CM32M4xxR 系列

32 位 RISC-V 微控制器

用户手册

中移物联网有限公司

2021.10

目 录

表目录

图目录

1 文中的缩写

1.1 寄存器描述表中使用的缩写列表

在对寄存器的描述中使用了下列缩写:

read/write(rw)	软件能读写此位。
read-only(r)	软件只能读此位。
write-only(w)	软件只能写此位，读此位将返回复位值。
read/clear(rc_w1)	软件可以读此位，也可以通过写‘1’清除此位，写‘0’对此位无影响。
read/clear(rc_w0)	软件可以读此位，也可以通过写‘0’清除此位，写‘1’对此位无影响。
read/clear by read(rc_r)	软件可以读此位，读此位将自动地清除它为‘0’，写‘0’对此位无影响。
read/set(rs)	软件可以读也可以设置此位，写‘0’对此位无影响。
read-only write trigger(rt_w)	软件可以读此位，写‘0’或‘1’触发一个事件但对此位数值没有影响。
toggle(t)	软件只能通过写‘1’来翻转此位，写‘0’对此位无影响。
Reserved(Res.)	保留位，必须保持默认值不变。

1.2 可用的外设

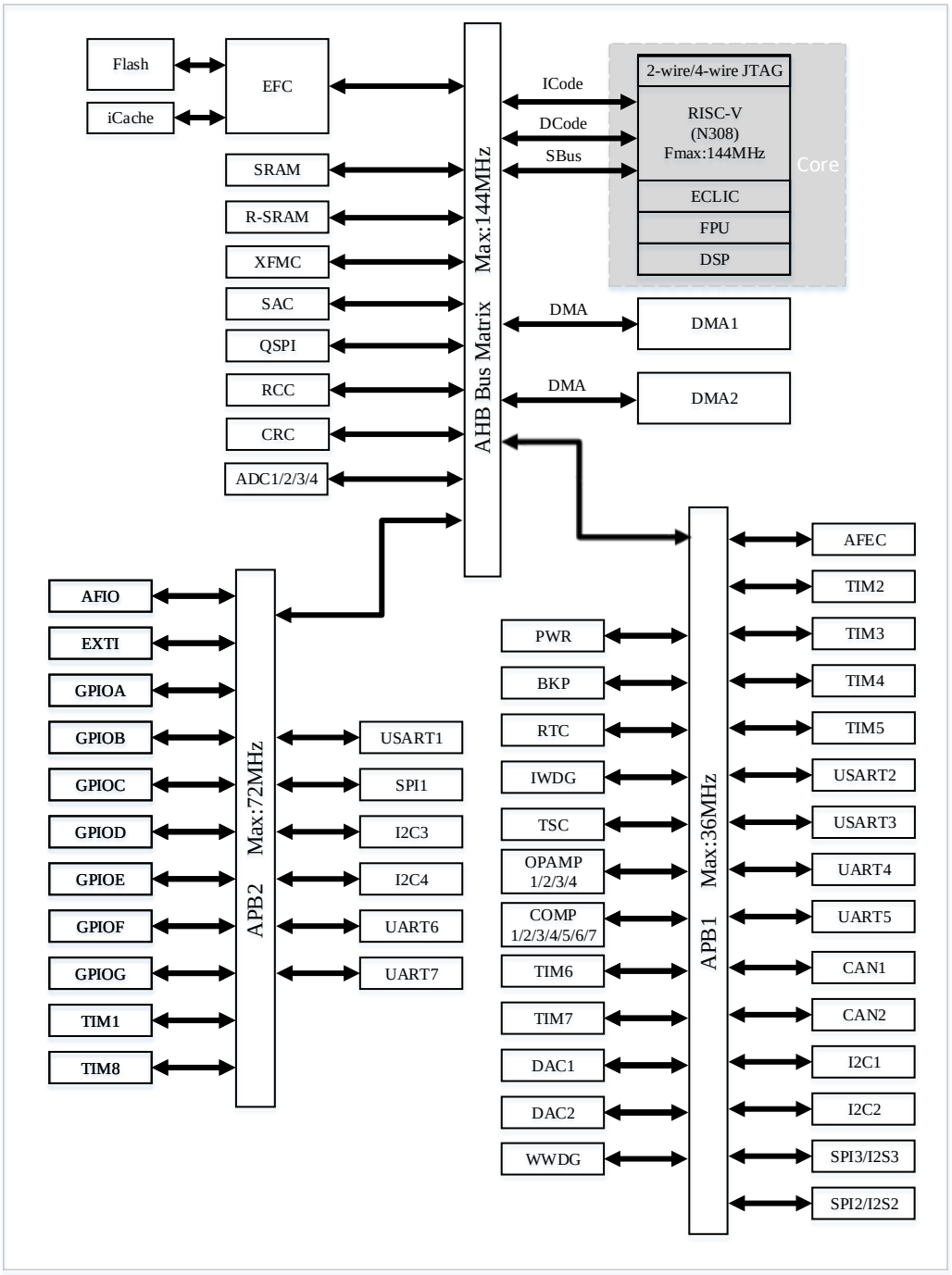
有关 CM32M4xxR 微控制器系列全部型号，某外设存在与否及其数目，请查阅相应型号的数据手册。

2 存储器和总线架构

2.1 系统架构

2.1.1 总线架构

图 2-1 总线架构图

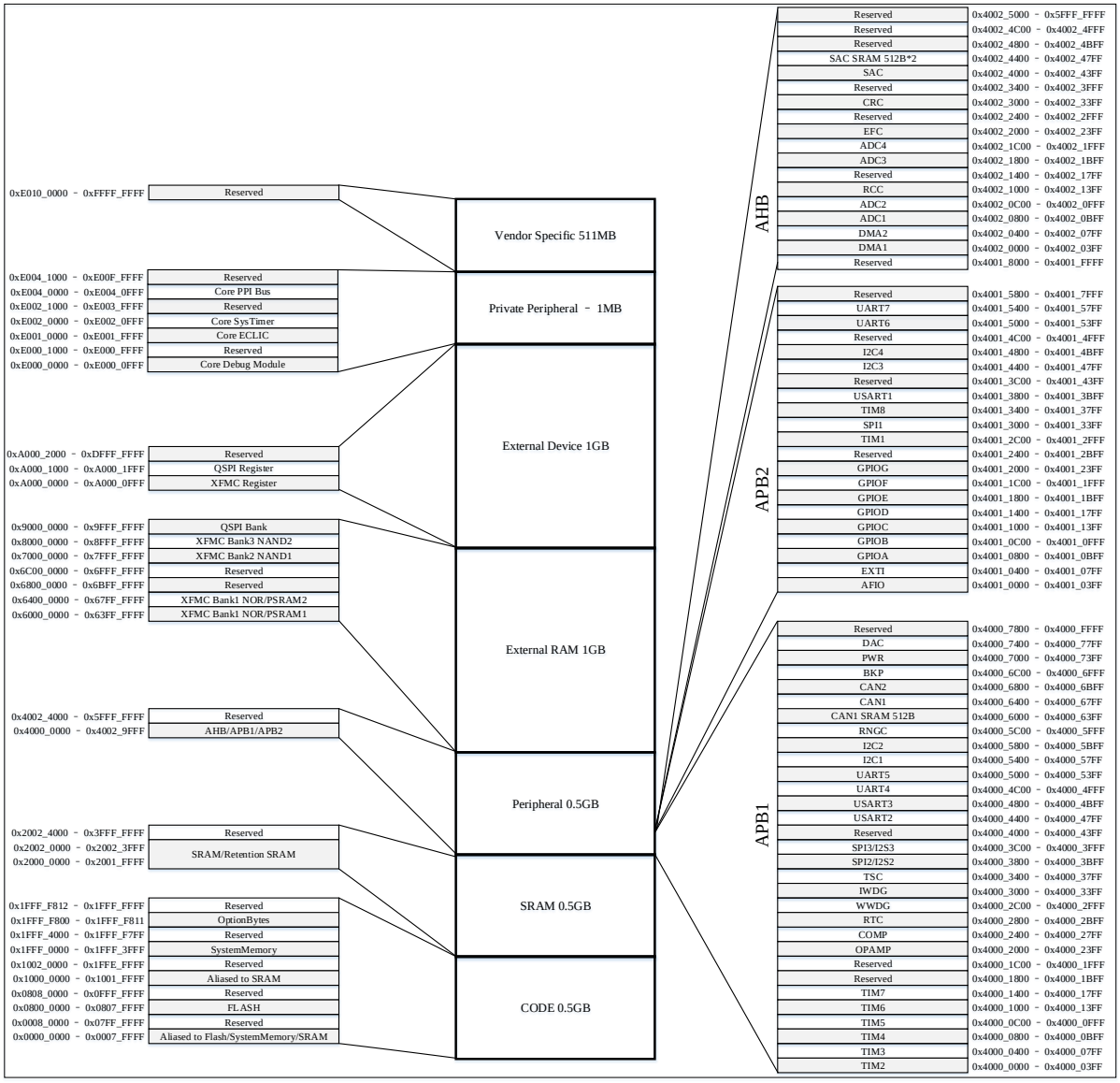


- ICode 总线将 N308 内核的 ICode 总线与闪存指令接口相连接。指令预取在此总线上完成。
- DCode 总线将 N308 内核的 DCode 总线与闪存存储器的数据接口相连接（常量加载和调试访问）。
- SBus 总线连接 N308 内核的 SBus 总线（外设总线）到总线矩阵，总线矩阵协调着内核和 DMA 间的访问。
- XFMC/SAC/CRC 没有 DMA 硬件握手，但设计了矩阵互联，支持软件触发的方式进行 DMA 传输。
- 系统包含 2 个 AHB2APB 桥，即 AHB2APB1 和 AHB2APB2。其中 APB1 包含 28 个低速 APB 外设，PCLK 的最高速度为 36MHz；APB2 包含 18 个高速 APB 外设，PCLK 最高速度等于 72MHz。

2.1.2 总线地址映射

总线地址映射包括所有 AHB 和 APB 外设：AHB 外设、APB1 外设、APB2 外设、Flash、SRAM、SystemMemory 等。具体映射如下

图 2-2 总线地址映射图



2.1.3 启动管理

2.1.3.1 启动地址

在系统启动时，可以通过 BOOT1 和 BOOT0 引脚来选择在复位后的启动模式，在系统复位后或从待机模式退出时，BOOT 引脚的值将被被重新锁存。经过启动延迟之后，CPU 从地址 0x0000_0000 开始执行代码。由于 N308 始终通过 ICode 总线从地址 0x0000_0000 开始执行程序，所以启动仅适合于从 CODE 代码区开始，设计上需要对启动空间进行地址重映射。有三种启动模式可选：

- 从主闪存存储器 Flash 启动：
  - 主闪存存储器被映射到启动空间（0x0000_0000）；

- ◆ 主闪存存储器可在两个地址区域访问，0x0000_0000 或 0x0800_0000（ICode/DCode/DMA1/DMA2）；

■ 从系统存储器 System Memory 启动：

- ◆ 系统存储器被映射到启动空间（0x0000_0000）；
- ◆ 系统存储器可在两个地址区域访问，0x0000_0000 或 0x1FFF_0000（ICode/DCode/DMA1/DMA2）；

■ 从内置 SRAM 启动：

- ◆ 内置 SRAM 被映射到启动空间（0x0000_0000）；
- ◆ 内置 SRAM 可在两个地址区域访问，0x0000_0000 或 0x2000_0000（ICode/DCode/SBus/DMA1/DMA2）；

2.1.3.2 启动配置

另外，SRAM 还可以通过虚拟地址段 0x1000_0000 进行存取访问，这使得 CPU 从 Flash 或 System Memory 启动后，可跳转到 SRAM 通过 ICode/DCode 运行程序（注意不是从 SRAM 启动程序，不属于启动模式）。除了 BOOT 引脚配置启动程序外，还有两种方式可以在 SRAM 运行程序：

■ 直接跳转到 SRAM 的物理地址段 0x2000_0000 运行程序，此时将通过 SBus 运行程序；

■ 跳转到 SRAM 的虚拟地址段 0x1000_0000，内部重映射到物理地址段 0x2000_0000 运行程序，此时将通过 ICode/DCode 高效运行程序。

表 2-1 启动模式列表

启动模式选择引脚		启动模式	对应启动模式下，访问内存空间的起始地址		
BOOT1	BOOT0		Flash	System Memory	SRAM
X	0	Flash 启动	0x0000_0000 0x0800_0000	0x1FFF_0000	0x1000_0000 0x2000_0000
0	1	System Memory 启动	0x08000000	0x0000_0000 0x1FFF_0000	0x1000_0000 0x2000_0000
1	1	SRAM 启动	0x08000000	0x1FFF_0000	0x0000_0000 0x1000_0000 0x2000_0000

2.1.3.3 自举程序

内嵌的自举程序存放在系统存储器 System Memory 内，用于通过 USART1 对闪存存储器进行重新编程，进一步的细节请查询自举程序手册。

2.2 存储系统（Memory System）

程序存储器、数据存储器、寄存器和输入输出端口被组织在同一个 4GB 的线性地址空间内。数据字节以小端格式存放在存储器中，一个字里的最低地址字节被认为是该字的最低有效字节，而最高地址字节是最高有效字节。对程序存储器和数据存储器的规格说明如下。

2.2.1 FLASH 规格

Flash 由主存储区、信息区组成，以下分别进行说明：（以下说明中的容量值不含 ECC）

- 主存储区最大为 512KB，也称作主闪存存储器，包含 256 个 Page，用于用户程序的存放和运行，以及数据存储。
- 信息区为 20KB，包含 10 个 Page，由系统存储区（16KB）、系统配置区（2KB）、选项字节区（2KB）组成：
  - 系统存储区为 16KB，包含 8 个 Page，也称作 System Memory，用于引导程序（BOOT）的存放和运行。
  - 系统配置区为 2KB，包含 1 个 Page。
  - 选项字节区为 2KB，包含 1 个 Page，也称作 OptionByte，有效空间为 18B，BOOT 程序、用户程序均可以读写擦。

2.2.1.1 闪存模块组织

主存储区、信息区都分配了总线地址空间。

表 2-2 Flash 总线地址列表

存储区	页名称	地址范围	大小
主存储区	页 0	0x0800_0000 – 0x0800_07FF	2KB
	页 1	0x0800_0800 – 0x0800_0FFF	2KB
	页 2	0x0800_1000 – 0x0800_17FF	2KB
	⋮	⋮	⋮
	页 255	0x0807_F800 – 0x0807_FFFF	2KB
信息区	系统存储区	0x1FFF_0000 – 0x1FFF_3FFF	16KB
	系统配置区	0x1FFF_F000 – 0x1FFF_F7FF	2KB
	选项字节区	0x1FFF_F800 – 0x1FFF_F811	18B
存储区接口寄存器	FLASH_AC	0x4002_2000 – 0x4002_2003	4B
	FLASH_KEY	0x4002_2004 – 0x4002_2007	4B
	FLASH_OPTKEY	0x4002_2008 – 0x4002_200B	4B
	FLASH_STS	0x4002_200C – 0x4002_200F	4B
	FLASH_CTRL	0x4002_2010 – 0x4002_2013	4B
	FLASH_ADD	0x4002_2014 – 0x4002_2017	4B
	保留	0x4002_2018 – 0x4002_201B	4B
	FLASH_OB	0x4002_201C – 0x4002_201F	4B
	FLASH_WRP	0x4002_2020 – 0x4002_2023	4B
	FLASH_ECC	0x4002_2024 – 0x4002_2027	4B
	保留	0x4002_2028 – 0x4002_202B	4B
	FLASH_RDN	0x4002_202C – 0x4002_202F	4B
	FLASH_CAH	0x4002_2030 – 0x4002_2033	4B

闪存存储器被组织成 32 位宽的存储器单元，可以存放代码和数据常数。

信息区分为三个部分：

- 系统存储区是用于存放在系统存储器自举模式下的启动程序，启动程序使用 USART1 串行接口实现对闪存存储器的编程。
- 系统配置区，包含芯片基本信息。
- 选项字节区，对主存储器和信息块的写入由内嵌的闪存编程/擦除控制器管理。

闪存存储器有两种保护方式防止非法的访问（读、写、擦除）：

- 页写入保护（WRP）
- 读出保护（RDP）

在执行闪存写操作时，任何对闪存的读操作都会锁住总线，在写操作完成后读操作才能正确地进行；即在进行写或擦除操作时，不能进行代码或数据的读取操作。

进行闪存编程操作时（写或擦除），必须打开内部的 RC 振荡器（HSI）。

注：在低功耗模式下，所有闪存存储器的操作都被中止。

2.2.1.2 读写操作

Flash 写操作仅支持 32 位操作，写操作之前先擦除 Flash，擦除最小块大小是一个 Page 2KB。写操作分为编程和擦除阶段。

读 Flash 时，读的等待周期数可以通过寄存器配置。使用时，需要结合 AHB 接口时钟频率进行计算，等待时间必须不小于 25ns。比如：当 HCLK<=32MHz 时，等待周期数最小为 0；当 32MHz<HCLK<=64MHz 时，等待周期数最小为 1；当 64MHz<HCLK<=96MHz 时，等待周期数最小为 2；当 96MHz<HCLK<=128MHz 时，等待周期数最小为 3；当 128MHz<HCLK<=144MHz 时，等待周期数最小为 4。

2.2.1.3 Flash 解锁

复位后，Flash 模块是被保护的，不能写入 FLASH_CTRL 寄存器，以防因电气干扰等原因产生对 Flash 的意外操作。通过写入特定的键值序列到 FLASH_KEY 寄存器，可以开启对 FLASH_CTRL 寄存器的操作权限，这个特定的序列是：第一次在 Flash 密钥寄存器（FLASH_KEY）中写入 KEY1 = 0x45670123，第二次则在 Flash 密钥寄存器（FLASH_KEY）中写入 KEY2 = 0xCDEF89AB。

如果顺序出现错误或键值出现错误，将返回总线错误并锁定 FLASH_CTRL 寄存器，直到下一次复位，软件可以通过查看 FLASH_CTRL 寄存器中的 LOCK 位来确认 Flash 是否已解锁。若需要进行正常的锁定设置，可以通过软件将 FLASH_CTRL 寄存器中的 LOCK 位置 1 来实现，此后可以通过在 FLASH_KEY 中写入正确的键值系列来对 Flash 解锁。

2.2.1.4 擦除和编程

2.2.1.4.1 主存储区擦除

主存储区可以按页擦除，也可以整片擦除。

页擦除

擦除一页应遵守下述过程：

- 检查 FLASH_STS 寄存器的 BUSY 位，以确认没有其他正在进行的闪存操作；

- 设置 FLASH_CTRL 寄存器的 PER 位为'1';
- 用 FLASH_ADD 寄存器选择要擦除的页;
- 设置 FLASH_CTRL 寄存器的 START 位为'1';
- 等待 BUSY 位变为'0';
- 如果需要，读出被擦除的页并做验证。

整片擦除

可以用整片擦除功能擦除整个主存储区。建议使用下述过程:

- 检查 FLASH_STS 寄存器的 BUSY 位，以确认没有其他正在进行的闪存操作;
- 设置 FLASH_CTRL 寄存器的 MER 位为'1';
- 设置 FLASH_CTRL 寄存器的 START 位为'1';
- 等待 BUSY 位变为'0';
- 读出所有页并做验证。

2.2.1.4.2 主存储区编程

对主存储区编程每次可以写入 32 位。当 FLASH_CTRL 寄存器的 PG 位为'1'时，在一个闪存地址写入一个字将启动一次编程；写入任何半字的数据，都会产生总线错误。在编程过程中(BUSY 位为'1')，任何读写闪存的操作都会使 CPU 暂停，直到此次闪存编程结束。

主存储区编程过程:

- 检查 FLASH_STS 寄存器的 BUSY 位，以确认没有其他正在进行的闪存操作;
- 设置 FLASH_CTRL 寄存器的 PG 位为'1';
- 在指定的地址写入要编程的字;
- 等待 BUSY 位变为'0';
- 读出写入的地址并验证数据。

注意： 当 FLASH_STS 寄存器的 BUSY 位为'1'时，不能对任何寄存器执行写操作。

2.2.1.4.3 选项字节区擦除和编程

对选项字节区的编程与主存储区不同。选项字节的数目只有 9 个字节(4 个字节作为写保护，2 个字节作为读保护，1 个字节为配置选项，2 个字节存储用户数据)。对 Flash 解锁后，必须分别写入 KEY1 和 KEY2(见 2.2.1.3)到 FLASH_OPTKEY 寄存器，从而解锁 FLASH_CTRL 寄存器的 OPTWE 位为'1'，此时可以对选项字节区进行编程：设置 FLASH_CTRL 寄存器的 OPTPG 位为'1'后写入字到指定的地址。

对选项字节区字编程时，使用半字中的低字节并自动地计算出高字节(高字节为低字节的反码)，并开始编程操作，这将保证选项字节和它的反码始终是正确的。

选项字节区擦除过程:

- 检查 FLASH_STS 寄存器的 BUSY 位，以确认没有其他正在进行的闪存操作;
- 解锁 FLASH_CTRL 寄存器的 OPTWE 位;

- 设置 FLASH_CTRL 寄存器的 OPTER 位为'1';
- 设置 FLASH_CTRL 寄存器的 START 位为'1';
- 等待 BUSY 位变为'0';
- 读出被擦除的选项字节并做验证。

选项字节区编程过程:

- 检查 FLASH_STS 寄存器的 BUSY 位，以确认没有其他正在进行的闪存操作;
- 解锁 FLASH_CTRL 寄存器的 OPTWE 位;
- 设置 FLASH_CTRL 寄存器的 OPTPG 位为'1';
- 写入要编程的字到指定的地址;
- 等待 BUSY 位变为'0';
- 读出写入的地址并验证数据。

2.2.1.5 ECC 功能

Flash 模块支持 ECC 功能，实现 1-bit 检错和 1-bit 纠错。ECC 编码、解码（纠错、检错）由硬件自动执行，如果检测到错误，置错误位并产生中断。

2.2.1.6 指令预取

Flash 模块的指令预取功能，支持 16B 的预取 Buffer。通过指令预取操作，可提高 CPU 的指令执行效率。指令预取功能可以通过寄存器配置为使能或除能，默认使能。

2.2.1.7 选项字节

选项字节块主要用于配置读写保护、软件/硬件看门狗以及系统处于待机或停机模式下的复位选项，并分配了总线地址空间，可以进行读写访问。它们由有 9 个选项字节组成：4 个字节作为写保护，2 个字节作为读保护，1 个字节作为配置选项，2 个字节由用户定义，这 9 个字节需要通过总线写入。选项字节块同时还包含与这 9 个选项字节相对应的反码，这些反码需要在总线写入选项字节时，由硬件自动计算出来，一起写入 Flash，并用于选项字节读取时的验证。

默认状态下，选项字节块始终是可以读且被写保护。要想对选项字节块进行写操作（编程/擦除），首先要解锁 Flash，然后解锁选项字节：在 OPTKEY 中写入正确的键值序列（KEY1 = 0x45670123，KEY2 = 0xCDEF89AB），随后对选项字节块的写操作将被允许。如果顺序出现错误或键值出现错误，将返回总线错误并锁定选项字节，直到下一次复位。若需要正常进行锁定设置，可以通过软件将 FLASH_CTRL 寄存器中的 OPTWE 位写 0 来实现，此后可以通过在 FLASH_OPTKEY 中写入正确的键值系列来对选项字节解锁。

每次系统复位后，从 Flash 的选项字节块中读出选项字节数据，并保存在具有只读属性的选项字节寄存器（FLASH_OB/FLASH_WRP）中；同时一起读出来的选项字节反码数据，将用于验证选项字节数据是否正确，如果不匹配，将产生一个选项字节错误标志（OBERR）。当发生选项字节错误时，对应的选项字节被强置为 0xFF。当选项字节和它的反码均为 0xFF 时（擦除后的状态），则略过上述验证步骤，无需进行验证。

表 2-3 选项字节列表

地址	[31:24] 相应反码	[23:16] 选项字节	[15:8] 相应反码	[7:0] 选项字节
0x1FFF_F800	nUSER	USER	nRDP1	RDP1
0x1FFF_F804	nData1	Data1	nData0	Data0
0x1FFF_F808	nWRP1	WRP1	nWRP0	WRP0
0x1FFF_F80C	nWRP3	WRP3	nWRP2	WRP2
0x1FFF_F810	-	-	nRDP2	RDP2

- 读保护 L1 等级选项字节：RDP1
  - 保护存储在闪存中的代码；
  - 当写入正确的数值时，将禁止读出闪存存储器；
  - RDP1 是否开启的结果，可通过 FLASH_OB[1]查询；
- 配置选项：USER
  - USER[7:3]：Reserved；
  - USER[2]：nRST_STDBY 配置选项，可通过 FLASH_OB[4]查询

    0：当进入待机模式时产生复位

    1：进入待机模式时不产生复位
  - USER[1]：nRST_STOP 配置选项，可通过 FLASH_OB[3]查询

    0：当进入停机（STOP0）模式时产生复位

    1：进入停机（STOP0）模式时不产生复位
  - USER[0]：WDG_SW 配置选项，可通过 FLASH_OB[2]查询

    0：硬件看门狗

    1：软件看门狗
- 2 个字节的用户数据：Datax
  - Data1（存储在 FLASH_OB[25:18]）；
  - Data0（存储在 FLASH_OB[17:10]）；
- 写保护选项字节：WRP0~3，可通过寄存器 FLASH_WRP[15:0]查询
  - WRP0：第 0 ~15 页的写保护，bit[0]对应 Page0/1，……，bit[7]对应 Page14/15；
  - WRP1：第 16~31 页的写保护，bit[0]对应 Page16/17，……，bit[7]对应 Page30/31；
  - WRP2：第 32~47 页的写保护，bit[0]对应 Page32/33，……，bit[7]对应 Page46/47；
  - WRP3：第 48~255 页的写保护，bit[0]对应 Page48/49，……，bit[6]对应 Page60/61，bit[7]对应 Page62~255；
- 读保护 L2 等级选项字节：RDP2

- 在 L1 的基础上增加保护功能，具体见 2.2.1.9 读保护的详细描述；
- RDP2 是否开启的结果，可通过 FLASH_OB[31]查询；

2.2.1.8 写保护

可以对 Flash 主存储区（最大 512KB）的所有 Page 配置写保护，以防在程序跑飞或电气干扰等原因导致的意外写操作，写保护的基本单位是：对于 Page0~61，每 2 页为一个基本保护单元，对于 Page62~255，共同作为一个保护单元。写保护可以通过设置选项字节块中的 WRP0~3 来进行配置；每次进行配置后，需要进行一次系统复位，配置的值才能被重新加载生效。如果试图在一个受保护的页面进行编程或擦除操作，在闪存状态寄存器（FLASH_STS）中将会返回一个保护错误标志。

系统信息区中的系统存储块（16KB），存放了 BOOT 程序，不可更改。

系统信息区中的系统配置块（2KB），存放了芯片基本信息，不可更改。

系统信息区中的选项字节块（2KB），存放了用户可配置选项字节信息，选项字节块的写保护，则是通过软件将 FLASH_CTRL 寄存器中的 OPTWE 位写 0 来实现，此后可以通过在 FLASH_OPTKEY 中写入正确的键值系列，来对选项字节解除写保护。封口前后、任意用户均可访问。

2.2.1.9 读保护

Flash 中的用户代码可以通过设置读保护来防止被非法读取。读保护主要是针对芯片完成封口操作后，保护主存储区和选项字节块的访问操作。读保护通过配置选项字节块中的 RDP 字节进行设置，可以配置 3 种不同的读保护级别，如下列表

表 2-4 读保护配置列表

读保护状态	RDP1	nRDP1	RDP2	nRDP2
L1 级别	0xFF	0xFF	RDP2! = 0xCC || nRDP2! = 0x33	
未保护	0xA5	0x5A	RDP2! = 0xCC || nRDP2! = 0x33	
L2 级别	0xXX	0xXX	0xCC	0x33
L1 级别	非以上 3 种配置			

- L0 级别：
  - 处于未保护状态，对应的(RDP1=0xA5 & nRDP1=0x5A) && (RDP2!=0xCC | nRDP2!=0x33)；
  - 主存储区和选项字节块可以被任意读取；
  - 可以配置各 Page 的写保护属性，以进行编程和擦除；
- L1 级别：
  - 对应的~(((RDP1=0xA5 & nRDP1=0x5A) && (RDP2!=0xCC | nRDP2!=0x33)) || (RDP2=0xCC & nRDP2=0x33))；
  - 只允许从用户代码中对主存储区的读操作，即以非调试方式从主闪存存储器启动程序的情况才允许对主存储区的读操作；
  - 第 0~1 页被自动加上了写保护；
  - 其它 Page 可以通过在主闪存存储器中执行的代码进行编程（实现 IAP 或数据存储等功能）；
  - 所有 Page 不允许在调试模式下或从内部 SRAM 启动后执行写或擦除操作（整片擦除除外）；

保护级别	启动模式 / 执行用户访问区域	Flash（BOOT0 = 0，BOOT1 = 0）				修改保护级别
		2-wire/4-wire JTAG	Flash启动[1]	跳转到System Memory	跳转到SRAM	
L0级别	Flash主存储区4KB前	读写擦[2]	读写擦	读写擦	读写擦	允许改为L1或L2
	Flash主存储区4KB后	读写擦	读写擦	读写擦	读写擦	
	Flash主存储区片擦[3]	允许	允许	允许	允许	
	Flash选项字节区	读写擦	读写擦	读写擦	读写擦	
	Flash系统存储区	禁止	禁止	读写擦	禁止	
	系统配置区	仅id[4]可读	仅id可读	id可读，分区信息可读[5]	仅id可读	
	SRAM（All）	读写	读写	读写	读写	
L1级别	Flash主存储区4KB前	禁止	只读	只读	只读	允许改为 L0 或 L2。改为L0时，主存储区将被自动擦除。
	Flash主存储区4KB后	禁止	读写擦	读写擦	读写擦	
	Flash主存储区片擦除	允许	允许	允许	允许	
	Flash选项字节区	读写擦	读写擦	读写擦	读写擦	
	Flash系统存储区	禁止	禁止	读写擦	禁止	
	系统配置区	禁止	仅id可读	id可读，分区信息可读	仅id可读	
	SRAM（All）	读写	读写	读写	读写	
L2级别	Flash主存储区4KB前	2-wire/4-wire JTAG接口被禁止	只读	只读	只读	不允许修改。
	Flash主存储区4KB后		读写擦	读写擦	读写擦	
	Flash主存储区片擦除		允许	允许	允许	
	Flash选项字节区		只读	只读	只读	
	Flash系统存储区		禁止	读写擦	禁止	
	系统配置区		仅id可读	id可读，分区信息可读	仅id可读	
	SRAM（All）		读写	读写	读写	
保护	启动模式	SRAM（BOOT0 = 1，BOOT1 = 1）				修改保护级别

级别	执行用户 / 访问区域	2-wire/4-wire JTAG	跳转到 Flash	跳转到 System Memory	SRAM 启动	
L0级别	Flash主存储区 4KB前	读写擦	读写擦	读写擦	读写擦	允许改为L1或L2
	Flash主存储区 4KB后	读写擦	读写擦	读写擦	读写擦	
	Flash主存储区片擦除	允许	允许	允许	允许	
	Flash选项字节区	读写擦	读写擦	读写擦	读写擦	
	Flash系统存储区	禁止	禁止	读写擦	禁止	
	系统配置区	仅id可读	仅id可读	id可读，分区信息可读	仅id可读	
	SRAM（All）	读写	读写	读写	读写	
L1级别	Flash主存储区 4KB前	禁止	只读	只读	禁止	允许改为 L0 或 L2。改为L0时，主存储区将被自动擦除。
	Flash主存储区 4KB后	禁止	读写擦	读写擦	禁止	
	Flash主存储区片擦除	允许	允许	允许	允许	
	Flash选项字节区	读写擦	读写擦	读写擦	读写擦	
	Flash系统存储区	禁止	禁止	禁止	禁止	
	系统配置区	禁止	仅id可读	id可读 分区信息可读	禁止	
	SRAM（All）	读写	读写	读写	读写	
L2级别	Flash主存储区 4KB前	L2保护级别，无法从SRAM启动				不允许修改。2-wire/4-wire JTAG 被禁止。
	Flash主存储区 4KB后					
	Flash主存储区片擦除					
	Flash选项字节区					
	Flash系统存储区					
	系统配置区					
	SRAM（All）					
保护级别	启动模式	System Memory（BOOT0 = 1，BOOT1 = 0）				修改保护级别
	执行用户 / 访问区域	2-wire/4-wire JTAG	跳转到 Flash	System Memory 启动	跳转到 SRAM	

L0级别	Flash主存储区4KB前	读写擦	读写擦	读写擦	读写擦	允许改为L1或L2
	Flash主存储区4KB后	读写擦	读写擦	读写擦	读写擦	
	Flash主存储区片擦除	允许	允许	允许	允许	
	Flash选项字节区	读写擦	读写擦	读写擦	读写擦	
	Flash系统存储区	禁止	禁止	读写擦	禁止	
	系统配置区	仅id可读	仅id可读	id可读，分区信息可读写(6)	仅id可读	
	SRAM（All）	读写	读写	读写	读写	
L1级别	Flash主存储区4KB前	禁止	只读	只读	只读	允许改为 L0 或 L2。 改为L0时，主存储区将被自动擦除。
	Flash主存储区4KB后	禁止	读写擦	读写擦	读写擦	
	Flash主存储区片擦除	允许	允许	允许	允许	
	Flash选项字节区	读写擦	读写擦	读写擦	读写擦	
	Flash系统存储区	禁止	禁止	读写擦	禁止	
	系统配置区	禁止	仅id可读	id可读，分区信息可读写	仅id可读	
	SRAM（All）	读写	读写	读写	读写	
L2级别	Flash主存储区4KB前	2-wire/4-wire JTAG接口被禁止	只读	只读	只读	不允许修改
	Flash主存储区4KB后		读写擦	读写擦	读写擦	
	Flash主存储区片擦除		允许	允许	允许	
	Flash选项字节区		只读	只读	只读	
	Flash系统存储区		禁止	读写擦	禁止	
	系统配置区		仅id可读	id可读，分区信息可读写	仅id可读	
	SRAM（All）		读写	读写	读写	

- 所有通过 2-wire/4-wire JTAG 向内置 SRAM 装载代码并执行代码的功能依然有效，亦可以通过 2-wire/4-wire JTAG 从内置 SRAM 启动，这个功能可以用来解除读保护；
- 当读保护的选项字节被改写为未保护的 L0 级别时，将会自动擦除全部主存储区，执行的过程如下：（擦除选项字节块不会导致自动的整片擦除操作，因为擦除的结果是 0xFF，相当于仍然处于 L1 级别的保护状态）

- 在 OPTKEY 中写入正确的键值序列解锁选项字节区；
- 总线发起命令擦除整个选项字节区（Page 擦）；
- 总线写入读保护选项字节 0xA5；
- 内部自动擦除全部主存储区；
- 内部自动写入 0xA5 到读保护选项字节；
- 进行系统复位（如软件复位等），选项字节块（包括新的 RDP 值 0xA5）将被重新加载到系统中，读保护被解除；

◆ 以下对闪存的访问操作都将被禁止：

- 通过从内置 SRAM 或从 XFMC 执行代码（包括使用 DMA）访问主闪存存储器；
- 通过 JTAG 和边界扫描方式访问主闪存存储器；

■ L2 级别：除了调试模式被禁止、保护级别不可修改（不可逆）之外，其余特性同 L1 级别。L2 级别通过配置另一个选项字节 RDP2 来实现，不管 RDP1 为何值，只要满足（RDP2=0xCC & nRDP2=0x33）即为 L2 级别。

表 2-5 Flash 读写擦权限控制表

注：1.用户选择 BOOT0/1 的电平，决定启动模式；
2.这里的擦是指 Flash 页擦除；
3.Flash 主存储区片擦除是指 mass erase；
4.id 包含 UCID、Chip ID、DBGMCU_IDCODE；
5.分区信息是指 MMU 模块的用户分区大小信息，用户分区信息也可以通过 MMU 的寄存器读到；
6.分区信息只能写一次。

2.2.2 iCache

为了达到更高的系统性能，高速 CPU 与低速 Flash 之间需要增加指令缓存器，以提高指令执行效率。由于指令缓存器的存在，CPU 将可以工作在更高的主频。当 CPU 请求的指令在指令缓存器里面时，CPU 将可以无延时地获得指令、实现零等待执行。当前指令序列、指令预取序列、指令缓存器均未命中时，将重新读取 Flash，并回填更新 Cache 缓存；依此，相当于 Cache 中只存储了程序的跳转头。

指令缓存器的主要特性如下：

■ 8KB iCache。

■ 支持相联方式：4WAY。

2.2.2.1 软件接口

■ 使能

◆ 提供软件使能/关闭 Icache 的配置。开关切换条件无限制（见 FLASH_AC ICAHEN 位）。

■ 复位

◆ 提供软件清空 Icache 接口，必须在 Icache 关闭时才发起。复位与切换不可同时切换，先关闭 ICAHEN，然后 ICAHRST 写 1，然后就可打开 ICAHEN。

- 锁定
  - 支持 cache 锁定机制，软件配置将程序放入其指定的 way 中。当所有 way 均锁定完成后，新的数据将不会写入 cache 中。软件复位 cache 后，锁定状态自动清除。
- 补充说明
  - 不支持 Cache 替换算法选择。
  - 为指令 Cache，不存在 CPU 写操作时 WB/WT 选择。

2.2.2.2 寄存器描述

FLASH_AC 寄存器中 ICAHEN 及 FLASH_AC 寄存器中 ICAHRST，其分别为 iCache 使能开关以及 icache 数据清零开关。

FLASH_CAH 寄存器中 LOCKSTRT 及 FLASH_CAH 寄存器中 LOCKSTOP，其分别为 iCache 对应方式锁定的开始锁存和停止锁存。iCache 复位后，FLASH_CAH 寄存器自动恢复为复位值。iCache 锁定的详细使用方法见 2.2.2.3.3**iCache 锁定**。

2.2.2.3 操作流程

2.2.2.3.1 iCache 启用与禁用

用户软件可以随时开关 iCache。如果用户程序需要在主存储区和其它存储区之间跳转时，必须关闭 iCache 并且将 iCache 数据清零，否者会产生指令获取错误。

2.2.2.3.2 iCache 数据刷新

iCache 设计为指令 cache，当指令被应用软件更新或者指令在主存储区和其它存储区之间跳转时，软件必须将 FLASH_AC 寄存器的 ICAHRST 位置 1 来清除指令 cache 内的数据。注意：ICAHRST 位是只写位，读该位时返回为 0。

2.2.2.3.3 iCache 锁定

用户软件控制 FLASH_CAH 寄存器来将一些重复使用的代码锁存在 iCache 中来提高代码执行的效率。iCache 模块有 4 个锁存通道，每个通道的大小为整个 cache 的 1/4。在使用单个通道时，必须确保需要锁存的代码量小于每个通道的大小。否者需要用更多的通道来锁存代码。可以按照下面的控制流程来使用锁存功能：

1. 将 LOCKSTRT[0]置 1；
2. 执行需要锁存在通道 0 的函数 1（函数 1 的代码量应该小于单个通道的大小）；
3. 函数 1 执行完成后，将 LOCKSTOP[0]置 1；
4. 接着将 LOCKSTRT[1]置 1；
5. 执行需要锁存在通道 1 的函数 2（函数 2 的代码量应该小于单个通道的大小）；
6. 函数 2 执行完成后，将 LOCKSTOP[1]置 1；

注意：1.通道锁存时寄存器操作必须要按照固定的流程 – 先设置 LOCKSTRT – 再设置 LOCKSTOP;
2.通道锁存的顺序必须要按照 0~3，否者会降低执行效率。

2.2.3 SRAM

SRAM 主要用于代码运行，存放程序执行过程中的变量和数据或堆栈，容量最大为 128KB。

SRAM 支持字节、半字、字的读写访问。

SRAM 支持代码运行（支持 SBus 和 ICode、DCode 的访问），可以在 SRAM 全速运行程序。SRAM 的最大地址范围是 0x2000 0000~0x2001 FFFF。

SRAM 在 Stop2、Standby 和 VBAT 模式下数据不能保持；其他工作模式（Run/Sleep/Stop0）数据可以正常保持。

主要特性如下:

- 容量最大总共为 128KB
- 支持字节/半字/字读写
- I/D/S/DMA1/DMA2 均可访问
- I/D BUS 可以 Remap 到 SRAM 全速运行程序

2.2.4 R-SRAM（Retention SRAM）

R-SRAM 也是主要用于代码运行，存放程序执行过程中的变量和数据或堆栈，容量总共为 16KB。R-SRAM 的总线地址与 SRAM 是连续相接的，应用上，可以把 SRAM 和 R-SRAM 当做一块 SRAM 来处理。最大情况下，R-SRAM 的 0 号物理地址对应总线起始地址为 0x2002 0000，对应的总线地址范围为 0x2002 0000~0x2002 3FFF。R-SRAM 支持字节、半字、字的读写访问，支持 SBus、DMA1、DMA2 的访问。

因为 R-SRAM 的总线地址与 SRAM 是连续相接的，而对于不同的产品型号，SRAM 可供有效使用的容量是不同的，所以对于不同的产品型号来说，R-SRAM 的总线起始地址是不一样的。

R-SRAM 支持 Retention，在 Stop2、Standby 模式下可以保持数据（可以配置为保持或不保持）；VBAT 模式下会掉电，同时丢失数据；其他工作模式（Run/Sleep/Stop0）下会一直带电，数据可以正常保持；需要 PWR 对其 Retention 进行控制管理。

表 2-6 SRAM 容量配置表

SRAM 容量	R-SRAM 容量	SRAM 总线地址范围	R-SRAM 总线地址范围
64KB	16KB	0x2000 0000~0x2000 FFFF	0x2001 0000~0x2001 3FFF
128KB	16KB	0x2000 0000~0x2001 FFFF	0x2002 0000~0x2002 3FFF

R-SRAM 主要特性如下:

- 容量总共为 16KB
- 字节/半字/字读写
- S/DMA1/DMA2 均可访问
- 总线起始地址与主存储器 SRAM 连续相接
- 总线起始地址跟随主存储器 SRAM 容量变化

- 可以 Retention，在 Stop2 和 Standby 模式下仍需保持数据（可配置为不保持）

2.2.5 FLASH 寄存器描述

必须以字（32 位）的方式操作这些外设寄存器。

2.2.5.1 FLASH 寄存器地址映像

表 2-7 FLASH 寄存器地址映像和复位值

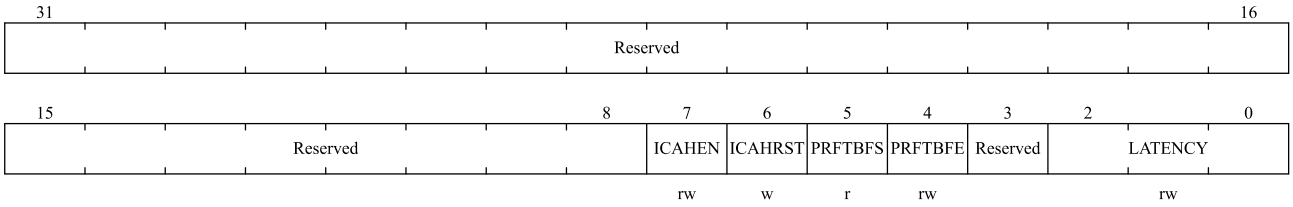
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	FLASH_AC	Reserved																								ICAHEN	ICAHRST	PRFTBFS	PRFTBFE	Reserved	LATENCY		
	Reset Value																									0	0	1	1		0	0	0
004h	FLASH_KEY	FKEY																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	FLASH_OPTKEY	OPTKEY																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	FLASH_STS	Reserved																								ECCERR	EVERR	EOP	WRPERR	PVERR	PGERR	Reserved	BUSY
	Reset Value																									0	0	0	0	0	0		0
010h	FLASH_CTRL	Reserved																		ECCERRITE	EOPITE	FERRITE	ERRITE	OPTWE	SMPSEL	LOCK	START	OPTER	OPTPG	Reserved	MER	PER	PG
	Reset Value																			0	0	0	0	0	0	1	0	0	0		0	0	0
014h	FLASH_ADD	FADD																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	Reserved																																
01Ch	FLASH_OB	RDPRT2	Reserved						Data1								Data0									Reserved				nRST_STDBY	nRST_STOP	WDG_SW	RDPRT1	OBERR
	Reset Value	0							1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					1	1	1	0	0
020h	FLASH_WRP	WRPT																															
	Reset Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
024h	FLASH_ECC	Reserved																		ECCHW						Reserved	ECCLW						
	Reset Value																			0	0	0	0	0	0		0	0	0	0	0	0	0
028h	Reserved																																
02Ch	FLASH_RDN	Reserved							FLASH_RDN1									Reserved								FLASH_RDN0							
	Reset Value								0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0	0
030h	FLASH_CAH	Reserved																								LOCKSTOP				LOCKSTRT			
	Reset Value																									0	0	0	0	0	0	0	0

2.2.5.2 FLASH 控制和状态寄存器

有关寄存器说明中的缩写，参见 1.1 节。

2.2.5.2.1 FLASH 访问控制寄存器（FLASH_AC）

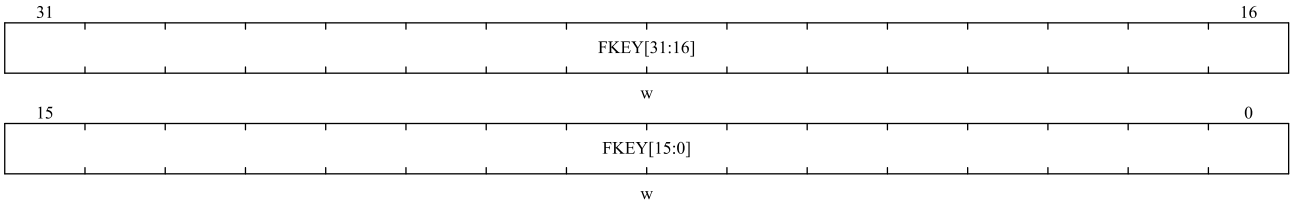
地址偏移量：0x00
复位值：0x0000 0030



位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7	ICAHEN	Icache 使能 0：关闭 Icache； 1：启用 Icache。
6	ICAHRST	Icache 复位 0：写'0'无效； 1：写'1'复位。
5	PRFTBFS	预取缓冲区状态 该位指示预取缓冲区的状态 0：预取缓冲区关闭； 1：预取缓冲区开启。
4	PRFTBFE	预取缓冲区使能 0：关闭预取缓冲区； 1：启用预取缓冲区。
3	Reserved	保留，写'0'或者'1'无效。
2:0	LATENCY	时延 这些位表示 HCLK（系统 AHB 时钟）周期与闪存访问时间的比例 000：零周期时延，当 0 < HCLK <= 32MHz 001：一个周期时延，当 32MHz < HCLK <= 64MHz 010：两个周期时延，当 64MHz < HCLK <= 96MHz 011：三个周期时延，当 96MHz < HCLK <= 128MHz 100：四个周期时延，当 128MHz < HCLK <= 144MHz

2.2.5.2.2 FLASH 键寄存器（FLASH_KEY）

地址偏移量：0x04
复位值：0x0000 0000

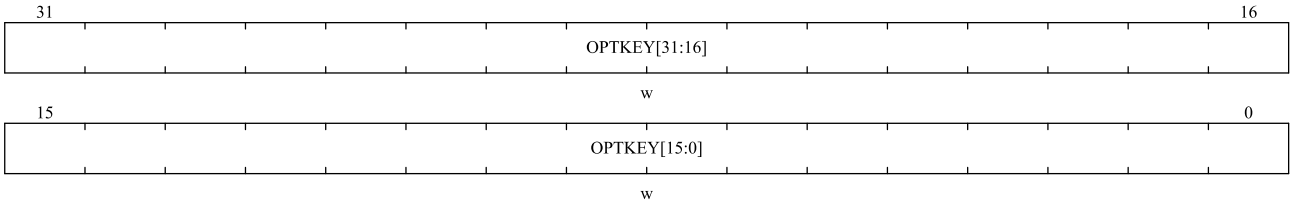


位域	名称	描述
31:0	FKEY	用于解锁 FLASH_CTRL 寄存器的'LOCK'位。

2.2.5.2.3 FLASH OPTKEY 寄存器（FLASH_OPTKEY）

地址偏移量：0x08

复位值：0x0000 0000

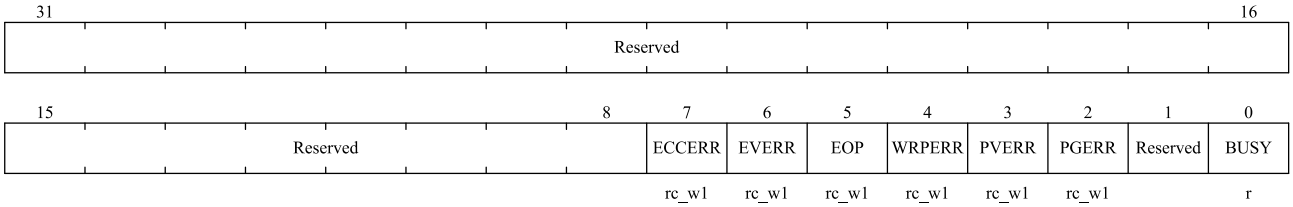


位域	名称	描述
31:0	OPTKEY	用于解锁 FLASH_CTRL 寄存器的'OPTWE'位。

2.2.5.2.4 FLASH 状态寄存器（FLASH_STS）

地址偏移量：0x0C

复位值：0x0000 0000



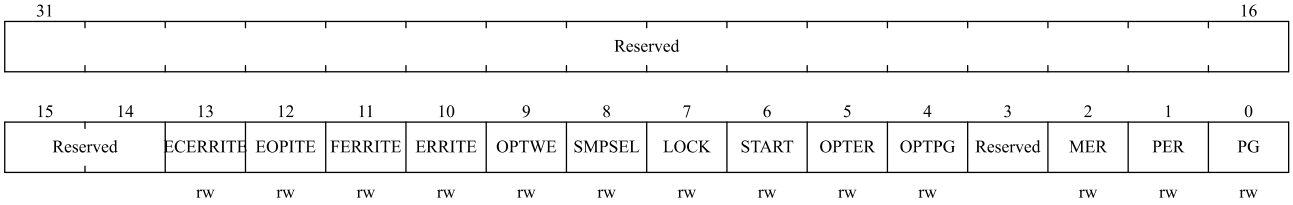
位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7	ECCERR	ECC 错误 读 FLASH 时报错，硬件设置这位为'1'，写入'1'可以清除这位状态。
6	EVERR	擦除校验错误 当页擦除后校验时报错，硬件设置这位为'1'，写入'1'可以清除这位状态。
5	EOP	操作结束 当闪存操作（编程/擦除）完成时，硬件设置这位为'1'，写入'1'可以清除这位状态。 *注：每次成功的编程或擦除都会设置 EOP 状态。*
4	WRPERR	写保护错误 试图对写保护的闪存地址编程时，硬件设置这位为'1'，写入'1'可以清除

位域	名称	描述
		这位状态。
3	PVERR	编程校验错误 当编程后校验时报错，硬件设置这位为'1'，写入'1'可以清除这位状态。
2	PGERR	编程错误 试图对内容不是'0xFFFF_FFFF'的地址编程时，硬件设置这位为'1'，写入'1'可以清除这位状态。 *注：进行编程操作之前，必须先清除 FLASH_CTRL 寄存器的 START 位。*
1	Reserved	保留，硬件强制为 0。
0	BUSY	忙 该位指示闪存操作正在进行。在闪存操作开始时，该位被设置为'1'；在操作结束或发生错误时该位被清除为'0'。

2.2.5.2.5 FLASH 控制寄存器（FLASH_CTRL）

地址偏移量：0x10

复位值：0x0000 0080



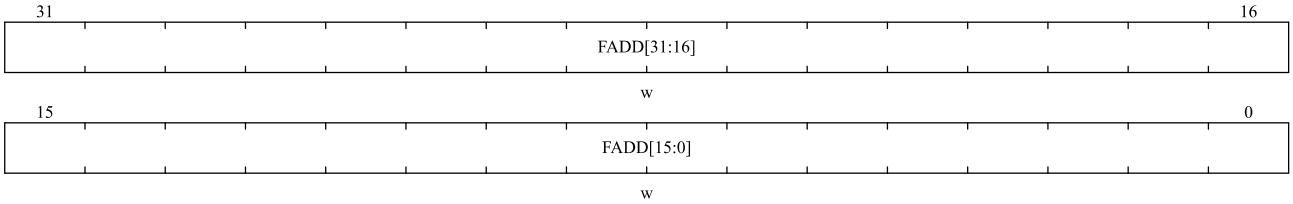
位域	名称	描述
31:14	Reserved	保留，硬件强制为 0。
13	ECERRITE	ECC 错误中断 该位允许在 FLASH_STS 寄存器中的 ECCERR 位变为'1'时产生中断。 0：禁止产生中断; 1：允许产生中断。
12	EOPITE	允许操作完成中断 该位允许在 FLASH_STS 寄存器中的 EOP 位变为'1'时产生中断。 0：禁止产生中断; 1：允许产生中断。
11	FERRITE	擦除/编程校验错误中断 该位允许在 FLASH_STS 寄存器中的 EVERR/PVERR 位变为'1'时产生中断。 0：禁止产生中断; 1：允许产生中断。
10	ERRITE	允许错误状态中断 该位允许在发生 Flash 错误时产生中断（当 FLASH_STS 寄存器中的 PGERR/WRPERR 置为'1'时）。 0：禁止产生中断; 1：允许产生中断。
9	OPTWE	允许写选项字节

位域	名称	描述
		当该位为'1'时，允许对选项字节进行编程操作。当在 FLASH_OPTKEY 寄存器写入正确的键序列后，该位被置为'1'。 软件可清除此位。
8	SMPSEL	Flash 编程模式选项 0：SMP1 模式。在进行编程之前，需要先读出编程所在地址的内容，并检查是否已被擦除过，若尚未被擦除则不执行编程操作，并在 FLASH_STS 寄存器中置起 PGERR 警告位； 1：SMP2 模式。在进行编程之前，不会判断编程所在地址的内容是否已经被擦除过，Flash 会直接启动编程。如果编程地址之前已经写入过数据，使用 SMP2 模式编程该地址时只能写入相同数据，否则无法保证数据写入正确。
7	LOCK	锁 只能写'1'。当该位为'1'时表示 Flash 和 FLASH_CTRL 被锁住。在检测到正确的解锁序列后，硬件清除此位为'0'。 在一次不成功的解锁操作后，下次系统复位前，该位不能再被改变。
6	START	开始 当该位为'1'时将触发一次擦除操作。该位只可由软件置为'1'并在 BUSY 变为'1'时清为'0'。
5	OPTER	擦除选项字节 擦除选项字节。
4	OPTPG	烧写选项字节 对选项字节编程。
3	Reserved	保留，硬件强制为 0。
2	MER	全擦除 选择擦除所有用户页。
1	PER	页擦除 选择擦除页。
0	PG	编程 选择编程操作。

2.2.5.2.6 FLASH 地址寄存器（FLASH_ADD）

地址偏移量：0x14

复位值：0x0000 0000



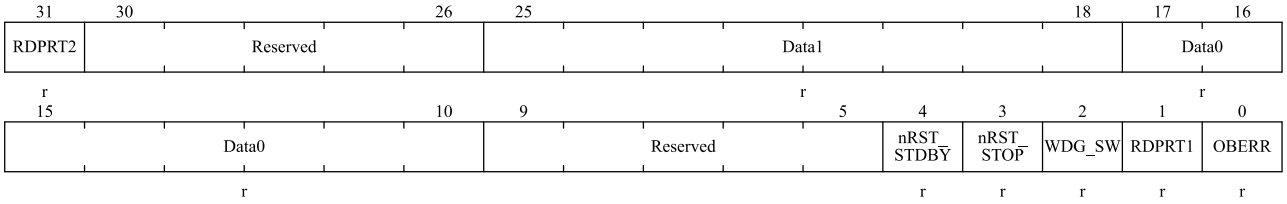
位域	名称	描述
31:0	FADD	闪存地址 当进行编程时选择要编程的地址，当进行页擦除时选择要擦除的页。

位域	名称	描述
		注意：当 FLASH_STS 中的 BUSY 位为'1'时，不能写这个寄存器。

2.2.5.2.7 选项字节寄存器（FLASH_OB）

地址偏移量：0x1C

复位值：0x03FF FFFC

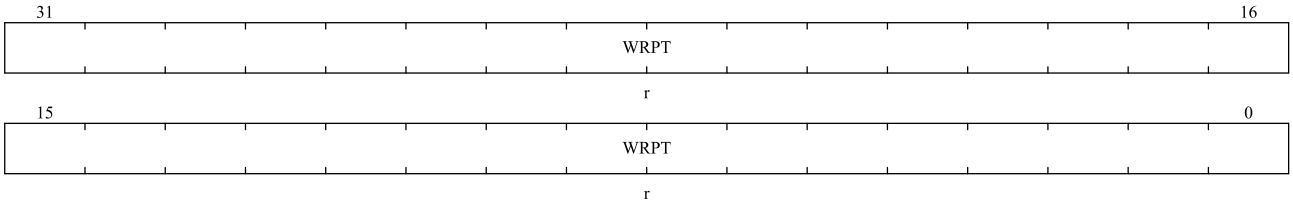


位域	名称	描述
31	RDPRT2	读保护 L2 级别 0：读保护 L2 级别未使能； 1：读保护 L2 级别使能。 *注：该位为只读。*
30:26	Reserved	保留，硬件强制为 0。
25:18	Data1[7:0]	Data1 *注：该位为只读。*
17:10	Data0[7:0]	Data0 *注：该位为只读。*
9:5	Reserved	保留，硬件强制为 0。
4	nRST_STDBY	进入 Standby 模式复位配置 0：进入 Standby 模式后立即产生复位； 1：进入 Standby 模式后不产生复位。 *注：该位为只读。*
3	nRST_STOP	进入 STOP0/STOP2 模式复位配置 0：进入 STOP0/STOP2 模式后立即产生复位； 1：进入 STOP0/STOP2 模式后不产生复位。 *注：该位为只读。*
2	WDG_SW	看门狗设置 0：硬件看门狗； 1：软件看门狗。 *注：该位为只读。*
1	RDPRT1	读保护 L1 级别 0：读保护 L1 级别未使能； 1：读保护 L1 级别使能。 *注：该位为只读。*
0	OBERR	选项字节错误 当该位为'1'时表示选项字节和它的反码不匹配。 *注意：该位为只读。*

2.2.5.2.8 写保护寄存器（FLASH_WRP）

地址偏移量：0x20

复位值：0xFFFF FFFF

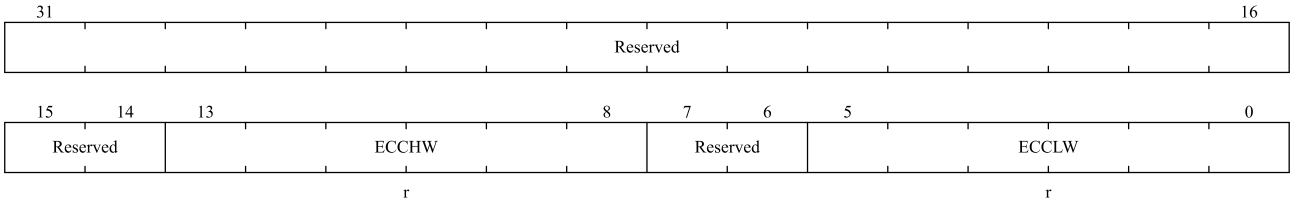


位域	名称	描述
31:0	WRPT	写保护 该寄存器包含由 OBL 加载的写保护选项字节。 0：写保护生效； 1：写保护失效。 *注意：这些位为只读。*

2.2.5.2.9 ECC 寄存器（FLASH_ECC）

地址偏移量：0x24

复位值：0x0000 0000

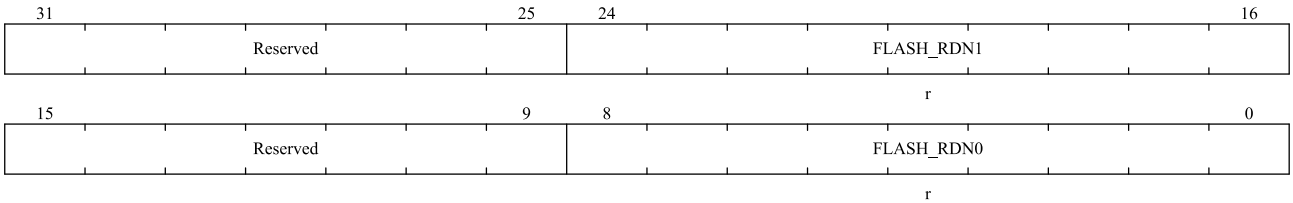


位域	名称	描述
31:16	Reserved	保留
15:14	Reserved	保留，硬件强制为 0。
13:8	ECCHW	32 位 Flash 地址对应的高 6-bit ECC 值。
7:6	Reserved	保留，硬件强制为 0。
5:0	ECCLW	32 位 Flash 地址对应的低 6-bit ECC 值。

2.2.5.2.10 RDN 寄存器（FLASH_RDN）

地址偏移量：0x2C

复位值：0x0000 0000

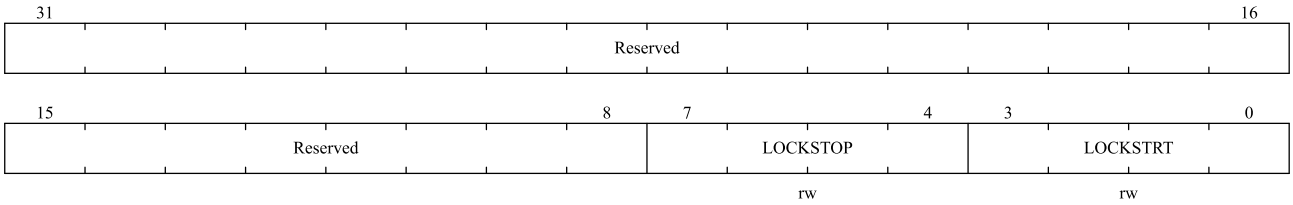


位域	名称	描述
31:25	保留	硬件强制为 0。

位域	名称	描述
24:16	FLASH_RDN1	Flash 冗余块页 1 的地址
15:9	保留	硬件强制为 0。
8:0	FLASH_RDN0	Flash 冗余块页 0 的地址

2.2.5.2.11 CAH 寄存器（FLASH_CAH）

地址偏移量：0x30
复位值：0x0000 0000



位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7:4	LOCKSTOP[3:0]	iCache 锁定停止（详细操作说明见 2.2.2.3.3iCache 锁定章节）。
3:0	LOCKSTRT[3:0]	iCache 锁定开始。

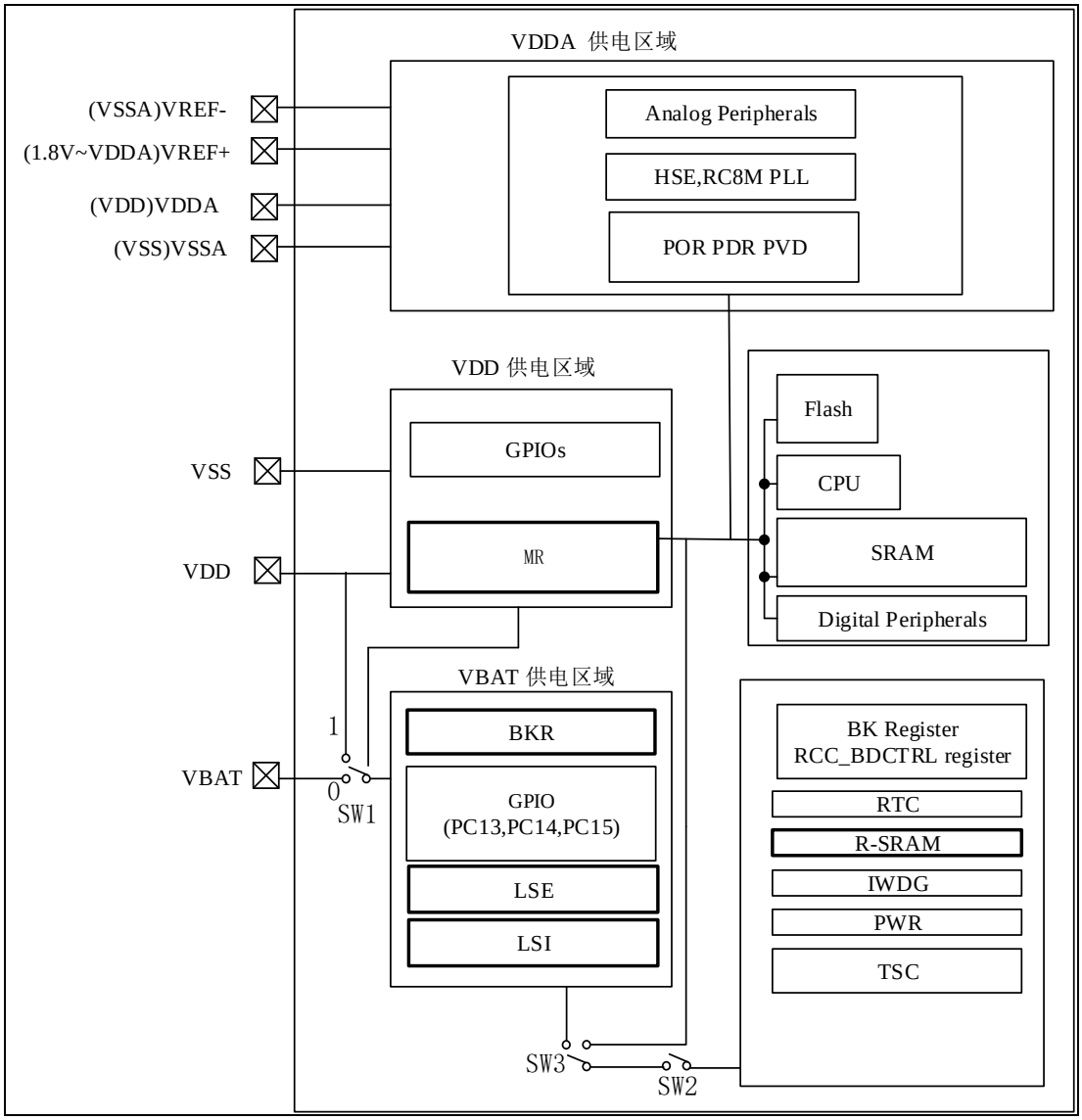
3 电源控制（PWR）

3.1 电源系统简介

CM32M4xxR 工作电压（VDD）为 1.8V~3.6V。它主要有 3 个模拟/数字电源区域（VDD、VBAT、VDDA）。具体请参考图 3-1 电源框图。

PWR 作为整个器件的电源控制模块，主要功能是控制 CM32M4xxR 进入不同的电源模式以及可以被其他事件或者中断唤醒。CM32M4xxR 支持 RUN、SLEEP、STOP0、STOP2、STANDBY 和 VBAT 模式。

图 3-1 电源框图



3.1.1 电源

为了说明不同的电源域的功能，下面将对一些电源区域进行介绍，本文档将在后面的的章节介绍电源区域的数字部分，模拟部分请参考模拟相关设计文档。

- VDD 区域：电压输入范围为 1.8V~3.6V，主要为 MR 提供电源输入，并为 CPU, AHB, APB, SRAM, FLASH 及大部分数字外设接口供电。
- VBAT 区域：输入电压范围为 1.8V~3.6V，为 BKR 和一些特殊的 IO（PC13, PC14, PC15）口供电。当 VDD 掉电时，开关把供电系统 VDD 切换到 VBAT。
- VDDA 区域：输入电压范围 1.8V~3.6V，主要为时钟及复位系统、大部分模拟外设供电。

3.1.1.1 数字模块供电系统

CM32M4xxR 的 VDD 和 VBAT 输入电压范围为 1.8V~3.6V，BKR 和 MR 是内部的电压调节器可以为数字模块供电系统提供电源。VDD 和 VBAT 一般由外部直接供电，VBAT 由电池供电来保持备份区域的内容，而 VDD 则由其他的外部供电系统供电。另外如果不需要接电池，那么 VBAT 必须直接连接到 VDD。

- **MR(RUN,SLEEP,STOP0)**

MR 是内部的主电源控制器，主要用在 RUN 模式、SLEEP 模式以及 STOP0 模式。MR 有两种模式，正常模式和低功耗模式，低功耗模式用于 STOP0 来进一步降低功耗。

当 MR 进入低功耗模式时，CPU 会进入深度睡眠状态。此时应设置 PWR_CTRL 寄存器 PDS 位为 0，LPS 位为 1。当 MR 进入正常模式，此时需要设置 PWR_CTRL 寄存器 PDS 位为 0，LPS 位也为 0。

- **BKR(STOP2,STANDBY,VBAT)**

BKR 是内部备电域电源控制器，用在 STOP2、STANDBY 和 VBAT 模式。在 STOP2 模式，CPU 状态是保持的，另外给数字备电区域、GPIO、IOM 和 EXTI 供电。当 CPU 进入深度睡眠，此时应设置 PWR_CTRL2 寄存器 STOP2S 位为 1。

数字备电区域主要模块包括 PWR、IO（PA0_WAKUP, PC13_TAMPER, PC14, PC15）、R-SRAM,、TSC、RTC、BKR 和 RCC_BDCTRL 寄存器。SW3 开关打开时，CPU 会进入深度睡眠状态。当 SW1 把供电系统切换到 VBAT 时，表明 VDD 此时已经掉电了。

3.1.2 备电区域

复位时，SW1 会把供电系统切换到 VDD 供电区域。在 STOP2、STANDBY 和 VBAT 模式，内部电压调整器 BKR 将给数字备电区域供电。

注意：

- 在 VDD 上升阶段或者 PDR 被检测到时，在 VBAT 和 VDD 之间的开关保持连接到 VBAT 区域。
- 在启动阶段，如果 VDD 快速建立，并且 VDD > VBAT + 0.6V，电流可以通过内部二极管连接注入到 VBAT。如果连接到 VBAT 的电源或者电池不支持这种电流的注入。强烈建议在该电源和 VBAT 引脚之间加一个低压的二极管。

  如果应用中没有外部电池。建议 VBAT 引脚连接到 VDD 上，同时并一个 100nF 的陶瓷电容。在

RUN、SLEEP、STOP0 模式，备电区域由 VDD 供电（SW1 连接到 VDD），下述功能可用：

- PC14 和 PC15 可以被用于普通的 IO 口或者 LSE 管脚
- PC13 可以被用于普通的 IO 口，TAMPER 管脚，RTC 校验时钟引脚，RTC 闹钟和秒输出

注意：
由于事实上 SW1 和 SW2 流过的电流是受限制的，最大为 3mA 。所以 PA0_WAKUP、PC13 到 PC15 这些 IO 输出模式是受限制的，在外挂 30PF 的电容时，最大的输出速度为 2MHz。另外这些 IO 不能当电流驱动，比如不能去驱动 LED。SW2 的电流会维持在 3mA 或者更低，因为 GPIO、IOM、EXTI 工作都会共同消耗电流。

当 VBAT 为备电区域供电时，此时可以使用以下功能：

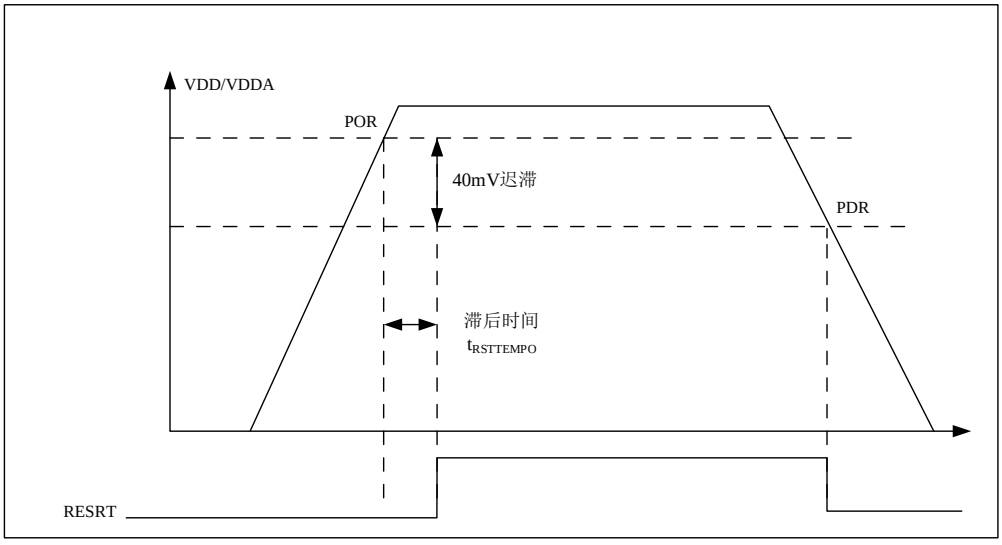
- PC14 和 PC15 只能用于 LSE 管脚
- PC13 被用于 TAMPER 管脚，RTC 闹钟或者秒输出

3.1.3 上电复位（POR）和下电复位（PDR）

CM32M4xxR 有一个集成的上电（POR）和掉电（PDR）电路模块。可以工作在最低 1.8V 的电压。不需要外部的复位电路，当 VDD 或者 VDDA 低于规定的阈值（$V_{POR/PDR}$）时，CM32M4xxR 会保持复位状态。

有关开关电源复位阀值的详细信息，请参阅相关数据手册手册电气特性部分。

图 3-2 上电复位和掉电复位的波形图



3.1.4 可编程电压监测器（PVD）

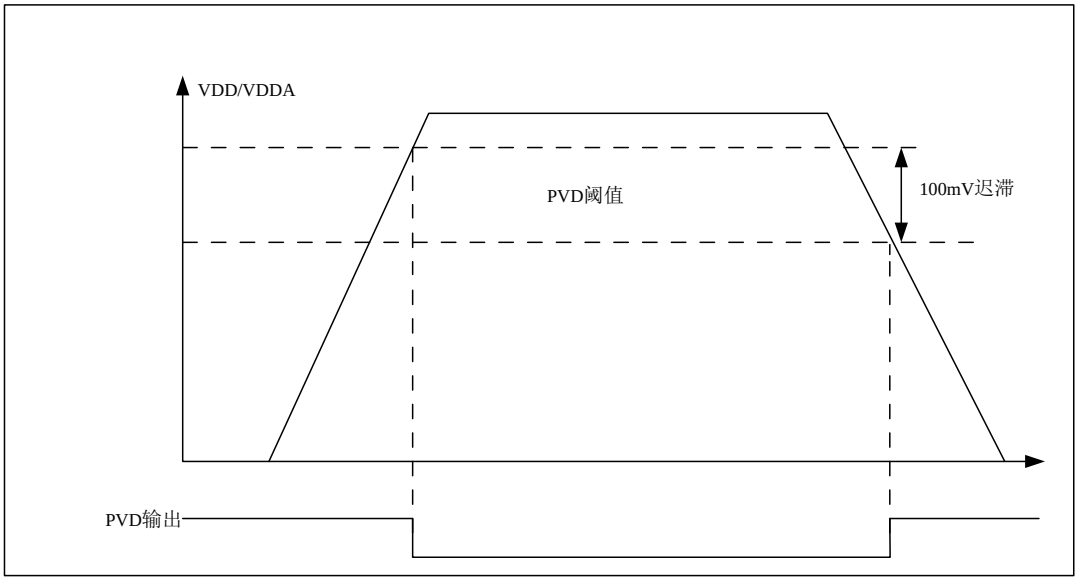
通过与电源控制寄存器（PWR_CTRL）中 PRS[2:0]位设置的阈值进行比较，PVD 可用于监控 VDD/VDDA 电源。通过设置 PVDEN 位启用 PVD 功能。

在电源控制状态寄存器（PWR_CTRLSTS）中，PVDO 标志位是可以配置的，用于指示 VDD/VDDA 是否高于或低于 PVD 阈值。此事件在内部连接到 EXTI line16，如果 EXTI 寄存器使能了，则可以生成中断。当 VDD/VDDA 低于 PVD 阈值或当 VDD/VDDA 高于 PVD 阈值时，可以根据 EXTI line16 的上升/下

降边缘配置生成 PVD 输出中断。例如，服务例程可以执行紧急关闭任务。

注意：CM32M4xxR PVD 阈值需要配合寄存器 PWR_CTRL3 中位 EXMODE 配置。具体请参考寄存器 PWR_CTRL 中位 MSB 和 PRS[2:0]，它们的组合可以组成 4 位 PVD 阈值配置。

图 3-3 PVD 阈值中断



3.1.5 Brown 复位（BOR）

BOR 是器件内置的掉电复位控制器，当 VDD 电压低于 1.62V（典型值）时，器件将保持为复位状态。

3.2 电源管理

CM32M4xxR 共有六种电源模式:RUN、SLEEP、STOP0、STOP2、STANDBY 和 VBAT，不同的模式具有不同的性能和功耗。CM32M4xxR 电源模式请参考下面表格:

表 3-1 电源模式

模式	条件	进入	退出	电流（典型）	唤醒时间
RUN	CPU 启动 外设配置	上电，系统复位，低功耗唤醒	进入睡眠、STOP0、STOP2、待机和 VBAT 模式	100uA/MHz& HCLK>2MHz	0
SLEEP	CPU 进入睡眠模式，内核停止。所有的外设配置，电压调节器仍在运行。任一中断和事件都可以唤醒 CPU	core_wfi_mode = 1, core_sleep_value = 0, WFI/WFE	唤醒: 1）如果通过 WFI 进入（wfe.WFE=0），任一 ECLIC 中断都可以退出 2）如果通过 WFI	60uA/MHz & HCLK>2MHz	6 Cycles

模式	条件	进入	退出	电流（典型）	唤醒时间
			进入（wfe.WFE=1），外部中断线退出		
STOP0[1]	CPU 深度睡眠方式: 外设时钟，所有数字模块和电压调节器仍运行。HSE / HSI / PLL 关闭。 LSE / LSI，RTC 或者其他外设可以配置用来唤醒。 所有的 SRAM 保持数据，所有的 IO 口，IWDG 和 RTC 可以用来唤醒 CPU。 唤醒之后开启 HSI，代码从挂起的地方启动。	WFI / WFE: 1）core_sleep_value = 1 PDS = 0, 2）LDS=0/1，选择主电压调节器工作模式	唤醒: 1）如果由 WFI 进入（wfe.WFE=0），任一 ECLIC 中断都可以退出 2）如果通过 WFI 进入（wfe.WFE=1），外部事件退出	120uA	20uS
STOP2[2]	CPU 深度睡眠方式: CPU 寄存器保持，所有的核心数字逻辑区域电源全部关闭。 主电压调节器（MR）关闭，HSE/HSI/PLL 关闭。LSE/LSI 可配置， GPIO 保持，外设 IO 复用不保持。16KB R-SRAM 保持，其他的 SRAM 和寄存器数据都丢失。 84B BK 寄存器保持。 GPIO，IOM 和 EXTI 开启。	WFI/WFE: 1）core_sleep_value = 1 2）PWR_CTRL2.STOP2S=1	唤醒: 1）如果由 WFI 进入（wfe.WFE=0），任一 ECLIC 中断都可以退出 2）如果通过 WFI 进入（wfe.WFE=1），外部事件退出	详细请参看各系列数据手册电气性能章节	40uS
STANDBY	主电压调节器关闭， HSE/HSI/PLL 关闭。 LSE/LSI 可配置。 16KB 字节 R-SRAM 保持，通过 PWR_CTRL2.SR2STBRET 配置。 其他的 SRAM 和寄存器数据都丢失。 除了以下 NRST/PA0_WKUP/PC13-TAMPER/PC14-OSC32_IN/PC15-OSC32_OUT，其他 IO 都是高阻态。 84 字节的 BK 寄存器仍保持数据值，IWDG，RTC/PAO/WKUP/NRST/	WFI/WFE: 1）core_sleep_value = 1 中断/事件 2）PDS=1	WKUP 上升沿，RTC alarm 上升沿，NRST 复位，IWDG 复位	详细请参看各系列数据手册电气性能章节	100uS

模式	条件	进入	退出	电流（典型）	唤醒时间
	TAMPER 可以唤醒 CPU。				
VBAT	CPU 关闭，所有的外设关闭，主电压调节器关闭，LSE/LSI 可配置，HSE/HSI/PLL 关闭。除了 NRST/PC13-TAMPER/PC14-OSC32_IN/PC15-OSC32_OUT，大部分 IO 口处于高阻态。	VDD 关闭	VDD 开启	详细请参看各系列数据手册电气性能章节	100uS

注意:
1. STOP0 模式，在唤醒后，代码可以从停止位置继续运行。
2. STOP2 模式，在唤醒后，在 R-SRAM 区的堆和全局变量可以从停止的位置恢复和继续，此时外设需要重新初始化。

3. 3 低功耗模式

默认情况下，CM32M4xxR 在系统复位或电源打开复位后处于运行模式。当 CPU 不需要运行时（例如在等待外部事件时），可以使用几种低功耗模式来节省功耗。由用户选择在低功耗、短启动时间和可用的唤醒源之间选择最佳低功耗模式。

CM32M4xxR 五种低功耗模式特征:

- SLEEP 模式(内核停止，所有外围设备包括 RISC-V Core 核心外设，如 ECLIC，系统 Timer(SysTimer)依然在运行)
- STOP0 模式 （大部分时钟源被关闭）
- STOP2 模式 （大部分数字控制中心被关闭）
- STANDBY 模式 （内核被关闭，备份区域打开，VDD 和 VBAT 保持）
- VBAT 模式 （VDD 关闭, VBAT 保持）

此外，运行模式下的功耗可以通过以下方法之一来降低:

- 降低系统时钟
- 关闭 APB 和 AHB 总线上未被使用的外设时钟

3. 3. 1 SLEEP 模式

3. 3. 1. 1 进入 SLEEP 模式

通过执行 WFI（等待中断）或 WFE（等待事件）和 core_sleep_value = 0，进入 SLEEP 模式。N 级别处理器内核可以通过 WFI 指令进入休眠状态。当处理器执行到 WFI 指令之后，将会:

- 立即停止执行当前的指令流;
- 等待处理器内核完成任何尚未完成的滞外操作（Outstanding Transactions），譬如取指令和数据读写操作，以保证发到总线上的操作都完成;
- 当所有的滞外操作（Outstanding Transactions）都完成后，处理器会安全地进入一种空闲状态，这

种空闲状态可以被称之为“休眠”状态。

在 SLEEP 模式下，所有 I/O 引脚保持与运行模式下相同的状态/功能。

注意：如果在等待总线上的操作完成的过程中发生了存储器访问错误异常，则会进入到异常处理模式，而不会休眠。

3.3.1.2 退出 SLEEP 模式

SLEEP 模式的退出按照下面的描述：

- 如果 WFI 指令（wfe.WFE = 0）用于进入 SLEEP 模式，那么嵌套的向量中断控制器（ECLIC）所响应的任何外围中断都可以将设备从 SLEEP 模式中唤醒
- 如果使用 WFI 指令（wfe.WFE = 1）进入 SLEEP 模式，则将在事件发生时立即退出 SLEEP 模式。
- 唤醒事件可以通过以下方式生成：
  - NMI，NMI 总能够唤醒处理器内核。当处理器内核检测到输入信号 nmi 的上升沿，处理器内核被唤醒，进入到 NMI 服务程序开始执行。
  - 中断，如果 CSR 寄存器 wfe.WFE 域被配置为 0，则： mstatus.MIE 域被配置为 1（表示全局中断被打开），则当 ECLIC（通过将外部请求的中断进行仲裁）向处理器内核发送了中断，处理器内核被唤醒，进入到中断服务程序开始执行；如果 mstatus.MIE 域被配置为 0（表示全局中断被关闭），则当 ECLIC（通过将外部请求的中断进行仲裁）向处理器内核发送了中断，内核被唤醒，继续顺序执行之前停止的指令流（而不是进入到中断服务程序）。
  - Event，CSR 寄存器 wfe.WFE 域被配置为 1，则等待 Event 唤醒。当处理器内核检测到输入信号 rx_evt（称之为 Event 信号）为高电平时，处理器内核被唤醒，继续执行之前停止的指令流（而不是进入到中断服务程序）。
  - Debug 请求，Debug 请求总能够唤醒处理器内核，如果调试器（Debugger）接入，也会将处理器内核唤醒而进入调试模式。

3.3.2 STOP0 模式

STOP0 模式基于 N308 CPU 核深度睡眠模式，并结合外设时钟控制机制。电压调整器可以配置为普通或低功率模式。在 STOP0 模式下，核心域中的时钟源大多数都是禁用的，如 PLL、HSI 和 HSE。但是 SRAM、R-SRAM 和所有寄存器内容都被保存。

在 STOP0 模式下，所有 I/O 引脚都保持与运行模式相同的状态。

3.3.2.1 进入 STOP0 模式

进入 STOP0 模式时，主要的区别是设置 sleepvalue. SLEEPVALUE= 1，PWR_CTRL2.STOP2S=0，PDS=0。另一个不同之处在于，MR 可以运行在正常模式或者低功耗模式，通过配置寄存器 PWR_CTRL 位 LPS。当 LPS = 1 时，MR 运行在低功耗模式。

当 LPS = 0 时，MR 运行在正常模式。

在 STOP0 模式下，所有 I/O 引脚保持与运行模式下相同的状态和功能。

如果正在进行 FLASH 操作，则进入 STOP0 模式的时间将被延迟，直到完成内存访问。

如果对 APB 区域的访问正在进行，则进入 STOP0 模式的时间将被延迟，直到 APB 访问完成。

在 STOP0 模式下，可以通过对各个控制位进行编程来选择以下特性：

- 独立看门狗（IWDG）：在它相关寄存器软件写入或者硬件操作时，独立看门狗将被启动，一旦启动将一直工作，直到产生一个复位信息
- RTC：可以通过寄存器 RCC_BDCTRL 中 RTCEN 位来开启
- 内部 RC 振荡器（LSI RC）：可以通过寄存器 RCC_CTRLSTS 中 LSIEN 位来开启
- 外部的 32.768KHz 晶振（LSE OSC）：可以通过寄存器 RCC_BDCTRL 中 LSEEN 位来开启

ADC 或 DAC 也可以在 STOP0 模式下耗电，可以在进入 STOP0 模式之前禁用 ADC 和 DAC。

注意：如果应用程序需要在进入停止模式之前禁用外部时钟，则必须首先禁用 HSEEN 位，然后将系统时钟切换到 HSI。否则，如果在进入停止模式时，HSEEN 位保持使能，并且去掉外部时钟（外部振荡器），则必须启用时钟安全系统（CSS）功能，以检测任何外部振荡器故障，并避免进入停止模式时出现故障行为。

3.3.2.2 退出 STOP0 模式

当产生中断或唤醒事件退出 STOP0 模式时，选择 HSI RC 振荡器作为系统时钟。

当电压调整器在低功率模式下工作时，从 STOP0 模式中唤醒时会产生额外的启动延迟。在 STOP0 模式下，通过内部调节器处于普通模式，这样可以减少启动时间，但相应的功耗会增加。

3.3.3 STOP2 模式

STOP2 模式基于内核的深度睡眠模式，所有的核心数字逻辑区域电源全部关闭。主电压调节器（MR）关闭，HSE/HSI/PLL 关闭。CPU 寄存器保持，LSE/LSI 可配置，GPIO 保持，外设 IO 复用不保持。16K 字节 R-SRAM 保持，其他的 SRAM 和寄存器数据都丢失。84 字节备份寄存器保持。GPIO，IOM 和 EXTI 开启。

3.3.3.1 进入 STOP2 模式

当进入 STOP2 模式。主要的区别是设置 sleepvalue. SLEEPVALUE= 1，PWR_CTRL2.STOP2S=1, PWR_CTRL 位 PDS=0, LPS=0 。

在 STOP2 模式中，如果正在对 FLSAH 进行操作时，则进入 STOP2 模式的时间将被延迟，直到完成内存访问。

如果对 APB 区域的访问正在进行，则进入 STOP2 模式的时间将被延迟，直到 APB 访问完成。

在 STOP2 模式下，可以通过对各个控制位进行编程来选择以下特性：

- 独立看门狗：在它相关寄存器软件写入或者硬件操作时，独立看门狗将被启动，一旦启动将一直工作直到产生一个复位信息
- RTC：可以通过寄存器 RCC_BDCTRL 中 RTCEN 位来开启
- 内部 RC 振荡器（LSI RC）：可以通过寄存器 RCC_CTRLSTS 中 LSIEN 位来开启
- 外部的 32.768KHz 晶振（LSE OSC）：可以通过寄存器 RCC_BDCTRL 中 LSEEN 位来开启

注：如果进入 STOP2 还想保持数据（全局变量、栈等），应当将其放到 R-SRAM 里。

3.3.3.2 退出 STOP2 模式

当通过发出中断或唤醒事件退出 STOP2 模式时，选择 HSI RC 振荡器作为系统时钟。退出 STOP2

时，代码将从停止的位置继续执行。

3.3.4 STANDBY 模式

STANDBY 模式可以实现更低的功耗，它基于 N308 CPU 核深度睡眠模式，核心域完全关闭，备电区域打开，为 VDD 和 BKR 供电。

3.3.4.1 进入 STANDBY 模式

当进入 STANDBY 模式。主要的区别是设置 sleepvalue. SLEEPVALUE= 1，PDS=1。

在 STANDBY 模式中，除 NRST、PA0_WKUP、PC13_TAMPER、PC14、PC15 外，所有 I/O 引脚都保持高阻状态。

如果正在对 FLASH 进行操作时，则进入 STANDBY 模式的时间将被延迟，直到完成内存访问。

如果对 APB 区域的访问正在进行，则进入 STANDBY 模式的时间将被延迟，直到 APB 访问完成。

在 STANDBY 模式下，可以通过对各个控制位进行编程来选择以下特性：

- 独立看门狗：在它相关寄存器软件写入或者硬件操作时，独立看门狗将被启动，一旦启动将一直工作直到产生一个复位信息
- RTC：可以通过寄存器 RCC_BDCTRL 中 RTCEN 位来开启
- 内部 RC 振荡器（LSI RC）：可以通过寄存器 RCC_CTRLSTS 中 LSIEN 位来开启
- 外部的 32.768KHz 晶振（LSE OSC）：可以通过寄存器 RCC_BDCTRL 中 LSEEN 位来开启
- R-SRAM 数据保持，可以通过寄存器 PWR_CTRL2 中 SR2STBRET 位来开启

3.3.4.2 退出 STANDBY 模式

当外部复位（NRST 引脚）、IWDG 复位、WKUP 引脚上升沿或 RTC 闹钟事件上升沿发生时，CM32M4xxR 退出 STANDBY 模式。除电源控制状态寄存器(PWR_CTRLSTS)外，所有寄存器在从 STANDBY 状态唤醒后都将复位。

从 STANDBY 模式中唤醒后，代码执行等同于复位后的执行（boot 管脚被触发、读取复位向量等）。电源控制状态寄存器（PWR_CTRLSTS）中的 SBF 状态标志表明 MCU 由待机模式退出。

3.3.5 VBAT 模式

在 VBAT 模式下 CPU 关闭，所有的外设关闭，主电压调节器关闭，LSE/LSI 可配置，HSE/HSI/PLL 关闭。除了 NRST/PC13-TAMPER/PC14-OSC32_IN/PC15-OSC32_OUT，大部分 IO 口处于高阻态。

在 VBAT 模式下，根据 VDD 掉电之前的配置，可以使用以下特性：

- RTC：可以通过寄存器 RCC_BDCTRL 中 RTCEN 位来开启
- 内部 RC 振荡器（LSI RC）：可以通过寄存器 RCC_CTRLSTS 中 LSIEN 位来开启
- 外部的 32.768KHz 晶振（LSE OSC）：可以通过寄存器 RCC_BDCTRL 中 LSEEN 位来开启
- R-SRAM 数据保持，可以通过寄存器 PWR_CTRL2 中 SR2VBRET 位来开启

3.3.5.1 进入 VBAT 模式

当 VDD 掉电时，将在任何时候进入 VBAT 模式。

注意：如使用 VBAT 模式，在进入 VBAT 模式时含 ADC 功能的 GPIO 管脚和 NRST 管脚须浮空或接地，否则在 VBAT 模式下，会由于从上述管脚漏电到芯片内部掉电区域，导致芯片 VBAT 模式异常。

3.3.5.2 退出 VBAT 模式

当 VDD 恢复到上电复位阈值时，CM32M4xxR 退出 VBAT 模式。在 VDD 恢复后，CM32M4xxR 核心区域将完整的按照上电顺序执行。从 VBAT 模式中醒来后，代码执行等同于复位后的执行。电源控制状态寄存器（PWR_CTRLSTS）中的 VBATF 状态标志表明 MCU 由 VBAT 模式退出。

3.3.6 Debug 模式

默认情况下，如果应用程序在使用调试特性时将 MCU 置于 STOP0、STOP2、STANDBY 或 VBAT 模式，则会丢失调试连接。这是由于 CPU 内核失去了时钟。

但是，通过在 DBG_CTRL 寄存器中设置一些配置位，即使在使用低功耗模式时，也可以对软件进行调试。如果配置了这些寄存器位，电压调节器和 HSI 将不会被禁用或关闭。

DBG_CTRL 中的 STDBY 和 STOP 位控制电源开关和调节器的执行。

STDBY:调试待机模式

0:（FCLK=Off, HCLK=Off）整个数字部分电源关闭。

从软件的角度来看，从待机中退出与获取复位向量是相同的（除了少数状态位表示 MCU 正在从待机中恢复）

1:（FCLK=On, HCLK=On）在这种情况下，数字部分是有电的，FCLK 和 HCLK 是由内部 RC 振荡器提供的，该振荡器保持激活状态。此外，MCU 在待机模式下生成系统复位，复位之后跟待机模式一样。

STOP:调试停止模式

0:（FCLK=Off, HCLK=Off）在停止模式下，时钟控制器禁用所有时钟（包括 HCLK 和 FCLK）。从停止模式退出时，时钟配置与复位后相同（CPU 由 8MHZ 内部 RC 振荡器（HIS）计时）。因此，软件必须重新配置时钟控制器，以启用锁相环、外部晶振等。

1:（FCLK=On, HCLK=On）在这种情况下，当进入停止模式时，FCLK 和 HCLK 由内部 RC 振荡器提供，RC 振荡器在停止模式下保持活动。退出停止模式时，软件必须重新配置时钟控制器，以启用锁相环、外部晶振等（与 STOP=0 时的方法相同）。

3.3.7 低功耗自动唤醒（AWU）模式

自动唤醒模式下, RTC 可以用于从不同的低功耗模式中唤醒 CM32M4xxR, 而不需要依赖外部中断。RTC 提供了一个可编程的时钟基准，用于从 STOP0、STOP2 和待机模式中定时唤醒。为此，可以通过在备份域控制寄存器（RCC_BDCTRL）中对 RTCSEL[1:0]位进行软件编程来选择三个可选的 RTC 时钟源中的两个:

- 32.768KHz 外部晶振时钟（LSE OSC）

  这个时钟源提供了一个精确的时钟基准，并且有非常低的功耗（小于 1μA，典型测试下）

- RC 内部晶振时钟（LSI RC）

  这个时钟源具有节省 32.768 KHz 晶振成本的优势。但是可能会增加少许功耗。

  若要使用 RTC 闹钟事件从 STOP0 和 STOP2 模式唤醒，需要:

- 配置 EXTI 17 上升沿触发
- 配置 RTC 开启 RTC 闹钟事件

  若要使用 RTC 闹钟事件从待机模式中唤醒，不需要配置 EXTI 17。

  VBAT 模式无法通过 RTC 唤醒。

3.4 PWR 寄存器

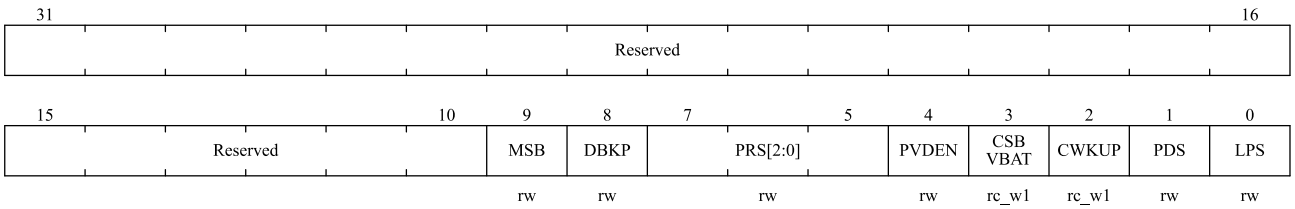
3.4.1 PWR 寄存器映射图

表 3-2 PWR 寄存器地址映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	PWR_CTRL	Reserved																						MSB	DBKP	PRS[2:0]			PVDEN	CSBVBAT	CWKUP	PDS	LPS
	Reset Value																							0	0	0	0	0	0	0	0	0	0
004h	PWR_CTRLSTS	Reserved																							WKUPEN	Reserved				VBATF	PVDO	SBF	WKUPF
	Reset Value																								0					0	0	0	0
008h	PWR_CTRL2	Reserved																				IWDGRSTEN	IWDGWPEN	LSITRIM[4:0]						TMPWPEN	SR2STBRET	SR2VBRET	STOP2S
	Reset Value																					1	1	0	1	1	1	0	0	1	0	0
00Ch	PWR_CTRL3	Reserved																															EXMODE
	Reset Value																																0

3.4.2 电源控制寄存器（PWR_CTRL）

偏移地址：0x00
复位值：0x0000 0000 （从待机模式唤醒时清除）

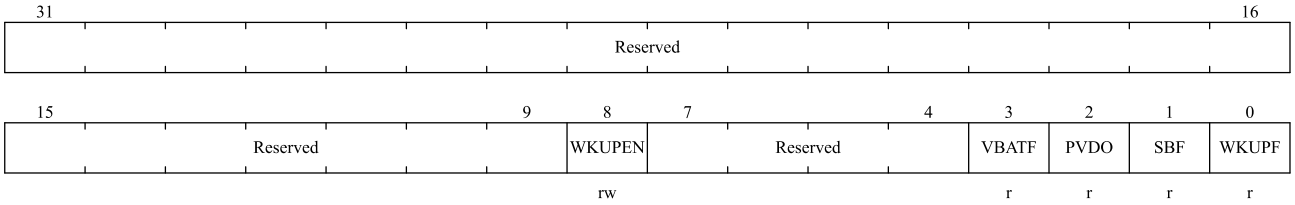


<table>
<tr><th>位域</th><th>名称</th><th>描述</th></tr>
<tr><td>31:10</td><td>Reserved</td><td>保留，硬件强制为 0。</td></tr>
<tr><td>9</td><td>MSB</td><td>4 比特位 PVD 阈值设置位。
EXMODE = 1 时，该位可操作。
因此需要先配置 PWR_CTRL3 中 EXMODE 位。
当 MSB 位为 0 时，阈值如下：
<table>
<tr><th>寄存器（4 位）</th><th>电压</th><th>模拟通道值</th></tr>
<tr><td>0000</td><td>2.2v</td><td>0100</td></tr>
<tr><td>0001</td><td>2.3v</td><td>0101</td></tr>
<tr><td>0010</td><td>2.4v</td><td>0110</td></tr>
<tr><td>0011</td><td>2.5v</td><td>0111</td></tr>
<tr><td>0100</td><td>2.6v</td><td>1000</td></tr>
<tr><td>0101</td><td>2.7v</td><td>1001</td></tr>
<tr><td>0110</td><td>2.8v</td><td>1010</td></tr>
<tr><td>0111</td><td>2.9v</td><td>1011</td></tr>
</table>
当 MSB 位为 1 时，阈值如下：
<table>
<tr><th>寄存器（4 位）</th><th>电压</th><th>模拟通道值</th></tr>
<tr><td>1000</td><td>1.88v</td><td>0000</td></tr>
<tr><td>1001</td><td>1.88v</td><td>0001</td></tr>
<tr><td>1010</td><td>1.98v</td><td>0010</td></tr>
<tr><td>1011</td><td>2.08v</td><td>0011</td></tr>
<tr><td>1100</td><td>3.06v</td><td>1100</td></tr>
<tr><td>1101</td><td>3.24v</td><td>1101</td></tr>
<tr><td>1110</td><td>3.42v</td><td>1110</td></tr>
<tr><td>1111</td><td>3.60v</td><td>1111</td></tr>
</table></td></tr>
<tr><td>8</td><td>DBKP</td><td>取消备电域的写保护。
在复位状态下，RTC 和备电域寄存器要保护起来，防止非法读出写入。必须设置此位以启用对这些寄存器的写访问。
0：禁用对 RTC 和备份寄存器的访问
1：启用对 RTC 和备份寄存器的访问
注意:如果 RTC 时钟为 HSE/128，这个位必须保持为 1。</td></tr>
<tr><td>7:5</td><td>PRS[2:0]</td><td>PVD 监测电压选择。
不同位的组合代表电压检测器不同的电压阈值。
000：2.2V　100：2.6V　001：2.3V
101：2.7V　010：2.4V　110：2.8V
011：2.5V　111：2.9V
如需 4 位配置，请结合 MSB 位进行配置。
注意：详细说明参见数据手册中的电气特性部分。</td></tr>
<tr><td>4</td><td>PVDEN</td><td>电源电压监测器（PVD）使能。
0：禁止 PVD
1：开启 PVD</td></tr>
<tr><td>3</td><td>CSBVBAT</td><td>清除 STANDBY/VBAT 位。</td></tr>
</table>

位域	名称	描述
		始终读出为 0 0：无效 1：清除 SBF 待机位（写）
2	CWKUP	清除唤醒位。 始终读出为 0 0：无效 1：2 个系统时钟周期后清除 WUF 唤醒位（写）
1	PDS	掉电深睡眠位。 与 LPS 位协同操作 0：当 CPU 进入深睡眠时进入停机模式，调压器的状态由 LPS 位控制。 1：CPU 进入深睡眠时进入待机模式。
0	LPS	深睡眠下的低功耗。 PDS=0 时，与 PDS 位协同操作 0：在停机模式下电压调压器开启 1：在停机模式下电压调压器处于低功耗模式

3.4.3 电源控制状态寄存器（PWR_CTRLSTS）

偏移地址：0x04
复位值：0x0000 0000（从待机模式唤醒时不被清零）

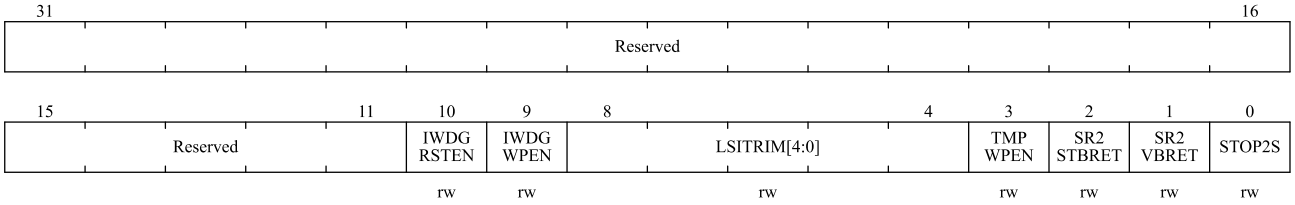


位域	名称	描述
31:9	Reserved	保留，硬件强制为 0。
8	WKUPEN	使能 WKUP 引脚位 0：WKUP 引脚为通用 I/O。WKUP 引脚上的事件不能将 CPU 从待机模式唤醒 1：WKUP 引脚用于将 CPU 从待机模式唤醒，WKUP 引脚被强置为输入下拉的配置（WKUP 引脚上的上升沿将系统从待机模式唤醒） *注意：在系统复位时清除这一位。*
7:4	Reserved	保留，硬件强制为 0。
3	VBATF	VBAT 标志位。 该位由硬件设置，仅通过 POR 或者 PDR（电源上电复位/电源掉电复位）清除或由电源控制寄存器（PWR_CTRL）中设置 CSBVBAT 位来设置。 0：设备未处于 VBAT 模式 1：设备已处于 VBAT 模式
2	PVDO	PVD 输出。 当 PVD 被 PVDEN 位使能后该位才有效

位域	名称	描述
		0：VDD/VDDA 高于由 PRS[2:0]选定的 PVD 阀值 1：VDD/VDDA 低于由 PRS[2:0]选定的 PVD 阀值 *注意：在待机模式下 PVD 被停止。因此，待机模式后或复位后，直到设置 PVDEN 位之前，该位为 0。*
1	SBF	待机标志位。 该位由硬件设置，并只能由 POR/PDR（上电/掉电复位）或设置电源控制寄存器（PWR_CTRL）的 CSBVBAT 位清除。 0：系统不在待机模式 1：系统进入待机模式
0	WKUPF	唤醒标志位。 该位由硬件设置，并只能由 POR/PDR（上电/掉电复位）或设置电源控制寄存器（PWR_CTRL）的 CWKUP 位清除。 0：没有发生唤醒事件 1：在 WKUP 引脚上发生唤醒事件或出现 RTC 闹钟事件。 *注：当 WKUP 引脚已经是高电平时，在（通过设置 WKUPEN 位）使能 WKUP 引脚时，会检测到一个额外的事件。*

3.4.4 电源控制寄存器 2（PWR_CTRL2）

偏移地址：0x08
复位值：0x0000 06E4（从待机模式唤醒时清除）

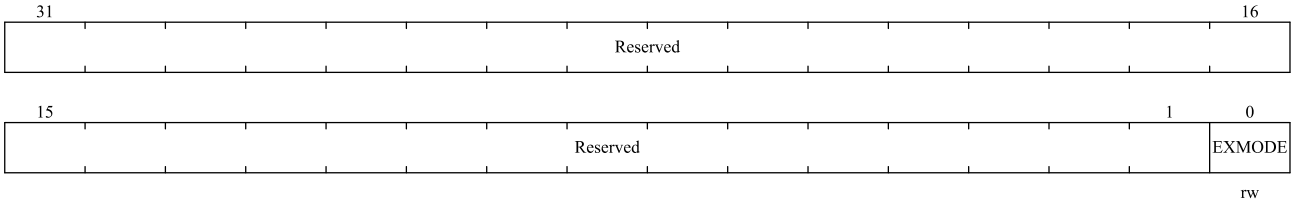


位域	名称	描述
31:11	Reserved	保留，硬件强制为 0。
10	IWDGRSTEN	独立看门狗复位使能。 0：独立看门狗不能产生复位给 RCC 1：独立看门狗能产生复位给 RCC
9	IWDGWPEN	独立看门狗唤醒使能。 0：独立看门唤醒禁止 1：独立看门唤醒使能
8:4	LSITRIM[4:0]	LSI 修正值
3	TMPWPEN	TAMPER 唤醒使能。 0：禁止 1：开启
2	SR2STBRET	R-SRAM 在待机模式下保持使能位。 0：在待机模式下，R-SRAM 保持禁止 1：在待机模式下，R-SRAM 保持开启

位域	名称	描述
1	SR2VBRET	R-SRAM 在 VBAT 模式下保持使能位。 0：在 VBAT 模式下，R-SRAM 保持禁止 1：在 VBAT 模式下，R-SRAM 保持开启
0	STOP2S	STOP2 模式使能位。 0：无使用 1：开启 STOP2 模式

3.4.5 电源控制寄存器 3（PWR_CTRL3）

偏移地址：0x0C
复位值：0x0000 5B70



位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	EXMODE	扩展模式控制位。上电后此位必须设置为 1。 0：普通模式 1：扩展模式

4 备份寄存器（BKP）

4.1 BKP 简介

备份存储器位于备份域里，电源 V_{DD} 关闭后由 V_{BAT} 供电维持。BKP 共有 42 个 16 位的寄存器，可用来存储并保护用户应用数据。这 84 个字节不受系统待机模式唤醒或系统复位的影响。

此外，BKP 控制寄存器具有侵入检测功能。

当系统发生复位，为保护备份域防止意外操作，所有的读写操作都被禁止。如需使能，首先设置寄存器 RCC_APB1PCLKEN 的 PWREN 和 BKPEN 位打开电源和后备接口时钟，然后设置电源控制寄存器（PWR_CTRL）的 DBKP 位使能对备份寄存器的读写操作，未设置电源控制寄存器（PWR_CTRL）的 DBKP 位使能前仅 BKP_CSTS 可访问。

4.2 BKP 特性

- 仅需 V_{BAT} 供电维持 84 字节数据备份寄存器
- 侵入源的有效电平可配
- 可实现侵入检测中断或事件的控制（BKP_CSTS）

4.3 BKP 功能描述

- **掉电备份**
- **侵入检测**

侵入检测事件会清除所有备份数据寄存器内容。配置备份控制寄存器 BKP_CTRL 的 TP_EN 位可使能 TAMPER 管脚的检测功能。需要注意的是，侵入检测信号是边沿检测信号与 TP_EN 位的逻辑与，因此侵入检测应该在 TAMPER 管脚使能前配置。

当检测到侵入事件时，BKP_CSTS 寄存器中的 TEF 位被置‘1’。如果侵入检测中断使能（BKP_CSTS 寄存器 TPINT_EN 位置’1’），会产生一个中断。

为避免软件在侵入检测管脚上仍然有侵入事件时对备份数据寄存器进行写操作，当清除侵入事件后，应关闭侵入检测引脚 TAMPER 的检测功能。备份数据寄存器操作结束后再对 TP_EN 位置‘1’。

备份控制寄存器 BKP_CTRL 的 TP_ALEV 位用于设定入侵检测事件的有效电平。当 TP_ALEV=0 或 1，如果 TAMPER 引脚在使能之前已经为高或低，尽管 TAMPER 引脚上没有上升或下降沿信号，仍会发生一个额外的侵入事件。因此，TAMPER 引脚应该在片外连接到正确的电平。

4.4 BKP 寄存器描述

4.4.1 BKP 寄存器映像

BKP 寄存器是 16 位的可寻址寄存器。

表 4-1 BKP 寄存器映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	Reserved																																
004h	BKP_DAT1	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	BKP_DAT2	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	BKP_DAT3	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	BKP_DAT4	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	BKP_DAT5	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	BKP_DAT6	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	BKP_DAT7	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
020h	BKP_DAT8	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
024h	BKP_DAT9	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028h	BKP_DAT10	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	Reserved																																
030h	BKP_CTRL	Reserved																														TP_ALEV	TP_EN
	Reset Value																															0	0
034h	BKP_CSTS	Reserved																						TINTF	TEF						TPINT_EN	CLRTINT	CLRTE
	Reset Value																							0	0						0	0	0
038h	Reserved																																
03Ch	Reserved																																
040h	BKP_DAT11	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
044h	BKP_DAT12	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
048h	BKP_DAT13	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04Ch	BKP_DAT14	Reserved																DAT[15:0]															

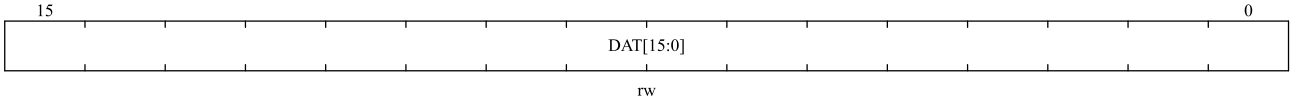
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
050h	BKP_DAT15	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
054h	BKP_DAT16	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
058h	BKP_DAT17	Reserved																DAT[15:0]															
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
05Ch	BKP_DAT18	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
060h	BKP_DAT19	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
064h	BKP_DAT20	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
068h	BKP_DAT21	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
06Ch	BKP_DAT22	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
070h	BKP_DAT23	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
074h	BKP_DAT24	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
078h	BKP_DAT25	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07Ch	BKP_DAT26	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
080h	BKP_DAT27	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
084h	BKP_DAT28	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
088h	BKP_DAT29	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
08Ch	BKP_DAT30	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
090h	BKP_DAT31	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
094h	BKP_DAT32	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
098h	BKP_DAT33	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
09Ch	BKP_DAT34	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0A0h	BKP_DAT35	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0A4h	BKP_DAT36	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0A8h	BKP_DAT37	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0ACh	BKP_DAT38	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0B0h	BKP_DAT39	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0B4h	BKP_DAT40	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0B8h	BKP_DAT41	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0BCh	BKP_DAT42	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.4.2 备份数据寄存器 x（BKP_DATx）（x = 1 … 42）

地址偏移：0x04 到 0x28，0x40 到 0xBC

复位值：0x0000

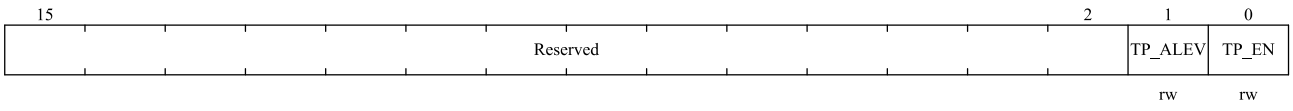


位域	名称	描述
15:0	DAT[15:0]	备份数据 这些位可以被用来写入用户数据。 *注：BKP_DATx 寄存器不会被系统复位、电源复位、从待机模式唤醒所复位。它们可以由备份域复位来复位或（如果侵入检测引脚 TAMPER 功能被开启时）由侵入引脚事件复位。*

4.4.3 备份控制寄存器（BKP_CTRL）

偏移地址：0x30

复位值：0x0000



位域	名称	描述
15:2	Reserved	保留，硬件强制为 0。
1	TP_ALEV	侵入检测 TAMPER 引脚有效电平 0：侵入检测 TAMPER 引脚上的高电平会清除所有数据备份寄存器 1：侵入检测 TAMPER 引脚上的低电平会清除所有数据备份寄存器

位域	名称	描述
0	TP_EN	启动侵入检测 TAMPER 引脚 0：侵入检测 TAMPER 引脚作为通用 IO 口使用 1：开启侵入检测引脚作为侵入检测使用

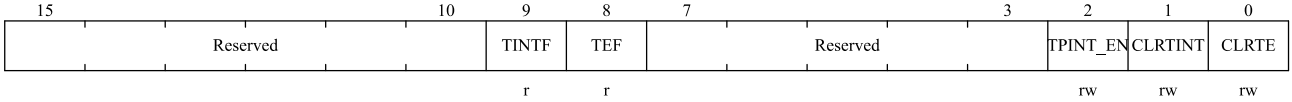
注： 同时设置 TP_ALEV 和 TP_EN 位总是安全的。然而，同时清除两者会产生一个假的侵入事件。因此，推荐只在 TP_EN 为 0 时才改变 TP_ALEV 位的状态；

TP_EN 置为 1 前，要确保要被用作 TAMPER 的 IO 先置成输入状态且通过上下拉或外部输入处于非 TAMPER 侵入电平状态，防止 TP_EN 使能时由于 IO 之前的残留状态造成侵入误报警。

4.4.4 备份控制/状态寄存器（BKP_CSTS）

偏移地址：0x34

复位值：0x0000



位域	名称	描述
15:10	Reserved	保留，硬件强制为 0。
9	TINTF	侵入中断标志 当检测到有侵入事件且 TPINT_EN 位为 1 时，此位由硬件置 1。通过向 CLRTINT 位写 1 来清除此标志位（同时也清除了中断）。如果 TPINT_EN 位被清除，则此位也会被清除。 0：无侵入中断 1：产生侵入中断 *注：仅当系统复位或由待机模式唤醒后才复位该位*
8	TEF	侵入事件标志 当检测到侵入事件时此位由硬件置 1。通过向 CLRTE 位写 1 可清除此标志位 0：无侵入事件 1：检测到侵入事件 *注：侵入事件会复位所有的 BKP_DATx 寄存器。只要 TEF 为 1，所有的 BKP_DATx 寄存器就一直保持复位状态。当此位被置 1 时，若对 BKP_DATx 进行写操作，写入的值不会被保存。*
7:3	Reserved	保留，硬件强制为 0。
2	TPINT_EN	允许侵入 TAMPER 引脚中断 0：禁止侵入检测中断 1：允许侵入检测中断（BKP_CTRL 寄存器的 TP_EN 位也必须被置 1） *注 1：侵入中断无法将系统内核从低功耗模式唤醒。* *注 2：仅当系统复位或由待机模式唤醒后才复位该位。*
1	CLRTINT	清除侵入检测中断 此位只能写入，读出值为 0。 0：无效 1：清除侵入检测中断和 TINTF 侵入检测中断标志

位域	名称	描述
0	CLRTE	清除侵入检测事件 此位只能写入，读出值为 0。 0：无效 1：清除 TEF 侵入检测事件标志（并复位侵入检测器）。

5 复位和时钟控制（RCC）

5.1 复位控制单元

CM32M4xxR 支持以下三种复位方式：

1. 电源复位
2. 系统复位
3. 备份域复位

5.1.1 电源复位

当以下事件中之一发生时，产生电源复位：

- 上电/掉电复位（POR/PDR 复位）
- 从待机模式中返回

电源复位将复位除了备份区域外的所有寄存器。（见图 3-1 电源框图）

图中复位源将最终作用于 NRST 引脚，并在复位过程中保持低电平。复位入口矢量被固定在地址 0x0000_0000。更多细节，参阅表 7-1 向量表。

5.1.2 系统复位

除了时钟控制器的 RCC_CTRLSTS 寄存器中的复位标志位和备份区域中的寄存器（见图 3-1 电源框图）以外，系统复位将复位所有寄存器至它们的复位状态。

当发生以下任一事件时，产生一个系统复位：

- NRST 引脚上的低电平（外部复位）
- 窗口看门狗计数终止（WWDG 复位）
- 独立看门狗计数终止（IWDG 复位）
- 软件复位（SW 复位）
- 低功耗管理复位
- 电源复位
- MMU 保护复位
- RAM 奇偶校验出错复位
- 备份域 EMC 复位
- 保持域 EMC 复位
- BOR 复位

可通过查看 RCC_CTRLSTS 控制状态寄存器中的复位状态标志位识别复位事件来源。

5.1.2.1 软件复位

通过将 N308 内核寄存器中的 msftrst. MSFTRST 置 1，可实现软件复位。请参考 Nuclei™ N 级别处理器内核指令架构手册获得进一步信息。

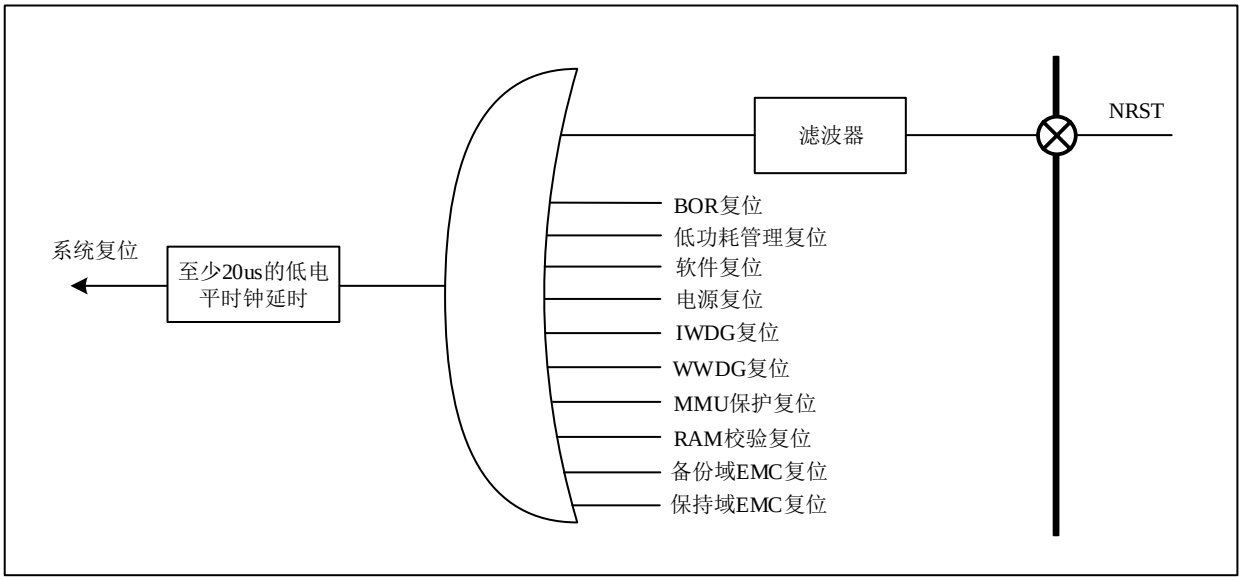
5.1.2.2 低功耗管理复位

在以下两种情况下可产生低功耗管理复位:

- 在进入待机模式时产生低功耗管理复位: 通过将用户选择字节中的 nRST_STDBY 位置 1 将使能该复位。这时，即使执行了进入待机模式的过程，系统将被复位而不是进入待机模式。
- 在进入停止模式时产生低功耗管理复位：通过将用户选择字节中的 nRST_STOP 位置 1 将使能该复位。这时，即使执行了进入停机模式的过程，系统将被复位而不是进入停机模式。

芯片内部的复位信号会在 NRST 引脚上输出，脉冲发生器保证每一个（外部或内部）复位源都能有至少 20μs 的脉冲延时；当 NRST 引脚被拉低产生外部复位时，它将产生复位脉冲。

图 5-1 复位电路



5.1.3 备份域复位

备份区域拥有两个专门的复位，它们只影响备份区域（见图 3-1 电源框图）。

当以下事件中之一发生时，产生备份区域复位。

- 软件复位，备份区域复位可由设置备份域控制寄存器（RCC_BDCTRL）中的 BDSFTRST 位产生。
- 在 V_{DD} 和 V_{BAT} 两者掉电的前提下，V_{DD} 或 V_{BAT} 上电才会引发备份区域复位。

5.2 时钟控制单元

三种不同的时钟源可被用来驱动系统时钟（SYSCLK）:

- HSI 振荡器时钟
- HSE 振荡器时钟

- PLL 时钟

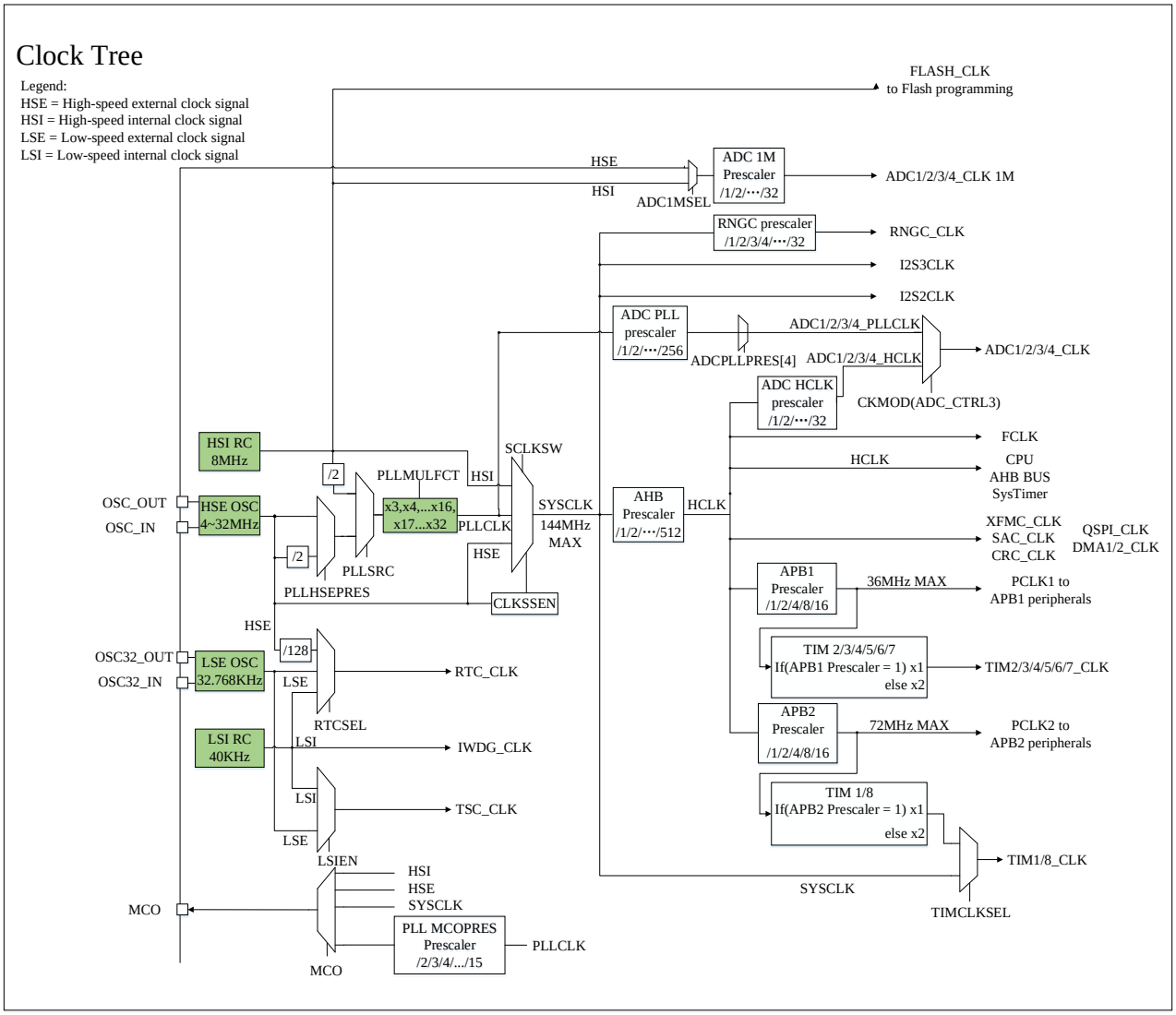
  这些设备有以下 2 种二级时钟源:

- 40KHz 低速内部 RC，可以用于驱动独立看门狗和通过程序选择驱动 RTC。RTC 用于从停止/待机模式下自动唤醒系统。

- 32.768KHz 低速外部晶体也可用来通过程序选择驱动 RTC（RTCCLK）。

  当不被使用时，任一个时钟源都可被独立地启动或关闭，由此优化系统功耗。

图 5-2 时钟树



1．当 HSI 被用于作为 PLL 时钟的输入时，系统时钟能得到的最大频率是 128MHz。

2．对于内部和外部时钟源的特性，请参考相应产品数据手册中“电气特性”章节

用户可通过多个预分频器配置 AHB、高速 APB（APB2）和低速 APB（APB1）域的频率。AHB 域的最大频率是 144MHz，APB2 域的最大频率是 72MHz，APB1 域的最大允许频率是 36MHz。

系统 Timer（SysTimer）可选 32KHz RTC 时钟或系统时钟 HCLK 作为计数时钟。通过对 SysTimer 控

制寄存器的设置，可选择 32KHz RTC 时钟或 HCLK 时钟作为 SysTick 时钟。ADC 时钟由 AHB/PLL 时钟经分频后获得。

定时器时钟频率分配由硬件按以下 2 种情况自动设置：

- 如果相应的 APB 预分频系数是 1，定时器的时钟频率与所在 APB 总线频率一致。
- 否则，定时器的时钟频率被设为与其相连的 APB 总线频率的 2 倍。

FCLK 是 N308 的自由运行时钟。

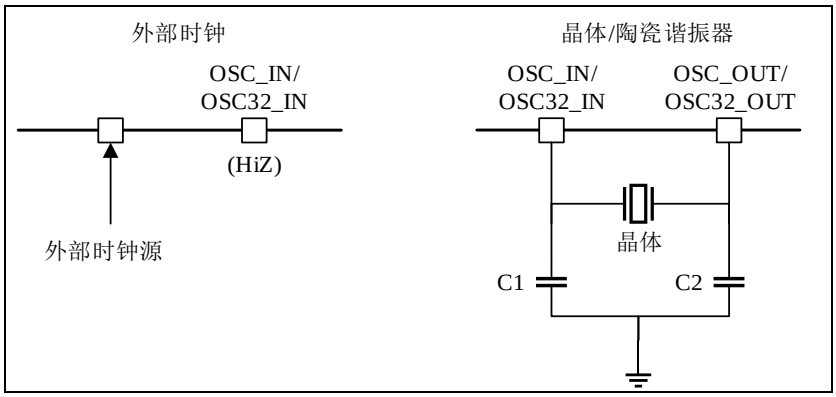
5.2.1 HSE 时钟

高速外部时钟信号（HSE）由以下两种时钟源产生：

- HSE 外部晶体/陶瓷谐振器
- HSE 用户外部时钟

为了减少时钟输出的失真和缩短启动稳定时间，晶体/陶瓷谐振器和负载电容器必须尽可能地靠近振荡器引脚。负载电容值必须根据所选择的振荡器来调整。

图 5-3 HSE/LSE 时钟源



5.2.1.1 外部时钟源（HSE 旁路）

在这个模式里，必须提供外部时钟。它的频率最高可达 32MHz。用户可通过设置在时钟控制寄存器中的 HSEBP 和 HSEEN 位来选择这一模式。外部时钟信号（50%占空比的方波、正弦波或三角波）必须连到 OSC_IN 引脚，同时保证 OSC_OUT 引脚悬空。见图 5-3。

5.2.1.2 外部晶体/陶瓷谐振器（HSE 晶体）

4~32MHz 外部振荡器可为系统提供更为精确的主时钟。相关的硬件配置可参考图 5-3，进一步信息可参考数据手册的电气特性部分。

在时钟控制寄存器 RCC_CTRL 中的 HSERDF 位用来指示高速外部振荡器是否稳定。在启动时，直到这一位被硬件置 1，时钟才被释放出来。如果在时钟中断寄存器 RCC_CLKINT 中允许产生中断，将会产生相应中断。

HSE 晶体可以通过设置时钟控制寄存器里 RCC_CTRL 中的 HSEEN 位被启动和关闭。

5.2.2 HSI 时钟

HSI 时钟信号由内部 8MHz 的 RC 振荡器产生，可直接作为系统时钟或在 2 分频后作为 PLL 输入。HSI RC 振荡器能够在不需要任何外部器件的条件下提供系统时钟。它的启动时间比 HSE 晶体振荡器短。然而，即使在校准之后它的时钟频率精度仍较差。

制造工艺决定了不同芯片的 RC 振荡器频率会不同，这就是为什么每个芯片的 HSI 时钟频率在出厂前已经被校准到 1%（25℃）的原因。系统复位时，工厂校准值被装载到时钟控制寄存器的 HSICAL[7:0]位。

如果用户的应用基于不同的电压或环境温度，这将会影响 RC 振荡器的精度。可以通过时钟控制寄存器里的 HSITRIM[4:0]位来调整 HSI 频率。

时钟控制寄存器中的 HSIRDF 位用来指示 HSI RC 振荡器是否稳定。在时钟启动过程中，直到这一位被硬件置 1，HSI RC 输出时钟才被释放。HSI RC 可由时钟控制寄存器中的 HSIEN 位来启动和关闭。

如果 HSE 晶体振荡器失效，HSI 时钟会被作为备用时钟源。参考 5.2.7 节时钟安全系统。

5.2.3 PLL

内部 PLL 可以用来倍频 HSI RC 的输出时钟或 HSE 晶体输出时钟。参考图 5-2 和时钟控制寄存器。PLL 的设置（选择 HSI 振荡器 2 分频或 HSE 振荡器不分频或 2 分频为 PLL 的输入时钟，和选择倍频因子）必须在其被激活前完成。一旦 PLL 被激活，这些参数就不能被改动。

如果 PLL 中断在时钟中断寄存器里被允许，当 PLL 准备就绪时，可产生中断申请。

5.2.4 LSE 时钟

LSE 晶体是一个 32.768KHz 的低速外部晶体或陶瓷谐振器。它为实时时钟或者其他定时功能提供一个低功耗且精确的时钟源。

LSE 晶体通过在备份域控制寄存器（RCC_BDCTRL）里的 LSEEN 位启动和关闭。

在备份域控制寄存器（RCC_BDCTRL）里的 LSERD 指示 LSE 晶体振荡是否稳定。在启动阶段，直到这个位被硬件置 1 后，LSE 时钟信号才被释放出来。如果在时钟中断寄存器里被允许，可产生中断申请。

5.2.4.1 外部时钟源（LSE 旁路）

在这个模式里必须提供一个最高 1MHz 频率的外部时钟源。用户可以通过设置在备份域控制寄存器（RCC_BDCTRL）里的 LSEBP 和 LSEEN 位来选择这个模式。具有 50%占空比的外部时钟信号（方波、正弦波或三角波）必须连到 OSC32_IN 引脚，见图 5-3。

5.2.5 LSI 时钟

LSI RC 可以在 STOP 和 STANDBY 模式下为 IWDG 和 AWU 提供时钟。LSI 时钟频率大约 40KHz。进一步信息请参考数据手册中有关电气特性部分。

LSI RC 可以通过控制/状态寄存器（RCC_CTRLSTS）里的 LSIEN 位来启动或关闭。

在控制/状态寄存器（RCC_CTRLSTS）里的 LSIRD 位指示低速内部振荡器是否稳定。在启动阶段，

直到这个位被硬件设置为 1 后，此时钟才被释放。如果在时钟中断寄存器（RCC_CLKINT）里被允许，将产生 LSI 中断申请。

5.2.5.1 LSI 校准

可以通过校准内部低速振荡器 LSI 来补偿其频率偏移，从而获得精度可接受的 RTC 时间基数，以及独立看门狗（IWDG）的超时时间（当这些外设以 LSI 为时钟源）。

校准可以通过使用 TIM5 的输入时钟（TIM5_CLK）测量 LSI 时钟频率实现。测量以 HSE 的精度为保证，软件可以通过调整 RTC 的 20 位预分频器来获得精确的 RTC 时钟基数，以及通过计算得到精确的独立看门狗（IWDG）的超时时间。

LSI 校准步骤如下：

1. 打开 TIM5，设置通道 4 为输入捕获模式；
2. 设置 AFIO_RMP_CFG 的 TIM5CH4_RMP 位为 1，在内部把 LSI 连接到 TIM5 的通道 4；
3. 通过 TIM5 的捕获/比较 4 事件或者中断来测量 LSI 时钟频率；
4. 根据测量结果和期望的 RTC 时间基数和独立看门狗的超时时间，设置 20 位预分频器。

5.2.6 系统时钟（SYSCLK）选择

系统复位后，HSI 振荡器被选为系统时钟。当时钟源被直接或通过 PLL 间接作为系统时钟时，它将不能被停止。

只有当目标时钟源准备就绪了（经过启动稳定阶段的延迟或 PLL 稳定），从一个时钟源到另一个时钟源的切换才会发生。在被选择时钟源没有就绪时，系统时钟的切换不会发生。直至目标时钟源就绪，才发生切换。

在时钟控制寄存器（RCC_CTRL）里的状态位指示哪个时钟已经准备好了，哪个时钟目前被用作系统时钟。

5.2.7 时钟安全系统（CLKSS）

时钟安全系统可以通过软件被激活。一旦其被激活，时钟监测器将在 HSE 振荡器启动延迟后被使能，并在 HSE 时钟关闭后关闭。

如果 HSE 时钟发生故障，HSE 振荡器被自动关闭，时钟失效事件将被送到高级定时器（TIM1 和 TIM8）的刹车输入端，并产生时钟安全中断 CLKSSIF，允许软件完成营救操作。此 CLKSSIF 中断连接到 N308 的 NMI 中断（不可屏蔽中断）。

一旦 CLKSS 被激活，并且 HSE 时钟出现故障，CLKSS 中断就产生，并且 NMI 也自动产生。NMI 将被不断执行，直到 CLKSS 中断挂起位被清除。因此，在 NMI 的处理程序中必须通过设置时钟中断寄存器（RCC_CLKINT）里的 CLKSSICLR 位来清除 CLKSS 中断。

如果 HSE 振荡器被直接或间接地作为系统时钟，(间接的意思是：它被作为 PLL 输入时钟，并且 PLL 时钟被作为系统时钟)，时钟故障将导致系统时钟自动切换到 HSI 振荡器，同时外部 HSE 振荡器被关闭。在时钟失效时，如果 HSE 振荡器时钟（被分频或未被分频）是用作系统时钟的 PLL 的输入时钟，PLL 也将被关闭。

5.2.8 RTC 时钟

通过设置备份域控制寄存器（RCC_BDCTRL）里的 RTCSEL[1:0]位，RTCCLK 时钟源可以由 HSE/128、LSE 或 LSI 时钟提供。除非备份域复位，此选择不能被改变。

LSE 时钟在备份域里，但 HSE 和 LSI 时钟不是。因此：

- 如果 LSE 被选为 RTC 时钟：
  - 只要 V_{BAT} 维持供电，尽管 V_{DD} 供电被切断，RTC 仍继续工作。
- 如果 LSI 被选为自动唤醒单元（AWU）时钟：
  - 如果 V_{DD} 供电被切断， AWU 状态不能被保证。有关 LSI 校准，详见 5.2.5 节 LSI 时钟。
- 如果 HSE 时钟 128 分频后作为 RTC 时钟：
  - 如果 V_{DD} 供电被切断或内部电压调压器被关闭（1.8V 域的供电被切断），则 RTC 状态不确定。
  - 必须设置电源控制寄存器 PWR_CTRL（见 3.4.2 节）的 DBKP 位（取消备电域的写保护）为 1。

5.2.9 看门狗时钟

如果独立看门狗已经由硬件选项或软件启动，LSI 振荡器将被强制在打开状态，并且不能被关闭。在 LSI 振荡器稳定后，时钟供应给 IWDG。

5.2.10 时钟输出

微控制器允许输出时钟信号到外部 MCO 引脚。

相应的 GPIO 端口寄存器必须被配置为相应功能。以下四个时钟信号可被选作 MCO 时钟：

- SYSCLK
- HSI
- HSE
- PLL 时钟分频

时钟的选择由时钟配置寄存器（RCC_CFG）中的 MCO[2:0]位控制。

5.3 RCC 寄存器

5.3.1 RCC 寄存器地址映像

表 5-1 RCC 寄存器地址映像和复位值

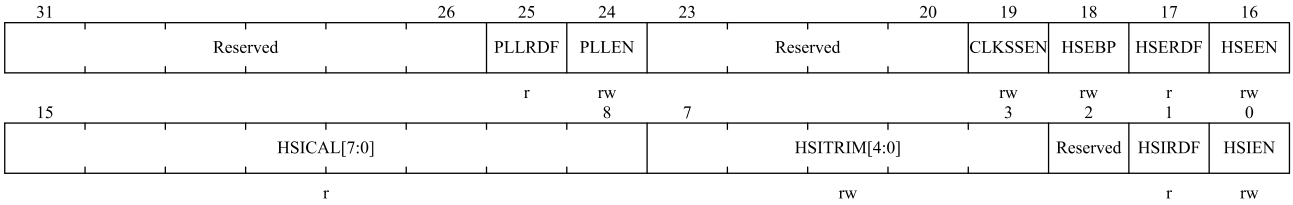
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	RCC_CTRL	Reserved						PLLRDF	PLLEN	Reserved				CLKSSEN	HSEBP	HSERDF	HSEEN	HSICAL[7:0]								HSITRIM[4:0]					Reserved	HSIRDF	HSIEN
	Reset Value							0	0					0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0		1	1

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
004h	RCC_CFG	MCOPRES[3:0]				PLLMULFCT[4]	MCO[2:0]			Reserved		PLLMULFCT[3:0]				PLLHSEPRES	PLLSRC	Reserved		APB2PRES[2:0]			APB1PRES[2:0]			AHBPRES[3:0]				SCLKSTS[1:0]		SCLKSW[1:0]	
	Reset Value	0	0	1	0	0	0	0	0			0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	RCC_CLKINT	Reserved								CLKSSICLR	Reserved		PLLRDICLR	HSERDICLR	HSIRDICLR	LSERDICLR	LSIRDICLR	Reserved			PLLRDIEN	HSERDIEN	HSIRDIEN	LSERDIEN	LSIRDIEN	CLKSSIF	Reserved		PLLRDIF	HSERDIF	HSIRDIF	LSERDIF	LSIRDIF
	Reset Value									0			0	0	0	0	0				0	0	0	0	0	0			0	0	0	0	0
00Ch	RCC_APB2PRST	Reserved											I2C4RST	I2C3RST	UART7RST	UART6RST	Reserved	Reserved	USART1RST	TIM8RST	SPI1RST	TIM1RST	Reserved			IOPGRST	IOPFRST	IOPERST	IOPDRST	IOPCRST	IOPBRST	IOPAMPRST	Reserved	AFIORST
	Reset Value												0	0	0	0			0	0	0	0				0	0	0	0	0	0	0		0
010h	RCC_APB1PRST	Reserved		DACRST	PWRRST	BKPRST	CAN2RST	CAN1RST	Reserved		I2C2RST	I2C1RST	UART5RST	UART4RST	USART3RST	USART2RST	Reserved	SPI3RST	SPI2RST	Reserved		WWDGRST	TSCRST	Reserved			TIM7RST	TIM6RST	TIM5RST	TIM4RST	TIM3RST	TIM2RST		
	Reset Value			0	0	0	0	0			0	0	0	0	0	0		0	0			0	0				0	0	0	0	0	0		
014h	RCC_AHBPCLKEN	Reserved														QSPI1EN	Reserved	ADC4EN	ADC3EN	ADC2EN	ADC1EN	SACEN	Reserved	RNGCEN	XFMCEN	Reserved	CRCEN	Reserved	FLITFEN	Reserved	SRAMEN	DMA2EN	DMA1EN	
	Reset Value															0		0	0	0	0	0		0	0		0		1		1	0	0	
018h	RCC_APB2PCLKEN	Reserved											I2C4EN	I2C3EN	UART7EN	UART6EN	Reserved	Reserved	USART1EN	TIM8EN	SPI1EN	TIM1EN	Reserved			IOPGEN	IOPFEN	IOPEEN	IOPDEN	IOPCEN	IOPBEN	IOPAMPEN	Reserved	AFIOEN
	Reset Value												0	0	0	0			0	0	0	0				0	0	0	0	0	0	0		0
01Ch	RCC_APB1PCLKEN	OPAMPEN	Reserved	DACEN	PWREN	BKPEN	CAN2EN	CAN1EN	Reserved		I2C2EN	I2C1EN	UART5EN	UART4EN	USART3EN	USART2EN	Reserved	SPI3EN	SPI2EN	Reserved		WWDGEN	TSCEN	Reserved			COMPFILTEN	COMPEN	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
	Reset Value	0		0	0	0	0	0			0	0	0	0	0	0		0	0			0	0			0	0	0	0	0	0	0	0	0
020h	RCC_BDCTRL	Reserved															BDSFTRST	RTCEN	Reserved					RTCSEL[1:0]		Reserved					LSEBP	LSERDIF	LSEEN	
	Reset Value																0	0						0	0						0	0	0	
024h	RCC_CTRLSTS	LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	MMURSTF	RMRSTF	RAMRSTF	Reserved		BKPEMCF	RETEMCF	BORRSTF	Reserved																	LSIRD	LSIEN	
	Reset Value	0	0	0	0	1	1	0	0	0			0	0	0																1	1	
028h	RCC_AHBPRST	Reserved														QSPI1RST	Reserved	ADC4RST	ADC3RST	ADC2RST	ADC1RST	SACRST	Reserved	RNGCRST	Reserved									
	Reset Value															0		0	0	0	0	0		0										
02Ch	RCC_CFG2	Reserved		TIMCLKSEL	RNGCPRES[4:0]					Reserved							ADC1MPRES[4:0]						ADC1MSEL	Reserved	ADCPLLPRES[4:0]					ADCHPRES[3:0]				
	Reset Value			0	0	0	0	0	0								0	0	1	1	1		0		0	0	0	0	0	0	0	0	0	
030h	RCC_CFG3	Reserved													TRNG1MEN	TRNG1MSEL	Reserved	TRNG1MPRES[4:0]					Reserved				BORRSTEN	Reserved						
	Reset Value														0	0		0	0	1	1	1					1							

5.3.2 时钟控制寄存器（RCC_CTRL）

偏移地址：0x00

复位值：0x0000 0083



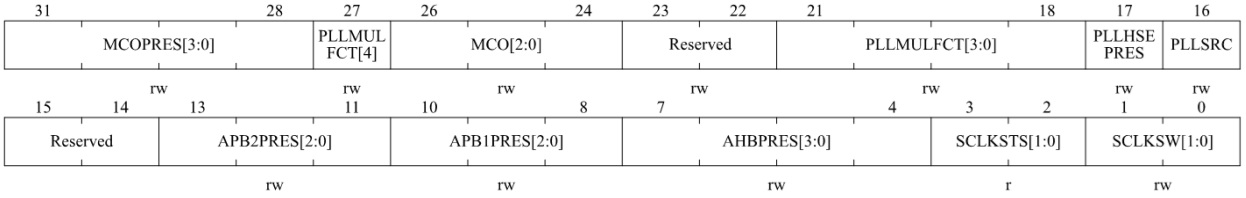
位域	名称	描述
31:26	Reserved	保留，硬件强制为 0。
25	PLLRDF	PLL 时钟准备就绪标志位。 硬件会在 PLL 就绪后置 1。 0：PLL 未就绪 1：PLL 已就绪
24	PLLEN	PLL 使能位。 软件置 1 或清零。当进入停止或待机模式时，由硬件清零；PLL 作为系统时钟时该位不能被清零。 0：关闭 PLL 1：打开 PLL
23:20	Reserved	保留，硬件强制为 0。
19	CLKSSEN	时钟安全系统使能位。 软件置 1 或清零。 0：关闭时钟检测器 1：如果外部高速时钟振荡器就绪，开启时钟检测器
18	HSEBP	外部高速时钟旁路使能位。 软件置 1 或清零。只有在外部高速时钟（HSE）振荡器关闭时才能写入该位。 0：关闭 HSE 振荡器旁路功能 1：开启 HSE 振荡器旁路功能
17	HSERDF	外部高速时钟就绪位。 硬件会在 HSE 就绪后置 1。该位在 HSEEN 位清零后需要 100ns 来清零。 0：HSE 未就绪 1：HSE 已就绪
16	HSEEN	外部高速时钟使能。 软件置 1 或清零。当进入停止或待机模式时，由硬件清零；HSE 作为系统时钟时该位不能被清零。 0：关闭 HSE 振荡器 1：开启 HSE 振荡器
15:8	HSICAL[7:0]	内部高速时钟校准值。 系统在上电时自动初始化这些位的值。
7:3	HSITRIM[4:0]	内部高速时钟修正值。 由软件写入。这些位的值将被加到 HSICAL[7:0]上，形成最终的校准值，用以校准内部 HSI RC 振荡器的频率。每步 HSICAL 的变化调整约为 40KHz，默认数值为 16，可以把 HSI 调整到 8MHz±1%。

位域	名称	描述
2	Reserved	保留，硬件强制为 0。
1	HSIRDF	内部高速时钟就绪标志位。 硬件会在 HSI 稳定后置 1。该位在 HSIEN 位清零后，需要 6 个内部 8M 振荡器时钟周期来清零。
0	HSIEN	内部高速时钟使能位。 软件置 1 或清零。HSI 作为系统时钟时该位不能被清零；当从停止或待机模式返回或 HSE 发生故障时，硬件会置 1 来启动 HSI 振荡器。 0：关闭 HSI 振荡器 1：开启 HSI 振荡器

5.3.3 时钟配置寄存器（RCC_CFG）

偏移地址：0x04

复位值：0x2000 0000



位域	名称	描述
31:28	MCOPRES[3:0]	MCO 预分频。 软件设置或清零。 0010：由 PLL 时钟 2 分频作为 MCO 时钟 0011：由 PLL 时钟 3 分频作为 MCO 时钟 0100：由 PLL 时钟 4 分频作为 MCO 时钟 0101：由 PLL 时钟 5 分频作为 MCO 时钟 0110：由 PLL 时钟 6 分频作为 MCO 时钟 0111：由 PLL 时钟 7 分频作为 MCO 时钟 1000：由 PLL 时钟 8 分频作为 MCO 时钟 1001：由 PLL 时钟 9 分频作为 MCO 时钟 1010：由 PLL 时钟 10 分频作为 MCO 时钟 1011：由 PLL 时钟 11 分频作为 MCO 时钟 1100：由 PLL 时钟 12 分频作为 MCO 时钟 1101：由 PLL 时钟 13 分频作为 MCO 时钟 1110：由 PLL 时钟 14 分频作为 MCO 时钟 1111：由 PLL 时钟 15 分频作为 MCO 时钟 其它值：不允许设置
27	PLLMULFCT[4]	此位与 21:18 位组合成 5 位的 PLL 倍频系数，请参考 PLLMULFCT[3:0]的说明。
26:24	MCO[2:0]	微控制器时钟输出选择。 由软件设置或清零。 0xx：无时钟输出

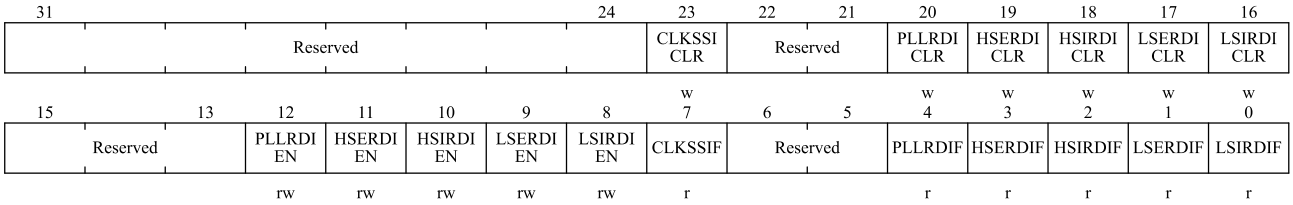
位域	名称	描述
		100：选择系统时钟（SYSCLK）输出 101：选择内部 RC 高速振荡器时钟（HSI）输出 110：选择外部高速振荡器时钟（HSE）输出 111：选择 PLL 分频后的输出（MCOPRES 设置分频系数） *注意:* *该时钟输出在启动和切换 MCO 时钟源时可能会被截断。* *在系统时钟作为输出至 MCO 引脚时，应保证输出时钟频率不超过 50MHz（I/O 口最高频率）。*
23:22	Reserved	保留
21:18	PLLMULFCT[3:0]	PLL 倍频系数（包括位 27）。 软件设置 PLL 的倍频系统。只有在 PLL 关闭时才能写入这些位，且 PLL 的输出时钟不能超过 144MHz。 00000：PLL 的输入 2 倍频到输出 00001：PLL 的输入 3 倍频到输出 00010：PLL 的输入 4 倍频到输出 00011：PLL 的输入 5 倍频到输出 00100：PLL 的输入 6 倍频到输出 00101：PLL 的输入 7 倍频到输出 00110：PLL 的输入 8 倍频到输出 00111：PLL 的输入 9 倍频到输出 01000：PLL 的输入 10 倍频到输出 01001：PLL 的输入 11 倍频到输出 01010：PLL 的输入 12 倍频到输出 01011：PLL 的输入 13 倍频到输出 01100：PLL 的输入 14 倍频到输出 01101：PLL 的输入 15 倍频到输出 01110：PLL 的输入 16 倍频到输出 01111：PLL 的输入 16 倍频到输出 10000：PLL 的输入 17 倍频到输出 10001：PLL 的输入 18 倍频到输出 10010：PLL 的输入 19 倍频到输出 10011：PLL 的输入 20 倍频到输出 10100：PLL 的输入 21 倍频到输出 10101：PLL 的输入 22 倍频到输出 10110：PLL 的输入 23 倍频到输出 10111：PLL 的输入 24 倍频到输出 11000：PLL 的输入 25 倍频到输出 11001：PLL 的输入 26 倍频到输出 11010：PLL 的输入 27 倍频到输出 11011：PLL 的输入 28 倍频到输出 11100：PLL 的输入 29 倍频到输出 11101：PLL 的输入 30 倍频到输出

位域	名称	描述
		11110：PLL 的输入 31 倍频到输出 11111：PLL 的输入 32 倍频到输出
17	PLLHSEPRES	PLL 输入的 HSE 分频器。 软件置 1 或清零。仅在 PLL 关闭时才能写入此位。 0：HSE 不分频作为 PLL 输入 1：HSE 2 分频作为 PLL 输入
16	PLLSRC	PLL 时钟源。 软件置 1 或清零，仅在 PLL 关闭时才能写入此位。 0：2 分频后的 HSI 时钟作为 PLL 输入时钟 1：HSE 时钟作为 PLL 输入时钟
15:14	Reserved	保留，硬件强制为 0。
13:11	APB2PRES[2:0]	高速 APB（APB2）预分频。 软件设置或清零，配置 APB2 时钟 PCLK2 的预分频系数。必须保证 APB2 的时钟频率不超过 72MHz。 0xx：HCLK 不分频 100：HCLK 2 分频 101：HCLK 4 分频 110：HCLK 8 分频 111：HCLK 16 分频
10:8	APB1PRES[2:0]	低速 APB（APB1）预分频。 软件设置或清零，配置 APB1 时钟 PCLK1 的预分频系数。必须保证 APB1 的时钟频率不超过 36MHz。 0xx：HCLK 不分频 100：HCLK 2 分频 101：HCLK 4 分频 110：HCLK 8 分频 111：HCLK 16 分频
7:4	AHBPRES[3:0]	AHB 预分频。 软件设置或清零，配置 AHB 时钟 HCLK 的预分频系数。 0xxx：SYSCLK 不分频 1000：SYSCLK 2 分频 1001：SYSCLK 4 分频 1010：SYSCLK 8 分频 1011：SYSCLK 16 分频 1100：SYSCLK 64 分频 1101：SYSCLK 128 分频 1110：SYSCLK 256 分频 1111：SYSCLK 512 分频
3:2	SCLKSTS	系统时钟切换状态。 硬件设置或清零，指示系统时钟来自哪一个时钟源。 00：系统时钟来自 HSI 01：系统时钟来自 HSE

位域	名称	描述
		10：系统时钟来自 PLL 输出 11：不可配置
1:0	SCLKSW	系统时钟设置 软件设置或清零，选择系统时钟源。 当从停止或待机模块返回，或 HSE 出问题且 CLKSSEN 已使能，硬件会强制将 HSI 设置为系统时钟。 00：选择 HSI 作为系统时钟 01：选择 HSE 作为系统时钟 10：选择 PLL 输出作为系统时钟 11：不可用

5.3.4 时钟中断寄存器（RCC_CLKINT）

偏移地址：0x08
复位值：0x0000 0000



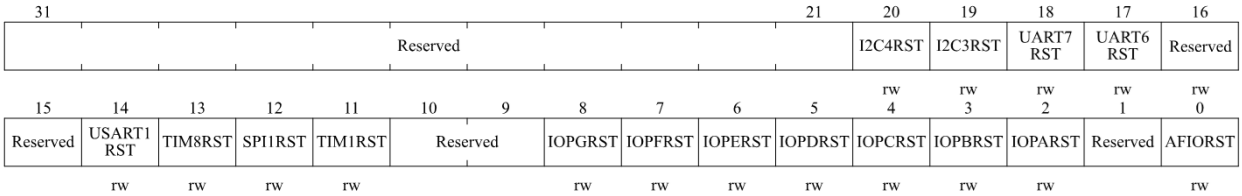
位域	名称	描述
31:24	Reserved	保留，硬件强制为 0。
23	CLKSSICLR	时钟安全系统中断清除位。 软件置 1 来清除 CLKSSIF 标志位。 0：无作用 1：清除 CLKSSIF 中断标志位
22:21	Reserved	保留，硬件强制为 0。
20	PLLRDICLR	PLL 就绪中断清除位。 软件置 1 来清除 PLLRDIF 标志位。 0：无作用 1：清除 PLLRDIF 中断标志位
19	HSERDICLR	HSE 就绪中断清除位。 软件置 1 来清除 HSERDIF 标志位。 0：无使用 1：清除 HSERDIF 中断标志位
18	HSIRDICLR	HSI 就绪中断清除位。 软件置 1 来清除 HSIRDIF 标志位。 0：无使用 1：清除 HSIRDIF 中断标志位
17	LSERDICLR	LSE 就绪中断清除位。 软件置 1 来清除 LSERDIF 标志位。

位域	名称	描述
		0：无使用 1：清除 LSERDIF 中断标志位
16	LSIRDICLR	LSI 就绪中断清除位。 软件置 1 来清除 LSIRDIF 标志位。 0：无使用 1：清除 LSIRDIF 中断标志位
15:13	Reserved	保留，硬件强制为 0。
12	PLLRDIEN	PLL 就绪中断使能位。 软件置 1 或清零，以使能或关闭 PLL 就绪中断。 0：关闭 PLL 就绪中断 1：使能 PLL 就绪中断
11	HSERDIEN	HSE 就绪中断使能位。 软件置 1 或清零，以使能或关闭 HSE 就绪中断。 0：关闭 HSE 就绪中断 1：使能 HSE 就绪中断
10	HSIRDIEN	HSI 就绪中断使能位。 软件置 1 或清零，以使能或关闭 HSI 就绪中断。 0：关闭 HSI 就绪中断 1：使能 HSI 就绪中断
9	LSERDIEN	LSE 就绪中断使能位。 软件置 1 或清零，以使能或关闭 LSE 就绪中断。 0：关闭 LSE 就绪中断 1：使能 LSE 就绪中断
8	LSIRDIEN	LSI 就绪中断使能位。 软件置 1 或清零，以使能或关闭 LSI 就绪中断。 0：关闭 LSI 就绪中断 1：使能 LSI 就绪中断
7	CLKSSIF	时钟安全系统中断标志位。 在外部 HSE 振荡器出现问题时，硬件将置 1。 0：没有 HSE 时钟错误引起的时钟安全系统中断 1：HSE 时钟错误引起了时钟安全系统中断
6:5	Reserved	保留，硬件强制为 0。
4	PLLRDIF	PLL 就绪中断标志位。 当 PLLRDIEN 被置 1 且 PLL 时钟就绪时，硬件会将此位置 1。 此位由软件对 PLLRDICLR 位置 1 来清零。 0：无 PLL 上锁产生的时钟就绪中断 1：PLL 上锁产生了时钟就绪中断
3	HSERDIF	HSE 就绪中断标志位。 当 HSERDIEN 被置 1 且外部高速时钟就绪时，硬件会将此位置 1。 此位由软件对 HSERDICLR 位置 1 来清零。 0：无 HSE 振荡器产生的时钟就绪中断 1：HSE 振荡器产生了时钟就绪中断

位域	名称	描述
2	HSIRDIF	HSI 就绪中断标志位。 当 HSIRDIEN 被置 1 且内部高速时钟就绪时，硬件会将此位置 1。 此位由软件对 HSIRDICLR 位置 1 来清零。 0：无 HSI 振荡器产生的时钟就绪中断 1：HSI 振荡器产生了时钟就绪中断
1	LSERDIF	LSE 就绪中断标志位。 当 LSERDIEN 被置 1 且外部低速时钟就绪时，硬件会将此位置 1。 此位由软件对 LSERDICLR 位置 1 来清零。 0：无 LSE 振荡器产生的时钟就绪中断 1：LSE 振荡器产生了时钟就绪中断
0	LSIRDIF	LSI 就绪中断标志位。 当 LSIRDIEN 被置 1 且内部低速时钟就绪时，硬件会将此位置 1。 此位由软件对 LSIRDICLR 位置 1 来清零。 0：无 LSI 振荡器产生的时钟就绪中断 1：LSI 振荡器产生了时钟就绪中断

5.3.5 APB2 外设复位寄存器（RCC_APB2PRST）

偏移地址：0x0c
复位值：0x0000 0000



位域	名称	描述
31:21	Reserved	保留，硬件强制为 0。
20	I2C4RST	I2C4 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 I2C4
19	I2C3RST	I2C3 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 I2C3
18	UART7RST	UART7 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 UART7
17	UART6RST	UART6 复位。 软件置 1 或清零。 0：配置后无作用

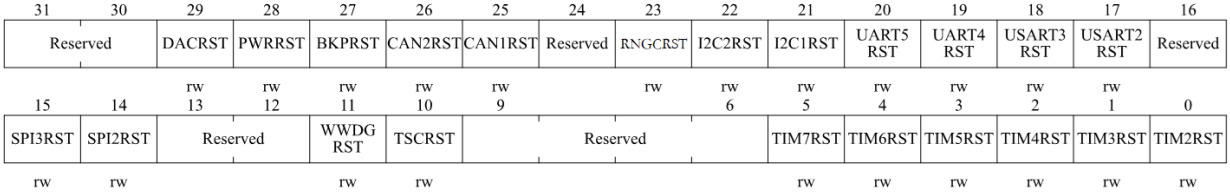
位域	名称	描述
		1：复位 UART6
16	Reserved	保留
15	Reserved	保留，硬件强制为 0。
14	USART1RST	USART1 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 USART1
13	TIM8RST	TIM8 定时器复位。 软件置 1 或清零。 0：配置后无作用 1：复位 TIM8
12	SPI1RST	SPI1 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 SPI1
11	TIM1RST	TIM1 定时器复位。 软件置 1 或清零。 0：配置后无作用 1：复位 TIM1
10:9	Reserved	保留，硬件强制为 0。
8	IOPGRST	GPIO 端口 G 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 GPIO 端口 G
7	IOPFRST	GPIO 端口 F 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 GPIO 端口 F
6	IOPERST	GPIO 端口 E 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 GPIO 端口 E
5	IOPDRST	GPIO 端口 D 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 GPIO 端口 D
4	IOPCRST	GPIO 端口 C 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 GPIO 端口 C
3	IOPBRST	GPIO 端口 B 复位。 软件置 1 或清零。

位域	名称	描述
		0：配置后无作用 1：复位 GPIO 端口 B
2	IOPARST	GPIO 端口 A 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 GPIO 端口 A
1	Reserved	保留，硬件强制为 0。
0	AFIORST	辅助功能 IO 复位。 软件置 1 或清零。 0：配置后无作用 1：复位辅助功能 IO

5.3.6 APB1 外设复位寄存器（RCC_APB1PRST）

偏移地址：0x10

复位值：0x0000 0000



位域	名称	描述
31:30	Reserved	保留，硬件强制为 0。
29	DACRST	DAC 接口复位。 软件置 1 或清零。 0：配置后无作用 1：复位 DAC 接口
28	PWRRST	电源接口复位。 软件置 1 或清零。 0：配置后无作用 1：复位电源接口
27	BKPRST	备份接口复位。 软件置 1 或清零。 0：配置后无作用 1：备份接口
26	CAN2RST	CAN2 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 CAN2
25	CAN1RST	CAN1 复位。 软件置 1 或清零。 0：配置后无作用

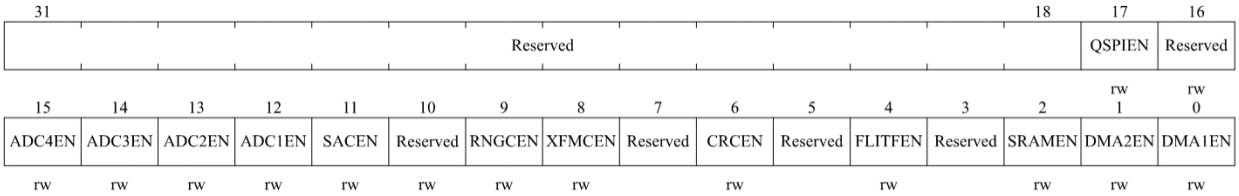
位域	名称	描述
		1：复位 CAN1
24	Reserved	保留，硬件强制为 0。
23	RNGCRST	RNGC 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 RNGC
22	I2C2RST	I2C2 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 I2C2
21	I2C1RST	I2C1 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 I2C1
20	UART5RST	UART5 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 UART5
19	UART4RST	UART4 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 UART4
18	USART3RST	USART3 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 USART3
17	USART2RST	USART2 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 USART2
16	Reserved	保留，硬件强制为 0。
15	SPI3RST	SPI3 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 SPI3
14	SPI2RST	SPI2 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 SPI2
13:12	Reserved	保留，硬件强制为 0。
11	WWDGRST	窗口看门狗复位。 软件置 1 或清零。

位域	名称	描述
		0：配置后无作用 1：复位窗口看门狗
10	TSCRST	触屏控制器复位。 软件置 1 或清零。 0：配置后无作用 1：复位触屏控制器
9:6	Reserved	保留，硬件强制为 0。
5	TIM7RST	TIM7 定时器复位。 软件置 1 或清零。 0：配置后无作用 1：复位 TIM7 定时器
4	TIM6RST	TIM6 定时器复位。 软件置 1 或清零。 0：配置后无作用 1：复位 TIM6 定时器
3	TIM5RST	TIM5 定时器复位。 软件置 1 或清零。 0：配置后无作用 1：复位 TIM5 定时器
2	TIM4RST	TIM4 定时器复位。 软件置 1 或清零。 0：配置后无作用 1：复位 TIM4 定时器
1	TIM3RST	TIM3 定时器复位。 软件置 1 或清零。 0：配置后无作用 1：复位 TIM3 定时器
0	TIM2RST	TIM2 定时器复位。 软件置 1 或清零。 0：配置后无作用 1：复位 TIM2 定时器

5.3.7 AHB 外设时钟使能寄存器（RCC_AHBPCLKEN）

偏移地址：0x14

复位值：0x0000 0014



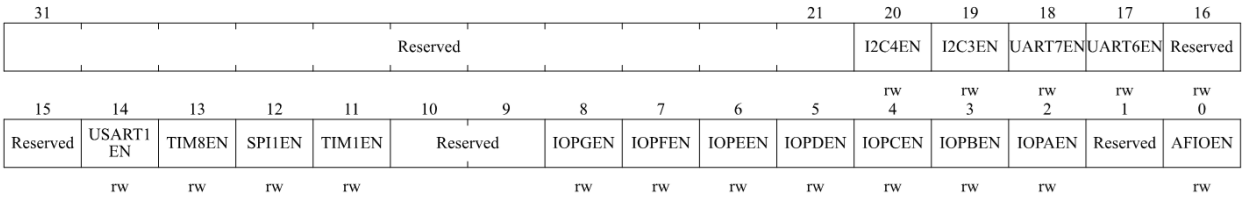
位域	名称	描述
31:18	Reserved	保留，硬件强制为 0。
17	QSPIEN	QSPI 时钟使能。 软件置 1 或清零。 0：关闭 QSPI 时钟 1：使能 QSPI 时钟
16	Reserved	保留
15	ADC4EN	ADC4 时钟使能。 软件置 1 或清零。 0：关闭 ADC4 时钟 1：使能 ADC4 时钟
14	ADC3EN	ADC3 时钟使能。 软件置 1 或清零。 0：关闭 ADC3 时钟 1：使能 ADC3 时钟
13	ADC2EN	ADC2 时钟使能。 软件置 1 或清零。 0：关闭 ADC2 时钟 1：使能 ADC2 时钟
12	ADC1EN	ADC1 时钟使能。 软件置 1 或清零。 0：关闭 ADC1 时钟 1：使能 ADC1 时钟
11	SACEN	SAC 时钟使能。 软件置 1 或清零。 0：关闭 SAC 时钟 1：使能 SAC 时钟
10	Reserved	保留
9	RNGCEN	RNGC 时钟使能。 软件置 1 或清零。 0：关闭 RNGC 时钟 1：使能 RNGC 时钟
8	XFMCEN	XFMC 时钟使能。 软件置 1 或清零。 0：关闭 XFMC 时钟 1：使能 XFMC 时钟
7	Reserved	保留，硬件强制为 0。
6	CRCEN	CRC 时钟使能。 软件置 1 或清零。 0：关闭 CRC 时钟 1：使能 CRC 时钟
5	Reserved	保留，硬件强制为 0。
4	FLITFEN	闪存接口电路时钟使能。

位域	名称	描述
		软件置 1 或清零。 0：关闭闪存接口电路时钟 1：使能闪存接口电路时钟
3	Reserved	保留，硬件强制为 0。
2	SRAMEN	SRAM 时钟使能。 软件置 1 或清零。 0：关闭睡眠模式时 SRAM 时钟 1：使能睡眠模式时 SRAM 时钟
1	DMA2EN	DMA2 时钟使能。 软件置 1 或清零。 0：关闭 DMA2 时钟 1：使能 DMA2 时钟
0	DMA1EN	DMA1 时钟使能。 软件置 1 或清零。 0：关闭 DMA1 时钟 1：使能 DMA1 时钟

5.3.8 APB2 外设时钟使能寄存器（RCC_APB2PCLKEN）

偏移地址：0x18

复位值：0x0000 0000



位域	名称	描述
31:21	Reserved	保留，硬件强制为 0。
20	I2C4EN	I2C4 时钟使能。 软件置 1 或清零。 0：关闭 I2C4 时钟 1：使能 I2C4 时钟
19	I2C3EN	I2C3 时钟使能。 软件置 1 或清零。 0：关闭 I2C3 时钟 1：使能 I2C3 时钟
18	UART7EN	UART7 时钟使能。 软件置 1 或清零。 0：关闭 UART7 时钟 1：使能 UART7 时钟
17	UART6EN	UART6 时钟使能。 软件置 1 或清零。

位域	名称	描述
		0：关闭 UART6 时钟 1：使能 UART6 时钟
16	Reserved	保留
15	Reserved	保留，硬件强制为 0。
14	USART1EN	USART1 时钟使能。 软件置 1 或清零。 0：关闭 USART1 时钟 1：使能 USART1 时钟
13	TIM8EN	TIM8 定时器时钟使能。 软件置 1 或清零。 0：关闭 TIM8 定时器时钟 1：使能 TIM8 定时器时钟
12	SPI1EN	SPI1 时钟使能。 软件置 1 或清零。 0：关闭 SPI1 时钟 1：使能 SPI1 时钟
11	TIM1EN	TIM1 定时器时钟使能。 软件置 1 或清零。 0：关闭 TIM1 定时器时钟 1：使能 TIM1 定时器时钟
10:9	Reserved	保留，硬件强制为 0。
8	IOPGEN	GPIO 端口 G 时钟使能。 软件置 1 或清零。 0：关闭 GPIO 端口 G 的时钟 1：使能 GPIO 端口 G 的时钟
7	IOPFEN	GPIO 端口 F 时钟使能。 软件置 1 或清零。 0：关闭 GPIO 端口 F 的时钟 1：使能 GPIO 端口 F 的时钟
6	IOPEEN	GPIO 端口 E 时钟使能。 软件置 1 或清零。 0：关闭 GPIO 端口 E 的时钟 1：使能 GPIO 端口 E 的时钟
5	IOPDEN	GPIO 端口 D 时钟使能。 软件置 1 或清零。 0：关闭 GPIO 端口 D 的时钟 1：使能 GPIO 端口 D 的时钟
4	IOPCEN	GPIO 端口 C 时钟使能。 软件置 1 或清零。 0：关闭 GPIO 端口 C 的时钟 1：使能 GPIO 端口 C 的时钟
3	IOPBEN	GPIO 端口 B 时钟使能。

位域	名称	描述
		软件置 1 或清零。 0：关闭 GPIO 端口 B 的时钟 1：使能 GPIO 端口 B 的时钟
2	IOPAEN	GPIO 端口 A 时钟使能。 软件置 1 或清零。 0：关闭 GPIO 端口 A 的时钟 1：使能 GPIO 端口 A 的时钟
1	Reserved	保留，硬件强制为 0。
0	AFIOEN	辅助功能 IO 时钟使能。 软件置 1 或清零。 0：关闭辅助功能 IO 时钟 1：使能辅助功能 IO 时钟

5.3.9 APB1 外设时钟使能寄存器（RCC_APB1PCLKEN）

偏移地址：0x1c

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPAMP EN	Reserved	DACEN	PWREN	BKPEN	CAN2EN	CAN1EN	Reserved	RNGCPEN	I2C2EN	I2C1EN	UART5EN	UART4EN	USART3 EN	USART2 EN	Reserved
rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3EN	SPI2EN	Reserved		WWDG EN	TSCEN	Reserved		COMP FILTEN	COMPEN	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
rw	rw			rw	rw			rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31	OPAMPEN	OPAMP 时钟使能。 软件置 1 或清零。 0：关闭 OPAMP 时钟 1：使能 OPAMP 时钟
30	Reserved	保留，硬件强制为 0。
29	DACEN	DAC 接口时钟使能。 软件置 1 或清零。 0：关闭 DAC 接口时钟 1：使能 DAC 接口时钟
28	PWREN	电源接口时钟使能。 软件置 1 或清零。 0：关闭电源接口时钟 1：使能电源接口时钟
27	BKPEN	备份接口时钟使能。 软件置 1 或清零。 0：关闭备份接口时钟 1：使能备份接口时钟
26	CAN2EN	CAN2 时钟使能。 软件置 1 或清零。

位域	名称	描述
		0：关闭 CAN2 时钟 1：使能 CAN2 时钟
25	CAN1EN	CAN1 时钟使能。 软件置 1 或清零。 0：关闭 CAN1 时钟 1：使能 CAN1 时钟
24	Reserved	保留，硬件强制为 0。
23	RNGCPEN	RNGC PCLK 时钟使能。 软件置 1 或清零。 0：关闭 RNGC PCLK 时钟 1：使能 RNGC PCLK 时钟
22	I2C2EN	I2C2 时钟使能。 软件置 1 或清零。 0：关闭 I2C2 时钟 1：使能 I2C2 时钟
21	I2C1EN	I2C1 时钟使能。 软件置 1 或清零。 0：关闭 I2C1 时钟 1：使能 I2C1 时钟
20	UART5EN	UART5 时钟使能。 软件置 1 或清零。 0：关闭 UART5 时钟 1：使能 UART5 时钟
19	UART4EN	UART4 时钟使能。 软件置 1 或清零。 0：关闭 UART4 时钟 1：使能 UART4 时钟
18	USART3EN	USART3 时钟使能。 软件置 1 或清零。 0：关闭 USART3 时钟 1：使能 USART3 时钟
17	USART2EN	USART2 时钟使能。 软件置 1 或清零。 0：关闭 USART2 时钟 1：使能 USART2 时钟
16	Reserved	保留，硬件强制为 0。
15	SPI3EN	SPI3 时钟使能。 软件置 1 或清零。 0：关闭 SPI3 时钟 1：使能 SPI3 时钟
14	SPI2EN	SPI2 时钟使能。 软件置 1 或清零。

位域	名称	描述
		0：关闭 SPI2 时钟 1：使能 SPI2 时钟
13:12	Reserved	保留，硬件强制为 0。
11	WWDGEN	窗口看门狗时钟使能。 软件置 1 或清零。 0：关闭窗口看门狗时钟 1：使能窗口看门狗时钟
10	TSCEN	触屏控制器时钟使能。 软件置 1 或清零。 0：关闭触屏控制器时钟 1：使能触屏控制器时钟
9:8	Reserved	保留，硬件强制为 0。
7	COMPFILTEN	比较器过滤器时钟使能。 0：关闭比较器过滤器时钟 1：使能比较器过滤器时钟
6	COMPEN	比较器时钟使能。 0：关闭比较器时钟 1：使能比较器时钟
5	TIM7EN	TIM7 定时器时钟使能。 软件置 1 或清零。 0：关闭 TIM7 定时器时钟 1：使能 TIM7 定时器时钟
4	TIM6EN	TIM6 定时器时钟使能。 软件置 1 或清零。 0：关闭 TIM6 定时器时钟 1：使能 TIM6 定时器时钟
3	TIM5EN	TIM5 定时器时钟使能。 软件置 1 或清零。 0：关闭 TIM5 定时器时钟 1：使能 TIM5 定时器时钟
2	TIM4EN	TIM4 定时器时钟使能。 软件置 1 或清零。 0：关闭 TIM4 定时器时钟 1：使能 TIM4 定时器时钟
1	TIM3EN	TIM3 定时器时钟使能。 软件置 1 或清零。 0：关闭 TIM3 定时器时钟 1：使能 TIM3 定时器时钟
0	TIM2EN	TIM2 定时器时钟使能。 软件置 1 或清零。 0：关闭 TIM2 定时器时钟 1：使能 TIM2 定时器时钟

5.3.10 备份域控制寄存器（RCC_BDCTRL）

偏移地址：0x20
复位值：0x0000 0000

位域	名称	描述
31:17	Reserved	保留，硬件强制为 0。
16	BDSFTRST	备份域软件复位。 软件置 1 或清零。 0：配置后无作用 1：复位整个备份域
15	RTCEN	RTC 时钟使能。 软件置 1 或清零。 0：关闭 RTC 时钟 1：使能 RTC 时钟
14:10	Reserved	保留，硬件强制为 0。
9:8	RTCSEL[1:0]	RTC 时钟源选择。 软件设置这些位以选择 RTC 时钟源。一旦 RTC 时钟源被选定，直到下次备份域被复位前不能改变。可通过设置 BDSFTRST 位来清零。 00：无时钟 01：选择 LSE 振荡器作为 RTC 时钟 10：选择 LSI 振荡器作为 RTC 时钟 11：选择 HSE 振荡器 128 分频后作为 RTC 时钟
7:3	Reserved	保留，硬件强制为 0。
2	LSEBP	外部低速时钟振荡器旁路。 在调试模式下由软件置 1 或清零来旁路 LSE。当外部 32KHz 振荡器关闭时才能写入。 0：LSE 时钟未被旁路 1：LSE 时钟被旁路
1	LSERD	外部低速时钟振荡器就绪。 硬件置 1 或清零，以指示 LSE 振荡器是否就绪。在 LSEEN 被清零后，该位需要 6 个外部低速振荡器的周期才能被清零。 0：外部 32KHz 振荡器未就绪 1：外部 32KHz 振荡器已就绪
0	LSEEN	外部低速时钟振荡器使能位。 软件置 1 或清零。 0：关闭外部 32KHz 振荡器

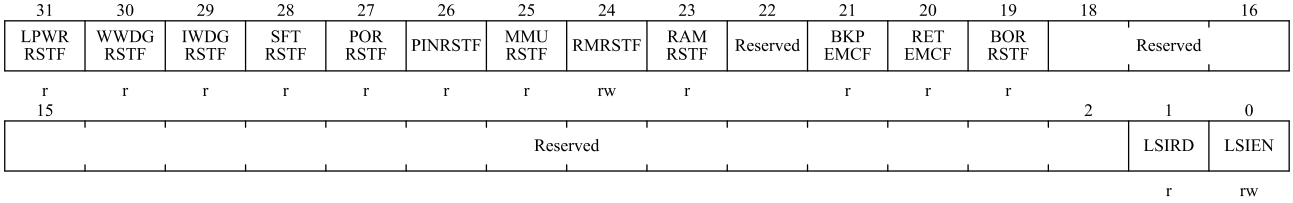
位域	名称	描述
		1：开启外部 32KHz 振荡器

注意： 备份域控制寄存器中（RCC_BDCTRL）的 LSEEN、LSEBP、RTCSEL 和 RTCEN 位处于备份域。因此，这些位在复位后处于写保护状态，只有在电源控制寄存器（PWR_CTRL）中的 DBKP 位置 1 后才能对这些位进行改动。这些位只能由备份域复位清除。任何内部或外部复位都不会影响这些位。

5.3.11 控制/状态寄存器（RCC_CTRLSTS）

偏移地址：0x24

复位值：0x0c000003

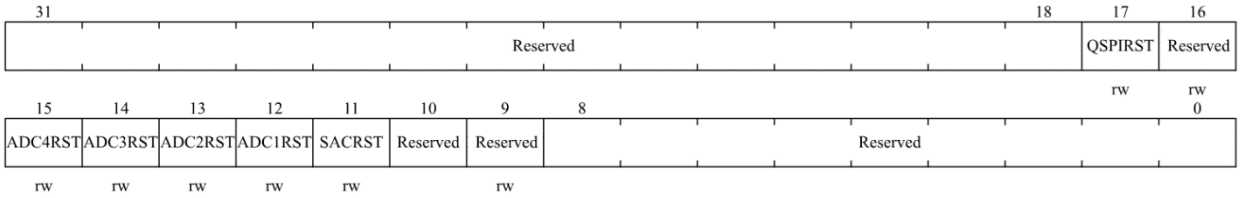


位域	名称	描述
31	LPWRRSTF	低功耗复位标志。 在低功耗管理复位发生时由硬件置 1，软件通过写 RMRSTF 位清零。 0：无低功耗管理复位发生 1：有低功耗管理复位发生
30	WWDGRSTF	窗口看门狗复位标志。 在窗口看门狗复位发生时由硬件置 1，软件通过写 RMRSTF 位清除。 0：无窗口看门狗复位发生 1：有窗口看门狗复位发生
29	IWDGRSTF	独立看门狗复位标志。 在独立看门狗复位发生时由硬件置 1，软件通过写 RMRSTF 位清除。 0：无独立看门狗复位发生 1：有独立看门狗复位发生
28	SFTRSTF	软件复位标志。 在软件复位发生时由硬件置 1，软件通过写 RMRSTF 位清除。 0：无软件复位发生 1：有软件复位发生
27	PORRSTF	上电/掉电复位标志。 在上电/掉电复位发生时由硬件置 1，软件通过写 RMRSTF 位清除。 0：无上电/掉电复位发生 1：有上电/掉电复位发生
26	PINRSTF	外部引脚复位标志位。 在 NRST 引脚复位发生时由硬件置 1，软件通过写 RMRSTF 位清除。 0：无 NRST 引脚复位发生 1：有 NRST 引脚复位发生
25	MMURSTF	MMU 复位标志。 在 MMU 复位发生时由硬件置 1，软件通过写 RMRSTF 位清除。

位域	名称	描述
		0：无 MMU 复位发生 1：有 MMU 复位发生
24	RMRSTF	清除复位标志。 软件置 1 来清除复位标志。 0：配置后无作用 1：清除复位标志
23	RAMRSTF	RAM 复位标志。 在 RAM 复位发生时由硬件置 1，软件通过写 RMRSTF 位清除。 0：无 RAM 复位发生 1：有 RAM 复位发生
22	Reserved	保留，硬件强制为 0。
21	BKPEMCF	备份域 EMC 复位标志。 在备份域 EMC 复位发生时由硬件置 1，软件通过写 RMRSTF 位清除。 0：无备份域 EMC 复位发生 1：有备份域 EMC 复位发生
20	RETEMCF	保持域 EMC 复位标志。 在保持域 EMC 复位发生时由硬件置 1，软件通过写 RMRSTF 位清除。 0：无保持域 EMC 复位发生 1：有保持域 EMC 复位发生
19	BORRSTF	BOR 复位标志。 在 BOR 复位发生时由硬件置 1，软件通过写 RMRSTF 位清除。 0：无 BOR 复位发生 1：有 BOR 复位发生
18:2	Reserved	保留，硬件强制为 0。
1	LSIRD	内部低速振荡器就绪。 硬件置 1 或清零，指示内部 40KHz RC 振荡器是否就绪。在 LSIEN 被清零后，需要 3 个内部 40KHz RC 振荡器时钟周期 LSIRD 被清零。 0：内部 40KHz RC 振荡器时钟未就绪 1：内部 40KHz RC 振荡器时钟就绪
0	LSIEN	内部低速振荡器使能。 软件置 1 或清零。 0：关闭内部 40KHz RC 振荡器 1：使能内部 40KHz RC 振荡器

5.3.12 AHB 外设复位寄存器（RCC_AHBPRST）

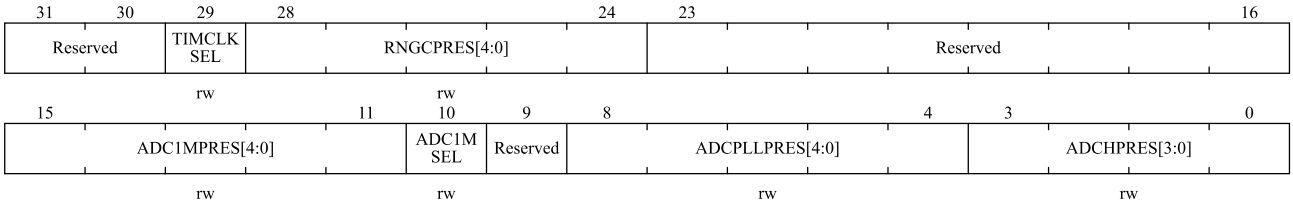
偏移地址：0x28
复位值：0x0000 0000



位域	名称	描述
31:18	Reserved	保留，硬件强制为 0。
17	QSPIRST	QSPI 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 QSPI
16	Reserved	保留
15	ADC4RST	ADC4 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 ADC4
14	ADC3RST	ADC3 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 ADC3
13	ADC2RST	ADC2 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 ADC2
12	ADC1RST	ADC1 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 ADC1
11	SACRST	SAC 复位。 软件置 1 或清零。 0：配置后无作用 1：复位 SAC
10	Reserved	保留，硬件强制为 0。
9	Reserved	保留
8:0	Reserved	保留，硬件强制为 0。

5.3.13 时钟配置寄存器 2（RCC_CFG2）

偏移地址：0x2c
复位值：0x0000 3800

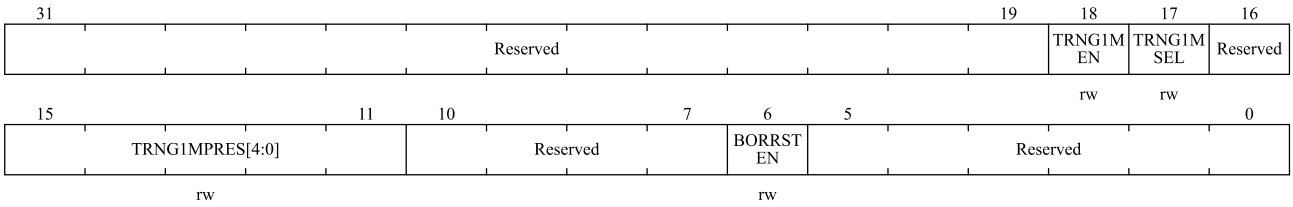


位域	名称	描述
31:30	Reserved	保留，硬件强制为 0。
29	TIMCLKSEL	TIM1/8 时钟源选择。 软件置 1 或清零。 0：tim1/8_clk 输入时钟被选择为 TIM1/8 时钟源 1：SYSCLK 输入时钟被选择为 TIM1/8 时钟源
28:24	RNGCPRES[4:0]	RNGC 预分频。 软件设置或清除这些位来配置 RNGC 时钟的预分频系数。 00000：SYSCLK 不分频 00001：SYSCLK 2 分频 00010：SYSCLK 3 分频 … 11110：SYSCLK 31 分频 11111：SYSCLK 32 分频
23:16	Reserved	保留，硬件强制为 0。
15:11	ADC1MPRES[4:0]	ADC 1M 时钟预分频。 软件设置或清除这些位来配置 ADC 1M 时钟源的预分频系数。 00000：ADC 1M 时钟源不分频 00001：ADC 1M 时钟源 2 分频 00010：ADC 1M 时钟源 3 分频 … 11110：ADC 1M 时钟源 31 分频 11111：ADC 1M 时钟源 32 分频
10	ADC1MSEL	ADC 1M 时钟源选择。 软件置 1 或清零。 0：选择 HSI 振荡器时钟作为 ADC 1M 的输入时钟 1：选择 HSE 振荡器时钟作为 ADC 1M 的输入时钟
9	Reserved	保留，硬件强制为 0。
8:4	ADCPLLPRES[4:0]	ADC PLL 预分频。 软件设置或清除这些位以配置 PLL 时钟到 ADC 的分频系数。 0xxxx：ADC PLL 时钟被关闭 10000：PLL 时钟不分频 10001：PLL 时钟 2 分频 10010：PLL 时钟 4 分频 10011：PLL 时钟 6 分频 10100：PLL 时钟 8 分频 10101：PLL 时钟 10 分频

位域	名称	描述
		10110：PLL 时钟 12 分频 10111：PLL 时钟 16 分频 11000：PLL 时钟 32 分频 11001：PLL 时钟 64 分频 11010：PLL 时钟 128 分频 11011：PLL 时钟 256 分频 其它值：PLL 时钟 256 分频
3:0	ADCHPRES[3:0]	ADC HCLK 预分频。 软件设置或清除这些位以配置 HCLK 时钟到 ADC 的分频系数。 0000：HCLK 时钟 1 分频 0001：HCLK 时钟 2 分频 0010：HCLK 时钟 4 分频 0011：HCLK 时钟 6 分频 0100：HCLK 时钟 8 分频 0101：HCLK 时钟 10 分频 0110：HCLK 时钟 12 分频 0111：HCLK 时钟 16 分频 1000：HCLK 时钟 32 分频 其它值：HCLK 时钟 32 分频

5.3.14 时钟配置寄存器 3（RCC_CFG3）

偏移地址：0x30
复位值：0x0000 3840



位域	名称	描述
31:19	Reserved	保留，硬件强制为 0。
18	TRNG1MEN	TRNG 模拟接口时钟使能。 软件置 1 或清零。 0：关闭 TRNG 模拟接口 1：使能 TRNG 模拟接口时钟
17	TRNG1MSEL	TRNG 1M 时钟选择。 软件置 1 或清零。 0：选择 HSI 振荡器作为 TRNG 1M 输入时钟 1：选择 HSE 振荡器作为 TRNG 1M 输入时钟
16	Reserved	保留，硬件强制为 0。
15:11	TRNG1MPRES[4:0]	TRNG 1M 时钟预分频。

位域	名称	描述
		软件设置或清除这些位以生成 TRNG 1M 时钟。 0000x：TRNG 1M 时钟源 2 分频 0001x：TRNG 1M 时钟源 4 分频 0010x：TRNG 1M 时钟源 6 分频 0011x：TRNG 1M 时钟源 8 分频 0100x：TRNG 1M 时钟源 10 分频 … 1111x：TRNG 1M 时钟源 32 分频
10:7	Reserved	保留，硬件强制为 0。
6	BORRSTEN	BOR 复位使能。 软件置 1 或清零。 0：不使能 BOR 复位 1：使能 BOR 复位
5:0	Reserved	保留，硬件强制为 0。

6 GPIO 和 AFIO

6.1 概述

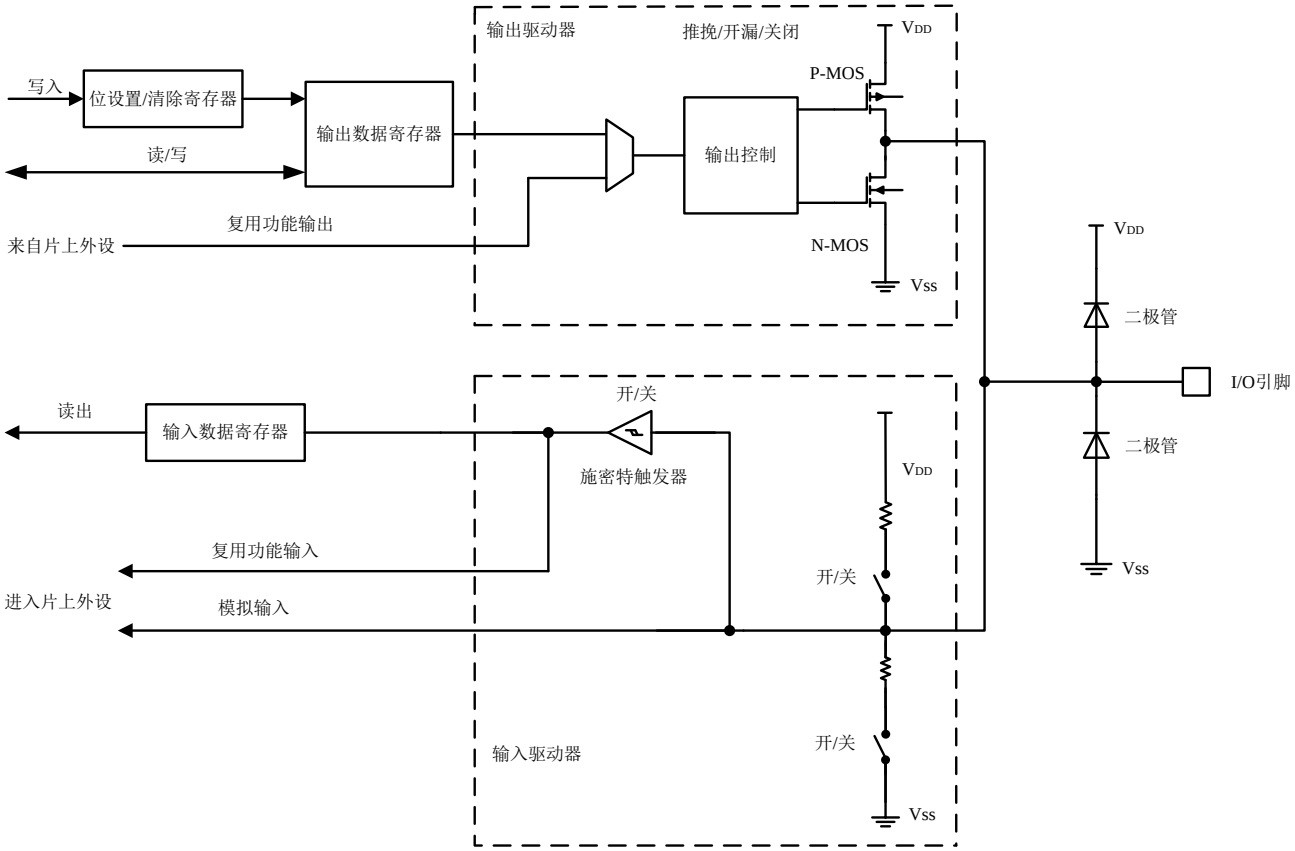
GPIO（General purpose input/output）即通用型 I/O，AFIO（Alternate-function input/output）即复用功能 I/O。芯片最多支持 97 个 GPIO，共被分为 7 组（GPIOA/GPIOB/GPIOC/GPIOD/GPIOE/GPIOF/GPIOG），每组 16 个端口（F 组共 10 个，G 组共 7 个）。GPIO 端口和其他的复用外设共用引脚，用户可以根据需求灵活配置。每个 GPIO 引脚都可以独立配置成输出、输入或复用的外设功能端口。除了模拟功能引脚外，其他的 GPIO 引脚都有大电流通过能力。

GPIO 端口可由软件分别配置成以下模式：

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟功能
- 开漏输出
- 推挽式输出
- 推挽式复用功能
- 开漏复用功能

每个 I/O 端口位可以任意编程，但必须按照 32 位字访问 I/O 端口寄存器（不允许 16 位半字或 8 位字节访问）。下图给出了一个 I/O 端口的基本结构。

图 6-1 I/O 端口的基本结构



6.2 IO 功能描述

6.2.1 IO 模式配置

IO 的模式控制由配置寄存器 GPIOx_PL_CFG，GPIOx_PH_CFG（x=A,B,C,D,E,F,G）以及输出寄存器 GPIOx_POD 来设置，不同的操作模式下的配置如下表所示:

表 6-1 IO 模式和配置关系

配置模式		PCFG1	PCFG0	PMODE1 PMODE0	PODx寄存器
通用输出	推挽（Push-Pull）	0	0	01：最大10MHz 10：最大2MHz 11：最大50MHz	0 或 1
	开漏（Open-Drain）		1		0 或 1
复用功能输出	推挽（Push-Pull）	1	0		不使用
	开漏（Open-Drain）		1		不使用
输入	模拟模式	0	0	00：保留	不使用
	浮空输入		1		不使用
	下拉输入	1	0		0
	上拉输入				1

IO 在不同的配置下的输入输出特性如下表所示:

表 6-2 IO 不同配置的输入输出特性

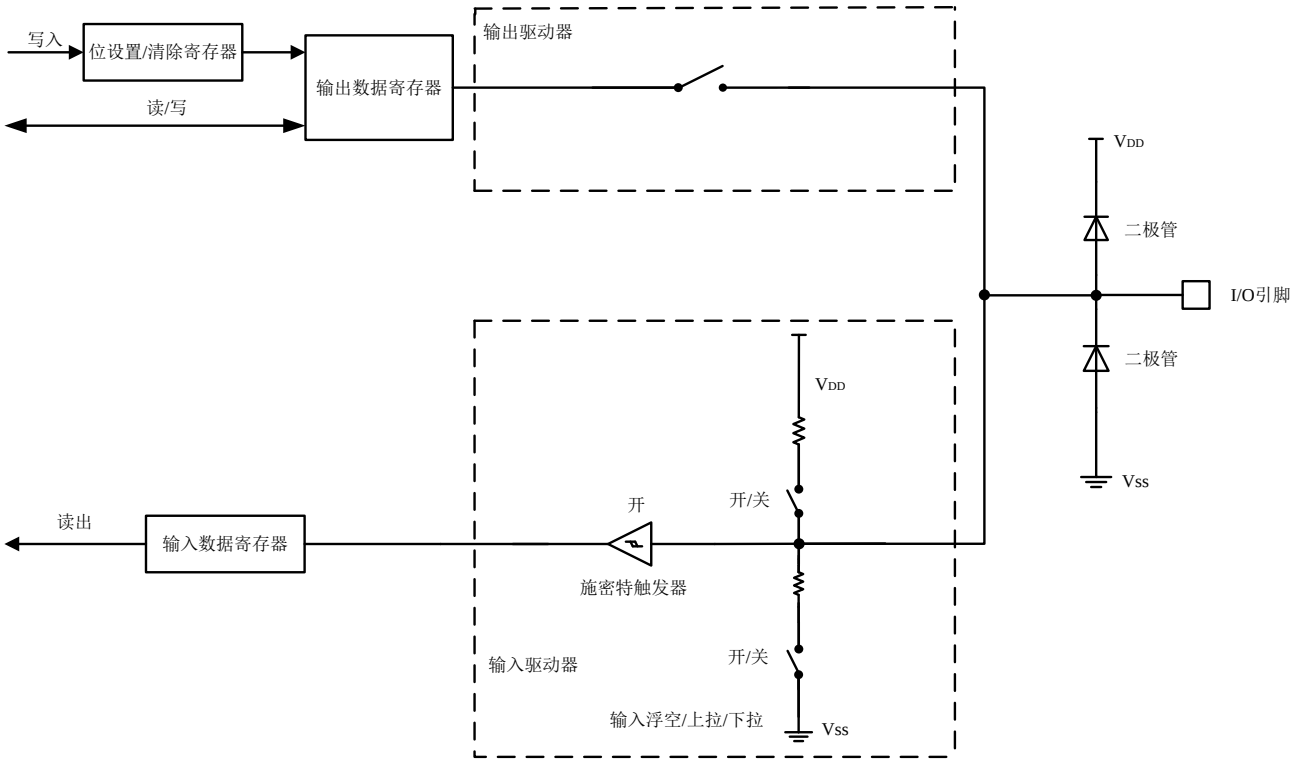
特性	GPIO输入	GPIO输出	模拟模式	外设复用
输出缓冲器	禁能	使能	禁能	根据外设功能配置
施密特触发器	使能	使能	禁能 输出值被强制为0	根据外设功能配置
上下拉/浮空	可配	禁能	禁能	根据外设功能配置
开漏模式	禁能	可配， 输出数据为”0”时GPIO输出0，”1”时GPIO高阻	禁能	可配，输出数据为”0”时GPIO输出0，”1”时GPIO高阻
输入数据寄存器（IO状态）	可读写	可读	读出为0	可读
输出数据寄存器（写入值）	无效	可读写	无效	可读

6.2.1.1 输入模式

当 I/O 端口配置为输入模式时：

- 出现在 I/O 脚上的数据在每个 APB2 时钟被采样到输入数据寄存器状态
- 通过对数据寄存器的读访问得到 I/O
- 输出缓冲器被禁止
- 施密特触发输入被激活
- 根据输入配置（上拉，下拉或浮动）的不同，弱上拉和下拉电阻被连接

图 6-2 输入浮空/上拉/下拉配置

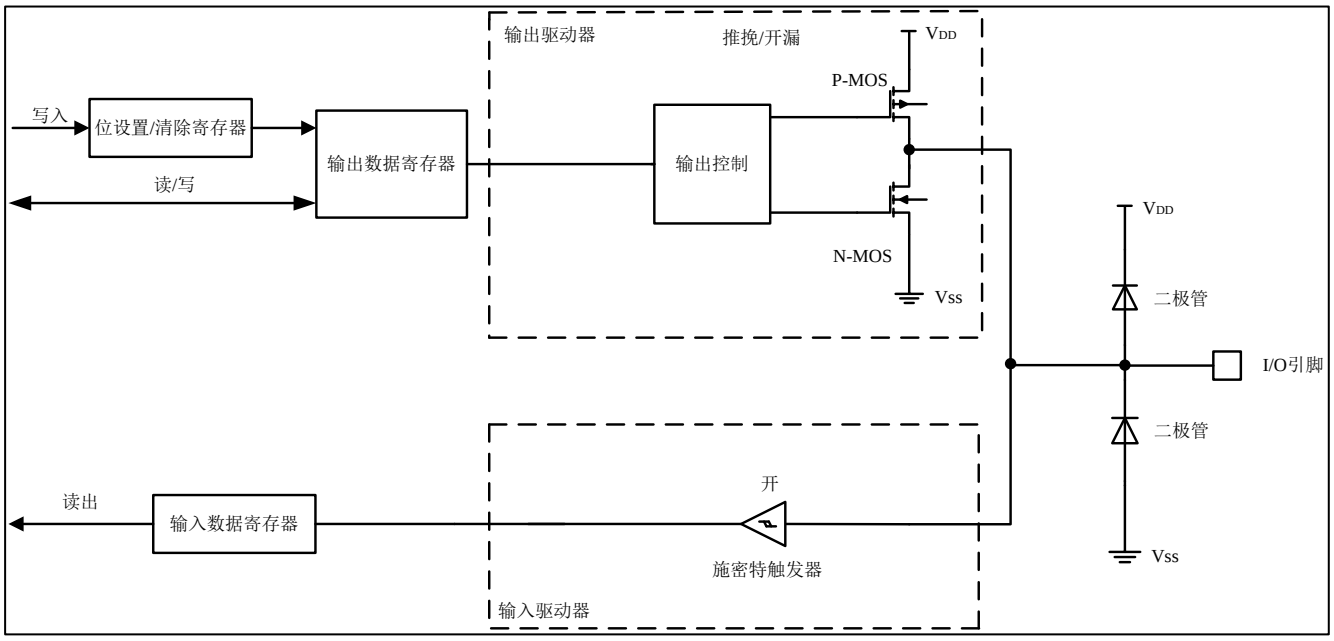


6.2.1.2 输出模式

当 I/O 端口配置为输出时：

- 施密特触发输入被激活
- 弱上拉和下拉电阻被禁止
- 输出缓冲器被激活
  - ◆ 开漏模式： 输出寄存器上的'0'激活 N-MOS，引脚输出低电平
    输出寄存器上的'1'端口置于高阻状态（P-MOS 从不被激活）
  - ◆ 推挽模式： 输出寄存器上的'0'激活 N-MOS，引脚输出低电平
    输出寄存器上的'1'激活 P-MOS，引脚输出高电平
- 出现在 I/O 脚上的数据在每个 APB2 时钟被采样到输入数据寄存器
- 对输入数据寄存器的读访问可得到 I/O 状态
- 对输出数据寄存器的读访问得到最后一次写的值

图 6-3 输出模式配置



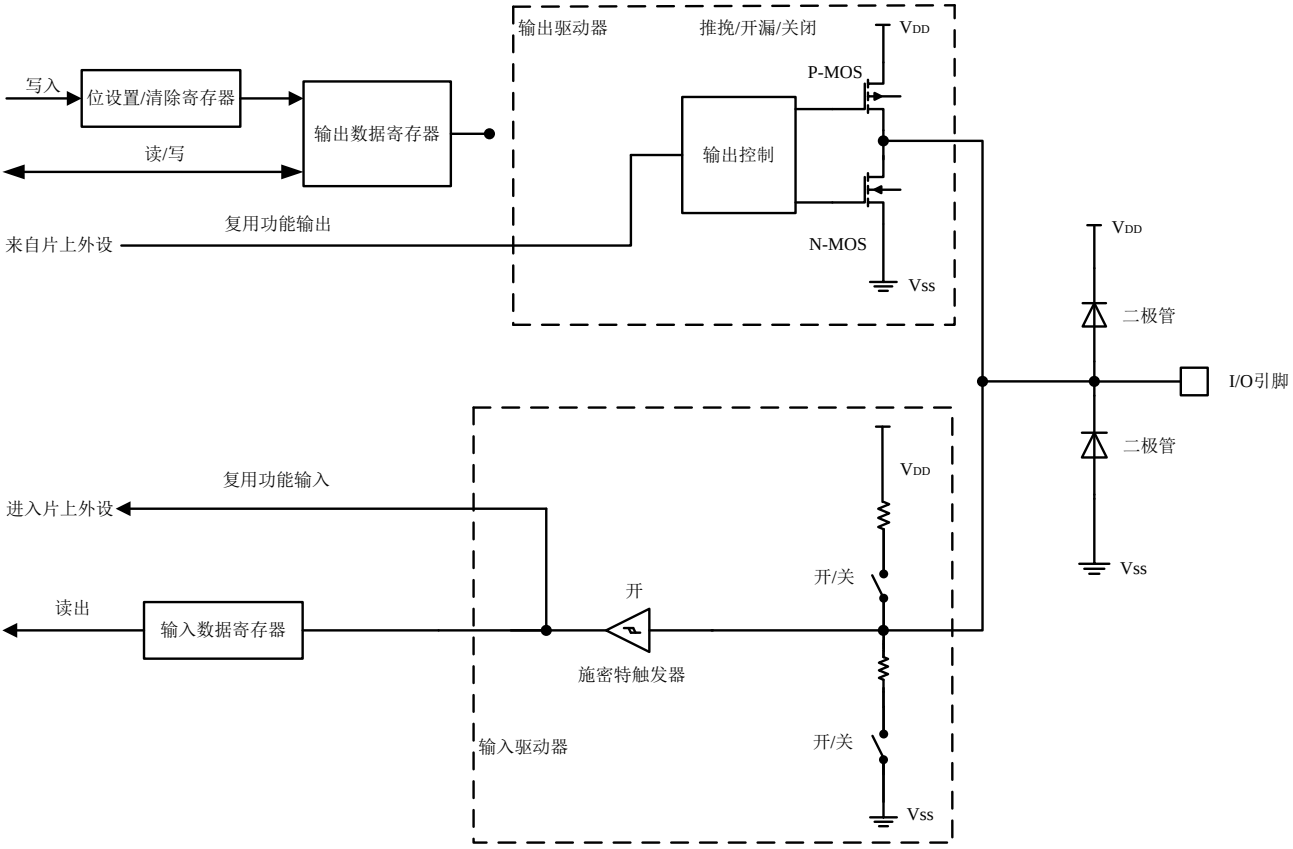
6.2.1.3 复用功能模式

当 I/O 端口配置为复用功能时：

- 施密特触发输入被激活
- 弱上拉和下拉电阻被禁止
- 在开漏或推挽式配置中，输出缓冲器被打开
- 内置外设的信号驱动输出缓冲器
- 在每个 APB2 时钟周期，出现在 I/O 脚上的数据被采样到输入数据寄存器

- 对输入数据寄存器的读访问可得到 I/O 状态
- 对输出数据寄存器的读访问得到最后一次写的值

图 6-4 复用功能配置

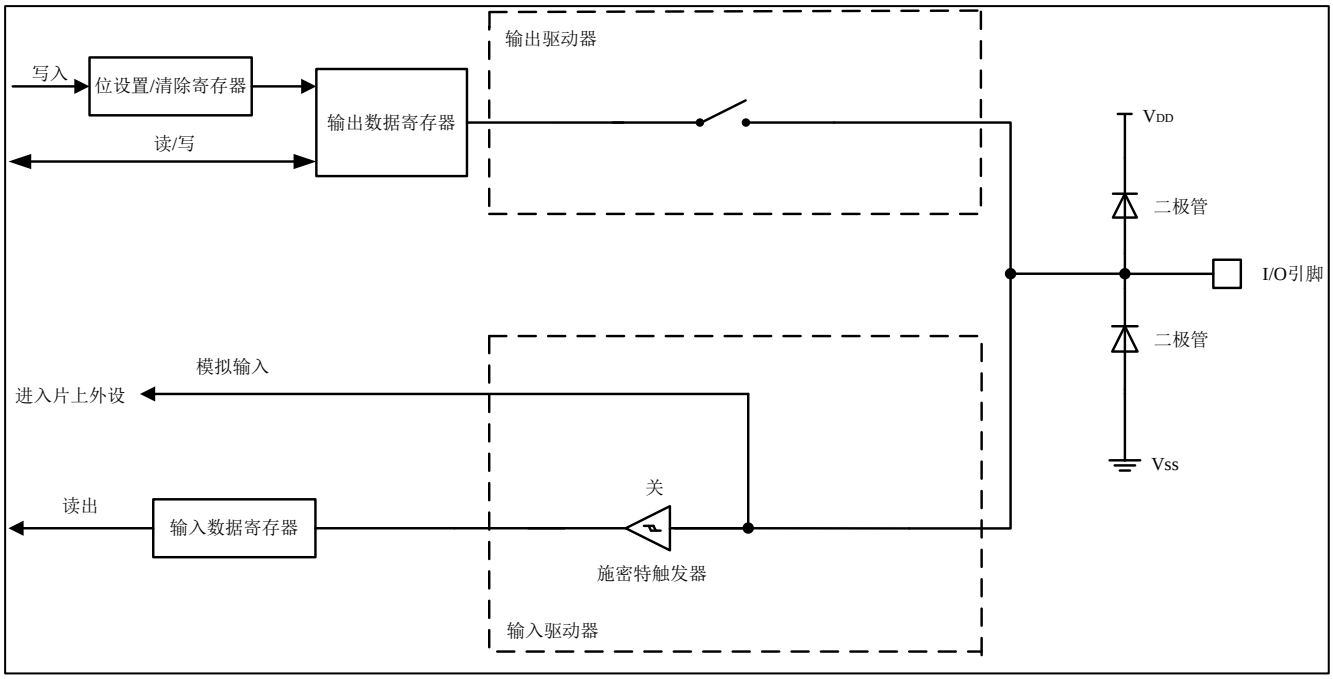


6.2.1.4 模拟模式

当 I/O 端口被配置为模拟模式配置时:

- 弱上拉和下拉电阻被禁止
- 读取输入数据寄存器时数值为'0'
- 输出缓存器被禁止
- 施密特触发输入被禁止，输出值被强置为'0'（ 实现了每个模拟 I/O 引脚上的零消耗）

图 6-5 高阻抗的模拟模式配置



6.2.2 复位后状态

复位期间和刚复位后，复用功能未开启，I/O 端口被配置成模拟功能模式（PCFGy[1:0]=00b，PMODEy[1:0]=00b）。但有以下几个例外的信号：

- BOOT0、NRST、OSC_IN、OSC_OUT 默认无 GPIO 功能：
  - BOOT0 引脚默认输入下拉
  - NRST 上拉输入输出
- 复位后，调试系统相关的引脚默认状态时启动 4-wire JTAG ，JTAG 引脚被置于输入上拉或下拉模式：
  - PA15:JTDI 置于输入上拉模式
  - PA14:JTCK 置于输入下拉模式
  - PA13:JTMS 置于输入上拉模式
  - PB3:JTDO 置于推挽输出无上下拉
- PD0 和 PD1：
  - PD0 和 PD1 在 80 及以上引脚封装默认模拟模式
  - PD0 和 PD1 在 80 以下引脚封装复用到 OSC_IN/OUT
- PC13、PC14、PC15：
  - PC13 ~ 15 为备电域下的三个 IO，备份域初次上电默认为 GPIO 模拟模式；
- PB2/BOOT1：

- PB2/BOOT1 处于 PB2/BOOT1 的下拉输入状态；
- BOOT0 默认输入下拉，参照下表，即 BOOT 的 PAD 若未连接，默认选择 Flash 主存储区

启动模式选择引脚		启动模式	说明
BOOT1	BOOT0		
x	0	主闪存存储器	主闪存存储器作为启动区域
0	1	系统存储器	系统存储器作为启动区域
1	1	内置SRAM	内置SRAM作为启动区域

6.2.3 单独的位设置和位清除

通过对置位寄存器（GPIOx_PBSC）和复位寄存器（GPIOx_PBC）中想要更改的位写'1'来实现对数据寄存器（GPIOx_POD）的个别位操作，可以写一个或多个位。写'1'的位被相应的置位或清除，没被写'1'的位将不被更改。软件不需要禁止中断，在单次 APB2 写操作里完成。

6.2.4 外部中断/唤醒线

所有端口都有外部中断能力，可以在 EXTI 模块中配置：

- 端口必须配置成输入模式
- 所有端口可配置用于 SLEEP/STOP0/STOP2 模式唤醒，支持上升或下降沿可配
- PA0 可用于 Standby 模式唤醒
- 通用 I/O 端口以图 7-2 的方式连接到 16 个外部中断/事件线上，由寄存器 AFIO_EXTI_CFGx 配置

6.2.5 复用功能

当端口配置位 AFIO 时，端口引脚用作外设复用功能，使用前必须对端口位配置寄存器（GPIOx_PL_CFG/ GPIOx_PH_CFG）编程，具体如下：

- 复用的输入功能：端口必须配置成输入模式（浮空、上拉或下拉）且输入引脚必须由外部驱动。
- 复用输出功能：端口必须配置成复用功能输出模式（推挽或开漏）。
- 双向复用功能：端口位必须配置复用功能输出模式（推挽或开漏）。此时，输入驱动器被配置成浮空输入模式。

端口在复用输出模式下，引脚和输出寄存器断开，并和片上外设的输出信号连接。如果软件把一个 GPIO 脚配置成复用输出功能，但是外设没有被激活，它的输出将不确定。

6.2.5.1 时钟输出 MCO

微控制器允许输出时钟信号到外部 MCO 引脚（PA8）。PA8 必须被配置为复用推挽输出功能模式。以下四个时钟信号可被选作 MCO 时钟，时钟的选择由时钟配置寄存器（RCC_CFG）中的 MCO[2:0]位控制：

- 分频的 SYSCLK

- HSI
- HSE
- 分频的 PLL 时钟

6.2.5.2 软件重新映射 I/O 复用功能

为拓展不同器件封装下的复用外设功能灵活性，可以把一些外设复用功能重新映射到其他引脚上。可以通过软件配置相应的寄存器（AFIO_RMP_CFG）来完成。这时，复用功能就和他原来的引脚断开，重新映射到新的引脚上了。

6.2.5.3 备电域 PC13～PC15 功能重映射

6.2.5.3.1 备电域 PC13～PC15 功能描述

PC13～PC15 三个 PAD 位于备电域下，可以作为 GPIO 模式或复用功能模式：

- 当备份区域由 VDD（内部模拟开关连到 VDD）供电时，下述功能可用：
  - PC14 和 PC15 可以用于 GPIO 或 LSE 引脚
  - PC13 可以作为通用 I/O 口、TAMPER 引脚、RTC 校准时钟、RTC 闹钟或秒输出（参见备份寄存器（BKP））

注：因为模拟开关只能通过少量的电流（3mA），在输出模式下使用 PC13 至 PC15 的 I/O 口功能是有限制的：速度必须限制在 2MHz 以下，最大负载为 30pF，而且这些 I/O 口绝对不能当作电流源（如驱动 LED）。

- 当后备区域由 VBAT 供电时（VDD 消失后模拟开关连到 VBAT），可以使用下述功能：
  - PC14 和 PC15 只能用于 LSE 引脚
  - PC13 可以作为 TAMPER 引脚、RTC 闹钟或秒输出（参见 RTC 时钟校准寄存器（BKP_RTCCR）），在 PC13 引脚（当该引脚不用于侵入检测时）上输出 RTC 校准时钟，RTC 闹钟脉冲或者秒脉冲。为方便测量，RTC 时钟可以经 64 分频输出到侵入检测引脚 TAMPER 上。通过设置 RTC 校验寄存器（BKP_RTCCR）的 CCO 位来开启这一功能。

6.2.5.3.2 PC13～PC15 功能映射

根据前面的描述，整理了 PC13～PC15 不同模式的配置条件如下：

PC14和PC15	条件	PAD模式配置
GPIO模式	LSE关闭，且备份域由VDD供电，且不进入低功耗模式（Standby, Stop2）关闭1.1V电源时，才能用于GPIO模式	GPIO的模式由应用决定
LSE模式	LSE优先级高，上述条件不满足即进入LSE模式	模拟模式
PC13	条件	PAD模式配置
GPIO模式	备电域由VDD供电，且不进入低功耗模式（Standby, Stop2）关闭1.1V电源时，才能用于GPIO模式	GPIO的模式由应用决定

TAMPER引脚	任何时候均可配置于此项输入	浮空输入
RTC校准时钟，RTC闹钟脉冲或者秒脉冲	当该引脚不用于侵入检测时，可用于输出RTC校准时钟，RTC闹钟脉冲或者秒脉冲，因此优先级TAMPER功能低。	复用推挽输出

由上，PC13 ~ PC15 三个 PAD 的 Iom 管理的电源域划分：

■ GPIOx_PH_CFG/GPIOx_PID/GPIOx_POD/GPIOx_PBSC/GPIOx_PBC/GPIOx_PLOCK_CFG 等 GPIO 相关的配置寄存器都在 Core 电源域；

■ 这三个 PAD 的 Mux 逻辑在 VBAT 域，默认为 GPIO 浮空输入模式；PC14 和 PC15 根据 LSEEN、备份域供电电源控制信号、芯片模式信号、GPIOC_PH_CFG 、TAMPER 和时钟输出控制决定处于何种模式以及 Mux。

注：备电域由 VDD 供电还是 VBAT 供电的控制信号，由 PWR 模块自动切换，切换信号由 VDD 域的 POR 信号得到。

6.2.5.4 把 OSC_IN/OSC_OUT 引脚作为 GPIO 端口 PD0/PD1

如应用中未开启外部高速振荡器，引脚 OSC_IN/OSC_OUT 可以用做 GPIO 的 PD0/PD1,通过设置复用重映射和调试 I/O 配置寄存器（AFIO_RMP_CFG）实现，如下描述：

位15	PD01_RMP ：端口D0/端口D1映像到OSC_IN/OSC_OUT （Port D0/Port D1 mapping on OSC_IN/OSC_OUT） 该位可由软件置'1'或置'0'。它控制PD0和PD1的GPIO功能映像。当不使用主振荡器HSE时（系统运行于内部的8MHz阻容振荡器），PD0和PD1可以映像到OSC_IN和OSC_OUT引脚。 0：不进行PD0和PD1的重映像； 1：PD0映像到OSC_IN，PD1映像到OSC_OUT。 此功能只能适用于48和64引脚的封装（在80、100脚和128脚的封装上有PD0和PD1，不必重映像）。

注：外部中断/事件功能没有被重映射。在 48 和 64 引脚的封装上，PD0 和 PD1 不能用来产生外部中断/事件。即对于 LQFP48/64 封装，只有 OSC_IN 和 OSC_OUT 两个引脚（没有 PD0 和 PD1），当不用外部晶振的时候，可以映射为 PD0 和 PD1;对于 80 脚和以上 封装，PD0 和 PD1 两个脚有了，OSC_IN 和 OSC_OUT 也有了，不需要重新将 PD0/1 映射到 OSC_IN/OSC_OUT。

6.2.5.5 2-wire/4-wire JTAG 复用功能重映射

芯片上电默认使能 4-wire JTAG 调试接口，调试接口被映射到 GPIO 端口上，如下表所示。

复用功能	GPIO端口
JTMS	PA13
JTCK	PA14
JTDI	PA15
JTDO	PB3

如调试期间需要使用其 GPIO 功能，可通过设置复用重映射和调试 I/O 配置寄存器（AFIO_RMP_CFG）的 SWJ_JTAG_CFG[2:0]位，可以改变上述重映像配置。参见下表。

表 6-3 调试端口映像

SWJ_JTAG_CFG[2:0]	可能的调试端口	SWJ I/O引脚分配			
		PA13/ JTMS	PA14/ JTCK	PA15/ JTDI	PB3/ JTDO
000	完全SWJ（4-wire/2-wire JTAG）（复位状态）	I/O不可用	I/O不可用	I/O不可用	I/O不可用
001	完全SWJ（4-wire/2-wire JTAG）	I/O不可用	I/O不可用	I/O不可用	I/O不可用
010	关闭4-wire JTAG， 启用2-wire JTAG	I/O不可用	I/O不可用	I/O可用	I/O可用
100	关闭4-wire JTAG， 关闭2-wire JTAG	I/O可用	I/O可用	I/O可用	I/O可用
其它	禁用				

6.2.5.5.1 SWJ_CFG 配置事项

当 APB 桥的写缓冲区满了的时侯，在写 AFIO_RMP_CFG 寄存器时需要多用一个 APB 周期。这是因为 JTAGSW 脚的释放需要两个 APB 周期完成，以保证 NTRST 和 TCK 的输入信号为一个干净的电平。

第一个周期：输入 1 / 0 的 JTAGSW 输入到内核的信号被接为 0 或 1（TDI/TMS 接 1，TCK 接 0）；

第二个周期：由 IOM 控制 SWJTAG I/O 引脚的控制信号（比如方向，上下拉，施密特输入等）。

6.2.5.5.2 上下拉配置

由于 JTAG 的引脚直接连接到内部的调试寄存器上（JTCK/SWCLK 直接连接到时钟端），因此必须保证 JTAG 的输入引脚不能处于浮空态。为了避免任何非可控的 IO 电平，JTAG 的输入引脚固定内部的上下拉：

- JTDI：内部上拉
- JTMS:内部上拉
- JTCK:内部下拉

只要 JTAG I/O 被用户代码释放，GPIO 控制器重新获得控制权，需要保证 GPIO 寄存器的复位态和上述保持一致。

6.2.5.6 ADC 外部触发复用功能重映射

ADC 的注入转换和规则转换的外部触发源支持重映射，参阅复用重映射和调试 I/O 配置寄存器（AFIO_RMP_CFG）。

表 6-4 ADC1 外部触发注入转换复用功能重映射

复用功能	ADC1_ETRI = 0	ADC1_ETRI = 1
ADC1外部触发注入转换	ADC1外部触发注入转换与EXTI15相连	ADC1外部触发注入转换与TIM8_CH4相连

表 6-5 ADC1 外部触发规则转换复用功能重映射

复用功能	ADC1_ETRR = 0	ADC1_ETRR = 1

ADC1外部触发规则转换	ADC1外部触发注入转换与EXTI11相连	ADC1外部触发注入转换与TIM8_TRGO相连

表 6-6 ADC2 外部触发注入转换复用功能重映射

复用功能	ADC2_ETRI = 0	ADC2_ETRI = 1
ADC2外部触发注入转换	ADC2外部触发注入转换与EXTI15相连	ADC2外部触发注入转换与TIM8_CH4相连

表 6-7 ADC2 外部触发规则转换复用功能重映射

复用功能	ADC2_ETRR = 0	ADC2_ETRR = 1
ADC2外部触发规则转换	ADC2外部触发注入转换与EXTI11相连	ADC2外部触发注入转换与TIM8_TRGO相连

表 6-8 ADC3 外部触发注入转换复用功能重映射

复用功能	ADC3_ETRI = 0	ADC3_ETRI = 1
ADC3外部触发注入转换	ADC3外部触发注入转换与EXTI14相连	ADC3外部触发注入转换与TIM5_CH3相连

表 6-9 ADC3 外部触发规则转换复用功能重映射

复用功能	ADC3_ETRR = 0	ADC3_ETRR = 1
ADC3外部触发规则转换	ADC3外部触发注入转换与EXTI10相连	ADC3外部触发注入转换与TIM5_CH4相连

表 6-10 ADC4 外部触发注入转换复用功能重映射

复用功能	ADC4_ETRI = 0	ADC4_ETRI = 1
ADC4外部触发注入转换	ADC4外部触发注入转换与EXTI14相连	ADC4外部触发注入转换与TIM8_CH3相连

表 6-11 ADC4 外部触发规则转换复用功能重映射

复用功能	ADC4_ETRR = 0	ADC4_ETRR = 1
ADC4外部触发规则转换	ADC4外部触发注入转换与EXTI10相连	ADC4外部触发注入转换与TIM8_CH4相连

6.2.5.7 TIMx 复用功能重映射

表 6-12 TIM5 复用功能重映射

复用功能	TIM5CH4_RMP = 0	TIM5CH4_RMP = 1
TIM5_CH4	TIM5的通道4连至PA3	LSI内部时钟连至TIM5_CH4的输入作为校准使用

表 6-13 TIM4 复用功能重映射

复用功能	TIM4_RMP = 0	TIM4_RMP = 1
TIM4_CH1	PB6	PD12
TIM4_CH2	PB7	PD13
TIM4_CH3	PB8	PD14

TIM4_CH4	PB9	PD15

表 6-14 TIM3 复用功能重映射

复用功能	TIM3_RMP[1:0] = 00 （没有重映像）	TIM3_RMP[1:0] = 10 （部分重映像）	TIM3_RMP[1:0] = 11 （完全重映像）
TIM3_CH1	PA6	PB4	PC6
TIM3_CH2	PA7	PB5	PC7
TIM3_CH3	PB0		PC8
TIM3_CH4	PB1		PC9

表 6-15 TIM2 复用功能重映射

复用功能	TIM2_RMP[1:0] = 00 （没有重映像）	TIM2_RMP[1:0] = 01 （部分重映像）	TIM2_RMP[1:0] = 10 （完全重映像）	TIM2_RMP[1:0] = 11 （完全重映像）
TIM2_CH1	PA0	PA15	PA0	PA15
TIM2_CH2	PA1	PB3	PA1	PB3
TIM2_CH3	PA2		PB10	
TIM2_CH4	PA3		PB11	

表 6-16 TIM1 复用功能重映射

复用功能	TIM1_RMP[1:0] = 00 （没有重映像）	TIM1_RMP[1:0] = 01 （部分重映像）	TIM1_RMP[1:0] = 10 （完全重映像）	TIM1_RMP[1:0] = 11 （完全重映像）
TIM1_ETR	PA12		PA12	PE7
TIM1_CH1	PA8		PA8	PE9
TIM1_CH2	PA9		PA9	PE11
TIM1_CH3	PA10		PA10	PE13
TIM1_CH4	PA11		PA11	PE14
TIM1_BKIN	PB12	PA6	PB5	PE15
TIM1_CH1N	PB13	PA7	PB13	PE8
TIM1_CH2N	PB14	PB0	PB14	PE10
TIM1_CH3N	PB15	PB1	PB15	PE12

表 6-17 TIM8 复用功能重映射

复用功能	TIM1_RMP[1:0] = 00 （没有重映像）	TIM1_RMP[1:0] = 10 （完全重映像）	TIM1_RMP[1:0] = 11 （完全重映像）
TIM8_ETR	PA0	PB4	PB4
TIM8_CH1	PC6	PC6	PD14
TIM8_CH2	PC7	PC7	PD15
TIM8_CH3	PC8	PC8	PC8
TIM8_CH4	PC9	PC9	PC9
TIM8_BKIN	PA6	PB3	PB3
TIM8_CH1N	PA7	PA15	PA15
TIM8_CH2N	PB0	PC12	PC12
TIM8_CH3N	PB1	PD2	PD2

6.2.5.8 CAN 复用功能重映射

6.2.5.8.1 CAN1 复用功能重映射

CAN1 信号可以被映射到端口 A、端口 B 或端口 D 上，如下表所示。对于端口 D，在 48 和 64 引脚的封装上没有重映射功能（这些封装上没有 PD0 和 PD1）。

表 6-18 CAN1 复用功能重映射

复用功能	CAN1_RMP[1:0] = 00	CAN1_RMP[1:0] = 01	CAN1_RMP[1:0] = 10	CAN1_RMP[1:0] = 11
CAN1_RX	PA11	PD8	PB8	PD0
CAN1_TX	PA12	PD9	PB9	PD1

6.2.5.8.2 CAN2 复用功能重映射

CAN2 信号可以被映射到端口 B 或端口 D 上，如下表所示。

表 6-19 CAN2 复用功能重映射

复用功能	CAN2_RMP[1:0] = 00	CAN2_RMP[1:0] = 01	CAN1_RMP[1:0] = 11
CAN2_RX	PB12	PB5	PD10
CAN2_TX	PB13	PB6	PD11

6.2.5.9 USARTx 复用功能重映射

6.2.5.9.1 USART1 管脚重映射

USART1/2/3 三个接口具有重映射功能，参见复用重映射配置寄存器（AFIO_RMP_CFG）。

表 6-20 USART1 复用功能重映射

复用功能	USART1_RMP = 0	USART1_RMP = 1
USART1_CTS	PA11	
USART1_RTS	PA12	
USART1_TX	PA9	PB6
USART1_RX	PA10	PB7
USART1_CK	PA8	

6.2.5.9.2 USART2 管脚重映射

表 6-21 USART2 复用功能重映射

复用功能	USART2_RMP[1:0] = 00	USART2_RMP[1:0] = 01	USART2_RMP[1:0] = 10	USART2_RMP [1:0] =11
USART2_CTS	PA0	PD3	PC6	PA15
USART2_RTS	PA1	PD4	PC7	PB3
USART2_TX	PA2	PD5	PC8	PB4
USART2_RX	PA3	PD6	PC9	PB5
USART2_CK	PA4	PD7	/	CK

6.2.5.9.3 USART3 管脚重映射

表 6-22 USART3 复用功能重映射

复用功能	USART3_RMP[1:0] = 00	USART3_RMP[1:0] = 01	USART3_RMP[1:0] = 11
USART3_TX	PB10	PC10	PD8
USART3_RX	PB11	PC11	PD9
USART3_CK	PB12	PC12	PD10
USART3_CTS	PB13		PD11
USART3_RTS	PB14		PD12

6.2.5.10 UARTx 复用功能重映射

UART4/5/6/7 三个接口具有重映射功能，参见复用重映射配置寄存器（AFIO_RMP_CFG3）。

6.2.5.10.1 UART4 管脚重映射

表 6-23 UART4 复用功能重映射

复用功能	UART4_RMP[1:0] = 00	UART4_RMP[1:0] = 01	UART4_RMP[1:0] = 10	UART4_RMP [1:0] =11
UART4_TX	PC10	PB2	PA13	PD0
UART4_RX	PC11	PE7	PA14	PD1

6.2.5.10.2 UART5 管脚重映射

表 6-24 UART5 复用功能重映射

复用功能	UART5_RMP[1:0] = 00	UART5_RMP[1:0] = 01	UART5_RMP[1:0] = 10	UART5_RMP [1:0] =11
UART5_TX	PC12	PB13	PE8	PB8
UART5_RX	PD2	PB14	PE9	PB9

6.2.5.10.3 UART6 管脚重映射

表 6-25 UART6 复用功能重映射

复用功能	UART6_RMP[1:0] = 00	UART6_RMP[1:0] = 10	UART6_RMP[1:0] = 11
UART6_TX	PE2	PC0	PB0
UART6_RX	PE3	PC1	PB1

6.2.5.10.4 UART7 管脚重映射

表 6-26 UART7 复用功能重映射

复用功能	UART7_RMP[1:0] = 00	UART7_RMP[1:0] = 01	UART7_RMP[1:0] = 11
UART7_TX	PC4	PC2	PG0
UART7_RX	PC5	PC3	PG1

6.2.5.11 I2C 复用功能重映射

6.2.5.11.1 I2C1 管脚重映射

参见复用重映射配置寄存器（AFIO_RMP_CFG）。

表 6-27 I2C1 管脚重映射

复用功能	I2C1_RMP= 0	I2C1_RMP= 1
I2C1_SCL	PB6	PB8
I2C1_SDA	PB7	PB9
I2C1_SMBA	PB5	

6.2.5.11.2 I2C2 管脚重映射

参见复用重映射配置寄存器（AFIO_RMP_CFG3）。

表 6-28 I2C2 管脚重映射

复用功能	I2C2_RMP[1:0] = 00	I2C2_RMP[1:0] = 10	I2C2_RMP[1:0] = 11
I2C2_SCL	PB10	PG2	PA4
I2C2_SDA	PB11	PG3	PA5
I2C2_SMBA	PB12		

6.2.5.11.3 I2C3 管脚重映射

参见复用重映射配置寄存器（AFIO_RMP_CFG3）。

表 6-29 I2C3 管脚重映射

复用功能	I2C3_RMP[1:0] = 00	I2C3_RMP[1:0] = 10	I2C3_RMP[1:0] = 11
I2C3_SCL	PC0	PF4	PC4
I2C3_SDA	PC1	PF5	PC5

6.2.5.11.4 I2C4 管脚重映射

参见复用重映射配置寄存器（AFIO_RMP_CFG3）。

表 6-30 I2C4 管脚重映射

复用功能	I2C4_RMP[1:0] = 00	I2C4_RMP[1:0] = 10	I2C4_RMP[1:0] = 11
I2C4_SCL	PC6	PD14	PA9
I2C4_SDA	PC7	PD15	PA10

6.2.5.12 SPI/I2S 复用功能重映射

6.2.5.12.1 SPI1 管脚重映射

参见复用重映射配置寄存器（AFIO_RMP_CFG）。

表 6-31 SPI1 管脚重映射

复用功能	SPI1_RMP[1:0] = 00	SPI1_RMP[1:0] = 01	SPI1_RMP[1:0] = 10	SPI1_RMP [1:0] =11
SPI1_NSS	PA4	PA15	PB2	PB2
SPI1_SCK	PA5	PB3	PA5	PE7
SPI1_MISO	PA6	PB4	PA6	PE8
SPI1_MOSI	PA7	PB5	PA7	PE9

6.2.5.12.2 SPI2/I2S2 管脚重映射

参见复用重映射配置寄存器（AFIO_RMP_CFG3）。

表 6-32 SPI2/I2S2 管脚重映射

复用功能	SPI2_RMP[1:0] = 00	SPI2_RMP[1:0] = 01	SPI2_RMP[1:0] = 11
SPI2_NSS/I2S2_WS	PB12	PC6	PE10
SPI2_SCK/I2S2_CK	PB13	PC7	PE11
SPI2_MISO	PB14	PC8	PE12
SPI2_MOSI/I2S2_SD	PB15	PC9	PE13
I2S2_MCK	PC6		

6.2.5.12.3 SPI3/I2S3 管脚重映射

参见复用重映射配置寄存器（AFIO_RMP_CFG3）。

表 6-33 SPI3/I2S3 管脚重映射

复用功能	SPI3_RMP[1:0] = 00	SPI3_RMP[1:0] = 01	SPI3_RMP[1:0] = 10	SPI3_RMP[1:0] = 11
SPI3_NSS/I2S3_WS	PA15	PD2	PD8	PC2
SPI3_SCK/I2S3_CK	PB3	PC10	PD9	PC3
SPI3_MISO	PB4	PC11	PD11	PA0
SPI3_MOSI/I2S3_SD	PB5	PC12	PD12	PA1
I2S3_MCK	PC7			

6.2.5.13 QSPI 复用功能重映射

QSPI 只支持主机模式，且不支持多主模式。单线模式下 QSPI_IO1 作为 MISO，支持推挽复用输出，还可以支持浮空输入或上拉输入。参见复用重映射配置寄存器（AFIO_RMP_CFG3）。

表 6-34 QSPI 管脚重映射

复用功能	QSPI_RMP[1:0] = 00	QSPI_RMP[1:0] = 01	QSPI_RMP[1:0] = 11
QSPI_NSS	PA4	PF0	PC10
QSPI_CLK	PA5	PF1	PC11
QSPI_IO0	PA6	PF2	PC12
QSPI_IO1	PA7	PF3	PD0
QSPI_IO2	PC4	PF4	PD1
QSPI_IO3	PC5	PF5	PD2

6.2.5.14 XFMC 复用功能重映射

参见复用重映射配置寄存器（AFIO_RMP_CFG3）。

表 6-35 XFMC 管脚重映射

复用功能	XFMC_RMP[1:0] = 00	XFMC_RMP[1:0] = 01	XFMC_RMP[1:0] = 11
XFMC_A0	PF0		
XFMC_A1	PF1		
XFMC_A2	PF2		
XFMC_A3	PF3		
XFMC_A4	PF4		
XFMC_A5	PF5		

复用功能	XFMC_RMP[1:0] = 00	XFMC_RMP[1:0] = 01	XFMC_RMP[1:0] = 11
XFMC_A6	PF12		
XFMC_A7	PF13		
XFMC_A8	PF14		
XFMC_A9	PF15		
XFMC_A10	PG0		
XFMC_A11	PG1		
XFMC_A12	PG2		
XFMC_A13	PG3		
XFMC_A14	PG4		
XFMC_A15	PG5		
XFMC_A16	PD11		PC6
XFMC_A17	PD12		PC7
XFMC_A18	PD13		
XFMC_A19	PE3		
XFMC_A20	PE4		
XFMC_A21	PE5		
XFMC_A22	PE6		
XFMC_A23	PE2		
XFMC_D0	PD14		
XFMC_D1	PD15		
XFMC_D2	PD0		
XFMC_D3	PD1		
XFMC_D4	PE7		
XFMC_D5	PE8		
XFMC_D6	PE9		
XFMC_D7	PE10		
XFMC_D8	PE11		
XFMC_D9	PE12		
XFMC_D10	PE13		
XFMC_D11	PE14	PB10	
XFMC_D12	PE15	PB11	
XFMC_D13	PD8		
XFMC_D14	PD9		
XFMC_D15	PD10		
XFMC_CLK	PD3		
XFMC_NOE	PD4		PC9
XFMC_NWE	PD5		PA10
XFMC_NWAIT	PD6		PB8
XFMC_NE1 XFMC_NCE2	PD7		PB9
XFMC_NE2	PG9		PB6

复用功能	XFMC_RMP[1:0] = 00	XFMC_RMP[1:0] = 01	XFMC_RMP[1:0] = 11
XFMC_NCE3			
XFMC_NADV	PB7		
XFMC_NBL0	PE0		
XFMC_NBL1	PE1		

6.2.6 外设的 IO 配置

表 6-36 ADC/DAC

ADC/DAC引脚	GPIO配置
ADC	模拟模式
DAC	模拟模式

表 6-37 TIM1/TIM8

TIM1/TIM8引脚	配置	PAD配置模式
TIM1/8_CHx	输入捕获通道x	浮空输入
	输出比较通道x	推挽复用输出
TIM1/8_CHxN	互补输出通道x	推挽复用输出
TIM1/8_BKIN	刹车输入	浮空输入
TIM1/8_ETR	外部触发时钟输入	浮空输入

表 6-38 TIM2/3/4/5

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

表 6-39 BxCAN

BxCAN引脚	GPIO配置
CAN_TX	推挽复用输出
CAN_RX	浮空输入或带上拉输入

表 6-40 USART

USART引脚	配置	GPIO配置
USARTx_TX	全双工模式	推挽复用输出
	半双工同步模式	推挽复用输出
USARTx_RX	全双工模式	浮空输入或带上拉输入
	半双工同步模式	未用，可作为通用I/O
USARTx_CK	同步模式	推挽复用输出
USARTx_RTS	硬件流量控制	推挽复用输出
USARTx_CTS	硬件流量控制	浮空输入或带上拉输入

表 6-41 I2C

I2C引脚	配置	GPIO配置
I2Cx_SCL	I2C时钟	开漏复用输出
I2Cx_SDA	I2C数据	开漏复用输出
I2Cx_SMBA	SMBA数据	推挽复用输出

表 6-42 SPI

SPI引脚	配置	GPIO配置
SPIx_SCK	主模式	推挽复用输出
	从模式	浮空输入
SPIx_MOSI	全双工模式/主模式	推挽复用输出
	全双工模式/从模式	浮空输入或带上拉输入或推挽复用输出
	简单的双向数据线/主模式	推挽复用输出
	简单的双向数据线/从模式	未用，可作为通用I/O
SPIx_MISO	全双工模式/主模式	浮空输入或带上拉输入或推挽复用输出
	全双工模式/从模式	推挽复用输出
	简单的双向数据线/主模式	未用，可作为通用I/O
	简单的双向数据线/从模式	推挽复用输出
SPIx_NSS	硬件主/从模式	浮空输入或带上拉输入或带下拉输入
	硬件主模式/NSS输出使能	推挽复用输出（作为主机时NSS可选择idle高阻或idle为1）
	软件模式	未用，可作为通用I/O

表 6-43 I2S

I2S引脚	配置	GPIO配置
I2Sx_WS	主模式	推挽复用输出
	从模式	浮空输入
I2Sx_CK	主模式	推挽复用输出
	从模式	浮空输入
I2Sx_SD	发送器	推挽复用输出
	接收器	浮空输入或带上拉输入或带下拉输入
I2Sx_MCK	主模式	推挽复用输出
	从模式	未用，可作为通用I/O

表 6-44 QSPI

QSPI引脚	GPIO配置	备注
QSPI_IO3	推挽复用输出	QSPI只支持主机模式，且不支持多主模式
QSPI_IO2	推挽复用输出	
QSPI_IO1	推挽复用输出	
QSPI_IO0	推挽复用输出	
QSPI_CLK	推挽复用输出	
QSPI_NSS	推挽复用输出	

表 6-45 XFMC

XFMC引脚	GPIO配置
XFMC_A[23:0]	推挽复用输出
XFMC_D[15:0]	推挽复用输出
XFMC_CLK	推挽复用输出
XFMC_NOE	推挽复用输出
XFMC_NWE	推挽复用输出
XFMC_NE[2:1] XFMC_NCE[3:2]	推挽复用输出
XFMC_NWAIT	浮空输入或带上拉输入
XFMC_NADV	推挽复用输出
XFMC_NBL[1:0]	推挽复用输出

表 6-46 其他

引脚	复用功能	GPIO配置
TAMPER-RTC	RTC输出	当配置BKP_CR和BKP_RTCCR寄存器时，由硬件强制设置
	侵入事件输入	
MCO	时钟输出	推挽复用输出
EXTI输入线	外部中断输入	浮空输入或带上拉输入或带下拉输入

6.2.7 GPIO 锁定机制

锁定机制用于冻结 IO 配置以防止被意外更改。当在一个端口位上执行了锁定（LOCK）程序，在下一次复位之前，不能再更改端口的配置，参考端口配置锁定寄存器 GPIOx_PLOCK_CFG。

- PLOCKK_CFG 即 GPIOx_PLOCK_CFG[16], 只有在对 PLOCKK_CFG 按照正确的序列 w1->w0->w1->r0（此处 r0 必须有）操作之后，才会变为 1；之后只有在进行系统复位才会变为 0。
- PLOCK_CFGy 即 GPIOx_PLOCK_CFG [15:0]，只有在 PLOCKK_CFG=0 也就是未锁定的时候才能进行修改。
- PLOCKK_CFG 只有在和非 0 的 GPIOx_PLOCK_CFG [15:0]同时写入的情况下, 序列 w1->w0->w1->r0 才会有效；序列写入的过程中，GPIOx_PLOCK_CFG [15:0]必须不能改变；
- 只要 PLOCKK_CFG=0，GPIOx_PH_CFG/GPIOx_PL_CFG 位都能修改，不受 GPIOx_PLOCK_CFG [15:0] 配置的影响。
- PLOCKK_CFG=1， GPIOx_PH_CFG/GPIOx_PL_CFG 受 GPIOx_PLOCK_CFG [15:0] 的控制， 对应 PLOCK_CFGy（y = 0…15）=1，为锁定配置，不可修改，PLOCK_CFGy=0，可以修改。
- 假如序列操作错误，必须重新进行 w1->w0->w1->r0 才能再次发起锁定操作。

6.3 GPIO 寄存器

必须以 32 位字的方式操作这些外设寄存器。

6.3.1 GPIO 寄存器地址映像

GPIOA 基地址：0x40010800
GPIOB 基地址：0x40010C00
GPIOC 基地址：0x40011000
GPIOD 基地址：0x40011400
GPIOE 基地址：0x40011800
GPIOF 基地址：0x40011C00
GPIOG 基地址：0x40012000

表 6-47 GPIO 寄存器地址映像和复位值

Offset	Register		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	GPIOx_PL_CFG		PCFG7[1:0]		PMODE7[1:0]		PCFG6[1:0]		PMODE6[1:0]		PCFG5[1:0]		PMODE5[1:0]		PCFG4[1:0]		PMODE4[1:0]		PCFG3[1:0]		PMODE3[1:0]		PCFG2[1:0]		PMODE2[1:0]		PCFG1[1:0]		PMODE1[1:0]		PCFG0[1:0]		PMODE0[1:0]	
	Reset Value	x=B	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
		x=A, C, D, E, F, G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	GPIOx_PH_CFG		PCFG15[1:0]		PMODE15[1:0]		PCFG14[1:0]		PMODE14[1:0]		PCFG13[1:0]		PMODE13[1:0]		PCFG12[1:0]		PMODE12[1:0]		PCFG11[1:0]		PMODE11[1:0]		PCFG10[1:0]		PMODE10[1:0]		PCFG9[1:0]		PMODE9[1:0]		PCFG8[1:0]		PMODE8[1:0]	
	Reset Value	x=A	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		x=B, C, D, E, F, G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	GPIOx_PID		Reserved																P1D15	P1D14	P1D13	P1D12	P1D11	P1D10	P1D9	P1D8	P1D7	P1D6	P1D5	P1D4	P1D3	P1D2	P1D1	P1D0
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	GPIOx_POD		Reserved																POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
	Reset Value	x=A																	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
		x=B																	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
		x=C, D, E, F, G																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	GPIOx_PBSC		PBC15	PBC14	PBC13	PBC12	PBC11	PBC10	PBC9	PBC8	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0	PBS15	PBS14	PBS13	PBS12	PBS11	PBS10	PBS9	PBS8	PBS7	PBS6	PBS5	PBS4	PBS3	PBS2	PBS1	PBS0
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	GPIOx_PBC		Reserved																PBC15	PBC14	PBC13	PBC12	PBC11	PBC10	PBC9	PBC8	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	GPIOx_PLOCK_CFG		Reserved															PLOCKK_CFG	PLOCK_CFG15	PLOCK_CFG14	PLOCK_CFG13	PLOCK_CFG12	PLOCK_CFG11	PLOCK_CFG10	PLOCK_CFG9	PLOCK_CFG8	PLOCK_CFG7	PLOCK_CFG6	PLOCK_CFG5	PLOCK_CFG4	PLOCK_CFG3	PLOCK_CFG2	PLOCK_CFG1	PLOCK_CFG0
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	Reserved																																	
020h	GPIOx_DS_CFG		Reserved																DS_CFG15	DS_CFG 14	DS_CFG13	DS_CFG12	DS_CFG11	DS_CFG10	DS_CFG9	DS_CFG8	DS_CFG7	DS_CFG6	DS_CFG5	DS_CFG4	DS_CFG3	DS_CFG2	DS_CFG1	DS_CFG0
	Reset Value	x=F																	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1
		x=G																	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1
		x=A, B, C, D, E																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Offset	Register		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
024h	GPIOx_SR_CFG		Reserved																SR_CFG15	SR_CFG14	SR_CFG13	SR_CFG12	SR_CFG11	SR_CFG10	SR_CFG9	SR_CFG8	SR_CFG7	SR_CFG6	SR_CFG5	SR_CFG4	SR_CFG3	SR_CFG2	SR_CFG1	SR_CFG0
	Reset Value	x=F																	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1
		x=G																	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1
		x=A, B, C, D, E																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.3.2 GPIO 端口低配置寄存器（GPIOx_PL_CFG）

偏移地址：0x00

复位值：0x0000 0000（x=A,C,D,E,F,G）；0x0008 0800（x=B）

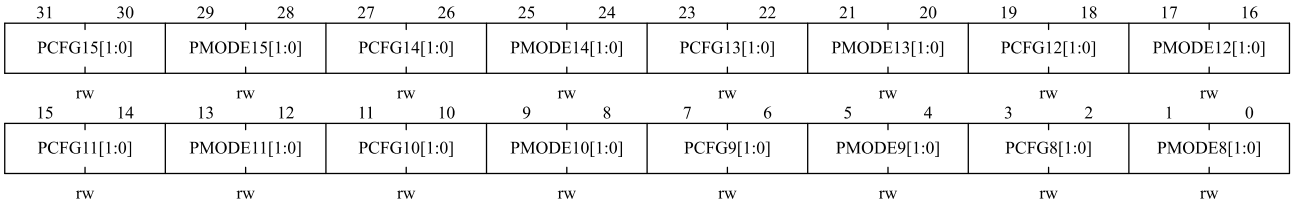
31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
PCFG7[1:0]	PMODE7[1:0]	PCFG6[1:0]	PMODE6[1:0]	PCFG5[1:0]	PMODE5[1:0]	PCFG4[1:0]	PMODE4[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PCFG3[1:0]	PMODE3[1:0]	PCFG2[1:0]	PMODE2[1:0]	PCFG1[1:0]	PMODE1[1:0]	PCFG0[1:0]	PMODE0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:30 27:26 23:22 19:18 15:14 11:10 7:6 3:2	PCFGy[1:0]	端口 x 配置位（y = 0…7） PMODE[1:0]=00 输入模式时： 00：模拟功能模式（复位后的状态） 01：浮空输入模式 10：上拉/下拉输入模式 11：保留 PMODE[1:0]>00 输出模式时： 00：通用推挽输出模式 01：通用开漏输出模式 10：复用功能推挽输出模式 11：复用功能开漏输出模式
29:28 25:24 21:20 17:16 13:12 9:8 5:4 1:0	PMODEy[1:0]	端口 x 的模式位（y = 0…7） 00：输入模式（复位后的状态） 01：输出模式，最大速度 2MHz 10：输出模式，最大速度 10MHz 11：输出模式，最大速度 50MHz

6.3.3 GPIO 端口高配置寄存器（GPIOx_PH_CFG）

偏移地址：0x04

复位值：0x8880 0000（x=A）；0x0000 0000（x=B,C,D,E,F,G）

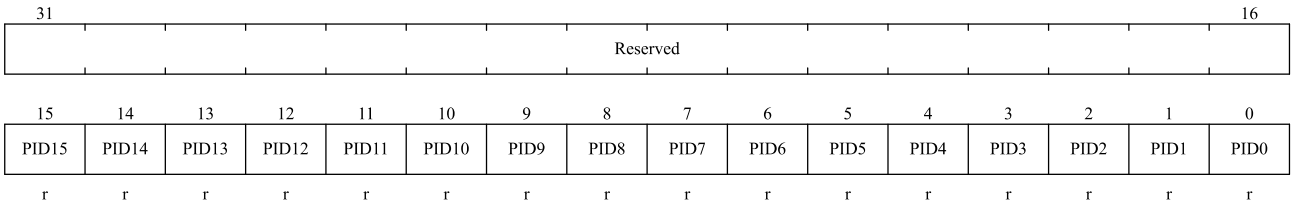


位域	名称	描述
31:30 27:26 23:22 19:18 15:14 11:10 7:6 3:2	PCFGy[1:0]	端口 x 配置位（y = 8…15） PMODE[1:0]=00 输入模式时： 00：模拟功能模式 01：浮空输入模式（复位后的状态） 10：上拉/下拉输入模式 11：保留 MODE[1:0]>00 输出模式时： 00：通用推挽输出模式 01：通用开漏输出模式 10：复用功能推挽输出模式 11：复用功能开漏输出模式
29:28 25:24 21:20 17:16 13:12 9:8 5:4 1:0	PMODEy[1:0]	端口 x 的模式位（y = 8…15） 00：输入模式（复位后的状态） 01：输出模式，最大速度 2MHz 10：输出模式，最大速度 10MHz 11：输出模式，最大速度 50MHz

6.3.4 GPIO 端口输入数据寄存器（GPIOx_PID）

偏移地址：0x08

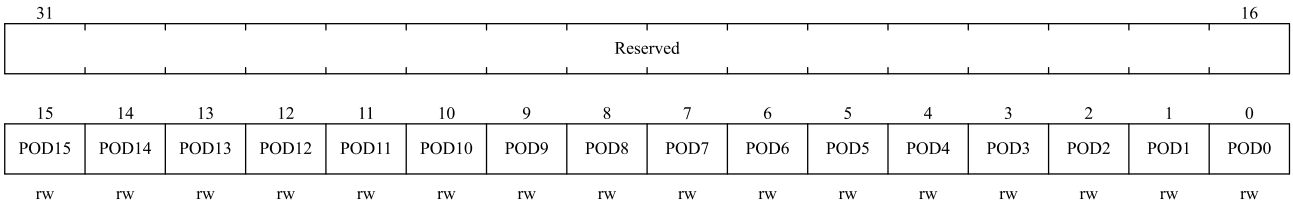
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	PIDy	端口输入数据（y = 0…15） 这些位为只读并只能以 16 位字的形式读出，读出的值为对应 I/O 口的状态。

6.3.5 GPIO 端口输出数据寄存器（GPIOx_POD）

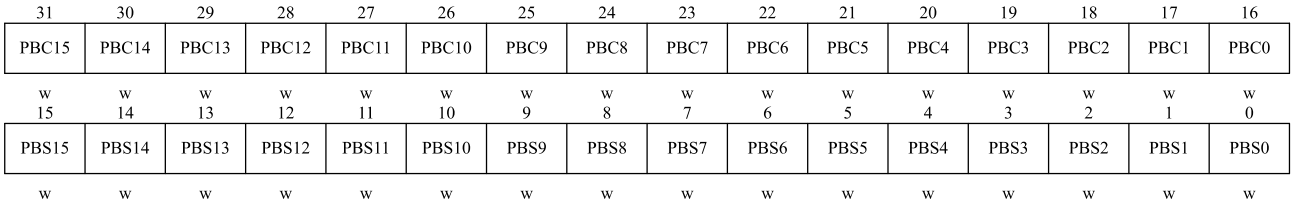
偏移地址：0x0C
复位值： 0x0000 A000（x=A）； 0x0000 0010（x=B）； 0x0000 0000（x=C,D,E,F,G）



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	PODy	端口输出数据（y = 0…15） 这些位只能以 16 位字的形式读或写操作。对 GPIOx_PBSC（x = A…E），可以对相应的 POD 位进行独立的设置/清除。

6.3.6 GPIO 端口位设置/清除寄存器（GPIOx_PBSC）

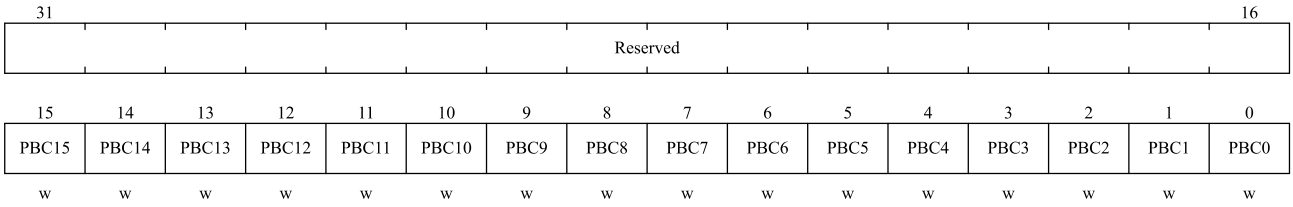
偏移地址：0x10
复位值：0x0000 0000



位域	名称	描述
31:16	PBCy	清除端口 GPIOx 的位 y （y = 0…15） 这些位只能写入并只能以字（16 位）的形式操作。 0：对相应的 PODy 位不产生影响 1：清除对应的 PODy 位为 0 *注：如果同时设置了 PBSy 和 PBCy 的对应位，PBSy 位起作用。*
15:0	PBSy	设置端口 GPIOx 的位 y （y = 0…15） 这些位只能写入并只能以字（16 位）的形式操作。 0：对相应的 PODy 位不产生影响 1：设置对应的 PODy 位为 1

6.3.7 GPIO 端口位清除寄存器（GPIOx_PBC）

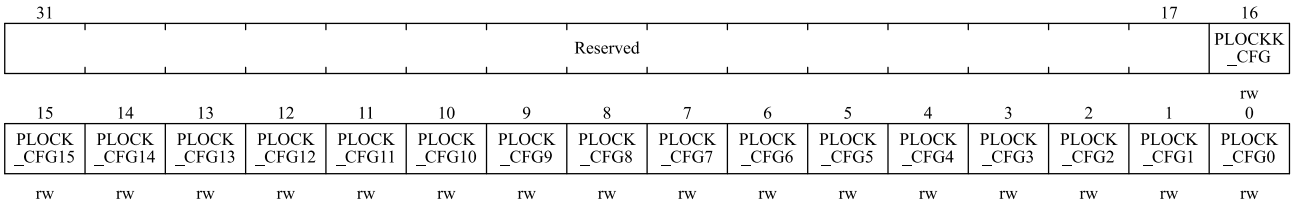
偏移地址：0x14
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	PBCy	清除端口 GPIOx 的位 y （y = 0…15） 这些位只能写入并只能以字（16 位）的形式操作。 0：对相应的 PODy 位不产生影响 1：清除对应的 PODy 位为 0

6.3.8 GPIO 端口锁定配置寄存器（GPIOx_PLOCK_CFG）

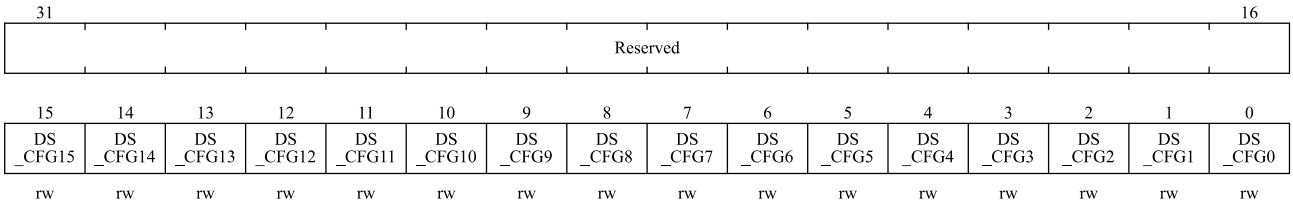
偏移地址：0x18
复位值：0x0000 0000



位域	名称	描述
31:17	Reserved	保留，硬件强制为 0。
16	PLOCKK_CFG	锁键。该位可随时读出，它只可通过锁键写入序列修改。 0：端口配置锁键位未激活 1：端口配置锁键位被激活，下次系统复位前 GPIOx_PLOCK_CFG 寄存器被锁住。锁键的写入序列： 写 1 -> 写 0 -> 写 1 -> 读 0 -> 读 1 最后一个读可省略，但可以用来确认锁键已被激活。 *注：在操作锁键的写入序列时，不能改变 PLOCK_CFG[15:0]的值。操作锁键写入序列中的任何错误将不能激活锁键。*
15:0	PLOCK_CFGy	端口 GPIOx 的配置锁定位 y （y = 0…15） 这些位可读可写但只能在 PLOCKK_CFG 位为 0 时写入。 0：不锁定端口的配置 1：锁定端口的配置

6.3.9 GPIO 驱动能力配置寄存器（GPIOx_DS_CFG）

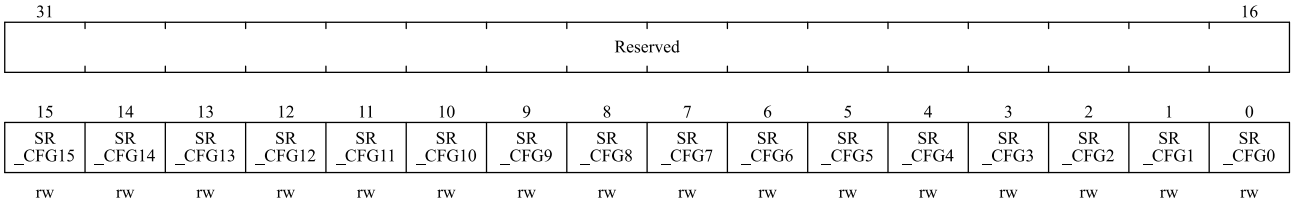
偏移地址：0x20
复位值：0x0000 FFFF（x = A,B,C,D,E）；0x0000 F03F（x=F）；0x0000 023F（x=G）



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	DS_CFGy	端口 GPIOx 的驱动能力配置位 y （y = 0…15） 这些位只能以 16 位字的形式读或写操作。 0：2mA 1：由 GPIOx_PH_CFG/GPIOx_PL_CFG 的 PMODEy 控制。 PMODE:00/01 时，4mA； PMODEy:10 时，8mA； PMODEy:11 时，12mA。

6.3.10 GPIO 翻转率配置寄存器（GPIOx_SR_CFG）

偏移地址：0x24
复位值：0x0000 FFFF（x= A,B,C,D,E）；0x0000 F03F（x=F）；0x0000 023F（x=G）



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	SR_CFGy	端口 GPIOx 的翻转率配置位 y （y = 0…15） 这些位只能以 16 位字的形式读或写操作。 0：快速翻转 1：慢速翻转

6.4 AFIO 寄存器

6.4.1 AFIO 寄存器地址映像

AFIO 基地址：0x40010000

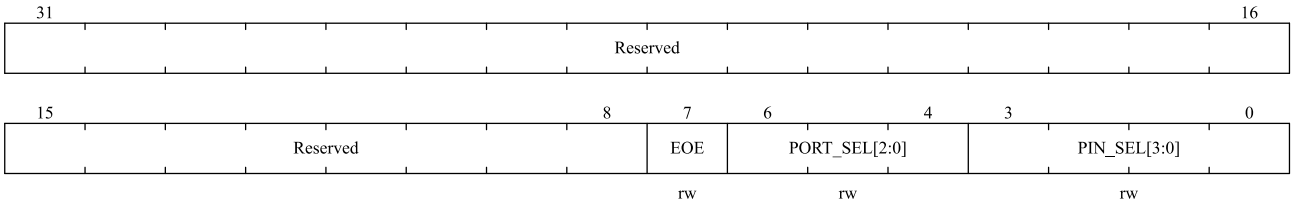
表 6-48 AFIO 寄存器地址映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	AFIO_ECTRL	Reserved																								EOE	PORT_SEL[2:0]			PIN_SEL[3:0]			
	Reset Value																									0	0	0	0	0	0	0	0
004h	AFIO_RMP_CFG	Reserved					SW_JTAG_CFG[2:0]			Reserved			ADC2_ETRR	ADC2_ETRI	ADC1_ETRR	ADC1_ETRI	TIM5CH4_RMP	PD01_RMP	CAN1_RMP[1:0]		TIM4_RMP	TIM3_RMP[1:0]		TIM2_RMP[1:0]		TIM1_RMP[1:0]		USART3_RMP[1:0]		USART2_RMP_0	USART1_RMP	I2C1_RMP	SPI1_RMP_0
	Reset Value						0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	AFIO_EXTI_CFG1	Reserved																EXTI3_CFG[3:0]				EXTI2_CFG[3:0]				EXTI1_CFG[3:0]				EXTI0_CFG[3:0]			
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	AFIO_EXTI_CFG2	Reserved																EXTI7_CFG[3:0]				EXTI6_CFG[3:0]				EXTI5_CFG[3:0]				EXTI4_CFG[3:0]			
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	AFIO_EXTI_CFG3	Reserved																EXTI11_CFG[3:0]				EXTI10_CFG[3:0]				EXTI9_CFG[3:0]				EXTI8_CFG[3:0]			
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	AFIO_EXTI_CFG4	Reserved																EXTI15_CFG[3:0]				EXTI14_CFG[3:0]				EXTI13_CFG[3:0]				EXTI12_CFG[3:0]			
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	AFIO_RMP_CFG2	Reserved																						XFMC_NADV	Reserved								
	Reset Value																							0									
020h	AFIO_RMP_CFG3	TIM8_RMP[1:0]		XFMC_RMP[1:0]		UART7_RMP[1:0]		UART6_RMP[1:0]		UART5_RMP[1:0]		UART4_RMP[1:0]		USART2_RMP_1	SPI1_RMP_1	Reserved		SPI3_RMP[1:0]		SPI2_RMP[1:0]		I2C4_RMP[1:0]		I2C3_RMP[1:0]		I2C2_RMP[1:0]		QSPI_RMP[1:0]		Reserved		CAN2_RMP[1:0]		Reserved
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0			0	0
024h	AFIO_RMP_CFG4	Reserved						QSPI_MISO	SPI3_NSS	SPI2_NSS	SPI1_NSS	Reserved		QSPI_XIP_EN	TSC_OUT_CTRL	ADC4_ETRR	ADC4_ETRI	ADC3_ETRR	ADC3_ETRI	Reserved	COMP7_RMP	COMP6_RMP[1:0]		COMP5_RMP[1:0]		COMP4_RMP[1:0]		COMP3_RMP[1:0]		COMP2_RMP[1:0]		COMP1_RMP[1:0]	
	Reset Value							0	0	0	0			0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
028h	AFIO_RMP_CFG5	Reserved								EMC_GB4_DETECT_EN	EMC_GB3_DETECT_EN	EMC_GB2_DETECT_EN	EMC_GB1_DETECT_EN	EMC_GBN4_DETECT_EN	EMC_GBN3_DETECT_EN	EMC_GBN2_DETECT_EN	EMC_GBN1_DETECT_EN	EMC_CLAMP4_DETECT_EN	EMC_CLAMP3_DETECT_EN	EMC_CLAMP2_DETECT_EN	EMC_CLAMP1_DETECT_EN	EMC_GB4_RST_EN	EMC_GB3_RST_EN	EMC_GB2_RST_EN	EMC_GB1_RST_EN	EMC_GBN4_RST_EN	EMC_GBN3_RST_EN	EMC_GBN2_RST_EN	EMC_GBN1_RST_EN	EMC_CLAMP4_RST_EN	EMC_CLAMP3_RST_EN	EMC_CLAMP2_RST_EN	EMC_CLAMP1_RST_EN
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.2 AFIO 事件控制寄存器（AFIO_ECTRL）

偏移地址：0x00

复位值：0x0000 0000



位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7	EOE	事件输出使能位。 当设置该位后，内核的事件输出信号将连接到由 PORT_SEL[2:0]和 PIN_SEL[3:0]选定的 I/O 口。 0：输出禁止 1：输出使能
6:4	PORT_SEL[2:0]	端口选择位 选择用于输出内核的事件输出信号的端口： 000：选择 Port A 001：选择 Port B 010：选择 Port C 011：选择 Port D 100：选择 Port E
3:0	PIN_SEL[3:0]	引脚选择位 选择用于输出内核的事件输出信号的引脚，（x=A…E） 与 PORT_SEL[2:0]选定的 I/O 对应。 0000：选择 Px0 0001：选择 Px1 0010：选择 Px2 0011：选择 Px3 0100：选择 Px4 0101：选择 Px5 0110：选择 Px6 0111：选择 Px7 1000：选择 Px8 1001：选择 Px9 1010：选择 Px10 1011：选择 Px11 1100：选择 Px12 1101：选择 Px13 1110：选择 Px14 1111：选择 Px15

6.4.3 AFIO 复用重映射配置寄存器（AFIO_RMP_CFG）

偏移地址：0x04
复位值：0x0000 0000

位域	名称	描述
31:27	Reserved	保留，硬件强制为 0。
26:24	SW_JTAG_CFG[2:0]	串行线 JTAG 配置 （Serial wire JTAG configuration） 这些位只可由软件写（读这些位，将返回未定义的数值），用于配置 SWJ 复用功能的 I/O 口。SWJ（串行线 JTAG）支持 4-wire JTAG 或 2-

位域	名称	描述
		wire JTAG 访问 RISC-V 内核的调试端口。系统复位后的默认状态是启用 SWJ，这种状态下可以通过 JTMS/JTCK 脚上的特定信号选择 4-wire JTAG 或 2-wire JTAG 模式。 000：完全 SWJ（4-wire JTAG + 2-wire JTAG）：复位状态； 001：完全 SWJ（4-wire JTAG + 2-wire JTAG）； 010：关闭 4-wire JTAG，启用 2-wire JTAG； 100：关闭 4-wire JTAG，关闭 2-wire JTAG； 其它值：无作用。
23	Reserved	保留。
22:21	Reserved	保留，硬件强制为 0。
20	ADC2_ETRR	ADC2 规则转换外部触发重映射 该位可由软件置'1'或置'0'。它控制与 ADC2 规则转换外部触发相连的触发输入。 0：ADC2 规则转换外部触发与 EXTI11 相连 1：ADC2 规则转换外部触发与 TIM8_TRGO 相连
19	ADC2_ETRI	ADC2 注入转换外部触发重映射 该位可由软件置'1'或置'0'。它控制与 ADC2 注入转换外部触发相连的触发输入。 0：ADC2 注入转换外部触发与 EXTI15 相连 1：ADC2 注入转换外部触发与 TIM8 通道 4 相连。
18	ADC1_ETRR	ADC1 规则转换外部触发重映射 该位可由软件置'1'或置'0'。它控制与 ADC1 规则转换外部触发相连的触发输入。 0：ADC1 规则转换外部触发与 EXTI11 相连 1：ADC1 规则转换外部触发与 TIM8_TRGO 相连
17	ADC1_ETRI	ADC1 注入转换外部触发重映射 该位可由软件置'1'或置'0'。它控制与 ADC1 注入转换外部触发相连的触发输入。 0：ADC1 注入转换外部触发与 EXTI15 相连 1：ADC1 注入转换外部触发与 TIM8 通道 4 相连
16	TIM5CH4_RMP	TIM5 通道 4 内部重映射 该位可由软件置'1'或置'0'。它控制 TIM5 通道 4 内部映像。 0：TIM5_CH4 与 PA3 相连 1：LSI 内部振荡器与 TIM5_CH4 相连，目的是对 LSI 进行校准
15	PD01_RMP	端口 D0/端口 D1 映像到 OSC_IN/OSC_OUT （Port D0/Port D1 mapping on OSC_IN/OSC_OUT） 该位可由软件置'1'或置'0'。它控制 PD0 和 PD1 的 GPIO 功能映像。当不使用主振荡器 HSE 时（系统运行于内部的 8MHz 阻容振荡器），PD0 和 PD1 可以映像到 OSC_IN 和 OSC_OUT 引脚。此功能只能适用于 80 以下引脚的封装（PD0 和 PD1 出现在 80 及以上引脚的封装上，不必重映像）。 0：不进行 PD0 和 PD1 的重映像；

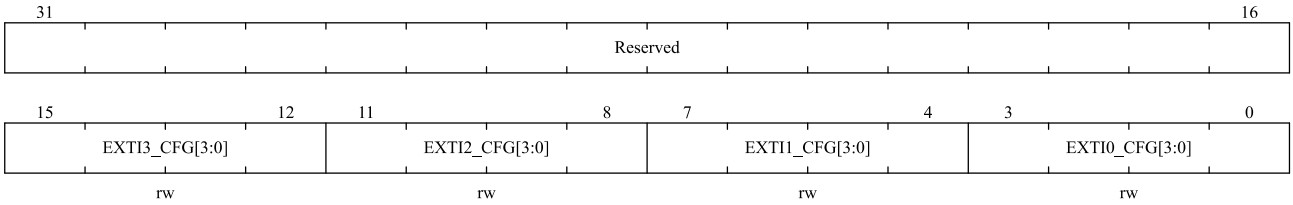
位域	名称	描述
		1：PD0 映像到 OSC_IN，PD1 映像到 OSC_OUT。
14:13	CAN1_RMP[1:0]	CAN 复用功能重映像 这些位可由软件置'1'或置'0'，在只有单个 CAN 接口的产品上控制复用功能 CAN_RX 和 CAN_TX 的重映像。 00：CAN_RX 映像到 PA11，CAN_TX 映像到 PA12； 01：CAN_RX 映像到 PD8，CAN_TX 映像到 PD9； 10：CAN_RX 映像到 PB8，CAN_TX 映像到 PB9（不能用于 36 脚的封装）； 11：CAN_RX 映像到 PD0，CAN_TX 映像到 PD1。
12	TIM4_RMP	定时器 4 的重映像 该位可由软件置'1'或置'0'，控制将 TIM4 的通道 1-4 映射到 GPIO 端口上。 0：没有重映像（TIM4_CH1/PB6，TIM4_CH2/PB7，TIM4_CH3/PB8，TIM4_CH4/PB9）； 1：完全映像（TIM4_CH1/PD12，TIM4_CH2/PD13，TIM4_CH3/PD14，TIM4_CH4/PD15）。 *注：重映像不影响在 PE0 上的 TIM4_ETR。*
11:10	TIM3_RMP[1:0]	定时器 3 的重映像 这些位可由软件置'1'或置'0'，控制定时器 3 的通道 1 至 4 在 GPIO 端口的映像。 00：没有重映像（CH1/PA6，CH2/PA7，CH3/PB0，CH4/PB1）； 01：未用组合； 10：部分映像（CH1/PB4，CH2/PB5，CH3/PB0，CH4/PB1）； 11：完全映像（CH1/PC6，CH2/PC7，CH3/PC8，CH4/PC9）。 *注：重映像不影响在 PD2 上的 TIM3_ETR。*
9:8	TIM2_RMP[1:0]	定时器 2 的重映像 这些位可由软件置'1'或置'0'，控制定时器 2 的通道 1 至 4 和外部触发（ETR）在 GPIO 端口的映像。 00：没有重映像（CH1/ETR/PA0，CH2/PA1，CH3/PA2，CH4/PA3）； 01：部分映像（CH1/ETR/PA15，CH2/PB3，CH3/PA2，CH4/PA3）； 10：部分映像（CH1/ETR/PA0，CH2/PA1，CH3/PB10，CH4/PB11）； 11：完全映像（CH1/ETR/PA15，CH2/PB3，CH3/PB10，CH4/PB11）。
7:6	TIM1_RMP[1:0]	定时器 1 的重映像 这些位可由软件置'1'或置'0'，控制定时器 1 的通道 1 至 4、1N 至 3N、外部触发（ETR）和刹车输入 （BKIN）在 GPIO 端口的映像。 00：没有重映像（ETR/PA12，CH1/PA8，CH2/PA9，CH3/PA10，CH4/PA11，BKIN/PB12， CH1N/PB13，CH2N/PB14，CH3N/PB15）； 01：部分映像（ETR/PA12，CH1/PA8，CH2/PA9，CH3/PA10，CH4/PA11，BKIN/PA6，CH1N/PA7，CH2N/PB0，CH3N/PB1）； 10：部分映像(ETR/PA12，CH1/PA8，CH2/PA9，CH3/PA10，CH4/PA11，BKIN/PB5，CH1N/PB13，CH2N/PB14，CH3N/PB15)；

位域	名称	描述
		11：完全映像（ETR/PE7，CH1/PE9，CH2/PE11，CH3/PE13，CH4/PE14，BKIN/PE15，CH1N/PE8，CH2N/PE10，CH3N/PE12）。
5:4	USART3_RMP[1:0]	USART3 的重映像 这些位可由软件置'1'或置'0'，控制 USART3 的 CTS、RTS、CK、TX 和 RX 复用功能在 GPIO 端口的映像。 00: 没有重映像（TX/PB10，RX/PB11，CK/PB12，CTS/PB13，RTS/PB14）； 01: 部分映像（TX/PC10，RX/PC11，CK/PC12，CTS/PB13，RTS/PB14）； 10: 未用组合； 11: 完全映像（TX/PD8，RX/PD9，CK/PD10，CTS/PD11，RTS/PD12）。
3	USART2_RMP_0	USART2 的重映像 这些位可由软件置'1'或置'0'， 与 USART2_RMP_1 配合使用，组成 USART2_RMP[1:0]，控制 USART2 的 CTS、RTS、CK、TX 和 RX 复用功能在 GPIO 端口的映像。 00: 没有重映像（CTS/PA0、RTS/PA1、TX/PA2，RX/PA3、CK/PA4）； 01: 重映像（CTS/PD3、RTS/PD4、TX/PD5，RX/PD6、CK/PD7）； 10: 重映像（CTS/PC6、RTS/PC7、TX/PC8，RX/PC9、CK/-）； 11: 重映像（CTS/PA15、RTS/PB3、TX/PB4，RX/PB5、CK/PA4）。 *注：10 时不支持同步模式*
2	USART1_RMP[1:0]	USART1 的重映像 该位可由软件置'1'或置'0'，控制 USART1 的 TX 和 RX 复用功能在 GPIO 端口的映像。 0: 没有重映像（TX/PA9，RX/PA10）； 1: 重映像（TX/PB6，RX/PB7）。
1	I2C1_RMP	I2C1 的重映像 该位可由软件置'1'或置'0'，控制 I2C1 的 SCL 和 SDA 复用功能在 GPIO 端口的映像。 0: 没有重映像（SCL/PB6，SDA/PB7）； 1: 重映像（SCL/PB8，SDA/PB9）。
0	SPI1_RMP_0	SPI1 的重映像 该位可由软件置'1'或置'0'， 与 SPI1_RMP_1 配合使用，组成 SPI1_RMP[1:0]，控制 SPI1 的 NSS、 SCK、 MISO 和 MOSI 复用功能在 GPIO 端口的映像。 00: 没有重映射(NSS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7)； 01: 重映射(NSS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5)； 10: 重映射(NSS/PB2, SCK/PA5, MISO/PA6, MOSI/PA7)； 11: 重映射(NSS/PB2, SCK/PE7, MISO/PE8, MOSI/PE9)；

6.4.4 AFIO 外部中断配置寄存器 1（AFIO_EXTI_CFG1）

偏移地址：0x08

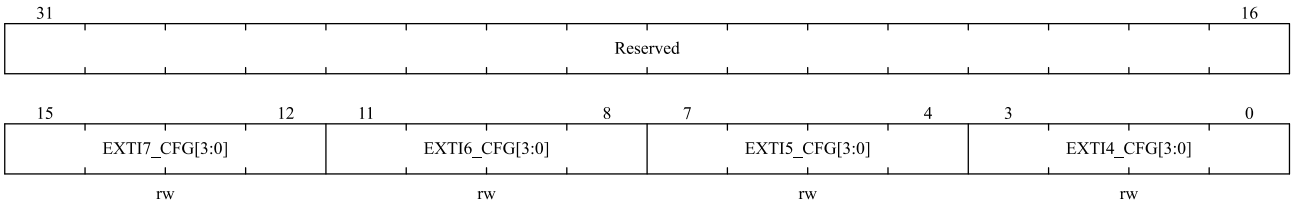
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	EXTIx_CFG[3:0]	EXTIx 配置（x = 0 … 3） 这些位可由软件读写，用于选择 EXTIx 外部中断的输入源。 0000：PAx 引脚　0001：PBx 引脚　0010：PCx 引脚 0011：PDx 引脚　0100：PEx 引脚　0101：PFx 引脚 0110：PGx 引脚

6.4.5 AFIO 外部中断配置寄存器 2（AFIO_EXTI_CFG2）

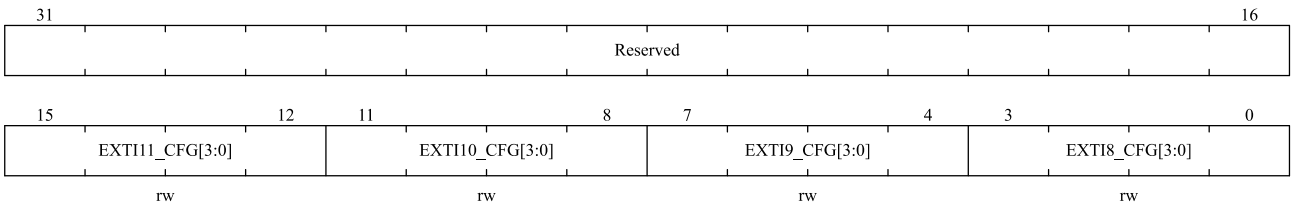
偏移地址：0x0C
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	EXTIx_CFG[3:0]	EXTIx 配置（x = 4 … 7） 这些位可由软件读写，用于选择 EXTIx 外部中断的输入源。 0000：PAx 引脚　0001：PBx 引脚　0010：PCx 引脚 0011：PDx 引脚　0100：PEx 引脚　0101：PFx 引脚 0110：PGx 引脚

6.4.6 AFIO 外部中断配置寄存器 3（AFIO_EXTI_CFG3）

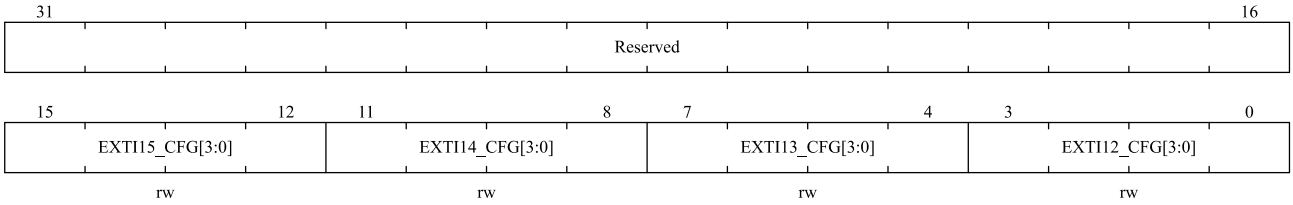
偏移地址：0x10
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	EXTIx_CFG[3:0]	EXTIx 配置（x = 8… 11） 这些位可由软件读写，用于选择 EXTIx 外部中断的输入源。 0000：PAx 引脚　0001：PBx 引脚　0010：PCx 引脚 0011：PDx 引脚　0100：PEx 引脚　0101：PFx 引脚 0110：PGx 引脚

6.4.7 AFIO 外部中断配置寄存器 4（AFIO_EXTI_CFG4）

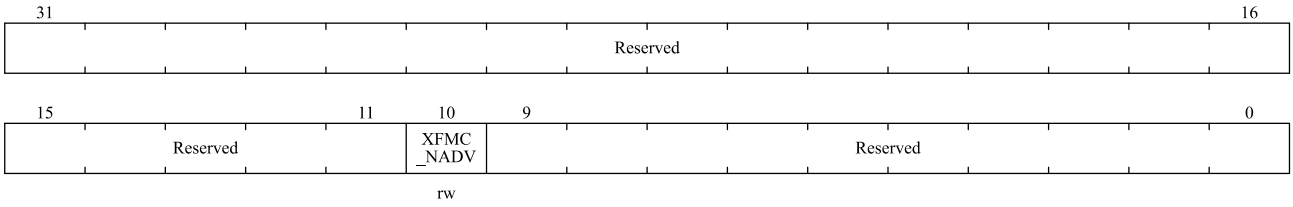
偏移地址：0x14
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	EXTIx_CFG[3:0]	EXTIx 配置（x = 12 … 15） 这些位可由软件读写，用于选择 EXTIx 外部中断的输入源。 0000：PAx 引脚　0001：PBx 引脚　0010：PCx 引脚 0011：PDx 引脚　0100：PEx 引脚　0101：PFx 引脚 0110：PGx 引脚

6.4.8 AFIO 复用重映射配置寄存器 2（AFIO_RMP_CFG2）

偏移地址：0x1C
复位值：0x0000 0000

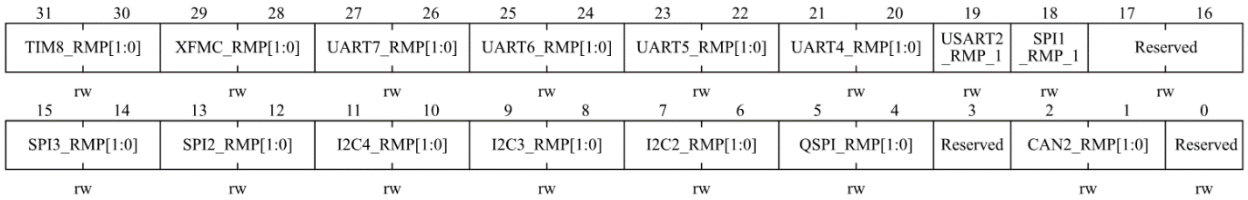


位域	名称	描述
31:11	Reserved	保留，硬件强制为 0。
10	XFMC_NADV	NADV 连接选择位。 此位可由软件读写，用于 XFMC_NADV 信号的输出。 0：NADV 信号连接到 PB7 端口输出（默认）； 1：NADV 信号未连接，PB7 端口可以用于其他外设。
9:0	Reserved	保留，硬件强制为 0。

6.4.9 AFIO 复用重映射配置寄存器 3（AFIO_RMP_CFG3）

偏移地址：0x20

复位值：0x0000 0000



位域	名称	描述
31:30	TIM8_RMP[1:0]	定时器 8 的重映像 这些位可由软件置'1'或置'0'，控制定时器 8 的通道 1 至 2 在 GPIO 端口的映像。 00：没有重映像（ETR,PA0，CH1/PC6，CH2/PC7，CH3/PC8，CH4/PC9,BKIN/PA6，CH1N/PA7，CH2N/PB0,CH3N/PB1）； 01：部分映像（ETR/PB4，CH1/PC6，CH2/PC7，CH3/PC8，CH4/PC9，BKIN/PB3，CH1N/PA15，CH2N/PC12，CH3N/PD2）； 10：未用组合； 11：部分映像（ETR/PB4，CH1/PD14，CH2/PD15，CH3/PC8，CH4/PC9，BKIN/PB3，CH1N/PA15，CH2N/PC12，CH3N/PD2）。
29:28	XFMC_RMP[1:0]	XFMC 的重映像 这些位可由软件置'1'或置'0'，控制 XFMC 在 GPIO 端口的映像。 00：A16/PD11，A17/PD12，D11/PE14，D12/PE15，NOE/PD4，NEW/PD5，NWAIT/PD6，NE1（NCE2）/PD7, NE2（NCE3）/PG9； 01：A16/PD11，A17/PD12，D11/PB10，D12/PB11，NOE/PD4，NEW/PD5，NWAIT/PD6，NE1（NCE2）/PD7, NE2（NCE3）/PG9； 10：未用组合； 11：A16/PC6，A17/PC7，D11/PB10，D12/PB11，NOE/PC9，NEW/PA10，NWAIT/PB8，NE1（NCE2）/PB9, NE2（NCE3）/PB6。
27:26	UART7_RMP[1:0]	UART7 的重映像 这些位可由软件置'1'或置'0'，控制 UART7 的 TX 和 RX 复用功能在 GPIO 端口的映像。 00: 没有重映像（TX/PC4，RX/PC5）； 01: 重映像（TX/PC2，RX/PC3）； 10: 未用组合； 11: 重映像（TX/PG0，RX/PG1）。
25:24	UART6_RMP[1:0]	UART6 的重映像 这些位可由软件置'1'或置'0'，控制 UART6 的 TX 和 RX 复用功能在 GPIO 端口的映像。 00: 没有重映像（TX/PE2，RX/PE3）； 01: 未用组合； 10: 重映像（TX/PC0，RX/PC1）； 11: 重映像（TX/PB0，RX/PB1）。

位域	名称	描述
23:22	UART5_RMP[1:0]	UART5 的重映像 这些位可由软件置'1'或置'0'，控制 UART5 的 TX 和 RX 复用功能在 GPIO 端口的映像。 00: 没有重映像（TX/PC12，RX/PD2）； 01: 重映像（TX/PB13，RX/PB14）； 10: 重映像（TX/PE8，RX/PE9）； 11: 重映像（TX/PB8，RX/PB9）。
21:20	UART4_RMP[1:0]	UART4 的重映像 这些位可由软件置'1'或置'0'，控制 UART4 的 TX 和 RX 复用功能在 GPIO 端口的映像。 00: 没有重映像（TX/PC10，RX/PC11）； 01: 重映像（TX/PB2，RX/PE7）； 10: 重映像（TX/PA13，RX/PA14； 11: 重映像（TX/PD0，RX/PD1）。
19	USART2_RMP_1	USART2 的重映像 这些位可由软件置'1'或置'0'， 与 USART2_RMP_0 配合使用，组成 USART2_RMP[1:0]，控制 USART2 的 CTS、RTS、CK、TX 和 RX 复用功能在 GPIO 端口的映像。 00: 没有重映像（CTS/PA0、RTS/PA1、TX/PA2，RX/PA3、CK/PA4）； 01: 重映像（CTS/PD3、RTS/PD4、TX/PD5，RX/PD6、CK/PD7）； 10: 重映像（CTS/PC6、RTS/PC7、TX/PC8，RX/PC9、CK/-）； 11: 重映像（CTS/PA15、RTS/PB3、TX/PB4，RX/PB5、CK/PA4）。 *注：10 时不支持同步模式*
18	SPI1_RMP_1	SPI1 的重映像 该位可由软件置'1'或置'0'， 与 SPI1_RMP_0 配合使用，组成 SPI1_RMP[1:0]，控制 SPI1 的 NSS、 SCK、 MISO 和 MOSI 复用功能在 GPIO 端口的映像。 00: 没有重映射(NSS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7)； 01: 重映射(NSS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5)； 10: 重映射(NSS/PB2, SCK/PA5, MISO/PA6, MOSI/PA7)； 11: 重映射(NSS/PB2, SCK/PE7, MISO/PE8, MOSI/PE9)；
17:16	Reserved	保留
15:14	SPI3_RMP[1:0]	SPI3 的重映像 该位可由软件置'1'或置'0'，控制 SPI3 的 NSS、SCK、MISO 和 MOSI 复用功能在 GPIO 端口的映像。 00: 没有重映像（NSS/WS/PA15，SCK/CK/PB3，MISO/PB4，MOSI/PB5）； 01: 重映像（NSS/WS/PD2，SCK/CK/PC10，MISO/PC11，MOSI/PC12）； 10: 重映像（NSS/WS/PD8，SCK/CK/PD9，MISO/PD11，MOSI/PD12）； 11: 重映像（NSS/WS/PC2，SCK/CK/PC3，MISO/PA0，MOSI/PA1）。
13:12	SPI2_RMP[1:0]	SPI2 的重映像 该位可由软件置'1'或置'0'，控制 SPI2 的 NSS、SCK、MISO 和 MOSI 复用

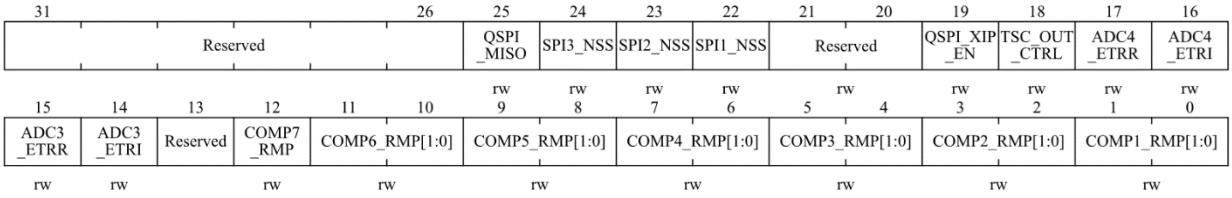
位域	名称	描述
		功能在 GPIO 端口的映像。 00: 没有重映像（NSS/WS/PB12，SCK/CK/PB13，MISO/PB14，MOSI/PB15）； 01: 重映像（NSS/WS/PC6，SCK/CK/PC7，MISO/PC8，MOSI/PC9）； 10: 未用组合； 11: 重映像（NSS/WS/PE10，SCK/CK/PE11，MISO/PE12，MOSI/PE13）。
11:10	I2C4_RMP[1:0]	I2C4 的重映像 该位可由软件置'1'或置'0'，控制 I2C4 的 SDA、SCL 复用功能在 GPIO 端口的映像。 00: 没有重映像（SCL/PC6，SDA/PC7）； 01: 重映像（SCL/PD14，SDA/PD15）； 10: 未用组合； 11: 重映像（SCL/PA9，SDA/PA10）。
9:8	I2C3_RMP[1:0]	I2C3 的重映像 该位可由软件置'1'或置'0'，控制 I2C3 的 SDA、SCL 复用功能在 GPIO 端口的映像。 00: 没有重映像（SCL/PC0，SDA/PC1）； 01: 未用组合； 10: 重映像（SCL/PF4，SDA/PF5）； 11: 重映像（SCL/PC4，SDA/PC5）。
7:6	I2C2_RMP[1:0]	I2C2 的重映像 该位可由软件置'1'或置'0'，控制 I2C2 的 SDA、SCL 复用功能在 GPIO 端口的映像。 00: 没有重映像（SCL/PB10，SDA/PB11）； 01: 重映像（SCL/PG2，SDA/PG3）； 10: 未用组合； 11: 重映像（SCL/PC4，SDA/PA5）。
5:4	QSPI_RMP[1:0]	QSPI 的重映像 该位可由软件置'1'或置'0'，控制 QSPI 的 IO0、IO1、IO2、IO3、CLK、NSS 复用功能在 GPIO 端口的映像。 00: 没有重映像（NSS/PA4，SCK/PA5，IO0/PA6，IO1/PA7，IO2/PC4，IO3/PC5）； 01: 未用组合； 10: 重映像（NSS/PF0，SCK/PF1，IO0/PF2，IO1/PF3，IO2/PF4，IO3/PF5）； 11: 重映像（NSS/PC10，SCK/PC11，IO0/PC12，IO1/PD0，IO2/PD1，IO3/PD2）。
3	Reserved	保留，硬件强制为 0。
2:1	CAN2_RMP[1:0]	CAN2 的重映像 该位可由软件置'1'或置'0'，控制 CAN2 的 CAN2_RX、CAN2_TX 复用功能在 GPIO 端口的映像。 00: 没有重映像（RX/PB12，TX/PB13）；

位域	名称	描述
		01: 重映像（RX/PB5，TX/PB6）; 10: 未用组合; 11: 重映像（RX/PD10，TX/PD11）。
0	Reserved	保留

6.4.10 AFIO 复用重映射配置寄存器 4（AFIO_RMP_CFG4）

偏移地址：0x24

复位值：0x0000 0000



位域	名称	描述
31:26	Reserved	保留，硬件强制为 0。
25	QSPI_MISO	QSPI 的 IO1 脚在标准单线模式下相当于 MISO，管脚属性可配置为: 0：由 QSPI 模块自动控制是输入还是输出 1：固定为浮空输入
24	SPI3_NSS	SPI3 的 NSS 模式位（NSS 配置为 AFIO 推挽时）。 0：空闲时 NSS 为高阻 1：空闲时 NSS 为 1
23	SPI2_NSS	SPI2 的 NSS 模式位（NSS 配置为 AFIO 推挽时）。 0：空闲时 NSS 为高阻 1：空闲时 NSS 为 1
22	SPI1_NSS	SPI1 的 NSS 模式位（NSS 配置为 AFIO 推挽时）。 0：空闲时 NSS 为高阻 1：空闲时 NSS 为 1
21:20	Reserved	保留
19	QSPI_XIP_EN	QSPI XIP 内存映射模式使能控制 0：禁止 1：使能
18	TSC_OUT_CTRL	TSC 输出控制 0：PA9 普通功能（AFIO/GPIO） 1：PA9 输出 TSC 的充电翻转信号（TSC_OUT）用于测量调试
17	ADC4_ETRR	ADC4 规则转换外部触发重映射 该位可由软件置'1'或置'0'。它控制与 ADC4 规则转换外部触发相连的触发输入。 0：ADC4 规则转换外部触发与 EXTI14 相连 1：ADC4 规则转换外部触发与 TIM5 通道 3 相连
16	ADC4_ETRI	ADC4 注入转换外部触发重映射 该位可由软件置'1'或置'0'。它控制与 ADC4 注入转换外部触发相连的触

位域	名称	描述
		发输入。 0：ADC4 注入转换外部触发与 EXTI10 相连 1：ADC4 注入转换外部触发与 TIM5 通道 3 相连。
15	ADC3_ETRR	ADC3 规则转换外部触发重映射 该位可由软件置'1'或置'0'。它控制与 ADC3 规则转换外部触发相连的触发输入。 0：ADC3 规则转换外部触发与 EXTI14 相连 1：ADC3 规则转换外部触发与 TIM5 通道 4 相连
14	ADC3_ETRI	ADC3 注入转换外部触发重映射 该位可由软件置'1'或置'0'。它控制与 ADC3 注入转换外部触发相连的触发输入。 0：ADC3 注入转换外部触发与 EXTI10 相连 1：ADC3 注入转换外部触发与 TIM5 通道 4 相连
13	Reserved	保留，硬件强制为 0。
12	COMP7_RMP	COMP7_OUT 的重映像 该位可由软件置'1'或置'0'。 0: 没有重映像（COMP7_OUT/PC2）; 1: 重映像（COMP7_OUT/PD12）。
11:10	COMP6_RMP[1:0]	COMP6_OUT 的重映像 该位可由软件置'1'或置'0'。 00: 没有重映像（COMP6_OUT/PC9）; 01: 重映像（COMP6_OUT/PA12）; 10：未用组合; 11：重映像（COMP6_OUT/PB7）。
9:8	COMP5_RMP[1:0]	COMP5_OUT 的重映像 该位可由软件置'1'或置'0'。 00: 没有重映像（COMP5_OUT/PB0）; 01: 重映像（COMP5_OUT/PB11）; 10：重映像（COMP5_OUT/PB6）; 11：重映像（COMP5_OUT/PA11）。
7:6	COMP4_RMP[1:0]	COMP4_OUT 的重映像 该位可由软件置'1'或置'0'。 00: 没有重映像（COMP4_OUT/PC5）; 01: 重映像（COMP4_OUT/PB12）; 10：未用组合; 11：重映像（COMP4_OUT/PC11）。
5:4	COMP3_RMP[1:0]	COMP3_OUT 的重映像 该位可由软件置'1'或置'0'。 00: 没有重映像（COMP3_OUT/PB10）; 01: 重映像（COMP3_OUT/PC10）; 10：未用组合; 11：重映像（COMP3_OUT/PA2）。

位域	名称	描述
3:2	COMP2_RMP[1:0]	COMP2_OUT 的重映像 该位可由软件置'1'或置'0'。 00: 没有重映像（COMP2_OUT/PA6）; 01: 重映像（COMP2_OUT/PA7）; 10：重映像（COMP2_OUT/PA12）; 11：重映像（COMP2_OUT/PB9）。
1:0	COMP1_RMP[1:0]	COMP1_OUT 的重映像 该位可由软件置'1'或置'0'。 00: 没有重映像（COMP1_OUT/PA0）; 01: 重映像（COMP1_OUT/PB1）; 10：重映像（COMP1_OUT/PA11）; 11：重映像（COMP1_OUT/PB8）。

6.4.11 AFIO 复用重映射配置寄存器 5（AFIO_RMP_CFG5）

偏移地址：0x28
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								EGB4_DET_EN	EGB3_DET_EN	EGB2_DET_EN	EGB1_DET_EN	EGBN4_DET_EN	EGBN3_DET_EN	EGBN2_DET_EN	EGBN1_DET_EN
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECLAMP4_DET_EN	ECLAMP3_DET_EN	ECLAMP2_DET_EN	ECLAMP1_DET_EN	EGB4_RST_EN	EGB3_RST_EN	EGB2_RST_EN	EGB1_RST_EN	EGBN4_RST_EN	EGBN3_RST_EN	EGBN2_RST_EN	EGBN1_RST_EN	ECLAMP4_RST_EN	ECLAMP3_RST_EN	ECLAMP2_RST_EN	ECLAMP1_RST_EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:24	Reserved	保留，硬件强制为 0。
23	EGB4_DET_EN	EMC GB4 检测使能位（地弹检测）。 0：禁止 1：使能
22	EGB3_DET_EN	EMC GB3 检测使能位。 0：禁止 1：使能
21	EGB2_DET_EN	EMC GB2 检测使能位。 0：禁止 1：使能
20	EGB1_DET_EN	EMC GB1 检测使能位。 0：禁止 1：使能
19	EGBN4_DET_EN	EMC GBN4 检测使能位。 0：禁止 1：使能
18	EGBN3_DET_EN	EMC GBN3 检测使能位。 0：禁止

位域	名称	描述
		1：使能
17	EGBN2_DET_EN	EMC GBN2 检测使能位。 0：禁止 1：使能
16	EGBN1_DET_EN	EMC GBN1 检测使能位。 0：禁止 1：使能
15	ECLAMP4_DET_EN	针对 VDD_4 的 EMC CLAMP4 检测使能位。 0：禁止 1：使能
14	ECLAMP3_DET_EN	针对 VDD_3 的 EMC CLAMP3 检测使能位。 0：禁止 1：使能
13	ECLAMP2_DET_EN	针对 VDD_2 的 EMC CLAMP2 检测使能位。 0：禁止 1：使能
12	ECLAMP1_DET_EN	针对 VDD_1 的 EMC CLAMP1 检测使能位。 0：禁止 1：使能
11	EGB4_RST_EN	当 EMC GB4 检测到时，系统复位使能位。 0：禁止 1：使能
10	EGB3_RST_EN	当 EMC GB3 检测到时，系统复位使能位。 0：禁止 1：使能
9	EGB2_RST_EN	当 EMC GB2 检测到时，系统复位使能位。 0：禁止 1：使能
8	EGB1_RST_EN	当 EMC GB1 检测到时，系统复位使能位。 0：禁止 1：使能
7	EGBN4_RST_EN	当 EMC GBN4 检测到时，系统复位使能位。 0：禁止 1：使能
6	EGBN3_RST_EN	当 EMC GBN3 检测到时，系统复位使能位。 0：禁止 1：使能
5	EGBN2_RST_EN	当 EMC GBN2 检测到时，系统复位使能位。 0：禁止 1：使能
4	EGBN1_RST_EN	当 EMC GBN1 检测到时，系统复位使能位。 0：禁止

位域	名称	描述
		1：使能
3	ECLAMP4_RST_EN	当 EMC CLAMP4 检测到时，系统复位使能位。 0：禁止 1：使能
2	ECLAMP3_RST_EN	当 EMC CLAMP3 检测到时，系统复位使能位。 0：禁止 1：使能
1	ECLAMP2_RST_EN	当 EMC CLAMP2 检测到时，系统复位使能位。 0：禁止 1：使能
0	ECLAMP1_RST_EN	当 EMC CLAMP1 检测到时，系统复位使能位。 0：禁止 1：使能

7 中断和事件

7.1 改进型内核中断控制器

特性

- 86 个可屏蔽中断通道。
- 16 个可编程的优先等级（使用了 4 位中断优先级）;
- 低延迟的异常和中断处理;
- 电源管理控制;
- 系统控制寄存器的实现;

改进型内核中断控制器（ECLIC）和处理器核的接口紧密相连，可以实现低延迟的中断处理和高效地处理晚到的中断。嵌套向量中断控制器管理着包括内核异常等中断。

详细请参考 Nuclei™ N 级别处理器内核指令架构手册获得进一步信息。

7.1.1 系统计时器

系统计时器可选 RTC 用 32KHz 时钟或系统时钟作为计数时钟，可通过配置选择时钟和配置计数值，产生固定的时间基准或记录系统上电后执行的总时长。

详细请参考 Nuclei™ N 级别处理器内核指令架构手册获得进一步信息。

7.1.2 中断和异常向量

表 7-1 向量表

向量编号	优先级类型	名称	说明	地址	中断信号类型
0	-	保留	保留	0x0000_0000	保留
1	-	保留	保留	0x0000_0004	保留
2	-	保留	保留	0x0000_0008	保留
3	可设置	软件中断	N级别处理器内核的TIMER单元生成的软件中断	0x0000_000C	高电平信号
4	-	保留	保留	0x0000_0010	保留
5	-	保留	保留	0x0000_0014	保留
6	-	保留	保留	0x0000_0018	保留
7	可设置	计时器中断	N级别处理器内核的TIMER单元生成的计时器中断	0x0000_001C	高电平信号
8	-	保留	保留	0x0000_0020	保留
9	-	保留	保留	0x0000_0024	保留
10	-	保留	保留	0x0000_0028	保留
11	-	保留	保留	0x0000_002C	保留

向量编号	优先级类型	名称	说明	地址	中断信号类型
12	-	保留	保留	0x0000_0030	保留
13	-	保留	保留	0x0000_0034	保留
14	-	保留	保留	0x0000_0038	保留
15	-	保留	保留	0x0000_003C	保留
16	-	保留	保留	0x0000_0040	保留
17	-	保留	保留	0x0000_0044	保留
18	-	保留	保留	0x0000_0048	保留
19	可设置	WWDG	窗口定时器中断	0x0000_004C	高电平信号
20	可设置	PVD	连到EXTI线16的电源电压检测（PVD）中断	0x0000_0050	高电平信号
21	可设置	TAMPER	侵入检测中断	0x0000_0054	高脉冲信号
22	可设置	RTC_WKUP	连接到EXTI线20的实时时钟（RTC）唤醒中断	0x0000_0058	高电平信号
23	可设置	FLASH	闪存全局中断	0x0000_005C	高电平信号
24	可设置	RCC	复位和时钟控制（RCC）中断	0x0000_0060	高电平信号
25	可设置	EXTI0	EXTI线0中断	0x0000_0064	高电平信号
26	可设置	EXTI1	EXTI线1中断	0x0000_0068	高电平信号
27	可设置	EXTI2	EXTI线2中断	0x0000_006C	高电平信号
28	可设置	EXTI3	EXTI线3中断	0x0000_0070	高电平信号
29	可设置	EXTI4	EXTI线4中断	0x0000_0074	高电平信号
30	可设置	DMA1通道1	DMA1通道1全局中断	0x0000_0078	高电平信号
31	可设置	DMA1通道2	DMA1通道2全局中断	0x0000_007C	高电平信号
32	可设置	DMA1通道3	DMA1通道3全局中断	0x0000_0080	高电平信号
33	可设置	DMA1通道4	DMA1通道4全局中断	0x0000_0084	高电平信号
34	可设置	DMA1通道5	DMA1通道5全局中断	0x0000_0088	高电平信号
35	可设置	DMA1通道6	DMA1通道6全局中断	0x0000_008C	高电平信号
36	可设置	DMA1通道7	DMA1通道7全局中断	0x0000_0090	高电平信号
37	可设置	ADC1_2	ADC1和ADC2全局中断	0x0000_0094	高脉冲信号
38	-	保留	保留	0x0000_0098	保留
39	-	保留	保留	0x0000_009C	保留
40	可设置	CAN1_RX1	CAN1接收1中断	0x0000_00A0	高电平信号
41	可设置	CAN1_SCE	CAN1 SCE中断	0x0000_00A4	高电平信号
42	可设置	EXTI9_5	EXTI线[9:5]中断	0x0000_00A8	高电平信号
43	可设置	TIM1_BRK	TIM1刹车中断	0x0000_00AC	高电平信号
44	可设置	TIM1_UP	TIM1更新中断	0x0000_00B0	高电平信号
45	可设置	TIM1_TRG_COM	TIM1触发和通信中断	0x0000_00B4	高电平信号
46	可设置	TIM1_CC	TIM1捕获比较中断	0x0000_00B8	高电平信号
47	可设置	TIM2	TIM2全局中断	0x0000_00BC	高电平信号
48	可设置	TIM3	TIM3全局中断	0x0000_00C0	高电平信号
49	可设置	TIM4	TIM4全局中断	0x0000_00C4	高电平信号

向量编号	优先级类型	名称	说明	地址	中断信号类型
50	可设置	I2C1_EV	I2C1事件中断	0x0000_00C8	高电平信号
51	可设置	I2C1_ER	I2C1错误中断	0x0000_00CC	高电平信号
52	可设置	I2C2_EV	I2C2事件中断	0x0000_00D0	高电平信号
53	可设置	I2C2_ER	I2C2错误中断	0x0000_00D4	高电平信号
54	可设置	SPI1	SPI1全局中断	0x0000_00D8	高电平信号
55	可设置	SPI2_I2S2	SPI2/I2S2全局中断	0x0000_00DC	高电平信号
56	可设置	USART1	USART1全局中断	0x0000_00E0	高电平信号
57	可设置	USART2	USART2全局中断	0x0000_00E4	高电平信号
58	可设置	USART3	USART3全局中断	0x0000_00E8	高电平信号
59	可设置	EXTI15_10	EXTI线[15:10]中断	0x0000_00EC	高电平信号
60	可设置	RTCAlarm	连接到EXTI线17的RTC闹钟中断	0x0000_00F0	高电平信号
61	-	保留	保留	0x0000_00F4	保留
62	可设置	TIM8_BRK	TIM8刹车中断	0x0000_00F8	高电平信号
63	可设置	TIM8_UP	TIM8更新中断	0x0000_00FC	高电平信号
64	可设置	TIM8_TRG_COM	TIM8触发和通信中断	0x0000_0100	高电平信号
65	可设置	TIM8_CC	TIM8捕获比较中断	0x0000_0104	高电平信号
66	可设置	ADC3_4	ADC3和ADC4全局中断	0x0000_0108	高脉冲信号
67	可设置	XFMC	XFMC全局中断	0x0000_010C	高电平信号
68	-	保留	保留	0x0000_0110	保留
69	可设置	TIM5	TIM5全局中断	0x0000_0114	高电平信号
70	可设置	SPI3_I2S3	SPI3/I2S3全局中断	0x0000_0118	高电平信号
71	可设置	UART4	UART4全局中断	0x0000_011C	高电平信号
72	可设置	UART5	UART5全局中断	0x0000_0120	高电平信号
73	可设置	TIM6	TIM6全局中断	0x0000_0124	高电平信号
74	可设置	TIM7	TIM7全局中断	0x0000_0128	高电平信号
75	可设置	DMA2通道1	DMA2通道1全局中断	0x0000_012C	高电平信号
76	可设置	DMA2通道2	DMA2通道2全局中断	0x0000_0130	高电平信号
77	可设置	DMA2通道3	DMA2通道3全局中断	0x0000_0134	高电平信号
78	可设置	DMA2通道4	DMA2通道4全局中断	0x0000_0138	高电平信号
79	可设置	DMA2通道5	DMA2通道5全局中断	0x0000_013C	高电平信号
80	-	保留	保留	0x0000_0140	保留
81	-	保留	保留	0x0000_0144	保留
82	可设置	CAN2_TX	CAN2发送中断	0x0000_0148	高电平信号
83	可设置	CAN2_RX0	CAN2接收0中断	0x0000_014C	高电平信号
84	可设置	CAN2_RX1	CAN2接收1中断	0x0000_0150	高电平信号
85	可设置	CAN2_SCE	CAN2的SCE中断	0x0000_0154	高电平信号
86	可设置	QSPI	QSPI全局中断	0x0000_0158	高电平信号
87	可设置	DMA2通道6	DMA2通道6全局中断	0x0000_015C	高电平信号
88	可设置	DMA2通道7	DMA2通道7全局中断	0x0000_0160	高电平信号

向量编号	优先级类型	名称	说明	地址	中断信号类型
89	可设置	I2C3_EV	I2C3事件中断	0x0000_0164	高电平信号
90	可设置	I2C3_ER	I2C3错误中断	0x0000_0168	高电平信号
91	可设置	I2C4_EV	I2C4事件中断	0x0000_016C	高电平信号
92	可设置	I2C4_ER	I2C4错误中断	0x0000_0170	高电平信号
93	可设置	UART6	UART6全局中断	0x0000_0174	高电平信号
94	可设置	UART7	UART7全局中断	0x0000_0178	高电平信号
95	可设置	DMA1通道8	DMA1通道8全局中断	0x0000_017C	高电平信号
96	可设置	DMA2通道8	DMA2通道8全局中断	0x0000_0180	高电平信号
97	-	保留	保留	0x0000_0184	保留
98	可设置	SAC	SAC全局中断	0x0000_0188	高电平信号
99	可设置	MMU	MMU全局中断	0x0000_018C	高脉冲信号
100	可设置	TSC	连接到EXTI线21的TSC全局中断	0x0000_0190	保留
101	可设置	COMP_1_2_3	COMP1、COMP2、COMP3全局中断	0x0000_0194	高脉冲信号
102	可设置	COMP_4_5_6	COMP4、COMP5、COMP6全局中断	0x0000_0198	高脉冲信号
103	可设置	COMP7	COMP7全局中断	0x0000_019C	高脉冲信号
104	可设置	R-SRAM	R-SRAM错误中断	0x0000_01A0	高电平信号

7.2 外部中断/事件控制器（EXTI）

7.2.1 EXTI 简介

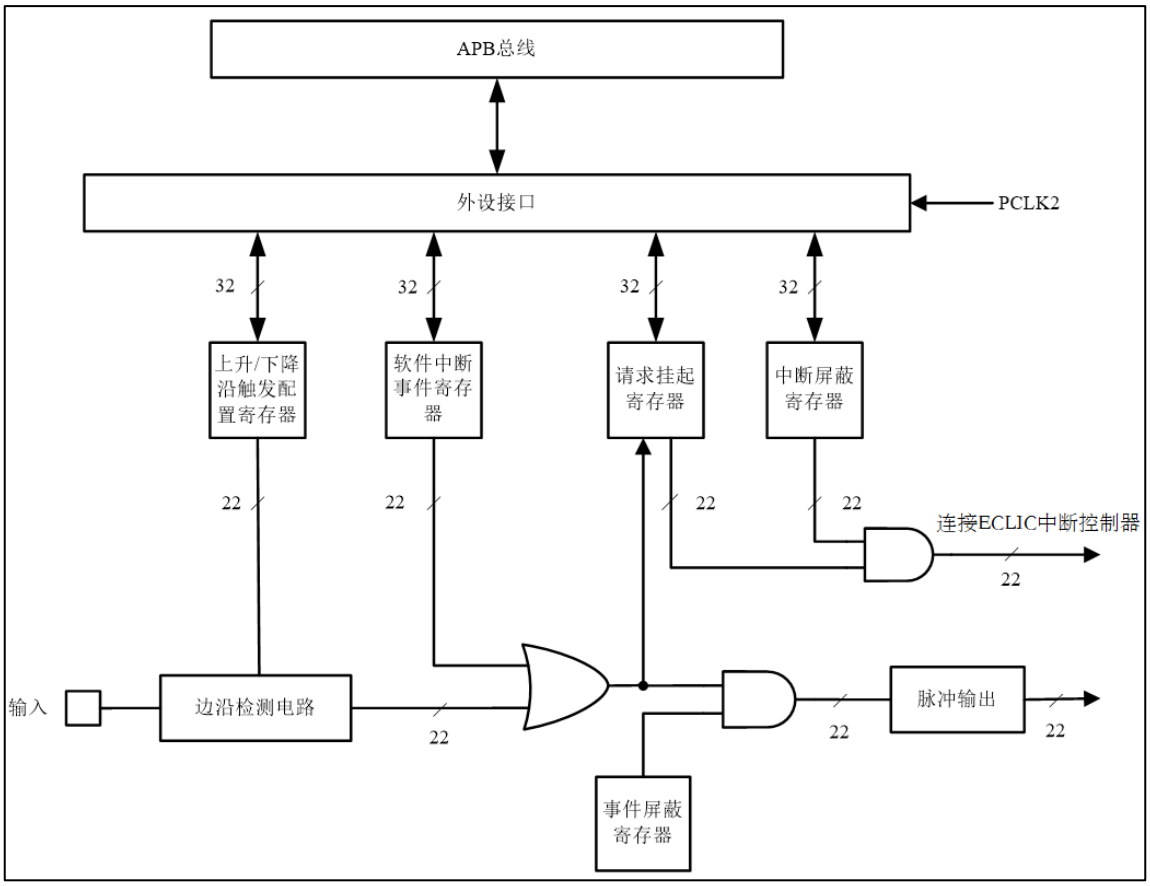
外部中断/事件控制器包含 22 个产生中断/事件触发的边沿检测电路，每条输入线可以独立地配置脉冲或挂起输入类型，以及上升沿、下降沿或者双边沿 3 种触发事件类型，也可以独立地被屏蔽。挂起寄存器保持着状态线的中断请求，可通过在挂起寄存器的对应位写'1'操作，清除中断请求。

7.2.2 EXTI 主要特性

EXTI 控制器的主要特性如下:

- 支持 22 个软件中断/事件请求
- 每条输入线对应的中断/事件都能独立配置触发或屏蔽
- 每条中断线都有独立的状态位
- 支持脉冲或挂起输入类型
- 支持上升沿、下降沿或双边沿 3 种触发事件类型
- 可唤醒退出省电模式

图 7-1 外部中断/事件控制器框图



7.2.3 功能描述

EXTI 包含 22 条中断线，其中 16 条来自 I/O 管脚，另 6 条来自内部模块。要产生中断，必须配置外部中断控制器的 ECLIC 中断通道使能相应的中断线。通过沿触发配置寄存器 EXTI_RT_CFG 和 EXTI_FT_CFG 选择上升沿、下降沿或双边沿触发事件类型，并将中断屏蔽寄存器 EXTI_IMASK 的相应位写'1'开放允许中断请求。当外部中断线上检测到预设的边沿触发极性，将产生一个中断请求，对应的挂起位也随之被置'1'。在挂起寄存器的对应位写'1'，将清除该中断请求。

要产生事件，必须配置并使能对应的事件线。根据需要的边沿检测极性，设置上升/下降沿触发配置寄存器，同时在事件屏蔽寄存器的相应位写'1'允许中断请求。当事件线上发生预设的边沿时，将产生一个事件请求脉冲，对应的挂起位不被置'1'。

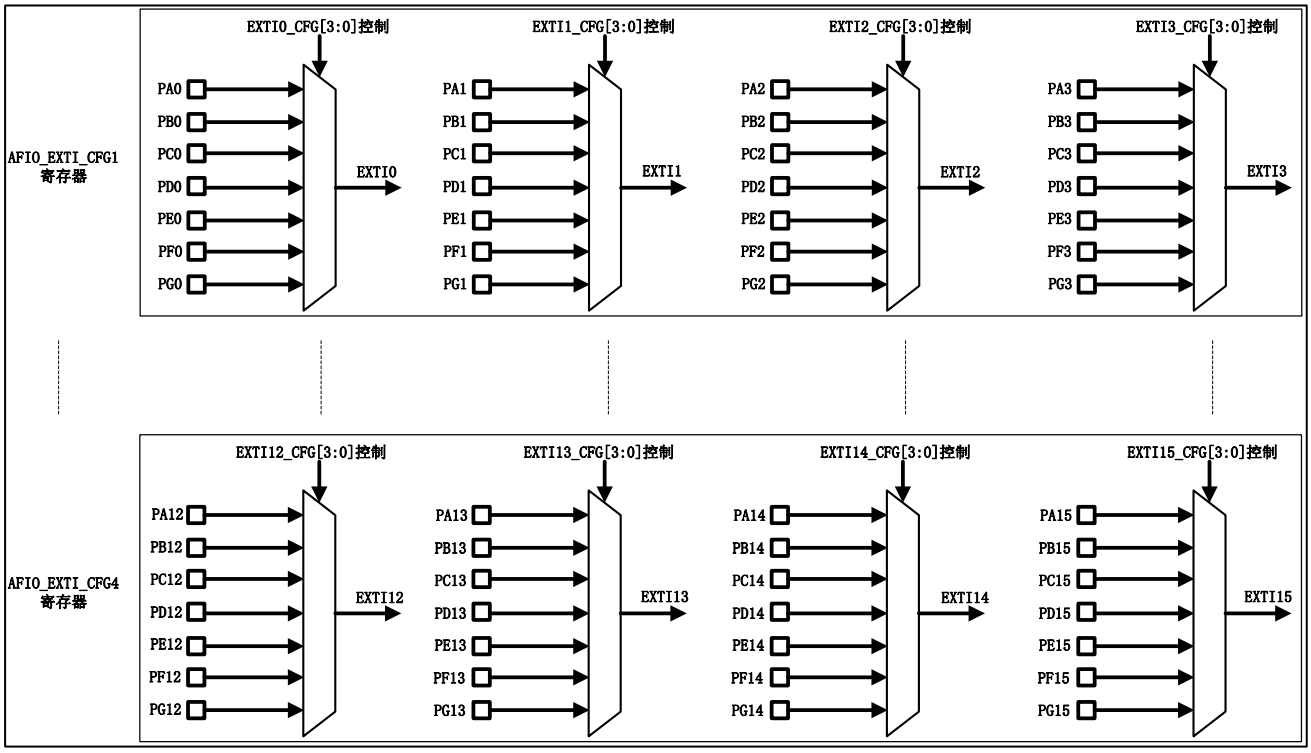
另外，通过在软件中断/事件寄存器写'1'，也可以通过软件产生中断/事件请求。

- 硬件中断配置，根据需要选择配置 22 条线路作为中断源：
  - 配置 22 条中断线的屏蔽位（EXTI_IMASK）；
  - 配置所选中断线的触发配置位（EXTI_RT_CFG 和 EXTI_FT_CFG）；
  - 配置对应到外部中断控制器的 ECLIC 中断通道的使能和屏蔽位，使 22 条中断线中的请求可以被正确地响应。

- 硬件事件配置，根据需要选择配置 22 条线路作为事件源：
  - 配置 22 条事件线的屏蔽位（EXTI_EMASK）；
  - 配置所选事件线的触发配置位（EXTI_RT_CFG 和 EXTI_FT_CFG）。
- 软件中断/事件配置，根据需要选择配置 22 条线路作为软件中断/事件线：
  - 配置 22 条中断/事件线屏蔽位（EXTI_IMASK,EXTI_EMASK）；
  - 配置软件中断事件寄存器的请求位（EXTI_SWIE）。

7.2.4 EXTI 线路映像

图 7-2 外部中断通用 I/O 映像



通过 AFIO_EXTI_CFGy 配置 GPIO 线上的外部中断/事件，必须先使能 AFIO 时钟。通用 I/O 端口以上图的方式连接到 16 条外部中断/事件线上。另外 6 条 EXTI 线的连接方式如下：

- EXTI 线 16 连接到 PVD 输出
- EXTI 线 17 连接到 RTC 闹钟事件
- EXTI 线 18 保留
- EXTI 线 19 保留
- EXTI 线 20 连接到 RTC 唤醒事件
- EXTI 线 21 连接到 TSC 全局中断

7.3 EXTI 寄存器

EXTI 基地址：0x40010400

7.3.1 EXTI 寄存器地址映像

表 7-2 EXTI 寄存器地址映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	EXTI_IMASK	Reserved										IMASK[21:20]		Reserved		IMASK[17:0]																	
	Reset Value											0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	EXTI_EMASK	Reserved										EMASK[21:20]		Reserved		EMASK[17:0]																	
	Reset Value											0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	EXTI_RT_CFG	Reserved										RT_CFG[21:20]		Reserved		RT_CFG[17:0]																	
	Reset Value											0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	EXTI_FT_CFG	Reserved										FT_CFG[21:20]		Reserved		FT_CFG[17:0]																	
	Reset Value											0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	EXTI_SWIE	Reserved										SWIE[21:20]		Reserved		SWIE[17:0]																	
	Reset Value											0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	EXTI_PEND	Reserved										PEND21	PEND20	Reserved		PEND17	PEND16	PEND15	PEND14	PEND13	PEND12	PEND11	PEND10	PEND9	PEND8	PEND7	PEND6	PEND5	PEND4	PEND3	PEND2	PEND1	PEND0
	Reset Value											0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	EXTI_TS_SEL	Reserved																												TSSEL[3:0]			
	Reset Value																													0	0	0	0

7.3.2 EXTI 中断屏蔽寄存器（EXTI_IMASK）

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										IMASK21	IMASK20	Reserved		IMASK17	IMASK16
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMASK15	IMASK14	IMASK13	IMASK12	IMASK11	IMASK10	IMASK9	IMASK8	IMASK7	IMASK6	IMASK5	IMASK4	IMASK3	IMASK2	IMASK1	IMASK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

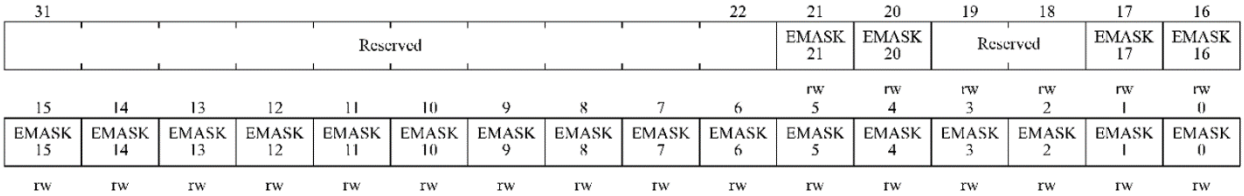
位域	名称	描述
31:22	Reserved	保留，硬件强制为 0。
21:20	IMASKx	线 x 上的中断屏蔽 0：屏蔽来自线 x 上的中断请求； 1：开放来自线 x 上的中断请求。
19:18	Reserved	保留
17:0	IMASKx	线 x 上的中断屏蔽 0：屏蔽来自线 x 上的中断请求；

位域	名称	描述
		1：开放来自线 x 上的中断请求。

7.3.3 EXTI 事件屏蔽寄存器（EXTI_EMASK）

偏移地址：0x04

复位值：0x0000 0000

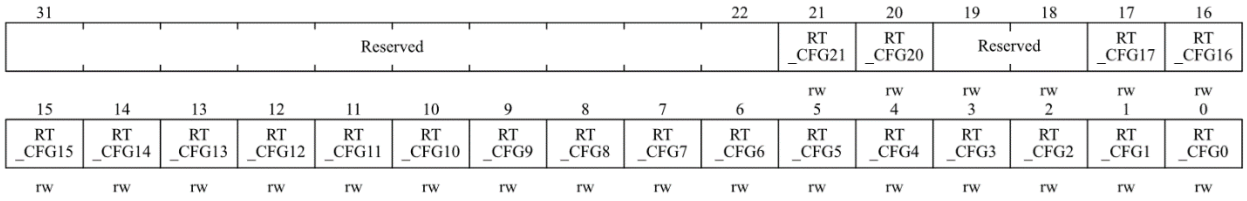


位域	名称	描述
31:22	Reserved	保留，硬件强制为 0。
21:20	EMASKx	线 x 上的事件屏蔽 0：屏蔽来自线 x 上的事件请求； 1：开放来自线 x 上的事件请求。
19:18	Reserved	保留
17:0	EMASKx	线 x 上的事件屏蔽 0：屏蔽来自线 x 上的事件请求； 1：开放来自线 x 上的事件请求。

7.3.4 EXTI 上升沿触发配置寄存器（EXTI_RT_CFG）

偏移地址：0x08

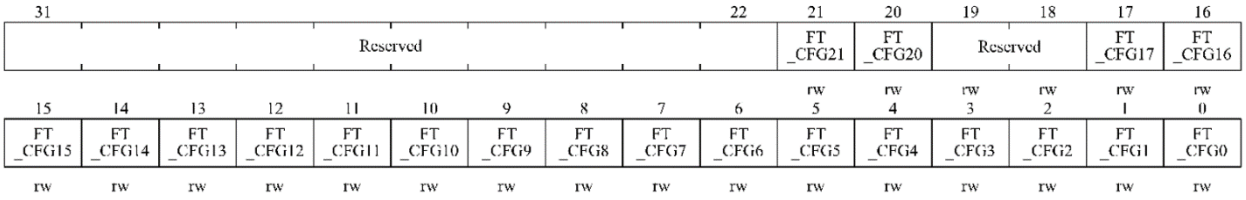
复位值：0x0000 0000



位域	名称	描述
31:22	Reserved	保留，硬件强制为 0。
21:20	RT_CFGx	线 x 上的上升沿触发配置位 0：禁止输入线 x 上的上升沿触发（中断和事件） 1：允许输入线 x 上的上升沿触发（中断和事件）
19:18	Reserved	保留
17:0	RT_CFGx	线 x 上的上升沿触发配置位 0：禁止输入线 x 上的上升沿触发（中断和事件） 1：允许输入线 x 上的上升沿触发（中断和事件）

7.3.5 EXTI 下降沿触发配置寄存器（EXTI_FT_CFG）

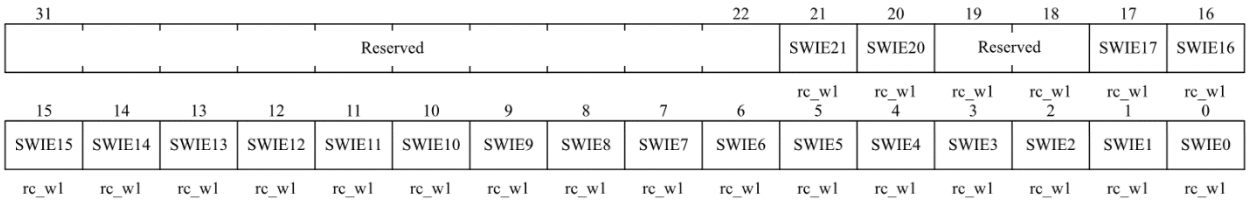
偏移地址：0x0C
复位值：0x0000 0000



位域	名称	描述
31:22	Reserved	保留，硬件强制为 0。
21:20	FT_CFGx	线 x 上的下降沿触发配置位 0：禁止输入线 x 上的下降沿触发（中断和事件） 1：允许输入线 x 上的下降沿触发（中断和事件）
19:18	Reserved	保留
17:0	FT_CFGx	线 x 上的下降沿触发配置位 0：禁止输入线 x 上的下降沿触发（中断和事件） 1：允许输入线 x 上的下降沿触发（中断和事件）

7.3.6 EXTI 软件中断事件寄存器（EXTI_SWIE）

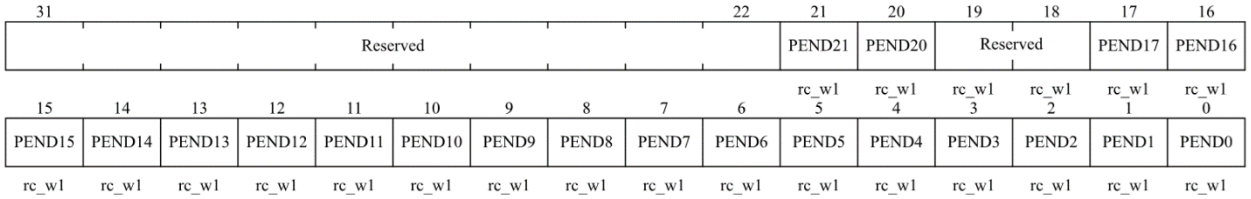
偏移地址：0x10
复位值：0x0000 0000



位域	名称	描述
31:22	Reserved	保留，硬件强制为 0。
21:20	SWIEx	线 x 上的软件中断 当该位为'0'时，写'1'将设置 EXTI_PEND 中相应的挂起位。如果在 EXTI_IMASK 和 EXTI_EMASK 中允许产生该中断，此时将产生一个中断。 *注：通过写入'1'清除 EXTI_PEND 的对应位，可以清除该位为'0'。*
19:18	Reserved	保留
17:0	SWIEx	线 x 上的软件中断 当该位为'0'时，写'1'将设置 EXTI_PEND 中相应的挂起位。如果在 EXTI_IMASK 和 EXTI_EMASK 中允许产生该中断，此时将产生一个中断。 *注：通过写入'1'清除 EXTI_PEND 的对应位，可以清除该位为'0'。*

7.3.7 EXTI 挂起寄存器（EXTI_PEND）

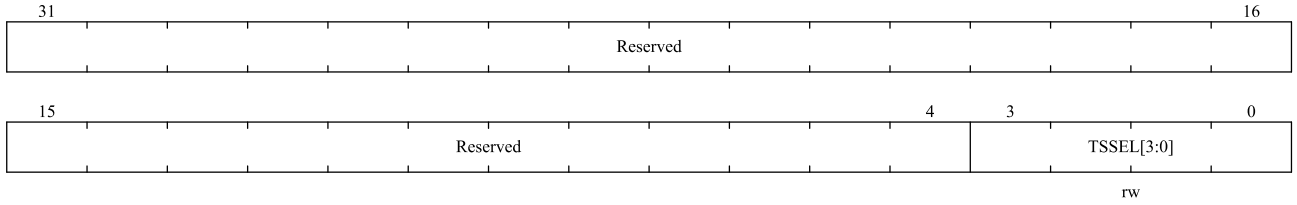
偏移地址：0x14
复位值：0x0000 0000



位域	名称	描述
31:22	Reserved	保留，硬件强制为 0。
21:20	PENDx	线 x 上的挂起位 0：没有发生挂起请求 1：发生了挂起触发请求 当外部中断线上发生了选择的边沿触发事件，该位被置'1'。在该位中写入'1'可以清除它，也可以通过改变边沿检测的极性清除此位。
19:18	Reserved	保留
17:0	PENDx	线 x 上的挂起位 0：没有发生挂起请求 1：发生了挂起触发请求 当外部中断线上发生了选择的边沿触发事件，该位被置'1'。在该位中写入'1'可以清除它，也可以通过改变边沿检测的极性清除此位。

7.3.8 EXTI 时间戳触发源选择寄存器（EXTI_TS_SEL）

偏移地址：0x18
复位值：0x0000 0000



位域	名称	描述
31:4	Reserved	保留，硬件强制为 0。
3:0	TSSEL[3:0]	选择外部中断输入作为时间戳事件的触发源 0：选择 EXTI0 作为时间戳事件的触发源； 1：选择 EXTI1 作为时间戳事件的触发源； …… 15：选择 EXTI15 作为时间戳事件的触发源。 注：时间戳功能请参考 RTC 章节相关介绍。

8 DMA 控制器

8.1 DMA 简介

直接存储器访问控制器（DMA）用于高速数据传输，提供了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通道。数据可以通过 DMA 通道快速地传输，不需要 CPU 干预，这样可以减少 CPU 的资源消耗用作其他应用。

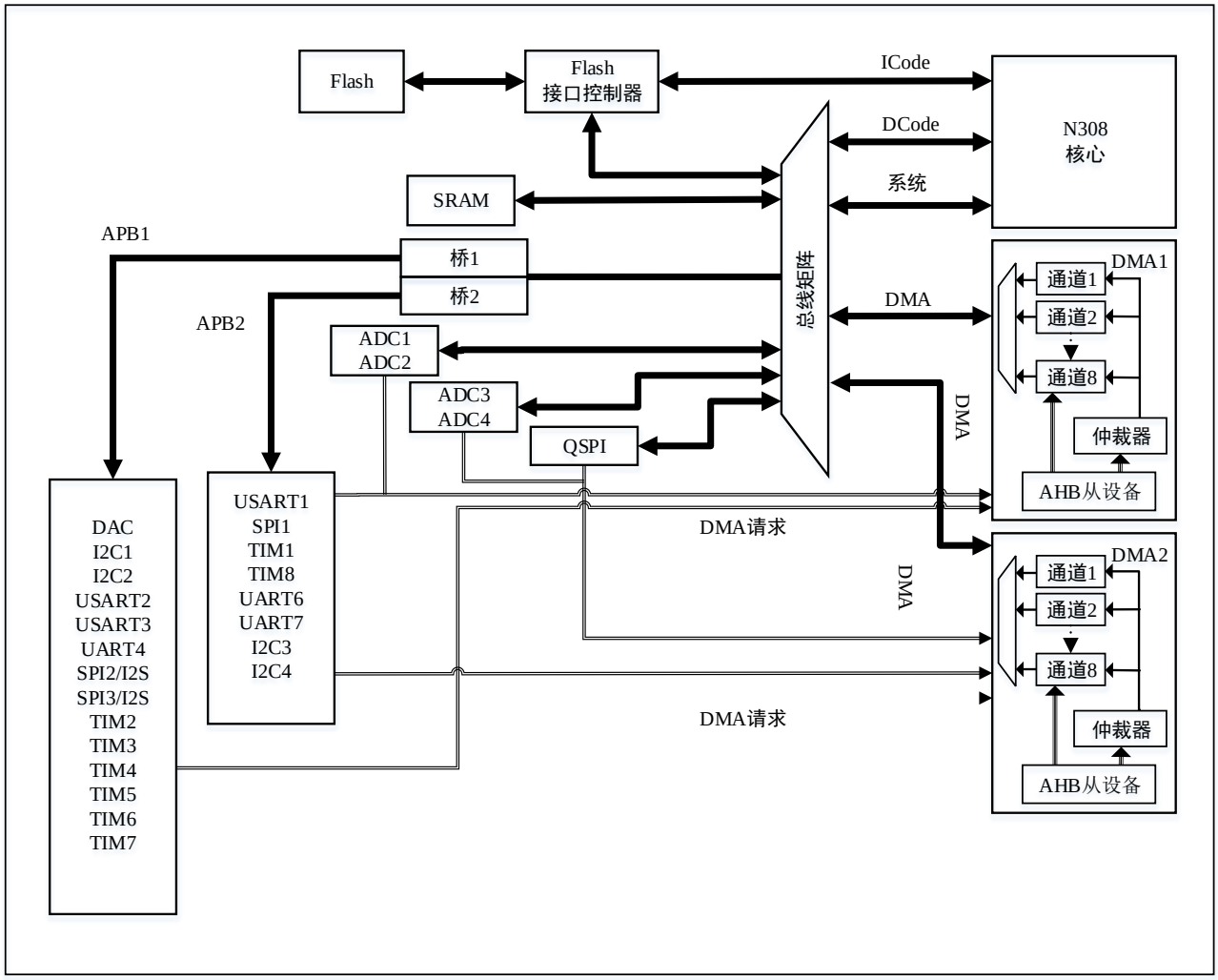
芯片有两个 DMA 控制器共 16 个通道（DMA1、DMA2 分别有 8 个通道），每个通道都能用来管理一个或多个外设对存储器的访问请求。还有一个仲裁器用来协调多个 DMA 通道同时访问的优先级问题。

8.2 DMA 主要特性

- 16 个独立的可配置的通道：DMA1 有 8 个通道，DMA2 有 8 个通道。
- 支持外设和存储器、存储器和外设之间的传输
- 支持存储器和存储器间的传输
- 支持循环的缓冲器管理
- 可编程的数据传输长度：最大 65535。
- 每个通道都直接连接专用的硬件 DMA 请求，同时支持软件触发。
- Flash、SRAM、外设的 SRAM、APB1、APB2 和 AHB 外设均可作为访问的源和目标。
- 多个请求间的优先权可以通过软件编程设置（共四级：最高、高、中和低），优先权设置相等时由硬件决定（低通道号的优先级更高）。
- 独立数据源和目标数据区的传输宽度（字节、半字、全字），模拟打包和拆包的过程。源和目标地址必须按数据传输宽度对齐。
- 每个通道都有 3 种事件标志（HTXF 半传输、TXCF 传输完成和 ERRF 传输出错），这 3 种事件标志可配置触发一个单独的中断请求。

8.3 功能框图

图 8-1 DMA 框图



8.4 功能描述

DMA 控制器和 N308 核心分时共用系统数据总线，实现直接存储器数据传输。DMA 向 CPU 发起请求以获取数据总线控制权，操作结束后将总线控制权交还给 CPU。当 CPU 和 DMA 同时访问相同的目标（RAM 或外设）时，DMA 请求会挂起 CPU 访问系统总线达若干个周期，由总线仲裁器执行循环调度，以协调 CPU 至少可以得到一半的系统总线（存储器或外设）访问时序资源。

8.4.1 DMA 处理

可通过外设硬件触发或者软件触发一个 DMA 请求，DMA 控制器按照通道的优先权等级处理请求。根据配置的传输地址和位宽从源地址读取数据，然后将读取到的数据存储到目标地址空间。执行一次操作后控制器计算剩余传输次数并更新下一次传输的源地址和目标地址。

每次 DMA 数据传送由 3 个操作组成：

- 取数据：根据传输方向确定源地址（DMA_PADDRx 或 DMA_MADDRx），读取源地址数据。
- 存数据：根据传输方向确定目标地址（DMA_PADDRx 或 DMA_MADDRx），将读取到的数据存储到目标地址空间。
- 计算未完成操作数目，执行一次 DMA_TXNUMx 寄存器的递减操作，并更新下一次操作源地址和目标地址。

8.4.2 仲裁器

DMA 仲裁器用来处理多个通道同时发起访问请求的情况，根据每个通道的优先级来决定响应哪一个通道请求。优先等级判断包括软件和硬件两种方式：

- 软件方式：有 4 个等级，每个通道优先级可以通过 DMA_CHCFGx 寄存器设置：
  - 最高优先级
  - 高优先级
  - 中优先级
  - 低优先级
- 硬件方式：当请求通道的软件优先级相同，则编号低的通道优先级高。例如：通道 1 和通道 2 配置为相同的软件优先级，则通道 1 的优先级更高。

8.4.3 DMA 通道

每个通道都可以在指定地址的外设寄存器和存储器地址之间执行 DMA 传输。DMA 传输数据量是可编程的，最大支持 65535。DMA_TXNUM 寄存器在每次传输后递减。

8.4.3.1 可编程的数据位宽

外设和存储器的传输数据位宽支持字节、半字、字，可以通过 DMA_CHCFGx 寄存器中的 PSIZE 和 MSIZE 位编程。

8.4.3.2 指针增量

DMA_CHCFGx 寄存器中的 PINC 和 MINC 位分别控制外设地址和存储器地址是否开启自增模式。传输过程中软件不能读取和写入地址寄存器。

- 在自动增量模式下，每次传输完成后下一个要传输的地址会根据数据位宽（1、2 或 4）自动增加。首次传输的地址存放在 DMA_PADDRx/DMA_MADDRx 寄存器中。
- 固定模式下，地址一直固定为初始化地址。

在一次传输结束时（即传输计数变为 0），会根据当前是否工作在循环模式进行不同处理：

- 非循环模式时，传输结束后将不再产生 DMA 操作。要开始新的 DMA 传输，需要在关闭 DMA 通道的情况下，在 DMA_TXNUMx 寄存器中重新写入传输数目。
- 在循环模式下，一次传输结束时，DMA_TXNUMx 寄存器的内容会自动地被重新加载为其初始数值，内部的当前外设/存储器地址寄存器也被重新加载为 DMA_PADDRx/DMA_MADDRx 寄存器设定的

初始基地址。

8.4.3.3 通道配置过程

下面是配置 DMA 通道 x 的过程（x 代表通道号）：

1. 在 DMA_PADDRx 寄存器中设置外设寄存器的地址。
2. 在 DMA_MADDRx 寄存器中设置数据存储器的地址。
3. 在 DMA_TXNUMx 寄存器中设置要传输的数据量。在每个数据传输后，这个数值递减。
4. 在 DMA_CHCFGx 寄存器中设置数据传输的方向、循环模式、外设和存储器的增量模式、外设和存储器的数据宽度、传输一半产生中断或传输完成产生中断。
5. 在 DMA_CHCFGx 寄存器的 PRIOLVL[1:0]位中设置通道的优先级。
6. 设置 DMA_CHCFGx 寄存器的 CHEN 位，启动该通道。

一旦启动了 DMA 通道，它即可响应连到该通道上的外设的 DMA 请求。传输过程中，当数据传输一半时 HTXF 被置一，当数据传输完成后 TXCF 被置一，当传输错误时 ERRF 被置一。这三个标志可通过开启对应中断使能共用一个独立中断资源。

8.4.3.4 循环模式

循环模式用于处理循环缓冲区和连续的数据传输（如 ADC 的扫描模式）。在 DMA_CHCFGx 寄存器中的 CIRC 位用于开启这一功能。当启动了循环模式，当数据传输的数目变为 0 时，将会自动地被恢复成配置通道时设置的初值，DMA 操作将会继续进行。

如果想要关闭循环模式，需要先对 DMA_CHCFGx 寄存器中的 CHEN 位写 0 以关闭 DMA 通道，然后再对 DMA_CHCFGx 寄存器中的 CIRC 位写 0（CHEN 位为 1 时，DMA_CHCFGx 寄存器中其它位不能被改写）。

8.4.3.5 存储器到存储器模式

DMA 通道可以配置为存储器到存储器模式，不需要外设请求情况下即可进行数据传输，不过该模式不能工作在循环模式下。通过将 DMA_CHCFGx 寄存器中的 MEM2MEM 位置一使能该模式，在软件设置了 DMA_CHCFGx 寄存器中的 CHEN 位启动 DMA 通道时，DMA 传输将马上开始，当 DMA_TXNUMx 寄存器变为 0 时，DMA 传输结束。

8.4.4 可编程的数据传输宽度、对齐方式和数据大小端

当 PSIZE 和 MSIZE 不相同时，DMA 模块按照下表进行数据对齐。

表 8-1 可编程的数据传输宽度和大小端操作（当 PINC = MINC = 1）

源端宽度	目标宽度	传输数目	源：地址/数据	传输操作	目标：地址/数据
8	8	4	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3	1：在0x0读B0[7:0]，在0x0写B0[7:0] 2：在0x1读B1[7:0]，在0x1写B1[7:0] 3：在0x2读B2[7:0]，在0x2写B2[7:0] 4：在0x3读B3[7:0]，在0x3写B3[7:0]	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3
8	16	4	0x0 / B0	1：在0x0读B0[7:0]，在0x0写00B0[15:0]	0x0 / 00B0

			0x1 / B1 0x2 / B2 0x3 / B3	2: 在0x1读B1[7:0]，在0x2写00B1[15:0] 3: 在0x2读B2[7:0]，在0x4写00B2[15:0] 4: 在0x3读B3[7:0]，在0x6写00B3[15:0]	0x2 / 00B1 0x4 / 00B2 0x6 / 00B3
8	32	4	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3	1: 在0x0读B0[7:0]，在0x0写000000B0[31:0] 2: 在0x1读B1[7:0]，在0x4写000000B1[31:0] 3: 在0x2读B2[7:0]，在0x8写000000B2[31:0] 4: 在0x3读B3[7:0]，在0xC写000000B3[31:0]	0x0 / 000000B0 0x4 / 000000B1 0x8 / 000000B2 0xC / 000000B3
16	8	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: 在0x0读B1B0[15:0]，在0x0写B0[7:0] 2: 在0x2读B3B2[15:0]，在0x1写B2[7:0] 3: 在0x4读B5B4[15:0]，在0x2写B4[7:0] 4: 在0x6读B7B6[15:0]，在0x3写B6[7:0]	0x0 / B0 0x1 / B2 0x2 / B4 0x3 / B6
16	16	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: 在0x0读B1B0[15:0]，在0x0写B1B0[15:0] 2: 在0x2读B3B2[15:0]，在0x2写B3B2[15:0] 3: 在0x4读B5B4[15:0]，在0x4写B5B4[15:0] 4: 在0x6读B7B6[15:0]，在0x6写B7B6[15:0]	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6
16	32	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: 在0x0读B1B0[15:0]，在0x0写0000B1B0[31:0] 2: 在0x2读B3B2[15:0]，在0x4写0000B3B2[31:0] 3: 在0x4读B5B4[15:0]，在0x8写0000B5B4[31:0] 4: 在0x6读B7B6[15:0]，在0xC写0000B7B6[31:0]	0x0 / 0000B1B0 0x4 / 0000B3B2 0x8 / 0000B5B4 0xC / 0000B7B6
32	8	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: 在0x0读B3B2B1B0[31:0]，在0x0写B0[7:0] 2: 在0x4读B7B6B5B4[31:0]，在0x1写B4[7:0] 3: 在0x8读BBBAB9B8[31:0]，在0x2写B8[7:0] 4: 在0xC读BFBEBDBC[31:0]，在0x3写BC[7:0]	0x0 / B0 0x1 / B4 0x2 / B8 0x3 / BC
32	16	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: 在0x0读B3B2B1B0[31:0]，在0x0写B1B0[15:0] 2: 在0x4读B7B6B5B4[31:0]，在0x2写B5B4[15:0] 3: 在0x8读BBBAB9B8[31:0]，在0x4写B9B8[15:0] 4: 在0xC读BFBEBDBC[31:0]，在0x6写BDBC[15:0]	0x0 / B1B0 0x2 / B5B4 0x4 / B9B8 0x6 / BDBC
32	32	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: 在0x0读B3B2B1B0[31:0]，在0x0写B3B2B1B0[31:0] 2: 在0x4读B7B6B5B4[31:0]，在0x4写B7B6B5B4[31:0] 3: 在0x8读BBBAB9B8[31:0]，在0x8写BBBAB9B8[31:0] 4: 在0xC读BFBEBDBC[31:0]，在0xC写BFBEBDBC[31:0]	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC

8.4.4.1 操作一个不支持字节或半字写的 AHB 设备

当 DMA 模块开始一个 AHB 的字节或半字写操作时，数据将在 HWDATA[31:0]总线中未使用的部分重复。因此，如果 DMA 以字节或半字写入不支持字节或半字写操作的 AHB 设备时（即 HSIZE 不适于该模块），不会发生错误，DMA 将按照下面两个例子写入 32 位 HWDATA 数据:

- 当 HSIZE=半字时，写入半字'0xABCD'，DMA 将设置 HWDATA 总线为'0xABCDABCD'。
- 当 HSIZE=字节时，写入字节'0xAB'，DMA 将设置 HWDATA 总线为'0xABABABAB'。

假定 AHB/APB 桥是一个 AHB 的 32 位从设备，它不处理 HSIZE 参数，它将按照下述方式把任何

AHB 上的字节或半字按 32 位传送到 APB 上:

- 一个 AHB 上对地址 0x0（或 0x1、0x2 或 0x3）的写字节数据'0xB0'操作，将转换到 APB 上对地址 0x0 的写字数据'0xB0B0B0B0'操作。
- 一个 AHB 上对地址 0x0（或 0x2）的写半字数据'0xB1B0'操作，将转换到 APB 上对地址 0x0 的写字数据'0xB1B0B1B0'操作。

例如，如果要写入 APB 后备寄存器（与 32 位地址对齐的 16 位寄存器），需要配置存储器数据源宽度（MSIZE）为'16 位'，外设目标数据宽度（PSIZE）为'32 位'。

8.4.5 错误管理

对一个保留的地址区域进行过 DMA 访问，将会产生 DMA 传输错误。当错误发生时，传输错误标志位（TEIF）被置为，硬件会自动的清除当前 DMA 通道使能位（DMA_CHCFGx 的 CHEN），该通道操作被停止。如在 DMA_CHCFGx 寄存器中设置了传输错误中断允许位，则将产生中断。

8.4.6 中断

每个 DMA 通道都拥有一个独立的中断，可在传输过半、传输完成、传输错误时产生中断，通过寄存器对应的中断使能位来打开这些中断，用户可以灵活配置。

表 8-2 DMA 中断请求

中断事件	事件标志位	使能控制位
传输过半	HTXF	HTXIE
传输完成	TXCF	TXCIE
传输错误	ERRF	ERRIE

8.4.7 DMA 请求映像

8.4.7.1 DMA1 控制器

DMA1 控制器的 8 个通道可从外设（TIMx[x=1、2、3、4]、ADCx[x=1、2]、SPI1、SPI/I2S2、I2Cx[x=1、2、3]、UART5 和 USARTx[x=1、2、3]）获得的 8 个独立的请求，通过逻辑或输入到 DMA1 控制器，参见下图的 DMA1 请求映像。根据通道优先级，同一时间只有一个请求有效。通过配置对应外设的寄存器，每个外设的 DMA 请求均可以独立的开启或关闭。

图 8-2 DMA1 请求映像

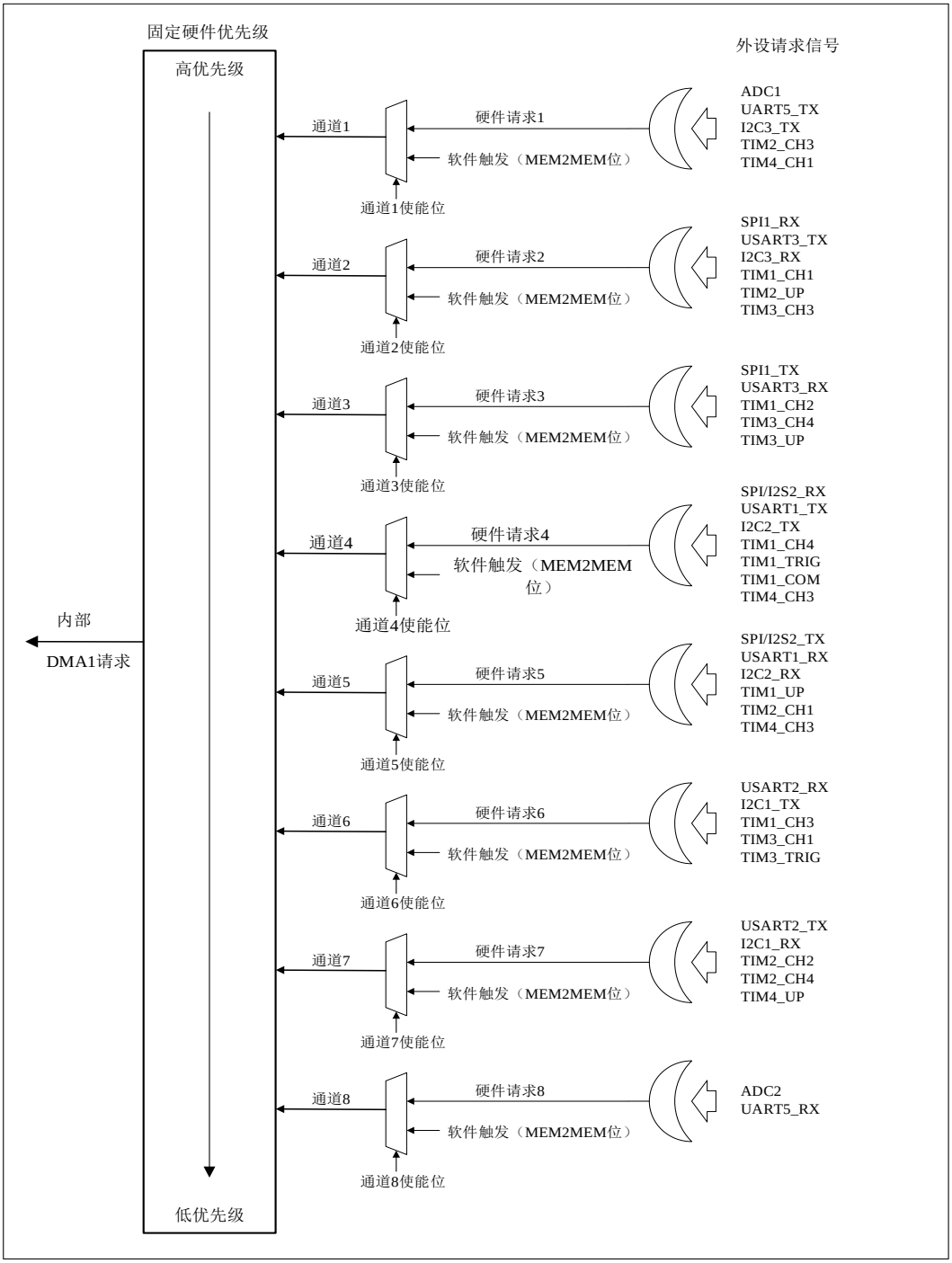


表 8-3 各个通道的 DMA1 请求一览

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7	通道 8
ADC	ADC1							ADC2
SPI/I2S		SPI1_RX	SPI1_TX	SPI/I2S2_RX	SPI/I2S2_TX			
USART	UART5_TX	USART3_TX	USART3_RX	USART1_TX	USART1_RX	USART2_RX	USART2_TX	UART5_RX
I2C	I2C3_TX	I2C3_RX		I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX	

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7	通道 8
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3		
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4	
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP			TIM3_CH1 TIM3_TRIG		
TIM4	TIM4_CH1			TIM4_CH2	TIM4_CH3		TIM4_UP	

8.4.7.2 DMA2 控制器

DMA2 控制器的 8 个通道可从外设（TIMx[x=5、6、7、8]、ADCx[x=3、4]、SPI/I2S3、UARTx[x=4、6、7]、QSPI、DAC 通道 1、2）获得的 6 个独立的请求，通过逻辑或输入到 DMA2 控制器，参见下图的 DMA1 请求映像。根据通道优先级，同一时间只有一个请求有效。通过配置对应外设的寄存器，每个外设的 DMA 请求均可以独立的开启或关闭。

图 8-3 DMA2 请求映像

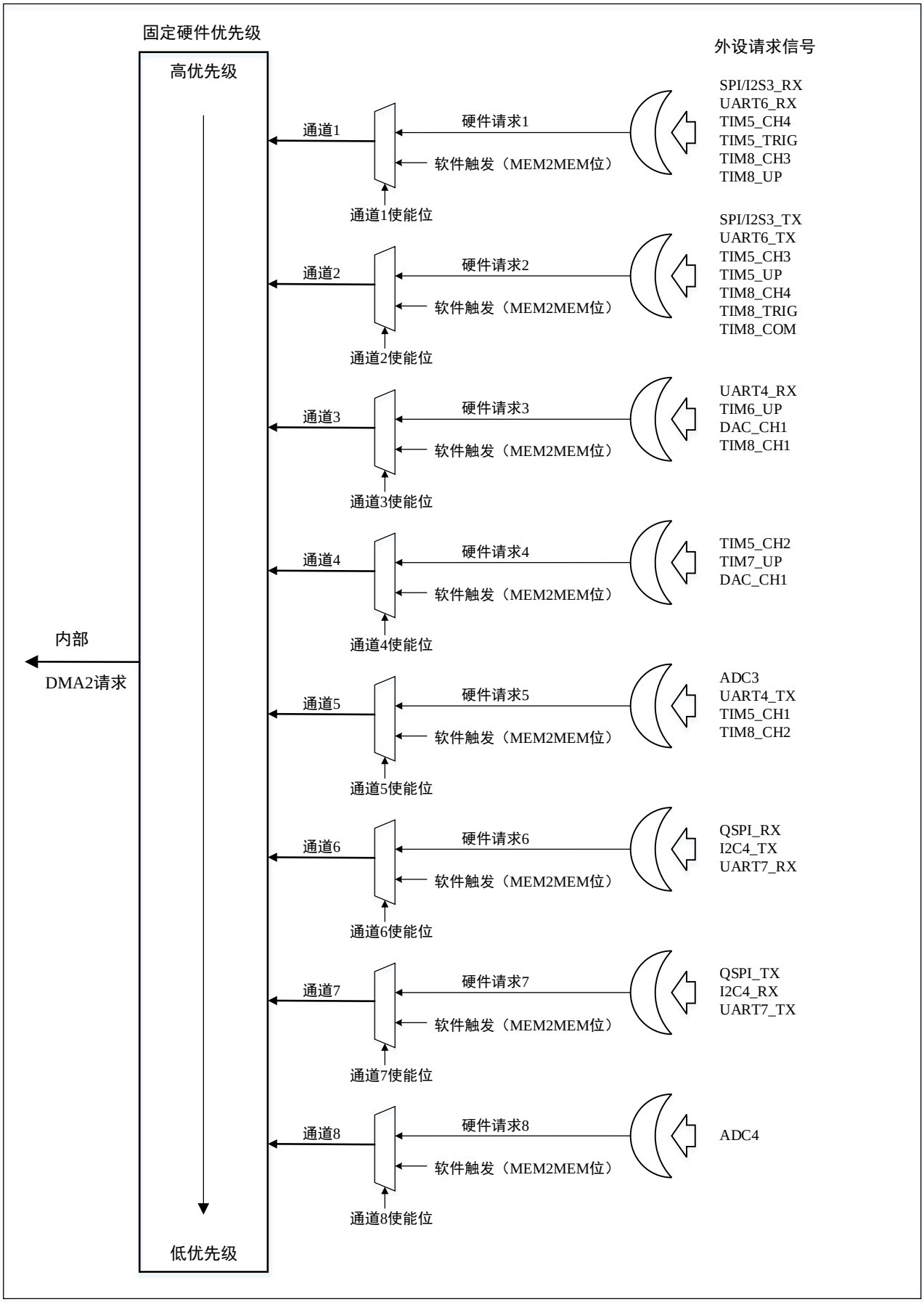

固定硬件优先级
高优先级
外设请求信号
通道1
硬件请求1
软件触发（MEM2MEM位）
通道1使能位
SPI/I2S3_RX
UART6_RX
TIM5_CH4
TIM5_TRIG
TIM8_CH3
TIM8_UP
通道2
硬件请求2
软件触发（MEM2MEM位）
通道2使能位
SPI/I2S3_TX
UART6_TX
TIM5_CH3
TIM5_UP
TIM8_CH4
TIM8_TRIG
TIM8_COM
通道3
硬件请求3
软件触发（MEM2MEM位）
通道3使能位
UART4_RX
TIM6_UP
DAC_CH1
TIM8_CH1
通道4
硬件请求4
软件触发（MEM2MEM位）
通道4使能位
TIM5_CH2
TIM7_UP
DAC_CH1
内部
DMA2请求
通道5
硬件请求5
软件触发（MEM2MEM位）
通道5使能位
ADC3
UART4_TX
TIM5_CH1
TIM8_CH2
通道6
硬件请求6
软件触发（MEM2MEM位）
通道6使能位
QSPI_RX
I2C4_TX
UART7_RX
通道7
硬件请求7
软件触发（MEM2MEM位）
通道7使能位
QSPI_TX
I2C4_RX
UART7_TX
通道8
硬件请求8
软件触发（MEM2MEM位）
通道8使能位
ADC4
低优先级

表 8-4 各个通道的 DMA2 请求一览

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7	通道 8
ADC					ADC3			ADC4
SPI/I2S3	SPI/I2S3_RX	SPI/I2S3_TX				QSPI_RX	QSPI_TX	
I2C4						I2C4_TX	I2C4_RX	
UART	UART6_RX	UART6_TX	UART4_RX		UART4_TX	UART7_RX	UART7_TX	
TIM5	TIM5_CH4 TIM5_TRIG	TIM5_CH3 TIM5_UP		TIM5_CH2	TIM5_CH1			
TIM6/DAC			TIM6_UP DAC_CH1					
TIM7/DAC				TIM7_UP DAC_CH2				
TIM8	TIM8_CH3 TIM8_UP	TIM8_CH4 TIM8_TRIG TIM8_COM	TIM8_CH1		TIM8_CH2			

8.5 DMA 寄存器

8.5.1 DMA 寄存器地址映像

表 8-5 DMA 寄存器地址映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	DMA_INTSTS	ERRF8	HTXF8	TXCF8	GLBF8	ERRF7	HTXF7	TXCF7	GLBF7	ERRF6	HTXF6	TXCF6	GLBF6	ERRF5	HTXF5	TXCF5	GLBF5	ERRF4	HTXF4	TXCF4	GLBF4	ERRF3	HTXF3	TXCF3	GLBF3	ERRF2	HTXF2	TXCF2	GLBF2	ERRF1	HTXF1	TXCF1	GLBF1
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	DMA_INTCLR	CERRF8	CHTXF8	CTXCF8	CGLBF8	CERRF7	CHTXF7	CTXCF7	CGLBF7	CERRF6	CHTXF6	CTXCF6	CGLBF6	CERRF5	CHTXF5	CTXCF5	CGLBF5	CERRF4	CHTXF4	CTXCF4	CGLBF4	CERRF3	CHTXF3	CTXCF3	CGLBF3	CERRF2	CHTXF2	CTXCF2	CGLBF2	CERRF1	CHTXF1	CTXCF1	CGLBF1
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	DMA_CHCFG1	Reserved																	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN	
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	DMA_TXNUM1	Reserved																	NDTX[15:0]														
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	DMA_PADDR1	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	DMA_MADDR1	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	DMA_CHSEL1	Reserved																										CH_SEL[5:0]					
	Reset Value																											0	0	0	0	0	0
01Ch	DMA_CHCFG2	Reserved																	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN	
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
020h	DMA_TXNUM2	Reserved																	NDTX[15:0]														
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
024h	DMA_PADDR2	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028h	DMA_MADDR2	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	DMA_CHSEL2	Reserved																										CH_SEL[5:0]					
	Reset Value																											0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
030h	DMA_CHCFG3	Reserved																	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
034h	DMA_TXNUM3	Reserved																	NDTX[15:0]														
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
038h	DMA_PADDR3	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03Ch	DMA_MADDR3	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
040h	DMA_CHSEL3	Reserved																										CH_SEL[5:0]					
	Reset Value																											0	0	0	0	0	0
044h	DMA_CHCFG4	Reserved																	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
048h	DMA_TXNUM4	Reserved																	NDTX[15:0]														
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04Ch	DMA_PADDR4	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
050h	DMA_MADDR4	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
054h	DMA_CHSEL4	Reserved																										CH_SEL[5:0]					
	Reset Value																											0	0	0	0	0	0
058h	DMA_CHCFG5	Reserved																	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
05Ch	DMA_TXNUM5	Reserved																	NDTX[15:0]														
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
060h	DMA_PADDR5	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
064h	DMA_MADDR5	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
068h	DMA_CHSEL5	Reserved																										CH_SEL[5:0]					
	Reset Value																											0	0	0	0	0	0
06Ch	DMA_CHCFG6	Reserved																	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
070h	DMA_TXNUM6	Reserved																	NDTX[15:0]														
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
074h	DMA_PADDR6	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
078h	DMA_MADDR6	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07Ch	DMA_CHSEL6	Reserved																										CH_SEL[5:0]					
	Reset Value																											0	0	0	0	0	0
080h	DMA_CHCFG7	Reserved																	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
084h	DMA_TXNUM7	Reserved																	NDTX[15:0]														
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
088h	DMA_PADDR7	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
08Ch	DMA_MADDR7	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
090h	DMA_CHSEL7	Reserved																										CH_SEL[5:0]					
	Reset Value																											0	0	0	0	0	0
094h	DMA_CHCFG8	Reserved																	MEM2MEM	PRIOLVL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
098h	DMA_TXNUM8	Reserved																	NDTX[15:0]														
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
09Ch	DMA_PADDR8	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0A0h	DMA_MADDR8	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0A4h	DMA_CHSEL8	Reserved																										CH_SEL[5:0]					
	Reset Value																											0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0A8h	DMA_CHMAPEN	Reserved																															MAP_EN
	Reset Value	0

8.5.2 DMA 中断状态寄存器（DMA_INTSTS）

偏移地址：0x00
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERRF8	HTXF8	TXCF8	GLBF8	ERRF7	HTXF7	TXCF7	GLBF7	ERRF6	HTXF6	TXCF6	GLBF6	ERRF5	HTXF5	TXCF5	GLBF5
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRF4	HTXF4	TXCF4	GLBF4	ERRF3	HTXF3	TXCF3	GLBF3	ERRF2	HTXF2	TXCF2	GLBF2	ERRF1	HTXF1	TXCF1	GLBF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位域	名称	描述
31/27/23/19/15/11/7/3	ERRFx	通道 x 的传输错误标志位（x=1…8）。 硬件置位，软件写 DMA_INTCLR 对应位为 1 清零。 0：在通道 x 没有传输错误 1：在通道 x 发生了传输错误
30/26/22/18/14/10/6/2	HTXFx	通道 x 的半传输完成标志位（x=1…8）。 硬件置位，软件写 DMA_INTCLR 对应位为 1 清零。 0：在通道 x 没有半传输事件 1：在通道 x 产生了半传输事件
29/25/21/17/13/9/5/1	TXCFx	通道 x 的传输完成标志位（x=1…8）。 硬件置位，软件写 DMA_INTCLR 对应位为 1 清零。 0：在通道 x 没有传输完成事件 1：在通道 x 产生了传输完成事件
28/24/20/16/12/8/4/0	GLBFx	通道 x 的全局中断标志位（x=1…8）。 硬件置位，软件写 DMA_INTCLR 对应位为 1 清零。 0：在通道 x 没有传输错误、半传输、传输完成事件 1：在通道 x 至少产生了传输错误、半传输、传输完成事件之一

8.5.3 DMA 中断标志清除寄存器（DMA_INTCLR）

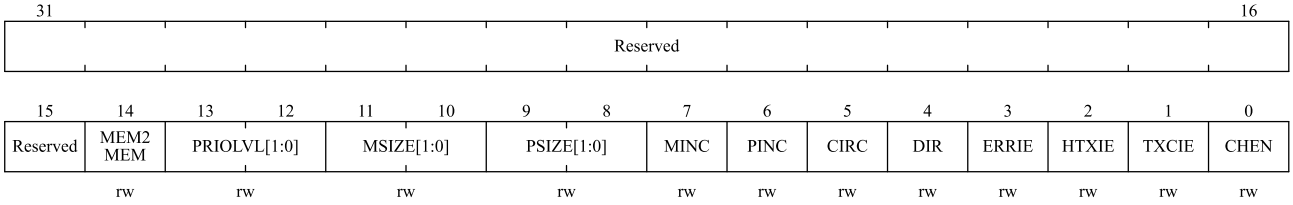
偏移地址：0x04
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CERRF8	CHTXF8	CTXCF8	CGLBF8	CERRF7	CHTXF7	CTXCF7	CGLBF7	CERRF6	CHTXF6	CTXCF6	CGLBF6	CERRF5	CHTXF5	CTXCF5	CGLBF5
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CERRF4	CHTXF4	CTXCF4	CGLBF4	CERRF3	CHTXF3	CTXCF3	CGLBF3	CERRF2	CHTXF2	CTXCF2	CGLBF2	CERRF1	CHTXF1	CTXCF1	CGLBF1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31/27/23/19/15/11/7/3	CERRFx	清除通道 x 的传输错误标志位（x=1…8）。 软件置位和清除。 0：无作用 1：清除 DMA_INTSTS 寄存器中对应的 ERRFx 标志
30/26/22/18/14/10/6/2	CHTXFx	清除通道 x 的半传输完成标志位（x=1…8）。 软件置位和清除。 0：无作用 1：清除 DMA_INTSTS 寄存器中对应的 HTXFx 标志
29/25/21/17/13/9/5/1	CTXCFx	清除通道 x 的传输完成标志位（x=1…8）。 软件置位和清除。 0：无作用 1：清除 DMA_INTSTS 寄存器中对应的 TXCFx 标志
28/24/20/16/12/8/4/0	CGLBFx	清除通道 x 的全局中断标志位（x=1…8）。 软件置位和清除。 0：无作用 1：清除 DMA_INTSTS 寄存器中对应的 GLBFx、TXCFx、HTXFx 以及 ERRFx 标志

8.5.4 DMA 通道 x 配置寄存器（DMA_CHCFGx）

x 为通道号，x=1 … 8
地址偏移：0x08 + 20 * (x – 1)
复位值：0x0000 0000



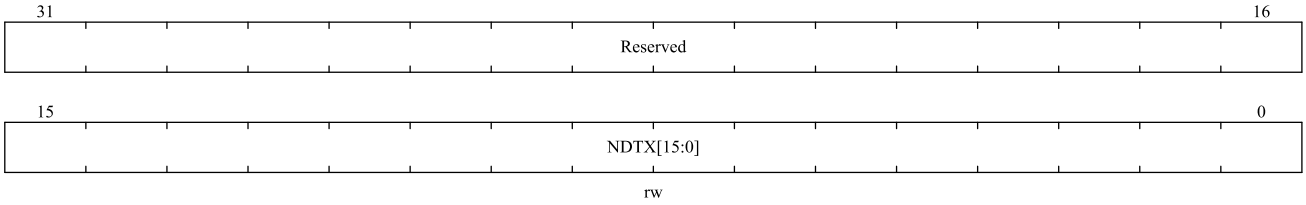
位域	名称	描述
31:15	Reserved	保留，硬件强制为 0。
14	MEM2MEM	存储器到存储器模式。 软件置位和清除。 0：非存储器到存储器模式 1：启动存储器到存储器模式
13:12	PRIOLVL[1:0]	通道优先级。 软件置位和清除。 00：低 01：中 10：高 11：最高
11:10	MSIZE[1:0]	存储器数据宽度。

位域	名称	描述
		软件置位和清除。 00：8 位 01：16 位 10：32 位 11：保留
9:8	PSIZE[1:0]	外设数据宽度。 软件置位和清除。 00：8 位 01：16 位 10：32 位 11：保留
7	MINC	存储器地址增量模式。 软件置位和清除。 0：固定地址模式 1：增量地址模式
6	PINC	外设地址增量模式。 软件置位和清除。 0：固定地址模式 1：增量地址模式
7	CIRC	循环模式。 软件置位和清除。 0：禁止循环模式 1：使能循环模式
4	DIR	数据传输方向。 软件置位和清除。 0：从外设读出，写入存储器 1：从存储器读出，写入外设
3	ERRIE	通道传输错误中断使能。 软件置位和清除。 0：禁止产生通道传输错误中断 1：使能产生通道传输错误中断
2	HTXIE	通道半传输完成中断使能。 软件置位和清除。 0：禁止产生通道半传输完成中断 1：使能产生通道半传输完成中断
1	TXCIE	通道传输完成中断使能。 软件置位和清除。 0：禁止产生通道传输完成中断 1：使能产生通道传输完成中断
0	CHEN	通道使能。 软件置位和清除。 0：禁止通道工作

位域	名称	描述
		1：使能通道

8.5.5 DMA 通道 x 传输数量寄存器（DMA_TXNUMx）

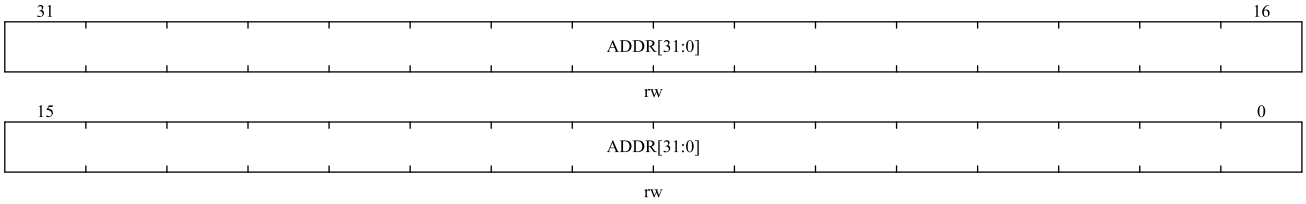
x 为通道号，x=1 … 8
地址偏移：0x0C + 20 * (x – 1)
复位值：0x0000 0000
当通道开启（DMA_CHCFGx 的 CHEN=1）时不能写该寄存器。



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	NDTX	数据传输数量。 取值为 0 至 65535。此寄存器只能在通道禁用时写入。通道开启后该寄存器变为只读，指示剩余的待传输字节数目。寄存器内容在每次 DMA 传输后递减。 数据传输结束后，寄存器的内容会被自动重装载为初始设置值。 当寄存器的内容为 0 时，无论通道是否开启，都不会发生任何数据传输。

8.5.6 通道 x 外设基地址寄存器（DMA_PADDRx）

x 为通道号，x=1 … 8
地址偏移：0x10 + 20 * (x – 1)
复位值：0x0000 0000
当通道开启（DMA_CHCFGx 的 CHEN=1）时不能写该寄存器。

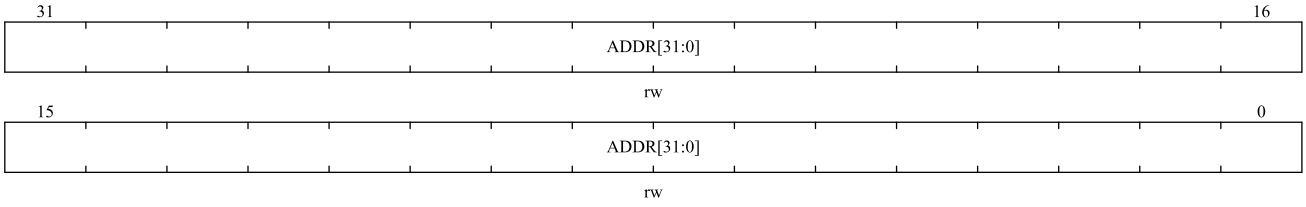


位域	名称	描述
31:0	ADDR	外设基地址，数据传输的源或目标地址。 当 PSIZE=01（16 位），不使用 ADDR[0]位。操作自动与半字地址对齐。 当 PSIZE=10（32 位），不使用 ADDR[1:0]位。操作自动与字地址对齐。

8.5.7 通道 x 存储器基地址寄存器（DMA_MADDRx）

x 为通道号，x=1 … 8

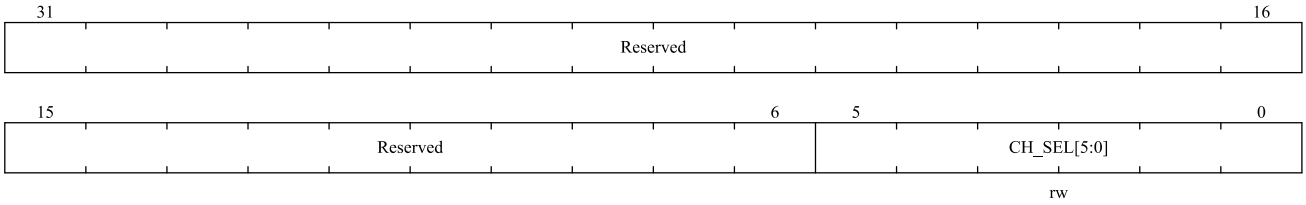
地址偏移：0x14 + 20 * (x – 1)
复位值：0x0000 0000
当通道开启（DMA_CHCFGx 的 CHEN=1）时不能写该寄存器。



位域	名称	描述
31:0	ADDR	存储器基地址，数据传输的源或目标地址。 当 MSIZE=01（16 位），不使用 ADDR[0]位。操作自动与半字地址对齐。 当 MSIZE=10（32 位），不使用 ADDR[1:0]位。操作自动与字地址对齐。

8.5.8 DMA1 通道 x 通道选择寄存器（DMA1_CHSELx）

x 为通道号，x=1 … 8
地址偏移：0x18 + 20 * (x – 1)
复位值：0x0000 0000
当通道 MAP 使能（DMA_CHMAPEN 的 MAP_EN=1）时该寄存器才有效。该寄存器用来管理 DMA1 外设请求映射的 DMA1 的通道。



位域	名称	描述
31:6	Reserved	保留，硬件强制为 0。
5:0	CH_SEL[5:0]	DMA 通道请求选择 40：UART5_RX 39：ADC2 38：I2C1_RX 37：TIM4_UP 36：TIM2_CH4 35：TIM2_CH2 34：USART2_TX 33：I2C1_TX 32：TIM3_TRIG 31：TIM3_CH1 30：TIM1_CH3 29：USART2_RX 28：I2C2_RX

位域	名称	描述
		27：TIM4_CH3 26：TIM2_CH1 25：SPI/I2S2_TX 24：TIM1_UP 23：USART1_RX 22：I2C2_TX 21：SPI/I2S2_RX 20：TIM4_CH2 19：TIM1_COM 18：TIM1_TRIG 17：TIM1_CH4 16：USART1_TX 15：SPI1_TX 14：TIM3_UP 13：TIM3_CH4 12：TIM1_CH2 11：USART3_RX 10：SPI1_RX 9：TIM3_CH3 8：TIM2_UP 7：TIM1_CH1 6：I2C3_RX 5：USART3_TX 4：TIM4_CH1 3：TIM2_CH3 2：I2C3_TX 1：UART5_TX 0：ADC1

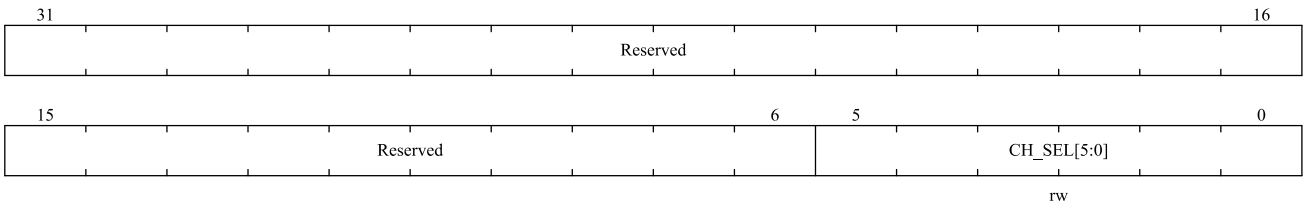
8.5.9 DMA2 通道 x 通道选择寄存器（DMA2_CHSELx）

x 为通道号，x=1 … 8

地址偏移：0x18 + 20 * (x – 1)

复位值：0x0000 0000

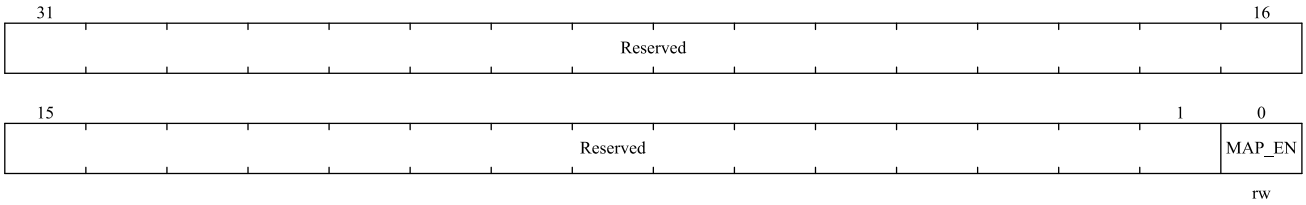
当通道 MAP 使能（DMA_CHMAPEN 的 MAP_EN=1）时该寄存器才有效。该寄存器用来管理 DMA2 外设请求映射的 DMA2 的通道。



位域	名称	描述
31:6	Reserved	保留，硬件强制为 0。
5:0	CH_SEL[5:0]	DMA 通道请求选择 32：保留 31：ADC4 30：UART7_TX 29：I2C4_RX 28：QSPI_TX 27：UART7_RX 26：I2C4_TX 25：QSPI_RX 24：UART4_TX 23：TIM5_CH1 22：TIM8_CH2 21：ADC3 20：DAC2 19：TIM7_UP 18：保留 17：TIM5_CH2 16：DAC1 15：TIM6_UP 14：UART4_RX 13：TIM8_CH1 12：UART6_TX 11：SPI/I2S3_TX 10：TIM5_UP 9：TIM5_CH3 8：TIM8_COM 7：TIM8_TRIG 6：TIM8_CH4 5：UART6_RX 4：SPI/I2S3_RX 3：TIM8_UP 2：TIM8_CH3 1：TIM5_TRIG 0：TIM5_CH4

8.5.10 DMA 通道 MAP 使能寄存器（DMA_CHMAPEN）

地址偏移：0xA8
复位值：0x0000 0000



位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	MAP_EN	通道 MAP 使能。 0：禁止通道 MAP 1：使能通道 MAP

9 模拟/数字转换（ADC）

9.1 ADC 介绍

12 位 ADC 是一种高速逐次逼近型模拟数字转换器。它有多个通道。各通道的 A/D 转换可以单次、连续、扫描或间断模式执行。ADC 的结果可以左对齐或右对齐方式存储在 16 位数据寄存器中。

模拟看门狗特性允许应用程序检测输入电压是否超出用户定义的高/低阀值。

ADC 的输入时钟不得超过 72MHz。

9.2 ADC 主要特征

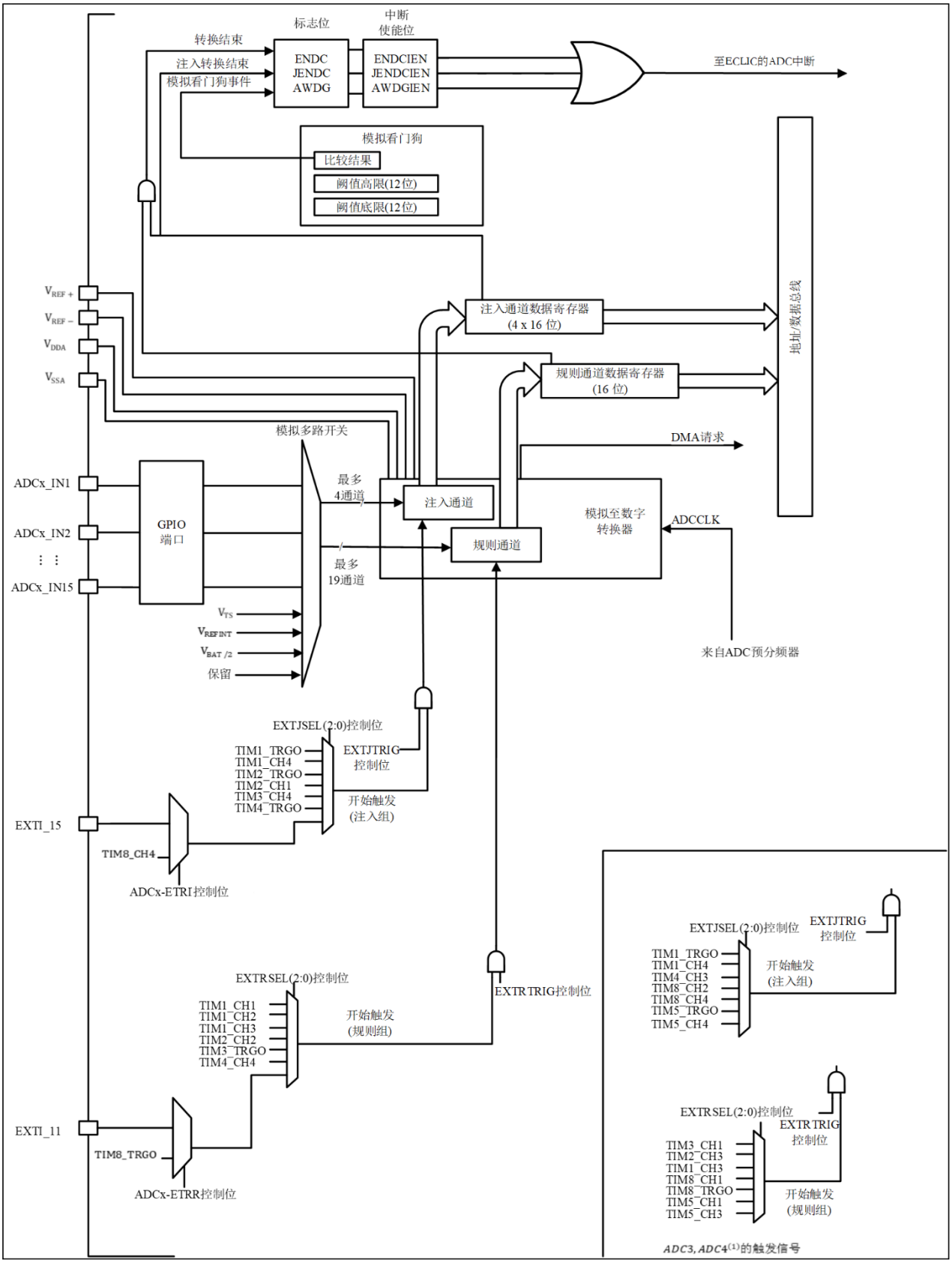
- 支持最多 4 个 ADC，支持单端输入和差分输入，可测量 40 个外部和 7 个内部信号源
- ADC1 支持 11 个外部通道，ADC2 支持 13 个外部通道，ADC3 支持 15 个外部通道，ADC4 支持 13 个外部通道
- 支持 12 位、10 位、8 位、6 位分辨率可配置
  - 12bit 分辨率下最高采样速率 5.14MSPS
  - 10bit 分辨率下最高采样速率 6MSPS
  - 8bit 分辨率下最高采样速率 7.2MSPS
  - 6bit 分辨率下最高采样速率 9MSPS
- ADC 时钟源分为工作时钟源、采样时钟源和计时时钟源
  - 仅可配置 AHB_CLK 作为工作时钟源，最高可到 144MHz
  - 可配置 PLL 作为采样时钟源，最高可到 72MHz，支持分频 1,2,4,6,8,10,12,16,32,64,128,256
  - 可配置 AHB_CLK 作为采样时钟源，最高可到 72MHz，支持分频 1,2,4,6,8,10,12,16,32
  - 计时时钟用于内部计时功能，频率必须配置成 1MHz
- 支持定时器触发 ADC 采样
- 转换结束、注入转换结束和发生模拟看门狗事件时产生中断
- 单次和连续转换模式
- 从通道 0 到通道 N 的自动扫描模式
- 支持自校准
- 带内嵌数据一致性的数据对齐
- 采样间隔可以按通道分别编程
- 规则转换和注入转换均有外部触发选项
- 间断模式
- 双重模式，ADC1 和 ADC2 组合、ADC3 和 ADC4 组合

- ADC 供电要求：1.8V 到 3.6V
- ADC 输入范围：$V_{REF-} \leqslant V_{IN} \leqslant V_{REF+}$
- 规则通道转换期间有 DMA 请求产生。

9.3 ADC 功能描述

下图为一个 ADC 模块的框图，表 9-1 为 ADC 引脚的说明。

图 9-1 单个 ADC 框图



1. ADC1 和 ADC2，ADC3 和 ADC4 的规则转换和注入转换触发参考具体的寄存器定义。

表 9-1 ADC 引脚

名称	信号类型	注解
V_{REF+}	输入，模拟参考正极	ADC使用的高端/正极参考电压，1.8V ≤ V_{REF+} ≤ V_{DDA}
V_{DDA}(1)	输入，模拟电源	等效于V_{DD}的模拟电源且：1.8V ≤ V_{DDA} ≤ V_{DD}(3.6V)
V_{REF-}	输入，模拟参考负极	ADC使用的低端/负极参考电压，V_{REF-} = V_{SSA}
V_{SSA}(1)	输入，模拟电源地	等效于V_{SS}的模拟电源地
ADCx_IN[15:1]	模拟输入信号	15个模拟外部输入通道

1． V_{DDA} 和 V_{SSA} 应该分别连接到 V_{DD} 和 V_{SS}。

2． 如果有 V_{REF-} 引脚（取决于封装），必须和 VSSA 相连接。

3． 如果没有 V_{REF+} 引脚（取决于封装），尽量保证 VDDA 和 VDD 的电压值一致，否则会影响 ADC 精度。

9.3.1 ADC 开关控制

用户必须等待 PowerUp 过程完成才可以进入下一步的操作, 可以通过查询 ADC_CTRL3 里面的 RDY 确认是否上电完成。

通过设置 ADC_CTRL2 寄存器的 ON 位可给 ADC 上电。当第一次设置 ON 位时，它将 ADC 从断电状态下唤醒。ADC 上电延迟一段时间后（t_{STAB}），再次设置 ON 位时开始进行转换。

通过清除 ON 位可以停止转换，并将 ADC 置于断电模式。在这个模式中，ADC 几乎不耗电（仅几个 μA）。用户可以通过查询 ADC_CTRL3 里面的 PDRDY 确认是否下电完成。

在 ADC Disable 的时候默认都是 PowerDown 模式，这个模式下只要不断电，不需要重新校正，校正值会在 ADC 自动保持。为了进一步的降低功耗 ，ADC 有设计一个深睡眠模式。会在 ADC Disable 进入深睡眠模式，ADC 内部的校正值会丢失，需要重新校正。深睡眠模式可以省大概 0.2μA 的功耗 ，

注意，当在双 ADC 模式时，最好双 ADC 都选同一种睡眠模式。控制 ADC 深睡眠模式的寄存器 ADC_CTRL3.DPWMOD。

9.3.2 ADC 时钟

ADC 需要三个时钟，ADC_CLK，HCLK, ADC_1MCLK。

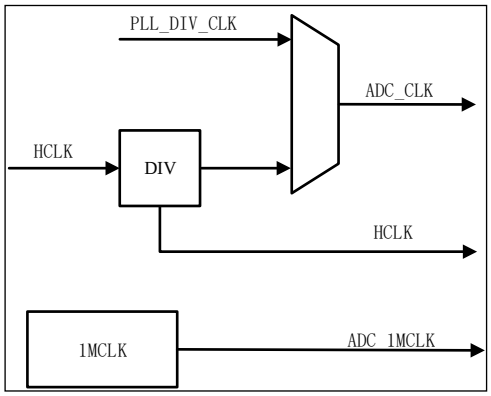
- HCLK 用于寄存器的访问时钟。
- ADC_CLK 为 ADC 的工作时钟，ADC_CLK 有两个源（HCLK 的分频或 PLL 的分频），HCLK 分频与系统是同步时钟，PLL 的分频与系统是异步时钟，用同步时钟的好处是在触发 ADC 响应触发时，没有不确定性，用 PLL 的分频时钟的好处是可以独立处理 ADC 的工作时钟，不会影响到挂在 HCLK 的其他模块
- ADC_1MCLK 用于内部计时功能，在 RCC 中配置，频率大小必须配置成 1M

注意

1. 配置 PLL 作为时钟源时，最高可到 72M,支持分频 1,2,4,6,8,10,12,16,32,64,128,256

2. 配置 AHB_CLK 分频作为工作时钟最高可到 72M,支持分频 1,2,4,6,8,10,12,16,32

图 9-2 ADC 时钟



9.3.3 通道选择

每个通道可以配置成规则组和注入组。在任意多个通道上以任意顺序进行的一系列转换。例如，可以如下顺序完成转换：通道 3、通道 8、通道 2、通道 2、通道 0、通道 2、通道 2、通道 15。

- **规则组**由多达 16 个转换组成。规则通道和它们的转换顺序在 ADC_RSEQx 寄存器中选择。规则组中转换的总数应写入 ADC_RSEQ1 寄存器的 LEN[3:0]位中。
- **注入组**由多达 4 个转换组成。注入通道和它们的转换顺序在 ADC_JSEQ 寄存器中选择。注入组里的转换总数目应写入 ADC_JSEQ 寄存器的 LEN[1:0]位中。

不允许在转换期间，ADC_RSEQx 或 ADC_JSEQ 寄存器被更改，只有当 ADC 空闲的时候可以更改 ADC_RSEQx 或 ADC_JSEQ 寄存器。

图 9-3 ADC1 和 ADC2 通道和 Pin 连接关系

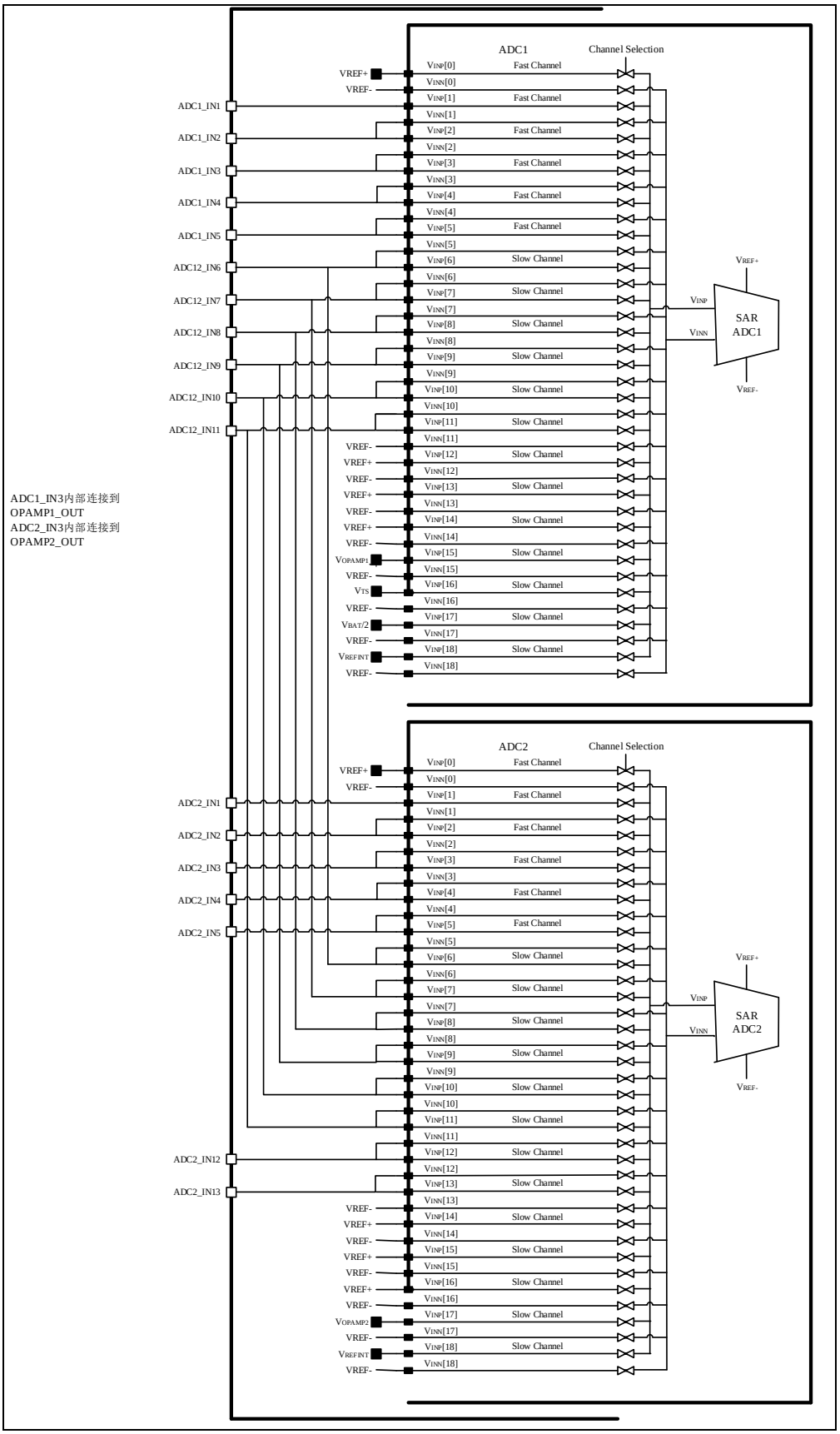
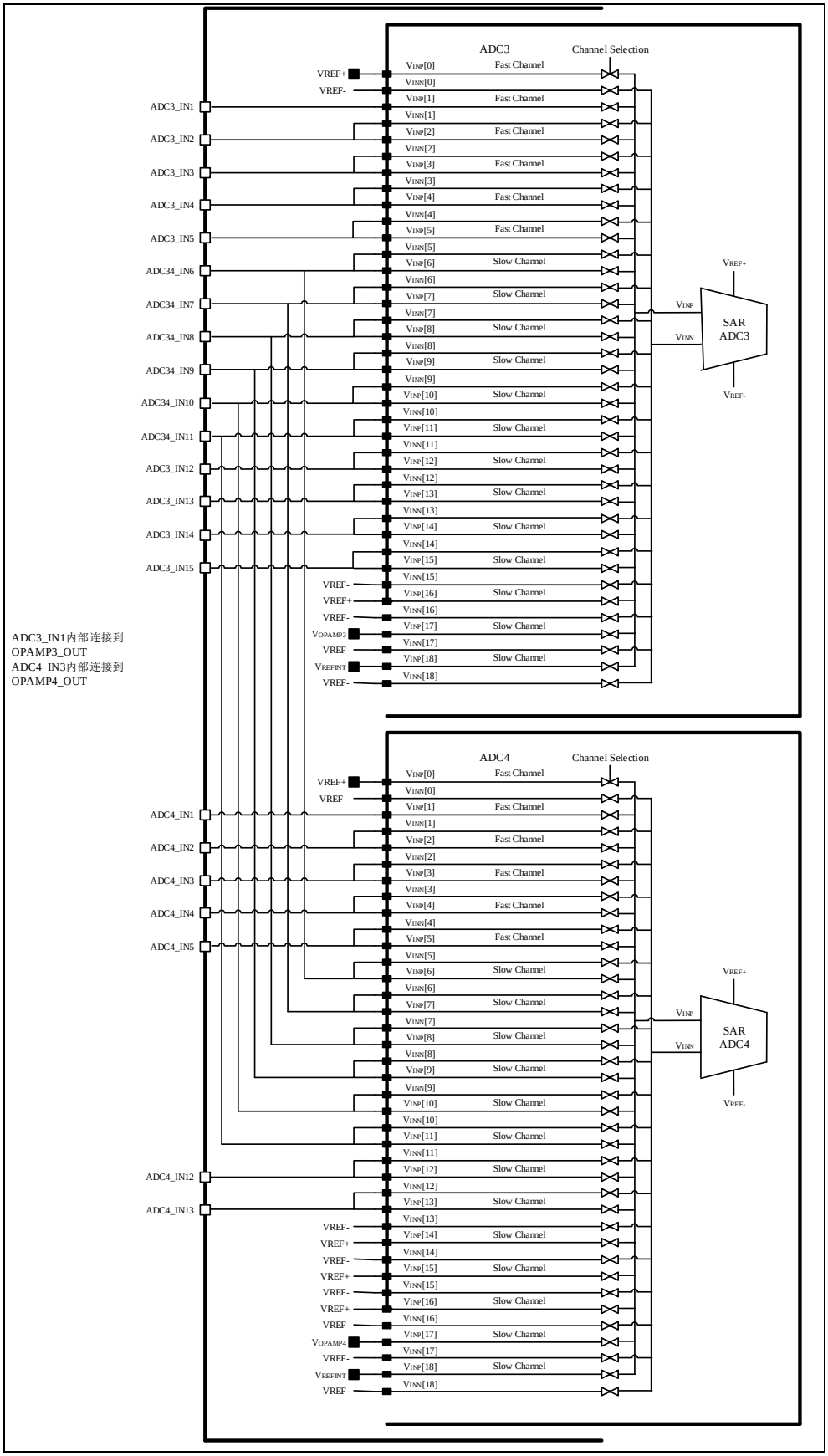


图 9-4 ADC3 和 ADC4 通道和 Pin 连接关系



9.3.4 内部通道

- 温度传感器和通道 ADC1_IN16 相连接
- V_{BAT}/2 和通道 ADC1_IN17 相连接
- 内部参照电压 V_{REFINT} 和 ADCx_IN18 相连接
- V_{OP1OUT} 输出和通道 ADC1_IN3 相连接
- V_{OP2OUT} 输出和通道 ADC2_IN3 相连接
- V_{OP3OUT} 输出和通道 ADC3_IN1 相连接
- V_{OP4OUT} 输出和通道 ADC4_IN3 相连接

可以按注入或规则通道对内部通道进行转换。

注意：温度传感器，VBAT/2 只能出现在主 ADC1 中。

9.3.5 单次转换模式

在单次转换模式下，ADC 只执行一次转换。该模式既可通过设置 ADC_CTRL2 寄存器的 ON 位（只适用于规则通道）启动也可通过外部触发启动（适用于规则通道或注入通道），这时 CTU 位为 0。

一旦选择通道的转换完成:

- 如果一个规则通道被转换:
  - 转换数据被储存在 16 位 ADC_DAT 寄存器中
  - ENDC（转换结束）标志被设置
  - 如果设置了 ENDCIEN，则产生中断。
- 如果一个注入通道被转换:
  - 转换数据被储存在 16 位的 ADC_JDATx 寄存器中
  - JENDC （注入转换结束）标志被设置
  - 如果设置了 JENDCIEN 位，则产生中断,

然后 ADC 停止。

9.3.6 连续转换模式

在连续转换模式中，当前面 ADC 转换一结束马上就启动另一次转换。此模式可通过外部触发启动或通过设置 ADC_CTRL2 寄存器上的 ON 位启动，此时 CTU 位是 1。

每次转换后:

- 如果一个规则通道被转换:
  - 转换数据被储存在 16 位的 ADC_DAT 寄存器中
  - ENDC（转换结束）标志被设置

- 如果设置了 ENDCIEN，则产生中断。

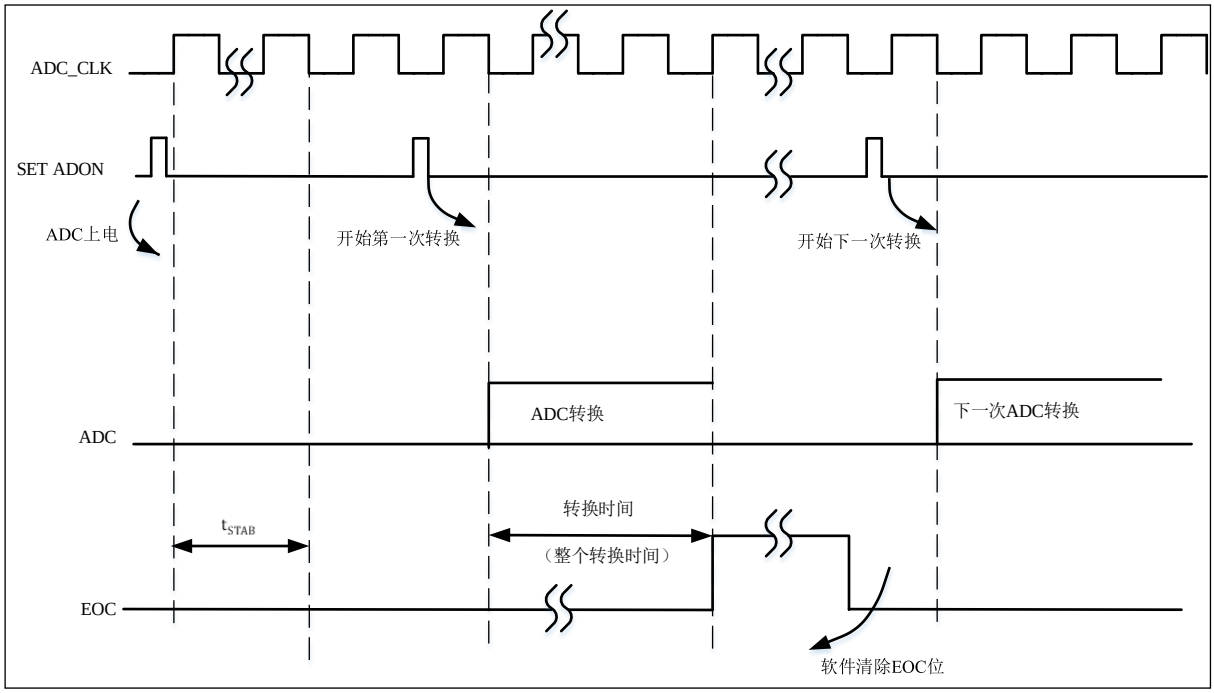
■ 如果一个注入通道被转换：

- 转换数据被储存在 16 位的 ADC_JDATx 寄存器中
- JENDC（注入转换结束）标志被设置
- 如果设置了 JENDCIEN 位，则产生中断。

9.3.7 时序图

如下图所示，ADC 在开始精确转换前需要一个稳定时间 t_{STAB}。在开始 ADC 转换以及 14 个时钟周期后，ENDC 标志被设置，根据配置的转换位数，转换结果存储在 ADC 数据寄存器中。

图 9-5 时序图



9.3.8 模拟看门狗

如果被 ADC 转换的模拟电压值低于低阀值或高于高阀值，AWDG 模拟看门狗状态位被设置。阀值位于 ADC_WDHIGH 和 ADC_WDLOW 寄存器的最低 12 个有效位中。通过设置 ADC_CTRL1 寄存器的 AWDGIEN 位以允许产生相应中断。

阀值独立于由 ADC_CTRL2 寄存器上的 ALIG 位（数据对齐模式）。比较是在对齐之前完成的。在转换位数为 10、8、6 位时，硬件自动忽略数据及阈值的低无效位，只对数据及阈值的高有效位进行比较。阀值固定配置于阈值寄存器的通过配置 ADC_CTRL1 寄存器，模拟看门狗可以作用于 1 个或多个通道上，如表 9-2 所示。

图 9-6 模拟看门狗警戒区

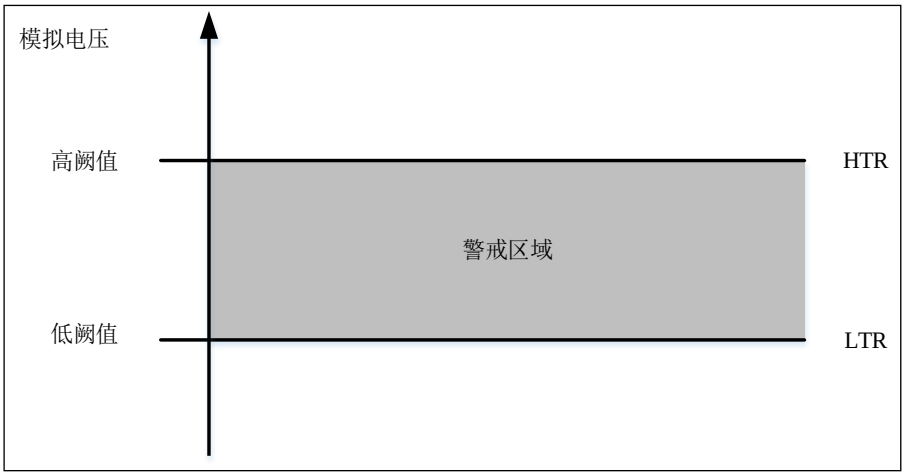


表 9-2 模拟看门狗通道选择

模拟看门狗警戒的通道	ADC_CTRL1寄存器控制位		
	AWDGSGLEN位	AWDGERCH位	AWDGEJCH位
无	任意值	0	0
所有注入通道	0	0	1
所有规则通道	0	1	0
所有注入和规则通道	0	1	1
单一的(1)注入通道	1	0	1
单一的(1)规则通道	1	1	0
单一的(1)注入或规则通道	1	1	1

（1） 由 AWDGCH[4:0]位选择

9.3.9 扫描模式

扫描模式用来扫描一组预先配置的模拟通道。通过设置 ADC_CTRL1 寄存器的 SCANMD 位来选择。一旦这个位被设置，ADC 扫描所有被 ADC_RSEQX 寄存器（对规则通道）或 ADC_JSEQ（对注入通道）选中的所有通道。在每个组的每个通道上执行单次转换。在每个转换结束时，同一组的下一个通道被自动转换。如果设置了 CTU 位，转换不会在选择组的最后一个通道上停止，而是再次从选择组的第一个通道继续转换。

ADC1，ADC3 支持 DMA 配置，如果设置了 DMA 位，在每次 ENDC 后，DMA 控制器把规则组通道的转换数据传输到 SRAM 中。而注入通道转换的数据总是存储在 ADC_JDATx 寄存器中。

注：双 ADC 模式时，ADC2 的规则通道上 DMA 功能需要通过 ADC1 的 DMA 来完成，ADC4 的规则通道上 DMA 功能需要通过 ADC3 的 DMA 来完成

9.3.10 注入通道管理

9.3.10.1 触发注入

ADC_CTRL1 寄存器的 AUTOJC 位清除，并且设置 SCANMD 位，即可使用触发注入功能。

- 利用外部触发或通过设置 ADC_CTRL2 寄存器的 ON 位，启动一组规则通道的转换。
- 如果在规则通道转换期间产生一外部注入触发，当前转换被复位，注入通道序列被以单次扫描方式进行转换。
- 然后，恢复上次被中断的规则组通道转换。如果在注入转换期间产生一规则事件，注入转换不会被中断，但是规则序列将在注入序列结束后被执行。图 9-7 是其定时图。

注意：当使用触发的注入转换时，必须保证触发事件的间隔长于注入序列。例如：序列长度为 28 个 ADC 时钟周期（即 2 个具有 1.5 个时钟间隔采样时间的转换），触发之间最小的间隔必须是 29 个 ADC 时钟周期。

9.3.10.2 自动注入

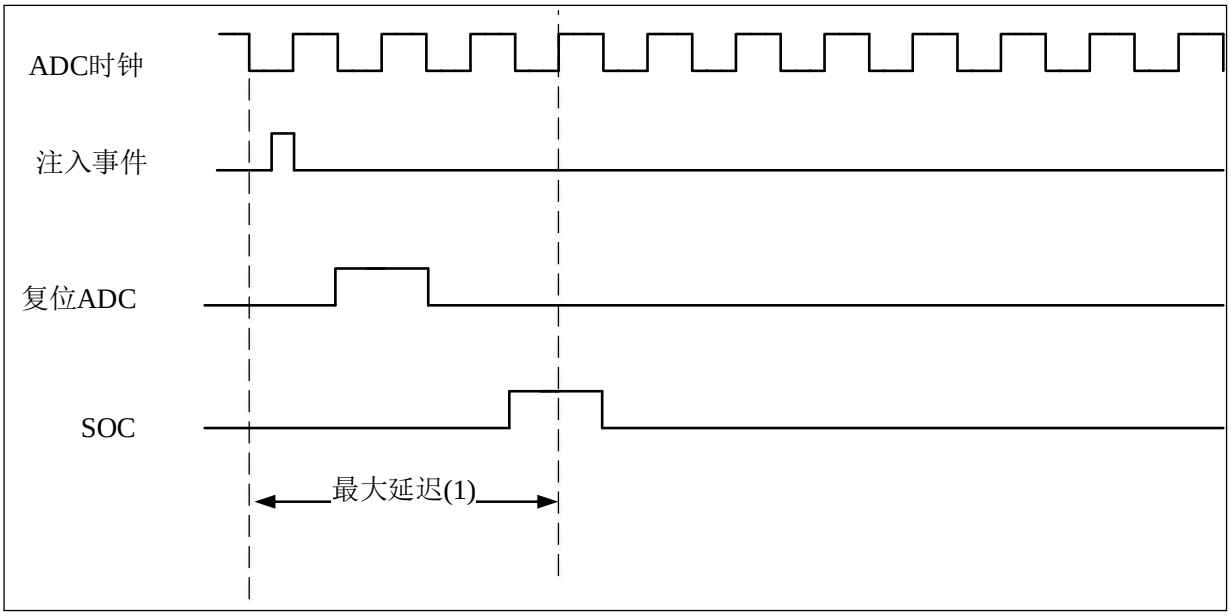
如果设置了 AUTOJC 位，在规则组通道之后，注入组通道被自动转换。这可以用来转换在 ADC_RSEQx 和 ADC_JSEQ 寄存器中设置的多至 20（16+4）个转换序列。在此模式里，必须禁止注入通道的外部触发。

如果除 AUTOJC 位外还设置了 CTU 位，规则通道至注入通道的转换序列被连续执行。

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

注意：不能同时使用自动注入和间断模式。

图 9-7 注入转换延时



（1） 最大延迟数值请参考数据手册中有关电气特性部分。

9.3.11 间断模式

9.3.11.1 规则组

此模式通过设置 ADC_CTRL1 寄存器上的 DREGCH 位激活。对配置的规则组的通道数一次至多允许执行的通道数 n（n<=8），此转换是 ADC_RSEQx 寄存器所选择的转换序列的一部分。数值 n 由 ADC_CTRL1 寄存器的 DCTU[2:0]位给出。一个外部触发信号可以启动 ADC_RSEQx 寄存器中描述的下一轮 n 次转换，直到此序列所有的转换完成为止。总的序列长度由 ADC_RSEQx 寄存器的 LEN[3:0]定义。

例子：

n=3，被转换的通道 = 0、1、2、3、6、7、9、10

第一次触发：转换的序列为 0、1、2

第二次触发：转换的序列为 3、6、7

第三次触发：转换的序列为 9、10，并产生 ENDC 事件

第四次触发：转换的序列 0、1、2

注意：当以间断模式转换一个规则组时，转换序列结束后不自动从头开始。当所有子组被转换完成，下一次触发启动第一个子组的转换。在上面的例子中，第四次触发重新转换第一子组的通道 0、1 和 2。

9.3.11.2 注入组

此模式通过设置 ADC_CTRL1 寄存器的 DJCH 位激活。在一个外部触发事件后，该模式按通道顺序逐个转换 ADC_JSQR 寄存器中选择的序列。

一个外部触发信号可以启动 ADC_JSQR 寄存器选择的下一个通道序列的转换，直到序列中所有的转换完成为止。总的序列长度由 ADC_JSQR 寄存器的 JLEN[1:0]位定义。

例子：

n=1，被转换的通道 = 1、2、3

第一次触发：通道 1 被转换

第二次触发：通道 2 被转换

第三次触发：通道 3 被转换，并且产生 ENDC 和 JENDC 事件第四次触发：通道 1 被转换

注意：

1. *当完成所有注入通道转换，下个触发启动第 1 个注入通道的转换。在上述例子中，第四个触发重新转换第 1 个注入通道 1。*
2. *不能同时使用自动注入和间断模式。*
3. *必须避免同时为规则和注入组设置间断模式。间断模式只能作用于一组转换。*

9.4 校准

每个 ADC 都有一个内置的自校准模式。校准可大幅减小因内部电容器组的变化而造成的准精度误差。在校准期间，在每个电容器上都会计算出一个误差修正码（数字值），这个码用于消除在随后的转

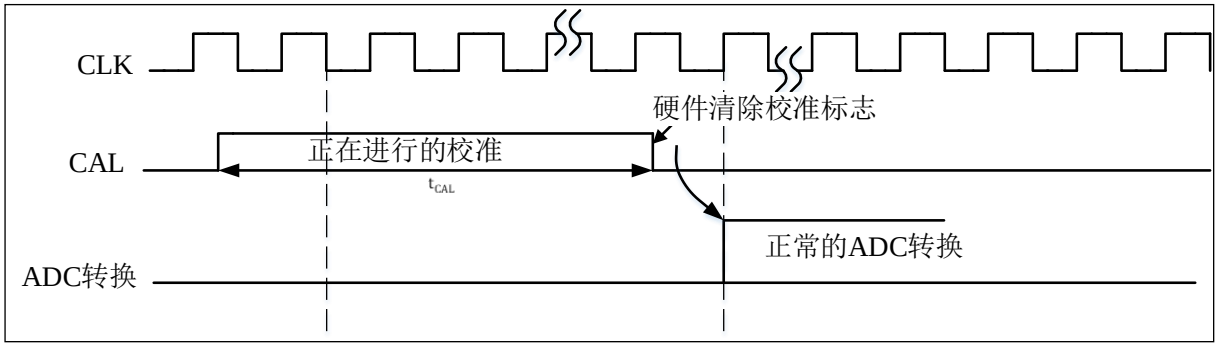
换中每个电容器上产生的误差。

通过设置 ADC_CTRL2 寄存器的 ENCAL 位启动校准。一旦校准结束，ENCAL 位被硬件复位，可以开始正常转换。建议在上电时执行一次 ADC 校准。

注意:

1. *建议在每次上电后执行一次校准。*
2. *默认为单端的校正，需要进行双端的自动校正，必须把寄存器 ADC_CTRL3 的 CALDIF 置 1。然后对 ADC_CTRL2 寄存器的 ENCAL 写 1 ， 然后等待校正完成（校正完成 ENCAL 会自动清 0)。*
3. *在既使用单端模式又使用差分模式的场景下，需打开* ADC_CTRL3 的 CALALD 位启动校准自动载入功能，在每次转换时自动载入对应模式的校准值。

图 9-8 校准时序图



9.5 数据对齐

ADC_CTRL2 寄存器中的 ALIG 位选择转换后数据储存的对齐方式。数据可以左对齐或右对齐，如图 9-9 和图 9-10 所示。

注入组通道转换的数据值已经减去了在 ADC_JOFFSETx 寄存器中定义的偏移量，因此结果可以是一个负值。SEXT 位是扩展的符号值。对于规则组通道，不需减去偏移值，因此只有 12 个位有效。

图 9-9 数据右对齐

注入组

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

规则组

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

图 9-10 数据左对齐

注入组

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

规则组

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

注意：转换位数为 10、8、6 位时参考转换位数为 12 位的对齐方式；
转换位数为 10、8、6 位左对齐时低位无效数据位值为 0。

9.6 可编程的通道采样时间

ADC 使用若干个 ADC_CLK 周期对输入电压采样，采样周期数目可以通过 ADC_SAMPT1 和 ADC_SAMPT2 寄存器中的 SAMP[2:0]位更改。每个通道可以分别用不同的时间采样。总转换时间如下计算：

T_{CONV} = 采样时间+ 12.5 个周期

例如：

当 ADCCLK=14MHz，采样时间为 1.5 周期
T_{CONV} = 1.5 + 12.5 = 14 周期 = 1μs

9.7 外部触发转换

转换可以由外部事件进行触发（例如定时器捕获，EXTI 线）。如果设置了 EXTRTRIG 或 EXTJTRIG 控制位，则外部事件就能够触发转换。EXTRSEL[2:0]和 EXTJSELL2:0]控制位允许应用程序选择 8 个可能的事件中的某一个，分别触发规则和注入组的采样。

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

表 9-3 ADC1 和 ADC2 用于规则通道的外部触发

触发源	类型	EXTRSEL[2:0]
TIM1_CC1事件	来自片上定时器的内部信号	000
TIM1_CC2事件		001
TIM1_CC3事件		010
TIM2_CC2事件		011
TIM3_TRGO事件		100
TIM4_CC4事件		101
EXTI线11/TIM8_TRGO事件(1)	外部引脚/来自片上定时器的内部信号	110
SWSTRRCH	软件控制位	111

1．对于规则通道，选中 EXTI 线路 11 或 TIM8_TRGO 作为外部触发事件，可以分别通过设置 AFIO_RMP_CFG 中的 ADC1_ETRR 位和 ADC2_ETRR 位实现。

表 9-4 ADC1 和 ADC2 用于注入通道的外部触发

触发源	连接类型	EXTJSEL[2:0]
TIM1_TRGO事件	来自片上定器的内部信号	000
TIM1_CC4事件		001
TIM2_TRGO事件		010
TIM2_CC1事件		011
TIM3_CC4事件		100

触发源	连接类型	EXTJSEL[2:0]
TIM4_TRGO事件		101
EXTI线15/TIM8_CC4事件(1)	外部引脚/来自片上定时器的内部信号	110
SWSTRJCH	软件控制位	111

1．对于注入通道，选中 EXTI 线路 15 和 TIM8_CC4 作为外部触发事件，可以分别通过设置 AFIO_RMP_CFG 中的 ADC1_ETRI 位和 ADC2_ETRI 位实现。

表 9-5 ADC3 和 ADC4 用于规则通道的外部触发

触发源	连接类型	EXTRSEL [2:0]
TIM3_CC1事件	来自片上定时器的内部信号	000
TIM2_CC3事件		001
TIM1_CC3事件		010
TIM8_CC1事件		011
TIM8_TRGO事件		100
TIM5_CC1事件		101
EXTI线10/TIM5_CC3事件		110
SWSTRRCH	软件控制位	111

1．对于规则通道，选中 EXTI 线路 10 或 TIM5_CC3 作为外部触发事件，可以分别通过设置 AFIO_RMP_CFG4 中的 ADC3_ETRR 位和 ADC4_ETRR 位实现。

表 9-6 ADC3 和 ADC4 用于注入通道的外部触发

触发源	连接类型	EXTJSEL[2:0]
TIM1_TRGO事件	来自片上定时器的内部信号	000
TIM1_CC4事件		001
TIM4_CC3事件		010
TIM8_CC2事件		011
TIM8_CC4事件		100
TIM5_TRGO事件		101
EXTI线14/TIM5_CC4事件		110
SWSTRJCH	软件控制位	111

1．对于注入通道，选中 EXTI 线路 14 或 TIM5_CC4 作为外部触发事件，可以分别通过设置 AFIO_RMP_CFG4 中的 ADC3_ETRI 位和 ADC4_ETRI 位实现。

软件触发事件可以通过对寄存器 ADC_CTRL2 的 SWSTRRCH 或 SWSTRJCH 位置‘1’产生。

规则组的转换可以被注入触发打断。

9.8 DMA 请求

因为规则通道转换的值储存在一个仅有的数据寄存器中，所以当转换多个规则通道时需要使用 DMA，这可以避免丢失已经存储在 ADC_DAT 寄存器中的数据。

只有在规则通道的转换结束时才产生 DMA 请求，并将转换的数据从 ADC_DAT 寄存器传输到用户指定的目的地址。

注意：独立 ADC 模式时，ADC1，ADC2，ADC3，ADC4 拥有 DMA 功能。双 ADC 模式时，由 ADC2 转化的数据在 ADC1 的数据寄存器中，由 ADC4 转化的数据在 ADC3 的数据寄存器中。

9.9 双 ADC 模式

ADC1（主）和 ADC2（从），ADC3（主）和 ADC4（从）可以组成双 ADC 模式（见图 9-11 双 ADC 框图）。

在双 ADC 模式里，根据 ADC1_CTRL1 寄存器中 DUSEL[2:0]位所选的模式，转换的启动可以是 ADC1 主和 ADC2 从的交替触发或同步触发。

注意：在双 ADC 模式里，当转换配置成由外部事件触发时，用户必须将其设置成仅触发主 ADC，从 ADC 设置成软件触发，这样可以防止意外的触发从转换。但是，主和从 ADC 的外部触发必须同时被激活。ADC3 和 ADC4 为一组，同 ADC1 和 ADC2 行为一致，本节描述的功能适用于 ADC3 和 ADC4.

共有 6 种可能的模式：

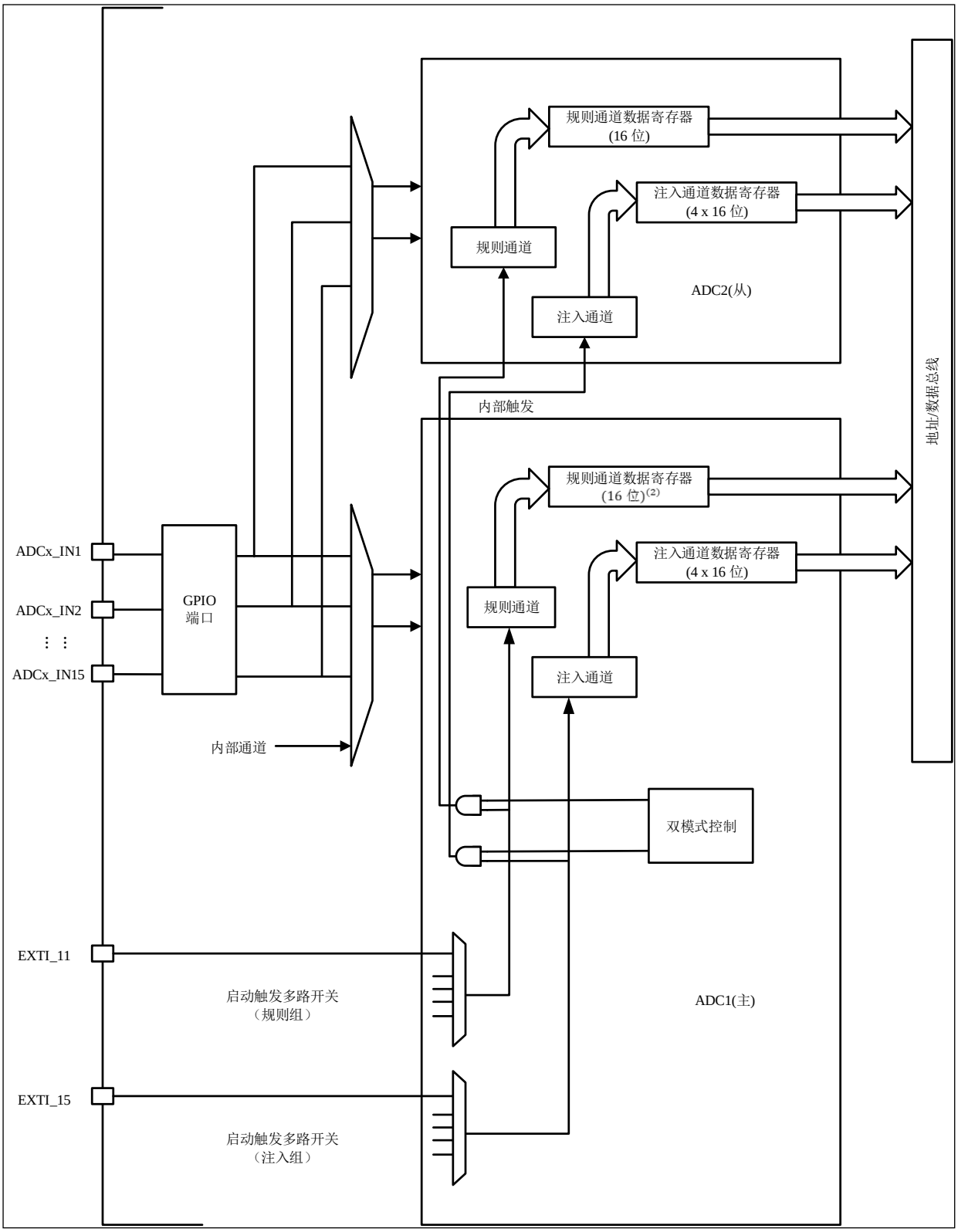
- 同步注入模式
- 同步规则模式
- 快速交叉模式
- 慢速交叉模式
- 交替触发模式
- 独立模式

还有可以用下列方式组合使用上面的模式：

- 同步注入模式 + 同步规则模式
- 同步规则模式 + 交替触发模式
- 同步注入模式 + 交叉模式

注意：在双 ADC 模式里，为了在主数据寄存器上读取从转换数据，必须使能 DMA 位，即使不使用 DMA 传输规则通道数据,从 ADC 不能使能 DMA。

图 9-11 双 ADC 框图(1)



1．外部触发信号作用于 ADC2，但在本图中没有显示。

2．在双 ADC 模式中，在完整的 ADC1 数据寄存器（ADC1_DAT）中包含了 ADC1 和 ADC2 的规则转换数据。

3．ADC3 和 ADC4 的逻辑关系等同 ADC1 和 ADC2

9.9.1 同步注入模式

此模式转换一个注入通道组。外部触发来自 ADC1 的注入组多路开关（由 ADC1_CTRL2 寄存器的 EXTJSEL[2:0]选择），它同时给 ADC2 提供同步触发。

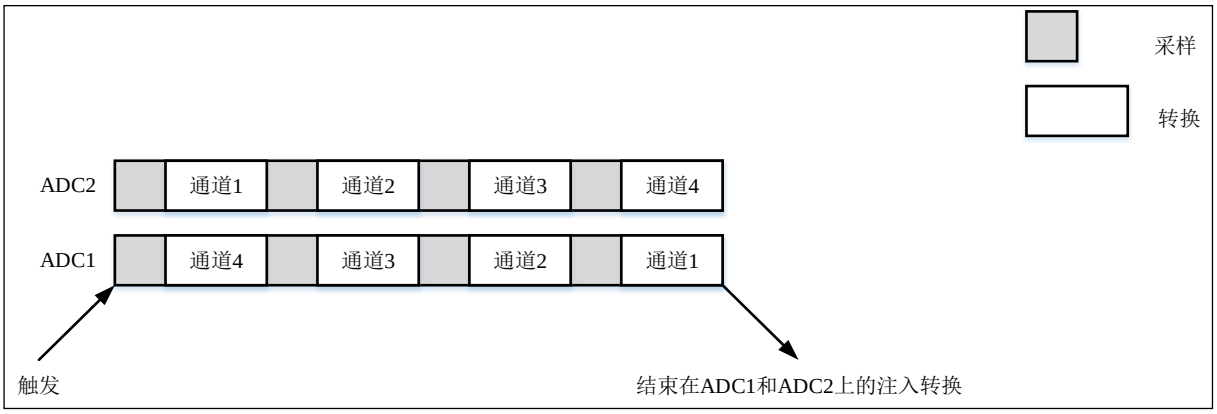
注意：不要在 2 个 ADC 上转换相同的通道（两个 ADC 在同一个通道上的采样时间不能重叠）。

在 ADC1 或 ADC2 的转换结束时：

- 转换的数据存储在每个 ADC 接口的 ADC_JDATx 寄存器中。
- 当所有 ADC1/ADC2 注入通道都被转换时，产生 JENDC 中断（若任一 ADC 接口开放了中断）。

注：在同步模式中，必须转换具有相同时间长度的序列，或保证触发的间隔比 2 个序列中较长的序列长，否则当较长序列的转换还未完成时，具有较短序列的 ADC 转换可能会被重启。

图 9-12 在 4 个通道上的同步注入模式



9.9.2 同步规则模式

此模式在规则通道组上执行。外部触发来自 ADC1 的规则组多路开关（由 ADC1_CTRL2 寄存器的 EXTRSEL[2:0]选择），它同时给 ADC2 提供同步触发。

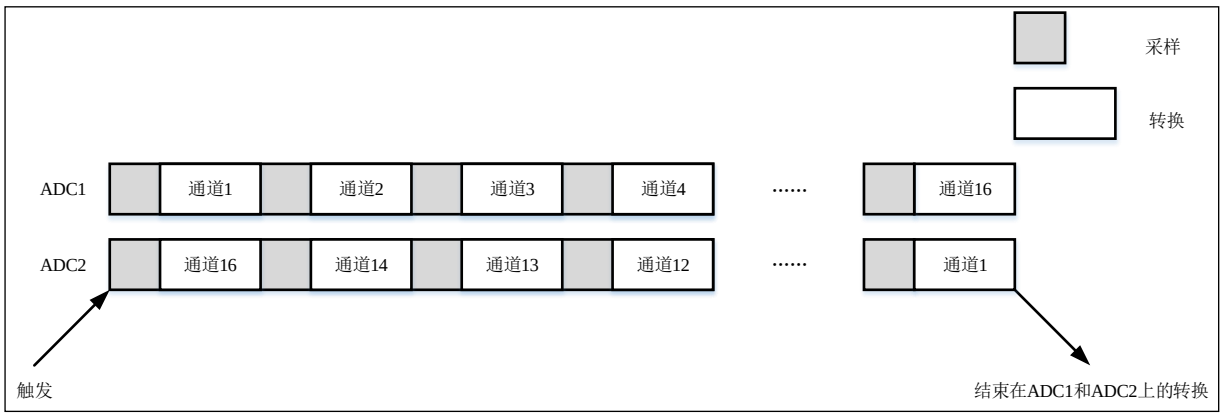
注意：不要在 2 个 ADC 上转换相同的通道（两个 ADC 在同一个通道上的采样时间不能重叠）。

在 ADC1 或 ADC2 的转换结束时：

- 产生一个 32 位 DMA 传输请求(如果设置了 DMA 位)，32 位的 ADC1_DAT 寄存器内容传输到 SRAM 中，它高半个字包含 ADC2 的转换数据，低半个字包含 ADC1 的转换数据。
- 当所有 ADC1/ADC2 规则通道都被转换完时，产生 ENDC 中断（若任一 ADC 接口开放了中断）。

注：在同步规则模式中，必须转换具有相同时间长度的序列，或保证触发的间隔比 2 个序列中较长的序列长，否则当较长序列的转换还未完成时，具有较短序列的 ADC 转换可能会被重启。

图 9-13 在 16 个通道上的同步规则模式



9.9.3 快速交叉模式

此模式只适用于规则通道组（通常为一个通道）。外部触发来自 ADC1 的规则通道多路开关。外部触发产生后:

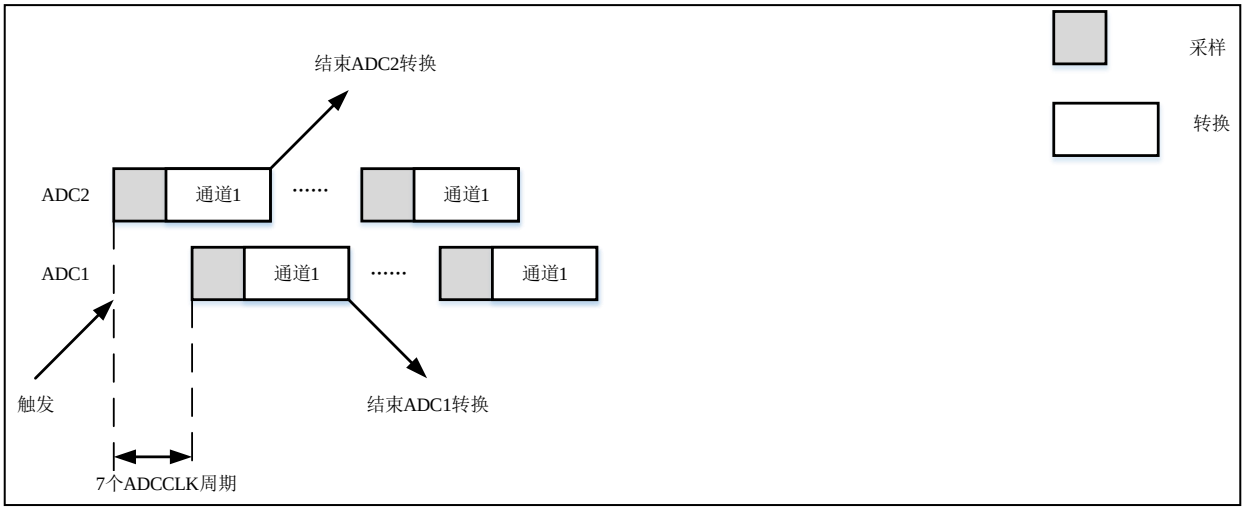
- ADC2 立即启动
- ADC1 在延迟 7 个 ADC 时钟周期后启动

如果同时设置了 ADC1 和 ADC2 的 CTU 位，所选的两个 ADC 规则通道将被连续地转换。

ADC1 产生一个 ENDC 中断后（由 ENDCIEN 使能），产生一个 32 位的 DMA 传输请求（如果设置了 DMA 位），ADC1_DAT 寄存器的 32 位数据被传输到 SRAM，ADC1_DAT 的上半个字包含 ADC2 的转换数据，低半个字包含 ADC1 的转换数据。

注意：最大允许采样时间<7 个 ADCCLK 周期，避免 ADC1 和 ADC2 转换相同通道时发生两个采样周期的重叠。

图 9-14 在 1 个通道上连续转换模式下的快速交叉模式



9.9.4 慢速交叉模式

此模式只适用于规则通道组（只能为一个通道）。外部触发来自 ADC1 的规则通道多路开关。外部触发产生后:

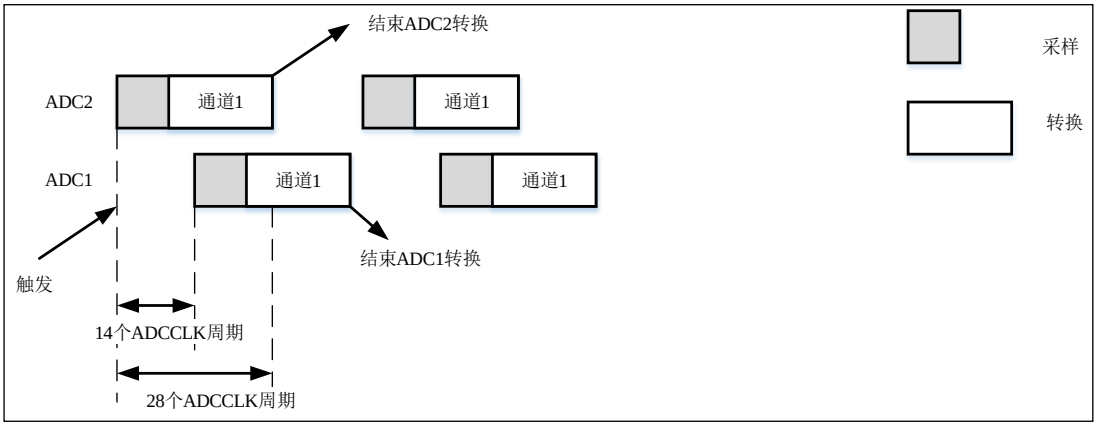
- ADC2 立即启动
- ADC1 在延迟 14 个 ADC 时钟周期后启动
- 在延迟第二次 14 个 ADC 周期后 ADC2 再次启动，如此循环。

注意：最大允许采样时间<14 个 ADCCLK 周期，以避免和下个转换重叠。

ADC1 产生一个 ENDC 中断后（由 ENDCIEN 使能），产生一个 32 位的 DMA 传输请求（如果设置了 DMA 位），ADC1_DAT 寄存器的 32 位数据被传输到 SRAM，ADC1_DAT 的上半个字包含 ADC2 的转换数据，低半个字包含 ADC1 的转换数据。在 28 个 ADC 时钟周期后自动启动新的 ADC2 转换。在这个模式下不能设置 CTU 位，因为它将连续转换所选择的规则通道。

注意：应用程序必须确保当使用交叉模式时，不能有注入通道的外部触发产生。

图 9-15 在 1 个通道上的慢速交叉模式



9.9.5 交替触发模式

此模式只适用于注入通道组。外部触发源来自 ADC1 的注入通道多路开关。

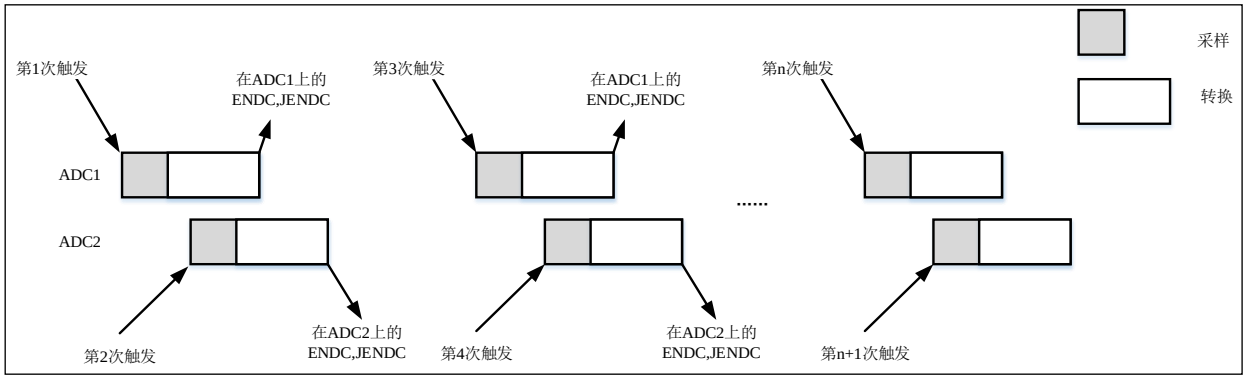
- 当第一个触发产生时，ADC1 上的所有注入组通道被转换。
- 当第二个触发到达时，ADC2 上的所有注入组通道被转换。
- 如此循环……

如果允许产生 JENDC 中断，在所有 ADC1 注入组通道转换后产生一个 JENDC 中断。

如果允许产生 JENDC 中断，在所有 ADC2 注入组通道转换后产生一个 JENDC 中断。

当所有注入组通道都转换完后，如果又有另一个外部触发，交替触发处理从转换 ADC1 注入组通道重新开始。

图 9-16 交替触发：每个 ADC1 的注入通道组



如果 ADC1 和 ADC2 上同时使用了注入间断模式：

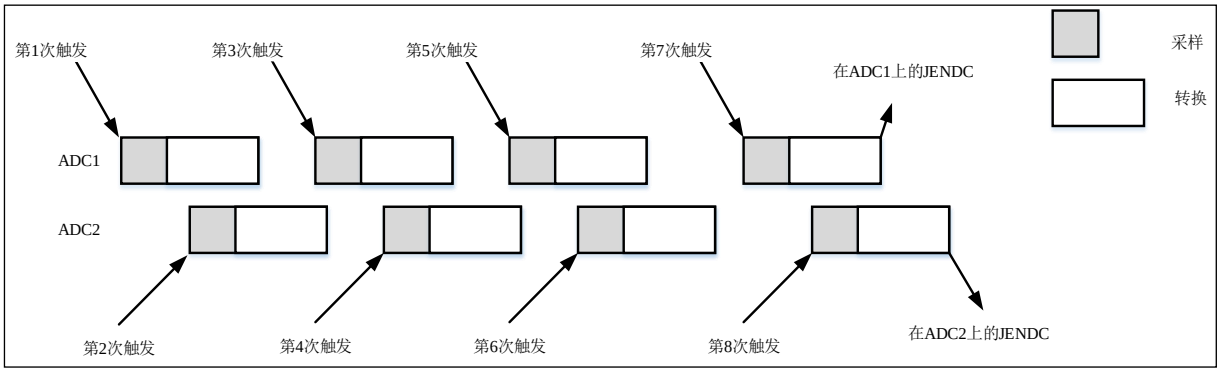
- 当第一个触发产生时，ADC1 上的第一个注入通道被转换。
- 当第二个触发到达时，ADC2 上的第一个注入通道被转换。
- 如此循环……

如果允许产生 JENDC 中断，在所有 ADC1 注入组通道转换后产生一个 JENDC 中断。

如果允许产生 JENDC 中断，在所有 ADC2 注入组通道转换后产生一个 JENDC 中断。

当所有注入组通道都转换完后，如果又有另一个外部触发，则重新开始交替触发过程。

图 9-17 交替触发：在间断模式下每个 ADC 上的 4 个注入通道



9.9.6 独立模式

此模式里，双 ADC 同步不工作，每个 ADC 接口独立工作。

9.9.7 混合的规则/注入同步模式

规则组同步转换可以被中断，以启动注入组的同步转换。

注：在混合的规则/注入同步模式中，必须转换具有相同时间长度的序列，或保证触发的间隔比 2 个序列中较长的序列长，否则当较长序列的转换还未完成时，具有较短序列的 ADC 转换可能会被重启。

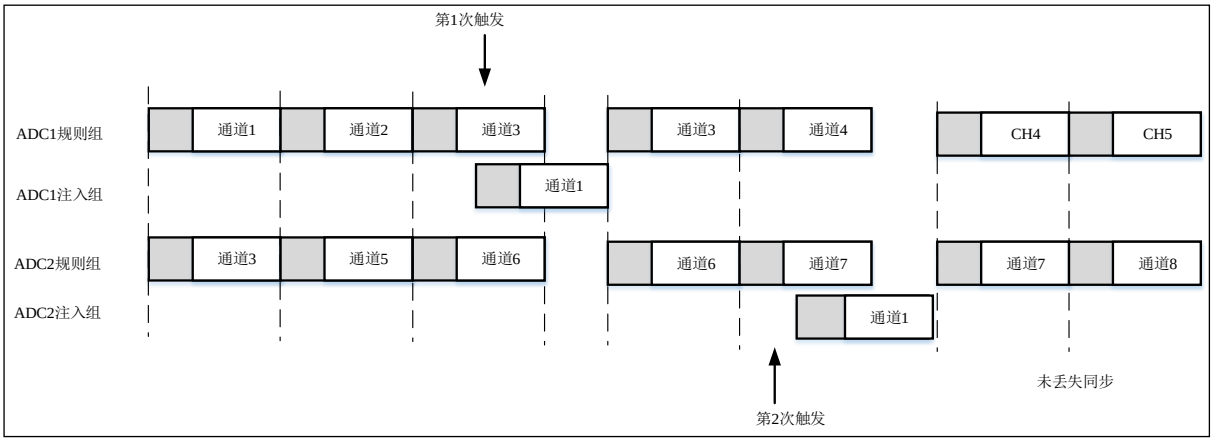
9.9.8 混合的同步规则+交替触发模式

规则组同步转换可以被中断，以启动注入组交替触发转换。图 9-18 显示了一个规则同步转换被交替触发所中断。

注入交替转换在注入事件到达后立即启动。如果规则转换已经在运行，为了在注入转换后确保同步，所有的 ADC（主和从）的规则转换被停止，并在注入转换结束时同步恢复。

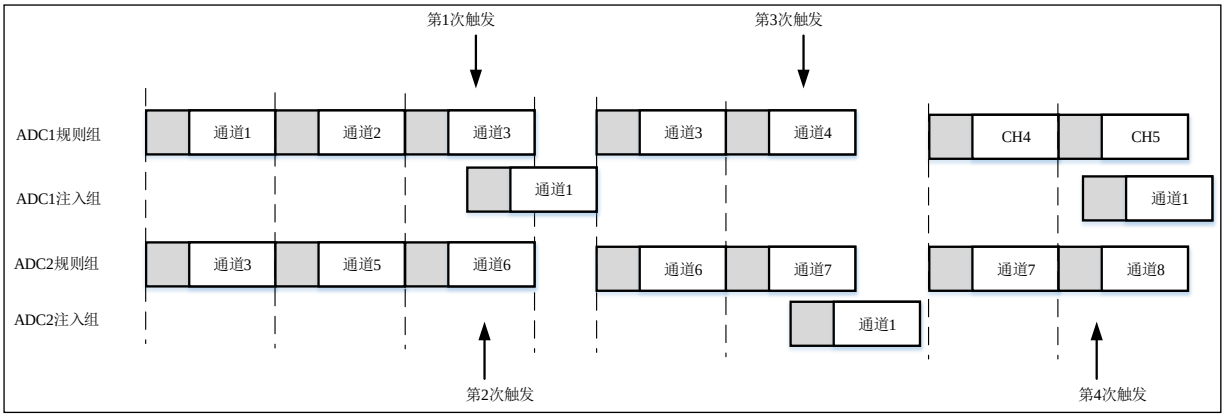
注：在混合的同步规则+交替触发模式中，必须转换具有相同时间长度的序列，或保证触发的间隔比 2 个序列中较长的序列长，否则当较长序列的转换还未完成时，具有较短序列的 ADC 转换可能会被重启。

图 9-18 交替+规则同步



如果触发事件发生在一个中断了规则转换的注入转换期间，这个触发事件将被忽略。下图示出了这种情况的操作（第 2 个触发被忽略）。

图 9-19 触发事件发生在注入转换期间

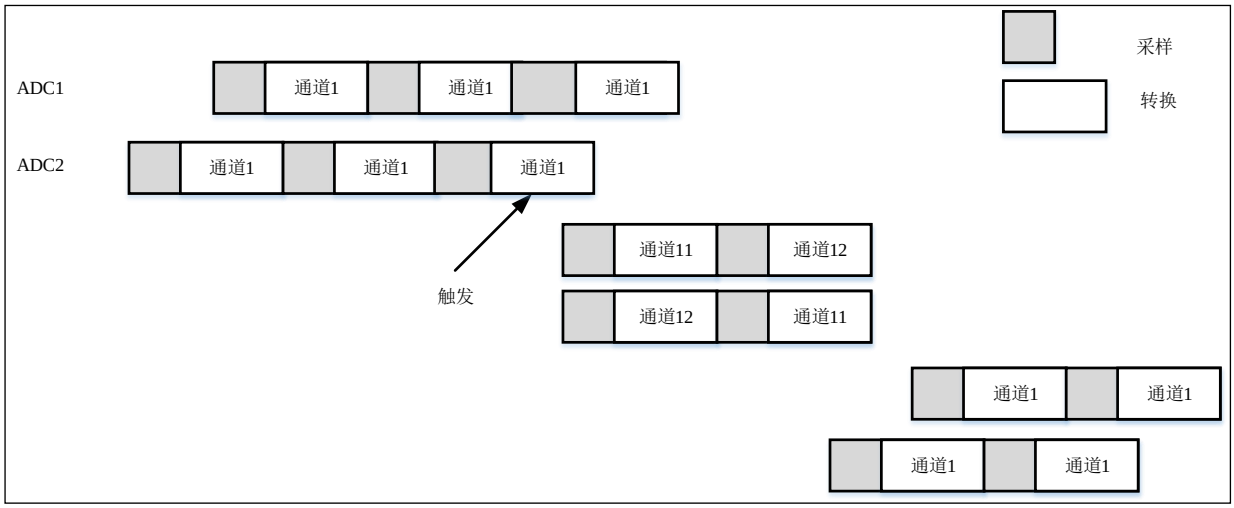


9.9.9 混合同步注入 + 交叉模式

一个注入事件可以中断一个交叉转换。这种情况下，交叉转换被中断，注入转换被启动，在注入序列转换结束时，交叉转换被恢复。下图是这种情况的一个例子。

注：当 ADC 时钟预分频系数设置为 4 时，交叉模式恢复后不会均匀地分配采样时间，采样间隔是 8 个 ADC 时钟周期与 6 个 ADC 时钟周期轮替，而不是均匀的 7 个 ADC 时钟周期。

图 9-20 交叉的单通道转换被注入序列 CH11 和 CH12 中断



9.10 温度传感器

温度传感器可以用来测量器件周围的温度（TA）。温度传感器在内部和 ADC1_IN16 输入通道相连接，此通道把传感器输出的电压转换成数字值。温度传感器模拟输入推荐采样时间是 17.1μs。当没有被使用时，传感器可以被关闭以降低功耗。

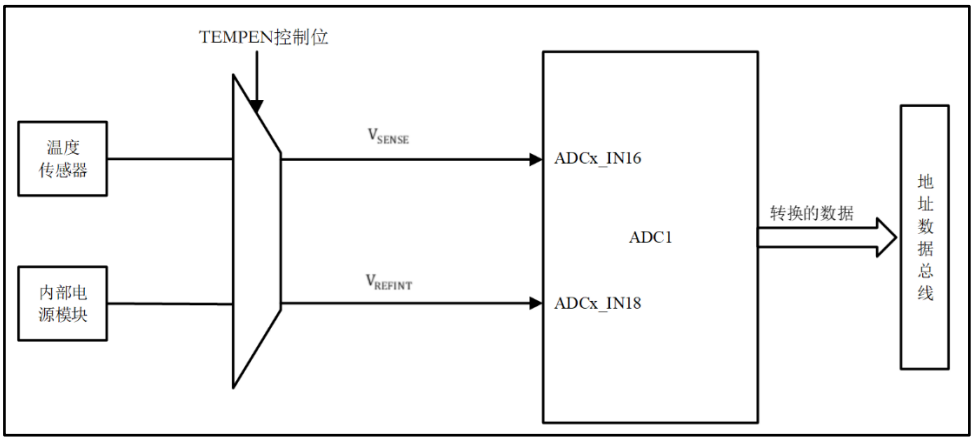
图 9-21 是温度传感器的方框图。

注意：必须设置 TEMPEN 位激活内部通道：ADC1_IN16（温度传感器）和 ADC1_IN18（VREFINT）的转换。

温度传感器输出电压随温度线性变化，由于生产过程的变化，温度变化曲线的偏移在不同芯片上会有不同（最大±3°C）。

内部温度传感器更适合于检测温度的变化，而不是测量绝对的温度。如果需要测量精确的温度，应该使用一个外置的温度传感器。

图 9-21 温度传感器和 V_{REFINT} 通道框图



9.10.1 读温度

使用温度传感器读温度配置如下:

- 选择 ADC1_IN16 输入通道
- 选择采样时间为 17.1 μs
- 设置 ADC 控制寄存器 2（ADC_CTRL2）的 TEMPEN 位，以唤醒关电模式下的温度传感器
- 通过设置 ON 位启动 ADC 转换（或用外部触发）
- 读 ADC 数据寄存器上的 VSENSE 数据结果
- 利用下列公式得出温度

温度（°C） = {(V25 - VSENSE) / Avg_Slope} + 25
这里:
V25 = VSENSE 在 25°C 时的数值
Avg_Slope = 温度与 VSENSE 曲线的平均斜率（单位为 mV/°C 或 μV/°C）

参考数据手册的电气特性章节中 V25 和 Avg_Slope 的实际值。

注意：传感器从关电模式唤醒后到可以输出正确水平的 V_{SENSE} 前，有一个建立时间。ADC 在上电后也有一个建立时间，因此为了缩短延时，应该同时设置 ON 和 TEMPEN 位。

9.11 ADC 中断

规则或注入组转换结束时能产生中断，当模拟看门狗状态位被设置时也能产生中断，任一注入通道转换结束中断 (JENDCAIEN)，任一规则通道转换结束中断 (ENDCAIEN)。它们都有独立的中断使能位。

注：ADC1 和 ADC2 的中断映射在同一个中断向量上，而 ADC3 和 ADC4 映射在同一个中断向量上。

ADC_STS 寄存器中有 2 个其他状态标志，但是没有相关联的中断:

- JSTR（注入组通道转换的启动）
- STR（规则组通道转换的启动）

表 9-7 ADC 中断

中断事件	事件标志	使能控制位
规则组转换结束	ENDC	ENDCIEN
注入组转换结束	JENDC	JENDCIEN
模拟看门狗标志位置位	AWDG	AWDGIEN
任何规则通道转换结束	ENDCA	ENDCAIEN
任何注入通道转换结束	JENDCA	JENDCAIEN

9.12 ADC 寄存器

寄存器描述中使用的一些缩略语请参考 1.1 节。

必须以字（32 位）的方式操作这些外设寄存器。

9.12.1 ADC 寄存器地址映像

下表列出了所有的 ADC 寄存器。

表 9-8 ADC 寄存器映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	ADC_STS	Reserved																									JENDCA	ENDCA	STR	JSTR	JENDC	ENDC	AWDG
	Reset Value																										0	0	0	0	0	0	0
004h	ADC_CTRL1	Reserved								AWDGERCH	AWDGEJCH	Reserved		DUSEL				DCTU			DJCH	DREGCH	AUTOJC	AWDGSGLEN	SCANMD	JENDCIEN	AWDGIEN	ENDCIEN	AWDGCH				
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	ADC_CTRL2	Reserved								TEMPEN	SWSTRRCH	SWSTRJCH	EXTRTRIG	EXTRSEL[2:0]			Reserved	EXTJTRIG	EXTJSEL			ALIG	Reserved		ENDMA	Reserved					ENCAL	CTU	ON
	Reset Value									0	0	0	0	0	0	0		0	0	0	0	0			0						0	0	0
00Ch	ADC_SMPT1	Reserved								SAMP17[2:0]				SAMP16[2:0]				SAMP15[2:0]				SAMP14[2:0]				SAMP13[2:0]				SAMP12[2:0]				SAMP11[2:0]				SAMP10[2:0]			
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	ADC_SMPT2	Reserved		SAMP9[2:0]				SAMP8[2:0]				SAMP7[2:0]				SAMP6[2:0]				SAMP5[2:0]				SAMP4[2:0]				SAMP3[2:0]				SAMP2[2:0]				SAMP1[2:0]				SAMP0[2:0]			
	Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	ADC_JOFFSET1	Reserved																				OFFSETJCH1[11:0]													
	Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	ADC_JOFFSET2	Reserved																				OFFSETJCH2[11:0]													
	Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	ADC_JOFFSET3	Reserved																				OFFSETJCH3[11:0]													
	Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0	0	0
020h	ADC_JOFFSET4	Reserved																				OFFSETJCH4[11:0]													
	Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0	0	0
024h	ADC_WDGHIGH	Reserved																HTH[11:0]																
	Reset Value																	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
028h	ADC_WDGLOW	Reserved																LTH[11:0]																
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	ADC_RSEQ1	Reserved								LEN[3:0]				SEQ16[4:0]				SEQ15[4:0]				SEQ14[4:0]								SEQ13[4:0]			
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
030h	ADC_RSEQ2	Reserved		SEQ12[4:0]						SEQ11[4:0]				SEQ10[4:0]				SEQ9[4:0]				SEQ8[4:0]								SEQ7[4:0]			
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
034h	ADC_RSEQ3	Reserved		SEQ6[4:0]						SEQ5[4:0]				SEQ4[4:0]				SEQ3[4:0]				SEQ2[4:0]								SEQ1[4:0]			
	Reset Value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
038h	ADC_JSEQ	Reserved										JLEN[1:0]		JSEQ4[4:0]				JSEQ3[4:0]				JSEQ2[4:0]								JSEQ1[4:0]			
	Reset Value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03Ch	ADC_JDAT1	Reserved																JDAT1[15:0]																
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
040h	ADC_JDAT2	Reserved																JDAT2[15:0]																
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
044h	ADC_JDAT3	Reserved																JDAT3[15:0]																
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
048h	ADC_JDAT4	Reserved																JDAT4[15:0]																
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04Ch	ADC_DAT	ADC2DAT[15:0]																DAT[15:0]																
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
050h	ADC_DIFSEL	Reserved													DIFSEL[18:0]																		Reserved
	Reset Value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
054h	ADC_CALFACT	Reserved									CALFACTD[6:0]							Reserved									CALFACTS[6:0]							
	Reset Value										0	0	0	0	0	0	0										0	0	0	0	0	0	0
058h	ADC_CTRL3	Reserved																				VBATMEN	DPWMOD	JENDCAIEN	ENDCAIEN	BPCAL	PDRDY	RDY	CKMOD	CALALD	CALDIF	RES[1:0]	
	Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0
05Ch	ADC_SAMPT3	Reserved																												SAMPSEL	SAMP18[2:0]		
	Reset Value																													0	0	0	0

9.12.2 ADC 状态寄存器（ADC_STS）

地址偏移：0x00

复位值：0x0000 0000

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

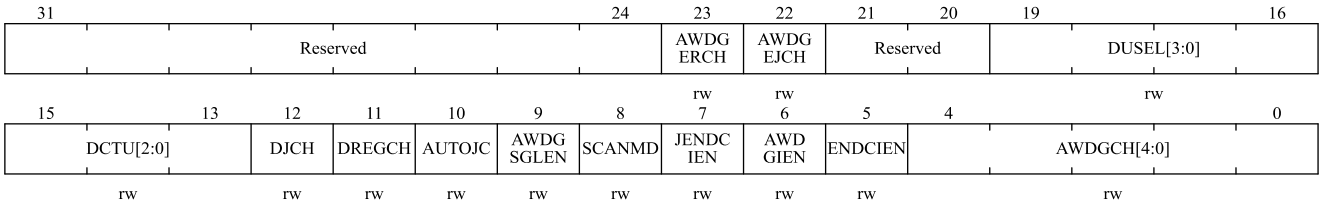
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									JENDCA	ENDCA	STR	JSTR	JENDC	ENDC	AWDG
									rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位域	名称	描述
31:15	Reserved	保留，硬件强制为0。
6	JENDCA	任意注入通道结束位 （Any Injected channel End of conversion flag） 该位由硬件在规则通道转换结束时设置，由软件清除。 0：转换没有完成； 1：转换完成。
5	ENDCA	任意通道结束位 （Any End of conversion flag） 该位由硬件在（规则或注入）任意通道转换结束时设置，由软件清除。 0：转换没有完成； 1：转换完成。
4	STR	规则通道开始位 （Regular channel Start flag） 该位由硬件在规则通道转换开始时设置，由软件清除。0：规则通道转换未开始； 1：规则通道转换已开始。
3	JSTR	注入通道开始位 （Injected channel Start flag） 该位由硬件在注入通道组转换开始时设置，由软件清除。 0：注入通道组转换未开始； 1：注入通道组转换已开始。
2	JENDC	注入通道转换结束位 （Injected channel end of conversion） 该位由硬件在所有注入通道组转换结束时设置，由软件清除 0：转换未完成； 1：转换完成。

位域	名称	描述
1	ENDC	转换结束位 （End of conversion） 该位由硬件在（规则或注入）通道组转换结束时设置，由软件清除或由读取ADC_DAT时清除 0：转换未完成； 1：转换完成。
0	AWDG	模拟看门狗标志位 （Analog watchdog flag） 该位由硬件在转换的电压值超出了ADC_LTR和ADC_HTR寄存器定义的范围时设置，由软件清除 0：没有发生模拟看门狗事件； 1：发生模拟看门狗事件。

9.12.3 ADC 控制寄存器 1（ADC_CTRL1）

地址偏移：0x04
复位值：0x0000 0000



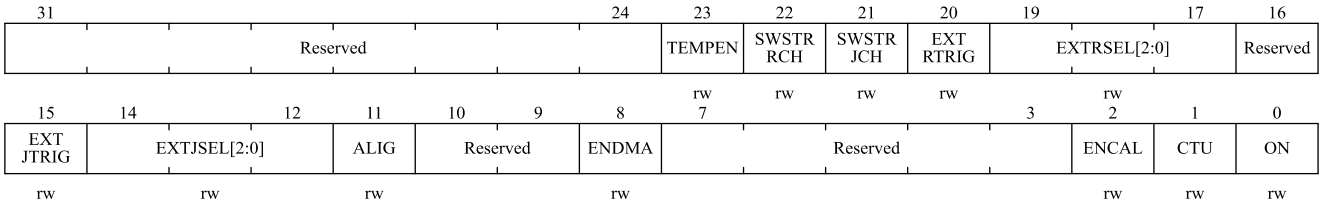
位域	名称	描述
31:24	Reserved	保留，硬件强制为0。
23	AWDGERCH	在规则通道上开启模拟看门狗 （Analog watchdog enable on regular channels） 该位由软件设置和清除。 0：在规则通道上禁用模拟看门狗； 1：在规则通道上使用模拟看门狗。
22	AWDGEJCH	在注入通道上开启模拟看门狗 （Analog watchdog enable on injected channels） 该位由软件设置和清除。 0：在注入通道上禁用模拟看门狗； 1：在注入通道上使用模拟看门狗。
21:20	Reserved	保留，硬件强制为0。

位域	名称	描述
19:16	DUSEL[3:0]	双模式选择 （Dual mode selection） 软件使用这些位选择操作模式。 0000：独立模式 0001：混合的同步规则+注入同步模式 0010：混合的同步规则+交替触发模式 0011：混合同步注入+快速交叉模式 0100：混合同步注入+慢速交叉模式 0101：注入同步模式 0110：规则同步模式 0111：快速交叉模式 1000：慢速交叉模式 1001：交替触发模式 *注： 在ADC2和ADC4中这些位为保留位* *在双模式中，改变通道的配置会产生一个重新开始的条件，这将导致同步丢失。建议在进行任何配置改变前关闭双模式。*
15:13	DCTU[2:0]	间断模式通道计数 （Discontinuous mode channel count） 软件通过这些位定义在间断模式下，收到外部触发后转换规则通道的数目 000：1个通道 001：2个通道 …… 111：8个通道
12	DJCH	在注入通道上的间断模式 （Discontinuous mode on injected channels） 该位由软件设置和清除，用于开启或关闭注入通道组上的间断模式 0：注入通道组上禁用间断模式； 1：注入通道组上使用间断模式。
11	DREGCH	在规则通道上的间断模式 （Discontinuous mode on regular channels） 该位由软件设置和清除，用于开启或关闭规则通道组上的间断模式 0：规则通道组上禁用间断模式； 1：规则通道组上使用间断模式。
10	AUTOJC	自动的注入通道组转换 （Automatic Injected Group conversion） 该位由软件设置和清除，用于开启或关闭规则通道组转换结束后自动的注入通道组转换 0：关闭自动的注入通道组转换； 1：开启自动的注入通道组转换。
9	AWDGSGLEN	扫描模式中在一个单一的通道上使用看门狗 （Enable the watchdog on a single channel in scan mode） 该位由软件设置和清除，用于开启或关闭由AWDGCH[4:0]位指定的通道上的模拟看门狗功能 0：在所有的通道上使用模拟看门狗； 1：在单一通道上使用模拟看门狗。

位域	名称	描述
8	SCANMD	扫描模式 （Scan mode） 该位由软件设置和清除，用于开启或关闭扫描模式。在扫描模式中，转换由ADC_SEQx或 ADC_JSEQx寄存器选中的通道。0：关闭扫描模式； 1：使用扫描模式。 *注：如果分别设置了ENDCIE或JENDCIE位，只在最后一个通道转换完毕后才会产生ENDC或JENDC中断。*
7	JENDCIEN	允许产生注入通道转换结束中断 （Interrupt enable for injected channels） 该位由软件设置和清除，用于禁止或允许所有注入通道转换结束后产生中断。 0：禁止JENDC中断； 1：允许JENDC中断。当硬件设置JENDC位时产生中断。
6	AWDGIEN	允许产生模拟看门狗中断 （Analog watchdog interrupt enable） 该位由软件设置和清除，用于禁止或允许模拟看门狗产生中断。在扫描模式下，如果看门狗检测到超范围的数值时，只有在设置了该位时扫描才会中止。 0：禁止模拟看门狗中断； 1：允许模拟看门狗中断。
5	ENDCIEN	允许产生ENDC中断 （Interrupt enable for ENDC） 该位由软件设置和清除，用于禁止或允许转换结束后产生中断。 0：禁止ENDC中断； 1：允许ENDC中断。当硬件设置ENDC位时产生中断。
4:0	AWDGCH[4:0]	模拟看门狗通道选择位 （Analog watchdog channel select bits） 这些位由软件设置和清除，用于选择模拟看门狗保护的输入通道。 00000：ADC模拟输入通道0 00001：ADC模拟输入通道1 ..... 01111：ADC模拟输入通道15 10000：ADC模拟输入通道16 10001：ADC模拟输入通道17 10010：ADC模拟输入通道18 保留所有其他数值。

9.12.4 ADC 控制寄存器 2（ADC_CTRL2）

地址偏移：0x08
复位值：0x0000 0000

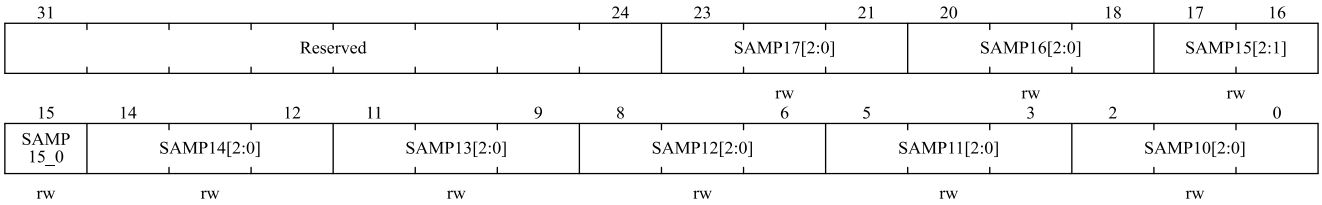


位域	名称	描述
31:24	Reserved	保留，硬件强制为0。
23	TEMPEN	温度传感器和V_{REFINT}使能 （Temperature sensor and V_{REFINT} enable） 该位由软件设置和清除，用于开启或禁止温度传感器和V_{REFINT}通道。在多于1个ADC的器件中，该位仅出现在ADC1中。 0：禁止温度传感器和V_{REFINT}； 1：启用温度传感器和V_{REFINT}。
22	SWSTRRCH	开始转换规则通道 （Start conversion of regular channels） 由软件设置该位以启动转换，转换开始后硬件马上清除此位。如果在EXTSEL[2:0]位中选择了SWSTART为触发事件，该位用于启动一组规则通道的转换， 0：复位状态； 1：开始转换规则通道。
21	SWSTRJCH	开始转换注入通道 （Start conversion of injected channels） 由软件设置该位以启动转换，软件可清除此位或在转换开始后硬件马上清除此位。如果在JEXTSEL[2:0]位中选择了JSWSTART为触发事件，该位用于启动一组注入通道的转换， 0：复位状态； 1：开始转换注入通道。
20	EXTRTRIG	规则通道的外部触发转换模式 （External trigger conversion mode for regular channels） 该位由软件设置和清除，用于开启或禁止可以启动规则通道组转换的外部触发事件。 0：不用外部事件启动转换； 1：使用外部事件启动转换。
19:17	EXTRSEL[2:0]	选择启动规则通道组转换的外部事件 （External event select for regular group） 这些位选择用于启动规则通道组转换的外部事件 ADC1和ADC2的触发配置如下 000：定时器1的CC1事件 100：定时器3的TRGO事件 001：定时器1的CC2事件 101：定时器4的CC4事件 010：定时器1的CC3事件 110：EXTI线11/TIM8_TRGO事件 011：定时器2的CC2事件 111：SWSTRRCH ADC3和ADC4的触发配置如下 000：定时器3的CC1事件 100：定时器8的TRGO事件 001：定时器2的CC3事件 101：定时器5的CC1事件 010：定时器1的CC3事件 110：EXTI线10/定时器5的CC3事件 011：定时器8的CC1事件 111：SWSTRRCH
16	Reserved	保留，硬件强制为0。
15	EXTJTRIG	注入通道的外部触发转换模式 （External trigger conversion mode for injected channels） 该位由软件设置和清除，用于开启或禁止可以启动注入通道组转换的外部触发事件。 0：不用外部事件启动转换； 1：使用外部事件启动转换。

位域	名称	描述
14:12	EXTJSEL[2:0]	选择启动注入通道组转换的外部事件 （External event select for injected group）这些位选择用于启动注入通道组转换的外部事件。 ADC1和ADC2的触发配置如下 000：定时器1的TRGO事件100：定时器3的CC4事件 001：定时器1的CC4事件 101：定时器4的TRGO事件 010：定时器2的TRGO事件110：EXTI线15/TIM8_CC4事件 011：定时器2的CC1事件 111：SWSTRJCH ADC3和ADC4的触发配置如下 000：定时器1的TRGO事件100：定时器8的CC4事件 001：定时器1的CC4事件 101：定时器5的TRGO事件 010：定时器4的CC3事件 110：EXTI线14/定时器5的CC4事件 011：定时器8的CC2事件 111：SWSTRJCH
11	ALIG	数据对齐 （Data alignment） 该位由软件设置和清除。参考图 9-9和图 9-10。 0：右对齐； 1：左对齐。
10:9	Reserved	保留，硬件强制为0。
8	ENDMA	直接存储器访问模式 （Direct memory access mode） 该位由软件设置和清除。详见DMA控制器章节。 0：不使用DMA模式； 1：使用DMA模式。 *注：只有ADC1和ADC3能产生DMA请求。*
7:3	Reserved	保留，硬件强制为0。
2	ENCAL	A/D校准 （A/D Calibration） 该位由软件设置以开始校准，并在校准结束时由硬件清除。 0：校准完成； 1：开始校准。
1	CTU	连续转换 （Continuous conversion） 该位由软件设置和清除。如果设置了此位，则转换将连续进行直到该位被清除。 0：单次转换模式； 1：连续转换模式。
0	ON	开/关A/D转换器 （A/D converter ON/OFF） 该位由软件设置和清除。当该位为'0'时，写入'1'将把ADC从断电模式下唤醒。 当该位为'1'时，写入'1'将启动转换。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}，参见图 9-5。 0：关闭ADC转换/校准，并进入断电模式； 1：开启ADC并启动转换。 *注：如果在这个寄存器中与ON一起还有其他位被改变，则转换不被触发。这是为了防止触发错误的转换。*

9.12.5 ADC 采样时间寄存器 1（ADC_SAMPT1）

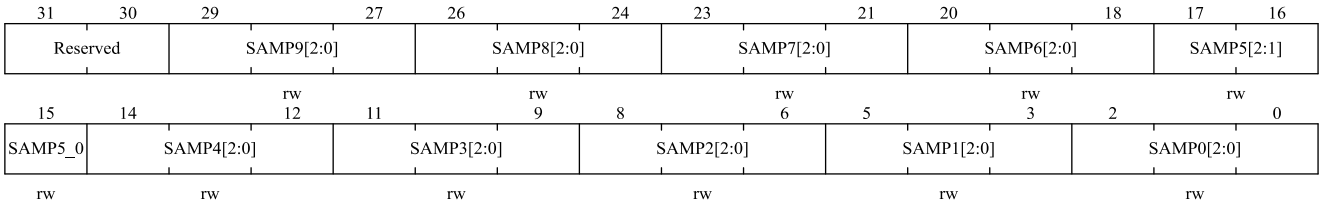
地址偏移：0x0C
复位值：0x0000 0000



位域	名称	描述
31:24	Reserved	保留，硬件强制为0。
23:0	SAMPx[2:0]	选择通道x的采样时间 （Channel x Sample time selection） 这些位用于独立地选择每个通道的采样时间。在采样周期中通道选择位必须保持不变。 ADC_SAMPT3.SAMPSEL位为0时，采样时间配置值如下 000：1.5周期 100：41.5周期 001：7.5周期 101：55.5周期 010：13.5周期 110：71.5周期 011：28.5周期 111：239.5周期 ADC_SAMPT3.SAMPSEL位为1时，采样时间配置值如下 000：1.5周期 100：19.5周期 100：2.5周期 101：61.5周期 010：4.5周期 110：181.5周期 011：7.5周期 111：601.5周期 *注：ADCx的模拟内部输入通道,参见“内部通道”部分。*

9.12.6 ADC 采样时间寄存器 2（ADC_SAMPT2）

地址偏移：0x10
复位值：0x0000 0000

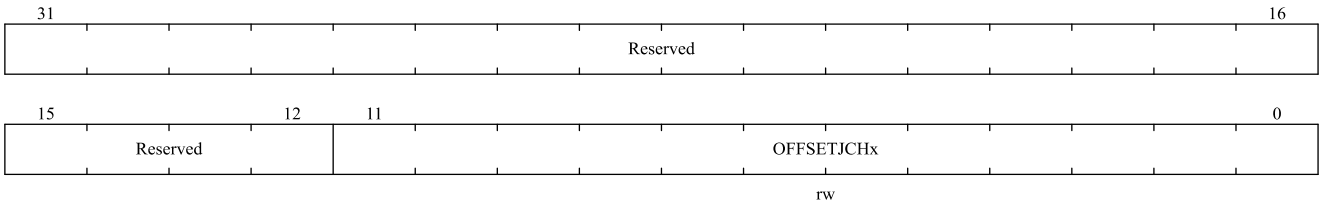


位域	名称	描述
31:30	Reserved	保留，硬件强制为0。

位域	名称	描述
29:0	SAMPx[2:0]	选择通道x的采样时间 （Channel x Sample time selection） 这些位用于独立地选择每个通道的采样时间。在采样周期中通道选择位必须保持不变。 ADC_SAMPT3.SAMPSEL位为0时，采样时间配置值如下 000：1.5周期 100：41.5周期 001：7.5周期 101：55.5周期 010：13.5周期 110：71.5周期 011：28.5周期 111：239.5周期 ADC_SAMPT3.SAMPSEL位为1时，采样时间配置值如下 000：1.5周期 100：19.5周期 100：2.5周期 101：61.5周期 010：4.5周期 110：181.5周期 011：7.5周期 111：601.5周期

9.12.7 ADC 注入通道数据偏移寄存器 x （ADC_JOFFSETx）（x=1..4）

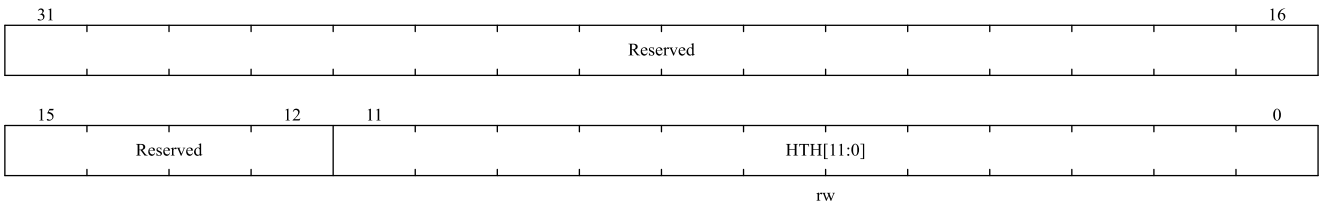
地址偏移：0x14-0x20
复位值：0x0000 0000



位域	名称	描述
31:12	Reserved	保留，硬件强制为0。
11:0	OFFSETJCHx[11:0]	注入通道x的数据偏移 （Data offset for injected channel x） 当转换注入通道时，这些位定义了用于从原始转换数据中减去的数值。转换的结果可以在ADC_JDATx寄存器中读出。

9.12.8 ADC 看门狗高阀值寄存器（ADC_WDGHIGH）

地址偏移：0x24
复位值：0x0000 0FFF

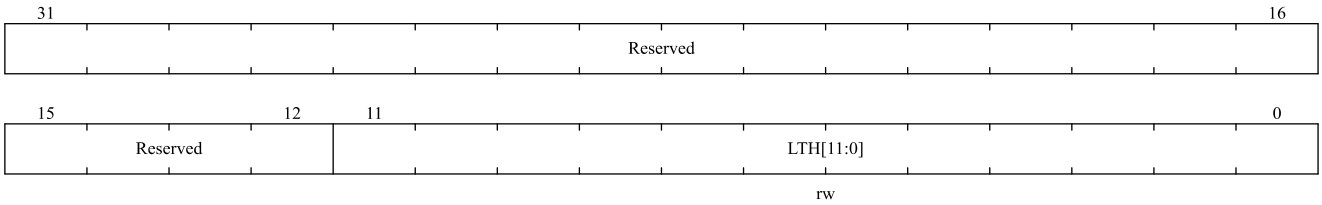


位域	名称	描述
31:12	Reserved	保留，硬件强制为0。

位域	名称	描述
11:0	HTH[11:0]	模拟看门狗高阀值 （Analog watchdog high threshold） 这些位定义了模拟看门狗的阀值高限。

9.12.9 ADC 看门狗低阀值寄存器（ADC_WDGLOW）

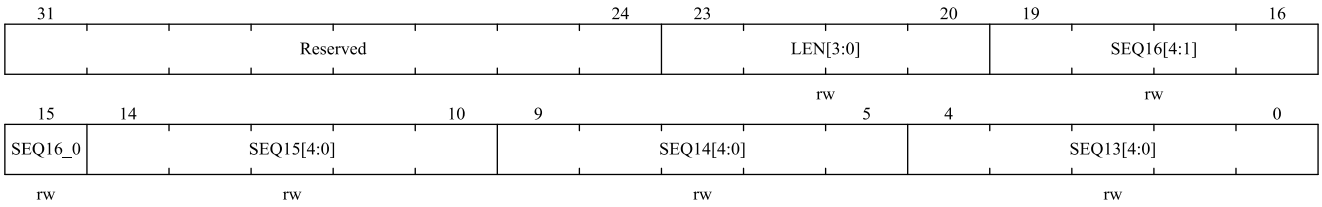
地址偏移：0x28
复位值：0x0000 0000



位域	名称	描述
31:12	Reserved	保留，硬件强制为0。
11:0	LTH[11:0]	模拟看门狗低阀值 （Analog watchdog low threshold） 这些位定义了模拟看门狗的阀值低限。

9.12.10 ADC 规则序列寄存器 1（ADC_RSEQ1）

地址偏移：0x2C
复位值：0x0000 0000

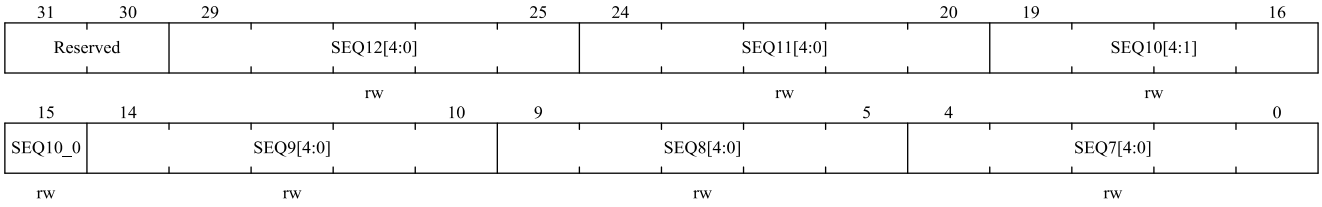


位域	名称	描述
31:24	Reserved	保留，硬件强制为0。
23:20	LEN[3:0]	规则通道序列长度 （Regular channel sequence length） 这些位由软件定义在规则通道转换序列中的通道数目。 0000：1个转换 0001：2个转换 ..... 1111：16个转换
19:15	SEQ16[4:0]	规则序列中的第16个转换 （16th conversion in regular sequence） 这些位由软件定义转换序列中的第16个转换通道的编号（0~18）。
14:10	SEQ15[4:0]	规则序列中的第15个转换 （15th conversion in regular sequence）
9:5	SEQ14[4:0]	规则序列中的第14个转换 （14th conversion in regular sequence）

位域	名称	描述
4:0	SEQ13[4:0]	规则序列中的第13个转换 （13th conversion in regular sequence）

9.12.11 ADC 规则序列寄存器 2（ADC_RSEQ2）

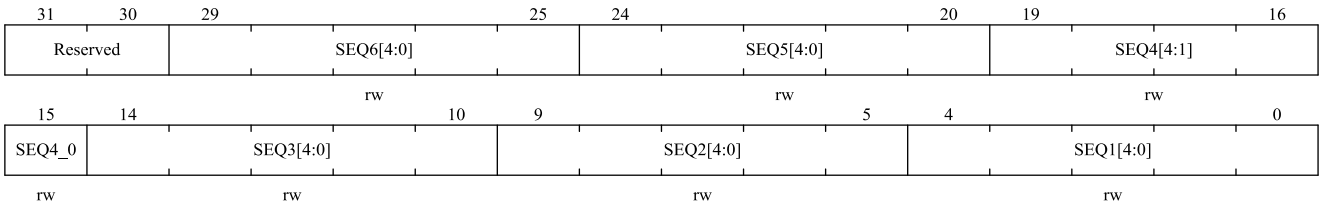
地址偏移：0x30
复位值：0x0000 0000



位域	名称	描述
31:30	Reserved	保留，硬件强制为0。
29:25	SEQ12[4:0]	规则序列中的第12个转换 （12th conversion in regular sequence） 这些位由软件定义转换序列中的第12个转换通道的编号（0~18）。
24:20	SEQ11[4:0]	规则序列中的第11个转换 （11th conversion in regular sequence）
19:15	SEQ10[4:0]	规则序列中的第10个转换 （10th conversion in regular sequence）
14:10	SEQ9[4:0]	规则序列中的第9个转换 （9th conversion in regular sequence）
9:5	SEQ8[4:0]	规则序列中的第8个转换 （82th conversion in regular sequence）
4:0	SEQ7[4:0]	规则序列中的第7个转换 （7th conversion in regular sequence）

9.12.12 ADC 规则序列寄存器 3（ADC_RSEQ3）

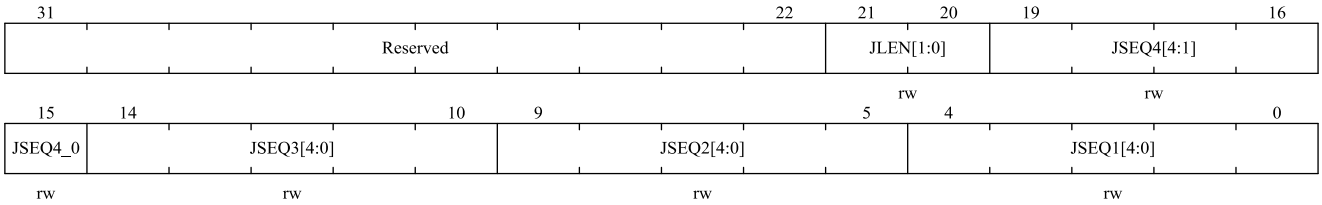
地址偏移：0x34
复位值：0x0000 0000



位域	名称	描述
31:30	Reserved	保留，硬件强制为0。
29:25	SEQ6[4:0]	规则序列中的第6个转换 （6th conversion in regular sequence） 这些位由软件定义转换序列中的第6个转换通道的编号（0~18）。
24:20	SEQ5[4:0]	规则序列中的第5个转换 （5th conversion in regular sequence）
19:15	SEQ4[4:0]	规则序列中的第4个转换 （4th conversion in regular sequence）
14:10	SEQ3[4:0]	规则序列中的第3个转换 （3rd conversion in regular sequence）
9:5	SEQ2[4:0]	规则序列中的第2个转换 （2nd conversion in regular sequence）
4:0	SEQ1[4:0]	规则序列中的第1个转换 （1st conversion in regular sequence）

9.12.13 ADC 注入序列寄存器（ADC_JSEQ）

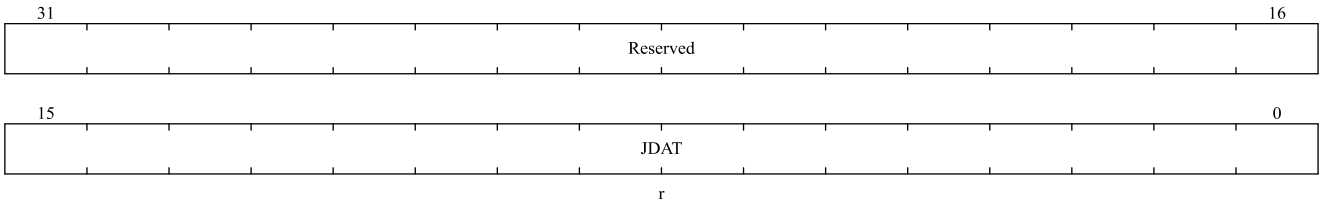
地址偏移：0x38
复位值：0x0000 0000



位域	名称	描述
31:22	Reserved	保留，硬件强制为0。
21:20	JLEN[1:0]	注入通道序列长度 （Injected sequence length） 这些位由软件定义在注入通道转换序列中的通道数目。 00：1个转换 01：2个转换 10：3个转换 11：4个转换
19:15	JSEQ4[4:0]	注入序列中的第4个转换 （4th conversion in injected sequence） 这些位由软件定义转换序列中的第4个转换通道的编号（0~1）。 *注：不同于规则转换序列，如果JL[1:0]的长度小于4，则转换的序列顺序是从（4-JL）开始。例 如：ADC_JSQR[21:0] = 10 00011 00011 00111 00010，意味着扫描转换将按下列通道顺序转换：7、3、3，而不是2、7、3。*
14:10	JSEQ3[4:0]	注入序列中的第3个转换 （3rd conversion in injected sequence）
9:5	JSEQ2[4:0]	注入序列中的第2个转换 （2nd conversion in injected sequence）
4:0	JSEQ1[4:0]	注入序列中的第1个转换 （1st conversion in injected sequence）

9.12.14 ADC 注入数据寄存器 x （ADC_JDATx） （x= 1..4）

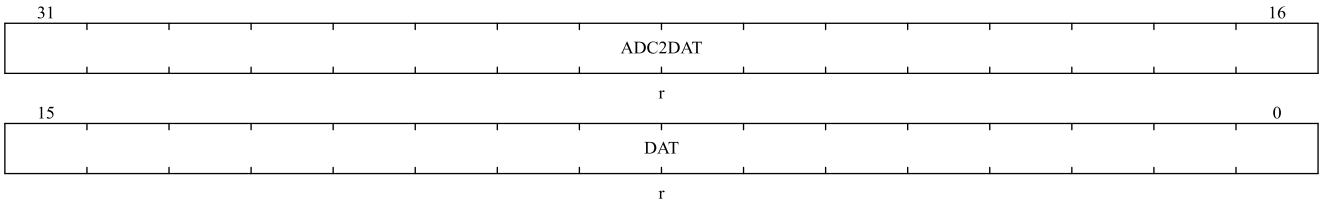
地址偏移：0x3C – 0x48
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为0。
15:0	JDAT[15:0]	注入转换的数据 （Injected data） 这些位为只读，包含了注入通道的转换结果。数据是左对齐或右对齐，如图 9-9图 9-9和图 9-10所示。

9.12.15 ADC 规则数据寄存器（ADC_DAT）

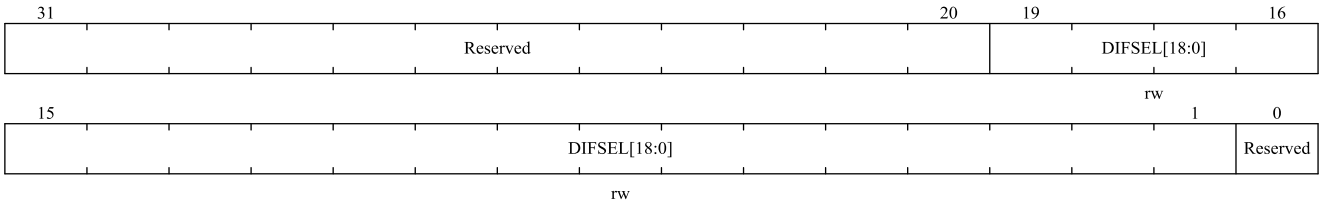
地址偏移：0x4C
复位值：0x0000 0000



位域	名称	描述
31:16	ADC2DAT[15:0]	ADC2/ADC4转换的数据 （ADC2 data,ADC4 data） 在ADC1和ADC3中：双模式下，这些位包含了ADC2或ADC4转换的规则通道数据。见9.9: 双ADC模式在ADC2和ADC4中：不使用这些位。
15:0	DAT[15:0]	规则转换的数据 （Regular data） 这些位为只读，包含了规则通道的转换结果。数据是左对齐或右对齐，如图 9-9 和图 9-10所示。

9.12.16 ADC 差分模式选择寄存器 （ADC_DIFSEL）

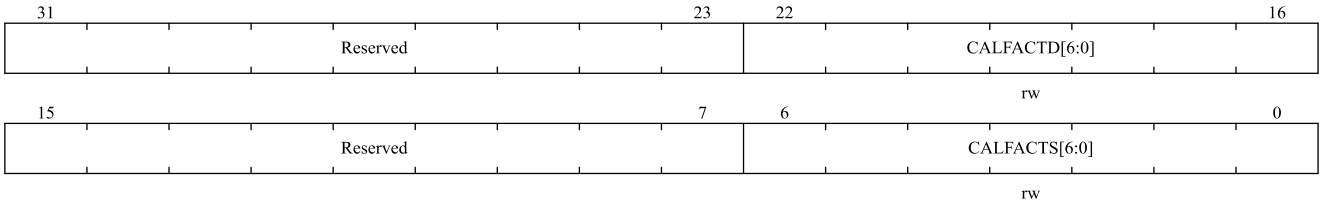
地址偏移：0x50
复位值：0x0000 0000



位域	名称	描述
31:20	Reserved	保留，硬件强制为0。
19:1	DIFSEL[18:0]	差分模式通道18—0选择（Differential mode for channels 18 to 0） DIFSEL[i] = 0:ADC 通道输入i配置成单端模式; DIFSEL[i] = 1:ADC 通道输入i配置成差分模式
0	Reserved	保留，硬件强制为0。

9.12.17 ADC 校正因子 （ADC_CALFACT）

地址偏移：0x54
复位值：0x0000 0000

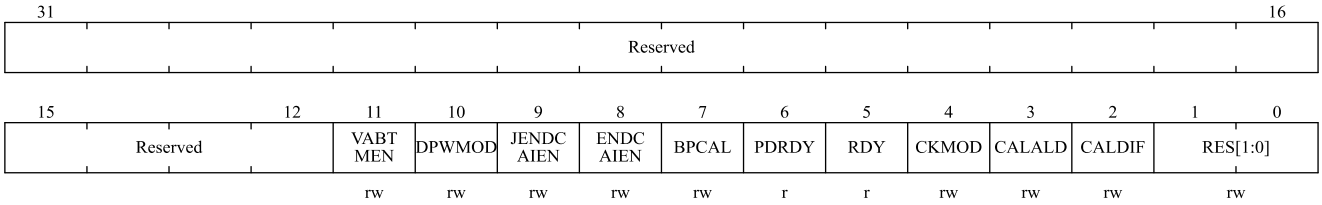


位域	名称	描述
31:23	Reserved	保留，硬件强制为0。
22:16	CALFACTD[6:0]	差分模式校正因子（Calibration Factors in differential mode） 此位可以被硬件或软件写 差分输入校准完成后，硬件会根据校准系数对其进行更新。 软件可以用新的校准因子写入这些位。如果新的校准系数与存储在模拟ADC中的当前系数不同，则在启动新的差分校准后将应用该系数。 *注意：软件仅当ADEN=1,ADSTR=0,JADSTR=0允许写入（ADC 没有处理转换中或开始转换）*
15:7	Reserved	保留，硬件强制为0。
6:0	CALFACTS[6:0]	单端模式校正因子（Calibration Factors in Single-Ended mode） 此位可以被硬件或软件写 单端输入校准完成后，硬件会根据校准系数对其进行更新。 软件可以用新的校准因子写入这些位。如果新的校准系数与存储在模拟ADC中的当前系数不同，则在启动新的单端校准后将应用该系数。 *注意：软件仅当ADEN=1,ADSTR=0,JADSTR=0允许写入（ADC 没有处理转换中或开始转换）*

9.12.18 ADC 控制寄存器 3 （ADC_CTRL3）

地址偏移：0x58

复位值：0x0000 0043

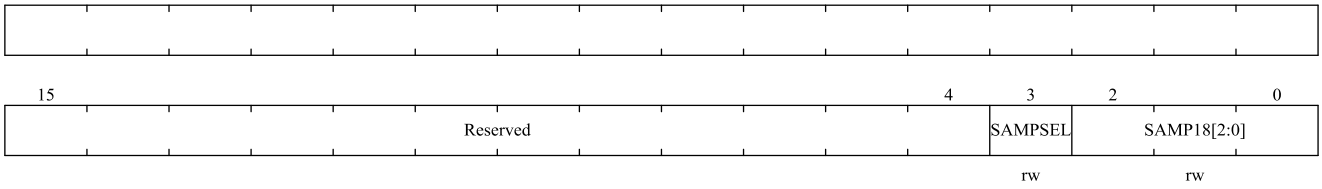


位域	名称	描述
31:12	Reserved	保留，硬件强制为0。
11	VBATMEN	Vbat监测使能 （Vbat monitor enable） 0：禁止 1：使能
10	DPWMOD	深度低功耗模式（Deep Power Mode） 0：当ADC不使能时，ADC 进入低功耗 1：当ADC不使能时，ADC 进入深度低功耗

位域	名称	描述
9	JENDCAIEN	任何注入通道中断使能 （Interrupt enable for any injected channels） 该位由软件设置和清除，以启用/禁用注入通道转换结束中断 0: JENDCA 中断禁止 1：JENDCA 中断使能
8	ENDCAIEN	任何通道中断使能 （Interrupt enable for any channels） 该位由软件设置和清除，以启用/禁用任意通道转换结束中断 0: ENDCA 中断禁止 1：ENDCA 中断使能
7	BPCAL	旁路校准 （Bypass calibration） 0：禁止 1：使能，旁路校准信号
6	PDRDY	ADC低功耗准备好（Power down ready） 0：没有准备好 1：准备好
5	RDY	ADC准备好（Ready） 0：没有准备好 1：准备好
4	CKMOD	时钟模式（Clock Mode） 0：同步时钟选择AHB 1：异步时钟选择PLL
3	CALALD	校准自动载入 （calibration auto load） 0：禁止自动载入 1：使能自动载入
2	CALDIF	差分模式校准（Differential mode for calibration） 该位由软件设置和清除，以配置校准的单端或差分输入模式 0：写入ENCAL位将在单端输入模式下启动校准 1：写入ENCAL位将在差分输入模式下启动校准
1:0	RES[1:0]	数据分辨率（Data resolution） 该位由软件设置和清除，,以选择转换的分辨率 00：6位 01：8位 10：10位 11：12位

9.12.19 ADC 采样时间寄存器 3 （ADC_SAMPT3）

地址偏移：0x5C
复位值：0x0000 0000



位域	名称	描述
31:4	Reserved	保留，硬件强制为0。
3	SAMPSEL	采样时间选择（Sample time selection） SAMPSEL位为0时，采样时间配置SAMPx[2:0]的取值如下 000：1.5周期　　100：41.5周期 001：7.5周期　　101：55.5周期 010：13.5周期　　110：71.5周期 011：28.5周期　　111：239.5周期 SAMPSEL位为1时，采样时间配置SAMPx[2:0]的取值如下 000：1.5周期　　100：19.5周期 100：2.5周期　　101：61.5周期 010：4.5周期　　110：181.5周期 011：7.5周期　　111：601.5周期
2:0	SAMP18[2:0]	通道采样时间（Channel Sample time） 通道采样时间定义和ADC_SAMPT2一致

10 数字/模拟转换（DAC）

10.1 DAC 介绍

DAC 是数字/模拟转换器，主要是数字输入，电压输出。DAC 数据有 8 位或 12 位两种模式，可以与 DMA 控制器配合使用。当工作在 12 位模式时，DAC 数据可以配置成左对齐或者右对齐。当工作在 8 位模式时，DAC 数据可以配置成右对齐。DAC 输出通道有 2 个，都有独立的转换器。在双 DAC 模式下，两个通道可以根据需要配置成独立转换或者同时进行转换同步更新输出。另外 DAC 为了达到更好的精确度，可以使用外部的引脚输入参考电压 VREF+。

10.2 DAC 主要特性

- 2 个独立的 DAC 转换器，各自对应 1 个输出通道
- 8 位或者 12 位单调输出
- 数据在 12 位模式下分右对齐和左对齐两种模式
- 同步更新机制
- 噪声波形、三角波形生成
- 双 DAC 通道同时或者分别转换
- 每个通道都有 DMA 功能
- 外部触发转换
- 输入参考电压 VREF+

单个 DAC 通道的框图如下图，表 10-1 给出了引脚的说明。

图 10-1 单个 DAC 通道框图

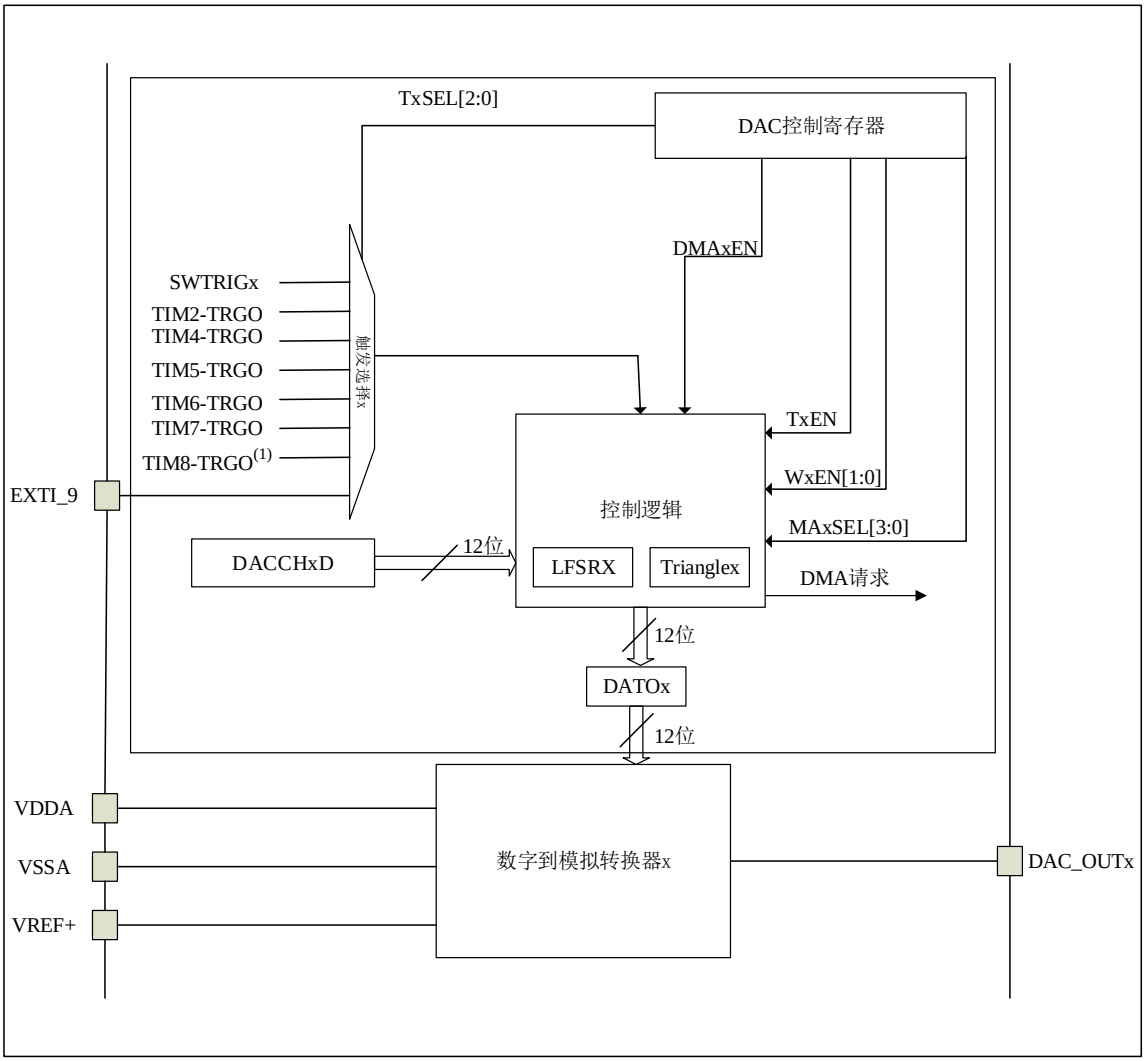


表 10-1 DAC 引脚

名称	型号类型	注释
V_{REF+}	输入，正模拟参考电压	DAC使用的高端/正极参考电压， 2.4V ≤ V_{REF+} ≤ V_{DDA} （3.3V）
V_{DDA}	输入，模拟电源	模拟电源
V_{SSA}	输入，模拟电源地	模拟电源的地线
DAC_OUTx	模拟输出信号	DAC通道x的模拟输出

注意：使能DACx通道，对应的GPIO管脚（PA4或者PA5）就会自动与DAC的模拟输出相连（DAC_OUTx）。

此时管脚PA4或者PA5在之前应当设置成模拟输入（AIN），这样可以避免寄生的干扰和额外的功耗。

10.3 DAC 功能描述与操作说明

10.3.1 开启 DAC 通道

对 DAC_CTRL 寄存器的 CHxEN 位置'1'将可以开启对应 DAC 通道 x 的供电。DAC 通道 x 经过一段启动时间 t_{WAKEUP} 即被打开。

注意：CHxEN 位开启的只是 DAC 通道 x 的模拟部分，即便该位被配置为'0'，DAC 通道 x 的数字部分仍然工作。

10.3.2 开启 DAC 输出缓存

DAC 的 2 个输出缓存，输出低阻抗，可直接驱动外部负载无需外部运放。每个 DAC 通道输出缓存可以通过配置 DAC_CTRL 寄存器的 BxEN 位来开启或者关闭。

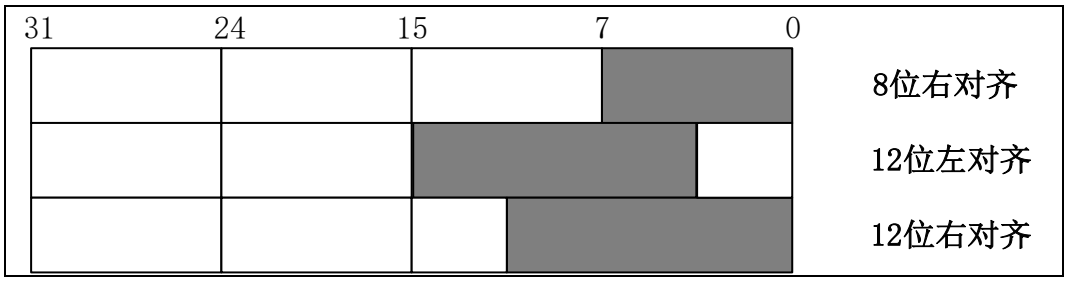
10.3.3 DAC 数据格式

依据数据对齐配置方式，数据配置对应寄存器如下：

- 单路 DAC 通道 x，有 3 种情况：
  - 12 位数据左对齐：配置数据写入寄存器 DL12CHx[15:4]位（实际是存入寄存器 DACCHxD [11:0]位）
  - 12 位数据右对齐：配置数据写入寄存器 DR12CHx[11:0]位（实际是存入寄存器 DACCHxD [11:0]位）
  - 8 位数据右对齐：配置数据写入寄存器 DR8CHx [7:0]位（实际是存入寄存器 DACCHxD[11:4]位）

配置对应的 DAC 数据保持寄存器，写入的数据经过相应的移位被保存到 DACCHxD 寄存器（DACCHxD 是内部的数据保存寄存器 x）。另外，保存到 DACCHxD 寄存器的数据可以通过软件触发或外部事件触发被传送到 DATOx 寄存器，同时也可以自动传送到 DATOx 寄存器。

图 10-2 单 DAC 通道模式的数据寄存器



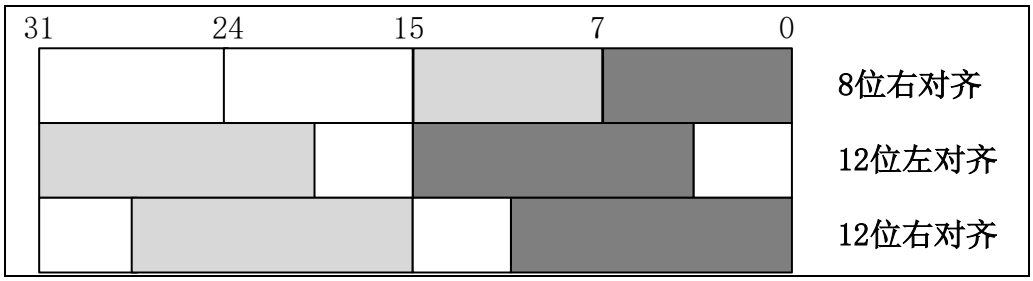
- 双 DAC 通道 x，有 3 种情况：
  - 8 位数据右对齐：配置 DAC 通道 1 数据写入寄存器 DAC_DR8DCH [7:0]位（实际是存入寄存器 DACCH1D[11:4]位），配置 DAC 通道 2 数据写入寄存器 DAC_DR8DCH[15:8]位（实际是存入寄存器 DACCH2D [11:4]位）
  - 12 位数据左对齐：配置 DAC 通道 1 数据写入寄存器 DAC_DL12DCH[15:4]位（实际是存入寄存器

DACCH1D[11:0]位），配置 DAC 通道 2 数据写入寄存器 DAC_DL12DCH[31:20]位（实际是存入寄存器 DACCH2D[11:0]位）

- 12 位数据右对齐：配置 DAC 通道 1 数据写入寄存器 DAC_DR12DCH[11:0]位（实际是存入寄存器 DACCH1D[11:0]位），配置 DAC 通道 2 数据写入寄存器 DAC_DR12DCH[27:16]位（实际是存入寄存器 DACCH2D[11:0]位）

配置对应的 DAC 双通道数据保持寄存器，写入的数据经过相应的移位被保存到 DACCHxD 寄存器（DACCHxD 是内部的数据保存寄存器 x）。另外，保存到 DACCHxD 寄存器的数据可以通过软件触发或外部事件触发被传送到 DATOx 寄存器，同时也可以自动传送到 DATOx 寄存器。

图 10-3 双 DAC 通道模式的数据寄存器



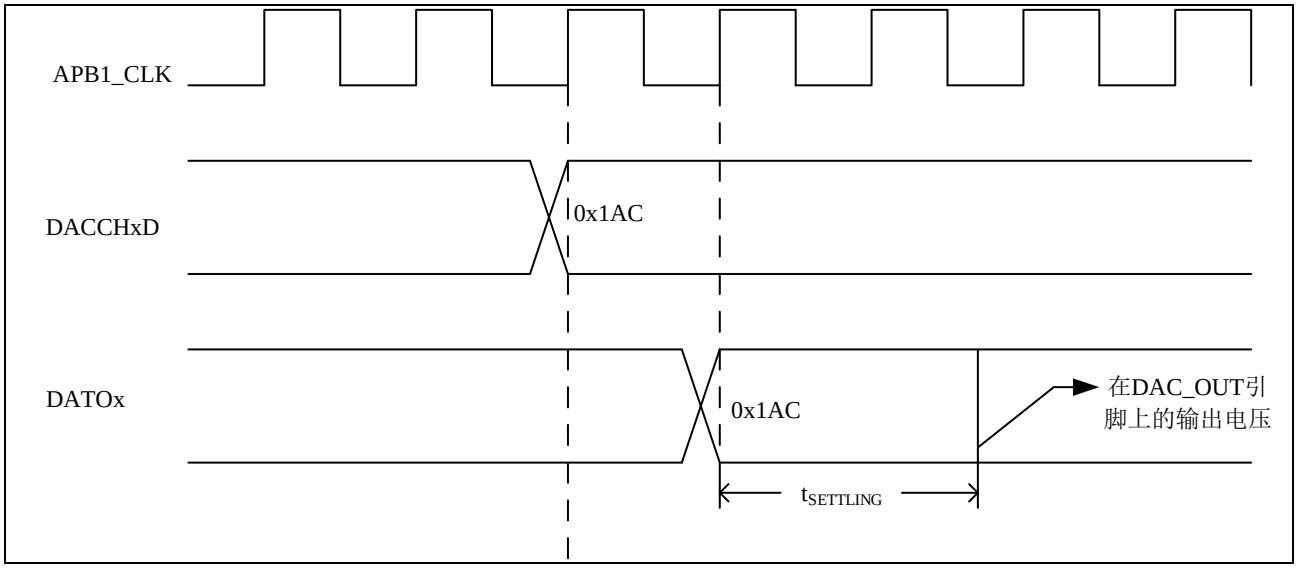
10.3.4 DAC 转换

输出到 DAC 通道 x 的数据都必须写入 DACCHxD 寄存器（数据实际写入 DAC_DR12CHx、DAC_DL12CHx、DAC_DR8CHx、DAC_DR12DCH、DAC_DL12DCH、或者 DAC_DR8DCH 寄存器）。禁止直接对寄存器 DAC_DATOx 写入数据，

寄存器 DAC_CTRL 的 TxEN 位置‘0’，表明没有选中硬件触发，那么存入 DAC 数据保持寄存器的数据会在一个 APB1 时钟周期后自动传至寄存器 DAC_DATOx。寄存器 DAC_CTRL 的 TxEN 位置‘1’，则选中了硬件触发，数据传输在触发发生以后 3 个 APB1 时钟周期后完成。

DAC 数据保持寄存器一旦将数据传递到 DAC_DATOx 寄存器，输出需要经过 $t_{SETTLING}$ 之后才有效，该时间会因电源电压和模拟输出负载不同会有所不同。

图 10-4 TEN=0 触发禁能时转换的时间框图



10.3.5 DAC 输出电压

数字输入通过 DAC 模块可以转换为线性的模拟电压输出，其范围为 0 到 VREF+。DAC 通道引脚上的输出电压满足下面的关系：

DAC 输出 = V_{REF} x (DATOx / 4095)。

10.3.6 DAC 触发

将 TxEN 位置 1，某外部事件（定时器计数器、外部中断线）将触发 DAC 转换。另外可以配置控制位 TxSEL[2:0]来选择 8 个触发事件之一触发 DAC 转换。

表 10-2 DAC 外部触发

触发源	类型	TxSEL[2:0]
定时器 6 TRGO 事件	来自片上定时器的内部信号	0
定时器 8 TRGO 事件		1
定时器 7 TRGO 事件		10
定时器 5 TRGO 事件		11
定时器 2 TRGO 事件		100
定时器 4 TRGO 事件		101
EXTI 线路 9	外部引脚	110
SWTRIG（软件触发）	软件控制位	111

定时器 TRGO 输出，或者外部中断线 9 的上升沿一旦触发 DAC 模块，存放在对齐数据保持寄存器中的数据会被传送到寄存器 DAC_DATOx 中。在 3 个 APB1 时钟周期后，寄存器 DAC_DATOx 更新为新

值。

软件触发由 DAC_SOTTR 寄存器配置，一旦 TRxEN 位置‘1’，转换即开始。在数据从对齐数据保持寄存器传送到 DAC_DATOx 寄存器后，TRxEN 位由硬件自动清‘0’。

注意:
1. 当 CHxEN 为‘1’时禁止改变 TxSEL[2:0]位。
2.软件触发模式中，数据从对齐数据保持寄存器传送到寄存器 DAC_DATOx 只需要一个 APB1 时钟周期。

10.3.7 DMA 请求

DAC 通道可以通过 DMAxEN 位来开启 DMA 功能。2 个 DMA 通道可分别用于 2 个 DAC 通道的 DMA 请求。

当 DMAxEN 位置‘1’，有外部触发发生，而不是软件触发，则生成一个 DMA 请求，之后对齐数据保持寄存器的数据被传送到 DAC_DATOx 寄存器。

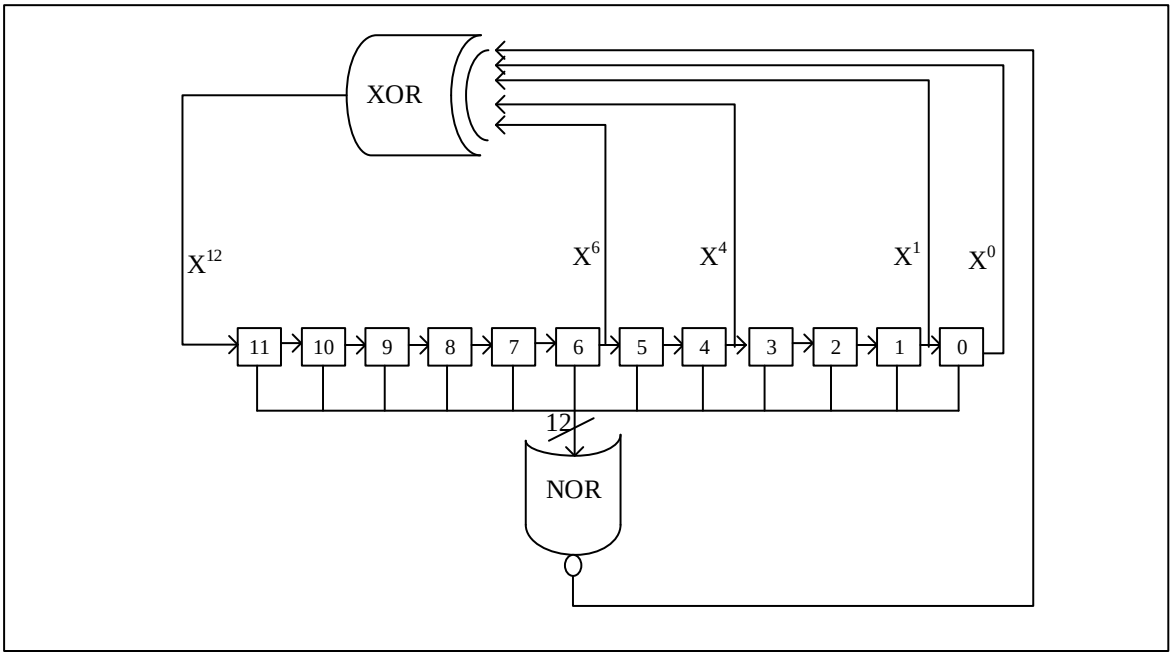
在双 DAC 模式下，程序可以在只使用一个 DMA 请求，一个 DMA 通道的情况下，处理工作在双 DAC 模式的 2 个 DAC 通道。实际只需要一个 DMA 传输，只需选择其中一个 DMAxEN 位置‘1’。因为 2 个通道的 DMAxEN 位都为‘1’，则会产生 2 个 DMA 请求。

注意：DAC 的 DMA 请求没有累计功能，当第 2 个外部触发发生在响应第 1 个外部触发之前，则不能处理第 2 个 DMA 请求，也没有报告错误机制。

10.3.8 噪声产生

DAC 具有生成噪声功能。可以通过线性反馈移位寄存器（Linear Feedback Shift Register LFSR）产生幅度变化的伪噪声。配置 WxEN [1:0]位为‘01’开启 DAC 噪声生成功能。寄存器 LFSR 的初始值为 0xAAA。按照特定算法，在每次触发事件后 3 个 APB1 时钟周期之后更新该寄存器的值。

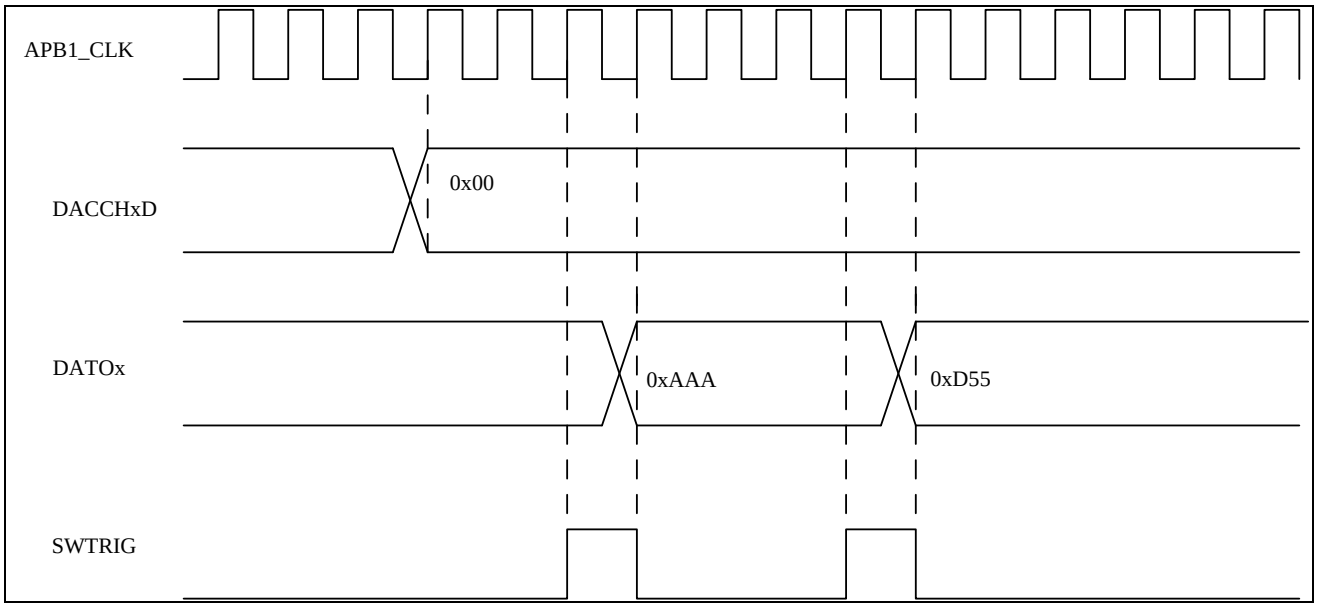
图 10-5 DAC LFS 寄存器算法



LFS 寄存器的数据可以通过配置 DAC_CTRL 寄存器的 MAxSEL[3:0]位来屏蔽部分或者全部，由此得到的 LSF 值与 DAC 对齐数据保持寄存器的数值相加，去除溢出位之后将被写入到 DAC_DATOx 寄存器。

若寄存器 LFS 寄存器值为 0x000，则会开启防锁定机制，注入‘1’。若 WxEN[1:0]位置‘0’可以复位 LFS 寄存器波形的生成算法。

图 10-6 带 LFS 寄存器波形生成的 DAC 转换（使能软件触发）



注意：为了产生噪声，必须开启 DAC 触发，即配置 DAC_CTRL 寄存器的 TxEN 位为‘1’。

10.3.9 三角波产生

DAC 模块可以通过配置 WxEN[1:0]位为‘10’选择 DAC 的三角波生成功能。原理是在 DC 或者变化缓慢的信号基准上叠加一个小幅度的三角波。三角波的幅度由 DAC_CTRL 寄存器的 MAxSEL[3:0]位决定。内部的三角波计数器的值和 DAC 对齐数据保持寄存器的数值将共同决定了 DAC 输出值，两个数值相加并丢弃溢出位之后传递到 DAC_DATOx 寄存器。另外内部的三角波计数器每次触发事件之后 3 个 APB1 时钟周期后进行更新。若 DAC_DATOx 寄存器的数值小于 MAxSEL[3:0] 位定义的最大幅度时，三角波计数器逐步累加。达到设置的最大幅度时，则计数器开始递减，达到 0 后再开始累加，周而复始。将 WxEN[1:0]位置‘0’可以复位三角波的生成。

图 10-7 DAC 三角波生成

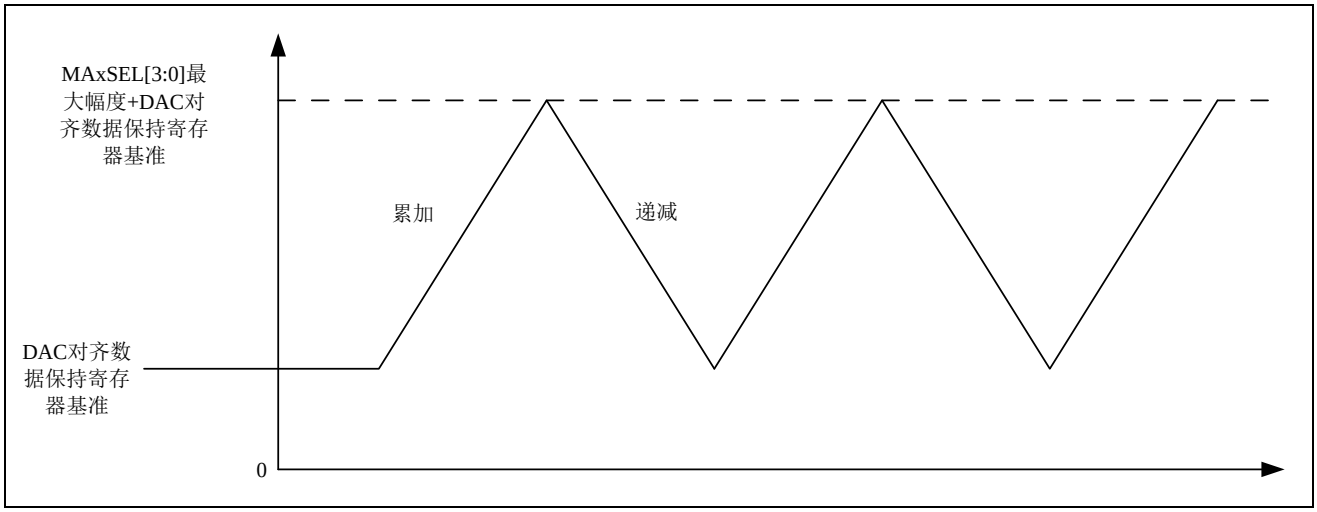
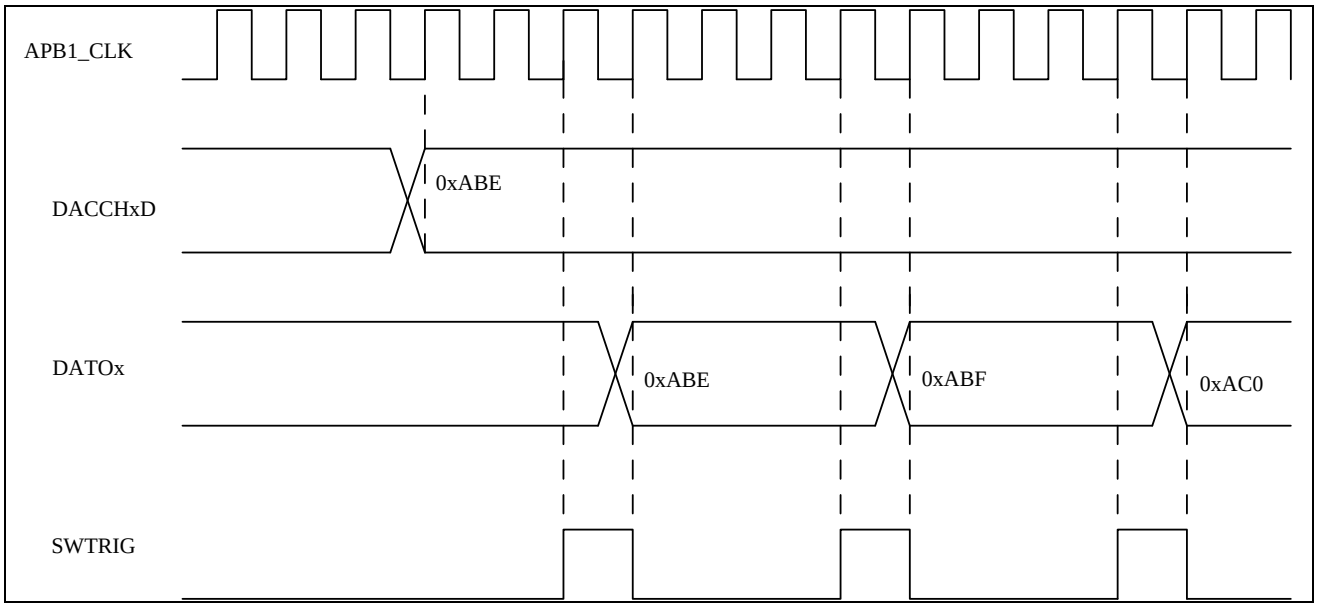


图 10-8 带三角生成的 DAC 转换（使能软件触发）



注意： 1. 为了产生三角波，必须使能 DAC 触发，即设 DAC_CTRL 寄存器的 TxEN 位为‘1’。
2. MAxSEL [3:0]位必须在使能 DAC 之前设置，否则其值不能修改。

10.4 DAC 双通道转换操作

DAC 两个通道即可以单独工作，也可以同时工作。该模式下，有 DR12DCH、DL12DCH 和 DR8DCH 总共 3 个寄存器可以使用，可以高效的利用总线带宽，每个寄存器都可以同时对 2 路 DAC 进行操作。

双 DAC 通道同时开启转换，总共有 11 种模式，在只使用一路 DAC 转换的情况下，另外一路 DAC 仍可以独立进行操作。具体请参考下面的章节描述。

10.4.1 不使用波形发生器的独立触发

配置 DAC 工作在此转换模式:

- 分别开启触发使能位，配置 T1EN 和 T2EN 为‘1’；
- 分别选择不同触发源，配置 T1SEL[2:0]和 T2SEL[2:0]位为不同值；
- 将双 DAC 通道转换数据装入所需的对齐数据保持寄存器（DR12DCH、DL12DCH 或 DR8DCH）。

DAC 通道 1 触发事件发生时，对齐数据保持寄存器的值将在延迟 3 个 APB1 时钟周期后传入寄存器

DAC_DATO1。

DAC 通道 2 触发事件发生时，对齐数据保持寄存器的值将在延迟 3 个 APB1 时钟周期后传入寄存器

DAC_DATO2。

10.4.2 使用相同的 LFS 寄存器的独立触发

配置 DAC 工作在此转换模式:

- 分别开启触发使能位，配置 T1EN 和 T2EN 为‘1’；
- 分别选择不同触发源，配置 T1SEL[2:0]和 T2SEL[2:0]位为不同值；
- 配置 2 个 DAC 通道的 W1EN[1:0]位为‘01’，并配置 MAxSEL3:0]为相同的 LFS 屏蔽值；
- 将双 DAC 通道转换数据装入到对应的对齐数据保持寄存器中（DR12DCH、DL12DCH 或 DR8DCH）。

DAC 通道 1 触发事件发生时，具有相同屏蔽的 LFS 寄存器 1 计数器值与对应数据保持寄存器数值相加，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO1，然后更新 LFS 寄存器 1 计数器。

DAC 通道 2 触发事件发生时，具有相同屏蔽的 LFS 寄存器 2 计数器值与对应数据保持寄存器数值相加，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO2，然后更新 LFS 寄存器 2 计数器。

10.4.3 使用不同的 LFS 寄存器的独立触发

配置 DAC 工作在此转换模式:

- 分别开启触发使能位，配置 T1EN 和 T2EN 为‘1’；
- 分别选择不同触发源，配置 T1SEL[2:0]和 T2SEL[2:0]位为不同值；

■ 配置 2 个 DAC 通道的 WxEN [1:0]位为‘1’，并配置 MAxSEL[3:0]为不同的 LFS 屏蔽值；
■ 将双 DAC 通道转换数据装入到对应的对齐数据保持寄存器中（DR12DCH、DL12DCH 或 DR8DCH）。

DAC 通道 1 触发事件发生时，依据 MA1SEL[3:0]所设屏蔽的 LFS 寄存器 1 计数器值与对应数据保持寄存器数值相加，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO1，然后更新 LFS 寄存器 1 计数器。

DAC 通道 2 触发事件发生时，依据 MA2SEL [3:0]所设屏蔽的 LFS 寄存器 2 计数器值与对应数据保持寄存器数值相加，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO2，然后更新 LFS 寄存器 2 计数器。

10.4.4 产生相同三角波的独立触发

配置 DAC 工作在此转换模式:

■ 分别开启触发使能位，配置 T1EN 和 T2EN 为‘1’；
■ 分别选择不同触发源，配置 T1SEL[2:0]和 T2SEL[2:0]位为不同值；
■ 配置 2 个 DAC 通道的 WxEN[1:0]位为‘1x’，并配置 MAxSEL[3:0]为相同的三角波幅值；
■ 将双 DAC 通道转换数据装入到对应的对齐数据保持寄存器中（DR12DCH、DL12DCH 或 DR8DCH）。

DAC 通道 1 触发事件发生时，三角波幅值加上对齐数据保持寄存器的值，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO1，然后更新 DAC 通道 1 三角波计数器。

DAC 通道 2 触发事件发生时，三角波幅值加上对应数据保持寄存器的值，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO2，然后更新 DAC 通道 2 三角波计数器。

10.4.5 产生不同三角波的独立触发

配置 DAC 工作在此转换模式:

■ 分别开启触发使能位，配置 T1EN 和 T2EN 为‘1’；
■ 分别选择不同触发源，配置 T1SEL[2:0]和 T2SEL[2:0]位为不同值；
■ 配置 2 个 DAC 通道的 WxEN[1:0]位为‘1x’，并设 MAxSEL[3:0]为不同的三角波幅值。

■ 将双 DAC 通道转换数据装入到对应的对齐数据保持寄存器中（DR12DCH、DL12DCH 或 DR8DCH)。DAC 通道 1 触发事件发生时，MA1SEL[3:0]所设的三角波幅值加上对应数据保持寄存器数值，

结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO1，然后更新 DAC 通道 1 三角波计数器。

DAC 通道 2 触发事件发生时，MA2SEL[3:0]所设的三角波幅值加上对应数据保持寄存器数值，

结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO2，然后更新 DAC 通道 2 三角波计数

器。

10.4.6 同时软件启动

配置 DAC 工作在此转换模式:

■ 分别开启触发使能位，配置 T1EN 和 T2EN 为‘1’；
■ 分别开启软件触发，配置 TR2EN 和 TR1EN 为‘1’；
■ 将双 DAC 通道转换数据装入到对应的对齐数据保持寄存器中（DR12DCH、DL12DCH 或 DR8DCH）。在此配置下，一个 APB1 时钟周期后，数据保持寄存器的数值即被分别传入 DAC_DATO1 和 DAC_DATO2 寄存器。

10.4.7 不使用波形发生器的同时触发

配置 DAC 工作在此转换模式:

■ 分别开启触发使能位，配置 T1EN 和 T2EN 为‘1’；
■ 分别设置 DAC 通道使用相同触发源，配置 T1SEL[2:0]和 T2SEL[2:0]位为相同值；
■ 将双 DAC 通道转换数据装入到对应的对齐数据保持寄存器中（DR12DCH、DL12DCH 或 DR8DCH）。

触发事件发生时,数据保持寄存器的数值将在延迟 3 个 APB1 时钟周期后被分别传入 DAC_DATO1 和 DAC_DATO2 寄存器。

10.4.8 使用相同 LFS 寄存器的同时触发

配置 DAC 工作在此转换模式:

■ 分别开启触发使能位，配置 T1EN 和 T2EN 为‘1’；
■ 分别设置 DAC 通道使用相同触发源，配置 T1SEL[2:0]和 T2SEL[2:0]位为相同值；
■ DAC 通道的 WxEN[1:0]位都配置为”01”，并设 MAxSEL[3:0]为相同的 LFS 屏蔽值；
■ 将双 DAC 通道转换数据装入到对应的对齐数据保持寄存器中（DR12DCH、DL12DCH 或 DR8DCH）。触发事件发生时，MA1SEL[3:0]所设屏蔽的 LFS1 计数器值与对应数据保持寄存器的数值相加，结果将在延迟 3 个 APB1 时钟周期后传入 DAC_DATO1 寄存器，然后更新 LFS1 计数器。

同样，MA2SEL[3:0]所设屏蔽的 LFS2 计数器值与对应数据保持寄存器的数值相加，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO2，然后更新 LFS2 计数器。

10.4.9 使用不同 LFS 寄存器的同时触发

配置 DAC 工作在此转换模式:

■ 分别开启触发使能位，配置 T1EN 和 T2EN 为‘1’；
■ 分别设置 DAC 通道使用相同触发源，配置 T1SEL[2:0]和 T2SEL[2:0]位为相同值；
■ DAC 通道的 WxEN[1:0]位都配置为”01”，并设 MAxSEL[3:0]为不同的 LFS 屏蔽值；
■ 将双 DAC 通道转换数据装入到对应的对齐数据保持寄存器中（DR12DCH、DL12DCH 或 DR8DCH）。

当发生触发事件时，具有相同屏蔽的 LFS1 计数器值与对应的数据保持寄存器数值相加，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO1，然后更新 LFS1 计数器。

同时，具有相同屏蔽的 LFS2 计数器值与对应的数据保持寄存器数值相加，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO2，然后更新 LFS2 计数器。

10.4.10 使用相同三角波发生器的同时触发

配置 DAC 工作在此转换模式:

■ 分别开启触发使能位，配置 T1EN 和 T2EN 为‘1’；
■ 分别设置 DAC 通道使用相同触发源，配置 T1SEL[2:0]和 T2SEL[2:0]位为相同值；
■ DAC 通道的 WxEN [1:0]位都配置为‘1 x’，并设 MAxSEL[3:0]为相同的三角波幅值。
■ 将双 DAC 通道转换数据装入到对应的对齐数据保持寄存器中（DR12DCH、DL12DCH 或 DR8DCH）。

触发事件发生时，相同的三角波幅值与对应数据保持寄存器数值相加，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO1，然后更新 LFS1 计数器。

同时，相同的三角波幅值与对应数据保持寄存器数值相加，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO2，然后更新 LFS2 计数器。

10.4.11 使用不同三角波发生器的同时触发

配置 DAC 工作在此转换模式:

■ 分别开启触发使能位，配置 T1EN 和 T2EN 为‘1’；
■ 分别设置 DAC 通道使用相同触发源，配置 T1SEL[2:0]和 T2SEL[2:0]位为相同值；
■ DAC 通道的 WxEN [1:0]位都配置为‘1 x’，并设 MAxSEL[3:0]为不同的三角波幅值。
■ 将双 DAC 通道转换数据装入到对应的对齐数据保持寄存器中（DR12DCH、DL12DCH 或 DR8DCH）。

触发事件发生时，MA1SEL[3:0]所设的三角波幅值与对应的数据保持寄存器数值相加，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO1，然后更新 LFS1 计数器。

同时，MA1SEL [3:0]所设的三角波幅值与对应的数据保持寄存器数值相加，结果将在延迟 3 个 APB1 时钟周期后传入寄存器 DAC_DATO2，然后更新 LFS2 计数器。

10.5 DAC 寄存器

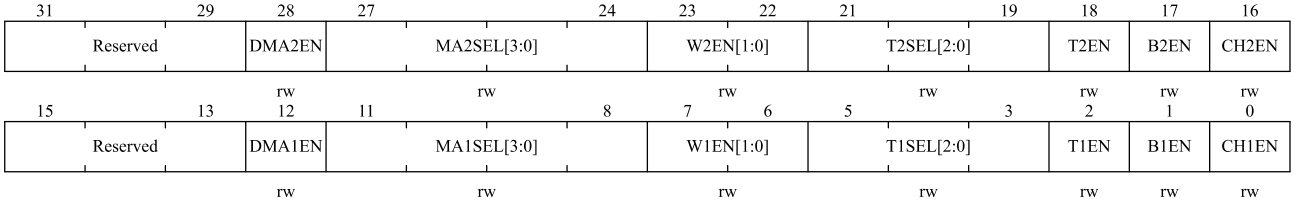
10.5.1 DAC 寄存器地址映像

表 10-3 DAC 寄存器地址映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	DAC_CTRL	Reserved			DMA2EN	MA2SEL[3:0]				W2EN[1:0]		T2SEL[2:0]			T2EN	B2EN	CH2EN	Reserved			DMA1EN	MA1SEL[3:0]				W1EN[1:0]		T1SEL[2:0]			T1EN	B1EN	CH1EN
	Reset Value				0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	DAC_SOTTR	Reserved																														TR2EN	TR1EN
	Reset Value																															0	0
008h	DAC_DR12CH1	Reserved																				DACCH1D[11:0]											
	Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0
00Ch	DAC_DL12CH1	Reserved																DACCH1D[11:0]														Reserved	
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0				
010h	DAC_DR8CH1	Reserved																								DACCH1D[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0
014h	DAC_DR12CH2	Reserved																				DACCH2D[11:0]											
	Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0
018h	DAC_DL12CH2	Reserved																DACCH2D[11:0]														Reserved	
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0				
01Ch	DAC_DR8CH2	Reserved																								DACCH2D[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0
020h	DAC_DR12DCH	Reserved			DACCH2D[11:0]													Reserved				DACCH1D[11:0]											
	Reset Value				0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
024h	DAC_DL12DCH	DACCH2D[11:0]												Reserved				DACCH1D[11:0]														Reserved	
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0				
028h	DAC_DR8DCH	Reserved																DACCH2D[7:0]								DACCH1D[7:0]							
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	DAC_DAT01	Reserved																				DACCH1DO[11:0]											
	Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0
030h	DAC_DAT02	Reserved																				DACCH2DO[11:0]											
	Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0

10.5.2 DAC 控制寄存器（DAC_CTRL）

偏移地址：0x00
复位值：0x0000 0000



位域	名称	描述
31:29	Reserved	保留，硬件强制为 0。
28	DMA2EN	DAC 通道 2DMA 使能位。

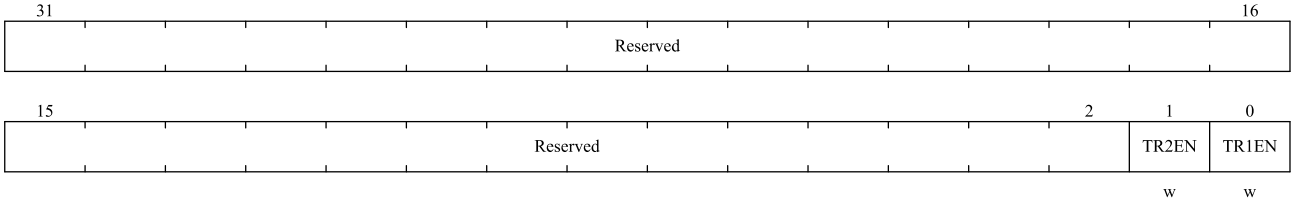
位域	名称	描述
		软件配置。 0：关闭 DAC 通道 2 DMA 模式 1：开启 DAC 通道 2 DMA 模式
27:24	MA2SEL[3:0]	DAC 通道 2 屏蔽/幅值选择器。 由软件配置这些位，用来在噪声生成模式下选择屏蔽位，在三角波生成模式下选择波形的幅值。 0000：不屏蔽 LSFR 位 0 / 三角波幅值等于 1 0001：不屏蔽 LSFR 位[1:0] / 三角波幅值等于 3 0010：不屏蔽 LSFR 位[2:0] / 三角波幅值等于 7 0011：不屏蔽 LSFR 位[3:0] / 三角波幅值等于 15 0100：不屏蔽 LSFR 位[4:0] / 三角波幅值等于 31 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
23:22	W2EN[1:0]	DAC 通道 2 噪声/三角波生成使能。 软件置 1 和清零。 00：关闭波形发生器 01：使能噪声波形发生器 1x：使能三角波发生器
21:19	T2SEL[2:0]	DAC 通道 2 触发选择。 该 3 位用于选择 DAC 通道 2 的外部触发事件 000：TIM6 TRGO 事件 001：TIM8 TRGO 事件 010：TIM7 TRGO 事件 011：TIM5 TRGO 事件 100：TIM2 TRGO 事件 101：TIM4 TRGO 事件 110：外部中断线 9 111：软件触发 *注意：该 3 位只能在 T2EN= 1（DAC 通道 2 触发使能）时设置。*
18	T2EN	DAC 通道 2 触发使能。 软件置 1 和清零，用来使能/关闭 DAC 通道 2 的触发。 0：关闭 DAC 通道 2 触发，写入 DAC 对其数据保持寄存器的数据在 1 个 APB1 时钟周期后传入寄存器 DAC_DATO2 1：使能 DAC 通道 2 触发，写入 DAC 对其数据保持寄存器的数据在 3 个 APB1 时钟周期后传入寄存器 DAC_DATO2 *注意：如果选择软件触发，写入 DAC 对其数据保持寄存器的数据只需要 1 个 APB1 时钟周期就可以传入寄存器 DAC_DATO2。*

位域	名称	描述
17	B2EN	使能 DAC 通道 2 输出缓存。 该位由软件置 1 和清零，用来使能/关闭 DAC 通道 2 的输出缓存。 0：关闭 DAC 通道 2 输出缓存 1：使能 DAC 通道 2 输出缓存
16	CH2EN	DAC 通道 2 使能。 该位由软件置 1 和清零，用来使能/关闭 DAC 通道 2。 0：关闭 DAC 通道 2 1：使能 DAC 通道 2
15:13	Reserved	保留，硬件强制为 0。
12	DMA1EN	DAC 通道 1 DMA 使能 该位由软件置 1 和清零。 0：关闭 DAC 通道 1 DMA 模式 1：使能 DAC 通道 1 DMA 模式
11:8	MA1SEL[3:0]	DAC 通道 1 屏蔽/幅值选择器。 由软件配置这些位，用来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
7:6	W1EN[1:0]	DAC 通道 1 噪声/三角波生成使能。 该 2 位由软件置 1 和清零。 00：关闭波形生成 01：使能噪声波形发生器 1x：使能三角波发生器
5:3	T1SEL[2:0]	DAC 通道 1 触发选择。 该位用于选择 DAC 通道 1 的外部触发事件。 000：TIM6 TRGO 事件 001：TIM8 TRGO 事件 010：TIM7 TRGO 事件 011：TIM5 TRGO 事件 100：TIM2 TRGO 事件 101：TIM4 TRGO 事件 110：外部中断线 9

位域	名称	描述
		111：软件触发 *注意：该位只能在 T1EN= 1（DAC 通道 1 触发使能）时设置。*
2	T1EN	DAC 通道 1 触发使能。 该位由软件置 1 和清零，用来使能/关闭 DAC 通道 1 的触发。 0：关闭 DAC 通道 1 触发，写入 DAC 对其数据保持寄存器的数据在 1 个 APB1 时钟周期后传入寄存器 DAC_DATO1 1：使能 DAC 通道 1 触发，写入 DAC 对其数据保持寄存器的数据在 3 个 APB1 时钟周期后传入寄存器 DAC_DATO1 *注意：如果选择软件触发，写入 DAC 对其数据保持寄存器的数据只需要 1 个 APB1 时钟周期就可以传入寄存器 DAC_DATO1。*
1	B1EN	使能 DAC 通道 1 输出缓存。 该位由软件置 1 和清零，用来使能/关闭 DAC 通道 1 的输出缓存。 0：关闭 DAC 通道 1 输出缓存 1：使能 DAC 通道 1 输出缓存
0	CH1EN	DAC 通道 1 使能。 该位由软件置 1 和清零，用来使能/禁能 DAC 通道 1。 0：关闭 DAC 通道 1 1：使能 DAC 通道 1

10.5.3 DAC 软件触发寄存器（DAC_SOTTR）

偏移地址：0x04
复位值：0x0000 0000

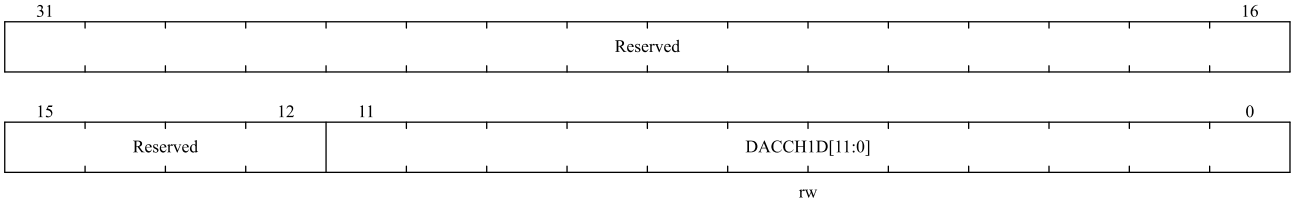


位域	名称	描述
31:2	Reserved	保留，硬件强制为 0。
1	TR2EN	DAC 通道 2 软件触发 . 该位由软件置 1 和清零，用来使能/关闭软件触发。 0：关闭 DAC 通道 2 软件触发; 1：使能 DAC 通道 2 软件触发。 *注意：一旦 DAC 对其数据保持寄存器的数据传入寄存器 DAC_DATO2，（1 个 APB1 时钟周期后）该位由硬件置'0'。*
0	TR1EN	DAC 通道 1 软件触发 . 该位由软件置 1 和清零，用来使能/关闭软件触发。 0：关闭 DAC 通道 1 软件触发; 1：使能 DAC 通道 1 软件触发。 *注意：一旦 DAC 对其数据保持寄存器的数据传入寄存器 DAC_DATO1，（1*

位域	名称	描述
		个 APB1 时钟周期后）该位由硬件置'0'。

10.5.4 DAC 通道 1 的 12 位右对齐数据保持寄存器（DAC_DR12CH1）

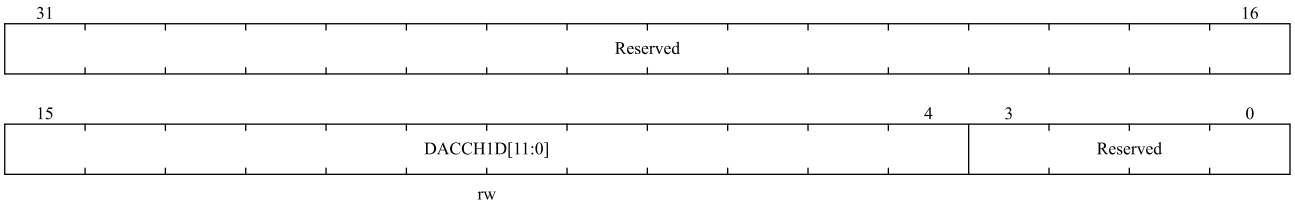
偏移地址：0x08
复位值：0x0000 0000



位域	名称	描述
31:12	Reserved	保留，硬件强制为 0。
11:0	DACCH1D[11:0]	DAC 通道 1 的 12 位右对齐数据 该位由软件写入，表示 DAC 通道 1 的 12 位数据。

10.5.5 DAC 通道 1 的 12 位左对齐数据保持寄存器（DAC_DL12CH1）

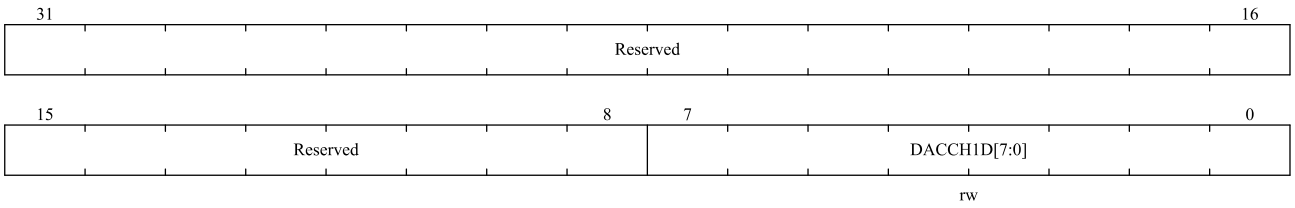
偏移地址：0x0c
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:4	DACCH1D[11:0]	DAC 通道 1 的 12 位左对齐数据 该位由软件写入，表示 DAC 通道 1 的 12 位数据。
3:0	Reserved	保留，硬件强制为 0。

10.5.6 DAC 通道 1 的 8 位右对齐数据保持寄存器（DAC_DR8CH1）

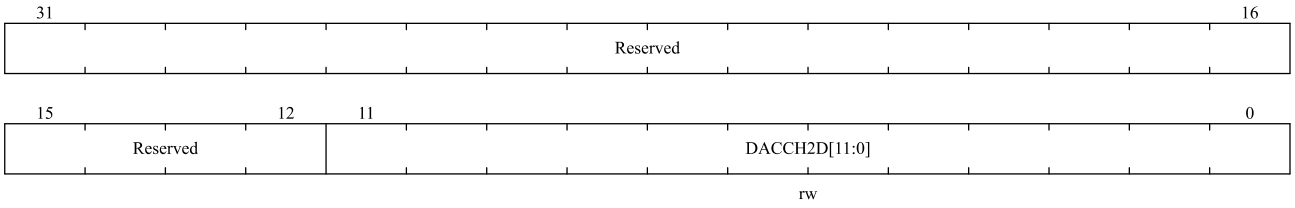
偏移地址：0x10
复位值：0x0000 0000



位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7:0	DACCH1D[7:0]	DAC 通道 1 的 8 位右对齐数据 该位由软件写入，表示 DAC 通道 1 的 8 位数据。

10.5.7 DAC 通道 2 的 12 位右对齐数据保持寄存器（DAC_DR12CH2）

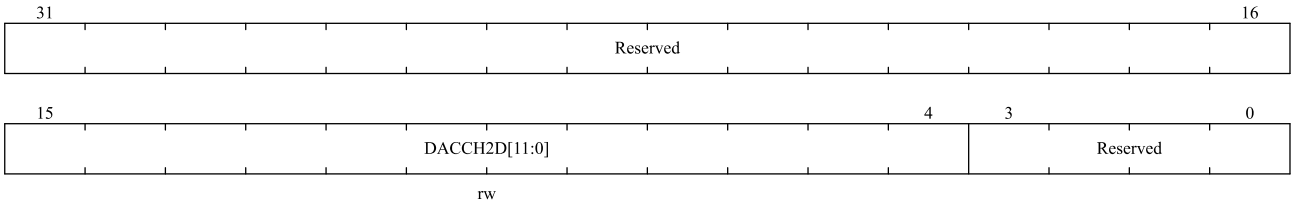
偏移地址：0x14
复位值：0x0000 0000



位域	名称	描述
31:12	Reserved	保留，硬件强制为 0。
11:0	DACCH2D[11:0]	DAC 通道 2 的 12 位右对齐数据 该位由软件写入，表示 DAC 通道 2 的 12 位数据。

10.5.8 DAC 通道 2 的 12 位左对齐数据保持寄存器（DAC_DL12CH2）

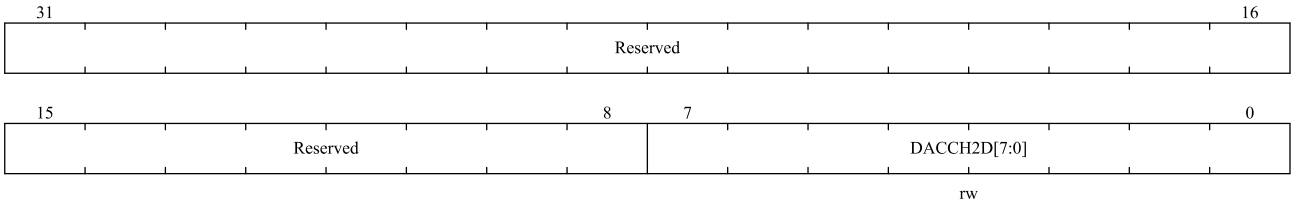
偏移地址：0x18
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:4	DACCH2D[11:0]	DAC 通道 2 的 12 位左对齐数据 该位由软件写入，表示 DAC 通道 2 的 12 位数据。
3:0	Reserved	保留，硬件强制为 0。

10.5.9 DAC 通道 2 的 8 位右对齐数据保持寄存器（DAC_DR8CH2）

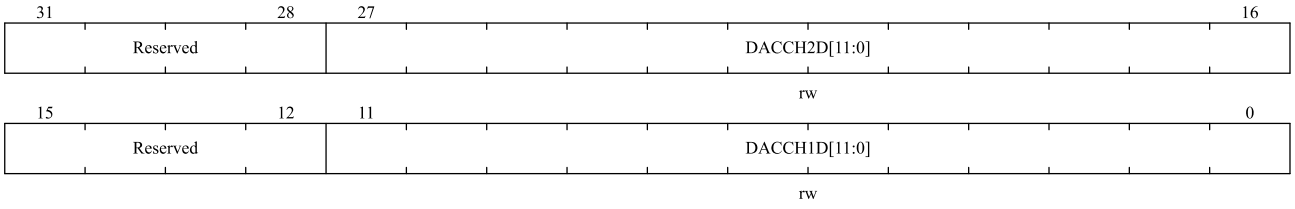
偏移地址：0x1C
复位值：0x0000 0000



位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7:0	DACCH2D[7:0]	DAC 通道 2 的 8 位右对齐数据 该位由软件写入，表示 DAC 通道 2 的 8 位数据。

10.5.10 双 DAC 的 12 位右对齐数据保持寄存器（DAC_DR12DCH）

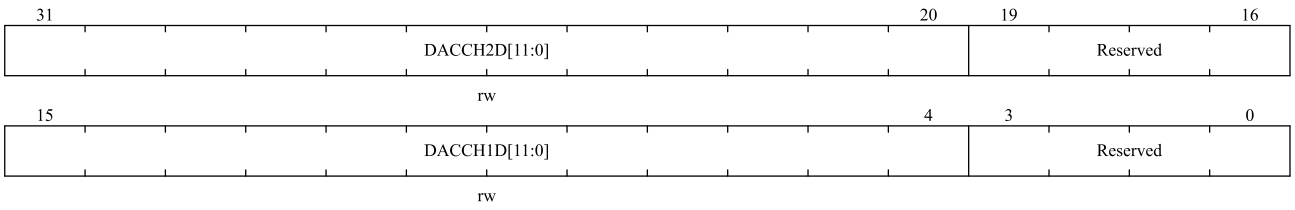
偏移地址：0x20
复位值：0x0000 0000



位域	名称	描述
31:28	Reserved	保留，硬件强制为 0。
27:16	DACCH2D[11:0]	DAC 通道 2 的 12 位右对齐数据。 该位由软件写入，表示 DAC 通道 2 的 12 位数据。
15:12	Reserved	保留，硬件强制为 0。
11:0	DACCH1D[11:0]	DAC 通道 1 的 12 位右对齐数据。 该位由软件写入，表示 DAC 通道 1 的 12 位数据。

10.5.11 双 DAC 的 12 位左对齐数据保持寄存器（DAC_DL12DCH）

偏移地址：0x24
复位值：0x0000 0000

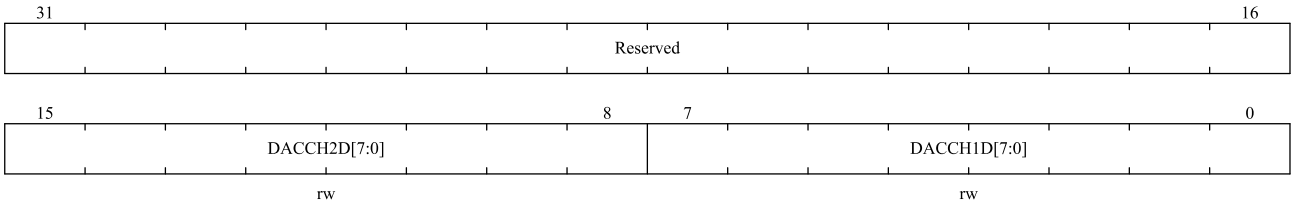


位域	名称	描述
31:20	DACCH2D[11:0]	DAC 通道 2 的 12 位左对齐数据。 该位由软件写入，表示 DAC 通道 2 的 12 位数据。
19:16	Reserved	保留，硬件强制为 0。

位域	名称	描述
15:4	DACCH1D[11:0]	DAC 通道 1 的 12 位左对齐数据。 该位由软件写入，表示 DAC 通道 1 的 12 位数据。
3:0	Reserved	保留，硬件强制为 0。

10.5.12 双 DAC 的 8 位右对齐数据保持寄存器（DAC_DR8DCH）

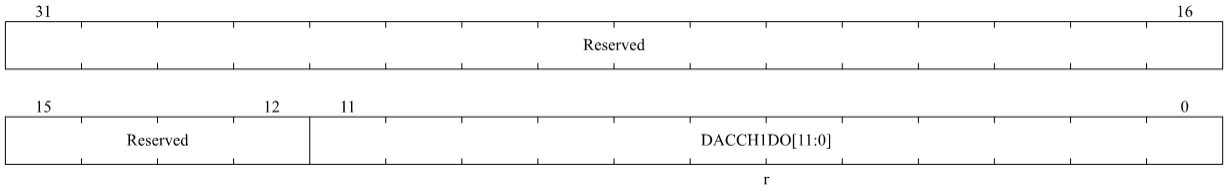
偏移地址：0x28
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:8	DACCH2D[7:0]	DAC 通道 2 的 8 位右对齐数据。 该位由软件写入，表示 DAC 通道 2 的 8 位数据。
7:0	DACCH1D[7:0]	DAC 通道 1 的 8 位右对齐数据。 该位由软件写入，表示 DAC 通道 1 的 8 位数据。

10.5.13 DAC 通道 1 数据输出寄存器（DAC_DAT01）

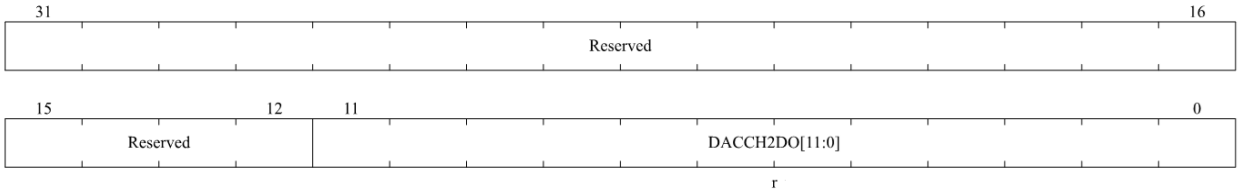
偏移地址：0x2C
复位值：0x0000 0000



位域	名称	描述
31:12	Reserved	保留，硬件强制为 0。
11:0	DACCH1D[11:0]	DAC 通道 1 输出数据。 只读，表示 DAC 通道 1 的输出数据

10.5.14 DAC 通道 2 数据输出寄存器（DAC_DAT02）

偏移地址：0x30
复位值：0x0000 0000



位域	名称	描述
31:12	Reserved	保留，硬件强制为 0。
11:0	DACCH2D[11:0]	DAC 通道 2 输出数据。 只读，表示 DAC 通道 2 的输出数据

11 高级控制定时器（TIM1 和 TIM8）

11.1 TIM1 和 TIM8 简介

高级控制定时器（TIM1 和 TIM8）由一个 16 位的自动装载计数器、一个可编程的重复计数器和一个可编程预分频计数器组成。

高级控制定时器可以用于测量输入信号的脉冲宽度（输入捕获）、产生输出波形（输出比较、PWM、嵌入死区时间的互补 PWM 等）以及对输入信号进行计数。

结合定时器预分频器和 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

高级控制定时器（TIM1 和 TIM8）和通用定时器（TIMx）不共享任何资源，完全独立，可以实现异步或同步操作，具体描述参看 11.3.20 节。

11.2 TIM1 和 TIM8 主要特性

TIM1 和 TIM8 定时器的功能包括:

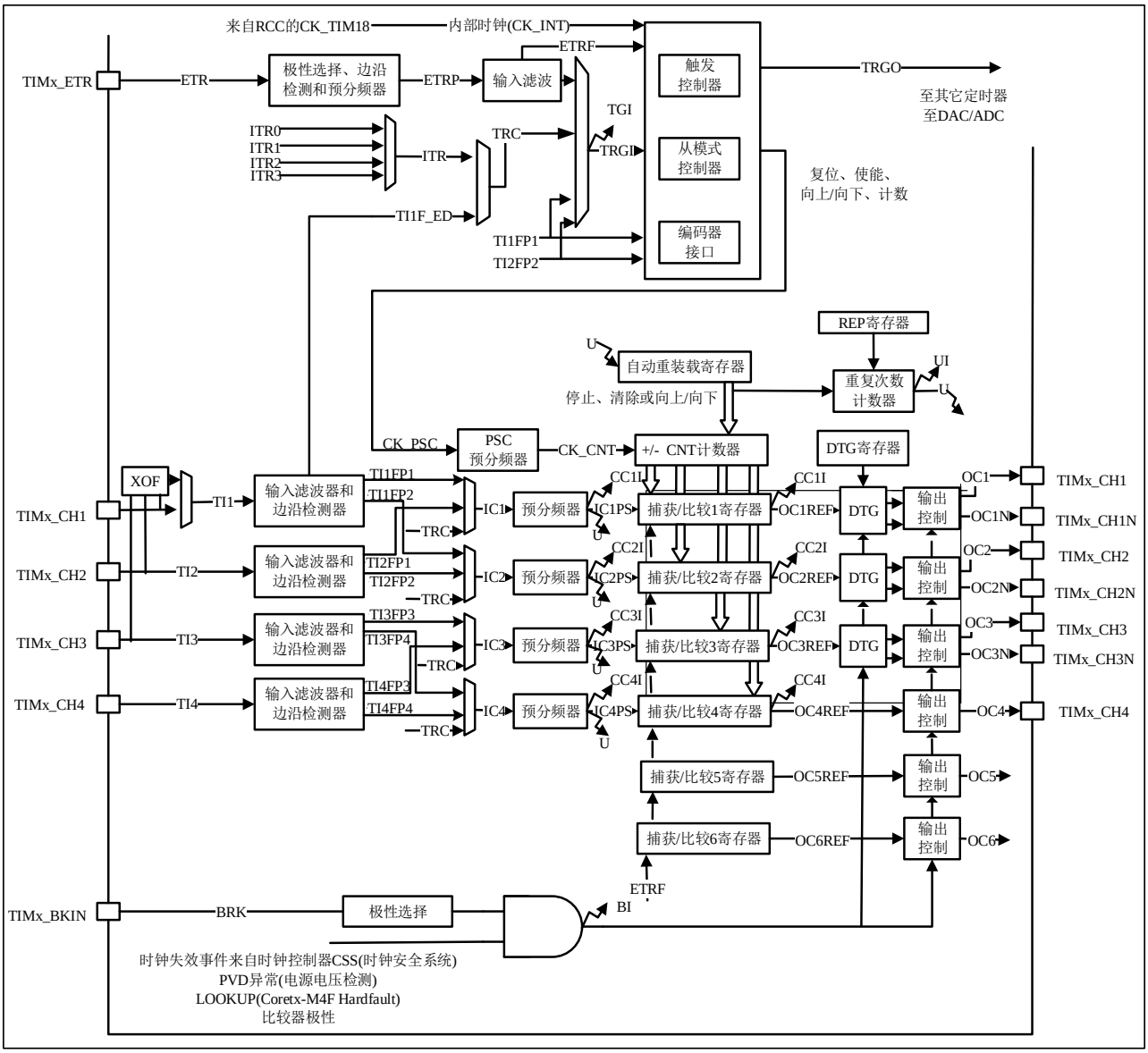
- 16 位向上、向下、向上/下自动装载计数器
- 16 位可编程预分频器，计数器时钟频率的分频系数为 1～65535 之间的任意数值
- 多达 4 个独立通道:
  - 输入捕获
  - 输出比较
  - PWM 生成（边缘或中间对齐模式）
  - 单脉冲模式输出
- TIM1_CC5 和 TIM8_CC5 用于 COMP 消隐，
- TIM1_CC6 用于 OPAMP1 和 OPAMP2 的输入通道切换，TIM8_CC6 用于 OPAMP3 和 OPAMP4 的输入通道切换
- 死区时间可编程的互补输出
- 外部信号结合内置同步电路对多个定时器进行统一控制
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 刹车输入信号可以将定时器输出信号置于复位状态或者一个已知状态
- 支持 IOM、CSS、PVD、比较器输出进行 PWM 刹车
- 支持捕获 IO 信号或比较器输出信号
- 如下事件发生时产生中断/DMA:
  - 更新：计数器向上溢出/向下溢出，计数器初始化（通过软件或者内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或者由内部/外部触发计数）

- ◆ 输入捕获
- ◆ 输出比较
- ◆ 刹车信号输入

■ 支持针对定位的增量（正交）编码器和霍尔传感器电路

■ 触发输入作为外部时钟或者按周期进行的通用管理

图 11-1 高级控制定时器框图



事件

中断和 DMA 输出

捕获通道 1 输入可以来自 IOM 或比较器输出

清除 OCxRef 可以来自 ETR 或比较器输出

11.3 TIM1 和 TIM8 功能描述

11.3.1 时基单元

高级定时器由预分频器、计数器、自动装载寄存器以及重复计数器组成。计数器可以向上计数、向下计数或者向上向下双向计数。预分频器分频后的时钟输入计数器。

在时基单元工作时，软件仍可以随时对计数器、自动装载寄存器和预分频器寄存器进行读写。

- 时基单元包含：预分频器寄存器 （TIMx_PSC）
- 计数器寄存器（TIMx_CNT）
- 自动装载寄存器 （TIMx_AR）
- 重复次数寄存器 （TIMx_REPCNT）

自动装载寄存器具有预先装载功能，读写自动重装载寄存器将访问预装载寄存器。根据在 TIMx_CTRL1 寄存器中的自动装载预装载使能位（ARPEN）的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。以下条件产生更新事件：

- 当计数器达到溢出条件(向下计数时的下溢条件)并当 TIMx_CTRL1 寄存器中的 UPDIS 位等于 0 时，产生更新事件。
- 软件产生。

预分频器的时钟输出 CK_CNT 驱动计数器，仅当设置了计数器 TIMx_CTRL1 寄存器中的计数器使能位（CNTEN）时，CK_CNT 才有效。(更多有关使能计数器的细节，请参见控制器的从模式描述)。

注意，在设置了 TIMx_CTRL 寄存器的 CNTEN 位的一个时钟周期后，计数器开始计数。

11.3.1.1 预分频器描述

预分频器（TIMx_PSC 寄存器）由 16 位计数器构成，可以将输入时钟频率按 1 到 65536 之间的任意值分频。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 11-2 和图 11-3 给出了在预分频器运行时，更改计数器参数的示例。

图 11-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

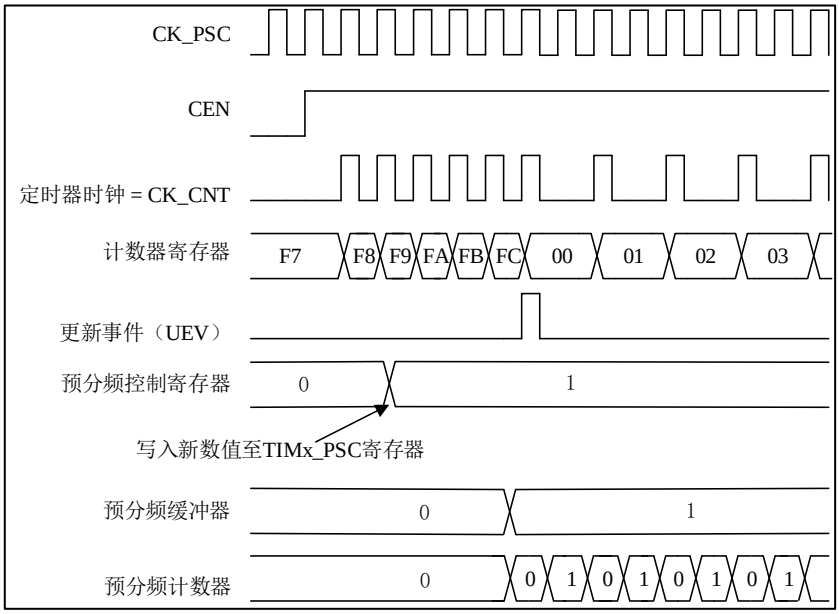
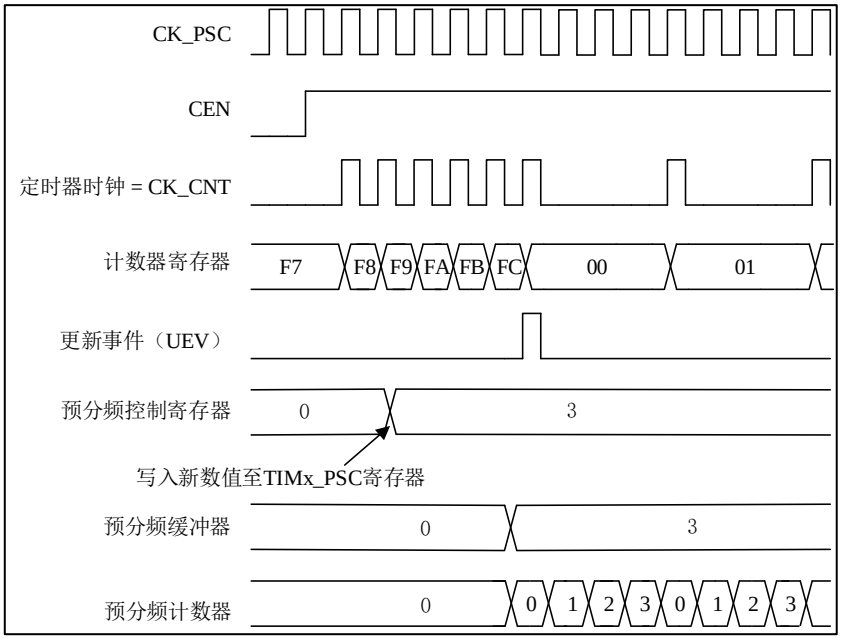


图 11-3 当预分频器的参数从 1 变到 4 时，计数器的时序图



11.3.2 计数器模式

11.3.2.1 向上计数模式

使用向上计数模式，计数器将从 0 计数到自动加载值（TIMx_AR 计数器的值），然后重新从 0 开始计数并且产生一个计数器溢出事件。

如果使用了重复计数器功能，在向上计数达到设置的重复计数次数（TIMx_REPCNT）时，产生更新事件（UEV）；否则每次计数器溢出时产生更新事件。

在 TIMx_EVTGEN 寄存器中（通过软件方式或者使用从模式控制器）设置 UDGN 位也同样可以产生一个更新事件。

为了避免在向预装载寄存器中写入新值时更新影子寄存器，可以设置 TIMx_CTRL1 寄存器中的 UPDIS 位，可以禁止更新事件。在 UPDIS 位被清'0'之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清'0'，同时预分频器的计数也被请 0（但预分频器的数值不变）。此外，如果设置了 TIMx_CTRL1 寄存器中的 UPRS 位（选择更新请求），设置 UDGN 位将产生一个更新事件 UEV，但硬件不设置 UDITF 标志（即不产生中断或 DMA 请求）。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

更新事件发生后，所有的寄存器都被更新，硬件同时（依据 UPRS 位）设置更新标志位（TIMx_STS 寄存器中的 UDITF 位）。

- 重复计数器被重新加载为 TIMx_REPCNT 寄存器的内容。
- 自动装载影子寄存器被重新置入预装载寄存器的值（TIMx_AR）。
- 预分频器的缓冲区被置入预装载寄存器的值（TIMx_PSC 寄存器的内容）。

下图给出一些示例，当 TIMx_AR=0x36 时计数器在不同时钟频率下的动作。

图 11-4 计数器时序图，内部时钟分频因子为 1

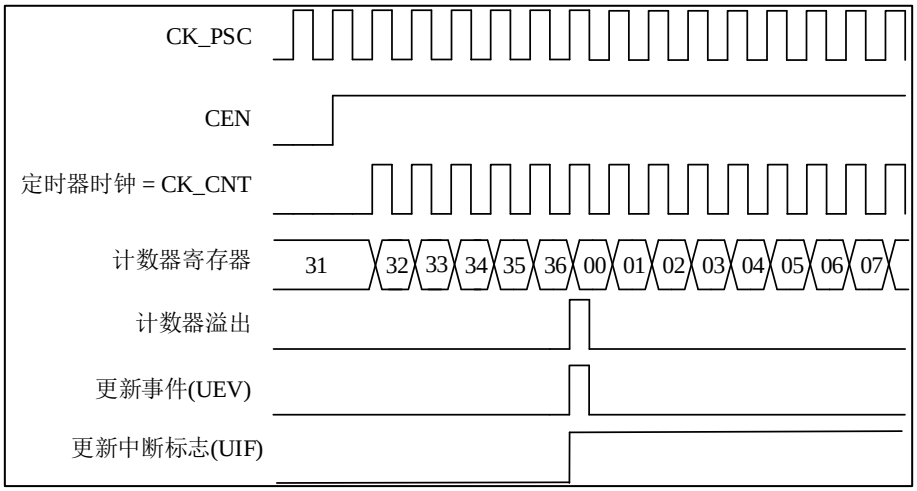


图 11-5 计数器时序图，内部时钟分频因子为 2

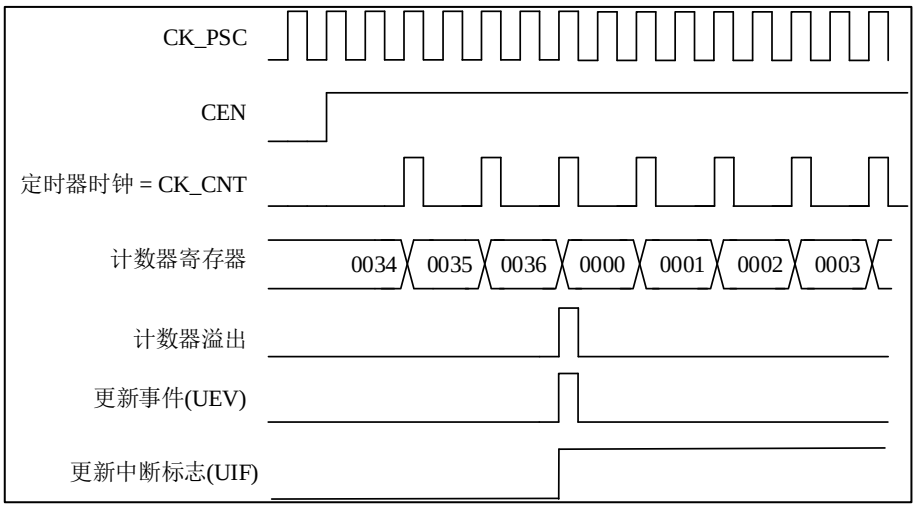


图 11-6 计数器时序图，内部时钟分频因子为 4

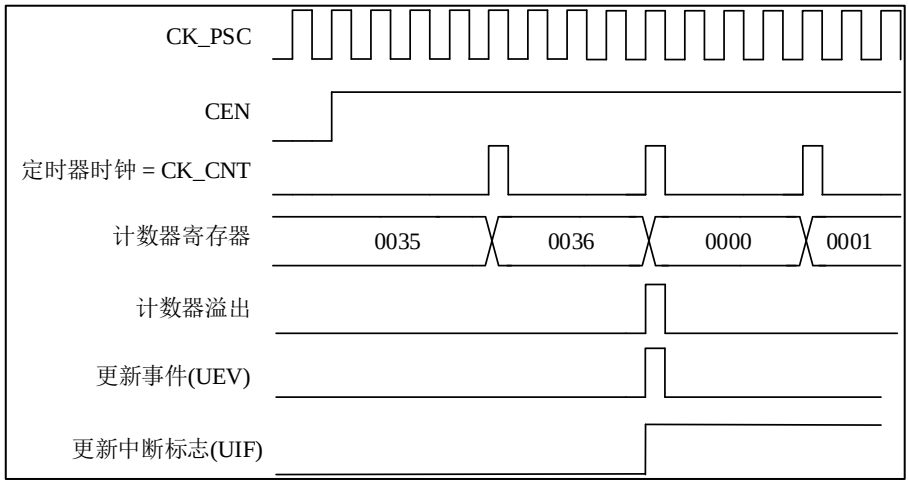


图 11-7 计数器时序图，内部时钟分频因子为 N

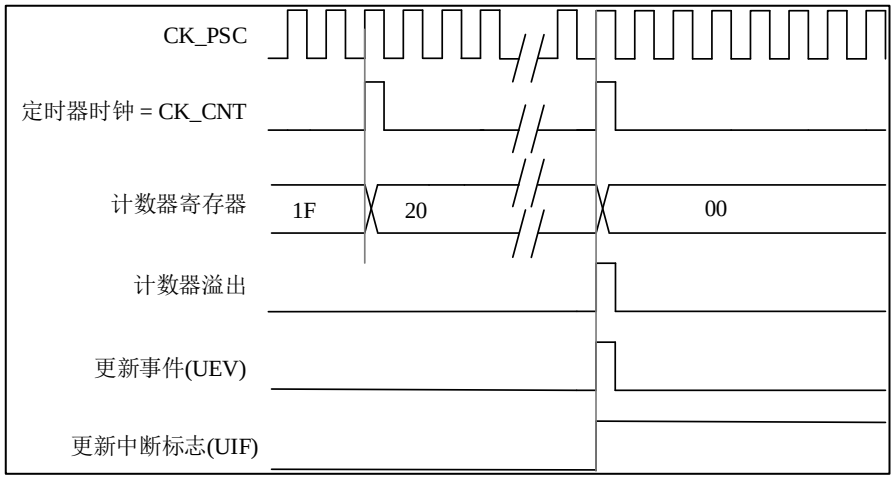


图 11-8 计数器时序图，当 ARPEN=0 时的更新事件（TIMx_AR 没有预装入）

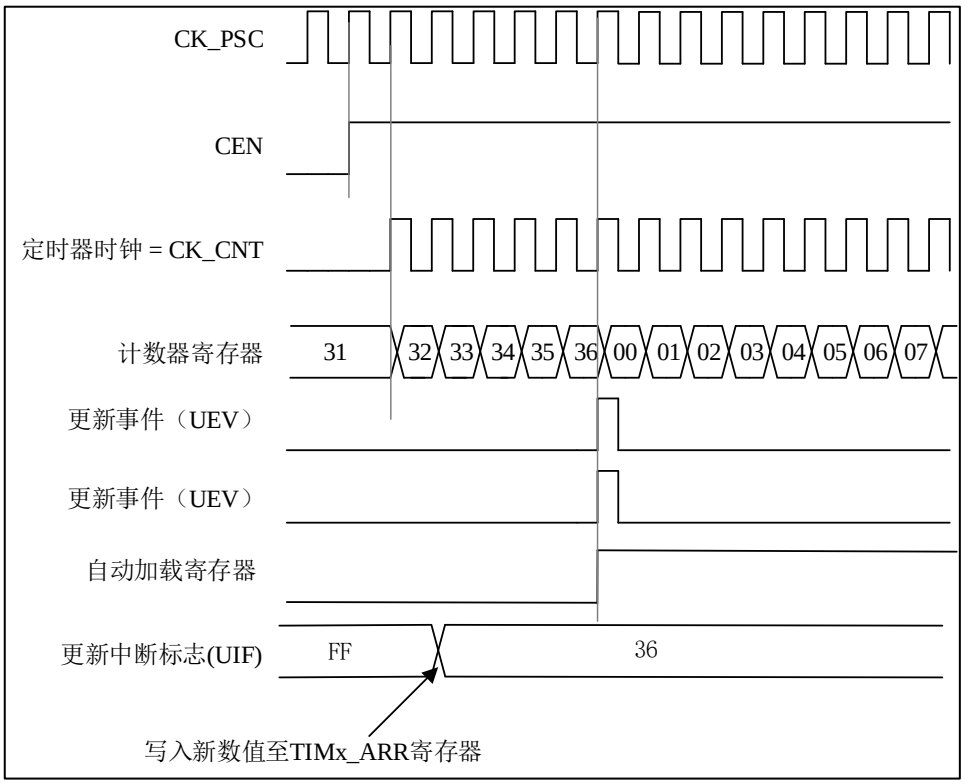
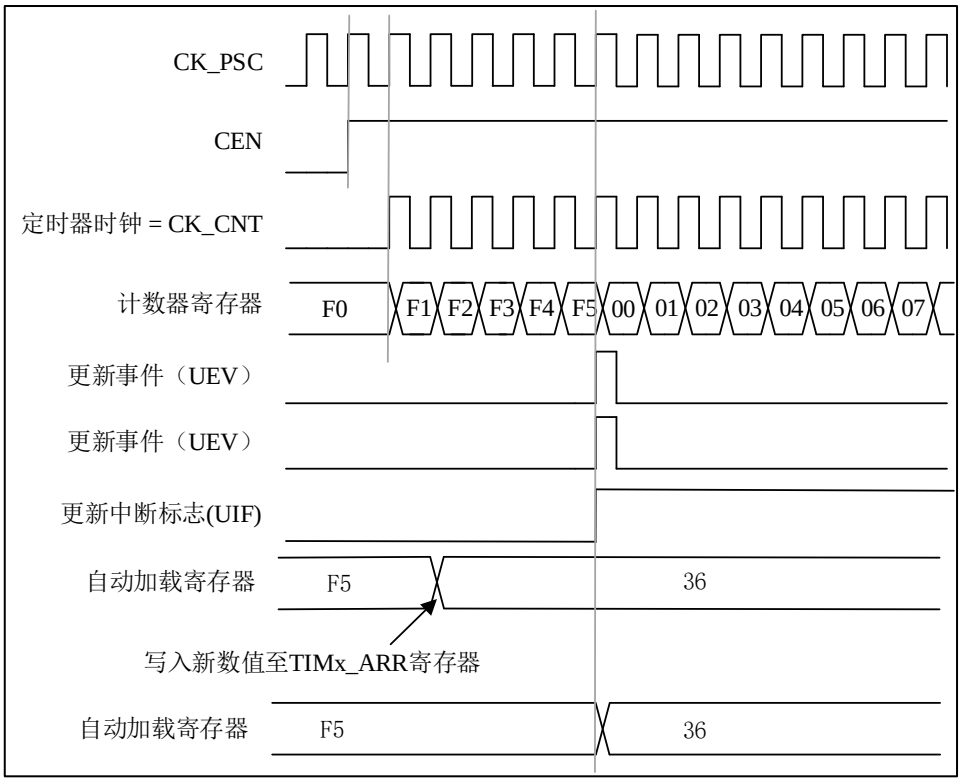


图 11-9 计数器时序图，当 ARPEN=1 时的更新事件（预装入了 TIMx_AR）



11.3.2.2 向下计数模式

如果使用向下计数模式，计数器将从自动装入的值（TIMx_AR 计数器的值）开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。

如果使用了重复计数器，当向下计数重复了重复计数寄存器（TIMx_REPCNT）中设定的次数后，将产生更新事件（UEV），否则每次计数器下溢时产生更新事件。

在 TIMx_EVTGEN 寄存器中（通过软件方式或者使用从模式控制器）设置 UDGN 位，也同样可以产生一个更新事件。

为了避免在向预装载寄存器中写入新值时更新影子寄存器，可以设置 TIMx_CTRL1 寄存器中的 UPDIS 位，可以禁止更新事件。因此 UPDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始（但预分频系数不变）。

此外，如果设置了 TIMx_CTRL1 寄存器中的 UPRS 位（选择更新请求），设置 UDGN 位将产生一个更新事件 UEV 但不设置 UDITF 标志（因此不产生中断和 DMA 请求），这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

更新事件发生后，所有的寄存器都被更新，硬件同时（依据 UPRS 位）设置更新标志位（TIMx_STS 寄存器中的 UDITF 位）。

- 重复计数器被重置为 TIMx_REPCNT 寄存器中的内容
- 预分频器的缓存器被加载为预装载的值（TIMx_PSC 寄存器的值）。

当前的自动加载寄存器被更新为预装载值（TIMx_AR 寄存器中的内容）。

注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。

以下是一些当 TIMx_AR=0x36 时，计数器在不同时钟频率下的操作示例。

图 11-10 计数器时序图，内部时钟分频因子为 1

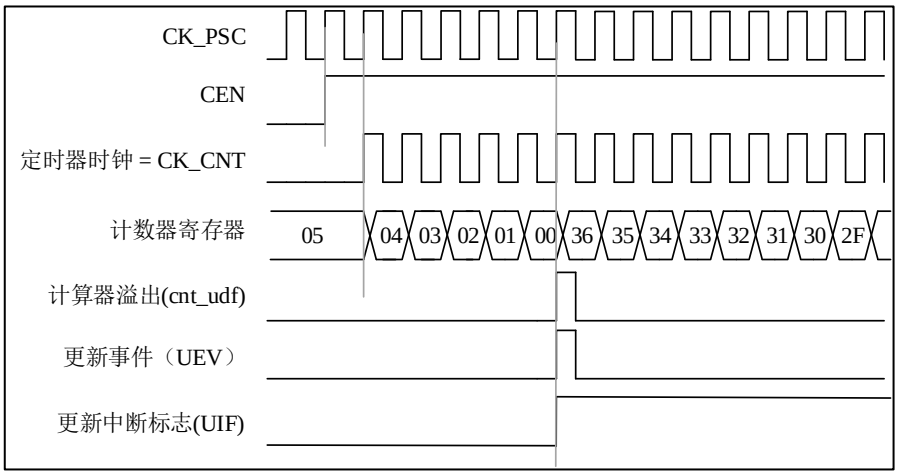


图 11-11 计数器时序图，内部时钟分频因子为 2

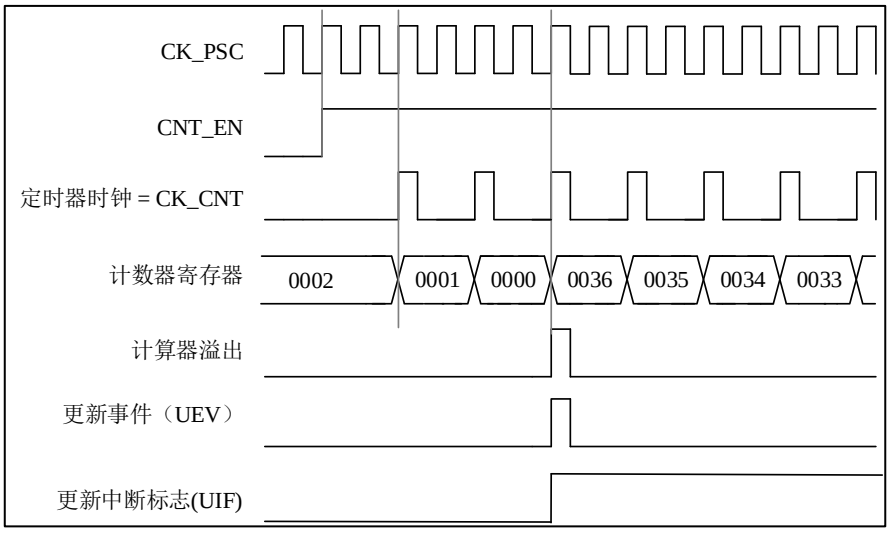


图 11-12 计数器时序图，内部时钟分频因子为 4

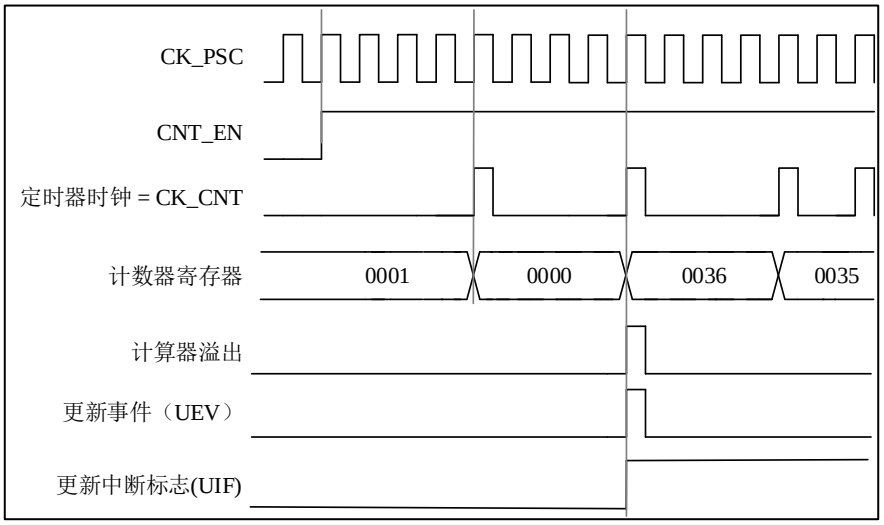


图 11-13 计数器时序图，内部时钟分频因子为 N

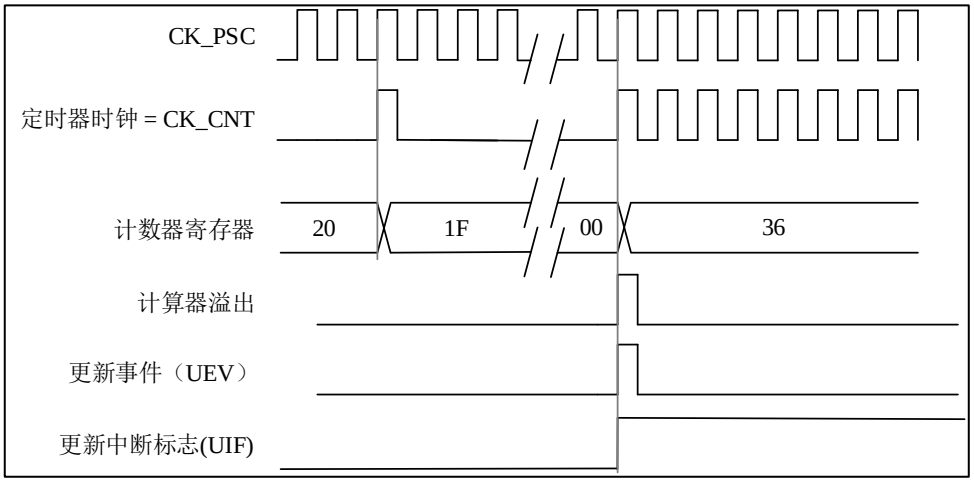
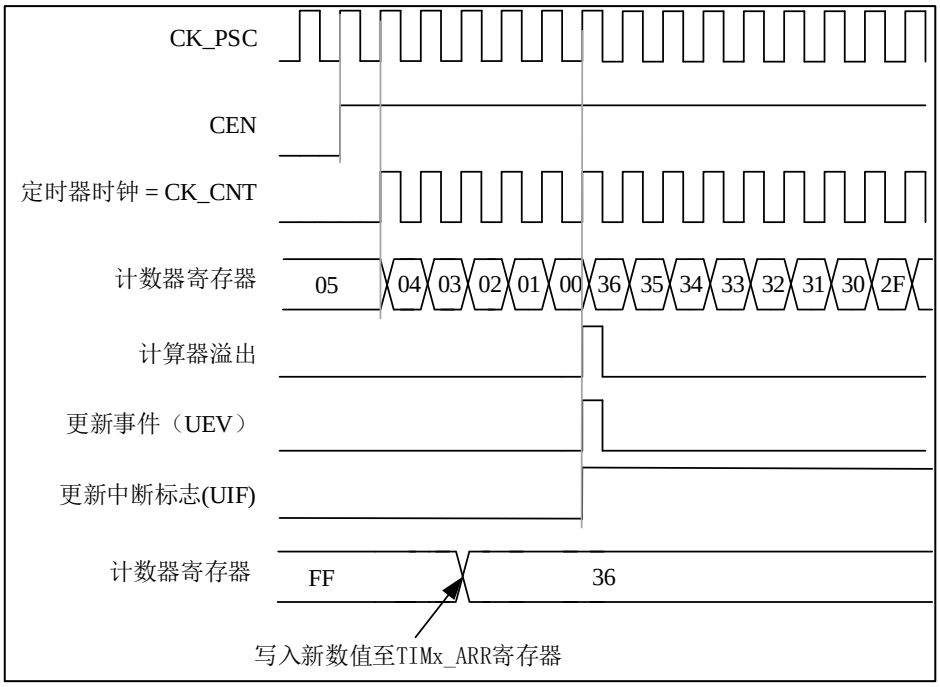


图 11-14 计数器时序图，当没有使用重复计数器时的更新事件



11.3.2.3 中央对齐模式（向上/向下计数）

如果使用中央对齐模式，计数器将从 0 开始计数到自动加载的值（TIMx_AR 寄存器）-1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

在此模式下，不能写入 TIMx_CTRL1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过（软件或者使用从模式控制器）设置 TIMx_EVTGEN 寄存器中的 UDGN 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

为了避免在向预装载寄存器中写入新值时更新影子寄存器，可以设置 TIMx_CTRL1 寄存器中的 UPDIS 位禁止 UEV 事件。因此 UPDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 TIMx_CTRL1 寄存器中的 UPRS 位（选择更新请求），设置 UDGN 位将产生一个更新事件 UEV 但不设置 UDITF 标志（因此不产生中断和 DMA 请求），这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

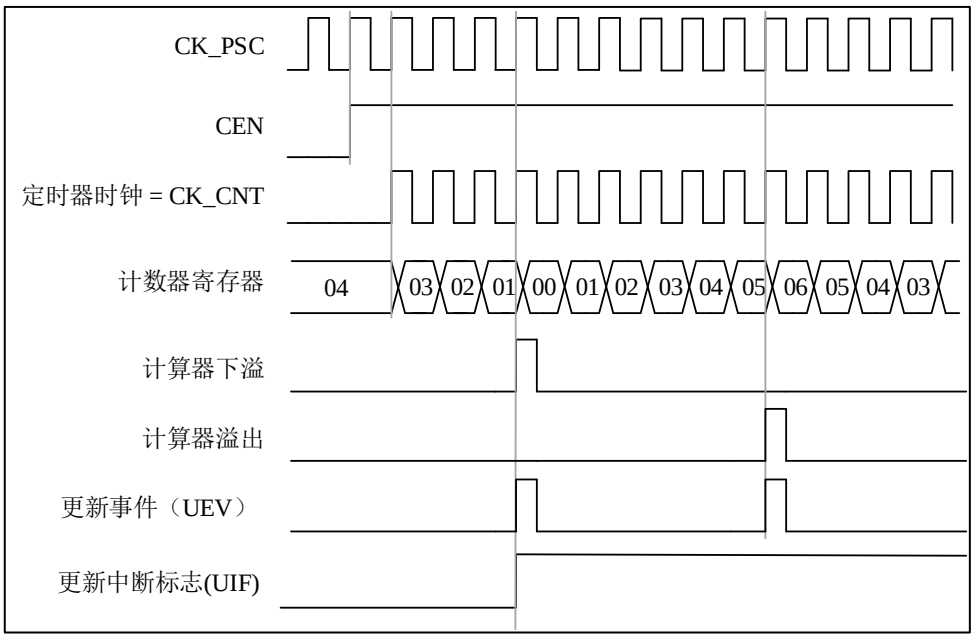
当发生更新事件时，所有的寄存器都被更新，并且（根据 UPRS 位的设置）更新标志位（TIMx_STS 寄存器中的 UDITF 位）也被设置。

- 重复计数器被重置为 TIMx_REPCNT 寄存器中的内容
- 预分频器的缓存器被加载为预装载（TIMx_PSC 寄存器）的值
- 当前的自动加载寄存器被更新为预装载值（TIMx_AR 寄存器中的内容）。

注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值（计数器被装载为新的值）。

以下是一些计数器在不同时钟频率下的操作的示例:

图 11-15 计数器时序图，内部时钟分频因子为 1，TIMx_AR=0x6



1． 这里使用了中心对齐模式 1（详见 11.4.2 节）。

图 11-16 计数器时序图，内部时钟分频因子为 2

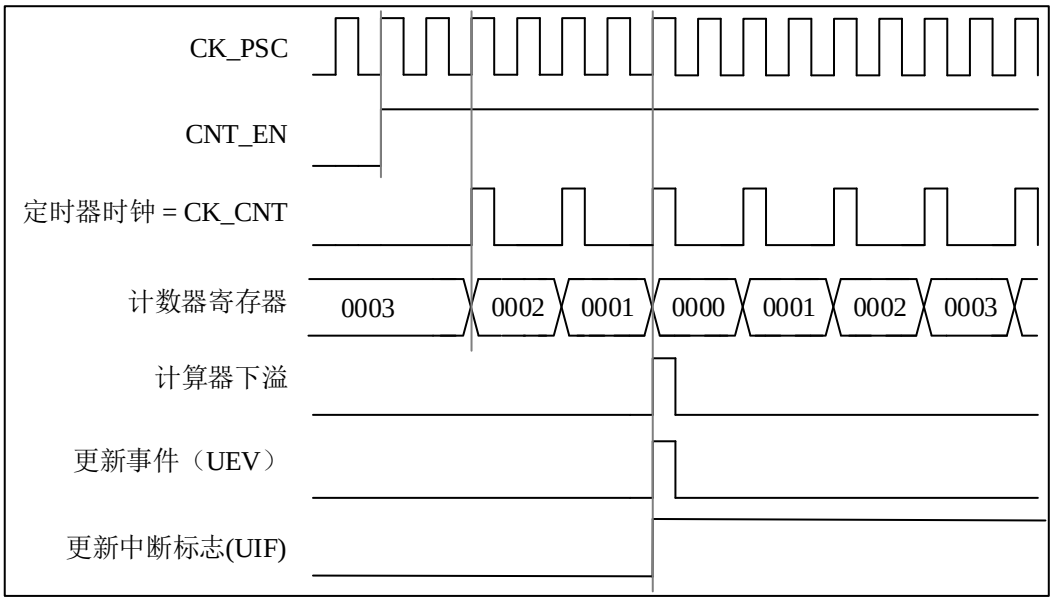


图 11-17 计数器时序图，内部时钟分频因子为 4，TIMx_AR=0x36

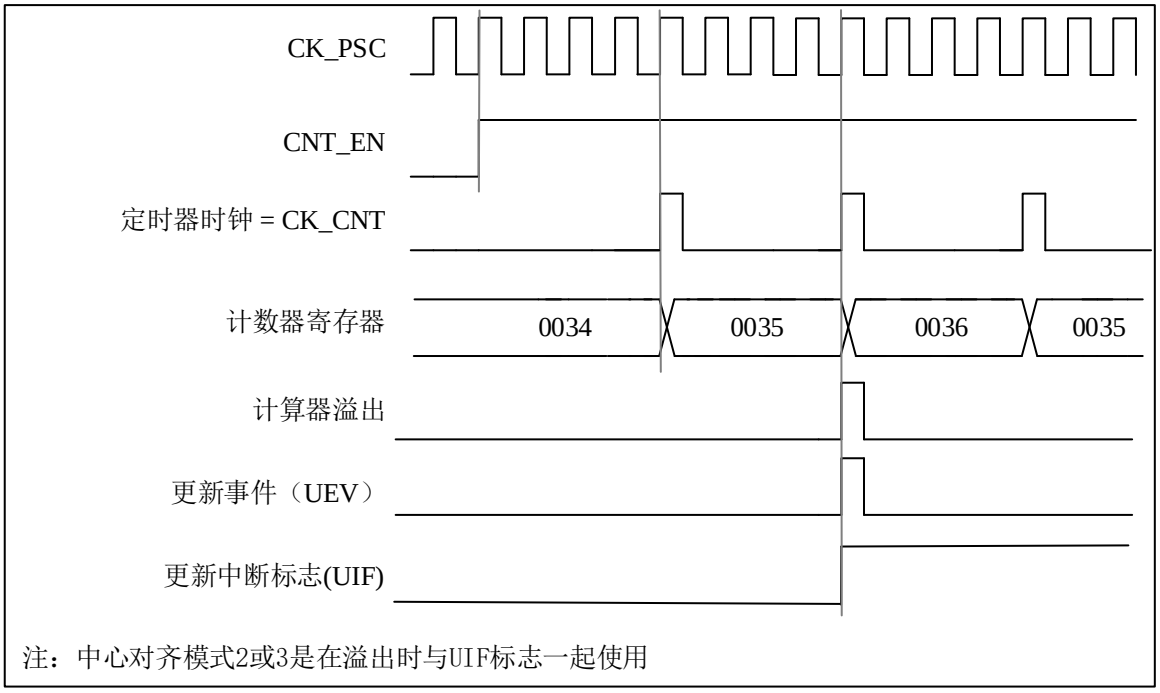


图 11-18 计数器时序图，内部时钟分频因子为 N

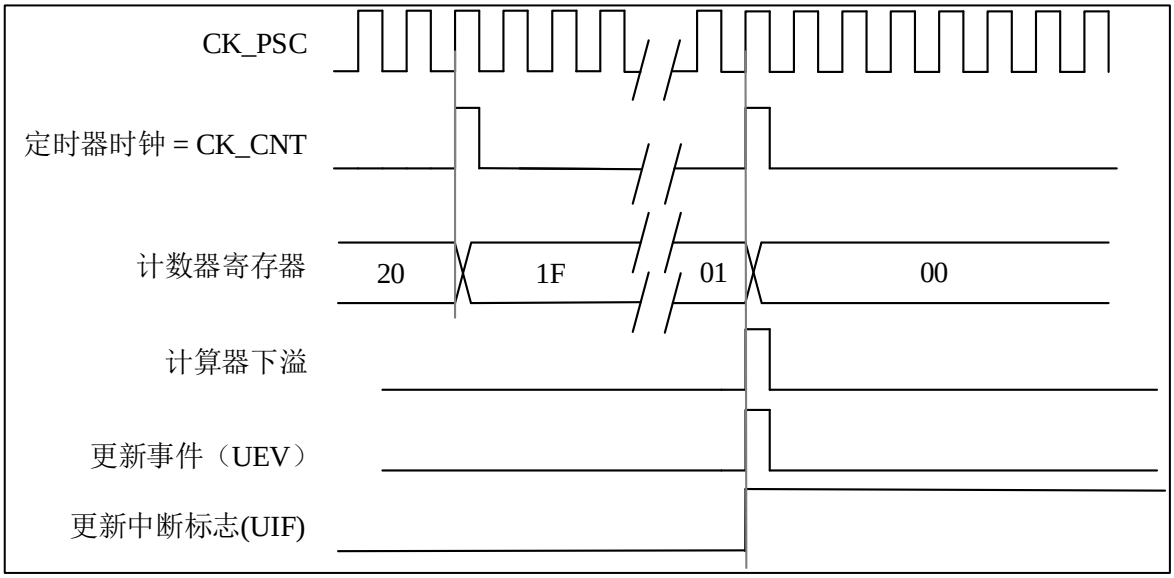


图 11-19 计数器时序图，ARPEN=1 时的更新事件（计数器下溢）

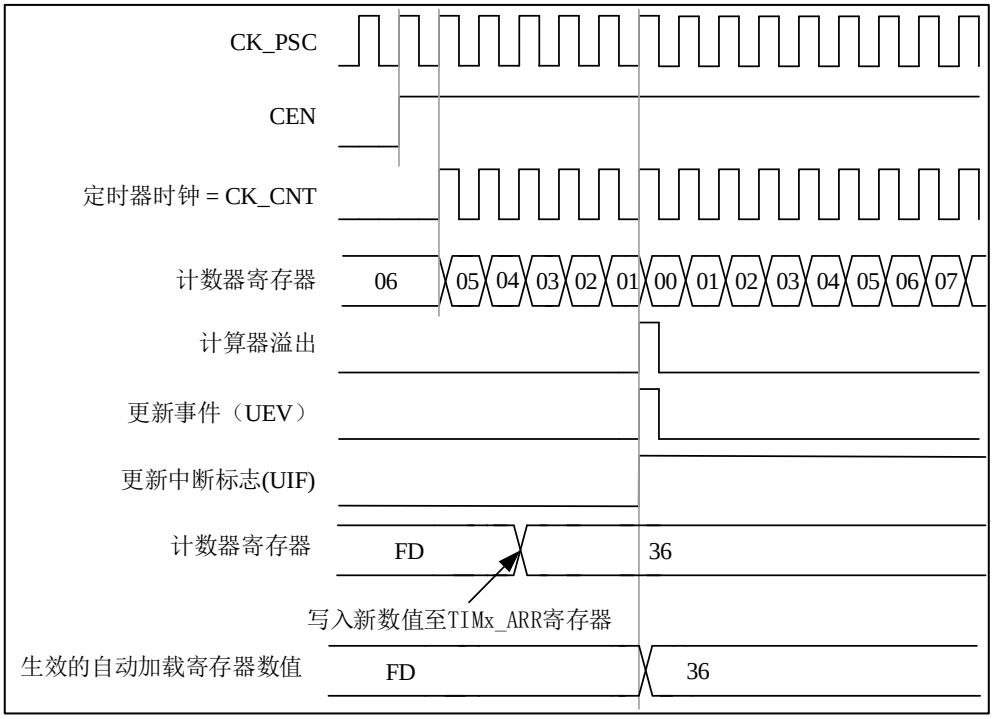
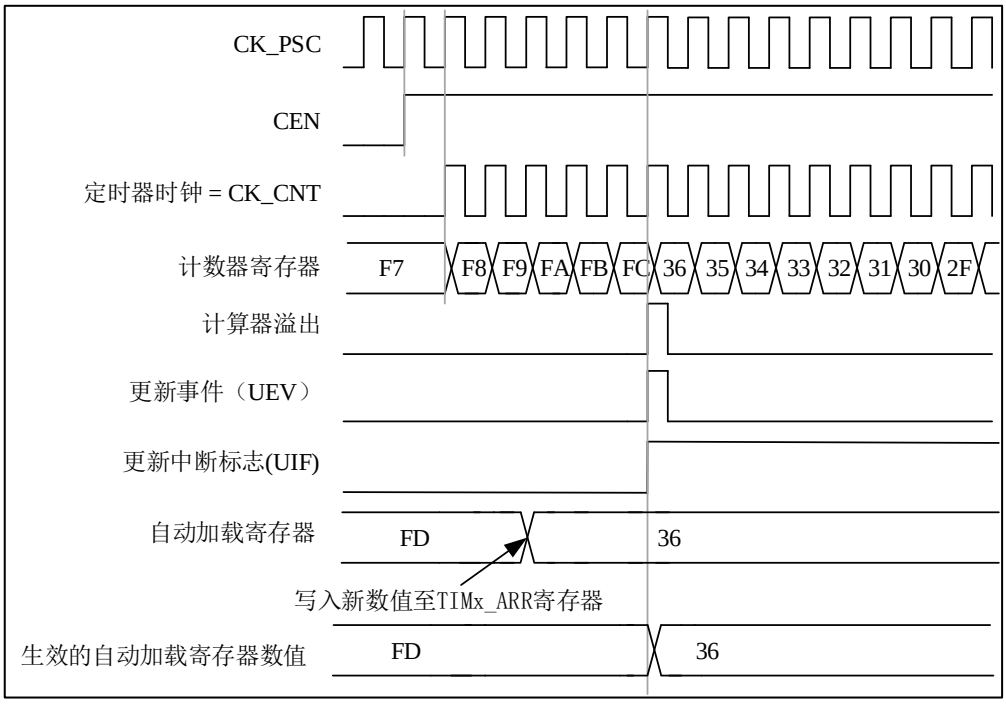


图 11-20 计数器时序图，ARPEN=1 时的更新事件（计数器溢出）



11.3.3 重复计数器

11.3.1 节“时基单元”解释了计数器上溢/下溢时更新事件（UEV）是如何产生的，然而事实上它只能在重复计数达到 0 的时候产生。这个特性对产生 PWM 信号非常有用。

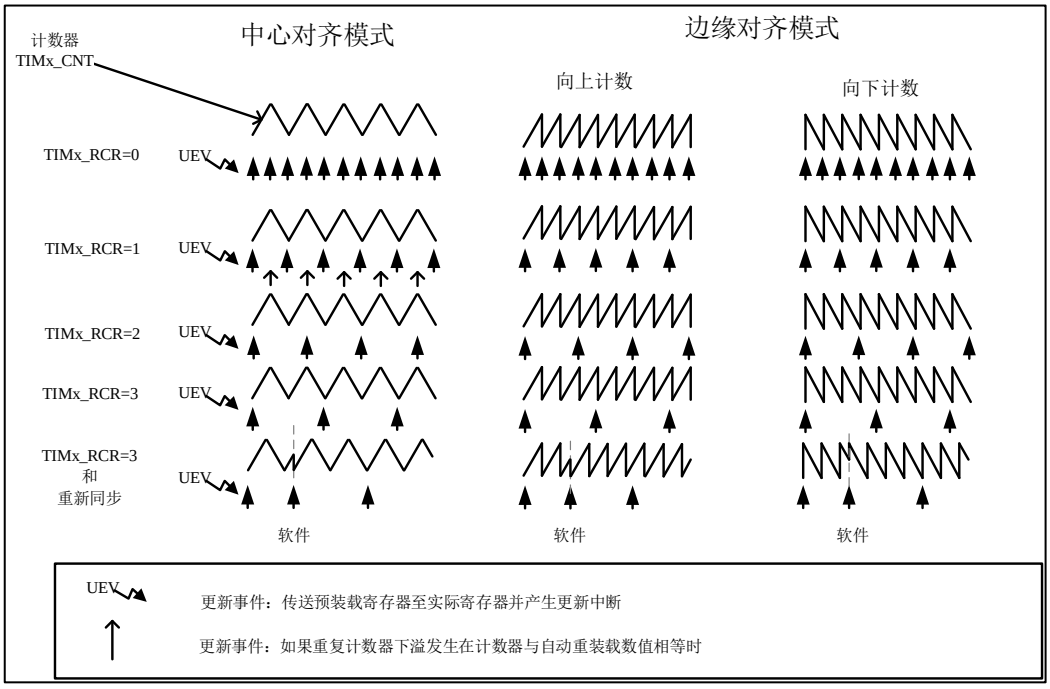
这意味着在每 N 次计数上溢或下溢时，数据从预装载寄存器传输到影子寄存器（TIMx_AR 自动重载入寄存器， TIMx_PSC 预装载寄存器，以及在比较模式下的捕获/比较寄存器 TIMx_CCDATx），N 是 TIMx_REPCNT 重复计数寄存器中的值。

重复计数器在下述任一条件成立时递减：

- 向上计数模式时，每次计数器溢出，
- 向下计数模式时，每次计数器下溢，
- 中央对齐模式时，每次上溢和每次下溢。虽然这样限制了 PWM 的最大循环周期为 128，但它能够在每个 PWM 周期 2 次更新占空比。在中央对齐模式下，因为波形是对称的，如果每个 PWM 周期中仅刷新一次比较寄存器，则最大的分辨率为 $2xT_{ck}$。

重复计数器是自动加载的，重复速率由 TIMx_REPCNT 寄存器的值定义（参看图 11-21）。当更新事件由软件产生（通过设置 TIMx_EVTGEN 中的 UDGN 位）或者通过硬件的从模式控制器产生，则无论重复计数器的值是多少，立即发生更新事件，并且 TIMx_REPCNT 寄存器中的内容被重载入到重复计数器。

图 11-21 不同模式下更新速率的例子，及 TIMx_REPCNT 的寄存器设置



11.3.4 时钟选择

计数器时钟由以下时钟源提供：

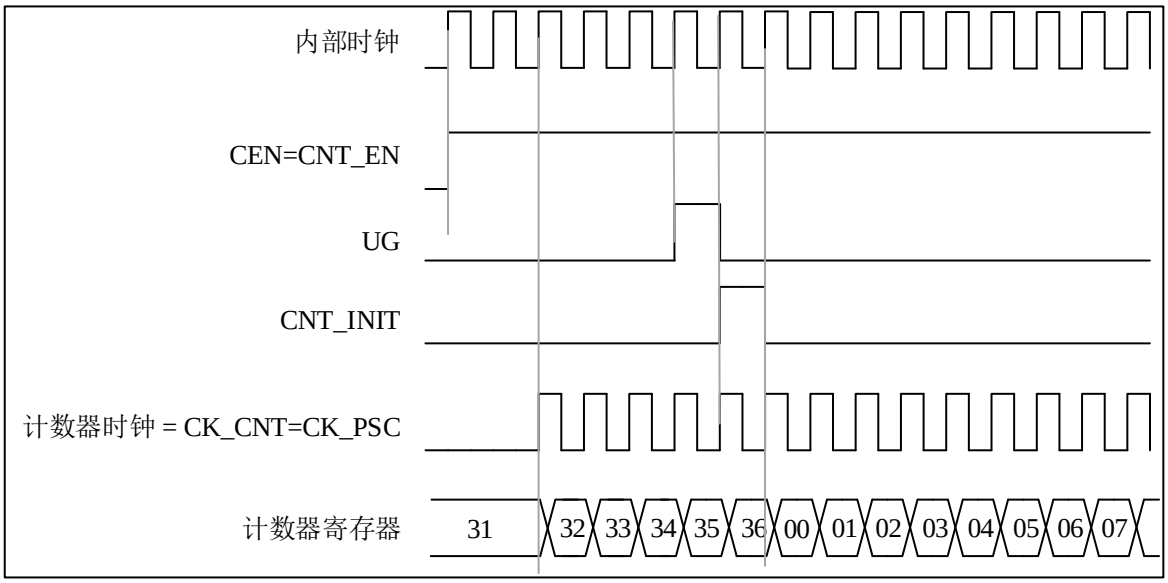
- 内部时钟（CK_INT）
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 内部触发输入（ITRx）：使用一个定时器作为另一个定时器的预分频器。如可以配置一个定时器 Timer1 而作为另一个定时器 Timer2 的预分频器。详见 12 章。

11.3.4.1 内部时钟源（CK_INT）

如果禁止从模式控制器（SMSEL=000），那么 CNTEN、DIR（TIMx_CTRL1 寄存器）和 UDGN 位（TIMx_EVTGEN 寄存器）是事实上的控制位，并且只能被软件修改（UDGN 位仍被自动清除）。只要 CNTEN 位被写成'1'，预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示了控制电路和向上计数器在一般模式下，不带预分频器时的操作。

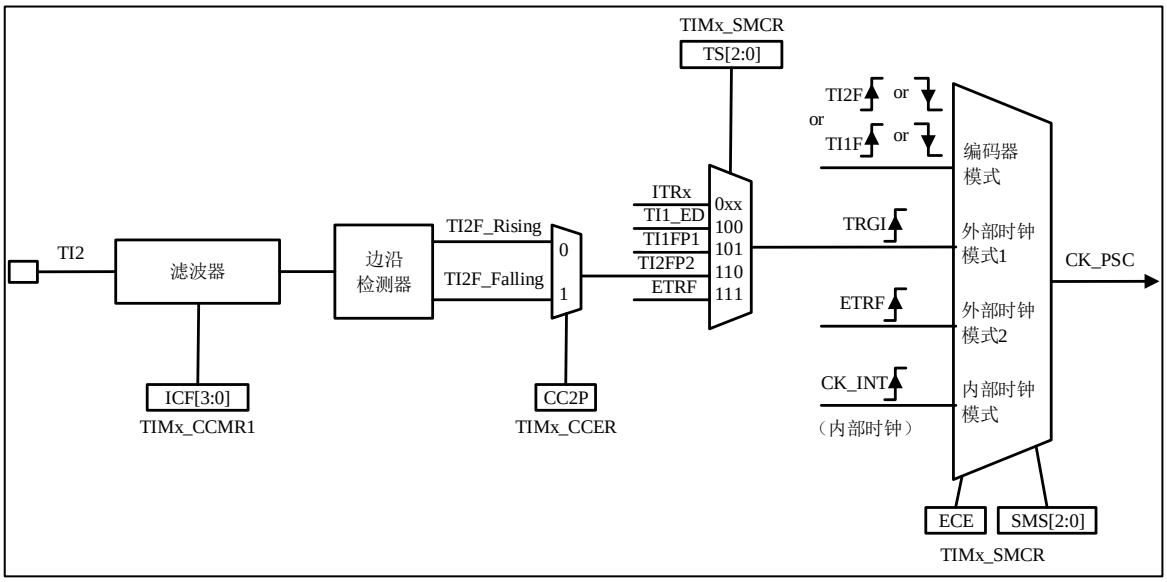
图 11-22 一般模式下的控制电路，内部时钟分频因子为 1



11.3.4.2 外部时钟源模式 1

如果 TIMx_SMCTRL 寄存器的 SMSEL=111，选择 TRGI 信号，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

图 11-23 TI2 外部时钟连接例子



例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤：

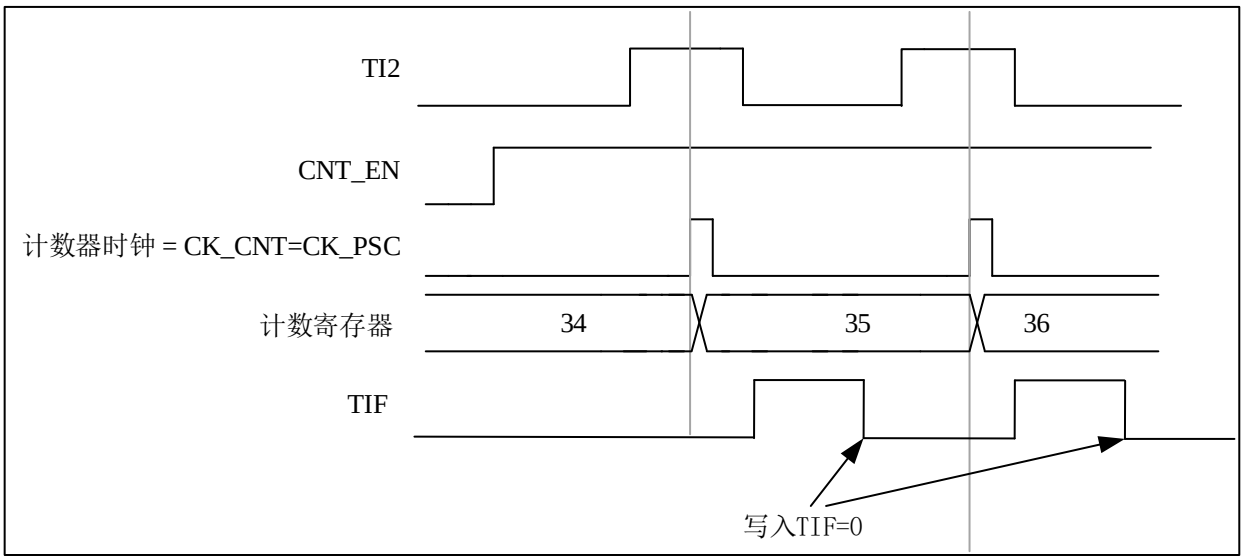
- 配置 TIMx_CCMOD1 寄存器 CC2SEL=01，配置通道 2 检测 TI2 输入的上升沿
- 配置 TIMx_CCMOD1 寄存器的 IC2F[3:0]，选择输入滤波器带宽（如果不需要滤波器，保持 IC2F=0000）
- 配置 TIMx_CCEN 寄存器的 CC2P=0，选定上升沿极性
- 配置 TIMx_SMCTRL 寄存器的 SMSEL=111，选择定时器外部时钟模式 1 配置 TIMx_SMCTRL 寄存器中的 TSEL=110，选定 TI2 作为触发输入源 设置 TIMx_CTRL1 寄存器的 CNTEN=1，启动计数器

注：捕获预分频器不用作触发，所以不需要对它进行配置

当上升沿出现在 TI2，计数器计数一次，且 TITF 标志被设置。

在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的重新同步电路。

图 11-24 外部时钟模式 1 下的控制电路

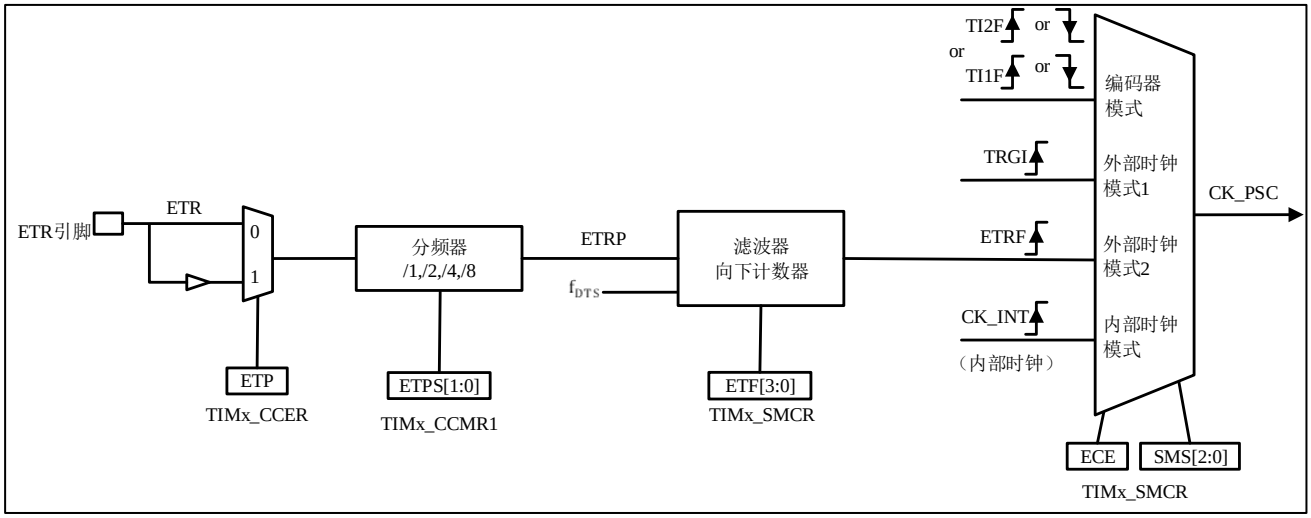


11.3.4.3 外部时钟源模式 2

选定此模式的方法是设置 TIMx_SMCTRL 寄存器中的 EXCEN=1（设置 TIMx_SMCTRL 寄存器中的 SMSEL=111，同时设置 TSEL=111 有相同效果）。

计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。下图是外部触发输入的框图。

图 11-25 外部触发输入框图

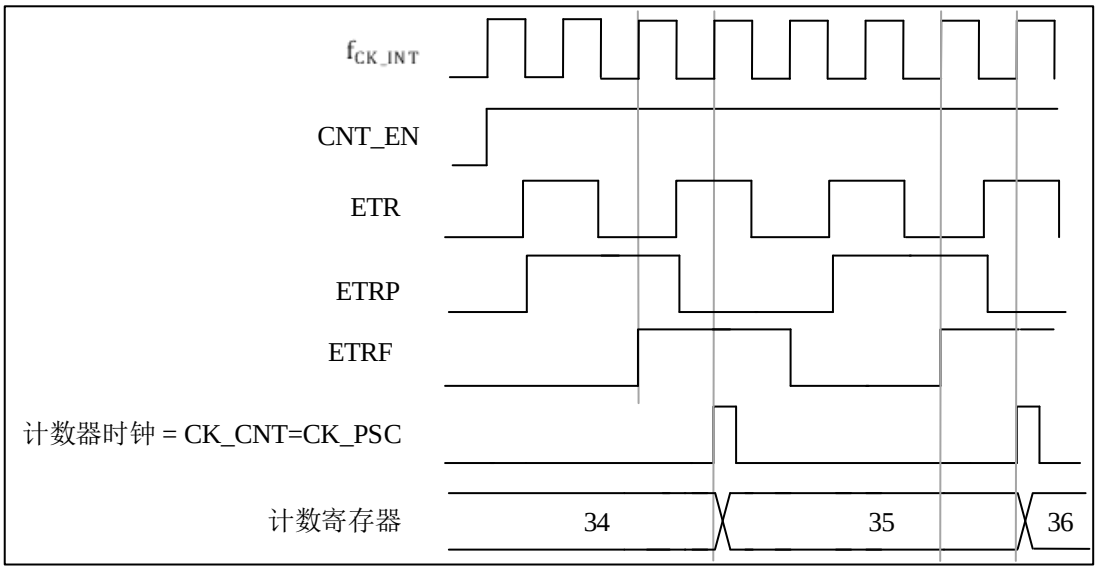


例如，要配置在 ETR 下每 2 个上升沿计数一次的向上计数器，使用下列步骤：

- 本例中不需要滤波器，置TIMx_SMCTRL寄存器中的EXTF[3:0]=0000
- 设置预分频器，置TIMx_SMCTRL寄存器中的EXTPS[1:0]=01
- 设置在ETR的上升沿检测，置TIMx_SMCTRL寄存器中的EXTP=0
- 开启外部时钟模式2，置TIMx_SMCTRL寄存器中的EXCEN=1
- 启动计数器，置TIMx_CTRL1寄存器中的CNTEN=1

计数器在每 2 个 ETR 上升沿计数一次。在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

图 11-26 外部时钟模式 2 下的控制电路



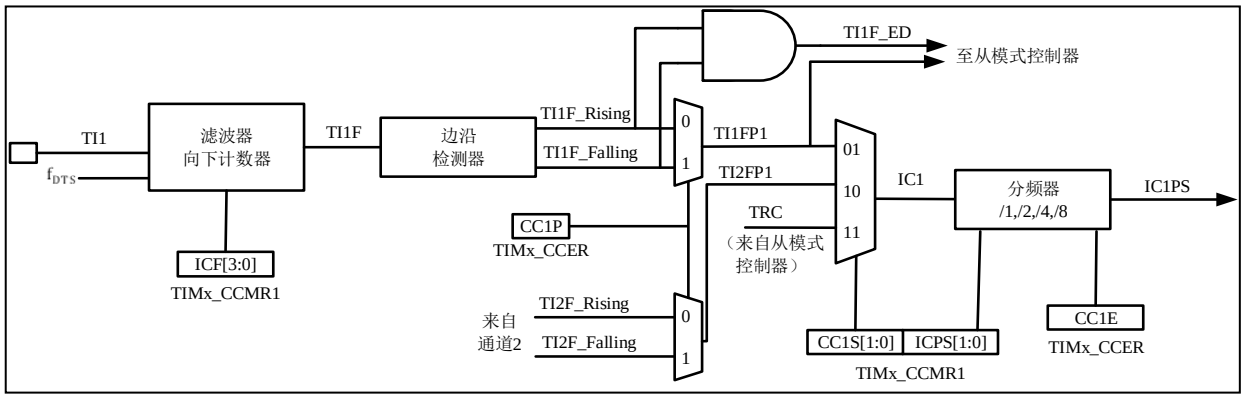
11.3.5 捕获/比较通道

捕获/比较通道都是独立的，都是围绕着一个捕获/比较寄存器（含影子寄存器），包括捕获的输入部分（数字滤波、多路复用和预分频器）和比较输出部分（比较器和输出控制）。

图 11-27 至图 11-30 是一个捕获/比较通道概览。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘监测器产生一个信号（TIxFPx），它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器（ICxPS）。

图 11-27 捕获/比较通道（如：通道 1 输入部分）



输出部分产生一个中间波形 OCxRef（高有效）作为基准，链的末端决定最终输出信号的极性。

图 11-28 捕获/比较通道 1 的主电路

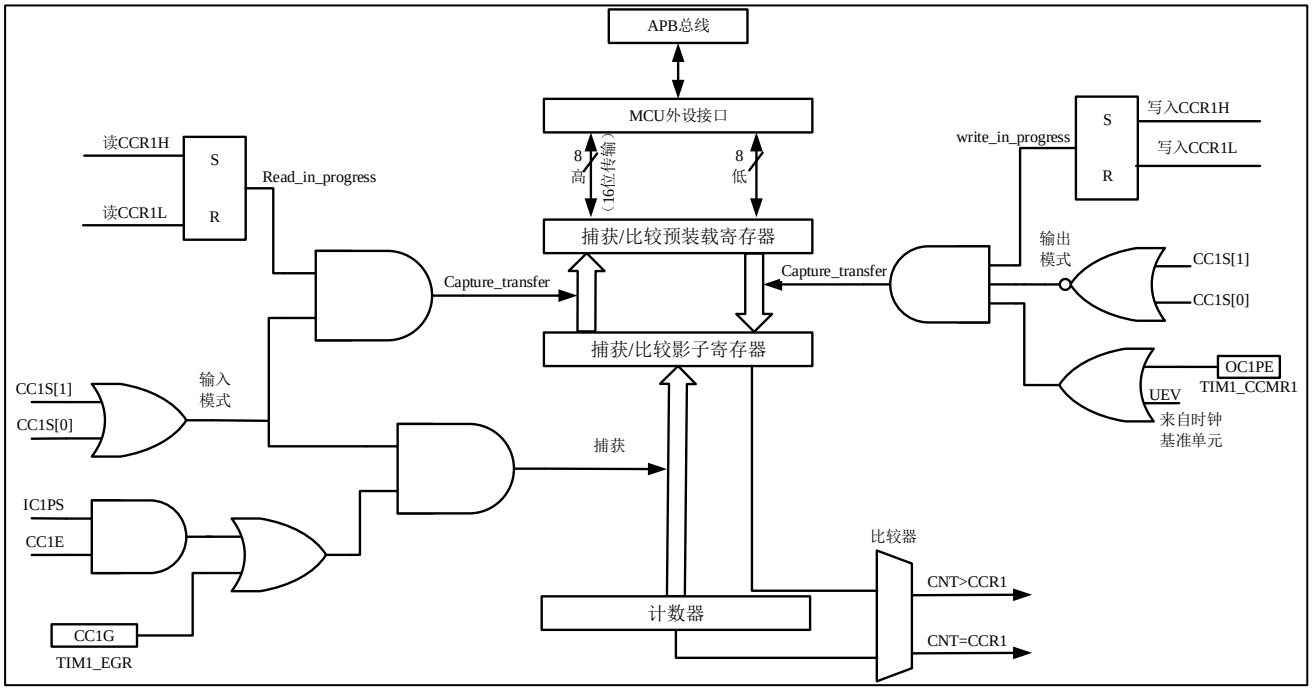


图 11-29 捕获/比较通道的输出部分（通道 1 至 3）

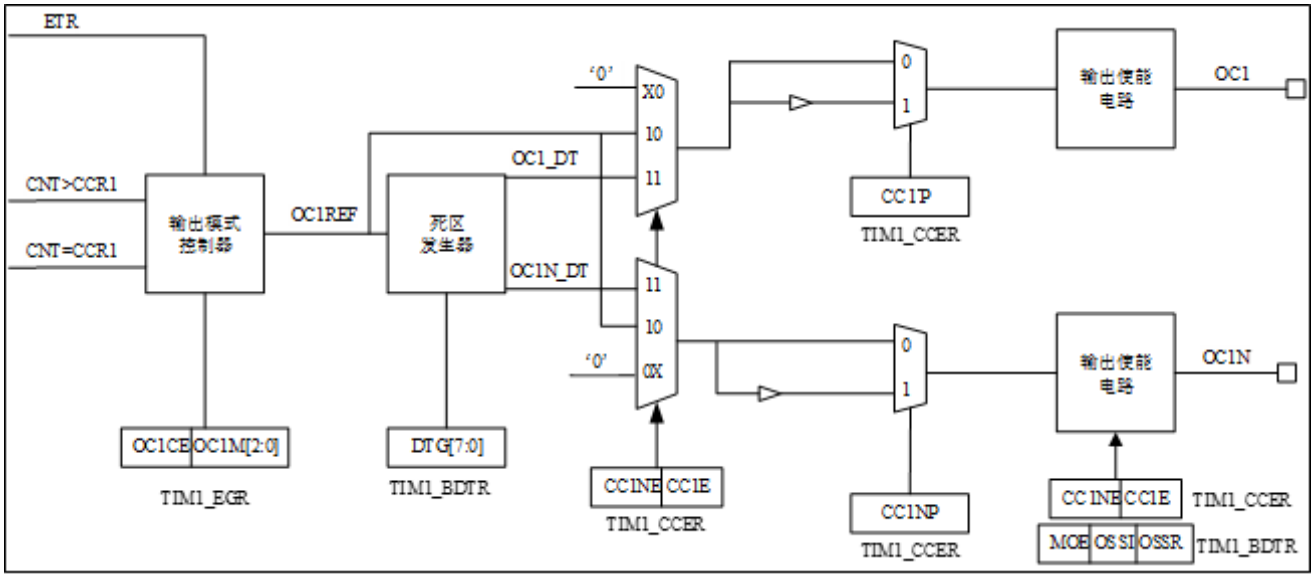
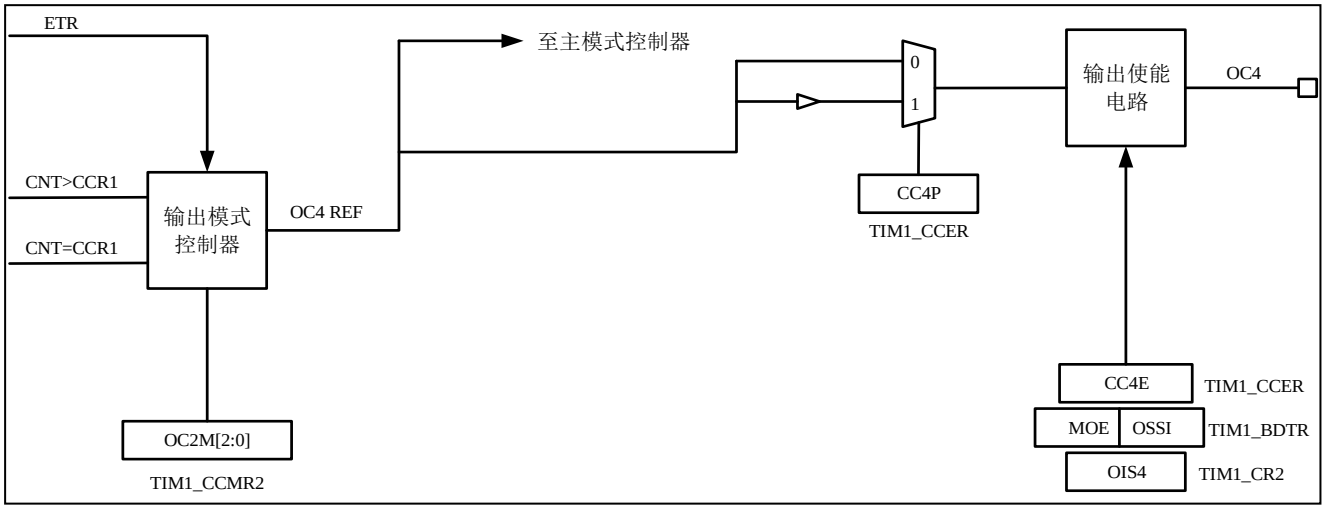


图 11-30 捕获/比较通道的输出部分（通道 4）



捕获/比较模块包括一个预装载寄存器和一个影子寄存器。读写过程中仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

11.3.6 输入捕获模式

配置成输入捕获模式时，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（TIMx_CCDATx）中。当发生捕获事件时，相应的 CCxIF 标志（TIMx_STS 寄存器）被置 1，如果开放了中断或者 DMA 操作，则将产生中断或者 DMA 请求。如果发生捕获事件时 CCxIF 标志已经为高，那么重复捕获标志 CCxOCF（TIMx_STS 寄存器）被置 1。写 CCxIF=0 可清除 CCxIF，或读取存储在 TIMx_CCDATx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOCF=0 可清除 CCxOCF。

以下示例说明如何将 TI1 输入的上升沿时捕获计数器的值捕获到 TIMx_CCDAT1 寄存器中，步骤如下：

- 选择有效输入端：TIMx_CCDAT1 必须连接到 TI1 输入，所以写入 TIMx_CCMOD1 寄存器中的 CC1SEL=01，只要 CC1SEL 不为'00'，通道被配置为输入，并且 TIMx_CCDAT1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽（即输入为 TIx 时，输入滤波器控制位是 TIMx_CCMODx 寄存器中的 ICxF 位）。假设输入信号在最多 5 个内部时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以（以 f_{DTS} 频率）连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIMx_CCMOD1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIMx_CCEN 寄存器中写入 CC1P=0（上升沿）。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止（写 TIMx_CCMOD1 寄存器的 IC1PS=00）。
- 设置 TIMx_CCEN 寄存器的 CC1EN=1，允许捕获计数器的值到捕获寄存器中。
- 必要的时候，通过设置 TIMx_DINTEN 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIMx_DINTEN 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMx_CCDAT1 寄存器。
- CC1ITF 标志被设置（中断标志）。当发生至少 2 个连续的捕获时，而 CC1ITF 未曾被清除，CC1OCF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

在处理捕获溢出时，建议在读出捕获溢出标志之前读取数据，这样可以避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注：设置 TIMx_EVTGEN 寄存器中相应的 CCxGN 位，可以通过软件产生输入捕获中断和/或 DMA 请求。

11.3.7 PWM 输入模式

此模式是输入捕获模式的一个特例，除以下区别外，操作与输入捕获模式相同：

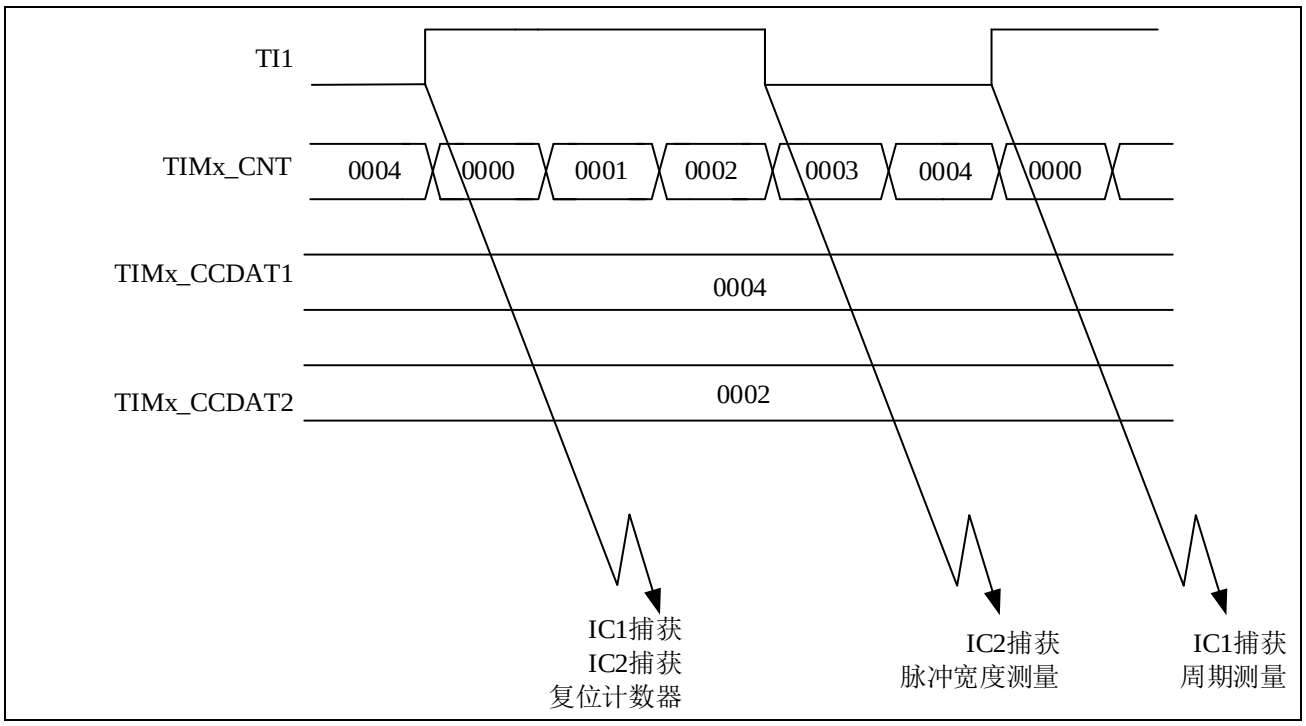
- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

例如，为了测量输入到 TI1 上的 PWM 信号的长度（TIMx_CCDAT1 寄存器）和占空比（TIMx_CCDAT2 寄存器），可以采用如下具体步骤（取决于 CK_INT 的频率和预分频器的值）

- 选择 TIMx_CCDAT1 的有效输入：置 TIMx_CCMOD1 寄存器的 CC1SEL=01（选中 TI1）。
- 选择 TI1FP1 的有效极性（用来捕获数据到 TIMx_CCDAT1 中和清除计数器）：置 CC1P=0（上升沿有效）。

- 选择 TIMx_CCDAT2 的有效输入：置 TIMx_CCMOD1 寄存器的 CC2SEL=10（选中 TI1）。
- 选择 TI1FP2 的有效极性（捕获数据到 TIMx_CCDAT2）：置 CC2P=1（下降沿有效）。
- 选择有效的触发输入信号：置 TIMx_SMCTRL 寄存器中的 TSEL=101（选择 TI1FP1）。
- 配置从模式控制器为复位模式：置 TIMx_SMCTRL 中的 SMSEL=100。
- 使能捕获：置 TIMx_CCEN 寄存器中 CC1EN=1 且 CC2EN=1。

图 11-31 PWM 输入模式时序



硬件上只有 TI1FP1 和 TI2FP2 连到了从模式控制器，所以 PWM 输入模式只能使用 TIMx_CH1/TIMx_CH2 信号。

11.3.8 强置输出模式

此模式具有使输出比较信号（OCxREF 和相应的 OCx/OCxN）直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

配置 TIMx_CCMODx 寄存器中相应的 OCxMD=101，即可强置输出比较信号（OCxREF/OCx）为有效状态。这样 OCxREF 被强置为高电平（OCxREF 始终为高电平有效），同时 OCx 得到 CCxP 极性相反的信号。

例如：CCxP=0（OCx 高电平有效），则 OCx 被强置为高电平。

配置 TIMx_CCMODx 寄存器中的 OCxMD=100，可强置 OCxREF 信号为低。

该模式下，在 TIMx_CCDATx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

11.3.9 输出比较模式

此项功能具有控制输出特定的波形，或者指示一段给定的的时间已经到时。

当计数器与捕获/比较寄存器的内容相同时，输出比较功能做如下操作:

- 将输出比较模式（TIMx_CCMODx 寄存器中的 OCxMD 位）和输出极性（TIMx_CCEN 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平(OCxMD=000)、被设置成有效电平(OCxMD=001)、被设置成无效电平(OCxMD=010)或进行翻转(OCxMD=011)。
- 设置中断状态寄存器中的标志位（TIMx_STS 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽（TIMx_DINTEN 寄存器中的 CCxIEN 位），则产生一个中断。
- 若设置了相应的使能位（TIMx_DINTEN 寄存器中的 CCxDEN 位，TIMx_CTRL2 寄存器中的 CCDS 位选择 DMA 请求功能），则产生一个 DMA 请求。

TIMx_CCMODx 中的 OCxPEN 位选择 TIMx_CCDATx 寄存器是否需要使用预装载寄存器。在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

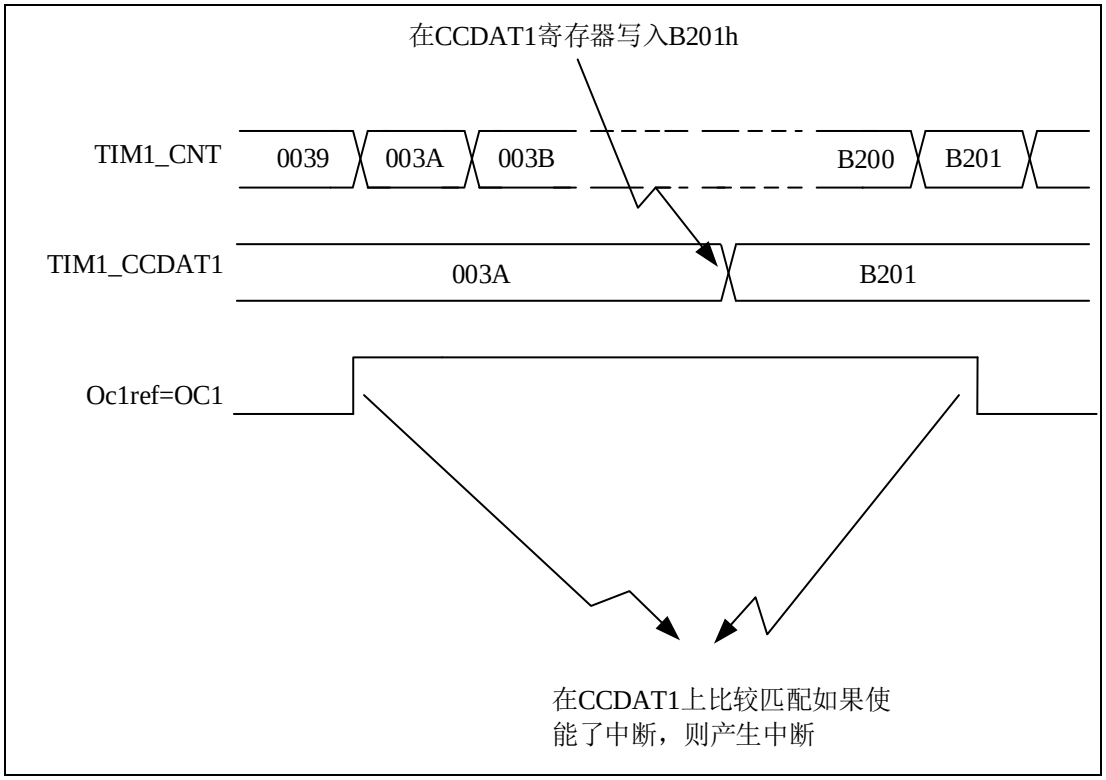
同步的精度可以达到计数器的一个计数周期。输出比较模式（在单脉冲模式下）也能用来输出一个单脉冲。

输出比较模式的配置步骤:

- 选择计数器时钟（内部，外部，预分频器)。
- 将相应的数据写入 TIMx_AR 和 TIMx_CCDATx 寄存器中。
- 如果要产生一个中断请求，设置 CCxIEN 位。
- 选择输出模式，例如:
  - 要求计数器与 CCDATx 匹配时翻转 OCx 的输出引脚，设置 OCxMD=011
  - 置 OCxPEN = 0 禁用预装载寄存器
  - 置 CCxP = 0 选择极性为高电平有效
  - 置 CCxEN = 1 使能输出
- 设置 TIMx_CTRL1 寄存器的 CNTEN 位启动计数器

TIMx_CCDATx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器（OCxPEN='0'，否则 TIMx_CCDATx 的影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

图 11-32 输出比较模式，翻转 OC1



11.3.10 PWM 模式

脉冲宽度调制（PWM）模式可以产生一个由 TIMx_AR 寄存器确定频率、由 TIMx_CCDATx 寄存器确定占空比的信号。

TIMx_CCMODx 寄存器中的 OCxMD 位写入'110'（PWM 模式 1）或'111'（PWM 模式 2），能够独立地设置每个 OCx 输出通道产生一路 PWM。必须通过设置 TIMx_CCMODx 寄存器的 OCxPEN 位使能相应的预装载寄存器，最后还要设置 TIMx_CTRL1 寄存器的 ARPEN 位（在向上计数或中心对称模式中）使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EVTGEN 寄存器中的 UDGN 位来初始化所有的寄存器。

OCx 的极性可以通过软件在 TIMx_CCEN 寄存器中的 CCxP 位进行设置，允许设置为高电平有效或低电平有效。OCx 的输出使能通过（TIMx_CCEN 和 TIMx_BKDT 寄存器中）CCxEN、CCxNEN、MOEN、OSSI 和 OSSR 位的组合控制。详见 11.4.10 TIMx_CCEN 寄存器的描述。

在 PWM 模式（模式 1 或模式 2）下，TIMx_CNT 和 TIMx_CCDATx 始终在进行比较，（依据计数器的计数方向）以确定是否符合 TIMx_CCDATx≤TIMx_CNT 或者 TIMx_CNT≤TIMx_CCDATx。

根据 TIMx_CTRL1 寄存器中 CAMSEL 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

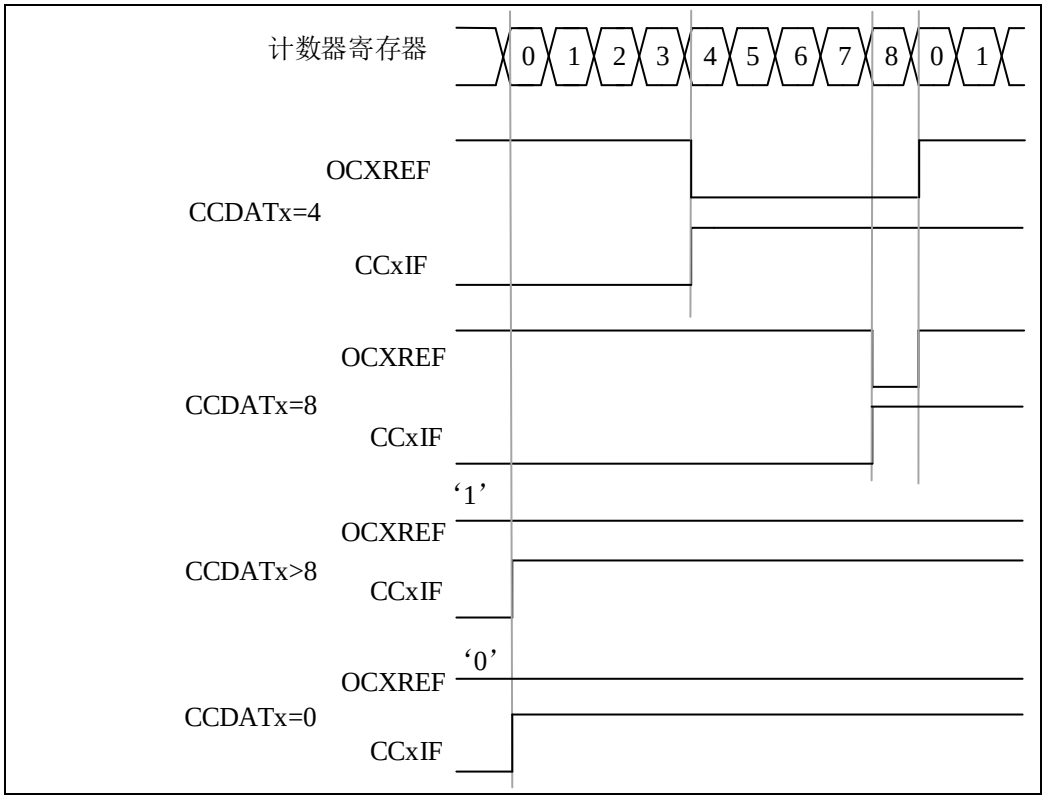
11.3.10.1 PWM 边沿对齐模式

- 向上计数配置

当 TIMx_CTRL1 寄存器中的 DIR 位为低的时候执行向上计数。参看 11.3.2 节。

在 PWM 模式 1 中，当 TIMx_CNT<TIMx_CCDATx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMx_CCDATx 中的比较值大于自动重装载值（TIMx_AR），则 OCxREF 保持为'1'。如果比较值为 0，则 OCxREF 保持为'0'。 下图给出了 TIMx_AR=8 时边沿对齐的 PWM 波形实例。

图 11-33 边沿对齐的 PWM 波形（AR=8）



- 向下计数的配置

当 TIMx_CTRL1 寄存器的 DIR 位为高时执行向下计数。参看 11.3.2 节。

在 PWM 模式 1 中，当 TIMx_CNT>TIMx_CCDATx 时参考信号 OCxREF 为低，否则为高。如果 TIMx_CCDATx 中的比较值大于 TIMx_AR 中的自动重装载值，则 OCxREF 保持为'1'。

注：该模式下不能产生 0%的 PWM 波形。

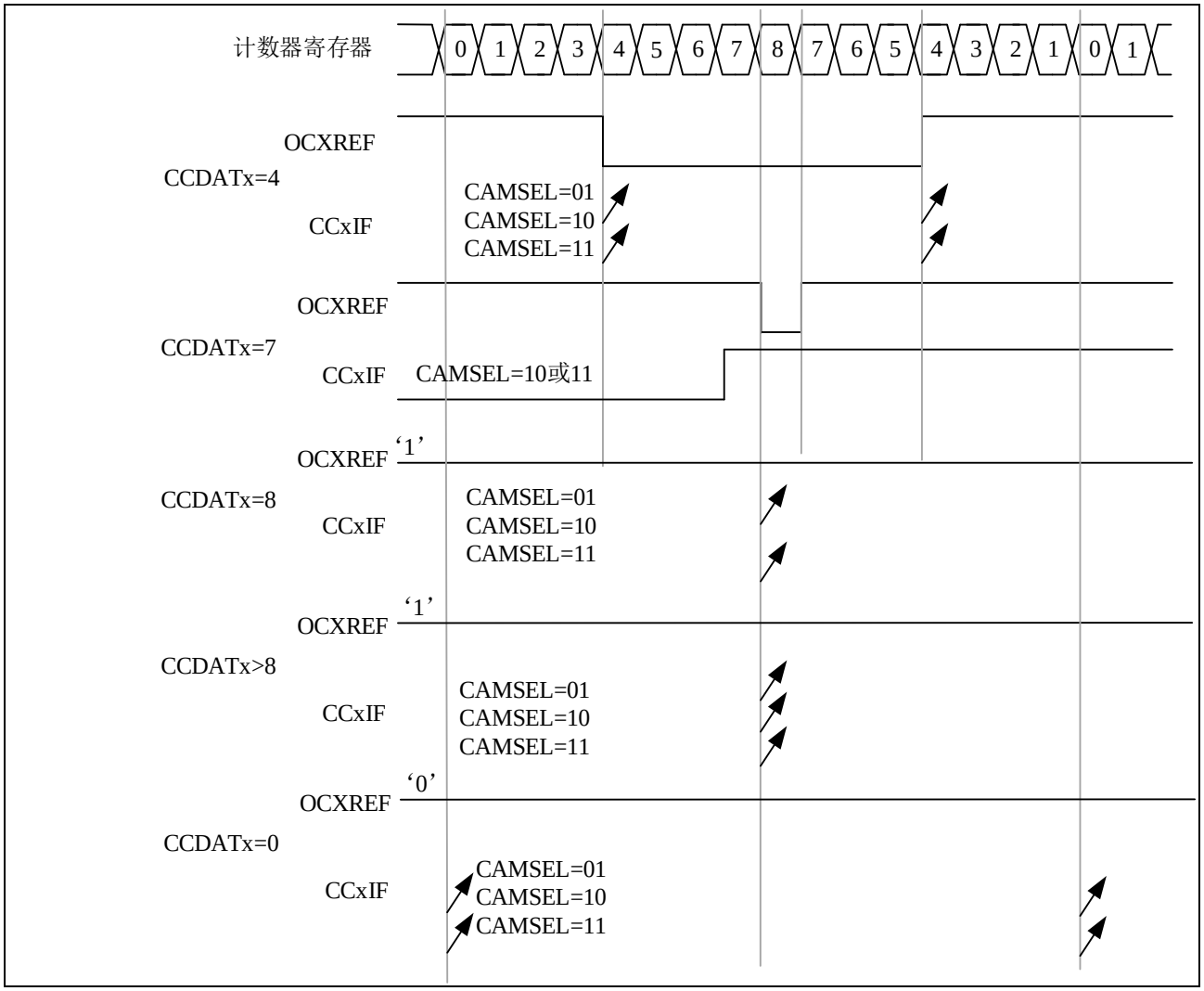
11.3.10.2 PWM 中央对齐模式

配置 TIMx_CTRL1 寄存器中的 CAMSEL 位不为'00'时为中央对齐模式（所有其他的配置对 OCxREF/OCx 信号都有相同的作用）。根据 CAMSEL 不同的位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIMx_CTRL1 寄存器中的计数方向位（DIR）由硬件更新，不要用软件修改。参看 11.3.2 节的中央对齐模式。

下图给出了一些中央对齐的 PWM 波形的示例。

- TIMx_AR=8
- PWM 模式 1
- TIMx_CTRL1 寄存器的 CAMSEL=01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。

图 11-34 中央对齐的 PWM 波形（APR=8）



11.3.10.3 使用中央对齐模式的提示：

- 在进入中央对齐模式的时候，使用当前的向上/向下计数配置值；即计数器向上还是向下计数取决于 TIMx_CTRL1 寄存器中 DIR 位的当前值。此外，软件不能同时对 DIR 和 CAMSEL 位进行修改。
- 不推荐当运行在中央对齐模式时改写计数器，因为这样可能会产生预期之外的 PWM 波形和软件行为。特别地：
  - 如果写入计数器的值大于自动重加载的值（TIMx_CNT>TIMx_AR），则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。
  - 如果将 0 或者 TIMx_AR 的值写入计数器，方向被更新，但不产生更新事件 UEV。
- 使用中央对齐模式最可靠的方法，就是在启动计数器之前产生一个软件更新（设置 TIMx_EVTGEN 位中的 UDGN 位）事件，并且不要在计数进行过程中修改计数器的值。

11.3.11 互补输出和死区插入

高级控制定时器（TIM1 和 TIM8）能够输出两路互补信号，并且能够管理输出的瞬时关断和接通时间。这段时间通常被称为死区时间，用户应该根据连接的输出器件和它们的特性（电平转换的延时、电源开关的延时等）来对死区时间进行调整。

通过配置 TIMx_CCEN 寄存器中的 CCxP 和 CCxNP 位，可以为每主输出 OCx 和互补输出 OCxN 独立选择极性。

主输出 OCx 和互补输出 OCxN 通过下列控制位的组合进行控制：TIMx_CCEN 寄存器的 CCxEN 和 CCxNEN 位，TIMx_BKDT 和 TIMx_CTRL2 寄存器中的 MOEN、OIx、OIxN、OSSI 和 OSSR 位，详见表 11-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。特别是在转换到 IDLE 状态时（MOEN 下降到 0）死区被激活。

通过同时设置 CCxEN 和 CCxNEN 位将插入死区，如果存在刹车电路，则还要设置 MOEN 位。每一个通道都有一个 10 位的死区发生器。参考信号 OCxREF 可以产生 2 路输出 OCx 和 OCxN。

如果 OCx 和 OCxN 为高有效：

- OCx 输出信号与参考信号相同，只是它的上升沿相对于参考信号的上升沿有一个延迟。
- OCxN 输出信号与参考信号相反，只是它的上升沿相对于参考信号的下降沿有一个延迟。如果延迟大于当前有效的输出宽度（OCx 或者 OCxN），则不会产生相应的脉冲。

下列时序演示了死区发生器的输出信号和当前参考信号 OCxREF 之间的关系。（假设 CCxP=0、CCxNP=0、MOEN=1、CCxEN=1 并且 CCxNEN=1）

图 11-35 带死区插入的互补输出

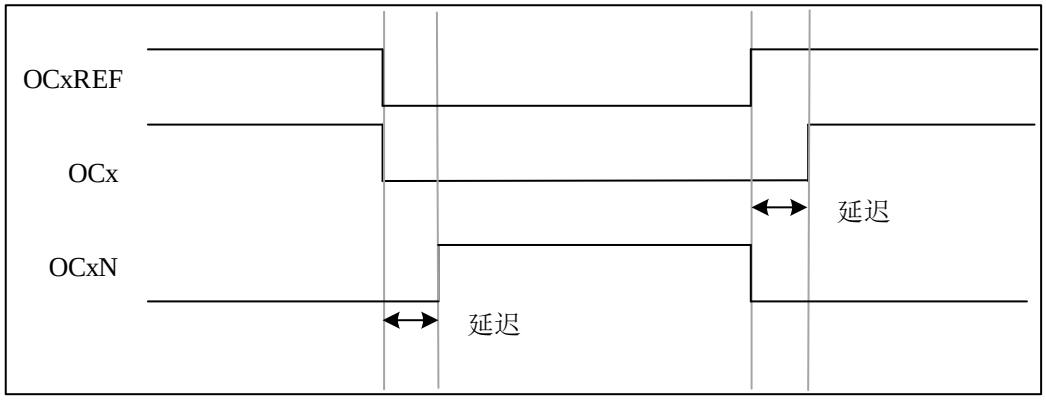


图 11-36 死区波形延迟大于负脉冲

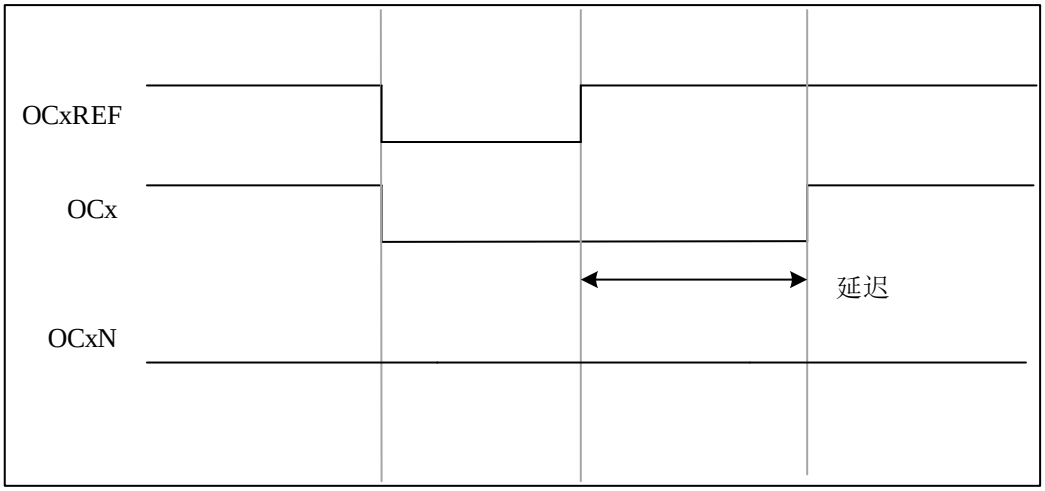
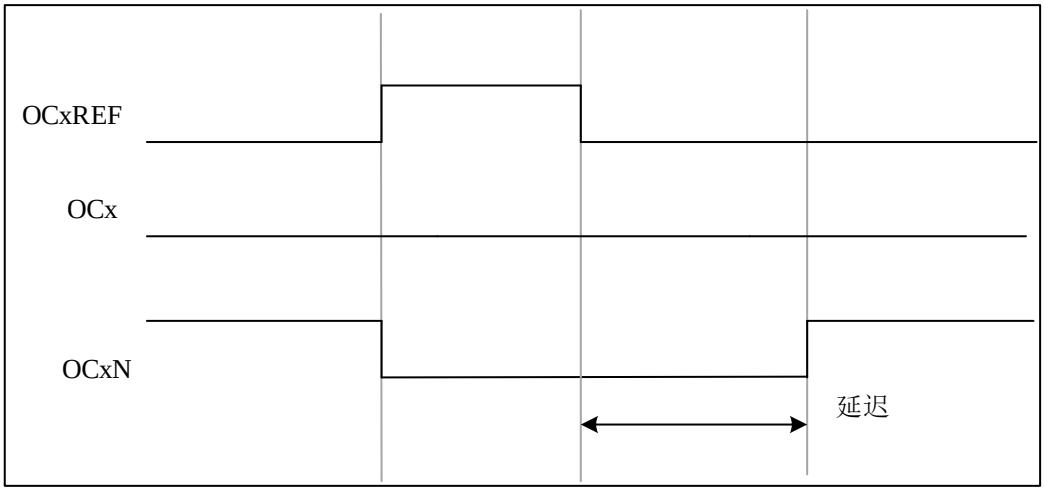


图 11-37 死区波形延迟大于正脉冲



每一个通道的死区延时统一由 TIMx_BKDT 寄存器中的 DTGN 位编程配置。详见 11.4.19TIM1 和 TIM8 刹车和死区寄存器（TIMx_BKDT）中的延时计算。

11.3.11.1 重定向 OCxREF 到 OCx 或 OCxN

在（强置、输出比较或 PWM）输出模式下，通过配置 TIMx_CCEN 寄存器的 CCxEN 和 CCxNEN 位，OCxREF 可以被重定向到 OCx 或者 OCxN 的输出。

此功能有二个作用:

- 可以在互补输出处于无效电平时，在某个输出上送出一个指定的波形（例如 PWM 或者静态有效电平）。
- 让两个输出同时处于无效电平，或处于有效电平和带死区的互补输出。

注：当只使能 OCxN（CCxEN=0, CCxNEN=1）时，它不会反相，当 OCxREF 有效时立即变高。例如，如果 CCxNP=0，则 OCxN=OCxREF。另一方面，当 OCx 和 OCxN 都被使能时（CCxEN=CCxNEN=1），当 OCxREF 为高时 OCx 有效；而 OCxN 相反，当 OCxREF 低时 OCxN 变为有效。

11.3.12 使用刹车功能

在使用刹车功能时，依据相应的控制位（TIMx_BKDT 寄存器中的 MOEN、OSSI 和 OSSR 位，TIMx_CTRL2 寄存器中的 OIx 和 OIxN 位），输出使能信号和无效电平都会被修改。但任意时候，OCx 和 OCxN 输出不能在同一时间同时处于有效电平上。详见表 11-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位。

有如下信号可以作为刹车源，当多个刹车信号都被使能后，各刹车信号构成一个或逻辑。

- 刹车输入管脚
- 一个时钟失效事件。时钟失败事件由复位时钟控制器中的时钟安全系统产生，详见 5.2.7 节时钟安全系统（CSS）。
- 一个电源电压检测（PVD）失效事件
- 比较器的输出信号（在比较器模块中配置，高电平刹车）

系统复位后，刹车电路被禁止，MOEN 位为低。设置 TIMx_BKDT 寄存器中的 BKEN 位可以使能刹车功能，刹车输入信号的极性可以通过配置同一个寄存器中的 BKP 位选择。BKEN 和 BKP 可以同时被修改。当写入 BKEN 和 BKP 位时，在真正写入之前会有 1 个 APB 时钟周期的延迟，因此需要等待一个 APB 时钟周期之后，才能正确地读回写入的位。

因为 MOEN 下降沿可以是异步的，在实际信号（作用在输出端）和同步控制位（在 TIMx_BKDT 寄存器中）之间设置了一个再同步电路。这个再同步电路会在异步信号和同步信号之间产生延迟。特别的，如果当它为低时写 MOEN=1，则读出它之前必须先插入一个延时（空指令）才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

在发生刹车时（在刹车输入端出现选定的电平），有下述动作：

- MOEN 位被异步地清除，将输出置于无效状态、空闲状态或者复位状态（由 OSSI 位选择）。这个特性在 MCU 的振荡器关闭时依然有效。
- 一旦 MOEN=0，每一个输出通道输出由 TIMx_CTRL2 寄存器中的 OIx 位设定的电平。如果 OSSI=0，则定时器释放使能输出，否则使能输出始终为高。
- 当使用互补输出时：
  - 输出首先被置于复位状态即无效的状态（取决于极性）。这是异步操作，即使定时器没有时钟时，此功能也有效。
  - 如果定时器的时钟依然存在，死区生成器将会重新生效，在死区之后根据 OIx 和 OIxN 位指示的电平驱动输出端口。即使在这种情况下，OCx 和 OCxN 也不能被同时驱动到有效的电平。注，因为重新同步 MOEN，死区时间比通常情况下长一些（大约 2 个 ck_tim 的时钟周期）。
  - 如果 OSSI=0，定时器释放使能输出，否则保持使能输出；或一旦 CCxEN 与 CCxNEN 之一变高时，使能输出变为高。
- 如果设置了 TIMx_DINTEN 寄存器中的 BIE 位，当刹车状态标志（TIMx_STS 寄存器中的 BITF 位）为'1'时，则产生一个中断。如果设置了 TIMx_DINTEN 寄存器中的 BDE 位，则产生一个 DMA 请求。
- 如果设置了 TIMx_BKDT 寄存器中的 AOEN 位，在下一个更新事件 UEV 时 MOEN 位被自动置位；例如，这可以用来进行整形。否则，MOEN 始终保持低直到被再次置'1'；此时，这个特性可以被用

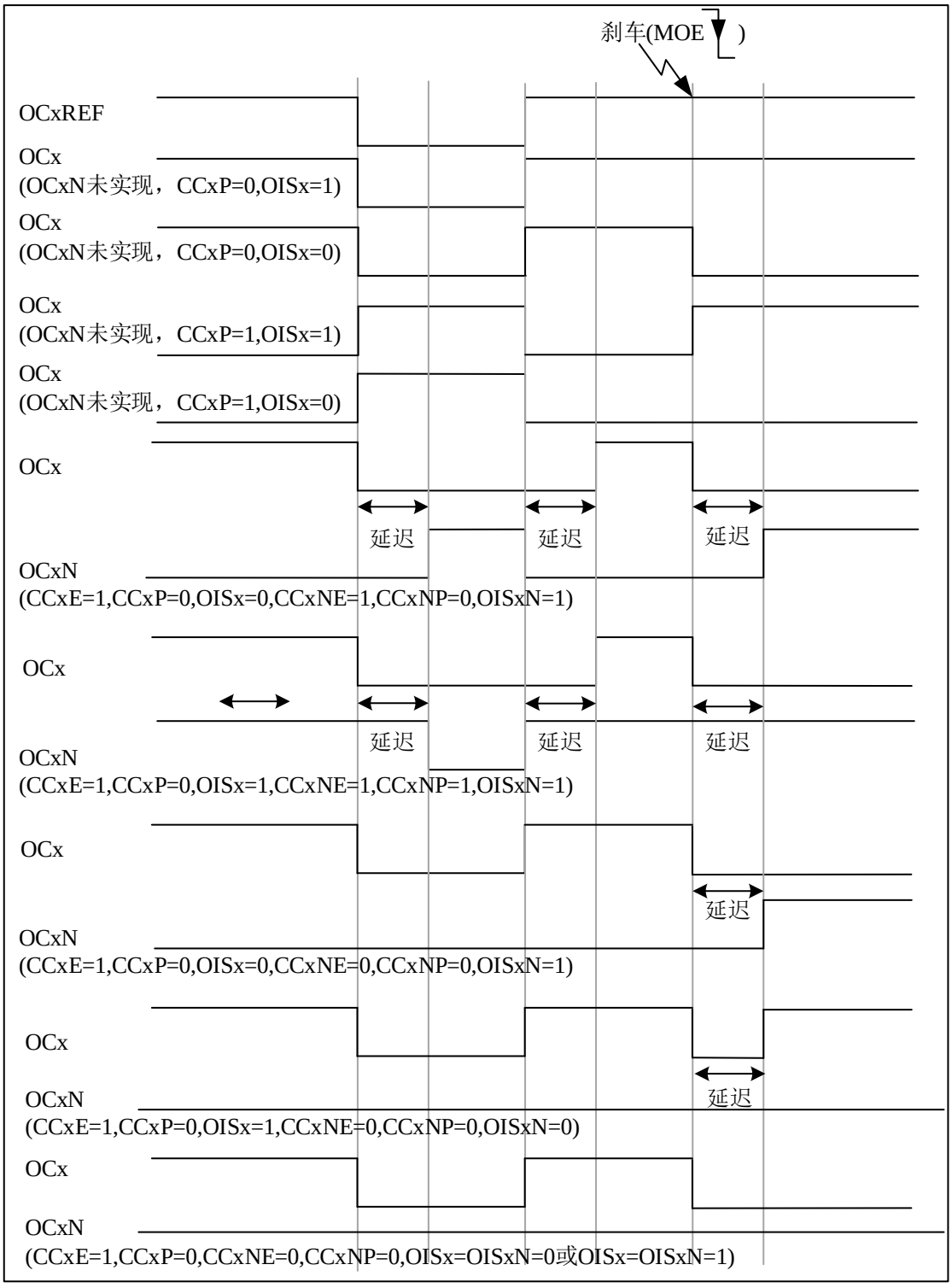
在安全方面，可以把刹车输入连到电源驱动的报警输出、热敏传感器或者其他安全器件上。

注：刹车输入为电平有效。所以，当刹车输入有效时，不能同时（自动地或者通过软件）设置 MOEN。 同时，状态标志 BITF 不能被清除。

刹车由 BRK 输入产生，它的有效极性是可编程的，且由 TIMx_BKDT 寄存器中的 BKEN 位开启。 除了刹车输入和输出管理，刹车电路中还实现了写保护以保证应用程序的安全。支持用户冻结几个配置参数（死区长度，OCx/OCxN 极性和被禁止的状态，OCxMD 配置，刹车使能和极性）。

用户可以通过 TIMx_BKDT 寄存器中的 LCKCFG 位，在三级保护中选择其中一种，参看 11.4.19TIM1 和 TIM8 刹车和死区寄存器（TIMx_BKDT）。在 MCU 复位后 LCKCFG 位只能被修改一次。下图演示实例了响应刹车的输出。

图 11-38 响应刹车的输出在外部事件时清除 OCxREF 信号



11.3.13 在外部事件时清除 OCxREF 信号

针对给定的通道，设置 TIMx_CCMODx 寄存器中对应的 OCxCEN 位为'1'，能够用 ETRF 输入端的高电平把 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

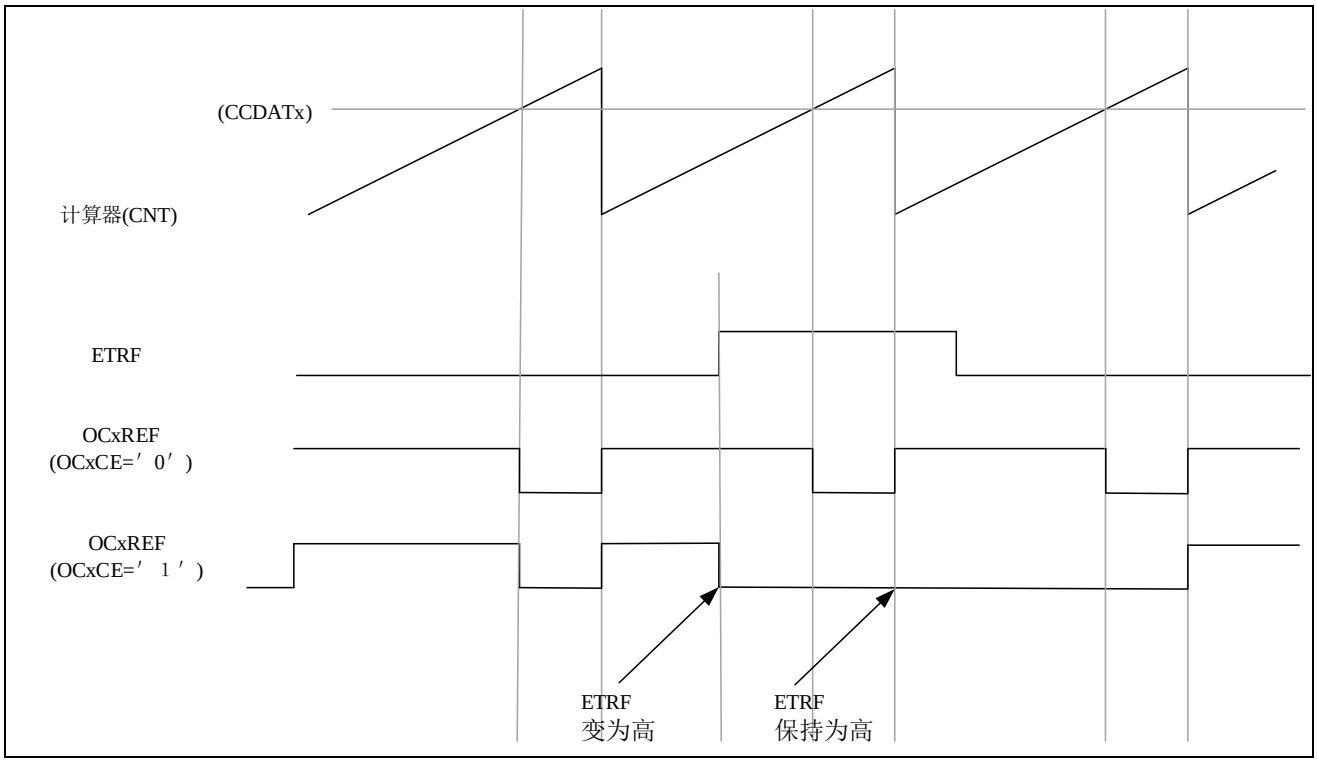
该功能只能用于输出比较和 PWM 模式，不能用于强置模式。

例如，OCxREF 信号可以联到一个比较器的输出，用于控制电流。这时，ETR 必须配置如下：

- 外部触发预分频器必须处于关闭：TIMx_SMCTRL 寄存器中的 EXTPS[1:0]=00。
- 必须禁止外部时钟模式 2：TIMx_SMCTRL 寄存器中的 EXCEN=0。
- 外部触发极性（EXTP）和外部触发滤波器（EXTF）可以根据需要配置。

下图定时器 TIMx 被置于 PWM 模式，显示了当 ETRF 输入变为高时，对应不同 OCxCEN 的值，OCxREF 信号的动作。在这个例子中。

图 11-39 清除 TIMx 的 OCxREF



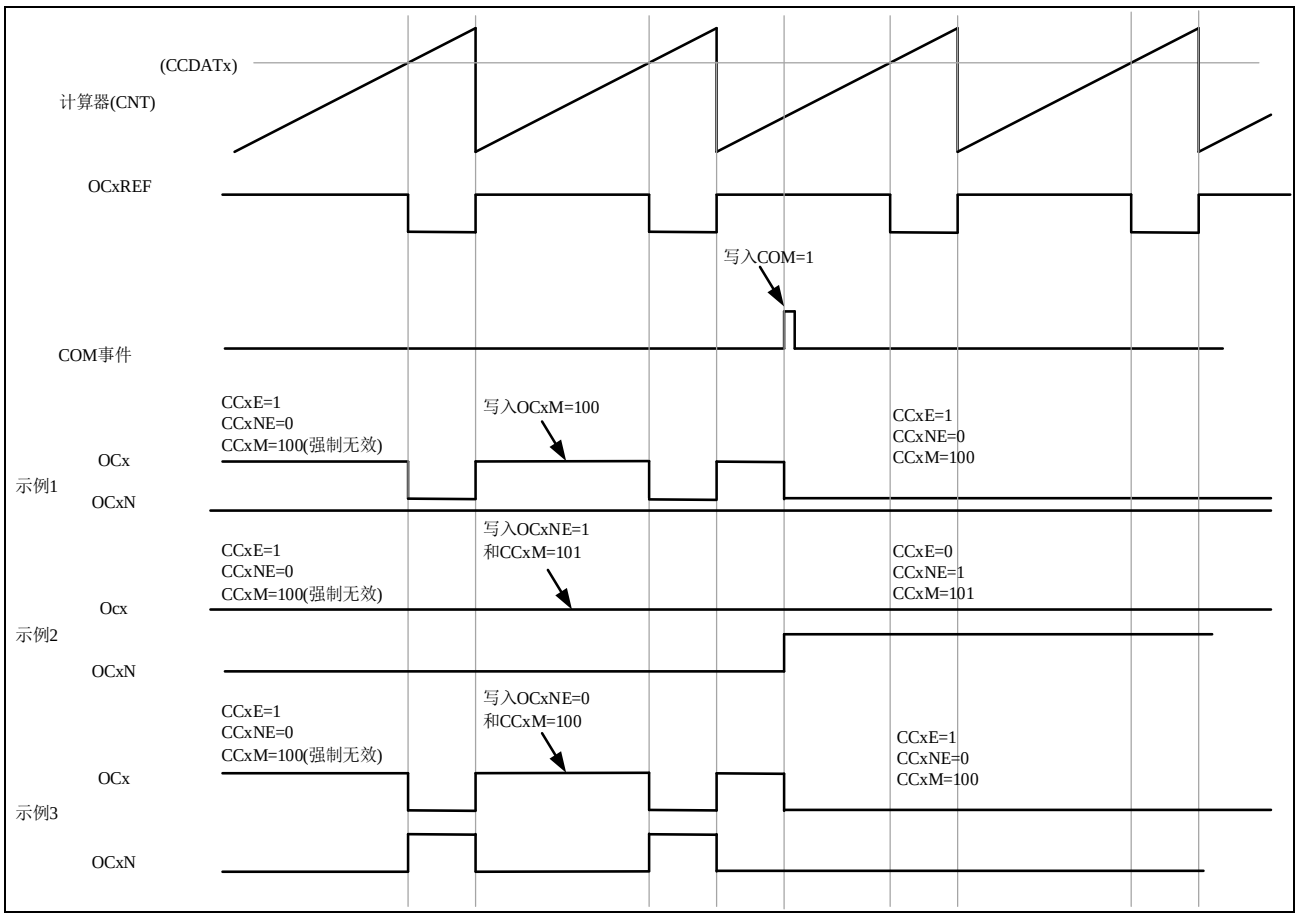
11.3.14 产生六步 PWM 输出

需要在一个通道上产生互补输出时，预装载位有 OCxMD、CCxEN 和 CCxNEN。在发生 COM 换相事件时，这些预装载位被传送到影子寄存器位。这样就可以预先设置好下一步骤配置，并在同一个时刻同时修更改所有通道的配置。COM 可以通过设置 TIMx_EVTGEN 寄存器的 CCUDGN 位由软件产生，或在 TRGI 上升沿由硬件产生。当发生 COM 事件时会设置一个标志位(TIMx_STS 寄存器中的 COMITF 位)，这时如果已设置了 TIMx_DINTEN 寄存器的 COMIE 位，则产生一个中断；如果已设置了 TIMx_DINTEN 寄存器的 COMDEN 位，则产生一个 DMA 请求。

注：COM 事件指比较更新事件。

下图显示当发生 COM 事件时，三种不同配置下 OCx 和 OCxN 输出。

图 11-40 产生六步 PWM，使用 COM 的例子（OSSR=1）



11.3.15 单脉冲模式

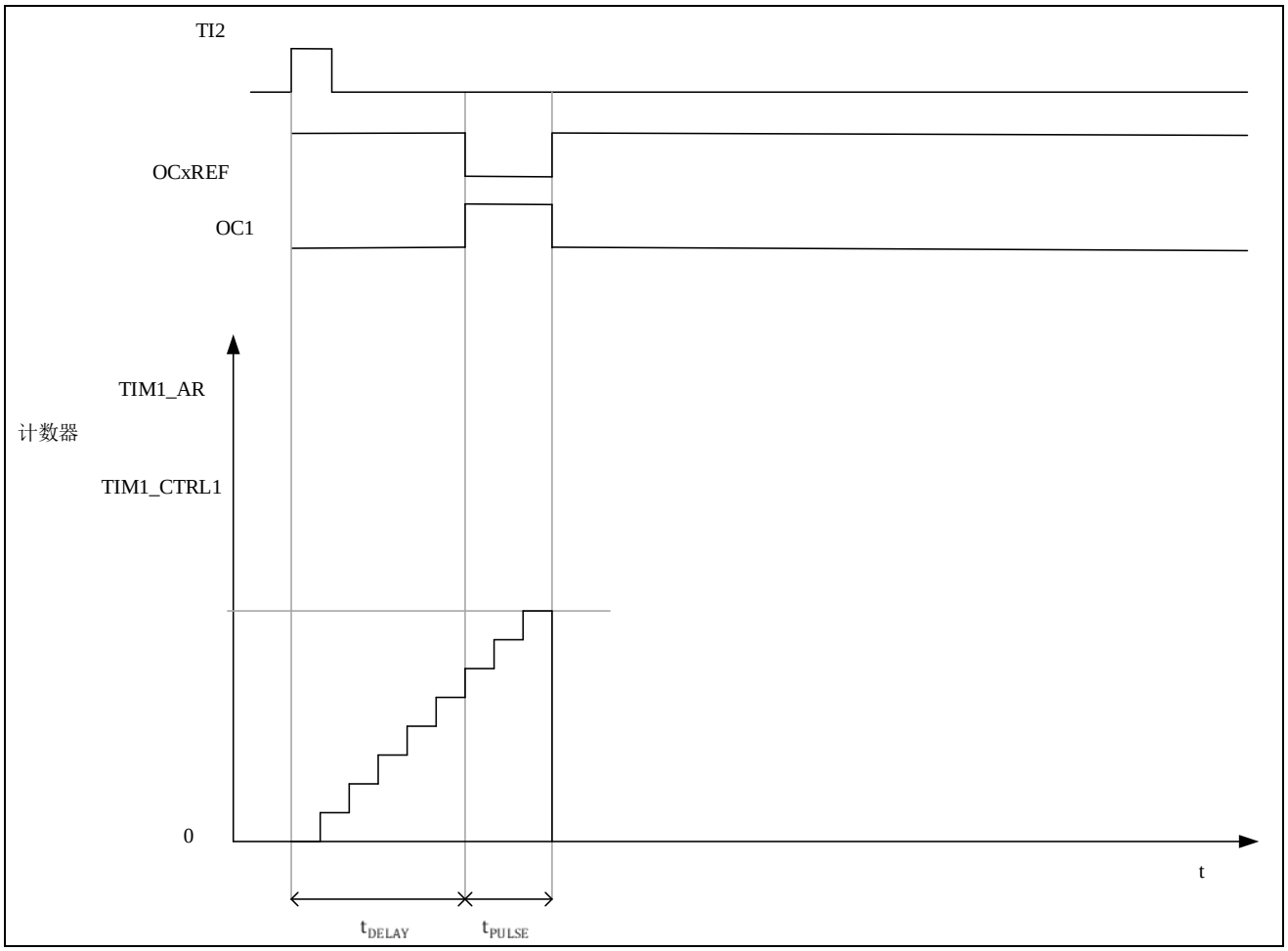
单脉冲模式（ONEPM）允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲，是前述众多模式的一个特例。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CTRL1 寄存器中的 ONEPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 后停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前（当定时器正在等待触发），必须进行如下配置：

- 向上计数方式：计数器 CNT < CCDATx ≤ AR （特别地，0 < CCDATx），
- 向下计数方式：计数器 CNT > CCDATx。

图 11-41 单脉冲模式的例子



例如，需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 t_{DELAY} 之后，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

假定 TI2FP2 作为触发 1:

- 置 TIMx_CCMOD1 寄存器中的 CC2SEL=01，把 TI2FP2 映像到 TI2。
- 置 TIMx_CCEN 寄存器中的 CC2P=0，使 TI2FP2 能够检测上升沿。
- 置 TIMx_SMCTRL 寄存器中的 TSEL=110，TI2FP2 作为从模式控制器的触发（TRGI）。
- 置 TIMx_SMCTRL 寄存器中的 SMSEL=110（触发模式），TI2FP2 被用来启动计数器。

  ONEPM 的波形由写入比较寄存器的数值决定（要考虑时钟频率和计数器预分频器）
- t_{DELAY} 由 TIMx_CCDAT1 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义（TIMx_AR - TIMx_CCDAT1）。
- 假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器达到预装载值时要产生一个从 1 到 0 的波形；首先置 TIMx_CCMOD1 寄存器的 OC1M=111，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIMx_CCMOD1 中的 OC1PEN=1 和 TIMx_CTRL1 寄存器中的 ARPEN；然后在

TIMx_CCDAT1 寄存器中填写比较值，在 TIMx_AR 寄存器中填写自动装载值，设置 UDGN 位来产生一个更新事件，然后等待在 TI2 上产生一个外部触发事件。在本示例中，CC1P=0。

在本示例中，TIMx_CTRL1 寄存器中的 DIR 和 CAMSEL 位置低。

因为只需要一个脉冲，所以必须设置 TIMx_CTRL1 寄存器中的 ONEPM=1，在下一个更新事件（当计数器从自动装载值翻转到 0）时停止计数。

11.3.15.1 特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIx 输入管脚的边沿检测逻辑设置 CNTEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY}。

如果要以最小延时输出波形，可以设置 TIMx_CCMODx 寄存器中的 OCxFEN 位；此时 OCxREF（和 OCx）直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFEN 只在通道配置为 PWM1 和 PWM2 模式时起作用。

11.3.16 编码器接口模式

选择编码器接口模式的方法有：

- 如果计数器只在 TI2 的边沿计数，则置 TIMx_SMCTRL 寄存器中的 SMSEL=001。
- 如果只在 TI1 边沿计数，则置 SMSEL=010。
- 如果计数器同时在 TI1 和 TI2 边沿计数，则置 SMSEL=011。

通过设置 TIMx_CCEN 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性；如果需要，还可以对输入滤波器编程。

两个输入 TI1 和 TI2 用来作为增量编码器的接口。参看表 11-1，假定计数器已经启动（TIMx_CTRL1 寄存器中的 CNTEN=1），则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变进行驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIMx_CTRL1 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数，在任一输入端（TI1 或者 TI2）的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIMx_AR 寄存器的自动装载值之间连续计数（根据方向，或是 0 到 AR 计数，或是 AR 到 0 计数）。所以在开始计数之前必须配置 TIMx_AR；同样，捕获器、比较器、预分频器、重复计数器、触发输出特性等仍工作如常。

注：编码器模式和外部时钟模式 2 不兼容，因此不能同时操作。

在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合， 假设 TI1 和 TI2 不同时变换。

表 11-1 计数方向与编码器信号的关系

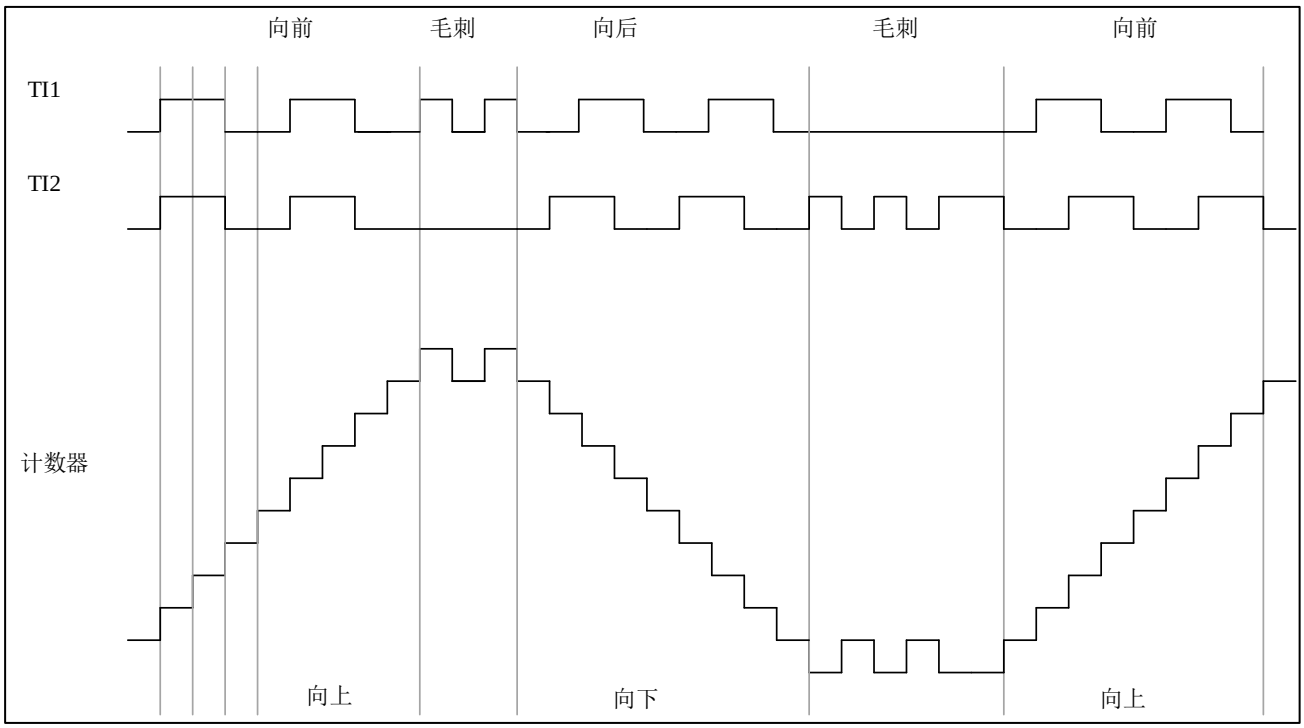
有效边沿	相对信号的电平（TI1FP1对应TI2，TI2FP2对应TI1）	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
仅在TI1计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在TI2计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

一个外部的增量式编码器可以直接与 MCU 管脚连接而不需要增加外部接口逻辑。但是，一般会使用比较器将编码器的差动输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械过零点，可以把它连接到一个外部中断输入并触发一个计数器复位。

下图是一个计数器操作的示例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

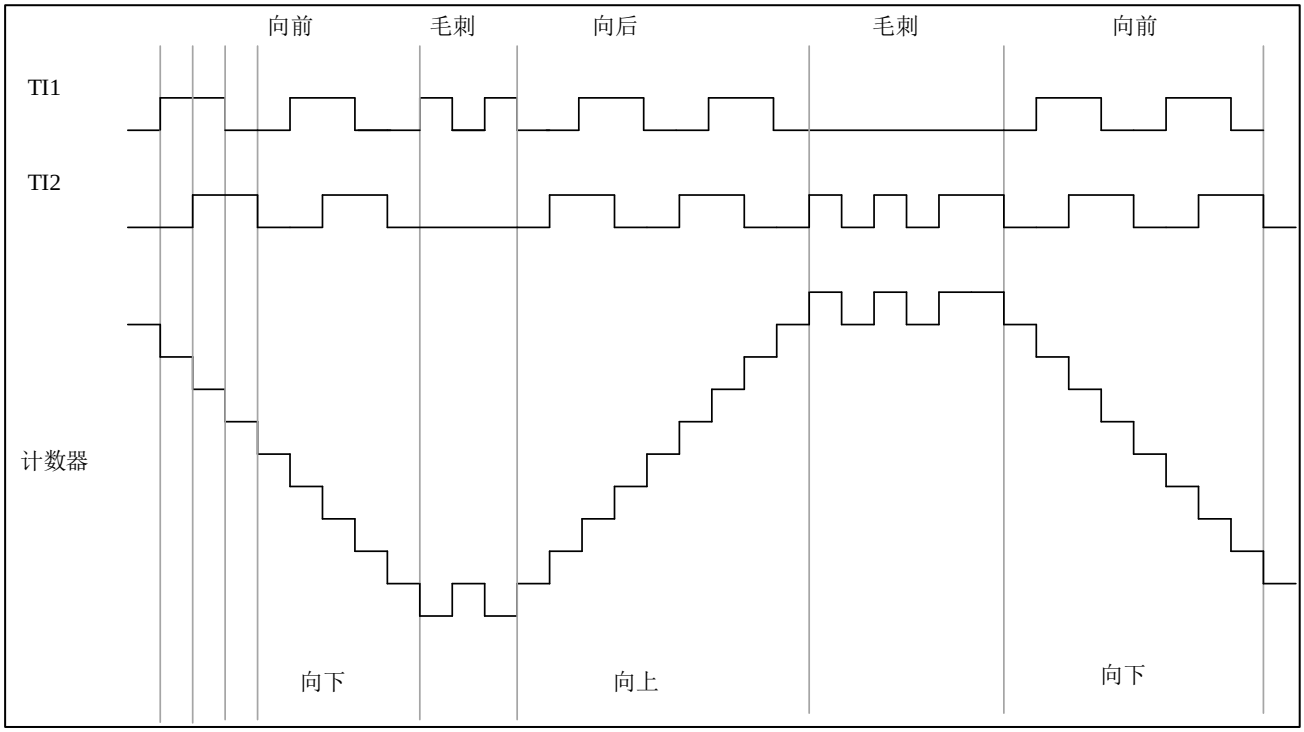
- CC1SEL='01' （TIMx_CCMOD1 寄存器，IC1FP1 映射到 TI1）
- CC2SEL='01' （TIMx_CCMOD2 寄存器，IC2FP2 映射到 TI2）
- CC1P='0' （TIMx_CCEN 寄存器，IC1FP1 不反相，IC1FP1=TI1）
- CC2P='0' （TIMx_CCEN 寄存器，IC2FP2 不反相，IC2FP2=TI2）
- SMSEL='011' （TIMx_SMCTRL 寄存器，所有的输入均在上升沿和下降沿有效）.
- CNTEN='1' （TIMx_CTRL1 寄存器，计数器使能）

图 11-42 编码器模式下的计数器操作实例



下图为当 IC1FP1 极性反相时计数器的操作实例（CC1P='1'，其他配置与上例相同）

图 11-43 IC1FP1 反相的编码器接口模式实例



定时器配置成编码器接口模式时，可以提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，可以测量两个编码器事件的间隔，获得动态的信息（速度，加速度，减速度）。指示机械过零

点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，可以把计数器的值锁存到第三个输入捕获寄存器（捕获信号必须是周期的并且可以由另一个定时器产生）；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

11.3.17 定时器输入异或功能

通过设置 TIMx_CTRL2 寄存器中的 TI1SEL 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。异或输出能够用于所有定时器的输入功能，如触发或者输入捕获。下节 11.3.18 给出了此特性用于连接霍尔传感器的例子。

11.3.18 与霍尔传感器的接口

当高级控制定时器（TIM1 或 TIM8）产生 PWM 信号驱动电机时，可以同时使用另一个通用 TIMx（TIM2、TIM3、TIM4 或 TIM5）定时器作为“接口定时器”来连接霍尔传感器，见图 11-44，3 个定时器输入脚（CC1、CC2、CC3）通过一个异或门连接到 TI1 输入通道（通过设置 TIMx_CTRL2 寄存器中的 TI1SEL 位来选择），“接口定时器”捕获这个信号。

“接口定时器”配置为从模式下的复位模式，从输入是 TI1F_ED。每当 3 个输入之一发生变化时，计数器从新从 0 开始计数。这样产生一个由霍尔输入端的任何变化而触发的时间基准。

“接口定时器”上的捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC（见图 11-27）。捕获值反映了两个输入变化间的时间延迟，延迟值给出了马达的速度信息。

“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以（通过触发一个 COM 事件）用于改变高级定时器 TIM1 或 TIM8 各个通道的属性，而高级控制定时器产生 PWM 信号驱动马达。因此“接口定时器”通道必须编程为在一个指定的延时（输出比较或 PWM 模式）之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器 TIM1 或 TIM8。

示例：

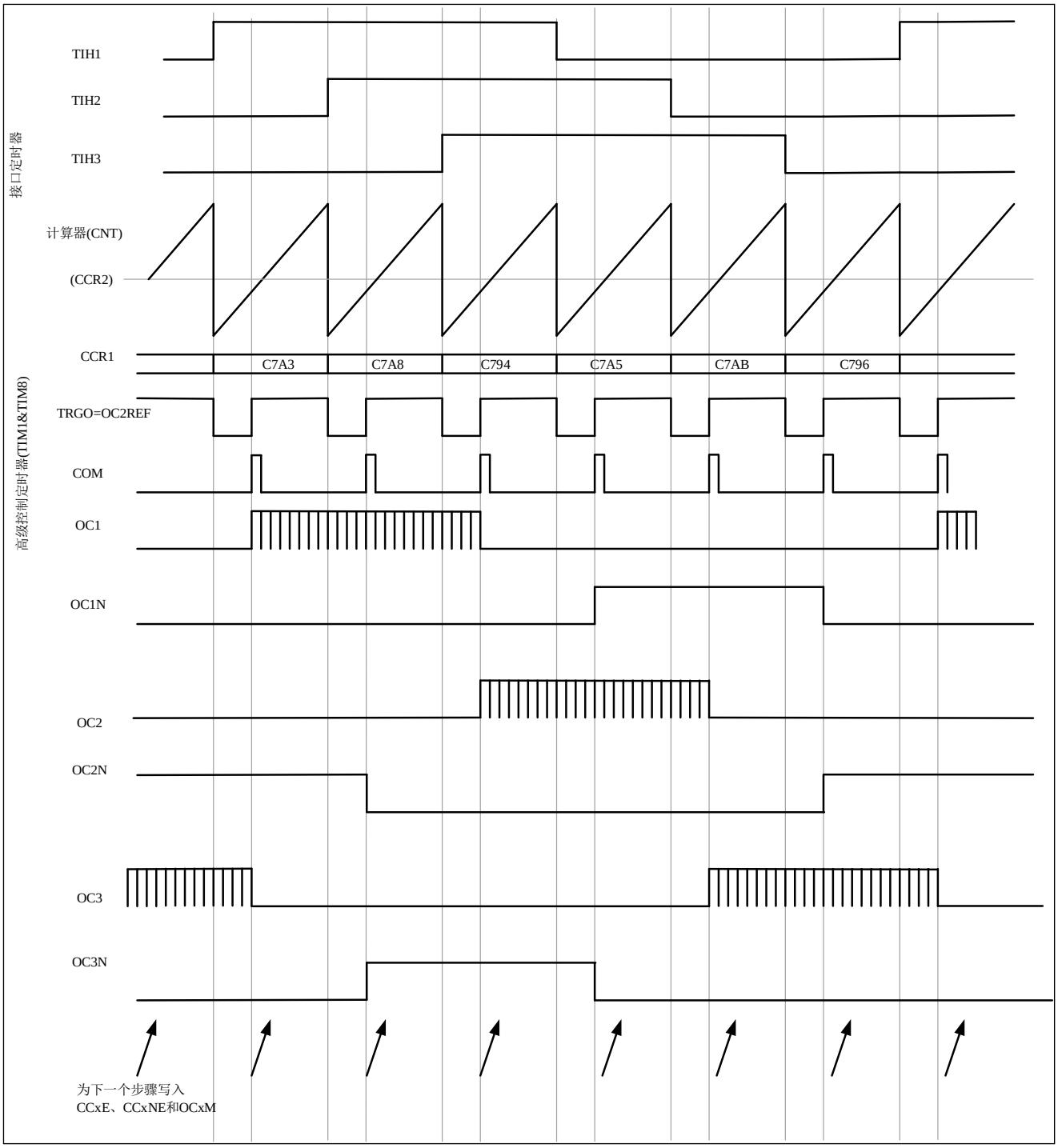
霍尔输入连接到 TIMx 定时器，要求每次任一霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器 TIMx 的 PWM 配置。

- 置 TIMx_CTRL2 寄存器的 TI1SEL 位为’1’，配置三个定时器输入逻辑或到 TI1 输入，
- 时基编程：置 TIMx_AR 为其最大值（计数器必须通过 TI1 的变化清零）。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。
- 设置通道 1 为捕获模式（选中 TRC）：置 TIMx_CCMOD1 寄存器中 CC1SEL=01，如果需要，还可以设置数字滤波器。
- 设置通道 2 为 PWM2 模式，并具有要求的延时：置 TIMx_CCMOD1 寄存器中的 OC2M=111 和 CC2SEL=00。
- 选择 OC2REF 作为 TRGO 上的触发输出：置 TIMx_CTRL2 寄存器中的 MMSEL=101。

在高级控制寄存器 TIMx 中，正确的 ITR 输入必须是触发器输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的（TIMx_CTRL2 寄存器中 CCPCTL=1），同时触发输入控制 COM 事件（TIMx_CTRL2 寄存器中 CCUSEL=1）。在一次 COM 事件后，写入下一步的 PWM 控制位（CCxEN、OCxMD），这可以在处理 OC2REF 上升沿的中断子程序里实现。

下图显示了这个示例

图 11-44 霍尔传感器接口的实例



11.3.19 TIMx 定时器和外部触发的同步

TIMx 定时器能够在多种从模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

11.3.19.1 从模式：复位模式

输入一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CTRL1 寄存器的 UPRS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器（TIMx_AR，TIMx_CCDATx）都被更新。

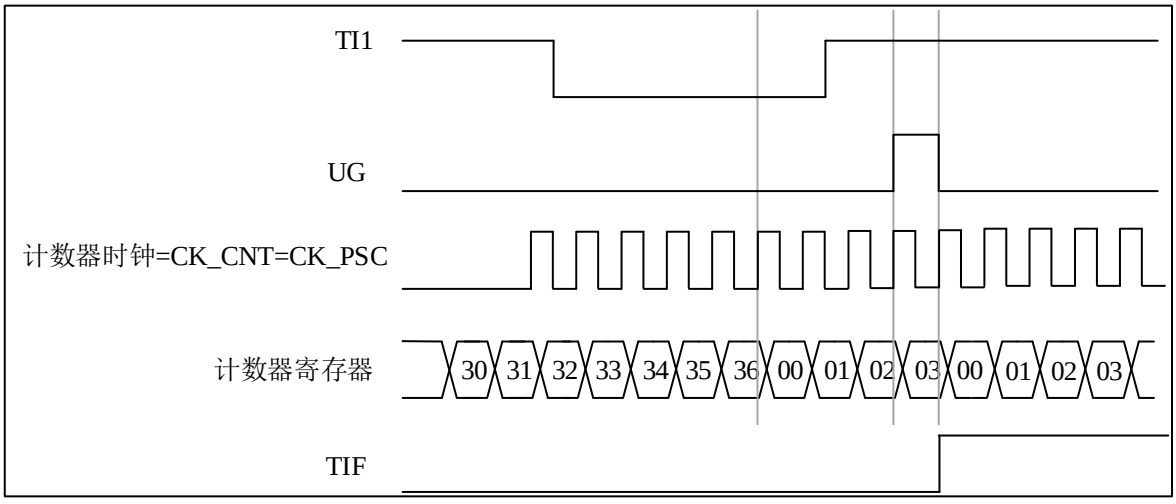
在以下的示例中，TI1 输入端的上升沿事件导致向上计数器被清零：

- 配置通道 1 以使能检测 TI1 的上升沿。配置输入滤波器的带宽（在本例中，不需要任何滤波器，因此保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置。CC1SEL 位只选择输入捕获源，即 TIMx_CCMOD1 寄存器中 CC1SEL=01。置 TIMx_CCEN 寄存器中 CC1P=0 以确定极性（只检测上升沿）。
- 置 TIMx_SMCTRL 寄存器中 SMSEL=100，配置定时器为复位模式；置 TIMx_SMCTRL 寄存器中 TSEL=101，选择 TI1 作为输入源。
- 置 TIMx_CTRL1 寄存器中 CNTEN=1，启动计数器。

计数器开始基于内部时钟进行计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志（TIMx_STS 寄存器中的 TITF 位）被设置，根据 TIMx_DINTEN 寄存器中 TIE（中断使能）位和 TDEN（DMA 使能）位的设置，产生一个中断请求或一个 DMA 请求。

下图显示了当自动重装载寄存器 TIMx_AR=0x36 时，TI1 上升沿事件到来后计数器的动作。在 TI1 上升沿和计数器的实际复位之间的延时长度取决于 TI1 输入端的重同步电路。

图 11-45 复位模式下的控制电路



11.3.19.2 从模式：门控模式

按照选中的输入端电平极性使能计数器。

在如下的示例中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽（本例中，不需要滤波，所以保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置。CC1SEL 位用于选择输入捕获源，置 TIMx_CCMOD1 寄存器中 CC1SEL=01。置 TIMx_CCEN 寄存器中 CC1P=1 以确定极性（只检测低电平）。
- 置 TIMx_SMCTRL 寄存器中 SMSEL=101，配置定时器为门控模式；置 TIMx_SMCTRL 寄存器中

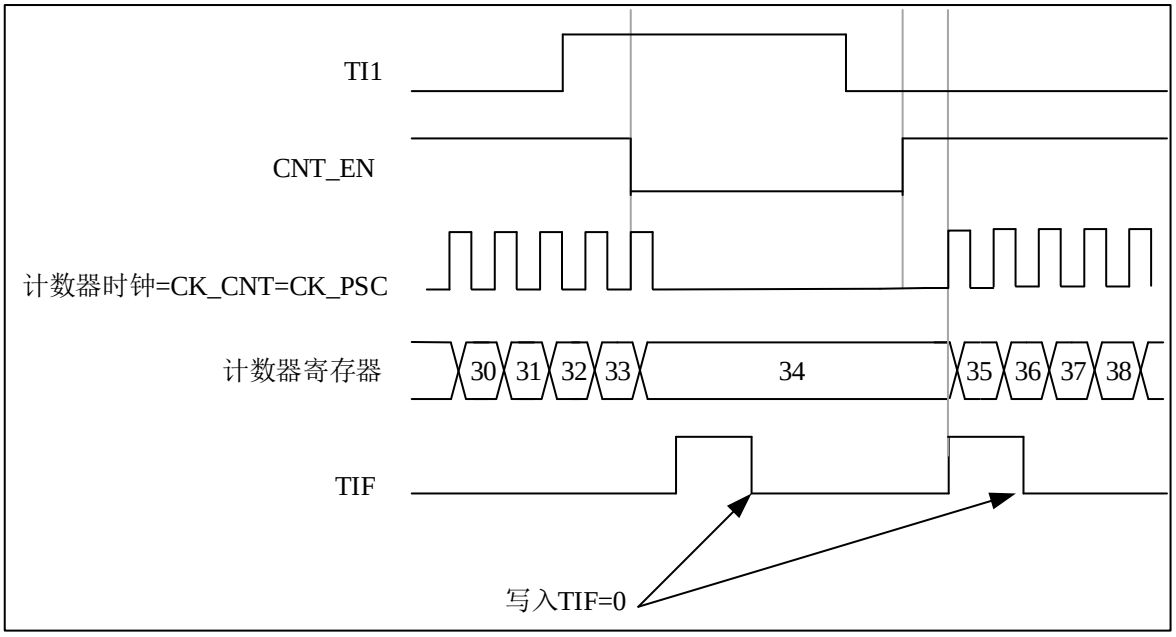
TSEL=101，选择 TI1 作为输入源。

- 置 TIMx_CTRL1 寄存器中 CNTEN=1，启动计数器。在门控模式下，如果 CNTEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟源计数，一旦 TI1 由低变高则停止计数。当计数器开始或停止时都设置 TIMx_STS 中的 TITF 标置。

TI1 上升沿和计数器实际停止之间的延时长度取决于 TI1 输入端的重同步电路。

图 11-46 门控模式下的控制电路



11.3.19.3 从模式：触发模式

在输入端上选中的事件（上升沿或下降沿）使能计数器。

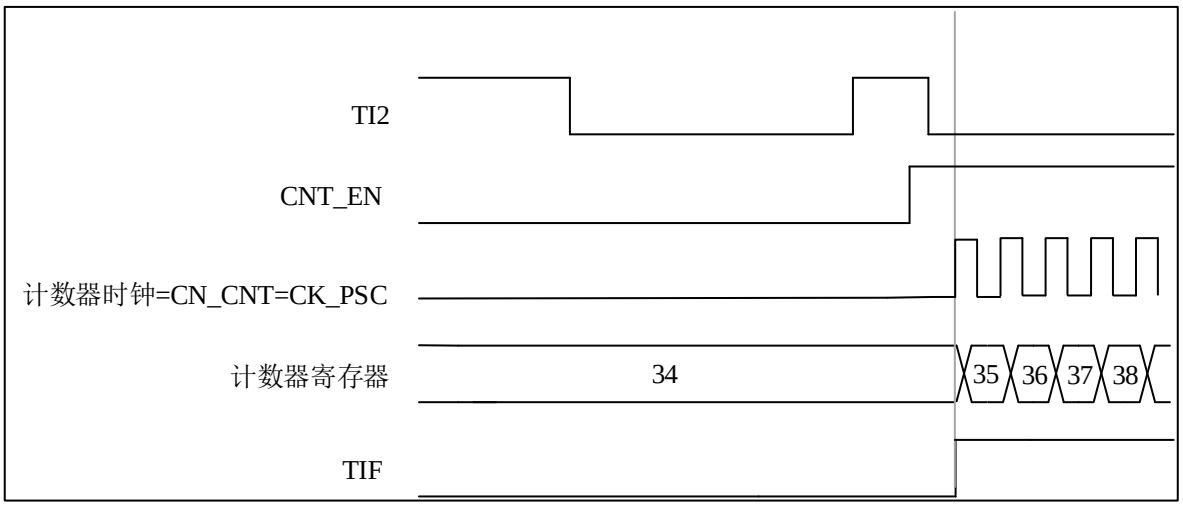
在下面的示例中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2SEL 位只用于选择输入捕获源，置 TIMx_CCMOD1 寄存器中 CC2SEL=01。置 TIMx_CCEN 寄存器中 CC2P=1 以确定极性（只检测低电平）。
- 置 TIMx_SMCTRL 寄存器中 SMSEL=110，配置定时器为触发模式；置 TIMx_SMCTRL 寄存器中 TS=110，选择 TI2 作为输入源。

只要 TI2 出现一个上升沿时，计数器开始在内部时钟源驱动下进行计数，同时设置 TITF 标志。

TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

图 11-47 触发器模式下的控制电路



11.3.19.4 从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可以与另一种从模式（外部时钟模式 1 和编码器模式除外）一起使用。这时 ETR 信号被用作外部时钟源的输入，在复位模式、门控模式或触发模式可以选择另一个输入作为触发输入。不建议使用 TIMx_SMCTRL 寄存器的 TSEL 位选择 ETR 作为 TRGI。

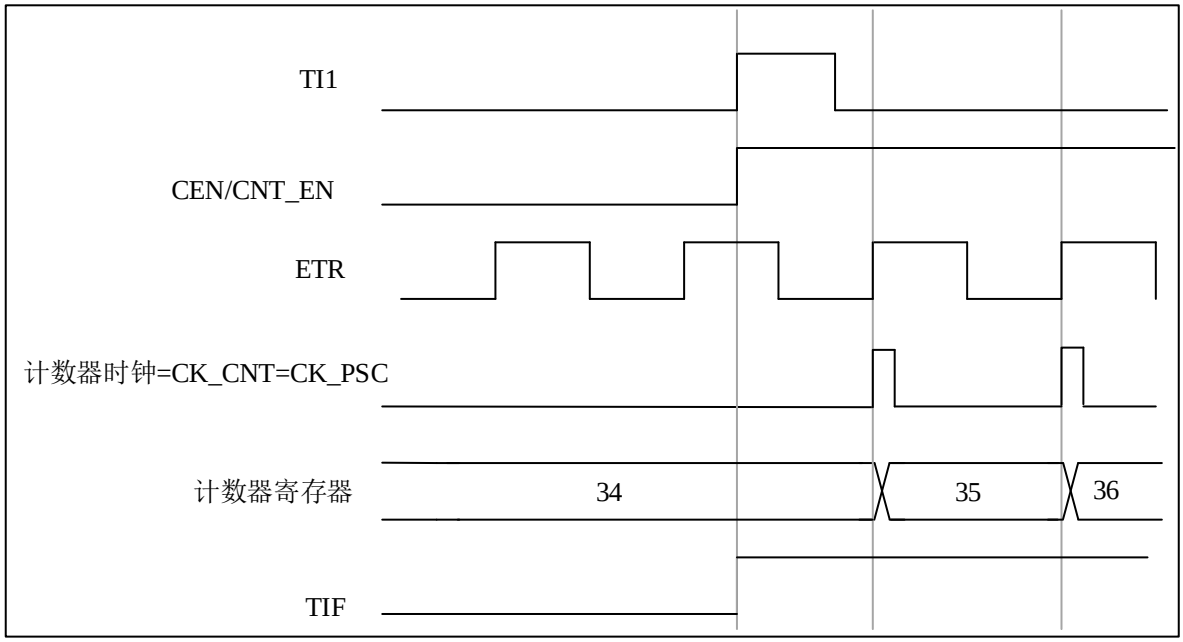
在下面的例子中，一旦在 TI1 上出现一个上升沿，计数器在 ETR 的每一个上升沿向上计数一次：

- 通过 TIMx_SMCTRL 寄存器配置外部触发输入电路：
  - EXTF=0000：没有滤波
  - EXTPS=00：不用预分频器
  - EXTP=0：检测 ETR 的上升沿，置 EXCEN=1 使能外部时钟模式 2。
- 按如下配置通道 1，检测 TI 的上升沿：
  - IC1F=0000：没有滤波
  - 触发操作中不使用捕获预分频器，不需要配置
  - 置 TIMx_CCMOD1 寄存器中 CC1SEL=01，选择输入捕获源
  - 置 TIMx_CCEN 寄存器中 CC1P=0 以确定极性（只检测上升沿）
- 置 TIMx_SMCTRL 寄存器中 SMSEL=110，配置定时器为触发模式。置 TIMx_SMCTRL 寄存器中 TSEL=101，选择 TI1 作为输入源。

  在 TI1 上出现一个上升沿时，TITF 标志被设置，计数器开始在 ETR 的上升沿计数。

  ETR 信号的上升沿和计数器实际复位间的延时长度，取决于 ETRP 输入端的重同步电路。

图 11-48 外部时钟模式 2＋触发模式下的控制电路



11.3.20 定时器同步

所有 TIM 定时器在内部相连，用于定时器同步或链接。详见下一章 12.3.15 节。

11.3.21 调试模式

当微控制器进入调试模式时（N308 核心停止），根据 DBG 模块中 DBG_TIMx_STOP 的设置值，TIMx 计数器可以继续正常操作，或者停止。详见随后的 29.3.2 节。

11.4 TIM1 和 TIM8 寄存器描述

关于在寄存器描述里面所用到的缩写，详见第 1.1 节。

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

11.4.1 TIM1 和 TIM8 寄存器图

下表中将 TIM1 和 TIM8 的所有寄存器映射到一个 16 位可寻址（编址）空间。

表 11-2 TIM1 和 TIM8–寄存器图和复位值

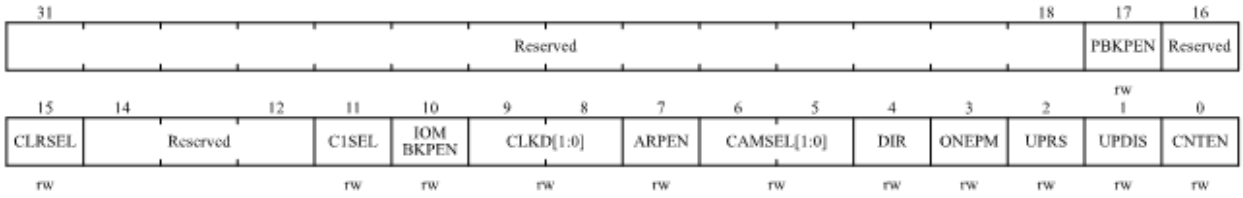
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	TIMx_CTRL1	Reserved														PBKPEN	Reserved	CLRSEL	C4SEL	C3SEL	C2SEL	C1SEL	IOMBKPEN	CLKD[1:0]		ARPEN	CAMSEL[1:0]		DIR	ONEPM	UPRS	UPDIS	CNTEN
	Reset Value															0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
004h	TIMx_CTRL2	Reserved													OI6	Reserved	OI5	Reserved	OI4	OI3N	OI3	OI2N	OI2	OI1N	OI1	TI1SEL	MMSEL[2:0]			CCDSEL	CCUSEL	Reserved	CCPCTL
	Reset Value														0		0		0	0	0	0	0	0	0	0	0	0	0	0	0		0
008h	TIMx_SMCTRL	Reserved																EXTP	EXCEN	EXTPS[1:0]		EXTF[3:0]				MSMD	TSEL[2:0]			Reserved	SMSEL[2:0]			
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
00Ch	TIMx_DINTEN	Reserved																	TDEN	COMDEN	CC4DEN	CC3DEN	CC2DEN	CC1DEN	UDEN	BIEN	TIEN	COMIEN	CC4IEN	CC3IEN	CC2IEN	CC1IEN	UIEN
	Reset Value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	TIMx_STS	Reserved														CC6ITF	CC5ITF	Reserved			CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved	BITF	TITF	COMITF	CC4ITF	CC3ITF	CC2ITF	CC1ITF	UDITF
	Reset Value															0	0				0	0	0	0		0	0	0	0	0	0	0	0
014h	TIMx_EVTGEN	Reserved																								BGN	TGN	CC1DGN	CC4GN	CC3GN	CC2GN	CC1GN	UDGN
	Reset Value																									0	0	0	0	0	0	0	0
018h	TIMx_CCMOD1 输出比较模式	Reserved																OC2CEN	OC2M[2:0]			OC2PEN	OC2FEN	CC2SEL[1:0]		OC1CEN	OC1M[2:0]			OC1PEN	OC1FEN	CC1SEL[1:0]	
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMOD1 输入捕获模式	Reserved																IC2F[3:0]				IC2PSC[1:0]		CC2SEL[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1SEL[1:0]	
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	TIMx_CCMOD2 输出比较模式	Reserved																OC4CEN	OC4M[2:0]			OC4PEN	OC4FEN	CC4SEL[1:0]		OC3CEN	OC3M[2:0]			OC3PEN	OC3FEN	CC3SEL[1:0]	
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMOD2 输入捕获模式	Reserved																IC4F[3:0]				IC4PSC[1:0]		CC4SEL[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3SEL[1:0]	
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
020h	TIMx_CCEN	Reserved											CC6P	CC6EN	Reserved	CC5P	CC5EN	Reserved	CC4P	CC4EN	CC3NP	CC3NEN	CC3P	CC3EN	CC2NP	CC2NEN	CC2P	CC2EN	CC1NP	CC1NEN	CC1P	CC1EN
	Reset Value												0	0		0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
024h	TIMx_CNT	Reserved																CNT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028h	TIMx_PSC	Reserved																PSC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	TIMx_AR	Reserved																AR[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
030h	TIMx_REPCNT	Reserved																								REPCNT[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0
034h	TIMx_CCDAT1	Reserved																CCDAT1[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
038h	TIMx_CCDAT2	Reserved																CCDAT2[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03Ch	TIMx_CCDAT3	Reserved																CCDAT3[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
040h	TIMx_CCDAT4	Reserved																CCDAT4[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
044h	TIMx_BKDT	Reserved																MOEN	AOEN	BKP	BKEN	OSSR	OSSI	LCKCFG[1:0]		DTGN[7:0]							
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
048h	TIMx_DCTRL	Reserved																			DBLEN[4:0]					Reserved			DBADDR[4:0]				
	Reset Value																				0	0	0	0	0				0	0	0	0	0
04Ch	TIMx_DADDR	Reserved																BURST[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
054h	TIMx_CCMOD3	Reserved																OC6CEN	OC6MD[2:0]			OC6PEN	OC6FEN	Reserved		OC5CEN	OC5MD[2:0]			OC5PEN	OC5FEN	Reserved	
	Reset Value																	0	0	0	0	0	0			0	0	0	0	0	0		
058h	TIMx_CCDAT5	Reserved																CCDAT5[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
05Ch	TIMx_CCDAT6	Reserved																CCDAT6[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.4.2 TIM1 和 TIM8 控制寄存器 1（TIMx_CTRL1）

偏移地址：0x00
复位值：0x0000



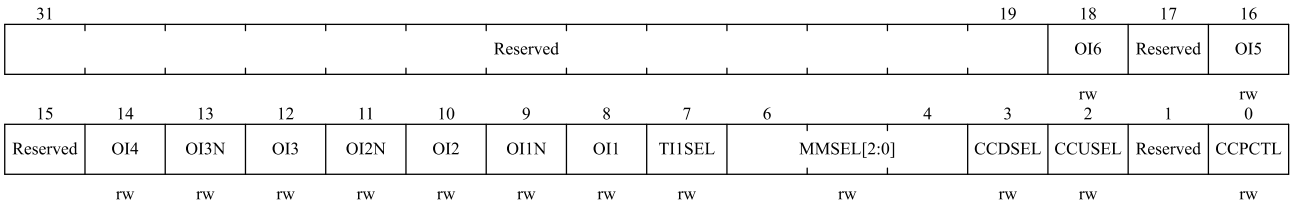
位域	名称	描述
31:18	Reserved	保留，硬件强制为 0。
17	PBKPEN	PVD 作为 BRK 启用（PVD as brk Enable） *注：操作此位前必须先开启芯片的扩展模式(置位 EXMODE)* 0：禁止 1：使能
16	Reserved	保留
15	CLRSEL	OCxRef 选择（OCxRef selection） *注：操作此位前必须先开启芯片的扩展模式(置位 EXMODE)* 0：选择外部 Ocxclr （ETR） 信号 1：选择内部 OCxclr（来自比较器）信号
14:12	Reserved	保留，硬件强制为 0。
11	C1SEL	通道 1 选择（Channel 1 selection） *注：操作此位前必须先开启芯片的扩展模式(置位 EXMODE)* 0：选择外部 CH1（来自 IOM）信号 1：选择内部 CH1（来自 COMP）信号
10	IOMBKPEN	IOM 作为 BKP 使能（IOM as bkp Enable）

位域	名称	描述
		注：操作此位前必须先开启芯片的扩展模式(置位 EXMODE) 0：使能 1：禁止
9:8	CLKD[1:0]	时钟分频因子 （Clock division） 这 2 位定义在定时器时钟（CK_INT）频率、死区时间和由死区发生器与数字滤波器（ETR,TIx）所用的采样时钟之间的分频比例。 00：$t_{DTS} = t_{CK\_INT}$ 01：$t_{DTS} = 2 \times t_{CK\_INT}$ 10：$t_{DTS} = 4 \times t_{CK\_INT}$ 11：保留，不要使用这个配置
7	ARPEN	自动重装载预装载允许位 （Auto-reload preload enable） 0：TIMx_AR 寄存器没有缓冲 1：TIMx_AR 寄存器被装入缓冲器
6:5	CAMSEL[1:0]	选择中央对齐模式 （Center-aligned mode selection） 00：边沿对齐模式。计数器依据方向位（DIR）向上或向下计数。 01：中央对齐模式 1。计数器交替地向上和向下计数。配置为输出的通道（TIMx_CCMODx 寄存器中 CCxSEL=00）的输出比较中断标志位，只在计数器向下计数时被设置。 10：中央对齐模式 2。计数器交替地向上和向下计数。配置为输出的通道（TIMx_CCMODx 寄存器中 CCxSEL=00）的输出比较中断标志位，只在计数器向上计数时被设置。 11：中央对齐模式 3。计数器交替地向上和向下计数。配置为输出的通道（TIMx_CCMODx 寄存器中 CCxSEL=00）的输出比较中断标志位，在计数器向上和向下计数时均被设置。 *注：在计数器开启时（CNTEN=1），不允许从边沿对齐模式转换到中央对齐模式。*
4	DIR	方向 （Direction） 0：计数器向上计数； 1：计数器向下计数。 *注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。*
3	ONEPM	单脉冲模式 （One pulse mode） 0：在发生更新事件时，计数器不停止； 1：在发生下一次更新事件（清除 CNTEN 位）时，计数器停止。
2	UPRS	更新请求源 （Update request source） 软件通过该位选择 UEV 事件的源 0：如果使能了更新中断或 DMA 请求，则下述任一事件产生更新中断或 DMA 请求： – 计数器溢出/下溢 – 设置 UDGN 位 – 从模式控制器产生的更新 1：如果使能了更新中断或 DMA 请求，则只有计数器溢出/下溢才产生更新中断或 DMA 请求。

位域	名称	描述
1	UPDIS	禁止更新 （Update disable） 软件通过该位允许/禁止 UEV 事件的产生 0：允许 UEV。更新（UEV）事件由下述任一事件产生： – 计数器溢出/下溢 – 设置 UDGN 位 – 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。（译注：更新影子寄存器） 1：禁止 UEV。不产生更新事件，影子寄存器（AR、PSC、CCDATx）保持它们的值。如果设置了 UDGN 位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CNTEN	使能计数器 （Counter enable） 0：禁止计数器； 1：使能计数器。 *注：在软件设置了 CNTEN 位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置 CNTEN 位。*

11.4.3 TIM1 和 TIM8 控制寄存器 2（TIMx_CTRL2）

偏移地址：0x04
复位值：0x0000



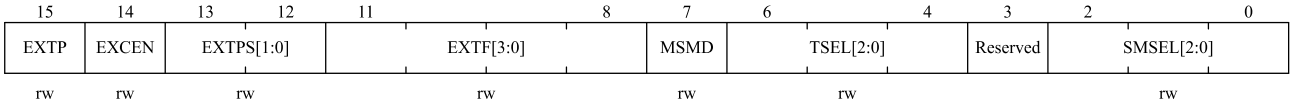
位域	名称	描述
31:19	Reserved	保留，硬件强制为0。
18	OI6	输出空闲状态6（OC6输出）。参见OI1位。
17	Reserved	保留，硬件强制为0。
16	OI5	输出空闲状态5（OC5输出）。参见OI1位。
15	Reserved	保留，硬件强制为0。
14	OI4	输出空闲状态4（OC4输出）。参见OI1位。
13	OI3N	输出空闲状态3（OC3N输出）。参见OI1N位。
12	OI3	输出空闲状态3（OC3输出）。参见OI1位。
11	OI2N	输出空闲状态2（OC2N输出）。参见OI1N位。
10	OI2	输出空闲状态2（OC2输出）。参见OI1位。

位域	名称	描述
9	OI1N	输出空闲状态1（OC1N输出） （Output Idle state 1） 0：当MOEN=0时，死区后OC1N=0； 1：当MOEN=0时，死区后OC1N=1。 *注：已经设置了LCKCFG（TIMx_BKR寄存器）级别1、2或3后，该位不能被修改。*
8	OI1	输出空闲状态1（OC1输出） （Output Idle state 1） 0：当MOEN=0时，如果实现了OC1N，则死区后OC1=0； 1：当MOEN=0时，如果实现了OC1N，则死区后OC1=1。 *注：已经设置了LCKCFG（TIMx_BKR寄存器）级别1、2或3后，该位不能被修改。*
7	TI1SEL	TI1选择 （TI1 selection） 0：TIMx_CH1引脚连到TI1输入； 1：TIMx_CH1、TIMx_CH2和TIMx_CH3引脚经异或后连到TI1输入。
6:4	MMSEL[2:0]	主模式选择 （Master mode selection） 这3位用于选择在主模式下送到从定时器的同步信息（TRGO）。可能的组合如下： 000：**复位** – TIMx_EVTGEN寄存器的UDGN位被用于作为触发输出（TRGO）。如果是触发输入产生的复位（从模式控制器处于复位模式），则TRGO上的信号相对实际的复位会有一个延迟。 001：**使能** – 计数器使能信号CNTEN被用于作为触发输出（TRGO）。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过CNTEN控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO上会有一个延迟，除非选择了主/从模式（见TIMx_SMCTRL寄存器中MSMD位的描述）。 010：**更新** – 更新事件被选为触发输入（TRGO）。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011：**比较脉冲** – 在发生一次捕获或一次比较成功时，当要设置CC1ITF标志时（即使它已经为高），触发输出送出一个正脉冲（TRGO）。 100：**比较** – OC1REF信号被用于作为触发输出（TRGO）。 101：**比较** – OC2REF信号被用于作为触发输出（TRGO）。 110：**比较** – OC3REF信号被用于作为触发输出（TRGO）。 111：**比较** – OC4REF信号被用于作为触发输出（TRGO）。
3	CCDSEL	捕获/比较的DMA选择 （Capture/compare DMA selection） 0：当发生CCx事件时，送出CCx的DMA请求； 1：当发生更新事件时，送出CCx的DMA请求。
2	CCUSEL	捕获/比较控制更新选择 （Capture/compare control update selection） 0：如果捕获/比较控制位是预装载的（CCPCTL =1），只能通过设置COM位更新它们； 1：如果捕获/比较控制位是预装载的（CCPCTL =1），可以通过设置COM位或TRGI上的一个上升沿更新它们。 *注：该位只对具有互补输出的通道起作用。*
1	Reserved	保留，硬件强制为0。
0	CCPCTL	捕获/比较预装载控制位 （Capture/compare preloaded control） 0：CCxEN，CCxNEN和OCxMD位不是预装载的； 1：CCxEN，CCxNEN和OCxMD位是预装载的；设置该位后，它们只在设置了COM位后被更新。

位域	名称	描述
		注：该位只对具有互补输出的通道起作用。

11.4.4 TIM1 和 TIM8 从模式控制寄存器（TIMx_SMCTRL）

偏移地址：0x08
复位值：0x0000



位域	名称	描述
15	EXTP	外部触发极性 （External trigger polarity） 该位选择是用ETR还是ETR的反相来作为触发操作 0：ETR不反相，高电平或上升沿有效； 1：ETR被反相，低电平或下降沿有效。
14	EXCEN	外部时钟使能位 （External clock enable） 该位启用外部时钟模式2 0：禁止外部时钟模式2； 1：使能外部时钟模式2。计数器由ETRF信号上的任意有效边沿驱动。 *注1：设置EXCEN位与选择外部时钟模式1并将TRGI连到ETRF（SMSEL=111和TSEL=111）具有相同功效。* *注2：下述从模式可以与外部时钟模式2同时使用：复位模式，门控模式和触发模式；但是，这时TRGI不能连到ETRF（TSEL位不能是'111'）。* *注3：外部时钟模式1和外部时钟模式2同时被使能时，外部时钟的输入是ETRF。*
13:12	EXTPS[1:0]	外部触发预分频 （External trigger prescaler） 外部触发信号ETRP的频率必须最多是TIMxCLK频率的1/4。当输入较快的外部时钟时，可以使用预分频降低ETRP的频率。 00：关闭预分频； 01：ETRP频率除以2； 10：ETRP频率除以4； 11：ETRP频率除以8。
11:8	EXTF[3:0]	外部触发滤波 （External trigger filter） 这些位定义了对ETRP信号采样的频率和对ETRP数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到N个事件后会产生一个输出的跳变。 0000：无滤波器，以fDTS采样 1000：采样频率fSAMPLING=fDTS/8，N=6 0001：采样频率fSAMPLING=fCK_INT，N=2 1001：采样频率fSAMPLING=fDTS/8，N=8 0010：采样频率fSAMPLING=fCK_INT，N=4 1010：采样频率fSAMPLING=fDTS/16，N=5 0011：采样频率fSAMPLING=fCK_INT，N=8 1011：采样频率fSAMPLING=fDTS/16，N=6 0100：采样频率fSAMPLING=fDTS/2，N=61100：采样频率fSAMPLING=fDTS/16，N=8 0101：采样频率fSAMPLING=fDTS/2，N=81101：采样频率fSAMPLING=fDTS/32，N=5

位域	名称	描述
		0110：采样频率fSAMPLING=fDTS/4，N=61110：采样频率fSAMPLING=fDTS/32，N=6 0111：采样频率fSAMPLING=fDTS/4，N=81111：采样频率fSAMPLING=fDTS/32，N=8
7	MSMD	主/从模式 （Master/slave mode） 0：无作用； 1：触发输入（TRGI）上的事件被延迟了，以允许在当前定时器（通过TRGO）与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
6:4	TSEL[2:0]	触发选择 （Trigger selection） 这3位选择用于同步计数器的触发输入。 000：内部触发0（IN_TR0） 100：TI1的边沿检测器（TI1F_ED） 001：内部触发1（IN_TR1） 101：滤波后的定时器输入1（TI1FP1） 010：内部触发2（IN_TR2） 110：滤波后的定时器输入2（TI2FP2） 011：内部触发3（IN_TR3） 111：外部触发输入（ETRF） 更多有关ITRx的细节，参见表 11-3。 *注：这些位只能在未用到（如SMSEL=000）时被改变，以避免在改变时产生错误的边沿检测。*
3	Reserved	保留，硬件强制为0。
2:0	SMSEL[2:0]	从模式选择 （Slave mode selection） 当选择了外部信号，触发信号（TRGI）的有效边沿与选中的外部输入极性相关（见输入控制寄存器和控制寄存器的说明） 000：关闭从模式 – 如果CNTEN=1，则预分频器直接由内部时钟驱动。 001：编码器模式1 – 根据TI1FP1的电平，计数器在TI2FP2的边沿向上/下计数。 010：编码器模式2 – 根据TI2FP2的电平，计数器在TI1FP1的边沿向上/下计数。 011：编码器模式3 – 根据另一个信号的输入电平，计数器在TI1FP1和TI2FP2的边沿向上/下计数。 100：复位模式 – 选中的触发输入（TRGI）的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 101：门控模式 – 当触发输入（TRGI）为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止（但不复位）。计数器的启动和停止都是受控的。 110：触发模式 – 计数器在触发输入TRGI的上升沿启动（但不复位），只有计数器的启动是受控的。 111：外部时钟模式1 – 选中的触发输入（TRGI）的上升沿驱动计数器。 *注：如果TI1F_EN被选为触发输入（TSEL=100）时，不要使用门控模式。这是因为，TI1F_ED在每次TI1F变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。*

表 11-3 TIMx 内部触发连接

从定时器	ITR0 （TSEL=000）	ITR1 （TSEL=001）	ITR2 （TSEL=010）	ITR3 （TSEL=011）
TIM1	TIM5	TIM2	TIM3	TIM4
TIM8	TIM1	TIM2	TIM4	TIM5

11.4.5 TIM1 和 TIM8 DMA/中断使能寄存器（TIMx_DINTEN）

偏移地址：0x0C

复位值：0x0000

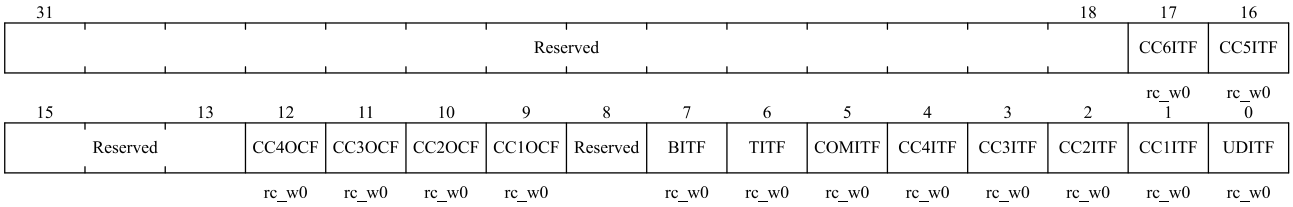
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDEN	COMDEN	CC4DEN	CC3DEN	CC2DEN	CC1DEN	UDEN	BIEN	TIEN	COMIEN	CC4IEN	CC3IEN	CC2IEN	CC1IEN	UIEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
15	Reserved	保留，硬件强制为0。
14	TDEN	允许触发DMA请求 （Trigger DMA request enable） 0：禁止触发DMA请求； 1：允许触发DMA请求。
13	COMDEN	允许COM的DMA请求 （COM DMA request enable） 0：禁止COM的DMA请求； 1：允许COM的DMA请求。
12	CC4DEN	允许捕获/比较4的DMA请求 （Capture/Compare 4 DMA request enable） 0：禁止捕获/比较4的DMA请求； 1：允许捕获/比较4的DMA请求。
11	CC3DEN	允许捕获/比较3的DMA请求 （Capture/Compare 3 DMA request enable） 0：禁止捕获/比较3的DMA请求； 1：允许捕获/比较3的DMA请求。
10	CC2DEN	允许捕获/比较2的DMA请求 （Capture/Compare 2 DMA request enable） 0：禁止捕获/比较2的DMA请求； 1：允许捕获/比较2的DMA请求。
9	CC1DEN	允许捕获/比较1的DMA请求 （Capture/Compare 1 DMA request enable） 0：禁止捕获/比较1的DMA请求； 1：允许捕获/比较1的DMA请求。
8	UDEN	允许更新的DMA请求 （Update DMA request enable） 0：禁止更新的DMA请求； 1：允许更新的DMA请求。
7	BIEN	允许刹车中断 （Break interrupt enable） 0：禁止刹车中断； 1：允许刹车中断。

位域	名称	描述
6	TIEN	触发中断使能 （Trigger interrupt enable） 0：禁止触发中断； 1：使能触发中断。
5	COMIEN	允许COM中断 （COM interrupt enable） 0：禁止COM中断； 1：允许COM中断。
4	CC4IEN	允许捕获/比较4中断 （Capture/Compare 4 interrupt enable） 0：禁止捕获/比较4中断； 1：允许捕获/比较4中断。
3	CC3IEN	允许捕获/比较3中断 （Capture/Compare 3 interrupt enable） 0：禁止捕获/比较3中断； 1：允许捕获/比较3中断。
2	CC2IEN	允许捕获/比较2中断 （Capture/Compare 2 interrupt enable） 0：禁止捕获/比较2中断； 1：允许捕获/比较2中断。
1	CC1IEN	允许捕获/比较1中断 （Capture/Compare 1 interrupt enable） 0：禁止捕获/比较1中断； 1：允许捕获/比较1中断。
0	UIEN	允许更新中断 （Update interrupt enable） 0：禁止更新中断； 1：允许更新中断。

11.4.6 TIM1 和 TIM8 状态寄存器（TIMx_STS）

偏移地址：0x10
复位值：0x0000



位域	名称	描述
31:18	Reserved	保留，硬件强制为0。
17	CC6ITF	捕获/比较6中断标记 （Capture/Compare 6 interrupt flag） 参考CC1ITF描述。
16	CC5ITF	捕获/比较5中断标记 （Capture/Compare 5 interrupt flag） 参考CC1ITF描述。
15:13	Reserved	保留，硬件强制为0。

位域	名称	描述
12	CC4OCF	捕获/比较4重复捕获标记 （Capture/Compare 4 overcapture flag） 参见CC1OCF描述。
11	CC3OCF	捕获/比较3重复捕获标记 （Capture/Compare 3 overcapture flag） 参见CC1OCF描述。
10	CC2OCF	捕获/比较2重复捕获标记 （Capture/Compare 2 overcapture flag） 参见CC1OCF描述。
9	CC1OCF	捕获/比较1重复捕获标记 （Capture/Compare 1 overcapture flag） 仅当相应的通道被配置为输入捕获时，该标记可由硬件置1。写0可清除该位。 0：无重复捕获产生； 1：计数器的值被捕获到TIMx_CCDAT1寄存器时，CC1ITF的状态已经为‘1’。
8	Reserved	保留，硬件强制为0。
7	BITF	刹车中断标记 （Break interrupt flag） 一旦刹车输入有效，由硬件对该位置’1’。如果刹车输入无效，则该位可由软件清’0’。 0：无刹车事件产生； 1：刹车输入上检测到有效电平。
6	TITF	触发器中断标记 （Trigger interrupt flag） 当发生触发事件（当从模式控制器处于除门控模式外的其它模式时，在TRGI输入端检测到有效边沿，或门控模式下的任一边沿）时由硬件对该位置’1’。它由软件清’0’。 0：无触发器事件产生； 1：触发中断等待响应。
5	COMITF	COM中断标记 （COM interrupt flag） 一旦产生COM事件（当捕获/比较控制位：CCxEN、CCxNEN、OCxMD已被更新）该位由硬件置’1’。它由软件清’0’。 0：无COM事件产生； 1：COM中断等待响应。
4	CC4ITF	捕获/比较4中断标记 （Capture/Compare 4 interrupt flag） 参考CC1ITF描述。
3	CC3ITF	捕获/比较3中断标记 （Capture/Compare 3 interrupt flag） 参考CC1ITF描述。
2	CC2ITF	捕获/比较2中断标记 （Capture/Compare 2 interrupt flag） 参考CC1ITF描述。
1	CC1ITF	捕获/比较1中断标记 （Capture/Compare 1 interrupt flag） **如果通道CC1配置为输出模式：** 当计数器值与比较值匹配时该位由硬件置1，但在中心对称模式下除外（参考TIMx_CTRL1寄存器的CAMSEL位）。它由软件清’0’。 0：无匹配发生； 1：TIMx_CNT的值与TIMx_CCDAT1的值匹配。 当TIMx_CCDAT1的内容大于TIMx_AR的内容时，在向上或向上/下计数模式时计数器溢出，或向下计数模式时的计数器下溢条件下，CC1ITF位变高 **如果通道CC1配置为输入模式：** 当捕获事件发生时该位由硬件置’1’，它由软件清’0’或通过读TIMx_CCDAT1清’0’。 0：无输入捕获产生；

位域	名称	描述
		1：计数器值已被捕获（拷贝）至TIMx_CCDAT1（在IC1上检测到与所选极性相同的边沿）。
0	UDITF	更新中断标记 （Update interrupt flag） 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0：无更新事件产生； 1：更新中断等待响应。当寄存器被更新时该位由硬件置'1'： – 若TIMx_CTRL1寄存器的UPDIS=0，当重复计数器数值上溢或下溢时（重复计数器=0时产生更新事件）。 – 若TIMx_CTRL1寄存器的UPRS=0、UPDIS =0，当设置TIMx_EVTGEN寄存器的UDGN=1时产生更新事件，通过软件对计数器CNT重新初始化时。 – 若TIMx_CTRL1寄存器的UPRS=0、UPDIS =0，当计数器CNT被触发事件重新初始化时。（参考11.4.4: TIM1和TIM8从模式控制寄存器（TIMx_SMCTRL））。

11.4.7 TIM1 和 TIM8 事件产生寄存器（TIMx_EVTGEN）

偏移地址:0x14
复位值:0x0000

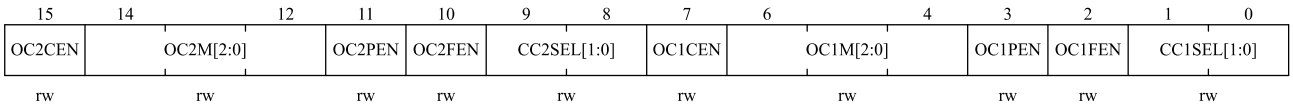
15 … 8	7	6	5	4	3	2	1	0
Reserved	BGN	TGN	CCUDGN	CC4GN	CC3GN	CC2GN	CC1GN	UDGN
	w	w	w	w	w	w	w	w

位域	名称	描述
15:8	Reserved	保留，硬件强制为0。
7	BGN	产生刹车事件 （Break generation） 该位由软件置'1'，用于产生一个刹车事件，由硬件自动清'0'。 0：无动作； 1：产生一个刹车事件。此时MOEN=0、BITF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。
6	TGN	产生触发事件 （Trigger generation） 该位由软件置'1'，用于产生一个触发事件，由硬件自动清'0'。 0：无动作； 1：TIMx_STS寄存器的TITF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。
5	CCUDGN	捕获/比较事件，产生控制更新 （Capture/Compare control update generation） 该位由软件置'1'，由硬件自动清'0'。 0：无动作； 1：当CCPCTL=1，允许更新CCxEN、CCxNEN、OCxMD位。 *注：该位只对拥有互补输出的通道有效。*
4	CC4GN	产生捕获/比较4事件 （Capture/Compare 4 generation） 参考CC1GN描述。
3	CC3GN	产生捕获/比较3事件 （Capture/Compare 3 generation） 参考CC1GN描述。
2	CC2GN	产生捕获/比较2事件 （Capture/Compare 2 generation） 参考CC1GN描述。

位域	名称	描述
1	CC1GN	产生捕获/比较1事件 （Capture/Compare 1 generation） 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。 0：无动作； 1：在通道CC1上产生一个捕获/比较事件： **若通道CC1配置为输出：** 设置CC1ITF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。 **若通道CC1配置为输入：** 当前的计数器值被捕获至TIMx_CCDAT1寄存器；设置CC1ITF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。若CC1ITF已经为1，则设置CC1OCF=1。
0	UDGN	产生更新事件 （Update generation） 该位由软件置'1'，由硬件自动清'0'。 0：无动作； 1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清'0'（但是预分频系数不变）。若在中心对称模式下或DIR=0（向上计数）则计数器被清'0'；若DIR=1（向下计数）则计数器取TIMx_AR的值。

11.4.8 TIM1 和 TIM8 捕获/比较模式寄存器 1（TIMx_CCMOD1）

偏移地址：0x18
复位值：0x0000
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 CCxSEL 位定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。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：

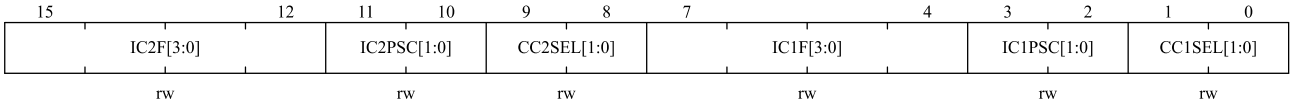


位域	名称	描述
15	OC2CEN	输出比较2清0使能 （Output Compare 2 clear enable）
14:12	OC2M[2:0]	输出比较2模式 （Output Compare 2 mode）
11	OC2PEN	输出比较2预装载使能 （Output Compare 2 preload enable）
10	OC2FEN	输出比较2快速使能 （Output Compare 2 fast enable）
9:8	CC2SEL[1:0]	捕获/比较2选择。（Capture/Compare 2 selection） 该位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。*

位域	名称	描述
7	OC1CEN	输出比较1清'0'使能 （Output Compare 1 clear enable） 0：OC1REF 不受ETRF输入的影响； 1：一旦检测到ETRF输入高电平，清除OC1REF=0。
6:4	OC1M[2:0]	输出比较1模式 （Output Compare 1 mode） 该3位定义了输出参考信号OC1REF的动作，而OC1REF决定了OC1、OC1N的值。 OC1REF 是高电平有效，而OC1、OC1N的有效电平取决于CC1P、CC1NP位。 000：冻结。输出比较寄存器TIMx_CCDAT1与计数器TIMx_CNT间的比较对OC1REF不起作用； 001 ： 匹配时设置通道1 为有效电平。 当计数器TIMx_CNT 的值与捕获/ 比较寄存器1 （TIMx_CCDAT1）相同时，强制OC1REF为高。 010 ： 匹配时设置通道1 为无效电平。 当计数器TIMx_CNT 的值与捕获/ 比较寄存器1 （TIMx_CCDAT1）相同时，强制OC1REF为低。 011：翻转。当TIMx_CCDAT1=TIMx_CNT时，翻转OC1REF的电平。 100：强制为无效电平。强制OC1REF为低。 101：强制为有效电平。强制OC1REF为高。 110：PWM模式1 – 在向上计数时，一旦TIMx_CNT<TIMx_CCDAT1时通道1为有效电平，否则为无效电平；在向下计数时，一旦TIMx_CNT>TIMx_CCDAT1时通道1为无效电平（OC1REF=0），否则为有效电平（OC1REF=1）。 111：PWM模式2 – 在向上计数时，一旦TIMx_CNT<TIMx_CCDAT1时通道1为无效电平，否则为有效电平；在向下计数时，一旦TIMx_CNT>TIMx_CCDAT1时通道1为有效电平，否则为无效电平。 *注1：一旦LOCK级别设为3（TIMx_BKDT寄存器中的LCKCFG位）并且CC1SEL=00（该通道配置成输出）则该位不能被修改。* *注2：在PWM模式1或PWM模式2中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到PWM模式时，OC1REF电平才改变。*
3	OC1PEN	输出比较1预装载使能 （Output Compare 1 preload enable） 0：禁止TIMx_CCDAT1寄存器的预装载功能，可随时写入TIMx_CCDAT1寄存器，并且新写入的数值立即起作用。 1：开启TIMx_CCDAT1寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIMx_CCDAT1的预装载值在更新事件到来时被加载至当前寄存器中。 *注1：一旦LOCK级别设为3（TIMx_BKDT寄存器中的LCKCFG位）并且CC1SEL=00（该通道配置成输出）则该位不能被修改。* *注2：仅在单脉冲模式下（TIMx_CTRL1寄存器的ONEPM=1），可以在未确认预装载寄存器情况下使用* *PWM模式，否则其动作不确定。*

位域	名称	描述
2	OC1FEN	输出比较1 快速使能 （Output Compare 1 fast enable） 该位用于加快CC输出对触发输入事件的响应。 0：根据计数器与CCDAT1的值，CC1正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活CC1输出的最小延时为5个时钟周期。 1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。 OCxFEN只在通道被配置成PWM1或PWM2模式时起作用。
1:0	CC1SEL[1:0]	捕获/比较1 选择。（Capture/Compare 1 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC1通道被配置为输出； 01：CC1通道被配置为输入，IC1映射在TI1上； 10：CC1通道被配置为输入，IC1映射在TI2上； 11：CC1通道被配置为输入，IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC1SEL仅在通道关闭时（TIMx_CCEN寄存器的CC1EN=0）才是可写的。*

输入捕获模式：

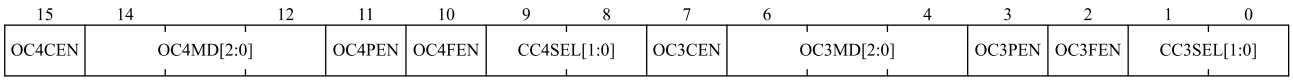


位域	名称	描述
15:12	IC2F[3:0]	输入捕获2滤波器 （Input capture 2 filter）
11:10	IC2PSC[1:0]	输入/捕获2预分频器 （Input capture 2 prescaler）
9:8	CC2SEL[1:0]	捕获/比较2选择 （Capture/Compare 2 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。*
7:4	IC1F[3:0]	输入捕获1滤波器 （Input capture 1 filter） 这几位定义了TI1输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到N个事件后会产生一个输出的跳变： 0000：无滤波器，以f_{DTS}采样 1000：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=6 0001：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=2 1001：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=8 0010：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=4 1010：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=5 0011：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=8 1011：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=6 0100：采样频率$f_{SAMPLING}=f_{DTS}/2$，N=6 1100：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=8 0101：采样频率$f_{SAMPLING}=f_{DTS}/2$，N=8 1101：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=5

位域	名称	描述
		0110：采样频率$f_{SAMPLING}=f_{DTS}/4$，N=6　1110：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=6 0111：采样频率$f_{SAMPLING}=f_{DTS}/4$，N=8　1111：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=8
3:2	IC1PSC[1:0]	输入/捕获1预分频器　（Input capture 1 prescaler） 这2位定义了CC1输入（IC1）的预分频系数。 一旦CC1EN=0（TIMx_CCEN寄存器中），则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获； 01：每2个事件触发一次捕获； 10：每4个事件触发一次捕获； 11：每8个事件触发一次捕获。
1:0	CC1SEL[1:0]	捕获/比较1选择　（Capture/Compare 1 Selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC1通道被配置为输出； 01：CC1通道被配置为输入，IC1映射在TI1上； 10：CC1通道被配置为输入，IC1映射在TI2上； 11：CC1通道被配置为输入，IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC1SEL仅在通道关闭时（TIMx_CCEN寄存器的CC1EN=0）才是可写的。*

11.4.9 TIM1 和 TIM8 捕获/比较模式寄存器 2（TIMx_CCMOD2）

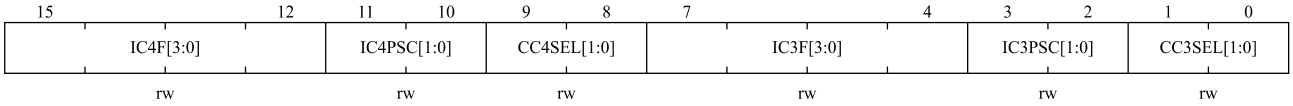
偏移地址：0x1C
复位值：0x0000
参看以上 CCMOD1 寄存器的描述
输出比较模式：



位域	名称	描述
15	OC4CEN	输出比较4清0使能　（Output compare 4 clear enable）
14:12	OC4MD[2:0]	输出比较4模式　（Output compare 4 mode）
11	OC4PEN	输出比较4预装载使能　（Output compare 4 preload enable）
10	OC4FEN	输出比较4快速使能　（Output compare 4 fast enable）

位域	名称	描述
9:8	CC4SEL[1:0]	捕获/比较4选择 （Capture/Compare 4 selection） 该2位定义通道的方向（输入/输出），及输入脚的选择: 00：CC4通道被配置为输出; 01：CC4通道被配置为输入，IC4映射在TI4上; 10：CC4通道被配置为输入，IC4映射在TI3上; 11：CC4通道被配置为输入，IC4映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC4SEL仅在通道关闭时（TIMx_CCEN寄存器的CC4EN=0）才是可写的。*
7	OC3CEN	输出比较3清0使能 （Output compare 3 clear enable）
6:4	OC3MD[2:0]	输出比较3模式 （Output compare 3 mode）
3	OC3PEN	输出比较3预装载使能 （Output compare 3 preload enable）
2	OC3FEN	输出比较3快速使能 （Output compare 3 fast enable）
1:0	CC3SEL[1:0]	捕获/比较3选择 （Capture/Compare 3 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择: 00：CC3通道被配置为输出; 01：CC3通道被配置为输入，IC3映射在TI3上; 10：CC3通道被配置为输入，IC3映射在TI4上; 11：CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC3SEL仅在通道关闭时（TIMx_CCEN寄存器的CC3EN=0）才是可写的。*

输入捕获模式:

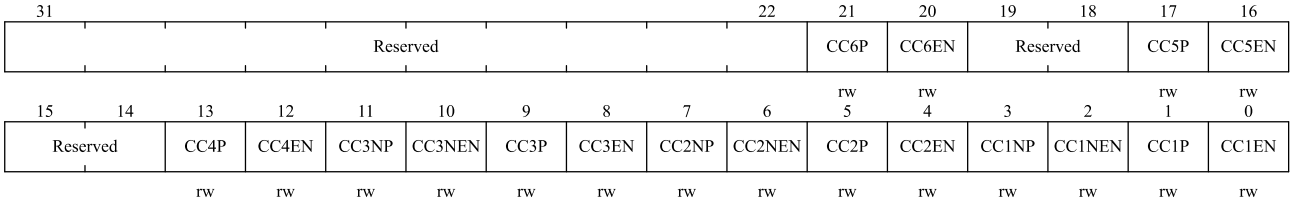


位域	名称	描述
15:12	IC4F[3:0]	输入捕获4滤波器 (Input capture 4 filter)
11:10	IC4PSC[1:0]	输入/捕获4预分频器 (Input capture 4 prescaler)
9:8	CC4SEL[1:0]	捕获/比较4选择 (Capture/Compare 4 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择: 00：CC4通道被配置为输出; 01：CC4通道被配置为输入，IC4映射在TI4上; 10：CC4通道被配置为输入，IC4映射在TI3上; 11：CC4通道被配置为输入，IC4映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 *注：CC4SEL仅在通道关闭时(TIMx_CCEN寄存器的CC4EN=0)才是可写的。*
7:4	IC3F[3:0]	输入捕获3滤波器 (Input capture 3 filter)
3:2	IC3PSC[1:0]	输入/捕获3预分频器 (Input capture 3 prescaler)

位域	名称	描述
1:0	CC3SEL[1:0]	捕获/比较3选择（Capture/compare 3 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC3通道被配置为输出； 01：CC3通道被配置为输入，IC3映射在TI3上； 10：CC3通道被配置为输入，IC3映射在TI4上； 11：CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发器输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 *注：CC3SEL仅在通道关闭时(TIMx_CCEN寄存器的CC3EN=0)才是可写的。*

11.4.10 TIM1 和 TIM8 捕获/比较使能寄存器（TIMx_CCEN）

偏移地址：0x20
复位值：0x0000



位域	名称	描述
31:22	Reserved	保留，硬件强制为0。
21	CC6P	输入/捕获6输出极性 （Capture/Compare 6 output polarity） 参考CC1P的描述。
20	CC6EN	输入/捕获6输出使能 （Capture/Compare 6 output enable） 参考CC1EN 的描述
19:18	Reserved	保留，硬件强制为0。
17	CC5P	输入/捕获4输出极性 （Capture/Compare 5 output polarity） 参考CC1P的描述。
16	CC5EN	输入/捕获4输出使能 （Capture/Compare 5 output enable） 参考CC1EN 的描述
15:14	Reserved	保留，硬件强制为0。
13	CC4P	输入/捕获4输出极性 （Capture/Compare 4 output polarity） 参考CC1P的描述。
12	CC4EN	输入/捕获4输出使能 （Capture/Compare 4 output enable） 参考CC1EN 的描述。
11	CC3NP	输入/捕获3互补输出极性 （Capture/Compare 3 complementary output polarity） 参考CC1NP的描述。
10	CC3NEN	输入/捕获3互补输出使能 （Capture/Compare 3 complementary output enable） 参考CC1NEN的描述。

位域	名称	描述
9	CC3P	输入/捕获3输出极性 （Capture/Compare 3 output polarity） 参考CC1P的描述。
8	CC3E	输入/捕获3输出使能 （Capture/Compare 3 output enable） 参考CC1E 的描述。
7	CC2NP	输入/捕获2互补输出极性 （Capture/Compare 2 complementary output polarity） 参考CC1NP的描述。
6	CC2NEN	输入/捕获2互补输出使能 （Capture/Compare 2 complementary output enable） 参考CC1NEN的描述。
5	CC2P	输入/捕获2输出极性 （Capture/Compare 2 output polarity） 参考CC1P的描述。
4	CC2EN	输入/捕获2输出使能 （Capture/Compare 2 output enable） 参考CC1EN的描述。
3	CC1NP	输入/捕获1互补输出极性 （Capture/Compare 1 complementary output polarity） 0：OC1N高电平有效； 1：OC1N低电平有效。 *注：一旦LOCK级别（TIMx_BKDT寄存器中的LCKCFG位）设为3或2且CC1SEL=00（通道配置为输出） 则该位不能被修改。*
2	CC1NEN	输入/捕获1互补输出使能 （Capture/Compare 1 complementary output enable） 0：关闭－ OC1N禁止输出，因此OC1N的电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1EN位的值。 1：开启－ OC1N信号输出到对应的输出引脚，其输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1EN位的值。
1	CC1P	输入/捕获1输出极性 （Capture/Compare 1 output polarity） **CC1通道配置为输出**: 0：OC1高电平有效； 1：OC1低电平有效。 **CC1通道配置为输入**: 该位选择是IC1还是IC1的反相信号作为触发或捕获信号。 0：不反相：捕获发生在IC1的上升沿；当用作外部触发器时，IC1不反相。 1：反相：捕获发生在IC1的下降沿；当用作外部触发器时，IC1反相。 *注：一旦LOCK级别（TIMx_BKDT寄存器中的LCKCFG位）设为3或2，则该位不能被修改。*
0	CC1EN	输入/捕获1输出使能 （Capture/Compare 1 output enable） **CC1通道配置为输出：** 0： 关闭－ OC1禁止输出，因此OC1的输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。 1： 开启－ OC1信号输出到对应的输出引脚，其输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。 **CC1通道配置为输入：** 该位决定了计数器的值是否能捕获入TIMx_CCDAT1寄存器。 0：捕获禁止； 1：捕获使能。

表 11-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位

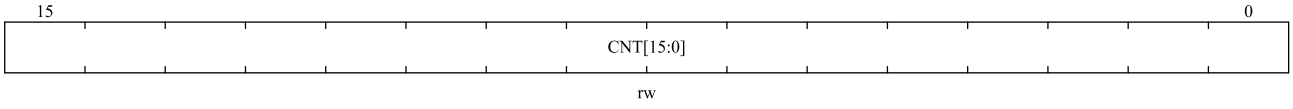
控制位					输出状态(1)	
MOEN	OSSI	OSSR	CCxEN	CCxNEN	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	输出禁止（与定时器断开） OCx=0，OCx_EN=0	输出禁止（与定时器断开） OCxN=0，OCxN_EN=0
		0	0	1	输出禁止（与定时器断开） OCx=0，OCx_EN=0	OCxREF + 极性， OCxN= OCxREF xor CCxNP，OCxN_EN=1
		0	1	0	OCxREF + 极性， OCx= OCxREF xor CCxP，OCx_EN=1	输出禁止（与定时器断开） OCxN=0，OCxN_EN=0
		0	1	1	OCxREF + 极性 + 死区，OCx_EN=1	OCxREF反相 + 极性 + 死区， OCxN_EN=1
		1	0	0	输出禁止（与定时器断开） OCx=CCxP，OCx_EN=0	输出禁止（与定时器断开） OCxN=CCxNP，OCxN_EN=0
		1	0	1	关闭状态（输出使能且为无效电平） OCx=CCxP，OCx_EN=1	OCxREF + 极性， OCxN= OCxREF xor CCxNP，CxN_EN=1
		1	1	0	OCxREF + 极性， OCx= OCxREF xor CCxP， OCx_EN=1	关闭状态（输出使能且为无效电平） OCxN=CCxNP，OCxN_EN=1
		1	1	1	OCxREF + 极性 + 死区， OCx_EN=1	OCxREF反相 + 极性 + 死区， OCxN_EN=1
0	0	X	0	0	输出禁止（与定时器断开）	
	0		0	1	异步： OCx=CCxP， OCx_EN=0， OCxN=CCxNP，OCxN_EN=0； 若时钟存在：假设OIx与OIxN并不都对应OCx和OCxN的有效电平。 经过一个死区时间后OCx=OIx，OCxN=OIxN	
	0		1	0		
	0		1	1		
	1		0	0	关闭状态（输出使能且为无效电平）	
	1		0	1	异步： OCx=CCxP， OCx_EN=1， OCxN=CCxNP，OCxN_EN=1； 若时钟存在：假设OIx与OIxN并不都对应OCx和OCxN的有效电平。 经过一个死区时间后OCx=OIx，OCxN=OIxN，	
	1		1	0		
	1		1	1		

1. 如果一个通道的 2 个输出都没有使用（CCxEN = CCxNEN = 0），那么 OIx，OIxN，CCxP 和 CCxNP 都必须清零。

注：引脚连接到互补的 OCx 和 OCxN 通道的外部 I/O 引脚的状态，取决于 OCx 和 OCxN 通道状态和 GPIO 以及 AFIO 寄存器。

11.4.11 TIM1 和 TIM8 计数器（TIMx_CNT）

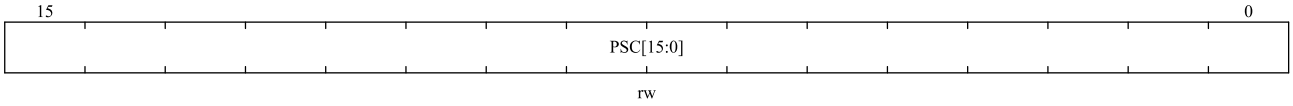
偏移地址：0x24
复位值：0x0000



位域	名称	描述
15:0	CNT[15:0]	计数器的值 （Counter value）

11.4.12 TIM1 和 TIM8 预分频器（TIMx_PSC）

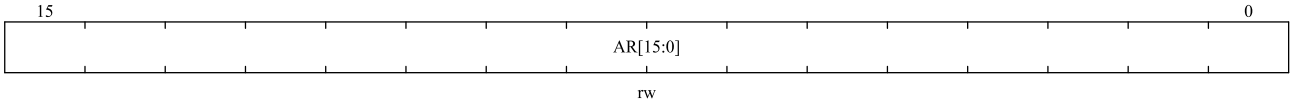
偏移地址：0x28
复位值：0x0000



位域	名称	描述
15:0	PSC[15:0]	预分频器的值 （Prescaler value） 计数器的时钟频率（CK_CNT）等于fCK_PSC/(PSC[15:0]+1)。 PSC包含了每次当更新事件产生时，装入当前预分频器寄存器的值；更新事件包括计数器被TIM_EVTGEN的UDGN位清'0'或被工作在复位模式的从控制器清'0'。

11.4.13 TIM1 和 TIM8 自动重装载寄存器（TIMx_AR）

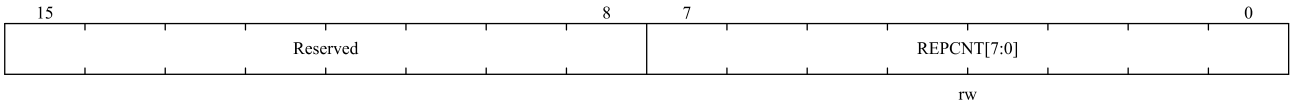
偏移地址:0x2C
复位值:0x0000



位域	名称	描述
15:0	AR[15:0]	自动重装载的值 （Auto-reload value） AR包含了将要装载入实际的自动重装载寄存器的值。详细参考11.3.1节：有关AR的更新和动作。 当自动重装载的值为空时，计数器不工作。

11.4.14 TIM1 和 TIM8 重复计数寄存器（TIMx_REPCNT）

偏移地址：0x30
复位值：0x0000

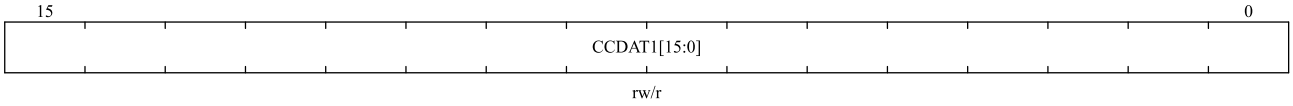


位域	名称	描述
15:8	Reserved	保留，硬件强制为0。

位域	名称	描述
7:0	REPCNT[7:0]	重复计数器的值 （Repetition counter value） 开启了预装载功能后，这些位允许用户设置比较寄存器的更新速率（即周期性地从预装载寄存器传输到当前寄存器）；如果允许产生更新中断，则会同时影响产生更新中断的速率。 每次向下计数器CNT达到0，会产生一个更新事件并且计数器CNT重新从REPCNT值开始计数。由于CNT只有在周期更新事件UDGN发生时才重载REPCNT值，因此对TIMx_REPCNT寄存器写入的新值只在下次周期更新事件发生时才起作用。 这意味着在PWM模式中，（REPCNT+1）对应着： – 在边沿对齐模式下，PWM周期的数目； – 在中心对称模式下，PWM半周期的数目；

11.4.15 TIM1 和 TIM8 捕获/比较寄存器 1（TIMx_CCDAT1）

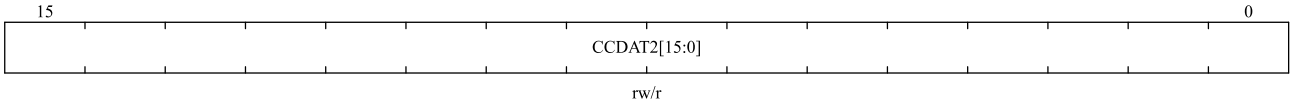
偏移地址：0x34
复位值：0x0000



位域	名称	描述
15:0	CCDAT1[15:0]	捕获/比较通道1的值 （Capture/Compare 1 value） **若CC1通道配置为输出：** CCDAT1包含了装入当前捕获/比较1寄存器的值（预装载值）。 如果在TIMx_CCMOD1寄存器（OC1PEN位）中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较1寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并在OC1端口上产生输出信号。 **若CC1通道配置为输入：** CCDAT1包含了由上一次输入捕获1事件（IC1）传输的计数器值。

11.4.16 TIM1 和 TIM8 捕获/比较寄存器 2（TIMx_CCDAT2）

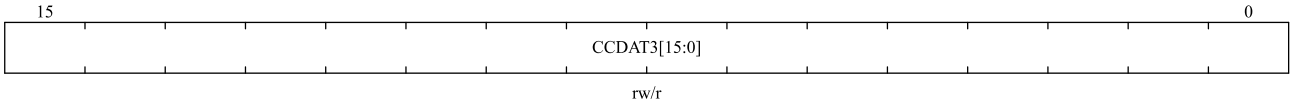
偏移地址：0x38
复位值：0x0000



位域	名称	描述
15:0	CCDAT2[15:0]	捕获/比较通道2的值 （Capture/Compare 2 value） **若CC2通道配置为输出：** CCDAT2包含了装入当前捕获/比较2寄存器的值（预装载值）。 如果在TIMx_CCMOD1寄存器（OC2PEN位）中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较2寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并在OC2端口上产生输出信号。 **若CC2通道配置为输入：** CCDAT2包含了由上一次输入捕获2事件（IC2）传输的计数器值。

11.4.17 TIM1 和 TIM8 捕获/比较寄存器 3（TIMx_CCDAT3）

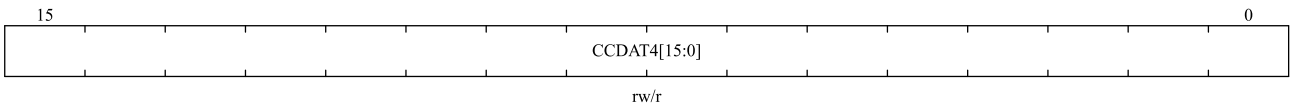
偏移地址：0x3C
复位值：0x0000



位域	名称	描述
15:0	CCDAT3[15:0]	捕获/比较通道3的值 （Capture/Compare 3 value） **若CC3通道配置为输出：** CCDAT3包含了装入当前捕获/比较3寄存器的值（预装载值）。 如果在TIMx_CCMOD2寄存器（OC3PEN位）中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较3寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并在OC3端口上产生输出信号。**若CC3通道配置为输入：** CCDAT3包含了由上一次输入捕获3事件（IC3）传输的计数器值。

11.4.18 TIM1 和 TIM8 捕获/比较寄存器 4（TIMx_CCDAT4）

偏移地址：0x40
复位值：0x0000



位域	名称	描述

15:0	CCDAT4[15:0]	捕获/比较通道4的值 （Capture/Compare 4 value） **若CC4通道配置为输出：** CCDAT4包含了装入当前捕获/比较4寄存器的值（预装载值）。 如果在TIMx_CCMOD2寄存器（OC4PEN位）中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较4寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并在OC4端口上产生输出信号。 **若CC4通道配置为输入：** CCDAT4包含了由上一次输入捕获4事件（IC4）传输的计数器值。

11.4.19 TIM1 和 TIM8 刹车和死区寄存器（TIMx_BKDT）

偏移地址：0x44

复位值：0x0000

15	14	13	12	11	10	9	8	7	0
MOEN	AOEN	BKP	BKEN	OSSR	OSSI	LCKCFG[1:0]		DTGN[7:0]	
rw	rw	rw	rw	rw	rw	rw		rw	

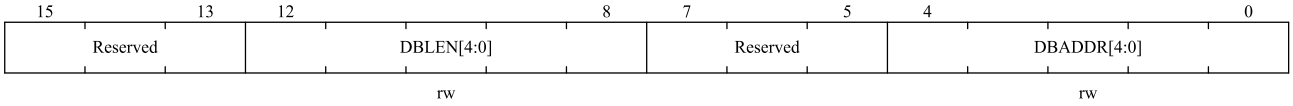
注释： 根据锁定设置，AOEN、BKP、BKEN、OSSI、OSSR 和 DTGN[7:0]位均可被写保护，有必要在第一次写入 TIMx_BKDT 寄存器时对它们进行配置。

位域	名称	描述
15	MOEN	主输出使能 （Main output enable） 一旦刹车输入有效，该位被硬件异步清'0'。根据AOEN位的设置值，该位可以由软件清'0'或被自动置1。它仅对配置为输出的通道有效。 0：禁止OC和OCN输出或强制为空闲状态； 1：如果设置了相应的使能位（TIMx_CCEN寄存器的CCxEN、CCxNEN位），则开启OC和OCN输出。有关OC/OCN使能的细节，参见11.4.10节，TIM1和TIM8捕获/比较使能寄存器（TIMx_CCEN）。
14	AOEN	自动输出使能 （Automatic output enable） 0：MOEN只能被软件置'1'； 1：MOEN能被软件置'1'或在下一个更新事件被自动置'1'（如果刹车输入无效）。 *注：一旦LOCK级别（TIMx_BKDT寄存器中的LCKCFG位）设为'1'，则该位不能被修改。*
13	BKP	刹车输入极性 （Break polarity） 0：刹车输入低电平有效； 1：刹车输入高电平有效。 *注：一旦LOCK级别（TIMx_BKDT寄存器中的LCKCFG位）设为'1'，则该位不能被修改。* *注：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。*
12	BKEN	刹车功能使能 （Break enable） 0：禁止刹车输入（BRK及CCS时钟失效事件）； 1：开启刹车输入（BRK及CCS时钟失效事件）。 *注：当设置了LOCK级别1时（TIMx_BKDT寄存器中的LCKCFG位），该位不能被修改。* *注：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。*

位域	名称	描述
11	OSSR	运行模式下“关闭状态”选择 （Off-state selection for Run mode） 该位用于当MOEN=1且通道为互补输出时。没有互补输出的定时器中不存在OSSR位。 参考OC/OCN使能的详细说明（11.4.10节，TIM1和TIM8捕获/比较使能寄存器（TIMx_CCEN））。 0：当定时器不工作时，禁止OC/OCN输出（OC/OCN使能输出信号=0）； 1：当定时器不工作时，一旦CCxEN=1或CCxNEN=1，首先开启OC/OCN并输出无效电平，然后置OC/OCN使能输出信号=1。 *注：一旦LOCK级别（TIMx_BKDT寄存器中的LCKCFG位）设为2，则该位不能被修改。*
10	OSSI	空闲模式下“关闭状态”选择 （Off-state selection for Idle mode） 该位用于当MOEN=0且通道设为输出时。 参考OC/OCN使能的详细说明（11.4.10节，TIM1和TIM8捕获/比较使能寄存器（TIMx_CCEN））。 0：当定时器不工作时，禁止OC/OCN输出（OC/OCN使能输出信号=0）； 1 ：当定时器不工作时，一旦CCxEN=1 或CCxNEN=1，OC/OCN首先输出其空闲电平，然后OC/OCN使能输出信号=1。 *注：一旦LOCK级别（TIMx_BKDT寄存器中的LCKCFG位）设为2，则该位不能被修改。*
9:8	LCKCFG[1:0]	锁定设置 （Lock configuration） 该位为防止软件错误而提供写保护。 00：锁定关闭，寄存器无写保护； 01：锁定级别1，不能写入TIMx_BKDT寄存器的DTGN、BKEN、BKP、AOEN位和TIMx_CTRL2寄存器的OIx/OIxN位； 10：锁定级别2，不能写入锁定级别1中的各位，也不能写入CC极性位（一旦相关通道通过CCxSEL位设为输出，CC极性位是TIMx_CCEN寄存器的CCxP/CCNxP位）以及OSSR/OSSI位； 11：锁定级别3，不能写入锁定级别2中的各位，也不能写入CC控制位（一旦相关通道通过CCxSEL位设为输出，CC控制位是TIMx_CCMODx寄存器的OCxMD/OCxPEN位）； *注：在系统复位后，只能写一次LCKCFG位，一旦写入TIMx_BKDT寄存器，则其内容冻结直至复位。*
7:0	UTG[7:0]	死区发生器设置 （Dead-time generator setup） 这些位定义了插入互补输出之间的死区持续时间。假设DT表示其持续时间： DTGN[7:5]=0xx => DT=DTGN[7:0] × T_{dtgn}，T_{dtgn} = T_{DTS}； DTGN[7:5]=10x => DT=(64+DTGN[5:0]) × T_{dtgn}，T_{dtgn} = 2 × T_{DTS}； DTGN[7:5]=110 => DT=(32+DTGN[4:0]) × T_{dtgn}，T_{dtgn} = 8 × T_{DTS}； DTGN[7:5]=111 => DT=(32+DTGN[4:0])× T_{dtgn}，T_{dtgn} = 16 × T_{DTS}； 例：若T_{DTS} = 125ns(8MHZ)，可能的死区时间为： 0到15875ns，若步长时间为125ns； 16us到31750ns，若步长时间为250ns； 32us到63us，若步长时间为1us； 64us到126us，若步长时间为2us； *注：一旦LOCK级别（TIMx_BKDT寄存器中的LCKCFG位）设为1、2或3，则不能修改这些位。*

11.4.20 TIM1 和 TIM8 DMA 控制寄存器（TIMx_DCTRL）

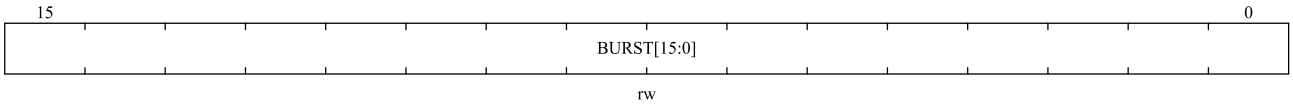
偏移地址：0x48
复位值：0x0000



位域	名称	描述
15:13	Reserved	保留，硬件强制为0。
12:8	DBLEN[4:0]	DMA连续传送长度 （DMA burst length） 这些位定义了DMA在连续模式下的传送长度（当对TIMx_DADDR寄存器进行读或写时，定时器则进行一次连续传送），即：定义传输的次数，传输可以是半字（双字节）或字节： 00000：1次传输 00001：2次传输 00010：3次传输 .... 10001：18次传输 例：我们考虑这样的传输：DBLEN=7，DBADDR=TIM2_CTRL1 ■ 如果DBLEN=7，DBADDR=TIM2_CTRL1表示待传输数据的地址，那么传输的地址由下式给出： (TIMx_CTRL1的地址) + DBADDR + (DMA索引)，其中 DMA索引 = DBLEN 其中(TIMx_CTRL1的地址) + DBADDR再加上7，给出了将要写入或者读出数据的地址，这样数据的传输将发生在从地址(TIMx_CTRL1的地址) + DBADDR开始的7个寄存器。 根据DMA数据长度的设置，可能发生以下情况： ■ 如果设置数据为半字（16位），那么数据就会传输给全部7个寄存器。 ■ 如果设置数据为字节，数据仍然会传输给全部7个寄存器：第一个寄存器包含第一个MSB字节，第二个寄存器包含第一个LSB字节，以此类推。因此对于定时器，用户必须指定由DMA传输的数据宽度。
7:5	Reserved	保留，硬件强制为0。
4:0	DBADDR[4:0]	DMA基地址 （DMA base address） 这些位定义了DMA在连续模式下的基地址（当对TIMx_DADDR寄存器进行读或写时），DBADDR定义为从TIMx_CTRL1寄存器所在地址开始的偏移量： 00000：TIMx_CTRL1， 00001：TIMx_CTRL2， 00010：TIMx_SMCTRL， .....

11.4.21 TIM1 和 TIM8 连续模式的 DMA 地址（TIMx_DADDR）

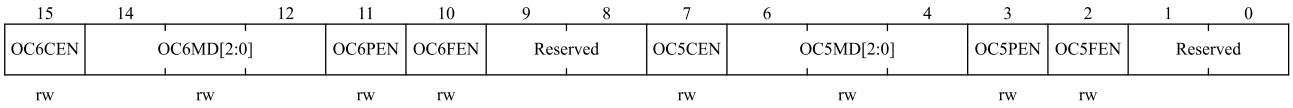
偏移地址：0x4C
复位值：0x0000



位域	名称	描述
15:0	BURST[15:0]	DMA连续传送寄存器 （DMA register for burst accesses） 对TIMx_DADDR寄存器的读或写会导致对以下地址所在寄存器的存取操作： TIMx_CTRL1地址 + DBADDR + DMA索引，其中： “TIMx_CTRL1地址”是控制寄存器1（TIMx_CTRL1）所在的地址； “DBADDR”是TIMx_DCTRL寄存器中定义的基地址； “DMA索引”是由DMA自动控制的偏移量，它取决于TIMx_DCTRL寄存器中定义的DBLEN。

11.4.22 TIM1 和 TIM8 捕获/比较寄存器（TIMx_CCMOD3）

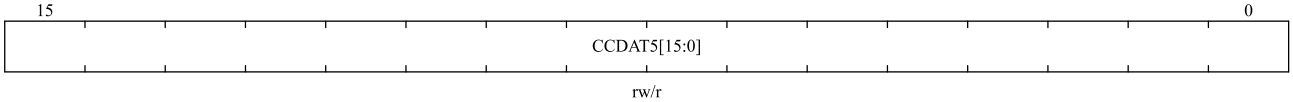
偏移地址：0x54
复位值：0x0000



位域	名称	描述
15	OC6CEN	输出比较6清0使能 （Output compare 6 clear enable）
14:12	OC6MD[2:0]	输出比较6模式 （Output compare 6 mode）
11	OC6PEN	输出比较6预装载使能 （Output compare 6 preload enable）
10	OC6FEN	输出比较6快速使能 （Output compare 6 fast enable）
9:8	Reserved	保留，硬件强制为0。
7	OC5CEN	输出比较5清0使能 （Output compare 3 clear enable）
6:4	OC5MD[2:0]	输出比较5模式 （Output compare 3 mode）
3	OC5PEN	输出比较5预装载使能 （Output compare 5 preload enable）
2	OC5FEN	输出比较5快速使能 （Output compare 5 fast enable）
1:0	Reserved	保留，硬件强制为0。

11.4.23 TIM1 和 TIM8 捕获/比较寄存器 5（TIMx_CCDAT5）

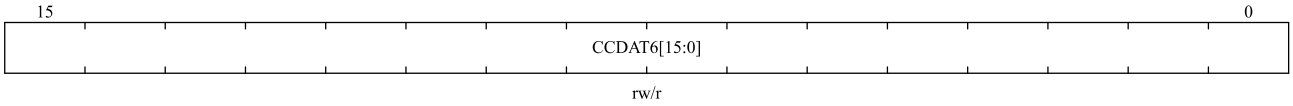
偏移地址：0x58
复位值：0x0000



位域	名称	描述
15:0	CCDAT5[15:0]	捕获/比较通道4的值 （Capture/Compare 5 value） **若CC5通道配置为输出：** CCDAT5包含了装入当前捕获/比较4寄存器的值（预装载值）。 如果在TIMx_CCMOD3寄存器（OC5PEN位）中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较5寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并产生内部CC5信号。 CC5信号用于比较器消隐输出。 **若CC5通道配置为输入：** 不能作为输入捕获。

11.4.24 TIM1 和 TIM8 捕获/比较寄存器 6（TIMx_CCDAT6）

偏移地址：0x5C
复位值：0x0000



位域	名称	描述
15:0	CCDAT6[15:0]	捕获/比较通道4的值 （Capture/Compare 6 value） **若CC6通道配置为输出：** CCDAT6包含了装入当前捕获/比较4寄存器的值（预装载值）。 如果在TIMx_CCMOD3寄存器（OC6PEN位）中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较6寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并产生内部CC6信号。 TIM1_CC6用OPAMP1和OPAMP2的输入通道自动切换。 TIM8_CC6用OPAMP3和OPAMP4的输入通道自动切换。 **若CC6通道配置为输入：** 不能作为输入捕获。

12 通用定时器（TIMx）

12.1 TIMx 简介

通用定时器是由一个 16 位的自动装载计数器和一个可编程预分频计数器组成。它适用于多种场合，包括测量输入信号的脉冲长度（输入捕获）或者产生输出波形（输出比较和 PWM）以及对输入信号进行计数。

结合定时器预分频器和 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

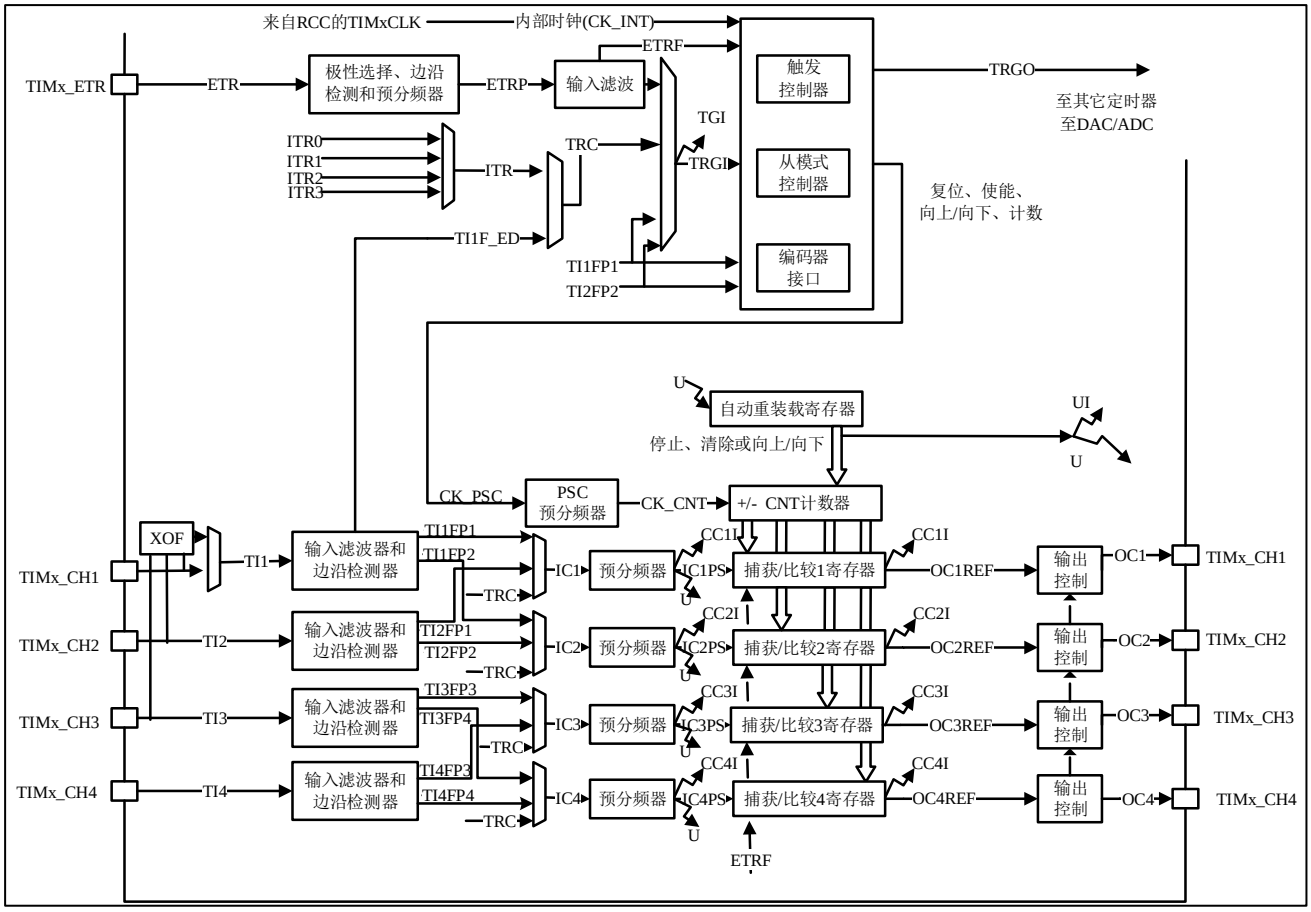
每个定时器都是完全独立的，没有互相共享任何资源，完全独立，可以实现异步或同步操作，具体参见 12.3.15 章节。

12.2 TIMx 主要功能

通用 TIMx （TIM2、TIM3、TIM4 和 TIM5）定时器功能包括:

- 16 位向上、向下、向上/向下自动装载计数器
- 16 位可编程（可以实时修改）预分频器，计数器时钟频率的分频系数为 1～65536 之间的任意数值
- 4 个独立通道:
  - 输入捕获
  - 输出比较
  - PWM 生成（边缘或中间对齐模式）
  - 单脉冲模式输出
- 使用外部信号控制定时器和定时器互连的同步电路
- 如下事件发生时产生中断/DMA:
  - 更新：计数器向上溢出/向下溢出，计数器初始化（通过软件或者内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或者由内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持针对定位的增量（正交）编码器和霍尔传感器电路
- 支持捕获 IO 信号或比较器输出信号
- 触发输入作为外部时钟或者按周期的电流管理

图 12-1 通用定时器框图



事件

中断和 DMA 输出

捕获通道 1/2/3/4 输入可以来自 IOM 或比较器输出（TIM5 的 CH4 除外）

清除 OCxRef 可以来自 ETR 或比较器输出（TIM2,TIM4 有此功能）

ITM5 无外部 ETR 功能

12.3 TIMx 功能描述

12.3.1 时基单元

通用定时器的主要部分是一个 16 位计数器和与其相关的自动装载寄存器。这个计数器可以向上计数、向下计数或者向上向下双向计数。此计数器时钟由预分频器分频得到。

计数器、自动装载寄存器和预分频器寄存器可以由软件读写，在计数器运行时仍可以读写。时基单元包含：

- 预分频器寄存器 （TIMx_PSC）
- 计数器寄存器（TIMx_CNT）

- 自动装载寄存器 （TIMx_AR）

自动装载寄存器具有预先装载功能，读写自动重装载寄存器将访问预装载寄存器。根据在 TIMx_CTRL1 寄存器中的自动装载预装载使能位（ARPEN）的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。以下条件产生更新事件：

- 当计数器达到溢出条件（向下计数时的下溢条件）并当 TIMx_CTRL1 寄存器中的 UPDIS 位等于'0'时，产生更新事件。
- 软件产生

预分频器的时钟输出 CK_CNT 驱动计数器，仅当设置了计数器 TIMx_CTRL1 寄存器中的计数器使能位（CNTEN）时，CK_CNT 才有效。（有关计数器使能的细节，请参见控制器的从模式描述）。

注意，在设置了 TIMx_CTRL 寄存器的 CNTEN 位的一个时钟周期后，计数器开始计数。

12.3.1.1 预分频器描述

预分频器（TIMx_PSC 寄存器）由 16 位计数器构成。可以将输入时钟频率按 1 到 65536 之间的任意值分频。因为这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 12-2 和图 12-3 给出了在预分频器运行时，更改计数器参数的示例。

图 12-2 当预分频器的参数从 1 变到 2 时，计数器的时序图

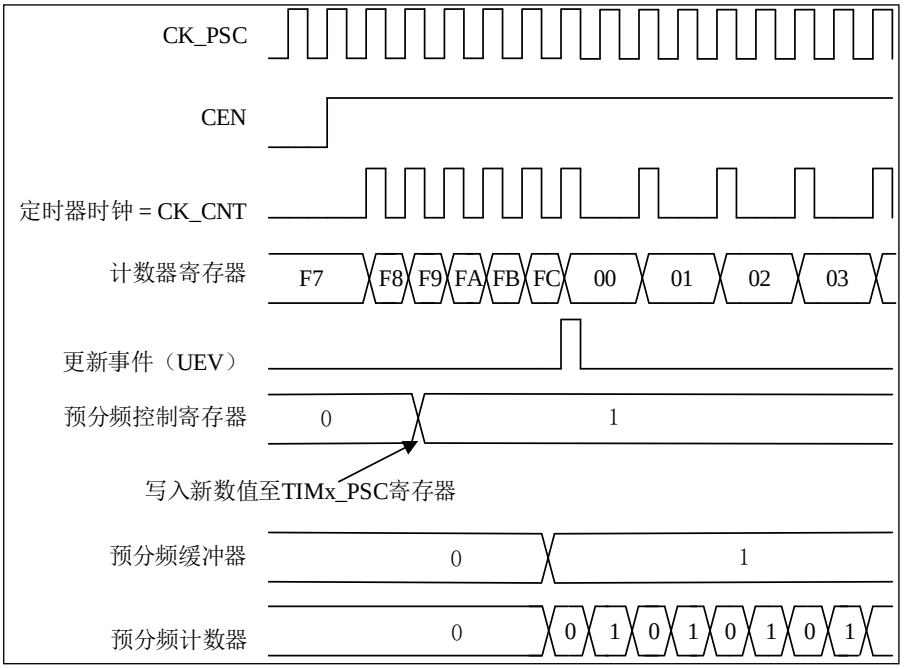
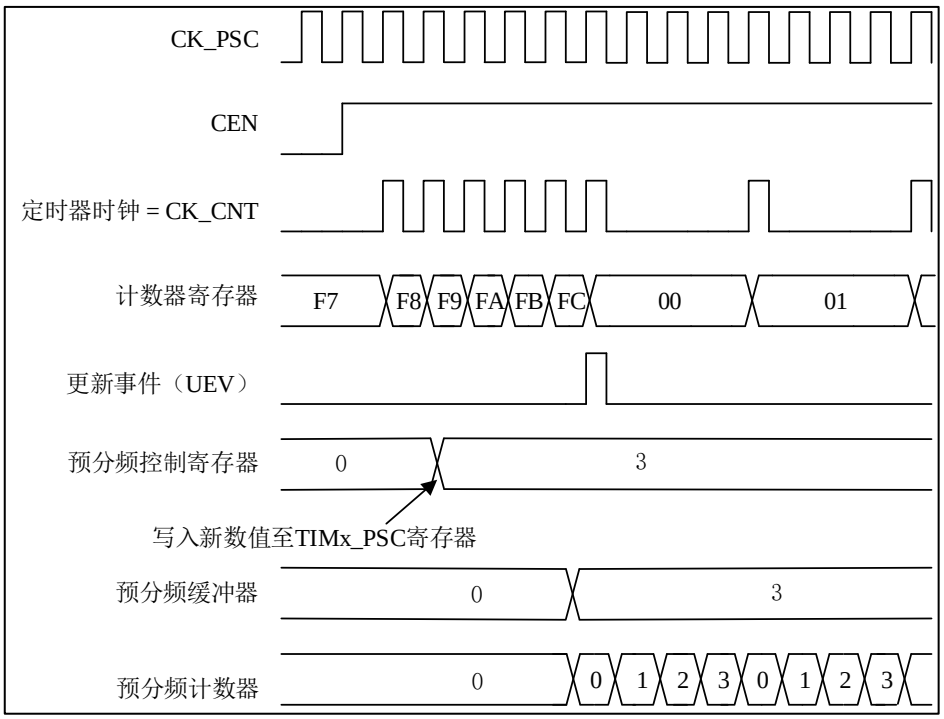


图 12-3 当预分频器的参数从 1 变到 4 时，计数器的时序图



12. 3. 2 计数器模式

12. 3. 2. 1 向上计数模式

使用向上计数模式，计数器从 0 计数到自动加载值（TIMx_AR 计数器的内容），然后重新从 0 开始计数并且产生一个计数器溢出事件。

每次计数器溢出时可以产生更新事件，在 TIMx_EVTGEN 寄存器中（通过软件方式或者使用从模式控制器）设置 UDGN 位也同样可以产生一个更新事件。

为了避免在向预装载寄存器中写入新值时更新影子寄存器，可以设置 TIMx_CTRL1 寄存器中的 UPDIS 位，可以禁止更新事件。在 UPDIS 位被清'0'之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清'0'，同时预分频器的计数也被清 0（但预分频系数不变）。此外，如果设置了 TIMx_CTRL1 寄存器中的 UPRS 位（选择更新请求），设置 UDGN 位将产生一个更新事件 UEV，但硬件不设置 UDITF 标志（即不产生中断或 DMA 请求）；这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

更新事件发生后，所有的寄存器都被更新，硬件同时（依据 UPRS 位）设置更新标志位（TIMx_STS 寄存器中的 UDITF 位）。

- 预分频器的缓冲区被置入预装载寄存器的值（TIMx_PSC 寄存器的内容）。
- 自动装载影子寄存器被重新置入预装载寄存器的值（TIMx_AR）。

下图给出一些示例，当 TIMx_AR=0x36 时计数器在不同时钟频率下的动作。

图 12-4 计数器时序图，内部时钟分频因子为 1

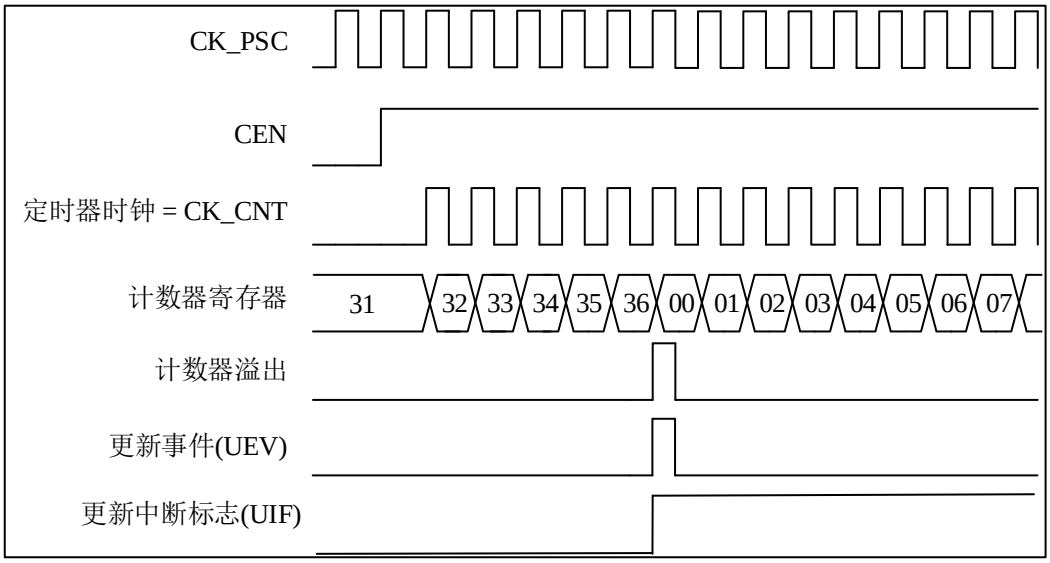


图 12-5 计数器时序图，内部时钟分频因子为 2

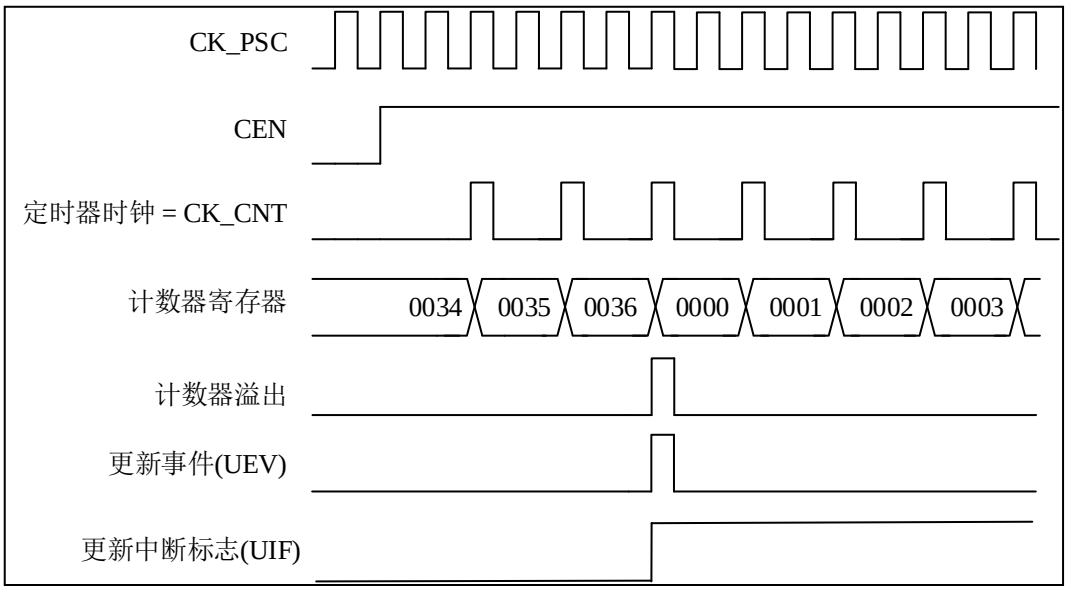


图 12-6 计数器时序图，内部时钟分频因子为 4

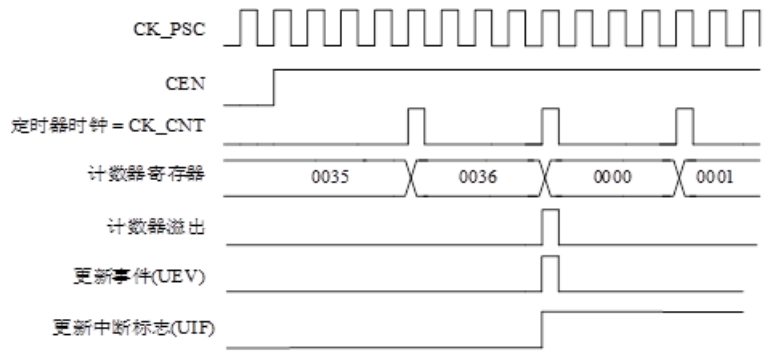


图 12-7 计数器时序图，内部时钟分频因子为 N

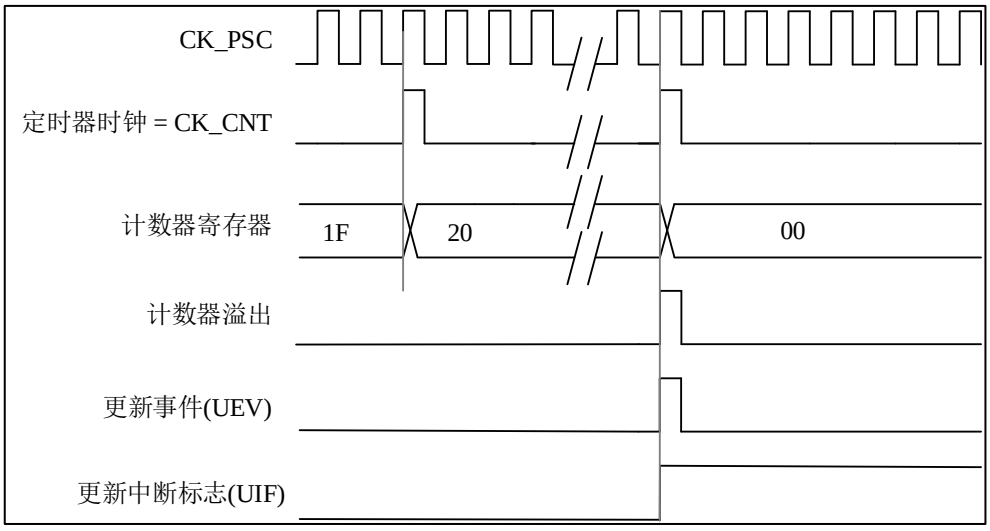


图 12-8 计数器时序图，当 ARPEN=0 时的更新事件（TIMx_AR 没有预装入）

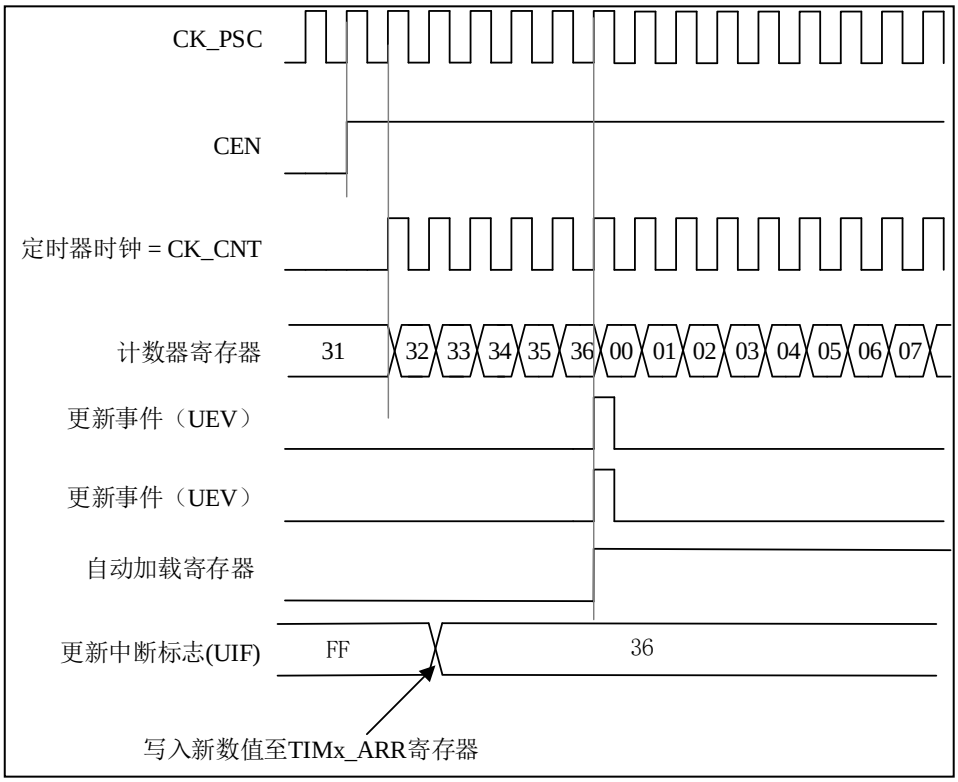
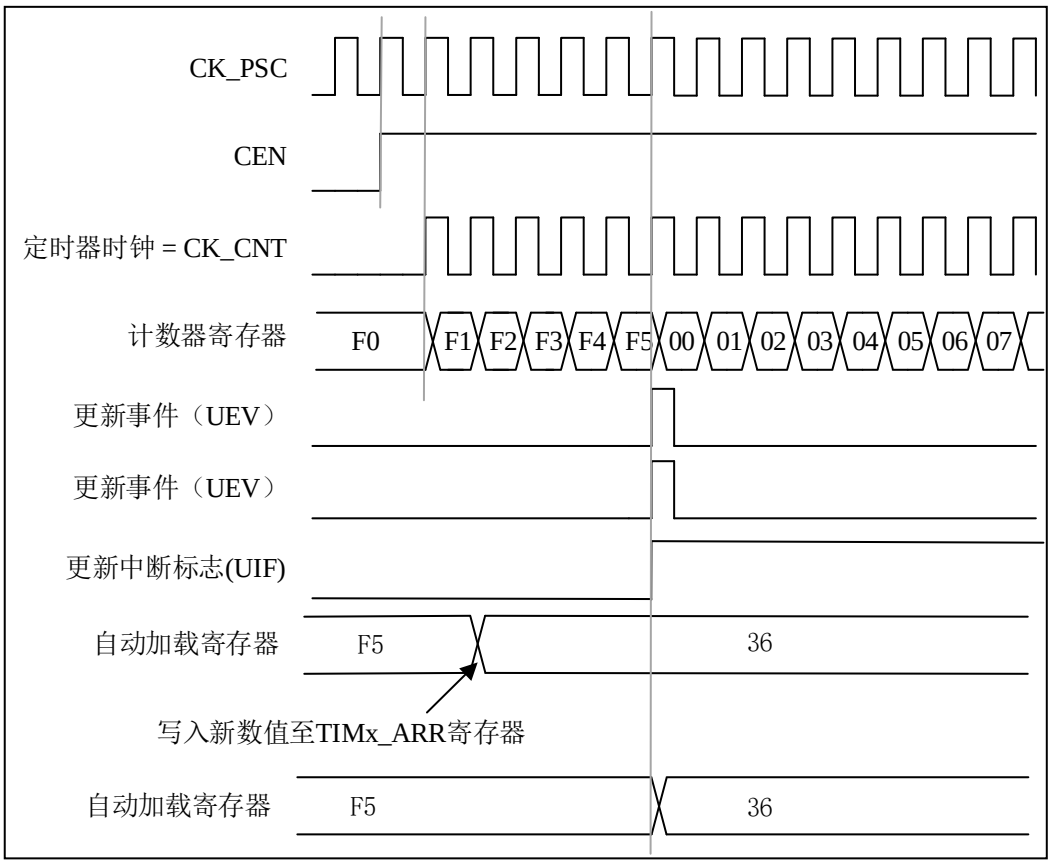


图 12-9 计数器时序图，当 ARPEN=1 时的更新事件（预装入了 TIMx_AR）



12.3.2.2 向下计数模式

如果使用向下计数模式，计数器从自动装入的值（TIMx_AR 计数器的值）开始向下计数到 0，然后从自动装入的值重新开始并且产生一个计数器向下溢出事件。

在每次计数器溢出时可以产生更新事件，在 TIMx_EVTGEN 寄存器中（通过软件方式或者使用从模式控制器）设置 UDGN 位，也同样可以产生一个更新事件。

为了避免在向预装载寄存器中写入新值时更新影子寄存器，可以设置 TIMx_CTRL1 寄存器中的 UPDIS 位，可以禁止更新事件。因此 UPDIS 位被清为'0'之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，同时预分频器的计数器重新从 0 开始（但预分频系数不变）。

此外，如果设置了 TIMx_CTRL1 寄存器中的 UPRS 位（选择更新请求），设置 UDGN 位将产生一个更新事件 UEV 但不设置 UDITF 标志（因此不产生中断和 DMA 请求），这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

更新事件发生后，所有的寄存器都被更新，硬件同时（根据 UPRS 位）设置更新标志位（TIMx_STS 寄存器中的 UDITF 位）。

- 预分频器的缓存器被置入预装载寄存器的值（TIMx_PSC 寄存器的值）。
- 当前的自动加载寄存器被更新为预装载值（TIMx_AR 寄存器中的内容）。*注：自动装载在计数器重载入之前被更新，因此下一个周期将是预期的值。*

以下是一些当 TIMx_AR=0x36 时，计数器在不同时钟频率下的操作示例。

图 12-10 计数器时序图，内部时钟分频因子为 1

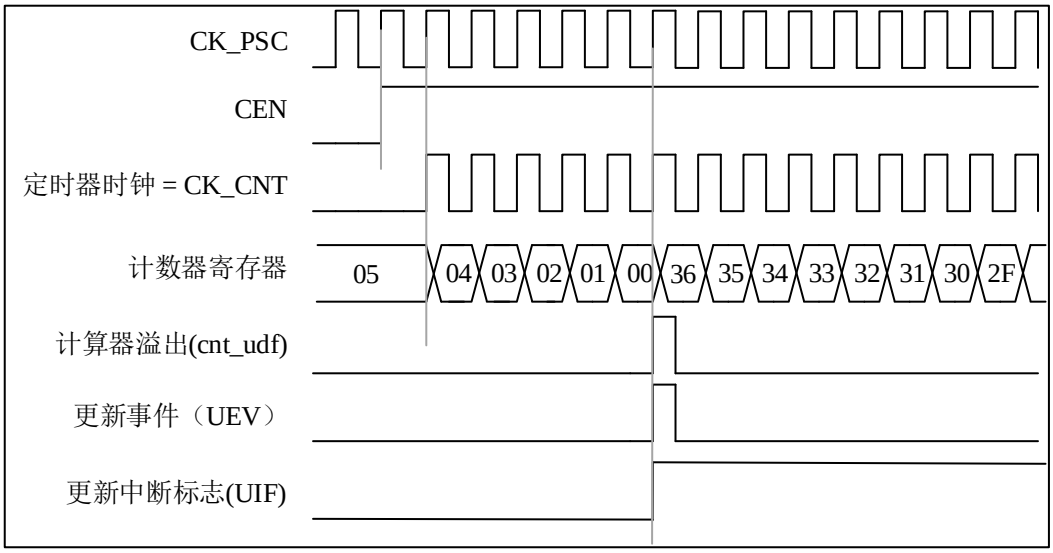


图 12-11 计数器时序图，内部时钟分频因子为 2

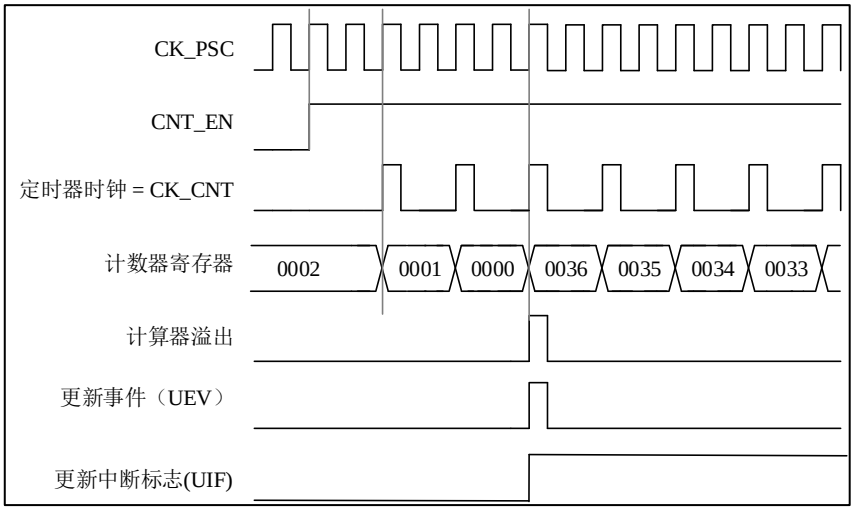


图 12-12 计数器时序图，内部时钟分频因子为 4

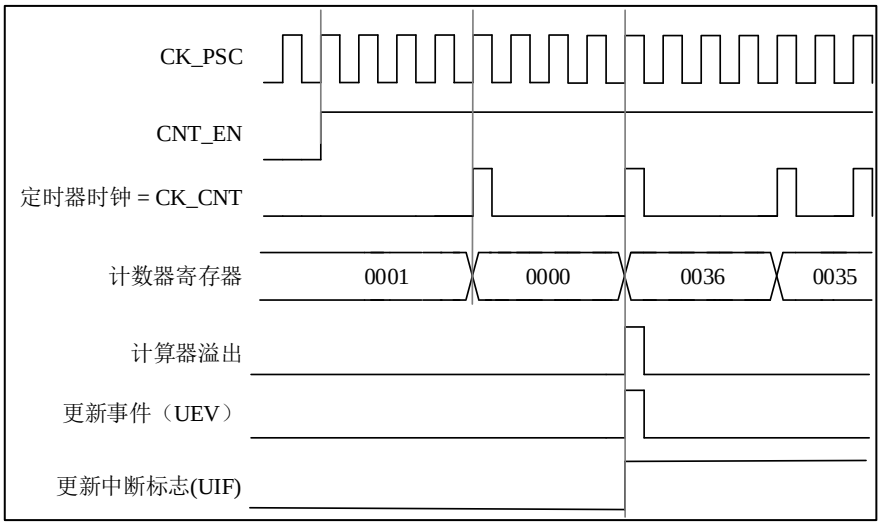


图 12-13 计数器时序图，内部时钟分频因子为 N

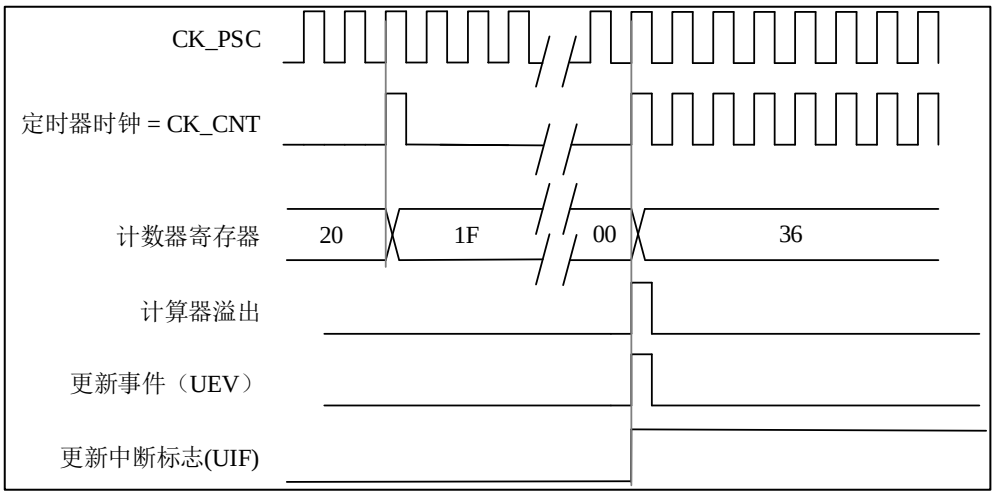
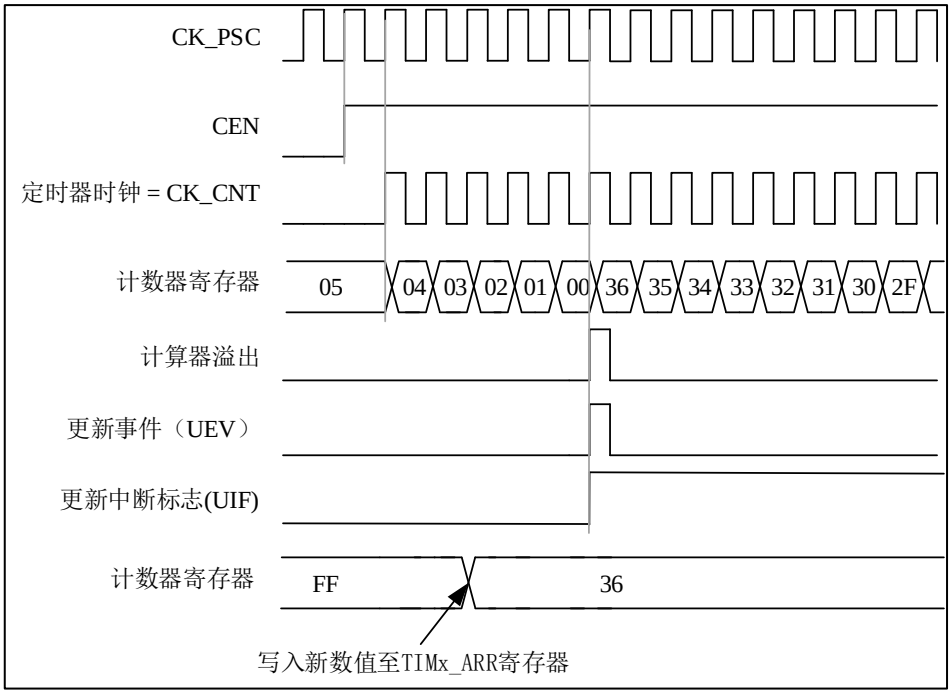


图 12-14 计数器时序图，当没有使用重复计数器时的更新事件



12.3.2.3 中央对齐模式（向上/向下计数）

如果使用中央对齐模式，计数器从 0 开始计数到自动加载的值（TIMx_AR 寄存器）-1，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

在此模式下，不能写入 TIMx_CTRL1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

可以在每次计数上溢和每次计数下溢时产生更新事件；也可以通过（软件或者使用从模式控制器）设置 TIMx_EVTGEN 寄存器中的 UDGN 位产生更新事件。然后，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

为了避免在向预装载寄存器中写入新值时更新影子寄存器，可以设置 TIMx_CTRL1 寄存器中的 UPDIS 位禁止 UEV 事件。因此 UPDIS 位被清为'0'之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

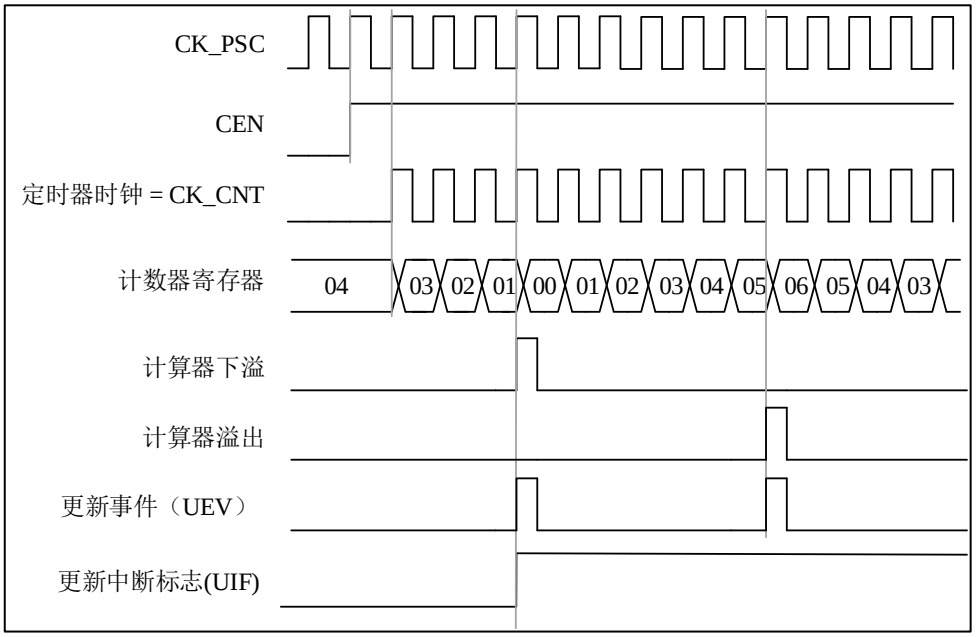
此外，如果设置了 TIMx_CTRL1 寄存器中的 UPRS 位（选择更新请求），设置 UDGN 位将产生一个更新事件 UEV 但不设置 UDITF 标志（因此不产生中断和 DMA 请求），这是为了避免在发生捕获事件并清除计数器时，同时产生更新和捕获中断。

在发生更新事件时，所有的寄存器都被更新，并且（根据 UPRS 位的设置）更新标志位（TIMx_STS 寄存器中的 UDITF 位）也被设置。

- 预分频器的缓存器被加载为预装载（TIMx_PSC 寄存器）的值。
- 当前的自动加载寄存器被更新为预装载值（TIMx_AR 寄存器中的内容）。*注：如果因为计数器溢出而产生更新，自动重装载将在计数器重载入之前被更新，因此下一个周期将是预期的值（计数器被装载为新的值）。*

以下是一些计数器在不同时钟频率下的操作的示例：

图 12-15 计数器时序图，内部时钟分频因子为 1，TIMx_AR=0x6



1． 这里使用了中心对齐模式 1（详见 12.4.2 节）。

图 12-16 计数器时序图，内部时钟分频因子为 2

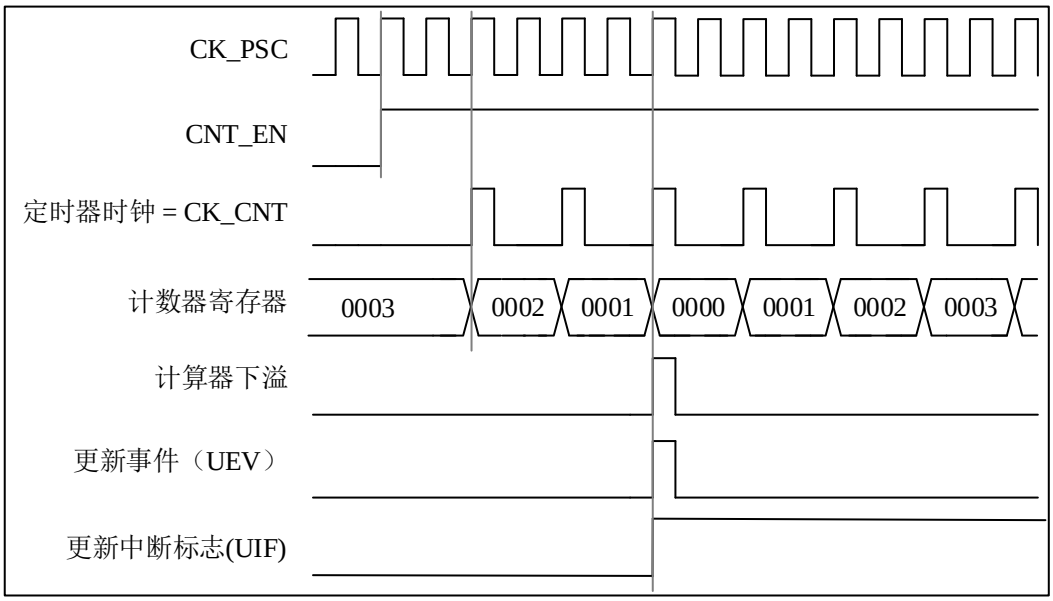


图 12-17 计数器时序图，内部时钟分频因子为 4，TIMx_AR=0x36

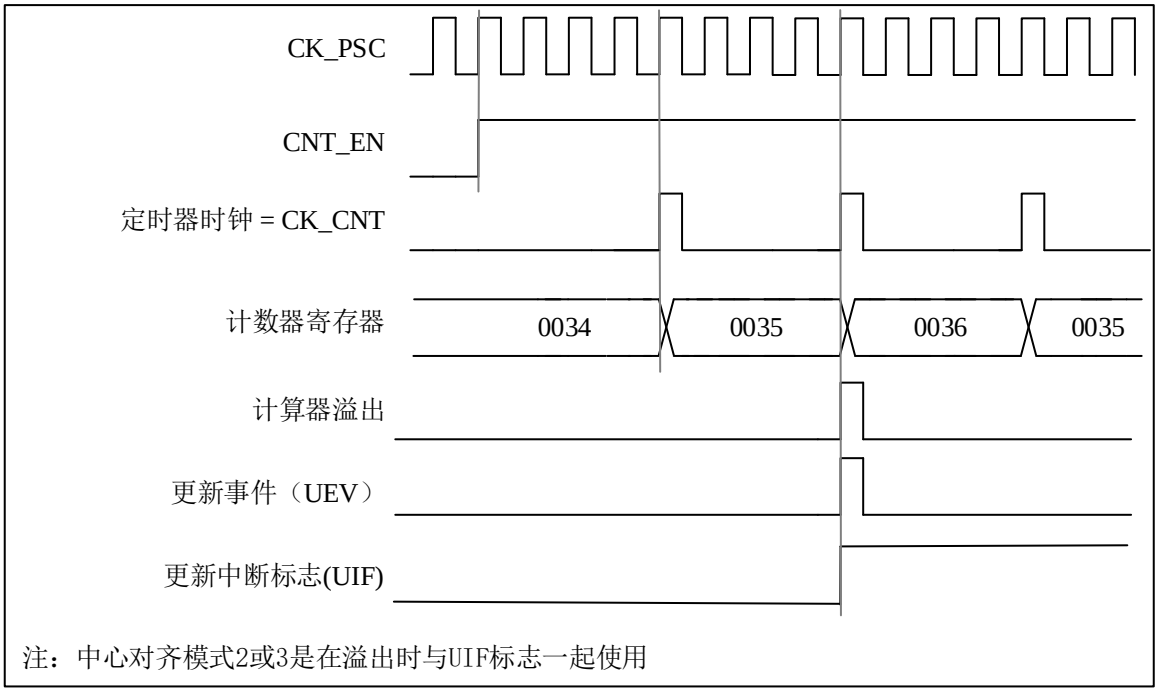


图 12-18 计数器时序图，内部时钟分频因子为 N

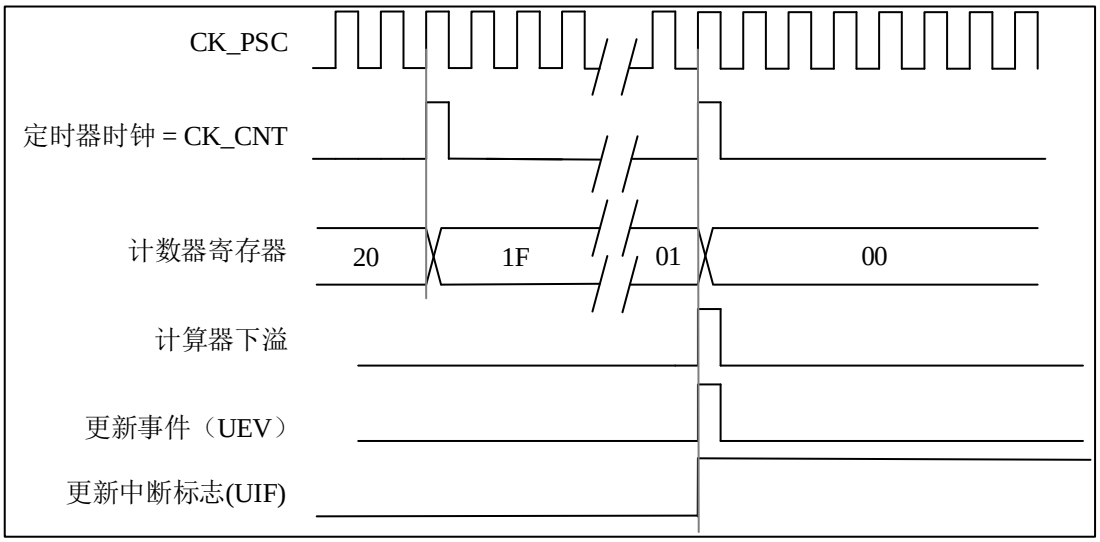


图 12-19 计数器时序图，ARPEN=1 时的更新事件（计数器下溢）

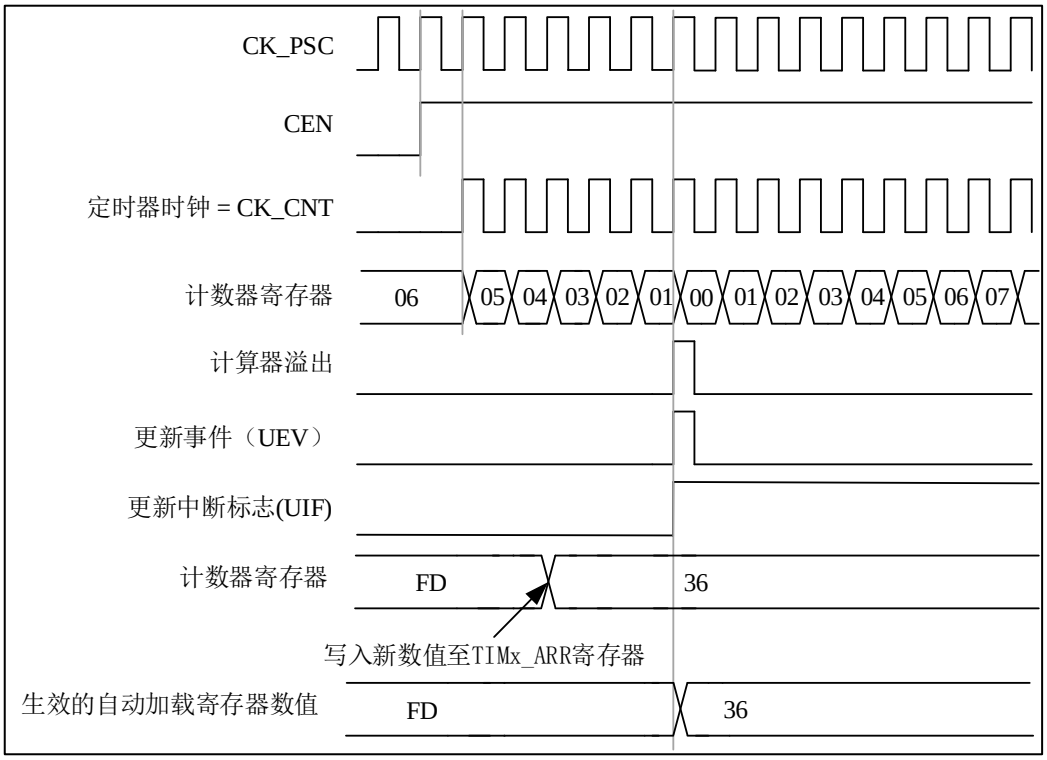
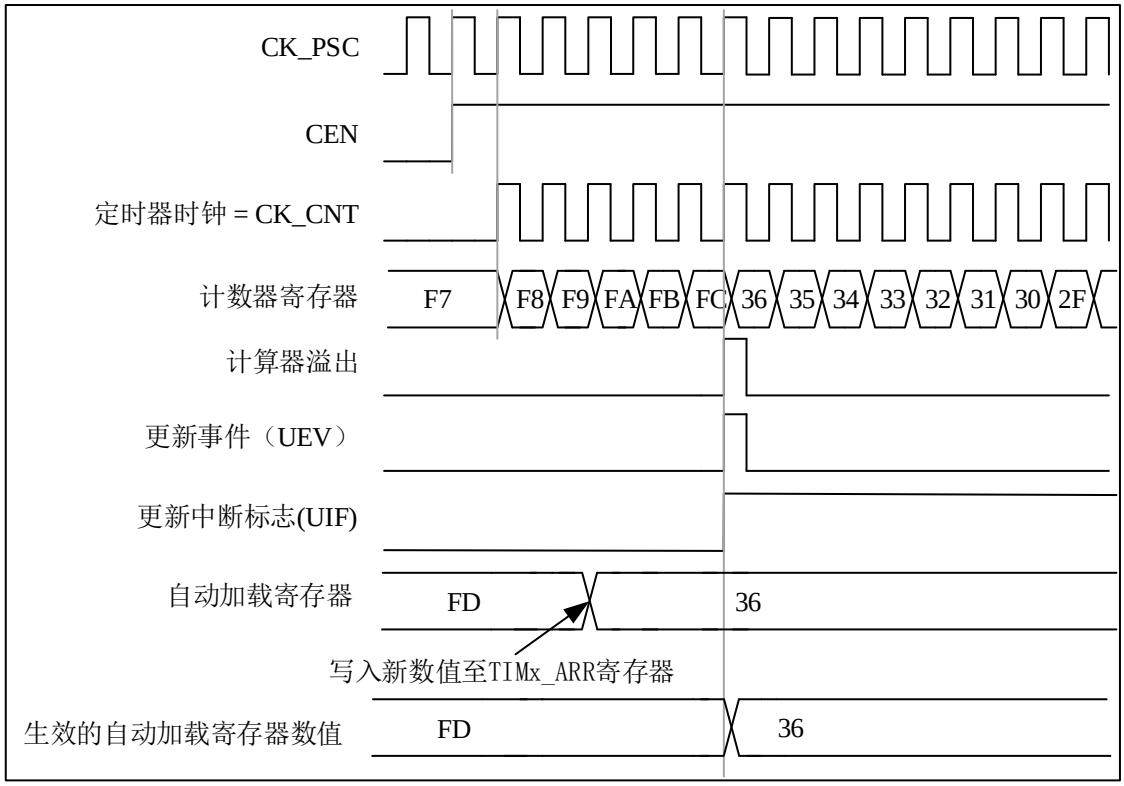


图 12-20 计数器时序图，ARPEN=1 时的更新事件（计数器溢出）



12.3.3 时钟选择

计数器时钟由以下时钟源提供:

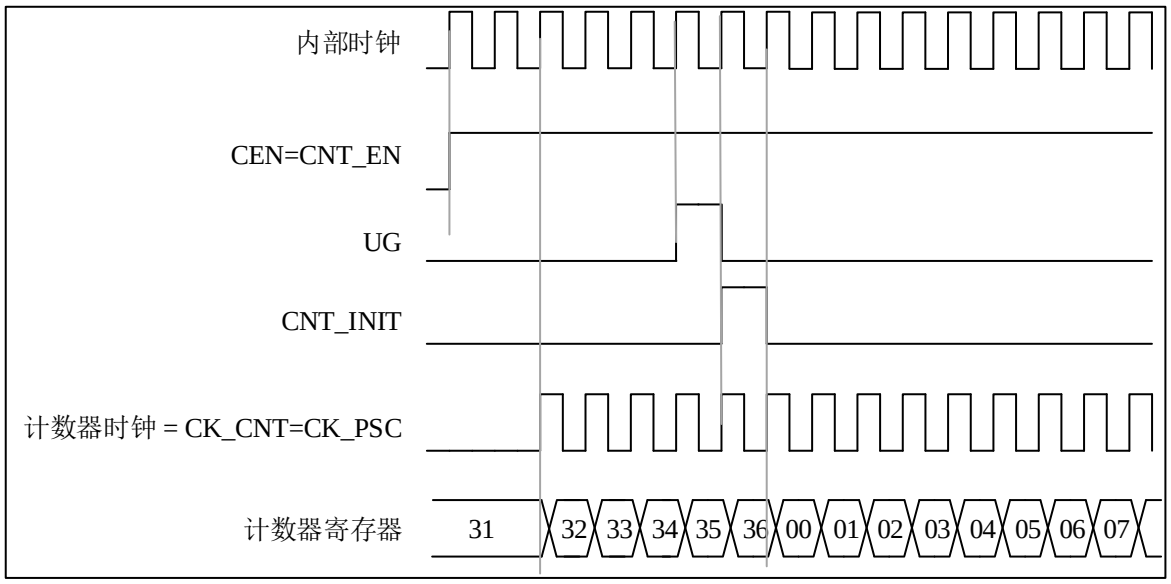
- 内部时钟（CK_INT）
- 外部时钟模式 1：外部输入脚（TIx）
- 外部时钟模式 2：外部触发输入（ETR）
- 内部触发输入（ITRx）：使用一个定时器作为另一个定时器的预分频器，如可以配置一个定时器 Timer1 而作为另一个定时器 Timer2 的预分频器。参见 12.3.15。

12.3.3.1 内部时钟源（CK_INT）

如果禁止了从模式控制器（TIMx_SMCTRL 寄存器的 SMSEL=000），则 CNTEN、DIR（TIMx_CTRL1 寄存器） 和 UDGN 位（TIMx_EVTGEN 寄存器）是事实上的控制位，并且只能被软件修改（UDGN 位仍被自动清除）。只要 CNTEN 位被写成'1'，预分频器的时钟就由内部时钟 CK_INT 提供。

下图显示了控制电路和向上计数器在一般模式下，不带预分频器时的操作。

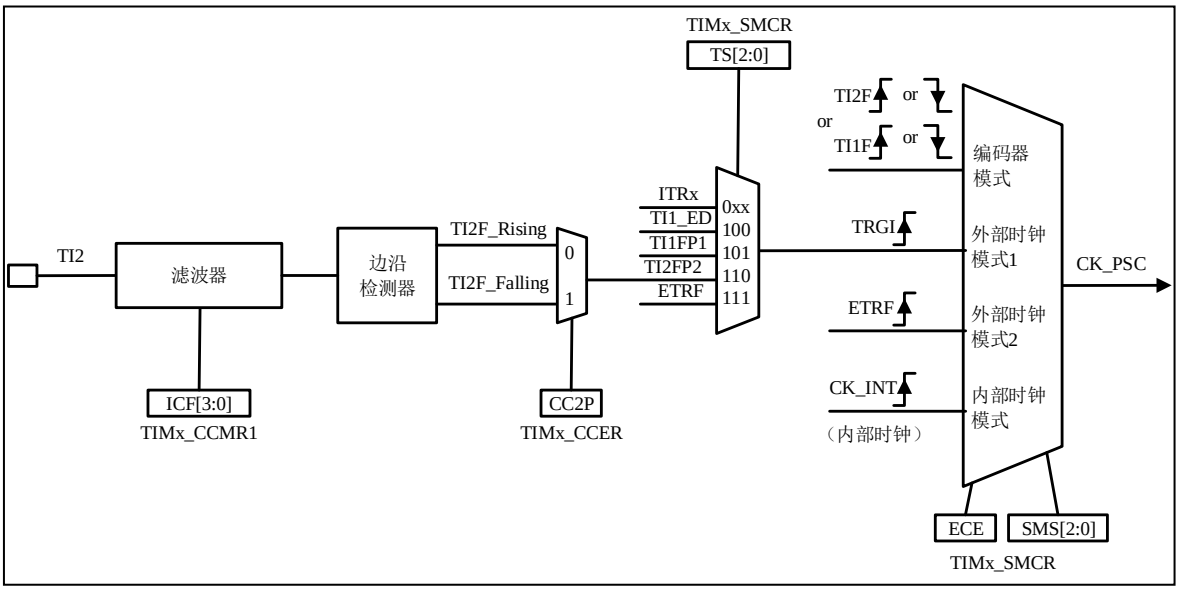
图 12-21 一般模式下的控制电路，内部时钟分频因子为 1



12.3.3.2 外部时钟源模式 1

当 TIMx_SMCTRL 寄存器的 SMSEL=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

图 12-22 TI2 外部时钟连接例子



例如，要配置向上计数器在 T12 输入端的上升沿计数，使用下列步骤:

- 配置 TIMx_CCMOD1 寄存器 CC2SEL='01'，配置通道 2 检测 TI2 输入的上升沿
- 配置 TIMx_CCMOD1 寄存器的 IC2F[3:0], 选择输入滤波器带宽(如果不需要滤波器, 保持 IC2F=0000)

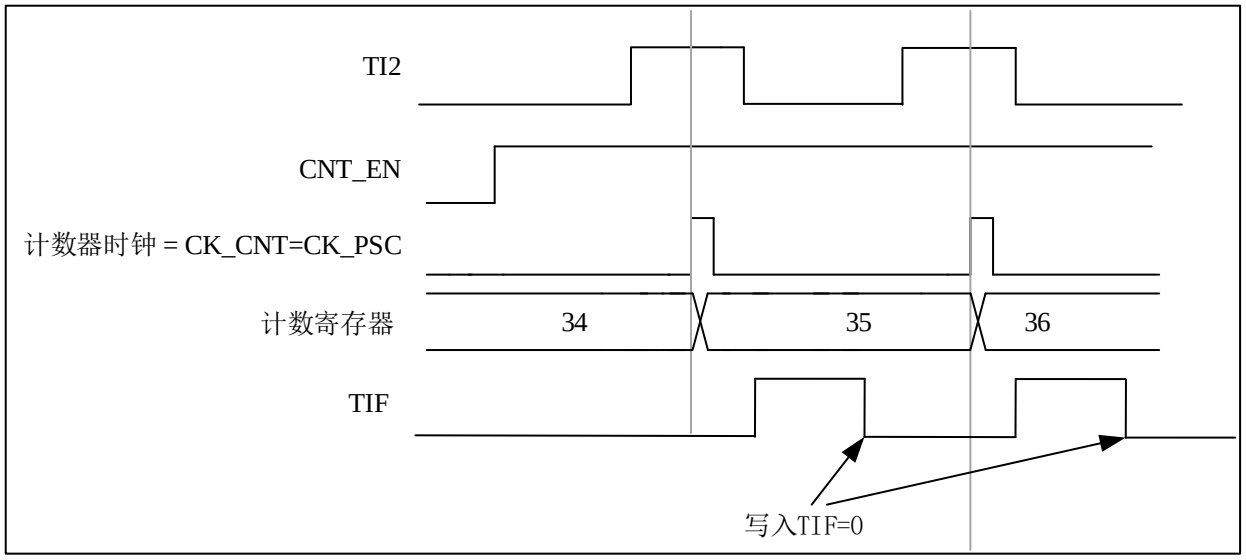
注：捕获预分频器不用作触发，所以不需要对它进行配置

- 配置 TIMx_CCEN 寄存器的 CC2P='0'，选定上升沿极性
- 配置 TIMx_SMCTRL 寄存器的 SMSEL='111'，选择定时器外部时钟模式 1
- 配置 TIMx_SMCTRL 寄存器中的 TSEL='110'，选定 TI2 作为触发输入源
- 设置 TIMx_CTRL1 寄存器的 CNTEN='1'，启动计数器

当上升沿出现在 TI2，计数器计数一次，且 TITF 标志被设置。

在 TI2 的上升沿和计数器实际时钟之间的延时，取决于在 TI2 输入端的重新同步电路。

图 12-23 外部时钟模式 1 下的控制电路

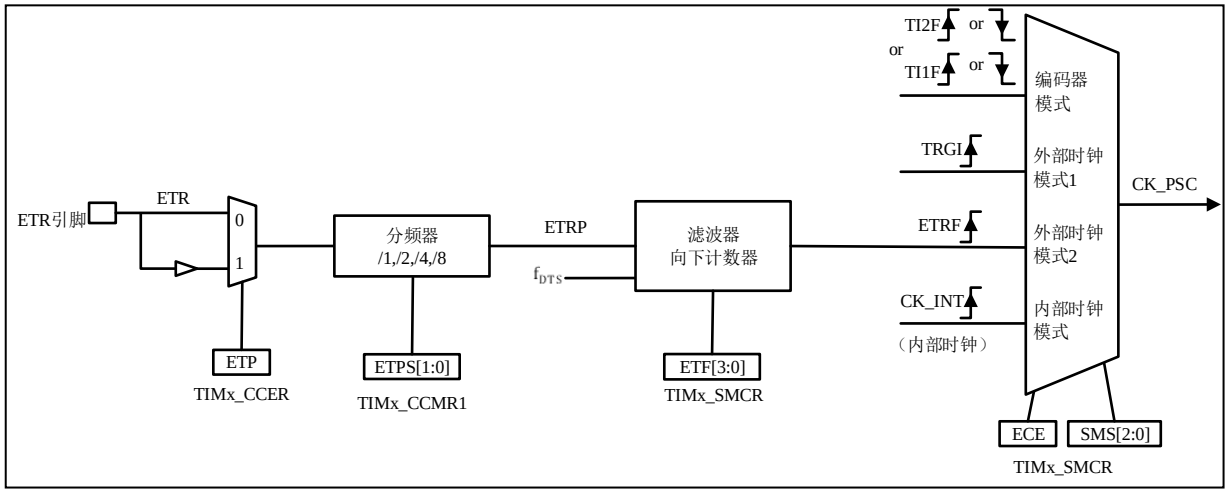


12.3.3.3 外部时钟源模式 2

选定此模式的方法是设置 TIMx_SMCTRL 寄存器中的 EXCEN=1（设置 TIMx_SMCTRL 寄存器中的 SMSEL=111，同时设置 TSEL=111 有相同效果）。

计数器能够在外部触发 ETR 的每一个上升沿或下降沿计数。下图是外部触发输入的框图。

图 12-24 外部触发输入框图



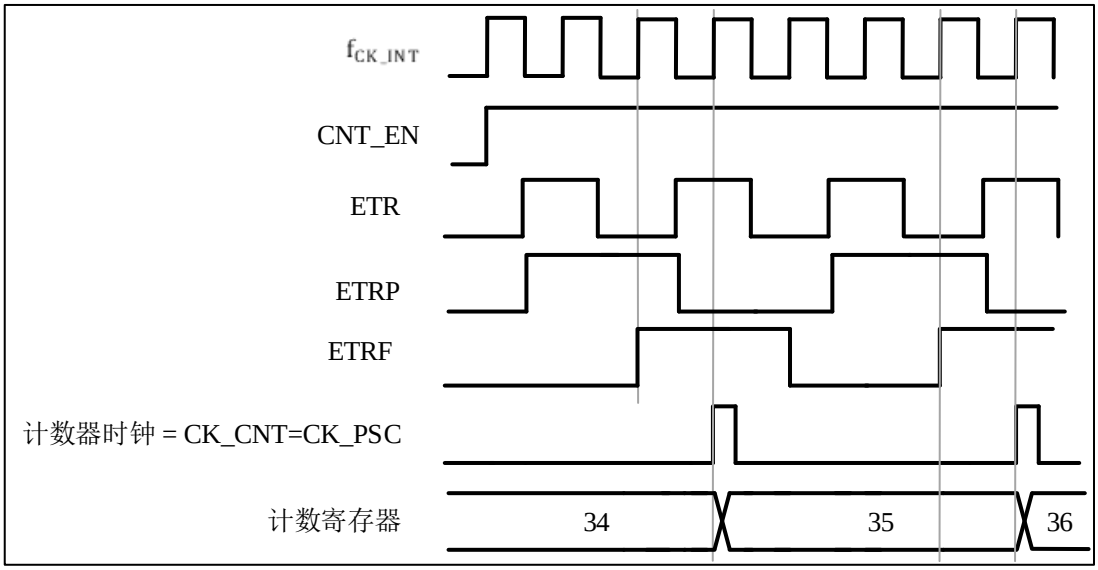
例如，要配置在 ETR 下每 2 个上升沿计数一次的向上计数器，使用下列步骤：

- 本例中不需要滤波器，置TIMx_SMCTRL寄存器中的EXTF[3:0]=0000
- 设置预分频器，置TIMx_SMCTRL寄存器中的EXTPS[1:0]=01
- 设置在ETR的上升沿检测，置TIMx_SMCTRL寄存器中的EXTP=0
- 开启外部时钟模式2，置TIMx_SMCTRL寄存器中的EXCEN=1
- 启动计数器，置TIMx_CTRL1寄存器中的CNTEN=1

计数器在每 2 个 ETR 上升沿计数一次。

在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

图 12-25 外部时钟模式 2 下的控制电路



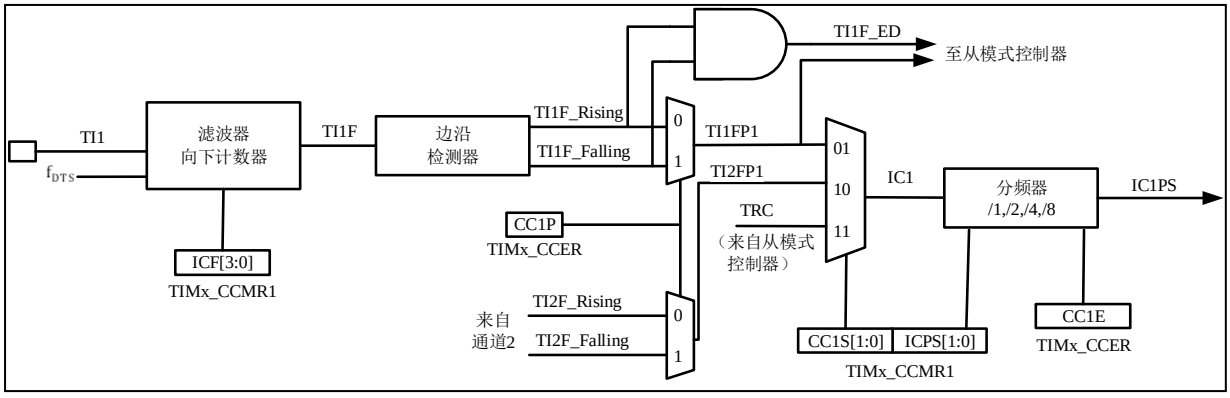
12.3.4 捕获/比较通道

捕获/比较通道都是独立的，都是围绕着一个捕获/比较寄存器（含影子寄存器），包括捕获的输入部分（数字滤波、多路复用和预分频器）和比较输出部分（比较器和输出控制）。

下面几张图是一个捕获/比较通道概览。

输入部分对相应的 TIx 输入信号采样，并产生一个滤波后的信号 TIxF。然后，一个带极性选择的边缘检测器产生一个信号（TIxFPx），它可以作为从模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器（ICxPS）。

图 12-26 捕获/比较通道（如：通道 1 输入部分）



输出部分产生一个中间波形 OCxRef（高有效）作为基准，链的末端决定最终输出信号的极性。

图 12-27 捕获/比较通道 1 的主电路

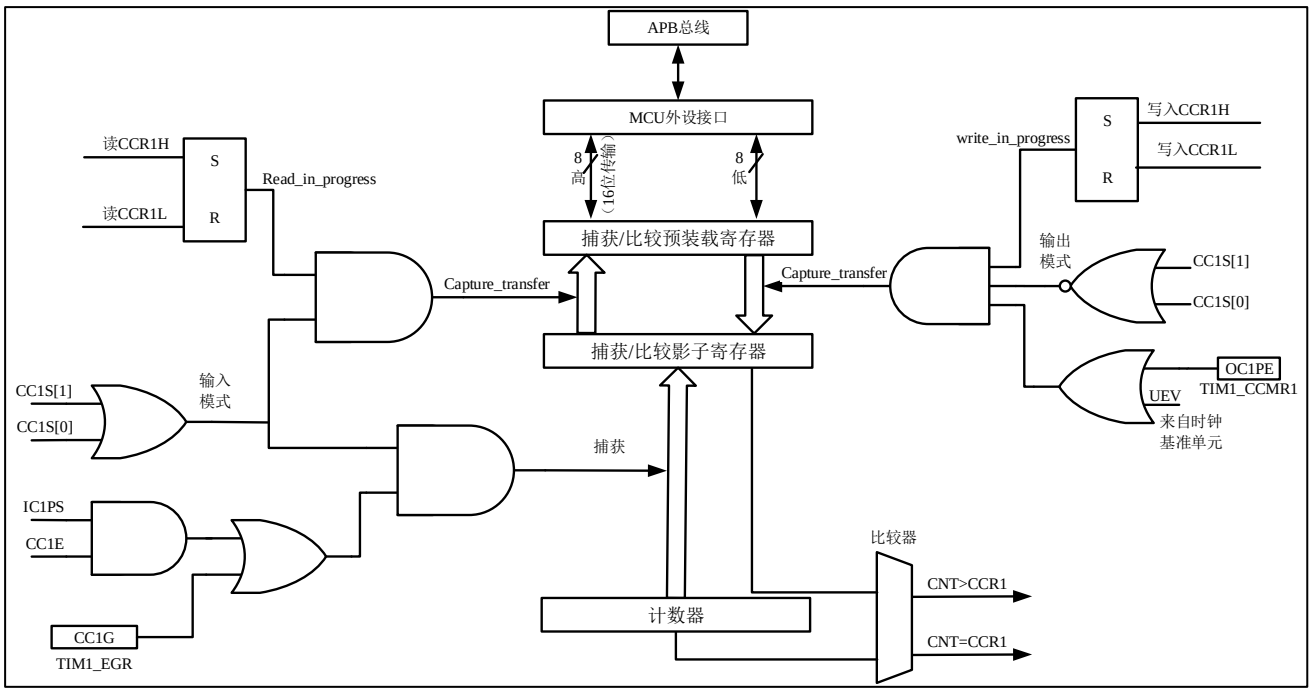
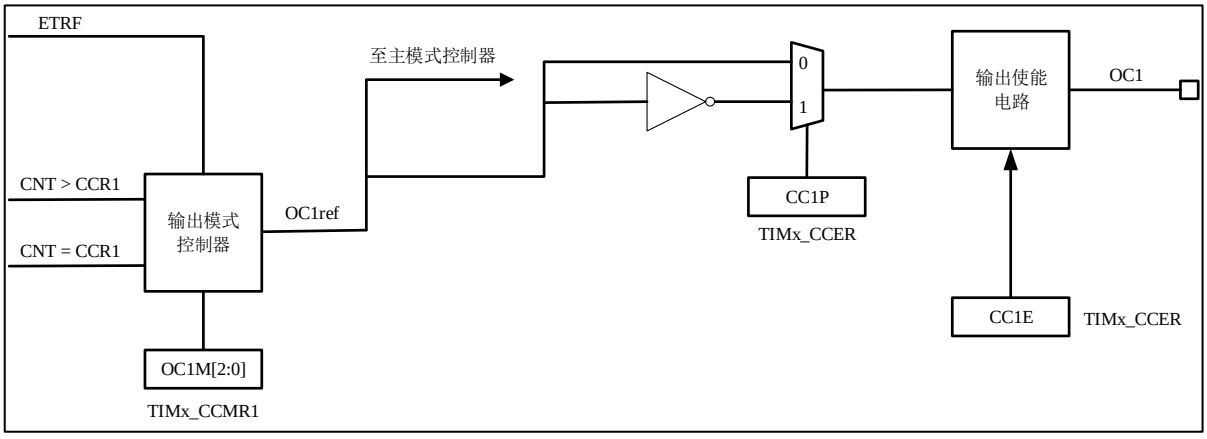


图 12-28 捕获/比较通道的输出部分（通道 1）



捕获/比较模块包括一个预装载寄存器和一个影子寄存器。读写过程仅操作预装载寄存器。

在捕获模式下，捕获发生在影子寄存器上，然后再复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器的内容和计数器进行比较。

12.3.5 输入捕获模式

配置成输入捕获模式时，当检测到 ICx 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（TIMx_CCDATx）中。当捕获事件发生时，相应的 CCxIF 标志（TIMx_STS 寄存器）被置'1'，如果使能了中断或者 DMA 操作，则将产生中断或者 DMA 操作。如果捕获事件发生时 CCxIF 标志已经为高，那么重复捕获标志 CCxOCF（TIMx_STS 寄存器）被置'1'。写 CCxIF=0 可清除 CCxIF，或读取存储在

TIMx_CCDATx 寄存器中的捕获数据也可清除 CCxIF。写 CCxOCF=0 可清除 CCxOCF。

以下示例说明如何将 TI1 输入的上升沿时捕获计数器的值到 TIMx_CCDAT1 寄存器中，步骤如下：

- 选择有效输入端：TIMx_CCDAT1 必须连接到 TI1 输入，所以写入 TIMx_CCMOD1 寄存器中的 CC1SEL=01，只要 CC1SEL 不为'00'，通道被配置为输入，并且 TM1_CCDAT1 寄存器变为只读。
- 根据输入信号的特点，配置输入滤波器为所需的带宽（即输入为 TIx 时，输入滤波器控制位是
- TIMx_CCMODx 寄存器中的 ICxF 位）。假设输入信号在最多 5 个内部时钟周期的时间内抖动， 我们须配置滤波器的带宽长于 5 个时钟周期。因此我们可以（以 f_{DTS} 频率）连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIMx_CCMOD1 寄存器中写入 IC1F=0011。
- 选择 TI1 通道的有效转换边沿，在 TIMx_CCEN 寄存器中写入 CC1P=0（上升沿）。
- 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止（写 TIMx_CCMOD1 寄存器的 IC1PS=00）。
- 设置 TIMx_CCEN 寄存器的 CC1EN=1，允许捕获计数器的值到捕获寄存器中。
- 必要的时候，通过设置 TIMx_DINTEN 寄存器中的 CC1IE 位允许相关中断请求，通过设置 TIMx_DINTEN 寄存器中的 CC1DE 位允许 DMA 请求。

当发生一个输入捕获时：

- 产生有效的电平转换时，计数器的值被传送到 TIMx_CCDAT1 寄存器。
- CC1ITF 标志被设置（中断标志）。当发生至少 2 个连续的捕获时，而 CC1ITF 未曾被清除，CC1OCF 也被置'1'。
- 如设置了 CC1IE 位，则会产生一个中断。
- 如设置了 CC1DE 位，则还会产生一个 DMA 请求。

在处理捕获溢出时，建议在读出捕获溢出标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的捕获溢出信息。

注：设置 TIMx_EVTGEN 寄存器中相应的 CCxGN 位，可以通过软件产生输入捕获中断和/或 DMA 请求。

12.3.6 PWM 输入模式

此模式是输入捕获模式的一个特例，除以下区别外，操作与输入捕获模式相同：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这 2 个 ICx 信号为边沿有效，但是极性相反。
- 其中一个 TIxFP 信号被作为触发输入信号，而从模式控制器被配置成复位模式。

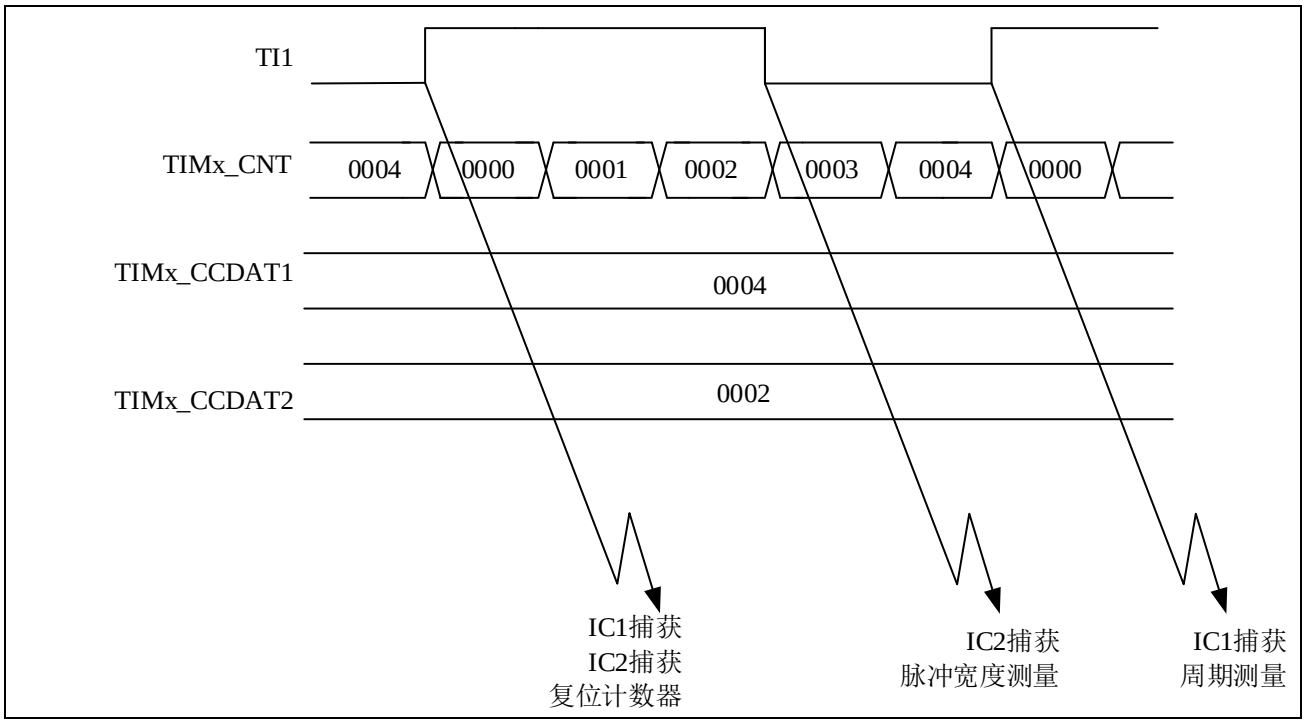
例如，为了测量输入到 TI1 上的 PWM 信号的长度（TIMx_CCDAT1 寄存器）和占空比（TIMx_CCDAT2 寄存器），可以采用如下具体步骤（取决于 CK_INT 的频率和预分频器的值）

- 选择 TIMx_CCDAT1 的有效输入：置 TIMx_CCMOD1 寄存器的 CC1SEL=01（选择 TI1）。
- 选择 TI1FP1 的有效极性（用来捕获数据到 TIMx_CCDAT1 中和清除计数器）：置 CC1P=0（上升沿

有效)。

- 选择 TIMx_CCDAT2 的有效输入：置 TIMx_CCMOD1 寄存器的 CC2SEL=10（选择 TI1)。
- 选择 TI1FP2 的有效极性（捕获数据到 TIMx_CCDAT2)：置 CC2P=1（下降沿有效)。
- 选择有效的触发输入信号：置 TIMx_SMCTRL 寄存器中的 TSEL=101（选择 TI1FP1)。
- 配置从模式控制器为复位模式：置 TIMx_SMCTRL 中的 SMSEL=100。
- 使能捕获：置 TIMx_CCEN 寄存器中 CC1EN=1 且 CC2EN=1。

图 12-29 PWM 输入模式时序



硬件上只有 TI1FP1 和 TI2FP2 连到了从模式控制器，所以 PWM 输入模式只能使用 TIMx_CH1/TIMx_CH2 信号。

12.3.7 强置输出模式

配置成输出模式(TIMx_CCMODx 寄存器中 CCxSEL=00)下，输出比较信号(OCxREF 和相应的 OCx)能够直接由软件强置为有效或无效状态，而不依赖于输出比较寄存器和计数器间的比较结果。

配置 TIMx_CCMODx 寄存器中相应的 OCxMD=101，即可强置输出比较信号（OCxREF/OCx）为有效状态。这样 OCxREF 被强置为高电平（OCxREF 始终为高电平有效)，同时 OCx 得到 CCxP 极性位相反的值。

例如：CCxP=0（OCx 高电平有效)，则 OCx 被强置为高电平。

配置 TIMx_CCMODx 寄存器中的 OCxMD=100，可强置 OCxREF 信号为低。

该模式下，在 TIMx_CCDATx 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改。

因此仍然会产生相应的中断和 DMA 请求。这将会在下面的输出比较模式一节中介绍。

12.3.8 输出比较模式

此项功能具有控制输出特定的波形，或者指示一段给定的的时间已经到时。

当计数器与捕获/比较寄存器的内容相同时，输出比较功能做如下操作:

- 将输出比较模式（TIMx_CCMODx 寄存器中的 OCxMD 位）和输出极性（TIMx_CCEN 寄存器中的 CCxP 位)定义的值输出到对应的引脚上。在比较匹配时，输出引脚可以保持它的电平(OCxMD=000)、被设置成有效电平(OCxMD=001)、被设置成无效电平(OCxMD=010)或进行翻转(OCxMD=011)。
- 设置中断状态寄存器中的标志位（TIMx_STS 寄存器中的 CCxIF 位)。
- 若设置了相应的中断屏蔽（TIMx_DINTEN 寄存器中的 CCxIEN 位），则产生一个中断。
- 若设置了相应的使能位（TIMx_DINTEN 寄存器中的 CCxDEN 位，TIMx_CTRL2 寄存器中的 CCDSEL 位选择 DMA 请求功能），则产生一个 DMA 请求。

TIMx_CCMODx 中的 OCxPEN 位选择 TIMx_CCDATx 寄存器是否需要使用预装载寄存器。在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出没有影响。

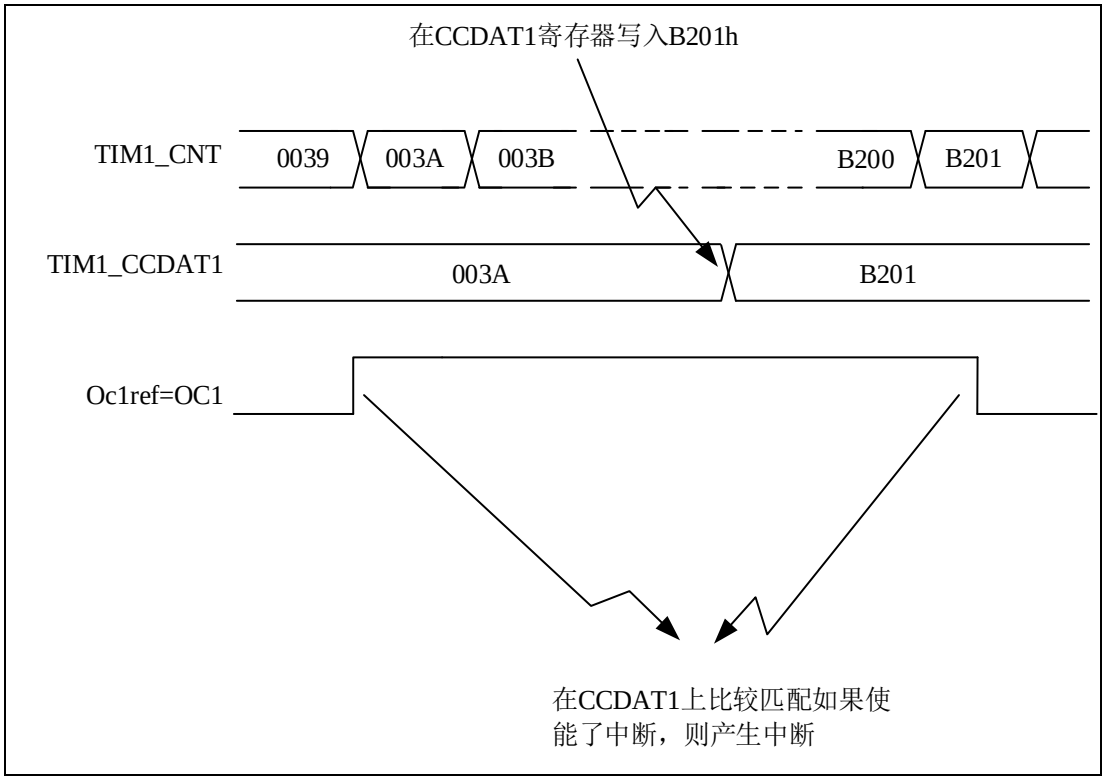
同步的精度可以达到计数器的一个计数周期。输出比较模式（在单脉冲模式下）也能用来输出一个单脉冲。

输出比较模式的配置步骤:

- 选择计数器时钟（内部，外部，预分频器）
- 将相应的数据写入 TIMx_AR 和 TIMx_CCDATx 寄存器中
- 如果要产生一个中断请求和/或一个 DMA 请求，设置 CCxIEN 位和/或 CCxDEN 位。
- 选择输出模式，例如:
  - 要求计数器与 CCDATx 匹配时翻转 OCx 的输出引脚，设置 OCxMD=011
  - 置 OCxPEN = 0 禁用预装载寄存器
  - 置 CCxP = 0 选择极性为高电平有效
  - 置 CCxEN = 1 使能输出
- 设置 TIMx_CTRL1 寄存器的 CNTEN 位启动计数器

TIMx_CCDATx 寄存器能够在任何时候通过软件进行更新以控制输出波形，条件是未使用预装载寄存器（OCxPEN='0'，否则 TIMx_CCDATx 影子寄存器只能在发生下一次更新事件时被更新)。下图给出了一个例子。

图 12-30 输出比较模式，翻转 OC1



12.3.9 PWM 模式

脉冲宽度调制（PWM）模式可以产生一个由 TIMx_AR 寄存器确定频率、由 TIMx_CCDATx 寄存器确定占空比的信号。

TIMx_CCMODx 寄存器中的 OCxMD 位写入'110'（PWM 模式 1）或'111'（PWM 模式 2），能够独立地设置每个 OCx 输出通道产生一路 PWM。必须设置 TIMx_CCMODx 寄存器 OCxPEN 位以使能相应的预装载寄存器，最后还要设置 TIMx_CTRL1 寄存器的 ARPEN 位，（在向上计数或中心对称模式中）使能自动重装载的预装载寄存器。

仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EVTGEN 寄存器中的 UDGN 位来初始化所有的寄存器。

OCx 的极性可以通过软件在 TIMx_CCEN 寄存器中的 CCxP 位进行设置，它可以设置为高电平有效或低电平有效。TIMx_CCEN 寄存器中的 CCxEN 位控制 OCx 输出使能。详见 TIMx_CCENx 寄存器的描述。

在 PWM 模式（模式 1 或模式 2）下，TIMx_CNT 和 TIMx_CCDATx 始终在进行比较，（依据计数器的计数方向）以确定是否符合 TIMx_CCDATx≤TIMx_CNT 或者 TIMx_CNT≤TIMx_CCDATx。然而为了与 OCREF_CLR 的功能(在下一个 PWM 周期之前, ETR 信号上的一个外部事件能够清除 OCxREF)一致，OCxREF 信号只能在下述条件下产生：

- 当比较的结果改变
- 当输出比较模式（TIMx_CCMODx 寄存器中的 OCxMD 位）从"冻结"（无比较，OCxMD='000'）切换到某个 PWM 模式（OCxMD='110'或'111'）。

这样在运行中可以通过软件强置 PWM 输出。

根据 TIMx_CTRL1 寄存器中 CAMSEL 位的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

12.3.9.1 PWM 边沿对齐模式

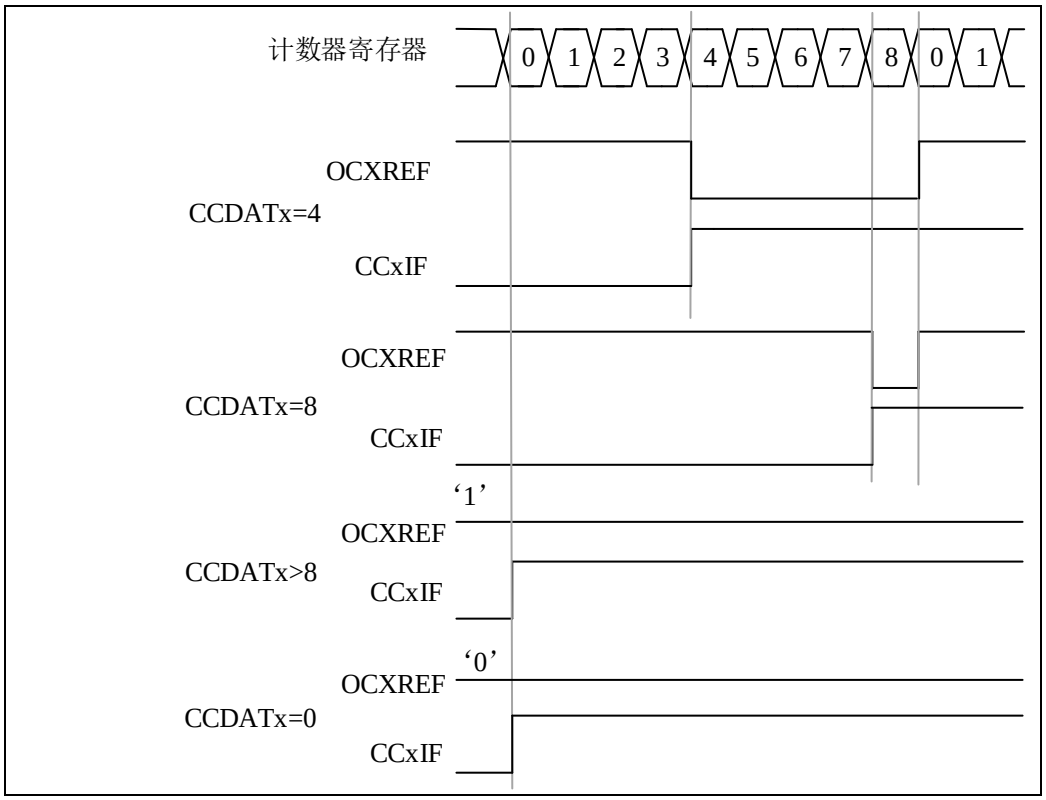
- 向上计数配置

当 TIMx_CTRL1 寄存器中的 DIR 位为低的时候执行向上计数。参看 12.3.2 节。

当 TIMx_CNT<TIMx_CCDATx 时 PWM 信号参考 OCxREF 为高，否则为低。如果 TIMx_CCDATx 中的比较值大于自动重装载值（TIMx_AR），则 OCxREF 保持为'1'。如果比较值为 0，则 OCxREF 保持为'0'。

下图为 TIMx_AR=8 时边沿对齐的 PWM 波形示例。

图 12-31 边沿对齐的 PWM 波形（AR=8）



- 向下计数的配置

当 TIMx_CTRL1 寄存器的 DIR 位为高时执行向下计数。参看 12.3.2 节。

在 PWM 模式 1，当 TIMx_CNT>TIMx_CCDATx 时参考信号 OCxREF 为低，否则为高。如果 TIMx_CCDATx 中的比较值大于 TIMx_AR 中的自动重装载值，则 OCxREF 保持为'1'。

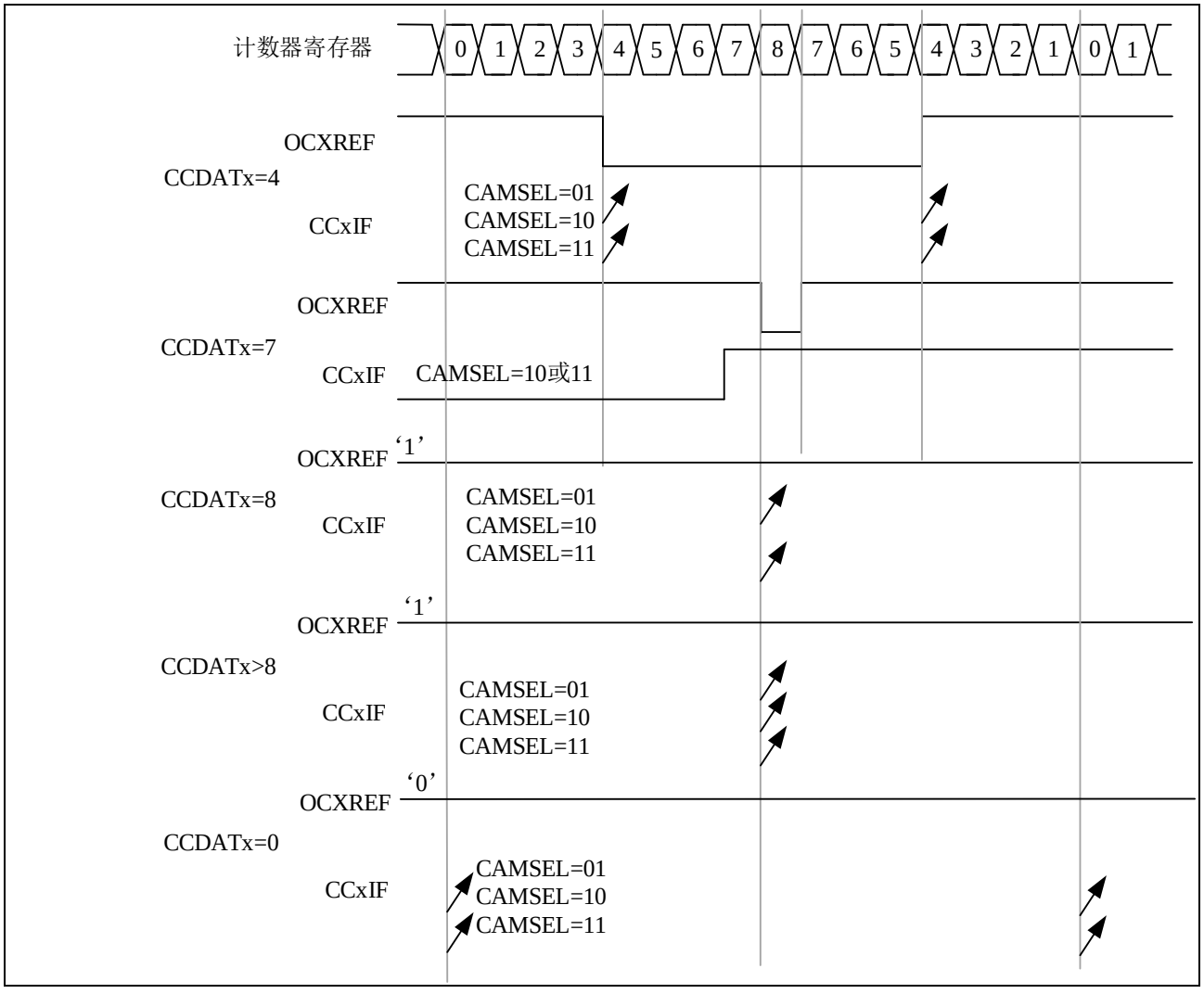
注：该模式下不能产生 0%的 PWM 波形。

- PWM 中央对齐模式
- 配置 TIMx_CTRL1 寄存器中的 CAMSEL 位不为'00'时为中央对齐模式（所有其他的配置对 OCxREF/OCx 信号都有相同的作用）。根据不同的 CAMSEL 位设置，比较标志可以在计数器向上计数时被置'1'、在计数器向下计数时被置'1'、或在计数器向上和向下计数时被置'1'。TIMx_CTRL1 寄

存器中的计数方向位（DIR）由硬件更新，不要用软件修改。参看 12.3.2 节的中央对齐模式。

- 下图给出了一些中央对齐的 PWM 波形的示例
- TIMx_AR=8
- PWM 模式 1
- TIMx_CTRL1 寄存器中的 CAMSEL=01，在中央对齐模式 1 时，当计数器向下计数时设置比较标志。

图 12-32 中央对齐的 PWM 波形（APR=8）



12.3.9.2 使用中央对齐模式的提示：

- 在进入中央对齐模式时的时候，使用当前的向上/向下计数配置；这就意味着计数器向上还是向下计数取决于 TIMx_CTRL1 寄存器中 DIR 位的当前值。此外，软件不能同时修改 DIR 和 CAMSEL 位。
- 不推荐当运行在中央对齐模式时改写计数器，因为这样可能会产生预期之外的 PWM 波形和软件行为。特别地：

- 如果写入计数器的值大于自动重加载的值（TIMx_CNT>TIMx_AR），则方向不会被更新。例如，如果计数器正在向上计数，它就会继续向上计数。
- 如果将 0 或者 TIMx_AR 的值写入计数器，方向被更新，但不产生更新事件 UEV。

- 使用中央对齐模式最可靠的方法，就是在启动计数器之前产生一个软件更新（设置 TIMx_EVTGEN 位中的 UDGN 位）事件，并且不要在计数进行过程中修改计数器的值。

12.3.10 单脉冲模式

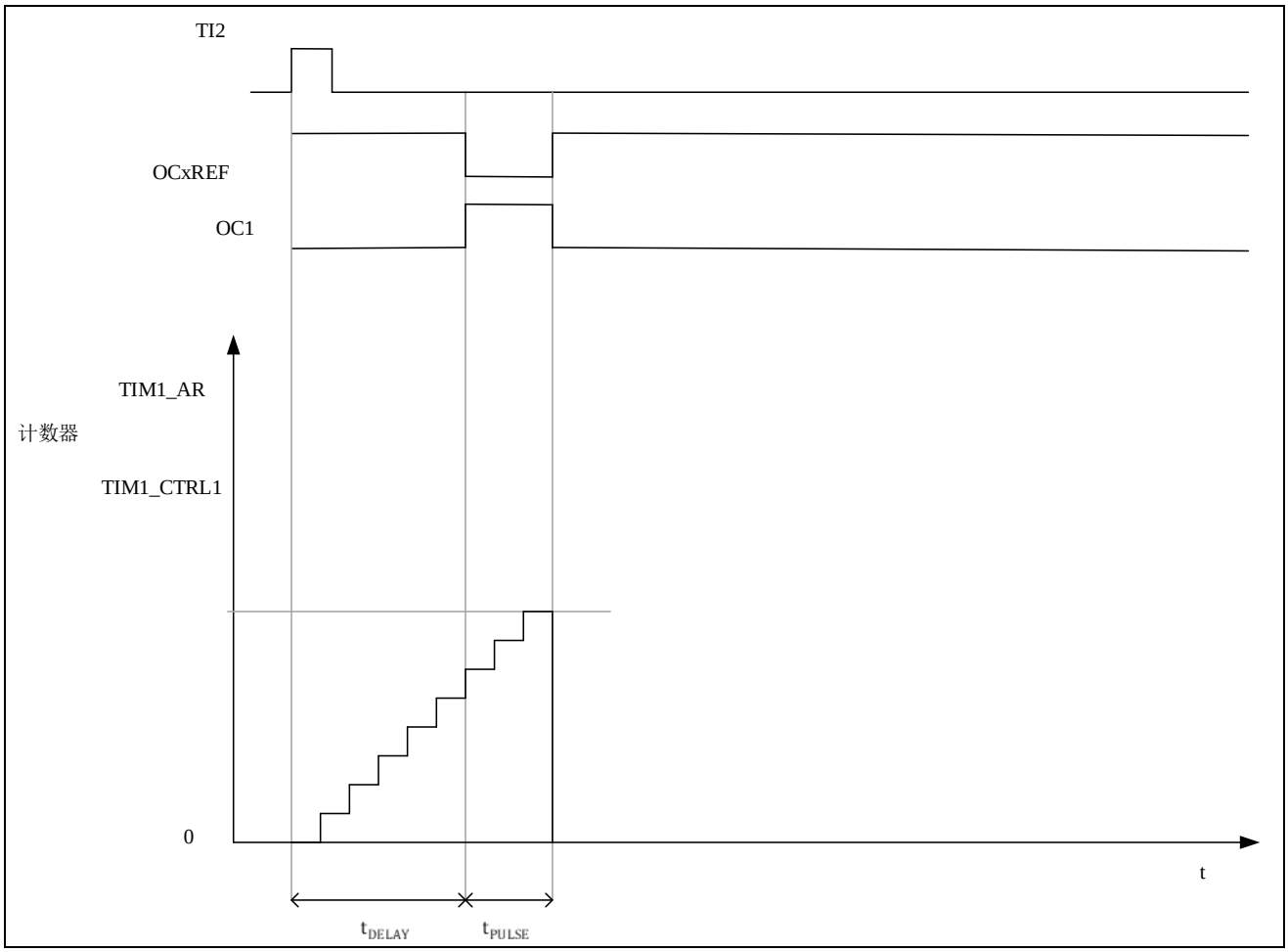
单脉冲模式（ONEPM）允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可程序控制的脉冲，是前述众多模式的一个特例。

可以通过从模式控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CTRL1 寄存器中的 ONEPM 位将选择单脉冲模式，这样可以让计数器自动地在产生下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前（当定时器正在等待触发），必须进行如下配置：

- 向上计数方式：CNT < CCDATx ≤ AR （特别地，0 < CCDATx），
- 向下计数方式：CNT > CCDATx。

图 12-33 单脉冲模式的例子



例如，需要在从 TI2 输入脚上检测到一个上升沿开始，延迟 t_{DELAY} 之后，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

假定 TI2FP2 作为触发 1:

- 置 TIMx_CCMOD1 寄存器中的 CC2SEL='01'，把 TI2FP2 映像到 TI2。
- 置 TIMx_CCEN 寄存器中的 CC2P='0'，使 TI2FP2 能够检测上升沿。
- 置 TIMx_SMCTRL 寄存器中的 TSEL='110'，TI2FP2 作为从模式控制器的触发（TRGI）。
- 置 TIMx_SMCTRL 寄存器中的 SMSEL='110'（触发模式），TI2FP2 被用来启动计数器。

  ONEPM 波形由写入比较寄存器的数值决定（要考虑时钟频率和计数器预分频器）
- t_{DELAY} 由写入 TIMx_CCDAT1 寄存器中的值定义。
- t_{PULSE} 由自动装载值和比较值之间的差值定义（TIMx_AR - TIMx_CCDAT1）。
- 假定当发生比较匹配时要产生从 0 到 1 的波形，当计数器到达预装载值时要产生一个从 1 到 0 的波形；首先要置 TIMx_CCMOD1 寄存器的 OC1M='111'，进入 PWM 模式 2；根据需要有选择地使能预装载寄存器：置 TIMx_CCMOD1 中的 OC1PEN='1'和 TIMx_CTRL1 寄存器中的 ARPEN；然后在

TIMx_CCDAT1 寄存器中填写比较值，在 TIMx_AR 寄存器中填写自动装载值，修改 UDGN 位来产生一个更新事件，然后等待在 TI2 上的一个外部触发事件。本例中，CC1P='0'。

在这个示例中，TIMx_CTRL1 寄存器中的 DIR 和 CAMSEL 位应该置低。

因为只需一个脉冲，所以必须设置 TIMx_CTRL1 寄存器中的 ONEPM='1'，在下一个更新事件（当计数器从自动装载值翻转到 0）时停止计数。

12.3.10.1 特殊情况：OCx 快速使能：

在单脉冲模式下，在 TIx 输入管脚的边沿检测逻辑设置 CNTEN 位以启动计数器。然后计数器和比较值间的比较操作产生了输出的转换。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY}。如果要以最小延时输出波形，可以设置 TIMx_CCMODx 寄存器中的 OCxFEN 位；此时 OCxREF（和 OCx）被强制响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCxFEN 只在通道配置为 PWM1 和 PWM2 模式时起作用。

12.3.11 在外部事件时清除 OCxREF 信号

对于给定的一个通道，设置 TIMx_CCMODx 寄存器中对应的 OCxCEN 位为'1'，能够用 ETRF 输入端的高电平把 OCxREF 信号拉低，OCxREF 信号将保持为低直到发生下一次的更新事件 UEV。

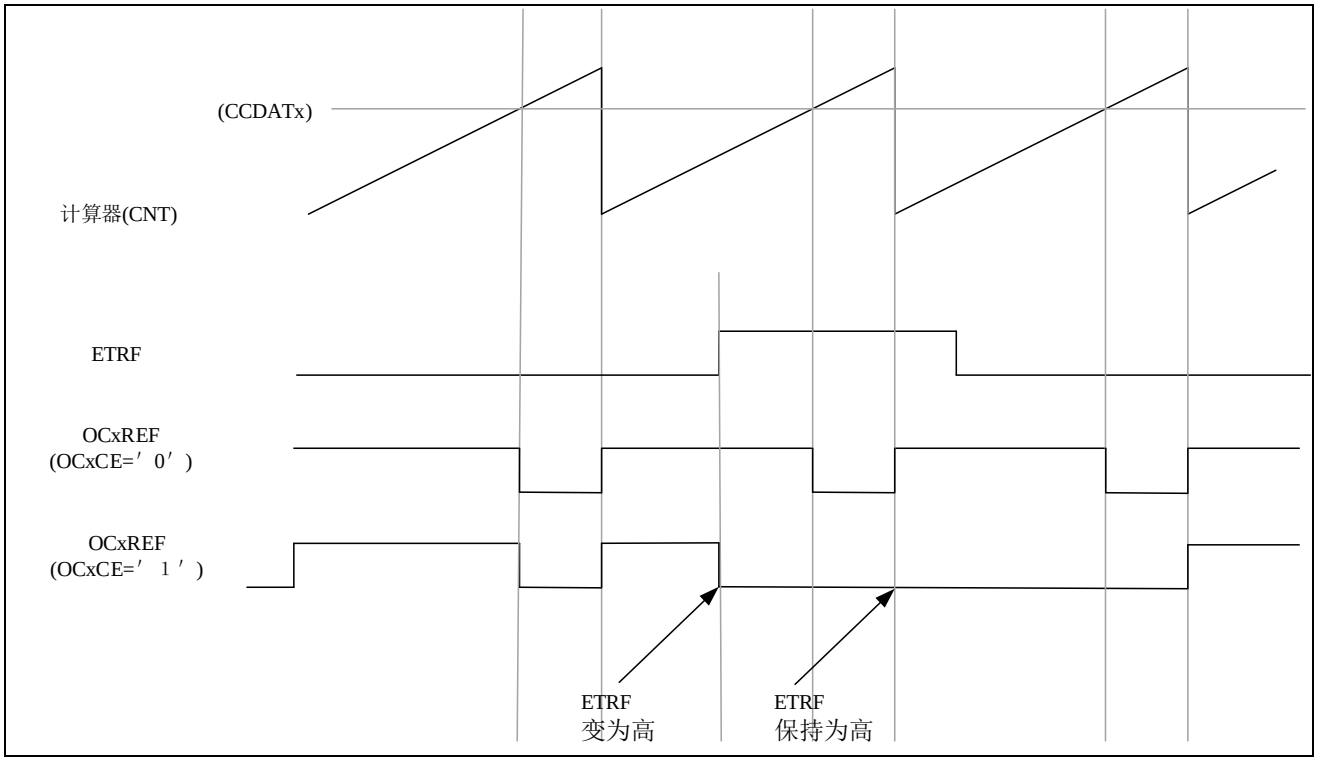
该功能只能用于输出比较和 PWM 模式，而不能用于强置模式。

例如，OCxREF 信号可以连接到一个比较器，用于控制电流。此时，ETR 必须配置如下：

- 外部触发预分频器必须处于关闭：TIMx_SMCTRL 寄存器中的 EXTPS[1:0]='00'。
- 必须禁止外部时钟模式 2：TIMx_SMCTRL 寄存器中的 EXCEN='0'。
- 外部触发极性（EXTP）和外部触发滤波器（EXTF）可以根据需要配置。

下图显示了当 ETRF 输入变为高时，对应不同 OCxCEN 的值，OCxREF 信号的动作。在这个例子中，定时器 TIMx 被置于 PWM 模式。

图 12-34 清除 TIMx 的 OCxREF



12.3.12 编码器接口模式

选择编码器接口模式的方法有:

- 如果计数器只在 TI2 的边沿计数，则置 TIMx_SMCTRL 寄存器中的 SMSEL=001。
- 如果只在 TI1 边沿计数，则置 SMSEL=010。
- 如果计数器同时在 TI1 和 TI2 边沿计数，则置 SMSEL=011。

通过设置 TIMx_CCEN 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性；如果需要，还可以对输入滤波器编程。

两个输入 TI1 和 TI2 用来作为增量编码器的接口。参看表 12-1，假定计数器已经启动 (TIMx_CTRL1 寄存器中的 CNTEN='1')，计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和变相，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数，同时硬件对 TIMx_CTRL1 寄存器的 DIR 位进行相应的设置。不管计数器是依靠 TI1 计数、依靠 TI2 计数或者同时依靠 TI1 和 TI2 计数。在任一输入端 (TI1 或者 TI2) 的跳变都会重新计算 DIR 位。

编码器接口模式基本上相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIMx_AR 寄存器的自动装载值之间连续计数（根据方向，或是 0 到 AR 计数，或是 AR 到 0 计数）。所以在开始计数之前必须配置 TIMx_AR；同样，捕获器、比较器、预分频器、触发输出特性等仍工作如常。

注：编码器模式和外部时钟模式 2 不兼容，因此不能同时操作。

在这个模式下，计数器依照增量编码器的速度和方向被自动的修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器旋转的方向对应。下表列出了所有可能的组合， 假设 TI1 和 TI2 不同时变换。

表 12-1 计数方向与编码器信号的关系

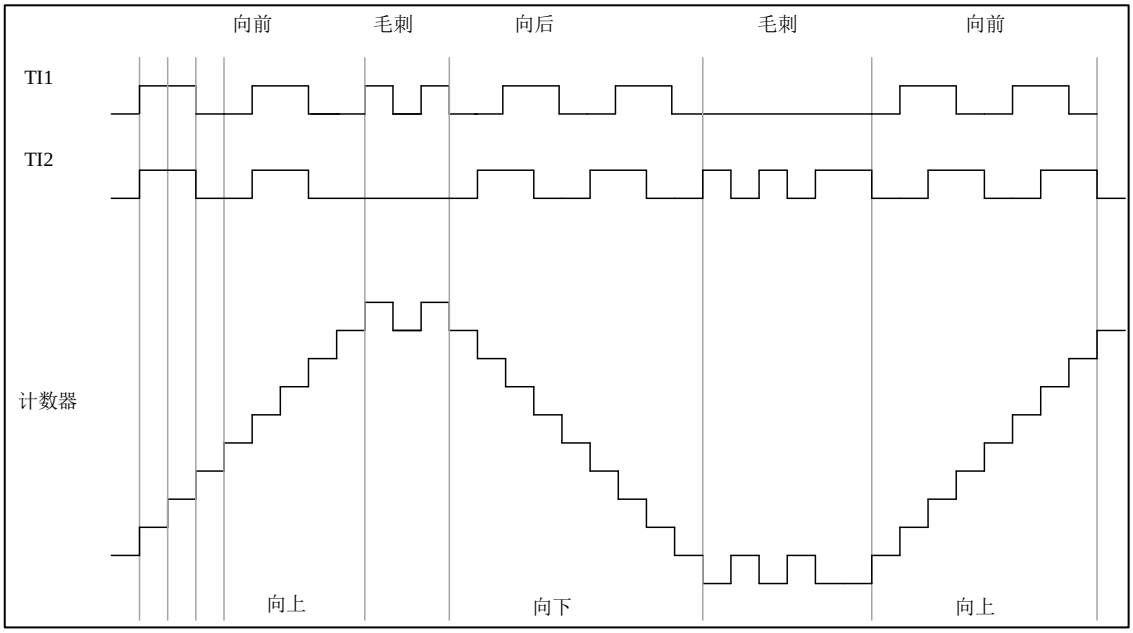
有效边沿	相对信号的电平（TI1FP1对应TI2, TI2FP2对应TI1）	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
仅在TI1计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在TI2计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

一个外部的增量式编码器可以直接与 MCU 管脚连接而不需要增加外部接口逻辑。但是，一般会使用比较器将编码器的差动输出转换到数字信号，这大大增加了抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发一个计数器复位。

下图是一个计数器操作的示例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

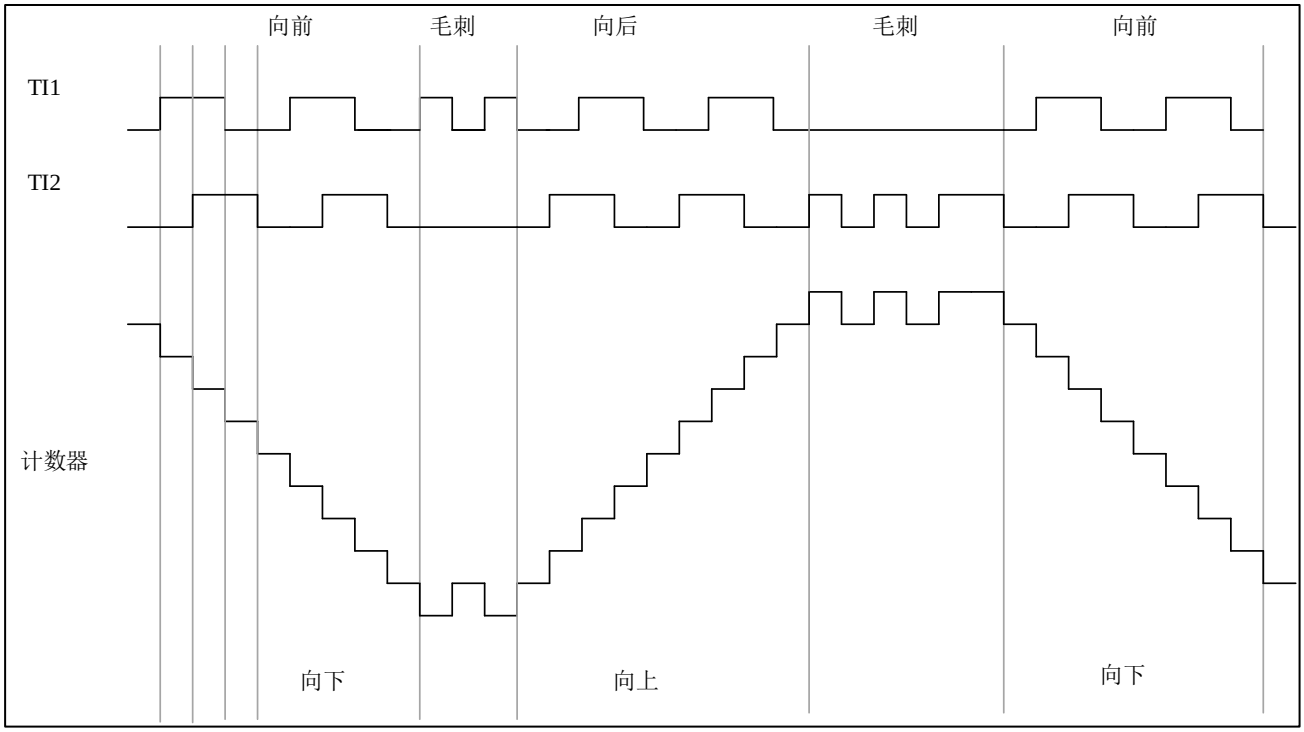
- CC1SEL='01' （TIMx_CCMOD1 寄存器，IC1FP1 映射到 TI1）
- CC2SEL='01' （TIMx_CCMOD2 寄存器，IC2FP2 映射到 TI2）
- CC1P='0' （TIMx_CCEN 寄存器，IC1FP1 不反相，IC1FP1=TI1）
- CC2P='0' （TIMx_CCEN 寄存器，IC2FP2 不反相，IC2FP2=TI2）
- SMSEL='011' （TIMx_SMCTRL 寄存器，所有的输入均在上升沿和下降沿有效）.
- CNTEN='1' （TIMx_CTRL1 寄存器，计数器使能）

图 12-35 编码器模式下的计数器操作实例



下图为当 IC1FP1 极性反相时计数器的操作实例（CC1P='1'，其他配置与上例相同）

图 12-36 IC1FP1 反相的编码器接口模式实例



定时器配置成编码器接口模式时，提供传感器当前位置的信息。使用第二个配置在捕获模式的定时器，可以测量两个编码器事件的间隔，获得动态的信息（速度，加速度，减速度）。指示机械零点的编码器输出可被用做此目的。根据两个事件间的间隔，可以按照固定的时间读出计数器。如果可能的话，可以把计数器的值锁存到第三个输入捕获寄存器（捕获信号必须是周期的并且可以由另一个定时器产

生)；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

12. 3. 13 定时器输入异或功能

通过设置 TIMx_CTRL2 寄存器中的 TI1SEL 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。

异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。上一章 11.3.18 节给出了此特性用于连接霍尔传感器的例子。

12. 3. 14 定时器和外部触发的同步

TIMx 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式和触发模式。

12. 3. 14. 1 从模式：复位模式

输入一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CTRL1 寄存器的 UPRS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器（TIMx_AR，TIMx_CCDATx）都被更新。

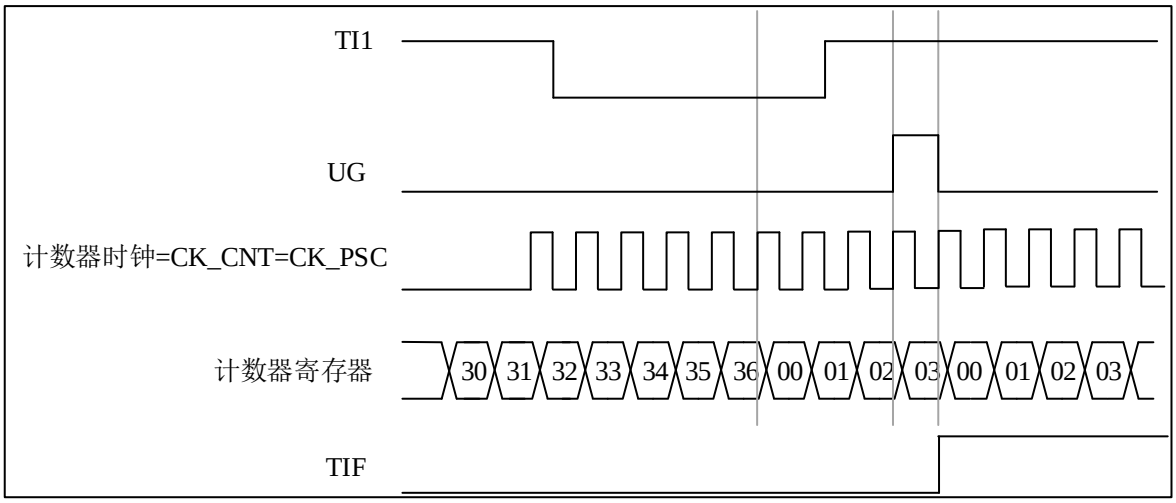
在以下的示例中，TI1 输入端的上升沿事件导致向上计数器被清零：

- 配置通道 1 以检测 TI1 的上升沿。配置输入滤波器的带宽（在本例中，不需要任何滤波器，因此保持 IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置它。CC1SEL 位只选择输入捕获源，即 TIMx_CCMOD1 寄存器中 CC1SEL=01。置 TIMx_CCEN 寄存器中 CC1P=0 以确定极性（只检测上升沿)。
- 置 TIMx_SMCTRL 寄存器中 SMSEL=100，配置定时器为复位模式；置 TIMx_SMCTRL 寄存器中 TSEL=101，选择 TI1 作为输入源。
- 置 TIMx_CTRL1 寄存器中 CNTEN=1，启动计数器。

计数器开始基于内部时钟计数，然后正常运转直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志（TIMx_STS 寄存器中的 TITF 位）被设置，根据 TIMx_DINTEN 寄存器中 TIE（中断使能）位和 TDEN（DMA 使能）位的设置，产生一个中断请求或一个 DMA 请求。

下图显示当自动重装载寄存器 TIMx_AR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时，取决于 TI1 输入端的重同步电路。

图 12-37 复位模式下的控制电路



12.3.14.2 从模式：门控模式

按照选中的输入端电平极性使能计数器。

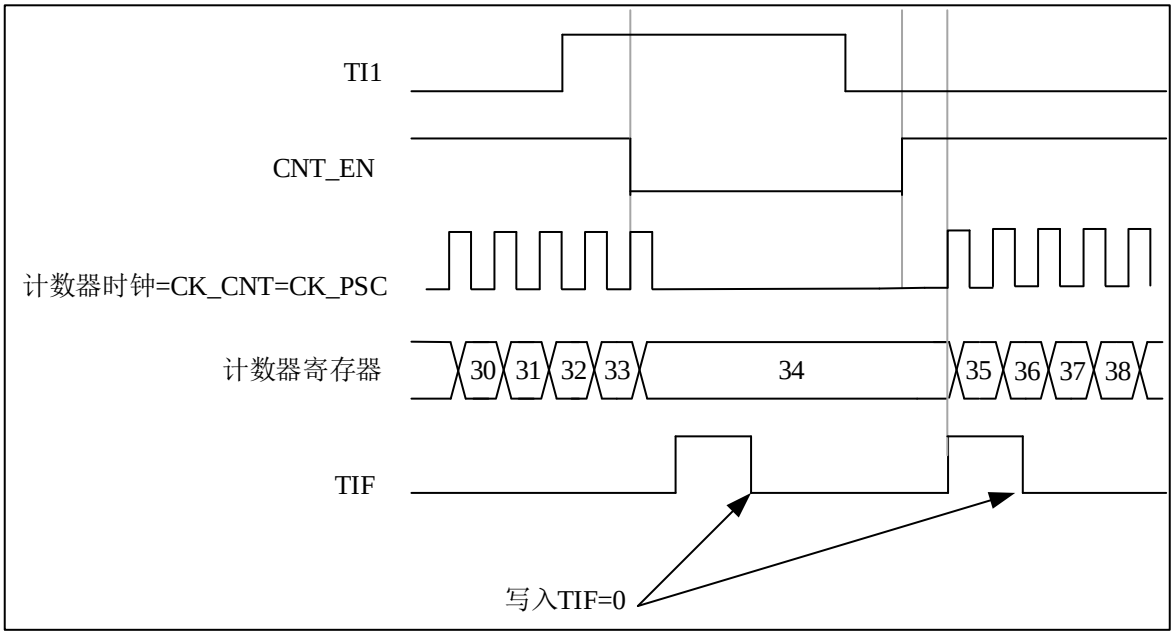
在如下的示例中，计数器只在 TI1 为低时向上计数:

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽（本例中，不需要滤波，所以保持 IC1F=0000）。触发操作中不使用捕获预分频器，所以不需要配置。CC1SEL 位用于选择输入捕获源，置 TIMx_CCMOD1 寄存器中 CC1SEL=01。置 TIMx_CCEN 寄存器中 CC1P=1 以确定极性（只检测低电平）。
- 置 TIMx_SMCTRL 寄存器中 SMSEL=101，配置定时器为门控模式；置 TIMx_SMCTRL 寄存器中 TS=101，选择 TI1 作为输入源。
- 置 TIMx_CTRL1 寄存器中 CNTEN=1，启动计数器。在门控模式下，如果 CNTEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟源计数，一旦 TI1 变高时停止计数。当计数器开始或停止时都设置 TIMx_STS 中的 TITF 标置。

TI1 上升沿和计数器实际停止之间的延时，取决于 TI1 输入端的重同步电路。

图 12-38 门控模式下的控制电路



12.3.14.3 从模式：触发模式

在输入端上选中的事件（上升沿或下降沿）使能计数器。

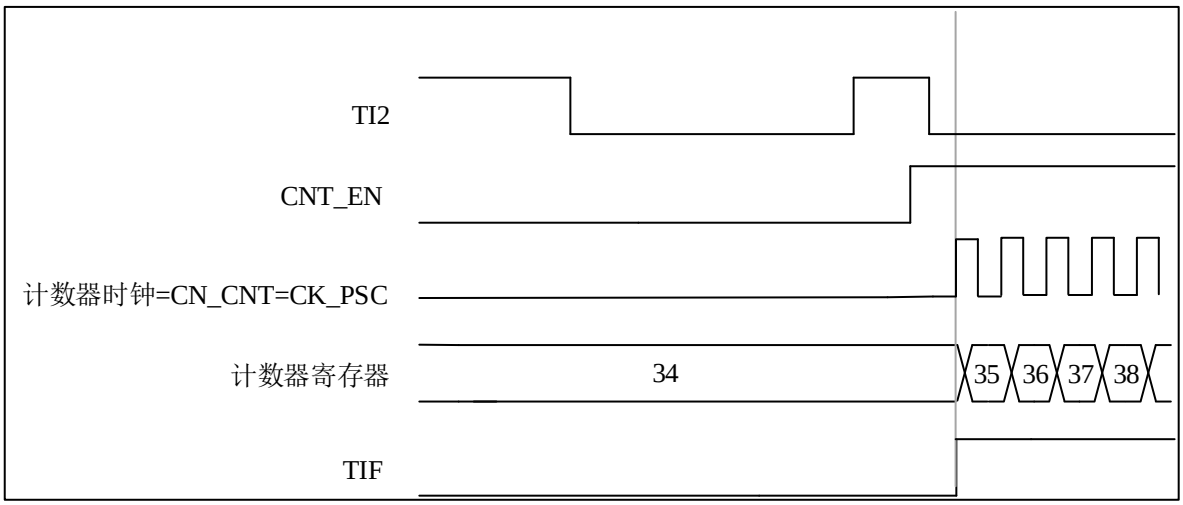
在下面的示例中，计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。CC2SEL 位只用于选择输入捕获源，置 TIMx_CCMOD1 寄存器中 CC2SEL=01。置 TIMx_CCEN 寄存器中 CC2P=1 以确定极性（只检测低电平）。
- 置 TIMx_SMCTRL 寄存器中 SMSEL=110，配置定时器为触发模式；置 TIMx_SMCTRL 寄存器中 TSEL=110，选择 TI2 作为输入源。

  只要 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下进行计数，同时设置 TITF 标志。

  TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

图 12-39 触发器模式下的控制电路



12. 3. 14. 4 从模式：外部时钟模式 2 ＋ 触发模式

外部时钟模式 2 可以与另一种从模式（外部时钟模式 1 和编码器模式除外）一起使用。这时 ETR 信号被用作外部时钟的输入，在复位模式、门控模式或触发模式时可以选择另一个输入作为触发源输入。不建议使用 TIMx_SMCTRL 寄存器的 TSEL 位选择 ETR 作为 TRGI。

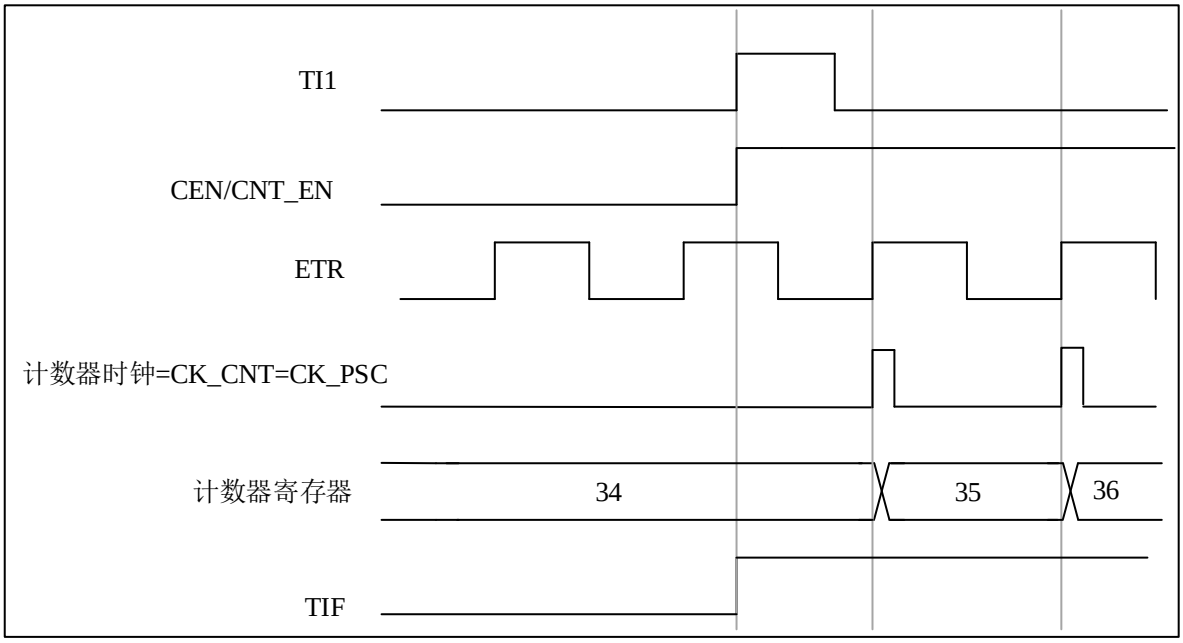
下面的示例中，TI1 上出现一个上升沿之后，计数器即在 ETR 的每一个上升沿向上计数一次：

- 通过 TIMx_SMCTRL 寄存器配置外部触发输入电路：
  - EXTF=0000：没有滤波
  - EXTPS=00：不用预分频器
  - EXTP=0：检测 ETR 的上升沿，置 EXCEN=1 使能外部时钟模式 2
- 按如下配置通道 1，检测 TI 的上升沿：
  - IC1F=0000：没有滤波
  - 触发操作中不使用捕获预分频器，不需要配置
  - 置 TIMx_CCMOD1 寄存器中 CC1SEL=01，选择输入捕获源
  - 置 TIMx_CCEN 寄存器中 CC1P=0 以确定极性（只检测上升沿）
- 置 TIMx_SMCTRL 寄存器中 SMSEL=110，配置定时器为触发模式。置 TIMx_SMCTRL 寄存器中 TSEL=101，选择 TI1 作为输入源。

  当 TI1 上出现一个上升沿时，TITF 标志被设置，计数器开始在 ETR 的上升沿计数。

  ETR 信号的上升沿和计数器实际复位间的延时，取决于 ETRP 输入端的重同步电路。

图 12-40 外部时钟模式 2＋触发模式下的控制电路



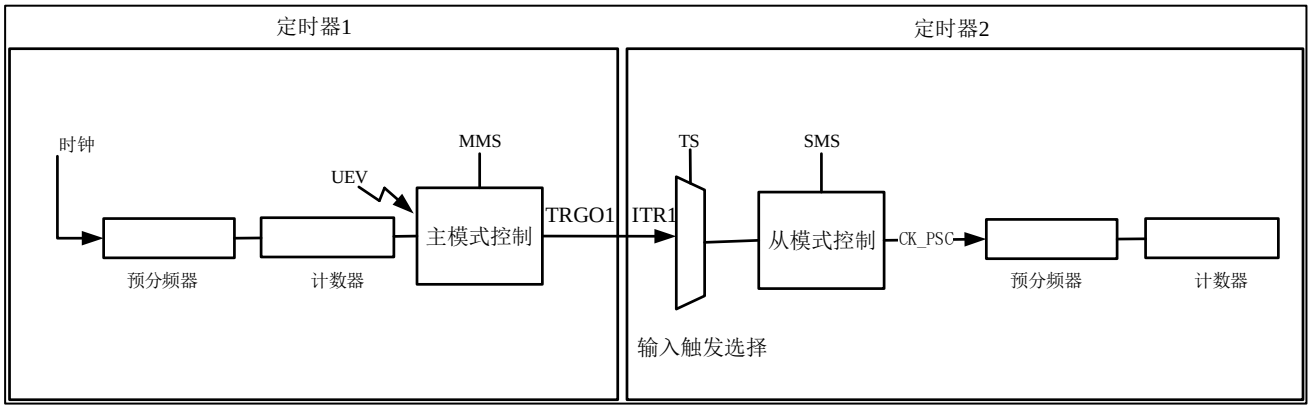
12. 3. 15 定时器同步

所有 TIMx 定时器在内部逻辑相连，用于定时器同步或链接。当一个定时器处于主模式时，它可以对另一个处于从模式的定时器的计数器进行复位、启动、停止或提供时钟等操作。

下图显示了触发选择和主模式选择模块的概况。

12. 3. 15. 1 使用一个定时器作为另一个定时器的预分频器

图 12-41 主/从定时器的例子



如：可以配置定时器 1 作为定时器 2 的预分频器。参考图 12-41 ，进行下述操作:

- 配置定时器 1 为主模式，它可以在每一个更新事件 UEV 时输出一个周期性的触发信号。在 TIM1_CTRL2 寄存器的 MMSEL='010'时，每当产生一个更新事件时在 TRGO1 上输出一个上升沿信号。
- 连接定时器 1 的 TRGO1 输出至定时器 2，设置 TIM2_SMCTRL 寄存器的 TSEL='000'，配置定时器 2

为使用 ITR1 作为内部触发的从模式。

- 然后把从模式控制器置于外部时钟模式 1（TIM2_SMCTRL 寄存器的 SMSEL=111）；这样定时器 2 即可由定时器 1 周期性的上升沿（即定时器 1 的计数器溢出）信号驱动。
- 最后，必须设置相应（TIMx_CTRL1 寄存器）的 CNTEN 位分别启动两个定时器。

注：如果 OCx 已被选中为定时器 1 的触发输出（MMSEL=1xx），它的上升沿用于驱动定时器 2 的计数器。

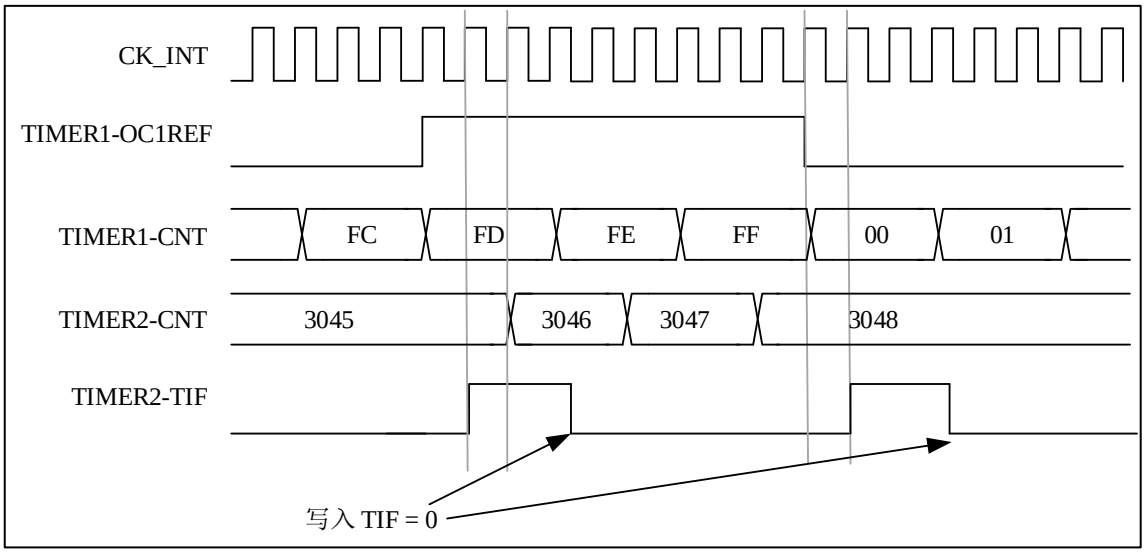
12.3.15.2 使用一个定时器使能另一个定时器

在本示例中，定时器 2 的使能由定时器 1 的输出比较控制。参考图 12-41 的连接。只当定时器 1 的 OC1REF 为高时，定时器 2 才对分频后的内部时钟计数。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3（$f_{CK\_CNT}=f_{CK\_INT}/3$）得到。

- 配置定时器 1 为主模式，送出它的输出比较参考信号（OC1REF）为触发输出（TIM1_CTRL2 寄存器的 MMSEL=100）
- 配置定时器 1 的 OC1REF 波形（TIM1_CCMOD1 寄存器）
- 配置定时器 2 从定时器 1 获得输入触发（TIM2_SMCTRL 寄存器的 TSEL=000）
- 配置定时器 2 为门控模式（TIM2_SMCTRL 寄存器的 SMSEL=101）
- 置 TIM2_CTRL1 寄存器的 CNTEN=1 以使能定时器 2
- 置 TIM1_CTRL1 寄存器的 CNTEN=1 以启动定时器 1

注：定时器 2 的时钟不与定时器 1 的时钟同步，这个模式只影响定时器 2 计数器的使能信号。

图 12-42 定时器 1 的 OC1REF 控制定时器 2



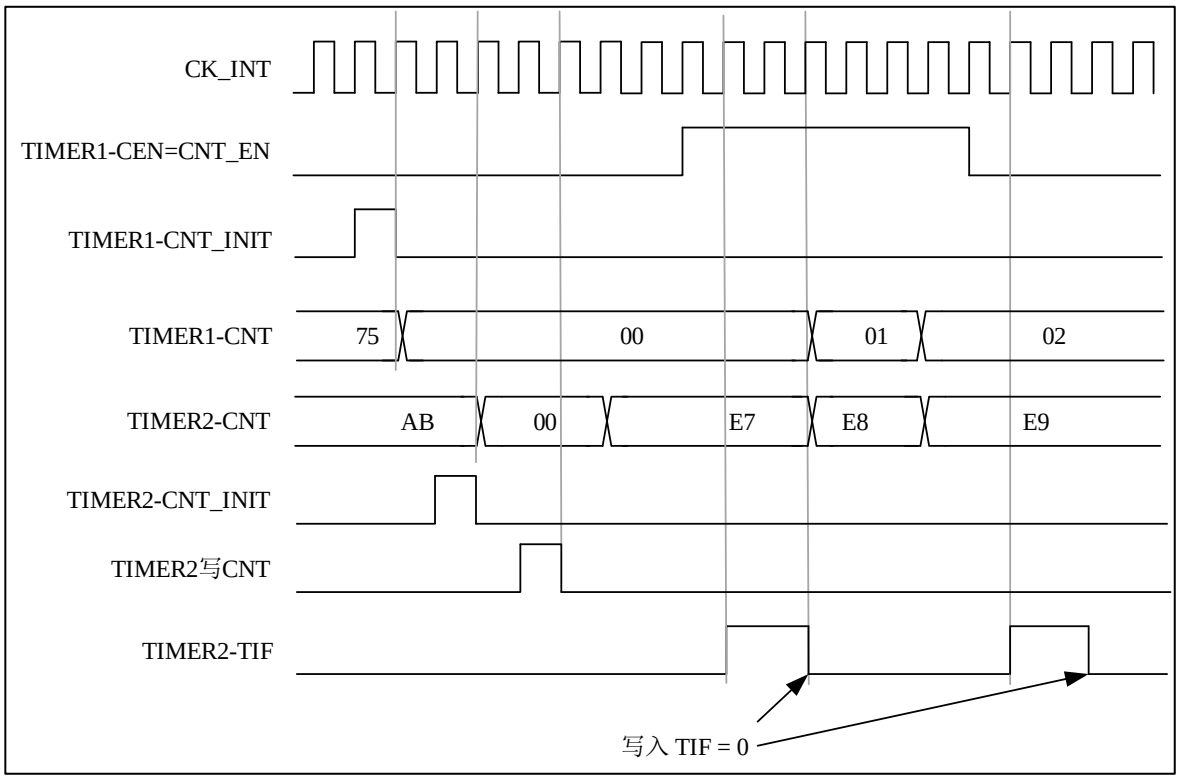
在图 12-42 的示例中，在定时器 2 启动之前，它们的计数器和预分频器未被初始化，因此它们从当前的数值开始计数。可以在启动定时器 1 之前复位 2 个定时器，使它们从给定的数值开始，即在定时器计数器中写入需要的任意数值。写 TIMx_EVTGEN 寄存器的 UDGN 位即可复位定时器。

在下一个示例中，需要同步定时器 1 和定时器 2。定时器 1 是主模式并从 0 开始，定时器 2 是从模

式并从 0xE7 开始; 2 个定时器的预分频器系数相同。写'0'到 TIM1_CTRL1 的 CNTEN 位将禁止定时器 1，定时器 2 随即停止。

- 配置定时器 1 为主模式，送出输出比较 1 参考信号（OC1REF）做为触发输出（TIM1_CTRL2 寄存器 MMSEL=100）。
- 配置定时器 1 的 OC1REF 波形（TIM1_CCMOD1 寄存器）。
- 配置定时器 2 从定时器 1 获得输入触发（TIM2_SMCTRL 寄存器的 TSEL=000）。
- 配置定时器 2 为门控模式（TIM2_SMCTRL 寄存器的 SMSEL=101）
- 置 TIM1_EGR 寄存器的 UDGN='1'，复位定时器 1。
- 置 TIM2_EGR 寄存器的 UDGN='1'，复位定时器 2。
- 写'0xE7'至定时器 2 的计数器（TIM2_CNTL），初始化为 0xE7。
- 置 TIM2_CTRL1 寄存器的 CNTEN='1'以使能定时器 2。
- 置 TIM1_CTRL1 寄存器的 CNTEN='1'以启动定时器 1。
- 置 TIM1_CTRL1 寄存器的 CNTEN='0'以停止定时器 1。

图 12-43 通过使能定时器 1 可以控制定时器 2

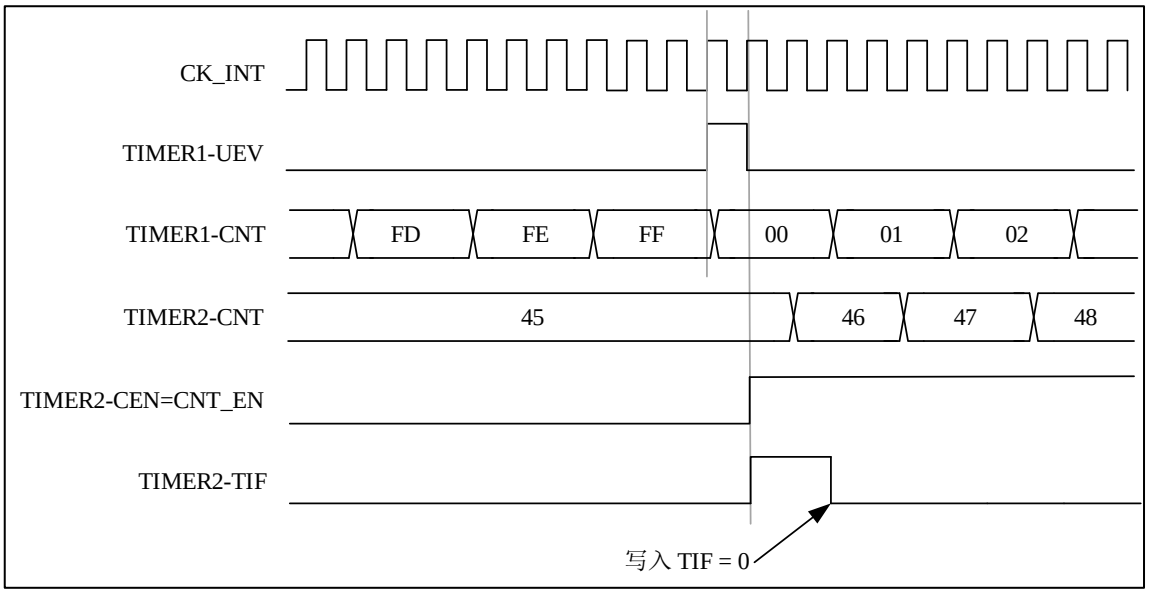


12. 3. 15. 3 使用一个定时器去启动另一个定时器

在本例子中，使用定时器 1 的更新事件使能定时器 2。参考图 12-41 的连接。一旦定时器 1 产生更新事件，定时器 2 即从它当前的数值（可以是非 0）按照分频的内部时钟开始计数。在收到触发信号时，定时器 2 的 CNTEN 位被自动置'1'，同时计数器开始计数直到写'0'到 TIM2_CTRL1 寄存器的 CNTEN 位。两个定时器的时钟频率都是由预分频器对 CK_INT 除以 3（$f_{CK\_CNT}=f_{CK\_INT}/3$）。

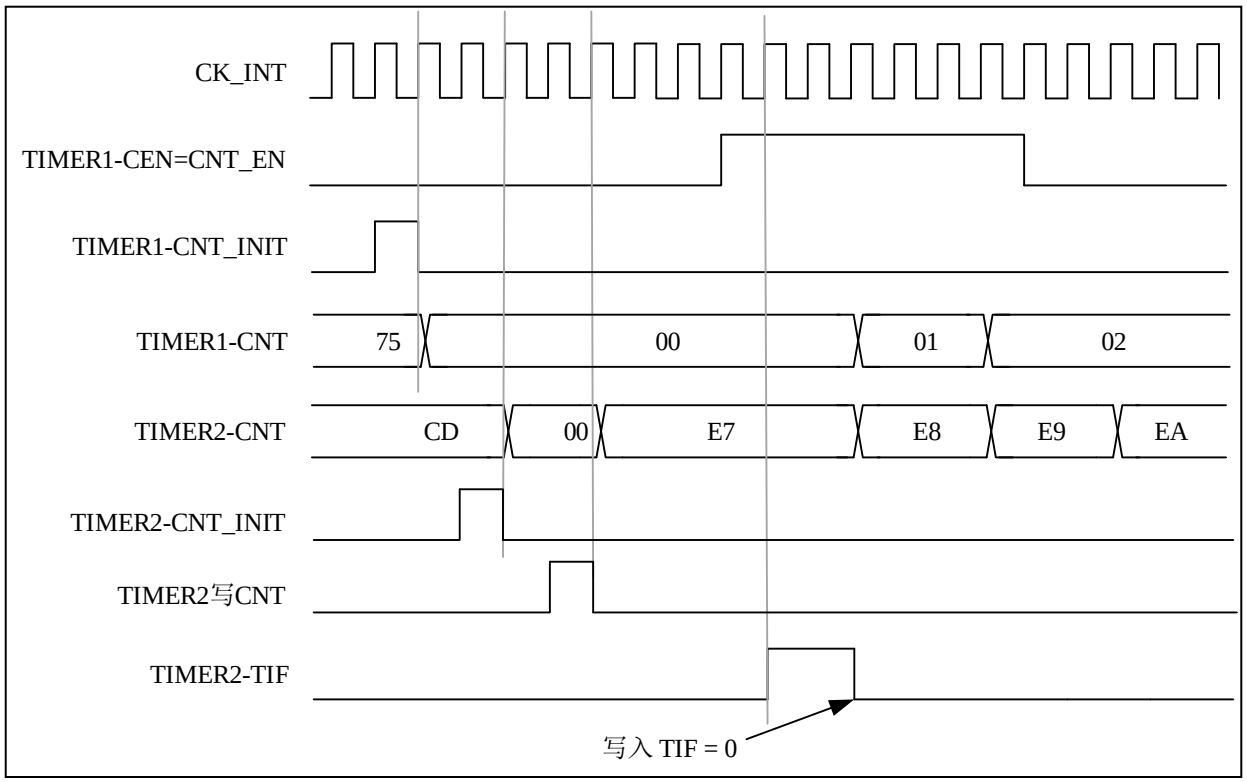
- 配置定时器 1 为主模式，送出它的更新事件（UEV）做为触发输出（TIM1_CTRL2 寄存器的 MMSEL=010）。
- 配置定时器 1 的周期（TIM1_AR 寄存器）。
- 配置定时器 2 从定时器 1 获得输入触发（TIM2_SMCTRL 寄存器的 TSEL=000）。
- 配置定时器 2 为触发模式（TIM2_SMCTRL 寄存器的 SMSEL=110）
- 配置 TIM1_CTRL1 寄存器的 CNTEN=1 以启动定时器 1。

图 12-44 使用定时器 1 的更新触发定时器 2



在上一个例子中，可以在启动计数之前初始化两个计数器。图 12-45 显示在与 0 相同配置情况下，使用触发模式而不是门控模式（TIM2_SMCTRL 寄存器的 SMSEL=110）的动作。

图 12-45 利用定时器 1 的使能触发定时器 2



12.3.15.4 使用一个定时器作为另一个的预分频器

这个例子使用定时器 1 作为定时器 2 的预分频器。参考图 12-41 的连接，配置如下:

- 配置定时器 1 为主模式，送出它的更新事件 UEV 做为触发输出（TIM1_CTRL2 寄存器的 MMSEL='010'）。然后每次计数器溢出时输出一个周期信号。
- 配置定时器 1 的周期（TIM1_AR 寄存器）。
- 配置定时器 2 从定时器 1 获得输入触发（TIM2_SMCTRL 寄存器的 TSEL=000）
- 配置定时器 2 使用外部时钟模式（TIM2_SMCTRL 寄存器的 SMSEL=111）
- 配置 TIM1_CTRL2 寄存器的 CNTEN=1 以启动定时器 2。
- 配置 TIM1_CTRL1 寄存器的 CNTEN=1 以启动定时器 1。

12.3.15.5 使用一个外部触发同步地启动 2 个定时器

这个例子中当定时器 1 的 TI1 输入上升时使能定时器 1，使能定时器 1 的同时使能定时器 2，参见图 12-41。为保证计数器的对齐，定时器 1 必须配置为主/从模式（对应 TI1 为从，对应定时器 2 为主）:

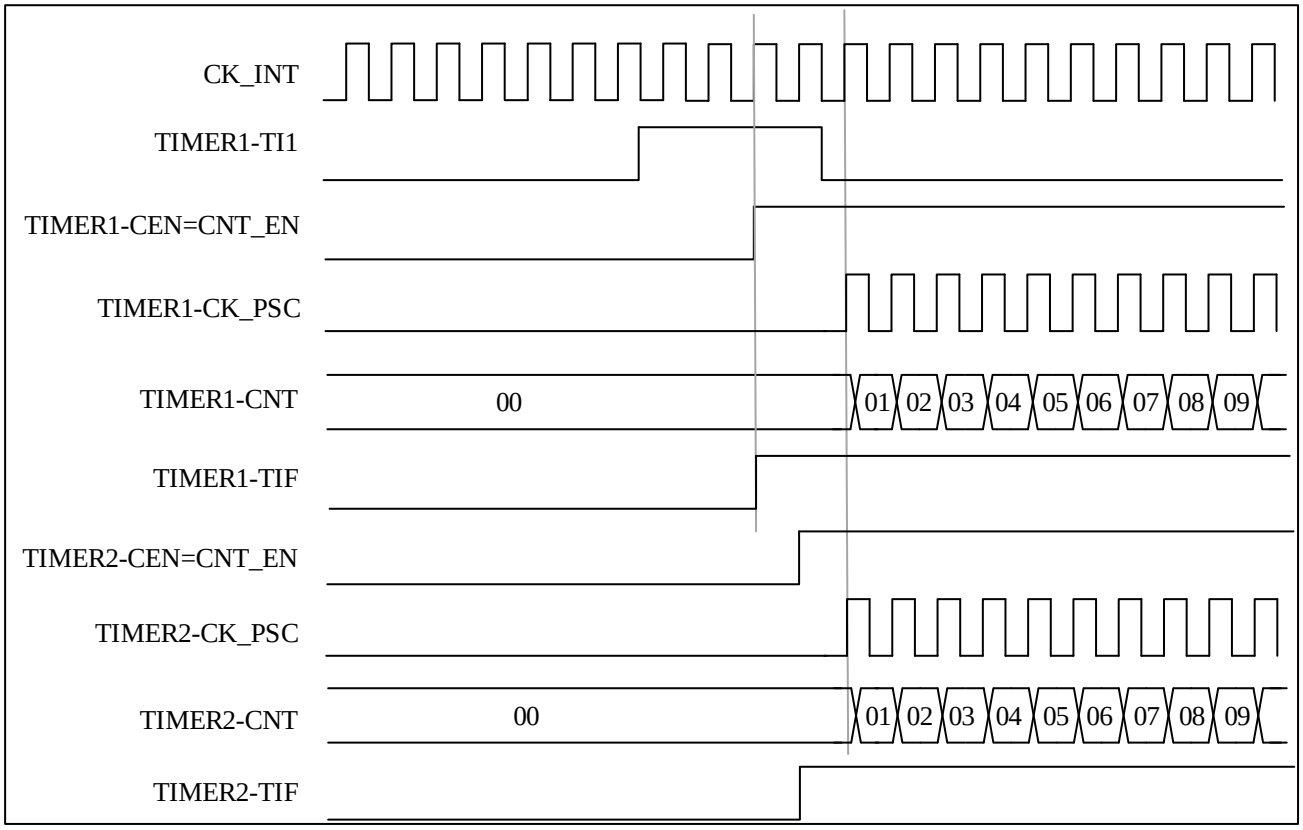
- 配置定时器 1 为主模式，送出它的使能做为触发输出（TIM1_CTRL2 寄存器的 MMSEL='001'）。
- 配置定时器 1 为从模式，从 TI1 获得输入触发（TIM1_SMCTRL 寄存器的 TSEL='100'）。
- 配置定时器 1 为触发模式（TIM1_SMCTRL 寄存器的 SMSEL='110'）。
- 配置定时器 1 为主/从模式，TIM1_SMCTRL 寄存器的 MSMD='1'。
- 配置定时器 2 从定时器 1 获得输入触发（TIM2_SMCTRL 寄存器的 TSEL=000）

- 配置定时器 2 为触发模式（TIM2_SMCTRL 寄存器的 SMSEL='110'）。

当定时器 1 的 TI1 上出现一个上升沿时，两个定时器同步地按照内部时钟开始计数，两个 TITF 标志也同时被设置。

注：在这个例子中，在启动之前两个定时器都被初始化（设置相应的 UDGN 位），两个计数器都从 0 开始， 但可以通过写入任意一个计数器寄存器（TIMx_CNT）在定时器间插入一个偏移。下图中能看到主/ 从模式下在定时器 1 的 CNT_EN 和 CK_PSC 之间有个延迟。

图 12-46 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2



12.3.16 调试模式

当微控制器进入调试模式（N308 核心停止），根据 DBG 模块中 DBG_TIMx_STOP 的设置，TIMx 计数器或者继续正常操作，或者停止。详见随后第 29.3.2 节： 外设调试支持。

12.4 TIMx 寄存器描述

关于在寄存器描述里面所用到的缩写，详见第 1.1 节。

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

12.4.1 TIMx 寄存器图

下表中将 TIMx 的所有寄存器映射到一个 16 位可寻址（编址）空间。

表 12-2 TIMx –寄存器图和复位

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	TIMx_CTRL1	Reserved																CLRSEL	C4SEL	C3SEL	C2SEL	C1SEL	Reserved	CLKD[1:0]		ARPEN	CAMSEL[1:0]		DIR	ONEPM	UPRS	UPDIS	CNTEN
	Reset Value																	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0
004h	TIMx_CTRL2	Reserved																							ETRSEL	TI1SEL	MMSEL[2:0]			CCDSEL	Reserved		
	Reset Value																								0	0	0	0	0	0			
008h	TIMx_SMCTRL	Reserved																EXTP	EXCEN	EXTPS[1:0]		EXTF[3:0]				MSMD	TSEL[2:0]			Reserved	SMSELEL[2:0]		
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
00Ch	TIMx_DINTEN	Reserved																	TDEN	Reserved	CC4DEN	CC3DEN	CC2DEN	CC1DEN	UDEN	Reserved	TIEN	Reserved	CC4IEN	CC3IEN	CC2IEN	CC1IEN	UIEN
	Reset Value																		0		0	0	0	0	0		0		0	0	0	0	0
010h	TIMx_STS	Reserved																			CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved		TITF	Reserved	CC4ITF	CC3ITF	CC2ITF	CC1ITF	UDITF
	Reset Value																				0	0	0	0			0		0	0	0	0	0
014h	TIMx_EVTGEN	Reserved																									TGN	Reserved	CC4GN	CC3GN	CC2GN	CC1GN	UDGN
	Reset Value																										0		0	0	0	0	0
018h	TIMx_CCMOD1 输出比较模式	Reserved																OC2CEN	OC2M[2:0]			OC2PEN	OC2FEN	CC2SEL[1:0]		OC1CEN	OC1M[2:0]			OC1PEN	OC1FEN	CC1SEL[1:0]	
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMOD1 输入捕获模式	Reserved																IC2F[3:0]				IC2PSC[1:0]		CC2SEL[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1SEL[1:0]	
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	TIMx_CCMOD2 输出比较模式	Reserved																OC4CEN	OC4M[2:0]			OC4PEN	OC4FEN	CC4SEL[1:0]		OC3CEN	OC3M[2:0]			OC3PEN	OC3FEN	CC3SEL[1:0]	
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	TIMx_CCMOD2 输入捕获模式	Reserved																IC4F[3:0]				IC4PSC[1:0]		CC4SEL[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3SEL[1:0]	
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
020h	TIMx_CCEN	Reserved																		CC4P	CC4EN	Reserved		CC3P	CC3EN	Reserved		CC2P	CC2EN	Reserved		CC1P	CC1EN
	Reset Value																			0	0			0	0			0	0			0	0
024h	TIMx_CNT	Reserved																CNT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028h	TIMx_PSC	Reserved																PSC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	TIMx_AR	Reserved																AR[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
030h	Reserved																																

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
034h	TIMx_CCDAT1	Reserved																CCDAT1[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
038h	TIMx_CCDAT2	Reserved																CCDAT2[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03Ch	TIMx_CCDAT3	Reserved																CCDAT3[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
040h	TIMx_CCDAT4	Reserved																CCDAT4[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
044h	Reserved																																
048h	TIMx_DCTRL	Reserved																DBLEN[4:0]					Reserved			DBADDR[4:0]							
	Reset Value																	0	0	0	0	0				0	0	0	0	0			
04Ch	TIMx_DADDR	Reserved																BURST[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

12.4.2 控制寄存器 1（TIMx_CTRL1）

偏移地址：0x00
复位值：0x0000

31															16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRSEL	C4SEL	C3SEL	C2SEL	C1SEL	Reserved	CLKD[1:0]		ARPEN	CAMSEL[1:0]		DIR	ONEPM	UPRS	UPDIS	CNTEN
rw	rw	rw	rw	rw		rw		rw	rw		rw	rw	rw	rw	rw

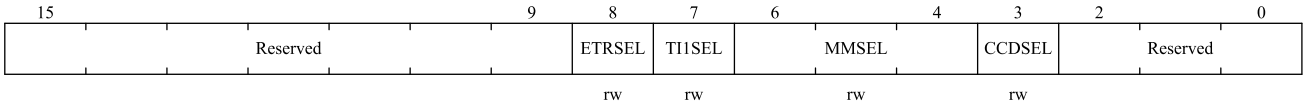
位域	名称	描述
31:16	Reserved	保留，硬件强制为0。
15	CLRSEL	OCxRef选择（OCxRef selection） *注：操作此位前必须先开启芯片的扩展模式(置位EXMODE)* 0：选择外部Ocxclr （ETR） 信号 1：选择内部OCxclr（来自比较器）信号 *注：TIM5置1无效*
14	C4SEL	通道四选择（Channel 4 selection） *注：操作此位前必须先开启芯片的扩展模式(置位EXMODE)* 0：选择外部CH4（来自IOM）信号 1：选择内部CH4（来自COMP）信号 *注：TIM5置1无效*
13	C3SEL	通道三选择（Channel 3 selection） *注：操作此位前必须先开启芯片的扩展模式(置位EXMODE)* 0：选择外部CH3（来自IOM）信号 1：选择内部CH3（来自COMP）信号
12	C2SEL	通道二选择（Channel 2 selection） *注：操作此位前必须先开启芯片的扩展模式(置位EXMODE)* 0：选择外部CH2（来自IOM）信号 1：选择内部CH2（来自COMP）信号

位域	名称	描述
11	C1SEL	通道一选择（Channel 1 selection） *注：操作此位前必须先开启芯片的扩展模式(置位EXMODE)* 0：选择外部CH1（来自IOM）信号 1：选择内部CH1（来自COMP）信号
10	Reserved	保留，硬件强制为0。
9:8	CLKD[1:0]	时钟分频因子 （Clock division） 这2位定义在定时器时钟（CK_INT）频率、死区时间和由死区发生器与数字滤波器（ETR,TIx）所用的采样时钟之间的分频比例。 00：tDTS = tCK_INT 01：tDTS = 2 x tCK_INT 10：tDTS = 4 x tCK_INT 11：保留，不要使用这个配置
7	ARPEN	自动重装载预装载允许位 （Auto-reload preload enable） 0：TIMx_AR寄存器没有缓冲； 1：TIMx_AR寄存器被装入缓冲器。
6:5	CAMSEL[1:0]	选择中央对齐模式 （Center-aligned mode selection） 00：边沿对齐模式。计数器依据方向位（DIR）向上或向下计数。 01：中央对齐模式1。计数器交替地向上和向下计数。配置为输出的通道（TIMx_CCMODx寄存器中CCxSEL=00）的输出比较中断标志位，只在计数器向下计数时被设置。 10：中央对齐模式2。计数器交替地向上和向下计数。配置为输出的通道（TIMx_CCMODx寄存器中CCxSEL=00）的输出比较中断标志位，只在计数器向上计数时被设置。 11：中央对齐模式3。计数器交替地向上和向下计数。配置为输出的通道（TIMx_CCMODx寄存器中CCxSEL=00）的输出比较中断标志位，在计数器向上和向下计数时均被设置。 *注：在计数器开启时（CNTEN=1），不允许从边沿对齐模式转换到中央对齐模式。*
4	DIR	方向 （Direction） 0：计数器向上计数； 1：计数器向下计数。 *注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。*
3	ONEPM	单脉冲模式 （One pulse mode） 0：在发生更新事件时，计数器不停止； 1：在发生下一次更新事件（清除CNTEN位）时，计数器停止。
2	UPRS	更新请求源 （Update request source） 软件通过该位选择UEV事件的源 0：如果使能了更新中断或DMA请求，则下述任一事件产生更新中断或DMA请求： – 计数器溢出/下溢 – 设置UDGN位 – 从模式控制器产生的更新 1：如果使能了更新中断或DMA请求，则只有计数器溢出/下溢才产生更新中断或DMA请求。

位域	名称	描述
1	UPDIS	禁止更新 （Update disable） 软件通过该位允许/禁止UEV事件的产生 0：允许UEV。更新（UEV）事件由下述任一事件产生： – 计数器溢出/下溢 – 设置UDGN位 – 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值。（译注：更新影子寄存器） 1：禁止UEV。不产生更新事件，影子寄存器（AR、PSC、CCDATx）保持它们的值。如果设置了 UDGN位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CNTEN	使能计数器 （Counter enable） 0：禁止计数器； 1：使能计数器。 *注：在软件设置了CNTEN位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CNTEN位。*

12.4.3 控制寄存器 2（TIMx_CTRL2）

偏移地址：0x04
复位值：0x0000



位域	名称	描述
15:9	Reserved	保留，硬件强制为0。
8	ETRSEL	外部触发选择寄存（ETR Selection） *注：操作此位前必须先开启芯片的扩展模式(置位EXMODE)* 0：选择外部ETR（来自IOM）信号； 1：选择内部ETR（来自TSC）信号； *注：TIM3有此功能，TIM2和TIM4置1无效，TIM5无外部ETR功能此位需置1.*
7	TI1SEL	TI1选择 （TI1 selection） 0：TIMx_CH1引脚连到TI1输入； 1：TIMx_CH1、TIMx_CH2和TIMx_CH3引脚经异或后连到TI1输入。
6:4	MMSEL[2:0]	主模式选择 （Master mode selection） 这3位用于选择在主模式下送到从定时器的同步信息（TRGO）。可能的组合如下： 000：**复位** – TIMx_EVTGEN寄存器的UDGN位被用于作为触发输出（TRGO）。如果是触发输入产生的复位（从模式控制器处于复位模式），则TRGO上的信号相对实际的复位会有一个延迟。 001：**使能** – 计数器使能信号CNTEN被用于作为触发输出（TRGO）。有时需要在同一时间启动多个定时器或控制在一段时间内使能从定时器。计数器使能信号是通过CNTEN控制位和门控模式下的触发输入信号的逻辑或产生。当计数器使能信号受控于触发输入时，TRGO上会有一个延迟，除非选择了主/从模式（见TIMx_SMCTRL寄存器中MSMD位

位域	名称	描述
		的描述)。 010：**更新** – 更新事件被选为触发输入（TRGO)。例如，一个主定时器的时钟可以被用作一个从定时器的预分频器。 011：**比较脉冲** – 在发生一次捕获或一次比较成功时，当要设置CC1ITF标志时（即使它已经为高)，触发输出送出一个正脉冲（TRGO)。 100：**比较** – OC1REF信号被用于作为触发输出（TRGO)。 101：**比较** – OC2REF信号被用于作为触发输出（TRGO)。 110：**比较** – OC3REF信号被用于作为触发输出（TRGO)。 111：**比较** – OC4REF信号被用于作为触发输出（TRGO)。
3	CCDSEL	捕获/比较的DMA选择 （Capture/compare DMA selection） 0：当发生CCx事件时，送出CCx的DMA请求; 1：当发生更新事件时，送出CCx的DMA请求。
2:0	Reserved	保留，硬件强制为0。

12.4.4 从模式控制寄存器（TIMx_SMCTRL）

偏移地址：0x08

复位值：0x0000

15	14	13 12	11 8	7	6 4	3	2 0
EXTP	EXCEN	EXTPS[1:0]	EXTF[3:0]	MSMD	TSEL[2:0]	Reserved	SMSEL[2:0]
rw	rw	rw	rw	rw	rw		rw

位域	名称	描述
15	EXTP	外部触发极性 （External trigger polarity） 该位选择是用ETR还是ETR的反相来作为触发操作 0：ETR不反相，高电平或上升沿有效; 1：ETR被反相，低电平或下降沿有效。
14	EXCEN	外部时钟使能位 （External clock enable） 该位启用外部时钟模式2 0：禁止外部时钟模式2; 1：使能外部时钟模式2。计数器由ETRF信号上的任意有效边沿驱动。 *注1：设置EXCEN位与选择外部时钟模式1并将TRGI连到ETRF（SMSEL=111和TSEL=111）具有相同功效。* *注2：下述从模式可以与外部时钟模式2同时使用：复位模式，门控模式和触发模式；但是，这时TRGI不能连到ETRF（TSEL位不能是'111')。* *注3：外部时钟模式1和外部时钟模式2同时被使能时，外部时钟的输入是ETRF。*
13:12	EXTPS[1:0]	外部触发预分频 （External trigger prescaler） 外部触发信号ETRP的频率必须最多是TIMxCLK频率的1/4。当输入较快的外部时钟时，可以使用预分频降低ETRP的频率。 00：关闭预分频; 01：ETRP频率除以2; 10：ETRP频率除以4; 11：ETRP频率除以8。

位域	名称	描述
11:8	EXTF[3:0]	外部触发滤波 （External trigger filter） 这些位定义了对ETRP信号采样的频率和对ETRP数字滤波的带宽。实际上，数字滤波器是一个事件计数器，它记录到N个事件后会产生一个输出的跳变。 0000：无滤波器，以f_{DTS}采样 1000：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=6 0001：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=2 1001：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=8 0010：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=4 1010：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=5 0011：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=8 1011：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=6 0100：采样频率$f_{SAMPLING}=f_{DTS}/2$，N=6 1100：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=8 0101：采样频率$f_{SAMPLING}=f_{DTS}/2$，N=8 1101：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=5 0110：采样频率$f_{SAMPLING}=f_{DTS}/4$，N=6 1110：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=6 0111：采样频率$f_{SAMPLING}=f_{DTS}/4$，N=8 1111：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=8
7	MSMD	主/从模式 （Master/slave mode） 0：无作用； 1：触发输入（TRGI）上的事件被延迟了，以允许在当前定时器（通过TRGO）与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
6:4	TSEL[2:0]	触发选择 （Trigger selection） 这3位选择用于同步计数器的触发输入。 000：内部触发0（ITR0），TIM1 100：TI1的边沿检测器（TI1F_ED） 001：内部触发1（ITR1），TIM2 101：滤波后的定时器输入1（TI1FP1）)10：内部触发2（ITR2），TIM3 110：滤波后的定时器输入2（TI2FP2） 011：内部触发3（ITR3），TIM4 111：外部触发输入（ETRF） 关于每个定时器中ITRx的细节，参见表 12-3。 *注：这些位只能在未用到（如SMS=000）时被改变，以避免在改变时产生错误的边沿检测。*
3	Reserved	保留，硬件强制为0。
2:0	SMSEL[2:0]	从模式选择 （Slave mode selection） 当选择了外部信号，触发信号（TRGI）的有效边沿与选中的外部输入极性相关（见输入控制寄存器和控制寄存器的说明） 000：关闭从模式 – 如果CNTEN=1，则预分频器直接由内部时钟驱动。 001：编码器模式1 – 根据TI1FP1的电平，计数器在TI2FP2的边沿向上/下计数。 010：编码器模式2 – 根据TI2FP2的电平，计数器在TI1FP1的边沿向上/下计数。 011：编码器模式3 – 根据另一个信号的输入电平，计数器在TI1FP1和TI2FP2的边沿向上/下计数。 100：复位模式 – 选中的触发输入（TRGI）的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。 101：门控模式 – 当触发输入（TRGI）为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止（但不复位）。计数器的启动和停止都是受控的。 110：触发模式 – 计数器在触发输入TRGI的上升沿启动（但不复位），只有计数器的启动是受控的。 111：外部时钟模式1 – 选中的触发输入（TRGI）的上升沿驱动计数器。 *注：如果TI1F_EN被选为触发输入（TSEL=100）时，不要使用门控模式。这是因为，TI1F_ED在每次TI1F变化时输出一个脉冲，然而门控模式是要检查触发输入的*

位域	名称	描述
		电平。

表 12-3 TIMx 内部触发连接[1]

从定时器	ITR0 （TSEL = 000）	ITR1 （TSEL = 001）	ITR2 （TSEL = 010）	ITR3 （TSEL = 011）
TIM2	TIM1	TIM8	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM2	TIM3	TIM4	TIM8

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

12.4.5 DMA/中断使能寄存器（TIMx_DINTEN）

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDEN	Reserved	CC4DEN	CC3DEN	CC2DEN	CC1DEN	UDEN	Reserved	TIEN	Reserved	CC4IEN	CC3IEN	CC2IEN	CC1IEN	UIEN
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位域	名称	描述
15	Reserved	保留，硬件强制为0。
14	TDEN	允许触发DMA请求 （Trigger DMA request enable） 0：禁止触发DMA请求； 1：允许触发DMA请求。
13	Reserved	保留，硬件强制为0。
12	CC4DEN	允许捕获/比较4的DMA请求 （Capture/Compare 4 DMA request enable） 0：禁止捕获/比较4的DMA请求； 1：允许捕获/比较4的DMA请求。

位域	名称	描述
11	CC3DEN	允许捕获/比较3的DMA请求 （Capture/Compare 3 DMA request enable） 0：禁止捕获/比较3的DMA请求； 1：允许捕获/比较3的DMA请求。
10	CC2DEN	允许捕获/比较2的DMA请求 （Capture/Compare 2 DMA request enable） 0：禁止捕获/比较2的DMA请求； 1：允许捕获/比较2的DMA请求。
9	CC1DEN	允许捕获/比较1的DMA请求 （Capture/Compare 1 DMA request enable） 0：禁止捕获/比较1的DMA请求； 1：允许捕获/比较1的DMA请求。
8	UDEN	允许更新的DMA请求 （Update DMA request enable） 0：禁止更新的DMA请求； 1：允许更新的DMA请求。
7	Reserved	保留，硬件强制为0。
6	TIEN	触发中断使能 （Trigger interrupt enable） 0：禁止触发中断； 1：使能触发中断。
5	Reserved	保留，硬件强制为0。
4	CC4IEN	允许捕获/比较4中断 （Capture/Compare 4 interrupt enable） 0：禁止捕获/比较4中断； 1：允许捕获/比较4中断。
3	CC3IEN	允许捕获/比较3中断 （Capture/Compare 3 interrupt enable） 0：禁止捕获/比较3中断； 1：允许捕获/比较3中断。
2	CC2IEN	允许捕获/比较2中断 （Capture/Compare 2 interrupt enable） 0：禁止捕获/比较2中断； 1：允许捕获/比较2中断。
1	CC1IEN	允许捕获/比较1中断 （Capture/Compare 1 interrupt enable） 0：禁止捕获/比较1中断； 1：允许捕获/比较1中断。
0	UIEN	允许更新中断 （Update interrupt enable） 0：禁止更新中断； 1：允许更新中断。

12.4.6 状态寄存器（TIMx_STS）

偏移地址：0x10
复位值：0x0000

15	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved		TITF	Reserved	CC4ITF	CC3ITF	CC2ITF	CC1ITF	UDITF
		rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位域	名称	描述
15:13	Reserved	保留，硬件强制为0。

位域	名称	描述
12	CC4OCF	捕获/比较4重复捕获标记 （Capture/Compare 4 overcapture flag） 参见CC1OCF描述。
11	CC3OCF	捕获/比较3重复捕获标记 （Capture/Compare 3 overcapture flag） 参见CC1OCF描述。
10	CC2OCF	捕获/比较2重复捕获标记 （Capture/Compare 2 overcapture flag） 参见CC1OCF描述。
9	CC1OCF	捕获/比较1重复捕获标记 （Capture/Compare 1 overcapture flag） 仅当相应的通道被配置为输入捕获时，该标记可由硬件置1。写0可清除该位。 0：无重复捕获产生； 1：计数器的值被捕获到TIMx_CCDAT1寄存器时，CC1ITF的状态已经为'1'。
8:7	Reserved	保留，硬件强制为0。
6	TITF	触发器中断标记 （Trigger interrupt flag） 当发生触发事件（当从模式控制器处于除门控模式外的其它模式时，在TRGI输入端检测到有效边沿，或门控模式下的任一边沿）时由硬件对该位置'1'。它由软件清'0'。 0：无触发器事件产生； 1：触发中断等待响应。
5	Reserved	保留，硬件强制为0。
4	CC4ITF	捕获/比较4中断标记 （Capture/Compare 4 interrupt flag） 参考CC1ITF描述。
3	CC3ITF	捕获/比较3中断标记 （Capture/Compare 3 interrupt flag） 参考CC1ITF描述。
2	CC2ITF	捕获/比较2中断标记 （Capture/Compare 2 interrupt flag） 参考CC1ITF描述。
1	CC1ITF	捕获/比较1中断标记 （Capture/Compare 1 interrupt flag） **如果通道CC1配置为输出模式：** 当计数器值与比较值匹配时该位由硬件置1，但在中心对称模式下除外（参考TIMx_CTRL1寄存器的CAMSEL位）。它由软件清'0'。 0：无匹配发生； 1：TIMx_CNT的值与TIMx_CCDAT1的值匹配。 当TIMx_CCDAT1的内容大于TIMx_AR的内容时，在向上或向上/下计数模式时计数器溢出，或向下计数模式时的计数器下溢条件下，CC1ITF位变高 **如果通道CC1配置为输入模式：** 当捕获事件发生时该位由硬件置'1'，它由软件清'0'或通过读TIMx_CCDAT1清'0'。 0：无输入捕获产生； 1：计数器值已被捕获（拷贝）至TIMx_CCDAT1（在IC1上检测到与所选极性相同的边沿）。

位域	名称	描述
0	UDITF	更新中断标记 （Update interrupt flag） 当产生更新事件时该位由硬件置'1'。它由软件清'0'。 0：无更新事件产生； 1：更新中断等待响应。当寄存器被更新时该位由硬件置'1'： – 若TIMx_CTRL1寄存器的UPDIS=0，当重复计数器数值上溢或下溢时（重复计数器=0时产生更新事件）。 – 若TIMx_CTRL1寄存器的UPRS=0、UPDIS =0，当设置TIMx_EVTGEN寄存器的UDGN=1时产生更新事件，通过软件对计数器CNT重新初始化时。 – 若TIMx_CTRL1寄存器的UPRS=0、UPDIS =0，当计数器CNT被触发事件重新初始化时。

12.4.7 事件产生寄存器（TIMx_EVTGEN）

偏移地址:0x14
复位值:0x0000

15 … 7	6	5	4	3	2	1	0
Reserved	TGN	Reserved	CC4GN	CC3GN	CC2GN	CC1GN	UDGN
	w		w	w	w	w	w

位域	名称	描述
15:7	Reserved	保留，硬件强制为0。
6	TGN	产生触发事件 （Trigger generation） 该位由软件置'1'，用于产生一个触发事件，由硬件自动清'0'。 0：无动作； 1：TIMx_STS寄存器的TITF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。
5	Reserved	保留，硬件强制为0。
4	CC4GN	产生捕获/比较4事件 （Capture/Compare 4 generation） 参考CC1GN描述。
3	CC3GN	产生捕获/比较3事件 （Capture/Compare 3 generation） 参考CC1GN描述。
2	CC2GN	产生捕获/比较2事件 （Capture/Compare 2 generation） 参考CC1GN描述。
1	CC1GN	产生捕获/比较1事件 （Capture/Compare 1 generation） 该位由软件置'1'，用于产生一个捕获/比较事件，由硬件自动清'0'。 0：无动作； 1：在通道CC1上产生一个捕获/比较事件： **若通道CC1配置为输出：** 设置CC1ITF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。 **若通道CC1配置为输入：** 当前的计数器值被捕获至TIMx_CCDAT1寄存器；设置CC1ITF=1，若开启对应的中断和DMA，则产生相应的中断和DMA。若CC1ITF已经为1，则设置CC1OCF=1。

位域	名称	描述
0	UDGN	产生更新事件 （Update generation） 该位由软件置'1'，由硬件自动清'0'。 0：无动作； 1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清'0'（但是预分频系数不变）。若在中心对称模式下或DIR=0（向上计数）则计数器被清'0'；若DIR=1（向下计数）则计数器取TIMx_AR的值。

12.4.8 捕获/比较模式寄存器 1（TIMx_CCMOD1）

偏移地址：0x18
复位值：0x0000

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 CCxSEL 定义。该寄存器其它位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输出模式下的功能。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：

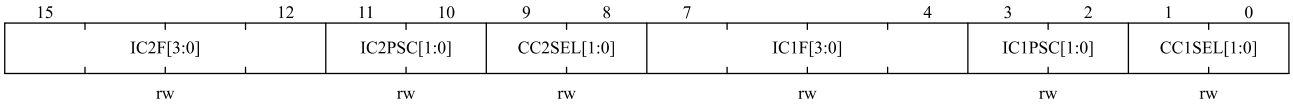
15	14 12	11	10	9 8	7	6 4	3	2	1 0
OC2CEN	OC2M[2:0]	OC2PEN	OC2FEN	CC2SEL[1:0]	OC1CEN	OC1M[2:0]	OC1PEN	OC1FEN	CC1SEL[1:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
15	OC2CEN	输出比较2清0使能 （Output Compare 2 clear enable）
14:12	OC2M[2:0]	输出比较2模式 （Output Compare 2 mode）
11	OC2PEN	输出比较2预装载使能 （Output Compare 2 preload enable）
10	OC2FEN	输出比较2快速使能 （Output Compare 2 fast enable）
9:8	CC2SEL[1:0]	捕获/比较2选择。（Capture/Compare 2 selection） 该位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。*
7	OC1CEN	输出比较1清'0'使能 （Output Compare 1 clear enable） 0：OC1REF 不受ETRF输入的影响； 1：一旦检测到ETRF输入高电平，清除OC1REF=0。

位域	名称	描述
6:4	OC1M[2:0]	输出比较1模式 （Output Compare 1 mode） 该3位定义了输出参考信号OC1REF的动作，而OC1REF决定了OC1、OC1N的值。 OC1REF 是高电平有效，而OC1、OC1N的有效电平取决于CC1P、CC1NP位。 000：冻结。输出比较寄存器TIMx_CCDAT1与计数器TIMx_CNT间的比较对OC1REF不起作用； 001 ： 匹配时设置通道1 为有效电平。 当计数器TIMx_CNT 的值与捕获/ 比较寄存器1 （TIMx_CCDAT1）相同时，强制OC1REF为高。 010 ： 匹配时设置通道1 为无效电平。 当计数器TIMx_CNT 的值与捕获/ 比较寄存器1 （TIMx_CCDAT1）相同时，强制OC1REF为低。 011：翻转。当TIMx_CCDAT1=TIMx_CNT时，翻转OC1REF的电平。 100：强制为无效电平。强制OC1REF为低。 101：强制为有效电平。强制OC1REF为高。 110：PWM模式1 – 在向上计数时，一旦TIMx_CNT<TIMx_CCDAT1时通道1为有效电平，否则为无效电平；在向下计数时，一旦TIMx_CNT>TIMx_CCDAT1时通道1为无效电平（OC1REF=0），否则为有效电平（OC1REF=1）。 111：PWM模式2 – 在向上计数时，一旦TIMx_CNT<TIMx_CCDAT1时通道1为无效电平，否则为有效电平；在向下计数时，一旦TIMx_CNT>TIMx_CCDAT1时通道1为有效电平，否则为无效电平。 *注1：一旦LOCK级别设为3（TIMx_BKDT寄存器中的LCKCFG位）并且CC1SEL=00（该通道配置成输出）则该位不能被修改。* *注2：在PWM模式1或PWM模式2中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到PWM模式时，OC1REF电平才改变。*
3	OC1PEN	输出比较1预装载使能 （Output Compare 1 preload enable） 0：禁止TIMx_CCDAT1寄存器的预装载功能，可随时写入TIMx_CCDAT1寄存器，并且新写入的数值立即起作用。 1：开启TIMx_CCDAT1寄存器的预装载功能，读写操作仅对预装载寄存器操作， TIMx_CCDAT1的预装载值在更新事件到来时被加载至当前寄存器中。 *注1：一旦LOCK级别设为3（TIMx_BKDT寄存器中的LCKCFG位）并且CC1SEL=00（该通道配置成输出）则该位不能被修改。* *注2：仅在单脉冲模式下（TIMx_CTRL1寄存器的ONEPM=1），可以在未确认预装载寄存器情况下使用* *PWM模式，否则其动作不确定。*
2	OC1FEN	输出比较1 快速使能 （Output Compare 1 fast enable） 该位用于加快CC输出对触发输入事件的响应。 0：根据计数器与CCDAT1的值，CC1正常操作，即使触发器是打开的。当触发器的输入有一个有效沿时，激活CC1输出的最小延时为5个时钟周期。 1：输入到触发器的有效沿的作用就象发生了一次比较匹配。因此，OC被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。 OCxFEN只在通道被配置成PWM1或PWM2模式时起作用。

位域	名称	描述
1:0	CC1SEL[1:0]	捕获/比较1 选择。（Capture/Compare 1 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC1通道被配置为输出； 01：CC1通道被配置为输入，IC1映射在TI1上； 10：CC1通道被配置为输入，IC1映射在TI2上； 11：CC1通道被配置为输入，IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC1SEL仅在通道关闭时（TIMx_CCEN寄存器的CC1EN=0）才是可写的。*

输入捕获模式：



位域	名称	描述
15:12	IC2F[3:0]	输入捕获2滤波器 （Input capture 2 filter）
11:10	IC2PSC[1:0]	输入/捕获2预分频器 （Input capture 2 prescaler）
9:8	CC2SEL[1:0]	捕获/比较2选择 （Capture/Compare 2 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。*
7:4	IC1F[3:0]	输入捕获1滤波器 （Input capture 1 filter） 这几位定义了TI1输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到N个事件后会产生一个输出的跳变： 0000：无滤波器，以f_{DTS}采样 1000：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=6 0001：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=2 1001：采样频率$f_{SAMPLING}=f_{DTS}/8$，N=8 0010：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=4 1010：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=5 0011：采样频率$f_{SAMPLING}=f_{CK_INT}$，N=8 1011：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=6 0100：采样频率$f_{SAMPLING}=f_{DTS}/2$，N=6 1100：采样频率$f_{SAMPLING}=f_{DTS}/16$，N=8 0101：采样频率$f_{SAMPLING}=f_{DTS}/2$，N=8 1101：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=5 0110：采样频率$f_{SAMPLING}=f_{DTS}/4$，N=6 1110：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=6 0111：采样频率$f_{SAMPLING}=f_{DTS}/4$，N=8 1111：采样频率$f_{SAMPLING}=f_{DTS}/32$，N=8
3:2	IC1PSC[1:0]	输入/捕获1预分频器 （Input capture 1 prescaler） 这2位定义了CC1输入（IC1）的预分频系数。 一旦CC1EN=0（TIMx_CCEN寄存器中），则预分频器复位。 00：无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获； 01：每2个事件触发一次捕获； 10：每4个事件触发一次捕获； 11：每8个事件触发一次捕获。

位域	名称	描述
1:0	CC1SEL[1:0]	捕获/比较1选择 （Capture/Compare 1 Selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC1通道被配置为输出； 01：CC1通道被配置为输入，IC1映射在TI1上； 10：CC1通道被配置为输入，IC1映射在TI2上； 11：CC1通道被配置为输入，IC1映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC1SEL仅在通道关闭时（TIMx_CCEN寄存器的CC1EN=0）才是可写的。*

12.4.9 捕获/比较模式寄存器 2（TIMx_CCMOD2）

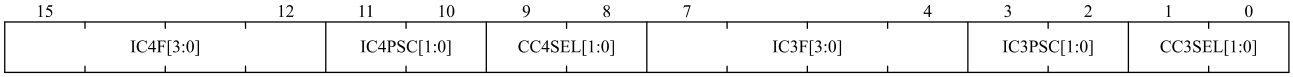
偏移地址：0x1C
复位值：0x0000
参看以上 CCMOD1 寄存器的描述
输出比较模式：

15	14 12	11	10	9 8	7	6 4	3	2	1 0
OC4CEN	OC4MD[2:0]	OC4PEN	OC4FEN	CC4SEL[1:0]	OC3CEN	OC3MD[2:0]	OC3PEN	OC3FEN	CC3SEL[1:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
15	OC4CEN	输出比较4清0使能 （Output compare 4 clear enable）
14:12	OC4MD[2:0]	输出比较4模式 （Output compare 4 mode）
11	OC4PEN	输出比较4预装载使能 （Output compare 4 preload enable）
10	OC4FEN	输出比较4快速使能 （Output compare 4 fast enable）
9:8	CC4SEL[1:0]	捕获/比较4选择 （Capture/Compare 4 selection） 该2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC4通道被配置为输出； 01：CC4通道被配置为输入，IC4映射在TI4上； 10：CC4通道被配置为输入，IC4映射在TI3上； 11：CC4通道被配置为输入，IC4映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC4SEL仅在通道关闭时（TIMx_CCEN寄存器的CC4EN=0）才是可写的。*
7	OC3CEN	输出比较3清0使能 （Output compare 3 clear enable）
6:4	OC3MD[2:0]	输出比较3模式 （Output compare 3 mode）
3	OC3PEN	输出比较3预装载使能 （Output compare 3 preload enable）
2	OC3FEN	输出比较3快速使能 （Output compare 3 fast enable）
1:0	CC3SEL[1:0]	捕获/比较3选择 （Capture/Compare 3 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC3通道被配置为输出； 01：CC3通道被配置为输入，IC3映射在TI3上； 10：CC3通道被配置为输入，IC3映射在TI4上； 11：CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。

位域	名称	描述
		注：CC3SEL仅在通道关闭时（TIMx_CCEN寄存器的CC3EN=0）才是可写的。

输入捕获模式：

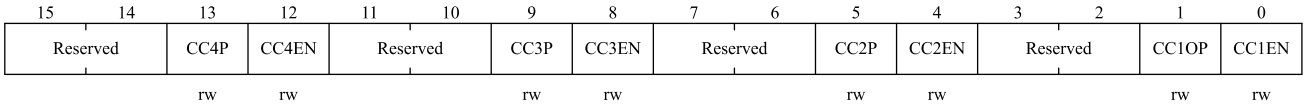


位域	名称	描述
15:12	IC4F[3:0]	输入捕获4滤波器 （Input capture 4 filter）
11:10	IC4PSC[1:0]	输入/捕获4预分频器 （Input capture 4 prescaler）
9:8	CC4SEL[1:0]	捕获/比较4选择 （Capture/Compare 4 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC4通道被配置为输出； 01：CC4通道被配置为输入，IC4映射在TI4上； 10：CC4通道被配置为输入，IC4映射在TI3上； 11：CC4通道被配置为输入，IC4映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC4SEL仅在通道关闭时（TIMx_CCEN寄存器的CC4EN=0）才是可写的。*
7:4	IC3F[3:0]	输入捕获3滤波器 （Input capture 3 filter）
3:2	IC3PSC[1:0]	输入/捕获3预分频器 （Input capture 3 prescaler）
1:0	CC3SEL[1:0]	捕获/比较3选择 （Capture/compare 3 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC3通道被配置为输出； 01：CC3通道被配置为输入，IC3映射在TI3上； 10：CC3通道被配置为输入，IC3映射在TI4上； 11：CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发器输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 *注：CC3SEL仅在通道关闭时（TIMx_CCEN寄存器的CC3EN=0）才是可写的。*

12.4.10 捕获/比较使能寄存器（TIMx_CCEN）

偏移地址：0x20

复位值：0x0000



位域	名称	描述
15:14	Reserved	保留，硬件强制为0。
13	CC4P	输入/捕获4输出极性 （Capture/Compare 4 output polarity） 参考CC1P的描述。
12	CC4EN	输入/捕获4输出使能 （Capture/Compare 4 output enable） 参考CC1EN 的描述。
11:10	Reserved	保留，硬件强制为0。
9	CC3P	输入/捕获3输出极性 （Capture/Compare 3 output polarity） 参考CC1P的描述。
8	CC3EN	输入/捕获3输出使能 （Capture/Compare 3 output enable） 参考CC1EN 的描述。
7:6	Reserved	保留，硬件强制为0。
5	CC2P	输入/捕获2输出极性 （Capture/Compare 2 output polarity） 参考CC1P的描述。
4	CC2EN	输入/捕获2输出使能 （Capture/Compare 2 output enable） 参考CC1EN的描述。
3:2	Reserved	保留，硬件强制为0。
1	CC1P	输入/捕获1输出极性 （Capture/Compare 1 output polarity） **CC1通道配置为输出：** 0：OC1高电平有效； 1：OC1低电平有效。 **CC1通道配置为输入：** 该位选择是IC1还是IC1的反相信号作为触发或捕获信号。 0：不反相：捕获发生在IC1的上升沿；当用作外部触发器时，IC1不反相。 1：反相：捕获发生在IC1的下降沿；当用作外部触发器时，IC1反相。 *注：一旦LOCK级别（TIMx_BKDT寄存器中的LCKCFG位）设为3或2，则该位不能被修改。*
0	CC1EN	输入/捕获1输出使能 （Capture/Compare 1 output enable） **CC1通道配置为输出：** 0： 关闭 – OC1禁止输出，因此OC1的输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。 1： 开启 – OC1信号输出到对应的输出引脚，其输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。 **CC1通道配置为输入：** 该位决定了计数器的值是否能捕获入TIMx_CCDAT1寄存器。 0：捕获禁止； 1：捕获使能。

表 12-4 标准 OCx 通道的输出控制位

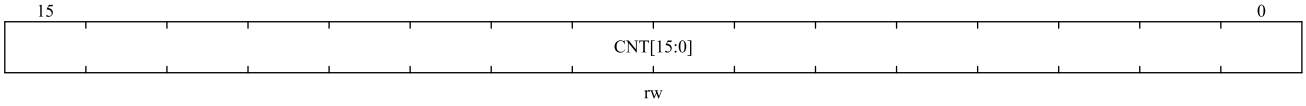
CCxEN位	OCx输出状态
0	禁止输出（OCx=0，OCx_EN=0）

1	OCx = OCxREF + 极性，OCx_EN=1

注：连接到标准 OCx 通道的外部 I/O 引脚状态，取决于 OCx 通道状态和 GPIO 以及 AFIO 寄存器。

12.4.11 计数器（TIMx_CNT）

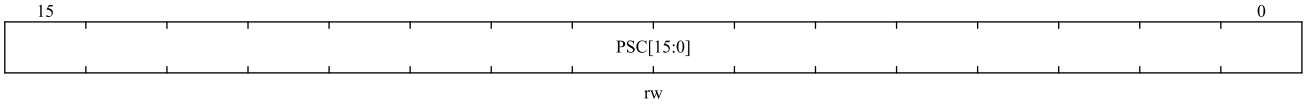
偏移地址：0x24
复位值：0x0000



位域	名称	描述
15:0	CNT[15:0]	计数器的值 （Counter value）

12.4.12 预分频器（TIMx_PSC）

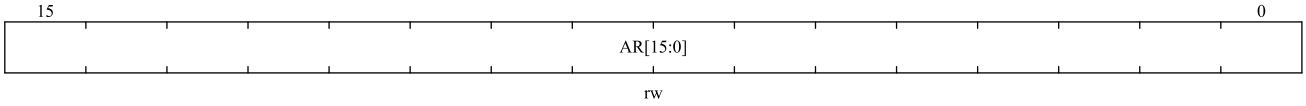
偏移地址：0x28
复位值：0x0000



位域	名称	描述
15:0	PSC[15:0]	预分频器的值 （Prescaler value） 计数器的时钟频率（CK_CNT）等于f_{CK_PSC}/(PSC[15:0]+1)。 PSC包含了每次当更新事件产生时，装入当前预分频器寄存器的值；更新事件包括计数器被TIM_EVTGEN的UDGN位清'0'或被工作在复位模式的从控制器清'0'。

12.4.13 自动重装载寄存器（TIMx_AR）

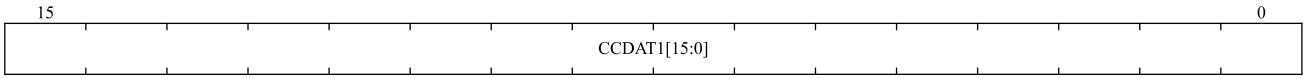
偏移地址:0x2C
复位值:0x0000



位域	名称	描述
15:0	AR[15:0]	自动重装载的值 （Prescaler value） AR 包含了将要装载入实际的自动重装载寄存器的值。 详细参考 12.3.1 节：有关 AR 的更新和动作。 当自动重装载的值为空时，计数器不工作。

12.4.14 捕获/比较寄存器 1（TIMx_CCDAT1）

偏移地址：0x34
复位值：0x0000



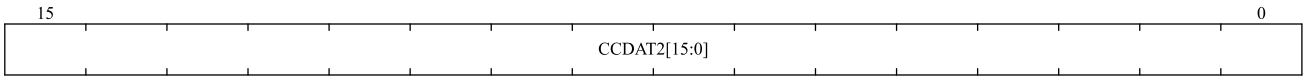
rw/r

位域	名称	描述
15:0	CCDAT1[15:0]	捕获/比较通道1的值 （Capture/Compare 1 value） **若CC1通道配置为输出：** CCDAT1包含了装入当前捕获/比较1寄存器的值（预装载值）。 如果在TIMx_CCMOD1寄存器（OC1PEN位）中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较1寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并在OC1端口上产生输出信号。 **若CC1通道配置为输入：** CCDAT1包含了由上一次输入捕获1事件（IC1）传输的计数器值。

12.4.15 捕获/比较寄存器 2（TIMx_CCDAT2）

偏移地址：0x38

复位值：0x0000



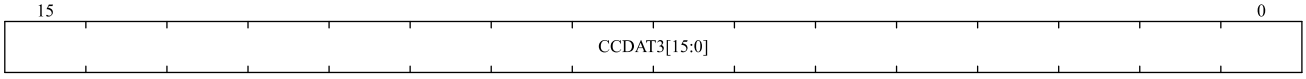
rw/r

位域	名称	描述
15:0	CCDAT2[15:0]	捕获/比较通道2的值 （Capture/Compare 2 value） **若CC2通道配置为输出：** CCDAT2包含了装入当前捕获/比较2寄存器的值（预装载值）。 如果在TIMx_CCMOD1寄存器（OC2PEN位）中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较2寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并在OC2端口上产生输出信号。 **若CC2通道配置为输入：** CCDAT2包含了由上一次输入捕获2事件（IC2）传输的计数器值。

12.4.16 捕获/比较寄存器 3（TIMx_CCDAT3）

偏移地址：0x3C

复位值：0x0000

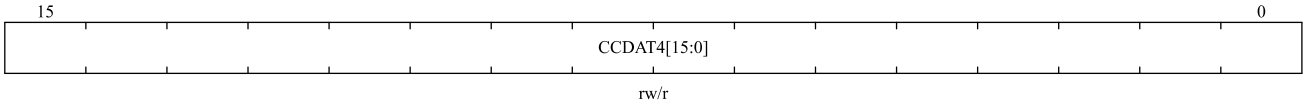


rw/r

位域	名称	描述
15:0	CCDAT3[15:0]	捕获/比较通道3的值 （Capture/Compare 3 value） **若CC3通道配置为输出：** CCDAT3包含了装入当前捕获/比较3寄存器的值（预装载值）。 如果在TIMx_CCMOD2寄存器（OC3PEN位）中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较3寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并在OC3端口上产生输出信号。若**CC3通道配置为输入：** CCDAT3包含了由上一次输入捕获3事件（IC3）传输的计数器值。

12.4.17 捕获/比较寄存器 4（TIMx_CCDAT4）

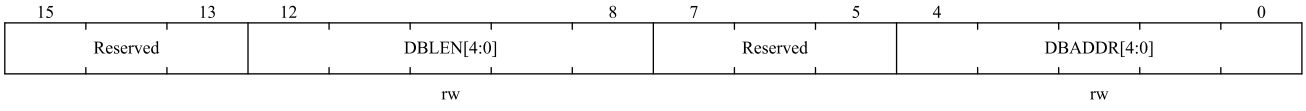
偏移地址：0x40
复位值：0x0000



位域	名称	描述
15:0	CCDAT4[15:0]	捕获/比较通道4的值 （Capture/Compare 4 value） **若CC4通道配置为输出：** CCDAT4包含了装入当前捕获/比较4寄存器的值（预装载值）。 如果在TIMx_CCMOD2寄存器（OC4PEN位）中未选择预装载特性，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较4寄存器中。当前捕获/比较寄存器参与同计数器TIMx_CNT的比较，并在OC4端口上产生输出信号。 **若CC4通道配置为输入：** CCDAT4包含了由上一次输入捕获4事件（IC4）传输的计数器值。

12.4.18 DMA 控制寄存器（TIMx_DCTRL）

偏移地址：0x48
复位值：0x0000

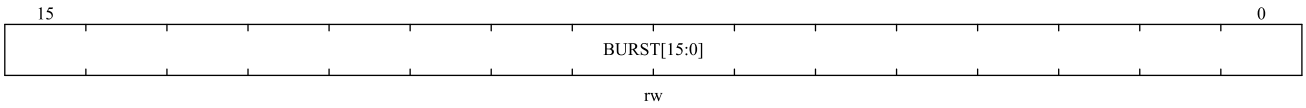


位域	名称	描述
15:13	Reserved	保留，硬件强制为0。

位域	名称	描述
12:8	DBLEN[4:0]	DMA连续传送长度 （DMA burst length） 这些位定义了DMA在连续模式下的传送长度（当对TIMx_DADDR寄存器进行读或写时，定时器则进行一次连续传送），即：定义传输的次数，传输可以是半字（双字节）或字节： 00000：1次传输 00001：2次传输 00010：3次传输 .... 10001：18次传输 例：我们考虑这样的传输：DBLEN=7，DBADDR=TIM2_CTRL1 如果DBLEN=7，DBADDR=TIM2_CTRL1表示待传输数据的地址，那么传输的地址由下式给出： (TIMx_CTRL1的地址) + DBADDR + (DMA索引)，其中 DMA索引 = DBLEN 其中(TIMx_CTRL1的地址) + DBADDR再加上7，给出了将要写入或者读出数据的地址，这样数据的传输将发生在从地址(TIMx_CTRL1的地址) + DBADDR开始的7个寄存器。 根据DMA数据长度的设置，可能发生以下情况： 如果设置数据为半字（16位），那么数据就会传输给全部7个寄存器。 如果设置数据为字节，数据仍然会传输给全部7个寄存器：第一个寄存器包含第一个MSB字节，第二个寄存器包含第一个LSB字节，以此类推。因此对于定时器，用户必须指定由DMA传输的数据宽度。
7:5	Reserved	保留，硬件强制为0。
4:0	DBADDR[4:0]	DMA基地址 （DMA base address） 这些位定义了DMA在连续模式下的基地址（当对TIMx_DADDR寄存器进行读或写时），DBADDR定义为从TIMx_CTRL1寄存器所在地址开始的偏移量： 00000：TIMx_CTRL1， 00001：TIMx_CTRL2， 00010：TIMx_SMCTRL， .....

12.4.19 连续模式的 DMA 地址（TIMx_DADDR）

偏移地址：0x4C
复位值：0x0000



位域	名称	描述
15:0	BURST[15:0]	DMA连续传送寄存器 （DMA register for burst accesses） 对TIMx_DADDR寄存器的读或写会导致对以下地址所在寄存器的存取操作： TIMx_CTRL1地址 + DBADDR + DMA索引，其中： “TIMx_CTRL1地址”是控制寄存器1（TIMx_CTRL1）所在的地址； “DBADDR”是TIMx_DCTRL寄存器中定义的基地址； “DMA索引”是由DMA自动控制的偏移量，它取决于TIMx_DCTRL寄存器中定义的DBLEN。

13 基本定时器（TIM6 和 TIM7）

13.1 TIM6 和 TIM7 简介

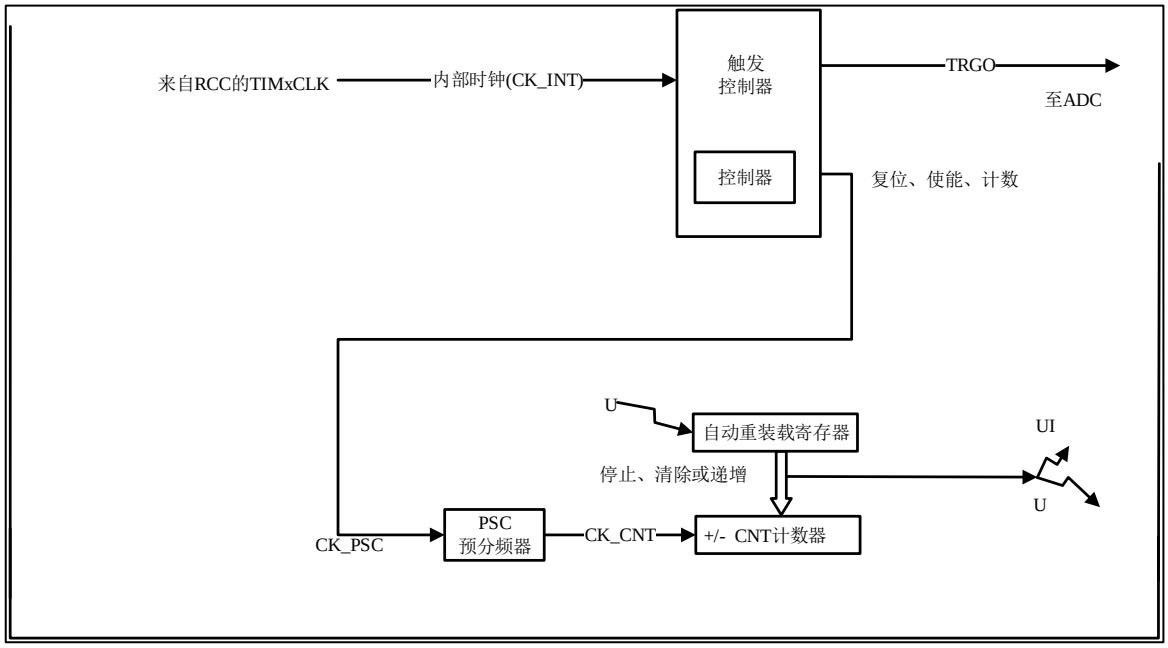
基本定时器（TIM6，TIM7）各包含一个 16 位自动装载计数器，由各自的可编程预分频器进行驱动。可以为通用定时器提供时间基准，特别地可以为数模转换器（DAC）提供时钟。实际上，它们在芯片内部直接连接到 DAC，并通过触发输出直接驱动 DAC。基本定时器互相独立，不共享任何资源。

13.2 TIM6 和 TIM7 的主要特性

TIM6 和 TIM7 定时器的主要功能包括:

- 16 位自动重装载累加计数器
- 16 位可编程（可实时修改）预分频器，用于对输入的时钟按系数为 1～65536 之间的任意数值进行分频
- 触发 DAC 的同步电路
- 在更新事件（计数器溢出）时产生中断/DMA 请求

图 13-1 基本定时器框图



13.3 TIM6 和 TIM7 的功能

13.3.1 时基单元

这个可编程定时器的主要部分是一个带有自动重装载的 16 位累加计数器，计数器的时钟来源于一个预分频器。

软件可以读写计数器、自动重装载寄存器和预分频寄存器，即使计数器运行时也可以操作。时基单元包含：

- 计数器寄存器（TIMx_CNT）
- 预分频寄存器（TIMx_PSC）
- 自动重装载寄存器（TIMx_AR）

自动重装载寄存器是预加载的，每次读写自动重装载寄存器时，实际上是通过读写预加载寄存器实现。根据 TIMx_CTRL1 寄存器中的自动重装载预加载使能位（ARPEN），写入预加载寄存器的内容能够立即或在每次更新事件时，传送到它的影子寄存器。

- 当 TIMx_CTRL1 寄存器的 UPDIS 位为'0'，则每当计数器达到溢出值时，硬件发出更新事件。
- 软件也可以产生更新事件。

关于更新事件的产生，随后会有详细的介绍。

计数器由预分频输出 CK_CNT 驱动，设置 TIMx_CTRL1 寄存器中的计数器使能位（CNTEN）使能计数器计数。

注意：实际的设置计数器使能信号 CNT_EN 相对于 CNTEN 滞后一个时钟周期。

13. 3. 1. 1 预分频器

预分频器可以介于 1 至 65536 之间的任意数值对计数器时钟进行分频。通过对一个 16 位寄存器（TIMx_PSC）的计数分频实现。由于 TIMx_PSC 控制寄存器具有缓冲功能，可以在运行过程中改变它的数值，新的预分频数值将在下一个更新事件时起作用。

以下两图是在运行过程中改变预分频系数的示例。

图 13-2 预分频系数从 1 变到 2 的计数器时序图

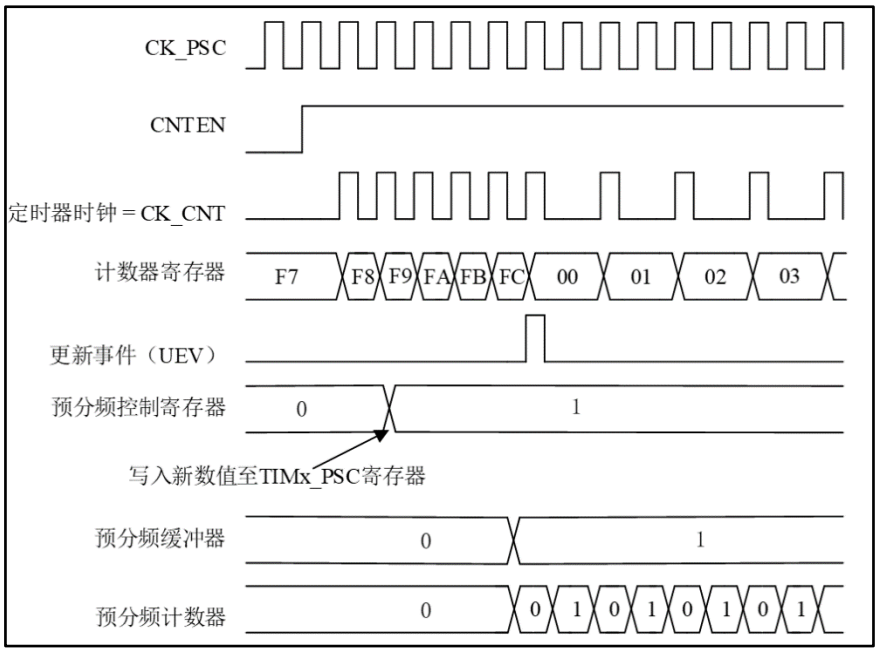
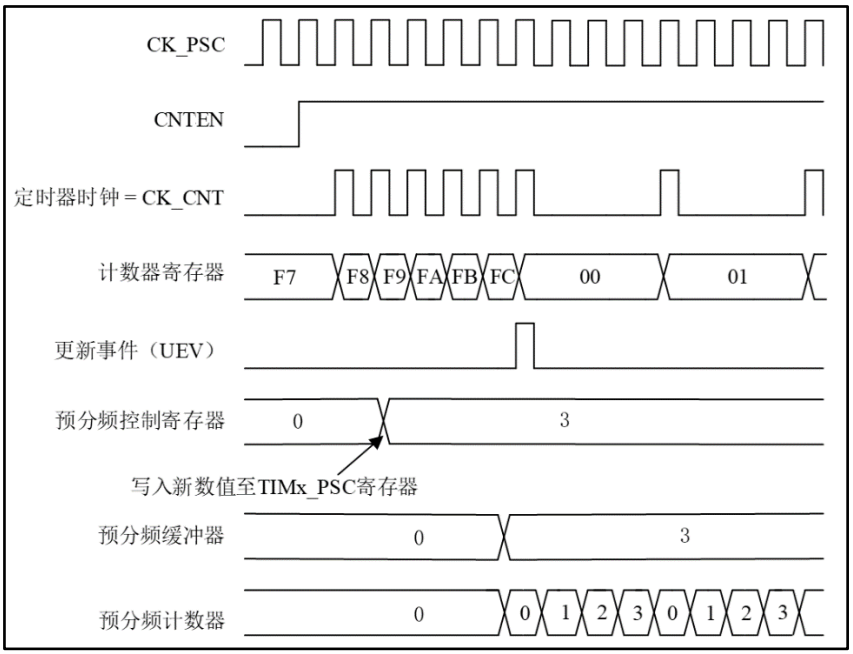


图 13-3 预分频系数从 1 变到 4 的计数器时序图



13. 3. 2 计数模式

计数器从 0 累加计数到自动重装载数值（TIMx_AR 寄存器），然后产生一个计数器溢出事件并重新从 0 开始计数。

- 每次计数器溢出时可以产生更新事件;
- （通过软件或使用从模式控制器）设置 TIMx_EVTGEN 寄存器的 UDGN 位也可以产生更新事件。

为避免在写入预加载寄存器时更改影子寄存器，可以设置 TIMx_CTRL1 中的 UPDIS 位以禁止产生 UEV 事件。在清除 UPDIS 位为'0'之前，将不再产生更新事件，但计数器和预分频器依然会在应产生更新事件时重新从 0 开始计数（但预分频系数不变）。另外，如果设置了 TIMx_CTRL1 寄存器中的 UPRS（选择更新请求），设置 UDGN 位可以产生一次更新事件 UEV，但不设置 UDITF 标志（即没有中断或 DMA 请求）。

当发生一次更新事件时，所有寄存器都会被更新并（根据 UPRS 位）设置更新标志（TIMx_STS 寄存器的 UDITF 位）:

- 传送预装载值（TIMx_PSC 寄存器的内容）至预分频器的缓冲区。
- 自动重装载影子寄存器被更新为预装载值（TIMx_AR）。

以下是一些在 TIMx_AR=0x36 时不同时钟频率下计数器工作的示例。

图 13-4 计数器时序图，内部时钟分频系数为 1

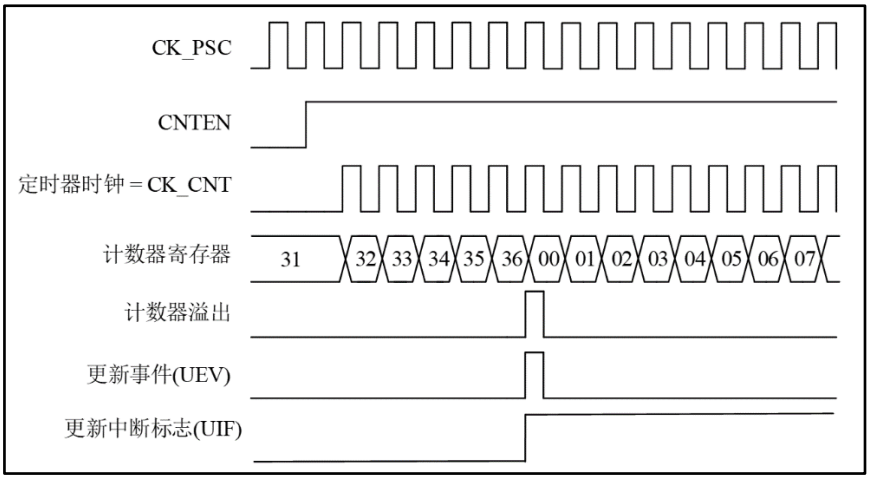


图 13-5 计数器时序图，内部时钟分频系数为 2

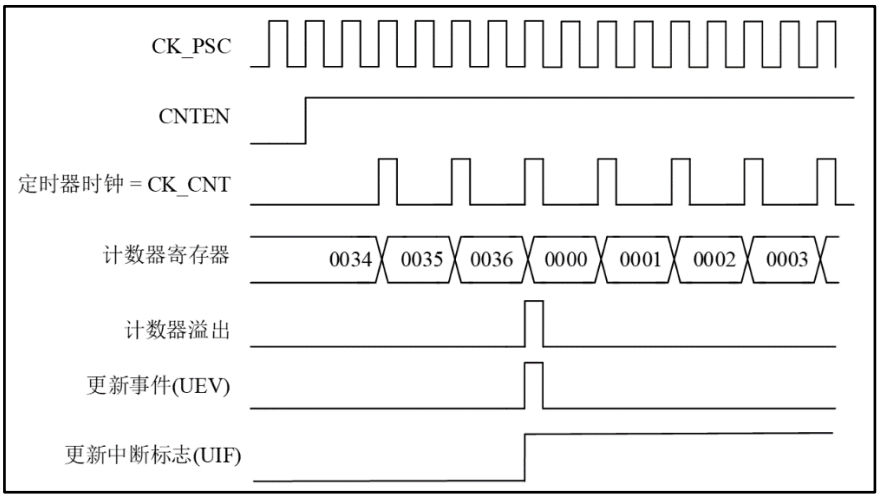


图 13-6 计数器时序图，内部时钟分频系数为 4

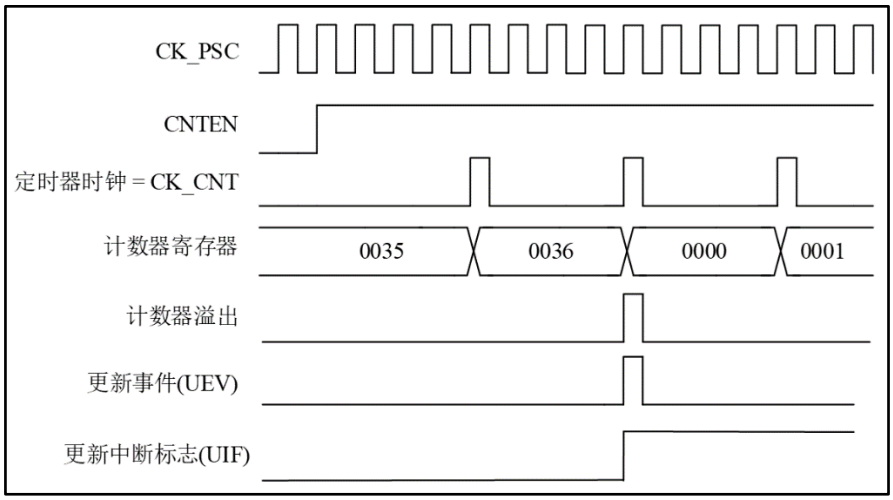


图 13-7 计数器时序图，内部时钟分频系数为 N

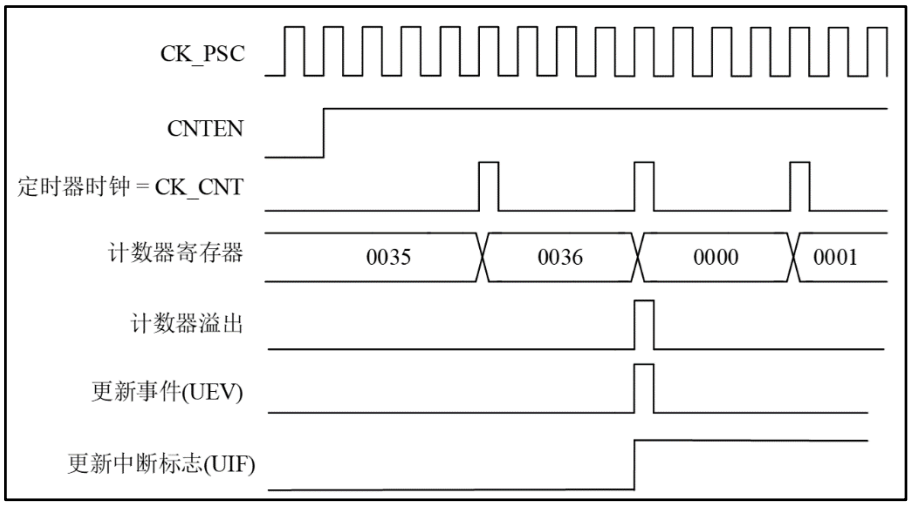


图 13-8 计数器时序图，当 ARPEN=0 时的更新事件（TIMx_AR 没有预装载）

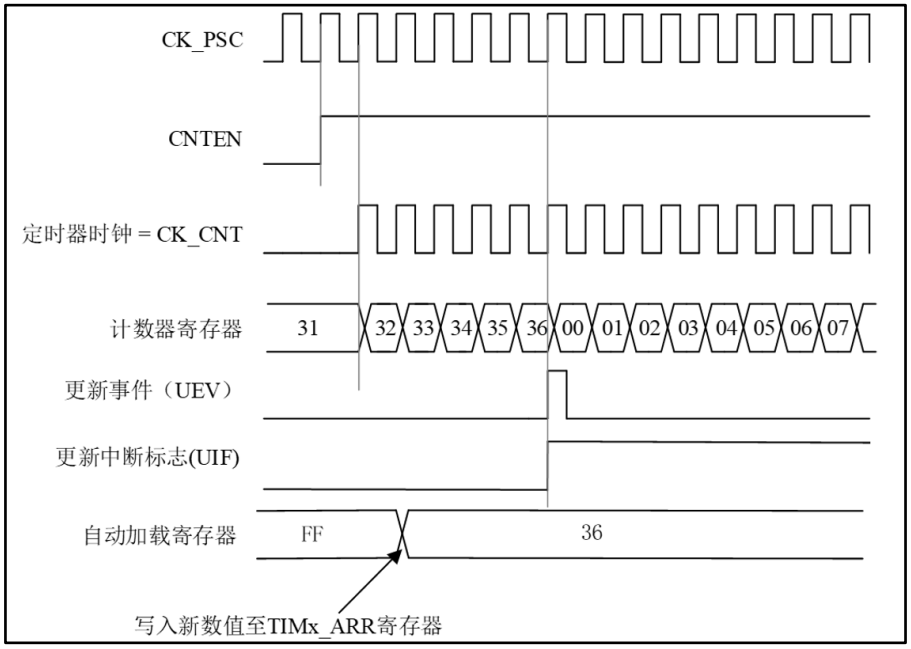
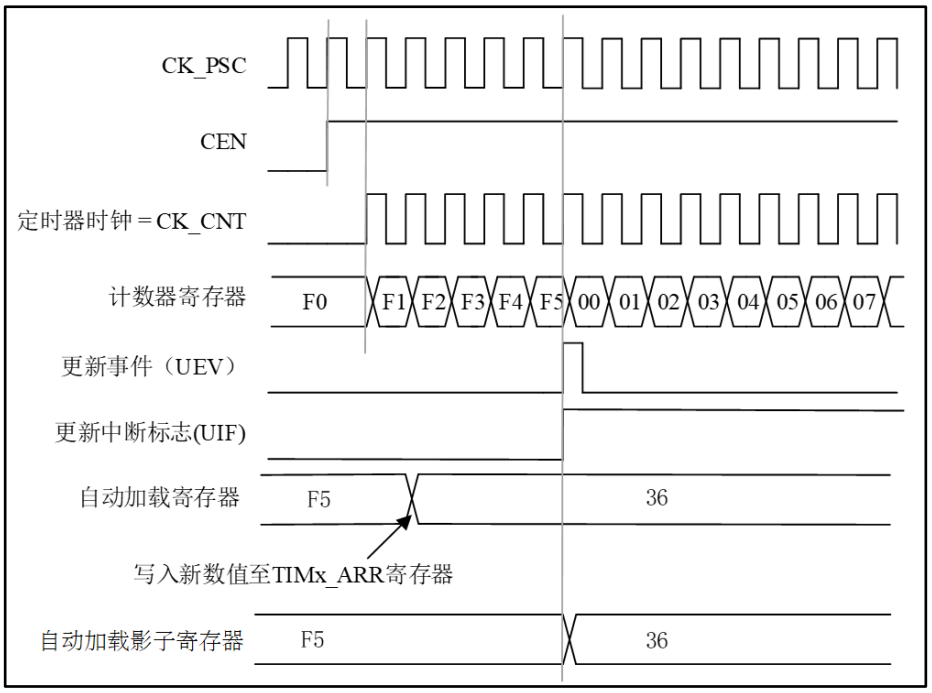


图 13-9 计数器时序图，当 ARPEN=1 时的更新事件（预装载 TIMx_AR）



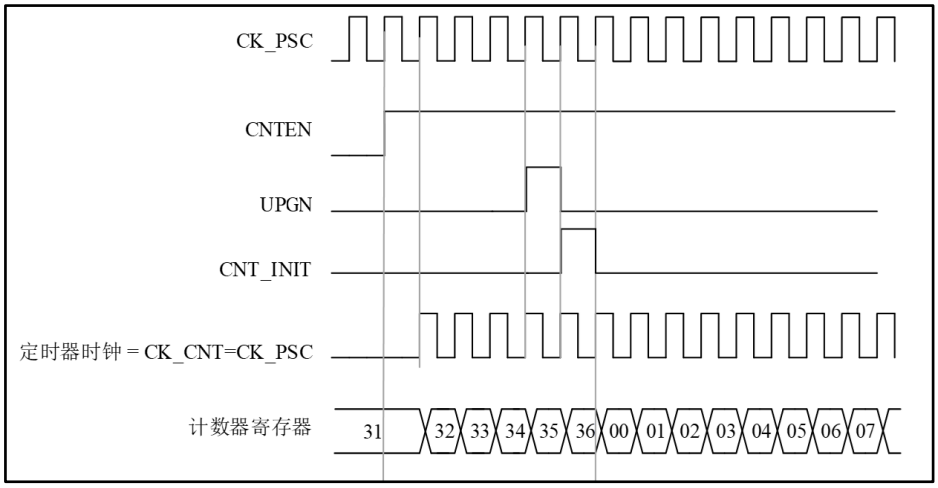
13. 3. 3 时钟源

计数器的时钟源由内部时钟（CK_INT）提供。

TIMx_CTRL1 寄存器的 CNTEN 位和 TIMx_EVTGEN 寄存器的 UDGN 位是实际的控制位，(除了 UDGN 位被自动清除外）只能通过软件改变它们。当 CNTEN 位为'1'，内部时钟即向预分频器提供时钟。

下图给出了普通模式下向上计数器，没有预分频器时的示例。

图 13-10 普通模式时序图，内部时钟分频系数为 1



13. 3. 4 调试模式

当微控制器进入调试模式（N308 核心停止）时，根据 DBG 模块中的配置位 DBG_TIMx_STOP 的设

置，TIMx 计数器或者继续计数或者停止工作。详见第 29.3.2 节。

13.4 TIM6 和 TIM7 寄存器

有关寄存器描述中用到的缩写，请参考第 1.1 节。

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

13.4.1 TIM6 和 TIM7 寄存器图

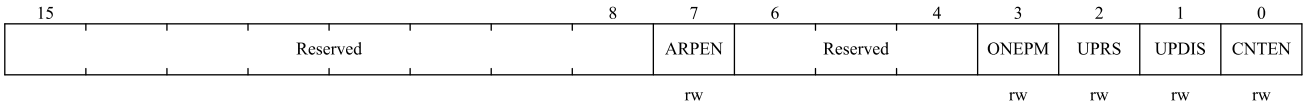
下表中将 TIMx 的所有寄存器映射到一个 16 位可寻址（编址）空间。

表 13-1 TIM6 和 TIM7–寄存器图和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	TIMx_CTRL1	Reserved																								ARPEN	Reserved			ONEPM	UPRS	UPDIS	CNTEN
	Reset Value																									0				0	0	0	0
004h	TIMx_CTRL2	Reserved																									MMSEL[2:0]			Reserved				
	Reset Value																										0	0	0				
008h	Reserved																																
00Ch	TIMx_DINTEN	Reserved																							UDEN	Reserved							UIEN
	Reset Value																								0								0
010h	TIMx_STS	Reserved																															UDITF
	Reset Value																																0
014h	TIMx_EVTGEN	Reserved																															UDGN
	Reset Value																																0
018h	Reserved																																
01Ch	Reserved																																
020h	Reserved																																
024h	TIMx_CNT	Reserved																CNT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028h	TIMx_PSC	Reserved																PSC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	TIMx_AR	Reserved																AR[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.2 TIM6 和 TIM7 控制寄存器 1（TIMx_CTRL1）

偏移地址：0x00
复位值：0x0000

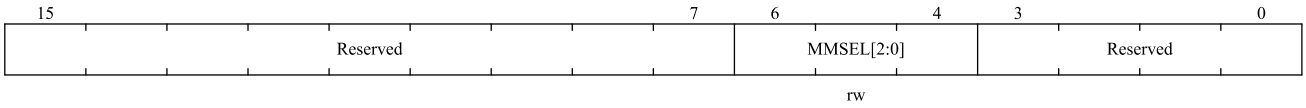


位域	名称	描述
15:8	Reserved	保留，硬件强制为0。
7	ARPEN	自动重装载预装载允许位 （Auto-reload preload enable） 0：TIMx_AR寄存器没有缓冲； 1：TIMx_AR寄存器被装入缓冲器。
6:4	Reserved	保留，硬件强制为0。
3	ONEPM	单脉冲模式 （One pulse mode） 0：在发生更新事件时，计数器不停止； 1：在发生下一次更新事件（清除CNTEN位）时，计数器停止。
2	UPRS	更新请求源 （Update request source） 软件通过该位选择UEV事件的源 0：如果使能了更新中断或DMA请求，则下述任一事件产生更新中断或DMA请求： – 计数器溢出/下溢 – 设置UDGN位 – 从模式控制器产生的更新 1：如果使能了更新中断或DMA请求，则只有计数器溢出/下溢才产生更新中断或DMA请求。
1	UPDIS	禁止更新 （Update disable） 软件通过该位允许/禁止UEV事件的产生 0：允许UEV。更新（UEV）事件由下述任一事件产生： – 计数器溢出/下溢 – 设置UDGN位 – 从模式控制器产生的更新 具有缓存的寄存器被装入它们的预装载值（更新影子寄存器）。 1：禁止UEV。不产生更新事件，影子寄存器（AR、PSC）保持它们的值。如果设置了UDGN位或从模式控制器发出了一个硬件复位，则计数器和预分频器被重新初始化。
0	CNTEN	使能计数器 （Counter enable） 0：禁止计数器； 1：使能计数器。 *注：在软件设置了CNTEN位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CNTEN位。*

13.4.3 TIM6 和 TIM7 控制寄存器 2（TIMx_CTRL2）

偏移地址：0x04

复位值：0x0000

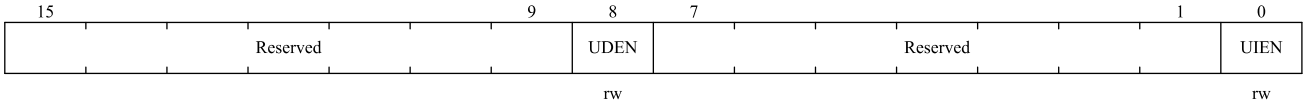


位域	名称	描述
15:7	Reserved	保留，硬件强制为0。
6:4	MMSEL[2:0]	主模式选择 （Master mode selection） 这些位用于选择在主模式下向从定时器发送的同步信息（TRGO），有以下几种组合： 000：**复位** – 使用TIMx_EVTGEN寄存器的UDGN位作为触发输出（TRGO）。如果触发输入产生了复位（从模式控制器配置为复位模式），则相对于实际的复位信号，TRGO上的信号有一定的延迟。 001：**使能** – 计数器使能信号CNT_EN被用作为触发输出（TRGO）。它可用于在同一时刻启动多个定时器，或控制使能从定时器的时机。计数器使能信号是通过CNTEN控制位和配置为门控模式时的触发输入的'逻辑或'产生。 当计数器使能信号是通过触发输入控制时，在TRGO输出上会有一些延迟，除非选择了主/从模式（见TIMx_SMCTRL寄存器的MSMD位）。 010：**更新** – 更新事件被用作为触发输出（TRGO）。例如一个主定时器可以作为从定时器的预分频器使用。
3:0	Reserved	保留，硬件强制为0。

13.4.4 TIM6 和 TIM7 DMA/中断使能寄存器（TIMx_DINTEN）

偏移地址：0x0C

复位值：0x0000

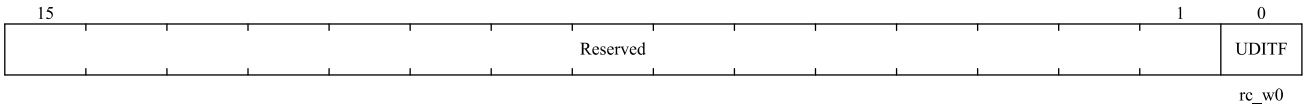


位域	名称	描述
15:9	Reserved	保留，硬件强制为0。
8	UDEN	更新DMA请求使能 （Update DMA request enable） 0：禁止更新DMA请求 1：使能更新DMA请求
7:1	Reserved	保留，硬件强制为0。
0	UIEN	更新中断使能 （Update interrupt enable） 0：禁止更新中断 1：使能更新中断

13.4.5 TIM6 和 TIM7 状态寄存器（TIMx_STS）

偏移地址：0x10

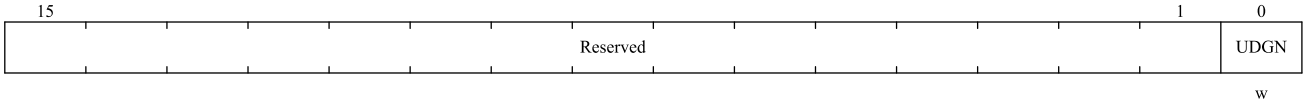
复位值：0x0000



位域	名称	描述
15:1	Reserved	保留，硬件强制为0。
0	UDITF	更新中断标志 （Update interrupt flag） 硬件在更新中断时设置该位，它由软件清除。 0：没有产生更新。 1：产生了更新中断。下述情况下由硬件设置该位： – 计数器产生上溢或下溢并且TIMx_CTRL1中的UPDIS=0； – 如果TIMx_CTRL1中的UPRS=0并且UPDIS=0，当使用TIMx_EVTGEN寄存器的UDGN位重新初始化计数器CNT时。

13.4.6 TIM6 和 TIM7 事件产生寄存器（TIMx_EVTGEN）

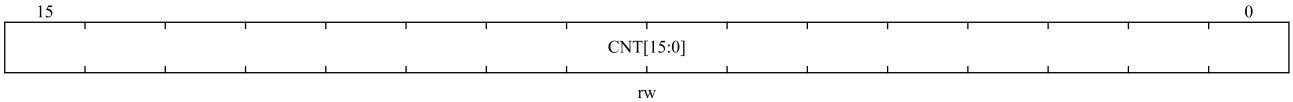
偏移地址：0x14
复位值：0x0000



位域	名称	描述
15:1	Reserved	保留，硬件强制为0。
0	UDGN	产生更新事件 （Update generation） 该位由软件设置，由硬件自动清除。 0：无作用 1：重新初始化定时器的计数器并产生对寄存器的更新。 *注意：预分频器也被清除（但预分频系数不变）。*

13.4.7 TIM6 和 TIM7 计数器（TIMx_CNT）

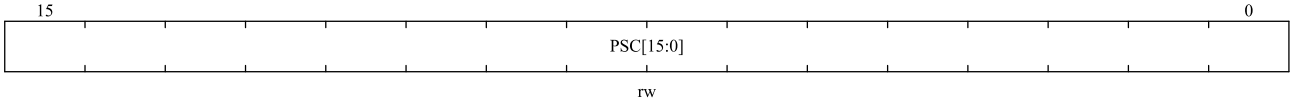
偏移地址：0x24
复位值：0x0000



位域	名称	描述
15:0	CNT[15:0]	计数器数值 （Counter value）

13.4.8 TIM6 和 TIM7 预分频器（TIMx_PSC）

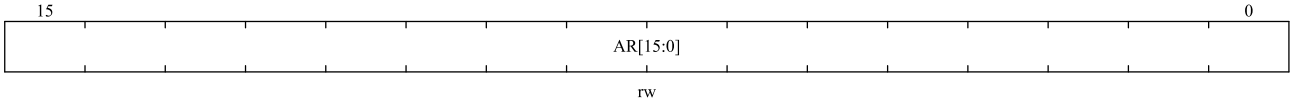
偏移地址：0x28
复位值：0x0000



位域	名称	描述
15:0	PSC[15:0]	预分频器数值 （Prescaler value） 计数器的时钟频率CK_CNT等于f_{CK_PSC}/(PSC[15:0]+1)。 在每一次更新事件时，PSC的数值被传送到实际的预分频寄存器中。

13.4.9 TIM6 和 TIM7 自动重装载寄存器（TIMx_AR）

偏移地址：0x2C
复位值：0x0000



位域	名称	描述
15:0	AR[15:0]	自动重装载数值 （Prescaler value） AR的数值将传送到实际的自动重装载寄存器中。关于AR的更新和作用，详见13.3.1。 如果自动重装载数值为0，则计数器停止。

14 实时时钟（RTC）

14.1 RTC 简介

RTC 提供了在低功耗模式下自动唤醒的功能。

实时时钟（RTC）是一个独立的 BCD 定时器/计数器。RTC 提供一个带有可编程的闹钟中断的时钟/日历。RTC 还包括一个具有中断功能的周期性可编程唤醒标志。两个 32 位寄存器包含十进制格式(BCD)表示亚秒、秒、分钟、小时（12 或 24 小时格式)、天（星期几)、日（几号)、月和年。

系统可以自动执行 28、29（闰年)、30 和 31 天的月补偿。也可以进行夏令时补偿。

亚秒值以二进制格式作为单独的 32 位寄存器提供。其他的 32 位寄存器包含可编程的闹钟亚秒、秒、分钟、小时、天和日。

数字校准功能可以补偿晶体振荡器精度上的偏差。

在 Backup 域复位之后，所有 RTC 寄存器都受到保护，以防止可能的意外写访问。

当启用 GPIO 上的事件时，在寄存器中保存当前日历时，可以启用时间戳功能。

只要 RTC 启用并且电压保持在工作范围内，RTC 就不会停止，无论设备状态如何（运行、休眠、停机或待机状态)。

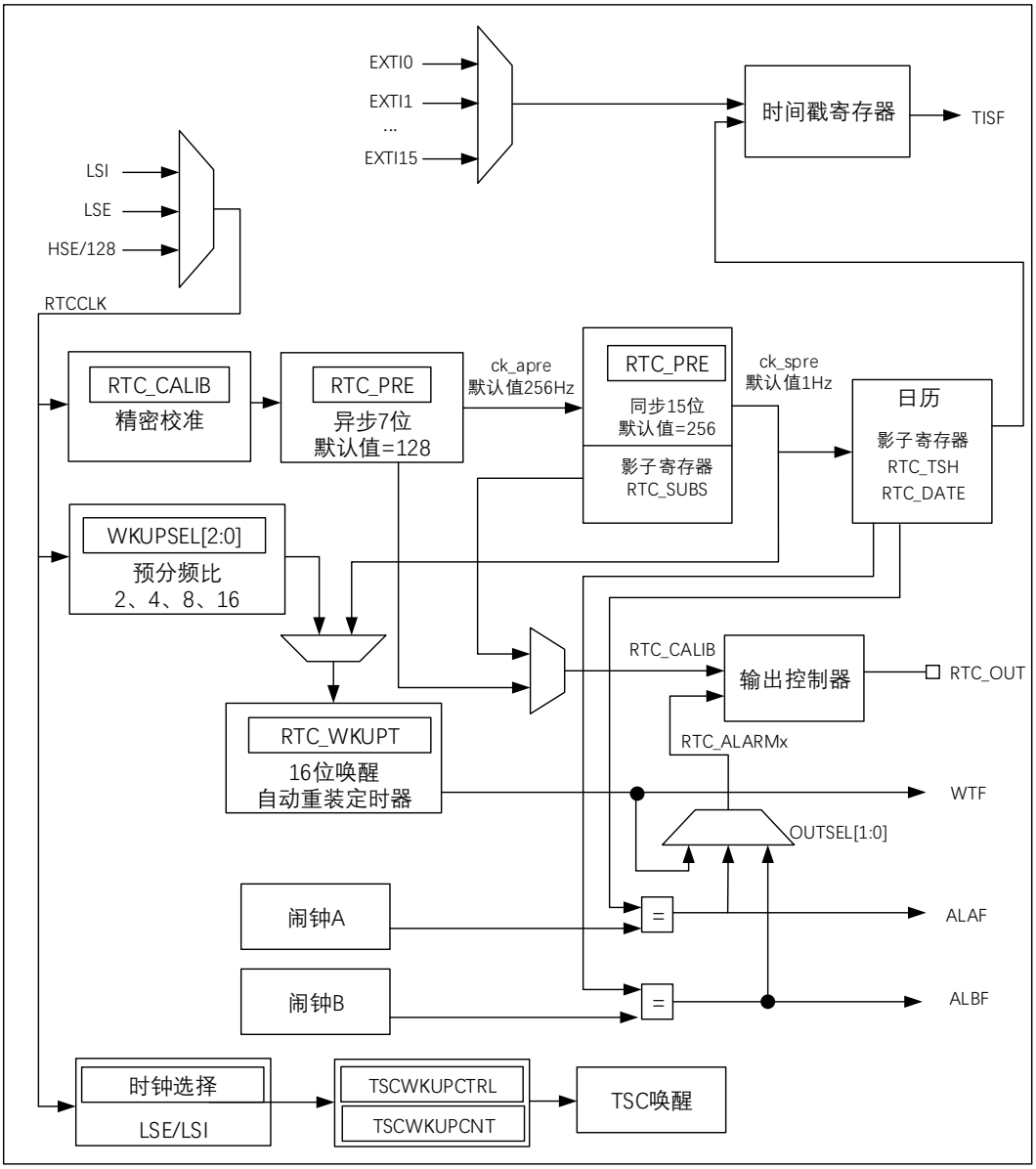
14.1.1 主要特性

特性	描述
时钟	RTC 时钟可以从 LSI、LSE 和 HSE 中分频出来，分别是 40KHz、32.768KHz 和 HSE/128
复位	APB 接口由系统复位。一些来自 RTC 模块的寄存器复位跟 APB 复位同步。 RTC 核心区域复位由后备区域复位。
日历	日历分亚秒、秒、分钟、小时（12 或 24 格式)、天（星期几)、日（几号)、月和年。这些数据都保存在 APB 模块中影子寄存器中。
唤醒	输出寄存器 RTC_OUT 可以配置为发送唤醒事件到 GPIO，同时我们可以选择中断/事件来唤醒 CPU 的 SLEEP，STOP0 和 STOP2 模式。
闹钟	可编程闹钟与中断功能。可以通过日历字段的任何组合触发闹钟。闹钟可以通过 RTC_OUT 配置输出到 GPIO，也可以唤醒 CPU 或触发 PWR 在匹配发生时从 SLEEP、STOP0、STOP2 和 STANDBY 模式中唤醒
中断/事件	Alarm A/Alarm B 唤醒中断

14. 2 RTC 功能描述

14. 2. 1 RTC 框图

图 14-1 RTC 框图



RTC 包括以下模块:

- 两个闹钟
- 来自 I/O 的一个时间戳事件
- 输出功能（RTC_OUT）：
  - RTC_CALIB: 256 Hz 或 1Hz 时钟输出（LSE 频率为 32.768 KHz）。这个输出是通过在 RTC_CTRL

寄存器中位 COEN 置 1 来启用的

- RTC_ALARMx:此输出是通过在 RTC_CTRL 寄存器中配置 OUTSEL[1:0]位来启用的，该寄存器选择 Alarm A、Alarm B 或唤醒输出

- 输入功能:
  - 时间戳事件（RTC_TS）
- 通过配置一些输出寄存器来控制 PC13 IO 口，主要包括:
  - 寄存器 RTC_OPT 位 TYPE 来配置 PC13 的开漏/推挽输出

14.2.2 RTC 控制的 GPIO

RTC_TS 输入需要 EXIT 模块来启动，具体请参考时间戳触发源选择寄存器（EXTI_TS_SEL）。RTC_OUT 映射到 PC13。无论 PC13 的 GPIO 配置如何，PC13 引脚的配置都由 RTC 控制。寄存器 RTC_OPT 位 TYPE 来配置 PC13 的开漏/推挽输出。

14.2.3 RTC 时钟和预分频

RTC 时钟源（RTCCLK）通过时钟控制器（RCC）在 LSE 时钟、LSI 振荡时钟和 HSE 时钟之间进行选择。一个可编程的预分频器产生一个 1Hz 的时钟，用于更新日历。为了降低功耗，该预分频器分为 2 个可编程的预分频器。主要如下:

- 通过 RTC_PRE 寄存器的 DIVA 位配置的 7 位异步预分频器
- 通过 RTC_PRE 寄存器的 DIVS 位配置的 15 位同步预分频器

使用 32.768 KHz 的 LSE，需要将异步分频因子设置为 128，同步分频因子设置为 256，可以得到的内部时钟频率为 1 Hz（ck_spre）。最小分频系数是 1，最大是 2^{22}。这对应于约为 4MHz 的最大的输入频率。

f_{ck_apre} 公式:

$$f_{ck\_apre} = \frac{f_{RTCCLK}}{DIVA+1}$$

ck_apre 时钟用于对二进制的 RTC_SUBS 亚秒递减计数器提供时钟。当到达 0 时，用 DIVS 的内容重新加载 RTC_SUBS。

f_{ck_spre} 公式:

$$f_{ck\_spre} = \frac{f_{RTCCLK}}{(DIVS+1)*(DIVA+1)}$$

ck_spre 时钟可用于更新日历或作为 16 位唤醒自动重载定时器的时基。为了获得较短的超时时间，还可以将 16 位唤醒自动重载定时器与经可编程的 4 位异步预分频器分频的 RTCCLK 一同运行。

14.2.4 RTC 日历时钟

RTC 日历时间和日期寄存器是通过与 PCLK （APB 时钟）同步的影子寄存器来访问的。也可以直接

访问它们，以避免等待同步时间。寄存器对照如下:

- RTC_SUBS 设置亚秒
- RTC_TSH 设置时间
- RTC_DATE 设置日期

每隔两个 RTCCLK 周期，便将当前日历值复制到影子寄存器中，并将 RTC_INITSTS 寄存器的 RSYF 位置 1。在停止和待机模式下不执行复制。退出这些模式时，影子寄存器将在最多 2 个 RTCCLK 周期后更新。

当应用程序读取日历寄存器时，它访问影子寄存器的内容。可以通过在 RTC_CTRL 寄存器中 BYPS 位置 1 来直接访问日历寄存器。默认情况下，这个位被清零，用户访问影子寄存器。

在 BYPS=0 模式下读取 RTC_SUBS、RTC_TSH 或 RTC_DATE 寄存器时，APB 时钟（f_{APB}）的频率必须至少是 RTC 时钟（f_{RTCCLK}）频率的 7 倍。影子寄存器通过系统复位来复位。

14. 2. 5 可编程闹钟

RTC 单元提供可编程的闹钟:闹钟 A 和闹钟 B。以下是对闹钟 A 的描述，也同样适用于闹钟 B。可通过 RTC_CTRL 寄存器中的 ALAEN 位来使能可编程闹钟功能。如果日历亚秒、秒、分钟、小时、日期与闹钟寄存器 RTC_ALRMASS 和 RTC_ALARMA 中编程的值相匹配，则 ALAF 标志会被置 1。可通过 RTC_ALARMA 寄存器的 MASKx 位以及 RTC_ALRMASS 寄存器的 MASKSSBx 位单独选择各日历字段。也可通过 RTC_CTRL 寄存器中的 ALAIEN 位来使能闹钟中断。

注意:如果选择了秒字段（在 RTC_ALARMA 中复位 MASK1 位），则 RTC_PRE 寄存器中设置的同步预分频器分频系数必须至少为 3，以确保执行正确。

可以将闹钟 A 和闹钟 B（如果在 RTC_CTRL 寄存器中启用了 OUTSEL[1:0]）映射到 RTC_ALARM 输出。RTC_ALARM 输出极性可以通过寄存器 RTC_CTRL 中位 OPOL 来配置。

14. 2. 6 周期性自动唤醒

周期唤醒标志是由一个 16 位可编程自动重载递减计数器生成。唤醒计时器的范围可以扩展到 17 位。唤醒功能是通过 RTC_CTRL 寄存器中的 WTEN 位来启用的。

这唤醒输入时钟如下:

- 2、4、8 或 16 分频的 RTC 时钟（RTCCLK）。

  当 RTCCLK 是 LSE（32.768KHz）时，可配置的唤醒中断周期，从 122 微秒到 32 秒，分辨率低至到 61 微秒

- ck_spre（通常为 1Hz 内部时钟）

  当 ck_spre 频率为 1Hz 时，可得到的唤醒时间为 1 秒到 18 小时左右，分辨率为 1 秒。从 1 小时到 18 小时，设置 WKUPSEL [2:1] = 10。

  当在 RTC_CTRL 寄存器中设置 WTIEN 位来启用周期性唤醒中断时，可以从低功耗模式退出设备。

如果通过 RTC_CTRL 寄存器位 OUTSEL[1:0]启用了周期性唤醒标志，则可以将周期性唤醒标志传到 RTC_ALARM 输出。RTC_ALARM 输出极性可以通过 RTC_CTRL 寄存器中的 OPOL 位来配置。系统复位、

低功耗模式（休眠、停机、待机）对唤醒计时器没有影响。

14.2.7 RTC 寄存器写保护

系统复位后，RTC 寄存器通过清除 PWR_CTRL 寄存器中的 DBKP 位（请参阅电源控制部分）来防止非法写入访问。DBKP 位必须置 1 来启用 RTC 寄存器的写访问。Backup 域复位后，除 RTC_CTRL、RTC_OPT、RTC_TSCWKUPCTRL、RTC_TSCWKUPCNT 寄存器外，其它所有有写权限的 RTC 寄存器都是写保护的。通过向写保护寄存器 RTC_WRP 写入一个密钥，来使能对 RTC 寄存器的写入。除 RTC_CTRL、RTC_OPT、RTC_TSCWKUPCTRL、RTC_TSCWKUPCNT 外，所有 RTC 寄存器都需要以下步骤来解锁写保护：

- 将 0xCA 写入 RTC_WRP 寄存器
- 将 0x53 写入 RTC_WRP 寄存器。

写入错误的密钥将重新激活写入保护。然而，FSM 只检查对这个寄存器的写操作，同时我们可以选择对其他寄存器进行写操作。保护机制不受系统复位影响

14.2.8 日历初始化和配置

要对初始时间和日期日历值（包括时间格式和预分配器配置）进行编程，需要以下顺序：

- 在 RTC_INITSTS 寄存器中将 INITM 位置 1，以进入初始化模式。在此模式下，将停止日历计数器并且其值可更新
- 对 RTC_INITSTS 寄存器位 INITF 进行轮询。当 INITF 置 1 时进入初始化阶段模式。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）
- 要为日历计数器生成一个 1Hz 的时钟基准，请在 RTC_PRE 寄存器中对两个预分频系数进行编程
- 在影子寄存器 （RTC_TSH 和 RTC_DATE）中加载初始时间和日期值，并通过 RTC_CTRL 寄存器中的 HFMT 位配置时间格式（12 或 24 小时）
- 通过清除 INITM 位退出初始化模式。然后自动加载实际的日历计数器值，并在 4 个 RTCCLK 时钟周期后重新开始计数

  初始化序列完成后，日历开始计数

14.2.9 夏令时时间配置

夏令时管理是通过 RTC_CTRL 寄存器的位 SU1H、AD1H 和 BAKP 来管理的。使用 SU1H 或 AD1H，软件只需要单次操作便可在日历中减少或增加一个小时，而无需经过初始化过程。

此外，软件还可以使用 BAKP 位来记录是否曾经执行过此操作。

14.2.10 闹钟配置

请按照下面的顺序配置闹钟 A 和闹钟 B：

- 清除寄存器 RTC_CTRL 中位 ALxEN 禁止闹钟 A/闹钟 B
- 配置闹钟 x 寄存器（RTC_ALRMxSS/RTC_ALARMx）
- 寄存器 RTC_CTRL 中位 ALxEN 置 1，使能闹钟 x。

注意：由于时钟同步缘故，RTC_CTRL 寄存器的每次更改需要在大约 2 个 RTCCLK 时钟周期后执行。

14. 2. 11 唤醒定时器配置

唤醒定时器自动重载值按照下面顺序配置：

- 清除寄存器 RTC_CTRL 位 WTEN 以禁用唤醒定时器
- 对 RTC_INITSTS 中位 WTWF 进行轮询直到被置 1，以确保允许访问唤醒自动重新加载定时器和 WKUPSEL[2:0]位。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）
- 配置唤醒自动重新加载值 WKUPT[15:0]，唤醒时钟选择（RTC_CTRL 中的 WKUPSEL[2:0]位）。将 RTC_CTRL 中位 WTEN 置 1 以再次启用定时器。唤醒定时器重新开始计数。由于时钟同步的缘故，WTWF 位是在清除 WTEN 之后的 2 个 RTCCLK 时钟周期后被清零。

14. 2. 12 日历读取

1．当寄存器 RTC_CTRL 中 BYPS 控制位清零时

为正确读取 RTC 日历寄存器（RTC_SUBS、RTC_TSH 和 RTC_DATE）时，APB 时钟频率（f_{PCLK}）必须等于或大于 RTC 时钟频率（f_{RTCCLK}）的 7 倍。这确保了同步机制的安全性。

如果 APB 时钟频率小于 RTC 时钟频率的 7 倍，则软件必须读取两次日历时间和日期寄存器。如果 RTC_TSH 的第二次读取与第一次读取得到的结果相同，则可以确保数据是正确的。否则必须进行第三次读访问。在任何情况下，APB 时钟频率都不能低于 RTC 时钟频率。

每次将日历寄存器数据复制到 RTC_SUBS、RTC_TSH 和 RTC_DATE 寄存器时， RTC_INITSTS 寄存器中 RSYF 被置 1。每两个 RTCCLK 周期执行一次复制。为了确保 3 个值之间时间点的一致性，读取 RTC_SUBS 或 RTC_TSH 将锁定高阶日历影子寄存器中的值，直到读取 RTC_DATE 为止。为避免软件对日历执行读访问的时间间隔小于 2 个 RTCCLK 周期。第一次读取日历之后必须通过软件将 RSYF 清零，并且软件必须等待到 RSYF 置 1 之后才可再次读取 RTC_SUBS、RTC_TSH 和 RTC_DATE 寄存器。

从低功耗模式（停机或待机）唤醒之后，必须通过软件清除 RSYF。然后，在读取 RTC_SUBS、RTC_TSH 和 RTC_DATE 寄存器之前，软件必须等待，直到 RSYF 再次置 1。RSYF 位必须在唤醒之后而不是进入低功耗模式之前进行清零。

系统复位后，软件必须等到 RSYF 置 1 后才能读取 RTC_SUBS、RTC_TSH 和 RTC_DATE 寄存器。实际上，系统复位会将影子寄存器复位为默认值。

初始化之后，软件必须等到 RSYF 置 1 后才能读取 RTC_SUBS、RTC_TSH 和 RTC_DATE 寄存器。

同步之后，软件必须等到 RSYF 置 1 后才能读取 RTC_SUBS、RTC_TSH 和 RTC_DATE 寄存器。

2．当在 RTC_CTRL 寄存器中位 BYPS 置 1 时

读取日历寄存器可以直接从日历计数器中获得值，因此无需等待 RSF 位置 1。这在从低功耗模式（停机或待机）退出后尤其有用，因为在这些模式下影子寄存器不会更新。当 BYPS 位被设置为 1 时，如果 RTCCLK 沿发生在对寄存器的两次读访问之间，则不同寄存器的结果可能不会一致。此外，如果在读取操作期间出现 RTCCLK 沿，则其中一个寄存器的值可能不正确。软件必须分两次读取所有寄存器，然后将两次结果加以比较来确认数据是否一致和正确。此外，软件也可以只比较两次读取日历寄存器得

到的结果的最低位。

注意：当 BYPS=1 时，读取日历寄存器的指令需要一个额外的 APB 周期才能完成。

14.2.13 复位 RTC

所有可用的系统复位源将日历影子寄存器（RTC_SUBS、RTC_TSH 和 RTC_DATE）和 RTC 初始化状态寄存器（RTC_INITSTS）的一些位复位为其默认值。

相反，以下寄存器通过 Backup 域复位被复位为其默认值，并且不受系统复位的影响。RTC 当前日历寄存器,RTC 控制寄存器（RTC_CTRL）,预分频器寄存器（RTC_PRE）, RTC 校准寄存器（RTC_CALIB），RTC 的时间戳寄存器（RTC_TSSS, RTC_TST 和 RTC_TSD）,唤醒定时器寄存器（RTC_WKUPT）,闹钟和闹钟 B 寄存器(RTC_ALRMASS / RTC_ALARMA 和 RTC_ALRMBSS / RTC_ALARMB),并选项寄存器(RTC_OPT)。

此外，当由 LSE 提供时钟时，如果复位源并非 Backup 域复位源（有关不受系统复位影响的 RTC 时钟源列表的详细信息，请参见“复位和时钟控制器”的“RTC 时钟”部分），则 RTC 将在系统复位时保持运行状态。发生 Backup 域复位时, RTC 会停止工作, 并且所有 RTC 寄存器都会设置为各自的复位值。

14.2.14 RTC 亚秒寄存器位移操作

RTC 可以与高精度的远程时钟同步。在读取亚秒字段（RTC_SUBS 或 RTC_TSSS）之后，可以计算远程时钟和 RTC 所维护的时间之间的精确偏移量。然后可以使用 RTC_SCTRL 将其时钟移动零点几秒，从而调整 RTC 以消除这种偏移。

RTC_SUBS 包含同步预分频计数器的值。这允许计算 RTC 所维护的精确时间，分辨率精确到 1 / (DIVS + 1)秒。因此，可以通过增大同步预分频的值（DIVS[14:0]）来提高分辨率。DIVS 设置为 0 x7FFF 时，可得到最大分辨率（30.52us，时钟频率为 32768 Hz 时钟）。

但是，提高 DIVS 意味着必须降低 DIVA，以便将同步预分配器输出保持在 1 Hz。这样，异步预分频器的输出频率会增大，那么会增加 RTC 的动态功耗。

可以使用 RTC 平移控制寄存器 （RTC_SCTRL） 对 RTC 进行微调。可以用大小为 1/(DIVS + 1) 秒分辨率对 RTC_SCTRL 进行写操作，将时钟平移（延迟或提前）最长 1 秒。在这种平移操作中，会将 ADFS[14:0] 值加到亚秒寄存器 SS[15:0] 中。这将加快时钟。如果同时将 SUB1S 位置 1，则会延迟一秒，与此同时增加的时间为零点几秒，因此将使时钟延迟。

在开始平移操作之前，用户必须检查 SS[15] = 0，以确保不会发生溢出。只要向 RTC_SCTRL 寄存器进行写操作，硬件就会设置 SHOPF 标志，表明平移操作处于挂起状态。一旦平移操作完成，这个位就会被硬件清除。

14.2.15 RTC 数字时钟精密校准

RTC 频率可以进行数字校准，分辨率约为 0.954 ppm，范围为-487.1 ppm 到+488.5 ppm。频率的校正是通过一系列的小的调整（添加和/或减去单独的 RTCCLK 脉冲）来完成的。这些微调的分布非常均匀，因此 RTC 的校准效果相当的好，即使在很短的时间内观察也是如此。

当输入频率为 32768 Hz 时，精密数字校准的周期约为 2^{20} 个 RTCCLK 脉冲或 32 秒。此周期由一个通过 RTCCLK 提供时钟信号的 20 位计数器 cal_cnt[19:0] 维持。

精密校准寄存器（RTC_CALIB）指定在 32 秒周期内要减少的 RTCCLK 时钟周期数：

- 将 CM[0]置 1 时，在 32 秒的周期内只减少一个脉冲
- 将 CM[1]置 1 时，将减少 2 个周期
- 将 CM[2]置 1 时，将减少 4 个周期
- 以此类推，直到将 CM[8]置 1，将减少 256 个周期

注意：CM[8:0] （RTC_CALIB）指定在 32 秒周期内要减少的 RTCCLK 脉冲数。使用适当分辨率时，CM 可使 RTC 频率减少最多 487.1 ppm，而 CP 可用于使频率增加 488.5 ppm。将 CP 置"1"，可每隔 2^{11} 个 RTCCLK 周期有效插入一个额外的 RTCCLK 脉冲，这意味着每 32 秒周期可增加 512 个时钟。

与 CM 和 CP 配合使用时，可在 32 秒周期内增加一个范围为 -511 到 +512 RTCCLK 周期的偏差，对应的校准范围为 -487.1 ppm 到 +488.5 ppm，分辨率约为 0.954 ppm。

若输入频率 （F_{RTCCLK}） 已知，可通过以下公式计算有效校准频率 （F_{CAL}）：

$$F_{CAL} = F_{RTCCLK} * [1 + (CP * 512 - CM) /(2^{20} + CM – CP * 512)]$$

当 DIVA<3 校准时

当异步预分频值（RTC_PRE 寄存器中的 DIVA 位）小于 3 时，CP 位将不能置 1。如果 CP 已经被置 1，并且 DIVA 位的值小于 3，那么 CP 将被忽略，并且校准操作就好像 CP 等于 0 一样。

要在 DIVA 小于 3 的条件下执行校准，应降低同步预分频器值 （DIVS） 以便每秒内可加速 8 个 RTCCLK 时钟周期，这意味着每 32 秒可增加 256 个时钟周期。因此，仅使用 CM 位，可在每 32 秒内有效增加 255 到 256 个时钟脉冲（对应的校准范围为 243.3 ppm 到 244.1 ppm）。

在标称 RTCCLK 频率 32768 Hz 下，当 DIVA 等于 1 时（分频系数为 2），应将 DIVS 设置为 16379 而不是 16383（少 4）。唯一相关的其他情况是，当 DIVA 等于 0 时，应将 DIVS 设置为 32759 而不是 32767（少 8）。

如果以这种方式减少 DIVS，则采用以下公式计算校准输入时钟的有效频率：

$F_{CAL} = F_{RTCCLK} * [1 + (256 - CM) / (2^{20} + CM - 256)]$

在这种情况下，如果 RTCCLK 恰好为 32768.00 Hz，则当 CM[7:0] 等于 0x100 时（CM 范围的中值），说明设置正确。

验证 RTC 校准

通过测量 RTCCLK 的精确频率，计算出正确的 CM 和 CP 值以保证 RTC 的精度。另外提供一个可选的 1Hz 输出，用来测量和验证 RTC 精度。

在有限的时间间隔内测量 RTC 的精确频率，根据数字校准周期与测量周期的对齐方式，在测量周期内可能导致最多 2 个 RTCCLK 时钟周期的测量误差。但是，如果测量周期与校准周期相同，则可以消除这种测量误差。在这种情况下，唯一观察到的误差是由于数字校准的分辨率引起的误差。

- 默认情况下，校准周期为 32 秒

  使用这种模式，在精确的 32 秒内测量 1Hz 输出的精度，可以保证测量误差值在 0.477 ppm 内（由于校准分辨率的限制，在 32 秒内为 0.5 个 RTCCLK 周期）。

- RTC_CALIB 寄存器的 CW16 位置 1，以强制 16 秒的校准周期

  在这种情况下，可以在 16 秒内测量 RTC 精度，最大误差为 0.954 ppm（16 秒内为 0.5 个 RTCCLK 周期）。然而，由于校准分辨率降低，长期的 RTC 精度也降低到 0.954 ppm。CW16 置 1 时，CM [0] 位始终为 0。

- RTC_CALIB 寄存器的 CW8 位置 1，以强制 8 秒的校准周期

  在这种情况下，RTC 精度可以在 8 秒内测量，最大误差为 1.907 ppm （8 秒内为 0.5 个 RTCCLK 周期超过）。长期的 RTC 精度也降低到 1.907 ppm。当 CW8 置 1 时，CM[1:0]位将始终保持为 00。

动态重校准

当寄存器 RTC_INITSTS 中位 INITF=0 时，校准寄存器（RTC_CALIB）可以动态更新，使用以下过程：

- 轮询寄存器 RTC_INITSTS 位 RECPF（重新校准待定标志）
- 如果该标志为 0，则在必要时将新值写入到 RTC_CALIB。然后将 RECPF 自动置 1
- 在对 RTC_CALIB 进行写操作后的三个 ck_apre 周期内，新的校准设置将生效

14. 2. 16 时间戳功能

通过将 RTC_CTRL 寄存器的 TSEN 位置 1 或 TSIEN 来启用时间戳。当在 RTC_TS 引脚上检测到时间戳事件时，日历将保存在时间戳寄存器（RTC_TSSS、RTC_TST、RTC_TSD）中。当一个时间戳事件发生时，在 RTC_INITSTS 寄存器中时间戳标志位（TISF）置 1。如果在时间戳标志（TISF）已经置 1 的条件下检测到新的时间戳事件，则将时间戳溢出标志（TISOVF）标志置 1，时间戳寄存器（RTC_TST 和 RTC_TSD）将保持前一个事件的结果。

再发生同步过程引发的时间戳事件后，TISF 在 2 个 RTC_CLK 周期置 1。TISOVF 的产生中不存在延迟。这意味着如果两个时间戳事件非常接近， TISOVF 可能为“1”，而 TISF 为“0”。 因此,建议只在检测到 TISF 为“1”后再轮询 TISOVF。

如果在 TISF 位清零后紧接着发生时间戳事件，则 TISF 和 TISOVF 位都将置 1。为防止在时间戳事件发生的同时屏蔽该事件，除非已将 TISF 位读取为“1”，否则应用程序不得将“0”写入 TISF 位。

14. 2. 17 校准时钟输出

将 RTC_CTRL 寄存器中的 COEN 位置 1 时，会在 RTC_CALIB 器件输出上提供一个参考时钟。如果 RTC_CTRL 寄存器中的 CALOSEL 位复位且 DIVA = 0x7F，则 RTC_CALIB 频率为 $f_{RTCCLK}/64$。这相当于 RTCCLK 频率为 32.768 KHz 时，256 Hz 的校准输出。RTC_CALIB 占空比是不规则的，下降沿上存在轻微抖动。因此推荐使用上升沿。

如果 CALOSEL 置 1 且“DIVS+1”为 256 的非零整数倍（比如：DIVS[7:0] = 0xFF），则 RTC_CALIB 频率为 $f_{RTCCLK}/(256 * (DIVA+1))$。这相当于 RTCCLK 频率为 32.768 KHz 时， 1 Hz 的校准输出，其中预分频器为默认值（DIVA = Ox7F、DIVS = 0xFF）。1 Hz 输出会在进行平移操作时受到影响，并且可能在平移操作期间发生翻转 （SHOPF = 1）。

注意：选择 RTC_CALIB 或 RTC_ALARM 输出时，RTC_OUT 引脚会自动配置为输出。CALOSEL 位清零时，RTC_CALIB 输出为异步预分频器的第 6 级输出。CALOSEL 位置 1 时，RTC_CALIB 输出为同步预分频器的第 8 级输出。

14. 2. 18 闹钟输出

RTC_CTRL 寄存器中的 OUTSEL[1:0] 控制位用于激活闹钟输出 RTC_ALARM，以及选择输出的功能。这些功能可反映 RTC_INITSTS 寄存器中相应标志的内容。

输出的极性由 RTC_CTRL 中的 OPOL 控制位确定，这样当 OPOL 置 1 时会输出选定标志位的相反值。

使用 RTC_OPT 寄存器中的控制位 TYPE 可将 RTC_ALARM 引脚配置为输出开漏或输出上拉。

注意：使能 RTC_ALARM 输出之后，其优先级高于 RTC_CALIB（与 COEN 位无关，该位必须保持清零）。
选择 RTC_CALIB 或 RTC_ALARM 输出时，RTC_OUT 引脚会自动配置为输出。

14. 2. 19 TSC 唤醒配置

TSC 唤醒模块默认不启用。用户通过将 RTC_TSCWKUPCTRL 寄存器中位 WKUPEN 置 1 来启用 TSC 唤醒模块。在启用 WKUP_EN 之前，用户还可以重装 RTC_TSCWKUPCTRL 寄存器的值。

通过对 RTC_TSCWKUPCTRL 寄存器中位 WAKUPCNF 进行配置，可以让外设进入配置模式。此外，只有在前一个写操作完成时，才会启用对 RTC_TSCWKUPCNT 寄存器的写操作。为了使软件能够检测这种情况，在 RTC_TSCWKUPCTRL 寄存器中提供了 WKUPOFF 状态位，以查询是否正在更新寄存器。只有当 WKUPOFF 状态位值为 1 时，才能将新值写入 RTC_TSCWKUPCNT 寄存器。

配置如下：

- 轮询 WKUPOFF，直到它的值为 1
- 配置 WKUPCNF 位进入配置模式
- 写入 RTC_TSCWKUPCNT 寄存器
- 清除 WKUPCNF 位以退出配置模式

轮询 WKUPOFF，直到它的值为 1 时才表明写操作的结束。

14. 2. 20 RTC 低功耗模式

模式	描述
SLEEP	无影响 RTC 中断可使器件退出睡眠模式
STOP0	当 RTC 时钟源为 LSE 或 LSI 时，RTC 保持工作状态。RTC 闹钟、周期性唤醒、时间戳事件会使器件退出 STOP 模式。
STOP2	当 RTC 时钟源为 LSE 或 LSI 时，RTC 保持工作状态。RTC 闹钟、周期性唤醒、时间戳事件会使器件退出 STOP 模式。

模式	描述
STANDBY	当 RTC 时钟源为 LSE 或 LSI 时，RTC 保持工作状态。RTC 闹钟、周期性唤醒、时间戳事件会使器件退出待机模式。

14.3 RTC 寄存器

14.3.1 RTC 寄存器地址映像

表 14-1 RTC 寄存器地址映像和复位值

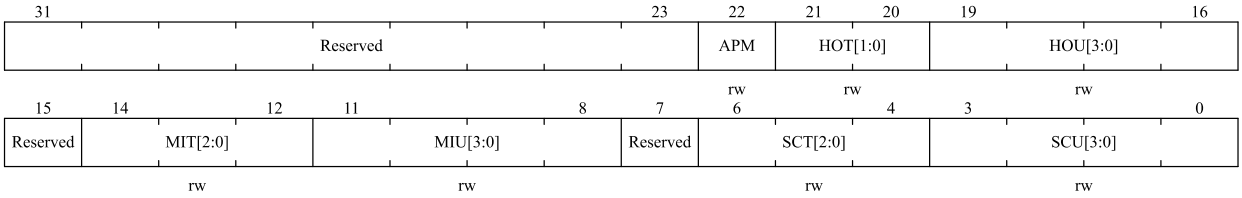
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	RTC_TSH	Reserved									APM	HOT[1:0]		HOU[3:0]				Reserved	MIT[2:0]			MIU[3:0]				Reserved	SCT[2:0]			SCU[3:0]			
	Reset Value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
004h	RTC_DATE	Reserved								YRT[3:0]				YRU[3:0]				WDU[2:0]			MOT	MOU[3:0]				Reserved	DAT[1:0]		DAU[3:0]			
	Reset Value									0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1		0	0	0	0	0	0	1
008h	RTC_CTRL	Reserved								COEN	OUTSEL[1:0]		OPOL	CALOSEL	BAKP	SU1H	AD1H	Reserved	WTIEN	ALBIEN	ALAIEN	TSEN	WTEN	ALBEN	ALAEN	Reserved	HFMT	BYPS	Reserved	TSPOL	WKUPSEL[2:0]		
	Reset Value									0	0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0		0	0	0	0
00Ch	RTC_INITSTS	Reserved															RECPF	Reserved			TISOVF	TISF	WTF	ALBF	ALAF	INITM	INITF	RSYF	INITSF	SHOPF	WTWF	ALBWF	ALAWF
	Reset Value																0				0	0	0	0	0	0	0	0	0	0	1	1	1
010h	RTC_PRE	Reserved								DIVA[6:0]								Reserved	DIVS[14:0]														
	Reset Value									1	1	1	1	1	1	1		0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
014h	RTC_WKUPT	Reserved																WKUPT[15:0]															
	Reset Value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
01Ch	RTC_ALARMA	MASK4	WKDSEL	DTT[1:0]		DTU[3:0]				MASK3	APM	HOT[1:0]		HOU[3:0]				MASK2	MIT[2:0]			MIU[3:0]				MASK1	SET[2:0]			SEU[3:0]			
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
020h	RTC_ALARMB	MASK4	WKDSEL	DTT[1:0]		DTU[3:0]				MASK3	APM	HOT[1:0]		HOU[3:0]				MASK2	MIT[2:0]			MIU[3:0]				MASK1	SET[2:0]			SEU[3:0]			
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
024h	RTC_WRP	Reserved																PKEY[7:0]															
	Reset Value																									0	0	0	0	0	0	0	0
028h	RTC_SUBS	Reserved																SS[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	RTC_SCTRL	SUB1S	Reserved															ADFS[14:0]															
	Reset Value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
030h	RTC_TST	Reserved									APM	HOT[1:0]		HOU[3:0]				Reserved	MIT[2:0]			MIU[3:0]				Reserved	SET[2:0]			SEU[3:0]			
	Reset Value																										0	0	0	0	0	0	0
034h	RTC_TSD	Reserved								YRT[3:0]				YRU[3:0]				WDU[2:0]			MOT	MOU[3:0]				Reserved	DAT[1:0]		DAU[3:0]			
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
038h	RTC_TSSS	Reserved																SSE[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03Ch	RTC_CALIB	Reserved																CP	CW8	CW16	Reserved				CM[8:0]								
	Reset Value																									0	0	0	0	0	0	0	0
044h	RTC_ALRMASS	Reserved				MASKSSB[3:0]				Reserved								SSV[14:0]															
	Reset Value					0	0	0	0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
048h	RTC_ALRMBSS	Reserved				MASKSSB[3:0]				Reserved								SSV[14:0]															
	Reset Value					0	0	0	0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04Ch	RTC_OPT	Reserved																															TYPE
	Reset Value																																0
064h	RTC_TSCWKUPCTRL	Reserved																												WKUPOFF	WKUPCNF	Reserved	WKUPEN
	Reset Value																													1	0		0
068h	RTC_TSCWKUPCNT	Reserved																CNT[13:0]															
	Reset Value																	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	0

14.3.2 RTC 日历时间寄存器（RTC_TSH）

偏移地址：0x00

复位值：0x0000 0000

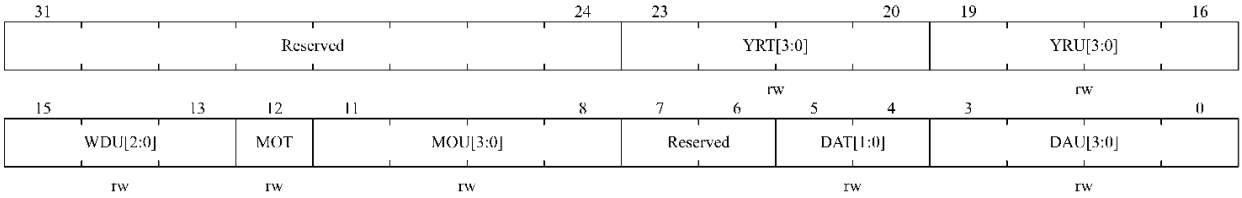


位域	名称	描述
31:23	Reserved	保留，硬件强制为 0。
22	APM	AM/PM 格式。 0：AM 格式或者 24 小时格式 1：PM 格式
21:20	HOT[1:0]	小时十位（BCD 格式）。
19:16	HOU[3:0]	小时的个位（BCD 格式）。
15	Reserved	保留，硬件强制为 0。
14:12	MIT[2:0]	分钟的十位（BCD 格式）。
11:8	MIU[3:0]	分钟的个位（BCD 格式）。
7	Reserved	保留，硬件强制为 0。
6:4	SCT[2:0]	秒的十位（BCD 格式）。
3:0	SCU[3:0]	秒的的个位（BCD 格式）。

14.3.3 RTC 日历日期寄存器（RTC_DATE）

偏移地址：0x04

复位值：0x0000 2101

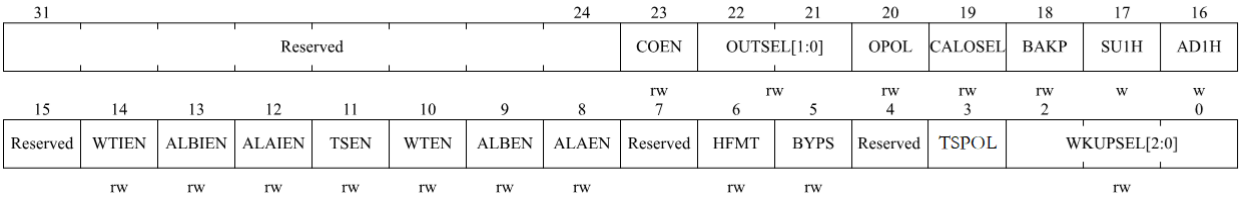


位域	名称	描述
31:24	Reserved	保留，硬件强制为 0。
23:20	YRT[3:0]	年的十位（BCD 格式）。
19:16	YRU[3:0]	年的个位（BCD 格式）。
15:13	WDU[2:0]	星期几。
12	MOT	月的十位（BCD 格式）。
11:8	MOU[3:0]	月的个位（BCD 格式）。
7:6	Reserved	保留，硬件强制为 0。
5:4	DAT[1:0]	天的十位（BCD 格式）。
3:0	DAU[3:0]	天的个位（BCD 格式）。

14.3.4 RTC 控制寄存器（RTC_CTRL）

偏移地址：0x08

复位值：0x0000 0000



位域	名称	描述
31:24	Reserved	保留，硬件强制为 0。
23	COEN	校准输出使能。 0：校准输出禁止 1：校准输出开启
22:21	OUTSEL[1:0]	输出选择位。 该位用来选择要连接到 RTC_ALARM 输出的标志。 00：输出禁止 01：闹钟 A 输出开启 10：闹钟 B 输出开启 11：唤醒输出开启
20	OPOL	输出极性位。 此位用于配置 RTC_ALARM 输出的极性。

位域	名称	描述
		0：当 ALAF/ALBF/WTF 标志位置 1（取决于 OUTSEL [1:0]）， 该引脚输出高电平 1：当 ALAF/ALBF/WTF 标志位置 1（取决于 OUTSEL [1:0]）， 该引脚输出低电平
19	CALOSEL	校准输出选择位。 当 COEN=1 时，该位选择在 RTC_CALIB 上输出哪个信号。 在 RTCCLK 为 32.768 kH 且预分频为默认值（DIVA=127 和 DIVS=255） 的条件下，这些频率有效。 0：校准输出 256Hz （默认预分频设置） 1：校准输出 1Hz（默认预分频设置）
18	BAKP	这个位可以由用户写入，以记住是否执行了夏令时的更改。
17	SU1H	减去 1 小时位（冬季时间更改）。 当设置此位时，如果当前小时不是 0，则从日历时间中减去 1 小时。这个位 总是被读取为 0。当当前小时为 0 时，设置此位无效。 0：无使用 1：用当前时间减去 1 小时。这可以用于冬季改变户外初始化模式
16	AD1H	加 1 小时位（夏季时间更改）。 设置此位后，将 1 小时添加到日历时间中。这个位总是被读作 0。 0：无使用 1：用当前时间加上 1 小时。这可以用于夏季改变户外初始化模式
15	Reserved	保留，硬件强制为 0。
14	WTIEN	唤醒定时器中断使能位。 0：唤醒定时器中断禁止 1：唤醒定时器中断开启
13	ALBIEN	闹钟 B 中断使能位。 0：闹钟 B 中断禁止 1：闹钟 B 中断开启
12	ALAIEN	闹钟 A 中断使能位。 0：闹钟 A 中断禁止 1：闹钟 A 中断开启
11	TSEN	时间戳使能位。 0：时间戳禁止 1：时间戳开启
10	WTEN	唤醒定时器使能位。 0：唤醒定时器禁止 1：唤醒定时器开启
9	ALBEN	闹钟 B 使能位。 0：闹钟 B 禁止 1：闹钟 B 开启
8	ALAEN	闹钟 A 使能位。 0：闹钟 A 禁止 1：闹钟 A 开启

位域	名称	描述
7	Reserved	保留，硬件强制为 0。
6	HFMT	小时格式位。 0：24 小时格式 1：AM/PM 格式
5	BYPS	旁路影子寄存器位。 0：日历值（从 RTC_SUBS、RTC_TSH 和 RTC_DATE 读取时）取自影子寄存器，影子寄存器每两个 RTCCLK 周期更新一次。 1：日历值（从 RTC_SUBS、RTC_TSH 和 RTC_DATE 读取时）直接从日历计数器中获取。 *注意：如果 APB1 时钟的频率小于 RTCCLK 的 7 倍，则 BYPS 必须设置为 1*
4	Reserved	保留，硬件强制为 0。
3	TSPOL	时间戳时间触发沿。 0:RTC_TS 输入上升沿生成一个时间戳事件 1:RTC_TS 输入下降沿生成一个时间戳事件 TSEN 必须在更改 TSPOL 时清零，以免造成 TISF 时间戳标志的误置位。
2:0	WKUPSEL[2:0]	唤醒时钟选择位。 000：选择 RTC/16 时钟 001：选择 RTC/8 时钟 010：选择 RTC/4 时钟 011：选择 RTC/2 时钟 10x：选择 ck_spre（通常 1Hz）时钟

14.3.5 RTC 初始状态寄存器（RTC_INITSTS）

偏移地址：0x0C
复位值：0x0000 0007

31:17	16
Reserved	RECPF
	r

15:13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TISOVF	TISF	WTF	ALBF	ALAF	INITM	INITF	RSYF	INITSF	SHOPF	WTWF	ALBWF	ALAWF
	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

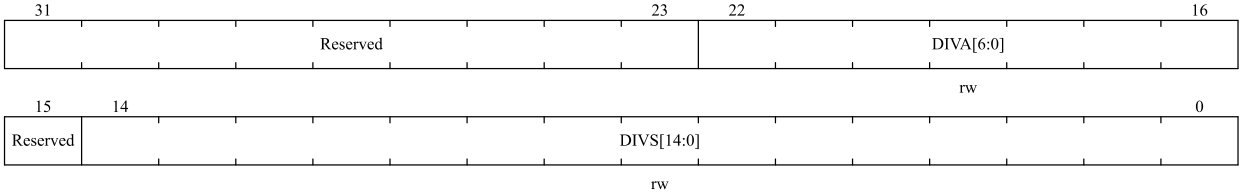
位域	名称	描述
31:17	Reserved	保留，硬件强制为 0。
16	RECPF	重新校准挂起标志位。 当软件写入 RTC_CALIB 寄存器时，RECPF 状态标志自动设置为 1，表示 RTC_CALIB 寄存器被阻塞。当校准设置开始执行后，这个位将恢复为 0。
15:13	Reserved	保留，硬件强制为 0。
12	TISOVF	时间戳溢出标志位。 当时间戳事件发生同时 TISF 已经被置 1 时，硬件将此标志置 1。建议在清除 TISF 位之后再检查并清除 TISOVF 位。否则，如果时间戳事件恰好在清除 TISF 位之前刚刚发生，则溢出事件可能会被漏掉。
11	TISF	时间戳标志位。

位域	名称	描述
		当发生时间戳事件时，硬件将设置此标志。此标志通过写入 0 被软件清除。
10	WTF	唤醒定时器标志位。 当唤醒自动重载计数器达到 0，硬件将设置此标志。此标志通过写入 0 被软件清除。在 WTF 再次设置为 1 之前，此标志必须由软件至少在 1.5 RTCCLK 周期内清除。
9	ALBF	闹钟 B 标志位。 当时间/日期寄存器（RTC_TSH 和 RTC_DATE）与闹钟 B 寄存器（RTC_ALARMB）匹配时，硬件将设置此标志。此标志通过写入 0 被软件清除。
8	ALAF	闹钟 A 标志位。 当时间/日期寄存器（RTC_TSH 和 RTC_DATE）与闹钟 A 寄存器（RTC_ALARMA）匹配时，硬件将设置此标志。此标志通过写入 0 被软件清除。
7	INITM	初始模式位。 0：自由运行模式 1：初始化模式用于配置时间和日期寄存器（RTC_TSH 和 RTC_DATE）和预分频寄存器（RTC_PRE）。当 INITM 被复位时，计数器将停止并从新值开始计数。
6	INITF	初始标志位。 当这个位设置为 1 时，RTC 处于初始化状态，可以更新时间、日期和预分频寄存器。 0：日历寄存器更新禁止 1：日历寄存器更新允许
5	RSYF	寄存器同步标志位。 每次将日历寄存器复制到影子寄存器（RTC_SUBSx、RTC_TSHx 和 RTC_DATEx）时，硬件将此位置 1。当移位操作处于挂起状态（SHOPF=1）或旁路阴影寄存器模式（BYPS=1）时，硬件将在初始化模式中清除该位。这个位也可以被软件清除。在初始化模式下，该位通过软件或硬件清除。 0：日历影子寄存器尚未同步 1：日历影子寄存器同步
4	INITSF	初始状态标志位。 当日历年字段不等于 0 （备份域复位状态）时，由硬件设置此位。 0：日历没有被初始化 1：日历已经被初始化
3	SHOPF	平移操作挂起标志位。 当向 RTC_SCTRL 寄存器写入一个平位操作时，硬件立即设置此标志。当执行相应的平移操作时，硬件将清除它。写入 SHOPF 位执行写入操作不起作用。 0：没有位移操作挂起 1：有位移操作挂起
2	WTWF	唤醒定时器写标志位。

位域	名称	描述
		在 RTC_CTRL 中将 WTEN 位设置为 0，最多 2 个 RTCCLK 之后，硬件将该位置 1。同理，在 WTEN 位置 1，最多 2 个 RTCCLK 之后，硬件将该位清除。 当 WTEN=0，WTWF=1 时，唤醒定时器值可以更改。 0：唤醒时间配置更新不允许 1：唤醒时间配置更新允许
1	ALBWF	闹钟 B 写标志。 当 RTC_CTRL 中的 ALBEN 位设置为 0，同时闹钟 B 值可更改时，硬件将该位置 1。它在初始化模式下被硬件清除。 0：闹钟 B 更新不允许 1：闹钟 B 更新允许
0	ALAWF	闹钟 A 写标志。 当 RTC_CTRL 中的 ALAEN 位设置为 0，同时闹钟 A 可更改时，硬件将该位置 1。它在初始化模式下被硬件清除。 0：闹钟 A 更新不允许 1：闹钟 A 更新允许

14.3.6 RTC 预分频寄存器（RTC_PRE）

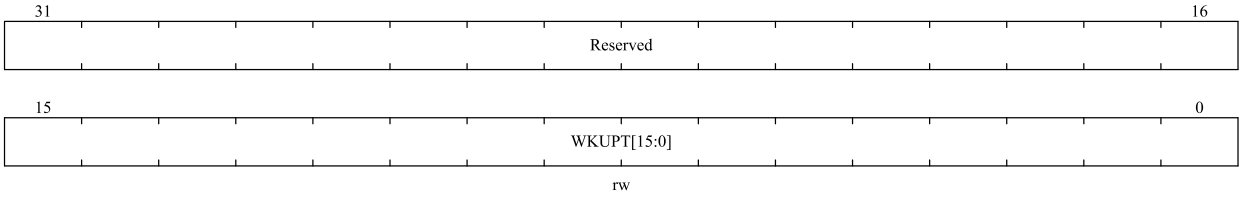
偏移地址：0x10
复位值：0x007F 00FF



位域	名称	描述
31:23	Reserved	保留，硬件强制为 0。
22:16	DIVA[6:0]	异步分频参数位。 公式如下： ck_apre 频率 = RTCCLK 频率/(DIVA+1)
15	Reserved	保留，硬件强制为 0。
14:0	DIVS[14:0]	同步分频位。 公式如下： ck_spre 频率 = ck_apre 频率/(DIVS+1)

14.3.7 RTC 唤醒定时器寄存器（RTC_WKUPT）

偏移地址：0x14
复位值：0x0000 FFFF

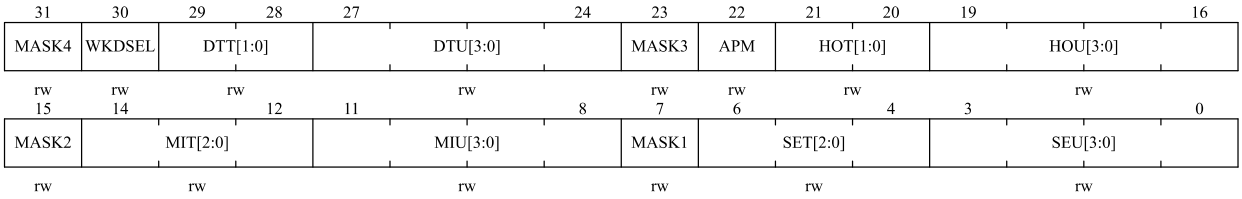


位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	WKUPT[15:0]	唤醒自动重装值位。 当唤醒定时器被启用（WTEN 设置为 1）时，WTF 标志位每隔 ck_wt 周期（WKUPT[15:0] + 1）被置 1 一次。ck_wt 周期是通过 RTC_CTRL 寄存器的 WKUPSEL[2:0]位选择的。当 WKUPSEL [2] = 1 时，唤醒定时器变成 17 位，WKUPSEL [1]等效于 WKUPT[16],即要重装到定时器的最高有效位。 WTF 第一次置 1 发生在 WTEN 置 1 之后 （WKUPT +1） 个 ck_wt 周期。禁止在 WKUPSEL [2:0]=011(RTCCLK/2) 时将 WKUPT [15:0] 设置为 0x0000。 *注意:* *该寄存器修改（比如第二次设置或以后设置），需要在唤醒中断里修改，否则修改后的设置不会立即生效，会在下次唤醒后生效;* *特别需要注意，当 RTC_CTRL 的 WKUPSEL[2:0]设为 010 时，修改后的设置不会立即生效，会在下个周期唤醒后生效;*

14.3.8 RTC 闹钟 A 寄存器（RTC_ALARMA）

偏移地址：0x1C

复位值：0x0000 0000



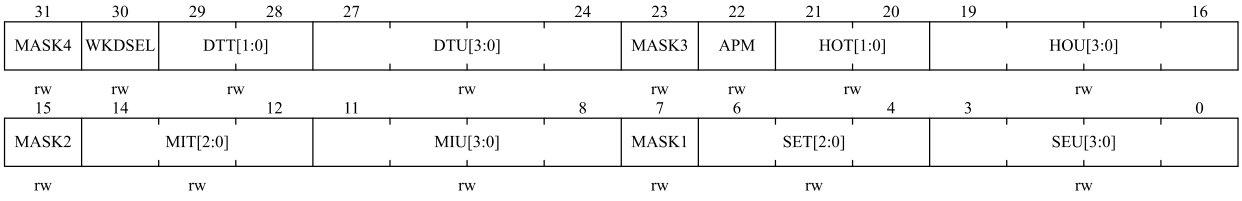
位域	名称	描述
31	MASK4	闹钟 A 日期掩码位。 0：如果日期/日匹配，则闹钟 A 置 1 1：在闹钟 A 比较中，日期/日无关
30	WKDSEL	日期/星期选择。 0：选择为日期（DTU[3:0]代表日期的个位，DTT[1:0]代表日期的十位） 1：选择为星期（DTU[3:0]代表星期几，DTT[1:0]设置无功能）
29:28	DTT[1:0]	WKDSEL 为 0 时，此两位指示日期的十位（BCD 格式） WKDSEL 为 1 时，此两位无功能
27:24	DTU[3:0]	WKDSEL 为 0 时，此两位指示日期的个位（BCD 格式） WKDSEL 为 1 时，此两位指示星期几
23	MASK3	闹钟 A 小时掩码位。 0：如果小时匹配，则闹钟 A 置 1

位域	名称	描述
		1：在闹钟 A 比较中，小时无关
22	APM	AM/PM 符号位。 0：AM 或 24 小时制 1：PM
21:20	HOT[1:0]	小时的十位（BCD 格式）。
19:16	HOU[3:0]	小时的个位（BCD 格式）。
15	MASK2	闹钟 A 分钟掩码位。 0：如果分钟匹配，则闹钟 A 置 1 1：在闹钟 A 比较中，分钟无关
14:12	MIT[2:0]	分钟的十位（BCD 格式）。
11:8	MIU[3:0]	分钟的个位（BCD 格式）。
7	MASK1	闹钟 A 秒掩码位。 0：如果秒匹配，则闹钟 A 置 1 1：在闹钟 A 比较中，秒无关
6:4	SET[2:0]	秒的十位（BCD 格式）。
3:0	SEU[3:0]	秒的个位（BCD 格式）。

14.3.9 RTC 闹钟 B 寄存器（RTC_ALARMB）

偏移地址：0x20

复位值：0x0000 0000

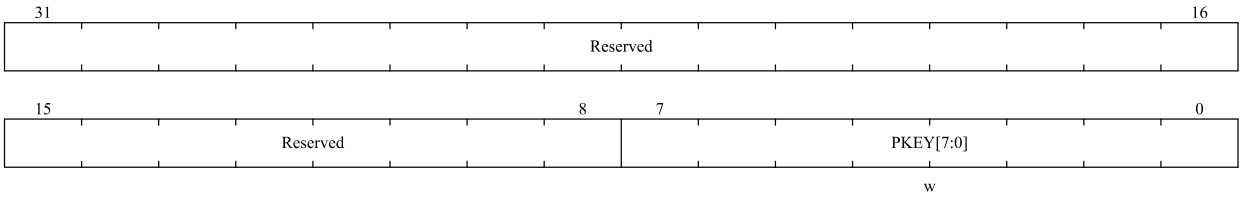


位域	名称	描述
31	MASK4	闹钟 B 日期掩码位。 0：如果日期/日匹配，则闹钟 B 置 1 1：在闹钟 B 比较中，日期/日无关
30	WKDSEL	星期几选择位。 0：DTU[3:0]代表日期的个位 1：DTU[3:0]代表星期几 DTT[1:0]为无关位
29:28	DTT[1:0]	日期的十位（BCD 格式）。
27:24	DTU[3:0]	日期的个位（BCD 格式）
23	MASK3	闹钟 B 小时掩码位。 0：如果小时匹配，则闹钟 B 置 1 1：在闹钟 B 比较中，小时无关
22	APM	AM/PM 符号位。 0：AM 或 24 小时制 1：PM

位域	名称	描述
21:20	HOT[1:0]	小时的十位（BCD 格式）。
19:16	HOU[3:0]	小时的个位（BCD 格式）。
15	MASK2	闹钟 B 分钟掩码位。 0：如果分钟匹配，则闹钟 B 置 1 1：在闹钟 B 比较中，分钟无关
14:12	MIT[2:0]	分钟的十位（BCD 格式）。
11:8	MIU[3:0]	分钟的个位（BCD 格式）。
7	MASK1	闹钟 B 秒掩码位。 0：如果秒匹配，则闹钟 B 置 1 1：在闹钟 B 比较中，秒无关
6:4	SET[2:0]	秒的十位（BCD 格式）。
3:0	SEU[3:0]	秒的个位（BCD 格式）。

14.3.10 RTC 写保护寄存器（RTC_WRP）

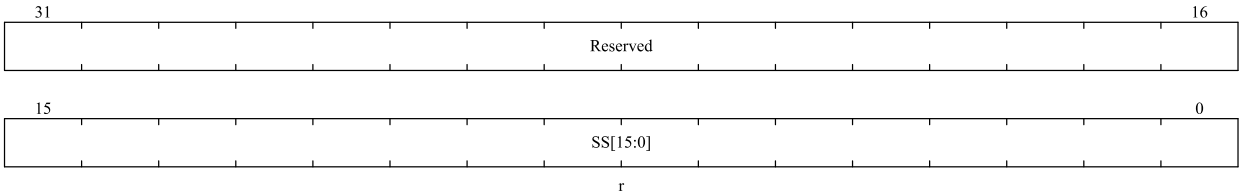
偏移地址：0x24
复位值：0x0000 0000



位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7:0	PKEY[7:0]	写保护关键字位。 可通过软件对该字节执行写操作。 读取该字节时，始终返回 0 有关如何解锁 RTC 寄存器写保护的介绍，请参考 RTC 寄存器写保护。

14.3.11 RTC 亚秒寄存器（RTC_SUBS）

偏移地址：0x28
复位值：0x0000 0000

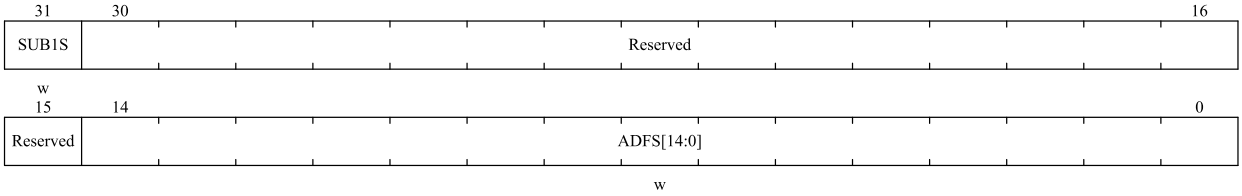


位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	SS[15:0]	亚秒值位。

位域	名称	描述
		SS[15:0]是同步预分频计数器的值。此亚秒值可根据以下公式得出： 亚秒值 = (DIVS - SS) / (DIVS + 1) *注意：仅当执行平移操作之后，SS 才能大于 DIVS。在这情况下，正确的时间/日期比 RTC_TSH/RTC_DATE 所指示的时间/日期慢一秒钟。*

14.3.12 RTC 平移控制寄存器（RTC_SCTRL）

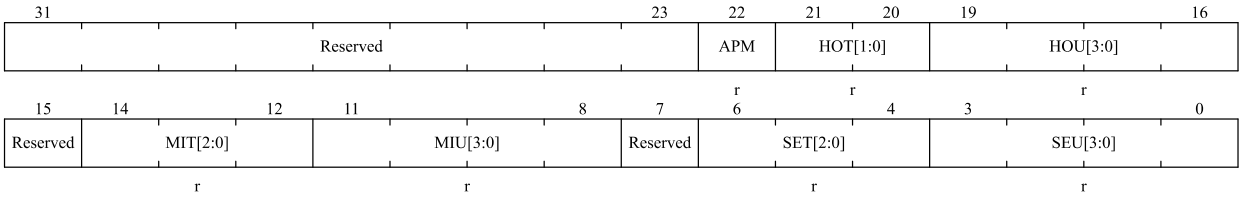
偏移地址：0x2C
复位值：0x0000 0000



位域	名称	描述
31	SUB1S	减一秒钟。 0：无影响 1：对时钟/日历减去一秒钟（亚秒值加上 255）。时钟延迟 1 秒钟 此位为只写位且始终读为 0。当平移操作挂起（RTC_INITSTS 中的 SHOPF=1）时，对该位执行写操作无作用。 此函数应与 ADFS 配合使用（请参见下文介绍），以便有效地向原子操作机制的时钟添加亚秒值。
30:15	Reserved	保留，硬件强制为 0。
14:0	ADFS[14:0]	加亚秒值。 此位为只写位且始终读为 0。当平移操作挂起（RTC_INITSTS 中的 SHOPF=1）时，对该位执行写操作无作用。 写入 ADFS 的值将被同步预分频器计数器减掉。由于该计数器递减计数，此操作可有效地从时钟加快以下时间： 加快（秒）= ADFS / (DIVS + 1) *注：对 ADFS 执行写操作之前，需要对 RTC_SUBS 寄存器进行读操作，保证 ADFS <SS;*

14.3.13 RTC 时间戳时间寄存器（RTC_TST）

偏移地址：0x30
复位值：0x0000 0000

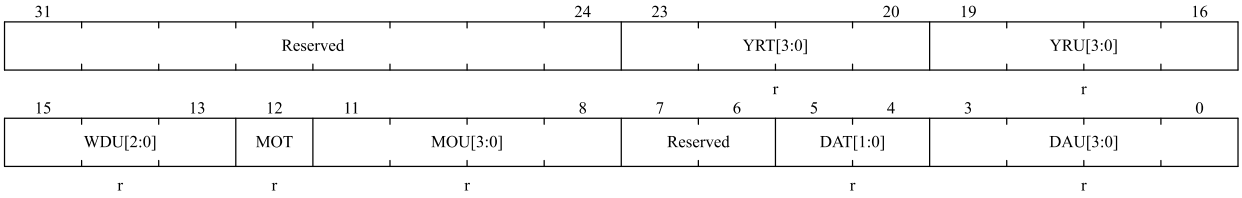


位域	名称	描述
31:23	Reserved	保留，硬件强制为 0。
22	APM	AM/PM 符号位。 0：AM 或 24 小时制 1：PM
21:20	HOT[1:0]	小时的十位（BCD 格式）。
19:16	HOU[3:0]	小时的个位（BCD 格式）。
15	Reserved	保留，硬件强制为 0。
14:12	MIT[2:0]	分钟的十位（BCD 格式）。
11:8	MIU[3:0]	分钟的个位（BCD 格式）。
7	Reserved	保留，硬件强制为 0。
6:4	SET[2:0]	秒的十位（BCD 格式）。
3:0	SEU[3:0]	秒的个位（BCD 格式）。

14.3.14 RTC 时间戳日期寄存器（RTC_TSD）

偏移地址：0x34

复位值：0x0000 0000

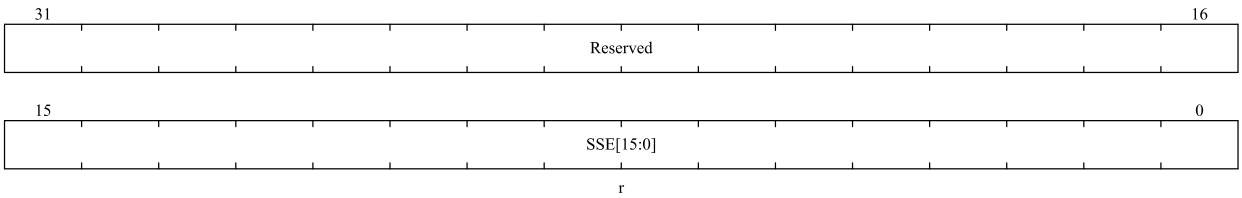


位域	名称	描述
31:24	Reserved	保留，硬件强制为 0。
23:20	YRT[3:0]	年十位（BCD 格式）。
19:16	YRU[3:0]	年个位（BCD 格式）。
15:13	WDU[2:0]	星期几的个位。
12	MOT	月份的十位（BCD 格式）。
11:8	MOU[3:0]	月份的个位（BCD 格式）。
7:6	Reserved	保留，硬件强制为 0。
5:4	DAT[1:0]	日期的十位（BCD 格式）。
3:0	DAU[3:0]	日期的个位（BCD 格式）。

14.3.15 RTC 时间戳亚秒寄存器（RTC_TSSS）

偏移地址：0x38

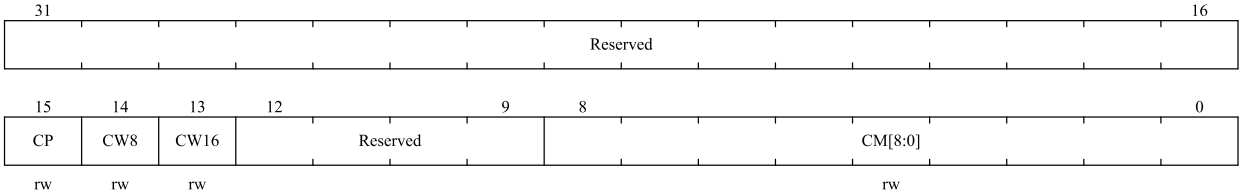
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	SSE[15:0]	亚秒值位。 当发生时间戳事件时，SSE[15:0]是同步预分频计数器的值。

14.3.16 RTC 校准寄存器（RTC_CALIB）

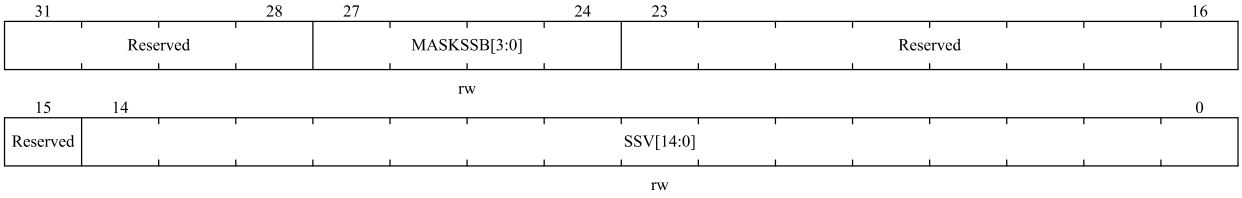
偏移地址：0x3C
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15	CP	将 RTC 的频率增加 488.5ppm 位。 此功能应与 CM 结合使用，后者在高分辨率下会降低日历的频率。如果输入频率为 32768 Hz，则在 32 秒窗口中增加的 RTCCLK 脉冲数按如下公式计算：(512 * CP) - CM 请参见 14.2.15 RTC 数字时钟精密校准。 0：不增加 RTCCLK 脉冲 1：每 2^{11} 个脉冲有效插入一个 RTCCLK 脉冲（将 RTC 的频率增加 488.5ppm）。
14	CW8	使用 8 秒校准周期位。 0：不使用 1：选择 8 秒校准周期 *注意：当 CM8 = 1 时，CM[1:0]将始终保持为'00'*
13	CW16	使用 16 秒校准周期位。 0：不使用 1：选择 16 秒校准周期，如果 CM8 = 1，则不能将该位置 1 *注意：当 CM16 = 1 时，CM[0]将始终保持为'0'*
12:9	Reserved	保留，硬件强制为 0。
8:0	CM[8:0]	负校准位。 在 2^{20} 个 RTCCLK 脉冲内屏蔽 CM 个脉冲（如果输入频率为 32768 Hz，则为 32 秒）来降低日历的频率。其分辨率为 0.9537 ppm。要提高日历的频率，则应将此功能与 CP 结合使用。

14.3.17 RTC 闹钟 A 亚秒寄存器（RTC_ALRMASS）

偏移地址：0x44
复位值：0x0000 0000

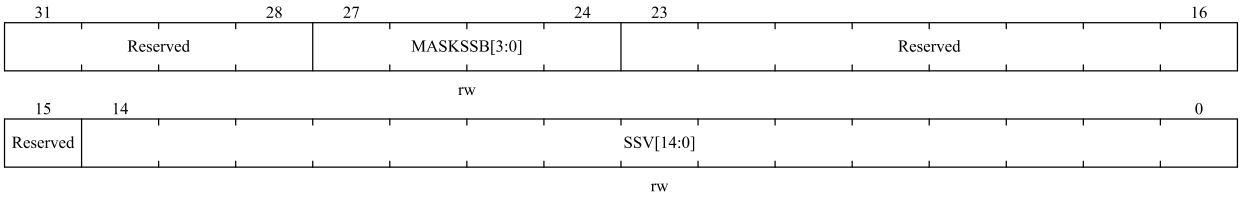


位域	名称	描述
31:28	Reserved	保留，硬件强制为 0。
27:24	MASKSSB[3:0]	屏蔽从此位开始的最高有效位。 0：不对闹钟 A 的亚秒进行比较。当秒单元递增时设置闹钟（假定其余字段均匹配）。 1：在闹钟 A 比较中，SSV[14:1] 为无关位。仅比较 SSV[0]。 2：在闹钟 A 比较中，SSV[14:2] 为无关位。仅比较 SSV[1:0]。 3：在闹钟 A 比较中，SSV[14:3] 为无关位。仅比较 SSV[2:0]。 … 12：在闹钟 A 比较中，SSV[14:12] 为无关位。比较 SSV[11:0]。 13：在闹钟 A 比较中，SSV[14:13] 为无关位。比较 SSV[12:0]。 14：在闹钟 A 比较中，SSV[14] 为无关位。比较 SSV[13:0]。 15：所有 15 个 SSV 位均进行比较，并且必须全部匹配才能激活闹钟。 同步计数器的溢出位（位 15）从不进行比较。仅当执行平移操作之后，此位才不为 0。
23:15	Reserved	保留，硬件强制为 0。
14:0	SSV[14:0]	亚秒值位。 该值与同步预分频器计数器的内容进行比较以确定是否要激活闹钟 A。仅比较位 0 到 MASKSSB-1。

14.3.18 RTC 闹钟 B 亚秒寄存器（RTC_ALRMBSS）

偏移地址：0x48

复位值：0x0000 0000

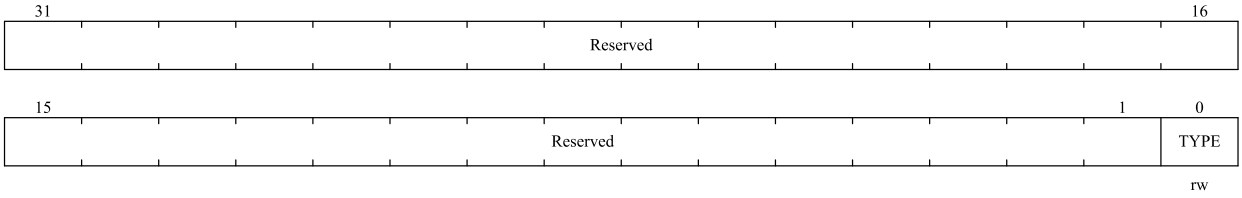


位域	名称	描述
31:28	Reserved	保留，硬件强制为 0。
27:24	MASKSSB[3:0]	屏蔽从此位开始的最高有效位。 0：不对闹钟 B 的亚秒进行比较。当秒单元递增时设置闹钟（假定其余字段均匹配）。 1：在闹钟 B 比较中，SSV[14:1] 为无关位。仅比较 SSV [0]。 2：在闹钟 B 比较中，SSV [14:2] 为无关位。仅比较 SSV [1:0]。 3：在闹钟 B 比较中，SSV [14:3] 为无关位。仅比较 SSV [2:0]。 …

位域	名称	描述
		12：在闹钟 B 比较中，SSV [14:12] 为无关位。比较 SSV [11:0]。 13：在闹钟 B 比较中，SSV [14:13] 为无关位。比较 SSV [12:0]。 14：在闹钟 B 比较中，SSV [14] 为无关位。比较 SSV [13:0]。 15：所有 15 个 SSV 位均进行比较，并且必须全部匹配才能激活闹钟。 同步计数器的溢出位（位 15）从不进行比较。仅当执行平移操作之后，此位才不为 0。
23:15	Reserved	保留，硬件强制为 0。
14:0	SSV[14:0]	亚秒值位。 该值与同步预分频器计数器的内容进行比较以确定是否要激活闹钟 B。仅比较位 0 到 MASKSSB-1。

14.3.19 RTC 选项寄存器（RTC_OPT）

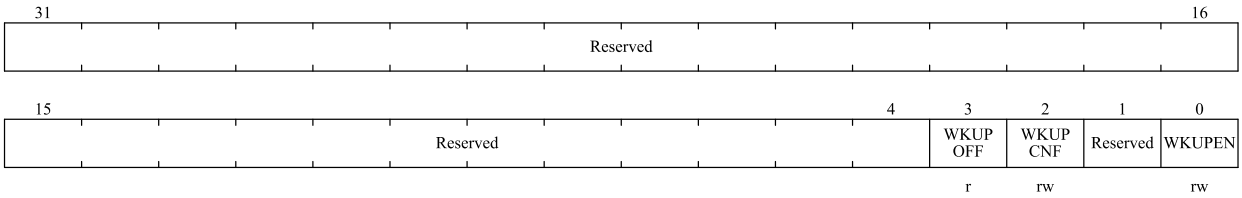
偏移地址：0x4C
复位值：0x0000 0000



位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	TYPE	PC13 上的 RTC_ALARM 输出类型位。 此位由软件置 1 和清零 0：RTC_ALARM 在映射到 PC13 上时为开漏输出 1：RTC_ALARM 在映射到 PC13 上时为推挽输出

14.3.20 RTC TSC 唤醒控制寄存器（RTC_TSCWKUPCTRL）

偏移地址：0x64
复位值：0x0000 0008



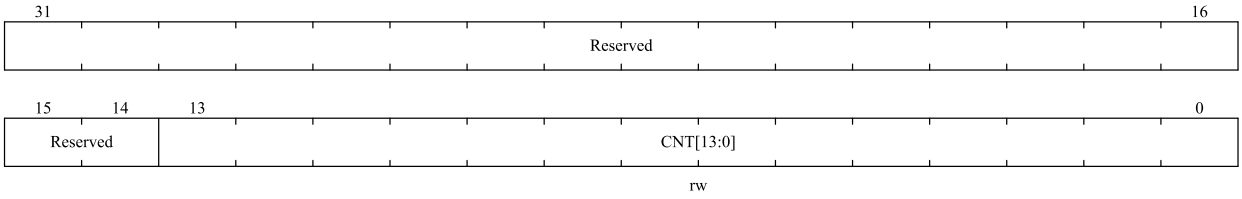
位域	名称	描述
31:4	Reserved	保留，硬件强制为 0。
3	WKUPOFF	上次写操作唤醒计数器状态位。 0：唤醒计数器上次写操作仍在进行 1：唤醒计数器上次写操作已经结束
2	WKUPCNF	唤醒配置标志位。

位域	名称	描述
		0：退出配置模式（开始更新唤醒计数值） 1：进入配置模式
1	Reserved	保留，硬件强制为 0。
0	WKUPEN	唤醒使能位。 0：禁止 1：开启

14.3.21 RTC TSC 唤醒计数器（RTC_TSCWKUPCNT）

偏移地址：0x68

复位值：0x0000 02FE



位域	名称	描述
31:14	Reserved	保留，硬件强制为 0。
13:0	CNT	唤醒计数器装载值 。 0x0~0x2fd：为无效值 0x2fe: 23.4ms …… 0x2efa : 367ms Other：为无效值 *注意:* *tsc_wakeup 模块有两个时钟源:LSE 和 LSI。* *tsc_wakeup 模块计数周期= CNT * LSE/LSI*

15 随机数（RNG）

15.1 RNG 简介

CM32M4xxR 随机数发生器由模拟随机源和数字后处理电路组成。

模拟随机源采用可靠性的电路实现方案，可产生较高速率的随机数比特流，同时可以保障随机数的质量。随机数后处理电路目的是提高随机数质量，CM32M4xxR 采用了高质量后处理电路，在不影响随机数产生速度的情况下可有效提高随机数质量。

15.2 RNG 主要特性

- 可一次性读取 32bit 随机数
- 随机源的时钟可以配置，时钟在 2MHz~16MHz 范围
- 两种模式随机数输出：真随机数、伪随机数

15.3 RNG 功能描述

随机数模块使用流程如下:

1. 设置 RNG_CTRL[1:0]为 2'b11（打开反相器环）;
2. 待反相器环起振（5us）;
3. 设置 RNG_CTRL[6:5]为 2'b01（设置生成模式 MODE 为随机源模式，开始生成随机数据种子）
4. 等待 RNG_D_RDY[0]为 1（随机数种子填充完毕）;
5. 设置 RNG_CTRL[6:5]为 2'b00（关闭随机数，RNG_DATA 中的随机数将不再更新）;
6. 设置 RNG_CTRL[6:5]（设置 MODE 为非 2'b00 开始采集随机数）;
7. 等待 RNG_D_RDY[0]为 1（32 位随机数准备完毕）;
8. 读 RNG_DATA（读取 32 位随机数）;
9. 如果要继续读随机数，回到步骤 7;

如果要重新设置随机数配置，转到步骤 5。

15.4 RNG 寄存器

15.4.1 RNG 寄存器地址映像

表 15-1 RNG 随机数模块寄存器映像和复位值

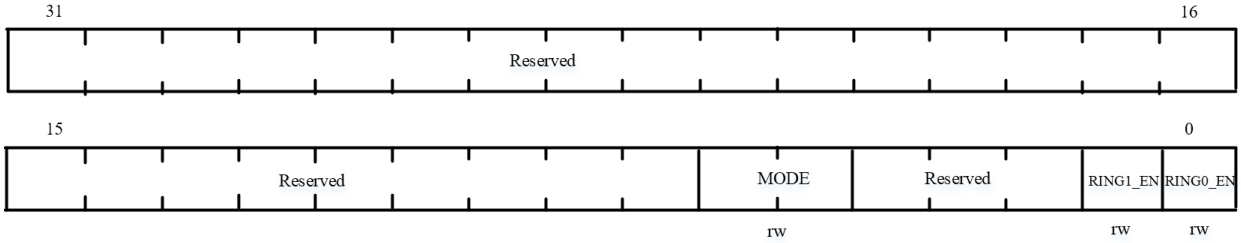
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	RNG_CTRL	Reserved																									MODE		Reserved				RING1_EN	RING0_EN
	Reset Value																										0	0	0	0	0	0	0

004h	RNG_WR_RDY	Reserved																														WR_RDY
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	RNG_DATA	RNG_DATA																														
	Reset Value	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
00Ch	RNG_D_RDY	Reserved																														D_RDY
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.4.2 随机数控制寄存器（RNG_CTRL）

偏移地址：0x00

复位值：0x0000 0003



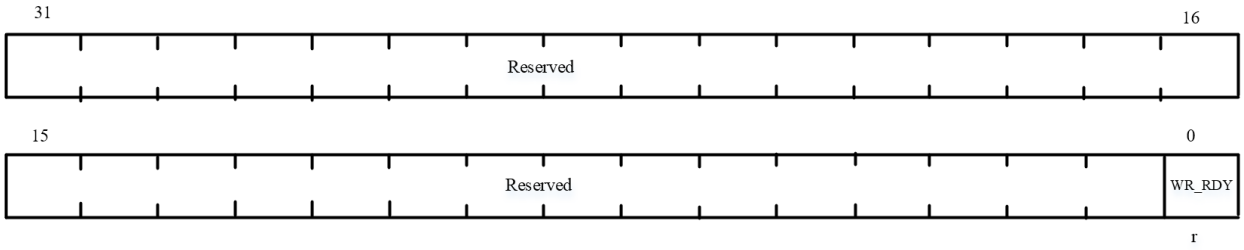
位域	名称	描述
31:7	Reserved	保留
6:5	MODE	随机数模式 00：关闭随机数 01：随机源 10：伪随机数 11：真随机数
4:2	Reserved	保留
1	RING1_EN	反相器环 1 使能 0：关闭 1：使能
0	RING0_EN	反相器环 0 使能 0：关闭 1：使能

注：写 RNG_CTRL 之前必须先确保 RNG_WR_RDY[0]为 1，切换 MODE 之前必须先关闭随机数（配置 MODE 为 2'b00）。

RNG_CTRL 寄存器复位受 RCC_AHBPRST 寄存器的 SACRST 位控制。

15.4.3 随机数写操作就绪寄存器（RNG_WR_RDY）

偏移地址：0x04
复位值：0x0000 0001

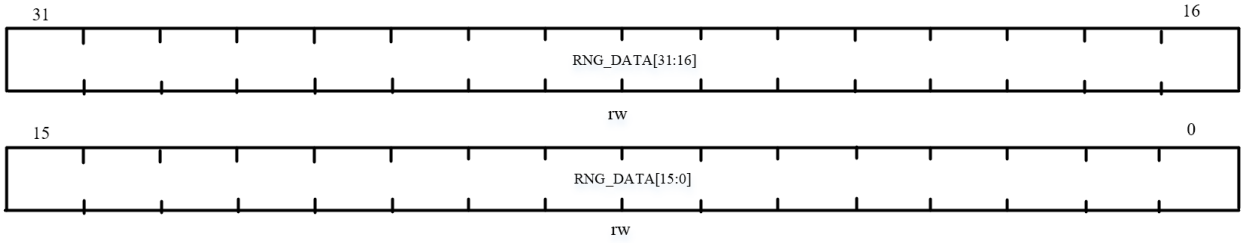


位域	名称	描述
31:1	Reserved	保留
0	WR_RDY	WR_RDY 为 1，表示可以进行写 RNG_CTRL 操作。

注: RNG_WR_RDY 寄存器复位受 RCC_AHBPRST 寄存器的 SACRST 位控制。

15.4.4 随机数数据寄存器（RNG_DATA）

偏移地址：0x08
复位值：0x0000 0000

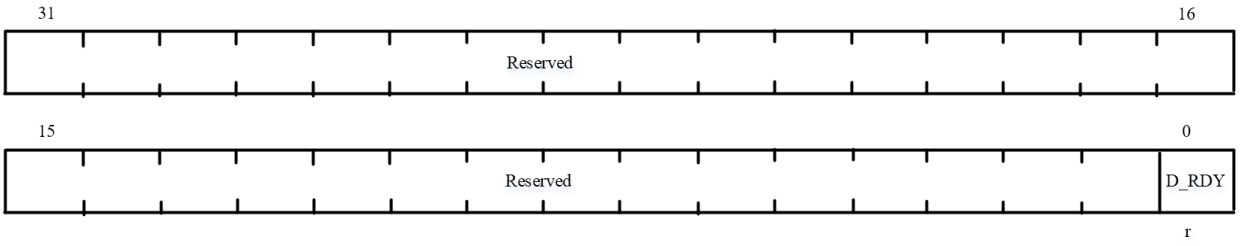


位域	名称	描述
31:0	RNG_DATA	32 位随机数

注：写 RNG_DATA 寄存器受 RCC_APB1PRST 寄存器的 RNGCRST 位控制。

15.4.5 随机数就绪寄存器（RNG_D_RDY）

偏移地址：0x0C
复位值：0x0000 0000



位域	名称	描述

31:1	Reserved	保留
0	D_RDY	D_RDY 为 1，表示已准备好随机数，可通过读 RNG_DATA 获取随机数。

注：写 RNG_D_RDY 寄存器受 RCC_APB1PRST 寄存器的 RNGCRST 位控制。

16 密码算法硬件加速引擎(SAC)

内嵌算法硬件加速引擎，支持多种国际算法及国家密码对称密码算法和杂凑密码算法加速，相较于纯软件算法而言能极大的提高加解密速度。

- 硬件支持的算法如下:
  - 支持 DES 对称算法
    - 支持 DES 和 3DES 加解密运算
    - TDES 支持 2KEY 和 3KEY 模式
    - 支持 CBC 和 ECB 模式
  - 支持 AES 对称算法
    - 支持 128bit/192bit/ 256bit 密钥长度
    - 支持 CBC、ECB、CTR 模式
  - 支持 SHA 杂凑算法
    - 支持 SHA1/SHA244/SHA256
  - 支持 MD5 摘要算法
  - 支持对称式国密 SM4 算法以及 SM3 杂凑算法

**注：密码算法性能及使用请联系中移物联销售人员*

17 循环冗余校验（CRC）

17.1 CRC 简介

本模块集成了 CRC32 和 CRC16 功能，循环冗余校验（CRC）计算单元是根据固定的生成多项式得到任一 CRC 计算结果。在其他的应用中， CRC 技术主要应用于核实数据传输或者数据存储的正确性和完整性。标准 EN/IEC 60335-1 即提供了一种核实闪存存储器完整性的方法。CRC 计算单元可以在程序运行时计算出软件的标识，之后与在连接时生成的参考标识比较，然后存放在指定的存储器空间。

17.2 CRC 主要特性

17.2.1 CRC32

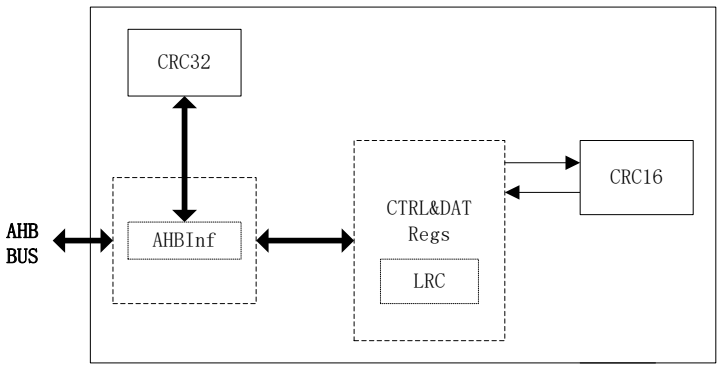
- CRC32($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$)
- 待校验数据 32 位，输出校验码 32 位
- CRC 计算时间：4 个 AHB 时钟周期（HCLK）
- 通用 8 位寄存器（可用于存放临时数据）

17.2.2 CRC16

- CRC16($X^{16} + X^{15} + X^{2} + 1$)
- 待校验数据 8 位，输出校验码 16 位
- CRC 计算时间：1 个 AHB 时钟周期（HCLK）
- 校验初始值可配，待校验数据大小端可配
- 支持 8bitLRC 校验值生成

下图为 CRC 计算单元框图

图 17-1 CRC 计算单元框图



17.3 CRC 功能描述

17.3.1 CRC32

CRC 计算单元含有 1 个 32 位数据寄存器:

- 对该寄存器进行写操作时，作为输入寄存器，可以输入要进行 CRC 计算的新数据。
- 对该寄存器进行读操作时，返回上一次 CRC 计算的结果。

每一次写入数据寄存器，其计算结果是前一次 CRC 计算结果和新计算结果的组合（对整个 32 位字进行 CRC 计算，而不是逐字节地计算）。

在 CRC 计算期间会暂停 CPU 的写操作，因此可以对寄存器 CRC_CRC32DAT 进行背靠背写入或者连续地写-读操作。

可以通过设置寄存器 CRC_CRC32CTRL 的 RESET 位来重置寄存器 CRC_CRC32DAT 为 0xFFFF FFFF。该操作不影响寄存器 CRC_CRC32IDAT 内的数据。

17.3.2 CRC16

若需要配置校验数据的大小端，可以配置 CRC_CRC16CTRL _ENDHL。

若需要清除上次 CRC 运算的结果，可以配置 CRC_CRC16CTRL _CLR 为 1 或 CRC_CRC16D 为 0。

若对 CRC 校验的初始值有要求，则先配置 CRC_CRC16D 寄存器，否则初始值为上次运行的结果。

LRC 计算过程跟 CRC 一样，二者计算同时进行。可以根据需要选择读取 CRC 或 LRC。若对初始值有要求，需先配置 LRC 寄存器。

17.4 CRC 寄存器

17.4.1 CRC 寄存器映像

下表列出了 CRC 的寄存器映像和复位值

表 17-1 CRC 计算单元寄存器映像和复位值

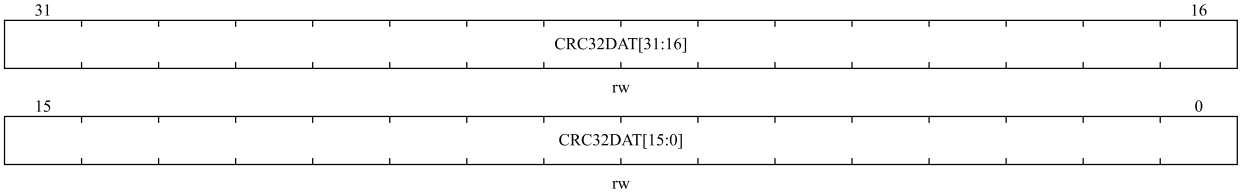
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	CRC32DAT	CRC32DAT[31:0]																															
	Reset Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
004h	CRC32IDAT	Reserved																								CRC32IDAT[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0
008h	CRC32CTRL	Reserved																															RESET
	Reset Value																																0
00Ch	CRC16CTRL	Reserved																													CLR	ENDHL	Reserved
	Reset Value																														0	0	0
010h	CRC16DAT	Reserved																								CRC16DAT[7:0]							

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset Value																									0	0	0	0	0	0	0	0
014h	CRC16D	Reserved																CRC16D[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	LRC	Reserved																								LRCDAT[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0

17.4.2 CRC32 数据寄存器（CRC_CRC32DAT）

地址偏移：0x00

复位值：0xFFFF FFFF

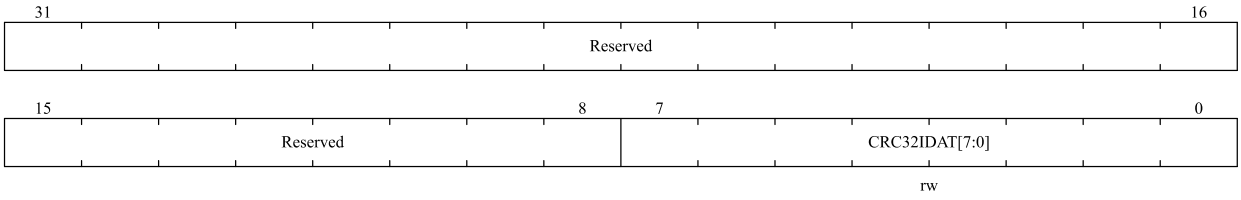


位域	名称	描述
31:0	CRC32DAT[31:0]	数据寄存器位 写入 CRC 计算器的新数据时，作为输入寄存器 读取时返回 CRC 计算的结果

17.4.3 CRC32 独立数据寄存器（CRC_CRC32IDAT）

地址偏移：0x04

复位值：0x0000 0000



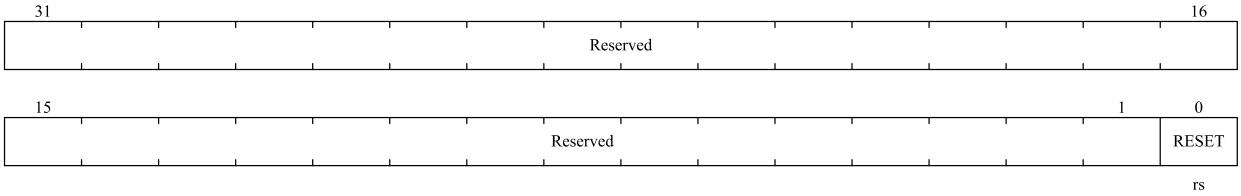
位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7:0	CRC32IDAT[7:0]	独立 8 位数据寄存器位 可用于临时存放 1 字节的数据。 寄存器 CRC_CRC32CTRL 的 RESET 位产生的 CRC 复位对本寄存器没有影响

注：此寄存器不参与 CRC 计算，可以存放任何数据。

17.4.4 CRC32 控制寄存器（CRC_CRC32CTRL）

地址偏移：0x08

复位值：0x0000 0000

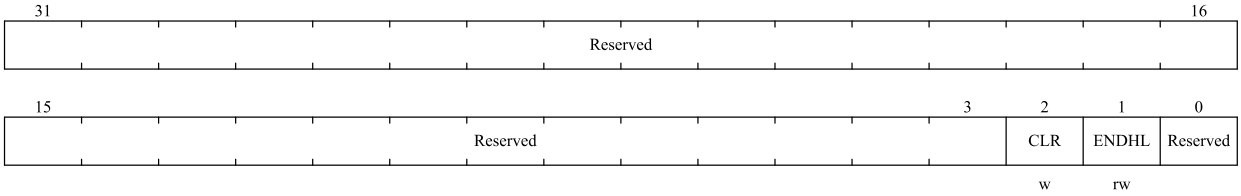


位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	RESET	复位 CRC 计算单元，设置数据寄存器为 0xFFFF FFFF。 只能对该位写'1'，它由硬件自动清'0'。

17.4.5 CRC16 控制寄存器（CRC_CRC16CTRL）

地址偏移：0x0C

复位值：0x0000 0000



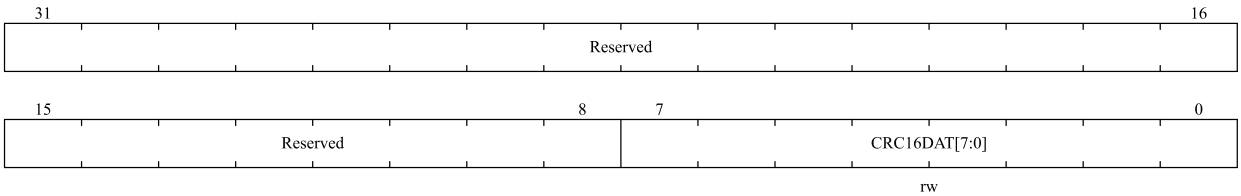
位域	名称	描述
31:3	Reserved	保留，硬件强制为 0。
2	CLR	清除 CRC16 校验值 0：不清除 1：清除，采用默认值 0x0000 该位软件置 1，仅维持 1 个周期，由硬件自动清零（软件读恒为 0）
1	ENDHL	选择待校验数据大小端 0：从高位到低位 1：从低位到高位 该 bit 仅针对待校验数据
0	Reserved	保留，硬件强制为 0。

注：支持 8、16、32 位操作

17.4.6 CRC16 待校验数据寄存器（CRC_CRC16DAT）

地址偏移：0x10

复位值：0x0000 0000



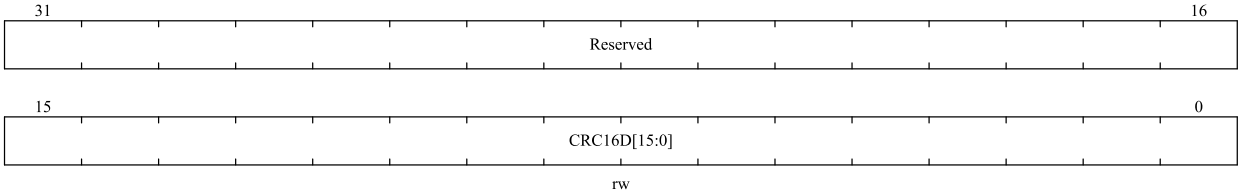
位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。

位域	名称	描述
7:0	CRC16DAT[7:0]	写入的待校验数据

注：支持 8、16、32 位操作

17.4.7 CRC 循环冗余校验码寄存器（CRC_CRC16D）

地址偏移：0x14
复位值：0x0000 0000

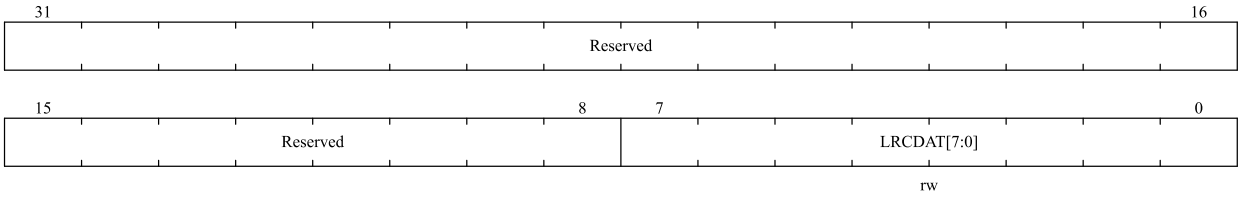


位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	CRC16D[15:0]	循环冗余校验码 16 位值 软件每次写 CRC16DAT 寄存器时，CRC16 计算的 16 位结果写入该寄存器

注：支持 8、16、32 位操作（8 位操作时必须连续配置 2 次，以确保写入 16 位的初值，读出 16 位正确结果）

17.4.8 LRC 校验值寄存器（CRC_LRC）

地址偏移：0x18
复位值：0x0000 0000



位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7:0	LRCDAT[7:0]	LRC 校验值寄存器： 该寄存器可直接读写，在使用之前，软件可先配置初始值； 之后每次写入 CRC16DAT 寄存器的数值都会跟 LRC 寄存器进行“异或”运算，结果回放该寄存器。软件读取结果，下次运算前需软件清零。

18 独立看门狗（IWDG）

18.1 IWDG 简介

CM32M4xxR 内置独立看门狗（IWDG）和窗口看门狗（WWDG）两个定时器用以解决软件错误引起的故障问题。看门狗定时器使用灵活，提高了系统的安全性和时间控制的精确性。

独立看门狗（IWDG）由专用时钟源（LSI）驱动，其工作状态不受主时钟故障影响。因此，IWDG 适用于独立工作在主程序之外且对时间精度要求较低的场合。

当电源控制寄存器（PWR_CTRL2）的 IWDGRSTEN 位置‘1’，IWDG 计数器达到 0 时，会产生系统复位（若该位置‘0’，IWDG 会计数但不产生复位）。当 IWDGWPEN 位置‘1’，IWDG 计数器达到 0 时，能够唤醒低功耗（若该位置‘0’，IWDG 会计数、复位但只能唤醒 Sleep/Stop0，无法唤醒 Stop2/Standby）。

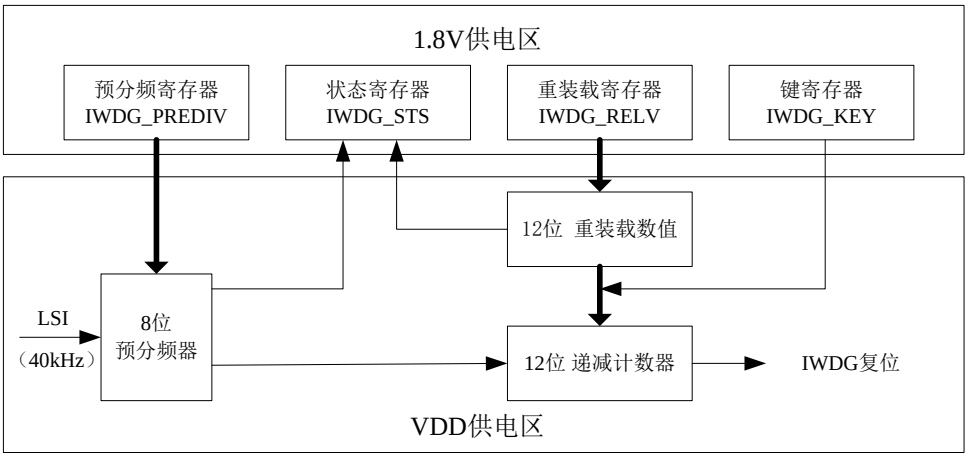
注：本章基于系统默认值 IWDGRSTEN=1，IWDGWPEN=1 讨论。

18.2 IWDG 主要性能

- 独立运行的 12 位递减计数器
- RC 振荡器提供独立时钟源，待机和深度睡眠模式下仍能正常工作
- 复位和低功耗唤醒可配
- 使能看门狗后，当递减计数器达到 0x000 时产生系统复位

18.3 IWDG 功能描述

图 18-1 为独立看门狗模块的功能框图。



注：看门狗功能处于 VDD 供电区，在待机和深度睡眠模式下仍能正常工作。

配置寄存器（IWDG_KEY）为 0xCCCC 即可开启 IWDG，计数器开始递减计数。当计数达到 0x000 时，产生系统复位(IWDG_RESET)。为阻止看门狗复位，可在任意时刻配置寄存器 IWDG_KEY 为 0xAAAA，将 IWDG_RELV 的数据重新加载到计数器中。

如果在选择字节中启用了“硬件看门狗定时器”功能，那么在系统上电复位后，看门狗会自动开始运行。在计数器计数结束前，软件重装载计数器，即可避免系统复位。

18.3.1 寄存器访问保护

IWDG_PREDIV 和 IWDG_RELV 寄存器具有写保护功能。在修改这两个寄存器数据之前，必须先配置 IWDG_KEY 寄存器为 0x5555。配置成其他任何数据，都将再次启动寄存器写保护。

重装载操作（IWDG_KEY 配置 0xAAAA）也会启动写保护功能。

当预分频值和递减计数器正在被更新时，状态寄存器 IWDG_STS 相应状态位置 1。

18.3.2 调试模式

进入调试模式时（N308 核心停止），当调试模块中的 DBG_IWDG_STOP=1，IWDG 停止工作；DBG_IWDG_STOP=0，IWDG 仍可继续工作。详见有关调试模块的章节 29.3.2。

表 18-1 看门狗超时时间（40KHz 的输入时钟（LSI））(1)

预分频系数	PREDIV [2:0]	最短时间（ms） REL [11:0]=0x000	最长时间（ms） REL [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

注：上表是 LSI=40KHz 的参考时间。通过对 LSI 进行校准可获得更加精确的看门狗定时器超时时间。详细介绍，见 5.2.5 节。

18.4 IWDG 寄存器描述

18.4.1 IWDG 寄存器映像

表 18-2 IWDG 寄存器映像和复位值

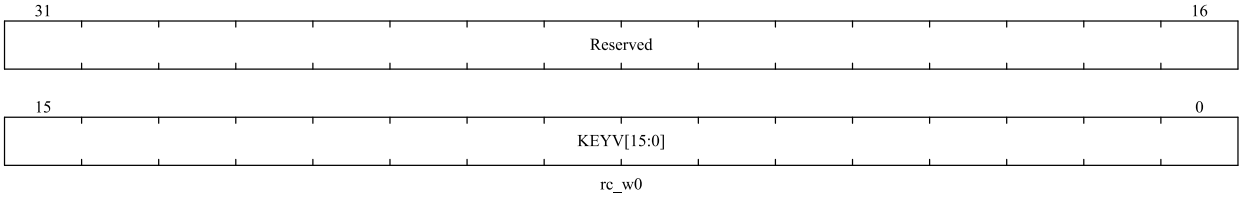
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	IWDG_KEY	Reserved																KEYV[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	IWDG_PREDIV	Reserved																													PD[2:0]		
	Reset Value																														0	0	0
008h	IWDG_RELV	Reserved																				REL[11:0]											
	Reset Value																					1	1	1	1	1	1	1	1	1	1	1	1
00Ch	IWDG_STS	Reserved																														CRVU	PVU

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

18.4.2 键寄存器（IWDG_KEY）

地址偏移：0x00

复位值：0x0000 0000 （在待机模式复位）

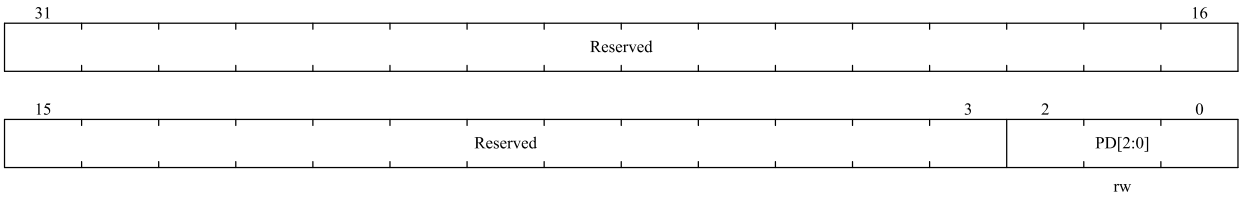


位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	KEYV[15:0]	只写寄存器，读出值为 0x0000 软件必须以一定的间隔写入 0xAAAA，否则，当计数器为 0 时，看门狗会产生复位。 写入 0x5555 表示允许访问 IWDG_PREDIV 和 IWDG_RELV 寄存器。（见 18.3.1 节） 写入 0xCCCC，启动看门狗工作（若选择了硬件看门狗则不受此命令字限制）。

18.4.3 预分频寄存器（IWDG_PREDIV）

地址偏移：0x04

复位值：0x0000 0000

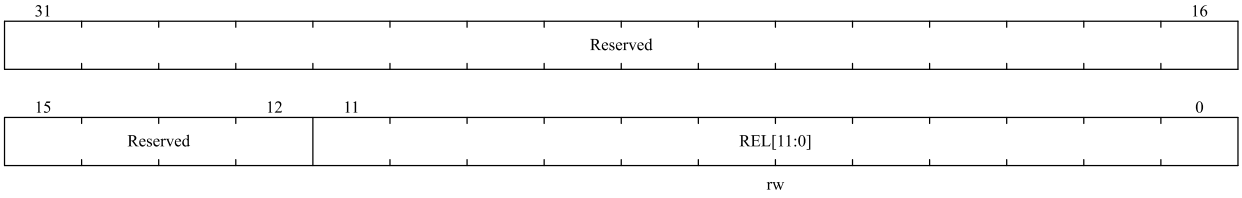


位域	名称	描述
31:3	Reserved	保留，硬件强制为 0。

位域	名称	描述
2:0	PD[2:0]	预分频因子 具有写保护设置，参见节 18.3.1 节。通过设置这些位来选择计数器时钟的预分频因子。要改变预分频因子，IWDG_STS 寄存器的 PVU 位必须为 0。 000: 预分频因子=4 001: 预分频因子=8 010: 预分频因子=16 011: 预分频因子=32 100: 预分频因子=64 101: 预分频因子=128 110: 预分频因子=256 111: 预分频因子=256 *注：对此寄存器进行读操作，将从 VDD 电压域返回预分频值。如果写操作正在进行，则读回的值可能是无效的。因此，只有当 IWDG_STS 寄存器的 PVU 位为 0 时，读出的值才有效。*

18.4.4 重装载寄存器（IWDG_RELV）

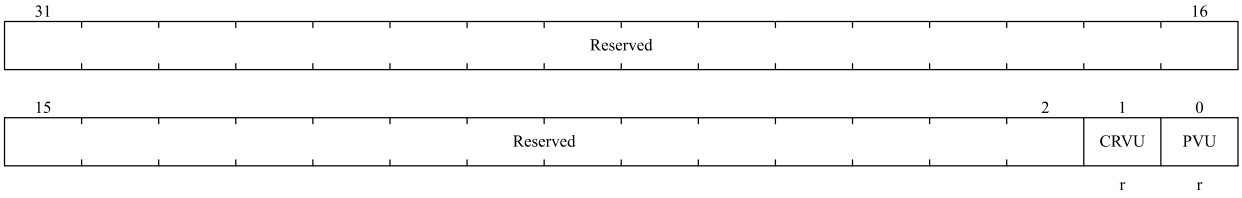
地址偏移：0x08
复位值：0x0000 0FFF（待机模式时复位）



位域	名称	描述
31:12	Reserved	保留，硬件强制为 0。
11:0	REL[11:0]	看门狗计数器重装载值 具有写保护功能 用于定义看门狗计数器的重装载值，每当向 IWDG_KEY 寄存器写入 0xAAAA 时，重装载值会被传送到计数器中。随后计数器从这个值开始递减计数。看门狗超时周期可通过此重装载值和时钟预分频值来计算，参照表 18-1。 只有当 IWDG_STS 寄存器中的 CRVU 位为 0 时，才能对此寄存器进行修改。 *注：对此寄存器进行读操作，将从 VDD 电压域返回预分频值。如果写操作正在进行，则读回的值可能是无效的。因此，只有当 IWDG_STS 寄存器的 CRVU 位为 0 时，读出的值才有效。*

18.4.5 状态寄存器（IWDG_STS）

地址偏移：0x0C
复位值：0x0000 0000 （待机模式时不复位）



位域	名称	描述
31:2	Reserved	保留，硬件强制为 0。
1	CRVU	看门狗计数器重装载值更新 硬件置'1'用来指示重装载值的更新正在进行中。当在 VDD 域中的重装载更新结束后，此位由硬件清'0'（最多需 5 个 40KHz 的 RC 周期）。重装载值只有在 CRVU 位被清'0'后才可更新。
0	PVU	看门狗预分频值更新 硬件置'1'用来指示预分频值的更新正在进行中。当在 VDD 域中的预分频值更新结束后，此位由硬件清'0'（最多需 5 个 40KHz 的 RC 周期）。预分频值只有在 PVU 位被清'0'后才可更新。

注： 如果在应用程序中使用了多个重装载值或预分频值，则必须在 CRVU 位被清除后才能重新改变预装载值，在 PVU 位被清除后才能重新改变预分频值。然而，在预分频和/或重装值更新后，不必等待 CRVU 或 PVU 复位，可继续执行下面的代码。(即是在低功耗模式下，此写操作仍会被继续执行完成。)

19 窗口看门狗（WWDG）

19.1 WWDG 简介

窗口看门狗（WWDG）的时钟由 APB1 时钟分频而来，通过时间窗口的配置来检测程序操作是否异常。因此，WWDG 适用于精确计时，通常用来监测由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。WWDG 的递减计数器在达到窗口寄存器数值之前或在 T6 位变成 0 后被刷新，会产生系统复位。

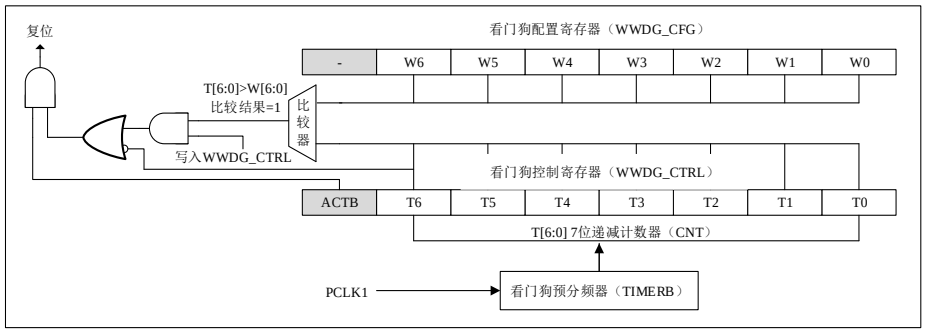
19.2 WWDG 主要特性

- 7 位独立运行递减计数器可编程
- WWDG 使能后，在以下条件产生复位
  - 递减计数器的值小于 0x40
  - 递减计数器的值大于窗口寄存器的值时被重新装载
- 提前唤醒中断：如果启动了看门狗并且使能中断，当计数值达到 0x40 时产生唤醒中断（EWINT）。

19.3 WWDG 功能描述

如果使能看门狗（WWDG_CTRL 寄存器中的 ACTB 位被置‘1’），当 7 位（T[6:0]）递减计数器达到 0x3F（T6 位清零），或软件在计数器值大于窗口寄存器的数值时重新装载计数器，都将产生系统复位。为避免系统复位，软件在正常运行过程中必须定期地在窗口内刷新计数器值。

图 19-1 看门狗框图



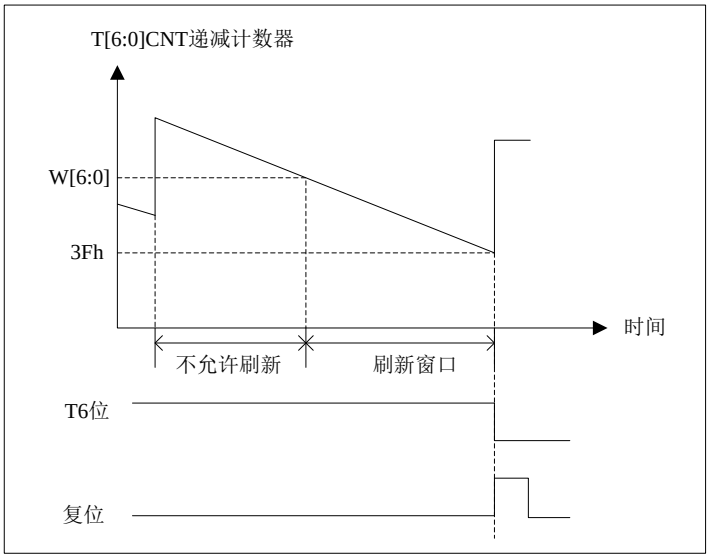
WWDG_CTRL 寄存器的 ACTB 位置‘1’使能看门狗，此后，除非发生复位，WWDG 将一直处于开启状态。7 位递减计数器处于独立运行状态，无论 WWDG 使能与否，计数器均保持递减计数。因此，使能看门狗前 T6 位始终置‘1’，避免立即产生复位。T[5:0]决定了两次重装载之间的最大时间间隔。时钟 APB1 和 WWDG_CFG 寄存器 TIMERB[1:0]设定的预分频值决定了计数器的递减速度。W[6:0]设置窗口的上限值。

当递减计数器在达到窗口寄存器数值之前或在 T6 位变成 0 后被刷新，会产生系统复位。图 19-2 描述了窗口寄存器的工作过程。

WWDG_CFG 寄存器的 EWINT 位置‘1’使能提前唤醒中断。当递减计数器到达 0x40 时产生中断，可以在相应的中断服务程序（ISR）中分析软件故障原因或保存重要数据，同时重新装载计数器以防止 WWDG 复位。WWDG_STS 寄存器的 EWINTF 位写’0’可以清除该中断。

19.4 如何编写看门狗超时程序

图 19-2 WWDG 时序图



计算窗口看门狗超时时间的公式如下：

$$T_{WWDG} = T_{PCLK1} \times 4096 \times 2^{TIMERB} \times (T[5:0] + 1); \quad (ms)$$

其中：

T_{WWDG}：WWDG 超时时间

T_{PCLK1}：APB1 以 ms 为单位的时钟间隔

在 PCLK1 = 36MHz 时的最小-最大超时值

TIMERB	最小超时值	最大超时值
0	113μs	7.28ms
1	227μs	14.56ms
2	455μs	29.12ms
3	910μs	58.25ms

19.5 调试模式

进入调试模式时（N308 核心停止），当调试模块中的 DBG_WWDG_STOP=1，WWDG 停止工作；DBG_WWDG_STOP=0，WWDG 仍可继续工作。详见第 29.3.2 节。

19.6 寄存器描述

19.6.1 WWDG 寄存器映像

表 19-1 WWDG 寄存器映像和复位值

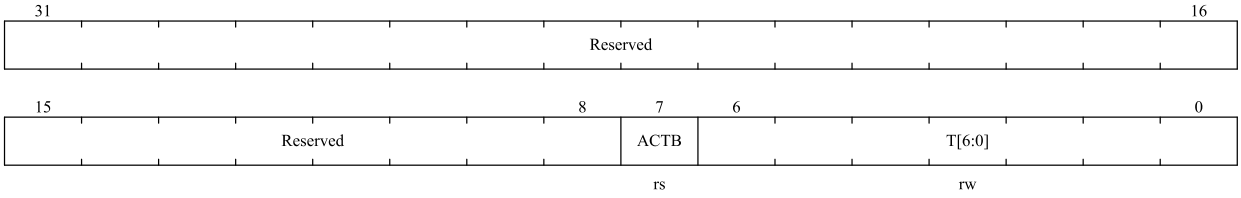
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	WWDG_CTRL	Reserved																								ACTB	T[6:0]						
	Reset Value																									0	1	1	1	1	1	1	1

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
004h	WWDG_CFG	Reserved																						EWINT	TIMERB[1:0]		W[6:0]						
	Reset Value																							0	0	0	1	1	1	1	1	1	1
008h	WWDG_STS	Reserved																															EWINTF
	Reset Value																																0

19.6.2 控制寄存器（WWDG_CTRL）

地址偏移量：0x00

复位值：0x7F

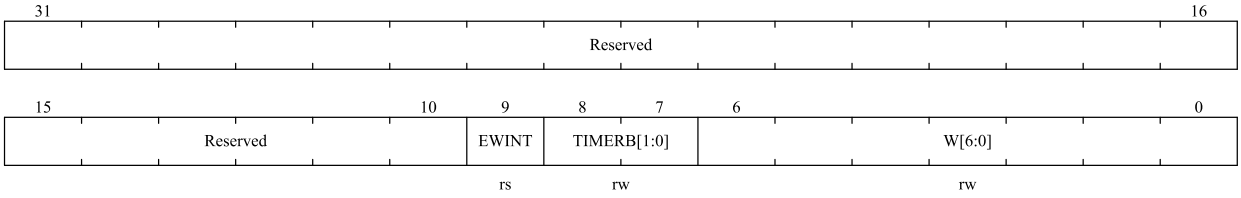


位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7	ACTB	激活位 由软件置'1'，但仅能由硬件在复位后清'0'。当 ACTB=1 时，看门狗可以产生复位。 0：禁止看门狗 1：启用看门狗
6:0	T[6:0]	7 位计数器用来存储看门狗的计数器值 每（4096x2^{TIMERB}）个 PCLK1 周期减 1。当计数器值从 40h 变为 3Fh 时（T6 变成 0），产生看门狗复位。

19.6.3 配置寄存器（WWDG_CFG）

地址偏移量：0x04

复位值：0x7F



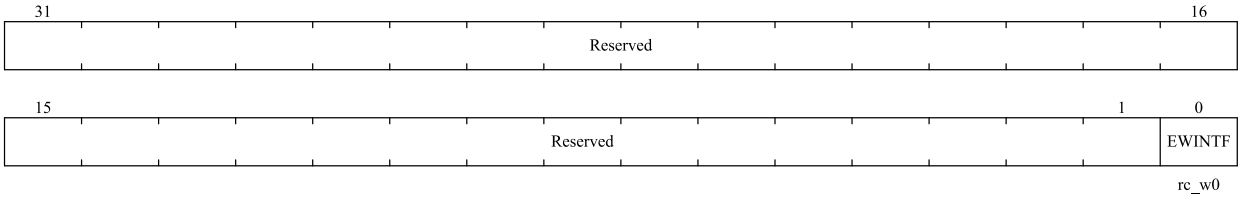
位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
9	EWINT	提前唤醒中断 若置'1'，则当计数器值达到 40h，即产生中断。 此中断只能由硬件在复位后清除。
8:7	TIMERB[1:0]	预分频器的时基,设置如下： 00: CK 计时器时钟（PCLK1 除以 4096）除以 1 01: CK 计时器时钟（PCLK1 除以 4096）除以 2 10: CK 计时器时钟（PCLK1 除以 4096）除以 4

位域	名称	描述
		11: CK 计时器时钟（PCLK1 除以 4096）除以 8
6:0	W[6:0]	7 位窗口值 用来与递减计数器进行比较用的窗口值。

19.6.4 状态寄存器（WWDG_STS）

地址偏移量：0x08

复位值：0x00



位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	EWINTF	提前唤醒中断标志 当计数器值达到 40h 时，由硬件置'1'。必须通过软件写'0'清除。写'1'无效。若中断未被使能，此位也会被置'1'。

20 扩展存储控制器接口（XFMC）

20.1 XFMC 功能描述

扩展存储器接口（XFMC）用来访问各种片外存储器，可根据应用需要，方便地进行不同类型大容量静态存储器的扩展，能够在不增加外部接口的情况下同时扩展多种不同类型的静态存储器。所有的外部存储器共享 XFMC 控制器输出的地址、数据和控制信号，XFMC 通过一个唯一的片选信号来区分不同的外部设备。 XFMC 在任一时刻只能访问一个外部设备。它的主要作用是：

- 将内核发送到 AHB 总线上的访问信号，转换为符合外部存储设备通信规则的信号，送到外部存储设备的相应管脚。
- 进行信号宽度和时序的调整，满足访问外部设备的时序要求

XFMC 具有下列主要功能：

- 支持外扩以下器件：
  - SRAM
  - PSRAM
  - ROM
  - NOR Flash
  - NAND Flash（SLC）
  - LCD（8080/6888）
- 支持两个 NAND 闪存块，硬件 1bit-ECC 可检测多达 8K 字节数据
- 支持对同步器件的成组（Burst）访问模式，如 NOR 闪存和 PSRAM
- 8 或 16 位数据总线
- 每一个存储器块都有独立的片选控制
- 通过时序编程可以支持各种不同的器件：
  - 多达 15 个周期的等待周期可编程
  - 多达 15 个周期的总线恢复周期可编程
  - 多达 15 个周期的输出使能和写使能延迟可编程
  - 独立的读写时序和协议，可以支持宽范围的存储器和时序
- 根据外部存储器的数据宽度，将 32 位的 AHB 访问请求，自动转换为连续的 16 位或 8 位，实现和外部 16 位或 8 位存储器件的通信将 32 位的 AHB 访问请求，转换到连续的 16 位或 8 位的，对外部 16 位或 8 位器件的访问
- 具有深度是 16，宽度为 32bit 的写入 FIFO，允许在写入较慢存储器的时候释放 AHB 总线进行其它操作。注意，在开始一次新的 XFMC 操作前，FIFO 要先被清空。
- PSRAM 和 SRAM 器件支持写使能和字节选择输出

- 支持两个 NAND 闪存块，硬件 1bit-ECC 可检测多达 8K 字节数据
- 支持对同步器件的成组（Burst）访问模式，如 NOR 闪存和 PSRAM

通常在系统复位或上电时，应该设置好所有配置外部存储器类型和特性的 XFMC 寄存器，以保证上电或者复位后系统能够正常运行，也可以在任何时候改变这些配置。

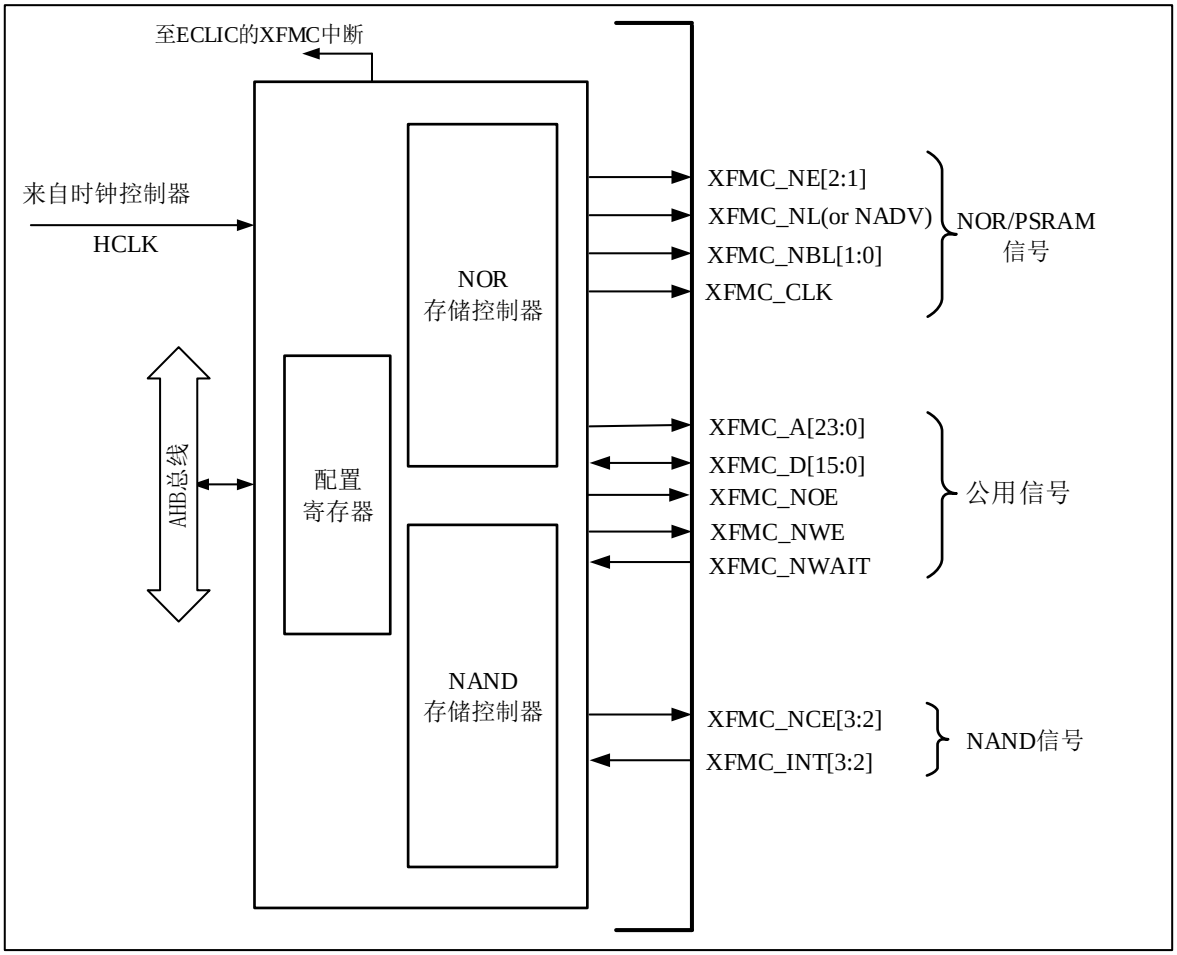
20.2 框图

XFMC 由五个模块组成：

- AHB 接口
- XFMC 配置寄存器
- NOR 闪存和 PSRAM 控制器
- NAND 闪存控制器
- 外部设备接口

XFMC 框图如下：

图 20-1 XFMC 框图



20.3 AHB 接口

XFMC 的参考时钟为 AHB 时钟（HCLK）。AHB 接口为内部 CPU 和其它总线控制设备访问外部静态存储器提供了通道。XFMC 是 AHB 总线至外部设备协议的转换接口，32 位的 AHB 读写操作可以根据选择的外部存储器的数据通道是 16 还是 8 位被分割成连续的 16 或 8 位的操作。

XFMC 操作规则

在数据传输的过程中，请求 AHB 操作的数据宽度可以是 8 位、16 位或 32 位，而外部设备则是固定的数据宽度，所以 AHB 数据宽度和存储器数据宽度可能不相同，为了保证数据传输的一致性，XFMC 读写访问需要遵从以下规则:

- 当存储器数据宽度等于 AHB 访问的数据宽度，则无数据传输一致性的问题。
- 当存储器的数据宽度小于 AHB 访问的数据宽度，XFMC 则将 AHB 操作分割成几个连续的数据宽度的传输，以适应外部设备的数据宽度。
- 当存储器的数据宽度大于 AHB 访问的数据宽度: 如果外部存储设备具有字节选择功能, 如 SRAM、

ROM、PSRAM，则可通过它的字节通道 BL[1:0]来访问对应的字节。否则禁止写操作，只允许读操作。

XFMC 由一组寄存器进行配置。XFMC 寄存器章节详细描述了 NOR 闪存、PSRAM 控制器寄存器以及 NAND 闪存寄存器。

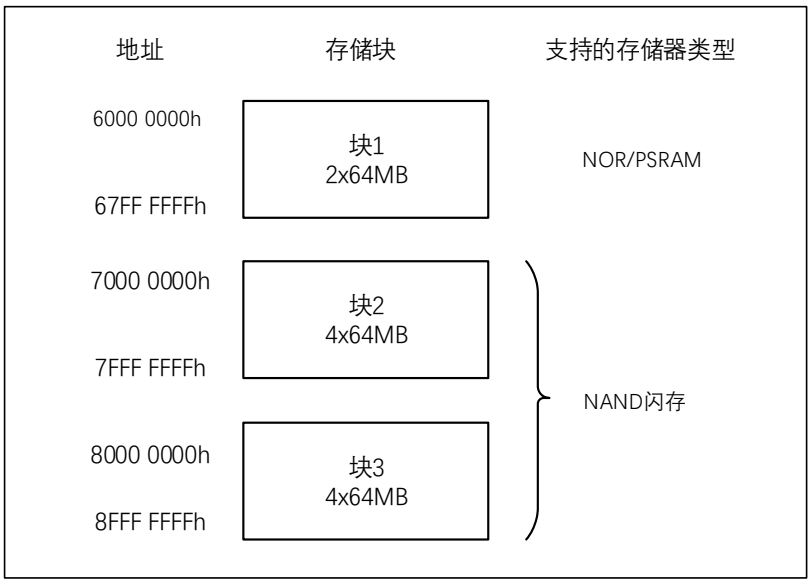
20.4 外部设备地址映像

XFMC 把外部存储器划分为固定大小的三个 Bank，其中 Bank1 又分为 2 个区域，每个区域占 64M 字节（有效区域 16M 字节），总共 128M 字节。Bank2 和 Bank3 大小分别为 256M，且都被分为 2 个 Section，分别是属性存储空间和通用存储空间，见下图。

- Bank1 用于访问 NOR 闪存或 SRAM/PSRAM 存储设备。这个存储区被划分为 2 个 NOR/PSRAM 区并有 2 个专用的片选，所以 BANK1 最多可以访问两个外部存储设备。
- Bank2 和 Bank3 用于访问 NAND 闪存设备，每个 Bank 可以连接一个 NAND 闪存。

每一个 Bank 或区域上的存储器类型都能进行独立的配置，由用户在相应配置寄存器中定义。

图 20-2 XFMC 存储块



20.4.1 NOR 和 PSRAM 地址映像

AHB 地址线 HADDR[27:26]用于两个区域的片选信号:

表 20-1 NOR/PSRAM 存储块选择

HADDR[27:26](1)	选择的存储块
00	存储块 1 NOR/PSRAM 1
01	存储块 1 NOR/PSRAM 2

（1）HADDR 是内部 AHB 地址线。

AHB 地址线 HADDR[23:0]包含外部存储器地址。 HADDR 是字节地址，而外部存储器访问有可能不是按字节访问的，所以可能会出现地址不一致的情况，XFMC 可以通过调整 HADDR 来适应外部存储

器的数据宽度。具体规则如下表:

表 20-2 外部存储器地址

数据宽度	连到存储器的地址线	最大访问存储器空间（位）
8 位	HADDR[23:0]与 XFMC_A[23:0]对应相连	16M 字节 x 8 = 128 M 位
16 位	HADDR[23:1]与 XFMC_A[22:0]对应相连，HADDR[0]未接	16M 字节/2 x 16 = 128 M 位

(1) 如果外部存储器的数据宽度是 8 位按字节对齐，HADDR[23:0]与 XFMC_A[23:0]相连，然后用 XFMC_A[23:0]去连接外部存储器的地址线

(2) 如果外部存储器的数据宽度是 16 位按半字节对齐，就需要将 HADDR 的字节地址转化为半字地址之后再连接外存储器 ，所以需要将 HADDR[23:1]与 XFMC_A[22:0]相连。XFMC_A[22:0]连接到外部存储地址线。

(3) 不论外部存储器的宽度是多少(16 位或 8 位)，XFMC_A[0]始终应该连到外部存储器的地址线 A[0]。

NOR 闪存和 PSRAM 的非对齐访问支持

通过设置相应寄存器，每个 NOR 闪存或 PSRAM 存储器块都可以配置成支持非对齐的数据访问。依据访问的方式是异步或同步，外部存储器满足以下标准:

- **异步模式：**在异步模式时，只要每次访问都有准确的地址，完全支持非对齐的数据访问。
- **同步模式：**同步模式时，XFMC 只发出一次地址信号，然后成组的数据通过时钟 CLK 按顺序进行传输。有些 NOR 存储器支持线性的非对齐成组访问，固定数目的数据字可以从连续的以 N（典型的 N 为 8 或 16）为模的地址读取。这种情况下，可以把存储器的非对齐访问模式设置为与 AHB 相同的模式。

注意：如果外部存储器的非对齐访问模式不能设置为与 AHB 相同的模式，应该配置 XFMC 寄存器的相应位禁止非对齐访问，并把非对齐的访问请求分开成两个连续的访问操作。

20.4.2 NAND 地址映像

Bank2 和 Bank3 可以用来访问 NAND Flash，每个 Bank 又被划分为属性访问空间和通用访问空间:

表 20-3 存储器映像和时序寄存器

起始地址	结束地址	XFMC 存储块	存储空间	时序寄存器
0x8800 0000	0x8BFF FFFF	块 3-NAND 闪存	属性	XFMC_ATTMEMTM3(0x8C)
0x8000 0000	0x83FF FFFF		通用	XFMC_CMEMTM3(0x88)
0x7800 0000	0x7BFF FFFF	块 2-NAND 闪存	属性	XFMC_ATTMEMTM2(0x6C)
0x7000 0000	0x73FF FFFF		通用	XFMC_CMEMTM2(0x68)

对于 NAND 闪存存储器，通用和属性空间低 256K 字节部分又被划分为 3 个区，见表 20-4:

表 20-4 NAND 存储块选择

区域名称	HADDR[17:16]	地址范围
地址区	1x	0x020000~0x03FFFF
命令区	01	0x010000~0x01FFFF

数据区	00	0x000000~0x00FFFF

AHB 利用 HADDR[17:16]来实现对以上三个区的选择:

- HADDR[17:16]=00（通用/属性空间的前 64K 字节区域），即选择数据区
- HADDR[17:16]=01（通用/属性空间的第 2 个 64K 字节区域），即选择命令区
- HADDR[17:16]=1x（通用/属性空间的第 2 个 128K 字节区域）即选择地址区

应用软件使用这 3 个区访问 NAND Flash，操作规则如下:

- **发送命令到 NAND Flash：**软件只需在命令区的任意一个地址写入要发送的命令即可。在命令传输过程中，XFMC 会使能命令锁存信号（CLE）， 即 XFMC_A[16]拉高。
- **指定操作 NAND Flash 的地址：**软件只需对地址区的任意一个地址写入地址即可。在地址传输过程中， XFMC 会使能地址锁存信号（ALE），即 XFMC_A[17]为高。因为一个 NAND 地址可以有 4 或 5 个字节（依实际的存储器容量而定），所以需要连续地执行对地址区的写才能输出完整的操作地址。

读写数据：软件只需对数据区的任意一个地址写入或读出数据即可。当 XFMC 在数据发送模式，软件需要在数据区写入数据， 当 XFMC 在数据接收模式，软件需要在数据区读取数据。 由于 NAND Flash 会自动累加其内部操作地址，所以在读写时不需要软件修改操作地址。

20.5 NOR 闪存和 PSRAM 控制器

XFMC 控制器 Bank0 可以产生适当的时序信号，驱动下述类型的存储器:

- 异步 SRAM 和 ROM
  - 8 位
  - 16 位
  - 32 位
- PSRAM（Cellular RAM）
  - 异步模式
  - 同步突发模式
- NOR 闪存
  - 异步模式或同步突发模式
  - 复用模式或非复用模式

XFMC 对 Bank0 的每个存储块输出一个唯一的片选信号 NE[2:1]，用于在 2 个存储块中进行片选，所有其它的（地址、数据和控制）信号则是共享的。在同步方式中，XFMC 向选中的外部设备提供时钟（CLK），该时钟的频率是 HCLK 时钟的分频。每个存储块的大小固定为 64M 字节（有效访问空间为 16M 字节）。

每个存储块都有专门的寄存器控制（见 20.7.2 节（NOR 闪存和 PSRAM 控制寄存器））。

在异步模式下，所有控制器输出信号在内部 AHB 总线时钟 HCLK 的上升沿改变。

在同步模式下，所有控制器输出数据在外部存储器时钟 XFMC_CLK 的下降沿改变。

可编程的存储器参数包括访问时序（见下表）、是否支持非对齐数据存取和等待周期管理（只针对突发模式下访问 PSRAM 和 NOR 闪存）。

表 20-5 可编程的 NOR/PSRAM 访问参数

参数	功能	访问方式	单位	最小	最大
地址建立时间	地址建立阶段的时间	异步	AHB 时钟周期（HCLK）	1	16
地址保持时间	地址保持阶段的时间	异步，复用 I/O	AHB 时钟周期（HCLK）	2	16
数据建立时间	数据建立阶段的时间	异步	AHB 时钟周期（HCLK）	2	256
总线恢复时间	总线恢复阶段的时间	异步或同步读	AHB 时钟周期（HCLK）	1	16
时钟分频因子	存储器访问的时钟周期（CLK）与 AHB 时钟周期的比例	同步	AHB 时钟周期（HCLK）	1	16
数据产生时间	突发模式下产生第一个数据所需的时钟数目	同步	存储器时钟周期（HCLK）	2	17

20.5.1 外部存储器接口信号

表 20-6、 表 20-7、 表 20-8 列出了与 NOR 闪存和 PSRAM 接口的典型信号。

注： 具有前缀“N”的信号表示低有效信号

20.5.1.1 NOR 闪存，非复用接口

表 20-6 非复用信号的 NOR 闪存接口

XFMC 信号名称	信号方向	功能
CLK	输出	时钟（同步突发模式使用）
A[23:0]	输出	地址总线
D[15:0]	输入/输出	双向数据总线
NE[x]	输出	片选，x = 1.2
NOE	输出	输出使能
NWE	输出	写使能
NWAIT	输入	NOR 闪存要求 XFMC 等待的信号

NOR 闪存存储器是按 16 位的字寻址，最大容量达 16M 字节（24 条地址线）。

20.5.1.2 NOR 闪存，复用接口

表 20-7 复用 NOR 闪存接口

XFMC 信号名称	信号方向	功能
CLK	输出	时钟（同步突发模式使用）
A[23:16]	输出	地址总线
AD[15:0]	输入/输出	16 位复用的，双向地址/数据总线
NE[x]	输出	片选，x=1.2
NOE	输出	输出使能
NWE	输出	写使能
NADV	输出	地址有效或锁存使能
NWAIT	输入	NOR 闪存要求 XFMC 等待的信号

NOR 闪存存储器是按 16 位的字寻址，最大容量达 16M 字节（24 条地址线）。

20.5.1.3 PSRAM

表 20-8 非复用信号的 PSRAM 接口

XFMC 信号名称	信号方向	功能
CLK	输出	时钟（同步突发模式使用）
A[23:0]	输出	地址总线
D[15:0]	输入/输出	双向数据总线
NE[x]	输出	片选，x = 1.2（PSRAM 称其为 NCE（Cellular RAM 即 CRAM））
NOE	输出	输出使能
NWE	输出	写使能
NADV	输出	地址有效
NWAIT	输入	PSRAM 要求 XFMC 等待的信号
NBL[1]	输出	高字节使能（存储器信号名称为：NUB）
NBL[0]	输出	低字节使能（存储器信号名称为：NLB）

PSRAM 存储器是按 16 位的字寻址，最大容量达 16M 字节（24 条地址线）。

20.5.2 支持的存储器及其操作

下表列出了支持的存储器、访问模式和操作方式， XFMC 不支持阴影部分的操作方式。

表 20-9 XFMC 支持的 NOR 闪存/PSRAM 存储器和操作方式

存储器	模式	读/写	AHB 数据宽度	存储器数据宽度	是否支持	注释
NOR 闪存（总线复用和非总线复用）	异步	读	8	16	支持	
	异步	写	8	16	不支持	
	异步	读	16	16	支持	
	异步	写	16	16	支持	
	异步	读	32	16	支持	分成 2 次 XFMC 访问
	异步	写	32	16	支持	分成 2 次 XFMC 访问
	异步页	读	-	16	不支持	不支持这种模式
	同步	读	8	16	不支持	
	同步	读	16	16	支持	
	同步	读	32	16	支持	
PSRAM（总线复用和非总线复用）	异步	读	8	16	支持	
	异步	写	8	16	支持	使用字节信号 NBL[1:0]
	异步	读	16	16	支持	
	异步	写	16	16	支持	
	异步	读	32	16	支持	分成 2 次 XFMC 访问
	异步	写	32	16	支持	分成 2 次 XFMC 访问
	异步页	读	-	16	不支持	不支持这种模式
	同步	读	8	16	不支持	
	同步	读	16	16	支持	

存储器	模式	读/写	AHB 数据宽度	存储器数据宽度	是否支持	注释
	同步	读	32	16	支持	
	同步	写	8	16	支持	使用字节信号 NBL[1:0]
	同步	写	16/32	16	支持	
SRAM 和 ROM	异步	读	8/16/32	8/16	支持	使用字节信号 NBL[1:0]
	异步	写	8/16/32	8/16	支持	使用字节信号 NBL[1:0]

20.5.3 时序规则

20.5.3.1 信号同步

- 所有的控制器输出信号在内部时钟（HCLK）的上升沿变
- 在同步写模式（PSRAM）下，输出的数据在存储器时钟（CLK）的下降沿变化。

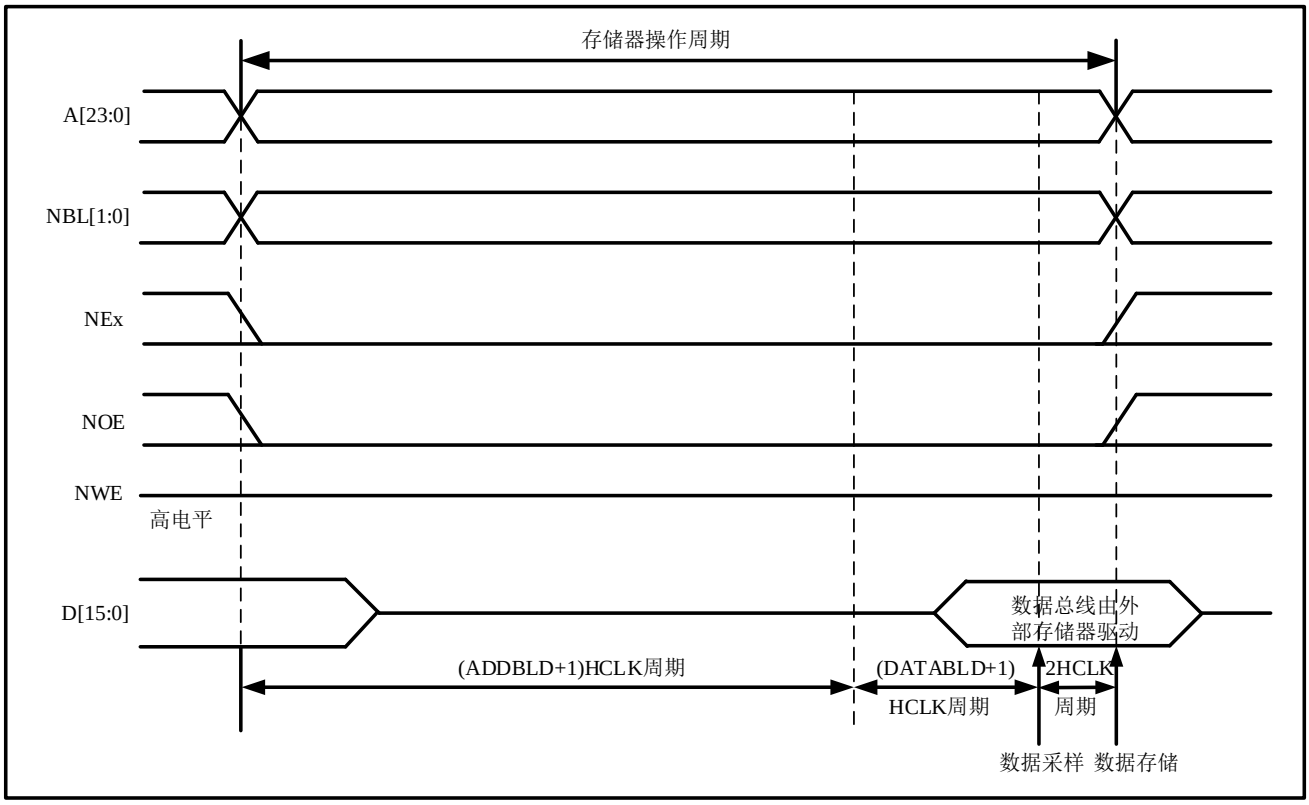
20.5.4 NOR 闪存和 PSRAM 控制器时序图

20.5.4.1 异步静态存储器（NOR 闪存和 PSRAM）

- 所有信号由内部时钟 HCLK 保持同步，但该时钟不会输出到存储器;
- 为了保证符合存储器的数据保持时序（片选失效至数据失效的间隔，通常最小为 0ns），XFMC 始终在片选信号 NE 失效前对数据线采样;
- 当设置了扩展模式，可以在读和写时混合使用模式 A、 B、 C 和 D（例如，允许以模式 A 进行读，而以模式 B 进行写）。

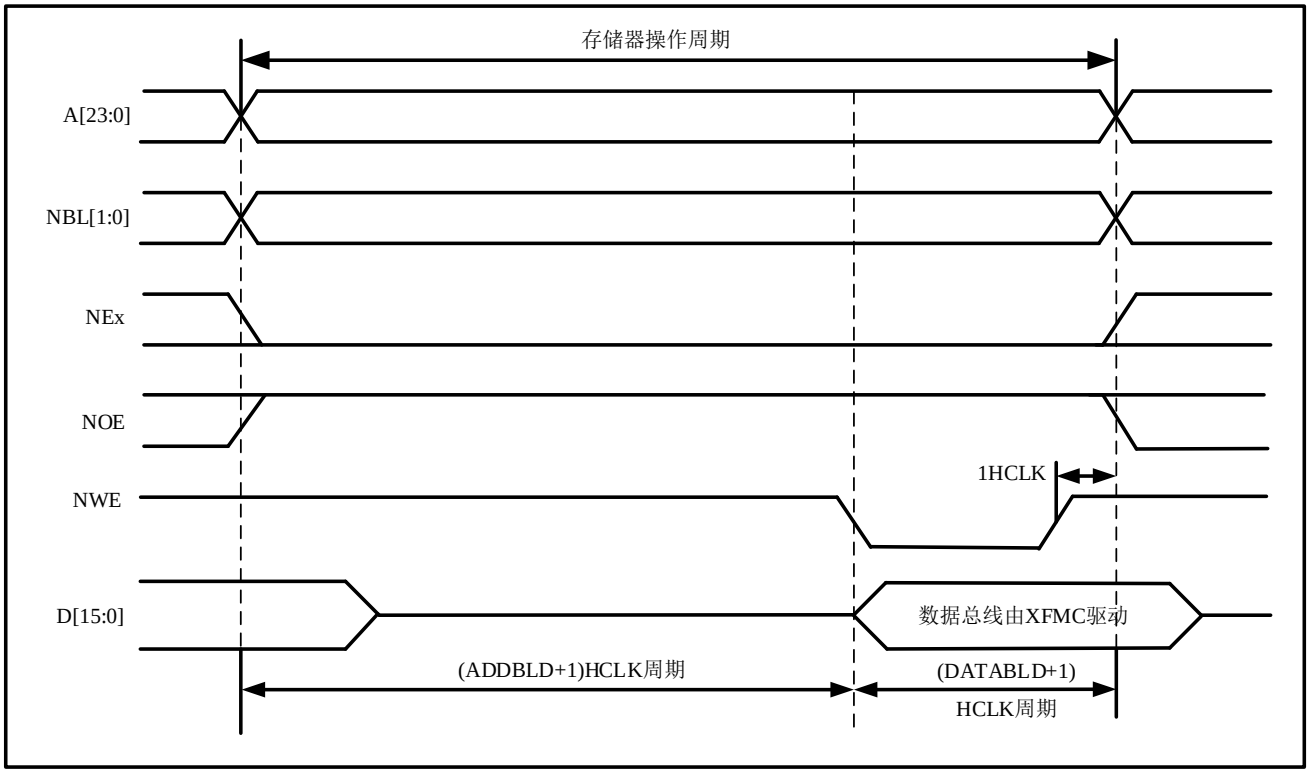
20.5.4.2 模式 1 — SRAM/CRAM

图 20-3 模式 1 读操作



注：ADDBLD：地址建立时间；DATABLD：数据建立时间。

图 20-4 模式 1 写操作



注：ADDBLD：地址建立时间；DATABLD：数据建立时间

在写操作的最后一个 HCLK 周期可以保证 NWE 上升沿后地址和数据的保持时间，因为存在这个 HCLK 周期， DATABLD 的数值必须大于 0（DATABLD > 0）。

表 20-10 XFMC_BANK1_CRx 位域（模式 1）

位编号	位名称	设置的数值
31-15		0x0000
14-10		0x0
9	WAITDIR	仅当位 15 为 ' 1 ' 时有意义
8	BURSTEN	0x0
7		-
6	ACCEN	-
5-4	MDBW	需要时设置
3-2	DTYP	需要时设置，不包含 10（NOR 闪存）
1	MUXEN	0x0
0	MBEN	0x1

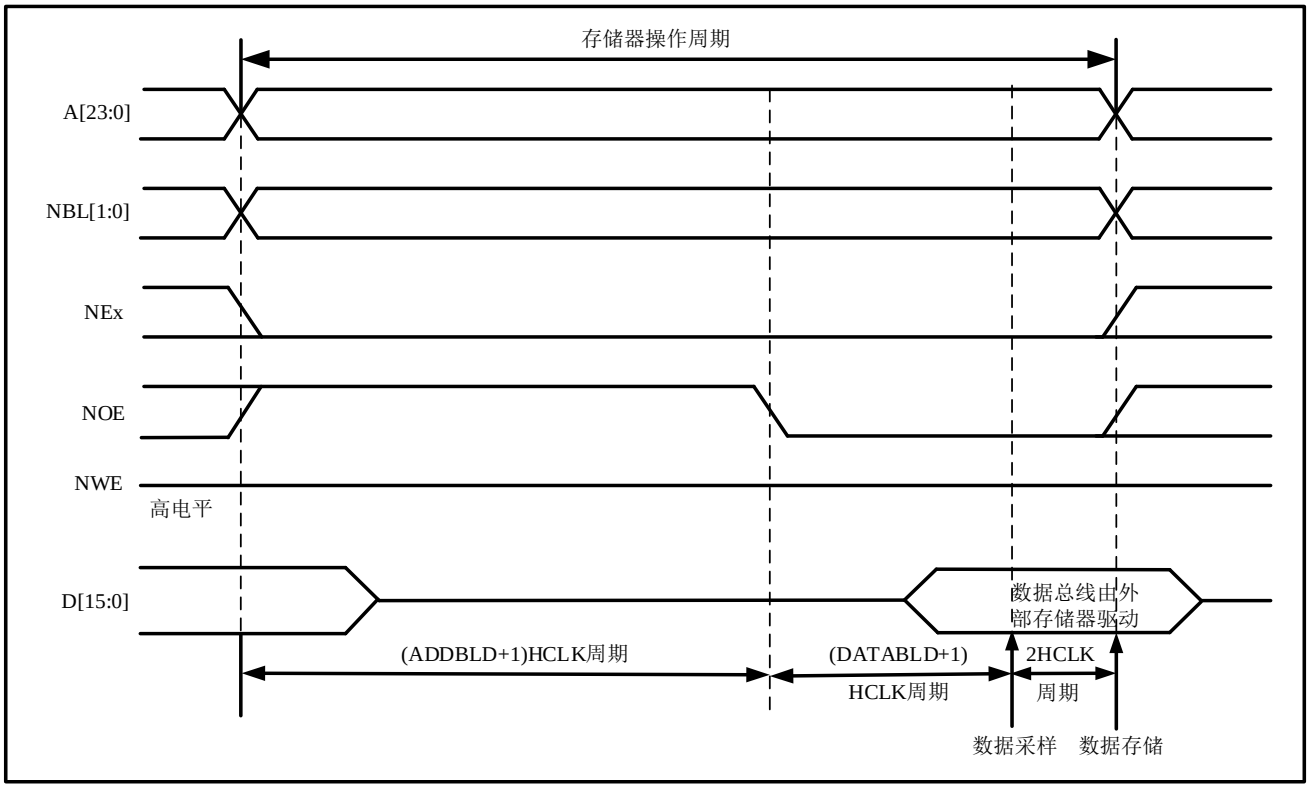
表 20-11 XFMC_BANK1_TRx 位域（模式 1）

位编号	位名称	设置的数值
31-16		0x0000
15-8	DATABLD	操作的第 2 个阶段的长度，写操作为（DATABLD+1 个 HCLK 周期）， 读操作为（DATABLD+3 个 HCLK 周期），这个域不能为 0，至少为 1

7-4		0x0
3-0	ADDBLD	操作的第 1 个阶段的长度（ADDBLD +1 个 HCLK 周期）

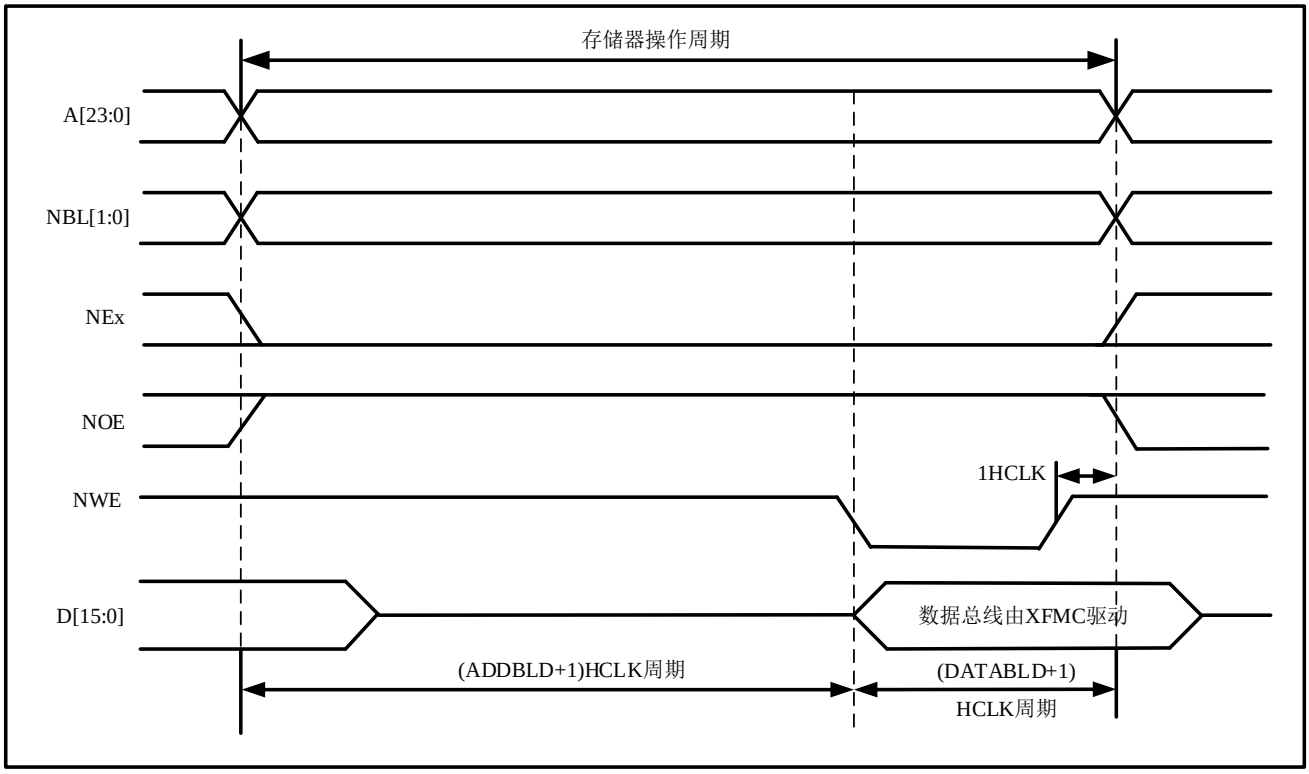
20.5.4.3 模式 A — SRAM/PSRAM（CRAM） OE 翻转

图 20-5 模式 A 读操作



注：ADDBLD：地址建立时间；DATABLD：数据建立时间

图 20-6 模式 A 写操作



注：ADDBLD：地址建立时间；DATABLD：数据建立时间

模式 A 和模式 1 的区别在于写时序，当两个模式的寄存器有相同的时序配置时， 模式 A 的写时序独立于读时序。

表 20-12 XFMC_BANK1_CRx 位域（模式 A）

位编号	位名称	设置的数值
31-16		0x0000
15		0x0
14	EXTEN	0x1
13-10		0x0
9	WAITDIR	仅当位 15 为 ' 1 ' 时有意义
8	BURSTEN	0x0
7		
6	ACCEN	
5-4	MDBW	需要时设置
3-2	DTYP	需要时设置，不包含 10（NOR 闪存）
1	MUXEN	0x0
0	MBEN	0x1

表 20-13 XFMC_BANK1_TRx 位域（模式 A）

位编号	位名称	设置的数值
31-30		0x0

位编号	位名称	设置的数值
29-28	ACCMODE	0x0
27-16		0x000
15-8	DATABLD	读操作的第 2 个阶段的长度（DATABLD +3 个 HCLK 周期），这个域不能为 0（至少为 1）
7-4		0x0
3-0	ADDBLD	读操作的第 1 个阶段的长度（ADDBLD+1 个 HCLK 周期）

表 20-14 XFMC_BANK1_WTRx 位域（模式 A）

位编号	位名称	设置的数值
31-30		0x0
29-28	ACCMODE	0x0
27-16		0x000
15-8	DATABLD	写操作的第 2 个阶段的长度（DATABLD +1 个 HCLK 周期），这个域不能为 0（至少为 1）
7-4		0x0
3-0	ADDBLD	写操作的第 1 个阶段的长度（ADDBLD+1 个 HCLK 周期）

20.5.4.4 模式 2/B — NOR 闪存

图 20-7 模式 2/B 读操作

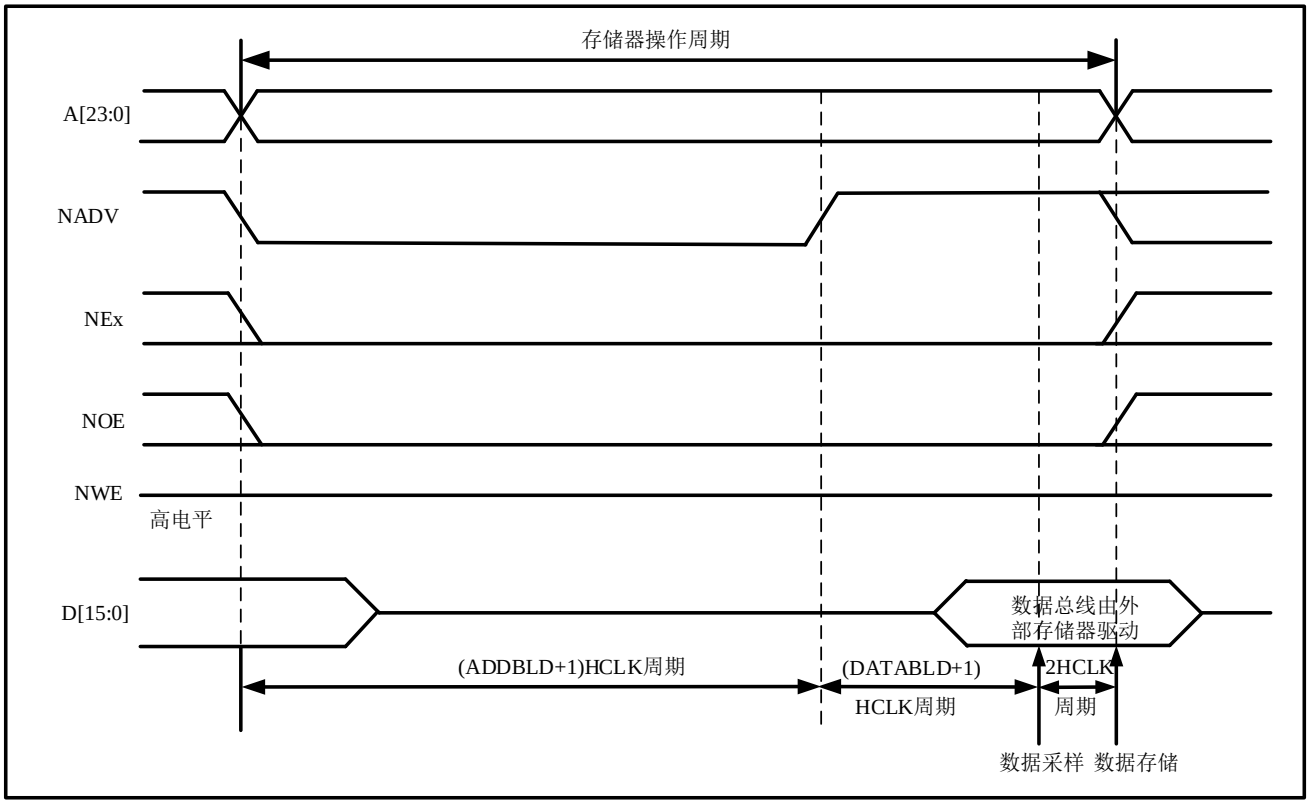


图 20-8 模式 2 写操作

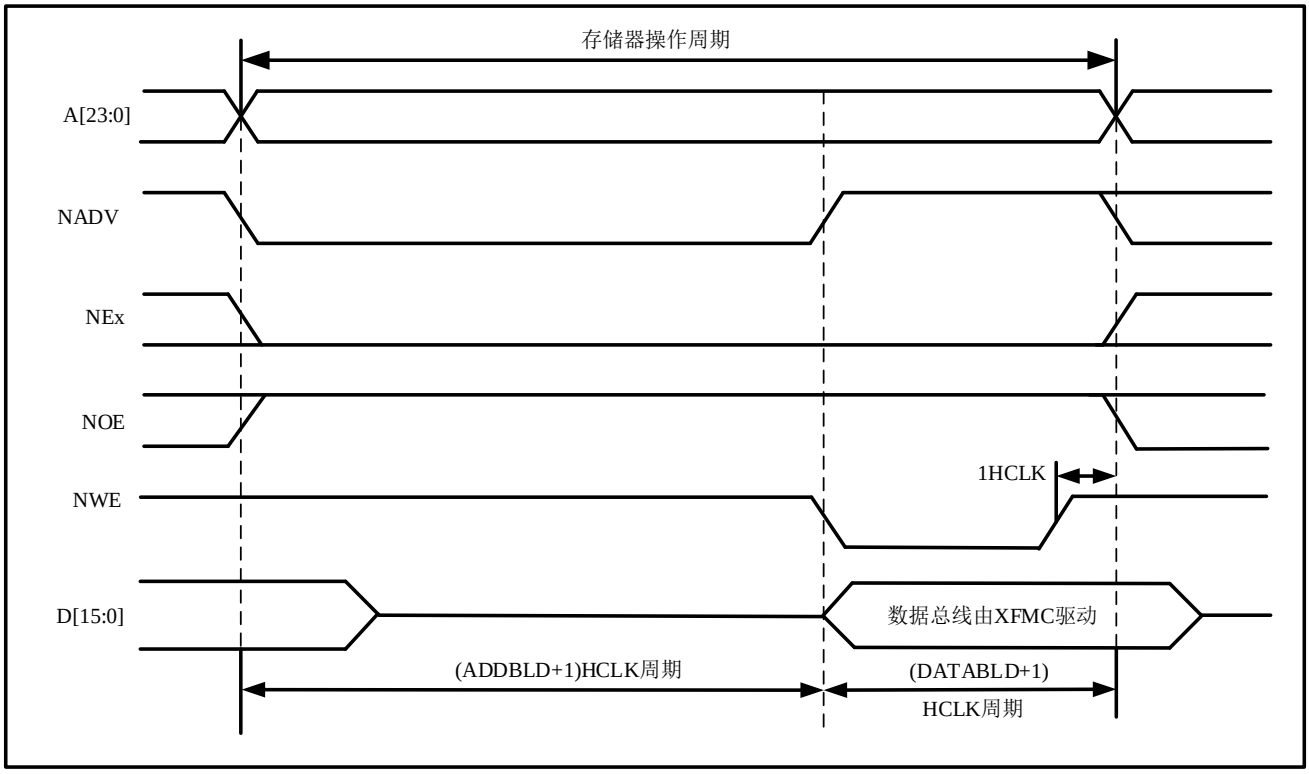
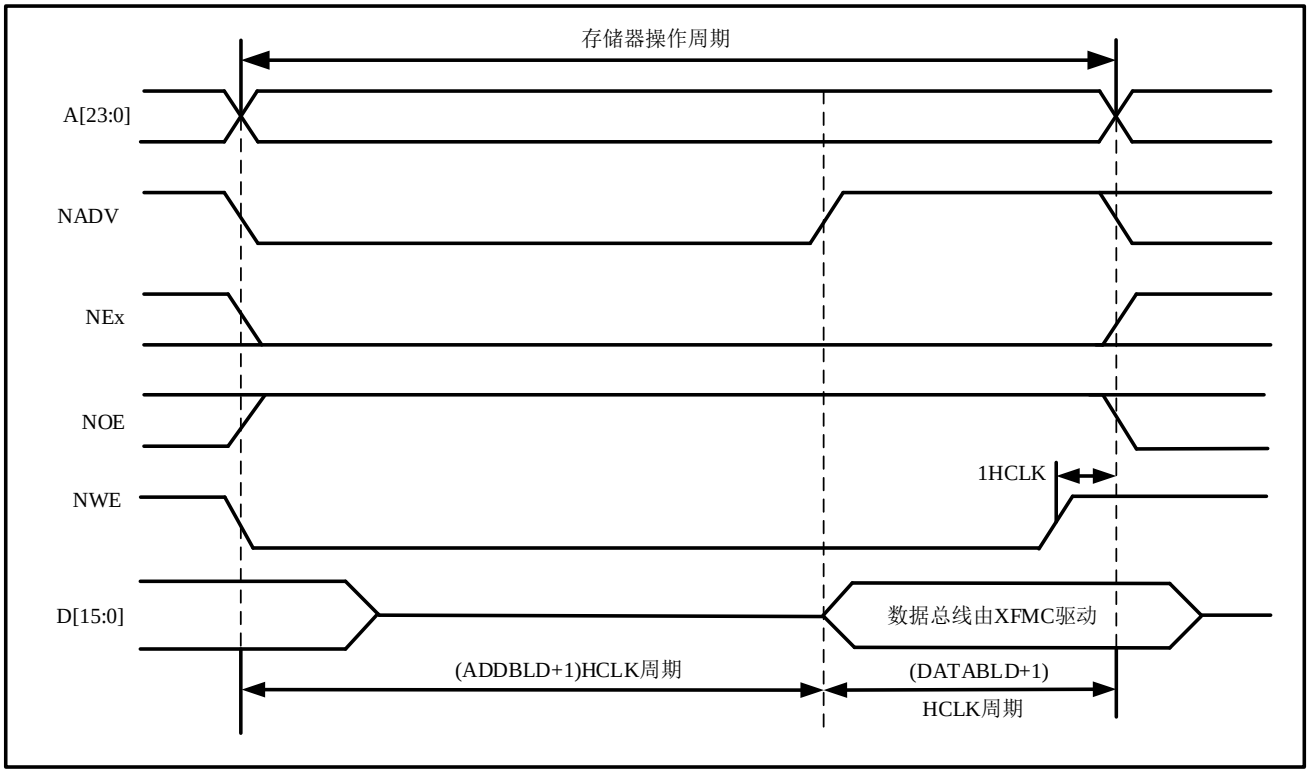


图 20-9 模式 B 写操作



注：ADDBLD：地址建立时间；DATABLD：数据建立时间

模式 2/B 与模式 1 相比较，不同的是 NADV 的变化，且在扩展模式下（模式 B）读写时序相互独立。

表 20-15 XFMC_BANK1_CRx 位域（模式 2/B）

位编号	位名称	设置的数值
31-15		0x0000
14	EXTEN	模式 B：0x1；模式 2：0x0
13-10		0x0
9	WAITDIR	仅当位 15 为 ' 1 ' 时有意义
8	BURSTEN	0x0
7		-
6	ACCEN	0x1
5-4	MDBW	需要时设置
3-2	MTYPE	10（NOR 闪存）
1	MUXEN	0x0
0	MBEN	0x1

表 20-16 XFMC_BANK1_TRx 位域（模式 2/B）

位编号	位名称	设置的数值
31-30		0x0
29-28	ACCMODE	如果使用了扩展模式，设置为 0x1
27-16		0x000
15-8	DATABLD	读操作的第 2 个阶段的长度（DATABLD +3 个 HCLK 周期），这个域不能为 0（至少为 1）
7-4		0x0
3-0	ADDBLD	读操作的第 1 个阶段的长度（ADDBLD+1 个 HCLK 周期）

表 20-17 XFMC_BANK1_WTRx 位域（模式 2/B）

位编号	位名称	设置的数值
31-30		0x0
29-28	ACCMODE	如果使用了扩展模式，设置为 0x1
27-16		0x000
15-8	DATABLD	写操作的第 2 个阶段的长度（DATABLD +1 个 HCLK 周期），这个域不能为 0（至少为 1）
7-4		0x0
3-0	ADDBLD	写操作的第 1 个阶段的长度（ADDBLD+1 个 HCLK 周期）

注：只有当设置了扩展模式时（模式 B），XFMC_BANK1_WTRx 才有效，否则该寄存器的内容不起作用。

20.5.4.5 模式 C — NOR 闪存 - OE 翻转

图 20-10 模式 C 读操作

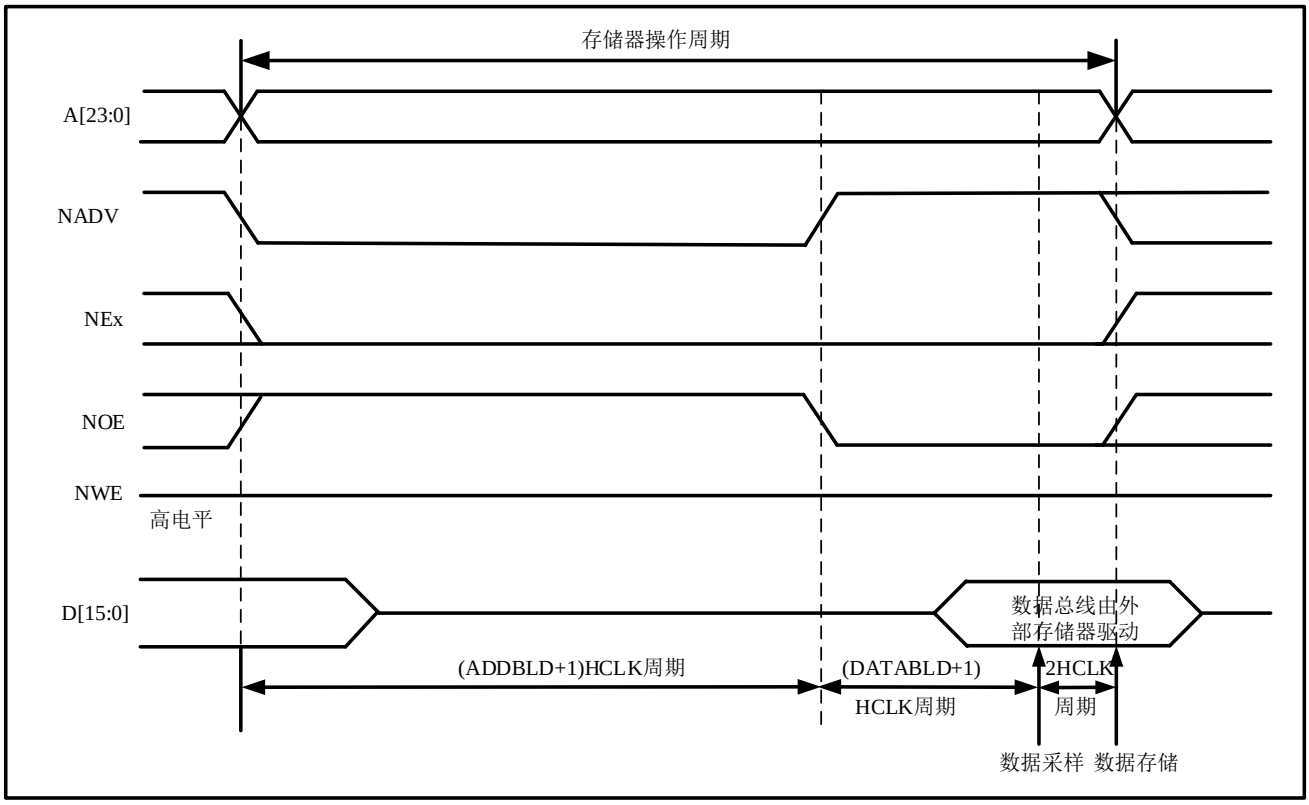
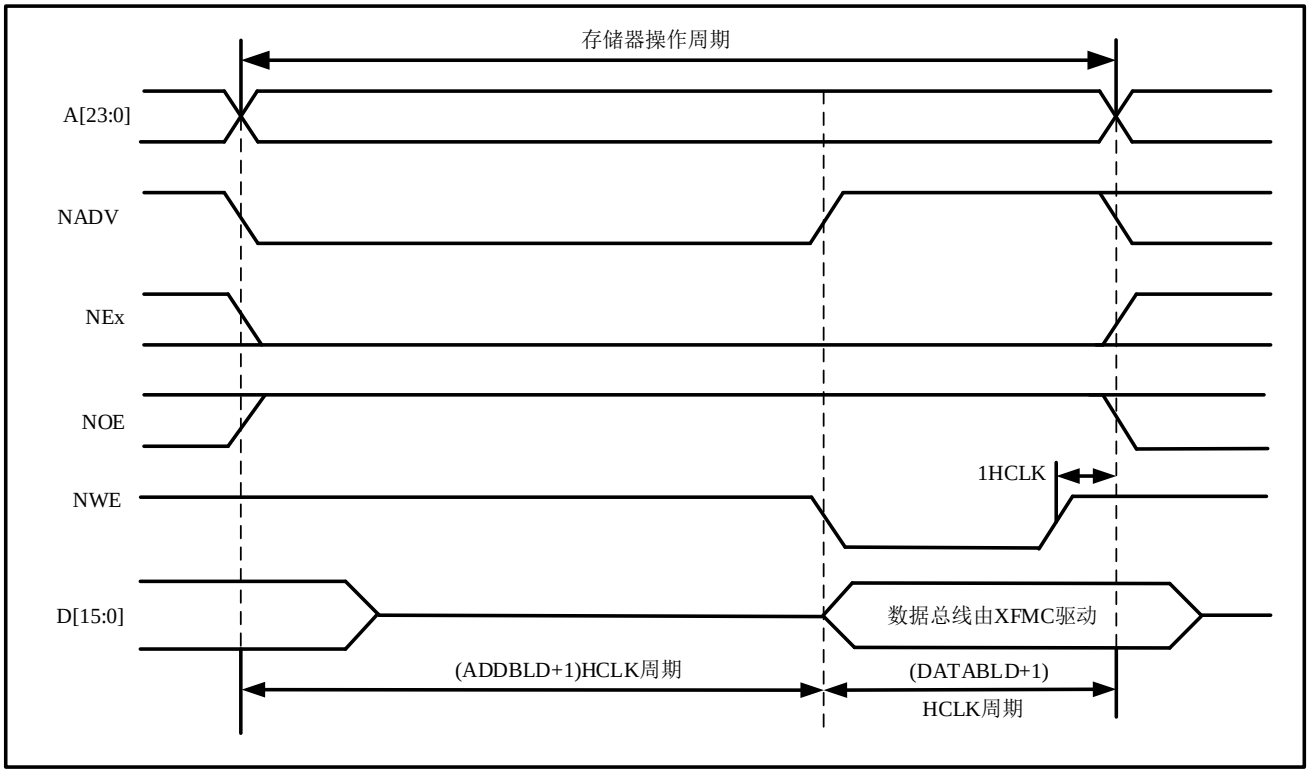


图 20-11 模式 C 写操作



注：ADDBLD：地址建立时间；DATABLD：数据建立时间

模式 C 和模式 1 的区别在于写时序，当两个模式的寄存器有相同的时序配置时， 模式 C 的写时序独立于读时序。

表 20-18 XFMC_BANK1_CRx 位域（模式 C）

位编号	位名称	设置的数值
31-15		0x0000
14	EXTEN	0x1
13-10		0x0
9	WAITDIR	仅当位 15 为 ' 1 ' 时有意义
8	BURSTEN	0x0
7		-
6	ACCEN	0x1
5-4	MDBW	需要时设置
3-2	MTYPE	0x2（NOR 闪存）
1	MUXEN	0x0
0	MBEN	0x1

表 20-19 XFMC_BANK1_TRx 位域（模式 C）

位编号	位名称	设置的数值
31-30		0x0
29-28	ACCMODE	0x2
27-16		0x000
15-8	DATABLD	读操作的第 2 个阶段的长度（DATABLD +3 个 HCLK 周期），这个域不能为 0（至少为 1）
7-4		0x0
3-0	ADDBLD	读操作的第 1 个阶段的长度（ADDBLD+1 个 HCLK 周期）

表 20-20 XFMC_BANK1_WTRx 位域（模式 C）

位编号	位名称	设置的数值
31-30		0x0
29-28	ACCMODE	0x2
27-16		0x000
15-8	DATABLD	写操作的第 2 个阶段的长度（DATABLD +1 个 HCLK 周期），这个域不能为 0（至少为 1）
7-4		0x0
3-0	ADDBLD	写操作的第 1 个阶段的长度（ADDBLD+1 个 HCLK 周期）

20.5.4.6 模式 D — 带地址扩展的异步操作

图 20-12 模式 D 读操作

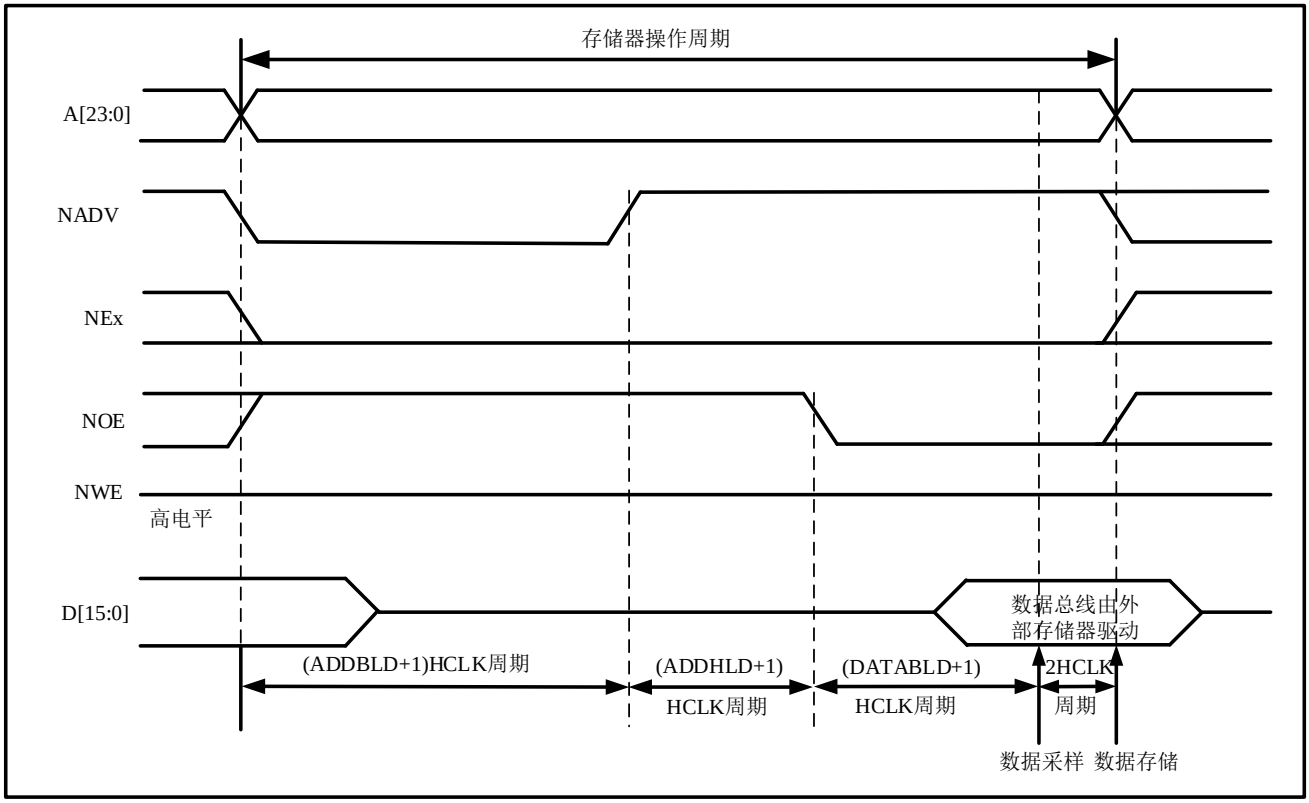
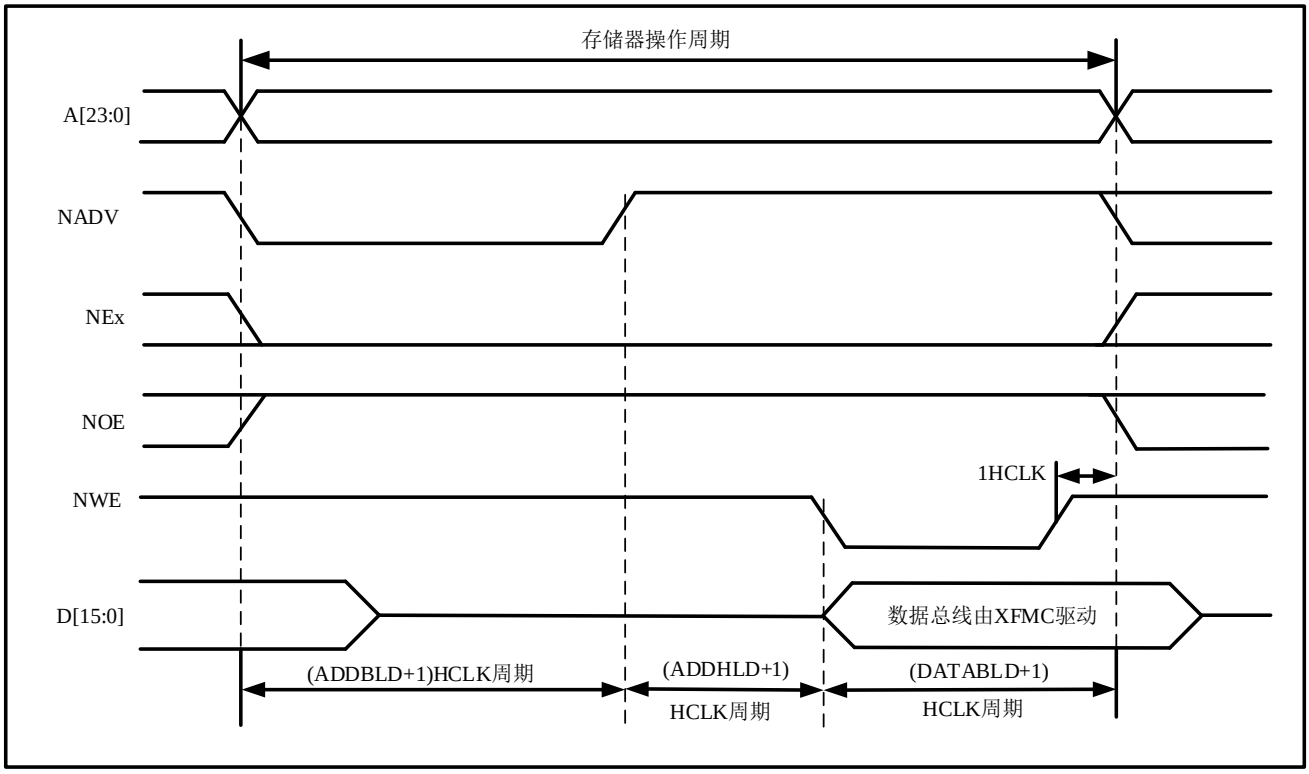


图 20-13 模式 D 写操作



注：ADDBLD：地址建立时间；ADDHLD：地址保持时间；DATABLD：数据建立时间

模式 D 与模式 1 不同的是 NADV 的翻转变化，NOE 的翻转出现在 NADV 翻转之后，并且具有独立的读写时序。

表 20-21 XFMC_BANK1_CRx 位域（模式 D）

位编号	位名称	设置的数值
31-15		0x0000
14	EXTEN	0x1
13-10		0x0
9	WAITDIR	仅当位 15 为 ' 1 ' 时有意义
8	BURSTEN	0x0
7		-
6	ACCEN	根据存储器设置
5-4	MDBW	需要时设置
3-2	MTYPE	需要时设置
1	MUXEN	0x0
0	MBEN	0x1

表 20-22 XFMC_BANK1_TRx 位域（模式 D）

位编号	位名称	设置的数值
31-30		0x0

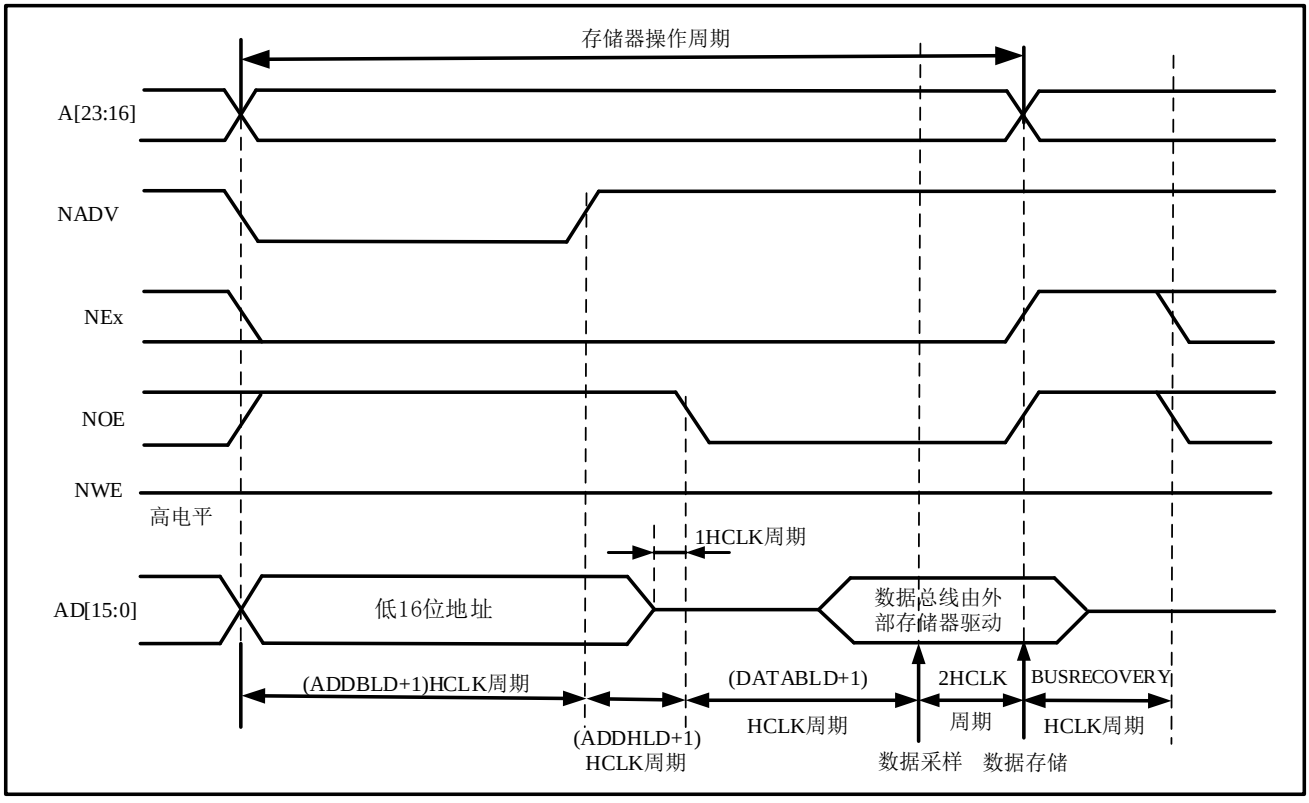
位编号	位名称	设置的数值
29-28	ACCMODE	0x2
27-16		0x000
15-8	DATABLD	读操作的第 2 个阶段的长度（DATABLD +3 个 HCLK 周期），这个域不能为 0（至少为 1）
7-4	ADDHLD	读操作的中间阶段的长度（ADDHLD +1 个 HCLK 周期）
3-0	ADDBLD	读操作的第 1 个阶段的长度（ADDBLD+1 个 HCLK 周期）

表 20-23 XFMC_BANK1_WTRx 位域（模式 D）

位编号	位名称	设置的数值
31-30		0x0
29-28	ACCMODE	0x2
27-16		0x000
15-8	DATABLD	写操作的第 2 个阶段的长度（DATABLD +1 个 HCLK 周期），这个域不能为 0（至少为 1）
7-4	ADDHLD	写操作的中间阶段的长度（ADDHLD +1 个 HCLK 周期）
3-0	ADDBLD	写操作的第 1 个阶段的长度（ADDBLD+1 个 HCLK 周期）

20.5.4.7 复用模式 — 地址/数据复用的 NOR 闪存异步操作

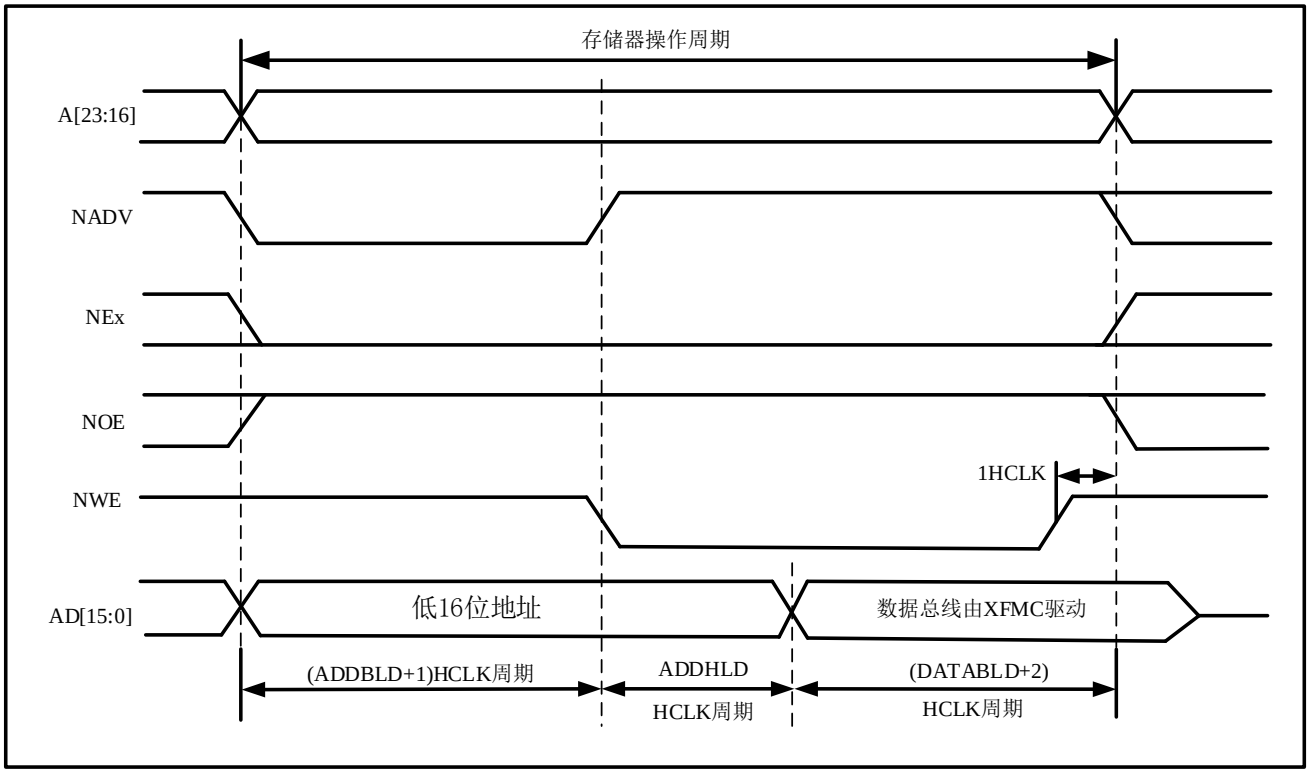
图 20-14 复用读操作



（1） ADDBLD：地址建立时间；DATABLD：数据建立时间；BUSRECOVERY：总线恢复时间

（2）总线恢复延迟（BUSRECOVERY+1）与连续 2 次读操作之间在内部产生的延迟有部分重叠，因此 BUSRECOVERY≤5 时将不影响输出时序。

图 20-15 复用写操作



注：ADDBLD：地址建立时间；ADDHLD：地址保持时间；DATABLD：数据建立时间

复用模式与模式 D 不同的是地址的低 16 位出现在数据总线上。

表 20-24 XFMC_BANK1_CRx 位域（复用模式）

位编号	位名称	设置的数值
31-15		0x0000
14	EXTEN	0x0
13-10		0x0
9	WAITDIR	仅当位 15 为 ' 1 ' 时有意义
8	BURSTEN	0x0
7		-
6	ACCEN	0x1
5-4	MDBW	需要时设置
3-2	MTYPE	0x2（NOR 闪存）
1	MUXEN	0x1
0	MBEN	0x1

表 20-25 XFMC_BANK1_TRx 位域（复用模式）

位编号	位名称	设置的数值
31-30		0x0
29-20		

位编号	位名称	设置的数值
19-16	BUSRECOVERY	操作的最后阶段的长度（BUSRECOVERY +1 个 HCLK 周期）
15-8	DATABLD	操作的第 2 个阶段的长度，读操作为（DATABLD +3 个 HCLK 周期），写操作为（DATABLD +1 个 HCLK 周期）这个域不能为 0（至少为 1）。
7-4	ADDHLD	操作的中间阶段的长度（ADDHLD +1 个 HCLK 周期）。这个域不能为 0（至少为 1）
3-0	ADDBLD	操作的第 1 个阶段的长度（ADDBLD+1 个 HCLK 周期）

20.5.5 同步的成组读

同步访问模式中，存储器时钟 CLK 与系统时钟 HCLK 的关系为：$CLK = \frac{HCLK}{CLKDIV+1}$，

其中 CLKDIV 是同步时钟分频比，通过配置 SRAM/NOR 闪存片选时序寄存器(XFMC_BANK1_TR1/2)中的 CLKDIV 位来设置不同的值。

NOR 闪存存储器有一个从 NADV 有效至 CLK 变高的最小时间限制，为了满足这个限制，在同步访问（NADV 有效之前）的第一个内部时钟周期中，XFMC 不会输出时钟到存储器。这样可以保证存储器时钟的上升沿产生于 NADV 低脉冲的中间。

20.5.5.1 数据延时与 NOR 闪存的延时

数据延时时间（DATAHLD）是指在采样数据之前需等待的 XFMC CLK 周期数目，用户可以调整 DATAHLD 的数值使其与 NOR 闪存配置寄存器中定义的数值相符合。XFMC 的数据延时参数不包含 NADV 为低时的时钟周期。它和 NOR 闪存延迟的关系如下：

- 当 NOR 闪存延时不包含 NADV 时，NOR 闪存延时 = DATAHLD + 2;
- 当 NOR 闪存延时包含 NADV 时，NOR 闪存延时 = DATAHLD + 3

如果存储器不在数据保持阶段输出 NWAIT 信号，XFMC 必须设置其数据保持时间（DATAHLD）和存储器端的数据保持时间一致，否则在访问存储器的初始阶段会丢失数据，或者 XFMC 得不到正确的数据。

如果存储器在数据保持阶段产生了一个 NWAIT 信号，这时 XFMC 可以设置其数据保持时间（DATAHLD）为最小值。XFMC 会采样 NWAIT 信号，并一直等待直到数据有效，XFMC 会在检测到存储器结束了保持阶段后读取正确的数据。

20.5.5.2 单次成组传输

当选中的存储器块配置为同步成组模式，如果仅需要进行一次 AHB 单次成组传输，如果 AHB 需要传输 16 位数据，则 XFMC 会执行一次长度为 1 的成组传输；如果 AHB 需要传输 32 位数据，则 XFMC 会分成 2 次 16 位传输，执行一次长度为 2 的成组传输；最后一个数据传输完毕时撤消片选信号。

20.5.5.3 等待管理

对于同步的 NOR 闪存成组访问，在预置的保持时间（DATAHLD+1 个 CLK 时钟周期）之后，需检测 NWAIT 信号。

- XFMC NWAIT 有效极性：
  WAITDIR = 1，NWAIT 高电平有效
  WAITDIR = 0，NWAIT 低电平有效

- 在同步突发模式中，NWAIT 信号有两种配置：
  WCFG= 1，NWAIT 信号有效时，当前时钟周期数据无效
  WCFG= 0，NWAIT 信号有效时，下一个时钟周期数据无效，这是复位后的默认配置。

如果检测到 NWAIT 为有效电平时， XFMC 将插入等待周期直到 NWAIT 变为无效电平。

当 NWAIT 变为无效时，XFMC 认为数据已经有效（WCFG=1），或数据将在下一个时钟边沿有效（WCFG=0）。

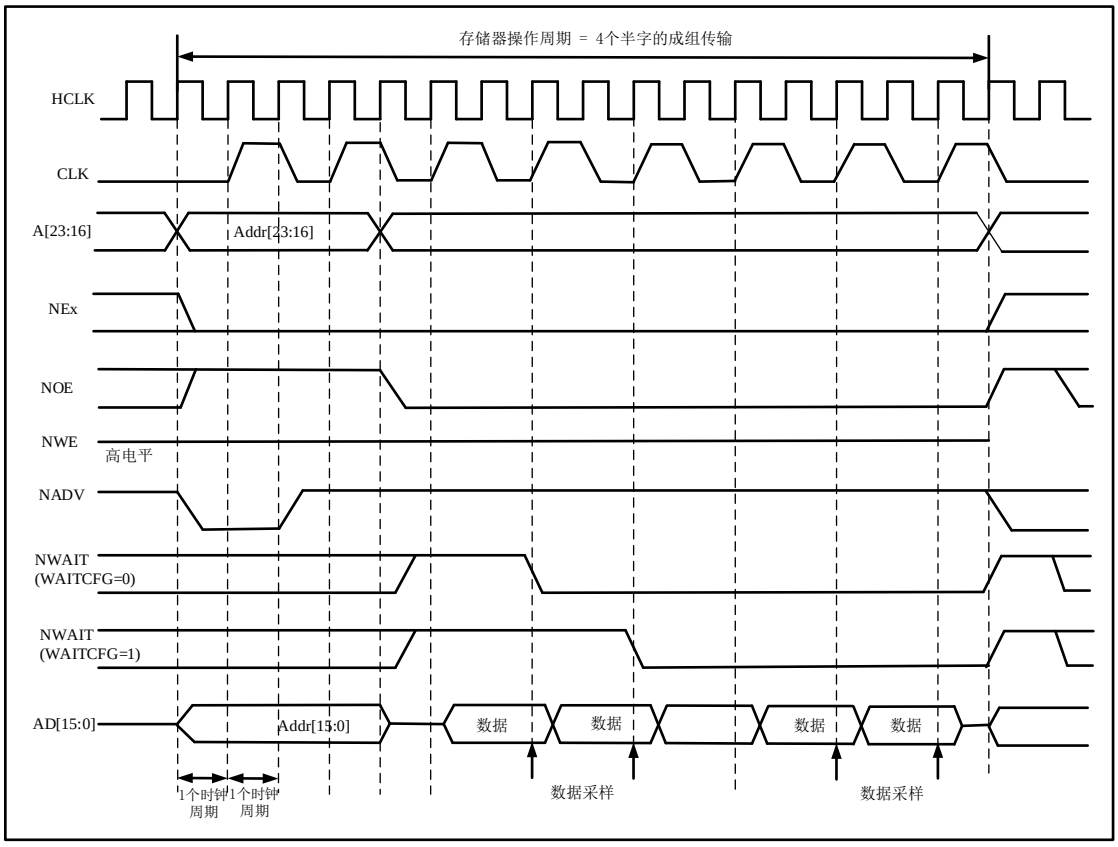
在 NWAIT 信号有效的等待周期内，XFMC 会持续的给存储器发送时钟脉冲、保持片选信号和输出使能有效，同时忽略总线上的无效数据。

在成组传输模式下，NOR 闪存的 NWAIT 信号有 2 种时序配置：

- 闪存存储器在等待状态之前的一个数据周期插入 NWAIT 信号（复位后的默认设置）；
- 闪存存储器在等待状态期间插入 NWAIT 信号。

通过配置 XFMC_BANK1_CRx 寄存器中的 WCFG 位，XFMC 在每个片选上都支持这 2 种 NOR 闪存的等待状态配置。

图 20-16 同步的总线复用读模式—NOR, PSRAM（CRAM）



BL 信号没有显示在图中，对于 NOR 闪存 BL 应该为高；对于 PSRAM（CRAM）， BL 应该为低。

表 20-26 XFMC_BANK1_CRx 位域（同步模式）

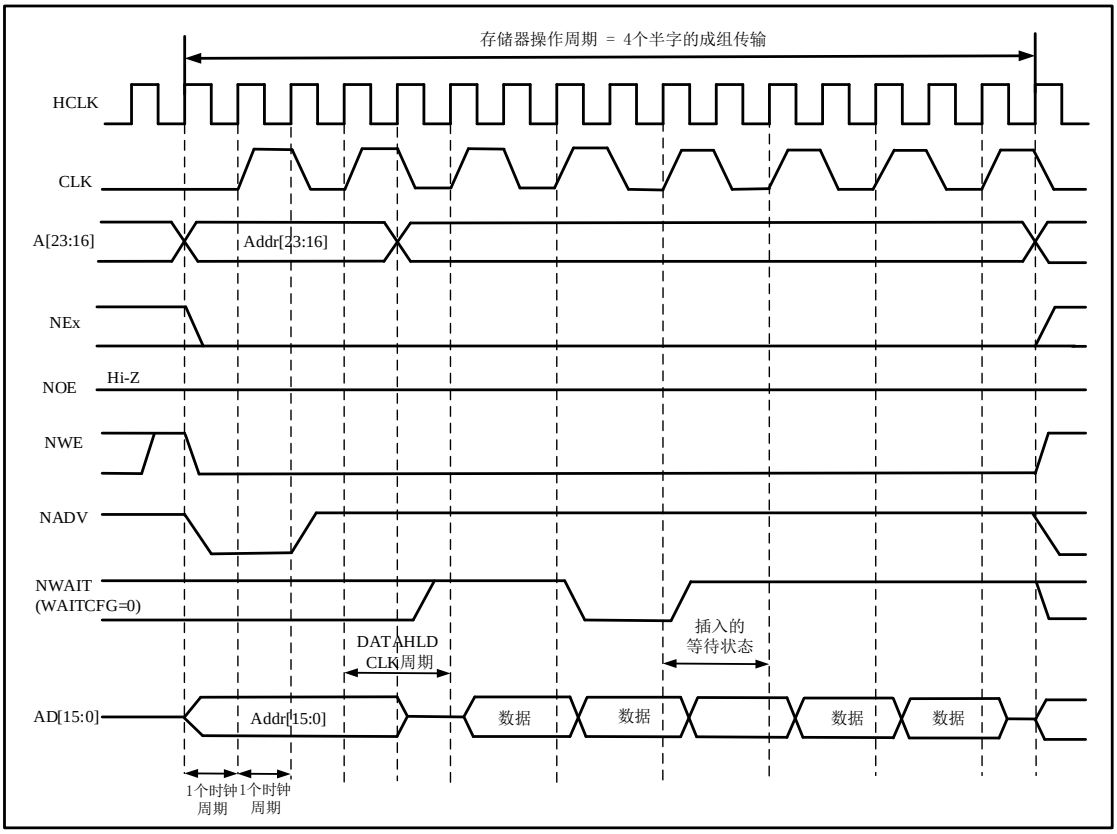
位编号	位名称	设置的数值
31-20		0x0000

位编号	位名称	设置的数值
19	BURSTWREN	对同步读模式不起作用
18-15		0x0
14	EXTEN	0x0
13	WAITEN	当此位为高时，不管存储器的等待数值是多少，XFMC 视数据保持阶段结束后的第一个数据有效
12	WREN	对同步读模式不起作用
11	WCFG	根据存储器特性设置
10	WRAPEN	根据存储器特性设置
9	WAITDIR	根据存储器特性设置
8	BURSTEN	0x1
7	FWPRLVL	设置此位防止存储器被意外写
6	ACCEN	根据存储器设置
5-4	MDBW	根据需要设置
3-2	MTYPE	0x1 或 0x2
1	MUXEN	根据需要设置
0	MBEN	0x1

表 20-27 XFMC_BANK1_TRx 位域（同步模式）

位编号	位名称	设置的数值
31-30	RESERVED	0x0
29-28	ACCMODE	0x0
27-24	DATAHLD	数据保持时间
23-20	CLKDIV	0x0—得到 CLK = HCLK（不支持） 0x1—得到 CLK = 2 x HCLK
19-16	BUSRECOVERY	不起作用
15-8	DATABLD	不起作用
7-4	ADDHLD	不起作用
3-0	ADDBLD	不起作用

图 20-17 同步写模式 – PSRAM（CRAM）



1．存储器必须提前一个周期产生 NWAIT 信号，同时 WAITCFG 应配置为 0。

2．字节选择 BL 输出没有显示在图中，当 NEx 为有效时它们为低。

表 20-28 XFMC_BANK1_CRx 位域（同步模式）

位编号	位名称	设置的数值
31-20		0x0000
19	BURSTWREN	0x1
18-15		0x0
14	EXTEN	0x0
13	WAITEN	当此位为高时，不管存储器的等待数值是多少，XFMC 视数据保持阶段结束后的第一个数据有效
12	WREN	0x1
11	WCFG	0x0
10	WRAPEN	根据存储器特性设置
9	WAITDIR	根据存储器特性设置
8	BURSTEN	对同步写模式不起作用
7	RESERVED	0x1
6	ACCEN	根据存储器设置
5-4	MDBW	根据需要设置
3-2	MTYPE	0x1

位编号	位名称	设置的数值
1	MUXEN	根据需要设置
0	MBEN	0x1

表 20-29 XFMC_BANK1_TRx 位域（同步模式）

位编号	位名称	设置的数值
31-30	RESERVED	0x0
29-28	ACCMODE	0x0
27-24	DATAHLD	数据保持时间
23-20	CLKDIV	0x0—得到 CLK = HCLK（不支持） 0x1—得到 CLK = 2 x HCLK
19-16	BUSRECOVERY	NEx 高和 NEx 低之间的时间（BUSRECOVERY HCLK）
15-8	DATABLD	不起作用
7-4	ADDHLD	不起作用
3-0	ADDBLD	不起作用

20.6 NAND 闪存控制器

XFMC 模块 Bank2、Bank3 支持 8 位、16 位的 Nand Flash，对于每个 Bank，XFMC 提供独立的寄存器来配置访问时序，同时还提供 ECC 计算模块，保证数据传输和保存的可靠性。

表 20-30 可编程 NAND

参数	功能	操作模式	单位	最小	最大
存储器建立时间	发出命令之前建立地址的（HCLK）时钟周期数目	读/写	AHB 时钟周期（HCLK）	1	256
存储器等待时间	发出命令的最短持续时间（HCLK 周期数目）	读/写		1	256
存储器保持时间	在发送命令结束后保持地址的（HCLK）时钟周期数目，写操作时也是数据的保持时间	读/写		1	255
存储器数据总线高阻时间	启动写操作之后保持数据总线为高阻态的时间	写		0	255

20.6.1 外部存储器接口信号

下表列出了用于接口 NAND 闪存的典型信号线。

前缀‘N’代表对应的信号线为低电平有效。

20.6.1.1 8 位 NAND 闪存

表 20-31 8 位 NAND 闪存

XFMC 信号名称	输入/输出	功能
A[17]	输出	NAND 闪存地址锁存允许信号（ALE）
A[16]	输出	NAND 闪存命令锁存允许信号（CLE）
D[7:0]	入/出	8 位复用的、双向地址/数据总线
NCE[x]	输出	片选， x = 2， 3
NOE	输出	输出使能

XFMC 信号名称	输入/输出	功能
NWE	输出	写使能
NWAIT	输入	NAND 闪存就绪/繁忙，输入至 XFMC 的信号

XFMC 可以根据需要产生多个地址周期，理论上 XFMC 不限制可以访问的 NAND 容量。

20.6.1.2 16 位 NAND 闪存

表 20-32 16 位 NAND 闪存

XFMC 信号名称	输入/输出	功能
A[17]	输出	NAND 闪存地址锁存允许信号（ALE）
A[16]	输出	NAND 闪存命令锁存允许信号（CLE）
D[15:0]	入/出	16 位复用的、双向地址/数据总线
NCE[x]	输出	片选， x = 2，3
NOE	输出	输出使能
NWE	输出	写使能
NWAIT	输入	NAND 闪存就绪/繁忙，输入至 XFMC 的信号

XFMC 可以根据需要产生多个地址周期，理论上 XFMC 不限制可以访问的 NAND 容量。

20.6.2 NAND 闪存支持的存储器及其操作

下表列出了支持的设备、操作模式和操作方式。在表格中有阴影的部分表示 NAND 闪存不支持对应的操作方式。

表 20-33 支持的存储器及其操作

存储器	模式	读/写	AHB 数据宽度	存储器数据宽度	是否支持	注释
8 位 NAND	异步	读	8	8	支持	
	异步	写	8	8	支持	
	异步	读	16	8	支持	分成 2 次 XFMC 访问
	异步	写	16	8	支持	分成 2 次 XFMC 访问
	异步	读	32	8	支持	分成 4 次 XFMC 访问
	异步	写	32	8	支持	分成 4 次 XFMC 访问
16 位 NAND	异步	读	8	16	支持	
	异步	写	8	16	不支持	
	异步	读	16	16	支持	
	异步	写	16	16	支持	
	异步	读	32	16	支持	分成 2 次 XFMC 访问
	异步	写	32	16	支持	分成 2 次 XFMC 访问

20.6.3 NAND 闪存时序图

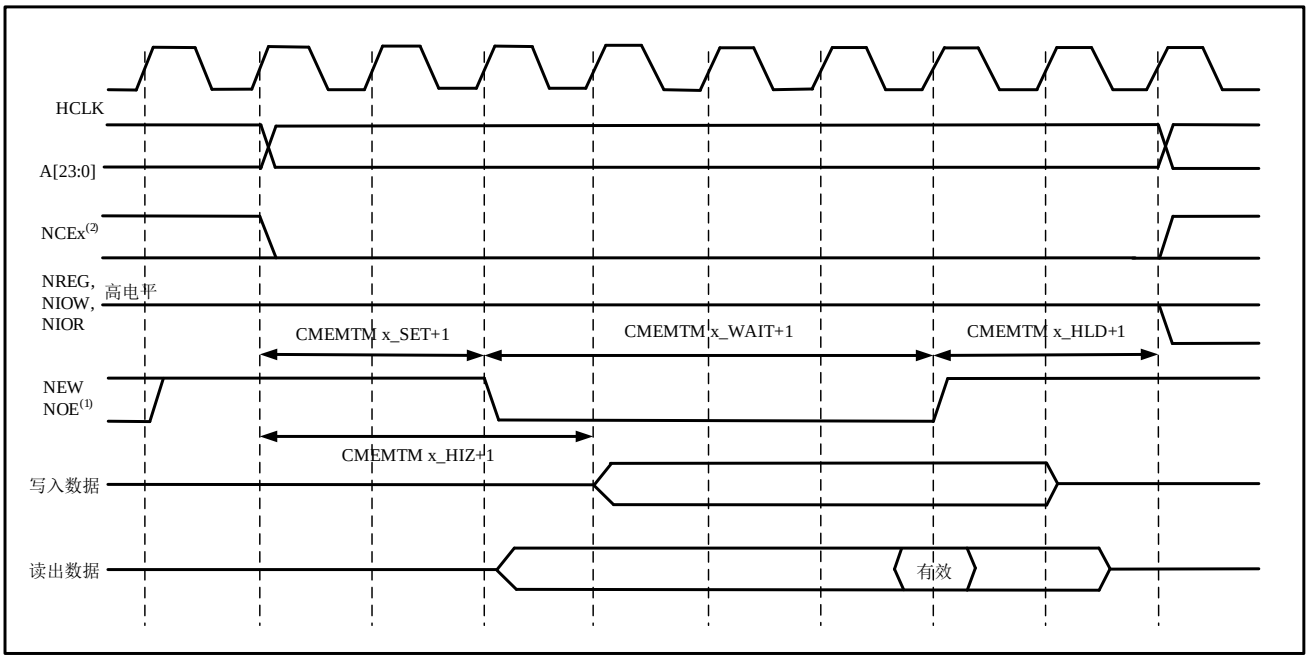
XFMC 能够为 Nand Flash 产生合适的时序信号，每个 Bank 都有相应的寄存器来对外部存储器进行管理和控制，涉及到的寄存器组如下：

- 控制寄存器：XFMC_CTRLx

- 中断状态寄存器：XFMC_STSx
- ECC 寄存器：XFMC_ECCx
- 通用存储器空间的时序寄存器：XFMC_CMEMTM x
- 属性存储器空间的时序寄存器：XFMC_ATTMEMTM x

每一个时序控制寄存器都包含 3 个参数，用于定义 NAND 闪存操作中三个阶段的 HCLK 周期数目，还有一个定义了写操作中 XFMC 开始驱动数据总线时机的参数，可以根据用户需求和外部存储器的特性来进行相应的配置。下图给出了在通用存储空间中操作的时序参数定义，属性存储空间中操作与此相似。

图 20-18 NAND 控制器通用存储空间的访问时序



1. 在写操作时 NOE 始终保持高（无效状态），在读操作时 NWE 始终保持高（无效状态）。

2. 只要请求 NAND 访问， NCEx 信号就变低并在访问其它存储器块之前保持为低。

20.6.4 NAND 闪存操作

NAND 闪存的命令锁存使能（CLE）和地址锁存使能（ALE）信号由 XFMC 的地址信号线驱动，即 XFMC 在对 NAND Flash 发送命令或地址时，需要利用其命令锁存信号（A[16]）或地址锁存信号（A[17]）这两条地址线，即 CPU 需要在特定的地址进行写操作。

NAND 闪存读操作步骤：

1. 根据 NAND 闪存的特性，通过 XFMC_CTRLx 和 XFMC_CMEMTM x 寄存器配置和使能相应的存储器块，对于某些 NAND 闪存如果需要预等待功能，可能还要配置 XFMC_ATTMEMTMx 寄存器（见 20.6.5 节—NAND 闪存预等待功能）。需要配置的位包括：BUSWID 指示 NAND 闪存的数据总线宽度，MEMTYPE =1， WAITEN =1，BANKEN =1，参见 XFMC_CMEMTM 2/3 寄存器的时序配置。
2. CPU 往通用空间写入 Nand Flash 读数据命令，即在片选和写信号有效期间（XFMC NCE 和 XFMC NWE 为低脉冲），NAND 闪存的 CLE（A16）变为有效电平（高），这时 CPU 写的字节被 NAND 闪

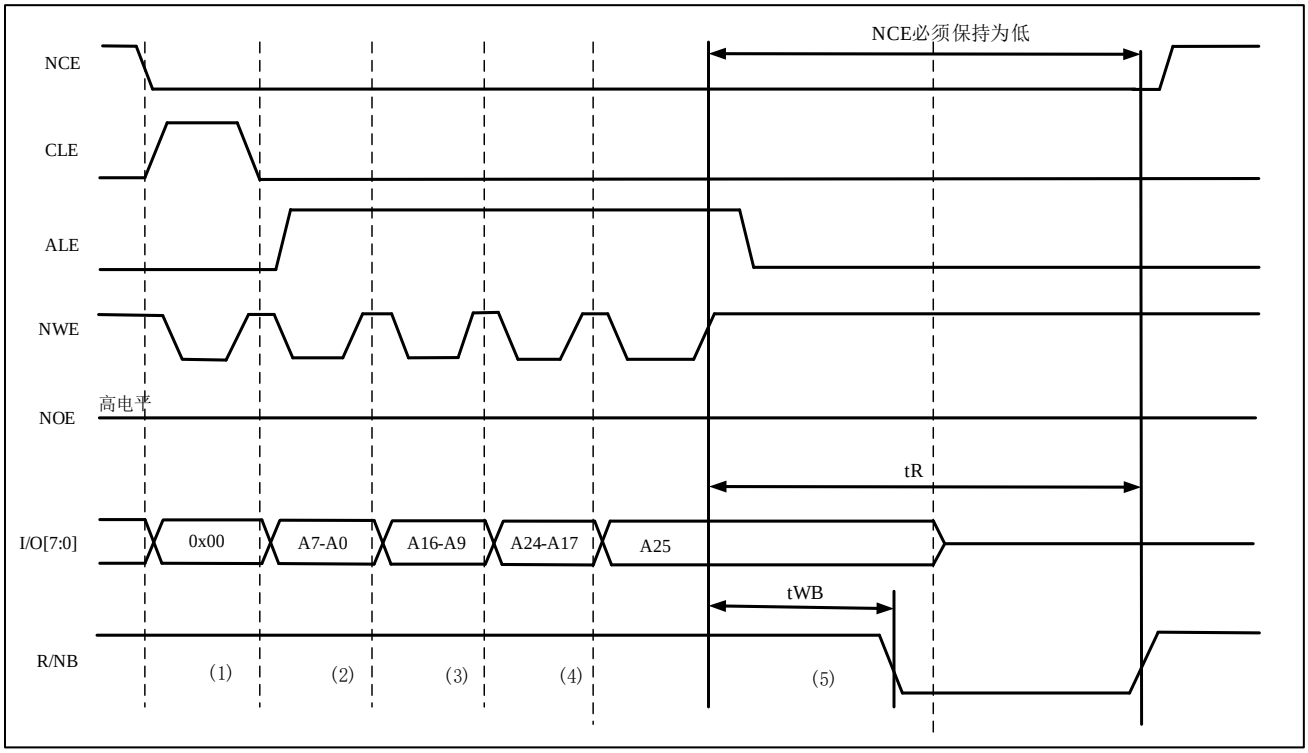
存识别为一个命令。一旦 NAND 闪存锁存了这个命令，随后的页读操作不必再发送相同的命令。

3. CPU 往通用存储器空间写入读操作的起始地址（大容量 NADN 闪存需要 4 个字节，较小容量的 NAND 闪存可能只需要三个字节）。即在片选和写信号有效期间（XFMC NCE 和 XFMC NWE 为低脉冲），NAND 闪存的 ALE（A17）变为有效电平（高），这时 CPU 写的字节被 NAND 闪存识别为读操作的开始地址。使用属性存储空间，可以使 XFMC 产生不同的时序，实现某些 NAND 闪存所需的预等待功能（见 20.6.5 节—NAND 闪存预等待功能）。
4. 等待 NAND 闪存准备就绪（R/B 信号变为高），在等待期间 NAND 控制器保持 NCE 信号一直有效（低电平）。
5. CPU 可以在通用存储空间执行字节读操作，逐字节地读出 NAND 闪存的存储页（数据域和后备域）
6. 在 CPU 不写入命令或地址的情况下，NAND 闪存的下一个页可以以下述任一种方式读出：
   a) 按照步骤 5 进行操作
   b) 返回步骤 3 开始输入一个新的地址
   c) 返回步骤 2 开始输入一个新的命令

20.6.5 NAND 闪存预等待功能

某些 NAND Flash 要求在输入最后一个地址字节后，控制器等待 NAND Flash 就绪，即 R/B 信号变低，并且还有一些对 XFMC 片选信号 NCE 敏感型的 NAND Flash 还要求在其就绪前 NCE 必须保持有效。如下图：

图 20-19 操作 CE 敏感型 NAND 闪存



1. CPU 往 NAND 的通用空间命令区（0x7001 0000）写入字节命令 0x00
2. CPU 往 NAND 的通用空间地址区（0x7002 0000）写入操作地址 A7~A0
3. CPU 往 NAND 的通用空间地址区（0x7002 0000）写入操作地址 A16~A9
4. CPU 往 NAND 的通用空间地址区（0x7002 0000）写入操作地址 A24~A17
5. CPU 往 NAND 的属性空间地址区（0x7802 0000）写 NAND 的地址 A25，这时 XFMC 使用 XFMC_ATTMEMTM 的时序定义执行写操作，此时 ATTHLD≥7 （这里：(7+1)xHCLK=112ns > t_{WB}的最大值）。这样可以保证 R/B 变低再变高的过程中 NCE 保持为低，只有那些对 CE 敏感型的 NAND 闪存有此要求。

当需要这样的功能时，可以通过配置 MEMHLD 的数值来保证 t_{WB} 的时序，但是任何随后的 CPU 对 NAND 闪存的读或写操作中，XFMC 控制器都会在 NWE 信号的上升沿至下一次操作之间插入一个保持延迟，延迟长度为（MEMHLD+1）个 HCLK 周期。

为了克服这个时序的限制，这里使用了属性存储空间配置 ATTHLD 的数值使之符合 t_{WB} 的时序，同时保持 MEMHLD 为其最小值。此时，CPU 必须在所有 NAND 闪存的读写操作时使用通用存储空间，只有在写入 NAND 闪存地址的最后一个字节时，CPU 需要写入属性存储空间。

20.6.6 NAND 闪存的纠错码 ECC 计算（NAND 闪存）

XFMC 的存储块 2 和 3 各有一个纠错码计算的硬件模块，该模块可以用于减小 CPU 在处理纠错码时的软件工作量。有两个相同的寄存器（XFMC_ECC2、XFMC_ECC3）分别对应存储块 2 和存储块 3。

XFMC 中实现的纠错码（ECC）算法可以在读或写 NAND 闪存时，在每 256、 512、 1024、 2048、 4096 或 8192 个字节中，矫正 1 个比特位的错误并且检测出 2 个比特位的错误，用户可以根据 XFMC_CTRLx 中的 ECCPGS 来选择 ECC 计算的页面大小。

当 NAND 存储器块使能时，ECC 模块就会监测 NAND 闪存的数据总线 D[15：0]以及 XFMC NCE、XFMC_NEW 信号，当已经完成 ECCPGS 大小字节的读写操作时，硬件就会完成 ECC 的计算并将结果保存在 XFMC_ECCx 寄存器中，软件通过读 XFMC_ECCx 中的结果来获取 ECC 值，如果需要再次开始 ECC 计算，软件需要先将 XFMC_CTRLx 中的 ECCEN 清 0 来清除 XFMC_ECC 寄存器中的值，再将 ECCEN 置 1 来重新启动 ECC 计算。

ECC 计算规则如下：

- 当在 Bank2 或 Bank3 访问 NAND 闪存时，出现在 D[15:0]总线上的数据被锁存并用于 ECC 计算。
- 当对 NAND 闪存的操作发生在其它地址时， ECC 电路不进行任何操作。因此输出 NAND 闪存命令和地址的写操作不会参与 ECC 计算。

ECC 纠错原理如下：

在 ECC 的数据里，用 2 个位来表示一个位的数据。比如 512 个字节共有 512*8=4096 个位，要表示到 4096 个数，也就是刚好 12 位就可以了，但是实际上根据表“ECC 结果有效位”（表 20-36）可知 512 个字节的 ECC 用了 24 位，1024 个字节的 ECC 用了 26 位（按常理说，只要多一个位，就可以表示多一倍的数据了。但是 ECC 要表示多一倍数据，需要多两个位）均可以看出这个规律。

我们以 512 个字节产生一个 ECC 数据为例，根据表“ECC 结果有效位”（表 20-36）可知 ECC 的有效位为 0 到 23，当写入 512 个字节时，产生一个 ECC 数据，然后，再读 512 个字节的时候，也会产生一个 ECC 数据，通过这两个数据相异或，产生一个 ECC 校验结果，通过这个判断结果，就可以分析出

有没有错，错在哪里，是错哪一位。当然，知道哪一位错了，也就可以纠正了。

下面举例说明 ECC 纠错原理：

1、 为了数据分析方便，向 Nand Flash 写入 512 字节的 0x11（其它数据也可以，数据有规律就可以），然后可以读出 ECC 数据为 0；
2、 改变其中某个字节的某个位，比如改变第 0 个字节的第 0 位（第 0 个字节改为 0x10），可以读出 ECC 数据为 0x55555555
3、 改变第 0 个字节的第 1 位（第一个字节改为 0x13），可以读出 ECC 数据为 0x55555556
4、 改变第 0 个字节的第 2 位（第一个字节改为 0x15），可以读出 ECC 数据为 0x55555559
5、 改变第 0 个字节的第 3 位（第一个字节改为 0x19），可以读出 ECC 数据为 0x5555555a

这个例子中，我们是以写入 0x11 为主，然后故意改错某一个位，再读出 ECC。因为写入全部的是 0x11，因此写入时产生的 ECC 是 0x0。那么再将读出来的 ECC 与 0x0 异或，得到的，也就是 ECC 校验结果了。下表是以上测试结果对应的 ECC 数据：

表 20-34 ECC 错误 bit 与 ECC 校验值对照表

错误 bit	校验结果（取低 24 位）	对应 24bit
Bit0	0x555555	010101010101010101010101
Bit1	0x555556	010101010101010101010110
Bit2	0x555559	010101010101010101011001
Bit3	0x55555a	010101010101010101011010

到这里就很容易看出来规律了，校验结果中用了 2 位来表示一个位的数据,其中 01 表示 0，10 表示 1。比如：第 0 个位错误的 24 位数据（010101010101010101010101）解码出来 12 位就是：000000000000（每一个 01 表示 0），也就是 0x0，就是第 0 个位错误；第 1 个位错误的 24 位数据解码出来 12 位就是：000000000001，也就是 0x1，就是第 1 个位错误；第 2 个位错误的 24 位数据解码出来 12 位就是：000000000010，也就是 0x2，就是第 2 个位错误；第 3 个位错误的 24 位数据解码出来 12 位就是：000000000011，也就是 0x3，就是第 3 个错误。到了这一步，纠正原理就很明白了。当知道是哪一个错误了，剩下的就简单了，就把错误的那一位把 0 变成 1，或者把 1 变成 0 就行了。

20.7 XFMC 寄存器

20.7.1 XFMC 寄存器地址映象

表 20-35 XFMC 寄存器地址映像

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	XFMC_BANK1_CR1	Reserved												BURSTWREN	PGSIZE			WAITASYNC	EXTEN	WAITEN	WREN	WCFG	WRAPEN	WAITDIR	BURSTEN	Reserved	ACCEN	MDBW		MTYPE		MUXEN	MBEN
	Reset Value													0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	1	0	1	1
004h	XFMC_BANK1_TR1	Reserved		ACCMODE		DATAHLD				CLKDIV				BUSRECOVERY				DATABLD								ADDHLD				ADDBLD			
	Reset Value			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
008h	XFMC_BANK1_CR2	Reserved												BURSTWREN	PGSIZE			WAITASYNC	EXTEN	WAITEN	WREN	WCFG	WRAPEN	WAITDIR	BURSTEN	Reserved	ACCEN	MDBW		MTYPE		MUXEN	MBEN
	Reset Value													0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	0	1	0
00Ch	XFMC_BANK1_TR2	Reserved		ACCMODE		DATAHLD				CLKDIV				BUSRECOVERY				DATABLD								ADDHLD				ADDBLD			
	Reset Value			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
060h	XFMC_CTRL2	Reserved												ECCPGS				ARDLY				CRDLY				Reserved	ECCEN	BUSWID		MEMTYPE	BANKEN	WAITEN	Reserved
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0		0	0	1	1	0	0	
064h	XFMC_STS2	Reserved																									FIFOEMPT	Reserved					
	Reset Value																										1						
068h	XFMC_CMEMTM2	HIZx								HLDx								WAITx								SETx							
	Reset Value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
06Ch	XFMC_ATTMEMTM2	HIZx								HLDx								WAITx								SETx							
	Reset Value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
074h	XFMC_ECC2	ECC2																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
080h	XFMC_CTRL3	Reserved												ECCPGS				ARDLY				CRDLY				Reserved	ECCEN	BUSWID		MEMTYPE	BANKEN	WAITEN	Reserved
	Reset Value													0	0	0	0	0	0	0	0	0	0	0	0		0	0	1	1	0	0	
084h	XFMC_STS3	Reserved																									FIFOEMPT	Reserved					
	Reset Value																										1						
088h	XFMC_CMEMTM3	HIZx								HLDx								WAITx								SETx							
	Reset Value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
08Ch	XFMC_ATTMEMTM3	HIZx								HLDx								WAITx								SETx							
	Reset Value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
094h	XFMC_ECC3	ECC3																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
104h	XFMC_BANK1_WTR1	Reserved		ACCMODE		DATAHLD				CLKDIV				Reserved				DATABLD								ADDHLD				ADDBLD			
	Reset Value			0	0	1	1	1	1	1	1	1	1					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10Ch	XFMC_BANK1_WTR2	Reserved		ACCMODE		DATAHLD				CLKDIV				Reserved				DATABLD								ADDHLD				ADDBLD			
	Reset Value			0	0	1	1	1	1	1	1	1	1					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

20.7.2 NOR 闪存和 PSRAM 控制器寄存器

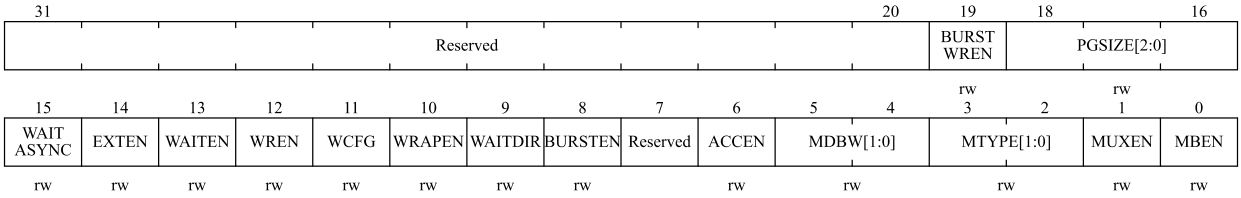
必须以字（32 位）的方式操作这些外设寄存器。

20.7.2.1 SRAM/NOR 闪存片选控制寄存器 1/2 （XFMC_BANK1_CR1/2）

地址偏移：0xA000 0000 + 8 * (x-1), x=1…2

复位值：0x0000 30DX

这个寄存器包含了每个存储器块的控制信息，可以用于 SRAM、 ROM、异步或成组传输的 NOR 闪存存储器。



位域	名称	描述
31:20	Reserved	保留，硬件强制为 0。
19	BURSTWREN	成组写使能位（Write burst enable） 对于 Cellular RAM，该位使能写操作的同步成组传输协议。 对于处于成组传输模式的闪存存储器，这一位允许/禁止通过 NWAIT 信号插入等待状态。 读操作的同步成组传输协议使能位是 XMC_BANK1_CRx 寄存器的 BURSTEN 位。 0：写操作始终处于异步模式 1：写操作为同步模式
18:16	PGSIZE[2:0]	CRAM 页大小（CRAM page size） 用于 Cellular RAM 1.5， 不允许跨页地址边界的成组访问。当配置这些位时，只要达到存 储器页大小时， XFMC 控制器会自动拆分成组访问（参考存储器参考手册的页大小） 。 000：当跨页地址边界时不会拆分成组访问，这是复位后的默认状态。 001：128 字节 010：256 字节 011：512 字节 100：1024 字节 其他：保留
15	WAITASYNC	在异步传输期间等待信号（Wait signal during asynchronous transfers） 此位使能/禁止 XFMC 去使用等待信号，即使在异步协议期间。 0：当运行异步协议时，不考虑 NWAIT 信号，这是复位后的默认状态。 1：在运行异步协议时，考虑 NWAIT 信号。
14	EXTEN	扩展模式使能（Extended mode enable） 该位允许 XFMC 使用 XFMC_BANK1_WTR 寄存器，即允许读和写使用不同的时序。 0：不使用 XFMC_BANK1_WTR 寄存器，这是复位后的默认状态。 1：XFMC 使用 XFMC_BANK1_WTR 寄存器。
13	WAITEN	等待使能位（Wait enable bit） 当闪存存储器处于成组传输模式时，这一位允许/禁止通过 NWAIT 信号插入等待状态。 0：禁用 NWAIT 信号，在设置的闪存保持周期之后不会检测 NWAIT 信号插入等待状态。 1：使用 NWAIT 信号，在设置的闪存保持周期之后根据 NWAIT 信号插入等待状态；这是复位后的默认状态。
12	WREN	写使能位（Write enable bit） 该位指示 XFMC 是否允许/禁止对存储器的写操作。

位域	名称	描述
		0：禁止 XFMC 对存储器的写操作，否则产生一个 AHB 错误。 1：允许 XFMC 对存储器的写操作；这是复位后的默认状态。
11	WCFG	配置等待时序（Wait timing configuration） 当闪存存储器处于成组传输模式时， NWAIT 信号指示从闪存存储器出来的数据是否有效或是否需要插入等待周期。该位决定存储器是在等待状态之前的一个时钟周期产生 NWAIT 信号，还是在等待状态期间产生 NWAIT 信号。 0：NWAIT 信号在等待状态前的一个数据周期有效；这是复位后的默认状态。 1：NWAIT 信号在等待状态期间有效（不适用于 Cellular RAM）。
10	WRAPEN	支持非对齐的成组模式（Wrapped burst mode support） 该位决定控制器是否支持把非对齐的 AHB 成组操作分割成 2 次线性操作；该位仅在存储器的成组模式下有效。 0：不允许直接的非对齐成组操作；这是复位后的默认状态。 1：允许直接的非对齐成组操作。
9	WAITDIR	等待信号极性 设置存储器产生的等待信号的极性；该位仅在存储器的成组模式下有效。 0：NWAIT 等待信号为低时有效；这是复位后的默认状态。 1：NWAIT 等待信号为高时有效
8	BURSTEN	成组模式使能（Burst enable bit） 允许对闪存存储器进行成组模式访问；该位仅在闪存存储器的同步成组模式下有效。 0：禁用成组访问模式；这是复位后的默认状态。 1：使用成组访问模式。
7	Reserved	保留，硬件强制为 0。
6	ACCEN	闪存访问使能（Flash access enable） 允许对 NOR 闪存存储器的访问操作。 0：禁止对 NOR 闪存存储器的访问操作. 1：允许对 NOR 闪存存储器的访问操作。
5:4	MDBW[1:0]	存储器数据总线宽度（Memory databus width） 定义外部存储器总线的宽度，适用于所有类型的存储器。 00：8 位 01：16 位（复位后的默认状态） 10：保留，不能用 11：保留，不能用
3:2	MTYPE[1:0]	存储器类型（Memory type） 定义外部存储器的类型: 00：SRAM、 ROM 01：PSRAM（Cellular RAM: CRAM） 10：NOR 闪存 11：保留
1	MUXEN	地址/数据复用使能位（Address/data multiplexing enable bit） 当设置了该位后，地址的低 16 位和数据将共用数据总线，该位仅对 NOR 和 PSRM 存储器有效。 0：地址/数据不复用。

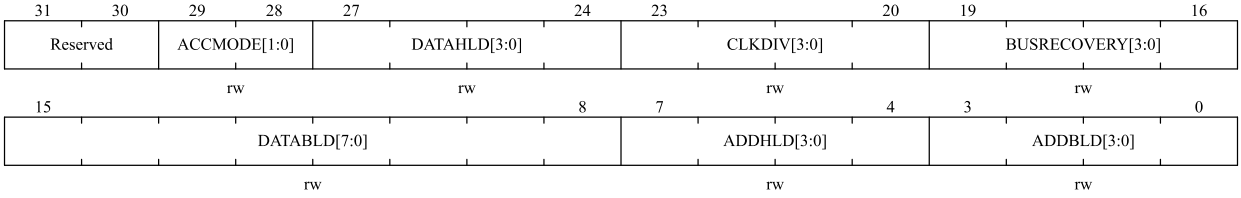
位域	名称	描述
		1：地址/数据复用数据总线；这是复位后的默认状态。
0	MBEN	存储器块使能位（Memory bank enable bit） 开启对应的存储器块。复位后存储器块 1 是开启的，其它所有存储器块为禁用。访问一个禁用的存储器块将在 AHB 总线上产生一个错误。 0：禁用对应的存储器块。 1：启用对应的存储器块。

20.7.2.2 SRAM/NOR 闪存片选时序寄存器 1/2 （XFMC_BANK1_TR1/2）

地址偏移： 0xA000 0000 + 0x04 + 8 * (x-1), x=1…2

复位值： 0x0FFF FFFF

这个寄存器包含了每个存储器块的控制信息，可以用于 SRAM、ROM 和 NOR 闪存存储器。如果 XFMC_BANK1_CRx 寄存器中设置了 EXTEN 位，则有两个时序寄存器分别对应读（本寄存器）和写操作（XFMC_BANK1__WTRx 寄存器）。



位域	名称	描述
31:30	Reserved	保留，硬件强制为 0。
29:28	ACCMODE[1:0]	访问模式（Access mode） 定义异步访问模式。这 2 位只在 XFMC_BANK1_CRx 寄存器的 EXTEN 位为 1 时起作用。 00：访问模式 A 01：访问模式 B 10：访问模式 C 11：访问模式 D
27:24	DATAHLD[3:0]	（见本表格下面的注释）： （同步成组式 NOR 闪存的）数据保持时间 （Data latency（for synchronous burst NOR Flash）） 处于同步成组模式的 NOR 闪存，需要定义在读取第一个数据之前等待的存储器周期数目。 这个时间参数不是以 HCLK 表示，而是以闪存时钟（CLK）表示。在访问异步 NOR 闪存、 SRAM 或 ROM 时，这个参数不起作用。操作 CRAM 时，这个参数必须为 0。 0000：第一个数据的保持时间为 2 个 CLK 时钟周期 …… 1111：第一个数据的保持时间为 17 个 CLK 时钟周期（这是复位后的默认数值）。
23:20	CLKDIV[3:0]	时钟分频比（CLK 信号）（Clock divide ratio （for CLK signal））

位域	名称	描述
		定义 CLK 时钟输出信号的周期，以 HCLK 周期数表示： 0000：保留 0001： 1 个 CLK 周期=2 个 HCLK 周期 0010： 1 个 CLK 周期=3 个 HCLK 周期 …… 1111： 1 个 CLK 周期=16 个 HCLK 周期（这是复位后的默认数值）。 在访问异步 NOR 闪存、 SRAM 或 ROM 时，这个参数不起作用。
19:16	BUSRECOVERY[3:0]	总线恢复时间（Bus turnaround phase duration） 这些位用于定义一次读操作之后在总线上的延迟（仅适用于总线复用模式的 NOR 闪存操作），一次读操作之后控制器需要在数据总线上为下次操作送出地址，这个延迟就是为了防止总线冲突。如果扩展的存储器系统不包含总线复用模式的存储器，或最慢的存储器可以在 6 个 HCLK 时钟周期内将数据总线恢复到高阻状态，可以设置这个参数为其最小值。 0000：总线恢复时间=1 个 HCLK 时钟周期 …… 1111：总线恢复时间=16 个 HCLK 时钟周期（这是复位后的默认数值）。
15:8	DATABLD[7:0]	数据保持时间（Data-phase duration） 这些位定义数据的保持时间，适用于 SRAM、 ROM 和异步总线复用模式的 NOR 闪存操作。 0000 0000：保留 0000 0001： DATABLD 保持时间=2 个 HCLK 时钟周期 0000 0010： DATABLD 保持时间=3 个 HCLK 时钟周期 …… 1111 1111： DATABLD 保持时间=256 个 HCLK 时钟周期（这是复位后的默认数值）。 对于每一种存储器类型和访问方式的数据保持时间，请参考对应的图表。 例如：模式 1、读操作、 DATABLD=1：数据保持时间=DATABLD+3=4 个 HCLK 时钟周期。
7:4	ADDHLD[3:0]	地址保持时间（Address-hold phase duration） 这些位定义地址的保持时间，适用于 SRAM、ROM 和异步总线复用模式的 NOR 闪存操作。 0000： ADDHLD 保持时间=1 个 HCLK 时钟周期 …… 1111： ADDHLD 保持时间=16 个 HCLK 时钟周期（这是复位后的默认数值）。 *注：在同步操作中，这个参数不起作用，地址保持时间始终是 1 个存储器时钟周期。*
3:0	ADDBLD[3:0]	地址建立时间（Address setup phase duration） 这些位定义地址的建立时间，适用于 SRAM、ROM 和异步总线复用模式的 NOR 闪存操作。 0000： ADDBLD 建立时间=1 个 HCLK 时钟周期 …… 1111： ADDBLD 建立时间=16 个 HCLK 时钟周期（这是复位后的默认数值）。

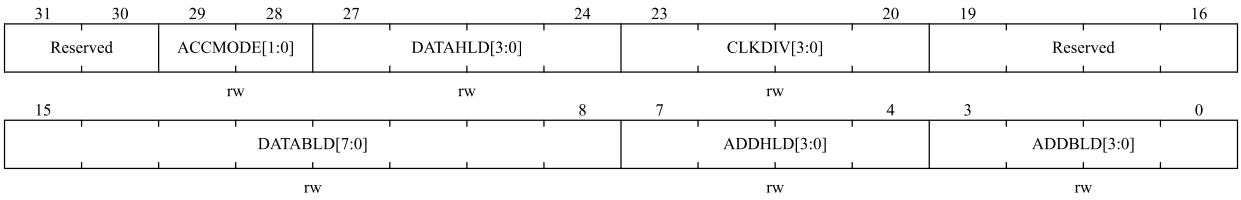
位域	名称	描述
		对于每一种存储器类型和访问方式的地址建立时间，请参考对应的图表。 例如：模式 2、读操作、 ADDBLD=1：地址建立时间=ADDBLD+1=2 个 HCLK 时钟周期 *注：在同步操作中，这个参数不起作用，地址建立时间始终是 1 个存储器时钟周期。*

注： 因为内部的刷新， PSRAM（CRAM）具有可变的保持延迟，因此这样的存储器会在数据保持期间输出 NWAIT 信号以延长数据的保持时间。
使用 PSRAM（CRAM）时 DATAHLD 域应置为 0，这样 XFMC 可以及时地退出自己的保持阶段并开始对存储器发出的 NWAIT 信号进行采样，然后在存储器准备好时开始读或写操作。
这个操作方式还可以用于操作最新的能够输出 NWAIT 信号的同步闪存存储器，详细信息请参考相应的闪存存储器手册。

20.7.2.3 SRAM/NOR 闪存写时序寄存器 1/2 （XFMC_BANK1_WTR1/2）

地址偏移：0xA000 0000 + 0x104 + 8 * (x-1), x=1…2
复位值： 0x0FFF FFFF

这个寄存器包含了每个存储器块的控制信息，可以用于 SRAM、 ROM 和 NOR 闪存存储器。如果 XFMC_BANK1_CRx 寄存器中设置了 EXTEN 位，则这个寄存器对应写操作。



位域	名称	描述
31:30	Reserved	保留，硬件强制为 0。
29:28	ACCMODE[1:0]	访问模式（Access mode） 定义异步访问模式。这 2 位只在 XFMC_BANK1_CRx 寄存器的 EXTEN 位为 1 时起作用。 00：访问模式 A 01：访问模式 B 10：访问模式 C 11：访问模式 D
27:24	DATAHLD[3:0]	（NOR 闪存的同步成组式）数据保持时间（Data latency （for synchronous burst NOR Flash）） 处于同步成组模式的 NOR 闪存，需要定义在读取第一个数据之前等待的存储器周期数目。 这个时间参数不是以 HCLK 表示，而是以闪存时钟（CLK）表示。在访问异步 NOR 闪存、 SRAM 或 ROM 时，这个参数不起作用。操作 CRAM 时，这个参数必须为 0。 0000：第一个数据的保持时间为 2 个 CLK 时钟周期 …… 1111：第一个数据的保持时间为 17 个 CLK 时钟周期（这是复位后的默认数值）。

位域	名称	描述
23:20	CLKDIV[3:0]	时钟分频比（CLK 信号）（Clock divide ratio（for CLK signal）） 定义 CLK 时钟输出信号的周期，以 HCLK 周期数表示： 0000：保留 0001： 1 个 CLK 周期=2 个 HCLK 周期 0010： 1 个 CLK 周期=3 个 HCLK 周期 …… 1111： 1 个 CLK 周期=16 个 HCLK 周期（这是复位后的默认数值）。 在访问异步 NOR 闪存、 SRAM 或 ROM 时，这个参数不起作用。
19:16	Reserved	保留，硬件强制为 0。
15:8	DATABLD[7:0]	数据保持时间（Data-phase duration） 这些位定义数据的保持时间，适用于 SRAM、 ROM 和异步总线复用模式的 NOR 闪存操作。 0000 0000：保留 0000 0001： DATABLD 保持时间=2 个 HCLK 时钟周期 0000 0010： DATABLD 保持时间=3 个 HCLK 时钟周期 …… 1111 1111： DATABLD 保持时间=256 个 HCLK 时钟周期（这是复位后的默认数值）。 对于每一种存储器类型和访问方式的数据保持时间，请参考对应的图表。 例如：模式 1、读操作、 DATABLD=1：数据保持时间=DATABLD+3=4 个 HCLK 时钟周期。
7:4	ADDHLD[3:0]	地址保持时间（Address-hold phase duration） 这些位定义地址的保持时间，适用于 SRAM、ROM 和异步总线复用模式的 NOR 闪存操作。 0000： ADDHLD 保持时间=1 个 HCLK 时钟周期 …… 1111： ADDHLD 保持时间=16 个 HCLK 时钟周期（这是复位后的默认数值）。 *注：在同步操作中，这个参数不起作用，地址保持时间始终是 1 个存储器时钟周期。*
3:0	ADDBLD[3:0]	地址建立时间（Address setup phase duration） 这些位定义地址的建立时间，适用于 SRAM、ROM 和异步总线复用模式的 NOR 闪存操作。 0000： ADDBLD 建立时间=1 个 HCLK 时钟周期 …… 1111： ADDBLD 建立时间=16 个 HCLK 时钟周期（这是复位后的默认数值）。 对于每一种存储器类型和访问方式的地址建立时间，请参考对应的图表。 例如：模式 2、读操作、 ADDBLD=1：地址建立时间=ADDBLD+1=2 个 HCLK 时钟周期 *注：在同步操作中，这个参数不起作用，地址建立时间始终是 1 个存储器时钟周期。*

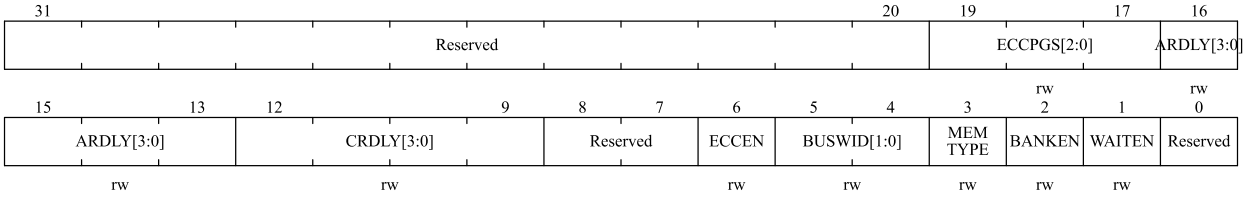
20.7.3 NAND 闪存控制器寄存器

必须以字（32 位）的方式操作这些外设寄存器

20.7.3.1 NAND 闪存控制寄存器 2/3 （XFMC_CTRL2/3）

地址偏移：0xA000 0000 + 0x40 + 0x20 * (x-1), x=2..3

复位值：0x0000 0018



位域	名称	描述
31:20	Reserved	保留，硬件强制为 0。
19:17	ECCPGS[2:0]	ECC 页面大小（ECC page size） 定义扩展的 ECC 页面大小： 000：256 字节； 001：512 字节； 010：1024 字节； 011：2048 字节； 100：4096 字节； 101：8192 字节。
16:13	ARDLY[3:0]	ALE 至 RE 的延迟（ALE to RE delay） 以 AHB 时钟周期（HCLK）为单位设置从 ALE 变低至 RE 变低的时间。 时间计算： t_ar = (ARDLY + SET + 4) x THCLK，这里 THCLK 表示 HCLK 周期长度 0000：1 个 HCLK 周期（默认值）； …… 1111：16 个 HCLK 周期。 *注：根据不同的地址空间，SET 是 CMEMTM SET 或是 ATTMEMTM SET。*
12:9	CRDLY[3:0]	CLE 至 RE 的延迟（CLE to RE delay） 以 AHB 时钟周期（HCLK）为单位设置从 CLE 变低至 RE 变低的时间。 时间计算：t_clr = (CRDLY + SET + 4) x THCLK，这里 THCLK 表示 HCLK 周期长度 0000：1 个 HCLK 周期（默认值）； …… 1111：16 个 HCLK 周期。 *注：根据不同的地址空间，SET 是 CMEMTM SET 或是 ATTMEMTM SET。*
8:7	Reserved	保留，硬件强制为 0。
6	ECCEN	ECC 计算电路使能位（ECC computation logic enable bit） 0：关闭并复位 ECC 电路（复位后的默认值）； 1：使能 ECC 电路。
5:4	BUSWID[1:0]	数据总线宽度（Databus width）

位域	名称	描述
		定义外部 NAND 闪存数据总线的宽度。 00：8 位（复位后的默认值）； 01：16 位； 10：保留，不要使用； 11：保留，不要使用。
3	MEMTYPE	存储器类型（Memory type） 定义对应的存储器块上连接的存储器类型。 0：PC 卡、CF 卡、CF+卡或 PCMCIA（不支持）； 1：NAND 闪存（复位后的默认值）。
2	BANKEN	NAND 存储器块使能位（NAND Flash memory bank enable bit） 使能存储器块。访问一个未使能的存储器块会产生一个 AHB 总线错误。 0：关闭对应的存储器块（复位后的默认值）； 1：使能对应的存储器块
1	WAITEN	等待功能使能位（Wait feature enable bit） 使能 NAND 闪存存储器块的等待功能 0：关闭（复位后的默认值）； 1：使能。
0	Reserved	保留，硬件强制为 0。

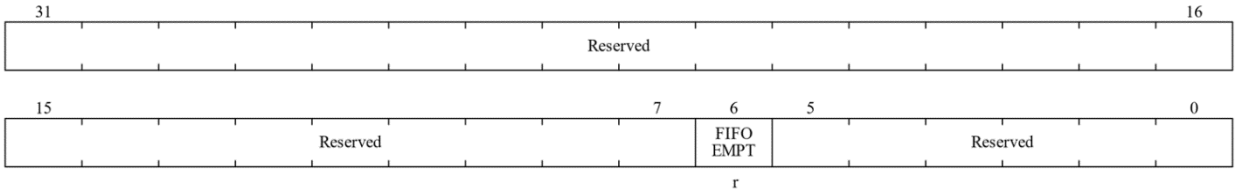
20.7.3.2 FIFO 状态和中断寄存器 2/3（XFMC_STS2/3）

地址偏移：0xA000 0000 + 0x44 + 0x20 * (x-1), x=2..3

复位值：0x0000 0040

XFMC 不支持中断功能。

为了计算 ECC 的需要，该寄存器有一个指示位反映了 FIFO 的状态。数据写到存储器时同时进行 ECC 计算，因此软件必须等待 FIFO 变空后才能读到正确的 ECC 数值。



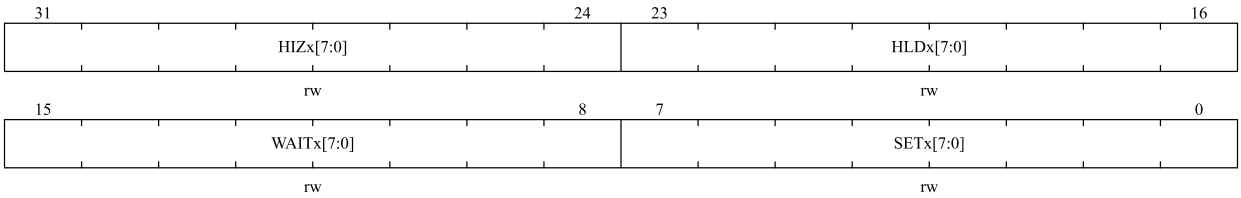
位域	名称	描述
31:7	Reserved	保留，硬件强制为 0。
6	FIFOEMPT	FIFO 空标志（FIFO empty） 只读位，指示 FIFO 状态 0：FIFO 不空； 1：FIFO 空
5-0	Reserved	保留，硬件强制为 0。

20.7.3.3 通用存储空间时序寄存器 2/3 （XFMC_CMEMTM2/3）

地址偏移： 0xA000 0000 + 0x48 + 0x20 * (x-1), x=2..3

复位值： 0xFFFF FFFF

每个 CMEMTMx（x=2..3）寄存器都包含操作 NAND 闪存存储块 x 的时序参数，这些参数适用于在通用存储空间发送 NAND 闪存的命令、地址和进行数据的读写操作



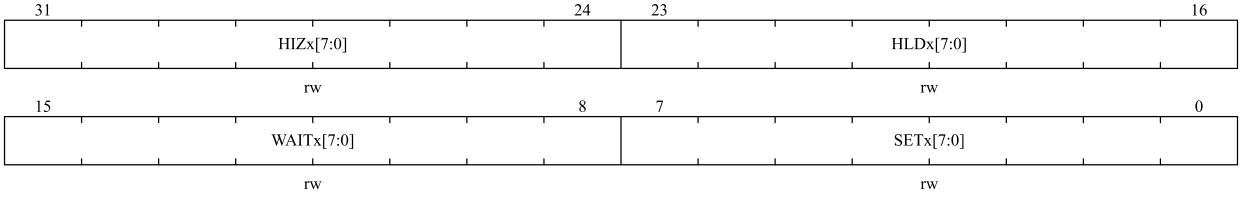
位域	名称	描述
31:24	HIZx[7:0]	在通用空间 x 数据总线的高阻时间（Common memory x databus Hiz time） 当在通用存储空间 x 开始执行对 NAND 闪存的写操作后，数据总线需要保持一段时间的高阻状态，该参数以 HCLK 时钟周期数目（NAND 类型时+1）定义数据总线高阻态的时间。这个参数仅对写操作有效。 0000 0000：（0x00）NAND 闪存为 1 个 HCLK 周期； …… 1111 1111：（0xFF）NAND 闪存为 256 个 HCLK 周期，这是复位后的默认值。
23:16	HLDx[7:0]	在通用空间 x 的保持时间（Common memory x hold time） 当在通用存储空间 x 对 NAND 闪存进行读或写操作时，该参数以 HCLK 时钟周期数目定义了发送命令（NWE、 NOE 变高）后，地址信号（对于写操作则是数据信号）保持的时间。 0000 0000：保留； 0000 0001：1 个 HCLK 周期； …… 1111 1111：255 个 HCLK 周期（复位后的默认值）。
15:8	WAITx[7:0]	在通用空间 x 的等待时间（Common memory x wait time） 当在通用存储空间 x 对 NAND 闪存进行读或写操作时，该参数以 HCLK（+1）时钟周期数目定义了保持命令（NWE、 NOE 为低）的最小时间。当该参数定义的时间结束时，如果等待信号（NWAIT）有效（低），则命令的保持时间会被拉长。 0000 0000：保留； 0000 0001：2 个 HCLK 周期（加上由 NWAIT 信号变低引入的等待周期）； …… 1111 1111：256 个 HCLK 周期（加上由卡的 NWAIT 信号变低引入的等待周期），这是复位后的默认值。
7:0	SETx[7:0]	在通用空间 x 的建立时间（Common memory x setup time） 当在通用存储空间 x 对 NAND 闪存进行读或写操作时，该参数以 HCLK（操作 NAND 闪存时+2）时钟周期数目定义了发送命令（NWE、 NOE 变低）之前建立地址信号的时间。 0000 0000：NAND 闪存为 2 个 HCLK 周期； …… 1111 1111：NAND 闪存为 257 个 HCLK 周期，这是复位后的默认值。

20.7.3.4 属性存储空间时序寄存器 2/3 （XFMC_ATTMEMTM2/3）

地址偏移：0xA000 0000 + 0x4C + 0x20 * (x-1)，x=2..3

复位值：0xFFFF FFFF

每个 ATTMEMTMx（x=2..3）读/写寄存器都包含操作 NAND 闪存存储块 x 的时序参数，这些参数适用于在 NAND 闪存的最后一个地址写操作的时序与其它操作不同的时候（关于就绪/繁忙的管理，参见 20.6.5 节）。



位域	名称	描述
31:24	HIZx[7:0]	在属性空间 x 数据总线的高阻时间（Attribute memory x databus Hiz time） 当在属性存储空间 x 开始执行对 NAND 闪存的写操作后，数据总线需要保持一段时间的高阻状态，该参数以 HCLK 时钟周期数目定义数据总线高阻态的时间。这个参数仅对写操作有效。 0000 0000：0 个 HCLK 周期； …… 1111 1111：255 个 HCLK 周期（复位后的默认值）。
23:16	HLDx[7:0]	在属性空间 x 的保持时间（Attribute memory x hold time） 当在属性存储空间 x 对 NAND 闪存进行读或写操作时，该参数以 HCLK 时钟周期数目定义了发送命令（NWE、 NOE 变高）后，地址信号（对于写操作则是数据信号）保持的时间。 0000 0000：保留； 0000 0001：1 个 HCLK 周期； …… 1111 1111：255 个 HCLK 周期（复位后的默认值）。
15:8	WAITx[7:0]	在属性空间 x 的等待时间（Attribute memory x wait time） 当在属性存储空间 x 对 NAND 闪存进行读或写操作时，该参数以 HCLK（+1）时钟周期数目定义了保持命令（NWE、 NOE 为低）的最小时间。当该参数定义的时间结束时如果等待信号（NWAIT）有效（低），则命令的保持时间会被拉长。 0000 0000：1 个 HCLK 周期（加上由 NWAIT 信号变低引入的等待周期）； …… 1111 1111：256 个 HCLK 周期（加上由卡的 NWAIT 信号变低引入的等待周期）（复位后的默认值）。
7:0	SETx[7:0]	在属性空间 x 的建立时间（Attribute memory x setup time） 当在属性存储空间 x 对 NAND 闪存进行读或写操作时，该参数以 HCLK（+1）时钟周期数目定义了发送命令（NWE、 NOE 变低）之前建立地址信号的时间。 0000 0000：1 个 HCLK 周期； …… 1111 1111：256 个 HCLK 周期（复位后的默认值）。

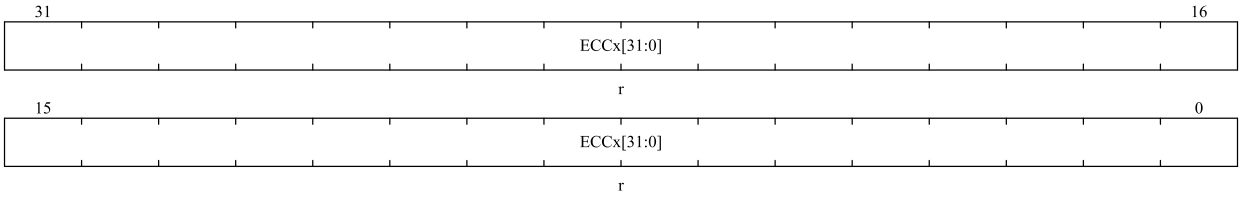
20.7.3.5 ECC 结果寄存器 2/3 （XFMC_ECC2/3）

地址偏移：0xA000 0000 + 0x54 + 0x20 * (x-1), x=2 或 3

复位值：0x0000 0000

这 2 个寄存器包含了由 XFMC 控制器的 ECC 计算模块得到的纠错码的当前数值，每个 NAND 闪存存储器块有一个 ECC 计算模块。当 CPU 在正确的地址（见 20.6.6 节）读/写 NAND 闪存的数据时，ECC

模块会自动地处理写入或读出的数据。根据 XFMC_CTRLx 中 ECCPGS 域的设置，在读出了每页的最后一个字节后，CPU 必须读出 XFMC_ECCx 寄存器中的 ECC 数值，并与记录在 NAND 闪存后备区域的数据进行比较，据此判断该页的数据是否正确并在可能的情况下，实行矫正。在读出 XFMC_ECCx 寄存器的数值后应设置 ECCEN 位为‘0’清除它的内容。需要计算一个新的数据页时，再次设置 ECCEN 为‘1’。



位域	名称	描述
31:0	ECCx[31:0]	ECC 结果 ECC 计算电路产生的计算结果。下表显示了这些位的内容。

表 20-36 ECC 结果有效位

ECCPGS[2:0]	页大小（字节）	ECC 有效位
000	256	ECC[21:0]
001	512	ECC[23:0]
010	1024	ECC[25:0]
011	2048	ECC[27:0]
100	4096	ECC[29:0]
101	8192	ECC[31:0]

21 控制器局域网（bxCAN）

21.1 bxCAN 简介

bxCAN 指的是基本扩展 CAN（Basic Extended CAN），它支持 CAN 协议 2.0A 和 2.0B。它在高效处理大量收到的报文的同时，极大地降低了 CPU 资源的消耗。可以通过软件配置报文发送的优先级特性。

对于一些安全要求较高的应用，bxCAN 的硬件功能支持时间触发通信模式。

21.2 bxCAN 主要特点

- 支持 CAN 协议 2.0A 和 2.0B 主动模式
- 波特率最高支持 1Mbit/s
- 支持时间触发通信功能

发送

- 有 3 个发送邮箱
- 软件可配置发送报文的优先级特性
- 记录发送 SOF 时刻的时间戳功能

接收

- 有 2 个接收 FIFO，每个 FIFO 都具有 3 级深度
- 共 14 个过滤器组（每个 CAN 模块单独拥有）
- 标识符列表模式
- 可配置的 FIFO 溢出处理方式
- 记录接收 SOF 时刻的时间戳功能

时间触发通信模式

- 支持禁止自动重传模式
- 16 位自由运行定时器
- 可配置最后 2 个数据字节为发送时间戳

管理

- 中断可屏蔽
- 邮箱单独分配了 1 块地址空间，用于提高软件执行的效率

双 CAN

- CAN1：管理在从 bxCAN 和 512 字节的 SRAM 存储器之间的通信
- CAN2：管理在从 bxCAN 和 512 字节的 SRAM 存储器之间的通信

- 这 2 个 bxCAN 模块各自拥有 512 字节的 SRAM 存储器（见图 21-2）

21.3 bxCAN 总体描述

随着 CAN 的广泛应用，CAN 网络的节点在飞速增长，而且多个 CAN 经常通过网关连接起来，因此整个 CAN 网中有大量的报文需要处理。整个 CAN 网中除了有应用层报文外，还充斥着大量网络管理和诊断的报文。

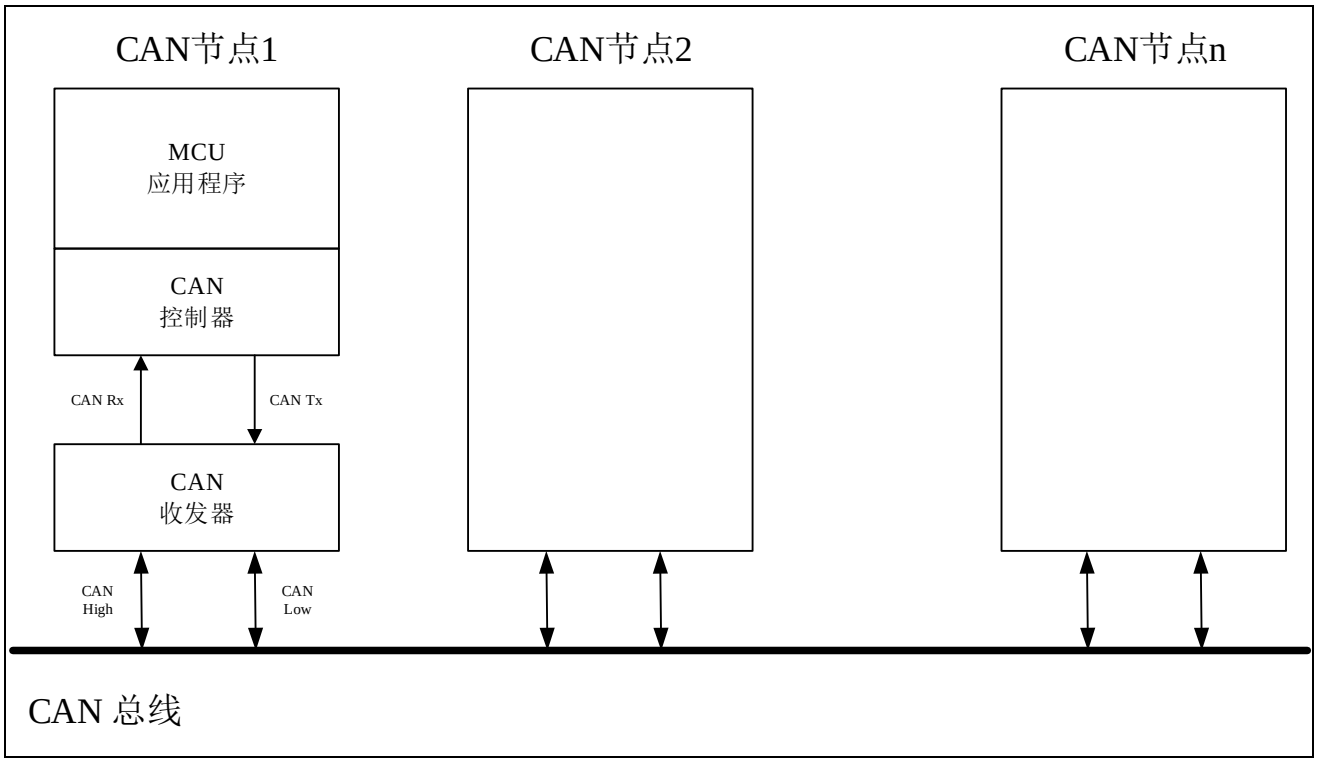
- 需要一个加强的报文过滤机制来处理不同类型的报文。

  而且处理应用层的任务需要占用大量的 CPU 时间，因此需要减少接收报文所需的实时响应时间。

- 接收 FIFO 的存在，让 CPU 即使花大量时间处理应用层的任务也不会丢失报文。

  应用程序要基于底层的 CAN 驱动程序，而且要提高与 CAN 控制器之间交互效率。

图 21-1 CAN 网拓扑结构



21.3.1 CAN 2.0B 主动内核

bxCAN 模块可以实现对 CAN 报文的自动接收和发送，支持标准标识符（11 位）和扩展标识符（29 位）。

21.3.2 控制、状态和配置寄存器

应用程序通过配置这些寄存器，可以实现：

- 配置 CAN 参数，如工作模式、波特率等；

- 开始报文发送;
- 处理报文接收;
- 配置中断;
- 读取诊断信息;

21.3.3 发送邮箱

应用程序可以通过 3 个发送邮箱来发送报文。3 个邮箱报文发送的先后是发送调度器根据报文优先级决定，而优先级可以选择由报文的标识符决定或者由发送请求的顺序决定。

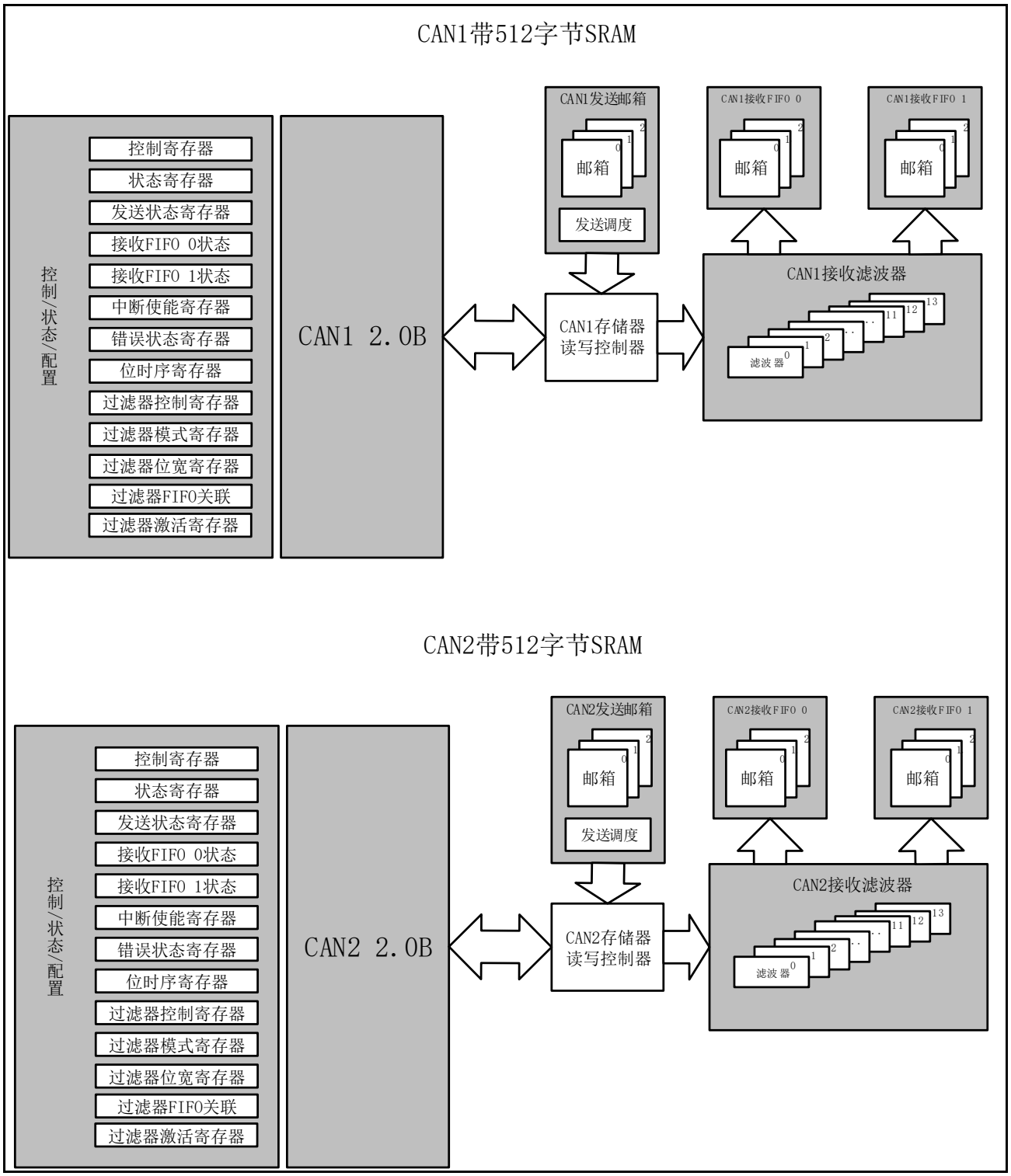
21.3.4 接收过滤器

每个 bxCAN 都有 14 个位宽可变/可配置的标识符过滤器组。应用程序配置标识符过滤器组后，接收邮箱会自动接收需要的报文，丢弃其它报文。

接收 FIFO

每个 bxCAN 都有 2 个接收 FIFO，每个 FIFO 都可以存放 3 个完整的报文。不需要应用程序来管理，完全由硬件来管理。

图 21-2 双 CAN 框图



21.4 bxCAN 工作模式

bxCAN 主要有 3 个工作模式：**初始化**、**正常**和**睡眠模式**。在硬件复位后，激活 CANTX 引脚的内部

上拉电阻，bxCAN 工作在睡眠模式来降低功耗。软件将 CAN_MCTRL 寄存器的 INIRQ 或 SLPRQ 位置'1'，可以配置 bxCAN 进入**初始化**或**睡眠模式**。软件读取 CAN_MSTS 寄存器的 INIAK 或 SLPAK 位置'1'来确认是否进入了**初始化**或**睡眠模式**，同时 CANTX 引脚内部上拉电阻被禁用。当 INIAK 和 SLPAK 位都为'0'时，bxCAN 则处于**正常模式**。bxCAN 必须跟 CAN 总线取得**同步**才能进入**正常模式**；当在 CANRX 引脚上监测到 11 个连续的隐性位时，CAN 总线为空闲状态，同步完成。

21.4.1 初始化模式

bxCAN 必须处于初始化模式时软件才能执行初始化配置。将 CAN_MCTRL 寄存器的 INIRQ 位置'1'，等待硬件对 CAN_MSTS 寄存器的 INIAK 位置'1'来确认 bxCAN 进入初始化模式。将 CAN_MCTRL 寄存器的 INIRQ 位置'0'， 等待硬件对 CAN_MSTS 寄存器的 INIAK 位置'0'来确认 bxCAN 退出初始化模式。

进入初始化模式时，不会影响寄存器的配置。当 bxCAN 处于初始化模式时，报文接收和发送被禁止，并且 CANTX 引脚输出隐性位（高电平）。

软件对 bxCAN 执行初始化配置，至少需要配置位时间特性寄存器（CAN_BTIM）和控制寄存器（CAN_MCTRL）。软件需要将 CAN_FMC 寄存器的 FINITM 位置'1'，才能初始化配置 bxCAN 的过滤器组（模式、位宽、FIFO 关联、激活和过滤器值）。初始化配置 bxCAN 的过滤器组不一定需要在初始化模式下。

注：当 FINITM=1 时，禁止接收报文。如果要修改相应过滤器的值，需要先对过滤器激活位清'0'（在 CAN_FA1 中）。需要将未使用的过滤器组保持非激活状态（保持其 FAC 位为清'0'状态）。

21.4.2 正常模式

在初始化完成后，软件配置 bxCAN 进入正常模式。将 CAN_MCTRL 寄存器的 INIRQ 位清'0'，等待硬件对 CAN_MSTS 寄存器的 INIRQ 位清'0'来确认 bxCAN 进入正常模式。在和 CAN 总线完成同步后，bxCAN 才能正常接收和发送报文。

必须在过滤器处于初始化模式下（CAN_FMC 寄存器的 FINITM 位为 1）完成过滤器组的位宽和模式的设置。而过滤器初值的设置必须在它处在非激活状态下完成（CAN_FA1 寄存器相应的 FAC 位为 0）。

21.4.3 睡眠模式（低功耗）

bxCAN 可工作在睡眠模式来降低功耗。将 CAN_MCTRL 寄存器的 SLPRQ 位置'1'，等待硬件对 CAN_MSTS 寄存器的 SLPAK 位置'1'来确认 bxCAN 进入睡眠模式。在睡眠模式下，bxCAN 的时钟停止工作，但软件仍然可以访问发送/接收邮箱寄存器。

当 bxCAN 处于睡眠模式，必须将 CAN_MCTRL 寄存器的 INIRQ 位置'1'并且同时对 SLPRQ 位清'0'，才能进入初始化模式。

有 2 种方式可以唤醒 bxCAN（bxCAN 退出睡眠模式）：通过将 CAN_MCTRL 寄存器的 SLPRQ 位清'0'或者硬件检测到 CAN 总线的活动。

当 CAN_MCTRL 寄存器的 AWKUM 位为'1'，即使能硬件自动唤醒时，一旦检测到 CAN 总线的活动，硬件就自动对 CAN_MSTS 寄存器的 SLPRQ 位清'0'来唤醒 bxCAN。

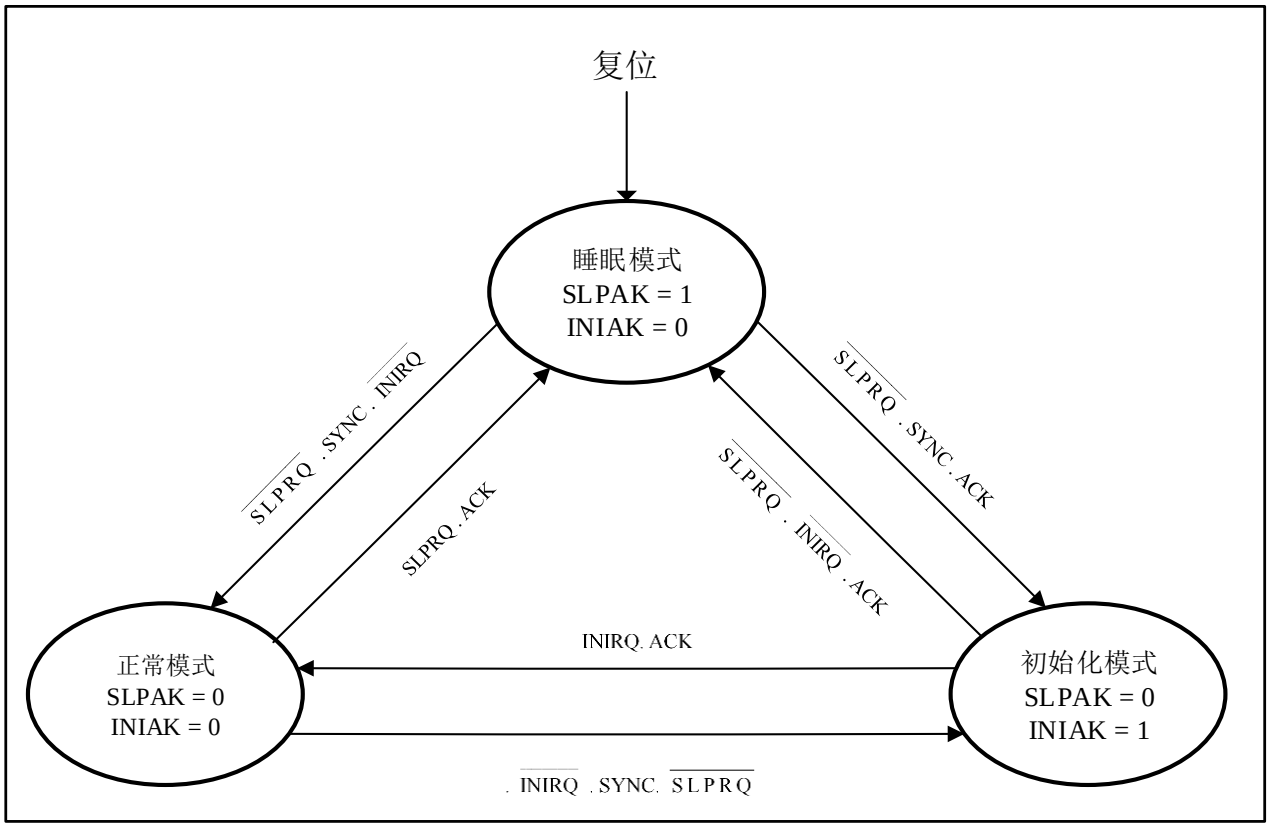
当 CAN_MCTRL 寄存器的 AWKUM 位为'0' ，即使能软件唤醒时，软件必须在唤醒中断处理函数里对 CAN_MSTS 寄存器的 SLPRQ 位清'0'才能退出睡眠状态。

注：如果使能唤醒中断（CAN_INTE 寄存器的 WKUITE 位为'1'），那么一旦检测到 CAN 总线活动就会产生唤醒中断，而不管是否使能硬件自动唤醒 bxCAN。

在对 CAN_MCTRL 寄存器的 SLPRQ 位清'0'后，bxCAN 必须与 CAN 总线同步后才能退出睡眠模式，请参考图 21-3：bxCAN 工作模式。

当 CAN_MSTS 寄存器的 SLPAK 位清'0'后，才能确认退出了睡眠模式。

图 21-3 bxCAN 工作模式



注：1.ACK = 硬件响应睡眠或初始化请求,而对 CAN_MSTS 寄存器的 INIAK 或 SLPAK 位置 1 的状态

2.SYNC = bxCAN 等待 CAN 总线变为空闲的状态，即在 CANRX 引脚上检测到连续的 11 个隐性位

21.5 测试模式

必须在初始化模式下，通过组合配置 CAN_BTIM 寄存器的 SLM、LBM 位来选择一种测试模式。在选择了一种测试模式后，软件需要对 CAN_MCTRL 寄存器的 INIRQ 位清'0'来退出初始化模式，并进入测试模式。

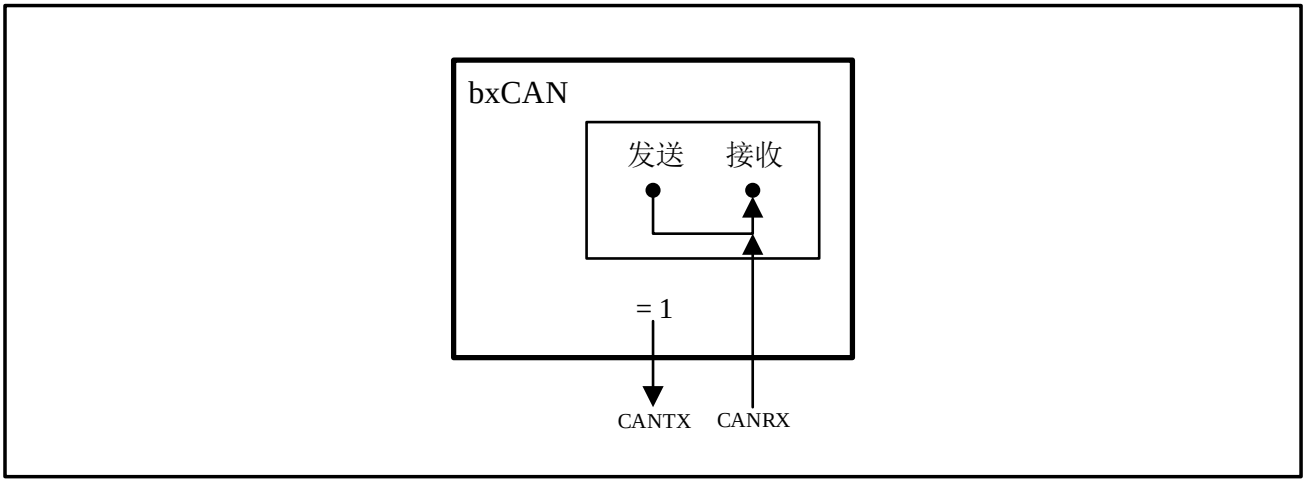
21.5.1 静默模式

配置 CAN_BTIM 寄存器的 SLM 位为'1'、LBM 位为'0'，则选择静默模式。

在静默模式下，bxCAN 可以正常地接收数据帧和远程帧，但只能发出隐性位，而不能真正发送报

文。如果 bxCAN 需要发出显性位（确认位、过载标志、主动错误标志），那么这样的显性位在内部被接回来从而可以被 CAN 内核检测到，同时 CAN 总线不会受到影响而仍然维持在隐性位状态。因此，静默模式通常用于分析 CAN 总线的活动，而不会对总线造成影响－显性位（确认位、错误帧）不会真正发送到总线上。

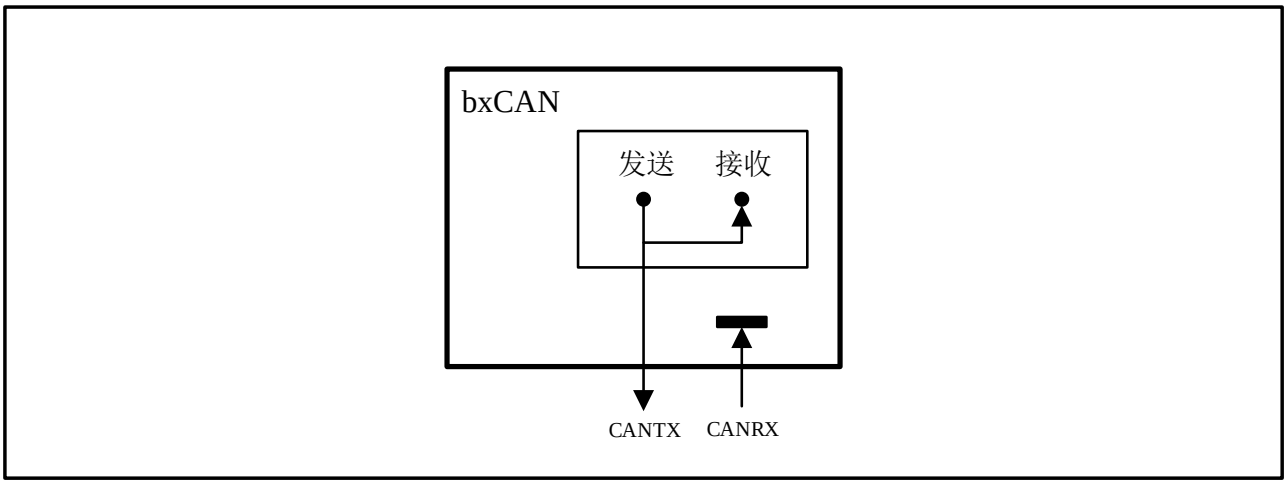
图 21-4 bxCAN 工作在静默模式



21.5.2 环回模式

配置 CAN_BTIM 寄存器的 SLM 位为'0'、LBM 位为'1'，则选择环回模式。在环回模式下，bxCAN 把发送的报文当作接收的报文并保存（如果可以通过接收过滤）在接收邮箱里。

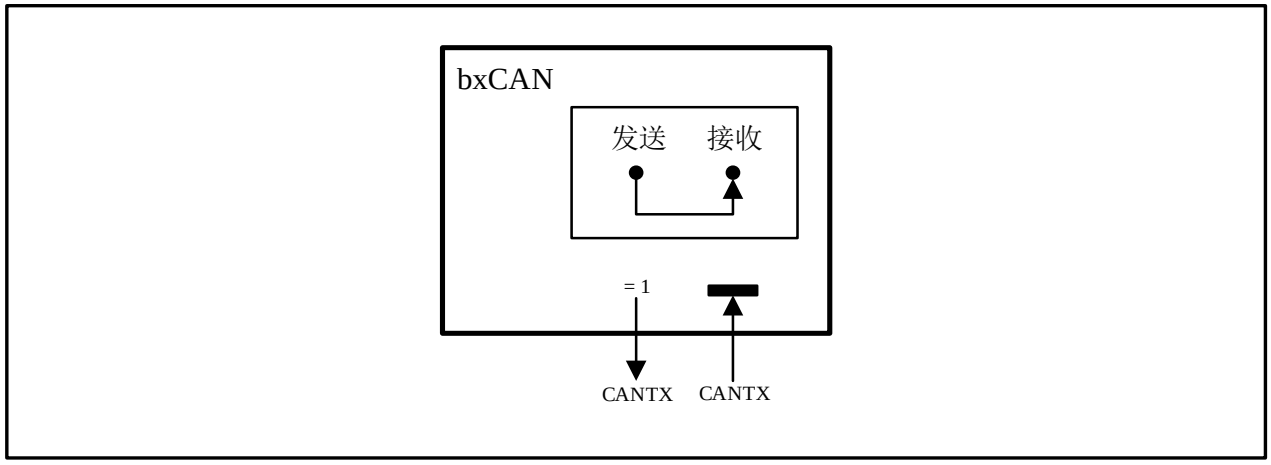
图 21-5 bxCAN 工作在环回模式



环回模式可用于自测试。为了避免外部的影响，在环回模式下 CAN 内核忽略确认错误（在数据/远程帧的确认位时刻，不检测是否有显性位）。在环回模式下，bxCAN 在内部把 Tx 输出回馈到 Rx 输入上，而完全忽略 CANRX 引脚的实际状态。发送的报文可以在 CANTX 引脚上检测到。

21.5.3 环回静默模式

图 21-6 bxCAN 工作在环回静默模式



配置 CAN_BTIM 寄存器的 SLM 位为'1'、LBM 位为'1'，则选择环回静默模式。该模式可用于"热自测试"，即可以像环回模式那样测试 bxCAN，但却不会影响 CANTX 和 CANRX 所连接的整个 CAN 系统。

在环回静默模式下，CANRX 引脚与 CAN 总线断开，同时 CANTX 引脚被驱动到隐性位状态。

21.6 调试模式

当微控制器处于调试模式时，N308 核心处于暂停状态，依据下述配置位的状态，bxCAN 可以继续正常工作或停止工作:

- 调试支持（DBG）模块中 CAN1 的 CAN1_STOP 位或 CAN2 的 CAN2_STOP 位。详见第 29.3.2 节: 支持定时器、看门狗、bxCAN 和 I2C 的调试。
- CAN_MCTRL 中的 DBGF 位。详见第 21.9.3 节：CAN 控制和状态寄存器。

21.7 bxCAN 功能描述

21.7.1 发送处理

发送报文的流程为：应用程序选择 1 个**空置**的发送邮箱；在发送邮箱寄存器中写入标识符、数据长度和待发送数据；然后对 CAN_TMIx 寄存器的 TXRQ 位置'1'，来请求发送。TXRQ 位置'1'后，邮箱就不再是空邮箱；而一旦邮箱不再为**空置**，软件对邮箱寄存器就不再有写的权限。TXRQ 位置 1 后，邮箱马上进入**挂起**状态，并等待成为最高优先级的邮箱，参见 21.7.1.1 发送优先级。一旦邮箱成为最高优先级的邮箱，其状态就变为**预定**发送状态。一旦 CAN 总线进入空闲状态，预定发送邮箱中的报文就马上被发送（进入**发送**状态）。一旦邮箱中的报文被成功发送后，它马上变为**空置**邮箱；硬件对 CAN_TSTS 寄存器相对应邮箱的 RQCPM 和 TXOKM 位置'1'，来表明一次成功发送。

CAN_TSTS 寄存器的 ALSTM 位置'1'表明发送失败是由仲裁引起的，而 TERRM 位置'1'表明由发送错误（具体错误可以查询错误状态寄存器 CAN_ESTS 的 LEC[2:0]错误代码位）引起的。

21.7.1.1 发送优先级

由标识符决定

当有超过 1 个发送邮箱在挂起时，发送顺序由邮箱中报文的标识符决定。根据 CAN 协议，标识符数值最低的报文具有最高的优先级。如果标识符的值相等，那么邮箱号小的报文先被发送。

由发送请求次序决定

配置 CAN_MCTRL 寄存器的 TXFP 位为'1'，可以把发送邮箱配置为发送 FIFO，这时发送的优先级由发送请求次序决定。该模式对分段发送很有用。

21.7.1.2 中止

配置 CAN_TSTS 寄存器的 ABRQM 位为'1'，可以中止发送请求。邮箱如果处于挂起或预定状态，发送请求立刻就被中止。如果邮箱处于发送状态，那么中止请求可能导致 2 种结果：如果邮箱中的报文被成功发送，那么邮箱变为空置邮箱，并且 CAN_TSTS 寄存器的 TXOKM 位被硬件置'1'；如果邮箱中的报文发送失败了，那么邮箱变为预定状态，然后发送请求被中止，邮箱变为空置邮箱且 TXOKM 位被硬件清'0'。因此当发送邮箱处于发送状态时，无论发送结果如何，在发送操作结束后邮箱都会变为空置邮箱。

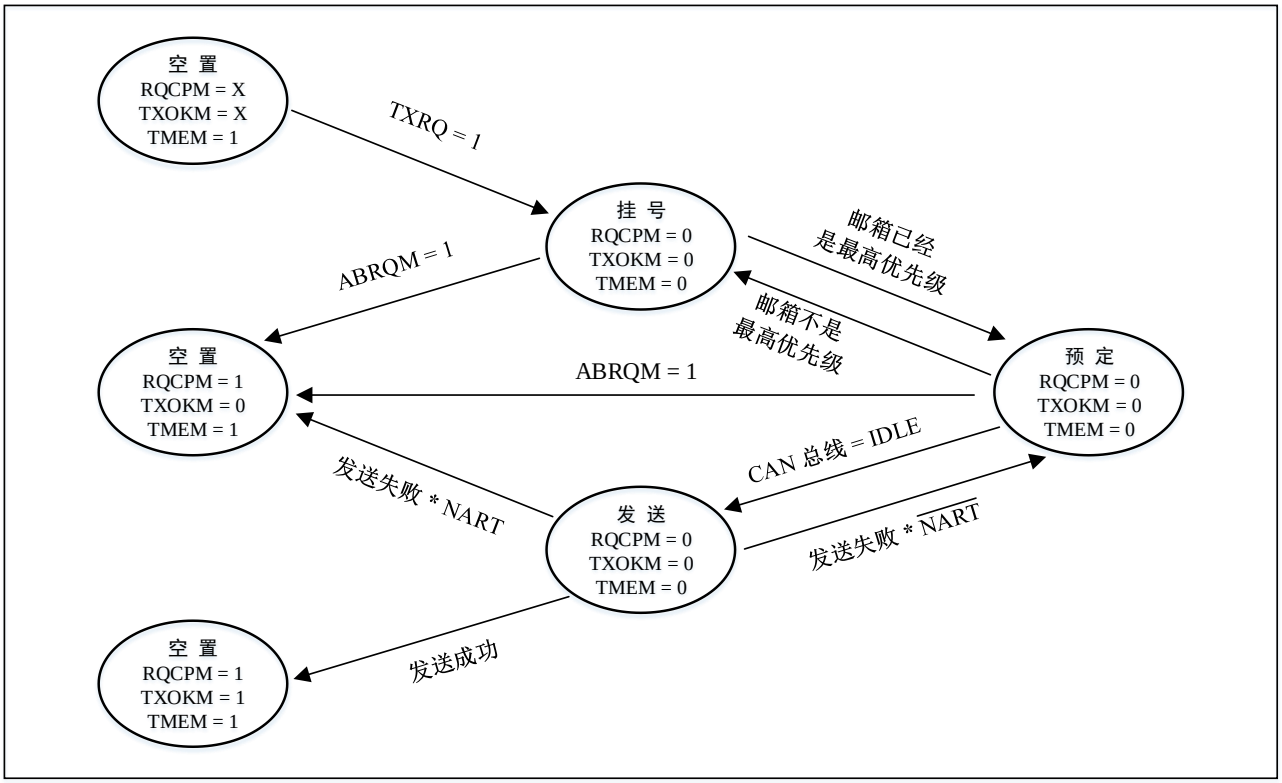
21.7.1.3 禁止自动重传模式

配置 CAN_MCTRL 寄存器的 NART 位为'1'，可以使能禁止自动重传模式模式。该模式对应 CAN 标准中对时间触发通信选项的功能。

在该模式下，发送操作只会执行一次。如果发送操作失败了，不管是由于仲裁丢失或出错，硬件都不会再自动发送该报文。

在一次发送操作结束后，硬件认为发送请求已经完成，硬件配置 CAN_TSTS 寄存器的 RQCPM 位为'1'，同时发送的结果可以查询 CAN_TSTS 寄存器的 TXOKM、ALSTM 和 TERRM 位。

图 21-7 发送邮箱状态



21.7.2 时间触发通信模式

在该模式下，会激活 bxCAN 的内部定时器，内部定时器用于产生（发送与接收邮箱的）时间戳，并将时间戳分别存储在 CAN_RMDTx/CAN_TMDTx 寄存器中。内部定时器在每个 CAN 位时间（见 21.7.7 节）累加。bxCAN 在接收和发送的帧起始位的采样点位置采样内部定时器的值，该采样值即为（发送与接收邮箱的）时间戳。

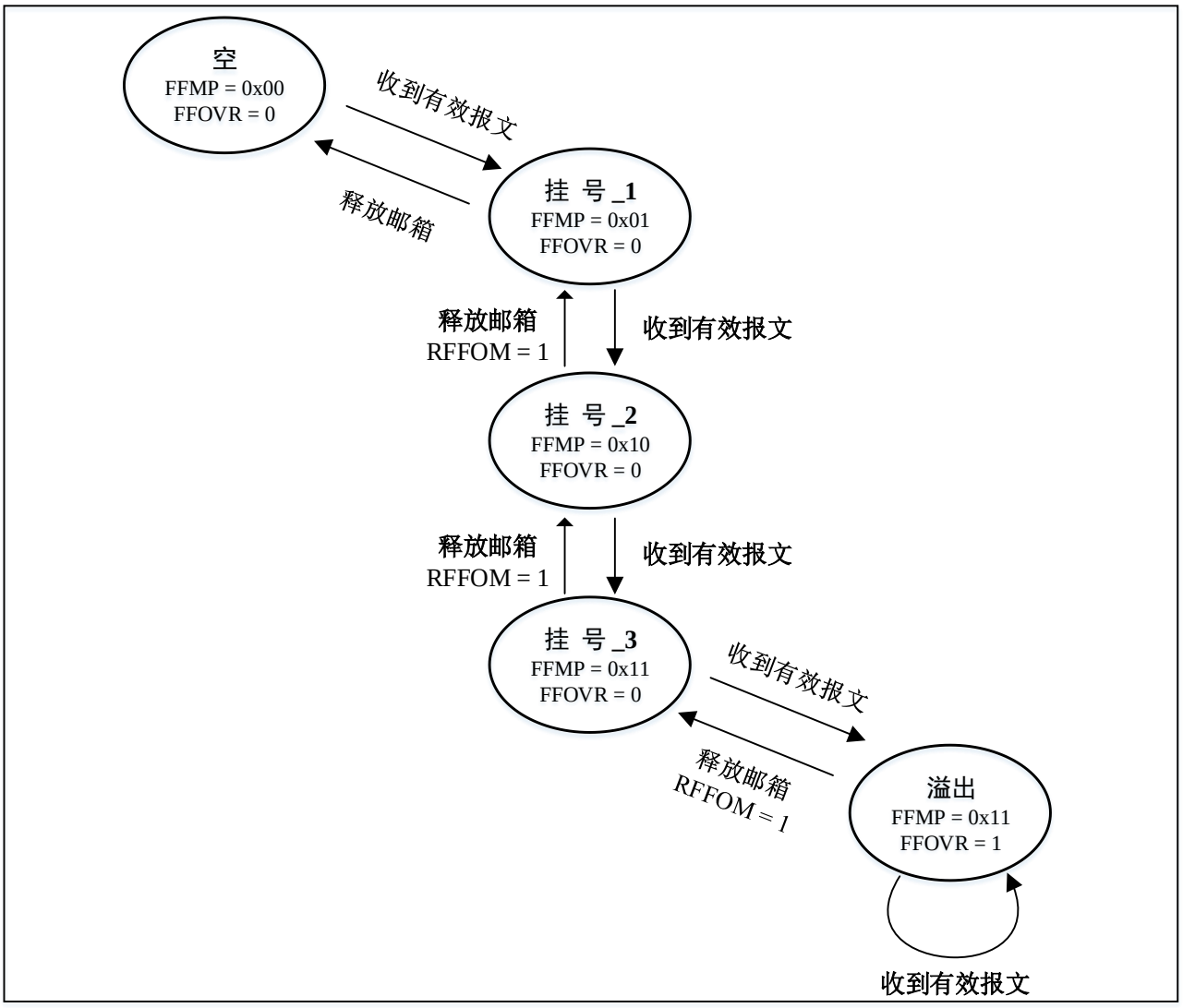
21.7.3 接收管理

3 级邮箱深度的 FIFO 用于存储接收到的报文。FIFO 完全由硬件来管理，这样可以简化应用程序并保证数据的一致性，减少 CPU 的处理时间。应用程序读取 FIFO 输出邮箱读到的是 FIFO 中最先收到的报文。

21.7.3.1 有效报文

根据 CAN 协议，当报文被正确接收（直到 EOF 域的最后一位都没有错误），且通过了标识符过滤，那么该报文被认为是有效报文。请参考 21.7.4 节：标识符过滤。

图 21-8 接收 FIFO 状态



21.7.3.2 FIFO 管理

FIFO 初始为空状态，在接收到第一个有效的报文后，FIFO 状态变为**挂起_1**（pending_1），CAN_RFF 寄存器的 FFMP[1:0]被硬件设置为'01'（二进制 01b）。软件通过读取 FIFO 输出邮箱来读出邮箱中的报文，再将 CAN_RFF 寄存器的 RFFOM 位设置为'1'来释放邮箱，这样 FIFO 又恢复为**空**状态。如果在释放邮箱的同时，又收到了一个有效的报文，那么 FIFO 仍为**挂起_1** 状态，软件通过读取 FIFO 输出邮箱来读出新收到的报文。

如果应用程序不释放邮箱，在接收到下一个有效的报文后，FIFO 状态变为**挂起_2**（pending_2），CAN_RFF 寄存器的 FFMP[1:0]被硬件设置为'10'（二进制 10b）。重复上面的过程，第三个有效的报文把 FIFO 变为**挂起_3** 状态（FFMP[1:0] = 11b）。此时，软件必须对 RFFOM 位设置'1'来释放邮箱，以便 FIFO 可以有空间来存放下一个有效的报文；否则，下一个有效的报文到来时就会导致一个报文的丢失参见 21.7.5 节：报文存储

21.7.3.3 溢出

当 FIFO 处于**挂起_3** 状态（即 FIFO 的 3 个邮箱都是满的），下一个有效的报文就会导致**溢出**，并且一个报文会丢失。此时，硬件对 CAN_RFF 寄存器的 FFOVR 位进行置'1'来表明溢出情况。至于哪个报文会被丢弃，取决于对 FIFO 的设置：

- 如果禁用了 FIFO 锁定功能（配置 CAN_MCTRL 寄存器的 RFLM 位为'0'），那么 FIFO 中最后收到的报文就被新报文所覆盖。这样，最新收到的报文不会被丢弃掉。
- 如果启用了 FIFO 锁定功能（配置 CAN_MCTRL 寄存器的 RFLM 位为'1'），那么新收到的报文就被丢弃，软件可以读到 FIFO 中最早收到的 3 个报文。

21.7.3.4 接收相关的中断

当接收 FIFO 存入一个报文，硬件就会更新 CAN_RFF 寄存器的 FFMP[1:0]位，如果当前已经使能了 FIFO 消息挂起中断（CAN_INTE 寄存器的 FMPITE 位为'1'），那么就会产生 FIFO 消息挂起中断请求。

当 FIFO 变满时（即第 3 个报文被存入），CAN_RFF 寄存器的 FFULL 位就被置'1'， 如果当前已经使能了 FIFO 满中断（CAN_INTE 寄存器的 FFITE 位为'1'），那么就会产生 FIFO 满中断请求。

在 FIFO 处于溢出的情况下，FFOVR 位被置'1'，如果当前已经使能了 FIFO 溢出中断（CAN_INTE 寄存器的 FOVITE 位为'1'），那么就会产生 FIFO 溢出中断请求。

21.7.4 标识符过滤

在 CAN 协议里，报文的标识符不代表节点的地址，而是跟报文的内容相关的。因此，发送者通过广播的形式把报文发送给所有的接收者。节点在接收报文时，根据**标识符的值**决定软件是否需要该报文；如果需要，就拷贝到 SRAM（bxCAN 自己的 SRAM）里；如果不需要，报文就被丢弃。这个过程不需要软件干预。

为满足这一需求，bxCAN 控制器为应用程序提供了 14 个位宽可变的、可配置的过滤器组（13~0），用于只接收那些软件需要的报文。硬件过滤的做法比软件过滤减少了对 CPU 的占用时间。每个过滤器组 x 包含 2 个 32 位寄存器，即 CAN_FxB0 和 CAN_FxB1。

21.7.4.1 可变的位宽

为了满足不同应用程序的不同需求，每个过滤器组的位宽都可以独立配置。每个过滤器组可以配置为两种不同的位宽：

- 1 个 32 位过滤器，包括：STDID[10:0]、EXTID[17:0]、IDE 和 RTRQ 位。
- 2 个 16 位过滤器，包括：STDID[10:0]、IDE、RTRQ 和 EXTID[17:15]位，参见图 21-9。

此外过滤器可配置为两种不同的模式，即屏蔽位模式和标识符列表模式。

21.7.4.2 屏蔽位模式

在屏蔽位模式下，标识符寄存器和屏蔽寄存器一起，指定报文标识符的任何一位，应该按照“必须匹配”或“不用关心”处理。

21.7.4.3 标识符列表模式

在标识符列表模式下，屏蔽寄存器也被当作标识符寄存器用。因此，不是采用一个标识符加一个屏蔽位的方式，而是使用 2 个标识符寄存器。接收报文标识符的每一位都必须跟过滤器标识符相同。

21.7.4.4 过滤器组位宽和模式的设置

过滤器组可以通过相应的 CAN_FMC 寄存器配置。在配置一个过滤器组前，必须通过清除 CAN_FA 寄存器的 FAC 位，把它设置为禁用状态。通过设置 CAN_FS1 的相应 FSCx 位，可以配置一个过滤器组的位宽，请参见图 21-9。通过 CAN_FM1 的 FBx 位，可以配置对应的屏蔽/ 标识符寄存器的**标识符列表**模式或**屏蔽位**模式。

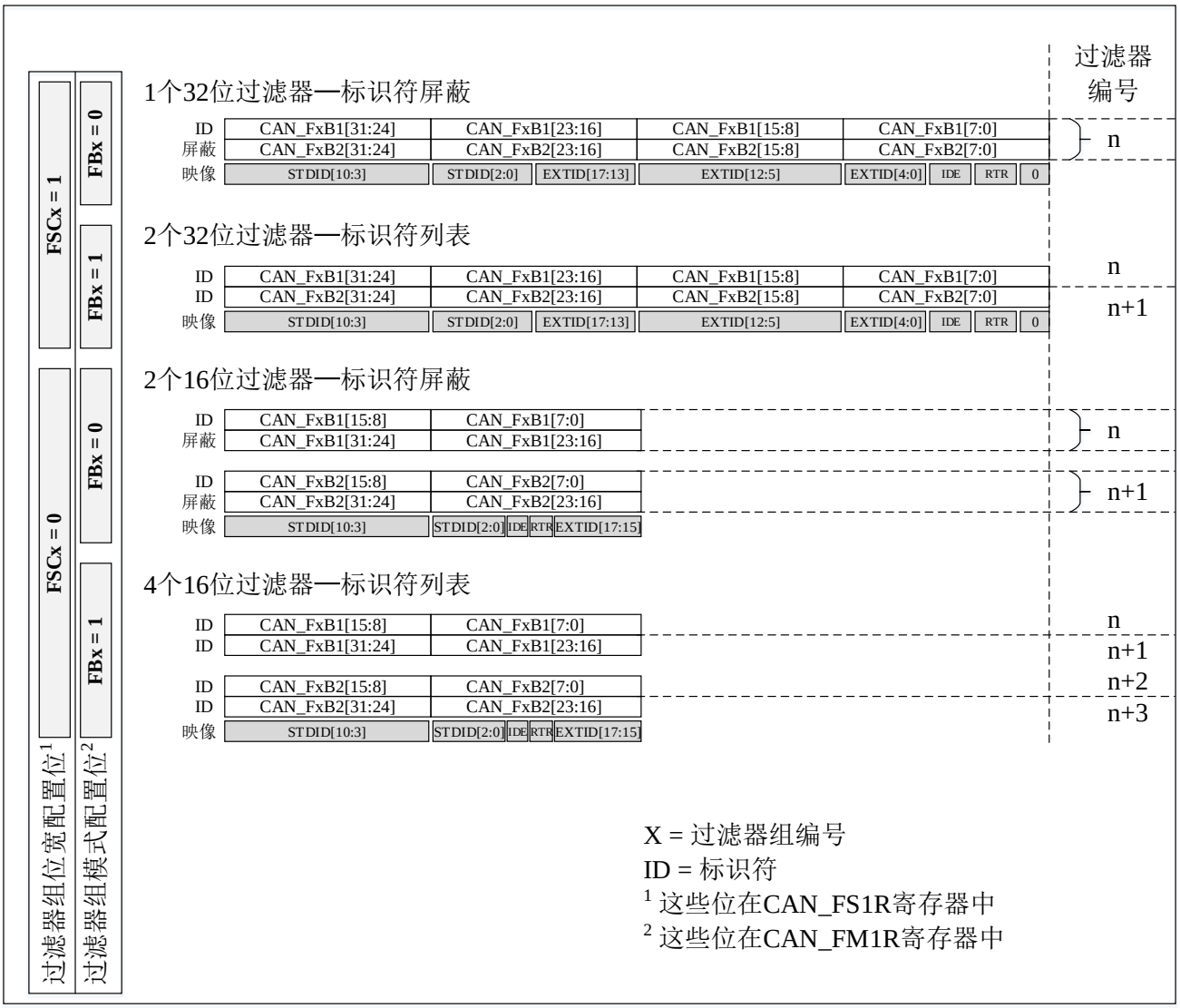
为了过滤出一组标识符，应该设置过滤器组工作在屏蔽位模式。

为了过滤出一个标识符，应该设置过滤器组工作在标识符列表模式。

应将应用程序不用的过滤器组保持在禁用状态。

过滤器组中的每个过滤器，都被编号（过滤器号，从 0 到某个最大数值）取决于过滤器组的模式和位宽的设置。关于过滤器配置，参见下图。

图 21-9 过滤器组位宽设置 – 寄存器组织



21.7.4.5 过滤器匹配序号

一旦收到的报文被存入 FIFO，就可被应用程序访问。通常情况下，报文中的数据被拷贝到 SRAM 中；为了把数据拷贝到合适的位置，应用程序需要根据报文的标识符来辨别不同的数据。bxCAN 提供了过滤器匹配序号，以简化这一辨别过程。

根据过滤器优先级规则，每个收到的报文都有与它相关联的过滤器匹配序号。接收到的报文过与对应的滤器匹配序号一起被存入接收邮箱中。

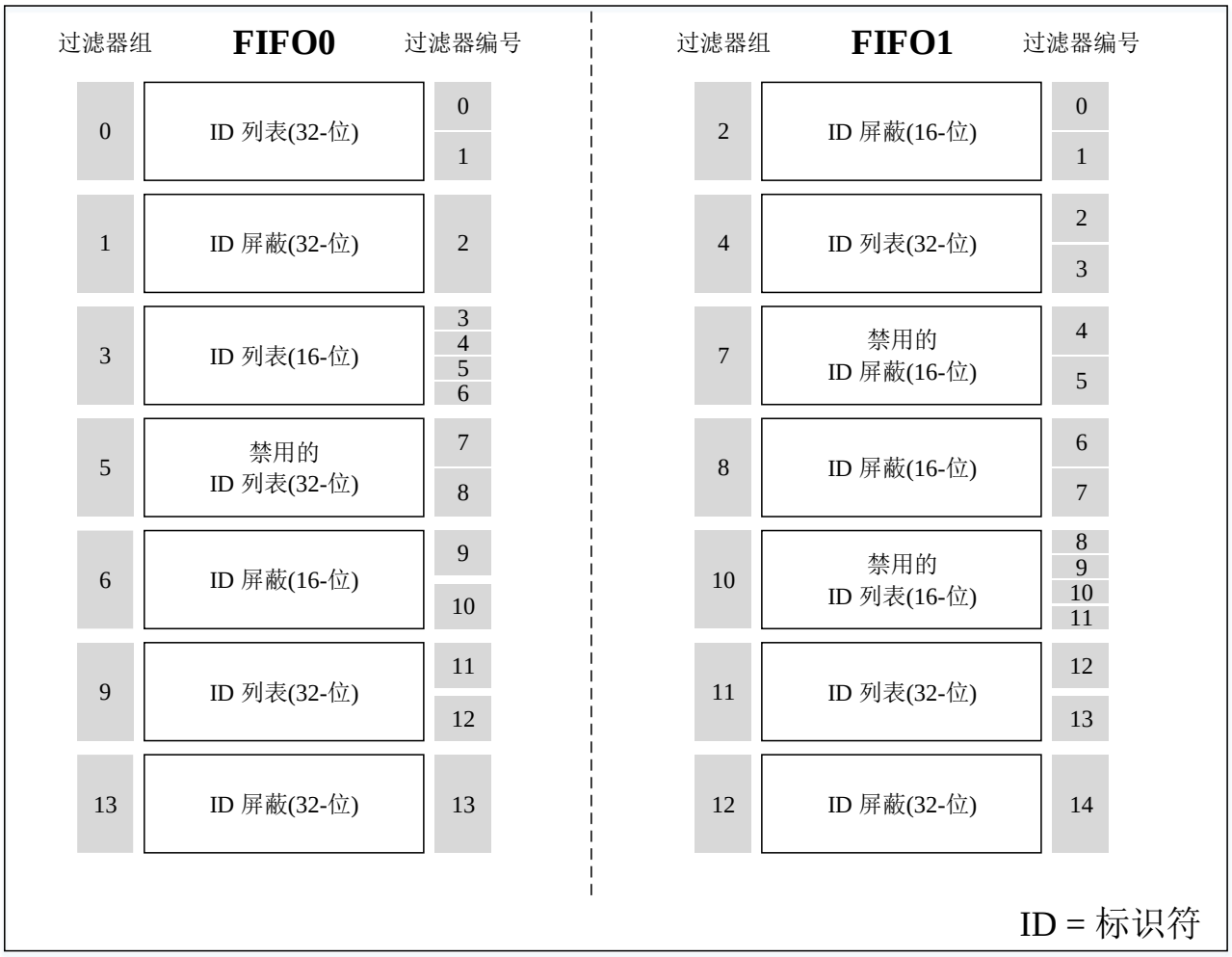
过滤器匹配序号可以通过下面两种方式来使用:

- 把过滤器匹配序号跟一系列所期望的值进行比较。
- 把过滤器匹配序号当作一个索引来访问目标地址。

对于标识符列表模式下的过滤器（非屏蔽方式的过滤器），软件不需要直接跟标识符进行比较。对于屏蔽位模式下的过滤器，软件只须对需要的那些屏蔽位（必须匹配的位）进行比较即可。

在给过滤器编号时，并不考虑过滤器组是否为激活状态。另外，每个 FIFO 各自对其关联的过滤器进行编号。请参考下图的例子。

图 21-10 过滤器编号的例子

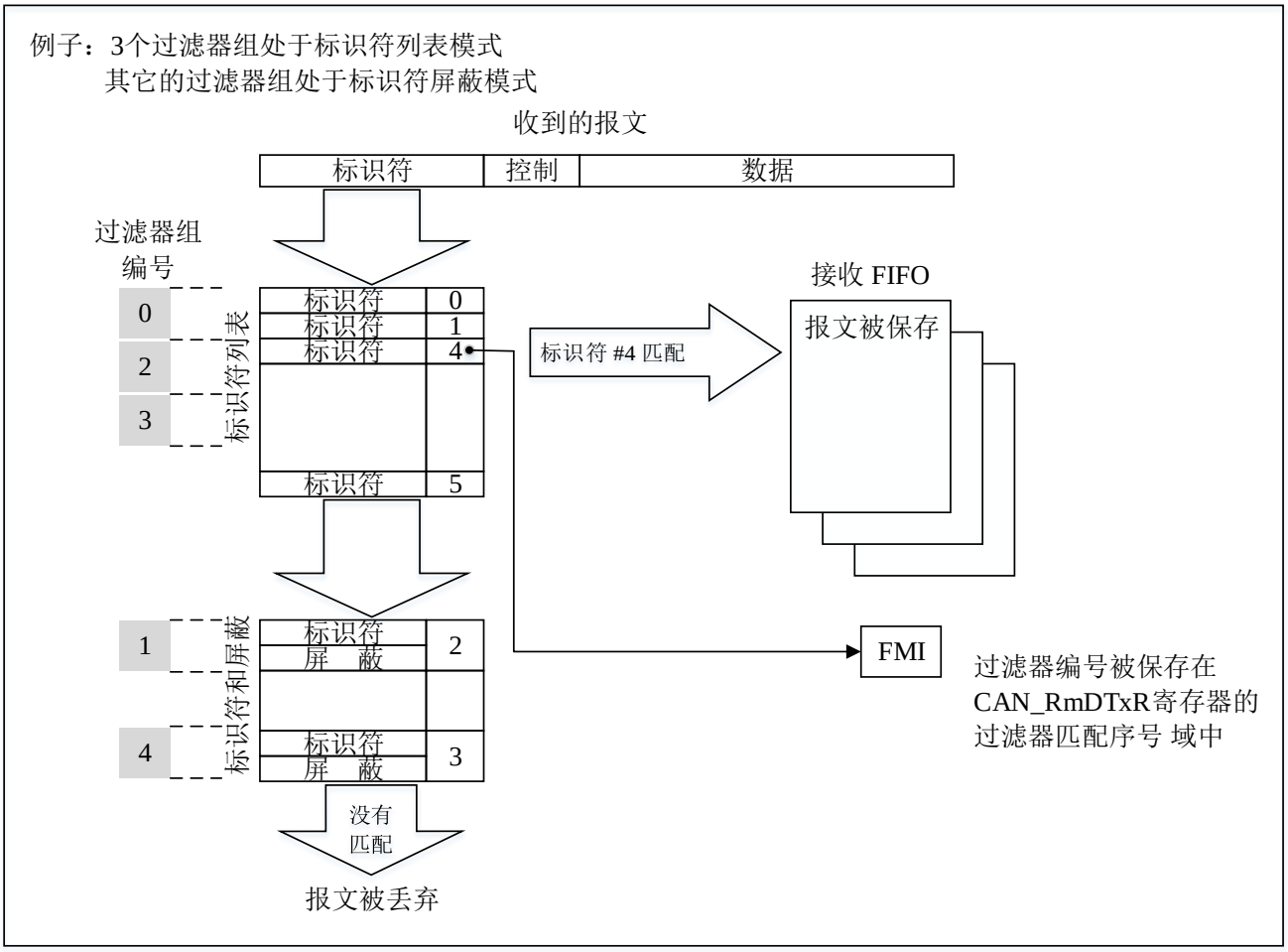


21.7.4.6 过滤器优先级规则

根据过滤器的不同配置，有可能一个报文标识符能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，根据下列优先级规则来确定：

- 32 位-位宽的过滤器，优先级高于 16 位-位宽的过滤器。
- 相同位宽的过滤器，标识符列表模式的优先级高于屏蔽位模式。
- 位宽和模式都相同的过滤器，优先级由过滤器号决定，过滤器号小的优先级高。

图 21-11 过滤器机制的例子



上面的例子说明了 bxCAN 的过滤器规则：在接收一个报文时，其标识符首先与配置在标识符列表模式下的过滤器相比较；如果匹配上，报文就被存放到相关联的 FIFO 中，并且所匹配的过滤器的序号被存入过滤器匹配序号中。如同例子中所显示，报文标识符跟#4 标识符匹配，因此报文内容和 FMI4 被存入 FIFO。

如果没有匹配，报文标识符接着与配置在屏蔽位模式下的过滤器进行比较。

如果报文标识符没有跟过滤器中的任何标识符相匹配，那么硬件就自动丢弃该报文，不需要软件干预。

21.7.5 报文存储

邮箱是软件和硬件之间传递报文的接口。邮箱包含了所有跟报文有关的信息：标识符、数据、控制、状态和时间戳信息。

21.7.5.1 发送邮箱

软件需要把待发送报文的各种信息配置到一个空的发送邮箱中，然后再使能发送请求。发送的状态可以查询 CAN_TSR 寄存器。

表 21-1 发送邮箱寄存器列表

相对发送邮箱基地址的偏移量	寄存器名
0	CAN_TMIx
4	CAN_TMDTx
8	CAN_TMDLx
12	CAN_TMDHx

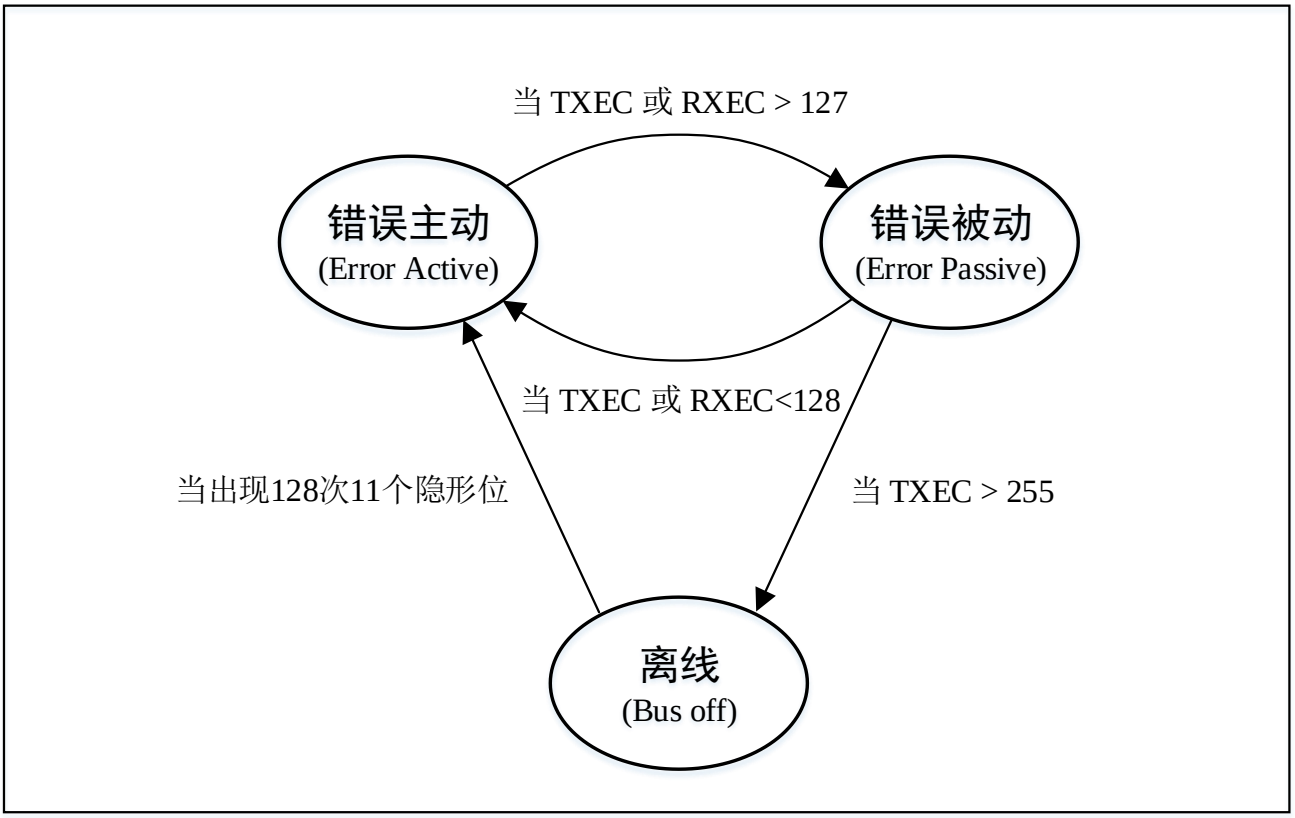
21.7.5.2 接收邮箱（FIFO）

软件可以访问接收 FIFO 的输出邮箱读取接收到的报文。一旦软件处理了报文（如把它读出来），软件就应该对 CAN_RFFx 寄存器的 RFFOM 位进行置'1'，来释放该报文，以便为后面收到的报文留出存储空间。过滤器匹配序号存放在 CAN_RMDTx 寄存器的 FMI 域中。16 位的时间戳存放在 CAN_RMDTx 寄存器的 MTIM[15:0]域中。

表 21-2 接收邮箱寄存器列表

相对发送邮箱基地址的偏移量	寄存器名
0	CAN_RMIx
4	CAN_RMDTx
8	CAN_RMDLx
12	CAN_RMDHx

图 21-12 CAN 错误状态图



21.7.6 出错管理

CAN 协议描述的出错管理，完全由硬件通过发送错误计数器（CAN_ESTS 寄存器里的 TXEC 域），和接收错误计数器（CAN_ESTS 寄存器里的 RXEC 域）来实现，其值根据错误的情况而增加或减少。关于 TXEC 和 RXEC 管理的详细信息，请参考 CAN 标准。

软件可以读出发送/接收错误计数器的值来判断 CAN 网络的稳定性。

通过读取错误状态寄存器 CAN_ESTS 的 LEC[2:0]位来获取当前错误状态的详细信息。通过设置 CAN_INTE 寄存器（比如 ERRITE 位），当检测到出错时软件可以灵活地控制中断的产生。

21.7.6.1 离线恢复

当 TXEC 大于 255 时，bxCAN 就进入离线状态，同时 CAN_ESTS 寄存器的 BOFFL 位被置'1'。在离线状态下，bxCAN 无法接收和发送报文。

根据 CAN_MCTRL 寄存器中 ABOM 位的设置，bxCAN 可以自动或在软件的请求下，从离线状态恢复（变为错误主动状态）。在这两种情况下，bxCAN 都必须等待一个 CAN 标准所描述的恢复过程（CAN RX 引脚上检测到 128 次 11 个连续的隐性位）。

如果 ABOM 位为'1'，bxCAN 进入离线状态后，就自动开启恢复过程。

如果 ABOM 位为'0'，软件必须先请求 bxCAN 进入然后再退出初始化模式，随后恢复过程才被开启。

*注：在初始化模式下，bxCAN 不会监视 CAN RX 引脚的状态，这样就不能完成恢复过程。**为了完成恢复过程，bxCAN 必须工作在正常模式**。*

21.7.7 位时间特性

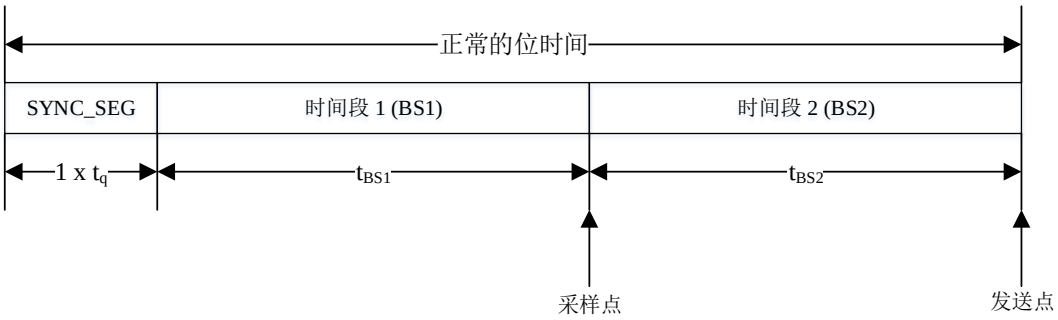
位时间特性逻辑通过采样来监视串行的 CAN 总线，并且通过与帧起始位的边沿进行同步，及通过与后面的边沿进行重新同步，来调整其采样点。

它的操作可以简单解释为，如下所述把名义上的每位时间分为 3 段：

- **同步段（SYNC_SEG）**：通常期望位的变化发生在该时间段内。其值固定为 1 个时间单元（1 x t_{CAN}）。
- **时间段 1（BS1）**：定义采样点的位置。它包含 CAN 标准里的 PROP_SEG 和 PHASE_SEG1。其值可以编程为 1 到 16 个时间单元，但也可以被自动延长，用来补偿因网络中不同节点的频率差异所造成的相位的正向漂移。
- **时间段 2（BS2）**：定义发送点的位置。它代表 CAN 标准里的 PHASE_SEG2。其值可以编程为 1 到 8 个时间单元，但也可以被自动缩短以补偿相位的负向漂移。
- 重新同步跳跃宽度（RSJW）定义了，在每位中可以延长或缩短多少个时间单元的上限。其值可以编程为 1 到 4 个时间单元。
- 有效跳变被定义为，当 bxCAN 自己没有发送隐性位时，从显性位到隐性位的第 1 次转变。
- 如果在时间段 1（BS1）而不是在同步段（SYNC_SEG）检测到有效跳变，那么 BS1 的时间就被延长最多 RSJW 那么长，从而采样点被延迟了。
- 相反如果在时间段 2（BS2）而不是在 SYNC_SEG 检测到有效跳变，那么 BS2 的时间就被缩短最多 RSJW 那么长，从而采样点被提前了。
- 为了避免软件的编程错误，对位时间特性寄存器（CAN_BTIM）的设置，只能在 bxCAN 处于初始化状态下进行。

注：关于 CAN 位时间特性和重同步机制的详细信息，请参考 ISO11898 标准。

图 21-13 位时序



$$波特率 = \frac{1}{正常的位时间}$$

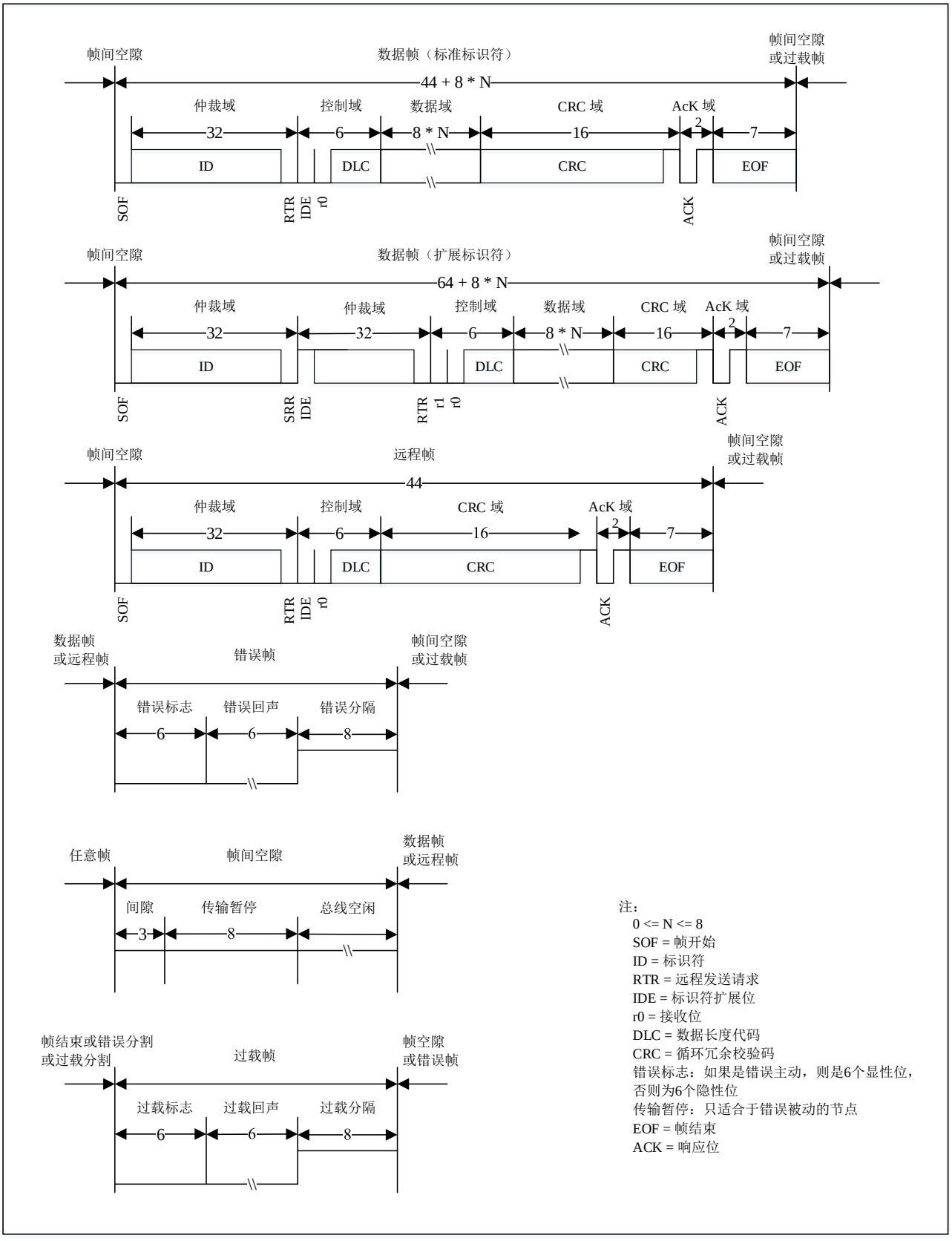
正常的位时间 = 1 x t_q + t_{BS1} + t_{BS2}

其中：

t_{BS1} = t_q x (TBS1[3:0] + 1),
t_{BS2} = t_q x (TBS2[2:0] + 1),
t_q = (BRTP[9:0] + 1) x t_{PCLK}，

t_q表示1个时间单元
t_{PCLK} = APB时钟的时间周期
BRTP[9:0]，TBS1[3:0]和TBS2[2:0]在CAN_BTIM寄存器中定义

图 21-14 各种 CAN 帧

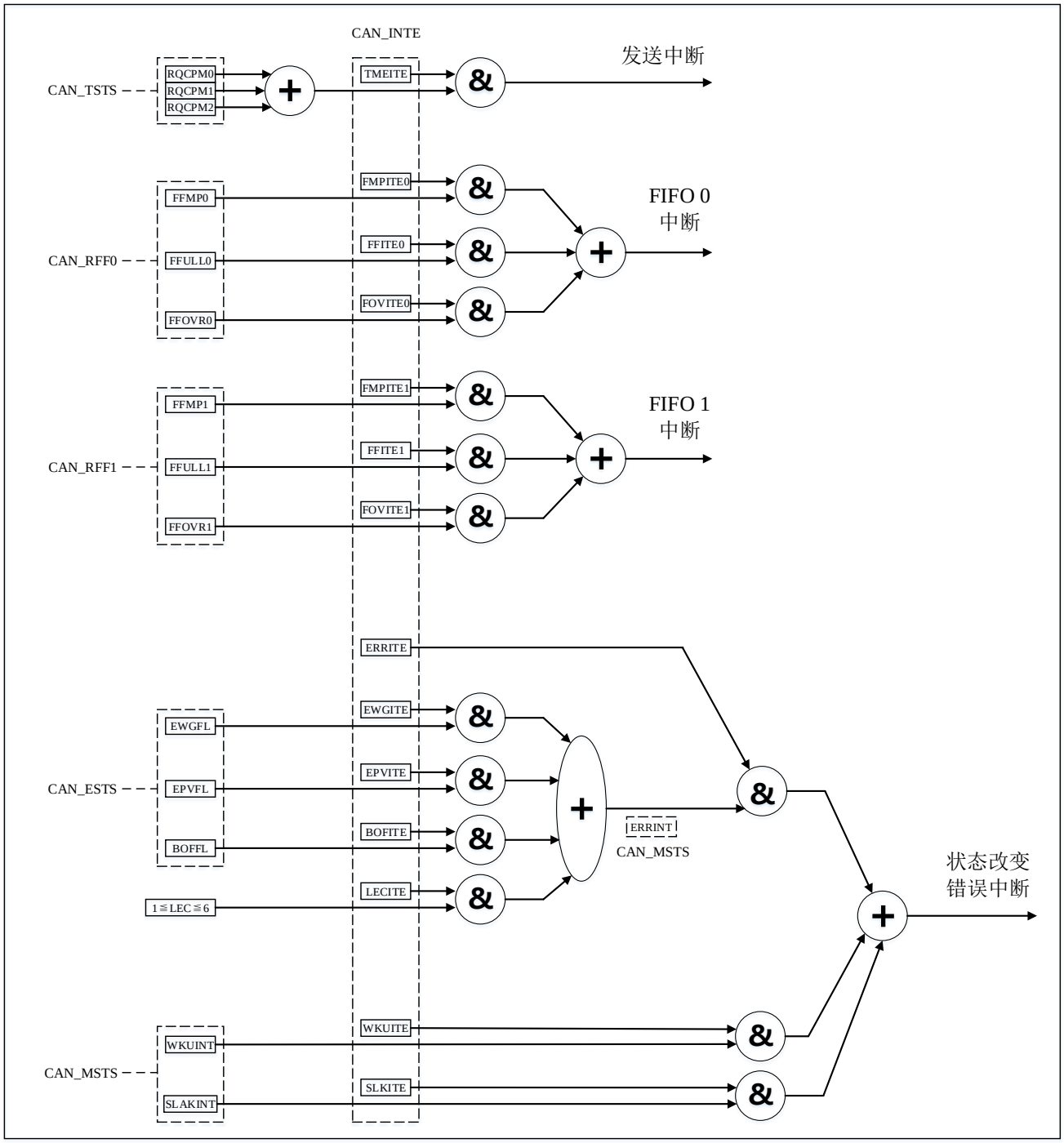


注：
0 <= N <= 8
SOF = 帧开始
ID = 标识符
RTR = 远程发送请求
IDE = 标识符扩展位
r0 = 接收位
DLC = 数据长度代码
CRC = 循环冗余校验码
错误标志：如果是错误主动，则是6个显性位，否则为6个隐性位
传输暂停：只适合于错误被动的节点
EOF = 帧结束
ACK = 响应位

21.8 bxCAN 中断

bxCAN 拥有 4 个中断向量。通过设置 CAN 中断使能寄存器（CAN_INTE），单独配置每个中断源的使能/禁用。

图 21-15 事件标志和中断产生



- 发送中断可由下列事件产生：
  - 发送邮箱 0 变为空，CAN_TSTS 寄存器的 RQCPM0 位被置'1'。

- 发送邮箱 1 变为空，CAN_TSTS 寄存器的 RQCPM1 位被置'1'。
- 发送邮箱 2 变为空，CAN_TSTS 寄存器的 RQCPM2 位被置'1'。

- FIFO0 中断可由下列事件产生：
  - FIFO0 接收到一个新报文，CAN_RFF0 寄存器的 FFMP0 位不再是'00'。
  - FIFO0 变为满的情况，CAN_RFF0 寄存器的 FFULL0 位被置'1'。
  - FIFO0 发生溢出的情况，CAN_RFF0 寄存器的 FFOVR0 位被置'1'。
- FIFO1 中断可由下列事件产生：
  - FIFO1 接收到一个新报文，CAN_RFF1 寄存器的 FFMP1 位不再是'00'。
  - FIFO1 变为满的情况，CAN_RFF1 寄存器的 FFULL1 位被置'1'。
  - FIFO1 发生溢出的情况，CAN_RFF1 寄存器的 FFOVR1 位被置'1'。
- **错误和状态变化中断**可由下列事件产生：
  - 出错情况，关于出错情况的详细信息请参考 CAN 错误状态寄存器（CAN_ESTS）。
  - 唤醒情况，在 CAN 接收引脚上监视到帧起始位（SOF）。
  - CAN 进入睡眠模式。

21.9 bxCAN 寄存器描述

必须以字（32 位）的方式操作这些外设寄存器。

21.9.1 寄存器访问保护

由于对某些寄存器的错误访问会导致当前 CAN 节点对整个 CAN 网络产生暂时性干扰。因此，应用软件只能在 CAN 处于初始化模式时修改 CAN_BTR 寄存器。

虽然错误数据的发送对 CAN 网的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，软件只能在发送邮箱为空的状态时修改发送的数据，请参见图 21-7。

过滤器的数值只能在关闭对应过滤器组的状态下，或设置 FINITM 位为'1'后才能修改。此外，只有在设置整个过滤器为初始化模式下(即 FINITM=1)，才能修改过滤器的设置，即修改 CAN_FMx, CAN_FSx 和 CAN_FFA 寄存器。

21.9.2 bxCAN 寄存器地址映像

表 21-3 bxCAN 寄存器地址映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	CAN_MCTRL	Reserved															DBGF	MRST	Reserved								TTCM	ABOM	AWKUM	NART	RFLM	TXFP	SLPRQ	INIRQ
	Reset Value																1	0									0	0	0	0	0	0	1	0
004h	CAN_MSTS	Reserved																				RXS	LSMP	RXMD	TXMD	Reserved			SLAKINT	WKUINT	ERRINT	SLPAK	INIAK

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset Value																					1	1	0	0				0	0	0	1	0
008h	CAN_TSTS	LOWM[2:0]			TMEM[2:0]			CODE[1:0]		ABRQM2	Reserved			TERRM2	ALSTM2	TXOKM2	RQCPM2	ABRQM1	Reserved			TERRM1	ALSTM1	TXOKM1	RQCPM1	ABRQM0	Reserved		TERRM0	ALSTM0	TXOKM0	RQCPM0	
	Reset Value	0	0	0	1	1	1	0	0	0				0	0	0	0	0				0	0	0	0	0			0	0	0	0	
00Ch	CAN_RFF0	Reserved																									RFFOM0	FFOVR0	FFULL0	Reserved		FFPMP0[1:0]	
	Reset Value																										0	0	0			0	0
010h	CAN_RFF1	Reserved																									RFFOM1	FFOVR1	FFULL1	Reserved		FFPMP1[1:0]	
	Reset Value																										0	0	0			0	0
014h	CAN_INTE	Reserved														SLKITE	WKUITE	ERRITE	Reserved			LECITE	BOFITE	EPVITE	EWGITE	Reserved	FOVITE1	FFITE1	FMPITE1	FOVITE0	FFITE0	FMPITE0	TMEITE
	Reset Value															0	0	0				0	0	0	0		0	0	0	0	0	0	0
018h	CAN_ESTS	RXEC[7:0]								TXEC[7:0]									Reserved								LEC[2:0]			Reserved	BOFFL	EPVFL	EWGFL
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										0	0	0		0	0	0
01Ch	CAN_BTIM	SLM	LBM	Reserved				RSJW[1:0]		Reserved	TBS2[2:0]			TBS1[3:0]					Reserved			BRTP[9:0]											
	Reset Value	0	0					0	0		0	1	0	0	0	1	1					0	0	0	0	0	0	0	0	0	0	0	0
020h – 17Fh	Reserved																																
180h	CAN_TMI0	STDID[10:0]/EXTID[28:18]											EXTID[17:0]																		IDE	RTRQ	TXRQ
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0
184h	CAN_TMDT0	MTIM[15:0]																	Reserved						TGT	Reserved				DLC[3:0]			
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x								x					x	x	x	x
188h	CAN_TMDL0	DATA3[7:0]								DATA2[7:0]									DATA1[7:0]											DATA0[7:0]			
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
18Ch	CAN_TMDH0	DATA7[7:0]								DATA6[7:0]									DATA5[7:0]											DATA4[7:0]			
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
190h	CAN_TMI1	STDID[10:0]/EXTID[28:18]											EXTID[17:0]																		IDE	RTRQ	TXRQ
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0
194h	CAN_TMDT1	MTIM[15:0]																	Reserved						TGT	Reserved				DLC[3:0]			
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x								x					x	x	x	x
198h	CAN_TMDL1	DATA3[7:0]								DATA2[7:0]									DATA1[7:0]											DATA0[7:0]			
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
19Ch	CAN_TMDH1	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1A0h	CAN_TMI2	STDID[10:0]/EXTID[28:18]											EXTID[17:0]																		IDE	RTRQ	TXRQ
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0
1A4h	CAN_TMDT2	MTIM[15:0]																Reserved							TGT	Reserved					DLC[3:0]			
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x								x					x	x	x	x
1A8h	CAN_TMDL2	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1ACh	CAN_TMDH2	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1B0h	CAN_RMI0	STDID[10:0]/EXTID[28:18]											EXTID[17:0]																		IDE	RTRQ	Reserved
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
1B4h	CAN_RMDT0	MTIM[15:0]																FMI[7:0]								Reserved					DLC[3:0]		
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x					x	x	x	x
1B8h	CAN_RMDL0	DATA3[7;0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1BCh	CAN_RMDH0	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1C0h	CAN_RMI1	STDID[10:0]/EXTID[28:18]											EXTID[17:0]																		IDE	RTRQ	Reserved
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
1C4h	CAN_RMDT1	MTIM[15:0]																FMI[7:0]								Reserved					DLC[3:0]		
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x					x	x	x	x
1C8h	CAN_RMDL1	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1CCh	CAN_RMDH1	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1D0h – 1FFh	Reserved																																
200h	CAN_FMC	Reserved																															FINITM

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset Value	Reserved																															1
204h	CAN_FM1	Reserved																		FB[13:0]													
	Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
208h	Reserved																																
20Ch	CAN_FS1	Reserved																		FSC[13:0]													
	Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
210h	Reserved																																
214h	CAN_FFA1	Reserved																		FAF[13:0]													
	Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
218h	Reserved																																
21Ch	CAN_FA1	Reserved																		FAC[13:0]													
	Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
220h	Reserved																																
224h – 23Fh	Reserved																																
240h	CAN_F0B1	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
244h	CAN_F0B2	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
248h	CAN_F1B1	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
24Ch	CAN_F1B2	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
. . .	. . .	Reserved																															
2A8h	CAN_F13B1	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
2ACh	CAN_F13B2	FBC[31:0]																															
	Reset Value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

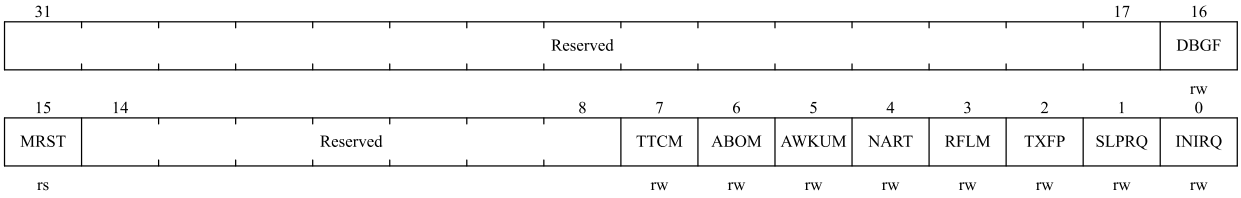
21.9.3 CAN 控制和状态寄存器

有关寄存器说明中的缩写，参见 1.1 节。

21.9.3.1 CAN 主控制寄存器（CAN_MCTRL）

地址偏移量：0x00

复位值：0x0001 0002



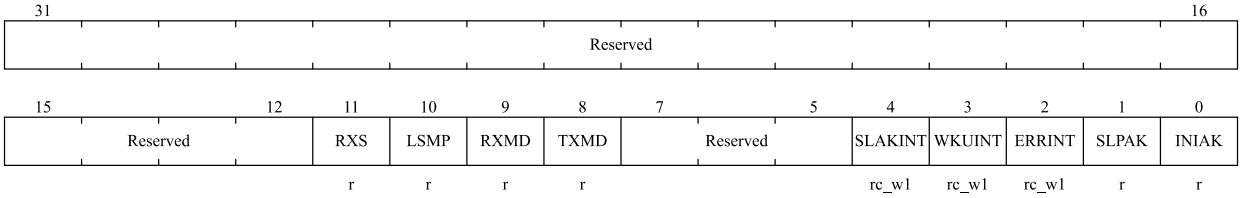
位域	名称	描述
31:17	Reserved	保留，硬件强制为 0。
16	DBGF	调试冻结（Debug freeze） 0：在调试时，CAN 照常工作 1：在调试时，冻结 CAN 的接收/发送。仍然可以正常地读写和控制接收 FIFO。
15	MRST	bxCAN 软件复位（bxCAN software master reset） 0：本外设正常工作; 1：对 bxCAN 进行强行复位，复位后 bxCAN 进入睡眠模式（FFMP 位和 CAN_MCTRL 寄存器被初始化为其复位值）。此后硬件自动对该位清'0'。
14:8	Reserved	保留，硬件强制为 0。
7	TTCM	时间触发通信模式（Time triggered communication mode） 0：禁止时间触发通信模式; 1：允许时间触发通信模式。 *注： 要想了解关于时间触发通信模式的更多信息，请参考 21.7.2：时间触发通信模式。*
6	ABOM	自动离线（Bus-Off）管理（Automatic bus-off management） 该位决定 CAN 硬件在什么条件下可以退出离线状态。 0：离线状态的退出过程是，软件对 CAN_MCTRL 寄存器的 INIRQ 位进行置'1'随后清'0'后，一旦硬件检测到 128 次 11 位连续的隐性位，则退出离线状态; 1：一旦硬件检测到 128 次 11 位连续的隐性位，则自动退出离线状态。 *注：关于离线状态的更多信息，请参考 21.7.6：出错管理。*
5	AWKUM	自动唤醒模式（Automatic wakeup mode） 该位决定 CAN 处在睡眠模式时由硬件还是软件唤醒 0：睡眠模式通过清除 CAN_MCTRL 寄存器的 SLPRQ 位，由软件唤醒; 1：睡眠模式通过检测 CAN 报文，由硬件自动唤醒。唤醒的同时，硬件自动对 CAN_MSTS 寄存器的 SLPRQ 和 SLPAK 位清'0'。
4	NART	禁止报文自动重传（No automatic retransmission）

位域	名称	描述
		0：按照 CAN 标准，CAN 硬件在发送报文失败时会一直自动重传直到发送成功； 1：CAN 报文只被发送 1 次，不管发送的结果如何（成功、出错或仲裁丢失）。
3	RFLM	接收 FIFO 锁定模式（Receive FIFO locked mode） 0：在接收溢出时 FIFO 未被锁定，当接收 FIFO 的报文未被读出，下一个收到的报文会覆盖原有的报文； 1：在接收溢出时 FIFO 被锁定，当接收 FIFO 的报文未被读出，下一个收到的报文会被丢弃。
2	TXFP	发送 FIFO 优先级（Transmit FIFO priority） 当有多个报文同时在等待发送时，该位决定这些报文的发送顺序 0：优先级由报文的标识符来决定； 1：优先级由发送请求的顺序来决定。
1	SLPRQ	睡眠模式请求（Sleep mode request） 软件对该位置'1'可以请求 CAN 进入睡眠模式，一旦当前的 CAN 活动（发送或接收报文）结束，CAN 就进入睡眠。 软件对该位清'0'使 CAN 退出睡眠模式。 当设置了 AWKUM 位且在 CAN Rx 信号中检测出 SOF 位时，硬件对该位清'0'。在复位后该位被置'1'，即 CAN 在复位后处于睡眠模式。
0	INIRQ	初始化请求（Initialization request） 软件对该位清'0'可使 CAN 从初始化模式进入正常工作模式：当 CAN 在接收引脚检测到连续的 11 个隐性位后， CAN 就达到同步，并为接收和发送数据作好准备了。为此，硬件相应地对 CAN_MSTS 寄存器的 INIAK 位清'0'。 软件对该位置 1 可使 CAN 从正常工作模式进入初始化模式：一旦当前的 CAN 活动（发送或接收）结束，CAN 就进入初始化模式。相应地，硬件对 CAN_MSTS 寄存器的 INIAK 位置'1'。

21.9.3.2 CAN 主状态寄存器（CAN_MSTS）

地址偏移量：0x04

复位值：0x0000 0C02



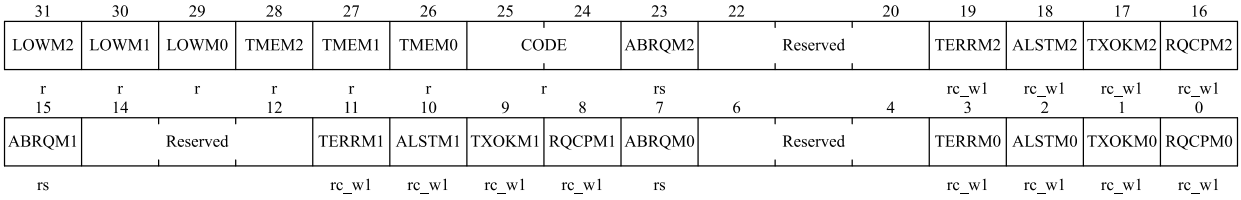
位域	名称	描述
31:12	Reserved	保留，硬件强制为 0。
11	RXS	CAN 接收电平（CAN Rx signal） 该位反映 CAN 接收引脚（CAN_RX）的实际电平。
10	LSMP	上次采样值（Last sample point） CAN 接收引脚的上次采样值（对应于当前接收位的值）。

位域	名称	描述
9	RXMD	接收模式（Receive mode） 该位为'1'表示 CAN 当前为接收器。
8	TXMD	发送模式（Transmit mode） 该位为'1'表示 CAN 当前为发送器。
7:5	Reserved	保留，硬件强制为 0。
4	SLAKINT	睡眠确认中断（Sleep acknowledge interrupt） 当 SLKITE=1，一旦 CAN 进入睡眠模式硬件就对该位置'1'，紧接着相应的中断被触发。当设置该位为'1'时，如果设置了 CAN_INTE 寄存器中的 SLKITE 位，将产生一个状态改变中断。 软件可对该位写 1 清'0'，当 SLAK 位被清'0'时硬件也对该位清'0'。 *注：当 SLKITE=0，不应该查询该位，而应该查询 SLPAK 位来获知睡眠状态。*
3	WKUINT	唤醒中断挂起（Wakeup interrupt） 当 CAN 处于睡眠状态，一旦检测到帧起始位（SOF），硬件就置该位为'1'；并且如果 CAN_INTE 寄存器的 WKUITE 位为'1'，则产生一个状态改变中断。 该位由软件写 1 清'0'。
2	ERRINT	出错中断挂起（Error interrupt） 当检测到错误时，CAN_ESTS 寄存器的某位被置'1'，如果 CAN_INTE 寄存器的相应中断使能位也被置'1'时，则硬件对该位置'1'；如果 CAN_INTE 寄存器的 ERRITE 位为'1'，则产生状态改变中断。该位由软件写 1 清'0'。
1	SLPAK	睡眠模式确认 该位由硬件置'1'，指示软件 CAN 模块正处于睡眠模式。 该位是对软件请求进入睡眠模式的确认（对 CAN_MCTRL 寄存器的 SLPRQ 位置'1'）。 当 CAN 退出睡眠模式时硬件对该位清'0'（需要跟 CAN 总线同步）。这里跟 CAN 总线同步是指，硬件需要在 CAN 的 RX 引脚上检测到连续的 11 位隐性位。 *注：通过软件或硬件对 CAN_MCTRL 的 SLPRQ 位清'0'，将启动退出睡眠模式的过程。有关清除 SLPRQ 位的详细信息，参见 CAN_MCTRL 寄存器的 AWKUM 位的描述。*
0	INIAK	初始化确认 该位由硬件置'1'，指示软件 CAN 模块正处于初始化模式。该位是对软件请求进入初始化模式的确认（对 CAN_MCTRL 寄存器的 INIRQ 位置'1'）。 当 CAN 退出初始化模式时硬件对该位清'0'（需要跟 CAN 总线同步）。这里跟 CAN 总线同步是指，硬件需要在 CAN 的 RX 引脚上检测到连续的 11 位隐性位。

21.9.3.3 CAN 发送状态寄存器（CAN_TSTS）

地址偏移量：0x08

复位值：0x1C00 0000



31	30	29	28	27	26	25	24	23	22	20	19	18	17	16
LOWM2	LOWM1	LOWM0	TMEM2	TMEM1	TMEM0	CODE		ABRQM2	Reserved		TERRM2	ALSTM2	TXOKM2	RQCPM2
r	r	r	r	r	r	r		rs			rc_w1	rc_w1	rc_w1	rc_w1
15	14		12	11	10	9	8	7	6	4	3	2	1	0
ABRQM1	Reserved			TERRM1	ALSTM1	TXOKM1	RQCPM1	ABRQM0	Reserved		TERRM0	ALSTM0	TXOKM0	RQCPM0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs			rc_w1	rc_w1	rc_w1	rc_w1

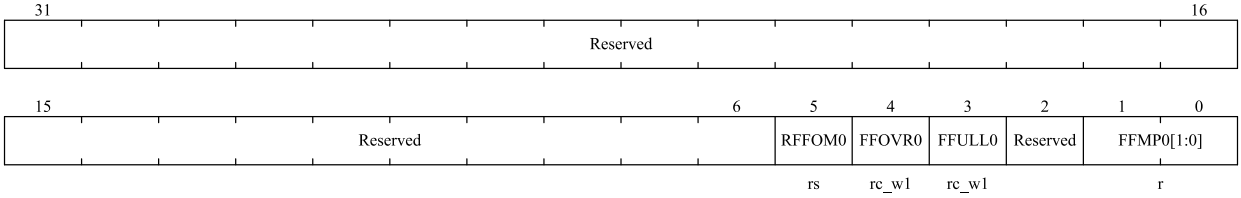
位域	名称	描述
31	LOWM2	邮箱 2 最低优先级标志（Lowest priority flag for mailbox 2） 当多个邮箱在等待发送报文，且邮箱 2 的优先级最低时，硬件对该位置'1'。
30	LOWM1	邮箱 1 最低优先级标志（Lowest priority flag for mailbox 1） 当多个邮箱在等待发送报文，且邮箱 1 的优先级最低时，硬件对该位置'1'。
29	LOWM0	邮箱 0 最低优先级标志（Lowest priority flag for mailbox 0） 当多个邮箱在等待发送报文，且邮箱 0 的优先级最低时，硬件对该位置'1'。*注：如果只有 1 个邮箱在等待，则 LOW[2:0]被清'0'。*
28	TMEM2	发送邮箱 2 空（Transmit mailbox 2 empty） 当邮箱 2 中没有等待发送的报文时，硬件对该位置'1'。
27	TMEM1	发送邮箱 1 空（Transmit mailbox 1 empty） 当邮箱 1 中没有等待发送的报文时，硬件对该位置'1'。
26	TMEM0	发送邮箱 0 空（Transmit mailbox 0 empty） 当邮箱 0 中没有等待发送的报文时，硬件对该位置'1'。
25:24	CODE[1:0]	邮箱号（Mailbox code） 当有至少 1 个发送邮箱为空时，这 2 位表示下一个空的发送邮箱号。 当所有的发送邮箱都为空时，这 2 位表示优先级最低的那个发送邮箱号。
23	ABRQM2	邮箱 2 中止发送（Abort request for mailbox 2） 软件对该位置'1'，可以中止邮箱 2 的发送请求，当邮箱 2 的发送报文被清除时硬件对该位清'0'。如果邮箱 2 中没有等待发送的报文，则对该位置'1'没有任何效果。
22:20	Reserved	保留，硬件强制为 0。
19	TERRM2	邮箱 2 发送失败（Transmission error of mailbox 2） 当邮箱 2 因为出错而导致发送失败时，对该位置'1'。
18	ALSTM2	邮箱 2 仲裁丢失（Arbitration lost for mailbox 2） 当邮箱 2 因为仲裁丢失而导致发送失败时，对该位置'1'。
17	TXOKM2	邮箱 2 发送成功（Transmission OK of mailbox 2） 每次在邮箱 2 进行发送尝试后，硬件对该位进行更新： 0：上次发送尝试失败； 1：上次发送尝试成功。 当邮箱 2 的发送请求被成功完成后，硬件对该位置'1'。请参见图 21-7。
16	RQCPM2	邮箱 2 请求完成（Request completed mailbox 2） 当上次对邮箱 2 的请求（发送或中止）完成后，硬件对该位置'1'。 软件对该位写'1'可以对其清'0'；当硬件接收到发送请求时也对该位清'0'（CAN_TMI2 寄存器的 TXRQ 位被置'1'）。

位域	名称	描述
		该位被清'0'时，邮箱 2 的其它发送状态位（TXOKM2, ALSTM2 和 TERRM2）也被清'0'。
15	ABRQM1	邮箱 1 中止发送（Abort request for mailbox 1） 软件对该位置'1'，可以中止邮箱 1 的发送请求，当邮箱 1 的发送报文被清除时硬件对该位清'0'。如果邮箱 1 中没有等待发送的报文，则对该位置'1'没有任何效果。
14:12	Reserved	保留，硬件强制为 0。
11	TERRM1	邮箱 1 发送失败（Transmission error of mailbox 1） 当邮箱 1 因为出错而导致发送失败时，对该位置'1'。
10	ALSTM1	邮箱 1 仲裁丢失（Arbitration lost for mailbox 1） 当邮箱 1 因为仲裁丢失而导致发送失败时，对该位置'1'。
9	TXOKM1	邮箱 1 发送成功（Transmission OK of mailbox 1） 每次在邮箱 1 进行发送尝试后，硬件对该位进行更新: 0：上次发送尝试失败; 1：上次发送尝试成功。 当邮箱 1 的发送请求被成功完成后，硬件对该位置'1'。请参见图 21-7。
8	RQCPM1	邮箱 1 请求完成（Request completed mailbox 1） 当上次对邮箱 1 的请求（发送或中止）完成后，硬件对该位置'1'。 软件对该位写'1'可以对其清'0'；当硬件接收到发送请求时也对该位清'0'（CAN_TMI1 寄存器的 TXRQ 位被置'1'）。 该位被清'0'时，邮箱 1 的其它发送状态位（TXOKM1, ALSTM1 和 TERRM1）也被清'0'。
7	ABRQM0	邮箱 0 中止发送（Abort request for mailbox 0） 软件对该位置'1'可以中止邮箱 0 的发送请求，当邮箱 0 的发送报文被清除时硬件对该位清'0'。如果邮箱 0 中没有等待发送的报文，则对该位置 1 没有任何效果。
6:4	Reserved	保留，硬件强制为 0。
3	TERRM0	邮箱 0 发送失败（Transmission error of mailbox 0） 当邮箱 0 因为出错而导致发送失败时，对该位置'1'。
2	ALSTM0	邮箱 0 仲裁丢失（Arbitration lost for mailbox 0） 当邮箱 0 因为仲裁丢失而导致发送失败时，对该位置'1'。
1	TXOKM0	邮箱 0 发送成功（Transmission OK of mailbox 0） 每次在邮箱 0 进行发送尝试后，硬件对该位进行更新: 0：上次发送尝试失败; 1：上次发送尝试成功。 当邮箱 0 的发送请求被成功完成后，硬件对该位置'1'。请参见图 21-7。
0	RQCPM0	邮箱 0 请求完成（Request completed mailbox 0） 当上次对邮箱 0 的请求（发送或中止）完成后，硬件对该位置'1'。 软件对该位写'1'可以对其清'0'；当硬件接收到发送请求时也对该位清'0'（CAN_TMI0 寄存器的 TXRQ 位被置'1'）。 该位被清'0'时，邮箱 0 的其它发送状态位（TXOKM0, ALSTM0 和 TERRM0）也被清'0'。

21.9.3.4 CAN 接收 FIFO 0 寄存器（CAN_RFF0）

地址偏移量：0x0C

复位值：0x0000 0000

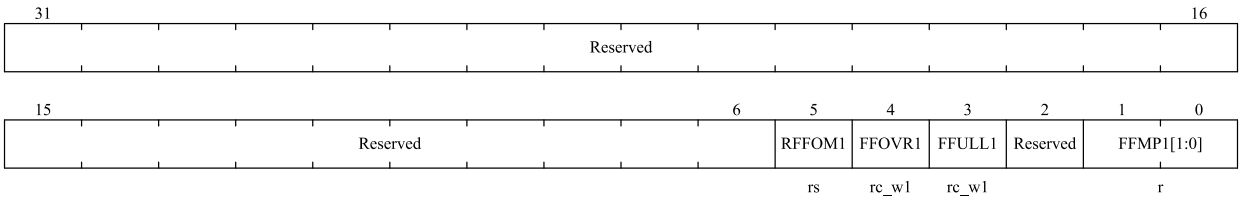


位域	名称	描述
31:6	Reserved	保留，硬件强制为 0。
5	RFFOM0	释放接收 FIFO 0 输出邮箱（Release FIFO 0 output mailbox） 软件通过对该位置'1'来释放接收 FIFO 的输出邮箱。如果接收 FIFO 为空，那么对该位置'1'没有任何效果，即只有当 FIFO 中有报文时对该位置'1'才有意义。如果 FIFO 中有 2 个以上的报文，由于 FIFO 的特点，软件需要释放输出邮箱才能访问第 2 个报文。 当输出邮箱被释放时，硬件对该位清'0'。
4	FFOVR0	FIFO 0 溢出（FIFO 0 overrun） 当 FIFO 0 已满，又收到新的报文且报文符合过滤条件，硬件对该位置'1'。该位由软件写 1 清'0'。
3	FFULL0	FIFO 0 满（FIFO 0 full） 当 FIFO 0 中有 3 个报文时，硬件对该位置'1'。该位由软件写 1 清'0'。
2	Reserved	保留，硬件强制为 0。
1:0	FFMP0[1:0]	FIFO 0 报文数目（FIFO 0 message pending） FIFO 0 报文数目这 2 位反映了当前接收 FIFO 0 中存放的报文数目。每当 1 个新的报文被存入接收 FIFO 0，硬件就对 FFMP0 加 1。 每当软件对 RFFOM0 位写'1'来释放输出邮箱，FFMP0 就被减 1，直到其为 0。

21.9.3.5 CAN 接收 FIFO 1 寄存器（CAN_RFF1）

地址偏移量：0x10

复位值：0x0000 0000



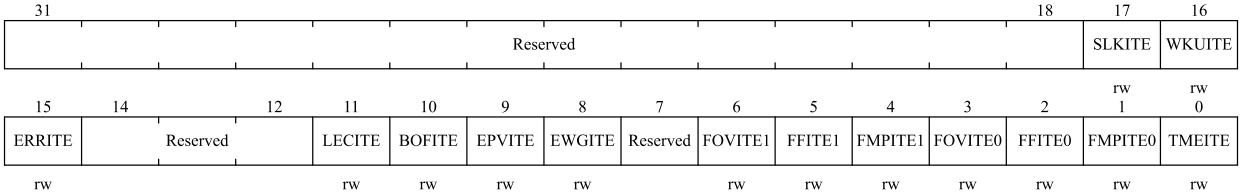
位域	名称	描述
31:6	Reserved	保留，硬件强制为 0。
5	RFFOM1	释放接收 FIFO 1 输出邮箱（Release FIFO 1 output mailbox） 软件通过对该位置'1'来释放接收 FIFO 的输出邮箱。如果接收 FIFO 为空，那么对该位置'1'没有任何效果，即只有当 FIFO 中有报文时对该位置'1'才有意义。如果 FIFO 中有 2 个以上的报文，由于 FIFO 的特点，软件需要释放输出邮箱才能访问第 2 个报文。

位域	名称	描述
		当输出邮箱被释放时，硬件对该位清'0'。
4	FFOVR1	FIFO 1 溢出（FIFO 1 overrun） 当 FIFO 1 已满，又收到新的报文且报文符合过滤条件，硬件对该位置'1'。该位由软件写 1 清'0'。
3	FFULL1	FIFO 1 满（FIFO 1 full） 当 FIFO 1 中有 3 个报文时，硬件对该位置'1'。该位由软件写 1 清'0'。
2	Reserved	保留，硬件强制为 0。
1:0	FFMP1[1:0]	FIFO 1 报文数目（FIFO 1 message pending） FIFO 1 报文数目这 2 位反映了当前接收 FIFO 1 中存放的报文数目。每当 1 个新的报文被存入接收 FIFO 1，硬件就对 FMP1 加 1。 每当软件对 RFFOM1 位写'1'来释放输出邮箱，FFMP1 就被减 1，直到其为 0。

21.9.3.6 CAN 中断使能寄存器（CAN_INTE）

地址偏移量：0x14

复位值：0x0000 0000



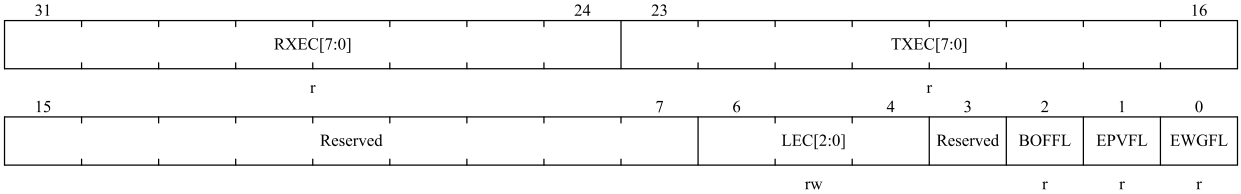
位域	名称	描述
31:18	Reserved	保留，硬件强制为 0。
17	SLKITE	睡眠中断使能（Sleep interrupt enable） 0：当 SLAKINT 位被置'1'时，不产生中断； 1：当 SLAKINT 位被置'1'时，产生中断。
16	WKUITE	唤醒中断使能（Wakeup interrupt enable） 0：当 WKUINT 位被置'1'时，不产生中断； 1：当 WKUINT 位被置'1'时，产生中断。
15	ERRITE	错误中断使能（Error interrupt enable） 0：当 CAN_ESTS 寄存器有错误挂起时，不产生中断； 1：当 CAN_ESTS 寄存器有错误挂起时，产生中断。
14:12	Reserved	保留，硬件强制为 0。
11	LECITE	上次错误号中断使能（Last error code interrupt enable） 0：当检测到错误，硬件设置 LEC[2:0]时，不设置 ERRINT 位； 1：当检测到错误，硬件设置 LEC[2:0]时，设置 ERRINT 位为'1'。
10	BOFITE	离线中断使能（Bus-off interrupt enable） 0：当 BOFFL 位被置'1'时，不设置 ERRINT 位； 1：当 BOFFL 位被置'1'时，设置 ERRINT 位为'1'。
9	EPVITE	错误被动中断使能（Error Passive Interrupt Enable） 0：当 EPVFL 位被置'1'时，不设置 ERRINT 位； 1：当 EPVFL 位被置'1'时，设置 ERRINT 位为'1'。

位域	名称	描述
8	EWGITE	错误警告中断使能（Error warning interrupt enable） 0：当 EWGFL 位被置'1'时，不设置 ERRINT 位； 1：当 EWGFL 位被置'1'时，设置 ERRINT 位为'1'。
7	Reserved	保留，硬件强制为 0。
6	FOVITE1	FIFO 1 溢出中断使能（FIFO overrun interrupt enable） 0：当 FIFO 1 的 FFOVR 位被置'1'时，不产生中断； 1：当 FIFO 1 的 FFOVR 位被置'1'时，产生中断。
5	FFITE1	FIFO 1 满中断使能（FIFO full interrupt enable） 0：当 FIFO 1 的 FFULL 位被置'1'时，不产生中断； 1：当 FIFO 1 的 FFULL 位被置'1'时，产生中断。
4	FMPITE1	FIFO 1 消息挂起中断使能（FIFO message pending interrupt enable） 0：当 FIFO 1 的 FFMP[1:0]位为非 0 时，不产生中断； 1：当 FIFO 1 的 FFMP[1:0]位为非 0 时，产生中断。
3	FOVITE0	FIFO 0 溢出中断使能（FIFO overrun interrupt enable） 0：当 FIFO 0 的 FFOVR 位被置'1'时，不产生中断； 1：当 FIFO 0 的 FFOVR 位被置'1'时，产生中断。
2	FFITE0	FIFO 0 满中断使能（FIFO full interrupt enable） 0：当 FIFO 0 的 FFULL 位被置’1’时，不产生中断； 1：当 FIFO 0 的 FFULL 位被置'1'时，产生中断。
1	FMPITE0	FIFO 0 消息挂起中断使能（FIFO message pending interrupt enable） 0：当 FIFO 0 的 FFMP[1:0]位为非 0 时，不产生中断； 1：当 FIFO 0 的 FFMP[1:0]位为非 0 时，产生中断。
0	TMEITE	发送邮箱空中断使能（Transmit mailbox empty interrupt enable） 0：当 RQCPMx 位被置'1'时，不产生中断； 1：当 RQCPMx 位被置'1'时，产生中断。 *注：请参考 21.8 节 bxCAN 中断。*

21.9.3.7 CAN 错误状态寄存器（CAN_ESTS）

地址偏移量：0x18

复位值：0x0000 0000



位域	名称	描述
31:24	RXEC[7:0]	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。按照 CAN 的标准，当接收出错时， 根据出错的条件，该计数器加 1 或加 8；而在每次接收成功后，该计数器减 1，或当该计数器的值大于 127 时，设置它的值为 120。当该计数器的值超过 127 时，CAN 进入错误被动状态。
23:16	TXEC[7:0]	9 位发送错误计数器的低 8 位（Least significant byte of the 9-bit

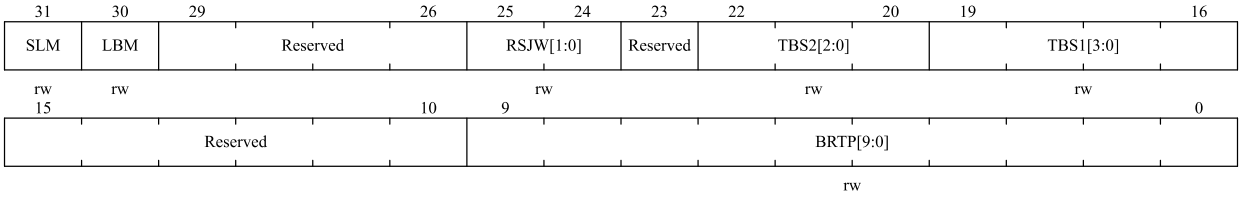
位域	名称	描述
		transmit error counter） 与上面相似，这个计数器按照 CAN 协议的故障界定机制的发送部分实现。
15:7	Reserved	保留，硬件强制为 0。
6:4	LEC[2:0]	上次错误代码（Last error code） 在检测到 CAN 总线上发生错误时，硬件根据出错情况设置。当 CAN 总线上未发生错误报文被正确发送或接收后，硬件设置此值为'0'。 软件可以设置此值为 0，通过判断此值是否变为 1~6 监测错误发生； 另外由于硬件没有使用且不会设置值 7，软件可以设置值 7，从而通过检查此位是否变为 0~6 判断接收报文动作的正确或错误类型。 000：没有错误； 001：位填充错； 010：格式（Form）错； 011：确认（ACK）错； 100：隐性位错； 101：显性位错； 110：CRC 错； 111：由软件设置。
3	Reserved	保留，硬件强制为 0。
2	BOFFL	离线标志（Bus-off flag） 当进入离线状态时，硬件对该位置'1'。当发送错误计数器 TXEC 溢出，即大于 255 时，CAN 进入离线状态。请参考 21.7.6。
1	EPVFL	错误被动标志（Error passive flag） 当出错次数达到错误被动的阈值时，硬件对该位置'1'。 （接收错误计数器或发送错误计数器的值>127）。
0	EWGFL	错误警告标志（Error warning flag） 当出错次数达到警告的阈值时，硬件对该位置'1'。 （接收错误计数器或发送错误计数器的值≥96）。

21.9.3.8 CAN 位时序寄存器（CAN_BTIM）

地址偏移量：0x1C

复位值：0x0123 0000

注：当 CAN 处于初始化模式时，该寄存器只能由软件访问。



位域	名称	描述
31	SLM	静默模式（用于调试）（Silent mode（debug）） 0：正常状态； 1：静默模式。

位域	名称	描述
30	LBM	环回模式（用于调试）（Loop back mode（debug）） 0：禁止环回模式； 1：允许环回模式。
29:26	Reserved	保留，硬件强制为 0。
25:24	RSJW[1:0]	重新同步跳跃宽度（Resynchronization jump width） 为了重新同步，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。 $t_{RJW} = t_{CAN} x$（RSJW[1:0] + 1）。
23	Reserved	保留，硬件强制为 0。
22:20	TBS2[2:0]	时间段 2（Time segment 2） 该位域定义了时间段 2 占用了多少个时间单元 $t_{BS2} = t_{CAN} x$（TBS2[2:0] + 1）。
19:16	TBS1[3:0]	时间段 1（Time segment 1） 该位域定义了时间段 1 占用了多少个时间单元 $t_{BS1} = t_{CAN} x$（TBS1[3:0] + 1） 关于位时间特性的详细信息，请参考 21.7.7 节位时间特性。
15:10	Reserved	保留，硬件强制为 0。
9:0	BRTP[9:0]	波特率分频器（Baud rate prescaler） 该位域定义了时间单元（tq）的时间长度 $t_q =$（BRTP[9:0]+1） $x\ t_{PCLK}$

21.9.4 CAN 邮箱寄存器

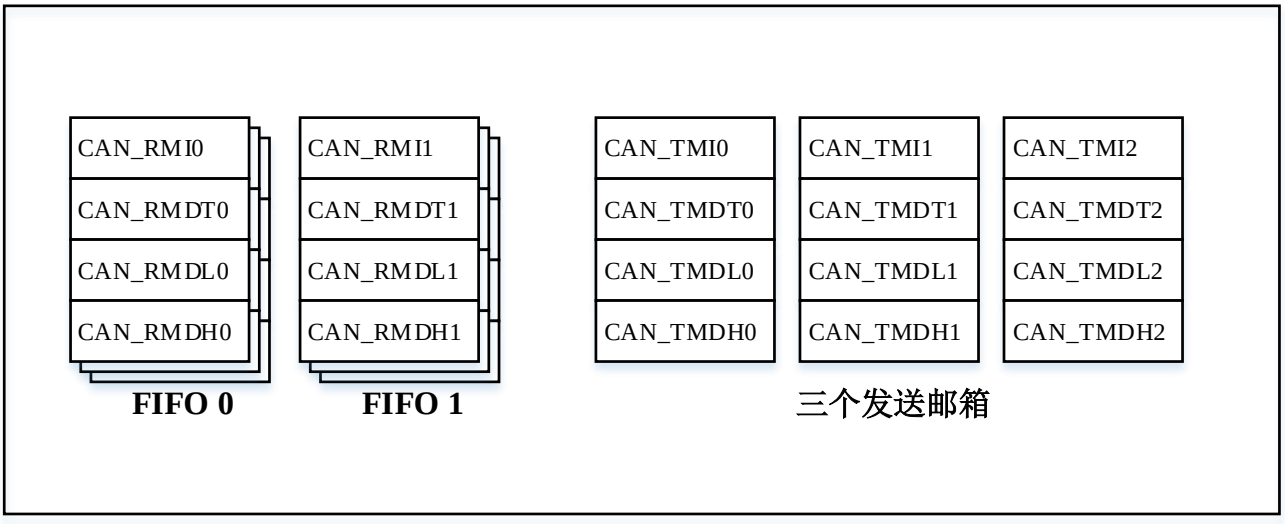
本节描述发送和接收邮箱寄存器。关于寄存器映像的详细信息，请参考 21.7.5 节报文存储。

除了下述例外，发送和接收邮箱几乎一样:

- CAN_RMDTx 寄存器的 FMI 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_TSTS 寄存器的相应 TMEM 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。



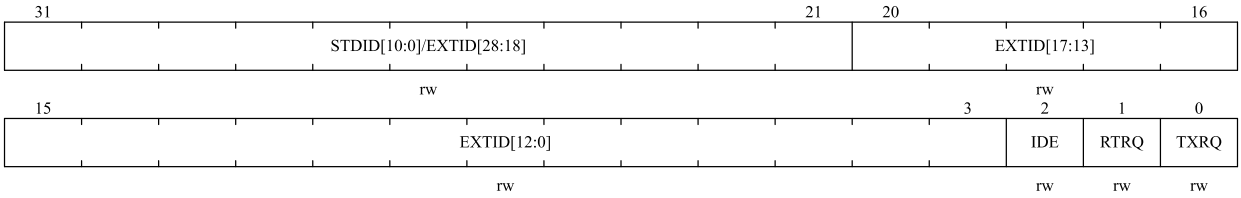
21.9.4.1 发送邮箱标识符寄存器（CAN_TMIx）（x=0..2）

地址偏移量：0x180，0x190，0x1A0

复位值：0xXXXX XXXX，X=未定义位（除了第 0 位，复位时 TXRQ=0）

注：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）－复位值为 0。



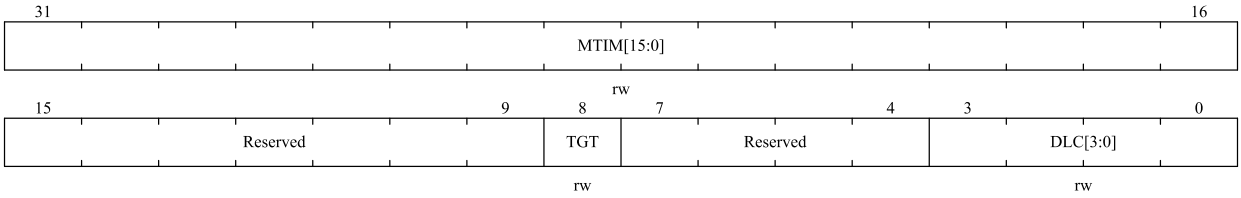
位域	名称	描述
31:21	STDID[10:0]/EXTID[28:18]	标准标识符或扩展标识符（Standard identifier or extended identifier） 依据 IDE 位的内容，这些位或是标准标识符，或是扩展身份标识的高字节。
20:3	EXTID[17:0]	扩展标识符（Extended identifier） 扩展身份标识的低字节。
2	IDE	标识符选择（Identifier extension） 该位决定发送邮箱中报文使用的标识符类型 0：使用标准标识符; 1：使用扩展标识符。
1	RTRQ	远程发送请求（Remote transmission request） 0：数据帧; 1：远程帧。
0	TXRQ	发送数据请求（Transmit mailbox request） 由软件对其置'1'，来请求发送邮箱的数据。当数据发送完成，邮箱为空时，硬件对其清'0'。

21.9.4.2 发送邮箱数据长度和时间戳寄存器（CAN_TMDTx）（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。

地址偏移量：0x184，0x194，0x1A4

复位值：未定义



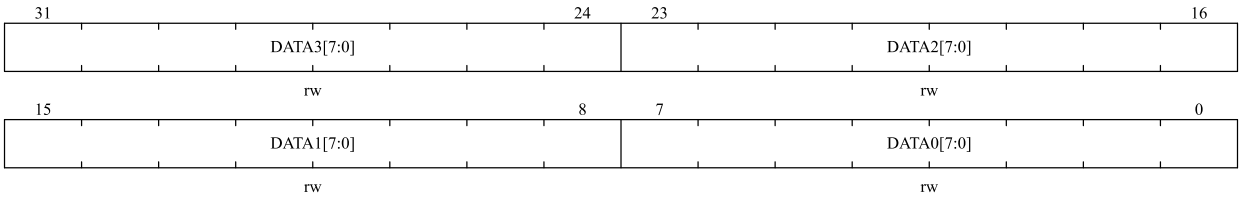
位域	名称	描述
31:16	MTIM[15:0]	报文时间戳（Message time stamp） 该域包含了，在发送该报文 SOF 的时刻，16 位定时器的值。
15:9	Reserved	保留，硬件强制为 0。
8	TGT	发送时间戳（Transmit global time） 只有在 CAN 处于时间触发通信模式，即 CAN_MCTRL 寄存器的 TTCM 位为'1'时，该位才有效。 0：不发送时间戳 MTIM[15:0]; 1：发送时间戳 MTIM[15:0]。在长度为 8 的报文中，时间戳 MTIM[15:0]是最后 2 个发送的字节: MTIM[7:0]作为第 7 个字节，MTIM[15:8]为第 8 个字节，它们替换了写入 CAN_TMDHx[31:16]的数据（DATA6[7:0]和 DATA7[7:0]）。为了把时间戳的 2 个字节发送出去，DLC 必须编程为 8。
7:4	Reserved	保留，硬件强制为 0。
3:0	DLC[3:0]	发送数据长度（Data length code） 该域指定了数据报文的数据长度或者远程帧请求的数据长度。1 个报文包含 0 到 8 个字节数据， 而这由 DLC 决定。

21.9.4.3 发送邮箱低字节数据寄存器（CAN_TMDLx）（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。

地址偏移量：0x188，0x198，0x1A8

复位值：未定义



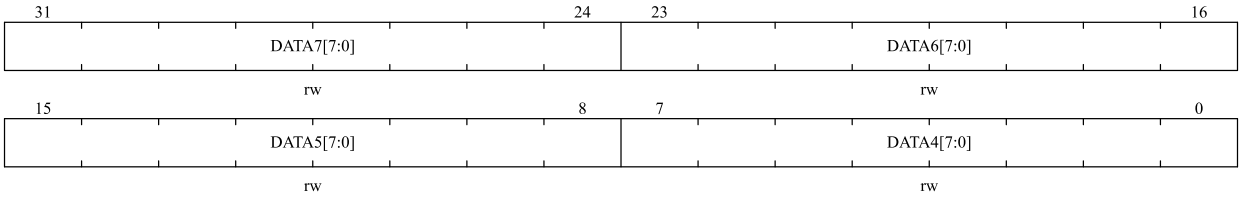
位域	名称	描述
31:24	DATA3[7:0]	数据字节 3（Data byte 3） 报文的数据字节 3。
23:16	DATA2[7:0]	数据字节 2（Data byte 2） 报文的数据字节 2。
15:8	DATA1[7:0]	数据字节 1（Data byte 1）

位域	名称	描述
		报文的数据字节 1。
7:0	DATA0[7:0]	数据字节 0（Data byte 0） 报文的数据字节 0。 报文包含 0 到 8 个字节数据，且从字节 0 开始。

21.9.4.4 发送邮箱高字节数据寄存器（CAN_TMDHx）（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。

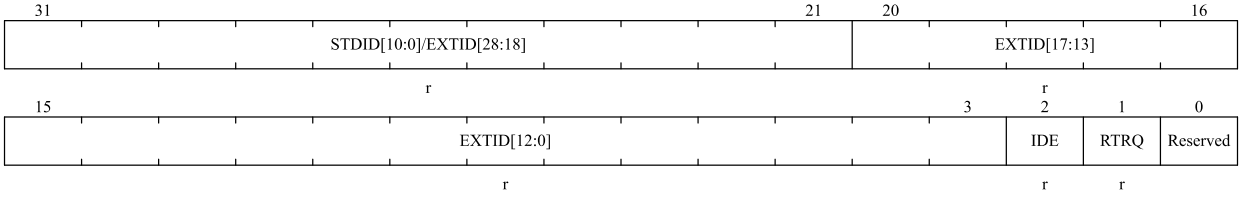
地址偏移量：0x18C，0x19C，0x1AC
复位值：未定义



位域	名称	描述
31:24	DATA7[7:0]	数据字节 7（Data byte 7） 报文的数据字节 7。 *注：如果 CAN_MCR 寄存器的 TTCM 位为'1'，且该邮箱的 TGT 位也为'1'，那么 DATA7 和 DATA6 将被 TIME 时间戳代替。*
23:16	DATA6[7:0]	数据字节 6（Data byte 6） 报文的数据字节 6。
15:8	DATA5[7:0]	数据字节 5（Data byte 5） 报文的数据字节 5。
7:0	DATA4[7:0]	数据字节 4（Data byte 4） 报文的数据字节 4。

21.9.4.5 接收 FIFO 邮箱标识符寄存器（CAN_RMIx）（x=0..1）

地址偏移量：0x1B0，0x1C0
复位值：未定义
注：所有接收邮箱寄存器都是只读的。



位域	名称	描述
31:21	STDID[10:0]/EXTID[28:18]	标准标识符或扩展标识符（Standard identifier or extended identifier） 依据 IDE 位的内容，这些位或是标准标识符，或是扩展身份标识的高字节。
20:3	EXTID[17:0]	扩展标识符（Extended identifier） 扩展身份标识的低字节。
2	IDE	标识符选择（Identifier extension）

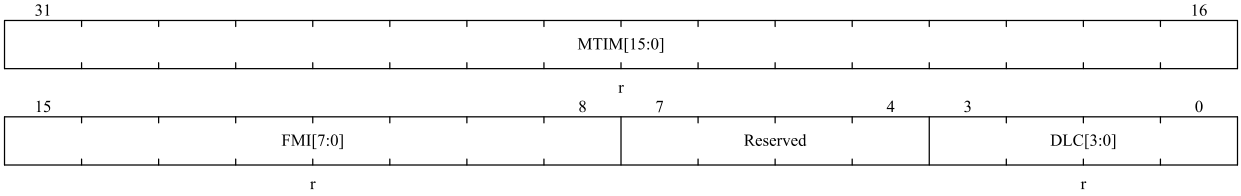
位域	名称	描述
		该位决定发送邮箱中报文使用的标识符类型 0：使用标准标识符； 1：使用扩展标识符。
1	RTRQ	远程发送请求（Remote transmission request） 0：数据帧； 1：远程帧。
0	Reserved	保留，硬件强制为 0。

21.9.4.6 接收 FIFO 邮箱数据长度和时间戳寄存器（CAN_RMDTx）（x=0..1）

地址偏移量：0x1B4，0x1C4

复位值：未定义

注：所有接收邮箱寄存器都是只读的。



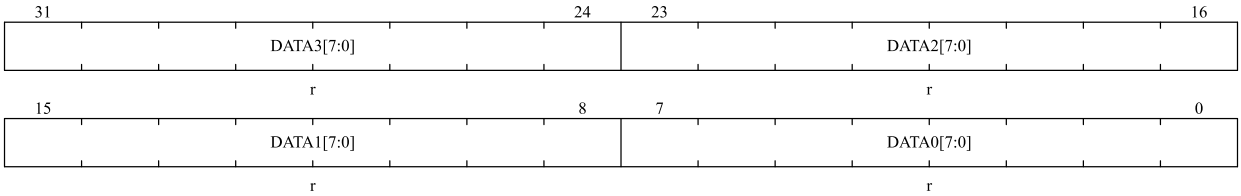
位域	名称	描述
31:16	MTIM[15:0]	报文时间戳（Message time stamp） 该域包含了，在发送该报文 SOF 的时刻，16 位定时器的值。
15:8	FMI[7:0]	过滤器匹配序号（Filter match index） 这里是存在邮箱中的信息传送的过滤器序号。关于标识符过滤的细节，请参考 21.7.4 中有关过滤器匹配序号。
7:4	Reserved	保留，硬件强制为 0。
3:0	DLC[3:0]	接收数据长度（Data length code） 该域表明接收数据帧的数据长度（0～8）。对于远程帧请求，数据长度 DLC 恒为 0。

21.9.4.7 接收 FIFO 邮箱低字节数据寄存器（CAN_RMDLx）（x=0..1）

地址偏移量：0x1B8，0x1C8

复位值：未定义

注：所有接收邮箱寄存器都是只读的。



位域	名称	描述
31:24	DATA3[7:0]	数据字节 3（Data byte 3） 报文的数据字节 3。
23:16	DATA2[7:0]	数据字节 2（Data byte 2） 报文的数据字节 2。

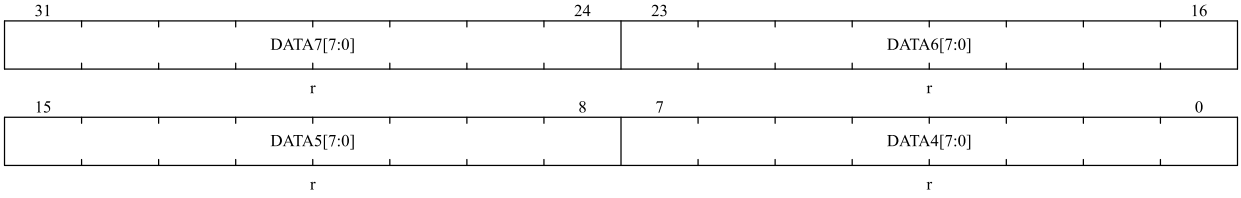
位域	名称	描述
15:8	DATA1[7:0]	数据字节 1（Data byte 1） 报文的数据字节 1。
7:0	DATA0[7:0]	数据字节 0（Data byte 0） 报文的数据字节 0。 报文包含 0 到 8 个字节数据，且从字节 0 开始。

21.9.4.8 接收 FIFO 邮箱高字节数据寄存器（CAN_RMDHx）（x=0..1）

地址偏移量：0x1BC，0x1CC

复位值：未定义

注：所有接收邮箱寄存器都是只读的。



位域	名称	描述
31:24	DATA7[7:0]	数据字节 7（Data byte 7） 报文的数据字节 7。
23:16	DATA6[7:0]	数据字节 6（Data byte 6） 报文的数据字节 6。
15:8	DATA5[7:0]	数据字节 5（Data byte 5） 报文的数据字节 5。
7:0	DATA4[7:0]	数据字节 4（Data byte 4） 报文的数据字节 4。

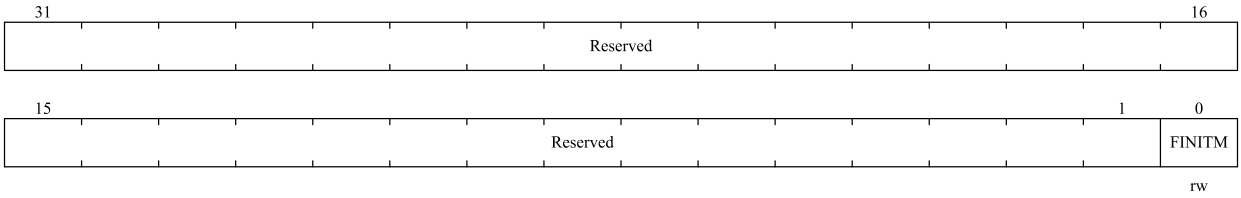
21.9.5 CAN 过滤器寄存器

21.9.5.1 CAN 过滤器主控制寄存器（CAN_FMC）

地址偏移量：0x200

复位值：0x2A1C 0E01

注：该寄存器的非保留位完全由软件控制。



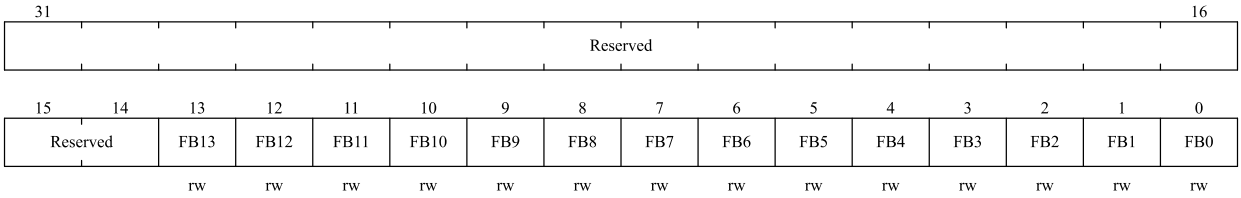
位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	FINITM	过滤器初始化模式（Filter init mode） 针对所有过滤器组的初始化模式设置。 0：过滤器组工作在正常模式； 1：过滤器组工作在初始化模式。

21.9.5.2 CAN 过滤器模式寄存器（CAN_FM1）

地址偏移量：0x204

复位值：0x0000 0000

注：只有在设置 CAN_FMC（FINITM=1），使过滤器处于初始化模式下，才能对该寄存器写入。



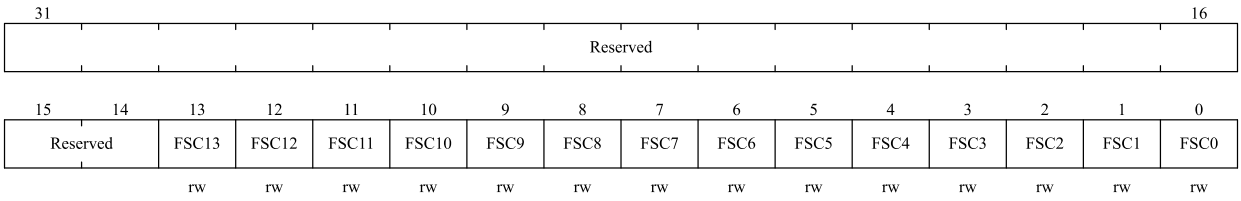
位域	名称	描述
31:28	Reserved	保留，硬件强制为 0。
13:0	FBx	过滤器模式（Filter mode） 过滤器组 x 的工作模式。 0：过滤器组 x 的 2 个 32 位寄存器工作在标识符屏蔽位模式； 1：过滤器组 x 的 2 个 32 位寄存器工作在标识符列表模式。

21.9.5.3 CAN 过滤器模式寄存器（CAN_FS1）

地址偏移量：0x20C

复位值：0x0000 0000

注：只有在设置 CAN_FMC（FINITM=1），使过滤器处于初始化模式下，才能对该寄存器写入。



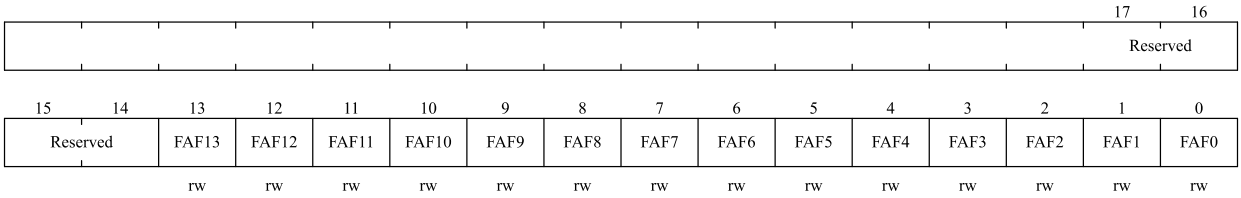
位域	名称	描述
31:28	Reserved	保留，硬件强制为 0。
13:0	FSCx	过滤器位宽设置（Filter scale configuration） 过滤器组 x（13～0）的位宽。 0：过滤器位宽为 2 个 16 位； 1：过滤器位宽为单个 32 位。

21.9.5.4 CAN 过滤器 FIFO 关联寄存器（CAN_FFA1）

地址偏移量：0x214

复位值：0x0000 0000

注：只有在设置 CAN_FMC（FINITM=1），使过滤器处于初始化模式下，才能对该寄存器写入。



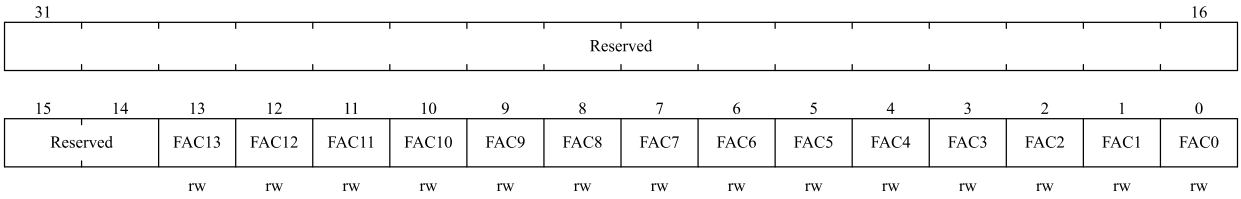
位域	名称	描述
31:28	Reserved	保留，硬件强制为 0。

位域	名称	描述
13:0	FAFx	过滤器位宽设置（Filter FIFO assignment for filter x） 报文在通过了某过滤器的过滤后，将被存放到其关联的 FIFO 中。 0：过滤器被关联到 FIFO0； 1：过滤器被关联到 FIFO1。

21.9.5.5 CAN 过滤器激活寄存器（CAN_FA1）

地址偏移量：0x21C

复位值：0x0000 0000



位域	名称	描述
31:28	Reserved	保留，硬件强制为 0。
13:0	FACx	过滤器激活（Filter active） 软件对某位设置'1'来激活相应的过滤器。只有对 FACx 位清'0'，或对 CAN_FMC 寄存器的 FINITM 位设置'1'后，才能修改相应的过滤器寄存器 x（CAN_FxB[0:1]）。 0：过滤器被禁用； 1：过滤器被激活。

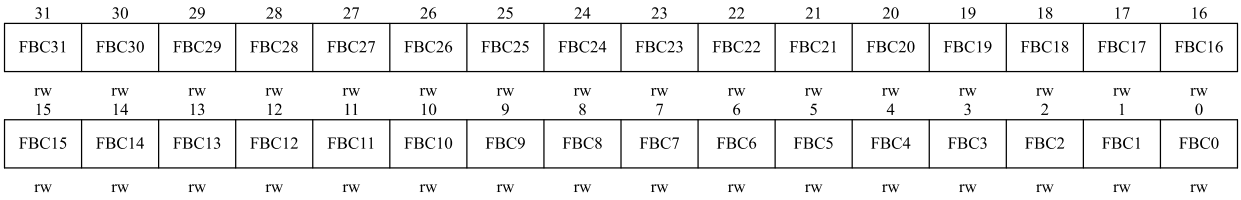
21.9.5.6 CAN 过滤器 i 的寄存器 x（CAN_FiRx）（i＝0..13； x=1..2）

地址偏移量：0x240h..0x31C

复位值：未定义

注：14 组过滤器：i=0..13。每组过滤 器由 2 个 32 位的寄存器，CAN_FiR[2:1]组成。

只有在 CAN_FAxR 寄存器相应的 FACTx 位清'0'，或 CAN_FMR 寄存器的 FINIT 位为'1'时，才能修改相应的过滤器寄存器。



位域	名称	描述
31:0	FBC[31:0]	过滤器位（Filter bits） 标识符模式 寄存器的每位对应于所期望的标识符的相应位的电平。 0：期望相应位为显性位； 1：期望相应位为隐性位。 屏蔽位模式 寄存器的每位指示是否对应的标识符寄存器位一定要与期望的标识符的相应位一致。

位域	名称	描述
		0：不关心，该位不用于比较； 1：必须匹配，到来的标识符位必须与滤波器对应的标识符寄存器位相一致。

注：根据过滤器位宽和模式的不同设置，过滤器组中的两个寄存器的功能也不尽相同。关于过滤器 的映射，功能描述和屏蔽寄存器的关联，请参见 21.7.4 节标识符过滤。

屏蔽位模式下*的屏蔽/标识符寄存器，跟****标识符列表模式****下的寄存器位定义相同。*

*关于过滤器组寄存器的地址，请参见*表 21-3。

22 串行外设接口（SPI）

22.1 SPI 简介

本模块中 SPI/I2S 接口复用，默认工作在 SPI 模式，可通过配置切换到 I2S 模式。二者都是同步串行接口通讯协议。串行外设接口（SPI）可工作于主机或从机模式，支持全双工、单工高速通信模式，具有硬件 CRC 计算并可配置多主模式。片上音频接口（I2S）在单工通讯中可在主、从两种模式运行，支持飞利浦 I2S 标准、MSB 对齐标准、LSB 对齐标准和 PCM 四种音频标准。

22.2 SPI 和 I2S 主要特征

22.2.1 SPI 特征

- 全双工和单工同步传输
- 支持主模式、从模式、多主模式
- 8 或 16 位数据帧格式
- 数据位顺序可编程
- 软件或硬件进行 NSS 管理
- 时钟极性和相位可编程
- 发送和接收支持硬件 CRC 计算、校验
- 支持 DMA

22.2.2 I2S 功能

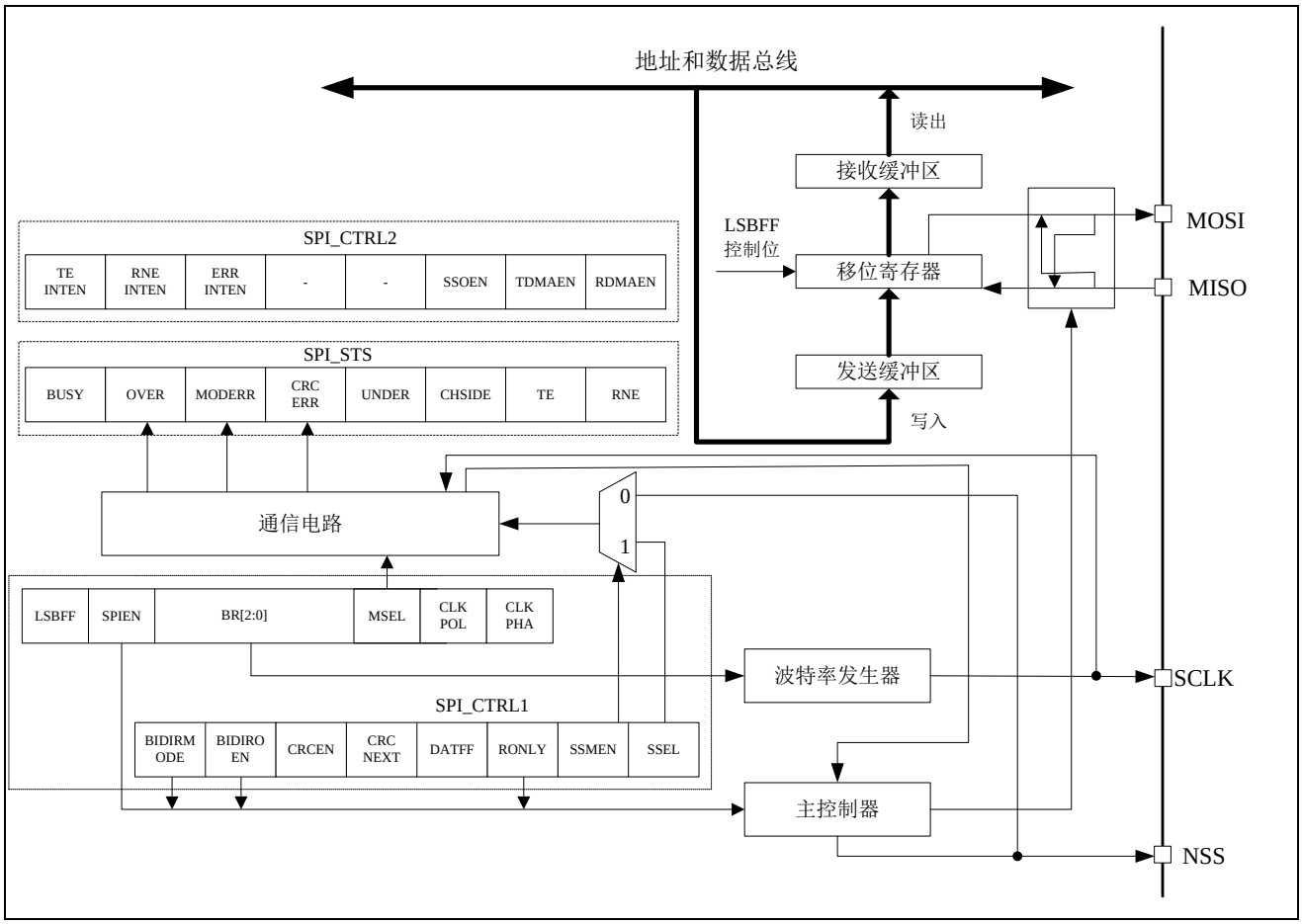
- 单工同步传输
- 支持主模式、从模式
- 支持四种音频标准：飞利浦 I2S 标准、MSB 对齐标准、LSB 对齐标准和 PCM 标准
- 可配置 8KHz 到 96KHz 的音频采样频率
- 支持 16 位，24 位或 32 位数据长度、数据帧格式（根据需求配置）
- 稳定态时钟极性可编程
- 数据方向总是 MSB 在先
- 支持 DMA

22.3 SPI 功能描述

22.3.1 概述

SPI 结构框图。

图 22-1 SPI 框图



SPI 共 4 个引脚与外部器件相连：

- SCLK：串口时钟，作为主设备的输出，从设备的输入
- MISO：主设备输入/从设备输出管脚。在主模式下接收数据，从模式下发送数据。
- MOSI：主设备输出/从设备输入管脚。在主模式下发送数据，从模式下接收数据。
- NSS：片选管脚。分为内部管脚和外部管脚，内部管脚检测到高电平时，SPI 工作在主模式；反之，工作在从模式。从设备的 NSS 管脚可以由主设备的一个标准 I/O 引脚来控制。

软件 NSS 模式

SPI_CTRL1 寄存器的 SSMEN=1 启用软件从设备管理（见图 22-2）。

该模式下不使用 NSS 管脚，通过写 SPI_CTRL1 的 SSEL 位驱动内部 NSS 信号电平（主模式 SSEL=1，

从模式 SSEL=0)。

硬件 NSS 模式

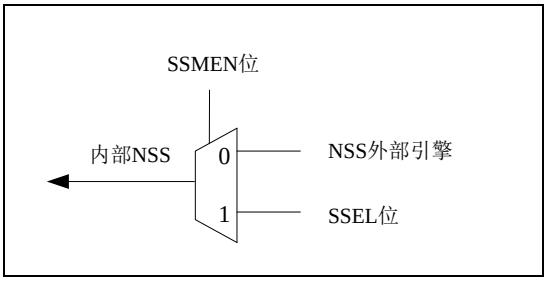
SPI_CTRL1 寄存器的 SSMEN=0 禁止软件从设备管理。

输入模式：配置为主模式、NSS 输出禁止（MSEL=1， SSOEN=0），允许操作于多主模式。主设备（从设备）在整个数据帧传输期间应把 NSS 脚外接到高电平（低电平）。

输出模式：配置为主模式、NSS 输出使能（MSEL=1，SSOEN=1），SPI 作为主设备必须将 NSS 拉低，所有被设置为 NSS 硬件模式并与之相连的的 SPI 设备会检测到低电平，自动进入从设备状态。若主设备不能拉低 NSS，则进入从模式，并产生主模式失效错误 MODERR 位置'1'。

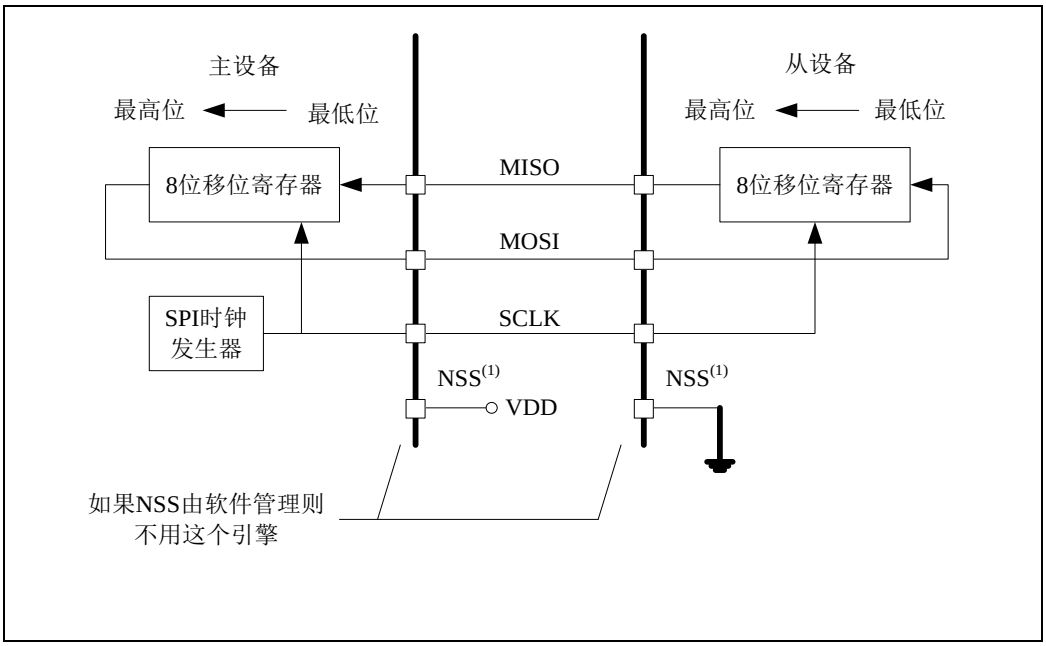
注：软件模式或硬件模式的选择，取决于通讯协议中是否需要 NSS 控制。如果不需要，可以选择软件模式，释放一个 GPIO 管脚另作他用。

图 22-2 硬件/软件的从选择管理



下图是一个单主和单从设备互连的例子。

图 22-3 单主和单从应用



注：1. NSS 引脚设置为输入

SPI 是一个环形总线结构，主设备通过 SCLK 管脚输出同步时钟信号，MOSI、MISO 管脚对应连接。主从之间串行传输数据，通过 MOSI 脚把数据发送给从设备并存在最低位，同时从设备的最高位通过

MISO 管脚传输到主设备的最低位，当第二位数据进行发送时，最低位的数据会向左移一位并将新数据存入最低位。

SPI 时序模式

CLKPOL 时钟极性和 CLKPHA 时钟相位的组合选择数据捕捉的时钟边沿。配置 SPI_CTRL1 寄存器的 CLKPOL 和 CLKPHA 位，有以下四种时序关系。

CLKPOL=0，CLKPHA=0：SCLK 引脚在空闲状态保持低电平，数据采样时在第一个边沿，即上升沿；

CLKPOL=0，CLKPHA=1：SCLK 引脚在空闲状态保持低电平，数据采样时在第二个边沿，即下降沿；

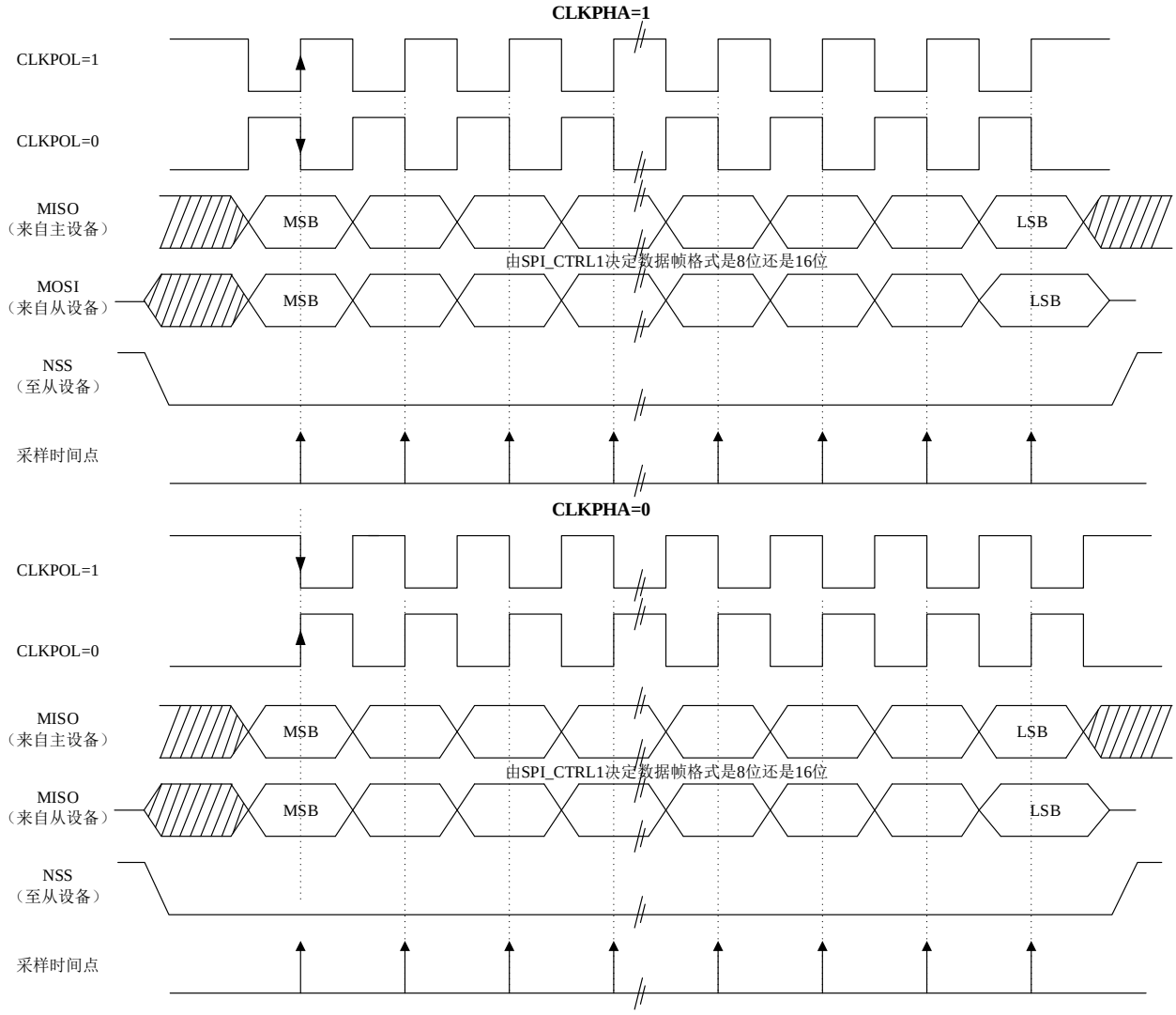
CLKPOL=1，CLKPHA=0：SCLK 引脚在空闲状态保持高电平，数据采样时在第一个边沿，即下降沿；

CLKPOL=1，CLKPHA=1：SCLK 引脚在空闲状态保持高电平，数据采样时在第二个边沿，即上升沿。

无论采用何种时序模式，必须保证主从配置相同。

图 22-4 是 SPI_CTRL1 寄存器 LSBFF=0 时，SPI 传输的 4 种 CLKPHA 和 CLKPOL 位组合时序。

图 22-4 数据时钟时序图



数据格式

配置 SPI_CTRL1 寄存器中的 LSBFF 位改变数据顺序，LSBFF=0 时，SPI 先发送高位数据（MSB）；LSBFF=1 时，SPI 先发送低位数据（LSB）。

配置 SPI_CTRL1 寄存器的 DATFF 位选择 8 位或 16 位数据帧。

22.3.2 SPI 工作模式

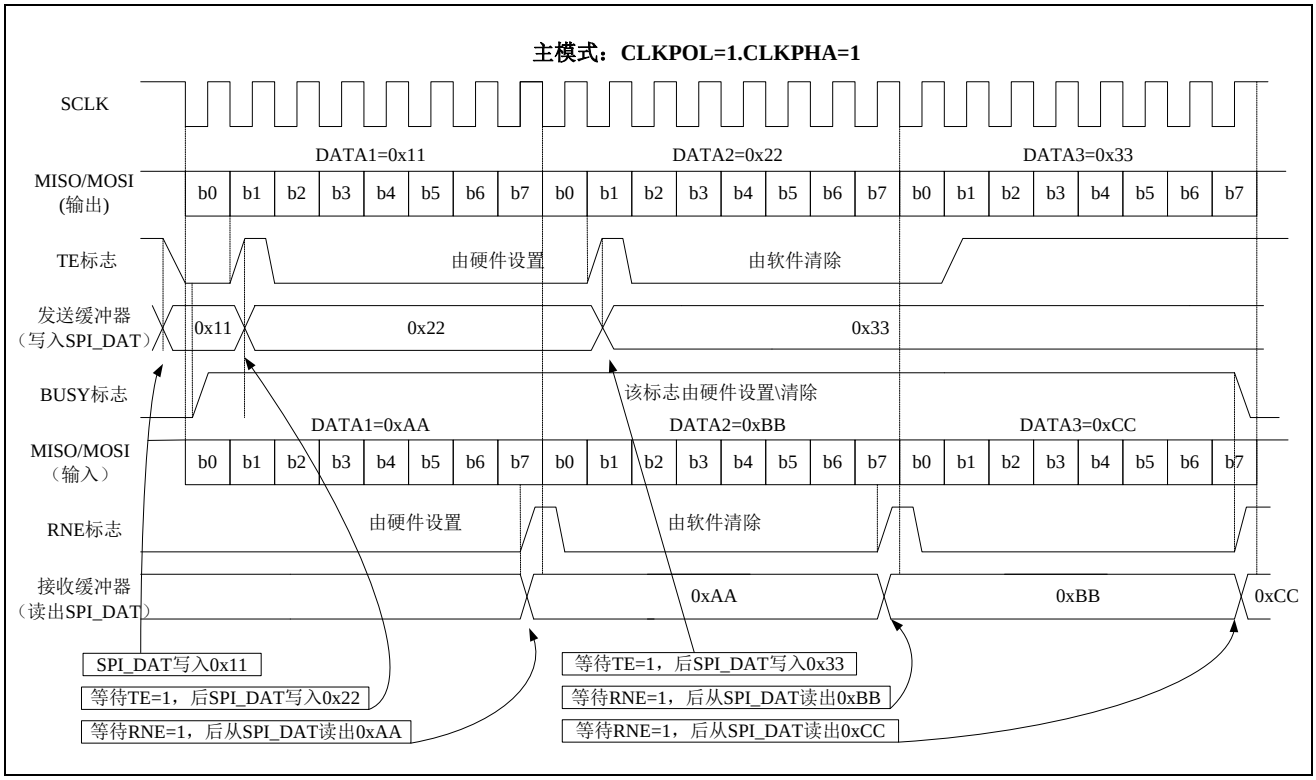
- **主机全双工模式（MSEL=1，BIDIRMODE=0，RONLY=0）**

寄存器 SPI_DAT（发送缓冲器）写入数据后，传输过程开始。在发送第一位数据的同时，数据被并行地从发送缓冲器传送到 8 位的移位寄存器中，SPI 根据 LSBFF 的配置按顺序将数据串行地移位到 MOSI 管脚上；同时，在 MISO 管脚上接收到的数据，按相同顺序被串行地移位进入 8 位的移位寄存器中，然后被并行地传送到 SPI_DAT 寄存器（接收缓冲器）中。软件操作流程如下（见图 22-5 主机全双工模式下连续传输时，TE/RNE/BUSY 的变化示意图）：

1. 使能SPI模块，SPIEN置‘1’；
2. 将第一个待发送数据写入SPI_DAT寄存器（该操作将标志位TE清零）；
3. 等待TE置‘1’，将第二个待发送数据写入SPI_DAT。等待RNE置‘1’，读取SPI_DAT得到第一个接收到的数据，读SPI_DAT的同时RNE位清零。重复以上操作，发送后续的数据同时接收n-1个数据；
4. 等待RNE置‘1’，接收最后一个数据；
5. 等待TE置‘1’，然后等待BUSY清零，关闭SPI模块。

数据收发的过程也可以在 RNE 或 TE 标志上升沿产生的中断处理程序中实现。

图 22-5 主机全双工模式下连续传输时，TE/RNE/BUSY 的变化示意图



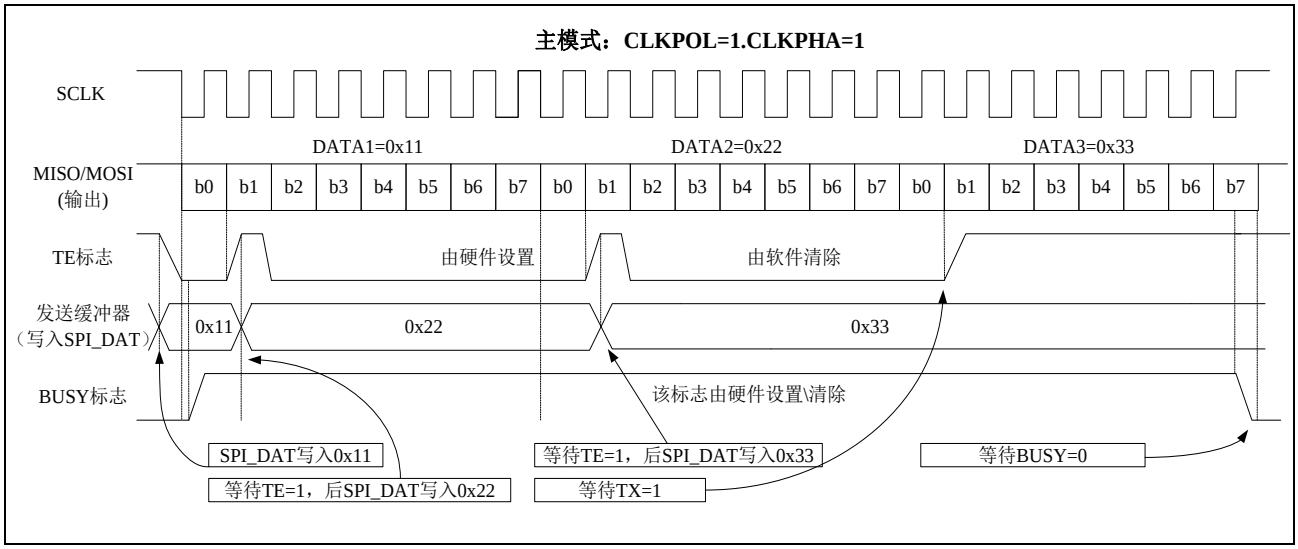
■ 主机单向只发送模式（MSEL=1，BIDIRMODE=0，RONLY=0）

单向只发送模式的传输原理同全双工模式。不同的是，该模式不会读取接收到的数据，故 SPI_STS 寄存器中的 OVER 位会置'1'，软件无需理会。软件操作流程如下（见图 22-6 主机单向只发送模式下连续传输时，TE/BUSY 变化示意图）：

1. 使能SPI模块，SPIEN置'1'；
2. 将第一个待发送数据写入 SPI_DAT 寄存器（该操作将标志位 TE 清零）；
3. 等待 TE 置'1'，将第二个待发送数据写入 SPI_DAT。重复这个操作，发送后续的数据；
4. 写入最后一个数据到 SPI_DAT 之后，等待 TE 置'1'；然后等待 BUSY 清零，完成所有数据的发送。

   数据发送的过程也可以在 TE 标志上升沿产生的中断处理程序中实现。

图 22-6 主机单向只发送模式下连续传输时，TE/BUSY 变化示意图



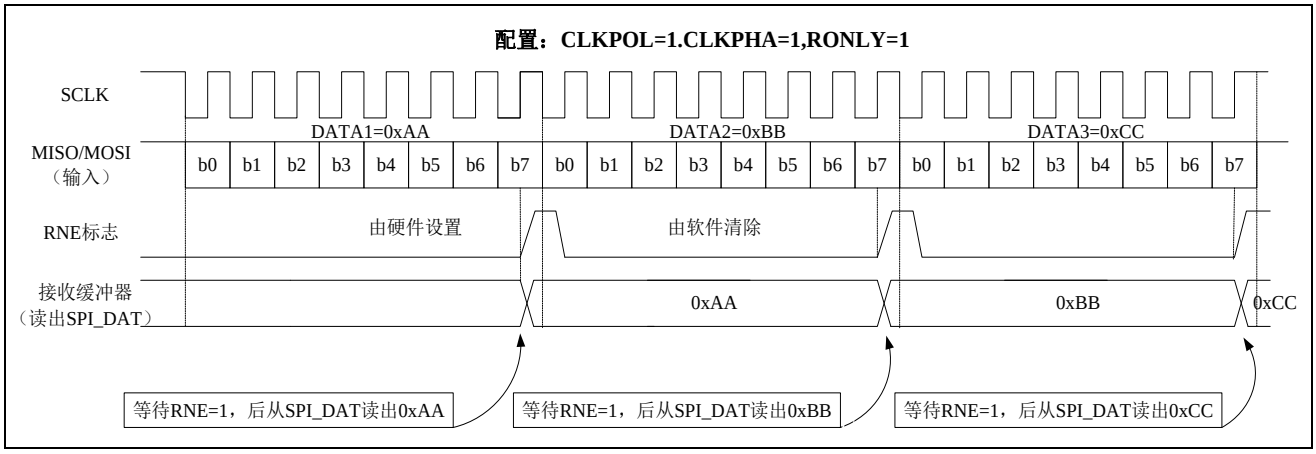
■ 主机单向只接收模式（MSEL=1，BIDIRMODE=0，RONLY=1）

SPIEN=1 时，接收过程开始。在 MISO 引脚上接收到的数据，按顺序被串行地移位进入 8 位的移位寄存器中，然后被并行地传送到 SPI_DAT 寄存器（接收缓冲器）中。软件操作流程如下（见图 22-7 只接收模式（BIDIRMODE=0 并且 RONLY=1）下连续传输时，RNE 变化示意图）：

1. 使能只接收模式（SPI_CTRL2 寄存器 RONLY=1）；
2. 使能 SPI 模块，SPIEN 置‘1’：主模式下，立刻产生 SCLK 时钟信号，在关闭 SPI（SPIEN=0）之前，不断地接收串行数据；从模式下，当 SPI 主设备拉低 NSS 信号并产生 SCLK 时钟时，接收串行数据。
3. 等待 RNE 置‘1’，读取 SPI_DAT 寄存器得到接收到的数据，读 SPI_DAT 的同时 RNE 位清零。重复这个操作接收所有数据。

   数据发送的过程也可以在 RNE 标志上升沿产生的中断处理程序中实现。

图 22-7 只接收模式（BIDIRMODE=0 并且 RONLY=1）下连续传输时，RNE 变化示意图



■ 主机双向发送模式（MSEL=1，BIDIRMODE=1，BIDIROEN=1，RONLY=0）

寄存器 SPI_DAT（发送缓冲器）写入数据后，发送过程开始。该模式不接收数据，在发送第一位数据的同时，数据被并行地从发送缓冲器传送到 8 位的移位寄存器中，SPI 根据 LSBFF 的配置按顺序将数据串行地移位到 MOSI 管脚上。

主机双向发送模式的软件操作流程同只发送模式。

- **主机双向接收模式（MSEL=1，BIDIRMODE=1，BIDIROEN=0，RONLY=0）**

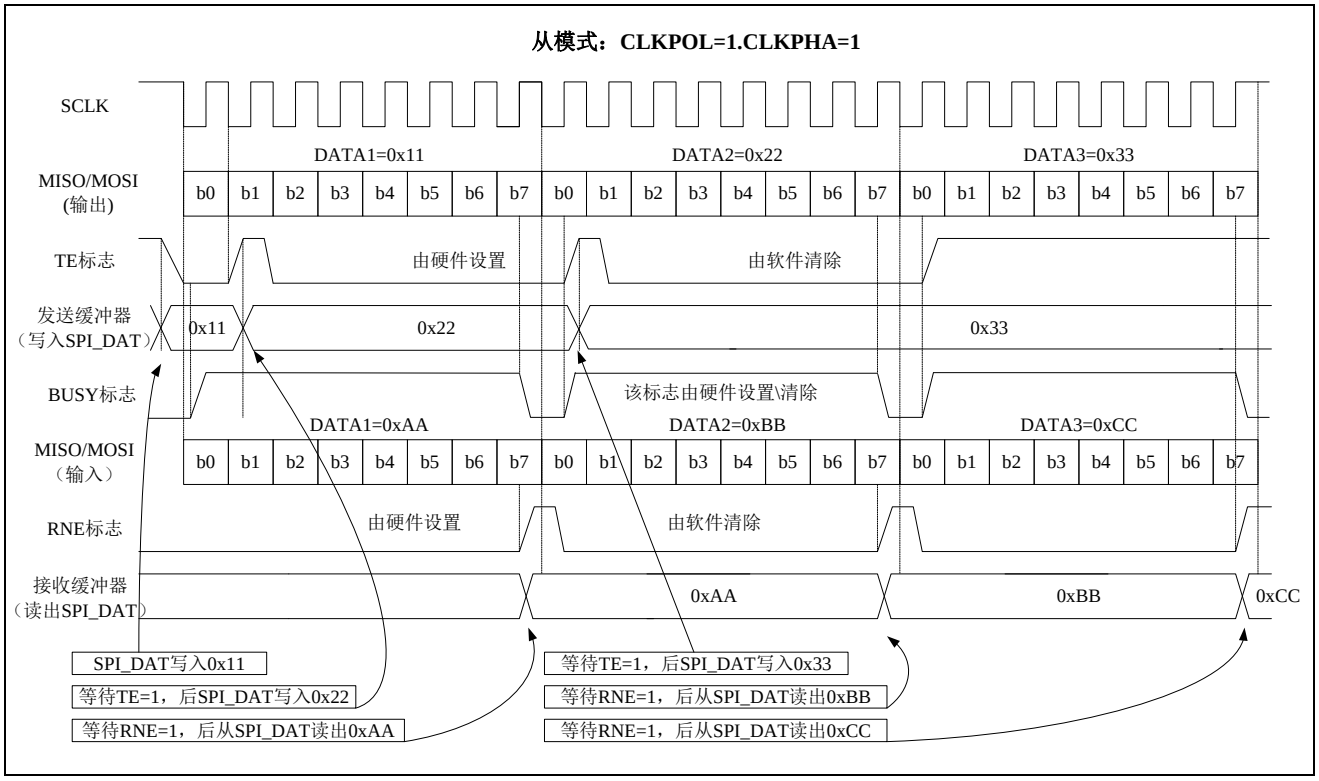
SPIEN=1、BIDIROEN=0 时，接收过程开始。该模式下无数据输出，接收到的数据按顺序被串行地移位进入 8 位的移位寄存器中，然后被并行地传送到 SPI_DAT 寄存器（接收缓冲器）中。

主机双向接收模式的软件操作流程同只接收模式。

- **从机全双工模式（MSEL=0，BIDIRMODE=0 并且 RONLY=0）**

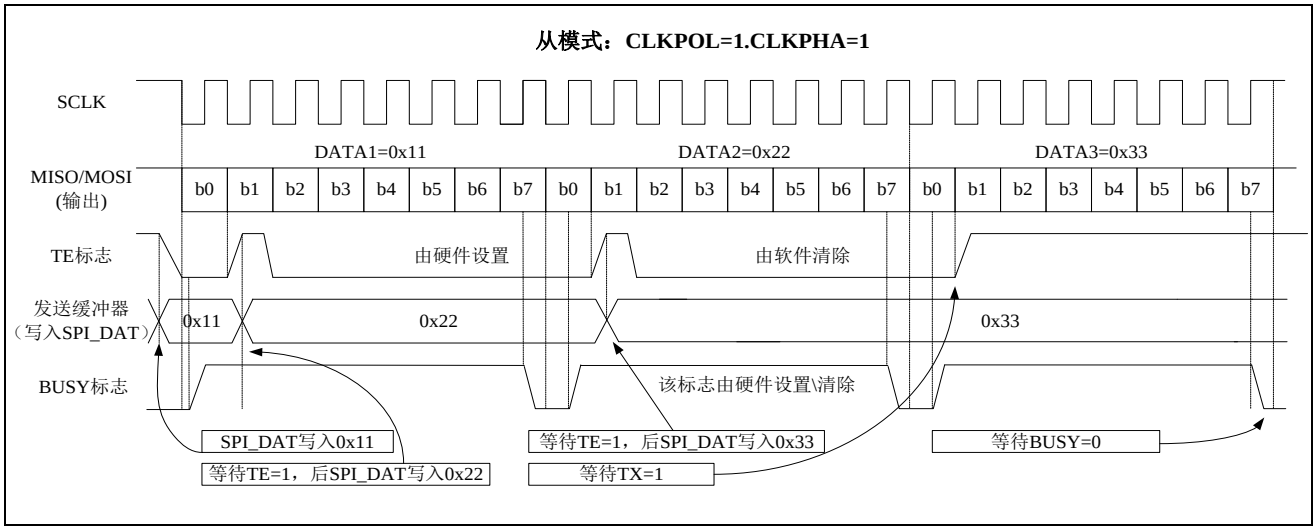
当从设备接收到时钟信号第一个边沿时，发送过程开始。软件必须保证在 SPI 主设备开始数据传输之前寄存器 SPI_DAT 中已写入待发送的数据。

图 22-8 从机全双工模式下连续传输时，TE/RNE/BUSY 的变化示意图



- **从机单向只发送模式（MSEL=0，BIDIRMODE=0 并且 RONLY=0）**

图 22-9 从机单向只发送模式下连续传输时，TE/BUSY 变化示意图



- **从机单向只接收模式（MSEL=0，BIDIRMODE=0 并且 RONLY=1）**

当从设备接收到时钟信号并且第一个数据位出现在它的 MOSI 时，数据接收过程开始。接收到的数据按顺序被串行地移位进入 8 位的移位寄存器中，然后被并行地传送到 SPI_DAT 寄存器（接收缓冲器）中。

- **从机双向发送模式（MSEL=0，BIDIRMODE=1 并且 BIDIROEN=1）**

当从设备接收到时钟信号第一个边沿时，发送过程开始。该模式下不接收数据，软件必须保证在 SPI 主设备开始数据传输之前寄存器 SPI_DAT 中已写入待发送的数据。

- **从机双向接收模式（MSEL=0，BIDIRMODE=1 并且 BIDIROEN=0）**

当从设备接收到时钟信号并且第一个数据位出现在它的 MOSI 时，数据传输开始。该模式下无数据输出，接收到的数据按顺序被串行地移位进入 8 位的移位寄存器中，然后被并行地传送到 SPI_DAT 寄存器（接收缓冲器）中。

注：从机的软件操作过程可参照主机。

SPI 初始化流程

1. 通过 SPI_CTRL1 寄存器的 BR[2:0]位定义串行时钟波特率（如果工作在从模式，忽略此步骤）。
2. 选择 CLKPOL 和 CLKPHA 位，定义数据传输和串行时钟间的相位关系（见图 22-4）。
3. 设置 DATFF 位来定义 8 位或 16 位数据帧格式。
4. 配置 SPI_CTRL1 寄存器的 LSBFF 位定义帧格式。
5. 按照上文 NSS 功能的描述配置 NSS 模式。
6. 通过 MSEL、BIDIRMODE、BIDIROEN、RONLY 位，配置运行模式。
7. 设置 SPIEN 位，使能 SPI。

基本的发送与接收流程

SPI 发送一个数据帧时，首先将这个数据帧从数据缓冲区加载到移位寄存器中，然后开始发送加载

的数据。当数据从发送缓冲器传送到移位寄存器时，SPI_STS 寄存器的发送缓冲为空的标志位 TE 置‘1’，此时可以装载下一个数据到发送缓冲器；如果 SPI_CTRL2 寄存器 TEINTEN 位置‘1’，则会产生一个中断；写入 SPI_DAT 寄存器可将 TE 位清‘0’。

在采样时钟的最后一个边沿，当数据被从移位寄存器传送到接收缓冲器时，SPI_STS 寄存器的接收缓冲器非空的标志位 RNE 置‘1’，此时数据已经就绪，可以从 SPI_DAT 寄存器读出；如果 SPI_CTRL2 寄存器 RNEINTEN 位置‘1’，则会产生一个中断；读出 SPI_DAT 寄存器即可清除 RNE 位。

在主模式下，当数据写入发送缓冲区时，发送过程开始。若当前数据帧发送完成前，下一个数据已写入 SPI_DAT 寄存器中，则实现连续发送功能。

在从模式下，NSS 管脚电平为低，当时钟信号第一个边沿来到时，发送过程开始。为避免意外的数据传输，软件必须在发送开始前，将数据写入发送缓冲区（建议在主设备发送时钟之前使能 SPI 从设备）。

在一些配置中，传输最后一个数据时，可以使用 BUSY 标志等待数据传输的结束。

连续和非连续传输

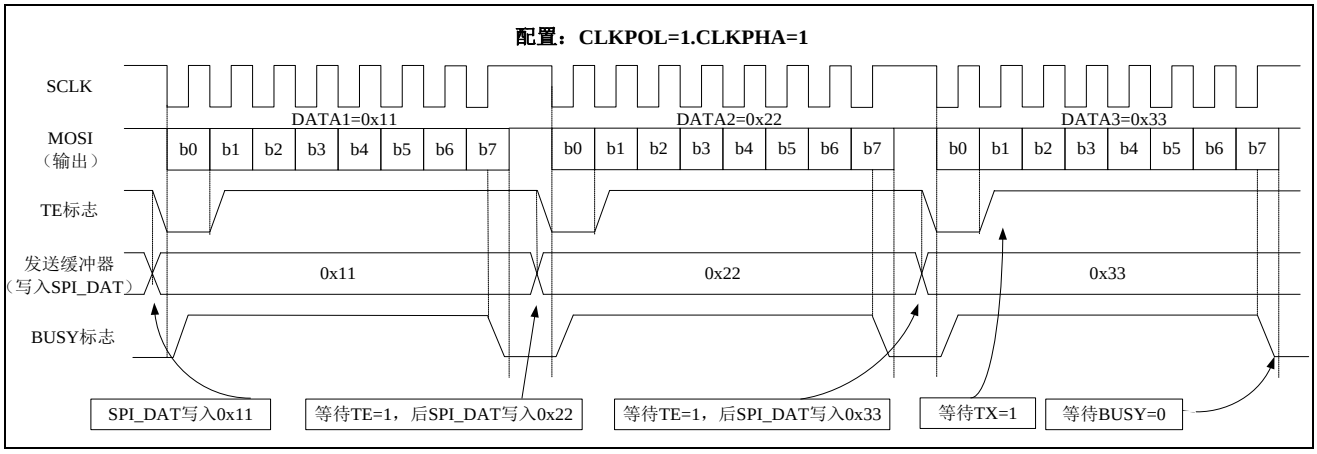
当在主模式下发送数据时，如果软件足够快，能够在检测到每次 TE 的上升沿（或 TE 中断），并立即在正在进行的传输结束之前写入 SPI_DAT 寄存器，此时，在每个数据项的传输之间的 SPI 时钟保持连续，BUSY 位不会被清除，能够实现连续的通信。

如果软件不够快，则会导致不连续的通信；这时，在每个数据传输之间 BUSY 位清零（见下图）。

在主机只接收模式下（RONLY=1），通信总是连续的，而且 BUSY 标志始终为高。

在从模式下，通信的连续性由 SPI 主设备决定。不管怎样，即使通信是连续的，BUSY 标志会在每个数据项之间至少有一个 SPI 时钟周期为低（见图 22-9）。

图 22-10 BIDIRMODE=0，RONLY=0 非连续传输发送时，TE/BUSY 变化示意图



22. 3. 3 状态标志

SPI_STS 寄存器中有以下 3 个标志位，用以监视 SPI 总线的状态。

发送缓存空标志位（TE）

发送缓冲器为空时，该位置‘1’，此时可以将新的待发送数据写入 SPI_DAT。

发送缓冲器非空时，该位清'0'。

接收缓存非空标志位（RNE）

接收缓存区非空时，该位置'1'，表示在接收缓存里已接收到的有效数据。

读取寄存器 SPI_DAT 时，该位清'0'

忙标志位（BUSY）

BUSY 标志可以检测传输是否结束，由硬件设置与清除（软件操作无效）。

SPI 通讯正在进行时 BUSY 标志位置'1'，但主模式的双向接收模式（MSEL=1、BIDIRMODE=1、BIDIROEN=0），在接收期间 BUSY 标志置'0'。当传输结束或关闭模块时，该标志位置'0'。

BUSY 标志被清'0'的情况：

- 传输结束（主模式下连续通信的情况除外）；
- 关闭 SPI 模块（SPIEN=0）；
- 产生主模式失效（MODERR=1）。

当通信不是连续时：BUSY 标志在每个数据项的传输之间被清'0'。

当通信是连续时：主模式下，BUSY 标志在整个传输过程中保持为高；从模式下，BUSY 标志在每个数据项的传输之间，有一个 SPI 时钟周期为低。因此，不要使用 BUSY 标志处理每一个数据项的发送和接收。

22.3.4 关闭 SPI

若要关闭 SPI 功能，需要根据不同的工作模式采取不同的操作步骤：

主机或从机全双工模式

1. 等待 RNE 置'1'并接收最后一个数据；
2. 等待 TE 置'1'；
3. 等待 BUSY 清'0'；
4. 关闭 SPI（SPIEN=0）。

主机或从机的单向只发送模式或双向的发送模式

1. 在寄存器 SPI_DAT 中写入最后一个数据后，等待 TE 置'1'；
2. 等待 BUSY 清'0'；
3. 关闭 SPI（SPIEN=0）。

主机单向只接收模式或双向的接收模式

1. 等待倒数第二个 RNE 置'1'；
2. 在关闭 SPI（SPIEN=0）之前等待一个 SPI 时钟周期（使用软件延迟）；
3. 在进入停机模式（或关闭该模块的时钟）之前等待最后一个 RNE=1。

从机单向只接收模式或双向的接收模式

1. 可以在任何时候关闭 SPI（SPIEN=0），SPI 会在当前的传输结束后被关闭；
2. 如果希望进入停机模式，在进入停机模式（或关闭该模块的时钟）之前必须首先等待 BUSY=0。

22.3.5 利用 DMA 的 SPI 通信

SPI 利用 DMA 传输数据，将应用程序从读写收发缓冲区的过程中释放出来，大大提高了系统效率。

当发送缓冲区 DMA 使能（SPI_CTRL2 寄存器 TDMAEN=1），每次 TE 被置'1'时发出 DMA 请求，DMA 自动将数据写入 SPI_DAT 寄存器，该操作会清除 TE 标志。当接收缓冲区 DMA 使能（SPI_CTRL2 寄存器 RDMAEN =1），每次 RNE 被置'1'时发出 DMA 请求，DMA 自动从 SPI_DAT 寄存器读出数据，该操作会清除 RNE 标志。

当只使用 SPI 发送数据时，只需使能 SPI 的发送 DMA 通道。此时，因为没有读取收到的数据，OVER 被置为'1'，此时软件无需处理这个标志。

当只使用 SPI 接收数据时，只需使能 SPI 的接收 DMA 通道。

在发送模式下，当 DMA 已经传输了所有要发送的数据（DMA_INTSTS 寄存器的 TXCF 标志变为'1'）后，可以通过监视 BUSY 标志以确认 SPI 通信结束，这样可以避免在关闭 SPI 或进入停止模式时，破坏最后一个数据的传输。因此软件需要先等待 TE=1，然后等待 BUSY=0。

图 22-11 使用 DMA 发送

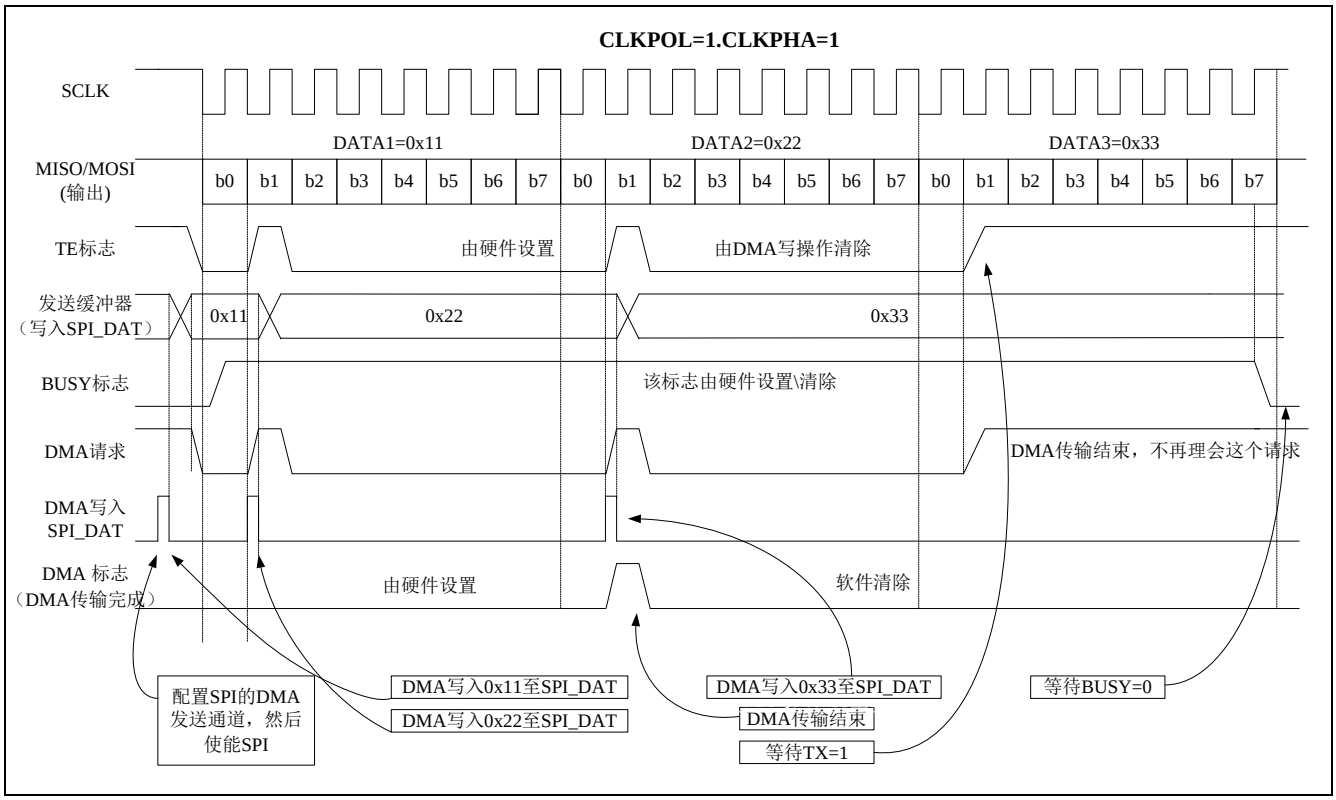
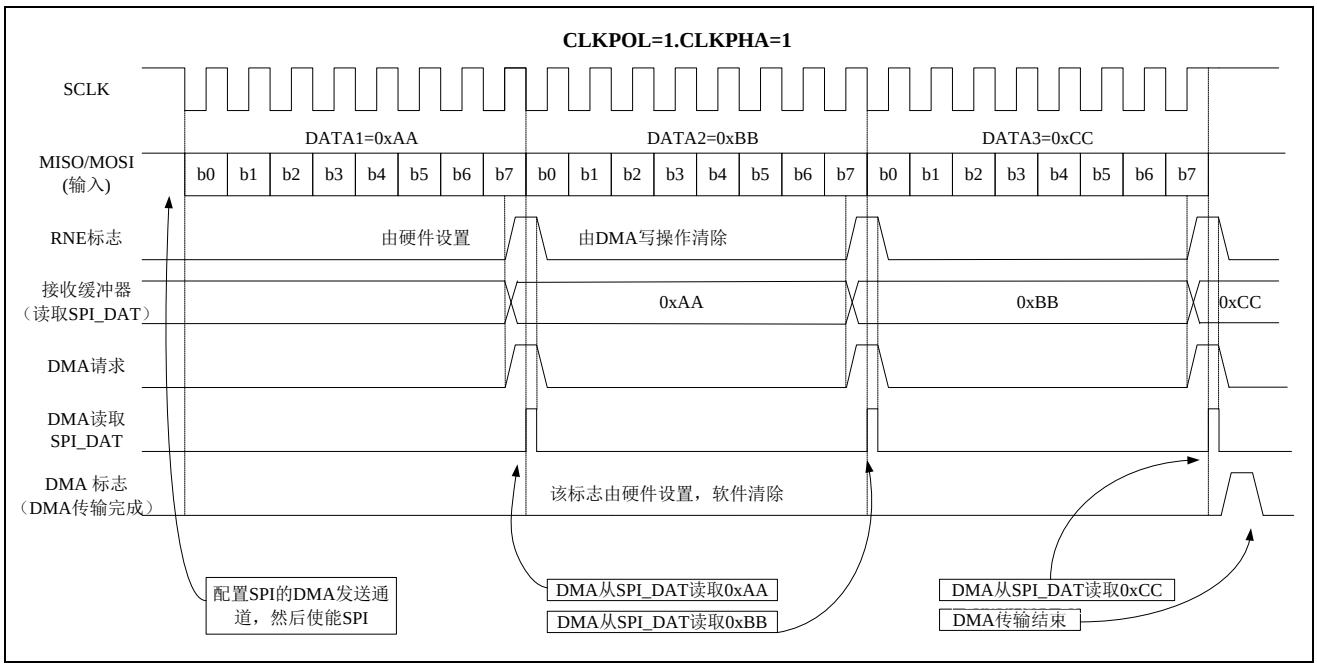


图 22-12 使用 DMA 接收



22.3.6 CRC 计算

SPI 包含用于数据发送和数据接收的两个独立硬件 CRC 计算器，以保证数据传输的可靠性。CRC 根据发送和/或接收的数据帧格式采用不同的计算方法：8 位数据帧采用 CRC8；16 位数据帧采用 CRC16。SPI_CRCPOLY 寄存器设置计算所需的多项式值，设置 SPI_CTRL1 寄存器的 CRCEN 位置‘1’使能 CRC 计算。

在发送模式下，最后一个数据写入发送缓冲区后，将 CRCNEXT 位置‘1’，指示硬件发送完这个数据后发送 CRC 的数值（SPI_CRCTDAT 的值）。在传输 CRC 期间，CRC 计算停止。

在接收模式下，在倒数第二个数据帧被接收后，将 CRCNEXT 位置‘1’。后续接收到的 CRC 与 SPI_CRCRDAT 值进行比较，如果二者不同，SPI_STS 寄存器的 CRCERR 标志位置‘1’，当 SPI_CTRL2 寄存器的 ERRINTEN 置‘1’时，产生中断。

为了保持主从设备端下次 CRC 计算结果的同步，应该清除主从两端的 CRC 数值。设置 CRCEN 位可同时复位 SPI_CRCRDAT 和 SPI_CRCTDAT。依次执行步骤：SPIEN=0；CRCEN=0；CRCEN=1；SPIEN=1。

值得一提的是，当配置 SPI 为从模式并且使能 CRC 时，只要 SCLK 引脚上有时钟脉冲，即使 NSS 引脚为高，仍然会执行 CRC 的计算。这种情况常见于主设备交替地与多个从设备进行通信的时候，需注意避免 CRC 的误操作。

当 SPI 硬件 CRC 校验使能（CRCEN=1）并且启用 DMA 模式时，在通信结束时，硬件自动完成 CRC 字节的发送和接收。

22.3.7 错误标志

主模式失效错误（MODERR）

以下两种情况下会发生主模式失效错误：

- NSS 引脚硬件模式管理，主设备的 NSS 脚被拉低；
- NSS 引脚软件模式管理，SSEL 位被置'0'。

发生主模式时效错误时，MODERR 标志位置'1'，若 ERRINTEN=1，则产生中断； SPIEN 位和 MSEL 位被写保护，硬件将二者清零，SPI 关闭，强制进入从机模式。

软件执行一次对 SPI_STS 寄存器的读或写操作，然后写 SPI_CTRL1 寄存器，可以清除 MODERR 位（在多主配置中，必须先拉高该主设备的 NSS 脚）。

通常配置下，从设备的 MODERR 位不能被置为'1'。然而，在多主配置里，一个设备可以在设置了 MODERR 位的情况下，处于从设备模式；此时，MODERR 位表示可能出现了多主冲突。中断程序可以执行一个复位或返回到默认状态来从错误状态中恢复。

溢出错误（OVER）

当 RNE 置'1'时，仍有数据发送至接收缓冲区即产生溢出错误。此时，OVER 标志位置'1'，若 ERRINTEN=1，则产生中断。所有新接收到数据会被丢失，SPI_DAT 寄存器中是之前未读的数据。

依次对 SPI_DAT 寄存器和 SPI_STS 寄存器进行读操作可清除 OVER 位。

CRC 错误（CRCERR）

CRC 错误标志用来核对接收数据的有效性。当接收到的 CRC 的值与 SPI_CRCRDAT 寄存器中的数值不匹配时产生 CRC 错误。此时，CRCERR 标志位置'1'，若 ERRINTEN=1，则产生中断。

22.3.8 SPI 中断

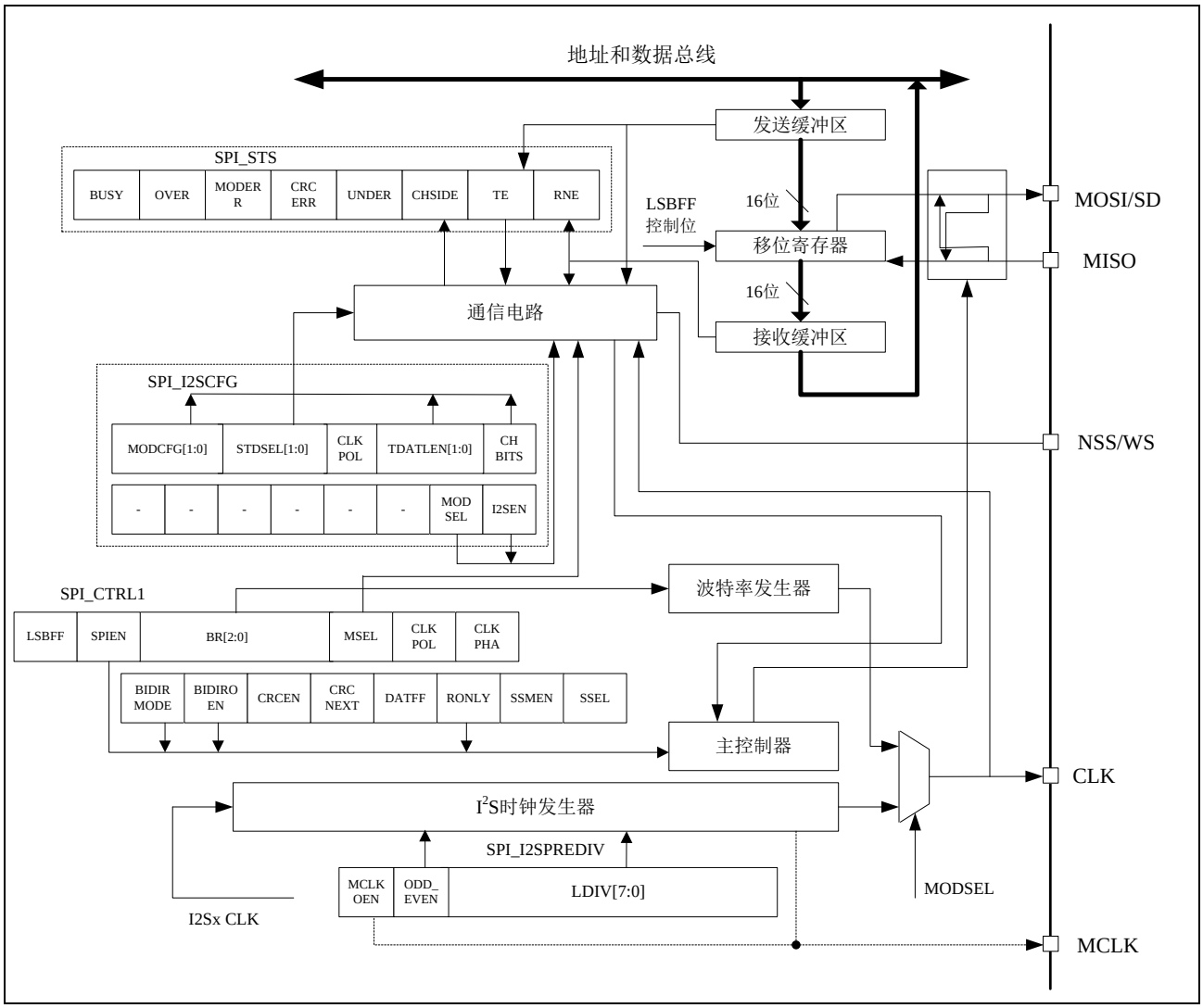
表 22-1 SPI 中断请求

中断事件	事件标志位	使能控制位
发送缓冲器空标志	TE	TEINTEN
接收缓冲器非空标志	RNE	RNEINTEN
主模式失效事件	MODERR	ERRINTEN
溢出错误	OVER	
CRC 错误标志	CRCERR	

22.4 I2S 功能描述

I2S 的框图如下图所示：

图 22-13 I2S 框图



I2S 接口与 SPI 接口使用大致相同的引脚、标志和中断。寄存器 SPI_I2SCFG 的 MODSEL 位置'1'，使能 I2S 音频接口。

I2S 共有 4 个管脚，其中 3 个管脚与 SPI 共用:

- CLK：串行时钟（与 SCLK 管脚共享），每发送 1 位数字音频数据，CLK 上都有一个脉冲，CLK 的频率=2 ×F_S×采样位数，主模式下作为时钟信号输出，从模式下作为输入;
- SD：串行数据（与 MOSI 管脚共享），用于收发数据;
- WS：声道选择（与 NSS 管脚共享），主模式下作为数据控制信号输出，从模式下作为输入;
- MCLK：主时钟（独立映射，可选），输出 256 × F_S的时钟频率。保证系统间更好的同步。

注：F_S是音频信号的采样频率

设置成主模式时，I2S 使用自身的时钟发生器来产生通信用的时钟信号。这个时钟发生器也是主时钟输出的时钟源（寄存器 SPI_I2SPREDIV 的 MCLKOEN 位置'1'，主时钟输出使能）。

22.4.1 支持的音频协议

I2S 接口支持四种音频标准，通过配置寄存器 SPI_I2SCFG 的 STDSEL[1:0]位进行选择：

- I2S 飞利浦标准
- MSB 对齐标准
- LSB 对齐标准
- PCM 标准

支持左声道和右声道两个声道上音频数据的时分复用，左声道总是先于右声道发送数据。通过寄存器 SPI_STS 的 CHSIDE 位区分接收到的数据属于哪个声道。PCM 协议中 CHSIDE 位无意义。

寄存器 SPI_I2SCFG 的 TDATLEN 位设置待传输定数据长度，CHBITS 位设置声道的数据位数。支持四种数据格式发送数据：

- 16 位数据打包成 16 位帧
- 16 位数据打包成 32 位帧（前 16 位（MSB）是有意义的数据，后 16 位（LSB）被硬件置 0）
- 24 位数据打包成 32 位帧（前 24 位（MSB）是有意义的数据，后 8 位（LSB）被硬件置 0）
- 32 位数据打包成 32 位帧

I2S 使用与 SPI 相同的寄存器 SPI_DAT 用于发送或接收 16 位宽模式数据。因此，传输数据长度为 24 位和 32 位的数据帧需要 CPU 对寄存器 SPI_DAT 进行 2 次读或写操作；而传输数据长度为 16 位的数据帧只需对 SPI_DAT 寄存器读写操作 1 次。

对于所有的数据格式和通讯标准，总是先发送数据位的最高位（MSB）。

I2S 飞利浦标准

I2S 飞利浦标准中，发送方在时钟信号的下降沿改变数据，接收方在上升沿读取数据。WS 信号在发送第一位数据（MSB）前 1 个时钟周期即为有效，在时钟信号的下降沿开始变化。

图 22-14 I2S 飞利浦协议波形（16/32 位全精度，CLKPOL = 0）

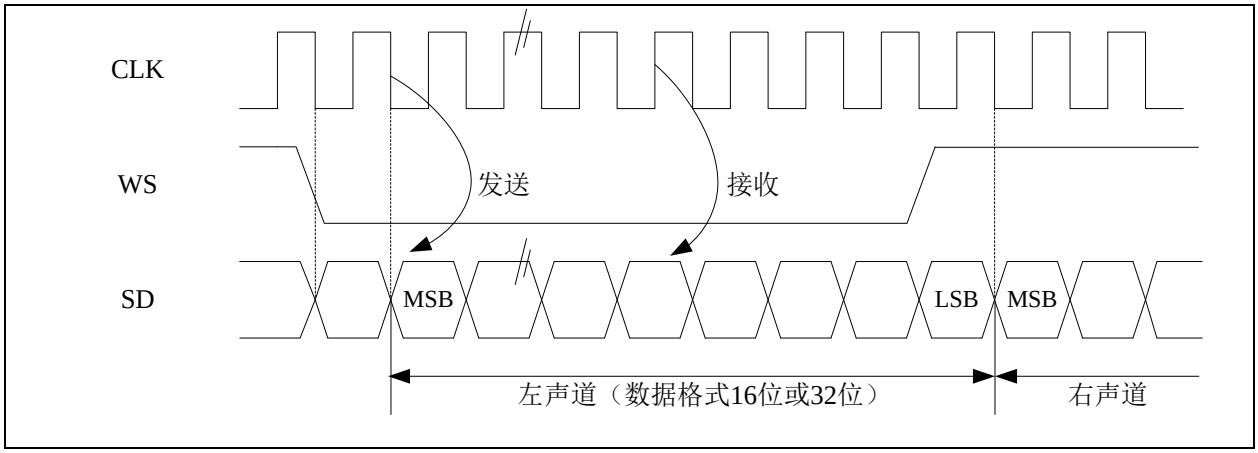
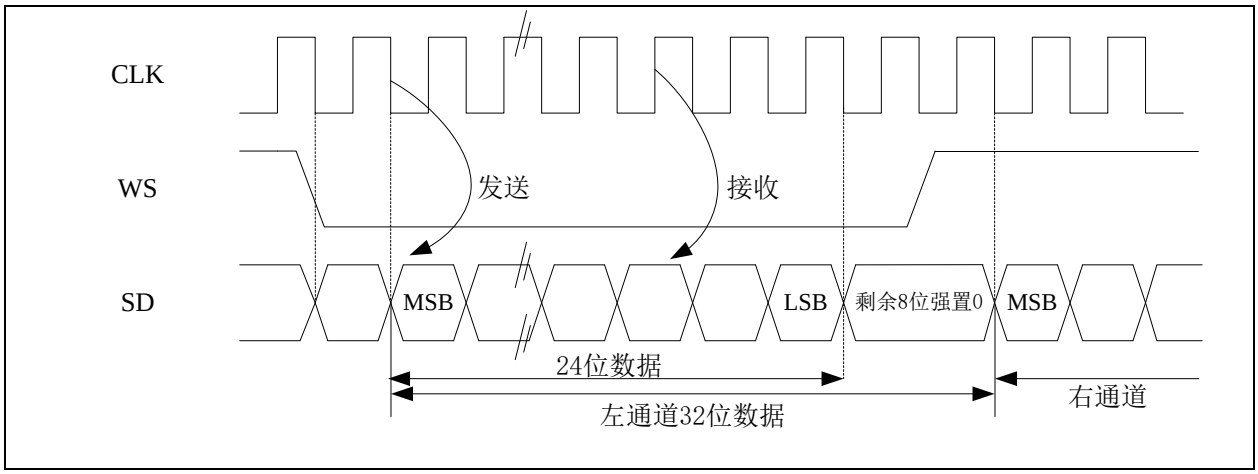
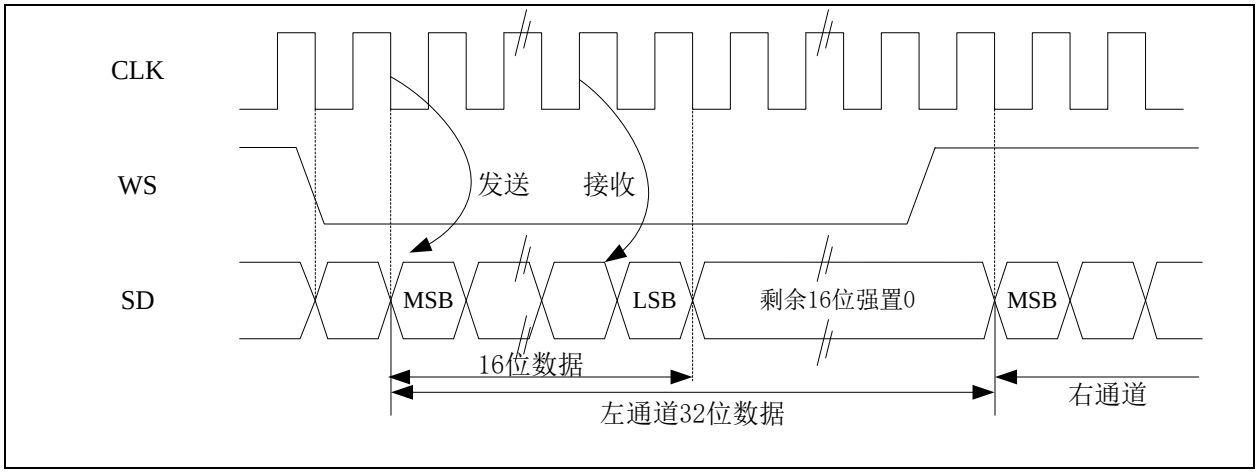


图 22-15 I2S 飞利浦协议标准波形（24 位帧，CLKPOL = 0）



当 24 位数据打包成 32 位数据帧的帧格式时，每完成 1 帧数据的传输需要对寄存器 SPI_DAT 进行 2 次读或写操作。例：发送 24 位数据 0x95AA66，第一次 SPI_DAT 写入 0x95AA，第二次 SPI_DAT 写入 0x66XX（只有高 8 位有效，低 8 位没有意义，可以是任意值）；接收 24 位数据 0x95AA66，第一次读 SPI_DAT 得到 0x95AA，第二次读 SPI_DAT 得到 0x6600（只有高 8 位有效，低 8 位始终为 0）。

图 22-16 I2S 飞利浦协议标准波形（16 位扩展至 32 位包帧，CLKPOL = 0）



当 16 位数据打包成 32 位数据帧的帧格式时，只需对寄存器 SPI_DAT 进行一次读写操作。用来扩展到 32 位的低 16 位被硬件置为 0x0000。例：待发送或者接收的 16 位数据是 0x89C1（扩展到 32 位是 0x89C10000）。在发送时，需要将高 16 位半字（0x89C1）写入寄存器 SPI_DAT；标志位 TE 为'1'表示可以写入新的数据，如果允许了相应的中断，则可以产生中断。发送是由硬件完成的，即使还未发送出后 16 位的 0x0000，也会设置 TE 并产生相应的中断；接收时，每次收到高 16 位半字（0x89C1）后，标志位 RNE 置'1'，如果允许了相应的中断，则可以产生中断。这样，在 2 次读和写之间有更多的时间，可以防止下溢或者上溢的情况发生。

MSB 对齐标准

MSB 对齐标准中，发送方在时钟信号的下降沿改变数据；接收方是在上升沿读取数据。WS 信号和第一个数据位（最高位 MSB）同时产生。

该标准数据收发处理方式与 I2S 飞利浦标准相同。

图 22-17 MSB 对齐 16 位或 32 位全精度，CLKPOL = 0

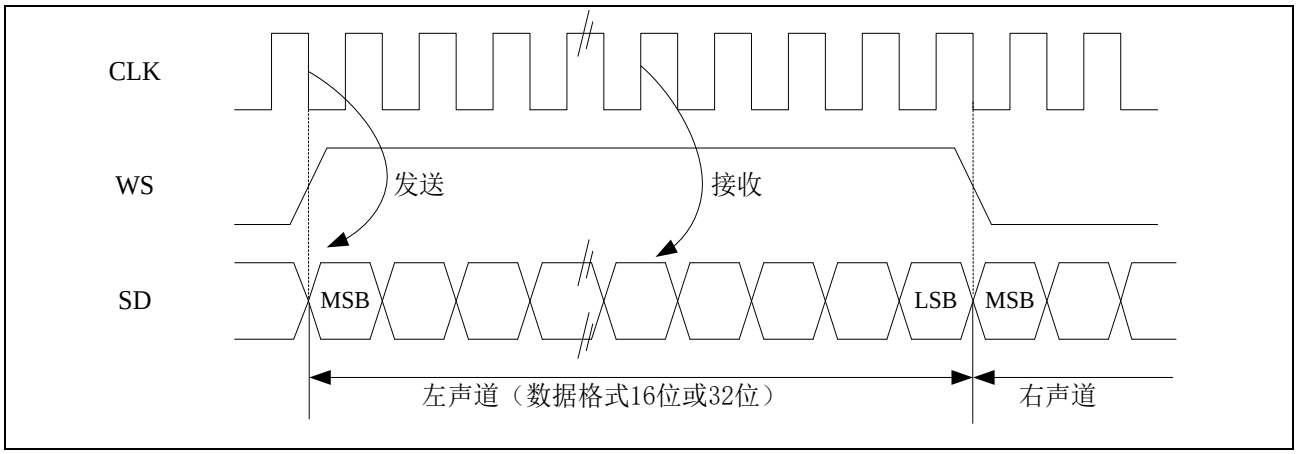


图 22-18 MSB 对齐 24 位数据，CLKPOL = 0

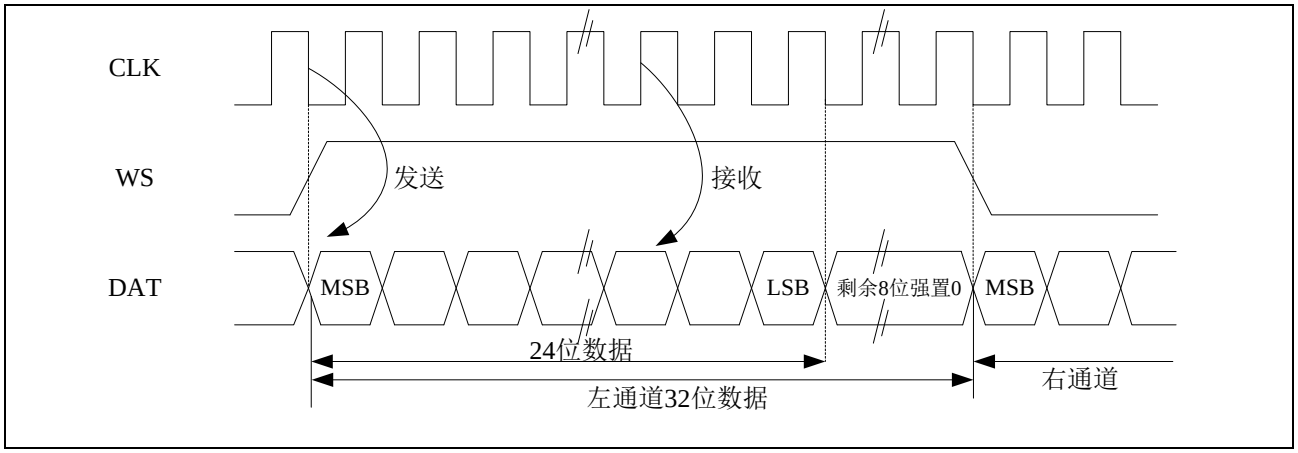
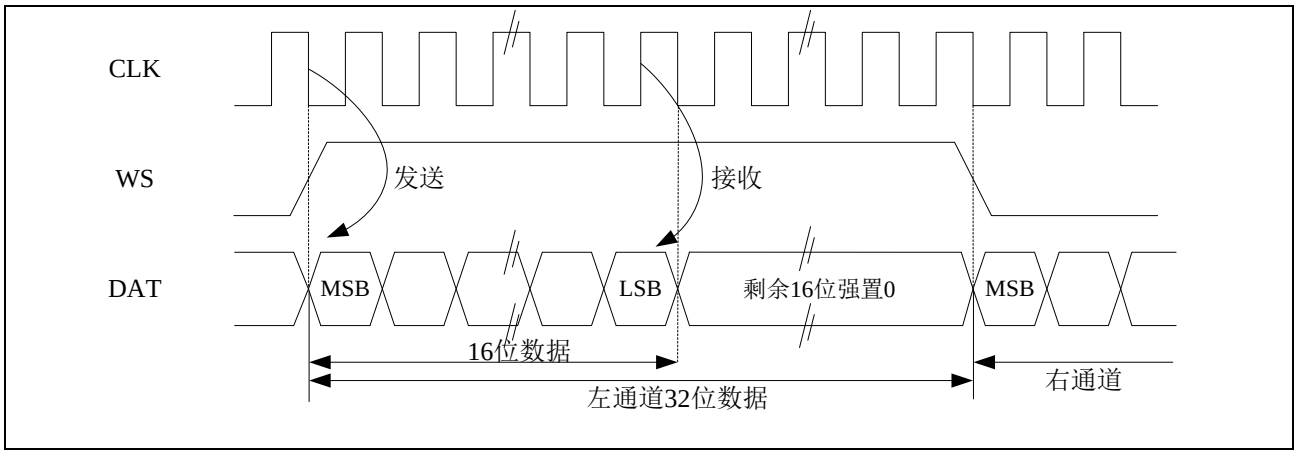


图 22-19 MSB 对齐 16 位数据扩展到 32 位包帧，CLKPOL = 0



LSB 对齐标准

16 位或 32 位全精度帧格式下，LSB 对齐标准与 MSB 对齐标准相同。

图 22-20 LSB 对齐 16 位或 32 位全精度，CLKPOL = 0

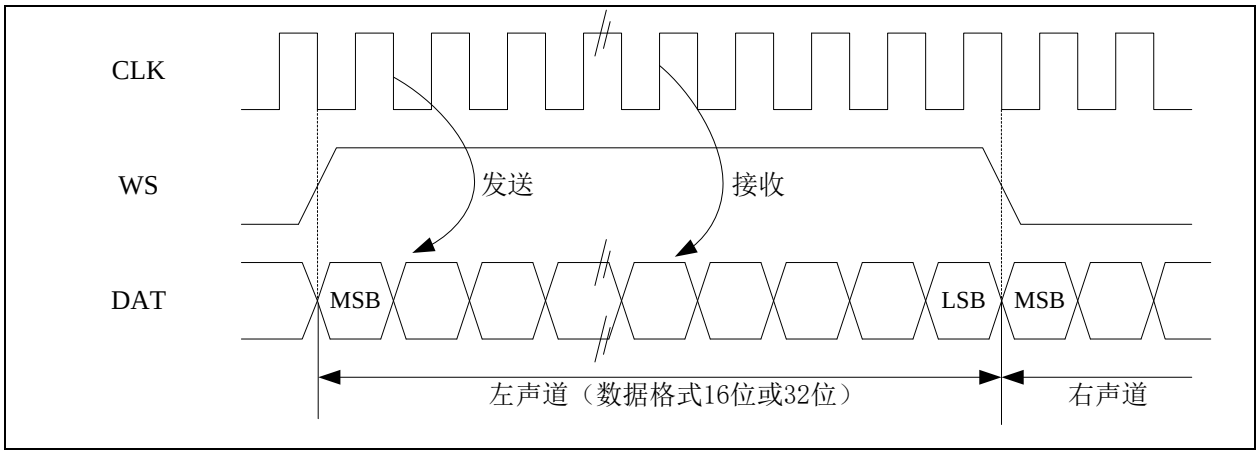
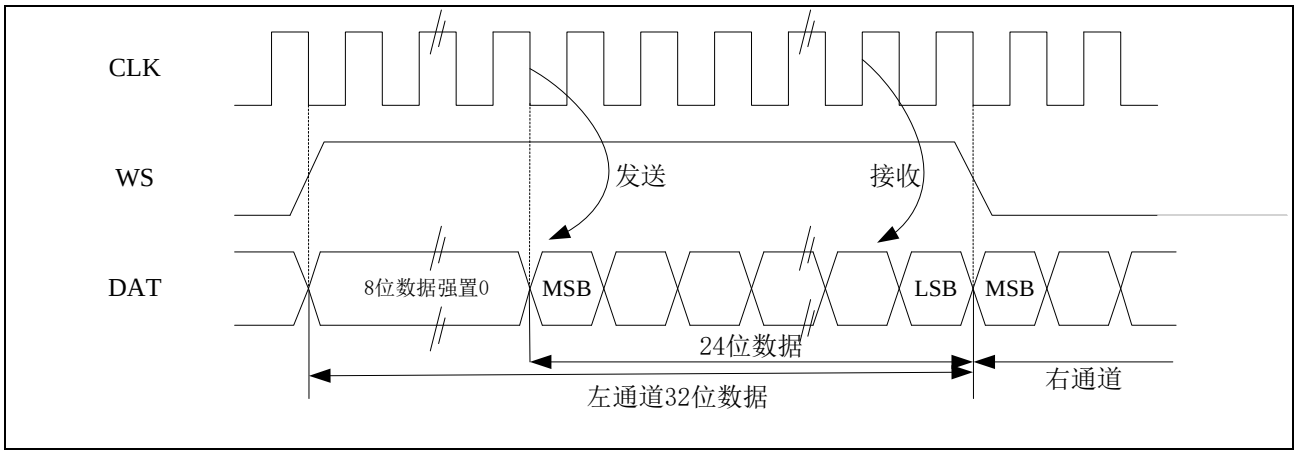
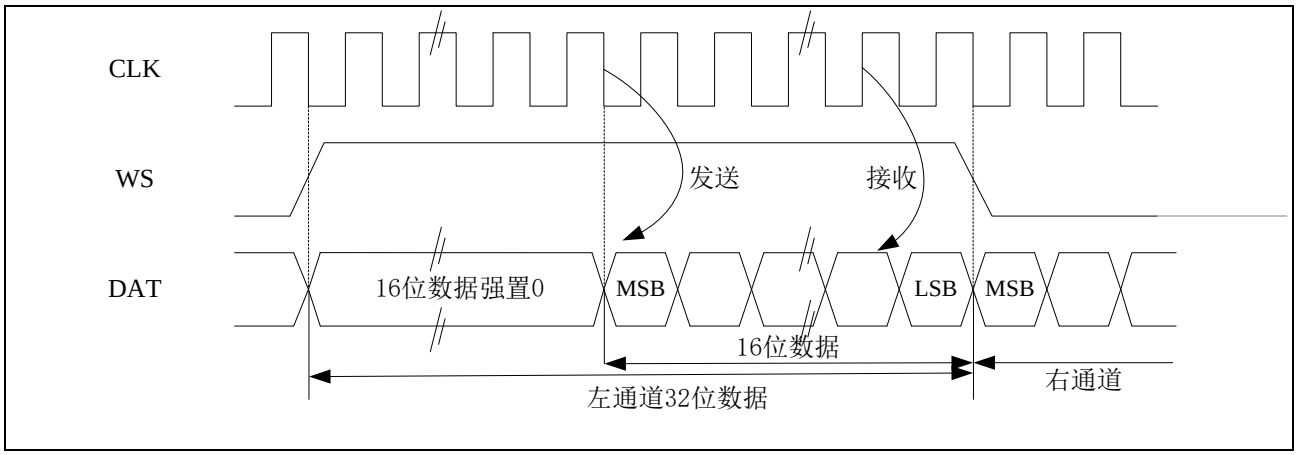


图 22-21 LSB 对齐 24 位数据，CLKPOL = 0



当 24 位数据打包成 32 位数据帧的帧格式时，每完成 1 帧数据的传输需要对寄存器 SPI_DAT 进行 2 次读或写操作。例：发送 24 位数据 0x95AA66，第一次 SPI_DAT 写入 0xXX95（只有低 8 位有效，高 8 位没有意义，可以是任意值），第二次 SPI_DAT 写入 0xAA66。接收 24 位数据 0x95AA66，第一次读 SPI_DAT 得到 0x0095（只有低 8 位有效，高 8 位始终为 0），第二次读 SPI_DAT 得到 0xAA66。

图 22-22 LSB 对齐 16 位数据扩展到 32 位包帧，CLKPOL = 0



当 16 位数据打包成 32 位数据帧的帧格式时，只需对寄存器 SPI_DAT 进行一次读写操作。用来扩展到 32 位的高 16 位被硬件置为 0x0000。例：待发送或者接收的 16 位数据是 0x89C1（扩展到 32 位是 0x000089C1）。在发送时，如果 TE 为'1'，用户需要写入待发送的数据（0x89C1）。用来扩展的高 16 位 0x0000 由硬件首先发送出去，一旦有效数据开始从 SD 引脚送出，即发生下一次 TE 事件。在接收时，一旦接收到有效数据（非 0x0000 部分），即发生 RNE 事件。这样，在 2 次读和写之间有更多的时间，可以防止下溢或者上溢的情况发生。

PCM 标准

PCM 标准中有短帧、长帧 2 种帧结构，通过设置寄存器 SPI_I2SCFG 的 PCMFSYNC 位选择。WS 信号表示帧同步信息。长帧用来同步的 WS 信号有效的时间为 13 位；短帧用来同步的 WS 信号长度为 1 位。

该标准数据收发处理方式与 I2S 飞利浦标准相同。

图 22-23 PCM 标准波形（16 位）

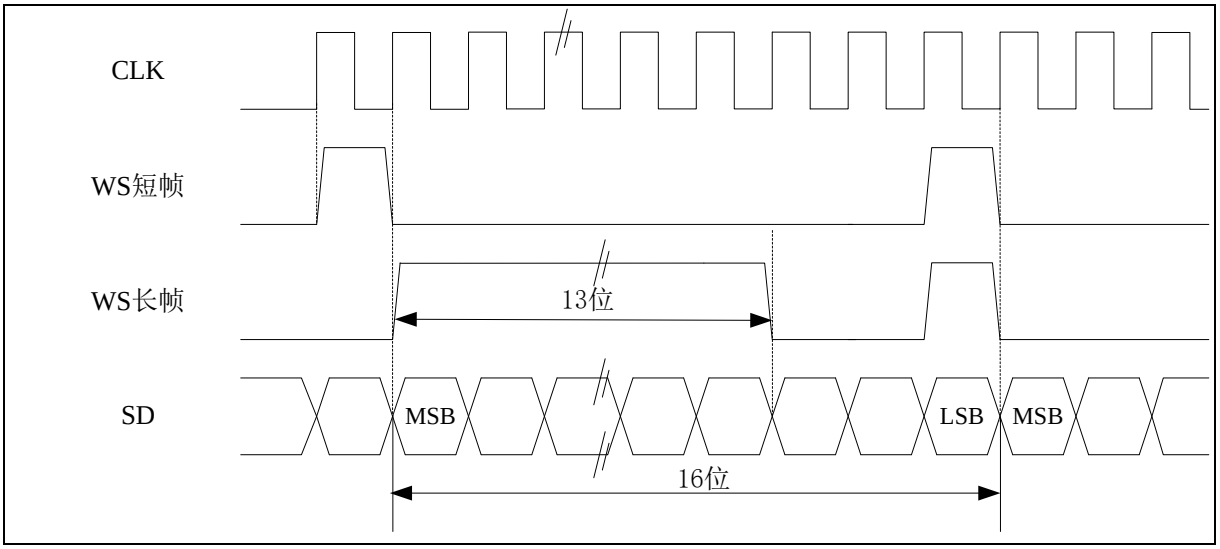
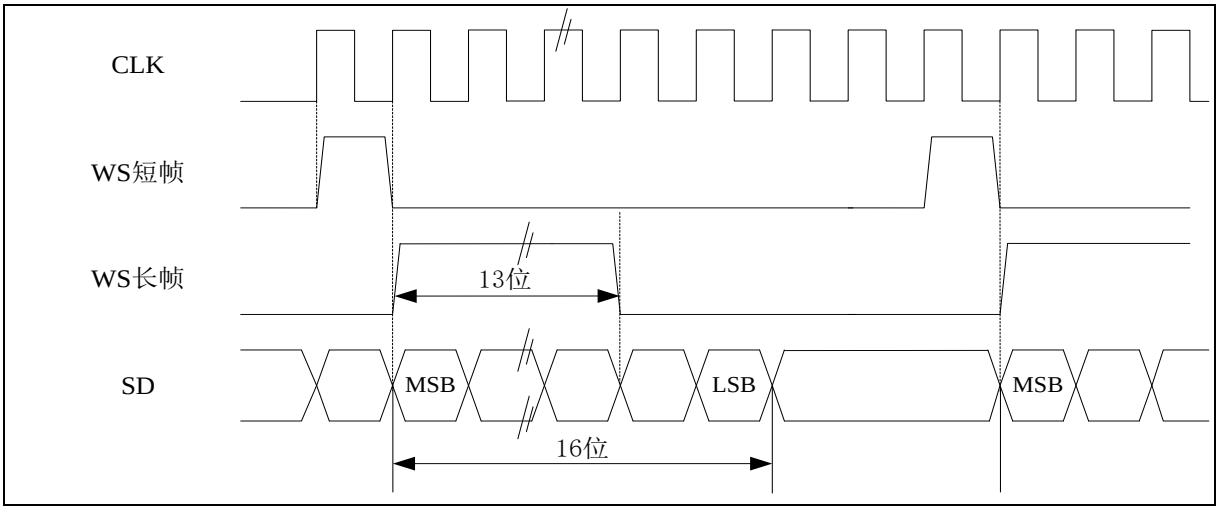


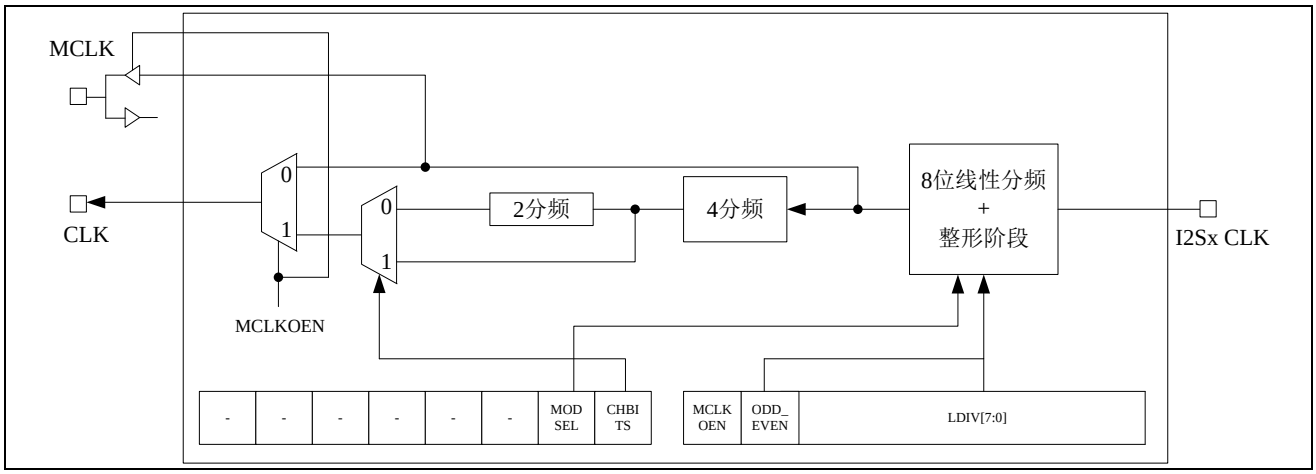
图 22-24 PCM 标准波形（16 位扩展到 32 位包帧）



22.4.2 时钟发生器

在主模式下，为了获得需要的音频频率，需要正确地对线性分频器进行设置。

图 22-25 I2S 时钟发生器结构



注：I2SxCLK 的时钟源是驱动 AHB 时钟的 HSI、HSE 或 PLL 系统时钟。

I2S 的比特率即确定了在 I2S 数据线上的数据流和 I2S 的时钟信号频率。

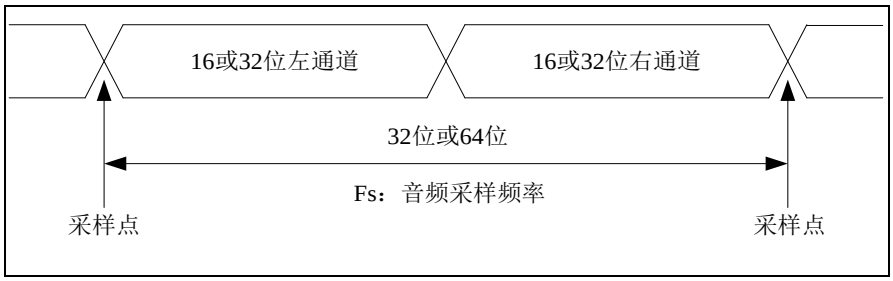
I2S 比特率 = 每个声道的比特数 × 声道数目 × 音频采样频率

对于一个具有左右声道和 16 位音频的信号，I2S 比特率计算如下：

I2S 比特率 = 16 × 2 × F_S

如果包长为 32 位，则有：I2S 比特率 = 32 × 2 × F_S

图 22-26 音频采样频率定义



音频的采样频率可以是 96KHz、48KHz、44.1KHz、32KHz、22.05KHz、16KHz、11.025KHz 或者 8KHz（或任何此范围内的数值）。参照以下公式设置线性分频器：

MCLKOEN=1，CHBITS=0 时，$F_S = I^2SxCLK/[(16 \times 2) \times ((2 \times LDIV) + ODD\_EVEN) \times 8]$

MCLKOEN=1，CHBITS=1 时，$F_S = I^2SxCLK/[(32 \times 2) \times ((2 \times LDIV) + ODD\_EVEN) \times 4]$

MCLKOEN=0，CHBITS=0 时，$F_S = I^2SxCLK/[(16 \times 2) \times ((2 \times LDIV) + ODD\_EVEN)]$

MCLKOEN=0，CHBITS=1 时，$F_S = I^2SxCLK/[(32 \times 2) \times ((2 \times LDIV) + ODD\_EVEN)]$

可参照下表的时钟配置得到精确的音频频率。

表 22-2 使用标准的 8MHz HSE 时钟得到精确的音频频率

SYSCLK (MHz)	I2S_LDIV		I2S_ODD_EVEN		MCLK	期望值 F_S (Hz)	实际 F_S（Hz）		误差	
	16 位	32 位	16 位	32 位			16 位	32 位	16 位	32 位
144	23	11	1	1	无	96000	95744.68	97826.09	0.27%	1.87%
144	47	23	0	1	无	48000	47872.34	47872.34	0.27%	0.27%
144	51	25	0	1	无	44100	44117.65	44117.65	0.04%	0.04%
144	70	35	1	0	无	32000	31914.89	32142.86	0.27%	0.44%
144	102	51	0	0	无	22050	22058.82	22058.82	0.04%	0.04%
144	140	70	1	1	无	16000	16014.23	15957.45	0.09%	0.27%
144	204	102	0	0	无	11025	11029.41	11029.41	0.04%	0.04%
144	281	140	1	1	无	8000	7992.90	8007.12	0.09%	0.09%
144	3	3	0	0	有	96000	93750.00	93750.00	2.40%	2.40%
144	6	6	0	0	有	48000	46875.00	46875.00	2.40%	2.40%
144	6	6	1	1	有	44100	43269.23	43269.23	1.92%	1.92%
144	9	9	0	0	有	32000	31250.00	31250.00	2.40%	2.40%
144	13	13	0	0	有	22050	21634.62	21634.62	1.92%	1.92%
144	17	17	1	1	有	16000	16071.43	16071.43	0.44%	0.44%
144	25	25	1	1	有	11025	11029.41	11029.41	0.04%	0.04%
144	35	35	0	0	有	8000	8035.71	8035.71	0.44%	0.44%

22.4.3 I2S 发送接收流程

I2S 初始化流程

1. 寄存器 SPI_I2SPREDIV 的 LDIV[7:0]和 ODD_EVEN 位配置预分频相关参数，串行时钟波特率。
2. 如果需要向外部的 DAC/ADC 音频器件提供主时钟 MCLK，将寄存器 SPI_I2SPREDIV 的 MCLKOEN 位置为'1'。（按照不同的 MCLK 输出状态，计算 LDIV 和 ODD_EVEN 的值，详见 22.4.2 节）。
3. 设置寄存器 SPI_I2SCFG 的 CLKPOL 位定义通信用时钟在空闲时的极性；MODSEL 位置'1'配成 I2S 模式，MODCFG[1:0]选择 I2S 主从模式和传输方向（发送或接收）；设置 STDSEL[1:0]选择所需 I2S 标准（PCM 标准下，设置 PCMFSYNC 位选取同步方式）；设置 TDATLEN[1:0]选择数据的比特数，设置 CHBITS 选择每个声道的数据位数。
4. 若需使用中断或 DMA，配置与 SPI 相同。
5. 最后启动 I2S 通信，I2SEN 位置'1'。

发送流程

主模式

当 I2S 工作在主模式，管脚 CLK 输出串行时钟，管脚 WS 产生声道选择信号。设置寄存器 SPI_I2SPREDIV 的 MCLKOEN 位来选择输出或者不输出主时钟（MCLK）。

当数据写入发送缓冲区时，发送过程开始。当前通道的数据并行的从发送缓存移到移位寄存器时，标志位 TE 置'1'，这时，要把另一个声道的数据写入 SPI_DAT。通过标志位 CHSIDE 确认当前待传输的数据所对应的通道。CHSIDE 的值在 TE 置'1'时更新。一个完整的数据帧包括左声道和右声道，不可以只传

输部分数据帧。当标志位 TE 置'1'，如果寄存器 SPI_CTRL2 的 TEINTEN 位为'1'，则产生中断。

写入数据的操作取决于所选择的 I2S 标准，详见 22.4.1 节。

关闭 I2S 功能时，等待标志位 TE=1 及 BUSY=0，再将 I2SEN 位清'0'。

从模式

从模式的发送流程与主模式相似，不同之处在于：

当 I2S 工作在从模式，无需配置时钟，管脚 CLK 和管脚 WS 与主设备对应管脚相连。当外部主设备发送时钟信号，并且当 WS 信号请求传输数据时，发送流程开始。必须先使能从设备，并且写入 I2S 数据寄存器之后，外部主设备才能开始通信。

当代表下一个数据传输的第一个时钟边沿到达之前，新的数据仍然没有写入寄存器 SPI_DAT，即发生下溢，标志位 UNDER 被置'1'，如果寄存器 SPI_CTRL2 的 ERRINTEN 位为'1'，则产生中断，表示发生了错误。

标志位 CHSIDE 提示了目前待传输的数据对应哪个声道。与主模式的发送流程相比，在从模式中，CHSIDE 取决于外部主 I2S 的 WS 信号（WS 信号为'1'表示发送左声道）。

接收流程

主模式

音频数据总是以 16 位包的形式接收。根据配置的数据和声道长度，收到的音频数据会需要 1 次或者 2 次传送到接收缓存的过程。

当数据被从移位寄存器传送到接收缓冲器时，SPI_STS 寄存器的接收缓冲器非空的标志位 RNE 置'1'，此时数据已经就绪，可以从 SPI_DAT 寄存器读出；如果 SPI_CTRL2 寄存器 RNEINTEN 位置'1'，则会产生一个中断；读出 SPI_DAT 寄存器即可清除 RNE 位。如果前一个接收到的数据还没有被读取，又接收到新数据，即发生上溢，标志位 OVER 被置为'1'，如果寄存器 SPI_CTRL2 的 ERRINTEN 位为'1'，则产生中断，表示发生了错误。

通过标志位 CHSIDE 确认当前传输的数据所对应的通道，CHSIDE 的值在 RNE 置'1'时更新。

读取数据的操作取决于所选择的 I2S 标准，详见 22.4.1 节。

关闭 I2S 功能时，不同的音频标准，数据长度和通道长度采用不同的操作步骤：

- 数据长度 16 位，通道长度 32 位（TDATLEN=00，CHBITS=1），LSB（低位）对齐标准（STDSEL=10）
  1. 等待倒数第二个 RNE 置'1'；
  2. 软件延迟，等待 17 个 I2S 时钟周期；
  3. 关闭 I2S（I2SEN=0）。
- 数据长度 16 位，通道长度 32 位（TDATLEN=00 并且 CHBITS=1），MSB（高位）对齐标准（STDSEL=01）、I2S 飞利浦标准（STDSEL=00）或 PCM 标准（STDSEL=11）
  1. 等待最后一个 RNE 置'1'；
  2. 软件延迟，等待 1 个 I2S 时钟周期；
  3. 关闭 I2S（I2SEN=0）。
- 其它 TDATLEN 和 CHBITS 的组合和 STDSEL 选择的任意音频模式：

1. 等待倒数第二个 RNE 置‘1’;
2. 软件延迟，等待一个 I2S 时钟周期;
3. 关闭 I2S（I2SEN=0）。

从模式

从模式的接收流程与主模式相似，不同之处在于:

标志位 CHSIDE 提示了目前待传输的数据对应哪个声道。与主模式的接收流程相比，在从模式中，CHSIDE 取决于外部主 I2S 的 WS 信号。关闭 I2S 功能时，等待标志位 RNE=1 时将 I2SEN 位清’0’。

22.4.4 状态标志位

SPI_STS 寄存器中有以下 4 个标志位，用以监视 I2S 总线的状态。

发送缓存空标志位（TE）

发送缓冲器为空时，该位置’1’，此时可以将新的待发送数据写入 SPI_DAT。

发送缓冲器非空时，该位清’0’。

接收缓存非空标志位（RNE）

接收缓存区非空时，该位置’1’，表示在接收缓存里已接收到的有效数据。

读取寄存器 SPI_DAT 时，该位清’0’。

忙标志位（BUSY）

BUSY 标志可以检测传输是否结束，由硬件设置与清除（软件操作无效）。

I2S 通讯正在进行时 BUSY 标志位置‘1’，但主接收模式（MODCFG=11）下，在接收期间 BUSY 标志被置‘0’。当传输结束或关闭 I2S 模块时，该标志位置‘0’。

在连续通信的从模式下，每个数据项传输之间，BUSY 标志在 1 个 I2S 时钟周期内变低。因此，不要使用 BUSY 标志处理每一个数据项的发送和接收。

声道标志位（CHSIDE）

CHSIDE 用来指示当前收发的数据所在的通道，在 PCM 标准下，这个标志位没有意义。

在发送模式下，该标志位在 TE 置‘1’时更新；在接收模式下，该标志位在 RNE 置‘1’时更新。在收发过程中，若发生上溢（OVER）或下溢（UNDER）错误，该标志位无意义，需要将 I2S 关闭再打开。

22.4.5 错误标志位

SPI_STS 寄存器有 2 个错误标志位。

上溢标志位（OVER）

当 RNE 置‘1’时，仍有数据发送至接收缓冲区即产生溢出错误。此时，OVER 标志位置’1’，若 ERRINTEN=1，则产生中断。所有新接收到数据会被丢失，SPI_DAT 寄存器中是之前未读的数据。

依次对 SPI_DAT 寄存器和 SPI_STS 寄存器进行读操作可清除 OVER 位。

下溢标志位（UNDER）

在从发送模式下，如果数据传输的第一个时钟边沿到达时，发送缓冲区仍为空时，UNDER 标志位置'1'。若 ERRINTEN=1，则产生中断。

对寄存器 SPI_STS 进行读操作可清除 UNDER 位。

22.4.6 I2S 中断

下表列举了全部 I2S 中断

表 22-3 I2S 中断请求

中断事件	事件标志位	使能控制位
发送缓冲器空标志	TE	TEINTEN
接收缓冲器非空标志	RNE	RNEINTEN
下溢标志位	UNDER	ERRINTEN
上溢标志位	OVER	

22.4.7 DMA 功能

I2S 模式下没有数据传输保护功能，故不支持 CRC 功能，其他 DMA 功能与 SPI 模式完全相同。

22.5 SPI 和 I2S 寄存器描述

22.5.1 SPI 寄存器地址映象

表 22-4 SPI 寄存器列表及其复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	SPI_CTRL1	Reserved																BIDIRMODE	BIDIROEN	CRCEN	CRCNEXT	DATFF	RONLY	SSMEN	SSEL	LSBFF	SPIEN	BR[2:0]			MSEL	CLKPOL	CLKPHA
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	SPI_CTRL2	Reserved																								TEINTEN	RNEINTEN	ERRINTEN	Reserved		SSOEN	TDMAEN	RDMAEN
	Reset Value																									0	0	0			0	0	0
008h	SPI_STS	Reserved																								BUSY	OVER	MODERR	CRCERR	UNDER	CHSIDE	TE	RNE
	Reset Value																									0	0	0	0	0	0	1	0
00Ch	SPI_DAT	Reserved																DAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	SPI_CRCPOLY	Reserved																CRCPOLY[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	SPI_CRCRDAT	Reserved																CRCRDAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	SPI_CRCTDAT	Reserved																CRCTDAT[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01Ch	SPI_I2SCFG	Reserved																				MODSEL	I2SEN	MODCFG [1:0]		PCMFSYNC	Reserved	STDSEL [1:0]		CLKPOL	TDATLEN [1:0]		CHBITS
	Reset Value																					0	0	0	0	0		0	0	0	0	0	0
020h	SPI_I2SPREDIV	Reserved																				MCLKOEN	ODD_EVEN	LDIV[7:0]									
	Reset Value																					0	0	0	0	0	0	0	0	0	0	1	0

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

地址偏移：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	3	2	1	0
BIDIR MODE	BIDIR OEN	CRCEN	CRC NEXT	DATFF	RONLY	SSMEN	SSEL	LSBFF	SPIEN	BR[2:0]		MSEL	CLKPOL	CLKPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

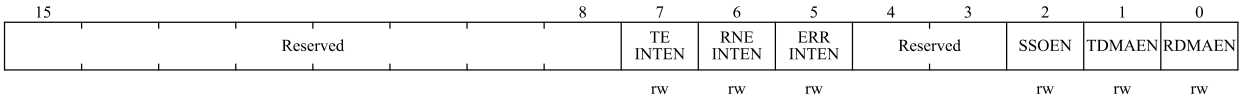
位域	名称	描述
15	BIDIRMODE	双向数据模式使能 0：选择“双线双向”模式； 1：选择“单线双向”模式。 *注：I2S 模式下不使用。*
14	BIDIROEN	双向模式下的输出使能 和 BIDIRMODE 位一起决定在“单线双向”模式下数据的输出方向 0：输出禁止（只收模式）； 1：输出使能（只发模式）。 这个“单线”数据线在主设备端为 MOSI 引脚，在从设备端为 MISO 引脚。 *注：I2S 模式下不使用。*
13	CRCEN	硬件 CRC 校验使能 0：禁止 CRC 计算； 1：启动 CRC 计算。 *注：只有在禁止 SPI 时（SPIEN=0），才能写该位，否则出错。* 该位只能在全双工模式下使用。 *注：I2S 模式下不使用。*
12	CRCNEXT	下一个发送 CRC 0：下一个发送的值来自发送缓冲区。 1：下一个发送的值来自发送 CRC 寄存器。 *注：在 SPI_DAT 寄存器写入最后一个数据后应马上设置该位。* *注：I2S 模式下不使用。*
11	DATFF	数据帧格式 0：使用 8 位数据帧格式进行发送/接收； 1：使用 16 位数据帧格式进行发送/接收。 *注：只有当 SPI 禁止（SPIEN=0）时，才能写该位，否则出错。* *注：I2S 模式下不使用。*

位域	名称	描述
10	RONLY	只接收 该位和 BIDIRMODE 位一起决定在“双线双向”模式下的传输方向。在多个从设备的配置中，在未被访问的从设备上该位被置 1，使得只有被访问的从设备有输出，从而不会造成数据线上数据冲突。 0：全双工（发送和接收）； 1：禁止输出（只接收模式）。 *注：I2S 模式下不使用。*
9	SSMEN	软件从设备管理 当 SSMEN 被置位时，NSS 引脚上的电平由 SSEL 位的值决定。 0：禁止软件从设备管理； 1：启用软件从设备管理。 *注：I2S 模式下不使用。*
8	SSEL	内部从设备选择 该位只在 SSMEN 位为'1'时有意义。它决定了 NSS 上的电平，在 NSS 引脚上的 I/O 操作无效。 *注：I2S 模式下不使用。*
7	LSBFF	帧格式 0：先发送 MSB； 1：先发送 LSB。 *注：当通信在进行时不能改变该位的值。* *注：I2S 模式下不使用。*
6	SPIEN	SPI 使能 0：禁止 SPI 设备； 1：开启 SPI 设备。 *注：I2S 模式下不使用。* *注：当关闭 SPI 设备时，请按照第 22.3.4 节的过程操作。*
5:3	BR[2:0]	波特率控制 000： fPCLK/2 001： fPCLK/4 010： fPCLK/8 011： fPCLK/16 100： fPCLK/32 101： fPCLK/64 110： fPCLK/128 111： fPCLK/256 当通信正在进行的时候，不能修改这些位。 *注：I2S 模式下不使用。*
2	MSEL	主设备选择 0：配置为从设备； 1：配置为主设备。 *注：当通信正在进行的时候，不能修改该位。* *注：I2S 模式下不使用。*

位域	名称	描述
1	CLKPOL	时钟极性 0： 空闲状态时，SCLK 保持低电平； 1： 空闲状态时，SCLK 保持高电平。 *注：当通信正在进行的时候，不能修改该位。* *注：I2S 模式下不使用。*
0	CLKPHA	时钟相位 0： 数据采样从第一个时钟边沿开始； 1： 数据采样从第二个时钟边沿开始。 *注：当通信正在进行的时候，不能修改该位。* *注：I2S 模式下不使用。*

22.5.3 SPI 控制寄存器 2（SPI_CTRL2）

地址偏移：0x04
复位值：0x0000

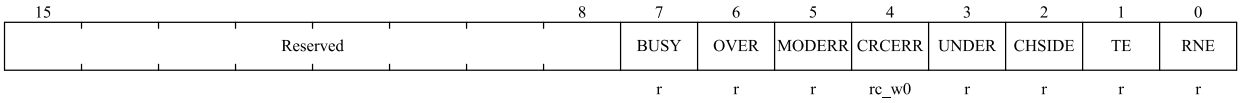


位域	名称	描述
15:8	Reserved	保留，硬件强制为 0。
7	TEINTEN	发送缓冲区空中断使能 0：禁止 TE 中断； 1：允许 TE 中断，当 TE 标志置位为’1’时产生中断请求。
6	RNEINTEN	接收缓冲区非空中断使能 0：禁止 RNE 中断； 1：允许 RNE 中断，当 RNE 标志置位时产生中断请求。
5	ERRINTEN	错误中断使能 当错误（CRCERR、OVER、MODERR）产生时，该位控制是否产生中断 0：禁止错误中断； 1：允许错误中断。
4:3	Reserved	保留，硬件强制为 0。
2	SSOEN	SS 输出使能 0：禁止在主模式下 SS 输出，该设备可以工作在多主设备模式； 1：设备开启时，开启主模式下 SS 输出，该设备不能工作在多主设备模式。 *注：I2S 模式下不使用。*
1	TDMAEN	发送缓冲区 DMA 使能 当该位被设置时，TE 标志一旦被置位就发出 DMA 请求 0：禁止发送缓冲区 DMA； 1：启动发送缓冲区 DMA。

位域	名称	描述
0	RDMAEN	接收缓冲区 DMA 使能 当该位被设置时，RNE 标志一旦被置位就发出 DMA 请求 0：禁止接收缓冲区 DMA； 1：启动接收缓冲区 DMA。

22.5.4 SPI 状态寄存器（SPI_STS）

地址偏移：0x08
复位值：0x0002

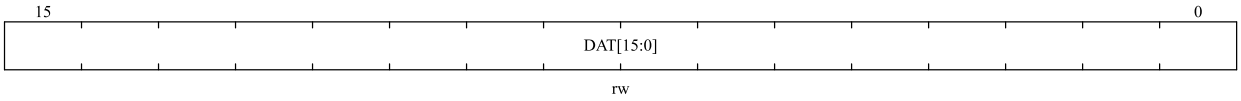


位域	名称	描述
15:8	Reserved	保留，硬件强制为 0。
7	BUSY	忙标志 0：SPI 不忙； 1：SPI 正忙于通信，或者发送缓冲非空。 该位由硬件置位或者复位。 *注：使用这个标志时需要特别注意，详见第 22.3.3 节和第 22.3.4 节。*
6	OVER	溢出标志 0：没有出现溢出错误； 1：出现溢出错误。 该位由硬件置位，由软件序列复位。关于软件序列的详细信息，参考 22.4.5 节。
5	MODERR	模式错误 0：没有出现模式错误； 1：出现模式错误。 该位由硬件置位，由软件序列复位。关于软件序列的详细信息，参考 22.3.7 节。 *注：I2S 模式下不使用。*
4	CRCERR	CRC 错误标志 0：收到的 CRC 值和 SPI_CRCRDAT 寄存器中的值匹配； 1：收到的 CRC 值和 SPI_CRCRDAT 寄存器中的值不匹配。 该位由硬件置位，由软件写'0'而复位。 *注：I2S 模式下不使用。*
3	UNDER	下溢标志位 0：未发生下溢； 1：发生下溢。 该标志位由硬件置'1'，由一个软件序列清'0'，详见 22.4.5 节。 *注：在 SPI 模式下不使用。*
2	CHSIDE	声道 0：需要传输或者接收左声道； 1：需要传输或者接收右声道。 *注：在 SPI 模式下不使用。在 PCM 模式下无意义。*

位域	名称	描述
1	TE	发送缓冲为空 0：发送缓冲非空； 1：发送缓冲为空。
0	RNE	接收缓冲非空 0：接收缓冲为空； 1：接收缓冲非空。

22.5.5 SPI 数据寄存器（SPI_DAT）

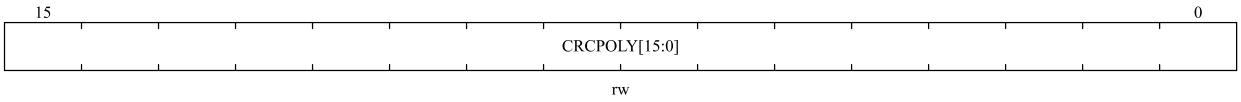
地址偏移：0x0C
复位值：0x0000



位域	名称	描述
15:0	DAT[15:0]	数据寄存器 待发送或者已经收到的数据 数据寄存器对应两个缓冲区：一个用于写（发送缓冲）；另外一个用于读（接收缓冲）。写操作将数据写到发送缓冲区；读操作将返回接收缓冲区里的数据。 对 SPI 模式的注释：根据 SPI_CTRL1 的 DATFF 位对数据帧格式的选择，数据的发送和接收可以是 8 位或者 16 位的。为保证正确的操作，需要在启用 SPI 之前就确定好数据帧格式。 对于 8 位的数据，缓冲器是 8 位的，发送和接收时只会用到 SPI_DAT[7:0]。在接收时，SPI_DAT[15:8]被强制为 0。 对于 16 位的数据，缓冲器是 16 位的，发送和接收时会用到整个数据寄存器，即 SPI_DAT[15:0]。

22.5.6 SPI CRC 多项式寄存器（SPI_CRCPOLY）（I2S 模式下不使用）

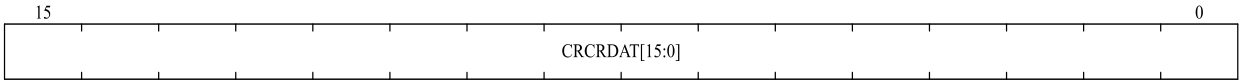
地址偏移：0x10
复位值：0x0007



位域	名称	描述
15:0	CRCPOLY [15:0]	CRC 多项式寄存器 该寄存器包含了 CRC 计算时用到的多项式。 其复位值为 0x0007，根据应用可以设置其他数值。 *注：在 I2S 模式下不使用。*

22.5.7 SPI Rx CRC 寄存器（SPI_CRCRDAT）（I2S 模式下不使用）

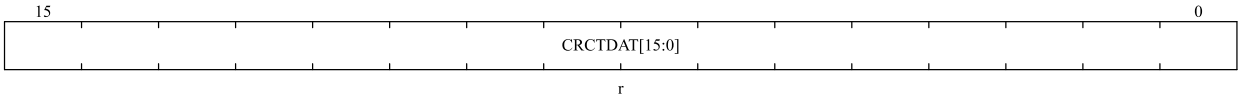
地址偏移：0x14
复位值: 0x0000



位域	名称	描述
15:0	CRCRDAT	接收 CRC 寄存器 在启用 CRC 计算时，CRCRDAT[15:0]中包含了依据收到的字节计算的 CRC 数值。当在 SPI_CTRL1 的 CRCEN 位写入'1'时，该寄存器被复位。CRC 计算使用 SPI_CRCPOLY 中的多项式。 当数据帧格式被设置为 8 位时，仅低 8 位参与计算，并且按照 CRC8 的方法进行；当数据帧格式为 16 位时，寄存器中的所有 16 位都参与计算，并且按照 CRC16 的标准。 *注：当 BUSY 标志为'1'时读该寄存器，将可能读到不正确的数值。* *注：在 I2S 模式下不使用。*

22.5.8 SPI Tx CRC 寄存器（SPI_CRCTDAT）

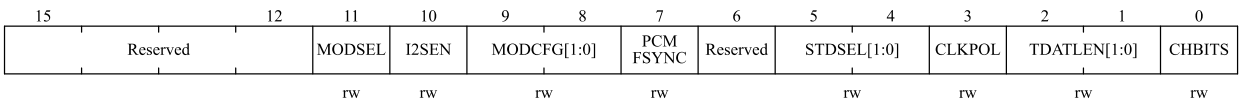
地址偏移：0x18
复位值：0x0000



位域	名称	描述
15:0	CRCTDAT	发送 CRC 寄存器 在启用 CRC 计算时，CRCTDAT[15:0]中包含了依据将要发送的字节计算的 CRC 数值。当在 SPI_CTRL1 中的 CRCEN 位写入'1'时，该寄存器被复位。CRC 计算使用 SPI_CRCPOLY 中的多项式。 当数据帧格式被设置为 8 位时，仅低 8 位参与计算，并且按照 CRC8 的方法进行；当数据帧格式为 16 位时，寄存器中的所有 16 个位都参与计算，并且按照 CRC16 的标准。 *注：当 BUSY 标志为'1'时读该寄存器，将可能读到不正确的数值。* *注：在 I2S 模式下不使用。*

22.5.9 SPI_I2S 配置寄存器（SPI_I2SCFG）

地址偏移：0x1C
复位值: 0x0000



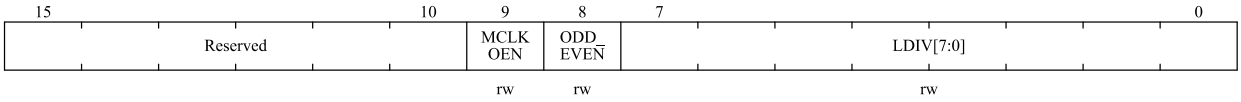
位域	名称	描述
15:12	Reserved	保留，硬件强制为 0。
11	MODSEL	I2S 模式选择 0: 选择 SPI 模式; 1: 选择 I2S 模式。 *注：该位只有在关闭了 SPI 或者 I2S 时才能设置。*
10	I2SEN	I2S 使能 0: 关闭 I2S; 1: I2S 使能。 *注：在 SPI 模式下不使用。*
9:8	MODCFG	I2S 模式设置 00: 从设备发送; 01: 从设备接收; 10: 主设备发送; 11: 主设备接受。 *注：该位只有在关闭了 I2S 时才能设置。* *在 SPI 模式下不使用。*
7	PCMFSYNC	PCM 帧同步 0: 短帧同步; 1: 长帧同步。 *注：该位只在 STDSEL = 11 （使用 PCM 标准）时有意义。* *在 SPI 模式下不使用。*
6	Reserved	保留，硬件强制为 0。
5:4	STDSEL	I2S 标准选择 00: I2S 飞利浦标准; 01: 高字节对齐标准 （左对齐）; 10: 低字节对齐标准（右对齐）; 11: PCM 标准。 关于 I2S 标准的细节，详见 22.4.1 节。 *注：为了正确操作，只有在关闭了 I2S 时才能设置该位。* *在 SPI 模式下不使用。*
3	CLKPOL	静止态时钟极性 0: I2S 时钟静止态为低电平; 1: I2S 时钟静止态为高电平。 *注：为了正确操作，该位只有在关闭了 I2S 时才能设置。* *在 SPI 模式下不使用。*
2:1	TDATLEN	待传输数据长度 00: 16 位数据长度; 01: 24 位数据长度; 10: 32 位数据长度; 11: 不允许。 *注：为了正确操作，该位只有在关闭了 I2S 时才能设置。* *在 SPI 模式下不使用。*

位域	名称	描述
0	CHBITS	声道长度 （每个音频声道的数据位数） 0: 16 位宽； 1: 32 位宽。 只有在 TDATLEN = 00 时该位的写操作才有意义，否则声道长度都由硬件固定为 32 位。 *注：为了正确操作，该位只有在关闭了 I2S 时才能设置。* *在 SPI 模式下不使用。*

22.5.10 SPI_I2S 预分频寄存器（SPI_I2SPREDIV）

地址偏移：0x20

复位值: 0x0002



位域	名称	描述
15:10	Reserved	保留，硬件强制为 0。
9	MCLKOEN	主设备时钟输出使能 0: 关闭主设备时钟输出； 1: 主设备时钟输出使能。 *注：为正确操作，该位只有在关闭 I2S 时才能设置。仅在 I2S 主设备模式下使用该位。* *在 SPI 模式下不使用。*
8	ODD _EVEN	奇系数预分频 0: 实际分频系数 = LDIV ×2； 1: 实际分频系数 = （LDIV × 2）+1。 参见 22.4.2 节。 *注：为正确操作，该位只有在关闭 I2S 时才能设置。仅在 I2S 主设备模式下使用该位。* *在 SPI 模式下不使用。*
7:0	LDIV	I2S 线性预分频 禁止设置 LDIV [7:0] = 0 或者 LDIV [7:0] = 1 参见 22.4.2 节。 *注：为正确操作，该位只有在关闭 I2S 时才能设置。仅在 I2S 主设备模式下使用该位。* *在 SPI 模式下不使用。*

23 I2C 接口

23.1 I2C 简介

I2C 总线是一种广泛应用的总线结构，它只有两根双向线，即数据总线 SDA 和时钟总线 SCL，通过这两根线，所有与 I2C 总线兼容的设备都可以通过 I2C 总线彼此直接通信。

I2C 接口连接微控制器和串行 I2C 总线，可用于 MCU 和外部 I2C 设备的通讯。I2C 接口模块实现了 I2C 协议的标速模式和快速模式，具备 CRC 计算和校验功能、支持 SMBus(系统管理总线)和 PMBus (电源管理总线)，此外它提供多主机功能，控制所有 I2C 总线特定的时序、协议、仲裁和定时。I2C 接口模块也支持 DMA 模式，可有效减轻 CPU 的负担。

23.2 I2C 主要特点

- 多主机功能：同一接口既可实现主机功能又可实现从机功能
- 是并行总线到 I2C 总线协议的转换器
- 支持 7 位和 10 位的地址模式和广播寻址
- 作为 I2C 主设备可以产生时钟、起始信号和停止信号
- 作为 I2C 从设备具有可编程的 I2C 地址检测、停止位检测的功能，以及可响应两个从地址的双地址能力
- 支持标速(最高 100 KHz)、快速(最高 400 KHz) 和快速+模式(最高 1MHz)
- 支持多种状态标志，比如：发送器/接收器模式标志、字节发送结束标志以及 I2C 总线忙标志。
- 支持多种错误标志，比如：主模式时的仲裁丢失标志、地址/数据传输后的应答（ACK/NAK）标志、检测到错位的起始或停止条件标志以及禁止拉长时钟功能时的上溢或下溢标志。
- 支持中断向量，字节成功发送中断和错误事件中断
- 可选的拉长时钟功能
- 支持 DMA 模式
- 可选择的 PEC（报文错误检测）生成和校验：发送模式中生成 PEC 并作为最后一个字节传输，接收模式中 PEC 作为最后一个接收字节来校验错误
- 兼容 SMBus 2.0 和 PMBus

注： 不是所有产品中都包含上述所有特性。请参考相关的数据手册，确认该产品支持的 I2C 功能。

23.3 I2C 功能描述

I2C 接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I2C 总线与外部设备进行通信，可以连接到标准（高达 100KHz）、快速（高达 400KHz）或快速+（高达 1MHz）的 I2C 总线。I2C 模块接收时将数据从串行转换成并行，发送时将数据从并行转换成串行。支持中断模式，用户可以根据需要开启或禁止中断。

23.3.1 SDA/SCL 线控制

I2C 模块有两条接口线：串行数据线（SDA）和串行时钟线（SCL）。连接到总线上的设备通过这两根线互相传递信息。SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。连接到总线的设备输出极必须带开漏或者开集，以提供线与功能。I2C 总线上的数据在标准模式下可以达到 100 kbit/s，在快速模式下可以达到 400kbit/s，在快速+模式下可以达到 1mbit/s。由于 I2C 总线上可能会连接不同工艺的设备，逻辑'0'和逻辑'1'的电平并不是固定的，取决于 VDD 的实际电平

如果允许时钟延长，即 SCL 线拉低，就可以避免在接收时发生过载错误以及在发送时发生欠载错误。

比如，在发送模式时，如果发送数据寄存器为空且字节发送结束位置起（TXDATE=1，BSF=1），I2C 接口在传输前保持时钟线为低，以等待软件读取 STS1 后把数据写进数据寄存器（缓冲器和移位寄存器都是空的）；在接收模式时，如果数据寄存器非空且字节发送结束位置起（RXDATNE =1，BSF=1），I2C 接口在接收到数据字节后保持时钟线为低，以等待软件读 STS1，然后读数据寄存器 DAT（缓冲器和移位寄存器都是满的）。

如果从模式中禁止时钟延长，在接收模式时，如果接收数据寄存器非空（RXDATNE=1），在接收到下个字节前数据还没有被读出，则发生过载错误，最后一字字节也将被丢弃。在发送模式时，如果发送数据寄存器空 (TXDATE=1)，在必须发送下个字节之前还没有新数据写进数据寄存器，则发生欠载错误。相同的字节将被重复发出。这种情况下不控制重复写冲突。

23.3.2 I2C 软件配置流程

I2C 设备的数据传输分为主机和从机。主机是指负责初始化总线上数据的传输并产生时钟信号的设备，此时任何被寻址的设备都是从机。不管 I2C 设备是主机还是从机，都可以发送或接收数据。因此，I2C 接口支持 4 种运行模式：

- 从机发送器模式
- 从机接收器模式
- 主机发送器模式
- 主机接收器模式

I2C 模块支持以上四种模式。系统复位后，I2C 默认工作在从机模式下。通过软件配置 I2C 接口在总线上发送一个起始位，随后接口自动地从从模式切换到主模式；当仲裁丢失或产生停止信号时，则从主模式切换到从模式。

23.3.2.1 开始和停止条件

所有的数据传输总是以起始位开始并以停止位结束。起始条件和停止条件都是在主模式下由软件控制产生。起始位 (START) 是指：在 SCL 为高时，SDA 线上出现一个从高到低的电平转换。停止位 (STOP) 是指在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。

23.3.2.2 时钟同步和仲裁

I2C 接口允许多主机功能，即两个主机可以同时在空闲总线上开始传送数据，因此就必须通过一些机

制来决定哪个主机获取总线的控制权，这一般是通过时钟同步和仲裁来完成的。单主机系统下不需要时钟同步和仲裁机制。

I2C 电路具有两个特点：

- SDA 和 SCL 为漏极开路结构，通过外部的上拉电阻实现了信号的“线与”逻辑；
- SDA 和 SCL 引脚在输出信号的同时还将引脚上的电平进行检测，检测是否与刚才输出一致。这为“时钟同步”和“总线仲裁”提供硬件基础。

I2C 设备对总线的操作是通过“把线路接地”来输出逻辑 0。基于 I2C 总线的特点，如果一设备发送逻辑 0，其他发送逻辑 1，那么线路看到的只有逻辑 0，所以线路上不可能出现电平冲突的现象。

总线的物理接法允许主设备往总线写数据的同时读取数据。这样两主设备争总线的时候发送逻辑 0 的并不知道竞争的发生，只有发送逻辑 1 的才会发现冲突（当写一个逻辑 1，却读到了 0）从而退出竞争。

时钟同步

SCL 线的高到低切换会使器件开始数它们的低电平周期，而且一旦器件的时钟变低电平，它会使 SCL 线保持这种状态直到到达时钟的高电平。但是，如果另一个时钟仍处于低电平周期，这个时钟的低到高切换不会改变 SCL 线的状态，因此，SCL 线被有最长低电平周期的器件保持低电平，此时，低电平周期短的器件会进入高电平的等待状态。

当所有有关的器件数完了它们的低电平周期后，时钟线被释放并且变成高电平，之后器件时钟和 SCL 线的状态没有差别，而且所有器件会开始数它们的高电平周期，首先完成高电平周期的器件会再次将 SCL 线拉低。

这样，产生的同步 SCL 时钟的低电平周期由低电平时钟周期最长的器件决定，而高电平周期由高电平时钟周期最短的器件决定。

仲裁

仲裁和同步一样，都是为了解决多主机情况下的总线控制冲突。仲裁的过程与从机无关。当两个主机在总线空闲的时候都产生了一个有效的起始位，这种情况就需要决定由哪个主机来完成数据传输，这就是仲裁的过程。

各个主控制器没有对总线实施控制的优先级别，都是由仲裁决定的。总线控制随即而定且逐位进行，他们遵循“低电平优先”的原则，即谁先发送低电平谁就会掌握对总线的控制权。在每一位的仲裁期间，当 SCL 为高时，每个主机都检查自己的 SDA 电平是否和自己发送的相同。理论上讲，如果两个主机所传输的内容完全相同，那么他们能够成功传输而不出现错误。如果一个主机发送高电平但检测到 SDA 电平为低，则认为自己仲裁失败并关闭自己的 SDA 输出驱动，而另一个主机则继续完成自己的传输。

23.3.2.3 I2C 数据通信流程

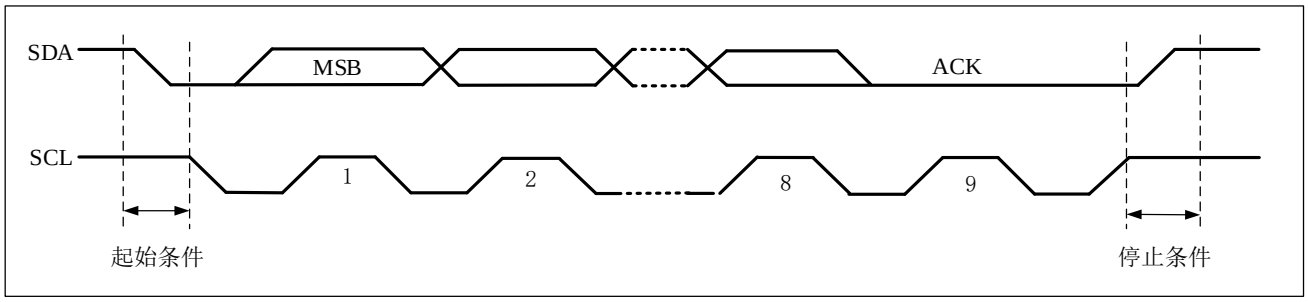
每个 I2C 设备都通过唯一的地址进行识别，根据设备功能，他们既可以是发送器也可作为接收器。

I2C 主机负责产生起始位和结束位来开始和结束一次传输，并且负责产生 SCL 时钟。

I2C 模块支持 7 位和 10 位的地址，用户可以通过软件配置 I2C 从机的地址。I2C 从机检测到 I2C 总线上的起始位之后，就开始从总线上接收地址，并把接收到的地址和自身的地址进行比较，一旦两个地址相同， I2C 从机将发送一个确认应答(ACK)，并响应总线的后续命令：发送或接受所要求的数据。此外，如果软件开启了广播呼叫，则 I2C 从机始终对一个广播地址 (0x00)发送确认应答。

数据和地址按 8 位字节进行传输，高位在前。跟在起始条件后的 1 或 2 个字节是地址（7 位模式为 1 个字节，10 位模式为 2 个字节）。地址只在主模式发送。在一个字节传输的 8 个时钟后的第 9 个时钟期间，接收器必须回送一个应答位（ACK）给发送器。如下图所示。

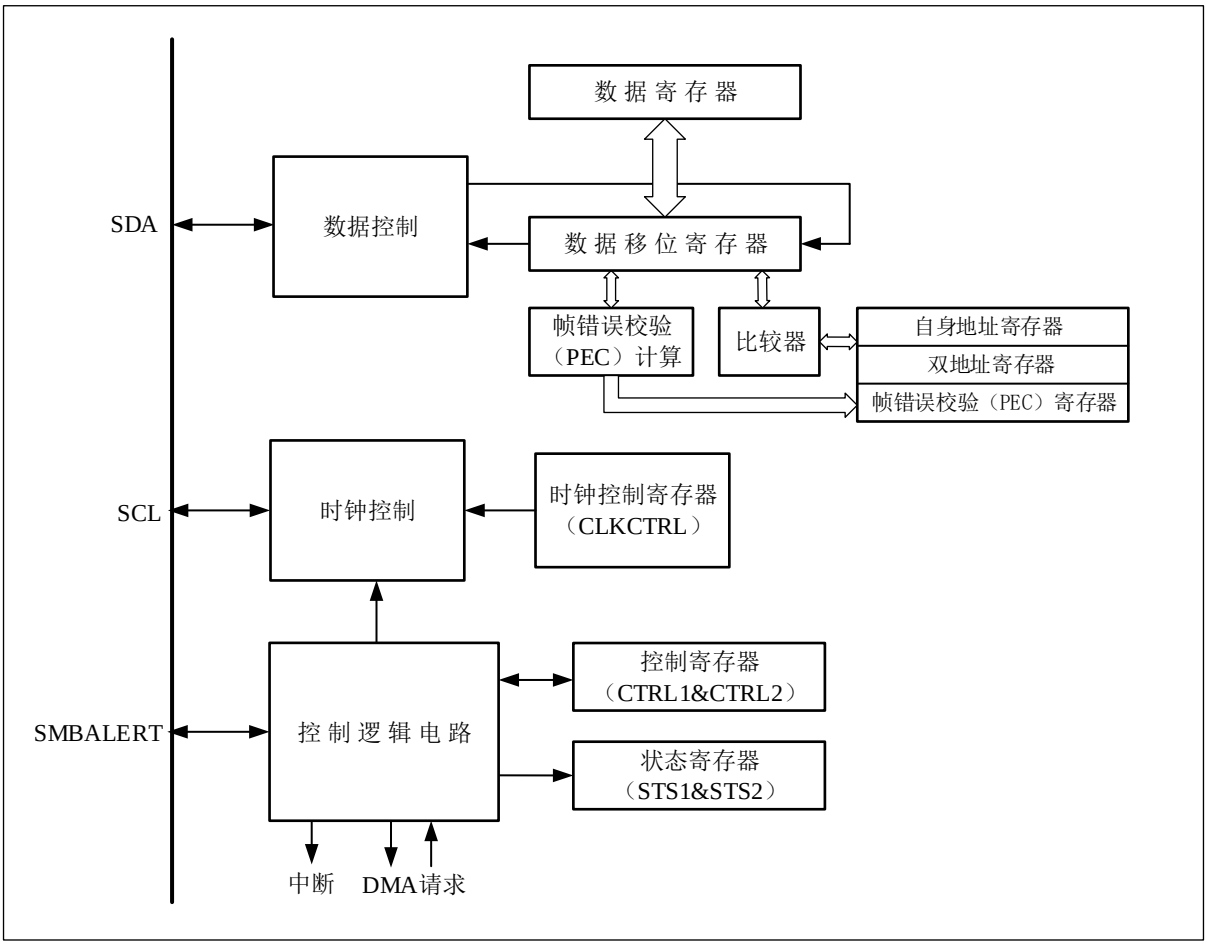
图 23-1 I2C 总线协议



软件可以开启或禁止应答（ACK），并可以设置 I2C 接口的地址（7 位、 10 位地址或广播呼叫地址）。

I2C 接口的功能框图示于下图。

图 23-2 I2C 的功能框图



注：在 SMBus 模式下，SMBALERT 是可选信号。如果禁止了 SMBus，则不能使用该信号.

23.3.2.4 I2C 从机发送模式

在从机模式下发送接收标记位（TRF）指示当前是处于接收器模式还是发送器模式。当在发送器模式下要发送数据到 I2C 总线，软件应该按照下面的步骤操作:

1、 首先，使能 I2C 外设时钟，配置 I2C_CTRL1 中时钟相关寄存器，确保输出正确的 I2C 时序。当这两步都完成以后，I2C 运行在从机模式下，等待接收起始位和地址。

2、 I2C 从机先收到一个起始位，随后收到匹配的 7 位或 10 地址，I2C 硬件将状态寄存器（I2C_STS1）的 ADDRF 位（接收到了地址并且和自身的地址匹配）置 1，软件应该定期查询此位或者中断监视此位，发现置位后，软件读 I2C_STS1 寄存器然后读 I2C_STS2 寄存器来清除 ADDRF 位。如果地址是 10 位格式，I2C 主机应该接着再产生一个 START（Sr）并发送一个地址头到 I2C 总线。从机在检测到 START（Sr）和紧接着的地址头之后会继续将 ADDRF 位置 1。软件继续通过读 I2C_STS1 寄存器和接着读 I2C_STS2 寄存器来第二次清除 ADDRF 位。

3、 I2C 进入数据发送状态，现在移位寄存器和数据寄存器 I2C_DAT 都是空的，所以硬件将 STS1 寄存器的 TXDATE（发送数据空）位置 1。此时软件可以写入第一个字节数据到 I2C_DAT 寄存器，但是，因为写入 I2C_DAT 寄存器的字节被立即移入内部移位寄存器了，所以 TXDATE 位并没有被清 0。当移位寄存器非空的时候，I2C 开始发送数据到 I2C 总线。

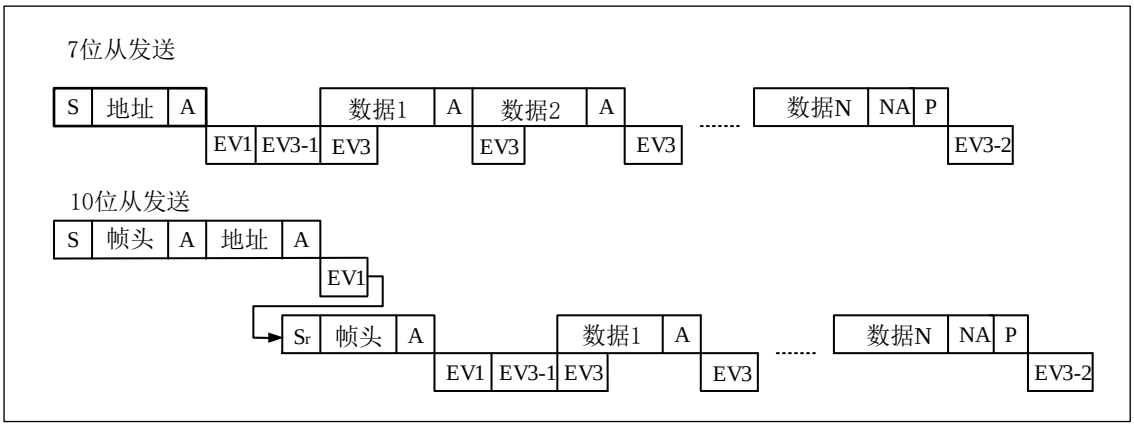
4、 第一个字节的发送期间，软件写第二个字节到 I2C_DAT，此时 I2C_DAT 寄存器和移位寄存器都不是空。TXDATE 位被清 0。

5、 第一个字节发送完成之后，TXDATE 再次被置起，软件写第三个字节到 I2C_DAT，同时 TXDATE 位被清 0。在此之后，只要依然有数据待等待被发送且 TXDATE 被置 1，软件都可以写入一个字节到 I2C_DAT 寄存器。

6、 倒数第二个字节发送期间，软件写最后一个数据到 I2C_DAT 寄存器来清除 TXDATE 标志位，之后就再不用关心 TXDATE 的状态。TXDATE 位会在倒数第二个字节发送完成后置起，直到检测到 STOP 结束位时被清 0。

7、 根据 I2C 协议，I2C 主机不会对接收到的最后一个字节发送应答，所以在最后一个字节发送结束后，I2C 从机的 ACKFAIL 位（应答出错）会置起以通知软件发送结束。软件写 0 到 ACKFAIL 位可以清除此位

图 23-3 从发送器的传送序列图



说明:

1. S=Start（起始条件），Sr=重复的起始条件，P=Stop（停止条件），A=响应，NA=非响应，EVx=事件（EVTINTEN=1 时产生中断）
2. EV1：ADDRF=1，读 STS1 然后读 STS2 将清除该事件。
3. EV3-1：TXDATE=1，移位寄存器空，数据寄存器空，写 DAT。
4. EV3：TXDATE=1，移位寄存器非空，数据寄存器空，写 DAT 将清除该事件。
5. EV3-2：ACKFAIL=1，在 STS1 寄存器的 ACKFAIL 位写'0'可清除 ACKFAIL 位。

注：a）EV1 和 EV3_1 事件拉长 SCL 低的时间，直到对应的软件序列结束。
b）EV3 的软件序列必须在当前字节传输结束之前完成。

23.3.2.5 I2C 从机接收模式

在从机模式下接收数据时，软件应该按如下步骤操作:

1、首先，使能 I2C 外设时钟，配置 I2C_CTRL1 中时钟相关寄存器，确保输出正确的 I2C 时序。当这两步都完成以后，I2C 运行在从机模式下，等待接收起始位和地址。

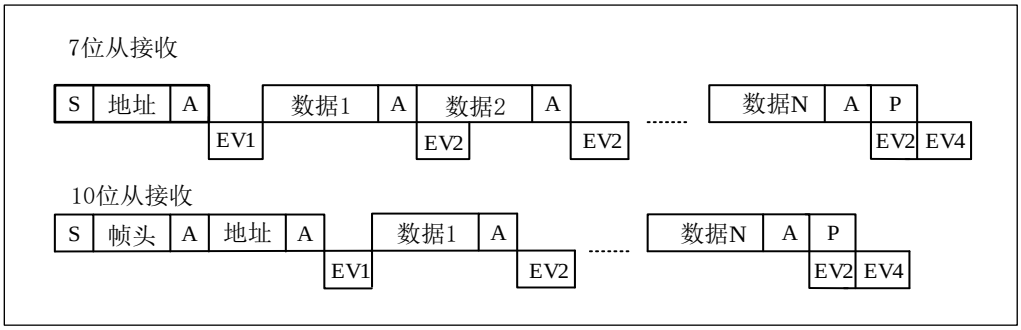
2、在接收到 START 起始条件和匹配的 7 位或 10 地址之后，I2C 硬件将 I2C 状态寄存器（I2C_STS1）的 ADDRF 位（接收到了地址并且和自身的地址匹配）置 1，此位应该通过软件轮询或者中断来检测，发现置起后，软件通过先读 I2C_STS1 寄存器然后再读 I2C_STS2 寄存器来清除 ADDRF 位。一旦 ADDRF 位被清 0，I2C 从机就开始接收来自 I2C 总线的数据。

3、当接收到第一个字节后，RXDATNE 位（接收数据非空）被硬件置 1，如果设置了 EVTINTEN 和 BUFINTEN 位，则产生一个中断。软件应该通过轮询或者中断来检测该位，一旦发现置起后，软件可以读取 I2C_DAT 寄存器的第一个字节，此时 RXDATNE 位被清 0。注意，如果设置了 ACKEN 位，则在接收到一个字节后，从机应该产生一个应答脉冲

4、任何时候，只要 RXDATNE 位被置 1，软件均可以从 I2C_DAT 寄存器读取一个字节。当接收到最后一个字节后，RXDATNE 被置 1，软件读取最后一个字节

5、当从机检测到 I2C 总线上的停止位（STOP）后，将 STOPF 位置 1，如果设置了 EVTINTEN 位，则产生一个中断。软件通过先读 I2C_STS1 寄存器再写 I2C_CTRL1 寄存器来清除 STOPF 位（(见下图的 EV4)）。

图 23-4 从接收器的传送序列图



说明:

1. S=Start（起始条件），Sr=重复的起始条件，P=Stop（停止条件），A=响应，NA=非响应，EVx=事件（EVTINTEN=1 时产生中断）

2. EV1：ADDRF=1，读 STS1 然后读 STS2 将清除该事件。
3. EV2：RXDATNE =1，读 DAT 将清除该事件。
4. EV4：STOPF=1，读 STS1 然后写 CTRL1 寄存器将清除该事件。

注：a）EV1 事件拉长 SCL 低的时间，直到对应的软件序列结束。
b）EV2 的软件序列必须在当前字节传输结束之前完成。

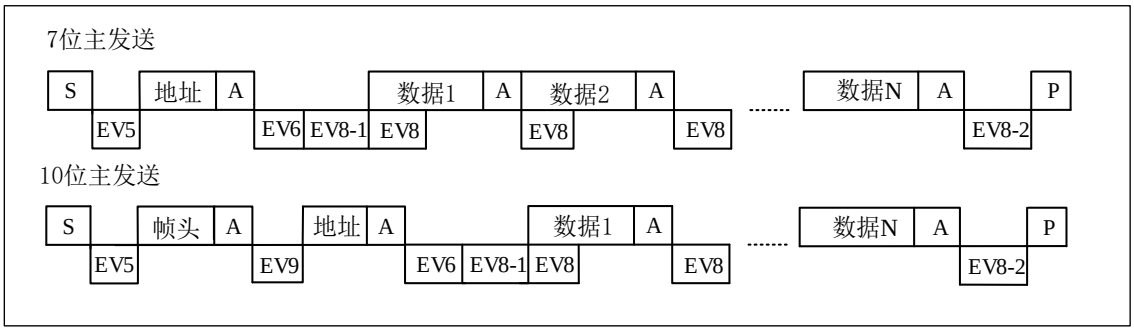
23.3.2.6 I2C 主机发送模式

在主模式时，I2C 接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。当通过 START 位在总线上产生了起始条件，设备就进入了主模式。

在主机模式下发送数据到 I2C 总线时，软件应该按如下步骤操作:

- 首先，使能 I2C 外设时钟，配置 I2C_CTRL1 中时钟相关寄存器，确保输出正确的 I2C 时序。当这两步都完成以后，I2C 默认运行在从机模式下，等待接收起始位和地址。
- 当 BUSY=0 时，设置 I2C_CTRL1 寄存器中的 STARTGEN 位为 1，I2C 接口将产生一个开始条件并切换至主模式（MSMODE 位置位），
- 一旦发出开始条件，I2C 硬件将 I2C_STS1 的 STARTBF 位（起始位标志）置 1 然后进入主机模式，如果设置了 EVTINTEN 位，则会产生一个中断。接着软件读 I2C_STS1 寄存器然后写一个 7 位地址位或带有地址头的 10 位地址位到 I2C_DAT 寄存器来清除 STARTBF 位。STARTBF 位被清 0 后 I2C 就开始发送地址或者地址头到 I2C 总线。如果发送的地址是 10 位带地址头的地址，硬件在发送地址头的时候会将 ADDR10F 位置 1，如果设置了 EVTINTEN 位，则产生一个中断，然后通过读 I2C_STS1 寄存器再写 10 位低地址到 I2C_DAT 来清除 ADDR10F 位。
- 7 位或 10 位的地址位发送完成后，I2C 硬件将 ADDRF 位（地址已被发送）置 1，如果设置了 EVTINTEN 位，则产生一个中断，软件通过读 I2C_STS1 寄存器然后读 I2C_STS2 寄存器来清除 ADDRF
- I2C 进入数据发送状态，因为移位寄存器和数据寄存器（I2C_DAT）都是空的，所以硬件将 TXDATE 位（发送数据空）置 1，接着软件写第一个字节数据到 I2C_DAT 寄存器，但是，因为写入 I2C_DAT 寄存器的字节被立即移入内部移位寄存器，所以 TXDATE 位此时不会被清零。一旦移位寄存器非空，I2C 就开始发送数据到总线。
- 在第一个字节的发送过程中，软件写第二个字节到 I2C_DAT，此时 TXDATE 被清零。任何时候，只要还有数据等待被发送，且 TXDATE 位被置 1，软件都可以向 I2C_DAT 寄存器写入一个字节。
- 在倒数第二个字节发送过程中，软件写入最后一个字节数据到 I2C_DAT 来清除 TXDATE 标志位，此后就不用关心 TXDATE 位的状态。TXDATE 位会在倒数第二个字节发送完成后被置起，直到发送停止位（STOP）时被清零。
- 最后一个字节发送结束后，因为移位寄存器和 I2C_DAT 寄存器此时都为空，I2C 主机将 BSF 位（字节发送结束）置位，在清除 BSF 位之前 I2C 接口将保持 SCL 为低电平；读出 I2C_STS2 之后再写入 I2C_DAT 寄存器将清除 BSF 位。软件此时设置 STOPGEN 位产生一个停止条件（见图 23-5 的 EV8_2），然后 I2C 接口将自动回到从模式（MSMODE 位清除）。

图 23-5 主发送器传送序列图



说明:

1. S=Start（起始条件），Sr=重复的起始条件，P=Stop（停止条件），A=响应，NA=非响应，EVx=事件（EVTINTEN=1 时产生中断）。
2. EV5：STARTBF=1，读 STS1 然后将地址写入 DAT 寄存器将清除该事件。
3. EV6：ADDRF=1，读 STS1 然后读 STS2 将清除该事件。
4. EV8_1：TXDATE=1，移位寄存器空，数据寄存器空，写 DAT 寄存器。
5. EV8：TXDATE=1，移位寄存器非空，数据寄存器空，写入 DAT 寄存器将清除该事件。
6. EV8_2：TXDATE =1，BSF=1，请求设置停止位。TXDATE 和 BSF 位由硬件在产生停止条件时清除。
7. EV9：ADDR10F =1，读 STS1 然后写入 DAT 寄存器将清除该事件。

注：a）EV5、 EV6、 EV9、 EV8_1 和 EV8_2 事件拉长 SCL 低的时间，直到对应的软件序列结束。
b）EV8 的软件序列必须在当前字节传输结束之前完成。
C）当 TXDATE 或 BSF 位置位时，停止条件应安排在出现 EV8_2 事件时。

23.3.2.7 I2C 主机接收模式

在主机模式下从 I2C 总线接收数据软件应该按如下步骤操作:

1、 首先，使能 I2C 外设时钟，配置 I2C_CTRL1 中时钟相关寄存器，确保输出正确的 I2C 时序。使能和配置以后，I2C 默认运行在从机模式下，等待接收起始位和地址。

2、 当 BUSY=0 时，设置 I2C_CTRL1 寄存器中的 STARTGEN 位为 1，I2C 接口将产生一个开始条件并切换至主模式（MSMODE 位置位），

3、 一旦发出开始条件，I2C 硬件将 I2C_STS1 的 STARTBF 位（起始位标志）置 1 然后进入主机模式，如果设置了 EVTINTEN 位，则会产生一个中断。接着软件读 I2C_STS1 寄存器然后写一个 7 位地址位或带有地址头的 10 位地址位到 I2C_DAT 寄存器来清除 STARTBF 位。STARTBF 位被清 0 后 I2C 就开始发送地址或者地址头到 I2C 总线。如果发送的地址是 10 位带地址头的地址，硬件在发送地址头的时候会将 ADDR10F 位置 1，如果设置了 EVTINTEN 位，则产生一个中断，然后通过读 I2C_STS1 寄存器再写 10 位低地址到 I2C_DAT 来清除 ADDR10F 位。

4、 7 位或 10 位的地址位发送完成后，I2C 硬件将 ADDRF 位（地址已被发送）置 1，如果设置了 EVTINTEN 位，则产生一个中断，软件通过读 I2C_STS1 寄存器然后读 I2C_STS2 寄存器来清除 ADDRF 在发送地址和清除 ADDRF 之后，如果地址是 10 位格式，软件应该再次将 STARTGEN 位置 1 来重新产生一个 START（Sr）。在 START 产生后，STARTBF 位会被置 1。软件应该通过先读 I2C_STS1 然后写地

址头到 I2C_DAT 来清除 STARTBF 位，然后地址头被发到 I2C 总线，ADDRF 再次被置 1。软件应该再次通过先读 I2C_STS1 然后读 I2C_STS2 来清除 ADDRF 位。

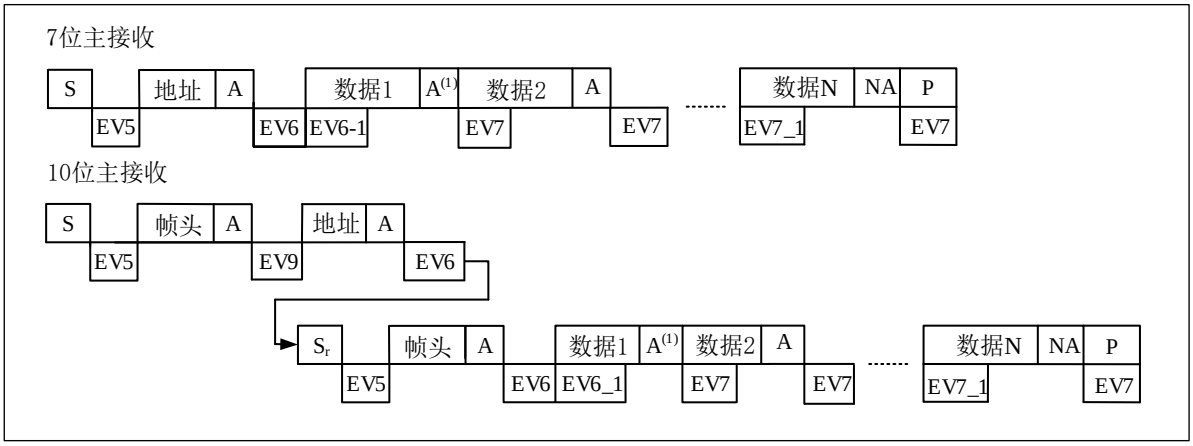
5、 在发送完地址和清除 ADDRF 之后，I2C 接口进入主机接收器模式。在此模式下，I2C 接口从 SDA 线接收数据字节，并通过内部移位寄存器送至 DAT 寄存器。一旦接收到第一个字节，硬件会将 RXDATNE 位（接收数据非空标志位）置 1，如果 ACKEN 位被置位，发出一个应答脉冲。此时软件可以从 I2C_DAT 寄存器读取第一个字节，之后 RXDATNE 位被清 0。此后，只要 RXDATNE 被置 1，软件就可以从 I2C_DAT 寄存器读取一个字节。

6、 主设备在从从设备接收到最后一个字节后发送一个 NACK。接收到 NACK 后，从设备释放对 SCL 和 SDA 线的控制；主设备就可以发送一个停止/重新起始条件。为了在收到最后一个字节后产生一个 NACK 脉冲，接收完倒数第二个字节(N-1)数据之后，软件应该立即清除 ACKEN 位。为了产生一个停止/重新起始条件，软件必须在读倒数第二个数据字节之后将 STOPGEN 位或者 STARTGEN 置 1，这一过程需要在最后一个字节接收完毕之前完成，以确保 NACK 发送给最后一个字节。

7、 最后一个字节接收完毕后，RXDATNE 位被置 1，软件可以读取最后一个字节。由于 ACKEN 已经在前一步骤中被清 0，I2C 不再为最后一个字节发送 ACK，并在最后一个字节发送完毕后产生一个 STOP 停止位。

注意：以上步骤要求字节数目 N>1，如果 N=1，步骤 6 应该在步骤 4 之后就执行，且需要在字节接收完成之前完成。

图 23-6 主接收器传送序列图



说明:

1. S=Start（起始条件），Sr=重复的起始条件，P=Stop（停止条件），A=响应，NA=非响应，EVx=事件（EVTINTEN=1 时产生中断）
2. EV5：STARTBF=1，读 STS1 然后将地址写入 DAT 寄存器将清除该事件。
3. EV6：ADDRF=1，读 STS1 然后读 STS2 将清除该事件。在 10 位主接收模式下，该事件后应设置 CTRL1 的 STARTGEN =1。
4. EV6_1：没有对应的事件标志，只适于接收 1 个字节的情况。恰好在 EV6 之后（即清除了 ADDRF 之后），要清除响应和停止条件的产生位。
5. EV7：RXDATNE=1，读 DAT 寄存器清除该事件。
6. EV7_1：RXDATNE =1，读 DAT 寄存器清除该事件。设置 ACKEN=0 和 STOPGEN 请求。

7. EV9：ADDR10F=1，读 STS1 然后写入 DAT 寄存器将清除该事件。

注：

如果收到一个单独的字节，则是 NA。

EV5、EV6 和 EV9 事件拉长 SCL 低电平，直到对应的软件序列结束。

EV7 的软件序列必须在当前字节传输结束前完成。

EV6_1 或 EV7_1 的软件序列必须在当前传输字节的 ACK 脉冲之前完成。

23.3.3 错误条件

I2C 的错误主要有总线错误、应答错误、仲裁丢失、过载\欠载错误。这些错误都可能造成通讯的失败。

23.3.3.1 总线错误（BUSERR）

在一个地址或数据字节传输期间，当 I2C 接口检测到一个外部的停止或起始条件则产生总线错误。此时：

- BUSERR 位被置位为‘1’；如果设置了 ERRINTEN 位，则产生一个中断；
- 在主模式情况下，硬件不释放总线，同时不影响当前的传输状态。此时由软件决定是否要中止当前的传输。
- 在从模式情况下，数据被丢弃，硬件释放总线。如果检测到错误的开始条件，从设备认为是一个重启动，并等待地址或停止条件。如果检测到错误的停止条件，从设备按正常的停止条件操作，同时硬件释放总线。

23.3.3.2 应答错误（ACKFAIL）

当接口检测到应答位与期望不符时，产生应答错误。此时：

- ACKFAIL 位被置位，如果设置了 ERRINTEN 位，则产生一个中断；
- 当发送器接收到一个 NACK 时，必须复位通讯：如果是处于从模式，硬件释放总线。如果是处于主模式，软件必须生成一个停止条件。

23.3.3.3 仲裁丢失（ARLOST）

当 I2C 接口检测到仲裁丢失时产生仲裁丢失错误，此时：

- ARLOST 位被硬件置位，如果设置了 ERRINTEN 位，则产生一个中断；
- I2C 接口自动回到从模式（MSMODE 位被清除）。当 I2C 接口丢失了仲裁，则它无法在同一个传输中响应它的从地址，但它可以在赢得总线的主设备重新发送起始条件之后响应；
- 硬件释放总线。

23.3.3.4 过载/欠载错误（OVERRUN）

在从机接收模式下，如果禁止时钟延长，当它已经接收到一个字节（RXDATNE），但在 DAT 寄存器中前一个字节数据还没有被读出，则发生过载错误。此时，最后接收的数据被丢弃；同时软件应清除 RXDATNE 位，发送器重新发送最后一次发送的字节。

在从机发送模式下，如果禁止时钟延长，在当前字节已经发送完成，而 DAT 仍然为空，(TXDATE=1)，

则发生欠载错误。此时，在 DAT 寄存器中的前一个字节将被重复发出；用户应该确定在发生欠载错时，接收端应丢弃重复接收到的数据。发送端应按 I2C 总线标准在规定的时间更新 DAT 寄存器。

在发送第一个字节时，必须在清除 ADDRF 之后并且第一个 SCL 上升沿之前写入 DAT 寄存器；如果不能做到这点，则接收方应该丢弃第一个数据。

23.3.4 SMBus

23.3.4.1 介绍

系统管理总线（System Management Bus，简写为 SMBus 或 SMB）是一种结构简单的单端双线制总线。通过它，各设备之间以及设备与系统的其他部分之间可以互相通信。SMBus 是 I2C 的一种衍生总线形式，为系统和电源管理相关的任务提供一条控制总线。

一个系统利用SMBus可以和多个设备互传信息，而不需使用独立的控制线路。系统管理总线(SMBus)标准涉及三类设备。

- 从设备：接收或响应命令的设备。
- 主设备：用来发送命令、产生时钟和终止发送的设备。
- 主机：一种专用的主设备，它提供与系统 CPU 的主接口。主机必须具有主-从机功能并且必须支持 SMBus 提醒协议。一个系统里只允许有一个主机。

23.3.4.2 SMBus 和 I2C 之间的相似点

SMBus 是 I2C 规范中数据传输格式的子集，SMBus 和 I2C 有以下几点相似:

- 总线协议都是 2 条线（一个时钟线（CLK）和一个数据线（SDA)），再加可选的 SMBus 提醒线;
- 数据格式相似，SMBus 数据格式类似于 I2C 的 7 位地址格式（见图 23-1);
- 都是主-从通信模式，且主设备提供时钟;
- 都支持多主机功能

23.3.4.3 SMBus 和 I2C 之间的不同点

下表列出了 SMBus 和 I2C 的不同点。

表 23-1 SMBus 与 I2C 的比较

SMBus	I2C
最大传输速度 100KHz	最大传输速度 1MHz
最小传输速度 10KHz	无最小传输速度
35ms 时钟低超时	无时钟超时
固定的逻辑电平	逻辑电平由 VDD 决定
不同的地址类型（保留的、动态的等）	7 位、10 位和广播呼叫从地址类型
不同的总线协议（快速命令、处理呼叫等）	无总线协议

23.3.4.4 SMBus 应用用途

SMBus 利用系统管理总线，可实现轻量级的通信需求。一般来说，SMBus 最常见于计算机主板，主要用于电源传输 ON/OFF 指令的通信，为系统和电源管理相关的任务提供控制总线。

23.3.4.5 设备标识

在系统管理总线上，任何一个作为从模式的设备都有一个唯一的地址，叫做从地址。保留的从地址表请参考 2.0 版的 SMBus 规范（http://smbus.org/specs/）

23.3.4.6 总线协议

SMBus 技术规范支持 9 个总线协议。有关这些协议的详细资料和 SMBus 地址类型，请参考 2.0 版的 SMBus 规范（http://smbus.org/specs/）。这些协议由用户的软件来执行。

SMBus 上每个报文交互都遵从 SMBus 协议中预定义的格式。SMBus 是 I2C 规范中数据传输格式的子集。只要 I2C 设备可通过 SMBus 协议之一进行访问，便视为兼容 SMBus 规范。

23.3.4.7 地址解析协议（ARP）

SMBus 规范包含一个地址解析协议，该协议通过给每个从设备动态地分配一个新的唯一地址，可以有效的解决从地址冲突的问题。地址解析协议（ARP）具有以下的特性：

- 使用标准 SMBus 物理层仲裁机制分配地址；
- 当设备维持供电期间，分配的地址仍保持不变，也允许设备在断电后保留其地址。
- 在地址分配以后，没有额外的 SMBus 的打包开销，也就是说访问分配地址的设备与访问固定地址的设备所用时间是一样的；
- 任何一个 SMBus 主设备都可以遍历总线。

23.3.4.8 唯一的设备标识符（UDID）

每个设备必须拥有一个唯一的设备标识符 UDID，用来区分不同的设备。关于在 ARP 上 128 位的 UDID 的详细信息，参考 2.0 版的 SMBus 规范（http://smbus.org/specs/）。

23.3.4.9 SMBus 提醒模式

SMBus 还有一个额外的共享的中断信号，称为 SMBALERT，该信号用于那些希望扩展它们的控制能力而牺牲一个引脚的设备。 SMBALERT 和 SCL、 SDA 信号一样，是一种线与信号。 SMBALERT 通常和 SMBus 广播呼叫地址一起使用。与 SMBus 有关的消息为 2 字节。

一个只具有从功能的设备，可以通过设置 I2C_CTRL1 寄存器上的 SMBALERT 位，使用 SMBALERT 给主机发信号通知主机它希望进行通信。主机处理该中断并通过提醒响应地址 ARA（Alert Response Address，地址值为 0001100x）访问所有 SMBALERT 设备。只有那些将 SMBALERT 拉低的设备能应答 ARA。此状态是由 I2C_STS1 寄存器中的 SMBALERT 状态标记来标识的。主机执行一个修改过的接收字节操作。由从发送设备提供的 7 位设备地址被放在字节的 7 个最高位上，第八个位可以是'0'或'1'。

如果多个设备把 SMBALERT 拉低，具有最小地址的设备拥有最高优先级，该设备将在地址传输期间通过标准仲裁赢得通信权。在确认从地址后，此设备不得再拉低它的 SMBALERT，如果当信息传输完成后，主机仍看到 SMBALERT 低，就知道需要再次读 ARA。没有实现 SMBALERT 信号的主机可以定期访问 ARA。

有关 SMBus 提醒模式的更多详细资料，请参考 2.0 版的 SMBus 规范（http://smbus.org/specs/）。

23.3.4.10 超时错误

SMBus 有一种超时特性：假如某个通信耗时太久，便会自动复位设备。这就是为什么 SMBus 有最小传输速率的要求——为了防止超时后长时间锁死总线。 I2C 在本质上可以视为一个“直流”总线，也就

是说当主机正在访问从机的时候，假如从机正在执行一些子程序无法及时响应，从机可以拉住主机的时钟。这样便可以提醒主机：从机正忙，但并不想放弃当前的通信。从机的当前任务结束之后，将可以继续 I2C 会话。 I2C 总线协议中并没有限制这个延时的上限，但在 SMBus 系统中，这个时间被限定为 35ms。按照 SMBus 协议的假定，如果某个会话耗时太久，就意味着总线出了问题，此时所有设备都应当复位以消除这种（问题）状态。这样就不允许从设备将时钟拉低太长时间。I2C_STS1 中的状态标志 TIMOUT 错误表明了这个特性的状态。更多超时细节请参考 2.0 版的 SMBus 规范（http://smbus.org/specs/）。

23.3.4.11 SMBus 通信流程

SMBus 的通讯流程和标准 I2C 的流程相似。如果要使用 SMBus 模式，在程序中需要配置几个 SMBus 特定的寄存器、响应一些 SMBus 特定标志位、实现那些在 SMBus 手册中介绍的上层协议。

- 首先，设置 I2C_CTRL1 寄存器中的 SMBMODE 位；并按照应用要求配置 I2C_CTRL1 寄存器中的 SMBTYPE 和 ARPEN 位。如果 ARPEN=1 且 SMBTYPE=0，使用 SMB 设备默认地址；如果 ARPEN=1 且 SMBTYPE=1，使用 SMB 主设备头字段。
- 为了支持 ARP 协议（ARPEN=1），在 SMBus 主机模式下（SMBTYPE=1），软件需要响应标志位 SMBHADDR（在 SMBus 从机模式下，响应 SMBDADDR 标志位），并实现 ARP 协议中的功能。
- 为了支持 SMBus 警告模式，软件应该响应 SMBALERT 标志位，并实现相应的功能。

23.3.5 DMA 请求

DMA 请求仅用于数据传输。按照前面的 I2C 通信流程描述每当发送数据寄存器变空或接收数据寄存器变满 之后，软件都应该写或读一个字节，这样将导致 CPU 的负荷较重。I2C 的 DMA 功能可以在发送数据寄存器变空或接收数据寄存器变满时，将产生 DMA 请求，DMA 自动进行一次写或读操作，从而减轻了 CPU 的负荷。

DMA 请求必须在当前字节传输结束之前被响应。当相应 DMA 通道设置的数据传输量已经完成时，DMA 控制器发送传输结束信号 ETO 到 I2C 接口，并且在中断允许时产生一个传输完成（EOT）中断：

- 如果是主机发送模式，在 EOT 中断服务程序中，需禁止 DMA 请求，然后在等到 BSF 事件后设置停止条件。
- 如果是主机接收模式，当要接收的数据数目大于或等于 2 时，DMA 控制器发送一个硬件信号 EOT_1，它对应 DMA 传输（字节数－1）。如果在 I2C_CTRL2 寄存器中设置了 DMALAST 位，硬件在发送完 EOT_1 后的下一个字节，将自动发送 NACK。在中断允许的情况下，用户可以在 DMA 传输完成的中断服务程序中产生一个停止条件。

23.3.5.1 DMA 发送

DMA 模式通过 I2C_CTRL2 寄存器中的 DMAEN 位使能。该位应该在清除 ADDRF 状态位后被置位，如果一个从机的 SCL 延长功能被禁止，DMAEN 位应该在 ADDRF 事件前被置位。只要 TXDATE 位被置位，数据将由 DMA 从预置的存储区装载进 I2C_DAT 寄存器。

设置 DMA 通道进行 I2C 发送，须执行以下步骤（x 是通道号）：

1. 在 DMA_PADRx 寄存器中设置 I2C_DAT 寄存器地址。数据将在每个 TXDATE 事件后从存储器传送至这个地址。
2. 在 DMA_MADRx 寄存器中设置存储器地址。数据在每个 TXDATE 事件后从这个存储区传送至

I2C_DAT。

3. 在 DMA_TXNUMx 寄存器中设置所需传输的字节数。在每个 TXDATE 事件后，此值递减。
4. 利用 DMA_CHCFGx 寄存器中的 PL[0:1]位配置通道优先级。
5. 设置 DMA_CHCFGx 寄存器中的 DIR 位，并根据应用要求可以配置在整个传输完成一半或全部完成时发出中断请求。
6. 通过设置 DMA_CHCFGx 寄存器上的 CHEN 位激活通道。
7. 当 DMA 控制器中设置的数据传输数目已经完成时，DMA 控制器给 I2C 接口发送一个传输结束的 EOT/ EOT_1 信号。在中断允许的情况下，将产生一个 DMA 中断。

注：如果使用 DMA 进行发送时，不要设置 I2C_CTRL2 寄存器的 BUFINTEN 位。

23.3.5.2 DMA 接收

DMA 模式通过 I2C_CTRL2 寄存器中的 DMAEN 位使能。每次接收到数据字节时，将由 DMA 把 I2C_DAT 寄存器的数据传送到设置的存储区（参考 DMA 说明）。

设置 DMA 通道进行 I2C 接收，须执行以下步骤（x 是通道号）：

1. 在 DMA_CPARx 寄存器中设置 I2C_DAT 寄存器的地址。数据将在每次 RXDATNE 事件后从此地址传送到存储区。
2. 在 DMA_CMARx 寄存器中设置存储区地址。数据将在每次 RXDATNE 事件后从 I2C_DAT 寄存器传送到此存储区。
3. 在 DMA_CNDTRx 寄存器中设置所需的传输字节数。在每个 RXDATNE 事件后，此值递减。
4. 用 DMA_CHCFGx 寄存器中的 PL[0:1]配置通道优先级。
5. 清除 DMA_CHCFGx 寄存器中的 DIR 位，根据应用要求可以设置在数据传输完成一半或全部完成时发出中断请求。
6. 设置 DMA_CHCFGx 寄存器中的 CHEN 位激活该通道。
7. 当 DMA 控制器中设置的数据传输数目已经完成时，DMA 控制器给 I2C 接口发送一个传输结束的 EOT/ EOT_1 信号。在中断允许的情况下，将产生一个 DMA 中断。

注：如果使用 DMA 进行接收时，不要设置 I2C_CTRL2 寄存器的 BUFINTEN 位。

23.3.6 包错误校验（PEC）

I2C 模块中有一个 PEC 模块，它使用 CRC-8 计算器来执行 I2C 数据的报文校验，从而提高通信的可靠性。包错误校验（PEC）计算器使用的 CRC-8 多项式为 C(x) = x8 + x2 + x + 1

将 I2C_CTRL1 寄存器的 PECEN 位置 1 就可以使能 PEC 功能，PEC 使用 CRC-8 算法对包括地址和读/写位在内所有信息字节进行计算。在发送模式时，软件可以在最后一个 TXDATE 事件时设置 I2C_CTRL1 寄存器的 PEC 传输位，PEC 将在最后一个字节后被发送。在接收模式时，软件在最后一个 RXDATNE 事件之后设置 I2C_CTRL1 寄存器的 PEC 位，然后接收 PEC 字节，并将接收到的 PEC 字节与内部计算的 PEC 值进行比较。如果不等于内部计算的 PEC，接收器发送一个 NACK。如果是主机接收器模式，不管校对的结果如何，PEC 后都将发送 NACK。需要注意，PEC 位必须在接收当前字节的 ACK 脉冲之前设

置。

在 I2C_STS1 寄存器中可获得 PECERR 错误标记/中断。

如果 DMA 和 PEC 计算器都被激活，I2C 将自动发送或者检查 PEC 值。在发送模式时，当 I2C 接口从 DMA 控制器处接收到 EOT 信号时，它在最后一个字节后自动发送 PEC。在接收模式时，当 I2C 接口从 DMA 处接收到一个 EOT_1 信号时，它将自动把下一个字节作为 PEC，并且和内部计算的 PEC 进行比较。在接收到 PEC 后产生一个 DMA 请求。

为了允许中间 PEC 传输，在 I2C_CTRL2 寄存器中有一个控制位（DMALAST 位）用于判别是否真是最后一个 DMA 传输。如果确实是最后一个主接收器的 DMA 请求，在接收到最后一个字节后自动发送 NACK。

当仲裁丢失的时候，PEC 计算失效。

23.4 I2C 中断请求

下表列出了所有的 I2C 中断请求

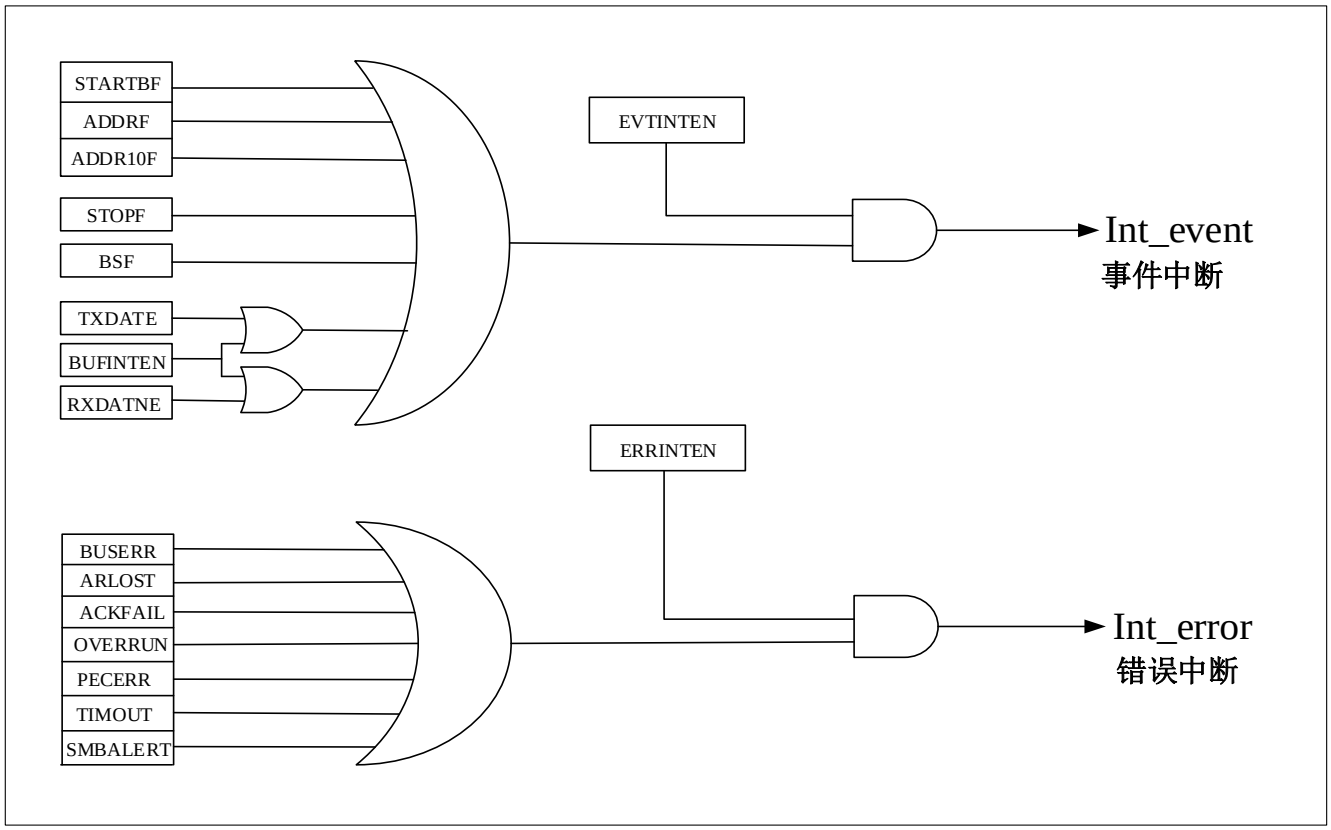
表 23-2 I2C 中断请求表：

中断事件	事件标志	开启控制位
起始位已发送（主）	STARTBF	EVTINTEN
地址已发送（主） 或 地址匹配（从）	ADDRF	
10 位头段已发送（主）	ADDR10F	
已收到停止（从）	STOPF	
数据字节传输完成	BSF	
接收缓冲区非空	RXDATNE	EVTINTEN 和 BUFINTEN
发送缓冲区空	TXDATE	
总线错误	BUSERR	ERRINTEN
仲裁丢失（主）	ARLOST	
响应失败	ACKFAIL	
过载/欠载	OVERRUN	
PEC 错误	PECERR	
超时/Tlow 错误	TIMOUT	
SMBus 提醒	SMBALERT	

注：1、STARTBF、DDRF、ADDR10F、STOPF、BSF、RXDATNE 和 TXDATE 通过逻辑或汇到同一个中断通道中。

2、BUSERR、ARLOST、ACKFAIL、OVERRUN、PECERR、TIMOUT 和 SMBALERT 通过逻辑或汇到同一个中断通道中。

图 23-7 I2C 中断映射图



23.5 I2C 调试模式

当微控制器进入调试模式（N308 核心处于停止状态）时，根据 DBG 模块中的 DBG_I2Cx_SMBUS_TIMEOUT 配置位，SMBUS 超时控制或者继续正常工作或者可以停止。详见第 31.3.2 节。

23.6 I2C 寄存器描述

关于在寄存器描述里面所用到的缩写，详见第 1.1 节。

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

23.6.1 I2C 寄存器地址映象

表 23-3 I2C 寄存器地址映象和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	I2C_CTRL1	Reserved																SWRESET	Reserved	SMBALERT	PEC	ACKPOS	ACKEN	STOPGEN	STARTGEN	NOEXTEND	GCEN	PECEN	ARPEN	SMBTYPE	Reserved	SMBMODE	EN
	Reset Value																	0		0	0	0	0	0	0	0	0	0	0	0		0	0
004h	I2C_CTRL2	Reserved																			DMALAST	DMAEN	BUFINTEN	EVTINTEN	ERRINTEN	Reserved		CLKFREQ[5:0]						

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset Value																				0	0	0	0	0			0	0	0	0	0	0
008h	I2C_OADDR1	Reserved																ADDRMODE	Reserved	Reserved				ADDR [9:8]		ADDR[7:1]							ADDR0
	Reset Value																	0	1					0	0	0	0	0	0	0	0	0	0
00Ch	I2C_OADDR2	Reserved																								ADDR2[7:1]							DUALEN
	Reset Value																									0	0	0	0	0	0	0	0
010h	I2C_DAT	Reserved																								DATA[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0
014h	I2C_STS1	Reserved																SMBALERT	TIMOUT	Reserved	PECERR	OVERRUN	ACKFAIL	ARLOST	BUSERR	TXDATE	RXDATNE	Reserved	STOPF	ADDR1OF	BYTEF	ADDRF	STARTBF
	Reset Value																	0	0		0	0	0	0	0	0	0		0	0	0	0	0
018h	I2C_STS2	Reserved																PECVAL[7:0]								DUALFLAG	SMBHADDR	SMBDADDR	GCALLADDR	Reserved	TRF	BUSY	MSMODE
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
01Ch	I2C_CLKCTRL	Reserved																FSMODE	DUTY	Reserved	CLKCTRL[11:0]												
	Reset Value																	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
020h	I2C_TMRISE	Reserved																										TMRISE[5:0]					
	Reset Value																											0	0	0	0	0	0

23.6.2 控制寄存器 1（I2C_CTRL1）

地址偏移：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SW RESET	Reserved	SMB ALERT	PEC	ACK POS	ACKEN	STOP GEN	START GEN	NO EXTEND	GCEN	PECEN	ARPEN	SMB TYPE	Reserved	SMB MODE	EN
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

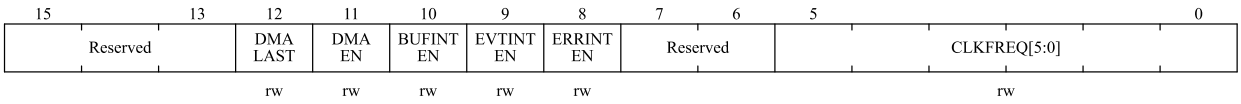
位域	名称	描述
15	SWRESET	软件复位（Software reset） 当被置位时，I2C 处于复位状态。在复位该位前确信 I2C 的引脚被释放，总线是空的。 0：I2C 模块不处于复位状态； 1：I2C 模块处于复位状态。 *注：该位可以用于 BUSY 位为'1'，在总线上又没有检测到停止条件时。*
14	Reserved	保留，硬件强制为 0。
13	SMBALERT	SMBus 提醒（SMBus alert） 软件可以设置或清除该位；当 EN=0 时，由硬件清除。 0：释放 SMBAlert 引脚使其变高。提醒响应地址头紧跟在 NACK 信号后面； 1：驱动 SMBAlert 引脚使其变低。提醒响应地址头紧跟在 ACK 信号后面。
12	PEC	数据包出错检测（Packet error checking） 软件可以设置或清除该位；当传送 PEC 后，或起始或停止条件时，或当 PE=0 时硬件将其清除。 0：无 PEC 传输；

位域	名称	描述
		1：PEC 传输（在发送或接收模式）。 *注：仲裁丢失时，PEC 的计算失效。*
11	ACKPOS	应答/PEC 位置（用于数据接收）（Acknowledge/PEC Position（for data reception）） 软件可以设置或清除该位，或当 EN =0 时，由硬件清除。 0：ACK 位控制当前移位寄存器内正在接收的字节的（N）ACK。PEC 位表明当前移位寄存器内的字节是 PEC; 1：ACK 位控制在移位寄存器里接收的下一个字节的（N）ACK。PEC 位表明在移位寄存器里接收的下一个字节是 PEC。 *注：ACKPOS 位只能用在 2 字节的接收配置中，必须在接收数据之前配置。* 为了 NACK 第 2 个字节，必须在清除 ADDRF 为之后清除 ACK 位。 为了检测第 2 个字节的 PEC，必须在配置了 ACKPOS 位之后，拉伸 ADDR 事件时设置 PEC 位。
10	ACKEN	应答使能（Acknowledge enable） 软件可以设置或清除该位，或当 EN =0 时，由硬件清除。 0：无应答返回; 1：在接收到一个字节后返回一个应答（匹配的地址或数据）。
9	STOPGEN	停止条件产生（Stop generation） 软件可以设置或清除该位；或当检测到停止条件时，由硬件清除；当检测到超时错误时，硬件将其置位。 在主模式下: 0：无停止条件产生; 1：在当前字节传输或在当前起始条件发出后产生停止条件。 在从模式下: 0：无停止条件产生; 1：在当前字节传输或释放 SCL 和 SDA 线。 *注：当设置了 STOPGEN、STARTGEN 或 PEC 位，在硬件清除这个位之前，软件不要执行任何对 I2C_CTRL1 的写操作；否则有可能会第 2 次设置 STOPGEN、STARTGEN 或 PEC 位。*
8	STARTGEN	起始条件产生（Start generation） 软件可以设置或清除该位，或当起始条件发出后或 EN=0 时，由硬件清除。 在主模式下: 0：无起始条件产生; 1：重复产生起始条件。 在从模式下: 0：无起始条件产生; 1：当总线空闲时，产生起始条件。
7	NOEXTEND	禁止时钟延长（从模式）（Clock stretching disable （Slave mode）） 该位用于当 ADDRF 或 BSF 标志被置位，在从模式下禁止时钟延长，直到它被软件复位。 0：允许时钟延长; 1：禁止时钟延长。
6	GCEN	广播呼叫使能（General call enable） 0：禁止广播呼叫。以非应答响应地址 00h;

位域	名称	描述
		1：允许广播呼叫．以应答响应地址 00h。
5	PECEN	PEC 使能（PEC enable） 0：禁止 PEC 计算； 1：开启 PEC 计算。
4	ARPEN	ARP 使能（ARP enable） 0：禁止 ARP； 1：使能 ARP。 如果 SMBTYPE=0，使用 SMBus 设备的默认地址。 如果 SMBTYPE=1，使用 SMBus 的主地址。
3	SMBTYPE	SMBus 类型（SMBus type） 0：SMBus 设备； 1：SMBus 主机。
2	Reserved	保留，硬件强制为 0。
1	SMBMODE	SMBus 模式（SMBus mode） 0：I2C 模式； 1：SMBus 模式。
0	EN	I2C 模块使能（Peripheral enable） 0：禁用 I2C 模块； 1：启用 I2C 模块：根据 SMBMODE 位的设置，相应的 I/O 口需配置为复用功能。 *注：如果清除该位时通讯正在进行，在当前通讯结束后，I2C 模块被禁用并返回空闲状态。* 由于在通讯结束后发生 EN = 0，所有的位被清除。 在主模式下，通讯结束之前，绝不能清除该位。

23.6.3 控制寄存器 2（I2C_CTRL2）

地址偏移：0x04
复位值：0x0000

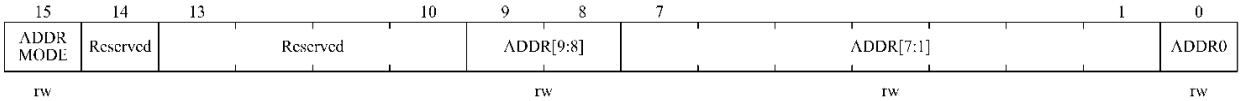


位域	名称	描述
15:13	Reserved	保留，硬件强制为 0。
12	DMALAST	DMA 最后一次传输（DMA last transfer） 0：下一次 DMA 的 EOT 不是最后的传输； 1：下一次 DMA 的 EOT 是最后的传输。 *注：该位在主接收模式使用，使得在最后一次接收数据时可以产生一个 NACK。*
11	DMAEN	DMA 请求使能（DMA requests enable） 0：禁止 DMA 请求； 1：当 TXDATE =1 或 RXDATNE =1 时，允许 DMA 请求。
10	BUFINTEN	缓冲器中断使能（Buffer interrupt enable） 0：当 TXDATE=1 或 RXDATNE=1 时，不产生任何中断；

位域	名称	描述
		1：当 TXDATE=1 或 RXDATNE=1 时，产生事件中断（不管 DMAEN 是何种状态）。
9	EVTINTEN	事件中断使能（Event interrupt enable） 0：禁止事件中断； 1：允许事件中断。 在下列条件下，将产生该中断： STARTBF = 1（主模式）； ADDR F= 1（主/从模式）； ADD10F= 1 （主模式）； STOPF = 1 （从模式）； BSF = 1，但是没有 TXDATE 或 RXDATNE 事件； 如果 BUFINTEN = 1， TXDATE 事件为 1； 如果 BUFINTEN = 1， RXDATNE 事件为 1。
8	ERRINTEN	出错中断使能（Error interrupt enable） 0：禁止出错中断； 1：允许出错中断。 在下列条件下，将产生该中断： BUSERR = 1； ARLOST = 1； ACKFAIL = 1； OVERRUN = 1； PECERR = 1； TIMOUT = 1； SMBALERT = 1。
7:6	Reserved	保留，硬件强制为 0。
5:0	CLKFREQ[5:0]	I2C 模块时钟频率（Peripheral clock frequency） 必须设置正确的输入时钟频率以产生正确的时序，允许的范围在 2 ~ 36MHz 之间： 000000：禁用 000001：禁用 000010：2MHz … 100100：36MHz 大于 100100：禁用。

23.6.4 自身地址寄存器 1（I2C_OADDR1）

复位地址偏移：0x08
复位值：0x0000

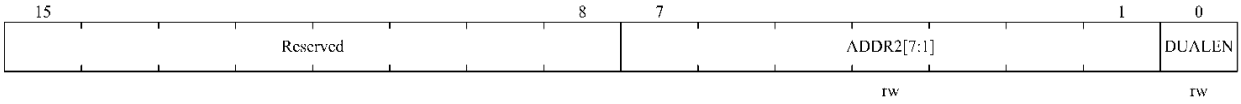


位域	名称	描述
15	ADDRMODE	寻址模式（从模式）（Addressing mode（slave mode））

位域	名称	描述
		0：7 位从地址（不响应 10 位地址）； 1：10 位从地址（不响应 7 位地址）。
14	Reserved	必须始终由软件保持为'1'。
13:10	Reserved	保留，硬件强制为 0。
9:8	ADDR[9:8]	接口地址（Interface address） 7 位地址模式时不用关心。 10 位地址模式时为地址的 9~8 位。
7:1	ADDR[7:1]	接口地址（Interface address） 地址的 7~1 位。
0	ADDR0	接口地址（Interface address） 7 位地址模式时不用关心。 10 位地址模式时为地址第 0 位。

23.6.5 自身地址寄存器 2（I2C_OADDR2）

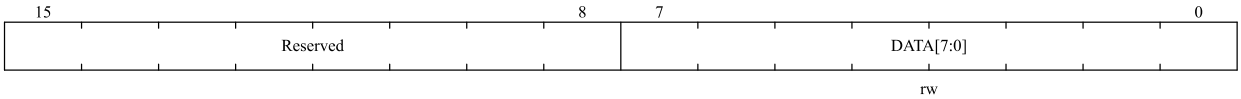
地址偏移：0x0C
复位值：0x0000



位域	名称	描述
15:8	Reserved	保留，硬件强制为 0。
7:1	ADDR2[7:1]	接口地址（Interface address） 在双地址模式下地址的 7~1 位。
0	DUALEN	双地址模式使能位（Dual addressing mode enable） 0：在 7 位地址模式下，只有 OADDR1 被识别； 1：在 7 位地址模式下，OADDR1 和 OADDR2 都被识别。

23.6.6 数据寄存器（I2C_DAT）

地址偏移：0x10
复位值：0x0000



位域	名称	描述
15:8	Reserved	保留，硬件强制为 0。
7:0	DATA[7:0]	8 位数据寄存器（8-bit data register） 用于存放接收到的数据或放置用于发送到总线的数据 发送器模式：当写一个字节至 DAT 寄存器时，自动启动数据传输。一旦传输开始（TXDATE=1），如果能及时把下一个需传输的数据写入 DAT 寄存器，I2C 模块将保持连续的数据流。

位域	名称	描述
		接收器模式：接收到的字节被拷贝到 DAT 寄存器（RXDATNE=1）。在接收到下一个字节（RXDATNE=1）之前读出数据寄存器，即可实现连续的数据传送。 *注：在从模式下，地址不会被拷贝进数据寄存器 DAT;* *注：硬件不管理写冲突（如果 TXDATE =0，仍能写入数据寄存器）;* *注：如果在处理 ACK 脉冲时发生 ARLOS 事件，接收到的字节不会被拷贝到数据寄存器里，因此不能读到它。*

23.6.7 状态寄存器 1（I2C_STS1）

地址偏移：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMB ALERT	TIM OUT	Reserved	PEC ERR	OVER RUN	ACK FAIL	AR LOST	BUS ERR	TXDATE	RXDAT NE	Reserved	STOPF	ADDR10F	BSF	ADDRF	START BF
rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	r	r		r	r	r	r	r

位域	名称	描述
15	SMBALERT	SMBus 提醒（SMBus alert） 在 SMBus 主机模式下： 0：无 SMBus 提醒； 1：在引脚上产生 SMBAlert 提醒事件。 在 SMBus 从机模式下： 0：没有 SMBAlert 响应地址头序列； 1：收到 SMBAlert 响应地址头序列至 SMBAlert 变低。 该位由软件写'0'清除，或在 EN=0 时由硬件清除。
14	TIMOUT	超时或 Tlow 错误（Timeout or Tlow error） 0：无超时错误； 1：SCL 处于低已达到 25ms（超时）；或者主机低电平累积时钟扩展时间超过 10ms（Tlow:mext）；或从设备低电平累积时钟扩展时间超过 25ms（Tlow:sext）。 当在从模式下设置该位：从设备复位通讯，硬件释放总线。 当在主模式下设置该位：硬件发出停止条件。 该位由软件写'0'清除，或在 EN=0 时由硬件清除。
13	Reserved	保留，硬件强制为 0。
12	PECERR	在接收时发生 PEC 错误（PEC Error in reception） 0：无 PEC 错误：接收到 PEC 后接收器返回 ACK（如果 ACKEN=1）； 1：有 PEC 错误：接收到 PEC 后接收器返回 NACK（不管 ACKEN 是什么值）。 该位由软件写'0'清除，或在 EN=0 时由硬件清除。
11	OVERRUN	过载/欠载（Overrun/Underrun） 0：无过载/欠载； 1：出现过载/欠载。 当 NOEXTEND =1 时，在从模式下该位被硬件置位，同时： 在接收模式中当收到一个新的字节时（包括 ACK 应答脉冲），数据寄存器里的内容还未被读出，则新接收的字节将丢失。

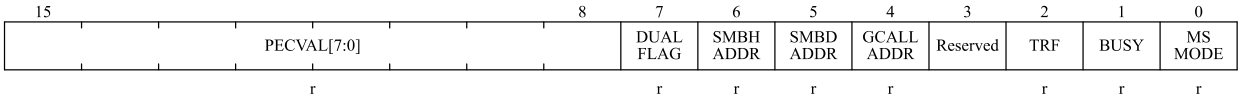
位域	名称	描述
		在发送模式中当要发送一个新的字节时，却没有新的数据写入数据寄存器，同样的字节将被发送两次。 该位由软件写‘0’清除，或在 EN=0 时由硬件清除。。 *注：如果数据寄存器的写操作发生时间非常接近 SCL 的上升沿，发送的数据是不确定的，并发生保持时间错误。*
10	ACKFAIL	应答失败（Acknowledge failure） 0：没有应答失败； 1：应答失败。 当没有返回应答时，硬件将置该位为‘1’。 该位由软件写‘0’清除，或在 EN=0 时由硬件清除。
9	ARLOST	仲裁丢失（主模式）（Arbitration lost（master mode）） 0：没有检测到仲裁丢失； 1：检测到仲裁丢失。 当接口失去对总线的控制给另一个主机时，硬件将置该位为‘1’。 该位由软件写‘0’清除，或在 EN=0 时由硬件清除。 在 ARLOST 事件之后，I2C 接口自动切换回从模式（MSMODE=0）。 *注：在 SMBUS 模式下，在从模式下对数据的仲裁仅仅发生在数据阶段，或应答传输区间（不包括地址的应答）。*
8	BUSERR	总线出错（Bus error） 0：无起始或停止条件出错； 1：起始或停止条件出错。 当接口检测到错误的起始或停止条件，硬件将该位置‘1’。 该位由软件写‘0’清除，或在 EN=0 时由硬件清除。
7	TXDATE	数据寄存器为空（发送时）（Data register empty（transmitters）） 0：数据寄存器非空； 1：数据寄存器空。 在发送数据时，数据寄存器为空时该位被置‘1’，在发送地址阶段不设置该位。 软件写数据到 DAT 寄存器可清除该位；或在发生一个起始或停止条件后，或当 EN=0 时由硬件自动清除。 如果收到一个 NACK，或下一个要发送的字节是 PEC（PEC=1），该位不被置位。 *注：在写入第 1 个要发送的数据后，或设置了 BSF 时写入数据，都不能清除 TXDATE 位，这是因为数据寄存器仍然为空。*
6	RXDATNE	数据寄存器非空（接收时）（Data register not empty（receivers）） 0：数据寄存器为空； 1：数据寄存器非空。 在接收时，当数据寄存器不为空，该位被置‘1’。在接收地址阶段，该位不被置位。 软件对数据寄存器的读写操作清除该位，或当 EN=0 时由硬件清除。 在发生 ARLOST 事件时，RXDATNE 不被置位。 *注：当设置了 BSF 时，读取数据不能清除 RXDATNE 位，因为数据寄存器仍然为满。*
5	Reserved	保留，硬件强制为 0。
4	STOPF	停止条件检测位（从模式）（Stop detection（slave mode）） 0：没有检测到停止条件；

位域	名称	描述
		1：检测到停止条件。 在一个应答之后（如果 ACK=1），当从设备在总线上检测到停止条件时，硬件将该位置‘1’。 软件读取 STS1 寄存器后，对 CTRL1 寄存器的写操作将清除该位，或当 EN=0 时，硬件清除该位。 *注：在收到 NACK 后，STOPF 位不被置位。*
3	ADDR10F	10 位头序列已发送（主模式）（10-bit header sent（Master mode）） 0：没有 ADD10F 事件发生； 1：主设备已经将第一个地址字节发送出去。 在 10 位地址模式下，当主设备已经将第一个字节发送出去时，硬件将该位置‘1’。 软件读取 STS1 寄存器后，对 CTRL1 寄存器的写操作将清除该位，或当 EN=0 时，硬件清除该位。 *注：收到一个 NACK 后，ADD10F 位不被置位。*
2	BSF	字节发送结束（Byte transfer finished） 0：字节发送未完成； 1：字节发送结束。 当 NOEXTEND =0 时，在下列情况下硬件将该位置‘1’： 在接收时，当收到一个新字节（包括 ACK 脉冲）且数据寄存器还未被读取（RXDATNE=1）。 在发送时，当一个新数据将被发送且数据寄存器还未被写入新的数据（TXDATE=1）。 在软件读取 STS1 寄存器后，对数据寄存器的读或写操作将清除该位；或在传输中发送一个起始或停止条件后，或当 EN=0 时，由硬件清除该位。 *注：在收到一个 NACK 后， BSF 位不会被置位。* *如果下一个要传输的字节是 PEC（I2C_STS2 寄存器中 TRF 为‘1’，同时 I2C_CTRL1 寄存器中 PEC 为‘1’），BSF 位不会被置位。*
1	ADDRF	地址已被发送（主模式）/地址匹配（从模式）（Address sent（master mode）/matched（slave mode）） 在软件读取 STS1 寄存器后，对 STS2 寄存器的读操作将清除该位，或当 EN=0 时，由硬件清除该位。 地址匹配（从模式） 0：地址不匹配或没有收到地址； 1：收到的地址匹配。 当收到的从地址与 OADDR 寄存器中的内容相匹配、或发生广播呼叫、或 SMBus 设备默认地址或 SMBus 主机识别出 SMBus 提醒时，硬件就将该位置‘1’（当对应的设置被使能时）。 地址已被发送（主模式） 0：地址发送没有结束； 1：地址发送结束。 10 位地址模式时，当收到地址的第二个字节的 ACK 后该位被置‘1’。 7 位地址模式时，当收到地址的 ACK 后该位被置‘1’。 *注：在收到 NACK 后，ADDRF 位不会被置位。*
0	STARTBF	起始位（主模式）（Start bit（Master mode））

位域	名称	描述
		0：未发送起始条件； 1：起始条件已发送。 当发送出起始条件时该位被置‘1’。 软件读取 STS1 寄存器后，写数据寄存器的操作将清除该位，或当 EN=0 时，硬件清除该位。

23.6.8 状态寄存器 2（I2C_STS2）

地址偏移：0x18
复位值：0x0000



位域	名称	描述
15:8	PECVAL[7:0]	数据包出错检测（Packet error checking register） 当 PECEN =1 时， PECVAL [7:0]存放内部的 PEC 的值。
7	DUALFLAG	双标志（从模式）（Dual flag（Slave mode）） 0：接收到的地址与 OADDR1 内的内容相匹配； 1：接收到的地址与 OADDR2 内的内容相匹配。 在产生一个停止条件或一个重复的起始条件时，或 EN=0 时，硬件将该位清除。
6	SMBHADDR	SMBus 主机头系列（从模式）（SMBus host header（Slave mode）） 0：未收到 SMBus 主机的地址； 1：当 SMBTYPE=1 且 ARPEN=1 时，收到 SMBus 主机地址。 – 在产生一个停止条件或一个重复的起始条件时，或 PE=0 时，硬件将该位清除。
5	SMBDADDR	SMBus 设备默认地址（从模式）（SMBus device default address（Slave mode）） 0：未收到 SMBus 设备的默认地址； 1：当 ARPEN=1 时，收到 SMBus 设备的默认地址。 在产生一个停止条件或一个重复的起始条件时，或 EN=0 时，硬件将该位清除。
4	GCALLADDR	广播呼叫地址（从模式）（General call address（Slave mode）） 0：未收到广播呼叫地址； 1：当 GCEN=1 时，收到广播呼叫的地址。 在产生一个停止条件或一个重复的起始条件时，或 EN=0 时，硬件将该位清除
3	Reserved	保留，硬件强制为 0。
2	TRF	发送/接收（Transmitter/receiver） 0：接收到数据； 1：数据已发送； 在整个地址传输阶段的结尾，该位根据地址字节的 R/W 位来设定。 在检测到停止条件（STOPF=1）、重复的起始条件或总线仲裁丢失（ARLOST=1）后，或当 EN=0 时，硬件将其清除。
1	BUSY	总线忙（Bus busy） 0：在总线上无数据通讯；

位域	名称	描述
		1：在总线上正在进行数据通讯。 在检测到 SDA 或 SCl 为低电平时，硬件将该位置‘1’; 当检测到一个停止条件时，硬件将该位清除。 该位指示当前正在进行的总线通讯，当接口被禁用（EN=0）时该信息仍然被更新。
0	MSMODE	主从模式（Master/slave） 0：从模式; 1：主模式。 当接口处于主模式（STARTBF=1）时，硬件将该位置位; 当总线上检测到一个停止条件、仲裁丢失（ARLOST=1）时、或当 EN=0 时，硬件清除该位。

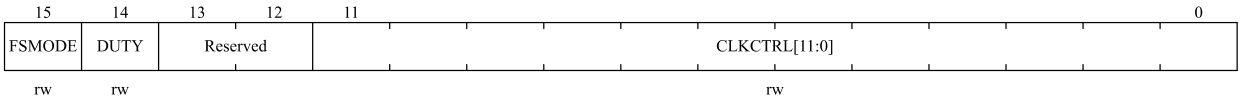
23.6.9 时钟控制寄存器（I2C_CLKCTRL）

地址偏移：0x1C

复位值：0x0000

注：1、要求 FPCLK1 应当是 10 MHz 的整数倍，这样可以正确地产生 400KHz 的快速时钟。

2、CLKCTRL 寄存器只有在关闭 I2C 时（EN=0）才能设置

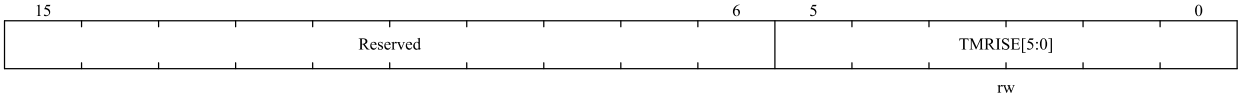


位域	名称	描述
15	FSMODE	I2C 主模式选项（I2C master mode selection） 0：标准模式的 I2C; 1：快速模式的 I2C。
14	DUTY	快速模式时的占空比（Fast mode duty cycle） 0：快速模式下：T_{low}/T_{high} = 2; 1：快速模式下：T_{low}/T_{high} = 16/9（见 CLKCTRL）。
13:12	Reserved	保留，硬件强制为 0。
11:0	CLKCTRL[11:0]	快速/标准模式下的时钟控制分频系数（主模式）（Clock control register in Fast/Standard mode（Master mode）） 该分频系数用于设置主模式下的 SCL 时钟。 在 I2C 标准模式或 SMBus 模式下： T_{high} = CLKCTRL ×T_{PCLK1} T_{low} = CLKCTRL ×T_{PCLK1} 在 I2C 快速模式下： 如果 DUTY = 0： T_{high} = CLKCTRL×T_{PCLK1} T_{low} = 2 ×CLKCTRL×T_{PCLK1} 如果 DUTY = 1：（速度达到 400KHz） T_{high} = 9 ×CLKCTRL×T_{PCLK1} T_{low} = 16 ×CLKCTRL×T_{PCLK1} 例如：在标准模式下，产生 100KHz 的 SCL 的频率：

位域	名称	描述
		如果 CLKFREQ = 08， T_{PCLK1}= 125ns，则 CLKCTRL 必须写入 0x28（40×125ns = 5000 ns）。 *注：1、允许设定的最小值为 0x04，在快速 DUTY 模式下允许的最小值为 0x01；* *2、T_{high} =$t_{r(SCL)}$+$t_{w(SCLH)}$，详见数据手册中对这些参数的定义；* *3、T_{low} =$t_{f(SCL)}$+$t_{w(SCLL)}$，详见数据手册中对这些参数的定义；* *4、这些延时没有过滤器；* *5、只有在关闭 I2C 时（EN = 0）才能设置 CLKCTRL 寄存器；* *6、f_{CK}应当是 10MHz 的整数倍，这样可以正确产生 400KHz 的快速时钟。*

23.6.10 TMRISE 寄存器（I2C_TMRISE）

地址偏移：0x20
复位值：0x0002



位域	名称	描述
15:6	Reserved	保留，硬件强制为 0。
5:0	TMRISE[5:0]	在快速/标准模式下的最大上升时间（主模式）(Maximum rise time in Fast/Standard mode（Master mode)） 这些位必须设置为 I2C 总线规范里给出的最大的 SCL 上升时间，增长步幅为 1。 例如：标准模式中最大允许 SCL 上升时间为 1000ns。如果在 I2C_CTRL2 寄存器中 CLKFREQ [5:0]中的值等于 0x08 且 T_{PCLK1}=125ns，故 TMRISE[5:0]中必须写入 09h（1000ns/125 ns = 8+1）。 滤波器的值也可以加到 TMRISE[5:0]内。 如果结果不是一个整数，则将整数部分写入 TMRISE[5:0]以确保 t_{HIGH} 参数。 *注：只有当 I2C 被禁用（EN=0）时，才能设置 TMRISE[5:0]。*

24 通用同步异步接收器（USART）

24.1 USART 简介

通用同步异步收发器（USART）是一种全双工或半双工，同步或异步的一个串行数据交换接口。USART 提供了可编程的波特率发生器，能对系统时钟进行分频产生 USART 发送和接收所需的特定频率。

USART 支持标准的异步收发模式、红外编码规范、SIR、智能卡协议、LIN、同步单双工模式、多处理器通信和 Modem 流控操作 (CTS/RTS)，为了实现高速数据通信还使用多缓冲器配置的 DMA 方式。

24.2 USART 主要特性

- 全双工异步通信
- 单线半双工通信

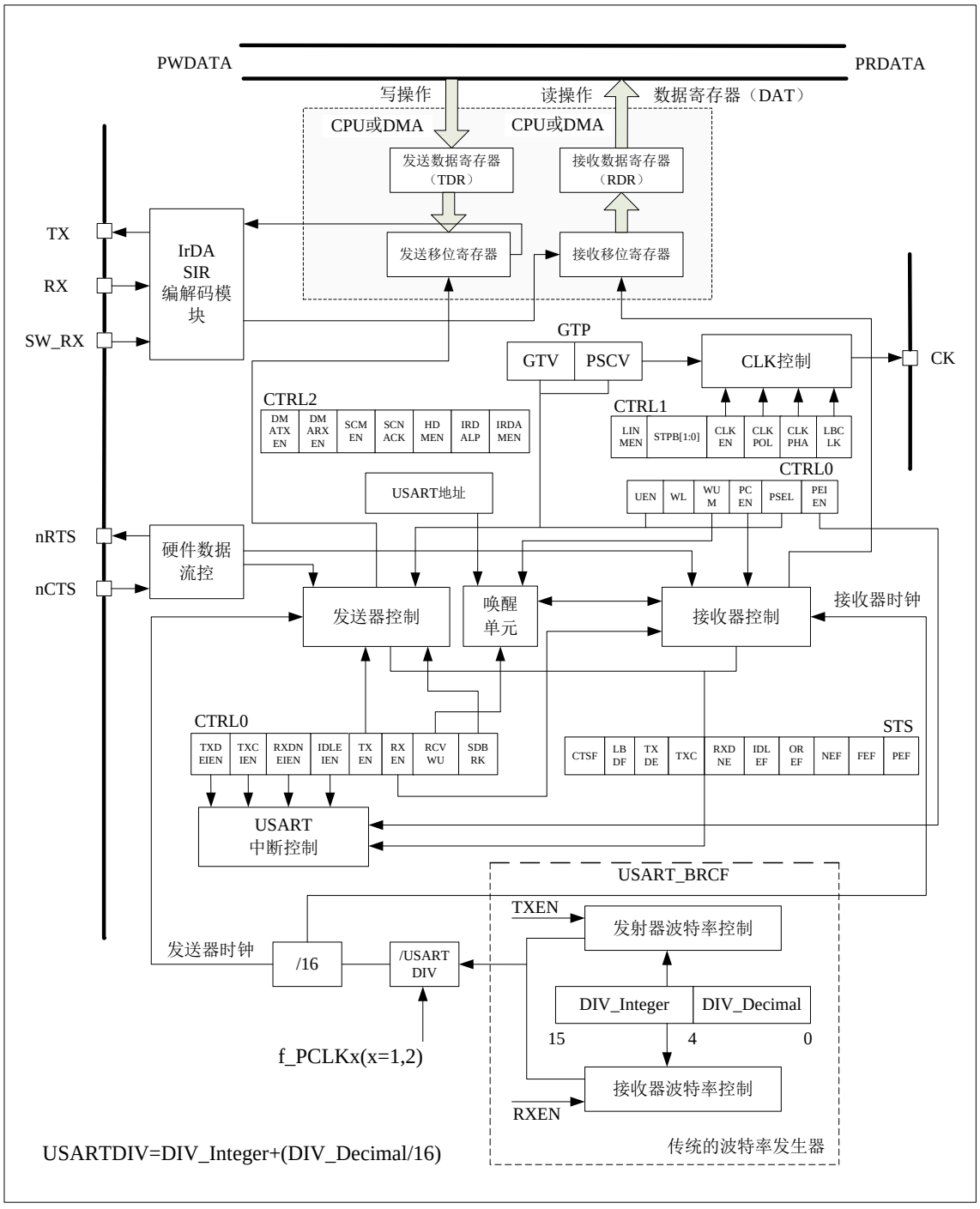
- NRZ 标准格式（Mark/Space）
- 可编程的波特率发生器
  - 发送和接收共用的可编程波特率，最高达 4.5MBits/s
- 可编程数据字长度（8 位或 9 位）
- 可配置的停止位-支持 1 或 2 个停止位
- LIN 主机有发送断开符的能力以及 LIN 从机有检测断开符的能力
  - 当 USART 硬件配置成 LIN 时，生成 13 位断开符；检测 10/11 位断开符
- 发送方为同步传输提供时钟
- IrDA SIR 编码器解码器
  - 在普通模式下支持 3/16 位的持续时间
- ISO7816-3 标准里定义的异步智能卡协议
  - 智能卡模式支持 0.5 或 1.5 个停止位
- 可配置的 DMA 多缓冲器通信
  - 利用 DMA 缓冲接收/发送数据
- 单独的发送器和接收器使能位
- 检测标志
  - 接收缓冲器满
  - 发送缓冲器空
  - 传输结束标志
- 校验控制
  - 发送校验位
  - 对接收数据进行校验
- 四个错误检测标志
  - 溢出错误
  - 噪音错误
  - 帧错误
  - 校验错误
- 10 个带标志的中断源。
  - CTS 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

- 多处理器通信 --如果地址不匹配，则进入静默模式
- 从静默模式中唤醒（通过空闲总线检测或地址检测）
- 两种唤醒接收器的方式：地址位（MSB，第 9 位），总线空闲

24.3 功能框图

图 24-1 USART 框图



24.4 USART 功能描述

接口通过三个引脚与其他设备连接在一起（见图 24-1）。任何 USART 双向通信至少需要两个脚: 接收数据输入（RX）和发送数据输出（TX）。

RX：串行数据输入端。为了区分数据和噪音,使用了过采样技术。

TX：串行数据输出端。当发送使能时，引脚默认高电平。TX 口也可以同时用于数据的发送和接收例如单线或智能卡模式，当发送未被使能，TX 口为原来的设置的 I/O 端口配置。

- 总线未发送或接收时应处于空闲状态
- 一个起始位
- 一个数据字（8 或 9 位），最低有效位在前
- 0.5，1，1.5，2 个停止位，表示数据帧的结束
- 使用分数波特率发生器——12 位整数和 4 位小数的表示方法
- 一个状态寄存器（USART_STS）
- 数据寄存器（USART_DAT）
- 一个波特率配置寄存器（USART_BRCF），12 位的整数和 4 位小数
- 一个智能卡模式下的保护时间寄存器（GTV）

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 24.7 节：USART 寄存器。

在同步模式中需要下列引脚:

- CK：此引脚输出用于同步传输的时钟，（在 Start 位和 Stop 位上没有时钟脉冲 ，USART_CR2 寄存器中的 LBCL 位控制最后一位数据对应时刻是否在 CK 引脚输出时钟脉冲）。数据可以在 RX 上同步被接收。这可以用来控制带有移位寄存器的外部设备（例如 LCD 驱动器）。CK 时钟相位和极性可以通过 USART_CR2 寄存器中 CPOL/CPHA 修改时钟极性和相位的。在智能卡模式下，CK 也可以提供时钟。

在硬件流控模式中需要下列引脚:

- CTS: 接收清零，若为低电平，表明在当前数据传输结束时可继续下一次的数据发送；若为高电平，在当前数据传输结束时阻断下一次的数据发送。
- RTS：发送请求，若为低电平，表明 USART 准备好接收数据。

24.4.1 USART 帧格式

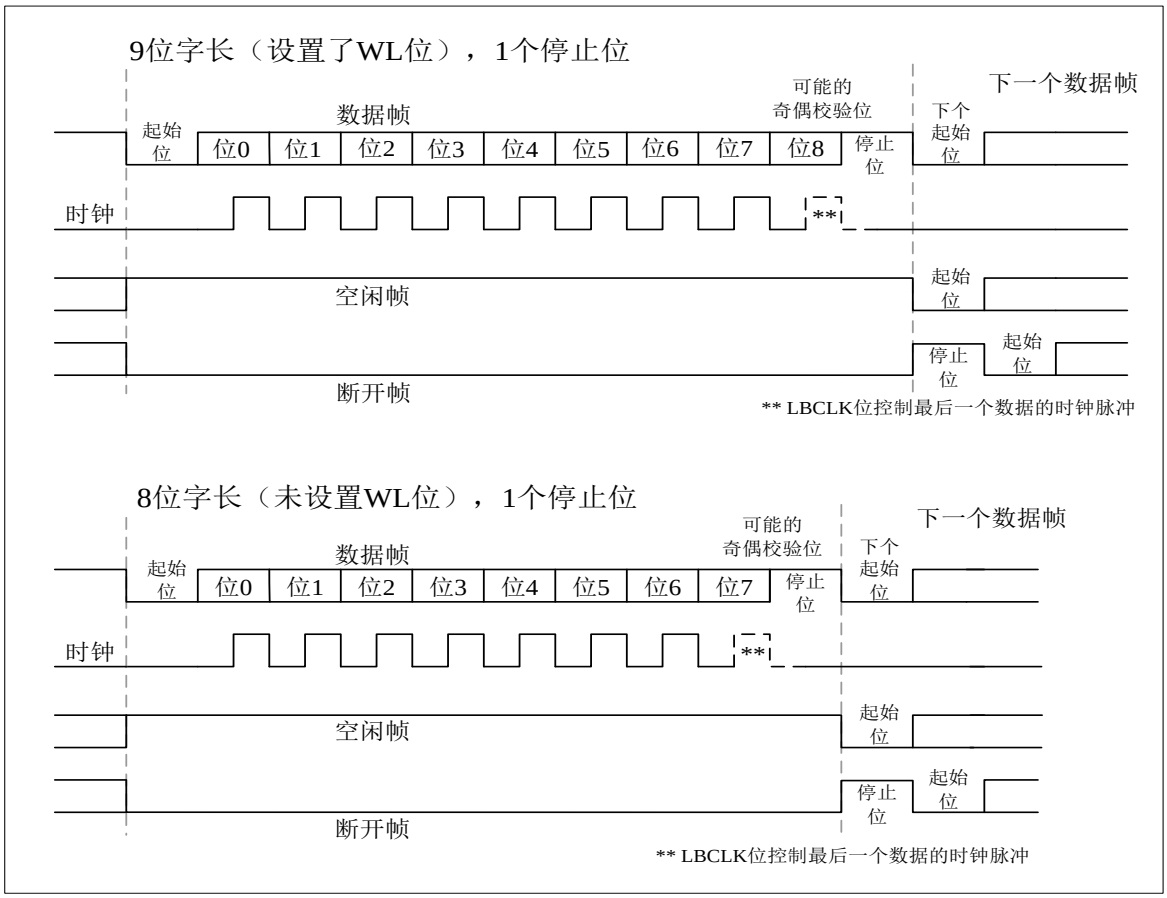
通过编程 USART_CTRL1 寄存器中的 WL 位字长选择成 8 或 9 位（见图 24-2）。在起始位期间，TX 脚处于低电平，在停止位期间处于高电平。

空闲帧：全部由'1'组成的一个完整的数据帧，也包含了数据的停止位。例如：如果 WL=0，则空闲帧由 10 个'1'组成；如果 WL=1，则空闲帧由 11 个'1'组成，

断开帧：被视为在一个帧周期内全部收到'0'（包括停止位期间，也是'0'）。在断开帧结束时，发送器再插入 1 或 2 个停止位（'1'）来应答起始位，WL=0 时，10 位低电平，后跟停止位；或者 WL=1 时，11 位低电平，后跟停止位，长度不能大于 10 位或者 11 位。

发送和接收均由一个共用的波特时钟发生器驱动，当发送器和接收器的使能位分别置位时，分别为其产生波特时钟。

图 24-2 字长设置



注意：在本章中，若未特殊说明，置位均表示某个寄存器被置为状态'1',复位或清零均表示某个寄存器被置为状态'0'；硬件或者程序均可能置位或者清零某个寄存器，请参考本章具体内容。

24. 4. 2 发送器

当发送使能位（TXEN）被置位时，且缓冲区内有数据，发送器根据 WL 位的状态发送 8 位或 9 位的数据字。发送移位寄存器中的数据在 TX 脚上输出，相应的时钟脉冲在 CK 脚上输出。

24. 4. 2. 1 字符发送

在 USART 发送数据时，TX 引脚首先移出数据的最低有效位。在字符发送模式里，USART_DAT 寄存器包含了一个内部总线和发送移位寄存器之间的缓冲器（见图 24-1）。

每个字符之前都有一个低电平的起始位；之后跟着的停止位，其数目可配置。

USART 支持 0.5、1、1.5 和 2 个停止位的配置。

注意：1.在数据传输期间不能复位 TXEN 位，否则将破坏 TX 脚上的数据，因为波特率计数器停止计数。正在传输的当前数据将丢失。2. 当 TXEN 位激活后，USART 将自动发送一个空闲帧。

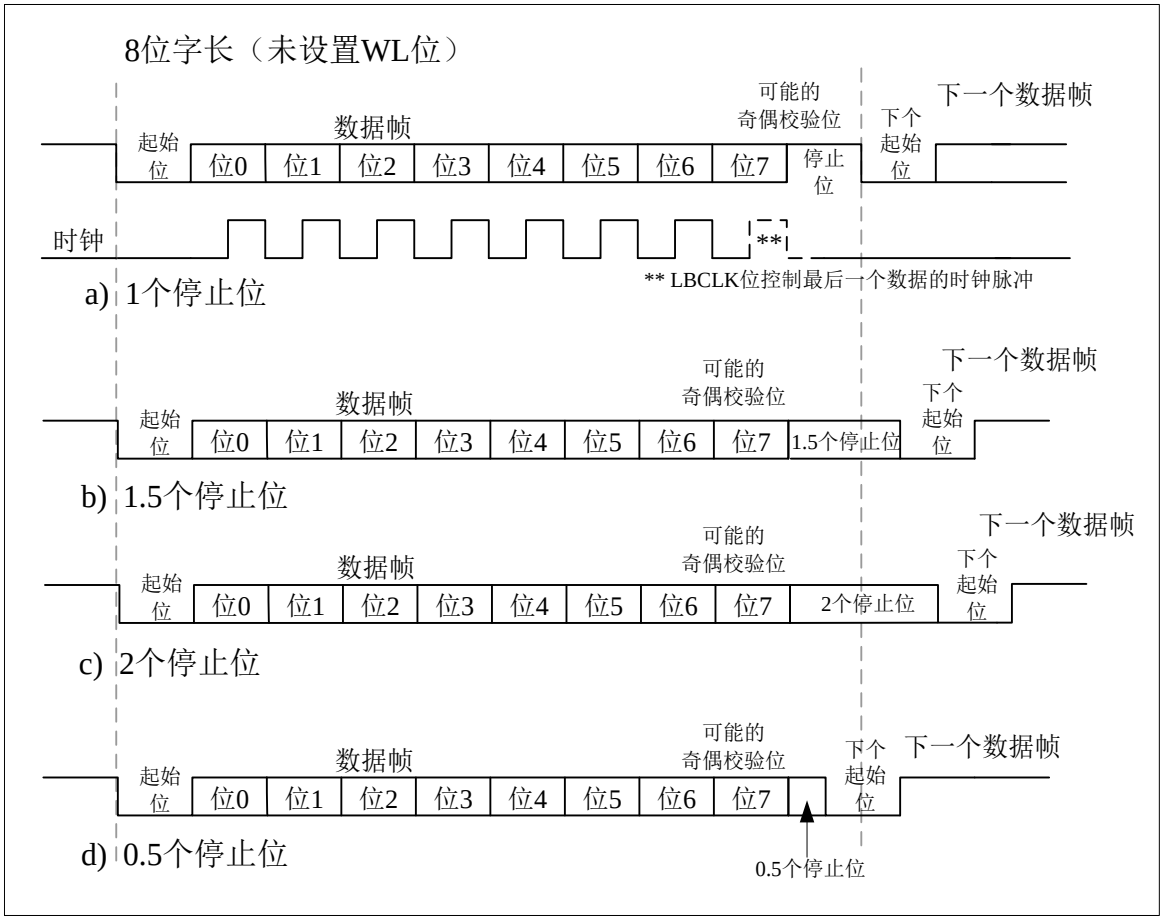
24. 4. 2. 2 可配置的停止位

在发送和接收中，每个字符的停止位的位数可以通过控制寄存器 1（USART_CTRL2）的 STPB[1:0]位进行配置。

表 24-1 停止位配置

STPB[1:0]	停止位长度（位）	功能描述
00	1	默认值
01	0.5	智能卡模式下接收
10	2	常规 USART 模式、单线模式以及调制解调器模式
11	1.5	智能卡模式下发送和接收

图 24-3 配置停止位



24. 4. 2. 3 单字节通信

清零 TXDE 位需要对数据寄存器的写操作来完成。

TXDE 位由硬件（也即 USART IP 核）来置位，此时：

- 数据已经从发送数据寄存器（TDR）移送到移位寄存器，发送数据寄存器被清空，数据开始发送。
- 下一个数据可以被写进 USART_DAT 寄存器。
- 如果 TXDEIEN 位被使能，此标志将产生一个中断。如果此时 USART 正在发送数据，对 USART_DAT 寄存器的写操作把数据存入发送数据寄存器，并在当前传输结束后把该数据复制到移位寄存器。

如果此时 USART 没有发送数据，处于空闲状态，对 USART_DAT 寄存器的写操作直接把数据放进移位寄存器，TXDE 位立即被置起，数据传输开始。

如果此时 USART 正在发送数据，对 USART_DAT 寄存器的写操作直接把数据放进发送数据寄存器（TDR），且在传输结束时，会把数据复制到移位寄存器。

当一帧发送完成时（停止位发送后）并且设置了 TXDE 位，TXC 位被置起，如果 USART_CTRL1 寄存器中的 TXCIEN 位被置起时，则会立刻产生中断。

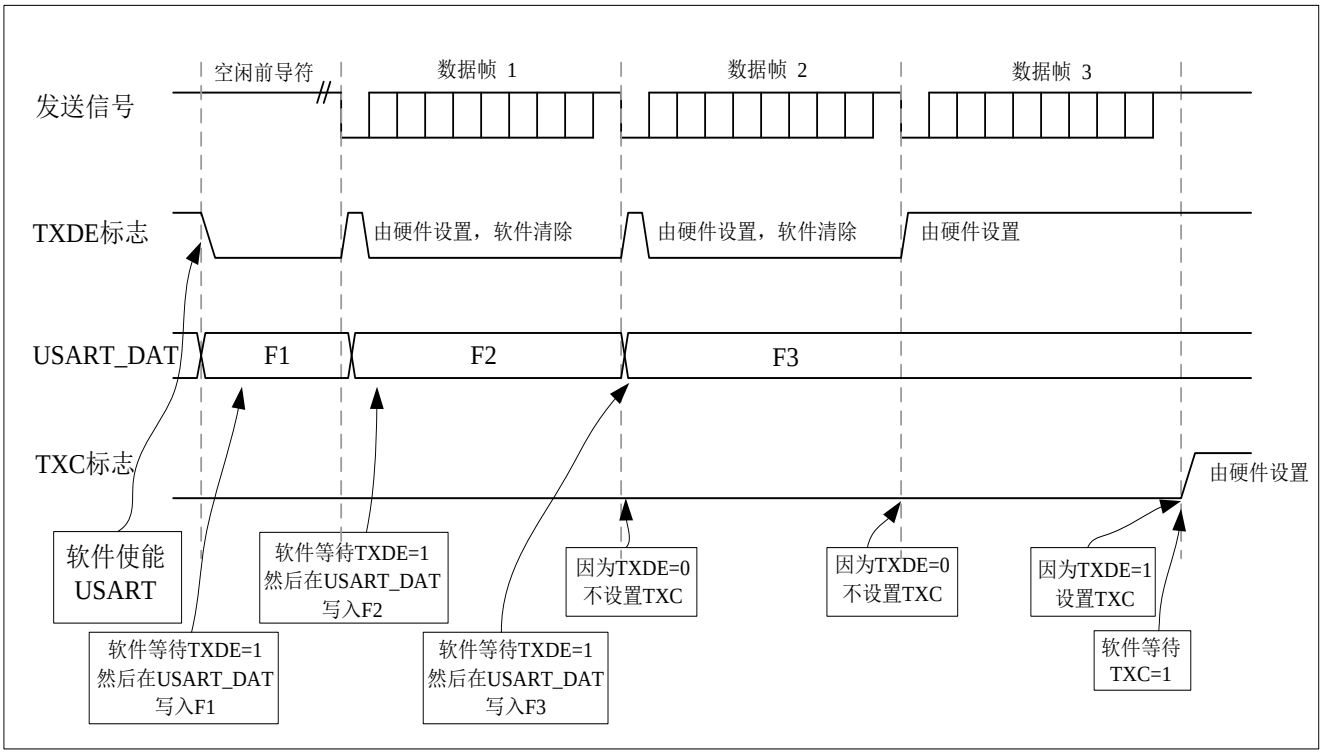
在 USART_DAT 寄存器中写入了最后一个数据字后，在关闭 USART 模块之前或设置微控制器进入低功耗模式（见图 24-4）之前，必须先等待 TXC=1。

可以使用软件操作清除 TXC 位:

1. 读一次 USART_STS 寄存器;
2. 写一次 USART_DAT 寄存器。

注意：TXC 位也可以通过软件对它写'0'来清除。此清零方式只推荐在 DMA 多缓冲器通信模式下使用。

图 24-4 发送时 TXC/TXDE 的变化情况



24. 4. 2. 4 断开帧

设置 SDBRK 位可发送一个断开帧如果设置 SDBRK=1，在完成当前数据发送后，将在 TX 线上发送一个断开帧。断开字符发送完成时（在断开帧的停止位时）SDBRK 被硬件复位。USART 在最后一个断开帧的结束处插入一逻辑'1'，以保证能识别下一帧的起始位，断开帧长度取决 WL 位（见图 24-2）。

注意：软件复位了 SDBRK 位之后，在开始发送断开帧之前，断开帧将不被发送。如果要发送两个连续的断开帧，SDBRK 位应该在前一个断开帧的停止位之后置起。

24. 4. 2. 5 空闲符号

置位 TXEN 将使得 USART 在第一个数据帧前先发送一空闲帧。

24. 4. 3 接收器

USART_CTRL1 的 WL 位决定接收 8 位或 9 位的数据字。

24. 4. 3. 1 起始位侦测

在 USART 中，如果辨认出一个特殊的采样序列，那么就认为侦测到一个起始位。

该序列为：1110X0X0X0000。

注意：如果该序列不完整，那么接收端将退出起始位侦测并回到空闲状态（不设置任何标志位）并等待下降沿。

1. 在第 3、5、7 位的采样，以及在第 8、9、10 位的采样都为'0'（也即 6 个'0'），则确认收到起始位，并且不会置位 NEF 噪声标志位。

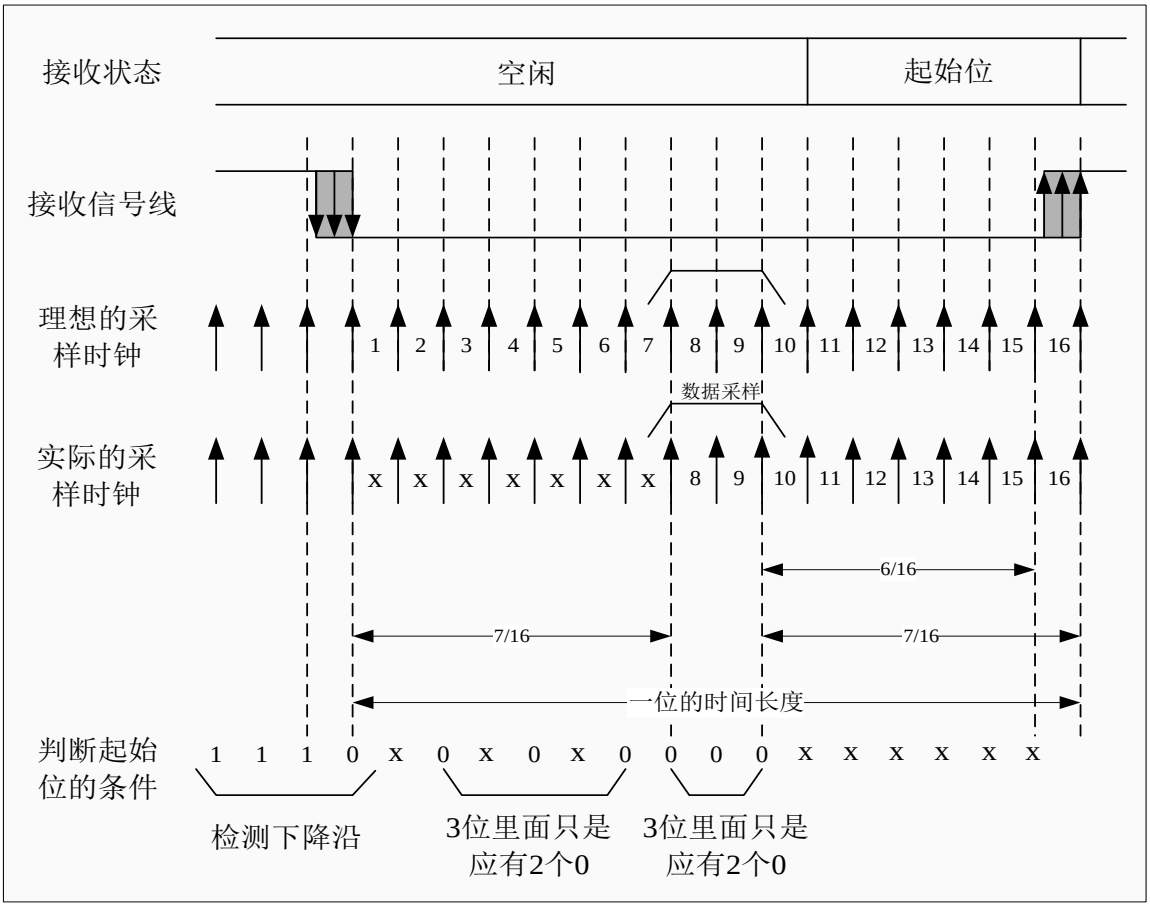
2. 第 3、5、7 位的采样有两个'0',与此同时，第 8、9、10 位的采样有两个'0'点，那么确认收到起始位，但是会置位 NEF 噪声标志位。

3. 第 3、5、7 位的采样有两个'0',与此同时，第 8、9、10 位的采样有三个'0'点，那么确认收到起始位，但是会置位 NEF 噪声标志位。

4. 第 3、5、7 位的采样有三个'0'，与此同时，第 8、9、10 位的采样有两个'0'点，那么确认收到起始位，但是会置位 NEF 噪声标志位。

5. 如果在第 3、5、7、8、9、10 位的采样值满足不了上面四种要求，则该 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。

图 24-5 起始位侦测



24. 4. 3. 2 字符接收

在 USART 接收期间，数据的最低有效位首先从 RX 脚移进。在此模式里，USART_DAT 寄存器包含的缓冲器位于内部 APB 总线和接收移位寄存器之间。

USART 接收器使能按以下步骤进行：

1. 在 USART_CTRL1 寄存器中置位 UEN 位，使能 USART；
2. 写 USART_CTRL1 寄存器的 WL 去设置字长；
3. 在 USART_CTRL2 寄存器中写 STPB[1:0]位来设置停止位的长度；
4. 如果选择了多级缓存通信方式，应该在 USART_CTRL3 寄存器中使能 DMA（DMARXEN 位）；
5. 在 USART_BRCF 寄存器中设置波特率；
6. 在 USART_CTRL1 中设置 RXEN 位。

当收到一个数据帧：

- RXDNE 位会置位，移位寄存器的内容被转移到 RDR（Receiver Data Register）。此时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）。
- 如果设置了 RXDNEIEN 位，则产生中断。
- 接收过程中会检测帧错误、噪音或溢出错误，这样错误标志将被置起。

- 在多缓冲器通信模式，RXDNE 标志位在每个字节接收后置，并由 DMA 对数据寄存器的读操作而清零。
- 在单缓冲器模式里，软件可以通过读 USART_DAT 寄存器完成对 RXDNE 位清除或者写 0 也可以清除 RXDNE 位。RXDNE 位必须在下一帧数据接收结束前被清零，以避免溢出错误。

  在接收过程中，RXNE 必须使能，否则当前数据帧会丢失。

24. 4. 3. 3 断开帧

当接收到一个断开帧时，USART 帧错误标志将被置起。

备注：在 LIN 模式下，当接收到一个断开帧时，将按断开帧处理，LINBDF 将会置起。

24. 4. 3. 4 空闲符号

当一空闲帧被检测到时，其处理步骤和接收到普通数据帧一样，但如果 IDLEIEN 位被设置将产生一个中断。

24. 4. 3. 5 溢出错误

如果接收的数据未及时被读走，导致 RXDNE 没有及时被复位，又接收到一个字符，则发生溢出错误。数据只有当 RXDNE 位被清零后才能从移位寄存器转移到 RDR 寄存器。RXDNE 标记是接收到每个字节后被硬件置位的。如果下一个数据已被收到或先前 DMA 请求还没被服务时，RXDNE 标志仍是置起的，溢出错误产生。

当溢出错误产生时：

- OREF 位被置位。
- RDR 内容将不会丢失。读 USART_DAT 寄存器仍能得到先前的数据。
- 移位寄存器中的内容将被覆盖。随后接收到的数据都将丢失。
- 如果 RXDNEIEN 位被设置或 ERRIEN 和 DMARXEN 位都被设置，中断产生。
- 顺序执行对 USART_STS 和 USART_DAT 寄存器的读操作，可复位 OREF。

注意：当 OREF 置位时，表明至少有 1 个数据已经丢失。有以下两种可能性：

- *如果 RXDNE=1，上一个有效数据还在接收寄存器 RDR 上，可以被读出。*
- *如果 RXDNE=0，这意味着上一个有效数据已经被读走，RDR 已经没有东西可读。当上一个有效数据在 RDR 中被读取的同时又接收到新的（也就是丢失的）数据时，此种情况可能发生。在读序列期间（在 USART_STS 寄存器读访问和 USART_DAT 读访问之间）接收到新的数据，此种情况也可能发生。*

  噪音错误使用过采样技术（同步模式除外），通过区别有效输入数据和噪音来进行数据恢复。

图 24-6 检测噪声的数据采样

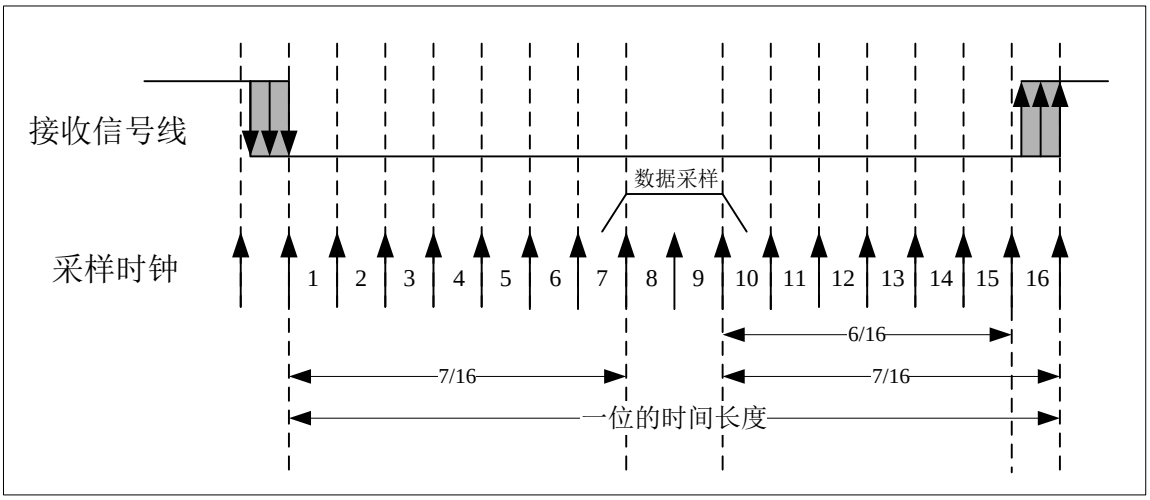


表 24-2 检测噪声的数据采样

采样值	NEF 状态	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

当在接收帧中检测到噪音时可以进行以下操作:

- 在 RDNE 位的上升沿设置 NEF 标志。
- 无效数据从移位寄存器传送到 USART_DAT 寄存器。
- 在单个字节通信情况下，没有噪声中断产生。但因为 NEF 标志位和 RXDNE 标志位是同时被置位，RXDNE 将产生中断。在多缓冲器通信情况下，如果已经设置了 USART_CTRL3 寄存器中 ERRIEN 位，将产生一个中断。

  顺序读 USART_STS，再读 USART_DAT 寄存器，将清除 NEF 标志位。

24. 4. 3. 6 帧错误

当以下情况发生时检测到帧错误：由于数据未同步或大量噪音的原因，停止位没有在预期的时间上接收和识别出来。

当帧错误被检测到时，硬件置起 FEF 位，无效数据将从移位寄存器传送到 USART_DAT 寄存器。

在单字节通信时，没有帧错误中断产生。然而，这个位和 RXDNE 位同时置起，后者将产生中断。

在多缓冲器通信情况下，如果 USART_CTRL3 寄存器中 ERRIEN 位被置位的话，将产生中断。

顺序执行对 USART_STS 和 USART_DAT 寄存器的读操作，可复位 FEF 位。

24.4.3.7 接收期间可配置的停止位

在数据接收期间，可以通过控制寄存器 2（USART_CTRL3 中的 STPB）的控制位来配置数据停止位的个数，在普通模式时，可以选择 1 或 2 个停止位。在智能卡模式里可以选择 0.5 或 1.5 个停止位。

1. 0.5 个停止位（智能卡模式中的接收）：不对 0.5 个停止位进行采样。因此，如果选择 0.5 个停止位则不能检测帧错误和断开帧。
2. 1 个停止位：默认情况下通过三个点对 1 个停止位的采样，选择第 8，第 9 和第 10 采样位上进行。
3. 1.5 个停止位(智能卡模式)：智能卡模式下发送数据时，器件必须检查数据是否被正确的发送出去。所以接收器功能块必须被激活（USART_CTRL2 寄存器中的 RXEN=1），并且在停止位的发送期间采样数据线上的信号。如果出现校验错误智能卡会在发送方采样 NACK 信号时拉低数据线，表示出现了帧错误。发送端的 FEF 在 1.5 个停止位结束时被置起。对 1.5 个停止位的采样是在第 16，第 17 和第 18 采样点进行的。而在智能卡发送端，1.5 个的停止位可以被分成一个是 0.5 个数据位的周期和 1 个数据位周期，0.5 个数据位的周期不做任何事情，随后是 1 个数据位的周期的停止位，在这段时间的中点处采样。对于智能卡接收端而言，1.5 个的停止位可以被分成一个是 0.5 个数据位的周期和 1 个数据位周期，判断是否有奇偶校验错误；如果有奇偶校验错误，则在随后的 1 个数据位的周期拉低数据总线；如果没有奇偶校验错误，则在随后的 1 个数据位的周期，释放数据总线（数据总线处于高电平）。详见第 24.4.11 节：智能卡模式。
4. 2 个停止位：对 2 个停止位的采样是在第一停止位的第 8，第 9 和第 10 个采样点完成的。如果第一个停止位期间检测到一个帧错误，帧错误标志将被置起。在第一个停止位结束时 RXDNE 标志将被设置。第二个停止位将不会检测帧错误。

24.4.4 分数波特率的产生

波特率分频系数是一个 16 位数字,其中分为 12 位整数部分和 4 位小数部分。波特率发生器使用这两部分组合所得的数值来确定波特率。由于具有小数部分的波特率分频系数，将使 USART 能够产生所有标准波特率。

波特率分频系数 （USARTDIV）与系统时钟（PCLK）具有如下关系：

$$\text{TX/RX 波特率} = f_{PCLK}/(16 * \text{USARTDIV})$$

这里的 f_{PCLK} 是给外设的时钟(PCLK1 用于 USART2、3、UART4、5, PCLK2 用于 USART1、UART6、7） USARTDIV 的值设置在 USART_BRCF 寄存器。

注意：在写入 USART_BRCF 之后，波特率计数器会被波特率寄存器的新值替换。因此，不要在通信过程中改变波特率寄存器的数值。

24.4.4.1 如何从 USART_BRCF 寄存器值得到 USARTDIV

例 1：

如果 DIV_Integer=27，DIV_Decimal=12（USART_BRCF =0x1BC），于是

DIV_Integer（USARTDIV）=27

DIV_Decimal（USARTDIV）=12/16=0.75

所以 USARTDIV=27.75

例 2:

要求 USARTDIV=25.62,就有:

DIV_Decimal=16*0.62=9.92

最接近的整数是：10=0x0A

DIV_Integer= DIV_Integer（25.620）=25=0x19

于是，USART_BRCF =0x19A

例 3:

要求 USARTDIV=50.99 就有:

DIV_Decimal=16*0.99=15.84

最接近的整数是：16=0x10=> DIV_Decimal[3：0]溢出=>进位必须加到小数部分

DIV_Integer= DIV_Integer（0d50.990+进位）=51=0x33

于是：USART_BRCF =0x330，USARTDIV=0d51.000

表 24-3 设置波特率时的误差计算

波特率		f_{pclk}=36M			f_{pclk}=72M		
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	937.5	0%	2.4	1875	0%
2	9.6	9.6	234.375	0%	9.6	468.75	0%
3	19.2	19.2	117.1875	0%	19.2	234.375	0%
4	57.6	57.6	39.0625	0%	57.6	78.125	0%
5	115.2	115.384	19.5	0.15%	115.2	39.0625	0%
6	230.4	230.769	9.75	0.16%	230.769	19.5	0.16%
7	460.8	461.538	4.875	0.16%	461.538	9.75	0.16%
8	921.6	923.076	2.4375	0.16%	923.076	4.875	0.16%
9	2250	2250	1	0%	2250	2	0%

10	4500	不可能	不可能	不可能	4500	1	0%

注意：

1. CPU 的时钟频率越低，则某一特定波特率的误差也越低。可以达到的波特率上限可以由这组数据得到。

2. USART1 使用 PCLK2（最高 72MHz）。其它 USART 使用 PCLK1（最高 36MHz）。

24. 4. 5 USART 接收器容忍时钟的变化

USART 异步接收器要求整体的时钟系统地变化小于 USART 异步接收器能够容忍的范围。影响这些变化的因素有：

- 1：由于发送器误差而产生的变化（包括发送器端振荡器的变化）
- 2：接收器端波特率取整所产生的误差
- 3：接收器端振荡器的变化
- 4：由于传输线路产生的变化（通常是由于收发器在由低变高的转换时序，与由高变低转换时序之间的不一致性所造成）。

需要满足以上四点所导致的误差： 1+2+3+4 < USART 接收器的容忍度

对于正常接收数据，USART 接收器的容忍度等于最大能容忍的变化，它依赖于下述选择：

- 由 USART_CTRL1 寄存器的 WL 位定义的 10 或 11 位字符长度
- 是否使用分数波特率产生

表 24-4 当 DIV_Decimal =0 时，USART 接收器的容忍度

WL 位	认为 NF 是错误	不认为 NF 是错误
0	3.75%	4.375%
1	3.41%	3.97%

表 24-5 当 DIV_Decimal !=0 时，USART 接收器的容忍度

WL 位	认为 NF 是错误	不认为 NF 是错误
0	3.33%	3.88%
1	3.03%	3.53%

注意：在特殊的情况下，即当收到的帧包含一些在 WL=0 时，正好是 10 位（WL=1 时是 11 位）的空闲帧，上面 2 个表格中的数据可能会有少许出入。

24. 4. 6 检验控制

设置 USART_CTRL1 寄存器上的 PCEN 位，使能奇偶控制（发送时生成一个奇偶位，接收时进行奇偶校验）。根据 M 位定义的帧长度，USART 帧格式列在下表。

表 24-6 帧格式

WL 位	PCEN 位	USART 帧
0	0	| 起始位 | 8 位数据 | 停止位 |
0	1	| 起始位 | 7 位数据 | 奇偶检验位 | 停止位 |
1	0	| 起始位 | 9 位数据 | 停止位 |
1	1	| 起始位 | 8 位数据 | 奇偶检验位 | 停止位 |

注意：在用地址标记唤醒设备时，地址的匹配只考虑到数据的 MSB 位，而不用关心校验位。（MSB 是数据位中最后发出的，后面紧跟校验位或者停止位）

传输模式：通过置位 USART_CTRL1 的 PCEN 位使能奇偶校验。如果奇偶校验失败，USART_STS 寄存器中的 PEF 标志被置'1'，如果设置了 USART_CTRL1 寄存器的 PEIEN，会产生中断。

奇校验：此校验位使得一帧中的 7 或 8 个 LSB 数据以及校验位中'1'的个数为奇数。

例如：数据=11011101，共有 6 个'1'，如果选择奇校验（在 USART_CTRL1 中的 PSEL = 1），校验位将是'1'。

偶校验：校验位使得一帧中的 7 或 8 个 LSB 数据以及校验位中'1'的个数为偶数。

例如：数据=11011101，共有 6 个'1'，如果选择偶校验（在 USART_CTRL1 中的 PSEL = 0），校验位将是'0'。

24. 4. 7 多处理器通信

多处理器通信指的是多个 USART 连接到一个共同的网络。例如某个 USART 设备可以是主，它的 TX 输出和其他 USART 从设备的 RX 输入相连接；USART 从设备各自的 TX 输出逻辑地与在一起，并且和主设备的 RX 输入相连接。

在多处理器配置中，对于一个设备来说，监视所有 RX 引脚是较大的处理器开销，可以开启静默模式，这样就可以减少由未被寻址的接收器的参与带来的多余的 USART 服务负担。

在静默模式下。

- 任何接收状态位都不会被设置。
- 所有接收中断被禁止。
- USART_CTRL1 寄存器中的 RCVWU 位被置 1。RCVWU 可以被硬件自动控制或在某个条件下由软件写入。

根据 USART_CTRL1 寄存器中的 WUM 位状态，USART 可以用二种方法进入或退出静默模式。

- 如果 WUM 位被复位：进行空闲总线检测。
- 如果 WUM 位被设置：进行地址标记检测。

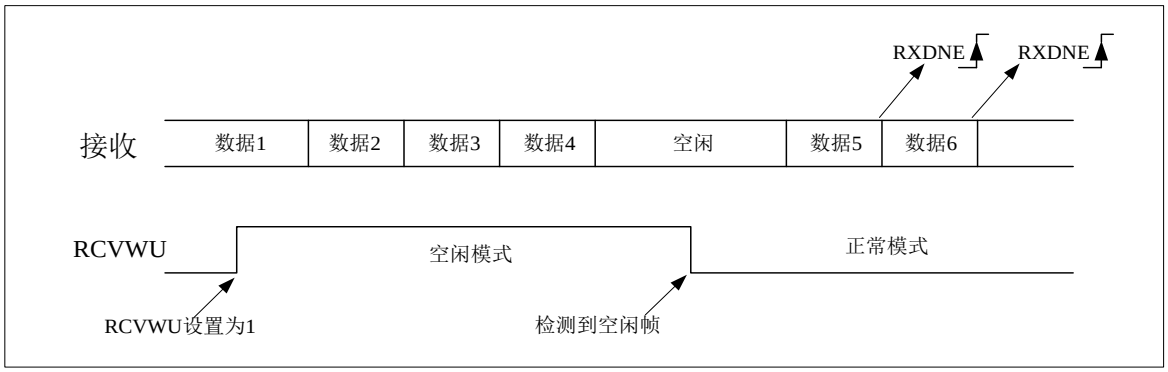
24. 4. 7. 1 空闲总线检测（WUM=0）

当 RCVWU 位置位，USART 进入静默模式。当 RX 引脚检测到一空闲帧时， RCVWU 被硬件清零，

但是 USART_STS 寄存器中的 IDLEF 位不会被置 1。RCVWU 也可以被软件写 0。

如下图 24-7 检测到空闲帧时 RCVWU 状态。

图 24-7 利用空闲总线检测的静默模式



24.4.7.2 地址标识（address mark）检测（WUM=1）

在这个模式里，最高位为地址标志位，如果为 1 的话，此字节为地址，否则为是数据。在一个地址字节中，目标接收器的地址被放在低 4 位中。如果这个 4 位地址于 USART_CTRL2 寄存器的 ADDR[3:0]不相同，

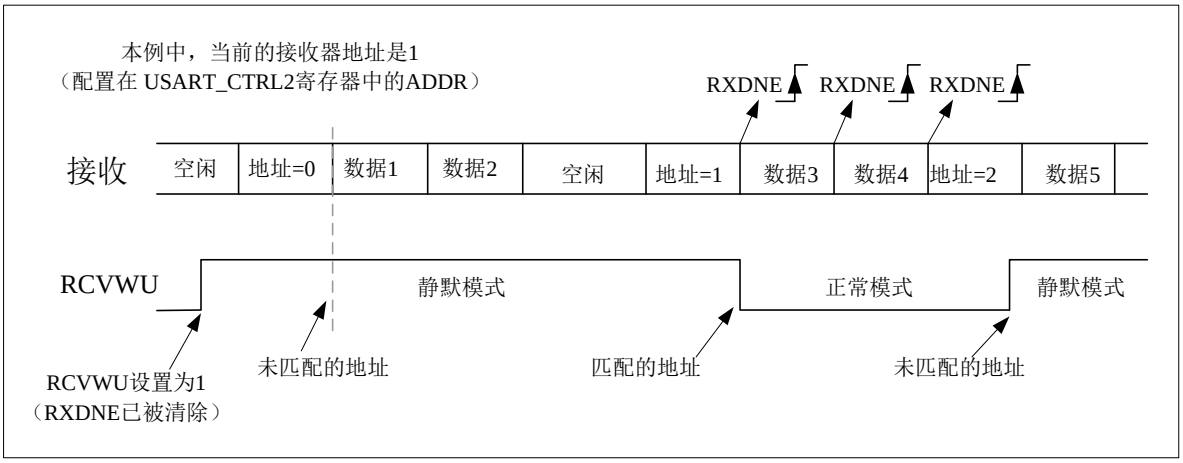
即接收到的字节与它的编程地址不匹配时，USART 将进入静默模式并设置 RCVWU 位。在这种情况下， RXDNE 也不会被置位。

当接收到的字节与接收器内编程地址相同时，接收将 USART 唤醒的数据帧，然后硬件清零 RCVWU 位，USART 退出静默模式，随后的字节被正常接收，收到这个匹配的地址字节时将设置 RXDNE 位，因为接收唤醒位 RCVWU 位已被清零。

当接收缓冲器不包含数据时（USART_STS 的 RXDNE=0），接收唤醒位 RCVWU 位可以被写 0 或 1。否则，该次写操作被忽略。

下图给出利用地址标记检测来唤醒和进入静默模式的例子。

图 24-8 利用地址标记检测的静默模式



24.4.8 LIN 模式

LIN 模式是通过设置 USART_CTRL2 寄存器的 LINMEN 位选择。在 LIN 模式下，以下寄存器必须清 0:

- USART_CTRL2 寄存器的 CLKEN 位
- USART_CTRL3 寄存器的 STPB[1:0]，SCMEN，HDMEN 和 IRDAMEN

24.4.8.1 LIN 发送与接收

LIN 的发送步骤与 USART 正常的操作流程有一些不同的地方，

1.发送普通数据帧时，LIN 发送过程与普通发送过程相同，但是数据长度只能为 8，所以清零 WL 位以配置 8 位字长，置位 USART_CTRL1 中的 SDBRK 将发送 13 位'0'作为断开符号。然后发一位停止位。

当 LIN 模式被使能时，断开符号检测功能完全独立于 USART 接收器，断开检测可以用在总线空闲时和发送某数据帧其间。

当接收器被激活时（USART_CTRL1 的 RXDEN=1），电路开始监测 RX 上的开始信号，当起始位被检测到后，采样每个位的第 8，9，10 个过采样时钟点上。如果 10 个或 11 个连续位都是'0'，并且又跟着一个定界符，USART_STS 的 LINBDF 标志被设置。如果 LINBDIEN 位=1，中断产生。在确认断开符号前，要检查定界符，因为它意味 RX 线已经回到高电平。

如果在第 10 或 11 个采样点之前采样到了'1'，检测电路重新寻找起始位且取消当前检测。LIN 模式如果被禁止的话，接收器不需要检测断开符号。

如果 LIN 模式使能（LINMEN=1），发生帧错误（也就是停止位检测到'0'，这种情况出现在断开帧），接收器将会被停止，直到断开符号检测电路接收到一个'1'（这种情况发生于断开符号没有完整的发出来），或一个定界符（这种情况发生于已经检测到一个完整的断开符号）。

图 24-9 为断开符号检测器状态机的行为和断开符号标志的关系。

图 24-10 为断开帧的例子。

图 24-9 LIN 模式下的断开检测（11 位断开长度–设置了 LINBDL 位）

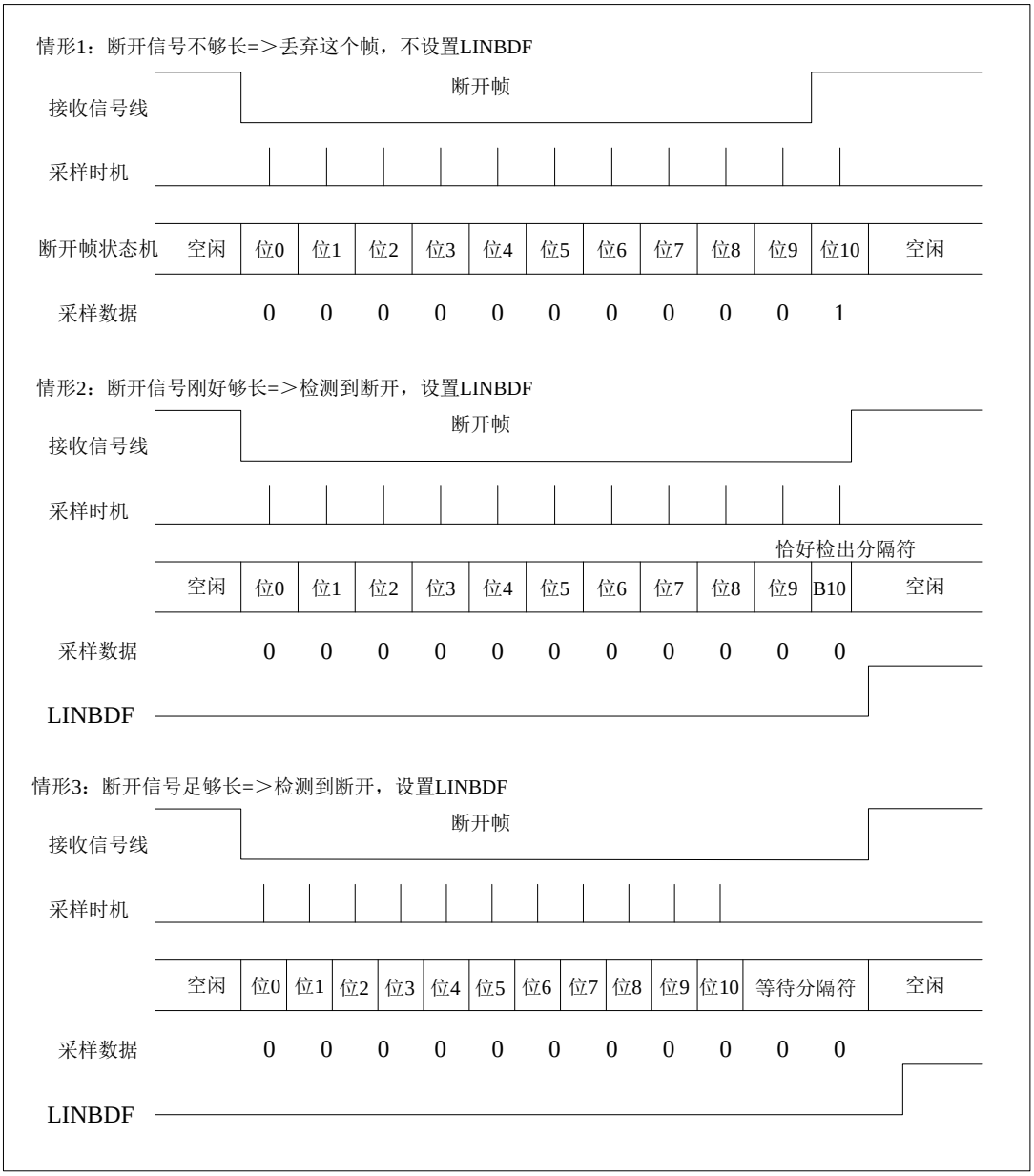
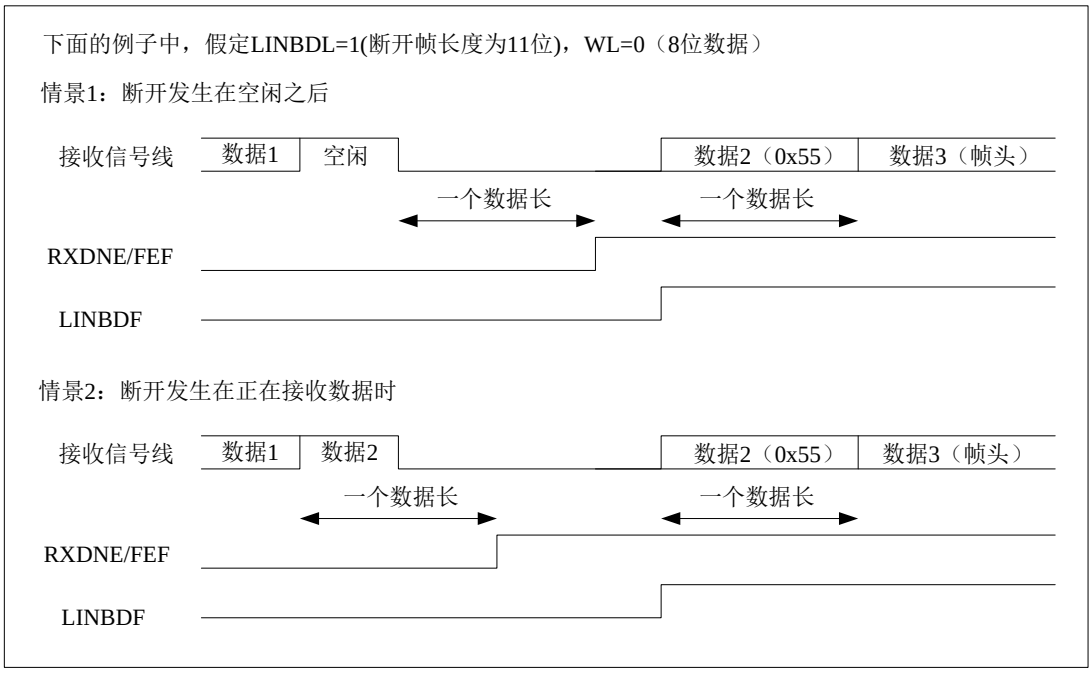


图 24-10 LIN 模式下的断开检测与帧错误的检测



24.4.9 USART 同步模式

通过在USART_CTRL2 寄存器上写CLKEN 位选择同步模式 ，USART_CTRL2 寄存器中的LINMEN 位，USART_CTRL3 寄存器中的 SCMEN、HDMEN 和 IRRAMEN 位必须为零。

USART 只支持主模式方式控制双向同步串行通信，不能用来自其他设备的输入时钟接收或发送数据（CK 永远是输出）。CK 脚作为 USART 发送器时钟的输出。在起始位和停止位期间，CK 脚不会输出时钟脉冲。USART_CTRL2 寄存器中 LBCLK 位决定最低位期间是否有时钟脉冲，其中 USART_CTRL2 寄存器的 CLKPOL 位选择时钟极性，CLKPHA 位选择外部时钟的相位(见图 24-11、图 24-12 和图 24-13)。

在总线空闲状态，实际数据到来之前以及发送断开符号的时候，外部 CK 时钟不被激活。CK 脚同 TX 脚一起联合工作。因而，只有在使能了发送器 (TXEN = 1)，并且发送数据时 (写入数据至 USART_DAT 寄存器）才提供时钟。同步模式的 USART 接收器工作方式与异步模式不同。如果 RXEN=1，数据在 CK 上采样，无需过采样。但必须考虑建立时间和持续时间（取决于波特率，1/16 位时间)。

注意:
1．同步模式不发送数据也不能接收同步数据。
2．LBCLK,CLKPOL 和 CLKPHA 位只能在同步模式不工作时配置。
3．同一条代码设置 TXEN 和 RXEN， 可以减少接收器的建立时间和保持时间。

图 24-11 USART 同步传输的例子

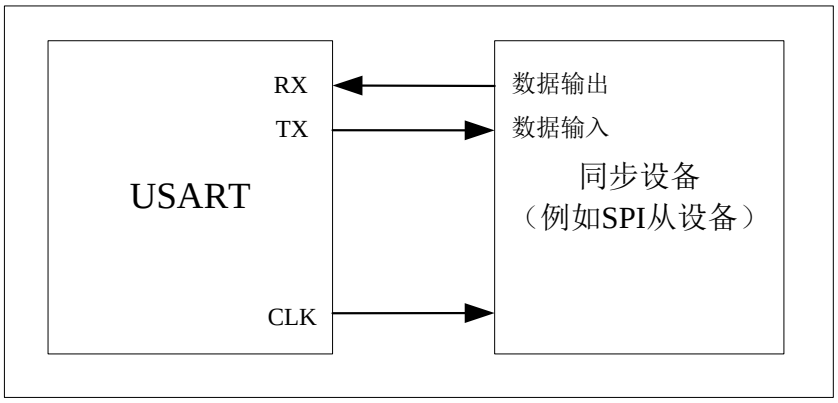


图 24-12 USART 数据时钟时序示例（WL=0）

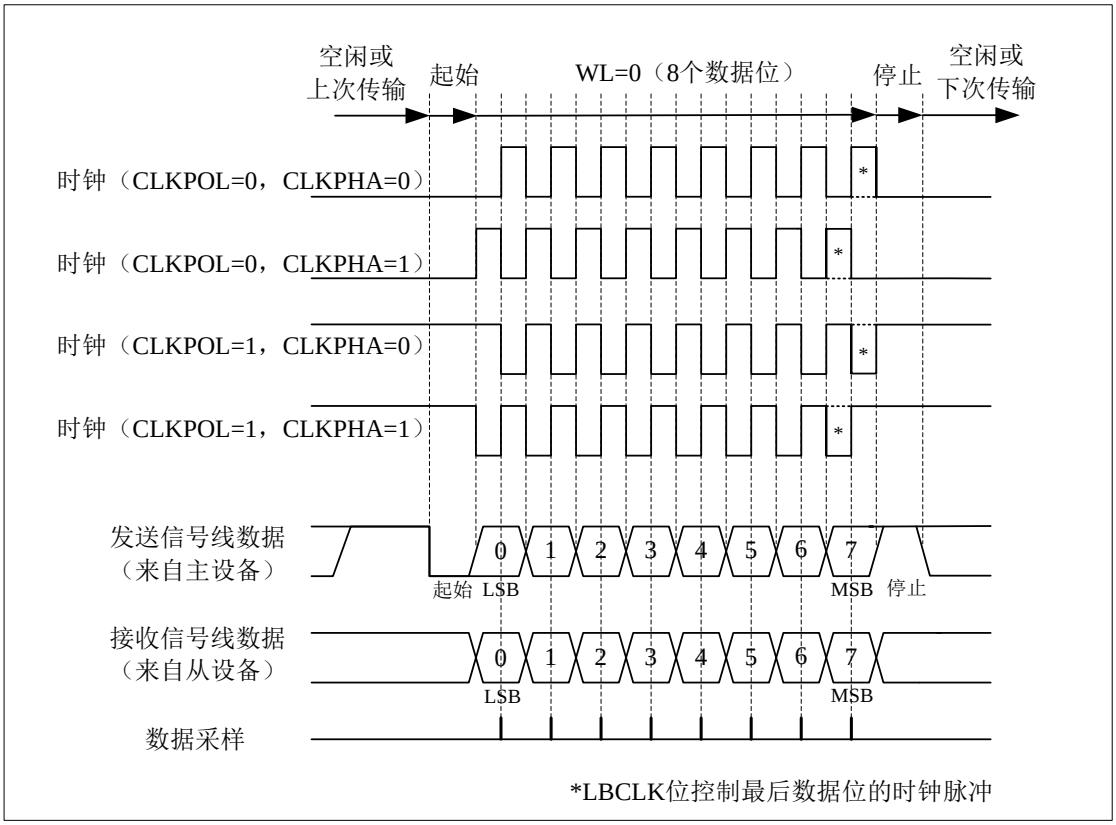


图 24-13 USART 数据时钟时序示例（WL=1）

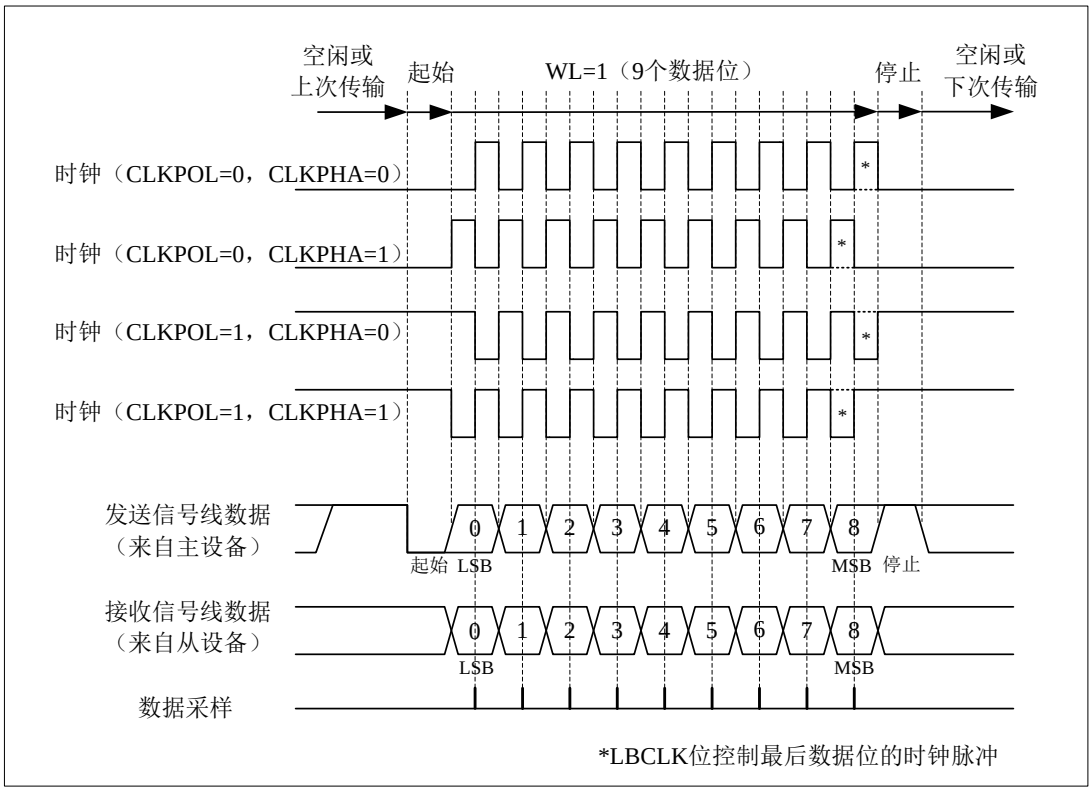
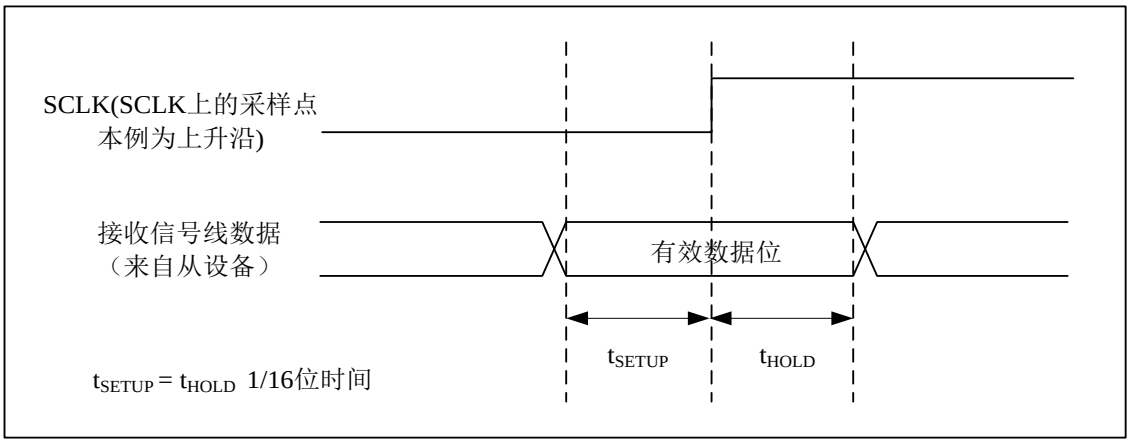


图 24-14 RX 数据采样/保持时间



注：在智能卡模式下 CK 的功能不同，有关细节请参考智能卡模式部分。

24. 4. 10 单线半双工通信

通过设置 USART_CTRL3 寄存器的 HDMEN 位选择单线半双方模式。下面的位必须置零，USART_CTRL2 寄存器的 LINMEN 和 CLKEN 位 ， USART_CTRL3 寄存器的 SCMEN 和 IRDAMEN 位

在单线半双工模式下, TX 和 RX 引脚在芯片内部互连, RX 引脚不再使用。USART_CTRL3 中的 HDMEN 位选择半双工和全双工通信。 当没有数据传输时，TX 总是被释放。因此，它在空闲状态的或接收状态时表现为一个标准 I/O 口。所以当 TX 口未被 USART 使能时，必须配置成悬空输入或者开漏的输出高。

通信的冲突由软件来管理（例如通过使用一个中央仲裁器）。

24.4.11 智能卡模式（ISO7816）

智能卡模式是一种异步通信模式，支持 ISO7816-3 协议。

设置 USART_CTRL3 寄存器的 SCMEN 位选择智能卡模式。在智能卡模式下，以下寄存器必须清零：USART_CTRL2 寄存器的 LINMEN 位,USART_CTRL3 寄存器的 HDMEN 位和 IRDAMEN 位

如果 CLKEN 位被置位，USART 通过 CK 引脚向智能卡提供时钟。

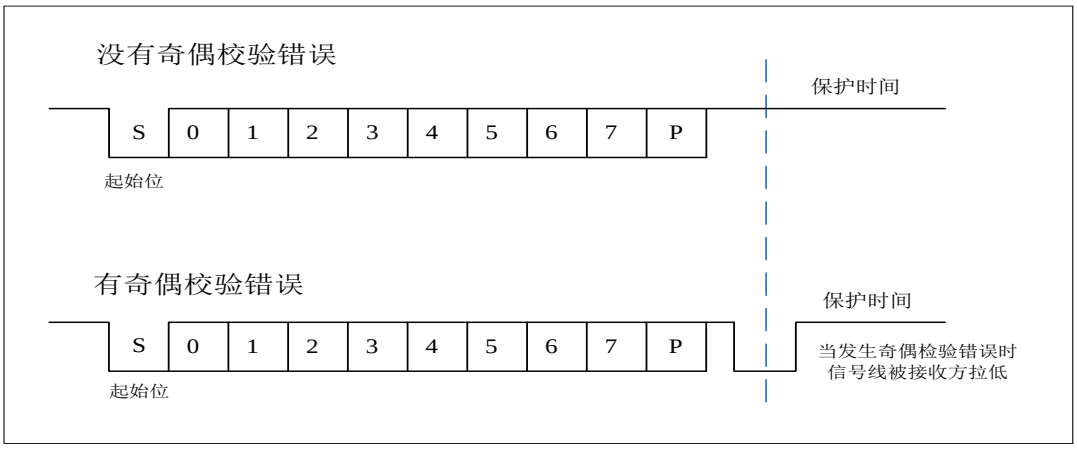
该接口支持智能卡异步协议。USART 应该被设置为：

- 8 位数据位加校验位： 此时 USART_CTRL1 寄存器中 WL=1、PCEN=1
- 发送和接收时为 1.5 个停止位：即 USART_CTRL2 寄存器的 STPB=11

注：也可以在接收时选择 1/2 个停止位，但为了避免在 2 种配置间转换，建议在发送和接收时使用 1.5 个停止位。

下图为有无校验位的两种情况。

图 24-15 ISO7816-3 异步协议



当与智能卡相连接时，TX 引脚需要被设置成开漏模式，外接上拉电阻，这个引脚将会与智能卡驱动同一条双向连线，输出使能位 TXEN 被置起在发送开始位和数据字节期间，在发送停止位过程中被释放（弱上拉），所以接收器可以将数据线拉低在发现校验错误的情况下。如果 TXEN 未使能，停止位期间 TX 将被拉到高电平。

- 从发送移位寄存器把数据发送出去，要延时大于等于半个波特时钟。在正常操作时，一个满的发送移位寄存器将在下一个波特时钟沿开始向外移出数据。在智能卡模式里，延时半个波特时钟。
- 接收一个设置为 0.5 或 1.5 个停止位的数据帧的过程中如果检测到一个奇偶校验错误，停止位结束时，发送线将会拉低一个波特时钟周期（NACK），通知智能卡 USART 未正确的收到数据。此 NACK 信号在发送端将产生一个帧错误，意味着发送端被配置成 1.5 个停止位。程序可以重新发送数据。如果设置了 SCNACK 控制位，发生校验错误时接收器会给出一个 NACK 信号；否则就不会发送 NACK。
- TXC 标志的置起可以通过编程保护时间寄存器得以延时。当发送移位寄存器清零并且没有新的发送请求时，TXC 被置高。在智能卡模式，空的发送移位寄存器将触发保护时间计数器开始向上计数，

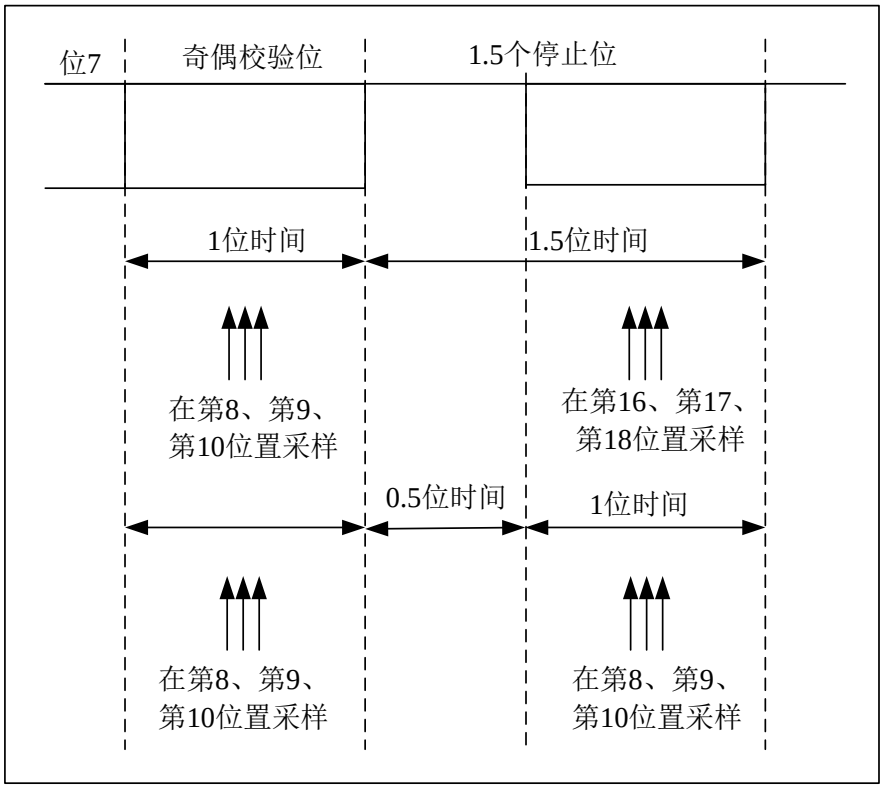
在计数值到达之前，TXC 被强制拉低，计数值到达后，TXC 被置高。

- 智能卡模式不会影响到 TXC 标志的撤销。
- 如果发送器检测到一个收到接收器的 NACK 信号，发送器的接收功能模块不会把 NACK 当作起始位检测。根据 ISO 协议，接收到的 NACK 的持续时间可以是 1 或 2 波特时钟周期。
- 在接收器这边，如果一个校验错误被检测到，NACK 也发送，接收器不会把 NACK 检测成起始位。

注意：1. 断开符号在智能卡模式里没有意义。一个带帧错误的 00h 数据将被当成数据而不是断开符号。
2. 当来回切换 TXEN 位时，没有 IDLE 帧被发送。ISO 协议没有定义 IDLE 帧。

图 24-16 为 USART 采样 NACK 信号。图中 USART 正在发送数据，并且被配置成 1.5 个停止位。为了检查数据的完整性和 NACK 信号，使能 USART 的接收功能块。

图 24-16 使用 1.5 停止位检测奇偶检验错



USART 可以为智能卡提供时钟通过 CK 输出。在智能卡模式里，CK 不和通信直接关联，而是先通过一个 5 位预分频器用内部的外设输入时钟来驱动智能卡的时钟。可以在预分频寄存器 USART_GTP 中配置分频率。CK 分频的频率从 $f_{CK}/2$ 到 $f_{CK}/62$，这里的 f_{CK} 是外设输入时钟。

24.4.12 串行 IrDA 红外解码功能

通过设置 USART_CTRL3 寄存器的 IRDAMEN 位选择 IrDA 模式，在此模式下 LINMEN、STPB、CLKEN、SCMEN、HDMEN 位需要清零。

IrDA SIR 物理层采用反相归零调制方案(RZI)，红外光源脉冲(RTZ 信号)表示逻辑 0(见图 24-17)。脉冲宽度规定为一个位周期的 3/16，IrDA 模式下，USART 数据帧由 SIR 发送编码器进行调制，调制后的

信号经由红外 LED 进行发送，经解调后将数据发送至 USART 接收器。对于编码器而言，波特率应小于 115200。

来自红外接收器的归零位比特流由 SIR 接收解码器进行解调，接收到的 NRZ 串行比特流再输出到 USART。在空闲状态里，解码器输入通常是高，发送编码器输出的极性和解码器的输入相反，当解码器输入低时，检测到一个起始位。

- 接收器可以与一低功耗发送器通信。
- SIR 发送逻辑把'0'作为高脉冲发送，把'1'作为低电平发送。脉冲的宽度规定为正常模式时位周期的 3/16（见图 24-18）。
- SIR 接收逻辑把高电平状态解释为'1'，把低脉冲解释为'0'。
- 发送编码器输出与解码器输入有着相反的极性。当空闲时，SIR 输出处于低状态。
- SIR 解码器把 IrDA 兼容的接收信号转变成给 USART 的比特流。
- 脉冲宽度是可编程的。IrDA 规范要求脉冲要宽于 1.41us。(PSCV 是在 IrDA 低功耗波特率寄存器 USART_GTP 中编程的预分频值)。那些宽度大于 2 个周期的被视为一个有效的脉冲。当 PSCV=0 时，IrDA 编码器/解码器不工作。
- 在 IrDA 模式里，USART_CTRL2 寄存器上的 STPB 位必须配置成 1 个停止位。

24.4.12.1 IrDA 低功耗模式的发送器与接收器

发送器在低功耗模式，脉冲宽度不再持续 3/16 个位周期。取而代之，脉冲的宽度是低功耗波特率的 3 倍，它最小可以是 1.42MHz。

接收器低功耗模式的接收类似于正常模式的接收。为了排除其它电源干扰，持续时间大于 2 个周期的 IrDA 低功耗波特率时钟的低电平信号被认为是有效的信号。

注意:
1. 宽度小于 1 个 PSCV 周期的脉冲一定被滤除掉，但是那些宽度大于 1 个而小于 2 个 PSCV 周期的脉冲可能被接收或滤除。
2.软件管理控制接收器的建立时间。IrDA 物理层技术规范协议规定了在发送和接收之间最小要有 10ms 的延时。

图 24-17 IrDA SIR ENDEC–框图

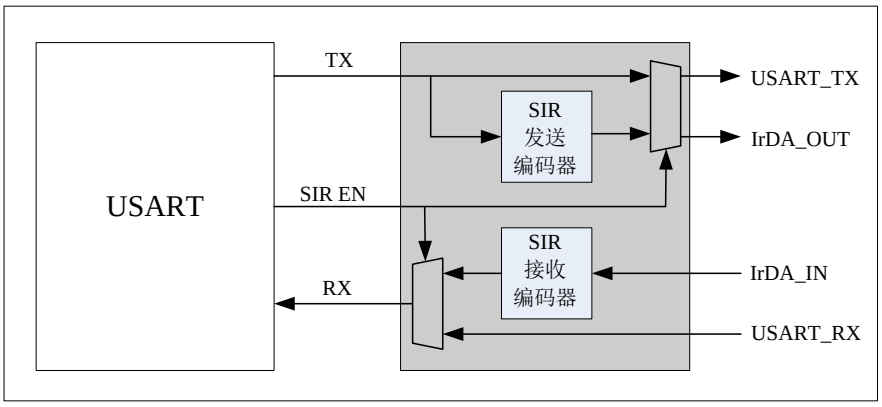
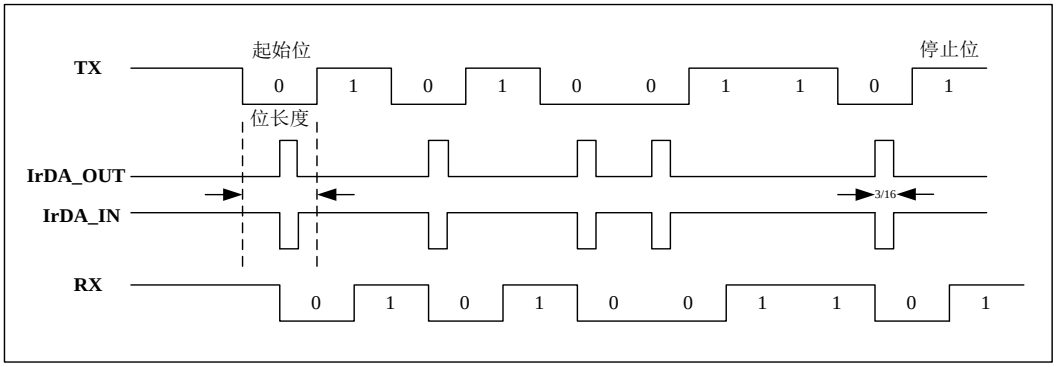


图 24-18 IrDA 数据调制（3/16）–普通模式



24.4.13 DMA 通信模式

USART 可以采用 DMA 的方式分别访问发送缓存区和接收缓存区。

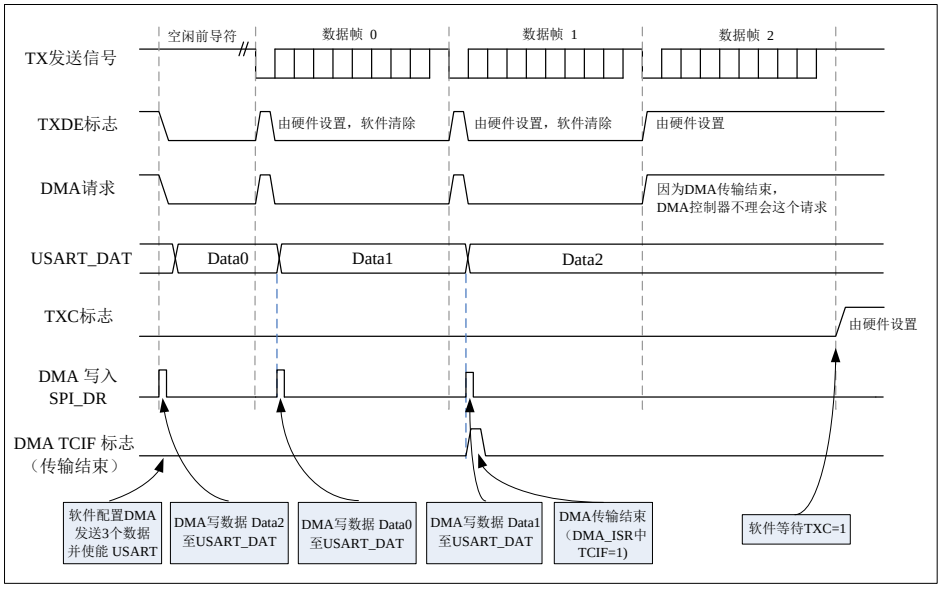
24.4.13.1 DMA 发送

为 USART 的发送分配一个 DMA 通道的步骤如下（x 表示通道号）:

1. USART_DAT 寄存器地址配置成 DMA 传输的目的地址，将存储器地址配置成 DMA 传输的源地址。
2. 配置要传输的总的字节数。
3. 配置通道优先级。
4. 配置在传输完成一半还是全部完成时产生 DMA 中断。
5. 激活该通道。

完成一次 DMA 传输，相应 DMA 通道上产生一次中断。在发送模式下，当 DMA 传输完所有要发送的数据时，DMA 控制器设置 DMA_INTSTS 寄存器的 TXCFx 标志，TXC 标志位由硬件置高表示传输完成，软件需要先等待 TXDE=1，再等待 TXC=1。

图 24-19 利用 DMA 发送



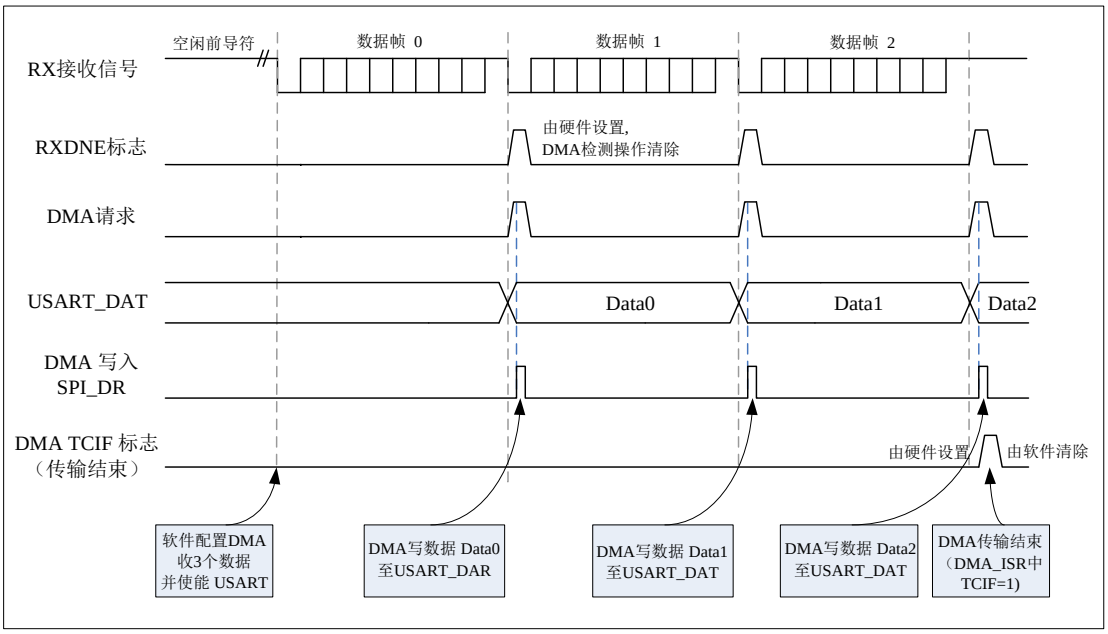
24.4.13.2 DMA 接收

为 USART 的接收分配一个 DMA 通道的步骤如下（x 表示通道号）：

1. 通过 DMA 控制寄存器把 USART_DAT 寄存器地址配置成传输的源地址，存储器地址设置传输的目的地址。
2. 配置要传输的 DMA 字节数。
3. 在 DMA 寄存器上配置通道优先级，配置传输。
4. 配置在传输完成一半还是全部完成时产生 DMA 中断。
5. 激活该通道。

当接收完成 DMA 控制器指定的传输量时，DMA 控制器在该 DMA 通道的中断矢量上产生一中断。

图 24-20 利用 DMA 接收



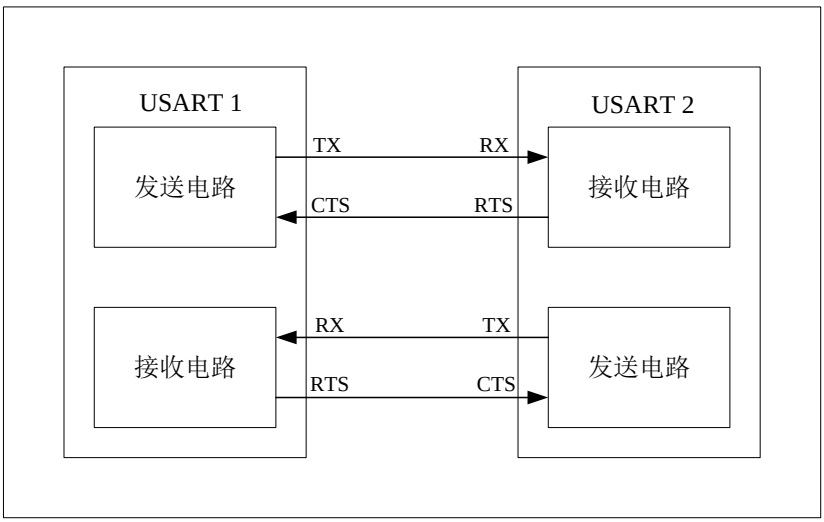
24.4.13.3 多缓冲通信中的错误标志和中断产生

在多缓冲器通信发生错误时，在当前字节传输完成将置起错误标志。如果中断使能位被设置，错误中断将产生。在单个字节接收时，和 RXDNE 一起被置起的帧错误、溢出错误与噪音标志，如果设置有单独的错误标志中断使能位，当前字节传输结束将会产生中断。

24.4.14 硬件流控

硬件流控制通过 nCTS 输入和 nRTS 输出实现功能。下图表明在这个模式里如何连接 2 个设备。

图 24-21 两个 USART 间的硬件流控制

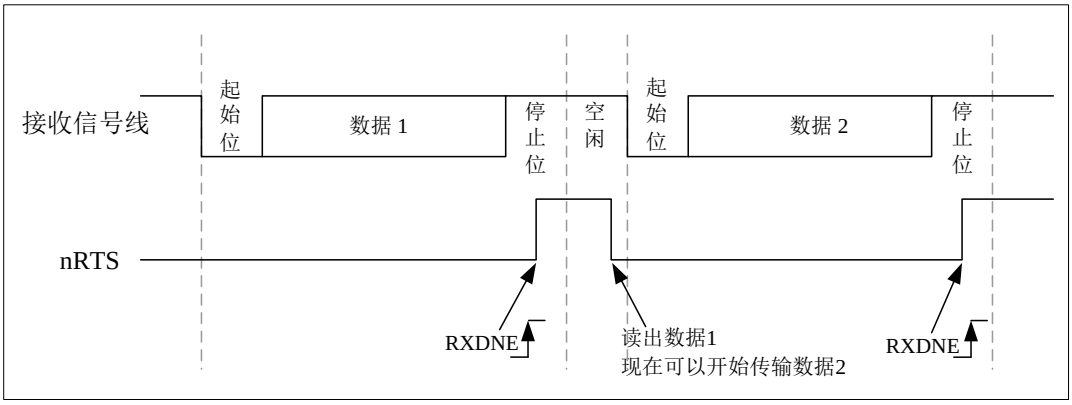


通过将 UASRT_CTRL3 中的 RTSEN 和 CTSEN 置位，可以分别独立地使能 RTS 和 CTS 流控制。

24.4.14.1 RTS 流控制

如果 RTS 流控制被使能（RTSEN=1），当一帧数据接收完，nRTS 为高电平，否则为低。下图是一个启用 RTS 流控制的通信的例子。

图 24-22 RTS 流控制

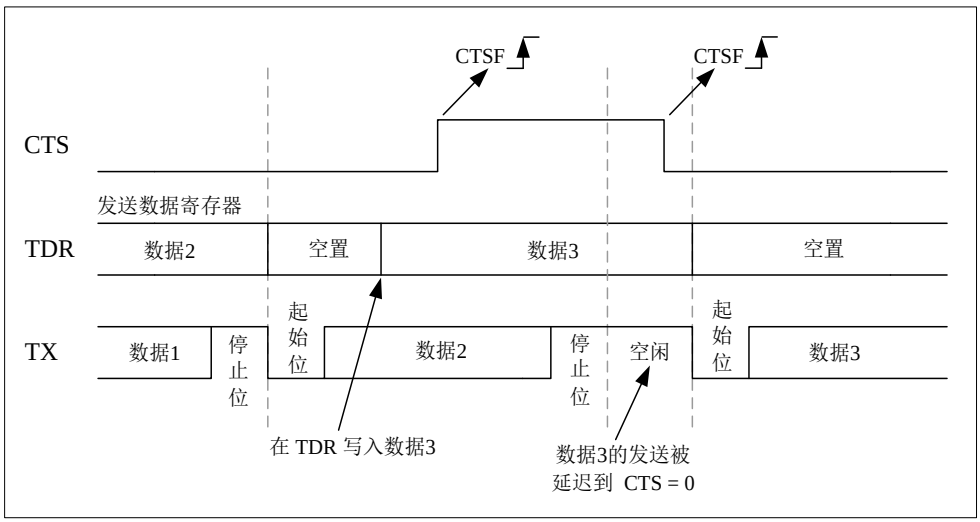


24.4.14.2 CTS 流控制

如果 CTS 流控制被使能时（CTSEN=1），发送器在发送下一帧前检查 nCTS 脚决定是否发送数据。如果 nCTS 被拉低（有效），发送器发送数据（假设那个数据是准备发送的，也就是 TXDE=0），。若 nCTS 在传输期间被拉高，当前数据帧传输完成后停止发送。

如果 CTS 流控制使能（CTSEN=1），nCTS 引脚信号位发生变化，CTSF 状态位置 1，如果设置了 USART_CTRL3 寄存器的 CTSIEN 位，则会产生中断，如图 24-23 开启 CTS 流控制。

图 24-23 CTS 流控制



24.5 USART 中断请求

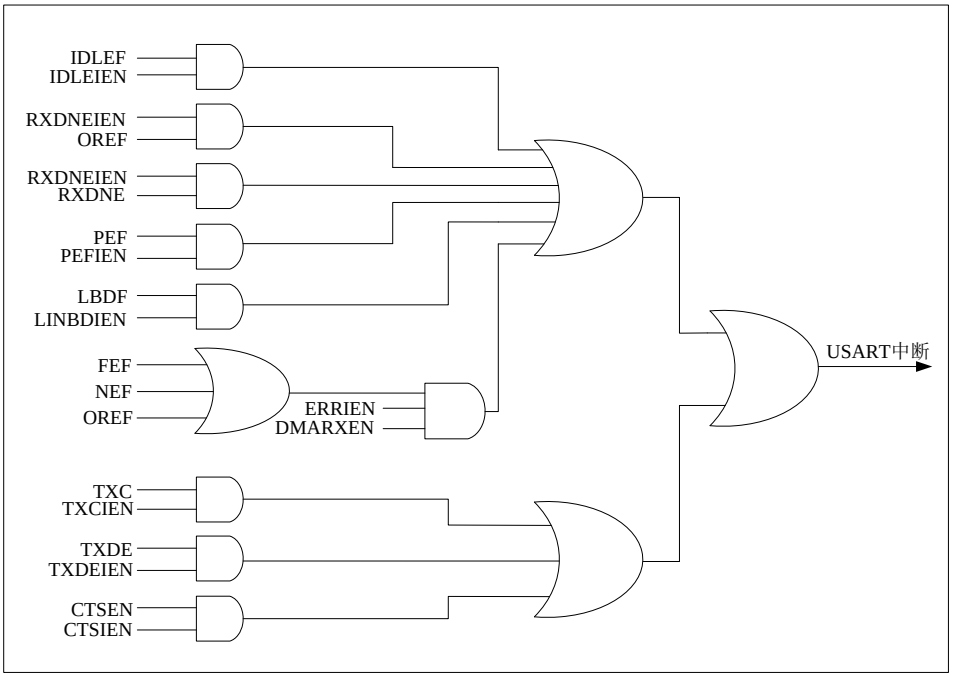
表 24-7 USART 中断请求

中断事件	事件标志	使能位
发送数据寄存器空	TXDE	TXDEIEN
CTS 标志	CTSF	CTSFIEN
发送完成	TXC	TXCIEN
接收数据就绪可读	RXDNE	RXDNEIEN
检测到数据溢出	ORERR	
检测到空闲线路	IDLEF	IDLEFIEN
奇偶检验错	PEF	PEFIEN
断开标志	LINBDF	LINBDFIEN
噪声标志，多缓冲通信中的溢出错误和帧错误	NEF/OREF/FEF	ERRIEN

1．仅当使用 DMA 接收数据时，才使用这个标志位。

USART 的各种中断事件是逻辑或的关系（见下图 24-24），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，但是同时只能产生一个中断请求。

图 24-24 USART 中断映像图



24.6 USART 模式配置

表 24-8 USART 模式设置(1)

通信模式	USART1	USART2	USART3	UART4	UART5	UART6	UART7
异步模式	Y	Y	Y	Y	Y	Y	Y
多处理器	Y	Y	Y	Y	Y	Y	Y
LIN	Y	Y	Y	Y	Y	Y	Y
同步模式	Y	Y	Y	N	N	N	N
单线模式（半双工）	Y	Y	Y	Y	Y	Y	Y
智能卡模式	Y	Y	Y	N	N	N	N
IrDA 红外模式	Y	Y	Y	Y	Y	Y	Y
DMA 通讯模式	Y	Y	Y	Y	Y	Y	Y
硬件流控模式	Y	Y	Y	N	N	N	N

（1）Y = 支持该模式，N = 不支持该模式

24.7 USART 寄存器

24.7.1 USART 寄存器地址映像

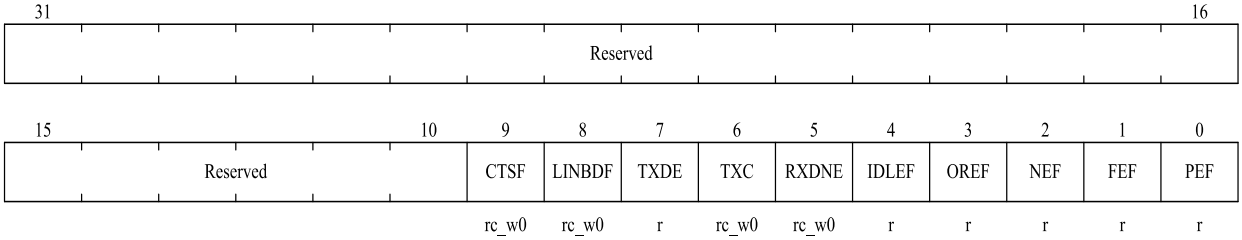
表 24-9 USART 寄存器地址映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	USART_STS	Reserved																						CTSF	LINBDF	TXDE	TXC	RXDNE	IDLEF	OREF	NEF	FEF	PEF
	Reset Value																							0	0	1	1	0	0	0	0	0	0
004h	USART_DAT	Reserved																						DATV[8:0]									
	Reset Value																							0	0	0	0	0	0	0	0	0	0
008h	USART_BRCF	Reserved																DIV_Integer[11:0]												DIV_Decimal [3:0]			
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00Ch	USART_CTRL1	Reserved																		UEN	WL	WUM	PCEN	PSEL	PEIEN	TXDEIEN	TXCIEN	RXDNEIEN	IDLEIEN	TXEN	RXEN	RCVWU	SDBRK
	Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	USART_CTRL2	Reserved																	LINMEN	STPB [1:0]		CLKEN	CLKPOL	CLKPHA	LBCLK	Reserved	LINBDIEN	LINBDL	Reserved	ADDR[3:0]			
	Reset Value																		0	0	0	0	0	0	0		0	0		0	0	0	0
014h	USART_CTRL3	Reserved																					CTSIEN	CTSEN	RTSEN	DMATXEN	DMARXEN	SCMEN	SCNACK	HDMEN	IRDALP	IRDAMEN	ERRIEN
	Reset Value																						0	0	0	0	0	0	0	0	0	0	0
018h	USART_GTP	Reserved																GTV[7:0]								PSCV[7:0]							
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.7.2 USART 状态寄存器（USART_STS）

偏移地址：0x00

复位值：0x0000 00C0



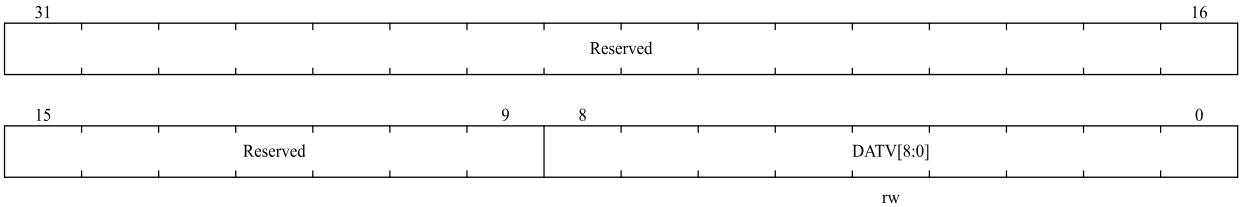
位域	名称	描述
31:10	Reserved	保留，硬件强制为 0。
9	CTSF	CTS 标志（CTS flag）。

位域	名称	描述
		如果设置了 USART_CTRL3 寄存器中 CTSEN 位，当 nCTS 输入变化时，该位由硬件置位。如果设置了 USART_CTRL3 寄存器中 CTSIEN 位，将产生中断。 该位由软件清 0。 0：nCTS 状态线没有变化。 1：nCTS 状态线发生变化。 *注意：该位对 UART4/5/6/7 无效。*
8	LINBDF	LIN 断开检测标志（LIN break detection flag）。 如果设置了 USART_CTRL2 寄存器中 LINMEN 位，当检测到 LIN 断开，该位由硬件置位。如果 USART_CTRL2 寄存器中 LINBDIEN 被置位时，将产生中断。 该位由软件清 0。 0：没有检测到 LIN 断开字符。 1：检测到 LIN 断开字符。
7	TXDE	发送数据缓冲区空（Transmit data register empty）。 上电复位或待发送数据已发送至移位寄存器后，该位置 1。USART_CTRL1 寄存器中 TXDEIEN 被置位将产生中断。 该位在软件将待发送数据写入 USART_DAT 时被清 0。 0：发送数据缓冲区不为空。 1：发送数据缓冲区空。
6	TXC	发送完成（Transmission complete）。 上电复位后，该位被置 1。如果 TXDE 置位，在当前数据发送完成时该位置 1。 USART_CTRL1 寄存器中 TXCIEN 被置位将产生中断。 该位由软件清 0。 0：发送没有完成。 1：发送完成。
5	RXDNE	读数据缓冲区非空（Read data register not empty）。 当读数据缓冲区接收到来自移位寄存器的数据时，该位置 1。当寄存器 USART_CTRL1 的 RXDNEIEN 位被置位，将会有中断产生。 软件可以通过对该位写 0 或读 USART_DAT 寄存器来将该位清 0。 0：读数据缓冲区为空。 1：读数据缓冲区不为空。
4	IDLEF	空闲线检测标志（IDLE line detected）。 在一个帧时间内，在 RX 引脚检测到空闲状态，该位置 1。当寄存器 USART_CTRL1 的 IDLEIEN 位被置位，将会有中断产生。 软件先读 USART_STS，再读 USART_DAT 可清除该位。 0：未检测到空闲帧。 1：检测到空闲帧。 *注意：IDLEF 位不会再次被置高直到 RXDNE 位被置起（即又检测到一次空闲总线）。*
3	OREF	溢出错误（Overrun error）。 在 RXDNE 置位的情况下，如果 USART_DAT 寄存器接收到来自移位寄存器

位域	名称	描述
		的数据，该位置 1。当寄存器 USART_CTRL3 的 ERRIEN 位被置位，将会有中断产生。 软件先读 USART_STS，再读 USART_DAT 可清除该位。 0：没有检测到溢出错误。 1：检测到溢出错误。
2	NEF	噪声错误标志（Noise error flag）。 在接收到的帧检测到噪音时，由硬件对该位置位。由软件序列对其清零（先读 USART_STS，再读 USART_DAT）。 0：没检测到噪声错误。 1：检测到噪声错误。 *注意：该位不会产生中断，因为它和 RXDNE 一起出现，硬件会在设置 RXDNE 标志时产生中断。在多缓冲区通信模式下，如果设置了 ERRIEN 位，则设置 NEF 标志时会产生中断。*
1	FEF	帧错误（Framing error）。 当检测到同步错位，过多的噪声或者检测到断开符，该位被硬件置位。由软件序列将其清零（先读 USART_STS，再读 USART_DAT）。 0：未检测到帧错误。 1：检测到帧错误或者断开帧（break frame）。 *注意：该位不会产生中断，因为它和 RXDNE 一起出现，硬件会在设置 RXDNE 标志时产生中断。如果当前传输的数据既产生了帧错误，又产生了过载错误，硬件还是会继续该数据的传输，并且只设置 OREF 标志位。在多缓冲区通信模式下，如果设置了 ERRIEN 位，则设置 FEF 标志时会产生中断。*
0	PEF	校验错误（Parity error）。 当接收到的数据帧校验位与预期校验值不同时，该位置位。 软件先读 USART_STS，再读 USART_DAT 可清除该位。 0：没检测到校验错误。 1：检测到校验错误。

24.7.3 USART 数据寄存器（USART_DAT）

偏移地址：0x04
复位值：未定义（不确定值）

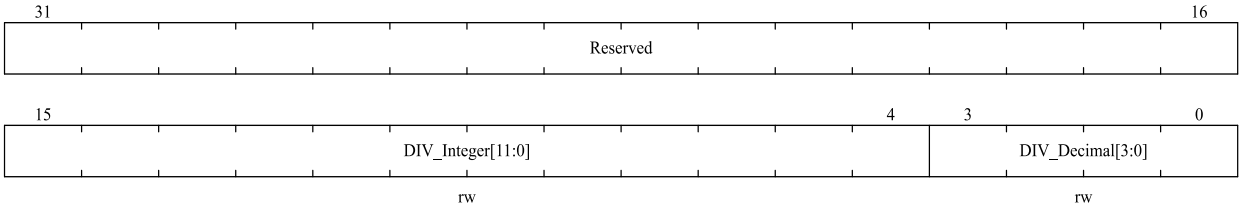


位域	名称	描述
31:9	Reserved	保留，硬件强制为 0。

位域	名称	描述
8:0	DATV[8:0]	数据值（Data value） 包含了发送或接收的数据；软件可以通过写这些位来改变发送数据，或读这些位的值来获取接收数据。 如果使能了奇偶校验，当发送数据被写入寄存器，数据的最高位（第 7 位或第 8 位取决于 USART_CTRL1 寄存器的 WL 位）将被校验位取代。

24.7.4 USART 波特率配置寄存器（DMA_BRCF）

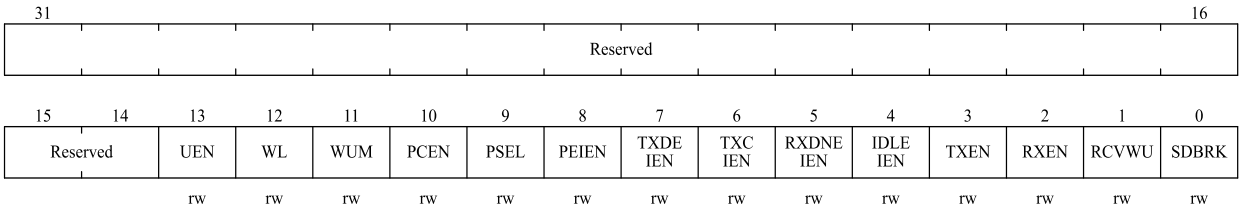
地址偏移：0x08
复位值：0x0000 0000
注意：使能 USART（UEN=1）时，不能写该寄存器；如果 TE 或 RE 被分别禁止，波特计数器停止计数。



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:4	DIV_Integer [11:0]	波特率分频器的整数部分。
3:0	DIV_Decimal[3:0]	波特率分频器的小数部分。

24.7.5 USART 控制寄存器 1（USART_CTRL1）

偏移地址：0x0C
复位值：0x0000 0000



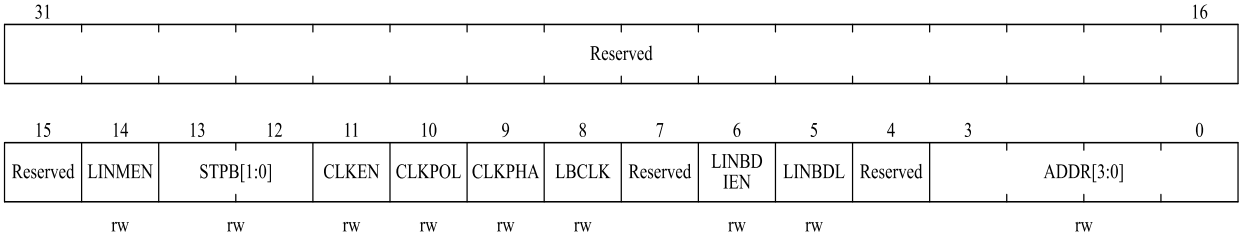
位域	名称	描述
31:14	Reserved	保留，硬件强制为 0。
13	UEN	USART 使能（USART enable）。 当该位被清零，在当前字节传输完成后 USART 的分频器和输出停止工作，以减少功耗。该位由软件设置和清零。 0：USART 禁用。 1：USART 使能。
12	WL	字长（Word length）。 0：8 数据位。

位域	名称	描述
		1：9 数据位。 *注意：在数据传输过程中（发送或者接收时），不能修改这个位。*
11	WUM	从静默模式唤醒方法（Wake up mode）。 0：空闲帧唤醒。 1：地址标识唤醒。
10	PCEN	校验控制使能（Parity control enable）。 0：校验控制禁用。 1：校验控制被使能。
9	PSEL	校验模式（Parity selection）。 0：偶校验。 1：奇校验。
8	PEIEN	校验错误中断使能（PE interrupt enable）。 如果该位置 1，USART_STS 寄存器中 PEF 被置位时产生中断。 0：校验错误中断禁用。 1：校验错误中断使能。
7	TXDEIEN	发送缓冲区空中断使能（TXDE interrupt enable）。 如果该位置 1，USART_STS 寄存器中 TXDE 被置位时产生中断。 0：发送缓冲区空中断禁止。 1：发送缓冲区空中断使能。
6	TXCIEN	发送完成中断使能（Transmission complete interrupt enable）。 如果该位置 1，USART_STS 寄存器中 TXC 被置位时产生中断。 0：发送完成中断禁用。 1：发送完成中断使能。
5	RXDNEIEN	读数据缓冲区非空中断和过载错误中断使能（RXDNE interrupt enable）。 如果该位置 1，USART_STS 寄存器中 RXDNE 或 OREF 被置位时产生中断。 0：读数据缓冲区非空中断和过载错误中断禁用。 1：读数据缓冲区非空中断和过载错误中断使能。
4	IDLEIEN	IDLE 线检测中断使能（IDLE interrupt enable）。 如果该位置 1，USART_STS 寄存器中 IDLEF 被置位时产生中断。 0：IDLE 线检测中断禁用。 1：IDLE 线检测中断禁用使能。
3	TXEN	发送器使能（Transmitter enable）。 0：发送器禁用。 1：发送器使能。
2	RXEN	接收器使能（Receiver enable）。 0：接收器禁用。 1：接收器使能。
1	RCVWU	接收器从静默模式中唤醒（Receiver wakeup） 软件可以通过将该位置 1 使得 USART 进入静默模式，将该位清 0 唤醒 USART。 空闲帧唤醒模式下（WUM=0），当检测到空闲帧时，该位由硬件清 0。地址标识唤醒模式下（WUM=1），当接收到一个地址匹配帧时，该位由硬件清

位域	名称	描述
		0；或接收到一个地址非匹配帧时，由硬件置 1。 0：接收器处于普通工作模式。 1：接收器处于静默模式。
0	SDBRK	发送断开帧（Send break）。 软件通过将该位置 1 发送断开帧。 断开帧传输结束由硬件清 0 该位。 0：没有发送断开帧。 1：发送断开帧。

24.7.6 USART 控制寄存器 2（USART_CTRL2）

偏移地址：0x10
复位值：0x0000 0000



位域	名称	描述
31:15	Reserved	保留，硬件强制为 0。
14	LINMEN	LIN 模式使能（LIN mode enable） 0：LIN 模式禁用 1：LIN 模式使能
13:12	STPB[1:0]	停止位长（STOP bits）。 00：1 停止位。 01：0.5 停止位。 10：2 停止位。 11：1.5 停止位。 *注意：对于 UART4/5/6/7，只有 1 位停止位和两位停止位是有效的。*
11	CLKEN	时钟使能（Clock enable） 0：CK 引脚禁用 1：CK 引脚使能 *注意：该位对 UART4/5/6/7 无效。*
10	CLKPOL	时钟极性（Clock polarity）。 该位用来设定在同步模式下 CK 引脚的极性。 0：CK 引脚不对外发送时保持为低电平。 1：CK 引脚不对外发送时保持为高电平。 *注意：该位对 UART4/5/6/7 无效。*
9	CLKPHA	时钟相位（Clock phase）。 该位用来设定在同步模式下 CK 引脚的相位。 0：在首个时钟边沿采样第一个数据。

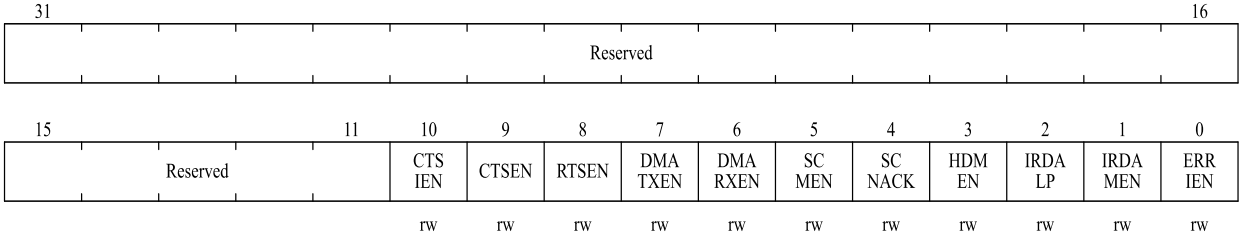
位域	名称	描述
		1：在第二个时钟边沿采样第一个数据。 *注意：该位对 UART4/5/6/7 无效。*
8	LBCLK	最后一位时钟脉冲（Last bit clock pulse）。 该位用来设定在同步模式下是否在 CK 引脚上输出最后发送的那个数据字节（MSB）对应的时钟脉冲。 0：最后一位数据的时钟脉冲不从 CK 输出。 1：最后一位数据的时钟脉冲会从 CK 输出。 *注意：该位对 UART4/5/6/7 无效。*
7	Reserved	保留，硬件强制为 0。
6	LINBDIEN	LIN 断开帧检测中断使能（LIN break detection interrupt enable）。 如果该位置 1，当 USART_STS 寄存器中 LBDF 被置位时将产生中断。 0：断开信号检测中断禁用 1：断开信号检测中断使能
5	LINBDL	LIN 断开帧检测长度（LIN break detection length）。 该位用来设定在断开帧长度。 0：10 位 1：11 位 *注意：LINBDL 可用于 LIN 模式及其他模式下的断开帧的检测长度控制，且检测长度和 LIN 模式相同。*
4	Reserved	保留，硬件强制为 0。
3:0	ADDR[3:0]	USART 地址。 在多处理器通信下的静默模式中使用的，使用地址标识来唤醒某个 USART 设备。 地址标识唤醒模式下（WUM=1），如果接收到的数据帧低四位与 ADDR[3:0]值不相等，USART 就会进入静默模式；如果接收到的数据帧低四位与 ADDR[3:0]值相等，USART 就会被唤醒。

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

24.7.7 USART 控制寄存器 3（USART_CTRL3）

偏移地址：0x14

复位值：0x0000 0000



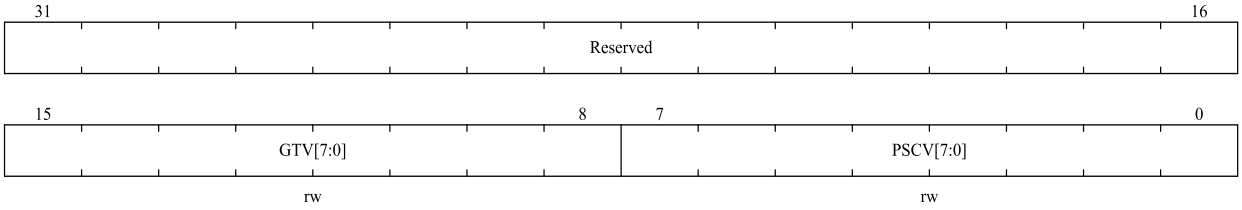
位域	名称	描述
31:11	Reserved	保留，硬件强制为 0。
10	CTSIEN	CTS 中断使能（CTS interrupt enable）。 如果该位置 1，当 USART_STS 寄存器中 CTSF 被置位时将产生中断。

位域	名称	描述
		0：CTS 中断禁用。 1：CTS 中断使能。 *注意：该位对 UART4/5/6/7 无效。*
9	CTSEN	CTS 使能（CTS enable）。 该位用于使能 CTS 硬件流控制功能。 0：CTS 硬件流控制禁用。 1：CTS 硬件流控制使能。 *注意：该位对 UART4/5/6/7 无效。*
8	RTSEN	RTS 使能（RTS enable）。 该位用于使能 RTS 硬件流控制功能。 0：RTS 硬件流控制禁用。 1：RTS 硬件流控制使能。 *注意：该位对 UART4/5/6/7 无效。*
7	DMATXEN	DMA 发送使能（DMA transmitter enable）。 0：DMA 发送模式禁用。 1：DMA 发送模式使能。
6	DMARXEN	DMA 接收使能（DMA receiver enable）。 0：DMA 接收模式禁用。 1：DMA 接收模式使能。
5	SCMEN	智能卡模式使能（Smart card mode enable）。 该位用于使能智能卡模式。 0：智能卡模式禁用。 1：智能卡模式使能。 *注意：该位对 UART4/5/6/7 无效。*
4	SCNACK	在智能卡模式 NACK 使能（Smart card NACK enable）。 该位用于智能卡模式在奇偶校验错误发生时使能发送 NACK。 0：当出现校验错误时不发送 NACK。 1：当出现校验错误时发送 NACK。 *注意：该位对 UART4/5/6/7 无效。*
3	HDMEN	半双工模式使能（Half-duplex mode enable）。 该位用于使能半双工模式。 0：半双工模式禁用。 1：半双工模式使能。
2	IRDALP	IrDA 低功耗模式（IrDA low-power）。 该位用于为 IrDA 模式选择低功耗模式。 0：正常模式。 1：低功耗模式。
1	IRDAMEN	IrDA 模式使能（IrDA mode enable）。 0：IrDA 禁用。 1：IrDA 使能。
0	ERRIEN	错误中断使能（Error interrupt enable）。 当 DMA 接收模式（DENR=1）使能时，如果该位被置 1，USART_STS 寄存

位域	名称	描述
		器中 FEF、OREF、NEF 被置位将产生中断。 0：错误中断禁用。 1：错误中断使能。

24.7.8 USART 保护时间和预分频寄存器（USART_GTP）

偏移地址：0x18
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:8	GTV[7:0]	智能卡模式下的保护时间值（Guard time value）。 该位域规定了以波特时钟为单位的保护时间。在智能卡模式下，需要这个功能。TXC 标志置位时间延时 GUAT[7:0]个波特时钟周期。 *注意：该位对 UART4/5/6/7 无效。*
7:0	PSCV[7:0]	预分频器值（Prescaler value）。 在 IrDA 低功耗模式下，这些位用来设定将外设时钟（PCLK1/PCLK2）分频产生低功耗频率的分频系数。 00000000：保留 – 不要写入该值 00000001：对源时钟 1 分频 … 11111111：对源时钟 255 分频 在 IrDA 正常模式下，PSCV 只能设置成 00000001。 在智能卡模式下，PSCV[4:0]用于设定外设时钟（APB1/APB2）生成智能卡时钟的分频 系数。实际的分频系数为 PSCV[4:0]设定值的两倍。 00000：保留 – 不要写入该值 00001：对源时钟 2 分频 00010：对源时钟 4 分频 … 11111：对源时钟 62 分频 在智能卡模式下，PSCV[7:5]保留。 *注意：该位对 UART4/5/6/7 无效。*

25 四线串行外设接口（QSPI）

25.1 QSPI 简介

QSPI 是用于单/双/四线 SPI 外设通信的接口。可以在间接和内存映射 2 种模式下工作。

间接模式：使用 QSPI 寄存器执行所有操作。

内存映射模式：外部闪存映射至微控制器地址空间，系统将其视为内部存储空间。

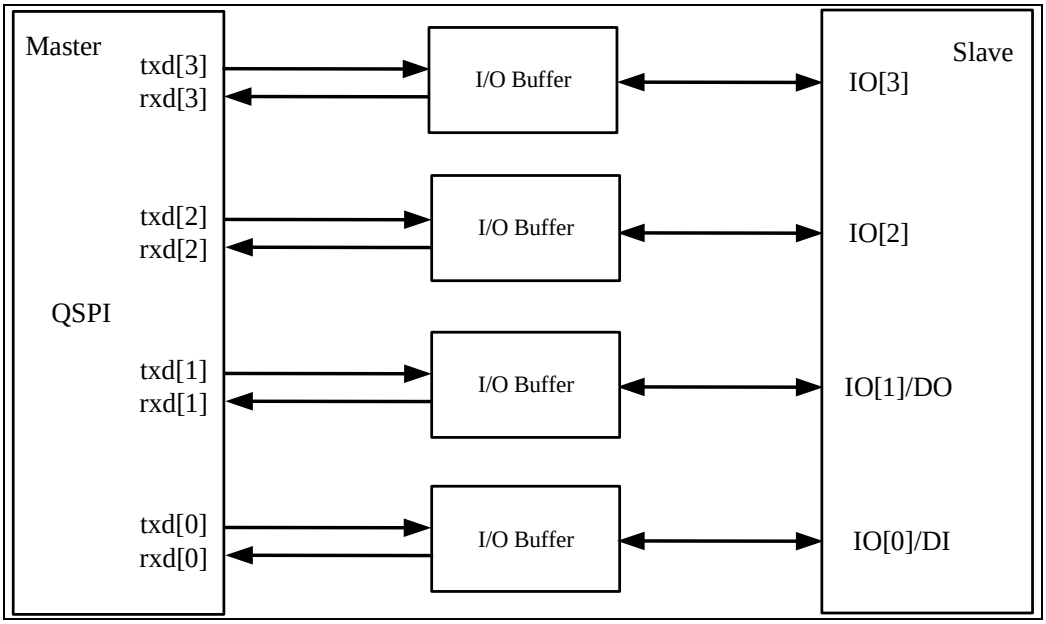
25.2 QSPI 主要特性

QSPI 控制器的主要特性如下：

- 支持 1 路 QSPI，可以配置成 Single SPI/Dual SPI/Quad SPI 模式。在 Single 模式下，支持标准的 SPI 操作，可以工作在半双工、全双工模式下；
- SPI 的操作方式可以配置成间接模式或内存映射模式，指令阶段的命令码可配置，模式字节阶段的模式字节可以配置；
- 支持 8-bit、16-bit、32-bit 的数据访问方式；
- 带数据收发 FIFO；
- 支持 DMA 操作；
- 支持 FIFO 中断、操作完成中断、超时中断、数据访问错误中断；
- 最大速度支持 4x36Mbps；
- 在间接模式或内存映射模式下，操作分为指令阶段、地址阶段、模式字节阶段、Dummy 阶段、数据阶段，这几个阶段可以配置为略过。

25.3 功能描述

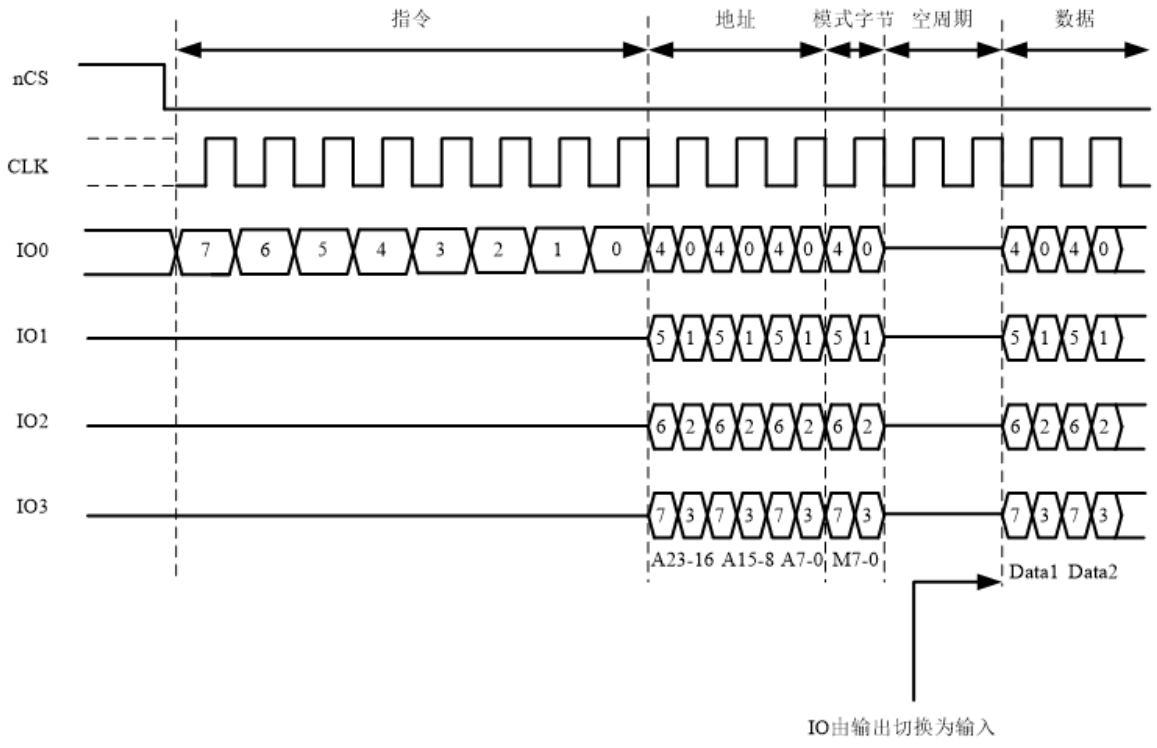
图 25-1 QSPI 功能框图



25.4 QSPI 命令序列

QSPI 通过命令与外设通信，每条命令包括指令、地址、模式字节、空指令和数据五个阶段，任一阶段均可跳过，但至少要包含指令、地址、模式字节或数据阶段之一。

图 25-2 QSPI 命令序列



25.5 操作流程

25.5.1 QSPI 间接模式

在间接模式下，通过写入 QSPI 寄存器来启动命令，并通过读写数据寄存器来传输数据，其方式与其它通信外设相同。

当 TMOD=00（QSPI_CTRL0[11：10]）时，处于发送与接收模式，发送/接收数据均有效。传输数据持续进行，直到发送 FIFO 为空为止。从外部设备接收的数据存储在接收 FIFO 存储器中，主机处理器可以访问该数据。

当 TMOD=01（QSPI_CTRL0[11：10]）时，QSPI 处于间接发送模式，其待发送字节在数据发送阶段送到闪存，通过写入数据寄存器 DATx 来提供数据。

当 TMOD=10（QSPI_CTRL0[11：10]）时，QSPI 处于间接接收模式，其待接收数据在数据接收阶段从闪存接收，通过读取数据寄存器 DATx 来获取数据。

当 TMOD=11（QSPI_CTRL0[11：10]）时，EEPROM 读取模式，发送数据用于将操作码/地址发送到 EEPROM 设备。

要读写的字节数在 NDF（QSPI_CTRL1[15：0]）中指定。

25.5.2 QSPI 间接发送操作

SPI_FRF（QSPI_CTRL0[23：22]）指定帧发送的格式（1/2/4 模式），配置 0x02 为 4 线模式

DFS（QSPI_CTRL0[4：0]）指定数据的长度，4 ~ 32bit

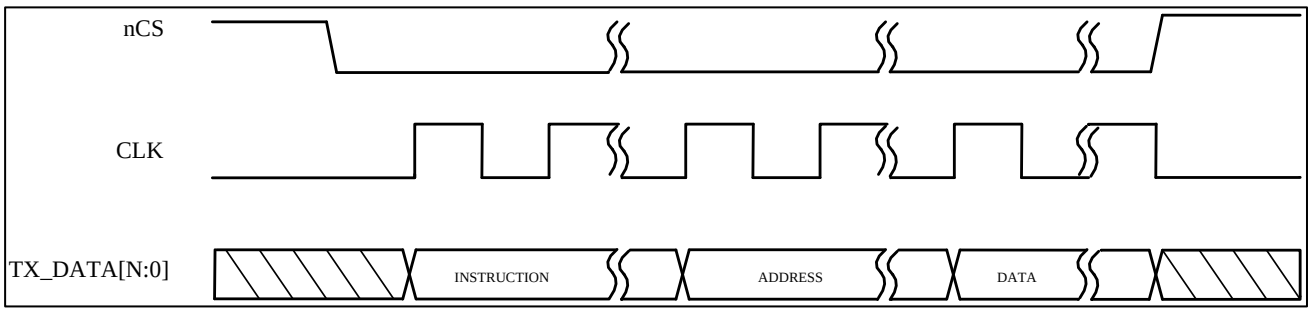
ADDR_LEN（QSPI_ENH_CTRL0 [5：2]），4bit ~ 60bit，可配置跳过地址阶段

INST_L（QSPI_ENH_CTRL0 [9：8]），4bit ~ 16bit，可配置跳过指令阶段

注意：1 条指令占用 1 个 FIFO 地址，地址可以占用多个 FIFO 位置。指令和地址都必须在数据寄存器 DATx 中编程。

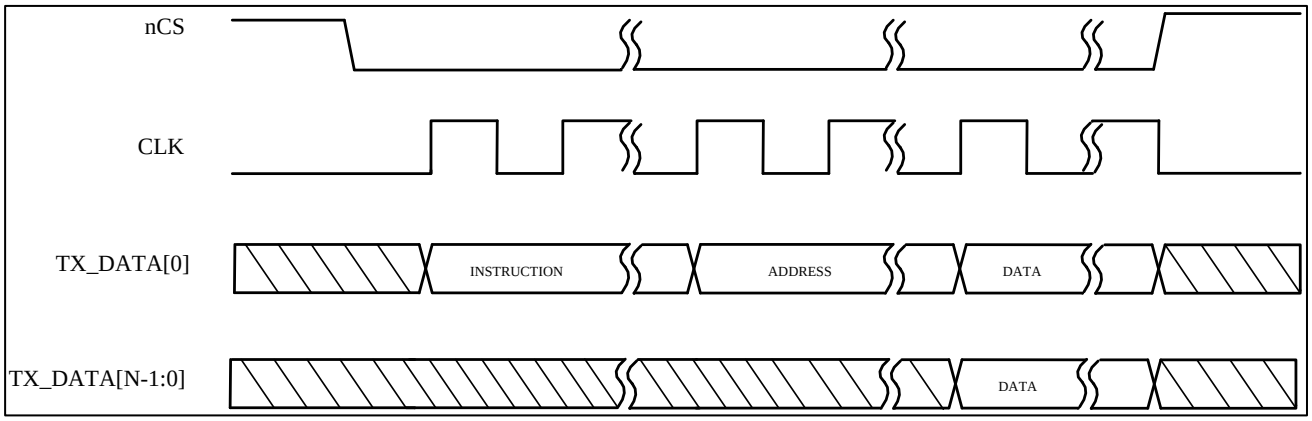
写操作可以分为 3 个阶段：指令阶段、地址阶段、数据阶段。

典型写操作时序



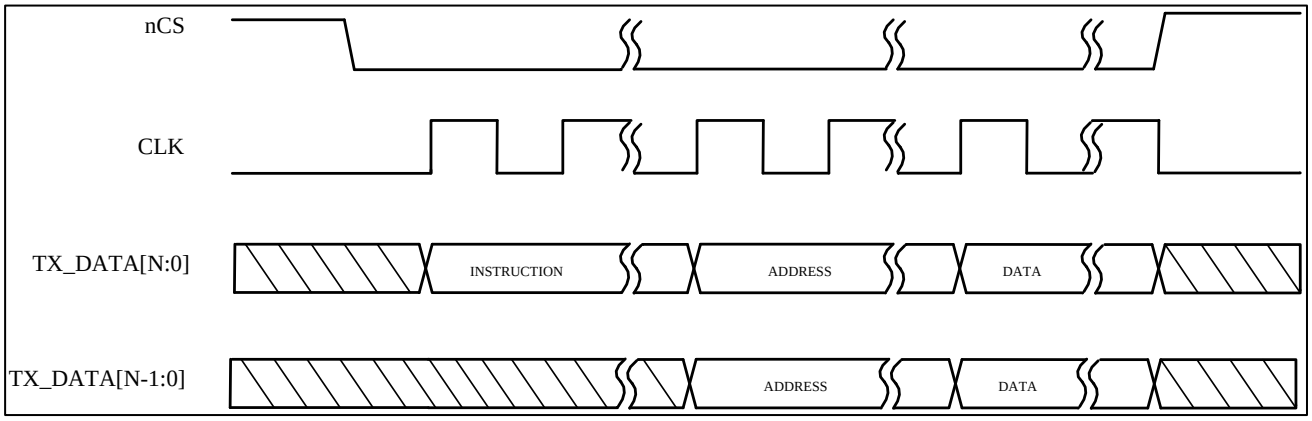
4 线模式下 N=3，对于 1 次写操作，指令和地址仅发送 1 次，然后是 DATx 中存储的数据帧，直到发送 FIFO 为空。

■指令和地址都以标准 SPI 格式发送时序



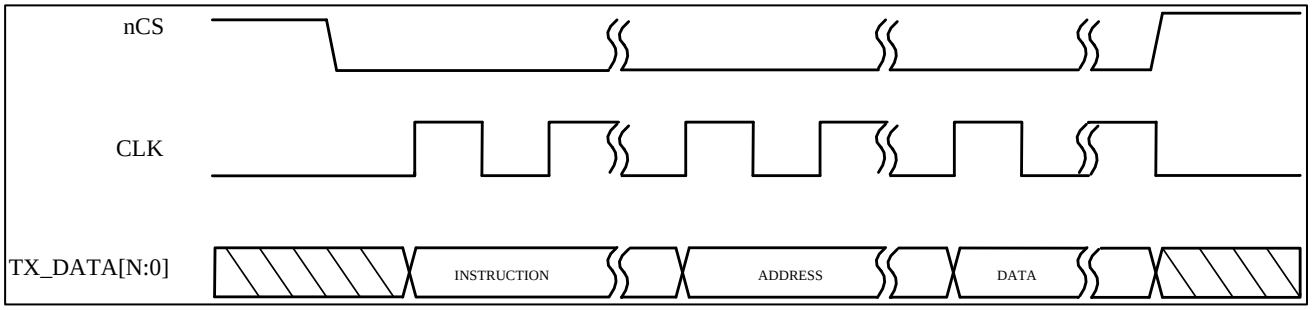
TRANS_TYPE（QSPI_ENH_CTRL0 [1:0]）须配置为 0。SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x02（4 线模式）时，N=3;SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x01（2 线模式）时，N=1。

■指令以标准 SPI 模式发送，地址以 CTRL0.SPI_FRF 制定模式发送时序



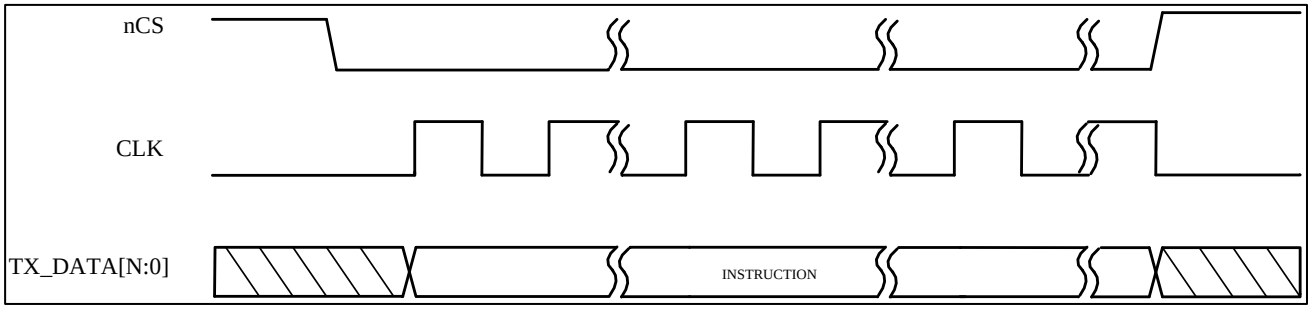
TRANS_TYPE（QSPI_ENH_CTRL0 [1:0]）须配置为 0x01。SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x02（4 线模式）时，N=3;SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x01（2 线模式）时，N=1。

■指令和地址以 CTRL0.SPI_FRF 制定模式发送时序



TRANS_TYPE（QSPI_ENH_CTRL0 [1:0]）须配置为 0x02。SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x02（4 线模式）时，N=3;SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x01（2 线模式）时，N=1。

■只有指令阶段的发送时序

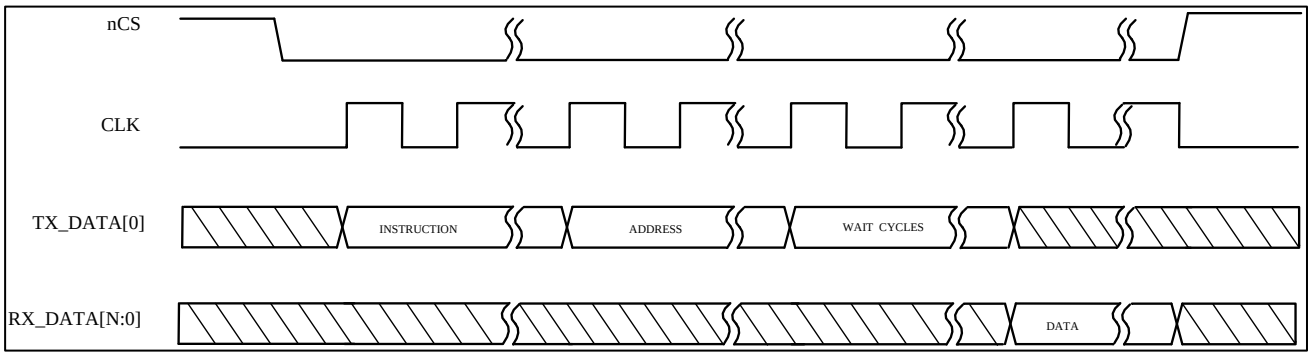


和 TRANS_TYPE（QSPI_ENH_CTRL0 [1:0]）配置无关。SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x02（4 线模式）时，N=3;SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x01（2 线模式）时，N=1。

25.5.3 QSPI 间接接收操作

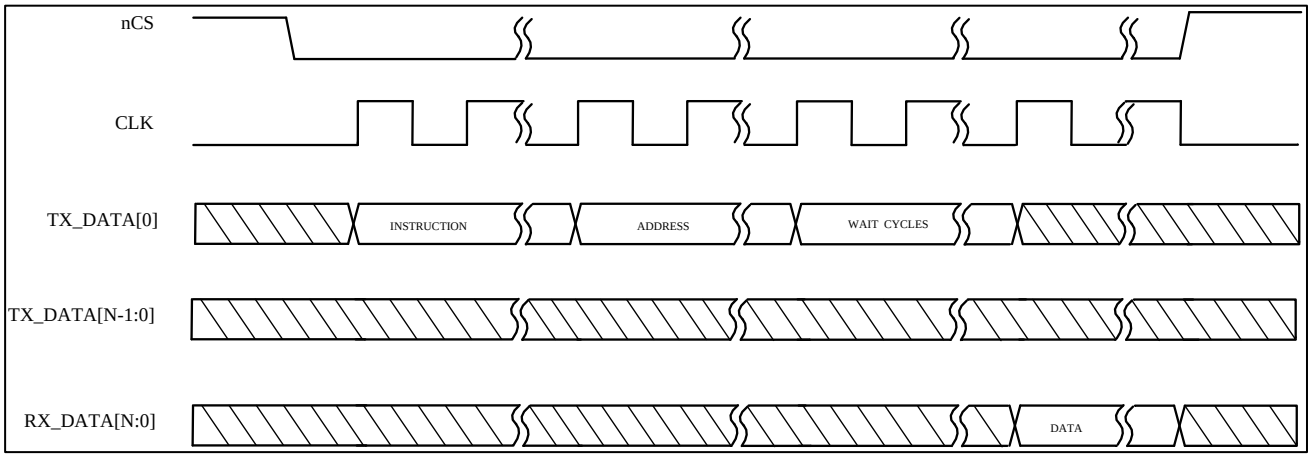
对于读操作，QSPI 发送 1 次指令和控制数据，直到收到数量等于 NDF（QSPI_CTRL1[15:0]）数量的数据，然后取消从机选择信号。读操作可以分为 4 个阶段：指令阶段、地址阶段、等待阶段、数据阶段。

典型读操作时序



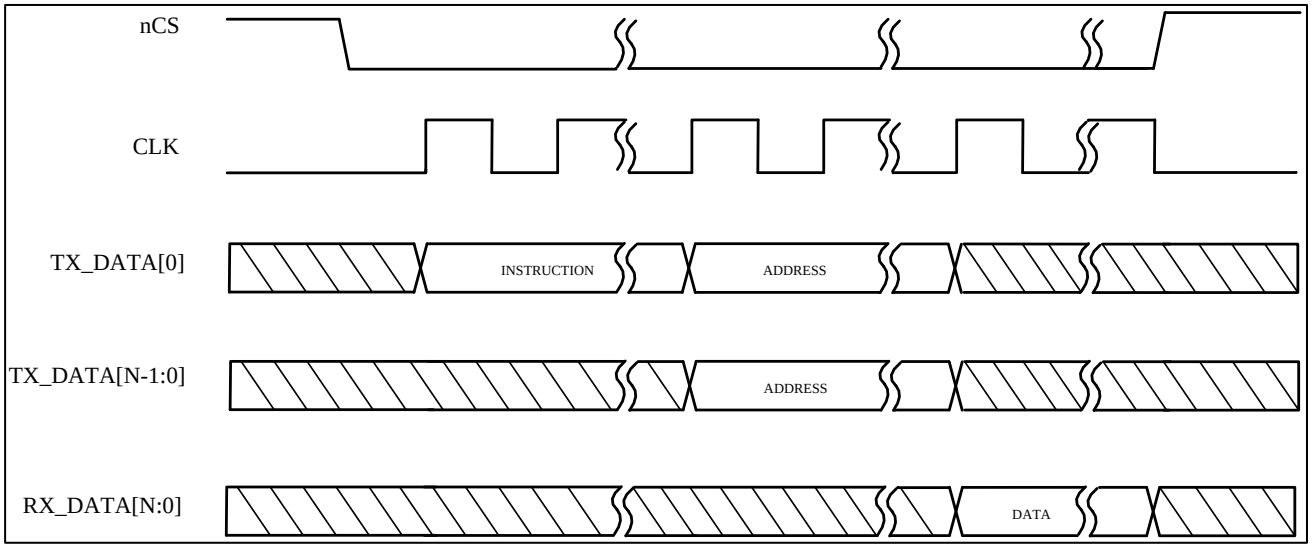
4 线模式下 N=3，每个读取命令数据将以 SPI_FRF（QSPI_CTRL0 [23：22]）配置的格式传输。配置为 0x2 为 4 线模式。

■ 地址和指令都以标准 SPI 格式接收时序



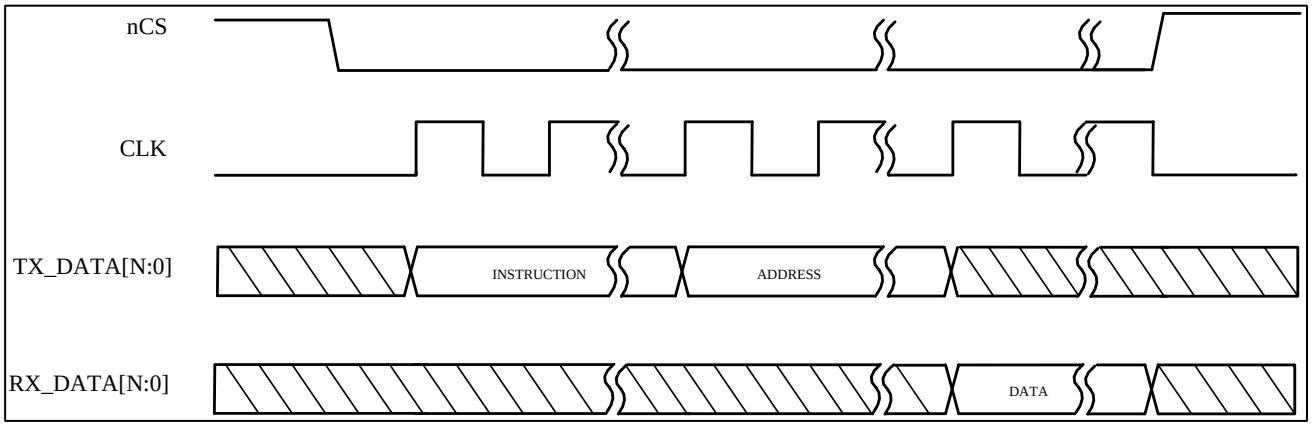
TRANS_TYPE（QSPI_ENH_CTRL0 [1:0]）应配置为 0x0，WAIT_CYCLES（QSPI_ENH_CTRL0 [15:11]）配置 WAIT 的周期。SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x02（4 线模式）时，N=3;SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x01（2 线模式）时，N=1。

■ 指令以标准 SPI 模式发送，地址以 QSPI_CTRL0.SPI_FRF 制定模式接收时序



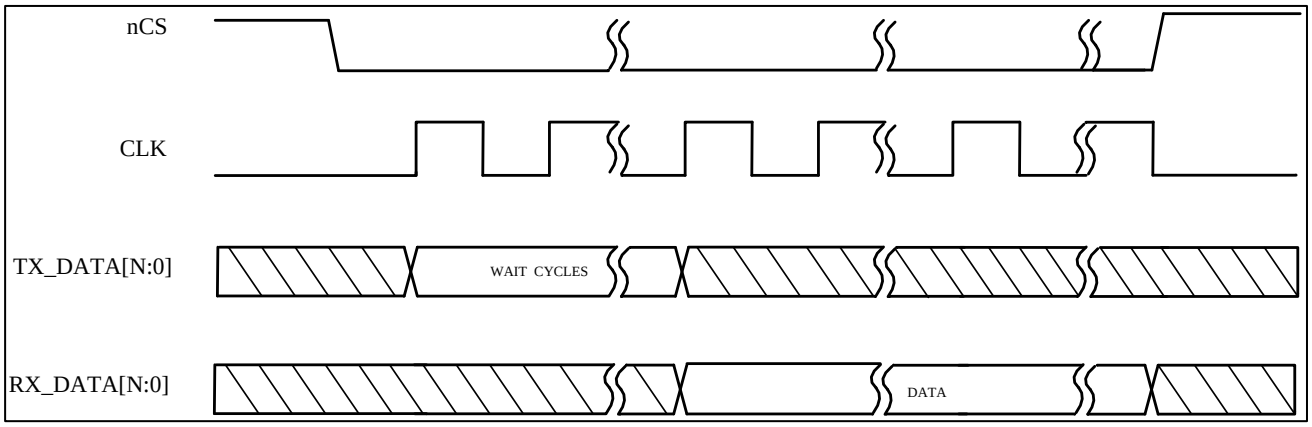
TRANS_TYPE（QSPI_ENH_CTRL0 [1:0]）应配置为 0x1，WAIT_CYCLES（QSPI_ENH_CTRL0 [15:11]）配置 WAIT 的周期。SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x02（4 线模式）时，N=3;SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x01（2 线模式）时，N=1。

■ 指令和地址以 QSPI_CTRL0.SPI_FRF 制定模式接收时序



TRANS_TYPE（QSPI_ENH_CTRL0 [1:0]）配置应为 0x2。SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x02（4 线模式）时，N=3;SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x01（2 线模式）时，N=1。

■ 只有 wait 阶段的接收时序



ADDR_LEN（QSPI_ENH_CTRL0 [5:2]）配置为 0，MI_LEN（QSPI_ENH_CTRL0 [9:8]）配置为 0，WAIT_CYCLES（QSPI_ENH_CTRL0 [15:11]）配置 WAIT 的周期。SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x02（4 线模式）时，N=3;SPI_FRF（QSPI_CTRL0 [23：22]）配置为 0x01（2 线模式）时，N=1。

25.6 QSPI 寄存器

QSPI 基地址：0xA0001000

25.6.1 QSPI 寄存器地址映像

表 25-1 QSPI 寄存器地址映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	QSPI_CTRL0	Reserved								SPI_FRF[1:0]		Reserved		CFS[3:0]				Reserved	SSTE	SRL	Reserved	TMOD[1:0]		SCPOL	SCPH	FRF[1:0]		Reserved	DFS[4:0]				
	Reset Value									1	0			0	0	0	0		1	0		0	1	0	0	0	0		0	0	1	1	1
004h	QSPI_CTRL1	Reserved																NDF[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	QSPI_EN	Reserved																															QEN
	Reset Value																																0
00Ch	QSPI_MW_CTRL	Reserved																													MHS_EN	MC_DIR	MWMOD
	Reset Value																														0	0	0
010h	QSPI_SLAVE_EN	Reserved																															SEN
	Reset Value																																0
014h	QSPI_BAUD	Reserved																CLK_DIV[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	QSPI_TXFT	Reserved											TXFT_ST[4:0]					Reserved											TXFT_TEI[4:0]				
	Reset Value												0	0	0	0	0												0	0	0	0	0
01Ch	QSPI_RXFT	Reserved																											RXFT_TFI[4:0]				
	Reset Value																												0	0	0	0	0
020h	QSPI_TXFN	Reserved																											TXFN[5:0]				
	Reset Value																												0	0	0	0	0
024h	QSPI_RXFN	Reserved																											RXFN[5:0]				
	Reset Value																												0	0	0	0	0
028h	QSPI_STS	Reserved																											TX_ERR	RXFF	RXFNE	TXFE	TXFNF	BUSY
	Reset Value																												0	0	0	1	1	0
02Ch	QSPI_IMASK	Reserved															XRXOIM	Reserved	RXFFIM	RXFOIM	RXFUIM	TXFOIM	TXFEIM											

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset Value																										1		1	1	1	1	1
030h	QSPI_ISTS	Reserved																									XRXOIS	Reserve	RXFFIS	RXFOIS	RXFUIS	TXFOIS	TXFEIS
	Reset Value																										0		0	0	0	0	0
034h	QSPI_RISTS	Reserved																									XRXORIS	Reserved	RXFFRIS	RXFORIS	RXFURIS	TXFORIS	TXFERIS
	Reset Value																										0		0	0	0	0	0
038h	QSPI_TXFOI_CLR	Reserved																															TXFOIC
	Reset Value																																0
03Ch	QSPI_RXFOI_CLR	Reserved																															RXFOIC
	Reset Value																																0
040h	QSPI_RXFUI_CLR	Reserved																															RXFUIC
	Reset Value																																0
044h	Reserved	Reserved																															
	Reset Value																																
048h	QSPI_ICLR	Reserved																															INTC
	Reset Value																																0
04Ch	QSPI_DMA_CTRL	Reserved																														TX_DMA_EN	RX_DMA_EN
	Reset Value																															0	0
050h	QSPI_DMATDL_CTRL	Reserved																									DMATDL[5:0]						
	Reset Value																										0	0	0	0	0	0	0
054h	QSPI_DMARDL_CTRL	Reserved																									DMARDL[5:0]						
	Reset Value																										0	0	0	0	0	0	0
058h	QSPI_IDCODE	IDCODE[31:0]																															
	Reset Value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
060h+004h*x(x=0～31)	QSPI_DATx	DATx[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0F0h	QSPI_RS_DELAY	Reserved															SES	Reserved									SDCN[7:0]						
	Reset Value																0										0	0	0	0	0	0	0
0F4h	QSPI_ENH_CTRLO	Reserved	CLK_STRETCH_EN	Reserved	XIP_MBL[1:0]		Reserved					XIP_CT_EN	XIP_INST_EN	XIP_DFS_HC	Reserved	INST_DDR_EN	SPI_DDR_EN	WAIT_CYCLESLES[4:0]					Reserved	INST_L[1:0]		MD_BIT_EN	Reserved	ADDR_LEN[3:0]				TRANS_TYPE[1:0]	
	Reset Value		0		0	0						0	0	0		1	0	0	0	0	0	0		0	0	0			0	0	0	0	0	0

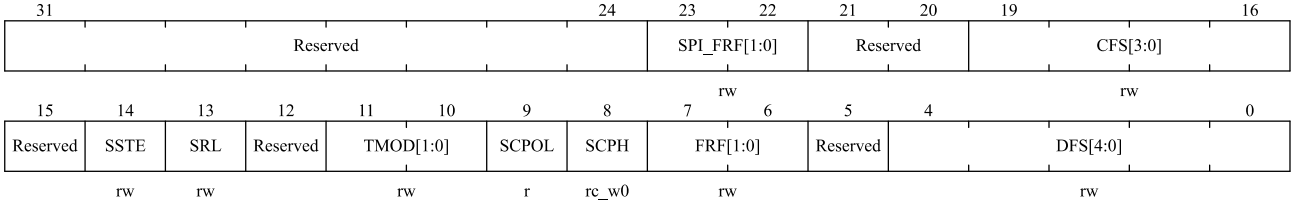
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0F8h	QSPI_DDR_TXDE	Reserved																								TXDE[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0
0FCh	XIP_MODE	Reserved																XIP_MD_BITS[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100h	XIP_INCR_TOC	Reserved																ITOC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
104h	XIP_WRAP_TOC	Reserved																WTOC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
108h	XIP_CTRL	Reserved				XIP_MBL[1:0]		Reserved		XIP_CT_EN	XIP_INST_EN	Reserved	INST_DDR_EN	DDR_EN	DFS_HC	WAIT_CYCLESLES[4:0]					MD_BITS_EN	Reserved	INST_L[1:0]		Reserved	ADDR_LEN[3:0]				TRANS_TYPE[1:0]		FRF[1:0]	
	Reset Value					0	0			0	0		0	0	0	0	0	0	0	0	0			0	0		0	0	0	0	0	0	0	0
10Ch	XIP_SLAVE_EN	Reserved																															SEN
	Reset Value																																0
110h	XIP_RXFOI_CLR	Reserved																															XRXFOIC
	Reset Value																																0
114h	XIP_TOUT	Reserved																								XTOUT[7:0]							
	Reset Value																									0	0	0	0	0	0	0	0

25.6.2 QSPI 控制寄存器 0（QSPI_CTRL0）

QSPI_CTRL0 寄存器控制串行数据传输，在 QSPI_EN 寄存器使能后不能写 QSPI_CTRL0 寄存器。

偏移地址：0x00

复位值：0x0080 4407



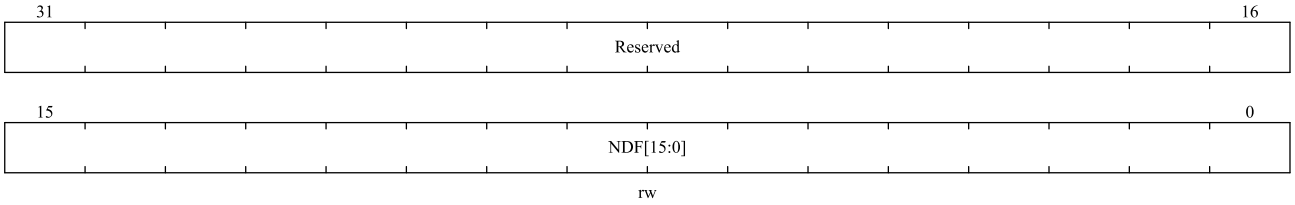
位域	名称	描述
31:24	Reserved	保留，硬件强制为 0。
23:22	SPI_FRF [1:0]	SPI Frame Format 帧格式 00：标准 SPI； 01：双线 SPI；

位域	名称	描述
		10：四线 SPI； 11：保留。
21:20	Reserved	保留，硬件强制为 0。
19:16	CFS[3:0]	Control Frame Size 选择 Microwire 帧格式的控制字长度 0000：1bit 控制字； 0001：2bit 控制字； 0010：3bit 控制字； 0011：4bit 控制字； …… 1110：15bit 控制字； 1111：16bit 控制字。
15	Reserved	保留，硬件强制为 0。
14	SSTE	Slave Select Toggle Enable 片选切换使能。 0：串行时钟 sclk 在整个传送期间片选连续为低； 1：串行时钟 sclk 在每一帧数据传输时片选为低。
13	SRL	Shift Register Loop 移位寄存器循环，用于测试，有效时将发送移位寄存器输出连接到接收移位寄存器输入 0：禁能； 1：使能。
12	Reserved	保留，硬件强制为 0。
11:10	TMOD[1:0]	Transfer Mode 传输模式 00：Tx and Rx； 01：Tx only； 10：Rx only； 11：EERPOM read。
9	SCPOL	Serial Clock Polarity 非活动串行时钟极性 0：低； 1：高。
8	SCPH	Serial Clock Phase 串行时钟相位 0：在串行时钟的第一个边沿捕获数据； 1：在从选择线激活后开始切换一个周期，并在串行时钟的第二个边沿捕获数据。
7:6	FRF[1:0]	Frame Format 帧格式 00：Motorola SPI 帧格式； 01：保留； 10：National Semiconductors Microwire 帧格式； 11：保留。
5	Reserved	保留，硬件强制为 0。
4:0	DFS[4:0]	Data Frame Size 数据帧长度，当数据帧小于 32bit 时自动右对齐 0x0/0x01/0x02：保留 0x03：4bit 0x04：5bit

位域	名称	描述
		0x05：6bit …… 0x1D：30bit 0x1E：31bit 0x1F：32bit

25.6.3 QSPI 控制寄存器 1（QSPI_CTRL1）

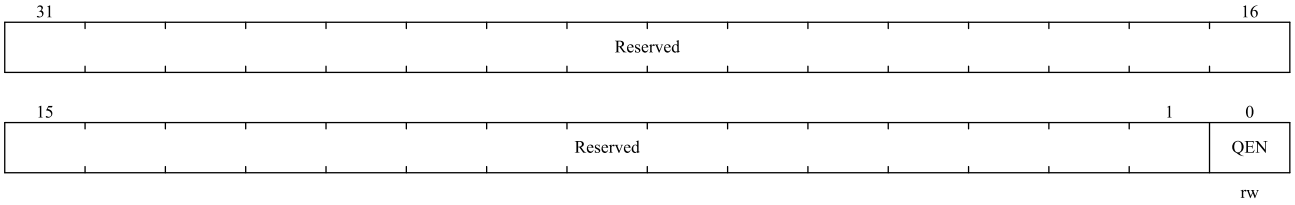
偏移地址：0x04
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	NDF[15:0]	Number of Data Frames 数据帧数量。当 TMOD=10 或 11 时，该寄存器配置连续接收的数据帧数量

25.6.4 QSPI 使能寄存器（QSPI_EN）

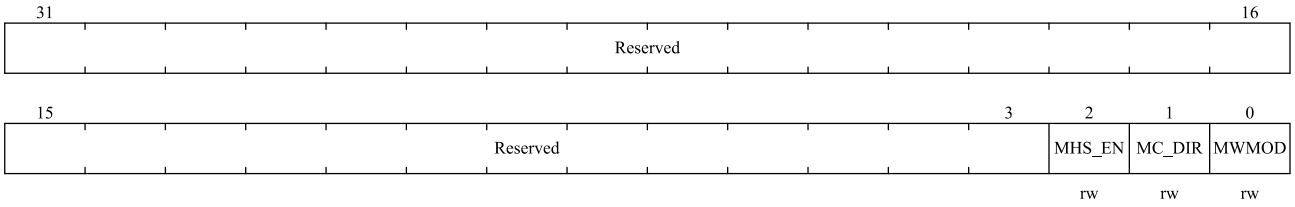
偏移地址：0x08
复位值：0x0000 0000



位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	QEN	QSPI Enable 使能 QSPI 0：禁能，当禁能 QSPI 时，所有串行传输立即停止； 1：使能 QSPI。

25.6.5 QSPI MW 控制寄存器（QSPI_MW_CTRL）

Microwire 串行协议控制寄存器。
偏移地址：0x0C
复位值：0x0000 0000

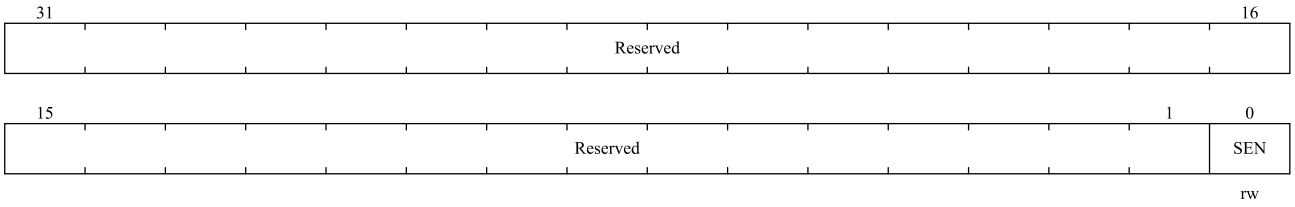


位域	名称	描述
31:3	Reserved	保留，硬件强制为 0。
2	MHS_EN	Microwire Handshaking Enable 使能 Microwire 握手 0：握手禁能； 1：握手使能，传输最后 1 个数据或控制位后，在清除 QSPI_STS 寄存器中的 BUSY 状态之前，从目标器件检查就绪状态。
1	MC_DIR	Direction of Data when Microwire Control 控制 Microwire 数据方向 0：Rx 1：Tx
0	MWMOD	Microwire Transfer Mode 传输模式 0：非顺序传输，必须为每个发送或接收数据字块指定控制字； 1：顺序传输，仅需一个控制字来发送或接收数据字节。

25.6.6 QSPI 从设备使能寄存器（QSPI_SLAVE_EN）

使能 QSPI_EN 寄存器，QSPI_SLAVE_EN 寄存器才会被启用，用于外部从设备选择使能片选。

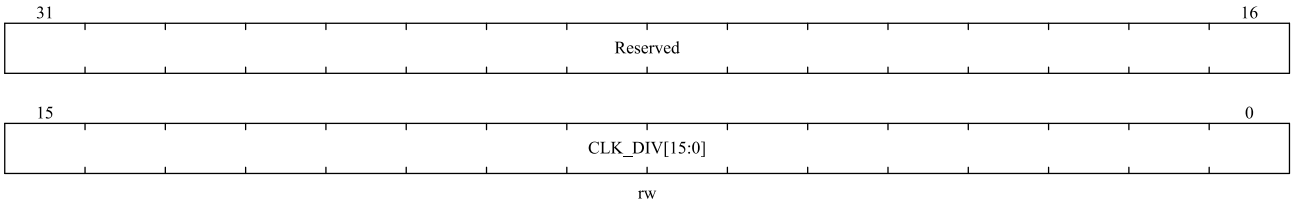
偏移地址：0x10
复位值：0x0000 0000



位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	SEN	Slave Enable 外部从设备片选使能 0：禁能 1：使能

25.6.7 QSPI 波特率选择寄存器（QSPI_BAUD）

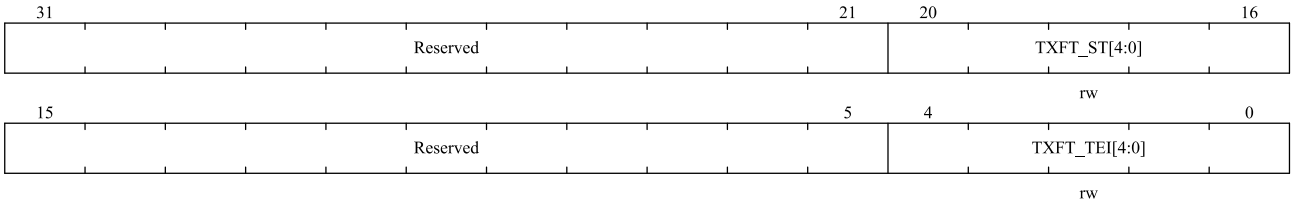
偏移地址：0x14
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	CLK_DIV[15:0]	Clock Divider 时钟分频 该字段 LSB 始终配置为 0，不受写操作影响，这样可以确保在该寄存器中保留偶数值。如果[15:0]=0x0000，则禁能串行输出时钟。串行输出时钟频率 sclk_out 计算公式为 Fsclk_out = Fqspi_hclk/CLK_DIV。qspi_hclk 即为 HCLK，CLK_DIV 取值范围为 2～65534 之间的偶数。

25.6.8 QSPI 发送缓存阈值寄存器（QSPI_TXFT）

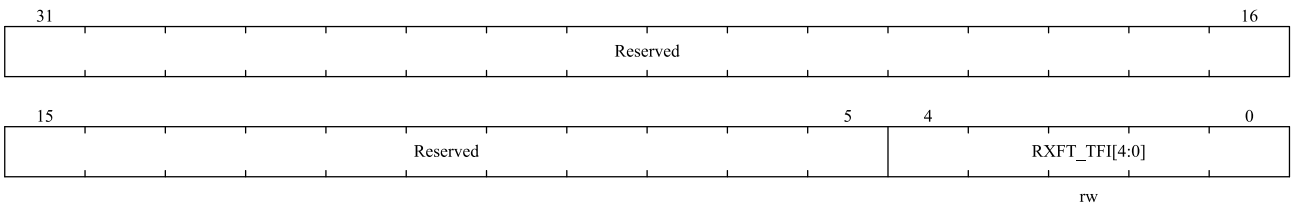
偏移地址：0x18
复位值：0x0000 0000



位域	名称	描述
31:21	保留	必须保持复位值。
20:16	TXFT_ST[4:0]	Transmit FIFO Threshold to Start to Transfer.Tx 传输阈值，达到该值后，开始 Tx 传输
15:5	Reserved	保留，硬件强制为 0。
4:0	TXFT_TEI[4:0]	Transmit FIFO Threshold to Trigger Empty Interrupt. Tx 传输阈值，当发送 FIFO 数量小于该值，触发空中断

25.6.9 QSPI 接收缓存阈值寄存器（QSPI_RXFT）

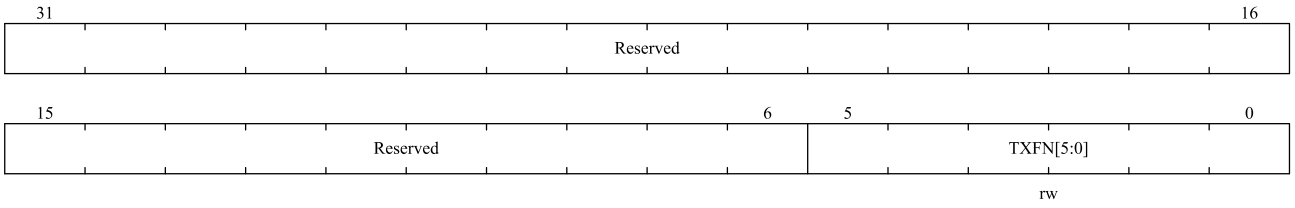
偏移地址：0x1C
复位值：0x0000 0000



位域	名称	描述
31:5	Reserved	保留，硬件强制为 0。
4:0	RXFT_TFI [4:0]	Receive FIFO Threshold to Trigger Full Interrupt. Rx 传输阈值，当接收 FIFO 数量大于该值加 1 时，触发满中断

25.6.10 QSPI 发送缓存数据量寄存器（QSPI_TXFN）

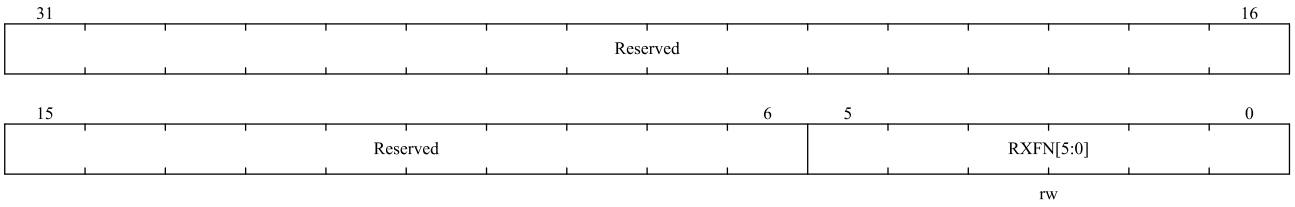
偏移地址：0x20
复位值：0x0000 0000



位域	名称	描述
31:6	Reserved	保留，硬件强制为 0。
5:0	TXFN[5:0]	Number of Transmit FIFO. Tx Fifo 数据量

25.6.11 QSPI 接收缓存数据量寄存器（QSPI_RXFN）

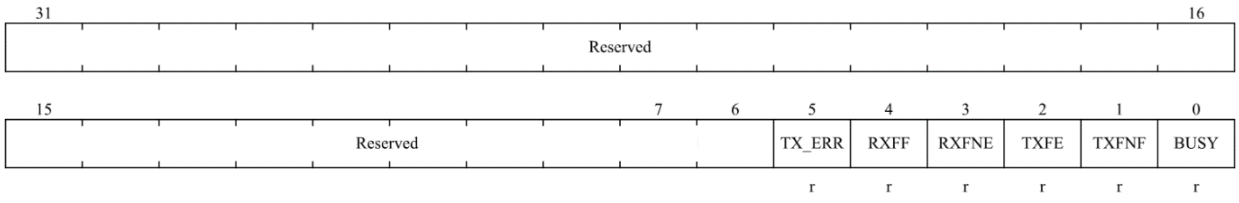
偏移地址：0x24
复位值：0x0000 0000



位域	名称	描述
31:6	Reserved	保留，硬件强制为 0。
5:0	RXFN [5:0]	Number of Receive FIFO. Rx Fifo 数据量

25.6.12 QSPI 状态寄存器（QSPI_STS）

偏移地址：0x28
复位值：0x0000 0006

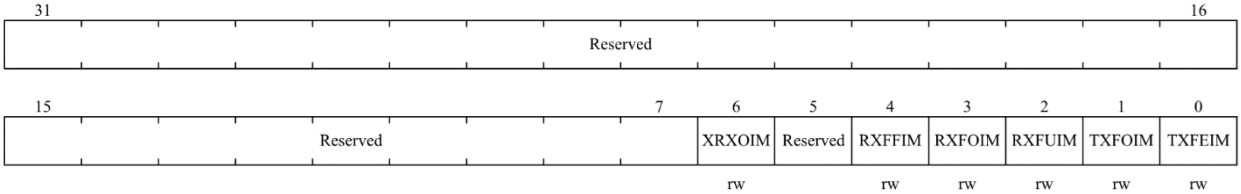


位域	名称	描述
31:6	Reserved	保留。
5	TX_ERR	Transmit Error 传输错误 0：无传输错误 1：发生传输错误 发生传输错误，该位置位，读取后清零。
4	RXFF	Receive FIFO Full. Rx Fifo 满 0：无 Rx Fifo 满 1：发生 Rx Fifo 满 Rx 时 Fifo 为满，该位置位，读取后清零。
3	RXFNE	Receive FIFO Not Empty. Rx Fifo 非空状态 0：Rx Fifo 空状态 1：发生 Rx Fifo 非空状态 Rx 时 Fifo 非空，该位置位，读取后清零。
2	TXFE	Transmit FIFO Empty. Tx Fifo 空状态 0：Tx Fifo 有数据状态 1：发生 Tx Fifo 空状态 Tx 时 Fifo 为空，该位置位，读取后清零。
1	TXFNF	Transmit FIFO Not Full. Tx Fifo 非满状态 0：Tx Fifo 满状态 1：发生 Tx Fifo 非满状态 Tx 时 Fifo 为非满，该位置位，读取后清零。
0	BUSY	Transfer Busy Flag. 传输忙标志 0：空闲或禁能 1：忙碌 置位表示正在进行传输。

25.6.13 QSPI 中断屏蔽寄存器（QSPI_IMASK）

偏移地址：0x2C

复位值：0x0000 007F

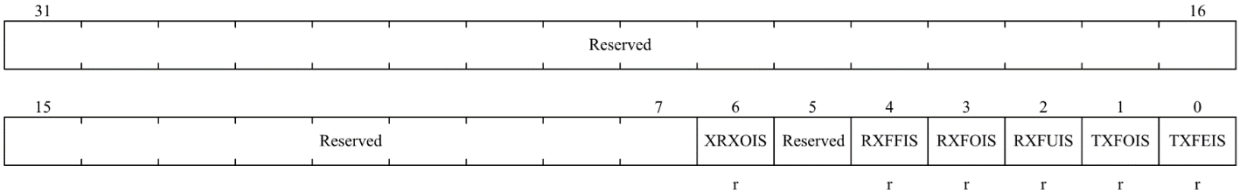


位域	名称	描述
31:7	Reserved	保留，硬件强制为 0。
6	XRXOIM	XIP Receive FIFO Overflow Interrupt Mask. XIP 接收寄存器上溢中断屏蔽 0：屏蔽 1：不屏蔽
5	Reserved	保留
4	RXFFIM	Receive FIFO Full Interrupt Mask. Rx Fifo 满中断屏蔽

位域	名称	描述
		0：屏蔽 1：不屏蔽
3	RXFOIM	Receive FIFO Overflow Interrupt Mask. Rx Fifo 上溢中断屏蔽 0：屏蔽 1：不屏蔽
2	RXFUIM	Receive FIFO Underflow Interrupt Mask. Rx Fifo 下溢中断屏蔽 0：屏蔽 1：不屏蔽
1	TXFOIM	Transmit FIFO Overflow Interrupt Mask. Tx Fifo 上溢中断屏蔽 0：屏蔽 1：不屏蔽
0	TXFEIM	Transmit FIFO Empty Interrupt Mask. Tx Fifo 空中断屏蔽 0：屏蔽 1：不屏蔽

25.6.14 QSPI 中断状态寄存器（QSPI_ISTS）

偏移地址：0x30
复位值：0x0000 0000



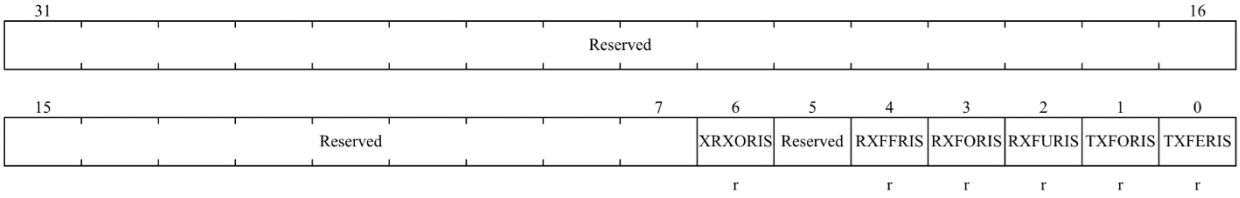
位域	名称	描述
31:7	Reserved	保留，硬件强制为 0。
6	XRXOIS	XIP Receive FIFO Overflow Interrupt Status. XIP 接收寄存器上溢状态 0：失效 1：有效
5	Reserved	保留
4	RXFFIS	Receive FIFO Full Interrupt Status. Rx Fifo 满状态 0：失效 1：有效
3	RXFOIS	Receive FIFO Overflow Interrupt Status. Rx Fifo 上溢状态 0：失效 1：有效
2	RXFUIS	Receive FIFO Underflow Interrupt Status. Rx Fifo 下溢状态 0：失效 1：有效
1	TXFOIS	Transmit FIFO Overflow Interrupt Status. Tx Fifo 上溢状态 0：失效 1：有效

位域	名称	描述
0	TXFEIS	Transmit FIFO Empty Interrupt Status. Tx Fifo 空状态 0：失效 1：有效

25.6.15 QSPI 原始中断状态寄存器（QSPI_RISTS）

偏移地址：0x34

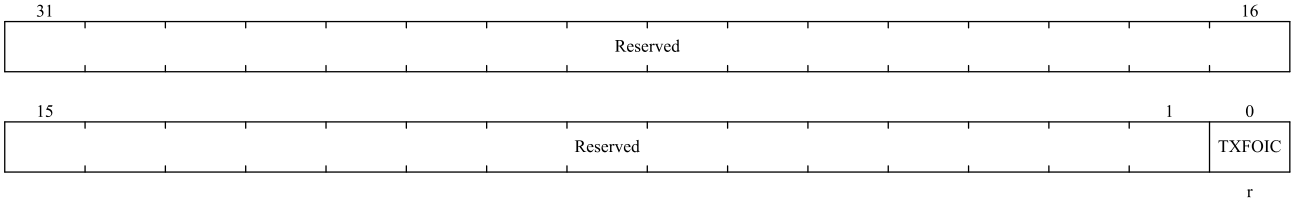
复位值：0x0000 0000



位域	名称	描述
31:7	Reserved	保留，硬件强制为 0。
6	XRXORIS	XIP Receive FIFO Overflow Raw Interrupt Status. XIP 接收寄存器上溢状态（在中断屏蔽前的状态） 0：失效 1：有效
5	Reserved	保留
4	RXFFRIS	Receive FIFO Full Raw Interrupt Status. Rx Fifo 满状态（在中断屏蔽前的状态） 0：失效 1：有效
3	RXFORIS	Receive FIFO Overflow Raw Interrupt Status. Rx Fifo 上溢状态（在中断屏蔽前的状态） 0：失效 1：有效
2	RXFURIS	Receive FIFO Underflow Raw Interrupt Status. Rx Fifo 下溢状态（在中断屏蔽前的状态） 0：失效 1：有效
1	TXFORIS	Transmit FIFO Overflow Raw Interrupt Status. Tx Fifo 上溢状态（在中断屏蔽前的状态） 0：失效 1：有效
0	TXFERIS	Transmit FIFO Empty Raw Interrupt Status. Tx Fifo 空状态（在中断屏蔽前的状态） 0：失效 1：有效

25.6.16 QSPI 发送缓存上溢中断清除寄存器（QSPI_TXFOI_CLR）

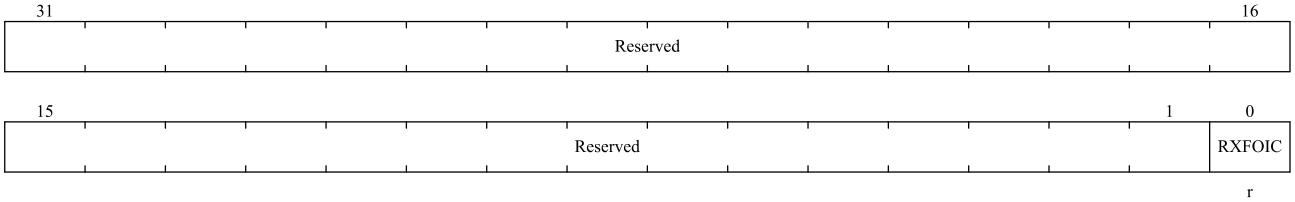
偏移地址：0x38
复位值：0x0000 0000



位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	TXFOIC	Clear Transmit FIFO Overflow Interrupt.读该寄存器后，清除 Tx Fifo 上溢中断状态

25.6.17 QSPI 接收缓存上溢中断清除寄存器（QSPI_RXFOI_CLR）

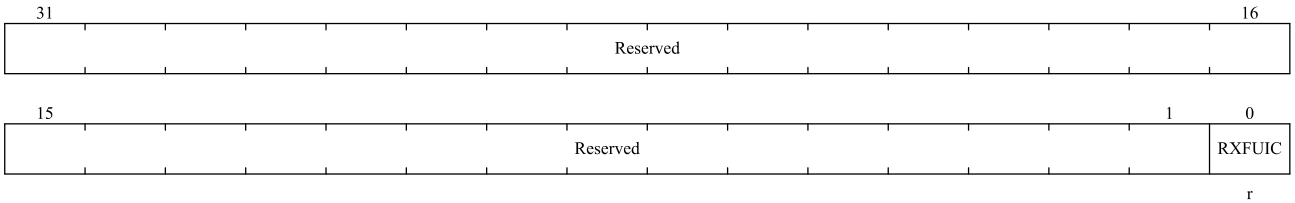
偏移地址：0x3C
复位值：0x0000 0000



位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	RXFOIC	Clear Receive FIFO Overflow Interrupt.读该寄存器后，清除 Rx Fifo 上溢中断状态

25.6.18 QSPI 接收缓存下溢中断清除寄存器（QSPI_RXFUI_CLR）

偏移地址：0x40
复位值：0x0000 0000



位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	RXFUIC	Clear Receive FIFO Underflow Interrupt.读该寄存器后，清除 Rx Fifo 下溢

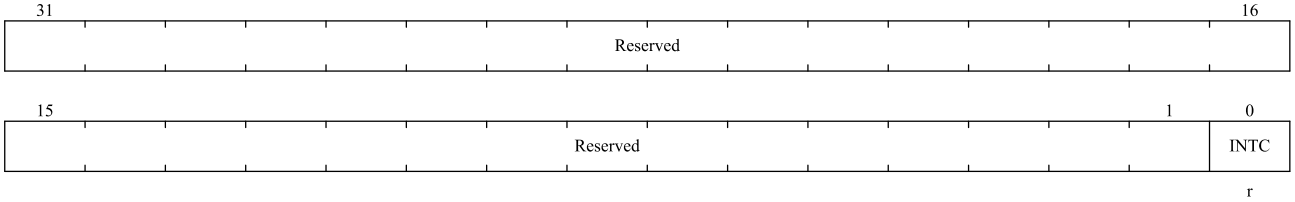
位域	名称	描述
		中断状态

25.6.19 保留寄存器

偏移地址：0x44
复位值：0x0000 0000
保留

25.6.20 QSPI 中断清除寄存器（QSPI_ICLR）

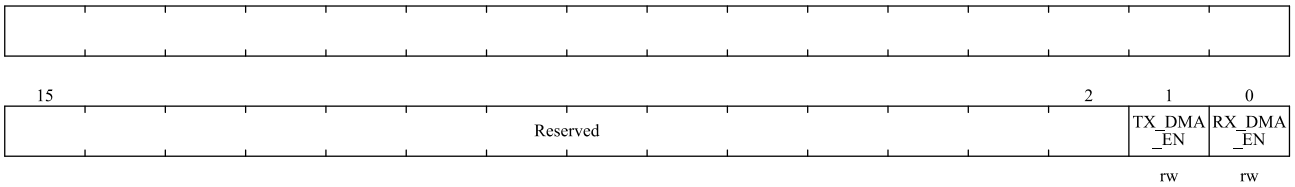
偏移地址：0x48
复位值：0x0000 0000



位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	INTC	Clear Interrupts.读该寄存器后，清除发送缓存上溢中断、接收缓存上溢中断、接收缓存下溢中断状态。

25.6.21 QSPI DMA 控制寄存器（QSPI_DMA_CTRL）

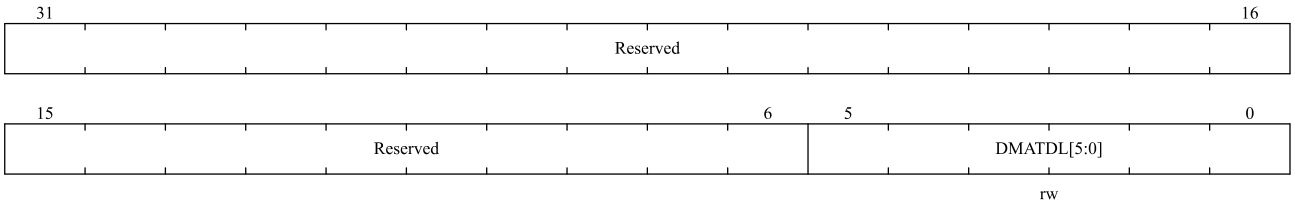
偏移地址：0x4C
复位值：0x0000 0000



位域	名称	描述
31:2	Reserved	保留，硬件强制为 0。
1	TX_DMA_EN	Transmit DMA Enable. Tx DMA功能使能 0：禁能 1：使能
0	RX_DMA_EN	Receive DMA Enable. Rx DMA功能使能 0：禁能 1：使能

25.6.22 QSPI DMA 发送水位控制寄存器（QSPI_DMATL_CTRL）

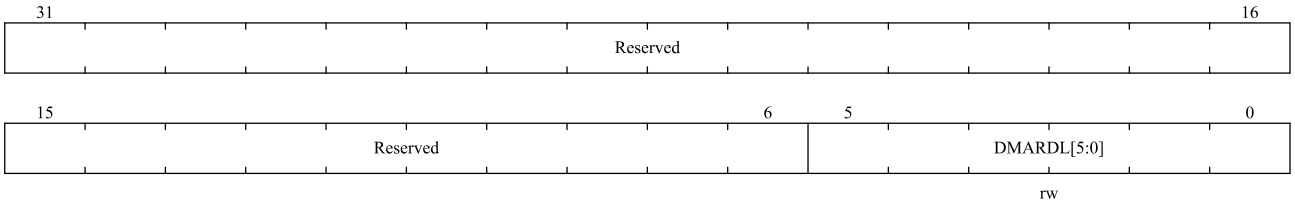
偏移地址：0x50
复位值：0x0000 0000



位域	名称	描述
31:6	Reserved	保留，硬件强制为 0。
5:0	DMATDL[5:0]	DMA Transmit Data Level.发出 DMA 发送请求的水位控制

25.6.23 QSPI DMA 接收水位控制寄存器（QSPI_DMARDL_CTRL）

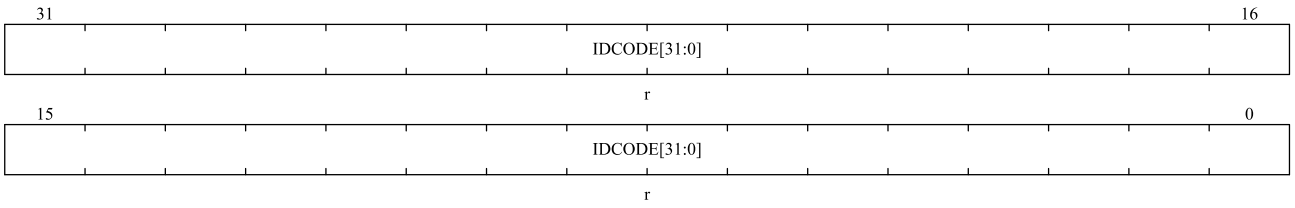
偏移地址：0x54
复位值：0x0000 0000



位域	名称	描述
31:6	Reserved	保留，硬件强制为 0。
5:0	DMARDL[5:0]	DMA Receive Data Level.发出 DMA 接收请求的水位控制

25.6.24 QSPI 校验码寄存器（QSPI_IDCODE）

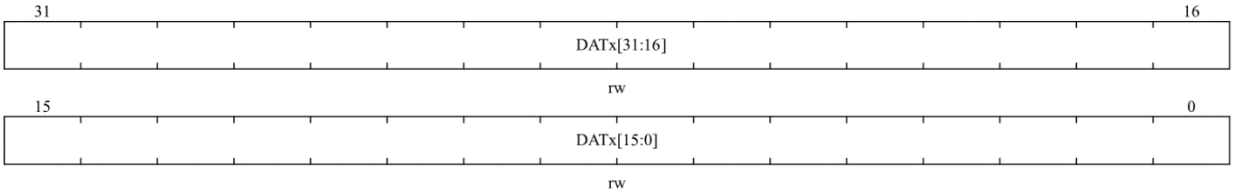
偏移地址：0x58
复位值：0xFFFF FFFF



位域	名称	描述
31:0	IDCODE[31:0]	Identification Code 外设校验码

25.6.25 QSPI 数据寄存器（QSPI_DATx）

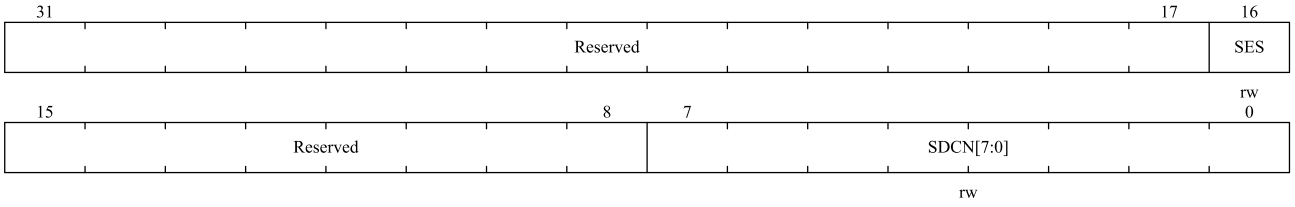
偏移地址：0x60+0x04 * x （x=0~31）
复位值：0x0000 0000



位域	名称	描述
31:0	DATx[31:0]	Data Register 数据寄存器 Rx 状态下为接收缓冲区，读取数据自动右对齐；Tx 状态下为发送缓冲区，写数据必须对数据进行右对齐。 *注意共 32 个寄存器地址为 0x60+(0:31)*0x04*

25.6.26 QSPI 接收采样延迟寄存器（QSPI_RS_DELAY）

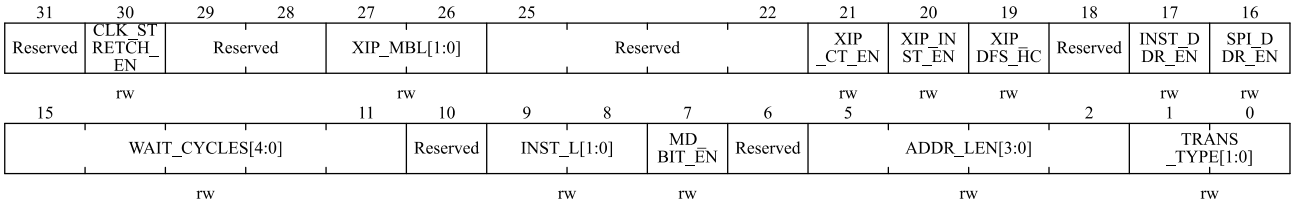
偏移地址：0xF0
复位值：0x0000 0000



位域	名称	描述
31:17	Reserved	保留，硬件强制为 0。
16	SES	Sampling Edge Select of Receive Data 采样沿选择位 0：上升沿采样 1：下降沿采样
15:8	Reserved	保留，硬件强制为 0。
7:0	SDCN[7:0]	Sample Delay Cycle Number 延迟采样的 ssi_clk 周期数量。 *注意：当超过 0x6 时，将采用 0 延迟采样。*

25.6.27 QSPI 增强型 SPI 模式控制寄存器（QSPI_ENH_CTRL0）

此寄存器用于增强型 SPI 操作模式下控制串行数据传输。只有当 SPI_FRF（在 CTRLR0 中）设置为 01 或 10 时，配置此寄存器才有效。在 QSPI_EN 寄存器使能后，无法写入此寄存器。
偏移地址：0xF4
复位值：0x0000 0200

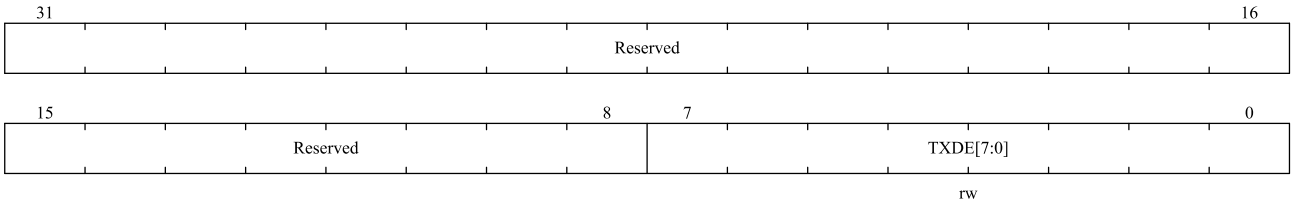


位域	名称	描述
31	Reserved	保留，硬件强制为 0。
30	CLK_STRETCH_EN	Clock Stretch Enable in SPI transfers.在写入的情况下，如果 FIFO 为空，则 QSPI 将拉伸时钟，直到 FIFO 有足够的数据。 在读取的情况下，如果 FIFO 为满，则 QSPI 将停止时钟，直到 FIFO 有足够的空间。
29:28	Reserved	保留，硬件强制为 0。
27:26	XIP_MBL [1:0]	XIP Mode bits length. XIP 模式下位宽 00：2 01：4 10：8 11：16
25:22	Reserved	保留，硬件强制为 0。
21	XIP_CT_EN	Enable Continuous Transfer in XIP mode. XIP 模式下启用连续传输
20	XIP_INST_EN	XIP Instruction Enable. XIP 指令使能。使能后，XIP 传输也将具有指令阶段。将根据 AHB 传输类型从 XIP_INCR_TOC 或 XIP_WRAP_TOC 寄存器中选择指令操作码
19	XIP_DFS_HC	Fix DFS for XIP transfers.使能后，XIP 数据帧长度固定为 QSPI_CTRL0.DFS 的配置值 0：数据帧长度和数量取决于 HSIZE and HBURST 信号 1：XIP 数据帧长度固定为 QSPI_CTRL0.DFS 的配置值
18	Reserved	保留，硬件强制为 0。
17	INST_DDR_EN	XIP instruction enable bit. 使能指令双速率传输
16	SPI_DDR_EN	SPI DDR Enable bit. DDR 模式使能，将启用 SPI 的 2/4 线模式下所有的双速率传输。
15:11	WAIT_CYCLESLES [4:0]	Wait Cycles in Dual/Quad mode between control frames transmit and data reception.在控制帧发送和数据接收之间的等待周期，以 QSPI 时钟为周期。
10	Reserved	保留，硬件强制为 0。
9:8	INST_L [1:0]	Dual/Quad mode instruction length in bits. 2/4 线模式指令长度 00：0 01：4bit 10：8 bit 11：16 bit
7	MD_BIT_EN	Mode bits enable in XIP mode.使能后，XIP 模式将在地址阶段后插入模式位。这些位在寄存器 XIP_MODE 中设置。模式位长度始终为 8bit。

位域	名称	描述
6	Reserved	保留，硬件强制为 0。
5:2	ADDR_LEN [3:0]	Length of Address to transmit. 要传输的地址长度 0x0：无地址 0x1：4bit 0x2：8bit 0x3：12bit …… 0xE：56bit 0xF：60bit
1:0	TRANS_TYPE [1:0]	Address and instruction transfer format.地址和指令传输格式 00：标准 SPI 模式 01：指令以标准 SPI 模式发送，地址以 QSPI_CTRL0.SPI_FRF 制定模式发送 10：指令和地址以 QSPI_CTRL0.SPI_FRF 制定模式发送 11：保留

25.6.28 QSPI 发送驱动边沿寄存器（QSPI_DDR_TXDE）

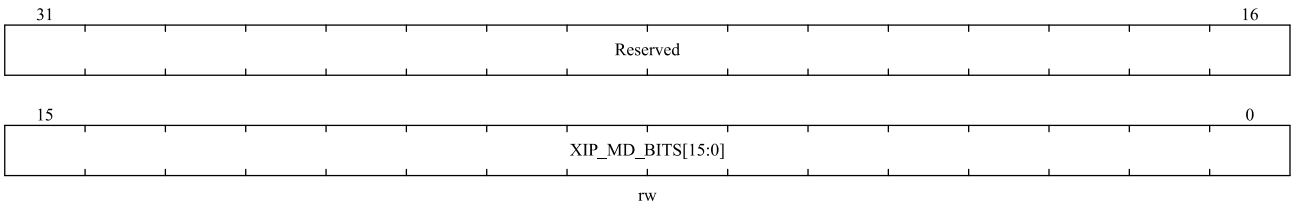
偏移地址：0xF8
复位值：0x0000 0000



位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7:0	TXDE[7:0]	Transmit Drive Edge.决定发送数据的驱动边沿，该寄存器的最大值=QSPI_BAUD/2 - 1。

25.6.29 QSPI XIP 模式寄存器（QSPI_XIP_MODE）

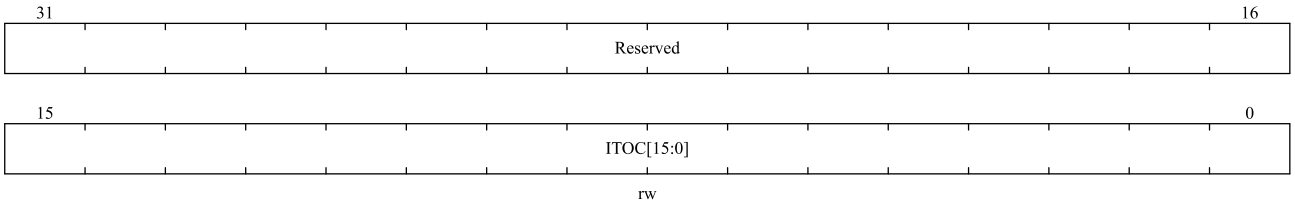
偏移地址：0xFC
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	XIP_MD_BITS [15:0]	XIP mode bits to be sent after address phase of XIP transfer.在 XIP 传输的地址阶段后发送的 XIP 模式位。

25.6.30 QSPI XIP INCR 传输操作码寄存器（QSPI_XIP_INCR_TOC）

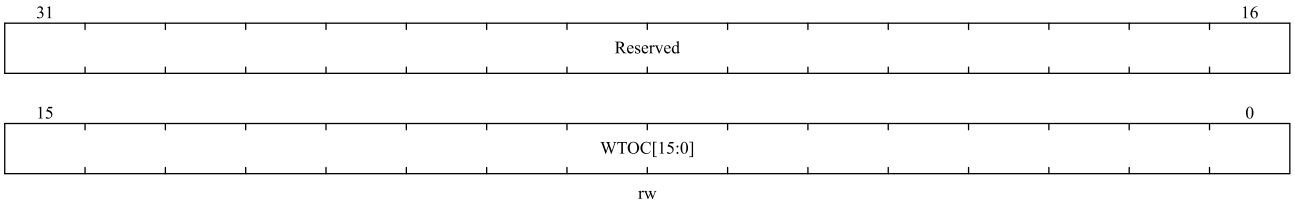
偏移地址：0x100
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	ITOC[15:0]	XIP INCR transfer OpCode. XIP_INCR传输操作码 当 SPI_CTRL0.XIP_INS_EN=1，QSPI 发送 XIP INCR 类传输指令码。指令阶段要发送的位数由 SPI_CTRL0. INST_L 字段确定。

25.6.31 QSPI XIP WRAP 传输操作码寄存器（QSPI_XIP_WRAP_TOC）

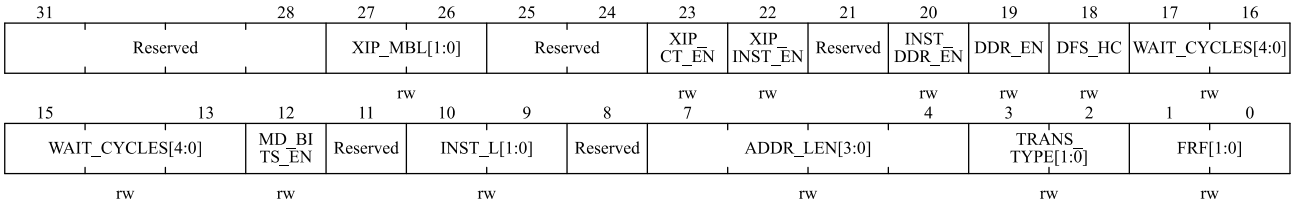
偏移地址：0x104
复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	WTOC[15:0]	XIP WRAP transfer OpCode. XIP_WRAP传输操作码 当 QSPI_ENH_CTRL0.XIP_INS_EN=1，QSPI 发送 XIP WRAP 类传输指令码。指令阶段要发送的位数由 QSPI_ENH_CTRL0. INST_L 字段确定。

25.6.32 QSPI XIP 控制寄存器（QSPI_XIP_CTRL）

偏移地址：0x108
复位值：0x0000 0200



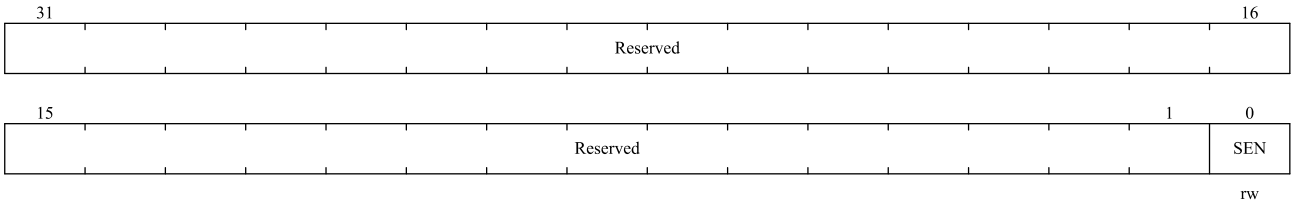
位域	名称	描述
31:28	Reserved	保留，硬件强制为 0。
27:26	XIP_MBL [1:0]	XIP Mode bits length. XIP 模式下位宽 00：2 01：4 10：8 11：16
25:24	Reserved	保留，硬件强制为 0。
23	XIP_CT_EN	Enable continuous transfer in XIP mode. XIP 模式下启用连续传输。
22	XIP_INST_EN	XIP instruction enable. XIP 指令使能。使能后，XIP 传输也将具有指令阶段。将根据 AHB 传输类型从 XIP_INCR_TOC 或 XIP_WRAP_TOC 寄存器中选择指令操作码
21	Reserved	保留，硬件强制为 0。
20	INST_DDR_EN	XIP instruction enable bit. 使能指令双速率传输
19	DDR_EN	SPI DDR Enable bit. DDR 模式使能，将启用 SPI 的 2/4 线模式下所有的双速率传输。
18	DFS_HC	Fix DFS for XIP transfers.使能后，XIP 数据帧长度固定为 QSPI_CTRL0.DFS 的配置值 0：数据帧长度和数量取决于 HSIZE and HBURST 信号 1：XIP 数据帧长度固定为 QSPI_CTRL0.DFS 的配置值
17:13	WAIT_CYCLESLES [4:0]	Wait Cycles in Dual/Quad mode between control frames transmit and data reception.在控制帧发送和数据接收之间的等待周期，以 QSPI 时钟为周期。
12	MD_BITS_EN	Mode bits enable in XIP mode.使能后，XIP 模式将在地址阶段后插入模式位。这些位在寄存器 XIP_MODE 中设置。模式位长度始终为 8bit。
11	Reserved	保留，硬件强制为 0。
10:9	INST_L [1:0]	Dual/Quad mode instruction length in bits. 2/4 线模式指令长度 00：0 01：4bit 10：8 bit 11：16 bit
8	Reserved	保留，硬件强制为 0。
7:4	ADDR_LEN[3:0]	Length of Address to transmit. 要传输的地址长度 0x0：无地址 0x1：4bit

位域	名称	描述
		0x2：8bit 0x3：12bit …… 0xE：56bit 0xF：60bit
3:2	TRANS_TYPE [1:0]	Address and instruction transfer format.地址和指令传输格式 00：标准 SPI 模式 01：指令以标准 SPI 模式发送，地址以 QSPI_XIP_CTRL. FRF 制定模式发送 10：指令和地址以 QSPI_XIP_CTRL. FRF 制定模式发送 11：保留
1:0	FRF[1:0]	帧格式选择 00：保留 01：2 线 10：4 线 11：保留

25.6.33 QSPI XIP 从设备使能寄存器（QSPI_XIP_SLAVE_EN）

使能 QSPI_EN 寄存器，QSPI_XIP_SLAVE_EN 寄存器才会被启用，用于外部从设备选择使能片选。

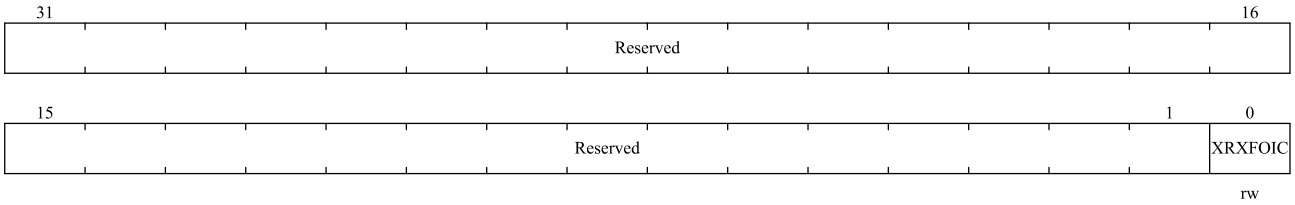
偏移地址：0x10C
复位值：0x0000 0000



位域	名称	描述
31:31	Reserved	保留，硬件强制为 0。
0	SEN	Slave Enable 外部从设备片选使能 0：禁能 1：使能

25.6.34 QSPI XIP 接收缓存上溢中断清除寄存器（QSPI_XIP_RXFOI_CLR）

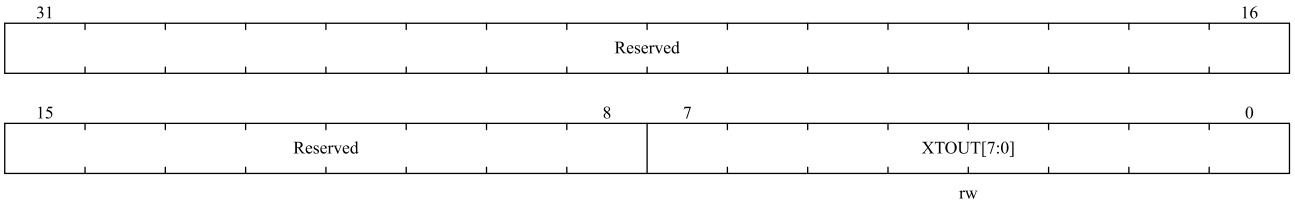
偏移地址：0x110
复位值：0x0000 0000



位域	名称	描述
31:1	Reserved	保留，硬件强制为 0。
0	XRXFOIC	Clear XIP Receive FIFO Overflow Interrupt.读该寄存器后，清除 Rx Fifo 上溢中断状态

25.6.35 QSPI XIP 连续传输超时寄存器（QSPI_XIP_TOUT）

偏移地址：0x114
复位值：0x0000 0000



位域	名称	描述
31:8	Reserved	保留，硬件强制为 0。
7:0	XTOUT[7:0]	XIP time out value in terms of HCLK.以 HCLK 为单位的 XIP 超时时间。一旦在连续 XIP 模式中选择了从机，如果没有请求的时间超出了该计数器指定时间，将取消从机选择。

26 比较器（COMP）

COMP 模块用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。当“INP”输入端电压高于”INM”输入端电压时，比较器输出为高电平，当“INP”输入端电压低于“INM”输入端电压时，比较器输出为低电平。

26.1 COMP 系统连接框图

COMP 模块最多支持 7 个独立比较器，统一挂接在 APB1 总线上。窗口模式通过配置窗口模式寄存器影响比较器 2、4、6 的 INPSEL 控制位实现，具体请参考窗口模式的说明。

图 26-1 比较器 1 和比较器 2 系统连接图

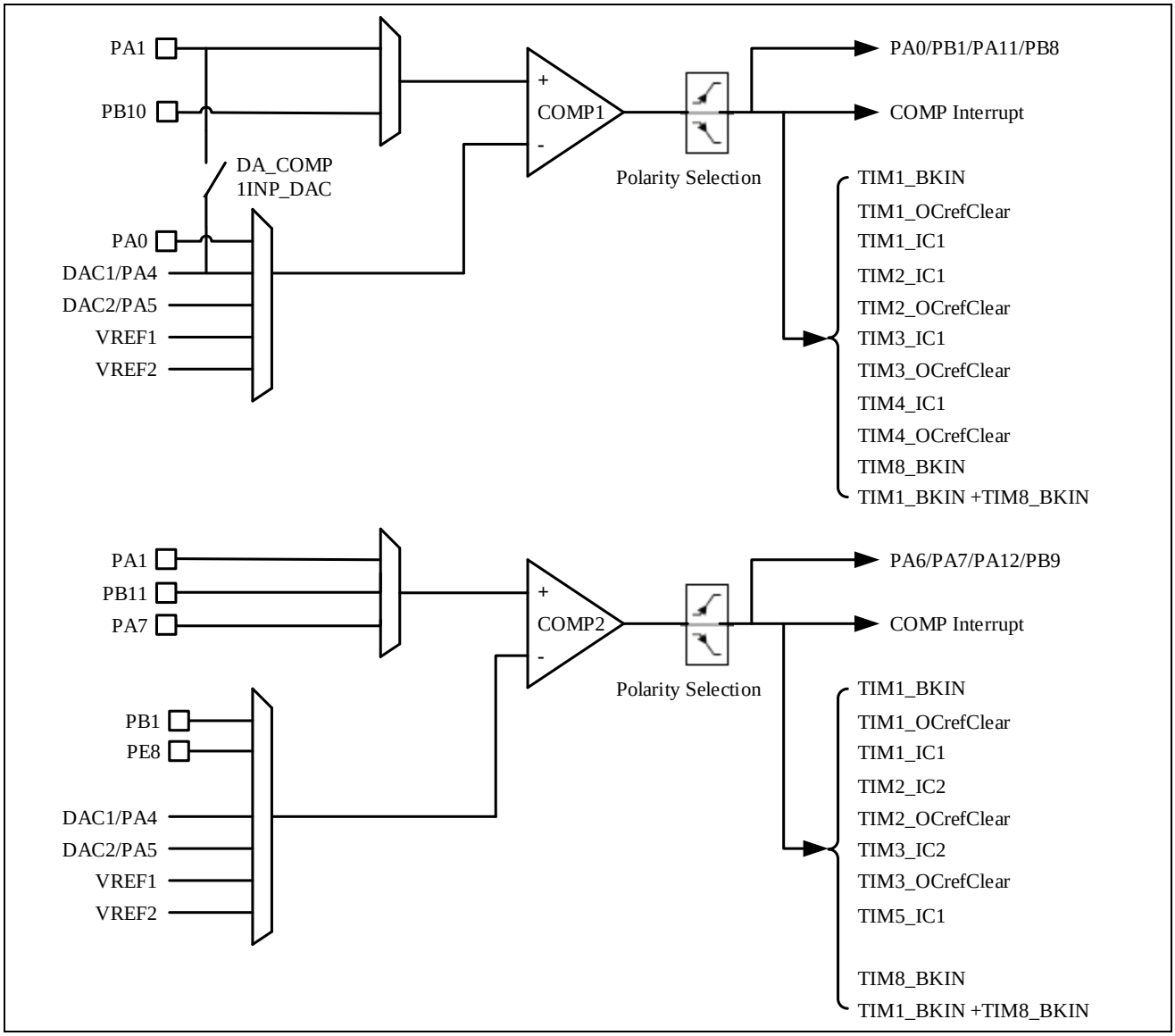


图 26-2 比较器 3 和比较器 4 系统连接图

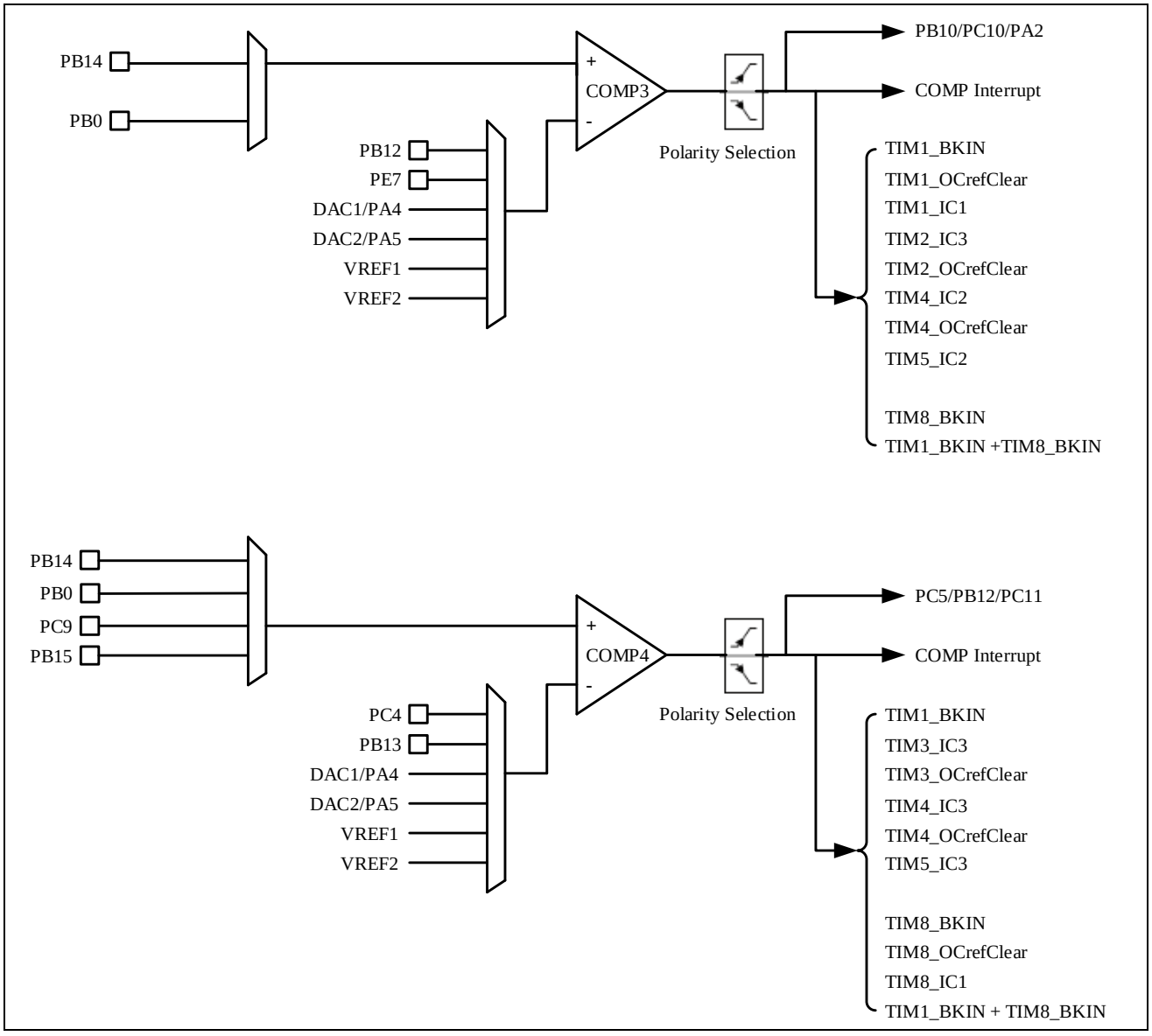
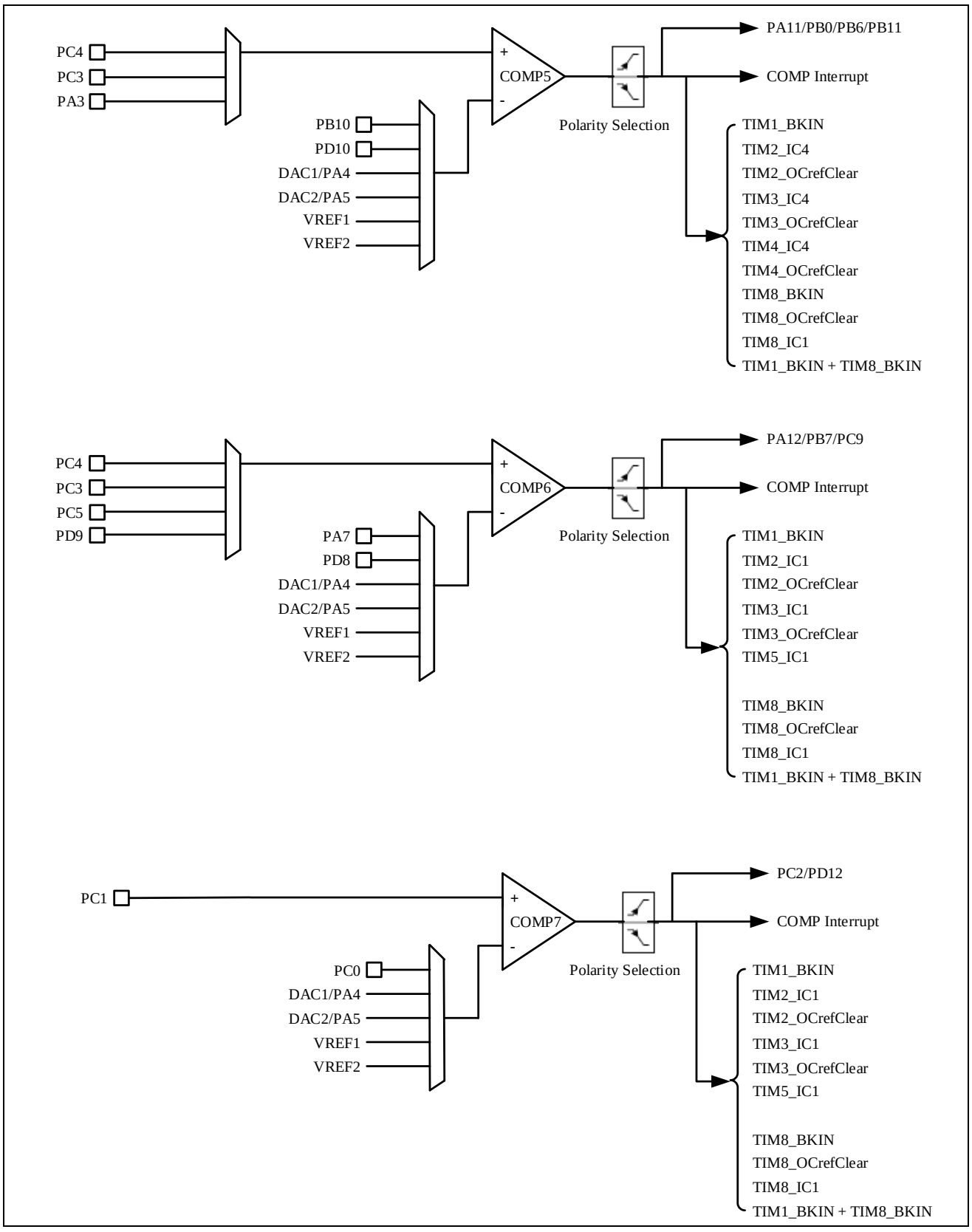


图 26-3 比较器 5，比较器 6，比较器 7 系统连接图



26.2 COMP 特性

- 最多 7 个独立的比较器
- 内置两个 64 级可编程的比较电压参考源 VREF1，VREF2
- 支持滤波时钟，滤波复位
- 输出极性可配置高、低
- 迟滞配置可配置无、低、中、高
- 比较结果可输出到 I/O 端口或触发定时器，用于捕获事件、OCREF_CLR 事件、刹车事件、产生中断
- 输入通道可复选 I/O 端口、DAC 的通道输出、VREF1、VREF2
- 可配只读或读写，在锁定的情况下需要复位才能解锁
- 支持消隐（Blanking），可配置产生 Blanking 的消隐源
- COMP1/COMP2、COMP3/COMP4、COMP5/COMP6 可以组成窗口比较器
- 可通过产生中断的方式将系统从 Sleep 模式唤醒
- 可配置滤波窗口大小
- 可配置滤波阈值大小
- 可配置用于滤波的采样频率

26.3 COMP 配置流程

完整的配置项包括如下所示，某些项目如果采用系统默认配置，跳过相应的配置项。

1. 可配置的迟滞 HYST 等级
2. 配置输出极性 POL
3. 配置输入选择，比较器正极 INPSEL，负极 INMSEL
4. 配置输出选择 OUTSEL
5. 配置消隐源 BLANK
6. 配置比较器窗口模式
7. 配置滤波器采样窗口 SAMPWIN（SAMPWIN+1）
8. 配置阈值 THRESH（阈值应当大于 SAMPWIN/2）
9. 配置滤波器采样频率（对于计时器应用，采样频率应当大于 5MHz）
10. 打开滤波器使能 FILC_FILEN
11. 打开比较器使能 COMP_EN

注:对于以上步骤，需先打开滤波器使能，再打开比较器使能，比较器使能需要在滤波（若启用）配置、使能完成后启用，此外在比较器控制寄存器锁定 LOCK 的情况下，只有通过复位才能取消锁定。

26.4 COMP 工作模式

26.4.1 窗口模式

7 个比较器可以组合成 3 个窗口比较器，如下:

- 比较器 1 和比较器 2 共享 PA1 形成窗口比较器
- 比较器 3 和比较器 4 共享 PB14 或 PB0 形成窗口比较器
- 比较器 5 和比较器 6 共享 PC3 或 PC4 形成窗口比较器

注：配置为窗口模式时，比较器 2、4、6 的 INPSEL 配置值不起作用，分别跟随比较器 1、3、5 的 INPSEL 配置值选择相应的模拟输入。

26.4.2 独立比较器

7 个比较器可独立配置，完成比较器功能。比较器的输出可以输出到 IO 端口，每一个比较器都有不同的重映射端口，通过配置可以选择比较器的输出，连接到相应的端口。

比较器输出，支持触发事件，比如可以配置成定时器 1，定时器 8 的刹车功能。

注：具体配置参考比较器互联关系

26.5 比较器互联关系

比较器输出端口的互联，可以参考 AFIO 重映射章节，在 AFIO_MAPR4（Address offset:0x24）定义了比较器 OUT 重映射的值，具体配置如下表所示。

OUT	COMP1[1:0]	COMP2[3:2]	COMP3[5:4]	COMP[7:6]	COMP5[9:8]	COMP6[11:10]	COMP7[12]
00	PA0	PA6	PB10	PC5	PB0	PC9	PC2
01	PB1	PA7	PC10	PB12	PB11	PA12	PD12
10	PA11	PA12	Not Used	Not Used	PB6	Not Used	--
11	PB8	PB9	PA2	PC11	PA11	PB7	--

注：1、COMP7 只有 1 位控制输出重映射

2、COMP1、COMP2、COMP5、COMP7 使能后，COMP1_OUT、COMP2_OUT、COMP5_OUT、COMP7_OUT 对应的 IO 端口，不能配置成其他外设的输出模式，可以是其他外设的输入模式、GPIO 的输入、GPIO 的输出。

比较器 INP 引脚有如下配置

INPSEL	COMP1	COMP2	COMP3	COMP4	COMP5	COMP6	COMP7
000	PA1	PA1	PB14	PB14	PC4	PC4	PC1
001	PB10	PB11	PB0	PB0	PC3	PC3	--
010	--	PA7	--	PC9	PA3	PC5	--
011	--	--	--	PB15	--	PD9	--
100	--	--	--	--	--	--	--
Other	--	--	--	--	--	--	--

比较器 INM 引脚有如下配置

INMSEL	COMP1	COMP2	COMP3	COMP4	COMP5	COMP6	COMP7
000	PA0	PB1	PB12	PC4	PB10	PA7	PC0
001	DAC1/PA4	PE8	PE7	PB13	PD10	PD8	DAC1/PA4
010	DAC2/PA5	DAC1/PA4	DAC1/PA4	DAC1/PA4	DAC1/PA4	DAC1/PA4	DAC2/PA5
011	VERF1	DAC2/PA5	DAC2/PA5	DAC2/PA5	DAC2/PA5	DAC2/PA5	VERF1
100	VERF2	VERF1	VERF1	VERF1	VERF1	VERF1	VERF2
101	--	VERF2	VERF2	VERF2	VERF2	VERF2	--
Other	--	--	--	--	--	--	--

比较器输出 TRIG 的信号有如下互联关系

TRIG	COMP1	COMP2	COMP3	COMP4	COMP5	COMP6	COMP7
0000	NC	NC	NC	NC	NC	NC	NC
0001	TIM1_BKIN	TIM1_BKIN	TIM1_BKIN	TIM1_BKIN	TIM1_BKIN	TIM1_BKIN	TIM1_BKIN
0010	TIM1_IC1	TIM1_IC1	TIM1_IC1	TIM3_IC3	TIM2_IC4	TIM2_IC1	TIM2_IC1
0011	TIM1_OCrefclear	TIM1_OCrefclear	TIM1_OCrefclear	TIM3_OCrefclear	TIM2_OCrefclear	TIM2_OCrefclear	TIM2_OCrefclear
0100	TIM2_IC1	TIM2_IC2	TIM2_IC3	TIM4_IC3	TIM3_IC4	TIM3_IC1	TIM3_IC1
0101	TIM2_OCrefclear	TIM2_OCrefclear	TIM2_OCrefclear	TIM4_OCrefclear	TIM3_OCrefclear	TIM3_OCrefclear	TIM3_OCrefclear
0110	TIM3_IC1	TIM3_IC2	TIM4_IC2	TIM5_IC3	TIM4_IC4	TIM5_IC1	TIM5_IC1
0111	TIM3_OCrefclear	TIM3_OCrefclear	TIM4_OCrefclear	--	TIM4_OCrefclear	--	--
1000	TIM4_IC1	TIM5_IC1	TIM5_IC2	TIM8_IC1	TIM8_IC1	TIM8_IC1	TIM8_IC1
1001	TIM4_OCrefclear	--	--	TIM8_OCrefclear	TIM8_OCrefclear	TIM8_OCrefclear	TIM8_OCrefclear
1010	TIM8_BKIN	TIM8_BKIN	TIM8_BKIN	TIM8_BKIN	TIM8_BKIN	TIM8_BKIN	TIM8_BKIN
1011	TIM1_BKIN + TIM8_BKIN	TIM1_BKIN + TIM8_BKIN	TIM1_BKIN + TIM8_BKIN	TIM1_BKIN + TIM8_BKIN	TIM1_BKIN + TIM8_BKIN	TIM1_BKIN + TIM8_BKIN	TIM1_BKIN + TIM8_BKIN
Other	--	--	--	--	--	--	--

26. 6 中断

COMP 支持中断响应，COMP1, COMP2,COMP3 共享 1 个中断入口，COMP4, COMP5,COMP6 共享 1 个中断入口，COMP7 独占 1 个中断入口。中断产生有如下 2 种情况。

- COMPx_CTRL.POL 极性不反转，中断使能，当 INPSEL > INMSEL 时，COMPx_CTRL.OUT 由硬件置为 1 时即产生比较器中断。

COMPx_CTRL.POL 极性反转，中断使能，当 INPSEL < INMSEL 时，COMPx_CTRL.OUT 由硬件置为 1 时即产生比较器中断。

26.7 COMP 寄存器

26.7.1 COMP 寄存器地址映像及复位值

表 26-1 OPAMP 寄存器地址映像和复位值

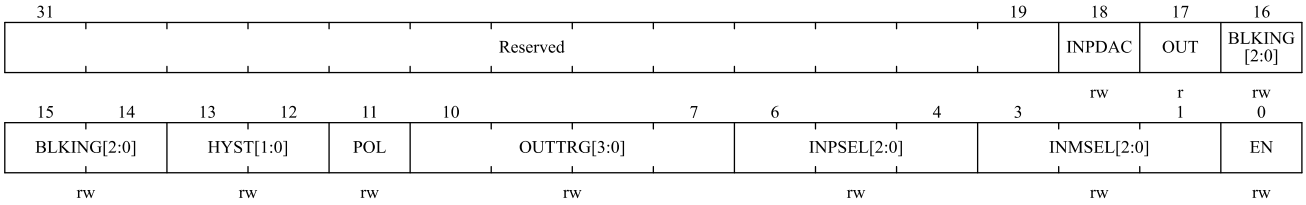
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
010h	COMP1_CTRL	Reserved													INPDAC	OUT	BLKING[2:0]			HYST[1:0]		POL	OUTSEL[3:0]				INPSEL[2:0]			INMSEL[2:0]			EN
	Reset Value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	COMP1_FILC	Reserved																					SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset Value																						0	0	0	0	0	0	0	0	0	0	0
018h	COMP1_FILP	Reserved																CLKPSC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01Ch	Reserved																																
020h	COMP2_CTRL	Reserved													INPDAC	OUT	BLKING[2:0]			HYST[1:0]		POL	OUTSEL[3:0]				INPSEL[2:0]			INMSEL[2:0]			EN
	Reset Value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
024h	COMP2_FILC	Reserved																					SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset Value																						0	0	0	0	0	0	0	0	0	0	0
028h	COMP2_FILP	Reserved																CLKPSC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	Reserved																																
030h	COMP3_CTRL	Reserved													INPDAC	OUT	BLKING[2:0]			HYST[1:0]		POL	OUTSEL[3:0]				INPSEL[2:0]			INMSEL[2:0]			EN
	Reset Value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
034h	COMP3_FILC	Reserved																					SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset Value																						0	0	0	0	0	0	0	0	0	0	0
038h	COMP3_FILP	Reserved																CLKPSC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03Ch	Reserved																																
040h	COMP4_CTRL	Reserved													INPDAC	OUT	BLKING[2:0]			HYST[1:0]		POL	OUTSEL[3:0]				INPSEL[2:0]			INMSEL[2:0]			EN
	Reset Value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
044h	COMP4_FILC	Reserved																					SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset Value																						0	0	0	0	0	0	0	0	0	0	0
048h	COMP4_FILP	Reserved																CLKPSC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04Ch	Reserved																																
050h	COMP5_CTRL	Reserved													INPDAC	OUT	BLKING[2:0]			HYST[1:0]		POL	OUTSEL[3:0]					INPSEL[2:0]			INMSEL[2:0]			EN
	Reset Value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
054h	COMP5_FILC	Reserved																					SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset Value																						0	0	0	0	0	0	0	0	0	0	0
058h	COMP5_FILP	Reserved																CLKPSC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
05Ch	Reserved																																
060h	COMP6_CTRL	Reserved													INPDAC	OUT	BLKING[2:0]			HYST[1:0]		POL	OUTSEL[3:0]					INPSEL[2:0]			INMSEL[2:0]			EN
	Reset Value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
064h	COMP6_FILC	Reserved																					SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset Value																						0	0	0	0	0	0	0	0	0	0	0
068h	COMP6_FILP	Reserved																CLKPSC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
06Ch	Reserved																																
070h	COMP7_CTRL	Reserved													INPDAC	OUT	BLKING[2:0]			HYST[1:0]		POL	OUTSEL[3:0]					INPSEL[2:0]			INMSEL[2:0]			EN
	Reset Value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
074h	COMP7_FILC	Reserved																					SAMPW[4:0]					THRESH[4:0]					FILEN
	Reset Value																						0	0	0	0	0	0	0	0	0	0	0
078h	COMP7_FILP	Reserved																CLKPSC[15:0]															
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07Ch	Reserved																																
080h	COMP_WINMODE	Reserved																													CMP56MD	CMP34MD	CMP12MD

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset Value																														0	0	0
084h	COMP_LOCK	Reserved																										CMP6LK	CMP5LK	CMP4LK	CMP3LK	CMP2LK	CMP1LK
	Reset Value																											0	0	0	0	0	0
088h	Reserved																																
08Ch	COMP_INTEN	Reserved																									CMP6IEN	CMP5IEN	CMP4IEN	CMP3IEN	CMP2IEN	CMP1IEN	CMP0IEN
	Reset Value																										0	0	0	0	0	0	0
090h	COMP_INTSTS	Reserved																									CMP7IS	CMP6IS	CMP5IS	CMP4IS	CMP3IS	CMP2IS	CMP1IS
	Reset Value																										0	0	0	0	0	0	0
094h	COMP_VREFSCL	Reserved																		VV2TRM[5:0]						VV2EN	VV1TRM[15:0]						VV1EN
	Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

26.7.2 COMP 控制状态寄存器（COMPx_CTRL）

比较器 1 的 INPDAC 有效，比较器 2 到比较器 7，INPDAC 位无效。
偏移地址:0x10，0x20，0x30，0x40，0x50，0x60，0x70
复位值:0x0000 0000

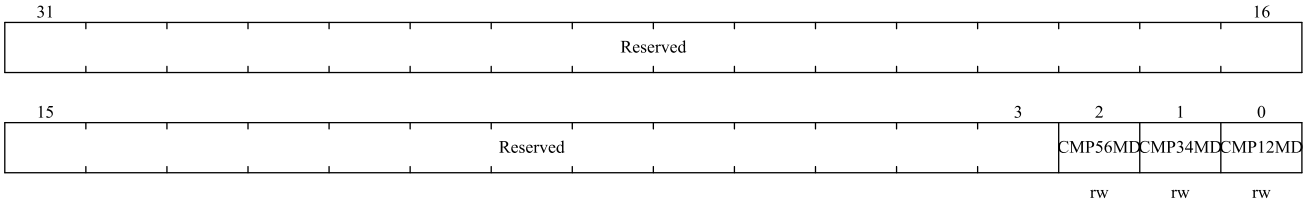


位域	名称	描述
31:19	Reserved	保留，硬件强制为 0。
18	INPDAC	比较器 1 非反相输入连接到 DAC 输出。 该位关闭比较器 1 非反转输入（PA1）和 DAC 输出 I/O（PA4）之间的开关。 0:开关打开 1:开关闭合 ***注:仅 COMP1 支持此位***
17	OUT	这个只读位是比较器输出状态。 0:输出低 1:输出高
16:14	BLKING[2:0]	这些位选择哪个定时器输出控制比较器输出消隐。 000:不消隐 001:选择 Tim1 OC5 作为消隐源 010:选择 Tim8 OC5 作为消隐源 其他配置:保留

位域	名称	描述
13:12	HYST[1:0]	这些位控制滞后水平。 00:无滞后 01:低滞后（仿真结果 5.1mV） 10:中等滞后（仿真结果 15mV） 其他配置:高滞后（仿真结果 25mV）
11	POL	该位用于反转比较器 1 的输出。 0:输出未反转 1:输出反转
10:7	OUTTRG[3:0]	这些位选择必须与比较器 1 输出连接的定时器输入。 *注意：具体枚举值对应关系参考 26.5"比较器互联关系"*
6:4	INPSEL[2:0]	比较器非反转输入选择。 *注意：具体枚举值对应关系参考 26.5"比较器互联关系"*
3:1	INMSEL[2:0]	这些位允许选择连接到比较器反向输入的源。 *注意：具体枚举值对应关系参考 26.5"比较器互联关系"*
0	EN	该位打开/关闭比较器。 0:比较器已禁用 1:比较器已启用

26.7.3 COMP 窗口比较寄存器（COMP_WINMODE）

偏移地址:0x80
复位值:0x0000 0000

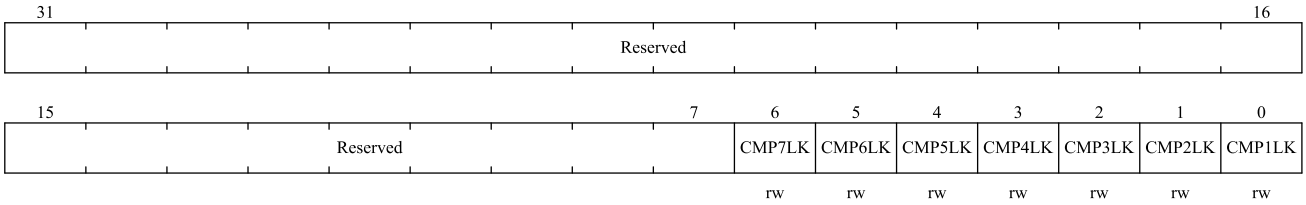


位域	名称	描述
31:3	Reserved	保留，硬件强制为 0。
2	CMP56MD	该位选择窗口模式:比较器的两个非反转输入共享非反转输入。 0:比较器 5 和 6 不能在窗口模式下使用。 1:比较器 5 和 6 可用于窗口模式。 *注：此位配置为 1 时，比较器 6 的 INPSEL 配置值不起作用，跟随比较器 5 的 INPSEL 配置值选择相应的模拟输入。*
1	CMP34MD	该位选择窗口模式:比较器的两个非反转输入共享非反转输入。 0:比较器 3 和 4 不能在窗口模式下使用。 1:比较器 3 和 4 可用于窗口模式。 *注：此位配置为 1 时，比较器 4 的 INPSEL 配置值不起作用，跟随比较器 3 的 INPSEL 配置值选择相应的模拟输入。*
0	CMP12MD	该位选择窗口模式:比较器的两个非反转输入共享非反转输入。 0:比较器 1 和 2 不能在窗口模式下使用。

位域	名称	描述
		1:比较器 1 和 2 可用于窗口模式。 *注：此位配置为 1 时，比较器 2 的 INPSEL 配置值不起作用，跟随比较器 1 的 INPSEL 配置值选择相应的模拟输入。*

26.7.4 COMP 锁寄存器（COMP_LOCK）

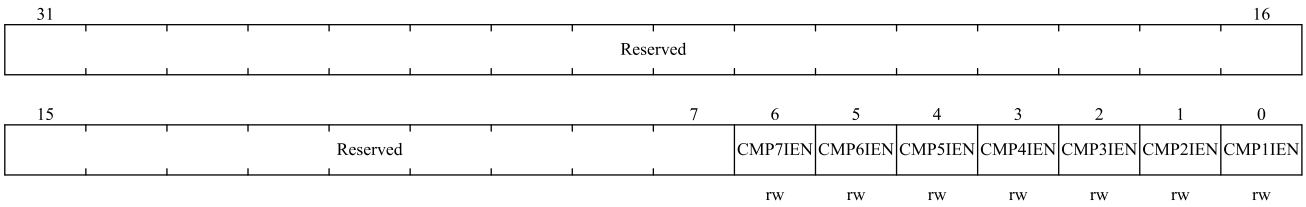
偏移地址:0x84
复位值:0x0000 0000



位域	名称	描述
31:7	Reserved	保留，硬件强制为 0。
6	CMP7LK	仅写一次，是由软件设置的，只能通过系统重置来清除，允许将 COMP7_CSR 注册为只读。 0: COMP7_CSR 是读写的。 1: COMP7_CSR 是只读的
5	CMP6LK	同 CMP7LK 的功能
4	CMP5LK	同 CMP7LK 的功能
3	CMP4LK	同 CMP7LK 的功能
2	CMP3LK	同 CMP7LK 的功能
1	CMP2LK	同 CMP7LK 的功能
0	CMP1LK	同 CMP7LK 的功能

26.7.5 COMP 中断使能寄存器（COMP_INTEN）

偏移地址:0x8C
复位值:0x0000 0000

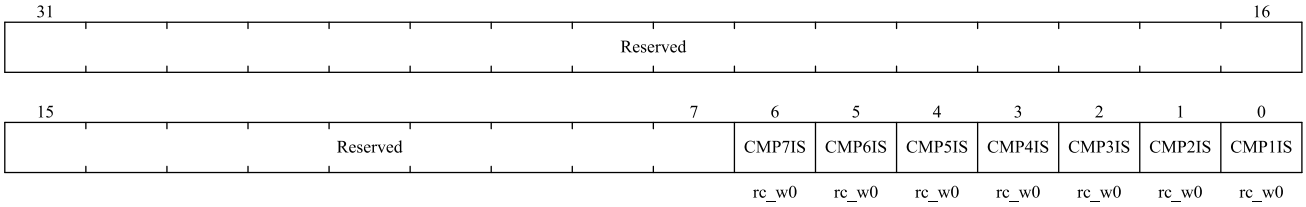


位域	名称	描述
31:7	Reserved	保留，硬件强制为 0。
6	CMP7IEN	该位控制 COMP7 的中断启用。 0:禁止

位域	名称	描述
		1:使用 ***注：COMPx_CTRL.OUT 高电平触发中断***
5	CMP6IEN	同 CMP7IEN 的功能
4	CMP5IEN	同 CMP7IEN 的功能
3	CMP4IEN	同 CMP7IEN 的功能
2	CMP3IEN	同 CMP7IEN 的功能
1	CMP2IEN	同 CMP7IEN 的功能
0	CMP1IEN	同 CMP7IEN 的功能

26.7.6 COMP 中断状态寄存器（COMP_INTSTS）

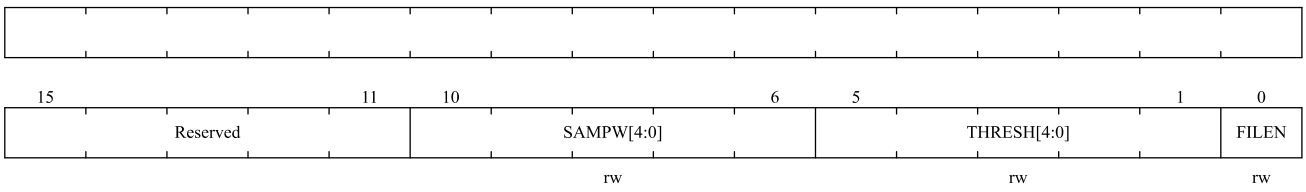
偏移地址:0x90
复位值:0x0000 0000



位域	名称	描述
31:7	Reserved	保留，硬件强制为 0。
6	CMP7IS	该位表示 COMP 的中断状态。 读入 1 表示中断状态产生。 写入 0 用以清除该状态位。
5	CMP6IS	同 CMP7IS 的功能
4	CMP5IS	同 CMP7IS 的功能
3	CMP4IS	同 CMP7IS 的功能
2	CMP3IS	同 CMP7IS 的功能
1	CMP2IS	同 CMP7IS 的功能
0	CMP1IS	同 CMP7IS 的功能

26.7.7 COMP 滤波寄存器（COMPx_FILC）

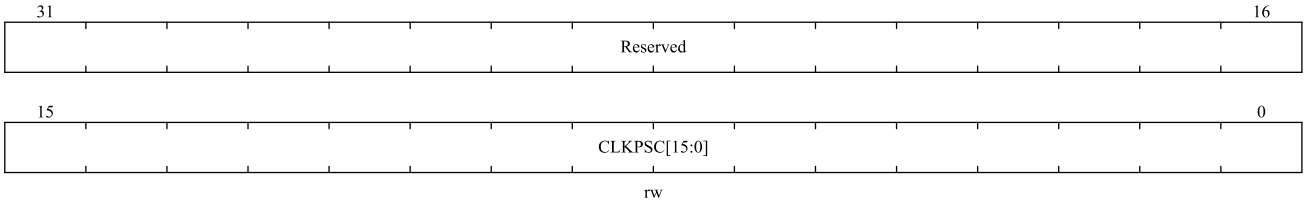
偏移地址:0x14，0x24，0x34，0x44，0x54，0x64，0x74
复位值:0x0000 0000



位域	名称	描述
31:11	Reserved	保留，硬件强制为 0。
10:6	SAMPW[4:0]	过滤器窗口大小。要监视的数量为 SAMPW +1。
5:1	THRESH[4:0]	过滤器阈值。至少相反状态的阈值采样必须出现在采样窗口中，以便输出更改状态。 为了正常工作，THRESH 的值必须大于 SAMPW /2。
0	FILEN	过滤器启用。 0:禁能 1:启用

26.7.8 COMP 滤波分频寄存器（COMPx_FILP）

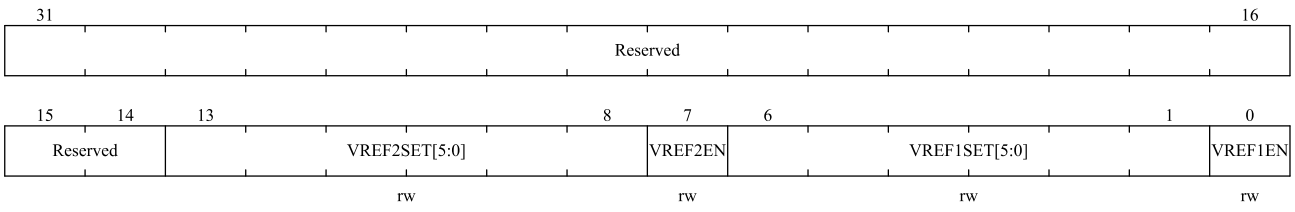
偏移地址:0x18，0x28，0x38，0x48，0x58，0x68，0x78
复位值:0x0000 0000



位域	名称	描述
31:16	Reserved	保留，硬件强制为 0。
15:0	CLKPSC[15:0]	低滤波器采样时钟预分频。 采样之间的系统时钟数= CLKPSC +1，例如 0:每个周期 1:每 2 个周期 2:每 3 个周期 …

26.7.9 COMP 电压参考寄存器（COMP_VREFSCL）

偏移地址:0x94
复位值:0x0000 0000



位域	名称	描述
31:14	Reserved	保留，硬件强制为 0。
13:8	VREF2SET[5:0]	VREF2 电压定标器微调值。
7	VREF2EN	VREF2 电压定标器:

位域	名称	描述
		0:禁用 1:启用
6:1	VREF1SET[5:0]	VREF1 电压定标器微调值。
0	VREF1EN	VREF1 电压定标器: 0:禁用 1:启用

27 运算放大器（OPAMP）

CM32M4xxR 系列内置 OPAMP 模块，最多集成 4 个 OPAMP，可以灵活配置，适用于独立运算放大器、可编程增益放大器(PGA)，和跟随器等模式应用。内部的 4 个运算放大器可以配置为具有不同增益的组合放大器，也可以使用外部电阻进行级联。OPAMP 的输入范围是 0V 到 V_{DDA}，输出范围是 0.1V 到 V_{DDA}-0.1V。OPAMP 的输出内部连接到 ADC 的输入通道用于模拟信号测量。

27.1 OPAMP 特性

- 最多支持 4 个独立可配的运算放大器
- 支持轨到轨输入，输入范围是 0 到 V_{DDA}，输出范围是 0.1 到 V_{DDA}-0.1 可编程增益
- 可通过外部电阻连接配置为仪表放大器
- 多种工作模式可配置:
  - 独立运放模式（所有端口引出，可通过外部电阻设置放大倍数）
  - 电压跟随器
  - 正/反向可编程增益放大器
  - 两个运放的差分运放
- 内部 PGA GAIN 可编程增益设置为 2X、4X、8X、16X、32X
- 最大 4MHz 的增益带宽积
- 支持 TIM1_CC6 对 OPAMP1 和 OPAMP2 输入 PIN 的自动切换，TIM8_CC6 对 OPAMP3 和 OPAMP4 的输入 PIN 自动切换
- 支持独立写保护

27.1.1 OPAMP 功能描述

四个 OPAMP 通过寄存器选择可以配置为各种不同的 PGA 模式，也可以配置为用户使用外部元件的 OPAMP 功能。OPAMP 的输出可以作为 ADC 的通道输入，四个 OPAMP 输出连接到 ADC 的模拟通道如下。

OPAMP1 的输出连接到 ADC1 的模拟输入通道 3

OPAMP2 的输出连接到 ADC2 的模拟输入通道 3

OPAMP3 的输出连接到 ADC3 的模拟输入通道 1

OPAMP4 的输出连接到 ADC4 的模拟输入通道 3

图 27-1 OPAMP1 和 OPAMP2 连接图框图

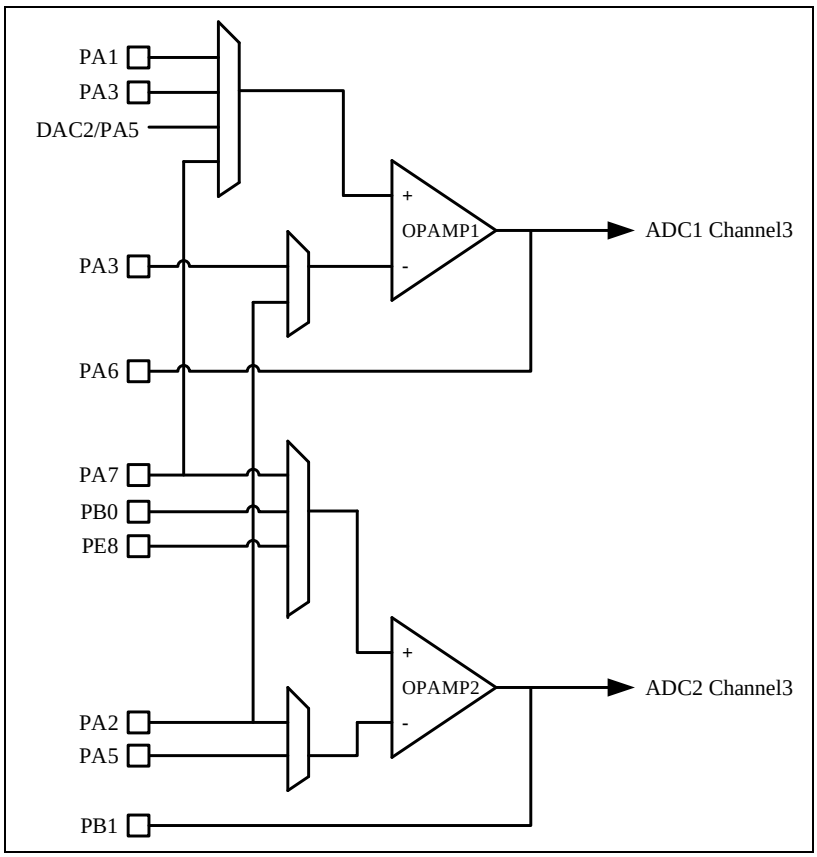
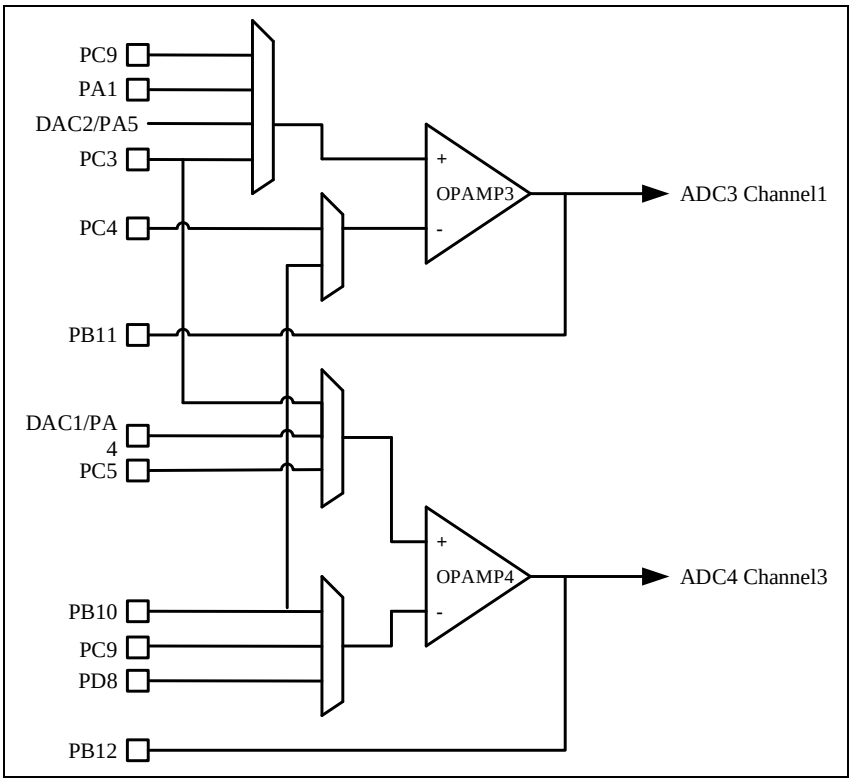


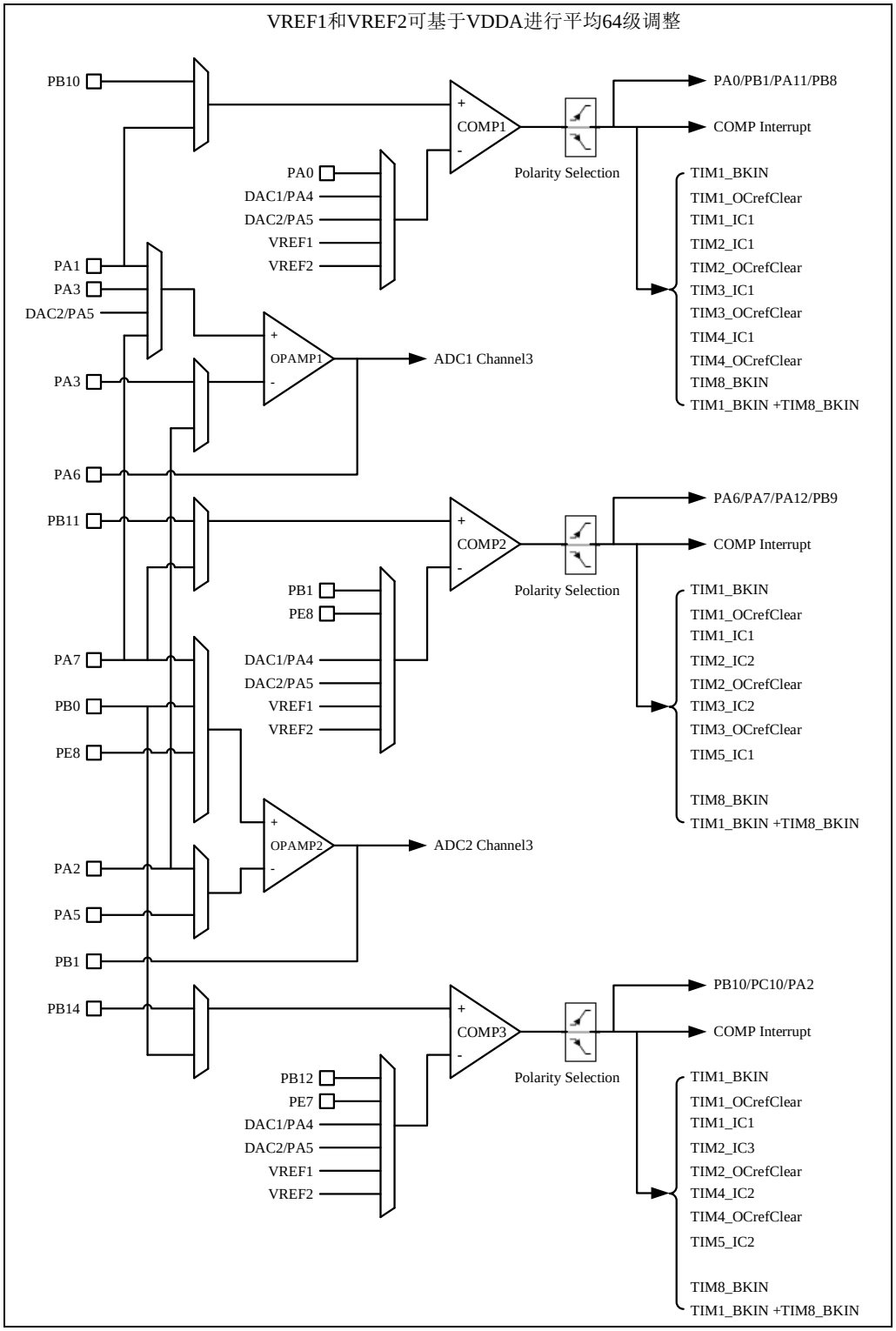
图 27-2 OPAMP3 和 OPAMP4 连接图框图



27.1.2 OPAMP 与 COMP 的内部的连接

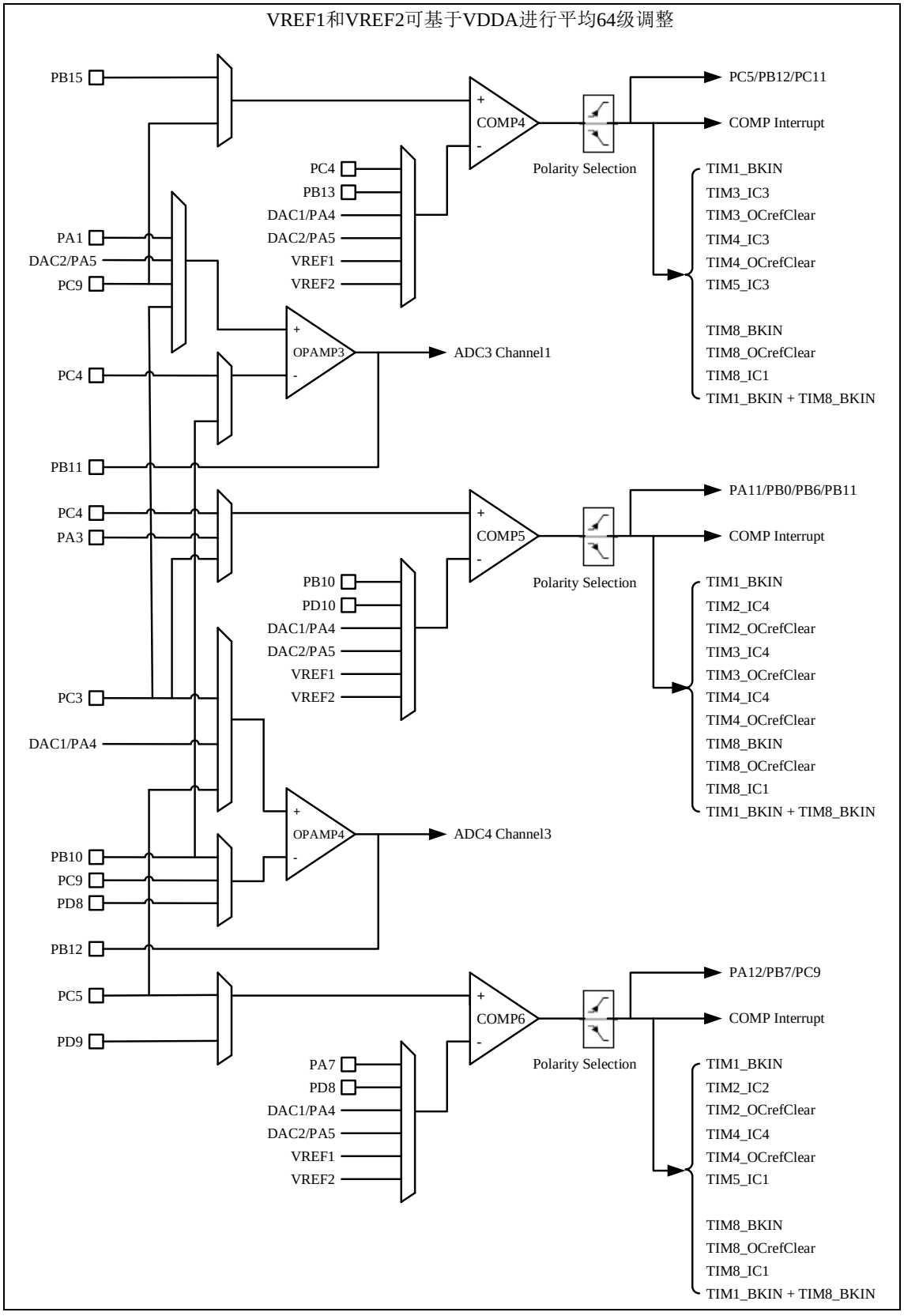
OPAMP1、OPAMP2、COMP1、COMP2、COMP3、ADC1 和 ADC2 构成一组模拟联动应用，其拓扑关系如下:

图 27-3 模拟模块联动关系 1



OPAMP3、OPAMP4、COMP4、COMP5、COMP6、ADC3 和 ADC4 构成一组模拟联动应用，其拓扑关系如下：

图 27-4 模拟模块联动关系 2



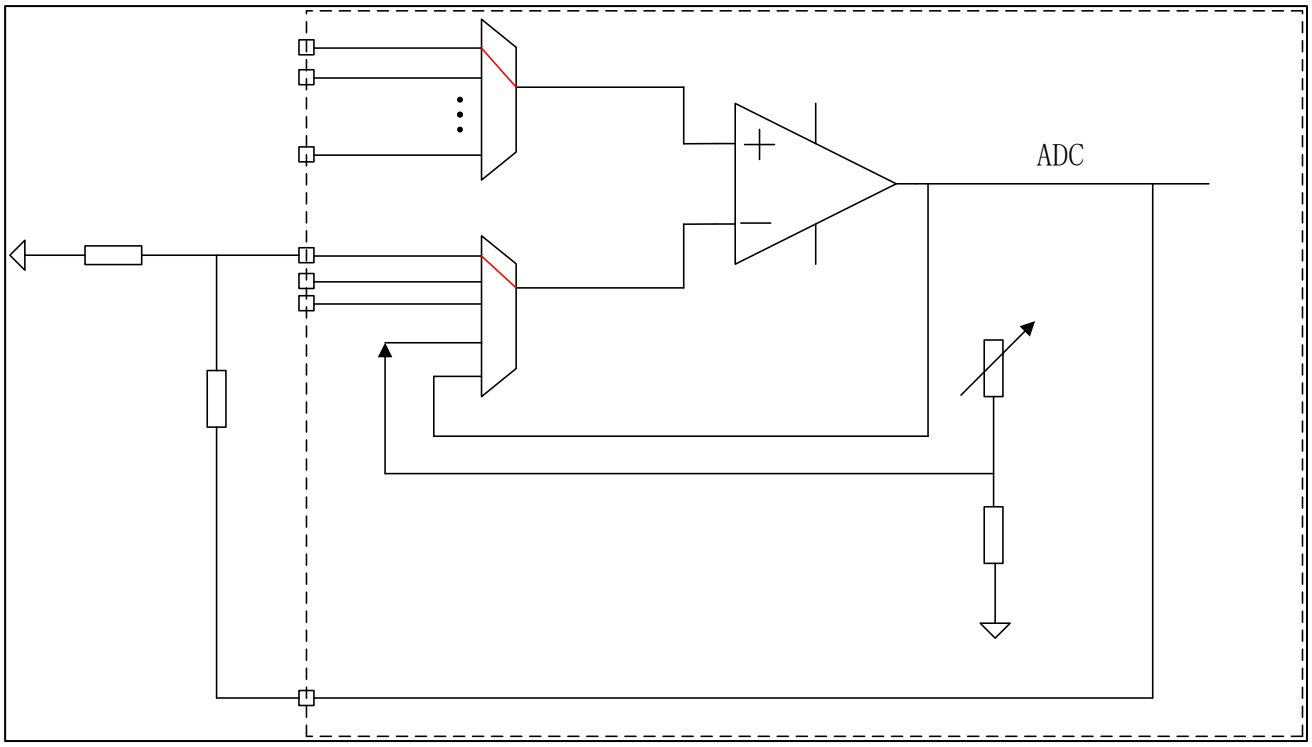
27.2 OPAMP 工作模式

27.2.1 OPAMP 独立运算放大器模式

外部放大模式即放大倍数由连接的电阻电容决定。OPAMP_CS.MOD 设置为 2'b00 或 2'b01 时为运放功能，OPAMP_CS.VPSSEL 或 OPAMP_CS.VPSEL 选择正端输入，OPAMP_CS.VMSSEL 或 OPAMP_CS.VMSEL 选择负端输入。使用外部电阻组成闭环放大系统。

四个完全独立 OPAMP,此时增益大小由外部电阻网络决定，也可以按需要进行级联，组成所需要的放大增益，比如通过三个 OPAMP 构成仪表放大器。如下图所示，OPAMP 正端、负端、输出端均连接至外部端口，放大系数由外部阻容网络决定。

图 27-5 OPAMP 独立运算放大器模式



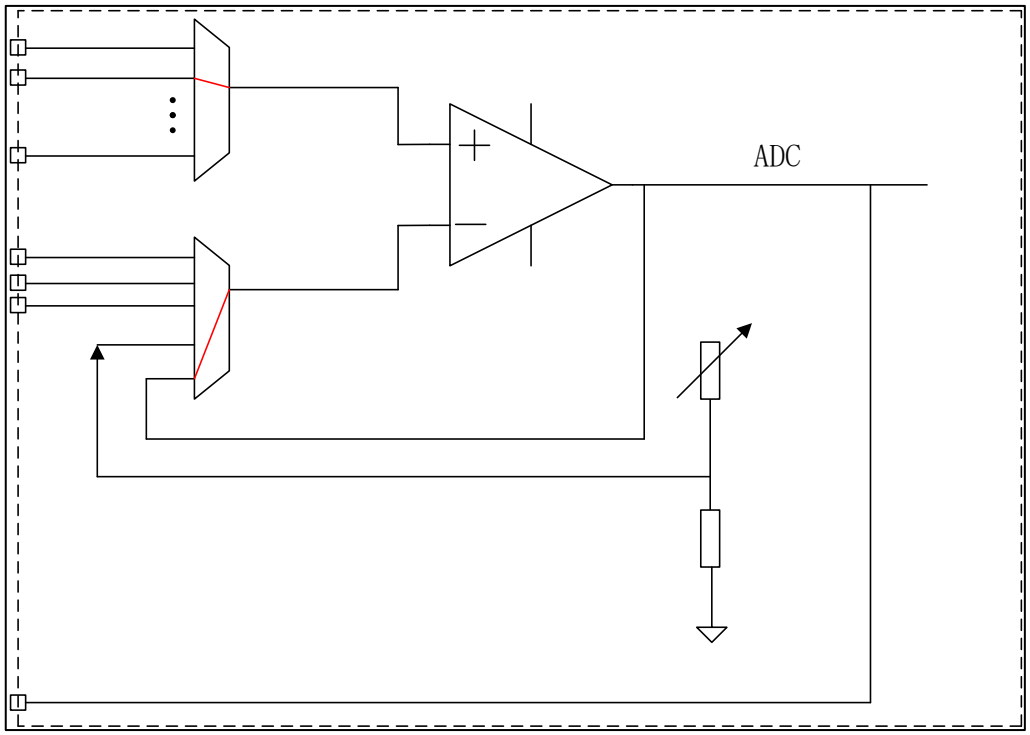
27.2.2 OPAMP 跟随模式

跟随模式，电压是直接跟随，VMSEL 端必须配置为和 OPAMP 输出端口直连。

OPAMP_CS.MOD 设置为 2'b11 为内部跟随功能，OPAMP_CS.VPSSEL 或 OPAMP_CS.VPSEL 选择正端输入，OPAMP_CS.VMSSEL 或 OPAMP_CS.VMSEL 由芯片内部连接到输出端口。

没有占用的 VM 引脚可以作为其他 GPIO 使用。

图 27-6 OPAMP 跟随模式



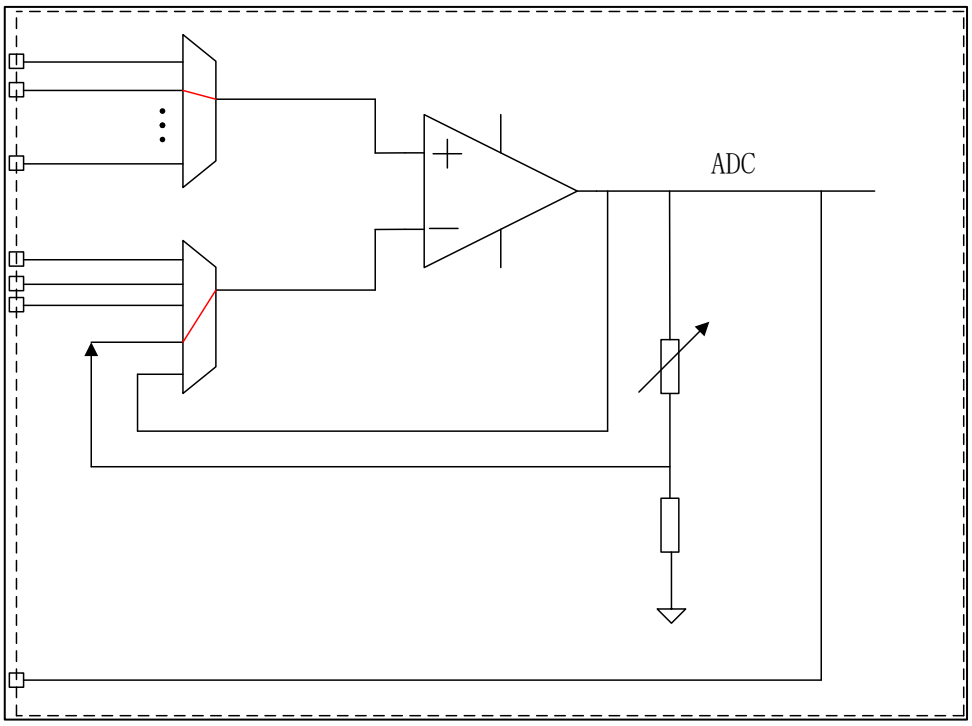
27.2.3 OPAMP 内部可编程增益（PGA）模式

内部放大模式，即 PGA 模式，通过内置的电阻反馈网络对输入电压进行放大。

OPAMP_CS.MOD 设置为 2'b10 为 PGA 功能，支持 2/4/8/16/32 放大倍数，OPAMP_CS.VMSSEL 或 OPAMP_CS.VMSEL 引脚必须设置为浮空。OPAMP_CS.VPSSEL 或 OPAMP_CS.VPSEL 选择正端输入。正端输入可以连接到外部引脚，该外部引脚可以是其他 OPAMP 的输出端口或者电阻网络。设置 OPAMP_CS.PGAGAN，进行增益选择。OPAMP 的输出可以是其他 OPAMP 的输入或者电阻网络。

OPAMP 的 VM 输入引脚可以作为普通 GPIO 使用。

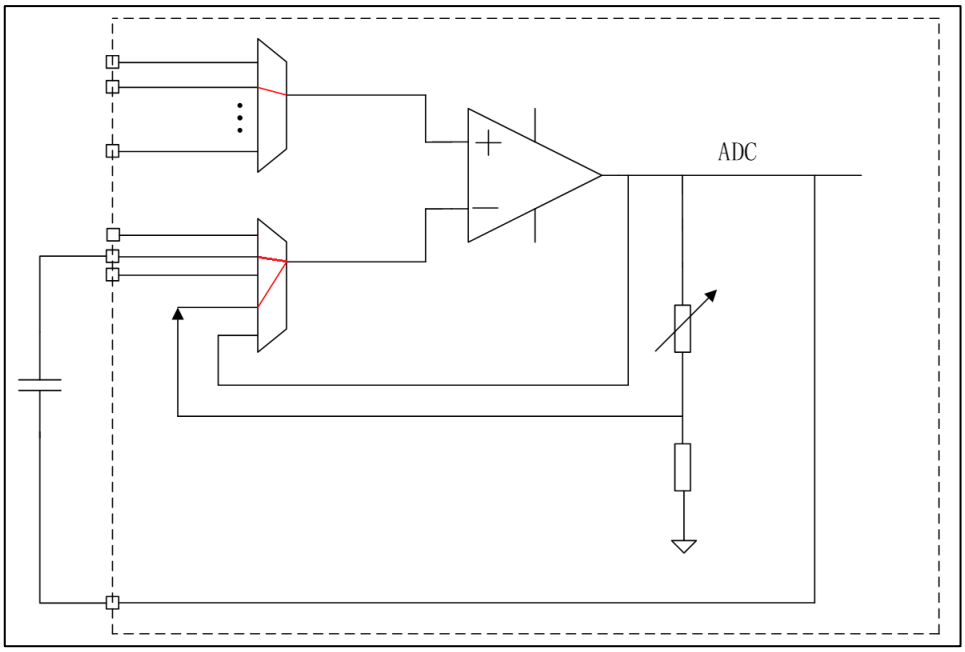
图 27-7 内部可编程增益模式



27.2.4 OPAMP 带滤波内部增益模式

带滤波内部放大模式，即在内部 PGA 模式下，放大电压可调，支持 2/4/8/16/32,同时 OPAMP_CS.VMSSEL 或 OPAMP_CS.VMSEL 引脚设置为和外部引脚相连，在 OPAMP 负端可以连接电容等元件。

图 27-8 带滤波内部增益模式



27.2.5 OPAMP 校正

芯片出产已进行校正，用户可以根据实际环境，再次进行校正。

注：具体校正方法联系中移物联网，获取相关资料。

27.2.6 OPAMP 独立写保护

通过配置 OPAMP_LOCK 寄存器，可以对 4 个 OPAMP 的写保护进行独立设置。当设置了写保护后，软件将不能对相应 OPAMP 寄存器进行写操作，只有在芯片复位后，才能取消写保护功能。

27.2.7 OPAMP TIMER 控制切换模式

在一些应用中，可以通过 TIMx_CC6 对比较器进行输入切换。TIM1_CC6 控制 OPAMP1 和 OPAMP2 的输入切换，TIM8_CC6 控制 OPAMP3 和 OPAMP4 的输入切换。

当 TIMx_CC6 为 1 时 OPAMP1 和 OPAMP2 的选择 VPSSEL/VMSSEL 所配置端口作为输入,否则使用 VPSEL/VMSEL。当 TIMx_CC8 为 1 时 OPAMP 和 OPAMP4 的 MUX 用的选择控制字是 VPSSEL/VMSSEL 所配置端口作为输入,否则使用 VPSEL/VMSEL。

OPAMP_CS.TCMEN 置 1，开启自动切换输入功能，配置自动切换的流程如下所示。

- 开启自动切换功能 TCMEN（四个 OPAMP 独立控制）
- 配置好两次转换的 MUX 配置（VPSEL,VMSEL,VPSSEL,VMSSEL）
- 启动 OPAMP 与 TIM

27.3 OPAMP 寄存器

27.3.1 OPAMP 寄存器地址映像及复位值

表 27-1 OPAMP 寄存器地址映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	OPAMP_CS1	Reserved								VREF_OPT[1:0]		VPSSEL[2:0]			VMSSEL[1:0]		TCMEN	RANGE	CALOUT	TSTREF	Reserved	CALON	VPSEL[2:0]			VMSEL[1:0]		PGAGAN[2:0]			MOD[1:0]		EN
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010h	OPAMP_CS2	Reserved								VREF_OPT[1:0]		VPSSEL[2:0]			VMSSEL[1:0]		TCMEN	RANGE	CALOUT	TSTREF	Reserved	CALON	VPSEL[2:0]			VMSEL[1:0]		PGAGAN[2:0]			MOD[1:0]		EN
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
020h	OPAMP_CS3	Reserved								VREF_OPT[1:0]		VPSSEL[2:0]			VMSSEL[1:0]		TCMEN	RANGE	CALOUT	TSTREF	Reserved	CALON	VPSEL[2:0]			VMSEL[1:0]		PGAGAN[2:0]			MOD[1:0]		EN
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
030h	OPAMP_CS4	Reserved								VREF_OPT[1:0]		VPSSEL[2:0]			VMSSEL[1:0]		TCMEN	RANGE	CALOUT	TSTREF	Reserved	CALON	VPSEL[2:0]			VMSEL[1:0]		PGAGAN[2:0]			MOD[1:0]		EN
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
040h	OPAMP_LOCK	Reserved																												OPAMP4LK	OPAMP3LK	OPAMP2LK	OPAMP1LK
	Reset Value																													0	0	0	0

27.3.2 OPAMP 控制状态寄存器（OPAMP_CS）

偏移地址：0x00, 0x10, 0x20, 0x30

复位值：0x0000 0000

31	24	23	22	21	19	18	17	16
Reserved		VREF_OPT		VPSSEL		VMSSEL		TCMEN
		rw		rw		rw		rw

15	14	13	12	11	10	8	7	6	5	3	2	1	0
RANGE	CALOUT	TSTREF	Reserved	CALON	VPSEL		VMSEL		PGAGAN		MOD		EN
rw	r	rw		rw	rw		rw		rw		rw		rw

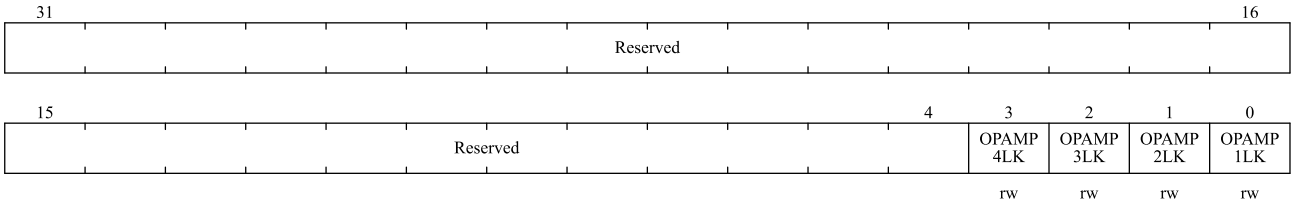
位域	名称	描述
31:22	Reserved	保留，硬件强制为 0。
23:22	VREF_OPT	校准电压 Trim 档位: 2'b00：opa_vref = 3.3% VDDA 2'b01：opa_vref = 10% VDDA(calibre PMOS) 2'b10：opa_vref = 50% VDDA 2'b11：opa_vref = 90% VDDA(calibre NMOS)
21:19	VPSSEL[2:0]	OPAMP 正向输入端从选择（Non inverted input secondary selection） 枚举值 | OPAMP1 | OPAMP2 | OPAMP3 | OPAMP4 000 | PA1 | PA7 | PC9 | PC3 001 | PA3 | PB0 | PA1 | DAC1/PA4 010 | DAC2/PA5 | PE8 | DAC2/PA5 | PC5 011 | PA7 | 保留 | PC3 | 保留 100 | 保留 | 保留 | 保留 | 保留 其他 | 保留 | 保留 | 保留 | 保留 *注：DAC2 直接连到 PA5*
18:17	VMSSEL[1:0]	OPAMP 反向输入端从选择（Inverted input secondary selection） 枚举值 | OPAMP1 | OPAMP2 | OPAMP3 | OPAMP4 00 | PA3 | PA2 | PC4 | PB10 01 | PA2 | PA5 | PB10 | PC9 10 | 保留 | 保留 | 保留 | PD8 11 | 浮空 | 浮空 | 浮空 | 浮空 ***注：VM 浮空（用于内部 PGA（No Filter）模式，Follow 模式）***
16	TCMEN	定时器控制自动切换模式使能（Timer controlled Mux mode enable）。

<table>
<tr><th>位域</th><th>名称</th><th>描述</th></tr>
<tr><td></td><td></td><td>此位由软件设置或清除，用于控制自动切换主次输入（VPSEL,VMSEL 和 VPSSEL,VMSSEL）。
TIM1_CC6 对 OPAMP1 和 OPAMP2 自动切换, TIM8_CC6 对 OPAMP3 和 OPAMP4 自动切换。
0：关闭自动切换
1：充许自动切换</td></tr>
<tr><td>15</td><td>RANGE</td><td>OPAMP 电压区间（OPAMP Operational amplifier power supply range）。
0：低电压区间（VDDA ＜ 2.4V）
1：高电压区间</td></tr>
<tr><td>14</td><td>CALOUT</td><td>OPAMP 校准输出（Operation amplifier calibration output）
校准 NMOS 管，当此信号由 1 到 0 变化时，表示校验 offset 值设定合适；
校准 PMOS 管，当此信号由 0 到 1 变化时，表示校验 offset 值设定合适；</td></tr>
<tr><td>13</td><td>TSTREF</td><td>内部保留，设置为 0</td></tr>
<tr><td>12</td><td>Reserved</td><td>保留，硬件强制为 0。</td></tr>
<tr><td>11</td><td>CALON</td><td>校准模式使能（Calibration mode enabled）
0：正常模式
1：校准模式</td></tr>
<tr><td>10:8</td><td>VPSEL[2:0]</td><td>OPAMP 正向输入端选择（Non Inverted input selection）
<table>
<tr><th>枚举值</th><th>OPAMP1</th><th>OPAMP2</th><th>OPAMP3</th><th>OPAMP4</th></tr>
<tr><td>000</td><td>PA1</td><td>PA7</td><td>PC9</td><td>PC3</td></tr>
<tr><td>001</td><td>PA3</td><td>PB0</td><td>PA1</td><td>DAC1/PA4</td></tr>
<tr><td>010</td><td>DAC2/PA5</td><td>PE8</td><td>DAC2/PA5</td><td>PC5</td></tr>
<tr><td>011</td><td>PA7</td><td>保留</td><td>PC3</td><td>保留</td></tr>
<tr><td>100</td><td>保留</td><td>保留</td><td>保留</td><td>保留</td></tr>
<tr><td>其他</td><td>保留</td><td>保留</td><td>保留</td><td>保留</td></tr>
</table>
注：DAC2 直接连到 PA5</td></tr>
<tr><td>7:6</td><td>VMSEL[1:0]</td><td>OPAMP 反向输入端从选择（Inverted input selection）
<table>
<tr><th>枚举值</th><th>OPAMP1</th><th>OPAMP2</th><th>OPAMP3</th><th>OPAMP4</th></tr>
<tr><td>00</td><td>PA3</td><td>PA2</td><td>PC4</td><td>PB10</td></tr>
<tr><td>01</td><td>PA2</td><td>PA5</td><td>PB10</td><td>PC9</td></tr>
<tr><td>10</td><td>保留</td><td>保留</td><td>保留</td><td>PD8</td></tr>
<tr><td>11</td><td>浮空</td><td>浮空</td><td>浮空</td><td>浮空</td></tr>
</table>
注：浮空（用于内部 PGA（No Filter）模式，follow 模式）</td></tr>
<tr><td>5:3</td><td>PGAGAN[2:0]</td><td>OPAMP 增益设置（Operational amplifier Programmable amplifier gain value）
000：内部 PGA 增益 2
001：内部 PGA 增益 4
010：内部 PGA 增益 8
011：内部 PGA 增益 16
100：内部 PGA 增益 32
其他：内部 PGA 增益 2</td></tr>
</table>

位域	名称	描述
2:1	MOD[1:0]	OPAMP 模式选择（Operational amplifier PGA mode） 0x：外部放大模式 10：内部 PGA 使能 11：内部跟随模式
0	EN	OPAMP 使能（Operational amplifier Enable） 0：禁能 1：使能

27.3.3 OPAMP 锁寄存器（OPAMP_LOCK）

偏移地址：0x40
复位值：0x0000 0000



位域	名称	描述
31:4	Reserved	保留，硬件强制为 0。
3	OPAMP4LK	OPAMP4 锁（OPAMP4 lock bit） 在复位后，此位仅能写一次 0：OPAMP 寄存器可读写 1：OPAMP 寄存器只读
2	OPAMP3LK	同 OPAMP 4LK
1	OPAMP2LK	同 OPAMP 4LK
0	OPAMP1LK	同 OPAMP 4LK

28 触摸传感器控制器（TSC）

28.1 TSC 简介

TSC 是 Touch Sensor Controller 的简称。主要应用于电容式触摸按键应用场景，硬件检测到有手触摸时唤醒 CPU，由其进行更加精确的触摸检测及后续的算法处理。TSC 相关时间使用 TSC_CLOCK 描述，TSC_CLOCK 的值为 LSE 的 32.768KHz 或 LSI 的 40KHz，由时钟选择确定。

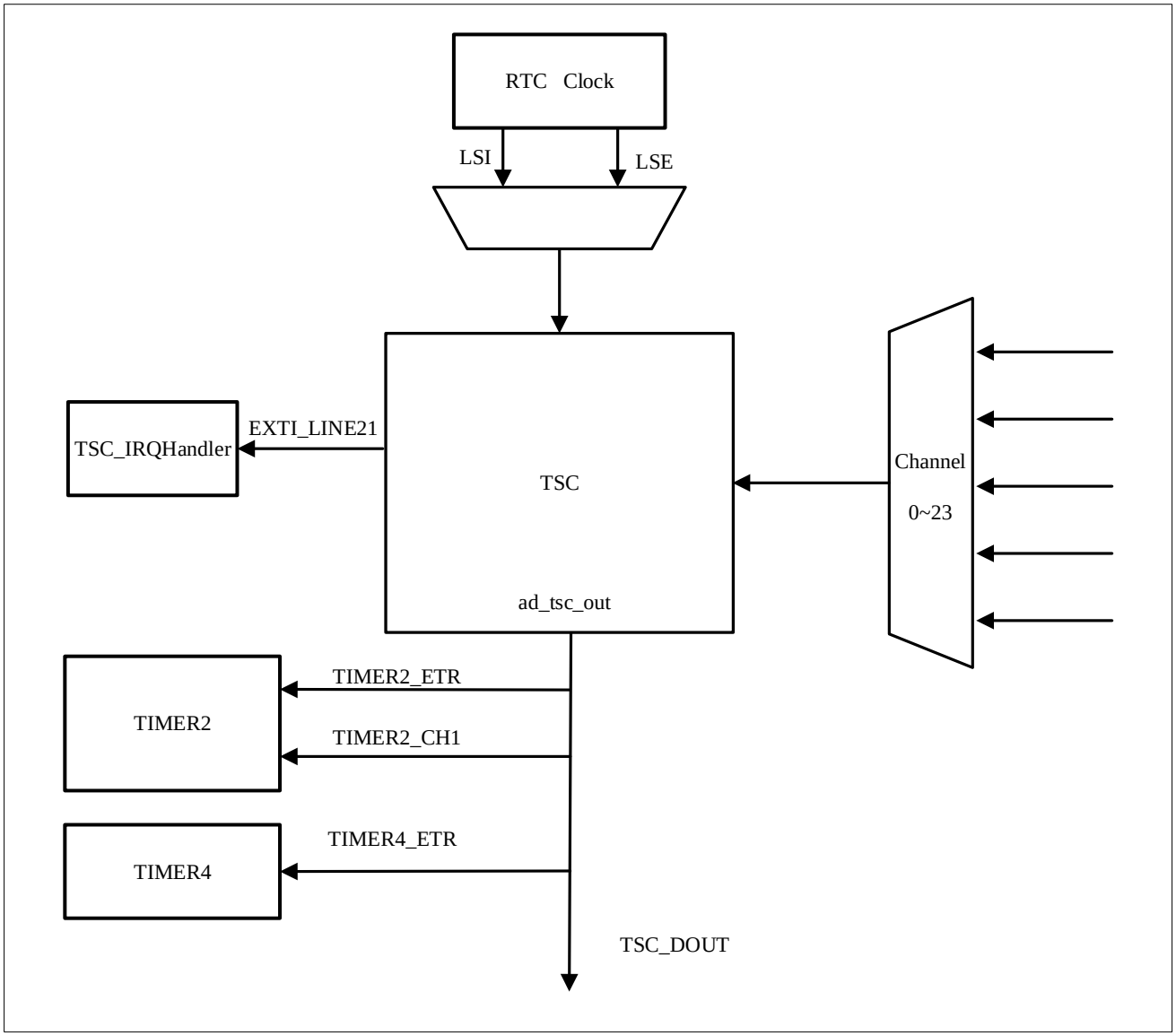
28.2 TSC 主要特性

TSC 控制器的主要特性如下：

- 支持电容式触控功能，最大支持 24 个通道，每个通道可固件使能；
- 在低功耗模式下（Sleep、Stop0、Stop2），检测到触控操作，唤醒 CPU（软件需要配置使能相应寄存器）；
- 在正常工作模式下，检测到触操作，中断/事件通知 CPU（软件需要配置使能相应寄存器）；
- 检测到触控操作，有寄存器指示相应通道产生唤醒，每个通道均有一个唤醒指示标志，供固件查询；
- 所有通道开启检测，触控唤醒的平均功耗<10uA；
- 触控唤醒检测的间隔可固件配置（TSCWKUP_CNT[13:0]的 WAKUPCNT[13:0]位），RTC 唤醒时间为 1024 * TSC_CLOCK ，选择 LSE 的 32.768KHz 时钟时，时间范围 23.4 ~ 367ms（步进 31.25ms）；选择 LSI 的 40KHz 时钟，时间范围 19.2 ~ 300.7ms（步进范围 25.6ms）；
- 每个通道的检测时间可固件配置（TSC_CTRL[3:0]的 DET_PERIOD[3:0]位），选择 LSE 的 32.768KHz 时钟时，时间范围 244.1us ~ 3173.8us（步进 8*32.768KHz=244.1us）；选择 LSI 的 40KHz 时钟时，时间范围 200us ~ 1600us（步进 8*40KHz=200us）；
- 检测的脉冲个数阈值固件设置，包括 Base 和 Delta 值，每个通道均有独立的阈值，根据固件设置值，硬件进行脉冲计数并比较（great=Base+Delta，less=Base–Delta，可配置 TSC_STS 寄存器的 CNT_VAL[10:0]大于 great 或小于 less 值时，触控有效）；
- 支持校准功能（软件修改 Base 及 Delta）；
- 硬件初步检测到按键信号后，转由软件算法准确判断触控信号。
  - 切换到定时器输入捕获通道，由固件进行触控算法实现；
  - 切换到触控唤醒自动检测逻辑，由硬件进行触控唤醒检测。

28.3 功能描述

图 28-1 TSC 框图



TSC 模块系统框图如上图所示，可以在 stop0、 stop2、 sleep 以及运行状态下检测。

28.4 软件操作流程

TSC 模块可以工作在各种芯片低功耗（standby 除外）及正常工作模式中，软件如需使能 TSC 进行扫描，需要进行以下配置:

- 设置需要使能的通道数，各使能通道的检测阈值，包括 base 及 delta
- 设置比较判断方式，包括大于检测（great 检测）、小于检测以及双向检测
- 设置检测滤波次数，各通道扫描时间以及 TSC 检测时间

- 使能硬件检测开关，由 TSC 进行定期检测。然后软件可以按照需要方式进入低功耗状态。待检测到按键触发后，硬件检测开关自动关闭。此时软件可以直接通过配置 TSC 中模拟寄存器使能对应通道并进行相关检测。如果不需要软件检测，软件可以重新使能硬件检测开关并继续按需要进行下一步操作。

  注意事项：

- 默认时钟是 LSI（RCC_CTRLSTS 默认值为 0x0c000003，RCC_CTRLSTS [0]的 LSIEN 位为 1），如果 LSIEN=0，选择 LSE，否则选择 LSI。RTC 唤醒时间间隔寄存器 TSCWKUP_CNT[13：0]对应时长是根据 RCC_CTRLSTS [0]的 LSIEN 位来控制和计算的。
- 一旦检测到有效按键后，硬件检测使能位 TSC_CTRL[6]的 HW_DET_MODE 会被自动清除，由软件处理接下来的工作。如果希望下次进入 STOP2 后使用 TSC 硬件检测，在下次进入 STOP2 前，需要把 HW_DET_MODE 重新写 1；如果本次没有检测到有效按键，HW_DET_MODE 不会被自动清除，下次进入 STOP2 前不需要重新配置。
- 如果 IWDG 使能了，那么 LSI 会被硬件自动打开。
- 应用示例：选择 LSE 时钟，通道的检测时间配置为 244.1us，运行时 TSC_OUT 触控脉冲输出在空闲无触摸时输出单个 244.1us 周期（例：输出 25 个脉冲），在触摸时输出多个 244.1us 周期（例：周期数由 TSC_CTRL[5:4]的 DET_FILTER[1:0]位决定，每个 244.1us 周期输出 7 个脉冲）。所以触摸应用中 base 值为 25，而 delta 值根据经验值一般配置为 5 左右，由此可计算出 less=base-delta=20 并配置。

28.5 TSC 寄存器

TSC 基地址：0x40003400

28.5.1 TSC 寄存器地址映像

表 28-1 TSC 寄存器映像和复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	TSC_CTRL	Reserved																		TIM2_ETR_CH1	TIM4_ETR	Reserved	DET_INTEN	GREAT_DET_SEL	LESS_DET_SEL	HW_DET_ST	HW_DET_MODE	DET_FILTER[1:0]		DET_PERIOD[3:0]			
	Reset Value																			0	0		1	1	1	0	0	0	0	0	0	1	1
004h	TSC_CHNEN	Reserved								CHN_SEL[23:0]																							
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	TSC_STS	Reserved											CHN_NUM[4:0]					Reserved		GREAT_DET	LESS_DET	Reserved	CNT_VAL[10:0]										
	Reset Value												0	0	0	0	0			0	0		0	0	0	0	0	0	0	0	0	0	0
010h	TSC_ANA_CTRL	Reserved																										SW_TSC_EN	SW_PAD_MUX[4:0]				
	Reset Value																											0	0	0	0	0	0

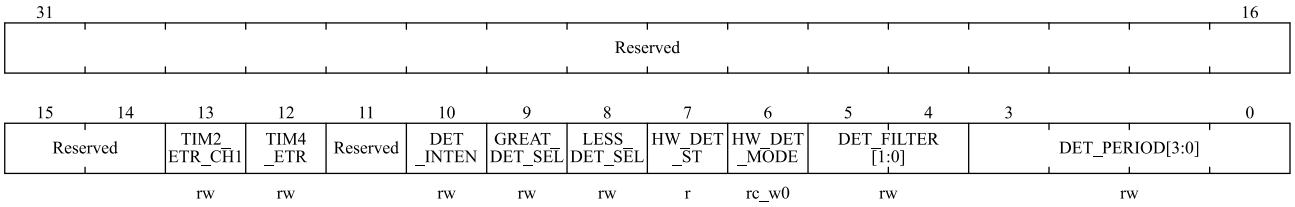
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
014h	TSC_ANA_SEL	Reserved																									PAD_OPT	SP_OPT[1:0]		Reserved			
	Reset Value																										0	0	0				
018h	TSC_RESR0	Reserved	CHN_RESIST7[2:0]			Reserved	CHN_RESIST6[2:0]			Reserved	CHN_RESIST5[2:0]			Reserved	CHN_RESIST4[2:0]			Reserved	CHN_RESIST3[2:0]			Reserved	CHN_RESIST2[2:0]			Reserved	CHN_RESIST1[2:0]			Reserved	CHN_RESIST0[2:0]		
	Reset Value		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0
01Ch	TSC_RESR1	Reserved	CHN_RESIST15[2:0]			Reserved	CHN_RESIST14[2:0]			Reserved	CHN_RESIST13[2:0]			Reserved	CHN_RESIST12[2:0]			Reserved	CHN_RESIST11[2:0]			Reserved	CHN_RESIST10[2:0]			Reserved	CHN_RESIST9[2:0]			Reserved	CHN_RESIST8[2:0]		
	Reset Value		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0
020h	TSC_RESR2	Reserved	CHN_RESIST23[2:0]			Reserved	CHN_RESIST22[2:0]			Reserved	CHN_RESIST21[2:0]			Reserved	CHN_RESIST20[2:0]			Reserved	CHN_RESIST19[2:0]			Reserved	CHN_RESIST18[2:0]			Reserved	CHN_RESIST17[2:0]			Reserved	CHN_RESIST16[2:0]		
	Reset Value		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0		1	0	0
024h	TSC_THRHD0	Reserved								DELTA0[7:0]								Reserved					BASE0[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
028h	TSC_THRHD1	Reserved								DELTA1[7:0]								Reserved					BASE1[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
02Ch	TSC_THRHD2	Reserved								DELTA2[7:0]								Reserved					BASE2[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
030h	TSC_THRHD3	Reserved								DELTA3[7:0]								Reserved					BASE3[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
034h	TSC_THRHD4	Reserved								DELTA4[7:0]								Reserved					BASE4[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
038h	TSC_THRHD5	Reserved								DELTA5[7:0]								Reserved					BASE5[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
03Ch	TSC_THRHD6	Reserved								DELTA6[7:0]								Reserved					BASE6[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
040h	TSC_THRHD7	Reserved								DELTA7[7:0]								Reserved					BASE7[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
044h	TSC_THRHD8	Reserved								DELTA8[7:0]								Reserved					BASE8[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
048h	TSC_THRHD9	Reserved								DELTA9[7:0]								Reserved					BASE9[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
04Ch	TSC_THRHD10	Reserved								DELTA10[7:0]								Reserved					BASE10[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
050h	TSC_THRHD11	Reserved								DELTA11[7:0]								Reserved					BASE11[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
054h	TSC_THRHD12	Reserved								DELTA12[7:0]								Reserved					BASE12[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
058h	TSC_THRHD13	Reserved								DELTA13[7:0]								Reserved					BASE13[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
05Ch	TSC_THRHD14	Reserved								DELTA14[7:0]								Reserved					BASE14[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
060h	TSC_THRHD15	Reserved								DELTA15[7:0]								Reserved					BASE15[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
064h	TSC_THRHD16	Reserved								DELTA16[7:0]								Reserved					BASE16[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
068h	TSC_THRHD17	Reserved								DELTA17[7:0]								Reserved					BASE17[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
06Ch	TSC_THRHD18	Reserved								DELTA18[7:0]								Reserved					BASE18[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
070h	TSC_THRHD19	Reserved								DELTA19[7:0]								Reserved					BASE19[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
074h	TSC_THRHD20	Reserved								DELTA20[7:0]								Reserved					BASE20[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
078h	TSC_THRHD21	Reserved								DELTA21[7:0]								Reserved					BASE21[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
07Ch	TSC_THRHD22	Reserved								DELTA22[7:0]								Reserved					BASE22[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0
080h	TSC_THRHD23	Reserved								DELTA23[7:0]								Reserved					BASE23[10:0]										
	Reset Value									1	0	0	0	0	0	0	0						1	0	0	0	0	0	0	0	0	0	0

28.5.2 TSC 控制寄存器（TSC_CTRL）

偏移地址：0x00

复位值：0x0000 0703



位域	名称	描述
31:14	Reserved	保留，必须保持复位值。
13	TIM2_ETR_CH1	0：无效； 1：TSC_OUT 映射到 TIM2_ETR 和 TIM2_CH1 端口。
12	TIM4_ETR	0：无效； 1：TSC_OUT 映射到 TIM4_ETR 端口。
11	Reserved	保留，必须保持复位值。
10	DET_INTEN	0：中断检测禁止； 1：中断检测使能。
9	GREAT_DET_SEL	0：无效；

位域	名称	描述
		1：使能 GREAT 阈值比较。在检测周期 DET_PERIOD[3:0]内，计数脉冲个数 CNT_VAL[10:0]大于 GREAT 阈值（GREAT 阈值= Base+delta），记为 1 次检测标记，并重新脉冲计数，检测标记次数达到 DET_FILTER[1:0]值，则识别为 1 次有效检测。
8	LESS_DET_SEL	0：无效； 1：使能 LESS 阈值比较。在检测周期 DET_PERIOD[3:0]内，计数脉冲个数 CNT_VAL[10:0]小于 LESS 阈值（LESS 阈值= Base-delta），记为 1 次检测标记，并重新脉冲计数，检测标记次数达到 DET_FILTER[1:0]值，则识别为 1 次有效检测。
7	HW_DET_ST	硬件检测状态 0：硬件检测配置为无效； 1：硬件检测配置为有效。
6	HW_DET_MODE	硬件检测模式配置 0：配置硬件检测为无效； 1：配置硬件检测为有效。
5:4	DET_FILTER[1:0]	检测连续有效滤波次数。 连续检测 1 ~ 4 次检测标记（大于 GREAT 阈值或小于 LESS 阈值），则识别为 1 次有效检测。 00：1 次； 01：2 次； 10：3 次； 11：4 次。
3:0	DET_PERIOD[3:0]	检测扫描 A/D 周期。 0000：8/TSC_CLOCK； 0001：16/TSC_CLOCK； 0002：24/TSC_CLOCK； …… 1100：104/TSC_CLOCK。

28.5.3 TSC 通道选择使能寄存器（TSC_CHNEN）

偏移地址：0x04
复位值：0x0000 0000

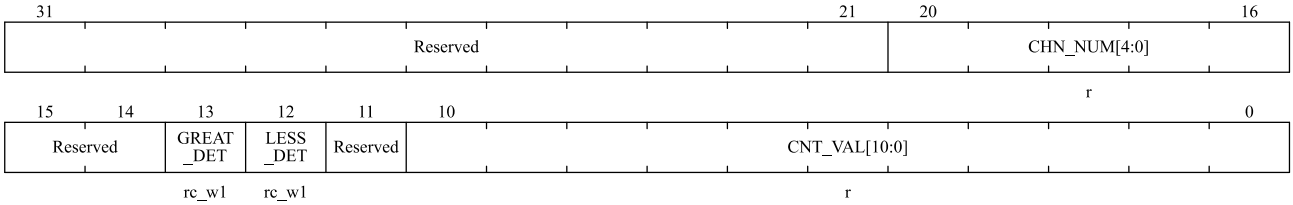
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CHN_SEL23	CHN_SEL22	CHN_SEL21	CHN_SEL20	CHN_SEL19	CHN_SEL18	CHN_SEL17	CHN_SEL16
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHN_SEL15	CHN_SEL14	CHN_SEL13	CHN_SEL12	CHN_SEL11	CHN_SEL10	CHN_SEL9	CHN_SEL8	CHN_SEL7	CHN_SEL6	CHN_SEL5	CHN_SEL4	CHN_SEL3	CHN_SEL2	CHN_SEL1	CHN_SEL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:24	Reserved	保留，必须保持复位值。

位域	名称	描述
23:0	CHN_SELx	硬件通道 x 选择使能位 0： 禁用通道 x； 1：选择使能通道 x。

28.5.4 TSC 状态寄存器（TSC_STS）

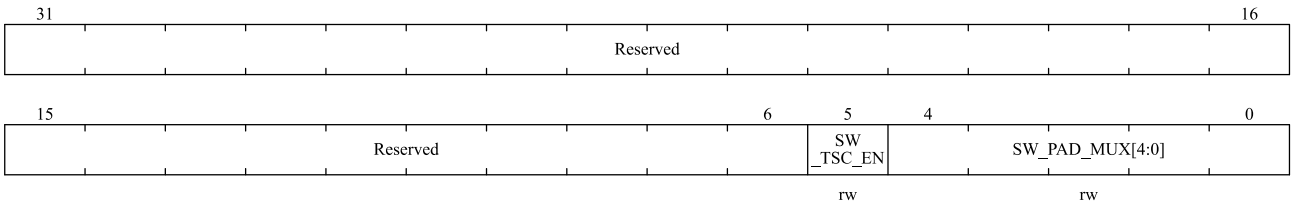
偏移地址：0x08
复位值：0x0000 0000



位域	名称	描述
31:21	Reserved	保留，必须保持复位值。
20:16	CHN_NUM[4:0]	检测的通道序号
15:14	Reserved	保留，必须保持复位值。
13	GREAT_DET	0：无效； 1：有效。
12	LESS_DET	0：无效； 1：有效。
11	Reserved	保留，必须保持复位值。
10:0	CNT_VAL[10:0]	当前通道计数值

28.5.5 TSC 模拟控制寄存器（TSC_ANA_CTRL）

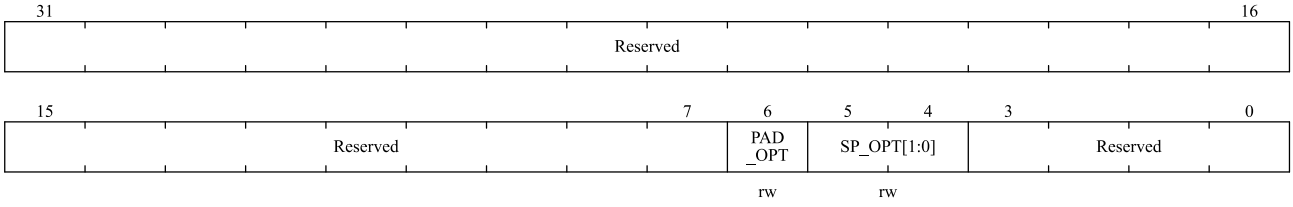
偏移地址：0x10
复位值：0x0000 0000



位域	名称	描述
31:6	Reserved	保留，必须保持复位值。
5	SW_TSC_EN	TSC 软件使能有效位（需先配置 HW_DET_MODE 为无效）。 0：无效 1：使能
4:0	SW_PAD_MUX[4:0]	TSC 检测通道软件选择位（对应 TSC_THRHDx，x=0~23）。

28.5.6 TSC 模拟选择寄存器（TSC_ANA_SEL）

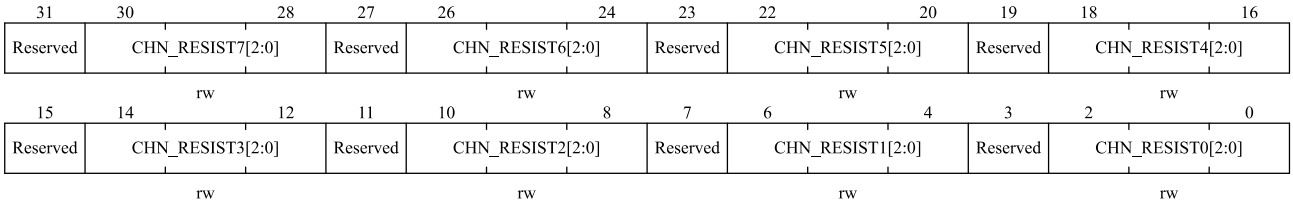
偏移地址：0x14
复位值：0x0000 0008



位域	名称	描述
31:7	Reserved	保留，必须保持复位值。
6	PAD_OPT	PAD 充电选择项 0：关闭，采用内部电阻； 1：开启 PAD 充电，采用外部电阻。
5:4	SP_OPT[1:0]	TSC 速度配置档位 00：针对低速 TSC（100K 左右）； 01/10：针对中速 TSC； 11：针对高速 TSC。
3:0	Reserved	保留，必须保持复位值。

28.5.7 TSC 内部电阻选择寄存器 0（TSC_RESR0）

偏移地址：0x18
复位值：0x4444 4444



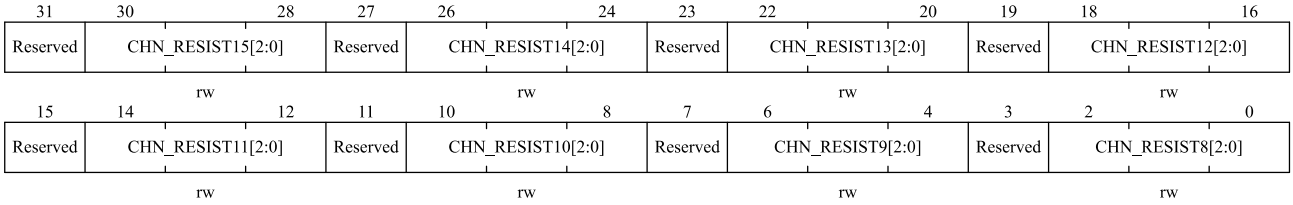
位域	名称	描述
31	Reserved	保留，必须保持复位值。
30:28	CHN_RESIST7[2:0]	选择通道 7 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
27	Reserved	保留，必须保持复位值。
26:24	CHN_RESIST6[2:0]	选择通道 6 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm；

位域	名称	描述
		…… 110：250K ohm； 111：125K ohm。
23	Reserved	保留，必须保持复位值。
22:20	CHN_RESIST5[2:0]	选择通道 5 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
19	Reserved	保留，必须保持复位值。
18:16	CHN_RESIST4[2:0]	选择通道 4 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
15	Reserved	保留，必须保持复位值。
14:12	CHN_RESIST3[2:0]	选择通道 3 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
11	Reserved	保留，必须保持复位值。
10:8	CHN_RESIST2[2:0]	选择通道 2 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
7	Reserved	保留，必须保持复位值。
6:4	CHN_RESIST1[2:0]	选择通道 1 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
3	Reserved	保留，必须保持复位值。
2:0	CHN_RESIST0[2:0]	选择通道 0 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm；

位域	名称	描述
		…… 110：250K ohm； 111：125K ohm。

28.5.8 TSC 内部电阻选择寄存器 1（TSC_RESR1）

偏移地址：0x1C
复位值：0x4444 4444



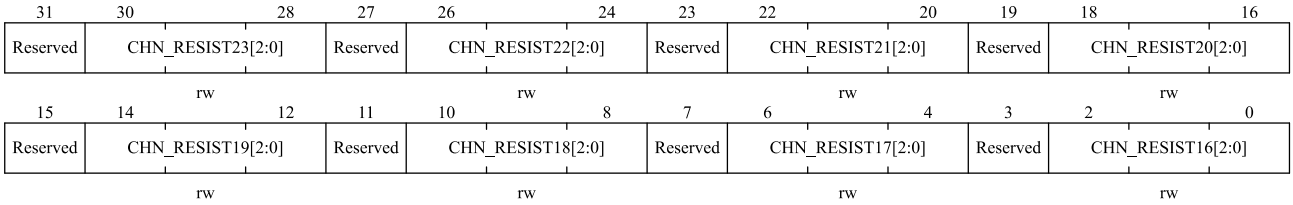
位域	名称	描述
31	Reserved	保留，必须保持复位值。
30:28	CHN_RESIST15[2:0]	选择通道 15 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
27	Reserved	保留，必须保持复位值。
26:24	CHN_RESIST14[2:0]	选择通道 14 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
23	Reserved	保留，必须保持复位值。
22:20	CHN_RESIST13[2:0]	选择通道 13 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
19	Reserved	保留，必须保持复位值。
18:16	CHN_RESIST12[2:0]	选择通道 12 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm；

位域	名称	描述
		111：125K ohm。
15	Reserved	保留，必须保持复位值。
14:12	CHN_RESIST11[2:0]	选择通道 11 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
11	Reserved	保留，必须保持复位值。
10:8	CHN_RESIST10[2:0]	选择通道 10 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
7	Reserved	保留，必须保持复位值。
6:4	CHN_RESIST9[2:0]	选择通道 9 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
3	Reserved	保留，必须保持复位值。
2:0	CHN_RESIST8[2:0]	选择通道 8 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。

28.5.9 TSC 内部电阻选择寄存器 2（TSC_RESR2）

偏移地址：0x20

复位值：0x4444 4444



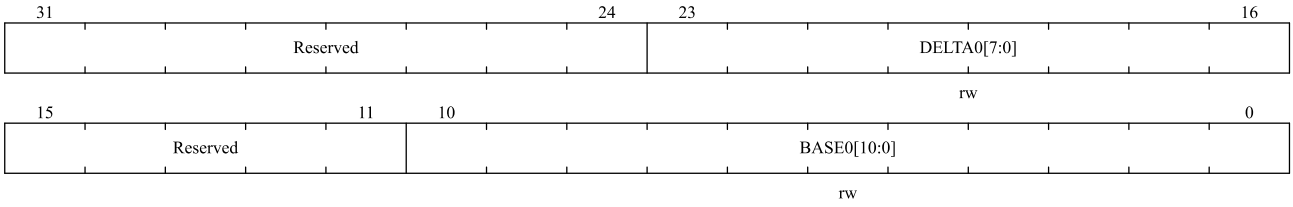
位域	名称	描述
31	Reserved	保留，必须保持复位值。

位域	名称	描述
30:28	CHN_RESIST23[2:0]	选择通道 23 的内部电阻值（step=125K ohm） 000：1M ohm; 001：875K ohm; …… 110：250K ohm; 111：125K ohm。
27	Reserved	保留，必须保持复位值。
26:24	CHN_RESIST22[2:0]	选择通道 22 的内部电阻值（step=125K ohm） 000：1M ohm; 001：875K ohm; …… 110：250K ohm; 111：125K ohm。
23	Reserved	保留，必须保持复位值。
22:20	CHN_RESIST21[2:0]	选择通道 21 的内部电阻值（step=125K ohm） 000：1M ohm; 001：875K ohm; …… 110：250K ohm; 111：125K ohm。
19	Reserved	保留，必须保持复位值。
18:16	CHN_RESIST20[2:0]	选择通道 20 的内部电阻值（step=125K ohm） 000：1M ohm; 001：875K ohm; …… 110：250K ohm; 111：125K ohm。
15	Reserved	保留，必须保持复位值。
14:12	CHN_RESIST19[2:0]	选择通道 19 的内部电阻值（step=125K ohm） 000：1M ohm; 001：875K ohm; …… 110：250K ohm; 111：125K ohm。
11	Reserved	保留，必须保持复位值。
10:8	CHN_RESIST18[2:0]	选择通道 18 的内部电阻值（step=125K ohm） 000：1M ohm; 001：875K ohm; …… 110：250K ohm; 111：125K ohm。
7	Reserved	保留，必须保持复位值。

位域	名称	描述
6:4	CHN_RESIST17[2:0]	选择通道 17 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。
3	Reserved	保留，必须保持复位值。
2:0	CHN_RESIST16[2:0]	选择通道 16 的内部电阻值（step=125K ohm） 000：1M ohm； 001：875K ohm； …… 110：250K ohm； 111：125K ohm。

28.5.10 TSC 阈值寄存器 0（TSC_THRHD0）

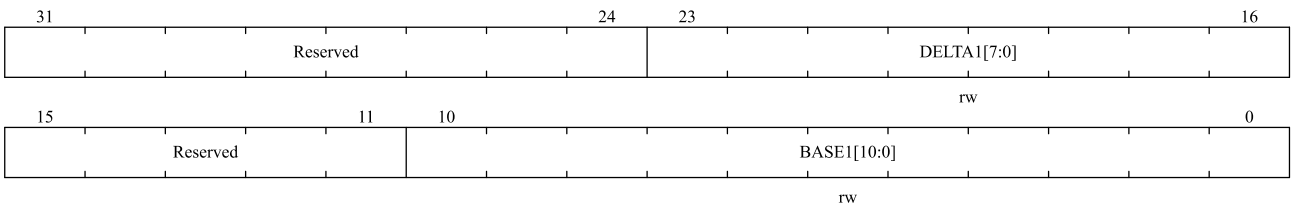
偏移地址：0x24
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA0[7:0]	通道 0 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE0[10:0]	通道 0 比较 base。

28.5.11 TSC 阈值寄存器 1（TSC_THRHD1）

偏移地址：0x28
复位值：0x0080 0400

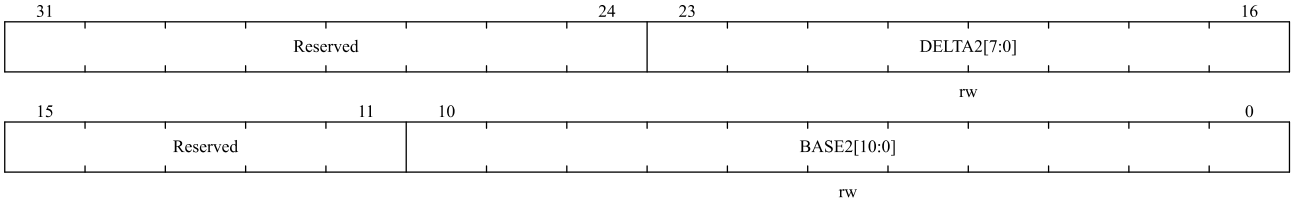


位域	名称	描述
31:24	Reserved	保留，必须保持复位值。

位域	名称	描述
23:16	DELTA1[7:0]	通道 1 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE1[10:0]	通道 1 比较 base。

28.5.12 TSC 阈值寄存器 2（TSC_THRHD2）

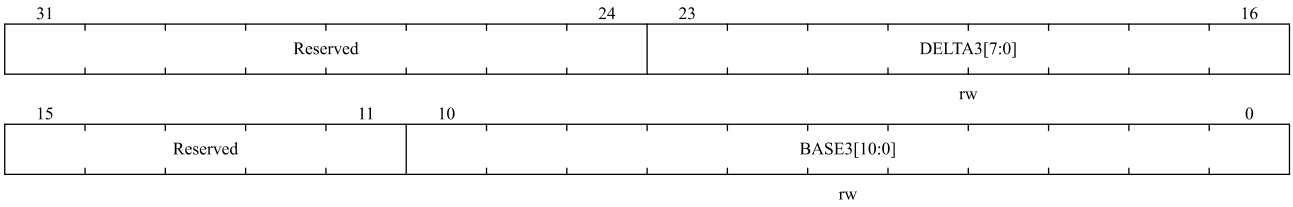
偏移地址：0x2C
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA2[7:0]	通道 2 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE2[10:0]	通道 2 比较 base。

28.5.13 TSC 阈值寄存器 3（TSC_THRHD3）

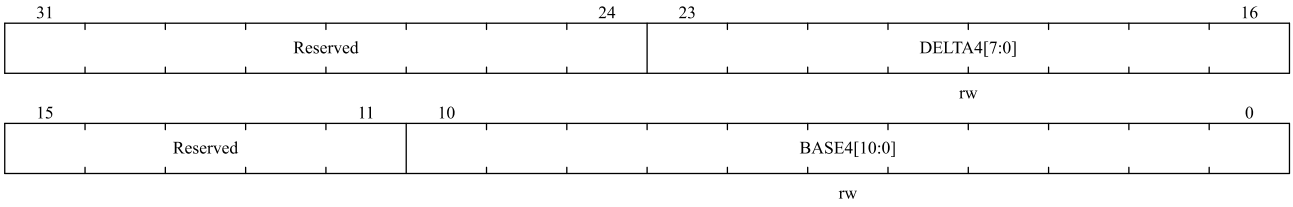
偏移地址：0x30
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA3[7:0]	通道 3 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE3[10:0]	通道 3 比较 base。

28.5.14 TSC 阈值寄存器 4（TSC_THRHD4）

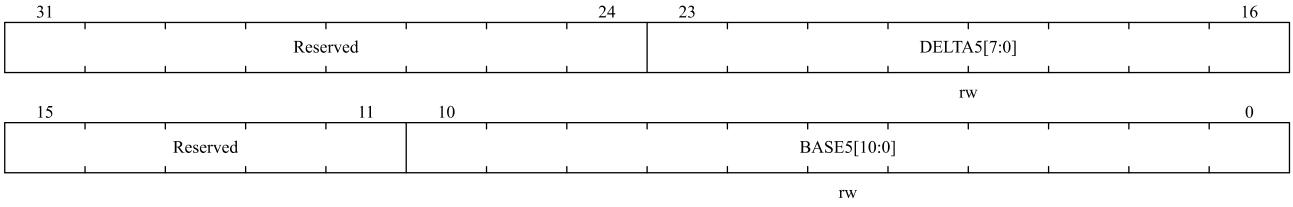
偏移地址：0x34
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA4[7:0]	通道 4 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE4[10:0]	通道 4 比较 base。

28.5.15 TSC 阈值寄存器 5（TSC_THRHD5）

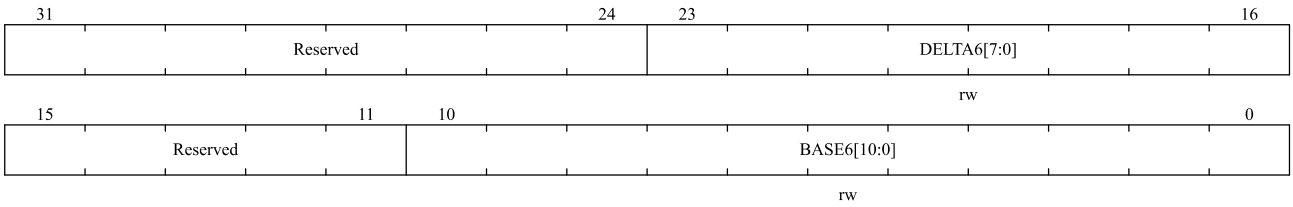
偏移地址：0x38
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA5[7:0]	通道 5 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE5[10:0]	通道 5 比较 base。

28.5.16 TSC 阈值寄存器 6（TSC_THRHD6）

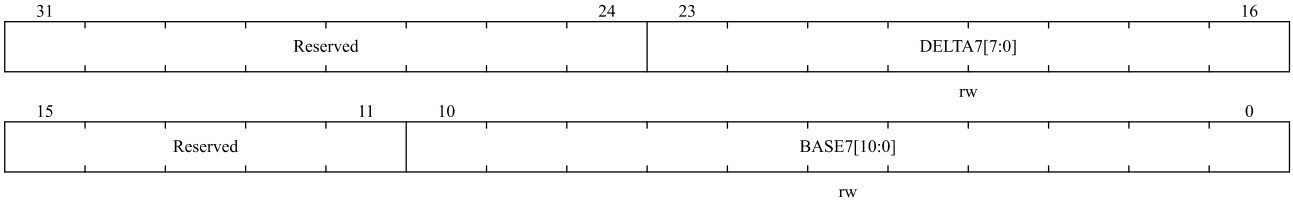
偏移地址：0x3C
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA6[7:0]	通道 6 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE6[10:0]	通道 6 比较 base。

28.5.17 TSC 阈值寄存器 7（TSC_THRHD7）

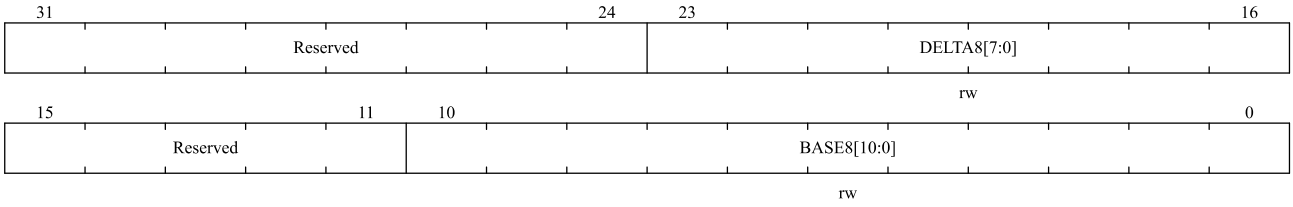
偏移地址：0x40
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA7[7:0]	通道 7 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE7[10:0]	通道 7 比较 base。

28.5.18 TSC 阈值寄存器 8（TSC_THRHD8）

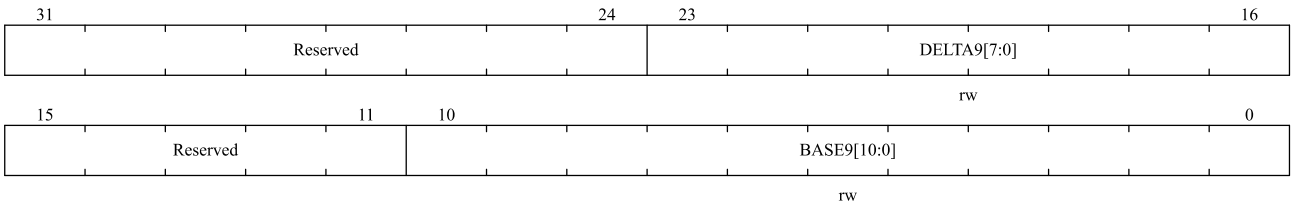
偏移地址：0x44
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA8[7:0]	通道 8 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE8[10:0]	通道 8 比较 base。

28.5.19 TSC 阈值寄存器 9（TSC_THRHD9）

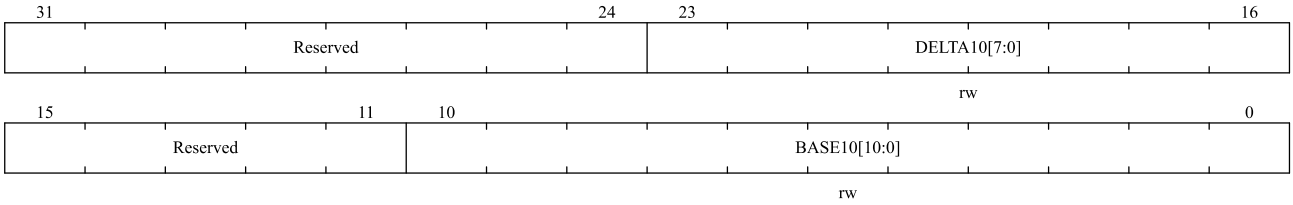
偏移地址：0x48
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA9[7:0]	通道 9 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE9[10:0]	通道 9 比较 base。

28.5.20 TSC 阈值寄存器 10（TSC_THRHD10）

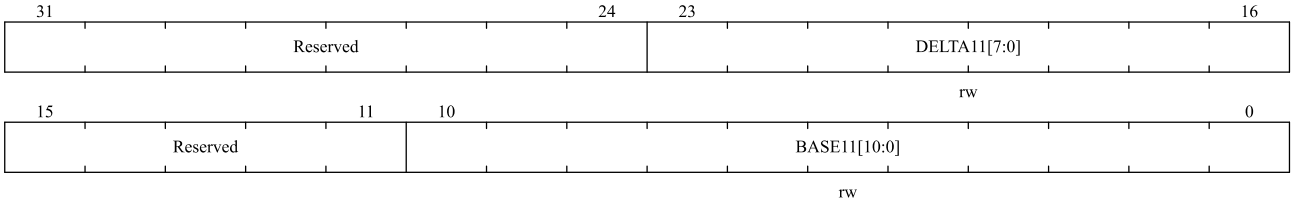
偏移地址：0x4C
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA10[7:0]	通道 10 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE10[10:0]	通道 10 比较 base。

28.5.21 TSC 阈值寄存器 11（TSC_THRHD11）

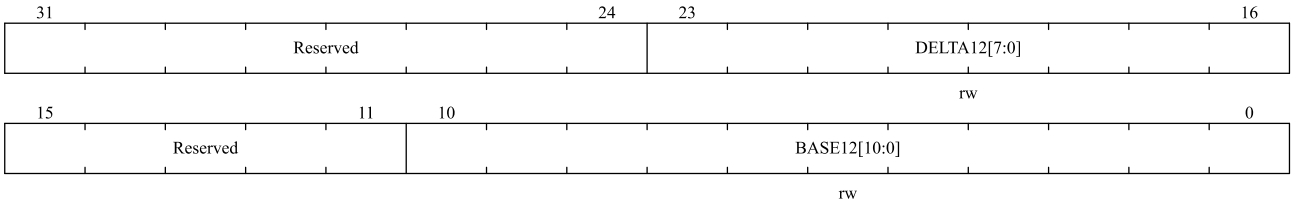
偏移地址：0x50
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA11[7:0]	通道 11 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE11[10:0]	通道 11 比较 base。

28.5.22 TSC 阈值寄存器 12（TSC_THRHD12）

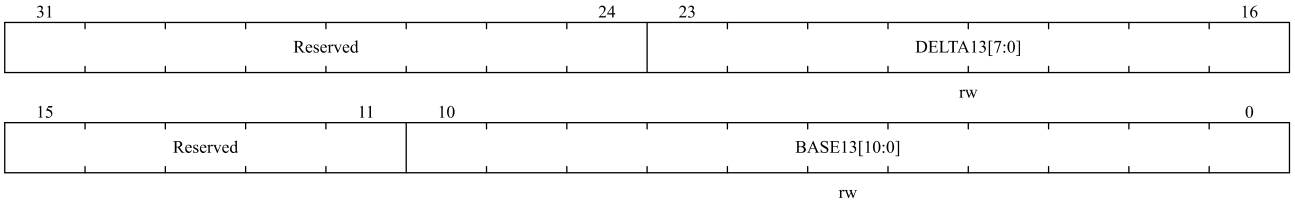
偏移地址：0x54
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA12[7:0]	通道 12 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE12[10:0]	通道 12 比较 base。

28.5.23 TSC 阈值寄存器 13（TSC_THRHD13）

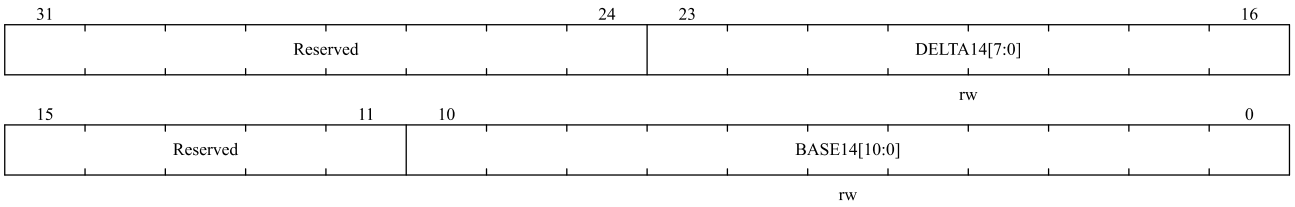
偏移地址：0x58
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA13[7:0]	通道 13 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE13[10:0]	通道 13 比较 base。

28.5.24 TSC 阈值寄存器 14（TSC_THRHD14）

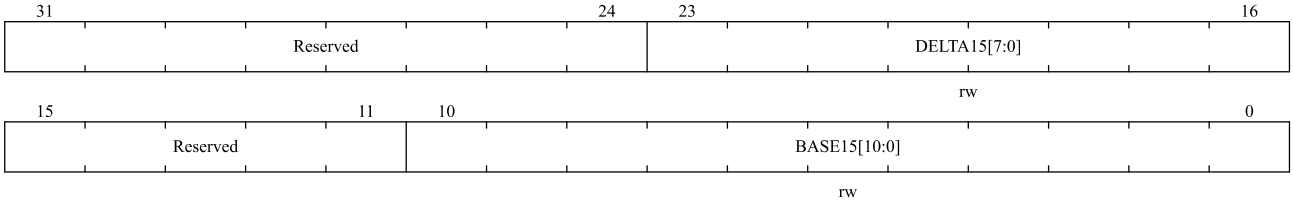
偏移地址：0x5C
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA14[7:0]	通道 14 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE14[10:0]	通道 14 比较 base。

28.5.25 TSC 阈值寄存器 15（TSC_THRHD15）

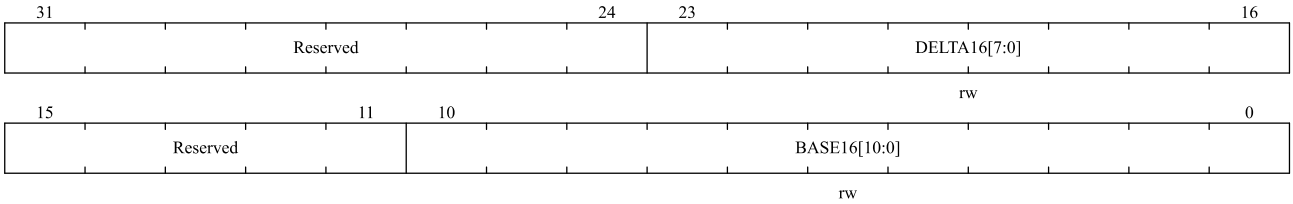
偏移地址：0x60
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA15[7:0]	通道 15 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE15[10:0]	通道 15 比较 base。

28.5.26 TSC 阈值寄存器 16（TSC_THRHD16）

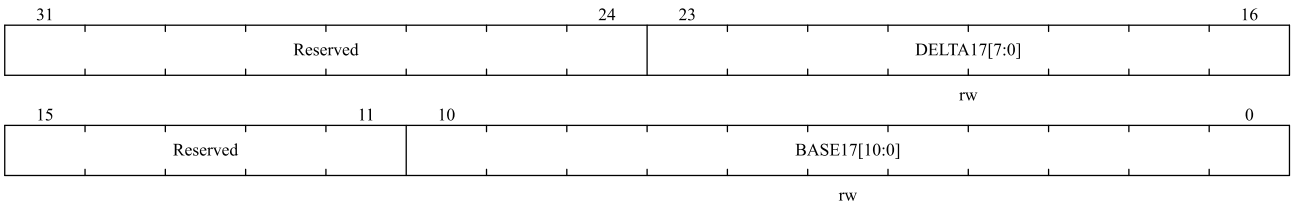
偏移地址：0x64
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA16[7:0]	通道 16 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE16[10:0]	通道 16 比较 base。

28.5.27 TSC 阈值寄存器 17（TSC_THRHD17）

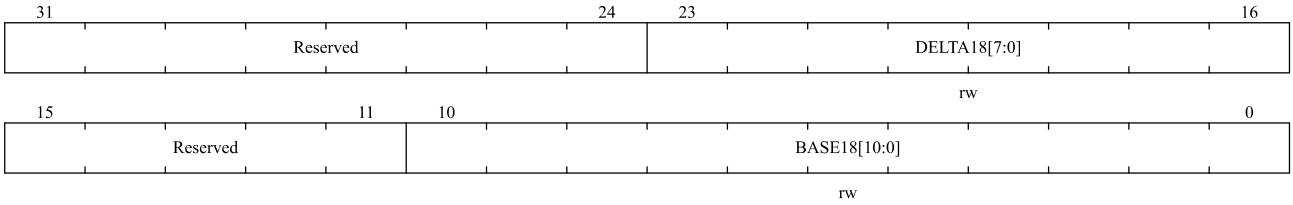
偏移地址：0x68
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA17[7:0]	通道 17 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE17[10:0]	通道 17 比较 base。

28.5.28 TSC 阈值寄存器 18（TSC_THRHD18）

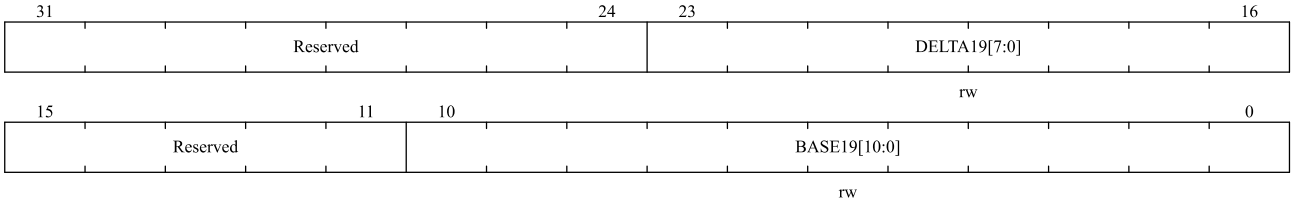
偏移地址：0x6C
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA18[7:0]	通道 18 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE18[10:0]	通道 18 比较 base。

28.5.29 TSC 阈值寄存器 19（TSC_THRHD19）

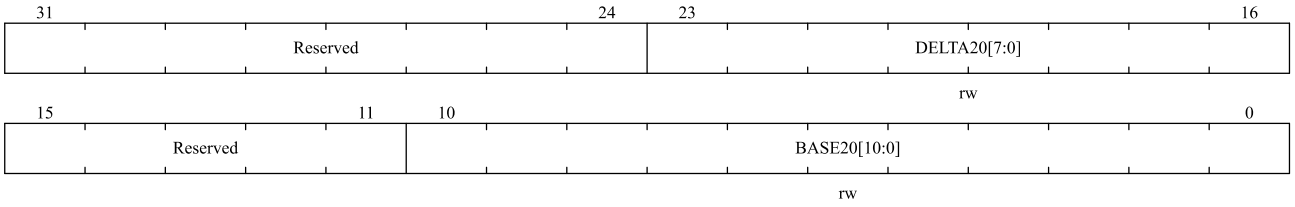
偏移地址：0x70
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA19[7:0]	通道 19 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE19[10:0]	通道 19 比较 base。

28.5.30 TSC 阈值寄存器 20（TSC_THRHD20）

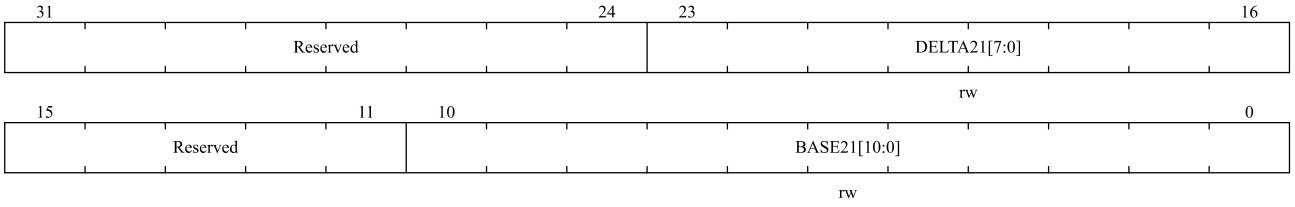
偏移地址：0x74
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA20[7:0]	通道 20 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE20[10:0]	通道 20 比较 base。

28.5.31 TSC 阈值寄存器 21（TSC_THRHD21）

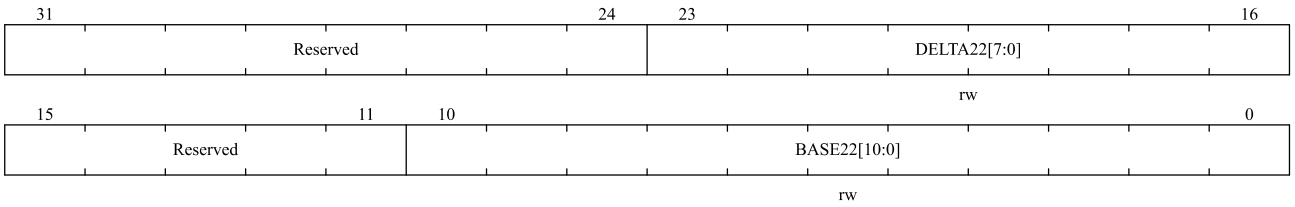
偏移地址：0x78
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA21[7:0]	通道 21 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE21[10:0]	通道 21 比较 base。

28.5.32 TSC 阈值寄存器 22（TSC_THRHD22）

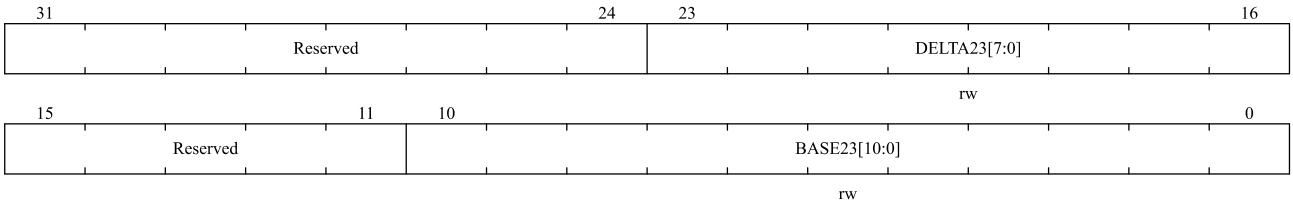
偏移地址：0x7C
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA22[7:0]	通道 22 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE22[10:0]	通道 22 比较 base。

28.5.33 TSC 阈值寄存器 23（TSC_THRHD23）

偏移地址：0x80
复位值：0x0080 0400



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23:16	DELTA23[7:0]	通道 23 比较 delta。
15:11	Reserved	保留，必须保持复位值。
10:0	BASE23[10:0]	通道 23 比较 base。

29 调试支持（DBG）

29.1 简介

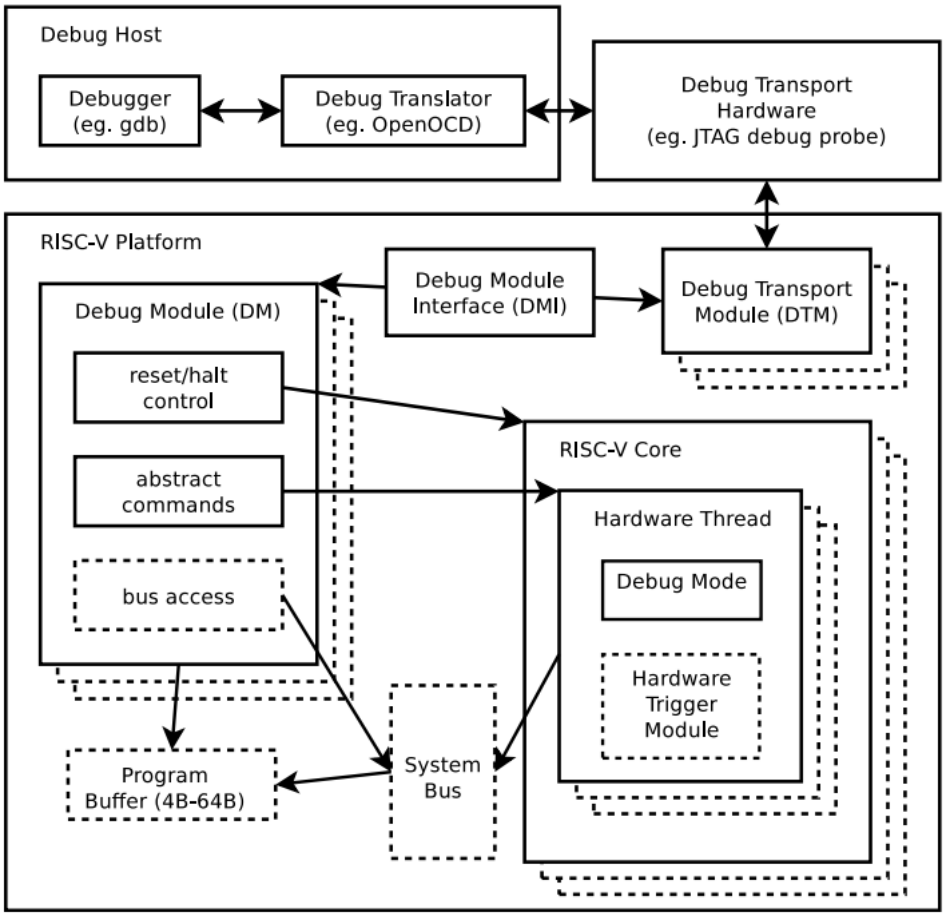
CM32M4xxR 使用 RISC-V N308 内核，内核集成硬件调试模块。支持指令断点（指令取值时停止）和数据断点（数据访问时停止）。当内核停止时，用户可以查看内核的内部状态和系统的外部状态。用户查询操作完成后，可以使内核和外设恢复，并继续执行相应程序。

CM32M4xxR 内核的硬件调试模块在连接到调试器时即可被使用（在未被禁止情况下）。

CM32M4xxR 支持以下调试接口：

- 4-wire JTAG 接口
- 2-wire JTAG 接口

图 29-1 CM32M4xxR 和 N308 的调试框图



RISC-V N308 内核硬件调试系统支持低功耗模式调试和对部分外设的调试。支持调试的外设包括：CAN、I2C 接口和 TIMER、WWDG、IWDG 模块。用户进行低功耗调试或者外设调试时，需要将调试控制寄存器（DBG_CTRL）的相应位置 1。

29.2 JTAG 功能

调试工具可以通过上述的 4-wire JTAG 调试接口或者 2-wire JTAG 调试接口来调用调试功能。

29.2.1 切换 JTAG 接口

芯片默认使用 4-wire JTAG 调试接口，如需切换到 2-wire JTAG 调试接口，相关描述请参考 IEEE-1149.7 标准。

29.2.2 引脚分配

4-wire JTAG 调试接口包含 4 个管脚：JTCK（JTAG 时钟管脚），JTMS（JTAG 模式选择引脚），JTDI（JTAG 数据输入管脚），JTDO（JTAG 数据输出管脚）。

2-wire JTAG 调试接口包含 2 个管脚：JTCK（JTAG 时钟管脚），JTMS（JTAG 模式选择引脚）。

JTAG 调试接口管脚分配见下表：

表 29-1 调试端口引脚

调试端口	引脚分配
JTMS（2 线复用）	PA13
JTCK（2 线复用）	PA14
JTDI	PA15
JTDO	PB3

- 4-wire JTAG 调试接口和 2-wire JTAG 调试接口都使能的情况下，复位后默认使用 4-wire JTAG 调试接口。
- 使用两线协议接口时，JTDI（PA15）、JTDO（PB3）两个管脚可作为通用 GPIO 使用。
- 不使用调试功能时，上述 4 个管脚都可作为通用 GPIO 使用。

29.3 MCU 调试功能

29.3.1 低功耗模式支持

CM32M4xxR 可以提供多种低功耗模式（详细参考 3.电源控制章节）。在进行调试时，要保证内核的 FCLK 和 HCLK 是开启状态，为内核调试提供必要的时钟。用户可以根据特定的操作（保证低功耗模式下 FCLK 或 HCLK 的输出），在低功耗模式下进行 MCU 调试。

用户想要在进行低功耗模式下调试 MCU，则首先需要调试器配置低功耗模式相关的寄存器：

- SLEEP 模式：

  调试器需配置 DBG_CTRL 寄存器的 SLEEP 位（配置 HCLK 输出同 FCLK 相同频率）。

- STOP 模式：

调试器需配置 DBG_CTRL 寄存器的 STOP 位，启动内部 RC 振荡器，为 HCLK 和 FCLK 提供时钟。

29.3.2 外设调试支持

当 DBG_CTRL 寄存器中外设控制位的相应位置 1 时，内核停止后相应的外设进入调试状态：

- 定时器外设：定时器的计数器停止并进行调试；
- I2C 外设：I2C 的 SMBUS 保持状态并进行调试；
- WWDG/IWDG 外设：WWDG/IWDG 计数器时钟停止并进行调试；
- CAN 外设：CAN 接口接收寄存器停止计数并进行调试。

29.4 寄存器

29.4.1 DBG 寄存器地址映像

下面列出了 DBG 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。寄存器的基地址为 0xE0040000。

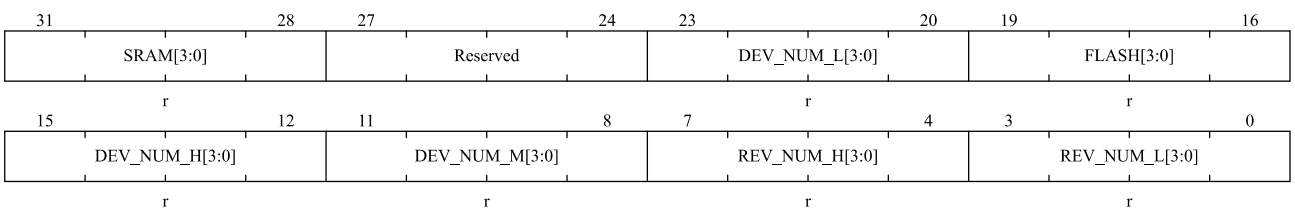
表 29-2 DBG 寄存器地址映像及复位值

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	DBG_ID	SRAM[3:0]				Reserved				DEV_NUM_L[3:0]				FLASH[3:0]				DEV_NUM_H[3:0]				DEV_NUM_M[3:0]				REV_NUM_H[3:0]				REV_NUM_L[3:0]			
	Reset Value	x	x	x	x					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
004h	DBG_CTRL	Reserved								JTAG_OVERRIDE_DM_SLEEP	JTAG_DWBYPASS	CAN2_STOP	TIM7_STOP	TIM6_STOP	TIM5_STOP	TIM8_STOP	I2C2SMBUS_TIMEOUT	I2C1SMBUS_TIMEOUT	CAN1_STOP	TIM4_STOP	TIM3_STOP	TIM2_STOP	TIM1_STOP	WWDG_STOP	IWDG_STOP	TRACE_MODE		TRACE_IOEN	Reserved	STDBY	STOP	SLEEP	
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	

29.4.2 ID 寄存器（DBG_ID）

CM32M4xxR 内含一个 MCU DBG ID 编码。这个 ID 定义了 MCU 的相关信息。

地址偏移：0x00
只支持 32 位访问，固定值不可修改。

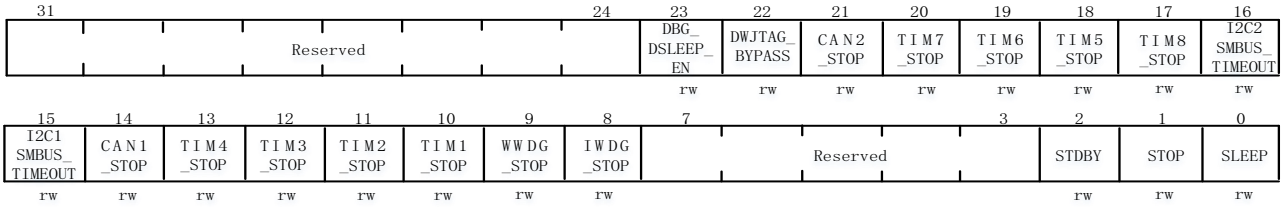


位域	名称	描述
31:28	SRAM[3:0]	SRAM 容量指示位。 芯片 SRAM 容量为 (SRAM[3:0] + 1) * 16KB
27:24	Reserved	保留，必须保持复位值。
23:20	DEV_NUM_L[3:0]	设备型号的低 4 位。 设备型号由高、中、低共 12 位组成，代表芯片的型号。取值如下: ■ 0x4xx：基本型 CM32M4xxR
19:16	FLASH[3:0]	FLASH 容量指示位。 芯片 FLASH 容量为 (FLASH[3:0]+1) * 64KB
15:12	DEV_NUM_H[3:0]	设备型号的高 4 位。 见 DEV_NUM_L[3:0]的说明。
11:8	DEV_NUM_M[3:0]	设备型号的中 4 位。 见 DEV_NUM_L[3:0]的说明。
7:4	REV_NUM_H[3:0]	芯片版本号高 4 位。
3:0	REV_NUM_L[3:0]	芯片版本号低 4 位。

29.4.3 调试控制寄存器（DBG_CTRL）

地址偏移：0x04

POR 复位值：0x0000 0000（不被系统复位所复位）



位域	名称	描述
31:24	Reserved	保留，必须保持复位值。
23	DBG_DSLEEP_EN	CPU deep sleep 模式时 debug 模块时钟使能 0：deep sleep 模式时 debug 模块时钟屏蔽 1：deep sleep 模式时 debug 模块时钟使能
22	DWJTAG_BYPASS	2-wire JTAG 功能屏蔽位。、 0：不屏蔽 2-wire JTAG 功能 1：屏蔽 2-wire JTAG 功能
21	CAN2_STOP	当内核进入调试状态时，CAN2 停止运行。 软件置 1 或清零。 0：CAN2 仍然正常运行; 1：CAN2 的接收寄存器不继续接收数据。
20:17	TIMx_STOP	当核心停止时停止定时器计数器（x=8,7,6,5）。 软件置 1 或清零。 0：当核心停止时，仍然向相关定时器的计数器提供时钟，定时器输出工

位域	名称	描述
		作正常； 1：当核心停止时，切断相关定时器的计数器的时钟，同时关闭定时器的输出（就好象对某一暂停事件的紧急响应，停止定时器）。
16:15	I2CxSMBUS_TIMEOUT	当核心停止时停止 SMBUS 超时模式（x=1,2）。 软件置 1 或清零。 0：与正常模式操作相同； 1：冻结 SMBUS 的超时控制。
14	CAN1_STOP	当内核进入调试状态时，CAN1 停止运行。 软件置 1 或清零。 0：CAN1 仍然正常运行； 1：CAN1 的接收寄存器不继续接收数据。
13:10	TIMx_STOP	当内核进入调试状态时计数器停止工作（x=4,3,2,1）。 软件置 1 或清零。 0：选中定时器的计数器仍然正常工作； 1：选中定时器的计数器停止工作。
9	WWDG_STOP	当内核进入调试状态时调试窗口看门狗停止工作。 软件置 1 或清零。 0：窗口看门狗计数器仍然正常工作； 1：窗口看门狗计数器停止工作。
8	IWDG_STOP	当内核进入调试状态时看门狗停止工作。 软件置 1 或清零。 0：看门狗计数器仍然正常工作； 1：看门狗计数器停止工作。
7:3	Reserved	保留，必须保持复位值。
2	STDBY	调试待机模式。 软件置 1 或清零。 0：（FCLK 关，HCLK 关）整个数字电路部分都断电。从软件的观点看，退出 STANDBY 模式与复位是一样的（除了一些状态位指示了微控制器刚从 STANDBY 状态退出）。 1：（FCLK 开，HCLK 开）数字电路部分不下电，FCLK 和 HCLK 时钟由内部 RC 振荡器提供时钟。另外，微控制器通过产生系统复位来退出 STANDBY 模式和复位是一样的。
1	STOP	调试停止模式。 软件置 1 或清零。 0：（FCLK 关，HCLK 关）在停止模式时，时钟控制器禁止一切时钟（包括 HCLK 和 FCLK）。当从 STOP 模式退出时，时钟的配置和复位之后的配置一样（微控制器由 8MHz 的内部 RC 振荡器（HSI）提供时钟）。因此，软件必需重新配置时钟控制系统启动 PLL，晶振等。 1：（FCLK 开，HCLK 开）在停止模式时，FCLK 和 HCLK 时钟由内部 RC 振荡器提供。当退出停止模式时，软件必需重新配置时钟系统启动 PLL，晶振等（与配置此比特位为 0 时的操作一样）。

位域	名称	描述
		注：此位为 1 且进入 STOP2 调试停止模式后，外部 NRST 管脚复位不能唤醒芯片，调试中需注意。
0	SLEEP	调试睡眠模式。 软件置 1 或清零。 0：（FCLK 开，HCLK 关）在睡眠模式时，FCLK 由原先已配置好的系统时钟提供，HCLK 则关闭。由于睡眠模式不会复位已配置好的时钟系统，因此从睡眠模式退出时，软件不需要重新配置时钟系统。 1：（FCLK 开，HCLK 开）在睡眠模式时，FCLK 和 HCLK 时钟都由原先配置好的系统时钟提供。

30 唯一设备序列号（UID/UCID）

30.1 简介

CM32M4xxR 系列产品内置两个不同长度的唯一设备序列号，分别为 96 位的 UID(Unique device ID)和 128 位的 UCID(Unique Customer ID)，这两个设备序列号存放在闪存存储器的系统配置块中，它们所包含的信息在出厂时编写，并保证任意一个 CM32M4xxR 微控制器在任何情况下都是唯一的，用户应用程序或外部设备可以通过 CPU 或 JTAG 接口读取，不可被修改。

UCID128 位，UID 为 96 位，通常用来做为序列号或作为密码，在编写闪存时，将此唯一标识与软件加解密算法相结合，进一步提高代码在闪存存储器内的安全性，也可用于激活带安全功能的自举程序(Secure Bootloader)。

UCID 和 UID，遵守中国移动芯片序列号定义，它包含芯片生产及版本相关信息。

30.2 UID 寄存器

UID 长度 96 位。每字节的地址及代表的信息如下表所示。

表 30-1 UID 位描述

编号	地址	描述	位数
1	0x1FFF_F7C0	厂商代码	8bit
2	0x1FFF_F7F4	芯片版本号	8bit
3	0x1FFF_F7C7	LOT ID	8bit
4	0x1FFF_F7C8		8bit
5	0x1FFF_F7C9		8bit
6	0x1FFF_F7CA		8bit
7	0x1FFF_F7CB		8bit
8	0x1FFF_F7CC		8bit
9	0x1FFF_F7CD		8bit
10	0x1FFF_F7F0	Wafer 号	8bit
11	0x1FFF_F7F1	Wafer X 坐标	8bit
12	0x1FFF_F7F2	Wafer Y 坐标	8bit

30.3 UCID 寄存器

UCID 长度 128 位。每字节的地址及代表的信息如下表所示。

表 30-2 UCID 位描述

编号	地址	描述	位数
1	0x1FFF_F7C0	厂商代码	8bit
2	0x1FFF_F7F4	芯片版本号	8bit
3	0x1FFF_F7C4	生产年份	8bit

编号	地址	描述	位数
4	0x1FFF_F7C5	生产月份	8bit
5	0x1FFF_F7C6	生产日期	8bit
6	0x1FFF_F7C7	LOT ID	8bit
7	0x1FFF_F7C8		8bit
8	0x1FFF_F7C9		8bit
9	0x1FFF_F7CA		8bit
10	0x1FFF_F7CB		8bit
11	0x1FFF_F7CC		8bit
12	0x1FFF_F7CD		8bit
13	0x1FFF_F7CE		8bit
14	0x1FFF_F7F0	Wafer 号	8bit
15	0x1FFF_F7F1	Wafer X 坐标	8bit
16	0x1FFF_F7F2	Wafer Y 坐标	8bit

31 文档版本历史

版本	变更日期	变更描述
V1.0	2021.10.26	初始版本

32 声明

中移物联网有限公司（以下简称中移物联）保有在不事先通知而修改这份文档的权利。中移物联认为提供的信息是准确可信的。尽管这样，中移物联对文档中可能出现的错误不承担任何责任。在购买前请联系中移物联获取该器件说明的最新版本。对于使用该器件引起的专利纠纷及第三方侵权中移物联不承担任何责任。另外，中移物联的产品不建议应用于生命相关的设备和系统，在使用该器件中因为设备或系统运转失灵而导致的损失中移物联不承担任何责任。中移物联对本手册拥有版权等知识产权，受法律保护。未经中移物联许可，任何单位及个人不得以任何方式或理由对本手册进行使用、复制、修改、抄录、传播等。