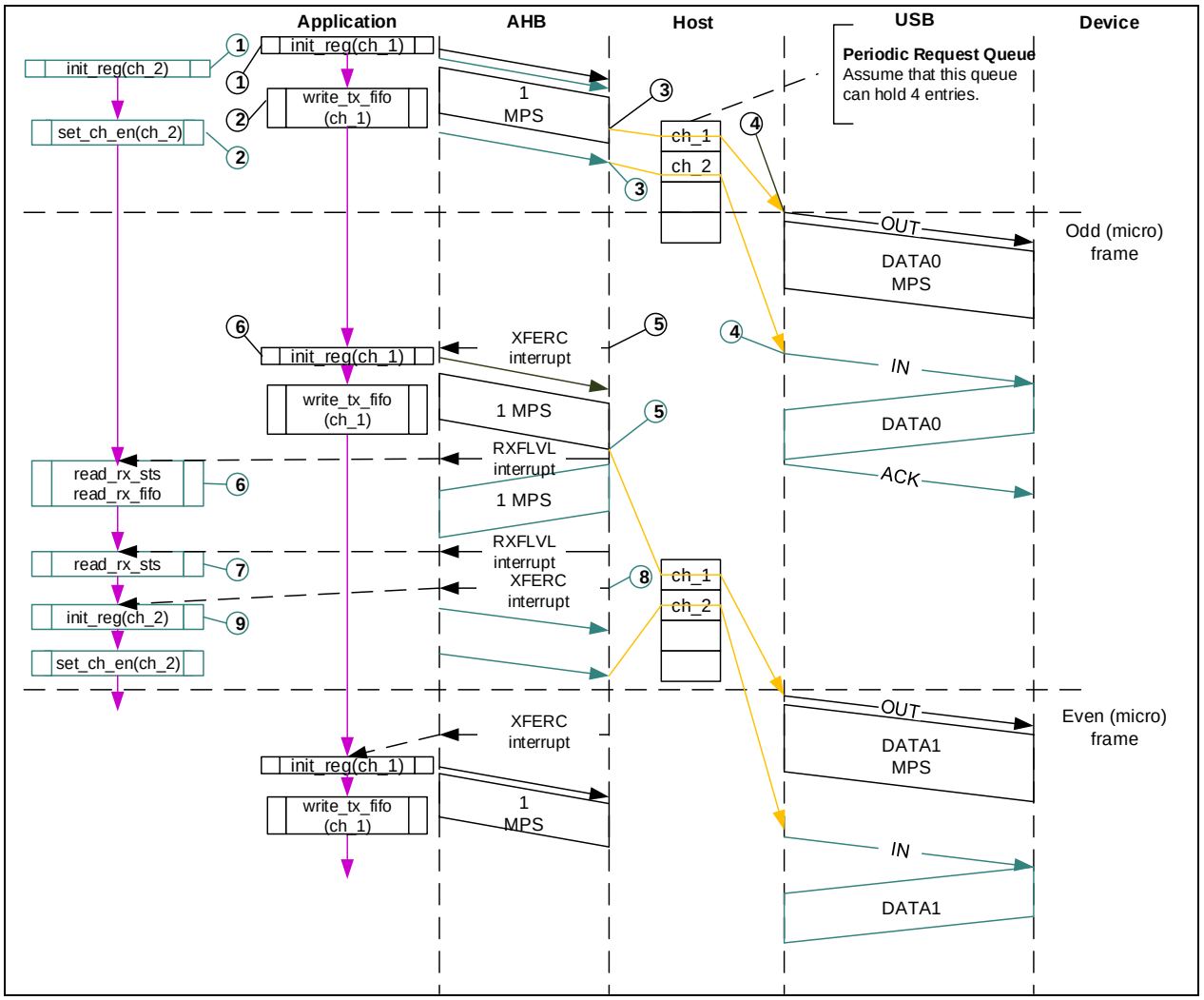
图 21-9（通道 2）所描述的操作流程如下：

1. 参考“OTGFS 通道初始化”初始化通道 1。应用程序必须设置奇数帧寄存器(OTGFS_HCCHAR1.ODDFRM) 位。
2. 向通道 1 写入第一个数据包。
3. 在写完每个包的最后一个 WORD 时，主机向周期性请求队列写入一个请求
4. OTGFS 尝试在下一个帧（奇数帧）发送 OUT 令牌
5. 一旦发送完成最后一个数据包，OTGFS 主机即会生成 XFERC 中断。
6. 为响应 XFERC 中断，需要重新初始化该通道进行下一个传输。

（2）处理中断

图 21-9 显示了同步 OUT 传输操作过程中与通道相关的中断服务程序。

图 21-9普通同步OUT/IN传输示例图



下列示例代码为同步 OUT 传输过程中与通道相关的中断服务程序。

```
Unmask (FRMOVRUN/XFERC)
if (XFERC)
   {
   De-allocate Channel
   }
else if (FRMOVRUN)
   {
   Unmask CHHLTD
   Disable Channel
   }
else if (CHHLTD)
   {
   Mask CHHLTD
   De-allocate Channel
   }
```

21.5.4 OTGFS设备模式

21.5.4.1 设备初始化

在设备开启时，上电时或者从主机模式切换到设备模式时，应用程序需要遵循下列步骤初始化控制器。

1. 配置 OTGFS 设备配置寄存器(OTGFS_DCFG)的相应位

- 设备速度
- 非零长度状态 OUT 握手
- 周期性帧间隔

2. 清除 OTGFS 设备控制寄存器(OTGFS_DCTL.SFTDISCON)位。控制器在清除此位后会启动连接。
3. 设置 OTGFS 中断屏蔽寄存器(OTGFS_GINTMSK)以解除下列中断屏蔽：
- USB 复位
- 枚举完成
- 早期挂起
- USB 挂起
- SOF

4. 等待 OTGFS 中断寄存器(OTGFS_GINTSTS.USBRESET)中断，该中断表示检测到 USB 总线上的复位信号已持续 10ms。应用程序一旦接收此中断，就必须按照“USB 复位时初始化”的步骤操作。
5. 等待 OTGFS 中断寄存器(OTGFS_GINTSTS.ENUMDONE)中断。该中断指示 USB 复位结束。应用程序在接收到此中断后，需读取 OTGFS 设备状态寄存器(OTGFS_DSTS)来确认枚举速度，并按照“枚举完成时初始化”的步骤来操作。此时，设备准备接受 SOF 包，并在控制端点 0 执行控制传输。

21.5.4.2 USB复位时初始化

本节介绍了当检测到 USB 复位信号时应用程序必须执行的操作。
1. 置起所有 OUT 端点的 NAK 位
- OTGFS_DOEPCTLx.SNAK = 0x1(针对所有 OUT 端点)

2. 解除下列位的中断屏蔽
- OTGFS_DAINTMSK.INEP0 = 0x1(控制 IN 端点 0)
- OTGFS_DAINTMSK.OUTEP0 = 0x1(控制 OUT 端点 0)
- OTGFS_DOEPMSK.SETUP = 0x1
- OTGFS_DOEPMSK.XFERC = 0x1
- OTGFS_DIEPMSK.XFERC = 0x1
- OTGFS_DIEPMSK.TIMEOUT = 0x1

3. 为了收发数据，设备必须遵守“设备初始化”流程对寄存器进行初始化
4.为每个 FIFO 分配 SRAM
- 设置 OTGFS 接收 FIFO 长度寄存器(OTGFS_GRXFSIZ)以确保能接收控制 OUT 数据和 SETUP 数据。所分配的 SRAM 至少是控制端点 0 的 1 个最大包长度+2 个 WORD（用于控制 OUT 数据包的状态信息）+10 个 WORD（用于 setup 包）。
- 配置 OTGFS 非周期性 TX FIFO 长度寄存器(OTGFS_GNPTXFSIZ)（具体情况取决于所选择的 FIFO 号），以确保能够发送控制 IN 数据。所分配的 SRAM 至少是控制端点 0 的 1 个最大包长度。

5. 复位“设备配置寄存器”的“设备地址”位。
6. 设置与端点控制寄存器中的下列位域，确保控制 OUT 端点 0 能够接收 SETUP 数据包。
- OTGFS_DOEPTSIZ0.SUPCNT = 0x3(可接收高达 3 个连续的 SETUP 数据包)

此时，用于接收 SETUP 数据包的所有初始化操作就完成了。

21.5.4.3 枚举完成时初始化

本节介绍了当检测到枚举完成中断信号时应用程序必须执行的操作。
- 检测到枚举完成中断信号时，读取 OTGFS 设备状态寄存器(OTGFS_DSTS)来获取枚举速度。
- 配置 OTGFS 设备控制 IN 端点 0 控制寄存器 (OTGFS_DIEPCTL0.MPS)位来设置最大包长度。这个操作是用于配置控制端点 0.控制端点的最大包长度是由枚举速度决定的。
- 解除 SOF 中断屏蔽。

此时，设备准备接收 SOF 包并已经设置好，准备在控制端点 0 执行控制传输。

21.5.4.4 SetAddress指令时初始化

本节介绍了当 SETUP 包收到了 SetAddress 指令时应用程序必须执行的操作。
- 使用 SetAddress 指令收到的设备地址来配置 OTGFS 设备配置寄存器(OTGFS_DCFG)。
- 设置控制器，发送 IN 包。

21.5.4.5 SetConfiguration/SetInterface时初始化

本节介绍了在收到 SetConfiguration / SetInterface 命令时应用程序必须执行的操作。

- 在收到 SetConfiguration 命令时，应用程序需根据新配置定义的有效端点的特性来设置端点寄存器。
- 在收到 SetInterface 命令时，应用程序需针对受到该命令影响的端点配置端点寄存器。
- 有些端点在之前的配置中是有效的，但是在新的配置中可能失效。必须停用这些无效端点。
- 关于如何激活或停用某个端点的，详见“激活端点”以及“USB 端点失效操作”
- 解除每个有效端点的中断屏蔽，并屏蔽 DAINTMSK 寄存器中的所有无效端点的中断
- 为每个 FIFO 配置 SRAM。详见“OTGFS FIFO 配置”。
- 在配置完所有端点后，应用程序需要设置控制器来发送状态 IN 包。

此时，设备控制器已准备好接收和发送任何类型的数据包。

21.5.4.6 激活端点

本节介绍了如何激活一个设备端点或者将现有的设备端点配置为新端点类型。

1. 配置 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx)（对于 IN 或双向端点）或 OTGFS 设备 OUT 端点 x 控制寄存器(OTGFS_DOEPCTLx)（对于 OUT 或双向端点）的以下位：

- 最大包长度
- USB 有效端点位 = 1
- 端点起始数据翻转（指中断和批量类型的端点）
- 端点类型
- 发送 FIFO 号

2. 一旦激活端点，控制器就开始解析发送到该端点的令牌，并针对该端点收到的每一个有效令牌发出一个握手有效信号

21.5.4.7 USB 端点失效操作

本节介绍的是如何使一个现存的端点失效。必须先终止挂起的传输，才能进行端点失效操作。

- 清除 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx)（对于 IN 或双向端点）或者 OTGFS 设备 OUT 端点 x 控制寄存器(OTGFS_DOEPCTLx)（对于 OUT 或双向端点）的“USB 有效端点”位。
- 一旦端点失效，控制器就会忽略发送到该端点的令牌，这将导致 USB 超时。

21.5.4.8 控制写传输(SETUP/Data OUT/Status IN)

本节介绍了控制写传输的操作流程。

应用程序配置流程为：

1. OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.SETUP)包中断置起表示已经向应用程序发送了一个有效 SETUP 包，且数据阶段已启动，详见“OUT 数据传输”。在 SETUP 阶段结束时，应用程序需重新向 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DOEPTSIZx.SUPCNT)位写 3 来接收下一个 SETUP 包。
2. 如果在 SETUP 中断生成之前收到的最后一个 SETUP 包指示数据 OUT 阶段，需按照“非同步 OUT 数据传输”配置控制器来执行控制 OUT 传输。
3. 对于控制端点 0 的单次 OUT 数据传输，应用程序最多可接收 64 字节数据。如果应用程序期望在数据 OUT 阶段接收不止 64 字节的数据，那么必须重新使能该端点另外再接收 64 字节数据，而且需要持续此类操作直到接收完数据阶段的所有数据。
4. 在最后一个 OUT 传输时置起 OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)中断表示该控制传输的数据 OUT 阶段结束。
5. 一旦完成数据 OUT 阶段，应用程序必须执行如下操作：

- 如果需要传输一个新的 SETUP 包，应用程序必须重新使能控制 OUT 端点（参考“OUT 数据传输”）

  OTGFS_DOEPCTLx.EPENA = 0x1
- 为了执行接收到的 SETUP 命令，应用程序必须配置控制器中的相应寄存器。这个是可选操作，视所收到的 SETUP 命令类型而定。

6. 在状态 IN 阶段，应用程序必须按照“非周期性（批量和控制）IN 数据传输”来配置寄存器以执行数据 IN 传输。
7. OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)中断表示控制传输的状态阶段已启动。当接收 FIFO 包状态寄存器的“数据传输完成模式”和“状态阶段开始”位置起时，控制器即会生成该中断。通过设置 OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)来清除“传输完成”中断。重复上述步骤直至检测到该端点上产生 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.XFERC)中断才表示该控制写传输已完成。

21.5.4.9 控制读传输(SETUP/Data IN/Status OUT)

本节介绍的是控制读传输。应用程序操作流程：

- OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.SETUP)包中断表示已向应用程序发送了一个有效的 SETUP 包，而且数据阶段已启动。详见“OUT 数据传输”。应用程序在 SETUP 阶段结束时，必须重新向 OTGFS 设备 OUT 端点 0 传输长度寄存器(OTGFS_DOEPTSIZ0.SUPCNT)位写 3 来接收下一个 SETUP 包。
- 如果在 SETUP 中断生成之前收到的最后一个 SETUP 包指示数据 IN 阶段，需按照“非周期性 IN 数据传输”配置控制器来执行控制 IN 传输。
- 对于控制端点 0 的单次 IN 数据传输，应用程序最多可接收 64 字节数据。如果应用程序期望在数据 IN 阶段发送不止 64 字节的数据，那么必须重新使能该端点另外再发送 64 字节数据，而且需要持续此类操作直到发送完数据阶段的所有数据。
- 重复上述步骤直至检测到该端点上每个 IN 传输都生成了 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.XFERC)中断。
- 在最后一个 IN 传输时置起 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.XFERC)中断表示该控制传输的数据 OUT 阶段结束
- 如需在状态 OUT 阶段执行数据 OUT 传输，应用程序需要按照“OUT 数据传输”来配置控制器。应用程序必须先正确设置 OTGFS 设备配置寄存器(OTGFS_DCFG.NZSTSOUTHSHK)握手位，再发送状态阶段的数据 OUT 传输。
- OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)中断表示该控制传输的状态 OUT 阶段结束，标志着控制读传输成功完成。

21.5.4.10 两个阶段的控制传输(SETUP/Status IN)

本节介绍了两个阶段的控制传输操作。应用程序操作流程：
1. OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.SETUP)包中断表示已向应用程序发送了一个有效的 SETUP 包，而且数据阶段已启动。详见“OUT 数据传输”。应用程序在 SETUP 阶段结束时，必须重新向 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DOEPTSIZx.SUPCNT)位写 3 来接收下一个 SETUP 包。
2. 在 SETUP 中断生成之前对所接收的最后一个 SETUP 包进行解析。如果 SETUP 包指示的两级控制命令，那么应用程序必须执行以下操作：

- 设置 OTGFS_DOEPCTLx.EPENA = 0x1
- 根据所收到的 SETUP 命令类型，应用程序需要配置控制器中的寄存器来执行所收到的 SETUP 命令。

3. 对于状态 IN 阶段，应用程序需按照“非周期性（批量和控制）IN 数据传输”配置寄存器来执行数据 IN 传输。
4. OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.XFERC)中断表示该控制传输的状态 IN 阶段已完成。

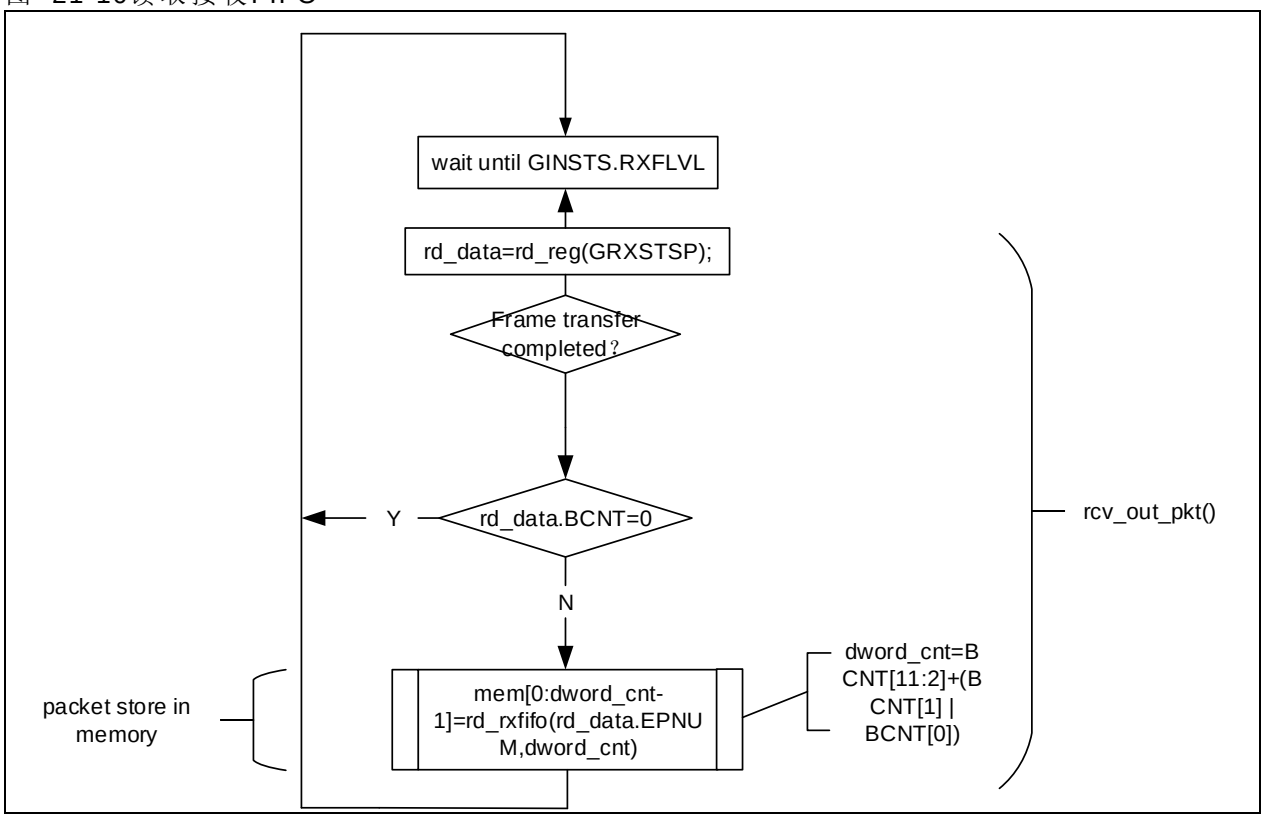
21.5.4.11 读取FIFO包

本节介绍了如何读取接收 FIFO 数据包（OUT 数据和 SETUP 包）
1. 一旦检测到 OTGFS 中断寄存器(OTGFS_GINTSTS.RXFLVL)中断，应用程序必须读取 OTG 状态读和 POP 寄存器(OTGFS_GRXSTSP)
2. 应用程序可以通过设置 OTGFS_GINTMSK.RXFLVL = 0x0 来屏蔽 OTGFS 中断寄存器(OTGFS_GINTSTS.RXFLVL)中断，直到从接收 FIFO 读取到数据包
3. 如果收到的数据包字节数不为 0，那么接收数据 FIFO 会弹出数据字节数并将其储存在存储器。如果所接收的数据包字节数为 0，那么接收数据 FIFO 则不会有数据读出。

4. 接收 FIFO 包状态读数指示发生下列情况：
5. 全局 OUT NAK 模式：PKTSTS = Global OUT NAK，BCNT = 0x000，EPNUM = Dont Care (0x0)，DPID = Dont Care (0x00)，指示全局 OUT NAK 位已生效。
- SETUP 包模式：PKTSTS = SETUP，BCnt = 0x008，EPNUM = Control EP Num，DPID = D0， 表示可以从接收 FIFO 读取某个指定端点的 SETUP 包。
- Setup 阶段完成模式：PKTSTS = Setup Stage Done，BCNT = 0x0，EPNUM = Control EP Num，DPID = Don't Care (0x00)， 表示某个指定端点的 Setup 阶段已结束，并开始进入数据阶段。当接收 FIFO 弹出此条目后，控制器会在指定的控制 OUT 端点上触发 Setup 中断。
- 数据 OUT 包模式：PKTSTS = DataOUT，BCnt =接收的数据 OUT 包的长度(0 ≤ BCNT ≤ 1024)，EPNUM =接收数据包的端点号，DPID =实际的数据 PID。
- 数据传输完成模式：PKTSTS = 数据 OUT 传输完成，BCNT = 0x0，EPNUM =完成数据传输的 OUT 端点号，DPID = Don't Care (0x00)。这些数据表示某个指定的 OUT 端点的 OUT 数据传输已完成。当接收 FIFO 弹出此条目后，控制器会在指定的 OUT 端点上触发传输完成中断。PKTSTS 代码位于本手册的寄存器章节的“接收状态调试读取/状态读和弹出寄存器”

6.接收 FIFO 弹出有效数据后，必须解除 OTGFS 中断寄存器(OTGFS_GINTSTS.RXFLVL)中断屏蔽。
7. 每当应用程序检测到由于 OTGFS 中断寄存器(OTGFS_GINTSTS.RXFLVL)而产生中断线时需要重复第 1-5 步。读取空接收 FIFO 会导致控制器出现意外后果。流程如下图：

图 21-10读取接收FIFO



21.5.4.12 OUT数据传输

本节介绍了数据 OUT 传输和 SETUP 传输过程中的内部数据流及应用程序的操作流程。

（1）执行 Setup 传输

本节介绍了如何处理 SETUP 数据包以及应用程序处理 SETUP 传输的操作流程。上电复位后，应用程序必须遵循“OTGFS 初始化”流程来初始化控制器。应用程序在与主机通信之前，必须按照“设备初始化”流程来初始化端点，并参考“读取 FIFO 包”

【应用程序要求】

1. 控制 OUT 端点的 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DOEPTSIZx.SUPCNT)位必须设置为非零值才能接收 SETUP 包。当应用程序将 SUPCNT 位设置为非零值时，控制器会接收到 SETUP 包并

将其写入接收 FIFO，不受 OTGFS 设备 OUT 端点 x 控制寄存器(OTGFS_DOEPCTLx.NAK)状态位和 OTGFS 设备 OUT 端点 x 控制寄存器(OTGFS_DOEPCTLx.EPENA)位的影响。SUPCNT 位在每次控制端点接收到一个 SETUP 包时会自动递减。如果在接收 SETUP 包之前，SUPCNT 位的设置值不合适，那么虽然控制器仍可以接收 SETUP 包并自动递减 SUPCNT 位，但是应用程序可能就无法确定在控制传输 SETUP 阶段所接收的 SETUP 包有多少个。

- OTGFS_DOEPTSIZx.SUPCNT = 0x3

2. 应用程序必须为接收数据 FIFO 分配一些额外的空间，以确保能够在一个控制端点接收多达 3 个 SETUP 包。

- 预留空间为 13WORDs，其中 4 个 WORDs 空间用于 1 个 SETUP 包，1 个 WORD 空间用于 Setup 阶段，8 个 WORD 空间用于存入控制端点的两个额外的 SETUP 包。
- 每个 SETUP 包需要 4 个 WORD 空间用于存放 8 个字节的 SETUP 数据，4 个字节的传输完成状态以及 4 个字节的 SETUP 状态（SETUP 包模式）。控制器需要为接收数据预留此空间。
- FIFO 仅用于写 SETUP 数据，而不会用于数据包

3. 应用程序需要从接收 FIFO 读取 2 个 WORD 的 SETUP 包。

4. 应用程序必须从接收 FIFO 读取传输完成状态 WORD 并丢弃它。并忽略由于读取而产生的传输完成中断。

【内部数据流】

1.当收到 SETUP 包时，控制器将接收的数据写入接收 FIFO，不需要确认接收 FIFO 是否有可用空间，也不需要检测控制端点的 NAK 和 Stall 位。

- 控制器会在收到 SETUP 包的控制 IN/OUT 端点上置起 IN NAK 和 OUT NAK 位。

2. USB 线上收到每个 SETUP 包后，都会写入 3 个 WORD 数据到接收 FIFO，SUPCNT 位自动递减 1.

- 首个 WORD 包含控制器内部使用的控制信息。
- 第二个 WORD 包含 SETUP 命令的前 4 个字节。
- 第三个 WORD 包含 SETUP 命令的最后 4 个字节。

3. 当从 SETUP 阶段切换到数据 IN/OUT 阶段时，控制器会写入 SETUP 状态完成 WORD 信息到接收 FIFO，指示 SETUP 阶段已结束。

4. 应用程序通过 AHB 总线读取 SETUP 包。

5. 当应用程序弹出了来自接收 FIFO 的状态阶段完成 WORD 信息时，控制器会生成 OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.SETUP)中断来打断应用程序，指示应用程序可以开始处理接收的 SETUP 包。

6. 控制器清除控制 OUT 端点的端点使能位。

【应用程序操作流程】

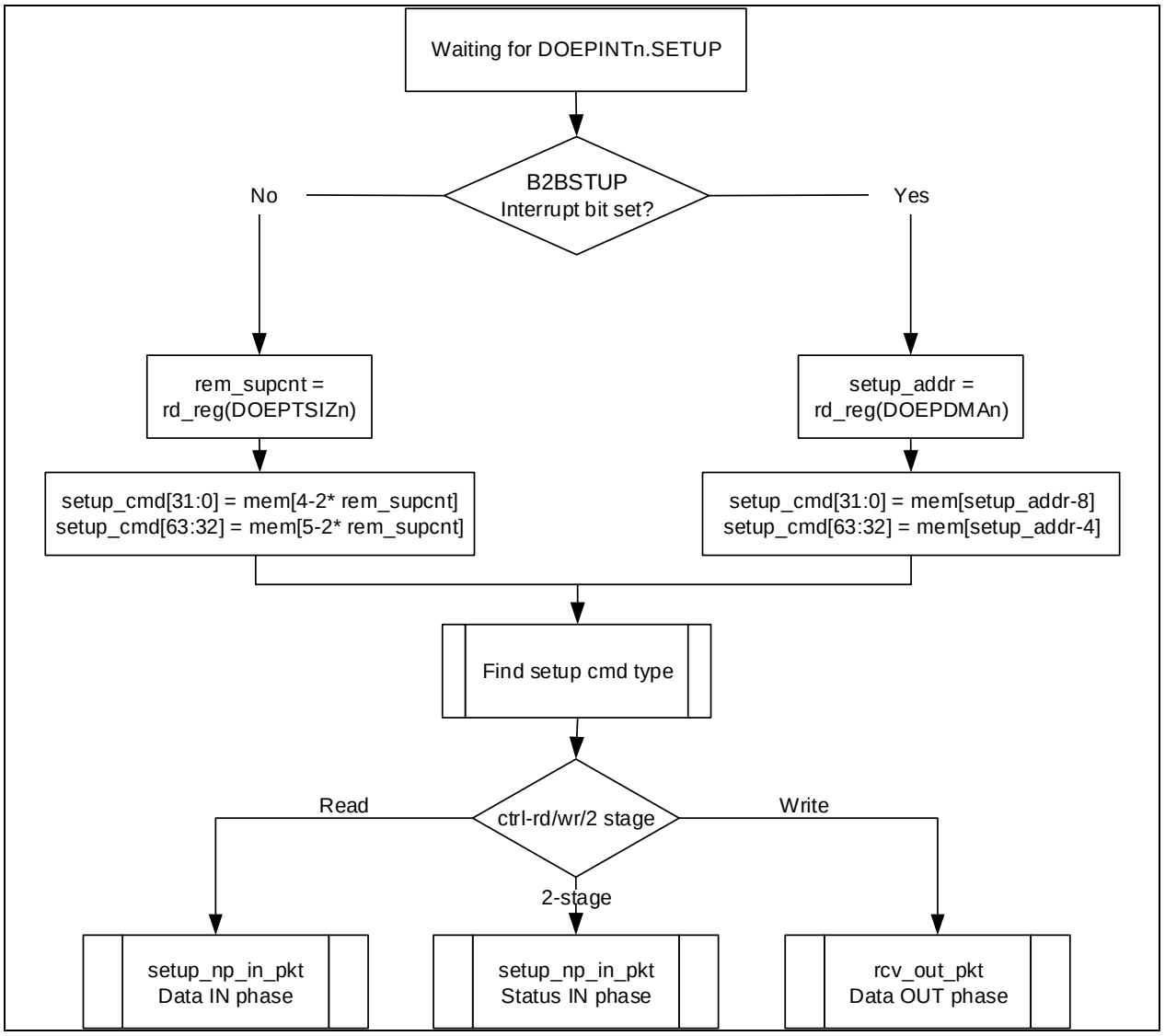
1. 配置 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DOEPTSIZx)。

- OTGFS_DOEPTSIZx.SUPCNT = 0x3

2. 等待 OTGFS 中断寄存器(OTGFS_GINTSTS.RXFLVL)中断并读取刷新接收 FIFO 的数据包（参考“读取 FIFO 包”）。可以多次重复此操作。

3. 宣告 OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.SETUP)中断表示 SETUP 数据传输已成功完成。在收到此中断后，应用程序需要读取 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DOEPTSIZx)来确认收到了多少个 SETUP 包，并处理最后接收到的一个 SETUP 包。

图 21-11 SETUP数据包流程图



（2）处理 3 个以上的连续的 SETUP 包

根据 USB2.0 规范，通常在 SETUP 包出现错误时，主机不会向同一个端点连续发送 3 个以上的 SETUP 包。然而，USB2.0 规范并没有限制主机可以向同一个端点发送连续 SETUP 包的数目。如果发生此类情况，OTGFS 控制器会生成一个中断(OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.B2BSTUP))。

21.5.4.13 IN数据传输

本节介绍了 IN 数据传输期间的内部数据流和应用程序的操作流程。

1. 应用程序可以选择轮询模式或中断模式。

- 如果选择轮询模式，应用程序将通过读取 OTGFS 设备 IN 端点传输 FIFO 状态寄存器(OTGFS_DTXFSTSx)来监测端点发送数据 FIFO 的状态以确认数据 FIFO 内是否有足够的空间可用。
- 如果选择中断模式，应用程序需要等待 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.TXFEMP)中断，然后读取 OTGFS 设备 IN 端点传输 FIFO 状态寄存器(OTGFS_DTXFSTSx)来确认数据 FIFO 内是否有足够的空间可用。
- 如果要写一个单独的非零长度的数据包，数据 FIFO 必须要有足够的空间写入整个数据包。
- 如果要写一个零长度的数据包，应用程序不需要查看 FIFO 空间。

2. 无论使用上述哪种方式，当应用程序确定了有足够的空间可以写入一个发送数据包时，可以先写入端点控制寄存器，再将数据写入数据 FIFO。通常，除了设置端点使能位之外，应用程序必须在 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx)进行读改写设置以防止该寄存器的内容被更改，如果空间足够

大的话，应用程序可以将同一个端点的多个数据包写入发送 FIFO。对于周期性 IN 端点，应用程序只能写入一个帧的数据包。只有在前一个传输发送完成的情况下，才能写入下一个周期性传输。

21.5.4.14 非周期性（批量和控制）IN数据传输

如果要在上电复位后初始化控制器，那么应用程序必须遵循“OTGFS 初始化”的流程。在与主机通讯前，必须先按照“设备初始化”流程对端点进行初始化操作。

【应用程序要求】

1. 对于 IN 传输而言，端点传输长度寄存器中的传输长度位表示的有效数据包含了多个最大包长度的数据包和一个短包。短包在传输结束时被传输 。

- 如果需要传输几个最大包长度的数据包和一个短包：
  传输长度[epnum] = n * mps[epnum] + sp，(其中 n 是 ≥ 0 的整数，且 0 ≤ sp < mps[epnum])
  如果(sp > 0)，那么包数目[epnum] = n + 1。否则，包数目[epnum] = n
- 如果需要传输一个单独的零长度的数据包：
  传输长度[epnum] = 0x0
  包数目[epnum] = 0x1
- 如果要传输几个最大包长度的数据包和一个零长度的数据包（在结束时传输），那么应用程序需要分成两部分进行传输。先发送最大包长度的数据包，再单独发送零长度的数据包。
  首次传输：传输长度[epnum] = n * mps[epnum]；包数目 = n；
  二次传输：传输长度[epnum] = 0x0；包数目= 0x1；

2. 如果使能了端点进行数据传输，则控制器会更新传输长度寄存器。在 IN 传输结束时，以端点禁用中断为结束标志，此时应用程序需要读取传输长度寄存器来确认 USB 线上发送了多少 FIFO 数据。

3.发送 FIFO 获取的数据 = 应用程序设置的首个传输长度 – 控制器更新的最终传输长度

- USB 已发送的数据 = (应用程序设置的首个包数目 – 控制器更新的最终包数目) * mps[epnum]
- USB 待发送的数据 = 应用程序设置的首个传输长度 – USB 已发送的数据

【内部数据流】

1. 应用程序需要设置端点控制寄存器中的传输长度和包数目位，并使能端点来传输数据。
2. 应用程序还需要将所需数据写入该端点的传输 FIFO。
3. 每当应用程序向发送 FIFO 写入一个数据包时，相应端点的传输长度会随着包长度而自动递减。需要持续写入数据直到该端点的传输长度为 0。将数据写入 FIFO 后，“FIFO 中的包数目”会递增（每个 IN 端点发送 FIFO 数据包是 3bit 数目，由内部控制器保存。 对于一个 IN 端点 FIFO，任何时候，控制器能保存的最大包数目为 8）。对于非零长度的数据包，每一个 FIFO 会设置一个单独的标志，FIFO 中不会有数据。
4. 当数据写入发送 FIFO 后，控制器在收到 IN 令牌时会读取该数据。对于每个以 ACK 握手信号结束的非同步 IN 数据包传输，该端点的包数目会自动减 1，直到包数目为 0，包数目不会因为超时而递减。
5. 对于零长度数据包（内部会置起一个零长度标志），控制器会根据 IN 令牌发送零长度数据包，并且包数目位会自动递减。
6. 如果收到了 IN 令牌，但 FIFO 没有数据，且该端点的包数目为 0，那么控制器会生成“当 FIFO 为空时收到了 IN 令牌”的中断，同时不设置该端点的 NAK 位。控制器会以 NAK 握手信号来回应 USB 线上的非同步端点
7. 控制器内部会绕回 FIFO 指针，除了控制 IN 端点外，不会产生超时中断，
8. 当传输长度为 0 并且包数目也为 0 时，会产生传输完成中断，并清除端点使能位。

【应用程序操作流程】

1. 根据传输长度和相应的包数目来设置 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DIEPTSIZx)。
2. 根据端点特性来设置 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx)，并设置 CNAK 和端点使能位。
3.如果发送非零长度的数据包，则应用程序必须轮询 OTGFS 设备 IN 端点传输 FIFO 状态寄存器(OTGFS_DTXFSTSx)（其中 n 是指与该端点相关的 FIFO 编号）来确认数据 FIFO 是否有足够的可用空间。应用程序也可以使用 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.TXFEMP)来确定是否写数据。

21.5.4.15 非同步OUT数据传输

如果要在上电复位后初始化控制器，那么应用程序必须遵循“OTGFS 初始化”的流程。在与主机通讯前，必须先按照“端点初始化”流程对端点进行初始化操作，并参考“读取 FIFO 包”。本章节介绍了常规非同步 OUT 传输操作（控制传输，批量传输或中断传输）

【应用程序要求】

1.对于 OUT 传输，端点传输长度寄存器的传输长度位必须是该端点最大包长度的倍数，并调整到 WORD 边界。

```
if (mps[epnum] mod 4) == 0
transfer size[epnum] = n * (mps[epnum]) //WORD Aligned
else
transfer size[epnum] = n * (mps[epnum] + 4 - (mps[epnum] mod 4)) //Non WORD
Aligned
packet count[epnum] = n
n > 0
```

2. 发生 OUT 端点中断时，应用程序需要读取端点的传输长度寄存器来计算内存中的数据量。所收到的有效数据长度必须小于设定的传输长度。

- 内存中的有效负载=应用程序设置的首个传输长度-控制器更新的最终传输长度
- 收到此有效负载的 USB 数据包数目=应用程序设置的首个包数目—控制器更新的最终包数目

【内部数据流】

1. 应用程序需要设置端点控制寄存器的传输长度和包数目位，清除 NAK 位，并使能接收数据的端点
2. 一旦清除 NAK 位，只要接收 FIFO 里有可用空间，控制器就开始接收数据并写入接收 FIFO。对于 USB 线收到的每个数据包，需要将数据包和其状态写入接收 FIFO。每次向接收 FIFO 中写入数据包（最大包长度或短包），包数目位会自减 1。
- 接收 FIFO 将自动清空所收到的含有 Bad Data CRC 的 OUT 数据包。
- 在 USB 发出数据包 ACK 信号后，控制器不会理会主机（因为未检测到 ACK）重新发送的非同步 OUT 数据包。应用程序没有检测到具有相同数据 PID 的同一个端点上的多个连续 OUT 数据包，此时，包数目不会自动递减。
- 如果接收 FIFO 里没有空间可用，同步或非同步数据包会被忽略，也不会写入接收 FIFO。另外，非同步 OUT 令牌还会收到一个 NAK 握手信号回应。
- 以上 3 种情况下，包数目不会递减，因为没有数据写入接收 FIFO。

3. 当包数目变为 0 或者当端点收到短包时，该端点会置起 NAK 位。一旦设置 NAK 位，同步或非同步数据包会被忽略，也不会写入接收 FIFO，而且非同步 OUT 令牌还会收到 NAK 握手信号回应。
4. 将数据写入接收 FIFO 后，应用程序会读取接收 FIFO 中的数据并写入外部存储器，每个端点 1 次 1 个包。
5. 在完成了数据包写入外部存储器之后，该端点的传输长度会随着写入包的长度减少而减少。
6. 在发生下列情况时，OUT 端点的 OUT 数据传输完成模式会写入接收 FIFO。
- 传输长度和包数目均为 0。
- 写入接收 FIFO 的最后一个 OUT 数据包是一个短包（0 ≤数据包长度<最大包长度）

7. 当应用程序弹出该条目（即 OUT 数据传输完成），即会产生“传输完成中断”，且清除该端点使能位。

【Application Programming Sequence】应用程序操作流程

1. 根据传输长度和相应的包数目来设置 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DOEPTSIZx)。
2. 根据端点特性设置 OTGFS 设备 OUT 端点 x 控制寄存器(OTGFS_DOEPCTLx)，并设置端点使能位和 ClearNAK 位。
- OTGFS_DOEPCTLx.EPENA = 0x1
- OTG-DOEPCTLx.CNAK = 0x1

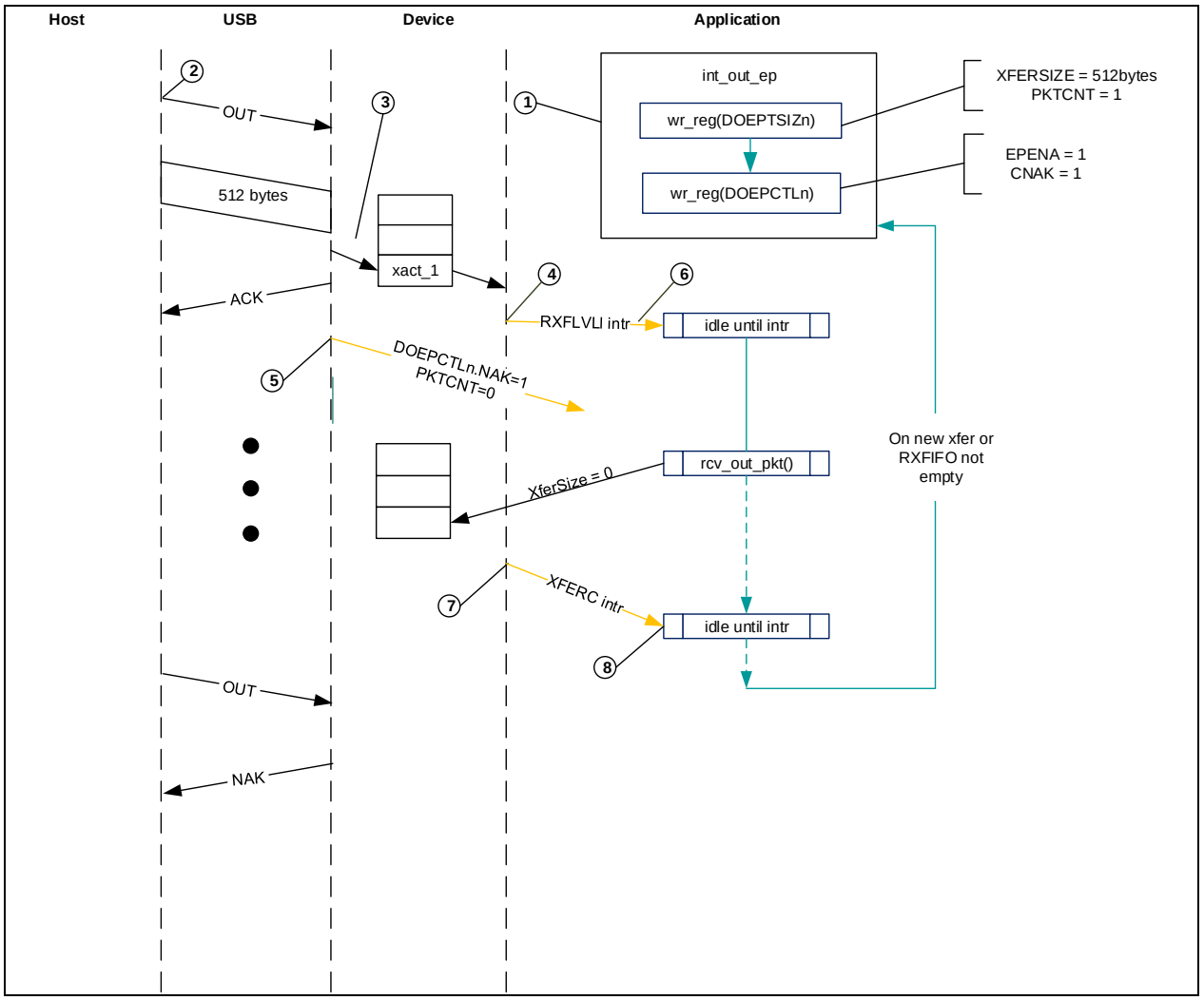
3. 等待 OTGFS 中断寄存器(OTGFS_GINTSTS.RXFLVL)中断，按照“读取 FIFO 包”的流程来读取接收 FIFO 中的所有数据包。
- 根据传输长度可重复此步骤

4. 置起 OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)中断即表示成功完成了非同步 OUT 数据传输。读取 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DOEPTSIZx)来确认收到了多少数据。

【批量 OUT 传输】

下图描绘了从 USB 到 AHB 收到一个单独的批量 OUT 数据包，以及该过程所涉及的相关事件。

图 21-12 BULK OUT传输示例图



在收到 SetConfiguration/SetInterface 命令后，应用程序将通过设置 OYG_DOEPCTLx.CNAK = 0x1 以及 OTGFS_DOEPCTLx.EPENA = 0x1 来初始化所有的 OUT 端点，并设置 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DOEPTSIZx)中的 XFERSIZE 和 PKTCNT 位。

1. 主机尝试给端点发送数据（OUT 令牌）
2. 当控制器收到 USB 发现的 OUT 令牌时，会将数据存放在接收 FIFO，因为接收 FIFO 里有可用空间。
3. 将完整的数据写入接收 FIFO 之后，控制器会解发 OTGFS 中断寄存器(OTGFS_GINTSTS.RXFLVL)中断。
4. 在收到 USB 包的包数目后，控制器内部会通过设置该端点的 NAK 位以防止收到更多的数据包。
5. 应用程序处理该中断并从接收 FIFO 读取该数据。
6. 当应用程序读取完所有数据（相当于 XFERSIZE）之后，控制器会生成 OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)中断。
7. 应用程序处理该中断，并使用 OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)中位后来判断已完成预期的传输。

21.5.4.16 同步OUT数据传输

如果需要在上电复位后初始化控制器，应用程序需遵循“OTGFS 初始化”的流程操作。在与主机通讯之前，需要按照“端点初始化”流程对端点进行初始化操作，并参考“读取 FIFO 包”。

本节介绍的是常规同步 OUT 数据传输。

【应用要求】

1. 与非同步 OUT 数据传输的应用程序要求相同。
2. 对于同步 OUT 数据传输，传输长度和包数目位必须设置为一个帧所能接收的最大包长度，不能超过这个数。同步 OUT 数据传输不能跨越超过 1 个帧。

- 1 ≤ 包数目[epnum] ≤ 3

3. 如果设备支持同步 OUT 端点，则应用程序必须在周期性帧结束之前（OTGFS 中断寄存器(OTGFS_GINTSTS.EOPF)中断）读取接收 FIFO 中的所有同步 OUT 数据包。
4. 如果要在下一个帧接收数据，必须在产生 OTGFS 中断寄存器(OTGFS_GINTSTS.EOPF)中断以及开始 OTGFS 中断寄存器(OTGFS_GINTSTS.SOF)信号之前使能同步 OUT 端点。

【内部数据流】

1. 同步 OUT 端点的内部数据流与非同步 OUT 端点是一样的，只有细微区别。
2. 如果通过设置端点使能位和清除 NAK 位使能了同步 OUT 端点，则奇/偶帧位也要进行适当设置。只有在满足下列条件时，控制器才能在某个帧的同步 OUT 端点接收数据。

- OTGFS 设备 OUT 端点 x 控制寄存器(OTGFS_DOEPCTLx.Even)/Odd microframe = OTGFS_DSTS.SOFFN[0]

3. 当应用程序完全读取了接收 FIFO 的同步 OUT 数据包（数据和状态）时，控制器会根据从接收 FIFO 读取的最后一个同步 OUT 数据包的数据 PID 来更新 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DOEPTSIZx.RXDPID)位。

【应用程序操作流程】

1. 设置 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DOEPTSIZx)的传输长度和相应包数目。
2. 根据端点特性设置 OTGFS 设备 OUT 端点 x 控制寄存器(OTGFS_DOEPCTLx)，并设置端点使能位，ClearNAK 和偶/奇帧位

- 端点使能 = 0x1
- CNAK = 0x1
- 偶/奇帧= (0x0：偶；0x1：奇)

3. 等待 OTGFS 中断寄存器(OTGFS_GINTSTS.RXFLVL)中断，并读取接收 FIFO 中的所有数据包，详见“读取 FIFO 包”

- 可以根据传输长度重复此动作。

4. OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)中断表示已完成同步 OUT 数据传输。但是此中断并不一定意味着存储器的数据都是好的。
5. 同步 OUT 传输并不一定能检测到该中断信号，但是应用程序可以检测到 OTGFS 中断寄存器(OTGFS_GINTSTS.INCOMPISOOUT)同步 OUT 数据中断，详见“不完整的同步 OUT 数据传输”。
6. 读取 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DOEPTSIZx)来确认所收到的传输长度，并判断帧内所收到的数据是否有效。只有在满足了下列条件时，应用程序才会将存储器收到的数据视为有效数据。

- OTGFS_DOEPTSIZx.RxDPID = 0xD0 且收到有效数据的 USB 包数目=0x1
- OTGFS_DOEPTSIZx.RxDPID = 0xD1 且收到有效数据的 USB 包数目=0x2
- OTGFS_DOEPTSIZx.RxDPID = 0xD2 且收到有效数据的 USB 包数目=0x3

收到有效数据的 USB 包数 = APP 设置的初始包数目 – 控制器更新的最终包数目

应用程序不理会无效数据包。

21.5.4.17 使能同步端点

主机将设置接口控制命令发送给设备后，会使能同步端点。随后，主机就可以发送任意帧内的首个同步 IN 令牌，然后再按照 BInterval 的顺序发送。

但是，在 OTGFS 控制器中，同步支持是基于单个传输级。应用程序必须根据每个帧重新配置控制器。OTGFS 控制会使能将要发生传输的帧之前的那个帧的同步端点。

例如，如果数据要在帧 n 上发送，那么必须使能帧 n-1 的端点。另外，OTGFS 通过设置偶/奇帧位来调度同步传输。

【同步 IN 传输中断】

需要处理好下列中断以确保成功调度同步传输。

- OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.XFERC) (基于端点)
- OTGFS 中断寄存器(OTGFS_GINTSTS.INCOMPISOIN) (全局中断)

【处理同步 IN 传输】

需要按照下列步骤来处理同步 IN 传输：

1. 通过设置 OTGFS 中断屏蔽寄存器(OTGFS_GINTMSK.INCOMISOINMSK)解除 OTGFS 中断寄存器(OTGFS_GINTSTS.incompISOOUT)中断
2. 通过设置 OTGFS 设备 OTGFSIN 端点通用中断屏蔽寄存器(OTGFS_DIEPMSK.XFERCMSK)解除 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.XFERC)中断

3. 通过执行下列操作来使能同步端点：

- 配置 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DIEPTSIZx)

OTGFS_DIEPTSIZx.XFERSIZE= n * OTGFS_DIEPCTLx.MPS + sp。其中 0 <= n <= 3，0 <= sp <OTGFS_DIEPCTLx.MPS 。当帧内传输的数据长度小于 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx.MPS)，n=0。 当帧内传输的数据长度是 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx.MPS)的倍数时，sp=0。

OTGFS_DIEPTSIZx.PKTCNT = 1。

OTGFS 设备端点 x 传输长度寄存器(OTGFS_DIEPTSIZx.MC)的设置值与 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DIEPTSIZx.PKTCNT)一样。

- 配置 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx)

读取 OTGFS 设备状态寄存器(OTGFS_DSTS)来确定当前的帧号

配置 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx.MPS)为最大包长度

配置 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx.USBACTEP)为 0x1

配置 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx.EPTYPE)为 0x1，表示同步

配置 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx.TXFNUM)为端点的 FIFO 号

配置 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx.CNAK)为 0x1

如果 OTGFS_DSTS.SOFFN[0] = 0x0，则 OTGFS_DIEPCTLx.SETEVENFR = 0x1 (否则 OTGFS_DIEPCTLx.SETEVENFR = 0x1)

如果 OTGFS_DSTS.SOFFN[0] = 0x1，则 OTGFS_DIEPCTLx.SETODDFR = 0x1(否则 OTGFS_DIEPCTLx.SETODDFR = 0x0)

配置 OTGFS_DIEPCTLx.EPENA = 0x1

4. 将端点数据写入相应的发送 FIFO

例如，写入地址范围

- EP1 对应 0x2000 - 0x2FFC
- EP2 对应 0x3000 - 0x3FFC
- EP3 对应 0x3000 - 0x3FFC
- …

5. 等待中断

- 当 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.XFERC)中断产生时，清除 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.XFERC)；对于下一个传输任务，重复步骤 3-5，直到完成传输。
- 当 OTGFS 中断寄存器(OTGFS_GINTSTS.INCOMPISOIN)中断产生时，清除 OTGFS 中断寄存器(OTGFS_GINTSTS.INCOMPISOIN)；对于任何一个同步 IN 端点，当奇/偶位与当前帧号位 0 一致并且在端点保持使能的情况下，控制器在帧结束时会生成该中断。下列情况会导致该中断：

（1）帧内没有令牌

（2）数据写入接收 FIFO 较晚，数据还没写完，IN 令牌就到了

（3）IN 令牌出错。

OTGFS 中断寄存器(OTGFS_GINTSTS.INCOMPISOIN)是一个全局中断。所以，当不止一个同步端点处于激活状态时，应用程序必须判断哪一个同步 IN 端点没有完成数据传输。

为了实现这一步，需要读取所有同步端点的 DSTS 和 DIEPCTLx 位。如果当前端点已使能，并且 OTGFS 设备状态寄存器(OTGFS_DSTS.SOFFN)的读取返回值是该端点的目标帧号，那么该端点就未完成传输。应用程序必须准确地跟踪并更新该同步端点的目标帧号。

如果某个端点未完成传输，那么需翻转奇/偶位。

接着：

（1）当 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx.DPID)位为 1（奇帧）时，向 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx.SETD0PID)写 1 使其成为一个偶帧，然后，当下一个帧有 IN 令牌输入时，就会发送数据。

（2）当 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx.DPID)为 0（偶帧）时，向 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx.SETD1PID)写 1 使其变成奇帧，这样，当下一个帧有 IN 令牌输入时，就会发送数据。

21.5.4.18 未完成同步OUT数据传输

若需要在上电复位时初始化控制器，那么应用程序需要根据“OTGFS 初始化”流程进行操作。在与主机通讯之前，必须先按照“端点初始化”的流程初始化端点。本节介绍了当控制器丢失了同步 OUT 数据包时，应用程序的设置流程。

【内部数据流】

1. 对于同步 OUT 端点来说，并不意味着始终会生成 OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)中断。如果控制器丢失了同步 OUT 数据包，那么应用程序在遇到下列情况时可能无法检测到 OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)中断。

- 当接收 FIFO 无法容纳完整的 ISO OUT 数据包时，控制器会丢失接收到的 ISO OUT 数据。
- 当接收的同步 OUT 数据包含有 CRC 错误时。
- 当控制器接收到的同步 OUT 令牌被损坏时。
- 当应用程序读取接收 FIFO 数据的速度非常慢的时候。

2.当控制器在传输完成之前检测到所有同步 OUT 端点的周期帧结束时，会生成未完成同步 OUT 数据中断，表示至少有一个同步 OUT 端点没有生成 OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)中断。此时，虽然未完成数据传输的这个端点保持使能状态，但是该端点并未进行有效传输。

【应用程序操作流程】

1. 宣告未完成同步 OUT 数据中断表示在当前帧内，至少有一个同步 OUT 端点没有完成数据传输。
2. 如果这是因为该端点的同步 OUT 数据没有完全读取导致的，那么应用程序必须读取接收 FIFO 中的所有同步 OUT 数据（包括数据和状态），再进行下一步处理。

- 在读取完接收 FIFO 中的所有数据后，应用程序会检测到 OTGFS 设备端点 x 中断寄存器(OTGFS_DOEPINTx.XFERC)中断。此时，应用程序需要按照“控制读传输(SETUP/Data IN/Status OUT)”的流程重新使能该端点来接收下一个帧的同步 OUT 数据。

3. 在收到到未完成同步 OUT 数据中断时，应用程序需要读取所有同步 OUT 端点的控制寄存器来确认当前帧内哪一个端点没有完成传输。如果下列两个条件都满足时，则表示端点就没有完成传输。

- OTGFS_DOEPCTLx.偶/奇帧位= OTGFS_DSTS.SOFFN[0]
- OTGFS_DOEPCTLx.端点使能 = 0x1

4. 必须在检测到 GINTSTS.SOF 中断之前执行上一个步骤，以确保当前帧号不被更改。
5. 对于没有完成传输的同步 OUT 端点，应用程序必须丢弃存储器中的数据，并通过 OTGFS_DOEPCTLx.端点禁用位来禁用此端点。
6. 等待 OTGFS_DOEPINTx.端点禁用中断，并按照“控制读传输(SETUP/Data IN/Status OUT)”的流程使能该端点接收下一个帧的新数据。由于控制器需要花一些时间禁用此端点，所以应用程序在接收到错误数据之后可能无法接收到下一个帧的数据。

21.5.4.19 未完成同步IN数据传输

本节介绍了当同步 IN 数据传输未完成时应用程序该如何操作。

【内部数据流】

1. 在下列情况下，同步 IN 传输被视为未完成。

- 控制器在不止一个同步 IN 端点接收到已损坏的同步 IN 令牌。此时应用程序会检测到 GINTSTS.未完成同步 IN 传输中断。
- 应用程序向发送 FIFO 写入完整数据的速度较慢，且还未完成写入就接收到 IN 令牌。此时，应用程序会检测到 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.INTKNTXFEMP)中断信号。应用程序会忽略此中断，因为这将导致在帧结束时生成 OTGFS_GINTSTS.未完成同步 IN 传输中断。控制器向 USB 发线一个零长度的数据包来回应接收到的 IN 令牌。

2. 无论处于以上哪种情况，应用程序必须尽快停止向发送 FIFO 写数据。
3. 应用程序必面设置该端点的 NAK 位和禁用位。
4. 控制器禁用此端点，清除禁用位，并触发端点禁用中断。

【应用程序操作流程】

1. 当发送 FIFO 为空时，应用程序会忽略任何一个同步 IN 端点上的 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.INTKNTXFEMP)中断信号，因为这会触发 OTGFS_GINTSTS.未完成同步 IN 传输中

断。
2. OTGFS_GINTSTS.未完成同步 IN 传输中断表示至少一个同步 IN 端点没有完成同步 IN 传输。
3. 应用程序必须读取所有同步 IN 端点的端点控制寄存器来确认哪一个端点没有完成 IN 数据传输。
4. 应用程序必须向这些端点的周期发送 FIFO 写入数据。
5. 通过设置 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx)的下列位域来禁用该端点
- OTGFS_DIEPCTLx.SETNAK = 0x1
- OTGFS_DIEPCTLx.端点使能= 0x1

6. DIEPINTx.端点禁用中断表示控制器已禁用了此断点。
7. 此时，应用程序必须通过使能该端点用于传输下一个帧来清空相关发送 FIFO 中的数据或者覆盖 FIFO 中现有的数据。应用程序必须通过 OTGFS 复位寄存器(OTGFS_GRSTCTL)来刷新该数据。

21.5.4.20 周期性IN（中断和同步）数据传输

本节介绍了典型的周期性 IN 数据传输操作。
如果需要在上电复位后初始化控制器，应用程序必须遵循“OTG 初始化”的流程进行操作。在与主机通讯之前，必须按照“端点初始化”流程来初始化端点。

【应用程序要求】

1. “非周期性（批量和控制）IN 数据传输”应用程序要求也适用于周期性 IN 数据传输，除了第 2 点要求略微不同。
- 应用程序发送的数据仅限于最大包长度的数据包的倍数包，以及短包。只有在满足下列条件时，才能发送几个最大包长度的数据包和短包。

传输长度[epnum] = n * mps[epnum] + sp，(其中 n、i 是≥ 0 的整数，且 0 ≤ sp < mps[epnum])
如果(sp > 0)，包数[epnum] = n + 1。否则，包数[epnum] = n，mc[epnum] = 包数 [epnum]

- 应用程序不能在传输结束时发送零长度的数据包。但其本身是可以发送一个单独的零长度的数据包，条件是：传输长度[epnum] = 0；包数目[epnum] = 1；mc[epnum] =包数目[epnum]

2. 应用程序一次只能调度传输 1 帧数据。
- (OTGFS_DIEPTSIZx.MC – 1) * OTGFS_DIEPCTLx.MPS ≤OTGFS_DIEPTSIZx.XFERSIZ ≤OTGFS_DIEPTSIZx.MC * OTGFS_DIEPCTLx.MPS
- OTGFS_DIEPTSIZx.PKTCNT = OTGFS_DIEPTSIZx.MC
- 如果 OTGFS_DIEPTSIZx.XFERSIZ < OTGFS_DIEPTSIZx.MC * OTGFS_DIEPCTLx.MPS，最后一个数据包传输是一个短包。

3. 对于周期性 IN 端点，必须在下一个帧传输之前预取 1 个帧数据。可以通过在调度数据传输的帧之前使能周期性 IN 端点 1 帧来完成这个操作。
4. 应用程序在接收周期性 IN 令牌之前必须将帧要传输的完整数据写入发送 FIFO。即使在收到周期 IN 令牌时，发送 FIFO 里缺失了帧需要传输的 1 WORD 数据，控制器将 FIFO 为空来处理。当发送 FIFO 为空时，USB 会针对 ISO IN 端点发送零长度数据包。并针对 INTR IN 端点发送 NAK 握手信号。

【内部数据流】

1. 应用程序必须设置端点控制器的传输长度和包数目位，并使能该端点来传输该数据。
2. 应用程序也可以将所需数据写入相关的发送 FIFO。
3. 每当应用程序将发送 FIFO 写入一个包时，该端点的传输长度会减去包长度。需要持续写入数据直到该端点的传输长度变为 0。
4. 当收到某个周期性端点的 IN 令牌时，应用程序会将数据写入 FIFO（如果有的话）。如果 FIFO 里不存在该帧的完整数据，那么控制器会生成 INTKNTXFEMP 中断。
- USB 针对同步 IN 端点发送一个零长度的数据包。
- USB 针对中断 IN 端点发送一个 NAK 握手信号。

5. 端点的包数目在遇到下列情况时会自动减 1：
- 对于同步端点，发送了长度或非零长度的数据包时
- 对于中断端点，当发送了 ACK 握手信号
- 当传输长和包数目都变为 0 时，会生成会该端点的传输完成中断，并清除端点使能位。

6. 在“周期性帧间隔”中（通过设置 OTGFS 设备配置寄存器(OTGFS_DCFG.PERFRINT)），当控制器检测到用于调度当前帧非空的任意一个 IN 端点 FIFO 非空时，控制器会生成 OTGFS 中断寄存器(OTGFS_GINTSTS.INCOMPISOIN)中断。

【应用程序操作流程（帧传输）】

1. 设置 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DIEPTSIZx)
2. 根据端点特性设置 OTGFS 设备端点 x 控制寄存器(OTGFS_DIEPCTLx)并设置 CNAK 和端点使能位。
3. 将下一个帧要传输的数据写入发送 FIFO。
4. 当产生 INTKNTXFEMP 中断表示应用程序尚未将所有待发送的数据写入到发送 FIFO。
5. 如果在检测到该中断时，中断端点已使能，请忽略此中断。如果未使能，请使能该端点以便在下一个 IN 令牌传输数据。如果在检测到该中断时同步端点已使能，详见“未完成同步 IN 数据传输”。
6. 当中断 IN 端点被设置为周期性端点，控制器内部可以处理中断 IN 端点发生的超时情况，并不需要应用程序的介入。因此，应用程序永远无法检测到周期性中断 IN 端点上的 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.TIMEOUT)中断信号。
7. 产生 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.XFERC)中断但是没有产生 INTKNTXFEMP 中断表示成功完成了同步 IN 传输。读取 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DIEPTSIZx)时，只有当传输长度为 0 且包数目均为 0 才表示 USB 线上的所有数据均已完成传输。
8. 产生 OTGFS 设备端点 x 中断寄存器(OTGFS_DIEPINTx.XFERC)中断但是不管有没有产生 INTKNTXFEMP 中断表示成功完成了一个中断 IN 传输。读取 OTGFS 设备端点 x 传输长度寄存器(OTGFS_DIEPTSIZx)时，只有当传输长度为 0 且包数目均为 0，才表示 USB 线上的所有数据均已完成传输。
9. 产生了 INCOMPISOIN 中断但没有产生上述所提到的中断即表示控制器没有接收到当前帧发送的至少 1 个周期性 IN 令牌。关于同步 IN 端点，详见“未完成同步 IN 数据传输”。

21.6 OTGFS控制和状态寄存器

应用程序通过 AHB 从接口来读写控制和状态寄存器(CSRx)，从而实现对 OTGFS 控制器的控制。这些寄存器都是 32 位访问的，并且 32 位地址对齐。

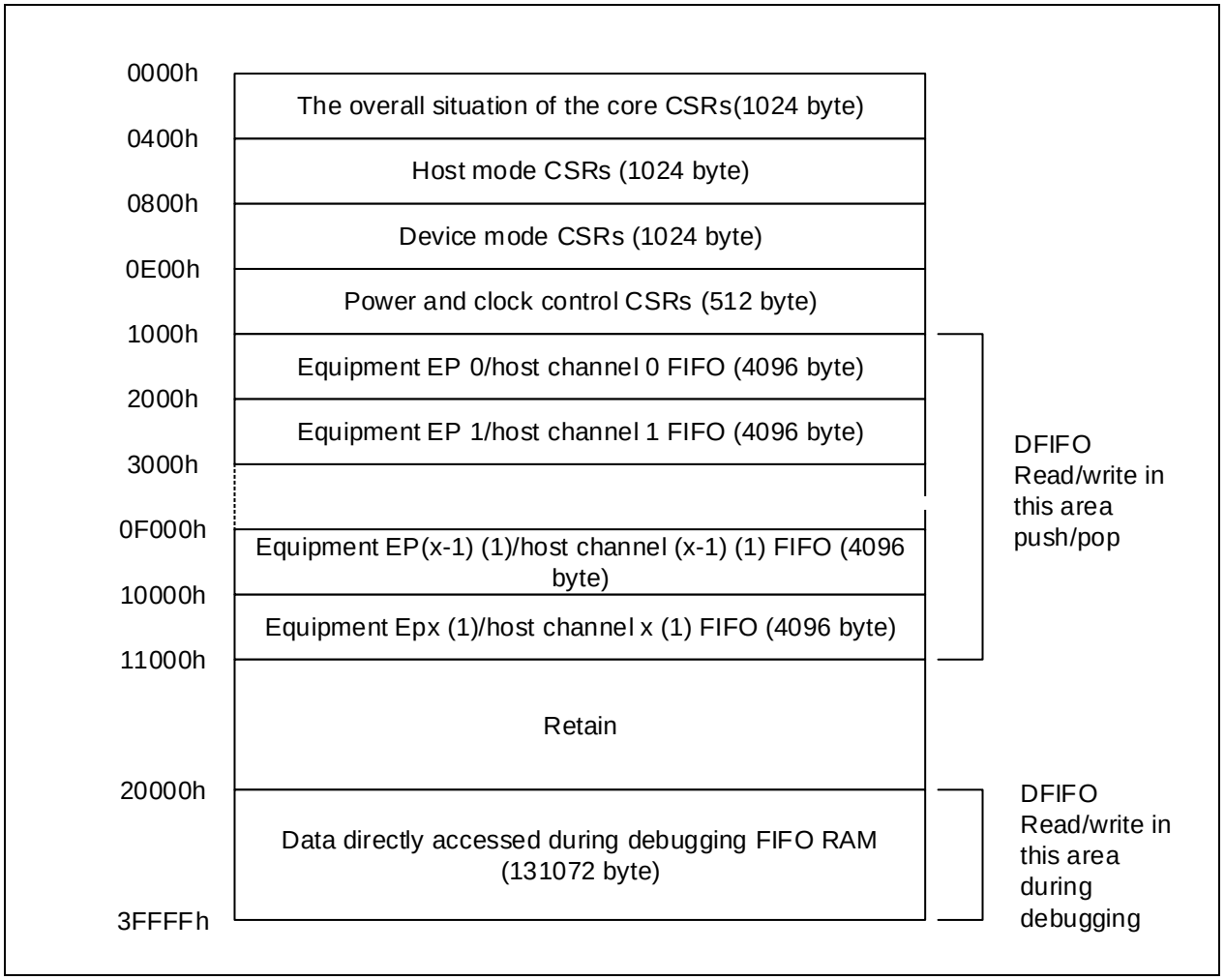
只有控制器全局寄存器，供电和时钟控制寄存器，数据 FIFO 访问寄存器以及主机端口控制和状态寄存器在主机模式和设备模式下都有效。无论 OTGFS 控制器工作在主机模式还是设备模式下，应用程序都不能访问另一种模式下的寄存器组。如果应用程序发生了非法访问，会产生模式不匹配中断并影响控制器中断寄存器的相应位(OTGFS_GINTSTS 寄存器的 MODMIS 位)。

当控制器从一种模式切换到另一种模式时，新模式下的寄存器组都需要和上电复位时一样的重新初始化。必须以字(32 位)的方式操作这些外设寄存器。

21.6.1 CSR寄存器映像

主机模式寄存器和设备模式寄存器占据不同的地址。所有的寄存器都位于 AHB 时钟域。

图 21-13 CSR存储器映像



在设备模式下 x 为 7，在主机模式下 x 为 15

OTGFS 控制和状态寄存器可分为 OTGFS 全局寄存器、主机模式下寄存器、设备模式下寄存器、数据 FIFO(DFIFO)访问寄存器、供电和时钟控制寄存器。

1、OTGFS 全局寄存器：这些寄存器在主机模式和设备模式下都有效，寄存器首字母缩写为 G；

2、主机模式下寄存器：这些寄存器在每次切换到主机模式时都需要配置，寄存器首字母缩写为 H；

3、设备模式下寄存器：这些寄存器在每次切换到设备模式时都需要配置，寄存器首字母缩写为 D；

4、数据 FIFO(DFIFO)访问寄存器：这组寄存器列在主机模式和设备模式下都有效，用于读写指定方向的特殊端点或通道的 FIFO。如果一个主机模式下的通道是 IN 类型的，相对应的 FIFO 只能进行读操作。同样地，如果一个主机模式下的通道是 OUT 类型的，相对应的 FIFO 只能进行写操作。

5、供电和时钟控制寄存器：只有一个寄存器用来控制供电和时钟控制，此寄存器在设备模式下和主机模式下都有效

21.6.2 OTGFS寄存器地址映象

下表给出了 USB OTG 寄存器映像和复位值。

必须以字（32 位）的方式操作这些外设寄存器。

表 21-4 OTGFS模块的寄存器图及其复位值

寄存器简称	基址偏移量	复位值
OTGFS_GOTGCTL	0x000	0x0001 0000

OTGFS_GOTGINT	0x004	0x0000 0000
OTGFS_GAHBCFG	0x008	0x0000 0000
OTGFS_GUSBCFG	0x00C	0x0000 1440
OTGFS_GRSTCTL	0x010	0x8000 0000
OTGFS_GINTSTS	0x014	0x0400 0020
OTGFS_GINTMSK	0x018	0x0000 0000
OTGFS_GRXSTSR	0x01C	0x0000 0000
OTGFS_GRXSTSP	0x020	0x0000 0000
OTGFS_GRXFSIZ	0x024	0x0000 0200
OTGFS_GNPTXFSIZ	0x028	0x0000 0200
OTGFS_GNPTXSTS	0x02C	0x0008 0200
OTGFS_GCCFG	0x038	0x0000 0000
OTGFS_GUID	0x03C	0x0000 1000
OTGFS_HPTXFSIZ	0x100	0x0200 0600
OTGFS_DIEPTXF1	0x104	0x0200 0400
OTGFS_DIEPTXF2	0x108	0x0200 0400
OTGFS_DIEPTXF3	0x10C	0x0200 0400
OTGFS_DIEPTXF4	0x110	0x0200 0400
OTGFS_DIEPTXF5	0x114	0x0200 0400
OTGFS_DIEPTXF6	0x118	0x0200 0400
OTGFS_DIEPTXF7	0x11C	0x0200 0400
OTGFS_DIEPTXF8	0x120	0x0200 0400
OTGFS_DIEPTXF9	0x124	0x0200 0400
OTGFS_DIEPTXF10	0x128	0x0200 0400
OTGFS_DIEPTXF11	0x12C	0x0200 0400
OTGFS_DIEPTXF12	0x130	0x0200 0400
OTGFS_DIEPTXF13	0x134	0x0200 0400
OTGFS_DIEPTXF14	0x138	0x0200 0400
OTGFS_DIEPTXF15	0x13C	0x0200 0400
OTGFS_DIEPTXF16	0x140	0x0200 0400
OTGFS_HCFG	0x400	0x0000 0000
OTGFS_HFIR	0x404	0x0000 EA60
OTGFS_HFNUM	0x408	0x0000 3FFF
OTGFS_HPTXSTS	0x410	0x0008 0100
OTGFS_HAINT	0x414	0x0000 0000
OTGFS_HAINTMSK	0x418	0x0000 0000
OTGFS_HPRT	0x440	0x0000 0000
OTGFS_HCCHAR0	0x500	0x0000 0000
OTGFS_HCINT0	0x508	0x0000 0000
OTGFS_HCINTMSK0	0x50C	0x0000 0000
OTGFS_HCTSIZ0	0x510	0x0000 0000

OTGFS_HCCHAR1	0x520	0x0000 0000
OTGFS_HCINT1	0x528	0x0000 0000
OTGFS_HCINTMSK1	0x52C	0x0000 0000
OTGFS_HCTSIZ1	0x530	0x0000 0000
OTGFS_HCCHAR2	0x540	0x0000 0000
OTGFS_HCINT2	0x548	0x0000 0000
OTGFS_HCINTMSK2	0x54C	0x0000 0000
OTGFS_HCTSIZ2	0x550	0x0000 0000
OTGFS_HCCHAR3	0x560	0x0000 0000
OTGFS_HCINT3	0x568	0x0000 0000
OTGFS_HCINTMSK3	0x56C	0x0000 0000
OTGFS_HCTSIZ3	0x570	0x0000 0000
OTGFS_HCCHAR4	0x580	0x0000 0000
OTGFS_HCINT4	0x588	0x0000 0000
OTGFS_HCINTMSK4	0x58C	0x0000 0000
OTGFS_HCTSIZ4	0x590	0x0000 0000
OTGFS_HCCHAR5	0x5A0	0x0000 0000
OTGFS_HCINT5	0x5A8	0x0000 0000
OTGFS_HCINTMSK5	0x5AC	0x0000 0000
OTGFS_HCTSIZ5	0x5B0	0x0000 0000
OTGFS_HCCHAR6	0x5C0	0x0000 0000
OTGFS_HCINT6	0x5C8	0x0000 0000
OTGFS_HCINTMSK6	0x5CC	0x0000 0000
OTGFS_HCTSIZ6	0x5D0	0x0000 0000
OTGFS_HCCHAR7	0x5E0	0x0000 0000
OTGFS_HCINT7	0x5E8	0x0000 0000
OTGFS_HCINTMSK7	0x5EC	0x0000 0000
OTGFS_HCTSIZ7	0x5F0	0x0000 0000
OTGFS_HCCHAR8	0x600	0x0000 0000
OTGFS_HCINT8	0x608	0x0000 0000
OTGFS_HCINTMSK8	0x60C	0x0000 0000
OTGFS_HCTSIZ8	0x610	0x0000 0000
OTGFS_HCCHAR9	0x620	0x0000 0000
OTGFS_HCINT9	0x628	0x0000 0000
OTGFS_HCINTMSK9	0x62C	0x0000 0000
OTGFS_HCTSIZ9	0x630	0x0000 0000
OTGFS_HCCHAR10	0x640	0x0000 0000
OTGFS_HCINT10	0x648	0x0000 0000
OTGFS_HCINTMSK10	0x64C	0x0000 0000
OTGFS_HCTSIZ10	0x650	0x0000 0000
OTGFS_HCCHAR11	0x660	0x0000 0000

OTGFS_HCINT11	0x668	0x0000 0000
OTGFS_HCINTMSK11	0x66C	0x0000 0000
OTGFS_HCTSIZ11	0x670	0x0000 0000
OTGFS_HCCHAR12	0x680	0x0000 0000
OTGFS_HCINT12	0x688	0x0000 0000
OTGFS_HCINTMSK12	0x68C	0x0000 0000
OTGFS_HCTSIZ12	0x690	0x0000 0000
OTGFS_HCCHAR13	0x6A0	0x0000 0000
OTGFS_HCINT13	0x6A8	0x0000 0000
OTGFS_HCINTMSK13	0x6AC	0x0000 0000
OTGFS_HCTSIZ13	0x6B0	0x0000 0000
OTGFS_HCCHAR14	0x6C0	0x0000 0000
OTGFS_HCINT14	0x6C8	0x0000 0000
OTGFS_HCINTMSK14	0x6CC	0x0000 0000
OTGFS_HCTSIZ14	0x6D0	0x0000 0000
OTGFS_HCCHAR15	0x6E0	0x0000 0000
OTGFS_HCINT15	0x6E8	0x0000 0000
OTGFS_HCINTMSK15	0x6EC	0x0000 0000
OTGFS_HCTSIZ15	0x6F0	0x0000 0000
OTGFS_DCFG	0x800	0x0820 0000
OTGFS_DCTL	0x804	0x0000 0002
OTGFS_DSTS	0x808	0x0000 0010
OTGFS_DIEPMSK	0x810	0x0000 0000
OTGFS_DOEPMSK	0x814	0x0000 0000
OTGFS_DAINT	0x818	0x0000 0000
OTGFS_DAINTMSK	0x81C	0x0000 0000
OTGFS_DIEPEMPMSK	0x834	0x0000 0000
OTGFS_DIEPCTL0	0x900	0x0000 0000
OTGFS_DIEPINT0	0x908	0x0000 0080
OTGFS_DIEPTSIZ0	0x910	0x0000 0000
OTGFS_DTXFSTS0	0x918	0x0000 0200
OTGFS_DIEPCTL1	0x920	0x0000 0000
OTGFS_DIEPINT1	0x928	0x0000 0080
OTGFS_DIEPTSIZ1	0x930	0x0000 0000
OTGFS_DTXFSTS1	0x938	0x0000 0200
OTGFS_DIEPCTL2	0x940	0x0000 0000
OTGFS_DIEPINT2	0x948	0x0000 0080
OTGFS_DIEPTSIZ2	0x950	0x0000 0000
OTGFS_DTXFSTS2	0x958	0x0000 0200
OTGFS_DIEPCTL3	0x960	0x0000 0000
OTGFS_DIEPINT3	0x968	0x0000 0080

OTGFS_DIEPTSIZ3	0x970	0x0000 0000
OTGFS_DTXFSTS3	0x978	0x0000 0200
OTGFS_DIEPCTL4	0x980	0x0000 0000
OTGFS_DIEPINT4	0x988	0x0000 0080
OTGFS_DIEPTSIZ4	0x990	0x0000 0000
OTGFS_DTXFSTS4	0x998	0x0000 0200
OTGFS_DIEPCTL5	0x9A0	0x0000 0000
OTGFS_DIEPINT5	0x9A8	0x0000 0080
OTGFS_DIEPTSIZ5	0x9B0	0x0000 0000
OTGFS_DTXFSTS5	0x9B8	0x0000 0200
OTGFS_DIEPCTL6	0x9C0	0x0000 0000
OTGFS_DIEPINT6	0x9C8	0x0000 0080
OTGFS_DIEPTSIZ6	0x9D0	0x0000 0000
OTGFS_DTXFSTS6	0x9D8	0x0000 0200
OTGFS_DIEPCTL7	0x9E0	0x0000 0000
OTGFS_DIEPINT7	0x9E8	0x0000 0080
OTGFS_DIEPTSIZ7	0x9F0	0x0000 0000
OTGFS_DTXFSTS7	0x9F8	0x0000 0200
OTGFS_DOEPCTL0	0xB00	0x0000 8000
OTGFS_DOEPINT0	0xB08	0x0000 0000
OTGFS_DOEPTSIZ0	0xB10	0x0000 0000
OTGFS_DOEPCTL1	0xB20	0x0000 0000
OTGFS_DOEPINT1	0xB28	0x0000 0000
OTGFS_DOEPTSIZ1	0xB30	0x0000 0000
OTGFS_DOEPCTL2	0xB40	0x0000 0000
OTGFS_DOEPINT2	0xB48	0x0000 0000
OTGFS_DOEPTSIZ2	0xB50	0x0000 0000
OTGFS_DOEPCTL3	0xB60	0x0000 0000
OTGFS_DOEPINT3	0xB68	0x0000 0000
OTGFS_DOEPTSIZ3	0xB70	0x0000 0000
OTGFS_DOEPCTL4	0xB80	0x0000 0000
OTGFS_DOEPINT4	0xB88	0x0000 0000
OTGFS_DOEPTSIZ4	0xB90	0x0000 0000
OTGFS_DOEPCTL5	0xBA0	0x0000 0000
OTGFS_DOEPINT5	0xBA8	0x0000 0000
OTGFS_DOEPTSIZ5	0xBB0	0x0000 0000
OTGFS_DOEPCTL6	0xBC0	0x0000 0000
OTGFS_DOEPINT6	0xBC8	0x0000 0000
OTGFS_DOEPTSIZ6	0xBD0	0x0000 0000
OTGFS_DOEPCTL7	0xBE0	0x0000 0000
OTGFS_DOEPINT7	0xBE8	0x0000 0000

OTGFS_DOEPTSIZ7	0xBF0	0x0000 0000
OTGFS_PCGCCTL	0xE00	0x0000 0000

21.6.3 OTGFS全局寄存器

该寄存器适用于主机模式和设备模式，在两种模式之间切换时不需要重新配置。

21.6.3.1 OTGFS状态控制器(OTGFS_GOTGCTL)

OTG 控制和状态寄存器用于控制 OTG 功能并反映其功能状态。

域	简称	复位值	类型	功能
位 31：22	保留	0x0000	resd	请保持默认值。
位 21	CURMOD	0x0	ro	当前工作模式(Current Mode of Operation) 适用模式：主机模式和设备模式 该位指示当前模式。 0：设备模式 1：主机模式
位 20：17	保留	0x0000	resd	请保持默认值。
位 16	CONIDSTS	0x1	ro	适用模式：主机模式和设备模式 连接器 ID 状态(Connector ID Status) 表示连接器 ID 状态。 0：OTGFS 控制器处于 A-设备模式 1：OTGFS 控制器处于 B-设备模式
位 15：0	保留	0x0000	resd	请保持默认值。

21.6.3.2 OTGFS OTG中断状态控制器(OTGFS_GOTGINT)

应用程序可以通过读此寄存器得知发生了何种 OTG 中断，并可以通过写此寄存器清除对应的 OTG 中断。

域	简称	复位值	类型	功能
位 31：3	保留	0x0000	resd	请保持默认值。
位 2	SESENDDET	0x0	rw1c	适用模式：主机模式和设备模式 会话结束检测(Session End Detected) 当控制器检测到 Bvalid（此芯片即 Vbus）信号断开时置起此位。该寄存器只能由硬件置 1，软件可通过写 1 清除此位。
位 1：0	保留	0x0000	resd	请保持默认值。

21.6.3.3 OTGFS AHB配置寄存器(OTGFS_GAHBCFG)

此寄存器用于在上电或改变控制器模式时配置控制器。此寄存器主要配置一些和 AHB 相关的参数。在初始化配置完成后，不要再修改此寄存器。应用程序在开始向 AHB 或 USB 传送数据前必需先配置好此寄存器。

域	简称	复位值	类型	功能
位 31：9	保留	0x000000	resd	保持默认值。
位 8	PTXFEMPLVL	0x0	rw	适用模式：仅适用于主机模式 周期性发送 FIFO 空级别(Periodic TxFIFO Empty Level) 表示何时会触发控制器中断寄存器(GINTSTS.PTXFEMP)的周期性发送 FIFO 空中断位。 0：GINTSTS.PTXFEMP 中断表示周期性发送 FIFO 是半空 1：GINTSTS.PTXFEMP 中断表示周期性发送 FIFO 是全空
位 7	NPTXFEMPLVL	0x0	rw	适用模式：主机模式和设备模式 非周期性发送 FIFO 空级别(Non-Periodic TxFIFO Empty Level) 在主机模式下，该位表示何时会触发控制中断寄存器(GINTSTS.NPTXFEMP)的非周期性发送 FIFO 空中断位。 在设备模式下，该位表示何时会触发 IN 端点发送 FIFO 空中断(DIEPINTn.TXFEMP)。

				0：DIEPINTn.TxFEMP 中断表示 IN 端点发送 FIFO 是半空 1：DIEPINTn.TxFEMP 中断表示 IN 端点发送 FIFO 是全空
位 6：1	保留	0x00	resd	保持默认值。
位 0	GLBINTMSK	0x0	rw	模式：主机模式和设备模式 全局中断屏蔽(Global Interrupt Mask) 应用程序通过该位来屏蔽/取消屏蔽中断线发给自己的中断。无论该位是否置起，控制器仍会更新中断状态寄存器。 0：屏蔽对应用程序的中断 1：不屏蔽对应用程序的中断

21.6.3.4 OTGFS_USB配置寄存器(OTGFS_GUSBCFG)

该寄存器用于在上电后或者在切换主机模式或设备模式时配置控制器。该寄存器包含了 USB 和 USB-PHY 相关的参数。应用程序在处理 AHB 或 USB 事务之前，必须先设置该寄存器。初始化配置之后，请不要再修改本寄存器。

域	简称	复位值	类型	功能
位 31	COTXPKT	0x0	rw	适用模式：主机模式和设备模式 发送包损坏(Corrupt Tx packet) 该位仅用于调试。请勿将该位设置为 1。
位 30	FDEVMODE	0x0	rw	适用模式：主机模式和设备模式 强制设备模式(Force Device Mode) 无论 ID 输入引脚的状态如何，向该位写 1 可强制控制器进入设备模式。 0：普通模式 1：强制设备模式 将该置起后，应用程序必须等待至少 25ms 才能使设置生效。
位 29	FHSTMODE	0x0	rw	适用模式：主机模式和设备模式 强制主机模式(Force Host Mode) 无论 ID 输入引脚的状态如何，向该位写 1 可强制控制器进入主机模式。 0：普通模式 1：强制主机模式 将该置起后，应用程序必须等待至少 25ms 才能使设置生效。
位 28：15	保留	0x0000	resd	保持默认值。
位 14	保留	0x0	resd	保持默认值。
位 13：10	USBTRDTIM	0x5	rw	适用模式：仅适用设备模式 USB 周转时间(USB Turnaround Time) 以 PHY 时钟为单位设置周转时间。规定了 MAC 向包 FIFO 控制器（PFC）发起请求从 DFIFO (SPRAM)提取数据的响应时间。这些位必须配置为： 0101：当 MAC 接口为 16-bit UTMI+时 1001：当 MAC 接口为 8-bit UTMI+时。 注意：以上数值是基于最低 30MHz 的 AHB 频率计算得出的。USB 周转时间对于用到长线和 5-Hubs 的认证至关重要，所以如果你需要 AHB 在低于 30 MHz 的频率下运行，而且 USB 周转时间不重要的话，可以将这些位域的值设置大一些。
位 9：3	保留	0x08	resd	保持默认值。
位 2：0	TOUTCAL	0x0	rw	适用模式：主机模式和设备模式 全速超时校准(FS Timeout Calibration) 将应用程序在这些位域设置的 PHY 时钟数添加到全速数据包间超时时段内，以补偿 PHY 引起的额外延迟。这个动作可能是必须的，因为 PHY 在生成线路状态条件时引起的延迟会因 PHY 不同而有所不同。

				全速模式下，USB 标准超时值是 16~18 个 bit(包含 18)时间。应用程序必须要根据枚举的速度来设置此位域。每个 PHY 时钟增加的 bit 时间为 0.25bit 时间

21.6.3.5 OTGFS复位寄存器(OTGFS_GRSTCTL)

应用程序通过本寄存器来复位控制器的各硬件模块。

域	简称	复位值	类型	功能
位 31	AHBIDLE	0x1	ro	适用模式：主机模式和设备模式 AHB 主机空闲(AHB Master Idle) 指示 AHB 主机状态机处于空闲状态。
位 30：11	保留	0x000	resd	保持默认值。
位 10：6	TXFNUM	0x00	rw	适用模式：主机模式和设备模式 发送 FIFO 编号(TxFIFO Number) 此位域表示哪个 FIFO 需要通过发送 FIFO 刷新位(TxFIFO Flush)来刷新。除非控制器已清除 TxFIFO Flush 位，否则不允许更改此位域。 00000： - 主机模式下非周期性 TxFIFO - 设备模式下 Tx FIFO 0 00001： - 主机模式下周期性 TxFIFO - 设备模式下 TXFIFO 1 00010： - 设备模式 TXFIFO 2 ... 01111： - 设备模式下 TXFIFO 15 10000： - 刷新位于设备模式或主机模式下的所有发送 FIFO
位 5	TXFFLSH	0x0	rw1s	适用模式：主机模式和设备模式 发送 FIFO 刷新(TxFIFO Flush) 该位有选择性地刷新单独一个的或所有的发送 FIFO，但是必须确保控制器没有正在处理的事务。 应用程序必须先确认控制器没有读/写 TxFIFO，才能对此位进行写操作。 通过这些寄存器来验证： 读：NAK 有效中断(NAK Effective Interrup)确保控制器没有读取 FIFO 写：GRSTCTL.AHBIDLE 确保控制器没有对 FIFO 进行写操作。 在重新配置 FIFO 时，通常建议用户进行刷新操作(Flushing)。 在设备端点禁用期间，也建议使用 FIFO 刷新操作。应用程序必须要等待控制器清除此位之后才能进行其它操作。 该位需要 8 个时钟来完成清除（以 phy_clk 或 hclk 的较慢的那个时钟为准来计算）。
位 4	RXFFLSH	0x0	rw1s	适用模式：主机模式和设备模式 接收 FIFO 刷新(RxFIFO Flush) 应用程序可以通过该位来刷新整个接收 FIFO，但必须先确保控制器没有正在处理的事务。应用程序必须先确认控制器没有读/写 RxFIFO，才能对此位进行写操作 应用程序必须要等待控制器清除此位之后才能进行其它操作。该位需要 8 个时钟周期来完成清除（以 PHY 或 AHB 最慢的那个时钟为准来计算）。
位 3	保留	0x0	resd	保持默认值。
位 2	FRMCNTRST	0x0	rw1s	适用模式：仅适用主机模式 主机帧计数器复位(Host Frame Counter Reset) 应用程序通过此位来复位控制器的帧数计数器。将帧计数器复位之后，控制器随后发出的 SOF 的帧号为 0。

				如果应用程序向该位写 1，它可能无法读取到这个值，因为该位会在几个时钟周期之后被控制器清除。
位 1	PIUSFTRST	0x0	rw1s	适用模式：主机模式和设备模式 PIU 全速专用控制器软件复位(PIU FS Dedicated Controller Soft Reset) 用于复位 PIU 全速专用控制器 PIU 全速专用控制器中的所有模块状态机即恢复到空闲状态。在发生 PHY 错误（比如操作中断或 babble）导致 PHY 保持在接收状态的时长超过 1 个帧时，可使用此位来复位 PIU 的全速专用控制器。 该位可自动清除，而且控制器中在所有必要逻辑电路复位之后会清除此位。
位 0	CSFTRST	0x0	rw1s	适用模式：主机模式和设备模式 控制器软件复位(Core Soft Reset) 复位 hclk and phy_clock 域，如下所示： 清除所有中断以及 CSR 寄存器，除如下寄存器位之外： - HCFG.FSLSPCS - DCFG.DECSPD - DCTL.SFTDIS 将所有模块状态机（除了 AHB 从机）复位到空闲状态，并清空所有的发送 FIFO 和接收 FIFO。 在完成最后一个阶段的 AHB 数据传输之后，AHB 主机上的所有任务都会尽可能快地结束。USB 上的所有任务会立即停止。 应用程序可以随时对该位进行写操作来复位控制器。此位可自动清除，控制器在将所有必要的逻辑电路复位之后会清除此位，控制器可能需要花几个时钟周期来清除，具体时间取决于控制器的当前所处的状态。清除此位后，应用程序必须要等待至少 3 个 PHY 时钟之后才能访问 PHY 域（同步延迟） 另外，应用程序在开始其它操作之前，必须要确保本寄存器的位 31 是置 1(AHB 主机是空闲状态)。 通常来讲，使用软件复位的情况为：在进行软件开发，以及当用户对上述所列的 USB 配置寄存器的 PHY 选择位进行了动态修改的情况下，需要通过软件复位。在修改 PHY 时，需要选择相应的 PHY 时钟并运用于 PHY 域。选择了新的时钟之后，必须复位 PHY 域才能进行正常操作。

21.6.3.6 OTGFS中断寄存器(OTGFS_GINTSTS)

本寄存器会因当前模式（设备模式或主机模式）发生系统事件而中断应用程序，如图 21-2 所示。
本寄存器的一些位仅适用于主机模式，一些位仅适用于设备模式。另外，该寄存器会指示当前位于哪种操作模式。
FIFO 状态寄存器中断位是只读的。一旦软件在处理这些中断时读写了 FIFO，则 FIFO 中断条件会自动被清除。
初始化时，应用程序在使能某个中断位之前，必须先清除 GINTSTS 寄存器，以避免在初始化之前产生中断。

域	简称	复位值	类型	功能
位 31	WKUPINT	0x0	rw1c	适用模式：主机模式和设备模式 检测恢复/远程唤醒信号产生中断(Resume/Remote Wakeup Detected Interrupt) 设备模式：仅当 USB 总线上检测到主机触发的恢复信号时，才产生中断 主机模式：仅当 USB 总线上检测到设备触发的远程唤醒信号时，才产生中断
位 30	保留	0x0	resd	保持默认值。
位 29	DISCONINT	0x0	rw1c	适用模式：仅适用于主机模式 检测到断开事件产生中断(Disconnect Detected Interrupt) 当检测到设备断开时，即产生中断
位 28	CONIDSCHG	0x0	rw1c	适用模式：主机模式和设备模式

				连接器 ID 状态变化(Connector ID Status Change) 当检测到连接器 ID 状态发生变化时，控制器将此位置起
位 27	保留	0x0	resd	保持默认值。
位 26	PTXFEMP	0x1	ro	适用模式：仅适用于主机模式 周期性发送 FIFO 空(Periodic TxFIFO Empty) 当周期性发送 FIFO（Periodic Transmit FIFO）处于半空或全空状态，且在周期性请求队列中有空间可写入一个请求时，即产生中断。具体是半空还是全空，取决于控制器 AHB 配置寄存器的周期性发送 FIFO 空级别位(Periodic TxFIFO Empty Level)。
位 25	HCHINT	0x0	ro	主机通道中断(Host Channels Interrupt) 控制器通过将该位置起，来指示控制器（主机模式下）的某一个通道有待处理的中断。应用程序必须读取主机全通道中断寄存器(Host All Channels Interrupt)以确定产生中断的具体通道编号，然后读取相应的主机通道编号中断寄存器(Host Channel-n Interrupt)以获取中断源。 应用程序必须通过清除 HCINTn (Host All Channels Interrupt)寄存器的相应状态位来清除该位。
位 24	PRTINT	0x0	ro	主机端口中断(Host Port Interrupt) 控制器通过将该位置起，来指示在主机模式下某个端口状态发生改变。应用程序必须通过读取主机端口控制和状态寄存器(Host Port Control and Status)来确认中断源。应用程序必须通过清除主机端口控制和状态寄存器(Host Port Control and Status)来清除此位。
位 23：22	保留	0x0	resd	保持默认值。
位 21	INCOMPIP INCOMPISOOUT	0x0	rw1c	未完成周期性传输(Incomplete Periodic Transfer) 适用模式：仅适用于主机模式 在主机模式下，如果当前帧尚有未完成的周期性传输待处理时，控制器会置起此中断位。 未完成同步 OUT 传输(Incomplete Isochronous OUT Transfer) 适用模式：仅适用于设备模式 在设备模式下，控制器置起此中断位以指示当前帧至少存在一个未完成传输的同步 OUT 端点。该中断会随同本寄存器的周期性帧结束中断位(End of Periodic Frame Interrupt)一同生成。
位 20	INCOMPISOIN	0x0	rw1c	适用模式：仅适用于设备模式 未完成同步 IN 传输(Incomplete Isochronous IN Transfer) 控制器置起此中断以指示当前帧至少存在一个未完成传输的同步 IN 端点。该中断会随同本寄存器的周期性帧结束中断位(End of Periodic Frame Interrupt)一同生成。
位 19	OEPTINT	0x0	ro	适用模式：仅适用于设备模式 OUT 端点中断(OUT Endpoints Interrupt) 控制器通过置起此位来指示控制器的某一个 OUT 端点有待处理的中断。应用程序必须通过读取设备全部端点中断寄存器(Device All Endpoints Interrupt)以确定该中断发生在哪一个 OUT 端点号上，接着会读取相应的设备 OUT 端点号中断寄存器(Device OUT Endpoint-n Interrupt)来确定本次中断源。应用程序必须通过清除设备 OUT 端点中断寄存器(Device OUT Endpoint-n Interrupt)的相应状态位来清除此位
位 18	IEPTINT	0x0	ro	适用模式：仅适用于设备模式 IN 端点中断(IN Endpoints Interrupt) 控制器通过置起此位来指示控制器的某一个 IN 端点有待处理的中断（设备模式下）。应用程序必须通过读取设备全部端点中断寄存器(Device All Endpoints Interrupt)以确定该中断发生在哪一个 IN 端点号上，接着会读取相应的设备 IN 端点号中断寄存器(Device IN Endpoint-n Interrupt)来确定本次中断源。应用程序必须通过清除相应的 Device IN Endpoint-n Interrupt 寄存器的相应状态位来清除此位。

位 17：16	保留	0x0	resd	保持默认值。
位 15	EOPF	0x0	rw1c	适用模式：仅适用于设备模式 周期性帧结束中断(End of Periodic Frame Interrupt) 该位表示当前帧已经到达了设备配置寄存器(Device Configuration)的周期帧间隔时间位所设置的周期。
位 14	ISOOUTDROP	0x0	rw1c	适用模式：仅适用于设备模式 同步 OUT 包丢失中断(Isochronous OUT Packet Dropped Interrupt) 控制器在以下情况会置起该位：控制器因为接收 FIFO 没有足够的空间来容纳同步 OUT 端点所需的最大容量的数据包而导致其无法向接收 FIFO 写入一个同步 OUT 包。
位 13	ENUMDONE	0x0	rw1c	适用模式：仅适用于设备模式 完成枚举(Enumeration Done) 控制器将该位置起以表示已完成速度枚举。 应用程序必须通过读取设备状态寄存器(Device Status)来获取枚举的速度信息。
位 12	USBRST	0x0	rw1c	适用模式：仅适用于设备模式 USB 复位(USB Reset) 控制器将该位置起以表示检测到 USB 总线上的复位信号。
位 11	USBSUSP	0x0	rw1c	适用模式：仅适用于设备模式 USB 挂起(USB Suspend) 控制器将该位置起以表示检测到 USB 总线上的挂起信号。当总线信号在很长一段时间内没有活动时，控制器进入挂起状态。
位 10	ERLYSUSP	0x0	rw1c	适用模式：仅适用于设备模式 早期挂起(Early Suspend) 控制器将该位置起以表示检测到 USB 总线空闲状态已持续 3ms。
位 9：8	保留	0x0	resd	保持默认值。
位 7	GOUTNAKEFF	0x0	ro	适用模式：仅适用于设备模式 全局 OUT NAK 生效(Global OUT NAK Effective) 该位表示设备控制寄存器(Device Control)设置 Global OUT NAK 位已生效。通过向设备控制寄存器(Device Control)写入 Clear Global OUT NAK 位可以清除该位。
位 6	GINNAKEFF	0x0	ro	适用模式：仅适用于设备模式 全局非周期性 IN NAK 生效(Global IN Non-periodic NAK Effective) 该位表示由应用程序所设置的设备控制寄存器(Device Control)的 Set Global Non-periodic IN NAK 位已生效。即，控制器对应用程序所设置的 Global IN NAK 位进行了采样。通过向设备控制寄存器(Device Control)写入 Clear Global Non-periodic IN NAK 位可以清除该位。此中断并不一定意味着 USB 总线上发送了 NAK 握手信号。STALL 位的优先级高于 NAK 位。
位 5	NPTXFEMP	0x1	ro	适用模式：主机模式和设备模式 非周期性发送 FIFO 空(Non-periodic TxFIFO Empty) 当非周期性发送 FIFO 处于半空或全空状态，并且有足够的空间可以向非周期性发送请求队列(Non-periodic Transmit Request Queue)写入至少一个请求时，即产生中断。具体是半空还是全空，取决于控制器 AHB 配置寄存器(Core AHB Configuration)的非周期性发送 FIFO 空级别位(Non-periodic TxFIFO Empty Level)。
位 4	RXFLVL	0x0	ro	适用模式：主机模式和设备模式 接收 FIFO 非空(RxFIFO Non-Empty) 表示接收 FIFO 中至少有一个包等待读取。
位 3	SOF	0x0	rw1c	适用模式：主机模式和设备模式 帧首包(Start of Frame) 在主机模式下，控制器将该位置起以指示 USB 总线发送了 SOF (全速)或者 Keep-Alive(低速)信号。 应用程序必须将位置 1 才能清除此中断。

				在设备模式下，控制器将该位置起以表示 USB 总线接收到了 SOF 令牌。应用程序必须读取设备状态寄存器(Device Status register)来获取当前帧号。只有当控制器工作在全速模式时，才能产生此中断。该位只能由控制器设置，应用程序必须写 1 才能清除此位。 注意：上电复位后立即读取该寄存器可能会返回 0x1。如果在上电复位后立即读取寄存器的值为 0x1，并不表示已发送 SOF(主机模式下)或已收到 SOF(设备模式下)。只有当主机和设备建立有效连接后，读取的寄存器的值才是有效的。如果在上电复位后将该位置起，那么应用程序可清除此位。
位 2	OTGINT	0x0	ro	适用模式：主机模式和设备模式 OTG 中断(OTG Interrupt) 控制器将该位置起以表示产生了一个 OTG 协议事件。应用程序必须读取 OTG 中断状态寄存器(OTG Interrupt Status)来确定是什么事件引起的中断。应用程序必须通过清除 OTG 中断状态寄存器(OTG Interrupt Status)的相应状态位以清除此位。
位 1	MODEMIS	0x0	rw1c	适用模式：主机模式和设备模式 模式不匹配中断(Mode Mismatch Interrupt) 当应用程序尝试访问以下内容时，控制器会置起此位： 当控制器运行在设备模式下时却尝试访问主机模式下的寄存器 当控制器运行在主机模式下时却尝试访问设备模式下的寄存器 在 AHB 上完成对寄存器的访问之后会出现 OKAY 响应，但控制器内部会忽略此种操作，所以不会影响到控制器的运行。 此位只能由控制器设置，应用程序通过写 1 才能清除此位。
位 0	CURMOD	0x0	ro	适用模式：主机模式和设备模式 当前运行模式(Current Mode of Operation) 此位指示的是当前的运行模式。 0：设备模式 1：主机模式

21.6.3.7 OTGFS中断屏蔽寄存器(OTGFS_GINTMSK)

本寄存器与“中断寄存器”(Interrupt Register)互相配合来中断应用程序。屏蔽某一个中断位后，就不会生成与该中断位相关的中断。然而与该中断所对应的中断寄存器位仍然是置起状态。

屏蔽中断：0

解除中断：1

域	简称	复位值	类型	功能
位 31	WKUPINTMSK	0x0	rw	适用模式：主机模式和设备模式 屏蔽由恢复/远程唤醒检测到的中断(Resume/Remote Wakeup Detected Interrupt Mask)
位 30	保留	0x0	resd	保持默认值。
位 29	DISCONINTMSK	0x0	rw	适用模式：主机模式和设备模式 屏蔽断开事件检测到的中断(Disconnect Detected Interrupt Mask)
位 28	CONIDSCHGMSK	0x0	rw	适用模式：主机模式和设备模式 屏蔽连接器 ID 状态改变(Connector ID Status Change Mask)
位 27	保留	0x0	resd	保持默认值。
位 26	PTXFEMPMSK	0x0	rw	适用模式：仅适用于主机模式 屏蔽周期性发送 FIFO 空(Periodic TxFIFO Empty Mask)
位 25	HCHINTMSK	0x0	rw	适用模式：仅适用于主机模式 主机通道中断屏蔽(Host Channels Interrupt Mask)
位 24	PRTINTMSK	0x0	ro	适用模式：仅适用于主机模式 主机端口中断屏蔽(Host Port Interrupt Mask)
位 23：22	保留	0x0	resd	保持默认值。

位 21	INCOMPIPMSK INCOMPISOOUTMSK	0x0	rw	未完成周期性传输屏蔽(Incomplete Periodic Transfer Mask) 适用模式：仅适用于主机模式 未完成同步 OUT 传输屏蔽 (Incomplete Isochronous OUT Transfer Mask) 适用模式：仅适用于设备模式
位 20	INCOMISOINMSK	0x0	rw	适用模式：仅适用于设备模式 未完成同步 IN 传输屏蔽(Incomplete Isochronous IN Transfer Mask)
位 19	OEPTINTMSK	0x0	rw	适用模式：仅适用于设备模式 OUT 端点中断屏蔽(OUT Endpoints Interrupt Mask)
位 18	IEPTINTMSK	0x0	rw	适用模式：仅适用于设备模式 IN 端点中断屏蔽(IN Endpoints Interrupt Mask)
位 17	保留	0x0	rw	保持默认值。
位 16	保留	0x0	resd	保持默认值。
位 15	EOPFMSK	0x0	rw	适用模式：仅适用于设备模式 周期性帧结束中断屏蔽(End of Periodic Frame Interrupt Mask)
位 14	ISOOUTDROPMSK	0x0	rw	适用模式：仅适用于设备模式下的同步 OUT 包丢失中断屏蔽(Device only Isochronous OUT Packet Dropped Interrupt Mask)
位 13	ENUMDONEMSK	0x0	rw	适用模式：仅适用于设备模式 枚举完成中断屏蔽(Enumeration Done Mask)
位 12	USBRSTMSK	0x0	rw	适用模式：仅适用于设备模式 USB 复位中断屏蔽(USB Reset Mask)
位 11	USBSUSPMSK	0x0	rw	适用模式：仅适用于设备模式 USB 挂起中断屏蔽(USB Suspend Mask)
位 10	ERLYSUSPMSK	0x0	rw	适用模式：仅适用于设备模式 早期挂起中断屏蔽(Early Suspend Mask)
位 9：8	保留	0x0	resd	保持默认值。
位 7	GOUTNAKEFFMSK	0x0	rw	适用模式：仅适用于设备模式 全局 OUT NAK 有效中断屏蔽(Global OUT NAK Effective Mask)
位 6	GINNAKEFFMSK	0x0	rw	适用模式：仅适用于设备模式 全局非周期性 IN NAK 有效中断屏蔽(Global Non-periodic IN NAK Effective Mask)
位 5	NPTXFEMPMSK	0x0	rw	适用模式：主机模式和设备模式 非周期性发送 FIFO 空中断屏蔽(Non-periodic TxFIFO Empty Mask)
位 4	RXFLVLMSK	0x0	rw	适用模式：主机模式和设备模式 接收 FIFO 非空中断屏蔽(Receive FIFO Non-Empty Mask)
位 3	SOFMSK	0x0	rw	适用模式：主机模式和设备模式 帧开始中断屏蔽(Start of Frame Mask)
位 2	OTGINTMSK	0x0	rw	适用模式：主机模式和设备模式 OTG 中断屏蔽(OTG Interrupt Mask)。
位 1	MODEMISMSK	0x0	rw	适用模式：主机模式和设备模式 模式不匹配中断屏蔽(Mode Mismatch Interrupt Mask)
位 0	保留	0x0	resd	保持默认值。

21.6.3.8 OTGFS接收状态调试读/OTG状态读和POP寄存器 (OTGFS_GRXSTSR / OTGFS_GRXSTSP)

读取“接收状态调试读寄存器(Receive Status Debug Read)”会返回接收 FIFO(Receive FIFO)顶层的数据。读取“接收状态读和 PoP 寄存器(Receive Status Read and Pop)”还会弹出接收 FIFO 顶层的数据。

在主机模式和设备模式下，对接收状态内容的解释并不相同。当接收 FIFO 为空时，控制器将忽略接收状态弹出/读取，并返回 0x0000 0000。当控制器中断寄存器(Core Interrupt register) 的接收 FIFO 非空位被置起时，应用程序才能弹出接收状态 FIFO。

主机模式：

域	简称	复位值	类型	功能
位 31：21	保留	0x000	resd	保持默认值。
位 20：17	PKTSTS	0x0	ro	数据包状态(Packet Status) 表示接收到的数据包的状态。 0010：接收到 IN 数据包 0011：完成 IN 传输 (触发中断) 0101：数据翻转出错(触发中断) 0111：通道中止(触发中断) 其它：保留 复位值：0
位 16：15	DPID	0x0	ro	数据 PID(Data PID) 表示接收到的数据包的数据 PID 00：DATA0 10：DATA1 01：DATA2 11：MDATA 复位值：0
位 14：4	BCNT	0x000	ro	字节数(Byte Count) 表示已接收到的 IN 数据包的字节数
位 3：0	CHNUM	0x0	ro	通道数(Channel Number) 表示当前接收到的数据包属于哪一个通道

设备模式：

域	简称	复位值	类型	功能
位 31：25	保留	0x00	resd	保持默认值。
位 24：21	FN	0x0	ro	帧号(Frame Number) 表示 USB 总线上所收到的数据包的帧号的最低 4 位。 此位域仅适用于支持同步 OUT 端点功能的情况。
位 20：17	PKTSTS	0x0	ro	数据包状态(Packet Status) 表示接收到的数据包的状态。 0001：全局 OUT NAK(触发中断) 0010：收到 OUT 数据包 0011：完成 OUT 传输(触发中断) 0100：完成 SETUP 任务(触发中断) 0110：收到 SETUP 数据包 其它：保留
位 16：15	DPID	0x0	ro	数据 PID(Data PID) 表示接收到的 OUT 数据包的数据 PID 00：DATA0 10：DATA1 01：DATA2 11：MDATA
位 14：4	BCNT	0x000	ro	字节数(Byte Count) 表示接收到的数据包的字节数
位 3：0	EPTNUM	0x0	ro	端点号(Endpoint Number) 表示当前所收到的数据包属于哪一个端点号

21.6.3.9 OTGFS接收FIFO长度寄存器(OTGFS_GRXFSIZ)

应用程序可以对必须要分配给接收 FIFO 的 SRAM 长度进行设置。

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持默认值。
位 15：0	RXFDEP	0x0200	ro/rw	接收 FIFO 深度(RxFIFO Depth) 此值以 32 位字为单位。 最小值为 16 最大值为 512 在配置期间，本寄存器的上电复位值被定义为接收数据 FIFO 的最大深度。

21.6.3.10 OTGFS非周期性TX FIFO长度寄存器(OTGFS_GNPTXFSIZ)/端点0 TX FIFO长度寄存器(OTGFS_DIEPTXF0)

应用程序可设置非周期性发送 FIFO 的 SRAM 长度和存储器起始地址，该寄存器的位域随主机模式或设备模式而变化。

主机：

域	简称	复位值	类型	功能
位 31：16	NPTXFDEP	0x0000	ro/rw	非周期性发送 FIFO 深度(Non-periodic TxFIFO depth) 这个数值的单位是 32 位的字 最小值是 16 最大值是 256。
位 15：0	NPTXFSTADDR	0x0200	ro/rw	非周期性发送 SRAM 起始地址(Non-periodic Transmit SRAM Start Address) 此位域包含了非周期性发送 FIFO SRAM (Non-periodic Transmit FIFO SRAM)的存储器起始地址。

设备：

域	简称	复位值	类型	功能
位 31：16	INEPT0TXDEP	0x0000	ro/rw	IN 端点发送 FIFO 0 深度(IN Endpoint TxFIFO 0 Depth) 这个数值的单位是 32 位的字 最小值是 16 最大值是 256。
位 15：0	INEPT0TXSTADDR	0x0200	ro/rw	IN 端点 FIFO 0 发送 SRAM 起始地址(IN Endpoint FIFO0 Transmit SRAM Start Address) 此位域包含了 IN 端点发送 FIFO 0 的存储器起始地址

21.6.3.11 OTGFS非周期性TX FIFO/请求队列状态寄存器(OTGFS_GNPTXSTS)

本寄存器仅适用于主机模式，本寄存器为只读，储存了非周期性发送(FIFO Non-periodic TxFIFO) 和非周期性发送请求队列(Non-periodic Transmit Request Queue)的可用空间信息。

域	简称	复位值	类型	功能
位 31	保留	0x0	resd	保持默认值。
位 30：24	NPTXQTOP	0x00	ro	非周期性发送请求队列的顶部(Top of the Non-periodic Transmit Request Queue) 表示 MAC 正在处理非周期性发送请求队列的条目 位[30：27]：通道/端点号 位[26：25]： 00：IN/OUT 令牌 01：零长度的发送包（设备 IN/主机 OUT） 10：PING/CSPLIT 令牌 11：通道中止指令 位[24]：结束（所选通道/端点的最后一个请求）
位 23：16	NPTXQSPCAVAIL	0x08	ro	非周期性发送请求队列的可用空间(Non-periodic Transmit Request Queue Space Available) 表示非周期性发送请求队列的可用空间。在主机模式下，此队列既支持 IN 请求也支持 OUT 请求。 00：非周期性发送请求队列的空间已满 01：1 个位置可用 02：2 个位置可用 n：n 个位置可用(0 ≤ n ≤ 8) 其它：保留 复位值：可设置
位 15：0	NPTXFSPCAVAIL	0x0200	ro	非周期性发送 FIFO 的可用空间(Non-periodic TxFIFO Space Avail) 表示非周期性发送 FIFO 的可用空间，此值以 32 位字节为单位。 00：非周期性发送 FIFO 已满 01：1 个字可用 02：2 个字可用 n：n 个字可用 (0 ≤ n ≤ 256)

				其它：保留 复位值：可设置

21.6.3.12 OTGFS通用控制器配置寄存器(OTGFS_GCCFG)

域	简称	复位值	类型	功能
位 31：22	保留	0x000	resd	保持默认值。
位 21	VBUSIG	0x0	rw	VBUS 忽略（VBUS ignored） 当该控制位被置位，OTGFS 并不监测 VBUS 引脚电压，并且认为在主机和设备模式下，VBUS 电压一直有效，然后可释放 VBUS 引脚作为其他用途。 0：VBUS 不被忽略 1：VBUS 被忽略，并认为 VBUS 电压一直有效
位 20	SOFOUTEN	0x0	rw	SOF 输出使能(SOF output enable) 0：不输出 SOF 脉冲； 1：输出 SOF 脉冲到引脚上。
位 19：18	保留	0x0	resd	保持默认值。
位 17	LP_MODE	0x0	rw	低功耗模式(Low-power mode) 作用：控制 OTG PHY 的功耗。软件置 1 时，OTG PHY 进入低功耗模式；软件置 0 时，为正常模式。 0：非低功耗模式； 1：低功耗模式。
位 16	PWRDOWN	0x0	rw	掉电(Power down) 用于在发送和接收时激活收发器，必需预先配置才能允许 USB 通信 0：使能掉电； 1：禁止掉电(收发器激活)。
位 15：0	保留	0x0000	resd	保持默认值。

21.6.3.13 OTGFS控制器ID寄存器(OTGFS_GUID)

此寄存器只读，保护产品的 ID。

域	简称	复位值	类型	功能
31：0	USERID	0x0000 1000	rw	产品 ID (Product ID field) 应用程序可以编程此 ID 位。

21.6.3.14 OTGFS主机周期性发送FIFO长度寄存器(OTGFS_HPTXFSIZ)

此寄存器保存着周期性发送 FIFO 的深度和存储器起始地址

域	简称	复位值	类型	功能
位 31：16	PTXFSIZE	0x02000	ro/rw	主机周期性发送 FIFO 深度 (Host periodic TxFIFO depth) 此位域的值以 32 位字为单位。 最小值是 16 最大值是 512
位 15：0	PTXFSTADDR	0x0600	ro/rw	主机周期性发送 FIFO 起始地址(Host Periodic TxFIFO Start Address) 该寄存器的上电复位值指的是接收数据 FIFO 最大深度与非周期性发送数据 FIFO 最大深度之和。

21.6.3.15 OTGFS设备IN端点发送FIFO长度寄存器 (OTGFS_DIEPTXFn)(其中n是FIFO的编号，x=1…15)

本寄存器保存着在设备模式下进行的 IN 端点发送 FIFO 的深度以及存储器起始地址。每个 FIFO 包含一个 IN 端点数据。本寄存器可重复用于实例化的 IN 端点 FIFO1~15。通过 GNPTXFSIZ 寄存器来设置 IN 端点 FIFO 0 的深度及内存起始地址。

域	简称	复位值	类型	功能
位 31：16	INEPTXFDEP	0x0200	ro/rw	IN 端点发送 FIFO 深度(IN Endpoint TxFIFO Depth) 以 32 位字为单位。 最小值为 16 最大值为 512 复位值是最大可能的 IN 端点发送 FIFO 的深度。

位 15：0	INEPTXFSTADDR	0x0400	ro/rw	IN 端点 FIFOn 发送 SRAM 起始地址(IN Endpoint FIFOn Transmit SRAM Start Address) 此值为 IN 端点 n 发送 FIFO 在 SRAM 中的起始地址。

21.6.4 主机模式下的寄存器

该寄存器影响主机模式下控制器的运行状况。不支持在设备模式下进行访问（因为设备模式下访问结果未定义）。主机模式下的寄存器分别为：

21.6.4.1 OTGFS主机模式配置寄存器(OTGFS_HCFG)

该寄存器用于上电后配置控制器。初始化之后，不得再更改本寄存器。

域	简称	复位值	类型	功能
位 31：3	保留	0x0000 0000	resd	保持默认值。
位 2	FSLSSUPP	0x0	ro	仅支持全速和低速设备(FS- and LS-Only Support) 应用程序通过此位来控制内核的枚举速度。应用程序可通过此位将控制器以全速主机模式来枚举，即使已连接的设备支持高速通信。在初始化设置之后，不得再更改此位域。 0：全速/低速，具体情况取决于连接设备所支持的最大速度 1：仅支持全速/低速，即使所连接的设备支持高速模式。
位 1：0	FSLSPCLKSEL	0x0	rw	全速/低速 PHY 时钟选择(FS/LS PHY Clock Select) 当控制器处于全速主机模式下： 01：PHY 时钟运行频率为 48MHz 其它值：保留 当控制器处于低速主机模式下： 00：保留； 01：PHY 时钟运行频率为 48MHz。 10：PHY 时钟运行频率为 6MHz。如果选择了 6MHz 时钟，则必须进行软件复位。 11：保留

21.6.4.2 OTGFS主机帧间隔寄存器(OTGFS_HFIR)

该寄存器用于设置当前枚举速度的帧间隔。

域	简称	复位值	类型	功能
位 31：17	保留	0x0000	resd	保持默认值。
位 16	HFIRRLDCTRL	0x0	rw	重新加载控制(Reload Control) 该位允许在运行时对主机帧间隔寄存器进行动态重新加载。 1：不能动态重新加载主机帧间隔寄存器 0：可以在运行时动态重新加载主机帧间隔寄存器 该位需在初始化时设置好，并且在运行期间不得更改其值。
位 15：0	FRINT	0xEA60	rw	帧间隔(Frame Interval) 应用程序通过此位域来设置是两个连续 SOF（全速）之间的间隔。 此位域中的 PHY 时钟个数即表示帧间隔。只有当主机端口控制及状态寄存器的端口使能位被设置后，应用程序才可以写入主机帧间隔寄存器。 如果未设定此位域，控制器将根据主机配置寄存器的 FS/LS PHY 时钟选择位所定义的 PHY 时钟频率来计算其值。初始化配置之后，不得再更改其值。 1 ms * (全速/低速 PHY 时钟频率)

21.6.4.3 OTGFS主机帧号/帧时间剩余寄存器(OTGFS_HFNUM)

该寄存器指示当前帧号，以及当前帧剩余时间（以 PHY 时钟个数表示）。

域	简称	复位值	类型	功能
位 31：16	FTREM	0x0000	ro	帧剩余时间(Frame Time Remaining)

				指示当前帧（全速/低速）还剩余多少时间，以 PHY 时钟个数表示。此位域的值会随每个 PHY 时钟个数而自动递减。当值减为 0 时，帧间隔寄存器的值会重新加载到此位域，并且向 USB 总线发出一个新 SOF。
位 15：0	FRNUM	0x3FFF	ro	帧号(Frame Number) 每当向 USB 总线发出一个新 SOF 时，此位域的值就会递增一次；当值增加到 16'h3FFF 时，此位域即复位为 0。

21.6.4.4 OTGFS主机周期性发送FIFO/请求队列寄存器(OTGFS_HPTXSTS)

该寄存器为只读，包含周期性发送 FIFO 和周期性发送请求队列的剩余空间信息。

域	简称	复位值	类型	功能
位 31：24	PTXQTOP	0x00	ro	周期性发送请求队列的顶层(Top of the Periodic Transmit Request Queue) 表示 MAC 正在处理的周期性发送请求队列的条目。该寄存器用于模块调试。 位[31]：奇数/偶数帧 0：偶数帧发送 1：奇数帧发送 位[30：27]：通道/端点号 位[26：25]：类型 00：IN/OUT 01：零长度包 10：保留 11：禁止通道指令 位[24]：终止（所选通道或端点的最后一个条目）
位 23：16	PTXQSPCAVAIL	0x08	ro	周期性发送请求队列剩余空间(Periodic Transmit Request Queue Space Available) 指示周期性发送请求队列里允许写入的可用空间。此队列包括 IN 和 OUT 请求。 00：周期性发送请求队列满额 01：1 个可用空间 10：2 个可用空间 n：n 个可用空间(0 ≤ n ≤ 8) 其它值：保留
位 15：0	PTXFSPCAVAIL	0x0100	rw	周期性发送数据 FIFO 剩余空间(Periodic Transmit Data FIFO Space Available) 指示周期性发送 FIFO 的可用空间，以 32 位字为单位 0000：周期性发送 FIFO 满额 0001：1 个字可用 0010：2 个字可用 n：n 个字可用(其中 0 ≤ n ≤ 512) 其它值：保留

21.6.4.5 OTGFS主机所有通道中断寄存器(OTGFS_HAINT)

当某个通道产生了标志事件时，主机所有通道中断断寄存器将通过控制器中断寄存器的主机通道中断位来中断应用程序，如图 21-2 所示。每个通道都有一个中断位，共 16 位。应用程序通过设置或清除相应主机通道 n 中断寄存器的相应位来设置或清除此位。

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持默认值。
位 15：0	HAINT	0x0000	ro	通道中断(Channel Interrupts) 每个通道对应一个位：通道 0 对应位 0，通道 15 对应位 15。

21.6.4.6 OTGFS主机所有通道中断屏蔽寄存器(OTGFS_HAINTMSK)

主机所有通道中断屏蔽寄存器与主机所有通道中断寄存器配置使用，用于控制在产生事件时是否打断应用程序。每个通道都有一个相对应的中断屏蔽位，共有 16 位。

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持默认值。
位 15：0	HAINTMSK	0x0000	rw	通道中断屏蔽(Channel Interrupt Mask) 每个通道对应一个位：位 0 对应通道 0，位 15 对应通道 15。

21.6.4.7 OTGFS主机端口控制和状态寄存器(OTGFS_HPRT)

该寄存器仅适用于主机模式。OTG 主机目前仅支持一个端口。

该寄存器包含 USB 端口相关的信息，比如每个端口的 USB 复位，使能，挂起，唤醒，连接状态以及测试模式等信息，如图 21-2 所示。类型为 rw1c 的寄存器可以通过控制器中断寄存器的主机端口中断位来中断应用程序。端口发生中断时，应用程序必须读取该寄存器，并且清除导致该中断的位。类型为 rw1c 的寄存器，应用程序需要写 1 来清除中断。

域	简称	复位值	类型	功能
位 31：19	保留	0x0000	resd	保持默认值。
位 18：17	PRTSPD	0x0	ro	端口速度(Port Speed) 表示连上端口的设备的速度。 00：保留 01：全速 10：低速 11：保留
位 16：13	PRTTSTCTL	0x0	rw	端口测试控制(Port Test Control) 应用程序通过向此位域写入一个非零值而将此端口置于测试模式，且端口会发出相应信号。 0000：测试模式关闭 0001：Test_J 模式 0010：Test_K 模式 0011：Test_SE0_NAK 模式 0100：Test_Packet 模式 0101：Test_Force_Enable 其它值：保留
位 12	PRTPWR	0x0	rw	端口供电(Port Power) 应用程序通过此位来控制对该端口的供电（写 1 或写 0）。 0：断电 1：供电 注意：该位与接口无关。应用程序需按照编程手册中的主机编程流程为各个接口设置此位。
位 11：10	PRTLNSTS	0x0	ro	端口线状态(Port Line Status) 表示当前 USB 数据线的逻辑状态。 位[10]：D+的逻辑电平 位[11]：D－的逻辑电平
位 9	保留	0x0	resd	保持默认值。
位 8	PRTRST	0x0	rw	端口复位(Port Reset) 当应用程序置起此位后，端口会启动复位序列。应用程序必须计算此次复位序列所需时间，并在复位结束后清除此位。 0：端口不复位 1：端口复位 应用程序必须使该位保持置起状态至少达到 USB2.0 规范第 7.1.7.5 章节规定的最短持续时间才能启动端口复位。除了最短持续时间外，应用程序在清除此位之前还可以额外增加 10ms，USB 规范没有设定最长持续时间。
位 7	PRTSUSP	0x0	rw1s	端口挂起(Port Suspend) 通过将该位置起，应用程序可使端口进入挂起模式，此时，控制器只停止发送 SOF。应用程序必须通过设置端口时钟停止位才能关闭 PHY 时钟。 读取该位将返回当前端口的挂起状态。 当控制器器检测到远程唤醒信号时会清除此位，或者当应用程序将该寄存器的端口复位位或端口唤醒位置起，或者

				将控制器中断寄存器的唤醒/远程唤醒检测中断或断开连接检测中断位置起后，也会清除此位。 即便设备未与主机相连，控制器仍然可以清除此位。 0：端口未处于挂起模式 1：端口处于挂起模式
位 6	PRTRES	0x0	rw	端口唤醒(Port Resume) 应用程序通过设置此位来驱动端口发出唤醒信号。控制器会持续触发唤醒信号直到应用程序清除此位。如果控制器检测到 USB 远程唤醒序列（按照控制器中断寄存器的端口唤醒/远程唤醒检测中断位的指示），控制器将不受应用程序干预自动触发唤醒信号。 读取此位时将返回控制器是否正在驱动唤醒信号的信息。 0：有触发唤醒 1：触发唤醒
位 5	PRTOVRCCHNG	0x0	rw1c	端口过流状态改变(Port Overcurrent Change) 当该寄存器的端口过电流有效位(位 4)的状态发生改变时，控制器将置起此位。只能通过控制器来设置此位，应用程序必须写 1 才能清除此位。
位 4	PRTOVRCACT	0x0	ro	端口过流有效位(Port Overcurrent Active) 表示端口的过流状况 0：不存在过电流 1：存在过电流
位 3	PRTENCHNG	0x0	rw1c	端口使能/禁止状态改变(Port Enable/Disable Change) 当该寄存器的端口使能位 2 的状态发生改变时，控制器将此位置起。只能通过控制器将该位置起，应用程序必须写 1 才能清除此位。
位 2	PRTENA	0x0	rw1c	端口使能(Port Enable) 此端口只有在复位序列后才能被控制器使能。在发生过电流，断开连接或者应用程序将此位清除时，即禁用此端口。应用程序不能通过写入寄存器来设置此位，只能通过清除此位来禁用端口。该位不会触发中断。 0：端口禁用 1：端口使能
位 1	PRTCONDET	0x0	rw1c	端口连接检测(Port Connect Detected) 控制器在检测到设备连接端口时，会通过控制器中断寄存器的主机端口中断位来设置该位。该位只能由控制器设置，应用程序必须写 1 才能清除此位。
位 0	PRTCONSTS	0x0	ro	端口连接状态(Port Connect Status) 0：没有设备连接端口 1：设备连接端口

21.6.4.8 OTGFS主机通道x特性寄存器(OTGFS_HCCHARx)(此处x代码通道号，x = 0...15)

域	简称	复位值	类型	功能
位 31	CHENA	0x0	rw1s	通道使能(Channel Enable) 此位由应用程序设置，并由 OTG 主机清除。 0：通道禁止 1：通道使能
位 30	CHDIS	0x0	rw1s	通道禁止(Channel Disable) 应用程序设置此位来停止发送/接收通道上的数据，即便通道传输还没完成。应用程序必须等到通道禁止中断产生，才视该通道已禁止。
位 29	ODDFRM	0x0	rw	奇数帧(Odd Frame) 应用程序通过设置/复位此位来指示 OTG 主机是否以奇数帧来传输。此位仅适用于周期性（同步和中断）传输。 0：偶数帧 1：奇数帧
位 28：22	DEVADDR	0x00	rw	设备地址(Device Address) 此位域用于选择可作为数据源或接收器的指定设备。
位 21：20	MC	0x0	rw	多次计数 Multi Count (MC)

				此位域通知主机该周期性端口的每个帧要执行的传输数目。 00：保留，此位域生成未定义的结果 01：1 个事务 10：每个帧有 2 个事务 11：每个帧有 3 个事务 此位域至少应该设置为 0x01。
位 19：18	EPTYPE	0x0	rw	端点类型(Endpoint Type) 表示所选择的传输类型。 00：控制传输 01：同步传输 10：批传输 11：中断传输
位 17	LSPDDEV	0x0	rw	低速设备(Low-Speed Device) 应用程序通过设置此位来指示该通道正在与低速设备通信。
位 16	保留	0x0	resd	保持默认值。
位 15	EPTDIR	0x0	rw	端点方向(Endpoint Direction) 指示传输是 IN 还是 OUT 0：OUT 1：IN
位 14：11	EPTNUM	0x0	rw	端点号(Endpoint Number) 指示设备（用作数据源或接收器）的端点号。
位 10：0	MPS	0x000	rw	最大包长度(Maximum Packet Size) 指示相应端点的最大包长度。

21.6.4.9 OTGFS主机通道x中断寄存器(OTGFS_HCINTx)(其中x代表通道号，x=0...15)

该寄存器包含着与 USB 和 AHB 相关事件的通道状态，如图 21-2 所示。当控制器中断寄存器的主机通道中断位置起时，应用程序必须读取该寄存器。在读寄存器之前，应用程序必须先读取主机所有通道中断寄存器以获理主机通道 n 中断寄存器的确切通道编号。应用程序必须通过清除该寄存器的相应位来清除 OTGFS_HAINT 和 OTGFS_GINTSTS 寄存器的相应位。

域	简称	复位值	类型	功能
位 31：11	保留	0x000000	resd	保持默认值。
位 10	DTGLERR	0x0	rw1c	数据翻转错误(Data Toggle Error) 只能由控制器设置此位，应用程序通过写 1 来清除此位。
位 9	FRMOVRUN	0x0	rw1c	帧溢出(Frame Overrun) 只能由控制器设置此位，应用程序通过写 1 来清除此位。
位 8	BBLERR	0x0	rw1c	Babble Error 只能由控制器设置此位，应用程序通过写 1 来清除此位。
位 7	XACTERR	0x0	rw1c	传输错误(Transaction Error) 表示 USB 总线发生了下列错误： CRC 校验失败 传输超时 位填充错误 EOP 错误 只能由控制器设置此位，应用程序通过写 1 来清除此位。
位 6	保留	0x0	resd	保持默认值。
位 5	ACK	0x0	rw1c	ACK 响应已收到/已发送中断(ACK Response Received/Transmitted Interrupt) 只能由控制器设置此位，应用程序通过写 1 来清除此位。
位 4	NAK	0x0	rw1c	NAK 响应已收到中断(NAK Response Received Interrupt) 只能由控制器设置此位，应用程序通过写 1 来清除此位。
位 3	STALL	0x0	rw1c	STALL 响应已收到中断(STALL Response Received Interrupt) 只能由控制器设置此位，应用程序通过写 1 来清除此位。
位 2	保留	0x0	resd	保持默认值。
位 1	CHHLTD	0x0	rw1c	通道中止(Channel Halted)

				此位指示传输异常结束，原因是 USB 传输出错或者为响应应用程序因传输完成而发出的中止请求。
位 0	XFERC	0x0	rw1c	传输完成(Transfer Completed) 传输正常结束，未报错。此位只能由控制器设置，应用程序通过写 1 来清除此位。

21.6.4.10 OTGFS主机通道x中断屏蔽寄存器(OTGFS_HCINTMSKx)(其中x为通道号，x=0...15)

本寄存器用于屏蔽上一节所描述的各类通道中断。

域	简称	复位值	类型	功能
位 31：11	保留	0x000000	resd	保持默认值。
位 10	DTGLERRMSK	0x0	rw	数据翻转错误屏蔽(Data Toggle Error Mask)
位 9	FRMOVRUNMSK	0x0	rw	帧溢出屏蔽(Frame Overrun Mask)
位 8	BBLERRMSK	0x0	rw	Babble 错误屏蔽(Babble Error Mask)
位 7	XACTERRMSK	0x0	rw	传输错误屏蔽(Transaction Error Mask)
位 6	NYETMSK	0x0	rw	NYET 响应已收到中断屏蔽(NYET Response Received Interrupt Mask)
位 5	ACKMSK	0x0	rw	ACK 响应已收到/已发送中断屏蔽(ACK Response Received/Transmitted Interrupt Mask)
位 4	NAKMSK	0x0	rw	NAK 响应已收到中断屏蔽(NAK Response Received Interrupt Mask)
位 3	STALLMSK	0x0	rw	STALL 响应已收到中断屏蔽(STALL Response Received Interrupt Mask)
位 2	保留	0x0	resd	保持默认值。
位 1	CHHLTDMSK	0x0	rw	通道中止屏蔽(Channel Halted Mask)
位 0	XFERCMSK	0x0	rw	传输完成屏蔽(Transfer Completed Mask)

21.6.4.11 OTGFS主机通道x传输长度寄存器(OTGFS_HCTSIZx)(其中x为通道号，x=0...15)

域	简称	复位值	类型	功能
位 31	保留	0x0	resd	保持默认值。
位 30：29	PID	0x0	rw	PID (Pid) 应用程序在此位域设置了用于首次传输的 PID 类型。主机控制着此位域用于后续的传输任务。 00：DATA0 01：DATA2 10：DATA1 11：MDATA(非控制)/SETUP(控制)
位 28：19	PKTCNT	0x000	rw	包数目(Packet Count) 应用程序在此位域设置了期望发送或期望接收的包数目。主机在每次成功发送/接收 OUT/IN 包时会递减包数目。当数据减为 0 时，应用程序将收到中断以指示传输正常结束。
位 18：0	XFERSIZE	0x00000	rw	传输长度(Transfer Size) 对于 OUT 传输来说，此位域指示主机在传输过程中发送的数据字节数。 对于 IN 传输来说，此位域指示应用程序为传输预留的缓冲区长度。 对于 IN 传输（周期性和非周期性），应用程序需要将此位域设置为最大包长度的整数倍。

21.6.5 设备模式下的寄存器

这些寄存器仅用于设备模式。主机模式不允许访问，因为访问结果未知。其中有些寄存器会影响到所有端点，而有些寄存器只影响某一个端点。

21.6.5.1 OTGFS设备配置寄存器(OTGFS_DCFG)

在复位后、特殊控制命令后或者枚举后，此寄存器用于配置设备模式下的控制器。在初始化之后，请不要再修改此寄存器。

域	简称	复位值	类型	功能
位 31：13	保留	0x04100	resd	保持默认值。
位 12：11	PERFRINT	0x0	rw	周期性帧间隔(Periodic frame interval) 此位域表示产生“周期性帧结束中断”位时占某个帧内的时间比。应用程序可以通过此中断来确认帧内的同步传输是否已完成。 00：80%的帧时间； 01：85%的帧时间； 10：90%的帧时间； 11：95%的帧时间。
位 10：4	DEVADDR	0x00	rw	设备地址(Device address) 应用程序在每次收到设置地址（SetAddress）控制命令后必须设置此位域。
位 3	保留	0x0	resd	保持默认值。
位 2	NZSTSOUTHSHK	0x0	rw	非零长度的状态 OUT 握手信号(Non-zero-length status OUT handshake) 在控制传输的状态阶段，如果收到了非零长度的数据包，控制器会发出握手信号，应用程序正是通过此位来选择控制器要发送的握手信号。 1：向非零长度的状态 OUT 传输发送一个 STALL 握手信号，但不向应用程序传送所收到的 OUT 数据包 0：向应用程序传递所收到的 OUT 数据包（要么零长度，要么非零长度），并且根据设备端点控制寄存器的 NAK 和 STALL 位选择发送握手信号。
位 1：0	DEVSPD	0x0	rw	设备速度(Device speed) 此位域指示的是应用程序需要控制器进行枚举的速度，或者应用程序可支持的最大速度。然而，总线实际速度只能在整个序列完成之后，才能根据控制器与 USB 主机相连的速度来决定。 00：保留； 01：保留； 10：保留； 11：全速(USB1.1 收发器，时钟为 48MHz)。

21.6.5.2 OTGFS设备控制寄存器(OTGFS_DCTL)

域	简称	复位值	类型	功能
位 31：12	保留	0x00000	resd	保持默认值。
位 11	PWROPRGDNE	0x0	wo	上电配置结束(Power-on programming done) 应用程序通过此位来指示从断电模式唤醒后，完成了对该寄存器的配置。
位 10	CGOUTNAK	0x0	wo	清除全局 OUT NAK (Clear global OUT NAK) 对此位进行写操作可以清除全局 OUT NAK 状态。
位 9	SGOUTNAK	0x0	wo	设置全局 OUT NAK (Set global OUT NAK) 对此位进行写操作可以设置全局 OUT NAK 状态。 应用程序通过此位向所有 OUT 端点发送一个 NAK 握手信号。只有在确认已清除了控制器中断寄存器的全局 OUT NAK 有效位时，应用程序才需要设置此位。
位 8	CGNPINNAK	0x0	wo	清除全局非周期性 IN NAK (Clear Global Non-periodic IN NAK) 对此位写操作可以清除全局非周期性 IN NAK 状态。
位 7	SGNPINNAK	0x0	wo	设置全局非周期性 IN NAK (Set global Non-periodic IN NAK) 对此位写操作可以设置全局非周期性 IN NAK 状态。 应用程序通过此位向所有非周期性 IN 端点发送一个 NAK 握手信号。应用程序只有在确认了控制器中断寄存器的全局 IN NAK 有效信号已清除的情况下才设置此位。
位 6：4	TSTCTL	0x0	rw	测试控制(Test control) 000：测试模式禁止； 001：Test_J 模式； 010：Test_K 模式；

				011：Test_SE0_NAK 模式； 100：Test_Packet 模式； 101：Test_Force_Enable； 其他：保留。
位 3	GOUTNAKSTS	0x0	ro	全局 OUT NAK 状态 (Global OUT NAK status) 0：根据 FIFO 状态，NAK 和 STALL 位的设定来发送握手信号 1：不管是否有可用空间，都不会写数据到接收 FIFO。向所有数据包（除了 SETUP 传输）发送一个 NAK 握手信号。丢弃所有同步 OUT 数据包。
位 2	GNPINNAKSTS	0x0	ro	全局非周期性 IN NAK 状态 (Global Non-periodic IN NAK status) 0：根据发送 FIFO 的数据状况来发送握手信号 1：不管发送 FIFO 内的数据状况如何，都会向所有的非周期性 IN 端点发送 NAK 握手信号。
位 1	SFTDISCON	0x1	rw	软件断开(Soft disconnect) 应用程序通过此位指示 OTGFS 控制器进行“软件断开”操作。一旦置起此位，那么主机会看到设备已断开，设备也收不收到 USB 总线信号。控制器就始终处于断开状态直至应用程序清除此位。 0：普通操作。当此位在设备软件断开后清除，控制器会向主机发起一个设备连接事件。USB 主机与设备重新连接后，会重启设备枚举。
位 0	RWKUPSIG	0x0	rw	远程唤醒信号(Remote wakeup signaling) 应用程序置起此位后，控制器会启动远程信号来唤醒 USB 主机。应用程序必须置起此位来指示控制器退出挂起状态。 按照 USB2.0 的规定，应用程序在设置此位 1－15 ms 后必须清除该位。

为了使 USB 主机检测到设备断开，软件断开位在各种状态下保持置起的最短持续时间在下表给出。 为了适应时钟抖动，建议应用程序在规定的最短持续时间的基础上再添加一些额外的延迟。

表 21-5 软件断开的最短持续时间

操作速度	设备状态	最短持续时间
全速	挂起	1ms + 2.5us
全速	空闲	2.5us
全速	不空闲或挂起(正在执行传输操作)	2.5us

21.6.5.3 OTGFS设备状态寄存器(OTGFS_DSTS)

此寄存器指示控制器与 OTGFS 相关的状态。此寄存器用于在发生设备所有中断(OTGFS_DAINT)寄存器事件时读取。

域	简称	复位值	类型	功能
位 31：22	保留	0x000	resd	保持默认值。
位 21：8	SOFFN	0x0000	ro	接收到的 SOF 的帧号(Frame number of the received SOF) 注意：如果在上电复位后立即读取此位域，可能返回非零值。如果在上电复位后立即读取此位域时返回的是非零值，并不表示已经收到主机发出的 SOF。只有当主机和设备进行连接后，此位域的读取值才是有效的。
位 7：4	保留	0x1	resd	保持默认值。
位 3	ETICERR	0x0	ro	随机误差(Erratic error) 此类错误会导致控制器挂起，并且控制器中断寄存器的早期挂起位会置起，随后产生中断。如果是因为异常错误而触发控制器早期挂起，那么应用程序只能执行软件断开进行修复解决。
位 2：1	ENUMSPD	0x0	ro	枚举速度(Enumerated speed)

				表示控制器在通过序列进行速度检测后所确定的执行速度。 01：保留； 10：保留； 11：全速(PHY 时钟运行在 48MHz)； 其他：保留。
位 0	SUSPSTS	0x0	ro	挂起状态(Suspend status) 在设备模式下，只要检测到 USB 总线上有挂起条件，此位就会置起。当 USB 总线信号长时间不活动时，控制器即会进入挂起状态。 控制器在遇到下列情况时会退出挂起状态： ■USB 总线信号开始活动 ■应用程序对设备控制寄存器的远程唤醒信号位进行写操作

21.6.5.4 OTGFS设备OTGFSIN端点通用中断屏蔽寄存器 (OTGFS_DIEPMSK)

本寄存器与各个端点的设备 IN 端点中断寄存器配合工作以产生 IN 端点中断。可以通过向设备 IN 端点通用中断屏蔽寄存器的相应位进行写操作来屏蔽设备 IN 端点中断寄存器(OTGFS_DIEPINTx)中某个特定状态的 IN 端点中断。状态位是默认屏蔽的。

域	简称	复位值	类型	功能
位 31：10	保留	0x000000	resd	保持默认值。
位 9	BNAINMSK	0x0	rw	BNA 中断屏蔽(BNA interrupt mask) 0：中断屏蔽； 1：中断不屏蔽。
位 8	TXFIFOUDRMSK	0x0	rw	FIFO 欠载中断屏蔽(FIFO underrun mask) 0：中断屏蔽； 1：中断不屏蔽。
位 7	保留	0x0	resd	保持默认值。
位 6	INEPTNAKMSK	0x0	rw	IN 端点 NAK 状态有效中断屏蔽(IN endpoint NAK effective mask) 0：中断屏蔽； 1：中断不屏蔽。
位 5	INTKNEPTMISMSK	0x0	rw	端点收到 IN 命令不匹配中断屏蔽(IN token received with EP mismatch mask) 0：中断屏蔽； 1：中断不屏蔽。
位 4	INTKNTXFEMPMSK	0x0	rw	当发送 FIFO 空时收到 IN 命令中断屏蔽(IN token received when TxFIFO empty mask) 0：中断屏蔽； 1：中断不屏蔽。
位 3	TIMEOUTMSK	0x0	rw	超时条件中断屏蔽(非同步端点) (Timeout condition mask (Non-isochronous endpoints)) 0：中断屏蔽； 1：中断不屏蔽。
位 2	保留	0x0	resd	保持默认值。
位 1	EPTDISMSK	0x0	rw	端点禁用中断屏蔽 (Endpoint disabled interrupt mask) 0：中断屏蔽； 1：中断不屏蔽。
位 0	XFERCMSK	0x0	rw	传输完成中断屏蔽(Transfer completed interrupt mask) 0：中断屏蔽； 1：中断不屏蔽。

21.6.5.5 OTGFS设备OUT端点通用中断屏蔽寄存器 (OTGFS_DOEPMSK)

此寄存器配合设备 OUT 端点中断寄存器(OTGFS_DOEPINTx)使用，产生 OUT 端点中断。OTGFS_DOEPINTx 寄存器的每一位都可以通过写此寄存器的相应位来屏蔽。在默认状态下，所有中断都屏蔽。

域	简称	复位值	类型	功能

位 31：10	保留	0x000000	resd	保持默认值。
位 9	BNAOUTMSK	0x0	rw	BNA 中断屏蔽 (BNA interrupt mask) 0：中断屏蔽； 1：中断不屏蔽。
位 8	OUTPERRMSK	0x0	rw	OUT 包错误中断屏蔽 (OUT packet error mask) 0：中断屏蔽； 1：中断不屏蔽。
位 7	保留	0x0	resd	保持默认值。
位 6	B2BSETUPMSK	0x0	rw	收到连续的 SETUP 包中断屏蔽 (Back-to-back SETUP packets received mask) 0：中断屏蔽； 1：中断不屏蔽。
位 5	保留	0x0	resd	保持默认值。
位 4	OUTTEPDMSK	0x0	rw	当端点被禁止时收到 OUT 命令中断屏蔽 (OUT token received when endpoint disabled mask) 0：中断屏蔽； 1：中断不屏蔽。
位 3	SETUPMSK	0x0	rw	SETUP 阶段完成中断屏蔽 (SETUP phase done mask) 仅对控制端点有效 0：中断屏蔽； 1：中断不屏蔽。
位 2	保留	0x0	resd	保持默认值。
位 1	EPTDISMSK	0x0	rw	端点被禁止中断屏蔽 (Endpoint disabled interrupt mask) 0：中断屏蔽； 1：中断不屏蔽。
位 0	XFERCMSK	0x0	rw	传输结束中断屏蔽 (Transfer completed interrupt mask) 0：中断屏蔽； 1：中断不屏蔽。

21.6.5.6 OTGFS设备所有端点中断寄存器(OTGFS_DAINT)

当某个端点发生事件时，设备所有端点中断寄存器可以通过设置控制器中断寄存器的设备 IN 端点中断位或设备 OUT 端点中断位来中断应用程序。每个端点有一个中断位，其中 OUT 端点和 IN 端点各自拥有多达 8 个中断位。对于双向端点而言，同时使用对应的 IN 中断位和 OUT 中断位。应用程序通过设置和清除对应的设备端点 x 中断寄寄器来设置和清除设备所有端点中断寄存器的相应位。

域	简称	复位值	类型	功能
位 31：24	保留	0x0000	resd	保持默认值。
位 23：16	OUTEPTINT	0x0000	ro	OUT 端点中断位 (OUT endpoint interrupt bits) 每个位对应一个 OUT 端点。位 16 对应 OUT 端点 0，位 18 对应 OUT 端点 2。
位 15：8	保留	0x0000	resd	保持默认值。
位 7：0	INEPTINT	0x0000	ro	IN 端点中断位 (IN endpoint interrupt bits) 每个位对应一个 IN 端点。位 0 对应 IN 端点 0，位 7 对应 IN 端点 7。

21.6.5.7 OTGFS所有端点中断屏蔽寄存器(OTGFS_DAINTMSK)

当某个设备端点发生事件时，设备端点中断屏蔽寄存器与设备端点中断寄存器共同配合来中断应用程序。尽管如此，与该中断相对应的设备所有端点中断寄存器仍是置起状态。

域	简称	复位值	类型	功能
位 31：24	保留	0x0000	resd	保持默认值。
位 23：16	OUTEPTMSK	0x0000	rw	OUT 端点中断屏蔽寄存器 (OUT EP interrupt mask bits) 每个位对应一个 OUT 端点。 位 16 对应 OUT 端点 0，位 18 对应 OUT 端点 2。 0：屏蔽中断； 1：不屏蔽中断。
位 15：8	保留	0x0000	resd	保持默认值。
位 7：0	INEPTMSK	0x0000	rw	IN 端点中断屏蔽位 (IN EP interrupt mask bits) 每个位对应一个 IN 端点。 位 0 对应 IN 端点 0，位 7 对应 IN 端点 7。 0：屏蔽中断；

				1：不屏蔽中断。

21.6.5.8 OTGFS设备IN端点FIFO空中断屏蔽寄存器(OTGFS_DIEPEMPMSK)

此寄存器配合 IN 端点 FIFO 空中断寄存器(TXFE_OTGFS_DIEPINTx)产生中断。

域	简称	复位值	类型	功能
位 31：8	保留	0x0000	resd	保持默认值。
位 7：0	INEPTXFEMSK	0x0000	rw	IN 端点发送 FIFO 空中断屏蔽位 (IN EP Tx FIFO empty interrupt mask bits) 此位域是设备 IN 端点中断寄存器的屏蔽位。 一个发送 FIFO 空中断位对应一个端点：位 0 对应 IN 端点 0，位 7 对应 IN 端点 7。 0：屏蔽中断； 1：不屏蔽中断。

21.6.5.9 OTGFS设备控制IN端点0控制寄存器(OTGFS_DIEPCTL0)

本节介绍了控制 IN 端点 0 控制寄存器。非零控制端点使用端点 1-7 寄存器。

域	简称	复位值	类型	功能
位 31	EPTENA	0x0	rw1s	端点使能 (Endpoint enable) 此应用程序设置此位，以启动在端点 0 上的数据传输。控制器在产生以下端点中断前会先清除此位： ■端点禁止； ■传输完成。
位 30	EPTDIS	0x0	ro	端点禁用 (Endpoint disable) 应用程序通过设置此位可以在完成传输前停止端点上的数据传输。应用程序必须要等到端点禁用中断产生才认为端点已禁用。控制器在生成端点禁用中断之前先清除此位。只有端点使能的情况下，应用程序才需要置起此位。
位 29：28	保留	0x0	resd	保持默认值。
位 27	SNAK	0x0	wo	设置 NAK (Set NAK) 通过对该位写操作可以设置端点的 NAK 位。应用程序可通过此位来控制端点上的 NAK 握手信号的发送。当该端点接收到 SETUP 数据包时，控制器也会置起此位。
位 26	CNAK	0x0	wo	清除 NAK (Clear NAK) 对该位写操作可清除端点的 NAK 位。
位 25：22	TXFNUM	0x0	rw	发送 FIFO 的编号 (TxFIFO number) 控制端点 0 只能使用 FIFO0
位 21	STALL	0x0	rw1s	STALL 握手 (STALL handshake) 应用程序置起此位，控制器会在收到 SETUP 令牌时清除此位。如果该位连同 NAK 位，全局非周期性 IN NAK 位或者全局 OUT NAK 位同时置起时，STALL 位享有优先级。
位 20	保留	0x0	resd	保持默认值。
位 19：18	EPTYPE	0x0	ro	端点类型 (Endpoint type) 对于控制端点，由硬件置为 00。
位 17	NAKSTS	0x0	ro	NAK 状态 (NAK status) 表示为： 0：表示控制器根据 FIFO 状态发送非 NAK 握手信号 1：表示控制器正在发送 NAK 握手信号。 当此位置起时（无论是应用程序设置的还是控制器设置的），控制器都会停止发送数据，即使发送 FIFO 有可用空间。不管该位是否置起，控制器始终以 ACK 握手信号来响应 SETUP 数据包。
位 16	保留	0x0	resd	保持默认值。
位 15	USBACEPT	0x1	ro	USB 有效端点(USB active endpoint) 此位始终置 1，表示不管在什么配置和接口中，控制端点 0 始终有效。
位 14：2	保留	0x0000	resd	保持默认值。
位 1：0	MPS	0x0	rw	适用 IN 端点和 OUT 端点。

				应用程序通过此位设置当前逻辑端点的最大数据包长度。 00：64 字节 01：32 字节 10：16 字节 11：8 字节

21.6.5.10 OTGFS设备IN端点x控制寄存器(OTGFS_DIEPCTLx)(其中x为端点号，x=1…7)

应用程序通过这些寄存器控制非 0 端点的操作特性。

域	简称	复位值	类型	功能
位 31	EPTENA	0x0	rw1s	端点使能 (Endpoint enable) 应用程序设置此位，表示该端点准备发送数据 控制器在产生以下端点中断之前会先清除该位： ■SETUP 阶段完成 ■端点禁用 ■传输完成
位 30	EPTDIS	0x0	rw1s	端点禁用 (Endpoint disable) 应用程序可通过设置此位停止某个端点上的数据发送/传输，即使传输还没完成。 应用程序需要等到端点禁用中断产生才视为该端点已被禁用。控制器在设置端点禁用中断前会清除此位。应用程序只有在端点已经使用端点时才能设置此位。
位 29	SETD1PID/ SETODDFR	0x0	wo	设置 DATA1 PID (Set DATA1 PID) 仅适用于中断/批量 IN 端点，写此位可以将该寄存器的端点数据 PID 位设置为 DATA1。 设置奇数帧 (Set odd frame) 仅适用于同步 IN 端点，写此位可以将奇偶帧位设置为奇数帧。 0：关闭 Set DATA1 PID 或者不强制奇数帧 1：使能 Set DATA1 PID 或者强制奇数帧
位 28	SETD0PID/ SETEVENFR	0x0	rw	设置 DATA0 PID (Set DATA0 PID) 仅适用于中断/批量 IN 端点，写此位可将该寄存器的端点数据 PID 位设置为 DATA0。 设置偶数帧 (Set Even frame) 仅适用于同步 IN 端点，写此位可以将偶数帧/奇数帧位设置为偶数帧。 0：关闭 Set DATA0 PID 或不强制偶数帧 1：端点数据 PID 设置为 DATA0 或将 EOFRNUM 设置为偶数帧
位 27	SNAK	0x0	wo	设置 NAK (Set NAK) 通过对该位写操作可以设置端点的 NAK 位。应用程序可通过此位来控制端点上的 NAK 握手信号的发送。控制器在接收到 SETUP 数据包或传输结束中断时置起此位。 数值: 0：未设置 NAK 1：设置 NAK
位 26	CNAK	0x0	wo	清除 NAK (Clear NAK) 通过对该位写操作可以清除端点的 NAK 位。 0：不清除 NAK 1：清除 NAK
位 25：22	TXFNUM	0x0	rw	发送 FIFO 的编号 (TxFIFO number) 给对应端点分配 FIFO 编号，必须给每个有效的 IN 端点分配一个独立的 FIFO 编号。此位仅对 IN 端点有效。
位 21	STALL	0x0	rw	STALL 握手 (STALL handshake) 适用于非控制非同步 IN 端点和 OUT 端点。 应用程序通过此位停止向该端点发送 USB 主机的令牌。 如果 NAK 位、全局非周期性 IN NAK 位或者全局 OUT

				NAK 位和该位同时置起时，STALL 位享有优先级。只有应用程序才能清除此位，控制器不能。 0：停止发送所有无效令牌 1：停止发送所有有效令牌
位 20	保留	0x0	resd	保持默认值。
位 19：18	EPTYPE	0x0	rw	端点类型 (Endpoint type) 表示逻辑端点支持的传输类型 00：控制； 01：同步； 10：块传输； 11：中断。
位 17	NAKSTS	0x0	ro	NAK 状态 (NAK status) 指示以下状态： 0：表示控制器根据 FIFO 状态发送非 NAK 握手信号 1：表示控制器正在发送 NAK 握手信号。 当此位置起时（无论是应用程序设置的还是控制器设置的） ■控制器停止接收 OUT 端点上的数据，即使接收 FIFO 有剩余空间可以用来容纳传入的数据包。 ■对于非同步 IN 端点：控制器停止发送端点上的数据，即使发送 FIFO 还有待发送的数据。 ■对于同步 IN 端点：控制器会发出一个零长度的数据包，即使发送 FIFO 还有剩余空间。 不管该位是否置起，控制器始终以 ACK 握手信号来响应 SETUP 数据包。
位 16	DPID/ EOFRNUM	0x0	ro	端点数据 PID (Endpoint data PID) 仅适用于中断/批量 OUT 端点。 此位包含的是该端点上要传输或要发送的数据包的 PID 信息。在端点使能后，应用程序需要设置该端点待发送或接收的首个数据包的 PID。应用程序通过该寄存器的 SetD1PID 和 SetD0PID 来设置 DATA0 或 DATA1 PID。 0：DATA0 1：DATA1 奇/偶数帧 (Even/odd frame) 仅适用于同步 OUT 端点。 表示控制器传输或接收该端点的同步数据的帧号。应用程序通过该寄存器的 SETEVNFR 和 SETODDFR 设置希望传输或接收同步数据的偶数帧号/奇数帧号。 0：偶数帧 1：奇数帧
位 15	USBACEPT	0x0	rw	USB 活跃端点 (USB active endpoint) 指示在当前配置和接口下，该端点是否活跃。控制器在检测到 USB 复位之后会清除此位(除了端点 0)。 应用程序在收到 SetConfiguration 和 SetInterface 命令时必须设置端点寄存器并置起此位。 0：不活跃 1：活跃
位 14：11	保留	0x0	resd	保持默认值。
位 10：0	MPS	0x000	rw	最大包长度 (Maximum packet size) 应用程序通过此位设置当前逻辑端点的最大包长度，以字节为单位。

21.6.5.11 OTGFS设备控制OUT端点0控制寄存器(OTGFS_DOEPCTL0)

本节介绍了控制 OUT 端点 0 控制寄存器。非零控制端点使用端点 1-7 寄存器。

域	简称	复位值	类型	功能
位 31	EPTENA	0x0	rw1s	端点使能 (Endpoint enable) 应用程序设置此位，以启动端点 0 上的数据传输。 控制器在产生以下端点中断之前会先清除该位： ■SETUP 阶段完成；

				■端点禁止； ■传输完成。
位 30	EPTDIS	0x0	ro	端点禁用 (Endpoint disable) 应用程序不能禁止控制 OUT 端点 0
位 29：28	保留	0x0	resd	保持默认值。
位 27	SNAK	0x0	wo	设置 NAK (Set NAK) 通过对该位写操作可以设置端点的 NAK 位。应用程序可通过此位来控制端点上的 NAK 握手信号的发送。控制器在接收到 SETUP 数据包或传输结束中断时置起此位。
位 26	CNAK	0x0	wo	清除 NAK (Clear NAK) 通过对该位写操作可以清除端点的 NAK 位。
位 25：22	保留	0x0	resd	保持默认值。
位 21	STALL	0x0	rw1s	STALL 握手 (STALL handshake) 应用程序置起此位，控制器会在收到 SETUP 令牌时清除此位。如果该位连同 NAK 位，全局非周期性 OUT NAK 位同时置起时，STALL 位享有优先级。 不管该位是否置起，控制器始终以 ACK 握手信号来响应 SETUP 数据包。
位 20	SNP	0x0	rw	监听模式 (Snoop mode) 此位将端点配置为 Snoop 模式。在 Snoop 模式下，控制器在将 OUT 包传输给应用存储器之前不会检查 OUT 包是否正确。
位 19：18	EPTYPE	0x0	ro	端点类型 (Endpoint type) 硬件设置将此位设置为 0 用于控制端点类型。
位 17	NAKSTS	0x0	ro	NAK 状态 (NAK status) 表示为： 0：表示控制器根据 FIFO 状态发送非 NAK 握手信号 1：表示控制器正在发送 NAK 握手信号。 当此位置起时（无论是应用程序设置的还是控制器设置的），控制器都会停止接收数据，即使接收 FIFO 有可用空间。不管该位是否置起，控制器始终以 ACK 握手信号来响应 SETUP 数据包。
位 16	保留	0x0	resd	保持默认值。
位 15	USBACEPT	0x1	ro	USB 有效端点 (USB active endpoint) 此位始终置 1，表示不管在什么配置和接口中，控制端点 0 始终有效。
位 14：2	保留	0x0000	resd	保持默认值。
位 1：0	MPS	0x0	ro	最大包长度 (Maximum packet size) 控制 OUT 端点 0 的最大包长度与控制器 IN 端点 0 的设置是一样的，如下： 00：64 字节； 01：32 字节； 10：16 字节； 11：8 字节。

21.6.5.12 OTGFS设备OUT端点x控制寄存器(OTGFS_DOEPCTLx)(其中x为端点号，x=1…7)

应用程序通过此寄存器来控制除了端点 0 的其他端点的操作特性。

域	简称	复位值	类型	功能
位 31	EPTENA	0x0	rw1s	端点使能 (Endpoint enable) 此位表示已设置好描述符结构和准备接收数据的数据缓冲区。控制器在产生以下端点中断之前会先清除该位： ■SETUP 阶段完成 ■端点禁用 ■传输完成
位 30	EPTDIS	0x0	ro	端点禁用(Endpoint disable) 应用程序可通过设置此位来停止某个端点上的数据发送/传输，即使传输还没完成。

				应用程序需要等到端点禁用中断产生才视为该端点已被禁用。控制器在设置端点禁用中断前会清除此位。应用程序只有在端点已经使用端点时才能设置此位。 数值: 0：无作用 1：端点禁用
位 29	SETD1PID/ SETODDFR	0x0	rw	设置 DATA1 PID (Set DATA1 PID) 仅适用于中断/批量 OUT 端点，写此位可以将该寄存器的端点数据 PID 位设置为 DATA1。 设置奇数帧 (Set odd frame) 仅适用于同步 OUT 端点，写此位可以将奇偶帧位设置为奇数帧。 0：关闭 Set DATA1 PID 或者不强制奇数帧 1：使能 Set DATA1 PID 或者强制奇数帧
位 28	SETD0PID/ SETEVENFR	0x0	rw	设置 DATA0 PID (Set DATA0 PID) 仅适用于中断/批量 OUT 端点，写此位可将该寄存器的端点数据 PID 位设置为 DATA0。 设置偶数帧 (Set Even frame) 仅适用于同步 OUT 端点，写此位可以将偶数帧/奇数帧位设置为偶数帧。 0：关闭 Set DATA0 PID 或不强制偶数帧 1：端点数据 PID 设置为 DATA0 或将 EOFRNUM 设置为偶数帧
位 27	SNAK	0x0	wo	设置 NAK (Set NAK) 通过对该位写操作可以设置端点的 NAK 位。应用程序可通过此位来控制端点上的 NAK 握手信号的发送。控制器在接收到 SETUP 数据包或传输结束中断时置起此位。 数值: 0：未设置 NAK 1：设置 NAK
位 26	CNAK	0x0	wo	清除 NAK (Clear NAK) 通过对该位写操作可以清除端点的 NAK 位。 数值: 0：不清除 NAK 1：清除 NAK
位 25：22	保留	0x0	resd	保持默认值。
位 21	STALL	0x0	rw	适用于非控制，非同步 IN 端点和 OUT 端点。 应用程序通过此位停止向该端点发送 USB 主机的令牌。如果 NAK 位，全局非周期性 IN NAK 位或者全局 OUT NAK 位以及该位同时置起时，STALL 位享有优先级。只有应用程序才能清除此位，控制器不能。
位 20	SNP	0x0	rw	监听模式 (Snoop mode) 设置此位将使端点进入监听模式。在监听模式下，控制器在将 OUT 数据包写入应用程序缓存区前不检查数据的正确性。
位 19：18	EPTYPE	0x0	rw	端点类型 (Endpoint type) 表示逻辑端点支持的传输类型。 00：控制 01：同步 10：块传输 11：中断
位 17	NAKSTS	0x0	ro	NAK 状态 (NAK status) 0：表示控制器根据 FIFO 状态发送非 NAK 握手信号 1：表示控制器正在发送 NAK 握手信号。 当此位置起时（无论是应用程序设置的还是控制器设置的） ■控制器停止接收 OUT 端点上的数据，即使接收 FIFO 有剩余空间可以用来容纳传入的数据包。

				■对于非同步 IN 端点：控制器停止发送端点上的数据，即使发送 FIFO 还有待发送的数据。 ■对于同步 IN 端点：控制器会发出一个零长度的数据包，即使发送 FIFO 还有剩余空间。 不管该位是否置起，控制器始终以 ACK 握手信号来响应 SETUP 数据包。
位 16	DPID/ EOFRNUM	0x0	ro	端点数据 PID (Endpoint data PID) 仅适用于中断/批量 OUT 端点。 此位包含的是该端点上要传输或要发送的数据包的 PID 信息。在端点使能后，应用程序需要设置该端点待发送或接收的首个数据包的 PID。应用程序通过该寄存器的 SetD1PID 和 SetD0PID 来设置 DATA0 或 DATA1 PID。 0：DATA0 1：DATA1 奇/偶数帧 (Even/odd frame) 仅适用于同步 OUT 端点。 表示控制器传输或接收该端点的同步数据的帧号。应用程序通过该寄存器的 SETEVNFR 和 SETODDFR 设置希望传输或接收同步数据的偶数帧号/奇数帧号。 0：偶数帧 1：奇数帧
位 15	USBACEPT	0x0	rw	USB 活跃端点(USB active endpoint) 指示在当前配置和接口下，该端点是否活跃。控制器在检测到 USB 复位之后会清除此位(除了端点 0)。应用程序在收到 SetConfiguration 和 SetInterface 命令时必须设置端点寄存器并置起此位。 0：不活跃 1：活跃
位 14：11	保留	0x0	resd	保持默认值。
位 10：0	MPS	0x000	rw	最大包长度(Maximum packet size) 应用程序通过此位设置当前逻辑端点的最大包长度，以字节为单位。

21.6.5.13 OTGFS设备IN端点x中断寄存器(OTGFS_DIEPINTx)(其中x为端点号，x=0…7)

本寄存器指示在发生 USB 及 AHB 相关事件时某个端点的状态，具体请参考图 21-2。当控制器中断寄存器的 IN 端点中断位(OTGFS_GINTSTS 寄存器的 IEPINT 位)为 1 时，程序必需先读设备所有端点中断寄存器(OTGFS_DAINT)来获得发生事件的端点号，然后再读相应端点的中断寄存器获得详细信息。应用程序必需通过清除此位来清除 OTGFS_DAINT 和 OTGFS_GINTST 寄存器的对应位。

域	简称	复位值	类型	功能
位 31：8	保留	0x000000	resd	保持默认值。
位 7	TXFEMP	0x1	ro	发送 FIFO 空 (Transmit FIFO empty) 当端点的发送 FIFO 为全空或半空时，会生成中断。具体是半空还是全空取决于控制器 AHB 配置寄存器的发送 FIFO 空级别位。
位 6	INEPTNAK	0x0	rw1c	IN 端点 NAK 有效 (IN endpoint NAK effective) 在将对应的 DIEPCTLx.CNAK 设置为 1 之前，需要先写 1 来清除此位。 此中断表示应用程序设置的 IN 端点 NAK 位已经生效。 此中断并不能保证已在 USB 线上发送了 NAK 握手信号。 STALL 位的优先级高于 NAK 位。 此位仅适用于端点已使能的情况。
位 5	保留	0x0	resd	保持默认值。
位 4	INTKNTXFEMP	0x0	rw1c	当发送 FIFO 空时收到 IN 命令 (IN token received when TxFIFO is empty) 此位表示当相关的发送 FIFO(不管是周期性或非周期性)时接收到一个 IN 令牌。在收到 IN 令牌的端点上会生成中断。

位 3	TIMEOUT	0x0	rw1c	超时条件(Timeout condition) 仅适用于控制 IN 端点，此位表示控制器已经侦测到该端点上最后一个 IN 令牌超时。
位 2	保留	0x0	resd	保持默认值。
位 1	EPTDISD	0x0	rw1c	端点禁用中断 (Endpoint disabled interrupt) 表示已经按照应用程序的请求禁用了端点。
位 0	XFERC	0x0	rw1c	传输完成中断 (Transfer completed interrupt) 表示 AHB 以及 USB 已完成该端点设置的传输任务。

21.6.5.14 OTGFS设备OUT端点x中断寄存器(OTGFS_DOEPINTx)(其中x为端点号，x=0…7)

此寄存器指示相应端点与 USB 和 AHB 相关的事件状态。具体请参考图 21-2。当控制器中断寄存器的 OUT 端点中断位(OTGFS_GINTSTS 寄存器的 OEPINT 位)为 1 时，应用程序必需先读设备所有端点中断寄存器(OTGFS_DAINT)来获得发生事件的端点号，然后再读相应端点的中断寄存器获得详细信息。应用程序必需通过清除此位来清除 OTGFS_DAINT 和 OTGFS_GINTST 寄存器的对应位。

域	简称	复位值	类型	功能
位 31：7	保留	0x0000000	resd	保持默认值。
位 6	B2BSTUP	0x0	rw1c	收到连续的 SETUP 包 (Back-to-back SETUP packets received) 此位表示接收到不止 3 个连接的 SETUP 包。
位 5	保留	0x0	resd	保持默认值。
位 4	OUTTEPD	0x0	rw1c	端点禁用时接收到 OUT 令牌(OUT token received when endpoint disabled) 仅对控制 OUT 端点有效。 此们表示在端点还没有使能的情况下接收到了 OUT 令牌。在收到 OUT 令牌的端点上会生成中断。
位 3	SETUP	0x0	rw1c	SETUP 阶段完成 (SETUP phase done) 仅适用于控制 OUT 端点，此位表示该控制端点的 SETUP 阶段已完成，当前控制传输不再接收连续的 SETUP 数据包。一旦生成该中断，应用程序就可以解析所收到的 SETUP 数据包。
位 2	保留	0x0	resd	保持默认值。
位 1	EPTDISD	0x0	rw1c	端点禁止中断 (Endpoint disabled interrupt) 表示已经按照应用程序的请求禁用了端点。
位 0	XFERC	0x0	rw1c	传输完成中断 (Transfer completed interrupt) 表示 AHB 以及 USB 已完成该端点设置的传输任务。

21.6.5.15 OTGFS设备IN端点0传输长度寄存器(OTGFS_DIEPTSIZ0)

应用程序必须在使能端点 0 之前设置该寄存器。一旦通过设备控制端点 0 控制寄存器的端点使能位将端点 0 使能后，就只能通过控制器修改本寄存器。一旦控制器清除了端点使能位，则应用程序只能读此寄存器。

域	简称	复位值	类型	功能
位 31：21	保留	0x000	resd	保持默认值。
位 20：19	PKTCNT	0x0	rw	包数目(Packet count) 表示 USB 数据包总数目，其构成端点 0 的数据传输长度。 每次从发送 FIFO 读取一个数据包时（不管是最大包长度还是短包），此位域就会自动递减。
位 18：7	保留	0x000	resd	保持默认值。
位 6：0	XFERSIZE	0x00	rw	传输长度 (Transfer size) 表示端点 0 上的传输长度（以字节为单位）。当传输长度变为 0 时，控制器就会中断应用程序。在每个包结束后，传输长度可以设置成该端点的最大包长度。 控制器在每次将外部存储器的一个数据包写入发送 FIFO 时，此位域就会自动递减。

21.6.5.16 OTGFS设备OUT端点0传输长度寄存器(OTGFS_DOEPTSIZ0)

应用程序必须在使能端点 0 之前设置该寄存器。一旦通过设备控制端点 0 控制寄存器的端点使能位将端点 0 使能后，就只能通过控制器修改本寄存器。一旦控制器清除了端点使能位，则应用程序只能读此寄存器。

域	简称	复位值	类型	功能
位 31	保留	0x0	resd	保持默认值。
位 30：29	SUPCNT	0x0	rw	SETUP 包数目 (SETUP packet count) 表示该端点能够收到的连续 SETUP 数据包数。 01：1 个数据包； 10：2 个数据包； 11：3 个数据包。
位 28：20	保留	0x000	resd	保持默认值。
位 19	PKTCNT	0	rw	包数目 (Packet count) 将数据包写入接收 FIFO 之后，此位自动递减为 0。
位 18：7	保留	0x000	resd	保持默认值。
位 6：0	XFERSIZE	0x00	rw	传输长度 (Transfer size) 表示端点 0 上的传输长度，以字节为单位。在传输长度变为 0 后，控制器会中断应用程序。传输长度可以设置成该端点的最大包长度，在每个包结束时中断。 控制器在每次将外部存储器的一个数据包写入发送 FIFO 时，此位域就会自动递减。 控制器在每次从接收 FIFO 读取一个数据包并将其写入外部存储器后，此位域就会自动递减。

21.6.5.17 OTGFS设备IN端点x传输长度寄存器(OTGFS_DIEPTSIZx)(其中 x 为端点号，x=1…7)

应用程序必须在使能端点 x 之前设置该寄存器。一旦通过设备控制端点 x 控制寄存器的端点使能位将端点 x 使能后，就只能通过控制器修改本寄存器。一旦控制器清除了端点使能位，则应用程序只能读此寄存器。

域	简称	复位值	类型	功能
位 31	保留	0x0	resd	保持默认值。
位 30：29	MC	0x0	rw	帧内包数目 (Multi count) 对于周期性 IN 端点来说，此位域表示 USB 总线上每个帧必须传输的包数目。控制器通过此位域计算同步 IN 端点发送的数据 PID。 01：1 个数据包 10：2 个数据包 11：3 个数据包
位 28：19	PKTCNT	0x000	rw	包数目 (Packet count) 表示 USB 数据包总数（端点上的数据传输长度）。 每次从发送 FIFO 读取一个数据包时（最大包长度和短包），此位域就会自动递减。
位 18：0	XFERSIZE	0x00000	rw	传输长度（Transfer Size） 此位域包含当前端点的传输字节数。控制器会在此位为 0 时产生中断并通知应用程序。可以配置此寄存器为端点的最大传输长度，并在每个数据包结束产生中断。 每次将外部存储器的一个数据包写入发送 FIFO 时，控制器就会自动递减此位域。

21.6.5.18 OTGFS设备IN端点传输FIFO状态寄存器 (OTGFS_DTXFSTSx)(其中x为端点号，x=0…7)

只读寄存器，储存的是设备 IN 端点发送 FIFO 的可用空间信息。

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持默认值。
位 15：0	INEPTXFSAV	0x0200	ro	IN 端点发送 FIFO 剩余空间 (IN endpoint TxFIFO space avail) 表示端点发送 FIFO 的可用空间。以 32 位的字为单位。 0x0：发送 FIFO 满；

				0x1：剩余 1 个字； 0x02：剩余 2 个字； 0xn：剩余 n 个字(其中 0 < n < 512)； 0x200：剩余 512 个字； 其他：保留。

21.6.5.19 OTGFS设备OUT端点x传输长度寄存器 (OTGFS_DOEPTSIZx)(其中x 为端点号，x=1…7)

应用程序必须在使能端点 x 之前设置该寄存器。一旦通过设备控制端点 x 控制寄存器的端点使能位将端点 x 使能后，就只能通过控制器修改本寄存器。一旦控制器清除了端点使能位，则应用程序只能读此寄存器。

域	简称	复位值	类型	功能
位 31	保留	0x0	resd	保持默认值。
位 30：29	RXDPID	0x0	ro	收到的数据 PID (Received data PID) 此位仅对同步 OUT 端点有效。表示收到的最后一个数据包的数据 PID： 00：DATA0； 01：DATA2； 10：DATA1； 11：MDATA。 SETUP 包数目 (SETUP packet count) 此位仅对控制 OUT 端点有效。表示端点收到的连续的 SETUP 包的数量： 01：1 个包； 10：2 个包； 11：3 个包。
位 28：19	PKTCNT	0x000	rw	包数目 (Packet count) 指示该端点上传输的 USB 包数目。 每次向接收 FIFO 写入一个数据包时（最大包长度和短包），此位域就会自动递减。
位 18：0	XFERSIZE	0x00000	rw	传输长度 (Transfer size) 此位指示该端点要传输的字节数。控制器在此域为 0 时会产生中断通知应用程序。可以配置此域 为端点的最大传输长度，并在每个数据包结束产生中断。 每当从接收 FIFO 读取一个数据包并将其写入外部存储器时，控制器就会自动递减此位域。

21.6.6 供电和时钟控制寄存器

21.6.6.1 OTGFS电源和时钟门控寄存器(OTGFS_PCGCCTL)

此寄存器既适用于主机模式也适用于设备模式。

域	简称	复位值	类型	功能
位 31：5	保留	0x0000000	resd	保持默认值。
位 4	SUSPENDM	0x0	ro	物理层挂起（PHY suspend） 指示 PHY 处于挂起状态
位 3：1	保留	0x0	resd	保持默认值。
位 0	STOPPCLK	0x0	rw	停止 PHY 时钟 (Stop PHY clock) 当 USB 挂起或会话无效或者设备断开时，应用程序通过此位停止 PHY 时钟。当 USB 恢复或者有新的会话请求时，应用程序会清除此位。

22 HICK 自动时钟校准（ACC）

22.1 简介

HICK 自动时钟校准器（HICK ACC）利用 USB 模块产生的 SOF 信号（周期为 1 毫秒）作为参考信号，实现对 HICK 时钟的采样和校准。
本模块主要功能就是实现对 USB 设备提供 48MHz±0.25%精度的时钟。
它采取“跨越回归”算法，可以将校准后的频率尽可能地靠近目标频率。

22.2 主要特性

- 可配置的中心频率
- 可配置的触发校准功能的边界频率
- 满足中心频率±0.25%的精度要求
- 状态检测标志
  - 校准就绪标志
  - 一个错误检测标志
  - 参考信号丢失错误
- 2 个带标志的中断源
  - 校准就绪标志
  - 参考信号丢失错误
- 两种校验方式：粗校验和精校验。

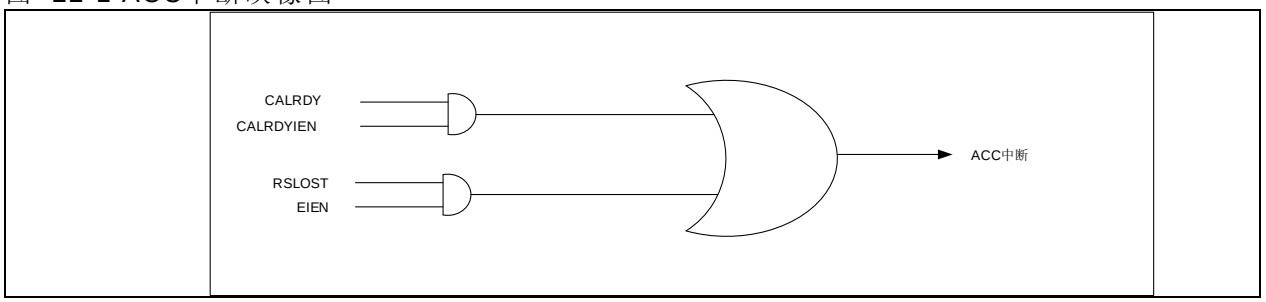
22.3 中断请求

表 22-1 ACC中断请求

中断事件	事件标志	使能位
校准就绪	CALRDY	CALRDYIEN
参考信号丢失错误	RSLOST	EIEN

ACC 的各种中断事件被连接到同一个中断向量（见下图），有以下各种中断事件：
- 校准期间：当校准就绪和参考信号丢失错误。

如果设置了对应的使能控制位，这些事件就可以产生各自的中断。
图 22-1 ACC中断映像图



22.4 功能概述

ACC 模块的功能：利用 USB 模块产生的 SOF 信号（周期为 1 毫秒）作为参考信号，实现对 HICK 时钟的采样和校准。特别的是，可以将 HICK 时钟的频率精度校准到±0.25%以内的精度，从而满足高精度时钟要求的应用场景，例如 USB 应用。
本模块的信号均未外接到芯片管脚，而是和芯片内部的 CRM、HICK 等模块相连。
- CRM_HICKCAL：复位和时钟控制（CRM）模块之 HICKCAL。此信号用于 bypass 模式下对内部高速时钟（HICK 模块）的校准，其值的大小由 CRM_CTRL 中的 HICKCAL[7:0]定义。

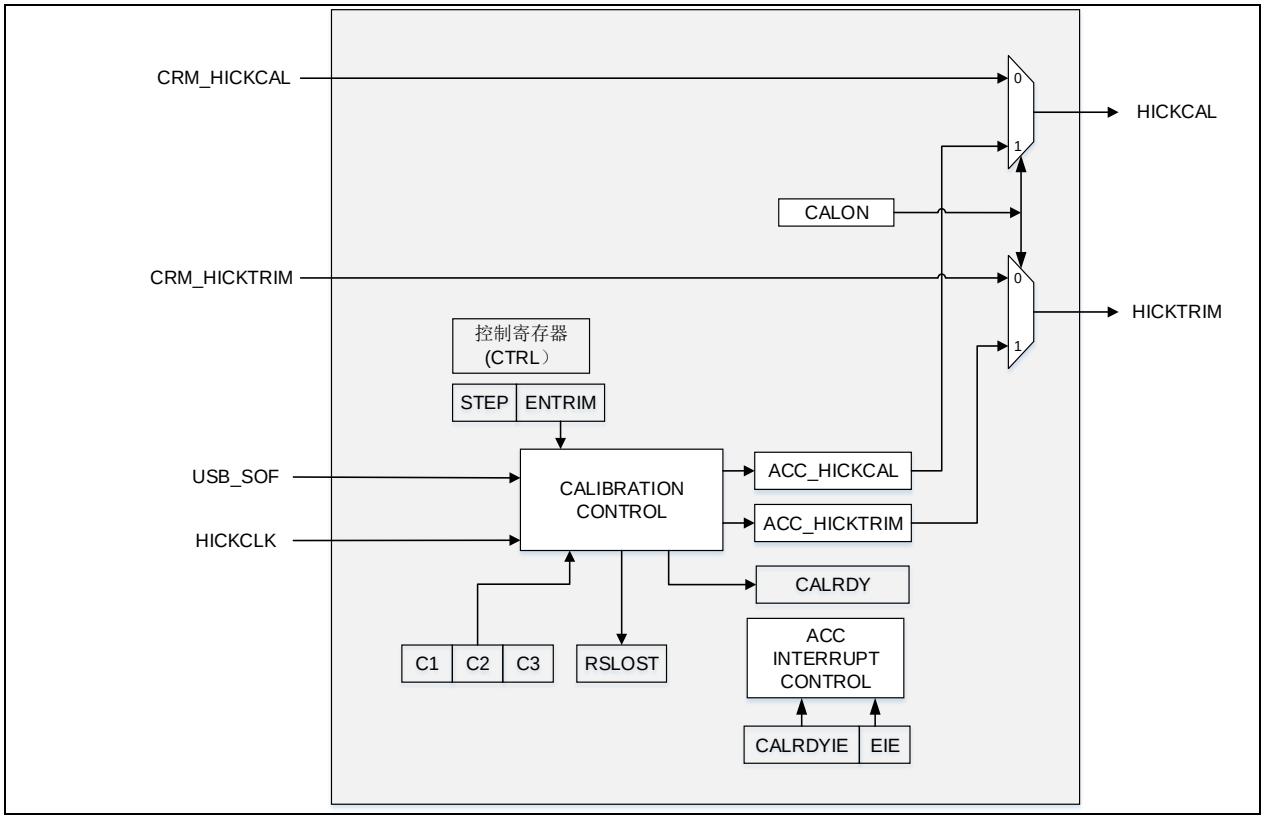
- CRM_HICKTWK：复位和时钟控制（CRM）模块之 HICKTWK。此信号用于 bypass 模式下对内部高速时钟校准（HICK）的校准，其值的大小由 CRM_CTRL 中的 HICKTWK[5:0]定义。

默认数值为 32，可以把 HICK 调整到 8MHz±0.25%；每步 CRM_HICKTWK 的变化调整 HICK 的频率 20kHz（设计值）。

- USB_SOF：USB 设备解析给出的帧开始信号（USB Start-of-Frame）。其高电平宽度为 64 个系统时钟周期，周期为 1 毫秒的脉冲信号。
- HICKCLK：HICK 时钟。本系列的 HICK 模块输出的原始时钟频率为 48MHz，但是 HICK 校准模块使用的采样时钟是除频（1/6）电路输出的时钟，频率约 8MHz。
- HICKCAL：HICK 模块的校验信号。对于除频（1/6）后的 HICK 时钟来讲，HICKCAL 每改变一步，除频（1/6）后 HICK 时钟频率改变 40KHz（设计值），且为正相关。换句话说，HICKCAL 每增加一，除频（1/6）后 HICK 时钟频率会增加 40KHz（设计值）；HICKCAL 每减少一，除频（1/6）后 HICK 时钟频率会减少 40KHz。
- HICKTWK：HICK 模块的校验信号。对于除频（1/6）后的 HICK 时钟来讲，HICKTWK 每改变一步，除频（1/6）后 HICK 时钟频率改变 20KHz（设计值），且为正相关。

关于以上寄存器中每个位的具体定义，请参考寄存器描述第 22.6 节：寄存器描述。

图 22-2 ACC框图

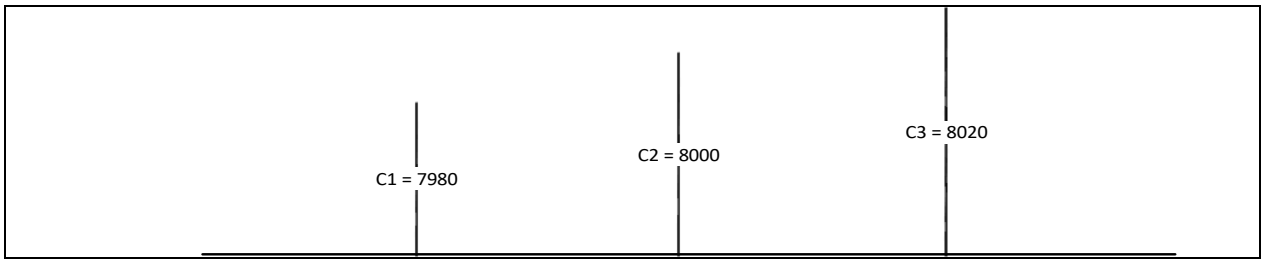


22.5 原理分析

USB_SOF 周期信号：1 毫秒的周期性必须是准确的，是自动校准模块能够正常工作的前提条件。

cross-return 策略：以计算出离理论值最近的校准值。从理论上来说，可以将校准后的实际频率调校到离目标频率（8MHz）约 0.5 个 step 的精度范围以内。

图 22-3 cross-return策略



如上图所示，一旦触发自动校准的条件满足，自动校准就会按照 step 所规定的步长调整 HICKCAL 或者 HICKTWK。
跨越（cross）：
在满足自动校验的条件后的第一个 1 毫秒采样周期内的实际采样值要么小于 C2，要么大于 C2。
当这个值小于 C2，自动校准按照 step 的定义，增加 HICKCAL 或者 HICKTWK，直到实际采样值比 C2 大，实现实际采样值由小到大对 C2 的跨越。
当这个值大于 C2，自动校准按照 step 的定义，减少 HICKCAL 或者 HICKTWK，直到实际采样值比 C1 小，实现实际采样值由大到小对 C2 的跨越。
回归（return）：
在跨越完成后，比较在跨越前后的实际采样值和 C2 之间的差值（按绝对值计算），得到离 C2 最近的实际采样值，从而得到最佳的校验值 HICKCAL 或者 HICKTWK。
若跨越后的实际采样值和 C2 之间的差值小于跨越前的实际采样值和 C2 之间的差值，则以跨越后的校验值为准，并结束校验流程，直到满足下一个满足自动校验的条件。
若跨越后的实际采样值和 C2 之间的差值大于跨越前的实际采样值和 C2 之间的差值，则以跨越前的校验值为准，那么校验值会退回一个 step，并返回到跨越前的那个校验值，并结束校验流程，直到满足下一个满足自动校验的条件。
按照 cross-return 策略，在理论上，可以得到离中心频率约 0.5 个 step 所对应的频率精度。

如下四种情形会启动自动校准：
第一， CALON 的上升沿（从 0 到 1）；
第二， 当 CALON=1 时，参考信号丢失之后又恢复；
第三， 当采样计数器的值小于 C1；
第四， 当采样计数器的值大于 C3。
在 CALON 的上升沿，即便采样计数器的值大于 C1 并小于 C3,也会启动自动校准，其目的在于，在 CALON 之后，能够尽快将 HICK 的频率调整到中心频率的 0.5 个 step 以内。
以上四种情形的自动校准的结果均能将 HICK 的频率调整到中心频率的 0.5 个 step 以内。所以为了获得最佳的校准精度，建议将 step 保持为默认值 1。若将 step 设为 0，则 HICKCAL 或者 HICKTWK 将无法改变，也即，无法校准。

22.6寄存器描述

有关寄存器描述里所使用的缩写，请参考“寄存器描述表中使用的缩写列表”。
必须用字（32 位）的方式操作这些外设寄存器。

22.6.1 ACC寄存器地址映象

表 22-2 ACC寄存器映像和复位值

寄存器简称	基址偏移量	复位值
ACC_STS	0x00	0x0000 0000
ACC_CTRL1	0x04	0x0000 0100
ACC_CTRL2	0x08	0x0000 2080
ACC_C1	0x0C	0x0000 1F2C
ACC_C2	0x10	0x0000 1F40

ACC_C3	0x14	0x00000 1F54

22.6.2 状态寄存器（ACC_STS）

域	简称	复位值	类型	功能
位 31：9	保留	0x000000	resd	保持默认值。
位 1	RSLOST	0x0	ro	参考信号丢失（Reference Signal Lost） 0：参考信号未丢失； 1：参考信号丢失。 注：在校验过程中，当校准模块的采样计数器的值为 C2 的 2 倍时，还未侦测到 SOF 参考信号，则意味着参考信号丢失。内部状态机回归到 idle 状态，除非再次侦测到 SOF 信号，否则内部时钟采样计数器保持为 0。在 CALON 位清零后，或者向 RSLOST 写入 0，则 RSLOST 立即被清零。仅仅在 CALON=1 时，才会检测参考信号。
位 0	CALRDY	0x0	ro	内部高速时钟就绪（Internal high-speed clock calibration ready） 0：内部 8MHz 振荡器校验没有就绪； 1：内部 8MHz 振荡器校验就绪。 注： 由硬件置’1’来指示内部 8MHz 振荡器已经校验到离 8MHz 最近的频率上。在 CALON 位清零后，或者向 CALRDY 写入 0，则 CALRDY 立即被清零。

22.6.3 控制寄存器1（ACC_CTRL1）

域	简称	复位值	类型	功能
位 31：12	保留位	0x00000	resd	硬件强制为 0
位 11：8	STEP	0x1	rw	校准的步长 这 4 位定义了每次校准改变的值。 备注：为了获得更高的校准精度，建议将 step 设为 1。 当 ENTRIM=0，仅校准 HICKCAL，若 step 改变 1，对应的 HICKCAL 也改变 1，HICK 频率改变 40KHz（设计值），为正相关关系。 当 ENTRIM=1，仅校准 HICKTWK，若 step 改变 1，对应的 HICKTWK 也改变 1，HICK 频率改变 20KHz（设计值），为正相关关系。
位 7：6	保留位	0x0	rw	硬件强制为 0
位 5	CALRDYIEN	0x0	rw	CALRDY 中断使能（CALRDY interrupt enable） 该位由软件设置或清除。 0：禁止产生中断； 1：当 ACC_STS 中的 CALRDY 为’1’时，产生 ACC 中断。
位 4	EIEN	0x0	rw	RSLOST 中断使能（RSLOST error interrupt enable） 该位由软件设置或清除。 0：禁止产生中断； 1：当 ACC_STS 中的 RSLOST 为’1’时，产生 ACC 中断。
位 3：2	保留位	0x0	rw	硬件强制为 0
位 1	ENTRIM	0x0	rw	TWK 使能（Enable trim） 该位由软件设置或清除。 0：仅校准 HICKCAL； 1：仅校准 HICKTWK。 注：为了获得更高的校准精度，建议将 ENTRIM 设为 1。
位 0	CALON	0x0	rw	Calibration 使能（Calibration on） 该位由软件设置或清除。 0：禁止校验； 1：使能校验，并开始搜寻 USB_SOF 上的脉冲。 注：如果没有 USB_SOF 参考信号，则本模块无法使用。若对 HICK 时钟的精度没有要求，也无需开启本模块以节省功耗。

22.6.4 控制寄存器2（ACC_CTRL2）

域	简称	复位值	类型	功能
位 31：14	保留位	0x00000	resd	硬件强制为 0
位 13：8	HICKTWK	0x20	ro	内部高速时钟自动调整（Internal high-speed auto clock trimming） 该位由软件读取，不可写。 由 ACC 自动校准模块来调整内部高速时钟，它们被叠加在 ACC_HICKCAL[7：0]数值上。这些位在 ACC_HICKCAL[7：0]的基础上，让用户可以输入一个调整数值，根据电压和温度的变化调整内部 HICK RC 振荡器的频率。 默认数值为 32，可以把 HICK 调整到 8MHz±0.25%；每步 ACC_HICKTWK 的变化调整 20kHz（设计值）。
位 7：0	HICKCAL	0x80	ro	内部高速时钟自动校准（Internal high-speed auto clock calibration） 该位由软件读取，不可写。 由 ACC 自动校准模块来调整内部高速时钟，让用户可以输入一个调整数值，根据电压和温度的变化调整内部 HICK RC 振荡器的频率。 默认数值为 128，可以把 HICK 调整到 8MHz±0.25%；每步 ACC_HICKCAL 的变化调整 40kHz（设计值）。

22.6.5 比较值1（ACC_C1）

域	简称	复位值	类型	功能
位 31：16	保留位	0x0000	resd	硬件强制为 0。
位 15：0	C1	0x1F2C	rw	比较值 1（Compare 1） 该值是触发校准的下边界，默认值为 7980。当自动校验模块在 1 毫秒的周期内采样的时钟个数小于或等于 C1，则会触发自动校验； 当实际采样值（1 毫秒内的时钟个数）大于 C1，且小于 C3，则不会触发自动校验。

22.6.6 比较值2（ACC_C2）

域	简称	复位值	类型	功能
位 31：16	保留位	0x0000	resd	硬件强制为 0。
位 15：0	C2	0x1F40	rw	比较值 2（Compare 2） 该值确定了理想频率（8MHz）时钟在 1 毫秒为采样周期的时钟个数，默认值为 8000，也是其理论值。 该值是 cross-return 策略的中心点，以计算出离理论值最近的校准值。从理论上来说，可以将校准后的实际频率调教到离目标频率（8MHz）约 0.5 个 step 的精度范围以内。

22.6.7 比较值3（ACC_C3）

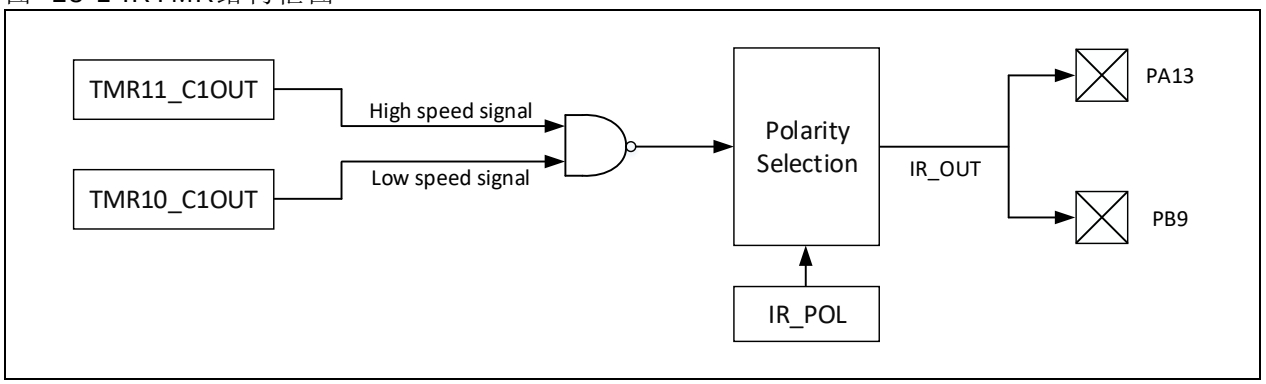
域	简称	复位值	类型	功能
位 31：16	保留位	0x0000	resd	硬件强制为 0。
位 15：0	C3	0x1F54	rw	比较值 3（Compare 3） 该值是触发校准的上边界。当自动校验模块在 1 毫秒的周期内采样的时钟个数大于或等于 C3，则会触发自动校验； 当实际采样值（1 毫秒内的时钟个数）大于 C1，且小于 C3，则不会触发自动校验。

23 红外线接口（IRTMR）

IRTMR 用于产生驱动红外 LED 的 IR_OUT 信号，进而实现红外控制功能。

IR_OUT 信号由低频调制包络信号和高频载波信号两部分组成，低频调制包络信号由 TMR10_C1OUT 提供；高频载波信号由 TMR11_C1OUT 提供，SCFG_CFG1 寄存器 IR_POL 可控制输出的 IR_OUT 信号是否反向。IR_OUT 通过 PB9 或 PA13 通过复用功能输出（需提前配置为复用模式）。

图 23-1 IRTMR结构框图



24 外部存储控制器（XMC）

24.1 XMC简介

XMC 是一个将 AHB 传输信号转换与外部存储器信号相互转换的外设。拥有两个在不同脚位的片选信号，最高可以一次接两个外部存储器。支持的外部存储器有静态随机存储器（SRAM）、NOR 闪存、PSRAM。

24.2 XMC主要特征

NOR/PSRAM 界面有以下特征：

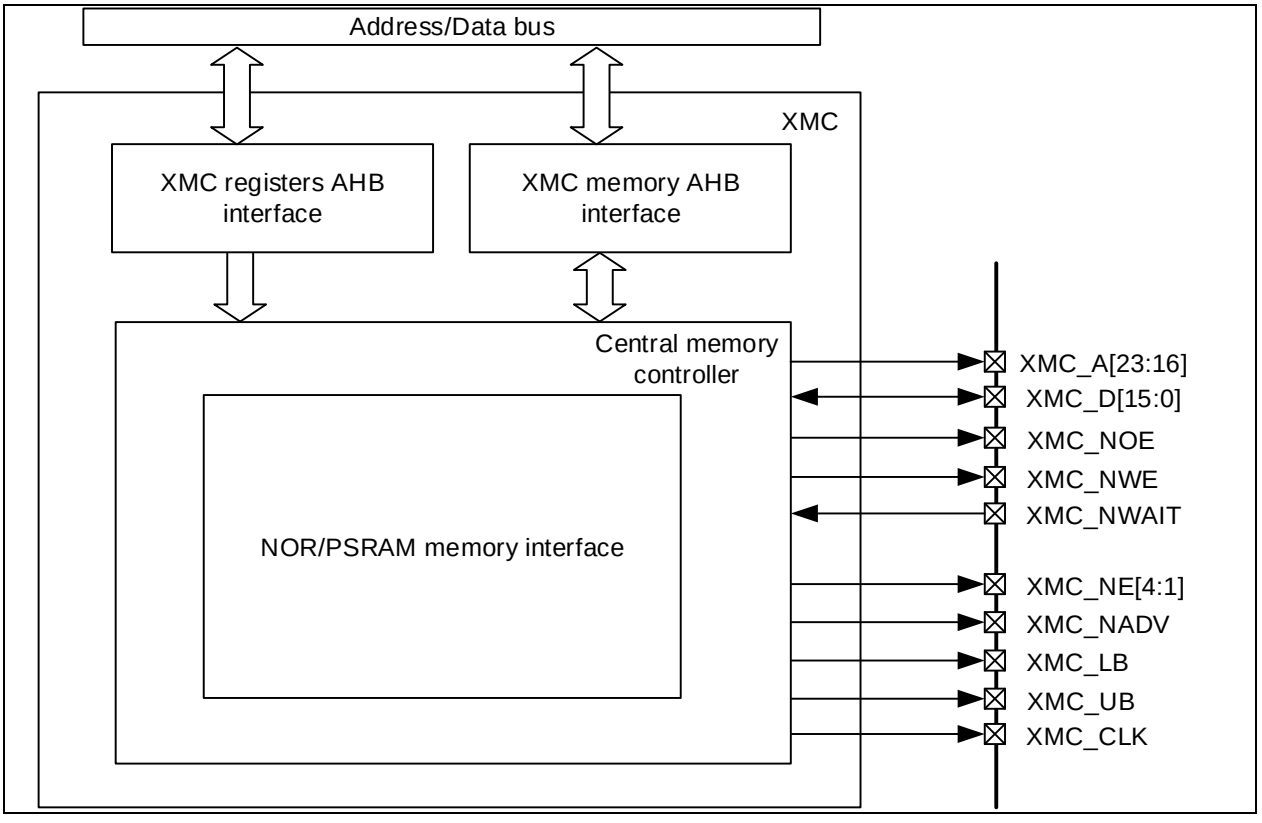
- 支持 3 个外部存储器的片选信号，拥有各自的控制寄存器
- 支持静态存储器件，包括：
  - 静态随机存储器（SRAM）
  - NOR闪存
  - PSRAM
- 支持 8 位与 16 位数据宽度存储器
- 提供多种时序模式选择
  - 读写相同时序的2种模式
  - 读写不同时序的4种模式
  - 地址数据复用的模式
  - 同步模式
- 具可编程的时序控制寄存器
- 支持将 AHB 数据宽度转换为外部存储器适用的数据宽度

24.3 XMC构造

24.3.1 框图

XMC 的架构如下所示

图 24-1XMC框图



与外部存储器沟通时，透过 NOR/PSRAM 界面所需要使用到的引脚如错误!未找到引用源。

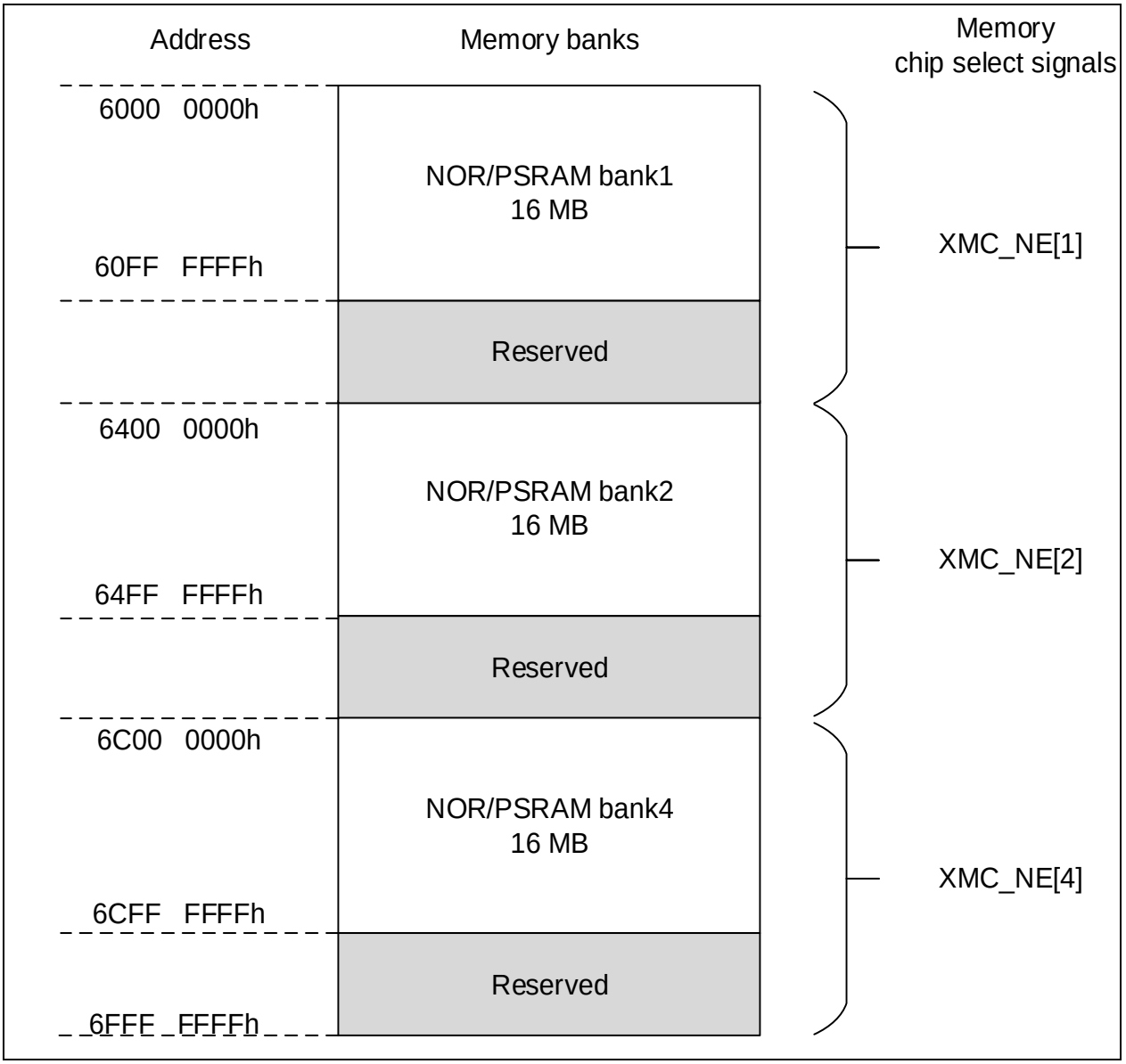
表 24-1NOR/PSRAM界面引脚

引脚	方向	介绍
XMC_CLK	输出	时钟
XMC_NE[x], x=1,4	输出	片选
XMC_NADV	输出	地址锁存或地址有效（NL）信号
XMC_A[x]	输出	地址总线
XMC_NOE	输出	输出使能信号
XMC_NWE	输出	写使能信号
XMC_LB、XMC_UB	输出	字节选择信号
XMC_D[15：0]	读输入/写输出	数据总线/地址数据复用总线
XMC_NWAIT	输入	等待信号

24.3.2 地址映射

XMC 地址分为多个存储块区，如下所示。

图 24-2 XMC存储块区



透过 HADDR 的部份特定位数，选择对哪个存储区块读写，如下表所示。

表 24-2 存储区块选择

HADDR[31：28]	HADDR[27：26]
0110：NOR/PSRAM	00：bank1
	01：bank2
	11：bank4

24.4 NOR/PSRAM界面

NOR/PSRAM 界面提供多种具有不同时序的访问模式，利用这些模式，可驱动多种存储器：NOR 闪存、SRAM、PSRAM 或 Cellular RAM。

3 个存储区块 bank1 到 bank4 有分开的控制寄存器，可使用不同的时序及不同的片选信号访问这 3 个存储区块。

24.4.1 操作方式

引脚使用

不同的外部存储器所需的信号不同，下表列出了典型信号。

表 24-3 NOR闪存与PSRAM典型引脚信号

XMC 引脚信号	NOR 闪存	PSRAM
XMC_CLK	时钟（同步模式）	时钟（同步模式）
XMC_NE[x]	片选信号	片选信号
XMC_NADV	地址锁存或地址有效信号	地址锁存或地址有效信号
XMC_A[23：16]	地址总线	地址总线
XMC_NOE	输出使能信号	输出使能信号
XMC_NWE	写使能信号	写使能信号
XMC_LB、XMC_UB	无使用 XMC_NBL[1：0]信号	XMC_LB：低字节选信号 XMC_UB：高字节选信号
XMC_D[15：0]	数据总线 地址数据复用总线（复用与同步模式）	数据总线 地址数据复用总线（复用与同步模式）
XMC_NWAIT	NOR 闪存要求等待信号	PSRAM 要求等待信号

注意：若存储器数据宽度为 8 位，典型数据总线为 XMC_D[7：0]。

访问地址

HADDR 的高地址用来选择存储区块，低地址选择数据存储地址。HADDR 是字节地址，XMC 可支持字节与半字地址的存储器，地址转换如下表所示。只要对特定地址作读写，XMC 即可根据 HADDR 启动片选信号并对外部存储器的地址做读写。

表 24-4 HADDR与外部存储器地址转换

外部存储器数据宽度	地址线连接	最大可访问存储器空间（位）
8 位	HADDR[23：0]与 XMC_A[23：0]相连。 复用与同步模式时 HADDR[15：0]与 XMC_D[15：0]在地址锁存时间相连。	16M 字节 x8=128 M 位
16 位	HADDR[23：1]与 XMC_A[22：0]相连。 复用与同步模式时 HADDR[16：1]与 XMC_D[15：0]在地址锁存时间相连。	8M 字节 x16=128 M 位

访问数据

在 AHB 数据宽度与存储器数据宽度不同时，XMC 针对外部存储器拥有的典型信号可做适度的处理，下表列出 XMC 支持的操作。

表 24-5 访问数据宽度与外部存储器数据宽度对照表

存储器	模式	AHB 数据宽度	存储器数据宽度	说明
SRAM	异步读写	8/16/32	8	1 次、分 2 次或 4 次 XMC 访问
	异步读写	8/16/32	16	使用字节信号 XMC_LB、XMC_UB、1 次或分 2 次 XMC 访问
NOR 闪存	异步读	8	16	
	异步读写	16	16	

	异步读写	32	16	分 2 次 XMC 访问
	同步读	16	16	
	同步读	32	16	分 2 次 XMC 访问
PSRAM	异步读	8	16	
	异步写	8	16	使用字节信号 XMC_LB、XMC_UB
	异步读写	16	16	
	异步读写	32	16	分 2 次 XMC 访问
	同步写	8	16	使用字节信号 XMC_LB、XMC_UB
	同步读写	16	16	
	同步读写	32	16	分 2 次 XMC 访问

24.4.2 访问模式

XMC 提供多种行为不同的访问模式，每种访问会依据时序参数动作，如表 24-6 所示，用户需依照外部存储器的规格与应用需求进行编程。

XMC 提供的访问模式有：

- 地址数据线复用的复用模式
- 有时钟的同步模式

表 24-6 NOR/PSRAM参数寄存器

参数寄存器	意义	访问模式	单位
ADDRST	地址建立时间	1、2、A、B、C、D、复用	HCLK 周期
ADDRHT	地址保持时间	D、复用	HCLK 周期
DTST	数据建立时间	1、2、A、B、C、D、复用	HCLK 周期
DTLAT	数据延迟时间	同步	XMC_CLK 周期
CLKPSC	时钟分频系数	同步	HCLK 周期

时序控制除了时序参数寄存器外，若是开启等待使能位（NWASEN 或 NWSEN），XMC 会在数据建立其间检查 XMC_NWAIT 信号，若是 XMC_NWAIT 信号处在请求等待状态，XMC 便会等待 XMC_NWAIT 回到就绪状态再进行数据传输。

24.4.2.1 复用模式

如表 24-7 与表 24-8 配置，XMC 即会使用复用模式访问外部存储器。读时序如图 24-3 所示，写时序如图 24-4 所示。

表 24-7 复用模式的SRAM/NOR闪存片选控制寄存器配置

域	名称	配置方式
位 31：20	保留	0x0
位 19	MWMC：对存储器写操作位	0x0
位 18：16	CRPGS：CRAM 页大小选择	0x0
位 15	NWASEN：异步传输等待信号使能	根据存储器规格配置
位 14	RWTD：读写时序不同控制	0x0
位 13	NWSEN：同步传输等待信号使能	0x0
位 12	WEN：写使能	根据需求配置
位 11	NWTCFG：等待时序配置	0x0
位 10	WRAPEN：支持非对齐的成组模式	0x0
位 9	NWPOL：等待信号极性	根据存储器规格配置
位 8	SYNCBEN：同步突发模式使能	0x0
位 7	保留	0x1
位 6	NOREN：NOR 闪存访问使能	根据存储器规格配置
位 5：4	EXTMDBW：外部存储器数据宽度	根据存储器规格配置
位 3：2	DEV：存储器类型	根据存储器规格配置，除 0x0（SRAM）外有效
位 1	ADMUXEN：地址/数据复用使能	0x1
位 0	EN：存储器块使能	0x1

表 24-8 复用模式的SRAM/NOR闪存片选时序寄存器配置

域	名称	配置方式
位 31：30	保留	0x0
位 29：28	ASYNCM：异步访问模式选择	0x0

位 27：24	DTLAT：数据延迟时间	0x0
位 23：20	CLKPSC：时钟分频系数	0x0
位 19：16	BUSLAT：总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间，根据需求与存储器规格配置
位 15：8	DTST：数据建立时间	参照图 24-3 与图 24-4，根据需求与存储器规格配置
位 7：4	ADDRHT：地址保持时间	参照图 24-3 与图 24-4，根据需求与存储器规格配置
位 3：0	ADDRST：地址建立时间	参照图 24-3 与图 24-4，根据需求与存储器规格配置

图 24-3 NOR/PSRAM界面复用模式读

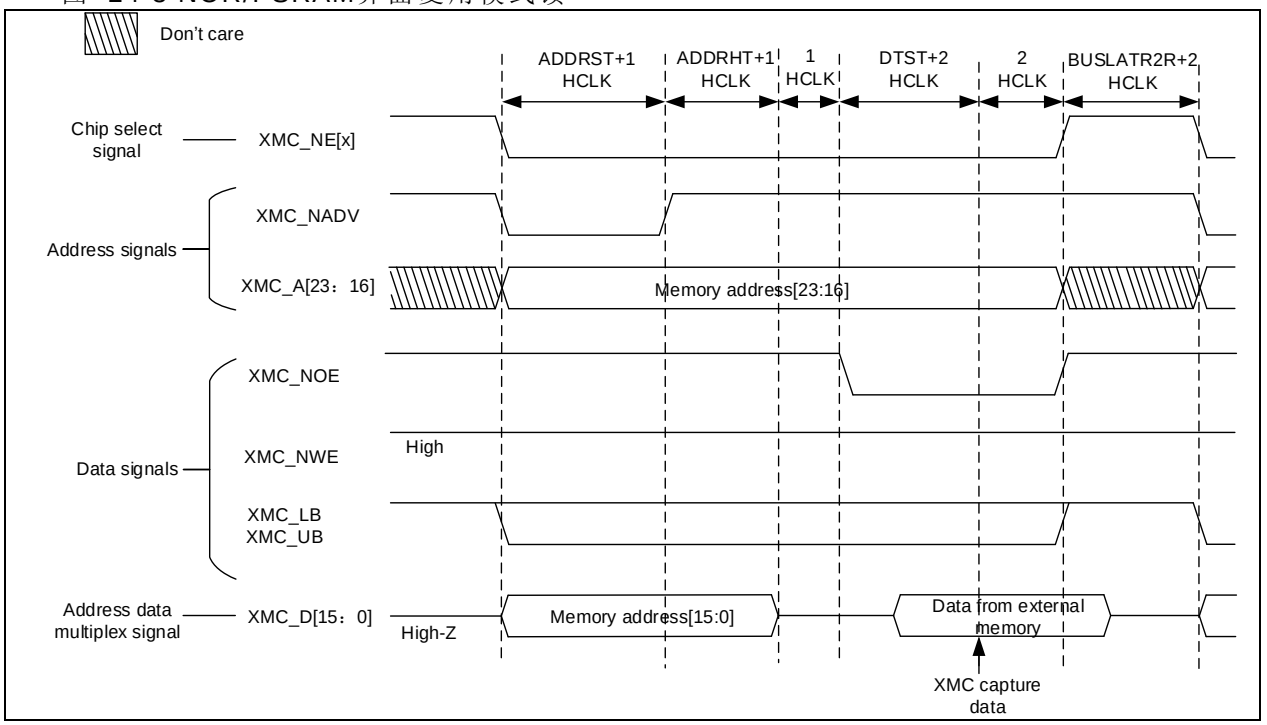
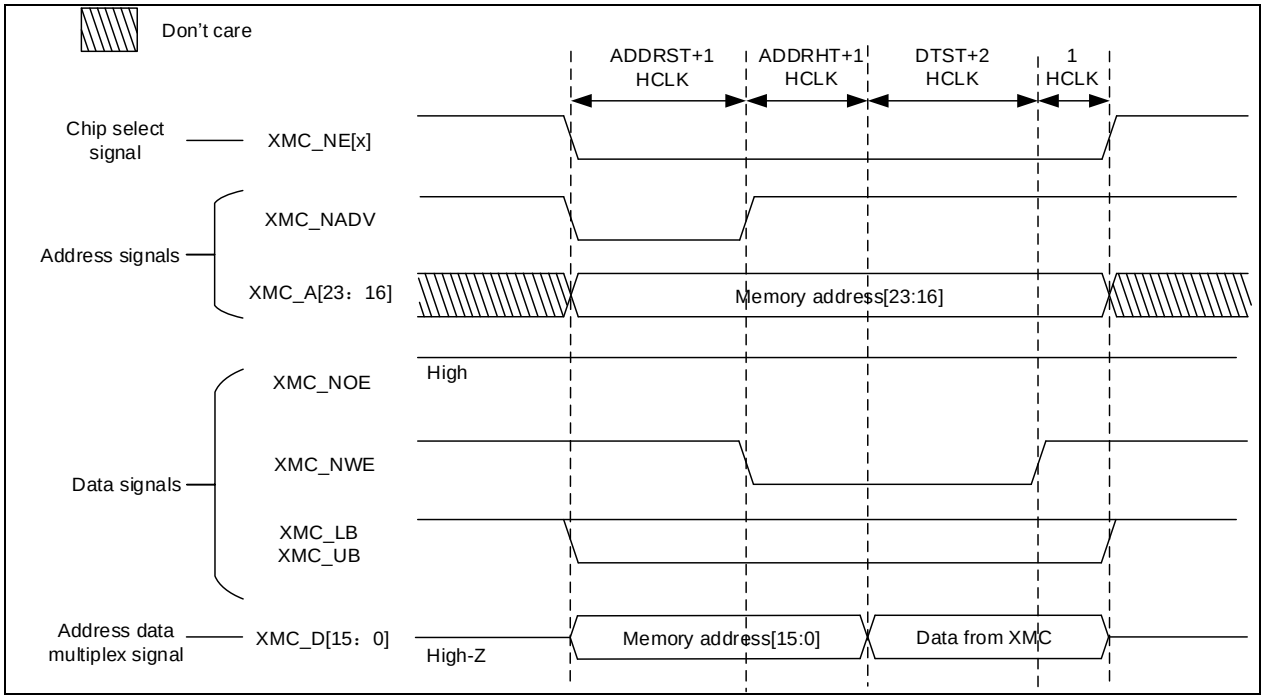


图 24-4 NOR/PSRAM界面复用模式写



24.4.2.2 同步模式

如表 24-9 与表 24-10 配置，XMC 即会使用同步模式访问外部存储器。

若存储器在地址锁存与数据传输之间插入 XMC_NWAIT 信号，XMC 除了等待 DTLAT+1 个 XMC_CLK 外，也会根据 XMC_NWAIT 进行等待。在数据传输途中，XMC 会根据 NWTCFG 的配置在 XMC_NWAIT

信号的下一个周期等待或是当个周期等待。
读时序如图 24-5 所示，写时序如图 24-6 所示。图 24-5 与图 24-6 皆是 XMC_NWAIT 信号的下一个周期等待（NWTCFG=0）做示范。

表 24-9 同步模式的SRAM/NOR闪存片选控制寄存器配置

域	名称	配置方式
位 31：20	保留	0x0
位 19	MWMC：对存储器写操作位	0x1
位 18：16	CRPGS：CRAM 页大小选择	根据存储器规格配置
位 15	NWASEN：异步传输等待信号使能	0x0
位 14	RWTD：读写时序不同控制	0x0
位 13	NWSEN：同步传输等待信号使能	根据存储器规格配置
位 12	WEN：写使能	根据需求配置
位 11	NWTCFG：等待时序配置	根据存储器规格配置
位 10	WRAPEN：支持非对齐的成组模式	根据需求配置
位 9	NWPOL：等待信号极性	根据存储器规格配置
位 8	SYNCBEN：同步突发模式使能	0x1
位 7	保留	0x1
位 6	NOREN：NOR 闪存访问使能	同步写：0x0 同步读：根据存储器规格配置
位 5：4	EXTMDBW：外部存储器数据宽度	根据存储器规格配置
位 3：2	DEV：存储器类型	同步写：0x1 同步读：根据存储器规格配置，除 0x0（SRAM）外有效
位 1	ADMUXEN：地址/数据复用使能	根据需求配置
位 0	EN：存储器块使能	0x1

表 24-10 同步模式的SRAM/NOR闪存片选时序寄存器配置

域	名称	配置方式
位 31：30	保留	0x0
位 29：28	ASYNCM：异步访问模式选择	0x0
位 27：24	DTLAT：数据延迟时间	参照图 24-5 与图 24-6，根据需求与存储器规格配置
位 23：20	CLKPSC：时钟分频系数	XMC_CLK 周期为 HCLK 周期*（CLKPSC+1）。参照图 24-5 与图 24-6，根据需求与存储器规格配置
位 19：16	BUSLAT：总线延迟时间	0x0
位 15：8	DTST：数据建立时间	0x0
位 7：4	ADDRHT：地址保持时间	0x0
位 3：0	ADDRST：地址建立时间	0x0

图 24-5 NOR/PSRAM界面同步模式复用读

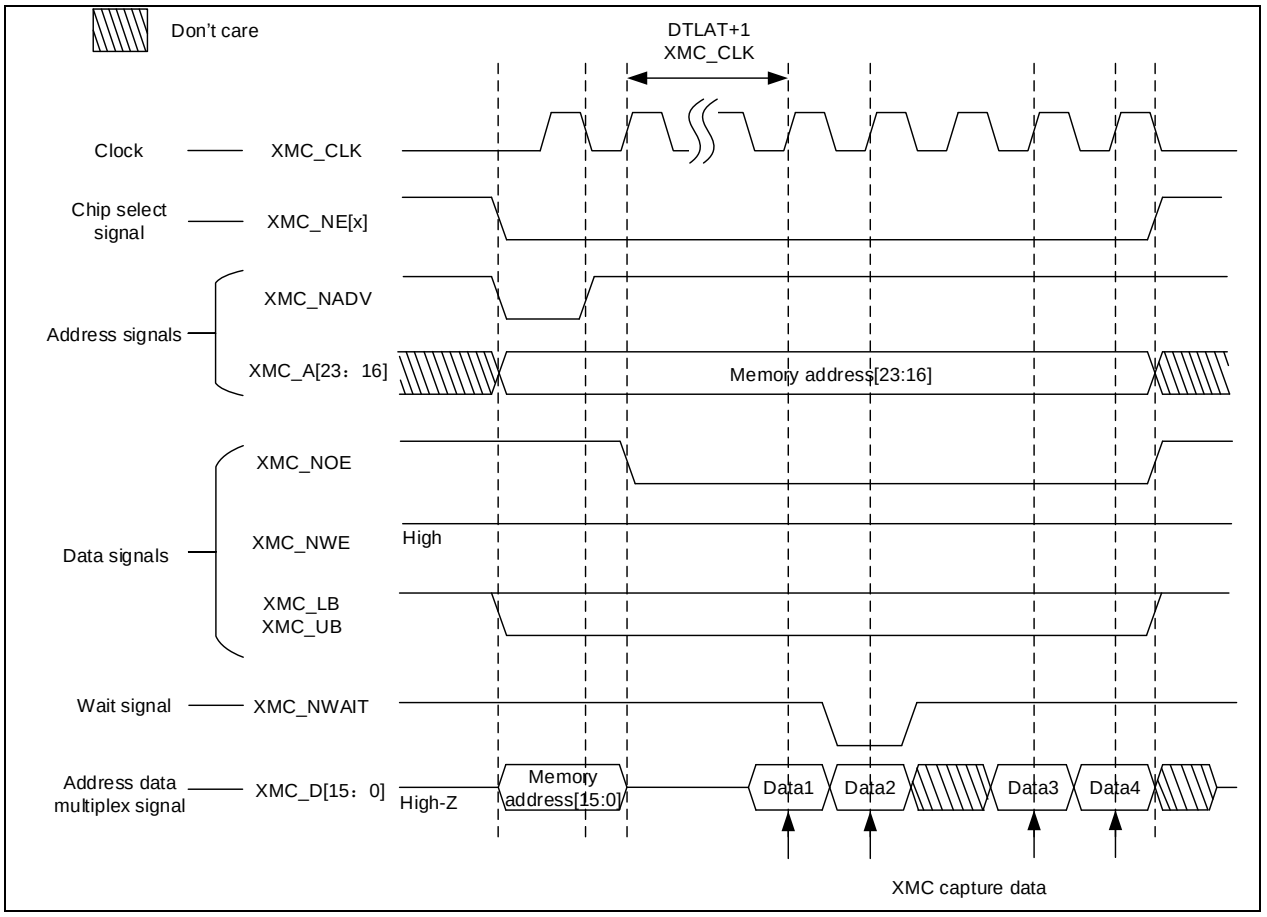
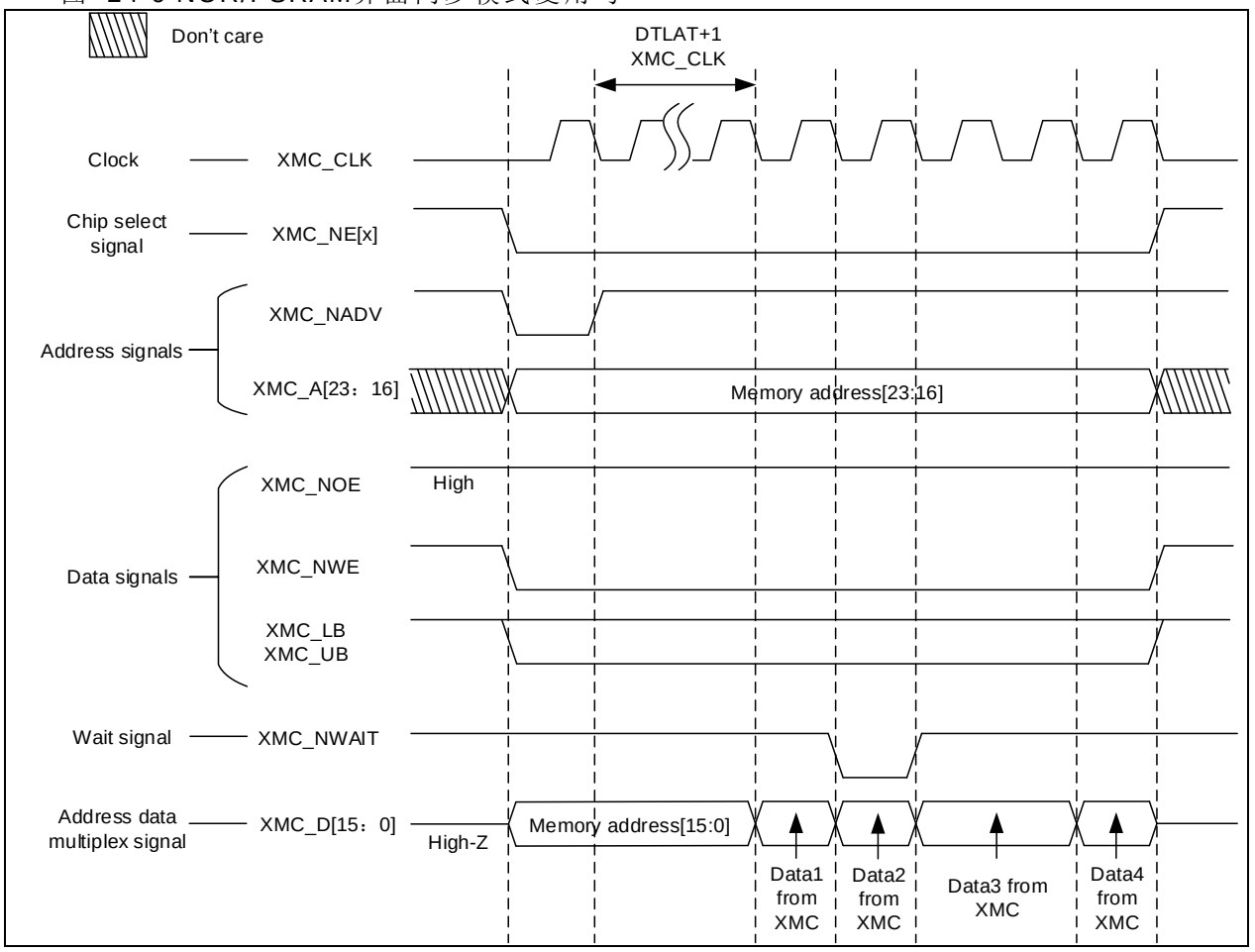


图 24-6 NOR/PSRAM界面同步模式复用写



24.5 XMC寄存器

必须以字（32 位）的方式操作这些外设寄存器。

表 24-11 XMC寄存器地址映像

寄存器简称	基址偏移量	复位值
XMC_BK1CTRL1	0x000	0x0000 30DB
XMC_BK1TMG1	0x004	0x0FFF FFFF
XMC_BK1CTRL2	0x008	0x0000 30D2
XMC_BK1TMG2	0x00C	0x0FFF FFFF
XMC_BK1CTRL4	0x018	0x0000 30D2
XMC_BK1TMG4	0x01C	0x0FFF FFFF
XMC_BK1TMGWR1	0x104	0x0FFF FFFF
XMC_BK1TMGWR2	0x10C	0x0FFF FFFF
XMC_BK1TMGWR4	0x11C	0x0FFF FFFF
XMC_EXT1	0x220	0x0000 0808
XMC_EXT2	0x224	0x0000 0808
XMC_EXT4	0x22C	0x0000 0808

24.5.1 NOR闪存和PSRAM控制器寄存器

必须以字（32 位）的方式操作这些外设寄存器。

24.5.1.1 SRAM/NOR闪存片选控制寄存器1（XMC_BK1CTRL1）

访问：字访问

域	简称	复位值	类型	功能
位 31：20	保留	0x000	resd	保持默认值。
位 19	MWMC	0x0	rw	对存储器写操作位（Memory write mode control） 0：写操作为异步模式； 1：写操作为同步模式。
位 18：16	CRPGS	0x0	rw	CRAM 页大小选择位（CRAM page size） Cellular RAM 1.5 不允许跨页地址边界的同步访问。同步模式配置这些位时，遇到跨页时，XMC 会自动拆开访问。 000：当跨页地址边界时不会拆开访问（默认值）； 001：128 字节； 010：256 字节； 011：512 字节； 100：1024 字节； 其他：保留。
位 15	NWASEN	0x0	rw	异步传输期间等待信号使能位（NWAIT in asynchronous transfer enable） 0：禁止 NWAIT 信号； 1：使能 NWAIT 信号。
位 14	RWTD	0x0	rw	读写时序不同控制位（Read-write timing different） 读存储器与写存储器使用不同的时序进行操作，即寄存器 XMC_BK1TMGWR 被开放。 0：读写时序相同； 1：读写时序不同。
位 13	NWSEN	0x1	rw	同步传输期间等待信号使能位（NWAIT in synchronous transfer enable） 0：禁用 NWAIT 信号； 1：使能 NWAIT 信号。

位 12	WEN	0x1	rw	写使能位（Write enable） 0：禁止； 1：使能。
位 11	NWTCFG	0x0	rw	等待时序配置（NWAIT timing configuration） 仅在同步模式有效。 0：NWAIT 信号在等待状态前的一个数据周期有效； 1：NWAIT 信号在等待状态期间有效。
位 10	WRAPEN	0x0	rw	支持非对齐的成组模式（Wrapped enable） XMC 于同步模式时是否支持将非对齐的 AHB 成组操作拆成 2 次操作； 0：不允许直接的非对齐成组操作； 1：允许直接的非对齐成组操作。
位 9	NWPOL	0x0	rw	等待信号极性（NWAIT polarity） 在同步模式下，此位设置 NWAIT 信号极性。 0：低有效； 1：高有效。
位 8	SYNCBEN	0x0	rw	同步突发模式使能（Synchronous burst enable） 允许对闪存存储器进行同步模式访问。 0：禁用； 1：使能。
位 7	保留	0x1	resd	保持默认值。
位 6	NOREN	0x1	rw	NOR 闪存访问使能（Nor flash access enable） 0：禁止 NOR 闪存的访问操作； 1：使能 NOR 闪存的访问操作。
位 5：4	EXTMDBW	0x1	rw	外部存储器数据宽度（External memory data bus width） 外部存储器的数据总线宽度。 00：8 位； 01：16 位； 10：保留； 11：保留。
位 3：2	DEV	0x2	rw	存储器类型（Memory device type） 00：SRAM/ROM； 01：PSRAM（Cellular RAM 或 CRAM）； 10：NOR 闪存； 11：保留。
位 1	ADMUXEN	0x1	rw	地址/数据复用使能位（Address/data multiplexing enable）此位必须配置为 1 0：地址/数据不复用； 1：地址/数据复用数据总线。
位 0	EN	0x1	rw	存储器块使能位（Memory bank enable） 0：禁用存储器块； 1：启用存储器块。

24.5.1.2 SRAM/NOR闪存片选控制寄存器x（XMC_BK1CTRLx）x=2,4

访问：字访问

域	简称	复位值	类型	功能
位 31：20	保留	0x000	resd	保持默认值。
位 19	MWMC	0x0	rw	对存储器写操作位（Memory write mode control） 0：写操作为异步模式； 1：写操作为同步模式。
位 18：16	CRPGS	0x0	rw	CRAM 页大小选择位（CRAM page size） Cellular RAM 1.5 不允许跨页地址边界的同步访问。同步模式配置这些位时，遇到跨页时，XMC 会自动拆开访问。 000：当跨页地址边界时不会拆开访问（默认值）； 001：128 字节； 010：256 字节； 011：512 字节； 100：1024 字节；

				其他：保留。
位 15	NWASEN	0x0	rw	异步传输期间等待信号使能位（NWAIT in asynchronous transfer enable） 0：禁止 NWAIT 信号； 1：使能 NWAIT 信号。
位 14	RWTD	0x0	rw	读写时序不同控制位（Read-write timing different） 读存储器与写存储器使用不同的时序进行操作，即寄存器 XMC_BK1TMGWR 被开放。 0：读写时序相同； 1：读写时序不同。
位 13	NWSEN	0x1	rw	同步传输期间等待信号使能位（NWAIT in synchronous transfer enable） 0：禁用 NWAIT 信号； 1：使能 NWAIT 信号。
位 12	WEN	0x1	rw	写使能位（Write enable） 0：禁止； 1：使能。
位 11	NWTCFG	0x0	rw	等待时序配置（NWAIT timing configuration） 仅在同步模式有效。 0：NWAIT 信号在等待状态前的一个数据周期有效； 1：NWAIT 信号在等待状态期间有效。
位 10	WRAPEN	0x0	rw	支持非对齐的成组模式（Wrapped enable） XMC 于同步模式时是否支持将非对齐的 AHB 成组操作拆成 2 次操作； 0：不允许直接的非对齐成组操作； 1：允许直接的非对齐成组操作。
位 9	NWPOL	0x0	rw	等待信号极性（NWAIT polarity） 在同步模式下，此位设置 NWAIT 信号极性。 0：低有效； 1：高有效。
位 8	SYNCBEN	0x0	rw	同步突发模式使能（Synchronous burst enable） 允许对闪存存储器进行同步模式访问。 0：禁用； 1：使能。
位 7	保留	0x1	resd	保持默认值。
位 6	NOREN	0x1	rw	NOR 闪存访问使能（Nor flash access enable） 0：禁止 NOR 闪存的访问操作； 1：使能 NOR 闪存的访问操作。
位 5：4	EXTMDBW	0x1	rw	外部存储器数据宽度（External memory data bus width） 外部存储器的数据总线宽度。 00：8 位； 01：16 位； 10：保留； 11：保留。
位 3：2	DEV	0x0	rw	存储器类型（Memory device type） 00：SRAM/ROM； 01：PSRAM（Cellular RAM 或 CRAM）； 10：NOR 闪存； 11：保留。
位 1	ADMUXEN	0x1	rw	地址/数据复用使能位（Address/data multiplexing enable）此位必须配置为 1 0：地址/数据不复用； 1：地址/数据复用数据总线。
位 0	EN	0x0	rw	存储器块使能位（Memory bank enable） 0：禁用存储器块； 1：启用存储器块。

24.5.1.3 SRAM/NOR闪存片选时序寄存器x（XMC_BK1TMG）x=1,2,4

访问：字访问

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	保持默认值。
位 29：28	ASYNCM	0x0	rw	异步访问模式选择位（Asynchronous mode） 只在 RWTD 位使能时有效。 00：模式 A； 01：模式 B； 10：模式 C； 11：模式 D。
位 27：24	DTLAT	0xF	rw	数据延迟（Data latency） 仅在同步模式有效。 0000：额外插入 0 个 XMC_CLK 周期； 0001：额外插入 1 个 XMC_CLK 周期； …… 1111：额外插入 15 个 XMC_CLK 周期。
位 23：20	CLKPSC	0xF	rw	时钟分频系数（Clock prescale） 仅在同步模式有效，定义 XMC_CLK 时钟的频率。 0000：保留； 0001：XMC_CLK 周期为 HCLK 周期的 2 倍； 0010：XMC_CLK 周期为 HCLK 周期的 3 倍； …… 1111：XMC_CLK 周期为 HCLK 周期的 16 倍。
位 19：16	BUSLAT	0xF	rw	总线延迟时间（Bus latency） 为了防止数据总线发生冲突，在复用模式或同步模式时，一次读操作之后 XMC 在数据总线上插入延迟。 0000：插入 1 个 HCLK 周期； 0001：插入 2 个 HCLK 周期； …… 1111：插入 16 个 HCLK 周期。
位 15：8	DTST	0xFF	rw	数据建立时间（Data setup time） 00000000：额外插入 1 个 HCLK 周期； 00000001：额外插入 2 个 HCLK 周期； …… 11111111：额外插入 256 个 HCLK 周期。
位 7：4	ADDRHT	0xF	rw	地址保持时间（Address-hold time） 0000：额外插入 0 个 HCLK 周期； 0001：额外插入 1 个 HCLK 周期； …… 1111：额外插入 15 个 HCLK 周期。
位 3：0	ADDRST	0xF	rw	地址建立时间（Address setup time） 0000：额外插入 0 个 HCLK 周期； 0001：额外插入 1 个 HCLK 周期； …… 1111：额外插入 15 个 HCLK 周期。

24.5.1.4 SRAM/NOR闪存写时序寄存器x（XMC_BK1TMGWRx）x=1,2,4

访问：字访问

域	简称	复位值	类型	功能
位 31：30	保留	0x0	resd	保持默认值。
位 29：28	ASYNCM	0x0	rw	异步访问模式选择位（Asynchronous mode） 只在 RWTD 位使能时有效。 00：模式 A； 01：模式 B； 10：模式 C； 11：模式 D。
位 27：20	保留	0xFF	resd	保持默认值。

位 19：16	BUSLAT	0xF	rw	总线延迟时间（Bus latency） 为了防止数据总线发生冲突，在复用模式或同步模式时，一次读操作之后 XMC 在数据总线上插入延迟。 0000：插入 1 个 HCLK 周期； 0001：插入 2 个 HCLK 周期； …… 1111：插入 16 个 HCLK 周期。
位 15：8	DTST	0xFF	rw	数据建立时间（Data setup time） 00000000：额外插入 1 个 HCLK 周期； 00000001：额外插入 2 个 HCLK 周期； …… 11111111：额外插入 256 个 HCLK 周期。
位 7：4	ADDRHT	0xF	rw	地址保持时间（Address-hold time） 0000：额外插入 0 个 HCLK 周期； 0001：额外插入 1 个 HCLK 周期； …… 1111：额外插入 15 个 HCLK 周期。
位 3：0	ADDRST	0xF	rw	地址建立时间（Address setup time） 0000：额外插入 0 个 HCLK 周期； 0001：额外插入 1 个 HCLK 周期； …… 1111：额外插入 15 个 HCLK 周期。

24.5.1.5 SRAM/NOR额外时序寄存器x（XMC_EXTx）x=1,2,4

访问：字访问

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	保持默认值。
位 15：8	BUSLATR2R	0x08	rw	连续读操作恢复时间（Bus turnaround phase for consecutive read duration） 用于定义连续读操作间总线恢复时间，为了避免总线冲突，在连续的两次读之间将插入延迟时间。 00000000：连续读操作插入 1 个 HCLK 周期； 00000001：连续读操作插入 2 个 HCLK 周期； …… 00001000：连续读操作插入 9 个 HCLK 周期（默认值）。 …… 11111111：连续读操作插入 256 个 HCLK 周期
位 7：0	BUSLATW2W	0x08	rw	连续写操作恢复时间（Bus turnaround phase for consecutive write duration） 用于定义连续写操作间总线恢复时间。为了避免总线冲突，在连续的两次写之间将插入延迟时间。 00000000：连续写操作插入 1 个 HCLK 周期； 00000001：连续写操作插入 2 个 HCLK 周期； …… 00001000：连续写操作插入 9 个 HCLK 周期（默认值）； …… 11111111：连续写操作插入 256 个 HCLK 周期。

25 调试（DEBUG）

25.1 简介

Cortex®-M4F 内核具有丰富的调试特性。除了支持暂停和单步等标准的调试特性外，还可以利用跟踪特性查看程序执行的细节。Cortex®-M4F 内核的调试可以通过两种接口实现：串行调试接口与 JTAG 调试接口。

ARM Cortex®-M4F 内核相关资料，可参考：

- Cortex®-M4技术参考手册(TRM)
- ARM调试接口V5
- ARM CoreSight 开发工具集(r1p0版)技术参考手册

25.2 调试与跟踪功能

支持不同外设的调试，还可以设置调试时外设的工作状态。对于定时器和看门狗用户可以选择在调试时是否停止或继续计数；对于 CAN，用户可以选择在调试期间是否停止或继续更新接收寄存器；对于 I²C，用户可以选择在调试期间是否停止或继续 SMBUS 超时计数。

另外支持在低功耗模式下调试代码。在睡眠模式下，HCLK 与 FCLK 保持代码配置的时钟继续工作。在深度睡眠模式下，HICK 振荡器将开启并为 FCLK 和 HCLK 提供时钟。

MCU 内部有多个 ID 编码，调试器可通过地址为 0xE0042000 的 DEBUG_IDCODE 来访问。它是 DEBUG 的一个组成部分，并且映射到外部 PPB 总线上。使用 JTAG 调试口或 SW 调试口或通过用户代码都可以访问此编码。即使当 MCU 处于系统复位状态下这个编码也可以被访问。

25.3 I/O控制

AT32F423 在所有的封装里都支持 SWJ-DP 调试，该调试共使用 5 个普通 I/O 口。复位以后，SWJ-DP 作为默认功能可立即供调试器使用。为了防止 JTAG 的输入引脚悬空（尤其是 SWCLK/JTCK 这些时钟引脚），NJTRST、JTDI 和 JTMS/SWDIO 默认开启了内部上拉功能，JTCK/SWCLK 默认开启了内部下拉功能。

当用户切换调试接口或不使用调试功能时，可配置 GPIO 和 IOMUX 寄存器来切换这些 I/O 口功能。

25.4 DEBUG寄存器

下面列出了 DEBUG 寄存器映象和复位数值。

必须以字（32 位）的方式操作这些外设寄存器。

表 25-1 DEBUG寄存器地址和复位值

寄存器简称	基地址	复位值
DEBUG_IDCODE	0xE004 2000	0xXXXX XXXX
DEBUG_CTRL	0xE004 2004	0x0000 0000
DEBUG_APB1_PAUSE	0xE004 2008	0x0000 0000
DEBUG_APB2_PAUSE	0xE004 200C	0x0000 0000
DEBUG_SER_ID	0xE004 2020	0x0000 XX0X

25.4.1 DEBUG设备ID（DEBUG_IDCODE）

MCU 集成了 ID code，通过 ID 可以识别 MCU 的版本编号。DEBUG_IDCODE 寄存器被映射到外部 PPB 总线，基地址为 0xE0042000。使用 JTAG 调试口或 SW 调试口或用户代码都可以访问此编号。

域	简称	复位值	类型	功能
位 31：0	PID	0xXXXX XXXX	ro	PID 信息

PID [31：0]	AT32 型号	FLASH 大小	封装
0x700A_3240	AT32F423VCT7	256KB	LQFP100
0x700A_21C1	AT32F423VBT7	128KB	LQFP100
0x7003_2102	AT32F423V8T7	64KB	LQFP100
0x700A_3243	AT32F423RCT7	256KB	LQFP64
0x700A_21C4	AT32F423RBT7	128KB	LQFP64
0x7003_2105	AT32F423R8T7	64KB	LQFP64
0x700A_3246	AT32F423RCT7-7	256KB	LQFP64
0x700A_21C7	AT32F423RBT7-7	128KB	LQFP64
0x7003_2108	AT32F423R8T7-7	64KB	LQFP64
0x700A_3249	AT32F423CCT7	256KB	LQFP48
0x700A_21CA	AT32F423CBT7	128KB	LQFP48
0x7003_210B	AT32F423C8T7	64KB	LQFP48
0x700A_324C	AT32F423CCU7	256KB	QFN48
0x700A_21CD	AT32F423CBU7	128KB	QFN48
0x7003_210E	AT32F423C8U7	64KB	QFN48
0x700A_3250	AT32F423TCU7	256KB	QFN36
0x700A_21D1	AT32F423TBU7	128KB	QFN36
0x7003_2112	AT32F423T8U7	64KB	QFN36
0x700A_3253	AT32F423KCU7-4	256KB	QFN32
0x700A_21D4	AT32F423KBU7-4	128KB	QFN32
0x7003_2115	AT32F423K8U7-4	64KB	QFN32

25.4.2 DEBUG控制寄存器（DEBUG_CTRL）

寄存器由 PORESET 异步复位（不被系统复位所复位）。当内核处于复位状态下时，调试器可写。

域	简称	复位值	类型	功能
位 31：3	保留	0x0000 0000	resd	必须保持为 0。
位 2	STANDBY_DEBUG	0x0	rw	待机模式调试控制位。 0：进入待机模式时，整个 1.2V 数字电路部分都断电； 1：进入待机模式时，整个 1.2V 数字电路部分不断电，系统时钟由内部 RC 振荡器（HICK）提供时钟。
位 1	DEEPSLEEP_DEBUG	0x0	rw	深度睡眠模式调试控制位。 0：进入深度睡眠模式时，关闭所有 1.2V 域的时钟，退出深度睡眠模式时，系统时钟选择开启内部 RC 振荡器（HICK），系统时钟选择 HICK 作为系统时钟源，软件需根据应用需求重新配置系统时钟； 1：进入深度睡眠模式时，系统时钟由内部 RC 振荡器（HICK）提供。退出深度睡眠模式时，系统时钟选择 HICK 作为系统时钟源，软件需根据应用需求重新配置系统时钟。
位 0	SLEEP_DEBUG	0x0	rw	睡眠模式调试控制位 0：进入睡眠模式时，CPU HCLK 时钟关闭，其他时钟均继续运行，退出睡眠模式时，不需要重新配置时钟系统； 1：进入睡眠模式时，所有时钟都继续运行。

25.4.3 DEBUG APB1暂停控制寄存器（DEBUG_APB1_PAUSE）

寄存器由 PORESET 异步复位（不被系统复位所复位）。当内核处于复位状态下时，调试器可写。

域	简称	复位值	类型	功能
位 31：29	保留	0x0	resd	请保持为复位值。
位 28	I2C3_SMBUS_TIMEOUT	0x0	rw	I2C3 暂停控制位。 0：正常工作； 1：I2C3 SMBUS 的超时控制停止工作。
位 27	I2C2_SMBUS_TIMEOUT	0x0	rw	I2C2 暂停控制位。 0：正常工作； 1：I2C2 SMBUS 的超时控制停止工作。
位 26	CAN2_PAUSE	0x0	rw	CAN2 暂停控制位。 0：CAN2 仍然正常运行； 1：CAN2 的接收寄存器不继续接收数据。
位 25	CAN1_PAUSE	0x0	rw	CAN1 暂停控制位。 0：CAN1 仍然正常运行； 1：CAN1 的接收寄存器不继续接收数据。
位 24	I2C1_SMBUS_TIMEOUT	0x0	rw	I2C1 暂停控制位。 0：正常工作； 1：I2C1 SMBUS 的超时控制停止工作。
位 23：16	保留	0x00	resd	请保持为复位值。
位 15	ERTC_512_PAUSE	0x0	rw	ERTC 512Hz 输出时钟暂停控制位。 0：与正常模式操作相同； 1：冻结 512Hz 输出时钟。
位 14：13	保留	0x0	rw	请保持为复位值。
位 12	WDT_PAUSE	0x0	rw	看门狗暂停控制位 0：看门狗正常工作； 1：看门狗停止工作。
位 11	WWDT_PAUSE	0x0	rw	窗口看门狗暂停控制位。 0：窗口看门狗正常工作； 1：窗口看门狗停止工作。
位 10	ERTC_PAUSE	0x0	rw	ERTC 暂停控制位。 0：ERTC 正常运行； 1：ERTC 停止工作。
位 9	保留	0x0	rw	请保持为复位值。
位 8	TMR14_PAUSE	0x0	rw	TMR14 暂停控制位。 0：定时器正常工作； 1：定时器停止工作。
位 7	TMR13_PAUSE	0x0	rw	TMR13 暂停控制位。 0：定时器正常工作； 1：定时器停止工作。
位 6	TMR12_PAUSE	0x0	rw	TMR12 暂停控制位。 0：定时器正常工作； 1：定时器停止工作。

位 5	TMR7_PAUSE	0x0	rw	TMR7 暂停控制位。 0：定时器正常工作； 1：定时器停止工作。
位 4	TMR6_PAUSE	0x0	rw	TMR6 暂停控制位。 0：定时器正常工作； 1：定时器停止工作。
位 3	保留	0x0	rw	请保持为复位值。
位 2	TMR4_PAUSE	0x0	rw	TMR4 暂停控制位。 0：定时器正常工作； 1：定时器停止工作。
位 1	TMR3_PAUSE	0x0	rw	TMR3 暂停控制位。 0：定时器正常工作； 1：定时器停止工作。
位 0	TMR2_PAUSE	0x0	rw	TMR2 暂停控制位。 0：定时器正常工作； 1：定时器停止工作。

25.4.4 DEBUG APB2暂停控制寄存器（DEBUG_APB2_PAUSE）

寄存器由 PORESET 异步复位（不被系统复位所复位）。当内核处于复位状态下时，调试器可写。

域	简称	复位值	类型	功能
位 31：19	保留	0x0000	resd	请保持为复位值。
位 18	TMR11_PAUSE	0x0	rw	TMR11 暂停控制位 0：定时器正常工作； 1：定时器停止工作。
位 17	TMR10_PAUSE	0x0	rw	TMR10 暂停控制位 0：定时器正常工作； 1：定时器停止工作。
位 16	TMR9_PAUSE	00	rw	TMR9 暂停控制位 0：定时器正常工作； 1：定时器停止工作。
位 15：1	保留	0x0000	resd	请保持为复位值。
位 0	TMR1_PAUSE	0x0	rw	TMR1 暂停控制位。 0：定时器正常工作； 1：定时器停止工作。

25.4.5 DEBUG SERIES ID寄存器（DEBUG_SER_ID）

DEBUG_SER_ID 寄存器用于识别 MCU 系列和版本号。被映射到外部 PPB 总线，寄存器由 PORESET 异步复位（不被系统复位所复位）。使用 SW 调试口或用户代码都可以访问此编号。

域	简称	复位值	类型	功能
位 31：16	保留	0x0000	resd	请保持为复位值。
位 15：8	SER_ID	0xXX	ro	MCU 型号识别标识 AT32F423：0x12

位 7：3	保留	0x0000	resd	请保持为复位值。
位 2：0	REV_ID	0xX	ro	版本标识 0x0：A 版

26 版本历史

日期	版本	变更
2021.03.28	2.00	新版本发布
2023.04.25	2.01	更新首页的描述
2023.08.02	2.02	1、修订 12.1 和 12.8.3 章节描述 2、修订 14 章节部分描述
2024.06.25	2.03	1、修订整个第 6 章节描述 2、修订 14.5.2 章节 3、修订 13 章节时序图

重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和 / 或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：（A）对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）航天应用或航天环境；（D）武器，且/或（E）其他可能导致人身伤害、死亡及财产损害的应用。如果采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险及法律责任仍将由采购商单独承担，且采购商应独力负责在前述应用中满足所有法律和法规要求。经销的雅特力产品如有不同于本文档中提出的声明和 / 或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。